1. Introduction to Java

- Features of Java
- Java Virtual Machine (JVM), JRE, and JDK
- Java's Platform Independence
- Writing and Executing Your First Java Program
- Java Development Environment Setup

---

2. Basics of Java Programming

- Data Types, Variables, and Constants
- Operators: Arithmetic, Relational, Logical, Bitwise
- Control Statements: if, else, switch
- Looping Constructs: for, while, do-while
- Input/Output: Using Scanner

---

3. Object-Oriented Programming (OOP)

- **Classes and Objects**
  - Defining Classes
  - Creating Objects
  - Constructors
- **Inheritance**
  - Types of Inheritance
  - super and this Keywords
- **Polymorphism**
  - Method Overloading and Method Overriding
- **Abstraction**
  - Abstract Classes and Methods
  - Interfaces
- **Encapsulation**
  - Access Modifiers (private, public, protected, default)

---

4. Java Core Concepts

- **Packages**

    - Built-in Packages (java.util, java.io)
    - User-defined Packages
- **Exception Handling**
    - try, catch, finally
    - throw and throws
    - Custom Exceptions
- **Static Keyword**
    - Static Variables, Methods, and Blocks
- **Final Keyword**
    - Final Variables, Methods, and Classes

---

5. Strings and Arrays

- String Basics
- String Manipulation (String, StringBuilder, StringBuffer)
- Arrays
    - Single-dimensional Arrays
    - Multi-dimensional Arrays
    - Array Methods

---

6. Collections Framework

- Introduction to Collections
- Common Interfaces: List, Set, Map
- Classes: ArrayList, LinkedList, HashMap, HashSet
- Iterators and Enhanced for Loop

---

7. Input/Output (I/O) in Java

- File Handling
    - Reading and Writing Files
    - Working with FileInputStream and FileOutputStream
- Serialization and Deserialization

---

8. Multithreading

- Threads and the Runnable Interface
- Thread Lifecycle
- Synchronization and Locks
- Inter-thread Communication

---

9. Miscellaneous Topics

- Inner Classes (Nested Classes)
- Enumerations (enum)
- Annotations
- Java Memory Management
    - Garbage Collection
- Basic Networking with Socket Programming

---

10. Mini Project

- Develop a small application using the above concepts, such as:
    - Enterprise resource planning

1. Introduction to HTML

- What is HTML?
- History and Evolution of HTML
- HTML vs. XHTML
- Tools to Write HTML Code (Text Editor, IDE)
- Basic Structure of an HTML Document
- HTML5 Overview

---

2. Basic HTML Tags

- **Document Structure Tags**
  - <html>, <head>, <body>
- **Text Formatting Tags**
  - <h1> to <h6> (Headings)
  - <p> (Paragraph),
 (Line Break)
  - <b> (Bold), <i> (Italic), <u> (Underline)
  - , , <mark>, <small>, <sub>, <sup>
- **Lists**
  - Ordered List (<ol>)
  - Unordered List (<ul>)
  - Definition List (<dl>, <dt>, <dd>)

---

3. Working with Links and Images

- **Hyperlinks**
  - <a> Tag and href Attribute
  - Internal Links, External Links, Anchor Links
- **Images**
  - <img> Tag and Attributes (src, alt, width, height)
  - Image Links

---

4. Tables

- Creating Tables with <table>
- Table Elements: <tr>, <th>, <td>

- Merging Cells: colspan and rowspan
- Adding Borders, Padding, and Spacing

---

5. Forms and Input

- <form> Tag and Attributes (action, method)
- Form Elements
    - Text Input: <input type="text">
    - Password Input: <input type="password">
    - Radio Buttons: <input type="radio">
    - Checkboxes: <input type="checkbox">
    - Dropdowns: <select> and <option>
    - Textarea: <textarea>
    - Submit and Reset Buttons: <input type="submit">, <input type="reset">
- Form Validation (Basic Attributes: required, pattern, etc.)

---

6. Multimedia in HTML

- Adding Audio
    - <audio> Tag and Attributes (controls, autoplay, loop)
- Adding Video
    - <video> Tag and Attributes (controls, autoplay, loop, poster)
- Embedding YouTube Videos (Using <iframe>)

---

7. Semantic HTML

- Importance of Semantic HTML
- Semantic Tags
    - <header>, <nav>, <section>, <article>, <aside>, <footer>
    - <main>, <figure>, <figcaption>, <time>, <mark>

---

8. HTML Attributes

- Global Attributes (e.g., id, class, style, title)
- Event Attributes (e.g., onclick, onmouseover, onkeydown)

---

9. Introduction to CSS in HTML

- Inline CSS (style Attribute)
- Internal CSS (<style> Tag)
- Basic CSS Properties (Colors, Fonts, Margins, Padding)
- Linking External CSS (<link> Tag)

---

10. Miscellaneous Topics

- Meta Tags (<meta>) for SEO and Page Settings
- Favicons
- Comments in HTML (<!-- -->)
- Doctype Declaration (<!DOCTYPE>)

---

11. Mini Project

- Build a static web page or website using the above concepts:
    - Personal Portfolio
    - Basic Registration Form
    - A Simple Product Page

1. Introduction to CSS

- What is CSS?
- Role of CSS in Web Development
- Types of CSS:
    - Inline CSS
    - Internal CSS
    - External CSS
- Advantages of CSS
- Syntax of CSS
- How to Link CSS to HTML (<link> Tag)

---

2. Selectors in CSS

Basic Selectors

- Universal Selector (*)
- Type Selector (element)
- ID Selector (#id)
- Class Selector (.class)

Group Selectors

- Combining Selectors (e.g., h1, p {})

Combinators

- Descendant Selector (A B)
- Child Selector (A > B)
- Adjacent Sibling Selector (A + B)
- General Sibling Selector (A ~ B)

Pseudo-Classes

- :hover, :focus, :first-child, :last-child, :nth-child(n)

Pseudo-Elements

- ::before, ::after, ::first-letter, ::first-line

---

3. CSS Properties

Text and Font Properties

- Font Family (font-family)

- Font Size (font-size)
- Font Style (font-style)
- Font Weight (font-weight)
- Text Alignment (text-align)
- Text Decoration (text-decoration)
- Line Height (line-height)
- Letter Spacing (letter-spacing)
- Word Spacing (word-spacing)

Color and Background Properties

- Text Color (color)
- Background Color (background-color)
- Background Images (background-image)
- Background Repeat (background-repeat)
- Background Position (background-position)
- Background Attachment (background-attachment)

Box Model Properties

- Margin (margin, margin-top, margin-right, etc.)
- Padding (padding, padding-left, padding-bottom, etc.)
- Borders (border, border-style, border-color, border-radius)
- Width and Height (width, height, max-width, min-height)

Positioning Properties

- Static, Relative, Absolute, and Fixed Positioning (position)
- Z-index (z-index)
- Overflow (overflow)

---

4. Layout with CSS

Display and Visibility

- display: block, inline, inline-block, none
- visibility: visible, hidden

Flexbox

- Container Properties:
    - display: flex

    - flex-direction, justify-content, align-items, align-content
- Child Properties:
    - flex-grow, flex-shrink, flex-basis
    - align-self

Grid

- Defining a Grid Container (display: grid)
- Defining Rows and Columns (grid-template-rows, grid-template-columns)
- Grid Gap (gap, row-gap, column-gap)
- Placing Items (grid-row, grid-column)

---

5. Advanced CSS Concepts

- **CSS Variables**
    - Declaring and Using Variables (--variable-name)
- **CSS Media Queries**
    - Responsive Design with @media
- **CSS Transitions**
    - Adding Effects (transition-property, transition-duration)
- **CSS Animations**
    - Defining Keyframes (@keyframes)
    - Using animation-name, animation-duration, animation-iteration-count
- **CSS Shadows**
    - Text Shadow (text-shadow)
    - Box Shadow (box-shadow)

---

6. CSS Units

- Absolute Units (px, cm, mm, in)
- Relative Units (em, rem, %, vw, vh)

---

7. CSS Best Practices

- Using Reset CSS
- Importance of Class and ID Naming Conventions
- Organizing CSS Files

- Avoiding Inline CSS for Scalability

---

8. Mini Project

- Create a responsive webpage with the following features:
    - Header, Footer, and Sidebar Layout
    - Navigation Bar with Hover Effects
    - Styled Form and Buttons
    - Animated Elements (e.g., a banner or call-to-action)

1. Introduction to SQL

- What is SQL?
- Importance of SQL in Databases
- Types of Databases (Relational, Non-relational)
- Overview of Relational Database Management Systems (RDBMS)
- Popular SQL Platforms (MySQL, PostgreSQL, SQL Server, Oracle)

---

2. Basics of SQL

- SQL Syntax and Structure
- Understanding Tables, Rows, and Columns
- Creating a Database (CREATE DATABASE)
- Using a Database (USE DATABASE)

---

3. Data Definition Language (DDL)

- **Creating Tables**
  - CREATE TABLE Syntax
  - Data Types (e.g., INT, VARCHAR, DATE, BOOLEAN)
- **Altering Tables**
  - ALTER TABLE (Add, Modify, Drop Columns)
- **Dropping Tables**
  - DROP TABLE
- **Truncating Tables**
  - TRUNCATE TABLE

---

4. Data Manipulation Language (DML)

- **Inserting Data**
  - INSERT INTO Statement
  - Inserting Multiple Rows
- **Updating Data**
  - UPDATE Statement
- **Deleting Data**
  - DELETE FROM Statement

---

5. Data Querying with SELECT

- Basic SELECT Statement
- Selecting Specific Columns
- Using Aliases (AS)
- Concatenating Columns
- Eliminating Duplicates (DISTINCT)

---

6. Filtering Data

- **WHERE Clause**
  - Comparison Operators (=, !=, >, <, >=, <=)
  - Logical Operators (AND, OR, NOT)
- **BETWEEN, IN, and LIKE**
  - Range Filtering (BETWEEN)
  - Matching Values (IN)
  - Pattern Matching (LIKE with % and _)

---

7. Sorting and Limiting Data

- Sorting Results (ORDER BY)
  - Ascending (ASC) and Descending (DESC)
- Limiting Results (LIMIT or TOP)

---

8. Aggregate Functions

- **Basic Aggregate Functions**
  - COUNT(), SUM(), AVG(), MIN(), MAX()
- **GROUP BY**
  - Grouping Data with Aggregate Functions
- **HAVING**
  - Filtering Groups

---

9. SQL Joins

- Introduction to Joins

- **Types of Joins**
    - Inner Join
    - Left (Outer) Join
    - Right (Outer) Join
    - Full (Outer) Join
- Self Joins

---

10. Working with Subqueries

- Inline Subqueries
- Correlated Subqueries
- Using Subqueries with SELECT, FROM, WHERE

---

11. Set Operations

- **UNION and UNION ALL**
- **INTERSECT** (if supported by the database)
- **EXCEPT or MINUS**

---

12. Data Integrity and Constraints

- **Primary Key**
- **Foreign Key**
- **Unique Key**
- **NOT NULL**
- **Default Values**
- **Check Constraints**

---

13. Indexing and Performance

- What is an Index?
- Creating Indexes (CREATE INDEX)
- Removing Indexes (DROP INDEX)
- Advantages and Disadvantages of Indexing

---

14. SQL Functions

- **String Functions**
  - UPPER(), LOWER(), CONCAT(), SUBSTRING(), LENGTH()
- **Date Functions**
  - NOW(), CURDATE(), DATEADD(), DATEDIFF()
- **Mathematical Functions**
  - ROUND(), ABS(), CEIL(), FLOOR()

---

15. Views

- Creating Views (CREATE VIEW)
- Updating Views
- Dropping Views (DROP VIEW)

1. Introduction to JavaScript

- What is JavaScript?
- Role of JavaScript in Web Development
- Embedding JavaScript in HTML:
    - Inline Script
    - Internal Script (<script> Tag)
    - External Script (<script src="file.js"></script>)
- JavaScript in the Browser vs. Node.js Overview

---

2. Basics of JavaScript

- **Variables and Constants**
    - var, let, const
- **Data Types**
    - Primitive: String, Number, Boolean, Null, Undefined, Symbol
    - Non-Primitive: Object, Array
- **Operators**
    - Arithmetic Operators (+, -, *, /, %)
    - Comparison Operators (==, ===, !=, !==, >, <, >=, <=)
    - Logical Operators (&&, ||, !)
    - Assignment Operators (=, +=, -=, etc.)
    - Ternary Operator (condition ? true : false)

---

3. Control Structures

- **Conditional Statements**
    - if, else if, else
    - switch Statement
- **Loops**
    - for, while, do-while
    - Loop Control: break and continue

---

4. Functions

- Defining and Calling Functions

    - Function Declaration
    - Function Expressions
    - Arrow Functions (=>)
- Parameters and Arguments
- Default Parameters
- Returning Values
- Scope of Variables: Global, Local, and Block Scope

---

5. JavaScript Objects

- Creating and Accessing Objects
- Adding, Updating, and Deleting Properties
- **Built-in Objects**
    - Math (Math.random(), Math.floor(), etc.)
    - Date (new Date())
    - JSON (JSON.stringify(), JSON.parse())
- Object Methods (this Keyword)

---

6. Arrays in JavaScript

- Creating Arrays
- Array Methods
    - Adding/Removing: push(), pop(), shift(), unshift()
    - Iterating: forEach(), map(), filter(), reduce()
    - Searching: indexOf(), find(), findIndex()
    - Sorting: sort(), reverse()

---

7. Events and DOM Manipulation

- **Event Handling**
    - Inline Events
    - Event Listeners (addEventListener, removeEventListener)
    - Common Events: onclick, onmouseover, onkeydown, onload
- **DOM (Document Object Model)**
    - Selecting Elements

        - getElementById, getElementsByClassName, querySelector
    - Modifying Elements
        - Changing Text/HTML Content (innerText, innerHTML)
        - Changing Styles (style Attribute)
    - Adding and Removing Elements
        - appendChild, removeChild, createElement
    - Working with Attributes
        - getAttribute, setAttribute

---

8. Error Handling

- try, catch, finally Statements
- Throwing Custom Errors (throw)

---

9. JavaScript Core Concepts

- **Hoisting**
- **Closures**
- **Promises**
    - Introduction to Asynchronous Programming
    - then() and catch()
- **Callback Functions**
- **Async/Await**

---

10. JavaScript in Browser

- **Timers**
    - setTimeout(), setInterval(), clearTimeout(), clearInterval()
- **Browser APIs**
    - alert(), prompt(), confirm()
    - Working with Local Storage and Session Storage
    - Navigator and Geolocation API

---

11. Debugging JavaScript

- Using console.log() and Other Console Methods
- JavaScript Debugger Tools in the Browser

---

12. Mini Project

- Create a basic interactive project, such as:
    - Dynamic Form Validation

1. Introduction to React

- What is React?
- Features and Advantages of React
- Virtual DOM vs. Real DOM
- React vs. Other Frameworks
- Setting Up a React Environment:
    - Using create-react-app
    - Using Vite (optional)

---

2. React Basics

- React Components
    - Functional Components
    - Class Components (Optional for Core Level)
- JSX (JavaScript XML)
    - Embedding Expressions
    - JSX Syntax and Rules
- Props (Properties)
    - Passing and Accessing Props
    - Default Props
- State
    - Introduction to React State
    - Managing State with useState Hook

---

3. Handling Events

- Event Handling in React
    - Adding Event Listeners
    - Event Binding
- Synthetic Events vs. Native DOM Events
- Passing Parameters in Event Handlers

---

4. Conditional Rendering

- Rendering Components Based on Conditions

    - if Statements
    - Ternary Operators
    - Short-circuit Evaluation

---

5. Lists and Keys

- Rendering Lists
    - Using the map() Method
- Keys in React
    - Why Keys are Important
    - Best Practices for Keys

---

6. Forms and Input Handling

- Controlled Components
    - Handling Form Inputs with onChange
    - value Attribute in Inputs
- Uncontrolled Components (Optional)
- Form Validation Basics

---

7. React Hooks (Core Level)

- **useState**
    - Initializing State
    - Updating State
- **useEffect**
    - Side Effects in React
    - Dependency Array
    - Cleaning Up Effects
- **Custom Hooks** (Introduction, Optional)

---

8. React Router (Basic)

- Setting Up react-router-dom
- Creating Routes (<Route> and <Routes>)
- Navigation with <Link> and <NavLink>

- Understanding useNavigate

---

9. Styling in React

- Inline Styles
- CSS Modules
- Styled Components (Optional)
- Using Third-Party Libraries (e.g., Bootstrap, Tailwind CSS)

---

10. Context API (Core Level)

- Context for Global State Management
- Creating Context
- Using useContext Hook
- Provider and Consumer Pattern

---

11. Debugging and Tools

- Using console.log() in React
- React Developer Tools in Browser
- Error Boundaries (Optional for Core Level)

---

11. Mini Project

- Build a small project using the above concepts:
    - A Todo List Application
    - Weather App (using an API)
    - Simple Counter App with Increment/Decrement

1. Introduction to PHP

- What is PHP?
- Features and Advantages of PHP
- Installing PHP and Setting Up Environment (XAMPP/WAMP/LAMP)
- PHP vs. Other Server-Side Languages
- Writing Your First PHP Script
- Embedding PHP in HTML

---

2. PHP Basics

- PHP Syntax and Rules
- PHP Tags (<?php ?>)
- Comments in PHP
- Variables
    - Declaring Variables
    - Variable Types (Strings, Integers, Floats, Booleans, Arrays, Objects, NULL)
    - Variable Scope (Global, Local, Static)

---

3. PHP Data Types and Operators

- Data Types in PHP
- Type Casting
- PHP Operators
    - Arithmetic Operators
    - Assignment Operators
    - Comparison Operators
    - Logical Operators
    - Increment/Decrement Operators
    - String Operators

---

4. Control Structures

- **Conditional Statements**
    - if, else if, else
    - switch

- **Loops**
    - for
    - while
    - do-while
    - foreach (For Arrays)
- Using break and continue

---

5. PHP Functions

- Defining and Calling Functions
- Function Parameters and Return Values
- Default and Optional Parameters
- Variable Functions
- Anonymous Functions (Closures)

---

6. PHP Arrays

- Types of Arrays
    - Indexed Arrays
    - Associative Arrays
    - Multidimensional Arrays
- Array Functions
    - count(), array_merge(), array_push(), array_pop()
    - sort(), rsort(), ksort()
    - Searching Arrays (in_array(), array_search())

---

7. Working with Forms

- **HTML Forms and PHP**
    - GET vs POST Methods
    - Retrieving Form Data ($_GET, $_POST)
- Form Validation Basics
- Handling File Uploads
- Preventing Cross-Site Scripting (XSS)

---

8. PHP Strings

- String Creation and Concatenation
- Common String Functions
    - strlen(), strpos(), str_replace(), substr(), trim()
    - explode(), implode()

---

9. PHP Superglobals

- Understanding Superglobals
    - $_GET, $_POST, $_REQUEST
    - $_FILES, $_SESSION, $_COOKIE
    - $_SERVER, $_ENV

---

10. File Handling

- Opening and Reading Files (fopen(), fread())
- Writing to Files (fwrite())
- File Functions (file_exists(), filesize(), unlink())

---

11. PHP Sessions and Cookies

- Understanding Sessions
- Starting and Managing Sessions (session_start(), $_SESSION)
- Destroying Sessions
- Understanding Cookies
- Setting and Retrieving Cookies (setcookie(), $_COOKIE)

---

12. PHP and Databases

- Connecting to a Database (MySQL)
    - Using mysqli and PDO
- Performing CRUD Operations
    - SELECT, INSERT, UPDATE, DELETE Queries
- Fetching Data from Databases (fetch_assoc(), fetch_array())
- Preventing SQL Injection (Prepared Statements)

---

13. Error Handling

- Error Reporting Levels
- Handling Errors with try, catch, finally
- Custom Error Handlers (set_error_handler())

---

14. PHP OOP (Object-Oriented Programming)

- Introduction to Classes and Objects
- Creating and Using Classes
- Properties and Methods
- Constructor and Destructor
- Inheritance
- Access Modifiers (public, private, protected)

---

15. PHP Date and Time

- Getting the Current Date and Time
- Formatting Dates (date(), strtotime())
- Date Manipulation

---

16. Security in PHP

- Input Validation and Sanitization
- Preventing SQL Injection
- Preventing Cross-Site Scripting (XSS)
- Hashing Passwords (password_hash() and password_verify())

---

17. Mini Project

- Create a simple PHP project like:
    - A Contact Form with Validation
    - A CRUD Application for Managing Users
    - A Login System with Sessions and Password Hashing

1. Introduction to Python

- What is Python?
- Features of Python
- Installing Python and Setting Up the Environment
- Running Python Code (IDLE, Terminal, VSCode, Jupyter Notebook)
- Writing Your First Python Program (print("Hello, World!"))

---

2. Python Basics

- Python Syntax and Indentation
- Comments in Python
- Variables and Constants
- Basic Input and Output
    - input() Function
    - print() with Formatting (f-strings, .format())

---

3. Data Types and Operators

- **Data Types**
    - Numeric (int, float, complex)
    - Text (str)
    - Boolean (True, False)
    - Sequence (list, tuple, range)
    - Sets and Dictionaries
- **Operators**
    - Arithmetic Operators (+, -, *, /, //, %, **)
    - Comparison Operators (==, !=, >, <, >=, <=)
    - Logical Operators (and, or, not)
    - Membership Operators (in, not in)
    - Identity Operators (is, is not)

---

4. Control Structures

- **Conditional Statements**
    - if, elif, else

- **Loops**
    - for Loop (Iterating Over Sequences)
    - while Loop
    - Loop Control (break, continue, pass)

---

5. Functions

- Defining and Calling Functions
- Function Parameters and Return Values
- Default Parameters
- Keyword Arguments
- Variable-Length Arguments (*args, **kwargs)
- Scope of Variables (Global and Local Scope)

---

6. Python Strings

- Creating Strings and String Operations
- Common String Methods (upper(), lower(), strip(), replace(), find())
- String Slicing ([start:end:step])
- String Formatting (f-strings, .format(), % formatting)

---

7. Python Lists

- Creating Lists and Accessing Elements
- List Methods (append(), extend(), pop(), remove(), sort(), reverse())
- List Slicing
- List Comprehensions

---

8. Tuples

- Creating Tuples and Accessing Elements
- Immutable Nature of Tuples
- Tuple Unpacking

---

9. Dictionaries

- Creating Dictionaries

- Accessing, Adding, and Removing Elements
- Dictionary Methods (keys(), values(), items(), get())
- Dictionary Comprehensions

---

10. Sets

- Creating Sets
- Adding and Removing Elements
- Set Operations (Union, Intersection, Difference, Symmetric Difference)

---

11. File Handling

- Reading Files (open(), read(), readlines())
- Writing Files (write(), writelines())
- Appending to Files
- Closing Files (with Statement for Automatic Closing)

---

12. Error Handling

- Understanding Exceptions
- Using try, except, else, and finally
- Raising Exceptions (raise)

---

13. Object-Oriented Programming (OOP) Basics

- Introduction to Classes and Objects
- Defining Classes and Creating Objects
- Constructors (__init__)
- Instance Variables and Methods
- Class Variables and Methods (@classmethod)
- Static Methods (@staticmethod)

---

14. Modules and Packages

- Importing Built-in Modules (math, random, datetime, etc.)
- Creating and Importing Custom Modules
- Using from and import Statements

- Introduction to Packages

---

15. Python Libraries Overview

- **Standard Libraries:**
  - math (Mathematical Functions)
  - random (Generating Random Numbers)
  - os (Interacting with the Operating System)
  - sys (System-Specific Parameters and Functions)
- **Third-Party Libraries (Intro Only):**
  - numpy, pandas (Data Manipulation)
  - matplotlib (Basic Visualization)

---

16. Iterators and Generators

- Understanding Iterators (__iter__, __next__)
- Creating Generators with yield

---

17. Python Decorators

- Understanding Decorators
- Writing Simple Decorators

---

18. Mini Project

- Build a small project, such as:
  - Basic Calculator
  - To-Do List (with File Persistence)
  - Simple Quiz Application

1. Introduction to Node.js

- What is Node.js?
- Features and Advantages of Node.js
- Installing Node.js and NPM (Node Package Manager)
- Checking Node.js and NPM Versions

- Understanding the Node.js Runtime Environment

---

2. Basics of Node.js

- Writing Your First Node.js Program
- Running JavaScript in the Node.js Environment
- Using the Node.js REPL (Read-Eval-Print Loop)
- Global Objects in Node.js
    - __dirname and __filename
    - process
    - console

---

3. Modules in Node.js

- Introduction to Modules
- Built-in Modules in Node.js (Core Modules)
    - fs (File System)
    - path (Path Utilities)
    - os (Operating System Information)
    - http (HTTP Module)
- Creating Custom Modules
- Exporting and Importing Modules (module.exports and require)

---

4. File System (fs Module)

- Reading and Writing Files
    - Synchronous (readFileSync, writeFileSync)
    - Asynchronous (readFile, writeFile)
- Appending and Deleting Files
- Creating and Removing Directories

---

5. Events and Event Emitters

- Introduction to the Event-Driven Architecture
- Using the events Module
    - Creating and Using an Event Emitter

- Listening for and Emitting Events

---

6. Streams and Buffers

- What are Streams?
- Types of Streams (Readable, Writable, Duplex, Transform)
- Reading and Writing Streams
- Using Buffers to Handle Binary Data

---

7. HTTP Module

- Creating a Basic HTTP Server
- Handling Requests and Responses
- Setting Response Headers and Status Codes
- Serving HTML Content from the Server

---

8. NPM (Node Package Manager)

- What is NPM?
- Initializing a Project (npm init)
- Installing and Using Packages (npm install)
- Managing Dependencies (package.json and package-lock.json)
- Installing Global Packages

---

9. Introduction to Middleware

- What is Middleware?
- Basic Middleware Functions in Node.js

---

10. Error Handling in Node.js

- Using try-catch Blocks
- Handling Errors in Callbacks
- Handling Errors in Promises (catch and finally)