

SARVAJANIK UNIVERSITY
FACULTY OF COMPUTER SCIENCE AND APPLICATIONS
M.Sc. (Information Technology)
Semester - I and Semester - II



SARVAJANIK
UNIVERSITY

INCLUSIVE | INTEGRATED | INNOVATIVE

Faculty of Computer Science and Applications

M.Sc. (Information Technology)

SEMESTER- 2

DETAILED CURRICULUM
(W.E.F. 2026-27)

SARVAJANIK UNIVERSITY
FACULTY OF COMPUTER SCIENCE AND APPLICATIONS
M.Sc. (Information Technology)
Semester - I and Semester - II

Name of College: Shree Ramkrishna Institute of Computer Education and Applied Sciences						
Faculty	Computer Science and Applications		Program		M.Sc. (Information Technology)	
Year	2026-27		Version		3	
Semester	2		Effective From		June 2026	
Course Code	MSCS31201	Course Name	DSC : Web Programming-II (TH)			
Teaching Scheme				Examination Scheme		
Credits	Lecture (L)	Tutorial (T)	Practical (P)	Continuous and Comprehensive Evaluation (CCE)	Semester end examinations (SEE)	Total
3	3	0	-	50	50	100

Purpose of Course	To provide comprehensive knowledge about JavaScript-based framework built on Google Chrome's JavaScript V8 Engine
Course Objective	To provide knowledge on how to develop I/O intensive web applications like video streaming sites, single-page applications, and other web applications using Node.js framework
Pre-requisite	Basic understanding of JavaScript, HTML, CSS and AJAX
Course Outcome	After having completed the course the student will gain: • Understanding of Node.js Environment • Knowledge of Node Modules Technical know-hows of Full Stack Node.js based development • Application of Node.js web development of real life application
Course Content	Unit 1: Introduction Node.js 1.1 Features and Applications 1.1.1 Installing Node, Node Hosting Environments 1.1.2 Node Building Blocks 1.2 Exception Handling. 1.3 REPL Terminal 1.4 Overview of Node Module System 1.5 Overview of Node Package Manager 1.6 Overview of Node Version Manager 1.7 Axios, bcrypt, jsonwebtokens, multer, nodemon, dotenv

SARVAJANIK UNIVERSITY
FACULTY OF COMPUTER SCIENCE AND APPLICATIONS
M.Sc. (Information Technology)
Semester - I and Semester - II

	Unit 2 : Node with the Local System and the Web 2.1 Streams and Pipes 2.2 Node and the File System- The fs.Stats class, The File System Watcher, File Read and Write, Directory access and Maintenance, File Streams 2.3 Resource Access with Path Unit 3: Node and Web Application 3.1 Using Apache to proxy Node application 3.2 Routing in NodeJS 3.3 Routing and Callback function 3.4 Modern Deployment Basics: PM2 3.5 Hosting overview: Render, Railway, Vercel 3.6 What is NGINX Unit 4: Full-Stack Node development 4.1 The Express Application Framework 4.2 Express Supportive Modules, Body-parser, Method Override 4.3 Basic EJS, Folder structure (routes/controllers/models) 4.4 Integrating NodeJS, MongoDB and REST APIs Unit 5: NodeJS Application Management 5.1 Project Structure, MVC architecture, Folder structuring for APIs 5.2 Controllers, Routes, Services, Models 5.3 Environment Variables, Using .env file, Config management 5.4 Modern JavaScript in Node.js - ES Modules (import/export) 5.5 Async/Await, Built-in fetch() in Node, Error Handling - Centralized error handler, Async error handling HTTP response structure Unit 6: Advanced NodeJS Operations 6.1 Basic Authentication -JWT basics, bcrypt hashing, Protected routes 6.2 Postman / Thunder Client - API testing, Status codes, Sending headers/body/params 6.3 NodeJS in Real-time -WebSocket, Socket.IO
Text and Reference Literature	1. Learning Node Moving to the server side Shelley Powers O'Reilly Literature Publication Building Node Applications with MongoDB and Backbone Mike Wilson O'Reilly SPD Publication 2. GEO, CouchDB & NodeJS Mick Thompson O'Reilly SPD Publication 3. Web Development with Node and Express, Ethan Brown, O'Reilly Publication 4. Node.js in Action, Alex Young, Bradley Meck, Mike Cantelon

SARVAJANIK UNIVERSITY
FACULTY OF COMPUTER SCIENCE AND APPLICATIONS
M.Sc. (Information Technology)
Semester - I and Semester - II

Name of College: Shree Ramkrishna Institute of Computer Education and Applied Sciences						
Faculty	Computer Science and Applications		Program	M.Sc. (Information Technology)		
Year	2026-27		Version	3		
Semester	2		Effective From	June 2026		
Course Code	MSCS31202	Course Name	DSC: Web Programming-II (PR)			
Teaching Scheme				Examination Scheme		
Credits	Lecture (L)	Tutorial (T)	Practical (P)	Continuous Assessments (CCE)	Semester end examinations (SEE)	Total
1	0	0	2	25	25	50

Purpose of Course	To provide comprehensive knowledge about JavaScript-based framework built on Google Chrome's JavaScript V8 Engine
Course Objective	To provide knowledge on how to develop I/O intensive web applications like video streaming sites, single-page applications, and other web applications using Node.js framework
Pre-requisite	Basic understanding of JavaScript, HTML, CSS and AJAX
Course Outcome	After having completed the course the student will gain: • Understanding of Node.js Environment • Knowledge of Node Modules Technical know-hows of Full Stack Node.js based development • Application of Node.js web development of real life application
Course Content	The course covers the fundamentals of NodeJS, including the NPM, NVM, express framework, EJS, and MVC implementations with practical applications.
Practical	List of practical will be prepared at the beginning of each semester
Reference Books	As per paper DSC– Web Programming-II (Th)

SARVAJANIK UNIVERSITY
FACULTY OF COMPUTER SCIENCE AND APPLICATIONS
M.Sc. (Information Technology)
Semester - I and Semester - II

Name of College: Shree Ramkrishna Institute of Computer Education and Applied Sciences						
Faculty	Computer Science and Applications		Program	M.Sc. (Information Technology)		
Year	2026-27		Version	3		
Semester	2		Effective From	June 2026		
Course Code	MSCS31203	Course Name	DSC: Advanced Cloud Programming (TH)			
Teaching Scheme				Examination Scheme		
Credits	Lecture (L)	Tutorial (T)	Practical (P)	Continuous and Comprehensive Evaluation (CCE)	Semester end examinations (SEE)	Total
3	3	0	-	50	50	100

Purpose of Course	The purpose of course is to establish the foundation of cloud-native application development using micro service architecture.
Course Objective	◆ To introduce cloud computing fundamentals, architectural models, service models, and deployment environments. ◆ To understand the evolution of software architectures and develop cloud-native applications using microservices. ◆ To design scalable microservices with REST APIs, business logic layers, and data persistence techniques. ◆ To explore cloud capabilities, security, monitoring, and emerging trends such as Edge, Quantum, and AI-driven cloud computing.
Pre-requisite	Fundamental knowledge of software engineering, programming and networking
Course Out come	After completion of the course, a student will be able to: ● Understand cloud computing concepts, architecture, service & deployment models. ● Compare monolithic, SOA, and microservice architectures and apply cloud-native principles. ● Design and implement microservices with REST APIs, business logic, and persistence layers. ● Apply data management, transactional messaging, and fault-tolerance patterns in MSA.
Course Content	UNIT 1 – Cloud Computing Fundamentals 1.1. Evolution of Cloud Computing 1.2. Characteristics, benefits & challenges 1.3. Cloud architectural model 1.4. Cloud service models (IaaS, PaaS, SaaS)

SARVAJANIK UNIVERSITY
FACULTY OF COMPUTER SCIENCE AND APPLICATIONS
M.Sc. (Information Technology)
Semester - I and Semester - II

1.5. Cloud deployment models (Public, Private, Hybrid)

UNIT 2 – Evolution of Architectural Styles for Cloud-Native Applications

- 2.1. Evolution of application architectures
 - 2.1.1 Monolithic architecture
 - 2.1.2 Service oriented architecture
 - 2.1.3 Micro service architecture
- 2.2. Characteristics of cloud-native systems
- 2.3. Need for distributed architectures
- 2.4. Virtualization – Concept, Hypervisor and its Types
- 2.5. Containerization

UNIT 3 – Microservices Architecture & Decomposition Strategies

- 3.1. Microservices architecture: principles & characteristics
- 3.2. Bounded context & domain-driven concepts
- 3.3. Decomposition strategies:
 - 3.3.1. By business capability
 - 3.3.2. By subdomain
 - 3.3.3. Self-contained services
 - 3.3.4. Service-per-team
- 3.4. Service communication patterns -REST and Event Driven Design

UNIT 4 – Microservice creation and orchestration

- 4.1 API & Service Layer
 - 4.1.1. RESTful API development
 - 4.1.2. Resource modelling & URI design
 - 4.1.3. JSON-based data exchange
 - 4.1.4. Dependency Injection & stateless services
 - 4.1.5. Error/exception handling strategies
 - 4.1.6. Packaging for deployment
- 4.2 Business Logic Layer
 - 4.2.1. Designing scalable service logic
 - 4.2.2. Stateless components for parallel execution
 - 4.2.3. Integration between API → Service → Data layers
 - 4.2.4. Designing scalable service logic
- 4.3 Persistence Layer
 - 4.3.1. Persistence concepts for microservices
 - 4.3.2. Entity modelling & CRUD operations
 - 4.3.3. Persistence unit configuration
- 4.4 Docker Architecture: Images, Containers, Volumes, Networks
- 4.5 Kubernetes Basics: Pods, Deployments, Services, Ingress
- 4.6 Container Orchestration

UNIT 5 – Cloud Platform Capabilities, Security & Observability

- 5.1 Cloud Platform Capabilities for Microservices
 - 5.1.1 Compute Services - AWS EC2, Elastic Beanstalk
 - 5.1.2. Storage Services - AWS S3
 - 5.1.3. Database Services - AWS RDS, DynamoDB
- 5.2 Cloud Security for Distributed Microservices
 - 5.2.1. Security challenges in distributed systems

SARVAJANIK UNIVERSITY
FACULTY OF COMPUTER SCIENCE AND APPLICATIONS
M.Sc. (Information Technology)
Semester - I and Semester - II

	5.2.2. Authentication vs Authorization 5.2.3. JSON Web Tokens – structure, flow & usage 5.3 Cloud Observability and Monitoring 5.3.1. Importance of observability in MSA 5.3.2. Three pillars: Logging, Metrics, Tracing 5.3.3. Metrics scraping and Visualization Unit 6 - Emerging Trends in Cloud Computing 6.1 Edge Computing & Fog Computing 6.2 Quantum Cloud Computing 6.3 AI/ML in Cloud - AWS Sage maker 6.4 Green Cloud Computing & Sustainability
Text and Reference Literature	1. Cloud Computing: Principles and Paradigms - R. Buyya et al-Wiley 2010 2. Cloud Computing Bible - Sosinsky - Wiley - India,2011 3. Cloud Computing Second Edition Dr. Kumar Saurabh - Wiley - India, 2012 4. Building Microservices Paperbackby Sam Newman, SPD Press, 2017 5. Microservices for Java EE Architects: Addendum for The Java EE Architect's Handbook by Derek C. Ashmore, 2017 6. Kubernetes Microservices with Docker by Deepak Vohra,Apress Publication, 2018 7. Docker Quick Start Guide: Learn Docker like a boss, and finally own your applications by Earl Waud, PACKT publications, 2018 8. Apache ZooKeeper Essentials by Saurav Haloi, PACKT publications, 2015 9. Hazelcast A Complete Guide - 2019 Edition by Gerardus Blokdyk publication: 5STARCooks, 2019 10. Microservices Patterns: With examples in Java by Chris Richardson, Publisher: Manning Publications, 2018 11. Microservices and Containers 1st Edition by Parminder Singh, Kocher Publisher - Addison-Wesley Professional, 2018 12. Hands-On Microservices with Kubernetes: Build, deploy, and manage scalable microservices on Kubernetes, by Gigi Sayfan, Packt Publications

SARVAJANIK UNIVERSITY
FACULTY OF COMPUTER SCIENCE AND APPLICATIONS
M.Sc. (Information Technology)
Semester - I and Semester - II

Name of College: Shree Ramkrishna Institute of Computer Education and Applied Sciences						
Faculty	Computer Science and Applications		Program		M.Sc. (Information Technology)	
Year	2026-27		Version		3	
Semester	2		Effective From		June 2026	
Course Code	MSCS31204	Course Name	DSC: Advanced Cloud Programming (PR)			
Teaching Scheme				Examination Scheme		
Credits	Lecture (L)	Tutorial (T)	Practical (P)	Continuous and Comprehensive Evaluation (CCE)	Semester end examinations (SEE)	Total
1	0	0	2	25	25	50

Purpose of Course	The purpose of course is to establish the foundation of cloud-native application development using micro service architecture.
Course Objective	◆ To understand the evolution of software architectures and develop cloud-native applications using microservices. ◆ To design scalable microservices with REST APIs, business logic layers, and data persistence techniques.
Pre-requisite	Fundamental knowledge of software engineering, programming and networking
Course Outcome	After completion of the course, a student will be able to implement microservices with REST APIs, business logic, and persistence layers and also be able to use Docker containerization to manage Microservices.
Course Content	The students will be required to carry out practical on Microservices and various cloud services concepts.
Practical	List of practical will be prepared at the beginning of each semester
Reference Books	As per paper DSC: Advanced Cloud Programming (Th)

SARVAJANIK UNIVERSITY
FACULTY OF COMPUTER SCIENCE AND APPLICATIONS
M.Sc. (Information Technology)
Semester - I and Semester - II

Name of College: Shree Ramkrishna Institute of Computer Education and Applied Sciences						
Faculty	Computer Science and Applications		Program		M.Sc. (Information Technology)	
Year	2026-27		Version		3	
Semester	2		Effective From		June 2026	
Course Code	MSCS31205	Course Name	DSC: Advanced Mobile Application Development (TH)			
Teaching Scheme				Examination Scheme		
Credits	Lecture (L)	Tutorial (T)	Practical (P)	Continuous and Comprehensive Evaluation (CCE)	Semester end examinations (SEE)	Total
3	3	0	-	50	50	100

Purpose of Course	To introduce the most demanding Android open-source technology
Course Objective	- Understand Android architecture, tools, application components, and project structure. - Design intuitive mobile interfaces using layouts, widgets, navigation, and dialogs. - Implement data handling through Shared Preferences, SQLite, Firebase, and APIs. - Integrate device hardware features and services like camera, audio/video, location, and notifications. - Build, package, and deploy Android applications for real-world release.
Pre-requisite	Fundamentals of Java programming, web technologies and fundamentals related to mobile OS.
Course Outcome	Students will know about Android, a widely used mobile OS and open-source platform, and its concepts. Various features of Android, like Application Design Essentials, User Interface Design Essentials, Use of Common Android APIs, data storage using SQLite and Firebase, and deploying Android applications
Course Content	Unit 1: Introduction to Android and its Tool chain 1.1 Architecture of Android 1.2 Android Development Tools (SDK, Emulator, ADB, Logcat) 1.3 Components of Android Application 1.4 Android Project Structure (Manifest, Resources, Gradle) 1.5 Fundamental of Activities and Fragments and their lifecycle Unit 2: User Interface and Navigation 2.1 Android Layouts 2.2 Basic UI Elements – EditText, TextView, Button, RadioButton, CheckBox,

SARVAJANIK UNIVERSITY
FACULTY OF COMPUTER SCIENCE AND APPLICATIONS
M.Sc. (Information Technology)
Semester - I and Semester - II

	Spinner, ImageView 2.3 Views & Widgets- ScrollView, WebView, ProgressBar 2.4 ListView, RecyclerView 2.5 Dialogs - AlertDialog, DatePicker, TimePicker 2.6 Intents 2.7 Menus - Options Menu, Context Menu, Popup Menu 2.8 Navigation - TabLayout with ViewPager2, Bottom Navigation Unit-3: Data Handling and Storage in Android applications 3.1 Shared Preferences: Purpose and Modes 3.2 Reading and Writing Data to and from Shared Preferences 3.3 SQLite Database and SQLiteOpenHelper 3.4 SQLite Database Operations (ExecSQL, RawQuery, Insert, Update, Delete) 3.5 Managing internal and external storage - Files and Directories 3.6 Introduction to Firebase 3.7 Firebase – Environment Setup 3.8 Firebase Authentication 3.9 Writing and Reading Data to Firebase Unit 4: Sensors and Device Interaction in Android 4.1 Introduction to Sensors 4.2 Types of Sensors (Motion, Position, Environmental) 4.3 Sensor Framework 4.4 Working with Motion Sensor (Accelerometer, Gyroscope, Device orientation) 4.5 Sensor-based applications 4.5.1 Displaying accelerometer values (X, Y, Z) 4.5.2 Shake Detection 4.5.3 Device Orientation Detection (Portrait / Landscape) Unit 5: API Integration 5.1 Introduction to APIs 5.2 API Integration in Android – GET, POST, PUT, DELETE 5.3 JSON Handling – Parsing & Serialization (JSONObject, JSONArray) 5.4 Handling API Responses and Error Management Unit 6: Device Capabilities, Notifications & Deployment 6.1 Runtime Permissions 6.2 Geocoding and reverse Geocoding 6.3 Audio, Video and Using the Camera 6.3.1. Playing and recording Audio and Video 6.3.2. Working with the Camera 6.4 Push Notification 6.5 Alarm Manager for scheduled events
--	---

SARVAJANIK UNIVERSITY
FACULTY OF COMPUTER SCIENCE AND APPLICATIONS
M.Sc. (Information Technology)
Semester - I and Semester - II

	6.6 Application Deployment - APK/AAB, signing & release flow
Text and Reference Literature	1. Professional Android 4 Application Development, Reto Meier, Wrox Publication, 2015 2. Android for Programmers: An App-Driven Approach, Deitel, Deitel, Deitel, and Morgano, Pearson Publication-2012 3. Android Programming Unleashed, Harwani, Pearson Publication-2013 4. Professional Android Programming-with MONO McClure, Blevins, Croft, Dick, and Hardy, Wiley India-2012 5. Android application development for Java programmer, James C Sheusi, Cenage Learning-2013 6. Android Essentials, Chris Haseman, Apress Publication, 2009 7. Beginning Android, Mark L Murphy, Wiley India Pvt Ltd, 2009 8. Pro Android, Sayed Y Hashimi and Satya Komatineni, Wiley India Pvt Ltd, Apress-2009

SARVAJANIK UNIVERSITY
FACULTY OF COMPUTER SCIENCE AND APPLICATIONS
M.Sc. (Information Technology)
Semester - I and Semester - II

Name of College: Shree Ramkrishna Institute of Computer Education and Applied Sciences						
Faculty	Computer Science and Applications		Program	M.Sc. (Information Technology)		
Year	2026-27		Version	3		
Semester	2		Effective From	June 2026		
Course Code	MSCS31206	Course Name	DSC: Advanced Mobile Application Development (PR)			
Teaching Scheme				Examination Scheme		
Credits	Lecture (L)	Tutorial (T)	Practical (P)	Continuous and Comprehensive Evaluation (CCE)	Semester end examinations (SEE)	Total
1	0	0	2	25	25	50

Purpose of Course	To introduce the most demanding and developing mobile app technology. Fundamentals of Android open-source technology
Course Objective	- To make students understand the fundamentals of mobile app technology. - To make students understand various built-in features of Android. - To make students understand the Android design essentials. - To make students understand Android user interface design basics.
Pre-requisite	Fundamentals of Java programming, web technologies and fundamentals related to mobile OS
Course Outcome	Students will know about Android, a widely used mobile OS and open-source platform, and its concepts. Various features of Android, like Application Design Essentials, User Interface Design Essentials, Use of Common Android APIs, data storage using SQLite and Firebase, and deploying Android applications.
Course Content	The students will be required to carry out a practical on Android using the methods and tools discussed in their theory.
Practical	A list of practicals will be prepared at the beginning of each semester
Reference Books	As per paper DSC - Advanced Mobile Application Development (Th)

SARVAJANIK UNIVERSITY
FACULTY OF COMPUTER SCIENCE AND APPLICATIONS
M.Sc. (Information Technology)
Semester - I and Semester - II

Name of College: Shree Ramkrishna Institute of Computer Education and Applied Sciences						
Faculty	Computer Science and Applications		Program	M.Sc. (Information Technology)		
Year	2026-27		Version	3		
Semester	2		Effective From	June 2026		
Course Code	MSCS31207	Course Name	DSC: Software Engineering Lab – II / Internship / OJT			
Teaching Scheme				Examination Scheme		
Credits	Lecture (L)	Tutorial (T)	Practical (P)	Continuous and Comprehensive Evaluation (CCE)	Semester end examinations (SEE)	Total
4	0	0	8	50	50	100

Purpose of Course	To provide hands-on experience in analysing real-world problems and systematically developing appropriate computing solutions through requirement analysis, modelling, design, development, testing, integration, and deployment/evaluation. The course enables students to apply suitable software engineering, data, artificial intelligence, security, and system/architecture concepts according to the nature of the selected problem.
Course Objective	· To analyse real-world problems and identify their requirements, scope, constraints, and objectives. · To select and justify appropriate models, methodologies, frameworks, architectures, and tools for a given problem. · To design requirements, data, interfaces, system components, and behavioural models using appropriate modelling techniques. · To develop and configure basic and detailed solutions using suitable programming languages, software platforms, tools, and frameworks. · To apply appropriate testing, validation, evaluation, and performance assessment techniques to the developed solution. · To integrate, package, and deploy/implement the developed solution and evaluate its system-level functionality.
Pre-requisite	Students should have basic knowledge of programming fundamentals and problem-solving techniques. fundamentals of database and data management, basic software engineering and system development concepts, basic knowledge of computer networks and operating systems, fundamental concepts of data analysis, artificial intelligence, or security, as applicable to the selected problem. Familiarity with basic development and modelling tools.
Course Outcome	After successful completion of the course, students will be able to: • Analyse a real-world problem and identify its requirements, scope, constraints, and objectives.

SARVAJANIK UNIVERSITY
FACULTY OF COMPUTER SCIENCE AND APPLICATIONS
M.Sc. (Information Technology)
Semester - I and Semester - II

	• Select and justify suitable models, methodologies, frameworks, architectures, and techniques for solving a given problem. • Design requirements, data, interfaces, system components, and behavioural models using appropriate tools and techniques. • Develop and configure appropriate software solutions, computational models, AI models, security solutions, or system architectures based on the identified requirements. • Perform unit testing, model validation, system evaluation, integration testing, and other appropriate testing techniques. • Integrate, implement, and evaluate the developed solution based on relevant functional, performance, usability, accuracy, security, or other quality requirements.
Course Content	Practical 1: Problem Statement and Process Selection (4 hrs) Aim: To analyse a real-world problem statement and select a suitable model with proper justification. Description: a) Define a requirement/ problem (e.g., Online Library Management System, Mover Recommendation System, Security detection and protection, cloud or network system &/or architecture etc.). b) Document the scope, operational constraints, and objectives. Choose appropriate models (SDLC models, ML Models, Security models, frameworks, architectures etc) and justify why a specific approach fits the chosen problem. Practical 2: Requirement Specification (8 hrs) Aim: To prepare a complete Requirement document for the chosen problem. Description: a) Prepare detailed requirement list, gaps, model limitations and prepare a complete requirement document (Data Flow Diagrams, ER Diagrams, Use Case Diagrams, Class Diagrams, comparative study of existing frameworks, models and methods/techniques etc.) using appropriate software tools. Practical 3: Data Design and Interface Design (8 hrs) Aim: To design data and Human interaction appropriate to problem and requirements. Description: a) Design Database (tables, relationships, constraints, keys etc.) /Data source (dataset, attributes, features, EDA etc.) using appropriate software tools. b) Design interface (Inputs, Outputs, Dashboards, Data Visualizations, interaction panels etc.) using appropriate Software Tools. Practical 4: Designing Interaction and Behavioural Models (8 hrs) Aim: To design various models indicating system component interactions and behaviour appropriate for the problem and requirements. Description: Design various system diagrams (like Sequence Diagram, Activity Diagram, StateChart Diagram, Deployment Diagrams, System Flowchart etc.) / component interactions in models/framework/system/architecture using appropriate software Tools. Practical 5: Basic Solution design &/or development Software Coding of core functionality/ Model Development/System/Architecture Design (20-24 hrs)

SARVAJANIK UNIVERSITY
FACULTY OF COMPUTER SCIENCE AND APPLICATIONS
M.Sc. (Information Technology)
Semester - I and Semester - II

	Aim: Design &/or Development &; testing (of Core functionalities/Admin Modules or Training, evaluation &; configuration of Core AI / Security model or system &/or architecture design etc.) of the basic/initial solution appropriate for the problem and requirements. Description: a) Design &/or Development (of Core functionalities/Admin Modules or Training, evaluation &; configuration of Core AI / Security model or system &/or architecture design etc.) of the required solution appropriate for the problem and requirements using appropriate software platforms, tools, programming &/or scripting language etc. b) Perform Unit Testing/model validation/architecture evaluation etc. of the basic/initial solution. Practical 6: Detailed Solution design &/or development Software Coding of core functionality/ Model Development/System/Architecture Design) (40-45 hrs) Aim: Design &/or Development &; testing/evaluation (of Core functionalities/Admin Modules or Training, evaluation &; configuration of Core AI / Security model or system &/or architecture design etc.) of the detailed solution appropriate for the problem and requirements. Description: a) Design &/or Development & testing (of Detailed functionalities/Admin Modules or Training, evaluation &; configuration of detailed AI / Security model or system &/or architecture design etc.) of the required solution appropriate for the problem and requirements. using appropriate software platforms, tools, programming &/or scripting language etc. b) Perform Unit Testing/model validation/architecture evaluation etc. of the detailed solutions. Practical 7: Integration and Detailed Testing/evaluation (8-12 hrs.) Aim: To integrate already designed &/or developed components (e.g. software modules, models, framework, system, architecture etc.) and perform integration testing / evaluation. Description: a) Integrate already designed &/or developed components (e.g. software modules, models, framework, system, architecture etc.). b) Perform Integration Testing of already designed &/or developed components (e.g. software modules, models, framework, system, architecture etc.). Practical 8: Packaging, Deployment and system level testing (8-12 hrs.) Aim: To deploy/host/upload/ implement the developed solution (software/ webapp /mobile app, model, framework, architecture, system etc) and test the deployment. Description: a) To deploy/host/upload/ implement the developed solution (software/ webapp /mobile app, model, framework, architecture, system etc) in user/client/ customer environment. b) Perform system and User - acceptance testings/ evaluation (like security testing, load testing, stress testing, usability testing, accuracy, precision etc.)
--	--

SARVAJANIK UNIVERSITY
FACULTY OF COMPUTER SCIENCE AND APPLICATIONS
M.Sc. (Information Technology)
Semester - I and Semester - II

Name of College: Shree Ramkrishna Institute of Computer Education and Applied Sciences						
Faculty	Computer Science and Applications		Program	M.Sc. (Information Technology)		
Year	2026-27		Version	3		
Semester	2		Effective From	June 2026		
Course Code	MSCS32201	Course Name	DSE: Cyber Security and Forensic - II			
Teaching Scheme				Examination Scheme		
Credits	Lecture (L)	Tutorial (T)	Practical (P)	Continuous and Comprehensive Evaluation (CCE)	Semester end examinations (SEE)	Total
4	4	0	-	50	50	100

Purpose of Course	Conceptualize the students with the concepts of computer forensics methodology
Course Objective	Familiarization with different objectives associated with different forensic techniques. Different stages of the forensic investigation process life cycle can focus on a broad idea of the forensic process.
Pre-requisite	Basic knowledge of Cyber security, Information security, computer network, operating systems and hardware mechanism of IT peripherals
Course Outcome	1. Explain the five core functions of NIST CSF 2.0 (Identify, Protect, Detect, Respond, Recover, Govern) and their interdependencies 2. Map adversary tactics to the Cyber Kill Chain model (Reconnaissance, Weaponization, Delivery, Exploitation, Installation, Command & Control, Actions on Objectives) 3. Distinguish between MITRE ATT&CK Tactics, Techniques, and Sub techniques for threat modelling 4. Explain the Shared Responsibility Model and distinguish provider vs. customer duties across AWS, Azure, GCP 5. Describe IAM identity types (users, roles, service accounts) and policy enforcement mechanisms 6. Compare encryption strategies (at-rest, in-transit, key management) and their trade-offs 7. Explain OWASP Top 10 2021 vulnerabilities (Injection, Broken Authentication, Sensitive Data Exposure, XML Injection, Broken Access Control, etc.) with attack scenarios. 8. Compare web, mobile (iOS/Android), and API security considerations 9. Discuss secure SDLC and DevSecOps integration points 10. Define CVE, CVSS, CWE taxonomies and explain scoring methodology 11. Describe vulnerability scanning methodologies and interpret scan results 12. Prioritize vulnerabilities using risk-based approach (CVSS + business context) 13. Explain NIST Incident Response lifecycle phases (Preparation, Detection, Containment, Eradication, Recovery) 14. Analyze phishing emails to identify IoCs; validate authenticity using

SARVAJANIK UNIVERSITY
FACULTY OF COMPUTER SCIENCE AND APPLICATIONS
M.Sc. (Information Technology)
Semester - I and Semester - II

	SPF/DKIM/DMARC 15. Design IR playbooks for common incident types 16. Explain forensic tool categories (disk, memory, network analysis) and Their use cases 17. Describe chain-of-custody requirements and forensic integrity Principles (hashing, write-blockers) 18. Explain Windows Registry structure, hives, and key artifacts for forensic investigation 19. Interpret Event Logs to identify user activity, privilege escalation, Malware execution 20. Describe Windows file system artifacts (NTFS, MFT, USN Journal) useful in forensics.
Course Content	Unit 1: Cyber Defense Strategy & Frameworks 1.1 NIST Cybersecurity Framework (CSF)2.0 1.1.1 Six core functions (Govern, Identify, Protect, Detect, Respond, Recover) 1.1.2 Categories and Sub-categories 1.1.3 Implementation Tiers (Partial, Risk-Informed, Repeatable, Adaptive) 1.1.4 Profiles (Current vs. Target) 1.2 Cyber Kill Chain(Lockheed Martin) 1.2.1 7 stages of attack lifecycle 1.2.2 Detection opportunities at each stage 1.2.3 Defensive implications 1.2.4 Case studies (APT28, APT29) 1.3 MITRE ATT&CK Framework 1.3.1 Tactics vs. Techniques vs. Sub-techniques 1.3.2 Threat Groups and Software Mapping 1.3.3 Mitigation strategies 1.3.4 Navigator tool usage 1.4 Defense-in-Depth Principles 1.4.1 Layered security model 1.4.2 Zero Trust Architecture principles 1.4.3 Risk acceptance vs. mitigation 1.4.4 Trade-offs between usability and security 1.5 Hands-on session based on the above topics Unit 2: Cloud Security Fundamentals 2.1 Shared Responsibility Model 2.1.1 Provider responsibilities (AWS/Azure/GCP) 2.1.2 Customer responsibilities 2.1.3 Shared services (IAM, encryption) 2.1.4 Common misunderstandings 2.2 Cloud IAM (Identity & Access Management) 2.2.1 Concepts: Authentication, Authorization, Accounting 2.2.2 Identity federation and SSO 2.2.3 Role-Based Access Control (RBAC) 2.2.4 Policy languages (AWS IAM, Azure RBAC) 2.2.5 Privilege escalation vectors 2.3 Data Protection: Encryption & Secrets 2.3.1 Encryption at rest (KMS, key rotation) 2.3.2 Encryption in transit (TLS, VPN)

SARVAJANIK UNIVERSITY
FACULTY OF COMPUTER SCIENCE AND APPLICATIONS
M.Sc. (Information Technology)
Semester - I and Semester - II

	2.3.3 Key management best practices 2.3.4 Secrets rotation (passwords, API keys) 2.3.5 Data classification 2.4 Cloud Network Security 2.4.1 VPC architecture and subnets 2.4.2 Security Groups vs. Network ACLs 2.4.3 Public vs. private connectivity 2.4.4 DDoS protection 2.4.5 WAF principles 2.5 Hands-on session based on the above topics Unit 3: Application & Mobile Security 3.1 OWASP Top 10(Web) 3.1.1 Injection (SQL, NoSQL, OS command) 3.1.2 Broken Authentication & Session Management 3.1.3 Sensitive Data Exposure 3.1.4 XML External Entities (XXE) 3.1.5 Broken Access Control (BOLA) 3.1.6 Security Misconfiguration 3.1.7 Cross-Site Scripting (XSS: Stored, Reflected, DOM) 3.1.8 Insecure Deserialization 3.1.9 Using Components with Known Vulnerabilities 3.1.10 Insufficient Logging & Monitoring 3.2 OWASP Mobile Top10 3.2.1 Improper Platform Usage (iOS/Android APIs) 3.2.2 Insecure Data Storage 3.2.3 Insecure Communication 3.2.4 Insecure Authentication 3.2.5 Insufficient Cryptography 3.2.6 Reverse Engineering & Tampering 3.2.7 Extraneous Functionality 3.2.8 Code Quality Issues 3.2.9 Memory Management 3.3 API Security 3.3.1 Authorization (BOLA, excessive data exposure) 3.3.2 Rate limiting & DDoS protection 3.3.3 Input validation 3.3.4 Common API vulnerabilities 3.4 Secure SDLC /DevSecOps 3.4.1 Threat modeling (STRIDE) 3.4.2 Secure code review practices 3.4.3 SAST vs. DAST tools 3.4.4 Dependency scanning 3.5 Hands-on session based on the above topics Unit 4: Vulnerability Management 4.1 CVE (Common Vulnerabilities & Exposures) 4.1.1 Numbering scheme and structure 4.1.2 CVE distribution channels (NVD, vendor advisories) 4.1.3 Using CVE databases for threat intelligence 4.2 CVSS (Common Vulnerability Scoring System) 4.2.1 CVSS v3.1 metrics (Base, Temporal, Environmental)
--	--

SARVAJANIK UNIVERSITY
FACULTY OF COMPUTER SCIENCE AND APPLICATIONS
M.Sc. (Information Technology)
Semester - I and Semester - II

	4.2.2 Severity ratings (None, Low, Medium, High, Critical) 4.2.3 Scoring calculations and limitations 4.2.4 Context-dependent scoring adjustments 4.3 CWE (Common Weakness Enumeration) 4.3.1 Weakness hierarchy and categories 4.3.2 Mapping CWE to OWASP Top 10 4.3.3 CWE Top 25 and business impact 4.3.4 Using CWE for defense planning 4.4 Vulnerability Scanning Tools & Workflows 4.4.1 Scanner types: Agent-based, network-based, application-based 4.4.2 False positives and false negatives 4.4.3 Scan scope definition (IP ranges, ports) 4.4.4 Results interpretation and prioritization 4.5 Vulnerability Lifecycle Management 4.5.1 Detection → Assessment → Prioritization → Remediation → Verification 4.5.2 Patch management integration 4.5.3 Risk acceptance documentation 4.6 Hands-on session based on the above topics Unit 5: Social Engineering & Incident Response 5.1 Phishing Analysis & Detection 5.1.1 Email header structure (From, To, Reply-To, Return-Path) 5.1.2 Phishing vectors (credential harvesting, malware delivery) 5.1.3 Social engineering psychology 5.1.4 URL analysis (domain registration, IP geolocation) 5.1.5 Email authentication bypasses 5.2 Email Authentication (SPF/DKIM/DMARC) 5.2.1 SPF record syntax and policy evaluation 5.2.2 DKIM signing and verification 5.2.3 DMARC alignment and failure handling 5.3 NIST IR Lifecycle 5.3.1 Phase 1: Preparation (tools, playbooks, training) 5.3.2 Phase 2: Detection & Analysis (alert triage, severity assessment) 5.3.3 Phase 3: Containment (short-term, long-term) 5.3.4 Phase 4: Eradication (root cause removal) 5.3.5 Phase 5: Recovery (system restoration, validation) 5.3.6 Post-incident review (lessons learned) 5.4 Malware & Indicators of Compromise (IoCs) 5.4.1 File hashes (MD5, SHA-1, SHA-256) 5.4.2 Network IoCs (IPs, domains, URLs) 5.4.3 Host IoCs (registry keys, process names) 5.4.4 Threat intelligence feeds (VirusTotal) 5.4.5 False indicators and context 5.5 Hands-on session based on the above topics Unit 6: Forensic Investigation Tools 6.1 Forensic Fundamentals & Tools Categories 6.1.1 Disk forensics (file systems, unallocated space) 6.1.2 Memory forensics (volatile data) 6.1.3 Network forensics (packet capture) 6.1.4 Log analysis (SIEM, event logs) 6.1.5 Chain of custody and evidence integrity
--	--

SARVAJANIK UNIVERSITY
FACULTY OF COMPUTER SCIENCE AND APPLICATIONS
M.Sc. (Information Technology)
Semester - I and Semester - II

	6.2 Splunk Log Analysis 6.2.1 Data ingestion methods (forwarders, HTTP Event Collector) 6.2.2 Search Processing Language (SPL) fundamentals 6.2.3 Queries for threat hunting (authentication, file access) 6.2.4 Dashboards and alerting 6.2.5 Log retention and compliance 6.3 Autopsy Disk Analysis 6.3.1 Filesystem analysis (NTFS, FAT, ext4) 6.3.2 Timeline generation 6.3.3 Artifact extraction (files, registry, media) 6.3.4 Deleted item recovery 6.3.5 Keyword search and tagging 6.4 Network Miner Packet Analysis 6.4.1 PCAP file parsing 6.4.2 Network reconstruction 6.4.3 File extraction from traffic 6.4.4 Metadata analysis 6.4.5 Anomaly detection 6.5 Hands-on session based on the above topics Unit 7: Windows Systems Forensics 7.1 Volatile Data Collection 7.1.1 RAM acquisition methods (FDPRO, FTK Imager, DumpIt) 7.1.2 Memory forensics tools (Volatility) 7.1.3 Process and network connection extraction 7.1.4 Rootkit detection in memory 7.1.5 Live response considerations 7.2 Registry Analysis 7.2.1 Hive structure (HKLM, HKCU, SAM, SECURITY) 7.2.2 User profiles and SID mapping 7.2.3 Run keys and persistence mechanisms 7.2.4 Network shares and mounted devices 7.2.5 Installed software and drivers 7.3 Browser Forensics 7.3.1 Chrome/Firefox/Edge artifact locations 7.3.2 History, bookmarks, cache analysis 7.3.3 Download history and timestamps 7.3.4 Cookie exfiltration indicators 7.3.5 Autocomplete data 7.4 Event Log Analysis 7.4.1 Security log events (login, privilege escalation) 7.4.2 System log events (services, driver loads) 7.4.3 Application log events (software execution) 7.4.4 Event IDs relevant to security (4624, 4688, 7045) 7.4.5 Log forwarding and timestamping issues 7.5 Hands-on session based on the above topics Unit 8: Cloud forensics 8.1 Introduction to cloud forensic 8.2 Challenges faced by CSP due to international law enforcement 8.3 Cloud storage forensic framework 8.4 Google drive and Drop box analysis
--	---

SARVAJANIK UNIVERSITY
FACULTY OF COMPUTER SCIENCE AND APPLICATIONS
M.Sc. (Information Technology)
Semester - I and Semester - II

	8.5 Case study
Text and Reference Literature	Frameworks & Standards 1.1 NIST Cybersecurity Framework (CSF) 2.0 1.2 NIST Incident Response Guide (SP 800-61) 1.3 MITRE ATT & CK Framework 1.4 Cyber Kill Chain (Lockheed Martin) 1.5 OWASP Top 10 (Web & Mobile) 1.6 ISO 27001/27002 (referenced in institutional context) 1.7 NIST SP 800-53: Security and Privacy Controls 2. Recommended Textbooks 2.1 Windows Forensic Analysis, by Harlan Carvey 2.2 Network Forensics: Tracking Hackers Through Cyberspace, by Sherri Davidoff & Jonathan Ham 2.3 The Incident Response Playbook, by Erica Friedberg 2.4 Practical Vulnerability Management, by Abhishek Arora 2.5 Cloud Security: A Comprehensive Beginner's Guide, by Robert Slade 3. Free/Open-Source Tools Endorsed 3.1 Splunk Free Edition 3.2 Autopsy (The Sleuth Kit) 3.3 Wireshark 3.4 Nessus Essentials 3.5 OpenVAS 3.6 Burp Suite Community 3.7 MobSF (Mobile Security Framework) 4. External References 4.1 SANS Cyber Aces (free labs) 4.2 PortSwigger Web Security Academy 4.3 HackTheBox (CTF platform) 4.4 TryHackMe (interactive labs) 4.5 YouTube: NetworkChuck, John Hammond, LiveOverflow 5. Computer forensic by John R. Vacca, Firewall media, 6. Computer forensic, Nina godbole, sunit belapure, wiley 7. Wireless crime and forensic investigation, Gregory kipper, Auerbach publication (Tallor and Francis group) 8. Computer forensic and cyber crime 3rd edition, by Marjie Britz, pearson 9. Computer forensic investigation network intrusion and cyber crime EC Council, course technology

SARVAJANIK UNIVERSITY
FACULTY OF COMPUTER SCIENCE AND APPLICATIONS
M.Sc. (Information Technology)
Semester - I and Semester - II

Name of College: Shree Ramkrishna Institute of Computer Education and Applied Sciences						
Faculty	Computer Science and Applications		Program	M.Sc. (Information Technology)		
Year	2026-27		Version	1		
Semester	7		Effective From	June 2026		
Course Code	MSCS32202	Course Name	Major: Generative AI (TH)			
Teaching Scheme				Examination Scheme		
Credits	Lecture (L)	Tutorial (T)	Practical (P)	Continuous and Comprehensive Evaluation (CCE)	Semester end examinations (SEE)	Total
3	3	0	-	50	50	100

Purpose of Course	The purpose of the course is to equip learners with the knowledge and skills needed to understand, build, and apply generative AI models. Learners are expected to be capable of building generative AI solutions, evaluating their performance, and applying them to solve practical problems creatively and responsibly.
Course Objective	1. To learn about generative artificial intelligence 2. To learn and implement different types of Gen AI Technology in the programming and content generation domain.
Pre-requisite	Fundamental concepts of Programming, Computerized Tools and Software.
Course Outcome	After completion of this course, the student will be capable to develop generative AI based solutions.
Course Content	Unit 1. Defining New Age AI 1.1 Artificial Narrow Intelligence 1.2 Artificial General Intelligence 1.3 Artificial Super Intelligence 1.4 Discriminative Vs Generative AI 1.5 Applications of Generative AI Unit 2. Data Representation in Gen AI 2.1 Uses of Generative AI - Text, Images, Video, Music, Code, Voices 2.2 Encoding, Embedding 2.3 Text Embedding

SARVAJANIK UNIVERSITY
FACULTY OF COMPUTER SCIENCE AND APPLICATIONS
M.Sc. (Information Technology)
Semester - I and Semester - II

	2.4 Image Embedding 2.5 Video Embedding 2.6 Key Technologies behind Gen AI - GAN, VAE, Transformers Unit 3 LLM and Prompt Engineering 3.1 Language Modelling 3.2 Large Language Models 3.3 Applications of LLMs 3.4 Introduction of lightweight LLMs and SLMs 3.5 Prompt Designing, Task Formulation using Prompts, Prompt Patterns 3.6 Prompt Tuning Techniques Unit 4 Retrieval Augmented Generation 4.1 Introduction to RAG 4.2 Vector Embeddings 4.3 Creating Knowledge Base 4.4 Store embedding in Vector DB 4.5 Query Embeddings Unit 5 Vector Embeddings & Storage 5.1 Embedding Models 5.2 Vector Databases & Indexing 5.3 Similarity Metrics - Cosine Similarity, Euclidean Distance, and Dot Product 5.4 Advanced RAG Patterns - Re-Ranking, Parent child RAG, GraphRAG Unit 6 AI Agent Frameworks 6.1 AI Agents 6.2 Introduction to Agentic AI 6.3 Open-source AI agent frameworks 6.3.1 Langchain 6.3.2 LangGraph 6.3.3 Crew AI 6.3.4 AutoGen (Microsoft) 6.3.5 BabyAGI
Text and Reference Literature	1. A First Course in Artificial Intelligence by Deepak Khemani, McGrawHill, ISBN :978-1-25-902998-1 2. Generative AI for Everyone: Deep learning, NLP, and LLMs for creative and practical applications by Karthikeyan Sabesan, Sivagamisundari, et al 3. Generative AI for Begginers Playbook by Branson Adams 4. Genrative AI for Everyone by Altaf Rehmani 5. Generative AI for Software Developers: Future-Proof Your Career with AI-Powered Development and Practical Hands-On Skills by Saurabh Shrivastava , Kamal Arora

SARVAJANIK UNIVERSITY
FACULTY OF COMPUTER SCIENCE AND APPLICATIONS
M.Sc. (Information Technology)
Semester - I and Semester - II

Name of College: Shree Ramkrishna Institute of Computer Education and Applied Sciences						
Faculty	Computer Science and Applications		Program	M.Sc. (Information Technology)		
Year	2026-27		Version	1		
Semester	7		Effective From	June 2026		
Course Code	MSCS32203	Course Name	Major: Generative AI (PR)			
Teaching Scheme				Examination Scheme		
Credits	Lecture (L)	Tutorial (T)	Practical (P)	Continuous and Comprehensive Evaluation (CCE)	Semester end examinations (SEE)	Total
1	0	0	2	25	25	50

Purpose of Course	The purpose of the course is to equip learners with the knowledge and skills needed to understand, build, and apply generative AI models. Learners are expected to be capable of building generative AI solutions, evaluating their performance, and applying them to solve practical problems creatively and responsibly.
Course Objective	1.To learn about generative artificial intelligence 2. To learn and implement different types of Gen AI Technology in the programming and content generation domain.
Pre-requisite	Fundamental concepts of Programming, Computerized Tools and Software.
Course Outcome	After completion of this course, the student will be capable to develop generative AI based solutions.
Course Content	This course covers understanding of LLM intrinsic working, vector representation of data and RAG implementation.
Practical	List of practical will be prepared at the beginning of each semester
Reference Books	As per paper Major - Generative AI(TH)

SARVAJANIK UNIVERSITY
FACULTY OF COMPUTER SCIENCE AND APPLICATIONS
M.Sc. (Information Technology)
Semester - I and Semester - II

Name of College: Shree Ramkrishna Institute of Computer Education and Applied Sciences						
Faculty	Computer Science and Applications		Program	M.Sc. (Information Technology)		
Year	2026-27		Version	3		
Semester	2		Effective From	June 2026		
Course Code	MSCS33201	Course Name	SEC: AI-Driven Development			
Teaching Scheme				Examination Scheme		
Credits	Lecture (L)	Tutorial (T)	Practical (P)	Continuous and Comprehensive Evaluation (CCE)	Semester end examinations (SEE)	Total
2	2	0	0	25	25	50

Purpose of Course	The purpose of the course is to introduce advanced ai tools to developers who can automate tasks, generate quality code, customize models, and accelerate software and application development.
Course Objective	◆ Understand the application of AI for software development work ◆ Use AI pair programming and/or code management and collaboration. ◆ Learn to use modern AI automation for handling repetitive writing, project management, and debugging so developers can focus on problem-solving and system architecture.
Pre-requisite	Programming skills in any language and Familiarity with command-line operations and AI fundamentals
Course Out come	After completion of this course, the student will be capable of developing and deploying applications with AI assistance.
Course Content	Unit 1: Foundations of AI-Driven Development 1.1 The Core Shift: From Coding to Architecting 1.2 Automated Code Generation 1.3 Introduction to Vibe Coding 1.4 AI Pair Programmer 1.5 Inline Autocomplete Plugins, AI-Native IDEs, Coding Agents Unit 2: AI Scaffolding and Templates 2.1 Stack-Aware Generation 2.2 Intelligent Templating 2.3 Prompt and Agent based project scaffolding 2.4 Layered Boundaries and Guardrails 2.5 AI code plugins and IDE plugins

SARVAJANIK UNIVERSITY
FACULTY OF COMPUTER SCIENCE AND APPLICATIONS
M.Sc. (Information Technology)
Semester - I and Semester - II

	Unit 3: Agentic Development Platform 3.1 Introduction to Google Antigravity 3.2 Antigravity CLI, Antigravity SDK and IDE 3.3 Setup, Installation and Configurations 3.4 Projects and Conversations 3.5 Project Configurations Unit 4: Agent Skills, Rules and Workflows 4.1 Agent skills 4.2 Creating Skills - skills.md file 4.3 Script Integrations 4.4 Building skill using Antigravity 4.5 Rules and Workflows 4.6 Plugins
Text and Reference Literature	1. AI Engineering: Building Applications with Foundation Models - by Chip Huyen 2. The Ultimate Guide to Google Antigravity: From Vibe Coding to Solopreneur: Build & Sell Your First App in 7 Days Without Hiring Devs (Google Antigravity Mastery Series Book 1) 3. Google Antigravity Pro: Harness Google AI to accelerate software delivery, improve code quality and scale modern AI-assisted development. (AI Coding) by Codapress Publishing