SARVAJANIK UNIVERSITY
INCLUSIVE | INTEGRATED | INNOVATIVE

Faculty of Science

Shree Ramkrishna Institute of Computer Education and Applied Sciences, Surat

M.Sc. Information Technology

Programme Subject List : Semester 2

Sem	Paper type	Paper No.	Paper Title
2	Core course	DSC-1	Mobile Application Development-1
		DSC-2	Web Programming-2
	Skill Enhancement	SEC-1	Advanced Cloud Programming
	Discipline Specific Elective	DSE	1. Cyber Security and Forensic-2
			2. Machine Learning
			3. Generative AI
	Practical	Practical	Practical and Project - 2

SARVAJANIK UNIVERSITY
INCLUSIVE | INTEGRATED | INNOVATIVE

Faculty of Science

Shree Ramkrishna Institute of Computer Education and Applied Sciences, Surat

M.Sc. Information Technology

SEMESTER- 2

Exam Scheme for M.Sc Information Technology													
Semester - II													
Sr. No	**Subject Code**	**Group**	**Subjects**		**Credit**			**Internal**				**External**	**Total**
				Hrs	**Lecture**	**Practical**	**Total**	**CCE/ Internal Practical Test and Viva Voce**	**Attendance**	**Assignment/Practical continuous evaluation**	**CCA**	**Term End**	
1	MSCS21201	DSC-3	Mobile Application Development-1	4	4		4	40	10	20	70	30	100
2	MSCS21202	DSC-4	Web Programming - 2	4	4		4	40	10	20	70	30	100
3	MSCS22201	DSE-2	Cyber Security and Forensic-2	4	4		4	40	10	20	70	30	100
	MSCS22202		Machine Learning										
	MSCS22203		Generative AI										
4	MSCS23201	SEC-2	Advanced Cloud Programming	4	4		4	40	10	20	70	30	100
5	MSCS26201	Practical	Practical and Project-2	16	-	8	8	60	20	60	140	60	200
		Total		**32**	**16**		**24**						**600**

Note: Following subjects are listed as elective subjects of the semester.

List of Elective Subjects

1. **Cyber security and Forensics**
2. **Machine Learning**
3. **Generative AI**

Name of College: Shree Ramkrishna Institute of Computer Education and Applied Sciences						
Faculty	Science		**Program**	Master of Science Information Technology		
Year	2025-26		**Version**	2.0		
Semester	2		**Effective From**	December, 2025		
Course Code	MSCS21201	**Course Name**	**Mobile Application Development – 1**			
Teaching Scheme				**Examination Scheme**		
Credits	**Lecture (L)**	**Tutorial (T)**	**Practical (P)**	**Continuous Assessments (CA)**	**Term end examinations (TEE)**	**Total**
4	4	0	0	70	30	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.
Pr-requisite	1. Pre-requisite: Fundamentals of web technologies and fundamentals related to mobile OS.
Course Out come	2 Students will have knowledge about Android, which is a widely used MobileOS and open source technology and its concepts. Various features of Android, like Application Design Essentials, User Interface Design

	Essentials, Use of Common Android APIs, data storage using SQLiteandFirebase 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 1.2.1 Android SDK and SDK Manager 1.2.2 Android Virtual Device (AVD) / Emulator 1.3.3 Device Manager (Virtual Devices / Emulator) 1.3.4 Android Debug Bridge (ADB) 1.3.5 Logcat 1.3 Components of Android Application - Activities, Services, Broadcast Receivers, Content Providers 1.4 Directory Structure of Android Application 1.4.1 AndroidManifest.xml 1.4.2 Layouts & Drawable Resources 1.4.3 Activity Java file 1.4.4 Gradle 1.5 Activity Lifecycle & Fragment Lifecycle **Unit 2 : Layout and Advanced UI Design** 2.1 Layouts- Linear , Relative , Constraint ,Frame 2.2 Basic UI Elements – EditText, TextView, Button, RadioButton, CheckBox, Spinner,Imageview 2.3 Views & Widgets- ScrollView,WebView,ProgressBar 2.4 Listview, RecyclerView 2.5 Dialogs - AlertDialog,DatePicker, TimePicker **Unit-3 : Navigating across activities** 3.1 Intents - Explicit, Implicit Intent, Intent Filters 3.2 Menus - Options Menu , Context Menu , Popup Menu 3.3 Navigation - TabLayout with ViewPager2,Bottom Navigation **Unit-4: Using Shared Preferences** 4.1 Purpose of Shared Preferences 4.2 Shared Preference Modes 4.3 Writing to shared Preferences 4.4 Methods of the editor class 4.5 Reading from Shared Preferences **Unit-5: Preserving and Saving data in Local Database** 5.1 Introduction to SQLite 5.2 SqliteOpenHelper Class 5.3 SQlite Methods - ExecSQL, Rawquery, Insert, Update, Delete **Unit - 6 : Firebase & Cloud Technologies** 6.1 Introduction to Firebase 6.2 Firebase – Environment Setup 6.3 Writing and Read Data to Firebase **Unit-7 : API Integration**

	7.1 Introduction to APIs 7.2 API Integration in Android – GET, POST, PUT, DELETE 7.3 JSON Handling – Parsing & Serialization (JSON Object, JSONArray) **Unit 8 : Device Capabilities, Notifications & Deployment** 8.1 Geocoding and reverse Geocoding 8.2 Audio, Video and Using the Camera 8.2.1. Playing and recording Audio and Video 8.2.2. Working with the Camera 8.3 Push Notification 8.4 Alarm Manager for scheduled events 8.5 Application Deployment - APK/AAB, signing & release flow

Name of College: Shree Ramkrishna Institute of Computer Education and Applied Sciences						
Faculty	Science		**Program**	Master of Science Information Technology		
Year	2025		**Version**	2.0		
Semester	2		**Effective From**	December, 2025		
Course Code	MSCS21202	**Course Name**	**Web Programming - 2**			
Teaching Scheme				**Examination Scheme**		
Credits	**Lecture (L)**	**Tutorial (T)**	**Practical (P)**	**Continuous Assessments (CCE)**	**Semester end examinations (SEE)**	**Total**
4	4	-	-	70	30	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- Global and Process objects, buffers, Typed arrays and Strings, Streams, Callbacks and Asynchronous, Event Handling- Event Queue, Event Emitter, Event Loop and Timers, Nested Callback 1.2 Exception Handling. 1.3 REPL Terminal **Unit 2 : Node Modules and Node Package Manager (NPM)**

	2.1 Overview of Node Module System 2.2 Overview of Node Package Manager 2.3 Overview of Node Version Manager 2.4 Creating and Publishing Node Modules 2.5 axios, bcrypt, jsonwebtokens, multer 2.6 nodemon, dotenv **Unit 3 : Node with the Local System and the Web** 3.1 Streams and Pipes 3.2 Node and the File System- The fs.Stats class, The File System Watcher, File Read and Write, Directory access and Maintenance, File Streams 3.3 Resource Access with Path **Unit 4 : Node and Web Application** 4.1 Using Apache to proxy Node application 4.2 Routing in NodeJS 4.3 Routing and Callback function 4.4 Modern Deployment Basics: PM2 (concept only) 4.5 Hosting overview: Render, Railway, Vercel 4.6 What is NGINX (concept only, not practical) **Unit 5 : NodeJS, MongoDB and MapReduce** 5.1 MongoDB Aggregation Pipeline 5.2 Grouping, Sorting, Filtering with Aggregation 5.3Indexing basics (single field, compound index) **Unit 6 : Full-Stack Node development** 6.1The Express Application Framework 6.2 Express Supportive Modules,Body-parser, Method Override 6.3 Basic EJS, Folder structure (routes/controllers/models) 6.4 Integrating NodeJS and MongoDB 6.5 Nodes JS and REST APIs **Unit 7 : NodeJS Application Management** 7.1 Project Structure, MVC architecture ,Folder structuring for APIs 7.2 Controllers, Routes, Services, Models 7.3 Environment Variables, Using .env file,Config management 7.4 Modern JavaScript in Node.js - ES Modules (import/export) 7.5 Async/Await , Built-in fetch() in Node 18+ 7.6 Error Handling - Centralized error handler , Async error handling HTTP response structure **Unit 8 : Advanced NodeJS Operations** 8.1 Basic Authentication -JWT basics,bcrypt hashing ,Protected routes 8.2 Postman / Thunder Client - API testing, Status codes ,Sending headers/body/params 8.3 NodeJS in Real-time -WebSocket , Socket.IO
Text and Reference	1. Learning Node Moving to the server side Shelley Powers O'Relly

Literature	Publication Building Node Applications with MongoDB and Backbone Mike Wilson O'Relly SPD Publication 2 GEO, CouchDB & NodeJS Mick Thompson O'Relly SPD Publication 3 Web Development with Node and Express, Ethan Brown , O'Relly Publication 4 Node.js in Action, Alex Young, Bradley Meck, Mike Cantelon, Tim Oxley , Marc Harter, T.J. Holowaychuk, Nathan Rajlich

Name of College: Shree Ramkrishna Institute of Computer Education and Applied Sciences						
Faculty	Science		**Program**	Master of Science Information Technology		
Year	1		**Version**	2.0		
Semester	2		**Effective From**	December, 2025		
Course Code	MSCS22201	**Course Name**	**Cyber Security and Forensics-2**			
Teaching Scheme				**Examination Scheme**		
Credits	**Lecture (L)**	**Tutorial (T)**	**Practical (P)**	**Continuous Assessments (CCE)**	**Semester end examinations (SEE)**	**Total**
4	4	0	0	70	30	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.
Prerequisite	Basic knowledge of Cyber security, Information security, computer network, operating systems and hardware mechanism of IT peripherals.
Course Out come	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 modeling 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 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 Description	Basic terminology associated with forensic investigation process, types of forensic technology, data recovery, evidence collection and data seizure, duplication and preservation of digital evidence, computer image verification and authentication, reconstructing past events to collect evidence. Moreover forensic investigation of computer, network, e-mail, android, i-phone, printer, scanner, pda etc. is covered as subject content.
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) 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 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 4.4.1Scanner 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 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 8.5 Case study
Reference Books	1. 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

Name of College: Shree Ramkrishna Institute of Computer Education and Applied Sciences						
Faculty	Science		**Program**	Master of Science Information Technology		
Year	2025		**Version**	2.0		
Semester	2		**Effective From**	December, 2025		
Course Code	MSCS22202	**Course Name**	**Machine Learning**			
Teaching Scheme				**Examination Scheme**		
Credits	**Lecture (L)**	**Tutorial (T)**	**Practical (P)**	**Continuous Assessments (CCE)**	**Semester end examinations (SEE)**	**Total**
4	4	-	-	70	30	100

Purpose of Course	The purpose of the course is to make students capable of implementing concepts, methods, and tools related to machine learning.
Course Objective	● To learn about the fundamentals of machine learning. ● To learn and implement different types of ML algorithms. ● To implement and evaluate various case studies of Machine Learning
Pre-requisite	Basics of Python Programming and Basics of Data Mining
Course Outcome	After completion of this course, the student will be capable of developing models and implementing predictive analytics.
]Course Content	**Unit 1: Introduction to Machine Learning** 1.1 Types of Learning 1.2 Machine Learning 1.3 Types of Problems in ML 1.4 Machine Learning Applications 1.5 New Challenges for ML **Unit 2: Supervised Learning - Regression** 2.1 Linear Regression 2.2 Polynomial Regression 2.3 Logistic Regression 2.4 Evaluation Metrics for Regression (RMSE, MSE, MAE, R^2 Score (Coefficient of Determination) 2.5 Other advanced metrics **Unit 3: Supervised Learning - Classification** 3.1 Classification: Examples and Applications NB, SVM, KNN Classifiers 3.2 Decision Trees : C4.5, ID3, Random Forest 3.3 Confusion Matrix 3.4 Evaluation Metrics for Classification (Accuracy, Precision, Recall, F1-score) 3.5 Other advanced metrics **Unit 4: Fundamentals of Artificial Neural Networks** 4.1 Neurons and Biological Motivation 4.2 Defining ANN 4.3 Layers and Multilayer Perceptron 4.4 Weights, bias, Activation Function, Loss function, Epochs **Unit 5: Artificial Neural Networks: Learning &Architectures** 5.1 Linear threshold units. Perceptrons: representational limitation and gradient descent training. 5.2 Types of Neural Network: Feed-Forward Neural Network, Backpropagation Neural Network, Error calculation in ANN 5.3 Learning in ANN and Learning Rate **Unit 6: Unsupervised Learning** 6.1 Learning from unclassified data. Clustering. 6.2 Hierarchical Agglomerative Clustering, k-means partitional clustering. 6.3 Expectation maximization (EM) for soft clustering.

	Semi-supervised learning with EM using labeled and unlabeled data. 6.4 Self-Organizing Maps 6.5 Hidden Markov Models **Unit 7: Model Validation** 7.1 ML Techniques overview 7.2 Validation Techniques (Cross-Validations) 7.3 Feature Reduction/Dimensionality Reduction 7.4 Principal components analysis (Eigen values, Eigen vectors, Orthogonality) 7.5 Generalization, Overfitting, and Underfitting: Relation of Model Complexity to Dataset Size **Unit 8: Advanced Machine Learning Concepts** 8.1 Reinforcement Learning 8.2 Transfer Learning 8.3 Federated Learning
Reference Books	1.AI and Machine Learning, Vinod Chandra SS, Anans Hareendran S. PHI Publication 2. Machine Learning with Python, Abhishek Vijayvargia, BPB Publication 3. Machine Learning Hand-On for Developers and Technical Professionals, Jason Bell, Wiley Publication 4. Machine Learning for Beginners: Learn to Build Machine Learning Systems Using Python , Harsh Bhasin, BPB Publication 5. Machine Learning - Tom M. Mitchell, McgrewHill Publication

Name of College: Shree Ramkrishna Institute of Computer Education and Applied Sciences						
Faculty	Science		**Program**	Master of Science Information Technology		
Year	2025-26		**Version**	2.0		
Semester	2		**Effective From**	December, 2025		
Course Code	MSCS22203	**Course Name**	**Generative AI**			
Teaching Scheme				**Examination Scheme**		
Credits	**Lecture (L)**	**Tutorial (T)**	**Practical (P)**	**Continuous Assessments (CCE)**	**Semester end examinations (SEE)**	**Total**
4	4	-	-	70	30	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	● To learn about generative artificial intelligence ● 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 Out come	After completion of this course, the student will be capable of developing generative AI based solutions.
Course Content	**Unit 1 Defining New Age AI** 1. Artificial Narrow Intelligence 2. Artificial General Intelligence 3. Artificial Super Intelligence 4. Discriminative Vs Generative AI 5. Applications of Generative AI **Unit 2 Generative AI Practices** 1. Transforming Text : Writing to Content Creation 2. Revoltinizing Art 3. AI in Sound and Music 4. AI generated Animation 5. Voice Generation with AI **Unit 3 Data Representation in Gen AI** 1. Uses of Generative AI - Text, Images, Video, Music, Code, Voices 2. Encoding, Embedding 3. Text Embedding 4. Image Embedding 5. Video Embedding

	6. Key Technologies behind Gen AI - GAN, VAE, Transformers **Unit 4 LLM and Prompt Engineering** 4.1 Language Modelling 4.2 Large Language Models 4.3 Applications of LLMs 4.4 Introduction of lightweight LLMs and SLMs 4.4 Prompt Designing, Task Formulation using Prompts, Prompt Patterns 4.5 Prompt Tuning Techniques **Unit 5 AI Agent Frameworks** 5.1 AI Agents 5.2 Introduction to Agent Orchestration 5.3 Open source AI agent frameworks 5.3.1 Langchain 5.3.2 LangGraph 5.3.3 Crew AI 5.3.4 AutoGen (Microsoft) 5.3.5 BabyAGI **Unit 6 Gen AI for Software Development** 1. Automated Code Generation - Github Co-pilot, Cursor IDE, Amazon Q Developer 2. Code Completion and Suggestion 3. Bug Detection –CodeGPT 4. Automated Testing - Selenium, Appium, Testim, Robot Framework **Unit 7 Gen AI Tools for Code Understanding** 1. Real time Code Completion-Tabine 2. Google's Gemini -debugging and code 3. AI2SQLfor Queries 4. Amazon CodeWhisperer 5. AI powered Code-Assistants **Unit 8 Working with GenAI** 8.1 Content creation tools 8.2 Code Generation tools 8.3 Story Telling with AI 8.4 Generative AI for Visual Creatives 8.5 Generative AI for Music Creation
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 Beginners Playbook by Branson Adams 4. Generative 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

Name of College: Shree Ramkrishna Institute of Computer Education and Applied Sciences						
Faculty	Science		**Program**	Master of Science Information Technology		
Year	2025-26		**Version**	2.0		
Semester	2		**Effective From**	December, 2025		
Course Code	MSCS23201	**Course Name**	**Advanced Cloud Programming**			
Teaching Scheme				**Examination Scheme**		
Credits	**Lecture (L)**	**Tutorial (T)**	**Practical (P)**	**Continuous Assessments (CA)**	**Term end examinations (TEE)**	**Total**
4	4	0	0	70	30	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** . Evolution of Cloud Computing a. Characteristics, benefits & challenges b. Cloud architectural model c. Cloud service models (IaaS, PaaS, SaaS) d. Cloud deployment models (Public, Private, Hybrid) **UNIT 2 – Evolution of Architectural Styles for Cloud-Native Applications** 3.1 Evolution of application architectures 2.1.1 Monolithic architecture 2.1.2 Service oriented architecture 2.1.3 Micro service architecture 3.2 Characteristics of cloud-native systems 3.3 Need for distributed architectures 3.4 Virtualization – Concept, Hypervisor and its Types 3.5 Containerization **UNIT 3 – Microservices Architecture & Decomposition Strategies** 3.3 Microservices architecture: principles & characteristics 3.4 Bounded context & domain-driven concepts 3.5 Decomposition strategies: 5.1.1 By business capability 5.1.2 By subdomain 5.1.3 Self-contained services 5.1.4 Service-per-team 3.6 Service communication patterns -REST and Event Driven Design **UNIT 4 – Microservice API, Business Logic & Persistence Design** 7.1 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 (WAR/JAR/Container-ready) 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 **UNIT 5 – Data & Transaction Management in Microservices** 5.1 Data Management Patterns 5.1.1. Database per service

	5.1.2. Shared database 5.1.3. Saga pattern 5.1.4 CQRS 5.1.5 Event sourcing 5.2 Transactional Messaging 5.2.1 Transactional outbox 5.2.2 Transaction log tailing 5.2.3 Polling publisher 5.3 Fault Tolerance 5.3.1. Circuit breaker pattern **UNIT 6 – DevOps for Microservices** 6.1. Need for DevOps in MSA 6.2. DevOps lifecycle 6.3. Continuous Integration & Continuous Delivery (CI/CD) 6.4. Containerization - Docker 6.5. Service orchestration - Kubernetes 6.6 Load balancing & service scaling **UNIT 7 – Cloud Platform Capabilities, Security & Observability** 7.1 Cloud Platform Capabilities for Microservices 7.1.1 Compute Services - AWS EC2, Elastic Beanstalk 7.1.2. Storage Services - AWS S3 7.1.3. Database Services - AWS RDS, DynamoDB 2.1 Cloud Security for Distributed Microservices 7.2.1. Security challenges in distributed systems 7.2.2. Authentication vs Authorization 7.2.3. JSON Web Tokens – structure, flow & usage 2.2 Cloud Observability and Monitoring 7.3.1. Importance of observability in MSA 7.3.2. Three pillars: Logging, Metrics, Tracing 7.3.3. Metrics scraping and Visualization **Unit 8 - Emerging Trends in Cloud Computing** 8.1 Edge Computing & Fog Computing 8.2 Quantum Cloud Computing 8.3 AI/ML in Cloud - AWS Sage maker 8.4 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 Paperback by 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

College: Shree Ramkrishna Institute of Computer Education and Applied Sciences						
Faculty	Science		**Program**	Master of Science Information Technology		
Year	2025-26		**Version**	2.0		
Semester	2		**Effective From**	December, 2025		
Course Code	MSCS26201	**Course Name**	**Practical and Project- 2**			
Teaching Scheme				**Examination Scheme**		
Credits	**Lecture (L)**	**Tutorial (T)**	**Practical (P)**	**Continuous Assessments (CA)**	**Term end examinations (TEE)**	**Total**
8	0	0	16	70	30	100

Purpose of Course	The purpose of the course is to make students capable of implementing concepts, methods, tools and techniques studied in courses of semester 1.
Course Objective	The objective of these courses is to enable students to learn practical implementation of DSC-3, DSC-4, SEC-2 and DSE-2 or implement in-house or industrial projects.
Pr-requisite	As per theory papers of semester -2
Course Out come	After completion of this course, the student will be capable of performing practical application of subjects given in semester -2.
Course Content	The students will be required to implement project related task using methods, techniques and tools discussed in(but not restricted to) DSC-1, DSC-2, SEC-1 and DSE-1 A Document journal must be prepared for the work done**.**
Reference Book	.As per paper DSC-3, DSC-4, SEC-2 and DSE-2.
Teaching Methodology	Inhouse Project Work and/or Internship