



Re- accredited with A ++ Grade (CGPA 3.65) by NAAC  
Category- I University Status awarded by UGC

No.2026/May/AAMS-III/C-83/42997

Date: 08<sup>th</sup> May, 2026

**CIRCULAR:-**

Attention of all the Principals of the Affiliated Colleges & Autonomous Colleges, all Academic Heads of University Departments, the Directors of the Recognized Institutions/Centers, Director, Centre for Distance & Online Education, Director, Dharmaveer Anand Dighe Thane Sub-Campus, Co-ordinator, School of Engineering and Applied Sciences, Kalyan Sub-Campus, Director, Chitrakar Padmabhushan Dr.Dhananjay Keer Ratnagiri Sub-Campus, Director, Sindhudurg Sub-Campus, Principal, Vishwabhushan Bharatratna Dr. B. A. Ambedkar College, Ambadave, Ratnagiri and Principal, V.V. Dalvie College, Talere, Sindhudurg is invited to this office Circular No. AAMS/ICD/2025-26/ 37 of dated 27 May, 2025 relating to the NEP UG & PG Syllabus.

They are hereby informed that the recommendations made by the **Ad-hoc Board of Studies in Computer Science** at its meeting held on 16<sup>th</sup> February, 2026 vide item No.1 and subsequently passed by the Board of Deans at its meeting held on 17<sup>th</sup> March, 2026 vide item No. 6.14 (N) have been accepted by the Academic Council at its meeting held on 25<sup>th</sup> March, 2026 vide item No. 6.33 (N). In accordance therewith syllabus of **B.Sc. (Computer Science) (Scheme - I) (Sem V & VI)** (NEP 2020) is introduced as per appendix with effect from the academic year 2026-27.

(The said circular is available on the University's website [www.mu.ac.in](http://www.mu.ac.in)).

MUMBAI – 400 032  
08<sup>th</sup> May, 2026

(Dr. Prasad Karande)  
REGISTRAR

To,

All the Principals of the Affiliated Colleges & Autonomous Colleges, all Academic Heads of University Departments, the Directors of the Recognized Institutions/Centers, Director, Centre for Distance & Online Education, Director, Dharmaveer Anand Dighe Thane Sub-Campus, Co-ordinator, School of Engineering and Applied Sciences, Kalyan Sub-Campus, Director, Chitrakar Padmabhushan Dr.Dhananjay Keer Ratnagiri Sub-Campus, Director, Sindhudurg Sub-Campus, Principal, Vishwabhushan Bharatratna Dr. B. A. Ambedkar College, Ambadave, Ratnagiri and Principal, V.V. Dalvie College, Talere, Sindhudurg.

**AC./6.33 (N)/25/3/2026**

Copy forwarded with Compliments for information to:-

- 1) The Chairman, Board of Deans,
- 2) The Dean, Faculty of Science,
- 3) The Chairman, **Ad-hoc Board of Studies in Board of Studies in Computer Science**
- 4) The Director, Board of Examinations and Evaluation,
- 5) The Director, Department of Students Development,
- 6) The Director, Department of Information & Communication Technology,
- 7) The Director, Centre for Distance and Online Education (CDOE)Vidyanagari,
- 8) The Deputy Registrar, Admission, Enrolment, Eligibility & Migration Department (AEM),

<b>Copy forwarded for information and necessary action to :-</b>	
1	The Deputy Registrar, (Admissions, Enrolment, Eligibility and Migration Dept)(AEM), <a href="mailto:dr@eligi.mu.ac.in">dr@eligi.mu.ac.in</a>
2	The Deputy Registrar, Result unit, Vidyanagari <a href="mailto:drresults@exam.mu.ac.in">drresults@exam.mu.ac.in</a>
3	The Deputy Registrar, Marks and Certificate Unit,. Vidyanagari <a href="mailto:dr.verification@mu.ac.in">dr.verification@mu.ac.in</a>
4	The Deputy Registrar, Appointment Unit, Vidyanagari <a href="mailto:dr.appointment@exam.mu.ac.in">dr.appointment@exam.mu.ac.in</a>
5	The Deputy Registrar, CAP Unit, Vidyanagari <a href="mailto:cap.exam@mu.ac.in">cap.exam@mu.ac.in</a>
6	The Deputy Registrar, College Affiliations & Development Department (CAD), <a href="mailto:deputyregistrar.uni@gmail.com">deputyregistrar.uni@gmail.com</a>
7	The Deputy Registrar, PRO, Fort, (Publication Section), <a href="mailto:Pro@mu.ac.in">Pro@mu.ac.in</a>
8	The Deputy Registrar, Executive Authorities Section (EA) <a href="mailto:eau120@fort.mu.ac.in">eau120@fort.mu.ac.in</a> He is requested to treat this as action taken report on the concerned resolution adopted by the Academic Council referred to the above circular.
9	The Deputy Registrar, Research Administration & Promotion Cell (RAPC), <a href="mailto:rapc@mu.ac.in">rapc@mu.ac.in</a>
10	The Deputy Registrar, Academic Appointments & Quality Assurance (AAQA) dy.registrar.tau.fort.mu.ac.in <a href="mailto:ar.tau@fort.mu.ac.in">ar.tau@fort.mu.ac.in</a>
11	The Deputy Registrar, College Teachers Approval Unit (CTA), <a href="mailto:concolsection@gmail.com">concolsection@gmail.com</a>
12	The Deputy Registrars, Finance & Accounts Section, fort <a href="mailto:draccounts@fort.mu.ac.in">draccounts@fort.mu.ac.in</a>
13	The Deputy Registrar, Election Section, Fort <a href="mailto:drelection@election.mu.ac.in">drelection@election.mu.ac.in</a>
14	The Assistant Registrar, Administrative Sub-Campus Thane, <a href="mailto:thanesubcampus@mu.ac.in">thanesubcampus@mu.ac.in</a>
15	The Assistant Registrar, School of Engg. & Applied Sciences, Kalyan, <a href="mailto:ar.seask@mu.ac.in">ar.seask@mu.ac.in</a>
16	The Assistant Registrar, Ratnagiri Sub-centre, Ratnagiri, <a href="mailto:ratnagirisubcentar@gmail.com">ratnagirisubcentar@gmail.com</a>
17	The Director, Centre for Distance and Online Education (CDOE), Vidyanagari, <a href="mailto:director@idol.mu.ac.in">director@idol.mu.ac.in</a>
18	Director, Innovation, Incubation and Linkages, Dr. Sachin Laddha <a href="mailto:pinkumanno@gmail.com">pinkumanno@gmail.com</a>
19	Director, Department of Lifelong Learning and Extension (DLLE), <a href="mailto:dlleuniversityofmumbai@gmail.com">dlleuniversityofmumbai@gmail.com</a>

<b>Copy for information :-</b>	
1	P.A to Hon'ble Vice-Chancellor, <a href="mailto:vice-chancellor@mu.ac.in">vice-chancellor@mu.ac.in</a>
2	P.A to Pro-Vice-Chancellor <a href="mailto:pvc@fort.mu.ac.in">pvc@fort.mu.ac.in</a>
3	P.A to Registrar, <a href="mailto:registrar@fort.mu.ac.in">registrar@fort.mu.ac.in</a>
4	P.A to all Deans of all Faculties
5	P.A to Finance & Account Officers, (F & A.O), <a href="mailto:camu@accounts.mu.ac.in">camu@accounts.mu.ac.in</a>

**To,**

1	The Chairman, Board of Deans <a href="mailto:pvc@fort.mu.ac.in">pvc@fort.mu.ac.in</a>
2	<p><b>Faculty of Humanities,</b></p> <p><b>Offg. Dean</b></p> <p>1. Prof.Anil Singh <a href="mailto:Dranilsingh129@gmail.com">Dranilsingh129@gmail.com</a></p> <p><b>Offg. Associate Dean</b></p> <p>2. Prof.Manisha Karne <a href="mailto:mkarne@economics.mu.ac.in">mkarne@economics.mu.ac.in</a></p> <p>3. Dr.Suchitra Naik <a href="mailto:Naiksuchitra27@gmail.com">Naiksuchitra27@gmail.com</a></p> <p><b>Faculty of Commerce &amp; Management,</b></p> <p><b>Offg. Dean,</b></p> <p>1 Prin.Ravindra Bambardekar <a href="mailto:principal@model-college.edu.in">principal@model-college.edu.in</a></p> <p><b>Offg. Associate Dean</b></p> <p>2. Dr.Kavita Laghate <a href="mailto:kavitalaghate@jbims.mu.ac.in">kavitalaghate@jbims.mu.ac.in</a></p> <p>3. Prin.Kishori Bhagat <a href="mailto:kishoribhagat@rediffmail.com">kishoribhagat@rediffmail.com</a></p>

	<p><b>Faculty of Science &amp; Technology</b></p> <p><b>Offg. Dean</b>  1. Prof. Shivram Garje  <a href="mailto:ssgarje@chem.mu.ac.in">ssgarje@chem.mu.ac.in</a></p> <p><b>Offg. Associate Dean</b>  2. Dr. Madhav R. Rajwade  <a href="mailto:Madhavr64@gmail.com">Madhavr64@gmail.com</a>  3. Prin. Deven Shah  <a href="mailto:sir.deven@gmail.com">sir.deven@gmail.com</a></p>
	<p><b>Faculty of Inter-Disciplinary Studies,</b></p> <p><b>Offg. Dean</b>  1. Dr. Anil K. Singh  <a href="mailto:aksingh@trcl.org.in">aksingh@trcl.org.in</a></p> <p><b>Offg. Associate Dean</b>  2. Prin. Chadrashekhhar Ashok Chakradeo  <a href="mailto:cachakradeo@gmail.com">cachakradeo@gmail.com</a>  3. Dr. Kunal Ingle  <a href="mailto:drkunalingle@gmail.com">drkunalingle@gmail.com</a></p>
3	Chairman, Board of Studies,
4	The Director, Board of Examinations and Evaluation, <a href="mailto:dboee@exam.mu.ac.in">dboee@exam.mu.ac.in</a>
5	The Director, Board of Students Development, <a href="mailto:dsd@mu.ac.in">dsd@mu.ac.in</a> <b>DSW</b> <a href="mailto:direcotr@dsw.mu.ac.in">direcotr@dsw.mu.ac.in</a>
6	The Director, Department of Information & Communication Technology, <a href="mailto:director.dict@mu.ac.in">director.dict@mu.ac.in</a>

# As Per NEP 2020

## University of Mumbai



**Syllabus for Major  
Vertical – 1, 4 & 6  
(Scheme – I)**

**Faulty of Science & Technology**

**Board of Studies in Computer Science**

**Name of the Programme – B.Sc. (Computer Science)**

**U.G. Third Year Programme**

**Exit Degree**

**B.Sc. (Computer Science)**

**Semester**

**V & VI**

**From the Academic Year**

**2026-27**

# University of Mumbai



(As per NEP 2020)

Sr. No.	Heading	Particulars
1	Title of program O: _____	B.Sc. (Computer Science)
2	Exit Degree	B.Sc. (Computer Science)
3	Scheme of Examination R: _____	NEP 40% Internal 60% External, Semester End Examination Individual Passing in Internal and External Examination
4	Standards of Passing R: _____	40% in each component
5	Credit Structure R: SU – 555 E (I) R: SU – 555 F (I)	Attached herewith
6	Semesters	Sem V & VI
7	Program Academic Level	5.5
8	Pattern	Semester
9	Status	New
10	To be implemented from Academic Year	From Academic Year: 2026-27

Sd/-

Sign of the BOS Chairman  
Dr. Jyotshna Dongardive  
Ad-hoc BOS (Computer Science)

Sd/-

Sign of the Offg. Associate Dean  
Dr. Madhav R. Rajwade  
Faculty of Science & Technology

Sd/-

Sign of Offg. Dean  
Prof. Shivram S. Garje  
Faculty of Science & Technology

# Preamble

## 1) Introduction

The Third Year of the B.Sc. (Computer Science) programme represents the culmination of undergraduate study at Academic Level 5.5 under the NEP 2020 framework. At this stage, the curriculum transitions from foundational and intermediate concepts to advanced, applied, and research-oriented domains of Computer Science. The focus is on developing depth in emerging areas such as Artificial Intelligence, Cyber & Information Security, Data Science, Cloud Computing, and Software Project Management, along with elective specializations in Information Retrieval and Linux Server Administration.

The programme is structured to integrate conceptual understanding with hands-on implementation through advanced practical courses and two structured mini-projects. Mini Project – I emphasize solution development for real-world problems using appropriate design methodologies and documentation standards, while Mini Project – II focuses on research orientation, experimentation, analysis, and scholarly reporting in established academic format. This dual approach ensures that learners are prepared for both industry deployment and academic progression.

In alignment with the vision of the University of Mumbai and the National Education Policy 2020, the Third Year curriculum promotes analytical rigor, innovation, ethical responsibility, and professional competence. It prepares graduates not only to meet contemporary industry demands but also to pursue higher studies, research, entrepreneurship, and lifelong learning in rapidly evolving technological ecosystems.

## 2) Aims and Objectives

**Advanced Conceptual Understanding:** To provide in-depth knowledge of advanced computing domains including Artificial Intelligence, Information Security, Data Science, and Cloud Computing, thereby strengthening the theoretical and applied foundations of Computer Science.

**Research and Analytical Competence:**

To cultivate research aptitude through literature review, methodology design, experimentation, data analysis, and structured research reporting.

**Industry-Oriented Skill Development:**

To enhance practical and professional skills through hands-on laboratory work, real-world project development, and exposure to current industry tools and technologies.

**Project Planning and Management Skills:**

To develop competencies in software project management, including planning, scheduling, risk management, quality assurance, and agile methodologies.

**Ethical and Responsible Computing:**

To instill ethical awareness in areas such as cybersecurity, responsible AI, data privacy, and professional conduct in computing practices.

### **Preparation for Higher Studies and Career Progression:**

To equip learners with the intellectual maturity, technical expertise, and research orientation required for postgraduate studies, research careers, and professional roles in the IT industry.

### **3) Learning Outcomes**

**By the end of the third year (T.Y.B.Sc.), students will be able to:**

- Demonstrate advanced knowledge in Artificial Intelligence, Data Science, Cloud Computing, Cyber Security, and related emerging domains.
- Design, develop, and deploy complete software solutions addressing real-world problems using appropriate architectures, tools, and technologies.
- Apply research methodology principles including literature review, problem formulation, hypothesis validation, experimentation, and result interpretation.
- Analyze datasets using statistical and machine learning techniques and interpret results using appropriate performance metrics.
- Implement secure computing practices by applying cryptographic techniques, authentication mechanisms, network security models, and system hardening principles.
- Plan, manage, and monitor software projects using structured methodologies including traditional and agile frameworks.
- Evaluate alternative technological solutions and select appropriate tools and platforms based on performance, scalability, security, and cost considerations.
- Work effectively as individuals and as team members in multidisciplinary environments, demonstrating leadership and collaborative skills.
- Prepare and present technical documentation, research papers, and project reports in standardized professional formats.
- Demonstrate ethical responsibility, professional integrity, and awareness of legal and societal implications of computing technologies.
- Pursue higher education, research, entrepreneurship, or professional employment with confidence and competence in advanced computing domains.

#### 4) Credit Structure of the Program

##### B.Sc. (Computer Science)

R: SU – 555 E (I)										
Level	Semester	Major		Minor	OE	VSC, SEC (VSEC)	AEC, VEC, IKS	OJT, FP, CEP, CC, RP	Cum. Cr. / Sem.	Degree/ Cum. Cr.
		Mandatory	Electives							
5.5	V	MJ11: Artificial Intelligence (TH) – 2	MJEL1: Software Testing & Quality Assurance (TH) – 2	4	-	VSC: 2  Ethical Hacking – 2	-	FP/CEP:2	22	UG Degree 132
		MJ12: Cyber & Information Security (TH) – 2	OR MJEL2: Wireless & Sensor Networks (TH) – 2							
		MJP5: Computer Science Practical 5 (PR) – 2	MJELP1: Software Testing & Quality Assurance Practical (PR) – 2							
MJP6: Mini Project – I (PR) – 2	OR MJELP2: Wireless & Sensor Networks Practical (PR) – 2									
MJ13: IKS in Computational Systems (TH) – 2										
10	4									
R: SU – 555 F (I)										
5.5	VI	MJ14: Data Science (TH) – 2	MJEL3: Information Retrieval (TH) – 2	4	-	-	-	OJT:4	22	
		MJ15: Cloud Computing (TH) – 2	OR MJEL4: Linux Server Administration (TH) – 2							
		MJ16: Software Project Management (TH) – 2	MJELP3: Information Retrieval Practical (PR) – 2							
MJP7: Computer Science Practical 6 (PR) – 2	OR MJELP4: Linux Server Administration Practical (PR) – 2									
MJP8: Mini Project – II (PR) – 2										
10	4									
Cum Cr.		48	8	18	12	8+6	8+4+2	8+6+4	132	
<b>Exit option: Award of UG Degree in Major with 132 credits OR Continue with Major and Minor</b>										

[Abbreviation - OE – Open Electives, VSC – Vocation Skill Course, SEC – Skill Enhancement Course, (VSEC), AEC – Ability Enhancement Course, VEC – Value Education Course, IKS – Indian Knowledge System, OJT – on Job Training, FP – Field Project, CEP – Continuing Education Program, CC – Co-Curricular, RP – Research Project ]

**Under Graduate Diploma in Computer Science**

## Program Structure for Sem. V & VI (NEP 2020)

Vertical No	Paper Title		Credits
<b>Sem. V</b>			
<b>1.</b>	<b>Mandatory</b>	Artificial Intelligence	2
		Cyber & Information Security	2
		Computer Science Practical 5	2
		Mini Project – I	2
		IKS in Computational Systems	2
	<b>Electives</b> <i>(any one)</i>	Software Testing & Quality Assurance	2
		Wireless & Sensor Networks	2
		Software Testing & Quality Assurance Practical	2
		Wireless & Sensor Networks Practical	2
<b>2.</b>	<b>Minor</b> <b>(To be selected from other discipline)</b>		Ethical Hacking
<b>4.</b>	<b>VSC (Practical)</b>	Ethical Hacking	2
<b>6.</b>	<b>CEP</b> <b>(To be selected from CEP topic list)</b>		2
<b>Credits</b>			<b>22</b>
<b>Sem. VI</b>			
<b>1.</b>	<b>Mandatory</b>	Data Science	2
		Cloud Computing	2
		Software Project Management	2
		Computer Science Practical 6	2
		Mini Project – II	2
	<b>Electives</b> <i>(any one)</i>	Information Retrieval	2
		Linux Server Administration	2
		Information Retrieval Practical	2
		Linux Server Administration Practical	2

2.	<b>Minor</b> (To be selected from other discipline)		
6.	VI	<b>OJT</b>	<b>4</b>
			<b>Credits</b> <b>22</b>
			<b>Total Credits</b> <b>44</b>

**Sem. – V**

**Sem. - V**  
**Vertical – 1**  
**Major**  
**Mandatory**  
**(2+2+2+2+2)**

# Syllabus

## B.Sc. (Computer Science)

### (Sem.- V)

**Name of the Course: Artificial Intelligence**

Sr. No.	Heading	Particulars
1	<b>Description the course:</b>	<p><b>Introduction:</b></p> <p>The Artificial Intelligence course introduces students to the foundational principles that enable machines to simulate intelligent behavior. It covers intelligent agents, search techniques, logical reasoning, probabilistic modeling, machine learning paradigms, reinforcement learning, and responsible AI. The course builds a structured understanding of how intelligent systems are designed, analyzed, and deployed.</p> <p><b>Relevance:</b></p> <p>This course is highly relevant in today's data-driven world where automation and intelligent decision-making are integral to almost every sector. Understanding AI bridges the gap between theoretical computer science concepts and real-world intelligent applications such as recommendation systems, predictive analytics, and autonomous systems.</p> <p><b>Usefulness:</b></p> <p>The course equips students with practical and analytical skills required to design AI-based solutions. It enhances problem-solving ability through search algorithms, logical inference, probabilistic reasoning, and machine learning techniques. These competencies are essential for developers, analysts, researchers, and technology professionals working with intelligent systems.</p> <p><b>Application:</b></p> <p>Artificial Intelligence principles are applied in domains such as healthcare diagnostics, financial forecasting, robotics, natural language processing, cybersecurity threat detection, recommendation engines, and generative AI systems. From chatbots and fraud detection systems to autonomous vehicles and smart assistants, AI forms the backbone of modern intelligent applications.</p> <p><b>Interest:</b></p> <p>The subject stimulates curiosity by demonstrating how</p>

		<p>machines can learn, reason, and make decisions. Concepts like A* search, game-playing algorithms (Minimax), Bayesian reasoning, neural networks, and reinforcement learning make the course intellectually challenging and closely connected to emerging technologies such as generative AI and deep learning.</p> <p><b>Connection with Other Courses:</b></p> <p>AI concepts are closely linked with Data Structures and Algorithms (search trees and graphs), Discrete Mathematics (logic and probability), Data Science (model building and evaluation), Software Engineering (AI system lifecycle), Cybersecurity (AI-driven threat detection), and Cloud Computing (AI deployment at scale). The course strengthens interdisciplinary understanding within the computer science curriculum.</p> <p><b>Demand in the Industry:</b></p> <p>The demand for AI and ML skills is rapidly increasing across IT services, fintech, healthcare, retail, manufacturing, and media industries. Organizations require professionals who understand intelligent systems, predictive modeling, decision automation, and ethical AI deployment, making this course strategically important for industry readiness.</p> <p><b>Job Prospects:</b></p> <p>Students with AI knowledge can pursue roles such as AI/ML intern, junior data analyst, AI application developer, or software engineer integrating AI solutions. With advanced study and project experience, they can progress to positions such as machine learning engineer, data scientist, AI research associate, or AI product specialist.</p>
2	<b>Vertical:</b>	Major (Mandatory)
3	<b>Type:</b>	Theory
4	<b>Credits:</b>	2 credits
5	<b>Hours Allotted:</b>	30 Hours
6	<b>Marks Allotted:</b>	50 Marks
7	<p><b>Course Objectives (CO):</b></p> <p><b>CO 1.</b> To introduce the foundational concepts of Artificial Intelligence and intelligent agent design.</p> <p><b>CO 2.</b> To develop problem-solving skills using search strategies and logical reasoning techniques.</p>	

	<p><b>CO 3.</b> To provide understanding of probabilistic reasoning and machine learning methodologies.</p> <p><b>CO 4.</b> To familiarize students with reinforcement learning and latent variable models.</p> <p><b>CO 5.</b> To cultivate awareness of ethical considerations and responsible AI practices.</p>
8	<p><b>Course Outcomes (OC):</b></p> <p>After successful completion of this course, students would be able to -</p> <p><b>OC 1.</b> Explain the principles of intelligent agents, search strategies, and logical reasoning frameworks.</p> <p><b>OC 2.</b> Apply informed and uninformed search algorithms and adversarial reasoning techniques to solve AI problems.</p> <p><b>OC 3.</b> Implement and evaluate basic machine learning models for classification, regression, and clustering tasks.</p> <p><b>OC 4.</b> Analyze probabilistic models including Bayesian networks and hidden variable models for uncertain scenarios.</p> <p><b>OC 5.</b> Assess AI systems from a responsible AI perspective, identifying bias, fairness, transparency, and generative AI risks.</p>
9	<p><b>Modules:</b></p> <p><b>Module 1 (15 hours):</b></p> <p><b>Foundations of AI &amp; Intelligent Agents:</b> What is AI? Rational agents vs human thinking, Computational agents, Agent–Environment interaction, Types of environments, Agent architectures (simple reflex, model-based, goal-based, utility-based, learning agents)</p> <p><b>Problem Solving &amp; Search:</b> Problem formulation, Uninformed search: BFS, DFS, Uniform Cost, IDS, Informed search: Greedy, A*, Heuristics: admissibility &amp; consistency, Adversarial search: Minimax &amp; Alpha-Beta pruning</p> <p><b>Knowledge Representation &amp; Logical Reasoning:</b> Knowledge-based agents, Propositional logic &amp; inference, First-Order Logic, Rule-based systems, Planning basics (STRIPS concept), Fuzzy Logic &amp; Fuzzification</p> <p><b>Reasoning Under Uncertainty:</b> Probabilistic reasoning, Bayes theorem, Conditional independence, Bayesian Networks</p> <p><b>Module 2 (15 hours):</b></p> <p><b>Introduction of Machine Learning:</b> Forms of learning (supervised, unsupervised, reinforcement), Parametric vs Nonparametric models, Bias–variance tradeoff, Overfitting &amp; regularization, Gradient descent (intuitive)</p> <p><b>Supervised Learning Models:</b> Classification vs Regression, k-NN, Decision Trees, Naive Bayes, SVM, Artificial Neural Networks:(single-layer &amp; concept of deep learning, Ensemble methods &amp; Boosting</p> <p><b>Probabilistic &amp; Latent Variable Models:</b> Statistical learning framework, Maximum Likelihood Estimation, Learning with complete data, Hidden variables, EM Algorithm, Hidden Markov Models</p>

	<p><b>Unsupervised &amp; Reinforcement Learning:</b> Concept of clustering, Association rule mining (Apriori concept), Reinforcement learning framework, Markov Decision Processes, Q-Learning (update rule intuition)</p> <p><b>Responsible AI:</b> Ethical issues in AI systems, Bias and fairness in AI models, Transparency and explainability, Accountability and human oversight, Risks in Generative AI (hallucination, deepfakes, misuse)</p>	
<b>10</b>	<p><b>Text Books</b></p> <p>1. Artificial Intelligence: A Modern Approach, Stuart Russell and Peter Norvig, 3rd Edition, Pearson, 2010.</p>	
<b>11</b>	<p><b>Reference Books</b></p> <p>1. Artificial Intelligence: Foundations of Computational Agents, David L Poole, Alan K. Mackworth, 2nd Edition, Cambridge University Press, 2017.</p> <p>2. Artificial Intelligence, Kevin Knight and Elaine Rich, 3rd Edition, 2017</p> <p>3) The Elements of Statistical Learning, Trevor Hastie, Robert Tibshirani and Jerome Friedman, Springer, 2013</p>	
<b>12</b>	<b>Internal Continuous Assessment: 40%</b>	<b>Semester End Examination: 60%</b>

## Name of the Course: Cyber and Information Security

Sr. No.	Heading	Particulars
1	Description the course:	<p><b>Introduction:</b></p> <p>The Cyber &amp; Information Security course introduces students to the principles and mechanisms used to protect digital information, communication systems, and computing infrastructure. It covers foundational security concepts, classical and modern cryptography, authentication mechanisms, network security protocols, intrusion detection, malware threats, and firewall architectures.</p> <p><b>Relevance:</b></p> <p>In an era of digital transformation, cybersecurity is critical for safeguarding data, privacy, and national infrastructure. This course is highly relevant as organizations increasingly depend on secure communication, encrypted transactions, identity management systems, and network defense mechanisms to prevent cyber threats.</p> <p><b>Usefulness:</b></p> <p>The course equips students with conceptual and practical understanding of encryption algorithms, authentication protocols, digital signatures, secure communication standards, and network protection mechanisms. This knowledge is essential for identifying vulnerabilities, implementing security controls, and designing secure systems in real-world environments.</p> <p><b>Application:</b></p> <p>Cyber and information security principles are applied in banking systems, e-commerce platforms, cloud infrastructure, government services, defense networks, and enterprise IT systems. Technologies such as AES, RSA, SSL/TLS, IPsec, Kerberos, digital signatures, intrusion detection systems, and firewalls are integral to securing modern digital ecosystems.</p> <p><b>Interest:</b></p> <p>The subject generates strong interest as it deals with real-world cyber-attacks, encryption techniques, malware analysis, and network defense strategies. Topics such as ethical hacking concepts, digital forensics basics, cryptographic algorithms, and DDoS countermeasures make the course both technically engaging and socially</p>

		<p>significant.</p> <p><b>Connection with Other Courses:</b></p> <p>This course is closely connected with Computer Networks (protocols and OSI model), Operating Systems (system security mechanisms), Discrete Mathematics (number theory and cryptographic foundations), Cloud Computing (secure infrastructure), and Software Engineering (secure coding practices). It strengthens interdisciplinary knowledge required for building secure computing systems.</p> <p><b>Demand in the Industry:</b></p> <p>There is a rapidly growing demand for cybersecurity professionals across IT services, fintech, healthcare, telecom, defense, and cloud service providers. Organizations require experts in encryption, authentication, network security, intrusion detection, and firewall management to mitigate evolving cyber threats and comply with regulatory standards.</p> <p><b>Job Prospects:</b></p> <p>Students can pursue roles such as cybersecurity analyst, network security engineer, information security officer, SOC analyst, penetration testing trainee, and security consultant. Advanced expertise can lead to careers in cryptography research, cloud security architecture, digital forensics, and security auditing.</p>
<b>2</b>	<b>Vertical:</b>	Major (Mandatory)
<b>3</b>	<b>Type:</b>	Theory
<b>4</b>	<b>Credits:</b>	2 credits
<b>5</b>	<b>Hours Allotted:</b>	30 Hours
<b>6</b>	<b>Marks Allotted:</b>	50 Marks
<b>7</b>	<p><b>Course Objectives (CO):</b></p> <p><b>CO 1.</b> To introduce foundational concepts of cybersecurity, including security services, attacks, and mechanisms.</p> <p><b>CO 2.</b> To develop understanding of symmetric and asymmetric cryptographic techniques and key management.</p> <p><b>CO 3.</b> To provide knowledge of authentication methods, digital signatures, and secure communication protocols.</p> <p><b>CO 4.</b> To familiarize students with network security technologies, intrusion detection, and malware countermeasures.</p> <p><b>CO 5.</b> To enable understanding of firewall design principles and security architecture implementation.</p>	

8	<p><b>Course Outcomes (OC):</b></p> <p>After successful completion of this course, students would be able to -</p> <p><b>OC 1.</b> Explain security architecture, attack models, and fundamental cryptographic principles.</p> <p><b>OC 2.</b> Apply symmetric and public-key cryptographic techniques (AES, RSA, Diffie-Hellman) for secure communication.</p> <p><b>OC 3.</b> Analyse and implement authentication mechanisms including digital signatures, hash functions, MACs, and PKI-based systems.</p> <p><b>OC 4.</b> Evaluate network and web security protocols such as SSL/TLS, IPsec, Kerberos, and email security mechanisms.</p> <p><b>OC 5.</b> Identify and assess intrusion techniques, malware threats, DDoS attacks, and firewall configurations for effective defence.</p>
9	<p><b>Modules:</b></p> <p><b>Module 1 (15 hours):</b></p> <p><b>Introduction:</b> Security Trends, The OSI Security Architecture, Security Attacks, Security Services, Security Mechanisms</p> <p><b>Classical Encryption Techniques:</b> Symmetric Cipher Model, Substitution Techniques, Transposition Techniques, Steganography, Block Cipher Principles, The Data Encryption Standard, The Strength of DES, AES (round details not expected), Multiple Encryption and Triple DES, Block Cipher Modes of Operation, Stream Ciphers</p> <p><b>Public-Key Cryptography and RSA:</b> Principles of Public-Key Cryptosystems, The RSA Algorithm</p> <p><b>Key Management:</b> Public-Key Cryptosystems, Key Management, Diffie-Hellman Key Exchange</p> <p><b>Message Authentication and Hash Functions:</b> Authentication Requirements, Authentication Functions, Message Authentication Codes, Hash Functions, Security of Hash Functions and Macs, Secure Hash Algorithm, HMAC</p> <hr/> <p><b>Module 2 (15 hours):</b></p> <p><b>Digital Signatures and Authentication:</b> Digital Signatures, Authentication Protocols, Digital Signature Standard</p> <p><b>Authentication Applications:</b> Kerberos, X.509 Authentication, Public-Key Infrastructure</p> <p><b>Electronic Mail Security:</b> Pretty Good Privacy, S/MIME</p> <p><b>IP Security:</b> Overview, Architecture, Authentication Header, Encapsulating Security Payload, Combining Security Associations, Key Management</p> <p><b>Web Security:</b> Web Security Considerations, Secure Socket Layer and Transport Layer Security, Secure Electronic Transaction</p> <p><b>Intrusion:</b> Intruders, Intrusion Techniques, Intrusion Detection</p> <p><b>Malicious Software:</b> Viruses and Related Threats, Virus Countermeasures, DDOS</p>

	<b>Firewalls:</b> Firewall Design Principles, Types of Firewalls	
<b>10</b>	<b>Text Books</b> 1. Cryptography and Network Security: Principles and Practice 7th edition, William Stallings, Pearson	
<b>11</b>	<b>Reference Books</b> 1. Cryptography and Network Security, 2nd edition, Behrouz A Fourouzan, Debdeep Mukhopadhyay, TMH. 2. Atul Kahate, "Cryptography and Network Security", Tata McGraw-Hill.	
<b>12</b>	<b>Internal Continuous Assessment: 40%</b>	<b>Semester End Examination: 60%</b>

**IKS**

**(2)**

## Name of the Course: IKS in Computational Systems

Sr. No.	Heading	Particulars
1	<b>Description the course:</b>	<p><b>Introduction:</b></p> <p>The IKS in Computational Systems course explores classical Indian Knowledge Systems (IKS) through the lens of modern computational thinking. It examines how śāstra-based formalization, Pāṇini's generative grammar, Piṅgala's combinatorics, Nyāya logic, Ayurvedic classification, and Arthaśāstra cryptography embody structured, rule-based, and algorithmic principles analogous to contemporary Computer Science concepts.</p> <p><b>Relevance:</b></p> <p>This course is highly relevant in the context of interdisciplinary education under NEP 2020, linking traditional knowledge frameworks with modern computational theory. It demonstrates that formal systems, symbolic encoding, logical inference, and structured reasoning have deep historical roots aligned with current developments in AI, formal grammars, combinatorics, and cybersecurity.</p> <p><b>Usefulness:</b></p> <p>The course develops conceptual clarity in knowledge representation, rule-based reasoning, generative systems, symbolic abstraction, and inference validation. It strengthens analytical thinking by mapping classical structures such as pramāṇa theory, praśtāra enumeration, and five-member syllogism to modern logic systems, parsing algorithms, and explainable AI.</p> <p><b>Application:</b></p> <p>The computational parallels of śāstra methodology, Pāṇinian grammar, and Nyāya logic apply to formal language theory, natural language processing, expert systems, decision trees, inference engines, and secure communication models. Students gain perspective on how structured reasoning models can inform AI system design, feature engineering, classification systems, and encryption foundations.</p> <p><b>Interest:</b></p> <p>The course is intellectually stimulating as it reveals algorithmic and logical sophistication embedded in ancient Indian texts. Discovering binary encoding in</p>

		<p>Piṅgala, generative grammar in Pāṇini, inference structures in Nyāya, and coded communication in Arthaśāstra creates a unique synthesis of philosophy and computer science.</p> <p><b>Connection with Other Courses:</b></p> <p>This course connects strongly with Artificial Intelligence (knowledge representation and inference), Formal Languages and Automata Theory (CFG and rewrite systems), Data Science (classification and feature mapping), Cyber Security (encryption principles), and Information Retrieval/NLP (grammar and parsing foundations). It enriches computational subjects through historical and conceptual depth.</p> <p><b>Demand in the Industry:</b></p> <p>Modern fields such as Artificial Intelligence, Natural Language Processing, and cybersecurity rely heavily on rule-based systems, logical inference, and structured knowledge representation. The course highlights the foundational principles behind these systems through classical knowledge frameworks.</p> <p><b>Job Prospects:</b></p> <p>The concepts covered support careers in AI, data science, computational linguistics, knowledge engineering, and cybersecurity. The analytical and interdisciplinary perspective gained also benefits students pursuing research and advanced studies in computer science.</p>
2	<b>Vertical:</b>	Major (Mandatory)
3	<b>Type:</b>	Theory
4	<b>Credits:</b>	2 credits
5	<b>Hours Allotted:</b>	30 Hours
6	<b>Marks Allotted:</b>	50 Marks
7	<p><b>Course Objectives (CO):</b></p> <p><b>CO 1.</b> To introduce the concept of śāstra as a structured and formal knowledge system aligned with computational thinking.</p> <p><b>CO 2.</b> To analyze algorithmic and combinatorial principles in Piṅgala's Chandaḥśāstra and their relation to binary systems and recursion.</p> <p><b>CO 3.</b> To examine Pāṇini's grammatical system as a generative formal grammar and relate it to modern language theory.</p> <p><b>CO 4.</b> To interpret Nyāya logic and Ayurvedic classification systems as structured inference and expert systems.</p>	

	<b>CO 5.</b> To explore cryptographic principles in Arthaśāstra and connect them with modern secure communication models.
<b>8</b>	<p><b>Course Outcomes (OC):</b></p> <p>After successful completion of this course, students would be able to -</p> <p><b>OC 1.</b> Explain and interpret classical Indian knowledge frameworks as structured computational models.</p> <p><b>OC 2.</b> Analyze Piṅgala’s combinatorial methods and relate them to binary encoding, recursion, and tree structures.</p> <p><b>OC 3.</b> Model and compare Pāṇini’s generative grammar with modern formal grammars and parsing systems.</p> <p><b>OC 4.</b> Apply Nyāya logical structures and Ayurvedic classification principles to modern inference engines and expert systems.</p> <p><b>OC 5.</b> Relate ancient cryptographic techniques to contemporary symmetric encryption and cybersecurity foundations.</p>
<b>9</b>	<p><b>Modules:</b></p> <p><b>Module 1 (15 hours):</b></p> <p><b>Śāstra Methodology and Knowledge Formalization</b>  Indian knowledge traditions systematized disciplines as <i>śāstra</i> — structured bodies of rule-governed knowledge.  Topics:</p> <ul style="list-style-type: none"> <li>• Concept of śāstra as a formal knowledge system</li> <li>• Sutra method as compressed symbolic encoding</li> <li>• Pramāṇa theory (Pratyakṣa, Anumāna, Āgama)</li> <li>• Knowledge validation and structured reasoning</li> </ul> <p>As Modern CS Concept: Knowledge representation, formal specification, symbolic abstraction.</p> <p><b>Piṅgala’s Chandaḥśāstra as Binary Encoding and Combinatorial Generation</b>  Study of prosodic structures in Piṅgala’s text and its algorithmic features.  Topics:</p> <ul style="list-style-type: none"> <li>• Laghu–Guru representation as binary encoding</li> <li>• Prastāra (systematic enumeration of patterns)</li> <li>• Naṣṭa and Uddiṣṭa (index retrieval mechanisms)</li> <li>• Meru-Prastāra and combinatorial expansion</li> </ul> <p>As Modern CS Concept: Binary number systems, Recursive enumeration, Tree structures, Pascal triangle and combinatorics, Algorithmic generation</p> <p><b>Pāṇini’s Aṣṭādhyāyī as a Generative Formal Grammar</b>  Examination of the structural architecture of Pāṇini’s grammatical system.  Topics:</p> <ul style="list-style-type: none"> <li>• Rule-based generative structure</li> <li>• Meta-rules and rule precedence</li> <li>• Context-sensitive operations</li> </ul>

	<ul style="list-style-type: none"> <li>• Conflict resolution mechanisms</li> </ul> <p>As Modern CS Concept: Context-Free Grammar (CFG), Rewrite systems, Automata theory, foundations, Parsing algorithms, Foundations of NLP</p> <p><b>Module 2 (15 hours):</b></p> <p><b>Nyāya Logic as Structured Inference and Knowledge Systems</b>  Study of classical Indian logical structures.  Topics:</p> <ul style="list-style-type: none"> <li>• Five-member syllogism</li> <li>• Anumāna (inference) structure</li> <li>• Debate methodology and validation</li> <li>• Padārtha (categorical ontology overview)</li> </ul> <p>As Modern CS Concept: Propositional logic, Predicate logic, Inference engines, Explainable AI</p> <p><b>Ayurvedic Classification as Rule-Based Expert System (5 Hours)</b>  Examination of structured diagnostic reasoning in Ayurvedic texts.  Topics:</p> <ul style="list-style-type: none"> <li>• Tridoṣa framework</li> <li>• Symptom–feature mapping</li> <li>• Multi-attribute classification</li> <li>• Decision principles in diagnosis</li> </ul> <p>As Modern CS Concept: Decision trees, Rule-based systems, Expert systems, Feature engineering, Multi-class classification</p> <p><b>Arthaśāstra Cryptography as Secure Communication Model</b>  Study of intelligence and secure communication practices.  Topics:</p> <ul style="list-style-type: none"> <li>• Secret communication techniques</li> <li>• Substitution and transposition systems</li> <li>• Concealment and coded messaging</li> <li>• Information protection mechanisms</li> </ul> <p>As Modern CS Concept: Symmetric encryption, Cipher algorithms, Basic steganography, Secure protocol abstraction, Foundations of cybersecurity</p>
<b>10</b>	<p><b>Text Books</b></p> <ol style="list-style-type: none"> <li>1. Computing Science in Ancient India, T.R.N. Rao/ Subhash Kak</li> <li>2. The Mathematics of India: Concepts, Methods, Connections P. P. Divakaran, Hindustan Book Agency, 2018</li> </ol>
<b>11</b>	<p><b>Reference Books</b></p> <ol style="list-style-type: none"> <li>1. Pāṇini: Background and introduction, George Cardona, Motilal Banarsidass, 1988</li> <li>2. Nyaya-Vaisesika (A History of Indian Literature) B. K. Matilal, Harrassowitz, 1977</li> <li>3. The Computation Meme: Explorations in Indic Computational Thinking K.</li> </ol>

	Gopinath & Shailaja D. Sharma, Indian Institute of Science (IISc), Bengaluru, 2022
	4. Indian Mathematics Engaging the World from Ancient to Modern Times G. G. Joseph, 2016
<b>12</b>	<b>Internal Continuous Assessment: 40%</b> <b>Semester End Examination: 60%</b>

## Name of the Course: Computer Science Practical 5

Sr. No.	Heading	Particulars
1	<b>Description</b> <b>the</b> <b>course:</b>	<p><b>Introduction:</b></p> <p>This practical course provides hands-on implementation experience in Artificial Intelligence and Cyber &amp; Information Security. It enables students to translate theoretical concepts into working models, algorithms, and configurations through structured laboratory exercises. The course emphasizes experimentation, performance evaluation, comparative analysis, and real-world system implementation.</p> <p><b>Relevance:</b></p> <p>The course is highly relevant as it integrates two critical domains—AI and Cyber Security—through applied learning. It strengthens students’ ability to implement algorithms such as search techniques, machine learning models, encryption methods, and network security mechanisms, thereby bridging the gap between conceptual understanding and practical competence.</p> <p><b>Usefulness:</b></p> <p>Students gain direct exposure to algorithm implementation, dataset handling, model evaluation, cryptographic programming, and security configuration. The practical approach enhances debugging skills, analytical comparison of algorithms, performance assessment, and secure system configuration, which are essential for industry-ready graduates.</p> <p><b>Application:</b></p> <p>The AI practicals support applications such as intelligent search systems, predictive modeling, classification tasks, neural networks, and data-driven decision systems. The cybersecurity practicals apply to secure communication, encryption systems, digital signatures, firewall configuration, intrusion detection, and secure web deployment—skills vital in enterprise and cloud environments.</p> <p><b>Interest:</b></p> <p>The course maintains high engagement through implementation-based learning, algorithm comparisons, model visualization, encryption experimentation, and network security setup. Working with tools like</p>

		<p>TensorFlow/OpenAI frameworks and configuring SSL/TLS or IDS systems makes the experience technically immersive and practically rewarding.</p> <p><b>Connection with Other Courses:</b></p> <p>This course complements Artificial Intelligence, Cyber &amp; Information Security, Data Structures, Computer Networks, Operating Systems, and Cloud Computing. It reinforces algorithmic thinking, system-level understanding, and applied cryptography, thereby strengthening interdisciplinary technical competence.</p> <p><b>Demand in the Industry:</b></p> <p>Industry increasingly demands professionals who can not only understand AI and security concepts but also implement, evaluate, and deploy them. Skills such as machine learning model building, encryption implementation, SSL/TLS configuration, IDS setup, and firewall management are highly valued across IT services, fintech, cybersecurity firms, and cloud platforms.</p> <p><b>Job Prospects:</b></p> <p>Students completing this practical course are better prepared for roles such as AI/ML intern, data analyst trainee, cybersecurity analyst, SOC analyst, security operations trainee, and junior DevOps engineer. The hands-on exposure also builds a strong foundation for advanced roles in machine learning engineering, security consulting, and system administration.</p>
<b>2</b>	<b>Vertical:</b>	Major (Mandatory)
<b>3</b>	<b>Type:</b>	Practical
<b>4</b>	<b>Credits:</b>	2 credits (1 credit = 30 Hours of Practical work in a semester)
<b>5</b>	<b>Hours Allotted:</b>	60 hours
<b>6</b>	<b>Marks Allotted:</b>	50 Marks
<b>7</b>	<p><b>Course Objectives (CO):</b></p> <p><b>CO 1.</b> To develop practical proficiency in implementing AI search algorithms and machine learning models.</p> <p><b>CO 2.</b> To enable students to train, evaluate, and compare supervised learning and ensemble models.</p> <p><b>CO 3.</b> To provide hands-on experience in implementing classical and modern cryptographic techniques.</p> <p><b>CO 4.</b> To familiarize students with secure communication protocols, intrusion detection, and firewall configuration.</p>	

	<b>CO 5.</b> To strengthen analytical and comparative skills through performance evaluation and system experimentation.
<b>8</b>	<p><b>Course Outcomes (OC):</b></p> <p>After successful completion of this course, students would be able to -</p> <p><b>OC 1.</b> Implement and compare AI search algorithms and supervised learning models, evaluating their performance using suitable metrics.</p> <p><b>OC 2.</b> Design and train machine learning models including decision trees, neural networks, SVM, ensemble methods, and association rule mining techniques.</p> <p><b>OC 3.</b> Develop and test cryptographic algorithms such as RSA, Diffie-Hellman, MACs, and digital signatures for secure communication.</p> <p><b>OC 4.</b> Configure and analyze network security mechanisms including SSL/TLS, IPsec, intrusion detection systems, and firewall rules.</p> <p><b>OC 5.</b> Evaluate the effectiveness, security implications, and performance characteristics of implemented AI and cybersecurity solutions.</p>
<b>9</b>	<p><b>Modules:</b></p> <p><b>Module 1 (30 hours):</b></p> <hr/> <p><b>Practical based on Artificial Intelligence</b></p> <hr/> <p><b>Breadth First Search &amp; Iterative Depth First Search</b></p> <ul style="list-style-type: none"> <li>• Implement the Breadth First Search algorithm to solve a given problem.</li> <li>• Implement the Iterative Depth First Search algorithm to solve the same problem.</li> <li>• Compare the performance and efficiency of both algorithms.</li> </ul> <p><b>A* Search and Recursive Best-First Search</b></p> <ul style="list-style-type: none"> <li>• Implement the A* Search algorithm for solving a pathfinding problem.</li> <li>• Implement the Recursive Best-First Search algorithm for the same problem.</li> <li>• Compare the performance and effectiveness of both algorithms.</li> </ul> <p><b>Decision Tree Learning</b></p> <ul style="list-style-type: none"> <li>• Implement the Decision Tree Learning algorithm to build a decision tree for a given dataset.</li> <li>• Evaluate the accuracy and effectiveness of the decision tree on test data.</li> <li>• Visualize and interpret the generated decision tree.</li> </ul> <p><b>Feed Forward Backpropagation Neural Network</b></p> <ul style="list-style-type: none"> <li>• Implement the Feed Forward Backpropagation algorithm to train a neural network.</li> <li>• Use a given dataset to train the neural network for a specific task.</li> <li>• Evaluate the performance of the trained network on test data.</li> </ul> <p><b>Support Vector Machines (SVM)</b></p> <ul style="list-style-type: none"> <li>• Implement the SVM algorithm for binary classification.</li> <li>• Train an SVM model using a given dataset and optimize its parameters.</li> <li>• Evaluate the performance of the SVM model on test data and analyze the results.</li> </ul> <p><b>Adaboost Ensemble Learning</b></p> <ul style="list-style-type: none"> <li>• Implement the Adaboost algorithm to create an ensemble of weak classifiers.</li> </ul>

- Train the ensemble model on a given dataset and evaluate its performance.
- Compare the results with individual weak classifiers.

**Naive Bayes' Classifier**

- Implement the Naive Bayes' algorithm for classification.
- Train a Naive Bayes' model using a given dataset and calculate class probabilities.
- Evaluate the accuracy of the model on test data and analyze the results.

**K-Nearest Neighbors (K-NN)**

- Implement the K-NN algorithm for classification or regression.
- Apply the K-NN algorithm to a given dataset and predict the class or value for test data.
- Evaluate the accuracy or error of the predictions and analyze the results.

**Association Rule Mining**

- Implement the Association Rule Mining algorithm (e.g., Apriori) to find frequent itemsets.
- Generate association rules from the frequent itemsets and calculate their support and confidence.
- Interpret and analyze the discovered association rules.

**Demo of OpenAI/TensorFlow Tools**

- Explore and experiment with OpenAI or TensorFlow tools and libraries.
- Perform a demonstration or mini-project showcasing the capabilities of the tools.
- Discuss and present the findings and potential applications.

**Module 2 (30 hours):**

**Practical based on Cyber & Information Security**

**Implementing Substitution and Transposition Ciphers:**

Design and implement algorithms to encrypt and decrypt messages using classical substitution and transposition techniques.

**RSA Encryption and Decryption:**

Implement the RSA algorithm for public-key encryption and decryption, and explore its properties and security considerations.

**Message Authentication Codes:**

Implement algorithms to generate and verify message authentication codes (MACs) for ensuring data integrity and authenticity.

**Digital Signatures:**

Implement digital signature algorithms such as RSA-based signatures, and verify the integrity and authenticity of digitally signed messages.

**Key Exchange using Diffie-Hellman:**

Implement the Diffie-Hellman key exchange algorithm to securely exchange keys between two entities over an insecure network.

**IP Security (IPsec) Configuration:**

	<p>Configure IPsec on network devices to provide secure communication and protect against unauthorized access and attacks.</p> <p><b>Web Security with SSL/TLS:</b></p> <p>Configure and implement secure web communication using SSL/TLS protocols, including certificate management and secure session establishment.</p> <p><b>Intrusion Detection System:</b></p> <p>Set up and configure an intrusion detection system (IDS) to monitor network traffic and detect potential security breaches or malicious activities.</p> <p><b>Malware Analysis and Detection:</b></p> <p>Analyze and identify malware samples using antivirus tools, analyze their behavior, and develop countermeasures to mitigate their impact.</p> <p><b>Firewall Configuration and Rule-based Filtering:</b></p> <p>Configure and test firewall rules to control network traffic, filter packets based on specified criteria, and protect network resources from unauthorized access.</p>
<b>10</b>	<p><b>Text Books</b></p> <ol style="list-style-type: none"> <li>1. Artificial Intelligence: A Modern Approach, Stuart Russell and Peter Norvig, 3rd Edition, Pearson, 2010.</li> <li>2. Cryptography and Network Security: Principles and Practice 7th edition, William Stallings, Pearson</li> </ol>
<b>11</b>	<p><b>Reference Books</b></p> <ol style="list-style-type: none"> <li>1. Artificial Intelligence: Foundations of Computational Agents, David L Poole, Alan K. Mackworth, 2nd Edition, Cambridge University Press, 2017.</li> <li>2. Artificial Intelligence, Kevin Knight and Elaine Rich, 3rd Edition, 2017</li> <li>3) The Elements of Statistical Learning, Trevor Hastie, Robert Tibshirani and Jerome Friedman, Springer, 2013</li> <li>3. Cryptography and Network Security, 2nd edition, Behrouz A Fourouzan, Debdeep Mukhopadhyay, TMH.</li> <li>4. Atul Kahate, "Cryptography and Network Security", Tata McGraw-Hill.</li> </ol>
<b>12</b>	<p><b>Internal Continuous Assessment: 40%</b>   <b>Semester End Examination: 60%</b></p>

**Name of the Course: Mini Project – I (*Real-World Application Development*)**

Sr. No.	Heading	Particulars
1	<b>Description of the course:</b>	<p><b>Introduction:</b></p> <p>Mini Project – I is a structured, development-oriented course that enables students to design and implement a complete real-world software solution. The course guides students through problem identification, requirement engineering, system modeling, architecture design, implementation, testing, and deployment. It emphasizes professional documentation, structured development practices, and industry-standard tools.</p> <p><b>Relevance:</b></p> <p>This course is highly relevant as it simulates an industry-style software development lifecycle from concept to deployment. Students experience real-world project constraints, stakeholder analysis, SDLC planning, and cloud deployment, preparing them for professional software development environments.</p> <p><b>Usefulness:</b></p> <p>The course develops practical competencies in requirement analysis, UML modeling, backend–frontend integration, database design, testing strategies, and version control using GitHub. It strengthens documentation skills, structured thinking, and collaborative development—essential capabilities for modern software engineers.</p> <p><b>Application:</b></p> <p>Students design solutions for real-world industry, social, or institutional problems such as service portals, management systems, automation tools, or mobile/web applications. The structured implementation and deployment process mirrors real software product development workflows used in startups, enterprises, and IT service organizations.</p> <p><b>Interest:</b></p> <p>The course generates strong engagement because students build a tangible, working application that solves a real problem. The opportunity to design architecture, implement features, deploy to cloud or mobile platforms, and demonstrate the final product makes the experience practically rewarding and professionally motivating.</p> <p><b>Connection with Other Courses:</b></p> <p>Mini Project – I integrates knowledge from Software</p>

		<p>Engineering, Database Management Systems, Web Development, Mobile Application Development, Cloud Computing, Cyber Security, and Data Structures. It acts as a capstone-style integrative experience that consolidates prior learning into a unified application.</p> <p><b>Demand in the Industry:</b></p> <p>Industry demands graduates who can independently conceptualize, design, develop, test, and deploy scalable applications. Skills such as requirement engineering, SDLC planning, UML modeling, GitHub version control, cloud deployment, and security validation are highly valued in software development, DevOps, and startup ecosystems.</p> <p><b>Job Prospects:</b></p> <p>Completion of this course strengthens readiness for roles such as software developer trainee, full-stack developer intern, backend/frontend developer, DevOps trainee, or application support engineer. A well-executed mini-project also enhances employability by serving as a portfolio project during placements.</p>
<b>2</b>	<b>Vertical:</b>	Major (Mandatory)
<b>3</b>	<b>Type:</b>	Practical
<b>4</b>	<b>Credits:</b>	2 credits (1 credit = 30 Hours of Practical work in a semester)
<b>5</b>	<b>Hours Allotted:</b>	60 hours
<b>6</b>	<b>Marks Allotted:</b>	50 Marks
<b>7</b>	<p><b>Course Objectives (CO):</b></p> <p><b>CO 1.</b> To enable students to identify and analyze real-world problems and conduct feasibility studies.</p> <p><b>CO 2.</b> To develop skills in requirement engineering, UML-based system modeling, and architecture design.</p> <p><b>CO 3.</b> To provide hands-on experience in application development, testing, and deployment.</p> <p><b>CO 4.</b> To inculcate best practices in version control, documentation, and project management.</p> <p><b>CO 5.</b> To prepare students for professional software development environments through structured SDLC implementation.</p>	
<b>8</b>	<p><b>Course Outcomes (OC):</b></p> <p>After successful completion of this course, students would be able to -</p> <p><b>OC 1.</b> Identify and analyse a real-world problem and prepare structured project documentation including SRS and feasibility reports.</p>	

	<p><b>OC 2.</b> Design complete system models using UML diagrams and develop suitable frontend, backend, and database architectures.</p> <p><b>OC 3.</b> Implement and integrate application components with authentication, validation, and error handling mechanisms.</p> <p><b>OC 4.</b> Test and deploy applications using systematic testing approaches and version control tools such as GitHub.</p> <p><b>OC 5.</b> Evaluate and present the developed solution through technical documentation, demonstration, and performance/security validation.</p>
9	<p><b>Modules:</b></p> <p><b>Module 1 (30 hours):</b></p> <hr/> <p><b>Problem Identification, Requirement Engineering &amp; System Design Phase</b></p> <p><b>Problem Identification &amp; Feasibility Study:</b> Identification of a real-world problem (industry / social / institutional), Problem justification and scope definition, Stakeholder identification, Feasibility analysis (technical, economic, operational)</p> <p><b>Requirement Engineering:</b> Functional requirements specification, Non-functional requirements, Use-case analysis, Requirement prioritization, Constraints and assumptions</p> <p><b>Software Development Life Cycle (SDLC) Planning:</b> Selection of SDLC model, Work Breakdown Structure (WBS), Project timeline (Gantt Chart), Resource planning</p> <p><b>System Modeling using UML:</b> Use Case Diagram, Class Diagram, Sequence Diagram, Activity Diagram, ER Diagram, Deployment Diagram</p> <p><b>System Architecture Design:</b> Frontend architecture, Backend architecture, Database schema design, API structure, Security considerations</p> <hr/> <p><b><u>Documentation Deliverables at the End of Module 1</u></b></p> <ul style="list-style-type: none"> <li>• Project Proposal</li> <li>• SRS Document</li> <li>• Complete UML Set</li> <li>• Architecture Design Document</li> </ul> <hr/> <p><b>Module 2 (30 hours):</b></p> <hr/> <p><b>Implementation, Testing, Deployment &amp; Evaluation Phase</b></p> <p><b>Application Development:</b> Frontend implementation, Backend implementation, Database integration, Authentication &amp; validation, Error handling</p> <p><b>Integration &amp; System Testing:</b> Unit testing, Black-box testing, Integration testing, Test case preparation, Bug tracking</p> <p><b>Deployment:</b> Cloud deployment / Local hosting, APK build, Server configuration (if applicable), Version control using GitHub (Mandatory)</p> <p><b>Performance &amp; Security Testing:</b> Basic load testing, Input validation checks, Security validation</p> <p><b>Final Documentation:</b> Technical Report, User Manual, Screenshots, Source Code Documentation</p>

	<b><u>Final Deliverables at the End of Module 2</u></b> <ul style="list-style-type: none"><li>• Working Application</li><li>• GitHub Repository</li><li>• Final Report</li><li>• Presentation &amp; Demonstration</li></ul>	
<b>10</b>	<b>Internal Continuous Assessment: 40%</b>	<b>Semester End Examination: 60%</b>

**SEM V**  
**Vertical – 1**  
**Electives**  
**(2+2)**

## Name of the Course: Software Testing & Quality Assurance

Sr. No.	Heading	Particulars
1	<b>Description the course:</b>	<p><b>Introduction:</b></p> <p>The Software Testing &amp; Quality Assurance course introduces students to systematic techniques for evaluating software quality and ensuring reliability. It covers testing fundamentals, SDLC alignment, verification and validation, white-box and black-box testing strategies, defect management, software metrics, statistical quality assurance, and quality improvement tools. The course builds a structured understanding of how quality is engineered into software systems.</p> <p><b>Relevance:</b></p> <p>In modern software-driven environments, delivering reliable, secure, and high-quality applications is critical. This course is highly relevant as it equips students with structured testing strategies and quality assurance principles that are essential in agile, DevOps, and large-scale enterprise development environments.</p> <p><b>Usefulness:</b></p> <p>The course provides practical knowledge of test case design, defect tracking, validation techniques, complexity metrics, and quality standards such as ISO 9000. It enables students to systematically evaluate software, measure reliability, improve processes, and contribute effectively to quality-driven development teams.</p> <p><b>Application:</b></p> <p>Software testing and QA principles are applied in web applications, enterprise systems, mobile apps, cloud platforms, embedded systems, and safety-critical systems. Techniques such as equivalence partitioning, boundary value analysis, unit and integration testing, defect lifecycle management, and statistical quality control are widely used in industry.</p> <p><b>Interest:</b></p> <p>The course is engaging as it reveals how hidden defects are identified, analyzed, and prevented through structured methodologies. Practical exposure to white-box and black-box techniques, defect metrics, reliability measurement, and quality tools like Pareto and cause-</p>

		<p>effect diagrams makes the subject analytically stimulating and professionally relevant.</p> <p><b>Connection with Other Courses:</b></p> <p>This course complements Software Engineering, Database Management Systems, Web Development, Operating Systems, and Cyber Security. It strengthens understanding of system validation, performance evaluation, reliability analysis, and structured development practices across the computer science curriculum.</p> <p><b>Demand in the Industry:</b></p> <p>There is consistent demand for professionals skilled in testing strategies, automation support, quality metrics, and defect management. IT companies, product-based firms, fintech organizations, and startups require trained QA engineers and test analysts to ensure software reliability and compliance with quality standards.</p> <p><b>Job Prospects:</b></p> <p>Students can pursue roles such as software test engineer, QA analyst, automation testing trainee, defect analyst, or quality assurance associate. With experience, career progression may include test lead, quality manager, process improvement specialist, or reliability engineer.</p>
2	<b>Vertical:</b>	Major (Elective)
3	<b>Type:</b>	Theory
4	<b>Credits:</b>	2 credits
5	<b>Hours Allotted:</b>	30 Hours
6	<b>Marks Allotted:</b>	50 Marks
7	<p><b>Course Objectives (CO):</b></p> <p><b>CO 1.</b> To develop understanding of software testing fundamentals and their integration within SDLC.</p> <p><b>CO 2.</b> To introduce verification, validation, and systematic testing strategies.</p> <p><b>CO 3.</b> To provide knowledge of white-box and black-box testing techniques and defect management processes.</p> <p><b>CO 4.</b> To familiarize students with software metrics, statistical quality assurance, and reliability concepts.</p> <p><b>CO 5.</b> To inculcate quality improvement practices and standards for professional software development.</p>	
8	<b>Course Outcomes (OC):</b>	

	<p>After successful completion of this course, students would be able to -</p> <p><b>OC 1.</b> Explain and apply fundamental testing principles, SDLC-based testing approaches, and V&amp;V mechanisms.</p> <p><b>OC 2.</b> Design effective test cases using black-box, white-box, and experience-based testing techniques.</p> <p><b>OC 3.</b> Analyse and manage defects using structured defect lifecycle and quality metrics.</p> <p><b>OC 4.</b> Evaluate software reliability and quality using statistical methods and software metrics.</p> <p><b>OC 5.</b> Apply quality assurance practices, review techniques, and quality improvement tools to enhance software processes.</p>
9	<p><b>Modules:</b></p> <p><b>Module 1 (15 hours):</b></p> <p><b>Software Testing Fundamentals:</b> Nature of errors and the need for testing, Basics of software testing process, Test case design principles and techniques, Test execution, reporting, and documentation</p> <p><b>Software Development Life Cycle (SDLC):</b> Overview of SDLC, phases and their relationship to testing, Role of testing in each phase, Software quality factors and their impact on testing</p> <p><b>Definition of Quality and Quality Assurance:</b> Understanding quality, in software development, Distinction between Quality Assurance (QA), Quality Control (QC), Quality Management (QM), and Software Quality Assurance (SQA)</p> <p><b>Verification and Validation (V&amp;V):</b> Definition of V&amp;V and its significance in software development, Different types of V&amp;V mechanisms, Concepts of Software Reviews, Inspection, and Walkthrough</p> <p><b>Software Testing Strategies:</b> Strategic approach to software testing, Unit Testing: purpose, techniques, and best practices; Integration Testing: approaches and challenges; Validation Testing: ensuring adherence to user requirements, System Testing: comprehensive end to end testing</p> <p><b>Module 2 (15 hours):</b></p> <p><b>White Box Testing and Black Box Testing:</b> Functional/Specification, based Testing as Black Box, Black box: Equivalence Partitioning, Boundary Value Analysis, Decision Table Testing, State Transition, Testing. Structural Testing as White Box, White Box: Statement testing, Branch testing. Experience-based: Error guessing, Exploratory testing, Checklist-based testing.</p> <p><b>Software Metrics:</b> Concept of software metrics and their importance, Developing and utilizing different types of metrics, Complexity metrics, and their significance in testing</p> <p><b>Defect Management:</b> Definition of defects and their lifecycle, Defect management process, including defect reporting and tracking, Metrics related to defects and their utilization for process improvement</p>

	<p><b>Software Quality Assurance:</b> Understanding quality concepts and the Quality Movement: Background issues and challenges in SQA, Activities and approaches in Software Quality Assurance, Statistical Quality Assurance and Software Reliability, Statistical process control techniques: Software reliability measurement and improvement, ISO 9000 Quality Standards</p> <p><b>Software Reviews &amp; Quality Improvement Techniques:</b> Formal Technical Reviews and their benefits; Introduction to quality, improvement methodologies, Utilizing quality costs for decision making, Introduction to quality improvement tools: Pareto Diagrams, Cause-effect Diagrams, Scatter Diagrams, Run charts</p>
<b>10</b>	<p><b>Text Books</b></p> <ol style="list-style-type: none"> <li>1. Software Engineering for Students, A Programming Approach, Douglas Bell, 4th Edition,, Pearson Education, 2005</li> <li>2. Software Engineering – A Practitioners Approach, Roger S. Pressman, 7th Edition, Tata McGraw Hill</li> </ol>
<b>11</b>	<p><b>Reference Books</b></p> <ol style="list-style-type: none"> <li>1. Quality Management, Donna C. S. Summers, 5th Edition, Prentice-Hall.</li> <li>2. Software Testing and Quality Assurance Theory and Practice, Kshirsagar Naik, Priyadarshi Tripathy , John Wiley &amp; Sons, Inc. , Publication.</li> </ol>
<b>12</b>	<p><b>Internal Continuous Assessment: 40%</b>      <b>Semester End Examination: 60%</b></p>

## Name of the Course: Wireless & Sensor Networks

Sr. No.	Heading	Particulars
1	<b>Description the course:</b>	<p><b>Introduction:</b></p> <p>The Wireless &amp; Sensor Networks course introduces students to the principles, architecture, and communication mechanisms of wireless sensor systems and mobile communication technologies. It covers sensor node design, network architecture, MAC and routing protocols, WSN operating systems, ad-hoc networking, and advanced wireless transmission fundamentals including GSM and satellite systems.</p> <p><b>Relevance:</b></p> <p>Wireless sensor networks form the backbone of modern IoT, smart cities, environmental monitoring, industrial automation, and mobile communication systems. This course is highly relevant as it provides foundational knowledge required to design, deploy, and optimize wireless and sensor-based communication systems in real-world environments.</p> <p><b>Usefulness:</b></p> <p>The course equips students with technical understanding of energy-efficient networking, routing challenges, MAC protocols, radio technologies, and wireless transmission principles. It develops analytical skills necessary for designing scalable, reliable, and secure wireless communication infrastructures.</p> <p><b>Application:</b></p> <p>WSN concepts are applied in smart agriculture, healthcare monitoring, industrial IoT, disaster management, military surveillance, environmental sensing, and smart grids. Knowledge of GSM, UMTS, satellite communication, and IEEE 802.15.4 standards supports applications in telecom networks, mobile communication systems, and satellite-based services.</p> <p><b>Interest:</b></p> <p>The subject is engaging as it explores how distributed sensor nodes communicate wirelessly under constraints such as limited power and bandwidth. Topics like ad-hoc networking, energy optimization, mobile handover, satellite routing, and spread spectrum communication make the course technically stimulating and industry-relevant.</p>

		<p><b>Connection with Other Courses:</b></p> <p>This course complements Computer Networks, Operating Systems, Embedded Systems, IoT, Cyber Security, and Cloud Computing. It integrates concepts of protocol design, system architecture, signal processing fundamentals, and secure communication in distributed environments.</p> <p><b>Demand in the Industry:</b></p> <p>There is growing demand for professionals skilled in IoT systems, wireless communication, telecom infrastructure, and sensor-based automation. Industries such as telecommunications, smart manufacturing, defense, automotive systems, and satellite communication require expertise in wireless network design and optimization.</p> <p><b>Job Prospects:</b></p> <p>Students can pursue roles such as wireless network engineer, IoT developer, telecom systems analyst, network planning engineer, embedded systems engineer, or communication systems trainee. Advanced specialization can lead to careers in telecom research, satellite communication systems, and next-generation wireless technologies.</p>
2	<b>Vertical:</b>	Major (Elective)
3	<b>Type:</b>	Theory
4	<b>Credits:</b>	2 credits
5	<b>Hours Allotted:</b>	30 Hours
6	<b>Marks Allotted:</b>	50 Marks
7	<p><b>Course Objectives (CO):</b></p> <p><b>CO 1.</b> To introduce architectural principles and design challenges of Wireless Sensor Networks.</p> <p><b>CO 2.</b> To develop understanding of WSN operating systems, MAC protocols, routing strategies, and middleware concepts.</p> <p><b>CO 3.</b> To provide knowledge of wireless transmission fundamentals and radio communication techniques.</p> <p><b>CO 4.</b> To familiarize students with GSM, advanced mobile systems, and satellite communication technologies.</p> <p><b>CO 5.</b> To enable analysis of energy efficiency, security, and optimization in wireless and sensor networks.</p>	
8	<p><b>Course Outcomes (OC):</b></p> <p>After successful completion of this course, students would be able to -</p>	

	<p><b>OC 1.</b> Explain WSN architecture, sensor node technologies, and network optimization principles.</p> <p><b>OC 2.</b> Analyze and apply MAC protocols, routing strategies, and middleware concepts in WSN environments.</p> <p><b>OC 3.</b> Evaluate energy efficiency, security, and design challenges in ad-hoc and sensor networks.</p> <p><b>OC 4.</b> Interpret wireless transmission fundamentals including modulation, spread spectrum, and cellular communication principles.</p> <p><b>OC 5.</b> Describe and assess mobile communication systems (GSM, UMTS) and satellite communication architectures.</p>
9	<p><b>Modules:</b></p> <p><b>Module 1 (15 hours):</b></p> <p><b>Introduction and Overview of WSNs:</b> Basic sensor network architectural elements, Advantages and challenges of WSNs, Applications of WSNs, Sensor node technology, Sensor taxonomy, WSN operating environment, Radio technology in WSNs, Network architecture and optimization, Optimization goals, figures of merit, Design principles for WSNs, Service interfaces of WSNs, Gateway concepts</p> <p><b>WSN Operating Systems and Ad-hoc Networks:</b> Overview of wireless sensor network operating systems, Examples of WSN operating systems (case/examples), Ad-hoc networks in WSNs, Characteristics and challenges of ad-hoc networks in WSNs, Energy efficiency considerations in ad-hoc networks, Security and privacy issues in ad-hoc networks</p> <p><b>Medium Access Control (MAC) in WSN:</b> Fundamentals of MAC protocols for sensor networks, Sensor-MAC (S-MAC) / Sensor-MAC case study</p> <p><b>Routing in WSN:</b> Routing challenges and design issues in WSNs, Routing strategies in WSNs, IEEE 802.15.4 LR-WPAN standard (case study),</p> <p><b>Transport Layer and Middleware in WSN:</b> Traditional transport control protocols, Transport protocol design issues in WSNs, WSN middleware architecture</p> <p><b>Module 2 (15 hours):</b></p> <p><b>Wireless Transmission Fundamentals:</b> Frequency for radio transmission, Signals, Antennas, Signal propagation, Multiplexing, Modulation, Spread spectrum, Cellular systems</p> <p><b>GSM and Mobile Communication Systems:</b> GSM mobile services, System architecture, Radio interface, Protocols, Localization and calling, Handover, Security, New data services</p> <p><b>Advanced Mobile Communication Systems:</b> DECT: system architecture and protocol architecture; TETRA, UMTS, IMT-2000</p> <p><b>Satellite and Broadcast Systems:</b> History of satellite systems, Applications, Satellite basics: GEO, LEO, MEO; Routing in satellite systems, Localization, Handover</p>

<b>10</b>	<p><b>Text Books</b></p> <ol style="list-style-type: none"> <li>1. Wireless Sensor Networks Technology, Protocols, and Applications ,Kazem Sohraby, Daniel Minoli and TaiebZnati, John Wiley &amp; Sons, 2017</li> <li>2. Protocols and Architectures for Wireless Sensor Network, Holger Kerl, Andreas Willig, John Wiley and Sons, 2015</li> </ol>
<b>11</b>	<p><b>Reference Books</b></p> <ol style="list-style-type: none"> <li>1. Fundamentals of Wireless Sensor Networks, Theory and Practice, Walteneus Dargie, Christian Poellabauer , Wiley Series on wireless Communication and Mobile Computing, 2011</li> <li>2. Networking Wireless Sensors, Bhaskar Krishnamachari , Cambridge University Press, 2005</li> </ol>
<b>12</b>	<p><b>Internal Continuous Assessment: 40%    Semester End Examination: 60%</b></p>

## Name of the Course: Software Testing & Quality Assurance Practical

Sr. No.	Heading	Particulars
1	<b>Description</b> the <b>course:</b>	<p><b>Introduction:</b></p> <p>The Software Testing &amp; Quality Assurance Practical course provides hands-on exposure to automated and manual testing methodologies using industry-standard tools. It covers Selenium IDE, WebDriver, Selenium Grid, TestNG, Jenkins, JMeter, and Bugzilla, enabling students to design, execute, and manage test cases while implementing automation frameworks and continuous integration workflows.</p> <p><b>Relevance:</b></p> <p>With software quality being a critical success factor in modern IT systems, practical expertise in automation and testing tools is highly relevant. This course bridges theoretical QA concepts with real-time implementation, preparing students for agile and DevOps-driven development environments.</p> <p><b>Usefulness:</b></p> <p>The course develops practical competencies in test suite creation, cross-browser testing, data-driven testing, framework development (POM), CI/CD integration, load testing, and defect lifecycle management. These skills enable students to validate application reliability, performance, and usability effectively.</p> <p><b>Application:</b></p> <p>The practical skills acquired are applicable in web application testing, enterprise software validation, regression automation, performance benchmarking, cross-browser compatibility testing, and CI-based quality monitoring. Tools like Selenium, JMeter, Jenkins, and Bugzilla are widely used in IT service companies and product-based organizations.</p> <p><b>Interest:</b></p> <p>The course is engaging because students automate real-world web functionalities such as login validation, file uploads, table handling, and API responses. Exposure to automation frameworks, continuous integration, load testing, and bug tracking provides an industry-simulated testing experience.</p> <p><b>Connection with Other Courses:</b></p>

		<p>This course complements Software Testing &amp; QA (theory), Software Engineering, Web Development, Operating Systems, and DevOps-related subjects. It reinforces understanding of SDLC, defect management, automation strategies, and performance optimization.</p> <p><b>Demand in the Industry:</b></p> <p>There is strong industry demand for automation testers, QA engineers, and DevOps-integrated testing professionals. Knowledge of Selenium, TestNG, Jenkins, JMeter, and defect tracking tools is highly valued across IT services, fintech, e-commerce, and product development firms.</p> <p><b>Job Prospects:</b></p> <p>Students can pursue roles such as automation test engineer, QA analyst, Selenium automation tester, performance testing trainee, DevOps testing associate, or defect management analyst. Advanced expertise can lead to roles such as test lead, automation architect, or quality assurance manager.</p>
2	<b>Vertical:</b>	Major (Elective)
3	<b>Type:</b>	Practical
4	<b>Credits:</b>	2 credits (1 credit = 30 Hours of Practical work in a semester)
5	<b>Hours Allotted:</b>	60 hours
6	<b>Marks Allotted:</b>	50 Marks
7	<p><b>Course Objectives (CO):</b></p> <p><b>CO 1.</b> To develop proficiency in designing and executing manual and automated test cases using Selenium tools.</p> <p><b>CO 2.</b> To enable implementation of automation frameworks such as Page Object Model and TestNG-based testing.</p> <p><b>CO 3.</b> To provide hands-on experience in cross-browser testing, CI integration, and load testing.</p> <p><b>CO 4.</b> To strengthen understanding of defect tracking and test reporting using industry tools.</p> <p><b>CO 5.</b> To prepare students for professional QA and automation roles through tool-based implementation.</p>	
8	<p><b>Course Outcomes (OC):</b></p> <p>After successful completion of this course, students would be able to -</p> <p><b>OC 1.</b> Design and execute automated test scripts using Selenium IDE and WebDriver for functional validation.</p> <p><b>OC 2.</b> Implement automation frameworks (POM, TestNG) and perform cross-browser testing using Selenium Grid.</p>	

	<p><b>OC 3.</b> Integrate automation scripts with CI tools like Jenkins and generate structured test reports.</p> <p><b>OC 4.</b> Perform load testing using Apache JMeter and analyze performance metrics.</p> <p><b>OC 5.</b> Manage and track software defects using Bugzilla and prepare structured defect lifecycle reports.</p>
9	<p><b>Modules:</b></p> <p><b>Module 1 (30 hours):</b></p> <hr/> <p><b>Creation of Test Suite Using Selenium IDE</b></p> <p>Tool: Selenium IDE (Chrome/Firefox Extension)</p> <p>Install Selenium IDE and create a test suite containing a minimum of four test cases to validate different web page formats (HTML page, XML response, JSON API output, and a dynamic form page). Execute the suite and export the automation script in WebDriver format.</p> <p><b>Cross-Website Functional Testing Using Selenium IDE</b></p> <p>Tool: Selenium IDE</p> <p>Develop and execute a test suite for two different websites. Perform functional operations such as clicking links, filling forms, selecting dropdown options, and verifying displayed content. Analyze test results and generate execution logs.</p> <p><b>Manual Test Case Design and Execution</b></p> <p>Tool: MS Excel / Google Sheets</p> <p>Prepare a Test Plan, Test Scenarios, and Test Cases for a given web application. Execute test cases manually and prepare a Test Execution Report and Defect Report as per STLC process.</p> <p><b>Selenium Server (Grid) Configuration and Remote Execution</b></p> <p>Tool: Selenium Server (Grid)</p> <p>Install and configure Selenium Server in Hub-Node architecture. Execute a test script on multiple browsers using Remote WebDriver and compare cross-browser results.</p> <p><b>Login Automation and Validation</b></p> <p>Tool: Selenium WebDriver (Java/PHP)</p> <p>Write an automation script to perform login on a specified web page. Verify successful login using assertions such as URL validation, welcome message validation, and logout visibility check.</p> <p><b>Web Element Interaction and Synchronization Handling</b></p> <p>Tool: Selenium WebDriver (Java)</p> <p>Develop a script to automate interaction with various web elements (textbox, radio button, checkbox, dropdown, alert, and frames). Implement implicit and explicit waits for synchronization handling.</p> <p><b>Data-Driven Testing Using Excel Integration</b></p> <p>Tool: Selenium WebDriver + Apache POI</p>

Write a program to update 10 student records in an Excel file and validate changes on the web application. Perform data reading, writing, and verification using Apache POI library.

#### **Data Extraction and Score Analysis from Web Table**

Tool: Selenium WebDriver

Automate extraction of student marks from a dynamic web table and determine the number of students scoring above 60 in at least one subject. Compute total count and percentage.

#### **Object Identification and Counting on Web Page**

Tool: Selenium WebDriver

Write a program to identify and count the total number of objects present on a web page including links, images, buttons, and input elements. Display categorized counts.

#### **List and Combo Box Item Verification**

Tool: Selenium WebDriver

Automate the process of identifying a list or dropdown box on a web page and determine the total number of items. Validate default selected value and perform selection testing.

### **Module 2 (30 hours):**

#### **Checkbox Identification and Validation**

Tool: Selenium WebDriver

Write a program to count the total number of checkboxes on a web page, including checked and unchecked counts. Perform dynamic selection and validation.

#### **Dynamic Web Table Handling and Sorting Validation**

Tool: Selenium WebDriver

Automate extraction of data from a dynamic web table. Validate sorting functionality and verify correctness of filtered results.

#### **File Upload and Download Automation**

Tool: Selenium WebDriver + Robot Class

Develop a script to automate file upload and download functionality. Validate successful upload and verify file existence after download.

#### **Screenshot Capture and Logging Mechanism**

Tool: Selenium WebDriver + Log4j

Implement screenshot capture on test failure and integrate logging mechanism using Log4j. Generate execution logs and analyze error messages.

#### **Page Object Model (POM) Framework Implementation**

Tool: Selenium WebDriver + TestNG

Design and implement automation framework using Page Object Model (POM).

	<p>Create reusable page classes and execute modular test cases.</p> <p><b>TestNG Framework Implementation and Reporting</b></p> <p>Tool: Selenium WebDriver + TestNG</p> <p>Create and execute automated test cases using TestNG annotations. Group test cases, define priorities, and generate HTML test reports.</p> <p><b>Continuous Integration using Jenkins</b></p> <p>Tool: Jenkins + Selenium WebDriver</p> <p>Configure Jenkins to execute Selenium automation scripts automatically. Generate build reports and analyze test execution history.</p> <p><b>Cross-Browser Testing Using Selenium Grid</b></p> <p>Tool: Selenium Grid</p> <p>Execute automation scripts across multiple browsers (Chrome, Firefox, Edge) using Selenium Grid. Compare execution time and compatibility issues.</p> <p><b>Load Testing Using Apache JMeter</b></p> <p>Tool: Apache JMeter</p> <p>Design and execute load testing scenarios for a web application. Configure thread groups, ramp-up time, and loop count. Analyze response time, throughput, and error percentage using graphical reports.</p> <p><b>Bug Tracking and Defect Lifecycle Management</b></p> <p>Tool: Bugzilla</p> <p>Log software defects using Bugzilla. Assign severity and priority levels. Track defect lifecycle stages and generate defect summary reports.</p>
<b>10</b>	<p><b>Text Books</b></p> <ol style="list-style-type: none"> <li>1. Software Engineering for Students, A Programming Approach, Douglas Bell, 4th Edition,, Pearson Education, 2005</li> <li>2. Software Engineering – A Practitioners Approach, Roger S. Pressman, 7th Edition, Tata McGraw Hill</li> </ol>
<b>11</b>	<p><b>Reference Books</b></p> <ol style="list-style-type: none"> <li>1. Quality Management, Donna C. S. Summers, 5th Edition, Prentice-Hall.</li> <li>2. Software Testing and Quality Assurance Theory and Practice, Kshirsagar Naik, Priyadarshi Tripathy, John Wiley &amp; Sons, Inc., Publication.</li> </ol>
<b>12</b>	<p><b>Internal Continuous Assessment: 40%   Semester End Examination: 60%</b></p>

## Name of the Course: Wireless & Sensor Networks Practical

Sr. No.	Heading	Particulars
1	<b>Description of the course:</b>	<p><b>Introduction:</b></p> <p>The Wireless &amp; Sensor Networks Practical course provides hands-on experience in simulating, configuring, and analyzing wireless sensor nodes, ad-hoc networks, and mobile communication systems. Using tools such as TinyOS, nesC, TOSSIM, and network simulators, students explore sensor node architecture, routing protocols, MAC mechanisms, and network performance evaluation in dynamic wireless environments.</p> <p><b>Relevance:</b></p> <p>With the rapid growth of IoT, smart cities, industrial automation, and mobile communication systems, practical understanding of WSN and MANET technologies is highly relevant. This course bridges theoretical concepts with simulation-based implementation, enabling students to analyze real-world wireless network behavior and performance constraints.</p> <p><b>Usefulness:</b></p> <p>The course equips students with skills in sensor node programming, event-driven system modeling, routing protocol simulation, MAC protocol comparison, and energy-aware communication design. It strengthens analytical abilities in evaluating throughput, delay, packet delivery ratio, and energy efficiency in wireless networks.</p> <p><b>Application:</b></p> <p>Practical exposure gained in this course applies to IoT deployments, environmental monitoring systems, mobile ad-hoc networks, telecom infrastructure, and satellite communication systems. Skills in routing protocol analysis, network coverage evaluation, and MAC optimization are essential in wireless system design and network planning.</p> <p><b>Interest:</b></p> <p>The course is technically engaging as students simulate dynamic topologies, observe packet animations, analyze routing behaviors, and evaluate energy conservation strategies. Working with TinyOS, nesC, TOSSIM, and protocol simulations makes learning interactive and closely aligned with real-world wireless communication scenarios.</p>

		<p><b>Connection with Other Courses:</b></p> <p>This course complements Wireless &amp; Sensor Networks (theory), Computer Networks, Operating Systems, Embedded Systems, IoT, and Cloud Computing. It reinforces protocol analysis, system-level programming, performance evaluation, and distributed system design concepts.</p> <p><b>Demand in the Industry:</b></p> <p>Industries working in telecommunications, IoT solutions, defense communication systems, automotive networks, and smart infrastructure require professionals skilled in wireless protocol design and performance evaluation. Expertise in routing protocols, MAC strategies, and energy-efficient network design is increasingly valued.</p> <p><b>Job Prospects:</b></p> <p>Students can pursue roles such as wireless network engineer, IoT system developer, telecom protocol analyst, embedded systems engineer, network simulation engineer, or communication systems trainee. Advanced specialization can lead to research and development roles in next-generation wireless technologies.</p>
2	<b>Vertical:</b>	Major (Elective)
3	<b>Type:</b>	Practical
4	<b>Credits:</b>	2 credits (1 credit = 30 Hours of Practical work in a semester)
5	<b>Hours Allotted:</b>	60 hours
6	<b>Marks Allotted:</b>	50 Marks
7	<p><b>Course Objectives (CO):</b></p> <p><b>CO 1.</b> To provide practical understanding of sensor node architecture and TinyOS-based programming.</p> <p><b>CO 2.</b> To develop skills in simulating wireless sensor and ad-hoc networks using TOSSIM and network simulation tools.</p> <p><b>CO 3.</b> To enable implementation and analysis of routing and MAC protocols in WSN and MANET environments.</p> <p><b>CO 4.</b> To evaluate network performance metrics including throughput, delay, coverage, and energy efficiency.</p> <p><b>CO 5.</b> To strengthen practical knowledge of wireless communication and mobile network simulation.</p>	
8	<p><b>Course Outcomes (OC):</b></p> <p>After successful completion of this course, students would be able to -</p>	

	<p><b>OC 1.</b> Design and simulate sensor node architectures and TinyOS-based applications using nesC and TOSSIM.</p> <p><b>OC 2.</b> Implement and analyze routing protocols (AODV, DSR) and generate routing tables under dynamic topologies.</p> <p><b>OC 3.</b> Simulate and evaluate MAC protocols (CSMA/CA, TDMA, energy-aware MAC) for performance and energy efficiency.</p> <p><b>OC 4.</b> Measure and compare network performance metrics such as throughput, packet delivery ratio, and delay.</p> <p><b>OC 5.</b> Model and interpret wireless, MANET, and cellular communication scenarios through simulation and coverage analysis.</p>
9	<p><b>Modules:</b></p> <p><b>Module 1 (30 hours):</b></p> <p><b>Study of Sensor Node Hardware Architecture</b> Design a conceptual block diagram of a wireless sensor node and analyse the role of sensing unit, microcontroller, transceiver, memory, and power supply in energy-constrained environments.</p> <p><b>Sensor Mote and Base Station Communication Study</b> Develop a simple communication scenario between multiple sensor motes and a base station to study data aggregation and sink node operation.</p> <p><b>Understanding TinyOS Architecture and Execution Model</b> Examine the event-driven architecture of TinyOS and implement a simple application to demonstrate its non-preemptive scheduler.</p> <p><b>Implementation of nesC Programming Model</b> Create a basic nesC application using modules and configurations to understand component wiring and interface binding.</p> <p><b>Implementation of Events, Commands, and Tasks in TinyOS</b> Develop a TinyOS program that toggles LEDs using events and tasks to demonstrate split-phase execution.</p> <p><b>Simulation of Single Mote using TOSSIM</b> Simulate a single sensor node in TOSSIM and observe execution logs to understand the TinyOS runtime environment.</p> <p><b>Mote-to-Mote Radio Communication Simulation</b> Simulate packet transmission between two motes and analyze signal strength and packet loss under varying radio gain values.</p> <p><b>Mote-to-PC Serial Communication Simulation</b> Implement serial communication between a sensor mote and PC to monitor real-time data transmission.</p> <p><b>Creation of Simple Adhoc Network using TOSSIM</b> Create a multi-node ad-hoc network in TOSSIM and evaluate broadcast</p>

	<p>communication among nodes.</p> <p><b>Routing Table Generation and Analysis in WSN</b></p> <p>Implement a simple routing mechanism and analyze routing table updates during topology changes.</p>
	<p><b>Module 2 (30 hours):</b></p>
	<p><b>Basic MANET Implementation with Packet Animation</b></p> <p>Simulate a mobile ad-hoc network and visualize packet movement to study dynamic topology formation.</p> <p><b>Implementation of AODV Routing Protocol</b></p> <p>Simulate the AODV routing protocol and analyze route discovery and route maintenance mechanisms.</p> <p><b>Implementation of DSR Routing Protocol</b></p> <p>Simulate the DSR protocol and compare routing overhead with AODV.</p> <p><b>Performance Evaluation of MANET Protocols</b></p> <p>Measure throughput, packet delivery ratio, and end-to-end delay for different MANET routing protocols.</p> <p><b>Simulation of CSMA/CA MAC Protocol</b></p> <p>Implement CSMA/CA protocol simulation and evaluate collision avoidance mechanisms.</p> <p><b>Simulation of TDMA-based MAC Protocol</b></p> <p>Simulate TDMA slot allocation and compare energy efficiency with CSMA.</p> <p><b>Energy-Aware MAC Protocol for WSN</b></p> <p>Implement duty-cycling MAC protocol and evaluate energy conservation in sensor networks.</p> <p><b>MANET Simulation with Directional Antenna</b></p> <p>Simulate MANET using directional antennas and analyze improvements in spatial reuse and interference reduction.</p> <p><b>Wireless Sensor Network Deployment and Coverage Analysis</b></p> <p>Simulate random and grid deployment of sensor nodes and evaluate network coverage and connectivity.</p> <p><b>Cellular Network Simulation with Client-Server Communication</b></p> <p>Simulate a mobile network consisting of a cell tower, central office server, web server, and web browser, and analyze packet flow during a data request-response cycle.</p>
<b>10</b>	<p><b>Text Books</b></p> <p>1. Wireless Sensor Networks Technology, Protocols, and Applications ,Kazem Sohrawy, Daniel Minoli and TaiebZnati, John Wiley &amp; Sons, 2017</p>

	2. Protocols and Architectures for Wireless Sensor Network, Holger Kerl, Andreas Willig, John Wiley and Sons, 2015
<b>11</b>	<p><b>Reference Books</b></p> <p>1. Fundamentals of Wireless Sensor Networks, Theory and Practice, Waltenegeus Dargie, Christian Poellabauer , Wiley Series on wireless Communication and Mobile Computing, 2011</p> <p>2. Networking Wireless Sensors, Bhaskar Krishnamachari , Cambridge University Press, 2005</p>
<b>12</b>	<b>Internal Continuous Assessment: 40%    Semester End Examination: 60%</b>

**Vertical – 4**

**VSC**  
**(2)**  
**(Practical)**

## Name of the Course: Ethical Hacking

Sr. No.	Heading	Particulars
1	Description the course:	<p><b>Introduction:</b></p> <p>The Ethical Hacking course provides a structured and defensive understanding of cyber-attack methodologies, system vulnerabilities, and penetration testing frameworks. It covers reconnaissance, scanning, enumeration, privilege escalation, cryptography weaknesses, web application threats, malware analysis, wireless security, exploit frameworks, and professional penetration testing methodologies. The course emphasizes legal boundaries and ethical responsibility in cybersecurity practices.</p> <p><b>Relevance:</b></p> <p>With increasing cyber threats targeting governments, enterprises, and individuals, ethical hacking has become a critical cybersecurity discipline. This course is highly relevant as it equips students with knowledge of attacker techniques from a defensive perspective, enabling them to identify, analyze, and mitigate vulnerabilities proactively.</p> <p><b>Usefulness:</b></p> <p>The course develops practical and analytical skills in vulnerability research, network scanning, web application security assessment, password security evaluation, and penetration testing lifecycle management. It strengthens defensive thinking by understanding exploit logic, attack surfaces, and security hardening strategies.</p> <p><b>Application:</b></p> <p>Ethical hacking principles are applied in security audits, penetration testing engagements, vulnerability assessments, SOC operations, web security testing, wireless security evaluation, and compliance verification. Knowledge of OWASP vulnerabilities, SQL injection logic, buffer overflow concepts, DoS mitigation, and exploit frameworks supports secure system design and organizational defense.</p> <p><b>Interest:</b></p> <p>The subject is engaging because it explores real-world cyber-attack techniques such as social engineering, session hijacking, ARP poisoning, SQL injection, and</p>

		<p>malware analysis. Understanding how attackers think and how vulnerabilities are exploited makes the course technically stimulating and socially impactful.</p> <p><b>Connection with Other Courses:</b></p> <p>This course complements Cyber &amp; Information Security, Computer Networks, Operating Systems, Cryptography, Software Testing, and Cloud Security. It integrates networking fundamentals, encryption principles, system-level understanding, and secure coding practices into a unified security-oriented framework.</p> <p><b>Demand in the Industry:</b></p> <p>There is strong global demand for ethical hackers, penetration testers, SOC analysts, and cybersecurity consultants. Organizations require professionals skilled in vulnerability assessment, exploit analysis, web security testing, and secure configuration management to protect digital infrastructure.</p> <p><b>Job Prospects:</b></p> <p>Students can pursue roles such as penetration testing trainee, cybersecurity analyst, SOC analyst, vulnerability assessment engineer, security consultant, or incident response associate. Advanced expertise can lead to positions such as red team specialist, security architect, or cybersecurity researcher.</p>
<b>2</b>	<b>Vertical:</b>	VSC
<b>3</b>	<b>Type:</b>	Practical
<b>4</b>	<b>Credits:</b>	2 credits (1 credit = 30 Hours of Practical work in a semester)
<b>5</b>	<b>Hours Allotted:</b>	60 hours
<b>6</b>	<b>Marks Allotted:</b>	50 Marks
<b>7</b>	<p><b>Course Objectives (CO):</b></p> <p><b>CO 1.</b> To introduce foundational concepts, terminology, and legal frameworks in ethical hacking.</p> <p><b>CO 2.</b> To develop understanding of reconnaissance, scanning, enumeration, and system exploitation techniques from a defensive perspective.</p> <p><b>CO 3.</b> To provide knowledge of web application, network, and wireless security vulnerabilities and mitigation strategies.</p> <p><b>CO 4.</b> To familiarize students with exploit frameworks and structured penetration testing methodologies.</p>	

	<b>CO 5.</b> To inculcate ethical responsibility and professional reporting standards in cybersecurity practice.
<b>8</b>	<p><b>Course Outcomes (OC):</b></p> <p>After successful completion of this course, students would be able to -</p> <p><b>OC 1.</b> Explain and analyze various cyber-attack methodologies including reconnaissance, scanning, and privilege escalation techniques.</p> <p><b>OC 2.</b> Identify and assess vulnerabilities in web applications, networks, wireless systems, and authentication mechanisms.</p> <p><b>OC 3.</b> Evaluate cryptographic weaknesses, session management flaws, and malware behaviors from a defensive standpoint.</p> <p><b>OC 4.</b> Apply structured penetration testing methodologies and prepare professional vulnerability assessment reports.</p> <p><b>OC 5.</b> Recommend appropriate mitigation strategies and security hardening measures while adhering to legal and ethical guidelines.</p>
<b>9</b>	<p><b>Modules:</b></p> <p><b>Module 1 (30 hours):</b></p> <p><b>Foundations of Ethical Hacking and Cyber Terminology:</b> Study core terminology including hacking types, hacker classes, hacktivism, ethical hacking phases, and legal boundaries. Prepare a structured lifecycle model of ethical hacking.</p> <p><b>Vulnerability Research and Disclosure Mechanisms:</b> Understand vulnerability lifecycle, CVE identification, CVSS scoring, responsible disclosure, and bug bounty frameworks. Analyze a real vulnerability case study.</p> <p><b>Footprinting and Information Gathering Methodology:</b> Explore passive and active reconnaissance techniques including OSINT, competitive intelligence, and search engine reconnaissance strategies.</p> <p><b>DNS Enumeration and Domain Intelligence:</b> Study DNS structure, record types (A, MX, NS, TXT, CNAME), WHOIS lookup, and ARIN databases to understand infrastructure mapping.</p> <p><b>Social Engineering and Human Exploitation Techniques:</b> Examine psychological manipulation attacks including phishing, pretexting, baiting, and impersonation. Analyze real-world social engineering cases.</p> <p><b>Network Scanning and Port Scanning Techniques:</b> Understand TCP/IP behavior and scanning methodologies including SYN, FIN, NULL, XMAS, and ACK scans, along with firewall detection logic.</p> <p><b>Enumeration and Service Identification:</b> Study enumeration techniques such as SNMP enumeration, NetBIOS scanning, and service fingerprinting to identify exposed system services.</p> <p><b>System Hacking and Privilege Escalation Concepts:</b> Analyze password cracking methods, privilege escalation techniques, rootkits, and spyware technologies from a defensive perspective.</p> <p><b>Cryptography and Password Security Analysis:</b> Differentiate between encryption</p>

	<p>and hashing mechanisms, password storage models, salting techniques, and cryptographic weaknesses.</p> <p><b>Network Sniffing and Man-in-the-Middle Attacks:</b> Study packet sniffing techniques, ARP poisoning, MAC flooding, DNS spoofing, and countermeasures to secure network communication.</p> <hr/> <p><b>Module 2 (30 hours):</b></p> <p><b>Denial of Service and Botnet Architecture:</b> Examine types of DoS/DDoS attacks including SYN flooding and Smurf attacks. Understand botnet structures and mitigation strategies.</p> <p><b>Session Hijacking and Token Security:</b> Analyze session management vulnerabilities, cookie manipulation, sequence prediction, and prevention mechanisms like secure flags and HTTPS.</p> <p><b>Web Server Vulnerabilities and Hardening Techniques:</b> Study common web server misconfigurations, directory traversal attacks, patch management practices, and hardening strategies.</p> <p><b>Web Application Threats and OWASP Vulnerabilities:</b> Examine major web application vulnerabilities (OWASP Top 10), input validation flaws, and search engine exploitation techniques.</p> <p><b>SQL Injection – Attack Logic and Prevention:</b> Understand SQL injection mechanisms, query manipulation techniques, database vulnerabilities, and secure coding practices.</p> <p><b>Buffer Overflow and Memory Exploitation Concepts:</b> Study stack-based buffer overflow, memory layout, return address overwriting, and secure programming techniques.</p> <p><b>Wireless Network Security and Authentication Mechanisms:</b> Compare WEP, WPA, WPA2, WPA3 protocols, wireless sniffing risks, rogue access points, and wireless security best practices.</p> <p><b>Malware, Keyloggers, and Spyware Analysis:</b> Examine the architecture of keyloggers and spyware, detection methods, behavioral analysis, and endpoint protection mechanisms.</p> <p><b>Exploit Frameworks and Ethical Use of Metasploit:</b> Study exploit lifecycle, payload concepts, post-exploitation activities, and ethical/legal considerations of penetration testing tools.</p> <p><b>Penetration Testing Methodologies and Reporting:</b> Understand PTES and OWASP methodologies, penetration testing phases, risk assessment models, and professional report preparation standards.</p>
10	<p><b>Text Books</b></p> <ol style="list-style-type: none"> <li>1. CEH official Certified Ethical Hacking Review Guide, Wiley India Edition</li> </ol>

<b>11</b>	<b>Reference Books</b> 1. Certified Ethical Hacker: Michael Gregg, Pearson Education 2. Certified Ethical Hacker: Matt Walker, TMH.
<b>12</b>	<b>Internal Continuous Assessment: 40%</b>   <b>Semester End Examination: 60%</b>

**Sem. – VI**

**Sem. - VI**  
**Vertical – 1**  
**Major**  
**Mandatory**  
**(2+2+2+2+2)**

# Syllabus

## B.Sc. (Computer Science)

### (Sem.- VI)

Name of the Course: Data Science

Sr. No.	Heading	Particulars
1	Description the course:	<p><b>Introduction:</b></p> <p>The Data Science course introduces students to the end-to-end process of extracting meaningful insights from data. It covers data collection, preprocessing, exploratory analysis, statistical inference, machine learning algorithms, model evaluation, visualization, data management, and ethical considerations. The course builds a structured understanding of how data-driven decision systems are designed and implemented.</p> <p><b>Relevance:</b></p> <p>Data Science is central to modern decision-making across industries including finance, healthcare, retail, governance, and technology. This course is highly relevant as it integrates statistics, machine learning, visualization, and data governance to prepare students for analytics-driven environments.</p> <p><b>Usefulness:</b></p> <p>The course equips students with practical skills in data cleaning, feature engineering, model building, evaluation metrics, cross-validation, and hyperparameter tuning. It strengthens analytical reasoning, statistical interpretation, and the ability to translate raw data into actionable insights.</p> <p><b>Application:</b></p> <p>Data Science principles are applied in customer segmentation, fraud detection, predictive maintenance, recommendation systems, demand forecasting, sentiment analysis, and risk modeling. Techniques such as regression, classification, clustering, ensemble learning, and ROC analysis are widely used in enterprise analytics and AI-powered platforms.</p> <p><b>Interest:</b></p> <p>The subject is engaging because it transforms raw data into meaningful patterns and predictive models. Hands-on exposure to visualization tools, hypothesis testing, feature engineering, and model comparison makes the</p>

		<p>learning experience both analytical and creative.</p> <p><b>Connection with Other Courses:</b></p> <p>This course complements Artificial Intelligence, Database Management Systems, Statistics, Machine Learning, Cloud Computing, and Data Visualization subjects. It integrates programming, statistical modeling, and computational thinking within a unified analytics framework.</p> <p><b>Demand in the Industry:</b></p> <p>There is strong demand for professionals skilled in data analytics, predictive modeling, and machine learning. Organizations across sectors require expertise in data preprocessing, model evaluation, visualization, ETL pipelines, and data governance to support strategic and operational decision-making.</p> <p><b>Job Prospects:</b></p> <p>Students can pursue roles such as data analyst, business intelligence analyst, junior data scientist, ML associate, or analytics consultant trainee. With advanced specialization, career paths may extend to data scientist, ML engineer, data engineer, or analytics lead.</p>
2	<b>Vertical:</b>	Major (Mandatory)
3	<b>Type:</b>	Theory
4	<b>Credits:</b>	2 credits
5	<b>Hours Allotted:</b>	30 Hours
6	<b>Marks Allotted:</b>	50 Marks
7	<p><b>Course Objectives (CO):</b></p> <p><b>CO 1.</b> To introduce foundational concepts and scope of Data Science and its relationship with AI, ML, and BI.</p> <p><b>CO 2.</b> To develop competency in data preprocessing, feature engineering, and exploratory data analysis.</p> <p><b>CO 3.</b> To provide understanding of supervised and unsupervised machine learning algorithms and optimization techniques.</p> <p><b>CO 4.</b> To enable effective model evaluation, validation, and hyperparameter tuning.</p> <p><b>CO 5.</b> To inculcate responsible data handling practices including ethics, privacy, and governance principles.</p>	
8	<p><b>Course Outcomes (OC):</b></p> <p>After successful completion of this course, students would be able to -</p> <p><b>OC 1.</b> Perform data preprocessing, transformation, feature selection, and exploratory data analysis using statistical techniques.</p>	

	<p><b>OC 2.</b> Build and evaluate machine learning models for classification, regression, and clustering tasks using appropriate metrics.</p> <p><b>OC 3.</b> Apply cross-validation and hyperparameter tuning techniques for model selection and performance optimization.</p> <p><b>OC 4.</b> Create and interpret effective visualizations to communicate analytical insights.</p> <p><b>OC 5.</b> Demonstrate awareness of data governance, privacy regulations (including GDPR and DPDP Act 2023), and ethical data practices.</p>
<b>9</b>	<p><b>Modules:</b></p> <p><b>Module 1 (15 hours):</b></p> <p><b>Introduction to Data Science:</b> Definition and scope of Data Science, Applications and domains of Data Science, Comparison with other fields like Business Intelligence (BI), Artificial Intelligence (AI), Machine Learning (ML), and Data Warehousing/Data Mining (DW-DM), Different types of data: structured, unstructured, semi-structured; Data sources: databases, files, APIs, web scraping, sensors, social media</p> <p><b>Data Preprocessing:</b> Data cleaning: handling missing values, outliers, duplicates; Data transformation: scaling, normalization, encoding categorical variables; Feature selection: selecting relevant features/columns; Data merging: combining multiple datasets; Data wrangling techniques: reshaping, pivoting, aggregating</p> <p><b>Exploratory Data Analysis (EDA):</b> Feature engineering: creating new features, handling time-series data; Dummification: converting categorical variables into binary indicators; Feature scaling: standardization, normalization; Data visualization techniques: histograms, scatter plots, box plots, etc.; Descriptive statistics: mean, median, mode, standard deviation, etc.; Hypothesis testing: t-tests, chi square tests, ANOVA, etc.</p> <p><b>Machine Learning Algorithms:</b> Supervised learning: classification and regression, Unsupervised learning: clustering and dimensionality reduction, Bias-variance tradeoff, underfitting, and overfitting, Simple linear regression, Multiple linear regression, Stepwise regression, Logistic regression for classification, Decision Trees and Random Forests, Support Vector Machines (SVM), Artificial Neural Networks (ANN), Ensemble Learning: Boosting and Bagging, K-Nearest Neighbors (K-NN), Gradient Descent for optimization.</p> <p><b>Module 2 (15 hours):</b></p> <p><b>Model Evaluation, Selection and Metrics:</b> Techniques for evaluating, model performance: accuracy, precision, recall, F1-score, Confusion, matrix and ROC – AUC analysis, evaluating models for imbalanced, datasets, Cross-validation: k-fold cross-validation, stratified cross- validation, Hyperparameter tuning and model selection</p> <p><b>Data Visualization and Communication:</b> Principles of effective data visualization, Types of visualizations: bar charts, line charts, scatter plots, etc. Data storytelling: communicating insights through visualizations</p>

	<p><b>Tools and Libraries:</b> Introduction to popular libraries and technologies, used in Data Science like Pandas, NumPy, Sci-kit Learn, etc. Visualization tools: matplotlib, seaborn, Tableau, etc.</p> <p><b>Data Management:</b> Introduction to data management activities, Data pipelines: data extraction, transformation, and loading (ETL), Data governance and data quality assurance, Data privacy and security considerations</p> <p><b>Data Ethics and Privacy:</b> Data lifecycle ethics (collection, use, reuse), Privacy risks and consent, Basic overview of GDPR and DPDP Act 2023, Responsible data handling practices</p>
<b>10</b>	<p><b>Text Books</b></p> <ol style="list-style-type: none"> <li>1. Data Science from Scratch First Principles with Python- Joel Grus O'reilly, 2nd Edition</li> <li>2. Advancing into Analytics From Excel to Python and R, George Mount, Oreilly, First Edition</li> <li>3. Introduction to Machine Learning with Python, Andreas C. Muller, Sarah Guido, Oreilly, First Edition</li> </ol>
<b>11</b>	<p><b>Reference Books</b></p> <ol style="list-style-type: none"> <li>1. Doing Data Science, Rachel Schutt and Cathy O'Neil, O'Reilly,2013</li> <li>2. Mastering Machine Learning with R, Cory Lesmeister, PACKT Publication,2015</li> <li>3. Hands-On Programming with R, Garrett Golemund,1st Edition, 2014</li> <li>4. An Introduction to Statistical Learning, James, G., Witten, D., Hastie, T., Tibshirani, R.,Springer,2015</li> </ol>
<b>12</b>	<p><b>Internal Continuous Assessment: 40%</b>      <b>Semester End Examination: 60%</b></p>

## Name of the Course: Cloud Computing

Sr. No.	Heading	Particulars
1	<b>Description the course:</b>	<p><b>Introduction:</b></p> <p>The Cloud Computing course introduces students to distributed computing concepts, web services (SOAP and REST), virtualization technologies, and modern cloud architectures. It covers essential cloud characteristics, service models, deployment strategies, security fundamentals, simulation using CloudSim, and practical exposure to platforms such as OpenStack and AWS.</p> <p><b>Relevance:</b></p> <p>Cloud computing is a foundational technology powering modern digital infrastructure, enterprise applications, and scalable services. This course is highly relevant as organizations increasingly migrate workloads to cloud platforms and require professionals skilled in virtualization, cloud security, orchestration, and infrastructure management.</p> <p><b>Usefulness:</b></p> <p>The course equips students with knowledge of web service protocols (WSDL, SOAP, REST), virtualization using KVM, cloud deployment models, and secure cloud software practices. It develops the ability to design, simulate, and manage cloud environments while understanding scalability, elasticity, and resource optimization.</p> <p><b>Application:</b></p> <p>Cloud computing principles are applied in SaaS platforms, enterprise application hosting, DevOps pipelines, scalable APIs, VPC-based architectures, private cloud deployment, and secure multi-tenant environments. Tools like CloudSim, OpenStack, and AWS are widely used in industry for infrastructure management and cloud-native application deployment.</p> <p><b>Interest:</b></p> <p>The course is engaging as it explores how large-scale systems are built and managed dynamically in distributed environments. Hands-on exposure to virtualization tools, OpenStack deployment, AWS architecture design, and cloud orchestration makes the subject practically immersive and aligned with real-</p>

		<p>world enterprise systems.</p> <p><b>Connection with Other Courses:</b></p> <p>This course complements Computer Networks, Operating Systems, Cyber Security, Software Engineering, Distributed Systems, and DevOps-related subjects. It integrates knowledge of virtualization, resource management, secure software development, and scalable architecture design.</p> <p><b>Demand in the Industry:</b></p> <p>Cloud expertise is in high demand across IT services, fintech, healthcare, e-commerce, and enterprise technology sectors. Skills in AWS architecture, private cloud deployment, virtualization management, cloud security policies, and orchestration are highly valued in modern IT infrastructure roles.</p> <p><b>Job Prospects:</b></p> <p>Students can pursue roles such as cloud engineer trainee, DevOps associate, cloud administrator, AWS associate, virtualization engineer, or infrastructure support engineer. With advanced experience, career paths may extend to cloud architect, site reliability engineer, or cloud security specialist.</p>
<b>2</b>	<b>Vertical:</b>	Major (Mandatory)
<b>3</b>	<b>Type:</b>	Theory
<b>4</b>	<b>Credits:</b>	2 credits
<b>5</b>	<b>Hours Allotted:</b>	30 Hours
<b>6</b>	<b>Marks Allotted:</b>	50 Marks
<b>7</b>	<p><b>Course Objectives (CO):</b></p> <p><b>CO 1.</b> To introduce foundational concepts of distributed systems, web services, and virtualization technologies.</p> <p><b>CO 2.</b> To develop understanding of cloud computing models, characteristics, and deployment architectures.</p> <p><b>CO 3.</b> To provide knowledge of cloud security fundamentals and secure cloud software practices.</p> <p><b>CO 4.</b> To familiarize students with cloud simulation tools and private cloud platforms such as OpenStack.</p> <p><b>CO 5.</b> To enable architectural understanding of AWS-based cloud infrastructure solutions.</p>	
<b>8</b>	<p><b>Course Outcomes (OC):</b></p> <p>After successful completion of this course, students would be able to -</p>	

	<p><b>OC 1.</b> Explain and differentiate cloud service models, deployment models, and virtualization techniques.</p> <p><b>OC 2.</b> Design and implement web services using SOAP and REST principles in distributed environments.</p> <p><b>OC 3.</b> Configure and analyze virtualized and cloud environments using tools such as KVM and OpenStack.</p> <p><b>OC 4.</b> Evaluate cloud security requirements and implement secure development and policy practices.</p> <p><b>OC 5.</b> Demonstrate architectural understanding of AWS services and design scalable cloud-based solutions.</p>
9	<p><b>Modules:</b></p> <p><b>Module 1 (15 hours):</b></p> <p><b>Cloud Computing Basics:</b> Web Services – Distributed Computing, Parallel Computing, WSDL structure, SOAP- Structure of SOAP Message (In JAX-WS), SOAP Messaging Architecture, SOAP Header, Client-side SOAP Handler, REST- What is REST? HTTP methods, Java API for RESTful Web Services (JAX-RS)</p> <p><b>Virtualization:</b> Characteristics of Virtualized Environments, Pros and Cons of Virtualization. Virtualization using KVM, Creating virtual machines, oVirt - management tool for virtualization environment.</p> <p><b>Introduction to Cloud Computing:</b> Definition, Types of Clouds, Deployment of software solutions and web applications, Types of Cloud Platforms, Essential characteristics – On-demand self-service, Broad network access, Location independent resource pooling ,Rapid elasticity , Measured service, Comparing cloud providers with traditional IT service providers</p> <p><b>Module 2 (15 hours):</b></p> <p><b>Cloud Computing Software Security fundamentals:</b> Cloud Information Security Objectives, Confidentiality, Integrity, Availability, Cloud Security Services, Relevant Cloud Security Design Principles, Secure Cloud Software Requirements, Secure Development practices, Approaches to Cloud Software Requirement Engineering, Cloud Security Policy Implementation.</p> <p><b>CloudSim:</b> Introduction to Simulator, understanding CloudSim simulator, CloudSim Architecture(User code, CloudSim, GridSim, SimJava) Understanding Working platform for CloudSim,</p> <p><b>OpenStack:</b> Introduction to OpenStack, OpenStack test-drive, Basic OpenStack operations, OpenStack CLI and APIs, Tenant model operations, Quotas, Private cloud building blocks, Controller deployment, Networking deployment, Block Storage deployment, Compute deployment, deploying and utilizing OpenStack in production environments, Building a production environment, Application orchestration using OpenStack Heat</p> <p><b>AWS:</b> Architecting on AWS, Building complex solutions with Amazon Virtual Private Cloud (Amazon VPC)</p>

<b>10</b>	<p><b>Text Books</b></p> <ol style="list-style-type: none"> <li>1. Enterprise Cloud Computing Technology, Architecture, Applications, Gautam Shroff, Cambridge University Press, 2010</li> <li>2. Mastering Cloud Computing, Rajkumar Buyya, Christian Vecchiola, S Thamarai Selvi, Tata McGraw Hill Education Private Limited, 2013</li> <li>3. OpenStack in Action, V. K. CODY BUMGARDNER, Manning Publications Co, 2016</li> <li>4. Java Web Services Up and Running 2nd edition, Martin Kalin, O'Reilly (2013)</li> <li>5. Pro Power Shell for Amazon Web Services, Brian Beach, Apress, 2014</li> </ol>
<b>11</b>	<p><b>Reference Books</b></p> <ol style="list-style-type: none"> <li>1. OpenStack Essentials, Dan Radez, PACKT Publishing, 2015 2</li> <li>2. OpenStack Operations Guide, Tom Fifield, Diane Fleming, Anne Gentle, Lorin Hochstein, Jonathan Proulx, Everett Toews, and Joe Topjian, O'Reilly Media, Inc., 2014</li> <li>3. <a href="https://www.openstack.org">https://www.openstack.org</a></li> </ol>
<b>12</b>	<p><b>Internal Continuous Assessment: 40%      Semester End Examination: 60%</b></p>

## Name of the Course: Software Project Management

Sr. No.	Heading	Particulars
1	<b>Description the course:</b>	<p><b>Introduction:</b></p> <p>The Software Project Management course introduces students to structured methodologies and best practices for planning, executing, monitoring, and closing software projects. It covers project initiation, scope definition, time and cost estimation, quality assurance, risk management, resource coordination, agile methodologies, governance, and ethical considerations in technology-driven environments.</p> <p><b>Relevance:</b></p> <p>In the digital era, successful software delivery depends not only on technical expertise but also on effective project planning and execution. This course is highly relevant as organizations increasingly require professionals who can manage scope, time, cost, risk, and quality while adapting to agile and distributed development models.</p> <p><b>Usefulness:</b></p> <p>The course equips students with practical knowledge of Work Breakdown Structures (WBS), scheduling techniques, Earned Value Management (EVM), risk assessment, procurement planning, stakeholder management, and agile frameworks. It develops managerial, analytical, and decision-making skills necessary for leading software projects efficiently.</p> <p><b>Application:</b></p> <p>Software project management principles are applied in IT services, product development, startup ecosystems, enterprise digital transformation initiatives, and cloud-based solution deployment. Techniques such as schedule monitoring, cost control, vendor management, agile sprint planning, and stakeholder communication are widely used in industry.</p> <p><b>Interest:</b></p> <p>The subject is engaging as it blends technical planning with leadership, negotiation, and strategic decision-making. Topics like agile management, virtual team coordination, Six Sigma concepts, governance structures, and ethical responsibilities provide practical insights into managing real-world technology projects.</p>

		<p><b>Connection with Other Courses:</b></p> <p>This course complements Software Engineering, Cloud Computing, DevOps, Quality Assurance, and Enterprise Systems subjects. It integrates technical development processes with managerial frameworks, strengthening students' holistic understanding of software lifecycle management.</p> <p><b>Demand in the Industry:</b></p> <p>There is strong demand for professionals skilled in project coordination, agile methodologies, cost management, and stakeholder engagement. IT firms, consulting companies, fintech organizations, and startups require trained project managers and team leads to ensure timely and cost-effective software delivery.</p> <p><b>Job Prospects:</b></p> <p>Students can pursue roles such as project coordinator, associate project manager, scrum master trainee, business analyst, or operations executive. With experience, career paths may extend to project manager, program manager, agile coach, or IT governance specialist.</p>
2	<b>Vertical:</b>	Major (Mandatory)
3	<b>Type:</b>	Theory
4	<b>Credits:</b>	2 credits
5	<b>Hours Allotted:</b>	30 Hours
6	<b>Marks Allotted:</b>	50 Marks
7	<p><b>Course Objectives (CO):</b></p> <p><b>CO 1.</b> To introduce fundamental project management processes and knowledge areas in software development.</p> <p><b>CO 2.</b> To develop skills in scope definition, scheduling, cost estimation, and quality management.</p> <p><b>CO 3.</b> To enable effective risk assessment, procurement management, and stakeholder coordination.</p> <p><b>CO 4.</b> To provide understanding of agile methodologies and digital project management tools.</p> <p><b>CO 5.</b> To cultivate leadership, ethical responsibility, and governance practices in project environments.</p>	
8	<p><b>Course Outcomes (OC):</b></p> <p>After successful completion of this course, students would be able to -</p> <p><b>OC 1.</b> Plan and manage software projects using scope, time, cost, and quality management techniques.</p>	

	<p><b>OC 2.</b> Apply risk assessment, earned value analysis, and procurement strategies for effective project control.</p> <p><b>OC 3.</b> Implement agile methodologies and manage iterative development processes.</p> <p><b>OC 4.</b> Demonstrate leadership, stakeholder communication, and conflict resolution skills in project teams.</p> <p><b>OC 5.</b> Adhere to governance structures and ethical standards in professional project management practice.</p>
9	<p><b>Modules:</b></p> <p><b>Module 1 (15 hours):</b></p> <p><b>Introduction to Project Management:</b> Definition and characteristics of a project, Importance of project management, Project management processes and knowledge areas</p> <p><b>Project Selection, Initiation and scope Management:</b> Project selection criteria and methods, Project initiation and charter development, Stakeholder identification and analysis, Scope planning and definition, Work Breakdown Structure (WBS) development, Scope verification and change control</p> <p><b>Project Time &amp; Cost Management:</b> Activity definition and sequencing, Estimating activity durations and resources, Developing the project schedule, Schedule control and monitoring, Cost estimation techniques, Budget development and monitoring, Earned Value Management (EVM), Cost control and analysis</p> <p><b>Project Quality &amp; Risk Management:</b> Quality planning and standards, Quality assurance and control, Process improvement and Six Sigma concept, Risk Management-Risk identification and assessment, Risk response planning, Risk monitoring and control</p> <p><b>Module 2 (15 hours):</b></p> <p><b>Project Resource &amp; Procurement Management:</b> Team development and roles, Staffing, training, and motivation, Conflict resolution and communication management, Procurement planning and contract types, Solicitation, source selection, and contract administration, Vendor management and relationship building</p> <p><b>Project Integration Management:</b> Project integration processes and activities, Change management and project closure, Lessons learned and knowledge transfer</p> <p><b>Agile Project &amp; Management in the Digital Age:</b> Agile principles and methodologies, Agile project planning and execution, Managing iterative and incremental development, Role of technology in project management, Virtual teams and distributed project management, Tools and software for project planning and collaboration</p> <p><b>Effective People Management in Projects:</b> Leadership styles and characteristics, Team building and motivation techniques, Emotional intelligence in project management, Stakeholder identification and analysis, Stakeholder engagement and communication strategies, Conflict resolution and negotiation skills</p>

	<b>Project Governance and Ethics:</b> Project governance structures and accountability, Ethical considerations in project management, Professional responsibility and codes of conduct	
<b>10</b>	<b>Text Books</b> 1. Project Management for Business and Technology, 3rd edition, Pearson Education. John M. Nicholas, 2000 2. Information Technology Project Management, by Jack T. Marchewka, 4th Wiley India 2013. 3. A Guide to the Project Management Body of Knowledge (PMBOK® Guide)– Sixth Edition 6th Edition, Project Management Institute, 2017	
<b>11</b>	<b>Reference Books</b> 1. Introduction to Software Project Management by Adolfo Villafiorita · 2016, CRC press, e book format. 2. Project Management Professional Workbook, Claudia M. Baca, Patti M. Jansen, Sybex Publication, 2013 3. Project Management, by S. J. Mantel, J. R. Meredith and etal., 1st edition, Wiley India, 2009.	
<b>12</b>	<b>Internal Continuous Assessment: 40%</b>	<b>Semester End Examination: 60%</b>

## Name of the Course: Computer Science Practical 6

Sr. No.	Heading	Particulars
1	<b>Description the course:</b>	<p><b>Introduction:</b></p> <p>Computer Science Practical – 6 provides integrated hands-on experience in Data Science and Cloud Computing technologies. The course enables students to perform data analysis, statistical modeling, visualization, and machine learning experiments, while also developing and deploying web services, virtualization environments, and cloud-based solutions using industry-relevant platforms.</p> <p><b>Relevance:</b></p> <p>This course is highly relevant as it combines two major technological domains driving digital transformation—Data Science and Cloud Computing. The practical orientation ensures that students not only understand analytical and cloud concepts but also implement and deploy real-world solutions using contemporary tools and frameworks.</p> <p><b>Usefulness:</b></p> <p>The course equips students with applied skills in data preprocessing, hypothesis testing, regression modeling, clustering, PCA, and storytelling using visualization tools. Simultaneously, it develops competencies in web service design (SOAP/REST), API integration, virtualization (KVM), cloud deployment (OpenStack/AWS), and service orchestration, thereby enhancing technical versatility.</p> <p><b>Application:</b></p> <p>Data Science practicals apply to predictive analytics, classification modeling, clustering analysis, regression forecasting, and decision support systems. Cloud Computing practicals apply to distributed service deployment, API consumption, virtual infrastructure provisioning, REST/SOAP service design, and enterprise cloud environment configuration.</p> <p><b>Interest:</b></p> <p>The course is engaging as students move from analyzing structured datasets and performing hypothesis testing to deploying cross-platform web services and configuring cloud infrastructures. Activities such as Google Maps API integration, OpenStack network creation, AWS workflow execution, and MTOM-based multimedia transfer create an</p>

		<p>immersive and industry-aligned learning experience.</p> <p><b>Connection with Other Courses:</b></p> <p>This course complements Data Science, Artificial Intelligence, Cloud Computing, Database Management Systems, Software Engineering, and Computer Networks. It integrates analytical modeling with scalable infrastructure deployment, reinforcing interdisciplinary competence across the curriculum.</p> <p><b>Demand in the Industry:</b></p> <p>Industry strongly demands professionals skilled in data analytics, machine learning modeling, API development, virtualization, and cloud infrastructure management. Practical exposure to tools like Pandas, Scikit-Learn, AWS, OpenStack, KVM, and RESTful services enhances employability in data engineering, cloud engineering, and DevOps domains.</p> <p><b>Job Prospects:</b></p> <p>Students completing this course are better prepared for roles such as data analyst trainee, junior data scientist, cloud engineer associate, DevOps trainee, API developer, or infrastructure support engineer. The combination of analytics and cloud deployment skills provides a strong foundation for advanced specialization in AI, data engineering, and cloud architecture.</p>
<b>2</b>	<b>Vertical:</b>	Major (Mandatory)
<b>3</b>	<b>Type:</b>	Practical
<b>4</b>	<b>Credits:</b>	2 credits (1 credit = 30 Hours of Practical work in a semester)
<b>5</b>	<b>Hours Allotted:</b>	60 hours
<b>6</b>	<b>Marks Allotted:</b>	50 Marks
<b>7</b>	<p><b>Course Objectives (CO):</b></p> <p><b>CO 1.</b> To develop practical skills in data preprocessing, statistical analysis, and machine learning implementation.</p> <p><b>CO 2.</b> To enable hands-on experience in web service development using SOAP and REST architectures.</p> <p><b>CO 3.</b> To provide practical exposure to virtualization and cloud infrastructure setup using KVM, OpenStack, and AWS.</p> <p><b>CO 4.</b> To strengthen skills in data visualization, storytelling, and performance evaluation.</p> <p><b>CO 5.</b> To integrate analytical modeling with scalable cloud deployment practices.</p>	
<b>8</b>	<b>Course Outcomes (OC):</b>	

	<p>After successful completion of this course, students would be able to -</p> <p><b>OC 1.</b> Perform and interpret data preprocessing, hypothesis testing, regression, clustering, and PCA-based analysis using practical datasets.</p> <p><b>OC 2.</b> Develop and deploy SOAP and REST web services and integrate external APIs in client applications.</p> <p><b>OC 3.</b> Configure and manage virtual machines and cloud environments using KVM, OpenStack, and AWS-based tools.</p> <p><b>OC 4.</b> Evaluate and visualize analytical results and communicate insights effectively through structured storytelling.</p> <p><b>OC 5.</b> Integrate data science solutions with cloud-based deployment and infrastructure management practices.</p>
9	<p><b>Modules:</b></p> <p><b>Module 1 (30 hours):</b></p> <p><b>Practical based on Data Science</b></p> <hr/> <p><b>Introduction to Excel</b></p> <ul style="list-style-type: none"> <li>• Perform conditional formatting on a dataset using various criteria.</li> <li>• Create a pivot table to analyze and summarize data.</li> <li>• Use VLOOKUP function to retrieve information from a different worksheet or table.</li> <li>• Perform what-if analysis using Goal Seek to determine input values for desired output.</li> </ul> <p><b>Data Frames and Basic Data Pre-processing</b></p> <ul style="list-style-type: none"> <li>• Read data from CSV and JSON files into a data frame.</li> <li>• Perform basic data pre-processing tasks such as handling missing values and outliers.</li> <li>• Manipulate and transform data using functions like filtering, sorting, and grouping.</li> </ul> <p><b>Feature Scaling and Dummification</b></p> <ul style="list-style-type: none"> <li>• Apply feature-scaling techniques like standardization and normalization to numerical features.</li> <li>• Perform feature dummification to convert categorical variables into numerical representations.</li> </ul> <p><b>Hypothesis Testing</b></p> <ul style="list-style-type: none"> <li>• Formulate null and alternative hypotheses for a given problem.</li> <li>• Conduct a hypothesis test using appropriate statistical tests (e.g., t-test, chi-square test).</li> <li>• Interpret the results and draw conclusions based on the test outcomes.</li> </ul> <p><b>ANOVA (Analysis of Variance)</b></p> <ul style="list-style-type: none"> <li>• Perform one-way ANOVA to compare means across multiple groups.</li> </ul>

- Conduct post-hoc tests to identify significant differences between group means.

### **Regression and Its Types**

- Implement simple linear regression using a dataset.
- Explore and interpret the regression model coefficients and goodness-of-fit measures.
- Extend the analysis to multiple linear regression and assess the impact of additional predictors.

### **Logistic Regression and Decision Tree**

- Build a logistic regression model to predict a binary outcome.
- Evaluate the model's performance using classification metrics (e.g., accuracy, precision, recall).
- Construct a decision tree model and interpret the decision rules for classification.

### **K-Means Clustering**

- Apply the K-Means algorithm to group similar data points into clusters.
- Determine the optimal number of clusters using elbow method or silhouette analysis.
- Visualize the clustering results and analyze the cluster characteristics.

### **Principal Component Analysis (PCA)**

- Perform PCA on a dataset to reduce dimensionality.
- Evaluate the explained variance and select the appropriate number of principal components.
- Visualize the data in the reduced-dimensional space.

### **Data Visualization and Storytelling**

- Create meaningful visualizations using data visualization tools
- Combine multiple visualizations to tell a compelling data story.
- Present the findings and insights in a clear and concise manner.

## **Module 2 (30 hours):**

### **Practical based on Cloud Computing**

#### **Web Service + Cross-Platform Client Calls**

Cross-Platform Currency Conversion Service (INR to USD) with Java and .NET Clients: Design and deploy a simple web service that accepts an amount in Indian Rupees (INR) and returns the equivalent value in US Dollars (USD) using a configurable exchange rate. Implement two client applications—one in Java and one in .NET—that invoke the service, pass parameters, receive the response, and display the converted value. Include input validation, basic error handling, and logs of requests/responses.

#### **SOAP Web Service**

Development and Deployment of a Simple SOAP-Based Web Service: Create a SOAP web service exposing at least two operations (e.g., convertCurrency(), getServiceStatus()), publish the WSDL, and test it using a SOAP client (SoapUI or programmatic client). Demonstrate request/response XML structure, handling invalid parameters, and service endpoint accessibility.

### **REST Web Service**

Design and Implementation of a Simple RESTful Service with CRUD APIs: Develop a REST service that exposes APIs using HTTP methods (GET/POST/PUT/DELETE) for a small resource (e.g., Student, Course, or Task). Ensure responses are in JSON, implement basic validations, HTTP status codes, and test using Postman/cURL. Document endpoints and sample payloads.

### **Consuming Google Search / Google Maps REST API:**

Integration of External REST API: Google Maps-Based Place Search and Geolocation Viewer: Build a client application that consumes Google Maps REST APIs (e.g., Geocoding / Places) to accept a user query (place/address) and display extracted results such as latitude, longitude, formatted address, and optionally show it on a map UI. Handle API key configuration, HTTP request building, JSON parsing, rate-limit/error responses, and present output in a user-friendly format.

### **KVM Virtualization Setup**

Installation and Configuration of KVM for Virtual Machine Provisioning: Install and configure KVM virtualization on a Linux host system. Verify hardware virtualization support, configure required packages (KVM, QEMU, libvirt, virt-manager), and create at least one VM with defined CPU/RAM/disk. Demonstrate VM lifecycle operations: start/stop/snapshot, and document networking mode (NAT/bridge) used.

### **Image/Video Upload-Download Using MTOM**

Binary Data Transfer using MTOM: Upload and Download of Multimedia Files: Implement a SOAP-based file transfer service using MTOM (Message Transmission Optimization Mechanism) to upload and download image/video files between client and server. Ensure efficient binary transfer (not base64-heavy payloads), validate file size/type, store files server-side, and verify successful retrieval with integrity (file opens correctly).

### **FOSS-Cloud VSI Functionality for IaaS + Storage**

Implementing FOSS-Cloud IaaS: Virtual Server Infrastructure (VSI) with Storage Provisioning: Create a mini IaaS-style setup using FOSS cloud tools where users can provision a virtual server instance with selected CPU/RAM/disk and attach storage (block/object as supported by the chosen tool). Demonstrate instance creation, storage creation, attachment/mounting, and basic monitoring/usage reporting.

### **FOSS-Cloud VSI Functionality for PaaS**

Implementing FOSS-Cloud PaaS: Deploying a Web Application on a Platform Service: Use an open-source PaaS (or PaaS-like layer) to deploy a sample application (e.g., simple web app/API). The platform should manage runtime, deployment, scaling configuration (manual/auto if possible), and service bindings (database/cache

	<p>optional). Demonstrate deploy, update (version change), view logs, and service health.</p> <p><b>AWS Flow Framework Workflow Application</b></p> <p>AWS Flow Framework: Hello-World Workflow with Activity and Worker Programs: Develop a workflow-based application using AWS Flow Framework where a workflow triggers an activity that prints “Hello World” to the console. Implement workflow and activity interfaces (contracts), their implementations, workflow coordination logic, and worker programs to host and execute them. Demonstrate workflow execution and tracking of workflow/activity invocation.</p> <p><b>OpenStack Setup: User Network + Private Network Creation</b></p> <p>OpenStack Networking Practical: Creating Tenant User Network and Private Network Topology: Set up (or use an existing) OpenStack environment and create a tenant configuration including user/tenant, private network, subnet, and optionally a router connected to an external network. Launch an instance in the private network, assign IP addressing, validate connectivity (ping/SSH if applicable), and document the network topology created.</p>
<b>10</b>	<p><b>Text Books</b></p> <ol style="list-style-type: none"> <li>1. Hands-On Programming with R, Garrett Grolemond, 1st Edition, 2014</li> <li>2. Mastering Machine Learning with R, Cory Lesmeister, PACKT Publication, 2015</li> <li>3. Java Web Services Up and Running 2nd edition, Martin Kalin, O’Reilly (2013)</li> <li>4. OpenStack Operations Guide, Tom Fifield, Diane Fleming, Anne Gentle, Lorin Hochstein, Jonathan Proulx, Everett Toews, and Joe Topjian, O’Reilly Media, Inc., 2014</li> </ol>
<b>11</b>	<p><b>Reference Books</b></p> <ol style="list-style-type: none"> <li>1. Introduction to Machine Learning with Python, Andreas C. Muller, Sarah Guido, O’Reilly, First Edition</li> <li>2. Mastering Cloud Computing, Rajkumar Buyya, Christian Vecchiola, S Thamarai Selvi, Tata McGraw Hill Education Private Limited, 2013</li> <li>3. OpenStack in Action, V. K. CODY BUMGARDNER, Manning Publications Co, 2016</li> </ol>
<b>12</b>	<p><b>Internal Continuous Assessment: 40%   Semester End Examination: 60%</b></p>

**Name of the Course: Mini Project – II** (*Research-Oriented Project with Implementation*)

Sr. No.	Heading	Particulars
1	<b>Description the course:</b>	<p><b>Introduction:</b></p> <p>Mini Project – II is a research-focused course designed to develop students’ analytical, experimental, and scholarly skills through systematic investigation and implementation. The course guides students from research problem identification and literature survey to methodology design, experimentation, result analysis, and research paper writing in IEEE format.</p> <p><b>Relevance:</b></p> <p>In the era of rapid technological advancement, the ability to conduct structured research and validate solutions experimentally is highly relevant. This course prepares students for higher studies, research roles, and innovation-driven industry environments by emphasizing scientific inquiry and evidence-based evaluation.</p> <p><b>Usefulness:</b></p> <p>The course equips students with competencies in literature review, research gap identification, dataset selection, experimental design, model implementation, performance evaluation, and academic writing. It strengthens critical thinking, comparative analysis, hypothesis validation, and documentation skills required in research and R&amp;D domains.</p> <p><b>Application:</b></p> <p>Research-oriented implementation can be applied in areas such as Artificial Intelligence, Data Science, Cyber Security, Cloud Computing, IoT, or Software Engineering. Students design prototypes, evaluate models against baselines, and generate measurable results, mirroring real-world research and innovation workflows.</p> <p><b>Interest:</b></p> <p>The course is intellectually stimulating as students explore emerging technologies, analyze prior research, design experimental frameworks, and validate findings through implementation. The opportunity to publish or present work in different formats enhances academic motivation and professional confidence.</p> <p><b>Connection with Other Courses:</b></p> <p>Mini Project – II integrates knowledge from Artificial</p>

		<p>Intelligence, Data Science, Cyber Security, Cloud Computing, Software Engineering, and Statistics. It builds upon earlier theoretical and practical courses, transforming technical knowledge into research-driven innovation and structured experimentation.</p> <p><b>Demand in the Industry:</b></p> <p>Industries engaged in AI, analytics, cybersecurity, cloud solutions, and advanced software development value professionals who can conduct experimentation, benchmark solutions, and interpret performance metrics. Research-oriented skills are especially valuable in R&amp;D divisions, product innovation teams, and technology consulting firms.</p> <p><b>Job Prospects:</b></p> <p>Completion of this course prepares students for roles such as research assistant, data scientist trainee, AI/ML engineer, cybersecurity analyst, or innovation associate. It also provides a strong foundation for postgraduate studies, research fellowships, and publication-oriented academic careers.</p>
<b>2</b>	<b>Vertical:</b>	Major (Mandatory)
<b>3</b>	<b>Type:</b>	Practical
<b>4</b>	<b>Credits:</b>	2 credits (1 credit = 30 Hours of Practical work in a semester)
<b>5</b>	<b>Hours Allotted:</b>	60 hours
<b>6</b>	<b>Marks Allotted:</b>	50 Marks
<b>7</b>	<p><b>Course Objectives (CO):</b></p> <p><b>CO 1.</b> To develop skills in research problem identification and structured literature review.</p> <p><b>CO 2.</b> To enable design of experimental methodology and selection of appropriate models and tools.</p> <p><b>CO 3.</b> To provide hands-on experience in implementing and evaluating research prototypes.</p> <p><b>CO 4.</b> To cultivate analytical skills for result interpretation and hypothesis validation.</p> <p><b>CO 5.</b> To train students in professional research documentation and IEEE-format paper writing.</p>	
<b>8</b>	<p><b>Course Outcomes (OC):</b></p> <p>After successful completion of this course, students would be able to -</p> <p><b>OC 1.</b> Formulate and justify a research problem supported by literature review and gap analysis.</p>	

	<p><b>OC 2.</b> Design and implement an experimental framework including dataset selection, model development, and evaluation metrics.</p> <p><b>OC 3.</b> Conduct and analyze experiments using appropriate performance comparisons and baseline studies.</p> <p><b>OC 4.</b> Interpret results critically, identify limitations, and propose future research directions.</p> <p><b>OC 5.</b> Prepare and present a structured research paper and working prototype following IEEE standards.</p>
<b>9</b>	<p><b>Modules:</b></p> <p><b>Module 1 (30 hours):</b></p> <p><b>Research Planning, Literature Survey &amp; Methodology Design Phase</b></p> <p><b>Research Problem Identification:</b> Selection of topic, Problem statement formulation, Research objectives, Hypothesis (if applicable)</p> <p><b>Literature Review:</b> Study of minimum 5–8 research papers, Identification of research gap, Comparative analysis, Summary of existing techniques</p> <p><b>Research Methodology Design:</b> Dataset selection or data collection, Algorithm/model selection, Tool selection, Experimental setup planning, Definition of evaluation metrics</p> <p><b>Research Design Documentation:</b> Research Proposal, Literature Review Matrix, System Model / Architecture Diagram, Experimental Design Plan</p> <p><b>Module 2 (30 hours):</b></p> <p><b>Implementation, Experimentation &amp; Evaluation Phase</b></p> <p><b>System Implementation:</b> Model building, Algorithm implementation, Experimental setup, Code optimization</p> <p><b>Experimental Evaluation:</b> Model training/testing, Performance metric calculation, Baseline comparison</p> <p><b>Result Analysis:</b> Graphical representation, Interpretation of results, Validation of hypothesis, Identification of limitations</p> <p><b>Research Paper Writing:</b> Abstract, Introduction, Related Work, Methodology, Results, Conclusion &amp; Future Scope</p> <p><b>Final Deliverables</b></p> <ul style="list-style-type: none"> <li>• Working prototype</li> <li>• Research Paper (IEEE format, 6–8 pages minimum)</li> <li>• GitHub Repository</li> <li>• Presentation</li> </ul>
<b>10</b>	<p><b>Internal Continuous Assessment: 40%</b>      <b>Semester End Examination: 60%</b></p>

## Name of the Course: Information Retrieval

Sr. No.	Heading	Particulars
1	<b>Description the course:</b>	<p><b>Introduction:</b></p> <p>The Information Retrieval (IR) course introduces students to the design and evaluation of systems that store, index, and retrieve relevant information from large document collections. It covers document representation, inverted indexing, retrieval models, query processing, spelling correction, text classification, clustering, web search architecture, link analysis, learning to rank, and advanced topics such as summarization and question answering.</p> <p><b>Relevance:</b></p> <p>Information Retrieval forms the backbone of search engines, digital libraries, e-commerce platforms, recommender systems, and enterprise search solutions. This course is highly relevant as data volumes continue to grow exponentially and efficient retrieval, ranking, and personalization techniques are essential for user-centric information access.</p> <p><b>Usefulness:</b></p> <p>The course equips students with technical understanding of indexing techniques, TF-IDF weighting, cosine similarity, probabilistic retrieval, evaluation metrics, and ranking algorithms. It develops analytical skills required to design scalable retrieval systems, evaluate performance rigorously, and improve search relevance.</p> <p><b>Application:</b></p> <p>IR principles are applied in web search engines, academic databases, document management systems, legal search platforms, recommendation systems, and multilingual search services. Techniques such as PageRank, HITS, clustering, text categorization, and learning-to-rank models are widely used in modern information access platforms.</p> <p><b>Interest:</b></p> <p>The subject is intellectually stimulating as it explores how large-scale search engines rank billions of documents within milliseconds. Topics like link analysis, near-duplicate detection, ranking algorithms, cross-lingual retrieval, and A/B testing provide insights into real-world search and recommendation systems.</p>

		<p><b>Connection with Other Courses:</b></p> <p>This course complements Data Structures and Algorithms (indexing and complexity), Artificial Intelligence (ranking and learning models), Data Science (evaluation metrics and analytics), Machine Learning (classification and ranking), and Natural Language Processing (text representation and summarization).</p> <p><b>Demand in the Industry:</b></p> <p>There is strong demand for professionals skilled in search technologies, ranking algorithms, recommendation systems, and multilingual information processing. Companies in search technology, e-commerce, fintech, social media, and content platforms require expertise in retrieval optimization and user behavior analysis.</p> <p><b>Job Prospects:</b></p> <p>Students can pursue roles such as search engineer, data analyst, NLP engineer trainee, information systems analyst, or recommendation system developer. Advanced specialization may lead to careers in search engine optimization (SEO analytics), machine learning for ranking, or research in information access systems.</p>
<b>2</b>	<b>Vertical:</b>	Major (Elective)
<b>3</b>	<b>Type:</b>	Theory
<b>4</b>	<b>Credits:</b>	2 credits
<b>5</b>	<b>Hours Allotted:</b>	30 Hours
<b>6</b>	<b>Marks Allotted:</b>	50 Marks
<b>7</b>	<p><b>Course Objectives (CO):</b></p> <p><b>CO 1.</b> To introduce fundamental concepts, architectures, and challenges of Information Retrieval systems.</p> <p><b>CO 2.</b> To develop understanding of document indexing, retrieval models, and ranking techniques.</p> <p><b>CO 3.</b> To enable evaluation of IR systems using standard performance metrics and experimental methodologies.</p> <p><b>CO 4.</b> To provide knowledge of web search technologies, link analysis, and learning-to-rank approaches.</p> <p><b>CO 5.</b> To familiarize students with advanced topics such as summarization, recommender systems, and multilingual retrieval.</p>	
<b>8</b>	<p><b>Course Outcomes (OC):</b></p> <p>After successful completion of this course, students would be able to -</p>	

	<p><b>OC 1.</b> Design and implement indexing and retrieval mechanisms using Boolean, vector space, and probabilistic models.</p> <p><b>OC 2.</b> Evaluate IR systems using precision, recall, F-measure, and advanced ranking metrics.</p> <p><b>OC 3.</b> Apply link analysis, clustering, text categorization, and learning-to-rank techniques for improved search relevance.</p> <p><b>OC 4.</b> Analyze web crawling strategies and near-duplicate detection methods in large-scale retrieval systems.</p> <p><b>OC 5.</b> Explain and assess advanced IR applications including summarization, question answering, recommender systems, and cross-lingual retrieval.</p>
<b>9</b>	<p><b>Modules:</b></p> <p><b>Module 1 (15 hours):</b></p> <p><b>Introduction to Information Retrieval (IR) systems:</b> Definition and goals of information retrieval, Components of an IR system, Challenges and applications of IR</p> <p><b>Document Indexing, Storage, and Compression:</b> Inverted index construction and compression techniques, Document representation and term weighting, Storage and retrieval of indexed documents</p> <p><b>Retrieval Models:</b> Boolean model: Boolean operators, query processing, Vector space model: TF-IDF, cosine similarity, query-document matching, Probabilistic model: Bayesian retrieval, relevance feedback</p> <p><b>Spelling Correction in IR Systems:</b> Challenges of spelling errors in queries and documents, Edit distance and string similarity measures, Techniques for spelling correction in IR systems</p> <p><b>Performance Evaluation:</b> Evaluation metrics: precision, recall, F-measure, average precision, Test collections and relevance judgments, Experimental design and significance testing</p> <p><b>Text Categorization and Filtering:</b> Text classification algorithms: Naive Bayes, Support Vector Machines, Feature selection and dimensionality reduction, Applications of text categorization and filtering</p> <p><b>Text Clustering for Information Retrieval:</b> Clustering techniques: K-means, hierarchical clustering, Evaluation of clustering results, Clustering for query expansion and result grouping</p> <p><b>Module 2 (15 hours):</b></p> <p><b>Web Information Retrieval:</b> Web search architecture and challenges, Crawling and indexing web pages, Link analysis and PageRank algorithm</p> <p><b>Link Analysis and its Role in IR Systems:</b> Web graph representation and link analysis algorithms, HITS and PageRank algorithms, Applications of link analysis in IR systems</p> <p><b>Crawling and Near-Duplicate Page Detection:</b> Web page crawling techniques: breadth-first, depth-first, focused crawling, Near-duplicate page detection</p>

	<p>algorithms, Handling dynamic web content during crawling</p> <p><b>Learning to Rank:</b> Algorithms and Techniques, Supervised learning for ranking: RankSVM, RankBoost, Pairwise and listwise learning to rank approaches Evaluation metrics for learning to rank</p> <p><b>Advanced Topics in IR:</b> Text Summarization: extractive and abstractive methods; Question Answering: approaches for finding precise answers; Recommender Systems: collaborative filtering, content-based filtering</p> <p><b>Cross-Lingual and Multilingual Retrieval:</b> Challenges and techniques for cross-lingual retrieval, Machine translation for IR, Multilingual document representations and query translation, Evaluation Techniques for IR Systems</p> <p><b>User-based evaluation:</b> user studies, surveys, Test collections and benchmarking, Online evaluation methods: A/B testing, interleaving experiments</p>
<b>10</b>	<p><b>Text Books</b></p> <ol style="list-style-type: none"> <li>1. Ricardo Baeza-Yates and Berthier Ribeiro-Neto, —Modern Information Retrieval: The Concepts and Technology behind Search, Second Edition, ACM Press Books</li> <li>2. C. Manning, P. Raghavan, and H. Schütze, —Introduction to Information Retrieval, Cambridge University Press</li> </ol>
<b>11</b>	<p><b>Reference Books</b></p> <ol style="list-style-type: none"> <li>1. Ricci, F, Rokach, L. Shapira, B. Kantor, —Recommender Systems HandbookI, First Edition.</li> <li>2. Bruce Croft, Donald Metzler, and Trevor Strohman, Search Engines: Information Retrieval in Practice, Pearson Education.</li> <li>3. Stefan Buttcher, Charlie Clarke, Gordon Cormack, Information Retrieval: Implementing and Evaluating Search Engines, MIT Press.</li> </ol>
<b>12</b>	<p><b>Internal Continuous Assessment: 40%</b>      <b>Semester End Examination: 60%</b></p>

## Name of the Course: Linux Server Administration

Sr. No.	Heading	Particulars
1	<b>Description the course:</b>	<p><b>Introduction:</b></p> <p>The Linux Server Administration course provides a comprehensive understanding of configuring, managing, and securing Linux-based server environments. It covers Linux architecture, package management, user and service administration, networking fundamentals, firewall configuration, and deployment of enterprise services such as DNS, FTP, Apache, MySQL, NFS, Samba, and VPN.</p> <p><b>Relevance:</b></p> <p>Linux servers power a significant portion of web infrastructure, cloud platforms, enterprise systems, and data centers. This course is highly relevant as it equips students with system-level administrative skills required to manage secure and scalable server environments in modern IT ecosystems.</p> <p><b>Usefulness:</b></p> <p>The course develops practical competencies in user management, file system administration, service control using systemctl, disk management, firewall configuration, and network troubleshooting. It also provides hands-on exposure to configuring internet and intranet services, strengthening real-world server management capabilities.</p> <p><b>Application:</b></p> <p>Linux server administration principles are applied in web hosting, enterprise application deployment, cloud infrastructure management, database hosting, mail server configuration, and secure remote access systems. Technologies such as Apache, MySQL, SSH, DNS, NFS, Samba, and VPN are widely used in industry and enterprise IT environments.</p> <p><b>Interest:</b></p> <p>The course is engaging as it exposes students to the inner workings of server environments, including service configuration, log monitoring, security hardening, and network management. Hands-on setup of services like LAMP stack, DHCP, and Kerberos provides a practical and immersive administrative experience.</p>

		<p><b>Connection with Other Courses:</b></p> <p>This course complements Operating Systems, Computer Networks, Cloud Computing, Cyber Security, and Database Management Systems. It integrates system-level concepts with networking, security, and application deployment practices across the computer science curriculum.</p> <p><b>Demand in the Industry:</b></p> <p>There is strong industry demand for Linux administrators in web hosting companies, cloud service providers, fintech firms, telecom organizations, and enterprise IT departments. Skills in server configuration, service deployment, firewall management, and system security are highly valued in DevOps and infrastructure roles.</p> <p><b>Job Prospects:</b></p> <p>Students can pursue roles such as Linux system administrator, server support engineer, DevOps associate, infrastructure support engineer, network administrator, or cloud operations trainee. Advanced specialization can lead to roles in site reliability engineering, cloud architecture, or enterprise system administration.</p>
2	<b>Vertical:</b>	Major (Elective)
3	<b>Type:</b>	Theory
4	<b>Credits:</b>	2 credits
5	<b>Hours Allotted:</b>	30 Hours
6	<b>Marks Allotted:</b>	50 Marks
7	<p><b>Course Objectives (CO):</b></p> <p><b>CO 1.</b> To introduce Linux server architecture, package management, and system-level administration concepts.</p> <p><b>CO 2.</b> To develop skills in user management, service control, file systems, and network configuration.</p> <p><b>CO 3.</b> To provide knowledge of firewall configuration and system security hardening practices.</p> <p><b>CO 4.</b> To enable configuration and management of internet and intranet services in enterprise environments.</p> <p><b>CO 5.</b> To prepare students for real-world Linux server deployment and maintenance tasks.</p>	
8	<p><b>Course Outcomes (OC):</b></p> <p>After successful completion of this course, students would be able to -</p>	

	<p><b>OC 1.</b> Configure and manage Linux server environments including users, services, file systems, and packages.</p> <p><b>OC 2.</b> Implement and troubleshoot networking and firewall configurations for secure server operations.</p> <p><b>OC 3.</b> Deploy and administer internet services such as DNS, Apache, FTP, SMTP, and SSH.</p> <p><b>OC 4.</b> Manage enterprise services including NFS, Samba, DHCP, MySQL, and LAMP-based applications.</p> <p><b>OC 5.</b> Apply system security and hardening principles to maintain reliable and secure Linux server infrastructures.</p>
9	<p><b>Modules:</b></p> <p><b>Module 1 (15 hours):</b></p> <p><b>Introduction to Linux Server Environment:</b> Technical summary of major Linux distributions (Debian, Ubuntu, RHEL, CentOS, Fedora), Linux architecture overview (Kernel, Shell, File System hierarchy), Package management systems (APT, YUM/DNF), Managing software repositories and updates</p> <p><b>Single-Host Administration:</b> User and group management (useradd, usermod, passwd, groupadd), File permissions and ownership, Boot process and system initialization, Booting and shutting down processes, Managing services using systemctl, Core system services and logging, Linux file systems (ext4, XFS, mounting, fstab), Disk management basics (fdisk, lsblk, df, du), Configuring and compiling the Linux kernel (basic process overview)</p> <p><b>Networking Fundamentals for Administrators:</b> TCP/IP model for system administrators, IP addressing, subnetting basics, Network configuration, Hostname resolution and DNS basics, Routing tables and basic troubleshooting,</p> <p><b>Linux Security and Firewall:</b> Linux firewall architecture (Netfilter framework), iptables / firewalld configuration basics, System hardening principles, User authentication mechanisms, Basic network security concepts, Log monitoring and intrusion detection overview</p> <p><b>Module 2 (15 hours):</b></p> <p><b>Internet Services Configuration:</b> Domain Name System (DNS) – concepts and configuration basics, File Transfer Protocol (FTP) server setup, Apache Web Server (HTTP/HTTPS configuration), Simple Mail Transfer Protocol (SMTP), Email services architecture</p> <p><b>Mail and Secure Remote Services:</b> Post Office Protocol (POP3), Internet Mail Access Protocol (IMAP), Secure Shell (SSH) configuration and hardening, Network Authentication Systems: Kerberos, LDAP, NIS; Security considerations in service deployment</p> <p><b>Intranet and Enterprise Services:</b> Network File System (NFS), Samba (Linux–Windows file sharing), Distributed File Systems (DFS), Dynamic Host Configuration Protocol (DHCP), File services management</p> <p><b>Database and Application Services:</b> MySQL server administration, LAMP stack</p>

	architecture (Linux, Apache, MySQL, PHP), Deploying LAMP applications, Email services configuration, Chat applications, Virtual Private Networking: concepts and configuration	
<b>10</b>	<b>Text Books</b>	
	1. Linux Administration: A Beginner's Guide, Wale Soyinka, Seventh Edition, McGraw-Hill Education, 2016	
	2. Ubuntu Server Guide, Ubuntu Documentation Team, 2016	
<b>11</b>	<b>Reference Books</b>	
	1. Mastering Ubuntu Server, Jay LaCroix, PACKT Publisher, 2016	
<b>12</b>	<b>Internal Continuous Assessment: 40%</b>	<b>Semester End Examination: 60%</b>

## Name of the Course: Information Retrieval Practical

Sr. No.	Heading	Particulars
1	<b>Description</b> <b>the</b> <b>course:</b>	<p><b>Introduction:</b></p> <p>The Information Retrieval Practical course provides hands-on implementation of core IR concepts using benchmark datasets such as the Cranfield Collection, 20 Newsgroups, BBC News dataset, SNAP web graphs, and LETOR. It covers text preprocessing, inverted and positional indexing, retrieval models, evaluation metrics, classification, clustering, web crawling, PageRank, and learning-to-rank techniques.</p> <p><b>Relevance:</b></p> <p>With search engines, recommender systems, and large-scale document repositories forming the backbone of digital platforms, practical IR implementation is highly relevant. This course enables students to move from theoretical retrieval models to measurable system performance using standard datasets and evaluation frameworks.</p> <p><b>Usefulness:</b></p> <p>The course equips students with technical skills in text preprocessing, TF-IDF computation, cosine similarity ranking, Boolean retrieval, precision-recall analysis, MAP computation, classification using Naïve Bayes and SVM, clustering quality evaluation, and ranking model assessment using NDCG. It strengthens algorithmic thinking and experimental evaluation competence.</p> <p><b>Application:</b></p> <p>Practical IR techniques are applied in web search engines, enterprise search systems, digital libraries, news aggregation platforms, recommendation engines, and multilingual retrieval systems. Skills such as inverted index construction, ranking optimization, clustering analysis, and link analysis are directly applicable in search technology and analytics domains.</p> <p><b>Interest:</b></p> <p>The course is intellectually stimulating as students build search engines from scratch, compare retrieval models, correct spelling errors dynamically, visualize precision-recall curves, implement PageRank, and train learning-to-rank models. Working with real-world datasets makes the experimentation realistic and analytically engaging.</p>

		<p><b>Connection with Other Courses:</b></p> <p>This course complements Information Retrieval (theory), Data Structures and Algorithms, Artificial Intelligence, Machine Learning, Natural Language Processing, and Data Science. It integrates text processing, statistical modeling, ranking evaluation, and graph analysis in a unified experimental framework.</p> <p><b>Demand in the Industry:</b></p> <p>There is strong demand for professionals skilled in search technologies, ranking optimization, document classification, and recommender systems. Companies in search technology, e-commerce, fintech, content platforms, and analytics firms require expertise in indexing, ranking models, clustering, and evaluation metrics.</p> <p><b>Job Prospects:</b></p> <p>Students can pursue roles such as search engineer, NLP engineer trainee, data analyst, machine learning associate, recommendation system developer, or information systems analyst. Advanced specialization may lead to careers in ranking optimization, large-scale search architecture, or research in information access systems.</p>
2	<b>Vertical:</b>	Major (Elective)
3	<b>Type:</b>	Practical
4	<b>Credits:</b>	2 credits (1 credit = 30 Hours of Practical work in a semester)
5	<b>Hours Allotted:</b>	60 hours
6	<b>Marks Allotted:</b>	50 Marks
7	<p><b>Course Objectives (CO):</b></p> <p><b>CO 1.</b> To develop practical skills in text preprocessing, indexing, and retrieval model implementation.</p> <p><b>CO 2.</b> To enable evaluation of IR systems using standard metrics such as Precision, Recall, MAP, and NDCG.</p> <p><b>CO 3.</b> To provide hands-on experience in text classification, clustering, and link analysis.</p> <p><b>CO 4.</b> To implement web crawling and ranking algorithms for search systems.</p> <p><b>CO 5.</b> To strengthen experimental analysis and comparative evaluation using benchmark datasets.</p>	
8	<p><b>Course Outcomes (OC):</b></p> <p>After successful completion of this course, students would be able to -</p> <p><b>OC 1.</b> Implement and evaluate inverted indexing, Boolean retrieval, vector space models, and TF-IDF-based ranking.</p>	

	<p><b>OC 2.</b> Compute and analyze IR performance metrics including Precision, Recall, F1, AP, MAP, and NDCG.</p> <p><b>OC 3.</b> Apply classification (Naïve Bayes, SVM) and clustering (K-Means, Hierarchical) techniques to text datasets.</p> <p><b>OC 4.</b> Develop web crawling mechanisms and implement PageRank on graph datasets.</p> <p><b>OC 5.</b> Design and assess learning-to-rank models and compare them with baseline retrieval approaches.</p>
9	<p><b>Modules:</b></p> <p><b>Module 1 (30 hours):</b></p> <hr/> <p><b>Text Preprocessing on Cranfield Collection</b></p> <p>Download the Cranfield document collection (1400 aeronautical abstracts).</p> <p>Perform: Tokenization, Stopword removal (NLTK stopwords list), Porter stemming, Compute vocabulary size before and after preprocessing, Report top 20 frequent terms</p> <p><b>Inverted Index Construction for Cranfield Corpus</b></p> <p>Using the preprocessed Cranfield corpus build an inverted index, Store: term → {docID, term frequency}, Compute document frequency for each term, Print posting list for terms: “flow”, “boundary”, “shock”</p> <p><b>Positional Index &amp; Phrase Query Processing</b></p> <p>Extend inverted index to store term positions. Process phrase queries: “boundary layer flow”, “supersonic shock wave”, Return matching document IDs.</p> <p><b>Boolean Retrieval using Cranfield Queries</b></p> <p>Using Cranfield query set to Implement Boolean retrieval, Process query: (boundary AND flow) AND NOT laminar, Return list of document IDs, Compare with relevance judgments file</p> <p><b>TF-IDF Weight Computation on Cranfield</b></p> <p>Compute TF, IDF, TF-IDF weights, Display top 10 highest weighted terms in Document ID 12 and Document ID 245. Dataset: Cranfield Collection</p> <p><b>Ranked Retrieval using Vector Space Model</b></p> <p>For Cranfield Query 1: Convert query and documents to TF-IDF vectors, Compute cosine similarity, Rank top 10 documents, Compare retrieved results with relevance judgments</p> <p><b>Evaluation of Boolean vs Vector Space Model</b></p> <p>For first 5 Cranfield queries: Retrieve top 10 results using Boolean model, Retrieve top 10 results using VSM, Compute Precision@10, Compare retrieval effectiveness</p> <p><b>Edit Distance Implementation for Query Correction</b></p> <p>Implement Levenshtein distance. Correct the following misspelled Cranfield query terms: “supersonc”, “aerodynamcs”, “turbulnce”. Suggest top 3 corrections from</p>

corpus vocabulary. Dataset: Cranfield vocabulary

### **Integrated Spelling Correction in IR System**

Modify retrieval system so that when user enters: supersonc boundary layr. System automatically: Suggests corrected query, Performs retrieval using corrected query, Dataset: Cranfield Collection

### **Precision, Recall and F1 on Cranfield Query 1**

Using Cranfield relevance judgments: Retrieve top 20 documents for Query 1

Compute Precision, Recall, F1-score, Dataset: Cranfield Collection + qrels

### **Module 2 (30 hours):**

#### **Average Precision and MAP**

For first 10 Cranfield queries: Compute Average Precision (AP) for each query, Compute Mean Average Precision (MAP), Compare with baseline Boolean model, Dataset: Cranfield Collection

#### **Precision-Recall Curve Visualization**

For Query 3 Compute precision at each recall level, Plot Precision–Recall curve, Compare Boolean vs VSM, Dataset: Cranfield Collection

#### **Naïve Bayes Classification using 20 Newsgroups**

Using 4 categories from 20 Newsgroups dataset: sci.space, comp.graphics, rec.sport.baseball, talk.politics.misc, Train Multinomial Naïve Bayes classifier and compute Accuracy, Confusion matrix, Macro F1-score, Dataset: 20 Newsgroups Dataset

#### **SVM Classification using 20 Newsgroups**

Using 20 Newsgroups dataset Train linear SVM, Compare performance with Naïve Bayes, Report Precision, Recall, F1, Dataset: 20 Newsgroups

#### **K-Means Clustering on BBC News Dataset**

Cluster BBC News dataset into 5 clusters. Analyze Top 10 terms per cluster, Compare clusters with actual categories, Dataset: BBC News Dataset

#### **Hierarchical Clustering on BBC Dataset**

Perform Agglomerative clustering. Generate dendrogram and Determine optimal number of clusters, Compare with K-Means, Dataset: BBC News Dataset

#### **Clustering Evaluation (Purity & Silhouette)**

For the above 2 practicals, Compute Cluster Purity, Silhouette Score For  $K = 3, 4, 5$ , Compare clustering quality., Dataset: BBC News Dataset

#### **Web Crawling & Link Analysis**

Web Crawling of a Website: Develop a crawler starting from a given URL, Crawl up to 50 pages, Extract page titles, Build inverted index, Respect robots.txt

#### **PageRank on Sample Web Graph**

Given web graph with 10 nodes Construct adjacency matrix, Implement iterative

	<p>PageRank, Use damping factor = 0.85, Show convergence after 20 iterations Dataset: SNAP small web graph dataset (Stanford Network Analysis Project)</p> <p><b>Learning to Rank using LETOR Dataset</b></p> <p>Using LETOR 4.0 dataset Train RankSVM model, Evaluate using NDCG@10, Compare with TF-IDF baseline, Dataset: LETOR 4.0</p>
<b>10</b>	<p><b>Text Books</b></p> <ol style="list-style-type: none"> <li>1. Ricardo Baeza-Yates and Berthier Ribeiro-Neto, —Modern Information Retrieval: The Concepts and Technology behind Search, Second Edition, ACM Press Books</li> <li>2. C. Manning, P. Raghavan, and H. Schütze, —Introduction to Information Retrieval, Cambridge University Press</li> </ol>
<b>11</b>	<p><b>Reference Books</b></p> <ol style="list-style-type: none"> <li>1. Ricci, F, Rokach, L. Shapira, B. Kantor, —Recommender Systems Handbook, First Edition.</li> <li>2. Bruce Croft, Donald Metzler, and Trevor Strohman, Search Engines: Information Retrieval in Practice, Pearson Education.</li> <li>3. Stefan Butcher, Charlie Clarke, Gordon Cormack, Information Retrieval: Implementing and Evaluating Search Engines, MIT Press.</li> </ol>
<b>12</b>	<p><b>Internal Continuous Assessment: 40%</b>      <b>Semester End Examination: 60%</b></p>

## Name of the Course: Linux Server Administration Practical

Sr. No.	Heading	Particulars
1	<b>Description the course:</b>	<p><b>Introduction:</b></p> <p>The Linux Server Administration Practical course provides hands-on experience in installing, configuring, securing, and managing Linux-based server environments. Students work with Ubuntu Server to implement user management, networking, firewall rules, core services, and enterprise-level server integration, simulating real-world system administration scenarios.</p> <p><b>Relevance:</b></p> <p>Linux servers form the backbone of web hosting, cloud platforms, enterprise IT systems, and network infrastructure. This course is highly relevant as it equips students with practical skills required to deploy and maintain secure, scalable, and service-oriented Linux environments in professional settings.</p> <p><b>Usefulness:</b></p> <p>The course develops applied competencies in user administration, static IP configuration, service control, firewall setup, SSH access, disk management, DNS/DHCP configuration, and system monitoring. It also strengthens understanding of enterprise services such as Apache, FTP, Mail, NFS, Samba, LDAP, NIS, and MySQL.</p> <p><b>Application:</b></p> <p>The practical skills gained are directly applicable in web hosting environments, enterprise IT networks, cloud infrastructure management, database hosting, centralized authentication systems, and integrated server deployments. Students learn to configure multi-service Linux ecosystems supporting both Linux and Windows clients.</p> <p><b>Interest:</b></p> <p>The course is engaging because students configure real services such as DNS, DHCP, Apache, MySQL, SSH, and firewall rules, and observe their integration in a live network environment. Building an enterprise-style integrated server infrastructure provides a comprehensive and industry-aligned experience.</p> <p><b>Connection with Other Courses:</b></p> <p>This course complements Linux Server Administration (theory), Operating Systems, Computer Networks, Cyber Security, Database Management Systems, and Cloud Computing. It integrates system-level configuration with</p>

		<p>networking, authentication, and application deployment concepts.</p> <p><b>Demand in the Industry:</b></p> <p>There is strong demand for professionals skilled in Linux server deployment, service integration, firewall configuration, and centralized authentication systems. Organizations across IT services, cloud hosting, fintech, telecom, and enterprise infrastructure rely heavily on Linux administrators.</p> <p><b>Job Prospects:</b></p> <p>Students can pursue roles such as Linux system administrator, server support engineer, infrastructure support associate, DevOps trainee, network administrator, or cloud operations engineer. Advanced expertise can lead to site reliability engineering, enterprise system administration, or cloud infrastructure architecture roles.</p>
<b>2</b>	<b>Vertical:</b>	Major (Elective)
<b>3</b>	<b>Type:</b>	Practical
<b>4</b>	<b>Credits:</b>	2 credits (1 credit = 30 Hours of Practical work in a semester)
<b>5</b>	<b>Hours Allotted:</b>	60 hours
<b>6</b>	<b>Marks Allotted:</b>	50 Marks
<b>7</b>	<p><b>Course Objectives (CO):</b></p> <p><b>CO 1.</b> To develop practical expertise in installing and configuring Linux server environments.</p> <p><b>CO 2.</b> To enable implementation of networking, firewall, and secure remote access services.</p> <p><b>CO 3.</b> To provide hands-on experience in deploying and managing enterprise-level network and application services.</p> <p><b>CO 4.</b> To strengthen skills in centralized authentication and file-sharing mechanisms.</p> <p><b>CO 5.</b> To integrate multiple services into a functional enterprise Linux server infrastructure.</p>	
<b>8</b>	<p><b>Course Outcomes (OC):</b></p> <p>After successful completion of this course, students would be able to -</p> <p><b>OC 1.</b> Install and configure Linux server environments with proper user, network, and service management.</p> <p><b>OC 2.</b> Implement and secure core services including SSH, DNS, DHCP, Apache, FTP, and Mail servers.</p> <p><b>OC 3.</b> Configure and manage enterprise services such as NFS, Samba, LDAP, NIS, and MySQL.</p>	

	<p><b>OC 4.</b> Apply firewall rules and system monitoring techniques to maintain secure server operations.</p> <p><b>OC 5.</b> Design and deploy an integrated Linux server infrastructure supporting multi-service enterprise environments.</p>
<p><b>9</b></p>	<p><b>Modules:</b></p> <p><b>Module 1 (30 hours):</b></p> <p><b>Initial Server Setup and User Administration</b></p> <p>Install Ubuntu 16.04 Server and perform initial configuration including creation of new users, assigning sudo privileges, managing groups, configuring password policies, and verifying user authentication mechanisms.</p> <p><b>Static IP Configuration and Network Setup</b></p> <p>Configure a static IP address on Ubuntu Server by modifying network configuration files. Verify connectivity using network utilities and disable IPv6 if not required.</p> <p><b>Service Management and Boot Configuration</b></p> <p>Display currently running services, enable and disable services at boot time, stop unnecessary services, and analyze service logs to understand system-level service control.</p> <p><b>Linux Firewall Configuration using UFW and iptables</b></p> <p>Configure host-based firewall rules to allow or block services such as SSH, HTTP, DNS, and restrict specific IP ranges using UFW and iptables.</p> <p><b>SSH Server Installation and Secure Remote Access</b></p> <p>Install and configure OpenSSH server to allow password-based remote login. Modify SSH configuration settings and test remote login from Ubuntu and Windows clients.</p> <p><b>NTP Server and Client Configuration</b></p> <p>Install and configure NTP daemon to synchronize system time. Configure Ubuntu as NTP server and configure Ubuntu/Windows clients to synchronize with it.</p> <p><b>Disk Partitioning and File System Management</b></p> <p>Create disk partitions, format them with appropriate file systems, mount partitions, configure persistent mounting, and analyze disk usage.</p> <p><b>Process Management and System Monitoring</b></p> <p>Monitor system processes, change process priorities, terminate unwanted processes, and analyze CPU and memory usage using system monitoring tools.</p> <p><b>DNS Server Installation and Configuration (BIND9)</b></p> <p>Install BIND9 DNS server and configure forward and reverse lookup zones. Test domain name resolution and restrict DNS query access to specific networks.</p> <p><b>DHCP Server Configuration</b></p> <p>Install and configure DHCP server to dynamically assign IP addresses within a specified range. Configure static IP mapping based on MAC addresses.</p>

	<p><b>Module 2 (30 hours):</b></p> <p><b>Integration of DNS and DHCP Services</b> Configure DHCP server to update DNS records dynamically. Verify seamless name resolution for dynamically assigned clients.</p> <p><b>Apache Web Server Installation and Virtual Hosting</b> Install Apache web server, deploy a test website, configure virtual hosts, and verify HTTP/HTTPS access from client systems.</p> <p><b>FTP Server Configuration (vsftpd)</b> Install and configure FTP server to allow authenticated users to upload and download files. Configure directory restrictions and security settings.</p> <p><b>Mail Server Configuration (SMTP, POP3, IMAP)</b> Install and configure Postfix and Dovecot to implement basic email services. Test sending and receiving emails within the local network.</p> <p><b>NFS Server and Client Configuration</b> Configure NFS server to export directories and allow Linux and Windows clients to access shared folders with appropriate permissions.</p> <p><b>Samba Server Configuration for Windows–Linux File Sharing</b> Install Samba server to share files between Linux and Windows systems. Configure user authentication and verify cross-platform access.</p> <p><b>LDAP Server Configuration and User Account Management</b> Install OpenLDAP server to centralize user authentication. Add user accounts and manage directory entries using phpLDAPadmin.</p> <p><b>NIS Server and Client Configuration</b> Configure NIS server to share user account information across the network and bind client machines to the NIS domain for centralized login.</p> <p><b>MySQL Server Installation and Web-based Administration</b> Install MySQL database server, create databases and users, configure secure access, and manage databases using phpMyAdmin via web browser.</p> <p><b>Enterprise Linux Server Deployment and Service Integration</b> Design and deploy an integrated Linux server infrastructure combining DNS, DHCP, SSH, LDAP/NIS, NFS/Samba, Apache, MySQL, and Firewall services. Validate functionality from Ubuntu and Windows clients.</p>	
10	<p><b>Text Books</b></p> <ol style="list-style-type: none"> <li>Linux Administration: A Beginner’s Guide, Wale Soyinka, Seventh Edition, McGraw-Hill Education, 2016</li> <li>Ubuntu Server Guide, Ubuntu Documentation Team, 2016</li> </ol>	
11	<p><b>Reference Books</b></p> <ol style="list-style-type: none"> <li>Mastering Ubuntu Server, Jay LaCroix, PACKT Publisher, 2016</li> </ol>	
12	<b>Internal Continuous Assessment: 40%</b>	<b>Semester End Examination: 60%</b>

## EVALUATION SCHEME

### A. Evaluation for Theory Courses (2 Credit Courses)

#### I. Internal Evaluation – 20 Marks

##### a. For Courses other than IKS in Computational Systems

Sr. No.	Component
1	<b>Class Tests</b> Class Test 1 on Module 1: 10 marks Class Test 2 on Module 2: 10 marks
	<b>Average of 2 Class Tests: 10 marks</b>
	2
<b>Total of 2 Assignments: 10 marks</b>	
<b>Total: 20 Marks</b>	

##### b. For IKS in Computational Systems course

Students shall undertake **ONE implementation** (*individual or group of two*) from the given list or any other relevant IKS topic. Each implementation must include: (i) problem statement in the form ‘IKS concept as CS concept’, (ii) formal specification of rules/algorithm, (iii) working code, (iv) minimum 10 test cases, and (v) a short analysis of correctness and limitations. Students will demonstrate the work through a live demo and viva.

1. Piṅgala’s Chandaḥśāstra as Binary Encoding and Combinatorial Generation
2. Pāṇini’s Aṣṭādhyāyī as a Rule-Based Grammar Engine
3. Nyāya Logic as an Inference Engine
4. Ayurvedic Classification as a Rule-Based Expert System
5. Arthaśāstra-Inspired Cryptography as Symmetric Cipher System
6. Meru-Prastāra as Pascal Triangle and Dynamic Programming Model
7. Padārtha Ontology (Nyāya) as Knowledge Representation Model
8. Śāstra Rule Precedence as Deterministic Finite Rewrite System

Each submission must include:

1. Title: “IKS Concept as CS Concept”
2. Problem Statement
3. Conceptual Mapping Table
4. Algorithm Specification (Pseudo-code)
5. Working Code
6. Minimum 10 Test Cases
7. Complexity & Limitations
8. Conclusion

## II. External Examination – 30 Marks

A Semester End Theory Examination of 1 hour duration for 30 marks as per the paper pattern given below:

**Total Marks: 30**

**Duration: 1 Hour**

Q. No.	Based On	Question Type	Marks
Q. 1	Module 1	Answer any 2 of the following: (Any 2 out of 4)	10
Q. 2	Module 2	Answer any 2 of the following: (Any 2 out of 4)	10
Q. 3	Module 1 & 2	Answer any 2 of the following: (Any 2 out of 4)	10

## B. Evaluation for Practical Courses (2 Credit Courses)

(other than Mini Project)

### I. Internal Evaluation – 20 Marks

Sr. No.	Component
1	Mid – Term Practical Examination 15 marks
2	Final Journal: 5 marks
	<b>Total: 20 Marks</b>

### II. External Examination – 30 Marks

A Semester End Practical Examination of 2 hours duration for 30 marks as per the paper pattern given below.

**Total Marks: 30**

**Duration: 2 Hours**

Question	Practical Question Based On	Marks
Q. 1	Module 1	15
Q. 2	Module 2	15

1. **Certified Journal** is **compulsory** for appearing at the time of Practical Exam
2. Minimum **80% practical** are required to be completed.

### **C. Evaluation for Mini Project (2 Credit Courses)**

<b>Mini Project – I</b>		
<b>Internal Evaluation</b> <i>(To be assessed by the Project Guide)</i>	<b>Component</b>	<b>Marks</b>
	Problem Identification & Project Proposal	5
	System Design (SRS, UML Diagrams, Architecture)	5
	Development Progress & Code Review	5
	Internal Presentation / Review	5
<b>External Evaluation</b> <i>(To be assessed by External Examiner)</i>	<b>Component</b>	<b>Marks</b>
	Working Application Demonstration	10
	Technical Design & Implementation	10
	Project Report Evaluation	5
	Viva Voce	5

<b>Mini Project – II</b>		
<b>Internal Evaluation</b> <i>(To be assessed by the Project Guide)</i>	<b>Component</b>	<b>Marks</b>
	Research Problem Identification	5
	Literature Review	5
	Methodology & Implementation Progress	5
	Internal Seminar / Review	5
<b>External Evaluation</b> <i>(To be assessed by External Examiner)</i>	<b>Component</b>	<b>Marks</b>
	Prototype / Experimental Implementation	10
	Result Analysis & Interpretation	10
	Research Report / Paper	5
	Viva Voce	5

## Letter Grades and Grade Points:

Semester GPA/ Programme CGPA Semester/ Programme	% of Marks	Alpha-Sign/ Letter Grade Result	Grading Point
9.00 - 10.00	90.0 – 100	O (Outstanding)	10
8.00 - < 9.00	80.0 - < 90.0	A+ (Excellent)	9
7.00 - < 8.00	70.0 - < 80.0	A (Very Good)	8
6.00 - < 7.00	60.0 - < 70.0	B+ (Good)	7
5.50 - < 6.00	55.0 - < 60.0	B (Above Average)	6
5.00 - < 5.50	50.0 - < 55.0	C (Average)	5
4.00 - < 5.00	40.0 - < 50.0	P (Pass)	4
Below 4.00	Below 40.0	F (Fail)	0
Ab (Absent)	----	Ab (Absent)	0

**Sd/-**  
**Sign of the BOS Chairman**  
**Dr. Jyotshna Dongardive**  
Ad-hoc BOS (Computer Science)

**Sd/-**  
**Sign of the Offg. Associate Dean**  
**Dr. Madhav R. Rajwade**  
Faculty of Science & Technology

**Sd/-**  
**Sign of Offg. Dean**  
**Prof. Shivram S. Garje**  
Faculty of Science & Technology