Java Full Stack Development

Month 1: Core Java + MySQL (Strong Foundation)

➤ **Core Java Programming**

- Introduction to Java, Installation (JDK, Eclipse/IntelliJ)

- First Java Program (Hello World, Compilation Process)

- Data Types, Variables, Keywords

- Operators (Arithmetic, Relational, Logical, Assignment, Unary, Bitwise)

- Input and Output in Java (Scanner Class, BufferedReader)

- Control Flow Statements

  ○ if-else

  ○ switch-case

  ○ for, while, do-while loops

  ○ break and continue

- Arrays (Single Dimensional and Multi-Dimensional)

- String Handling

  ○ String Class, StringBuilder, StringBuffer

  ○ String Methods (equals, compareTo, substring, etc.)

- Object-Oriented Programming (OOPS)

  ○ Class and Object

  ○ Methods, Constructors

- **Inheritance (IS-A relationship)**

- **Polymorphism** (Method Overloading, Method Overriding)

- **Encapsulation**

- **Abstraction (Abstract Classes & Interfaces)**

- Wrapper Classes (Integer, Character, Double, etc.)

- Exception Handling

  - try-catch-finally

  - throw, throws

  - Custom Exceptions

- Java Collections Framework

  - List (ArrayList, LinkedList)

  - Set (HashSet, TreeSet)

  - Map (HashMap, LinkedHashMap)

  - Iterator, ListIterator

- Introduction to Java 8 Features

  - Lambda Expressions

  - Functional Interfaces (Predicate, Consumer, Supplier)

  - Stream API Basics

- Mini Project 1: **Library Book Management System (Console-Based)**

➤ **SQL & MySQL Database**

- What is DBMS, RDBMS

- Database vs Table

- Introduction to MySQL Workbench

- SQL Basics

    ○ DDL: CREATE, ALTER, DROP

    ○ DML: INSERT, UPDATE, DELETE

    ○ TCL: COMMIT, ROLLBACK

- Basic SELECT Queries

- Filtering and Sorting Data (WHERE, ORDER BY, GROUP BY, HAVING)

- Functions: COUNT(), SUM(), AVG(), MIN(), MAX()

- Joins

    ○ INNER JOIN

    ○ LEFT JOIN

    ○ RIGHT JOIN

    ○ FULL JOIN

- Subqueries, Nested Queries

- Views, Indexes

- Constraints (Primary Key, Foreign Key, Unique, Not Null)

- Stored Procedures & Triggers (Introduction)

- Mini Project 2: **Student Result Management System Database**

Month 2: JDBC + Servlet + JSP (Dynamic Web Development)

➤ **JDBC (Java Database Connectivity)**

- Introduction to JDBC

- JDBC Architecture

- JDBC API: DriverManager, Connection, Statement, PreparedStatement, ResultSet

- Steps to Connect Java Program with MySQL

- Inserting, Updating, Deleting, and Fetching Records using JDBC

- Transactions in JDBC (commit & rollback)

- CallableStatement (for stored procedures)

- Batch Processing in JDBC

- Mini Project 3: **Login/Signup System (Console with Database Integration)**

➤ **Servlet Programming**

- Web Application Basics

- Introduction to Servlets

- Servlet Life Cycle

- Deploying Servlet in Apache Tomcat Server

- ServletRequest & ServletResponse

- doGet() and doPost() methods

- RequestDispatcher (forward & include)

- Session Management

  - Cookies

  - Hidden Form Fields

  - URL Rewriting

  - HttpSession

- ServletConfig & ServletContext

- Servlet Filters (Authentication Example)

➤ **JSP (JavaServer Pages)**

- Introduction to JSP

- JSP Life Cycle

- JSP Scripting Elements: Scriptlet, Expression, Declaration

- Directives: page, include, taglib

- Implicit Objects (request, response, session, application)

- JSTL (JSP Standard Tag Library) and EL (Expression Language)

- Form Handling using JSP

- MVC Architecture using Servlet + JSP

➤ **Mini Project 4: User Registration + Login Portal (JSP + Servlet + JDBC)**

Month 3: Hibernate ORM + Spring Core + Spring MVC

➤ **Hibernate (ORM Framework)**

- ORM Concept

- Hibernate Architecture

- Hibernate Setup (Maven + Hibernate + MySQL)

- Hibernate Configuration (hibernate.cfg.xml)

- Hibernate Annotations

- Entity Mapping

    - One-to-One

    - One-to-Many

    - Many-to-One

    - Many-to-Many

- Hibernate CRUD Operations

- HQL (Hibernate Query Language)

- Named Queries and Criteria API

- Caching in Hibernate (Introduction)

- Mini Project 5: **Employee Database CRUD Operations using Hibernate**

➤ **Spring Core**

- **Introduction to Spring Framework**

- Spring Modules Overview

- Dependency Injection (DI)

  - Constructor Injection

  - Setter Injection

- Inversion of Control (IoC) Container

  - BeanFactory vs ApplicationContext

- Spring Bean Life Cycle

- Autowiring

- Stereotype Annotations (@Component, @Service, @Repository)

➤ **Spring MVC**

- Spring MVC Architecture

- DispatcherServlet, HandlerMapping, Controller, ViewResolver

- Creating Controllers

- Form Handling in Spring MVC

- Data Binding and Validation

- Exception Handling (ControllerAdvice)

- Using JSP with Spring MVC

- Mini Project 6: **Simple E-commerce Website Backend (Spring MVC + Hibernate)**

Month 4: Spring Boot + Web Technologies + Final Full Stack Project

➤ **Spring Boot (Rapid Backend Development)**

- Introduction to Spring Boot

- Advantages of Spring Boot over Spring MVC

- Spring Boot Project Setup with Spring Initializr

- @SpringBootApplication

- @RestController, @RequestMapping, @GetMapping, @PostMapping

- Connecting Spring Boot Application with MySQL

- Spring Data JPA (Repository Layer)

- Pagination and Sorting

- Exception Handling in Spring Boot (Global)

- Validations with Hibernate Validator

- Application Properties (Configuration)

- Packaging and Deploying Spring Boot Application

- Introduction to Basic Security Concepts (Spring Boot Security)

  - Securing REST APIs with Basic Authentication (optional if time permits)

➤ **Frontend: Web Technologies**

➤ **HTML5**
- Structure of HTML Document

- Forms, Tables, Semantic Elements

➤ **CSS3**
- Selectors

- Box Model

- Flexbox and Grid

- Media Queries (Responsive Design)

➤ **Bootstrap (v5)**
- Grid System

- Cards, Alerts, Forms, Navbars

➤ **JavaScript**
- Variables, Functions, Arrays

- DOM Manipulation

- Event Handling

- Fetch API (Calling Backend APIs)

- Promises & Async/Await

➤ **Integration**

- Consuming Spring Boot REST API using JavaScript Fetch/Axios

- Sending Data (POST), Getting Data (GET)

- Making Frontend and Backend Connected Project

➤ **Capstone Project: Full Stack Development**

Build a Full Stack Application like:

- **Student Management System**

- **Inventory Management System**

- **Book Store**

- **Task Management Application**

Frontend: HTML5 + CSS3 + Bootstrap + JavaScript
Backend: Spring Boot (REST APIs)
Database: MySQL
ORM: Hibernate (via Spring Data JPA)

Bonus Add-ons (Recommended)
- Introduction to Git & GitHub

- Maven Basics (pom.xml explained)

- Postman (API testing tool)

- Resume Preparation and Mock Interviews

- Hosting Spring Boot app on Railway / Render.com