V. G. Vaze College of Arts, Science and Commerce (Autonomous)

Syllabus for B.Sc. (Information Technology)
(Sem. – V & VI)
(Choice Based Credit System)

(Introduced from the academic year 2025-26)

Semester V			
Code	Course Type	Course Title	Credits
VSIT300	Major	Software Project Management	2
VSIT301	Major	Next Generation Technologies	2
VSIT302	Major	Next Generation Technologies Practical	2
VSIT303	Major	Linux Administration	2
VSIT304	Major	Linux Administration Practical	2
VSIT305	Elective 1	Geographic Information system	2
VSIT306	Elective 1 Practical	Geographic Information system Practical	2
VSIT307	Elective 2	Enterprise JAVA	2
VSIT308	Elective 2 Practical	Enterprise JAVA Practical	2
VSIT309	Minor	Artificial Intelligence	2
VSIT310	Minor	Artificial Intelligence Practical	2
VSIT311	Vocational Skill Enhancement Course	MERN	2
VSIT312	Field Project	Project	2
		Total	22

Semester VI			
Code	Course Type	Course Title	Credits
VSIT350	Major	DevOps	2
VSIT351	Major	Security in Computing	2
VSIT352	Major	Security in Computing Practical	2
VSIT353	Major	Business Intelligence	2
VSIT354	Major	Business Intelligence Practical	2
VSIT355	Elective 1	Cloud Computing	2
VSIT356	Elective 1 Practical	Cloud Computing Practical	2
VSIT357	Elective 2	Virtual & Augmented Reality	2
VSIT358	Elective 2 Practical	Virtual & Augmented Reality Practical	2
VSIT359	Minor	Exploratory Data Analysis	2
VSIT360	Minor	Exploratory Data Analysis Practical	2
VSIT361	Research Project	Project	4
		Total	22

Program Specific Outcomes Information Technology

PSO 1: Software Development and Programming Proficiency :

Graduates will be able to develop web, mobile, and desktop applications using modern programming languages (Python, Java, C++, JavaScript, etc.). and also able to apply data structures, algorithms, and design patterns for efficient software development.

PSO 2: Database and Data Management :

Graduates will be able to design and manage relational (SQL) and NoSQL databases and develop expertise in data warehousing, data mining, and business intelligence.

PSO 3: Software Engineering and Project Management :

Graduates will be able to apply methodologies for software development lifecycle and develop IT solutions addressing real-world business and industrial problems.

PSO 4: Networking and Cybersecurity Skills :

Graduates will be able to Configure and manage computer networks, network security, and firewalls and understand cybersecurity principles

PSO 5: Artificial Intelligence and Cloud Computing :

Graduates will be able to apply the concepts of artificial intelligence to solve real-world problems and understand cloud architecture (IaaS, PaaS, SaaS).

Program Outcomes Information Technology

1. Fundamental Knowledge and problem solving skills :

- ➢ Graduates will be able to understand core concepts of computer science, mathematics, logic and apply knowledge of data structures, algorithms, and programming languages.
- ➢ Graduates will be able to analyze complex IT problems and find optimized solutions and develop skills in debugging, testing, and performance optimization.

2. Software and Hardware Proficiency :

- ➢ Graduates will be able to gain hands-on experience with software development tools, IDEs, and frameworks.
- ➢ Graduates will be able to understand computer architecture, operating systems, and hardware components.
- ➢ Graduates will be able to work with cloud computing and virtualization technologies.

3. Database and Information Management :

- ➢ Graduates will be able to design and implement relational and NoSQL databases.
- ➢ Graduates will be able to develop knowledge of big data technologies and business intelligence.

4. Networking and Cybersecurity :

- ➢ Graduates will be able to understand the fundamentals of networking, protocols, and security mechanisms.
- ➢ Graduates will be able to implement firewalls, encryption, and intrusion detection systems.
- ➢ Graduates will be able to apply cybersecurity practices to prevent cyber threats and data breaches.

5. Artificial Intelligence :

- ➢ Graduates will understand the concepts of Artificial Intelligence and apply AI to solve real life problem
- ➢ Graduates will use Python / R AI automation.

6. Project Management and Professional Ethics :

- ➢ Graduates will be able to use software development methodologies to plan, execute, and manage IT projects.
- ➢ Graduates will understand IT laws, intellectual property rights.

SEMESTER V

B. Sc. (Information Technology)		Semester – V	
Course Name: Software Project Management		Course Code: VSIT300	
Periods per week (1 Period is 60 minutes)		2	
Credits		2	
		Hours	**Marks**
Evaluation System	Theory Examination	2	60
	Internal	--	40

Course Objective
To make learner aware of
1. Basics of software project management.
2. Various project approaches and effort estimation.
3. Risk management, activity planning, and activity planning
4. People management in software development, managing contracts, monitoring and control
5. Software quality and team management

Unit	Details	Lectures
I	**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. **Project Evaluation and Programme Management**: Introduction, Business Case, Project Portfolio Management. **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,Step4:Identify Project Products and activities, Step 5: Estimate Effort for Each Activity, Step 6: Identify Activity Risks,Step7:Allocate Resources,Step8:Review/Publicize Plan, Steps 9 and10:ExecutePlan/Lower Levels of Planning	10

	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, Atern/Dynamic Systems Development Method, Scrum, Lean Software Development, Managing Iterative Processes, Selecting the Most Appropriate Process Model. **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 Judgment, Estimating by Analogy, Albrecht Function Point Analysis.	
II	**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. **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"sTop10 Risks and Counter Measures, Applying the PERT Technique, Monte Carlo Simulation, Critical Chain Concepts. **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.	**10**
III	**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). **Managing Contracts**: Introduction, Types of Contract, Stages in Contract Placement, Typical Terms of a Contract, Contract Management, Acceptance. **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.	**10**

| | **Working in Teams:** Introduction, becoming a Team, Decision Making, Organization and Team Structures, Coordination Dependencies, Dispersed and Virtual Teams, Communication Genres, Communication Plans, Leadership.
 Software Quality: Introduction, The Place of Software Quality in Project Planning, Importance of Software Quality, Defining Software Quality, Software Quality Models.
 Project Closeout: Introduction, Reasons for Project Closure, Project Closure Process, Performing a Financial Closure, Project Closeout Report. | |

Course Outcome
Learner will be able to
CO1   To understand the basic concepts, terminologies and issues of software project management. To apply appropriate methods and models for the development of solutions. To analyze the cost-benefits of calculations so as to optimize the selection strategy
CO2   To understand and implement the concepts of resource allocation, risk management and activity planning.
CO3   To understand Principles of organizational behavior and motivation to manage software teams effectively, focusing on leadership, communication, and decision-making processes.

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

B.Sc.(Information Technology)		Semester–V	
Course Name: Project		**Course Code:** VSIT312	
Periods per week(1Period is 120 minutes)		2	
Credits		2	
		Hours	**Marks**
Evaluation System	**Practical Examination**	2	100
	Internal	--	--

The details are given in Appendix – I

B. Sc. (Information Technology)		Semester – V	
Course Name: Next Generation Technologies		**Course Code:** VSIT301	
Periods per week (1 Period is 60 minutes)		2	
Credits		2	
		Hours	**Marks**
Evaluation System	**Theory Examination**	2	60
	Internal	--	40

Course Objective
To make learner aware of
1. Concept of JSON.
2. Concept of Big data, NoSQL, MongoDB.
3. MongoDB architecture, shell and data model.
4. MongoDB Storage engine, use cases, limitation and best practices.
5. Concept of JQuery.

Unit	Details	Lectures
I	**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. **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 **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. **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	10
II	**The MongoDB Data Model**: The Data Model, JSON and BSON, The Identifier (_id), Capped Collection, Polymorphic Schemas, Object- Oriented Programming, Schema Evolution	10

	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 **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		
III	**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 **JQuery Effects:** (jQuery Hide/Show,jQuery Fade,jQuery Slide, jQuery Animate, jQuery stop(). jQuery Callback, jQuery Chaining), Events (binding, removing, event attributes–alt Key, ctrl Key, data, key Code, meta Key, shift Key, type, event methods - bind, hover, on, off, ready, trigger, unbind, blur, click, dblclick, change, keydown, keypress, keyup, mouseover, mouseup, mousemove, select, submit), Ajax with jQuery, jQuery Plug-ins (flickerplate, slideshow) , replaceWith, text, wrap, **JQuery Effects:** (jQuery Hide/Show, jQuery Fade, jQuery Slide, jQueryAnimate jQuery stop(). jQuery Callback, jQuery Chaining), Events (binding, removing, event attributes –alt Key, ctrl Key, data, key Code, meta Key, shift Key, type, event methods - bind, hover, on, off, ready, trigger, unbind, blur, click, dblclick, change, keydown, keypress, keyup, mouseover, mouseup, mousemove, select, submit), Ajax with jQuery, jQuery Plug-ins (flickerplate, slideshow).	**10**

Course Outcome	
Learner will be able to	
CO1	Get familiar with the concept of Big Data, JSON and MongoDB.
CO2	Comprehend the MongoDB data mode, perform basic querying operations in MongoDB using the shell as well as understand the core processes and components of MongoDB architecture.
CO3	Understand the limitations of traditional disk-based storage and the advantages of SSD and in-memory databases as well as learn how to use jQuery.

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

B. Sc. (Information Technology)		Semester – V	
Course Name: Next Generation Technologies Practical		**Course Code: VSIT302**	
Periods per week (1 Period is 120 minutes)		**2**	
Credits		**2**	
		Hours	**Marks**
Evaluation System	**Practical Examination**	**3**	**100**
	Internal	**--**	**--**

Course Objective
To make learner aware of :
1. JSON concepts including its syntax, data structures, and practical applications, enabling to effectively work with JSON data in various programming contexts.
2. A foundational understanding of NoSQL databases, MongoDB's document-oriented data model, and practical skills in performing CRUD operations, data modeling, and utilizing advanced features like indexing and aggregation.
3.   Developing Java, PHP and Python applications using MongoDB as the database.
4. The syntax and structure of jQuery, including selectors, methods, and event handling, to modify HTML elements and their attributes, add or remove elements, and traverse the DOM tree , event handling. As well as effects and animations.

Sr. No.	Details
1	**JSON**
a	Creating JSON
b	Parsing JSON
c	Persisting JSON
2	**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.
3	
a	Create a JSON file and import it to MongoDB
b	Export MongoDB to JSON.
c	Write a MongoDB query to delete JSON object from MongoDB
4	**Simple Queries with MongoDB**

5		**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.
6		**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.
7		**Java and MongoDB**
		Connecting Java with MongoDB and inserting, retrieving, updating and deleting.
8		**PHP and MongoDB**
		Connecting PHP with MongoDB and inserting, retrieving, updating and deleting.
9		**Python and MongoDB**
		Connecting Python with MongoDB and inserting, retrieving, updating and deleting.
10		**Programs on Basic jQuery**
a		jQuery Basic, jQuery Events
b		jQuery Selectors, jQuery Hide and Show effects
c		jQuery fading effects, jQuery Sliding effects
11		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

Course Outcome	
Learner will be able to	
CO1	Gain hands-on experience in creating and manipulating JSON data.
CO2	Write MongoDB queries for data retrieval and manipulation.
CO3	Learn how to create a replica set in MongoDB to mirror a database as well as how to back up MongoDB data and restore it from a backup.
CO4	Learn how to connect Java, PHP and Python applications to MongoDB.
CO5	Gain understanding of jQuery, including working with jQuery events, selectors, and effects AND create complex animations and effects

B. Sc. (Information Technology)		Semester – V	
Course Name: Linux Administration		Course Code: VSIT303	
Periods per week (1 Period is 60 minutes)		2	
Credits		2	
		Hours	Marks
Evaluation System	Theory Examination	2	60
	Internal	--	40

Course Objective
Learner will be able to:
1. Understand the Fundamentals of Red Hat Enterprise Linux and Install and Configure Red Hat Enterprise Linux.
2. Administer Users and Groups.
3. Manage Software Packages and System Updates.
4. Develop Shell Scripts for Task Automation.
5. Configure TCP/IP Networking and Network Services ,Deploy Domain Name System (DNS) Servers, Administer Web and Internet Services, Implement File Sharing and Network Access Services.

Unit	Details	Lectures
I	**Introduction to Red Hat Enterprise Linux:** Linux, Open Source and Red Hat, Origins of Linux, Distributions, Duties of Linux System Administrator. **Installing Red Hat Linux** **RedHat Linux File System:** Understanding the Red Hat Linux File System Structure, Using File System Commands, Working With Linux – Supported File Systems, Linux Disk Management **Red Hat System Configuration Files:** Examining the System Configuration Files, Examining the Network Configuration Files, **Administering Users and Groups:** Understanding the Root Account, Implementing Sudo, Working With Users and Groups, Using File System Quotas, **Installing and Upgrading Software Packages:** Using the Red Hat Package Manager, Checking Software Versions, Obtaining Newer Software, Installing Software, **Using Scripts to Automate Tasks:** Understanding Bash Programming, Using Processes and Job Control, Creating Backups, Automating Scripts, Writing, Testing, and Debugging Scripts, Selecting a Scripting Language	10

II	**TCP/IP Networking:** TCP/IP Explained, Understanding Network Classes, Setting Up a Network Interface Card (NIC), Understanding Subnetting, Classless Inter Domain Routing(CIDR), Gateways and Routers, Configuring Dynamic Host Configuration Protocol(DHCP), Configuring a Point-to-Point Protocol(PPP) Connection **The Network File System:** NFS Overview, Configuring an NFS Server, Examining NFS Security **Connecting to Microsoft Networks:** Installing Samba, Configuring the Samba Server, Configuring the Samba Client, Using a Windows Printer from the Linux Computer, Testing the Samba Server **Internet Services:** Secure Services, Less Secure Services, Using Your Linux Machine as a Server, The Inetd Server, Xinetd, Inetd and Xinetd vs.Stand-Alone, Linux Firewall Packages	10
III	**The Domain Name System:** Understanding DNS, Installing the Software, Understanding Types of Domain Servers, Examining Server Configuration Files, Configuring a CachingServer, Configuring a SlaveServer, Configuring a MasterServer, Using DNS Tools **Configuring FTP Services :** What FTP Software is Available?, Installing WU-FTPD, Configuring the Server, Maintaining the Server, Strengthening FTP Security, **Configuring Mail Services :** E-Mail Explained, Introducing SMTP, Configuring Sendmail, Configuring the E-Mail Client, Using Elm, Maintaining E-Mail Security, Using Newsgroups **Configuring a Web Server:** Introducing Apache, How Web Servers Work, Installing Apache, Configuring Apache, Configuring Apache for SSI, CGIScripts, Creating a Secure Server with SSL **Securing Server with iptables**	10

Course Outcome	
Learner will be able to	
CO1	Understanding the Fundamentals of Red Hat Enterprise Linux (RHEL) and Managing the Linux File System and System Configuration
CO2	User and Group Administration
CO3	Installing and Managing Software Packages and Securing Linux Servers using iptables and Security Policies
CO4	Automating Tasks Using Shell Scripting
CO5	Configuring TCP/IP Networking and Network Services, Domain Name System (DNS) and FTP Services, Implementing Network File System (NFS) and Samba, Administering Internet and Web Services

Books and References:

Sr. No.	Title	Author/s	Publisher	Edition	Year
1.	Red hat Linux Networking and System Administration	Terry Collings and Kurt Wall	Wiley	3rd	
2.	Red Hat Enterprise Linux 6 Administration	Sander van Vugt	John Wiley and Sons		2013
3.	Linux Administration: A Beginner's Guide	Wale Soyinka	TMH	5th	

B. Sc. (Information Technology)		Semester – V	
Course Name: Linux Administration Practical		Course Code: VSIT304	
Periods per week (1 Period is 120 minutes)		2	
Credits		2	
		Hours	Marks
Evaluation System	Practical Examination	3	100
	Internal	--	--

Course Objective
Learner will be able to :
1. Understand and perform installation of RHEL 6.X and navigate both GUI and CLI environments.
2. Manage users, groups, permissions, and understand Linux file systems.
3. Work with RPM packages and configure network settings in Linux and create and manage local repositories for software installation and updates
4. Set up and manage various servers including file sharing (NFS, Samba, FTP), DNS, DHCP, Mail, and Web Servers and configure firewall rules using iptables for system security
5. Write and execute shell scripts to automate system tasks and manage GRUB bootloader settings.

List of Practical :	
1.	Installation of RHEL 6.X

2.	Graphical User Interface and Command Line Interface and Processes
	    a. Exploring the Graphical Desktop
	    b. The Command Line Interface
	    c. Managing Processes

3.	Working with Users, Groups, and Permissions

4.	Working with RPM Storage and Networking
	    a. Using Query Options
	    b. Extracting Files From RPMs
	    c. Configuring and Managing Storage
	    d. Connecting to the Network

5.	Configuring Server for File Sharing
	    a. Configuring NFS Server and Client
	    b. Configuring Samba
	    c. Configuring FTP

6.	DNS, DHCP and Mail Server
	    a. Configuring DNS
	    b. Configuring DHCP
	    c. Setting Up a Mail Server

7.	Web Server
	    a. Configuring Apache on Red Hat Enterprise Linux
	    b. Writing a Script to Monitor Activity on the Apache Web Server

8.	Shell Scripts
	    a. Writing Shell Scripts
	    b. Configuring Booting with GRUB

9.	Working with iptable commands

10.	Storage Devices and Links, Backup and Repository
	    a. Working with Storage Devices and Links
	    b. Making a Backup
	    c. Creating a Repository

Course Outcome
By the end of this course, students will be able to:
CO1   Install and configure RHEL 6.X, and effectively operate both graphical and command-line interfaces.
CO2   Manage users, groups, and permissions to ensure proper access control and security in Linux systems.
CO3   Use RPM commands for package management and configure storage and basic networking and use iptables to implement firewall rules and perform system backups and repository management.
CO4   Set up file sharing services using NFS, Samba, and FTP servers, configure essential network services including DNS, DHCP, and Mail servers for enterprise environments. as well as set up and monitor Apache Web Server, and write basic monitoring and automation scripts.
CO5   Write shell scripts for system automation and configure system boot settings using GRUB.

B. Sc. (Information Technology)		Semester – V	
Course Name: Geographic Information System		Course Code: VSIT305	
Periods per week (1 Period is 60 minutes)		2	
Credits		2	
		Hours	**Marks**
Evaluation System	Theory Examination	2	60
	Internal	--	40

Course Objective
To make learner aware of
1. Basics of GIS.
2. Concepts of real world and its representation in GIS.
3. Concept that how the computer representation can be done in GIS.
4. Concept of Stages of spatial data handling, spatial database and spatial query analysis.
5. Concept of spatial referencing, satellite-based positioning, data entry and preparation in GIS.
6. Concept of data entry, combining data from multiple sources and preparation of data in GIS.

Unit	Details	Lectures
I	**Introduction to GIS, The nature of GIS:** Some fundamental observations, Defining GIS, GISsystems, GISscience andGIS Applications, Spatial data and Geoinformation. **The real world and representations of it:** Models and modeling, Maps, Databases, Spatial databases and spatial analysis **Models and Representations of the real world, Geographic** **Phenomena:** Defining geographic phenomena, types of geographic phenomena, Geographic fields, Geographic objects, Boundaries. **Computer Representations of Geographic Information:** Regular tessellations, irregular tessellations, Vector representations, Topology and Spatial relationships, Scale and Resolution, Representation of Geographic fields, Representation of Geographic objects **Organizing and Managing Spatial Data The Temporal Dimension**	10
II	**Data Management and Processing Systems Hardware and Software Trends, Spatial Referencing and Positioning Geographic Information Systems:** GIS Software, GIS Architecture and functionality, Spatial Data Infrastructure (SDI)	10

	Stages of Spatial Data handling: Spatial data handling and preparation, Spatial Data Storage and maintenance, Spatial Query and Analysis, Spatial Data Presentation. **Database management Systems:** Reasons for using a DBMS, Alternatives for data management, The relational data model, Querying the relational database. **GIS and Spatial Databases:** Linking GIS and DBMS, Spatial database functionality. **Spatial Referencing:** Reference surfaces for mapping, Coordinate Systems, Map Projections, Coordinate Transformations **Satellite-based Positioning:** Absolute positioning, Errors in absolute positioning, Relative positioning,	
III	**Data Entry and Preparation** **Spatial Data Input:** Direct spatial data capture, Indirect spatial data capture, Obtaining spatial data elsewhere. **Data Quality:** Accuracy and Positioning, Positional accuracy, Attribute accuracy, temporal accuracy, Lineage, Completeness, Logical consistency **Data Preparation:** Data checks and repairs, Combining data from multiple sources. **Point Data Transformation:** Interpolating discrete data, Interpolating continuous data. **Data Visualization GIS and Maps, The Visualization Process Visualization Strategies: Present or explore?** **The cartographic toolbox:** What kind of data do I have? How can I map my data? **How to map?** How to map qualitative data, How to map quantitative data, How to map the terrain elevation, How to map time series **Map Cosmetics, Map Dissemination**	10

Course Outcome	
Learner will be able to	
CO1	Understand the GIS and observations of GIS.
CO2	Understand the models, representations and geographic phenomena.
CO3	Understand how computer representation is used to represent geographic phenomena.
CO4	Understand the concept of spatial data, how to handle spatial data, spatial database and how spatial query can be written and analyzed.
CO5	Understand the concept of spatial referencing, satellite - based positioning which can be used as an input and for data quality.

Books and References:

Sr. No.	Title	Author/s	Publisher	Edition	Year
1.	Principles of Geographic Information Systems- An Introductory Text Book	Editors: Otto Huisman and Rolf A.	The International Institute of Geoinformation Science and Earth Observation	Fourth	2009
2.	Principles of Geographic Information Systems	P.A Burrough and R.A. McDonnell	Oxford University Press	Third	1999
3.	Fundamentals of Spatial Information Systems	R.Laurini and D. Thompson,	Academic Press		1994
4.	Fundamentals of Geographic Information Systems	Michael N.Demers	Wiley Publications	Fourth	2009
5.	Introduction to Geographic Information Systems	Chang Kang- tsung (Karl),	McGraw Hill	Any above 3rdEdition	2013 7thEdition
6.	GIS Fundamentals: A First Text on Geographic Information Systems	Pau Bolsatd	XanEdu Publishing Inc	5thEdition	

B. Sc. (Information Technology)		Semester – V	
Course Name: Geographic Information System Practical		**Course Code: VSIT306**	
Periods per week (1 Period is 120 minutes)		2	
Credits		2	
		Hours	**Marks**
Evaluation System	**Practical Examination**	3	100
	Internal	--	--

List of Practical :	
	Familiarizing Quantum GIS: Installation of QGIS, datasets for both Vector and Raster data, Maps.
1.	Creating and Managing Vector Data: Adding vector layers, setting properties, formatting, calculating line lengths and statistics.
2.	Exploring and Managing Raster data: Adding raster layers, raster styling and analysis.
3.	Exploring raster mosaicking and clipping.
4.	Making a Map, Working with Attributes, Importing Spreadsheets or CSV files Using Plugins , Searching and Downloading OpenStreetMap Data.
5.	Working with attributes, terrain Data.
6.	Working with Projections and WMS Data.
7.	Georeferencing TopoSheets and ScannedMaps.
8.	Georeferencing Aerial Imagery and Digitizing Map Data
9.	Managing Data Tables and Spatial datasets: Table joins, spatial joins, points in polygon analysis, performing spatial queries.
10.	Validating Map data.

Course Outcome
By the end of this course, students will be able to:
CO1 Understand and apply basic GIS concepts in a practical environment.
CO2 Demonstrate the ability to create, manage, and manipulate spatial data layers using GIS software.
CO3 Perform spatial analysis techniques such as buffer, overlay, and spatial query.
CO4 Georeference maps and digitize spatial features with appropriate attributes.
CO5 Create thematic maps and interpret spatial patterns for decision-making.

B. Sc. (Information Technology)		Semester – V	
Course Name: Enterprise JAVA		**Course Code: VSIT307**	
Periods per week (1 Period is 60 minutes)		2	
Credits		2	
		Hours	**Marks**
Evaluation System	**Theory Examination**	2	60
	Internal	--	40

Course Objective
The objectives of this course are to:
1. Introduce Java Enterprise Edition (Java EE) and explain the architecture of Java EE applications, including servers, containers, and system architectures.
2. Teach the fundamentals of Java Servlets, their lifecycle, and how they handle client-server communication.
3. Demonstrate database connectivity using JDBC and explore session management techniques, including cookies and HttpSession, for maintaining user state.
4. Introduce Java Server Pages (JSP) and explain the use of Java Server Pages Standard Tag Library (JSTL) to simplify JSP development.
5. Explain the Java Naming and Directory Interface (JNDI), Introduce Hibernate ORM framework,

Unit	Details	Lectures
I	**Understanding Java EE:** What is an Enterprise Application? What is java enterprise edition? Java EE Technologies, Java EE evolution, Glassfish server **Java EE Architecture, Server and Containers:** Types of System Architecture, Java EE Server, Java EE Containers. **Introduction to Java Servlets:** The Need for Dynamic Content, Java Servlet Technology, Why Servlets? What can Servlets do? **Servlet API and Lifecycle:** Java Servlet API, The Servlet Skeleton, The Servlet Life Cycle, A Simple Welcome Servlet **Working with Servlets:** Getting Started, Using Annotations Instead of Deployment Descriptor. **Working with Databases:** What Is JDBC? JDBC Architecture, Accessing Database, The Servlet GUI and Database Example.	10

	Request Dispatcher: Resquest dispatcher Interface, Methods of Request dispatcher, Request dispatcher Application.	
II	**COOKIES:** Kinds of Cookies, Where Cookies Are Used? Creating Cookies Using Servlet, Dynamically Changing the Colors of A Page **SESSION:** What Are Sessions? Lifecycle of HttpSession, Session Tracking With Servlet API, A Servlet Session Example **Introduction To Java Server Pages:** Why use Java Server Pages? Disadvantages Of JSP, JSP v\s Servlets, Life Cycle of a JSP Page, How does a JSP function? How does JSP execute? About Java Server Pages **Getting Started With Java Server Pages:** Comments, JSP Document, JSP Elements, JSP GUI Example. **Action Elements:** Including other Files, Forwarding JSP Page to Another Page, Passing Parameters for other Actions, Loading a Javabean. **Implicit Objects, Scope and El Expressions:** Implicit Objects, Character Quoting Conventions, Unified Expression Language[UnifiedEl], Expression Language. **Java Server Pages Standard Tag Libraries:** What is wrong in using JSP Scriptlet Tags? How JSTL Fixes JSP Scriptlet's Shortcomings? Disadvantages Of JSTL, Tag Libraries.	**10**
III	**Introduction To Enterprise Javabeans:** Enterprise Bean Architecture, Benefits of Enterprise Bean, Types of Enterprise Bean, Accessing Enterprise Beans, Enterprise Bean Application, Packaging Enterprise Beans **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. **Working with Message Driven Beans:** Lifecycle of a Message Driven Bean, Uses of Message Driven Beans, The Message Driven Beans Example. **Interceptors:** Request and Interceptor, Defining An Interceptor, Around Invoke Method, Applying Interceptor, Adding An Interceptor To An Enterprise Bean, Build and Run the Web Application. **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, Data source Resource Definition in Java EE. **Introduction to Hibernate:** What is Hibernate? Why Hibernate? Hibernate, Database and The Application, Componentsof Hibernate, Architecture of Hibernate. How Hibernate Works? **Writing Hibernate Application:** Application Requirement Specifications, Software Requirements, The Application Development Approach, Creating	**10**

	Database and Tables InMysql, creating a Web Application, Adding the Required Library Files, creating a Javabean Class, Creating Hibernate Configuration File, Adding a Mapping Class, Creating JSPS, Running The Hibernate Application.	

Course Outcome
By the end of this course, students will be able to:

CO1	Understand Enterprise Application Development as well as develop Web Applications Using Servlets and Manage Data with JDBC and Request Handling.
CO2	Implement Session Management Techniques and Develop Dynamic Web Applications with Java Server Pages (JSP).
CO3	Work with Enterprise JavaBeans (EJB) and Implement Java Naming and Directory Interface (JNDI) for Resource Management as well as Develop Enterprise Applications Using Hibernate

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: Build reliable applications with themost robust and mature technology forenterprise development	Elder Moraes	Packt F	First	2018
3.	Advanced Java Programming.	Uttam Kumar Roy	Oxford Press		2015

B. Sc. (Information Technology)		Semester – V	
Course Name: Enterprise JAVA Practical		Course Code: VSIT308	
Periods per week (1 Period is 120 minutes)		2	
Credits		2	
		Hours	Marks
Evaluation System	Practical Examination	3	100
	Internal	--	--

Course Objective
Learners will be able to
1. Design and develop web-based applications using Servlets and JSP and Develop database-driven applications using Java Persistence API (JPA) with object-relational mapping (ORM).
2. Implement Session Management techniques using Cookies, Sessions, and Request Dispatchers and Perform file upload/download and I/O operations using Servlet I/O streams
3. Connect to relational databases using JDBC and perform CRUD operations via Servlets and JSP and Use JSP Expression Language (EL) and Java Standard Tag Library (JSTL) to build dynamic and reusable pages.
4. Build scalable, modular enterprise applications using Enterprise Java Beans (EJB) including Stateless, Stateful, Singleton, and Message-Driven Beans.
5. Implement Hibernate for advanced ORM features, including annotations, associations, and multi-page dynamic websites

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 " else a message "loginfailed" 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). a. 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 no, name, age, designation, salary of employee. Now on submit send this data to a JSP page which will update the employee table of database with matching employee no. 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]. a. 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.

Course Outcome	
By the end of this course, learner will be able to	
CO1	Create and deploy Servlet-based applications handling user input, authentication, and database interactions using JDBC.
CO2	Apply session tracking mechanisms using Cookies, Sessions, and Request Dispatchers to manage user state andHandle file I/O through Servlets for file upload, download, and dynamic file processing
CO3	Build dynamic JSP applications utilizing intrinsic objects, form validations, and multi-page data flow and Utilize JSP EL and JSTL for effective data binding, iteration, and conditionals in JSP pages.
CO4	Develop EJB-based enterprise applications including currency converters, shopping carts, and reservation systems using appropriate session bean types.
CO5	Access and manipulate relational data using JPA in applications like inventory systems, guestbooks, and data viewers. and apply Hibernate ORM concepts to map Java classes to database tables and perform complex data operations.

B. Sc. (Information Technology)		Semester – V	
Course Name: Artificial Intelligence		**Course Code: VSIT309**	
Periods per week (1 Period is 60 minutes)		2	
Credits		2	
		Hours	**Marks**
Evaluation System	**Theory Examination**	2	60
	Internal	--	40

Course Objective
The objectives of this course are to enable students to
1. Understand the Foundations of AI and explore Intelligent Agents and Search Strategies.
2. Analyze Adversarial Search and Game Playing.
3. Develop Logical Reasoning and Knowledge Representation Skill and Apply AI Techniques in Knowledge Representation Systems.
4. Solve Constraint Satisfaction Problems (CSPs).
5. Study AI Planning and Decision Making.

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 behavior, nature of environment, the structure of agents. **Solving Problems by Searching:** Problem solving agents, examples problems, searching for solutions, uninformed search, informed search strategies, heuristic functions.	10
II	**Beyond Classical Search:** local search algorithms, searching with non-deterministic action, searching with partial observations, online search agents and unknown environments. **Adversarial Search:** Games, optimal decisions in games, alphabeta pruning, stochastic games, partially observable games, stateof-the-are game programs. **Logical Agents:** Knowledge base agents, The Wumpus world, logic, propositional logic, propositional theorem proving, effective propositional model checking, agents based on propositional logic **Constraint Satisfaction Problems:** Defining Constraint Satisfaction Problems, Constraint Propagation: Inference in CSPs, Backtracking Search for CSPs, Local Search for CSPs, The Structure of Problems.	10
III	**First Order Logic:** Syntax and semantics, using First Order Logic, Knowledge engineering in First Order Logic.	10

	Inference in First Order Logic: propositional vs. First Order, unification and lifting, forward and backward chaining, resolution. **Planning:** Definition of Classical Planning, Algorithms for planning as state 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. **Knowledge Representation:** Categories and Objects, events, mental events and objects, reasoning systems for categories, reasoning with default information, Internet shopping world	

Course Outcome	
Learner will be able to	
CO1	Understanding the Fundamentals of AI
CO2	Problem Solving using Search Techniques
CO3	Adversarial Search, Game Playing and Solving Constraint Satisfaction Problems (CSPs)
CO4	Knowledge Representation and Logical Reasoning and AI-Based Knowledge Representation Systems
CO5	AI Planning and Decision Making

Books and References:

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

B. Sc. (Information Technology)		Semester – V	
Course Name: Artificial Intelligence Practical		**Course Code: VSIT310**	
Periods per week (1 Period is 120 minutes)		2	
Credits		2	
		Hours	**Marks**
Evaluation System	**Practical Examination**	3	100
	Internal	--	--

Course Objective
The objectives of this course are to enable students to
1. Develop a solid understanding of fundamental algorithms such as Depth First Search (DFS) and Breadth First Search (BFS) and design and implement game-theoretic applications such as Tic-Tac-Toe using Min-Max algorithm to understand decision-making processes in AI.
2. Learn and implement artificial intelligence techniques such as Alpha-Beta pruning, Hill climbing, and A* algorithm to solve optimization and search problems.
3. Gain skills in recursive problem-solving by tackling classical problems like the Towers of Hanoi, N-Queens, and others that leverage recursive algorithms.
4. Explore the domain of constraint satisfaction problems (CSP) by implementing various examples such as Sudoku, map coloring, and other logical puzzles.
5. Understand predicate logic and its applications in AI by implementing rules and relationships in a family tree context.

List of Practical :	
1.	a. Write a program to implement depth first search algorithm.
	b. Write a program to implement breadth first search algorithm.
2.	a. Write a program to simulate 4-Queen / N-Queen problem.
	b. Write a program to solve tower of Hanoi problem.
3.	a. Write a program to implement alpha beta search.
	b. Write a program for Hill climbing problem.
4.	a. Write a program to implement A* algorithm.
	b. Write a program to implement AO* algorithm.
5.	a. Write a program to solve water jug problem.
	b. Design the simulation of tic – tac – toe game using min-max algorithm.
6.	a. Write a program to solve Missionaries and Cannibals problem.
	b. Design an application to simulate number puzzle problem.
7.	a. Write a program to shuffle Deck of cards.
	b. Solve traveling salesman problem using artificial intelligence technique.
8.	Solve constraint satisfaction problem

	a. Sudoku Solving using CSP b. Map Coloring c. Zebra Puzzle d. Magic Squares
9.	Implementation of heuristic search techniques in Python.
10.	Write a program to derive the predicate. (for e.g.: Sachin is batsman , batsman is cricketer) - > Sachin is Cricketer. b 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 FamilyTree. ii. Define: Clauses, Facts, Predicates and Rules with conjunction and disjunction

Course Outcome	
By the end of this course, students should be able to:	
CO1	Successfully write, test, and debug programs that use DFS and BFS for navigating graphs and create a functional Tic-Tac-Toe game using predictive algorithms, showcasing an understanding of game theory and decision-making algorithms.
CO2	Implement and explain various artificial intelligence search techniques, including A* and Alpha-Beta pruning, and apply them to a range of problems.
CO3	Demonstrate proficiency in solving recursive problems such as N-Queens and Towers of Hanoi, articulating the recursive structure in solutions.
CO4	Employ various algorithms to solve constraint satisfaction problems, demonstrating an understanding of their underlying principles and how they apply to puzzles like Sudoku and map coloring.
CO5	Write and analyze programs that implement predicate logic for representing relationships and rules in a family tree, including defining clauses, facts, and rules.

B. Sc. (Information Technology)		Semester – V	
Course Name: : MERN		**Course Code: VSIT311**	
Periods per week (1 Period is 120 minutes)		2	
Credits		2	
		Hours	**Marks**
Evaluation System	**Practical Examination**	2	60
	Internal	--	40

Course Objective
To make learner able to :
1. Understand the Node environment, including installation, hosting environments, and Node C/C++ Add-ons.
2. Explore Node's building blocks, such as global objects, events, asynchronous event handling, and the basics of Node modules and npm.
3. Understand React and its foundational concepts, including React elements, components, JSX, and functional programming with JavaScript.
4. Provide an introduction to TypeScript and cover core concepts of Angular, including modules, components, directives, and decorators.
5. Cover advanced topics in Node and React, including state management with hooks, routing, forms, HTTP, and dependency injection in Angular.

Unit	Details	Lectures
I	What is node JS ? Basic knowledge of Node JS **Mastering React: Foundations, State Management, Routing, and Data Handling** **Introduction to React:** A Strong Foundation, React's Past and Future, React Developer Tools, JavaScript forReact, Functional Programming with JavaScript **How React Works:** Page Setup, React Elements, ReactDOM, React Components, React with JSX, Mapping Arrays with JSX, React Fragments. **React State Management & Hooks:** The useState Hook, State in Component Trees, Sending State Down a Component Tree, Sending Interactions Back up a Component, Using Refs, useContext, useEffect, useReducer **Routing, Incorporating Data & Error Handling** : React Router, Requesting Data, Render Props, Suspense, Error Boundaries	**10**

II	**Mastering Angular: TypeScript Fundamentals, Core Concepts, and Essential Features** **Introduction:** Intro to TypeScript, Template, Angular Modules, Bootstrapping, Component Tree, Classes, Looping, Property & Event Binding **Core :** Angular CLI, Lifecycle Hooks, ViewChildren & ContentChildren, Components, Built-in Directives, Decorators, Modules, Pipes	**10**
III	Advanced Angular: Forms, HTTP Handling, and Routing Techniques **Forms:** Template Driven Forms, Reactive Model Form, Dependency Injection & Providers: Injectors, Provider, **HTTP:** Core HTTP API, Promises, Handling errors, Headers, Observables **Routing**: Route Configuration, Navigation, Parameterised Routes, Nested Routes, Router Guards	**10**

Course Outcome	
Learner will achieve competency in	
CO1	Develop skills in working with streams, pipes, the file system, networking, and security in Node. Understand how to create command-line utilities and compress/decompress data
CO2	Gain proficiency in working with the Node environment, including installing Node, using global objects, handling events, and utilizing Node modules and npm.
CO3	Acquire knowledge and practical experience in React, including understanding React elements, components, JSX, and state management with hooks.
CO4	Master the fundamentals of Angular, including TypeScript, Angular CLI, components, modules, directives, and lifecycle hooks.
CO5	Demonstrate proficiency in working with forms, HTTP, routing, and dependency injection in Angular. Understand how to handle data, errors, and implement route guards.

Books and References:					
Sr. No.	**Title**	**Author/s**	**Publisher**	**Edition**	**Year**
1.	Learning Node Moving to the Server Side	Shelley Powers	O'Reilly	Second	2016
2.	Node: Up and Running	Tom Hughes-Croucher and Mike Wilson	O'Reilly	First	2009
3.	The NodeJS Handbook	Flavio Copes	-	-	-
4.	Learning React Modern Patterns for Developing React Apps	Alex Banks, Eve Porcello	O'Reilly	Second	2020
5.	Angular From Theory To Practice	Asim Hussain	Code Craft	Third Edition	2016

B.Sc.(Information Technology)		Semester–V	
Course Name: MERN Practical		**Course Code: VSIT311**	
Periods per week (1 Period is 120 minutes)		2	
Credits		2	
		Hours	**Marks**
Evaluation System	**Practical Examination**	2	100
	External		

S

Course Objective
To make learner able to :
1. Backend Development such as Real-time Chat Application with Socket.IO, CRUD Operations with Express and MongoDB, File Upload Handling as well as Creating and Publishing a Node Module
2. Frontend Development (React) such as TODO List Application, User Form Creation and Validation as well as API Data Fetching and Rendering
3. Frontend Development (Angular) such as TODO List Application, User Form Creation and Validation, API Data Fetching and Rendering as well as User Data Creation and Update

Sr No	Details
1	Create Chat application by using Socket.IO
2	a. Perform CRUD operations by using Express with MongoDB
	b. Perform File (image/doc) Upload operation by using MongoDB
3	Creating and Publishing Your Own Node Module
4	a. Create TODO list by using react hooks with different components.
	b. Create and validate the user form in React.
5	Fetch the API data and render the list on React app.
6	In Angular create TODO list with different components.
7	Create and validate the user form in Angular (Reactive Form/Template driven Form).
8	Fetch the API data and render the list on Angular app.
9	In Angular create and update the user data by using API.

Course Outcome
Learner will achieve competency in
CO1   Backend Development such as Real-time Chat Application with Socket.IO, CRUD Operations with Express and MongoDB, File Upload Handling as well as Creating and Publishing a Node Module
CO2   Frontend Development (React) such as TODO List Application, User Form Creation and Validation as well as API Data Fetching and Rendering
CO3   Frontend Development (Angular) such as TODO List Application, User Form Creation and Validation, API Data Fetching and Rendering as well as User Data Creation and Update

SEMESTER VI

B. Sc. (Information Technology)		Semester – VI	
Course Name: DevOps		**Course Code: VSIT350**	
Periods per week (1 Period is 60 minutes)		2	
Credits		2	
		Hours	**Marks**
Evaluation System	**Theory Examination**	2	60
	Internal	--	40

Course Objective
To make learner aware of
1. Understand the fundamentals of DevOps, including its origins, practices,culture, and its role in accelerating agile development and overcoming technical challenges in the software industry.
2. Explore case studies and best practices in the build, test, deploy, and manage phases of DevOps, focusing on topics such as API management, agile testing, continuous delivery, and agile operations.
3. Learn practical approaches for adopting DevOps within an organization and developing a business case for a DevOps transformation, including defining customer segments, value propositions, revenue streams, and cost structures.
4. Understand the integration of DevOps with enterprise architecture, information security, ITservice management, and lean startup methodologies.

Unit	Details	Lectures
I	**DevOps: An Overview:**Origins, Roots, Practices, Culture. **DevOps in the Ascendency:** Accelerating Agile Practices in Today's Software Factory, Embracing DevOps in the Application Economy, DevOps as a Critical Requirement, Banking on DevOps Practices, DevOps: A Key Component of Business Agility, DevOps: A Practice for Champions. **ITImpasse:** A World of 'Wicked' Business Problems, The Emergence of Agile Development, Modern Application Architectures, Ending the Technical Impasse. **DevOps Foundations:** What Characterizes DevOps Culture?, Lean Thinking to Reduce Waste, DevOps Metrics.	**10**
II	**Build:** Case Study: IceMobile, From Little API Acorns Big Things Grow, API Management: Stakeholders and Requirements, APIs Are Products, Managing the API Lifecycle, API Management: Essential Integrations, Taking a Strategic Approach, Building an API Future, Faster. **Test:** Case Study: AutoTrader.com, Testing Times, Agile Testing Trifecta. **Deploy:** Case Study: Citrix, Obstacles to Continuous Delivery, Continuous Delivery Maturity, Accelerating Maturity: Three Ways, Essential Tool chain	**10**

	Integrations, Release Automation: Capability Checklist, Recommendations and Action Plan. **Manage:** CaseStudy: ANZBank, MoreChange, MoreComplexity, New IT Operations Imperatives, Toward Agile Operations.	
III	**Adopting DevOps:** Developing the Playbook. **Developing a Business Case for a DevOps Transformation:** Developing The Business Case, Completing The Business Model Canvas, Customer Segments, Value Propositions, Channels, Customer Relationships, Revenue Streams, Key Resources, Key Activities, Key Partnerships, Cost Structures. **DevOps Plays for Optimizing the Delivery Pipeline:** DevOps as an Optimization Exercise, Core Themes, The DevOps Plays, Specializing Core Plays. **Practical DevOps:** DevOps and Enterprise Architecture, DevOps and Information Security, DevOps and IT Service Management, DevOps and Lean Startup. **DevOps Plays for Driving Innovation:** Optimize to Innovate, The Uber Syndrome, Innovation and the Role of Technology, Core Themes, Play: Build a DevOps Platform, Play: Deliver Microservices Architectures, Play: Develop an API Economy, Play: Organizing for Innovation.	**10**

Course Outcome	
Learner will be able to	
CO1	Develop a foundational understanding of DevOps principles,lean thinking,and metrics to drive efficiency and effectiveness in software development and Deployment processes.
CO2	Apply the knowledge gained to develop strategies and techniques for efficient and effective build, test, deploy, and management processes in a DevOps environment, leveraging API management, continuous delivery practices, and Agile operations principles.
CO3	Develop a comprehensive play book for DevOps adoption, including a business case for transformation, leveraging core DevOps plays, and optimizing the Delivery pipeline for improved efficiency and productivity.
CO4	Implement practical DevOps strategies for enterprise architecture, information security, IT service management, and lean startup initiatives, while effectively Measuring the impact and return on investment of DevOps automation.

Books and References:

Sr. No.	Title	Author/s	Publisher	Edition	Year
1.	DevOps for Digital Leaders	Aruna Ravichandran Kieran Taylor Peter Waterhouse	A press		2016
2.	The DevOps Adoption Playbook	Sanjeev Sharma	Wiley, IBM Press		2017

B.Sc.(Information Technology)		Semester–VI	
Course Name: Research Project		Course Code: VSIT361	
Periods per week(1Period is 120 minutes)		2	
Credits		2	
		Hours	Marks
Evaluation System	Practical Examination	2	100
	Internal	--	--

The details are given in Appendix – I

B. Sc. (Information Technology)		Semester – VI	
Course Name: Security in Computing		Course Code: VSIT351	
Periods per week (1 Period is 60 minutes)		2	
Credits		2	
		Hours	Marks
Evaluation System	Theory Examination	2	60
	Internal	--	40

Course Objective
To make learner understand and use
1. Basics of security, risk analysis, secure design principles.
2. Concept of encryption, authentication and authorization, database security.
3. Concept of secure network design, firewalls, wireless network security.
4. Concept of IDPS, VOIP, operating system security models.
5.Concept of secure application design, physical security and VM and cloud computing.

Unit	Details	Lectures
I	**Information Security Overview**: The Importance of Information Protection, The Evolution of Information Security, Justifying Security Investment, Security Methodology, How to Build a Security Program, The Impossible Job, The Weakest Link, Strategy and Tactics, Business Processes vs. Technical Controls. **Risk Analysis** : Threat Definition, Types of Attacks, Risk Analysis. **Secure Design Principles**: The CIA Triad and Other Models, Defense Models, Zones of Trust, Best Practices for Network Defense. **Authentication and Authorization**: Authentication, Authorization **Encryption**: A Brief History of Encryption, Symmetric-Key Cryptography, Public Key Cryptography, Public Key Infrastructure.	10
II	**Storage Security**: Storage Security Evolution, Modern Storage Security, Risk Remediation, Best Practices. **Database Security**: General Database Security Concepts, Understanding Database Security Layers, Understanding Database- Level Security, Using Application Security, Database Backup and Recovery, Keeping Your Servers Up to Date, Database Auditing an Monitoring. **Secure Network Design**: Introduction to Secure Network Design, Performance, Availability, Security. **Network Device Security**: Switch and Router Basics, Network Hardening.	10

	Firewalls: Overview, The Evolution of Firewalls, Core Firewall Functions, Additional Firewall Capabilities, Firewall Design.	
III	**Intrusion Detection and Prevention Systems**: IDS Concepts, IDS Types and Detection Models, IDS Features, IDS Deployment Considerations, Security Information and Event Management (SIEM). **Operating System Security Models**: Operating System Models, Classic Security Models, Reference Monitor, Trustworthy Computing, International Standards for Operating System Security. **Virtual Machines and Cloud Computing**: Virtual Machines, Cloud Computing. **Secure Application Design**: Secure Development Lifecycle, Application Security Practices, Web Application Security, Client Application Security, Remote Administration Security. **Physical Security**: Classification of Assets, Physical Vulnerability Assessment, Choosing Site Location for Security, Securing Assets: Locks and Entry Controls, Physical Intrusion Detection.	**10**

Course Outcome	
Learner will be able to	
CO1	Understand the concepts of information security, security Design principles. Application design process.
CO2	Understand and implement various concepts like database security, storage security ,Secure Network design, network security design ,encryption, firewall etc.
CO3	Understand and implement the concept of IDS and IPS, Operating System Security Models, Virtual Machines and Cloud Computing, Virtual Machines and Cloud Computing, Secure Application Design, Physical Security

Books and References:					
Sr. No.	**Title**	**Author/s**	**Publisher**	**Edition**	**Year**
1.	The Complete Reference: Information Security	Mark Rhodes-Ousley	McGraw-Hill	2nd	2013
2.	Essential Cyber security Science	Josiah Dykstra	O"Reilly	Fifth	2017
3.	Principles of Computer Security: CompTIA Security+ and Beyond	Wm. Arthur Conklin, Greg White	McGraw Hill	Second	2010

B. Sc. (Information Technology)		Semester – VI	
Course Name: Security in Computing Practical		**Course Code: VSIT352**	
Periods per week (1 Period is 120 minutes)		2	
Credits		2	
		Hours	**Marks**
Evaluation System	**Practical Examination**	3	100
	Internal	--	--

Course Objective
To make learner aware of
1. Understand and apply various security measures, including both AAA (Authentication, Authorization, and Accounting) and Layer 2 security protocols, to safeguard network infrastructure.
2. Learn to create and implement Extended Access Control Lists (ACLs) to manage and restrict network traffic effectively, both for IPv4 and IPv6.
3. Learn to configure Zone-Based Policy Firewalls to manage traffic flow and protect resources on the network.
4. Explore the implementation and modification of IOS-based IPS to detect and mitigate potential security threats on the network.
5   Understand the security implications of VLANs and how to secure them to prevent vulnerabilities related to Layer 2 devices.

Sr No	Details
1	**Configure Routers**
a	OSPF MD5 authentication.
b	NTP.
c	To log messages to the syslog server.
d	to support SSH connections.
2	**Configure AAA Authentication**
a	Configure a local user account on Router and configure authenticate on the console and Vty lines using local AAA
b	Verify local AAA authentication from the Router console and the PC-A client
3	**Configuring Extended ACLs**
a	**Configure, Apply and Verify an Extended Numbered ACL**
4	**Configure IP ACLs to Mitigate Attacks and IPV6 ACLs**
a	Verify connectivity among devices before firewall configuration.

b	Use ACLs to ensure remote access to the routers is available only from Management station PC-C.
c	Configure ACLs on to mitigate attacks.
d	Configuring IPv6 ACLs
5	**Configuring a Zone-Based Policy Firewall**
6	Configure IOS Intrusion Prevention System(IPS)Using the CLI
a	Enable IOS IPS.
b	Modify an IPS signature.
7	**Layer2 Security**
a	Assign the Central switch as the root bridge.
b	Secure spanning-tree parameters to prevent STP manipulation attacks.

Course Outcome	
Learner will be able to	
CO1	Successfully perform configuration tasks on routers, including OSPF MD5 authentication, SSH access, NTP, and logging to syslog servers.
CO2	Demonstrate the ability to create, apply, and verify Extended Numbered ACLs to control traffic flow on a network.
CO3	Deploy ACLs to manage access and mitigate potential attacks, applying both IPv4 and IPv6 configurations.
CO4	Implement an IOS IPS to monitor and prevent intrusion attempts, as well as modify existing IPS signatures based on organizational needs.
CO5	Assign root bridges and apply relevant security measures to protect against bridging manipulation and CAM table overflow attacks.

B.Sc.(Information Technology)		Semester–VI	
Course Name: Business Intelligence		**Course Code: VSIT353**	
Periods per week (1 Period is 60 minutes)		4	
Credits		2	
		Hours	**Marks**
Evaluation System	**Theory Examination**	2	60
	Internal	--	40

Course Objective
To make learner understand and use
1. Concept of BI and Decision Support System
2. Mathematical models, data mining.
3. Data mining algorithms (Classification/Clustering).
4. Knowledge management and expert system.

Unit	Details	Lectures
I	**Business intelligence:** Effective and timely decisions, Data, information and knowledge, The role of mathematical models, Business intelligence architectures, Ethics and business intelligence **Decision support systems:** Definition of system, Representation of the decision-making process, Evolution of information systems, Definition of decision support system, Development of a decision support system **Mathematical models for decision making:** Structure of mathematical models, Development of a model, Classes of models	10
II	**Data mining:** Definition of data mining, Representation of input data, Data mining process, Analysis methodologies **Data preparation**: Data validation, Data transformation, Data reduction **Classification**: Classification problems, Evaluation of classification models, Bayesian methods, Logistic regression, Neural networks, Support vector machines **Clustering:** Clustering methods, Partition methods, Hierarchical methods, Evaluation of clustering models	10
III	**Knowledge Management:** Introduction to Knowledge Management, Organizational Learning and Transformation, Knowledge Management Activities, Approaches to Knowledge Management, Information Technology (IT) in Knowledge Management, Knowledge Management Systems Implementation, Roles of People in Knowledge Management. **Expert Systems:** Basic Concepts of Expert Systems, Applications of Expert Systems, Structure of Expert Systems, Knowledge Engineering, Development of Expert Systems	10

Course Outcome	
Learner will be able to	
CO1	Understand the basic concept of Business Intelligence, Decision Support System and mathematical models for decision making.
CO2	Understand concept of data mining, Identify, organize and prepare data for mining purpose as well as implement various data mining algorithm.
CO3	Understand various concepts related to BI like knowledge management, expert system.

Books and References:					
Sr. No.	**Title**	**Author/s**	**Publisher**	**Edition**	**Year**
1.	Business Intelligence: Data Mining and Optimization for Decision Making	Carlo Vercellis	Wiley	First	2009
2.	Decision support and Business Intelligence Systems	Efraim Turban, Ramesh Sharda, Dursun Delen	Pearson	Ninth	2011
3.	Fundamental of Business Intelligence	GrossmannW, Rinderle-Ma	Springer	First	2015

B.Sc.(Information Technology)		Semester–VI	
Course Name: Business Intelligence Practical		**Course Code: VSIT354**	
Periods per week (1 Period is 120 minutes)		2	
Credits		2	
		Hours	**Marks**
Evaluation System	**Practical Examination**	3	100
	Internal	--	-

Course Objective
To make learner understand and use
1. Understand and apply the ETL process for data integration.
2. Build and manage data warehouses using SQL Server.
3. Create and manipulate OLAP cubes for multidimensional analysis
4. Analyze and visualize data using tools like Excel, Pivot Tables, and Charts.
5. Learn how to interpret results from analytical models and perform what-if analysis for decision-making.
6. Implement machine learning algorithms such as classification, clustering, linear regression, and logistic regression on real-world data.

Sr No	Details
1	Import the legacy data from different sources such as (Excel, SqlServer, Oracle etc.) and load in the target system. (You can download sample database such as Adventureworks, Northwind, foodmart etc.)
2	Perform the Extraction Transformation and Loading (ETL) process to construct the database in the Sqlserver.
3	a. Create the Data staging area for the selected database. b. Create the cube with suitable dimension and fact tables based on ROLAP, MOLAP and HOLAP model.
4	a. Create the ETL map and setup the schedule for execution. b. Execute the MDX queries to extract the data from the data warehouse.
5	a. Import the data warehouse data in Microsoft Excel and create the Pivot table and Pivot Chart.
	b. Import the cube in Microsoft Excel and create the Pivot table and Pivot Chart to perform data analysis.
6	Apply the what –if Analysis for data visualization. Design and generate necessary reports based on the data warehouse data.
7	Perform the data classification using classification algorithm.
8	Perform the data clustering using clustering algorithm.
9	Perform the Linear regression on the given data warehouse data.
10	Perform the logistic regression on the given data warehouse data.

The BI tools such as Tableau / Power BI / BIRT / R /Excel or any other can be used.

Course Outcome
Learner will be able to
CO1 Perform ETL process for any system with different data sources as well as setup the schedule for execution.
CO2 Perform data analysis using Excel and MDX.
CO3 Use what-if analysis and reporting tools for data visualization and decision-making.
CO4 Apply machine learning algorithms for data classification, clustering, and regression.

B.Sc.(Information Technology)		Semester–VI	
Course Name: Cloud Computing		**Course Code: VSIT355**	
Periods per week (1Period is 60 minutes)		4	
Credits		2	
		Hours	**Marks**
Evaluation System	**Theory Examination**	2	60
	Internal	--	40

<table>
<tr><td colspan="2">Course Objective</td></tr>
<tr><td colspan="2">To make learner understand and use</td></tr>
<tr><td>1.</td><td>To learn how to use Cloud Services.</td></tr>
<tr><td>2.</td><td>To implement Virtualization, Task Scheduling algorithms.</td></tr>
<tr><td>3.</td><td>Apply Map-Reduce concept to applications.</td></tr>
<tr><td>4.</td><td>To build Private Cloud.</td></tr>
<tr><td>5.</td><td>Broadly educate to know the impact of engineering on legal and societal issues involved.</td></tr>
</table>

Unit	Details	Lectures
I	**Introduction to Cloud Computing:** Introduction, Historical developments, Building Cloud Computing Environments. **Principles of Parallel and Distributed Computing:** Eras of Computing, Parallel v/s distributed computing, Elements of Parallel Computing, Elements of distributed computing, Technologies for distributed computing. **Virtualization:** Introduction, Characteristics of virtualized environments, Taxonomy of virtualization techniques, Virtualization and cloud computing, Pros and cons of virtualization, Technology examples. Logical Network Perimeter, Virtual Server, Cloud Storage Device, Cloud usage monitor, Resource replication, Ready-made environment.	10
II	**Cloud Computing Architecture:** Introduction, Fundamental concepts and models, Roles and boundaries, Cloud Characteristics, Cloud Delivery models, Cloud Deployment models, Economics of the cloud, Open challenges. **Fundamental Cloud Security:** Basics, Threat agents, Cloud security threats, additional considerations. **Industrial Platforms and New Developments:** Amazon Web Services, Google App Engine, Microsoft Azure.	10

III	**Specialized Cloud Mechanisms:** Automated Scaling listener, Load Balancer, SLA monitor, Pay-per-use monitor, Audit monitor, fail over system, Hypervisor, Resource Centre, Multi device broker, State Management Database. **Cloud Management Mechanisms:** Remote administration system, Resource Management System, SLA Management System, Billing Management System, **Cloud Security Mechanisms:** Encryption, Hashing, Digital Signature, Public Key Infrastructure (PKI), Identity and Access Management (IAM), Single Sign-On(SSO),Cloud-Based Security Groups, Hardened Virtual Server Images		**10**

Course Outcome	
Learner will be able to	
CO1	Analyze the Cloud computing setup with its vulnerabilities and applications using Different architectures.
CO2	Design different work flows according to requirements and apply map reduce Programming model.
CO3	Apply and design suitable Virtualization concept , Cloud Resource Management and Design scheduling algorithms.

Books and References:					
Sr. No.	Title	Author/s	Publisher	Edition	Year
1.	Mastering Cloud Computing Foundations and Applications Programming	Rajkumar Buyya, Christian Vecchiola,S. Thamarai Selvi	Elsevier	-	201 3
2.	Cloud Computing Concepts, Technology &Architecture	Thomas Erl, Zaigham Mahmood, and Ricardo Puttini	Prentice Hall	-	2013

B.Sc.(Information Technology)		Semester–VI	
Course Name: Cloud Computing Practical		**Course Code: VSIT356**	
Periods per week(1Periodis 120 minutes)		2	
Credits		2	
		Hours	**Marks**
Evaluation System	Practical Examination	3	100
	Internal	--	--

Course Objective	
To make learner understand and use	
1. To learn how to use Cloud Services.	
2. To implement our own machine.	
3. To create a server.	
4. To build Private Cloud.	

Sr No	Topic
1	Implement Windows/ Linux Cluster.
2	Developing application for windows Azure.
3	Implementing private cloud with Xen Server.
4	Implementing operating system on XenServer.
5	Implement Hadoop.
6	Develop application using GAE.
7	Implement VMWAREESXI server.
8	Native virtualization using HyperV.
9	Implementing operating system on HyperV.
10	Using Open Nebula to manage heterogeneous distributed datacenter infrastructure.

Course Outcome	
Learner will be able to	
CO1	To Develop Cloud Services.
CO2	Develop our own machine.
CO3	Develop and implement their own cloud and private server.

B.Sc.(Information Technology)		Semester–VI	
Course Name: Virtual & Augmented Reality		Course Code: VSIT357	
Periods per week(1 Period is 60 minutes)		4	
Credits		2	
		Hours	Marks
Evaluation System	Theory Examination	2	60
	Internal	--	40

Course Objective
To make learner understand and use
1. Understand the fundamental concepts and principles of Virtual Reality (VR), including its history , different realities , immersion , presence ,and perceptual models.
2. Explore the perceptual aspects of VR , including perception of space and time , perceptual stability , attention , and action . Understand the hardware challenges and Adverse health effects associated with VR.
3. Develop skills in VR content creation, including environmental design, influencing user behavior, and interaction design. Explore human- centered interaction principles And various input devices for VR.
4. Examine the intersection of design and art in digital realities, including optimizing Dart for VR, understanding computer vision in augmented reality, and exploring cross-platform theories.
5. Explore advanced topics and tools in VR , including open-source frameworks , data visualization in spatial computing, character AI and behaviors , and the healthcare technology ecosystem in VR and AR.

Unit	Details	Lectures
I	Introduction: What Is Virtual Reality, A History of VR, An Overview of Various Realities, Immersion, Presence, and Reality Trade-Offs, The Basics: Design Guidelines, Objective and Subjective Reality , Perceptual Models and Processes , Perceptual Modalities.	10
II	Perception of Space and Time, Perceptual Stability, Attention, and Action, Perception: Design Guidelines, Adverse Health Effects, Motion Sickness, Eye Strain, Seizures, and Aftereffects, Hardware Challenges , Latency , Measuring Sickness , Reducing Adverse Effects , Adverse Health Effects : Design Guidelines.	10
III	Content Creation, Concepts of Content Creation, Environmental Design , Affecting Behavior , Transitioning to VR Content Creation, Content Creation: Design Guidelines , Interaction , Human - Centered Interaction , VR Interaction Concepts , Input Devices , Interaction Patterns and Techniques , Interaction : Design Guidelines.	10

Course Outcome	
Learner will be able to	
CO1	Understand various concepts of networking.
CO2	Understand the working of various protocols like IP,DHCPetc.
CO3	Configure different protocols.
CO4	Implement various concepts like routing using CISCO packet tracer.
CO5	Implement appropriate network protocol for real life situation.

Books and References:

Sr. No.	Title	Author/s	Publisher	Edition	Year
1.	The VR Book, Human Centered Design for Virtual Reality	Jason Jerald	ACM Books	1st	2016
2.	Creating Augmented and Virtual Realities	Erin Pangilinan, Steve Lukas,Vasanth Mohan	O'Reilly	1st	2019
3.	VirtualrealitywithVRTK4	Rakesh Baruah	APress	1st	2020

B.Sc.(Information Technology)		Semester–VI	
Course Name: Virtual & Augmented Reality Practical		Course Code: VSIT358	
Periods per week(1 Period is 60 minutes)		4	
Credits		2	
		Hours	Marks
Evaluation System	Theory Examination	2	60
	Internal	--	40

Course Objective
To make learner understand and use
1. Introduce students to the fundamental principles of 3D modeling, including the use of geometric primitives such as cubes, spheres, and cylinders for creating simple 3D environments.
2. Teach students how to design basic 3D architecture for virtual spaces, such as simple houses or rooms, using 3D modeling tools like Blender, SketchUp, or Unity.
3. Enable students to develop basic interactive applications and games using 3D models, with focus on physics, collision detection, and game mechanics, using platforms like Unity or Unreal Engine.

Sr No	Details
1.	Build an environment using primitives like boxes, spheres and cylinders
2.	Develop a simple architecture of a house
3.	Develop a simple architecture of a house
4.	Develop a simple game application using 3D modeling and rendering Techniques
5.	Develop a concept of simple virtual classroom
6.	Show the simulation of a plant growing using augmented reality
7.	Show the animation of a fish swimming using augmented reality
8.	Simulate a basket ball court using augmented reality
9.	Show the simulation of a moving car using augmented reality
10.	Develop an application in any domain of interest e.g, Gaming, Human Pose etc

Course Outcome	
Learner will be able to	
CO1	Demonstrate proficiency in creating 3D models of environments, objects, and architecture using tools such as Blender, SketchUp, or Unity. They will be able to design simple structures like houses and rooms.
CO2	Design and develop interactive AR applications that overlay virtual objects (like a growing plant, fish swimming, or moving vehicles) in the real world using ARKit, ARCore, and Unity's AR Foundation.
CO3	Apply 3D and AR technologies to develop virtual classrooms and educational simulations, enhancing learning experiences through immersive environments

B. Sc. (Information Technology)		Semester – VI	
Course Name: Exploratory Data Analysis		**Course Code: VSIT359**	
Periods per week (1 Period is 60 minutes)		2	
Credits		2	
		Hours	**Marks**
Evaluation System	**Theory Examination**	2	60
	Internal	--	40

<table>
<tr><td colspan="2">Course Objective</td></tr>
<tr><td colspan="2">To make learner aware of</td></tr>
<tr><td colspan="2">1. To understand importance of data and its types in Exploratory Data Analysis.</td></tr>
<tr><td colspan="2">2. To understand difference between EDA and summary statistics in context of interpretation.</td></tr>
<tr><td colspan="2">3. To understand the importance of data pre-processing for Exploratory Data Analysis.</td></tr>
<tr><td colspan="2">4. To understand the significance of missing value imputations in better EDA interpretations.</td></tr>
<tr><td colspan="2">5. To understand the importance measure of central tendency in describing the quick view of data set.</td></tr>
<tr><td colspan="2">6. To understand the importance of measure of dispersion and its interpretation in spreadness of data.</td></tr>
</table>

Unit	Details	Lectures
I	**INTRODUCTION TO DATA AND ITS TYPES** : Definition and importance of data, classification of data : based on observation – Cross Sectional, times series and panel data, based on measurement – ratio, interval, ordinal and nominal, based on availability – primary, secondary, tertiary, based on structural form – structured, semi structured and unstructured, based on inherent nature – quantitative and qualitative, concepts on sample data and population, small sample and large sample, statistic and parameter, types of statistics and its application in different business scenarios, frequency distribution of data. **INTRODUCTION TO EXPLORATORY DATA ANALYSIS (EDA)** : Definition of EDA, difference between EDA with classical and Bayesian Analysis, comparison of EDA with Classical data summary measures, goals of EDA, Underlying assumptions in EDA, importance of EDA in data exploration techniques, introduction to different techniques to test the assumptions involved in EDA, role of graphics in data exploration, introduction to unidimensional, bidimensional and multidimensional graphical representation of data	10
II	**DATA PREPARATION**: Introduction to data exploration process for data preparation, data discovery, issues related with data access, characterization of data, consistency and pollution of data, duplicate or redundant variables,	10

	outliers and leverage data, noisy data, missing values, imputation of missing and empty places, with different techniques, missing pattern and its importance, handling non numerical data in missing places. **UNIVARIATE DATA ANALYSIS**: Description and summary of data set, measure of central tendency – mean: Arithmetic, geometric and harmonic mean – Raw and grouped data, confidence limit of mean, median, mode, quartile and percentile, interpretation of quartile and percentile values, measure of dispersion, concepts on error, range, variance, standard deviation, confidence limit of variance and standard deviation, coefficient of variation, mean absolute deviation, mean deviation, quartile deviation, interquartile range, concepts on symmetry of data, skewness and kurtosis, robustness of parameters, measures of concentration.	
III	**BIVARIATE DATA ANALYSIS**: Introduction to bivariate distributions, association between two nominal variables, contingency tables, Chi-Square calculations, Phi Coefficient, scatter plot and its causal interpretations, correlation coefficient, regression coefficient, relationship between two ordinal variables – Spearman Rank correlation, Kendall's Tau Coefficients, measuring association between mixed combination of numerical, ordinal and nominal variables.	**10**

Course Outcome
Upon the successful completion of this course, students will be able to:
CO1 Understand importance of data and its types in Exploratory Data Analysis as well as classify EDA and summary statistics in context of interpretation.
CO2 Understand the significance of missing value imputations in better EDA interpretations as well as Analyse the measure of central tendency in describing the quick view of data set..
CO3 Categorize measure of dispersion and its interpretation in spread ness of data.

Books and References:

Sr. No.	Title	Author/s	Publisher	Edition	Year
1.	Exploratory Data Analysis	John W Tukey	Addison Wesley	1st	1977
2.	Exploratory Data Analysis in Business and Economics	Thomas Cleff	Springer	1st	2014
3.	Graphical Exploratory Data Analysis	S.H.C. du Toit A.G.W. Steyn R.H. Stumpf	Springer	1st	1986
4.	Hand book of Data Visualization	Chun-houh Chen, Wolfgang Härdle	Springer	1st	2008

B. Sc. (Information Technology)		Semester – VI	
Course Name: Exploratory Data Analysis Practical		Course Code: VSIT360	
Periods per week (1 Period is 120 minutes)		2	
Credits		2	
		Hours	Marks
Evaluation System	Practical Examination	3	100
	Internal	--	--

<table>
<tr><td colspan="2">Course Objective</td></tr>
<tr><td colspan="2">To make learner aware of</td></tr>
<tr><td colspan="2">1. Understand the data and its types for the appropriate exploratory data analysis.</td></tr>
<tr><td colspan="2">2. Understand the importance of Exploratory Data Analysis over summary statistics.</td></tr>
<tr><td colspan="2">3. Understand the importance Univariate statistics in EDA.</td></tr>
<tr><td colspan="2">4. Plot Univariate statistical graphs for the better representation and interpretation.</td></tr>
<tr><td colspan="2">5. Plot bivariate statistical graphs for the better representation and interpretation.</td></tr>
</table>

List of Practical :	
1	Managing Data Frames with the dplyr package
2	Use dplyr Grammar for inbuilt data set car.
3	Use group by(), %>%,mutate(), rename(),arrange(), filter(), select()
4	Use the data set air quality from inbuilt data sets library. Use summary statistics and find the important key values from the output Use boxplot and find the interquartile range. Interpret the boxplot and inner and outer fencing of outliers Check the missing value in the data set and fine the suitable solution for the missing values. Using histogram, find the distribution of data and give proper comment over the dataset.
5	Use bar plot and identify the difference between bar plot and histograms. Conclude the appropriate use of bar plot and histogram.
6	Explore the two dimensional data
7	Scatter plot between two variables
8	Five number summary in exploratory data analysis
9	Multiple histogram and multiple boxplots, Multiple scatter plots and colouring the graph
10	Lattice system in R environment and Graphical window in R and its uses

<table>
<tr><td colspan="2">Course Outcome</td></tr>
<tr><td colspan="2">Upon the successful completion of this course, students will be able to:</td></tr>
<tr><td>CO1</td><td>Experiment with exploratory data analysis; use its features in the field of data science.</td></tr>
<tr><td>CO2</td><td>Make use of data and its types for the appropriate exploratory data analysis.</td></tr>
<tr><td>CO3</td><td>Understand the importance of Exploratory Data Analysis over summary statistics.</td></tr>
<tr><td>CO4</td><td>Interpret and make use of Univariate statistics in EDA</td></tr>
<tr><td>CO5</td><td>Build Univariate statistical graphs for the better representation and interpretation.</td></tr>
<tr><td>CO6</td><td>Build bivariate statistical graphs for the better representation and interpretation.</td></tr>
</table>

Project Semester V / VI

- 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 factfinding.
- 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.

- 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), TcL/TK,
.NET Platform	F#,C#. Net, Visual C#. Net, ASP.Net

MIDDLEWARE(COMPONENT) TECHNOLOGIES	COM/DCOM, Active-X, EJB
UNIX INTERNALS	Device Drivers, RPC, Threads, Socket programming
NETWORK/WI RELESS TECHNOLOGIES	-
REALTIME OPERATING SYSTEM/ EMBEDDED SKILLS	LINUX, Raspwerry 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. Background

    2. Objectives

3. Purpose, Scope, and Applicability

1. Purpose

2. Scope

3. Applicability

4. Achievements

5. Organisation of Report

CHAPTER 2: SURVEY OF TECHNOLOGIES AND / OR LITURATURE REVIEW

CHAPTER 3: REQUIREMENTS AND ANALYSIS

1. Problem Definition

2. Requirements Specification

3. Planning and Scheduling

4. Software and Hardware Requirements

5. Preliminary Product Description

6. Conceptual Models CHAPTER 4 : SYSTEM DESIGN

1. Basic Modules

2. Data Design

1. Schema Design

2. Data Integrity and Constraints

3. Procedural Design

1. Logic Diagrams

2. Data Structures

3. Algorithms Design

4. User interface design

5. Security Issues

6. Test Cases Design

The documentation should use tools like star UML, Visual 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

1. Implementation Approaches

2. Coding Details and Code Efficiency

3. Code Efficiency

4. Testing Approach

1. Unit Testing

2. Integrated Testing

3. Beta Testing

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 50

/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: Summarising 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 student's 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 student 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 with in 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 optimization.

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.

For example

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:10.1080/03640210701703740

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.

0. ISO/IEC 12207: Software Life Cycle Process (http://www.software.org/quagmire/descriptions/iso-iec12207.asp).

0. IEEE 1063: Software User Documentation(http://ieeexplore.ieee.org).

0. ISO/IEC: 18019: Guidelines for the Design and Preparation of User Documentation for Application Software.

0. http://www.sce.carleton.ca/squall.

0. http://en.tldp.org/HOWTO/Software-Release-Practice- HOWTO/documentation.html.

0. http://www.sei.cmu.edu/cmm/

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

TITLE OF THE PROJECT

(NOT EXCEEDING 2 LINES, 24 BOLD, ALLCAPS)

A Project Report (12Bold)

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(12bold)   **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 1.5line spacing and all paragraphs should start with 1 tab space.

ACKNOWLEDGEMENT
(20, BOLD, ALL CAPS, CENTERED)

The acknowledgement should be in timesnewroman, 12 font with 1.5 line spacing, justified.

(Declaration page format)

DECLARATION (20 bold, centered, allcaps)

Content (12, justified)

I here by 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 to be submitted as final semester project as part of our **BACHELOR OF SCIENCE (INFORMATION TECHNOLOGY)** curriculum.

Name and Signature of the Student

TABLE OF CONTENTS (20bold, caps, centered)

Should be generated automatically using word processing software.

2. Integration System (Test cases and Test Results) Chapter 5:

    Results and Discussions (Output Screens)

Chapter 6: Conclusion and Future Work Chapter 7:

    References

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.

(Project Introduction page format)

Chapter 1

Introduction (20 Bold, centered)

Content or text (12, justified)

Note: Introduction has to cover brief description of the project with minimum 4 pages.

Chapter 2

System Analysis (20 bold, Centered)

Subheadings are as shown below with following format (16 bold, CAPS)

1. Existing System (16Bold)

1. (14 bold, title case)

1. (12 bold, title case)

2.   Proposed System

3.   Requirement Analysis

4.   Hardware Requirements

5.   Software Requirements

6.   Justification of Platform – (how h/w & s/w satisfying the project) Table

2.1: Caption

Chapter 3
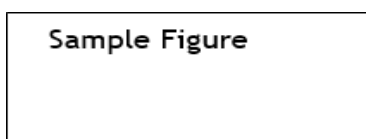
System Design (20 bold, centered)

Subheadings are as shown below with following format (16 bold, CAPS) Specify figures as

Fig 11.1 – caption

1.   Module Division

2.   Data Dictionary

3.   E-R Diagrams

4.   Data Flow Diagrams /UML

Note: write brief description at the bottom of all diagrams

Sample Figure

Fig. 3.1: Caption

Chapter 4

Implementation and Testing (20 bold, centered)

1. Code (Place Core segments)

Content includes description about coding phase in your project (Font-12) (*don't

include complete code just description)

0. Testing Approach

Subheadings are as shown below with following format (16 bold, CAPS)

1. Unit Testing

2. Integration Testing Note:

- Explain about above testing methods.

- Explain how the above techniques are applied in your project.

- Provide Test plans, test cases, etc. relevant to your project.

Chapter 5

Results and Discussions (20 bold, centered)

Note: Place Screen Shots and write the functionality of each screen at the bottom

Chapter 6

Conclusion and Future Work (20 bold, centered)

The conclusions can be summarized in a fairly short chapter around 500 words. Also include limitations of your system and future scope (12, justified)

Chapter 7

Chapter 7

References (20 bold, centered)

Content (12, LEFT)

1. Title of the book, Author

2. Full URL of online references [3]

*<u>NOTE ABOUT PROJECT VIVA VOCE:</u>

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.

The plagiarism should be maintained as per the UGC guidelines.

Evaluation Scheme

Internal Evaluation (40 Marks).

Test: 1 Class test of 15 marks. (Can be taken online)

Q. No	Attempt _any three_ of the following:	15
a.		
b.		
c.		
d.		
e.		

15 marks project / presentation / Assignment

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

External Examination: (60marks)

	All questions are compulsory	
Q1	**(Based on Unit 1) Attempt _any four_ of the following:**	**20**
a.		
b.		
c.		
d.		
e.		
f.		
Q2	**(Based on Unit 2) Attempt _any two_ of the following:**	**20**
Q3	**(Based on Unit 3) Attempt _any two_ of the following:**	**20**

Practical Exam: 100 marks

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

1.	**Practical Question 1**	**40**
2.	**Practical Question 2**	**40**
3.	**Journal**	**10**
4.	**Viva Voce**	**10**

KET's V. G. Vaze College(Autonomous) Board of Studies – Information Technology comprised of following members

Mrs. Pournima P Bhangale, Chairperson

Mrs. Vandana Y Kadam, Member

Mrs. Rakhee D Rane, Member

Mrs. Nanda A Rupnar, Member

Ms. Mohini Bhole, Member

Ms. Pranali Pawar, Member

Dr. Hiren Dand , VC nominee

Professor (Dr.) Ajay S Patil, Subject Expert, North Maharashtra University

Mr. Milind Narayan Kolambe, Subject Expert, Pune University

Mr. Tejpal Khachane, Industry Expert

Mr. Abhishek Ghorpade, Postgraduate meritorious alumnus

Mrs. Pournima P Bhangale   Dr. Hiren Dand
Chairperson   VC Nominee