University of Mumbai

Website – mu.ac.in
Email id - dr.aams@fort.mu.ac.in
aams3@mu.ac.in

Academic Authorities,
Meetings &Services (AAMS)
Room No. 128, M. G. Road, Fort,
Mumbai – 400 032.
Tel.022-68320033

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

No.2026 May AAMS C-106/44024 Date: 14th May, 2026

CIRCULAR:-

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

They are hereby informed that the recommendations made by the **Board of Studies in Information Technology** at its meeting held on 17th February, 2026 vide item No.1 and subsequently passed by the Board of Deans at its meeting held on 17th March, 2026 vide item No. 6.16 (N) have been accepted by the Academic Council at its meeting held on 25th March, 2026 **vide** item No. 6.34 (N). In accordance therewith syllabus of **B.Sc. (Information Technology) (Scheme - I) (Sem V & VI)** (NEP 2020) is introduced as per appendix with effect from the academic year 2026-27.

(The said circular is available on the University's website www.mu.ac.in).

(Dr. Prasad Karande)
REGISTRAR

MUMBAI – 400 032
14th May, 2026

To

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

AC./6.34 (N)/25/3/2026

Copy forwarded with Compliments for information to:-

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

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

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

To,

1	The Chairman, Board of Deans pvc@fort.mu.ac.in
2	**Faculty of Humanities,** **Offg. Dean** 1. Prof.Anil Singh Dranilsingh129@gmail.com **Offg. Associate Dean** 2. Prof.Manisha Karne mkarne@economics.mu.ac.in 3. Dr.Suchitra Naik Naiksuchitra27@gmail.com
	Faculty of Commerce & Management, **Offg. Dean,** 1 Prin.Ravindra Bambardekar principal@model-college.edu.in **Offg. Associate Dean** 2. Dr.Kavita Laghate kavitalaghate@jbims.mu.ac.in 3. Prin.Kishori Bhagat kishoribhagat@rediffmail.com

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

AC – 25/03/2026
Item No. 6.34 (N)

As Per NEP 2020

University of Mumbai

Syllabus for Major Vertical – 1, 4 & 6 (Scheme – I)		
Faulty of Science & Technology		
Board of Studies in Information Technology		
Name of the Programme – B.Sc. (Information Technology)		
U.G. Third Year Programme	**Exit Degree**	**B.Sc. (Information Technology)**
Semester		**V & VI**
From the Academic Year		**2026-27**

University of Mumbai

(As per NEP 2020)

Sr.No.	Heading	Particulars
1	**Title of program** **O:_____**	**B.Sc. (Information Technology)**
2	**Exit Degree**	**B.Sc. (Information Technology)**
3	**Scheme of Examination** **R: _____**	**NEP** **40% Internal** **60% External, Semester End Examination** **Individual Passing in Internal and External Examination**
4	**Standards of Passing** **R: _____**	**40%**
5	**Credit Structure** **R: SU-510 E(I) R: SU-510 F(I)**	**Attached herewith**
6	**Semesters**	**Sem. V & VI**
7	**Program Academic Level**	**5.5**
8	**Pattern**	**Semester**
9	**Status**	**New**
10	**To be implemented from Academic Year**	**2026-27**

Sd/-
Sign of the BOS
Dr. R. Srivaramangai
Chairman/Co-ordinator
BOS/Ad-hoc BOS in Information Technology

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

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

B.Sc. (Information Technology)

Credit Structure (Sem. V & VI)

(B. Sc.)- Major & Minor

	R: SU-510 E (I)								
Vertical		**1**		**2**	**3**	**4**	**5**	**6**		
Level	**Semester**	**Major (M1)**		**Minor (M2)**	**OE**	**VSC, SEC (VSEC)**	**AEC, VEC, IKS**	**OJT, FP, CEP, CC,RP**	**Cum. Cr. / Sem.**	**Degree/ Cum. Cr.**
		Mandatory	**Electives**							
5.5	V	10 Dot Net Core and Progressive Web Application (2) Dot Net Core and Progressive Web Application(Practical) (2) Artificial Intelligence and Application Development (2) Artificial Intelligence Application Development and Jira (Practical) (2) Indian Knowledge Systems in Information Technology (2)	4 Enterprise Java Practical (2) **OR** Android Practical (2) Full Stack development MERN Practical (2) **OR** Linux Administration Practical (2)	4 (2+2)	----	VSC: 2 Internet of Things Practical (2)	----	CEP:2	22	**UG Degree 132**

	R: SU-510 F (I)								
	VI	10 Project Management (2) Computer Security (2) Data Analytics and Business Intelligence (2) Data Analytics and Business Intelligence and Computer Security Practical (2) Cloud Computing and Testing Tools Practical (2)	4 Geographical Information System Practical (2) **OR** Enterprise Networking Practical (2) Cyber law and Digital Policy (2) **OR** IT Service Management (2)	4 (2+2)	----	----	----	OJT :4	22	
	Cum Cr.	48	8	18	12	8+6	8+4+2	8+2+2 +2+4	132	
Exit option: Award of UG Degree in Major and Minor with 132 credits OR Continue with Major and Minor									

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

[* 2 Credit IKS Major paper (Core Subject Specific Theory may be included in Sem III or V)]

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

Vertical No		Paper Title		Credits
Sem. V				
1.	**Mandatory**	1. Dot Net Core and Progressive Web Application		2
		2. Dot Net Core and Progressive Web Application(Practical)		2
		3. Artificial Intelligence and Application Development		2
		4. Artificial Intelligence Application Development and Jira (Practical)		2
		5. Indian Knowledge Systems in Information Technology		2
	Electives	1	1. Enterprise Java Practical Or 2. Android Practical	2
		2	1. Full Stack development MERN Practical Or 2. Linux Administration Practical	2
2.	**Minor** **(To be selected from other programme)**			4 (2+2)
4.	**VSC (Practical)**	1. Internet of Things Practical		2
6.	CEP (To be selected from CEP topic list)			2
			Credits	**22**
Sem. VI				
1.	**Mandatory**	1. Project Management		2
		2. Computer Security		2
		3. Data Analytics and Business Intelligence		2
		4. Data Analytics and Business Intelligence and Computer Security Practical		2
		5. Cloud Computing and Testing Tools Practical		2
	Electives	1.	1. Geographical Information System Practical Or 2. Enterprise Networking Practical	2
		2	1. Cyber law and Digital Policy Or 2. IT Service Management.	2
2.	**Minor** (To be selected from other discipline)			4 (2+2)
6.	**VI**	**OJT**		4
			Credits	**22**
			Total Credits	**44**

Sem. - V

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

Syllabus
B.Sc. (Information Technology)
(Sem.- V)

Name of the Course: Dot .Net Core and Progressive Web Application

Sr.No.	Heading	Particulars
1	**Description the course :** **Including but Not limited to:**	.NET Core and PWA is a major theory course It bridges two of the most important domains in contemporary full-stack and cross-platform web development: server-side application development on the modern Microsoft .NET platform using C# and ASP.NET Core, and client-side Progressive Web Application (PWA) engineering using open web standards. Together, they equip students with an end-to-end skill set for building enterprise-grade, cross-device, and offline-capable web applications — a profile that is in exceptionally high demand across Indian IT industry, product companies, and global MNCs. language fundamentals (data types, control flow, methods, error handling) through object-oriented programming (classes, inheritance, polymorphism, interfaces, exception handling) — before progressing to ASP.NET Core architecture: the request pipeline, middleware, routing, the Model-View-Controller (MVC) pattern, Razor Pages, and RESTful Web API development. The second half pivots to Progressive Web Applications — exploring the PWA philosophy, Service Workers and their lifecycle, the Web App Manifest, caching strategies, offline functionality, push notifications, responsive/mobile-first design, and the critical integration of PWA front-ends with ASP.NET Core back-ends, rounded off by security fundamentals (CORS, JWT overview) and performance optimisation using industry tools such as Lighthouse.
2	**Vertical :**	Major
3	**Type :**	Theory
4	**Credits :**	2 credits (1 credit = 15 Hours for Theory)
5	**Hours Allotted :**	30 Hours
6	**Marks Allotted:**	50 Marks
7	**Course Objectives(CO):**	

	CO 1: To provide students with fundamental and advanced knowledge of C# programming and the modern .NET platform. CO 2: To enable students to design and develop web applications using ASP.NET Core architecture and MVC concepts. CO 3: To familiarize students with restful API development and integration with front-end applications. CO 4: To introduce the concepts, architecture, and implementation of Progressive Web Applications (PWA). CO 5: To develop skills in building responsive, secure, and performance-optimized web applications using modern web standards.
8	**Course Outcomes (OC):** OC 1: Develop applications using C# and implement object-oriented programming concepts in the .NET environment. OC 2: Design and deploy dynamic web applications using ASP.NET Core, routing, middleware, and MVC architecture. OC 3: Build and consume restful APIs and integrate backend services with client-side applications. OC 4: Create Progressive Web Applications with features such as service workers, web app manifest, and offline caching. OC 5: Apply modern web development practices including responsive design, security mechanisms, and performance optimization in real-world applications.
9	**Modules:-**
	Module 1: Introduction to C#.NET
	1. Overview of .NET Ecosystem:.NET Framework vs. .NET Core / .NET 5+ / .NET 7+, Common Language Runtime (CLR), Assemblies, CLI tools 2. Basics of C# Language: Data types, Variables, Operators, Expressions, Control Structures: If, Switch, Loops, Methods, Parameters, Error Handling 3. Object Oriented Programming in C#: Classes & Objects, inheritance, Polymorphism, Interfaces, Namespaces and Exception Handling 4. ASP.NET Core Fundamentals: Web Server Basics, Middleware, Request Lifecycle, Routing, Controllers, Views (MVC),Razor pages and REST API endpoints, Web Forms
	Module 2: PWA 1. Introduction to PWA: Concept and evolution of Progressive Web Applications, key characteristics (reliable, fast, engaging), comparison with native and hybrid apps, requirement of HTTPS. 2. Core PWA Components: Web App Manifest (properties and configuration), Service Workers (registration and lifecycle), caching strategies, offline functionality, and basic push notifications. 3. Responsive Web Design: Mobile-first approach, media queries, flexible layouts, and UI/UX principles for cross-device compatibility.

	4. Integration with .NET Backend: Consuming ASP.NET Core Web APIs, deployment basics, performance testing using tools like Lighthouse. Security and Optimization: CORS basics, authentication concepts (JWT overview), performance and accessibility best practices.
10	**Text Books** 1. Beginning ASP.NET 4.5 in C#, Matthew MacDonald,Apress 2. C# 2015, Anne Bohem and Joel Murach,Murach,Third 3. Progressive Web App,Dean Alan Hume
11	**Reference Books** 1. C# 4.0: The Complete Reference Paperback – 27 April 2010 by Herbert Schildt (Author) 2. C#: A Beginner's Guide, McGraw-Hill Professional 3. Progressive Web Applications by Jason Grigsby
12	**Internal Continuous Assessment: 40%**
	Continuous Evaluation through: 1. Class test of 1 of 15 marks 2. Class test of 2 of 15 marks Average of the two: 15 marks 3. Quizzes/ Presentations/ Assignments: 5 marks Total: 20 marks
13	**Semester End Examination: 60%**
	Format of Question Paper: External Examination (30 Marks)– 1 hr duration
14	**Format of Question Paper: (Semester End Examination : 30 Marks. Duration:1 hour)** Q1: Attempt any three (out of five/Six) from Module 1 (15 marks) Q2: Attempt any three (out of five/Six) from Module 2 (15 marks)

Syllabus
B.Sc. (Information Technology)
(Sem.- V)

Name of the Course: (Dot) .Net Core and Progressive Web Application (Practical)

Sr.No.	Heading	Particulars
1	**Description the course :** **Including but Not limited to:**	.Net Core and PWA practical course focuses on developing hands-on skills in C# and .NET technologies along with modern web development. Students work on basic programming tasks like arithmetic operations, sorting, and patterns, and gradually learn advanced concepts such as inheritance, interfaces, delegates, and exception handling. It also includes building web applications using ASP.NET, working with databases through Entity Framework Core, and performing CRUD operations. In addition, students are introduced to Progressive Web Applications (PWAs), where they learn to create web apps with offline support and real-world features.
2	**Vertical :**	Major
3	**Type :**	Practical
4	**Credits :**	2 credits (1 credit = 30 Hours of Practical work in a semester)
5	**Hours Allotted :**	60 Hours
6	**Marks Allotted:**	50 Marks
7	**Course Objectives(CO):** **CO1.** To provide fundamental knowledge of C# programming and object-oriented concepts. **CO2.** To develop problem-solving skills using advanced C# features such as delegates, events, interfaces, and exception handling. **CO3.** To enable students to design and develop web applications using ASP.NET Web Forms and ASP.NET Core. **CO4.** To introduce database connectivity and ORM concepts using SQL Server and Entity Framework Core. **CO5.** To familiarize students with Web API development and modern web technologies such as Progressive Web Applications (PWA).	
8	**Course Outcomes (OC):** OC1. Develop C# programs using basic programming constructs, arrays, strings, and pattern generation. OC2. Apply Object-Oriented Programming concepts such as inheritance, polymorphism, interfaces, constructors, delegates, and exception handling.	

	OC3. Design and develop ASP.NET Web Forms and ASP.NET Core applications using server controls and validation techniques. OC4. Implement database connectivity using SQL Server and Entity Framework Core to perform CRUD operations. OC5. Develop Web APIs and Progressive Web Applications (PWA) with offline support, service workers, and manifest configuration.
9	**Modules:-**
	Module 1:
	1.Write the program for the following: a. Create an MVC application to perform basic **Arithmetic Operations**. b. Create an application to print Floyd's triangle till n rows in C#. c. Write C# code to arrange the name of cities in sorted order. Accept name of 10 cities from the user **2. Write the program with different features of C#** a. Function Overloading b. Inheritance (all types) c. Constructor overloading d. Interfaces e. Using Delegates and events f. Exception handling 3. **Write the program for the following**: a. Create methods add(). multiply(), substract() ,divide() with suitable parameters and call these methods using concept of C# delegate. b. Write a program using multicast delegate. c. Create a class **BankAccount** with AccountNumber and Balance. Implement property for Balance with validation (Balance cannot be negative). **4. Create an ASP.NET Web Forms application using simple server controls.** a. Design a Student Registration Form using the following ASP.NET controls: i. Label and TextBox to accept Student Name ii. RadioButton to select Gender (Male/Female) iii. DropDownList to select Course (BSc IT, BSc CS, BCA) iv. CheckBox for accepting Terms & Conditions v. Calendar control to select Date vi. Button control to submit the form vii. Label control to display the output b. Create a Registration form to demonstrate use of various Validation controls. c. Create Web Form to demonstrate use of Adrotator Control. **5. Create the following application:** a. Create a web application to demonstrate the use of different types of Cookies. b. Create a **Product Table** in SQL Server and perform **Insert Operation** using Entity Framework Core.

	c. Develop a web application to perform **CRUD Operations (Create, Read, Update, Delete)** on Employee table using EF Core. d. Authentication and Authorization in ASP.NET Core.
	Module 2:
	1. a. Create a simple website for a College Information Portal and convert it into a Progressive Web Application. b. Create a PWA that displays a custom offline page when the user is not connected to the internet. 2. a. Develop a PWA that works in offline mode using Service Worker. b. Create a Student Record Management PWA. 3. a. Create an HTML page that would be the starting point of the application. This HTML will contain a link to the file named manifest.json. This is an important file that would be created in the next step. b. Create a simple Web API that returns Student data in JSON format. 4. a. Develop CRUD operations using ASP.NET Core Web API and Entity Framework Core. b. Create an API endpoint to calculate Factorial of a Number and test it using Postman. 5. a. Create a PWA with a Web App Manifest file including: App name Icons Start URL Display mode Implement Push Notifications in a PWA. b. Develop an E-Commerce PWA with product listing and offline cart functionality.
10	**Text Books** 1. Programming in C# – E. Balagurusamy, McGraw Hill Education. 2. ASP.NET Core in Action – Andrew Lock, Manning Publications. 3. Pro ASP.NET Core MVC – Adam Freeman, Apress. 4. Entity Framework Core in Action – Jon P Smith, Manning Publication
11	**Reference Books** 1. Professional ASP.NET Core – Jon Galloway et al., Wrox Publication. 2. Learning Progressive Web Apps – John Wargo, Addison-Wesley.
12	**Internal Continuous Assessment: 40%**
	Continuous Evaluation through: Students are expected to attend each practical and submit the written practical of the previous session. Performing Practical and write-up submission will be continuous internal evaluation. 2.5 marks can be awarded for each practical performance and write-up submission totalling to 50 marks and can be converted to 20 marks.

13	**Semester End Examination: 60%**
	30 marks Semester End Practical Examination. (2 hours duration)
14	**Format of Question Paper: (Semester End Examination : 30 Marks. Duration: 2 hours)** **Practical Slip:** Q1. From Module 1 13 marks Q2. From Module 2 12marks Q3. Journal and Viva 05 marks

Syllabus
B.Sc. (Information Technology)
(Sem.- V)

Title of Paper <u>Artificial Intelligence and Application Development</u>

Sr. No.	Heading	Particulars
1	Description of the course : Including but Not limited to : 	This course introduces the fundamental concepts and practical methodologies of Artificial Intelligence with a strong emphasis on real-world application development. It covers intelligent agents, problem-solving through search, probabilistic reasoning, machine learning, and deep learning techniques using standard AI frameworks and development tools. Students learn to design, implement, evaluate, and deploy AI models as part of complete application systems. The course integrates theoretical foundations with hands-on development, enabling learners to build intelligent solutions such as predictive analytics systems, recommendation engines, chatbots, and computer vision applications.
2	Vertical :	Major
3	Type :	Theory
4	Credit:	2 credits (1 credit = 15 Hours for Theory)
5	Hours Allotted :	30 Hours
6	Marks Allotted:	50 Marks
7	Course Objectives(CO): CO 1. To understand and apply core AI concepts including intelligent agents, search techniques, probabilistic reasoning, and fundamental machine learning models. CO 2.To implement supervised and introductory deep learning algorithms with appropriate evaluation and performance analysis. CO3. To design end-to-end AI pipelines integrating data preprocessing, model development, validation, and inference within application architectures. CO4. To develop and deploy AI-based applications while ensuring scalability, robustness, and ethical considerations.	
8	**Course Outcomes (OC):** OC1. Apply intelligent agent models, search algorithms, probabilistic reasoning, and fundamental machine learning techniques to solve structured AI problems. OC2. Implement and evaluate supervised and basic deep learning models using appropriate performance metrics.	

	OC3. Design and construct end-to-end AI pipelines including data preprocessing, model training, validation, and inference integration within applications. OC4. Develop and deploy AI-based applications ensuring scalability, robustness, and adherence to ethical and responsible AI practices.
9	**Modules**:- Per credit One module can be created
	Module 1: Foundations and theoretical approach
	1. **Introduction**: What Is AI? The Foundations of Artificial Intelligence, The State the Art, Risks and Benefits of AI. 2. **Intelligent Agents**: Agents and Environments, Good Behaviour: The Concept of Rationality, The Nature of Environments, The Structure of Agents 3. **Solving Problems by Searching**: Problem-Solving Agents, Example Problems, Search Algorithms, Uninformed Search Strategies, Informed (Heuristic) Search Strategies, Heuristic Functions 4. **Quantifying Uncertainty**: Acting under Uncertainty, Basic Probability Notation, Inference Using Full Joint Distributions, Independence, Bayes' Rule and Its Use, Naive Bayes Models, The Wumpus World Revisited **5. Probabilistic Reasoning**: Representing Knowledge in an Uncertain Domain, The Semantics of Bayesian Networks, Exact Inference in Bayesian Networks, Approximate Inference for Bayesian Networks, Causal Networks
	Module 2: AI Application Development and Ethics
	1. Learning from Examples: Forms of Learning, Supervised Learning, Learning Decision Trees, Model Selection and Optimization, The Theory of Learning, Linear Regression and Classification, Nonparametric Models, Ensemble Learning, Developing Machine Learning Systems 2. **The Machine Learning Landscape** :What Is Machine Learning?, Why Use Machine Learning?, Types of Machine Learning Systems, Supervised and Unsupervised Learning, Batch and Online Learning, Instance-Based Versus Model-Based Learning, Main Challenges of Machine Learning, Insufficient Quantity of Training Data, Nonrepresentative Training Data, Poor-Quality Data, Irrelevant Features, Overfitting the Training Data, Underfitting the Training Data, Stepping Back, Testing and Validating, Exercises 3. **Classification**: MNIST, training a Binary Classifier, Performance Measures, Measuring Accuracy Using Cross-Validation, Confusion Matrix, Precision and Recall, Precision/Recall Tradeoff, The ROC Curve, Multiclass Classification, Error Analysis, Multilabel Classification, Multioutput Classification 4. **Decision Trees:** Training and Visualizing a Decision Tree, Making Predictions, Estimating Class Probabilities, The CART Training Algorithm, Computational Complexity, Gini Impurity or Entropy? Regularization Hyperparameters, Regression, Instability, Exercises

	5. **Philosophy, Ethics, and Safety of AI:** The Limits of AI, Can Machines Really Think?, The Ethics of AI
10	**Text Books** 1. Artificial Intelligence: A Modern Approach – Stuart Russell and Peter Norvig – Pearson Education, 2020 2. Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow – Aurélien Géron,O'Reilly Media, 2022 (Third Edition)
11	**Reference Books:** 1. The Hundred-Page Machine Learning Book – Andriy Burkov – Andriy Burkov, 2020 (First Edition) 2. Machine Learning for Absolute Beginners – Oliver Theobald – Independently published, 2020 (First Edition) 3. Artificial Intelligence: Foundations of Computational Agents – David L. Poole and Alan K. Mackworth– Cambridge University Press, 2023 (Third Edition) 4. Online Resources: https://artint.info/3e/slides/index.html https://soclibrary.futa.edu.ng/books/ Machine%20Learning%20Engineering%20(Andriy%20Burkov)%20(Z-Library).pdf
12	**Internal Continuous Assessment: 40%**
	Continuous Evaluation through: 1. Class test of 1 of 15 marks 2. Class test of 2 of 15 marks Average of the two: 15 marks 3. Quizzes/ Presentations/ Assignments: 5 marks Total: 20 marks
13	**Semester End Examination: 60%**
	External Examination (30 Marks)– 1 hr duration
14	**Format of Question Paper: (Semester End Examination : 30 Marks. Duration:1 hour)** Q1: Attempt any three (out of five/six) from Module 1 (15 marks) Q2: Attempt any three (out of five/six) from Module 2 (15 marks)

Syllabus
B.Sc. (Information Technology)
(Sem.- V)

Name of the Course: Artificial Intelligence Application Development and Jira (Practical)

Sr.No.	Heading	Particulars
1	**Description the course :** **Including but Not limited to:**	This practical component provides hands-on experience in implementing Artificial Intelligence and machine learning algorithms using modern tools. It covers search techniques, probabilistic modelling, classification, regression, and model evaluation. Students work with datasets to build, validate, and analyse AI systems**.** This course also provides hands-on training in Agile Project Management using Jira Cloud. Students will learn how to create and manage Scrum and Kanban projects, configure workflows, manage issues, generate reports, implement automation, and apply security controls. The course emphasizes practical implementation aligned with industry standards to enhance employability and project management skills.
2	**Vertical :**	Major
3	**Type :**	Practical
4	**Credits :**	2 credits (1 credit = 30 Hours of Practical work in a semester)
5	**Hours Allotted :**	60 Hours
6	**Marks Allotted:**	50 Marks
7	**Course Objectives(CO):** CO1. Implement intelligent agent models and search algorithms to solve AI problems. CO2. Develop probabilistic models and perform inference using appropriate tools. CO3. Build and train machine learning models for classification and regression tasks. CO4. Evaluate AI models using validation techniques and performance metrics. CO5. Develop and deploy a complete AI application using datasets. CO6. Assess bias, fairness, and ethical considerations during AI model development. CO7. To understand Agile project management concepts. CO8. To configure and manage projects using Jira Cloud. CO9. To implement Scrum and Kanban frameworks.	

	CO10. To customize workflows, permissions, dashboards, and automation.
8	Course Outcomes (OC): **OC1.** Implement intelligent agent models and classical search algorithms to solve structured AI problems. OC2. Construct probabilistic models and perform inference using Bayes' rule, Naive Bayes, and Bayesian networks. OC3. Develop and train machine learning models for classification and regression using real-world datasets. OC4. Evaluate and compare AI models using appropriate performance metrics such as accuracy, precision, recall, F1-score, and ROC curves. OC5. Apply cross-validation and basic hyperparameter tuning techniques to improve model performance. OC6. Design and deploy a basic AI application while analysing fairness, bias, and ethical considerations. OC7. Create and configure Jira projects. OC8. Manage issues and workflows effectively. OC9. Implement Scrum and Kanban practices. OC10. Use JQL, reports, and dashboards for project tracking. OC11. Configure permissions, security, and automation rules
9	Modules:-
	Module 1: **Introduction to AI and Intelligent Agents** **1.** Design a Simple Rational Agent by defining PEAS for Vacuum Cleaner Environment and Autonomous Taxi. Classify environments (fully/partially observable**,** deterministic/stochastic, episodic/sequential). Implement a simple table-driven agent in Python. **Problem Solving by Searching (Uninformed Search)** **2.** Given an initial configuration of the 8-puzzle and a goal configuration, write a Python program to find the shortest sequence of moves to reach the goal state using Breadth-First Search (BFS). **3.** Given two water jugs of 4 litres and 3 litres capacity, write a Python program to obtain exactly 2 litres in one jug using Depth-First Search (DFS). [Vary capacity of jugs] Given a weighted graph representing cities and distances between them, write a Python program to find the least-cost path from Arad to Bucharest using Uniform Cost Search (UCS). [Provide any other weighted graph for applying UCS] **Problem Solving by Searching (Informed Search)** **1.** Given an initial configuration of the 8-puzzle and a goal configuration, write a Python program to find the solution using Greedy Best-First Search with the Manhattan Distance heuristic. **2.** Given an initial configuration of the 8-puzzle and a goal configuration, write a Python program to find the shortest path using A* search with the Manhattan Distance heuristic.

	3. Given a weighted graph representing cities and distances between them, write a Python program to find the shortest path from Arad to Bucharest using **A*** search with a heuristic function (straight-line distance to destination). **Bayes' Rule Application** 1. **Scenario:** A medical test is used to detect a particular disease. The probability that a randomly selected person has the disease is 1%. If a person has the disease, the test returns positive with probability 99%. If a person does not have the disease, the test still returns positive with probability 5% (false positive rate). A person takes the test and receives a positive result. **Problem Statement:** Write a Python program to compute the probability that the person actually has the disease given that the test result is positive. **Naïve Bayes' Classification** 1. Using a small dataset (e.g., weather or spam dataset), implement a Naive Bayes classifier using scikit-learn**.** **Probability Reasoning (Bayesian Networks & Inference)** 1. Using a simple example (Burglary–Alarm), create a Bayesian Network using pgmpy**.** 2. Write a program to compute Posterior Probability using Bayesian Networks(Burglar|Alarm = True) **Machine Learning** 1. Using any small dataset (e.g., Iris or a CSV dataset), write a Python program to demonstrate the basic machine learning workflow. a. Load dataset using Pandas. b. Perform basic preprocessing (handling missing values or scaling). c. Split dataset into training and testing sets. d. Train a simple classifier. e. Display training and testing accuracy. 2. Using a suitable regression dataset, implement Linear Regression using scikit-learn. 3. Using the MNIST dataset (digit vs not digit), implement a binary classifier. 4. Implement k-fold cross-validation on a classification model. 5. Implement a Decision Tree classifier using the Iris or any relevant dataset. 6. Implement Decision Tree regression on a regression dataset. 7. Implement k-Nearest Neighbors (kNN) classifier. 8. Implement multiclass classification using Logistic Regression or Decision Tree. **Ethics and Bias Analysis** Select any publicly available dataset (for example, Iris, Adult Income, or a classification dataset of your choice) and analyze it for possible bias and class imbalance**.**
	Module 2: Jira Setup & Project Configuration, Advanced Configuration, Reporting & Governance
	Practical 1: Jira Setup & Project Creation

	A. Create a Jira Cloud account and explore the dashboard. B. Create a Company-managed Scrum/Kanban Project. C. Configure project details **Practical 2: Team Creation & Role Assignment** A. Add Team Members B. Assign Roles **Practical 3: Issue Types & Configuration** A. Create and configure Issue Types (Epic, Story, Task, Bug) B. Create Components and Versions. **Practical 4: Issue Management** A. Create issues under an Epic. B. Assign issues to users and set Priority & Due Date. Practical 5: Workflow Customization A. Design a custom workflow B. Add transitions and conditions. C. Publish and associate workflow with a project. Practical 6: Scrum Board & Sprint Management A. Create and configure Scrum Board. B. Create and start a Sprint. C. Complete Sprint and generate Sprint Report **Practical 7: Kanban Board Management** A. Create a Company-managed Kanban Project. B. Configure Kanban Board C. Track issue progress and generate Control Chart **Practical 8: JQL & Filters** A. Write basic JQL queries (Status, Assignee, Priority). B. Create and save filters. **Practical 9: Reports & Dashboards** A. Generate Burndown Chart and Velocity Chart. B. Create Pie Chart / Issue Statistics Report. C. Create a Custom Dashboard with gadgets. **Practical 10: Permission & Security Management** A. View and modify Project Permission Scheme. B. Configure Project Roles and assign permissions. **Practical 11: Jira Automation** A. Create an Automation Rule (Trigger → Condition → Action). B. Configure rule (e.g., When issue is moved to Done → send email / auto-assign). C. Test and enable automation rule. **Practical 12: Introduction to Jira Mobile App** A. Install and configure the Jira Mobile App. B. View and update issues from mobile (Status, Comment, Assign). Receive notifications and manage real-time updates

10	**Text Books** 1. Artificial Intelligence: A Modern Approach – Stuart Russell and Peter Norvig – Pearson Education, 2020 2. Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow – Aurélien Géron,O'Reilly Media, 2022 (Third Edition) 3. Schwaber, Ken & Sutherland, Jeff. The Scrum Guide. **4.** Official Documentation of Jira Software by Atlassian.
11	**Reference Books** 1. The Hundred-Page Machine Learning Book – Andriy Burkov – Andriy Burkov, 2020 (First Edition) 2. Machine Learning for Absolute Beginners – Oliver Theobald – Independently published, 2020 (First Edition) 3. Artificial Intelligence: Foundations of Computational Agents – David L. Poole and Alan K. Mackworth – Cambridge University Press, 2023 (Third Edition) 4. Online Resources: a. https://artint.info/3e/slides/index.html b. https://soclibrary.futa.edu.ng/books/Machine%20Learning%20Engineering%20(Andriy%20Burkov)%20(Z-Library).pdf 5. Highsmith, Jim. *Agile Project Management: Creating Innovative Products.* Addison-Wesley. 6. Cohn, Mike. *Succeeding with Agile: Software Development Using Scrum.* Addison-Wesley. 7. Rubin, Kenneth S. *Essential Scrum: A Practical Guide to the Most Popular Agile Process.* Addison-Wesley. 8. https://www.atlassian.com/software/jira/guides 9. JIRA 4.3 Documentation User documentation for JIRA Software Server 7.1
12	**Internal Continuous Assessment: 40%**
	Continuous Evaluation through: Students are expected to attend each practical and submit the written practical of the previous session. Performing Practical and write-up submission will be continuous internal evaluation. 2.5 marks can be awarded for each practical performance and write-up submission totalling to 50 marks and can be converted to 20 marks.
13	**Semester End Examination: 60%**
	30 marks Semester End Practical Examination. (2 hours duration)
14	**Format of Question Paper: (Semester End Examination : 30 Marks. Duration: 2 hours)** **Practical Slip:** Q1. From Module 1 13 marks Q2. From Module 2 12marks Q3. Journal and Viva 05 marks

IKS
(2)

Syllabus
B.Sc. (Information Technology)
(Sem.- VI)

Name of the Course: Indian Knowledge Systems in Information Technology

Sr.No.	Heading	Particulars
1	**Description the course: Including but Not limited to:**	This course explores Indian Knowledge Systems through philosophy, mathematics, logic, linguistics, and science, highlighting their relevance to information technology. It examines epistemology, algorithms, number systems, logic, Sanskrit grammar, artificial intelligence, mechanical computation, and modern digital preservation. The syllabus connects ancient Indian thought with contemporary computing, interdisciplinary research, ethical technology, and future innovations across education, governance, culture, and emerging global technologies.
2	**Vertical :**	IKS
3	**Type :**	Theory
4	**Credits :**	2 credits (1 credit = 15 Hours for Theory)
5	**Hours Allotted :**	30 Hours
6	**Marks Allotted:**	50 Marks
7	**Course Objectives (CO):** CO 1. To Analyze the core methods of knowledge generation in Indian Knowledge System CO 2. To Explore selected contributions from Nyāya, Vaiśeṣika, Vedic mathematics, and ancient computational traditions in modern data science CO 3. To Integrate IKS-informed perspectives into data ethics and responsible AI CO 4. To Leverage IKS domains for domain-specific data science innovation CO 5. To Develop culturally rooted, multidisciplinary problem-solving competence	
8	**Course Outcomes (OC):** OC 1. Apply classical Indian logical, mathematical, and algorithmic traditions OC 2. Compare and contrast epistemological frameworks for reliable knowledge generation in data-intensive environments. OC 3. Integrate ethical and value-based principles from Indian Knowledge Systems OC 4. Analyze and adapt domain-specific insights from traditional Indian knowledge repositories OC 5. Design multidisciplinary, culturally grounded data science solutions for pressing Indian societal challenges.	
9	**Modules: -**	
	Module 1:	

	Unit 1: Introduction to Indian Knowledge Systems and Their Relevance to Information Technology **Philosophical foundations of knowledge in Indian traditions:** Concepts of *Pramana* (means of knowledge), *Pratyaksha* (perception), *Anumana* (inference), *Upamana* (comparison), and *Shabda* (verbal testimony); epistemological frameworks in Nyaya, Vedanta, and Mimamsa schools. **Relationship between Indian Knowledge Systems and Information Technology:** Understanding how structured knowledge representation, systematic reasoning, and rule-based frameworks in Indian traditions relate to modern computational models and information systems. **Relevance of IKS to modern computing and interdisciplinary research:** Role of traditional knowledge in algorithmic thinking, data representation, knowledge management, and artificial intelligence research. **Development of the decimal place value system :** Evolution of the base-10 positional number system in ancient India; transmission of the system to other civilizations; importance of place value in digital computation. **Concept of Zero and its computational significance :** Philosophical and mathematical interpretations of *Shunya*; Brahmagupta's rules for arithmetic operations involving zero; role of zero in modern computing and binary logic. **Binary systems and combinatorics in Pingala's Chandas Shastra :** Study of binary representation of poetic meters; patterns of short and long syllables; combinatorial methods resembling binary enumeration and algorithmic procedures. **Unit 2: Mathematical Foundations and Indian Logic & Its Applications in Ancient India and Their Relevance to Information Technology** **Aryabhata's computational algorithms :** Methods for calculating square roots, cube roots, and trigonometric values; use of iterative procedures and mathematical tables in astronomical calculations. Algorithmic thinking in ancient Indian mathematics : Use of step-by-step computational procedures, rule-based calculations, recursion-like structures, and systematic numerical methods. **Fundamentals of Nyaya logic :** Structure of logical reasoning in the Nyaya school; classification of knowledge, fallacies, and logical arguments. **Indian syllogism and inference structures :** Five-part syllogism consisting of proposition, reason, example, application, and conclusion; role of inference in knowledge validation. **Comparison of Indian and Western logical systems :** Differences between Nyaya logic and Aristotelian logic; multi-step reasoning and contextual inference in Indian philosophy. **Inference systems and reasoning models :** Types of inference such as *Purvavat*, *Sheshavat*, and *Samanyatodrishta*; relevance of these inference mechanisms in logical reasoning. **Kautilya's Arthashastra and strategic decision-making :** Analytical thinking, statecraft strategies, problem-solving frameworks, and structured planning models described in the Arthashastra. **Applications in modern computing systems :** Relevance of logical frameworks in programming languages, knowledge-based systems, rule-based expert systems, and artificial intelligence reasoning mechanisms. **Unit 3: Sanskrit Grammar and Computational Linguistics**

	Panini's Ashtadhyayi and rule-based linguistic systems : Structure of Panini's grammar consisting of nearly four thousand rules; meta-rules and rule hierarchy; systematic description of phonetics and morphology. **Panini's grammar and formal language theory :** Similarity between Panini's generative rules and modern formal grammars; production rules and symbol manipulation. **Generative grammar and rule-based systems : Processes** of derivation and transformation of words and sentences; parallels with modern programming structures and algorithmic rules. **Sanskrit and programming language development :** Precision and unambiguity of Sanskrit syntax; influence on formal language design and structured programming concepts. **Sanskrit grammar and compiler design :** Lexical analysis, syntax rules, and grammar parsing concepts comparable with modern compiler design principles. **Natural Language Processing and Sanskrit :** Application of Sanskrit grammatical structures in machine translation, semantic representation, computational linguistics, and AI-based language processing.
	Module 2:
	Unit 4: Artificial Intelligence and Cognitive Models in Ancient Indian Thought **Concepts of cognition and consciousness in Indian philosophy :** Understanding perception, cognition, memory, and reasoning in Vedanta, Nyaya, and Yoga traditions. **Knowledge representation in classical Indian systems :** Categorization of knowledge, classification of reality, and systematic organization of concepts. **Learning models and mental processes** : Processes of observation, inference, and experiential learning described in philosophical texts. **Artificial entities and automata in Indian mythology :** Descriptions of mechanical beings and artificial constructs in classical narratives and epics. **Predictive astronomy and computational models :** Astronomical calculations in Aryabhatiya and later mathematical texts; prediction of planetary movements using mathematical formulas. **Pattern recognition and classification in ancient Indian texts :** Use of classification systems in linguistics, medicine, and philosophy to organize knowledge systematically. **Unit 5: Mechanical Devices and Early Computational Concepts** **Early counting devices and numerical tools :** Use of counting boards, abacus-like tools, and other calculation aids in trade and astronomy. **Chaturanga and combinatorial reasoning :** Strategic planning and move combinations in the ancient Indian game Chaturanga; connection with modern game theory and computational strategy. **Combinatorics and recursion in Indian mathematics :** Permutation and combination techniques used in poetic meters, mathematics, and astronomy. **Cryptography and coded communication in ancient India :** Use of symbolic codes, secret writing, and encrypted communication in governance and literature. **Indian astronomy and early computational models :** Mathematical models used to compute planetary positions, eclipses, and calendrical calculations. **Comparison with early global computational devices :** Comparative understanding of Indian computational techniques with inventions such as the Antikythera mechanism. **Unit 6: Present Innovations and Future Directions**

	Digitization of traditional knowledge systems : Efforts to digitize manuscripts, preserve classical texts, and create digital repositories. **Knowledge management and digital archives :** Use of databases, metadata standards, and information systems to organize traditional knowledge. **Application of cloud computing and AI in preserving IKS :** Using modern computing technologies to analyze, classify, and preserve ancient texts and data. **Integration of Indian Knowledge Systems with emerging technologies :** Interdisciplinary research connecting IKS with artificial intelligence, data science, and computational linguistics. **Ethical and philosophical perspectives for future technologies :** Applying principles from Indian philosophy to guide ethical AI development and responsible technological innovation.
10	**Text Books** 1. Bhanu Murthy, T. S. A Modern Introduction to Ancient Indian Mathematics. New Age International, 1992. 2. Kak, Subhash and Rao, T. R. N. Computing Science in Ancient India. Center for Advanced Computer Studies, University of Southwestern Louisiana, 1998. 3. Mahadevan, B., Bhat, Vinayak Rajat, and Nagendra Pavana R. Introduction to Indian Knowledge System: Concepts and Applications. PHI Learning, 2022. 4. Kapoor, Kapil and Singh, Avadesh Kumar. Indian Knowledge Systems. D.K. Printworld. 5. Plofker, Kim. Mathematics in India. Princeton University Press. 6. Matilal, B. K. The Character of Logic in India. SUNY Press. 7. Cardona, George. Panini: A Survey of Research. Motilal Banarsidass. 8. Radhakrishnan, S. Indian Philosophy (Vol. 1 & 2). Oxford University Press. 9. Briggs, Rick. Knowledge Representation in Sanskrit and Artificial Intelligence. AI Magazine, 1985. 10. https://www.researchgate.net/publication/383541219_Sanskrit's_Role_in_Advancing_AI_A_Comprehensive_Study
11	**Reference Books** • Mahadevan, B., Bhat, Vinayak Rajat, and Nagendra Pavana R. Introduction to Indian Knowledge System: Concepts and Applications. PHI Learning, 2022. • Kapil Kapoor and Avadesh Kumar Singh. Indian Knowledge Systems. D.K. Printworld. • Kak, Subhash and Rao, T. R. N. Computing Science in Ancient India. Center for Advanced Computer Studies, University of Southwestern Louisiana, 1998. • Bhanu Murthy, T. S. A Modern Introduction to Ancient Indian Mathematics. New Age International, 1992. • Kim Plofker. Mathematics in India. Princeton University Press, 2009. • C. K. Raju. Cultural Foundations of Mathematics. Pearson Education. • Briggs, Rick. Knowledge Representation in Sanskrit and Artificial Intelligence. AI Magazine, 1985.
12	**Internal Continuous Assessment: 40%**
	Continuous Evaluation through: 1. Class test of 1 of 15 marks 2. Class test of 2 of 15 marks Average of the two: 15 marks

	3. Quizzes/ Presentations/ Assignments: 5 marks **4.** Total: 20 marks
13	**Semester End Examination: 60%**
	30 marks Semester End Examination
14	**Format of Question Paper: (Semester End Examination: 30 Marks. Duration: 1hr)** Q1: Attempt any three (out of five/six) from Module 1 (15 marks) Q2: Attempt any three (out of five/six) from Module 2 (15 marks)

SEM V
Vertical – 1
Electives
(2+2)

Syllabus
B.Sc. (Information Technology)
(Sem.- V)

Name of the Course: Enterprise Java Practical

Sr.No.	Heading	Particulars
1	**Description the course :** **Including but Not limited to:**	This course equips learners with the knowledge and practical skills required to design, develop, and deploy Enterprise Java applications through extensive hands-on laboratory sessions. It covers core and advanced technologies including Servlets, JDBC, Cookies and Session management, JSP, EJB, Hibernate, and the Spring Framework. Emphasis is placed on building scalable, secure, and database-driven web applications. By the end of the course, learners will be able to integrate these technologies effectively to develop robust, real-world enterprise solutions aligned with industry standards.
2	**Vertical :**	Electives
3	**Type :**	Practical
4	**Credits :**	2 credits (1 credit = 30 Hours of Practical work in a semester)
5	**Hours Allotted :**	60 Hours
6	**Marks Allotted:**	50 Marks
7	**Course Objectives(CO):** CO1:To understand the concept of and develop applications using servlets and database connectivity. CO2:To develop applications that can handle cookies, sessions and file operations. CO3:To understand the concept of and design applications using Java server pages. CO4:To understand the concept of and design applications using Enterprise Java Beans. CO5:To understand the concepts of persistence, Hibernate and develop JPA Applications, Hibernate applications. CO6: To understand Spring Framework	
8	**Course Outcomes (OC):** OC1. Proficiently understand and apply servlets and database connectivity concepts to develop dynamic web applications. OC2. Demonstrate the ability to develop applications capable of managing cookies, sessions, and performing file operations effectively.	

	OC3. Proficient in understanding and designing applications using Java Server Pages (JSP), enabling dynamic and interactive web content creation OC4. Adept at comprehending and designing applications utilizing Enterprise Java Beans (EJB), facilitating the development of scalable and distributed enterprise-level applications OC5. Possess a thorough understanding of persistence concepts, Hibernate framework, and the ability to develop Java Persistence API (JPA) and OC6. Hibernate applications proficiently. OC7. Apply Spring Framework, dependency injection with spring
9	**Modules:-**
	Module 1: 1. Implement the following Simple Servlet applications. a. Create a simple calculator application using servlet. b. Create a servlet for a login page. If the username and password are correct then it says message “Hello <username>” else a message “login failed” c. Create a registration servlet in Java using JDBC. Accept the details such as Username, Password, Email, and Country from the user using HTML Form and store the registration details in the database. 2. Implement the following Servlet applications with Cookies and Sessions. a. Using Request Dispatcher Interface create a Servlet which will validate the password entered by the user, if the user has entered "Servlet" as password, then he will be forwarded to Welcome Servlet else the user will stay on the index.html page and an error message will be displayed. b. Create a servlet that uses Cookies to store the number of times a user has visited servlet. c. Create a servlet demonstrating the use of session creation and destruction. Also check whether the user has visited this page first time or has visited earlier also using sessions. 3. Implement the Servlet IO and File applications. a. Create a Servlet application to upload and download a file. b. Develop Simple Servlet Question Answer Application using Database. c. Create simple Servlet application to demonstrate Non-Blocking Read Operation. 4. Implement the following JSP applications a. Develop a simple JSP application to display values obtained from the use of intrinsic objects of various types. b. Develop a simple JSP application to pass values from one page to another with validations. (Name-txt, age-txt, hobbies-checkbox, email-txt, gender-radio button).

	c. Create a registration and login JSP application to register and authenticate the user based on username and password using JDBC 5. Implement the Servlet IO and File applications. a. Create a Servlet application to upload and download a file. b. Develop Simple Servlet Question Answer Application using Database. c. Create simple Servlet application to demonstrate Non-Blocking Read Operation. 6. Implement the following JSP JSTL and EL Applications. a. Create an html page with fields, eno, name, age, designation, salary. Now on submit this data to a JSP page which will update the employee table of database with matching eno. b. Create a JSP page to demonstrate the use of Expression language. c. Create a JSP application to demonstrate the use of JSTL.
	Module 2: 7. Implement the following EJB Applications a. Create a Currency Converter application using EJB b. Develop a Simple Room Reservation System Application Using EJB c. Develop simple shopping cart application using EJB [Stateful Session Bean]. 8. Implement the following EJB applications with different types of Beans a. Develop simple EJB application to demonstrate Servlet Hit count using Singleton Session Beans. b. Develop simple visitor Statistics application using Message Driven Bean [Stateless Session Bean]. c. Develop simple Marks Entry Application to demonstrate accessing Database using EJB. 9. Implement the following JPA applications. a. Develop a simple Inventory Application Using JPA. b. Develop a Guestbook Application Using JPA. c. Create simple JPA application to store and retrieve Book details. 10. Implement the following JPA applications with ORM and Hibernate. a. Develop a JPA Application to demonstrate use of ORM associations. b. Develop a Hibernate application to store Feedback of Website Visitor in MySQL Database c. Develop a Hibernate application to store and retrieve employee details in MySQL Database. 11. Implement the following Hibernate a. Develop an application to demonstrate Hibernate One- To -One Mapping Using Annotation.

	b. Develop Hibernate application to enter and retrieve course details with ORM Mapping c. Develop a five page web application site using any two or three Java EE Technologies. 12. Implement the following using Spring Concept a. Build web application in Java with spring boot3 b. Develop application using Spring Framework, Lightweight c. Containers and Dependency Injection with Spring
10	**Text Books** 1. Java EE 8 Application, David R Heffelfinger, Packt, First, 2017 2. Java EE Essentials, Arun Gupta, O'reilly, First, 2013
11	**Reference Books** 1. Java EE 7 For Beginners, Sharanam Shah & Vaishali Shah, SPD First, 2017 2. Java EE cookbook, Elder Moraes, Packt, First, 2018 3. Advanced Java Programming, Uttam Kumar Roy, Oxford Press , 2015
12	**Internal Continuous Assessment: 40%**
	Continuous Evaluation through: Students are expected to attend each practical and submit the written practical of the previous session. Performing Practical and writeup submission will be continuous internal evaluation. 2.5 marks can be awarded for each practical performance and writeup submission totaling to 50 marks and can be converted to 20 marks.
13	**Semester End Examination: 60%**
	30 marks Semester End Practical Examination (2 hours duration)
14	**Format of Question Paper: (Semester End Examination : 30 Marks. Duration: 2 Hrs)** **Certified copy of Journal is compulsory to appear for the practical examination** Practical Slip: Q1. From Module 1 13 marks Q2. From Module 2 12marks Q3. Journal and Viva 05 marks

Syllabus
B.Sc. (Information Technology)
(Sem.- V)

Name of the Course: Android Practical

Sr.No.	Heading	Particulars
1	**Description the course : Including but Not limited to:**	This course introduces students to Android application development using Android Studio. It covers user interface design, Android components, layouts, intents, services, database integration, networking, background processing, media APIs, and security features to develop complete mobile applications. The course emphasizes hands-on practical implementation to build real-world, user-friendly, and data-driven Android applications.
2	**Vertical :**	Electives
3	**Type :**	Practical
4	**Credits :**	2 credits
5	**Hours Allotted :**	60
6	**Marks Allotted:**	50 Marks
7	**Course Objectives(CO):** CO 1. To understand the fundamentals of Android application development including project structure, components, and lifecycle management. CO 2. To design and implement user interfaces using various Android layouts and UI controls. CO 3. To develop applications using fragments, intents, events, listeners, and adapters for effective user interaction. CO 4. To implement background processing using services, threads, handlers, and broadcast receivers. CO 5. To develop data-driven applications using SQLite database and JSON-based network communication. CO 6. To implement advanced Android features such as media APIs, notifications, security, and runtime permission handling.	
8	**Course Outcomes (OC):** OC 1: Design and develop basic Android applications using Android Studio and implement Activity lifecycle methods. OC 2: Build responsive user interfaces using various Android layouts, UI controls, and resource files. OC 3: Implement fragments, intents, events, listeners, and adapters to enable effective user interaction and navigation. OC4: Develop applications incorporating services, broadcast receivers, notifications, and background processing mechanisms. OC 5: Create data-driven Android applications using SQLite database and JSON-based network communication. OC 6: Apply advanced Android features including media APIs, threading, security mechanisms, and runtime permission handling in real-world applications.	

9	**Modules**
	Module I: Android Fundamentals & UI Design
	Android Basics and Resources 1. Develop a simple "Hello World" Android application demonstrating project creation, Android Studio interface, Android components, AVD creation, and application execution. 2. Develop an Android application demonstrating the use of Programming Resources (colors.xml, themes.xml, strings.xml, dimens.xml, drawable resources) and apply them in UI components. 3. Develop an Android application demonstrating complete Activity Lifecycle methods and analyze execution using Logcat. **Fragments and Activity Interaction** 4. Develop an Android application to create and integrate a Fragment and demonstrate Fragment lifecycle methods. 5. Develop an Android application implementing Fragment-based UI navigation between multiple screens. 6. Develop an Android application demonstrating communication between Activity and Fragment. **Basic Layout Management** 7. Develop an Android application using LinearLayout (Horizontal and Vertical) to design structured UI. 8. Develop an Android application using RelativeLayout to align UI components dynamically. 9. Develop an Android application using TableLayout to design a structured Login/Registration Form. **Advanced Layout and View Handling** 10. Develop an Android application using ScrollView to display long scrollable content. 11. Develop an Android application using AbsoluteLayout demonstrating fixed positioning. 12. Develop an Android application using FrameLayout to overlay UI elements. **Adapter Based Views** 13. Develop an Android application using ListView with ArrayAdapter and implement item click handling. 14. Develop an Android application using GridView with Custom Adapter and image selection handling. 15. Develop an Android application implementing RecyclerView as a modern alternative to ListView.
	Module 2: Android System Components & Data Handling
	Menus, Dialogs and UI Interaction 1. Develop an Android application implementing AppBar (Toolbar) and Options Menu. 2. Develop an Android application demonstrating AlertDialog and custom DialogFragment. 3. Develop an Android application implementing contextual menus and popup menus. **Intents, Events and Listeners**

	4. Develop an Android application implementing Explicit and Implicit Intents. 5. Develop an Android application demonstrating Event Handling using Listeners (OnClick, OnLongClick, etc.). 6. Develop an Android application using Adapters to populate Spinner or ListView dynamically. **Services, Notifications and Broadcast Receivers** 7. Develop an Android application implementing Started and Bound Services. 8. Develop an Android application implementing Notifications using Notification Manager. 9. Develop an Android application implementing Static and Dynamic Broadcast Receivers. **Database and Networking** 10. Develop an Android application implementing SQLite database CRUD operations. 11. Develop an Android application integrating SQLite with ListView or RecyclerView. 12. Develop an Android application to fetch and parse JSON data from a web service and display it in UI components. **Concurrency, Media and Security** 13. Develop an Android application demonstrating Threads and Handlers for background processing. 14. Develop an Android application implementing Media APIs and Telephone APIs. 15. Develop an Android application implementing Runtime Permissions and security handling in AndroidManifest.xml.
10	**Text Books** 1. Phillips, Bill; Stewart, Chris; Marsicano, Kristin Hardy, Android Programming: The Big Nerd Ranch Guide, Big Nerd Ranch Guides, 5th Edition, 2022. 2. Meier, Reto, Professional Android, Wrox Publishing, 4th Edition, 2018. 3. Griffiths, Dawn; and Griffiths, David, Head First Android Development, O'Reilly Media, 2nd Edition, 2017. 4. Haseman, Chris; and Sutton, Kevin, Learning Android Application Development, Addison-Wesley Professional, 1st Edition, 2020. 5. Darwin, Ian F., Android Cookbook: Problems and Solutions for Android Developers, O'Reilly Media, 2nd Edition, 2017.

11	**Reference Books** 1. Subramanian, S.; and Bali, Rachit, Android Application Development: Basics to Advanced, McGraw-Hill Education, 1st Edition, 2021. 2. Murphy, Shaun, Android Studio 4 Development Essentials – Java Edition, Payload Media, 10th Edition, 2021. 3. Sayed Hashimi, and Satya Komatineni, Mastering Android Application Development, Wiley, 2nd Edition, 2018. 4. Kotlinlang.org, Kotlin for Android Developers (Official Guide), JetBrains Documentation, 2024 (Online Reference). 5. Google Developers Documentation Team, Android Developers Official Guide, Google Inc., 2024 (Online Reference).
12	**Internal Continuous Assessment: 40%**
	Continuous Evaluation through: Students are expected to attend each practical and submit the written practical of the previous session. Performing Practical and writeup submission will be continuous internal evaluation. 2.5 marks can be awarded for each practical performance and writeup submission totaling to 50 marks and can be converted to 20 marks.
13	**Semester End Examination: 60%**
	30 marks Semester End Practical Examination (2 hours duration)
14	**Format of Question Paper: (Semester End Examination : 30 Marks. Duration: 2 hrs)** **Certified copy of Journal is compulsory to appear for the practical examination** **Practical Slip:** Q1. From Module 1 13 marks Q2. From Module 2 12marks Q3. Journal and Viva 05 marks

Syllabus
B.Sc. (Information Technology)
(Sem.- V)

Name of the Course: Full Stack development MERN Practical

Sr.No.	Heading	Particulars
1	**Description the course :** **Including but Not limited to:**	This practical course provides structured training in full-stack web application development using the MERN stack (MongoDB, Express, React, and Node.js). The course is designed to develop the fundamental skills required to build dynamic, data-driven web applications by integrating frontend and backend technologies in a systematic manner. The course begins with environment setup and core JavaScript programming concepts, forming the foundation for frontend and backend development. Learners progressively build user interfaces using React, implement components with props and state, manage routing, and handle data from external APIs. In the backend module, students develop server-side applications using Node.js and Express, create RESTful APIs, and perform database operations using MongoDB. Emphasis is placed on hands-on implementation of CRUD operations, API creation and integration, data handling, and validation techniques. Learners will gain practical exposure to connecting the frontend with backend services and executing a complete data flow from the user interface to the database. Installation and configuration of development tools, JavaScript fundamentals and ES6 features, React components and state management, routing and API handling, Express server development, REST API implementation, MongoDB CRUD operations, frontend-backend integration, and validation and execution of a complete MERN-based web application.
2	**Vertical :**	Practical
3	**Type :**	2 credits (1 credit = 30 Hours of Practical work in a semester)
4	**Credits :**	60 Hours
5	**Hours Allotted :**	50 Marks
6	**Marks Allotted:**	Practical

7	**Course Objectives(CO):** **CO1:** To install and configure the MERN development environment and execute basic JavaScript programs using Node.js. **CO2:** To apply fundamental JavaScript concepts such as operators, functions, arrays, objects and control structures in program development. **CO3:** To develop basic user interfaces using React and understand component-based architecture. **CO4:** To implement React components using props, state and event handling for interactive applications. **CO5:** To implement routing and handle data from external APIs in React applications. **CO6:** To develop simple server-side applications and APIs using Node.js and Express. **CO7:** To perform database creation and CRUD operations using MongoDB and connect it with Node.js. **CO8:** To integrate frontend and backend for sending, receiving and displaying data. **CO9:** To implement data management operations such as insert, update and delete using the MERN stack. **CO10:** To execute, test and validate the complete workflow of a full stack web application.
8	**Course Outcomes (OC):** **OC1:** Learners will be able to install required development tools and run basic JavaScript programs using Node.js. **OC2:** Learners will be able to develop JavaScript programs using essential programming constructs and functions. **OC3:** Learners will be able to create and execute basic React applications using functional components. **OC4:** Learners will be able to design interactive React components using props, state and events. **OC5:** Learners will be able to develop multi-page React applications and display data retrieved from APIs. **OC6:** Learners will be able to create simple backend servers and APIs using Express and Node.js. **OC7:** Learners will be able to create databases and perform CRUD operations using MongoDB. **OC8:** Learners will be able to connect React frontend with backend APIs and display dynamic data. **OC9:** Learners will be able to perform CRUD operations through integrated frontend and backend applications. **OC10:** Learners will be able to test and execute a complete MERN-based application ensuring proper data flow and validation.
9	**Modules:-**
	Module 1: Frontend Development using React

	1.**Environment Setup and JavaScript Basics** a. Install Node.js and Visual Studio Code, and verify installation. b. Create a simple Node.js file and run it using the terminal. c. Write a Node.js script demonstrating basic JavaScript commands (console log, variables, and arithmetic operations). d. Create and run a simple program using variables and functions. e. Demonstrate use of operators and control statements in JavaScript f. Write programs using loops and template literals. 2. **Core JavaScript Concepts for MERN** a. Create functions and call them with parameters. b. Work with arrays and objects in JavaScript. c. Demonstrate use of arrow functions. d. Write a program using array methods (map, filter, reduce). e. Create a program that manipulates objects and displays output. f. Implement a small JavaScript program combining functions, arrays, and objects. 3.**React Introduction and JSX** a. Create a React application using Vite/CRA and display "Welcome to MERN Stack Development". b. Display dynamic data using JSX. c. Declare variables and write a function to calculate the sum of two numbers and display the result. d. Apply basic CSS styling to a React component e. Create a simple functional component and render it in App.js. f. Create a React component that displays the current date and time dynamically using JavaScript. 4.**React Components, Props, and State** a. Create multiple components and render them on a single page. b. Pass data between components using props c. Use the useState hook to manage and update data. d. Create a form component to accept user input and display it dynamically. e. Display a list of items stored in the state using map(). f. Build a small interactive component (e.g., counter or to-do list). 5.**React Routing and Data Handling** a. Install and configure React Router in the React application. b. Create multiple pages and implement navigation between them. c. Use Link and Routes to move between pages. d. Fetch sample data from a public API using Axios. a. Display fetched data in table or list format. b. Implement search or filter functionality on displayed data.
	Module 2: Backend Development Using Node.js, Express, and MongoDB

	6.**Node.js and Express Fundamentals** a. Install Express and create a simple server. b. Display a message on the browser using the server response. c. Create different routes in Express. d. Send a JSON response from the server. e. Test routes using browser or Postman. f. Run the server and observe the output in the console. 7.**MongoDB Basics** a. Install and start MongoDB. b. Create a database and collection. c. Insert records in MongoDB. d. View records from the collection. e. Update and delete records. f. Connect MongoDB with Node.js. 8.**API Creation and Integration** a. Create a simple REST API using Express b. Implement a GET API to retrieve data. c. Fetch data from the API in a React application. d. Send data from React to the server using a POST request. e. Store data in MongoDB. f. Display stored data in React page. 9.**User Data Management** a. Create an API to add user details using Express and MongoDB. b. Design a React form to send user data to the backend. c. Retrieve and display user data from MongoDB. d. Implement an update operation for existing records. e. Implement the delete operation for selected records. f. Display updated records dynamically in React. 10.**Full Stack Execution and Validation** a. Integrate React frontend with Node and Express backend. b. Perform insert, update, and delete operations through UI. c. Display database records dynamically in React. d. Implement frontend form validation. e. Test all APIs using Postman. f. Execute and verify the complete working of the frontend, backend, and database together.
10	**Text Books**

11	**Reference Books** 1. Web Development with Node and Express. Ethan Brown. O'Reilly Media 2nd Edition, 201 4. Learning React Alex Banks & Eve Porcello. O'Reilly Media 2nd Edition, 2020 5. MongoDB: The Definitive Guide Shannon Bradshaw, Eoin Brazil & Kristina Chodorow O'Reilly Media 3rd Edition, 2019 6. Ultimate Full-Stack Web Development with MERN Nabendu Biswas BPB Publications 1st Edition, 2022 7. Full Stack Development with MongoDB, Express, Angular and Node Colin J. Ihrig & Adam Bretz McGraw Hill (TMH) 2nd Edition, 2018
12	**Internal Continuous Assessment: 40%**
	Continuous Evaluation through: Students are expected to attend each practical and submit the written practical of the previous session. Performing Practical and writeup submission will be continuous internal evaluation. 2.5 marks can be awarded for each practical performance and writeup submission totaling to 50 marks and can be converted to 20 marks.
13	**Semester End Examination: 60%**
	30 marks Semester End Examination
14	**Format of Question Paper: (Semester End Examination : 30 Marks. Duration: 2 hours)** **Certified copy of Journal is compulsory to appear for the practical examination** **Practical Slip:** Q1. From Module 1 13 marks Q2. From Module 2 12marks Q3. Journal and Viva 05 marks

Syllabus
B.Sc. (Information Technology)
(Sem.- V)

Name of the Course: Linux Administration Practical

Sr.No.	Heading	Particulars
1	**Description the course :** **Including but Not limited to:**	This course introduces learners to the fundamental and applied concepts of Linux system administration through extensive hands-on laboratory practice. Students will install, configure, secure, and manage a Linux operating system environment using virtualization platforms. The course focuses on developing practical abilities required for entry-level roles such as System Administrator The laboratory-oriented pedagogy enables learners to work in real-world computing
2	**Vertical :**	Elective Major
3	**Type :**	Practical
4	**Credits :**	2 credits
5	**Hours Allotted :**	60 Hrs
6	**Marks Allotted:**	50 Marks
7	**Course Objectives(CO):** CO1. Understand the architecture, features, and working environment of the Linux operating system and perform installation using virtualization tools. CO2. Use Linux command-line interface effectively to manage files, directories, text processing, permissions, and software packages. CO3. Administer users, groups, file systems, and storage devices while applying appropriate access control and security practices. CO4. Configure essential networking services including IP addressing, SSH remote access, web server, and network file sharing. CO5. Monitor system performance, manage processes, implement firewall rules, and troubleshoot common system issues. CO6. Automate administrative tasks using shell scripting and scheduling tools such as cron and at.	
8	**Course Outcomes (OC):** OC 1. Install and configure a Linux operating system and work in graphical and command-line environments. OC 2. Perform file handling, text processing, and package management using standard Linux utilities and commands. OC 3. Create and manage users, groups, permissions, and access control lists to secure system resources. OC 4. Configure networking, remote login (SSH), and basic server services such as Apache web server and network storage. OC 5. Analyse and manage processes, partitions, logical volumes, and system performance.	

	OC 6. Write basic shell scripts and schedule automated tasks to manage and monitor system operations.
9	**Modules:-**
	Module 1:
	1. a. Installing Red Hat Enterprise Linux (Virtual Box/ VmWare) b. Linux Graphical Environment i. Display/Login Manager ii. Desktop Environment iii. Linux Directory Structure and File Systems 2. Basic System Commands a. Use input-output redirection (>, >>, |, 2>, etc) b. Use grep and regular expressions to analyse text c. Log in and switch users in multi-user targets d. Working with Text Files cat,cut,less,hesd,tail,sort,wc,grep, awk 3. Essential File Management Tools a. Managing Files ,Using Links b. Working with Archives and Compressed Files c. Create, delete, copy, and move files and directories d. Archive, compress, unpack, and uncompress files using tar, star, gzip, and bzip2 e. Create hard and soft links f. Managing File Ownership, Managing Basic Permissions g. Managing Advanced Permissions, Managing ACLs h. Setting Default Permissions with umask i. Working with User-Extended Attributes j. List, set, and change standard ugo/rwx permissions k. Create and configure set-GID directories for collaboration l. Create and manage access control lists 4. Working with Users, Groups, and Permissions a. Creating and Managing Users b. Creating and Managing Groups c. Using Permissions and Advanced Permissions d. Change passwords and adjust password aging for local user accounts e. Configure superuser access -> su,sudo,root 5. Managing Software a. Managing Software Packages with YUM b. Using yum c. Managing Software Packages with RPM d. Install and update software packages repository from the local file system
	Module 2:
	6. Managing Processes a. Introduction to Process Management b. Managing Shell Jobs

	c. Using Common Command-Line Tools for Process Management d. Using top to Manage Processes e. Using tuned to Optimize Performance f. Identify CPU/memory-intensive processes and kill processes 7. Managing Storage a. Understanding MBR and GPT Partitions b. Managing Partitions and File Systems c. Mounting File Systems d. List, create, delete partitions on MBR and GPT disks e. Configure systems to mount file systems at boot by universally unique ID (UUID) or label f. Add new partitions and logical volumes, and swap to a system non-destructively g. Create, mount, unmount, and use vfat, ext4, and xfs file systems 8. Configuring Networking a. Managing Network Addresses and Interfaces b. Validating Network Configuration c. Managing Network Configuration with nmtui and nmcli d. Configure IPv4 and IPv6 addresses e. Configure DNS-hostname resolution 9. Configuring SSH a. Using SSH and Related Utilities b. Access remote systems using SSH c. Configuring Key-Based Authentication with Passphrases d. Configure key-based authentication for SSH 10. Configuring web server a. Configuring a Basic Apache Server b. Understanding Apache Configuration Files c. Creating Apache Virtual Host 11. Configure Firewall a. Linux Firewall with iptables b. Working with Firewalld 12. Accessing Network Storage a. Using NFS Services b. Using CIFS Services c. Mounting Remote File Systems Through fstab d. Using Automount to Mount Remote File Systems 13. Creating Shell scripts a. automate or monitor tasks/processes b. Getting and setting system information 14. Scheduling Tasks a. Configuring Cron to Automate Recurring Tasks b. Configuring At to Schedule Future Tasks c. Schedule tasks using at and cron
10	Text Books
11	Reference Books 4. Red Hat RHCA 8 Cert Guide by Sander van Vugt

	5. RHCSA Red Hat Enterprise Linux 8 Training and Exam Preparation Guide (EX200) by Asghar Ghori 6. Red Hat Enterprise Linux 8 Administration - Master Linux Administration Skills and Prepare for the RHCSA Certification Exam (Packt, 2021)
12	**Internal Continuous Assessment: 40%**
	Continuous Evaluation through: Students are expected to attend each practical and submit the written practical of the previous session. Performing practical and writeup submission will be continuous internal evaluation. 2.5 marks can be awarded for each practical performance and writeup submission totalling to 50 marks and can be converted to 20 marks
13	**Semester End Examination: 60%**
	30 marks Semester End Examination
14	**Format of Question Paper: (Semester End Examination : 30 Marks. Duration: 2Hr)** **Certified copy of Journal is compulsory to appear for the practical examination Practical Slip:** Q1. From Module 1 13 marks Q2. From Module 2 12marks Q3. Journal and Viva 05 marks

AC – 25/03/2026
Item No. – 8.4 (N) (4)

As Per NEP 2020

University of Mumbai

Syllabus for Minor Vertical 2	
Faculty of Faculty of Commerce & Management	
Board of Studies in Bachelor of Management Studies	
Third Year Programme in Minor B.Com. (Management Studies)	
Semester	**V**
Title of Paper	Industry & Service management IV **(Production & Design thinking Management)**
Credits	**4**
Semester	**VI**
Title of Paper	Industry & Service management V **(Integrity and Sustainability Management)**
Credits	**4**
From the Academic Year	**2026-27**

Sem. - V

Syllabus
B.Com. Management Studies
(Sem.- V)

Title of Paper: Industry & Service management IV
(Production & Design thinking Management)

Sr. No.	Heading	Particulars
1	**Description the course :** **Including but Not limited to :**	This course provides a dual perspective on managing complex manufacturing systems and high-engagement service environments. It explores the evolution of operational philosophies from traditional production models to digital-first, service-driven architectures. Students will master technical skills in process planning, quality systems like Six Sigma, and the visualization of customer journeys through service blueprinting. A unique feature of this curriculum is the integration of Design Thinking for human-centric problem-solving and prototyping. Finally, the course addresses the impact of Industry 4.0, AI, and sustainable practices in transforming modern industrial and service landscapes.
2	**Vertical : 3**	Minor
3	**Type :**	Theory
4	**Credit**:	4 credits
5	**Hours Allotted :**	60 Hours
6	**Marks Allotted**:	100 Marks
7	**Course Objectives**: • To understand the structural distinctions and strategic frameworks governing both industrial production and service delivery systems. • To equip students with tools for Production Planning, Lean Management, and Six Sigma to optimize resource utilization and quality. • To develop a human-centric innovation mindset through the application of Design Thinking, Empathy Mapping, and Iterative Prototyping. • To analyze the role of digital transformation (Industry 4.0, AI, and IoT) in creating resilient, ethical, and sustainable operational roadmaps.	
8	**Course Outcomes**: • Students will be able to categorize and design different Production Systems and Process Strategies based on volume and variety requirements. • Students will demonstrate proficiency in creating Service Blueprints to identify potential fail points and optimize customer touchpoints. • Students will be capable of applying Lean and Six Sigma basics to eliminate waste and reduce process variability in industrial workflows. • Students will be able to facilitate Design Thinking workshops using ideation techniques like SCAMPER and Empathy Mapping.	

	• Students will understand the integration of Industry 4.0 technologies (Digital Twins, IoT, and Cloud) in modern manufacturing and service models.
9	**Modules**:
	Module 1: Foundations of Industry and Service Operations
	• Fundamentals of Industry and Service Systems: Conceptual frameworks and structural distinctions in operational environments • Evolution of Production and Service Management: Historical progression and contemporary shifts in operational philosophies • Types of Production Systems and Process Strategies: Comparative study of manufacturing models and strategic operational approaches • Service Delivery Models and Customer Interaction Patterns: Design of service encounters and behaviour-driven service architecture
	Module 2: Process Planning, Quality Systems, and Service Excellence
	• Production Planning and Control: Synchronizing resources, scheduling and monitoring industrial workflows, Process Layout, Workflow Design, and Capacity Utilization • Quality Management, Lean Systems, and Six Sigma Basics: Structured frameworks for defect reduction, process precision, and waste elimination • Service Blueprinting and Customer Journey Mapping: End-to-end visualization of service experience and performance touchpoints • Service Failure, Recovery, and Performance Improvement: Strategies for managing breakdowns and building resilient service standards
	Module 3: **Design Thinking and Creative Problem-Solving Frameworks**
	• Introduction to Design Thinking: Human-centric innovation mindset and solution-building philosophy • Empathy Mapping and Problem Framing: Decoding user needs and defining actionable problem statements • Ideation Techniques: Brainstorming, SCAMPER, Creative Modeling, Tools for expansive thinking and conceptual exploration • Prototyping, Experimentation, and Iterative Feedback: Transforming concepts into testable models for refinement and validation
	Module 4: **Technology, Transformation, and Contemporary Industrial Practices**
	• Technology in industry: Introduction to digital transformation and its impact on business and industry, Industry 4.0, Automation , Digital Operations, • Service transformation: AI, Data Analytics, Smart Systems, cloud computing, IoT, • Sustainable operations and Ethical Service Practices, Developing digital strategies, roadmaps, and delivering business value through technology. • Case studies: Manufacturing, Retail, Healthcare, Hospitality and Public Service Models
10	**Reference Books:** • *Brown, T. (2019). Change by design: How design thinking transforms organizations and inspires innovation. Harper Business.*

	• *Chase, R. B., Shankar, R., & Jacobs, F. R. (2023). Operations and supply chain management (16th ed.). McGraw Hill.* • *Fitzsimmons, J. A., & Fitzsimmons, M. J. (2024). Service management: Operations, strategy, information technology (10th ed.). McGraw Hill.* • *Schwab, K. (2017). The fourth industrial revolution. Portfolio Penguin.* • *Shostack, G. L. (1984). Designing services that deliver. Harvard Business Review.* • *Aswathappa, K., & Bhat, K. S. (2021). Production and operations management (2nd ed.). Himalaya Publishing House.* • *Besterfield, D. H. (2023). Total quality management (5th ed.). Pearson Education.* • *Buffa, E. S., & Sarin, R. K. (2017). Modern production/operations management (8th ed.). Wiley India.* • *Heizer, J., Render, B., & Munson, C. (2023). Operations management: Sustainability and supply chain management (14th ed.). Pearson.* • *Johnston, R., Clark, G., & Shulver, M. (2022). Service operations management: Improving service delivery (5th ed.). Pearson.* • *Lovelock, C. H., Wirtz, J., & Chatterjee, J. (2024). Services marketing: People, technology, strategy (9th ed.). Pearson.* • *Nahmias, S., & Olsen, T. L. (2022). Production and operations analysis (8th ed.). Waveland Press.* • *Norman, D. (2013). The design of everyday things (Revised & expanded ed.). Basic Books.* • *Slack, N., & Lewis, M. (2023). Operations strategy (7th ed.). Pearson.* • *Womack, J. P., & Jones, D. T. (2021). Lean thinking: Banish waste and create wealth in your corporation (2nd ed.). Free Press.*	
11	**Internal Continuous Assessment: 40%**	**External, Semester End Examination 60%**
	Individual Passing in Internal and External Examination	
12	**Continuous Evaluation through:** Quizzes, Class Tests, presentation, project, role play, creative writing, assignment etc.(at least 3)	

QUESTION PAPER PATTERN
(External and Internal)
Paper Pattern 4 Credits (Total 100 Marks)

Internal = 40 Marks
External = 60 Marks

Internal Paper Pattern (40 Marks)

1. Case Study writing **OR** Assignment
2. Quiz **OR** Group discussion **OR** Role Playing
3. Project Presentation **OR** Research Paper

} any two (10 Marks each) 20 Marks

4. Class Test - (Mandatory) with Objective questions 20 Marks

Total 40 Marks

External Paper Pattern (60 Marks)

External Paper Pattern (60 Marks)

Write any **FOUR** questions from the following

Q1. Answer the following 15 marks

A

B

Q2. Answer the following 15 Marks

A

B

Q3. Answer the following 15 Marks

A

B

Q4. Answer the following 15 marks

A

B

Q5. Answer the following 15 Marks

A

B

Q6. Answer the following 15 Marks

A

B

Vertical - 4

VSC
(2)
(Practical)

Syllabus
B.Sc. (Information Technology)
(Sem.- V)

Name of the Course: Internet of Things Practical

Sr.No.	Heading	Particulars
1	**Description the course : Including but Not limited to:**	This IoT practical course uses Raspberry Pi to build real-world projects with sensors, displays, security modules, GPS, ADCs, relays, and cameras. Students learn Python-based interfacing, automation, data acquisition, and MQTT communication through hands-on exercises, gaining strong foundational skills for developing functional IoT systems and smart applications.
2	**Vertical :**	Electives
3	**Type :**	Theory / Practical
4	**Credits :**	2 credits (1 credit = 1 30 Hours of Practical work in a semester)
5	**Hours Allotted :**	60 Hours
6	**Marks Allotted:**	50 Marks
7	**Course Objectives(CO):** CO 1. To introduce students to the fundamentals of Internet of Things (IoT) through hands-on practical experiments using Raspberry Pi. CO 2. To develop skills in interfacing digital, analog, and communication modules such as sensors, RFID, fingerprint readers, GPS, relays, and displays. CO 3. To enable students to write Python programs for device control, sensor data processing, automation, and real-time applications. CO 4. To familiarize learners with IoT communication protocols such as MQTT for data exchange and remote monitoring. CO 5. To build confidence in assembling complete IoT systems combining hardware, software, and networking components. CO 6. To cultivate problem-solving and prototyping skills essential for designing IoT-based smart solutions.	
8	OC 6. Course Outcomes (OC): OC 7. After completing this course, students will have: OC 8. Ability to Interface and Control Digital & Analog Hardware Components. OC 9. Competence in Implementing Real-World IoT Communication and Automation. OC 10. Skills in Reading, Processing, and Displaying Sensor Data OC 11. Ability to Implement Identity & Security Mechanisms OC 12. Capability to Design and Deploy an Integrated IoT-Based Smart System	

	OC 13. . Ability to Integrate and Program Multiple Hardware Interfaces for Security and Automation OC 14. Proficiency in Designing Multi-Layer Authentication and Access Control Systems OC 15. Competence in IoT Communication and Web-Based Control of Embedded Devices OC 16. Skills in Implementing Intelligent Monitoring Systems Using Raspberry Pi OC 17. Capability to Deploy, Test, and Evaluate Complete Embedded Solutions on Raspberry Pi
9	**Modules:-**
	Module 1:
	1. Create different blinking patterns with LEDs by connecting to GPIO pints of Raspberry Pi. 2. Use Telegram to control Raspberry Pi GPIO. 3. Interface TM1637 with Raspberry Pi and Display current time. 4. Interface GPS module with Raspberry Pi. 5. Interface ADS 1115 with Raspberry Pi to act as oscilloscope. 6. Interface 16x2 LCD with Raspberry Pi using I2C module to display text. 7. Interface RFID module with Raspberry Pi to read and write to RFID tags/cards. 8. Interface Fingerprint Sensor with Raspberry Pi. 9. Interface Pi Camera with Raspberry Pi to capture image and video. 10. Design IoT based home automation system using Raspberry Pi

	Module 2: 11. Interface Fingerprint Sensor and 16X2 LCD with Raspberry and display message RFID UID matched or unmatched. 12. Simulate door lock by interfacing fingerprint sensor and relay moule with raspberry Pi. 13. Implement basic Publisher – Subscriber using MQTT. Publish messages using paho mqtt. Subscribe and control LED. 14. Design a web dashboard using Flask. Control relay module via web. 15. Create two factor authentication system by interfacing fingerprint sensor, RFID and relay with Raspberry Pi. 16. Create a motion detector by interfacing camera and 16X2 display with Raspberry Pi. Print the appropriate message on the display. 17. Design and implement a Node-RED flow on Raspberry Pi that integrates a physical push-button and an LED with a web dashboard. The flow must allow users to toggle the LED from the dashboard and via the physical button. A button press should force the LED ON for 5 seconds and then automatically restore the LED to the dashboard's last state. 18. Write a program to read the CPU%, Memory% and CPU temperature of Raspberry Pi. 19. Design and implement an embedded access-control prototype that grants entry only when either a whitelisted RFID tag is detected or a legitimate fingerprint is verified, provides real-time visual feedback on an LCD, indicates status via an LED, and pulses a relay (active-LOW) to unlock for a fixed interval. The system must log all events (time, method, result) for auditability. 20. Install Windows IoT 10 core on Raspberry pi and run the demo program.

10 and 11	**Reference Books:** 1. Programming the Raspberry Pi: Getting Started with Python — Simon Monk, McGraw-Hill, 3rd ed. 2. Raspberry Pi Cookbook — Simon Monk, O'Reilly, 2nd ed. 3. Exploring Raspberry Pi — Derek Molloy, Wiley. 4. Getting Started with Python on Raspberry Pi — (Raspberry Pi Press / Foundation) 5. Exploring Arduino — Jeremy Blum, 2nd ed., Apress. 6. Practical Electronics for Inventors — Paul Scherz & Simon Monk, McGraw-Hill, 4th ed. 7. **RFID Handbook: Fundamentals and Applications** — Klaus Finkenzeller, Wiley, 3rd ed. 8. **Near Field Communication (NFC): From Theory to Practice** — Vedat Coskun et al., Wiley, 2nd ed. 9. **Handbook of Fingerprint Recognition** — Davide Maltoni et al., Springer, 2nd ed. 10. **Understanding GPS/GNSS: Principles and Applications** — Elliott D. Kaplan & Christopher Hegarty, Artech House 11. **GPS for Land Surveyors** — Jan Van Sickle, CRC Press. 12. **Programming the Raspberry Pi: Picamera2 and libcamera Guides** — (Foundation/Pi Press articles, community books) 13. **MQTT Essentials – A Lightweight IoT Protocol** — Gastón C. Hillar (Packt) 14. **Node-RED: A visual tool for wiring the Internet of Things** — Nick O'Leary & Dave Conway-Jones 15. **Flask Web Development** — Miguel Grinberg, O'Reilly, 2nd ed. *16.* **Designing Connected Products** — Claire Rowland et al., O'Reilly.
12	**Internal Continuous Assessment: 40%**
	Students are expected to attend each practical and submit the written practical of the previous session. Performing Practical and writeup submission will be continuous internal evaluation. 2.5 marks can be awarded for each practical performance and writeup submission totalling to 50 marks and can be converted to 20 marks..
13	**Semester End Examination: 60%**
	30 marks Semester End Examination
14	**Format of Question Paper: (Semester End Examination: 30 Marks. Duration: 2 hours)** Q1. From Module 1 13 marks Q2. From Module 2 12marks Q3. Journal and Viva 05 marks

Vertical-6

As Per NEP 2020

University of Mumbai

Syllabus for Community Engagement Project (CEP) Vertical-6	
Faulty of Science & Technology	
Board of Studies in Information Technology	
Name of the Programme – B.Sc. (Information Technology)	
Semester	**V**
Credit	**2**
Duration	**30 hrs (Field Work+ Survey) + 15hrs (Discussion + Report Writing) : Total - 45 hrs**
From the Academic Year	**2026-27**

Name of Faculty:- Science & Technology
Name of Programme :- B.Sc. (Information Technology)
Duration :- 30 hrs (Field Work+ Survey) + 15hrs (Discussion + Report Writing) : Total - 45 hrs

Indicative Topics for CEP

Sr. No.	Name of the Topic
1.	Digital Literacy Training for Senior Citizens
2.	Basic Computer Skills for Rural School Students
3.	Smartphone Usage & Online Safety for Women Self-Help Groups
4.	Digital Awareness Program for Persons with Disabilities
5.	Teaching Internet Basics to First-Generation Learners
6.	Bridging the Digital Divide in Urban Slum Communities
7.	Multilingual Digital Literacy Content Development
8.	Helping Small Vendors Go Digital (UPI, QR, Apps)
9.	Cyber Awareness Campaign for Rural Communities
10.	Digital Skills Bootcamp for Unemployed Youth
11.	Cybercrime Awareness and Prevention Workshops
12.	Social Media Safety & Digital Footprint Awareness
13.	Phishing, Scam & Fraud Detection Awareness Program
14.	Password Hygiene and Personal Data Protection Campaign
15.	Cyber Ethics and Responsible Internet Usage Program
16.	Online Safety for Children and Adolescents
17.	Digital Privacy Awareness for Community Organizations
18.	Secure Online Banking Awareness Programme
19.	Community Helpdesk for Cyber Safety Queries
20.	Fake News Detection and Media Literacy Programme
21.	Assisting Citizens in Using Government Digital Portals
22.	Awareness Program on Digital India Initiatives
23.	Helping Communities Access Online Healthcare Services
24.	Training on DigiLocker, Aadhaar & PAN Services
25.	Online Grievance Redressal Systems Awareness
26.	Digital Tax Filing Support for Small Businesses

27.	Voter Awareness Using Digital Platforms
28.	Digital Skill Support for Local Government Offices
29.	Smart Village / Smart City IT Enablement Project
30.	Creating Digital Guides for Public Welfare Schemes
31.	Developing E-Learning Content for Government Schools
32.	Setting Up Virtual Classrooms for Rural Schools
33.	Coding Awareness Program for School Children
34.	STEM Education Support Using Free Digital Tools
35.	Educational App Development for Local Language Learning
36.	Digital Library Creation for Community Centers
37.	Online Career Guidance Portal for Underprivileged Students
38.	Learning Management System (LMS) Setup for NGOs
39.	Assistive Learning Tools for Differently-Abled Students
40.	Digital Assessment Tools for Community Schools
41.	Health Awareness Campaign Using Mobile Applications
42.	Telemedicine Awareness and IT Support Program
43.	Environmental Monitoring Using Sensors & Data Analytics
44.	Waste Management Tracking System for Local Bodies
45.	Air & Water Quality Awareness Using Data Visualization
46.	IT-Based Disaster Preparedness and Alert Systems
47.	Nutrition & Health Tracking App for Community Use
48.	Digital Record System for Community Health Workers
49.	Smart Farming Advisory System for Farmers
50.	Climate Change Awareness Using Interactive Dashboards
51.	Website & Digital Presence Development for NGOs
52.	IT Support for Self-Help Groups and Micro-Entrepreneurs
53.	Digital Marketing Training for Small Businesses
54.	Inventory & Billing Software for Local Shops
55.	Data Management Solutions for Non-Profit Organizations
56.	Crowdfunding Platform Development for Social Causes
57.	Social Media Strategy for Community-Based Organizations
58.	Technology Support for Rural Startups
59.	Building Open-Source Solutions for Social Problems
60.	Community-Based Hackathon for Social Innovation

The topics are indicative and the faculty members should allot Community Engagement Project that are relevant and important as per core Subject. The Community Engagement Project may be taken individual or in a group up to 4 students with proper guidance from Faculty.

Evaluation Pattern:-

Evaluation during CEP Program involves two key components :-

External Evaluation 60%

Internal Evaluation 40%

Evaluation Chart

(i) External Evaluation (Marks 30)

Criteria	Marks
Objectives, Literature Review , Methodology, Data Analysis, Conclusion and Recommendations	15
Overall Project Report Structure and Style	5
Presentation Skills & Communication	10
Total	30

(ii) Internal Evaluation by Guide (Marks 20)

Criteria	Marks
Attendance, Community interactions completion and interaction with Supervisor	10
Overall Report quality	10
Total	20

*** Please see the Guidelines for Community Engagement Project for UG Students, as per NEP 2020**

Sd/-
Sign of the BOS
Dr. R. Srivaramangai
Chairman/Co-ordinator
BOS/Ad-hoc BOS in Information Technology

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

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