Java Lectureflow

Module-1) SE - Overview of IT Industry	5

- Session 1 - Welcome to the IT World Explain careers in IT (developer, tester, designer, data analyst, AI engineer). Discuss product vs service companies; real-world examples (Google, Zoho, TCS). Overview of how teams collaborate using GitHub, Slack, Jira. Trainer Demo: Show an open-source GitHub project (e.g., “Awesome-Java”) and walk through contributors’ roles. AI Tool Tip: Ask ChatGPT to list top emerging tech jobs (to discuss in class).
- Session 2: How Software Works Define software application and its components. Explain software architecture layers: UI ? Business Logic ? Database. Compare desktop, web, and mobile apps. Explain client-server communication with example (browser request ? backend ? database ? response). Trainer Demo: Use Draw.io or Lucidchart AI to visually show data flow (e.g., for “Instagram feed refresh”).
- Session 3: Internet & Networking Basics Internet Fundamentals: How computers communicate. Protocols: Define HTTP, HTTPS, FTP, SMTP (overview only, 5 mins each). Domain & DNS: Explain how domain names map to IP addresses. Demonstrate using nslookup google.com. Explain DNS hierarchy: root, TLD, domain, subdomain. IP Addressing: What is an IP (IPv4 vs IPv6)? HTTP Lifecycle: Request ? Response ? Status Codes (focus on 200, 404, 500). Trainer Demo: Use browser DevTools ? Network tab
- Session 4: Software Development Lifecycle (SDLC) Explain SDLC phases: Requirement Design Development Testing Deployment Maintenance. Discuss Agile vs Waterfall, Scrum roles (PO, Scrum Master, Dev Team). Relate each phase to a real-world app (e.g., “Spotify new feature rollout”). Trainer Demo: Show a Jira board mock-up with stories in “To Do / In Progress / Done.”
- Session 5: Git & Version Control Why version control is needed. Git concepts: Repository, Commit, Branch, Merge, Push, Pull. GitHub overview: Forks, Pull Requests, Collaboration. Trainer Demo: Initialize a local Git repo (git init, git add ., git commit) Push to GitHub. AI Integration: Use GitHub Copilot Chat to explain git errors in real-time. Depth: Students should confidently push their first small project to GitHub.

Module-2) SE- Introduction to Web Technologies - HTML and CSS	12

- Session 1: Introduction to Web Design Coverage: How browsers render HTML + CSS. Difference between front-end & back-end. Developer tools overview (Chrome DevTools). Demo: Inspect elements on Instagram home page. Depth: Conceptual understanding of DOM structure.
- Session 2: HTML Page Structure Coverage: HTML boilerplate (<!DOCTYPE>, <html>, <head>, <body>). Tags: Headings, paragraphs, lists, anchors, images, tables. Demo: Create personal “About Me” page.
- Session 3: Forms & Inputs Coverage: Form elements (input, select, textarea), labels, validation attributes. Demo: Registration form for Social Bio Link App. Depth: Show POST request preview in DevTools Network tab.
- Session 4: Semantic HTML 5 Coverage: Semantic tags (header, nav, article, footer). Demo: Rebuild a YouTube-style page layout.

- Session 5: Introduction to CSS 3 Coverage: Inline, Internal, External styles; Selectors, Colors, Font families. Demo: Color-theme switch between light and dark mode.
- Session 6: Box Model & Positioning Coverage: Margin, Padding, Border, Display properties, Position (relative, absolute, fixed). Demo: Create a profile-card layout.
- Session 7: Flexbox & Grid Coverage: CSS Flexbox (container, justify, align) and Grid (template columns & rows). Demo: Responsive gallery of social icons.
- Session 8: Responsive Design Coverage: Media queries, Viewport, Breakpoints. Demo: Make the profile card mobile-friendly.
- Session 9: Bootstrap Basics Coverage: Bootstrap Grid System, Containers, Rows, Columns. Demo: 2-column layout with profile image and link list.
- Session 10: Bootstrap Components Coverage: Navbar, Buttons, Cards, Carousel, Forms. Demo: Navbar for Social Bio Link App.
- Session 11: Bootstrap Utilities & Customization Coverage: Spacing, Typography, Colors, Custom CSS overrides. Demo: Theme switcher using Bootstrap classes.
- Session 12: Integration Review Coverage: Integrate HTML + CSS + Bootstrap for a cohesive layout. Demo: Static mock-up of Bio Link landing page.
- Mini Project 6 – “Social Bio Link Page – UI Design” Objective: Build a personal bio link page (links to Instagram, LinkedIn, Spotify, etc.) with responsive design. Trainer Notes: Guide students to implement a form that accepts link inputs and shows preview cards.

Module 3) SE - Introduction to Programming	**15**

- Session 1: What is Programming? Topics: Define “program” and “programming language.” Explain compilers and interpreters. Discuss importance of algorithms and flowcharts. Trainer Activity: Draw a flowchart for “Ordering food online.”
- Session 2: Writing Your First C Program - Topics: Setting Up Environment -Installing a C Compiler (e.g., GCC), Choosing an IDE (DevC++. VS Code, Codeblocks, etc) Basic structure of C program (header, main, return). Compile and run using GCC or VS Code terminal. Example: “Hello, TOPS Technologies!”
- Session 3: Data Types and Variables Topics: Data types (int, float, char, double). Variable naming rules. Constants, literals. Example: Store and print student’s name, age, marks.
- Session 4: Operators and Expressions Topics: Arithmetic, Logical, Relational, Increment/Decrement. Operator precedence. Example: Simple calculator for addition, subtraction, etc.
- Session 5: Conditional Statements Topics: If, Else If Ladder, Nested If, Switch Case. Example: “Study Mood Bot” – takes user’s input and prints motivation message
- Session 6: Loops (For, While, Do-While) Topics: Difference between entry-controlled and exit-controlled loops. Use cases (e.g., menu-driven programs). Example: Countdown timer using loops.
- Session 7: Nested Loops & Pattern Printing Topics: Using nested loops for grids or triangle patterns. Example: Star pyramid pattern.
- Session 8: Functions Topics: Declaration, Definition, Calling. Pass-by-value vs pass-by-reference.
- Session 9: Arrays Topics: 1D & 2D Arrays, Array Traversal. Example: Record 7-day screen time & calculate average.
- Session 10: Strings Topics: Declaring and handling strings. Common functions: strlen(), strcpy(), strcmp(). Example: Create a username from name input.

- Session 11: Pointers Topics: Pointer declaration, dereferencing, pointer arithmetic. Example: Swap numbers using pointers. Trainer Demo: Use memory diagram to explain address referencing.
- Session 12: Structures Topics: Declaring, initializing structures. Nested structures. Example: Store student details (name, marks, grade).
- Session 13: File Handling Topics: Open, write, read, close files. Modes: r, w, a. Example: Write and read “daily goals.txt.”
- Session 14: Debugging & Best Practices Topics: Common syntax/runtime errors. Importance of commenting & indentation.

Module-4) Introduction to OOPS Programming	**8**

- Session 1: Procedural vs Object Thinking Topics: Limitations of C (global data, no encapsulation). Why OOP solves these. Example: Compare “Task List” written in C vs OOP style.
- Session 2: Classes and Objects Topics: Define class & object, syntax, member functions. Example: Class Note (title, date, isDone).
- Session 3: Constructors & Destructors Topics: Default, Parameterized, Copy constructors. Destructor lifecycle. Example: Auto-save user data upon exit.
- Session 4: Inheritance Topics: Single, Multilevel, Hierarchical inheritance. Example: ContentCreator ? YouTuber / Podcaster.
- Session 5: Polymorphism Topics: Compile-time (overloading) vs Runtime (overriding). Example: uploadContent() works differently for YouTube vs Instagram.
- Session 6: Encapsulation & Abstraction Topics: Access modifiers: public, private, protected. Abstract classes & pure virtual functions (conceptual). Example: Class Platform with abstract upload() method.
- Session 7: File Handling Topics: Streams (ifstream, ofstream), reading/writing files. Example: Save and retrieve creator analytics data.

Module-5) SE - Database Management – SQL & PL/SQL	**9**

- Session 1: Introduction to Databases Trainer Coverage: What is a database, DBMS, RDBMS (with examples). Difference between MySQL, PostgreSQL, Oracle, SQLite. Tables, rows, columns, and relationships. Demo: Show data stored in MySQL Workbench.
- Session 2: SQL Basics & Commands Trainer Coverage: SQL syntax rules. DDL commands: CREATE, ALTER, DROP. Creating basic users and expenses tables. Demo: Execute commands in MySQL CLI / Workbench.
- Session 3: DML Operations Trainer Coverage: Insert, Update, Delete records. WHERE and LIMIT usage. Example: Insert 5 sample users and edit their data.
- Session 4: Data Retrieval with SELECT Trainer Coverage: SELECT queries, filtering, sorting, aliases. Aggregate functions (SUM, AVG, COUNT). Demo: Generate “total expenses” by category.
- Session 5: Joins and Relationships Trainer Coverage: Foreign keys and primary keys. Inner Join, Left Join, Right Join. Demo: Link users with their transactions table.
- Session 6: Grouping and Subqueries Trainer Coverage: GROUP BY, HAVING. Subqueries and nested queries. Example: “Show users whose average daily spend > ?500.”

- Session 7: Constraints & Views Trainer Coverage: Constraints: NOT NULL, UNIQUE, PRIMARY KEY, FOREIGN KEY. Creating and querying VIEWS. Example: Create “ActiveUserView” showing users with >5 transactions.
- Session 8: Introduction to PL/SQL Trainer Coverage: PL/SQL block structure (DECLARE, BEGIN, EXCEPTION, END). Variables and cursors. Control structures (IF, LOOP). Demo: Simple stored procedure to calculate user’s total spend.
- Session 9: Transactions & Error Handling Trainer Coverage: COMMIT, ROLLBACK, SAVEPOINT. Triggers: use case “Auto update user’s balance after expense entry.” AI Integration: Use ChatGPT to simulate test triggers and rollback scenarios.

Module 6) - Java - Core Java	**18**

- Session 1: Introduction to Java & Environment Setup History and significance of Java, JDK, JRE, JVM (runtime vs development). Installation of JDK and Eclipse/IntelliJ. Writing the first “Hello Java” program. Anatomy of a Java class: class declaration, main method, System.out.println(). Demo: “Hello Digital Wallet!” program.
- Session 2: Data Types, Variables & Operators Primitive vs Non-Primitive types. Declaring and initializing variables. Arithmetic, Logical, Relational, and Ternary operators. Type casting (implicit/explicit). Demo: Calculate interest rate on a wallet balance. Depth: Relate variables to “columns” in a database table.
- Session 3: Conditional Statements if, if-else, if-else-if, switch statements. Nested conditions and real-life logic flow. Demo: If balance < 100 ? “Low Balance Warning.” AI Tool: ChatGPT to generate alternate logic for same condition (ternary operator). Depth: Emphasize mapping decisions to SQL WHERE clauses (logical filtering).
- Session 4: Loops & Flow Control For, While, Do-While loops. Break, Continue, Nested Loops. Demo: Simulate monthly transaction summary in loop. Depth: Relate loops to “iterating through database records.”
- Session 5: Arrays Declaring and initializing arrays. Single vs Multi-dimensional arrays. Traversing arrays using loops. Demo: Store last 10 transactions and display them. Depth: Explain how arrays model database tables (rows).
- Session 6: Introduction to Object-Oriented Programming Why OOP? Advantages over procedural programming. Real-world analogy: User, Wallet, and Transaction as “objects.” Demo: Class User with name and balance attributes.
- Session 7: Classes, Objects & Methods Defining classes, creating objects. Instance vs static members. Methods: Declaration, return type, parameters. Demo: Class Wallet with addMoney() and checkBalance() methods. Depth: Relate classes to “tables” and methods to “CRUD operations.”
- Session 8: Constructors & Method Overloading Constructor types (default, parameterized, copy). Method overloading principles. Demo: Class User – multiple constructors for quick vs full registration. Depth: Link overloaded constructors to different forms of inserting records (INSERT queries).
- Session 9: Inheritance & Reusability Parent-child relationships, extends keyword. Single, Multilevel, Hierarchical inheritance. Use of super and this. Demo: User ? PremiumUser (extra cashback feature).

- Session 10: Polymorphism Method overriding, Dynamic Dispatch. Compile-time vs Runtime polymorphism. Demo: Different wallet top-up methods (Card, UPI, Crypto).
- Session 11: Encapsulation & Abstraction Access modifiers: public, private, protected, default. Getters/Setters. Abstract classes and interfaces. Demo: Abstract class Payment with abstract method processPayment(). Depth: Explain abstraction as API contracts between layers (Service ? Database).
- Session 12: String Handling String vs StringBuilder vs StringBuffer. Common methods: length, substring, equals, compareTo. Demo: Generate masked wallet IDs (e.g., “WAL1234 ? ***1234”).
- Session 13: Exception Handling – Basics Need for exception handling. Try-catch-finally, throw & throws. Demo: Throw exception if withdrawal exceeds balance.
- Session 14: Exception Handling – Advanced Multiple catch blocks, custom exceptions. Create InsufficientFundsException. Demo: Handle multiple errors in transaction flow.
- Session 15: File I/O FileInputStream, FileOutputStream, FileWriter, FileReader. Reading/writing simple text files. Demo: Save transaction logs to file. Depth: Explain how file I/O simulates basic persistence before database use.
- Session 16: Collection Framework – Part 1 Introduction to Collections API. List, Set, Map overview. Using ArrayList and HashSet. Demo: Store transaction objects in ArrayList. Depth: Relate Collections to in-memory storage before DB persistence.
- Session 17: Collection Framework – Part 2 HashMap, LinkedHashMap, Iterators. Sorting and filtering collections. Demo: Map of transaction IDs ? Transaction details. Depth: Reinforce how keys and values map to columns and records in databases.
- Session 18: Threads & Recap Thread lifecycle, Runnable interface. Use of thread for asynchronous tasks (e.g., background interest calculation). Demo: Thread to auto-update cashback after every 10 seconds. Depth: Explain parallelism in backend operations.
- Mini Project 4 – “Digital Wallet Console App” Duration: 2–3 sessions (integrated within 18 sessions) Objective: Develop a console-based “Digital Wallet” where users can create accounts, log in, check balance, deposit or spend money, and view transaction history. Core Functionalities: Registration & Login: Store users temporarily using ArrayList or HashMap. Use encapsulation to secure credentials. Transaction Management: Add/Spend funds, validate balance, update transaction lis

Module 7) Java - RDBMS & Database Programming With JDBC	**6**

- Session 1: Introduction to JDBC Trainer Coverage: What is JDBC, why it’s needed. JDBC architecture (Driver Manager, Connection, Statement). Demo: Show diagram of Java ? MySQL connection.
- Session 2: JDBC Setup Download and configure MySQL connector JAR. Establishing a connection using DriverManager.getConnection(). Handling exceptions. Demo: Create test connection program.
- Session 3: Statement & Prepared Statement Difference between Statement and PreparedStatement. Insert, Update, Delete queries using PreparedStatement. Importance of preventing SQL Injection.
- Session 4: Select Queries & ResultSet Trainer Coverage: ExecuteQuery(), traverse ResultSet, display formatted output. Example: “Show all expenses for a given user.” Discuss pagination and sorting logic.

- Session 5: Transactions & Batch Processing Trainer Coverage: Auto-commit, commit, rollback in JDBC. Executing multiple inserts in a batch. Example: Upload multiple expense records at once.
- Session 6: Integration & Error Handling Trainer Coverage: Use exception handling with JDBC (try-with-resources). Close connections safely. Overview of DatabaseMetaData and ResultSetMetaData. AI Integration: Use Copilot for JDBC template generation.
- Mini Project 5 – “Smart Expense Tracker” Objective: Integrate Java + JDBC + SQL into one mini application. Project Scope: Console app with user registration/login. Add/view/edit/delete expenses. Display monthly report (total by category). Bonus: Add “AI Tips” section that fetches budgeting tips from ChatGPT API (optional advanced trainer demo). Trainer Focus: Emphasize MVC thinking (Model = DB, Controller = JDBC, View = Console). Reinforce error handling and validation. Encourage

Module 8) Java - Web Technologies In Java	**12**

- Session 1: Introduction to Web Architecture Coverage Depth: Recap: client-server model & HTTP communication. Static vs Dynamic web pages. Understanding how servers handle requests. Demo: Flow diagram – React frontend ? Java Servlet ? Database. Goal: Students visualize data flow from UI to DB.
- Session 2: Introduction to Servlets Coverage Depth: Servlet lifecycle (init(), service(), destroy()). Role of web.xml deployment descriptor. Demo: Create “Hello Servlet” and run on Tomcat.
- Session 3: HTTPServlet & Request Handling Coverage Depth: doGet() and doPost() methods. Handling parameters using request.getParameter(). Demo: Build Login form ? Send data to Servlet ? Print welcome message.
- Session 4: ServletConfig & ServletContext Coverage Depth: Configuration parameters in web.xml. Application-wide vs Servlet-specific configs. Demo: Store database connection string in context parameter.
- Session 5: Response Handling & Redirection Coverage Depth: sendRedirect(), RequestDispatcher.forward(). HTTP status codes (200, 302, 404). Demo: Redirect to “DashboardServlet” post login.
- Session 6: Session Tracking Coverage Depth: Cookies, URL rewriting, HttpSession API. Login persistence using sessions. Demo: Keep user logged in until session invalidates.
- Session 7: JSP Basics Coverage Depth: JSP lifecycle, syntax, directives (<%@ %>), scriptlets. Expression Language (EL) introduction. Demo: Dynamic “Welcome {username}” JSP page. Depth: Compare JSP to React components (dynamic rendering).
- Session 8: JSP Actions & Tags Coverage Depth: Standard Action Tags: <jsp:include>, <jsp:useBean>, <jsp:setProperty>. Custom tag libraries intro. Demo: Include a header and footer dynamically.
- Session 9: MVC Pattern Using Servlet & JSP Coverage Depth: Model-View-Controller flow: Servlet = Controller, JSP = View, Java Bean = Model. Data flow between components. Demo: Build Student Registration ? Controller (Servlet) ? Model ? View (JSP).
- Session 10: Form Validation & Error Pages Coverage Depth: Server-side validation (null check, length, email format). JSP error pages & exception handling. Demo: Display “Invalid Credentials” on login failure.

- Session 11: JDBC Integration with Servlets Coverage Depth: Connect Servlet to MySQL using JDBC. Fetch user data & display via JSP table. Demo: Student list fetched from DB ? shown on JSP page.
- Session 12: Servlet & JSP Review Coverage Depth: Recap: Lifecycle, Sessions, MVC, JDBC Integration. Discuss transition to frameworks (Spring).
- Mini Project 9 – “Campus Connect Portal (Servlet–JSP Version)” Description: Create a basic web version of the Campus Connect Portal, allowing: Student registration & login Event posting (title, date, description) Displaying events list dynamically using JSP Trainer Focus: Show end-to-end flow: Form ? Servlet ? Database ? JSP. Reinforce MVC pattern.

Module 9) Java - Hibernate Framework	**5**

- Session 1 – Introduction to ORM & Hibernate Basics Coverage: What is ORM? Why Hibernate over JDBC? Hibernate architecture: SessionFactory, Session, Transaction. Configuration via hibernate.cfg.xml and annotations. Demo: Map Student entity ? students table, save record via save(). Depth: Draw parallels between Hibernate entities ? database tables.
- Session 2 – Entity Mapping & Annotations Coverage: @Entity, @Table, @Id, @GeneratedValue, @Column. Transient vs persistent states. Demo: Create Event entity for Campus Connect Portal.
- Session 3 – Relationships (One-to-One, One-to-Many, Many-to-Many) Coverage: Association mappings using @OneToMany, @ManyToOne, @JoinColumn. Cascade & Fetch types. Demo: User ? Event relationship (one user hosts many events).
- Session 4 – CRUD Operations & Queries Coverage: Hibernate Query Language (HQL). Create, Read, Update, Delete examples. Pagination and Criteria API basics. Demo: Display all events hosted by a specific user.
- Session 5 – Integration & Caching Coverage: Integrate Hibernate with Spring Boot. Enable 1st & 2nd Level Cache (Ehcache / Hazelcast). Performance tuning tips. Demo: Compare query performance with/without cache.
- Mini Project 11 – “Event Manager with Hibernate” Objective: Persist Campus Connect Portal data using Hibernate instead of JDBC. Features: CRUD for Users & Events + One-to-Many relations. Trainer Focus: Annotating entities correctly; configuring persistence.

Module 10) Java - Software Design Pattern And Project	**10**

- Session 1 – Understanding Software Design Patterns Coverage Depth: Why patterns are needed (maintainability, scalability). Overview: Creational, Structural, Behavioral patterns. Focused Patterns for Java Web Dev: Singleton (Database Connection Pool) Factory (Object creation abstraction) DAO (Data Access Object) pattern MVC (Model-View-Controller) pattern Demo: Show how a simple “UserDAO” separates logic from controller.
- Session 2 – Implementing MVC + DAO Design Pattern Coverage Depth: MVC Flow in Servlets: Controller (Servlet), View (JSP), Model (POJO/Bean). DAO layer for database operations. Advantages of abstraction & reusability. Demo: Implement “Login Flow” using MVC + DAO architecture. Depth: Explain how this pattern forms the base of Spring MVC framework.

- Session 3 – Session Management in Java Web Applications Coverage Depth: Why session tracking is required. Techniques: Session (HttpSession API) Cookies Hidden Form Fields URL Rewriting Session security best practices. Demo: Create “Login + Welcome Page” maintaining user session until logout. Depth: Compare session.invalidate() with JWT-based token expiry (for awareness).
- Session 4 – Template Integration (Front-End + Back-End Sync) Coverage Depth: Integrating Bootstrap/HTML template with Servlet or Spring Boot. Separating static resources (CSS/JS/images). Dynamic content rendering (JSP + JSTL + EL). Demo: Integrate a free Admin Template (e.g., AdminLTE or BootstrapDash). Depth: Teach folder structure for professional-grade web apps.
- Session 5 – File & Image Upload / Download Coverage Depth: Handling file uploads with MultipartConfig (Servlet 3.0). Saving file path vs saving file blob in DB. Download servlet setup. Demo: Profile image upload for student profile in Campus Connect. Depth: Show folder vs database storage trade-offs. AI Tool: Copilot to auto-generate upload/download code snippets.
- Session 6 – Mail Integration (JavaMail API) Coverage Depth: SMTP concept and configuration. Sending HTML emails using JavaMail. Attaching documents. Demo: Send “Registration Confirmation” email post signup. Depth: Explain difference between transactional & marketing emails.
- Session 7 – OTP Verification via Mail Integration Coverage Depth: Generate random OTPs using Java. Store OTP temporarily in session or DB. Validate OTP input by user. Demo: Email-based OTP login flow (2-step authentication). Depth: Discuss OTP expiry timers and best practices. AI Tool: ChatGPT or LangChain4J to auto-generate OTP logic function.
- Session 8 – Online Payment Integration Coverage Depth: Understanding payment gateway APIs (Razorpay / Stripe). REST API calls for order creation & transaction verification. Front-end integration using JavaScript SDK. Demo: Integrate Razorpay “Pay Now” button with Spring Boot backend. Depth: Emphasize security, API key management, webhook verification. AI Integration: ChatGPT for testing API response payloads.
- Session 9 – AJAX & API Integration Coverage Depth: What is AJAX, XMLHttpRequest, and Fetch API. Asynchronous form submissions (without reload). Real-time validation (email availability check). Demo: Check if email exists before registration (AJAX + Servlet). Depth: Bridge concept to React Axios for next project.
- Session 10 – Introduction to CI/CD & Deployment Workflow Coverage Depth: CI/CD Basics (Build ? Test ? Deploy cycle). GitHub Actions or Jenkins pipeline basics. Deploy Java app on Render/Heroku/AWS. Integrating automated testing (JUnit) in pipeline. Demo: Create CI pipeline to auto-deploy Spring Boot app to Render.

Module 11) Java - Spring	**7**

- Session 1: Introduction to Spring & its Architecture Coverage Depth: Evolution from Servlets to Spring. Core modules: Spring Core, Context, JDBC, AOP. Demo: Explain flow with simple diagram.
- Session 2: Inversion of Control (IoC) & Dependency Injection (DI) Coverage Depth: What is IoC & why DI improves modularity. Setter vs Constructor injection. Demo: Create StudentService and inject dependency via XML.
- Session 3: Spring Bean Configuration Coverage Depth: Defining beans in applicationContext.xml. Bean scopes (singleton, prototype). Autowiring basics. Demo: Configure and access multiple beans using IoC container.

- Session 4: Annotation-Based Configuration Coverage Depth: @Component, @Autowired, @Configuration, @Bean. Replace XML with annotation config. Demo: Annotate service class and autowire dependencies.
- Session 5: Spring JDBC Integration Coverage Depth: Using JdbcTemplate for CRUD operations. Configuring datasource and DAO layer. Demo: Perform CRUD on Student table using Spring JDBC.
- Session 6: Spring MVC Introduction Coverage Depth: DispatcherServlet, Controllers, Views, ModelMap. Handling GET and POST requests. Demo: Simple “Add Event” form handled via Spring Controller.
- Session 7: Mini Project – Spring MVC Module Integration Description: Convert the “Event Management” feature of Campus Connect into a Spring MVC module. Use annotations (@Controller, @RequestMapping). View pages via JSP.

Module 12) Java - Spring Boot	**5**

- Session 1: Introduction to Spring Boot Coverage Depth: What is Spring Boot, benefits over Spring Core. Auto-configuration and embedded server (Tomcat). Demo: Create Spring Boot app using Spring Initializr.
- Session 2: REST API Development Coverage Depth: REST principles (GET, POST, PUT, DELETE). @RestController, @RequestMapping, @PathVariable, @RequestBody. Demo: Create /events API for Campus Connect Portal.
- Session 3: Connecting Spring Boot with Database Coverage Depth: Spring Data JPA, CRUDRepository. Entity, Repository, Service layers. Demo: Event entity saved into MySQL via Spring Data JPA.
- Session 4: Security & Validation Coverage Depth: Introduction to Spring Security. Basic Auth + Validation annotations (@Valid, @NotNull). Demo: Protect “Add Event” API using Basic Auth.
- Session 5: Integration, Testing & Postman Demo Coverage Depth: Test API using Postman. Enable CORS to connect with React frontend. Demo: React app fetches event list from /events API.
- Mini Project 10 – “Campus Connect Portal (Spring Boot Edition)” Description: Upgrade the existing web app into a REST-based, scalable version: Frontend: React (Bio Link builder layout repurposed). Backend: Spring Boot REST API. Features: User registration & login (secure) Event management CRUD API integration with Axios (React). Trainer Focus: Emphasize microservice-ready design. Teach deployment readiness (Heroku or Render). AI Integration: Postman AI for automated API test cases.

Module 13) Java - Spring WebServices	**6**

- Session 1 – Introduction to AI in Java Coverage: What AI APIs can Java use? Overview of OpenAI API, Google Vertex AI, Hugging Face Inference API. Demo: Show REST call from Java to OpenAI “text-davinci-003.”
- Session 2 – Working with REST APIs in Java Coverage: HttpClient usage for API calls. Parsing JSON responses (Jackson ObjectMapper). Demo: Fetch AI quote of the day from public API.
- Session 3 – Integrating OpenAI API with Spring Boot Coverage: Generate API key + environment setup. Build REST endpoint that calls OpenAI API. Demo: “Ask CampusBot” endpoint ? OpenAI response.

- Session 4 – Building Chatbot with LangChain4J Coverage: What is LangChain4J (LLM wrapper for Java). Creating prompt templates and chains. Demo: CampusBot answers FAQ using LangChain4J + OpenAI.

Module 14) Java - Microservices with Spring Boot, Spring Cloud	6

- Session 1 – Microservices Concepts & Architecture Coverage: Monolith vs Microservices. Advantages: scalability, independent deployment. Demo: Diagram of “Campus Connect Microservices Architecture.”
- Session 2 – Creating Microservices with Spring Boot Coverage: Building independent services (UserService, EventService). REST API communication between them. Demo: Create UserService with CRUD APIs.
- Session 3 – Service Discovery using Eureka Coverage: Eureka Server setup, registration of clients. Fetch service instances programmatically. Demo: Register User & Event services with Eureka.
- Session 4 – API Gateway & Load Balancing Coverage: Spring Cloud Gateway basics, routing requests. Load balancing with Ribbon / Eureka. Demo: Route requests through gateway to EventService.
- Session 5 – Inter-Service Communication (Feign Client) Coverage: Calling other services via Feign. Error handling and fallbacks. Demo: UserService calls EventService to fetch event details.
- Session 6 – Monitoring & Logging (Actuator + Prometheus) Coverage: Add Spring Boot Actuator for health checks. Integrate Prometheus and Grafana for monitoring.
- Mini Project 12 – “Campus Connect Microservices Edition” Objective: Split the Campus Connect Portal into User and Event services communicating via REST and Eureka. Trainer Focus: Demonstrate API gateway routing and service monitoring.