VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Jnana Sangama, Belagavi – 590018, Karnataka, India

A MINI PROJECT REPORT

ON

"ECOMMERCE WEBSITE"

A Mini Project Work Report Submitted in partial fulfillment of the requirement for the degree of

BACHELOR OF ENGINEERING

In

COMPUTER SCIENCE & ENGINEERING

Submitted by

BAHADURI PRACHITI JAGDISH	1RG17CS009
PRAJWAL B MANI	1RG17CS037

Under the Guidance of

Mrs. PRAGATHI M

Asst. Prof. Dept. of CSE
RGIT, Bengaluru32

Department of Computer Science & Engineering

RAJIV GANDHI INSTITUTE OF TECHNOLOGY

Cholanagar, R.T. Nagar Post, Bengaluru-560036

2020 - 2021

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

(Affiliated To Visvesvaraya Technological University)

Cholanagar, R.T.Nagar Post, Bangalore-560032

Department of Computer Science Engineering

CERTIFICATE

This is to certify that the Mini Project Report entitled **"ECOMMERCE WEBSITE"** is a bonafide work carried out by **Ms. Bahaduri Prachiti Jagdish (1RG17CS009) and Mr. Prajwal B Mani (1RG17CS037)** in partial fulfillment for the award of Bachelor of Engineering in Computer Science Engineering under Visvesvaraya Technological University, Belgavi, during the year 2020-2021. It is certified that all corrections/suggestions given for Internal Assessment have been incorporated in the report. This Mini Project report has been approved as it satisfies the academic requirements in respect of seminar work.

Signature of Guide	**Signature of HOD**
Mrs. Pragathi M	**Mrs. Arudra A**
Asst. Professor	**Professor & HOD**
Dept. of CSE	**Dept. of CSE**
RGIT, Bengaluru	**RGIT, Bengaluru**

External Viva

Name of the Examiners	**Signature With Date**
1. ______________________	
2. ______________________	

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

JNANA SANGAMA, BELAGAVI-590018

RAJIV GANDHI INSTITUTE OF TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

DECLARATION

We hereby declare that the mini project work entitled "ECOMMERCE WEBSITE" **Visvesvaraya Technological University, Belagavi** during the academic year **2020-2021**, is record of an original work done by us under the guidance of **Mrs. Pragathi M, Assistant Professor, Department of Computer Science and Engineering, Rajiv Gandhi Institute of Technology, Bengaluru** and this project work is submitted in the partial fulfillment of requirements for the award of the degree of **Bachelor of Engineering in Computer Science** & **Engineering**. The results embodied in this thesis have not been submitted to any other University or Institute for award of any degree or diploma.

BAHADURI PRACHITI JAGDISH (1RG17CS009)

PRAJWAL B MANI (1RG17CS037)

ACKNOWLEDGEMENT`

We take this opportunity to express our sincere gratitude and respect to the **Rajiv Gandhi Institute of Technology, Bengaluru** for providing us an opportunity to carry out our project work.

We express our sincere regards and thanks to **Dr. NAGRAJ A M, Principal, RGIT, Bengaluru** and **Mrs. Arudra A, Professor and Head, Department of Computer Science & Engineering, RGIT, Bengaluru,** for her encouragement and support throughout the Project.

With profound sense of gratitude, we acknowledge the guidance and support extended by **Mrs. Pragathi M, Asst. Prof, Department of Computer Science & Engineering, RGIT, Bengaluru.** Her incessant encouragement and valuable technical support have been of immense help in realizing this project. Her guidance gave us the environment to enhance our knowledge, skills and to reach the pinnacle with sheer determination, dedication and hard work.

We also extend our thanks to the entire faculty of the Department of CSE, RGIT, Bengaluru, who have encouraged us throughout the course of Bachelor Degree.

BAHADURI PRACHITI JAGDISH (1RG17CS009)

PRAJWAL B MANI (1RG17CS037)

ABSTRACT

In this project, we are trying to replicate an Ecommerce website using the Django framework. Irrespective of whether the user is logged in or not the user can add products to the cart and the session will be saved. Then for the checkout step, the user is asked for the authentication process. We have used SQLite for the database with the CRUD approach. The products are displayed as card elements. The user can search for the products using our search bar and the products are retrieved based on keywords in the title and description of the products. The cart is updated every time when the user alters the cart items. Every product and the user has a unique id so it's easy to track furthermore even we have a unique slug URLs generator for the products.

TABLE OF CONTENTS

LIST OF TABLES AND FIGURES

INTRODUCTION

Chapter 1

INTRODUCTION

Our project is titled **"Ecommerce Website"**. This is an online project developed using HTML and Django web framework. The purpose of this project is to provide a portal for users wherein they can comfortably and securely purchase products. To make effective use the website, every user who is new to the website has to register first in order to lookout for any kind of products to buy. After registering and logging in as the users, they can choose to put items in the cart. The user wishing to buy the products can click on the lookout option, and can enter the details of their shipping address. Once this task is done, the product including its image will be transferred and stored in the database. The details of the product to be delivered are then sent to an admin and the user will be redirected to the success page. If the user wants to continue buying the products, they can simply go back to Products page and add the desired product to the cart. There is also a feature in our website where a user may wish to remove the products from the cart, if not required. Each product is displayed with its price along with the description set by the admin. Once the user buys a certain product, the user can go back to the Products page to buy any other product the user wants to buy. Our system is more secure as we provide access to the authenticated users via admin.

1.1 ABOUT THE PROJECT

The amount of researching that goes into finding the right product is massive if the products are not properly organised. Nowadays the user don't have to go the shopping place, yet they only need to turn on their computers and enter the website at any time. Online shopping had an initial reach that far outgrew the expectations of anyone. This is mainly because online shopping has broken down and removed the physical limitations of buying such as geography, presence, time, space, and a small target audience. It is an implementation of shopping webistes, like Amazon, Flipkart, and others with their own UI and database. This website has an intuitive interface and unique visual objects that is user friendly. Common client operations were implemented in this application. In addition, we used an authentication system to reduce spam. There are 6 common pages when it comes to online shopping such as Home page, Login page, Register page, Cart page, Contact page, Products page. This is because we aim to maximize the profit earned by each seller. This even gives the familiarity of a real life shopping

1.2 SCOPE

The main purpose of Ecommerce website is to provide a simpler and more secure site for users to choose from a wide range of products and put them in cart in case of purchasing. It also provides an ability for admins to track new products to be added in the products page. These product entries will then be updated in the database. The admin will then have an option to add or delete products if required.

The scope and reach of shopping on ecommerce website have been propelled by the Internet to a level beyond what the initial purveyors had anticipated. Ecommerce website include Business-to-Business (B2B), Business-to-Consumer (B2C), and Consumer-to-Consumer (C2C), Business-to-Administration (B2A), Consumer-to-Administration (C2A). Ecommerce websites have greatly increased the availability of variety of goods and services that can be purchased along with expanding the possibilities for the ways to promote this kind of shopping. In the current web environment there are hundreds, if not thousands, of websites dedicated to online shopping.

1.3 OBJECTIVES

The purpose of this project is to provide the complete information about the products that are available for sale. There are 2 different types of pages. First the user visits the site and enters the crendentials such as the username and password of the user. Then these details will be stored in the webiste database controlled by the admin. The admin of E – Commerce webiste will then login and update all the products whichever needs to removed or added to the jargon. If the user deems to be valid the user will be approved by the website. However the user will be rejected if wrong password or username is entered and that particular user will have to register first before logging in the website. Then the users are provided with the different products to purchase from in the products page which the customer can choose from and the user will add the desired product to the cart. If the customer wishes to buy the product, he will go the cart and confirm the products to be brought. Once the user is confirmed to buy the products in the cart, the user will click on the lookout option and the user will be sent to the page where he will have to enter the destination for the delivery of their product. Once the address is confirmed and the order is placed, it will redirect the user to success page and will send a confirmation message to the customer.

E – Commerce website is a good system for the manufacturing industry that provides feature of buy the product hassel – free. The system incentivizes users to buy more on a product as the chances of them acquiring the product is higher that way.

The goals of our system are:

- To provide a service to any user in any given location.
- To provide a secure way to buy their products.
- To protect the buyers from unnecessary spam.
- To provide a hassle-free system for buying.

1.4 OVERVIEW

The project has been successfully developed and the system performance is quite efficient. The buyer's uncertainty towards product makes it difficult for the buyers to differentiate between the good and the bad products, the lack of differentiation may force higher quality products to sell out easily from the market since their quality is significantly better compared to products do not signal and reward with fair prices thus reducing transaction activity. What this new system is trying to accomplish is to create a higher level of buyer's certainty on the type of the products that they choose to make buy. Through the use of effective information like the use of visual and textual product description, product price value and product usage. The successful implementation of this project results in an online shopping system that allows quality of the product that is far much effective and that comes close or equal the physical quality of the product.

REQUIREMENT SPECIFICATION

Chapter 2

SYSTEM REQUIREMENTS

To be used efficiently, all computer software needs certain hardware components or other software resources to be present on a computer, these prerequisites are known as system requirements. System Requirements only tell you what your system must have and what it must allow users to do.

The system requirements are of two types:

- Hardware Requirments
- Software Requirements

2.1 HARDWARE REQUIREMENTS

- **Processor**: **Intel® Core i3-1710 CPU @ 2.40GHz** – Intel Core is a line of mid- to-high end consumer, workstation, and enthusiast central processing units (CPU) marketed by Intel Corporation. These processors displaced the existing mid-to- high end Pentium processors of the time, moving the Pentium to the entry level, and bumping the Celeron series of processors to low end. Information processor, a system which takes information in one form and transforms it into another form by an algorithmic process. Data processing system, a combination of machines, people, and processes that for a set of inputs produces a defined set of outputs. Information system, a system composed of people and computers that processes or interprets information. Identical or more capable versions of Core processors are also sold as Xeon processors for the server and workstation markets.

- **Memory(RAM)** : **4.00 GB** – Random – access memory is a form of computer data storage that stores data and machine code currently being used. Memory is not a perfect processor, and is affected by many factors. The manner information is encoded, stored, and retrieved can all be corrupted. A random-access memory device allows data items to be read or written in almost the same amount of time irrespective of the physical location of data inside the memory. In contrast, with other direct-access data storage media such as hard disks, CD-RWs, DVD- RWs and the older magnetic tapes and drum memory, the time required to read and write data items varies significantly depending on their physical locations on the recording medium, due to mechanical limitations such as media rotation speeds and movement.

- **System type** : **64-bit Operating System** - An operating system (OS) is a system software that manages computer hardware and software resources and provides common services for computer programs. The system type defines the system that can perform the desired operations. Time-sharing operating systems schedule tasks for efficient use of the system and may also include accounting software for cost allocation of processor time, mass storage, printing, and other resources.
- **Hard Disk Capacity**: **500 GB** - A hard disk drive (HDD), hard disk, hard drive, or fixed drive is an electromechanical data storage device that uses magnetic storage to store and retrieve digital information using one or more rigid rapidly rotating disks (platters) coated with magnetic material.

2.2 SOFTWARE REQUIREMENTS

For Website:

- **Technology Implemented: Web Technology:** The World Wide Web (WWW), commonly known as the Web, is an information system where documents and other web resources are identified by Uniform Resource Locators like https://www.example.com/ , which may be interlinked by hypertext, and are accessible over the Internet. The resources of the Web are transferred via the Hypertext Transfer Protocol (HTTP), may be accessed by users by a software application called a web browser, and are published by a software application called a web server
- **Language Used: Python 3.7.4** - Python is an interpreted, high-level and general-purpose programming language. Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object- oriented approach aim to help programmers write clear, logical code for small and large-scale projects.
- **Database : Django (web framework) 3.1.3** – Django is a Python-based free and open-source web framework that follows the model-template-views (MTV) architectural pattern. Django's primary goal is to ease the creation of complex, database-driven websites. The framework emphasizes reusability and "pluggability" of components, less code, low coupling, rapid development, and the

principle of don't repeat yourself. Django also provides an optional administrative create, read, update and delete interface that is generated dynamically through introspection and configured via admin models.

- **User Interface Design** : **HTML 5, CSS, BOOTSTRAP -** Bootstrap is a free and open source front end frame work for designing web sites and web applications. It contains HTML and CSS based design templates for typography, forms, buttons, navigation and other interface components, as well as optional JavaScript extensions. Unlike many earlier web frameworks, it concerns itself with front-end development only.
- **Hypertext Markup Language (HTML)** is the standard markup language for creating web pages and web applications. With Cascading Style Sheets (CSS) and JavaScript, it forms a triad of cornerstone technologies for the World Wide Web. Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page. HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items.
- **Cascading Style Sheets (CSS)** is a style sheet language used for describing the presentation of a document written in a markup language like HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript. CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts of the style sheets. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple web pages to share formatting by specifying the relevant CSS in a separate css file, and reduce complexity and repetition in the structural content. Separation of formatting and content also makes it feasible to present the same markup page in different styles for different rendering

methods, such as on- screen, in print, by voice (via speech-based browser or screen reader), and on Braille-based tactile devices. CSS also has rules for alternate formatting if the content is accessed on a mobile device.

For Software Product:

- **Operating System: Windows 10** - An operating system is system software that manages computer hardware and software resources and provides common services for computer programs.

2.3 USER REQUIREMENTS

User requirements only describe how user expectations are and how they will interact with the product. Requirements do not tell you how to design or develop the site to have those features, functions and contents.

The expectations of user may be:

- Easy to understand and should be simple.
- Access to the system should be specific and easy with reduced maintenance.
- Any changes to the system must be allowable.

SYSTEM DESIGN

Chapter 3

SYSTEM DESIGN

3.1 THREE - TIER ARCHITECTURE

In software engineering, multitier architecture (often referred to as n-tier architecture) or multilayered architecture is a client–server architecture in which presentation, application processing, and data management functions are physically separated. The most widespread use of multitier architecture is the three-tier architecture. N-tier application architecture provides a model by which developers can create flexible and reusable applications. A three-tier architecture is typically composed of a presentation tier, a domain logic tier, and a data storage tier. While the concepts of layer and tier are often used interchangeably, one fairly common point of view is that there is indeed a difference.

- Presentation layer**-** the top-most level of the application is the interface. The main function is the user interface. The main function of the interface is to translate tasks and results to something the user can understand.
- Logic layer**-** this layer coordinates the application. Processes commands, makes logical decisions and evaluations, and performs calculations. It also moves and processes data between the two surrounding layers.
- Data layer**-** here information is stored and retrieved from a database or file system. The information is then passed back to the logic layer for processing, and then eventually back to the user. Our entire project is built on 3-tier architecture.

This view holds that a layer is a logical structuring mechanism for the elements that make up the software solution, while a tier is a physical structuring mechanism for the system infrastructure. For example, a three-layer solution could easily be deployed on a single tier, such as a personal workstation. Our entire project is built on 3-tier architecture.

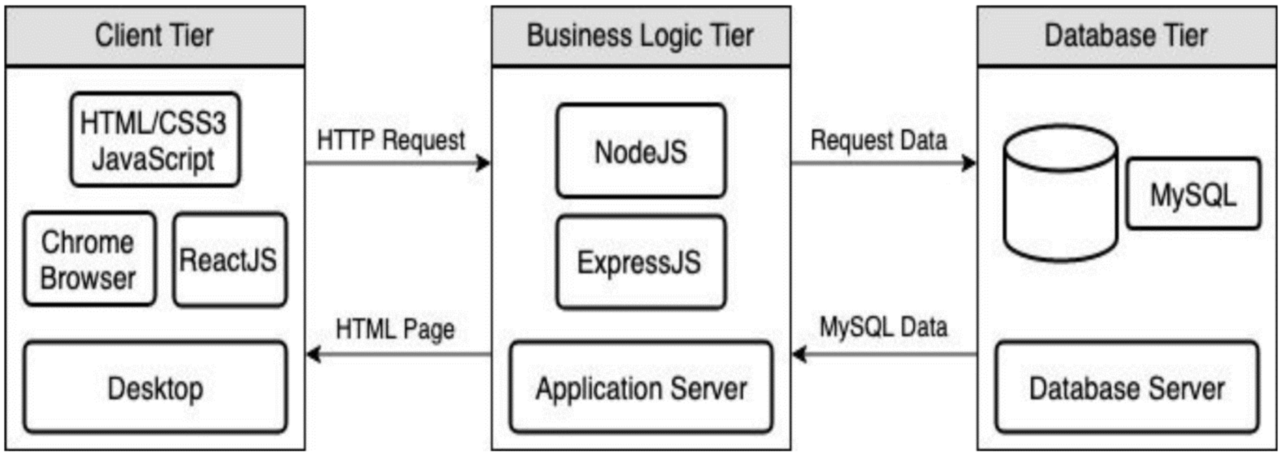


Figure: 3.1.1 Three – Tier Architecture

SYSTEM IMPLEMENTATION

Chapter 4

SYSTEM IMPLEMENTATION

4.1 urls.py

```
from django.conf.urls import url,include
from django.contrib import admin
from django.conf import settings
from django.conf.urls.static import static
from django.views.generic import TemplateView
from django.contrib.auth.views import LogoutView
from accounts.views import login_page,register_page,guest_register_view
from .views import home_page,about_page,contact_page
from carts.views import cart_home
from addresses.views import checkout_address_create_view,checkout_success
urlpatterns = [
    url(r'^admin/', admin.site.urls),
    url(r'^$', home_page,name='home'),
    url(r'^about/$', about_page,name='about'),
    url(r'^contact/$', contact_page,name='contact'),
    url(r'^login/$', login_page,name='login'),
    url(r'^logout/$', LogoutView.as_view(),name='logout'),
    url(r'^checkout/address/create/$',
checkout_address_create_view,name='checkout_address_create_view'),
    url(r'^register/$', register_page,name='register'),
    url(r'^products/',include("products.urls",namespace='products')),
    url(r'^search/',include("search.urls",namespace='search')),
    url(r'^cart/',include("carts.urls",namespace='cart')),
    url(r'^bootstrap/$',TemplateView.as_view(template_name='bootstrap/example.html')),
    url(r'^resgister/guest/$', guest_register_view,name='guest_register_view'),
]
if settings.DEBUG:
    urlpatterns = urlpatterns + static(settings.STATIC_URL, document_root=settings.STATIC_ROOT)
    urlpatterns = urlpatterns + static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
```

4.2 Login Page

```
{% extends "base.html" %}
{% block content %}
<form method="POST">  {% csrf_token %}
 <h1>Login</h1>
    {{form}}
      <button type="submit" class="btn btn-success">Submit</button>
  </form>
 {% endblock %}
```

4.3 Register Page

```
{% extends "base.html" %}
{% block content %}
<form method="POST"> {% csrf_token %}
  <h1>Register</h1>
    {{form}}
      <button type="submit" class="btn btn-primary">Submit</button>
  </form>
  {% endblock %}
```

4.4 view.html

```
{% extends "base.html" %}
{% block content %}
<!-- <div class="container"> -->
    <div class="row  mb-3">
    {% if query %}
    <div class="col-12">
        <h3>Results for <b>{{ query }}</b><hr/></h3>
    </div>
    {% else %}
      <div class="col-12 col-md-6 mx-auto py-5">
        {% include 'search/snippets/search-form.html' %}
    </div>
      <div class="col-12">
```

```
          <hr>
      </div>
      {% endif %}
    </div>
<div class="row">
   {% for obj in object_list %}
   <div class="col">
      <!-- {{ forloop.counter }} -->
      {% include 'products/snippets/card.html' with instance=obj %}
      {% if forloop.counter|divisibleby:3 %}
   </div></div><div class="row"><div class="col-12"><hr/></div>
   {% elif forloop.counter|divisibleby:2 %}
   </div></div><div class="row"><div class="col-12"><hr/></div>
   {% else %}
</div>
 {% endif %}
     {% endfor %}
    </div>
<!-- </div> -->
{% endblock %}
```

4.5 cart.html

```
{% extends "base.html" %
{% block content %}
<h1>Cart</h1>
{% if cart.products.exists %}
<table class="table">
    <thead>
      <tr>
        <th scope="col">#</th>
        <th scope="col">Product Name</th>
        <th scope="col">Price</th>
      </tr>
```

```
   </thead>
   <tbody>
      {% for product in cart.products.all %}
     <tr>
       <th scope="row">{{ forloop.counter }}</th>
       <td> <a href="{{ product.get_absolute_url }}">{{ product.title  }}</a>
         {% include 'products/snippets/update-cart.html' with product=product cart=cart in_cart=True %}
       </td>
       <td>{{ product.price  }}</td>
     </tr>
     {% endfor %}
     <tr>
       <th colspan="2"></th>
       <td><b>Subtotal</b> {{ cart.subtotal }}</td>
     </tr>
     <tr>
       <th colspan="2"></th>
       <td><b>Total</b> {{ cart.total }}</td>
     </tr>
     <tr>
       <th colspan="2"></th>
       <td><a class='btn btn-lg btn-success'href='{% url "cart:checkout" %}'>Checkout</a></td>
     </tr>
   </tbody>
  </table>
{% else %}
<p class="lead">Cart is empty</p>
{% endif %}
{% endblock %}
```

4.6 search_view.py

```
{% extends "base.html" %}
{% block content %}
```

```
<h1>Cart</h1>
{% if cart.products.exists %}
<table class="table">
  <thead>
    <tr>
      <th scope="col">#</th>
      <th scope="col">Product Name</th>
      <th scope="col">Price</th>
    </tr>
  </thead>
  <tbody>
      {% for product in cart.products.all %}
    <tr>
      <th scope="row">{{ forloop.counter }}</th>
      <td> <a href="{{ product.get_absolute_url }}">{{ product.title  }}</a>
        {% include 'products/snippets/update-cart.html' with product=product cart=cart in_cart=True %}
      </td>
      <td>{{ product.price  }}</td>
    </tr>
    {% endfor %}
    <tr>
      <th colspan="2"></th>
      <td><b>Subtotal</b> {{ cart.subtotal }}</td>
    </tr>
    <tr>
      <th colspan="2"></th>
      <td><b>Total</b> {{ cart.total }}</td>
    </tr>
    <tr>
      <th colspan="2"></th>
      <td><a class='btn btn-lg btn-success'href='{% url "cart:checkout" %}'>Checkout</a></td>
    </tr>
    </tbody>
  </table>
{% else %}
<p class="lead">Cart is empty</p>
{% endif %}
{% endblock %}
```

4.7 products_views.py

```
from django.shortcuts import render,get_object_or_404
from django.http import Http404
from django.views.generic.list import ListView
from django.views.generic.detail import DetailView
from .models import Product
from carts.models import Cart

class ProductFeaturedListView(ListView):
    template_name="products/list.html"
    def get_queryset(self,*args,**kwargs):
        request=self.request
        return Product.objects.all().featured()

class ProductDetailSlugView(DetailView):
    queryset=Product.objects.all()
    template_name="products/detail.html"

    def get_context_data(self,*args,**kwargs):
        context=super(ProductDetailSlugView,self).get_context_data(*args,**kwargs)
        cart_obj,new_obj=Cart.objects.new_or_get(self.request)
        context['cart']=cart_obj
        return context

    def get_object(self,*args,**kwargs):
        request=self.request
        slug=self.kwargs.get('slug')
        # instance=get_object_or_404(Product,slug=slug,active=True)
        try:
            instance=Product.objects.get(slug=slug ,active=True)
        except Product>doesNotExist:
            raise Http404("Not found..")
        except Product.MutipleObjectsReturned:
            qs=Product.objects.filter(slug=slug,active=True)
            instance= qs.first()
        except:
            raise Http404("Uhmm!")
        return instance

class ProductFeaturedDetailView(DetailView):
    queryset=Product.objects.featured()
    template_name="products/featured-detail.html"

class ProductListView(ListView):
    # queryset=Product.objects.all()
    template_name="products/list.html"
```

4.8 billing_model.py

```
from django.db import models
from django.conf import settings
from django.db.models.signals import post_save
from accounts.models import GuestEmail
User=settings.AUTH_USER_MODEL
class BillingProfileManager(models.Manager):
    def new_or_get(self,request):
        user=request.user
        quest_email_id=request.session.get('guest_email_id')
        created=False
        obj=None
        if user.is_authenticated():
            obj,created=self.model.objects.get_or_create(user=user,email=user.email)
        elif quest_email_id is not None:
            guest_email_obj=GuestEmail.objects.get(id=guest_email_id)
            obj,created=self.model.objects.get_or_create(email=guest_email_obj.email)
        else:
            pass
        return obj,created

class BillingProfile(models.Model):
    user = models.OneToOneField(User, null=True, blank=True)
    email = models.EmailField()
    active = models.BooleanField(default=True)
    update = models.DateTimeField(auto_now=True)
    timestamp = models.DateTimeField(auto_now_add=True)

    objects=BillingProfileManager()
    def __str__(self):
        return self.email

def user_created_receiver(sender,instance,created,*args,**kwargs):
    if created and instance.email:
        BillingProfile.objects.get_or_create(user=instance,email=instance.email)

post_save.connect(user_created_receiver,sender=User)
```

TESTING

Chapter 5

TESTING

5.1 TEST CASE 1:

The test case describes the test conditions where the admin logs-in and enters the username and the respective password with numbers, a special character and some characters.

Username	Password	Status
mani	mani1234	Fail
mani	mani	Pass

Table: 5.1 Testcase 1

5.2 TEST CASE 2:

The test case describes the test condition where we will be entering a word in our search bar and after that the query is executed and the products are retrieved and display on the website.

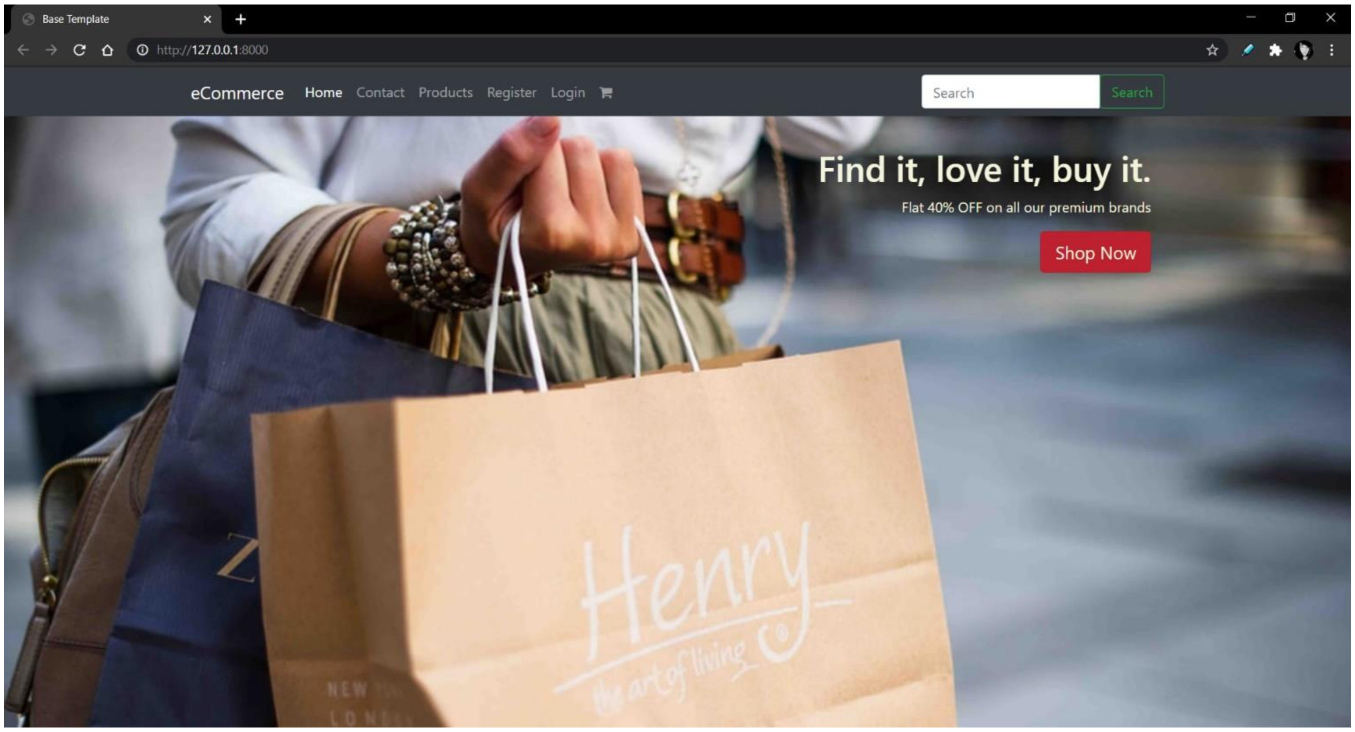
Tags	Retrieved YES/NO
Book	YES
Car	NO

Table: 5.2 Testcase 2

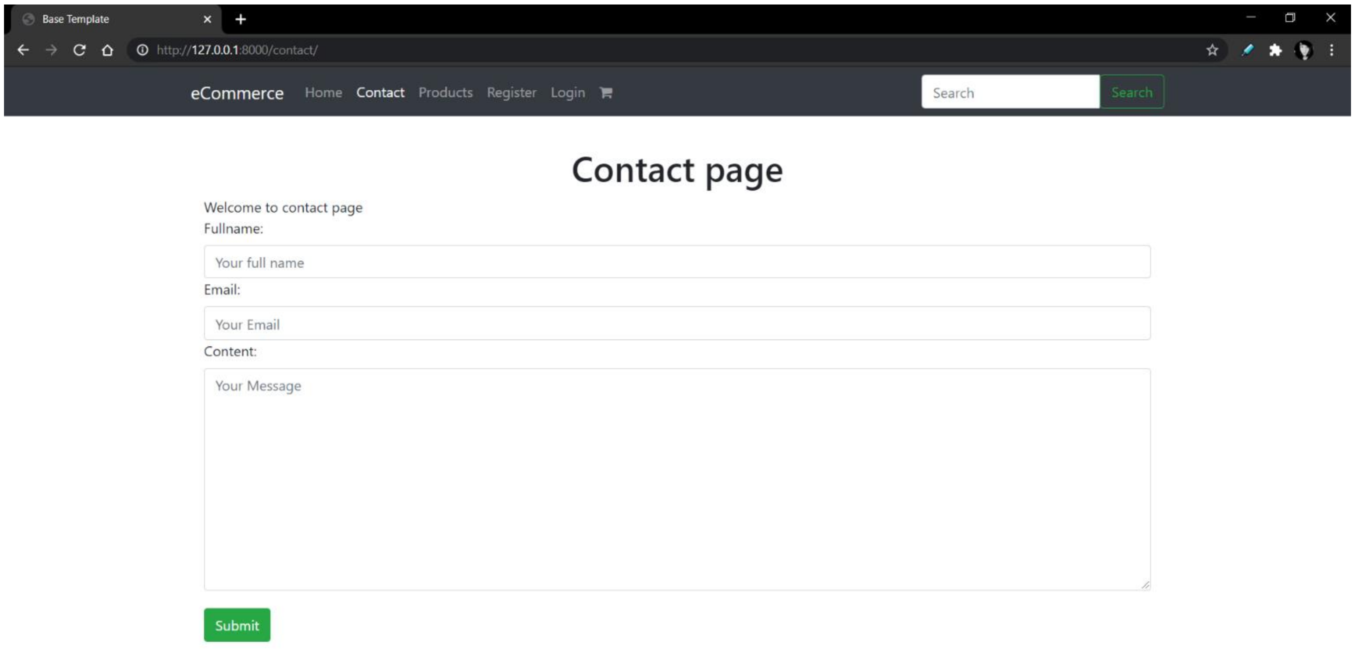
SCREENSHOTS

Chapter 6

SCREENSHOTS



Screenshot 6.1: Home Page



Contact page

Welcome to contact page

Fullname:

Your full name

Email:

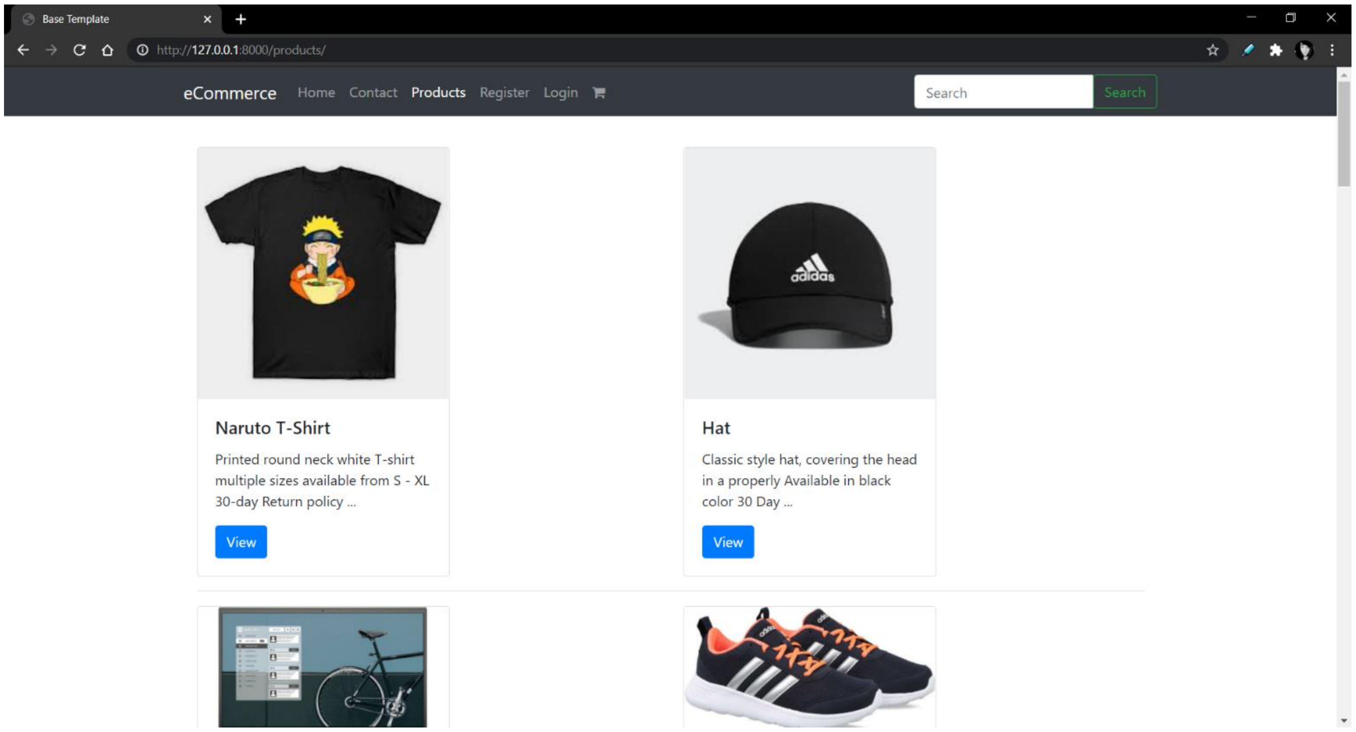
Your Email

Content:

Your Message

Submit

Screenshot 6.2: Contact Page



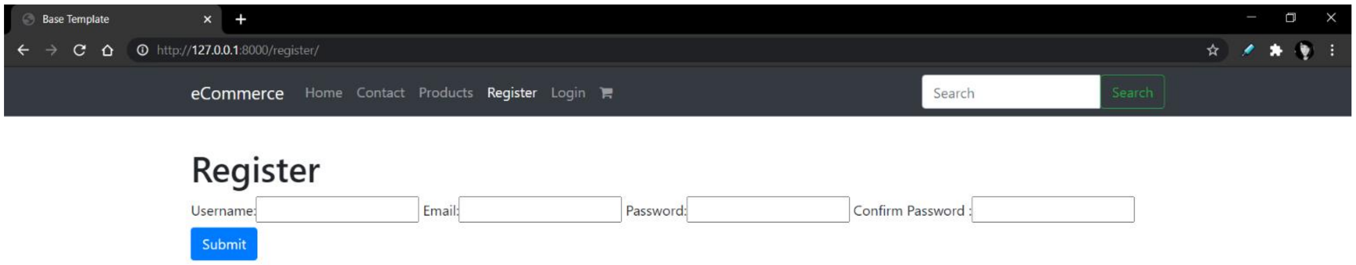
Naruto T-Shirt

Printed round neck white T-shirt multiple sizes available from S - XL 30-day Return policy ...

Hat

Classic style hat, covering the head in a properly Available in black color 30 Day ...

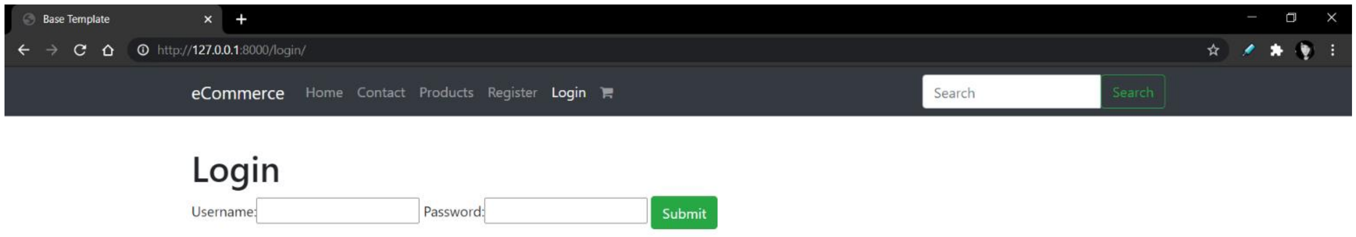
Screenshot 6.3: Products Page



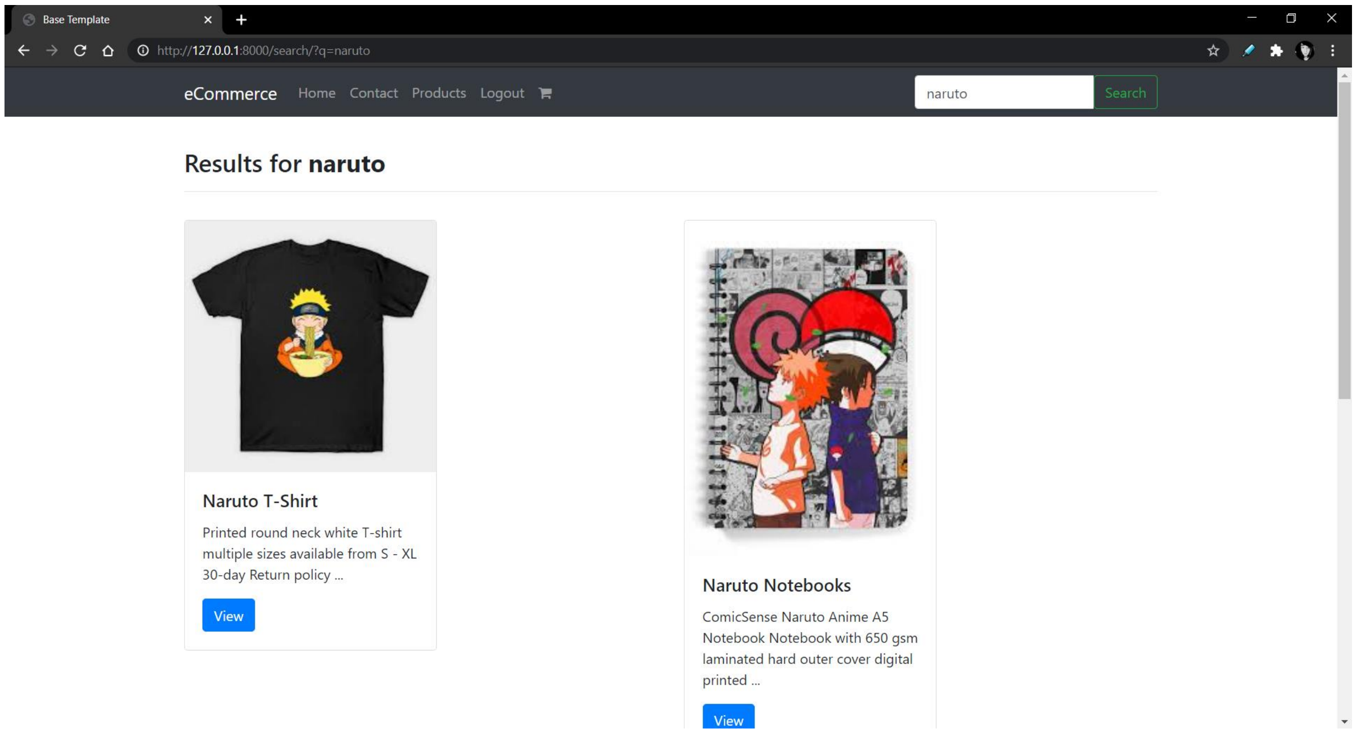
Register

Username: Email: Password: Confirm Password :

Screenshot 6.4: Register Page


Base Template
http://127.0.0.1:8000/login/
eCommerce Home Contact Products Register Login
Search
Search
Login
Username:
Password:
Submit

Screenshot 6.5: Login Page


Base Template
http://127.0.0.1:8000/search/?q=naruto
eCommerce Home Contact Products Logout
naruto
Search
View
View

Results for **naruto**

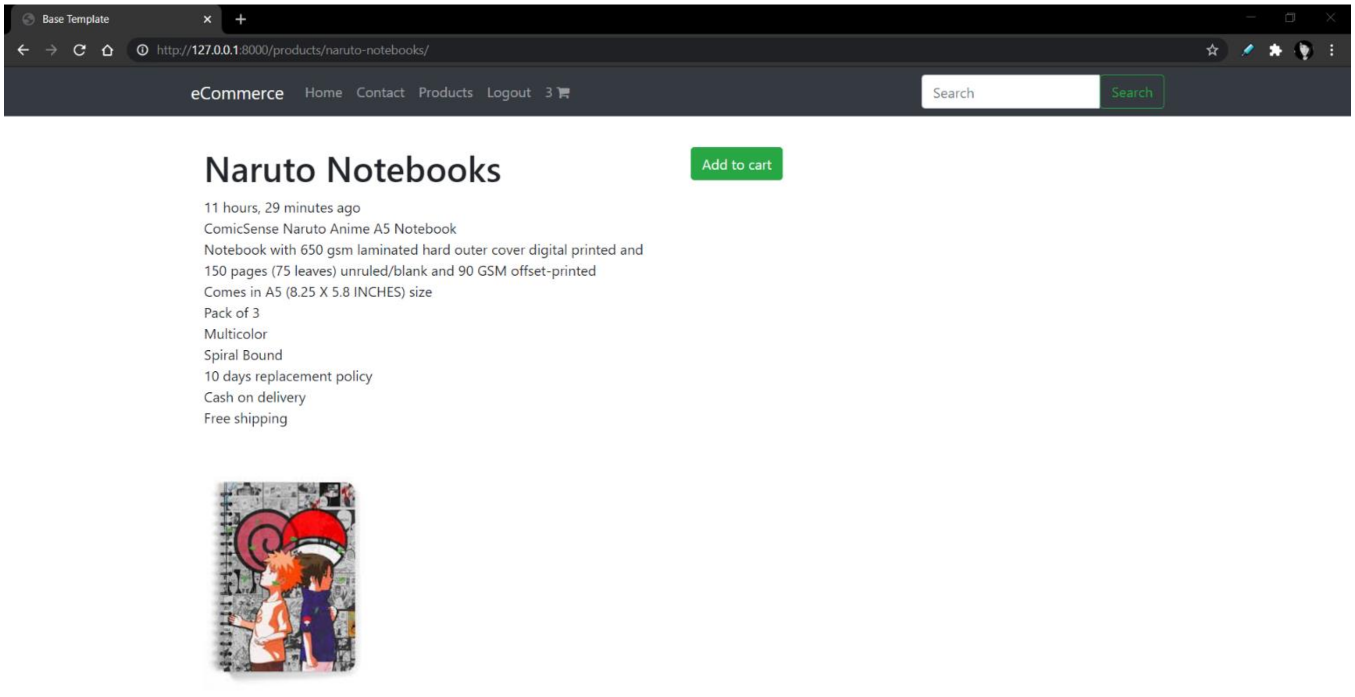
Naruto T-Shirt

Printed round neck white T-shirt multiple sizes available from S - XL 30-day Return policy ...

Naruto Notebooks

ComicSense Naruto Anime A5 Notebook Notebook with 650 gsm laminated hard outer cover digital printed ...

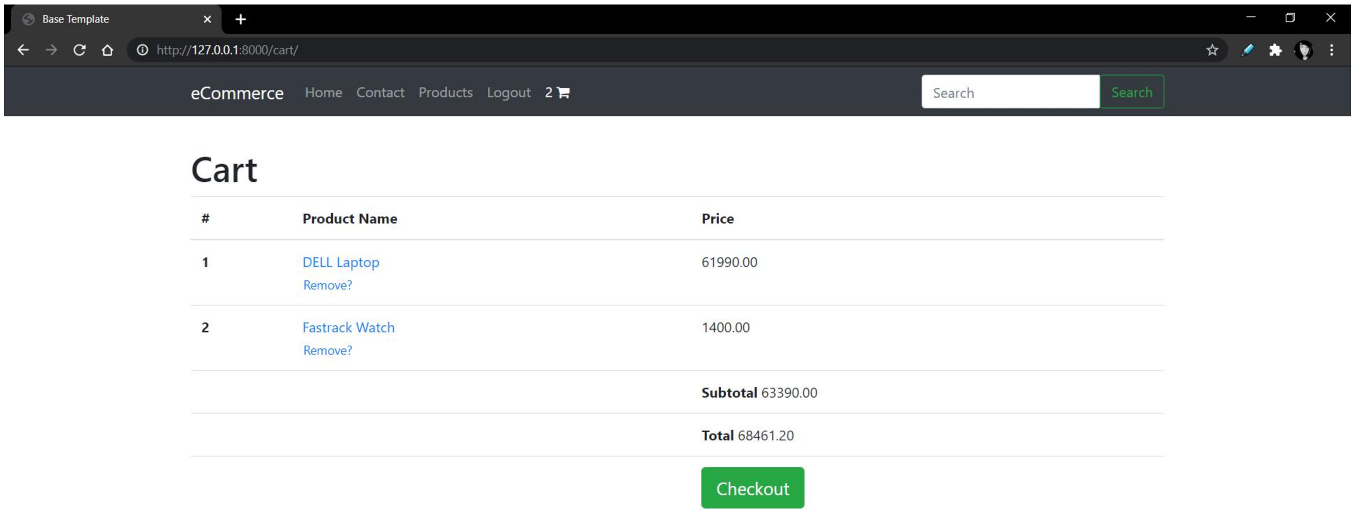
Screenshot 6.6: Results Page


Base Template
http://127.0.0.1:8000/products/naruto-notebooks/
eCommerce Home Contact Products Logout 3
Search
Search
Add to cart

Naruto Notebooks

11 hours, 29 minutes ago
ComicSense Naruto Anime A5 Notebook
Notebook with 650 gsm laminated hard outer cover digital printed and
150 pages (75 leaves) unruled/blank and 90 GSM offset-printed
Comes in A5 (8.25 X 5.8 INCHES) size
Pack of 3
Multicolor
Spiral Bound
10 days replacement policy
Cash on delivery
Free shipping

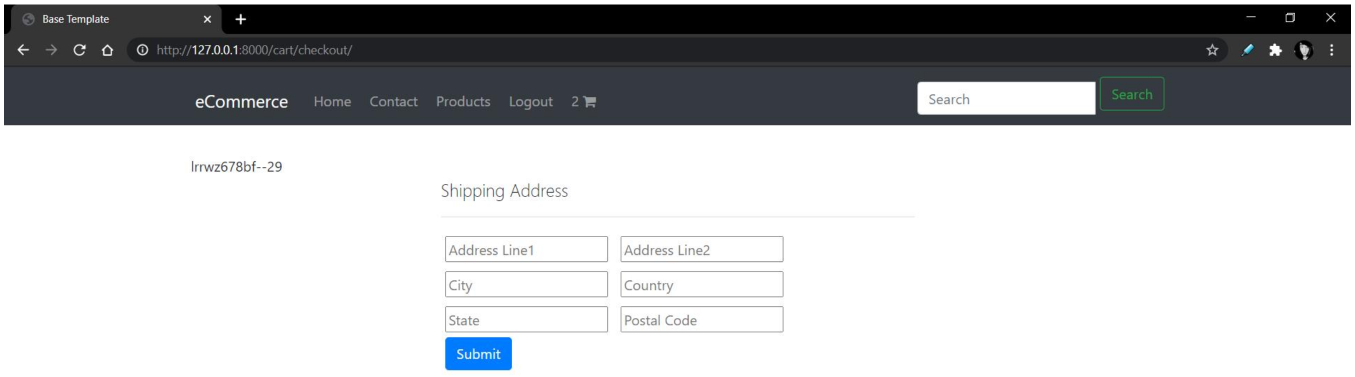
Screenshot 6.7: Single Product Page


Base Template
http://127.0.0.1:8000/cart/
eCommerce Home Contact Products Logout 2
Search
Search
Checkout

Cart

#	Product Name	Price
1	DELL Laptop Remove?	61990.00
2	Fastrack Watch Remove?	1400.00
		Subtotal 63390.00
		Total 68461.20

Screenshot 6.8: Cart Page

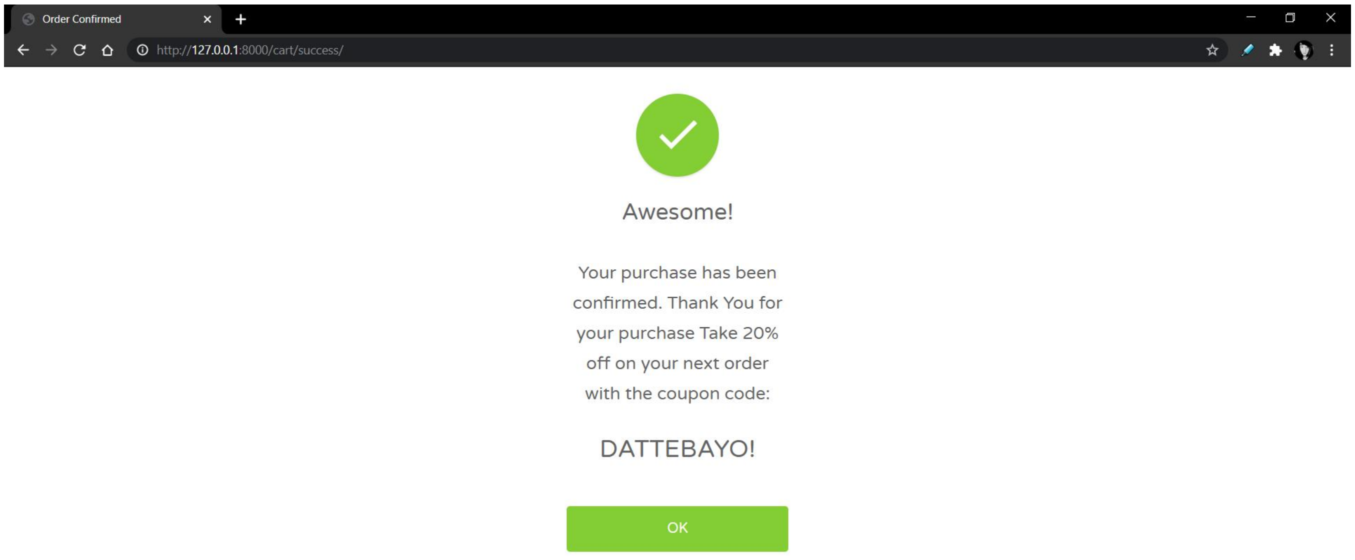


lrrwz678bf--29

Shipping Address

Address Line1	Address Line2
City	Country
State	Postal Code

Screenshot 6.9: Shipping Address Page



Awesome!

Your purchase has been confirmed. Thank You for your purchase Take 20% off on your next order with the coupon code:

DATTEBAYO!

Screenshot 6.10: Success Page

CONCLUSION

Chapter 7

CONCLUSION AND FUTURE SCOPE

7.1 CONCLUSION

E – Commerce Website is a new experience and has greatly impacted the lives of consumers in its short time of existence. It is expected to grow constantly in years to come with advancements in technology. E – Commerce website has made consumers more effective and efficient in their behavior and has driven businesses to a new level, forcing many to make the necessary adjustments and changes to reach the new market of knowledgeable consumers. Customers can browse any product and their details and can purchase the products. User has to get the order through the delivery policy. Admin can keep track of transactions through admin panel.

7.2 FUTURE ENHANCEMENT

The future plan of this project is to improved design; implementation and documentation in such a way that anyone can use this project for better perform. We will develop the site more dynamically and the database work as well. There are some limitations in my project. More security in the system, More user friendly system.

BIBLIOGRAPHY

Textbooks:

- CSS: The Definitive Guide
- HTML and CSS: Design and Build websites

Websites:

- https://docs.djangoproject.com/en/3.1/
- https://getbootstrap.com/
- https://www.w3schools.com/bootstrap4/default.asp
- https://www.w3schools.com/html/default.asp