**A Midterm Progress Report**

**on**

**E-COMMERCE WEB APPLICATION**

**Submitted in partial fulfilment of the requirements for the award**

**of the degree of**

**BACHELOR OF TECHNOLOGY**

(Computer Science & Engineering)

SUBMITTED BY:

PIYUSH GOYAL

URN-2104153

**July-2025 to Jan-2025**



**GURU NANAK DEV ENGINEERING COLLEGE,**

**LUDHIANA**

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| Fig. No. | Figure Description | Page No. |
| 1 | ER-Diagram | 5 |
| 2 | Data Flow Diagram (DFD) | 6 |
| 3 | User Interface Design Screenshots | 7 |
| 4 | Database Design | 8 |
| 5 | Backend Interface Screenshots | 9 |

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| Table No. | Table Description | Page No. |
| 1 | Category Table | 10 |
| 2 | Products Table | 11 |
| 3 | Orders Table | 12 |
| 4 | User Registration Table | 13 |
| 5 | Cart Table | 14 |
| 6 | Cart Item Table | 15 |

**Chapter 1: Introduction**

1.1 Introduction to Project

In the modern business environment, achieving growth requires more than just traditional methods. As the world rapidly shifts towards digital platforms, businesses must adapt to stay competitive. The introduction of e-commerce has revolutionized how businesses interact with customers, enabling them to reach a global audience while improving user experience, increasing sales, and reducing operational costs. This project focuses on creating an e-commerce platform for an offline business, providing a seamless online experience that extends the business's reach, enhances customer interaction, and boosts profitability. By leveraging the power of the internet, businesses can offer 24/7 availability, a wide range of products, and personalized recommendations, all while simplifying payment and order processes.

1.2 Project Category

The project falls under the category of Application Development in the domain of E-commerce. It involves the design and development of a web-based application that facilitates online transactions between businesses and customers. The system will enable businesses to showcase products, handle customer interactions, and manage transactions in a digital environment, providing convenience to both the business owners and their customers

1.3 Problem Formulation

Businesses, particularly small-scale offline businesses, face numerous challenges in expanding their reach and increasing sales. Limited physical store presence, time constraints of customers, and the inability to serve a larger customer base can restrict business growth. The problem lies in the lack of an effective and accessible platform for businesses to cater to the modern consumer’s preferences—convenience, 24/7 availability, and seamless online transactions. There is a need for an e-commerce solution that can help businesses bridge this gap and cater to the growing demand for online shopping.

1.4 Identification/Recognition of Need

With increasing customer reliance on the internet for shopping, businesses must acknowledge the need for an online platform to remain competitive. The recognition of this need arises from the following challenges:

* Limited operational hours for offline businesses.
* Difficulty in reaching a broader customer base.
* The inconvenience of traditional shopping for time-pressed consumers.
* Increasing demand for convenience, variety, and competitive pricing. To address these challenges, businesses must transition to an e-commerce model that offers easy access, flexible shopping options, and real-time customer interaction.

1.5 Existing System

Currently, many offline businesses operate within traditional brick-and-mortar settings, lacking an online presence. Customers must visit physical locations to purchase products, resulting in limited accessibility. Additionally, the customer experience is limited to in-store interactions, with no option for browsing products online or making purchases after business hours. The existing systems are often inefficient, requiring manual tracking of sales and inventory, and they do not offer the seamless integration of payment, product recommendations, or order tracking that modern consumers expect.

1.6 Objective

The primary objective of this project is to design and develop an e-commerce website that allows businesses to:

* Reach a broader audience by operating 24/7.
* Offer an improved user experience that encourages customers to make purchases.
* Increase customer satisfaction through easy access to products and services.
* shipping services for faster delivery.

**Chapter 2: Requirement Analysis and System Specification**

2.1 Feasibility Study (Technical, Economical, Operational)

The feasibility study for the proposed e-commerce platform is an essential step to assess the viability and practicality of the system from three perspectives: technical, economical, and operational. It aims to determine if the project is achievable, sustainable, and aligned with the goals of the business.

1. Technical Feasibility

Technical feasibility focuses on the capability of the technology, hardware, and software to support the proposed system and meet the business requirements.

**Platform and Technology Stack:** The e-commerce website will rely on modern web technologies such as

* Frontend: HTML5, CSS3, JavaScript (with frameworks like React, Vue.js, or Angular) for a responsive and dynamic user interface.
* Backend: A reliable server-side framework such as Node.js, Django (Python), or Laravel (PHP) will handle business logic, user management, and payment transactions.
* Database: SQL-based databases like MySQL or PostgreSQL will store customer, product, and order data, while NoSQL options (like MongoDB) may be used for more scalable data models.
* Payment Gateway Integration: The platform will integrate third-party payment systems such as PayPal, Stripe, or Razorpay to securely process transactions.
* Security: The system will use SSL encryption, two-factor authentication, and regular security audits to ensure the protection of sensitive customer data, such as payment details and personal information.

Infrastructure and Hosting:

* The platform will be hosted on a cloud service provider (AWS, Google Cloud, or Microsoft Azure) to ensure high availability, reliability, and scalability.
* Content Delivery Network (CDN) will be used for faster content delivery to global customers.
* Server Load Balancing and Auto-Scaling will be implemented to handle high traffic volumes, particularly during peak sales periods.

Scalability and Maintenance:

* The system will be designed with scalability in mind to accommodate future growth, whether through additional product listings, user accounts, or higher transaction volumes.
* The architecture will support modular updates and maintenance to ensure continuous improvement without disruption to the user experience.

Technical Constraints:

* Efficient coding practices will be followed to ensure that the platform performs well even on lower-end devices.
* The website's database size will be managed carefully to optimize performance and minimize latency.

2. Economical Feasibility

Economic feasibility examines whether the project is financially viable and if the costs involved can be justified by the expected returns.

Development Costs:

* The initial development will require costs for hiring skilled developers, designers, and other professionals, as well as purchasing any third-party tools or software required.
* Cloud Hosting and Infrastructure: There will be recurring costs for cloud hosting, databases, and other services (e.g., email marketing tools, analytics platforms).
* Payment Gateway Fees: Payment gateways charge a transaction fee for each sale, which will affect profit margins.

Operational Costs:

* Ongoing expenses will include server maintenance, updates, security patches, customer support, and marketing efforts (advertising, promotions, and SEO).
* Marketing and Customer Acquisition Costs: Strategies like online ads, social media campaigns, and email marketing will incur additional costs. However, these efforts are expected to bring in consistent traffic and sales.

Revenue Generation:

* The platform’s primary source of revenue will come from product sales, including both direct sales and commissions for third-party sellers.
* Additional revenue streams can be generated through advertising, affiliate marketing, and premium services (e.g., subscriptions for premium memberships or exclusive products).
* Transactional Fees: The platform may also charge sellers a commission on each sale made through the site.

Return on Investment (ROI):

* The expected ROI is promising, given the growing e-commerce market and the business's ability to scale. By providing a seamless, user-friendly shopping experience and utilizing targeted marketing efforts, the platform can attract a large number of customers, increasing both revenue and brand visibility.

Profitability:

* With efficient operations, reduced overhead costs compared to physical stores, and an expanding customer base, the platform has the potential to reach profitability within the first year of operations. The business model is based on recurring revenue, making it sustainable in the long term.

3. Operational Feasibility

Operational feasibility assesses the ability of the organization to manage and run the proposed system, focusing on day-to-day operations, customer service, and vendor management.

User Acceptance:

* The system will be easy to use, and the general public is already accustomed to shopping online. This makes user adoption highly feasible.
* With a well-designed user interface (UI) and user experience (UX), the platform will cater to both novice and experienced online shoppers. A smooth, intuitive experience is essential to attracting and retaining customers.

Vendor and Product Management:

* The platform will allow vendors to list products, manage inventory, and fulfill orders. Integration of vendor-specific dashboards and real-time inventory tracking will ensure smooth operation for both customers and vendors.
* Customer Interaction: Customers will be able to interact with vendors via the chat system, ask questions, and clarify details before making a purchase.

Customer Support:

* The platform will integrate a robust customer support system, including live chat, email support, and frequently asked questions (FAQ) sections.
* Automated systems such as chatbots can help address basic customer queries, while dedicated customer support agents will handle more complex issues.

Order Fulfillment:

* The platform will support multi-vendor sales, so it must integrate with third-party logistics (3PL) providers for shipping and tracking.
* Clear communication regarding order status, tracking numbers, and estimated delivery times will be provided to customers.

Compliance and Legal Considerations:

* The platform will comply with data protection laws such as GDPR (for European users) or CCPA (for California residents), ensuring customers' data privacy and rights.
* Terms and Conditions will be clearly outlined on the website, along with return/refund policies and shipping details.

Training and Support for Employees:

* Employees will need training in managing product listings, processing orders, handling customer complaints, and maintaining security standards.
* Ongoing technical and customer support training will ensure the team is equipped to handle any operational issues effectively.

Maintenance and Upkeep:

* Routine maintenance will ensure that the platform remains up to date with new technologies and security protocols.
* Updates to the product catalog, customer data, and system security will be performed regularly to ensure the platform runs smoothly and securely.

2.2 Software Requirement Specification Document

The Software Requirement Specification (SRS) document outlines the detailed requirements for the proposed e-commerce platform, serving as a guide for developers, stakeholders, and future validation of the software. It includes various categories of requirements such as data, functional, performance, dependability, maintainability, security, and look and feel.

Data Requirements:

* The system must handle a wide range of data related to customers, products, transactions, and vendors.
* Data Types: User data (personal details, login credentials, purchase history), product data (names, descriptions, prices, stock levels), transaction data (payment, order tracking), and vendor data.

**Chapter 3: System Design**

3.1 Design Approach (Function-Oriented or Object-Oriented)

For the proposed e-commerce website, the Object-Oriented Design (OOD) approach will be used. This approach is ideal for managing the complexity of an online shopping platform that involves a wide range of entities, such as users, products, transactions, and payment systems. The key principles of object-oriented design, such as encapsulation, inheritance, polymorphism, and abstraction, will allow for the creation of modular, scalable, and maintainable components. Here’s a breakdown of why Object-Oriented Design is the preferred approach for this system:

* **Modularity:** The system will be divided into distinct classes and objects representing real-world entities such as Customers, Products, ShoppingCart, Orders, etc. Each object will have its own set of attributes and methods, making the code easier to manage and extend.
* **Scalability:** As the business grows, new functionalities (such as additional payment methods or new categories of products) can be added by extending existing objects or creating new ones, ensuring the system remains scalable.
* **Maintainability:** Object-oriented systems are easier to maintain and modify. Changes can be made to individual objects or classes without affecting the entire system. For example, the checkout process can be updated independently of the product catalog.
* **Reusability:** With the OOD approach, common features or modules can be reused across different parts of the application. For instance, customer authentication, product management, and order processing modules can be reused or extended.

In contrast, function-oriented design focuses on procedures or functions that operate on data, which can make the system harder to maintain as it grows, especially when handling complex, real-world entities. Thus, Object-Oriented Design is the most appropriate choice for developing a robust, scalable, and maintainable e-commerce platform.

3.2 Detail Design

Flow Chart / Block Diagrams

Flow charts or block diagrams visually represent the step-by-step flow of processes within the system. For an e-commerce website, a flow chart can outline the journey of a user from accessing the website, browsing products, adding them to the cart, proceeding to checkout, and completing the payment. These diagrams simplify complex processes into understandable sequences using standardized symbols like rectangles, diamonds, and arrows.

UML Diagrams

Unified Modeling Language (UML) diagrams are used to visualize, specify, and document the structure and behavior of a system. For an e-commerce project, UML diagrams help model the interactions between different components, such as users, databases, and servers, ensuring a clear understanding of system workflows and relationships.

Data Flow Diagrams (DFDs)

DFDs depict how data flows within the system, focusing on inputs, processes, storage, and outputs. For the e-commerce website, a DFD can represent processes like user login, product search, order placement, and payment processing, ensuring clarity on how data is managed across modules.

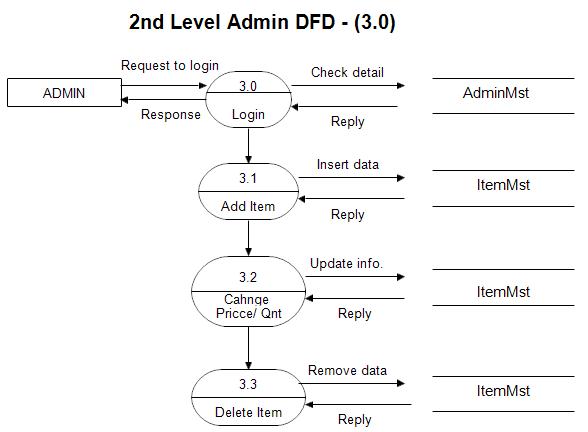
Level-0 DFD for Online shopping website project



‎Level-1 Admin Side DFD

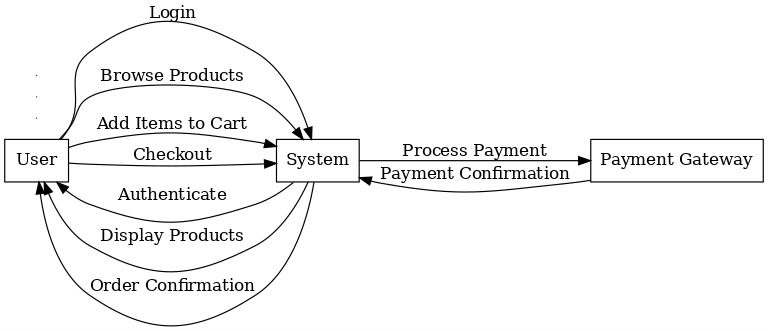
****

‎2nd Level – Admin Side DFD

****

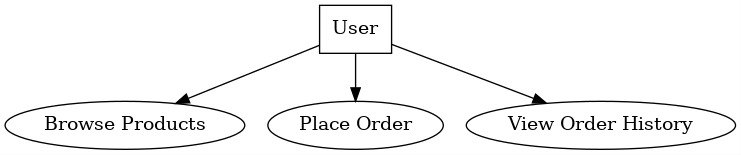
‎Sequence Diagrams

Sequence diagrams capture the order of interactions between system components over time. For the e-commerce project, they can detail interactions between a customer, the website, and the database during actions like adding a product to the cart, initiating payment, and confirming the order.



Use Case

Use case diagrams illustrate the functionalities of the system from a user perspective. In an e-commerce website, they can represent use cases like "Browse Products," "Add to Cart," "Make Payment," and "Track Order," showing the actors (users/admins) involved in each activity and their interactions.



Activity Diagrams

Activity diagrams show the dynamic workflows of a system. For an e-commerce website, they can represent processes like the order placement workflow, where users select products, add them to the cart, input delivery details, make payments, and receive confirmation.

Activity Flow associated with the system administrator

In this the process Starts from the beginning then we login with User Login credentials if it is successful we will Browse for the products and Add the items to the cart then the items will be sent to checkout and after doing the payment our order confirmation is done and it’s done at the End.

If the process starts and User Login credentials aren’t satisfied then we will try again.

If trying again gives same result then we will use correct login credentials for the user.

State Diagrams

State diagrams represent the various states an object in the system can have and how transitions occur between them. For instance, in an e-commerce website, the order object can transition between states such as "Pending," "Processing," "Shipped," and "Delivered" based on user and system actions.

Communication Diagrams

Communication diagrams show the interaction between objects or components of the system in terms of message exchange. For the e-commerce project, they can represent the communication between the user interface, the backend server, and the database during tasks like product retrieval or payment processing.

Deployment Diagrams

Deployment diagrams illustrate the physical deployment of software components onto hardware. For an e-commerce website hosted on AWS, the diagram would show components like the web server, application server, database server, and their deployment on AWS EC2 instances, with connections to services like S3 and Route 53.

Database Design

Database design focuses on creating structured schemas to efficiently store and retrieve data. For the e-commerce project, this includes designing tables for users, products, orders, and transactions, along with defining relationships such as one-to-many (users to orders) or many-to-many (products in orders) to ensure smooth data operations.

**Category Table**

|  |  |  |
| --- | --- | --- |
| Field | Type | Constraint |
| ID | Number(5) | Primary Key |
| Name | VarVarchar (50) | Not Null |
| Description | Text Field |  |
| Image | Image |  |

E-R Diagram

An Entity-Relationship (ER) diagram is a critical tool in database design, providing a graphical representation of the system's data and the relationships between entities. It is an essential step in creating a robust database structure, ensuring all system requirements are captured efficiently. ER modeling was introduced by Peter Chen in 1976 and has evolved significantly, with modern approaches incorporating advanced concepts like inheritance, generalization-specialization, and enhanced modeling tools.

In the context of an e-commerce website, an ER diagram is invaluable for mapping entities such as Users, Products, Orders, and Payments. Each entity represents a real-world object or concept, and the relationships between them—such as "User places Order" or "Order contains Product"—are depicted using connectors. Attributes, such as a user's email or a product's price, further enrich the model by adding details to each entity.

The latest advancements in ER modeling integrate support for NoSQL databases and hybrid schemas to cater to the growing demand for flexible and scalable systems. Tools like MySQL Workbench, Lucidchart, and ER/Studio now provide enhanced functionalities such as real-time collaboration, auto-generation of schema code, and support for cloud-based deployments.

For an e-commerce system, the ER diagram typically includes:

**User:** Attributes like User\_ID, Name, Email, and Password.

**Product:** Attributes such as Product\_ID, Name, Description, Price, and Stock.

**Order:** Attributes including Order\_ID, Order\_Date, and Total\_Amount.

**Payment:** Attributes like Payment\_ID, Payment\_Mode, and Transaction\_Status.

The relationships among these entities are designed to ensure data consistency and avoid redundancy. For instance, a one-to-many relationship exists between Users and Orders, as a single user can place multiple orders. Similarly, a many-to-many relationship connects Orders and Products since each order can include multiple products, and a product can appear in multiple orders. The ER diagram is not merely a theoretical design—it forms the backbone of database development. It ensures that the system aligns with the intended functionality, minimizes anomalies, and facilitates future scalability. With the advent of cloud-based databases and distributed systems, modern ER modeling now also considers factors like data partitioning, replication, and optimization for high-traffic environments. This approach ensures that the database design meets both current needs and potential future expansions.

3.3 User Interface Design

The user interface (UI) of the e-commerce website is designed to provide a smooth, intuitive, and user-friendly experience for both customers and administrators. It ensures easy navigation, efficient browsing, and a seamless transaction process. Below are the main features of the UI design for different sections:

**For Customers:**

Home Page:

* Navigation bar with links to categories, search bar, user profile, cart, and help section.
* Featured products and categories with high-quality images.
* Banners showcasing promotions, offers, and discounts.
* Easy access to new arrivals, popular products, and limited-time deals.

Product Page:

* Product images with zoom-in functionality for better viewing.
* Product name, price, and a detailed description.
* Product specifications, including size, color, and other important attributes.
* Stock availability and shipping details.
* “Add to Cart” and “Buy Now” buttons.
* Product reviews and ratings from other customers.
* Related products and suggestions for further browsing.

Shopping Cart:

* List of added products with their images, names, quantities, and prices.
* Options to modify cart items (change quantity, remove items).
* Display of total cost, taxes, and shipping fees.
* Checkout button to proceed with the order.
* Option to save the cart for later or continue shopping.

Checkout Page:

* Form for customers to enter billing and shipping information.
* Summary of the products in the cart, including quantity and pricing details.
* Multiple payment options, including credit/debit cards, cash on delivery, or online payment gateways.
* Option to apply discount codes or promotional offers.
* Order confirmation and final review before completing the purchase.

Order Confirmation:

* Order number and summary of purchased items.
* Estimated delivery date and shipping method.
* A thank-you message with instructions for tracking the order.
* Option to view order history or continue shopping.
* User-friendly login page to access personal accounts.

**For Admins:**

Admin Dashboard:

* Overview of site activities, including total sales, number of orders, and active users.
* Quick links to key functions like product management, order management, and customer support.
* System notifications for low stock, pending orders, or customer inquiries.

Product Management:

* Interface to add, edit, or delete products.
* Manage product details such as name, description, price, images, and specifications.
* Set stock levels and manage inventory.
* Manage categories and product attributes.

Order Management:

* View and manage customer orders, including order status (pending, shipped, delivered).
* Option to cancel, update, or refund orders.
* Track payment statuses and shipping details.

Customer Management:

* View customer profiles, including their contact details, order history, and preferences.
* Manage customer support tickets and inquiries.
* Send promotional offers or updates to customers.

Report Generation:

* Generate sales reports, including total sales, revenue, and customer trends.
* Delivery and shipping reports, including pending deliveries, delivery statuses, and courier details.
* Customer behavior reports, including frequent purchases, order frequency, and product preferences.

3.4 Methodology

The methodology followed in the development of the e-commerce website is Agile Development, particularly using the Scrum Framework. Agile is chosen due to its iterative nature, allowing for flexibility and continuous client feedback integration, ensuring the project stays on track and adapts to changing requirements.

**Development Phases:**

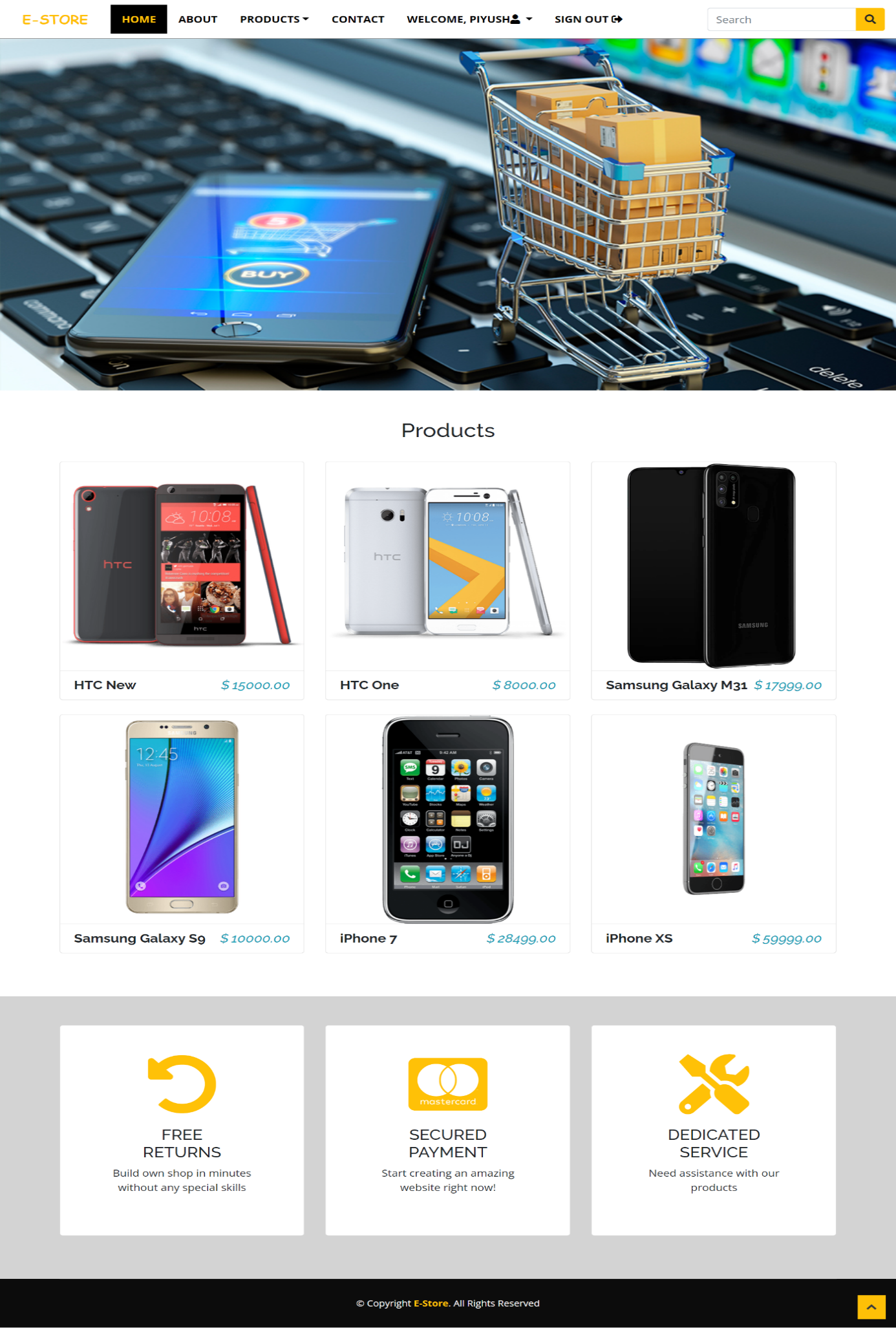
Requirement Gathering and Analysis:

* Initial meetings with stakeholders to gather and analyze requirements.
* Detailed discussion on features, functionality, and design expectations.
* Creation of user stories and use cases for the development process.

Design Phase:

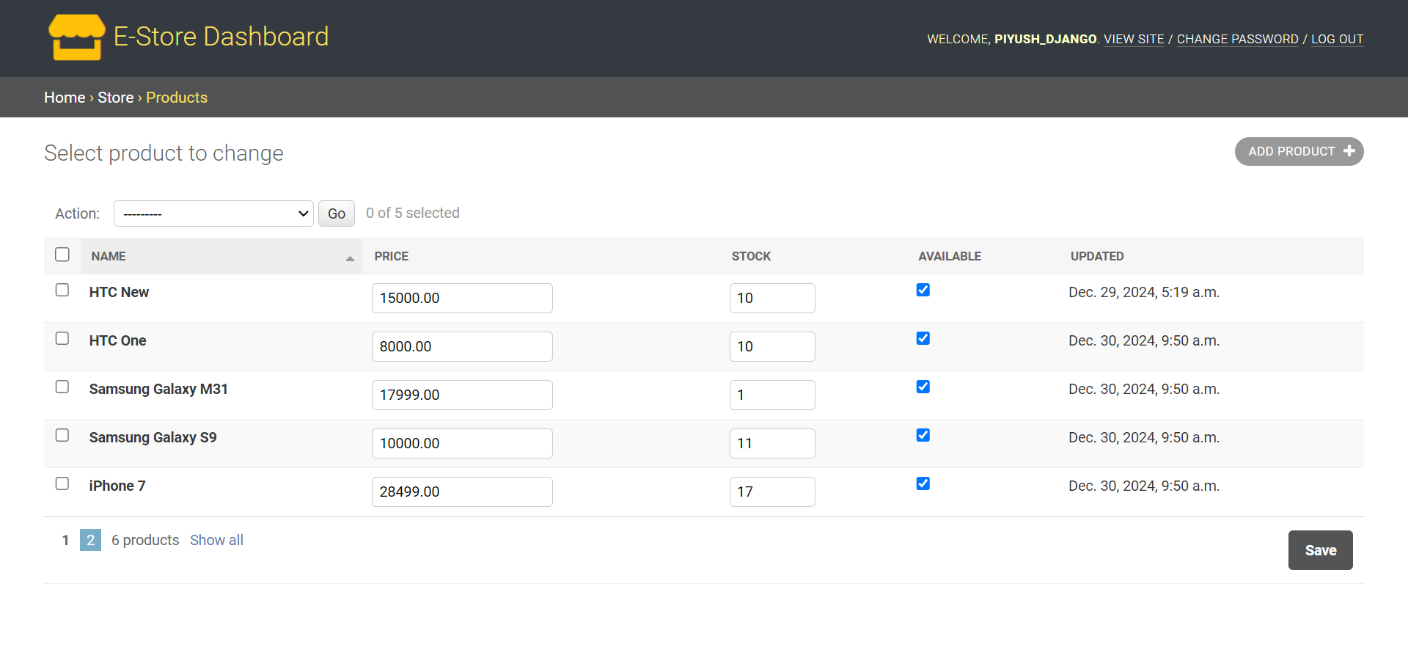
* Creation of wireframes, UI mockups, and database schema.
* Design of the system architecture, considering scalability and performance.
* Selection of development tools and frameworks.

**HOME PAGE**

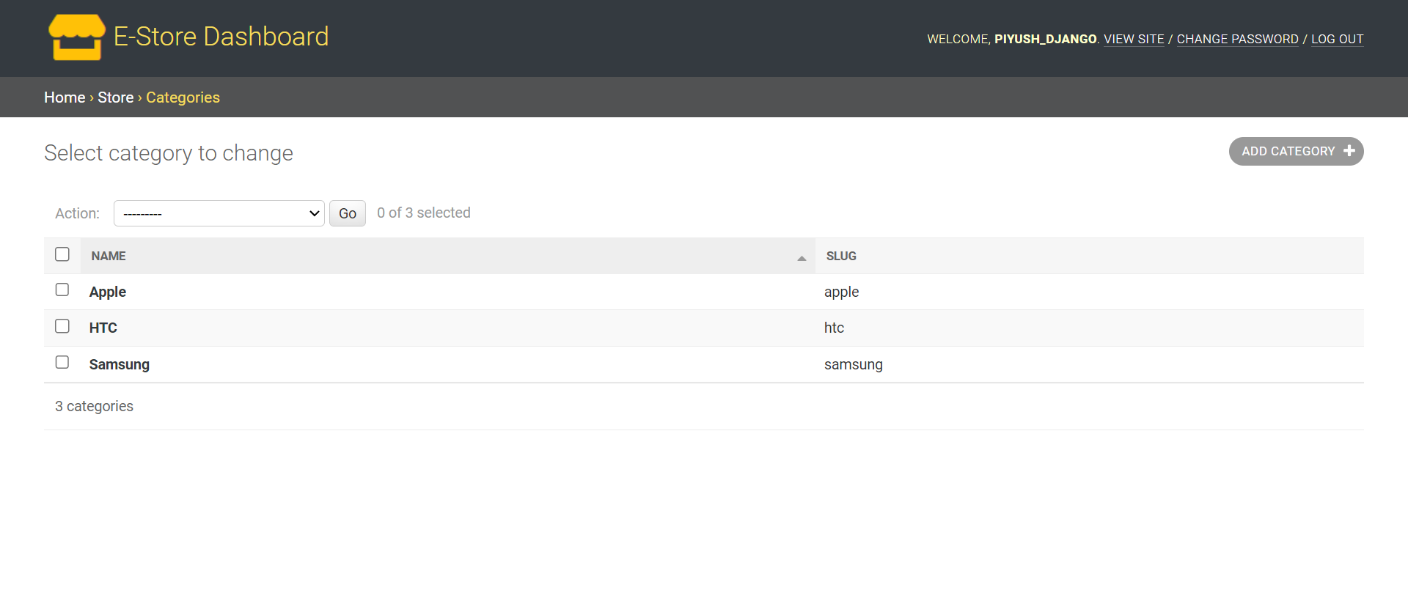


**BACKEND INTERFACE**

**ADMIN PRODUCTS TABLE**



**ADMIN CATEGORY TABLE**



**Chapter 4: Implementation and Testing**

4.1 Introduction to Languages, IDE’s, Tools, and Technologies Used for Project Work

For the development of the e-commerce website, the following technologies and tools are utilized:

* Languages:
* HTML5: For structuring the web pages and creating the layout of the site.
* CSS3: For styling and responsive design, ensuring the website is aesthetically pleasing and works across various devices.
* JavaScript: For client-side scripting to handle interactive elements such as the shopping cart, user login, product search, and more.
* PHP: For server-side scripting to handle user authentication, product management, and transactions.
* SQL: For database management and handling customer, product, and order data.
* IDE (Integrated Development Environments):
* Visual Studio Code: A lightweight but powerful code editor used for writing HTML, CSS, JavaScript, PHP, and SQL. It has a large selection of extensions to aid in efficient coding.
* PHPStorm: Primarily used for PHP development, providing an advanced feature set like debugging, code completion, and integration with databases.
* Database Management System:
* MySQL: Used for storing and managing data such as user details, product information, and transaction records.
* Web Server:
* Apache: A popular open-source web server used to serve the website to users and run PHP scripts.
* Version Control:
* Git: Used for version control, ensuring that changes to the codebase are tracked and managed efficiently.
* GitHub: Used for hosting the code repository and collaborating with the development team.

Other Tools:

* Postman: Used for testing the APIs, ensuring correct responses are returned.

Cloud Hosting with Amazon AWS

The e-commerce website is hosted on Amazon Web Services (AWS) to ensure scalability, reliability, and high performance. The following AWS services are utilized:

Amazon EC2 (Elastic Compute Cloud):

* Provides virtual servers (instances) for hosting the website.
* Ensures flexibility in scaling the infrastructure up or down based on the website traffic.
* Offers robust performance and control over server configurations.

Amazon S3 (Simple Storage Service):

* Used for storing and managing static files such as product images, videos, and other media files.
* Provides high durability and availability for large amounts of data.

Amazon RDS (Relational Database Service):

* Hosts the MySQL database in a managed environment.
* Handles backups, patching, and scaling for optimal database performance.

AWS CloudFront:

* A Content Delivery Network (CDN) that speeds up the delivery of static and dynamic content by caching data at edge locations worldwide.
* Ensures a low-latency experience for users across the globe.

AWS IAM (Identity and Access Management):

* Ensures secure access management for AWS resources.
* Defines user roles and permissions for safe collaboration among developers.

AWS Elastic Load Balancer (ELB):

* Distributes incoming traffic across multiple EC2 instances, ensuring high availability and fault tolerance.

AWS Auto Scaling:

* Automatically adjusts the number of EC2 instances to handle varying traffic levels.
* By integrating AWS services, the e-commerce website benefits from a highly scalable, secure, and cost-effective hosting environment. This ensures uninterrupted performance, even during peak traffic, while keeping data secure and accessible.

The e-commerce website is hosted on Amazon Web Services (AWS), utilizing the following AWS services:

**Amazon EC2 (Elastic Compute Cloud):** Provides scalable computing capacity to run the website on virtual servers. EC2 allows for easy scaling based on traffic demand.

**Amazon S3 (Simple Storage Service):** Used for storing static assets such as images, product details, and other media files. S3 provides high availability and durability for these resources.

The website can be accessed using its public IP address, offering direct access to the application without the need for a domain name. This is useful for internal testing, development, or situations where a custom domain may not be set up yet.

4.2 Algorithm/Pseudocode Used

Below is a sample pseudocode for implementing the user login system for the e-commerce website.

START

INPUT username, password

IF username exists in database THEN

FETCH stored\_password from database

IF password matches stored\_password THEN

DISPLAY "Login Successful"

STORE session for user

REDIRECT to home page

ELSE

DISPLAY "Invalid Password"

END IF

ELSE

DISPLAY "Username not found"

END IF

END

4.3 Testing Techniques: In Context of Project Work

Testing is a crucial phase of the project to ensure that the e-commerce system is functioning correctly. The following testing techniques were employed:

**Unit Testing:**

* Each individual component (e.g., login system, product search, checkout process) was tested in isolation to ensure that it behaves as expected. Unit tests were written using PHPUnit for PHP.

**Integration Testing:**

* Integration testing was performed to check if different modules of the system (such as login, product search, and cart management) work together seamlessly. This ensures that the data flows correctly between modules.

**Functional Testing:**

* Functional testing was done to verify that the system meets the business requirements, including verifying user registration, login, product search, order placement, and payment transactions.

**Usability Testing:**

* A group of users tested the e-commerce website for user-friendliness, navigation, and overall experience. Feedback was collected to improve the website's design and interaction flow.

4.4 Test Cases Designed for the Project Work

The following test cases were designed to validate various functionalities of the e-commerce website:

* Test Case 1: User Registration

**Objective:** Ensure that a new user can successfully register on the site.

**Steps:**

1. Navigate to the registration page.
2. Enter valid username, password, email, and other required details.
3. Click the “Register” button.
4. Expected Result: User is successfully registered, and a confirmation email is sent.

* Test Case 2: User Login

**Objective:** Ensure that an existing user can successfully log in.

**Steps:**

1. Navigate to the login page.
2. Enter valid username and password.
3. Click the “Login” button.
4. Expected Result: User is successfully logged in and redirected to the homepage.

* Test Case 3: Product Search

**Objective:** Ensure the search functionality returns relevant products.

**Steps:**

1. Navigate to the search bar.
2. Enter a product name or keyword.
3. Click the search icon.
4. Expected Result: Relevant products related to the search term are displayed.

* Test Case 4: Add to Cart

**Objective:** Ensure that users can add products to the shopping cart.

**Steps:**

1. Browse the product catalog.
2. Select a product and click the "Add to Cart" button.
3. Expected Result: Product is added to the shopping cart, and the cart icon is updated with the correct count.

* Test Case 5: Checkout and Payment

**Objective:** Ensure that the checkout process and payment gateway work correctly.

**Steps:**

1. Go to the shopping cart and review the selected products.
2. Click "Proceed to Checkout."
3. Enter shipping and payment information.
4. Confirm the order.
5. Expected Result: User is redirected to a confirmation page with order details.

* Test Case 6: Order History

**Objective:** Ensure that the user can view their order history.

**Steps:**

1. Log in as a registered user.
2. Navigate to the “Order History” section.
3. Expected Result: The list of past orders is displayed with details like product name, quantity, and status.

* Test Case 7: Security Check

**Objective:** Ensure that the login system is secure and does not allow SQL injection.

**Steps:**

1. Enter a SQL injection string in the username or password field (e.g., ' OR 1=1;--).
2. Click the “Login” button.
3. Expected Result: The system should reject the input and show an error message indicating invalid login.

This chapter ensures that the e-commerce website is developed and tested thoroughly to meet the specified requirements and deliver a functional, secure, and reliable system.

**Chapter 5: Results and Discussions**

5.1 User Interface Representation (of Respective Project)

The e-commerce website's user interface has been meticulously designed to provide a seamless and engaging experience for users. The UI includes modules for browsing products, managing accounts, and completing purchases. Each module is optimized for clarity, responsiveness, and ease of use, ensuring users can efficiently navigate and complete their tasks.

5.1.1 Brief Description of Various Modules of the System

1. **Home Page**

The home page serves as the gateway to the e-commerce platform. It includes prominently displayed banners showcasing featured products, trending items, and ongoing offers or discounts. Users can browse highlighted categories, including electronics, clothing, and accessories, directly from the homepage. A search bar is available at the top to allow users to quickly find products by typing keywords. The home page also displays user account links for logging in or registering and quick links to the shopping cart and customer service.

1. **Login/Registration Module**

The login and registration module enables users to authenticate their accounts. New users can register by providing details like username, email, and password. Registered users can log in with their credentials. The module includes a password recovery option to retrieve forgotten credentials. A confirmation email is sent upon successful registration to validate user accounts. The login page redirects authenticated users to their account dashboard.

1. **Product Catalog**

The product catalog allows users to browse through a variety of products organized into categories and subcategories. Each product is displayed with its image, name, price, and rating. Users can filter products based on criteria like price range, brand, and rating or sort them by popularity, relevance, or price (ascending/descending). Clicking on a product redirects users to the product details page, which provides a comprehensive description, specifications, and customer reviews.

1. **Product Details Page**

This module provides an in-depth view of individual products. It includes a high-resolution image gallery, detailed product specifications, pricing information, and customer reviews. Users can view ratings and leave reviews for products they have purchased. An "Add to Cart" button allows users to save the product for purchase.

1. **Shopping Cart**

The shopping cart module displays a list of products that the user has added for purchase. Each item in the cart includes the product name, price, quantity, and subtotal. Users can update quantities or remove items from the cart. The cart also displays the total cost, including applicable taxes and shipping fees. From here, users can proceed to checkout or continue shopping.

References/Bibliography:

1. **Framework and Languages**:

* Django Web Framework Documentation: <https://docs.djangoproject.com>
* PostgreSQL Database Official Documentation: <https://www.postgresql.org/docs>
* Bootstrap Framework: <https://getbootstrap.com>

1. **Testing Techniques**:

* Python unittest Library: <https://docs.python.org/3/library/unittest.html>
* Selenium Web Testing: <https://www.selenium.dev>

1. **Development Tools**:

* Visual Studio Code Documentation: <https://code.visualstudio.com/docs>
* HTML, CSS, and JavaScript Specifications: <https://developer.mozilla.org>

1. **Design Principles**:

* Modular Programming: IEEE Modular Programming Standards
* Object-Oriented Design: Grady Booch's *Object-Oriented Analysis and Design with Applications*

1. **Additional Resources**:

* Official Bootstrap Icons and Templates: https://icons.getbootstrap.com
* W3Schools Tutorials on Web Development: <https://www.w3schools.com>

1. **Testing Frameworks**:

* Django Testing Framework: <https://docs.djangoproject.com/en/3.2/topics/testing>
* Selenium Testing Suite: <https://selenium-python.readthedocs.io>

‎1.‎ <https://www.wikipedia.org/>‎  
‎2.‎ <https://www.w3schools.com/>‎  
‎3.‎ <https://getbootstrap.com/>‎  
‎4.‎ <https://www.djangoproject.com/>‎  
‎5.‎ <https://www.python.org/>‎  
‎6.‎ <https://www.postgresql.org/>‎  
‎7.‎ <https://fontawesome.com/>‎  
‎8.‎ <https://michalsnik.github.io/aos/>‎  
‎9.‎ <https://stripe.com/en-in>  
‎10.‎ <https://www.quora.com/>

Appendix A: Development Environment

The development environment outlines the tools, platforms, and configurations used during the development of the E-commerce Website project. This ensures consistency, collaboration, and efficient development practices. Below are the details of the development environment.

Hardware Configuration

* Processor: Intel Celeron 3205U 1.50GHz
* RAM: 8GB
* Hard Disk: 1TB
* Printer: Laser Printer (for report printing and documentation)
* Pen Drive: 5GB (for portability of project files)

Software Configuration

* Operating System: Windows 10
* Screen Resolution: 1024×768
* Editors: Visual Studio Code (for writing and debugging code), MS Word (for documentation and reports)
* Graphic Tools: Paint (for creating or editing basic visuals and mockups)

Languages and Frameworks

* HTML5: For structuring web pages
* CSS3: For styling and responsive design
* Bootstrap: For creating a modern, responsive UI
* JavaScript: For adding interactivity to the website
* Python: For backend development
* Django: As the primary web framework
* PostgreSQL: For database management

Cloud Environment-AWS Services:

* EC2: For hosting the website.