Solardad Institute Of Information Technology(SIIT), Okene

Professional Diploma
in
Web Development
Programme

TAUGHT BY	COURSE LENGTH	COURSE FEE
Solardad Institute of Information Technology (SIIT)	18 Months	N 35,000 (Per Term)

SYLLABUS

PROFESSIONAL DIPLOMA

IN WEB DEVELOPMENT

INTRODUCTION

Welcome to the **Solardad Institute of Information Technology Web Development Bootcamp**, an intensive, hands-on program designed to transform beginners and aspiring developers into job-ready web developers. This bootcamp equips students with the skills to build modern, responsive, and accessible web applications using industry-standard tools and practices. Our goal is to empower you to create dynamic, user-focused websites and prepare you for a successful career in the tech industry.

The bootcamp focuses on:

- **Practical Skill Development**: Master core web technologies like HTML, CSS, JavaScript, React.js, Node.js, and databases through real-world projects.

- **Career Readiness**: Build a professional portfolio, refine your resume, and practice for technical interviews to stand out in the job market.

- **Collaboration and Community**: Foster teamwork through pair programming, group projects, and peer learning to mirror real-world development environments.

- **Industry-Relevant Practices**: Learn version control, deployment, and accessibility standards to meet the demands of modern web development.

By the end of the program, you will have the confidence and expertise to develop full-stack web applications and launch your career as a web developer.

Learning Outcomes

Upon successful completion of the bootcamp, students will be able to:

- **Build Responsive Websites**: Create visually appealing, responsive, and accessible websites using HTML, CSS, and JavaScript.

- **Develop Dynamic Web Applications**: Use React.js (or similar frameworks) to build interactive, component-based user interfaces.

- **Create Backend Systems**: Implement server-side logic, APIs, and databases using Node.js, Express.js, and SQL/NoSQL.

- **Collaborate Effectively**: Use Git and GitHub for version control and team collaboration on codebases.

- **Deploy Applications**: Deploy web applications to platforms like Netlify, Vercel, or Heroku with confidence.

- **Showcase Professional Work**: Produce aapstone project and a polished portfolio to demonstrate skills to potential employers.

- **Prepare for Interviews**: Articulate technical concepts and perform well in

SYLLABUS

PROFESSIONAL DIPLOMA

IN WEB DEVELOPMENT

COURSE FEE

The tuition for the Solardad Web Development Bootcamp is **N35,000**, payable every four months. This fee covers:

- Access to all course materials, including lectures, assignments, and project resources.

- Instructor-led sessions, code reviews, and one-on-one mentorship.

- Career support services, including resume reviews and mock interviews.

- Access to a dedicated online platform for coursework and collaboration.

Payment Schedule:

- Payments are due at the start of each four-month period.

- Flexible payment plans may be available; contact the Solardad Institute administration for details.

TEACHING METHODS

The **Solardad Web Development Bootcamp** uses a hands-on, collaborative approach to build technical and professional skills over 18 months, mirroring real-world development environments.

- **Hands-On Coding Sessions**:
  - Daily labs with instructor-led coding (e.g., building responsive navbars, fetching APIs). • Structured exercises using tools like VS Code and cloud environments.

- **Pair Programming, Peer Learning, and Group Projects**:
  - Weekly pair programming (driver/navigator roles) for tasks like React components. Peer discussions to share solutions (e.g., JavaScript challenge walkthroughs). Team projects (e.g., full-stack event app) using agile workflows and GitHub. *Advice*: Practice clear communication in pair programming and use tools like Trello for group projects.

- **Weekly Assignments and Code Reviews**:
  - Weekly tasks (e.g., CSS animations, APIs) submitted via the learning platform. • Code reviews by instructors/peers via GitHub, focusing on structure and best practices. *Example*: Review API code for error handling and naming conventions.

COURSE CONTENT

Web Development Fundamentals

- o *How Websites Work: Understand the roles of HTML, CSS, and JavaScript in website creation.*

- o *How the Internet Works: Learn how websites are served and delivered to users.*

- o *Git, GitHub, and Version Control: Master version control workflows, including forking, branching, and cloning.*

- o *Troubleshooting and Debugging: Develop skills to identify and resolve issues in code and projects.*

HTML5

- o *HTML Syntax and Structure: Learn the anatomy of HTML, including boilerplate, doctypes, indentation, and nesting.*

- o *HTML Tags: Use tags for headings, paragraphs, lists (ordered and unordered), images, and hyperlinks.*

- o *Multi-Page Websites: Create websites with multiple pages and consistent navigation.*

- o *HTML Best Practices: Apply standards for clean, semantic, and maintainable code.*

CSS3

- o *Cascading Style Sheets: Understand CSS selectors, properties, and styling techniques.*

- o *CSS Integration: Use inline, internal, and external CSS for flexible styling.*

- o *CSS Best Practices: Learn specificity, inheritance, and the CSS Box Model.*

- o *Developer Tools: Inspect and diagnose CSS using browser developer tools.*

- o *Positioning and Display: Master CSS positioning, display properties, float, and clear.*

- o *Font Styling: Apply web-safe fonts and modern typography techniques.*

- o *Responsive Design: Implement media queries for mobile-first, responsive layouts. Selector Priority: Combine selectors and understand priority rules.*

SYLLABUS

PROFESSIONAL DIPLOMA

IN WEB DEVELOPMENT

COURSE CONTENT (Continued)

Flexbox

o *Introduction to Flexbox: Understand the purpose and benefits of Flexbox in CSS.*

o *Flexbox Layout: Use flex direction, align, and justify properties to control layout.*

o *Flexbox Sizing: Apply the flex shorthand property for sizing and distributing child items.*

CSS Grid

o *Grid vs. Flexbox: Compare strengths and weaknesses of Grid and Flexbox.*

o *Grid Layout: Create and arrange grids using grid sizing, rows, and columns.*

o *Grid Positioning: Position items across rows and columns for complex layouts.*

o *Combining Grid and Flexbox: Use both tools to build sophisticated, native CSS layouts.*

Bootstrap

o *Native CSS vs. Frameworks: Understand the role of external frameworks like Bootstrap.*

o *Bootstrap Installation: Integrate Bootstrap into websites.*

o *Bootstrap Layout & Components: Use the 12-column grid system for responsive design. Implement buttons, carousels, cards, and navigation bars.*

o *Designer Icons: Incorporate Bootstrap icons and use pre-built templates for rapid development.*

Web Design

o *Web Design Principles: Learn UI and UX fundamentals for user-friendly websites.*

o *Color Theory & Typography: Choose modern color palettes for professional aesthetics. Select free, commercial-use typefaces for readability and style.*

o *User Interface (UI) Design: Manage user attention with effective UI principles.*

COURSE CONTENT (Continued)

Document Object Model (DOM)

- o *DOM Structure: Understand the tree structure of HTML-based websites.*
- o *DOM Traversal: Navigate and manipulate HTML elements using object notation.*
- o *DOM Manipulation: Change elements dynamically with JavaScript.*
- o *Separation of Concerns: Apply coding best practices for maintainable code.*

JavaScript ES6

- o *JavaScript Overview: Compare JavaScript with markup and stylesheet languages. Code Editors and IDEs: Work effectively with development environments.*
- o *Basic Syntax and Data Types. Variables: Use let and const with proper scope. Data Types: Work with strings, numbers, booleans, null, undefined, and symbols.*
- o *Operators: Apply arithmetic, comparison, and logical operators.*
- o *Control Structures:*
- o *Loops: Use for, while, for...of, and for...in for iteration.*
- o *Conditionals: Implement if/else statements for logic control.*
- o *Functions: Function declarations, expressions, arrow functions, and closures. Higher-order functions: Use map, filter, and reduce for array manipulation.*
- o *Arrays: Create, access, and modify arrays using methods like slice, concat, indexOf, and spread syntax.*
- o *Object-Oriented Programming: Create objects with literals, constructor functions, and classes. Use this, super, Object.keys, Object.values, and Object.assign for object manipulation.*

React.js

- o *Front-End Development with React: Build dynamic, component-based applications.*
- o *React Components: Create and manage reusable components. Props and JSX: Pass data with props and write JSX syntax.*
- o *React DOM: Understand rendering and virtual DOM.*

COURSE CONTENT (Continued)

- o *State Management: Manage state using hooks and conditional rendering.*

- o *Class vs. Functional Components: Compare and apply both component types.*

Unix Command Line

- o *Basic Bash Commands: Navigate and manipulate files/folders in a Unix/Linux terminal.*

- o *Command-Line Operations: Install software and manage files without a GUI.*

Git, GitHub, and Version Control

- o *Git Workflows: Use commits, branches, merges, and rollbacks.*

- o *GitHub Collaboration: Manage remote repositories, pull requests, and forks.*

- o *Integration with Tools: Use Git with development environments like Xcode.*

Node.js

- o *Back-End Development: Apply data types, objects, methods, and OOP in Node.js.*

- o *Server-Side JavaScript: Run Node on the command line with NPM.*

- o *Node.js Features:*

Event loop, emitters, file system interactions, modules, and native drivers.

Express.js

- o *Express Installation: Set up Express in Node applications.*

- o *Express Servers: Create servers with RESTful routing and middleware. Middleware Usage: Implement middleware for enhanced functionality.*

Application Programming Interfaces (APIs)

- o *API Fundamentals: Understand HTTP, JSON, XML, and server-to-server communication.*

- o *API Interaction: Call and authenticate APIs, read documentation, and handle responses.*

COURSE CONTENT (Continued)

EJS

- o EJS Templating: Use EJS with Node and Express for dynamic content. EJS Features: Run code in templates, pass data, and create layouts/partials.

Database Fundamentals

- o Data Relationships: Design data models with ERM and ORM.
- o Relational vs. Alternative Databases: Compare SQL and NoSQL approaches.

SQL Databases

- o Database Schemas: Create and manage schemas for relational databases.
- o CRUD Operations: Implement Create, Read, Update, Destroy queries. Database Joins: Query data with joins for relational data.

PostgreSQL Databases

- o PostgreSQL Queries: Perform CRUD operations with best practices.
- o Node Integration: Use the pg package for PostgreSQL in Node.js.
- o Relationship Modeling: Implement one-to-many, one-to-one, and many-to-many relationships.

Deployment

- o Hosting Platforms: Deploy static sites with GitHub Pages and server-based apps with Heroku or Mongo Atlas. Deployment Workflows: Manage hosting and deployment processes.

Building RESTful APIs

- o REST Principles: Design APIs with GET, POST, PUT, PATCH, and DELETE endpoints.
- o MongoDB Integration: Use Robo 3T for MongoDB GUI and build APIs from scratch. Express Route Handlers: Implement chained handlers for efficient routing.

Authentication and Security

- o Authentication Needs: Understand the importance of securing user data.
- o Encryption & Hashing: bcrypt for hashing and salting passwords.

COURSE CONTENT (Continued)

- o *Sessions and Cookies: Persist user logins with sessions and cookies.*

- o *Passport and OAuth 2.0: Implement local authentication and logins via Google/Facebook.*

- o *Environment Variables: Secure sensitive keys with environment variables.*

Web3 Basics and Blockchain Development

- o *Web3 and Blockchain: Understand decentralized applications and blockchain concepts.*

- o *Canisters and Motoko: Learn about canisters and Motoko actors on the Internet Computer.*

- o *Deploy CV Website on Blockchain:*

- o *Set up a cycles wallet and obtain free cycles from Dfinity.*

- o *Build and deploy a static CV website to the Internet Computer.*

- o *Monitor cycles balance for canisters.*

- o *React and Blockchain Integration:*

- o *Use a React frontend with a blockchain backend.*

- o *Implement CRUD operations on a canister for note-taking.*

- o *Test apps using the Candid interface.*

- o *Crypto Token Development:*

- o *Create an ERC-20-style token and deploy it on the Internet Computer.*

- o *Build a faucet contract to dispense tokens.*

- o *Use Plug wallet to track and transfer token balances.*

- o *NFT and Marketplace Development:*

- o *Create an ERC-721-style NFT and deploy it to the Internet Computer.*

- o *Build a marketplace website for minting, buying, and selling NFTs.*

- o *Enable NFT ownership transfers via canisters.*

CAPSTONE PROJECT

FAQs

What is the Solardad Web Development Bootcamp, and what are its main goals?

The Solardad Web Development Bootcamp is an 18-month, intensive program designed to transform beginners into job-ready full-stack web developers. Its goals include equipping students with skills to build modern, responsive web applications, fostering collaboration through real-world projects, and preparing students for tech careers with a professional portfolio and interview readiness.

Who is this bootcamp for, and are there any prerequisites?

The bootcamp is for beginners, career changers, aspiring developers, and entrepreneurs with little to no coding experience. Prerequisites include basic computer literacy, a reliable computer, and internet access. Familiarity with HTML or programming concepts is helpful but not required.

I don't have any prior experience or a technical background, can I join the bootcamp?

The course is designed for students of all levels. The most important thing is perseverance and willingness to learn. If you have that then you'll do great!

What teaching methods are used in the bootcamp?

The bootcamp employs hands-on coding sessions, pair programming, peer learning, group projects, weekly assignments, and code reviews. These methods simulate professional workflows, fostering technical skills and teamwork.

Will I get a certificate at the end bootcamp?

Yes, you will get an Internationally recognised Professional certificate at the end of the program, with which you can apply for Jobs and opportunities to advance your career.

Solardard Institute of Information Technology (SIIT).
No.1 Ikuehi Close, GRA, Okene @ Solardad Schools.