**SEMESTER TRAINING REPORT**

**ON**

**E-COMMERCE WEBSITE**

***Submitted in partial fulfillment of requirements***

***for the award of the degree***

***Bachelor of Technology***

***In***

**Information Technology**

To

**Punjab Technical University, Jalandhar**

**SUBMITTED BY:**

**Name: Ashish Kumar Poonia**

**Roll no.: 1802556**

**Semester: 8th**

**Batch: B.Tech IT**

**Under the guidance of**

**Mr. Anchal**

****

**DEPARTMENT OF INFORMATION TECHNOLOGY**

**Chandigarh Engineering College –Landran**

**Mohali, Punjab - 140307**

**CERTIFICATE**

This is to certify that Mr. Ashish Kumar Poonia has completed the Semester Training during the period from 4 February 2022 to15 August 2022in our Organization / Industry as a Partial Fulfillment of Degree of Bachelor of Technology in Information Technology.

**(Signature of Project supervisor)**

**Date: 15June 2022**

**DECLARATION**

I hereby declare that the Project Report entitled ("E-Commerce Website") is an authentic record of my own work as requirements of 8th sem academic during the period from January to July for the award of degree of B. Tech. (Information Techmnology**, Chandigarh Engineering College, Landran** under the guidance of (Mr. Anchal).

**ASHISH KUMAR POONIA**

**1802556**

**Date: 15June 2022**

Certified that the above statement made by the student is correct to the best of our knowledge and belief.

**Signatures**

**Examined by:**

1. 2. 3. 4.

**Head of Department**

**(Signature and Seal)**

**ACKNOWLEDGMENT**

I take this opportunity to express my sincere gratitude to the Director- Principal Dr. Rajdeep Singh Chandigarh Engineering College, Landran for providing this opportunity to carry out the present work.

I am highly grateful to the Dr. Sushil Kamboj, HOD-IT, Chandigarh Engineering College, Landran (Mohali). I would like to expresses my gratitude to other faculty members of Information Technology department of CEC, Landran for providing academic inputs, guidance & Encouragement throughout the training period. The help rendered by Dr. Amanpreet Kaur and Mr. Mandeep Singh Devgan, Supervisor for Experimentation is greatly acknowledged. Finally, I express my indebtedness to all who have directly or indirectly contributed to the successful completion of my software training.

**ABOUT THE COMPANY**

**Solitaire Infosys** private limited is a private incorporated on 6 june 2011. It is classified as non-govt. Company and registered at registrar of companies, Chandigarh its authorized share capital is Rs.900000 and its paid up capital is Rs.900000. It is involved in other computer related activities.

## Complete IT solution provider

It is a leading software and web application development company, based in Mohali, which provides high quality comprehensive services to enterprises across wide range of platforms and technologies are major areas of expertise are in providing quality, cost effective software and web development.

Our focus is on understanding the diverse and mission critical need of each of our client to understand is to be able to deliver. The competence and experience of our company gives us a competitive edge by making sure we provide the best service and product to our clients. Our high quality standards unable us to deliver reliable and error free software application despite the complexity.

## Campus Edge

Campus Edge is a beneficial campus resource planning software that helps institutes in a better functioning. The management, storage, and administration tasks get easier with our software. This ERP software will reduce our manual efforts and we will be able to maintain all our records more easily.

## Believe in Empowering Innovation through Technology

Company design bilateral and interactive online experiences for their clients with vast knowledge of the latest industry trends and technologies.Company solution and service are created keeping in mine our client’s requirements & expectations and international standard as well.

## Company Expertise

Solitaire Infosys helps enterprise, whether established or startups, to build and grow customercentric digital products for mobile and web. Our clients trust our experience of our decade and our expertise that we have gained after numerous successful deliveries in various fields. We delivered satisfactory services to our clients with an aim of helping their business in growing and reaching their organizational goals.

## Our Vision

We visualize becoming the most trusted and respected IT service provider across the globe with our vibrant, dynamic, and value-based IT solutions that revolve around our clients, team, and international standards. Solitaire Infosys Pvt. Ltd. envisions becoming a reliable partner to all its clients and focusing on doing everything ethically and rightly. We are always open to accept our mistakes and have the nerve to do the necessary changes.

## Our Goal

To grow and flourish as a leading company, to be admired globally to provide atmost IT infrastructure and technical support services.

**Delivering great work experience to our clients is what we aim at.**

Solitaire Infosys Pvt. Ltd. Is an acclaimed IT service provider contributing its part in the development of many businesses around the globe. We socialize with our clients to get a superior cognizance of their business and requirements and help in fabricating websites and applications for their business. Founded in 2011 by a dynamic duo with the same aim and zeal, we have come a long way in satisfying our clients.

We are serving our clients with the world-class services for more than seven years now. The clients are delivered with the best IT solutions after we have developed a great understanding of their business and requirements. Our team works on the client projects like its own and that is the reason why we hold the edge in the league.

With every project that we deliver, we deliver our respect, creativity, quality, transparency, and teamwork to our clients. We have the experience, expertise, and capabilities to enable organizations to accelerate their service processes in every possible way. We are known for our excellent customer satisfaction, cost-effectiveness, and innovative skills that are unparalleled.

## Work

How is working in solitaire Infosys special, find out about the work culture and career growth opportunities in solitaire Infosys at their office in Mohali. Follow them here and find out about their team, the fun and the work they do and stay updated with recruitment news and updates of solitaire Infosys.

## Looking for in youth

As per the latest jobs posted by Solitaire Infosys they are regularly searching for job candidate with talents. They have latest openings. They freely communicated and hire fresher and experienced candidates who have filled their profiles and applied on their jobs at youth4work. What solitaire Infosys are looking for in the talents they add in their team is generally beyond just degrees and certificates but is the real passion for work a youth does. Which is generally seen by the details profiles of youth and talent matrix of youth?

**SOLITAIRE INFOSYS** is a leading software and web application development company, based in Mohali that provides high quality compressive services to enterprises across a wide range of platform and technologies. Our major areas of expertise are in providing quality, cost effective software or web development.

Our focus in own understanding the diverse and mission-critical needs of each of our clients. To understand is to be able to deliver. The competence and experience of our company gives us a competitive edge of making sure we provide the best services and products to our clients. Our high quality standard enables us to delivered reliable and errors-free software applications, despite their complexity.

We provide web design/web development, B2B&B2C E-commerce solutions, SEO &web promotions strategies implementation consulting services to both domestic as well as international clients‟ at the most affordable rates.

**TABLE OF CONTENTS**

1. **ACKNOWLEDGEMENT i**
2. **ABOUT COMPANY ii**
3. **TABLE OF CONTENT iii**
4. **LIST OF FIGURES iv**
5. **CHAPTERS**
6. **INTRODUCTION** 1-8
7. **SOFTWARE REQUIREMENT SPECIFICATION** 9-12
8. **INTRODUCTION AND DETAIL OF SOFTWARE** 13-15
9. **SYSTEM DESIGN** 16-22
10. **SYSTEM IMPLEMENTATION AND TESTING** 23-26
11. **IMPLEMENTATION** 27-61
12. **TESTING** 62-65
13. **SCREENSHOTS** 66-69
14. **CONCLUSION AND FUTURE SCOPE** 70
15. **REFERENCES**  71

**List Of Figures**

1. **Process Flow 17**
2. **Data Flow Diagrams 18**
3. **Flow Chart 19**
4. **UML Diagram 20**
5. **Data Design 21**
6. **Darta Relationship 22**
7. **Add New Product Page 66**
8. **Admin Homepage 66**
9. **Admin Login Page 67**
10. **Category page 67**
11. **HomePage 68**
12. **Customer Login Page 68**
13. **Customer Registration Page 69**
14. **INTRODUCTION**

An E-commerce website requires appropriate strategy of successful design and implementation. Everything is required to plan from scratch to end of website. The e-commerce sector is seen the exponential growth thus a new option will easily part of this regatta of commercial website. The e-commerce website will feature the online shopping facility of various fashion products under a single web space. The proposed web application will allow business personnel to make their total business using it and increase their reachability thousands of times more than today they have, over the internet. It will allow multiple shopping vendors to sale their products online.

The product management in the system will be done in the form of categories. The safety of information is the main requirement of the system and will be handling according to that. To formulate this project first task is to do is cost estimation. For probabilistic assessment of the project cost estimation is required. Cost estimation covers the accurate; estimations of cost and effort required for the project. As a project manager and developer as well, it’s is estimates are defined to early stage in the project.

Cost estimation in application development project includes the set of procedures and techniques that will be utilized, required to produce by organisation for development (Alex,2013). The available resources of a company are also affecting the cost estimation. It will be very complex project. To demonstrate knowledge learnt in class, tech communities and online materials, I will undertake the entire project alone even though it requires a team of 6 or more. It will take time of 3months to get the shape or get the basic structure. The environment variants depend on the further requirements of the ecommerce web application.

1. **BACKGROUND OF THE STUDY**

The traditional marketing and management of fashion industry is experiencing a revolution because of the emergence of e-commerce. Since the birth of e-commerce, businesses have been able to make use of the Internet in reducing costs associated with purchasing, managing supplier relationships, streamlining logistics and inventory, and developing strategic advantage and successful implementation of business re-engineering. E-commerce allows companies to improve communications within the supply chain and enhance service offering, thus providing chances for competitive differentiation.

Fashion is a distinctive and often habitual trend in the style in which a person dress. It is the prevailing styles in behaviour and the newest creations of textile designers. Fashion design is influenced by cultural and social latitudes, and has varied over time and place. Fashion designers attempt to design clothes which are functional as well as aesthetically pleasing. They must consider who is likely to wear a garment and the situations in which it will be worn. Anthropology, the study of culture and human societies, studies fashion by asking why certain styles are deemed socially appropriate and others are not. A certain way is chosen and that becomes the fashion as defined by a certain people as a whole, so if a particular style has a meaning in an already occurring set of beliefs that style will become fashion. According to Ted Polhemus and Lynn Procter (2008), fashion can be described as a beautification. Fashion changes very quickly and is not affiliated with one group or area of the world but is spread out throughout the world wherever people can communicate easily with each other.

The fashion industry is a product of the modern age. Historically, prior to the mid-19th century, most clothing was custom-made. It was handmade for individuals, either as home production or on order from dressmakers and tailors. By the beginning of the 20th century with the rise of new technologies such as the sewing machine, the rise of global capitalism and the development of the factory system of production, and the proliferation of retail outlets such as department stores clothing had increasingly come to be mass-produced in standard sizes and sold at fixed prices.

Although the fashion industry developed first in Europe and America, as of 2014 it is an international and highly globalized industry, with clothing often designed in one country, manufactured in another, and sold world-wide. For example, an American fashion company might source fabric in China and have the clothes manufactured in Vietnam, finished in Italy, and shipped to a warehouse in the United States for distribution to retail outlets internationally. The fashion industry has long been one of the largest employers in the Kenya, and it remains so in the 21st century (from Mitumba sellers to custom made tailors). However, Kenyan government declined considerably as production increasingly shipped from overseas, especially from China. Instead, they encourage people to buy Kenyan made fashion products under the slogan “Buy Kenya, Build Kenya.” Because data on the fashion industry typically are reported for national economies and expressed in terms of the industry’s many separate sectors, aggregate figures for world production of textiles and clothing are difficult to obtain. However, by any measure, the clothing industry accounts for a significant share of world economic output.

The fashion industry consists of four levels:

* The production of raw materials, principally fibres and textiles but also leather and fur.
* The production of fashion goods by designers, manufacturers, contractors, and others.
* Retail sales.
* Various forms of advertising and promotion.

These levels consist of many separate but interdependent sectors. These sectors are Textile Design and Production, Fashion Design and Manufacturing, Fashion Retailing, Marketing and Merchandising, Fashion Shows, and Media and Marketing. Each sector is devoted to the goal of satisfying consumer demand for apparel under conditions that enable participants in the industry to operate at a profit. AP Fashions falls under the retail category and truly need an ecommerce website to market and sell their products. The ecommerce system will make their daily operations and customer service efficient, error free and effective.

1. **STATEMENT OF THE PROBLEM**

Traditionally, customers are used to buying the products at the real, in other words, factual shops or supermarkets. It needs the customers to show up in the shops in person, and walk around different shopping shelves, and it also needs the owners of shops to stock, exhibit, and transfer the products required by customers.

It takes labour, time and space to process these operations. Furthermore, the spread of the Covid-19 pandemic has caused a lot of changes in our lifestyle, people fearing to get outside their homes, transportation almost shut down and social distancing becoming all the more important. Big to small scale business that relied on the traditional incur a lot of consequence due to the lockdown issues. Some tend to more towards using social media platforms like Facebook to sell their product. However, the social media platforms have been beneficial for marketing purposes alone but leaves the whole task of customer and massive order management via direct messaging (DM), which takes a lot of time to respond to all customers. Invaddition, everyone tends to use social media, posing a great challenge to differentiate between scammers (fraudsters) and legit sellers.

1. **THE SOLUTION**

AP Fashions Store is an Online shopping system provides a solution to reduce and optimize these expenses. Authorized Customers do not need to go to the factual shops to choose, and bring the products they need by hands. They simply browse their Personal computers or cell phones to access shops, and evaluate the products description, pictures on the screen to choose products. In addition, the owners of the shop do not need to arrange or exhibit their stocks products. They just input the description, prices of products, and upload their pictures. Simply, both customers and shop owners do not need to touch the real products in the whole process of shopping, and management. In the end the logistic centre will distribute the products required by customers, or products ordered by shop owners to their locations. The customers are able to track the status of their orders until delivery, after which they can leave a review of the type of service they received. The payment and products’ quantity will be saved in database through the data flow. These shopping, management and distribution processes greatly simplify and optimize the retail business.

1. **AIMS AND OBJECTIVES**

The main objective of the study is to develop an online fashion brochure system. The system aims to achieve the following objectives:

* To design an online fashion system.
* To provides a solution to reduce and optimize the expenses of customer order management
* To create an avenue where people can shop for fashion products online.
* To develop a database to store information on fashion products and services.

1. **SCOPE AND LIMITATION**

Every project is done to achieve a set of goals with some conditions keeping in mind that it should be easy to use, feasible and user friendly. As the goal of this project is to develop an online fashion brochure system, this system will be designed keeping in mind the conditions (easy to use, feasibility and user friendly) stated above. It may help in effective and efficient order management. In every shot time, the collection will be obvious, simple and sensible. It is very possible to observe the customer potentials and purchase patterns because all the ordering history is store in the database. It is efficient managing all the operations of an online store within a single platform. The project aims to automate the business process of AP Fashion store. The proposed project would cover:

**Customer Side**

* Customer can view/search products without login.
* Customer can also add/remove product to cart without login (if customer try to add same product in cart. It will add only one)
* When customer try to purchase product, then he/she must login to system. • After creating account and login to system, he/she can place order.
* If customer click on pay button, then their payment will be successful and their order will be placed.
* Customer can check their ordered details by clicking on orders button.
* Customer can see the order status (Pending, Confirmed, Delivered) for each order
* Customer can Download their order invoice for each order • Customer can send feedback to admin (without login)

**Administrator Side`**

* Admin can provide username, email, password and your admin account will be created.
* After login, there is a dashboard where admin can see how many customers is registered, how many products are there for sale, how many orders placed.
* Admin can add/delete/view/edit the products.
* Admin can view/edit/delete customer details.
* Admin can view/delete orders.
* Admin can change status of order (order is pending, confirmed, out for delivery, delivered)
* Admin can view the feedbacks sent by customers

Additionally, if customer places order and admin deleted that user (fraud detection), then their orders will automatically be deleted. Suppose one (1) customer places four (4) products order and admin deleted two (2) product from website, then that two-product order will also be deleted and other two will be there. Also, if user click on purchase button without having products in their cart, then website will ask to add product in cart first.

On the contrary, designing web applications is characterized by some constraints and limitations. Developers are limited to a small set of graphical widgets for use in presenting a user interface. Web-based applications require high investment in software, as well as maintenance costs for the software and personnel for software administration. In this study, verification of credentials for membership cannot be done. Besides, there is a payment page although just for demo. Customers are advised to fill in pseudo details (DONT FILL YOUR ORIGINAL CARD DETAILS THERE). By the way, website do not save these details.

1. **RESEARCH METHODOLOGY**

The research method used for this project work is quantitative research reviews the current system, provide its description, identifying the discrepancies and eventually giving a suitable solution. Therefore, the method used in the design and collections of information from various sources are as follows:

* Studying the present system in detail and the organizational style.
* Knowing and understanding the input and output processes of the existing system.
* A qualitative form of interview was conducted in the organization to understand the mode of operation of the old system.

1. **SIGNIFICANCE OF THE STUDY**

With the aid of an efficient information system, fashion associations can be able to react quickly by giving out information about changes in the market and latest trends to the public. An online application not only saves time and money, but also minimizes administrative efforts and cost. It provides an avenue to market products to a whole new audience. Here are benefits of having an AP Fashion brochure system;

* Easy advertisement of new products and services
* Saves time on the part of the buyer due to the fact that they can do transactions for any product or make enquiries about any product or services provided by a company anytime and anywhere.
* It creates an avenue for expansion to national and international markets.
* An online fashion brochure system improves the brand image of a company.
* It aids a fashion company in providing better customer service.
* It helps to simplify business processes and make them faster and more efficient.

1. **DEFINATION OF TERMS**

* **Fashion**: fashion is a distinctive and often habitual trend in the style in which a person dress.
* **Fashion design**: fashion design is the art of the application of design and natural beauty to clothing and accessories.
* **Fashion designer**: a fashion designer is someone who loves to study fashion trends, sketch designs, select materials, and have a part in all the production aspects of their designs. they contribute to the creation of millions, if not billions of pieces of clothing and accessories purchased by consumers on a yearly basis. fashion designers create women’s, men’s and children’s apparel. this might include sportswear, maternity wear, outerwear, underwear, formalwear, eyewear and footwear. there are also accessory designers who design belts, scarves, hats, handbags and hosiery.
* **Association**: an association is an organized body of people who have an interest, activity, or purpose in common; a society.
* **Web based application:** a web-based application is a software package that can be accessed through the web browser. the software and database reside on a central server rather than being installed on the desktop system and is accessed over a network.
* **Web browser:** a web browser is a software application used to enable computer users locate and access web pages.
* **Brochure:** a brochure is an advertising piece mainly used to introduce a company or an organization and inform about products or services to a target audience.
* **E-commerce**: electronic commerce is the buying and selling of goods and services, or the transmission of funds or data, over an electronic network, primarily the internet.
* **Catalogue:** a product catalogue is a file that contains a list of all the products you want to advertise. Each line of the product catalogue contains a description of each product, including an id, name, category, availability, product URL, image URL and other product attributes.

1. **SOFTWARE REQUIREMENT SPECIFICATION**

A complete understanding of software requirement is essential to the success of a webdevelopment effort. No matter how well designed or well coded, a poorly analysed and specific program will disappoint user and bring grief to the developers.

The requirement analysis task is process of discovery, refinement, modified and specification. The software scope, initially established by the system engineer and refined during project planning, is refined in detail. Models of the required data, information and control flow, and operational behaviour are created. Alternative solutions are analysed and various project element.

Currently who want to buy some shoes or any clothing type they have to go to the shop and buy them this is very tedious for customer therefore we upload this site on internet. This web-site should be developed with an aim to simplify shopping process and keeping transparency and flexibility in performing each operation.

1. **Requirements Gathering**

Also known as data collection. Data Collection is an important aspect of any type of research study. Inaccurate data collection can impact the results of a study and ultimately lead to invalid results. The methods used to gather the projects requirements involves Quantitative research to review the existing systems in the market.

1. **Data Collection Methods**

This study used quantitative techniques like online survey and questionnaire. Qualitative data collection methods play an important role in impact evaluation by providing information useful to understand the processes behind observed results and assess changes in people’s perceptions of their well-being. Furthermore, qualitative methods can be used to improve the quality of surveybased quantitative evaluations by helping generate evaluation hypothesis; strengthening the design of survey questionnaires and expanding or clarifying quantitative evaluation findings. These methods are characterized by the following attributes:

* They tend to be open-ended and have less structured protocols
* They rely more heavily on interactive interviews; respondents may be interviewed several times to follow up on a particular issue, clarify concepts or check the reliability of data
* They use triangulation to increase the credibility of their findings
* Generally, their findings are not generalizable to any specific population, rather each case study produces a single piece of evidence that can be used to seek general patterns among different studies of the same issue

Existing written and visual materials were assessed to find important data and information towards the development of the system. Information about appointment managements, patient’s management were collected. During data collection, the investigation found out how the current system operates, not only that but also tried out which problems are faced and how best they can be settled.

Requirement analysis and specification may appear to be relatively simple task, but appearances are deceiving. Communication content is very high, chances for misinterpretations or misinformation abound. Ambiguity is probable. The dilemma that confronts a software engineer may best be understood by repeating the statement of an anonymous customer: “I know you believe you’re understood what you think I said, but I am not sure you realize that what you heard is not what I meant”.

1. **Requirements**

The requirements form the proposed system was categorized into functional and nonfunctional requirements.

**Functional Requirements**

The following is the desired functionality of the new system. The proposed project would cover:

**Customer Module**

* Customer can view/search products without login.
* Customer can also add/remove product to cart without login (if customer try to add same product in cart. It will add only one)
* When customer try to purchase product, then he/she must login to system.
* After creating account and login to system, he/she can place order.
* If customer click on pay button, then their payment will be successful and their order will be placed.
* Customer can check their ordered details by clicking on orders button.
* Customer can see the order status (Pending, Confirmed, Delivered) for each order
* Customer can Download their order invoice for each order
* Customer can send feedback to admin (without login)

**Admin Module**

* Admin can provide username, email, password and your admin account will be created.
* After login, there is a dashboard where admin can see how many customers is registered, how many products are there for sale, how many orders placed.
* Admin can add/delete/view/edit the products.
* Admin can view/edit/delete customer details.
* Admin can view/delete orders. Admin can change status of order (order is pending, confirmed, out for delivery, delivered)
* Admin can view the feedbacks sent by customers

**Non-functional Requirements**

It specifies the quality attribute of a software system. They judge the software system based on Responsiveness, Usability, Security, Portability and other non-functional standards that are critical to the success of the software system.

* Availability: The system should remain operational in any day and any place. • Accuracy: There is a need to optimize the system to ensure more accurate results and calculations.
* Usability: The system should provide a User-friendly user interface and tooltips to enhance itself and be effectively responsive.
* Secure: The system must be able to provide security against any external injections by using a layered security system. Implementation of user login functionalities also ensures the system is secure from unauthorized persons.
* Performance of the system: Response time is very good for given piece of work. The system will support multi user environment.
* Reliability of the system: The system will be highly reliable and it generates all the updates information in correct order. Data validation and verification is done at every stage of activity. System recovery will also be speed.

1. **System Specifications**

This section describes the hardware components and software requirements needed for effective and efficient running of the system

**Hardware Requirements**

* Processor - 2.4 GHz Processor speed
* Memory - 2 GB RAM
* Disk Space - 500 GB

**Software Requirements**

* Operating System - Windows 8, Windows 10 or MAC Ox 10.8,10.9, or 10.11, LINUX
* Database Management System - SQL Lite 3
* Runtime Environment - PyCharm or Visual Studio Code

1. **Introduction and detail of software used**

E-Commerce, also known as electronic commerce or internet commerce, is an activity of buying and selling goods or services over the internet or open networks. So, any kind of transaction (whether money, funds, or data) is considered as E-commerce. So, E-commerce can be defined in many ways, some define E-Commerce as buying and selling goods and services over the Internet, others define E-Commerce as retail sales to consumers for which the transaction takes place on open networks. The buying and selling of products, services, and digital products through the Internet all fall under the umbrella of e-commerce

E-commerce is an advanced way of conducting businesses online and across the borders and because of that it has various advantages to it:

* Its reach is across the global market and with minimum investments.
* It enables sellers to sell their products on a global level and allows customers to make a broader choice. Now sellers and buyers can meet in the virtual world, without the barrier of borders.
* E-commerce process reduces the product distribution chain to a considerable extent.
* It helps in making a direct and transparent business and transaction channel between the producers, wholesalers and final customers.
* It provides quick delivery of goods and customer complaints are also addressed quickly. It also saves time, energy and effort for both the consumers and the company.
* E-commerce leads to increased productivity and better service as it brings sellers and customers closer.
* The customer can choose between different sellers.
* Customers now have access to virtual stores all the time.
* E-Commerce leads to considerable cost reduction of goods and services. Transaction costs are also reduced in E-commerce and due to that customers get to buy products at a comparatively lower rate.

Details of Software Used to develop this Application:

* **Visual Studio:**

Visual Studio is an Integrated Development Environment(IDE) developed by Microsoft to develop GUI(Graphical User Interface), console, Web applications, web apps, mobile apps, cloud, and web services, etc. With the help of this IDE, you can create managed code as well as native code. It uses the various platforms of Microsoft software development software like Windows store, Microsoft Silverlight, and Windows API, etc. It is not a language-specific IDE as you can use this to write code in C#, C++, VB(Visual Basic), Python, JavaScript, and many more languages. It provides support for 36 different programming languages. It is available for Windows as well as for macOS.

Evolution of Visual Studio: The first version of VS(Visual Studio) was released in 1997, named as Visual Studio 97 having version number 5.0. The latest version of Visual Studio is 15.0 which was released on March 7, 2017. It is also termed as Visual Studio 2017. The supported .Net Framework Versions in latest Visual Studio is 3.5 to 4.7. Java was supported in old versions of Visual Studio but in the latest version doesn’t provide any support for Java language.

There are 3 editions of Microsoft Visual Studio as follows:

1. Community: It is a free version which is announced in 2014. All other editions are paid. This contains the features similar to Professional edition. Using this edition, any individual developer can develop their own free or paid apps like .Net applications, Web applications and many more. In an enterprise organization, this edition has some limitations. For example, if your organization have more than 250 PCs and having annual revenue greater than $1 Million(US Dollars) then you are not permitted to use this edition. In a non-enterprise organization, up to five users can use this edition. Its main purpose is to provide the Ecosystem(Access to thousands of extensions) and Languages(You can code in C#, VB, F#, C++, HTML, JavaScript, Python, etc.) support.

2. Professional: It is the commercial edition of Visual Studio. It comes in Visual Studio 2010 and later versions. It provides the support for XML and XSLT editing and includes the tool like Server Explorer and integration with Microsoft SQL Server. Microsoft provides a free trial of this edition and after the trial period, the user has to pay to continue using it. Its main purpose is to provide Flexibility(Professional developer tools for building any application type), Productivity(Powerful features such as CodeLens improve your team’s productivity), Collaboration(Agile project planning tools, charts, etc.) and Subscriber benefits like Microsoft software, plus Azure, Pluralsight, etc.

3. Enterprise: It is an integrated, end to end solution for teams of any size with the demanding quality and scale needs. Microsoft provides a 90-days free trial of this edition and after the trial period, the user has to pay to continue using it. The main benefit of this edition is that it is highly scalable and deliver high-quality software.

* **SQLite 3**

SQLite is a self-contained, high-reliability, embedded, full-featured, public-domain, SQL database engine. It is the most used database engine in the world. It is an in-process library and its code is publicly available. It is free for use for any purpose, commercial or private. It is basically an embedded SQL database engine. Ordinary disk files can be easily read and write by SQLite because it does not have any separate server like SQL. The SQLite database file format is cross-platform so that anyone can easily copy a database between 32-bit and 64-bit systems. Due to all these features, it is a popular choice as an Application File Format.

1. **System Design**

The section describes the system study, analysis, design strengths and weaknesses of the current system, Contest level diagrams, Entity Relationship Diagram, Architectural design. After interpretation of the data, tables were drawn and process of data determined to guide the researcher of the implementation stage of the project. The tools, which were employed during this methodology stage, where mainly tables, Data Flow Diagrams and Entity Relationship Diagrams. The design ensures that only allows authorized users to access the systems information.

1. **Process Flow**

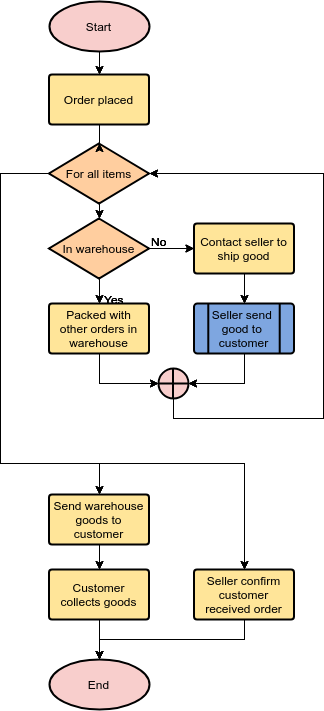


Figure 1: Process Flow Diagram

1. **Data Flow Diagrams**

Figure 2: Data Flow Diagram

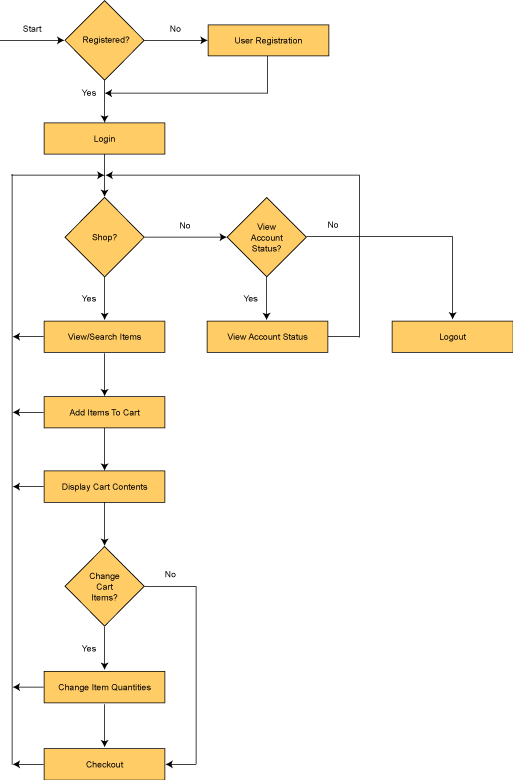
1. **Flow Chart**

Figure 3: Customer shopping flow chat

1. **UML Diagram**

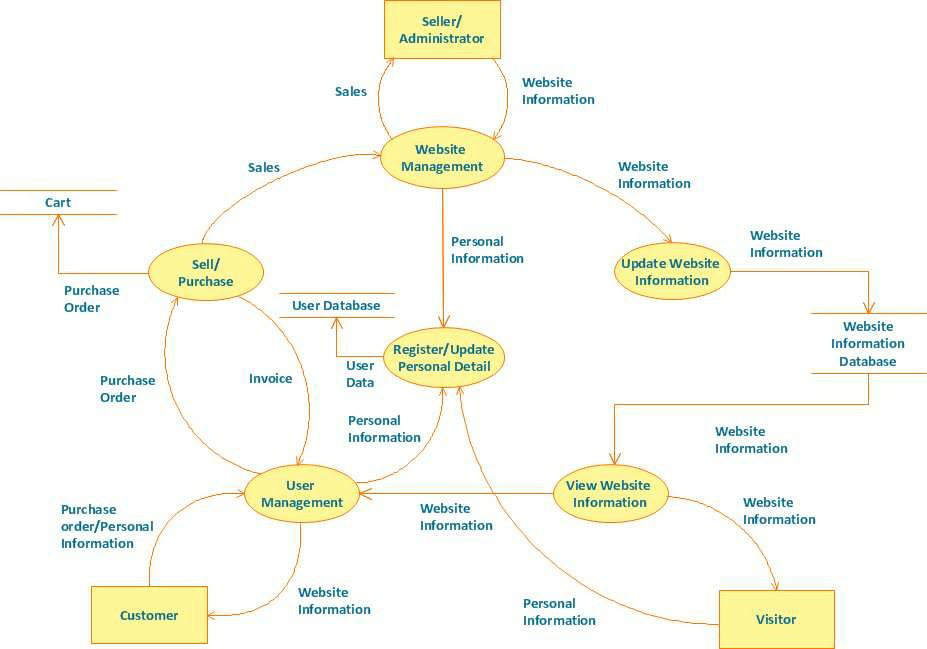


Figure 4: UML Diagram

1. **Data Design**

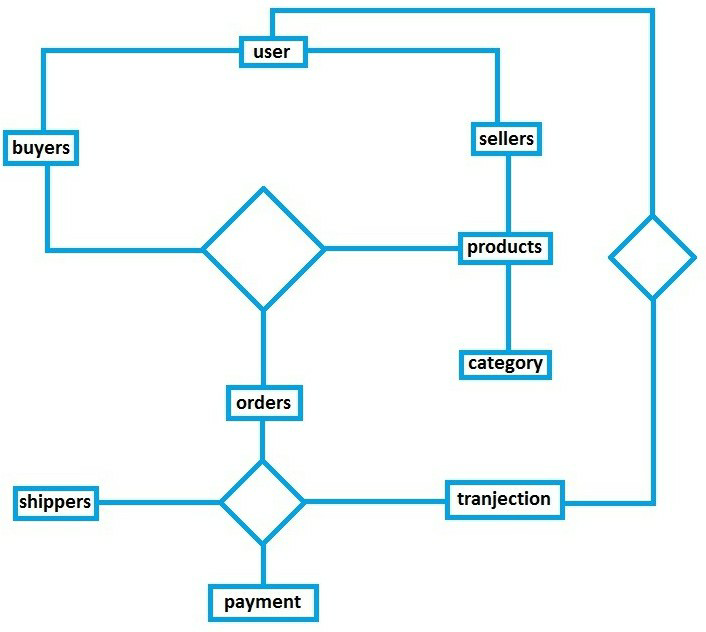


Figure 5: Data Diagram

1. **Data Relationships**

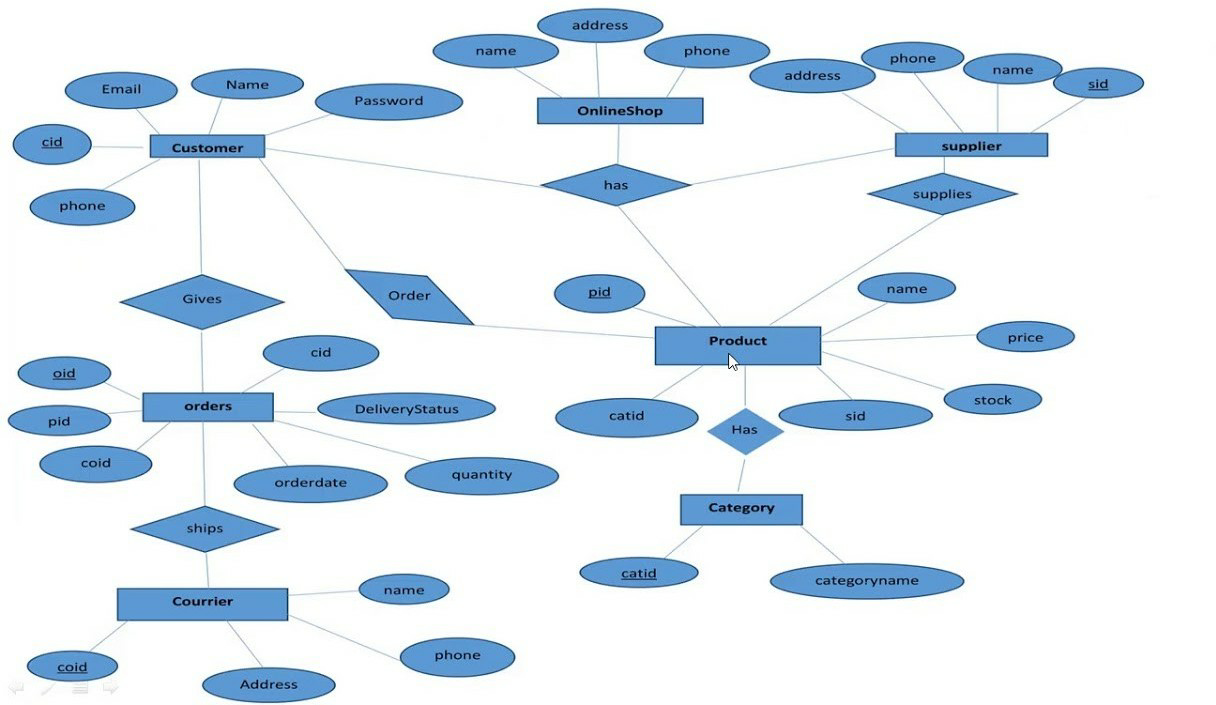


Figure 6: Data Relationship Diagrams

1. **SYSTEM IMPLEMENTATION AND TESTING**

System Testing is a level of testing that validates the complete and fully integrated software product. The purpose of a system test is to evaluate the end-to-end system specifications. Usually, the software is only one element of a larger computer-based system. Ultimately, the software is interfaced with other software/hardware systems. System Testing is a series of different tests whose sole purpose is to exercise the full computer-based system.

* Testing the fully integrated applications including external peripherals in order to check how components interact with one another and with the system as a whole. This is also called End to End testing scenario.
* Verify thorough testing of every input in the application to check for desired outputs.
* Testing of the user’s experience with the application.

There are two Categories of Software Testing

* Black Box Testing
* White Box Testing

System test falls under the black box testing category of software testing.

**White box testing** is the testing of the internal workings or code of a software application. In contrast, black box or System Testing is the opposite. System test involves the external workings of the software from the user’s perspective.

Users interact with your app on a variety of levels, from pressing a button to downloading information onto their device. Accordingly, you should test a variety of use cases and interactions as you iteratively develop your app.

**Organize your code for testing**

As your app expands, you might find it necessary to fetch data from a server, interact with the device's sensors, access local storage, or render complex user interfaces. The versatility of your app demands a comprehensive testing strategy.

**Create and test code iteratively**

When developing a feature iteratively, you start by either writing a new test or by adding cases and assertions to an existing unit test. The test fails at first because the feature isn't implemented yet.

It's important to consider the units of responsibility that emerge as you design the new feature. For each unit, you write a corresponding unit test. Your unit tests should nearly exhaust all possible interactions with the unit, including standard interactions, invalid inputs, and cases where resources aren't available. Take advantage of [Jetpack libraries](https://developer.android.com/jetpack) whenever possible; when you use these well-tested libraries, you can focus on validating behavior that's specific to your app.

Your unit tests should nearly exhaust all possible interactions with the unit, including standard interactions, invalid inputs, and cases where resources aren't available. Take advantage of [Jetpack libraries](https://developer.android.com/jetpack) whenever possible; when you use these well-tested libraries, you can focus on validating behavior that's specific to your app.

When developing a feature iteratively, you start by either writing a new test or by adding cases and assertions to an existing unit test. The test fails at first because the feature isn't implemented yet.

The testing development cycle consists of writing a failing unit
           test, writing code to make it pass, and then refactoring. The entire
           feature development cycle exists inside one step of a larger,
           UI-based cycle.

**Figure -7** Test Cycle

The full workflow, as shown in Figure 7, contains a series of nested, iterative cycles where a long, slow, UI-driven cycle tests the integration of code units. You test the units themselves using shorter, faster development cycles. This set of cycles continues until your app satisfies every use case.

**View your app as a series of modules**

To make your code easier to test, develop your code in terms of modules, where each module represents a specific task that users complete within your app. This perspective contrasts the stack-based view of an app that typically contains layers representing the UI, business logic, and data.

For example, a "task list" app might have modules for creating tasks, viewing statistics about completed tasks, and taking photographs to associate with a particular task. Such a modular architecture also helps you keep unrelated classes decoupled and provides a natural structure for assigning ownership within your development team.

It's important to set well-defined boundaries around each module, and to create new modules as your app grows in scale and complexity. Each module should have only one area of focus, and the APIs that allow for inter-module communication should be consistent. To make it easier and quicker to test these inter-module interactions, consider creating fake implementations of your modules. In your tests, the real implementation of one module can call the fake implementation of the other module.

As you create a new module, however, don't be too dogmatic about making it full featured right away. It's OK for a particular module to not have one or more layers of the app's stack.

**Unit Testing**

The very first stage of any application test. Here, the system’s separate modules will undergo assessments to see if they, individually, function correctly and to maximum capacity. Strictly considering the situation and program of the build, the units may vary significantly from one another. The tester must be a professional with vast knowledge, even on the most minute details, to successfully undergo this stage. Whenever modifications are made, especially on the core of the software code, it’s necessary to apply unit testing. This is so that any underlying issues are resolved efficiently and on time.

**Integration Testing**

This level of testing is mostly about verifying the modules and checking their readiness and their collective, integral cooperation. The modules are each tested separately and also as a group. This aids the testers to identify any issues with two or more components working together or individually to execute functions.

No matter how healthy any component seems to be, one can’t tell with all certainty if the software is working to its full potential until an integration test is performed.

**System Testing**

This testing level closely simulates the final production environment and is very significant as the final testing stage, especially when it needs to ensure that the applications under scrutiny always meet up to full functional requirements. This is where the test team ascertains if the integrated components are collectively showing optimal performance levels or not. For this, every software build must undergo testing up to this point, using client requirements as a bench-mark.

**Acceptance Testing**

Some requirements to be implemented could be easily misconstrued before, or even during the development process after it commences. In any case, this happens, it’s difficult to point an error out since there is practically none that hinders normal functions. This is essentially why acceptance testing is undertaken with end-users’ help to test the software functions in a similar environment as the real end-users would. In general, acceptance testing ensures they work as they should and to maximum capacity.

1. **Implementation**
2. **Base.html**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>{% block title %}{% endblock %}</title>

<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">

</head>

<body>

<nav class="navbar navbar-expand-lg navbar-dark bg-info">

<a class="navbar-brand" href="{% url 'ecomapp:home' %}">AP Fashion</a>

<button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse" id="navbarSupportedContent">

<ul class="navbar-nav mr-auto">

<li class="nav-item active">

<a class="nav-link" href="{% url 'ecomapp:home' %}">Home <span class="sr-only">(current)</span></a>

</li>

<li class="nav-item">

<a class="nav-link" href="{% url 'ecomapp:allproducts' %}">Categories</a>

</li>

<li class="nav-item">

<a class="nav-link" href="{% url 'ecomapp:mycart' %}">Cart</a>

</li>

<li class="nav-item dropdown">

<a class="nav-link dropdown-toggle" href="#" id="navbarDropdown" role="button" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">

My Account

</a>

<div class="dropdown-menu" aria-labelledby="navbarDropdown">

{% if request.user.customer %}

<a class="dropdown-item" href="{% url 'ecomapp:customerprofile' %}">My Profile</a>

<a class="dropdown-item" href="{% url 'ecomapp:customerlogout' %}">Logout</a>

{% else %}

<a class="dropdown-item" href="{% url 'ecomapp:customerregistration' %}">Register</a>

<a class="dropdown-item" href="{% url 'ecomapp:customerlogin' %}">Login</a>

{% endif %}

</div>

</li>

</ul>

<form class="form-inline my-2 my-lg-0" action="{% url 'ecomapp:search' %}" method="GET">

<input name="keyword" class="form-control mr-sm-2" type="search" placeholder="Search" aria-label="Search" required>

<button class="btn btn-danger my-2 my-sm-0" type="submit">Search</button>

</form>

</div>

</nav>

<hr>

{% block content %}

{% endblock %}

<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js" integrity="sha384-DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj" crossorigin="anonymous"></script>

<script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.1/dist/umd/popper.min.js" integrity="sha384-9/reFTGAW83EW2RDu2S0VKaIzap3H66lZH81PoYlFhbGU+6BZp6G7niu735Sk7lN" crossorigin="anonymous"></script>

<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js" integrity="sha384-B4gt1jrGC7Jh4AgTPSdUtOBvfO8shuf57BaghqFfPlYxofvL8/KUEfYiJOMMV+rV" crossorigin="anonymous"></script>

</body>

</html>

1. **About.html**

|  | {% extends 'base.html' %} |
| --- | --- |
|  |  |
|  | {% block title %}About us{% endblock %} |
|  |  |
|  | {% block content %} |
|  | <h1>We are the emerging ecommerce in the City.</h1> |
|  | {% endblock %} |

1. **Addtocart.html**

|  | {% extends 'base.html' %} |
| --- | --- |
|  |  |
|  | {% block title %}Add to cart{% endblock %} |
|  |  |
|  | {% block content %} |
|  | <div class="container"> |
|  | <h3>Product added to cart successfully.</h3> |
|  | </div> |
|  | {% endblock %} |

1. **Allproducts.html**

{% extends 'base.html' %}

{% block title %} All Products {% endblock %}

{% block content %}

<div class="container">

<h3>All Products</h3><hr>

{% for cat in allcategories %}

<h4>{{cat.title}}</h4><hr>

<div class="row">

{% for p in cat.product\_set.all %}

<div class="col-md-3">

<div class="m-4">

<h4><a href="{% url 'ecomapp:productdetail' p.slug %}">{{p.title}}</a></h4>

<img src="{{p.image.url}}" alt="" class="img-fluid" style="height: 2500; object-fit: cover;">

<p class="mt-3">Price: <strike>Rs. {{p.marked\_price}}</strike> Rs. {{p.selling\_price}}</p>

<a href="{% url 'ecomapp:addtocart' p.id %}" class="btn btn-primary">Add To Cart</a>

</div>

</div>

{% endfor %}

</div>

{% endfor %}

</div>

{% endblock %}

1. **Checkout.html**

|  | {% extends 'base.html' %} |
| --- | --- |
|  |  |
|  | {% block title %}Checkout {% endblock %} |
|  |  |
|  | {% block content %} |
|  | <div class="container"> |
|  | <div class="row"> |
|  | <div class="col-md-6"> |
|  | <h3>Items in the cart</h3> |
|  | <a href="[{% url 'ecomapp:mycart' %}](about:blank)">Update Cart</a> |
|  | <table class="table"> |
|  | <thead> |
|  | <tr> |
|  | <th>SN</th> |
|  | <th>Product</th> |
|  | <th>Rate</th> |
|  | <th>Quantity</th> |
|  | <th>Subtotal</th> |
|  | </tr> |
|  | </thead> |
|  | <tbody> |
|  | {% for cp in cart.cartproduct\_set.all %} |
|  | <tr> |
|  | <td>{{forloop.counter}}</td> |
|  | <td>{{cp.product.title}}</td> |
|  | <td>{{cp.rate}}</td> |
|  | <td>{{cp.quantity}}</td> |
|  | <td>{{cp.subtotal}}</td> |
|  | </tr> |
|  | {% endfor %} |
|  | <tr> |
|  | <th colspan="4" class="text-right">Total</th> |
|  | <th>Rs. {{cart.total}}</th> |
|  | </tr> |
|  | </tbody> |
|  | </table> |
|  | </div> |
|  | <div class="col-md-6"> |
|  | <h3>Checkout Form</h3> |
|  | <form action="" method="POST"> |
|  | {% csrf\_token %} |
|  | {{form.as\_p}} |
|  | <button class="btn btn-info">Place order</button> |
|  | </form> |
|  | </div> |
|  | </div> |
|  | </div> |
|  | {% endblock %}   1. **Contactus.html**  |  | {% extends 'base.html' %} | | --- | --- | |  |  | |  | {% block title %}Contact us{% endblock %} | |  |  | |  | {% block content %} | |  | <h1>To Contact us, Call the following numbers</h1> | |  | <p>989898989898</p> | |  | <p>989898989898</p> | |  | {% endblock %}   1. **Customerlogin.html**  |  | {% extends 'base.html' %} | | --- | --- | |  |  | |  | {% block title %}Customer Login{% endblock %} | |  |  | |  | {% block content %} | |  | <style> | |  | .errorlist { | |  | color: red; | |  | } | |  | </style> | |  | <div class="container"> | |  | <div class="row"> | |  | <div class="col-md-6 mx-auto"> | |  | <h3>Customer Login Form</h3> | |  | <hr> | |  | <form action="" method="POST"> | |  | {% csrf\_token %} | |  | {{form.as\_p}} | |  | <button class="btn btn-primary">Login as a Customer</button> | |  | {% if error %} | |  | <li style="color: red;">{{error}}</li> | |  | {% endif %} | |  | </form> | |  | <p>Forgot Password?<a href="[{% url 'ecomapp:passworforgot' %}](about:blank)">Reset now.</a></p> | |  | <p> or dont have account? <a href="[{% url 'ecomapp:customerregistration' %}{% if request.GET.next %}?next={{request.GET.next}}{% endif %}](about:blank)">Register here</a></p> | |  | </div> | |  | </div> | |  | </div> | |  | {% endblock %}   1. **Customerorderdetails.html**  |  | {% extends "base.html" %} | | --- | --- | |  | {% load humanize %} | |  |  | |  | {% block title %}Order Detail {% endblock %} | |  |  | |  | {% block content %} | |  | <div class="container"> | |  | <div class="row"> | |  | <div class="col-md-8 mx-auto"> | |  | <h3>Order Detail</h3><hr> | |  | <p><span class="font-weight-bold">Order ID: </span> #ORDER\_{{ord\_obj.id}}</p> | |  | <p><span class="font-weight-bold">Order Status: </span> {{ord\_obj.order\_status}}</p> | |  | <p><span class="font-weight-bold">Order On: </span> {{ord\_obj.created\_at}}({{ord\_obj.created\_at|timesince}} ago)</p> | |  | <p><span class="font-weight-bold">Customer Information: </span> By {{ord\_obj.ordered\_by}}, {{ord\_obj.mobile}}</p> | |  | <p><span class="font-weight-bold">Shipping Address: </span> By {{ord\_obj.shipping\_address}} ({{ord\_obj.email}})</p> | |  | <h4>Ordered Items</h4> | |  | <table class="table"> | |  | <thead> | |  | <tr> | |  | <th>SN</th> | |  | <th>Item</th> | |  | <th>Rate</th> | |  | <th>Quantity</th> | |  | <th>Subtotal</th> | |  | </tr> | |  | </thead> | |  | <tbody> | |  | {% for cp in ord\_obj.cart.cartproduct\_set.all %} | |  | <tr> | |  | <td>{{forloop.counter}}</td> | |  | <td><a target="\_blank" href="[{% url 'ecomapp:productdetail' cp.product.slug %}](about:blank)">{{cp.product.title}}({{cp.product.id}})</a></td> | |  | <td>{{cp.rate}}</td> | |  | <td>{{cp.quantity}}</td> | |  | <td>{{cp.subtotal}}</td> | |  | </tr> | |  | {% endfor %} | |  | <tr> | |  | <th colspan="4" class="text-right">Total: </th> | |  | <th>Rs {{ord\_obj.total|intcomma}}/-</th> | |  | </tr> | |  | </tbody> | |  | </table> | |  | </div> | |  | </div> | |  | </div> | |  |  | |  | {% endblock %} | |  1. **Home.html**  |  | {% extends 'base.html' %} | | --- | --- | |  |  | |  | {% block title %}Home{% endblock %} | |  |  | |  | {% block content %} | |  | <div class="container"> | |  |  | |  | <h1 class="text-center">Welcome to our ecommerce website. </h1> | |  | <hr> | |  | <div class="row mt-3"> | |  | {% for p in product\_list %} | |  | <div class="col-md-3 card"> | |  | <div class="m-4"> | |  | <h4><a href="[{% url 'ecomapp:productdetail' p.slug %}](about:blank)">{{p.title}}</a></h4> | |  | <img src="[{{p.image.url}}](about:blank)" alt="" class="img-fluid" style="height: 200px; object-fit: contain;"> | |  | <p class="mt-3">Price: <strike>Rs. {{p.marked\_price}}</strike> Rs. {{p.selling\_price}}</p> | |  | <a href="[{% url 'ecomapp:addtocart' p.id %}](about:blank)" class="btn btn-primary">Add To Cart</a> | |  | </div> | |  | </div> | |  | {% endfor %} | |  | </div> | |  | <nav aria-label="..."> | |  | <ul class="pagination"> | |  | {% if product\_list.has\_previous %} | |  | <li class="page-item"> | |  | <a class="page-link" href="[?page={{ product\_list.previous\_page\_number }}](about:blank)">Previous</a> | |  | </li> | |  | {% else %} | |  | <li class="page-item disabled"> | |  | <a class="page-link" href="[#](about:blank)" tabindex="-1" aria-disabled="true">Previous</a> | |  | </li> | |  | {% endif %} | |  |  | |  | {% for i in product\_list.paginator.page\_range %} | |  | {% if i == product\_list.number %} | |  | <li class="page-item active"><a class="page-link">{{i}}</a></li> | |  | {% else %} | |  | <li class="page-item"><a class="page-link" href="[?page={{i}}](about:blank)">{{i}}</a></li> | |  | {% endif %} | |  | {% endfor %} | |  |  | |  | {% if product\_list.has\_next %} | |  | <li class="page-item"> | |  | <a class="page-link" href="[?page={{ product\_list.next\_page\_number }}](about:blank)">Next</a> | |  | </li> | |  | {% else %} | |  | <li class="page-item disabled"> | |  | <a class="page-link" href="[#](about:blank)" tabindex="-1" aria-disabled="true">Next</a> | |  | </li> | |  | {% endif %} | |  | </ul> | |  | </nav> | |  |  | |  | </div> | |  | {% endblock %}   1. Mycart.html  |  | {% extends 'base.html' %} | | --- | --- | |  |  | |  | {% block title %} My cart{% endblock %} | |  |  | |  | {% block content %} | |  | <div class="container"> | |  | <div class="row"> | |  | <div class="col-md-8"> | |  | <h4>Items in my cart</h4><hr> | |  | <table class="table"> | |  | <thead> | |  | <tr> | |  | <th>SN</th> | |  | <th>Product</th> | |  | <th>Rate</th> | |  | <th>Quantity</th> | |  | <th>Subtotal</th> | |  | <th>Action</th> | |  | </tr> | |  | </thead> | |  | <tbody> | |  | {% for cp in cart.cartproduct\_set.all %} | |  | <tr> | |  | <td>{{forloop.counter}}</td> | |  | <td>{{cp.product.title}}</td> | |  | <td>{{cp.rate}}</td> | |  | <td>{{cp.quantity}}</td> | |  | <td>{{cp.subtotal}}</td> | |  | <td> | |  | <a href="[{% url 'ecomapp:managecart' cp.id %}?action=inc](about:blank)" class="btn btn-primary"> + </a> | |  | <a href="[{% url 'ecomapp:managecart' cp.id %}?action=dcr](about:blank)" class="btn btn-warning"> - </a> | |  | <a href="[{% url 'ecomapp:managecart' cp.id %}?action=rmv](about:blank)" class="btn btn-danger"> x </a> | |  | </td> | |  | </tr> | |  | {% endfor %} | |  | <tr> | |  | <th colspan="4" class="text-right">Total</th> | |  | <th>Rs. {{cart.total}}</th> | |  | {% if cart.cartproduct\_set.all %} | |  | <th><a href="[{% url 'ecomapp:emptycart' %}](about:blank)">Empty Cart</a></th> | |  | {% endif %} | |  | </tr> | |  | {% if cart.cartproduct\_set.all %} | |  | <tr> | |  | <th><a class="btn btn-info" href="[{% url 'ecomapp:checkout' %}](about:blank)"> Checkout </a></th> | |  | </tr> | |  | {% endif %} | |  | </tbody> | |  | </table> | |  | </div> | |  | </div> | |  | </div> | |  | {% endblock %} |  1. **Passwordreset.html**  |  | {% extends 'base.html' %} | | --- | --- | |  |  | |  | {% block title %}Reset password{% endblock %} | |  |  | |  | {% block content %} | |  | <div class="container"> | |  | <div class="row"> | |  | <div class="col-md-8"> | |  | <form action="" method="POST"> | |  | {% csrf\_token %} | |  | {{form.as\_p}} | |  | <button>Reset password</button> | |  | </form> | |  | </div> | |  | </div> | |  | </div> | |  |  | |  | {% endblock %} |  1. **Productdetail.html**  |  | {% extends 'base.html' %} | | --- | --- | |  |  | |  | {% block title %}{{product.title}}{% endblock %} | |  |  | |  | {% block content %} | |  | <div class="container"> | |  | <h3>Product: {{product.title}}</h3><hr> | |  | <div class="row"> | |  | <div class="col-md-4"> | |  | <img src="[{{product.image.url}}](about:blank)" alt="" class="img-fluid"> | |  | <div class="row"> | |  | {% for pi in product.productimage\_set.all %} | |  | <div class="col-md-4 p-3"> | |  | <a href="[{{pi.image.url}}](about:blank)" target="\_blank"> | |  | <img src="[{{pi.image.url}}](about:blank)" class="img-fluid" style="height: 100px; object-fit: contain;" alt=""> | |  | </a> | |  | </div> | |  | {% endfor %} | |  | </div> | |  | </div> | |  | <div class="col-md-8"> | |  | <h4>{{product.title}} (Viewed {{product.view\_count}} times)</h4> | |  | <h5>Category: {{product.category}}</h5> | |  | <h5>Price: <strike>Rs. {{product.marked\_price}}</strike> Rs. {{product.selling\_price}}</h5> | |  | <a href="[{% url 'ecomapp:addtocart' product.id %}](about:blank)" class="btn btn-info">Add To Cart</a> | |  | <hr> | |  | <p class="text-info">Warranty: {{product.warranty}}</p> | |  | <p class="text-info">Return Policy:{{product.return\_policy}}</p> | |  | <hr> | |  | <p>{{product.description}}</p> | |  | </div> | |  | </div> | |  |  | |  | </div> | |  | {% endblock %} |  1. **Seaech.html**  |  | {% extends 'base.html' %} | | --- | --- | |  |  | |  | {% block title %}Search {% endblock %} | |  |  | |  | {% block content %} | |  | <div class="container"> | |  | <div class="row"> | |  | <div class="col-md-12"> | |  | <h3>Search Results for <span class="text-info">"{{request.GET.keyword}}"</span></h3> | |  | <hr> | |  | {% for result in results %} | |  | <div class="row"> | |  | <div class="col-md-3"> | |  | <img src="[{{result.image.url}}](about:blank)" class="img-fluid" alt=""> | |  | </div> | |  | <div class="col-md-9"> | |  | <h4><a href="[{% url 'ecomapp:productdetail' result.slug %}](about:blank)">{{result.title}}</a></h4> | |  | <p>{{result.description|truncatewords:50}}</p> | |  | <p>Return Policy: {{result.return\_policy}}</p> | |  | <a href="[{% url 'ecomapp:addtocart' result.id %}](about:blank)" class="btn btn-primary">Add To Cart</a> | |  | </div> | |  | </div> | |  | {% endfor %} | |  | </div> | |  | </div> | |  | </div> | |  |  | |  | {% endblock %} |  1. **Adminbase.html**  |  | <!DOCTYPE html> | | --- | --- | |  | <html lang="en"> | |  | <head> | |  | <meta charset="UTF-8"> | |  | <meta name="viewport" content="width=device-width, initial-scale=1.0"> | |  | <title>{% block title %}{% endblock %}</title> | |  | <link rel="stylesheet" href="<https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css>"> | |  | </head> | |  | <body> | |  | <div class="container"> | |  | <a href="[{% url 'ecomapp:adminhome' %}](about:blank)">Home</a> | | |  | <a href="[{% url 'ecomapp:adminorderlist' %}](about:blank)">All Orders</a> | | |  | <a href="[{% url 'ecomapp:adminproductlist' %}](about:blank)">All Products</a> | | |  | <a href="[{% url 'ecomapp:home' %}](about:blank)" target="\_blank">Visit site</a> | |  | </div> | |  | {% block content %} | |  |  | |  | {% endblock %} | |  | </body> | |  | </html> |  1. **Adminproductlist.html**  |  | {% extends 'adminpages/adminbase.html' %} | | --- | --- | |  | {% block title %}All Products{% endblock %} | |  |  | |  |  | |  | {% block content %} | |  | <div class="container"> | |  | <div class="row"> | |  | <div class="col-md-12"> | |  | <h3>All Product list</h3><hr> | |  | <a href="[{% url 'ecomapp:adminproductcreate' %}](about:blank)">Add New Product</a> | |  | <table class="table table-striped"> | |  | <thead> | |  | <tr> | |  | <th>SN</th> | |  | <th>Product Title</th> | |  | <th>Image</th> | |  | <th>Price</th> | |  | <th>Views</th> | |  | </tr> | |  | </thead> | |  | <tbody> | |  | {% for p in allproducts %} | |  | <tr> | |  | <td>{{forloop.counter}}</td> | |  | <td>{{p.title}}</td> | |  | <td><img src="[{{p.image.url}}](about:blank)" style="height: 60px;" alt=""></td> | |  | <td>{{p.selling\_price}}</td> | |  | <td>{{p.view\_count}}</td> | |  | </tr> | |  | {% endfor %} | |  | </tbody> | |  | </table> | |  | </div> | |  | </div> | |  | </div> | |  | {% endblock %} |  1. **Adminproductcreate.html**  |  | {% extends 'adminpages/adminbase.html' %} | | --- | --- | |  |  | |  | {% block title %}Add New Product{% endblock %} | |  |  | |  | {% block content %} | |  | <div class="container"> | |  | <div class="row"> | |  | <div class="col-md-10 mx-auto"> | |  | <h3>Add New product</h3> | |  | <hr> | |  | <form action="" method="POST" enctype="multipart/form-data"> | |  | {% csrf\_token %} | |  | {{form.as\_p}} | |  | <button class="btn btn-info">Submit Data</button> | |  | </form> | |  | </div> | |  | </div> | |  | </div> | |  | {% endblock %} |  1. **Forgetpassword.html**  |  | {% extends 'base.html' %} | | --- | --- | |  |  | |  | {% block title %}Forgot password{% endblock %} | |  |  | |  | {% block content %} | |  | <div class="container"> | |  | <div class="row"> | |  | <div class="col-md-8"> | |  | {% if request.GET.m == "s" %} | |  | <h3>Password reset link is sent to your email address. Please check your email immediately. </h3> | |  | {% else %} | |  | <h3>Enter your email used in making customer account to get password reset link</h3> | |  | {% if request.GET.m == "e" %} | |  | <h4 class="text-danger">Something went wrong, please enter your email again.</h4> | |  | {% endif %} | |  | <form action="" method="POST"> | |  | {% csrf\_token %} | |  | {{form.as\_p}} | |  | <button> Get Reset Link</button> | |  | </form> | |  | {% endif %} | |  | </div> | |  | </div> | |  | </div> | |  | {% endblock %} |  1. **Adminorderdetails.html**  |  | {% extends 'adminpages/adminbase.html' %} | | --- | --- | |  | {% load humanize %} | |  |  | |  | {% block title %}Order Detail{% endblock %} | |  |  | |  | {% block content %} | |  | <div class="container"> | |  | <div class="row"> | |  | <div class="col-md-8 mx-auto"> | |  | <h3>Order Detail</h3> | |  | <hr> | |  | <p><span class="font-weight-bold">Order ID: </span> #ORDER\_{{ord\_obj.id}}</p> | |  | <p><span class="font-weight-bold">Order Status: </span> {{ord\_obj.order\_status}}</p> | |  | <p><span class="font-weight-bold">Order On: </span> {{ord\_obj.created\_at}}({{ord\_obj.created\_at|timesince}} ago)</p> | |  | <p><span class="font-weight-bold">Customer Information: </span> By {{ord\_obj.ordered\_by}}, {{ord\_obj.mobile}}</p> | |  | <p><span class="font-weight-bold">Shipping Address: </span> By {{ord\_obj.shipping\_address}} ({{ord\_obj.email}})</p> | |  | <p><span class="font-weight-bold">Payment Information: </span> Using {{ord\_obj.payment\_method}} ({% if ord\_obj.payment\_completed %}Payment Completed{% else %}Not Paid{% endif %})</p> | |  | <h4>Ordered Items</h4> | |  | <table class="table"> | |  | <thead> | |  | <tr> | |  | <th>SN</th> | |  | <th>Item</th> | |  | <th>Rate</th> | |  | <th>Quantity</th> | |  | <th>Subtotal</th> | |  | </tr> | |  | </thead> | |  | <tbody> | |  | {% for cp in ord\_obj.cart.cartproduct\_set.all %} | |  | <tr> | |  | <td>{{forloop.counter}}</td> | |  | <td><a target="\_blank" | |  | href="[{% url 'ecomapp:productdetail' cp.product.slug %}](about:blank)">{{cp.product.title}}({{cp.product.id}})</a> | |  | </td> | |  | <td>{{cp.rate}}</td> | |  | <td>{{cp.quantity}}</td> | |  | <td>{{cp.subtotal}}</td> | |  | </tr> | |  | {% endfor %} | |  | <tr> | |  | <th colspan="4" class="text-right">Total: </th> | |  | <th>Rs {{ord\_obj.total|intcomma}}/-</th> | |  | </tr> | |  | </tbody> | |  | </table> | |  | <form action="{% url 'ecomapp:adminorderstatuschange' ord\_obj.id %}" method="POST"> | |  | {% csrf\_token %} | |  | <select name="status"> | |  | {% for i, j in allstatus %} | |  | <option value="{{i}}" {% if i == ord\_obj.order\_status %}selected{% endif %}>{{j}}</option> | |  | {% endfor %} | |  | </select> | |  | <button class="btn btn-info">Change</button> | |  |  | |  | </form> | |  | </div> | |  | </div> | |  | </div> | |  | {% endblock %} | |  |  | | | |

1. **Models.py**

from django.db import models

from django.contrib.auth.models import User

# Create your models here.

class Admin(models.Model):

user = models.OneToOneField(User, on\_delete=models.CASCADE)

full\_name = models.CharField(max\_length=50)

image = models.ImageField(upload\_to="admins")

mobile = models.CharField(max\_length=20)

def \_\_str\_\_(self):

return self.user.username

class Customer(models.Model):

user = models.OneToOneField(User, on\_delete=models.CASCADE)

full\_name = models.CharField(max\_length=200)

address = models.CharField(max\_length=200, null=True, blank=True)

joined\_on = models.DateTimeField(auto\_now\_add=True)

def \_\_str\_\_(self):

return self.full\_name

class Category(models.Model):

title = models.CharField(max\_length=200)

slug = models.SlugField(unique=True)

def \_\_str\_\_(self):

return self.title

class Product(models.Model):

title = models.CharField(max\_length=200)

slug = models.SlugField(unique=True)

category = models.ForeignKey(Category, on\_delete=models.CASCADE)

image = models.ImageField(upload\_to="products")

marked\_price = models.PositiveIntegerField()

selling\_price = models.PositiveIntegerField()

description = models.TextField()

warranty = models.CharField(max\_length=300, null=True, blank=True)

return\_policy = models.CharField(max\_length=300, null=True, blank=True)

view\_count = models.PositiveIntegerField(default=0)

def \_\_str\_\_(self):

return self.title

class ProductImage(models.Model):

product = models.ForeignKey(Product, on\_delete=models.CASCADE)

image = models.ImageField(upload\_to="products/images/")

def \_\_str\_\_(self):

return self.product.title

class Cart(models.Model):

customer = models.ForeignKey(

Customer, on\_delete=models.SET\_NULL, null=True, blank=True)

total = models.PositiveIntegerField(default=0)

created\_at = models.DateTimeField(auto\_now\_add=True)

def \_\_str\_\_(self):

return "Cart: " + str(self.id)

class CartProduct(models.Model):

cart = models.ForeignKey(Cart, on\_delete=models.CASCADE)

product = models.ForeignKey(Product, on\_delete=models.CASCADE)

rate = models.PositiveIntegerField()

quantity = models.PositiveIntegerField()

subtotal = models.PositiveIntegerField()

def \_\_str\_\_(self):

return "Cart: " + str(self.cart.id) + " CartProduct: " + str(self.id)

ORDER\_STATUS = (

("Order Received", "Order Received"),

("Order Processing", "Order Processing"),

("On the way", "On the way"),

("Order Completed", "Order Completed"),

("Order Canceled", "Order Canceled"),

)

METHOD = (

("Cash On Delivery", "Cash On Delivery"),

("Khalti", "Khalti"),

("Esewa", "Esewa"),

)

class Order(models.Model):

cart = models.OneToOneField(Cart, on\_delete=models.CASCADE)

ordered\_by = models.CharField(max\_length=200)

shipping\_address = models.CharField(max\_length=200)

mobile = models.CharField(max\_length=10)

email = models.EmailField(null=True, blank=True)

subtotal = models.PositiveIntegerField()

discount = models.PositiveIntegerField()

total = models.PositiveIntegerField()

order\_status = models.CharField(max\_length=50, choices=ORDER\_STATUS)

created\_at = models.DateTimeField(auto\_now\_add=True)

payment\_method = models.CharField(

max\_length=20, choices=METHOD, default="Cash On Delivery")

payment\_completed = models.BooleanField(

default=False, null=True, blank=True)

def \_\_str\_\_(self):

return "Order: " + str(self.id)

1. **Forms.py**

from django import forms

from .models import Order, Customer, Product

from django.contrib.auth.models import User

class CheckoutForm(forms.ModelForm):

class Meta:

model = Order

fields = ["ordered\_by", "shipping\_address",

"mobile", "email", "payment\_method"]

class CustomerRegistrationForm(forms.ModelForm):

username = forms.CharField(widget=forms.TextInput())

password = forms.CharField(widget=forms.PasswordInput())

email = forms.CharField(widget=forms.EmailInput())

class Meta:

model = Customer

fields = ["username", "password", "email", "full\_name", "address"]

def clean\_username(self):

uname = self.cleaned\_data.get("username")

if User.objects.filter(username=uname).exists():

raise forms.ValidationError(

"Customer with this username already exists.")

return uname

class CustomerLoginForm(forms.Form):

username = forms.CharField(widget=forms.TextInput())

password = forms.CharField(widget=forms.PasswordInput())

class ProductForm(forms.ModelForm):

more\_images = forms.FileField(required=False, widget=forms.FileInput(attrs={

"class": "form-control",

"multiple": True

}))

class Meta:

model = Product

fields = ["title", "slug", "category", "image", "marked\_price",

"selling\_price", "description", "warranty", "return\_policy"]

widgets = {

"title": forms.TextInput(attrs={

"class": "form-control",

"placeholder": "Enter the product title here..."

}),

"slug": forms.TextInput(attrs={

"class": "form-control",

"placeholder": "Enter the unique slug here..."

}),

"category": forms.Select(attrs={

"class": "form-control"

}),

"image": forms.ClearableFileInput(attrs={

"class": "form-control"

}),

"marked\_price": forms.NumberInput(attrs={

"class": "form-control",

"placeholder": "Marked price of the product..."

}),

"selling\_price": forms.NumberInput(attrs={

"class": "form-control",

"placeholder": "Selling price of the product..."

}),

"description": forms.Textarea(attrs={

"class": "form-control",

"placeholder": "Description of the product...",

"rows": 5

}),

"warranty": forms.TextInput(attrs={

"class": "form-control",

"placeholder": "Enter the product warranty here..."

}),

"return\_policy": forms.TextInput(attrs={

"class": "form-control",

"placeholder": "Enter the product return policy here..."

}),

}

class PasswordForgotForm(forms.Form):

email = forms.CharField(widget=forms.EmailInput(attrs={

"class": "form-control",

"placeholder": "Enter the email used in customer account..."

}))

def clean\_email(self):

e = self.cleaned\_data.get("email")

if Customer.objects.filter(user\_\_email=e).exists():

pass

else:

raise forms.ValidationError(

"Customer with this account does not exists..")

return e

class PasswordResetForm(forms.Form):

new\_password = forms.CharField(widget=forms.PasswordInput(attrs={

'class': 'form-control',

'autocomplete': 'new-password',

'placeholder': 'Enter New Password',

}), label="New Password")

confirm\_new\_password = forms.CharField(widget=forms.PasswordInput(attrs={

'class': 'form-control',

'autocomplete': 'new-password',

'placeholder': 'Confirm New Password',

}), label="Confirm New Password")

def clean\_confirm\_new\_password(self):

new\_password = self.cleaned\_data.get("new\_password")

confirm\_new\_password = self.cleaned\_data.get("confirm\_new\_password")

if new\_password != confirm\_new\_password:

raise forms.ValidationError(

"New Passwords did not match!")

return confirm\_new\_password

1. **Urls.py**

from django.urls import path

from .views import \*

app\_name = "ecomapp"

urlpatterns = [

# Client side pages

path("", HomeView.as\_view(), name="home"),

path("about/", AboutView.as\_view(), name="about"),

path("contact-us/", ContactView.as\_view(), name="contact"),

path("all-products/", AllProductsView.as\_view(), name="allproducts"),

path("product/<slug:slug>/", ProductDetailView.as\_view(), name="productdetail"),

path("add-to-cart-<int:pro\_id>/", AddToCartView.as\_view(), name="addtocart"),

path("my-cart/", MyCartView.as\_view(), name="mycart"),

path("manage-cart/<int:cp\_id>/", ManageCartView.as\_view(), name="managecart"),

path("empty-cart/", EmptyCartView.as\_view(), name="emptycart"),

path("checkout/", CheckoutView.as\_view(), name="checkout"),

path("khalti-request/", KhaltiRequestView.as\_view(), name="khaltirequest"),

path("khalti-verify/", KhaltiVerifyView.as\_view(), name="khaltiverify"),

path("esewa-request/", EsewaRequestView.as\_view(), name="esewarequest"),

path("esewa-verify/", EsewaVerifyView.as\_view(), name="esewaverify"),

path("register/",

CustomerRegistrationView.as\_view(), name="customerregistration"),

path("logout/", CustomerLogoutView.as\_view(), name="customerlogout"),

path("login/", CustomerLoginView.as\_view(), name="customerlogin"),

path("profile/", CustomerProfileView.as\_view(), name="customerprofile"),

path("profile/order-<int:pk>/", CustomerOrderDetailView.as\_view(),

name="customerorderdetail"),

path("search/", SearchView.as\_view(), name="search"),

path("forgot-password/", PasswordForgotView.as\_view(), name="passworforgot"),

path("password-reset/<email>/<token>/",

PasswordResetView.as\_view(), name="passwordreset"),

# Admin Side pages

path("admin-login/", AdminLoginView.as\_view(), name="adminlogin"),

path("admin-home/", AdminHomeView.as\_view(), name="adminhome"),

path("admin-order/<int:pk>/", AdminOrderDetailView.as\_view(),

name="adminorderdetail"),

path("admin-all-orders/", AdminOrderListView.as\_view(), name="adminorderlist"),

path("admin-order-<int:pk>-change/",

AdminOrderStatuChangeView.as\_view(), name="adminorderstatuschange"),

path("admin-product/list/", AdminProductListView.as\_view(),

name="adminproductlist"),

path("admin-product/add/", AdminProductCreateView.as\_view(),

name="adminproductcreate"),

]

1. **Testing**

eCommerce testing is defined as testing of an eCommerce (online shopping) application. It helps in the prevention of errors and adds value to the product by ensuring conformity to client requirements.

The objective of testing is to ensure

* Software reliability
* Software quality
* System Assurance
* Optimum performance and capacity utilization

Setting up an E-commerce system is a complex process and subject to many market-specific variables. To maintain the integrity of the E Commerce system, testing becomes compulsory

## General Test Cases

* Verify that the user is able to navigate through all the products across different categories.
* Verify that all the links and banners are redirecting to correct product/category pages and none of the links are broken.
* Verify that the company logo is clearly visible.
* Verify that all the text – product, category name, price, and product description are clearly visible.
* Verify that all the images – product and banner are clearly visible.
* Verify that category pages have a relevant product listed specific to the category.
* Verify that the correct count of total products is listed on the category pages.
* Search – Verify that on searching all the product satisfying the search criteria are visible on the search result page
* Search – Verify the more relevant product for the search term is displayed on the top for a particular search term.
* Search – Verify that count of products is correctly displayed on the search result page for a particular search term.
* Filtering – Verify that filtering functionality correctly filters products based on the filter applied.
* Filtering – Verify that filtering works correctly on category pages.
* Filtering – Verify that filtering works correctly on the search result page.
* Filtering – Verify that the correct count of total products is displayed after a filter is applied.
* Sorting – Verify that all the sort options work correctly – correctly sort the products based on the sort option chosen.
* Sorting – Verify that sorting works correctly on the category pages.
* Sorting – Verify that sorting works correctly on the search result page.
* Sorting – Verify that sorting works correctly on the pages containing the filtered result, after applying filters.
* Sorting – Verify that product count remains intact irrespective of sorting option applied.

## Product Buy Flow – Test cases

* Verify that on the product page, the user can select the desired attribute of the product e.g. size, color, etc.
* Verify that the user can add to cart one or more products.
* Verify that users can add products to the wishlist.
* Verify that the user can buy products added to the cart after signing in to the application (or as per the functionality of the website).
* Verify that the user can successfully buy more than one product that was added to his/her cart
* Verify that the user cannot add more than the available inventory of the product.
* Verify that the limit to the number of products a user can by is working correctly by displaying an error message and preventing the user from buying more than the limit.
* Verify that the delivery can be declined for the places where shipping is not available.
* Verify that the Cash on Delivery option of payment is working fine.
* Verify that the different pre-paid methods of payments are working fine.
* Verify that product return functionality works fine.

## User(Buyer) Registration – Test cases

* Verify that all the specified fields are present on the registration page.
* Verify that the required/mandatory fields are marked with \* against the field.
* Verify that for better user interface dropdowns, radio buttons and checkboxes, etc fields are displayed wherever possible instead of just textboxes
* Verify the page has both submit and cancel/reset buttons at the end.
* Verify that clicking submits button after entering all the required fields, submits the data to the server.
* Verify that clicking cancels/reset button after entering all the required fields, cancels the submit request, and reset all the fields.
* Verify that whenever possible validation should take place at client side
* Verify that not filling the mandatory fields and clicking the submit button will lead to validation error.
* Verify that not filling the optional fields and clicking the submit button will still send data to the server without any validation error.
* Check the upper limit of the textboxes.
* Check validation on the date and email fields (only valid dates and valid email Ids should be allowed.
* Check validation on numeric fields by entering alphabets and special characters.
* Verify that leading and trailing spaces are trimmed.
* Verify that entering blank spaces on mandatory fields leads to validation error.
* Verify that after making a request to the server and then sending the same request again with the same unique key will lead to server-side validation error.

## Seller – Product creation Test cases

* Verify that authenticated sellers get access to product creation panels specific to the authorized categories.
* Verify that the product creation panel is working fine for single product creation.
* Verify that the product creation panel is working fine for multiple product creation.
* Verify that the maximum product creation limit for the seller is working fine, limiting the seller to create more than the desired number of products.
* Verify panel validation for checking mandatory fields.
* Verify that duplicate product creation is restricted through the panel.
* Verify that seller can update the information and price of existing products.
* Verify that products created by sellers get visible on the website after a certain period of time.
* Verify that updates made by the seller get visible on the website after a certain period of time.

1. **SCREENSHOTS**
2. **Add New Product Page**

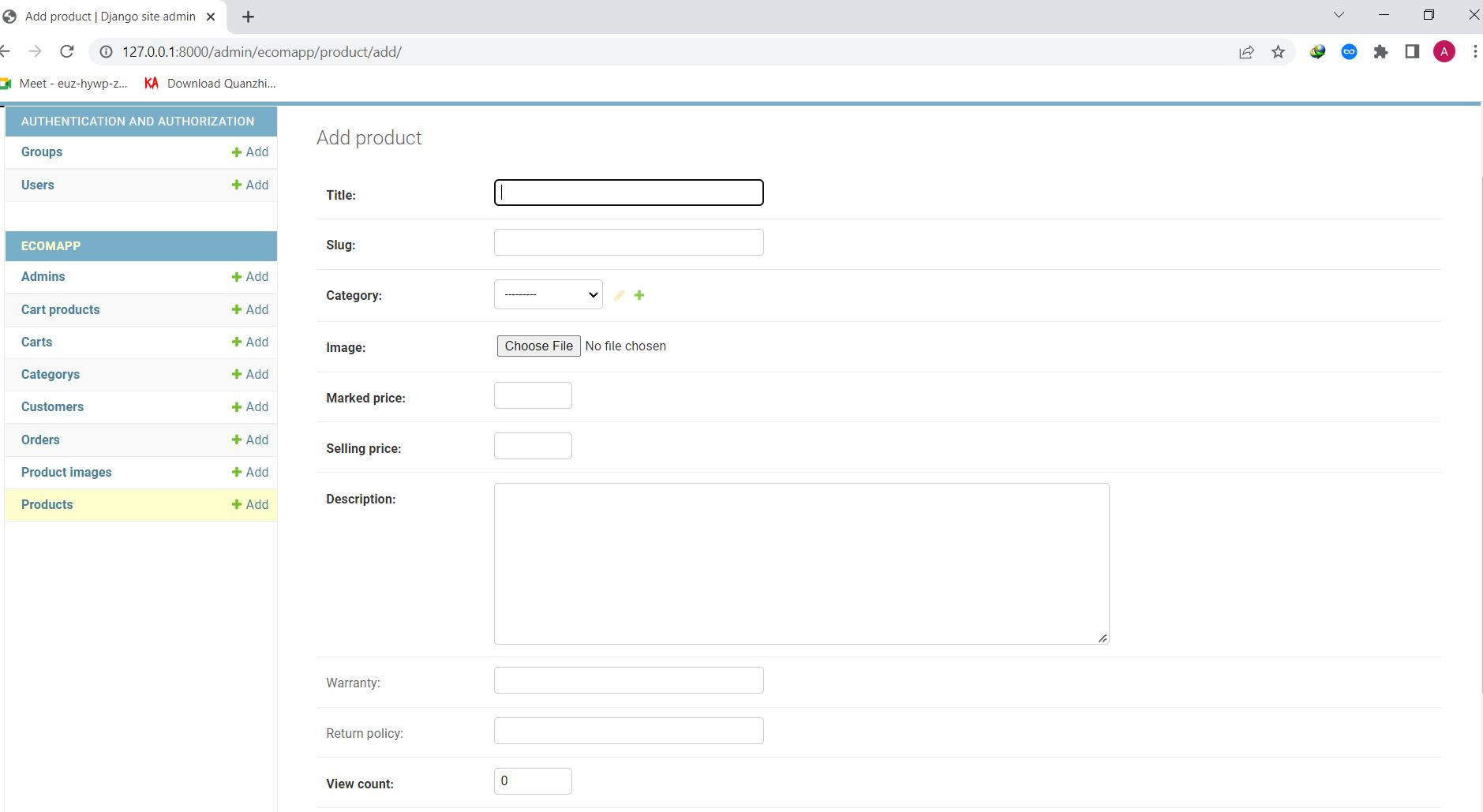
****

Figure 7 – Add New Product Page

1. **Admin Homepage**

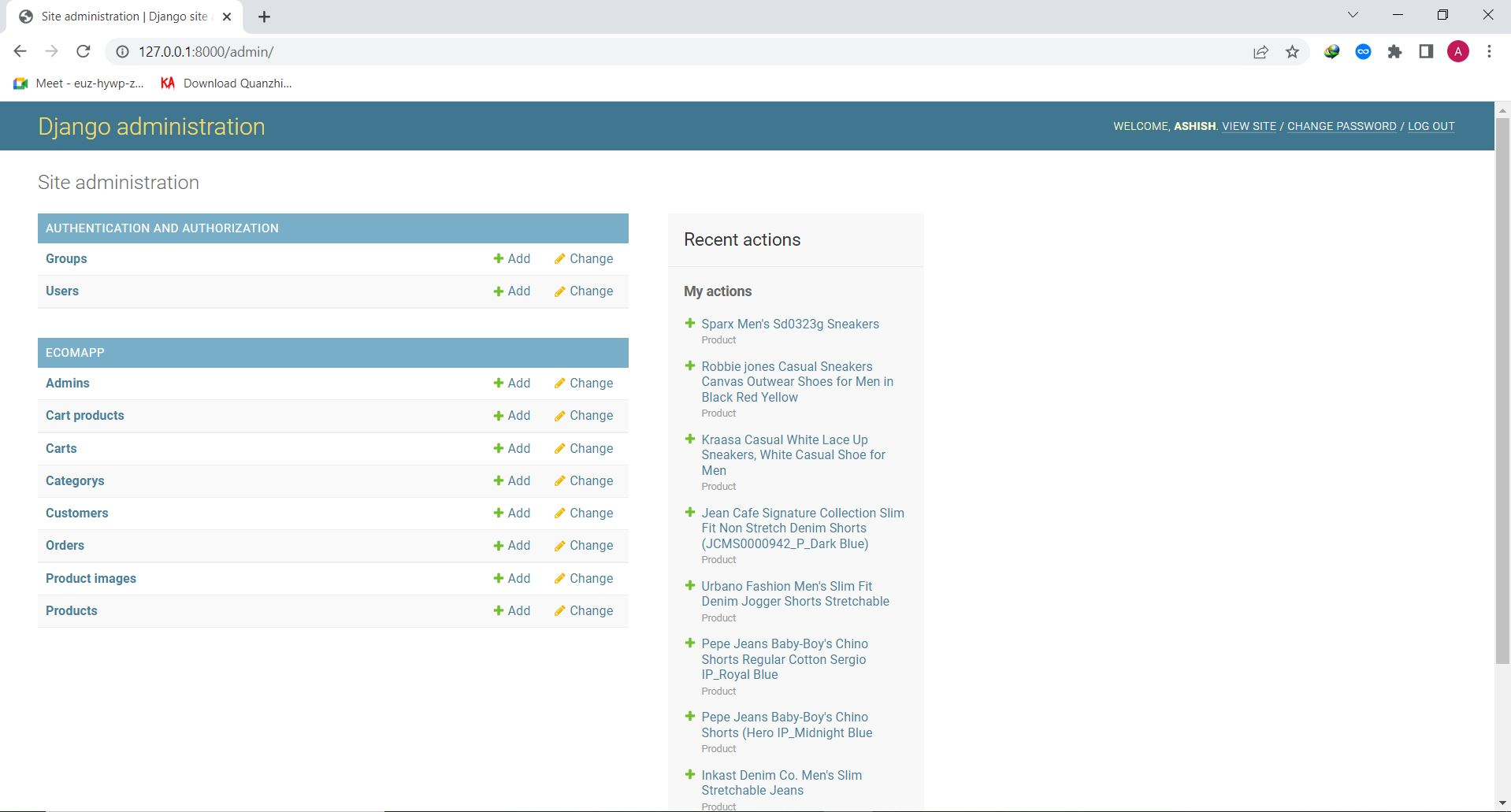
****

Figure 8- Admin Homepage

1. **Admin Login Page**

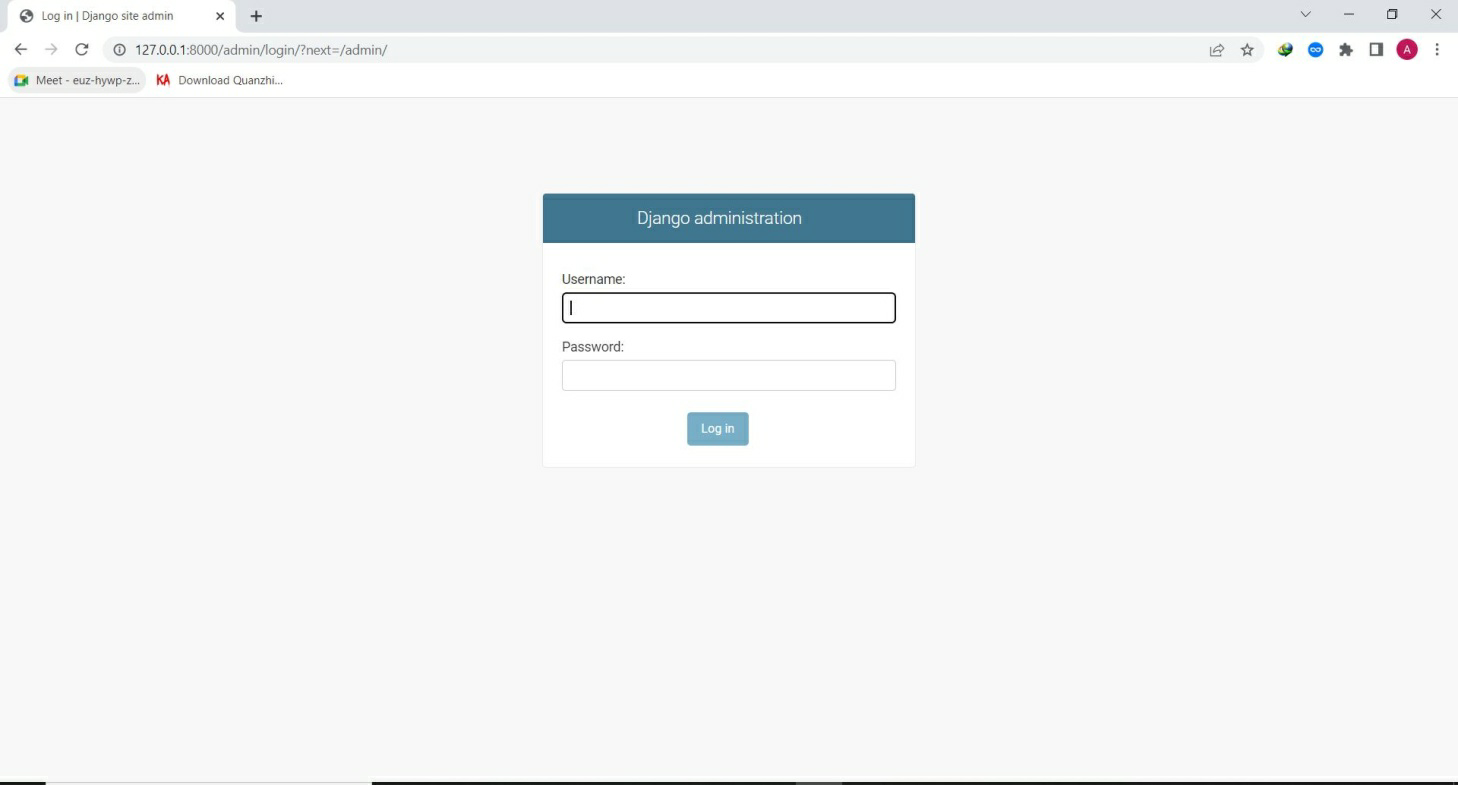
****

Figure9 – Admin Login Page

1. **Admin Category Page**

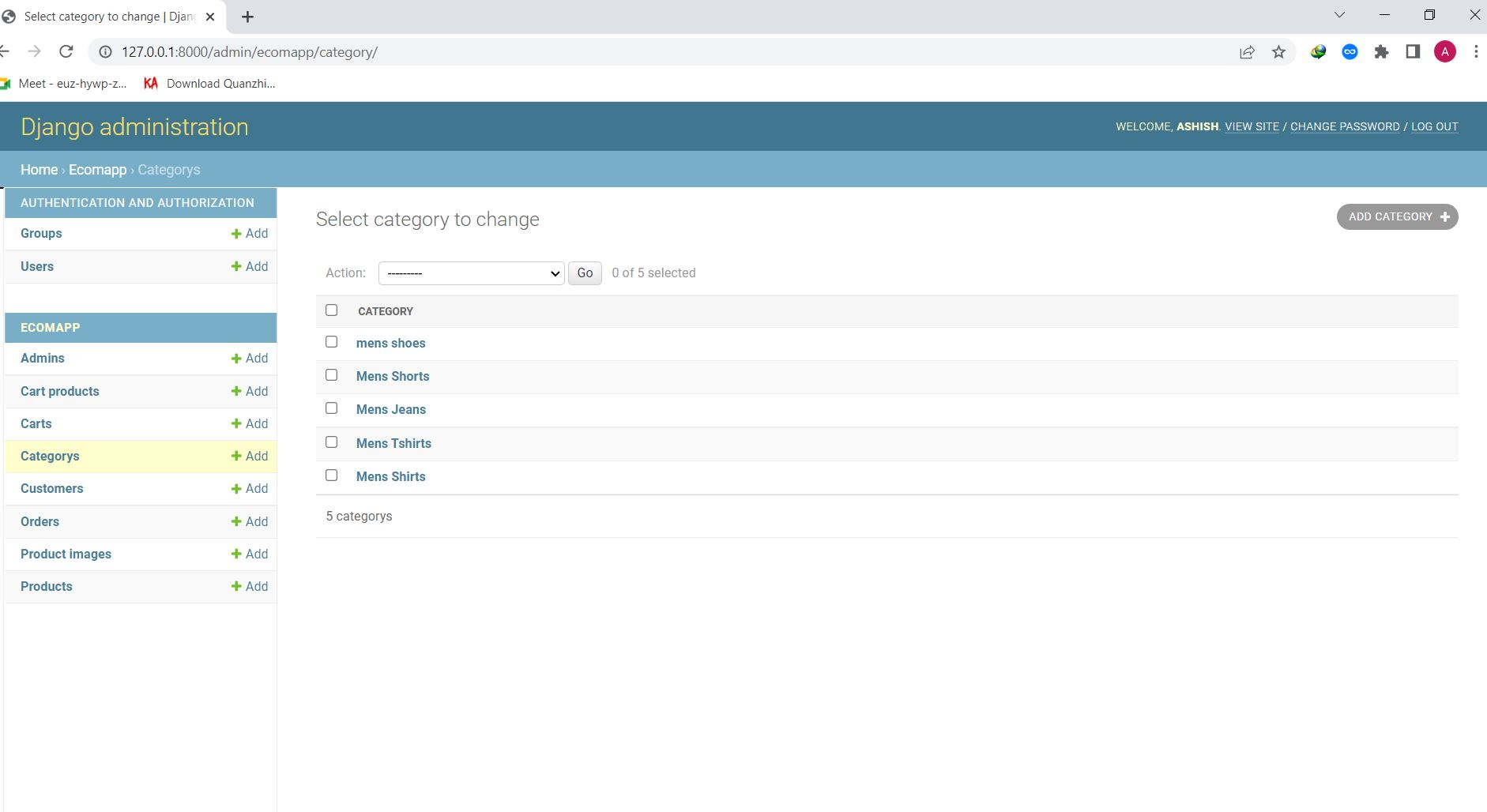
****

Figure 10 – Category page

1. **HomePage**

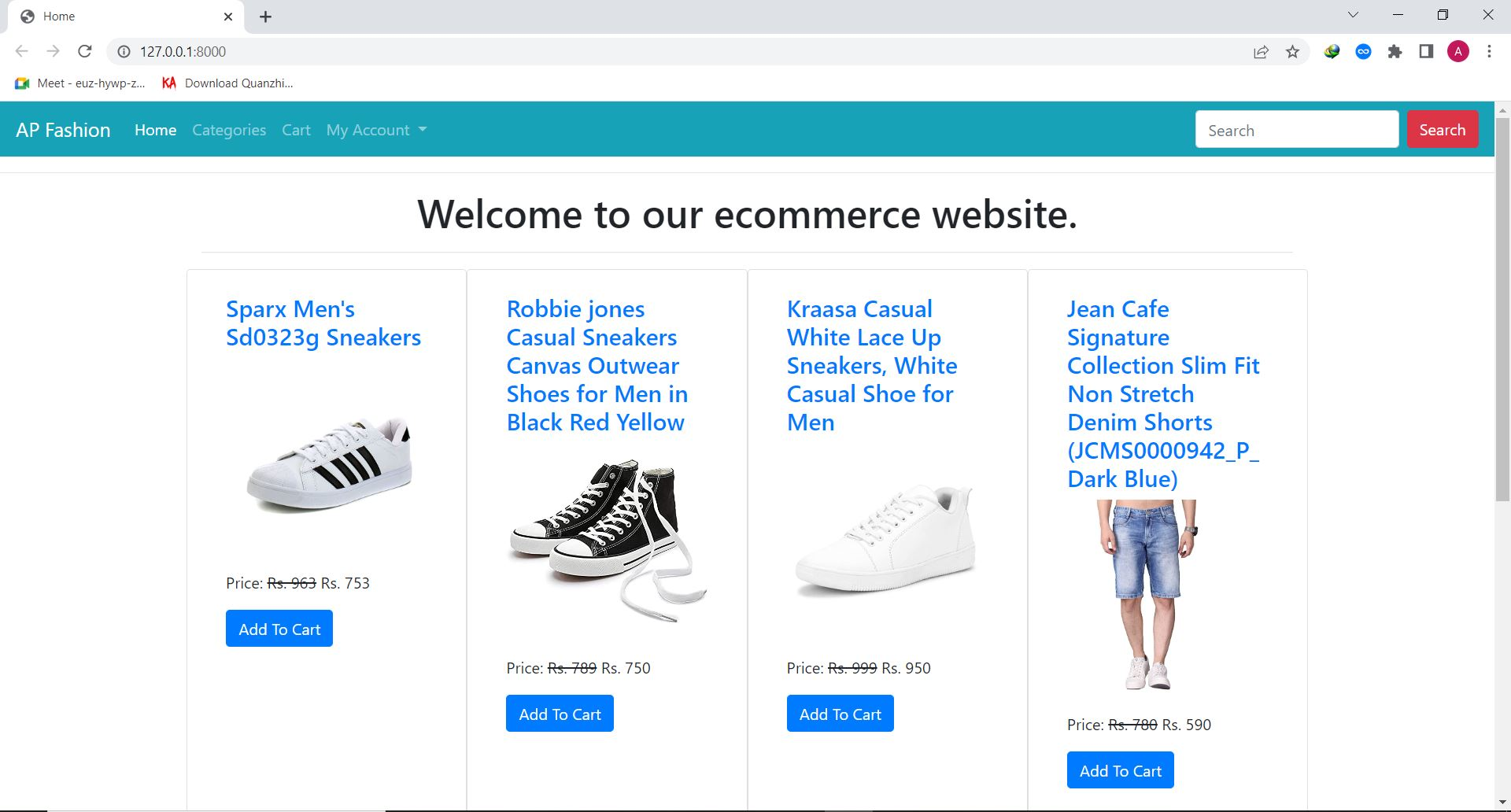
****

Figure 11 – HomePage

1. **Customer Login Page**

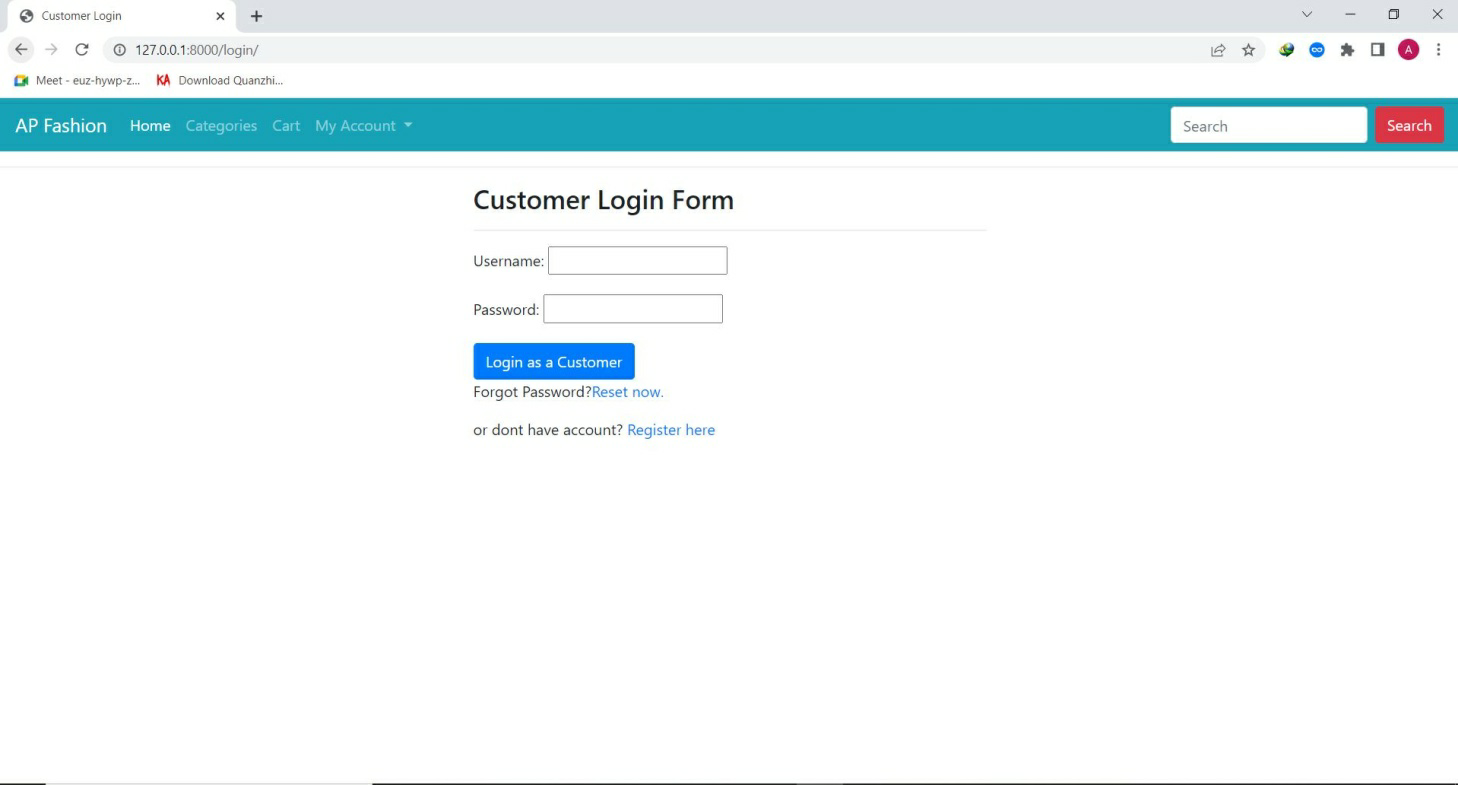
****

Figure 12 – Customer Login Page

1. **Customer Registration Page**

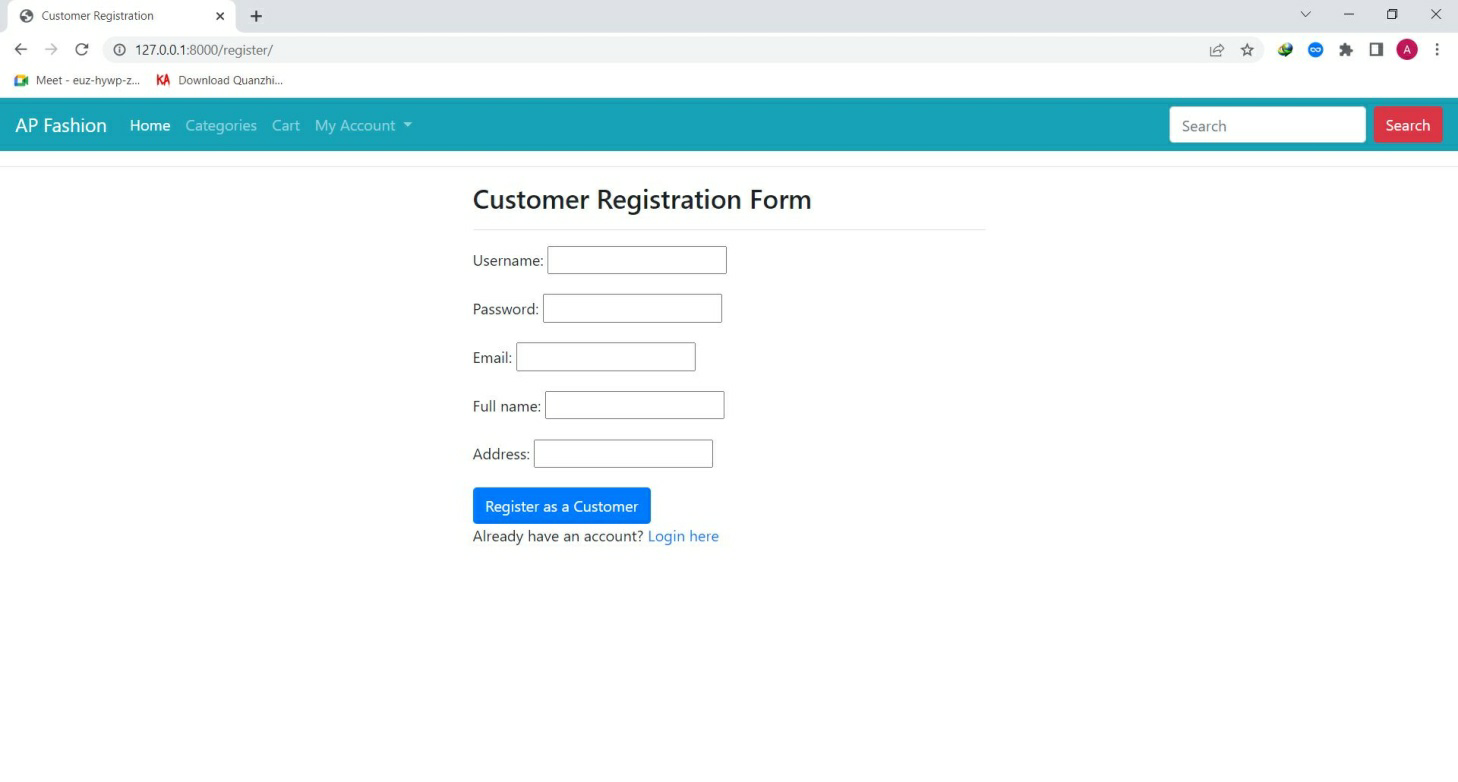
****

Figure 13- Customer Registration Page

1. **CONCLUSION AND FUTURE SCOPE**

**CONCLUSION**

The project entitled AP Fashion Store system was completed successfully. The system has been developed with much care and free of errors and at the same time it is efficient and less time consuming. The purpose of this project was to develop a web application for purchasing items from a fashion shop. This project enabled me gain valuable information and practical knowledge on several topics like designing web pages using html & CSS, usage of responsive templates, designing of full stack Django application, and management of database using SQLite 3. The entire system is secured. Also, the project helped me understanding about the development phases of a project and software development life cycle. I learned how to test different features of a project. This project has given me great satisfaction in having designed an application which can be implemented to any nearby shops or branded shops selling various kinds of products by simple modifications. However, it was very challenging learning and developing an application using a new technology.

**FUTURE SCOPE**

There is a scope for further development in our project to a great extent. A number of features can be added to this system in future like providing. The feature like adding an authenticated payment system using Mpesa which is widely used in Kenya. Another feature we wished to implement was providing classes for customers so that different offers can be given to each class. System may keep track of history of purchases of each customer and provide suggestions based on their history using Machine Learning Algorithm. These features could have been implemented if time and skills did not limit me.

1. **REFERENCES**

Fernandez, A.& Anthony D. Miyazaki. Consumer Perceptions of Privacy and Security Risks for Online Shopping.

The Journal of Consumer Affairs 35.1: 27-44. Kandra, A. (2001). The Myth of Secure E-shopping. PC World July: 29-32. Halpin, M. (2011).

10 Things Your Mother Never Told You About Online Shopping. Yahoo! Internet Life: 61-63. Wu, J., & Guo, X. (2011). Online booking system design and experimental realization.

2011 International Conference on Electrical and Control Engineering. doi:10.1109/iceceng.2011.6057629 Kumar, V., & Singh, R. (2014).

Women Online Shopping: A Critical Review of Literature. SSRN Electronic Journal. doi:10.2139/ssrn.2466824