

University of Mumbai



No. AAMS_UGS/ICC/2024-25/20

CIRCULAR:-

Attention of the Principals of the Affiliated Colleges and Directors of the Recognized Institutions in Faculty of Science & Technology is invited to this office circular No.UG/76 of 2018-19 dated 6th July, 2018 relating to the revised syllabus as per (CBCS) for the T.Y.B.Sc in Information Technology (Sem – V & VI).

They are hereby informed that the recommendations made by the Ad-hoc **Board of Studies in Information Technology** at its meeting held on 03rd June, 2024 and subsequently passed by the Board of Deans at its meeting held on 27th June, 2024 **vide** item No. 6.6 (R) have been accepted by the Academic Council at its meeting held on 28th June, 2024 **vide** item No.6.6 (R) and that in accordance therewith, the **revised syllabus** as per the (CBCS) for the **T.Y.B.Sc. in Information Technology (Sem.-V & VI)**, has been brought into force with effect from the academic year 2024-25.

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

MUMBAI – 400 032

24th July, 2024

Baliram
(Prof.(Dr) Baliram Gaikwad)
I/c Registrar

To

The Principals of the Affiliated Colleges and Directors of the Recognized Institutions in Faculty of Science & Technology.

A.C/6.6(R)/28/06/2024

Copy forwarded with Compliments for information to:-

- 1) The Dean, Faculty of Science & Technology,
- 2) The Chairman, Ad-hoc Board of Studies in Information Technology,
- 3) The Director, Board of Examinations and Evaluation,
- 4) The Director, Board of Students Development,
- 5) The Director, Department of Information & Communication Technology.
- 6) The Co-ordinator, MKCL.
- 7) The Deputy Registrar, Admissions, Enrolment, Eligibility & Migration Department (AEM),
- 8) The Deputy Registrar, Result Unit,
- 9) The Deputy Registrar, College Affiliations Development Department (CAD)

Copy to :-

- 1. The Deputy Registrar, Academic Authorities Meetings and Services (AAMS),**
- 2. The Deputy Registrar, College Affiliations & Development Department (CAD),**
- 3. The Deputy Registrar, (Admissions, Enrolment, Eligibility and Migration Department (AEM),**
- 4. The Deputy Registrar, Research Administration & Promotion Cell (RAPC),**
- 5. The Deputy Registrar, Executive Authorities Section (EA),**
- 6. The Deputy Registrar, PRO, Fort, (Publication Section),**
- 7. The Deputy Registrar, (Special Cell),**
- 8. The Deputy Registrar, Fort/ Vidyanagari Administration Department (FAD) (VAD), Record Section,**
- 9. The Director, Institute of Distance and Open Learning (IDOL Admin), Vidyanagari,**

They are requested to treat this as action taken report on the concerned resolution adopted by the Academic Council referred to in the above circular and that on separate Action Taken Report will be sent in this connection.

- 1. P.A to Hon'ble Vice-Chancellor,**
- 2. P.A Pro-Vice-Chancellor,**
- 3. P.A to Registrar,**
- 4. All Deans of all Faculties,**
- 5. P.A to Finance & Account Officers, (F.& A.O),**
- 6. P.A to Director, Board of Examinations and Evaluation,**
- 7. P.A to Director, Innovation, Incubation and Linkages,**
- 8. P.A to Director, Board of Lifelong Learning and Extension (BLLE),**
- 9. The Director, Dept. of Information and Communication Technology (DICT) (CCF & UCC), Vidyanagari,**
- 10. The Director of Board of Student Development,**
- 11. The Director, Department of Students Welfare (DSD),**
- 12. All Deputy Registrar, Examination House,**
- 13. The Deputy Registrars, Finance & Accounts Section,**
- 14. The Assistant Registrar, Administrative sub-Campus Thane,**
- 15. The Assistant Registrar, School of Engg. & Applied Sciences, Kalyan,**
- 16. The Assistant Registrar, Ratnagiri sub-centre, Ratnagiri,**
- 17. The Assistant Registrar, Constituent Colleges Unit,**
- 18. BUCTU,**
- 19. The Receptionist,**
- 20. The Telephone Operator,**
- 21. The Secretary MUASA**

for information.

University of Mumbai



Revised Syllabus for
T.Y.B.Sc. (Information Technology)
Semester – (Sem V)
(Choice Based Credit System)

(With effect from the academic year 2024-25)

University of Mumbai



Syllabus for Approval

O: _____	Title of Course	T.Y.B.Sc. (Information Technology)
O: _____	Eligibility	As per University Ordinance
R: _____	Passing Marks	40%
No. of years/Semesters:		Sem. V
Level:		U.G.
Pattern:		Semester
Status:		Revised
To be implemented from Academic Year :		From Academic Year: 2024-25

Sign of Chairperson
Dr. Mrs. R.
Srivaramangai
Ad-hoc BoS (IT)

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

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

Semester – 5			
Course Code	Course Type	Course Title	Credits
USIT501	Skill Enhancement Course	Software Project Development	2
USIT502	Skill Enhancement Course	Internet of Things: Theory and Practice	2
USIT503	Skill Enhancement Course	Advanced Web Development	2
USIT504	Discipline Specific Elective (Any One)	Artificial Intelligence and Applications	2
USIT505		Linux Server Administration	
USIT506	Discipline Specific Elective (Any One)	Advanced Java Technologies	2
USIT507		Emerging Technologies	
USIT5P1	Skill Enhancement Course Practical	Project Dissertation	2
USIT5P2	Skill Enhancement Course Practical	Internet of Things: Theory and Practice Practical	2
USIT5P3	Skill Enhancement Course Practical	Advanced Web Development Practical	2
USIT5P4	Discipline Specific Elective Practical (Any One)*	Artificial Intelligence and Applications Practical	2
USIT5P5		Linux Server Administration Practical	
USIT5P6	Discipline Specific Elective Practical (Any One)*	Advanced Java Technologies Practical	2
USIT5P7		Emerging Technologies Practical	
Total Credits			20

(All the practical mentioned in the syllabi are compulsory as per the courses chosen)

***The choice of Practical course is based on the theory Course. For Semester V, USIT504, USIT505, USIT506 and USIT507, the practical courses are USIT5P4, USIT5P5 USIT5P6, USIT5P7. For Semester VI, USIT604, USIT605 the practical courses are USIT6P4, USIT6P5 respectively. Practical Course USIT6P6 is compulsory.**

Semester-V

Software Project Development

B. Sc. (Information Technology)		Semester – V	
Course Name: Software Project Development		Course Code: USIT501	
Periods per week (1 Period is 50 minutes)		5	
Credits		2	
		Hours	Marks
Evaluation System	Theory Examination	2½	75
	Internal	--	25

Course Objective:

- Understanding on the need for Software Project Management.
- Understanding the basic steps of project evaluation, planning.
- Understanding the basic steps of activity planning, risk management.
- Understanding the basic steps of controlling cost, managing contracts, managing people.
- Understanding how to work in team, maintaining quality and successfully closing the project.

Unit	Details	Lectures
I	<p>Introduction to Software Project Management: Introduction, Why is Software Project Management Important? What is a Project? Software Projects versus Other Types of Project, Contract Management and Technical Project Management, Activities Covered by Software Project Management, Plans, Methods and Methodologies, Some Ways of Categorizing Software Projects, Project Charter, Stakeholders, Setting Objectives, The Business Case, Project Success and Failure, What is Management? Management Control, Project Management Life Cycle, Traditional versus Modern Project Management Practices.</p> <p>Project Evaluation and Programme Management: Introduction, Business Case, Project Portfolio Management, Evaluation of Individual Projects, Cost–benefit Evaluation Techniques, Risk Evaluation, Programme Management, Managing the Allocation of Resources within Programmes, Strategic Programme Management, Creating a Programme, Aids to Programme Management, Some Reservations about Programme Management, Benefits Management.</p> <p>An Overview of Project Planning: Introduction to Step Wise Project Planning, Step 0: Select Project, Step 1: Identify Project Scope and Objectives, Step 2: Identify Project Infrastructure, Step 3: Analyse Project Characteristics, Step 4: Identify Project Products and Activities, Step 5: Estimate Effort for Each Activity, Step 6: Identify Activity Risks, Step 7: Allocate Resources, Step 8: Review/Publicize Plan, Steps 9 and 10: Execute Plan/Lower Levels of Planning</p>	12

<p>II</p>	<p>Selection of an Appropriate Project Approach: Introduction, Build or Buy? Choosing Methodologies and Technologies, Software Processes and Process Models, Choice of Process Models, Structure versus Speed of Delivery, The Waterfall Model, The Spiral Model, Software Prototyping, Other Ways of Categorizing Prototypes, Incremental Delivery, Atern/Dynamic Systems Development Method, Rapid Application Development, Agile Methods, Extreme Programming (XP), Scrum, Lean Software Development, Managing Iterative Processes, Selecting the Most Appropriate Process Model.</p> <p>Software Effort Estimation: Introduction, Where are the Estimates Done? Problems with Over- and Under-Estimates, The Basis for Software Estimating, Software Effort Estimation Techniques, Bottom-up Estimating, The Top-down Approach and Parametric Models, Expert Judgement, Estimating by Analogy, Albrecht Function Point</p> <p>Analysis, Function Points Mark II, COSMIC Full Function Points, COCOMO II: A Parametric Productivity Model, Cost Estimation,</p> <p>Staffing Pattern, Effect of Schedule Compression, Capers Jones Estimating Rules of Thumb.</p>	<p>12</p>
<p>III</p>	<p>Activity Planning: Introduction, Objectives of Activity Planning, When to Plan, Project Schedules, Projects and Activities, Sequencing and Scheduling Activities, Network Planning Models, Formulating a Network Model, Adding the Time Dimension, The Forward Pass, Backward Pass, Identifying the Critical Path, Activity Float, Shortening the Project Duration, Identifying Critical Activities, Activity-on-Arrow Networks.</p> <p>Risk Management: Introduction, Risk, Categories of Risk, Risk Management Approaches, A Framework for Dealing with Risk, Risk Identification, Risk Assessment, Risk Planning, Risk Management, Evaluating Risks to the Schedule, Boehm’s Top 10 Risks and Counter Measures, Applying the PERT Technique, Monte Carlo Simulation, Critical Chain Concepts.</p> <p>Resource Allocation: Introduction, Nature of Resources, Identifying Resource Requirements, Scheduling Resources, Creating Critical Paths, Counting the Cost, Being Specific, Publishing the Resource Schedule, Cost Schedules, Scheduling Sequence.</p>	<p>12</p>
<p>IV</p>	<p>Monitoring and Control: Introduction, Creating the Framework, Collecting the Data, Review, Visualizing Progress, Cost Monitoring, Earned Value Analysis, Prioritizing Monitoring, Getting the Project Back to Target, Change Control, Software Configuration Management (SCM).</p> <p>Managing Contracts: Introduction, Types of Contract, Stages in Contract Placement, Typical Terms of a Contract, Contract Management, Acceptance.</p> <p>Managing People in Software Environments: Introduction, Understanding Behaviour, Organizational Behaviour: A Background, Selecting the Right Person for the Job, Instruction in the Best Methods, Motivation, The Oldham–Hackman Job Characteristics Model, Stress, Stress Management, Health and Safety, Some Ethical and Professional Concerns.</p>	<p>12</p>

V	<p>Working in Teams: Introduction, becoming a Team, Decision Making, Organization and Team Structures, Coordination Dependencies, Dispersed and Virtual Teams, Communication Genres, Communication Plans, Leadership.</p> <p>Software Quality: Introduction, The Place of Software Quality in Project Planning, Importance of Software Quality, Defining Software Quality, Software Quality Models, ISO 9126, Product and Process Metrics, Product versus Process Quality Management, Quality Management Systems, Process Capability Models, Techniques to Help Enhance Software Quality, Testing, Software Reliability, Quality Plans.</p> <p>Project Closeout: Introduction, Reasons for Project Closure, Project Closure Process, Performing a Financial Closure, Project Closeout Report.</p>	12
----------	--	-----------

Books and References:					
Sr. No.	Title	Author/s	Publisher	Edition	Year
1.	Software Project Management	Bob Hughes, Mike Cotterell, Rajib Mall	TMH	6th	2018
2.	Project Management and Tools & Technologies – An overview	Shailesh Mehta	SPD	1st	2017
3.	Software Project Management	Walker Royce	Pearson		2005

Course Outcome:

After completing the course, the learner will be able to:

CO1: Describe the basic concepts of software project management with its life cycle

CO2: Apply project estimation and evaluation techniques to real world problem

CO3: Apply Key project management system techniques like PERT, CRM

CO4: Identify project risk, monitor and track project deadlines

CO5: Work in teams to evaluate the different modes of communication among people.

Internet of Things: Theory and Practice

B. Sc. (Information Technology)		Semester – V	
Course Name: Internet of Things: Theory and Practice		Course Code: USIT502	
Periods per week (1 Period is 50 minutes)		5	
Credits		2	
		Hours	Marks
Evaluation System	Theory Examination	2½	75
	Internal	--	25

Course Objectives:

- To understand the concept of the Internet of Things and its impact on technology
- To learn about the design principles of connected devices for a seamless user experience
- To delve into the workings of the internet and its communication protocols
- To explore different techniques of prototyping embedded devices and their physical design
- To gain knowledge about various business models and funding options for Internet of Things startups.

Unit	Details	Lectures
I	<p>The Internet of Things: An Overview : The Flavour of the Internet of Things, The “Internet” of “Things”, The Technology of the Internet of Things, Enchanted Objects, Who is Making the Internet of Things?</p> <p>Design Principles for Connected Devices: Calm and Ambient Technology, Magic as Metaphor, Privacy, Keeping Secrets, Whose Data Is It Anyway? Web Thinking for Connected Devices, Small Pieces, Loosely Joined, First-Class Citizens On The Internet, Graceful Degradation, Affordances.</p> <p>Internet Principles: Internet Communications: An Overview, IP, TCP, The IP Protocol Suite (TCP/IP), UDP, IP Addresses, DNS, Static IP Address Assignment, Dynamic IP Address Assignment, IPv6, MAC Addresses, TCP and UDP Ports, An Example: HTTP Ports, Other Common Ports, Application Layer Protocols, HTTP, HTTPS: Encrypted HTTP, Other Application Layer Protocols.</p>	12
II	<p>Thinking About Prototyping: Sketching, Familiarity, Costs versus Ease of Prototyping, Prototypes and Production, Changing Embedded Platform, Physical Prototypes and Mass Personalisation, Climbing into the Cloud, Open Source versus Closed Source, Why Closed? Why Open? Mixing Open and Closed Source, Closed Source for Mass Market Projects, Tapping into the Community.</p> <p>Prototyping Embedded Devices: Electronics, Sensors, Actuators, Scaling Up the Electronics, Embedded Computing Basics, Microcontrollers, System-on-Chips, Choosing Your Platform, Arduino, Developing on the Arduino, Some Notes on the Hardware, Openness, Raspberry Pi, Cases and Extension Boards, Developing on the Raspberry Pi, Some Notes on the Hardware, Openness.</p>	12
III	<p>Prototyping the Physical Design: Preparation, Sketch, Iterate, and Explore, Nondigital Methods, Laser Cutting, Choosing a Laser Cutter, Software, Hinges and Joints, 3D Printing, Types of 3D Printing, Software, CNC Milling, Repurposing/Recycling.</p> <p>Prototyping Online Components: Getting Started with an API, Mashing Up APIs, Scraping, Legalities, Writing a New API, Clockodillo, Security,</p>	12

	Implementing the API, Using Curl to Test, Going Further, Real-Time Reactions, Polling, Comet, Other Protocols, MQ Telemetry Transport, Extensible Messaging and Presence Protocol, Constrained Application Protocol.	
IV	<p>Techniques for Writing Embedded Code: Memory Management, Types of Memory, Making the Most of Your RAM, Performance and Battery Life, Libraries, Debugging,</p> <p>Business Models: A Short History of Business Models, Space and Time, From Craft to Mass Production, The Long Tail of the Internet, Learning from History, The Business Model Canvas, Who Is the Business Model For? Models, Make Thing, Sell Thing, Subscriptions, Customisation, Be a Key Resource, Provide Infrastructure: Sensor Networks, Take a Percentage, Funding an Internet of Things Startup, Hobby Projects and Open Source, Venture Capital, Government Funding, Crowdfunding, Lean Startups.</p>	12
V	<p>Moving to Manufacture: What Are You Producing? Designing Kits, Designing Printed circuit boards, Software Choices, The Design Process, Manufacturing Printed Circuit Boards, Etching Boards, Milling Boards. Assembly, Testing, Mass-Producing the Case and Other Fixtures, Certification, Costs, Scaling Up Software, Deployment, Correctness and Maintainability, Security, Performance, User Community.</p> <p>Ethics: Characterizing the Internet of Things, Privacy, Control, Disrupting Control, Crowdsourcing, Environment, Physical Thing, Electronics, Internet Service, Solutions, The Internet of Things as Part of the Solution, Cautious Optimism, The Open Internet of Things Definition.</p>	12

Books and References:

Sr. No.	Title	Author/s	Publisher	Edition	Year
1.	Designing the Internet of Things	Adrian McEwen, Hakim Cassimally	WILEY	First	2014
2.	Internet of Things – Architecture and Design	Raj Kamal	McGraw Hill	First	2017
3.	Getting Started with the Internet of Things	Cuno Pfister	O'Reilly	Sixth	2018
4.	Getting Started with Raspberry Pi	Matt Richardson and Shawn Wallace	SPD	Third	2016

Course Outcomes:

Upon completion of the course, students will be able to:

- CO1. Explain the concept of the Internet of Things and its applications in real-life scenarios.
- CO2. Apply design principles to create user-friendly and connected devices.
- CO3. Understand the communication protocols of the internet and design efficient web-connected devices.
- CO4. Prototype embedded devices and create their physical design using various tools and techniques.
- CO5. Understand the business models and funding options available for startups in the Internet of Things domain and apply them to real-life scenarios.

Advanced Web Development

B. Sc. (Information Technology)		Semester – V	
Course Name: Advanced Web Development		Course Code: USIT503	
Periods per week (1 Period is 50 minutes)		5	
Credits		2	
		Hours	Marks
Evaluation System	Theory Examination	2½	75
	Internal	--	25

Course Objective:

- Understand .NET Fundamentals: Gain a comprehensive understanding of the .NET framework, including its languages (C#), the Common Language Runtime, and the .NET Class Library.
- Learn the fundamentals of the C# language, covering variables, data types, conditional logic, loops, methods, and advanced topics like object-based manipulation.
- Develop skills in writing code for ASP.NET applications, utilizing server controls, handling events, and configuring applications. Understand various form controls, validation techniques, and navigation concepts. Handling error handling, debugging, and exception handling techniques in ASP.NET.
- Learn about state management on both client and server sides, including view state, transferring information between pages, and working with cookies.
- Gain expertise in ADO.NET components, data providers, SQL commands, and queries.
- Understand the data provider model, direct data access, and disconnected data access.
- Learn to work with data source controls and integrate with Microsoft Azure Database.

Unit	Details	Lectures
I	.NET Technology and Framework, C#, VB, and the .NET Languages, The Common Language Runtime, The .NET Class Library. The C# Language: C# Language Basics, Variables and Data Types, Variable Operations, Object-Based Manipulation, Conditional Logic, Loops, Methods. Types, Objects, and Namespaces: The Basics About Classes, Building a Basic Class, Value Types and Reference Types, Understanding Namespaces and Assemblies, Advanced Class Programming.	12
II	ASP.NET Web Forms, Writing Code, Using the Code-Behind Class, Adding Event Handlers, Understanding the Anatomy of an ASP.NET Application, Introducing Server Controls, Using the Page Class, Using Application Events, Configuring an ASP.NET Application. Form Controls: Stepping Up to Web Controls, Web Control Classes, List Controls, Table Controls, Web Control Events and AutoPostBack, Validation, Understanding Validation, Using the Validation Controls, Rich Controls, The Calendar, The AdRotator, Pages with Multiple Views, User Controls and Graphics, User Controls, Dynamic Graphics, The Chart Control, Website Navigation: Site Maps, URL Mapping and Routing, The SiteMapPath Control, The TreeView Control, The Menu Control.	12
III	Tracing, Error Handling, Debugging, Exception Handling, User defined Exceptions, Page Tracing, Client and Server side state management, View State, Transferring Information Between Pages, Cookies- creation, retrieval, reading, Types of Cookies, Using Master Pages and Content Pages	12

IV	ADO.NET Components, Data Providers, SQL Basic command and queries, Understanding the Data Provider Model, Using Direct Data Access, Using Disconnected Data Access. Working with Data Source Controls, Data Controls: GridView, DetailsView and FormView.	12
V	Understanding Security Requirements, Introduction to Authentication and Authorization NuGet package manager for .NET and Visual Studio- Installation, Basics, framework, Bootstrap basics, Structure of the Page, Typography, Forms Ajax- Using Partial Refreshes, Using Progress Notification, Implementing Timed Refreshes, Working with the ASP.NET AJAX Control Toolkit.	12

Books and References:					
Sr. No.	Title	Author/s	Publisher	Edition	Year
1.	Beginning Visual C# 2010	K. Watson, C. Nagel, J.H Padderson	Wrox (Wiley)		2010
2.	Murach's ASP.NET 4.6 Web Programming in C#2015	Mary Delamater and Anne Bohem	SPD	Sixth	2016
3.	ASP.NET 4.0 programming	J. Kanjilal	Tata McGraw Hill		2011
4.	Programming ML.NET	Dino Esposito, Francesco Esposito	Paperback		
5.	Ajax : A Beginner's Guide	Steven Holzner	Paperback		2017
6	Introducing Bootstrap 4	Jörg Krause	Apress		2016

Web reference:

NuGet package :- <https://learn.microsoft.com/en-us/nuget/quickstart/install-and-use-a-package-in-visual-studio>

Course Outcome:

After completing the course, the learner will be able to:

CO1: Learner will explore the foundations of .NET Development, .NET Ecosystem, C# and Fundamental Concepts

CO2: Comprehensive Understanding and Practical Application using controls.

CO3: ASP.NET Web Development Essentials: Tracing, Debugging, and State Management Techniques

CO4: Mastering Data Access in ASP.NET: ADO.NET Essentials and Azure Integration

CO5: Securing ASP.NET Applications: Authentication, Authorization, and Advanced Techniques with Ajax and Bootstrap

Artificial Intelligence and Applications

B. Sc. (Information Technology)		Semester – V	
Course Name: Artificial Intelligence and Applications		Course Code: USIT504 (Elective 1)	
Periods per week (1 Period is 50 minutes)		5	
Credits		2	
		Hours	Marks
Evaluation System	Theory Examination	2½	75
	Internal	--	25

Pre-requisites:

1. Proficiency in at least one programming language such as C++, Python or Java.
2. Knowledge of fundamental data structures (arrays, linked lists, stacks, queues, trees, graphs) and their algorithms.
3. Understanding of discrete mathematics, linear algebra, probability and statistics.
4. Familiarity with logical reasoning.

Course Objectives:

- To understand the fundamental concepts and history of Artificial Intelligence, including the current state of the field and its foundational principles.
- To develop proficiency in problem-solving techniques using various search algorithms, both uninformed and informed, and apply heuristic functions effectively.
- To gain knowledge and skills in adversarial search, including strategies for games, optimal decision-making, and dealing with uncertainty in partially observable environments.
- To master logical reasoning and inference methods, including propositional and first-order logic, and apply them to knowledge representation and problem-solving tasks.
- To explore advanced topics such as planning algorithms, including classical planning and multi-agent planning, as well as generative AI techniques and their applications.

Unit	Details	Lectures
I	Introduction: What is Artificial Intelligence? Foundations of AI, History, the state of art AI today. Intelligent Agents: Agents and Environment, Good Behaviour, Nature of Environment, Structure of Agents.	12
II	Solving Problems by Searching: Problem Solving Agents, Searching for Solutions, Uninformed Search, Informed Search Strategies, Heuristic Functions. Beyond Classical Search: Local Search Algorithms, Searching with Non-Deterministic Action, Searching with Partial Observations, Online Search Agents and Unknown Environments.	12
III	Adversarial Search: Games, Optimal Decisions in Games, Alpha-Beta Pruning, Stochastic Games, Partially Observable Games. Logical Agents: Knowledge Base Agents, The Wumpus World, Propositional Logic, Propositional Theorem Proving. Probabilistic Reasoning: Uncertainty, Conditional Probability, Bayes Theorem.	12
IV	First Order Logic: Need For First Order Logic, Difference between Propositional and First Order Logic, Knowledge Engineering in First Order Logic. Inference in First Order Logic: Unification and Lifting, Forward and Backward Chaining, Resolution. Artificial Neural Network: Architecture of ANN, Merits and Demerits of ANN, Types of ANN.	12
V	Planning: Definition of Classical Planning, Algorithms for Planning as State	12

Space Search, Planning Graphs, Other Classical Planning Approaches, Analysis of Planning Approaches, Time, Schedules and Resources, Hierarchical Planning, Planning and Acting in Nondeterministic Domains, Multiagent Planning.
Generative AI: What is Generative AI? Types of Generative AI

Books and References:

Sr. No.	Title	Author/s	Publisher	Edition	Year
1.	Artificial Intelligence: A Modern Approach	Stuart Russel and Peter Norvig	Pearson	Third	2015
2.	A First Course in Artificial Intelligence	Deepak Khemani	TMH	First	2017
3.	Artificial Intelligence: A Rational Approach	Rahul Deva	Shroff Publisher	First	2018
4.	Artificial Intelligence	Elaine Rich, Kevin Knight and Shivashankar Nair	TMH	Third	2009
5.	Artificial Intelligence & Soft Computing for Beginners	Anandita Das Bhattacharjee	SPD	First	2013
6.	Artificial Intelligence & Generative AI for Beginners: The Complete Guide	David M. Patel	GD Publishing	First	2023

Course Outcomes:

After completing the course, the learner will be able to:

- CO1:** Articulate the historical development and current trends in Artificial Intelligence, demonstrating a comprehensive understanding of its foundations and principles.
- CO2:** Demonstrate proficiency in implementing and analyzing various search algorithms, utilizing both uninformed and informed strategies to solve complex problems efficiently.
- CO3:** Apply adversarial search techniques to decision-making in competitive environments, including games, and effectively manage uncertainty and partial observability.
- CO4:** Demonstrate competency in logical reasoning and inference, utilizing propositional and first-order logic to represent and solve real-world problems in AI applications.
- CO5:** Gain practical experience in planning algorithms and generative AI techniques, enabling them to design and implement AI systems capable of planning actions and generating novel content autonomously.

Linux Server Administration

B. Sc. (Information Technology)		Semester – V	
Course Name: Linux Server Administration		Course Code: USIT505 (Elective-I)	
Periods per week (1 Period is 50 minutes)		5	
Credits		2	
		Hours	Marks
Evaluation System	Theory Examination	2½	75
	Internal	--	25

Course Objective:

- To gain a deep understanding of various Linux distributions.
- To be able to install, configure, and optimize for specific server roles.
- To develop expertise in securing Linux servers from unauthorized access, attacks, and vulnerabilities.
- To acquire the skills to monitor server performance, identify bottlenecks, and optimize resource utilization.
- To understand Bash scripting language to automate repetitive tasks and server management processes.

Unit	Details	Lectures
I	<p>Getting Started Starting with Linux , Duties of the System Administrator, Installing Red Hat Enterprise Linux Server</p> <p>Working with Users, Groups, and Permissions: Managing Users and Groups, Commands for User Management, Managing Passwords, Modifying and Deleting User Accounts, Configuration Files, Creating Groups, Using Graphical Tools for User, and Group Management, Managing Permissions, the Role of Ownership, Basic Permissions: Read, Write, and Execute, Advanced Permissions, Working with Access Control Lists, Setting Default Permissions with umask, Working with Attributes.</p> <p>Gaining Privileges, Console Access, Installing and Managing Software YUM: Checking for and updating Packages, Packages and Package Groups, Configuring Yum and Yum Repositories, Yum Plug-ins, Additional resources Packagekit: Updating packages with software update, Using Add/Remove Software, Packagekit Architecture, Additional Resources</p>	12
II	<p>Configuring and Managing Storage: Understanding Partitions and Logical Volumes, Creating Partitions, Creating File Systems, File Systems Overview, Creating File Systems, Changing File System Properties, Checking the File System Integrity, Mounting File Systems Automatically Through fstab, Working with Logical Volumes, Creating Logical Volumes, Resizing Logical Volumes, Working with Snapshots, Replacing Failing Storage Devices, Creating Swap Space, Working with Encrypted Volumes</p> <p>Networking, Network Manager: The Network Manager Daemon, Interacting with Network Manager, Establishing Connections</p> <p>Network Interfaces: Network Configuration Files, Interface Configuration files, Interface Control Scripts, Static routes and the default gateway, Configuring static routes in ifcfg files, Network function files, ETHTool.</p>	12

	<p>Infrastructure Services: Services and Daemons- Configuring the default runlevel, Configuring the Services, Running Services, Additional Resources.</p> <p>Configuring Authentication- Configuring System Authentication, Using and Caching credentials with SSSD.</p> <p>OpenSSH- The SSH Protocol, Configuring OpenSSH, Using OpenSSH certificate authentication, OpenSSH clients, More than a Secure Shell, Additional Resources</p>	
III	<p>TigerVNC- VNC servers, sharing an existing Desktop, Using a VNC Viewer, Additional Resources</p> <p>Servers:</p> <p>DHCP Servers: Why Use DHCP? Configuring DHCPv4 server, Configuring DHCPv4 Client.</p> <p>DNS Servers – Introduction to DNS, BIND</p> <p>Web Servers – The Apache HTTP server</p> <p>Mail Servers – Email Protocols, Email Program Classification, Mail Transport Agents, Mail Delivery Agents, Mail User Agents, Additional Resources</p> <p>File Servers – Samba, FTP</p>	12
IV	<p>Monitoring and Automation</p> <p>System Monitoring Tools – Viewing System Processes, Viewing Memory Usage, Viewing CPU Usage, Viewing Block Devices and Filesystems, Viewing Hardware Information, Monitoring Performance with NET-SNMP, Additional Resources</p> <p>Viewing and Managing Log Files – Installing rsyslog, Locating Log files, Basic Configuration of rsyslog, working with queues in rsyslog, configuring rsyslog on a logging server, using rsyslog modules, debugging rsyslog, Managing Log files in graphical environment</p> <p>Automating System Tasks – Cron and Anacron, At and Batch</p>	12
V	<p>Securing Server with iptables: Understanding Firewalls, Setting Up a Firewall with system-config-firewall, Allowing Services, Trusted Interfaces, Masquerading, Configuration Files, Setting Up a Firewall with iptables, Tables, Chains, and Rules, Composition of Rule, Configuration Example, Advanced iptables Configuration, Configuring Logging, The Limit Module, Configuring NAT</p> <p>Setting Up Cryptographic Services: Introducing SSL, Proof of Authenticity: The Certificate Authority, Managing Certificates with OpenSSL, creating a Signing Request, Working with GNU Privacy Guard, Creating GPG Keys, Key Transfer, Managing GPG Keys, Encrypting Files with GPG, GPG Signing, Signing RPM Files</p> <p>Introducing Bash Shell Scripting</p> <p>Introduction, Elements of a Good Shell Script, Executing the Script, Working with Variables and Input, Understanding Variables, Variables, Subshells, and Sourcing, Working with Script Arguments, Asking for Input, Using Command Substitution, Substitution Operators, Changing Variable Content with Pattern Matching, Performing Calculations, Using Control Structures, Using if...then...else, Using case, Using while, Using until, Using for, Configuring booting with GRUB</p>	12

Books and References:						
Sr. No.	Title	Author/s	Publisher	Edition	Year	
1	Red Hat Enterprise Linux 6 Deployment Guide	Red Hat	Red Hat Content Services	First	2021	
2.	Linux Bible	Christopher Negus	Wiley	Tenth	2020	
3.	Red hat Linux Networking and System Administration	Terry Collings and Kurt Wall	Wiley	Third	2005	
4.	Red Hat Enterprise Linux6 Administration	Sander van Vugt	Sybex	First	2013	
5.	Linux Administration: A Beginner's Guide	Wale Soyinka	TMH	Eighth	2020	

Course Outcome:

After completing the course, the learner will be able to:

CO1: Demonstrate a comprehensive understanding of various Linux distributions, packages and proficiency in developing effective access control lists to manage users and group accounts.

CO2: Acquire the ability to manage storage and partitions, and develop efficiency in configuring network and infrastructure services.

CO3: Implement different servers with their server configuration roles and parameters.

CO4: Analyze system logs to identify potential problems and security incidents.

CO5: Develop expertise in implementing security measures like firewalls, access controls, password policies, and encryption to protect the system and ability to write scripts to automate tasks

Advanced Java Technologies

B. Sc. (Information Technology)		Semester – V	
Course Name: Advanced Java Technologies		Course Code: USIT506 (Elective-II)	
Periods per week (1 Period is 50 minutes)		5	
Credits		2	
		Hours	Marks
Evaluation System	Theory Examination	2½	75
	Internal	--	25

Course Objective:

- To understand the concept of and develop applications using servlets and database connectivity.
- To develop applications that can handle cookies, sessions and file operations.
- To understand the concept of and design applications using Java server pages.
- To understand the concept of and design applications using Enterprise Java Beans.
- To understand the concepts of persistence, Hibernate and develop JPA applications, Hibernate applications.

Unit	Details	Lectures
I	<p>Understanding Java EE: What is an Enterprise Application? What is java enterprise edition? Java EE Technologies, Java EE evolution, Glassfish server</p> <p>Java EE Architecture, Server and Containers: Types of System Architecture, Java EE Server, Java EE Containers.</p> <p>Introduction to Java Servlets: The Need for Dynamic Content, Java Servlet Technology, Why Servlets? What can Servlets do?</p> <p>Servlet API and Lifecycle: Java Servlet API, The Servlet Skeleton, The Servlet Life Cycle, A Simple Welcome Servlet</p> <p>Working with Servlets: Getting Started, Using Annotations Instead of Deployment Descriptor.</p> <p>Working with Databases: What Is JDBC? JDBC Architecture, Accessing Database, The Servlet GUI and Database Example.</p>	12
II	<p>Request Dispatcher: Requestdispatcher Interface, Methods of Requestdispatcher, Requestdispatcher Application.</p> <p>COOKIES: Kinds of Cookies, Where Cookies Are Used? Creating Cookies, Using Servlet, Dynamically Changing the Colors of A Page</p> <p>SESSION: What Are Sessions? Lifecycle of HttpSession, Session Tracking With Servlet API, A Servlet Session Example</p> <p>Working with Files: Uploading Files, Creating an Upload File Application, Downloading Files, Creating a Download File Application.</p> <p>Working with Non-Blocking I/O: Creating a Non- Blocking Read Application, Creating The Web Application, Creating Java Class, Creating Servlets, Retrieving The File, Creating index.jsp</p>	12
III	<p>Introduction To Java Server Pages: Why use Java Server Pages? Disadvantages Of JSP, JSP v\vs Servlets, Life Cycle of a JSP Page, How does a</p>	12

	<p>JSP function? How does JSP execute? About Java Server Pages Getting Started With Java Server Pages: Comments, JSPDocument, JSPElements, JSP GUI Example.</p> <p>Action Elements: Including other Files, Forwarding JSP Page to Another Page, Passing Parameters for other Actions, Loading a Java bean.</p> <p>Implicit Objects, Scope and El Expressions: Implicit Objects, Character Quoting Conventions, Unified Expression Language [UnifiedEl], Expression Language.</p> <p>Java Server Pages Standard Tag Libraries: What is wrong in using JSP Scriptlet Tags? How JSTL Fixes JSP Scriptlet's Short comings? Disadvantages Of JSTL, Tag Libraries.</p>	
IV	<p>Introduction To Enterprise Javabeans: Enterprise Bean Architecture, Benefits of Enterprise Bean, Types of Enterprise Bean, Accessing Enterprise Beans, Enterprise Bean Application, Packaging Enterprise Beans</p> <p>Working with Session Beans: When to use Session Beans? Types of Session Beans, Remote and Local Interfaces, Accessing Interfaces, Lifecycle of Enterprise Beans, Packaging Enterprise Beans, Example of Stateful Session Bean, Example of Stateless Session Bean, Example of Singleton Session Beans.</p> <p>Working with Message Driven Beans: Lifecycle of a Message Driven Bean, Uses of Message Driven Beans, The Message Driven Beans Example.</p> <p>Interceptors: Request and Interceptor, Defining An Interceptor, AroundInvoke Method, Applying Interceptor, Adding An Interceptor To An Enterprise Bean, Build and Run the Web Application.</p> <p>Java Naming and Directory Interface: What is Naming Service? What is Directory Service? What is Java Naming and Directory interface? Basic Lookup, JNDI Namespace in Java EE, Resources and JNDI, Datasource Resource Definition in Java EE.</p>	12
V	<p>Persistence, Object/Relational Mapping And JPA: What is Persistence? Persistence in Java, Current Persistence Standards in Java, Why another Persistence Standards? Object/Relational Mapping.</p> <p>Introduction to Java Persistence API: The Java Persistence API, JPA, ORM, Database and the Application, Architecture of JPA, How JPA Works? JPA Specifications.</p> <p>Writing JPA Application: Application Requirement Specifications, Software Requirements, The Application Development Approach, Creating Database and Tables in Mysql, creating a Web Application, Adding the Required Library Files, creating a Javabean Class, Creating Persistence Unit [Persistence.Xml], Creating JSP's, The JPA Application Structure, Running the JPA Application.</p> <p>Introduction to Hibernate: What is Hibernate? Why Hibernate? Hibernate, Database and The Application, Components of Hibernate, Architecture of Hibernate, How Hibernate Works?</p> <p>Writing Hibernate Application: Application Requirement Specifications, Software Requirements, The Application Development Approach, Creating Database and Tables in Mysql, Creating a Web Application, Adding the Required Library Files, Creating a Java bean Class, Creating Hibernate Configuration File, Adding a Mapping Class, Creating JSP's, Running The Hibernate Application.</p>	12

Books and References:					
Sr. No.	Title	Author/s	Publisher	Edition	Year
1.	Java EE 7 For Beginners	Sharanam Shah, Vaishali Shah	SPD	First	2017
2.	Java EE 8 Cookbook	Elder Moraes	Packt	First	2018
3.	Advanced Java Programming	Uttam Kumar Roy	Oxford Press	First	2015
4.	Java EE 8 Application Development	David R. Heffelfinger	Packt	First	2017
5.	Java EE 7 Essentials	Arun Gupta	O'Reilly	First	2013

Course Outcome:

After completing the course, the learner will be able to:

CO1: Proficiently understand and apply servlets and database connectivity concepts to develop dynamic web applications.

CO2: Demonstrate the ability to develop applications capable of managing cookies, sessions, and performing file operations effectively.

CO3: Proficient in understanding and designing applications using Java Server Pages (JSP), enabling dynamic and interactive web content creation.

CO4: Adept at comprehending and designing applications utilizing Enterprise Java Beans (EJB), facilitating the development of scalable and distributed enterprise-level applications.

CO5: Possess a thorough understanding of persistence concepts, Hibernate framework, and the ability to develop Java Persistence API (JPA) and Hibernate applications proficiently.

Emerging Technologies

B. Sc. (Information Technology)		Semester – V	
Course Name: Emerging Technologies		Course Code: USIT507 (Elective-II)	
Periods per week (1 Period is 50 minutes)		5	
Credits		2	
		Hours	Marks
Evaluation System	Theory Examination	2½	75
	Internal	--	25

Course Objective:

- To gain a comprehensive understanding of Big Data, develop insights into NoSQL databases, their history, advantages, and differences from SQL databases.
- To have an in-depth understanding of MongoDB's data modeling techniques, and to be proficient in using the MongoDB shell.
- To have a comprehensive understanding of MongoDB's storage engine and indexing mechanisms, be able to identify and address limitations in MongoDB deployments.
- To gain a comprehensive understanding of the evolution and advantages of SSD, exploring technologies such as TimesTen, Redis etc, while also mastering jQuery for efficient DOM traversal and manipulation.
- To understand and apply JSON grammar, syntax, and data types; create, parse, and persist JSON objects and arrays; compare JSON with XML for data interchange.

Unit	Details	Lectures
I	<p>Big Data: Getting Started, Big Data, Facts About Big Data, Big Data Sources, Three Vs of Big Data, Volume, Variety, Velocity, Usage of Big Data, Visibility, Discover and Analyze Information, Segmentation and Customizations, Aiding Decision Making, Innovation, Big Data Challenges, Policies and Procedures, Access to Data, Technology and Techniques, Legacy Systems and Big Data, Structure of Big Data, Data Storage, Data Processing, Big Data Technologies</p> <p>NoSQL: SQL, NoSQL, Definition, A Brief History of NoSQL, ACID vs. BASE, CAP Theorem (Brewer's Theorem), The BASE, NoSQL Advantages and Disadvantages, Advantages of NoSQL, Disadvantages of NoSQL, SQL vs. NoSQL Databases, Categories of NoSQL Databases</p> <p>Introducing MongoDB: History, MongoDB Design Philosophy, Speed, Scalability, and Agility, Non-Relational Approach, JSON-Based Document Store, Performance vs. Features, Running the Database Anywhere, SQL Comparison</p>	12
II	<p>The MongoDB Data Model: The Data Model, JSON and BSON, The Identifier (_id), Capped Collection, Polymorphic Schemas, Object- Oriented Programming, Schema Evolution</p> <p>Using MongoDB Shell: Basic Querying, Create and Insert, Explicitly Creating Collections, Inserting Documents Using Loop, Inserting by Explicitly Specifying _id, Update, Delete, Read, Using Indexes, Stepping Beyond the Basics, Using Conditional Operators, Regular Expressions, MapReduce, aggregate(), Designing an Application's Data Model, Relational Data Modeling and Normalization, MongoDB Document Data Model Approach</p>	12

	<p>MongoDB Architecture: Core Processes, mongod, mongo, mongos, MongoDB Tools, Standalone Deployment, Replication, Master/Slave Replication, Replica Set, Implementing Advanced Clustering with Replica Sets, Sharding, Sharding Components, Data Distribution Process, Data Balancing Process, Operations, Implementing Sharding, Controlling Collection Distribution (Tag-Based Sharding), Points to Remember When Importing Data in a Sharded Environment, Monitoring for Sharding, Monitoring the Config Servers, Production Cluster Architecture, Scenario 1, Scenario 2, Scenario 3, Scenario 4</p>	
III	<p>MongoDB Storage Engine: Data Storage Engine, Data File (Relevant for MMAPv1), Namespace (.ns File), Data File (Relevant for WiredTiger), Reads and Writes, How Data Is Written Using Journaling, GridFS – The MongoDB File System, The Rationale of GridFS, GridFS under the Hood, Using GridFS, Indexing, Types of Indexes, Behaviors and Limitations</p> <p>MongoDB Use Cases: Use Case 1 -Performance Monitoring, Schema Design, Operations, Sharding, Managing the Data, Use Case 2 – Social Networking, Schema Design, Operations, Sharding</p> <p>MongoDB Limitations: MongoDB Space Is Too Large (Applicable for MMAPv1), Memory Issues (Applicable for Storage Engine MMAPv1), 32-bit vs. 64-bit, BSON Documents, Namespaces Limits, Indexes Limit, Capped Collections Limit - Maximum Number of Documents in a Capped Collection, Sharding Limitations, Shard Early to Avoid Any Issues, Shard Key Can't Be Updated, Shard Collection Limit, Select the Correct Shard Key, Security Limitations, No Authentication by Default, Traffic to and from MongoDB Isn't Encrypted, Write and Read Limitations, Case-Sensitive Queries, Type-Sensitive Fields, No JOIN, Transactions, MongoDB Not Applicable Range</p> <p>MongoDB Best Practices: Deployment, Hardware Suggestions from the MongoDB Site, Few Points to be Noted, Coding, Application Response Time Optimization, Data Safety, Administration, Replication Lag, Sharding, Monitoring</p>	12
IV	<p>The End of Disk? SSD and In-Memory Databases: The End of Disk?, Solid State Disk, The Economics of Disk, SSD-Enabled Databases, In-Memory Databases, TimesTen, Redis, SAP HANA, VoltDB, Oracle 12c “in-Memory Database, Berkeley Analytics Data Stack and Spark, Spark Architecture</p> <p>jQuery: Introduction, Traversing the DOM, DOM Manipulation with jQuery, Events, Ajax with jQuery, jQuery Plug-ins, jQuery Image Slider</p>	12
V	<p>JSON: Introduction, JSON Grammar, JSON Values, JSON Tokens, Syntax, JSON vs XML, Data Types, Objects, Arrays, Creating JSON, JSON Object, Parsing JSON, Persisting JSON, Data Interchange, JSON PHP, JSON HTML, JSONP</p>	12

Books and References:

Sr. No.	Title	Author/s	Publisher	Edition	Year
1.	Practical MongoDB	Shakuntala Gupta dward NavinSabharwal	Apress	First	2015
2.	Beginning jQuery	Jack Franklin Russ Ferguson	Apress	Second	2017

3.	Next Generation Databases	Guy Harrison	Apress	First	2015
4.	Beginning JSON	Ben Smith	Apress	First	2015
5.	Mastering MongoDB	Marko Aleksendric, Arek Borucki, Leandro Domingues	Packt	First	2015

Course Outcome:

After completing the course, the learner will be able to:

CO1: Understand and articulate the foundational concepts and practical applications of Big Data, and compare and contrast SQL and NoSQL databases.

CO2: Design and implement effective MongoDB data models utilizing JSON, BSON, and various schema strategies; efficiently perform data operations using the MongoDB shell.

CO3: Understand and utilize MongoDB's storage engine, journaling, and GridFS file system; effectively design and implement MongoDB use cases.

CO4: Evaluate the benefits and implications of transitioning from traditional disk storage to SSD and in-memory databases, proficiently use technologies like TimesTen, Redis, etc., and effectively utilize jQuery for DOM manipulation.

CO5: Understanding of JSON, including its grammar, syntax, and data types, and be able to create, parse, and persist JSON objects and arrays. They will also learn to compare JSON with XML for data interchange and effectively use JSON in various web development contexts.

Internet of Things: Theory and Practice Practical

B. Sc. (Information Technology)		Semester – V	
Course Name: Internet of Things: Theory and Practice Practical		Course Code: USIT5P2	
Periods per week (1 Period is 50 minutes)		3	
Credits		2	
		Hours	Marks
Evaluation System	Practical Examination	2	50
	Internal	--	--

Practical No	Details
0	Starting Raspbian OS, Familiarizing with Raspberry Pi Components and interface, Connecting to ethernet, Monitor, USB.
1	Displaying different LED patterns with Raspberry Pi.
2	Displaying Time over 4-Digit 7-Segment Display using Raspberry Pi
3	Interfacing 16X2 LCD with Raspberry Pi to display different messages.
4	Raspberry Pi Based Oscilloscope
5	Controlling Raspberry Pi with WhatsApp.
6	Fingerprint Sensor interfacing with Raspberry Pi
7	Raspberry Pi GPS Module Interfacing
8	IoT based Web Controlled Home Automation using Raspberry Pi
9	Interfacing Pi Camera with Raspberry Pi
10	Interfacing Raspberry Pi with RFID.
11	Installing Windows 10 IoT Core on Raspberry Pi (Demo Practical)

Books and References:

Sr. No.	Title	Author/s	Publisher	Edition	Year
1.	Designing the Internet of Things	Adrian McEwen, Hakim Cassimally	WILEY	First	2014
2.	Official Raspberry Pi Beginner's Guide	Gareth Halfacre	Raspberry Pi Press	First	2018
3.	Programming the Raspberry Pi	by Simon Monk	McGraw Hill	Third	2021
4.	Getting Started with Raspberry Pi	Matt Richardson and Shawn Wallace	SPD	Third	2016

Advanced Web Development Practical

B. Sc. (Information Technology)		Semester – V	
Course Name: Advanced Web Development Practical		Course Code: USIT5P3	
Periods per week (1 Period is 50 minutes)		3	
Credits		2	
		Hours	Marks
Evaluation System	Practical Examination	2½	50
	Internal	--	--

List of Practical

1.	Write the program for the following:
a.	Create an application to print on screen the output of adding, subtracting, multiplying and dividing two numbers entered by the user in C#.
b.	Create an application to print Floyd's triangle till n rows in C#.
c.	Create an application to demonstrate following operations i. Generate Fibonacci series. ii. Test for prime numbers.
2.	Write the program for the following:
a.	Create a simple application to demonstrate the concepts boxing and unboxing.
b.	Create a simple application to perform addition and subtraction using delegate.
c.	Create a simple application to demonstrate use of the concepts of interfaces.
3.	Write the program for the following:
a.	Create a simple web page with various server controls to demonstrate setting and use of their properties. (Example : AutoPostBack)
b.	Create a simple application to demonstrate your vacation using calendar control.
c.	Demonstrate the use of Treeview operations on the web form.
4.	Write the program for the following:
a.	Create a Registration form to demonstrate use of various Validation controls.
b.	Create Web Form to demonstrate use of Adrotator Control.
c.	Create Web Form to demonstrate use User Controls
5.	Write the program for the following:
a.	Create Web Form to demonstrate use of Website Navigation controls.
b.	Create a web application to demonstrate use of Master Page and content page.
c.	Create a web application to demonstrate various states of ASP.NET Pages.
6.	Write the program for the following:
a.	Create a web application for inserting and deleting records from a database.
b.	Create a web application to display Using Disconnected Data Access and Databinding using GridView.

7.	Write the program for the following:
a.	Create a web application to demonstrate the use of different types of Cookies.
b.	Create a web application to demonstrate Form Security and Windows Security with proper Authentication and Authorization properties.
8.	Write the program for the following:
a.	Create a web application for inserting and deleting records from a database. (Using ExecuteNon Query).
b.	Create a web application for user defined exception handling.
9.	Write the program for the following:
a.	Create a web application to demonstrate use of GridView button column and GridView events along with paging and sorting.
b.	Create a web application to demonstrate data binding using DetailsView and FormView Control.
10.	Write the program for the following:
a.	Create a web application to demonstrate JS Bootstrap Button.
b.	Create a web application to demonstrate use of various Ajax controls.
c.	Create a web application to demonstrate Installation and use of NuGet package.

Books and References:					
Sr. No.	Title	Author/s	Publisher	Edition	Year
1.	Beginning Visual C# 2010	K. Watson, C. Nagel, J.H Padderson, J.D. Reid, M.Skinner	Wrox (Wiley)		2010
2.	Murach's ASP.NET 4.6 Web Programming in C#2015	Mary Delamater and Anne Bohem	SPD	Sixth	2016
3.	ASP.NET 4.0 programming	J. Kanjilal	Tata McGrawHill		2011
4.	Programming ML.NET	Dino Esposito (Author), Francesco Esposito (Author)	Paperback		
5.	Ajax : A Beginner's Guide	Steven Holzner	Paperback		2017
6	Introducing Bootstrap 4	By Jörg Krause	Apress		2016

Artificial Intelligence and Applications Practical

B. Sc. (Information Technology)		Semester – V	
Course Name: Artificial Intelligence and Applications Practical		Course Code: USIT5P4 (Elective I)	
Periods per week (1 Period is 50 minutes)		3	
Credits		2	
		Hours	Marks
Evaluation System	Practical Examination	2½	50
	Internal	--	--

List of Practical	
1.	Write programs for the following:
a.	Implement depth first search algorithm.
b.	Implement breadth first search algorithm.
2.	Write programs for the following:
a.	Simulate 4-Queen / N-Queen problem.
b.	Solve tower of Hanoi problem.
3.	Write programs for the following:
a.	Implement alpha beta search.
b.	Implement hill climbing problem.
4.	Write programs for the following:
a.	Implement A* algorithm.
b.	Solve water jug problem.
5.	Write programs for the following:
a.	Simulate tic – tac – toe game using min-max algorithm.
b.	Shuffle deck of cards.
6.	Write program for the following:
a.	Design an application to simulate number puzzle problem.
7.	Write program for the following:
a.	Solve constraint satisfaction problem.
8.	Write programs for the following:
a.	Derive the expressions based on Associative Law.
b.	Derive the expressions based on Distributive Law.
9.	Write program for the following:
a.	Derive the predicate. (for e.g.: Sachin is batsman, batsman is cricketer) - > Sachin is Cricketer

10.	Write program for the following:
a.	Write a program which contains three predicates: male, female, parent. Make rules for following family relations: father, mother, grandfather, grandmother, brother, sister, uncle, aunt, nephew and niece, cousin. Question: i. Draw Family Tree. ii. Define: Clauses, Facts, Predicates and Rules with conjunction and disjunction

Books and References:					
Sr. No.	Title	Author/s	Publisher	Edition	Year
1.	Artificial Intelligence: A Modern Approach	Stuart Russel and Peter Norvig	Pearson	Third	2015
2.	A First Course in Artificial Intelligence	Deepak Khemani	TMH	First	2017
3.	Artificial Intelligence: A Rational Approach	Rahul Deva	Shroff Publisher	First	2018
4.	Artificial Intelligence	Elaine Rich, Kevin Knight and Shivashankar Nair	TMH	Third	2009
5.	Artificial Intelligence & Soft Computing for Beginners	Anandita Das Bhattacharjee	SPD	First	2013

Essentials of Linux Server Administration Practical

B. Sc. (Information Technology)		Semester – V	
Course Name: Essentials of Linux Server Administration Practical		Course Code: USIT5P5 (Elective-I)	
Periods per week (1 Period is 50 minutes)		3	
Credits		2	
		Hours	Mark s
Evaluation System	Practical Examination	2½	50
	Internal	--	--

List of Practical	
0.	Installation of RHEL6.x
1.	Working with Users, Groups, and Permissions
a.	Creating and Managing Users
b.	Creating and Managing Groups
c.	Using Permissions and Advanced Permissions
2.	Working with Console, RPM, and Yum
a.	Managing Console and Gaining Privileges
b.	Working with RPM
c.	Working with Yum
3.	Working with Storage, Network and Infrastructure services
a.	Configuring and Managing Storage
b.	Connecting to the Network
c.	Configuring and enabling Infrastructure services
d.	Configuring OpenSSH
4.	Configuring Server for File Sharing
a.	Configuring NFS Server and Client
b.	Configuring Samba
c.	Configuring FTP
5.	Configuring DHCP, DNS, Web and Mail Servers
a.	Configuring DHCP server
b.	Configuring DNS server
c.	Configuring Web server
d.	Configuring Mail server
6.	Configuring Monitoring and Automation tools
a.	Configuring System Monitoring tools
b.	Configuring rsyslog
c.	Configuring cron, anacron, At and Batch Utility

7.	Firewall and Cryptographic services
a.	Securing Server with iptables
b.	Setting Up Cryptographic Services
8.	Write the program for the following:
a.	Programs on Shell scripting
b.	Shell scripts to automate or monitor tasks/processes
9.	Perform Daily System Administration tasks
a.	Managing Processes
b.	Working with Storage Devices and Links
c.	Working with repository
10.	Configuring selinux
a.	Working with selinux

Books and References:					
Sr. No.	Title	Author/s	Publisher	Edition	Year
1	Red Hat Enterprise Linux6 Administration	Sander van Vugt	Sybex	First	2013
2	Red Hat Enterprise Linux 6 Deployment Guide	Red Hat	Red Hat Content Services	First	2021
3	Red hat Linux Networking and System Administration	Terry Collings and Kurt Wall	Wiley	Third	2005
4	Linux Administration: A Beginner's Guide	Wale Soyinka	TMH	Eighth	2020
5	RedHat certified System Administrator	William Maning	Emero Publishing	Second	2012

Advanced Java Technologies Practical

B. Sc. (Information Technology)		Semester – V	
Course Name: Advanced Java Technologies Practical		Course Code: USIT5P6 (Elective-II)	
Periods per week (1 Period is 50 minutes)		3	
Credits		2	
		Hours	Marks
Evaluation System	Practical Examination	2½	50
	Internal	--	--

List of Practical	
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 following JSP JSTL and EL Applications.
a.	Create an html page with fields, eno, name, age, desg, 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.
6.	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].
7.	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.
8.	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.
9.	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.
10.	Implement the following Hibernate applications.
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.

Books and References:					
Sr. No.	Title	Author/s	Publisher	Edition	Year
1.	Java EE 7 For Beginners	Sharanam Shah, Vaishali Shah	SPD	First	2017
2.	Java EE 8 Cookbook	Elder Moraes	Packt	First	2018
3.	Advanced Java Programming	Uttam Kumar Roy	Oxford Press	First	2015
4.	Java EE 8 Application Development	David R. Heffelfinger	Packt	First	2017
5.	Java EE 7 Essentials	Arun Gupta	O'Reilly	First	2013

Emerging Technologies Practical

B. Sc. (Information Technology)		Semester – V	
Course Name: Emerging Technologies Practical		Course Code: USIT5P7 (Elective-II)	
Periods per week (1 Period is 50 minutes)		3	
Credits		2	
		Hours	Marks
Evaluation System	Practical Examination	2½	50
	Internal	--	--

List of Practical	
1.	MongoDB Basics
a.	Write a MongoDB query to create and drop database.
b.	Write a MongoDB query to create, display and drop collection
c.	Write a MongoDB query to insert, query, update and delete a document.
2.	Simple Queries with MongoDB
3.	Implementing Aggregation
a.	Write a MongoDB query to use sum, avg, min and max expression.
b.	Write a MongoDB query to use push and addToSet expression.
c.	Write a MongoDB query to use first and last expression.
4.	Replication, Backup and Restore
a.	Write a MongoDB query to create Replica of existing database.
b.	Write a MongoDB query to create a backup of existing database.
c.	Write a MongoDB query to restore database from the backup.
5.	Java and MongoDB
a.	Connecting Java with MongoDB and inserting, retrieving, updating and deleting.
6.	PHP and MongoDB
a.	Connecting PHP with MongoDB and inserting, retrieving, updating and deleting.
7.	Python and MongoDB
a.	Connecting Python with MongoDB and inserting, retrieving, updating and deleting.

8.	Programs on Basic jQuery
a.	jQuery Basic, jQuery Events
b.	jQuery Selectors, jQuery Hide and Show effects
c.	jQuery fading effects, jQuery Sliding effects
9.	jQuery Advanced
a.	jQuery Animation effects, jQuery Chaining
b.	jQuery Callback, jQuery Get and Set Contents
c.	jQuery Insert Content, jQuery Remove Elements and Attribute
10.	JSON & Mongo DB
a.	Creating JSON
b.	Parsing JSON
c.	Persisting JSON
d.	Export MongoDB to JSON.
e.	Write a MongoDB query to delete JSON object from MongoDB.

Books and References:					
Sr. No.	Title	Author/s	Publisher	Edition	Year
1.	Practical MongoDB	Shakuntala Gupta Edward Navin Sabharwal	Apress	First	2015
2.	Beginning jQuery	Jack Franklin			
3.	Russ Ferguson	Apress	Second	2017	
4.	Next Generation Databases	Guy Harrison	Apress	First	2015
5.	Beginning JSON	Ben Smith	Apress	First	2015

Evaluation Scheme:

1. Internal Evaluation (25 Marks).

i. Test: 1 Class test of 20 marks. (Can be taken online)

Q	Attempt <i>any four</i> of the following:	20
a.		
b.		
c.		
d.		
e.		
f.		

ii. 5 marks: Active participation in the class, overall conduct, attendance.

2. External Examination: (75 marks)

	All questions are compulsory	
Q1	(Based on Unit 1) Attempt <i>any three</i> of the following:	15
a.		
b.		
c.		
d.		
e.		
f.		
Q2	(Based on Unit 2) Attempt <i>any three</i> of the following:	15
Q3	(Based on Unit 3) Attempt <i>any three</i> of the following:	15
Q4	(Based on Unit 4) Attempt <i>any three</i> of the following:	15
Q5	(Based on Unit 5) Attempt <i>any three</i> of the following:	15

3. Practical Exam: 50 marks

A Certified copy journal is essential to appear for the practical examination.

1.	Practical Question 1	20
2.	Practical Question 2	20
3.	Journal	5
4.	Viva Voce	5

OR

1.	Practical Question	40
2.	Journal	5
3.	Viva Voce	5

Project Dissertation Semester V and Project Implementation Semester VI

Chapter 1 to 4 should be submitted in Semester V in spiral binding. These chapter have also to be included in Semester VI report. Semester VI report has to be hard bound with golden embossing. Students will be evaluated based on the dissertation in semester V and dissertation and viva voce in Semester VI.

I. OBJECTIVES

- Describe the Systems Development Life Cycle (SDLC).
- Evaluate systems requirements.
- Complete a problem definition.
- Evaluate a problem definition.
- Determine how to collect information to determine requirements.
- Perform and evaluate feasibility studies like cost-benefit analysis, technical feasibility, time feasibility and Operational feasibility for the project.
- Work on data collection methods for fact finding.
- Construct and evaluate data flow diagrams.
- Construct and evaluate data dictionaries.
- Evaluate methods of process description to include structured English, decision tables and decision trees.
- Evaluate alternative tools for the analysis process.
- Create and evaluate such alternative graphical tools as systems flow charts and state transition diagrams.
- Decide the S/W requirement specifications and H/W requirement specifications.
- Plan the systems design phase of the SDLC.
- Distinguish between logical and physical design requirements.
- Design and evaluate system outputs.
- Design and evaluate systems inputs.
- Design and evaluate validity checks for input data.
- Design and evaluate user interfaces for input.
- Design and evaluate file structures to include the use of indexes.
- Estimate storage requirements.
- Explain the various file update processes based on the standard file organizations.
- Decide various data structures.
- Construct and evaluate entity-relationship (ER) diagrams for RDBMS related projects.
- Perform normalization for the unnormalized tables for RDBMS related projects
- Decide the various processing systems to include distributed, client/server, online and others.
- Perform project cost estimates using various techniques.
- Schedule projects using both GANTT and PERT charts.
- Perform coding for the project.
- Documentation requirements and prepare and evaluate systems documentation.
- Perform various systems testing techniques/strategies to include the phases of testing.
- Systems implementation and its key problems.

- Generate various reports.
- Be able to prepare and evaluate a final report.
- Brief the maintenance procedures and the role of configuration management in operations.
- To decide the future scope and further enhancement of the system.
- Plan for several appendices to be placed in support with the project report documentation.
- Decide the various processing systems to include distributed, client/server, online and others.
- Perform project cost estimates using various techniques.
- Schedule projects using both GANTT and PERT charts.
- Perform coding for the project.
- Documentation requirements and prepare and evaluate systems documentation.
- Perform various systems testing techniques/strategies to include the phases of testing.
- Systems implementation and its key problems.
- Generate various reports.
- Be able to prepare and evaluate a final report.
- Brief the maintenance procedures and the role of configuration management in operations.
- To decide the future scope and further enhancement of the system.
- Plan for several appendices to be placed in support with the project report documentation.
- Work effectively as an individual or as a team member to produce correct, efficient, well organized and documented programs in a reasonable time.
- Recognize problems that are amenable to computer solutions, and knowledge of the tool necessary for solving such problems.
- Develop of the ability to assess the implications of work performed.
- Get good exposure and command in one or more application areas and on the software
- Develop quality software using the software engineering principles
- Develop of the ability to communicate effectively.

II. Type of the Project

The majority of the students are expected to work on a real-life project preferably in some industry/ Research and Development Laboratories/Educational Institution/Software Company.

Students are encouraged to work in the areas listed below. However, it is *not mandatory* for a student to work on a real-life project. The student can formulate a project problem with the help of her/his Guide and submit the project proposal of the same. **Approval of the project proposal is mandatory.** If approved, the student can commence working on it, and complete it. Use the latest versions of the software packages for the development of the project.

III. SOFTWARE AND BROAD AREAS OF APPLICATION

FRONT END / GUI Tools	.Net Technologies,Java
DBMS/BACK END	Oracle, SQL Plus, MY SQL, SQL Server,
LANGUAGES	C, C++, Java, VC++, C#, R,Python
SCRIPTING LANGUAGES	PHP,JSP, SHELL Scripts (Unix), TeL/TK,
.NET Platform	F#,C#. Net, Visual C#. Net, ASP.Net
MIDDLE WARE (COMPONENT) TECHNOLOGIES	COM/DCOM, Active-X, EJB
UNIX INTERNALS	Device Drivers, RPC, Threads, Socket programming
NETWORK/WIRELESS TECHNOLOGIES	-
REALTIME OPERATING SYSTEM/ EMBEDDED SKILLS	LINUX, Raspberry Pi, Arduino, 8051
APPLICATION AREAS	Financial / Insurance / Manufacturing / Multimedia / Computer Graphics / Instructional Design/ Database Management System/ Internet / Intranet / Computer Networking-Communication Software development/ E-Commerce/ ERP/ MRP/ TCP-IP programming / Routing protocols programming/ Socket programming.

IV. Introduction

The project report should be documented with scientific approach to the solution of the problem that the students have sought to address. The project report should be prepared in order to solve the problem in a methodical and professional manner, making due references to appropriate techniques, technologies and professional standards. The student should start the documentation process from the first phase of software development so that one can easily identify the issues to be focused upon in the ultimate project report. The student should also include the details from the project diary, in which they will record the progress of their project throughout the course. The project report should contain enough details to enable examiners to evaluate the work. The important points should be highlighted in the body of the report, with details often referred to appendices.

1.1 PROJECT REPORT:

Title Page

Original Copy of the Approved Proforma of the Project Proposal

Certificate of Authenticated work

Role and Responsibility Form

Abstract

Acknowledgement

Table of Contents

Table of Figures

CHAPTER 1: INTRODUCTION

1.1 Background

1.2 Objectives

1.3 Purpose, Scope, and Applicability

1.3.1 Purpose

1.3.2 Scope

1.3.3 Applicability

1.4 Achievements

1.5 Organisation of Report

CHAPTER 2: SURVEY OF TECHNOLOGIES

CHAPTER 3: REQUIREMENTS AND ANALYSIS

3.1 Problem Definition

3.2 Requirements Specification

3.3 Planning and Scheduling

3.4 Software and Hardware Requirements

3.5 Preliminary Product Description

3.6 Conceptual Models

CHAPTER 4: SYSTEM DESIGN

4.1 Basic Modules

4.2 Data Design

4.2.1 Schema Design

4.2.2 Data Integrity and Constraints

4.3 Procedural Design

4.3.1 Logic Diagrams

4.3.2 Data Structures

4.3.3 Algorithms Design

4.4 User interface design

4.5 Security Issues

4.6 Test Cases Design

The documentation should use tools like star UML, Visuo for windows, Rational Rose for design as part of Software Project Management Practical Course. The documentation should be spiral bound for semester V and the entire documentation should be hard bound during semester VI.

CHAPTER 5: IMPLEMENTATION AND TESTING

- 5.1 Implementation Approaches
- 5.2 Coding Details and Code Efficiency
 - 5.2.1 Code Efficiency
- 5.3 Testing Approach
 - 5.3.1 Unit Testing
 - 5.3.2 Integrated Testing
 - 5.3.3 Beta Testing
- 5.4 Modifications and Improvements
- 5.5 Test Cases

CHAPTER 6: RESULTS AND DISCUSSION

- 6.1 Test Reports
- 6.2 User Documentation

CHAPTER 7: CONCLUSIONS

- 7.1 Conclusion
 - 7.1.1 Significance of the System
- 7.2 Limitations of the System
- 7.3 Future Scope of the Project

REFERENCES

GLOSSARY

APPENDIX A

APPENDIX B

V. EXPLANATION OF CONTENTS

Title Page

Sample format of Title page is given in Appendix 1 of this block. Students should follow the given format.

Original Copy of the Approved Proforma of the Project Proposal

Sample Proforma of Project Proposal is given in Appendix 2 of this block. Students should follow the given format.

Certificate of Authenticated work

Sample format of Certificate of Authenticated work is given in Appendix 3 of this block. Students should follow the given format.

Role and Responsibility Form

Sample format for Role and Responsibility Form is given in Appendix 4 of this block. Students should follow the given format.

Abstract

This should be one/two short paragraphs (100-150 words total), summarising the project work. It is important that this is not just a re-statement of the original project outline. A suggested flow is background, project aims and main achievements. From the abstract, a reader should be able to ascertain if the project is of interest to them and, it should present results of which they may wish to know more details.

Acknowledgements:

This should express student's gratitude to those who have helped in the preparation of project.

Table of Contents:

The table of contents gives the readers a view of the detailed structure of the report. The students would need to provide section and subsection headings with associated pages. The formatting details of these sections and subsections are given below. Table of Figures: List of all Figures, Tables, Graphs, Charts etc. along with their page numbers in a table of figures.

Chapter 1: Introduction

The introduction has several parts as given below:

Background: A description of the background and context of the project and its relation to work already done in the area. Summarise existing work in the area concerned with the project work.

Objectives: Concise statement of the aims and objectives of the project. Define exactly what is going to be done in the project; the objectives should be about 30 /40 words.

Purpose, Scope and Applicability: The description of Purpose, Scope, and Applicability are given below:

- **Purpose:** Description of the topic of the project that answers questions on why this project is being done. How the project could improve the system its significance and theoretical framework.
- **Scope:** A brief overview of the methodology, assumptions and limitations. The students should answer the question: What are the main issues being covered in the project? What are the main functions of the project?

- **Applicability:** The student should explain the direct and indirect applications of their work. Briefly discuss how this project will serve the computer world and people.
- **Achievements:** Explain what knowledge the student achieved after the completion of the work. What contributions has the project made to the chosen area? Goals achieved - describes the degree to which the findings support the original objectives laid out by the project. The goals may be partially or fully achieved, or exceeded.
- **Organisation of Report:** Summarizing the remaining chapters of the project report, in effect, giving the reader an overview of what is to come in the project report.

Chapter 2: Survey of Technologies

In this chapter Survey of Technologies should demonstrate the students awareness and understanding of Available Technologies related to the topic of the project. The student should give the detail of all the related technologies that are necessary to complete the project. The should describe the technologies available in the chosen area and present a comparative study of all those Available Technologies. Explain why the student selected the one technology for the completion of the objectives of the project.

Chapter 3: Requirements and Analysis

Problem Definition: Define the problem on which the students are working in the project.

Provide details of the overall problem and then divide the problem in to sub-problems. Define each sub-problem clearly.

Requirements Specification: In this phase the student should define the requirements of the system, independent of how these requirements will be accomplished. The Requirements Specification describes the things in the system and the actions that can be done on these things. Identify the operation and problems of the existing system.

Planning and Scheduling: Planning and scheduling is a complicated part of software development. Planning, for our purposes, can be thought of as determining all the small tasks that must be carried out in order to accomplish the goal. Planning also takes into account, rules, known as constraints, which, control when certain tasks can or cannot happen. Scheduling can be thought of as determining whether adequate resources are available to carry out the plan. The student should show the Gantt chart and Program Evaluation Review Technique (PERT).

Software and Hardware Requirements: Define the details of all the software and hardware needed for the development and implementation of the project.

- **Hardware Requirement:** In this section, the equipment, graphics card, numeric co-processor, mouse, disk capacity, RAM capacity etc. necessary to run the software must be noted.
- **Software Requirements:** In this section, the operating system, the compiler, testing tools, linker, and the libraries etc. necessary to compile, link and install the software must be listed.

Preliminary Product Description: Identify the requirements and objectives of the new system. Define the functions and operation of the application/system the students are developing as project.

Conceptual Models: The student should understand the problem domain and produce a model of the system, which describes operations that can be performed on the system, and the allowable sequences of those operations. Conceptual Models could consist of complete Data Flow Diagrams, ER diagrams, Object-oriented diagrams, System Flowcharts etc.

Chapter 4: System Design

Describes desired features and operations in detail, including screen layouts, business rules, process diagrams, pseudocode and other documentation.

Basic Modules: The students should follow the divide and conquer theory, so divide the overall problem into more manageable parts and develop each part or module separately. When all modules are ready, the student should integrate all the modules into one system. In this phase, the student should briefly describe all the modules and the functionality of these modules.

Data Design: Data design will consist of how data is organised, managed and manipulated.

- **Schema Design:** Define the structure and explanation of schemas used in the project.
- **Data Integrity and Constraints:** Define and explain all the validity checks and constraints provided to maintain data integrity.

Procedural Design: Procedural design is a systematic way for developing algorithms or procedurals.

- **Logic Diagrams:** Define the systematical flow of procedure that improves its comprehension and helps the programmer during implementation. e.g., Control Flow Chart, Process Diagrams etc.
- **Data Structures:** Create and define the data structure used in procedures.
- **Algorithms Design:** With proper explanations of input data, output data, logic of processes, design and explain the working of algorithms.

User Interface Design: Define user, task, environment analysis and how to map those requirements in order to develop a “User Interface”. Describe the external and internal components and the architecture of user interface. Show some rough pictorial views of the user interface and its components.

Security Issues: Discuss Real-time considerations and Security issues related to the project and explain how the student intends avoiding those security problems. What are the security policy plans and architecture?

Test Cases Design: Define test cases, which will provide easy detection of errors and mistakes within a minimum period of time and with the least effort. Explain the different conditions in which the students wish to ensure the correct working of the project.

Chapter 5: Implementation and Testing

Implementation Approaches: Define the plan of implementation, and the standards the students have used in the implementation. **Coding Details and Code Efficiency:** Students not need include full source code, instead, include only the important codes (algorithms, applets code, forms code etc). The program code should contain comments needed for explaining the work a piece of code does. Comments may be needed to explain why it does it, or, why it does a particular way. The student can explain the function of the code with a shot of the output screen of that program code.

- **Code Efficiency:** The student should explain how the code is efficient and how the students have handled code optimisation.

Testing Approach: Testing should be according to the scheme presented in the system design chapter and should follow some suitable model – e.g., category partition, state machine-based. Both functional testing and user-acceptance testing are appropriate. Explain the approach of testing.

- **Unit Testing:** Unit testing deals with testing a unit or module as a whole. This would test the interaction of many functions but, do confine the test within one module.

- Integrated Testing: Brings all the modules together into a special testing environment, then checks for errors, bugs and interoperability. It deals with tests for the entire application. Application limits and features are tested here.

Modifications and Improvements: Once the students finish the testing they are bound to be faced with bugs, errors and they will need to modify your source code to improve the system. Define what modification are implemented in the system and how it improved the system.

Chapter 6: Results and Discussion

Test Reports: Explain the test results and reports based on the test cases, which should show that the project is capable of facing any problematic situation and that it works fine in different conditions. Take the different sample inputs and show the outputs.

User Documentation: Define the working of the software; explain its different functions, components with screen shots. The user document should provide all the details of the product in such a way that any user reading the manual, is able to understand the working and functionality of the document.

Chapter 7: Conclusions

Conclusion: The conclusions can be summarised in a fairly short chapter (2 or 3 pages). This chapter brings together many of the points that would have made in the other chapters. Limitations of the System: Explain the limitations encountered during the testing of the project that the students were not able to modify. List the criticisms accepted during the demonstrations of the project.

Future Scope of the Project describes two things: firstly, new areas of investigation prompted by developments in this project, and secondly, parts of the current work that was not completed due to time constraints and/or problems encountered.

REFERENCES

It is very important that the students acknowledge the work of others that they have used or adapted in their own work, or that provides the essential background or context to the project.

The use of references is the standard way to do this. Please follow the given standard for the references for books, journals, and online material. The citation is mandatory in both the reports.

E.g:

Linhares, A., & Brum, P. (2007). Understanding our understanding of strategic scenarios: What role do chunks play? *Cognitive Science*, 31(6), 989-1007.

<https://doi.org/doi:10.1080/03640210701703725>

Lipson, Charles (2011). *Cite right : A quick guide to citation styles; MLA, APA, Chicago, the sciences, professions, and more (2nd ed.)*. Chicago [u.a.]: University of Chicago Press. p. 187.

ISBN 9780226484648.

Elaine Ritchie, J Knite. (2001). *Artificial Intelligence, Chapter 2 ,p.p 23 - 44*. Tata McGrawHill.

GLOSSARY

If you the students any acronyms, abbreviations, symbols, or uncommon terms in the project report then their meaning should be explained where they first occur. If they go on to use any of them extensively then it is helpful to list them in this section and define the meaning.

APPENDICES

These may be provided to include further details of results, mathematical derivations, certain illustrative parts of the program code (e.g., class interfaces), user documentation etc.

In particular, if there are technical details of the work done that might be useful to others who wish to build on this work, but that are not sufficiently important to the project as a whole to justify being discussed in the main body of the project, then they should be included as appendices.

VI. SUMMARY

Project development usually involves an engineering approach to the design and development of a software system that fulfils a practical need. Projects also often form an important focus for discussion at interviews with future employers as they provide a detailed example of what the students are capable of achieving. In this course the students can choose your project topic from the lists given in Unit 4: Category-wise Problem Definition.

VII. FURTHER READINGS

1. Modern Systems Analysis and Design; Jeffrey A. Hoffer, Joey F. George, Joseph,S. Valacich; Pearson Education; Third Edition; 2002.
2. ISO/IEC 12207: Software Life Cycle Process
(<http://www.software.org/quagmire/descriptions/iso-iec12207.asp>).
3. IEEE 1063: Software User Documentation (<http://ieeexplore.ieee.org>).
4. ISO/IEC: 18019: Guidelines for the Design and Preparation of User Documentation for Application Software.
5. <http://www.sce.carleton.ca/squall>.
6. <http://en.tldp.org/HOWTO/Software-Release-Practice-HOWTO/documentation.html>.
7. <http://www.sei.cmu.edu/cmm/>

PROFORMA FOR THE APPROVAL PROJECT PROPOSAL

(Note: All entries of the proforma of approval should be filled up with appropriate and complete information. Incomplete proforma of approval in any respect will be summarily rejected.)

PNR No.:

Rollno: _____

1. Name of the Student

2. Title of the Project

3. Name of the Guide

4. Teaching experience of the Guide _____

5. Is this your first submission? Yes No

Signature of the Student

Signature of the Guide

Date:

Date:

Signature of the Coordinator

Date:

(All the text in the report should be in times new roman)

TITLE OF THE PROJECT
(NOT EXCEEDING 2 LINES, 24 BOLD, ALL CAPS)

A Project Report (12 Bold)

Submitted in partial fulfillment of the

Requirements for the award of the Degree of (size-12)

BACHELOR OF SCIENCE (INFORMATION TECHNOLOGY)

(14 BOLD, CAPS)

By(12 Bold)

Name of The Student (size-15, title case)

Seat Number (size-15)

Under the esteemed guidance of (13 bold)

Mr./Mrs. Name of The Guide (15 bold, title case)

Designation (14 Bold, title case)

COLLEGE LOGO

DEPARTMENT OF INFORMATION TECHNOLOGY(12 BOLD, CAPS)

COLLEGE NAME (14 BOLD, CAPS)

(Affiliated to University of Mumbai) (12, Title case, bold, italic)

CITY, PIN CODE(12 bold, CAPS)

MAHARASHTRA (12 bold, CAPS)

YEAR (12 bold)

COLLEGE NAME (14 BOLD, CAPS)

(Affiliated to University of Mumbai) (13, bold, italic)

CITY-MAHARASHTRA-PINCODE(13 bold, CAPS)

DEPARTMENT OF INFORMATION TECHNOLOGY (14 BOLD, CAPS)

College Logo

CERTIFICATE (14 BOLD, CAPS, underlined, centered)

This is to certify that the project entitled, "**Title of The Project** ", is bonafied work of **NAME OF THE STUDENT** bearing Seat.No: (**NUMBER**) submitted in partial fulfillment of the requirements for the award of degree of BACHELOR OF SCIENCE in INFORMATION TECHNOLOGY from University of Mumbai. (12, times new roman, justified)

Internal Guide (12 bold)

Coordinator

(Don't write names of lecturers or HOD)

External Examiner

Date:

College Seal

COMPANY CERTIFICATE (if applicable)

(Project Abstract page format)

Abstract (20bold, caps, centered)

Content (12, justified)

Note: Entire document should be with 1.5 line spacing and all paragraphs should start with 1 tab space.

ACKNOWLEDGEMENT

(20, BOLD, ALL CAPS, CENTERED)

The acknowledgement should be in times new roman, 12 font with 1.5 line spacing, justified.

(Declaration page format)

DECLARATION (20 bold, centered, allcaps)

Content (12, justified)

I hereby declare that the project entitled, "Title of the Project" done at **place where the project is done**, has not been in any case duplicated to submit to any other university for the award of any degree. To the best of my knowledge other than me, no one has submitted to any other university.

The project is done in partial fulfillment of the requirements for the award of degree of **BACHELOR OF SCIENCE (INFORMATION TECHNOLOGY)** to be submitted as final semester project as part of our curriculum.

Name and Signature of the Student

TABLE OF CONTENTS (20 bold, caps, centered)

Should be generated automatically using word processing software.

List of Tables (20 bold, centered, Title Case)

Should be generated automatically using word processing software.

List of Figures (20 bold, centered, Title Case)

Should be generated automatically using word processing software.

**** *The plagiarism should be maintained as per the UGC guidelines.***

**** NOTE ABOUT PROJECT VIVA VOCE:**

Student may be asked to write code for problem during VIVA to demonstrate his coding capabilities and he/she may be asked to write any segment of coding used in the in the project.

The project can be done in group of at most four students. However, the length and depth of the project should be justified for the projects done in group. A big project can be modularised and different modules can be assigned as separate project to different students.

Marks Distribution:

Semester V: 50 Marks

Documentation: 50 marks

Semester VI: 150 Marks

Documentation: 50 Marks:

Implementation and Viva Voce: 100 Marks

Evaluation Scheme:**1. Internal Evaluation (25 Marks).****i. Test: 1 Class test of 20 marks. (Can be taken online)**

Q	Attempt <u>any four</u> of the following:	20
a.		
b.		
c.		
d.		
e.		
f.		

ii. 5 marks: Active participation in the class, overall conduct, attendance.**2. External Examination: (75 marks)**

	All questions are compulsory	
Q1	(Based on Unit 1) Attempt <u>any three</u> of the following:	15
a.		
b.		
c.		
d.		
e.		
f.		
Q2	(Based on Unit 2) Attempt <u>any three</u> of the following:	15
Q3	(Based on Unit 3) Attempt <u>any three</u> of the following:	15
Q4	(Based on Unit 4) Attempt <u>any three</u> of the following:	15
Q5	(Based on Unit 5) Attempt <u>any three</u> of the following:	15

3. Practical Exam: 50 marks

A Certified copy journal is essential to appear for the practical examination.

1.	Practical Question 1	20
2.	Practical Question 2	20
3.	Journal	5
4.	Viva Voce	5

OR

1.	Practical Question	40
2.	Journal	5
3.	Viva Voce	5

Sign of Chairperson
Dr. Mrs. R.
Srivaramangai
Ad-hoc BoS (IT)

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

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