

Full-Stack Software Development for Web, Mobile, and Desktop Applications

Duration: 15 Days

Suggested Technology Stack

- HTML5, CSS3 and JavaScript
- TypeScript
- React
- Node.js and Express.js
- REST APIs
- SQL and NoSQL Databases
- React Native
- Electron.js
- Git and GitHub
- Testing and deployment tools

Prerequisites

- Basic programming knowledge
- Understanding of variables, conditions, loops and functions
- Basic knowledge of databases is helpful
- Familiarity with command-line tools is beneficial

Module 1: Software Development Foundations

Day 1: Software Development Lifecycle, Development Environment and Modern JavaScript

Topics

- Introduction to web, mobile and desktop software applications
- Software Development Life Cycle; Agile and Scrum fundamentals
- Functional and non-functional requirements
- Client-server, monolithic and layered architectures
- Configuration of Visual Studio Code, Node.js, npm, Git and browser developer tools
- Git and GitHub fundamentals: repositories, commits, branches, merges and push operations
- JavaScript syntax, variables, data types, operators, conditions and loops
- Functions, arrow functions, arrays, objects, destructuring, spread/rest operators and modules
- Exception handling and browser-based debugging

Hands-on Labs

- Configure the complete development environment and create a Git repository
- Build and debug a JavaScript-based employee record manager
- Analyse requirements and outline the architecture of a sample business application

Outcome

Participants will understand the software development process, configure the required tools and build modular programs using modern JavaScript.

Day 2: Object-Oriented Programming and TypeScript

Topics

- Object-oriented programming concepts: classes, objects, encapsulation, inheritance, polymorphism and abstraction
- Interfaces and reusable object-oriented designs
- TypeScript type annotations, arrays, tuples and enums
- Interfaces, type aliases, classes and access modifiers
- Generics and reusable typed components
- TypeScript configuration and project structure
- Compiling, running and debugging TypeScript applications

Hands-on Labs

- Convert an existing JavaScript application into TypeScript
- Build a class-based inventory management module using interfaces and generics

Outcome

Participants will be able to apply object-oriented design and TypeScript to create maintainable, strongly typed applications.

Module 2: Web Application Development

Day 3: HTML5, CSS3, Responsive Design and Browser Programming

Topics

- HTML5 document structure, semantic markup, forms, validation, tables and multimedia
- Accessibility fundamentals, WCAG overview and search-engine-friendly markup
- CSS selectors, box model, typography and visual hierarchy
- Flexbox, CSS Grid, media queries, mobile-first design and CSS variables
- Transitions, animations and overview of Bootstrap or another CSS framework
- Document Object Model, element selection and modification
- Events, form handling, local storage and session storage
- JSON handling, promises, async/await, Fetch API and asynchronous error handling

Hands-on Labs

- Create a responsive, accessible multi-page corporate website
- Build a browser-based task management application with local storage
- Retrieve and display data from a REST API

Outcome

Participants will be able to build responsive, accessible and dynamic browser applications that consume asynchronous services.

Day 4: React Fundamentals and Component-Based Interfaces

Topics

- Introduction to component-based development and React application architecture
- Creating and structuring a React project
- Functional components and JSX
- Properties, component communication and composition
- State management with useState
- Event handling and conditional rendering
- Rendering lists and reusable user-interface components
- React development tools and debugging

Hands-on Labs

- Build reusable user-interface components
- Develop a searchable product catalogue application

Outcome

Participants will be able to design and build component-based web applications using React.

Day 5: Advanced React Application Development

Topics

- React hooks: useEffect, useRef and custom hooks
- Forms, controlled components and validation
- React Router, nested routes and protected routes
- API integration with loading, empty and error states
- Context API and introduction to application state management
- Search, sorting, filtering and pagination in user interfaces
- Performance considerations and component optimization

Hands-on Labs

- Develop a multi-page customer management application

- Implement navigation, validation, API integration and protected routes

Outcome

Participants will be able to develop structured, data-driven and multi-page React applications.

Module 3: Backend and Database Development

Day 6: Backend Development with Node.js and Express

Topics

- Backend responsibilities and Node.js architecture
- Node package management and project organization
- Express.js fundamentals
- Routes, controllers and middleware
- Request and response handling
- Query, route and body parameters
- REST API design principles
- HTTP methods and status codes
- Centralized error handling and logging
- API testing using Postman

Hands-on Labs

- Build a REST API for employee management
- Test and troubleshoot API endpoints using Postman

Outcome

Participants will be able to design, implement and test REST APIs using Node.js and Express.

Day 7: Database Design and Backend Integration

Topics

- Relational and non-relational database concepts
- Data modelling, tables, records, relationships, primary keys and foreign keys
- Normalization overview
- SQL CRUD operations, joins and aggregations
- MongoDB documents, collections and CRUD concepts
- Connecting Node.js with SQL or MongoDB databases
- Repository and service patterns
- Database validation, transactions overview and error handling

Hands-on Labs

- Design a database for an order management system
- Connect the REST API to a SQL or MongoDB database
- Implement validated CRUD operations

Outcome

Participants will be able to design application data models and integrate backend services with relational or NoSQL databases.

Day 8: Authentication, API Security and Full-Stack Integration

Topics

- Authentication and authorization
- Password hashing and secure credential handling
- JSON Web Tokens and login/registration workflows
- Role-based access control and secured API routes
- Input validation, sanitization, CORS and secure HTTP headers
- Environment variables and configuration management
- Common web vulnerabilities and OWASP security awareness
- Connecting React with REST APIs
- Authentication state management and role-based navigation

- CRUD interfaces, search, filtering, pagination and global error handling
- File upload overview and production-ready application structure

Hands-on Labs

- Implement user registration, login and JWT-based authentication
- Secure API routes using user roles
- Integrate the React frontend with the Node.js backend and database

Outcome

Participants will complete a secure full-stack web application with authentication, authorization and CRUD functionality.

Module 4: Mobile Application Development

Day 9: React Native Fundamentals, Navigation and Forms

Topics

- Native, hybrid and cross-platform mobile applications
- React Native ecosystem, Expo and command-line development approaches
- Project structure and development environment setup
- Core components: views, text, images, buttons and lists
- Styling with Flexbox and responsive mobile layouts
- Platform-specific behaviour and running applications on emulators/devices
- Stack and tab navigation
- Passing data between screens
- Mobile forms, validation and touch interactions
- Local storage and secure storage overview

Hands-on Labs

- Configure a React Native development environment
- Build a multi-screen employee directory application
- Implement a mobile registration and profile workflow

Outcome

Participants will be able to create navigable, responsive cross-platform mobile applications.

Day 10: Mobile API Integration, Device Features, Testing and Packaging

Topics

- Connecting mobile applications to REST APIs
- Authentication, token storage and token renewal
- Displaying, creating and updating server data
- Network error handling, offline/online awareness and pull-to-refresh
- Loading, empty and error states
- Camera, image picker, location services and permissions
- Handling Android and iOS differences
- Mobile application security and configuration
- Component and functional testing
- Performance optimization, efficient list rendering and asset optimization
- Crash/log analysis, icons, splash screens, Android builds and signing overview

Hands-on Labs

- Connect the mobile application to the common backend API
- Implement authenticated API operations and image selection
- Test and optimize the application
- Generate a testable Android application package

Outcome

Participants will be able to build, test, optimize and package a secure mobile application connected to backend services.

Module 5: Desktop Application Development

Day 11: Desktop Application Development with Electron

Topics

- Desktop development approaches and introduction to Electron.js
- Electron architecture: main and renderer processes
- Creating and structuring an Electron project
- Browser windows and application lifecycle
- Desktop menus, dialog boxes and system notifications
- Inter-process communication
- Integrating React with Electron
- Desktop application security considerations

Hands-on Labs

- Create an Electron desktop application
- Implement menus, dialogs, system notifications and IPC
- Integrate a React interface into Electron

Outcome

Participants will be able to create cross-platform desktop applications using Electron and React.

Day 12: Desktop Data, File System, API Integration and Packaging

Topics

- Reading and writing files using the local file system
- Importing and exporting data
- Local database options and SQLite overview
- Desktop configuration management
- Connecting desktop applications to REST APIs
- Authentication and secure storage of sensitive information
- Offline data handling and synchronization strategies
- Auto-update concepts and error logging
- Performance, packaging and installer considerations

Hands-on Labs

- Build a desktop data management application
- Add file import/export and local data storage
- Connect the desktop application to the common backend API
- Generate a distributable desktop package

Outcome

Participants will be able to develop and package desktop applications with local and remote data access.

Module 6: Quality, Deployment and Capstone Project

Day 13: Application Testing, DevOps and Deployment

Topics

- Software quality assurance and testing strategy
- Unit, integration and end-to-end testing
- Testing JavaScript and TypeScript applications
- API testing and frontend component testing
- Code quality, linting and debugging across platforms
- Git branching strategies and code review practices
- Introduction to continuous integration and delivery
- Environment-based configuration
- Web application and backend API deployment
- Mobile and desktop packaging considerations
- Application monitoring and logging

Hands-on Labs

- Write tests for frontend and backend components
- Configure a basic continuous integration workflow
- Build production packages for web and desktop applications
- Prepare deployment documentation

Outcome

Participants will be able to test, package and prepare multi-platform applications for deployment.

Day 14: Integrated Capstone Project – Architecture and Development

Topics

- Capstone scope selection and requirement finalization
- Architecture of a shared Node.js REST API and centralized database
- Planning responsive web, cross-platform mobile and desktop clients
- Authentication, role-based access, dashboard and CRUD design
- Source-control workflow and task allocation
- Incremental integration and code reviews

Hands-on Labs

- Create the capstone repository, architecture and database model
- Develop the shared backend API and authentication layer
- Implement core web, mobile and desktop features
- Perform intermediate integration and peer code review

Outcome

Participants will apply the complete technology stack to develop an integrated business application across web, mobile and desktop platforms.

Day 15: Capstone Integration, Testing, Deployment and Presentation

Topics

- Final cross-platform integration
- Data validation, search, filtering and centralized error handling
- Testing and issue resolution
- Deployment or packaging documentation
- Project demonstration and architecture presentation
- Trainer feedback and knowledge assessment

Hands-on Labs

- Complete and test the integrated capstone solution
- Package or deploy the application components
- Demonstrate the web, mobile and desktop interfaces
- Present the architecture, design decisions and lessons learned

Outcome

Participants will demonstrate their ability to design, develop, integrate, test and present a full-stack multi-platform application.

Suggested Capstone Project Options

- Employee leave and attendance management system
- Service request and help-desk system
- Inventory and order management system
- Learning and training management system
- Field-service management application

Minimum Capstone Requirements

- User registration and authentication
- Role-based access control
- Dashboard and CRUD operations
- Search, sorting and filtering

- Data validation and centralized error handling
- Responsive web interface
- Cross-platform mobile application interface
- Cross-platform desktop application interface
- Shared REST API and centralized database
- Basic automated testing
- Source-code management
- Deployment or packaging documentation