ANNEXURE: 1 MODULE WISE COURSE CONTENT AND OUTCOME

SL. NO	MODULE NAME	MODULE CONTENT	MODULE LEARNING OUTCOME	DURATION (HRS)
1	Introduction to Java Full-Stack Development	- Overview of Full-Stack Development. - Introduction to Java (Versions 8 to 17). - Setting up the development environment (IDE, JDK). - Introduction to Java Platform for web development. - Key tools and technologies in Full-Stack Development.	Understand full-stack development concepts and set up the development environment with Java technologies.	2
2	Core Java Basics	- Understanding variables, data types, and operators. - Conditional Statements (if, if-else, switch). - Loops (for, while, do-while). - Arrays and String Manipulation. - Introduction to Object-Oriented Programming (Classes, Objects, Inheritance, Polymorphism,	Learn the basics of Java programming, control flow, and fundamental OOP principles.	4

		Encapsulation).		
3	HTML, CSS, and JavaScript for Front-End Development	- Introduction to HTML: Structure of web pages, elements, and attributes. - Basic CSS: Styling, colors, fonts, margins, paddings. - JavaScript Basics: Variables, Functions, DOM Manipulation. - Introduction to Responsive Web Design with CSS (Media Queries).	Build basic, responsive web pages using HTML, CSS, and JavaScript with simple interactivity.	4
4	Spring Boot Framework Basics	- Introduction to Spring Boot and its features. - Setting up a Spring Boot application. - Spring Boot Architecture: Dependency Injection, Controllers, Services. - Building a simple REST API with Spring Boot.	Set up a basic Spring Boot application and understand its core components.	4
5	Spring MVC Architecture and Thymeleaf	- Overview of MVC (Model-View-Controller) architecture. - Introduction to Thymeleaf templating engine.	Learn to build dynamic web applications with Spring Boot and Thymeleaf for server-side	4

		- Setting up Thymeleaf with Spring Boot. - Rendering dynamic content using Thymeleaf. - Handling forms and form validation with Thymeleaf. - Spring MVC integration with Thymeleaf.	rendering.	
6	Spring Data JPA and Database Integration	- Introduction to databases and Spring Data JPA. - Configuring a relational database (e.g., MySQL, PostgreSQL). - Performing CRUD operations with Spring Data JPA. - Introduction to Entity, Repository, and Spring Data features. - Introduction to Flyway for database migrations.	Integrate databases into Spring Boot applications, performing CRUD operations and managing schema with Flyway.	3
7	Building RESTful APIs with Spring Boot	- Introduction to RESTful Web Services. - Creating REST APIs with Spring Boot (GET, POST, PUT, DELETE). - Handling JSON	Learn how to build and test RESTful APIs using Spring Boot, with proper request handling and response	3

		data in REST API requests/responses. - Exception handling in REST APIs. - Testing REST APIs using Postman.	formatting.	
8	Spring Security Fundamentals	- Introduction to Spring Security and authentication mechanisms. - Configuring Spring Security for basic authentication. - Securing REST APIs using Spring Security. - Introduction to authorization and access control. - Role-based access control in applications.	Understand how to secure Java applications using Spring Security for authentication and authorization.	4
9	Spring JWT Authentication	- Introduction to JSON Web Tokens (JWT). - Setting up JWT for Spring Boot. - Generating and validating JWT tokens. - Configuring Spring Security to use JWT for authentication. - Best practices	Implement JWT authentication in Spring Boot applications for secure API access.	4

		for JWT-based security.		
10	Thymeleaf Advanced Features	- Thymeleaf Expressions and Variable Substitution. - Thymeleaf Iteration (for-each loop). - Using Thymeleaf fragments for reusable templates. - Conditional rendering in Thymeleaf. - Handling form input and validation with Thymeleaf.	Master advanced Thymeleaf features for building dynamic and reusable web pages.	3
11	Connecting Front-End with Back-End	- Integrating JavaScript with Spring Boot. - Using Fetch API to make AJAX requests to Spring Boot back-end. - Binding data between Thymeleaf front-end and Spring Boot back-end. - Managing session data in web applications.	Learn to connect the front-end (HTML, CSS, JavaScript) with the back-end (Spring Boot), handling dynamic data updates.	2
12	Spring Boot Deployment and Performance	- Preparing a Spring Boot application for deployment.	Learn how to package, deploy, and optimize	2

		- Packaging a Spring Boot app as a JAR. - Deploying applications on a local server.	Spring Boot applications for production use.	
13	Testing and Debugging in Spring Boot	- Unit testing Spring Boot applications using JUnit. - Writing integration tests for Spring Boot services. - Mocking data with Mockito. - Debugging Spring Boot applications.	Learn to write and execute unit and integration tests for Spring Boot applications to ensure quality and reliability.	2
14	Building a Real-World Project (Hands-on)	- Hands-on project: Building a simple web application using HTML, CSS, JavaScript, Thymeleaf, Spring Boot, and Spring Security. - Features include: User registration, login, CRUD operations	Apply the concepts learned to build a fully functioning, secure web application.	2
15	Final Review and Best Practices	- Recap of key concepts in Java Full-Stack Development. - Code quality practices (clean code, optimization).	Revise key concepts, implement best practices, and prepare for real-world Java Full-Stack	2

		- Best practices in Spring Boot, Thymeleaf, and Spring Security. - Introduction to career paths and opportunities in Java Full-Stack Development.	roles.	

ANNEXURE : 2 Industry Use Cases/Final Projects

INDUSTRY USECASES		
Learning Outcome	**Assessment Criteria**	**Use Cases**
Explore the basics of Java FullStack Development with Spring.	Successfully set up the development environment (IDE, JDK, Spring Boot) and project structure using Spring Initializr.	1. Setting up a Java FullStack project for a student portal using Spring Boot.
Apply Java programming concepts and OOP principles with Spring.	Ability to demonstrate the use of classes, objects, inheritance, and interfaces in Spring Boot projects.	2. Implementing a service class to manage user data using Spring's Dependency Injection.
Use Spring Data JPA for database interaction.	Implement JPA repositories and perform CRUD operations on a relational database.	3. Implementing a Spring Data JPA repository for managing student records.
Create RESTful APIs with Spring Boot.	Build REST APIs with proper HTTP methods (GET, POST, PUT, DELETE) and manage routing with @RequestMapping or @GetMapping.	4. Creating a REST API for a student management system using Spring Boot.
Implement Spring Security for authentication and authorization.	Set up user authentication, configure role-based access, and protect routes using Spring Security annotations.	5. Implementing login functionality with Spring Security, including JWT-based authentication.
Use JWT (JSON Web Tokens) for stateless authentication.	Implement JWT generation and validation in Spring Boot to secure APIs.	6. Creating a login API that returns a JWT token for authenticated users.
Design front-end web pages using HTML, CSS, and JavaScript.	Structuring HTML elements, applying CSS for styling, and using JavaScript for dynamic behavior.	7. Creating a responsive login page and styling it with CSS for a modern web application.

Apply AJAX for asynchronous communication between front-end and back-end.	Use JavaScript with AJAX to interact with the backend without page reloads.	8. Creating a dynamic search functionality with AJAX that fetches student data without refreshing the page.
Integrate front-end with Spring back-end.	Develop a full-stack web application where the front-end makes requests to Spring Boot APIs and handles responses.	9. Building an interactive dashboard that displays user data dynamically using JavaScript and Spring Boot.
Implement database interaction and CRUD operations with Spring Data JPA.	Execute basic CRUD operations such as saving, updating, deleting, and querying data using Spring Data JPA.	10. Developing a user registration system that stores data in a MySQL database using Spring Data JPA.
Develop services and controllers in Spring Boot.	Create and manage service classes for business logic and controller classes for routing HTTP requests.	11. Implementing a REST API that manages customer records using service and controller classes.
Understand and implement Spring Boot configuration.	Use application properties to configure Spring Boot services and data sources.	12. Configuring data sources, profiles, and security settings for a Spring Boot application.
Test Java FullStack applications with Spring Boot.	Implement unit tests for controllers, services, and repositories using Spring Boot testing features and JUnit.	13. Writing unit tests for a student service to check CRUD operations using Spring Boot Test and JUnit.
Debug Java FullStack applications.	Identify and resolve errors in both front-end JavaScript code and back-end Spring Boot code using debugging tools.	14. Debugging a failed user authentication scenario and fixing errors in Spring Security configuration.
Deploy a Spring Boot application on a server.	Deploy the Spring Boot application to a local or cloud server (e.g., Apache Tomcat, Spring Boot embedded server).	15. Deploying a Spring Boot application on Apache Tomcat and ensuring the app is accessible from the browser.
Implement security	Implement authentication	16. Implementing role-based

features in web applications using Spring Security.	(login, logout) and authorization (role-based access control) in Spring Boot.	access control in a Spring Boot application to restrict admin functionality.
Handle form data and sessions in Spring Boot web apps.	Use Spring MVC to handle form submissions and store session data.	17. Creating a session management system that tracks user preferences across multiple pages in an e-commerce application.
Create a complete FullStack project with front-end and back-end integration.	Build a fully functional Java FullStack web application that integrates all front-end and back-end components.	18. Developing an online ticket booking system using Spring Boot, React, and MySQL for full-stack integration.

LIST OF FINAL PROJECTS

1. **Library Management System**
   Develop a system for managing book catalog, member registration, and book borrowing with features like due date tracking and overdue notifications.
2. **Online Bookstore**
   Build an online bookstore where users can browse, search, and purchase books, with admin functionality to manage inventory, pricing, and orders.
3. **Online Food Delivery System**
   Create an online food ordering and delivery system that includes restaurant menus, order tracking, payment gateways, and delivery updates.
4. **Online Learning Platform**
   Build a learning management system (LMS) where users can register for courses, view lessons, take quizzes, and track progress.
5. **Movie Review Website**
   Develop a platform where users can rate and review movies, see movie details, and explore reviews from other users.
6. **Personal Finance Management System**
   Create a financial planning app where users can track income, expenses, and generate reports for better financial management.

7. **Fitness Tracker App**
   Develop a system where users can log workouts, track goals, and see progress reports based on their fitness activities.
8. **Event Ticket Booking System**
   Build a ticket booking system for events like concerts, sports, or theater, with seat selection, pricing, and event details.
9. **Online Voting System**
   Create a secure and transparent online voting platform with user authentication, voting, and result display for different election types.
10. **Online Recipe Sharing Platform**
    Build a platform where users can share, search, and rate recipes, with categories like vegetarian, vegan, and non-vegetarian.
11. **Pet Adoption System**
    Develop a pet adoption system where users can browse and adopt pets, manage profiles, and receive updates about pet health.
12. **Car Rental System**
    Create a car rental platform where users can browse available vehicles, make bookings, and manage rental periods and payments.
13. **Task Management Application**
    Build a task management system where users can create, track, and complete personal or work-related tasks with deadlines and notifications.
14. **Hotel Reservation System**
    Create a hotel booking system where users can search for available rooms, make reservations, and manage bookings.
15. **Social Media Platform**
    Develop a social networking site where users can create profiles, post content, follow other users, and interact through likes, comments, and shares.
16. **Online Auction System**
    Build an auction platform where users can bid on products in real-time, view current bid amounts, and track auction outcomes.
17. **Inventory Management System**
    Create a system that tracks product stock levels, manages orders, and provides reporting for businesses.
18. **Online Classified Ads System**
    Develop a platform for posting and browsing classified ads in categories such as jobs, real estate, and services.
19. **Donation Management System**
    Build a system for managing donations to charities, including donation tracking, receipts, and donor management.
20. **Job Application Tracker**
    Create a web-based job application tracker where users can store job applications, track the hiring process, and follow up with potential employers**.**

ANNEXURE 3 – COURSE ASSESSMENT

COURSE ASSESSMENT RUBRICS (TOTAL MARKS: 70)				
ASSESSMENT CRITERIA	**DESCRIBE THE CRITERIA OF THE BELOW CATEGORY PERFORMANCE**			**TOTAL MARKS**
	FAIR	**GOOD**	**EXCELLENT**	
MCQ/ Programming /Project Submission Round	Above 40	Above 55	Above 65	70

Category	**Assessment Criteria**	**Performance Levels**	**Weightage (Marks)**
Practical Skills Proficiency	Demonstrates ability to perform job-specific tasks effectively, using relevant tools, techniques, or methodologies (e.g., Tally for accounting, consignment tracking).	Fair, Good, Excellent	20
Technical Knowledge Application	Applies theoretical concepts to practical scenarios with accuracy and relevance (e.g., compliance with GST laws, financial planning, or logistics protocols).	Fair, Good, Excellent	15
Project Execution	Completes assigned projects or use cases demonstrating innovation, thoroughness, and skill application relevant to industry standards.	Fair, Good, Excellent	25
Communication and Reporting	Clearly presents findings, solutions, or project outcomes using professional communication and documentation standards (e.g., reports, presentations).	Fair, Good, Excellent	10