Full Stack Web Development Beginners

Learn how to create world-class web applications with high-class user interfaces. At the end of the program, you will be able to create your own business and individual advanced and dynamic web applications and host to an online server for global access or even sell to web markets.

Requirement

- Be Able to Use PC at A Beginner Level, Including Being Able to Install Programs
- Prior Knowledge of the English language and good listening
- A Desire to learn Web Development.

Prerequisite

This program requires no prerequisite courses. It's designed for beginners learning from scratch. Our goal is to help you go from 0 to 100 and learn enough to learn more.

How do you teach?

We teach in real-time online 1-on-1 using Google Meet, Skype, or Zoom. You can ask any questions any time, and you will get them answered. Our tutors are ready to mentor you.

Do I get a certificate?

Yes. We will provide you a Course Certificate on the condition that you complete and submit all projects and assignments by the end of the course. The certificate is not academic; it is a professional training certificate.

Do I need equipment?

Yes. You need your computer. Windows, Mac, and Linux operating systems are all supported by the curriculum. You also need to have a stable internet connection.

Do you accept beginners?

Yes. The program is from zero to hero, so all knowledge will be covered. No much experience required to join this course.

What is Full stack development?

Full-stack development: It refers to the development of both front end (client-side) and back end (server-side) portions of the web application.

What can you do?

We will guide you through to become a full-stack web developer.

Why Full stack development?

This Web Development course is without a doubt one of the most comprehensive web development courses available online. Even if you have zero programming experience, this course will take you from beginner to mastery.

What you'll learn?

- Build Full Stack JavaScript, ReactJs, Redux, Nodejs, and many other useful libraries.
- Work on the web, creating high-class user interfaces. Building reusable components.
- Gain an understanding of how to turn your ideas into a reality
- How to use remote version controls platform, especially Git and GitBash as well
- How to build secured and reliable restful APIs
- How to build real-time data sharing apps
- How to build complex UI and UI services
- How to turn a business plan to an application
- And many more

Course Content

1. Unlocking the power of HTML

- HTML: Section Overview
- HTML Fundamentals
- Create a H1 heading and a paragraph.
- Block & Inline Elements + HTML Entities
- Build a Basic HTML Site Project
- Build a Basic HTML Site Solution
- Learning Tables and Forms
- Divs & Spans
- Classes & ID's
- What is Semantic HTML? HTML5 Tags
- Coding Project | Transform your Skillset
- Coding Project Walkthrough
- Recapping HTML
- Create a HTML Site about your Favorite Movie

- Image Gallery
- Debugging HTML
- What is the difference between an Unordered and Ordered List?

2. The Beauty of CSS

- CSS: Section Overview
- CSS Fundamentals
- Styling Classes/ID's
- Further Selector Knowledge
- Linking up Pages & Styles
- Typography Rules
- Setting up Backgrounds
- The Box Model
- Floats
- Coding Project | Favorite Band Website
- Coding Project Review | Band Website
- Positioning
- The Art of Cascading
- Media Queries & Responsive Design
- Coding Project | Crafting a Portfolio Site
- Coding Project Walkthrough | Portfolio
- CSS3
- Gradients + Shadows
- CSS3 Animations & Transitions

3. Bootstrap

- Bootstrap: Overview
- What is it + How Do We Use it?
- The Bootstrap Grid
- Typography + Buttons
- Forms & Tables
- Other Components
- Building a Site with Bootstrap
- Rebuild Wikipedia in Bootstrap
- Wikipedia in Bootstrap Walkthrough Part 1.
- Wikipedia in Bootstrap Walkthrough Part 2.
- Independent Project

4. JQuery

- jQuery - Overview
- jQuery - Basics
- jQuery - Selectors
- jQuery - Attributes
- jQuery - Traversing
- jQuery - CSS
- jQuery - DOM

- jQuery - Events
- jQuery - AJAX
- jQuery - Effects
- jQuery UI
- jQuery - Interactions
- jQuery - Widgets

5. Beginner JavaScript

- Beginner JavaScript Overview
- Variables & Datatypes
- The Console
- Logic & Conditionals
- Conditionals Extra | The Switch Statement
- Should the Employee Be Payed Overtime?
- Functions
- Scope
- Loops | Part One
- Loops | Part Two
- Mini Project | Number of Letters
- Arrays

6. React.js

- What is React?
- What we will make in this React module
- Introduction to Code Sandbox and the Structure of the Module
- Introduction to JSX and Babel
- JSX Code Practice
- JavaScript Expressions in JSX & ES6 Template Literals
- JavaScript Expressions in JSX Practice
- JSX Attributes & Styling React Elements
- Inline Styling for React Elements
- React Styling Practice
- React Components
- React Components Practice
- JavaScript ES6 - Import, Export and Modules
- JavaScript ES6 Import, Export and Modules Practice
- [Windows] Local Environment Setup for React Development
- [Mac] Local Environment Setup for React Development
- Keeper App Project - Part 1 Challenge
- Keeper App Part 1 Solution
- React Props
- React Props Practice
- React DevTools
- Mapping Data to Components
- Mapping Data to Components Practice
- JavaScript ES6 Map/Filter/Reduce
- JavaScript ES6 Arrow functions
- Keeper App Project - Part 2
- React Conditional Rendering with the Ternary Operator & AND Operator
- Conditional Rendering Practice
- State in React - Declarative vs. Imperative Programming
- React Hooks - useState

- useState Hook Practice
- JavaScript ES6 Object & Array Destructuring
- JavaScript ES6 Destructuring Challenge Solution
- Event Handling in React
- React Forms
- Class Components vs. Functional Components
- Changing Complex State
- Changing Complex State Practice
- JavaScript ES6 Spread Operator
- JavaScript ES6 Spread Operator Practice
- Managing a Component Tree
- Managing a Component Tree Practice
- Keeper App Project - Part 3
- React Dependencies & Styling the Keeper App

7. Node.js

- Node.js - Introduction
- Node.js - Environment Setup
- Node.js - First Application
- Node.js - REPL Terminal
- Node.js - Package
- Manager (NPM)
- Node.js - Callbacks Concept
- Node.js - Event Loop
- Node.js - Event Emitter
- Node.js - Buffers
- Node.js - Stre.ms
- Node.js - File System
- Node.js - Global Objects
- Node.js - Utility Modules
- Node.js - Web Module
- Node.js - Express Framework
- Node.js - RESTful API
- Node.js - Scaling Application
- Node.js - Packaging