Full Stack Development Course

Level 1: Beginner (3 Months)

Phase 1: Frontend Basics with Tailwind CSS and JavaScript (1 Month)

Topics to Learn:

1. **HTML**
    a. Semantic HTML and its importance.
    b. Forms, tables, and multimedia integration.
    c. Accessibility and SEO basics.
2. **CSS**
    a. Box model, Flexbox, and Grid layout systems.
    b. Responsive design using media queries.
3. **Tailwind CSS**
    a. Utility-first CSS principles.
    b. Building responsive components and layouts with Tailwind classes.
    c. Theme customization and dark mode setup.
4. **JavaScript Basics**
    a. Data types, variables, operators, and conditionals.
    b. Loops, functions, and DOM manipulation.
    c. Event handling and debugging techniques.

Expected Outcome:

- Ability to create responsive, accessible, and visually appealing webpages.
- Example Project: A responsive landing page for a product or service.

Phase 2: React Basics (1 Month)

Topics to Learn:

1. **React Setup**
    a. Introduction to React and creating projects with create-react-app or Vite.
    b. JSX syntax and functional components.
2. **State Management**

    a. Using useState for managing component-level state.
    b. Props and passing data between components.
3. **React Router**
    a. Setting up single-page applications (SPA).
    b. Navigation between pages with dynamic routes.

Expected Outcome:

- Ability to build single-page applications (SPAs) with React.
- Example Project: A basic task manager app with navigation.

Phase 3: Backend Basics with Node.js and MongoDB (1 Month)

Topics to Learn:

1. **Node.js Basics**
    a. Setting up a Node.js project.
    b. Working with the file system and modules.
2. **Express.js**
    a. Creating a REST API.
    b. Middleware, routing, and handling HTTP methods.
3. **MongoDB**
    a. NoSQL concepts and CRUD operations.
    b. Introduction to Mongoose for schema definition and data validation.

Expected Outcome:

- Ability to create a backend server and connect it to a database.
- Example Project: A simple blog API with endpoints for creating and reading posts.

Level 2: Advanced (3 Months)

Phase 1: Advanced Frontend Features (1 Month)

Topics to Learn:

1. **React Advanced Concepts**

    a. Using Context API for global state management.
    b. Optimizing performance with React.memo, useMemo, and lazy loading.
2. **Tailwind CSS Advanced**
    a. Using plugins and customizing configurations.
    b. Advanced animations and transitions.
3. **Testing and Debugging**
    a. Writing tests with Jest and React Testing Library.
    b. Debugging tools like React DevTools.

Expected Outcome:

- Build scalable and high-performance React applications.
- Example Project: A dashboard app with real-time updates.

Phase 2: Advanced Backend Development (1 Month)

Topics to Learn:

1. **Node.js Advanced Concepts**
    a. Building real-time features with WebSocket.
    b. Handling file uploads and large-scale data processing.
2. **MongoDB Optimization**
    a. Query optimization and indexing.
    b. Aggregation framework for analytics.
3. **Security and Authentication**
    a. Implementing OAuth and role-based access control.
    b. Data validation and protection against common vulnerabilities.

Expected Outcome:

- Build secure and efficient backend systems with real-time capabilities.
- Example Project: A live chat or notification system.

Phase 3: Full Stack Integration and Deployment (1 Month)

Topics to Learn:

1. **Frontend and Backend Integration**
    a. Fetching data with axios or fetch.
    b. Handling authentication and protecting routes.
2. **Deployment**
    a. Hosting frontend on Vercel/Netlify and backend on Render/Heroku.
    b. Using MongoDB Atlas for database hosting.
3. **Capstone Project**
    a. Combine all concepts to build a feature-rich, production-ready application.

Expected Outcome:

- Build and deploy a full-stack application.
- Example Project: An e-commerce platform with user authentication and payment integration.

Flutter Mobile App Development Course

Level 1: Beginner (3 Months)

Phase 1: Flutter and Dart Basics (1 Month)

Topics to Learn:

1. **Dart Programming Basics**
    a. Variables, loops, functions, and OOP concepts.
    b. Asynchronous programming with Future and Stream.
2. **Flutter Setup**
    a. Installing Flutter SDK and setting up the environment.
    b. Overview of Flutter architecture and widget tree.
3. **UI Components**
    a. Stateless and Stateful widgets.
    b. Layouts: Row, Column, Container, and Stack.

Expected Outcome:

- Build static UI with user interaction.
- Example Project: A basic counter app.

Phase 2: State Management and Navigation (1 Month)

Topics to Learn:

1. **State Management**
    a. Using setState for local state management.
    b. Basics of Provider for global state.
2. **Navigation**
    a. Multi-screen apps with Navigator 1.0.
    b. Passing arguments between screens.
3. **Theming**
    a. Creating light and dark themes.
    b. Customizing app-wide styles.

Expected Outcome:

- Build apps with dynamic multi-screen navigation.
- Example Project: A simple recipe app with a details page.

Phase 3: Backend Integration (1 Month)

Topics to Learn:

1. **Networking**
    a. Fetching data from REST APIs using http.
    b. Handling JSON data and displaying it in widgets.
2. **Persistent Storage**
    a. Using Shared Preferences for local storage.
    b. Basics of SQLite for structured data storage.

Expected Outcome:

- Build data-driven apps with API integration.
- Example Project: A weather app fetching live data.

Level 2: Advanced (3 Months)

Phase 1: Advanced State Management (1 Month)

Topics to Learn:

1. **State Management**
    a. Advanced usage of Provider and ChangeNotifier.
    b. Introduction to BLoC (Business Logic Component).
2. **Error Handling**
    a. Managing errors in network calls.
    b. Using try-catch blocks and custom error widgets.

Expected Outcome:

- Build robust and maintainable apps with advanced state handling.
- Example Project: A news app with categorized content.

Phase 2: Advanced UI and Animations (1 Month)

Topics to Learn:

1. **Animations**
    a. Using AnimationController and implicit animations.
    b. Creating custom animations for widgets.
2. **Device Features**
    a. Accessing camera, GPS, and sensors.
    b. Integrating notifications and background tasks.

Expected Outcome:

- Build visually engaging apps with advanced animations and native features.

- Example Project: A fitness tracker app with activity animations.

Phase 3: Deployment and Capstone Project (1 Month)

Topics to Learn:

1. **App Deployment**
    a. Optimizing apps for performance.
    b. Publishing on Google Play and Apple App Store.
2. **Capstone Project**
    a. Combine all skills to create a feature-rich app.

Expected Outcome:

- Build and publish a production-ready app.
- Example Project: An e-commerce app with cart, authentication, and payment.