

Full-Stack Software Engineering Curriculum (Week-by-Week Breakdown)

Kenya Online College is designed to equip you with the practical skills you need to become a successful full-stack software engineer in a **flexible, project-based learning environment**. This curriculum breakdown provides a detailed roadmap of the knowledge and abilities you'll gain throughout the program.

Module 1: Web Development Foundations & Programming Fundamentals (4 weeks)

- **Week 1:** Introduction to Web Development & Programming Concepts
 - The World Wide Web: Understanding how the internet works and the core technologies behind web applications.
 - Programming Fundamentals: Learning the basic building blocks of programming, including variables, data types, operators, control flow, and functions.
 - Setting Up Your Development Environment: Installing essential tools like code editors, browsers, and version control systems (Git).
- **Week 2:** HTML Fundamentals
 - Structure & Semantics: Diving deep into HTML tags, elements, and attributes to create the basic building blocks of web pages.
 - Building Layouts: Mastering techniques for structuring web pages using HTML sections, divs, and spans.
 - Introduction to CSS: Exploring how CSS styles web pages and understanding basic selectors and properties.
- **Week 3:** Advanced HTML & Introduction to CSS
 - Forms & User Interaction: Implementing forms for user input, data validation, and basic interactivity.
 - Responsive Design Principles: Understanding how to create websites that adapt to different screen sizes and devices.
 - Deep Dive into CSS: Learning essential CSS properties for styling text, layouts, borders, and backgrounds.

<https://kenyaonlinecollege.live/> **Contact:** 0797 532 345

- **Week 4:** Introduction to JavaScript
 - Programming Fundamentals in JavaScript: Exploring variables, data types, operators, control flow, and functions within the JavaScript language.
 - DOM Manipulation: Learning how to interact with the Document Object Model (DOM) to dynamically modify web pages using JavaScript.
 - Introduction to Browser Developer Tools: Understanding how to leverage browser developer tools for debugging and inspection.

Module 2: Front-End Development Deep Dive (6 weeks)

- **Week 5:** Intermediate JavaScript & Event Handling
 - Working with Objects and Arrays: Learning how to structure complex data and manipulate it effectively in JavaScript.
 - Event Handling: Understanding how to respond to user interactions like clicks, scrolls, and form submissions using JavaScript.
- **Week 6 & 7:** Introduction to a Front-End Framework (React or Angular)**
 - Framework Fundamentals: Choosing and learning a popular front-end framework like React or Angular (curriculum will focus on one).
 - Components & State Management: Understanding how to build reusable components and manage data flow within the chosen framework.
- **Week 8 & 9:** Building Interactive UIs with the Framework
 - Advanced Features of the Framework: Learning deeper concepts like routing, forms, and HTTP requests within the chosen framework.
 - Building Dynamic and Interactive User Interfaces: Applying your framework skills to create complex and engaging user interfaces.
- **Week 10 & 11:** Advanced Front-End Techniques & Best Practices
 - Client-Side Routing & Navigation: Implementing navigation systems within your application using the chosen framework.
 - Implementing User Authentication: Understanding basic user authentication principles and integrating them into your application.
 - Front-End Optimization & Best Practices: Learning techniques for optimizing performance, accessibility, and code maintainability.

Module 3: Back-End Development Power-Up (6 weeks)

- **Week 12:** Introduction to Node.js & Express.js
 - Server-Side Development Fundamentals: Understanding the role of the server and the basics of server-side scripting with Node.js.
 - Building APIs with Express.js: Learning how to create APIs (Application Programming Interfaces) using the popular Node.js framework Express.js.
 - Handling HTTP Requests & Responses: Implementing logic to handle different types of HTTP requests (GET, POST, etc.) and send responses.
- **Week 13 & 14:** Building RESTful APIs with Node.js & Express.js
 - RESTful API Design Principles: Understanding the principles of RESTful APIs for creating well-structured and scalable APIs.
 - Building CRUD Operations (Create, Read, Update, Delete) in your APIs: Implementing functionality for creating, retrieving, updating, and deleting data using APIs.
 - Data Validation & Security: Learning essential techniques for validating user input and securing your APIs from vulnerabilities.
- **Week 15 & 16:** Introduction to Databases & Database Management
 - Introduction to Relational Databases: Understanding the concepts of relational databases and how they store data.
 - Working with MySQL or PostgreSQL: Learning basic commands and functionalities for interacting with popular relational databases like MySQL or PostgreSQL.

Full-Stack Software Engineering Curriculum (Week-by-Week Breakdown) (Continued)

Module 4: Database Management Mastery (4 weeks)

- **Week 17 & 18:** Advanced SQL Queries & Database Design
 - Writing Complex SQL Queries: Learning how to write advanced SQL queries to efficiently retrieve and manipulate data.
 - Database Design Principles: Understanding best practices for designing efficient and scalable databases.
 - Data Relationships & Normalization: Learning about data relationships and normalization techniques to optimize database performance.

- **Week 19 & 20:** Integrating Databases with Node.js Applications
 - Connecting to a Database from Node.js: Learning how to establish a connection between your Node.js application and a database.
 - Performing CRUD Operations with Databases: Implementing functionality to create, read, update, and delete data within your application using the database.
 - Introduction to Object-Relational Mappers (ORMs): Exploring the use of ORMs like Sequelize to simplify database interactions in Node.js.

Module 5: Full-Stack Development Projects (8 weeks)

- **Week 21 - 24:** Project 1: Building a Single-Page Application (SPA)
 - Planning & Design: Defining the project scope, user stories, and designing the application's functionalities.
 - Front-End Development with the Chosen Framework: Building the user interface and functionalities of the SPA using the chosen front-end framework (React or Angular).
 - Back-End Development with Node.js & Express.js: Developing the server-side logic and APIs to handle user requests and interact with the database.
 - Integration & Deployment: Connecting the front-end and back-end components, deploying the application to a hosting platform.
- **Week 25 - 28:** Project 2: Building a Full-Stack Web Application with Authentication
 - Project Planning & Feature Definition: Defining a more complex project with functionalities requiring user authentication.
 - Implementing User Authentication & Authorization: Learning and integrating user authentication mechanisms into your application.
 - Building Advanced Features: Implementing additional functionalities based on the chosen project theme.
 - Project Refinement & Polishing: Optimizing performance, user experience, and preparing the project for presentation.

Module 6: Career Launchpad (2 weeks)

- **Week 29:** Crafting Your Developer Brand
 - Building a Winning Resume & Portfolio: Learn how to showcase your skills and experience effectively through a tailored resume and a compelling portfolio website.
 - Interview Preparation & Mock Sessions: Develop your technical interview skills through mock interview sessions and insider tips.
- **Week 30:** Launching Your Tech Career
 - Navigating the Job Market: Learn strategies for effective job search, including networking tips and identifying potential employers.
 - Career Resources & Support: Gain access to our dedicated career services team for ongoing mentorship and guidance as you launch your full-stack development career.

Note: This curriculum breakdown is an overview and may be subject to change based on industry trends and feedback from our partners.