CONCEPT

Full-Stack Web Development: React, Node.js, Express and SQL Databases

Full-Stack Web Development: React, Node.js, Express and SQL Databases

Introduction:

Many development teams split web work between separate front-end and back-end specialists, so features stall at hand-offs, APIs drift from the screens that call them and releases depend on manual steps. Full-stack web development closes that gap by giving one developer the skills to build the interface, the API, the database layer and the release pipeline together. This Core Concept course trains developers to build React interfaces, Node.js and Express APIs and relational data models, add authentication and automated tests, and ship in containers through a CI pipeline. Participants deploy a capstone full-stack application.

Course Objectives:

- Structure a web application into front-end, API and relational database layers and set up a Git repository with a branching and pull request workflow
- Build React interfaces from components, props, state and Hooks, with client-side routing, forms and data loaded through the Fetch API
- Design REST endpoints and implement them in Node.js with Express routing, middleware and centralised error handling
- Model relational data and query it through an ORM with schema migrations, joins, pagination and transactions
- Implement session or JSON Web Token authentication and cover the application with unit, API and end-to-end tests
- Package the application in containers and release it to a cloud platform through a CI pipeline as a working capstone application

Target Audience:

- Front-end developers who build browser interfaces and need to write the APIs and data layers behind them
- Back-end developers who maintain server code and databases and need to build modern component-based interfaces
- Web programmers maintaining older server-rendered sites who are moving to JavaScript-based application stacks
- Software engineers in internal IT teams who build line-of-business web applications end to end
- Technical team members who review code and pull requests across both client and server parts of a web product

CONCEPT

Course Outline:

Day 1: Modern Web Platform, JavaScript and Full-Stack Architecture

- Full-Stack Architecture Layers: Front End, Back End and Relational Database in the PERN Stack
- Semantic HTML Structure and Responsive CSS Layout with Flexbox and Grid
- JavaScript ES Modules: Named and Default Exports, Imports and Dynamic Import
- Asynchronous JavaScript with Promises, Async/Await and the Fetch API
- Git Repository Setup, Feature Branching and Pull Request Workflow

Day 2: React Components, State and Client-Side Routing

- React Components, JSX and Props for Reusable Interface Parts
- State with the useState Hook, Rules of Hooks and Lifting State Up
- Side Effects and Remote Data Loading with the useEffect Hook
- Client-Side Routing: Nested Routes, Route Parameters and Guarded Pages
- Controlled Forms, Client-Side Validation and Loading and Error States

Day 3: REST API Design, Node.js and Express Back End and Relational Data

- REST Constraints and Resource-Oriented Endpoint Design with HTTP Methods and Status Codes
- Express Routing, Middleware Chain and Centralised Error Handling in Node.js
- Relational Data Modelling: Entities, Primary and Foreign Keys, One-to-Many and Many-to-Many Relations
- ORM Models, Schema Migrations and Seed Data for PostgreSQL
- SQL Joins, Pagination, Filtering and Transactions Through the ORM

Day 4: Authentication, Automated Testing and Secure Coding Habits

- Server Sessions Versus JSON Web Tokens: Claims, Signing, Expiry and Refresh
- Login Flow, Password Hashing and Role Checks in Express Middleware
- Secure Coding Habits: Input Validation, Environment Secrets and CORS Settings
- Unit Tests for React Components and JavaScript Functions with a Test Runner
- API Integration Tests and Browser End-to-End Tests for Critical User Journeys

Day 5: Containerised Deployment, CI Pipeline and the Capstone Application

- Dockerfile and Multi-Container Setup for Front End, API and Database
- CI Pipeline Stages: Lint, Test and Build on Every Pull Request
- Cloud Platform Release: Environment Variables, Managed Database and Application Logs
- Lab: Capstone Full-Stack Application Build in Pairs from a Feature Backlog
- Capstone Deployment, Code Walkthrough and Peer Review

Skills You Will Gain:

- React Component Development
- REST Endpoint Design
- Express Middleware Implementation
- Relational Schema Modelling
- ORM Query Writing
- Token and Session Authentication
- Automated Test Authoring
- Container and CI Release

Why Attend This Course:

- Leave with a deployed capstone full-stack application, its Git repository, test suite and CI pipeline as a reusable starting point
- Deliver features end to end without waiting on separate front-end, back-end and operations hand-offs
- Write and debug real code in daily labs on your own laptop rather than following slide walkthroughs
- Compare stack choices and team practices with developers from banking, retail, government services and technology firms

Conclusion:

Full-stack web development is a craft of connecting layers so that a screen, an endpoint, a table and a release all fit together. The week moves from HTML, CSS, JavaScript modules and Git, through React components, state and routing, to REST design, Express middleware and relational data through an ORM. It then adds authentication, secure coding habits and automated tests, and ends with containers, a CI pipeline and a capstone full-stack application deployed to a cloud platform.