

NATIONAL INSTITUTE OF ELECTRONICS AND INFORMATION TECHNOLOGY

DEEMED TO BE UNIVERSITY UNDER DISTINCT CATEGORY
(AN AUTONOMOUS INSTITUTION UNDER MINISTRY OF ELECTRONICS & IT, GOVT. OF INDIA)

Curriculum and Syllabi for Post Graduate Diploma - Computer Applications (PGDCA) (2025-26 Scheme)



Main Campus at Ropar and Constituent Units at Aizawl, Agartala, Aurangabad, Calicut, Gorakhpur, Imphal, Itanagar, Kekri, Kohima, Patna and Srinagar

Vision

To equip students with strong theoretical knowledge and hands-on skills in computer applications, preparing them for the evolving needs of the IT and business world. It aims to create competent professionals who can efficiently design, develop, and manage software systems, as well as perform essential office automation tasks such as data handling, document processing, presentations, and spreadsheet analysis. By integrating practical training in tools like MS Office, Google Workspace, and other automation software, the programme enables students to streamline everyday business operations.

Mission

To provide quality education in computer science and applications, with a focus on developing technical proficiency, problem-solving abilities, and practical skills relevant to the IT industry. It aims to bridge the gap between academic knowledge and industry requirements by offering a curriculum that includes programming, database management, web development, and office automation tools. The program is committed to fostering a learning environment that encourages innovation, ethical practices, and continuous improvement. Through hands-on training and real-world projects, it prepares students to take on professional roles in software development, IT services, and office management, contributing effectively to organizational and societal development.

Program Education Objectives

The program objective is to provide a strong foundation in programming, software development, database management, web technologies, office automation, and data analysis. The program also introduces students to the basics of cybersecurity, helping them understand how to protect data, systems, and networks from potential threats. Along with technical skills, it focuses on developing the ability to solve real-world problems, support business operations, and improve workplace productivity. It also emphasizes communication skills, teamwork, ethical values, and continuous learning. A key objective is to support the overall personality development of students by enhancing their confidence, professional behavior, and adaptability to the dynamic demands of the IT industry.

Program Outcomes

The PGDCA program is designed to help students build strong computer knowledge along with practical skills needed in today's job market. By the end of the course, students will be able to develop software applications, manage databases, design websites, and analyze data using spreadsheets. They will also learn to use office automation tools like MS Word, Excel, and PowerPoint to make everyday office tasks easier and more efficient. In addition to technical training, the programme also focuses on personality development by improving communication skills, confidence, teamwork, and professional behavior. It encourages students to think critically, solve problems effectively, and stay updated with new technologies, preparing them for successful careers in IT, software development, and office management roles.

Program Specific Outcomes

By the end of the programme, students will be able to develop and manage software applications using various programming languages and development tools. They will gain practical proficiency in office automation, including the use of word processors, spreadsheets, and presentation software to streamline routine tasks and improve efficiency. A key outcome is the ability to perform data analysis using spreadsheets, enabling students to organize, visualize, and interpret data for better decision-making in business environments. They will also be skilled in designing interactive websites, managing databases, and providing IT support through system maintenance and troubleshooting, making them well-rounded professionals ready to contribute to digital workplaces.

CATEGORY WISE CREDIT DISTRIBUTION

Category	Category Code	Credits
Skill / Employment Enhancement Courses (Project/Internship/Seminar/Dissertation/Research)	EE	4
Audit Courses (without grade or credit)	AU	0
Open Elective Courses	OE	4
Professional Elective Courses	PE	12
Program Core Courses	PC	20
Total Credits (Common track)		40

EVALUATION AND ASSESSMENT

1. Performance of a student in a semester shall be evaluated through continuous Class assessment, Tutorial/Lab assessment, Mid-Semester Examination (MSE) and End-Semester Examination (ESE). Both the MSE and ESE shall be the University examination and will be conducted as notified by the Controller of Examinations (CoE) of the University.
2. The continuous assessment shall be based on assignments, tutorials, paper presentation / quizzes/ viva-voce / flipped classes, lab work / projects / fieldwork and attendance, etc.
3. The MSE/ESE shall be comprising of written papers.
4. The overall assessment of the students will be done as per the following scheme:

S. No.	Assessment Type	Marks Weightage
1.	Mid Semester Examination	20
2.	End Semester Examination	40
3.	Continuous Assessment - Practical /Lab/ Tutorial	25
4.	Continuous Assessment - Theory	15
Total		100

For more details, please refer the applicable ordinance for this programme and Assessment SOPs.

CURRICULUM

Semester 1

Semester Code	Course Code	Course Title	L	T	P	Cr
PC-PGDCAP-101	DCSA250060	Office Automation Tools	2	0	4	4
PC-PGDCAP-102	DCSE250038	Computer Organization	3	1	0	4
PC-PGDCAP-103	DCSE250079	Fundamentals of Programming in Python	2	0	4	4
PE-PGDCAP-104	Elective	Program Elective				4
OE-PGDCAP-105	Elective	Open Elective				4
AU-PGDCAP-106	Elective	Audit Elective				0

Semester 2

Semester Code	Course Code	Course Title	L	T	P	Cr
PC-PGDCAP-201	DCSA250084	Web Technologies	3	0	2	4
PC-PGDCAP-202	DCSE250041	Fundamentals of Database Systems	3	0	2	4
PE-PGDCAP-203	Elective	Program Elective				4
PE-PGDCAP-204	Elective	Program Elective				4
EE-PGDCAP-205	EE	Project				4

Elective Courses

Elective Courses for PE Category

Course Code	Course Title	L	T	P	Cr	For Sem./Code
DCSA250031	Full Stack Web Development	3	0	2	4	
DCSE250020	Cloud Computing	3	0	2	4	
DCSE250037	Computer Networks and Security	3	0	2	4	
DCSE250044	Cyber Security	3	0	2	4	
DOAI250005	Artificial Intelligence and Machine Learning	2	0	4	4	
DOAI250026	Data Analysis Using Spreadsheet	3	0	2	4	

Elective Courses for OE Category

Students may opt courses under MOOC, NPTEL, SWAYAM, NSQF/NCVET aligned courses or other relevant technical subjects not included in their core curriculum. The exercise of option shall be subject to the Course Schedule of the respective Constituent Unit, the credit requirements and availability of seats. Guidelines for choosing MOOC/Online courses are given separately and shall have to be adhered to.

Elective Courses for AU (Audit) Category

Course Code	Course Title	L	T	P	Cr
DASH240027	Disaster Management	2	0	0	0
DASH240047	English for Research Paper Writing	2	0	0	0
DASH240052	Environmental Science	3	0	0	0
DASH240085	Pedagogy Studies	2	0	0	0
DASH240086	Personality Development through Life Enlightenment Skills	2	0	0	0
DASH240097	Sanskrit for Technical Knowledge	2	0	0	0
DASH240103	Stress Management by Yoga	2	0	0	0
DASH240104	Technical Presentation and Report Writing	0	0	2	0
DASH240106	Value Education	2	0	0	0
DASH240124	Constitution of India - AU	2	0	0	0
DASH250023	Constitution of India	2	1	0	0
DASH250027	Disaster Management	3	0	0	0
DASH250034	Energy and Environmental Engineering	3	0	0	0
DASH250047	English for Research Paper Writing	2	0	0	0
DASH250054	Essence of Indian Knowledge and Tradition	2	0	0	0
DASH250085	Pedagogy Studies	2	0	0	0
DASH250086	Personality Development	2	0	0	0
DASH250097	Sanskrit for Technical Knowledge	3	0	0	0
DASH250103	Stress Management by Yoga	3	0	0	0
DASH250106	Value Education	2	0	0	0
DASH250117	Innovative Thinking and Entrepreneurship Skills	1	0	0	0
DASH250118	Intellectual Property Rights	2	0	0	0

DCSA240080	Training on Computer Networking	0	0	2	0
DCSA240304	Lab Based on Statistical Package	0	0	2	0
Any other Scheduled Course as per the Course Schedule of the respective Constituent Unit can also be chosen, subject to availability of seats. However, there shall be no assessment and grading for the course chosen as Audit Course.					

The choice of all type of Electives available shall be as per the Course Schedule of the respective Constituent Unit.

Guidelines for Massive Open Online Courses (MOOC) / approved Online Courses

1. Relevant Swayam, MOOC, NPTEL, NIELIT NSQF and any other online courses approved by UGC/AICTE shall also be allowed as Open Electives(OE).
2. One credit of MOOC course should be minimum of 30 hours.
3. A student can choose any approved online course as Open Elective, subject to minimum credit requirements.
4. The students willing to opt OE under MOOC in their next semester, will submit their request to the MOOC Coordinator at their respective campus.
5. Once approved, the campus shall display the list of accepted courses.
6. The student has to submit certificate issued by the institution offering the MOOCs course, along with the number of credits and grades, to get credits transferred into his/her marks certificate issued by NDU.
7. The transfer of credits shall be as adopted by NDU.

Detailed Syllabus

Course Code	Course Name	L	T	P	Cr
DCSA250060	Office Automation Tools	2	0	4	4

Course Outcomes:

CO1: Explain the fundamentals of computers, their components, data representation, number systems, programming languages, and operating systems.

CO2: Use MS Word to create, edit, format, and manage documents with features such as tables, images, headers/footers, and mail merge.

CO3: Apply MS Excel tools for data entry, formatting, formulas, functions, charts, and data management using validation and filtering.

CO4: Design effective presentations using MS PowerPoint with layouts, multimedia, animations, transitions, and slide shows.

CO5: Operate Tally software for basic accounting tasks including company setup, ledger creation, voucher entry, inventory management, and generating financial reports with GST features.

Module 1:

Introduction to Computers: Definition, Characteristics, Applications, Types of Computers: Analog, Digital, Hybrid, Components of a Computer System: Input, Output, CPU, Memory, Data Representation: Bits, Bytes, Binary, Octal, Hexadecimal, Number Systems: Conversion and Arithmetic, Computer Languages: Machine, Assembly, High-Level Languages, Introduction to Operating Systems, Types of Operating Systems: Batch, Time Sharing, Real-time, Distributed, Functions of OS: Process Management, File Management, Memory Management, GUI vs. CLI, Overview of Windows and Linux operating systems, File and Folder Management in Windows.

Module 2:

Introduction to MS Word: Interface, Creating, Saving, and Opening Documents, Text Formatting: Font, Paragraph, Styles, Page Layout: Margins, Orientation, Page Breaks, Working with Tables, Images, Headers/Footers, Mail Merge, Spell Check, Grammar, and Thesaurus, Printing Documents.

Module 3:

Introduction to MS Excel: Interface, Workbook and Worksheet Management, Data Entry, Formatting, and Sorting, Basic Formulas and Functions, Charts: Column, Line, Pie, Cell Referencing: Relative, Absolute, Data Validation and Filtering.

Module 4:

Introduction to MS PowerPoint, Creating, Opening, and Saving Presentations, Slide Layouts and Design Templates, Working with Text, Images, SmartArt, Animation and Transition Effects, Creating and Using Slide Shows.

Module 5:

Overview of Tally and its Features, Company Creation and Configuration, Ledgers and Groups, Voucher Entry: Payment, Receipt, Contra, Journal, Inventory Management: Stock Groups, Items, Units Generating Reports: Balance Sheet, Profit & Loss, Trial Balance, GST Basics in Tally.

Labs / Practicals:

Part A: Hardware

1. Identification of the peripherals of a computer, components in a CPU and their functions.
2. Assembling and disassembling the system hardware components of personal computer.

3. Basic Computer Hardware Trouble shooting.
4. LAN and WiFi Basics.
5. Operating System Installation – Windows OS, UNIX/LINUX, Dual Booting.
6. Installation and Uninstallation of Software – Office Tools, Utility Software (like Anti-Virus, System Maintenance tools); Application Software - Like Photo/Image Editors, ...); Freeware, Shareware, Internet Browsers, Programming IDEs,
7. System Configuration – BIOS Settings, Registry Editor, MS Config, Task Manager, System Maintenance.

Part B: Software

1. Activities using Word Processor Software
2. Activities using Spreadsheets Software.
3. Activities using Presentation Software
4. Activities using Tally Software

References:

1. Basics of Information Technology-Sumita Arora,Dhanpat Rai & Co (Pvt.) Ltd.,New Delhi.
2. Foundation of Information Technology – D.S. Yadav, New Age International Publisher.
3. Fundamentals of Computers – V. Rajaraman / P. K. Sinha
4. Mastering Tally ERP 9 – Tally Solutions
5. Information Technology and Office Automation, R. Parameswaran, S Chand Publishing

Course Code	Course Name	L	T	P	Cr
DCSE250038	Computer Organization	3	1	0	4

Course Outcomes:

CO1: Explain number systems, data representation, computer evolution, and Von Neumann architecture.

CO2: Apply Boolean algebra and minimization techniques to design combinational and sequential logic circuits.

CO3: Describe processor organization, instruction execution, addressing modes, subroutines, and instruction pipelining.

CO4: Understand Assembly language and implement arithmetic algorithms for integer and floating-point operations.

CO5: Analyze input/output mechanisms, interrupts, DMA, and memory organization including cache, associative, and virtual memory.

Module 1:

Number System: Binary, Octal, Hexadecimal; Positive & negative numbers; Arithmetic operations: Addition, Subtraction. Evolution of computers, Stored program concept, Von Neumann Architecture. Information representation, ASCII & EBCDIC codes.

Module 2:

Building blocks: Combinational logic (gates, multiplexers, decoders, encoders), Sequential blocks (flip-flops, registers, counters, RAM), Boolean Algebra: Variables, functions, Canonical & standard forms, Truth table, Karnaugh map minimization. Combinational circuits: Multiplexer, Demultiplexer, Decoder, Encoder.

Module 3:

Processor Architecture: Simple computer organization, Instruction execution via microinstructions, Instruction formats, Addressing modes, Stacks, Subroutines, Interrupt handling, Instruction pipelining (stages, hazards, solutions).

Module 4:

Machine & Assembly language, Pseudo-operations, Subroutines, Interrupt & I/O programming. Arithmetic Algorithms: Addition/subtraction for sign magnitude & 2's complement numbers, Integer multiplication (shift & add, Booth's algorithm), Integer division, Floating point representations & arithmetic.

Module 5:

I/O Organization: Strobe-based & handshake-based communication, Vector & priority interrupts, DMA-based data transfer. Memory Organization: Classification, Static & dynamic RAM, Large memory construction, Associative memory, Cache memory, Virtual memory organization.

Labs / Practicals:

II

References:

1. Mano, M. (1992). Computer System Architecture. 3rd edition. Pearson Education.
2. Mano, M. (1995). Digital Design. Pearson Education Asia.
3. Null, L., & Lobur, J. (2018). The Essentials of Computer Organization and Architecture. 5th edition. (Reprint) Jones and Bartlett Learning.
4. Stallings, W. (2010). Computer Organization and Architecture Designing for Performance 8th edition. Prentice Hall of India.

Course Code	Course Name	L	T	P	Cr
DCSE250079	Fundamentals of Programming in Python	2	0	4	4

Course Outcomes:

CO1: Develop algorithmic solutions to solve simple and well-defined computational problems effectively.

CO2: Write and demonstrate Python programs using basic statements, expressions, and syntax constructs.

CO3: Explain and implement control flow and function concepts in Python to solve programming tasks.

CO4: Apply Python data structures, including lists, tuples, and dictionaries, for managing compound data.

CO5: Describe and use file handling, exceptions, modules, and packages in Python-based problem-solving.

Module 1:

Python Introduction, Technical Strength of Python, Introduction to Python Interpreter and program execution, Using Comments, Literals, Constants, Python's Built-in Data types, Numbers (Integers, Floats, Complex Numbers, Real, Sets), Strings (Slicing, Indexing, Concatenation, other operations on Strings), Accepting input from Console, printing statements, Simple 'Python' programs. Flow Chart Symbols, Basic algorithms/flowcharts for sequential processing, decision based processing and iterative processing.

Module 2:

Assignment statement, expressions, Arithmetic, Relational, Logical, Bitwise operators and their precedence, Conditional statements: if, if-else, if-elif-else; simple programs, Notion of iterative computation and control flow –range function, While Statement, For loop, break statement, Continue Statement, Pass statement, else, assert. Lists, tuples, and dictionaries—slicing, indexing, concatenation, and mutability. Operations such as finding max, min, mean, linear search, and frequency count using dictionaries.

Module 3:

Top-down approach of problem solving, Modular programming and functions, Function parameters, Local variables, the Return statement, DocStrings, global statement, Default argument values, keyword arguments, VarArgs parameters. Library function-input(), eval(), print(), Recursion.

Module 4:

String Functions: count(), find(), rfind(), capitalize(), title(), lower(), upper(), swapcase(), islower(), isupper(), istitle(), replace(), strip(), lstrip(),rstrip(), split(), partition(), join(), isspace(), isalpha(), isdigit(), isalnum(), startswith(), endswith(), encode(), decode(), String: Slicing, Membership, Pattern Matching, Numeric Functions: eval(), max(), min(), pow(), round(), int(), random(), ceil(), floor(), sqrt(), Date & Time Functions.

Module 5:

File opening in various modes and closing of a file, Reading from a file, Writing onto a file, File functions-open(), close(), read(), readline(), readlines(), write(), writelines(), tell(), seek(), Command Line arguments.

Labs / Practicals:

1. Write a program to take name and age from user and display it.
2. Write a program illustrating the various operators in python (arithmetic, relational, logical, and bitwise operators).
3. Write a program to check the data type of given variables.
4. Write a program to swap two variables.
5. Write a program to check whether a number is even or odd.
6. Write a program to find the largest among three numbers.
7. Write a program to check if a number is positive, negative or zero.
8. Write a program to check whether a year is a leap year or not.
9. Write a program to calculate grade based on marks.
10. Write a program to print the multiplication table of a number.
11. Write a program to find the factorial of a number using a loop.

12. Write a program to display Fibonacci series up to n terms.
13. Write a program to find the sum of digits of a number.
14. Write a program to print all prime numbers in a given range
15. Write a Python function to produce pyramid triangles.
16. Write a program to count vowels and consonants in a string.
17. Write a program to reverse a string.
18. Write a program to check whether a string is a palindrome.
19. Write a program to find the length of a string without using len().
20. Write a program to remove all punctuation from a string.
21. Write a program to find the largest and smallest number in a list.
22. Write a program to add elements to a list and sort them.
23. Write a program to find the sum and average of list elements.
24. Write a program to find duplicates in a list.
25. Write a program to convert a list to a tuple and vice versa.
26. Write a function to calculate factorial of a number.
27. Write a function to check whether a number is prime.
28. Write a function that returns the square of a number.
29. Write a program with a function to count even and odd numbers in a list.
30. Write a recursive function to find the nth Fibonacci number.
31. Write a function that takes the lengths of three sides: side1, side2 and side3 of the triangle as the input from the user using input function and return the area of the triangle as the output. Also, assert that sum of the length of any two sides is greater than the third side.
32. Write a program to create and write to a text file.
33. Write a program to read contents of a file and display it.
34. Write a program to count the number of lines and words in a file.
35. Write a program to append data to an existing file.
36. Write a program to copy contents from one file to another

References:

1. Python Network Programming Cookbook by Pradeeban Kathiravelu, Dr. M. O. Faruque Sarkar, PACKT.
2. Head First Python by Paul Berry, O'Reilly
3. Dive into Python by Mark Pilgrim, APress
4. Beginning Programming with Python Dummies by John Paul Mueller.

Course Code	Course Name	L	T	P	Cr
DCSA250084	Web Technologies	3	0	2	4

Course Outcomes:

CO1: Explain the fundamentals of web technology, including its history, evolution, client-server architecture, web browsers, web servers, types of web pages, and development environments.

CO2: Develop structured and styled web pages using HTML and CSS, incorporating responsive design principles.

CO3: Apply JavaScript for programming logic, event handling, DOM manipulation, form validation, and AJAX-based interactions.

CO4: Build dynamic web applications using server-side scripting (PHP/Node.js) and integrate them with databases (MySQL).

CO5: Demonstrate skills in website deployment, hosting, web security, content management systems, and version control with Git.

Module 1:

Definition of Web Technology, history, and evolution of the web. Understanding client-server architecture. Web Browsers and Web Servers: Definition, features, and functionalities. Web page types: static, dynamic, responsive, and interactive. Introduction to HTML, CSS, and JavaScript. Overview of web development environments.

Module 2:

Basic structure of an HTML document, HTML tags, attributes, and elements. Creating and formatting text, links, images, lists, tables, and forms. Introduction to Cascading Style Sheets, syntax, selectors, and types (inline, internal, external). Styling text, layouts, and images. Responsive design with media queries.

Module 3:

Introduction, syntax, variables, operators, and control structures. Functions, events, and event handling. DOM manipulation and traversal. Form validation using JavaScript. Introduction to AJAX for asynchronous data loading. Integrating JavaScript with HTML and CSS.

Module 4:

Introduction to server-side scripting languages: PHP, Node.js. Setting up a local server environment. Database concepts: introduction to MySQL, SQL basics, and connecting a web application with a database.

Module 5:

Types of hosting services, domain name registration, and hosting providers. Uploading and managing websites on servers. FTP and file management. Web security measures: authentication, encryption (SSL), and data protection. Introduction to content management systems (CMS) like WordPress. Version control using Git.

Labs / Practicals:

1. Create a simple web page with your name, photo, and contact details.
2. Design a student registration form using different input controls (text, radio, checkbox, etc.).
3. Create an HTML page using tables to display a college timetable.
4. Create a web page with internal and external linking.
5. Write a HTML program to design a form which should allow to enter your personal data (Hint: make use of text field, password field, e-mail, lists, radio buttons, checkboxes, submit button).
6. Write HTML Code to demonstrate the use of Anchor Tag for the Following:-
 - a. Creating a web link that opens in a new window.
 - b. Creating a web link that opens in the same window
 - c. Reference within the same html document.
 - d. Reference to some image.
 - e. Making an image a hyperlink to display second image.

7. Create an html page with following specifications. Title should be about my City, Place your City name at the top of the page in large text and in blue color. Add names of landmarks in your city each in a different color, style and typeface. One of the landmark, your college name should be blinking. Add scrolling text with a message of your choice.
8. Create an html page with 7 separate lines in different colors. State color of each line in its text.
9. Design a webpage with a navigation bar using lists.
10. Apply inline, internal, and external CSS to format text and layout.
11. Create a webpage with a fixed header and footer using CSS.
12. Design a responsive layout using Flexbox or Grid.
13. Create a webpage with hover effects on buttons and menus.
14. Build a styled form using CSS (with focus and validation effects).
15. Write a JavaScript program to validate a user login form.
16. Write a JavaScript program to display the current date and time.
17. Create a simple calculator using JavaScript.
18. Write a JavaScript program for form input validation (e.g., email, number).
19. Use JavaScript to change the content and style of an HTML element dynamically.
20. Implement an image slideshow or gallery using JavaScript.
21. Write a PHP script to display your name and current date/time.
22. Create a simple PHP calculator (Add, Subtract, Multiply, Divide).
23. Write a PHP script to process and display form data (POST/GET).
24. Write a PHP program to connect to MySQL database and display data.

References:

1. Web Technologies by Uttam K. Roy ,Oxford University Press
2. Web Technology: Theory and Practice by M. Srinivasan, Pearson Education
3. Internet and Web Technologies by Rajkamal, Tata McGraw Hill
4. Web Programming using HTML, CSS, JavaScript & PHP by Dr. Neeraj Kumar, Neeraj Publications

Course Code	Course Name	L	T	P	Cr
DCSE250041	Fundamentals of Database Systems	3	0	2	4

Course Outcomes:

CO1: Explain database concepts, architecture, data models, ER/EER modeling, and construct ER/EER diagrams.

CO2: Apply relational model concepts and relational algebra operations to formulate database queries.

CO3: Develop SQL queries for data definition, manipulation, joins, nested queries, and set operations.

CO4: Perform ER/EER to relational mapping and apply normalization techniques up to BCNF using functional dependencies.

CO5: Analyze transaction management, concurrency control mechanisms, and recovery techniques in database systems.

Module 1:

Introduction to databases: Characteristics of the database approach, advantages over file-based systems. Database system architecture, data independence, and data abstraction. Data models, Entity-Relationship (ER) modeling – entity types, attributes, relationships, constraints, and ER diagrams.

Enhanced ER (EER) model concepts – specialization, generalization, and aggregation.

Module 2:

Relational model concepts: Schema, tuple, attribute, domain, keys, and integrity constraints.

Relational Algebra operations: Select, Project, Join, Union, Intersection, Set Difference, Cartesian Product, Rename.

Understanding expressions and queries using relational algebra.

Module 3:

SQL basics: Data Definition Language (DDL), data types, and constraints (Primary Key, Foreign Key, Not Null, Unique, Check).

Data Manipulation Language (DML): Retrieval using SELECT, conditions, ordering, and grouping.

INSERT, UPDATE, DELETE statements.

Nested queries, joins (inner, outer), and set operations.

Creating and managing tables.

Module 4:

ER/EER to relational mapping: Steps and examples.

Functional dependencies and inference rules.

Normalization: First Normal Form (1NF), Second (2NF), Third (3NF), and Boyce-Codd Normal Form (BCNF).

Module 5:

Introduction to transaction processing: Concept of transaction, ACID properties.

Concurrency control: Locking protocols, deadlock handling, and prevention, Database recovery techniques.

Labs / Practicals:

Consider two tables: Student(StudentID, Name, Age, DeptID, Marks)

Departments(DeptID, Department, HOD)

1. Write an SQL query to display all the records from the Students table.
2. Write a query to display the names and marks of students who scored more than 80 marks.
3. Write a query to display the list of students who belong to the 'IT' department and are younger than 22.

4. Write a query to list all students whose names start with the letter 'A'.
5. Write an SQL query to display students ordered by their marks in descending order.
6. Write a query to find students who scored marks between 70 and 90.
7. Write an SQL query to display student names and corresponding HOD names using a JOIN on the Department column.
8. Write a query to list students from either 'CSE' or 'IT' departments.
9. Write a query to display the average marks scored by students in each department using GROUP BY.
10. Write a query to display departments where the average student marks are greater than 80 using GROUP BY and HAVING clause.

References:

1. A.K. Majumdar, P. Bhattacharyya, Database Management Systems, Tata McGraw-Hill, 1996
2. H. Korth, A. Silberschatz, Database System Concepts, McGraw-Hill (Second Edition), 1991
3. R. Elmasri, S. Navathe, Fundamentals of Database System, Benjamin Cummings (Second Edition), 1994
4. Bipin Desai, An Introduction to Database Systems, Galgotai Publication (West Publishing), 1991
5. F. McFadden, J. Hoffer, Modern Database Management, Benjamin Cummings (Narosa), (Fourth Edition), 1994

Course Code	Course Name	L	T	P	Cr
DCSA250031	Full Stack Web Development	3	0	2	4

Course Outcomes:

CO1: Explain full stack development concepts, client-server model, and use essential development tools effectively.

CO2: Develop interactive web pages using HTML, CSS, JavaScript events, form validation, and dynamic content.

CO3: Build React applications with components, JSX, props, state, interactive forms, and conditional rendering.

CO4: Create backend APIs using Node.js and Express, connect with frontend, and exchange JSON data.

CO5: Integrate frontend and backend features into a functional mini project with data persistence.

Module 1:

What is full stack development? Difference between front-end and back-end, Understanding how websites work (client/server model), Introduction to tools: VS Code, Chrome, Node.js, GitHub.

Module 2:

Simple web page structure using HTML tags, CSS for styling layout and elements JavaScript basics: variables, events, interaction, Handling user actions like clicks and input, Creating and validating simple forms, Dynamic content display using JavaScript.

Module 3:

Introduction to React and benefits, Setting up a React project, Understanding components and JSX, Passing data with props and using useState, Creating interactive forms, /Handling input changes and button clicks, Rendering lists and conditional content.

Module 4:

Introduction to Backend and APIs, What is Node.js? What is a server? Creating a simple Express serve, Sending and receiving JSON. Simple REST API with Express, Create routes: GET, POST, Connecting Backend to Frontend: Calling backend from React using fetch, Displaying backend data in UI, Submitting form data to backend.

Module 5:

Developing a mini project Using the Features: Frontend form (React), Backend data saving (Node), displaying data list.

Labs / Practicals:

preparing

References:

1. "Full Stack Development with React and Node.js", By David Choi
2. "Pro MERN Stack" (2nd Edition), By Vasan Subramanian
3. JavaScript and JQuery: Interactive Front-End Web Development", By Jon Duckett
4. "Learning Full-Stack JavaScript Development", By Eric Bush

Course Code	Course Name	L	T	P	Cr
DCSE250020	Cloud Computing	3	0	2	4

Course Outcomes:

CO1: Explain the evolution, architecture, and key characteristics of cloud computing, including service models (IaaS, PaaS, SaaS) and deployment types (public, private, hybrid).

CO2: Describe various types and implementation levels of virtualization, including CPU, memory, storage, and network virtualization in cloud environments.

CO3: Analyze layered cloud architecture and address design challenges related to inter-cloud resource management and platform deployment.

CO4: Apply distributed and parallel programming paradigms such as MapReduce and Hadoop for cloud-based application development and understand various cloud platforms like AWS, OpenStack, and Google App Engine.

CO5: Evaluate cloud security challenges and solutions, including data and application security, VM security, and risk management strategies in cloud environments.

Module 1:

Introduction to Cloud Computing

Evolution of Cloud Computing, System Models for Distributed and Cloud Computing, NIST Cloud Computing Reference Architecture, Features of Cloud Computing, Cloud Services – IaaS, PaaS, SaaS, Cloud service Providers – Public, Private and Hybrid Clouds.

Module 2:

Introduction to component virtualization

Basics of Virtualization, Types of Virtualization, Implementation Levels of Virtualization, Virtualization of CPU, Memory, I/O Devices, Desktop Virtualization, Server Virtualization, Storage Virtualization, Network Virtualization.

Module 3:

Architectural Design of Compute and Storage Clouds

Layered Cloud Architecture Development, Design Challenges, Inter Cloud Resource Management, Resource Provisioning and Platform Deployment, Global Exchange of Cloud Resources.

Module 4:

Parallel and Distributed Programming Paradigms

Map Reduce, Twister and Iterative MapReduce, Hadoop Library from Apache, Mapping Applications, Programming Support, Google App Engine, Amazon AWS, Cloud Software Environments - Eucalyptus, Open Nebula, OpenStack.

Module 5:

Security Overview

Cloud Security Challenges, Software-as-a-Service Security, Security Governance, Risk Management, Security Monitoring, Security Architecture Design, Data Security, Application Security, Virtual Machine Security.

Labs / Practicals:

1. Exploring Cloud Deployment and Service Models
2. Simulating Public, Private, and Hybrid Clouds
3. Virtualization of CPU and Memory
4. Network and Storage Virtualization
5. Deployment of a Layered Cloud Architecture
6. Resource Provisioning and Scaling
7. Deploying a Web Application
8. Simulating Security Attacks and Defenses in a Cloud VM
9. Implementing Access Control and Encryption on Cloud Storage

References:

- [1] R. Buyya, C. Vecchiola, and T. Selvi, Mastering Cloud Computing: Foundations and Applications Programming. New Delhi, India: McGraw-Hill Education, 2013.
- [2] A. T. Velte, T. J. Velte, and R. Elsenpeter, Cloud Computing: A Practical Approach. New Delhi, India: McGraw-Hill Education, 2010.
- [3] K. Hwang, G. C. Fox, and J. J. Dongarra, Distributed and Cloud Computing: From Parallel Processing to the Internet of Things. Burlington, MA, USA: Morgan Kaufmann, 2012.
- [4] T. Erl, R. Puttini, and Z. Mahmood, Cloud Computing: Concepts, Technology & Architecture. Boston, MA, USA: Pearson Education, 2013.
- [5] G. Reese, Cloud Application Architectures: Building Applications and Infrastructure in the Cloud. Sebastopol, CA, USA: O'Reilly Media, 2009.

Course Code	Course Name	L	T	P	Cr
DCSE250037	Computer Networks and Security	3	0	2	4

Course Outcomes:

CO1: Understand the fundamentals of computer networks, including protocols, architectures, and networking technologies.

CO2: Analyze and design network architectures for efficient data transmission, routing, and switching.

CO3: Gain proficiency in network security concepts, including cryptography, authentication, and access control.

CO4: Explore advanced topics such as wireless networks, network management, and cloud computing.

CO5: Develop practical skills in configuring and securing network devices, conducting vulnerability assessments, and implementing security measures to protect against cyber threats.

Module 1:

Introduction: Data Communications, Networks, The Internet, Protocols and Standards, Network Models, Layered Tasks, The OSI Model, Layers in the OSI Model, TCP/IP Protocol Suite, Addressing, Physical Layer and Media, Data and Signals, Analog and Digital, Periodic Analog Signals, Digital Signals, Transmission impairment, Data Rate Limits, Performance, Network topologies.

Module 2:

Physical and Data Link Layer: Bandwidth utilization: Multiplexing and Spreading, Multiplexing, Spread Spectrum, Transmission Media, Guided Media, Unguided Media: Wireless, Switching, Circuit-Switched Networks, Datagram Networks, Virtual-Circuit Networks, Structure of a Switch, Introduction, Block Coding, Linear Block Codes, Cyclic Codes, Checksum, Data Link Control, Framing, Flow and Error Control, Protocols, Noiseless Channels, HDLC, Point-to-Point Protocol, Multiple Access, Random Access, Aloha, Controlled Access, Channelization, IEEE Standards, Standard Ethernet, Changes in the Standard, Fast Ethernet, Gigabit Ethernet, IEEE 802.11, Bluetooth, Connecting LANs, Backbone Networks, and Virtual LANs, Connecting Devices, Backbone Networks, Virtual LANs, Sonet Networks, Virtual Tributaries, Virtual-Circuit Networks: Frame Relay and ATM, Frame Relay, ATM, ATM LANs.

Module 3:

Network Layer and Transport layer: Logical Addressing, IPv4 Addresses, IPv6 Addresses, Network Layer: Internet Protocol, Internetworking, IPv4, IPv6, Transition from IPv4 to IPv6, Network Layer: Address Mapping, Error Reporting and Multicasting, Address Mapping, ICMP, IGMP, ICMPv6, Network Layer: Delivery, Forwarding and Routing, Delivery, Forwarding, Unicast Routing Protocols, Multicast Routing Protocols. Process-to-Process Delivery: UDP, TCP and SCTP, Process-to-Process Delivery, User Datagram Protocol (UDP), TCP, SCTP, Congestion Control and Quality of Service, Data Traffic, Congestion, Congestion Control, Two Examples, Quality Service, Techniques to improve QoS, Integrated Services, Differentiated Services, QoS in Switched Networks.

Module 4:

Application Layer : Domain Name System, Name Space, Domain Name Space, Distribution of Name Space, DNS in the Internet, Resolution, DNS Messages, Types of Records, Registrars, Dynamic Domain Name System (DDNS), Encapsulation, Remote Logging, Electronic Mail and File Transfer, Remote Logging, Telnet, Electronic Mail, File Transfer, WWW and HTTP: Architecture, Web Documents, HTTP, Network Management: SNMP, Network Management System, Simple Network Management Protocol (SNMP), Multimedia, Digitizing Audio and Video, Audio and Video Compression, Streaming Stored Audio/Video, Streaming Live Audio/Video, Real-Time Interactive Audio/Video, RTP, RTCP, Voice over IP.

Module 5:

Network Security: Security concepts and terminology TCP/IP and OSI network security access control issues (packet filters, firewalls) communication security (OSI layer security protocols) security tools cryptography- Public Key Cryptography And Its Application, Cyber Security and its application

Labs / Practicals:

1. Simulate and analyze different network topologies (bus, ring, star, mesh) using simulation tools like Cisco Packet Tracer or NS2.
2. Configure static and dynamic IP addressing in a LAN setup using network simulation software.
3. Implement error detection techniques such as parity check, checksum, and cyclic redundancy check (CRC) in C/C++.
4. Simulate flow and error control protocols such as Stop-and-Wait, Go-Back-N, and Selective Repeat ARQ.
5. Configure and test Ethernet and Wi-Fi connectivity using physical lab setup or simulator.
6. Demonstrate the working of switching (circuit, packet, and message switching) using practical scenarios or simulators.
7. Configure and test routing protocols like RIP, OSPF, and EIGRP using network simulation tools.
8. Demonstrate TCP and UDP socket programming using simple client-server communication in Python or C.
9. Implement encryption and decryption using Caesar Cipher, RSA, or AES algorithms in a high-level programming language.
10. Simulate firewall and packet filtering rules using open-source tools like Wireshark and IPtables.

References:

1. Data Communications and Networking, Fourth Edition by Behrouza A. Forouzan, TMH.
2. Computer Networks, A.S. Tanenbaum, 4th edition, Pearson education.
3. Introduction to Data communications and Networking, W. Tomasi, Pearson education.
4. Data and Computer Communications, G.S. Hura and M. Singhal, CRC Press, Taylor and Francis Group.
5. An Engineering Approach to Computer Networks-S. Keshav, 2nd Edition, Pearson Education
6. Understanding communications and Networks, 3rd Edition, W. A. Shay, Cengage Learning.
7. "Modern Cryptography, Theory & Practice", Pearson Education. Wenbo Mao
8. "Computer Security", Pearson Education. Matt Bishop

Course Code	Course Name	L	T	P	Cr
DCSE250044	Cyber Security	3	0	2	4

Course Outcomes:

CO1: Explain fundamentals of cybersecurity, security goals, common threats, attacks, malware, cybercrime, and legal frameworks.

CO2: Apply cryptographic techniques including classical ciphers, modern encryption algorithms, and hashing functions for securing information.

CO3: Analyze network security concepts such as firewalls, IDS, VPNs, and wireless security, and simulate basic attacks and defenses.

CO4: Demonstrate system and application security through OS hardening, malware analysis, and secure coding practices against web-based attacks.

CO5: Use cybersecurity tools (e.g., Wireshark, Nmap, Burp Suite, Metasploit) and prepare incident/case study reports to evaluate real-world cyber threats.

Module 1:

Introduction to Cyber Security:

Basics of Information Security, CIA Triad: Confidentiality, Integrity, Availability, Types of Threats and Attacks, Malware: Virus, Worm, Trojan, Ransomware, Phishing, Social Engineering, Security Goals and Challenges, Cybercrime and Cyber Laws (IT Act 2000).

Module 2:

Cryptography Fundamentals :

History and Evolution of Cryptography, Symmetric vs Asymmetric Encryption, Classical Ciphers: Caesar, Vigenère, Playfair, Modern Cryptography: AES, DES, RSA, Hash Functions: MD5, SHA-1, SHA-256.

Module 3:

Network Security :

Basics of Computer Networks, Network Threats: DoS, DDoS, IP Spoofing, Sniffing, Firewalls and Intrusion Detection Systems (IDS), VPNs and Secure Tunneling, Wireless Security (WEP, WPA2).

Module 4:

System and Application Security:

OS Security: Windows and Linux, Secure Configuration and Hardening, Malware Analysis Basics, Web Application Security: SQL Injection, Cross-site Scripting (XSS), Cross-site Request Forgery (CSRF), Secure Coding Practices.

Module 5:

Cyber Security Tools and Case Studies:

Security Tools: Nmap, Metasploit, Burp Suite, Digital Forensics Basics, Security Audit Process, Ethical Hacking Overview, Recent Cybersecurity Case Studies.

Labs / Practicals:

Module 1: Introduction to Cyber Security

1. Simulate a Phishing Attack Detection:

Identify and report phishing emails from a set of real-looking email samples. Explain the signs you observed.

2. Analyze a Cybercrime Case Study:

Select a real-world cyber attack (e.g., Facebook Data Leak, WannaCry) and create a report explaining the type of attack, vulnerabilities exploited, and impact.

Module 2: Cryptography Fundamentals

3. Implement Caesar Cipher in Python:

Write a program to encrypt and decrypt a given message using Caesar Cipher with a shift key.

4. Implement Vigenère Cipher in Python:

Create a script to encode and decode text using a Vigenère Cipher.

5. Basic RSA Encryption-Decryption Program:

Simulate key generation, encryption, and decryption using basic RSA algorithm (use Python or any language).

6. Generate Hashes of a File:

Use command-line tools or a Python script to generate MD5 and SHA256 hashes of a file and verify integrity.

Module 3: Network Security

7. Packet Analysis using Wireshark:

Capture live network traffic and analyze packets to identify IP addresses, protocols, and suspicious patterns.

8. Scan Network using Nmap:

Use Nmap to scan a local network or localhost and generate a report on open ports and running services.

9. Firewall Rule Configuration (Windows/Linux):

Configure basic firewall rules to block/allow specific IP addresses or ports and verify the effect.

10. Simulate a Ping Flood DoS Attack (in test environment):

Use tools like hping3 or ping with large packet sizes to simulate a ping flood and monitor the system/network behavior.

Module 4: System and Application Security

11. Perform SQL Injection on a Test Web App (DVWA):

Use DVWA (Damn Vulnerable Web App) or bWAPP to perform SQL injection and extract data.

12. Cross-site Scripting (XSS) Demonstration:

Identify and exploit an XSS vulnerability in a test environment and demonstrate how input sanitization can fix it.

13. System Hardening Checklist (Windows/Linux):

Apply at least 5 system hardening techniques to your OS (disable services, create strong passwords, enable firewall) and document them.

Module 5: Tools and Case Studies

14. Use Burp Suite to Intercept HTTP Traffic:

Use Burp Suite to intercept, analyze, and modify HTTP requests/responses between browser and web app.

15. Create a Cybersecurity Incident Report:

Prepare a structured report analyzing a recent cyber attack (use open data or news) including type of attack, tools used, impact, and lessons learned.

References:

1 William Stallings – Cryptography and Network Security

2 Nina Godbole – Cyber Security: Understanding Cyber Crimes

3 Online platforms: Cybrary, OWASP, TryHackMe, Hacker101

Course Code	Course Name	L	T	P	Cr
DOAI250005	Artificial Intelligence and Machine Learning	2	0	4	4

Course Outcomes:

CO1: Understand AI problem-solving techniques, heuristic search algorithms, and reinforcement learning.

CO2: Apply knowledge representation, logic-based reasoning, and probabilistic models in AI systems.

CO3: Develop and evaluate machine learning models using supervised and unsupervised learning techniques.

CO4: Design and implement deep learning architectures for vision, language, and sequential data applications.

CO5: Analyze advanced AI methods, ethical challenges, and emerging trends in artificial intelligence.

Module 1:

AI Techniques and Search Strategies

History of AI, problem-solving using search, uninformed vs. informed search, heuristic search techniques (Hill Climbing, Simulated Annealing, A* Algorithm), adversarial search (Minimax Algorithm, Alpha-Beta Pruning), AI-driven game strategies (Chess, Chinese Checkers), Reinforcement Learning fundamentals (Markov Decision Processes, Q-learning), heuristic optimization techniques (Genetic Algorithms, Particle Swarm Optimization).

Module 2:

Knowledge Representation, Reasoning, and Probabilistic Models

Propositional logic, inference mechanisms, forward and backward chaining, probabilistic reasoning techniques (Bayesian Networks, Markov Models, Probabilistic Graphical Models), logic-based AI vs. machine learning-driven reasoning, Natural Language Processing (NLP) fundamentals (tokenization, embeddings, language models), AI-driven text processing, machine translation.

Module 3:

Machine Learning Fundamentals and Neural Networks

Supervised and unsupervised learning paradigms, classification and regression techniques (decision trees, support vector machines, ensemble learning methods: Random Forest, XGBoost), artificial neural networks (activation functions, loss functions, back propagation, multilayer perceptions), overfitting, bias-variance tradeoff, hyperparameter tuning, performance evaluation metrics (precision-recall, ROC curves, F1-score).

Module 4:

Deep Learning and Advanced Neural Architectures

Convolutional Neural Networks (CNNs) for image classification and object detection (convolution layers, pooling, fully connected layers), Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks for sequential data processing, Transformer models (BERT, GPT) for NLP applications, optimization techniques (batch normalization, dropout regularization, transfer learning), deep learning applications (medical imaging, self-driving vehicles, generative AI).

Module 5:

Unsupervised Learning, Generative Models, and Ethical AI

Clustering techniques (K-Means, Hierarchical Clustering), Autoencoders, Generative Adversarial Networks (GANs), self-supervised learning, foundation models in AI, AI applications in healthcare, finance, and autonomous systems, AI ethics (fairness, bias mitigation, adversarial attacks, explainable AI), advancements in AI (quantum computing, federated learning, AI governance frameworks).

Labs / Practicals:

1. Implement a basic uninformed search algorithm (e.g., Breadth-First Search, Depth-First Search).
2. Implement informed search algorithms using heuristics (e.g., A* Algorithm for pathfinding).
3. Implement the Hill Climbing algorithm for local search problems.
4. Develop a solution using Simulated Annealing for optimization problems.
5. Implement the Minimax algorithm to solve a simple game (e.g., Tic-Tac-Toe).
6. Implement Alpha-Beta Pruning for optimizing the Minimax algorithm.
7. Build a Reinforcement Learning agent using Q-learning to solve a basic maze problem.
8. Implement Genetic Algorithms to solve optimization problems (e.g., travelling salesman).
9. Build a Particle Swarm Optimization model for function optimization.
10. Implement propositional logic and inference mechanisms (e.g., forward and backward chaining).
11. Develop a basic Bayesian Network for probabilistic reasoning.

References:

1. Tom M. Mitchell: Machine Learning. McGraw Hill
2. Artificial Intelligence: A Modern Approach by Stuart Russell and Peter Norvig
3. Pattern Recognition and Machine Learning by Christopher M. Bishop
4. Machine Learning: A Probabilistic Perspective by Kevin P. Murphy
5. Reinforcement Learning: An Introduction by Richard S. Sutton and Andrew G. Barto
6. Natural Language Processing with Python by Steven Bird, Ewan Klein, and Edward Loper
7. The Elements of Statistical Learning" by Trevor Hastie, Robert Tibshirani, and Jerome Friedman
8. Michael Nielsen: Neural Networks and Deep Learning (free online book)
9. Stuart Russel & Peter Norvig: Artificial Intelligence– A Modern Approach. Pearson, 4th edition, 2023
10. Deep Learning (Adaptive Computation and Machine Learning series) Illustrated Edition, by Ian Goodfellow, Yoshua Bengio, Aaron Courville, MIT Press, London

Course Code	Course Name	L	T	P	Cr
DOAI250026	Data Analysis Using Spreadsheet	3	0	2	4

Course Outcomes:

CO1: Understand spreadsheet interfaces, data types, entry methods, and formatting techniques.

CO2: Apply formulas, cell referencing, and commonly used functions for data manipulation.

CO3: Perform data filtering, cleaning, and summarization using Pivot Tables and advanced tools.

CO4: Create Pivot Tables, Pivot Charts, and apply best practices for effective data visualization.

CO5: Design dynamic charts, apply conditional formatting, and develop interactive dashboards.

Module 1:

Overview of spreadsheet applications (Excel, Google Sheets, LibreOffice Calc), Interface and basic operations: cells, rows, columns, worksheets Data types and data entry (text, numbers, dates, formulas). Formatting cells and data (number, text, conditional formatting)

Module 2:

Introduction to formulas and cell referencing (relative, absolute) ,Common functions,Text Functions: CONCATENATE, LEFT, RIGHT, MID, LEN, TRIM, Mathematical Functions: SUM, AVERAGE, MIN, MAX, COUNT, ROUND,Logical Functions: IF, AND, OR, NOT
Lookup Functions: VLOOKUP, HLOOKUP, XLOOKUP, INDEX & MATCH

Module 3:

AutoFilter and Advanced Filter, Custom views and data ranges, Removing duplicates and handling missing data, Using slicers for dynamic filtering (in Excel),Pivot Tables and Data Summarization .

Module 4:

Introduction to Pivot Tables, Creating Pivot Tables from raw data, Grouping data in Pivot Tables ,Using Pivot Charts, Dynamic data analysis using Pivot Table filters and slicers, Charts and Graphs for Data Visualization :Creating basic charts: Bar, Line, Pie, Column, Customizing chart elements: Titles, Axis, Legends, Data Labels, Best practices for visualizing data clearly

Module 5:

Combination charts (e.g., bar + line), Dynamic charts using named ranges and tables, Using sparklines for trend analysis, Conditional formatting with charts, Dashboards overview: integrating charts, tables, and filters.

Labs / Practicals:

1. Create a spreadsheet to store student records (Name, Roll No, Marks in 3 subjects, Total, Percentage).
2. Apply basic formatting (font style, size, color, borders, cell merge, wrap text).
3. Use conditional formatting to highlight students scoring below 40%.
4. Sort data based on Total Marks (ascending and descending).
5. Use basic formulas to calculate Total and Percentage.
(Use =SUM(), =AVERAGE(), =IF(), =MAX(), =MIN())
6. Calculate Grade using IF and nested IF statements.
7. Use text functions like LEFT, RIGHT, LEN, CONCATENATE or TEXTJOIN.
8. Date and time functions: Calculate number of days between two dates using DATEDIF() or TODAY().
9. Create a Pivot Table to summarize department-wise average marks.
10. Use filters and slicers to analyze data for specific conditions (e.g., show only students with A grade).
11. Create data validation (dropdown list) for selecting department names.
12. Create bar chart to compare marks of students across subjects.
13. Create pie chart to represent percentage distribution of grades.
14. Use sparkline charts in Google Sheets for visualizing trends in student marks.

15. Apply statistical functions like STDEV(), VAR(), MODE(), MEDIAN() on a dataset.
16. Perform correlation analysis between two data columns using CORREL().

References:

1. Data Analysis Using Excel by Shanthi Narayanan, Cengage Learning India
2. Statistical Data Analysis Using Excel by J.K. Sharma, Vikas Publishing
3. Excel for Beginners and Data Analysis by Dr. Pradeep K. Sinha, BPB Publications