GANGA INSTITUTE OF TECHNOLOGY AND MANAGEMENT, KABLANA (JHAJJAR)

An Autonomous Institute (UGC)

Approved by AICTE, New Delhi and Affiliated to MDU, Rohtak

NAAC ‘A’ GRADE

Scheme of Studies and Examination
Bachelor of Computer Applications

1. DEFINITION OF CREDIT

1	1 Lecture (L) per week	1 Credit
2	1Tutoria l(T) per week	1 Credit
3	1 Practical (P) per week	0.5 Credit
4	2 Practical (Lab) per week	1 Credit

2. RANGE OF CREDIT

A range of credits from136 to144 for a student to be eligible to get Under Graduate degree in Computer Application. A student will be eligible to get Under Graduate degree with Honours or Research Program, if he/she completes an additional 48 credits.

3. STRUCTURE OF UNDER GRADUATE PROGRAM (BCA)

Sr. No.	Category	Break up of Credits BCA	BCA (Honours)	BCA (With Research)
1	Discipline-Specific Courses (DSC)-Major Course	76*	116*	104*
2	Minor (MIC)/MIC Vocational (VOC)/Skill Enhancement Courses (SEC)/Internship	37*	45*	45*
3	Multidisciplinary Courses	9*	9*	9*
4	Ability Enhancement Course	8*	8*	8*
5	Value Added Course	6*	6*	6*
6	Research Project/Dissertation			12*
	Total Credits	**136***	**184***	**184***

* Minor variations all owed as per need of the respective disciplines.

* Students exiting the program after second semester and securing48 credits including 4 credits of summer internship will be awarded UG Certificate in the relevant Discipline/Subject. Students exiting the program after fourth semester and securing 96 credits including 4 credits of summer internship will be awarded UG Diploma in the relevant Discipline/Subject. Students will be awarded 3-year UG Degree in relevant major Discipline/Subject upon securing 136 credits.

* Four credits of internship earned by a student during summer internship after 2nd semester or 4th semester will be counted in 5th semester of a student who pursue 3-year UG Programs without taking exit option.

4. COURSE CODE AND DEFINITIONS

Sr. No.	Category	Course Code
1	Discipline-Specific Courses-Major Course	DSC
2	Minor Vocational	MV
3	Skill Enhancement Courses	SEC
4	Multidisciplinary Courses	MDC
5	Ability Enhancement Courses	AEC
6	Value Added Courses	VAC
7	Internship	INT
8	Research Project	PR
9	Dissertation	DISS

GANGA INSTITUTE OF TECHNOLOGY AND MANAGEMENT, JHAJJAR (HR.), DELHI-NCR

Scheme of Studies and Examination

BCA – 3rd Semester

w.e.f. 2025-26

Sr. No.	Category	Course Code	Course Title	Hours per Week			Total Load Per Week	Credits	Examination Schedule (Marks)				Exam Duration In Hours
				L	T	P			Assessment	End Semester Exam	Practical	Total	
1	Discipline-Specific Courses-Major Courses	DSC-BCA-201A	Operating Systems	4	0	0	4	4	30	70	-	100	3
2	Discipline-Specific Courses-Major Courses	DSC-BCA-203A	Object Oriented Programming Using C++	3	0	0	3	3	25	50	-	75	3
		LC-BCA-205A	Object Oriented Programming Using C++ Lab	0	0	2	2	1	10	-	15	25	3
3	Discipline-Specific Courses-Major Courses	DSC-BCA-207A	Database Management System & SQL	3	0	0	3	3	25	50	-	75	3
		LC-BCA-209A	Database Management System & SQL Lab	0	0	2	2	1	10	-	15	25	3
4	Minor Vocational	MV-BCA-211A	Web Development-I	2	0	0	2	2	15	35	-	50	3
		LC-BCA-213A	Web Development-I Lab	0	0	4	4	2	15	-	35	50	3
5	Skill Enhancement Courses	SEC-215A	Data Science	1	0	0	1	1	-	25	-	25	3
		LC-SEC-217A	Data Science Lab	0	0	4	4	2	15	-	35	50	3
6	Multidisciplinary Courses	Refer Table VII	-----------------------------------	3	0	0	3	3	25	50	-	75	3
7	Ability Enhancement Courses	Refer Table VIII	----------------------------------	2	0	0	2	2	15	35	-	50	3
Total Credits								**24**	**185**	**315**	**100**	**600**	

Scheme of Studies and Examination

BCA – 4th Semester

w.e.f. 2025-26

Sr. No.	Category	Course Code	Course Title	Hours per Week			Total Load Per Week	Credits	Examination Schedule (Marks)				Exam Duration In Hours
				L	T	P			Assessment	End Semester Exam	Practical	Total	
1	Discipline-Specific Courses Major Courses	DSC-BCA-202A	Computer System Architecture	4	0	0	4	4	30	70	-	100	3
2	Discipline-Specific Courses Major Courses	DSC-BCA-204A	Data Communication & computer Networks	4	0	0	4	4	30	70	-	100	3
3	Discipline-Specific Courses-Major Courses	DSC-BCA-206A	Java Programming	2	0	0	2	2	15	35	-	50	3
		LC-BCA-208A	Java Programming Lab	0	0	4	2	2	15	-	35	50	3
4	Discipline-Specific Courses-Major Courses	DSC-BCA-210A	Computer Graphics	3	0	0	3	3	25	50		75	3
		LC-BCA-212A	Computer Graphics Lab	0	0	2	2	1	10	-	15	25	3
5	Minor Vocational	MV-BCA-214A	Web Development-II	1	0	0	1	1	-	25		25	3
		LC-BCA-216A	Web Development-II Lab	0	0	6	3	3	25	-	50	75	3
6	Ability Enhancement Courses	Refer Table IX	----------------------------------	2	0	0	2	2	15	35	-	50	3
7	Value Added Courses	Refer Table X	----------------------------------	2	0	0	2	-	15	35	-	50	3
Total Credits								**24**	**180**	**320**	**100**	**600**	

Note:

1. The duration of all the end-term theory examinations shall be 3 hours.
2. The Criteria for awarding the internal assessment throughout the programme shall be as under:
   a) Sessional Examination: 20 Marks (66.67%)
   b) Assignments /Presentations /Seminars and Class Participation: 5 Marks (16.67%)
   c) Attendance : 5 Marks (16.67%)

Less than 65%	: 0% of Attendance marks
65% - 69.99%	: 40% of Attendance marks
70% - 74.99%	: 60% of Attendance marks
75% - 80%	: 80% of Attendance marks
Above 80%	: 100% of Attendance marks

3. The Criteria for awarding the Internal assessment practical course throughout the Programs hall be asunder:
   a) Practical Assignments /Practical File :06 Marks (60% Weightage of Assessment)
   b) Attendance (Criteria as mentioned above in 2 (c)) :04 Marks (40% Weightage of Assessment)
4. The panel of examiners for end-semester theory examinations shall be prepared and approved by the Department of Computer Science and Application of the internal/external examiners based on their expertise/ specialization/ area of interest.
5. The panel of examiners based on the examiners' expertise/specialization/area of interest for practicum/viva-voce examination shall be recommended by the Department of Computer Science and Application. In case of unavailability of external examiners due to unavoidable circumstances, the Controller of Examinations may allow the conduct of practical examinations by the internal examiners.
6. The panel of examiners for assessment of the Project Reports/Dissertation/Research Project/Summer Internship Reports shall be recommended by the Department of Computer Science and Application.
7. A student while selecting the minor discipline has to ensure that the courses of the opted minor discipline do not match with the major courses of his/her opted programme.
8. The student may choose the Multidisciplinary Courses from other than the courses offered by their own/respective department.

Semester-III

Table No VII (Multidisciplinary Courses)

S. No.	Category	Course Code	Course Title	Offered By
1	Multidisciplinary Courses	MDC-BBA-201A	Corporate Social Responsibility	Department of Management (UG)
2		MDC-BBA-203A	Finance for Non-Finance Professional	Department of Management (UG)
3		MDC-BBA-205A	Group Dynamics and Leadership	Department of Management (UG)
4		MDC-CSA-209A	Graphic Designing	Department of Computer Application
5		MDC-CSA-211A	Office and Spreadsheet tools learning	Department of Computer Application
6		MDC-CSA-213A	Web Technology	Department of Computer Application

Table No. VIII (Ability Enhancement Course)

S. No.	Category	Course Code	Course Name
1	Ability Enhancement Courses	AEC-201A	English-II
2		AEC-203A	German

Semester-IV

Table No. IX (Ability Enhancement Course)

S. No.	Category	Course Code	Course Name
1	Ability Enhancement Course	AEC-202A	Hindi-II
2		AEC-204A	French

Table No. X (Value Added Courses)

S. No.	Category	Course Code	Course Title
1	Value Added Courses	VAC-213A	Basics of GST
2		VAC-215A	Professional Networking Skills
3		VAC-217A	Social and Emotional Learning
4		VAC-219A	Social Networks
5		VAC-221A	Cyber Security

<table>
<tr><td>Course Code</td><td colspan="5"><b>DSC-BCA-201A</b></td></tr>
<tr><td>Category</td><td colspan="5">Discipline- Specific Courses-Major Courses</td></tr>
<tr><td>Course Title</td><td colspan="5"><b>Operating Systems</b></td></tr>
<tr><td rowspan="2">Scheme and Credits</td><td>L</td><td>T</td><td>P</td><td><b>Credits</b></td><td rowspan="2"><b>Semester III</b></td></tr>
<tr><td>4</td><td>0</td><td>0</td><td><b>4</b></td></tr>
<tr><td>Course Objectives</td><td colspan="5">The objectives of this course are
• To understand the fundamental concepts of operating systems, including process management, memory management, and scheduling algorithms.
• To explore I/O management, secondary storage structures, and the role of system calls in OS operations.</td></tr>
<tr><td>Assessment</td><td colspan="5">30 Marks</td></tr>
<tr><td>End Semester Examination</td><td colspan="5">70 Marks</td></tr>
<tr><td>Total</td><td colspan="5">100 Marks</td></tr>
<tr><td>Duration of Exam</td><td colspan="5">03 Hours</td></tr>
</table>

Course Outcomes: After successful completion of this course, the students will be able to

COs	Skills Demonstrated	RBT Level
CO1	Recall fundamental concepts, types, operations, and structures related to operating systems, processes, memory, I/O, and file systems.	Level 1: Remember
CO2	Explain the functioning of processes, scheduling, memory management, I/O systems, and file structures in operating systems.	Level 2: Understand
CO3	Apply process scheduling, memory allocation, and file access strategies to optimize system performance.	Level 3: Apply
CO4	Analyze process interactions, memory techniques, I/O mechanisms, and file system methods for system-level efficiency.	Level 4: Analyze

Note: Examiner will set nine questions in total. Question one will be compulsory. Question one will have 7 parts of 2 marks each from all units and remaining eight questions of 14 marks each to be set by taking two questions from each unit. The students have to attempt five questions in total, first being compulsory and selecting one from each unit.

Unit-I

Introduction to Operating Systems: Objectives and Characteristics. Classification: Batch, Multiprogramming, Multi-processing, Multi-tasking, Time-sharing, Distributed, Network and Real time Operating systems. System Calls and Services.

Functions and Structures: Operating System Functions- Process management, Memory management, Secondary storage management, I/O management, File management, Protection and Security. StructuresSimple Structure, Monolithic structure, Layered approach

Unit-II

Process Management and Scheduling: Process concept- Process State Model, Process Control Block and Threads. Process Scheduling- Scheduling Queues, Schedulers and Context Switch. Operations on Processes, Cooperating processes and Inter-Process Communication.

Process Scheduling: Scheduling Criteria, Scheduling Algorithms: Single Processor Scheduling: FCFS, SJF, Round Robin, Multi Feedback Queue. Multiple Processor Scheduling and Real Time scheduling. Scheduling Algorithm Evaluation.

Unit-III

Memory Management: Concepts of Memory Management, Logical and Physical address space, Swapping, Memory allocation: Contiguous and Non-Contiguous. Paging: Hardware Support. Page Map Table and Protection. Segmentation: Hardware Support and Protection and Sharing.

Virtual Memory: Need of Virtual Memory, Demand paging, Pure Demand Paging. Handling page faults, Performance of Demand Paging. Page replacement Algorithms and Allocation of Frames: Allocation algorithms and Global vs Local Allocation. Thrashing.

Unit-IV

I/O Management: Basic I/O Devices, Types of I/O Devices: Block and Character Devices. I/O Software: Device Independent I/O, User Space I/O and Kernel I/O Software. Device Controllers, Device Drivers and Interrupt Handlers. Communication Approaches to I/O Devices: Special Instruction I/O, Memory Mapped I/O and Direct Memory Access (DMA). Secondary Storage Structure: Disk Structure and Disk Scheduling Algorithms.

File System Interface: File Concept: Attributes, Operations and Types. File Access Methods: Sequential Access, Direct Access and Indexed Sequential. Free Space Management. Directory Structures: Single Level, Two level and Tree Structured. File Protection and Sharing.

Suggested Readings:

- Operating System Concepts by Abraham Silberschatz, Peter B. Galvin, and Greg Gagne
- Modern Operating Systems by Andrew S. Tanenbaum and Herbert Bos
- Operating Systems: Internals and Design Principles by William Stallings
- Operating System Principles by Peter B. Galvin and Abraham Silberschatz
- Operating Systems: A Concept-Based Approach by D.M. Dhamdhere
- Understanding Operating Systems by Ann McHoes and Ida M. Flynn

Useful Video Links:

Unit No.	Topics	Links
Unit-I	Introduction to Operating Systems	https://archive.nptel.ac.in/courses/106/105/106105214/
	Operating System Structures	https://archive.nptel.ac.in/courses/106/105/106105214/
Unit-II	Process Management	https://archive.nptel.ac.in/courses/106/105/106105214/
	CPU Scheduling	https://archive.nptel.ac.in/courses/106/105/106105214/
	Threads and Inter-Process Communication	https://archive.nptel.ac.in/courses/106/105/106105214/
Unit-III	Memory Management	https://archive.nptel.ac.in/courses/106/105/106105214/
	Virtual Memory	https://archive.nptel.ac.in/courses/106/105/106105214/
	Page Replacement Algorithms	https://archive.nptel.ac.in/courses/106/105/106105214/
Unit-IV	I/O Management	https://archive.nptel.ac.in/courses/106/105/106105214/
	File Systems	https://archive.nptel.ac.in/courses/106/105/106105214/
	Disk Scheduling	https://archive.nptel.ac.in/courses/106/105/106105214/

Course Code	**DSC-BCA-203A**				
Category	Discipline- Specific Courses-Major Courses				
Course Title	**Object Oriented Programming Using C++**				
Scheme and Credits	L	T	P	**Credits**	**Semester III**
	3	0	0	**3**	
Course Objectives	The objectives of this course are • To understand object-oriented programming concepts such as classes, inheritance, polymorphism, and encapsulation in C++. • To develop proficiency in memory management, exception handling, file operations, and the use of templates in C++ applications.				
Assessment	25 Marks				
End Semester Examination	50 Marks				
Total	75 Marks				
Duration of Exam	03 Hours				

Course Outcomes: After successful completion of this course, the students will be able to

COs	Skills Demonstrated	RBT Level
CO1	Recall the basic concepts of object-oriented programming, C++ syntax, data types, classes, inheritance, polymorphism, and file handling.	Level 1: Remember
CO2	Explain the principles of OOP, class structures, memory management, function overloading, templates, and exception handling in C++.	Level 2: Understand
CO3	Apply OOP principles to develop C++ programs using classes, inheritance, polymorphism, dynamic memory, and file I/O operations.	Level 3: Apply
CO4	Analyze OOP features like inheritance, polymorphism, encapsulation, and exception handling for efficient and modular program design.	Level 4: Analyze

Note: Examiner will set nine questions in total. Question one will be compulsory. Question one will have 5 parts of 2 mark each from all units and remaining eight questions of 10 marks each to be set by taking two questions from each unit. The students have to attempt five questions in total, first being compulsory and selecting one from each unit.

Unit-I

Introduction to OOP concepts: Procedural Vs. Object- Oriented Programming, Principles of OOP and their benefits. Object, classes, Inheritance, Abstraction, Encapsulation Polymorphism, Dynamic Binding, Message passing.

C++ Programming Basics: Syntax and structure of C++ programs, Data types, variables, and constants in C++, Control structures: decision making and looping constructs

Unit-II

Classes and objects: Defining and using classes and objects, Member functions and data members, Access specifiers: public, private, protected, Functions and parameter passing in C++, Arrays and strings in C++, Pointer, Constructors and destructors.

Inheritance: Derived class and Base class, Types of inheritance: single, multiple, multilevel, hierarchical, Access control in inheritance.

Unit-III

Polymorphism: function overloading, Operator overloading, Virtual functions and dynamic polymorphism, Abstract classes and pure virtual functions, Encapsulation and data hiding, Friend functions, static function.
Memory Management: Dynamic Memory Allocation: new, delete, Object Creation at run time

Unit-IV

Exception handling: Throwing, Catching, Re-throwing an exception, specifying exception: processing unexpected exceptions; try-catch blocks, Exception propagation, Templates: class and function templates, Standard Template Library: benefits of STL and generic programming
Working with Files: Stream Classes, File input and output Operations in C++, Error handling during file operations.

Suggested Readings:

- Object-Oriented Programming with C++ by E. Balagurusamy
- Let Us C++ by Yashavant Kanetkar
- Programming: Principles and Practice Using C++ by Bjarne Stroustrup
- C++ Primer (5th Edition) by Stanley B. Lippman, Josée Lajoie, and Barbara E. Moo
- Accelerated C++: Practical Programming by Example by Andrew Koenig and Barbara E. Moo
- C++ Programming: From Problem Analysis to Program Design by D.S. Malik

Useful Video Links:

Unit No.	**Topics**	**Links**
Unit-I	Introduction to OOPS	https://onlinecourses.nptel.ac.in/noc25_cs34/preview
	Classes and Objects	https://onlinecourses.nptel.ac.in/noc25_cs34/preview
	Introduction to C++	https://onlinecourses.nptel.ac.in/noc20_cs07/preview
Unit-II	Inheritance	https://onlinecourses.nptel.ac.in/noc25_cs34/preview
	Classes and Objects	https://onlinecourses.nptel.ac.in/noc20_cs07/preview
Unit-III	Polymorphism	https://onlinecourses.nptel.ac.in/noc25_cs34/preview
Unit-IV	Exception Handling	https://onlinecourses.nptel.ac.in/noc25_cs34/preview
	File Handling	https://onlinecourses.nptel.ac.in/noc25_cs34/preview
	Templates and Generics	https://onlinecourses.nptel.ac.in/noc25_cs34/preview

Course Code	**LC-BCA-205A**				
Category	Discipline- Specific Courses-Major Courses				
Course Title	**Object Oriented Programming Using C++ Lab**				
Scheme and Credits	L	T	P	**Credits**	**Semester III**
	0	0	2	**1**	
Course Objectives	The objectives of this course are - To develop object-oriented programming skills in C++ through concepts like classes, inheritance, polymorphism, and memory management. - To implement data handling, control structures, operator overloading, and dynamic memory allocation in real-world applications.				
Assessment	10 Marks				
End Semester Examination	15 Marks				
Total	25 Marks				
Duration of Exam	03 Hours				

Course Outcomes: After successful completion of this course, the students will be able to

COs	Skills Demonstrated	RBT Level
CO1	Apply fundamental C++ programming concepts including variables, control structures, functions, and memory management to develop functional applications.	Level 3: Apply
CO2	Analyze and implement object-oriented programming principles such as encapsulation, inheritance, and polymorphism in class-based design.	Level 4: Analyze
CO3	Evaluate program design decisions using friend functions, access specifiers, and function overloading to enhance code readability and efficiency.	Level 5 Evaluate
CO4	Create robust and reusable C++ applications by integrating constructors, destructors, operator overloading, and advanced OOP techniques.	Level 6 Create

Exp. no.	Contents
1.	Write a C++ program to demonstrate basic syntax, variables, constants, and control structures (if-else, loops).
2.	Implement a simple program that demonstrates Object-Oriented Programming (OOP) principles: encapsulation, abstraction, and polymorphism.
3.	Define a Student class with attributes (name, age, grade) and methods to display student details. Create multiple objects and test them.
4.	Write a program that accepts user input and uses loops (for, while, do-while) and conditional statements (if-else, switch) to perform computations.
5.	Implement a BankAccount class with attributes like account number, balance, and methods for deposit and withdrawal. Use access specifiers.
6.	Create a class Employee that stores an array of employee names and their salaries. Implement functions to update and display employee details.
7.	Write a program that dynamically allocates memory for an integer array using new and deallocates it using delete.
8.	Implement a class Car with a constructor to initialize brand and model details and a destructor to clean up resources.
9.	Demonstrate single, multiple, and multilevel inheritance by creating a Person base class and derived classes (Employee, Student).

10.	Implement function overloading for calculating the area of different shapes. Overload the + operator to add two complex numbers.
11.	Create a base class Shape with a virtual function draw(). Derive classes like Circle and Rectangle and override the function.
12.	Implement a Rectangle class with private members and a friend function to calculate the area.

<table>
<tr><td>Course Code</td><td colspan="5"><b>DSC-BCA-207A</b></td></tr>
<tr><td>Category</td><td colspan="5">Discipline- Specific Courses-Major Courses</td></tr>
<tr><td>Course Title</td><td colspan="5"><b>Database Management System & SQL</b></td></tr>
<tr><td rowspan="2">Scheme and Credits</td><td>L</td><td>T</td><td>P</td><td><b>Credits</b></td><td rowspan="2"><b>Semester III</b></td></tr>
<tr><td>3</td><td>0</td><td>0</td><td><b>3</b></td></tr>
<tr><td>Course Objectives</td><td colspan="5">The objectives of this course are
• To understand the principles of database management, including database design, relational models, and query processing.
• To apply normalization techniques, transaction management, concurrency control, and recovery mechanisms for reliable and efficient database operations.</td></tr>
<tr><td>Assessment</td><td colspan="5">25 Marks</td></tr>
<tr><td>End Semester Examination</td><td colspan="5">50 Marks</td></tr>
<tr><td>Total</td><td colspan="5">75 Marks</td></tr>
<tr><td>Duration of Exam</td><td colspan="5">03 Hours</td></tr>
</table>

Course Outcomes: After successful completion of this course, the students will be able to

COs	Skills Demonstrated	RBT Level
CO1	Recall basic concepts of database systems, models, ER design, relational structures, normalization, and recovery techniques.	Level 1: Remember
CO2	Explain database design approaches, relational operations, query languages, normalization forms, and concurrency control methods.	Level 2: Understand
CO3	Apply ER modeling, relational algebra/calculus, normalization techniques, and transaction protocols to develop efficient databases.	Level 3: Apply
CO4	Analyze schema designs, normalization impacts, concurrency issues, and recovery mechanisms for data consistency and efficiency.	Level 4: Analyze

Note: Examiner will set nine questions in total. Question one will be compulsory. Question one will have 5 parts of 2 marks each from all units and remaining eight questions of 10 marks each to be set by taking two questions from each unit. The students have to attempt five questions in total, first being compulsory and selecting one from each unit.

Unit-I

Database Management System: Introduction, Database System Applications, History of Database Systems, Database System Vs. File Processing System, View of Data, Data Abstraction, Instances and Schemas. DBMS Environment, Database languages, Database Models.

Database design and ER Model: Physical, Conceptual and Logical Database design, Entity- Relationship model: Entities, Relationships, Representation of entities, attributes, Representation of relationship set, Generalization, Aggregation, Conceptual design with ER Mode

Unit-II

Relational Model: Introduction to the Relational Model, Attributes, Domains, Tuples, Relations and their schemes, relation representation, Keys, relationship, relational operations, Integrity Constraint Over relations, Enforcing Integrity constraints, querying relational data, View: Introduction to Views, Destroying / altering Views.

Relational Algebra and Calculus: Relational Algebra & its operations, Relational calculus & its types, Power of Algebra and calculus.

Unit-III

Normalization: Schema Refinement, Problems caused by redundancy, Decomposition & its properties; Normalization: First, Second, Third Normal forms, BCNF, Multivalued Dependencies, Join Dependencies.

Transaction Management & Concurrency Control: ACID properties, Transactions and Schedules, Concurrent execution of transaction, Serializability and Recoverability, Lock based Concurrency control, Lock Management, Lock Conversion, Dealing with deadlocks, Concurrency without Locking.

Unit-IV

Crash Recovery and Backup: Failure classifications, storage structure, Recovery & Atomicity, Log base recovery, Recovery with concurrent transactions, Failure with loss of nonvolatile storage, Database backup & recovery from catastrophic failure, Remote Backup System.

Storage and File Organization: Overview of physical storage media, Storage access; File organization, Operations on Files, Serial Files, Sequential Files, Index-Sequential Files, Direct Files.

Suggested Readings:

- Database Management Systems by Raghu Ramakrishnan and Johannes Gehrke
- Database Management System by R.P. Mahapatra and Govind Verma
- Database Management Systems: A Practical Approach by Rajiv Chopra
- Fundamentals of Database Systems by Ramez Elmasri and Shamkant B. Navathe
- An Introduction to Database Systems by C.J. Date
- Database System Concepts by Abraham Silberschatz, Henry F. Korth, and S. Sudarshan

Useful Video Links:

Unit No.	**Topics**	**Links**
Unit-I	Database Management System	https://archive.nptel.ac.in/courses/106/105/106105175/?utm
Unit-II	Relational Model	https://archive.nptel.ac.in/courses/106/105/106105175/?utm
Unit-III	Transaction Management & Concurrency Control	https://archive.nptel.ac.in/courses/106/105/106105175/?utm
Unit-IV	Recovery & Atomicity	https://archive.nptel.ac.in/courses/106/105/106105175/?utm

<table>
<tr><td>Course Code</td><td colspan="5"><b>LC-BCA-209A</b></td></tr>
<tr><td>Category</td><td colspan="5">Discipline- Specific Courses-Major Courses</td></tr>
<tr><td>Course Title</td><td colspan="5"><b>Database Management System & SQL Lab</b></td></tr>
<tr><td rowspan="2">Scheme and Credits</td><td>L</td><td>T</td><td>P</td><td><b>Credits</b></td><td rowspan="2"><b>Semester III</b></td></tr>
<tr><td>0</td><td>0</td><td>2</td><td><b>1</b></td></tr>
<tr><td>Course Objectives</td><td colspan="5">The objectives of this course are
• To design, implement, and manage relational databases using SQL, including schema creation, data manipulation, and view management.
• To apply normalization techniques, transaction handling, and backup strategies for database optimization, integrity, and recovery.</td></tr>
<tr><td>Assessment</td><td colspan="5">10 Marks</td></tr>
<tr><td>End Semester Examination</td><td colspan="5">15 Marks</td></tr>
<tr><td>Total</td><td colspan="5">25 Marks</td></tr>
<tr><td>Duration of Exam</td><td colspan="5">03 Hours</td></tr>
</table>

Course Outcomes: After successful completion of this course, the students will be able to

COs	Skills Demonstrated	RBT Level
CO1	Apply relational database concepts to create, modify, and query databases using SQL.	Level 3: Apply
CO2	Analyze database structures and queries by interpreting ER models, comparing file systems vs. DBMS, and applying relational algebra operations.	Level 4: Analyze
CO3	Evaluate database designs and normalization techniques to improve schema efficiency and eliminate anomalies.	Level 5 Evaluate
CO4	Create reliable and secure database systems by implementing transaction management, concurrency control, and backup mechanisms.	Level 6 Create

Exp. no.	Contents
1.	Install MySQL/PostgreSQL/Oracle. Create and manage a simple database with tables.
2.	Demonstrate how data is stored and retrieved in file systems vs. databases.
3.	Design an ER diagram for a university database with entities like Student, Course, Instructor, and Department.
4.	Convert an ER diagram into a relational schema and implement it using SQL.
5.	Write SQL queries to create tables with different constraints (Primary Key, Foreign Key, NOT NULL, UNIQUE).
6.	Perform basic SQL operations: SELECT, INSERT, UPDATE, DELETE with WHERE conditions.
7.	Create, modify, and delete views for a relational database.
8.	Implement relational algebra operations (Selection, Projection, Union, Intersection, Join) using SQL.
9.	Normalize a given unnormalized database schema step by step up to BCNF.
10.	Create transactions for a banking system using SQL commands like COMMIT, ROLLBACK.
11.	Demonstrate deadlock occurrence and resolution strategies in a DBMS.
12.	Perform a full database backup and restore using MySQL/PostgreSQL.

<table>
<tr><td>Course Code</td><td colspan="5"><b>MV-BCA-211A</b></td></tr>
<tr><td>Category</td><td colspan="5">Minor Vocational</td></tr>
<tr><td>Course Title</td><td colspan="5"><b>Web Development-I</b></td></tr>
<tr><td rowspan="2">Scheme and Credits</td><td>L</td><td>T</td><td>P</td><td><b>Credits</b></td><td rowspan="2"><b>Semester III</b></td></tr>
<tr><td>2</td><td>0</td><td>0</td><td><b>2</b></td></tr>
<tr><td>Course Objectives</td><td colspan="5">The objectives of this course are
• To understand the fundamentals of Node.js, including its history, features, and use cases. Set up the development environment and write basic scripts.
• To create and manage REST APIs, perform file system operations, and utilize version control with Git for effective project management and collaboration.</td></tr>
<tr><td>Assessment</td><td colspan="5">15 Marks</td></tr>
<tr><td>End Semester Examination</td><td colspan="5">35 Marks</td></tr>
<tr><td>Total</td><td colspan="5">50 Marks</td></tr>
<tr><td>Duration of Exam</td><td colspan="5">03 Hours</td></tr>
</table>

Course Outcomes: After successful completion of this course, the students will be able to

COs	**Skills Demonstrated**	**RBT Level**
CO1	Recall fundamental concepts of JavaScript, Node.js features, core modules, and development environment setup.	Level 1: Remember
CO2	Explain Node.js architecture, modular programming, advanced JavaScript usage, and Git-based version control processes.	Level 2: Understand
CO3	Apply Node.js tools to develop servers, manage dependencies, create REST APIs, and perform file operations.	Level 3: Apply
CO4	Analyze project structures, code modularity, error handling, and integration of Node.js features in real-time solutions.	Level 4: Analyze

Note: Examiner will set nine questions in total. Question one will be compulsory. Question one will have 5 parts of 1 marks each from all units and remaining eight questions of 5 marks each to be set by taking two questions from each unit. The students have to attempt five questions in total, first being compulsory and selecting one from each unit.

Unit-I

Node.js Overview and JavaScript Fundamentals: Introduction, history, key features, and applications. Comparison between JavaScript and Node.js. Differences between client-side and server-side programming. Use cases of Node.js in real-time applications and its integration with JavaScript.

Installation and Environment Setup: Installing Node.js and npm, understanding the Node.js runtime, writing and executing basic scripts.

JavaScript Refresher: JavaScript fundamentals including variables, data types, functions, control structures (if/else, loops), and arrays.

Unit-II

Node.js Modules and HTTP Server: Understanding modules: core modules (fs, http, path, etc.), global modules, and non-global modules. Creating and managing a basic HTTP server.

Advanced JavaScript Concepts in Node.js: Using functions as parameters, callback functions, and arrow function syntax. Understanding modular programming and organizing code into separate files for scalability.

Unit-III

Project Structure and Dependency Management: Introduction to package.json, initializing Node.js projects, managing dependencies, and configuring scripts.

Version Control and Git Integration: Basic Git commands, initializing repositories, tracking changes, gitignore usage, and project collaboration using Git.

Development Tools: Usage of Nodemon and other automation tools for improving the development experience.

Unit-IV

API Development and File System Operations: Creating simple REST APIs using Node.js, handling GET and POST requests, parsing and sending JSON data.

File System Module: Reading from and writing to files, directory operations, working with file paths, and handling errors during file operations.

Practical Application and Mini Project: End-to-end mini project integrating concepts from all units, including module usage, server creation, file operations, and basic API routes.

Suggested Readings:

- Node.js in Action by Mike Cantelon, Marc Harter, and T.J. Holowaychuk
- Learning Node.js Development by Andrew Mead
- Node.js for JavaScript Developers by Richard Blum
- Mastering Node.js by Sandro Pasquali
- Node.js 8 Cookbook by David Mark Clements
- The Node Beginner Book by Manuel Kiessling

Useful Video Links:

Unit No.	Topics	Links
Unit-I	Introduction to Client-side JavaScript	https://archive.nptel.ac.in/courses/106/106/106106156/
	APIs and Mobile Apps Use Web-Servers	https://archive.nptel.ac.in/courses/106/106/106106156/
Unit-II	Introduction to Server-side JavaScript and HTML/CSS	https://archive.nptel.ac.in/courses/106/106/106106156/
	Introduction to Databases	https://archive.nptel.ac.in/courses/106/106/106106156/
Unit-III	Setting up a Website	https://onlinecourses.nptel.ac.in/noc20_cs52/preview
	Using Third-party Web Services	https://onlinecourses.nptel.ac.in/noc20_cs52/preview
Unit-IV	Extended Project	https://onlinecourses.nptel.ac.in/noc20_cs52/preview

Course Code	**LC-BCA-213A**				
Category	Minor Vocational				
Course Title	**Web Development-I Lab**				
Scheme and Credits	L	T	P	**Credits**	**Semester III**
	0	0	4	**2**	
Course Objectives	The objectives of this course are • To set up Node.js and npm, write and execute a "Hello, World!" script, and ensure Node.js is correctly installed. • Create a basic HTTP server using the http module to handle GET requests and respond with a "Hello, World!" message.				
Assessment	15 Marks				
End Semester Examination	35 Marks				
Total	50 Marks				
Duration of Exam	03 Hours				

Course Outcomes: After successful completion of this course, the students will be able to

COs	Skills Demonstrated	RBT Level
CO1	Apply foundational Node.js concepts to build and execute basic scripts, handle files, and create simple servers.	Level 3: Apply
CO2	Analyze and structure Node.js applications by organizing modules, handling asynchronous operations, and integrating external packages.	Level 4: Analyze
CO3	Evaluate development tools and practices by integrating Git version control, automation tools like nodemon, and dependency management in Node.js projects.	Level 5 Evaluate
CO4	Create full-fledged RESTful APIs and file-based backend services using Node.js and Express, implementing modular and scalable code.	Level 6 Create

Exp. no.	Contents
1.	Install Node.js and npm. Write a "Hello, World!" script and execute it to ensure that Node.js is set up correctly.
2.	Use the fs module to create a script that reads the contents of a file and displays it on the console.
3.	Create a basic HTTP server using the http module in Node.js that responds with a "Hello, World!" message on a GET request.
4.	Write a JavaScript function that takes two numbers as arguments and returns their sum. Add error handling to ensure both arguments are numbers.
5.	Implement a function that takes a callback to read a file asynchronously and display its contents once the reading is complete.
6.	Organize your Node.js application into modules. Split the HTTP server code into a separate module and import it into your main file.
7.	Initialize a Node.js project using npm init, install a package (e.g., express), and create a basic server using that package.
8.	Initialize a Git repository in your project. Commit the initial changes and push to a remote repository (e.g., GitHub). Add node_modules to .gitignore.
9.	Install and set up nodemon to automatically restart your server whenever there are changes to your code.

10.	Use express to create a simple REST API with routes for handling GET and POST requests. Ensure that the server can handle basic JSON data.
11.	Create a script that uses the fs module to write data to a file, read the content of the file, and display it on the console.
12.	Build a simple API in Node.js that reads data from a file when a GET request is made and writes new data to the file when a POST request is sent.

Course Code	**SEC-215A**				
Category	Skill Enhancement Courses				
Course Title	**Data Science**				
Scheme and Credits	L	T	P	**Credits**	**Semester III**
	1	0	0	**1**	
Course Objectives	The objectives of this course are • To apply Excel-based data manipulation, transformation, and visualization techniques for effective data analysis and decision-making. • To utilize predictive analysis, descriptive statistics, and advanced Excel functions for building data models, generating insights, and creating interactive dashboards.				
Assessment					
End Semester Examination	25 Marks				
Total	25 Marks				
Duration of Exam	03 Hours				

Course Outcomes: After successful completion of this course, the students will be able to

COs	Skills Demonstrated	RBT Level
CO1	Recall key concepts of data science, Excel fundamentals, data types, functions, and basic statistical measures.	Level 1: Remember
CO2	Explain techniques for data cleaning, transformation, validation, visualization, and analysis using Excel tools.	Level 2: Understand
CO3	Apply Excel functions, PivotTables, charts, and predictive tools for data manipulation, forecasting, and optimization.	Level 3: Apply
CO4	Analyze datasets using statistical methods, advanced Excel functions, and dashboards to derive and present insights.	Level 4: Analyze

Note: Examiner will set nine questions in total. Question one will be compulsory. Question one will have 7 parts of 1 marks each from all units and remaining eight questions of 7 marks each to be set by taking two questions from each unit. The students have to attempt five questions in total, first being compulsory and selecting one from each unit.

Unit-I

Introduction to Data Science: Overview of Data Science and its significance in various industries, Key concepts: Data collection, data analysis, data visualization, Data Science workflow: Data Cleaning, Data Analysis, Data Visualization, Data Modeling.

Excel Fundamentals for Data Science: Introduction to Excel Interface and Key Features Basic Excel Functions: SUM, AVERAGE, COUNT, COUNTA, MAX, MIN, Data Formatting: Conditional formatting, cell formatting, number formatting, Working with Excel Sheets and Workbooks

Data Types and Data Structure in Excel: Types of Data: Numeric, Text, Date, Boolean, organizing data: Rows, Columns, and Tables, Importing Data into Excel (CSV, text files, and databases)

Unit-II

Data Cleaning Techniques: Handling missing values: Techniques for filling or removing missing data, Duplicate Data: Identifying and removing duplicates, Data Validation: Using data validation rules to maintain data integrity.

Data Transformation in Excel: Using Text Functions: CONCATENATE, LEFT, RIGHT, MID, TRIM, Date and Time Functions: YEAR, MONTH, DAY, TEXT, Data Sorting and Filtering: Basic Sorting, Advanced Filters, Custom Filters.

Data Manipulation Techniques: Using Excel Pivot Tables for summarizing data, Aggregating data with SUMIF, COUNTIF, AVERAGEIF functions, Working with Excel Power Query for data transformation.

Unit-III

Descriptive Statistics: Calculating measures of central tendency: Mean, Median, Mode, Measures of Dispersion: Range, Variance, Standard Deviation, Creating Frequency Distributions and Histograms.

Excel Charts and Graphs: Introduction to Excel charts: Bar, Line, Pie, Scatter, and Area charts, Customizing charts: Axis formatting, adding titles, changing colors, Creating Pivot Charts.

Data Analysis Tools in Excel: Using the Data Analysis Toolpak for Regression Analysis, ANOVA, and Descriptive Statistics, Introduction to Excel Solver for optimization problems, Scenario Analysis using Excel What-If tools: Goal Seek, Data Tables.

Unit-IV

Introduction to Predictive Analysis: Introduction to basic predictive analytics: Linear regression, Using Excel's Regression tool to perform simple linear regression, forecasting with Excel: Using TREND, FORECAST, and LINEST functions.

Advanced Excel Functions for Data Science: Array Formulas and Dynamic Arrays, Using INDEX-MATCH functions for advanced lookups, Conditional aggregation: SUMIFS, AVERAGEIFS, COUNTIFS.

Reporting and Presentation: Creating Dashboards with Excel: Combining charts, tables, and PivotTables, Automating reports with Excel macros and VBA, Best practices for presenting data and insights: Formatting, layout, and storytelling with data.

Suggested Readings:

- Data Analysis with Excel by Manisha Nigam
- Data Analysis and Business Modelling Using Microsoft Excel by Hansa Lysander Manohar
- Statistical Data Analysis Using MS-Excel by Dr. B.G. Kore
- Modern Data Analytics in Excel by George Mount
- Microsoft Excel Data Analysis for Dummies by Stephen L. Nelson
- Business Statistics Using Excel: A Complete Course in Data Analytics by R. Panneerselvam

Useful Video Links:

Unit No.	**Topics**	**Links**
Unit-I	Introduction to Data and Excel Environment	https://nptel.ac.in/noc/courses/noc22/SEM1/106106235/Week_01.html
	Data Types and Handling Data in Excel	https://nptel.ac.in/noc/courses/noc22/SEM1/106106235/Week_02.html
Unit-II	Data Cleaning and Manipulation	https://nptel.ac.in/noc/courses/noc22/SEM1/106106235/Week_03.html
	Advanced Data Transformation	https://nptel.ac.in/noc/courses/noc22/SEM1/106106235/Week_04.html
Unit-III	Descriptive Statistics and PivotTables	https://nptel.ac.in/noc/courses/noc22/SEM1/106106235/Week_05.html

	Charts and Visualization Tools in Excel	https://nptel.ac.in/noc/courses/noc22/SEM1/106106235/Week_06.html
Unit-IV	Regression, Forecasting, and Solver	https://nptel.ac.in/noc/courses/noc22/SEM1/106106235/Week_07.html
	Dashboards and Reporting with Excel	https://nptel.ac.in/noc/courses/noc22/SEM1/106106235/Week_08.html

<table>
<tr><td>Course Code</td><td colspan="5"><b>LC-SEC-217A</b></td></tr>
<tr><td>Category</td><td colspan="5">Skill Enhancement Courses</td></tr>
<tr><td>Course Title</td><td colspan="5"><b>Data Science Lab</b></td></tr>
<tr><td rowspan="2">Scheme and Credits</td><td>L</td><td>T</td><td>P</td><td><b>Credits</b></td><td rowspan="2"><b>Semester III</b></td></tr>
<tr><td>0</td><td>0</td><td>4</td><td><b>2</b></td></tr>
<tr><td>Course Objectives</td><td colspan="5">The objectives of this course are
• To develop proficiency in Node.js, focusing on core modules, JavaScript fundamentals, and building real-time applications using server-side programming.
• To create and manage REST APIs, perform file system operations, and utilize version control with Git for effective project management and collaboration.</td></tr>
<tr><td>Assessment</td><td colspan="5">15 Marks</td></tr>
<tr><td>End Semester Examination</td><td colspan="5">35 Marks</td></tr>
<tr><td>Total</td><td colspan="5">50 Marks</td></tr>
<tr><td>Duration of Exam</td><td colspan="5">03 Hours</td></tr>
</table>

Course Outcomes: After successful completion of this course, the students will be able to

COs	**Skills Demonstrated**	**RBT Level**
CO1	Apply Excel functions and tools to organize, analyze, and visualize data for real-world business scenarios.	Level 3: Apply
CO2	Analyze and manipulate complex data sets by identifying patterns, performing calculations, and cleaning data to ensure accuracy.	Level 4: Analyze
CO3	Evaluate and optimize data organization and visualization strategies to improve decision-making and reporting accuracy.	Level 5 Evaluate
CO4	Create advanced Excel solutions to automate tasks, generate forecasts, and develop interactive dashboards for data-driven decision-making.	Level 6 Create

Exp. no.	**Contents**
1.	Import the "Employee_Data.csv" file into Excel, convert it into a table, and apply conditional formatting to highlight salaries above $50,000 in green and below $20,000 in red.
2.	Identify and remove duplicate records based on "Employee ID" and apply data validation to the "Email" column for valid formats.
3.	Calculate Mean, Median, Mode, Range, Variance, and Standard Deviation for the "Monthly Sales" data and create a Histogram to visualize the distribution.
4.	Create Bar, Line, and Pie charts for the "Quarterly Revenue" dataset and customize them with titles, axis labels, and data points.
5.	Concatenate "First Name" and "Last Name" into a "Full Name" column and extract the first three characters of the last name into a separate column.
6.	Create a Pivot Table from "Sales_Data.xlsx" to display total sales by region and product category, and filter the top 5 regions.
7.	Perform linear regression on the "House Price Data" using Excel's Analysis Toolpak and plot the regression line on a scatter chart.
8.	Identify missing values in the "Order Date" and "Order Amount" columns, and fill the dates with the average date and amounts with the median value.
9.	Sort the "Product Inventory" sheet by category and quantity in descending order, and filter only

	"Electronics" with quantities below 50.
10.	Create an interactive dashboard using Excel with charts, PivotTables, and slicers to filter data by region and product type.
11.	Record a macro to automate sorting data by "Revenue," applying conditional formatting, and generating a summary report.
12.	Perform a 6-month sales forecast using the "Monthly_Sales_Data.xlsx" with Excel's FORECAST and TREND functions, and visualize the results in a line chart.

<table>
<tr><td>Course Code</td><td colspan="5">**DSC-BCA-202A**</td></tr>
<tr><td>Category</td><td colspan="5">Discipline-Specific Courses Major Courses</td></tr>
<tr><td>Course Title</td><td colspan="5">**Computer System Architecture**</td></tr>
<tr><td rowspan="2">Scheme and Credits</td><td>L</td><td>T</td><td>P</td><td>**Credits**</td><td rowspan="2">**Semester IV**</td></tr>
<tr><td>4</td><td>0</td><td>0</td><td>**4**</td></tr>
<tr><td>Course Objectives</td><td colspan="5">The objectives of this course are
• To understand the fundamental structure and functioning of digital computers, including data representation, register transfer, and CPU organization.
• To explore instruction formats, memory and I/O organization, addressing modes, and interrupt handling mechanisms in computer systems.</td></tr>
<tr><td>Assessment</td><td colspan="5">30 Marks</td></tr>
<tr><td>End Semester Examination</td><td colspan="5">70 Marks</td></tr>
<tr><td>Total</td><td colspan="5">100 Marks</td></tr>
<tr><td>Duration of Exam</td><td colspan="5">03 Hours</td></tr>
</table>

Course Outcomes: After successful completion of this course, the students will be able to

COs	Skills Demonstrated	RBT Level
CO1	Recall concepts of computer architecture, including data representation, register operations, instruction types, and memory hierarchy.	Level 1: Remember
CO2	Explain the structure and functioning of CPU, instruction execution, addressing modes, and interrupt mechanisms.	Level 2: Understand
CO3	Apply knowledge of register transfer, instruction cycles, and addressing modes to analyze computer system behavior.	Level 3: Apply
CO4	Analyze CPU and memory operations, I/O organization, and performance aspects like cache coherence and DMA for efficient data handling.	Level 4: Analyze

Note: Examiner will set nine questions in total. Question one will be compulsory. Question one will have 7 parts of 2 marks each from all units and remaining eight questions of 14 marks each to be set by taking two questions from each unit. The students have to attempt five questions in total, first being compulsory and selecting one from each unit.

Unit-I

Basic Computer Concepts: Computer Types: Analog and Digital, Digital Computer Organization and Architecture: Functional Units, Basic Organizational Concepts, Von-Neumann Architecture, Bus Structure. Data representation: Number Systems, Fixed and Floating point representation.

Register Transfer and Micro-operations: Basic concepts and types of Registers, Register Transfer Language, Data Transfer between Registers, Bus and Memory Transfer, Micro-operations: Arithmetic, Logic and Shift Micro-operations.

Unit-II

Basic Computer Design: Instruction codes, Common Bus System Architecture, Computer Instructions: Instruction Set and Instruction Cycle. Types of Instructions: Register Reference, Memory Reference and Input-Output Reference.

Interrupts: Classifications of Interrupts: Maskable and Non maskable, Hardware and Software. Interrupt Service Routine, Context Switching, Interrupt Identification: Daisy Chaining, Polling and Vectored Interrupt. Interrupt Cycle.

Unit-III

Central Processing Unit: Introduction, General Register Organization, Stack Organization: Register and /memory Stack, Reverse Polish Notation. Instruction Formats: Three Address, Two Address, One Address and Zero Address.

Data Transfer and Manipulation: Addressing Modes: Implied, Immediate, Register Direct and Indirect Mode and Direct Address Mode. Data Transfer and Data Manipulation Instructions, Program Control, Reduced Instruction Set Computer (RISC) and Complex Instruction Set Computer (CISC).

Unit-IV

Memory Organization: Memory Hierarchy, Main Memory Technologies, Auxiliary Memory, Associative Memory: Hardware requisites, Working Principle and Operations. Cache Memory: Characteristics, Types of Mapping, Writing into Cache, Cache Coherence.

Input-Output Organization: Peripheral Devices, Input-Output Interface, Asynchronous Data Transfer, Modes of Transfer: Programmed I/O and Interrupt Initiated I/O. Concepts related to Priority Interrupt: Daisy Chaining Priority and Parallel Priority Interrupt. Direct Memory Access (DMA): DMA Controller and DMA Transfer.

Suggested Readings:

- M. Morris Mano: Computer System Architecture (3rd Edition), Prentice Hall of India.
- Carl Hamacher, Zvonko Vranesic, Safwat Zaky: Computer Organization, 5th Edition, McGraw Hill.
- William Stallings: Computer Organization and Architecture: Designing for Performance, 10th Edition, Pearson Education.
- John L. Hennessy and David A. Patterson: Computer Architecture: A Quantitative Approach, Morgan Kaufmann.
- P. Pal Chaudhuri: Computer Organization and Design, Prentice Hall of India.
- Govindarajalu B.: Computer Architecture and Organization: Design Principles and Applications, Tata McGraw Hill.

Useful Video Links:

Unit No.	**Topics**	**Links**
Unit-I	Introduction and Evolution of Computers	https://nptel.ac.in/noc/courses/noc18/SEM1/noc18-cs13/lecture1.html
	Data Representation	https://nptel.ac.in/noc/courses/noc18/SEM1/noc18-cs13/lecture2.html
	Bus and Memory Transfers	https://nptel.ac.in/noc/courses/noc18/SEM1/noc18-cs13/lecture4.html
Unit-II	Instruction Set and Cycle	https://nptel.ac.in/noc/courses/noc18/SEM1/noc18-cs13/lecture5.html
	Common Bus System and Instruction Types	https://nptel.ac.in/noc/courses/noc18/SEM1/noc18-cs13/lecture6.html
	Interrupt Handling Techniques	https://nptel.ac.in/noc/courses/noc18/SEM1/noc18-cs13/lecture7.html
Unit-III	General Register and Stack Organization	https://nptel.ac.in/noc/courses/noc18/SEM1/noc18-cs13/lecture8.html

	Addressing Modes	https://nptel.ac.in/noc/courses/noc18/SEM1/noc18-cs13/lecture9.html
	Instruction Formats and RISC vs CISC	https://nptel.ac.in/noc/courses/noc18/SEM1/noc18-cs13/lecture10.html
Unit-IV	Memory Hierarchy and Cache	https://nptel.ac.in/noc/courses/noc18/SEM1/noc18-cs13/lecture11.html
	Associative Memory	https://nptel.ac.in/noc/courses/noc18/SEM1/noc18-cs13/lecture12.html

<table>
<tr><td>Course Code</td><td colspan="5"><b>DSC-BCA-204A</b></td></tr>
<tr><td>Category</td><td colspan="5">Discipline-Specific Courses –Major Courses</td></tr>
<tr><td>Course Title</td><td colspan="5"><b>Data Communication and Computer Network</b></td></tr>
<tr><td rowspan="2">Scheme and Credits</td><td>L</td><td>T</td><td>P</td><td><b>Credits</b></td><td rowspan="2"><b>Semester IV</b></td></tr>
<tr><td>4</td><td>0</td><td>0</td><td><b>4</b></td></tr>
<tr><td>Course Objectives</td><td colspan="5">The objectives of this course are
• To understand the fundamental principles of data communication, network models, transmission media, and protocols across various network layers.
• To explore network architectures, protocols, and mechanisms in the data link, network, transport, and application layers for effective communication.</td></tr>
<tr><td>Assessment</td><td colspan="5">30 Marks</td></tr>
<tr><td>End Semester Examination</td><td colspan="5">70 Marks</td></tr>
<tr><td>Total</td><td colspan="5">100 Marks</td></tr>
<tr><td>Duration of Exam</td><td colspan="5">03 Hours</td></tr>
</table>

Course Outcomes: After successful completion of this course, the students will be able to

COs	Skills Demonstrated	RBT Level
CO1	Recall key concepts of data transmission, network models, transmission media, and basic network protocols.	Level 1: Remember
CO2	Explain the functions and protocols of data link, network, transport, and application layers in a computer network.	Level 2: Understand
CO3	Apply network protocols and algorithms to establish connections, route packets, manage congestion, and support communication services.	Level 3: Apply
CO4	Analyze network architectures, communication techniques, and protocol mechanisms for performance and efficiency in network operations.	Level 4: Analyze

Note: Examiner will set nine questions in total. Question one will be compulsory. Question one will have 7 parts of 2 marks each from all units and remaining eight questions of 14 marks each to be set by taking two questions from each unit. The students have to attempt five questions in total, first being compulsory and selecting one from each unit.

Unit-I

Introduction: Data Transmission concepts, transmission impairments, switching, modulation, multiplexing.
Network Hardware: LAN, MAN, WAN, Wireless networks, Internet-works.
Network Software: Layer, Protocols, interfaces and services.
Reference Models: OSI, TCP/IP and their comparison

Transmission Media: Magnetic, twisted pair, coaxial cable, fibre optics, wireless transmission (radio, microwave, infrared).

Unit-II

Data Link Layer: Framing, Error control, sliding window protocols (one bit, Go back n, selective repeat). DLL Protocols–HDLC, PPP.

Medium Access Sub layer: Channel Allocation, MAC protocols – ALOHA, CSMA protocols, Collision free protocols, Limited Contention Protocols, Wireless LAN protocols, IEEE 802.3, 802.4, 802.5 standards and their comparison.

Unit-III

Network Layer: Design issues, Routing algorithms (shortest path, flooding, flow based, distance vector, hierarchical, broadcast, multicast, for mobile hosts), Congestion control algorithms (Leaky bucket, Token bucket, Choke Packet, Load shedding), Internetworking, IP Protocol, ARP, RARP.

Unit-IV

Transport Layer: Addressing, establishing and releasing connection, flow control, buffering, Internet Transport Protocol (TCP and UDP).

Application Layer: Domain name system, E-mail, File transfer protocol, HTTP, HTTPS, TELNET.

Suggested Readings:

- Tanenbaum, Andrew S.: Computer Networks (4th Edition), PHI.
- Gill Nasib Singh: Handbook of Computer Fundamentals, Khanna Book Publishing Company(Pvt.) Ltd, New Delhi.
- Forouzan, B. A.: Data Communications and Networking, Fourth Edition, Tata McGraw Hill.
- Douglas E. Comer: Internet Working with TCP/IP (Vol.1, 4th Edition), CPE.
- Stallings, William: Data and Computer Communications (8th Edition), PHI.
- Nance, Bary: Introduction to Networking, PHI, 4th Edition.
- Doerr Alan &Levasseur Kenneth: Applied Discrete Structures for Computer Science, Galgotia Pub. Pvt. Ltd.
- Any other book covering the contents of the subject.

Useful Video Links:

Unit No.	Topics	Links
Unit-I	Introduction to Computer Networks	https://nptel.ac.in/noc/courses/noc20/SEM1/noc20cs29/lec01.html
	OSI and TCP/IP Models	https://nptel.ac.in/noc/courses/noc20/SEM1/noc20cs29/lec04.html
Unit-II	Framing & Error Detection	https://nptel.ac.in/noc/courses/noc20/SEM1/noc20cs29/lec07.html
Unit-III	Congestion Control	https://nptel.ac.in/noc/courses/noc20/SEM1/noc20cs29/lec13.html
Unit-IV	Transport Layer (TCP & UDP)	https://nptel.ac.in/noc/courses/noc20/SEM1/noc20cs29/lec15.html

Course Code	**DSC-BCA-206A**				
Category	Discipline-Specific Courses –Major Courses				
Course Title	**Java Programming**				
Scheme and Credits	L	T	P	**Credits**	**Semester IV**
	2	0	0	**2**	
Course Objectives	The objectives of this course are • To understand the core concepts of Java programming, including object-oriented principles, language syntax, exception handling, and multithreading. • To explore Java packages, I/O streams, and advanced features like interfaces, strings, vectors, and file handling for robust application development.				
Assessment	15 Marks				
End Semester Examination	35 Marks				
Total	50 Marks				
Duration of Exam	03 Hours				

Course Outcomes: After successful completion of this course, the students will be able to

Cos	Skills Demonstrated	RBT Level
CO1	Recall the fundamental concepts of Java, including language syntax, data types, control structures, and object-oriented elements.	Level 1: Remember
CO2	Explain object-oriented principles, Java classes, exception handling, and Java standard library usage.	Level 2: Understand
CO3	Apply Java programming constructs to develop multithreaded applications, handle files and streams, and manage user-defined exceptions.	Level 3: Apply
CO4	Analyze Java code for object-oriented structure, exception flow, and thread behavior to ensure efficient application performance.	Level 4: Analyze

Note: Examiner will set nine questions in total. Question one will be compulsory. Question one will have 7 parts of 1 marks each from all units and remaining eight questions of 7 marks each to be set by taking two questions from each unit. The students have to attempt five questions in total, first being compulsory and selecting one from each unit.

Unit-I

Introduction to Java: Java Features, Java Virtual Machine (JVM), Byte code, Java API, Java Development Kit (JDK), Garbage Collection. Language Basics: Keywords, Constants, Variables and Data Types, Operators and Expressions, Decision Making, Branching and Looping.

Introducing Classes, Objects and Methods: Defining a Class, Methods Declaration, Creating Objects and accessing Class members, Constructors, Methods Overloading, Wrapper Classes, Inheritance, Methods Overriding, Final Class, variables and methods, Abstract Class and Methods, Interfaces.

Unit-II

Arrays, Strings and Vectors: Creating and using Arrays, String operations, String Buffer, String builder, and String Tokenizer class, Vector class.

Packages and Exceptions: Java API packages, Creating and using packages, static import, Exceptions handling, Types of Exceptions, multiple catch statements, 'throw' and 'throws', using 'finally' statement, Creating your own exceptions.

Unit-III

Multithreaded Programming: Single threaded and multi-threaded program, Creating threads using Thread class, Life cycle of a Thread, Stopping and blocking a Thread, getting and setting the Thread Priority, Synchronization, implementing the Runnable interface.

Managing Input/Output Streams: Concept to fstreams, Byte and Character streams, Reading and Writing from Console and Files. Input output exceptions.

Unit-IV

Applet Programming: How Applets differs from Java Application, Applet Life Cycle, APPLET Tag, Running an Applet, Passing Parameters to Applet.

Event Handling: Mechanism, The Delegation Event Model, Event Classes, Event Listener Interfaces, Adapter and inner classes.

GUI Programming: Working with Frame Window, Graphics and Text, AWT Controls and classes. Layout Managers, working with Menus.

Suggested Readings:

- Balagurusamy, E.: Programming with Java: A Primer, McGraw Hill Education.
- Schildt, Herbert: Java: The Complete Reference (11th Edition), McGraw Hill.
- Deitel & Deitel: Java: How to Program, Pearson Education.
- Bruce Eckel: Thinking in Java, Prentice Hall.
- Cay S. Horstmann, Gary Cornell: Core Java Volume I—Fundamentals, Prentice Hall.

Useful Video Links:

Unit No.	**Topics**	**Links**
Unit-I	Introduction to Java & OOP Concepts	https://archive.nptel.ac.in/courses/106/105/106105191/
Unit-II	Java Programming Steps	https://archive.nptel.ac.in/courses/106/105/106105191/
Unit-III	Encapsulation	https://archive.nptel.ac.in/courses/106/105/106105191/
Unit-IV	Multithreaded Programming	https://archive.nptel.ac.in/courses/106/105/106105191/

Course Code	**LC-BCA-208A**				
Category	Discipline-Specific Courses –Major Courses				
Course Title	**Java Programming-Lab**				
Scheme and Credits	L	T	P	**Credits**	**Semester IV**
	0	0	4	**2**	
Course Objectives	The objectives of this course are • To equip students with the knowledge and practical skills required to apply object-oriented programming principles using Java, including inheritance, polymorphism, abstraction, and exception handling. • To enable students to design and develop robust Java applications by leveraging multithreading, file I/O, user-defined packages, and standard library features.				
Assessment	15 Marks				
End Semester Examination	35 Marks				
Total	50 Marks				
Duration of Exam	03 Hours				

Course Outcomes: After successful completion of this course, the students will be able to

COs	Skills Demonstrated	RBT Level
CO1	Apply object-oriented programming concepts such as classes, objects, method overloading, method overriding, inheritance, and abstraction in Java.	Level 3: Apply
CO2	Analyze Java programs that implement exception handling, packages, and vector operations to understand program behavior and flow.	Level 4: Analyze
CO3	Evaluate multithreading, synchronization, and file I/O operations in Java to determine efficient and secure program execution.	Level 5 Evaluate
CO4	Design and develop robust and reusable Java applications by integrating object-oriented principles, custom exceptions, packages, and file handling techniques.	Level 6 Create

Exp. no.	Contents
1.	Write a Java program to define a class `Student` with data members for name and age. Include member functions to input and display the details of a student.
2.	Create a Java program to demonstrate method overloading by creating multiple methods with the same name but different parameters in a class
3.	Create a Java program to demonstrate method overriding by creating a base class `Animal` and a derived class `Dog` that overrides a method from the base class.
4.	Implement a Java program to define an abstract class `Shape` with an abstract method `draw()`. Create derived classes `Circle` and `Square` that implement the `draw()` method.
5.	Implement a Java program to demonstrate the use of the `Vector` class by performing operations like adding, removing, and displaying elements of a vector.
6.	Write a Java program to create a user-defined package and use it in another class.
7.	Create a Java program to handle multiple exceptions using try-catch blocks.
8.	Implement a Java program to demonstrate the use of the `finally` statement in exception handling.
9.	Write a Java program to create and throw a custom exception.
10.	Create a Java program to implement a multithreaded application by extending the `Thread` class.
11.	Implement a Java program to demonstrate thread synchronization.

12.	Create a Java program to read and write data to a file using `File Input Stream` and File Output Stream`.

Course Code	**DSC-BCA-210A**				
Category	Discipline-Specific Courses –Major Courses				
Course Title	**Computer Graphics**				
Scheme and Credits	L	T	P	**Credits**	**Semester IV**
	3	0	0	**3**	
Course Objectives	The objectives of this course are • To understand the core concepts, mathematical foundations, and algorithms used in 2D and 3D computer graphics. • To explore various techniques including transformations, clipping, shading, and rendering to create and manipulate graphical content.				
Assessment	25 Marks				
End Semester Examination	50 Marks				
Total	75 Marks				
Duration of Exam	03 Hours				

Course Outcomes: After successful completion of this course, the students will be able to

COs	**Skills Demonstrated**	**RBT Level**
CO1	Recall basic concepts, coordinate systems, and fundamental graphics algorithms used in 2D and 3D computer graphics.	Level 1: Remember
CO2	Explain drawing techniques, color models, transformations, and clipping methods in both 2D and 3D environments.	Level 2: Understand
CO3	Apply transformation algorithms, clipping techniques, and projection models to implement graphical rendering and image processing tasks.	Level 3: Apply
CO4	Analyze rendering techniques, lighting models, and hidden surface algorithms to enhance visual realism in computer-generated imagery.	Level 4: Analyze

Note: Examiner will set nine questions in total. Question one will be compulsory. Question one will have 5 parts of 2 marks each from all units and remaining eight questions of 10 marks each to be set by taking two questions from each unit. The students have to attempt five questions in total, first being compulsory and selecting one from each unit.

Unit-I

Introduction to Computer Graphics: Overview of computer graphics and its applications, Historical development, Application areas e.g. entertainment, scientific visualization, user interfaces, Basics of raster and vector graphics, differences between raster and vector graphics, Representation and storage formats, Graphics pipeline and rendering process, stages of the graphics pipeline (modelling, transformation, rendering), concepts of rendering primitives, vertices, and fragments.

Graphics primitives: Coordinate systems, types of primitives (points, lines, polygons), Cartesian and homogeneous coordinate systems, transformation matrices and their applications

Unit-II

2D Graphics Programming: Pixel operations and drawing algorithms, color models (RGB, CMYK, HSL), pixel operations (blending, interpolation), Line drawing algorithms (Digital differential analyzer algorithm, Bresenham's line drawing algorithm), Circle drawing algorithms (Midpoint circle algorithm, Bresenham's circle drawing algorithm).

2D Transformations: translation, rotation, scaling in 2D space, mirror reflection basic and advance problems involving composite operations.

Windowing techniques: basic terminologies-window, viewport, clipping window, region codes.

Unit-III

Clipping techniques: Mathematics of point clipping, line clipping (midpoint subdivision method, Cohen-Sutherland line clipping algorithm), polygon clipping (Sutherland-Hodgman polygon clipping algorithm)
3D Graphics Programming: 3D transformations, Translation, Rotation, Scaling in 3D space, Homogeneous coordinates and transformations.

Projection techniques: Orthographic projection, Perspective projection and vanishing points

Unit-IV

Lighting and shading models: Phong reflection model, Gouraud and Phong shading techniques
Hidden surface removal algorithms: Z-buffer algorithm, Scan line algorithm for hidden surface removal

Texture mapping and image rendering: Texture coordinates and mapping techniques, Texture filtering and mipmapping.

Suggested Readings:

- J. F. Hughes, A. van Dam, M. McGuire, D. F. Sklar, and J. D. Foley: Computer Graphics: Principles and Practice, Addison-Wesley.
- D. Shreiner, G. Sellers, J. M. Kessenich, and B. M. Licea-Kane: OpenGL Programming Guide: The Official Guide to Learning OpenGL, Addison-Wesley.
- D. Hearn and M. P. Baker: Computer Graphics with OpenGL, Pearson.
- E. Angel and D. Shreiner: Interactive Computer Graphics: A Top-Down Approach with WebGL, Addison-Wesley.
- F. Klawonn: Introduction to Computer Graphics: Using Java 2D and 3D, Springer.

Useful Video Links:

Unit No.	Topics	Links
Unit-I	Introduction to Computer Graphics	https://nptel.ac.in/noc/courses/noc19/SEM2/noc19cs34/lec1.html
	Raster Graphics	https://nptel.ac.in/noc/courses/noc19/SEM2/noc19cs34/lec2.html
Unit-II	2D Transformations	https://nptel.ac.in/noc/courses/noc19/SEM2/noc19cs34/lec6.html
	Composite Transformations	https://nptel.ac.in/noc/courses/noc19/SEM2/noc19cs34/lec7.html
Unit-III	Clipping	https://nptel.ac.in/noc/courses/noc19/SEM2/noc19cs34/lec8.html
	Polygon Clipping	https://nptel.ac.in/noc/courses/noc19/SEM2/noc19cs34/lec9.html
Unit-IV	Illumination Models	https://nptel.ac.in/noc/courses/noc19/SEM2/noc19cs34/lec17.html
	Shading Techniques	https://nptel.ac.in/noc/courses/noc19/SEM2/noc19cs34/lec18.html

Course Code	**LC-BCA-212A**				
Category	Discipline-Specific Courses –Major Courses				
Course Title	**Computer Graphics Lab**				
Scheme and Credits	L	T	P	**Credits**	**Semester IV**
	0	0	2	**1**	
Course Objectives	The objectives of this course are • To develop a foundational understanding of 2D graphics concepts and algorithms such as line and circle drawing, transformations, and clipping techniques. • To enable students to apply and implement computer graphics algorithms for visualizing geometric transformations, object rendering, and graphical manipulations using programming.				
Assessment	10 Marks				
End Semester Examination	15 Marks				
Total	25 Marks				
Duration of Exam	03 Hours				

Course Outcomes: After successful completion of this course, the students will be able to

COs	Skills Demonstrated	RBT Level
CO1	Apply basic computer graphics algorithms such as DDA, Bresenham’s line and circle drawing to construct and render 2D primitives.	Level 3: Apply
CO2	Implement 2D geometric transformations such as translation, rotation, scaling, reflection, and shearing on graphical objects.	Level 4: Analyze
CO3	Analyze and compare different line and polygon clipping algorithms, and implement solutions to clip graphics primitives within a defined window.	Level 5 Evaluate
CO4	Evaluate the effect of various transformations and coloring techniques in rendering graphical scenes.	Level 6 Create

Exp. no.	Contents
1.	Implement a program to define and use basic graphics primitives such as points, lines, and polygons.
2.	Implement a program to draw a line using the Digital Differential Analyzer (DDA) algorithm.
3.	Create a program to draw a line using Bresenham's line drawing algorithm.
4.	Write a program to draw a circle using the Midpoint Circle algorithm.
5.	Implement a program to draw a circle using Bresenham's circle drawing algorithm.
6.	Create a program to perform 2D transformations: translation, rotation, and scaling on a given set of points.
7.	Write a program to perform mirror reflection of a 2D shape across the x-axis and y-axis.
8.	Create a program to draw basic two-dimensional objects like rectangles, triangles, and polygons using inbuilt functions
9.	Write a program to implement the Cohen-Sutherland line clipping algorithm to clip a line within a rectangular clipping window.
10.	Write a program to apply a shearing transformation to a 2D object (like a square or triangle) and display the original and transformed objects
11.	Write a program to apply various coloring techniques to 2D pictures..

12.	Write a program to implement line clipping using the Cohen-Sutherland line clipping algorithm

<table>
<tr><td>Course Code</td><td colspan="5">**MV-BCA-214A**</td></tr>
<tr><td>Category</td><td colspan="5">Minor Vocational</td></tr>
<tr><td>Course Title</td><td colspan="5">**Web Development-II**</td></tr>
<tr><td rowspan="2">Scheme and Credits</td><td>L</td><td>T</td><td>P</td><td>**Credits**</td><td rowspan="2">**Semester IV**</td></tr>
<tr><td>2</td><td>0</td><td>0</td><td>**2**</td></tr>
<tr><td>Course Objectives</td><td colspan="5">The objectives of this course are
• Understand the fundamentals of backend development using Node.js, including asynchronous programming and the event-driven architecture.
• Learn to develop scalable and modular backend applications using Express.js and implement routing and middleware logic.</td></tr>
<tr><td>Assessment</td><td colspan="5">-</td></tr>
<tr><td>End Semester Examination</td><td colspan="5">25 Marks</td></tr>
<tr><td>Total</td><td colspan="5">25 Marks</td></tr>
<tr><td>Duration of Exam</td><td colspan="5">03 Hours</td></tr>
</table>

Course Outcomes: After successful completion of this course, the students will be able to

COs	Skills Demonstrated	RBT Level
CO1	Recall fundamental concepts of backend development including Node.js runtime, event-driven architecture, and the difference between frontend and backend systems.	Level 1: Remember
CO2	Explain the use of Express.js, routing mechanisms, middleware functionalities, and the structure of modular backend applications.	Level 2: Understand
CO3	Apply backend development techniques to create RESTful APIs, integrate MongoDB using Mongoose, and manage data operations with validation and error handling.	Level 3: Apply
CO4	Analyze backend system requirements to implement authentication, secure APIs, and enable real-time communication using WebSockets and deployment practices.	Level 4: Analyze

Note: Examiner will set nine questions in total. Question one will be compulsory. Question one will have 5 parts of 1 marks each from all units and remaining eight questions of 5 marks each to be set by taking two questions from each unit. The students have to attempt five questions in total, first being compulsory and selecting one from each unit.

Unit-I

Backend Overview and Node.js Runtime Fundamentals: Introduction to backend development, differences between frontend and backend, key features of Node.js in server-side development. Understanding event-driven architecture and non-blocking I/O model in Node.js.
Installation and Project Setup: Installing Node.js and npm. Creating backend projects using npm init. Understanding Node.js runtime, REPL, and writing backend scripts.
Asynchronous Programming in Node.js: Introduction to callbacks, promises, and async/await. Understanding the event loop and how asynchronous execution improves performance.

Unit-II

Express.js Framework and Routing Mechanism: Introduction to Express.js. Creating RESTful routes with GET, POST, PUT, DELETE methods. Route parameters, query strings, and router-level middleware.
Middleware in Express.js: Using built-in, third-party, and custom middleware for handling requests, logging, authentication, and error responses.

Project Structure and Static Files: Organizing backend code into controllers, routes, and services. Serving static files and managing modular code for scalability.

Unit-III

Database Integration with MongoDB: Introduction to NoSQL databases and MongoDB. Installing MongoDB locally and using cloud services like MongoDB Atlas.
Using Mongoose for Data Handling: Creating schemas and models using Mongoose. Performing CRUD operations. Implementing data validation, error handling, and defining schema relationships using population.
Version Control and Development Tools: Using Git for version control. Basic commands, creating repositories, .gitignore usage, collaboration. Using Nodemon and other tools for live-reloading and automation.

Unit-IV

Authentication and Security in Backend: Implementing user registration and login using Express and MongoDB. Password hashing with bcrypt. Token-based authentication using JSON Web Tokens (JWT).
API Protection and Deployment Practices: Route protection using middleware, validating inputs, setting up environment variables using .env. Deploying backend projects using platforms like Render, Railway, or Vercel.
Real-Time Communication and Mini Project: Implementing real-time features using WebSockets and Socket.IO. Integrating backend with frontend using fetch/axios. Final mini project showcasing full-stack backend integration with authentication, CRUD, and real-time functionality.

Suggested Readings:

- Sharma, Narayan Prusty: Learning Node.js Development, Packt Publishing.
- Valeri Karpov & Bret McLaughlin: Professional JavaScript for Web Developers, Wrox Press.
- Ethan Brown: Web Development with Node and Express (2nd Edition), O'Reilly Media.
- Brad Traversy: Modern Full-Stack Development, Packt Publishing.
- Saswat Priyadarshi: Mastering Node.js: Develop and deploy scalable real-time apps, Packt Publishing.
- MongoDB, Inc.: The Little MongoDB Book, Open-source Publication.
- Any other book covering the contents of the subject.

Useful Video Links:

Unit No.	Topics	Links
Unit-I	Introduction to Node.js and Backend Development	https://nptel.ac.in/courses/106/106/106106222/
	Event-Driven Architecture and Non-Blocking I/O	https://nptel.ac.in/courses/106/106/106106222/
Unit-II	Introduction to Express.js	https://nptel.ac.in/courses/106/106/106106222/
	Middleware in Express.js	https://nptel.ac.in/courses/106/106/106106222/
Unit-III	Introduction to MongoDB and Mongoose	https://nptel.ac.in/courses/106/106/106106222/
	CRUD Operations and Schema Relationships	https://nptel.ac.in/courses/106/106/106106222/
Unit-IV	User Authentication with JWT	https://nptel.ac.in/courses/106/106/106106222/
	Real-Time Communication with Socket.IO	https://nptel.ac.in/courses/106/106/106106222/

<table>
<tr><td>Course Code</td><td colspan="5"><b>LC-BCA-216A</b></td></tr>
<tr><td>Category</td><td colspan="5">Minor Vocational</td></tr>
<tr><td>Course Title</td><td colspan="5"><b>Web Development-II Lab</b></td></tr>
<tr><td rowspan="2">Scheme and Credits</td><td>L</td><td>T</td><td>P</td><td><b>Credits</b></td><td rowspan="2"><b>Semester IV</b></td></tr>
<tr><td>0</td><td>0</td><td>4</td><td><b>2</b></td></tr>
<tr><td>Course Objectives</td><td colspan="5">The objectives of this course are
• Enable students to set up and configure backend environments using Node.js and Express.js.
• Teach the development of RESTful APIs with proper routing, middleware, and data persistence using MongoDB and Mongoose.</td></tr>
<tr><td>Assessment</td><td colspan="5">15 Marks</td></tr>
<tr><td>End Semester Examination</td><td colspan="5">35 Marks</td></tr>
<tr><td>Total</td><td colspan="5">50 Marks</td></tr>
<tr><td>Duration of Exam</td><td colspan="5">03 Hours</td></tr>
</table>

Course Outcomes: After successful completion of this course, the students will be able to

COs	**Skills Demonstrated**	**RBT Level**
CO1	Apply Node.js and Express.js to build RESTful APIs with routing, middleware integration, and MongoDB connectivity using Mongoose.	Level 3: Apply
CO2	Analyze backend application structure, route protection mechanisms, and middleware to manage authentication and error handling effectively.	Level 4: Analyze
CO3	Evaluate backend services for efficiency, maintainability, and security, including use of .env configuration and JWT-based authentication.	Level 5 Evaluate
CO4	Design, build, and deploy a full-stack application integrating real-time communication using WebSocket, backend API, and frontend interface on a cloud platform.	Level 6 Create

Exp. no.	**Contents**
1.	Install and configure Node.js, npm, and set up backend project using npm init.
2.	Create a basic Express.js server with a homepage route.
3.	Implement RESTful routes using Express for a resource (e.g., users).
4.	Integrate MongoDB and perform CRUD operations using Mongoose.
5.	Design a schema and model for a blog or user profile system.
6.	Use middleware to log request info and handle errors.
7.	Implement JWT-based login and signup routes with hashed passwords.
8.	Create a protected route that requires authentication.
9.	Upload environment variables using .env and use them in config.
10.	Build a WebSocket-based real-time chat feature using Socket.IO.
11.	Connect backend API to frontend UI (fetch/display data).
12.	Develop and deploy a full-stack mini project on a cloud platform.