

Re-Accredited 'B++' 2.86 CGPA by NAAC

**VEER NARMAD SOUTH GUJARAT UNIVERSITY**

University Campus, Udhna-Magdalla Road, SURAT - 395 007, Gujarat, India.

**વીર નર્મદ દક્ષિણ ગુજરાત યુનિવર્સિટી**

યુનિવર્સિટી કેમ્પસ, ઉદ્ધના-મગદલ્લા રોડ, સુરત - ૩૯૫ ૦૦૭, ગુજરાત, ભારત.

Tel : +91 - 261 - 2227141 to 2227146, Toll Free : 1800 2333 011, Digital Helpline No. - 0261 2388888

E-mail : info@vnsgu.ac.in, Website : www.vnsgu.ac.in

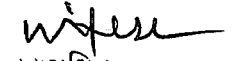
## **:- પરિપત્ર :-**

કોમ્પ્યુટર સાયન્સ વિદ્યાશાખા હેઠળની સંલગ્ન તમામ કોલેજોના આચાર્યશ્રીઓને જણાવવાનું કે, શૈક્ષણિક વર્ષ ૨૦૨૫-૨૬ થી અમલમાં આવનાર M.Sc. ( Computer Application) Sem.- 1 & 2 નો પેટાસમિતિ દ્વારા તૈયાર કરવામાં આવેલ અભ્યાસક્રમ કોમ્પ્યુટર સાયન્સ વિદ્યાશાખા તથા કોમ્પ્યુટર સાયન્સ વિષયની અભ્યાસ સમિતિની તા.૧૬/૦૬/૨૦૨૫ ની સંયુક્ત સભાના ઠરાવ ક્રમાંક:૦૮ થી મંજૂર કરી એકેડેમિક કાઉન્સિલને કરેલ ભલામણ એકેડેમિક કાઉન્સિલની તા.૨૪/૧૨/૨૦૨૪ ની સભાના ઠરાવ ક્રમાંક:૩૫૩ અન્વયે માનનીય કુલપતિશ્રીને આપેલ સત્તા અંતર્ગત એકેડેમિક કાઉન્સિલ વતી માન.કુલપતિશ્રીએ મંજૂર કરેલ છે. જેનો અમલ કરવા આથી જાણ કરવામાં આવે છે.

બિડાણ: ઉપર મુજબ

ક્રમાંક:ઓથો./પરિપત્ર/૧૫૧૮૭/૨૦૨૫

તા.૧૬-૦૬-૨૦૨૫

  
કુલસચિવ

પ્રતિ,

- ૧) યુનિવર્સિટી સંલગ્ન તમામ કોલેજોના આચાર્યશ્રીઓ.  
.....આપશ્રીની કોલેજ/વિભાગના સંબંધિત શિક્ષકો/વિદ્યાર્થીને જાણ કરી અમલ કરવા સારું.
- ૨) ઈ.યા.ડી.નશ્રી, કોમ્પ્યુટર સાયન્સ વિદ્યાશાખા.
- ૩) પરીક્ષા નિયામકશ્રી, પરીક્ષા વિભાગ, વીર નર્મદ દ. ગુ. યુનિવર્સિટી, સુરત.  
.....તરફ જાણ તેમજ અમલ સારું.

**M.Sc.(Computer Application)**  
**(Two Years Post Graduate Program)**  
**Sem- I and Sem-II Structure & Syllabus**

**(First Year M.Sc.(CA) )**

# M.Sc. (Computer Application)(Two year Program)

## Sem-I and II Structure & Syllabus:

**Aim:** To impart application oriented knowledge after completing three years traditional graduation degree course in the field of computers.

**Objective:** On completion of 3 years of traditional graduation program or four year of Honours, the student can avail specialization in the applied fields of computer science and application. The objective of the two years of master's program is to provide additional two years to enhance technically by imparting technology by means of application-oriented study. The important aspects of the program are : (i) Advance knowledge and skills, Specialization in the fields of latest and future technologies and computer applications in the fields of web development, mobile applications, artificial intelligence, data science, cybersecurity etc. (ii) To aligning the program with industry requirements and trends, ensuring that graduates are equipped with the necessary knowledge and skills to meet the demands of the rapidly evolving technology industry. (iii) To develop students ability to undertake independent research projects in computer applications. This includes promoting innovation, critical thinking, problem-solving skills, and the ability to apply theoretical concepts to practical scenarios. (iv) To align the program with industry requirements and trends, ensuring that graduates are equipped with the necessary knowledge and skills to meet the demands of the rapidly evolving technology industry. (v) To enhance students' professional skills, including communication, teamwork, project management, and leadership abilities. This enables them to effectively collaborate with colleagues, manage complex projects, and assume leadership roles in their careers. (vi) To emphasize the importance of ethical behavior and legal compliance in the field of computer applications. This includes raising awareness about privacy, security, intellectual property, and professional codes of conduct. (vii) To encourage entrepreneurial thinking among students and provide them with the necessary skills and knowledge to start their own ventures in the field of computer applications. This includes fostering creativity, business acumen, and an understanding of market dynamics. (viii) To instill a passion for continuous learning and self-improvement in students. The program aims to develop their ability to adapt to emerging technologies and industry trends throughout their professional lives. (ix) To provide students with a global perspective on computer applications, including exposure to international best practices, cultural diversity, and global industry standards. (x) To maintain high academic standards and promote excellence in teaching, research, and student performance. The program aims to produce graduates who are well-rounded, knowledgeable, and capable of making significant contributions to the field of computer applications.

<b>Name of Program:</b>	<b>Masters of Science in Computer Application (M.Sc.(Computer Application)) (Two Year Post Graduate Program)</b>
<b>Abbreviation:</b>	<b>M.Sc.(C.A.)</b>
<b>Duration:</b>	Two Year ( Four Semesters )
<b>Eligibility, Course Structure and Framework:</b>	As per the course structure University Circular No: S/Syllabus/Paripatr/17650/2023 Dated : 12/07/2022.
<b>Objective of the Program:</b>	<p>Masters of Science in Computer Application M.Sc.(C.A.) is post-graduate program in computer application area. Objective of the program is to impart advance level of computer technologies applicable at various application areas in the field of computers, Application developments in fields of Web Designing, Big data, Database Handling, Cloud computing, Data Analysis, software and information technology. Main objective is to equip the students with strong advance level of knowledge in the field of applied areas of various MVC, tools, coding, web technologies, database handling, analysis of database, managing and handling large-scale databases using various technologies, project development, deployment and testing, protection of data from various threats, problem-solving skills and development of analytical and logical skills. It also aim to provide knowledge in latest trends and advancements in field of computer technologies.</p> <p>The program caters to the needs of the students aspiring to excel in the field of computer science, application and technologies. The program is designed to develop computer professionals versatile in almost all field of computer application. It also aim to enhance communication and interpersonal skills.</p>
<b>Program Outcome:</b>	<p><b>PO1:</b> Ability to analyze a problem, identify and define the Computing requirements appropriate to its solution.</p> <p><b>PO2:</b> Foster creativity and innovation in students, encouraging them to develop novel solutions to real-world problem. Enhancing the problem solving, logical, reasoning and analysis capabilities of a problem and integrate the ability with the coding using specific computer programming languages.</p> <p><b>PO3:</b> Develop student’s ability to analyze, evaluate and solve complex problems in the field of computer applications, using critical thinking and problem-solving skills.</p> <p><b>PO4:</b> Design, implement and evaluate a computer-based system, processing, component or program to meet desired goal with the help of various programming languages, application software, packages, tools, databases, data analytics and representation of data on various platforms.</p> <p><b>PO5:</b> An ability to apply design and development principles in construction of software systems of varying complexity using various algorithmic principles, modeling, coding and design of computer-based systems.</p> <p><b>PO6:</b> Prepare the aspiring students to become computer professionals in applied areas who can work in corporate/software industry at entry to advanced level as well as independent developers.</p> <p>Overall, the program outcomes aim to produce graduates who are: (a) competent in computer application, development and design. (b) Adapt to changing technology and industry trends. (c) Can make significant contributions to the software applications coding, designing, database managements, testing, deployments and ready to adapt any upcoming technologies.</p>
<b>Program Specific Outcome:</b>	<p><b>PSO1:</b> Provide students with a strong understanding of programming languages, algorithms and data structures, necessary for software development. (PO-1 &amp; 2)</p> <p><b>PSO2:</b> Equip with skills in database management, data modeling and data analysis to develop efficient and effective data-driven software solutions.</p> <p><b>PSO3:</b> Develop expertise in field of software engineering, covering software design, testing and maintenance to ensure the production of high-quality software products.</p> <p><b>PSO4:</b> Develop knowledge and skills in web development, including web design, client-side and server-side programming and web security.</p> <p><b>PSO5:</b> Develop students to capabilities for self-learning, skill development through self-practicing and problem solving abilities.</p> <p><b>PSO6:</b> Develop students to address and work on the real-world problems as an individual and as part of team. Understand the business problems and ability to work on their solutions by applying various software technologies.</p> <p><b>PSO7:</b> To enhance development skills at various level including problem analysis, data analysis, logical and critical analysis of the problems and implementing the solutions by imparting various recent and upcoming technologies.</p> <p><b>PSO8:</b> Enhance the passion among the students for updating knowledge, innovative</p>

	ideas, upskilling and implementing the knowledge in applied areas and research areas by understanding the real world problems, addressing the real world problems and their possible solutions that lead to build a successful Professional career.								
<b>PO and PSO mapping:</b>		PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7	PSO8
	PO1								
	PO2								
	PO3								
	PO4								
	PO5								
	PO6								
<b>Medium of Instruction:</b>	English								
<b>Program Structure:</b>	Semester-wise Breakup of the course is given as follows :								
<b>Course Fees:</b>	As per Norms of University [Fees for any Additional Certificate courses(2-credit certification) in semester 1 and semester-2 will be paid separately by the students as per the university certificate courses norms. It is not part of the Tuition Fees.]								
<b>Internal/External Assessment:</b>	<ul style="list-style-type: none"> <li>- <b>Internal assessment:</b> Continuous assessment process (Assignment, Attendance, Class Test, Viva)</li> <li>- <b>External Assessment:</b></li> </ul> <p><b>For 4-credit Theory Courses :</b>  <b>Theory Exams:</b> University exam of 50 Marks (2 Hours duration).</p> <p><b>For 4-Credit Theory + Practical/project Courses courses :</b>  <b>Theory Exams :</b> University exam of 25 marks ( 1 Hours Duration)  <b>Practical Exams:</b> University exam of 25 marks ( 2 Hours duration)</p> <ul style="list-style-type: none"> <li>- <b>Project Exam (Course Code) :</b> Project exam will evaluate performance of the student.</li> <li>- <b>Project Evaluation:</b> Project presentation and viva-voce. The evaluation of the Practical/project will be carried out by the examiner panel of minimum three examiners consist of one local examiner, one expert from the Information technology/software Industry or from any other university except V.N.S.G.U. and one external examiner from any other institute affiliated to University having minimum eleven years of teaching experience at graduation level.</li> </ul>								

### Exam Pattern and Schedule:

- **Internal assessment:** Continuous assessment process (Assignment, Attendance, Class Test, Viva)
- **External Assessment:** At end of every module (course), University exams for 4 credit course:
  - (i) 4 credit course of Theory course : 50 marks external + 50 Marks internal
  - (ii) 4 Credit course of Theory+Practical : External: 25 marks Theory + 25 marks practical. Internal: 25 marks theory + 25 marks practical.
  - (iii) 2 Credit course : 50 marks External + 50 marks Internal.

Course Fees: As per the norms of the University.

Laboratory Utilization Fees: As per norms of University.

**Veer Narmad South Gujarat University, Surat**  
**Program Structure: First Year M.Sc. (C.A.)**  
**(SEM – I and SEM – II)**  
**(w.e.f. Academic Year June, 2025)**

Course Code	Course Title	Level of Course	Course Credits	Teaching Hours/week	
				Th.+Pra.	Theory
101	Version Control and Database Management	300-399	2	2	0
102-01 OR 102-02	Web Development and Design OR Fundamentals of Mobile Application Development	400-499	4	2	4
103-01 OR 103-02	Web Development Frameworks OR Event and media handling	400-499	4	2	4
104-01 OR 104-02	Web Development Operations OR Animation, Device and Components handling	400-499	4	2	4 (Project work – External)
105-01 OR 105-02	Automated Testing Framework OR Android API and Framework	400-499	4	2	4 (Project work – External)
106	Project	400-499	8	-	-
107	Seminar and Presentation	400-499	2	2	-
Total			28		

Course Code	Course Title	Course Credit	University Exam Type	Exam Duration	External Marks	Internal Marks	Total Marks
101	Version Control and Database Management	2	Theory/ Written	1 Hours	25	25	50
102-01 OR 102-02	Web Development and Design OR Fundamentals of Mobile Application Development	4	Theory/ Written  Practical/Project	1 Hours  2 Hours	25  25	25  25	100
103-01 OR 103-02	Web Development Frameworks OR Event and media handling	4	Theory/ Written  Practical/Project	1 Hours  2 Hours	25  25	25  25	100
104-01 OR 104-02	Web Development Operations OR Animation, Device and Components handling	4	Theory/ Written : Practical :	1 Hours 2 Hours	25 25	25 25	100
105-01 OR 105-02	Automated Testing Framework OR Android API and Framework	4	Theory/ Written : Practical:	1 Hours 2 Hours	25 25	25 25	100
106	Project	8	-	-	50	50	100
107	Seminar and Presentation	2	-	-	25	25	50
<b>Total</b>		<b>28</b>			<b>300</b>	<b>300</b>	<b>600</b>

# Course: 101: Version Control and Database Management

<b>Course Code</b>	<b>101</b>						
<b>Course Title</b>	<b>Version Control and Database Management</b>						
<b>Credit</b>	2						
<b>Minimum hours per Semester</b>	30 hours (Including class work, examination, preparation etc.)						
<b>Review / Revision</b>	June 2025						
<b>Pre-requisite</b>	Knowledge of RDBMS, Python, statistical methods.						
<b>Course outcome</b>							
<b>Mapping between Cos with PSOs</b>		PS01	PS02	PS03	PS04	PSO5	PS06
	CO1						
	CO2						
	CO3						
	CO4						
	CO5						
<b>Course Content</b>	<p><b>Unit-1: Modern Version Control and Git Essentials</b></p> <p>1.1 Concepts of Version Control System (VCS)</p> <p>    1.1.1 Purpose and importance of Version Control in software development</p> <p>    1.1.2 Types of VCS – Centralized vs Distributed</p> <p>    1.1.3 Advantages and real-world use cases of VCS in collaborative environments</p> <p>1.2 Introduction to Git</p> <p>    1.2.1 Installation and configuration of Git on Windows/Linux/macOS</p> <p>    1.2.2 Initializing a Git project and understanding Git directory structure</p> <p>    1.2.3 Git file states: Untracked, Modified, Staged, Committed</p> <p>1.3 Git Basic Operations</p> <p>    1.3.1 Creating repository, adding files, committing changes</p> <p>    1.3.2 Viewing status, logs, and differences</p> <p>    1.3.3 Undoing changes and working with .gitignore</p> <p><b>Unit-2: GitHub and Collaborative Workflows</b></p> <p>2.1 GitHub Concepts and Repository Hosting</p> <p>    2.1.1 Creating GitHub account and repositories</p> <p>    2.1.2 Pushing local repository to GitHub</p> <p>    2.1.3 Cloning repositories from GitHub</p> <p>2.2 Git Branching and Merging</p> <p>    2.2.1 Creating and switching branches</p> <p>    2.2.2 Merging branches and resolving conflicts</p> <p>    2.2.3 Understanding rebase, stash, and tags</p> <p>2.3 GitHub for Team Collaboration</p> <p>    2.3.1 Issues, pull requests, and code reviews</p> <p>    2.3.2 Project boards and GitHub Actions (intro to CI/CD)</p> <p>    2.3.3 GitHub Desktop and integration with IDEs (VS Code, JetBrains)</p> <p><b>Unit-3: Containerization Using Docker</b></p> <p>3.1 Concepts of Docker</p> <p>    3.1.1 Purpose and need for containerization in modern development</p> <p>    3.1.2 Installing and setting up Docker Desktop (Windows/macOS/Linux)</p> <p>    3.1.3 Docker Architecture and Terminologies</p> <p>        3.1.3.1 Images, Containers, Docker Daemon, CLI, Docker Hub</p>						

	<p>3.1.3.2 Docker commands: run, pull, ps, exec, stop</p> <p>3.2 Working with Docker Images and Containers</p> <p>3.2.1 Writing Dockerfiles and building custom images</p> <p>3.2.2 Managing volumes and networks</p> <p>3.2.3 Static site deployment using Docker</p> <p>3.3 Docker Compose and Publishing</p> <p>3.3.1 Introduction to Docker Compose (multi-container apps)</p> <p>3.3.2 Publishing images to Docker Hub</p> <p>3.3.3 Best practices for container management</p> <p><b>Unit-4: Data Warehousing and Integration Tools</b></p> <p>4.1 Concepts of Data Warehouse</p> <p>4.1.1 Definition, features, and types of data warehouses</p> <p>4.1.2 OLAP vs OLTP – comparative analysis</p> <p>4.1.3 Data Marts, Metadata, Metadata Repository, Data Cube</p> <p>4.2 ETL and Integration Tools</p> <p>4.2.1 Introduction to Extraction, Transformation, Loading (ETL)</p> <p>4.2.2 Data cleaning and transformation approaches</p> <p>4.2.3 Integration of heterogeneous databases</p> <p>4.2.3.1 Query-driven vs Update-driven approaches</p> <p>4.2.3.2 Tools: Overview of Talend/OpenRefine (demo-based)</p> <p>4.3 Data Warehouse Architecture and Process</p> <p>4.3.1 Data warehouse process flow: Extract, Clean, Load, Backup, Query</p> <p>4.3.2 3-tier architecture and virtual warehouse concepts</p> <p>4.3.3 Roles of Load Manager, Warehouse Manager, Query Manager</p> <p>4.3.4 Business Analysis Framework and real-world BI use cases</p>
<p><b>Reference Books</b></p>	<ol style="list-style-type: none"> <li>1. The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling, 3rd Edition, Ralph Kimball, Margy Ross , ISBN-13: 978- 1118530201, Wiley Inc.</li> <li>2. Database Systems: Introduction to Databases and Data Warehouses 1st Edition, Nenad Jukic, Susan Vrbsky, Svetlozar Nestorov, ISBN-13: 978-1943153190, Prospect Press</li> <li>3. Building a Scalable Data Warehouse with Data Vault 2.0 - 1st Edition, Daniel Linstedt, Michael Olschimke, ISBN-13: 978-0122025109</li> <li>4. Data Warehousing Fundamentals for IT Professionals 2nd Edition, Paulraj Ponniah, ISBN-13: 978-0410462072, Wiley Inc.</li> <li>5. The Kimball Wiley Inc.Group Reader: Relentlessly Practical Tools for Data Warehousing and Business Intelligence Remastered Collection 2nd Edition, ISBN-13: 978-1119216315, Wiley Inc.</li> <li>6. The Pragmatic Programmer: From Journeyman to Master 1st Edition, Andrew Hunt, David Thomas, ISBN-13: 978-0201616224</li> <li>7. Code Complete 2e (Developer Best Practices), Steve McConnell, ISBN- 13: 978-0735619678, Microsoft Press US</li> <li>8. The Docker Book, James Turnbull , Publisher: James Turnbull; 1809 2nd edition</li> <li>9. Docker in Action, 2nd Edition, Jeff Nickoloff, Stephen Kuenzli, ISBN- 13: 978-1617294761</li> <li>10. Learning Docker - Second Edition: Build, ship, and scale faster, Jeeva S. Chelladhurai, Vinod Singh, Pethuru Raj, ISBN-13: 978-1786462923</li> <li>11. Docker: Up &amp; Running, Karl Matthias, Sean P. Kane, ISBN-13: 978-1491917572</li> </ol>
<p><b>Teaching Methodology</b></p>	<p>Class Work, Discussion, Self-Study, Seminars and/or Assignments</p>
<p><b>Evaluation Method</b></p>	<p>50% Internal assessment. (Internal Unit tests, assignments and attendance)</p> <p>50% External assessment. (External Theory Examination)</p>



<b>Course Content</b>	<p><b>Unit 1: Core JavaScript for Web Development</b></p> <ul style="list-style-type: none"><li>1.1 Introduction to JavaScript (ES6+ syntax and concepts)</li><li>1.2 Working with the DOM: Document and Window objects</li><li>1.3 Variables, Data Types, and Operators</li><li>1.4 String and Math Operations</li><li>1.5 Objects, Arrays, and JSON</li><li>1.6 Date, Time, and Built-in Objects</li><li>1.7 Control Flow: if/else, switch-case</li><li>1.8 Iteration: for, while, for..Each</li><li>1.9 Functions and Arrow Functions</li></ul> <p><b>Unit 2: React Fundamentals and Component Design</b></p> <ul style="list-style-type: none"><li>2.1 Introduction to React and JSX</li><li>2.2 Functional vs Class Components (focus on Functional)</li><li>2.3 Props and State Management (useState)</li><li>2.4 Handling Events in React</li><li>2.5 Conditional Rendering and List Rendering</li><li>2.6 Component Lifecycle (with Hooks – useEffect)</li><li>2.7 React Forms and Controlled Components</li></ul> <p><b>Unit 3: Advanced React and Routing</b></p> <ul style="list-style-type: none"><li>3.1 React Router (v6+):<ul style="list-style-type: none"><li>3.1.1 Nested Routing, Navigation, Dynamic Routes</li></ul></li><li>3.2 Error Boundaries and Error Handling in React</li><li>3.3 Context API for Global State</li><li>3.4 Introduction to Redux Toolkit:<ul style="list-style-type: none"><li>3.4.1 State, Actions, Reducers</li><li>3.4.2 Redux Store Configuration</li></ul></li><li>3.5 Integrating REST APIs using fetch and Axios</li><li>3.6 React Portals and Fragments</li></ul> <p><b>Unit 4: Redux Integration, Middleware &amp; Project Work</b></p> <ul style="list-style-type: none"><li>4.1 Redux with React using React-Redux<ul style="list-style-type: none"><li>4.1.1 Connecting Components to Store</li><li>4.1.2 React-Redux Patterns</li></ul></li><li>4.2 Redux Middleware Concepts<ul style="list-style-type: none"><li>4.2.1 Logging, Crash Reporting, Routing</li><li>4.2.2 Thunks for Asynchronous Actions</li></ul></li><li>4.3 Redux Toolkit Middleware and Slices</li><li>4.4 Differences between React, React.js and React Native</li><li>4.5 Capstone Project: Build a React + Redux App (API integrated)<ul style="list-style-type: none"><li>4.5.1 Project Planning, UI Design, State Flow</li><li>4.5.2 Testing, Deployment using Vite or Create React App</li></ul></li></ul>
-----------------------	--

<b>Teaching Methodology</b>	Class Work, Discussion, Self-Study, Seminars and/or Assignments
<b>Evaluation Method</b>	<p>50% Internal assessment.          (Internal Unit tests, assignments and attendance)          (Practical and Project work assessment)</p> <p>50% External assessment.          (External Theory Examination)          (External Practical Examination)</p> <p>[students are required to acquire minimum passing marks in combination of Theory and Practical/Project Examination]</p>



<b>Course Content:</b>	<p><b>Unit 1: Introduction to Modern Angular (v17+)</b></p> <ul style="list-style-type: none"><li>1.1 Evolution from AngularJS to Angular – Key Differences</li><li>1.2 Features and Architecture of Angular</li><li>1.3 Setting up Angular Environment with Angular CLI</li><li>1.4 Angular Modules and Components</li><li>1.5 Data Binding: One-way, Two-way ( [ ( ngModel ) ] )</li><li>1.6 Directives: *ngIf, *ngFor, ngClass, ngStyle</li><li>1.7 User Inputs and DOM Events (click, mouse, key events)</li></ul> <p><b>Unit 2: Forms, Validation, and Services in Angular</b></p> <ul style="list-style-type: none"><li>2.1 Template-Driven and Reactive Forms</li><li>2.2 Form Controls: Input, Select, Radio, Textarea</li><li>2.3 Form Validation: Built-in and Custom<ul style="list-style-type: none"><li>2.3.1 Validating mobile numbers and whitespace</li></ul></li><li>2.4 Angular Pipes and Custom Pipes</li><li>2.5 Dependency Injection and Services</li><li>2.6 HTTP Client Module – Connecting to REST APIs</li><li>2.7 AJAX and JSON Data Handling in Angular</li></ul> <p><b>Unit 3: Advanced Angular Features</b></p> <ul style="list-style-type: none"><li>3.1 Angular Routing and Navigation<ul style="list-style-type: none"><li>3.1.1 Dynamic Routes, Route Guards</li></ul></li><li>3.2 Image Upload in Angular</li><li>3.3 Pagination using third-party directives (ngx-pagination)</li><li>3.4 Working with Observables (RxJS)</li><li>3.5 Responsive Design and Screen Handling</li><li>3.6 Capstone Task: Display JSON data using components and services</li></ul> <p><b>Unit 4: Backend Development using Express.js</b></p> <ul style="list-style-type: none"><li>4.1 Introduction to Express.js and Node.js<ul style="list-style-type: none"><li>4.1.1 Installing Express, Understanding Middleware</li></ul></li><li>4.2 Express Routing: Static and Dynamic Routes</li><li>4.3 HTTP Methods: GET, POST, PUT, DELETE</li><li>4.4 File Upload, Cookies, Session Handling</li><li>4.5 Template Engines and Basic Scaffolding</li><li>4.6 Integration: Angular Frontend with Express Backend (overview)</li></ul>
------------------------	---

<b>Reference Books</b>	<ol style="list-style-type: none"> <li>1. Web Development with Node and Express, Ethan Brown, O'Reilly Media, Inc., ISBN: 978-1-491-94930-6</li> <li>2. Node.js, MongoDB, React, React Native Full-Stack Fundamentals and Beyond, Eric Bush, Blue Sky Productions Inc., ISBN: 978-0-9971966-8-9</li> <li>3. Fullstack React: The Complete Guide to ReactJS and Friends, Anthony Accomazzo, Lean Publishing, Ari Learner, Clay Allsopp, David Guttman, Tyler McGinnis, Nate Murray,</li> <li>4. The Road to React: Your journey to master React.js in JavaScript, by Robin Wieruch</li> <li>5. Beginning React Native with Hooks, Greg Lim</li> <li>6. Full-Stack React Projects: Learn MERN stack development by building modern web apps using MongoDB, Express, React, and Node.js, 2nd Edition</li> <li>7. Angular from Theory to Practice, Asim Hussain, Version 1.2.0, 2017-11-24</li> <li>8. Angular: Up and Running: Learning Angular, Step by Step, Shyam Seshadri, O'Reilly Media, Inc.</li> <li>9. Mastering Web Application Development with AngularJS, Pawel Kozlowski Peter and Bacon Darwin, Packt Publishing</li> <li>10. The Complete Redux Book, Ilya Gelman and Boris Dinkevich, Lean Publishing</li> <li>11. Redux in Action, Marc Garreau and Will Faurot, ISBN 9781617294976</li> </ol>
<b>Teaching Methodology</b>	Class Work, Discussion, Self-Study, Seminars and/or Assignments
<b>Evaluation Method</b>	<p>50% Internal assessment.        (Internal Unit tests, assignments and attendance)        (Practical and Project work assessment)</p> <p>50% External assessment.        (External Theory Examination)        (External Practical Examination)</p> <p>[students are required to acquire minimum passing marks in combination of Theory and Practical/Project Examination]</p>

## Course: 104-01: Web Development Operations

<b>Course Code</b>	<b>104-01</b>						
<b>Course Title</b>	<b>Web Development Operations</b>						
<b>Credit</b>	4						
<b>Minimum hours per Semester</b>	30 Hours Theory + 60 Hours Practical/Project (Including class work, examination, preparation etc.)						
<b>Review / Revision</b>	June 2025						
<b>Pre-requisite</b>	Understanding about basics of Web Development Framework.						
<b>Course outcome</b>	<p><b>CO1:</b> <i>Explain</i> the principles of DevOps, its lifecycle, and how it integrates with Agile and traditional development models.</p> <p><b>CO2:</b> <i>Demonstrate</i> the use of key DevOps automation tools such as Maven, Jira, and Ansible for configuration and workflow automation.</p> <p><b>CO3:</b> <i>Create</i> Ansible Playbooks and Roles to automate deployment tasks and infrastructure management using YAML.</p> <p><b>CO4:</b> <i>Analyze and implement</i> control structures, variables, and exception handling within Ansible for complex web deployment scenarios.</p> <p><b>CO5:</b> <i>Build</i> a secure CI/CD pipeline using Jenkins to automate integration, testing, and deployment of web applications.</p>						
<b>Mapping between Cos with PSOs</b>		PS01	PS02	PS03	PS04	PSO5	PS06
	CO1						
	CO2						
	CO3						
	CO4						
	CO5						
<b>Course Content</b>	<p><b>Unit 1: Introduction to DevOps and Agile Integration</b></p> <p>1.1 Concepts of Development and Operations  1.2 Integration and Need for DevOps  1.3 DevOps Architecture and Lifecycle  1.4 Traditional vs. Modern Methodologies:      1.4.1 Waterfall vs. Agile      1.4.2 Agile vs. DevOps  1.5 DevOps Workflow in Web Development</p> <p><b>Unit 2: DevOps Tools Overview and Maven/Ansible Basics</b></p> <p>2.1 Overview of DevOps Automation Tools      2.1.1 Maven, Jira, Splunk, Ansible (purpose and roles)  2.2 Maven for Project Build Management  2.3 Introduction to Ansible      2.3.1 Installation and Configuration  2.4 YAML Basics:      2.4.1 Key-Value, Lists, Dictionaries      2.4.2 YAML Syntax in Ansible Tasks  2.5 File Transfer, Package Management using Ansible</p> <p><b>Unit 3: Automating with Ansible Playbooks and Roles</b></p> <p>3.1 Creating and Using Ansible Playbooks      3.1.1 Structure: name, hosts, vars, tasks  3.2 Managing Tasks using Tags and Variables  3.3 Creating Roles and Role Directory Structure  3.4 Breaking Playbook into Roles  3.5 Control Structures:</p>						

	<p>3.5.1 Blocks, Loops, Conditionals</p> <p>3.6 Exception Handling in Playbooks</p> <p><b>Unit 4: CI/CD with Jenkins and Pipeline Security</b></p> <p>4.1 Introduction to Jenkins</p> <p>    4.1.1 Architecture, Features, Installation</p> <p>4.2 Jenkins for Continuous Integration</p> <p>    4.2.1 Creating and Running Jobs</p> <p>4.3 CI/CD Pipeline Concepts</p> <p>    4.3.1 Continuous Integration</p> <p>    4.3.2 Continuous Delivery &amp; Deployment</p> <p>4.4 Pipeline Security Basics</p> <p>4.5 Building a CI/CD Pipeline using Jenkins</p>
<b>Reference Books</b>	<ol style="list-style-type: none"> <li>1. DevOps For Beginners, Joseph Joyner , Publisher: Mihails Konoplovs</li> <li>2. Practical Devops, Second Edition, Joakim Verona, Publisher: Ingram short title, ISBN-13: 978-1788392570</li> <li>3. DevOps For Beginners , Berg Craig, ISBN: 9798653362941</li> <li>4. The DevOps Handbook, Second Edition, Gene Kim, Jez Humble, Patrick Debois, John Willis, Nicole Forsgren</li> <li>5. Ansible: From Beginner to Pro 1st ed. Edition, Michael Heap , Apress Publications, ISBN-13: 978-1484216606</li> <li>6. Learning Ansible 2 - Second Edition, Fabio Alessandro Locati, Packt Publishing, ISBN-13: 978-1786464231</li> <li>7. Ansible Automation Platform, PETER SMITH, ISBN-13 979-8742550914</li> <li>8. Jenkins 2: Up and Running, Brent Laster , ISBN-13: 978-1491979594</li> <li>9. Continuous Delivery with Docker and Jenkins, 2nd Edition, Rafal Leszko, ISBN-13: 978-1838552183</li> <li>10. CI/CD Pipeline Using Jenkins Unleashed, Pranoday Dingare, ISBN-13: 978-1484275078</li> </ol>
<b>Teaching Methodology</b>	Class Work, Discussion, Self-Study, Seminars and/or Assignments
<b>Evaluation Method</b>	<p>50% Internal assessment. (Internal Unit tests, assignments and attendance) (Practical and Project work assessment)</p> <p>50% External assessment. (External Theory Examination) (External Practical Examination)</p> <p>[students are required to acquire minimum passing marks in combination of Theory and Practical/Project Examination]</p>

## Course: 105-01: Automated Testing Framework

<b>Course Code</b>	<b>105-01</b>							
<b>Course Title</b>	<b>Automated Testing Framework</b>							
<b>Credit</b>	4							
<b>Minimum hours per Semester</b>	30 Hours Theory + 60 Hours Applied/Project wok (Including class work, examination, preparation etc.)							
<b>Review / Revision</b>	June 2023							
<b>Pre-requisite</b>	Concepts of Web Development and Operations							
<b>Course outcome</b>	<p><b>CO1:</b> <i>Describe</i> the principles of automated testing, and differentiate between manual and automation tools and strategies.</p> <p><b>CO2:</b> <i>Apply</i> Selenium WebDriver using Python to automate real-time web applications including login, form submission, and navigation.</p> <p><b>CO3:</b> <i>Analyze</i> and <i>handle</i> dynamic elements, dropdowns, alerts, popups, and mouse events in modern web applications.</p> <p><b>CO4:</b> <i>Develop</i> end-to-end test scripts using Selenium-Python integrated with Excel sheets and database operations.</p> <p><b>CO5:</b> <i>Evaluate</i> synchronization techniques and implement advanced testing features like Ajax handling, listeners, and reporting.</p>							
<b>Mapping between Cos with PSOs</b>		PS01	PS02	PS03	PS04	PSO5	PS06	
	CO1							
	CO2							
	CO3							
	CO4							
	CO5							

<p><b>Course Content</b></p>	<p><b>Unit 1: Introduction to Automated Testing and Selenium Tools</b></p> <p>1.1 Concepts of Software Testing</p> <p>    1.1.1 Manual vs. Automated Testing – Pros &amp; Cons</p> <p>    1.1.2 Types of Tests Suitable for Automation</p> <p>1.2 Selenium Overview</p> <p>    1.2.1 Selenium IDE, RC, WebDriver, and Grid</p> <p>    1.2.2 Installation of Selenium IDE, FireBug, FirePath</p> <p>    1.2.3 Selenium Architecture</p> <p>    1.2.4 JSON Wire Protocol, Browser Drivers</p> <p>    1.2.5 Selenium Client Libraries</p> <p><b>Unit 2: Selenium WebDriver with Python</b></p> <p>2.1 Introduction to Selenium-Python</p> <p>    2.1.1 Installing Python-Selenium packages</p> <p>    2.1.2 Basic Web Navigation using <code>get()</code></p> <p>    2.1.3 Interacting with Web Elements</p> <p>2.2 Locating Web Elements</p> <p>    2.2.1 <code>find_element_by_id, name, xpath, tag_name, etc.</code></p> <p>    2.2.2 Dynamic XPath &amp; CSS Selectors</p> <p>2.3 Action Chains for Advanced User Interactions</p> <p>    2.3.1 Click, Click &amp; Hold, Double Click, Drag &amp; Drop</p> <p>    2.3.2 Key Events: Key Down/Up, Perform, Pause, Release</p> <p><b>Unit 3: Web Automation with Real-time Use Cases</b></p> <p>3.1 Selenium WebDriver with Different Browsers</p> <p>    3.1.1 Chrome, Firefox, Edge, IE Configuration</p> <p>3.2 Form Handling and Navigation</p> <p>    3.2.1 Login Page Automation (e.g., Facebook, Gmail)</p> <p>    3.2.2 Methods: <code>maximize_window(), get(), send_keys(), close()</code></p> <p>3.3 Handling UI Components</p> <p>    3.3.1 Drop-downs, Alerts, Popups, and Multi-window Handling</p> <p>3.4 Mouse and Keyboard Interactions</p> <p>    3.4.1 Mouse Hover, Right Click, Double Click, Drag and Drop</p> <p>3.5 Screenshot Capture</p> <p>    3.5.1 Element Screenshot vs. Full Page Screenshot</p> <p><b>Unit 4: Synchronization, Data Handling, and Backend Testing</b></p> <p>4.1 Selenium Waits</p> <p>    4.1.1 Implicit, Explicit, and Fluent Waits</p> <p>4.2 Apache POI for Excel File Integration</p> <p>    4.2.1 Read/Write Excel Data</p> <p>4.3 Database Testing</p> <p>    4.3.1 Using MySQL or DB2 with Selenium</p> <p>4.4 Handling Ajax Calls and Listeners</p> <p>    4.4.1 JavaScript Executor for Dynamic Content</p> <p>4.5 Test Reporting &amp; Debugging Techniques (Optional Intro to TestNG or PyTest)</p>
<p><b>Reference Books</b></p>	<p>1. The Art of Software Testing, 3rd Edition, Glenford J. Myers, Corey Sandler, Tom Badgett,</p> <p>2. Software Testing, 2nd Edition, 2005, Ron Patton, Sams Publishing, ISBN-13: 978-0672327988,</p> <p>3. Selenium with Python, Pallavi R Sharma, BPB Publication, ISBN-13: 978-9389328813</p>

	<p>4. Python Testing with Selenium, Sujay Raghavendra, ISBN-13: 978-1484262481</p> <p>5. Selenium WebDriver, Rajeev Gupta, ISBN-13: 978-9332526297</p> <p>6. Guide To Test Automation Using Selenium, Garg and Aditya, McGraw Hill, ISBN: 9781259005930</p> <p>7. Fundamentals Of Database Systems, Ramez Elmasri, ISBN:9788131716250</p>
<b>Teaching Methodology</b>	Class Work, Discussion, Self-Study, Seminars and/or Assignments
<b>Evaluation Method</b>	<p>50% Internal assessment.          (Internal Unit tests, assignments and attendance)          (Practical and Project work assessment)</p> <p>50% External assessment.          (External Theory Examination)          (External Practical Examination)</p> <p>[students are required to acquire minimum passing marks in combination of Theory and Practical/Project Examination]</p>

## Course : 106 : Project

Course Code	106
Course Title	Project
Credit	08
Duration/Hours:	Students will undergo an external Part-time project of 480 Hours during the semester including weekly reporting at institute, Presentation and report submission hours.
Minimum weeks per Semester	15 Weeks (Including Lab work, examination, preparation etc.)
Review / Revision	June – 2025
Purpose of Course	Hands on practice is essential for all application-oriented subjects. The courses relevant to web-design and development or mobile app developments can be learnt appropriately if the knowledge is applied in terms of applications. Various applications include portions relevant to web development from scratch till the testing and deployment and android based application developments and deployment. Project is based on Course102-01 to 105-01 in case the student has opted web development electives. In option to this, the students who selected elective as android based course, the project will be based on 102-02 to 105-02.
Course Content:	Students will work on Project based on any one of the following groups: Group-1:102-01, 103-01, 104-01 and 105-01 Courses. Group-2:102-02, 103-02, 104-02 and 105-02 Courses.
Teaching Methodology	Applied knowledge, External project work, Lab. Work, hands-on-experience, webinar, seminar, demonstrations, expert lectures.
Evaluation Method	50% Internal assessment. (Internal project reporting, presentation, viva-voce) 50% External assessment. (Project report (no-codes require), presentation and viva-voce.

## Course : 107 : Seminar

Course Code	107
Course Title	Seminar
Credit	2
Lab / Week	
Minimum weeks per Semester	15 Weeks (Including Lab work, examination, preparation etc.)
Review / Revision	June – 2023
Purpose of course	<ul style="list-style-type: none"><li>• During the semester, students will undergo the applied technology related to web design and development interactive app development or Mobile technology based application development and deployment. The syllabus covers various innovative technologies. To apply these technologies and enhance their acquired skills during semester; students will work on an in-house project.</li><li>• Students are expected to develop an interactive and dynamic web application or android based mobile application covering all technical skills learnt during the semester.</li><li>• Any open source database can be used for the purpose of project development. The project work will be in-house and continuous process since the commencement of the semester.</li><li>• At end of the semester, students will submit the project and project report.</li><li>• The internal and external evaluation will be based on developed app through viva-voce and presentation of the developed app.</li><li>• Students are expected to develop project individually.</li></ul>
Pre-requisite	Practical knowledge based on courses : 102, 103, 104 and 105 Courses.
Course outcome	CO1: Students will be able to understand the concepts of styles and theme CO2: Students will have Knowledge of testing Apps and publishing Apps CO3: Students will have knowledge about cross platform application development CO4: Students will have knowledge of various technologies covered during the semester

## Course: 102-02: Fundamentals of Mobile Application Development

Course Code	102-02						
Course Title	Fundamentals of Mobile Application Development						
Credit	4						
Minimum hours per Semester	30 Hours Theory + 60 Hours Practical						
Review / Revision	June – 2025						
Pre-requisite	Concepts of Mobile technology, Android studio, database and object oriented concepts are desirable.						
Course outcome	<p>CO1: Students will be able to understand the concepts of Android</p> <p>CO2: Students will be get knowledge of layout designing using different widgets and its installation.</p> <p>CO3: Students will have knowledge of activity and intent method to handle database.</p> <p>CO4: Students will be able to learn how to Implement mobile application with backend technology.</p>						
Mapping between Cos with PSOs		PS01	PS02	PS03	PS04	PSO5	PS06
	CO1						
	CO2						
	CO3						
	CO4						
Course Content	<p><b>Unit-1:</b></p> <p>1.1 Introduction of Android</p> <p>1.1.1 Architecture of Android</p> <p>1.1.2 Installation of Android and Android Emulator</p> <p>1.1.3 Dalvik VM</p> <p>1.2 Android Widgets:</p> <p>1.2.1 Button, Toast, Custom Toast</p> <p>1.2.2 CheckBox, RadioButton, AlertDialog</p> <p>1.2.3 Spinner, ListView, TextView</p> <p><b>Unit-2:</b></p> <p>2.1 Advance Widgets:</p> <p>2.1.1 WebView, RatingBar, TimePicker, DatePicker</p> <p>2.1.2 ProgressBar, Vertical and Horizontal ScrollView</p> <p>2.1.3 SeekBar, ImageSwitcher, ImageSlider</p> <p>2.2 Table and Search:</p> <p>2.2.1 TablLayout, FrameLayout</p> <p>2.2.2 SearchView, SearchView on Toolbar</p> <p>2.2.3 Text Exditing with TextWatcher</p> <p><b>Unit-3:</b></p> <p>3.1 Activity and Intents:</p> <p>3.1.1 LifeCycle, Implicit Intent, Explicit Intent</p> <p>3.1.2 StartActivityForResult</p> <p>3.1.3 Share App Data</p>						

	<p>3.2 Android Fragments</p> <p>3.3 Android Menu</p> <p>3.3.1 Option Menu</p> <p>3.3.2 Context Menu</p> <p>3.3.3 Popup Menu</p> <p><b>Unit-4:</b></p> <p>4.1 Database Handling:</p> <p>4.1.1 Android AlarmManager</p> <p>4.2 Read Data from Device Storage:</p> <p>4.2.1 From Internal memory and External memory</p> <p>4.2.1.1 FileInputStream, FileOutputStream</p> <p>4.2.1 Preferences</p> <p>4.3 Database handling: SQLite</p> <p>4.3.1 Database methods:</p> <p>4.3.1.1 SQLiteOpenHelper class (onCreate(), onUpgrade(), close() methods)</p> <p>4.3.1.2 Handling database ( create, open, drop, close)</p> <p>4.3.2 SQLiteDatabase class to handle SQLQuery: execSQL(), insert(), update(), query()</p> <p>4.3.2.1 Performing CRUD</p> <p>4.3.3 getWritableDatabase() method</p> <p>4.4 Handling Cursor</p> <p>4.4.1 fetching data in list view</p>
Reference Books	<ol style="list-style-type: none"> <li>1) Android Application Development (With Kitkat Support), Author: Pradeep Kothari, Publisher: DreamTech Press., ISBN: 978-9351194095</li> <li>2) Android Studio 3.0 Development Essentials: Android 8 Edition Author – Neil Smyth, Publisher: Payload Media, ISBN – 13: 978 – 1977540096</li> <li>3) Android Programming for Beginners - Second Edition, Author: John Horton, Publisher: Image Short ISBN: 978-1789538502</li> <li>4) Android 9 Development Cookbook, Author: Rick Boyer, Publisher: Packet Publishing, ISBN: 978-1788991216</li> <li>5) Professional Android – fourth Edition, Author: Reto Meier, Ian Lake, Publisher: Wrox, ISBN – 13: 978-1118949528</li> <li>6) Android Programming: Pushing the Limits 1st Edition, Author: Erik Hellman, Publisher: Wiley, ISBN – 13: 978-1118717370</li> <li>7) Fundamentals of Android App Development : Android Development for Beginners to Learn Android Technology, SQLite, Firebase and Unity, Author: Sujit Kumar Mishra, Publisher: BPB Publication, ISBN: 978-93-89845-204</li> <li>8) Starting with Android: Android application development guide 1st Edition, Author: Dr. M. M. Sharma, Publisher :BPB Publication, ISBN: 978-9386551955</li> <li>9) Introducing SQLite for Mobile Developers 1<sup>st</sup> Edition, Author: Jesse Feiler, Publisher: Apress, ISBN : 978-1484217658</li> <li>10) Android Programming Unleashed, Author: B. M. Harwani, Publisher: Pearson Education India, ISBN-13: 978-0-672-33628-7</li> </ol>
Teaching Methodology	Class Work, Discussion, Self-Study, Seminars and/or Assignments

Evaluation Method	50% Internal assessment. (Internal Unit tests, assignments and attendance) (Practical and Project work assessment) 50% External assessment. (External Theory Examination) (External Practical Examination)
----------------------	---

## Course: 103-02: Event and Media handling

Course Code	103-02						
Course Title	Event and Media handling						
Credit	4						
Minimum hours per Semester	30 Hours Theory + 60 Hours Practical/Project						
Review / Revision	June – 2025						
Pre-requisite	Knowledge of Fundamentals of Mobile Application Development						
Course outcome	<p>CO1: Students will be able to understand the concepts of event handling in Android</p> <p>CO2: Students will be get knowledge of XML and JSON and will learn method to implement the concept of XML and JSON in android application.</p> <p>CO3: Students will learn concept of Media handling.</p> <p>CO4: Students will learn and implement concept of Media Recording and text to speech conversion in andorid.</p>						
Mapping between Cos with PSOs		PS01	PS02	PS03	PS04	PSO5	PS06
	CO1						
	CO2						
	CO3						
	CO4						
Course Content	<p><b>Unit-1:</b></p> <p>1.1 Event Handling:</p> <p style="padding-left: 20px;">1.1.1 Input Events capturing</p> <p style="padding-left: 20px;">1.1.2 View Class and Event Listeners</p> <p>1.2 Callback methods of Event Listener interface:</p> <p style="padding-left: 20px;">1.2.1 onClick(), OnKey(), onTouch()</p> <p style="padding-left: 20px;">1.2.2 OnLongClick(), onFocusChange(), onCreateContextMenu()</p> <p>1.3 Event Listeners Registration</p> <p>1.4 Event Handlers:</p> <p style="padding-left: 20px;">1.4.1 onKeyDown(), onKeyUp(), onTrackballEvent()</p> <p style="padding-left: 20px;">1.4.2 onTouchEvent(), onFocusChanged()</p>						

**Unit-2:**

## 2.1 XML parsers:

- 2.1.1 Purpose of parsers
- 2.1.2 SAX (Simple API for XML) parser
- 2.1.3 Android DOM (Document object Model) parser
- 2.1.4 comparison among SAX and DOM

## 2.2 Structure of XML File

## 2.3 Parsing the XML file using SAX parser

## 2.4 Parsing the XML file using DOM parser

## 2.5 XMLPullParser and its methods to parse XML file

**Unit-3:**

## 3.1 JSON (Javascript Object Notation)

- 3.1.1 Comparing JSON with XML
- 3.1.2 Structure of JSON
- 3.1.3 JSON object, array

## 3.2 JSON parsing

- 3.2.1 JSONArray class

**Unit-4:**

## 4.1 Handling Audio Files:

## 4.1.1 MediaPlayer class

- 4.1.1.1 Methods to handle Audio files: (start(),stop(),pause(), prepare(), setDataSource(), prepare(), isPlaying())

## 4.2 Handling Video files:

## 4.2.1 MediaController and VideoView class

- 4.2.1.1 Methods to handle video files: (start(), pause(),suspend(), resume(), seekTo(), setMediaController(), setVideoURI())

## 4.3 Media Recording

## 4.3.1 Sound and videoRecording:

- 4.3.1.1 recording sound and video
- 4.3.1.2 Storing in external folder

## 4.4 Use of MediaRecorder class

- 4.4.1 Audio Manager class:
- 4.4.2 Controlling volume and ringer
- 4.4.3 setRingerMode() method
- 4.4.4 Ringer Modes (Normal,Silent,Vibrate)

## 4.5 Convert Text to Speech:

## 4.5.1 TextToSpeech class

- 4.5.1.1 Important Methods: speak(), setSpeechRate(),
- 4.5.1.2 setPitch(), setLanguage(), shutdown(), stop()

## 4.5.2 OnInitListener Interface:

- 4.5.2.1 onInit() method

## 4.5.3 Controlling the speech methods:

- 4.5.3.1 setSpeechRate(), setPitch()

Reference Books	<p>1) Android Application Development (With Kitkat Support), Author: Pradeep Kothari, Publisher: DreamTech Press., ISBN: 978-9351194095</p> <p>2) Android Studio 3.0 Development Essentials: Android 8 Edition Author – Neil Smyth, Publisher: Payload Media, ISBN – 13: 978 – 1977540096</p> <p>3) Android Programming for Beginners - Second Edition, Author: John Horton, Publisher: Image Short ISBN: 978-1789538502</p> <p>4) Android Studio new Media Fundamentals: Content Production of Digital Audio/Video, Illustration and 3D Animation, Author: Wallace Jackson, Publisher: Apress, ISBN – 13: 978-1484216408</p> <p>5) Android Programming: Pushing the Limits 1st Edition, Author: Erik Hellman, Publisher: Wiley, ISBN – 13: 978-1118717370</p> <p>6) Android 9 Development Cookbook, Author: Rick Boyer, Publisher: Packet Publishing, ISBN: 978-1788991216</p> <p>7) JSON Quick Syntax Reference, Author: Wallace Jackson, Publisher: Apress, ISBN: 9781484218631</p> <p>8) Beginning Json, Author: Ben Smith, Publisher: Apress, ISBN: 9781484240427</p> <p>9) Java XML and JSON 1st ed. Edition. Author : Jeff Friesen, Publisher: Apress, ISBN – 13: 978-1484219157</p> <p>10) Android Programming Unleashed, Author: B. M. Harwani, Publisher: Pearson Education India, ISBN-13: 978-0-672-33628-7</p> <p>11) Android Wireless Application Development: Advanced Topics ( Developer’s Library) 3<sup>rd</sup> Edition, Author: Shane Conder, Publisher: Addison Wesley, ISBN-13: 978-0321813848</p>
Teaching Methodology	Class Work, Discussion, Self-Study, Seminars and/or Assignments
Evaluation Method	<p>50% Internal assessment. (Internal Unit tests, assignments and attendance) (Practical and Project work assessment)</p> <p>50% External assessment. (External Theory Examination) (External Practical Examination)</p>

## Course: 104-02: Animation, Device and Components handling

Course Code	104-02						
Course Title	Animation, Device and Components handling Using Android						
Credit	4						
Minimum hours per Semester	30 hrs Theory + 60 Hours Practical/Project work						
Review / Revision	June – 2025						
Pre-requisite	Concepts of Event and Media handling using android.						
Course outcome	CO1: Students will be able to understand the concepts of animation and effects CO2: Students will have knowledge of Device and component handling CO3: Students will have knowledge of using Google MAP in android application CO4: Students will have knowledge of Android Telephony and Different sensors						
Mapping between COs with PSOs		PS01	PS02	PS03	PS04	PSO5	PS06
	CO1						
	CO2						
	CO3						
	CO4						
Course Content	<p><b>Unit-1:</b></p> <p>1.1 Animations and Effects:</p> <p>1.1.1 Load and Start animation</p> <p>1.1.2 Applying different animations:</p> <p>1.1.2.1 Zoom in-out, Fade in-out</p> <p>1.1.2.2 Slide up-down, Rotate in-Out</p> <p>1.1.2.3 Clockwise - Anti-Clockwise</p> <p>1.2 Intents:</p> <p>1.2.1 Concepts of Intents</p> <p>1.2.2 Starting Activity and Service, Delivering broadcast</p> <p>1.2.3 types of Intents: Implicit and Explicit</p> <p>1.3 Handling Device Camera:</p> <p>1.3.1 android.hardware.camera2 API</p> <p>1.3.2 camera intent</p> <p><b>Unit-2:</b></p> <p>2.1 Purpose of Bluetooth</p> <p>2.1.1 Bluetooth API</p> <p>2.1.2 Checking Internet connection</p> <p>2.2 Functionalities using Bluetooth API</p> <p>2.2.1 Scan for the available Bluetooth devices within the range</p> <p>2.2.2 Use local Bluetooth adapter for paired Bluetooth devices</p> <p>2.2.3 Connect to other devices through service discovery</p> <p>2.2.4 Transfer data to and from other devices using bluetooth</p> <p>2.2.5 Manage multiple connections</p> <p>2.3 Wi-Fi connectivity:</p> <p>2.3.1 Scan for the available Wi-Fi networks within the range</p> <p>2.3.2 Allow devices to connect to the internet</p> <p>2.3.3 Connect to other devices through service discovery</p> <p>2.3.4 Manage list of configured networks.</p>						

	<p>2.3.5 Manage multiple connections</p> <p><b>Unit-3:</b></p> <p>3.1 Drag and Drop Framework:</p> <p>3.1.1 Drag Event class</p> <p>3.1.2 Drag Listeners</p> <p>3.1.3 Helper methods and Class</p> <p>3.2 Drag and Drop Process: (Start, continuing, Dropped, Ended)</p> <p>3.3 Goolge Map android API:</p> <p>3.3.1 Installing and Using Google Map API in android device</p> <p>3.4 Sending Email through Android App</p> <p>3.5 Sending sms using SMSManager API</p> <p>3.6 SMS using Intent</p> <p><b>Unit-4:</b></p> <p>4.1 Andrioid telephony:</p> <p>4.1.1 android.telephony.TelephonyManager class for telephony service</p> <p>4.1.2 Determining call state</p> <p>4.1.3 Phone call by invoking default phone calls app using an Intent object</p> <p>4.2 Sensors : Motion Sensors, Enviornmental Sensors, Position sensors</p> <p>4.2.1 Sensor framework, frame work classes and interfaces</p> <p>4.2.2 SensorManager, Sensor, SensorEvent, SensorEventListener</p> <p>4.3 Types of maps</p> <p>4.4 Google Map API and its methods</p> <p>addCircle(), addPolygon, addTileOverlay()</p> <p>clear(), getMyLocation(), snapshot()</p> <p>4.5 Displaying Current location</p> <p>Callback methods</p> <p>4.6 Searching location on Google Map</p> <p>Using Geocoder class and its methods</p>
Reference Books	<ol style="list-style-type: none"> <li>1) Advanced Android™ Application Development, Fourth Edition, Author: Joseph Annizzi Jr., Lauren Darcey, Shane Conder, Publisher: Addison Wesley Professional, ISBN : 9720133892420</li> <li>2) Android 3.0 Animations : Beginner’s Guide, Author: Alex Shaw, Publisher: PACKT Publishing, ISBN: 978-1-84951-528-3</li> <li>3) Learning Android Google Map, Author: Raj Amal W., Publisher: Packt Publishing, ISBN: 9781849698863</li> <li>4) Android Sensor Programming By Example, Author: Varun Nagpal, Publisher: Packt Publication, ISBN-13: 978- 1785285509</li> <li>5) Android Wireless Application Development: Advanced Topics ( Developer’s Library) 3<sup>rd</sup> Edition, Author: Shane Conder, Publisher: Addison Wesley, ISBN-13: 978-0321813848</li> <li>6) Professional Android 4<sup>th</sup> Edition, Author: Reto Meier, Publication: Wrox, ISBN – 13: 978-1118949528</li> <li>7) Android 9 Development Cookbook, Author: Rick Boyer, Publisher: Packet Publishing, ISBN:978-1788991216</li> </ol>

	<p>8) Android Application Development (With Kitkat Support),  Author: Pradeep Kothari, Publisher: DreamTech Press., ISBN: 978-9351194095</p> <p>9) Android Studio 3.0 Development Essentials: Android 8 Edition  Author – Neil Smyth, Publisher: Payload Media, ISBN – 13: 978 – 1977540096</p>
Teaching Methodology	Class Work, Discussion, Self-Study, Seminars and/or Assignments
Evaluation Method	<p>50% Internal assessment.  (Internal Unit tests, assignments and attendance)  (Practical and Project work assessment)</p> <p>50% External assessment.  (External Theory Examination)  (External Practical Examination)</p>

## Course: 105-02: Android API and Framework

Course Code	105-02						
Course Title	Android API and Framework						
Credit	4						
Minimum hours per Semester	30 Hours Theory + 60 Hours Practical/Project work						
Review / Revision	June – 2025						
Pre-requisite	Concepts of Animation, Device and Components handling						
Course outcome	<p>CO1: Students will be able to understand the concepts of styles and theme, API and Framework.</p> <p>CO2: Students will have knowledge of testing Apps and publishing Apps</p> <p>CO3: Students will have knowledge of Flutter for cross platform application development</p> <p>CO4: Students will have knowledge of React Native</p>						
Mapping between Cos with PSOs		PS01	PS02	PS03	PS04	PSO5	PS06
	CO1						
	CO2						
	CO3						
	CO4						
Course Content	<p><b>Unit-1:</b></p> <p>1.1 Android styles and theme:  1.1.1 Defining and using Style  1.1.2 Style Inheritance</p> <p>1.2 Applying colors to Theme attributes  1.2.1 Nine-Patch Buttons  1.2.2 Applying Theme</p> <p>1.3 Testing the App  1.3.1 Enable USB Debugging on Device  1.3.2 Build APK and Install on Device</p> <p>1.4 Publishing App on Google Store:  1.4.1 Generate signed APK version of App  1.4.2 Registering Google Play store  1.4.3 Uploading signed APK</p> <p><b>Unit-2:</b></p> <p>2.1 Flutter Introduction:  2.1.1 Introduction, features , Advantages and Dis-Advnatages  2.1.2 Creating Flutter application in Android Studio  2.1.3 Architecture of Flutter Framework(Widgets,Gestures,States, Layers)</p> <p>2.2 Categories of Widgets:  2.2.1 Platform specific widgets  2.2.2 Layout widgets  2.2.3 State maintenance widgets  2.2.4 Platform independent, basic widgets</p> <p>2.3 Widget Layouts:  2.3.1 Single child widget  2.3.2 Multiple child widget</p> <p>2.4 Gestures and their events:  2.4.1 Gestures and their events:  Tap, Double Tap, Drag, Pinch, Spread/Zoom, Panning</p>						

	<p>2.4.2 Widgets for specific advacned Gestures: Dismissible, Draggable, LonPressDraggable, DragTarget, IgnorePointer, AbsorbPointer, Scrollable</p> <p>2.5 Navigation and Routing</p> <p>2.5.1 State categories: Ephermeral , App State</p> <p>2.5.2 Routing concepts and its relavant class and methods</p> <p>2.5.2.1 MaterialPageRoute class</p> <p>2.5.2.2 Navigator.push, Navigator.pop methods</p> <p><b>Unit-3:</b></p> <p>3.1 Flutter Animation system:</p> <p>3.1.1 Animation Class : Animation&lt;double&gt;, Animation&lt;Color&gt;, Animation&lt;Size&gt;, AnimationController</p> <p>3.1.2 Non-linear Animation (CurvedAnimation)</p> <p>3.2 Flutter framework Package</p> <p>3.2.1 Categories : Dart Package, Flutter Package</p> <p>3.2.2 Flutter Plugin Package</p> <p>3.3 REST based Application:</p> <p>3.3.1 http class functionality</p> <p>3.3.1.1 methods: read, get, post, put, head, delete</p> <p>3.3.1.2 Data access from Web server and display using ListView</p> <p>3.4 Working with Database:</p> <p>3.4.1 use of sqlite package</p> <p>3.4.2 Functionality of SQLite package: create and open SQLite database</p> <p>3.5 Execute SQL statements (CRUD operation)</p> <p>3.5.1 Filter queries ( IN, Between, LIKE, NOT LIKE)</p> <p>3.5.2 Internationalization</p> <p>3.6 Testing and Deployment:</p> <p>3.6.1 Automated testing of application</p> <p>3.6.2 Types of Testing(Unit test, Widget Test, Integration test)</p> <p>3.6.3 Deploying Flutter application in Android (Create Apk file)</p> <p>3.6.4 Deploying Flutter application in iOS (Create apk file)</p> <p><b>Unit-4:</b></p> <p>4.1 Fundamentals of React Native:</p> <p>4.1.1 Features, Advantages, Limitations</p> <p>4.1.2 Installing and configuration</p> <p>4.1.3 Creating basic app "hello world" using React Native</p> <p>4.1.4 State, Props and Container Component</p> <p>4.1.5 Container and Presentation Component</p> <p>4.2 Layouts and components:</p> <p>4.2.1 flexbox and properties</p> <p>4.2.2 ListView, Text Input, ScrollView</p> <p>4.2.3 Placing images, Buttons</p> <p>4.3 React Native HTTP:</p> <p>4.3.1 Handling network request using fetch</p> <p>4.4 React Native Router:</p> <p>4.4.1 Install router</p> <p>4.4.2 Add Router</p> <p>4.4.3 Create components</p>
Reference Books	<p>1) Android Application Development (With Kitkat Support), Author: Pradeep Kothari, Publisher:DreamTech Press.,ISBN:978-9351194095</p> <p>2) Advanced Android™ Application Development, Fourth Edition, Author: Joseph Annizzi Jr., Lauren Darcey, Shane Conder, Publisher: Addison Wesley Professional, ISBN : 9720133892420</p> <p>3) Flutter for Beginners: An introductory guide to building cross-</p>

	<p>platform mobile applications with Flutter and Dart 2, Author: Alessandro Biessek, Packt Publishing House,ISBN:978-1788996082</p> <p>4) Beginning Flutter: A Hands On Guide to App Development, Author: Marco L. Napoli, Publisher: Wrox, ISBN:978-1119550822</p> <p>5) Practical Flutter: Improve your Mobile Development with Google’s Latest Open-Source SDK ,Author: Frank Zammetti, Publisher: Apress, ISBN-13: 978-1484249710</p> <p>6) Flutter Complete Reference: create beautiful, fast and native apps for any device, Author: Alberto Miola, Kindle Edition</p> <p>7) React Native for Mobile Development: Harness the Power of React Native to Create Stunning iOS and Android Applications 2nd ed. Edition, Author: Akshat Paul, Publisher: Apress, ISBN – 13: 978-1484244531</p> <p>8) React Native By Example: Native mobile development with React, Author: Richard Kho, Publisher: Packt Publication, ISBN – 13: 978- 1786464750</p> <p>9) React Native Cookbook: Recipes for solving common React Native development problems, 2nd Edition, Author: Dan Ward, Publisher: Packt Publication, ISBN – 13: 978-1788991926</p> <p>10) React and React Native: A complete hands-on guide to modern web and mobile development with React.js, 3rd Edition, Author: Adam Boduch, Publisher: Packt Publication, ISBN – 13: 978-1839211140</p>
Teaching Methodology	Class Work, Discussion, Self-Study, Seminars and/or Assignments
Evaluation Method	<p>50% Internal assessment. (Internal Unit tests, assignments and attendance) (Practical and Project work assessment)</p> <p>50% External assessment. (External Theory Examination) (External Practical Examination)</p>

# First Year M.Sc. (CA)

## Semester – II

Course Code	Course Title	Level of Course	Course Credits	Teaching Hours/week	
				Th.+Pra.	Theory Practical/ Fieldwork /Project/ Internship
201	Open Source Cloud and Database Deployment	400-499	4	2	4
202	Data Analytics and Visualization	400-499	4	2	4
203	Project	400-499	12	-	Full time Project at Industry
204	On-the-Job Training and Industry Immersion	400-499	4	-	Documentation , Content writing, Presentation and Viva-voce
Total			22		

Course Code	Course Title	Course Credit	University Exam Type	Exam Duration	External Marks	Internal Marks	Total Marks
201	Open Source Cloud and Database Deployment	4	Theory/ Written	1 Hours	25	25	100
			Practical/Project	2 Hours	25	25	
202	Data Analytics and Visualization	4	Theory/ Written	1 Hours	25	25	100
			Practical/Project	2 Hours	25	25	
203	Project	12	Practical/Project	Project viva-voce	50	50	100
204	On-the-Job Training and Industry Immersion	4	Documentation , Reporting, Presentation and Viva-voce	Presentation and Viva-voce	50	50	100
<b>Total</b>		<b>24</b>			<b>200</b>	<b>200</b>	<b>400</b>

For Project: Students will individually develop a full-scale project and submit progress report to their concerned internal guides every week. One hour load will be considered per every four students/week for Project work.

## Course: 201 : Open Source Cloud and Database Deployment

Course Code	201						
Course Title	Open Source Cloud and Database Deployment						
Credit	4 credits						
Duration	30 hours Theory + 60 Hours of Practical/Project work.						
Review / Revision	June – 2025						
Purpose of Course	<p>The course "Open Source Cloud and Database Deployment" is designed to equip students with foundational and practical knowledge of deploying applications and databases using open-source cloud platforms. It introduces learners to essential concepts of cloud computing, helping them understand how to work with virtual machines, manage data remotely, and integrate cloud-hosted databases into real-world applications. The course emphasizes hands-on experience in uploading, organizing, and maintaining datasets on the cloud while using tools that require no proprietary licenses. Students will learn to connect applications with cloud-based databases and ensure smooth data flow between client and server components. A key focus is placed on deploying both static and dynamic websites and full-stack projects using open-source infrastructure. This course empowers students to independently manage cloud-based projects from development to deployment, making them industry-ready for cloud-centric roles.</p>						
Course Objective	<p>CO1: Understand and explain the fundamentals of cloud computing and open-source cloud platforms.</p> <p>CO2: Demonstrate the ability to upload and manage datasets on open cloud infrastructure.</p> <p>CO3: Apply techniques to maintain and manipulate cloud-hosted databases and tables.</p> <p>CO4: Integrate cloud-based datasets with full-stack applications using open-source tools.</p> <p>CO5: Deploy static and dynamic websites or projects on open-source cloud environments.</p>						
Pre-requisite	<p>Students should have basic knowledge of web development using HTML, CSS, JavaScript, and fundamentals of backend development using Node.js.</p> <p>Familiarity with Git, GitHub, and basic database concepts is recommended for effective understanding and hands-on practice.</p>						
Mapping between Cos with PSOs		<b>PSO1</b>	<b>PSO2</b>	<b>PSO3</b>	<b>PSO4</b>	<b>PSO5</b>	<b>PSO6</b>
	CO1	✓	✓				✓
	CO2	✓	✓	✓		✓	
	CO3	✓	✓	✓		✓	✓
	CO4		✓	✓	✓	✓	
	CO5	✓	✓		✓	✓	✓

Course Content	<p><b>Unit-1: Introduction to Open Source Cloud</b></p> <p>1.1 Fundamentals of Cloud Computing</p> <ul style="list-style-type: none"> <li>1.1.1 Definition and Characteristics of Cloud Computing</li> <li>1.1.2 Cloud Service Models: IaaS, PaaS, SaaS</li> <li>1.1.3 Deployment Models: Public, Private, Hybrid, Community Cloud</li> <li>1.1.4 Advantages of using Open Source Cloud Solutions</li> </ul> <p>1.2 Open Source Cloud Platforms</p> <ul style="list-style-type: none"> <li>1.2.1 Introduction to platforms like Linode, DigitalOcean, and OpenStack</li> <li>1.2.2 Setting up a Virtual Machine (VM) on open cloud platforms</li> <li>1.2.3 SSH access, security setup, and remote server basics</li> </ul> <p>1.3 Cloud Storage Basics</p> <ul style="list-style-type: none"> <li>1.3.1 Concepts of cloud-based file systems and object storage</li> <li>1.3.2 Working with open-source storage tools (e.g., NextCloud, MinIO)</li> </ul> <p><b>Unit-2: Uploading and Managing Data on Cloud</b></p> <p>2.1 Data Preparation and Format</p> <ul style="list-style-type: none"> <li>2.1.1 CSV, JSON, and SQL dump formats</li> <li>2.1.2 Structuring data for cloud compatibility</li> </ul> <p>2.2 Uploading Datasets to Cloud</p> <ul style="list-style-type: none"> <li>2.2.1 Using SCP, SFTP, and cloud dashboards</li> <li>2.2.2 Uploading tables to cloud databases (e.g., PostgreSQL, MariaDB)</li> <li>2.2.3 Importing datasets using CLI and GUI (e.g., Adminer, DBeaver)</li> </ul> <p>2.3 Managing Cloud-hosted Databases</p> <ul style="list-style-type: none"> <li>2.3.1 Creating and modifying tables</li> <li>2.3.2 Running SQL queries remotely</li> <li>2.3.3 Backup and recovery of datasets</li> </ul> <p><b>Unit-3: Application and Dataset Integration</b></p> <p>3.1 Backend Integration</p> <ul style="list-style-type: none"> <li>3.1.1 Connecting cloud databases with Node.js/Express applications</li> <li>3.1.2 Environment variables and security practices (dotenv)</li> <li>3.1.3 Using ORMs (Sequelize/Mongoose) for database operations</li> </ul> <p>3.2 Frontend Data Display</p> <ul style="list-style-type: none"> <li>3.2.1 Fetching data from backend APIs</li> <li>3.2.2 Displaying dynamic data using React components</li> <li>3.2.3 Error handling and live refresh</li> </ul> <p>3.3 Real-time Database Interaction</p> <ul style="list-style-type: none"> <li>3.3.1 API calls using Axios/Fetch</li> <li>3.3.2 Form submission and storage</li> <li>3.3.3 Validation and data binding</li> </ul> <p><b>Unit-4: Cloud-based Deployment of Projects</b></p> <p>4.1 Deployment of Static Websites</p> <ul style="list-style-type: none"> <li>4.1.1 Hosting using GitHub Pages or Netlify (for frontend-only apps)</li> <li>4.1.2 CI/CD overview and automation for static deployment</li> </ul> <p>4.2 Deployment of Full Stack Projects</p> <ul style="list-style-type: none"> <li>4.2.1 Using open-source PaaS: Render, Railway, CapRover</li> <li>4.2.2 Deploying Node.js apps with database connection</li> <li>4.2.3 Dockerizing full stack applications for cloud deployment</li> </ul> <p>4.3 Post-deployment Operations</p> <ul style="list-style-type: none"> <li>4.3.1 Monitoring, log management using open tools (e.g., Logtail, Grafana basics)</li> <li>4.3.2 Common deployment issues and troubleshooting</li> <li>4.3.3 Maintenance and update of live websites</li> </ul>
----------------	--

Reference Books	<p>1) <i>Cloud Computing: Concepts, Technology &amp; Architecture</i> by Thomas Erl, published by Pearson Education, ISBN: 9780133387520.</p> <p>2) <i>Learning MySQL and MariaDB: Heading in the Right Direction with MySQL and MariaDB</i> by Russell J.T. Dyer, published by O'Reilly Media, ISBN: 9781449357283.</p> <p>3) <i>Docker: Up &amp; Running</i> by Karl Matthias and Sean P. Kane, published by O'Reilly Media, ISBN: 9781492036739.</p> <p>4) <i>Node.js Web Development</i> by David Herron, published by Packt Publishing, ISBN: 9781838987575.</p> <p>5) <i>Learning React: Functional Web Development with React and Redux</i> by Alex Banks and Eve Porcello, published by O'Reilly Media, ISBN: 9781492051725.</p> <p>6) <i>Database Systems: Models, Languages, Design and Application Programming</i> by Ramez Elmasri and Shamkant B. Navathe, published by Pearson Education, ISBN: 9789332582705.</p> <p>7) <i>Fundamentals of Database Systems</i> by C.J. Date, published by Pearson Education, ISBN: 9780133970777.</p> <p>8) <i>Cloud Computing</i> by Dr. Kumar Saurabh, published by Wiley India Pvt. Ltd., ISBN: 9788126547295.</p> <p>9) <i>Cloud Computing: A Hands-on Approach</i> by Arshdeep Bahga and Vijay Madisetti, published by University Press, ISBN: 9780996025515.</p> <p>10) <i>OpenStack in Action</i> by Cody Bumgardner, published by Manning Publications, ISBN: 9781617292163.</p>
Teaching Methodology	Class Work, Discussion, Self-Study, Seminars and/or Assignments, Case study, Project work.
Evaluation Method	<p>50% Internal assessment.          (Internal Unit tests, assignments and attendance)          (Practical and Project work assessment)</p> <p>50% External assessment.          (External Theory Examination)          (External Practical Examination)</p> <p>[students are required to acquire minimum passing marks in combination of Theory and Practical/Project Examination]</p>

## Course: 202 : Data Analytics & Visualization

<b>Course Code</b>	202								
<b>Course Title</b>	Data Analytics & Visualization								
<b>Credit</b>	4 credits								
<b>Duration</b>	30 Hours of Theory + 60 Hours of Applied work (practical / project)								
<b>Review / Revision</b>	June – 2025								
<b>Purpose of Course</b>	<p>This course is designed to introduce students to the core concepts and tools used in data analytics and visualization with Python, beginning from the fundamental level. It enables students to explore and analyze real-world datasets using Python libraries such as Pandas, NumPy, Matplotlib, Seaborn, Plotly, and Streamlit. The course emphasizes practical and application-driven learning, allowing students to interact with databases like SQLite and MongoDB for data extraction and manipulation. Students will also gain hands-on experience in building interactive dashboards using Tableau. Through projects and lab work, learners will develop the ability to extract insights from data, build visual stories, and support data-driven decision-making. This course prepares students for roles such as data analysts, business intelligence developers, and data visualization specialists.</p>								
<b>Pre-requisite</b>	Basic understanding of Python programming and database concepts is recommended. Familiarity with any programming logic and web interface is helpful but not mandatory.								
<b>Course outcome</b>	<p>CO1: Understand and apply fundamental concepts of data analysis using Python libraries like Pandas and NumPy.</p> <p>CO2: Perform data cleaning, transformation, and exploration using real-world datasets from databases.</p> <p>CO3: Visualize data effectively using libraries like Matplotlib, Seaborn, Plotly, and Streamlit.</p> <p>CO4: Connect Python with databases such as SQLite and MongoDB for integrated data handling.</p> <p>CO5: Design interactive dashboards and reports using Tableau for business and analytical insights..</p>								
<b>Mapping Course Outcome(CO) and Program Specific Outcomes(PSO).:</b>		<b>PS O1</b>	<b>PS O2</b>	<b>PS O3</b>	<b>PS O4</b>	<b>PS O5</b>	<b>PS O6</b>	<b>PS O7</b>	<b>PS O8</b>
	CO1	✓		✓					✓
	CO2		✓	✓		✓			
	CO3			✓	✓	✓			✓
	CO4	✓	✓			✓		✓	
	CO5			✓	✓		✓		✓

<b>Course Content:</b>	<p><b>Unit-1: Data Analysis Using Python (Pandas and NumPy)</b></p> <p>1.1 Intermediate Pandas Operations</p> <p>    1.1.1 Multi-indexing, merging, pivoting, reshaping data</p> <p>    1.1.2 GroupBy, windowing, aggregation, rolling functions</p> <p>1.2 Efficient Computation with NumPy</p> <p>    1.2.1 Broadcasting, slicing, filtering, conditional logic</p> <p>    1.2.2 Matrix algebra and statistical computations</p> <p>1.3 Working with Complex Data</p> <p>    1.3.1 Date/time series operations</p> <p>    1.3.2 Reading and writing data from Excel, JSON, XML</p> <p><b>Unit-2: Database Integration and Data Preprocessing</b></p> <p>2.1 Database Connectivity</p> <p>    2.1.1 Connecting to SQLite using sqlite3</p> <p>    2.1.2 Connecting to MongoDB using pymongo</p> <p>    2.1.3 Reading/writing SQL and NoSQL data in Pandas</p> <p>2.2 Data Cleaning and Preparation</p> <p>    2.2.1 Handling missing values, duplicates, outliers</p> <p>    2.2.2 Encoding categorical variables</p> <p>    2.2.3 Data normalization, transformation pipelines</p> <p><b>Unit-3: Data Visualization using Python Libraries</b></p> <p>3.1 Static Visualization</p> <p>    3.1.1 Matplotlib: Plot types, layout, styling, exporting</p> <p>    3.1.2 Seaborn: Statistical plots – boxplots, heatmaps, pairplots</p> <p>3.2 Interactive Visualization</p> <p>    3.2.1 Plotly: Line, scatter, pie, treemap, 3D plots</p> <p>    3.2.2 Streamlit: Layout, input widgets, charts, interactivity</p> <p>    3.2.3 Hosting and sharing Streamlit dashboards</p> <p><b>Unit-4: Tableau for Business Intelligence &amp; Capstone Project</b></p> <p>4.1 Tableau Fundamentals</p> <p>    4.1.1 Tableau Public interface and data connection</p> <p>    4.1.2 Calculated fields, filters, parameters, hierarchies</p> <p>    4.1.3 Designing bar, line, pie, map charts</p> <p>4.2 Dashboard Design and Data Storytelling</p> <p>    4.2.1 Creating interactive dashboards and reports</p> <p>    4.2.2 Using Tableau and Python for comparative analysis</p> <p>4.3 Capstone Project</p> <p>    4.3.1 Problem definition, data collection from SQLite/MongoDB</p> <p>    4.3.2 Cleaning, analyzing, and visualizing data</p> <p>    4.3.3 Building dashboards in Tableau/Streamlit and preparing a project report</p>
------------------------	--

<b>Reference Books:</b>	<p>1) <i>Python for Data Analysis</i>, Wes McKinney, O'Reilly Media, ISBN: 9781491957660</p> <p>2) <i>Practical Statistics for Data Scientists</i>, Peter Bruce, Andrew Bruce, O'Reilly Media, ISBN: 9781492072942</p> <p>3) <i>Hands-On Data Analysis with Pandas</i>, Stefanie Molin, Packt Publishing, ISBN: 9781800563452</p> <p>4) <i>Python Data Science Handbook</i>, Jake VanderPlas, O'Reilly Media, ISBN: 9781491912058</p> <p>5) <i>Data Visualization with Python and JavaScript</i>, Kyran Dale, O'Reilly Media, ISBN: 9781491920510</p> <p>6) <i>Interactive Data Visualization with Python</i>, Abha Belorkar, BPB Publications, ISBN: 9789389898305</p> <p>7) <i>Mastering Tableau</i>, Marleen Meier and David Baldwin, Packt Publishing, ISBN: 9781839211393</p> <p>8) <i>Data Wrangling with Python</i>, Jacqueline Kazil and Katharine Jarmul, O'Reilly Media, ISBN: 9781491948811</p> <p>9) <i>Streamlit for Data Science</i>, Tyler Richards, Packt Publishing, ISBN: 9781800567429</p> <p>10) <i>Hands-On Data Visualization with Tableau</i>, Tristan Guillevin, Packt Publishing, ISBN: 9781838983628</p>
<b>Teaching Methodology:</b>	Class Work, Discussion, Self-Study, Seminars and/or Assignments, Case study, Project work.
<b>Evaluation</b>	<p>50% Internal assessment.        (Internal Unit tests, assignments and attendance)        (Practical and Project work assessment)</p> <p>50% External assessment.        (External Theory Examination)        (External Practical Examination)</p> <p>[students are required to acquire minimum passing marks in combination of Theory and Practical/Project Examination]</p>

## Course: 203: Project

<b>Course Code</b>	203
<b>Course Title</b>	Project
<b>Credit</b>	12 Credits
<b>Duration</b>	360 Hours
<b>Duration</b>	Part time project
<b>Review / Revision</b>	June – 2025
<b>Purpose of Course</b>	<p>This course aims to provide students with hands-on, full-time industry project experience where they apply, integrate, and demonstrate knowledge and skills acquired from previous coursework including software development, data science, AI, cloud deployment, version control, web technologies, and visualization tools. The project will help students learn project planning, communication with industry mentors, adherence to professional practices, and delivery of a functional, demonstrable solution. Emphasis is placed on real-world problem-solving, client interaction, teamwork, version control, documentation, and deployment.</p> <p>It is desirable that students join an organization (through Industry / Software firms / Any other relevant organization / internship with any software development organization) in process to apply their acquired knowledge related to their selected Track. Students will develop and deploy a reasonable size of project applying all relevant skills and understand the project development phases.</p> <p>Project work will be on any of the platforms including Web technology, Android and mobile technology with integration.</p>
<b>Pre-requisite</b>	Prior knowledge of Courses semester-1 courses 102 to 105
<b>Course outcome</b>	<p>CO1: Identify and define a real-world industry problem and propose a feasible project plan.</p> <p>CO2: Apply appropriate methodologies, tools, and technologies to implement the project solution.</p> <p>CO3: Collaborate effectively with industry mentors and project team members through agile practices.</p> <p>CO4: Develop and demonstrate a working solution with technical documentation and code repository.</p> <p>CO5: Present the project outcome through a well-structured report, deployment, and viva-voce.</p>

<p><b>Phases and components</b></p>	<p><b>1: Project Planning and Proposal</b>  1.1 Identifying a real-world industry problem in consultation with industry mentor or organization  1.2 Conducting requirement analysis and defining objectives, scope, and deliverables  1.3 Preparing system design document – architecture, modules, tech stack  1.4 Creating a project schedule using agile or iterative models with task breakdown  1.5 Setting up version control (Git/GitHub) and basic documentation</p> <p><b>2: Implementation and Iterative Development</b>  2.1 Development using relevant tools: programming, frameworks, databases, APIs  2.2 Code versioning, team collaboration, and code reviews via GitHub  2.3 Integrating with cloud services, third-party APIs, or datasets as needed  2.4 Implementing backend/frontend, dashboards, data pipelines, etc.  2.5 Conducting mid-phase internal review and progress evaluation</p> <p><b>3: Testing, Deployment, and Delivery</b>  3.1 Testing methodologies: Unit, Integration, User Acceptance Testing  3.2 UI/UX review and client feedback cycles  3.3 Deployment of solution on a test/live server (AWS, Render, Firebase, Streamlit, etc.)  3.4 Final delivery to the organization/client with live walkthrough  3.5 Gathering user feedback and preparing revision plan if required</p> <p><b>4: Project Report, Presentation, and Viva</b>  4.1 Comprehensive project report: Objective, Literature Survey, Methodology, Design, Implementation, Results, and Conclusion  4.2 Creating user manual, system guide, and deployment instructions  4.3 Final submission of GitHub repo with commit history  4.4 Presentation and viva-voce to departmental/industry panel  4.5 Evaluation based on innovation, execution, presentation, and documentation</p>												
<p><b>Project Evaluation:</b></p>	<p>50% Internal : Based on presentation of the project and viva-voce.  50% External: Based on project presentation and viva-voce.</p> <table border="1" data-bbox="517 1288 1161 1480"> <thead> <tr> <th>Component</th> <th>Weightage</th> </tr> </thead> <tbody> <tr> <td>Problem Definition &amp; Proposal</td> <td>10%</td> </tr> <tr> <td>Design and Implementation</td> <td>30%</td> </tr> <tr> <td>Testing &amp; Deployment</td> <td>20%</td> </tr> <tr> <td>Project Report</td> <td>15%</td> </tr> <tr> <td>Presentation &amp; Viva</td> <td>25%</td> </tr> </tbody> </table> <p>The evaluation of the external assessment will be carried out by panel of three examiners: (i) One examiner from the same institute (ii) One examiner from software industry/corporate (iii) One examiner from other institute affiliated to the university having minimum 11 years of teaching experience at graduation level in computer faculty.</p>	Component	Weightage	Problem Definition & Proposal	10%	Design and Implementation	30%	Testing & Deployment	20%	Project Report	15%	Presentation & Viva	25%
Component	Weightage												
Problem Definition & Proposal	10%												
Design and Implementation	30%												
Testing & Deployment	20%												
Project Report	15%												
Presentation & Viva	25%												

## Course: 204: On-the-Job Training and Industry Immersion

<b>Course Code</b>	204						
<b>Course Title</b>	On-the-Job Training and Industry Immersion						
<b>Credit</b>	4 Credits						
<b>Duration</b>	120 Hours (including field visits, analysis, mentoring, report writing, and presentation)						
<b>Implementation Year:</b>	June – 2025						
<b>Purpose of Course</b>	This course aims to provide students with first-hand exposure to the real-time working environment of the software and IT industry. Through structured industry visits and observational learning, students will understand the software development lifecycle, team structures, technology stacks, HR strategies, project methodologies, and company work culture. This immersion will bridge the gap between academic learning and industrial practices, enhancing the student's readiness for employment. The culminating activities—presentation and report—help students synthesize their learning and develop documentation and communication skills.						
<b>Pre-requisite</b>	Students must have completed the first semester of the M.Sc. (Computer Application) program and should possess fundamental knowledge of software development and IT concepts.						
<b>Course outcome</b>	<p><b>CO1:</b> Understand organizational structure, working hierarchy, and operational workflow in a software company.</p> <p><b>CO2:</b> Identify the technology stack, development practices, and software tools used in real-world environments.</p> <p><b>CO3:</b> Analyze HR functions, team dynamics, and work culture strategies in IT companies.</p> <p><b>CO4:</b> Demonstrate knowledge integration through structured documentation and presentation.</p> <p><b>CO5:</b> Reflect on industry practices and propose self-improvement plans for better employability.</p>						
<b>Course Outcome / Program Specific Outcome mapping:</b>	CO / PSO	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6
	CO1	✓	✓	✓		✓	
	CO2	✓	✓	✓	✓		✓
	CO3		✓		✓	✓	✓
	CO4	✓		✓	✓		✓
	CO5			✓	✓	✓	

<p><b>Phases and components of Course Outcomes:</b></p>	<p><b>1: Orientation and Preparation for Industry Visit</b>  1.1 Understanding the Purpose of OJT  1.2 Guidelines for Professional Conduct during Visits  1.3 Basics of Organizational Structures in IT  1.4 Overview of Roles in a Software Company (Developer, QA, Analyst, HR, DevOps, etc.)  1.5 Tools and Templates for Observation, Interview, and Note-Taking</p> <p><b>2: Industry Visit and Observation</b>  2.1 Visiting Software Companies (Startups, Mid-size, or Large IT firms)  2.2 Understanding Software Development Lifecycle (SDLC/Agile)  2.3 Identifying Technologies, Tools, and Platforms in Use  2.4 Observing Team Collaboration, Roles &amp; Responsibilities  2.5 Collecting Data: Hierarchy Charts, Project Flow, Software Architecture (as allowed)</p> <p><b>3: HR, Strategy, and Organizational Analysis</b>  3.1 HR Functions: Hiring Strategies, Performance Appraisal, Onboarding  3.2 Employee Benefits and Career Growth Pathways  3.3 Corporate Culture, Diversity, Work-Life Balance Initiatives  3.4 CSR, Sustainability, and Company Vision  3.5 Learning from Employee Interactions: Q&amp;A, Mentorship, Informal Interviews</p> <p><b>4: Reporting, Documentation, and Presentation</b>  4.1 Preparing a Structured Industry Visit Report  4.2 Creating Hierarchy Diagrams, Workflows, Tech Stack Documentation  4.3 Analyzing Key Takeaways and Lessons Learned  4.4 Preparing and Delivering Presentation (Team/Individual)  4.5 Viva-Voce or Reflective Discussion with Faculty Panel</p>												
<p><b>Project Evaluation:</b></p>	<p>50% Internal : Based on presentation on following components and viva-voce.  50% External: Based on presentation on following components and viva-voce.</p> <table border="1" data-bbox="730 1193 1251 1444"> <thead> <tr> <th>Component</th> <th>Weightage</th> </tr> </thead> <tbody> <tr> <td>Participation &amp; Conduct</td> <td>10%</td> </tr> <tr> <td>Observation Logs &amp; Notes</td> <td>10%</td> </tr> <tr> <td>Final Report</td> <td>40%</td> </tr> <tr> <td>Presentation &amp; Demo</td> <td>30%</td> </tr> <tr> <td>Reflective Discussion/Viva</td> <td>10%</td> </tr> </tbody> </table> <p>The evaluation of the external assessment will be carried out by panel of three examiners: (i) One examiner from the same institute (ii) One examiner from software industry/corporate (iii) One examiner from other institute affiliated to the university having minimum 11 years of teaching experience at graduation level in computer faculty.</p>	Component	Weightage	Participation & Conduct	10%	Observation Logs & Notes	10%	Final Report	40%	Presentation & Demo	30%	Reflective Discussion/Viva	10%
Component	Weightage												
Participation & Conduct	10%												
Observation Logs & Notes	10%												
Final Report	40%												
Presentation & Demo	30%												
Reflective Discussion/Viva	10%												