


Front-End Development with React.js (Beginner to Intermediate)

With this course, students will gain hands-on experience building a full-stack application from scratch. You'll learn how to build the front-end using React.js and Tailwind CSS and the back-end using Appwrite for database management and authentication.

You will learn to:

- Become a master in CRUD operations for reading, creating, updating, and deleting blog posts.
- Build user authentication for secure sign-up, login, and logout.
- Create responsive UIs with the new tools like React.js and Tailwind CSS.
- Implement the front-end with a live database through Appwrite SDK.
- Get hands-on experience with basic concepts like API integration, state management, and routing.

Before starting the course, Student are required to download these softwares in their computer

- **Visual Studio Code (VS Code)** [Download VS Code](#) (Compatible with Windows, macOS, and Linux)
- **Node.js**  [Download Node.js](#)

Syllabus

Topic No.	Title	Content
1	HTML5 and CSS	<ul style="list-style-type: none">- Introduction to HTML5- Elements & Attributes- Headings, Paragraphs, Text Formatting- Quotation & Citation- Comments- Links (Internal, External, Anchor)- Tables (Basic to Complex)- Lists (Ordered, Unordered, Definition)- Forms (Input Types, Form Attributes)- Multimedia: Audio, Video, and Images

		<ul style="list-style-type: none"> - Embedding Google Maps and YouTube Videos - Syntax & Selectors (Basic and Advanced) - Colors (RGBA, Gradients) - Backgrounds (Images, Gradients) - Margins & Padding (Box Model) - Height & Width (Units: px, %, em, rem) - Text Styling (Fonts, Icons) - Lists & Tables Styling - Display: Inline, Block, Inline-Block - CSS Positioning (Static, Relative, Absolute, Fixed) - Flexbox (Responsive Layouts)
2	JavaScript (ES6+)	<ul style="list-style-type: none"> - Introduction to JavaScript - Syntax & Statements - Comments (Single-Line, Multi-Line) - Variables (let, const, var Comparison) - Operators (Arithmetic, Assignment, Comparison) - Data Types (Primitives and Objects) - Functions (Function Declarations, Arrow Functions) - Objects and Arrays (Methods, Iteration) - Control Flow (if-else, switch) - DOM Manipulation (Query Selectors, Event Handling) - JSON (Parsing, Stringifying) - Form Validation (Client-Side) - Modern JavaScript: Modules, Promises, async await
3	Tailwind CSS	<ul style="list-style-type: none"> - Introduction to Tailwind CSS (Utility-First Framework) - Setup: Integrating Tailwind with React (Vite) - Responsive Design with Breakpoints - Utility Classes: Colors, Typography, Spacing, Borders - Flexbox & Grid Layout (Building Responsive UIs) - Customizing Tailwind with tailwind.config.js - Handling Forms and Inputs with Tailwind

4	React.js	<ul style="list-style-type: none"> - Introduction to React: What is React? Why React? - Understanding Virtual DOM - Environment Setup: Create a React App using Vite - Folder Structure and Component Organization - Components: Functional and Class (Focus on Functional Components) - JSX: Syntax, Expressions in JSX - Props: Passing Data Between Components - State Management: Using useState Hook - Conditional Rendering: Ternary Operator, Short-circuit Evaluation - Event Handling: Handling Events in React - Form Handling: Controlled vs. Uncontrolled Components - Hooks: useEffect, useRef, useContext - React Router DOM: Setting up Multi-Page Navigation (v6) - State Management with Redux Toolkit: Setting up Store, Slices, Using useDispatch and useSelector - Context API: Managing Global State without Redux - API Integration: Fetching Data using axios and fetch - React Lifecycle: Component Mounting, Updating, and Unmounting - Advanced Topics: Next.js Introduction, Server-side Rendering (SSR) - Integrating External Libraries: Axios for REST API Communication
5	Appwrite	<ul style="list-style-type: none"> - Introduction to Appwrite: What is Appwrite? Why use it? - Environment Setup: Creating and Configuring Appwrite Project - Database Management: Creating and Managing Collections - CRUD Operations: Create, Read, Update, Delete Data from Appwrite Database - User Authentication: Sign Up, Sign In, Logout (Session Management)

		<ul style="list-style-type: none"> - File Storage: Uploading and Fetching Files - Security & Permissions: Managing Access Control (Roles and Policies) - Appwrite SDK Integration: Connecting React App with Appwrite Backend
--	--	--

PROJECTS

CRUD Application

- Description: Built a fully functional Create, Read, Update, Delete (CRUD) application to manage dynamic data.
- Technologies Used: React.js, Tailwind CSS, JavaScript (ES6+), Appwrite
- Key Features:
 - Implemented core CRUD operations with an intuitive and responsive UI.
 - Integrated Appwrite backend for database management and user authentication.
 - Ensured optimal performance with efficient state management using React Hooks.

Full Stack Blog Application

Description: Developed a blog platform allowing users to create, edit, and manage blog posts with secure authentication.

- Technologies Used: React.js, Redux Toolkit, Tailwind CSS, Appwrite
- Key Features:
 - User Authentication: Implemented secure sign-up, login, and logout using Appwrite SDK.
 - State Management: Utilized Redux Toolkit for managing complex application state.
 - Responsive Design: Built a fully responsive UI with Tailwind CSS and modern styling practices.
 - API Integration: Leveraged axios for seamless REST API communication with the Appwrite backend.