

**PROJECT REPORT**  
**ON**  
**E-Commerce Website**

submitted in  
partial fulfilment of the award of degree  
Of  
**Bachelor of Technology**  
**In**  
**COMPUTER SCIENCE & ENGINEERING**



ESTD 1998

**BATCH 2018-2022**

**Submitted by**  
**Pooja Rai (1818326)**  
**Priyanka (1818327)**

**Under the Supervision of**  
**Er Prerna Dhall**  
**H.O.D Computer Science Department**

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**  
**HARYANA ENGINEERING COLLEGE, JAGADHRI**

## **CERTIFICATE**

This is to certify that Pooja Rai, Priyanka **Roll No: 1818326, 1818327** a bonafide student of Bachelor of Technology in Computer Science Engineering being run by Haryana Engineering College, Jagadhari of batch 2018–2022 has completed his/her Project-I entitled “***E-commerce website***” at HEC, Jagadhari under the supervision of Er. Prerna Dhall.

Date.....

**Er. Prerna Dhall**

**H.O.D (CSE Dept.)**

## **CERTIFICATE**

This is to certify that **Pooja rai, Priyanka, Roll No: 1818326, 1818327** a bonafide student of Bachelor of Technology in Computer Science Engineering being run by Haryana Engineering College, Jagadhari of batch 2018 – 2022 has completed his/her Project-I entitled “**E-commerce website**” at HEC, Jagadhari under my supervision & guidance.

It is further certified that the work done in this project is a result of candidate's own efforts.

I wish him/her all success in life.

Date .....

**Er. Raman Preet Kaur**

**Assistant Professor**

**CSE department**

## **CERTIFICATE**

I/we hereby certify that the work which is presented in the Project-I entitled "***E-commerce website***" in fulfilment of the requirement for the award of the Degree of Bachelor of Technology and submitted to the Department of Computer Science & Engineering of Haryana Engineering College (HEC), Jagadhari affiliated to Kurukshetra University, Kurukshetra is an authentic record of my/our own, carried out during a period from August, 2021 to January, 2022, under the supervision of Er. Prerna Dhall.

The matter presented in this report has not been submitted by me/us for the award of any other degree of this or any other Institute/University.

**Signature:**

**Pooja Rai (1818326)**

**Priyanka (1818327)**

## **Acknowledgement**

I/we wish to express my/our deep sense of indebtedness and sincerest gratitude to my/our guide, **Er Prerna Dhall (H.O.D)**, Department of Computer Science & Engineering, Haryana engineering College, Jagadhari, Haryana, for his/her invaluable guidance and constructive criticism throughout this Project. He/she has displayed unique tolerance and understanding at every step of progress and encourages me/us. I/we deem it my/our privilege to have carried out my/our project work under his/her able guidance.

I/we would especially like to thank Er Prerna Dhall (H.O.D), Department of Computer Science & Engineering, Haryana Engineering College, Jagadhari, Haryana whom, this work would not have been as it is now.

As a Final Personal Note, I am grateful to my parents, who are inspirational to me/us in their understanding, patience and constant encouragement.

**Pooja Rai**

**Priyanka**

## **LIST OF TABLES**

<b>TABLE NO.</b>	<b>CAPTION</b>	<b>PAGE NO.</b>
2.1	SQLite features and Description	26
2.2	DDL Command	27
2.3	DML Command	27
2.4	DQL Commands	28
3.1	Required Hardware	36
3.2	Required Software	36

## **LIST OF FIGURES**

Fig. 1.1 E-Commerce.....	1
Fig. 1.2 Types of E-Commerce.....	2
Fig. 2.1 Frontend and Backend développent.....	7
Fig. 2.2 HTML Code Format.....	8
Fig. 2.3 CSS Code Example.....	9
Fig. 2.4 JavaScript Example.....	10
Fig. 2.5 Bootstrap 4 Logo.....	12
Fig. 2.6 Python logo.....	13
Fig. 2.7 Python Rating.....	15
Fig. 2.8 Google.....	17
Fig. 2.9 Facebook Content.....	18
Fig. 2.10 Dropbox View.....	19
Fig. 2.11 Django Logo.....	19
Fig. 2.12 Disques View.....	22
Fig. 2.13 Instagram View.....	23
Fig. 2.14 SQLite Logo.....	24
Fig. 2.14 Razorpay Logo.....	29
Fig. 3.1 Wrapper Module.....	37
Fig. 3.2 Main Module.....	38
Fig. 3.3 Saler Module.....	38
Fig. 3.4 Entity Relationship Diagram.....	39
Fig. 3.5 Flow Chart.....	40

Fig. 4.1 Login Page.....	41
Fig. 4.2 Sign Up Page.....	41
Fig. 4.3 Seller Registration Page.....	42
Fig. 4.4 User Home Page.....	42
Fig. 4.5 Catégories Filtration Sidé Bar.....	43
Fig. 4.6 Product List.....	43
Fig. 4.7 Particular Product Details Page.....	44
Fig. 4.8 Size sélection Buttons.....	44
Fig. 4.9 Checkout Page.....	45
Fig. 4.10 Payments Page.....	45
Fig. 4.11 Razorpay Gateway Login page.....	46
Fig. 4.12 Online Payments Mode Selection.....	46
Fig. 4.13 Payments Success/Failure Page.....	46
Fig. 4.14 Payments Confirmation Page.....	47
Fig. 4.15 User Carto View.....	48
Fig. 4.16 Place Order Page.....	48
Fig. 4.17 Order Placed Conformation.....	49
Fig. 4.18 Placed Ordres Statues.....	49
Fig. 4.19 Delivered Product List.....	50
Fig. 4.20 Cancelled Product List.....	50
Fig. 4.21 User Profile Updations Page.....	51
Fig. 4.22 User Password Updations Page.....	52
Fig. 4.23 Seller Home Page.....	52
Fig. 4.24 Seller Sidebar.....	53

Fig. 4.25 Seller View Ordres.....	53
Fig. 4.26 Seller View Delivered Product.....	54
Fig. 4.27 Seller Contact Page.....	54
Fig. 4.28 Seller Addo New Product Page.....	55
Fig. 4.29 Seller Product List.....	55
Fig. 4.30 Seller Profile Updations Page.....	56
Fig. 4.31 Seller Password Updations Page.....	56
Fig. 4.32 Seller Account Setting.....	57
Fig. 4.34 Seller Carto View.....	57
Fig. 4.35 Seller Checkout Page.....	58
Fig. 4.36 Seller Order Statues Page.....	58
Fig. 4.37 Admin Page.....	58
Fig. 4.38 Crude Operations at Admin Site.....	59

## **ABSTRACT**

Online Shopping play a great importance in the modern business environment. WRAPPER-INDIA-ONLINE-MASTER e-commerce website has opened the door of opportunity and advantage to the firms. This paper analyzed the different issue of online shopping. The research aims to provide theoretical contribution in understanding the present status of online shopping. The Study Discuss the consumers' online shopping behaviors. Paper also identifies the problems face by the consumers when they want to accept internet shopping. Present paper is an expressive study based on the detailed review of earlier pertinent studies related to the various concepts of online shopping to discover the concept of online shopping. Solitude and safety risk emerge regularly as a reason for being cautious about internet shopping. Shopping convenience, information seeking, social contact, and diversity affects the consumer attitude towards online shopping. The impossibility of product testing, problems with complaints, product return and missus of personal data are the main doubts regarding on-line shopping

Keywords:

**Python, Django, SQL-lite3, Bootstrap 3, CSS, HTML, ecommerce, customer orders  
Fashion Shop**

# **CONTENTS**

<b>Chairperson's Declaration Certificate.....</b>	<b>i</b>
<b>Guide's Declaration Certificate .....</b>	<b>ii</b>
<b>Candidate's Declaration Certificate.....</b>	<b>iii</b>
<b>Acknowledgement.....</b>	<b>iv</b>
<b>List of Tables.....</b>	<b>v</b>
<b>List of Figures.....</b>	<b>vi-viii</b>

## **ABSTRACT**

<b>Chapter 1: INTRODUCTION.....</b>	<b>1-6</b>
1.1 What is Ecommerce.....	1-2
1.1.1 Types of Ecommerce Models.....	2-3
1.1.2 History of Ecommerce.....	3
1.1.3 Advantages of E-Commerce.....	3-4
1.1.4 Disadvantages of E-Commerce.....	4-5
1.2 Statement of The Problem.....	5
1.3 The Solution.....	5-6
1.4 Aims and Objectives.....	6
<b>Chapter 2 SYSTEM DESIGN.....</b>	<b>7-31</b>
2.1 Web Tools and Techniques.....	8-31
2.1.1 Frontend web development.....	7-13
2.1.1.1 HTML.....	8-9
2.1.1.2 CSS (Cascading style sheets) .....	9-10
2.1.1.3 JavaScript.....	10-11

2.1.1.4 Bootstrap 4.....	11-13
2.1.2 Python.....	13-19
2.1.2.1 Why is Python good for web development.....	14-16
2.1.2.2 Who uses Python.....	16-19
2.1.3 Django Framework.....	19-24
2.1.3.1 What's the difference between core Python and Django.....	20
2.1.3.2 Why is Django good for web development.....	20-22
2.1.3.3 Who uses Django.....	22-24
2.1.4 SQLite.....	24-31
2.1.4.1 Why SQLite.....	24-25
2.1.4.2 SQLite A Brief History.....	25
2.1.4.3 SQLite Limitations.....	25-26
2.1.4.4 SQLite Commands.....	26-28
2.1.5 Razor pay Payment Gateway.....	28-31
2.1.5.1 How does Razor pay Work.....	29-31
<b>Chapter 3: PROJECT DETAIL WITH REQUIREMENTS.....</b>	<b>32-40</b>
3.1 Objective.....	32
3.2 Major Functionalities Used in E-commerce Website.....	32-35
3.2.1 Features of E-Commerce Website in admin side.....	32-33
3.2.2 E-Commerce Website User front end side.....	33-34
3.2.3 E-Commerce Website Seller front end side.....	34-35
3.3 System specifications.....	35-36
3.3.1 Hardware requirement.....	35-36
3.3.2 Software requirement.....	36

3.4 The Steps to Run a Django E-Commerce Website Project with Source Code.....	36-39
3.5 Entity Relationship Diagram.....	39
3.4 Flow Chart.....	40
<b>Chapter 4: RESULTS.....</b>	<b>41-59</b>
<b>Chapter 5: CONCLUSION &amp; FUTURE WORK.....</b>	<b>60-61</b>
References.....	62
<b>PARTIAL CODE.....</b>	<b>63-65</b>

# Chapter 1

## INTRODUCTION

### 1.1 What is Ecommerce:

The term electronic commerce (e-commerce) refers to a business model that allows companies and individuals to buy and sell goods and services over the Internet. E-Commerce operates in four major market segments and can be conducted over computers, tablets, smartphones, and other smart devices.



**Figure :1.1 E-Commerce**

- Ecommerce is the buying and selling of goods and services over the Internet.
- It is conducted over computers, tablets, smartphones, and other smart devices.
- Almost anything can be purchased through ecommerce today.
- It can be a substitute for brick-and-mortar stores, though some businesses choose to maintain both.

### Understanding E-Commerce:

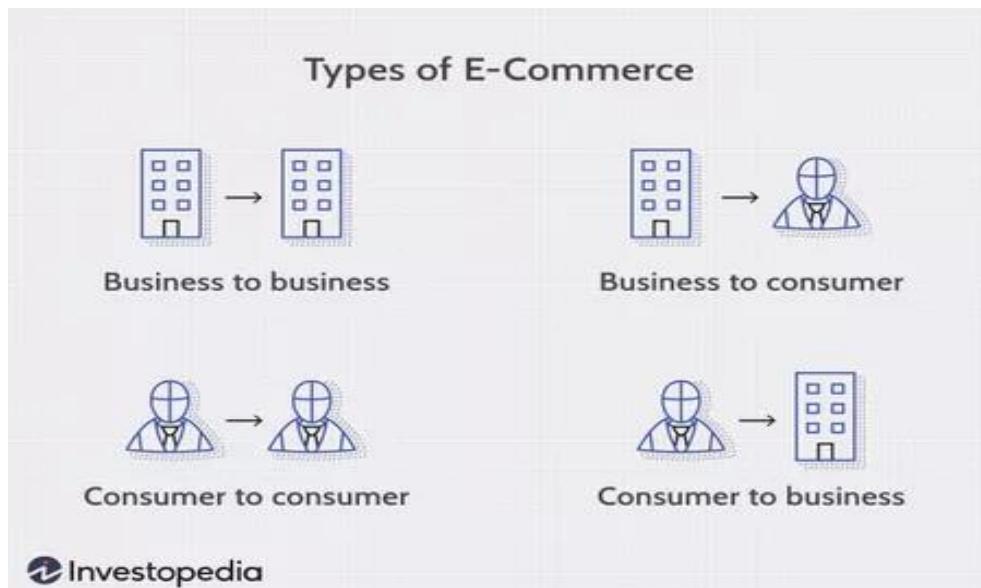
As noted above, ecommerce is the process of buying and selling tangible products and services online. It involves more than one party along with the exchange of data or currency

to process a transaction. It is part of the greater industry that is known as electronic business (business), which involves all of the processes required to run a company online.

Ecommerce has helped businesses (especially those with a narrow reach like small businesses) gain access to and establish a wider market presence by providing cheaper and more efficient distribution channels for their products or services.

### **1.1.1 Types of Ecommerce Models:**

There are four main types of ecommerce models that can describe almost every transaction that takes place between consumers and businesses.



**Figure :1.2 Types of E-Commerce**

#### **1. Business to Consumer (B2C):**

When a business sells a good or service to an individual consumer (e.g., You buy a pair of shoes from an online retailer).

#### **2. Business to Business (B2B):**

When a business sells a good or service to another business (e.g., A business sells software-as-a-service for other businesses to use)

### **3. Consumer to Consumer (C2C):**

When a consumer sells a good or service to another consumer (e.g., You sell your old furniture on eBay to another consumer).

### **4. Consumer to Business (C2B):**

When a consumer sells their own products or services to a business or organization (e.g., An influencer offers exposure to their online audience in exchange for a fee, or a photographer licenses their photo for a business to use).

#### **1.1.2 History of Ecommerce:**

The history of ecommerce begins with the first ever online sale: on the August 11, 1994 a man sold a CD by the band Sting to his friend through his website Net Market, an American retail platform. This is the first example of a consumer purchasing a product from a business through the World Wide Web—or “ecommerce” as we commonly know it today.

Since then, ecommerce has evolved to make products easier to discover and purchase through online retailers and marketplaces. Independent freelancers, small businesses, and large corporations have all benefited from ecommerce, which enables them to sell their goods and services at a scale that was not possible with traditional offline retail.

Global retail ecommerce sales are projected to reach \$27 trillion by 2020

#### **1.1.3 Advantages of E-Commerce:**

Benefits of e-commerce include its around-the-clock availability, the speed of access, the wide availability of goods and services for the consumer, easy accessibility and international reach.

- **Availability.** Aside from outages or scheduled maintenance, e-commerce sites are available 24x7, allowing visitors to browse and shop at any time. Brick-and-mortar businesses tend to open for a fixed number of hours and may even close entirely on certain days.

- **Speed of access.** While shoppers in a physical store can be slowed by crowds, e-commerce sites run quickly, which is determined by compute and bandwidth considerations on both consumer device and e-commerce site. Product pages and shopping cart pages load in a few seconds or less. An e-commerce transaction can comprise a few clicks and take less than five minutes.
- **Wide availability.** Amazon's first slogan was "Earth's Biggest Bookstore." They could make this claim because they were an e-commerce site and not a physical store that had to stock each book on its shelves. E-commerce enables brands to make a wide array of products available, which are then shipped from a warehouse after a purchase is made. Customers will likely have more success finding what they want.
- **Easy accessibility.** Customers shopping a physical store may have a hard time determining which aisle a particular product is in. In e-commerce, visitors can browse product category pages and use the site search feature to find the product immediately.
- **International reach.** Brick-and-mortar businesses sell to customers who physically visit their stores. With e-commerce, businesses can sell to any customer who can access the web. E-commerce has the potential to extend a business' customer base
- **Lower cost.** pure play e-commerce businesses avoid the cost associated with physical stores, such as rent, inventory and cashiers, although they may incur shipping and warehouse costs.
- **Personalization and product recommendations.** E-commerce sites can track visitors' browse, search and purchase history. They can use this data to present useful and personalized product recommendations, and obtain valuable insights about target markets. Examples include the sections of Amazon product pages labeled "Frequently bought together" and "Customers who viewed this item also viewed."

#### **1.1.4 Disadvantages of E-Commerce:**

- **Limited customer service.** If a customer has a question or issue in a physical store, he or she can see a clerk, cashier or store manager for help. In an e-commerce store, customer service may be limited: The site may only provide support during certain hours of the day, or a call to a customer service phone number may keep the customer on hold.

- **Not being able to touch or see.** While images on a webpage can provide a good sense about a product, it's different from experiencing it "directly," such as playing music on speakers, assessing the picture quality of a television or trying on a shirt or dress. E-commerce can lead consumers to receive products that differ from their expectations, which leads to returns. In some scenarios, the customer bears the burden for the cost of shipping the returned item to the retailer.
- **Wait time.** If a customer sees an item that he or she likes in a store, the customer pays for it and then goes home with it. With e-commerce, there is a wait time for the product to be shipped to the customer's address. Although shipping windows are decreasing as next day delivery is now quite common, it's not instantaneous.
- **Security.** Skilled hackers can create authentic-looking websites that claim to sell well-known products. Instead, the site sends customers forfeit or imitation versions of those products -- or, simply collects customers' credit card information. Legitimate e-commerce sites also carry risk, especially when customers store their credit card information with the retailer to make future purchases easier. If the retailer's site is hacked, hackers may come into the possession of customers' credit card information.

## 1.2 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 move 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. In addition, everyone tends to use social media, posing a great challenge to differentiate between scammers (fraudsters) and legit sellers.

## **1.3 The Solution**

WRAPPER-INDIA-ONLINE-MASTER e-commerce website 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.

## **1.4 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:

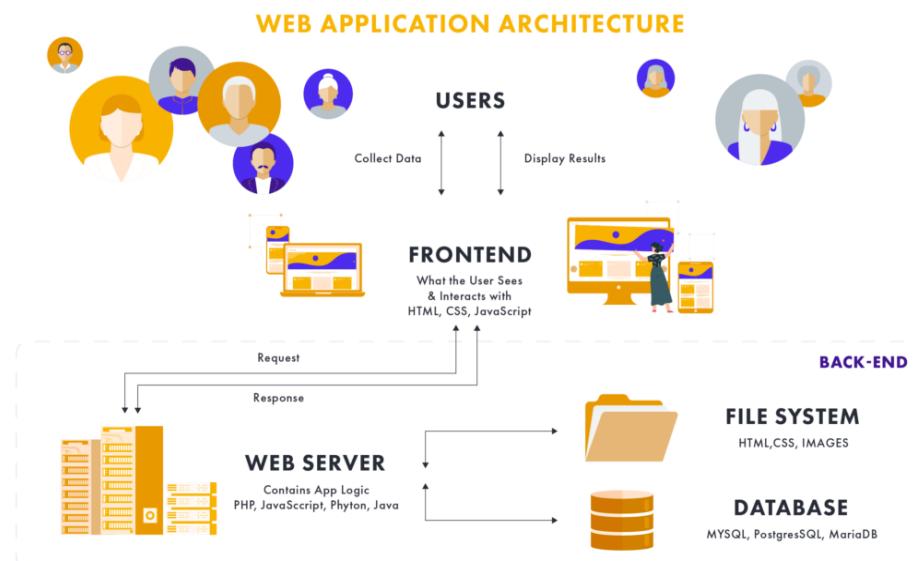
- To design an online fashion system.
- To provides a solution to reduce and optimize the expenses of customer order management
- To create an avenue where people can shop for fashion products online.
- To develop a database to store information on fashion products and

# Chapter 2

## SYSTEM DESIGN

### 2.1 Web Tools and Techniques

Rapid development of web tools and technologies bring more conveniences in our daily life. Millions of data can be transmitted easily from one place to another with the help of internet. In one-way technology development has not only simplified our daily lives but also has reduced cost and time in general. Online business has changed traditional method of buying and selling products. People save time and effort with wide variety of products available 24/7. E-commerce website should be attractive and always updated with all new techniques for better profit. For Our E-commerce website, we have used various tools and technologies such as Notepad++ for programming, chrome, HTML, CSS, JavaScript, Bootstrap 4, Ajax, SQL database which are described below.



**Figure:2.1 Front-end and Back-end development**

#### 2.1.1 Frontend web development:

Front end refers to anything that a user sees first in their application or a webpage such as form fonts, colors, side bar and dropdown menu. The look and feel of a site are the main responsibilities of frontend developers. Front end web development is also known as client side of development. HTML, CSS and JavaScript are Front end developing languages

whereas PHP, Java, Python and .NET are Back-end web development. Backend development makes server, database and application communicate with each other with the help of server-side languages such as PHP, Ruby, Python and Java.

### **2.1.1.1 HTML:**

HTML describes the structure of web page. It stands for Hyper Text Markup language. In 1989, HTML was created by Tim Berners-Lee, Robert Calliou and others. HTML2.0 was published in late 1999. HTML5 is the latest version of HTML. HTML is one of the most widely used languages over web to display documents. HTML is the basic block whereas other technologies besides HTML such as CSS are for decoration of page and JavaScript is for more functionality and behavior of web page. When users save file name with HTML extension and views through browser such as Internet explorer, browser reads files and translates code into visible form and displays designed web page.

```
<!DOCTYPE html>
<html>

    <head>
        <title> Title here </title>
    </head>

    <body>
        Web page content goes here.
    </body>

</html>
```

**Figure:2.2 HTML Code Format**

DOCTYPE declaration must be in the first line of web document which tells browser that codes are written in HTML5. HTML tag is container for whole HTML document and tells browser in which language document is written. Head tag especially works behind the scene and contains information about documents. Only few elements inside head tag are displayed in web pages. Charset attribute specifies character encoding for HTML document. UTF-8 can

support almost all the character and symbol inside HTML document. Title tag are displayed on search engine of browser and defines title of document. Body tag is visible part and define the document body. Body element is the actual content of document. It contains different text images, videos, links and other. HTML element contains start tag and end tag and contents are written in between.

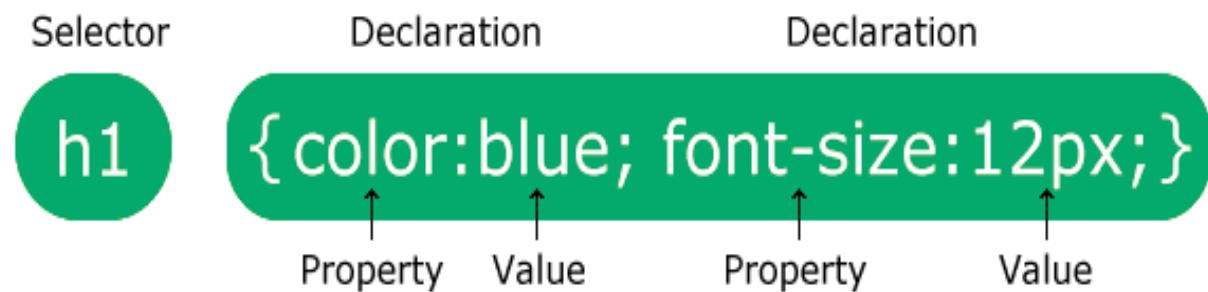
Example:

<Tag name> content goes here </tag name>

<h1> Heading 1 </h1> (heading tag).

### 2.1.1.2 CSS (Cascading style sheets):

Cascading style sheets is a mechanism for adding styles such as font spacing, colors to web documents written in HTML. CSS reduce work and save time .CSS can change layout of multiple web pages just by changing styling one page. CSS is a prominent technology of the worldwide web styles for different rendering method like onscreen, in print, by voice and Braille based tactile devices. CSS also has rules for alternate formatting for the content accessed on mobile devices. In 1994 October 10, Haakon Wium Lie introduces CSS. There are three ways of inserting CSS on web page: Inline CSS, External CSS, and Internal CSS.



**Figure:2.3 CSS Code Example**

In Inline style sheet CSS code is written in same line for single element as shown below in example:

<p style = “font-size: 25px; color: red; text-align: center;”> this is example of Inline CSS</p>

In External style sheet, the code is written in another file and saved with CSS extension as shown below.

```
<!DOCTYPE HTML>

<HTML>

<head>

<link rel =”stylesheet” type =“text/css” href =“external style file.css”>

</head>.
```

### 2.1.1.3 JavaScript:

JavaScript (JS) is a dynamic, weakly typed programming language which is compiled at run time. JavaScript is one the most popular programming languages. Enormous companies like Netflix, Walmart and PayPal build entire application with JavaScript. It can be executed as a part of web page in browser. Java script was invented to make webpage more dynamic and to change the content on page directly from inside the browser. Real time networking applications like chats and video streaming services command-line tools and game can be built with JavaScript. In 1995 Netscape introduced Live Script but due to the popularity of Java it was renamed as JavaScript.



**Figure:2.4 JavaScript Example**

However, it is totally independent from Java and has nothing in common with Java. JavaScript was mainly used for a spamming, annoying overlays and pop-ups but another

problem was writing different scripts for different browsers. In late 1996 people noticed fragmentation could be a problem and JavaScript language was submitted to ECMA committee to start standardization. New features are being added and it is getting better and better programming languages. It is interpreted language which means it is compiled before it runs and not compiled during development. JavaScript was originally designed to run only in browser.

Every browser has JavaScript engine to execute JavaScript code. JavaScript engine reads and understands JavaScript code then compile it to machine code because machine code executes faster and execute machine codes. Google's JavaScript was extracted to run JavaScript anywhere and many non-browser environments such as Node.js, Adobe Acrobat use JavaScript. JavaScript can manipulate the HTML code, CSS, send background HTTP requests. In HTML, JavaScript code must be between `<script>` and `</script>` tag.

```
<HTML>
<body >
<script>
alert("Hello JavaScript")
</script>
<p> Introduction to JavaScript </p>
</body >
</HTML>
```

#### **2.1.1.4 Bootstrap 4:**

Bootstrap 4 is the newest version of Bootstrap. Bootstrap can be defined as a free and open-source framework that can be used to create responsive, mobile-first, front-end web pages.

#### **Why should one use Bootstrap?**

There are multiple reasons for that, such as:

- It is a way more fast and simpler way to develop websites.
- It can create platform-independent web pages.
- It can also create responsive web pages.

- The web pages created by using Bootstrap 4 can be easily made responsive.



**Figure:2.5 Bootstrap 4 Logo**

### **What are some key usages of Bootstrap 4 in today's world?**

There are various key usages of Bootstrap 4, the examples are given below:

1. Supported by various Browsers: It can be supported by every browser.
2. Simple to start and implement: It is very easy to start and implement when the user has a fair amount of knowledge about HTML and CSS. In addition to that, the documentation is readily available on the official site.
3. Responsive design and looks: The web pages that are created by using the Bootstrap framework are responsive and it can adapt to any screen size like mobile, desktop, etc.
4. Easily Customized: It also provides some built-in components and functionalities that can be used for the purpose of easily customizing the web pages.

### **Advantages of Bootstrap 4:**

There are various different advantages of Bootstrap, a few of them are given below:

- The first and foremost advantage of using Bootstrap is that it is very easy to use and implement. If a person has some basic knowledge of HTML and CSS, that user can easily use Bootstrap.

- The fact that Bootstrap can adapt to the size of any phone, tablet, desktop and so on is also very interesting feature.
- Bootstrap 4 is also useful because it is compatible with all modern browsers which include Google Chrome, Firefox, Internet Explorer 10+, Edge, Safari, and Opera.
- It also produces less cross-browser bugs.
- It is light weighted and consequently it can be widely used as a framework for creating responsive sites.
- Lastly, Bootstrap 4 is a very simple and yet very effective grid system.

#### **Disadvantages of bootstrap 4:**

Bootstrap 4 also has many disadvantages; those disadvantages are given below:

1. In many cases, Bootstrap cannot be considered very practical for businesses that need a big amount of investment.
2. Moreover, Bootstrap 4 can take a lot of time to create a website. Therefore, it is not a very bright idea to use bootstrap 4 when there is no investment.
3. A person using Bootstrap 4 is not likely to earn any money even after investment.
4. And this is one of the reasons why a user can very easily end up in a lot of debt.

#### **2.1.2 Python**

- According to [the Python Software Foundation](https://www.python.org/), “Python is an interpreted, object-oriented, high-level programming language with dynamic semantics.”



**Figure:2.6 Python logo**

- Python is a high-level programming language, meaning it contains words and phrases comprehensible to humans. To translate this high-level language into machine code, Python uses an interpreter. An interpreter converts source code into code understood by computers. As an interpreted language, Python uses memory efficiently, is easy to debug, and allows developers to perform complex tasks in a couple of steps and edit code quickly.
- Most modern programming languages, including Python, are considered object-oriented. The idea behind object-oriented languages is that the program is split into independent objects that communicate between themselves. This pretty much looks like a collection of mini-programs. Objects help developers write better structured code, resulting in software that's easier and faster to build, maintain, and debug.
- Python is a dynamic language. This means that developers don't have to declare variable types. Python verifies types and errors at runtime (when the program is launched), but not during compilation (when source code is converted to machine code). This results in less code, faster development, and greater flexibility and resilience.

#### **2.1.2.1 Why is Python good for web development?**

Python continues to be one of the most popular languages. According to [TIOBE](#), Python is ranked among the top programming languages. [Built With](#) states that there are almost a million websites that use Python.

##### **1. Powerful and fast**

Python is powerful, and it can be used for literally anything. First and foremost, it can be used for anything because it offers many features by default, with standard libraries that cover almost any programming task. Whether you're running scientific calculations, processing images, or developing operating system interfaces or protocols, Python saves time and effort by presenting ready-made solutions.

Mar 2019	Mar 2018	Change	Programming Language	Ratings	Change
1	1		Java	14.880%	-0.06%
2	2		C	13.305%	+0.55%
3	4	▲	Python	8.262%	+2.39%
4	3	▼	C++	8.126%	+1.67%
5	6	▲	Visual Basic .NET	6.429%	+2.34%
6	5	▼	C#	3.267%	-1.80%
7	8	▲	JavaScript	2.426%	-1.49%
8	7	▼	PHP	2.420%	-1.59%
9	10	▲	SQL	1.926%	-0.76%
10	14	▲	Objective-C	1.681%	-0.09%
11	18	▲	MATLAB	1.469%	+0.06%
12	16	▲	Assembly language	1.413%	-0.29%
13	11	▼	Perl	1.302%	-0.93%
14	20	▲	R	1.278%	+0.15%
15	9	▼	Ruby	1.202%	-1.54%
16	60	▲	Groovy	1.178%	+1.04%
17	12	▼	Swift	1.158%	-0.99%
18	17	▼	Go	1.016%	-0.43%
19	13	▼	Delphi/Object Pascal	1.012%	-0.78%
20	15	▼	Visual Basic	0.954%	-0.79%

**Figure:2.7 Python Rating**

Additionally, Python serves well for products that you want to get to market as soon as possible. For both startups and established businesses, this means faster return on investment (ROI) and a chance to tailor the product based on reviews and feedback from real users.

## 2. Plays well with others

Python is functional enough to interface with code written in other programming languages. You can embed your Python project into frameworks of different languages, and vice versa. This means that you can create projects that mix Python with other programming languages (for example, C++) and get the best of two worlds.

### **3. Runs everywhere**

Python is available for almost any operating system, including UNIX-based operating systems, Windows, macOS, iOS, and Android. It also runs on various platforms, including IBM, AIX, Solaris, and VMS.

### **4. Friendly and easy to learn**

Python is considered a highly effective way of coding because of its simple syntax and readability. Python is an easy-to-learn language, and thus there's a large enough talent pool.

Because Python uses spacing instead of braces and other special characters to indicate the beginning and end of programming constructs, readability is a built-in feature. Whenever you need to update a project in Python, it's relatively easy to find someone who can look at its code and make the necessary changes even if they've never seen it before. Developers don't have to spend a great deal of time thinking about complexities in the language or deciphering someone else's code.

### **5. Open**

Python is an open-source language administered and supported by an independent nonprofit foundation: the Python Software Foundation. One of the major advantages of open-source software is that it's free to use, modify, and distribute. Since the Python community is open to everyone and encourages diversity, millions of experienced developers around the world contribute to it and add to the growing Python open-source database. There are plenty of conferences and meetups, and there's lots of code collaboration. This means that Python is less inclined to bugs and more secure than some other popular languages. Additionally, it helps you minimize development expenses.

#### **2.1.2.2 Who uses Python?**

Python is a multi-purpose language and can be used to build practically anything. Companies around the world [use Python for artificial intelligence and machine learning](#), website development, scientific and numerical computing, gaming, and many other uses.

As for artificial intelligence, Python stands above other programming languages and is considered the best programming language for AI-powered applications. In fact, Python

addresses any AI task: [machine learning](#), data analysis and visualization, natural language processing, and computer vision.

There are dozens of companies that use Python in one form or another for its flexibility, scalability, performance, and rapid development. Below are examples of businesses that use Python extensively.

- **Google**

Python has been an important part of Google since the beginning, and remains so as the system grows and evolves. Today dozens of Google engineers use Python, and we're looking for more people with skills in this language.

**Peter Norvig, Google's Research Director**

One of the largest companies that relies on Python is Google. The first version of the Google search engine and the entire technology stack were actually written in Python. Today, Python is one of the three main programming languages used by Google, together with Java and C++.

Google has a finger in every pie: search, email, music and video streaming, hardware, driverless cars, and advertising. In fact, YouTube's entire technology stack is written in Python. Originally, YouTube was actually written in PHP. However, because of fast user growth and the need to quickly develop and implement new features, YouTube soon migrated to Python. Cuong Do — a software architect at YouTube — says that “Python is fast enough for our site and allows us to produce maintainable features in record times, with a minimum of developers.”

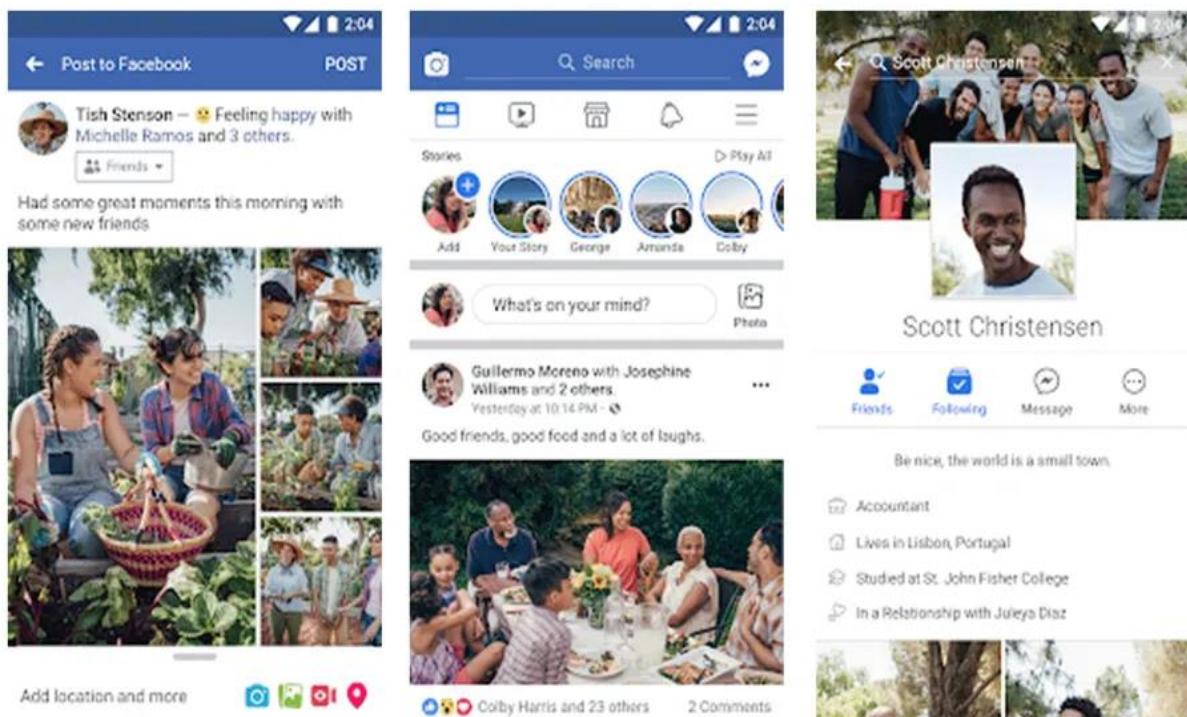


**Figure:2.8 Google**

- **Facebook**

Another big player that uses Python is Facebook. Alongside PHP and C++, Python helps Facebook maintain, manage, and scale their infrastructure efficiently and reliably.

Facebook also uses Python for machine learning. On Facebook, machine learning algorithms are used to curate content in the News Feed and to detect objects in photos. Facebook users are recognized in pictures even when they aren't tagged. And for people with vision impairments, Facebook describes photos with words.

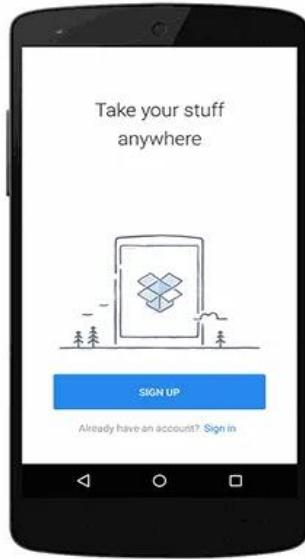


**Figure:2.9 Facebook Content**

- **Dropbox**

Dropbox is a cloud storage platform for documents, music, images, video, and more. The Dropbox application is available on most devices and offers users anywhere, anytime access. Dropbox has been using Python for the entire technology stack right from the start. First, it was Python code using PyWin32: one of the Python libraries for Windows-specific functionality. Later on, they moved to Python 2, and then to Python 3. At present, the Dropbox application uses a customized version of Python 3.5. They also use the Django

framework, which powers file storage functionality, account synchronization, and file sharing.



**Figure:2.10 Dropbox View**

### **2.1.3 Django Framework:**

Django is a free, open-source, full-stack web application framework written in Python. In simple words, Django is a set of ready-made components that help you rapidly build websites with pragmatic and clean architectures. When crafting websites, developers frequently need the same components for tasks like authenticating users, uploading files, and implementing CMS systems. So as not to reinvent the wheel each time you make a new site, Django provides ready-made solutions.



**Figure:2.11 Django Logo**

### **2.1.3.1 What's the difference between core Python and Django?**

Python and Django are intertwined but not the same. Python is a programming language that's used for many different applications: artificial intelligence, machine learning, desktop apps, etc. On the other hand, Django is a Python framework for full-stack web application development and server development.

You can build a website either from scratch in core Python, arriving at the solution yourself, or you can craft your app with Django using prewritten bits of code.

### **2.1.3.2 Why is Django good for web development?**

Among Python frameworks, Django is often regarded as the best framework for web application development. It was even included in the list of [most loved frameworks in the 2018 Stack Overflow Developer Survey](#).

According to the Django Software Foundation, Django was “invented to meet fast-moving newsroom deadlines, while satisfying the tough requirements of experienced web developers.” The framework was invented by Adrian Holovaty and Simon Willison, developers who worked at the news website World Online. At the time, Adrian and Simon were writing in PHP. But since the World Online website required quick updating, it became hard to meet deadlines. Adrian and Simon needed something they could build the website with quickly. That was when they decided to move to Python. Later on, they started working on Django to make coding even simpler and more effective. In 2005, Django was already publicly released as an open-source project. Today, Django is known as **The web framework for perfectionists with deadlines**.

#### **1. For perfectionists**

Thanks to simple syntax and readable code, programs written in Django are clean and well-structured. This help developers make their ideas a reality and meet requirements faster.

#### **2. With deadlines**

Django was crafted to help build software as quickly as possible. The Don't Repeat Yourself (DRY) philosophy on which Django is built motivates developers to reuse code they've written for other projects, further reducing the time needed to develop your product. So, if

you're short on time and have a limited budget or if you often work tight deadlines and place a high premium on innovation, then Django is a perfect solution.

These aren't the only things that make Django your top choice for web development. There are even more great features that the [Django web framework](#) provides.

### **3. Batteries included**

Django's batteries included approach means that the framework has everything necessary to develop a full-fledged application. Django offers built-in HTML templating, URL routing, object-relational mapping, and session management, helping developers avoid the vexing search for third-party tools. Hundreds of Django packages help you quickly create anything from [chatbots](#) to [complex GPS-enabled solutions](#).

### **4. Extends and scales with ease**

Django components are decoupled, meaning they can be added or removed as necessary just like Lego pieces. Depending on your specific product requirements, development can be scaled up or down, altering the number and complexity of Django components as necessary.

Read more: [Best practices for working with Django models in Python](#)

### **5. Offers robust security**

By default, Django protects applications. It mitigates some of the more common security mistakes related to Python while also protecting apps from cross-site scripting, request forgery, and SQL injection.

### **6. Works with most major databases**

Django's object-relational mapping (ORM) is compatible with a number of popular databases, but its key feature is that it allows developers to work with several databases at once. Moreover, Django makes it possible to migrate from one database to another and perform common operations without having to write much additional code.

### **7. Helps with search engine optimization**

Among other things, Django offers a number of useful search engine optimization (SEO) tools. With the [Django SEO framework](#), developers can reduce page loading time by using [cached templates](#) and [compressing CSS and JavaScript](#). There's even [a tool to manage robots.txt](#). All in all, the capabilities of the [Django content management system](#) exceed the

expectations of many SEO specialists, proving that Django is one of the best frameworks for search engine optimization.

#### *Constantly evolves thanks to its community*

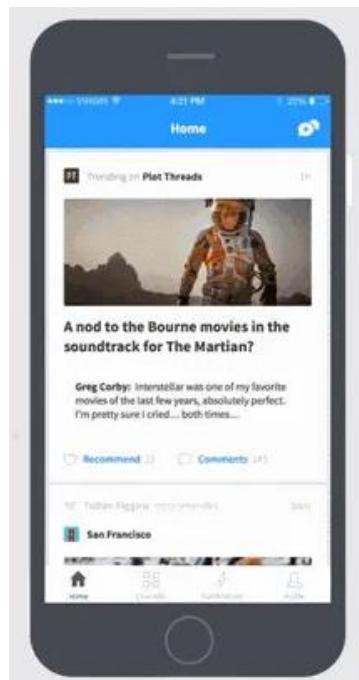
There's a large community around Django, with more enthusiasts joining every day. They constantly update and improve components of the framework while also developing new libraries to solve issues that professionals often face when developing web applications.

### **2.1.3.3 Who uses Django?**

Below, we've listed some of the major companies that benefit from this technology. Let's have a look at how Django has helped these companies grow and develop.

- **Disqus**

Disqus is a third-party comment application for blogs and online communities. It helps with user retention and audience analytics, provides advertising opportunities, and offers tools for user engagement. When starting off, Disqus considered various architectures but opted for Python, Django, Thoonk, and Nginx Push Stream Module. On Disqus, Django enables custom analytics, social media integration, and online presence management.



**Figure:2.12 Disqus View**

Disqus used the Django framework not only for development and scaling but also to create Sentry — a side project initially for internal use. Later, Sentry became an open-source application that helps businesses track and debug errors and fix crashes in their systems. Today, more than 30,000 companies including Door dash, Instagram, Reddit, Prezi, and Datadog use Sentry.

- **Instagram**

The Django web framework is popular among social networking platforms, since it can handle large volumes of data and manage a significant number of user interactions per minute.

With more than one billion active monthly users, over 500 million stories posted every day, and 100+ million photos uploaded daily, Instagram needs to ensure maximum efficiency so the platform can keep on scaling smoothly without sacrificing quality. Django is what helps them handle this. In fact, Instagram has the biggest deployment of Django in the world.



**Figure:2.13 Instagram View**

As Instagram grew, they started considering a new technology stack. Yet after carefully weighing the pros and cons, they decided to stick to Python and Django. In an interview, Hui

Ding, the former head of the Instagram engineering team, says that “we had been able to get to a few hundred million users with our Python/Django stack, so we decided we would continue. Also significant in the decision was that our engineers really love Python. It’s actually a reason people want to come work for us.”

## 2.1.4 SQLite

SQLite is an in-process library that implements a self-contained, serverless, zero-configuration, transactional SQL database engine. It is a database, which is zero-configured, which means like other databases you do not need to configure it in your system.



**Figure:2.14 SQLite Logo**

SQLite engine is not a standalone process like other databases, you can link it statically or dynamically as per your requirement with your application. SQLite accesses its storage files directly.

### 2.1.4.1 Why SQLite?

- SQLite does not require a separate server process or system to operate (serverless).
- SQLite comes with zero-configuration, which means no setup or administration needed.
- A complete SQLite database is stored in a single cross-platform disk file.
- SQLite is very small and light weight, less than 400KiB fully configured or less than 250KiB with optional features omitted.
- SQLite is self-contained, which means no external dependencies.

- SQLite transactions are fully ACID-compliant, allowing safe access from multiple processes or threads.
- SQLite supports most of the query language features found in SQL92 (SQL2) standard.
- SQLite is written in ANSI-C and provides simple and easy-to-use API.
- SQLite is available on UNIX (Linux, Mac OS-X, Android, iOS) and Windows (Win32, WinCE, WinRT).

#### **2.1.4.2 SQLite A Brief History**

- 2000 - D. Richard Hip designed SQLite for the purpose of no administration required for operating a program.
- 2000 - In August, SQLite 1.0 released with GNU Database Manager.
- 2011 - Hip announced to add UNQL interface to SQLite DB and to develop UNQLite (Document oriented database).

#### **2.1.4.3 SQLite Limitations**

There are few unsupported features of SQL92 in SQLite which are listed in the following table.

Sr.No.	Feature & Description
1	<b>RIGHT OUTER JOIN</b> Only LEFT OUTER JOIN is implemented
2	<b>FULL OUTER JOIN</b> Only LEFT OUTER JOIN is implemented.
3	<b>ALTER TABLE</b> The RENAME TABLE and ADD COLUMN variants of the ALTER TABLE command are supported. The DROP COLUMN, ALTER COLUMN, ADD

	CONSTRAINT are not supported
4	<p><b>Trigger support</b></p> <p>FOR EACH ROW triggers are supported but not FOR EACH STATEMENT triggers</p>
5	<p><b>VIEWS</b></p> <p>VIEWS in SQLite are read-only. You may not execute a DELETE, INSERT, or UPDATE statement on a view.</p>
6	<p><b>GRANT and REVOKE</b></p> <p>The only access permissions that can be applied are the normal file access permissions of the underlying operating system.</p>

**Table : 2.1 SQLite features and Description**

#### 2.1.4.4 SQLite Commands

The standard SQLite commands to interact with relational databases are similar to SQL. They are CREATE, SELECT, INSERT, UPDATE, DELETE and DROP. These commands can be classified into groups based on their operational nature –

- **DDL - Data Definition Language**

Sr.No.	Command & Description
1	<b>CREATE</b>

	Creates a new table, a view of a table, or other object in database.
2	<b>ALTER</b>  Modifies an existing database object, such as a table.
3	<b>DROP</b>  Deletes an entire table, a view of a table or other object in the database.

**Table : 2.2 DDL Commands**

- **DML - Data Manipulation Language**

Sr.No.	Command & Description
1	<b>INSERT</b>  Creates a record
2	<b>UPDATE</b>  Modifies records
3	<b>DELETE</b>  Deletes records

**Table : 2.3 DML Commands**

- **DQL - Data Query Language**

Sr.No.	Command & Description
1	<p><b>SELECT</b></p> <p>Retrieves certain records from one or more tables</p>

**Table : 2.4 DQL Commands**

#### **Frontend and backend web development:-**

Front end refers to anything that a user sees first in their application or a webpage such as form fonts, colors, side bar and dropdown menu. The look and feel of a site are the main responsibilities of frontend developers. Front end web development is also known as client side of development. HTML, CSS and JavaScript are Front end developing languages whereas PHP, Java, Python and .NET are Back-end web development. Backend development makes server, database and application communicate with each other with the help of server-side languages such as PHP, Ruby, Python and Java.

#### **2.1.5 Razor pay Payment Gateway**

Razor pay is a payment mode founded by the group of [IIT Roorkee](#) alumni. It's a kind of payment mode that combines more than one banking system. It is an online payment system that can access all your credit, debit cards, UPI, and popular mobile wallets. It makes nuisance-free payments in India for others. In this payment system, different banks and wallets get connected to it to make payments easier for everyone.

### **What is Razor pay?**

Razor pay is the in-between mode of payment which transact the payments to the other banks which are connected and available in all kinds of banks, credit, debit, wallets (Netflix, Airtel, Swingy, and others).



**Figure:2.14 Razor pays Logo**

Razor pays subscription also brings with it the useful feature of UPI autopay, nowadays transactions via cards are decreasing day by day due to digital payments. It helps in registered business also where payments get easier for others from different apps. After the pandemic and big break from the direct interaction, people have started to get more comfortable in digital payments where all the business has got a lot of upstream.

This platform of digital payment makes a link between many other apps in which one of the levels is crossed by Razor pay. Apart from the payment gateway, it provides more other benefits to the customers like Razor pay provides where it provides the merchant to automate bank transfer, invoices, and the account for the payroll.

Razor pays capital provides a loan for the business to avoid your cash flow. It benefits all the small business and cashier flow for the people. It helps businesses to make their market place and make available working capital rolls. Now if we come how does it work?

#### **2.1.5.1 How does Razor pay Work?**

Razor pay makes payment convenient for others through different transaction mode if we discuss in normal terms as the Razor pay app first transact the money from the authorized bank with all details after that it provides that amount to the [digital marketing](#) app so when the user starts online payment the marketing app get the money from the Razor pay app this how digitally it makes flow. Now Razor pays provides service of loans also for small

businesses to grow and lend money to avoid the cashier flow, they help in small business-like freelancers, entrepreneurs, enterprises and other businesses.

Mostly this Razor pay is freely used in India and companies like India Mart intermesh limited, Ample technologies, Brio technologies private limited, Savarin car rentals Private limited. While using a Razor pay account a person can receive transactions from the customers online to activate your Razor pay account. Following are the processes that have to be followed to activate your account in Razor pay. It just takes a few minutes from your clock to enjoy your Hassle-free transaction.

### **Standard Plan**

Login into your Razor pay account to access your dashboard, click on the activate account option in the menu panel, for business purpose you have to fill in some details regarding it like your business type, full business name, billing label, your identification proof, website/app, etc. after providing all the details click on the activate account.

Another beneficiary plans are there for customers who plan for their business purpose. The first one is The Standard Plan. This plan mostly works for small business start-ups. It takes 2% transaction fee + GST on every transaction made by the consumer from its credit, debit, wallets like UPI, airtel app, etc. 3% transaction + GST on every transaction made using diners and AMEX cards, other international cards and business credits.

### **Enterprise Plan**

The other plan is The [Enterprise](#) Plan this plan mostly available for the business those who are based on daily transaction bases for this type of plan the Razor pay provide little bit more facilities like it provide 24x7 priority support, plan activation in 24 hours and an account manager for a separate account who can give instant support to the individual account, besides the nominal charges it does not take any charges for setting up free or annual maintenance charges. Making it a widely used payment gateway in India.

There are many different reasons to use Razor pay because it lent capital for your start-ups without an exchange it or any collateral or personal guarantees according to your business only it provides you that kind of facilities like your own account manager for an individual account, time updated and help service according to your account needs, it does not require any type of arrears.

### **Bottom Line**

Choosing Razor pay will not lend you any kind of disadvantage. Every type of generation can have a chance to stand in the market now and have back support with Razor pay to prove themselves. Apart from people, it has made each one of our lives easier in the digital world for payment support. After the big crises in the pandemic our nation has been more updated with such type of companies.

## **Chapter 3**

# **PROJECT DETAIL WITH REQUIREMENTS**

### **3.1 Objective**

E-commerce is quickly becoming a widely accepted and used business model. More and more businesses are developing web sites that allow them to conduct commercial transactions over the internet. It is fair to argue that online purchasing is growing more common. The goal of this project is to create a general-purpose e-commerce store where people can buy things like garments from the comfort of their own homes over the Internet. However, for the purposes of implementation, this project will focus on online clothing buying.

A virtual store on the Internet where clients can browse the catalogue and select products of interest is known as an online store. A shopping cart can be used to collect the specified items. The goods in the shopping cart will be displayed as an order at the moment of checkout. More information will be required at that time to complete the transaction. Typically, the client will be requested to Fill in or choose a billing address, mailing address, shipment option, and payment details like credit card number.

The project's scope will be limited to certain aspects of the e-commerce website. It will display products, and customers will be able to choose catalogues and products, as well as remove items from their cart and specify the number of each item. A cart will be used to collect selected items. The item on the card will be presented as an order at checkout. Customers can finalize an order by paying for the products in their cart. This project has a lot of potential in the future. The project also uses login IDs and passwords to ensure that no unauthorized persons have access to your account. The software can only be accessed by those who are authorized and have the relevant access authority.

### **3.2 Major Functionalities Used in E-commerce Website**

#### **3.2.1 Features of E-Commerce Website in admin side:**

- **Dashboard** – From the admin dashboard, you will have access to all of the system's core functions. Products, whole Sale products, orders, users, trends and categories are just a few examples.

- **Admin Access to Product Management Information System** – The admin has access to the product management information system. He has the ability to add, update, view and delete products and its size table. The admin can also view the user review on specific product. And admin can also perform crude operations on product reviews.
- **Manage Orders** – As one of the admin's primary responsibilities, the admin can check whether or not an order has been paid for by the customer on a case-by-case basis and also update the placed order status like accepted, cancel, on the way, packed or delivered etc.
- **Handle Categories** – When it comes to categories, the admin has the ability to manage the categories and also manage subcategories inside it.
- **Manage Users** – The account of a user and seller can be managed by the administrator. The system administrator has the ability to add, update, and delete users. Also manage the Users cart.
- **Handle slider in user interface** – The admin can add, delete, update picture in user and seller slider that are show on their respective interface.
- **Secure Login and Logout** – One of the system's security features is the secure login and logout system, which is enabled by default.

### **3.2.2 E-Commerce Website User front end side:**

- **Main Page** – You may see a direct list of products for sale on the home page, and you can filter the results by category and subcategory. At the top of front page there is a slider. And search bar for product searching operation. Below of slider there is another row in which trending products are shown. (Trending products are added on the basis of number of product purchase by default)
- **Viewing Products** – The customer can view all of the products on the frontend. By default, some limited products are shown in respective category. We can view all the product by clicking *view all* button in specific category product. The customer can have detailed view by clicking on the product where they can view available product sizes, review about this product if product have and also add their own review after login.

- **Checkout Order** – The customer can place an order in the frontend if login, which the admin will validate in the backend. The customer can place an order of individual item as well as all cart items at once. For placing an order of multiple items customer must add different items to cart, after then customer can place an order.
- **Add to cart** – The customer can add various items in it cart. The customer can vary the quantity of particular item in cart and also can remove the particular item from cart. Minimum value of selected item is one and maximum value of that item is equal to quantity of that item present in stock. The customer can view cart element by clicking cart icon. The value of cart items is sustained after user log out.
- **Payment mode** – The customer can make payment by cash on delivery or make online payment through razor pay gateway which provide secure gateway for money transaction and also provide different way of online transaction like UPI, Paytm wallet, Google pay, net banking etc.
- **Track order** – The customer can track it placed order status. And also cancel the order before order's packing. Once the order is packed Customer cannot cancel the order.
- **Customers' Profiles** – Customers can register their profiles on the frontend. In a nutshell, this system includes a customer management system.

### **3.2.3 E-Commerce Website Seller front end side:**

- **Main Page** – You may see a direct list of products for sale on the home page, and you can filter the results by category and subcategory. At the top of front page there is a slider. At right side of slider there is another row in which trending products are shown. (Trending products are added on the basis of number of product purchase by default)
- **Viewing Products** – The seller can view all of the wholesale products on the frontend. By default, some limited products are shown in respective category. We can view all the product by clicking *view all* button in specific category product. The customer can have detailed view by clicking on the product.

- **Checkout Order** – The customer can place an order in the frontend if login, which the admin will validate in the backend.
- **Add to cart** – The customer can add various items in it cart. The customer can vary the quantity of particular item in cart and also can remove the particular item from cart. Minimum value of selected item in one bundle already limited by admin. The customer can view cart element by clicking cart icon. The value of cart items is sustained after user log out.
- **Track order** – The customer can track its placed order status. And also cancel the order before order's packing. Once the order is packed Customer cannot cancel the order.
- **Seller's Profiles** – Seller can register their profiles on the frontend. In a nutshell, this system includes a Seller management system. Seller can update his/her profile, bank details or password etc.
- **Add Product**- Seller have an add product section where they can add their product detail for sale.
- **View Product**- Where Seller can view or delete their own added product from the main Product database.
- **Dashboard**- In dashboard section seller can track/change the status of their placed order product. By clicking on view product button, they will get all the ordered product in front end. Once they get the product detail, they can change customer ordered products status. There is a view all delivered order button present, by clicking on it seller can see their all customer delivered product. Seller can view and handle only those customer product orders which are related with them.
- **Contact**- Seller can contact with admin by sending his/her queries along with his name, email id etc. Admin can respond their queries through provided email id.

### **3.3 System specifications:**

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

### **3.3.1 Hardware requirement:**

Sr No.	Hardware	Minimum System Requirement
1	Processor	2.4 GHz Processor speed
2	Memory	2 GB
3	Disk Space	500 GB

**Table : 3.1 Required Hardware**

### **3.3.2 Software requirement:**

Sr No.	Hardware	Minimum System Requirement
1	Operating system	Window 8, 10 or MAC Ox 10.8, 10.9, or LINUX
2	DBMS	SQL Lite 3
3	Runtime Environment	PyCharm or Visual Studio Code

**Table : 3.2 Required Software**

## **3.4 The Steps to Run a Django E-Commerce Website Project with Source Code**

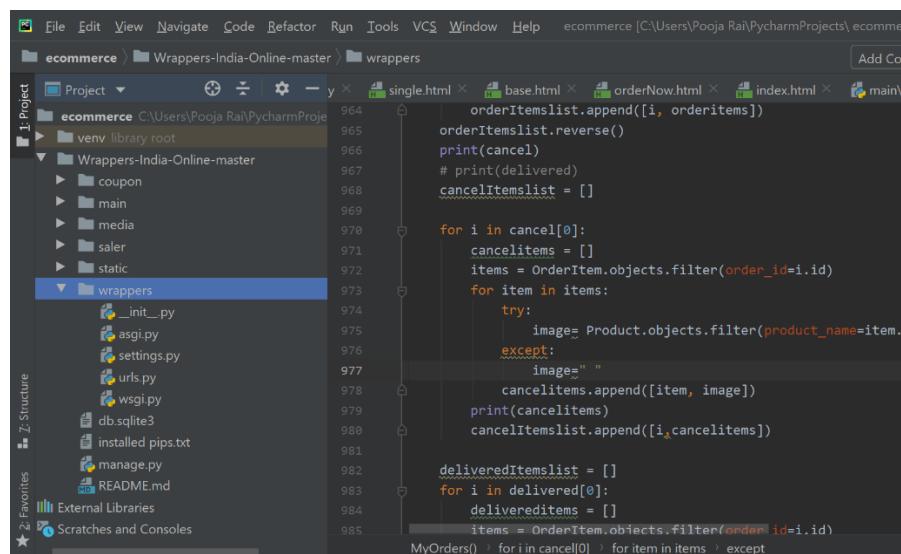
Here's the steps to run an E-Commerce Website Project using Django

Django is a **Model View Control (MVC)** framework.

- **Mange.py** -This file is used basically as a command-line utility and for deploying, debugging, or running our web application. It contains code for run-server, or make migrations or migrations, etc. that we use in the shell. Anyway, we do not need to make any changes to the file.
- **Db.sqlite3** is the database file.
- **Static** folder contains all the static files like CSS and images.
- **Env** folder is the project specific development environment. It's created through a command '*virtualenv Env*'

**Wrapper folder** is the Django projects consisting of init, admin, models, views, apps and forms.

- **Models** are basically the blueprints of the database we are using and hence contain the information regarding attributes and the fields etc. of the database.
- **Views** is a crucial one, it contains all the Views (usually as classes). Views.py can be considered as a file that interacts with the client. Views are a user interface for what we see when we render a Django Web application.
- **URLs** Just like the project urls.py file, this file handles all the URLs of our web application.



The screenshot shows the PyCharm IDE interface. The left sidebar displays the project structure under 'Project'. The 'wrappers' directory is selected, showing files like \_\_init\_\_.py, asgi.py, settings.py, urls.py, wsgi.py, db.sqlite3, installed.pip.txt, manage.py, and README.md. The right side shows a code editor with Python code. The code appears to be part of a script for handling orders, specifically dealing with order items, cancel items, and delivered items. It includes imports for OrderItem and Product, and logic for filtering and appending items to lists based on their status (order, cancel, delivered).

```

File Edit View Navigate Code Refactor Run Tools VCS Window Help ecommerce [C:\Users\Pooja Rai\PycharmProjects\ecommerce]
ecommerce > Wrappers-India-Online-master > wrappers
Project ecommerce C:\Users\Pooja Rai\PycharmProjects\ecommerce
  venv library root
    Wrappers-India-Online-master
      coupon
      main
      media
      saler
      static
      wrappers
        __init__.py
        asgi.py
        settings.py
        urls.py
        wsgi.py
        db.sqlite3
        installed.pip.txt
        manage.py
        README.md
  Favorites External Libraries Scratches and Consoles
Add Co
964     orderItemslist.append([i, orderitems])
965     orderItemslist.reverse()
966     print(cancel)
967     # print(delivered)
968     cancelItemslist = []
969
970     for i in cancel[0]:
971         cancelItems = []
972         items = OrderItem.objects.filter(order_id=i.id)
973         for item in items:
974             try:
975                 image= Product.objects.filter(product_name=item.
976             except:
977                 image= " "
978             cancelItems.append([item, image])
979             print(cancelItems)
980             cancelItemslist.append([i, cancelItems])
981
982     deliveredItemslist = []
983     for i in delivered[0]:
984         deliveredItems = []
985         items = OrderItem.objects.filter(order_id=i.id)

```

**Figure:3.1 Wrapper Module**

**Ecommerce** folder is the Django application directory consisting of the following essential files.

- **Init.py** This file remains empty and is present them only to tell that this particular directory is a package.
- **Settings.py** This file is present for adding all the applications and the middleware application present. Also, it has information about templates and databases. Overall, this is the main file of our Django web application.
- **Urls.py** This file handles all the URLs of our web application. This file has the lists of all the endpoints that we will have for our website.

- **Wsgi.py** This file mainly concerns with the WSGI server and is used for deploying our applications on to servers like Apache etc.
  - **Asgi.py** In the newer versions of Django, you will also find a file named as asgi.py apart from wsgi.py. ASGI can be considered as a succeeded interface to the WSGI. ASGI, short for Asynchronous Server Gateway interface

The screenshot shows the PyCharm IDE interface with the following details:

- File Path:** ecommerce (C:\Users\PoojaR\PycharmProjects\ecommerce) - \views.py
- Code Editor:** The current file is `views.py`. The code handles order processing logic, including reversing order items, canceling items, and filtering delivered items.
- Project Explorer:** Shows the project structure with packages `main`, `migrations`, `PayTm`, `static`, and `templates`. Inside `templates/main`, files like `account_base.html`, `index.html`, and `orderNow.html` are listed.
- Code Navigator:** A sidebar on the left lists `__init__.py`, `models.py`, `views.py`, and `wsgi.py`.
- Status Bar:** Shows the current file as `views.py`, line 985, and the status `972-2 LF, UTF-8, 4 spaces, Python 3.7 (ecommerce)`.

**Figure:3.2 Main Module**

**Main folder** is the Django projects consisting of init, admin, models, views, apps and forms. All this file is related to user. Template folder inside it carry all html pages of user front end side.

The screenshot shows the PyCharm IDE interface with the following details:

- File Menu:** File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, Help.
- Project Bar:** ecommerce (Wrappers-India-Online-master) - main - templates - main - account\_base.html
- Code Editor:** The code for `account_base.html` is displayed. It includes logic for handling cancellation of items, displaying product images, and filtering delivered items. The code uses Python and Jinja2 syntax.
- File Browser:** Shows the project structure under `Project`:
  - `saler` (containing `migrations`, `static`, and `templates` folders)
  - `templates/saler` (containing files: `account_settings.html`, `add_product.html`, `admin2.html`, `base.html`, `cart.html`, `checkout.html`, `dashboard.html`, `dashboard1.html`, `index.html`, `myorders.html`, `payment.html`, `productview.html`, `productview1.html`, `seller.signup.html`, `view_all.html`, and `view_products.html`)
- Toolbars:** Includes icons for Undo, Redo, Cut, Copy, Paste, Find, Replace, and others.
- Bottom Navigation:** Includes tabs for TODO, Terminal, Python Console, and Event Log.

### **Figure:3.3 Saler Module**

**Seller folder** is the Django projects consisting of init, admin, models, views, apps and forms. All this file is related to seller. Template folder inside it carry all html pages of seller front end side.

### 3.5 Entity Relationship Diagram:

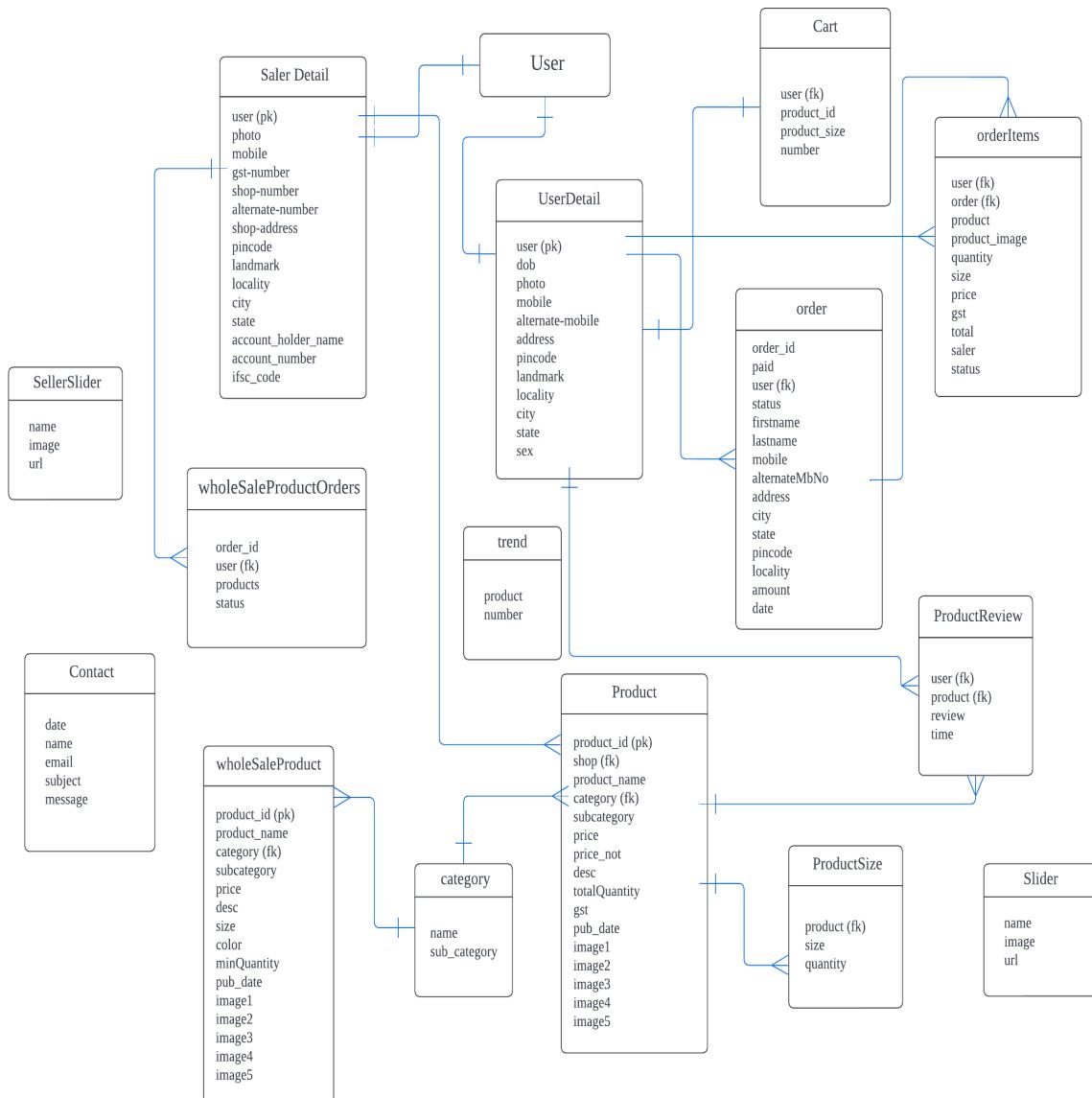
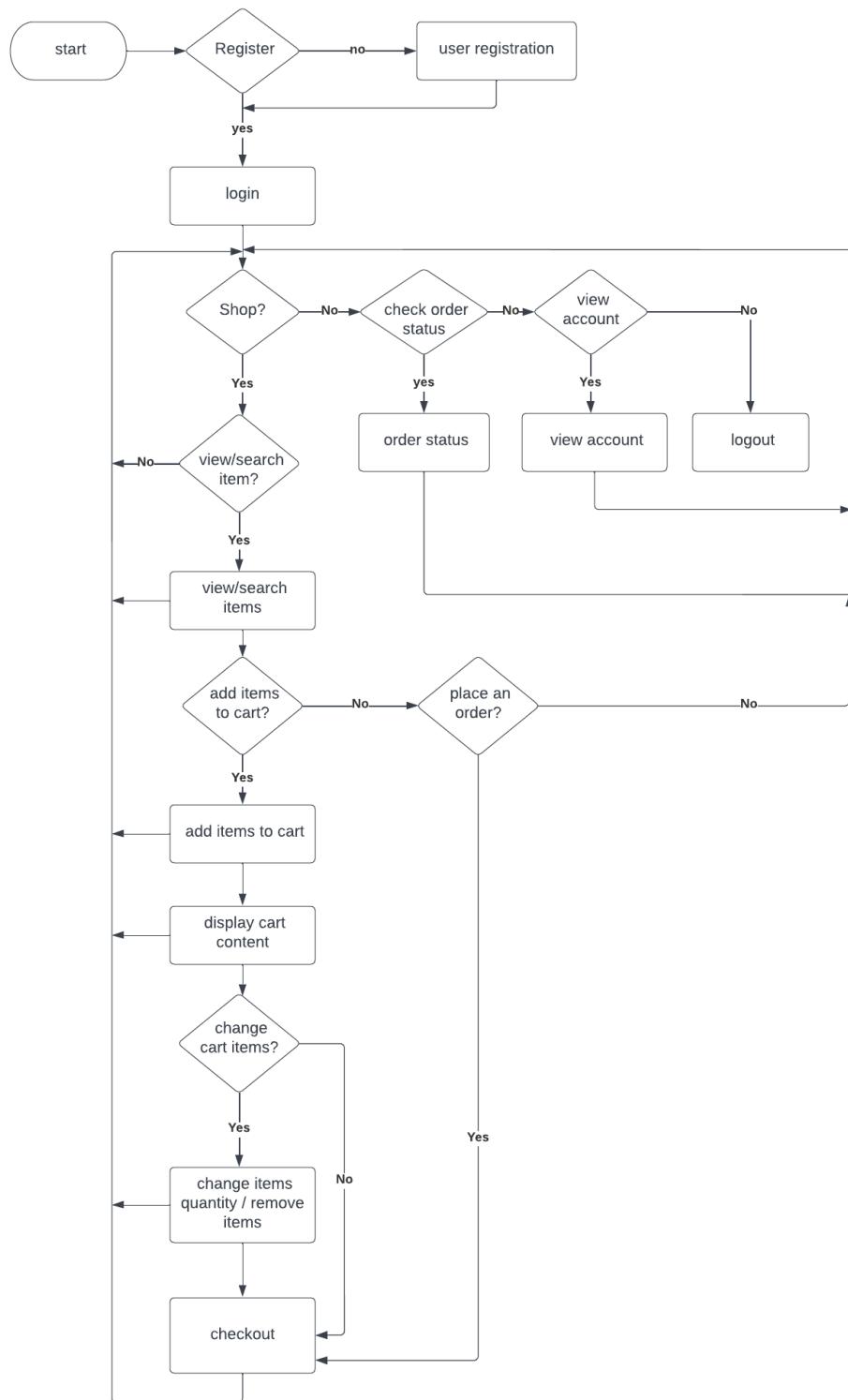


Figure:3.4 Entity Relationship Diagram

### 3.6 Flow Chart:



**Figure:3.5 Flow Chart**

## Chapter 4

# RESULTS

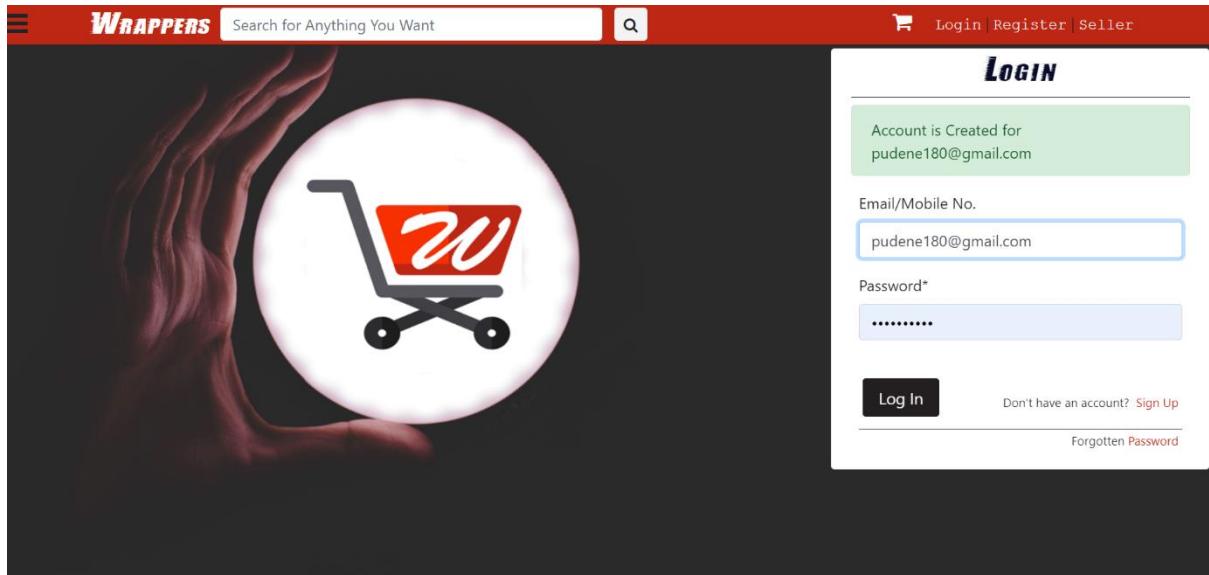


Figure:4.1 Login Page

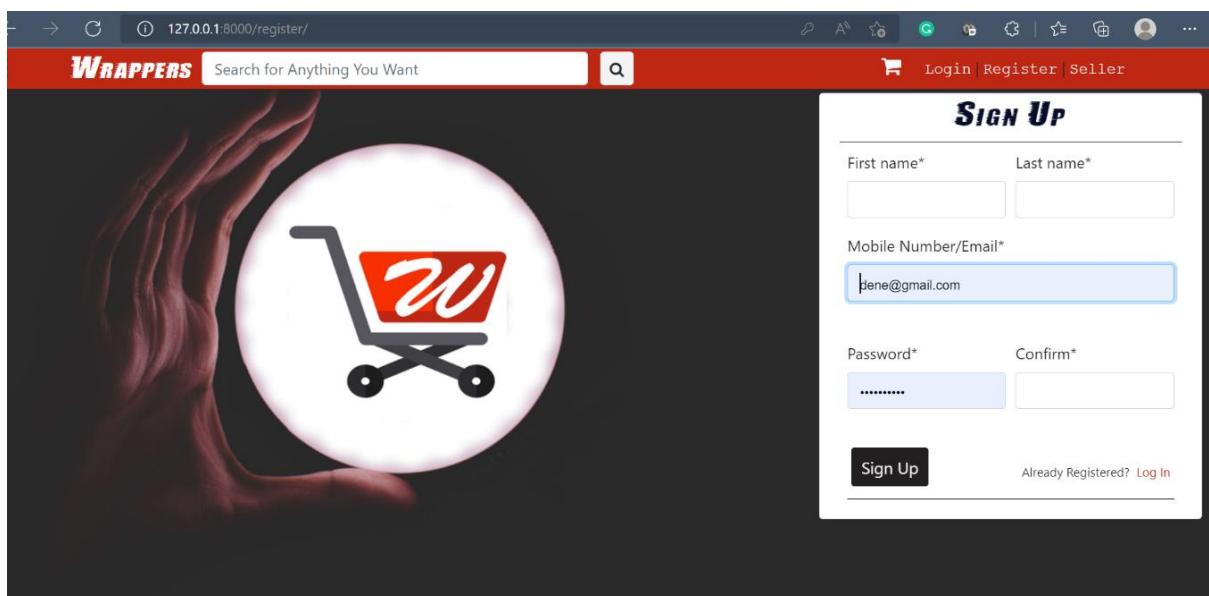


Figure:4.2 Sign Up Page

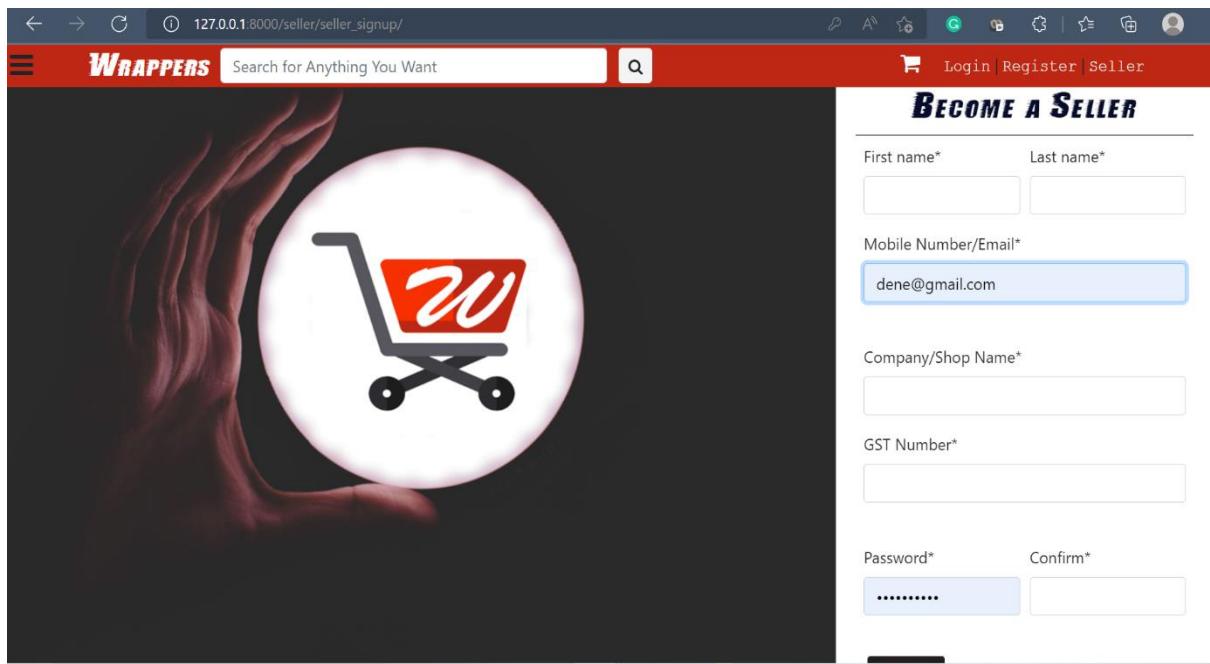


Figure:4.3 Seller Registration Page

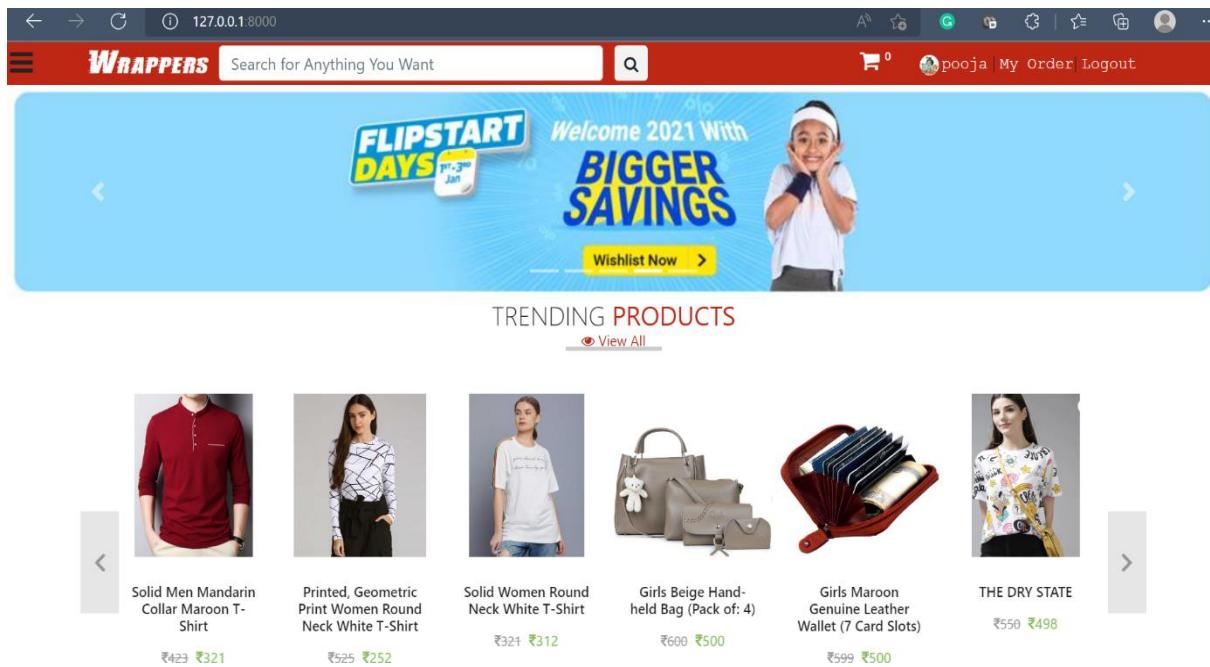
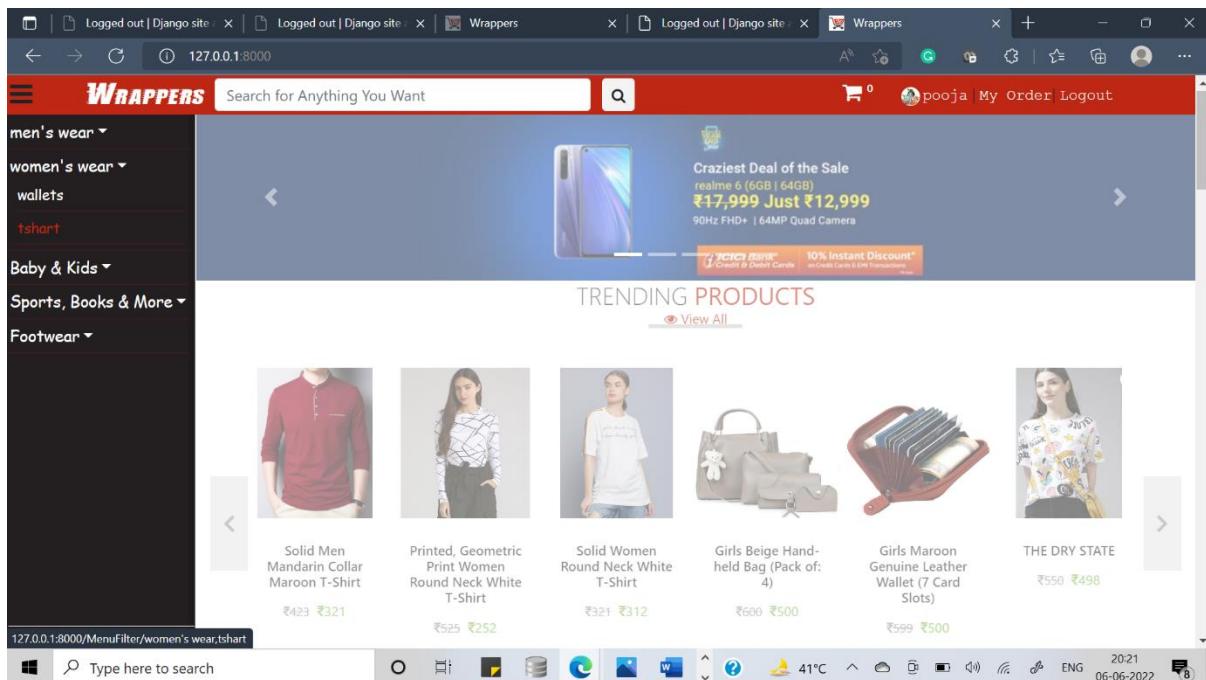
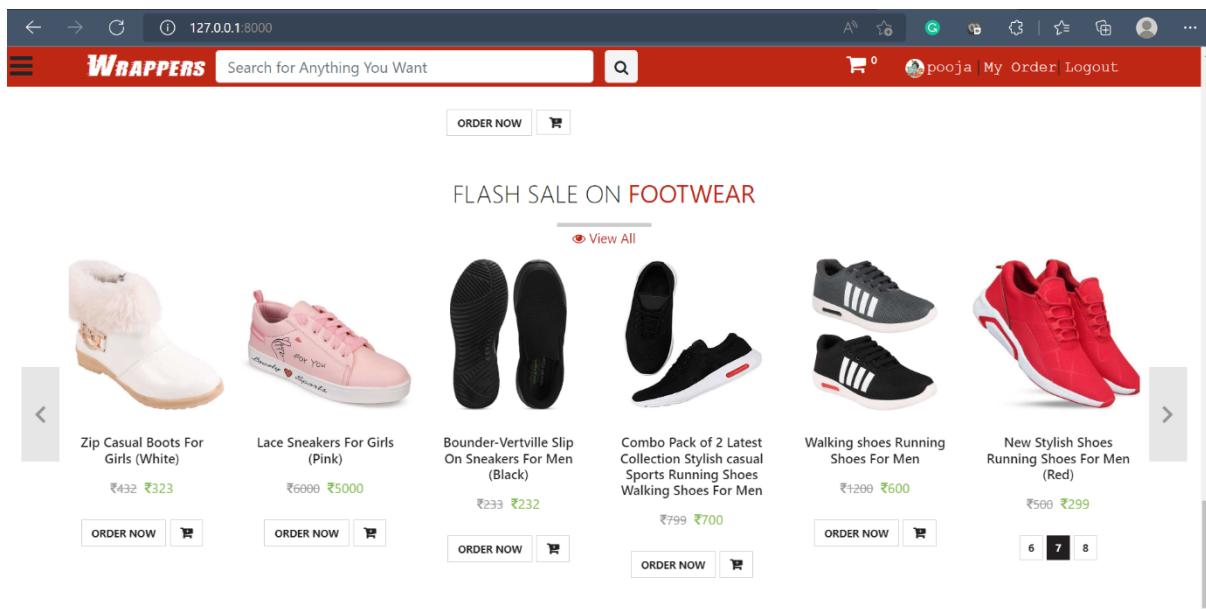


Figure:4.4 User Home Page



**Figure:4.5 Category Filtration Side Bar**



**Figure:4.6 Product List**

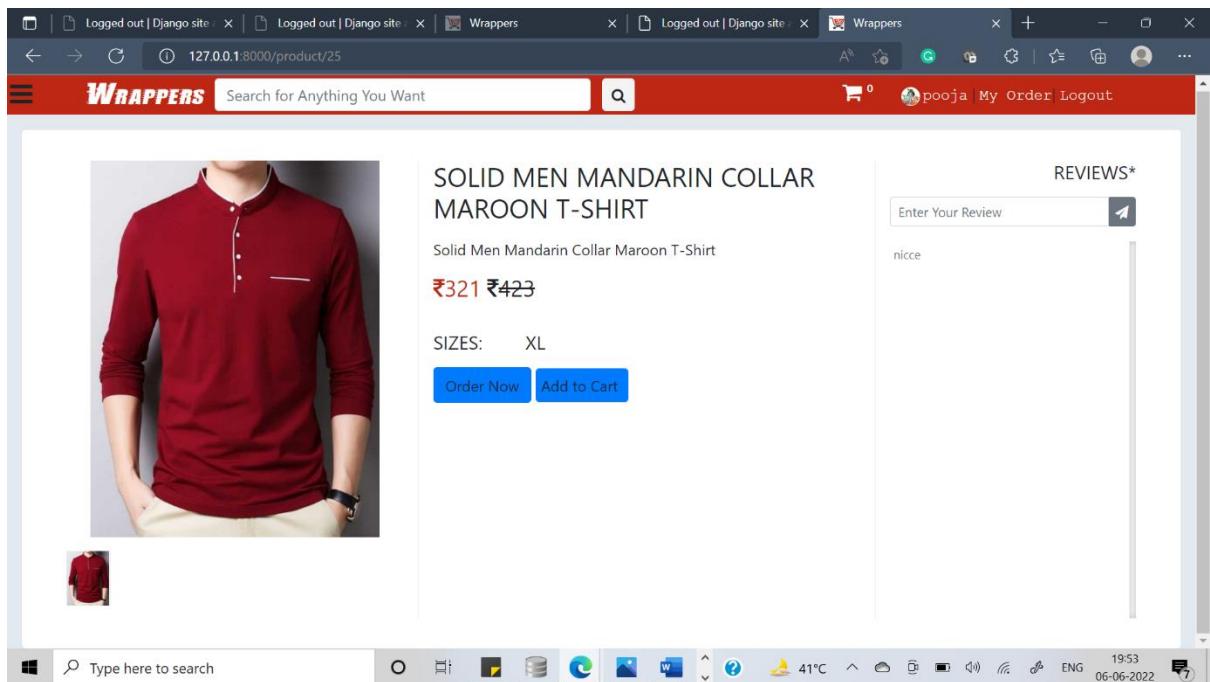


Figure:4.7 Particular Product Details Page

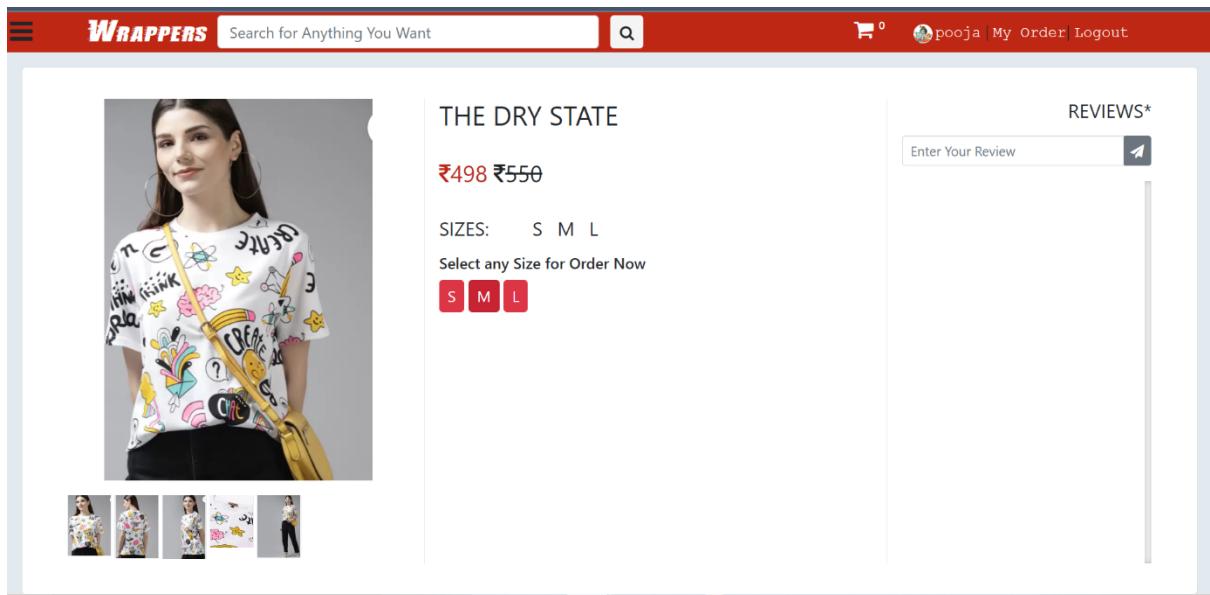


Figure:4.8 Size selection Buttons

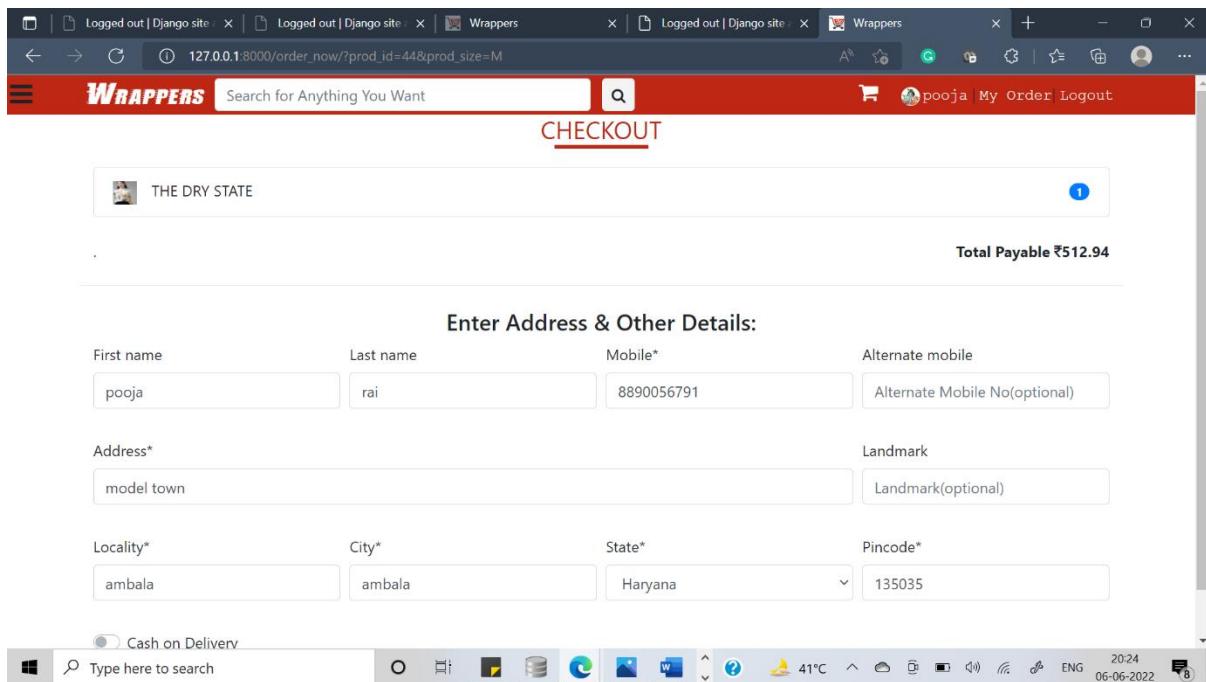


Figure:4.9 Checkout Page

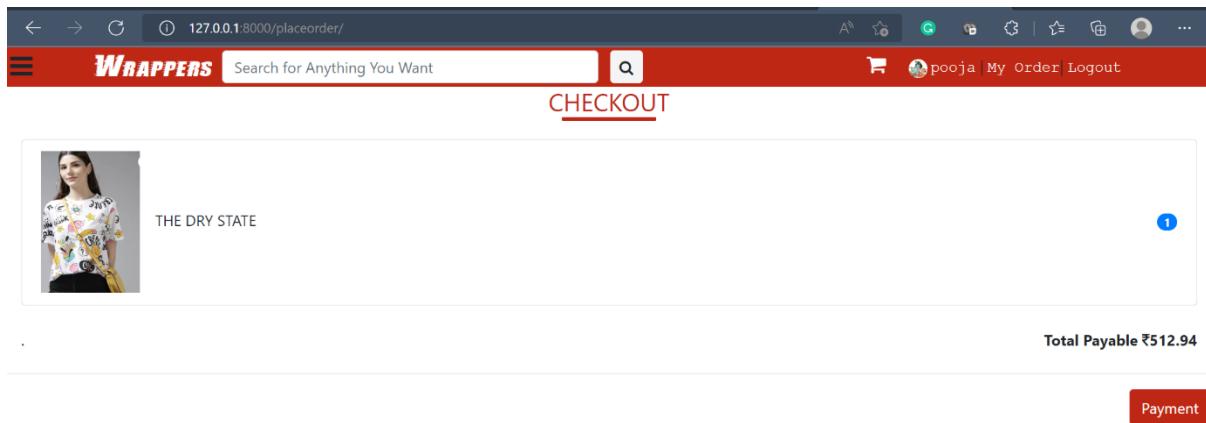
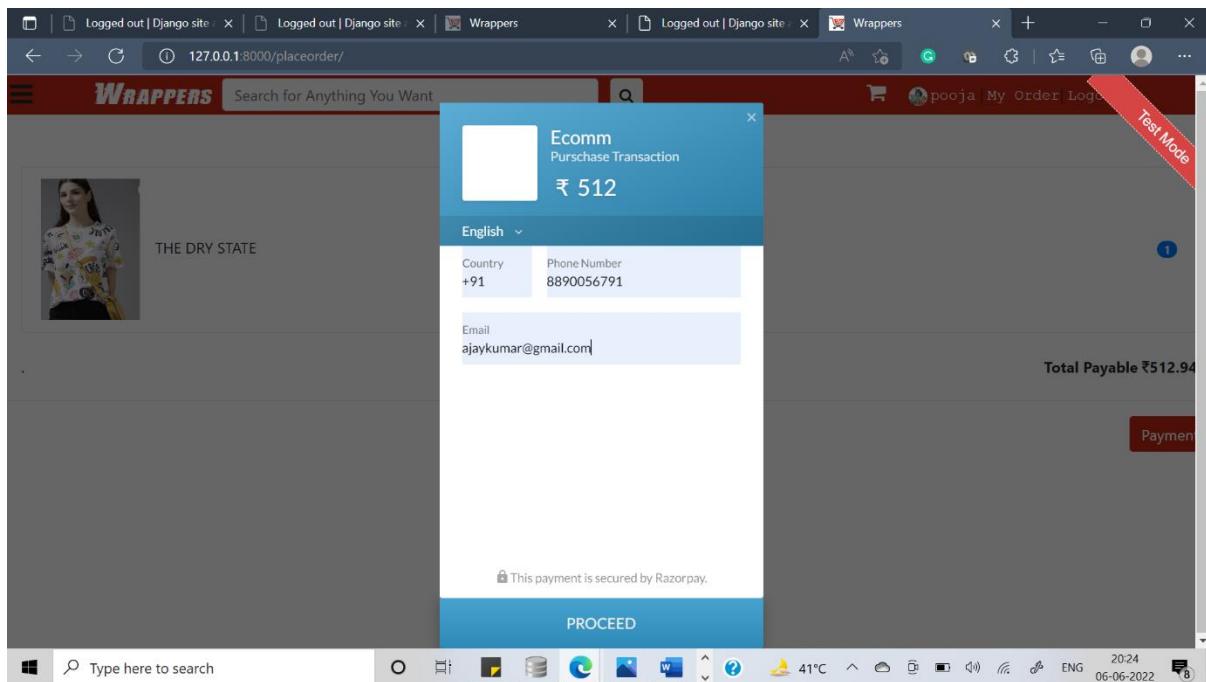
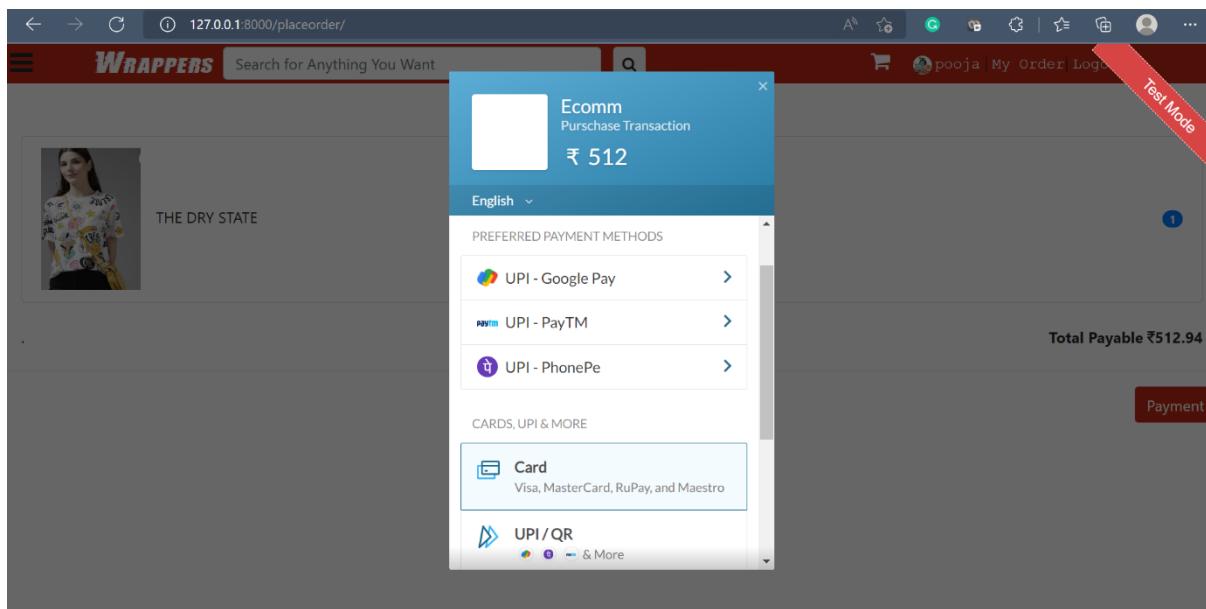


Figure:4.10 Payment Page



**Figure:4.11 Razor pays Gateway Login page**



**Figure:4.12 Online Payment Mode Selection**

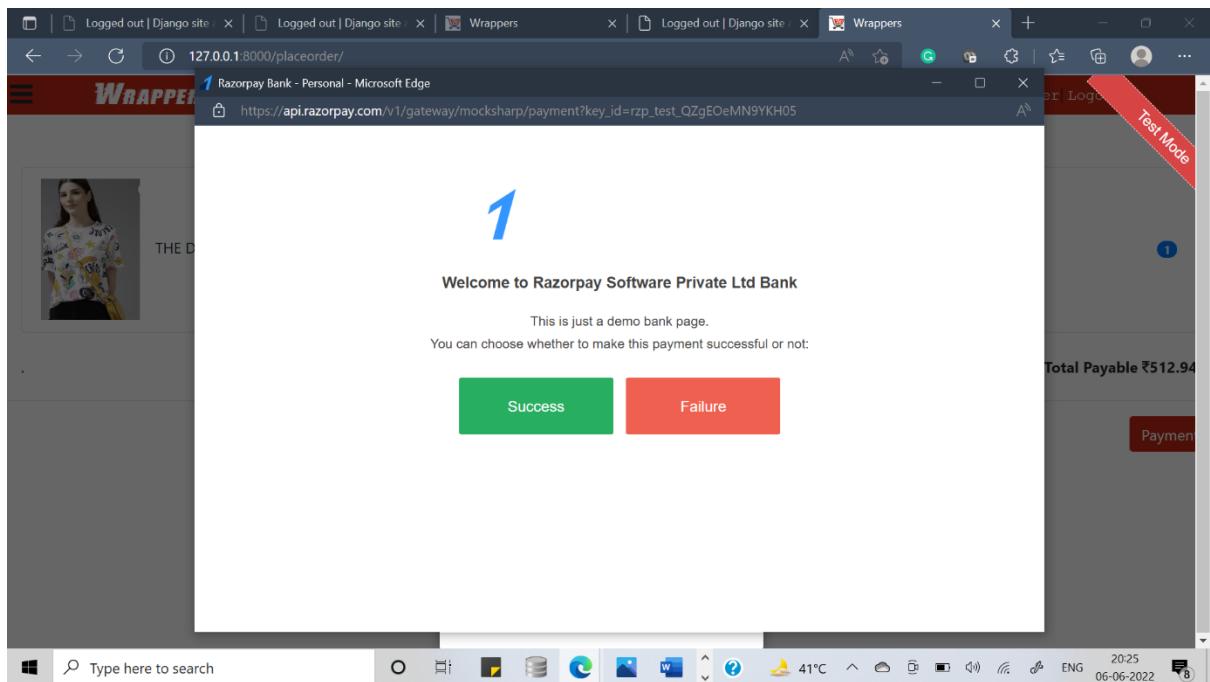


Figure:4.13 Payment Success/Failure Page

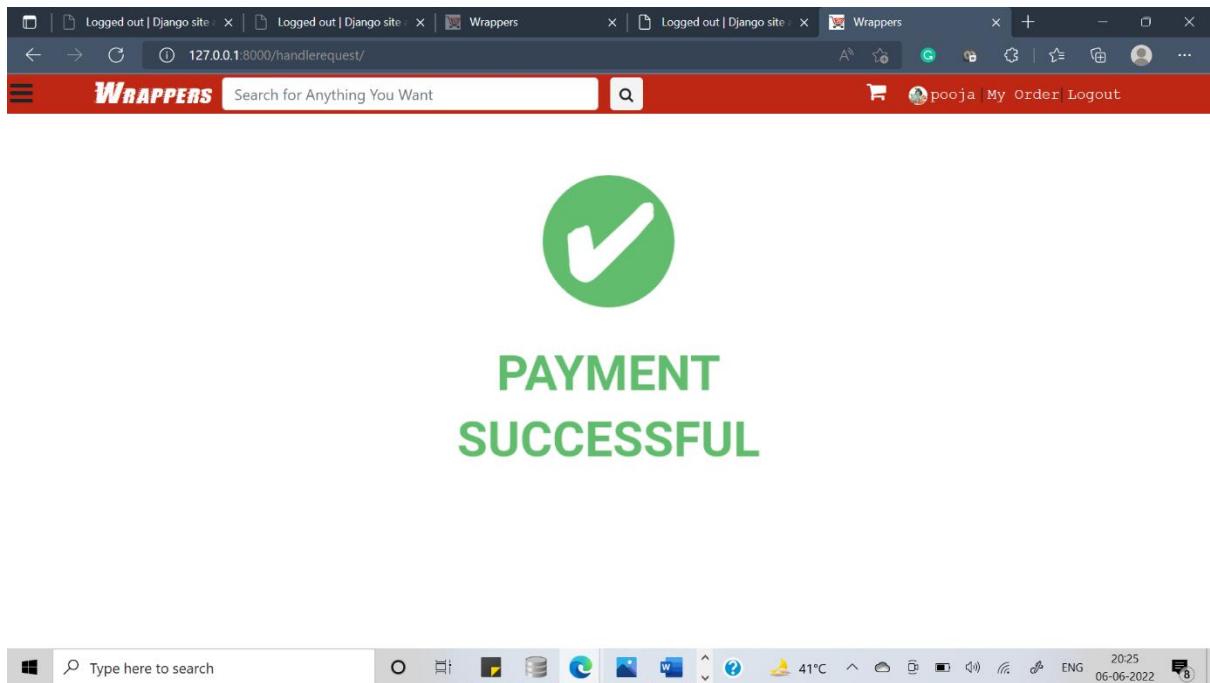
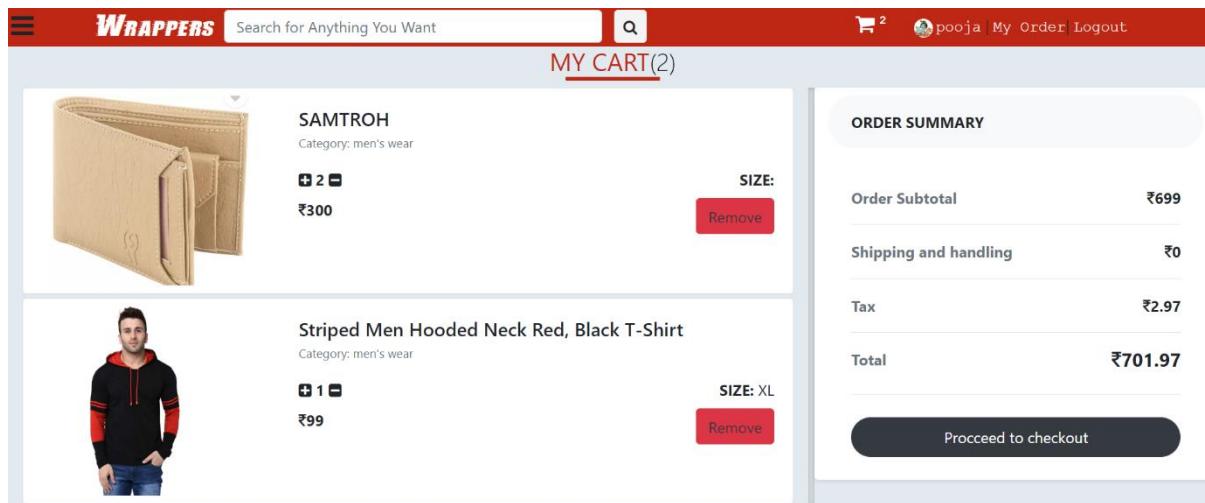


Figure:4.14 Payment Confirmation Page



**Figure:4.15 User Cart View**

**WRAPPERS** Search for Anything You Want

Striped Men Hooded Neck Red, Black T-Shirt

Total Payable ₹701.97

**Enter Address & Other Details:**

First name	Last name	Mobile*	Alternate mobile
pooja	rai	8890056791	Alternate Mobile No(optional)
Address*		Landmark	
model town		Landmark(optional)	
Locality*	City*	State*	Pincode*
ambala	ambala	Haryana	135035

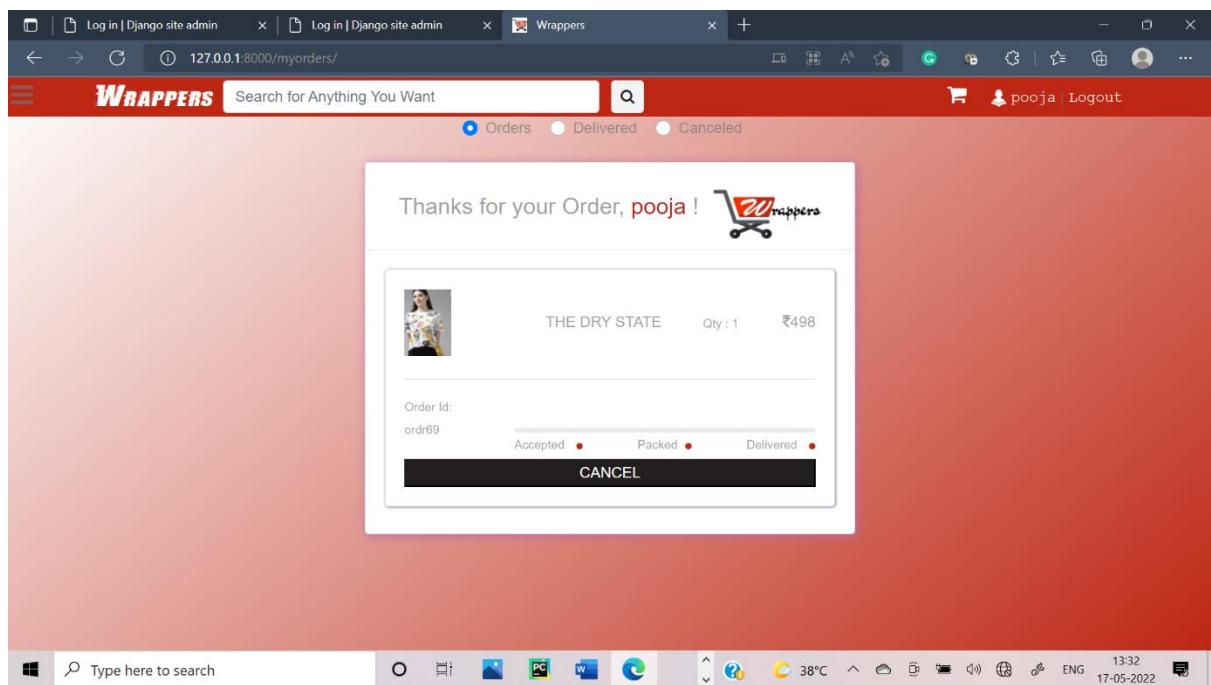
Cash on Delivery

**Place Order**

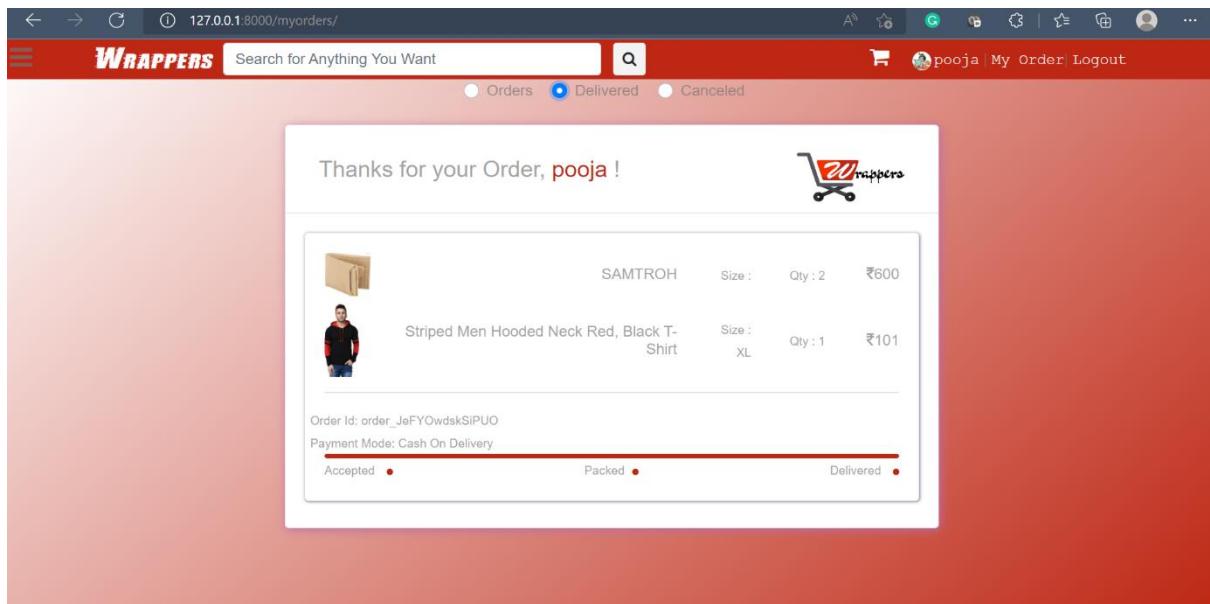
**Figure:4.16 Place Order Page**



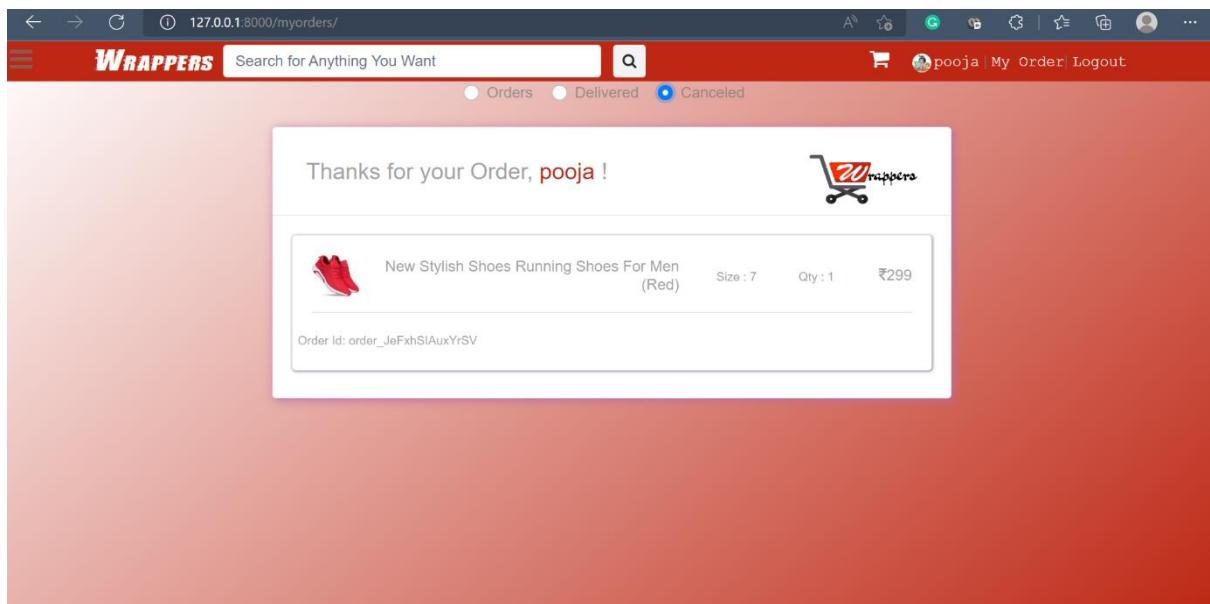
**Figure:4.17 Order Placed Conformation**



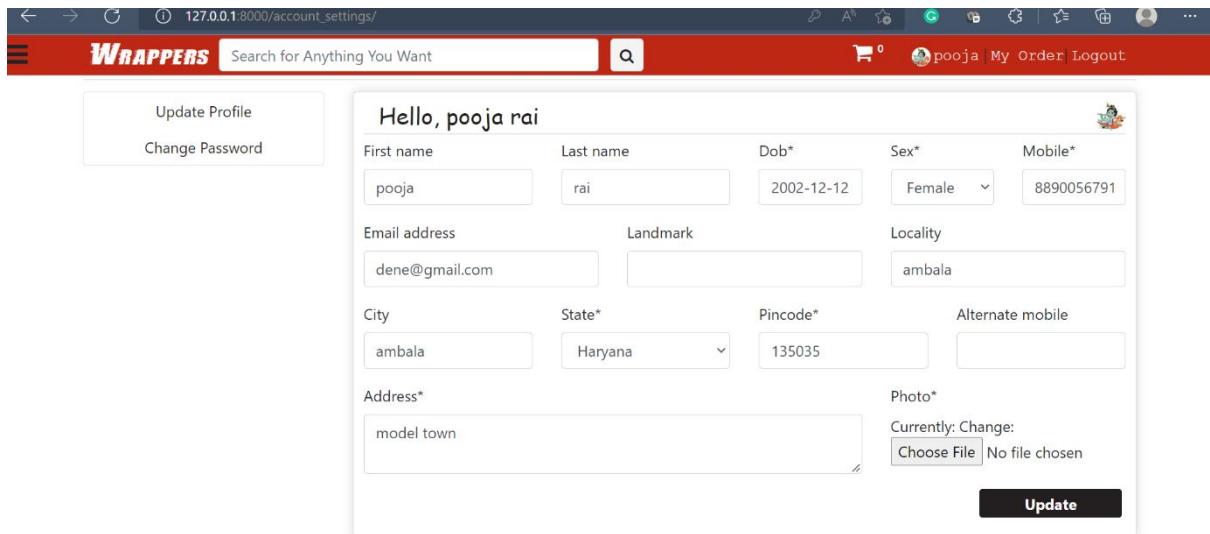
**Figure:4.18 Placed Orders Status**



**Figure:4.19 Delivered Product List**



**Figure:4.20 Cancelled Product List**

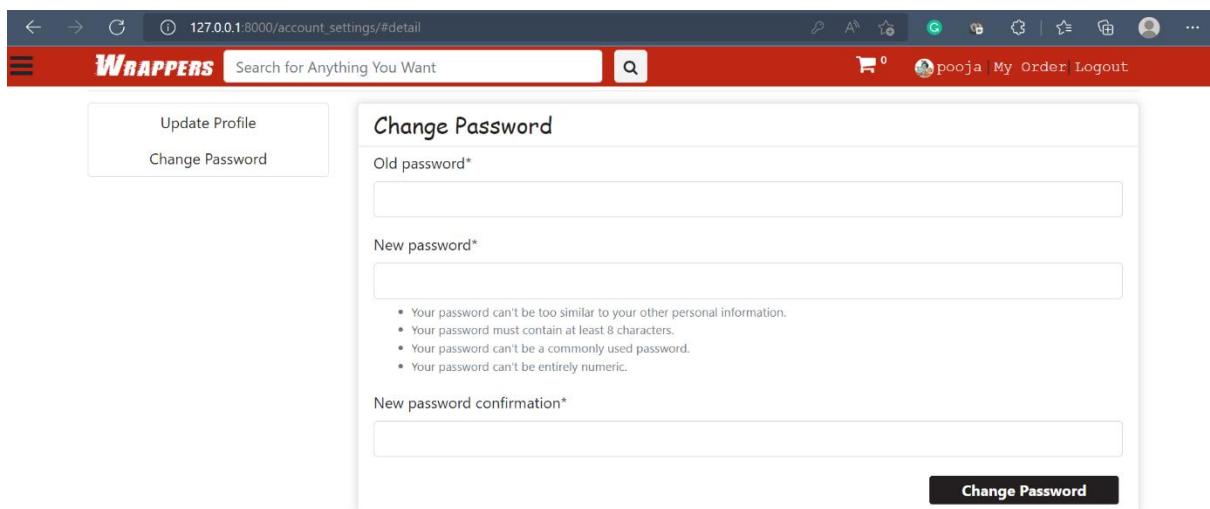


The screenshot shows a user profile update form on a website. The header includes the brand name "WRAPPERS" and a search bar. The main content area has a title "Hello, pooja rai". The form fields include:

- First name: pooja
- Last name: rai
- Dob\*: 2002-12-12
- Sex\*: Female
- Mobile\*: 8890056791
- Email address: dene@gmail.com
- Landmark:
- Locality: ambala
- City: ambala
- State\*: Haryana
- Pincode\*: 135035
- Address\*: model town
- Photo\*: Currently: Change: Choose File | No file chosen

A black "Update" button is at the bottom right.

**Figure:4.21 User Profile Updating Page**



The screenshot shows a password change form on a website. The header includes the brand name "WRAPPERS" and a search bar. The main content area has a title "Change Password". The form fields include:

- Old password\*
- New password\*
- New password confirmation\*

Below the "New password" field is a list of validation rules:

- Your password can't be too similar to your other personal information.
- Your password must contain at least 8 characters.
- Your password can't be a commonly used password.
- Your password can't be entirely numeric.

A black "Change Password" button is at the bottom right.

**Figure:4.22 User Password Updating Page**

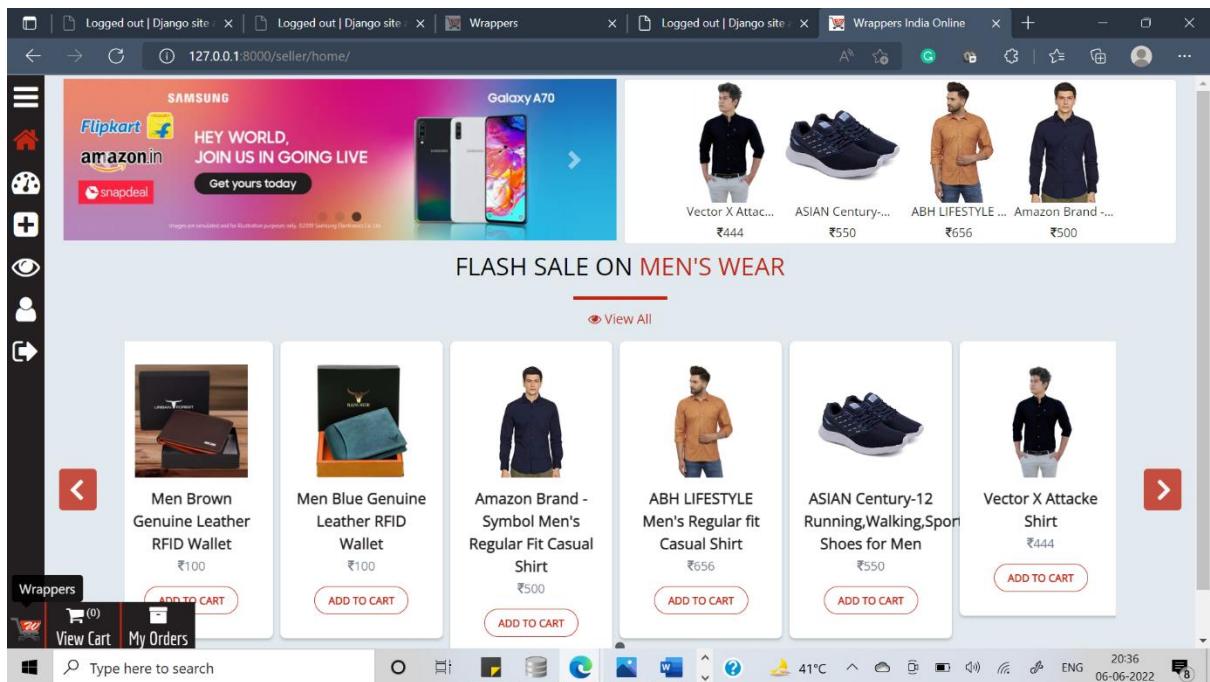


Figure:4.23 Seller Home Page

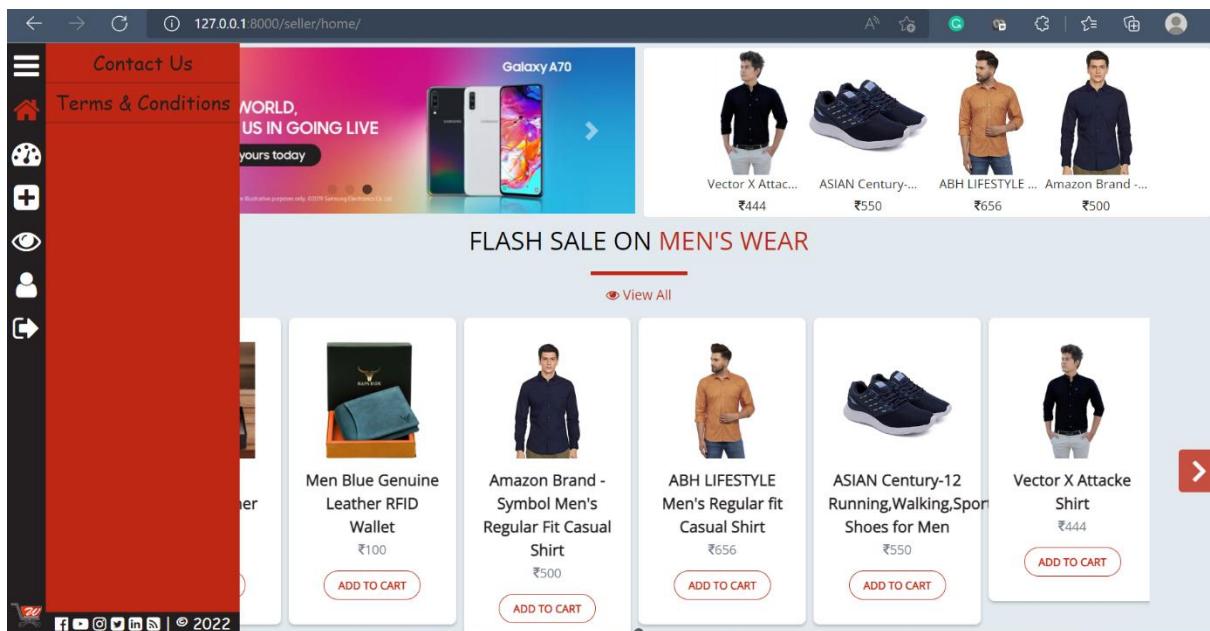
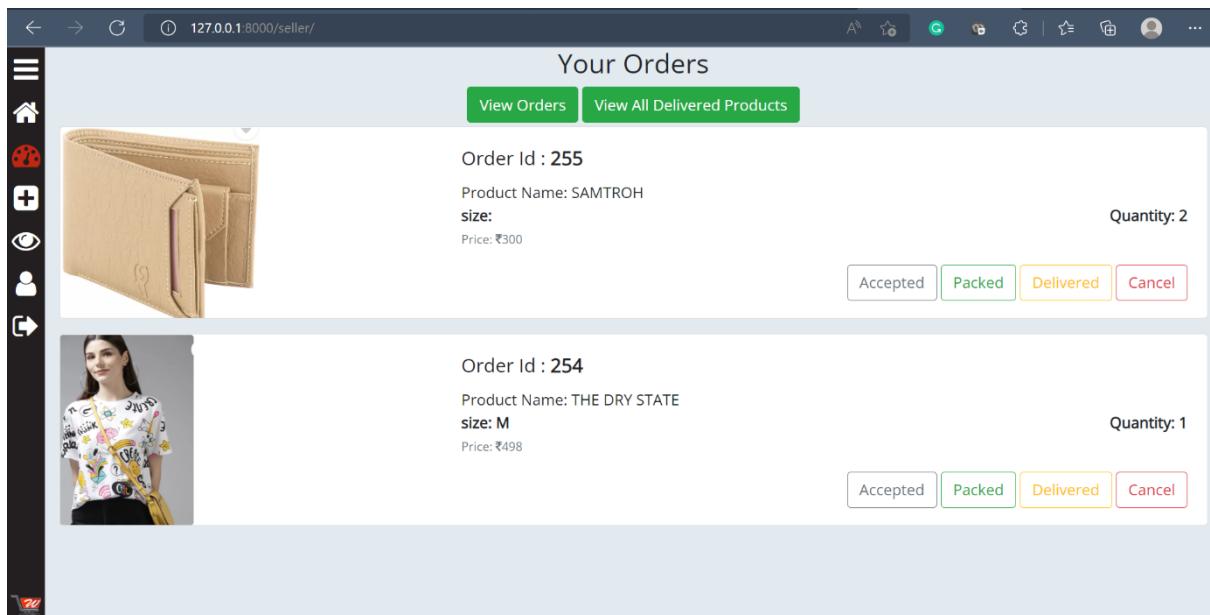
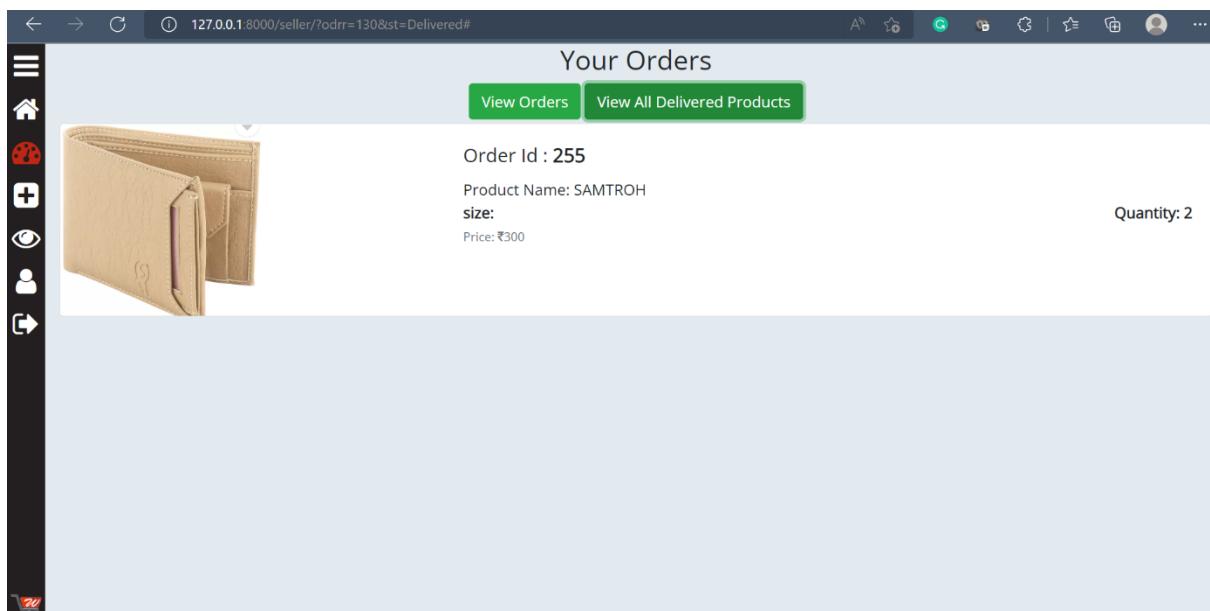


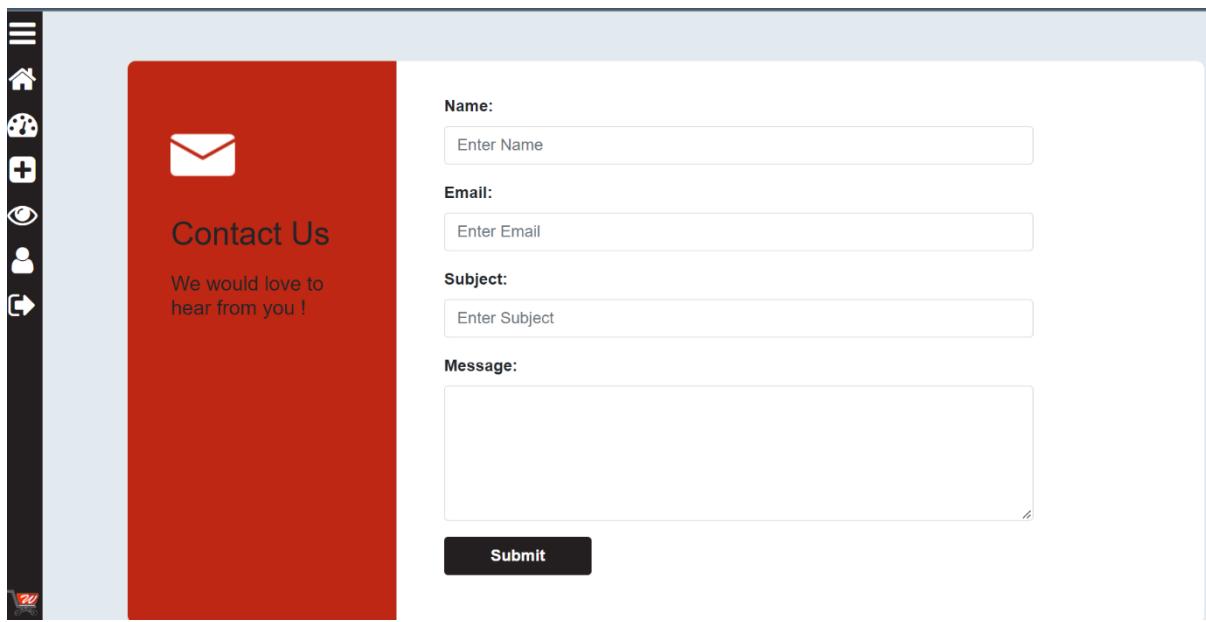
Figure:4.24 Seller Sidebar



**Figure:4.25 Seller View Orders**

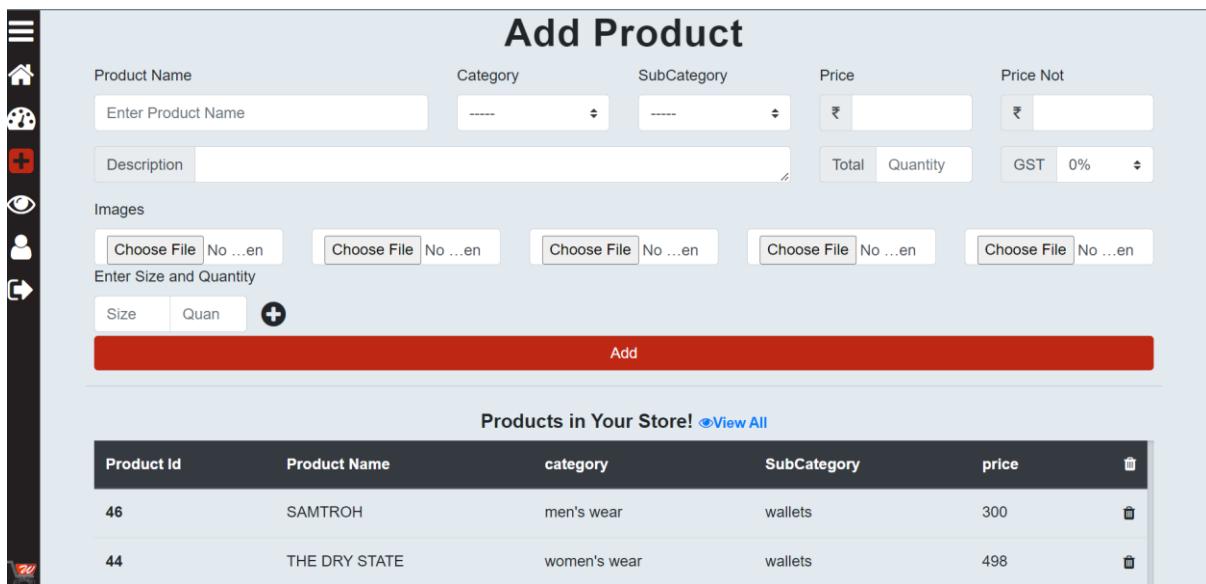


**Figure:4.26 Seller View Delivered Products**



The screenshot shows a contact form titled "Contact Us" on a red background. The form includes fields for Name, Email, Subject, and Message, each with an "Enter" placeholder. A "Submit" button is at the bottom. The left sidebar has icons for Home, Paint, Plus, Eye, Person, and Print.

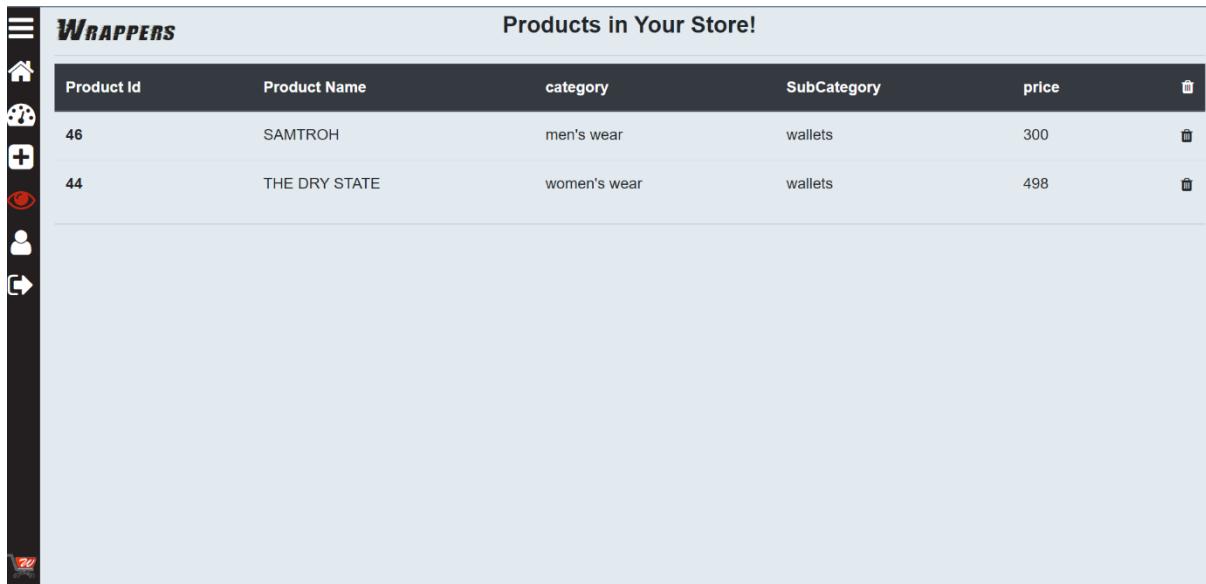
**Figure:4.27 Seller Contact Page**



The screenshot shows the "Add Product" page. It features fields for Product Name, Category, SubCategory, Price, and Price Not. Below these are fields for Description, Total, Quantity, GST, and 0%. There's a section for Images with five "Choose File" buttons. Under "Enter Size and Quantity", there are "Size" and "Quan" input fields and a "+" button. A large red "Add" button is at the bottom. Below this is a table titled "Products in Your Store! <a href="#">View All". The table has columns: Product Id, Product Name, category, SubCategory, price, and a delete icon. Two rows are shown: one for product ID 46 (SAMTROH, men's wear, wallets, 300) and one for product ID 44 (THE DRY STATE, women's wear, wallets, 498).

Product Id	Product Name	category	SubCategory	price	
46	SAMTROH	men's wear	wallets	300	
44	THE DRY STATE	women's wear	wallets	498	

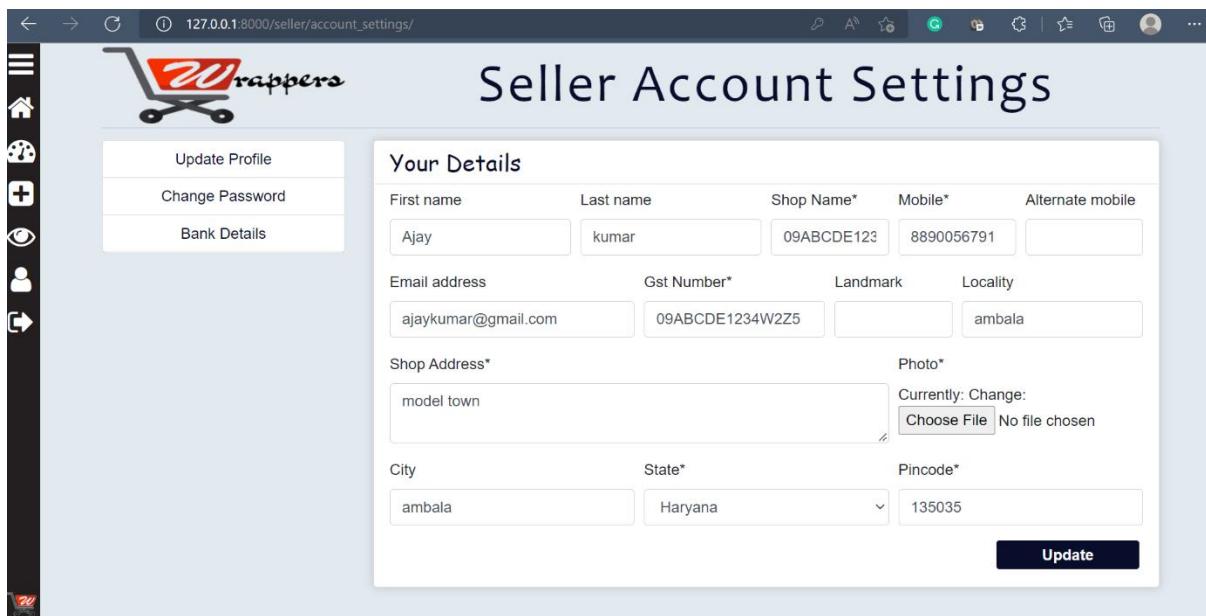
**Figure:4.28 Seller Add New Products Page**



The screenshot shows a table titled "Products in Your Store!" with columns: Product Id, Product Name, category, SubCategory, price, and a delete icon. There are two items listed:

Product Id	Product Name	category	SubCategory	price	
46	SAMTROH	men's wear	wallets	300	
44	THE DRY STATE	women's wear	wallets	498	

**Figure:4.29 Seller Product List**



The screenshot shows the "Seller Account Settings" page. It features a sidebar with icons for Home, Add Product, Add Category, Logout, and a profile picture. The main area has a title "Seller Account Settings" and a logo for "wrappers". A sidebar on the left lists "Update Profile", "Change Password", and "Bank Details". The "Your Details" section contains fields for First name, Last name, Shop Name\*, Mobile\*, Alternate mobile, Email address, Gst Number\*, Landmark, Locality, Shop Address\*, Photo\*, City, State\*, Pincode\*, and a file upload button. A "Choose File" button is shown with the message "No file chosen". A "Currently: Change:" label is also present. At the bottom right is a blue "Update" button.

**Figure:4.30 Seller Profile Updating Page**

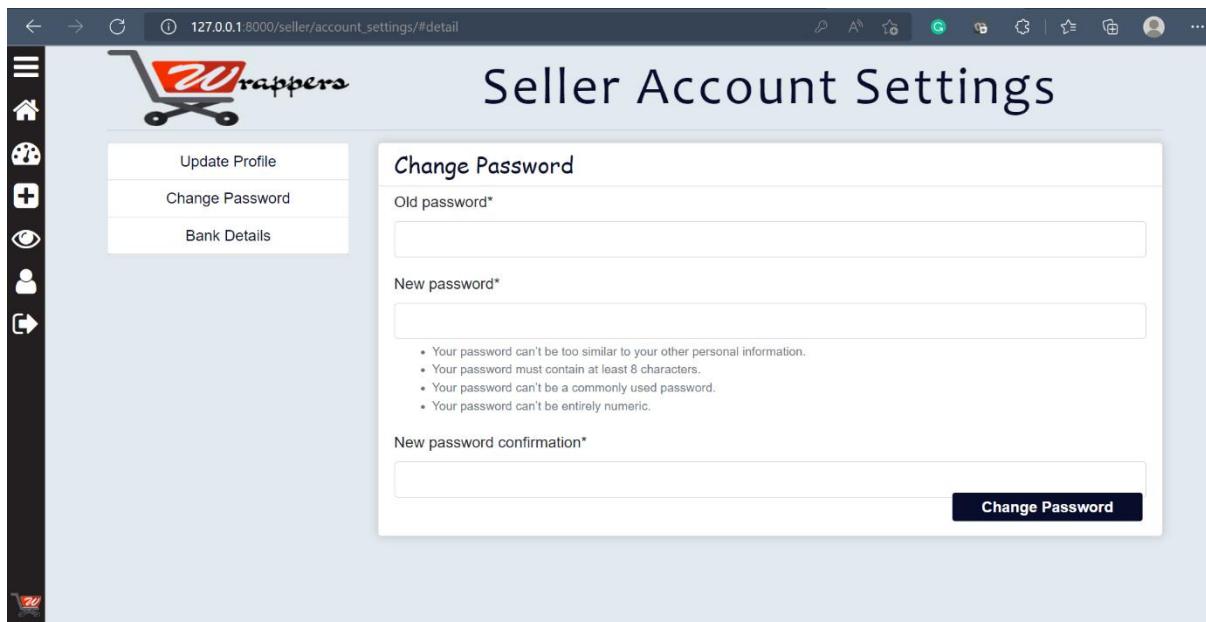


Figure:4.31 Seller Password Updating Page

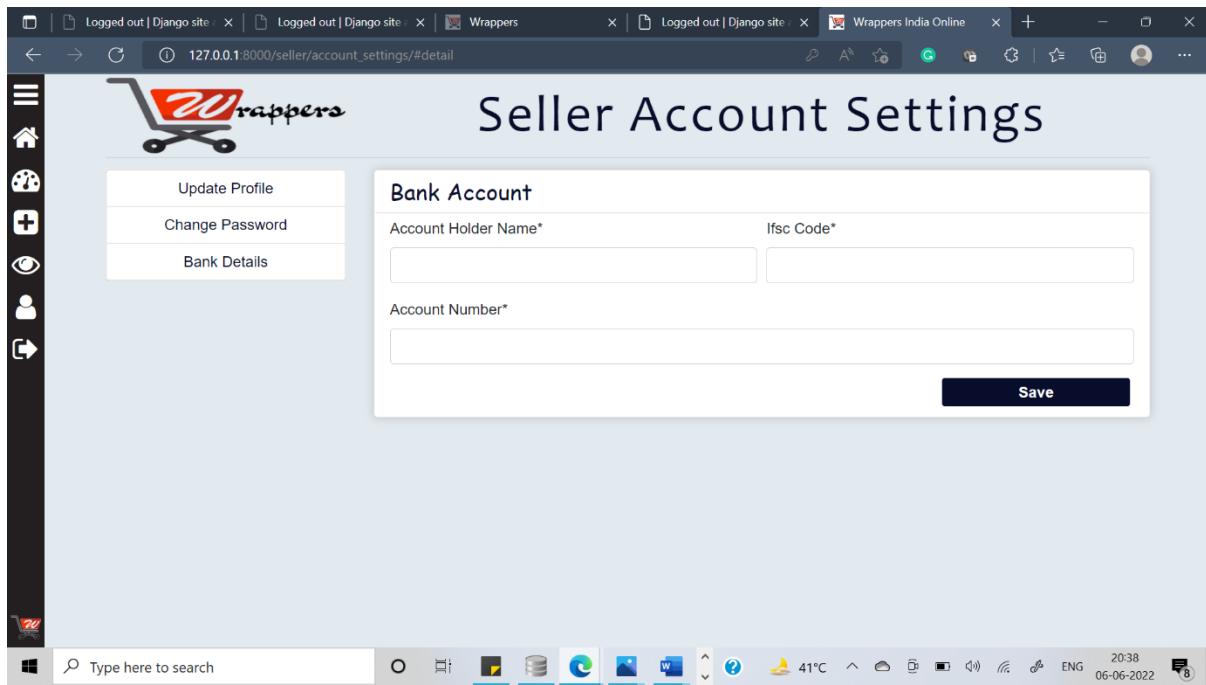
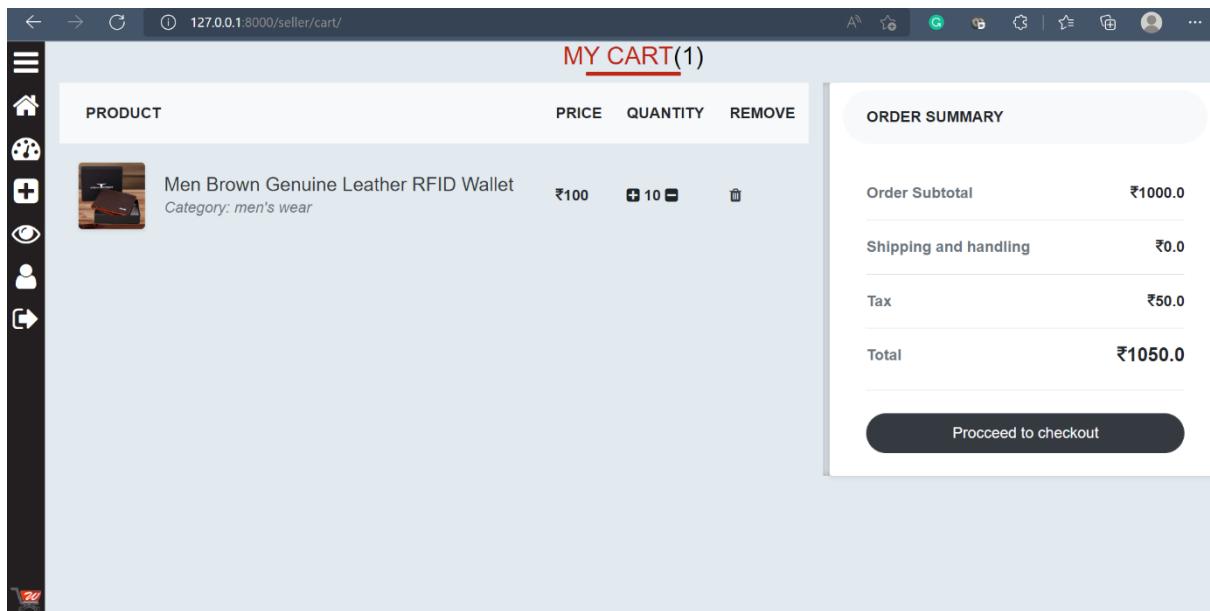


Figure:4.32 Seller Account Setting

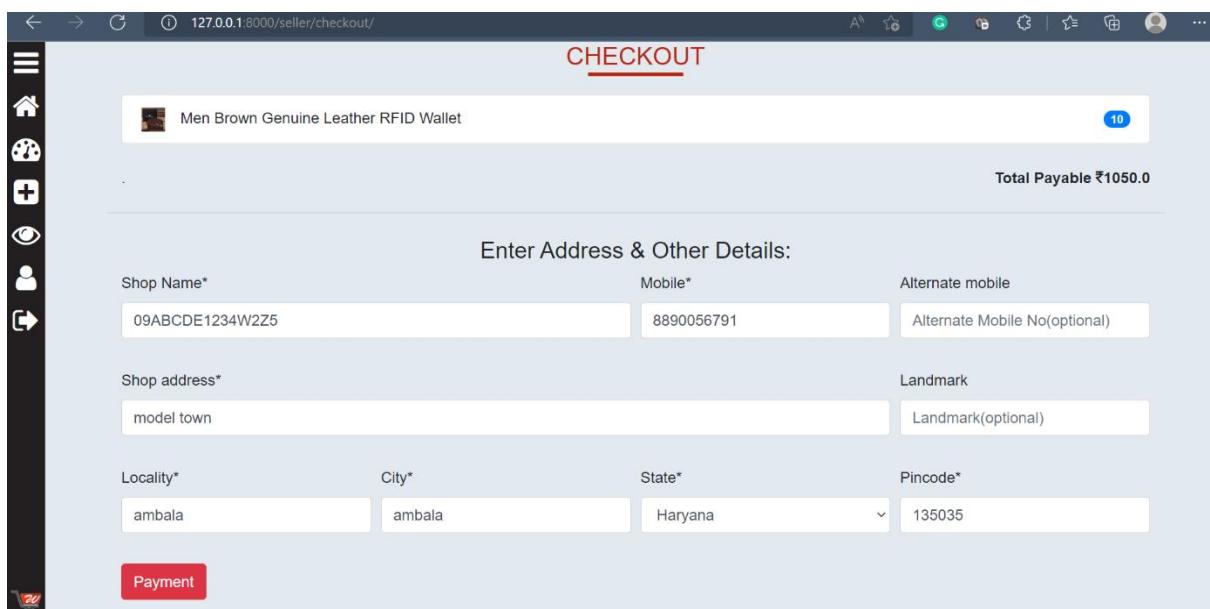


The screenshot shows the 'MY CART(1)' section of a seller's dashboard. On the left, there is a vertical sidebar with icons for Home, Add Product, Categories, Eye, User, and Logout. The main area has a header 'MY CART(1)'. Below it, a table lists a single item: 'Men Brown Genuine Leather RFID Wallet' with a price of ₹100 and a quantity of 10. To the right is an 'ORDER SUMMARY' table showing the breakdown of the total amount:

Order Subtotal	₹1000.0
Shipping and handling	₹0.0
Tax	₹50.0
Total	₹1050.0

A 'Proceed to checkout' button is located at the bottom right.

**Figure:4.33 Seller Cart View**



The screenshot shows the 'CHECKOUT' page. The top bar displays the URL '127.0.0.1:8000/seller/checkout/'. The main title is 'CHECKOUT'. Below it, a product summary shows 'Men Brown Genuine Leather RFID Wallet' with a quantity of 10. The total payable amount is listed as 'Total Payable ₹1050.0'. The page then asks for 'Enter Address & Other Details:' with fields for Shop Name\*, Mobile\*, Alternate mobile, Shop address\*, Landmark, Locality\*, City\*, State\*, Pincode\*, and Payment.

**Figure:4.34 Seller Checkout Page**

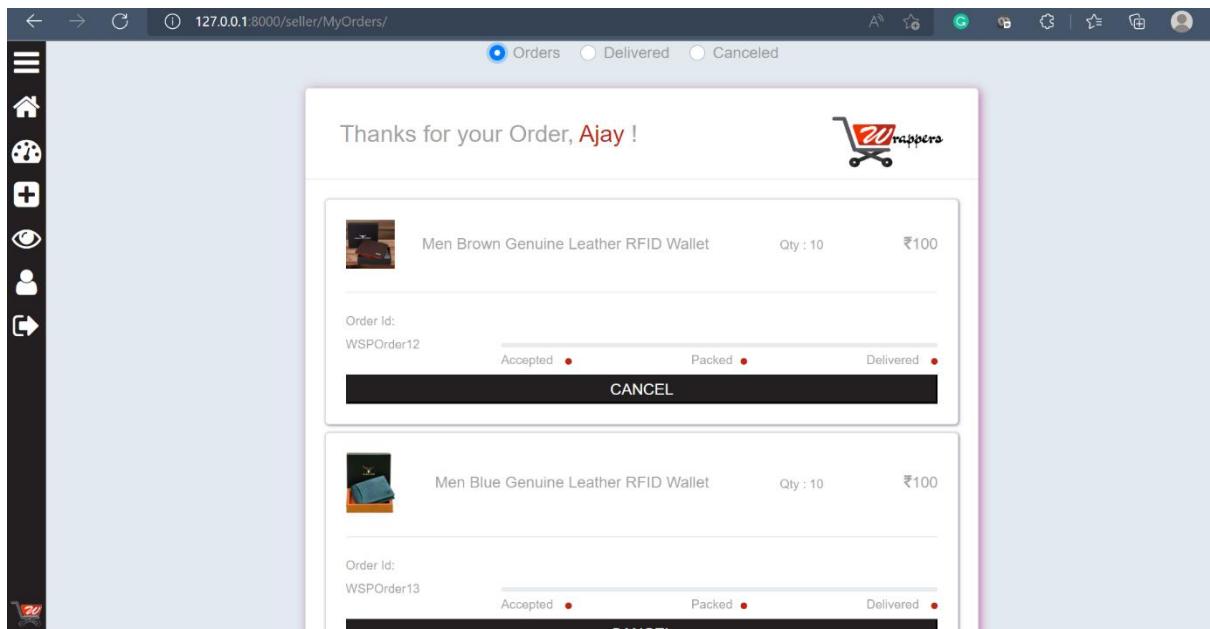


Figure:4.35 Seller Order Status Page

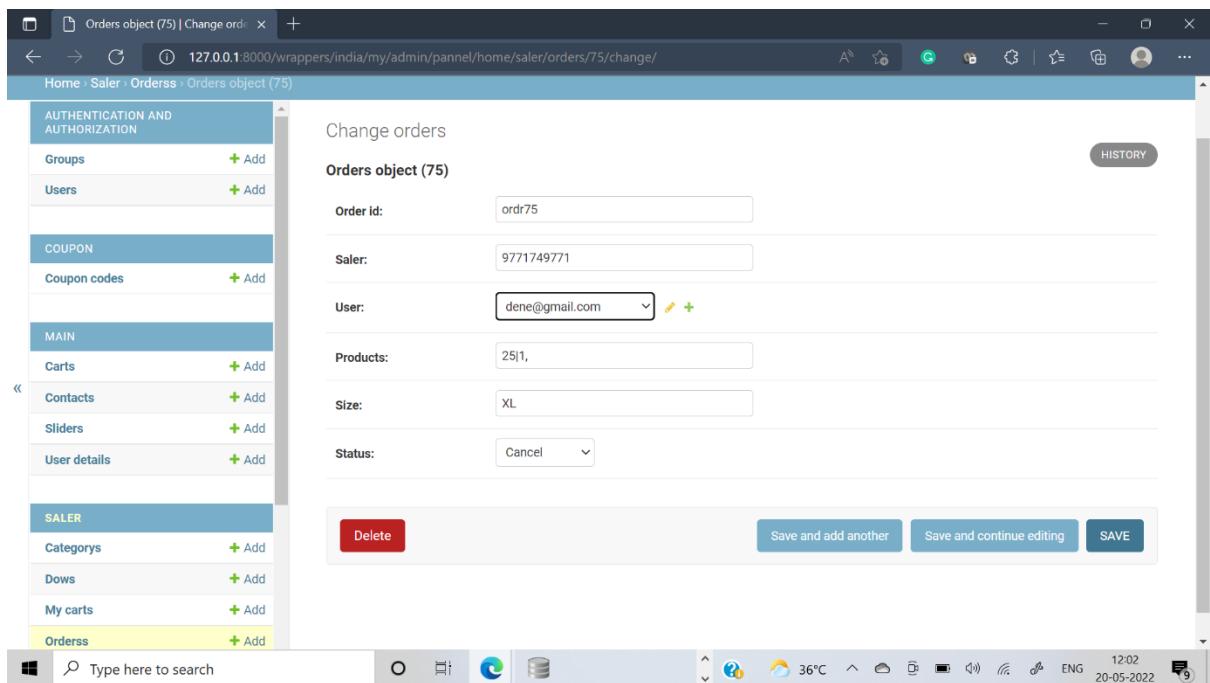


Figure:4.36 Admin Page

Successfully deleted 1 user.

Select user to change

Action:  Go 1 of 6 selected

	EMAIL ADDRESS	FIRST NAME	LAST NAME	STAFF STATUS
<input checked="" type="checkbox"/> 9771749771		Rajtosh	Ranjan	<span style="color: green;">✓</span>
<input type="checkbox"/> admin				<span style="color: green;">✓</span>
<input type="checkbox"/> ajay@gmail.com	ajay@gmail.com	Ajay	kumar	<span style="color: red;">✗</span>
<input type="checkbox"/> ajaykumar@gmail.com	ajaykumar@gmail.com	Ajay	kumar	<span style="color: green;">✓</span>
<input type="checkbox"/> dene@gmail.com	dene@gmail.com	pooja	rai	<span style="color: red;">✗</span>
<input type="checkbox"/> test@gmail.com	test@gmail.com	Ashish	Singh	<span style="color: red;">✗</span>

6 users

**FILTER**

By staff status

All  
Yes  
No

By superuser status

All  
Yes  
No

By active

All  
Yes  
No

**Figure:4.37 Crude Operation at Admin Site**

## Chapter 5

# CONCLUSION & FUTURE WORK

### CONCLUSION

The project entitled Wrapper e-commerce website 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 an e-commerce website. 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:

- In future we can use **artificial intelligence** to make our website more interactive and **machine learning technology** to provide product recommendation for users to buy.
- We want to add **customer support service** or **chatbot** through which user can find the answer of their problem and short out their product/service-related problems easily.
- We can also add **coupon** facilities to provide more discount on customer order.
- We can also add **super coins** facilities in our ecommerce website. User can use this coin in their product purchase. User can collect super coin by purchasing order.
- 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.
- We can build more **advanced order track system** to provide live location of order. And make prediction when the order will be delivered to customer.

## References

[E-commerce Website using Django - GeeksforGeeks](#)

[Bootstrap 4 Tutorial \(w3schools.com\)](#)

[Django Tutorial \(tutorialspoint.com\)](#)

[HTML Styles CSS \(w3schools.com\)](#)

[Learn jQuery Tutorial - javatpoint](#)

[JavaScript Tutorial](#)

[Django Tutorial for Beginners | Full Course - YouTube](#)

[Django Tutorial In Hindi - YouTube](#)

[Python Django 7 Hour Course - YouTube](#)

## PARTIAL CODE

```
from django.db import models
from django.contrib.auth.models import User
from PIL import Image

class UserDetail(models.Model):
    SEX_CHOICES = (( "Male", "Male"), ("Female", "Female"), ("Other", "Other"))
    STATE_CHOICES = (
        ("Andaman & Nicobar Islands", "Andaman & Nicobar Islands"),
        ("Andhra Pradesh", "Andhra Pradesh"),
        ("Arunachal Pradesh", "Arunachal Pradesh"),
        ("Assam", "Assam"),
        ("Bihar", "Bihar"),
        ("Chandigarh", "Chandigarh"),
        ("Chhattisgarh", "Chhattisgarh"),
        ("Dadra & Nagar Haveli", "Dadra & Nagar Haveli"),
        ("Daman and Diu", "Daman and Diu"),
        ("Delhi", "Delhi"),
        ("Goa", "Goa"),
        ("Gujarat", "Gujarat"),
        ("Haryana", "Haryana"),
        ("Himachal Pradesh", "Himachal Pradesh"),
        ("Jammu & Kashmir", "Jammu & Kashmir"),
        ("Jharkhand", "Jharkhand"),
        ("Karnataka", "Karnataka"),
        ("Kerala", "Kerala"),
        ("Lakshadweep", "Lakshadweep"),
        ("Madhya Pradesh", "Madhya Pradesh"),
        ("Maharashtra", "Maharashtra"),
        ("Manipur", "Manipur"),
        ("Meghalaya", "Meghalaya"),
        ("Mizoram", "Mizoram"),
        ("Nagaland", "Nagaland"),
```

```

        ("Odisha",'Odisha'), ("Puducherry",'Puducherry'), ("Punjab",'Punjab'),
        ("Rajasthan",'Rajasthan'),
        ("Sikkim",'Sikkim'),
        ("Tamil Nadu",'Tamil Nadu'),
        ("Telangana",'Telangana'),
        ("Tripura",'Tripura'),
        ("Uttarakhand",'Uttarakhand'),
        ("Uttar Pradesh",'Uttar Pradesh'),
        ("West Bengal",'West Bengal'),
    )

user = models.OneToOneField(User, on_delete=models.CASCADE,primary_key=True)
dob = models.DateField(null = True)
photo = models.ImageField(default='default.png',upload_to='user_photos')
mobile = models.CharField(max_length=10,null=True)
alternate_mobile = models.CharField(max_length=10,null=True,blank=True)
address = models.TextField()
pincode = models.CharField(max_length=6, null=True)
landmark = models.CharField(max_length=500, null=True, blank=True)
locality = models.CharField(max_length=100, null=True, blank=True)
city = models.CharField(max_length=100, null=True, blank=True)
state = models.CharField(max_length=50,choices=STATE_CHOICES, null=True)
sex = models.CharField(max_length=6,choices=SEX_CHOICES, null=True)

def save(self, *args, **kwargs):
    super().save(*args, **kwargs)

    img = Image.open(self.photo.path)
    if img.height > 300 or img.width > 300:
        output_size = (300, 300)
        img.thumbnail(output_size)
        img.save(self.photo.path)

class Slider(models.Model):
    name = models.CharField(max_length=50, default = "", null=True)

```

```

image = models.ImageField(upload_to='slider_img')
url = models.CharField(max_length=200, default = "#", null=True)

def __str__(self):
    return f'{self.name}'

def save(self, *args, **kwargs):
    super().save(*args, **kwargs)

    img = Image.open(self.image.path)
    if img.height > 1024 or img.width > 1024:
        output_size = (1024, 1024)
        img.thumbnail(output_size)
        img.save(self.image.path)

class Cart(models.Model):
    user = models.ForeignKey(User, on_delete=models.CASCADE)
    product_id = models.CharField(max_length=100)
    product_size = models.CharField(max_length=20,default="",null=True)
    number = models.PositiveIntegerField(default=0)
    razor_pay_order_id = models.CharField(max_length=100 ,null=True,blank=True)
    razor_pay_payment_id = models.CharField(max_length=100, null=True, blank=True)
    razor_pay_payment_signature = models.CharField(max_length=100, null=True,
    blank=True)

class Contact(models.Model):
    date = models.DateField(auto_now=True)
    name = models.CharField(max_length=100)
    email = models.EmailField()
    subject = models.CharField(max_length=100)
    message = models.TextField()

def __str__(self):
    return self.email

```