TECHGENX

Advance Java+ Spring Boot API + MySQL + AWS + Docker + React + AngularJS

Please Note: All below course content will be covered in practical scenarios and regular assignments will be shared. All sessions will be recorded and shared with student for future reference (free of cost). Along with below course.

Java Project Development Concepts

1. **Core Java Topics (Core Java Link)**
2. **Install Spring Tool Suite**
3. **Configuring JDK in STS**
4. **Install MySQL and MySQL Workbench**
5. **Install Postman**
6. **Maven Projects and troubleshooting around Maven project**
7. **What are micro services**
8. **Why Micros services**
9. **Four Layers and Classes**
10. **Why Layers**
11. **Layers and Technologies**
12. **Six Key Classes**

Data Access Layer with Live Project

1. **Introduction**
2. **Create DB schema**
3. **Create Spring Boot project**
4. **Spring 2.X Updates**
5. **Create Model Class**
6. **Create Repository**
7. **Use @GeneratedValue**
8. **Configure the Data Source**
9. **Create Student Class**
10. **Read Student**
11. **Using Spring Boot 2.X**
12. **Update Student**
13. **Delete Student**
14. **Migration to Spring Boot 2.X**
15. **Assignment**

Presentation Layer with Live Project

1. **Introduction**
2. **Create Workflow**
3. **Display Locations Workflow**
4. **Delete Workflow**
5. **Update Workflow**
6. **Create Location Table**

7. **Create locationweb Project**
8. **Create the Model Class**
9. **Create the Repository**
10. **Configure the Data Source**
11. **Create the services layer**
12. **Implement the methods**
13. **Create the view for save location**
14. **Create the controller**
15. **Configure prefix and suffix**
16. **Add the Jasper dependency**
17. **Update the Application Context**
18. **Handle the create request**
19. **Send Response back**
20. **View, Links and Controller method**
21. **Create the JSP**
22. **Use JSTL**
23. **Add create Location Link**
24. **Delete Request Flow**
25. **Delete Response Flow**
26. **Add Edit button and show update controller**
27. **Edit Location jsp**
28. **Handle Update**

Utility Classes with Live Project

1. **Email use case**
2. **Add Maven Dependency**
3. **Create Utility Classes**
4. **Configure mail properties**
5. **Setup a email account**
6. **Use the Util classes**

Reports Utility with Live Project

1. **Low level workflow**
2. **Add Maven dependency**
3. **Add repository method**
4. **Create utility class**
5. **Generate the report**
6. **Create the controller method**
7. **Create the view**

Integration Layer with Live Project

1. **What is REST?**
2. **Create the REST controller**
3. **Get all locations**
4. **Create Location**
5. **Update Location**

6. **Delete Location**
7. **Get one location**

Micro Services with Live Project

1. **Introduction to Micro Services**
2. **Create DB Schema**
3. **Create a Project**
4. **Create Model Classes**
5. **JPA annotations**
6. **Define relationship**
7. **Create data access layer**
8. **Create User Registration view and controller**

User Registration use case (Live Project)

1. **Requirements and design**
2. **Update spring boot and maven configuration**
3. **Access User Registration**
4. **Handle user registration request**
5. **Test User Registration**
6. **Handle Login**

Search Flights (Live Project)

1. **Requirements and Design**
2. **Find flight jsp and controller**
3. **Repository method**
4. **Display flights**
5. **Load Flight data**
6. **MySQL-JDBC URL**
7. **Handle Flight selection**
8. **Complete reservation view**

Create Reservation (Live Project)

1. **Requirements and Design**
2. **Request and Controller method**
3. **Services layer**
4. **Service layer classes**
5. **Reservation confirmation view**
6. **Home Page**

Create Integration Layer (Live Project)

1. **Find Reservation REST service**
2. **Update reservation REST service**
3. **Add Cross origin support**

Create Check-In App (Live Project)

1. **Create project**
2. **Model and REST client classes**
3. **Find Reservation REST client**
4. **Update Reservation REST client**
5. **Check-In UI**
6. **Handle Start Check-in**
7. **Display reservation details**
8. **Implement the Check-in Method**
9. **Check in Confirmation**

Generate and Email Itinerary (Live Project)

1. **Add IText dependencies**
2. **Create the PDFGenerator**
3. **Add Flight Details**
4. **Add Passenger Details**
5. **Add Email Support**
6. **Create EmailUtil**
7. **Use the Utility Classes**
8. **Test Book Flight**
9. **Test Check**

Logging (Live Project)

1. **Configure and use loggers**
2. **Configure Root Log Level**
3. **Logging in controller layer**
4. **Logging in Services Layer**
5. **Util and Integration Layer**
6. **Test Logging**
7. **Log to file**
8. **Logging Components**
9. **Create logback.xml**
10. **Define the appender log pattern**
11. **Configure Rolling Policy**
12. **Define Root Logger**
13. **Test Logback**

Externalized Configuration (Live Project)

Externalize Flight Reservation Properties

Security (Live Project)

1. **Security Introduction**
2. **Encode Password**
3. **Create ADD Flight View**
4. **Create DB Tables**
5. **Create Role Entity**

6. **Define relationships**
7. **Create Role Repository**
8. **Implement UserDetailsService**
9. **Create a SecurityService**
10. **Set token in to SecurityContext**
11. **Update the UserController**
12. **Create Security Configuration**
13. **Create Password Encoder Bean**
14. **Configure the AuthenticationManager Bean**
15. **Skip REST URIs**

Transaction Management (Live Projects)

1. **Introduction**
2. **Which layer should a transaction start?**

Deployment (Live Project)

1. **JAR Deployment**
2. **WAR Deployment**

Create CheckIn Application using AngularJS (Live Project)

1. **Introduction**
2. **Angular in brief**
3. **Installation**
4. **Create the project**
5. **Project Structure**
6. **Create Components and Services**
7. **Implement the getReservation Service**
8. **Implement the checkIn Service**
9. **Start implementing the components**
10. **Create the Routing Module**
11. **Update the App Module Config**
12. **Use Routing in HTML**
13. **Bind Reservation ID and Navigate**
14. **Fetch Reservation**
15. **Display Flight and Passenger Information**
16. **Implement CheckIn Method**
17. **Confirm CheckIn**
18. **Apply Styles**

Project Usecase (Live Project)

Usecase and Requirements

Create the REST Backend (Live Project)

1. **Download the completed projects**
2. **Setup the database**
3. **Executable Java Project**
4. **Create the project**
5. **Create the Model**
6. **Define Entity Relationships**
7. **Create the Repositories**
8. **Create the Patient REST Controller**
9. **Implement the GET and Save Patient methods**
10. **Test GET methods**
11. **Test save method**
12. **Implement save clinical data api**
13. **Test save clinical data**
14. **Implement Analyze Method**
15. **Calculate BMI**
16. **Filter the results**
17. **Test Analyze Method**
18. **Implement GET Clinical Data Method**
19. **Reuse BMI Logic**
20. **Test GET Clinical Data**
21. **Add Cross Site Origin Support**

Frontend Development using REACT (Live Project)

1. **What is React?**
2. **Install Node**
3. **Install Yarn**
4. **Install React CLI**
5. **Install Visual Studio**
6. **Create the project**
7. **Create Components**
8. **Configure Routing**
9. **Start implementing the Home Page**
10. **Implement RowCreator**
11. **Implement Add patient**
12. **Add Patient - Handle Submit**
13. **Add Patient – Toastify**
14. **Test Add Patient**
15. **Implement the Add Clinical Data Component**
16. **Create the HTML Form**
17. **Handle Submit**
18. **Implement Analyze Component**
19. **Create the TableCreator Component**

Deploy to AWS (Live Project)

1. **AWS EC2 and S3 Quick Start**
2. **Usecase**
3. **Launch a EC2 Instance**
4. **Mac Only - Connecting to EC2**
5. **Windows - Use MobaXterm and connect**
6. **Install java and mysql**
7. **Upload jar to S3 Bucket**
8. **Get , Run APP and Test**

Containerize Spring Boot App using Docker (Live Project)

1. **Introduction**
2. **Install Docker**
3. **Usecase**
4. **Launch a MySql Docker Container**
5. **Dockerize the Flight Services APP**
6. **Docker in action**

Uploading and Downloading Documents (Live Project)

1. **Create the DB Table**
2. **Create the Project**
3. **Create the Model and Repository**
4. **Create the Upload JSP**
5. **Create the Controller**
6. **Implement the upload method**
7. **Save the document**
8. **Test Upload**
9. **Download requirement**
10. **Update the upload method**
11. **Update the View**
12. **Test and run display documents**
13. **Implement the download method**
14. **Stream the file back**
15. **Save the file name**