Course Name: Java, Jquery, Node.js and API Development Training Program

Course Overview: Java, jQuery, Node.js, and API Development Training Program is designed to equip learners with the skills necessary to build full-stack web applications. This course covers front-end development with Java and jQuery, and back-end development using Node.js and API design. Learners will gain practical experience through hands-on projects and real-world case studies, preparing them for roles like Full-Stack Developer, API Developer, or Java Developer.

Course Duration:

16 weeks (4 months)

8-10 hours/week (Lectures, coding assignments, hands-on projects, and case studies)

Tech Learniversity, 170 1/1, Opposite HDFC Bank, Vijay Nagar,
J.N. Road, Mulund (West), Mumbai-400080, Maharashtra, India
E-mail ID- info@techlearniversity.com , Mobile No. +91-9082949171/+91-7021789240

Syllabus

Module 1: Introduction to Web Development (Week 1)

1.1 Overview of Web Development**

- The architecture of a web application.
- Client-side vs. server-side development.
- Full-stack development roles and responsibilities.

1.2 Introduction to Front-End and Back-End Technologies**

- Overview of HTML, CSS, JavaScript, and their roles in front-end development.
- Introduction to back-end development with Node.js, API design, and databases.

1.3 Tools for Web Development**

- Setting up the development environment (IDEs, Git, GitHub).
- Introduction to version control with Git.

Module 2: Java Programming for Web Development (Week 2-4)

2.1 Introduction to Java Programming

- Setting up a Java development environment (JDK, IntelliJ/Eclipse).
- Understanding Java syntax, variables, data types, and operators.
- Control structures: Loops, conditional statements.

2.2 Object-Oriented Programming (OOP) in Java

- Principles of OOP: Classes, objects, inheritance, encapsulation, and polymorphism.
- Working with constructors, methods, and interfaces.
- Exception handling in Java.

2.3 Java for Web Applications

- Introduction to Java servlets for web development.
- Handling HTTP requests and responses in Java.
- Building simple web applications with Java servlets and JSP (Java Server Pages).

2.4 Case Study: Developing a Java Web Application

- Building a basic web application using Java servlets.
- Handling form data and displaying dynamic content.

Module 3: jQuery for Interactive Web Pages (Week 5-6)

3.1 Introduction to jQuery

- Overview of jQuery and its role in front-end development.
- Adding jQuery to your web application.

3.2 DOM Manipulation with jQuery

- Understanding the Document Object Model (DOM).
- Selecting and manipulating HTML elements using jQuery.
- Changing content and attributes dynamically.

3.3 Event Handling in jQuery

- Handling user events: clicks, form submissions, keypresses, etc.
- Creating interactive forms and dynamic content with jQuery.

3.4 AJAX with jQuery

- Introduction to AJAX (Asynchronous JavaScript and XML).
- Making asynchronous HTTP requests using jQuery's `$.ajax()` method.
- Fetching data from a server without reloading the page.

3.5 Case Study: Building an Interactive Web Page

- Developing an interactive web page with jQuery and AJAX.
- Fetching data from APIs and displaying dynamic content.

Module 4: Introduction to Node.js (Week 7-8)

4.1 Introduction to Node.js

- Understanding Node.js and its event-driven, non-blocking architecture.
- Setting up a Node.js environment.

- Writing your first "Hello World" application in Node.js.

4.2 Working with Node.js Modules

- Understanding the module system in Node.js (require, exports).
- Using built-in Node.js modules like `fs` (file system), `http`, `path`, etc.

4.3 Asynchronous Programming in Node.js

- Handling asynchronous operations with callbacks, promises, and async/await.
- Event-driven programming in Node.js.

4.4 Building a Simple Web Server with Node.js

- Creating an HTTP server using the `http` module.
- Handling requests and serving HTML, CSS, and JavaScript files.

Module 5: Express.js Framework for Node.js (Week 9-10)

5.1 Introduction to Express.js

- What is Express.js and why use it for web development?
- Setting up an Express project.

5.2 Building Routes in Express.js

- Creating routes to handle different HTTP requests (GET, POST, PUT, DELETE).
- Handling URL parameters and query strings.

5.3 Middleware in Express.js

- Understanding the middleware pattern in Express.
- Using built-in middleware for serving static files, handling errors, and parsing request bodies.

5.4 Building RESTful APIs with Express.js

- Designing RESTful API endpoints with Express.js.
- Returning JSON data from API routes.
- Connecting APIs with front-end using AJAX/jQuery.

5.5 Case Study: Building a RESTful API

- Developing a complete API using Express.js.

- Implementing CRUD operations (Create, Read, Update, Delete) for a resource.

Module 6: Working with Databases (Week 11-12)

6.1 Introduction to Databases

- Understanding the role of databases in web applications.

- Overview of SQL (Structured Query Language) and NoSQL databases.

6.2 Using MongoDB with Node.js

- Introduction to MongoDB, a NoSQL database.

- Connecting a Node.js application to MongoDB using Mongoose.

6.3 Performing CRUD Operations with MongoDB

- Creating, reading, updating, and deleting records in MongoDB.

- Querying data with MongoDB.

6.4 Data Validation and Error Handling

- Validating data before saving it to the database.

- Handling errors in API routes and database operations.

6.5 Case Study: Full-Stack Application with MongoDB

- Building a complete web application with a MongoDB backend.

- Implementing database operations in a real-world project.

Module 7: APIs and Web Services (Week 13-14)

7.1 Introduction to APIs and Web Services

- What are APIs and how do they work?

- Understanding REST (Representational State Transfer) and SOAP (Simple Object Access Protocol).

7.2 Designing RESTful APIs

- Best practices for designing RESTful APIs.

- Implementing authentication and authorization (JWT, OAuth).

7.3 Consuming Third-Party APIs

- Fetching data from public APIs (e.g., OpenWeatherMap, GitHub API).
- Integrating third-party APIs into your web applications.

7.4 Testing APIs

- Testing API endpoints using Postman.
- Writing unit tests for API routes using Mocha and Chai.

7.5 Case Study: Building an API-Driven Web Application

- Developing a web application that consumes data from multiple APIs.
- Implementing error handling and API rate limiting.

Module 8: Final Project – Building a Full-Stack Web Application (Week 15-16)

8.1 Project Overview and Requirements

- Defining the scope and requirements for a full-stack project.
- Designing both front-end and back-end architecture.

8.2 Full-Stack Project: Front-End Development

- Developing the front-end of the application using HTML, CSS, JavaScript, and jQuery.
- Implementing dynamic content and form submissions with AJAX.

8.3 Full-Stack Project: Back-End Development

- Setting up the Node.js back-end with Express.js.
- Creating and consuming APIs to handle front-end requests.
- Implementing MongoDB for data storage.

8.4 Project Presentation and Feedback

- Presenting the final full-stack application.
- Receiving feedback and suggestions for improvement.

Tools & Platforms Covered:

- Front-End Development: HTML, CSS, JavaScript, jQuery.

- Back-End Development: Node.js, Express.js.

- APIs: RESTful API design, third-party API integration.

- Database Management: MongoDB, Mongoose.

- Version Control: Git, GitHub.

- Testing: Postman, Mocha, Chai.

Assessment & Certification:

- Weekly coding assignments, quizzes, and case studies.

- Final full-stack project evaluation.

- Certification of completion after passing all modules and the final project.

Outcome:

By the end of the course, learners will be able to:

- Build interactive web pages using Java, jQuery, and AJAX.

- Develop server-side applications and APIs with Node.js and Express.js.

- Work with databases to store and retrieve data in MongoDB.

- Design and consume RESTful APIs for web applications.

- Be job-ready for roles such as Full-Stack Developer, Java Developer, or API Developer.

This course provides a comprehensive understanding of front-end and back-end web development, focusing on modern tools like Node.js, Express.js, and MongoDB, with practical projects to help learners gain industry-relevant experience.