SHOPPING MART- A DJANGO E-COMMERCE WEBSITE

Amishi Agrawal[1], Nikita Chouhan[2]

[1,2]Dept. of Computer Science & Engg. Tha.ShivKumar Singh Memorial Engg.

College, Burhanpur (M.P.), India.

ABSTRACT

This project is an e-commerce website developed using Django, a high-level Python web framework. The website allows users to browse products, add items to their shopping cart, search items and checkout. Unregistered users can browse the products, search them and add them to cart but for order checkout the user have to register themselves. Registered users can also manage their profiles and view their order history. The back-end is designed to handle data, with models for products, orders, contact and customers. The front-end is built using HTML, CSS, JavaScript and Bootstrap with templates that are rendered dynamically using Django's templating engine. Django's templating engine is an important component of the Django web framework that makes it easy for us to build dynamic, data-driven web applications. The website is fully responsive and works well on desktop and mobile devices. Overall, this project demonstrates how Django can be used to build a scalable, feature-rich e-commerce platform.

1. INTRODUCTION

E-commerce websites are becoming increasingly popular, with more businesses turning to online sales to reach a wider audience. The covid times have taught us the importance of online business now more than ever. Python and Django are two popular technologies that can be used to build these websites. In this study, we present an analysis of an e-commerce website built using Python and Django. The website allows users to browse products, search by category or keyword, add items to a cart, and check out securely. It also includes a contact form for users to get in touch with any questions or concerns that they might have. At the heart of this website is the aim to deliver a smooth shopping experience to users and relieve them of any inconvenience.

2. METHODOLOGY

* **Choosing a SDLC model** –Software development life cycle model helps developers plan the project from beginning to very end. It provides a clear view of what steps or stages our project should go through so that we can get the desired product at the end. It breaks the entire development project into phases, each phase has significant impact on the final product. Instead of doing everything at once, it breaks development into phases in a particular sequence. SDLC has various models, we can choose one that best suits our project.For our project we chose Incremental Model, as ecommerce website is a very big project and dividing it into smaller modules will surely make it easy for us to handle and develop. The plan was to start with the basic functionality like user registration and verification, displaying product catalogue, navigation menu etc. After the basic functionality was laid out, start working on other modules to add to the existing one which included cart module, contact module, order module, payment module etc.
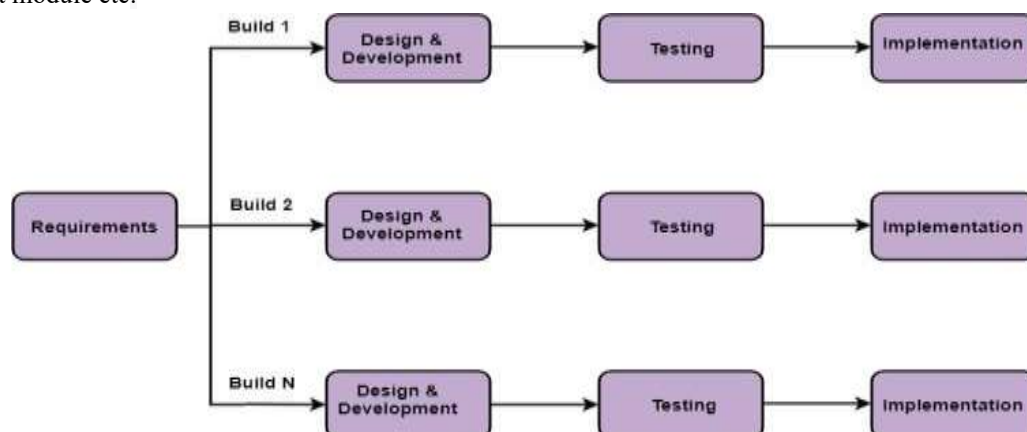
Incremental Model

* **Designing** – After the requirements was laid out, the designing process begin. This process is essential before you actually get to the project implementation. In this process, we designed modeling diagrams so that each possible aspect of system can be visualized. Each diagram helped with a particular aspect which are listed below –

1. **ER – Diagram –** An ER diagram, or Entity-Relationship diagram, is a type of data modeling diagram used to represent the relationships between entities in a database. It is a visual representation of the database schema. Entity is a something about which we want to store data and relationship describes how they are connected with each other.

2. **DFD – Diagram -** A DFD diagram, or Data Flow Diagram, is a graphical which represents the flow or movement of data through a system. It has four main components which are entities, Processes, Data Stores and Data Flows. It has various levels and each level breaks a process into multiple subprocesses to reduce its complexity.

3. **Use Case Diagram -** A Use Case Diagram is a type of behavioural diagram in Unified Modeling Language (UML) that illustrates the interactions between actors(users) and a system. In this, modules or use cases are enclosed in the system and actors(external entities) are connected to those modules which they have the access to.

4. **Sequence Diagram -** A Sequence Diagram is a type of behavioural diagram in Unified Modeling Language (UML) that illustrates the interactions between objects in the sequence in which they occurs. It displays the sequence in which the processes happen. Its major components includes Objects, Lifeline, Messages and activati0n.

5. **Activity Diagram -** An Activity Diagram is a type of behavioural diagram in Unified Modeling Language (UML) that visualizes and represent the flow of activities performed in a system. It is similar to a flow chart. Its components include Start and End nodes, actions, Decision Points, Control Flow, Forks and Joins.

6. **Class Diagram -** A class diagram is a type of static structure diagram in the Unified Modeling Language (UML) that shows the structure of a system by representing the classes, their attributes, operations and relationships among objects.

Implementation -When it comes to implementation of a website, there are three critical components that need to be developed: the front-end, backend, and database. The front-end is what users see and interact with when they visit the website which includes design and layout, the backend is the behind-the-scenes logic that powers the website, including the server, application code, and database queries. Finally, the database stores all the data needed to power the website, including product information, user data, and transaction details. Below is the description of the technologies we used for our project.

Front-end

- **HTML –** HTML stands for Hyper Text Markup Language. It is a markup language used to create and design web pages. HTML consists of a series of elements that are used to define the structure and content of a web page.

- **CSS –** CSS stands for Cascading Style Sheets. It is a style sheet language used to describe the presentation of HTML (and other markup languages) documents. With CSS, you can define the style and layout of a web page, including colors, fonts, margins, padding, and much more.

- **Javascript/JQuery –** JavaScript is a high-level programming language used to create interactive web pages and web applications. It is a client-side scripting language, which means that it is executed by the user's web browser rather than on the server. JQuery allows developers to write less code, as many common JavaScript tasks can be achieved with just a few lines of jQuery code. It provides a set of functions and methods that can be used to manipulate HTML elements, handle events, and perform animations.

- **Bootstrap –** It is a CSS framework which is directed towards making the website responsive. It has various classes and components which we can use to make the designing process easier.

Backend

- **Python –** Python is a high-level, interpreted programming language that is widely used for web development, scientific computing, data analysis, artificial intelligence, and many other applications. It was first released in 1991 by Guido van Rossum and has since become one of the most popular programming languages in the world. We used python as it offers really easy to understand syntax. We used python version 3.10 for our project.

- **Django –** Django is a high-level Python web framework that is designed to help developers build complex, data-driven websites quickly and easily. It was first released in 2005 by a team of developers at Lawrence Journal-World newspaper. We chose to use this framework as it offers many built-in and easy to use functionalities like built in database SQLite, user authentication system and many more.

## 3. HARDWARE AND SOFTWARE REQUIREMENTS –

➤ **Software –**

- **Browser –** Mozilla Firefox, Microsoft Edge, Opera, Chrome or any other.
- **IDE –** Visual Studio Code, Sublime, PyCharm or any other.
- **Operating System –** Windows 7 and above.
- **Languages –** HTML, CSS, JS, Bootstrap, Python and Django.
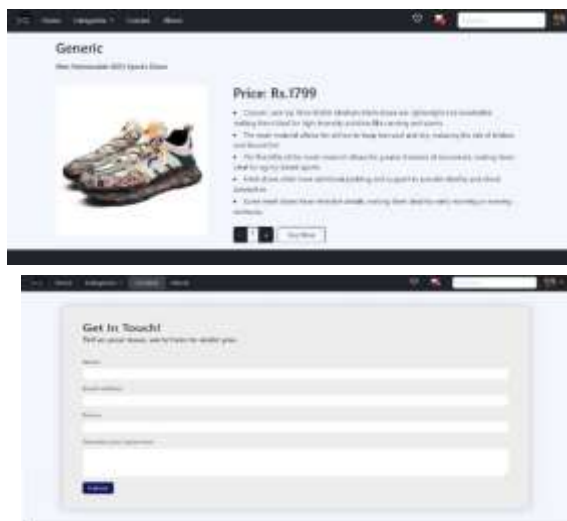
➢ **Hardware –**
• **Processor -** Intel Pentium IV or above.
• **RAM -** 1GB or more.
• **Hard Disk -** 40 GB or more.

4. RESULTS

Homepage and Category page of Shopping Mart Website.
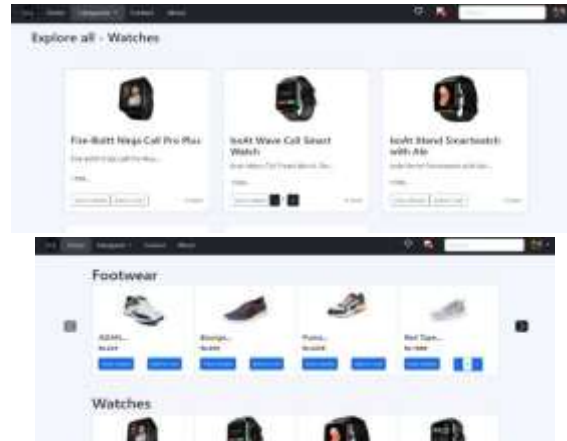
Product page and Contact page.

5. CONCLUSION

Overall, this project delivered the requirements and met the expectations. This website simplifies the shopping process for customers as they can just browse for the product they want and choose from a wide range, add the product to cart and in case of any queries they can also fill and send the query form. Developing this website taught us many concepts that we were unfamiliar with, this gives us the idea of how things are done practically, the process and also the issues that comes with it. At the very last, the ecommerce website using Django has successfully achieved its intended purpose as it will save customers a lot of time and effort and avoid any other inconvenience that comes with offline shopping while also providing valuable learning experiences for the developers.

6. REFERENCES

[1] Software Engineering with UML - Book By Bhuvan Unhelkar.
[2] Software Engineering Book By Chandramouli Subramanian, Saikat Dutt, Chandramouli Seetharaman, B. G Geetha.
[3] docs.djangoproject.com
[4] www.python.org
[5] www.javatpoint.com
[6] www.wikipedia.com
[7] www.w3schools.com
[8] GeeksForGeeks.com