MK UNIVERSITY

PATAN, GUJARAT

ESTABLISHED BY THE GUJARAT GOVT. **RECOGNIZED BY UGC UNDER SECTION 2(f) OF UGC ACT,1956**

MK University, Patan
Faculty of Engineering Technology,
Department of Information Technology

DIPLOMA (INFORMATION TECHNOLOGY) SEM-I									
SR NO.	COURSE TYPE	COURSE CODE	COURSE NAME	LECTURE (HRS.)/WEEK	PRACTICAL (HRS.)/WEEK	CREDITS	EXAMINATION		TOTAL MARKS
							INTERNAL	EXTERNAL	
1	MAJOR	DITE101	ENGINEERING MATHEMATICS-I	4	0	4	40	60	100
2	MAJOR	DITE102	ENGINEERING PHYSICS	4	2	6	90	60	150
3	MAJOR	DITE103	ENGINEERING CHEMISTRY	4	2	6	90	60	150
4	MAJOR	DITE104	PYTHON PROGRAMMING	4	2	6	90	60	150
5	MINOR	DITE105	WORKSHOP PRACTICE	0	2	2	50	00	50
TOTAL				16	8	24	360	240	600

DIPLOMA (INFORMATION TECHNOLOGY) SEM-II									
SR NO.	COURSE TYPE	COURSE CODE	COURSE NAME	LECTURE (HRS.)/WEEK	PRACTICAL (HRS.)/WEEK	CREDITS	EXAMINATION		TOTAL MARKS
							INTERNAL	EXTERNAL	
1	MAJOR	DITE201	ENGINEERING MATHEMATICS-II	4	0	4	40	60	100
2	MAJOR	DITE202	DATA STRUCTURES AND ALGORITHMS	4	2	6	90	60	150
3	MAJOR	DITE203	DATABASE MANAGEMENT SYSTEMS	4	2	6	90	60	150
4	MINOR	DITE204	WEB TECHNOLOGIES	4	2	6	90	60	150
5	SEC	DITE205	COMMUNICATION SKILL	2	0	2	00	50	50
TOTAL				18	6	24	310	290	600

MK UNIVERSITY

PATAN, GUJARAT

ESTABLISHED BY THE GUJARAT GOVT. **RECOGNIZED BY UGC UNDER SECTION 2(f) OF UGC ACT,1956**

DIPLOMA (INFORMATION TECHNOLOGY) SEM-III									
SR NO.	COURSE TYPE	COURSE CODE	COURSE NAME	LECTURE (HRS.)/ WEEK	PRACTICAL (HRS.)/ WEEK	CREDITS	EXAMINATION		TOTAL MARKS
							INTERNAL	EXTERNAL	
1	MAJOR	DITE301	OBJECT ORIENTED PROGRAMMING	4	2	6	90	60	150
2	MAJOR	DITE302	COMPUTER NETWORKS	4	2	6	90	60	150
3	MAJOR	DITE303	OPERATING SYSTEMS	4	2	6	90	60	150
4	MINOR	DITE304	INDUSTRIAL VISIT REPORT	0	2	2	50	00	50
5	IKS	DITE305	IKS-ANNCIENT INDIAN ENGINEERING PRACTICE	0	2	2	50	00	50
TOTAL				12	10	22	370	180	550

DIPLOMA (INFORMATION TECHNOLOGY) SEM-IV									
SR NO.	COURSE TYPE	COURSE CODE	COURSE NAME	LECTURE (HRS.)/ WEEK	PRACTICAL (HRS.)/ WEEK	CREDITS	EXAMINATION		TOTAL MARKS
							INTERNAL	EXTERNAL	
1	MAJOR	DITE401	CYBER SECURITY	4	0	4	40	60	100
2	MAJOR	DITE402	MOBILE APPLICATION DEVELOPMENT	4	2	6	90	60	150
3	MAJOR	DITE403	CLOUD COMPUTING BASICS	4	0	4	40	60	100
4	MINOR	DITE404	DIGITAL ELECTRONICS	4	2	6	90	60	150
5	VAC	DITE405	ENVIRONMENTAL SCIENCE	2	0	2	00	50	50
TOTAL				18	4	22	260	290	550

MK UNIVERSITY

PATAN, GUJARAT

ESTABLISHED BY THE GUJARAT GOVT. **RECOGNIZED BY UGC UNDER SECTION 2(f) OF UGC ACT,1956**

DIPLOMA (INFORMATION TECHNOLOGY) SEM-V									
SR NO.	COURSE TYPE	COURSE CODE	COURSE NAME	LECTURE (HRS.)/ WEEK	PRACTICAL (HRS.)/WEEK	CREDITS	EXAMINATION		TOTAL MARKS
							INTERNAL	EXTERNAL	
1	MAJOR	DITE501	DATA SCIENCE FUNDAMENTALS	4	2	6	90	60	150
2	MAJOR	DITE502	INTERNET OF THINGS (IOT)	4	2	6	90	60	150
3	MAJOR	DITE503	DEVOPS FUNDAMENTALS	4	0	4	40	60	100
4	MINOR	DITE504	ENTERPRENEURSHIP IN IT	4	0	4	40	60	100
5	SEC	DITE505	MIN PROJECT	0	2	2	50	00	50
TOTAL				16	6	22	310	240	550

DIPLOMA (INFORMATION TECHNOLOGY) SEM-VI									
SR NO.	COURSE TYPE	COURSE CODE	COURSE NAME	LECTURE (HRS.)/ WEEK	PRACTICAL (HRS.)/WEEK	CREDITS	EXAMINATION		TOTAL MARKS
							INTERNAL	EXTERNAL	
1	MAJOR	DITE601	ARTIFICIAL INTELLIGENCE& ML BASICS	4	2	6	90	60	150
2	MAJOR	DITE602	SOFTWARE ENGINEERING	4	2	6	90	60	150
3	MAJOR	DITE603	BIG DATA ANALYTICS	4	2	6	90	60	150
4	MINOR	DITE604	DIPLOMA PROJECT	0	6	6	150	00	150
TOTAL				12	12	24	420	180	600

MK UNIVERSITY

PATAN, GUJARAT

ESTABLISHED BY THE GUJARAT GOVT. RECOGNIZED BY UGC UNDER SECTION 2(f) OF UGC ACT,1956

SEMESTER-I

COURSE CODE: DITE101

COURSE NAME: ENGINEERING MATHEMATICS-I

Course Objectives:

- To develop foundational mathematical skills essential for mechanical engineering applications
- To apply algebraic, trigonometric, and calculus concepts to solve engineering problems
- To interpret and analyze data using statistical methods
- To build problem-solving abilities through applied mathematics
- To prepare students for advanced engineering mathematics in subsequent semesters

Course Outcomes: At the end of the course students shall be able to

CO1	**Solve** algebraic equations and apply them to engineering problems
CO2	**Apply** trigonometric functions to analyze mechanical systems
C03	**Perform** basic differentiation and integration relevant to engineering applications
C04	**Analyze** data using measures of central tendency and dispersion

Unit	Content	Credit	Weightage
I	**Algebra and Trigonometry** **Topics:** • **Algebra:** Quadratic equations, simultaneous linear equations (2 and 3 variables) • Arithmetic and geometric progressions • **Trigonometry:** Trigonometric ratios, identities, compound angles • Heights and distances (engineering applications) • Complex numbers: basics and operations • **Applications:** Simple harmonic motion, projectile motion, force resolution	1	25%

II	**Differential Calculus**
	Topics:
	• Functions, limits, and continuity
	• Derivatives: Standard formulas
	• Rules of differentiation: Product, quotient, chain rule
	• Applications of derivatives:
	  ○ Rate of change (velocity, acceleration)
	  ○ Maxima and minima (optimization problems)
	  ○ Tangents and normals
	• Partial differentiation (introduction)
	• **Applications:** Optimization in design, motion analysis, slope of curves
		1	25%

III	**Integral Calculus** **Topics:** • Indefinite integrals: Standard formulas	1	25%

	• Methods of integration: Substitution, integration by parts
	• Definite integrals and properties
	• Applications of integration:
	  ○ Area under curves
	  ○ Volume of solids of revolution
	  ○ Center of gravity/centroid (basic concepts)
	• **Applications:** Area calculation in engineering drawings, volume of tanks/containers		

IV	**Statistics and Probability** **Topics:** • **Statistics:** Data classification, frequency distribution • Measures of central tendency: Mean, median, mode • Measures of dispersion: Range, standard deviation, variance • Graphical representation: Histogram, frequency polygon, ogive • **Probability:** Basic concepts, addition and multiplication theorems • **Applications:** Quality control, measurement analysis, manufacturing data interpretation	1	25%

Textbooks:

- Primary: *Engineering Mathematics* — NP Bali & Dr. Manish Goyal
- Primary: *A Textbook of Engineering Mathematics* — B.S. Grewal

Reference books:

- *Advanced Engineering Mathematics* — H.K. Das
- *Engineering Mathematics* — D. G. Gupta
- *Basic Technical Mathematics with Calculus* — Allyn J. Washington
- *Mathematics for Mechanical Engineering* — B.V. Ramana

Online Platforms:

1. NPTEL Videos: "Basic Course in Mathematics" for engineering
2. Coursera: "Pre-Calculus" by University of California, Irvine

COURSE CODE: DITE102

COURSE NAME: ENGINEERING PHYSICS

Course Objectives:

- To understand fundamental physics principles relevant to mechanical engineering
- To apply physics concepts to solve practical engineering problems
- To develop skills in measurement, experimentation, and data analysis
- To correlate theoretical physics with mechanical systems and applications
- To build foundation for advanced engineering courses

Course Outcomes: At the end of the course students shall be able to

CO1	**Apply** mechanics principles to analyze forces, motion, and energy in mechanical systems
CO2	**Explain** thermal physics concepts relevant to heat engines and refrigeration
C03	**Demonstrate** understanding of optics and acoustics in

	engineering contexts
C04	**Perform** measurements using physical instruments and analyze experimental data

Unit	Content	Credit	Weightage
I	**Mechanics & Properties of Matter** **Topics:** • Scalars and vectors, force resolution, moment of force • Laws of motion, friction, work, energy, power • Circular motion, centripetal force • Elasticity: Stress, strain, Hooke's law, Young's modulus • Surface tension and viscosity (basic concepts) • **Applications:** Machine design, material strength, fluid mechanics basics	1	25%
II	**Thermal Physics & Thermodynamics** **Topics:** • Heat and temperature, thermal expansion • Calorimetry, specific heat capacity • Laws of thermodynamics (zeroth, first, second) • Heat transfer: conduction, convection, radiation • Kinetic theory of gases (basic) • **Applications:** Heat engines, refrigeration, insulation materials	1	25%
III	**Waves, Optics & Acoustics** **Topics:** • Simple harmonic motion, wave motion • Sound: characteristics, intensity, Doppler effect • Ultrasonics and applications • Reflection, refraction, lenses, optical instruments • Fiber optics (basic principles) • **Applications:** Machine vibration, NDT, optical measurements, noise control	1	25%
IV	**Modern Physics & Material Science** **Topics:** • Quantum physics basics: photons, matter waves • Lasers: principles, types, applications • Semiconductors: basics • Superconductivity (elementary concepts) • Nanotechnology introduction • **Applications:** Laser machining, sensors, advanced materials	1	25%

Textbooks:

- Primary: *Engineering Physics* — D. R. Khanna & H. N. Srivastava
- Primary: *Engineering Physics* — R. K. Gaur & S. L. Gupta

Reference books:

- *Fundamentals of Physics* — Halliday, Resnick & Walker
- *Concepts of Physics* — H. C. Verma

- *Engineering Physics* — M. N. Avadhanulu & P. G. Kshirsagar
- *Practical Physics* — C. L. Arora

Online Platforms:

- SWAYAM/NPTEL: "Engineering Physics" courses by IITs/NITs

PRACTICAL LIST:

Section A: Mechanics

1. Vernier Calipers & Screw Gauge: Measurement of dimensions of given objects
2. Simple Pendulum: Determination of 'g' and study of laws of pendulum
3. Young's Modulus: By Searle's method or cantilever
4. Coefficient of Friction: Using inclined plane
5. Force Table: Verification of law of parallelogram of forces

Section B: Thermal Physics

6. Specific Heat Capacity: Of solid/liquid using calorimeter
7. Thermal Conductivity: Of good conductor (Searle's apparatus)
8. Mechanical Equivalent of Heat: Using Joule's calorimeter
9. Coefficient of Linear Expansion: Using optical lever

Section C: Waves & Optics

10. Sonometer: Verification of laws of vibrating strings
11. Melde's Experiment: Transverse and longitudinal modes
12. Compound Pendulum: Determination of 'g' and radius of gyration
13. Optical Bench: Focal length of convex lens
14. Prism: Refractive index using spectrometer

Section D: Modern Physics

15. LASER: Determination of wavelength using diffraction grating
16. Photoelectric Effect: Verification of Einstein's equation
17. PN Junction Diode: Characteristics
18. Thermistor: Temperature-resistance characteristics

COURSE CODE: DITE103

COURSE NAME: ENGINEERING CHEMISTRY

Course Objectives:

- To provide fundamental knowledge of chemistry relevant to mechanical engineering applications
- To understand material properties, corrosion, fuels, and lubricants from chemical perspective
- To develop skills in chemical analysis, quality control, and material testing
- To correlate chemical principles with mechanical systems and manufacturing processes
- To build foundation for materials science, metallurgy, and environmental engineering

Course Outcomes: At the end of the course students shall be able to

CO1	**Explain** water treatment processes for industrial applications
CO2	**Analyze** properties of fuels and lubricants used in mechanical systems
C03	**Identify** corrosion mechanisms and prevention methods
C04	**Apply** principles of electrochemistry to batteries and corrosion control

Unit	Content	Credit	Weightage

I	**Water Chemistry & Treatment** **Topics:** • Water impurities: hardness, alkalinity, pH • Water softening methods: lime-soda, ion exchange • Boiler feed water treatment: scale and sludge formation, prevention • Cooling water treatment • Drinking water standards • **Applications:** Boiler operations, cooling systems, industrial water supply	1	25%
II	**Fuels & Combustion** **Topics:** • Classification of fuels: solid, liquid, gaseous • Calorific value determination: bomb calorimeter • Solid fuels: coal analysis (proximate & ultimate) • Liquid fuels: petroleum refining, petrol, diesel, octane/cetane number • Gaseous fuels: LPG, CNG, biogas • Combustion calculations • **Applications:** IC engines, furnaces, power generation	1	25%
III	**Lubricants & Corrosion** **Topics:** • Lubrication: mechanisms, types of lubricants • Properties: viscosity index, flash point, pour point • Additives in lubricants • Corrosion: types, mechanisms (electrochemical) • Factors affecting corrosion • Corrosion prevention methods • **Applications:** Machine maintenance, automotive, industrial equipment	1	25%
IV	**Engineering Materials & Polymers** **Topics:** • Cement: composition, setting and hardening • Refractories: properties, classification • Polymers: addition, condensation, engineering plastics • Composite materials: introduction • Batteries: primary, secondary, fuel cells • **Applications:** Construction materials, polymers in engineering, energy storage	1	25%

Textbooks:

- Primary: *Engineering Chemistry* — Jain & Jain
- Primary: *Engineering Chemistry* — Dr. O. P. Verma

Reference books:

- *A Textbook of Engineering Chemistry* — S. S. Dara & S. S. Umare
- *Engineering Chemistry* — R. P. Mani & K. N. Mishra
- *Chemistry for Engineering Students* — B. S. Jai Prakash & R. Venugopal
- *Applied Chemistry* — H. D. Gesser

Online Platforms:

- SWAYAM/NPTEL: "Engineering Chemistry" courses by IITs
- Khan Academy: Chemistry fundamentals
- MERLOT: Chemistry learning materials

PRACTICAL LIST:

Section A: Water Analysis

1. Determination of Hardness: By EDTA titration method
2. Alkalinity Determination: Using acid-base titration
3. pH Measurement: Using pH meter/universal indicator
4. Chloride Content: By argentometric method

Section B: Fuels & Lubricants

5. Viscosity Measurement: Using Ostwald viscometer/Redwood viscometer
6. Flash Point Determination: Using Abel/Pensky Martens apparatus
7. Calorific Value: Bomb calorimeter demonstration
8. Saponification Value: Of oil sample

Section C: Corrosion & Electrochemistry

9. Corrosion Rate Measurement: Weight loss method
10. Galvanic Series Determination
11. Electroplating: Copper plating on iron
12. EMF Measurement: Of simple galvanic cell

Section D: Materials & Polymers

13. Cement Setting Time: Initial and final setting time
14. Polymer Identification Tests
15. Preparation of Polymer: Phenol-formaldehyde/Bakelite
16. Refractory Properties: Porosity, thermal shock resistance

COURSE CODE: DITE104

COURSE NAME: PYTHON PROGRAMMING

Course Objectives:

- To introduce fundamental concepts of Python programming and its applications.
- To develop skills in writing Python programs using control structures, functions, and data structures.
- To enable students to implement file handling, exception handling, and modular programming in Python.
- To prepare students for advanced topics like OOP, database connectivity, and data processing.
- To foster problem-solving abilities using Python for real-world applications.

Course Outcomes: At the end of the course students shall be able to

CO1	**Explain** Python basics, data types, control structures, and functions.
CO2	**Implement** programs using lists, tuples, dictionaries, sets, and strings.
C03	**Apply** file handling, exception handling, and modular programming in Python.
C04	**Develop** Python applications using OOP, modules, and database connectivity.

Unit	Content	Credit	Weightage

I	**Python Basics & Control Structures** • Introduction to Python: Features, installation, IDLE, Jupyter Notebook • Python syntax, variables, data types, operators • Input/output functions: input(), print() • Control structures: if, if-else, nested if, loops (while, for) • Loop control: break, continue, pass • **Applications**: Simple calculator, number guessing game, pattern printing	1	25%
II	**Data Structures & Functions** • Python data structures: Lists, tuples, dictionaries, sets • String manipulation: Methods, slicing, formatting • Functions: Definition, parameters, return values, scope • Lambda functions, map, filter, reduce • List comprehensions • **Applications**: Student mark processing, word frequency counter, to-do list app	1	25%
III	**File Handling & Exception Handling** • File handling: Opening, reading, writing, appending (open(), read(), write()) • File modes, with statement • Exception handling: try, except, else, finally • User-defined exceptions • **Applications**: Log file analyzer, data backup script, error logging system	1	25%
IV	**OOP, Modules & Database Connectivity** • Object-Oriented Programming: Classes, objects, constructors, methods • Inheritance, polymorphism, encapsulation • Modules and packages: Creating, importing • Introduction to database connectivity using sqlite3 or mysql-connector • CRUD operations in Python • **Applications**: Employee management system, library system, simple CRM	1	25%

Textbooks:

- *Python Programming: A Modern Approach* – Vamsi Kurama
- *Learning Python* – Mark Lutz

Reference books:

- *Python Crash Course* – Eric Matthes
- *Core Python Programming* – R. Nageswara Rao
- *Python Cookbook* – David Beazley & Brian K. Jones
- *Head First Python* – Paul Barry

Online Platforms:

- NPTEL
  - *Programming, Data Structures and Algorithms in Python* by IIT Madras

- *Python for Data Science* by IIT Roorkee
- Coursera
  - *Python for Everybody* by University of Michigan
  - *Crash Course on Python* by Google

PRACTICAL LIST:

Module I: Python Basics & Control Structures

1. Write a program to calculate the area of a circle, rectangle, and triangle.
2. Write a program to check if a number is even or odd.
3. Write a program to print the multiplication table of a given number.
4. Write a program to find the factorial of a number using a loop.
5. Write a program to check if a string is a palindrome.
6. Write a program to generate Fibonacci series up to n terms.

Module II: Data Structures & Functions

7. Write a program to find the largest and smallest elements in a list.
8. Write a program to count the frequency of each character in a string using a dictionary.
9. Write a program to merge two dictionaries.
10. Write a function to check if a number is prime.
11. Write a program using list comprehension to create a list of squares from 1 to 10.
12. Write a program to sort a list of tuples based on the second element.

Module III: File Handling & Exception Handling

13. Write a program to read a text file and count the number of lines, words, and characters.
14. Write a program to copy the contents of one file to another.
15. Write a program to handle ZeroDivisionError and ValueError exceptions.
16. Write a program to log errors into a file when exceptions occur.
17. Write a program to read CSV file data and display it in tabular format.

Module IV: OOP, Modules & Database Connectivity

18. Create a class Student with attributes name, roll, marks and methods to display details.
19. Implement inheritance: class Vehicle (base) and Car (derived).
20. Create a module calculator.py with functions for basic operations and import it in another script.
21. Write a program to connect to a SQLite database and create a table Employees.
22. Write a program to insert, update, delete, and retrieve records from the Employees table.

MK UNIVERSITY

PATAN, GUJARAT

ESTABLISHED BY THE GUJARAT GOVT. RECOGNIZED BY UGC UNDER SECTION 2(f) OF UGC ACT,1956

SEMESTER-II

COURSE CODE: DITE201

COURSE NAME: ENGINEERING MATHEMATICS-II

Course Objectives:

- To build advanced mathematical skills for engineering problem-solving
- To apply differential equations to model mechanical systems
- To understand vector algebra and its applications in mechanics
- To develop skills in numerical methods for engineering computations
- To prepare mathematical foundation for subjects like Mechanics, Thermodynamics, and Machine Design

Course Outcomes: At the end of the course students shall be able to

CO1	**Solve** ordinary differential equations relevant to engineering systems
CO2	**Apply** vector algebra to analyze forces and motions in 3D space
C03	**Perform** numerical computations using interpolation, differentiation, and integration methods
C04	**Analyze** data using probability distributions and statistical methods

Unit	Content	Credit	Weightage
I	**Differential Equations** **Topics:** • First order differential equations: variable separable, homogeneous, exact • Linear differential equations of first order • Applications: Newton's law of cooling, growth and decay, simple circuits • Second order linear differential equations with constant coefficients • Complementary function and particular integral methods • **Applications:** Spring-mass systems, electrical circuits, vibration analysis	1	25%
II	**Vector Algebra & 3D Geometry** **Topics:** • Vectors: dot product, cross product, scalar triple product • Vector differentiation • Gradient, divergence, curl (basic concepts) • Lines and planes in 3D space • Direction cosines and ratios • **Applications:** Force analysis, moment of force, work done by force	1	25%
III	**Numerical Methods** **Topics:**	1	25%

	• Solution of algebraic equations: Bisection method, Newton-Raphson method • Interpolation: Newton's forward and backward difference formulas • Numerical differentiation and integration • Trapezoidal rule and Simpson's rules • **Applications:** Root finding in design equations, area calculations, data analysis		
IV	**Probability & Laplace Transforms** **Topics:** • **Probability:** Basic concepts, addition and multiplication theorems • Random variables, probability distributions (Binomial, Poisson, Normal) • Mean, variance, standard deviation • **Laplace Transforms:** Definition, basic transforms • Properties: linearity, shifting, differentiation • Application to differential equations • **Applications:** Quality control, reliability analysis, system dynamics	1	25%

Textbooks:

- Primary: *Higher Engineering Mathematics* — B.S. Grewal
- Primary: *Engineering Mathematics* — NP Bali & Dr. Manish Goyal

Reference books:

- *Advanced Engineering Mathematics* — H.K. Das
- *Numerical Methods* — S.S. Sastry
- *Probability and Statistics for Engineers* — Dr. J. Ravichandran
- *Mathematical Methods* — B.V. Ramana

Online Platforms:

- NPTEL Videos: "Differential Equations for Engineers"
- Khan Academy: Complete probability and statistics
- MIT OCW: "Single Variable Calculus" continuation
  Coursera: "Introduction to Numerical Methods"

COURSE CODE: DITE202

COURSE NAME: DATA STRUCTURES AND ALGORITHMS

Course Objectives:

- To introduce fundamental data structures and their implementation.
- To develop problem-solving skills using appropriate data structures.
- To analyze algorithm efficiency using time and space complexity.
- To prepare students for advanced topics, machine learning, and software development.
- To enable students to implement and apply data structures in real-world scenarios.

Course Outcomes: At the end of the course students shall be able to

CO1	**Explain** the concepts of arrays, linked lists, stacks, and queues.
CO2	**Implement** trees and graphs for hierarchical and network data representation.
C03	**Apply** sorting and searching algorithms to solve computational

	problems.
C04	**Analyze** algorithm efficiency and select appropriate data structures for given problems.

Unit	Content	Credit	Weightage
I	**Introduction to Data Structures & Linear Lists** Introduction: Data types, ADTs, time and space complexity - Arrays: 1D, 2D, operations, applications - Linked Lists: Singly, doubly, circular linked lists - Stacks: Array and linked list implementation, applications - Queues: Simple, circular, priority queues - **Applications:** Expression evaluation, job scheduling, memory management	1	25%
II	**Trees & Hierarchical Data Structures** Trees: Terminology, binary trees, traversal (inorder, preorder, postorder) - Binary Search Trees (BST): Insertion, deletion, searching - AVL Trees: Rotations, balancing - Heaps: Min-heap, max-heap, heap operations - **Applications:** File systems, database indexing, priority queues	1	25%
III	**Graphs & Hashing** Graphs: Terminology, representation (adjacency matrix, list) - Graph traversals: BFS, DFS - Hashing: Hash functions, collision resolution (chaining, open addressing) - **Applications:** Social networks, GPS navigation, spell checking	1	25%
IV	**Sorting, Searching & Algorithm Design** Sorting algorithms: Bubble, selection, insertion, merge, quick, heap sort - Searching algorithms: Linear, binary, interpolation search - Algorithm design techniques: Greedy, divide and conquer, dynamic programming (intro) - **Applications:** Data analysis, AI search algorithms, database query optimization	1	25%

Textbooks:

- *Data Structures and Algorithms in C* — Reema Thareja
- *Data Structures Using C* — Aaron M. Tenenbaum

Reference books:

- *Introduction to Algorithms* — Thomas H. Cormen et al.
- *Data Structures and Algorithm Analysis in C* — Mark Allen Weiss
- *The Algorithm Design Manual* — Steven S. Skiena
- *Data Structures Through C* — Yashavant Kanetkar

Online Platforms:

- NPTEL:
  1. *Data Structures and Algorithms* by Prof. Naveen Garg (IIT Delhi)

2. *Programming, Data Structures and Algorithms* by IIT Madras

- Coursera:
  1. *Data Structures and Algorithms* by University of California, San Diego
  2. *Algorithms Specialization* by Stanford University

PRACTICAL LIST:

- Section A: Linear Data Structures
- Implement stack using array and linked list with push, pop, display.
- Implement queue using array and linked list with enqueue, dequeue, display.
- Implement circular queue with basic operations.
- Implement singly linked list with insertion, deletion, traversal.
- Implement doubly linked list with insertion, deletion, display.
- Implement polynomial addition using linked list.
- Section B: Trees & Heaps
- Implement binary tree traversal (inorder, preorder, postorder) recursively.
- Implement binary search tree with insertion, deletion, searching.
- Implement heap (max-heap) with insertion and deletion.
- Implement AVL tree with rotations (LL, RR, LR, RL).
- Implement expression tree from postfix expression and evaluate it.
- Section C: Graphs & Hashing
- Implement graph using adjacency matrix and perform BFS and DFS.
- Implement graph using adjacency list and perform BFS and DFS.
- Implement hashing with separate chaining for collision resolution.
- Implement hashing with linear probing for collision resolution.
- Find shortest path in unweighted graph using BFS.
- Section D: Sorting & Searching Algorithms
- Implement bubble sort, selection sort, and insertion sort.
- Implement merge sort and quick sort.
- Implement heap sort.
- Implement binary search recursively and iteratively.
- Implement linear search and count occurrences in an array.
- Implement a menu-driven program to compare sorting algorithm performance.

MK UNIVERSITY

PATAN, GUJARAT

ESTABLISHED BY THE GUJARAT GOVT. RECOGNIZED BY UGC UNDER SECTION 2(f) OF UGC ACT,1956

COURSE CODE: DITE203

COURSE NAME: DATABASE MANAGEMENT SYSTEMS

Course Objectives:

- To introduce fundamental concepts of databases, data models, and DBMS architecture.
- To develop skills in designing databases using ER modeling and normalization.
- To enable students to write SQL queries for data definition, manipulation, and control.
- To understand transaction management, concurrency control, and database security.
- To prepare students for real-world database applications in AI and data science.

Course Outcomes: At the end of the course students shall be able to

CO1	**Explain** DBMS architecture, data models, and ER diagrams.
CO2	**Design** normalized relational databases using ER modeling and normalization.
C03	**Write** SQL queries for data definition, manipulation, and retrieval.
C04	**Analyze** transaction management, concurrency control, and database security mechanisms.

Unit	Content	Credit	Weightage
I	**Introduction to DBMS & Data Modeling** Database concepts: Data, database, DBMS, advantages - DBMS architecture: 3-tier architecture, data independence - Data models: Hierarchical, network, relational, ER model - ER diagrams: Entities, attributes, relationships, keys - **Applications:** Banking, inventory, student management systems	1	25%
II	**Relational Database Design & SQL** Relational model: Relations, tuples, attributes, keys - Relational algebra: Operations, selection, projection, join - SQL: DDL, DML, DCL commands - SQL queries: SELECT, WHERE, GROUP BY, HAVING, ORDER BY - Joins: Inner, outer, self, cross joins - **Applications:** Querying real databases, report generation	1	25%
III	**Normalization & Advanced SQL** Normalization: 1NF, 2NF, 3NF, BCNF, functional dependencies - Advanced SQL: Subqueries, views, indexes, sequences - PL/SQL basics: Cursors, procedures, functions, triggers - **Applications:** Data integrity, performance tuning, automation	1	25%
IV	**Transaction Management & Security** Transaction concepts: ACID properties, states - Concurrency control: Locks, two-phase locking, deadlock - Database recovery: Log-based recovery, checkpoints - Database security: Authentication, authorization, encryption - **Applications:** Banking transactions, multi-user systems,	1	25%

	secure databases		

Textbooks:

- *Database System Concepts* — Abraham Silberschatz, Henry F. Korth, S. Sudarshan
- *Fundamentals of Database Systems* — Ramez Elmasri, Shamkant B. Navathe

Reference books:

- *SQL: The Complete Reference* — James R. Groff, Paul N. Weinberg
- *Database Management Systems* — Raghu Ramakrishnan, Johannes Gehrke
- *An Introduction to Database Systems* — C.J. Date
- *Oracle PL/SQL Programming* — Steven Feuerstein

Online Platforms:

- NPTEL:
  1. *Database Management System* by Prof. P.K. Biswas (IIT Kharagpur)
  2. *Introduction to Database Systems* by IIT Madras
- Coursera:
  1. *Databases and SQL for Data Science* by IBM
  2. *SQL for Data Science* by University of California, Davis

PRACTICAL LIST:

Section A: Database Design & ER Modeling

1. Draw ER diagrams for:
   - Library Management System
   - Hospital Management System
   - Online Shopping System
2. Convert ER diagrams into relational schemas.
3. Create tables using DDL commands with primary keys, foreign keys, and constraints.

Section B: SQL Queries – Basic to Intermediate

4. Create a database for Student Management System with tables: Student, Course, Enrollment.
5. Insert sample data into all tables.
6. Write SQL queries to:
   - Retrieve all students enrolled in a specific course.
   - Find the average marks of students.
   - List students with marks above average.
   - Count number of students per course.
7. Implement different types of joins (inner, left, right, full) on the database.

Section C: Advanced SQL & PL/SQL

8. Create views for:
   - Students with marks > 75%
   - Course-wise student count
9. Write SQL queries using:
   - Subqueries (nested, correlated)
   - GROUP BY, HAVING, ORDER BY
10. Create indexes on frequently queried columns.
11. Write a PL/SQL procedure to calculate total marks of a student.
12. Create a trigger to log changes in the Student table.

Section D: Normalization & Transaction Control

13. Take an unnormalized table and normalize it to 3NF.
14. Demonstrate transaction control commands:

- COMMIT, ROLLBACK, SAVEPOINT

15. Simulate concurrent transactions and demonstrate locking.
16. Implement user roles and grant/revoke permissions.
17. Perform backup and recovery operations on a sample database.

COURSE CODE: DITE204

COURSE NAME: WEB TECHNOLOGIES

Course Objectives:

- To introduce the fundamentals of web technologies, internet protocols, and web architecture.
- To develop skills in designing responsive web pages using HTML, CSS, and JavaScript.
- To enable students to create dynamic web applications using front-end and back-end technologies.
- To understand web APIs, AJAX, and integration with databases.
- To prepare students for building web-based AI applications and interactive dashboards.

Course Outcomes: At the end of the course students shall be able to

CO1	**Explain** web architecture, internet protocols, and client-server models.
CO2	**Design** responsive and interactive web pages using HTML, CSS, and JavaScript.
C03	**Develop** dynamic web applications using front-end frameworks and back-end scripting.
C04	**Integrate** databases and web APIs to create data-driven web applications.

Unit	Content	Credit	Weightage
I	**Web Fundamentals & HTML5** Internet basics: WWW, HTTP/HTTPS, DNS, web browsers - Web architecture: Client-server model, static vs dynamic websites - HTML5: Structure, tags, forms, multimedia, semantic elements - HTML5 APIs: Geolocation, local storage, canvas basics - **Applications:** Portfolio websites, forms, simple web pages	1	25%
II	**CSS3 & Responsive Web Design** CSS3: Selectors, box model, positioning, flexbox, grid - Responsive design: Media queries, mobile-first approach - CSS frameworks: Bootstrap basics - Animations and transitions - **Applications:** Responsive layouts, navigation bars, styled forms	1	25%
III	**JavaScript & Front-End Development** JavaScript basics: Syntax, DOM manipulation, events - ES6 features: let/const, arrow functions, promises - AJAX and Fetch API for async requests - Front-end frameworks: Introduction to React.js/Vue.js - **Applications:** Interactive UI, form validation, dynamic content loading	1	25%
IV	**Back-End Basics & Web APIs**	1	25%

	Back-end introduction: Node.js, Express.js basics - RESTful APIs: CRUD operations, JSON handling - Database integration: MongoDB/MySQL with Node.js - Authentication: JWT, session-based auth basics - **Applications:** Simple web apps, API integration, user authentication		

Textbooks:

- *Web Technologies: HTML, JavaScript, PHP, Java, JSP, XML and AJAX* — Uttam K. Roy
- *HTML & CSS: Design and Build Websites* — Jon Duckett

Reference books:

- *Eloquent JavaScript* — Marijn Haverbeke
- *Learning PHP, MySQL & JavaScript* — Robin Nixon
- *Node.js Design Patterns* — Mario Casciaro
- *React: Up & Running* — Stoyan Stefanov

Online Platforms:

- NPTEL:
  1. *Web Technologies* by Prof. D. Janakiram (IIT Madras)
  2. *Introduction to Modern Application Development* by IIT Madras
- Coursera:
  1. *HTML, CSS, and JavaScript for Web Developers* by Johns Hopkins University
  2. *Web Design for Everybody* by University of Michigan

PRACTICAL LIST:

Section A: HTML5 & CSS3

1. Create a personal portfolio website using HTML5 and CSS3.
2. Design a responsive registration form with validation using HTML5 form elements.
3. Build a webpage layout using CSS Grid and Flexbox.
4. Implement a navigation bar with dropdown menus using CSS.
5. Create a photo gallery with hover effects and transitions.

Section B: JavaScript & DOM Manipulation

6. Create a calculator using JavaScript.
7. Build a to-do list application with add, edit, delete, and mark-as-done features.
8. Develop a digital clock with date and time display.
9. Implement form validation (email, password, phone) using JavaScript.
10. Create a slideshow/carousel using JavaScript and CSS.

Section C: Responsive Design & Bootstrap

11. Convert a given webpage into a mobile-responsive layout using media queries.
12. Design a landing page using Bootstrap 5 (navbar, cards, modal, forms).
13. Create a responsive dashboard layout with Bootstrap grid system.
14. Build a blog template with Bootstrap components.

Section D: Back-End Integration & APIs

15. Create a simple REST API using Node.js and Express that returns JSON data.
16. Build a weather app using a public API (e.g., OpenWeatherMap) with Fetch/AJAX.
17. Develop a note-taking app with local storage (CRUD operations).

MK UNIVERSITY

PATAN, GUJARAT

ESTABLISHED BY THE GUJARAT GOVT. **RECOGNIZED BY UGC UNDER SECTION 2(f) OF UGC ACT,1956**

SEMESTER-III

COURSE CODE: DITE301

COURSE NAME: OBJECT ORIENTED PROGRAMMING C++

Course Objectives:

- To introduce the principles and concepts of Object-Oriented Programming (OOP).
- To develop skills in designing, implementing, and testing C++ programs using OOP features.
- To enable students to understand and apply encapsulation, inheritance, polymorphism, and abstraction.
- To prepare a foundation for advanced programming, software development, and system design.
- To foster good programming practices and problem-solving using OOP.

Course Outcomes: At the end of the course students shall be able to

CO1	**Explain** the fundamentals of OOP and the structure of C++ programs.
CO2	**Implement** classes, objects, constructors, and destructors in C++.
C03	**Apply** inheritance, polymorphism, and operator overloading in program design.
C04	**Develop** C++ programs using file handling, templates, and exception handling.

Unit	Content	Credit	Weightage
I	**Introduction to OOP & C++ Basics** • Introduction to OOP: Concepts, advantages over procedural programming • C++ basics: Structure of a C++ program, iostream, namespace • Data types, operators, and control structures in C++ • Functions: Function overloading, default arguments, inline functions • Introduction to classes and objects (basic) • **Applications**: Simple calculator, menu-driven programs	1	25%
II	**Classes, Objects & Constructors** • Classes and objects: Declaration, definition, access specifiers • Member functions: Inside and outside class definition • Constructors: Default, parameterized, copy constructors • Destructors • this pointer • Friend functions and friend classes • **Applications**: Student record system, bank account management	1	25%
III	**Inheritance & Polymorphism**	1	25%

	• Inheritance: Types (single, multiple, multilevel, hierarchical, hybrid) • Access specifiers in inheritance: public, protected, private • Polymorphism: Compile-time and run-time polymorphism • Function overriding and virtual functions • Abstract classes and pure virtual functions • Operator overloading: Unary and binary operators • **Applications**: Payroll system, shape hierarchy, library management		
IV	**Advanced C++ Features** • File handling in C++: if stream, of stream, f stream • Templates: Function templates and class templates • Exception handling: try, catch, throw • Standard Template Library (STL) basics: vector, list, map • Dynamic memory allocation: new and delete operators • **Applications**: File-based data storage, generic programming, error handling	1	25%

Textbooks:

- *Object-Oriented Programming with C++* – E. Balagurusamy
- *Let Us C++* – Yashwant Kanetkar

Reference books:

- *The C++ Programming Language* – Bjarne Stroustrup
- *C++: The Complete Reference* – Herbert Schildt
- *Object-Oriented Programming in C++* – Robert Lafore
- *Programming: Principles and Practice Using C++* – Bjarne Stroustrup

Online Platforms:

- NPTEL
  - *Programming in C++* by Prof. P. P. Chakraborty (IIT Kharagpur)
  - *Object-Oriented Programming in C++* by IIT Bombay
- Coursera
  - *Object-Oriented Programming in C++* by University of London
  - *C++ For C Programmers* by University of California, Santa Cruz

PRACTICAL LIST:

Module I: C++ Basics & Functions

1. Write a C++ program to find the sum and average of three numbers.
2. Write a program to swap two numbers using call by reference.
3. Write a program to check whether a number is prime or not.
4. Write a program to display the Fibonacci series using recursion.
5. Write a program to demonstrate function overloading (add numbers and concatenate strings).
6. Write a program to find the factorial of a number using inline function.

Module II: Classes & Objects

7. Write a program to create a class Student with data members roll_no, name, marks and member functions to input and display data.
8. Write a program to implement a class Circle with member functions to calculate area and circumference.
9. Write a program to demonstrate the use of constructors (default, parameterized, copy).

10. Write a program to show the use of this pointer.
11. Write a program to demonstrate friend function accessing private members of two different classes.

Module III: Inheritance & Polymorphism

12. Write a program to implement single inheritance: class Employee (base) and Manager (derived).
13. Write a program to demonstrate multiple inheritance.
14. Write a program to implement function overriding and use of virtual functions.
15. Write a program to create an abstract class Shape with pure virtual function area().
16. Write a program to overload + operator to add two complex numbers.

Module IV: Advanced C++ Features

17. Write a program to read from a file and count the number of words.
18. Write a program to write student records into a file and display them.
19. Write a program to demonstrate function template for swapping two values of any data type.
20. Write a program to implement exception handling for division by zero.

COURSE CODE: DITE302

COURSE NAME: COMPUTER NETWORKS

Course Objectives:

- To introduce fundamental concepts, architectures, and models of computer networks.
- To develop understanding of network protocols, data communication, and network devices.
- To enable students to design, configure, and troubleshoot basic network setups.
- To prepare students for understanding network-related aspects in AI, IoT, and cloud systems.
- To foster awareness of network security, performance, and ethical considerations.

Course Outcomes: At the end of the course students shall be able to

CO1	**Explain** network models, topologies, and transmission media.
CO2	**Describe** the functions and protocols of data link, network, and transport layers.
C03	**Configure** basic network services and understand routing principles.
C04	**Analyze** network security threats and apply basic security measures.

Unit	Content	Credit	Weightage
I	**Introduction to Computer Networks & Physical Layer** Basics: Components, network types (LAN, MAN, WAN), topologies - Network models: OSI model, TCP/IP model, layers and functions - Transmission media: Guided (twisted pair, coaxial, fiber) and unguided (radio, microwave) - Switching techniques: Circuit, packet, message switching - **Applications:** Home networks, office networks, internet connectivity	1	25%
II	**Data Link Layer & Media Access Control**	1	25%

	Functions of data link layer: Framing, error detection (CRC), flow control - MAC protocols: CSMA/CD, CSMA/CA, Ethernet (IEEE 802.3) - Network devices: Hubs, switches, bridges - VLANs (Virtual LANs) – basic concepts - **Applications:** Ethernet LANs, Wi-Fi networks, network segmentation		
III	**Network Layer & Routing** IPv4 addressing: Classes, subnetting, CIDR - IPv6 basics - Routing algorithms: Static vs dynamic routing, distance vector, link state - Routers and routing tables - ARP, ICMP, DHCP basics - **Applications:** IP addressing, routing in networks, internet connectivity	1	25%
IV	**Transport Layer, Application Layer & Network Security** Transport layer: TCP and UDP, ports, sockets, congestion control - Application layer protocols: HTTP, HTTPS, DNS, SMTP, FTP - Network security: Threats (malware, DoS, phishing), firewalls, encryption basics - Introduction to IoT networks and cloud networking - **Applications:** Web browsing, email, secure communication, IoT devices	1	25%

Textbooks:

- *Data Communications and Networking* — Behrouz A. Forouzan
- *Computer Networks* — Andrew S. Tanenbaum

Reference books:

- *TCP/IP Illustrated* — W. Richard Stevens
- *Computer Networking: A Top-Down Approach* — James F. Kurose & Keith W. Ross
- *Network Security Essentials* — William Stallings
- *CCNA Routing and Switching Complete Study Guide* — Todd Lammle

Online Platforms:

- NPTEL:
  1. *Computer Networks* by Prof. Sujoy Ghosh (IIT Kharagpur)
  2. *Introduction to Computer Networks* by IIT Bombay
- Coursera:
  1. *The Bits and Bytes of Computer Networking* by Google
  2. *Computer Networks* by University of Washington

PRACTICAL LIST:

Section A: Network Basics & Configuration

1. Identify network components and create a simple LAN topology diagram.
2. Configure IP addresses on multiple devices and verify connectivity using ping.
3. Use basic network commands: ipconfig, ifconfig, ping, tracert/traceroute.

4. Create a crossover cable and test connectivity between two PCs.
   Section B: Data Link Layer & Switching
5. Configure a VLAN on a switch and assign ports to VLANs.
6. Analyze Ethernet frames using Wireshark.
7. Simulate CSMA/CD in a network simulator (e.g., Cisco Packet Tracer).
8. Configure port security on a switch.
   Section C: Network Layer & Routing
9. Perform subnetting exercises: Calculate subnets, network addresses, broadcast addresses.
10. Configure static routing between two or more routers.
11. Configure DHCP server on a router and assign IPs dynamically.
12. Analyze IP packets using Wireshark.
    Section D: Transport/Application Layer & Security
13. Configure a simple web server and access it via HTTP/HTTPS.
14. Use nslookup and dig commands for DNS query analysis.
15. Set up a basic firewall rule (using Windows Firewall or iptables basics).
16. Capture and analyze TCP/UDP packets using Wireshark.
17. Mini-project: Design and simulate a small office network with multiple VLANs, routing, and basic security.

COURSE CODE: DITE303

COURSE NAME: OPERATING SYSTEMS

Course Objectives:

- To introduce fundamental concepts, functions, and structures of operating systems.
- To develop understanding of process management, CPU scheduling, and synchronization.
- To enable students to comprehend memory management, file systems, and storage organization.
- To prepare students for understanding system-level programming and modern OS architectures.
- To foster awareness of security, protection, and distributed systems basics.

Course Outcomes: At the end of the course students shall be able to

CO1	**Explain** OS structures, functions, and types of operating systems.
CO2	**Analyze** process management, scheduling algorithms, and synchronization mechanisms.
C03	**Describe** memory management techniques, paging, segmentation, and virtual memory.
C04	**Understand** file systems, storage management, and basic OS security concepts.

Unit	Content	Credit	Weightage
I	**Introduction to OS & System Structures** • Introduction to OS: Definition, objectives, functions • Types of OS: Batch, multiprogramming, time-sharing, real-time, distributed • OS structure: Monolithic, layered, microkernel, modular • System calls and types • Operating system services	1	25%

	• **Applications**: Command-line interface, shell scripting basics		
II	**Process Management & CPU Scheduling** • Process concept: Process states, PCB, operations • Threads: Multithreading models, benefits • CPU scheduling: Criteria, algorithms (FCFS, SJF, Priority, Round Robin) • Process synchronization: Critical section, semaphores, mutex • Deadlocks: Conditions, prevention, avoidance, detection, recovery • **Applications**: Task scheduling in OS, multi-threaded applications	1	25%
III	**Memory Management** • Memory hierarchy: Registers, cache, RAM, secondary storage • Contiguous memory allocation: Fixed and variable partition • Fragmentation: Internal and external • Paging: Basic concept, page table, TLB • Segmentation • Virtual memory: Demand paging, page replacement algorithms (FIFO, LRU, Optimal) • **Applications**: Memory allocation in programs, virtual memory management	1	25%
IV	**File Systems & OS Security** • File concepts: Attributes, operations, types • Directory structure: Single-level, two-level, tree-structured • File allocation methods: Contiguous, linked, indexed • Disk scheduling algorithms: FCFS, SSTF, SCAN, C-SCAN • Protection and security: Access control, authentication, threats, malware • **Applications**: File management in OS, basic disk management, user authentication	1	25%

Textbooks:

- *Operating System Concepts* – Abraham Silberschatz, Peter B. Galvin, Greg Gagne
- *Operating Systems: A Concept-Based Approach* – D. M. Dhamdhere

Reference books:

- *Modern Operating Systems* – Andrew S. Tanenbaum
- *Operating Systems: Principles and Design* – P. C. P. Bhatt
- *Operating Systems: Internals and Design Principles* – William Stallings
- *Operating Systems* – Achyut S. Godbole & Atul Kahate

Online Platforms:

- NPTEL
  - *Operating Systems* by Prof. P. K. Biswas (IIT Kharagpur)
  - *Introduction to Operating Systems* by IIT Madras
- Coursera

- *Operating Systems and You: Becoming a Power User* by Google
- *Introduction to Operating Systems* by University of Colorado

PRACTICAL LIST:

Module I: Introduction & System Calls

1. Write a shell script to display system information (OS version, kernel, memory).
2. Write a shell script to list files and directories in a given path.
3. Write a C program to create a child process using fork() system call.
4. Write a program to demonstrate the use of exec() family of system calls.
5. Write a shell script to automate backup of a directory.

Module II: Process Scheduling & Synchronization

6. Write a C program to simulate FCFS CPU scheduling algorithm.
7. Write a C program to simulate Round Robin CPU scheduling.
8. Write a program to implement SJF (Shortest Job First) scheduling.
9. Write a program to demonstrate producer-consumer problem using semaphores.
10. Write a program to simulate deadlock detection algorithm.

Module III: Memory Management

11. Write a C program to simulate First Fit memory allocation.
12. Write a program to simulate Best Fit memory allocation.
13. Write a program to simulate FIFO page replacement algorithm.
14. Write a program to simulate LRU page replacement algorithm.
15. Write a program to calculate internal and external fragmentation.

Module IV: File Systems & Disk Scheduling

16. Write a C program to simulate single-level directory structure.
17. Write a program to simulate two-level directory structure.
18. Write a program to simulate FCFS disk scheduling algorithm.
19. Write a program to simulate SCAN disk scheduling algorithm.

MK UNIVERSITY

PATAN, GUJARAT

ESTABLISHED BY THE GUJARAT GOVT. RECOGNIZED BY UGC UNDER SECTION 2(f) OF UGC ACT,1956

SEMESTER-IV

COURSE CODE: DITE401

COURSE NAME: CYBER SECURITY FUNDAMENTALS

Course Objectives:

- To introduce the foundational concepts, principles, and importance of cybersecurity.
- To develop an understanding of common cyber threats, vulnerabilities, and attack vectors.
- To enable students to implement basic security measures, cryptography, and access controls.
- To prepare students for roles in network security, ethical hacking, and security policy implementation.
- To foster awareness of legal, ethical, and regulatory aspects of cybersecurity.

Course Outcomes: At the end of the course students shall be able to

CO1	**Explain** core cybersecurity concepts, threats, and attack methodologies.
CO2	**Apply** basic cryptographic techniques and access control mechanisms.
C03	**Analyze** network security vulnerabilities and implement protective measures.
C04	**Evaluate** security policies, ethical hacking principles, and incident response strategies.

Unit	Content	Credit	Weightage
I	**Introduction to Cybersecurity & Threats** • Introduction to cybersecurity: Definition, importance, CIA triad (Confidentiality, Integrity, Availability) • Cyber threats: Malware, phishing, ransomware, DDoS, insider threats • Attack vectors: Social engineering, password attacks, SQL injection, XSS • Vulnerability assessment basics • Cybersecurity roles and certifications • **Applications**: Case studies of recent cyber attacks	1	25%
II	**Cryptography & Access Control** • Cryptography basics: Symmetric vs asymmetric encryption • Common algorithms: AES, DES, RSA • Hash functions: MD5, SHA-1, SHA-256 • Digital signatures and certificates • Access control models: DAC, MAC, RBAC • Authentication methods: Passwords, biometrics, MFA • **Applications**: Secure messaging, password hashing, access control in OS	1	25%
III	**Network Security & Defense** • Network security fundamentals: Firewalls, IDS, IPS • VPNs and secure communication protocols (SSL/TLS) • Wireless security: WEP, WPA, WPA2, WPA3 • Basic packet analysis using Wireshark	1	25%

	• Security policies: BYOD, remote work security • **Applications**: Configuring firewall rules, setting up VPN, analyzing network traffic		
IV	**Ethical Hacking & Incident Response** • Introduction to ethical hacking: Phases, tools, legal aspects • Footprinting, scanning, enumeration • Penetration testing basics • Incident response: Steps, roles, recovery • Cybersecurity laws and regulations: IT Act, GDPR, Cybercrime laws • **Applications**: Simulated penetration testing, incident response planning	1	25%

Textbooks:

- *Cybersecurity Fundamentals* – Rick Howard
- *Computer Security: Principles and Practice* – William Stallings & Lawrie Brown

Reference books:

- *The Web Application Hacker's Handbook* – Dafydd Stuttard & Marcus Pinto
- *Cryptography and Network Security* – Atul Kahate
- *Hacking: The Art of Exploitation* – Jon Erickson
- *Cybersecurity for Dummies* – Joseph Steinberg

Online Platforms:

- NPTEL
  - *Introduction to Cybersecurity* by IIT Madras
  - *Cryptography and Network Security* by IIT Kharagpur
- Coursera
  - *Introduction to Cybersecurity Tools & Cyber Attacks* by IBM
  - *Cybersecurity for Everyone* by University of Maryland

COURSE CODE: DITE402

COURSE NAME: MOBILE APPLICATION DEVELOPMENT

Course Objectives:

- To introduce fundamental concepts of mobile app development and platform ecosystems.
- To develop skills in designing, building, and testing mobile applications for Android and cross-platform environments.
- To enable students to implement UI/UX principles, data storage, and API integration in mobile apps.
- To prepare students for deploying, publishing, and maintaining mobile applications.
- To foster awareness of mobile security, performance optimization, and emerging trends.

Course Outcomes: At the end of the course students shall be able to

CO1	**Explain** mobile app development platforms, architectures, and lifecycle.
CO2	**Design** user interfaces and implement navigation in mobile apps.
C03	**Develop** functional mobile apps with data storage, APIs, and device features.
C04	**Deploy** and test mobile applications on real/virtual devices and

	app stores.

Unit	Content	Credit	Weightage
I	**Introduction to Mobile App Development** • Overview of mobile platforms: Android, iOS, cross-platform • Mobile app architectures: Native, Hybrid, Web apps • Setting up development environment: Android Studio, Flutter/React Native basics • App lifecycle and components (Activity, Fragment in Android) • Introduction to Kotlin/Java for Android or Dart for Flutter • **Applications**: Hello World app, basic calculator app	1	25%
II	**UI/UX Design & Core Components** • UI design principles: Material Design (Android), Human Interface Guidelines (iOS) • Layouts: Linear, Relative, Constraint (Android) or Column/Row (Flutter) • UI components: Buttons, TextViews, EditText, ListView, RecyclerView • Navigation: Intents, Navigation Component, Bottom Navigation • Responsive design for multiple screen sizes • **Applications**: Login screen, profile page, news feed UI	1	25%
III	**Data Handling & APIs** • Data persistence: Shared Preferences, SQLite, Room Database (Android) • File storage: Internal and external storage • Networking: HTTP requests, REST API integration (Retrofit/Volley) • JSON parsing and handling API responses • Using device sensors: Camera, GPS, accelerometer • **Applications**: Weather app, todo list with local storage, news app with API	1	25%
IV	**Advanced Features & Deployment** • Notifications: Local and push notifications (Firebase Cloud Messaging) • Multimedia: Playing audio/video, image handling • App security: Data encryption, secure storage, permissions • Testing: Unit testing, UI testing (Espresso/JUnit) • App deployment: Generating APK, app signing, Google Play Store submission • **Applications**: Music player, location-based reminder, deployed app on Play Store	1	25%

Textbooks:

- *Android Programming: The Big Nerd Ranch Guide* – Bill Phillips & Chris Stewart
- *Flutter Complete Reference* – Alberto Miola

Reference books:

- *Head First Android Development* – Dawn Griffiths & David Griffiths
- *Android App Development for Dummies* – Michael Burton
- *Learning React Native* – Bonnie Eisenman
- *Mobile App Development with Ionic* – Chris Griffith

Online Platforms:

- NPTEL
  - *Mobile Application Development* by IIT Kharagpur
  - *Android App Development* by IIT Madras
- Coursera
  - *Android App Development* by Vanderbilt University
  - *Flutter Development Bootcamp* by London App Brewery

COURSE CODE: DITE403

COURSE NAME: CLOUD COMPUTING BASICS

Course Objectives:

- To introduce the fundamental concepts, models, and architectures of cloud computing.
- To develop skills in deploying, managing, and scaling applications using major cloud platforms.
- To enable students to leverage cloud services for AI/ML model training, storage, and deployment.
- To prepare students for cloud-native development and understanding of security, compliance, and cost management.
- To foster awareness of emerging trends such as serverless computing, containers, and hybrid cloud.

Course Outcomes: At the end of the course students shall be able to

CO1	**Explain** cloud computing models, service models, and deployment models.
CO2	**Deploy** and manage virtual machines, storage, and networking in a cloud environment.
C03	**Utilize** cloud-based AI/ML services for model training and deployment.
C04	**Apply** cloud security best practices and manage cloud costs effectively.

Unit	Content	Credit	Weightage
I	**Introduction to Cloud Computing** What is Cloud Computing? Evolution, characteristics, benefits, and challenges - Cloud service models: IaaS, PaaS, SaaS, FaaS - Cloud deployment models: Public, private, hybrid, community cloud - Key cloud providers: AWS, Azure, GCP overview - **Applications:** Web hosting, data storage, scalable computing	1	25%
II	**Cloud Infrastructure & Services** Virtualization: Hypervisors, virtual machines, containers (Docker basics) - Compute services: EC2 (AWS), VM (Azure), Compute	1	25%

	Engine (GCP) - Storage services: S3, Blob Storage, Cloud Storage - Networking: VPC, load balancers, CDN basics - **Applications:** Deploying web apps, media streaming, backup solutions		
III	**Cloud for AI & DevOps** AI/ML cloud services: AWS SageMaker, Azure ML, GCP AI Platform - Serverless computing: AWS Lambda, Azure Functions - Containers and orchestration: Docker, Kubernetes basics - DevOps in the cloud: CI/CD pipelines (GitHub Actions, Jenkins basics) - **Applications:** Training ML models, auto-scaling apps, microservices	1	25%
IV	**Cloud Security, Management & Trends** Cloud security: Identity and Access Management (IAM), encryption, compliance - Cost management: Pricing models, budgeting, monitoring tools - Migration strategies: Lift-and-shift, re-platforming, cloud-native - Emerging trends: Edge computing, hybrid cloud, sustainable cloud - **Applications:** Secure cloud deployments, cost-optimized architectures, green IT	1	25%

Textbooks:

- *Cloud Computing: Concepts, Technology & Architecture* — Thomas Erl et al.
- *Cloud Computing for Dummies* — Judith Hurwitz et al.

Reference books:

- *Architecting the Cloud* — Michael J. Kavis
- *Cloud Native Transformation* — Pini Reznik et al.
- *AWS Certified Solutions Architect Study Guide* — Ben Piper & David Clinton
- *Cloud Computing Bible* — Barrie Sosinsky

Online Platforms:

- NPTEL:
  1. *Cloud Computing* by Prof. Soumya Kanti Ghosh (IIT Kharagpur)
  2. *Introduction to Cloud Computing* by IIT Madras
- Coursera:
  1. *Cloud Computing Specialization* by University of Illinois
  2. *AWS Fundamentals* by Amazon Web Services

MK UNIVERSITY

PATAN, GUJARAT

ESTABLISHED BY THE GUJARAT GOVT. **RECOGNIZED BY UGC UNDER SECTION 2(f) OF UGC ACT,1956**

COURSE CODE: DITE404

COURSE NAME: DIGITAL ELECTRONICS

Course Objectives:

- To introduce fundamental concepts of digital systems, binary arithmetic, and logic gates.
- To analyze and simplify Boolean functions using algebraic and K-map techniques.
- To design and implement combinational and sequential logic circuits.
- To develop skills in working with digital ICs, flip-flops, counters, and registers.

Course Outcomes: At the end of the course students shall be able to

CO1	Convert number systems, apply Boolean algebra, and implement logic gates.
CO2	Simplify logic expressions and design combinational circuits (adders, multiplexers, encoders).
C03	Analyze and design sequential circuits using flip-flops, counters, and registers.
C04	Interface digital circuits with basic real-world applications.

Unit	Content	Credit	Weightage
I	**Number Systems & Logic Gates** • Number systems: binary, octal, hexadecimal, conversions. • Binary arithmetic: addition, subtraction (1's and 2's complement), multiplication. • Logic gates: AND, OR, NOT, NAND, NOR, XOR, XNOR – truth tables, symbols, ICs. • Boolean algebra: laws, De Morgan's theorem. • Logic families: TTL and CMOS characteristics, comparison.	1	25%
II	**Combinational Logic Design** • Canonical forms: SOP and POS. • Karnaugh map (up to 4 variables) – simplification of Boolean functions. • Arithmetic circuits: half adder, full adder, half subtractor, full subtractor. • Multiplexers and demultiplexers (4:1, 8:1). • Encoders and decoders (binary to decimal, BCD to 7-segment). • Parity generators and checkers.	1	25%
III	**Sequential Logic Fundamentals** • Latches: SR latch. • Flip-flops: SR, JK, D, T – truth tables, excitation tables, race-around condition in JK FF. • Master-slave JK flip-flop. • Registers: SISO, SIPO, PISO, PIPO, shift registers (bidirectional). • Counters: asynchronous (ripple) counters, synchronous	1	25%

	counters, up/down counters.		
IV	**Advanced Sequential Circuits & Applications** • Mod-N counters, design using flip-flops. • Ring counter, Johnson counter. • Introduction to finite state machines (Mealy and Moore models). • Memory devices: ROM, RAM (basic concepts). • Introduction to PLDs: PROM, PAL, PLA (basic block diagrams only). • Simple digital applications: digital clock, traffic light control (concept only).	1	25%

Textbooks:

- *Digital Logic and Computer Design* – M. Morris Mano
- *Modern Digital Electronics* – R.P. Jain
- *Digital Electronics* – S. Salivahanan & S. Arivazhagan

Reference books:

- *Fundamentals of Digital Circuits* – A. Anand Kumar
- *Digital Design* – M. Morris Mano & Michael D. Ciletti
- *Digital Systems: Principles and Applications* – Ronald J. Tocci

Online Platforms:

- NPTEL: *Digital Circuits* by Prof. S. Srinivasan

PRACTICAL LIST:

- Verification of Truth Tables of Basic and Universal Logic Gates.
- Implementation of Boolean Functions Using Logic Gates.
- Design and Verification of Half/Full Adder and Subtractor Circuits.
- Design and Testing of 4-bit Parity Generator/Checker.
- Implementation and Testing of Multiplexer (4:1) and Demultiplexer (1:4).
- Design and Testing of Encoder (8:3) and Decoder (3:8).
- Study and Testing of Flip-Flops (SR, JK, D, T).
- Design and Testing of 4-bit Shift Register (SISO, SIPO).
- Design and Testing of Asynchronous (Ripple) Up/Down Counter.
- Design and Testing of Synchronous Mod-10 Counter.
- Interfacing BCD to 7-Segment Decoder and Display.
- Simulation of Digital Circuits Using Software (Logisim/Proteus).
- Mini-Project: Simple Digital Clock/Traffic Light Controller.

MK UNIVERSITY

PATAN, GUJARAT

ESTABLISHED BY THE GUJARAT GOVT. RECOGNIZED BY UGC UNDER SECTION 2(f) OF UGC ACT,1956

SEMESTER-V

COURSE CODE: DITE501

COURSE NAME: DATA SCIENCE FUNDAMENTALS

Course Objectives:

- To introduce the fundamental concepts and workflow of data science
- To develop skills in data manipulation, cleaning, and analysis using Python
- To apply statistical and machine learning techniques to IT datasets
- To enable effective data storytelling through visualization and reporting
- To prepare students for entry-level data science roles in IT industry

Course Outcomes: At the end of the course students shall be able to

CO1	Understand the data science lifecycle and its applications in IT domains
CO2	Perform data collection, cleaning, and exploratory data analysis using Python
C03	Apply statistical methods and machine learning algorithms for data analysis
C04	Create data visualizations and communicate insights effectively

Unit	Content	Credit	Weightage
I	**Introduction to Data Science & Python Ecosystem** • **Data Science Overview:** Definition, evolution, and applications in IT • **Data Science Workflow:** CRISP-DM methodology (Business Understanding to Deployment) • **Python for Data Science:** NumPy, Pandas, Matplotlib, Seaborn basics • **Data Types & Sources:** Structured vs unstructured data, databases, APIs, web scraping basics • **Data Collection Methods:** Reading files (CSV, Excel, JSON), database queries, APIs • **Setting up Development Environment:** Jupyter Notebook, Google Colab, VS Code • **IT Applications:** Log analysis, user behavior tracking, system performance data	1	25%

II	**Data Preparation & Exploratory Data Analysis (EDA)**
	• **Data Cleaning:**
	  ○ Handling missing values (imputation techniques)
	  ○ Outlier detection and treatment
	  ○ Data type conversions
	  ○ Duplicate removal
	• **Data Transformation:**
	  ○ Normalization and standardization
		1	25%

	○ Encoding categorical variables (Label, One-Hot) ○ Feature scaling • **Exploratory Data Analysis (EDA):** ○ Descriptive statistics ○ Univariate analysis (distributions) ○ Bivariate analysis (correlations) ○ Multivariate analysis • **Data Quality Assessment:** Completeness, accuracy, consistency checks • **Case Study:** Preparing IT server logs for analysis		
III	**Statistical Foundations & Machine Learning Basics** • **Statistical Concepts Review:** ○ Probability distributions (Normal, Binomial, Poisson) ○ Hypothesis testing (t-test, chi-square) ○ Confidence intervals ○ Correlation and covariance • **Supervised Learning:** ○ Regression: Linear Regression, evaluation metrics (MSE, RMSE, R^2) ○ Classification: Logistic Regression, evaluation metrics (Accuracy, Precision, Recall, F1) ○ Decision Trees basics ○ Model evaluation: Train-test split, cross-validation • **Unsupervised Learning:** ○ Clustering: K-Means algorithm ○ Dimensionality Reduction: PCA basics • **Model Selection:** Bias-variance tradeoff, overfitting vs underfitting • **IT Applications:** Customer segmentation, sales forecasting, anomaly detection	1	25%
IV	**Data Visualization & Communication** • **Visualization Principles:** Perception, chart selection, color theory • **Static Visualizations:** ○ Matplotlib customization ○ Statistical plots with Seaborn ○ Geospatial plotting basics • **Interactive Visualizations:** Introduction to Plotly • **Dashboard Creation:** Introduction to Streamlit or Dash • **Data Storytelling:** Creating narratives with data • **Report Generation:** Jupyter to PDF/HTML, presentation skills	1	25%

	• **Ethical Considerations:** Data privacy, bias in data, responsible AI		

Textbooks:

- "Python for Data Analysis" (2nd Edition) – Wes McKinney
- "Introduction to Machine Learning with Python" – Andreas C. Müller & Sarah Guido
- "Practical Statistics for Data Scientists" – Peter Bruce & Andrew Bruce
- "Storytelling with Data" – Cole Nussbaumer Knaflic

Reference books:

- "The Hundred-Page Machine Learning Book" – Andriy Burkov
- "Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow" – Aurélien Géron
- "Data Science from Scratch" – Joel Grus
- "Data Visualization: A Practical Introduction" – Kieran Healy

Online Platforms:

- Data Camp – Interactive Python & data science courses
- Kaggle Learn – Free data science micro-courses
- Google Colab – Cloud-based Python notebooks with free GPU
- Jupyter Notebooks – Local development environment

PRACTICAL LIST:

1. Python Data Science Environment Setup
   - Install Anaconda and create virtual environment
   - Set up Jupyter Notebook/Google Colab
   - Import essential libraries (NumPy, Pandas, Matplotlib)
2. Basic Data Manipulation with Pandas
   - Load datasets from CSV, Excel files
   - Perform basic operations (head, tail, info, describe)
   - Select, filter, and sort data
3. Data Collection from Multiple Sources
   - Read data from SQL database (SQLite/MySQL)
   - Extract data from API (using requests library)
   - Basic web scraping with BeautifulSoup
4. Data Cleaning Pipeline
   - Handle missing values using different strategies
   - Detect and treat outliers using IQR method
   - Convert data types and format dates
5. Exploratory Data Analysis (EDA)
   - Generate descriptive statistics
   - Create distribution plots for numerical features
   - Analyze categorical variables using count plots
6. Correlation Analysis
   - Compute correlation matrix
   - Visualize correlations using heatmaps
   - Identify highly correlated features
7. Feature Engineering Basics
   - Create new features from existing ones
   - Apply one-hot encoding to categorical variables
   - Normalize numerical features

8. Statistical Analysis
    - Perform hypothesis testing (t-test)
    - Calculate confidence intervals
    - Analyze probability distributions
9. Linear Regression Implementation
    - Build simple linear regression model
    - Evaluate using MSE, RMSE, R^2
    - Visualize regression line and residuals
10. Classification with Logistic Regression
    - Implement binary classification
    - Create confusion matrix
    - Calculate precision, recall, F1-score
11. Clustering with K-Means
    - Apply K-Means clustering
    - Determine optimal k using elbow method
    - Visualize clusters

COURSE CODE: DITE502

COURSE NAME: INTERNET OF THINGS (IOT)

Course Objectives:

- To introduce fundamental concepts, architectures, and applications of IoT.
- To develop skills in designing, prototyping, and implementing IoT systems using sensors, microcontrollers, and communication modules.
- To enable students to integrate IoT devices with cloud platforms and data analytics tools.
- To prepare students for roles in smart systems, industrial automation, and embedded IoT solutions.
- To foster awareness of IoT security, privacy, and ethical considerations.

Course Outcomes: At the end of the course students shall be able to

CO1	**Explain** IoT architecture, components, protocols, and application domains.
CO2	**Design** and implement basic IoT systems using microcontrollers and sensors.
C03	**Integrate** IoT devices with cloud platforms for data storage and visualization.
C04	**Develop** secure and scalable IoT solutions with real-world applications.

Unit	Content	Credit	Weightage
I	**Introduction to IoT & Architecture** • Introduction to IoT: Definition, evolution, characteristics • IoT architecture: 3-layer and 5-layer models • IoT components: Sensors, actuators, microcontrollers, gateways, cloud • IoT communication models: Request-Response, Publish-Subscribe, Push-Pull • IoT application domains: Smart home, healthcare, agriculture, industrial automation	1	25%

	• **Applications**: Case studies of IoT in daily life and industry		
II	**IoT Hardware & Communication Protocols** • Microcontrollers: Arduino, ESP8266/ESP32, Raspberry Pi • Sensors and actuators: Temperature, humidity, motion, IR, relays, motors • Communication protocols: Wi-Fi, Bluetooth, Zigbee, LoRa, MQTT, CoAP • Embedded programming basics using Arduino IDE • IoT prototyping platforms: Node-RED, Blynk • **Applications**: Weather monitoring station, smart lighting system	1	25%
III	**IoT Data Handling & Cloud Integration** • IoT data acquisition and preprocessing • Cloud platforms for IoT: AWS IoT, Google Cloud IoT, Thing Speak, Adafruit IO • Data storage and visualization on cloud dashboards • REST APIs and MQTT brokers for IoT communication • Edge computing basics • **Applications**: Cloud-based sensor monitoring, real-time data logging	1	25%
IV	**IoT Applications & Security** • Smart systems: Home automation, smart agriculture, health monitoring • Industrial IoT (IIoT): Predictive maintenance, asset tracking • IoT security challenges: Device security, network security, data privacy • Security protocols: TLS/SSL, encryption, authentication • Ethical and legal considerations in IoT • **Applications**: Secure smart door lock, IoT-based irrigation system	1	25%

Textbooks:

- *Internet of Things: A Hands-On Approach* – Arshdeep Bahga & Vijay Madisetti
- *Building the Internet of Things* – Maciej Kranz

Reference books:

- *IoT Fundamentals: Networking Technologies, Protocols, and Use Cases* – David Hanes et al.
- *Getting Started with the Internet of Things* – Cuno Pfister
- *Practical Internet of Things Security* – Brian Russell & Drew Van Duren
- *IoT and Edge Computing for Architects* – Perry Lea

Online Platforms:

- NPTEL
  - *Introduction to Internet of Things* by Prof. Sudip Misra (IIT Kharagpur)
  - *IoT and Cloud Computing* by IIT Bombay
- Coursera
  - *An Introduction to Programming the Internet of Things (IoT)* by University of California, Irvine

- *IoT (Internet of Things) Specialization* by University of Illinois

PRACTICAL LIST:

Module I: Introduction & Architecture

1. Research and present a case study on an IoT application (smart home/healthcare).
2. Draw and explain the 5-layer IoT architecture with real-world components.
3. Identify IoT sensors and actuators used in daily appliances.
4. Prepare a report on IoT communication models with examples.

Module II: IoT Hardware & Communication

5. Interface a temperature sensor (DHT11) with Arduino and display readings on the serial monitor.
6. Control an LED using a smartphone via Bluetooth (HC-05 module).
7. Send sensor data to a mobile app using Blynk platform.
8. Implement MQTT protocol to publish and subscribe sensor data between two devices.
9. Build a motion-activated security light using PIR sensor and relay.

Module III: Cloud Integration & Data Handling

10. Send temperature and humidity data to ThingSpeak cloud and visualize it.
11. Create a cloud dashboard (Adafruit IO) to monitor sensor data remotely.
12. Use Node-RED to create a flow for processing and visualizing IoT data.
13. Store IoT sensor data in a cloud database (Firebase/Google Sheets).
14. Implement a simple alert system (email/SMS) when sensor thresholds are crossed.

Module IV: Applications & Security

15. Develop a smart home system with multiple sensors and remote control.
16. Build an IoT-based plant watering system with soil moisture sensor.
17. Implement basic encryption for data transmitted from an IoT device.
18. Design an access control system using RFID/NFC and IoT.
19. Mini-Project: Create an end-to-end IoT solution (e.g., smart weather station, health monitor, smart parking system).

COURSE CODE: DITE503

COURSE NAME: DEVOPS FUNDAMENTALS

Course Objectives:

- To introduce DevOps philosophy and its business value in modern IT
- To develop skills in automation, continuous integration, and deployment
- To enable containerization and orchestration using Docker and Kubernetes basics
- To implement infrastructure as code and configuration management
- To prepare students for entry-level DevOps roles and certifications

Course Outcomes: At the end of the course students shall be able to

CO1	Understand DevOps culture, principles, and its impact on IT delivery lifecycle
CO2	Implement version control, CI/CD pipelines, and automated testing
C03	Configure and manage containers using Docker and container orchestration basics
C04	Deploy infrastructure as code using configuration management tools

Unit	Content	Credit	Weightage
I	**Introduction to DevOps & Version Control** • **DevOps Fundamentals:** o What is DevOps? History and evolution o DevOps culture: Collaboration, automation, measurement, sharing o Benefits and challenges of DevOps adoption o DevOps vs Agile vs Traditional IT • **Version Control Systems:** o Git fundamentals: Repositories, commits, branches, merges o Git workflows: Feature Branch, GitFlow, GitHub Flow o Git commands: clone, add, commit, push, pull, merge, rebase o Resolving merge conflicts • **Git Hosting Platforms:** o GitHub, GitLab, Bitbucket overview o Creating repositories, managing branches o Pull requests and code review process	1	25%
II	**Continuous Integration & Deployment (CI/CD)** • **CI/CD Pipeline Concepts:** o Continuous Integration: Build automation, testing o Continuous Delivery vs Continuous Deployment o Pipeline stages: Build, Test, Deploy • **Build Tools & Package Management:** o Build tools: Maven, Gradle (Java), npm (Node.js), pip (Python) o Artifact repositories: JFrog Artifactory, Nexus • **CI/CD Tools:** o Jenkins: Installation, configuration, pipeline creation o GitHub Actions: Workflow basics o GitLab CI/CD pipelines	1	25%
III	**Containerization with Docker** • **Containerization Fundamentals:** o What are containers? Difference from VMs o Docker architecture: Docker Engine, Images, Containers o Benefits of containerization • **Docker Commands & Operations:** o Docker CLI: run, build, push, pull, exec, logs o Dockerfile creation: Best practices o Multi-stage builds o Docker volumes and networking • **Docker Compose:**	1	25%

	○ Defining multi-container applications ○ Compose file structure ○ Service definitions, networks, volumes • **Docker Registries:** ○ Docker Hub: Public and private repositories ○ Private registry setup (basic) • **Container Security Basics:** ○ Image scanning, vulnerability management ○ Best practices for secure containers		
IV	**Monitoring, Logging & Cloud DevOps** • **Monitoring & Observability:** ○ Monitoring vs Observability ○ Metrics, Logs, Traces (Three Pillars) ○ Basic monitoring with Prometheus and Grafana ○ Application Performance Monitoring (APM) basics • **Centralized Logging:** ○ ELK Stack introduction (Elasticsearch, Logstash, Kibana) ○ Log aggregation and analysis • **Cloud DevOps:** ○ DevOps on Cloud platforms: AWS, Azure, GCP ○ Cloud-native services for DevOps	1	25%

Textbooks:

- "The DevOps Handbook: How to Create World-Class Agility, Reliability, & Security in Technology Organizations" – Gene Kim, Jez Humble, Patrick Debois, John Willis
- "The Phoenix Project: A Novel About IT, DevOps, and Helping Your Business Win" – Gene Kim, Kevin Behr, George Spafford
- "Docker Deep Dive" – Nigel Poulton
- "Learning DevOps: Continuously Deliver Better Software" – Mikael Krief

Reference books:

- "Accelerate: The Science of Lean Software and DevOps" – Nicole Forsgren, Jez Humble, Gene Kim
- "Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation" – Jez Humble, David Farley
- "Infrastructure as Code: Managing Servers in the Cloud" – Kief Morris
- "Site Reliability Engineering: How Google Runs Production Systems" – Betsy Beyer, Chris Jones, Jennifer Petoff, Niall Richard Murphy

MK UNIVERSITY

PATAN, GUJARAT

ESTABLISHED BY THE GUJARAT GOVT. RECOGNIZED BY UGC UNDER SECTION 2(f) OF UGC ACT,1956

COURSE CODE: DITE504

COURSE NAME: ENTERPRENEURSHIP IN IT

Course Objectives:

- To cultivate entrepreneurial mindset and identify opportunities in IT industry
- To teach methodologies for validating and developing IT business ideas
- To develop skills in business planning, financial management, and marketing for IT ventures
- To provide understanding of legal, funding, and operational requirements for IT startups
- To prepare students to launch and manage IT-based entrepreneurial ventures

Course Outcomes: At the end of the course students shall be able to

CO1	Understand entrepreneurial mindset, opportunities, and challenges in IT sector
CO2	Develop and validate IT-based business ideas using lean startup methodology
C03	Create business models, financial projections, and marketing strategies for IT ventures
C04	Navigate legal, funding, and operational aspects of IT startups

Unit	Content	Credit	Weightage

I	**Entrepreneurial Mindset & IT Opportunities**
	• **Entrepreneurship Fundamentals:**
	  ○ What is entrepreneurship? Traits of successful entrepreneurs
	  ○ Entrepreneur vs. Intrapreneur
	  ○ Myths and realities of entrepreneurship
	• **IT Industry Landscape:**
	  ○ Current trends: AI, Cloud, IoT, Cybersecurity, SaaS
	  ○ Emerging opportunities in digital transformation
	  ○ Niche markets in IT: EdTech, FinTech, HealthTech, AgriTech
	• **Idea Generation & Opportunity Recognition:**
	  ○ Techniques: Problem identification, trend analysis, gap analysis
	  ○ Sources of IT business ideas
	  ○ Idea validation methods
	• **Entrepreneurial Ecosystem:**
	  ○ Startup ecosystem components: Incubators, accelerators, investors
	  ○ Government initiatives: Startup India, Make in India
	  ○ Role of IT parks and innovation hubs
		1	25%

II	**Business Planning & Model Development** • **Lean Startup Methodology:**	1	25%

	○ Build-Measure-Learn feedback loop ○ Minimum Viable Product (MVP) concept ○ Customer development process • **Business Model Canvas (BMC):** ○ Nine components of BMC ○ Applying BMC to IT businesses ○ Value Proposition Canvas • **Market Research & Analysis:** ○ Target market identification ○ Competitive analysis in IT sector ○ Customer persona development • **Business Plan Development:** ○ Components of a business plan ○ Executive summary, company description ○ Market analysis, organization structure		
III	**Financial, Legal & Operational Aspects** • **Financial Planning & Management:** ○ Startup costs estimation ○ Revenue models for IT businesses (Subscription, Freemium, Transaction-based) ○ Basic financial statements: Income Statement, Balance Sheet, Cash Flow ○ Break-even analysis • **Funding Options:** ○ Bootstrapping ○ Angel investors and venture capital ○ Government schemes and grants ○ Crowdfunding platforms • **Legal & Regulatory Framework:** ○ Business registration: Proprietorship, Partnership, LLP, Private Limited ○ Intellectual Property in IT: Copyrights, Trademarks, Patents ○ IT Act compliance ○ Data privacy regulations (GDPR basics) • **Operational Planning:** ○ Team building and hiring for IT startups ○ Technology stack selection ○ Vendor and partner management ○ Quality assurance and customer support	1	25%
IV	**Marketing, Growth & Pitch Development** • **Digital Marketing for IT Startups:** ○ Content marketing, SEO, SEM ○ Social media strategies ○ Growth hacking techniques • **Sales & Customer Acquisition:** ○ Sales funnel for IT products/services	1	25%

	○ B2B vs B2C sales strategies ○ Customer retention strategies • **Scaling & Growth Strategies:** ○ Scaling technology infrastructure ○ International expansion considerations ○ Strategic partnerships and alliances • **Risk Management:** ○ Identifying startup risks ○ Mitigation strategies ○ Crisis management basics		

Textbooks:

- "The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses" – Eric Ries
- "Zero to One: Notes on Startups, or How to Build the Future" – Peter Thiel with Blake Masters
- "Business Model Generation: A Handbook for Visionaries, Game Changers, and Challengers" – Alexander Osterwalder & Yves Pigneur
- "The $100 Startup: Reinvent the Way You Make a Living, Do What You Love, and Create a New Future" – Chris Guillebeau

Reference books:

- "Startup India: How India Became a Hub for Innovators and Entrepreneurs" – D.D. Mishra
- "The Innovator's Dilemma: When New Technologies Cause Great Firms to Fail" – Clayton M. Christensen
- "Hooked: How to Build Habit-Forming Products" – Nir Eyal
- "Traction: How Any Startup Can Achieve Explosive Customer Growth" – Gabriel Weinberg & Justin Mares
- "The Hard Thing About Hard Things: Building a Business When There Are No Easy Answers" – Ben Horowitz

Online platforms:

- Coursera:
  - "Entrepreneurship Specialization" by Wharton School
  - "Startup Entrepreneurship" by Technion - Israel Institute of Technology
- edX:
  - "Entrepreneurship in Emerging Economies" by Harvard University
  - "How to Build a Startup" by Udacity
- Udemy: Various entrepreneurship and startup courses
- SWAYAM/NPTEL:
  - "Entrepreneurship" by IIT Kharagpur
  - "New Venture Planning" by IIM Bangalore

MK UNIVERSITY

PATAN, GUJARAT

ESTABLISHED BY THE GUJARAT GOVT. RECOGNIZED BY UGC UNDER SECTION 2(f) OF UGC ACT,1956

COURSE CODE: DITE601

COURSE NAME: ARTIFICIAL INTELLIGENCE AND ML BASICS

Course Objectives:

- To introduce core concepts of Artificial Intelligence and Machine Learning
- To develop practical skills in implementing ML algorithms using Python
- To enable understanding of model evaluation and selection techniques
- To provide exposure to real-world AI applications in IT domains
- To foster awareness of ethical implications and responsible AI practices

Course Outcomes: At the end of the course students shall be able to

CO1	Understand fundamental concepts, history, and applications of AI & ML
CO2	Implement supervised learning algorithms for classification and regression
C03	Apply unsupervised learning techniques for clustering and dimensionality reduction
C04	Build, evaluate, and deploy basic AI/ML models using Python libraries

Unit	Content	Credit	Weightage

I	**Introduction to AI & ML Fundamentals**
	• **AI Fundamentals:**
	  ○ What is Artificial Intelligence? History and evolution
	  ○ Types of AI: Narrow AI vs General AI vs Superintelligent AI
	  ○ AI vs Machine Learning vs Deep Learning
	  ○ Applications in IT: Chatbots, Recommendation Systems, Fraud Detection
	• **ML Basics:**
	  ○ Types of Machine Learning: Supervised, Unsupervised, Reinforcement
	  ○ ML workflow: Data → Model → Evaluation → Deployment
	  ○ Common terminology: Features, Labels, Training, Testing, Prediction
	• **Python for AI/ML:**
	  ○ Essential libraries: NumPy, Pandas, Matplotlib, Scikit-learn
	  ○ Setting up AI development environment (Jupyter/Colab)
	  ○ Basic data manipulation for ML
	• **Mathematics for ML (Basics):**
	  ○ Linear algebra essentials: Vectors, Matrices
		1	25%

	○ Statistics: Mean, Variance, Standard Deviation ○ Probability basics for ML		
II	**Supervised Learning Algorithms** • **Regression Algorithms:** ○ Linear Regression: Concept, implementation, interpretation ○ Evaluation metrics: MSE, RMSE, R^2 ○ Polynomial regression basics • **Classification Algorithms:** ○ Logistic Regression: Binary classification ○ Decision Trees: Construction, pruning, interpretation ○ Random Forest: Ensemble method basics ○ k-Nearest Neighbors (k-NN) • **Model Evaluation & Selection:** ○ Train-Test split ○ Cross-validation techniques ○ Performance metrics: Accuracy, Precision, Recall, F1-Score, ROC-AUC ○ Confusion matrix interpretation • **Feature Engineering:** ○ Handling categorical variables: One-hot encoding, Label encoding ○ Feature scaling: Normalization, Standardization ○ Feature selection basics	1	25%
III	**Unsupervised Learning & Advanced Topics** • **Unsupervised Learning:** ○ Clustering: K-Means algorithm, elbow method ○ Hierarchical clustering basics ○ Evaluation: Silhouette score • **Dimensionality Reduction:** ○ Principal Component Analysis (PCA) basics ○ t-SNE for visualization • **Introduction to Neural Networks:** ○ Biological vs Artificial neurons ○ Perceptron model ○ Activation functions: Sigmoid, ReLU, Tanh	1	25%
IV	**Model Deployment & Ethical AI** • **Model Deployment Basics:** ○ Saving and loading models (Pickle, Joblib) ○ Creating simple ML API with Flask/FastAPI ○ Introduction to cloud AI services (AWS SageMaker, Google AI Platform basics) • **AI in IT Applications:**	1	25%

	○ Chatbot development basics ○ Recommendation systems: Collaborative filtering ○ Anomaly detection in IT systems • **Ethical AI & Responsible ML:** ○ Bias in AI: Data bias, algorithmic bias ○ Fairness, Accountability, Transparency (FAT)		

Textbooks:

- "Python Machine Learning" (3rd Edition) – Sebastian Raschka & Vahid Mirjalili
- "Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow" (3rd Edition) – Aurélien Géron
- "Artificial Intelligence: A Modern Approach" (4th Edition) – Stuart Russell & Peter Norvig
- "The Hundred-Page Machine Learning Book" – Andriy Burkov

Reference books:

- "Pattern Recognition and Machine Learning" – Christopher M. Bishop
- "Introduction to Machine Learning with Python" – Andreas C. Müller & Sarah Guido
- "Machine Learning Yearning" – Andrew Ng
- "AI and Machine Learning for Coders" – Laurence Moroney
- "Ethics of Artificial Intelligence and Robotics" – Vincent C. Müller

Online platforms:

- Coursera:
  - "Machine Learning" by Andrew Ng (Stanford)
  - "AI For Everyone" by Andrew Ng
- edX:
  - "Introduction to Artificial Intelligence" by IBM
  - "Machine Learning Fundamentals" by University of California, San Diego

PRACTICAL LIST:

1. Python for ML Setup
   - Install Python and essential libraries
   - Set up Jupyter Notebook/Google Colab
   - Basic NumPy and Pandas operations
2. Exploratory Data Analysis (EDA)
   - Load and explore dataset (Iris/Titanic)
   - Generate descriptive statistics
   - Create visualizations: histograms, scatter plots, box plots
3. Data Preprocessing Pipeline
   - Handle missing values
   - Encode categorical variables
   - Scale numerical features
   - Split data into train/test sets
4. Linear Regression Implementation
   - Predict housing prices (Boston Housing dataset)
   - Implement simple and multiple linear regression
   - Evaluate using MSE, RMSE, R^2
   - Visualize predictions vs actual values

5. Logistic Regression for Classification
    - Binary classification on Titanic dataset
    - Implement logistic regression
    - Create confusion matrix
    - Calculate accuracy, precision, recall
6. Decision Tree Classifier
    - Classify Iris flower species
    - Visualize decision tree
    - Tune hyperparameters (max_depth, min_samples_split)
7. Random Forest Implementation
    - Compare with single decision tree
    - Analyze feature importance
    - Tune number of estimators
8. k-NN Classifier
    - Implement k-Nearest Neighbors
    - Find optimal k value
9. K-Means Clustering
    - Customer segmentation (Mall dataset)
    - Determine optimal clusters using elbow method
    - Visualize clusters
    - Calculate silhouette score
10. PCA for Dimensionality Reduction
    - Apply PCA on high-dimensional data
    - Visualize data in 2D/3D
    - Analyze explained variance ratio

COURSE CODE: DITE602

COURSE NAME: SOFTWARE ENGINEERING

Course Objectives:

- To introduce fundamental concepts, principles, and processes of software engineering.
- To develop skills in software requirements analysis, design, and modeling techniques.
- To enable students to understand software development methodologies and testing strategies.
- To prepare students for roles in software project planning, management, and quality assurance.
- To foster awareness of software maintenance, reusability, and ethical practices in software development.

Course Outcomes: At the end of the course students shall be able to

CO1	**Explain** software engineering concepts, life cycle models, and requirement engineering.
CO2	**Apply** software design principles, modeling techniques, and architectural styles.
C03	**Implement** software testing strategies, verification, and validation methods.
C04	**Analyze** software project management, maintenance, and quality assurance practices.

MK UNIVERSITY

PATAN, GUJARAT

ESTABLISHED BY THE GUJARAT GOVT. RECOGNIZED BY UGC UNDER SECTION 2(f) OF UGC ACT,1956

Unit	Content	Credit	Weightage
I	**Introduction to SE & Process Models** • Introduction to Software Engineering: Definition, importance, challenges • Software Development Life Cycle (SDLC) phases • Process models: Waterfall, Iterative, Incremental, Spiral, V-Model • Agile methodologies: Scrum, Kanban, Extreme Programming (XP) • Software process metrics and improvement • **Applications**: Case studies of software projects	1	25%
II	**Requirement Engineering & Design** • Requirements engineering: Elicitation, analysis, specification, validation • Types of requirements: Functional and non-functional • Use case modeling and diagrams • Software design: Architectural design, modular design, cohesion, coupling • Design principles: Abstraction, decomposition, modularity, information hiding • UML diagrams: Class, sequence, activity, state diagrams • **Applications**: SRS preparation, design document creation	1	25%
III	**Software Testing & Quality Assurance** • Software testing fundamentals: Objectives, principles, levels • Testing strategies: Unit testing, integration testing, system testing • Testing techniques: Black-box, white-box, regression, acceptance • Software quality assurance: Standards, quality metrics, reviews, inspections • Introduction to automated testing tools • **Applications**: Test case design, bug tracking, quality metrics calculation	1	25%
IV	**Project Management & Maintenance** • Software project management: Planning, scheduling, risk management • Cost estimation techniques: COCOMO, Function Point Analysis • Software configuration management: Version control, change management • Software maintenance: Types, maintenance models, re-engineering • Software ethics and professional practices • **Applications**: Project planning using Gantt charts, version control with Git	1	25%

Textbooks:

- *Software Engineering: A Practitioner's Approach* – Roger S. Pressman
- *Software Engineering* – Ian Sommerville

Reference books:

- *Fundamentals of Software Engineering* – Rajib Mall
- *Software Engineering: Concepts and Practices* – Ugrasen Suman
- *Software Engineering Principles and Practice* – Hans van Vliet
- *Agile Software Development: Principles, Patterns, and Practices* – Robert C. Martin

Online Platforms:

- NPTEL
  - *Software Engineering* by Prof. Rajib Mall (IIT Kharagpur)
  - *Software Project Management* by IIT Bombay
- Coursera
  - *Software Development Lifecycle* by University of Minnesota
  - *Introduction to Software Engineering* by IBM

PRACTICAL LIST:

Module I: Process Models & Agile

1. Prepare a report comparing Waterfall and Agile methodologies with examples.
2. Create a case study analysis of a software project using the Spiral model.
3. Develop a basic Scrum board (digital/paper) for a small project.
4. Draw a V-Model diagram and explain each phase with examples.

Module II: Requirement Analysis & Design

5. Write a Software Requirements Specification (SRS) for a Library Management System.
6. Create use case diagrams for an Online Shopping System.
7. Design class diagrams for a Hospital Management System.
8. Create sequence diagrams for ATM transaction processing.
9. Prepare a modular design structure for a Student Information System.

Module III: Testing & Quality Assurance

10. Write test cases for a login module (black-box testing).
11. Perform white-box testing on a small program (e.g., find maximum of three numbers).
12. Design test cases for integration testing of a payroll system.
13. Prepare a bug report using a bug tracking template.
14. Calculate software metrics (e.g., cyclomatic complexity) for a given code.

Module IV: Project Management & Tools

15. Create a project schedule using Gantt chart (manual/tool-based).
16. Estimate the cost of a software project using basic COCOMO.
17. Set up a Git repository and perform basic version control operations (clone, commit, push, branch).
18. Perform a change management simulation for a software requirement change.
19. Prepare a software maintenance plan for an existing system.

Write a report on ethical issues in software development (privacy, security, piracy).

MK UNIVERSITY

PATAN, GUJARAT

ESTABLISHED BY THE GUJARAT GOVT. RECOGNIZED BY UGC UNDER SECTION 2(f) OF UGC ACT,1956

COURSE CODE: DITE603

COURSE NAME: BIG DATA ANALYTICS

Course Objectives:

- To introduce fundamental concepts, challenges, and technologies in big data.
- To develop skills in processing, storing, and analyzing large-scale datasets using distributed computing frameworks.
- To enable students to apply big data analytics techniques for data-driven decision-making and AI model training.
- To prepare students for real-world big data pipelines, data lakes, and cloud-based analytics.
- To foster awareness of data governance, privacy, and ethical handling of big data.

Course Outcomes: At the end of the course students shall be able to

CO1	**Explain** big data characteristics, architecture, and storage technologies.
CO2	**Process** and analyze large datasets using Hadoop, Spark, and NoSQL databases.
C03	**Apply** data mining and machine learning techniques on big data platforms.
C04	**Design** and implement scalable data pipelines for analytics and AI applications.

Unit	Content	Credit	Weightage
I	**Introduction to Big Data** What is Big Data? 5 V's: Volume, Velocity, Variety, Veracity, Value - Big data ecosystem: Hadoop, Spark, NoSQL, data lakes - Data storage: HDFS, cloud storage (AWS S3, Google Cloud Storage) - Big data challenges: Scalability, fault tolerance, real-time processing - **Applications:** Social media analytics, IoT data, log processing	1	25%
II	**Big Data Processing with Hadoop & Spark** Hadoop architecture: HDFS, MapReduce, YARN - Apache Spark: RDDs, DataFrames, Spark SQL - Spark MLlib for machine learning on big data - Stream processing: Spark Streaming, Kafka basics - **Applications:** Batch processing, real-time analytics, ETL pipelines	1	25%
III	**NoSQL Databases & Data Warehousing** NoSQL databases: Types (Document, Key-Value, Columnar, Graph) - MongoDB, Cassandra, HBase basics - Data warehousing: OLAP vs OLTP, star schema, snowflake schema - ETL tools: Apache NiFi, Talend (overview) - **Applications:** Customer 360, time-series data,	1	25%

	recommendation systems		
IV	**Big Data Analytics & AI Integration** Data mining techniques: Clustering, association rule mining - Machine learning at scale: Distributed training, model serving - Big data visualization: Tableau, Power BI, Apache Superset - Ethics and governance: Data privacy, GDPR, data anonymization - **Applications:** Predictive maintenance, fraud detection, business intelligence	1	25%

Textbooks:

- *Big Data: Principles and Best Practices* — Thomas Erl et al.
- *Hadoop: The Definitive Guide* — Tom White

Reference books:

- *Spark: The Definitive Guide* — Bill Chambers & Matei Zaharia
- *NoSQL Distilled* — Pramod J. Sadalage & Martin Fowler
- *Designing Data-Intensive Applications* — Martin Kleppmann
- *Big Data Analytics with Python* — Venkat Ankam

Online Platforms:

- NPTEL:
  1. *Big Data Computing* by Prof. Rajiv Misra (IIT Patna)
  2. *Introduction to Big Data* by IIT Kharagpur
- Coursera:
  1. *Big Data Specialization* by University of California San Diego
  2. *Apache Spark with Python* by IBM

PRACTICAL LIST:

Section A: Big Data Fundamentals & HDFS

1. Set up a single-node Hadoop cluster (using Docker or VM).
2. Perform basic HDFS operations: Upload, download, list, delete files.
3. Write a MapReduce program (WordCount) in Java or Python (MRJob).
4. Analyze large log files using MapReduce to find top IP addresses.

Section B: Apache Spark & Data Processing

5. Install PySpark and perform basic RDD operations (map, filter, reduceByKey).
6. Use Spark DataFrames to clean and analyze a large dataset (e.g., NYC taxi data).
7. Perform SQL queries on big data using Spark SQL.
8. Implement a streaming word count using Spark Streaming (or Structured Streaming).

Section C: NoSQL Databases & ETL

9. Perform CRUD operations in MongoDB using PyMongo.
10. Design a schema and insert data into Apache Cassandra.
11. Build a simple ETL pipeline using Apache NiFi to move data from CSV to HDFS.
12. Query data from a data warehouse (e.g., using Amazon Redshift or Google BigQuery).

Section D: Big Data Analytics & Mini-Project

13. Perform clustering on big data using Spark MLlib (K-means).
14. Build a recommendation system using collaborative filtering on a large dataset.

Create a dashboard using Tableau Public or Apache Superset to visualize big data insights.