



# “E-Commerce Web Site Using Full stack Development”

Prof. Amisha Naik <sup>1</sup>, Shravani Kolte<sup>2</sup>

<sup>1</sup>Faculty Computer Engineering Vidya Prasarini Sabha's Collage of Engineering and Technology, Lonavala

<sup>2</sup>Student Computer Engineering Vidya Prasarini Sabha's Collage of Engineering and Technology, Lonavala

**ABSTRACT:** The Full Stack Development Internship at CodeAlpha is designed to provide hands-on experience in building complete web applications by working across both front-end and back-end technologies. The internship focuses on developing responsive user interfaces using HTML, CSS, and JavaScript, while implementing robust server-side logic with modern frameworks and databases. During the internship period, practical exposure is given to concepts such as user authentication, database management, API integration, and deployment of web applications. Emphasis is placed on writing clean, efficient code, understanding application architecture, and following industry-standard development practices. This internship enhances problem-solving abilities, strengthens programming fundamentals, and builds a strong foundation in full stack development. By working on real-world projects, interns gain practical skills and confidence required for future roles in software development and the IT industry.

**Keywords:** Full Stack Development, Web Development, Front-End Development, Back-End Development, HTML, CSS, JavaScript, Responsive Design, Database Management, SQL, APIs, User Authentication, CRUD Operations, MVC Architecture, Version Control, Git, Debugging, Deployment, Problem Solving, Software Development Life Cycle (SDLC).

## 1.INTRODUCTION

Full Stack Development is the process of designing, developing, and maintaining both the front-end and back-end of a web application. The front-end, also known as the client side, is the part that users directly interact with and is built using technologies like HTML for structure, CSS for styling, and JavaScript for adding interactivity and dynamic behaviour. It focuses on creating a user-friendly interface and smooth user experience. The back end, or server side, is responsible for handling business logic, data processing, and communication with databases.

It uses programming languages such as Java, Python, PHP, and Node.js, along with databases like MySQL and MongoDB to store and manage data securely. Full Stack Development also involves working with APIs (Application Programming Interfaces), which allow different systems to communicate with each other, as well as version control systems like Git for managing code changes. A Full Stack Developer is skilled in both front-end and back-end technologies and can build complete web applications independently. They understand how the entire system works, from designing user interfaces to managing servers and databases. This makes development faster and more efficient, as one developer can handle multiple tasks. **Objectives**

The main objective of the internship is to gain practical knowledge of full stack web development by working with both front-end and back-end technologies. The internship aims to help interns understand the complete development process of web applications, from design and implementation to testing and deployment.

- The key objectives of the internship were as follows:
- To understand the fundamental concepts of full stack development, covering frontend, backend, and database components of web applications.
- To gain hands-on experience in developing dynamic and responsive web applications using modern development technologies.
- To apply academic programming concepts to practical problem-solving scenarios and real-time project tasks.
- To acquire knowledge of server-side programming, database operations, and integration between client and server.

### 1.1 Literature Review

A literature review is an important part of any research or project that involves studying and analysing existing information related to a particular topic. In the context of Full Stack Development, the literature review focuses on understanding various technologies, tools, frameworks, and methodologies used in building modern web applications. It includes reviewing books, research papers, online articles, and documentation related to front-end technologies such as HTML, CSS, and JavaScript, as well as back-end technologies like Java, Python, Node.js, and databases such as MySQL and MongoDB. The literature review helps in identifying the current trends, best practices, and challenges in full stack development. It also provides knowledge about different development frameworks such as Angular, React, and Django, which are widely used to simplify and speed up the development process. By studying previous work and existing systems, developers can understand what approaches are effective and what limitations need improvement. Overall, the literature review provides a strong theoretical foundation for the project, helps in better decision-making, avoids duplication of work, and ensures that the development process follows modern and efficient techniques.

The literature review also highlights software development methodologies like Agile and Waterfall, version control tools like Git and GitHub, and important security practices such as authentication and data protection. It helps identify current trends, compare different technologies, and understand the strengths and limitations

of existing systems. Overall, it provides a strong foundation for building an efficient and reliable full stack application.

## 2.PROPOSED SYSTEM

The proposed system is a modern web-based application developed using full stack development technologies, which include both front-end (client-side) and backend (server-side) components. The system aims to provide an efficient, scalable, and user-friendly solution to overcome the limitations of traditional systems.KP-based authentication.

### **E-Commerce Platform Development**

The application acts as a centralized platform where users can search, view, and buy products from different categories. It simplifies the shopping process by providing detailed products. information, filtering options, and quick access to services. The system improves customer convenience by enabling 24/7 availability.

Users can apply filters such as price range, category, and ratings to quickly find desired products. The platform ensures smooth navigation and quick response time, which enhances user satisfaction. Additionally, the system offers 24/7 availability, allowing customers to shop anytime and from anywhere, thus improving convenience and accessibility.

### **User Registration and Authentication**

The system provides a secure user account system where new users can register and existing users can log in. Authentication ensures that only authorized users can access their accounts. Password encryption and validation techniques are used to enhance security and protect user data from unauthorized access.

To enhance security, the system uses password encryption techniques, validation methods, secure session handling. These measures protect user data from hacking, unauthorized access, and data breaches. The inclusion of KP-based authentication further strengthens the login process by adding an additional layer of verification.

### **Frontend Design and User Interface**

The frontend is designed using modern technologies like HTML, CSS, JavaScript, Bootstrap to create an attractive and responsive layout. The interface is easy to use, ensuring that users can navigate between pages without confusion. It adapts to different screen sizes, providing a consistent experience across devices.

The design is simple and easy to understand, allowing users to navigate between different pages without confusion. The responsive layout ensures that the application works smoothly on various devices, including desktops, tablets, and smartphones. Proper use of colors, fonts, and layouts makes the interface visually appealing and user-friendly.

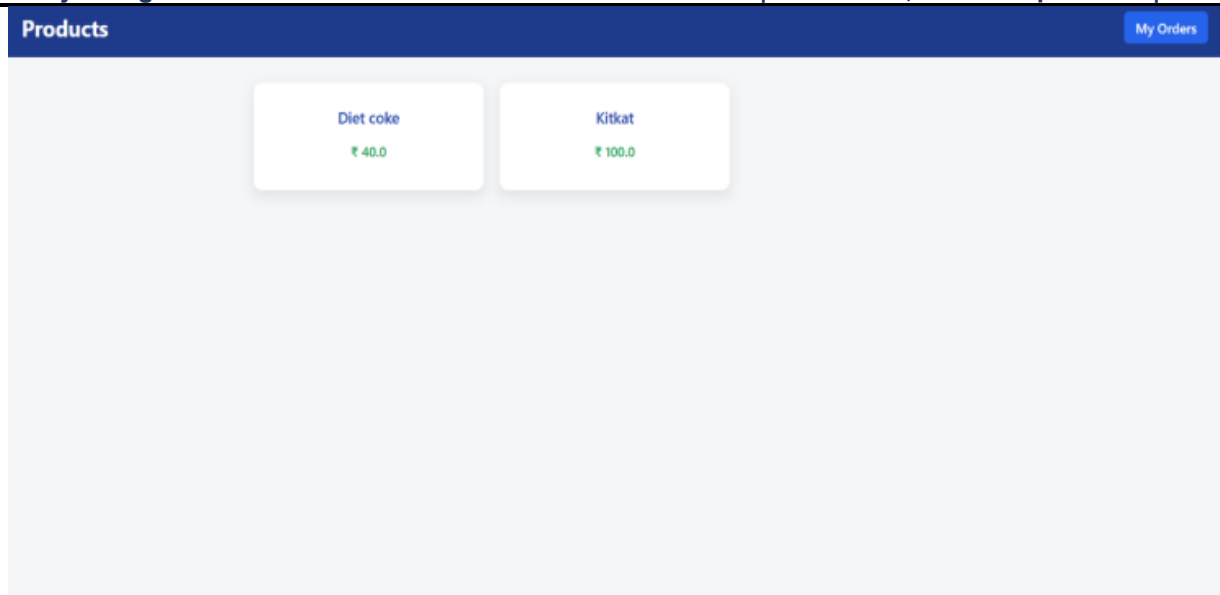


Fig2.1: E-Commerce Frontend View

## Backend Processing and Logic

The backend handles application logic, processes user requests, and manages communication between the frontend and database. APIs are used to ensure efficient and structured data exchange.

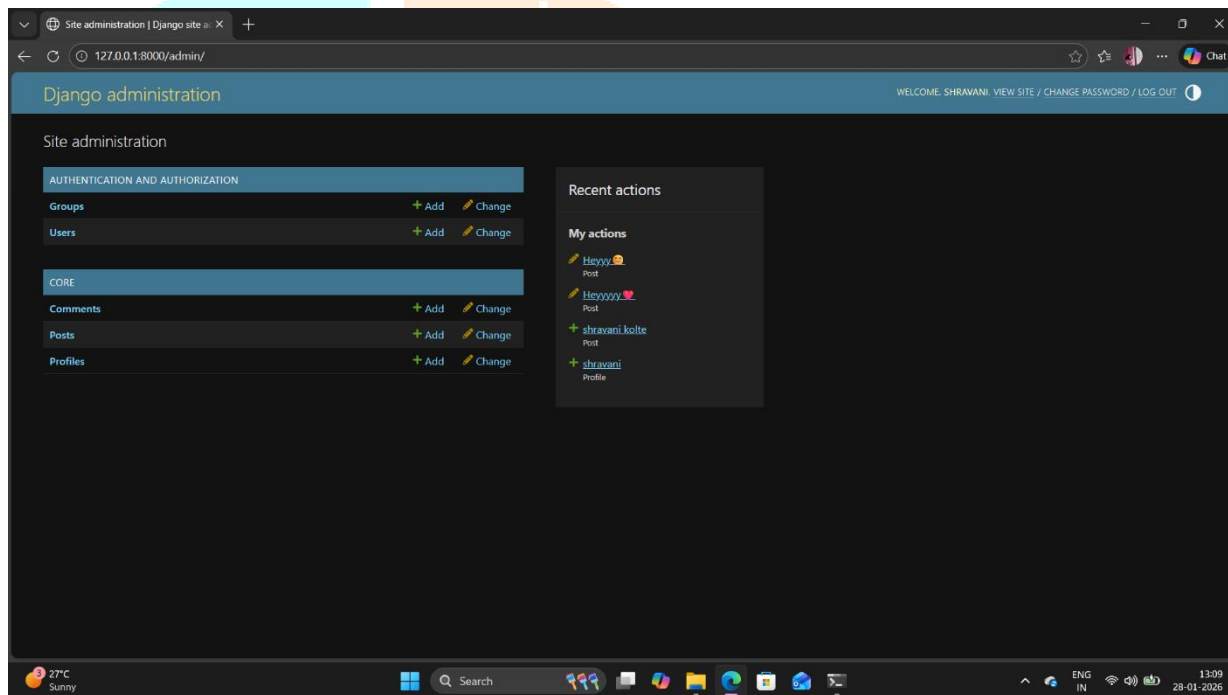


Fig2.2: E-Commerce Backend View

## Database Management System

The system uses a database to store user details, product information, cart items, and order records. Proper data management ensures consistency, security.

## Product and Cart Management

Users can view products, check details, add items to the cart, update quantities, and remove items. This module ensures smooth shopping operations and better user experience.

## Order and Payment Processing

The system allows users to place orders and complete transactions. Payment processing ensures that transactions are handled securely and order details are stored for future reference.

### 3.IMPLEMENTATION AND RESULT

The implementation phase involves converting the proposed system design into a working Full Stack web application. In this phase, the front-end is developed using HTML, CSS, and JavaScript to create an attractive, responsive, and easy-to-use user interface. The back end is developed using technologies such as Node.js, Java, or Python to handle server-side logic, process user requests, and manage application flow. A database like MySQL or MongoDB is used to store and organize data securely. APIs are created to establish communication between the front-end and back-end, ensuring smooth data transfer.

Important modules such as user registration, login authentication, dashboard, and data management (add, update, delete) are implemented. Input validation is applied to ensure correct data entry, and error handling techniques are used to prevent system failures. Security features such as authentication and authorization are included to protect user information. The system is tested using different test cases to identify and fix bugs, ensuring proper functionality and performance. The application is also optimized for speed and designed to be responsive across multiple devices like mobile phones, tablets, and desktops.

Authentication latency was evaluated by measuring the total time required from user login initiation to access approval. Due to the use of off-chain ZKP proof generation and on-chain verification through optimized smart contracts, the system achieved significantly lower latency compared to fully on-chain authentication models. Off-chain computation reduced blockchain congestion and minimized transaction confirmation delays.

### 5.CONCLUSION:

The Full Stack Development internship was a valuable learning experience that provided indepth knowledge of modern web development technologies and practices. Throughout the internship, exposure was gained to front-end development using HTML, CSS, and JavaScript, as well as back-end development involving server-side logic, database management, and application integration. Working on practical tasks and projects helped in understanding how different components of a web application interact to deliver a complete and functional system. In addition to technical learning, the internship offered insight into real-world software development processes such as project planning, coding standards, version control, testing, debugging, and deployment. This experience significantly improved problem-solving skills, logical thinking, and the ability to write clean and efficient code. The internship also enhanced understanding of teamwork, time management, and professional responsibility within a development environment.

Overall, the internship successfully bridged the gap between academic concepts and industry requirements. The knowledge and experience gained during this period have built a strong foundation for future learning and career opportunities in web development and the information technology sector. This internship has contributed positively to professional growth and has prepared the intern to face real-world challenges in the software development field with greater confidence.

## 6.REFERENCES:

- [1] Mozilla Developer Network (MDN Web Docs). Web Technologies Documentation. MDN provides comprehensive and reliable documentation on HTML, CSS, JavaScript, and modern web standards. It is widely used by developers for learning best practices, syntax, and implementation techniques.
- [2] W3Schools. Web Development Tutorials and Examples. W3Schools offers beginner-friendly tutorials, examples, and references for front-end and back-end web technologies, making it useful for understanding basic to advanced concepts.
- [3] JavaScript Official Documentation. Official documentation resources were referred to for understanding JavaScript fundamentals, functions, DOM manipulation, and event handling used in front-end development.
- [4] GitHub. Version Control and Open-Source Resources. GitHub was used to understand version control concepts, project collaboration, and code management. Open-source repositories helped in learning real-world coding practices.
- [5] Online Learning Platforms and Technical Blogs. Various online learning platforms and technical blogs were referred to for gaining conceptual clarity, implementation guidance, and industry insights related to full stack development.
- [6] Company Training Materials and Internship Guidelines. The official training materials, tasks, and documentation provided by the internship organization served as primary references for project development and understanding workflow requirements.
- [7] Software Development Life Cycle (SDLC) Resources. Reference materials related to SDLC were used to understand planning, development, testing, deployment, and maintenance phases in web application development.