

Dashboard Designing with Angular and Express JS

Duration: 7 days

Description: -

This course involves building a full-stack web application leveraging Angular for the front-end, Express.js for the back-end server, and SQL Server as the database. The Angular framework will handle the user interface and client-side logic, while Express.js will serve as the backend, managing API endpoints and routing. SQL Server will be used to store and retrieve data, ensuring efficient database management and integration with the backend through an ORM or direct SQL queries.

Angular 17

Table of contents:

Module 1: Introduction to Angular and TypeScript

- Introduction to Angular
- Introduction to TypeScript
- Setting up the Development Environment and Running the Angular Initial Application
- Angular Directory Structure
- Angular Building Blocks

Module 2: Components and Data Binding

- Working with standalone Components
 - @Component Decorator and its Metadata
 - Generating a Component using the Angular CLI
 - Invoking Components
- Data Binding
 - Interpolation
 - Property Binding
 - Event Binding
 - Two-way Binding
- Control-flow grammar
 - Conditionals:
 - @if
 - @else
 - Loops:
 - @for
 - Switch Statements:
 - @switch
 - @case
 - @default

Module 3: Dependency Injection and HTTP Communication

- Angular Forms
 - Template Driven Form and its validation

- Reactive Form and its validation
- Interface
- Dependency Injection
- Working with Services

- Calling Backend APIs
 - Working with JSON Data
 - CRUD operations from a Rest API using HttpClient

Module 4: Component Communication and Routing

- Angular Component Communication with @Input/@output decorator

- Introduction to Routing
 - Implementing Routing in an Angular App
 - Wildcard Route / Default Route
 - Navigate Programmatically
 - Implementing Child Routes
 - Router Inputs

- Working with Route Guards in Angular
- Lazy Loading components

Module 5: Pipes & Directive

- Attribute Directive
 - NgStyle, NgClass
- Pipes
 - Introduction to Pipes in Angular
 - Using Built-in Pipes
 - Creating Custom Pipes in Angular

Module 6: Overview of Data Visualization

- **Importance of Data Visualization**
 - Why data visualization is important in modern web applications
 - Common chart types and their use cases (bar, line, pie, etc.)

- **Popular Charting Libraries for Angular**
 - Introduction to libraries like Chart.js, Ng2-Charts, and Highcharts
 - When to use which library based on the project

- **Installing Chart.js and Ng2-Charts**

- Integrating Chart.js and Ng2-Charts in an Angular project
- Setting up the basic project structure
- **Bar Chart with Chart.js**
 - Setting up a basic bar chart component in Angular
 - Configuring datasets and chart options
- **Pie Chart with Chart.js**
 - Creating a pie chart with dynamic data
 - Understanding labels, colors, and customizing the pie chart
- **Line Chart with Chart.js**
 - Creating a simple line chart with Ng2-Charts
 - Configuring axis, grid lines, and labels

Express JS

Table of contents

- **What is Express.js?**
 - Introduction to Express.js and its key features
 - Benefits of using Express for building web applications
- **Installing Node.js and Express.js**
 - Setting up a Node.js environment
 - Installing Express.js using npm
 - Setting up a basic Express project
- **Express.js Fundamentals**
 - Creating a Basic Express Server
 - How to create an Express server
 - Understanding the basic structure of an Express app
 - Handling GET and POST requests
 - Routing in Express
 - Creating routes to handle different HTTP requests (GET, POST, PUT, DELETE)
 - Route parameters and query strings
 - Organizing routes using Router objects

- **Middleware in Express.js**
 - What is Middleware?
 - Understanding middleware functions and the request-response cycle
 - Built-in middleware in Express (e.g., `express.static`, `express.json`)
 - Creating Custom Middleware
 - Writing your own middleware functions
 - Using `next()` to pass control in the middleware chain
 - Error handling middleware
- **Database Integration with SQL Server**
 - Installing `mssql` or `sequelize` package for SQL Server integration
 - Configuring database connection in an Express app
 - **Performing CRUD Operations with SQL Server**
 - Writing routes that interact with the database
 - Querying the database to retrieve and send data
 - Implementing basic CRUD operations with SQL Server