

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION TO SQL

Today, data is the basis of any business. The world of enterprise computing is the epitome of data-driven businesses. The importance of structured storage of data is unquestionably in the focus. And at this time, with the computing model quickly shifting towards the cloud, and storage costs falling rapidly, enterprises are leveraging data more and more to tune their businesses.

It is, then, imperative that a ground-zero understanding of this structured model of storage and retrieval of data is of paramount importance. A beginner, however, naturally feels lost in the sea of information available out in the wild world. This series, Introduction to SQL, comes to the rescue, by starting at the very base of SQL — the very fabric — and then, launching from there, towards the bigger picture.

The concept of SQL has been around for four decades now. However, it has been evolving and continues to evolve. We all studied SQL at some point or another in our education. We are here to make it better by moving away from the archaic tools and practices. When you learn a certain system using archaic tools and practices, you need to assimilate it and rethink the application of this system in the modern industry when you become part of it. With this book, on the other hand, you would be learning the concepts using industry-standard tools and practices.

Understand that SQL is not going anywhere. If anything, it has only become a hot cake with a further push in the competition by some strategic drivers like Microsoft, who have now made SQL available for Linux as well. The different cloud offerings such as Microsoft Azure and Amazon Web Services — the two largest players in the field — have a very dedicated approach to the relational database model, and SQL, subsequently. And this is just the beginning of a new era.

1.2 INTRODUCTION TO FRONT END SOFTWARE

The front end stack is made up of many different languages and libraries. While these vary from application to application, there are only a few generic languages understood by all web browsers. These three main front-end coding languages are HTML, CSS and JavaScript.

Together, they create the underlying scaffolding that web browsers use to render the web pages that we interact with every day. All other libraries and front-end engineering are built upon these three main languages, which makes them must-have skills for any front-end developer.

HTML

HyperText Markup Language, commonly referred to as HTML, is the standard markup language used to create web pages. Web browsers can read HTML files and render them into visible or audible web pages. HTML describes the structure of a website semantically along with cues for presentation, making it a markup language, rather than a programming language.

CSS

Cascading Style Sheets (CSS) is a style sheet language used for describing the look and formatting of a document written in a markup language. Although most often used to change the style of web pages and user interfaces written in HTML and XHTML, the language can be applied to any kind of XML document, including plain XML, SVG and XUL. Along with HTML and JavaScript, CSS is a cornerstone technology used by most websites to create visually engaging webpages, user interfaces for web applications, and user interfaces for many mobile applications.

JavaScript

JavaScript is a runtime language for web browsers. This means that when you open a web page, the page will load both the foundational JavaScript that is standard with the page and any new JavaScript added to a page. The new JavaScript will load in parallel with it and can perform actions and make decisions.

JavaScript differentiates from back-end languages such as Java or Ruby in that it's visible to the user and is not compiled ahead of time to run secretly behind the curtain. It's the true programming language of front-end engineering and the underlying language that ties everything together.

Django

Django is a web application framework written in Python programming language. It is based on MVT (Model View Template) design pattern. The Django is very demanding due to its rapid development feature. It takes less time to build application after collecting client requirement.

This framework uses a famous tag line: The web framework for perfectionists with deadlines. By using Django, we can build web applications in very less time. Django is designed in such a manner that it handles much of configure things automatically, so we can focus on application development only.

1.3 PROJECT REPORT OUTLINE

The report is arranged in the following way:

Chapter 1: Introduction

Chapter 2: Requirement Specification

Chapter 3: Objective of the project

Chapter 4: Implimentation

Chapter 5: Frontend Design

Chapter 6: Testing

Chapter 7: Result

CHAPTER 2

REQUIREMENT SPECIFICATION

2.1 SOFTWARE REQUIREMENTS

Operating System : Windows 8

Database : MySQL Server.

Tools : Visual Studio, Google Chrome, Command Prompt.

2.2 HARDWARE REQUIREMENTS

Processor : Any Processor above 500 MHz.

RAM : 4.00 GB.

Hard Disk : 64-Bit Operating System.

Compact Disk : 650 MB.

Input device : KeyBoard and Mouse.

Output device : Monitor.

CHAPTER 3

OBJECTIVE OF THE PROJECT

The main objective of the E-Commerce Website is to manage the details of shopping, bills, customer. It manages all the information about products,customers,shopping. The Project is totally built at administrative end and thus only the administrator is guranteed the access. The purpose of the project is to build an application program to reduce the manual work of managing. It tracks all the details about bills and customers.

Features are as follows :

- Provides the searching facilities on various factors.
- Manage the information of bills and customers.
- Shows the information and description of the shopping.
- It generates the bills, Increases Efficiency.
- Deals with monitoring the information.
- Editing, Adding and Updating of records is improved which results in proper resource management of the website.
- Integration of all records of buyers.

CHAPTER 4

IMPLEMENTATION

4.1 ER DIAGRAM

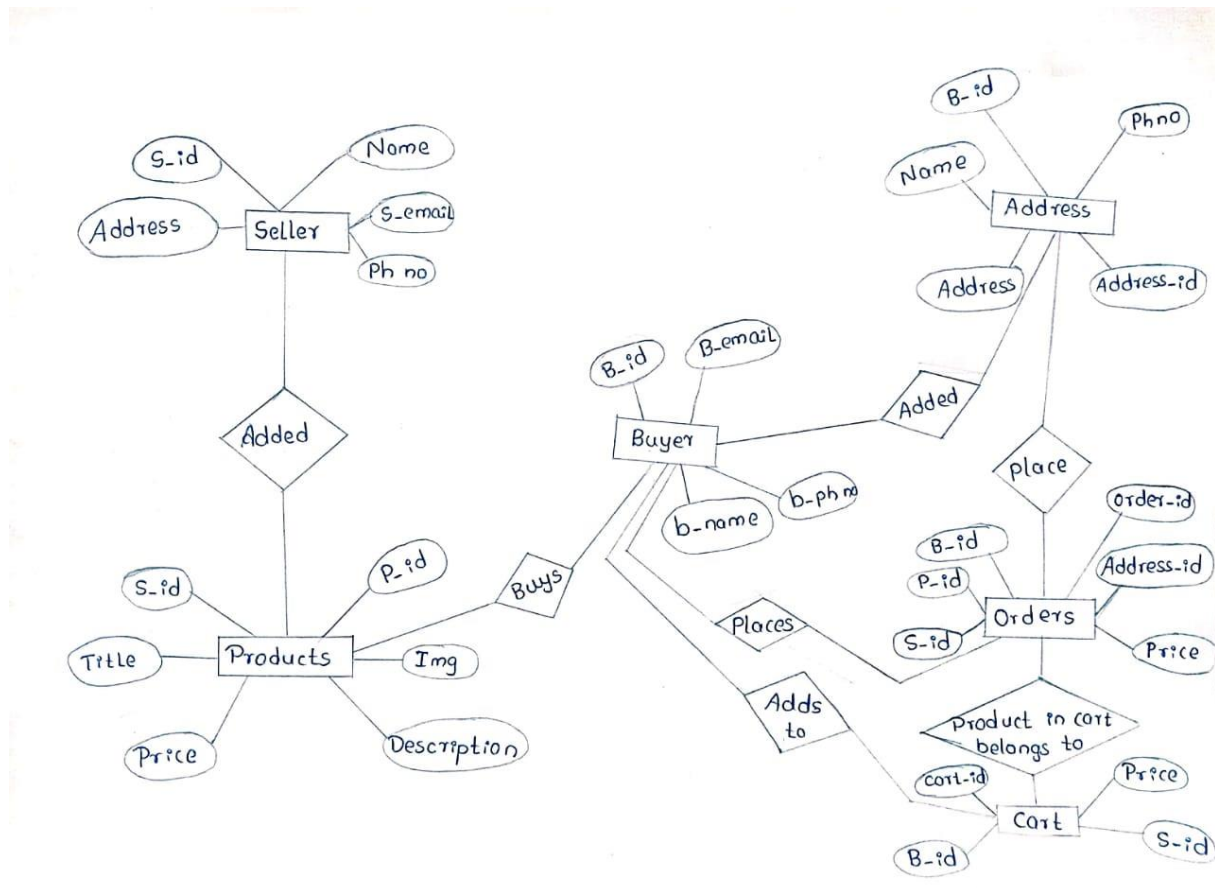


Fig 1: ER Diagram

4.2 MAPPING OF ER DIAGRAM TO SCHEMA DIAGRAM

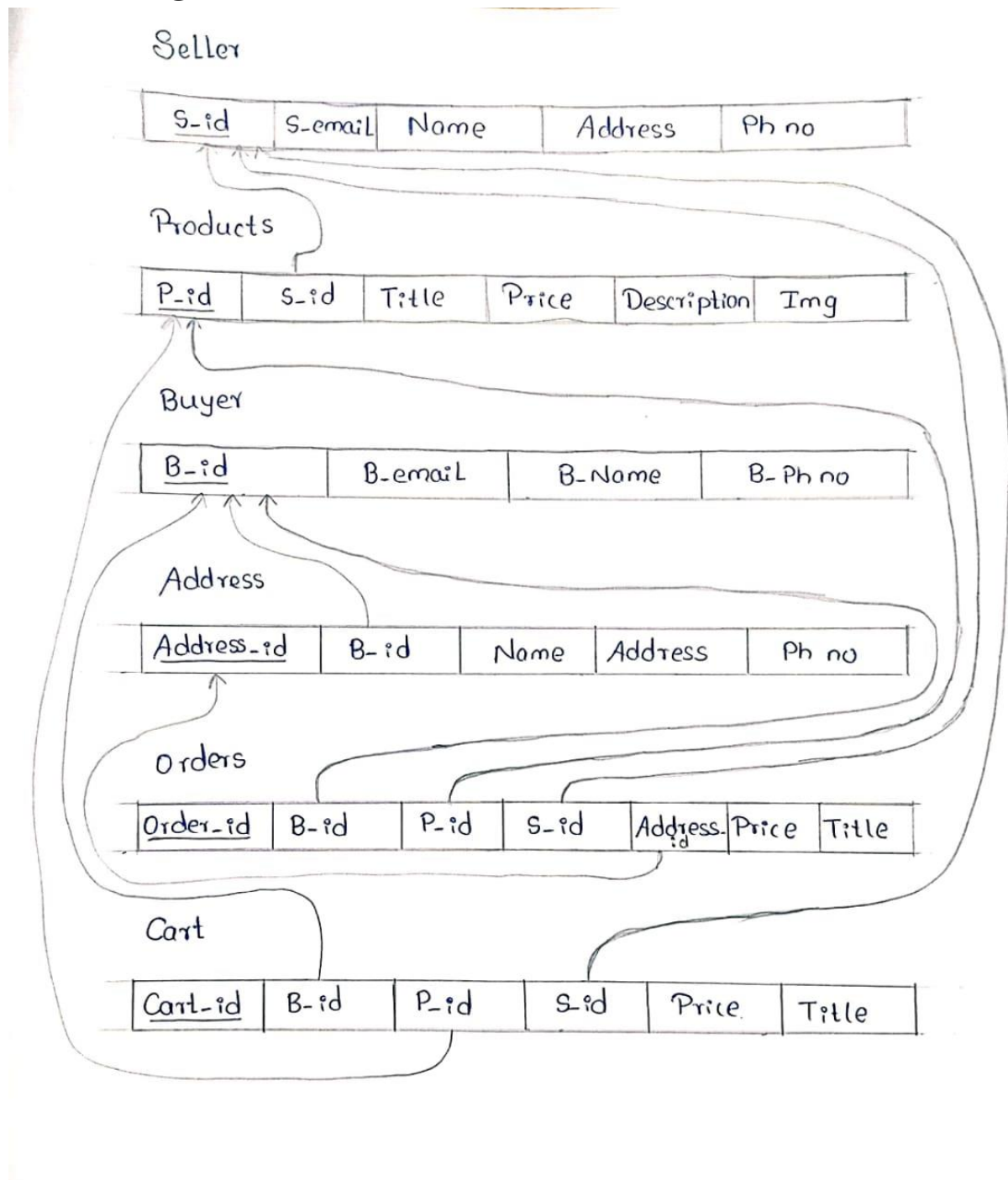


Fig 2: Schema diagram

4.3 MAPPING OF THE ER SCHEMA TO RELATIONS

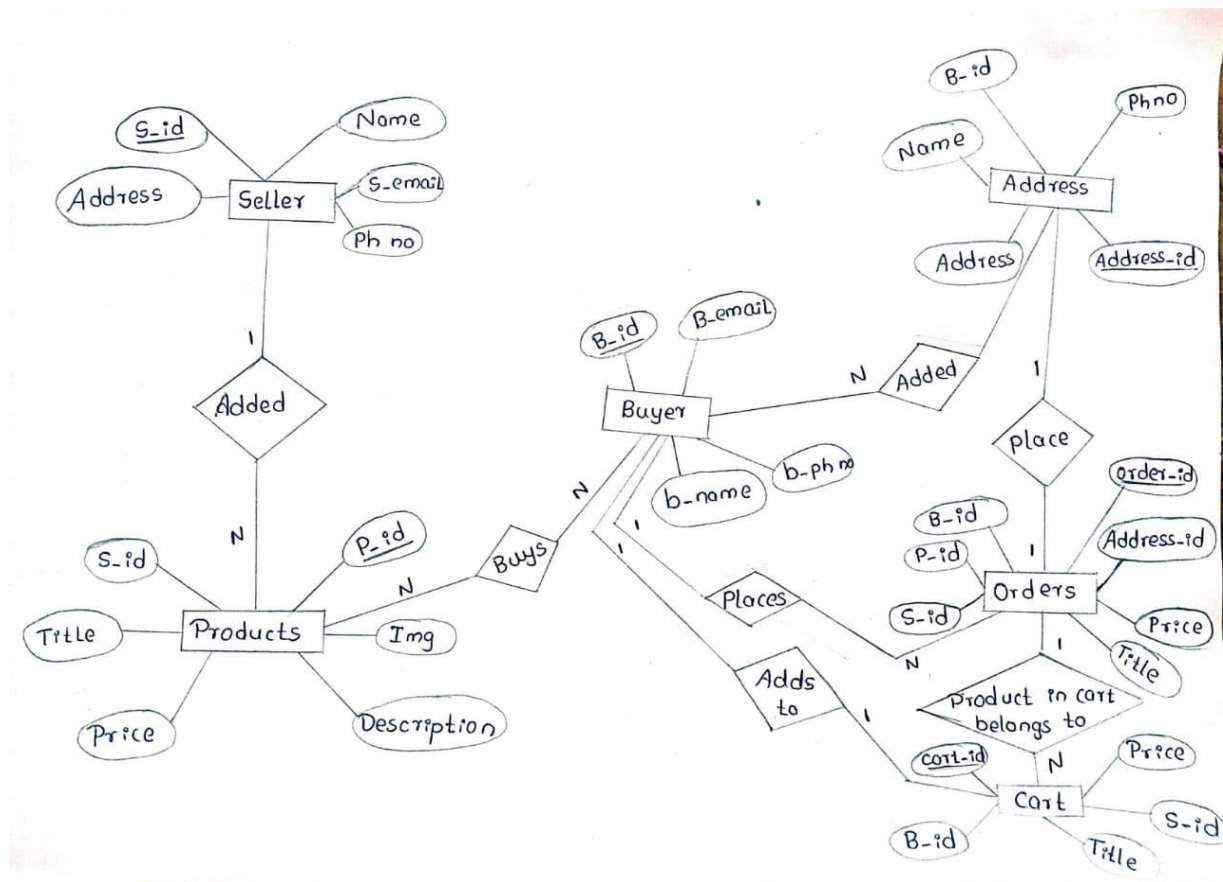


Fig 3: ER Schema to Relations diagram

4.4 CREATION OF TABLES

operations = [

migrations.CreateModel(

name='buyer',

fields=[

('b_email', models.CharField(max_length=100, primary_key=True, serialize=False)),

('b_name', models.CharField(max_length=100)),

('b_phno', models.CharField(max_length=10, null=True)),

],

options={

'db_table': 'buyer',


```

    },

    ),

    migrations.CreateModel(

        name='seller',
        fields=[
            ('id',models.AutoField(auto_created=True,primary_key=True,serialize=False,
verbose_name='ID')),
            ('s_name', models.CharField(max_length=50)),
            ('s_address', models.CharField(max_length=200)),
            ('s_phno', models.CharField(max_length=20)),
            ('email',models.ForeignKey(on_delete=django.db.models.deletion.CASCADE,
to='webapp.buyer')),
        ],
        options={
            'db_table': 'seller',
        },
    ),

    migrations.CreateModel(

        name='product',
        fields=[
            ('id',models.AutoField(auto_created=True,primary_key=True,serialize=False,
verbose_name='ID')),
            ('title', models.CharField(blank=True, max_length=100)),
            ('description', models.TextField(blank=True)),
            ('image', models.ImageField(blank=True, upload_to='product_image')),
            ('price', models.IntegerField(blank=True, default=0)),
            ('sid',models.ForeignKey(on_delete=django.db.models.deletion.CASCADE,
to='webapp.seller')),
        ],

        options={
            'db_table': 'product',

```

```
    },  
    ),  
  
    migrations.CreateModel(  
        name='address',  
  
        fields=[  
            ('id',models.AutoField(auto_created=True,primary_key=True,serialize=False,  
verbose_name='ID')),  
            ('name', models.CharField(max_length=30)),  
            ('phno', models.CharField(max_length=10)),  
            ('address', models.TextField()),  
            ('bid',models.ForeignKey(on_delete=django.db.models.deletion.CASCADE,  
to='webapp.buyer')),  
        ],  
        options={  
            'db_table': 'address',  
        },  
    ),  
    migrations.CreateModel(  
        name='orders',  
        fields=[  
            ('id',models.AutoField(auto_created=True,primary_key=True,serialize=False,  
verbose_name='ID')),  
            ('oid', models.CharField(editable=False, max_length=128)),  
            ('price', models.IntegerField()),  
            ('quantity', models.IntegerField()),  
            ('address',models.ForeignKey(on_delete=django.db.models.deletion.CASCADE,  
to='webapp.address')),  
  
            ('bid',models.ForeignKey(on_delete=django.db.models.deletion.CASCADE,  
to='webapp.buyer')),
```

```

        ('pid',models.ForeignKey(on_delete=django.db.models.deletion.CASCADE,
to='webapp.product')),
        ('sid',models.ForeignKey(on_delete=django.db.models.deletion.CASCADE,
to='webapp.seller')),

    ],
    options={
        'db_table': 'orders',
        'unique_together': {('bid', 'pid', 'sid')},

    },

),
migrations.CreateModel(
    name='cart',
    fields=[
        ('id',models.AutoField(auto_created=True,primary_key=True,serialize=False,
verbose_name='ID')),
        ('quantity', models.IntegerField()),
        ('price', models.IntegerField()),
        ('bid',models.ForeignKey(on_delete=django.db.models.deletion.CASCADE,
to='webapp.buyer')),
        ('pid',models.ForeignKey(on_delete=django.db.models.deletion.CASCADE,
to='webapp.product')),
        ('sid',models.ForeignKey(on_delete=django.db.models.deletion.CASCADE,
to='webapp.seller')),
    ],
    options={
        'db_table': 'cart',
        'unique_together': {('pid', 'bid')},

    },
),
]

```

In SQL

1. Creating table for buyers

Create Table Buyer

```
(  
    B_id int primary key,  
    B_email Varchar(20),  
    B_name Varchar(20),  
    B_phno int  
  
);
```

2. Creating table for sellers

Create Table Seller

```
(  
    S_id int primary key,  
    S_email Varchar(20),  
    S_name Varchar(20),  
    S_address Varchar(20),  
    B_phno int  
  
);
```

3. Creating table for products

Create Table Products

```
(  
    P_id int primary key ,
```

S_id References Seller(S_id) on Delete Cascade,

```
    Title Varchar(20),  
    price int,
```

```
Description Varchar(50),  
Img ImgFeild  
);
```

4. Creating table for Address

```
Create Table Address  
(  
Address_id int primary key ,  
B_id References Buyer(B_id) on Delete Cascade,  
Name Varchar(20),  
  
Address Varchar(50),  
Phno int  
  
);
```

5. Creating table for Cart

```
Create Table Cart  
(  
Cart_id int primary key ,  
B_id References Buyer(B_id) on Delete Cascade,  
P_id References Products(P_id) on Delete Cascade,  
S_id References Seller(S_id) on Delete Cascade,  
Price int,  
Title Varchar(20)  
);
```

6. Creating table for Orders

```
Create Table Orders
```

E-Commerce Website

```
(  
    Order_id int primary key ,  
    B_id References Buyer(B_id) on Delete Cascade,  
    P_id References Products(P_id) on Delete Cascade,  
    S_id References Seller(S_id) on Delete Cascade,  
    Price int,  
  
    Title Varchar(20)  
);
```

4.5 INSERTION OF TUPLES

Create your views here.

start of logout

```
def logout_app(request):  
    logout(request)  
    return HttpResponseRedirect("/")
```

end of logout page

start of home page

```
def home(request):  
    if request.user.is_authenticated:  
        user_email=request.user.email
```

```
query="select * from buyer where b_email='"+user_email+'";"  
user = len(list(buyer.objects.raw(query)))  
cursor=connection.cursor()
```

E-Commerce Website

```
    if(user == 0):

        cursor.execute("insert            into            buyer            values(''            +
user_email+'',''+request.user.username+'','null')")

        cursor = connection.cursor()

        return render(request, 'home.html')

# end of home page


# start of add_to _cart in db
def add_to_cart(request, id):

    if request.user.is_authenticated:

        products = product.objects.get(id=id)

        cursor = connection.cursor()

        user_email = request.user.email

        p_id = str(products.id)

        query = "select * from cart where bid_id='"+user_email+"' and pid_id='"+p_id

        c = len(list(cart.objects.raw(query)))

        if(c == 0):

            bid = buyer.objects.get(b_email=request.user.email)

            added    =    cart(pid=products,    bid=bid,price=products.price,    sid=products.sid,
quantity=1)

            added.save()

            #
            cursor.callproc("add_to_cart",[product_now.id,user_email,product_now.price,product_now.s
_id_id])

        else:

            # print("enters else")

            cursor.execute("update cart set quantity=quantity+1 where pid_id=" + p_id+" and
bid_id='"+user_email+"';")

            cursor.execute("update cart set price="+str(products.price) + "*quantity where
pid_id='"+p_id+" and bid_id='"+user_email+"';")
```

```

        return HttpResponseRedirect('/products')

else:
    # print('entered else')

    return HttpResponseRedirect('/oauth/login/google-
oauth2/?next=/add_to_cart/product/'+id)
# end of add_to_cart

# start of cart page with products from db
def cartpage(request):
    cart_items = []
    if request.user.is_authenticated:
        user_email = request.user.email
        # products = product.objects.get(id=id)
        cursor = connection.cursor()

    # p_id = str(products.id)
    # query = "select * from cart where bid_id='"+user_email+" and pid_id="+p_id
    query = "select * from buyer where b_email='"+user_email+'";"

    user = len(list(buyer.objects.raw(query)))
    cursor = connection.cursor()
    if(user == 0):
        cursor.execute("insert into buyer values('"+
user_email+"','"+request.user.username+"',null)")
        # cursor.execute("select pid_id,quantity, price from cart where
bid_id='"+user_email+'";")

    total = 0

    cursor = cart.objects.filter(bid = request.user.email)

    for row in cursor:

```


E-Commerce Website

```
    print(row)

    # products=product.objects.get(id=row[0])

    l={
        'id':row.pid.id,
        'title':row.pid.title,

# 'quantity': row[1],
        'price':row.pid.price,
    }

    total += row.pid.price
    cart_items.append(l)
    print(total)
    print(len(cart_items))
    context= {
        'cart_items':cart_items,
        'total':total
    }

if(len(cart_items)==0):
    return render(request, 'cart.html')
else:
    return render(request, 'cartpage.html',context)

else:
    if(len(cart_items)==0):
        return render(request, 'cart.html')
    return render(request, 'cart.html')

# end of cart page

# start of remove cart function
```

E-Commerce Website

```
def remove_cart(request, id):
```

```
    user_email = request.user.email
```

```
    cursor = connection.cursor()
```

```
    cursor.execute("delete from cart where bid_id='" + user_email+"'" and pid_id='"+str(id)+"';")
```

```
    response = redirect('/cartpage')
```

```
    return response
```

```
# end of remove cart function
```

```
# start of adding address
```

```
def add_address(request):
```

```
    if request.user.is_authenticated:
```

```
        user_email = request.user.email #address is been created in db
```

```
        print(request.POST)
```

```
        address1 = address(
```

```
            bid=buyer.objects.get(b_email=user_email),
```

```
            phno=request.POST['phone'],
```

```
            name=request.POST['name'],
```

```
            address=request.POST['address'],
```

```
)
```

```
        address1.save()
```

```
        buyer1 = buyer(
```

```
            b_phno=request.POST['phone'],
```

```
            b_name=request.POST['name'],
```

```
)
```

```
        buyer1.save()
```

E-Commerce Website

```
items = cart.objects.raw("select * from cart where bid_id='"+request.user.email+"'")
#orders are created in db

cursor=connection.cursor()

# if(len(list(address.objects.raw("select * from address where
bid_id='"+request.user.email+"'")))):

# details = address.objects.raw("select id from address where
bid_id='"+request.user.email+"'")

details = address1

# cursor.execute("select id from address where bid_id='"+request.user.email+"'")

# for d in details:

# id1=d.id

for i in items:

    order1 = orders(

        bid= i.bid,

        pid = i.pid,

        sid = i.sid,

        price = i.price,

        address = details

    )

    order1.save()

# cursor.execute("insert into orders(bid_id,pid_id,sid_id,price,address_id)
values('"+i.bid_id+"','"+

# str(i.pid_id)+"','"+str(i.sid_id)+"','"+str(i.price)+"','"+str(details.id)+"")

# cursor.execute("insert into address values('"+

user_email+"','"+request.user.username+"',null'")

return redirect('/buyer_tq')

# end of adding address
```

In SQL

1. Insert record for buyers

Insert values into buyer(01, 'xyz@gmail.com', 'Xyz', 1111111111);

Insert values into buyer(02, 'uvw@gmail.com', 'Uvw', 2222222222);

2. Creating table for sellers

Insert values into sellers(01, 'aaa@gmail.com', 'Aaa', 'Sample Address 1',
9999999999);

Insert values into sellers(01, 'bbb@gmail.com', 'bbb', 'Sample Address 2',
8888888888);

3. Creating table for products

Insert values into products(1, 03, 'abc', 100.00, 'Sample Description 1',
'Sample product 1');

Insert values into products(2, 04, 'bcd', 200.00, 'Sample Description 2',
'Sample product 2');

4. Creating table for Address

Insert values into address(1, 02, 'aab', 'Sample address 1', 7676767676);

Insert values into address(2, 04, 'bbc', 'Sample address 2', 8787878787);

5. Creating table for Cart

Insert values into cart(1, 02, 3, 04, 100.0, 'aba');

Insert values into cart(2, 05, 2, 03, 200.0, 'bcb');

6. Creating table for Orders

Insert values into orders(01, 02, 3, 04, 'Sample address 1', 'pqr');

Insert values into borders(02, 05, 2, 03, 'Sample address 2' 'stu');

4.6 CREATION OF TRIGGERS

create trigger my_trigger

E-Commerce Website
AFTER INSERT

ON orders

FOR EACH ROW

CALL empty_cart(NEW.bid_id);

4.7 CREATION OF STORED PROCEDURES

create procedure empty_cart(bid varchar(100))

delete from cart where bid_id=bid;

CHAPTER 5

FRONT END DESIGN

5.1 CONNECTIVITY TO DATABASE

```
DATABASES = {  
    'default': {  
        # 'NAME': os.path.join(BASE_DIR, 'db.sqlite3'),  
        'ENGINE': 'mysql.connector.django',  
        'NAME': 'mydb',  
        'USER': 'root',  
        'PASSWORD': 'amrutha@2020',  
        'HOST': 'localhost', # Or an IP Address that your DB is hosted on  
        'PORT': '3306',  
        'OPTIONS': {  
            'autocommit': True,  
        },  
    }  
}
```

5.2 FRONT END CODE

Systems design is the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. Systems design could see it as the application of systems theory to product development. There is some overlap with the disciplines of systems analysis, systems architecture and systems engineering. If the broader topic of product development "blends the perspective of marketing, design, and manufacturing into a single approach to product development," then design is the act of taking the marketing information and creating the design of the product to be manufactured. Systems

design is therefore the process of defining and developing systems to satisfy specified requirements of the user.

Until the 1990s systems design had a crucial and respected role in the data processing industry. In the 1990s standardization of hardware and software resulted in the ability to build

modular systems. The increasing importance of software running on generic platforms has enhanced the discipline of software engineering.

Object-oriented analysis and design methods are becoming the most widely used methods for computer systems design.[citation needed] The UML has become the standard language in object-oriented analysis and design.[citation needed] It is widely used for modeling software systems and is increasingly used for high designing non-software systems and organizations.[citation needed]

System design is one of the most important phases of software development process. The purpose of the design is to plan the solution of a problem specified by the requirement documentation. In other words the first step in the solution to the problem is the design of the project.

Header.html

```
{% load static %}
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Shoppy</title>
  <link href="https://fonts.googleapis.com/css2?family=Merienda&display=swap" rel="stylesheet">
  <link href="https://fonts.googleapis.com/css2?family=Poppins:wght@400;700&display=swap" rel="stylesheet">
```

```

<link href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css" rel="stylesheet">
<link href="https://cdnjs.cloudflare.com/ajax/libs/animate.css/3.7.2/animate.min.css" rel="stylesheet">
<link rel="stylesheet" href="{ % static 'css/header.css' % }">
</head>
<body>
<div class="nav_1">
<div class="container-fluid">
<nav>
<input type="checkbox" id="nav" class="hidden">
<label for="nav" class="nav-btn">
<i></i>
<i></i>
<i></i>
</label>
<div class="logo">
<a href="/">SHOPPY</a>
</div>
<div class="nav-wrapper">
<ul>

<li><a href="/products">Products</a></li>
<li><a href="/my_orders">My Orders</a></li>
<li><a href="/cartpage">Cart</a></li>
<li>
{ % if user.is_authenticated % }
<a href="/logout">Logout</a>
{ % else % }
<a href="{ % url 'social:begin' 'google-oauth2' % }">Login</a>
{ % endif % }

</li>

```



```

        </ul>
      </div>
    </nav>
  </div>
</div>
</body>
</html>

```

Header.css

```

/* HEADER */
nav {
  padding: 8px;
}

.logo {
  float: left;
  padding: 8px;
  margin-left: auto;
  margin-top: 8px;
}

.logo a {
  color: #233c53;
  text-transform: uppercase;
  font-weight: 700;
  font-size: 30px;
  letter-spacing: 0px;
  text-decoration: none;
  font-family: "Times New Roman", Times, serif;
}

.logo a:hover{
  background: #e6d816 ;
}

```

```

background: -moz-linear-
gradient(45deg, #e6d816 0%, #f14fb3 25%, #f14fb3 50%, #e6d816 75%, #e6d816 100%);
background: -webkit-linear-
gradient(45deg, #e6d816 0%, #f14fb3 25%, #e6d816 50%, #000080 75%, #e6d816 100%);
background: linear-
gradient(45deg, #e6d816 0%, #e6d816 25%, #f14fb3 50%, #f14fb3 75%, #f14fb3 100%);
filter: progid:DXImageTransform.Microsoft.gradient( startColorstr='#f09433', endColorstr
='#bc1888', GradientType=1 );
color: transparent;
-webkit-background-clip: text;
background-clip: text;
-webkit-background-clip: text;
background-clip: text;
cursor: pointer;
}
nav ul {
float: right;
}

nav ul li {
display: inline-block;
float: left;
}

nav ul li:not(:first-child) {
margin-left: 20px;
}

nav ul li:last-child {
margin-right: -10px;
}

```

```

nav ul li a {

display: inline-block;
  outline: none;
  color: #233c53;
  text-transform: uppercase;
  text-decoration: none;
  font-size: 18px;
  letter-spacing: 1.2px;
  font-weight: 600;
  font-family: Arial, Helvetica, sans-serif;
}

nav ul li a: hover{
  background: #e6d816 ;
  background: -moz-linear-
gradient(45deg, #e6d816 0%, #f14fb3 25%, #f14fb3 50%, #e6d816 75%, #e6d816 100%);
  background: -webkit-linear-
gradient(45deg, #e6d816 0%, #f14fb3 25%, #e6d816 50%, #000080 75%, #e6d816 100%);
  background: linear-
gradient(45deg, #e6d816 0%, #e6d816 25%, #f14fb3 50%, #f14fb3 75%, #f14fb3 100%);
  filter: progid:DXImageTransform.Microsoft.gradient( startColorstr='#f09433', endColorstr
='#bc1888', GradientType=1 );
  color: transparent;
  -webkit-background-clip: text;
  background-clip: text;
  -webkit-background-clip: text;
  background-clip: text;
  cursor: pointer;
}

@media screen and (max-width: 864px) {
  .logo {
    padding: 0;
  }
}

```

```
}

.nav-wrapper {

position: fixed;
    top: 0;
    left: 0;
    width: 100%;
    height: 100%;
    z-index: -1;
    background: #fff;
    opacity: 0;
    transition: all 0.2s ease;
}

.nav-wrapper ul {
    position: absolute;
    top: 50%;
    transform: translateY(-50%);
    width: 100%;
}

.nav-wrapper ul li {
    display: block;
    float: none;
    width: 100%;
    text-align: right;
    margin-bottom: 10px;
}

.nav-wrapper ul li:nth-child(1) a {
    transition-delay: 0.2s;
}
```

```
.nav-wrapper ul li:nth-child(2) a {  
    transition-delay: 0.3s;  
}  
  
.nav-wrapper ul li:nth-child(3) a {  
    transition-delay: 0.4s;  
}  
  
.nav-wrapper ul li:nth-child(4) a {  
    transition-delay: 0.5s;  
}  
  
.nav-wrapper ul li:not(:first-child) {  
    margin-left: 0;  
}  
  
.nav-wrapper ul li a {  
    padding: 10px 24px;  
    opacity: 0;  
    color: #233c53;  
    font-size: 14px;  
    font-weight: 600;  
    letter-spacing: 1.2px;  
    transform: translateX(-20px);  
    transition: all 0.2s ease;  
}  
  
.nav-btn {  
    position: fixed;  
    right: 10px;  
    top: 10px;  
    display: block;  
    width: 48px;
```

```
height: 48px;
cursor: pointer;
z-index: 9999;
border-radius: 50%;
}

.nav-btn i {
display: block;
width: 20px;
height: 2px;
background: #000;
border-radius: 2px;
margin-left: -18px;
}

.nav-btn i:nth-child(1) {
margin-top: 16px;
}

.nav-btn i:nth-child(2) {
margin-top: 4px;
opacity: 1;
}

.nav-btn i:nth-child(3) {
margin-top: 4px;
}
}

#nav:checked + .nav-btn {
transform: rotate(45deg);
}
```

```
#nav:checked + .nav-btn i {  
    background: #000;  
    transition: transform 0.2s ease;  
}  
  
#nav:checked + .nav-btn i:nth-child(1) {  
    transform: translateY(6px) rotate(180deg);  
}  
  
#nav:checked + .nav-btn i:nth-child(2) {  
    opacity: 0;  
}  
  
#nav:checked + .nav-btn i:nth-child(3) {  
    transform: translateY(-6px) rotate(90deg);  
}  
  
#nav:checked ~ .nav-wrapper {  
    z-index: 9990;  
    opacity: 1;  
}  
  
#nav:checked ~ .nav-wrapper ul li a {  
    opacity: 1;  
  
    transform: translateX(0);  
}  
  
.hidden {  
    display: none;  
}  
  
/* END HEADER */
```

```

<!DOCTYPE html>

<html>

  <head>

    <meta name="description" content="This is the description">

    <script src="https://code.jquery.com/jquery-3.3.1.min.js"

integrity="sha384-
tsQFqpEReu7ZLhBV2VZlAu7zcOV+rXbYlF2cqB8txI/8aZajjp4Bqd+V6D5IgvKT"
    crossorigin="anonymous"></script>

    <script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js"></script>

    <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"></script>

    <script src="https://use.fontawesome.com/releases/v5.0.8/js/all.js"></script>

    <link
                                                                    rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css" />

    <link
        href="https://maxcdn.bootstrapcdn.com/font-awesome/4.7.0/css/font-
awesome.min.css" rel="stylesheet"

        integrity="sha384-
wvfXpqpZZVQGK6TAh5PVIGfQNHSoD2xbE+QkPxCafIINEvoEH3SIl0sibVcOQVnN"
        crossorigin="anonymous" />

    <link rel="stylesheet" href="/static/css/cartpage.css" />
  </head>

  <body>

    { % include "header.html" % }

    <div class="cartpage-1">

      <h2> Your Cart Items</h2>

      <table>

        <tr>

          <th>Product</th>

```



```

        <th>Price</th>

        <th></th>

    </tr>

    {% for i in cart_items %}

    <tr>

        <th>{{ i.title }}</th>

        <th>{{ i.price }}</th>

        <th class="text-left">

            <a href="/remove_cart/{{ i.id }}">

                <button class="btn btn-sm btn-danger">

                    <i class="fa fa-trash"></i>

                </button>

            </a>

        </th>

    </tr>

    {% endfor %}

    <tr>

    <th>Total</th>

    <th>{{ total }}</th>

    </tr>

</table>

<a href="/address_form"> <button class="cartpage-btn" >Add Address</button> </a>

</div>

</body>

</html>

```

Cartpage.css

```

.cartpage-1 h2{
    font-size: 45px;
    color:#233c53;
    margin-top:3%;

```

E-Commerce Website

```
    text-align: center;
}
table {
    font-family: arial, sans-serif;
    border-collapse: collapse;
    width: 100%;
}
td, th {

border: 1px solid #dddddd;
    text-align: left;
    padding: 8px;
    font-size: 20px;
}
tr:nth-child(even) {
    background-color: #dddddd;
}
```

E-Commerce Website

The screenshot shows the Django administration interface for an e-commerce website. The browser address bar indicates the URL is `127.0.0.1:8000/admin/webapp/cart/33/change/`. The page title is "Django administration". The left sidebar contains navigation links for "AUTHENTICATION AND AUTHORIZATION", "PYTHON SOCIAL AUTH", and "WEBAPP". The "WEBAPP" section is expanded, showing links for "Addresss", "Buyers", "Carts", "Orderss", "Products", and "Sellers". The "Carts" link is highlighted. The main content area is titled "Change cart" and contains a form with the following fields:

- Bid:** A dropdown menu with the selected value "buyer object (amruthaanand@gmail.com)".
- Pid:** A dropdown menu with the selected value "chottu pens".
- Sid:** A dropdown menu with the selected value "seller object (2)".
- Quantity:** A text input field with the value "1".
- Price:** A text input field with the value "50".

At the bottom of the form are four buttons: "Delete", "Save and add another", "Save and continue editing", and "SAVE".

Fig 4: Cart Table

Home.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Shoppo</title>

  <link href="https://fonts.googleapis.com/css2?family=Merienda&display=swap" rel="stylesheet">

  <link href="https://fonts.googleapis.com/css2?family=Poppins:wght@400;700&display=swap" rel="stylesheet">
  <link href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css" rel="stylesheet">
  <link href="https://cdnjs.cloudflare.com/ajax/libs/animate.css/3.7.2/animate.min.css" rel="stylesheet">
```

```

<link rel="stylesheet" href="/static/css/home.css">
</head>
<body>
  { % include "header.html" % }

  <div class="carousel slide" data-ride="carousel" id="carouselExampleIndicators">
    <ol class="carousel-indicators">
      <li class="active" data-slide-to="0" data-target="#carouselExampleIndicators"></li>
      <li data-slide-to="1" data-target="#carouselExampleIndicators"></li>
      <li data-slide-to="2" data-target="#carouselExampleIndicators"></li>
    </ol>
    <div class="carousel-inner">
      <div class="carousel-item active">
        
        <div class="carousel-caption d-none d-md-block">
          <h5 class="animated bounceInRight" style="animation-
delay: 1s">Welcome to shoppy</h5>
          <p class="animated bounceInLeft" style="animation-
delay: 2s">You can always find something you want.</p>
          <p class="animated bounceInRight" style="animation-delay: 3s">Shop now</p>
        </div>
      </div>

      <div class="carousel-item">
        
        <div class="carousel-caption d-none d-md-block">
          <h5 class="animated slideInDown" style="animation-
delay: 1s">Shop Things you like</h5>

```

```

        <p class="animated fadeInUp" style="animation-
delay: 2s">You can always find something you want.</p>
        <p class="animated zoomIn" style="animation-delay: 3s">Order now</p>
    </div>
</div>
<div class="carousel-item">

    
    <div class="carousel-caption d-none d-md-block">
        <h5 class="animated zoomIn" style="animation-
delay: 1s">Shop what you want!!!</h5>
        <p class="animated fadeInLeft" style="animation-
delay: 2s">You can always find something you want.</p>
        <p class="animated zoomIn" style="animation-delay: 3s">Buy now</p>
    </div>
</div>
</div><a class="carousel-control-prev" data-
slide="prev" href="#carouselExampleIndicators" role="button"><span aria-
hidden="true" class="carousel-control-prev-icon"></span> <span class="sr-
only">Previous</span></a> <a class="carousel-control-next" data-
slide="next" href="#carouselExampleIndicators" role="button"><span aria-
hidden="true" class="carousel-control-next-icon"></span> <span class="sr-
only">Next</span></a>
</div>

<script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.3/umd/popper.min.js"></scr
ipt>

```

```
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/js/bootstrap.min.js"></script>
    {% include "footer.html" %}
</body>
</html>
```

Home.css

```
/* CAROUSEL */
.carousel-item {
    height: 89vh;
    min-height: 300px;
}

.carousel-caption {
    bottom: 220px;
}

.carousel-caption h5 {
    font-size: 45px;
    text-transform: uppercase;
    letter-spacing: 2px;
    font-family: merienda;
    float: right;
}

.carousel-caption p {
    width: 60%;
    margin-top: 5px;
    font-size: 18px;

    line-height: 1.9;
    font-family: poppins;
```

```
float: right;
}

.carousel-caption a {
    text-transform: uppercase;
    background-color: #c0cc17; /* For browsers that do not support gradients */
    background-image: linear-gradient(to top right, #e6d816 12%, #f14fb3);
    color: #fff;
    padding: 10px 30px;
    display: inline-block;
    cursor: pointer;
    outline: none;
    text-align: center;
}

.carousel-caption a:hover{
    background: #ffd658;

    filter: progid:DXImageTransform.Microsoft.gradient( startColorstr='#f09433', endColorstr='#bc1888',GradientType=1 );
    color:transparent;
    -webkit-background-clip: text;
    background-clip: text;
    -webkit-background-clip: text;
    background-clip: text;
    cursor: pointer;
}

.navbar-nav a {
    font-family: poppins;
    font-size: 18px;
    text-transform: uppercase;

    font-weight: bold;
}
```

```
.navbar-light .navbar-brand {  
  
    color: #fff;  
    font-size: 25px;  
    text-transform: uppercase;  
    font-weight: bold;  
    letter-spacing: 2px;  
}  
  
.navbar-light .navbar-nav .active>.nav-link, .navbar-light .navbar-nav .nav-  
link.active, .navbar-light .navbar-nav .nav-link.show, .navbar-light .navbar-nav .show>.nav-  
link {  
    color: #fff;  
}  
  
.navbar-light .navbar-nav .nav-link {  
    color: #fff;  
}  
  
.navbar-nav {  
    text-align: center;  
}  
  
.nav-link {  
    padding: .2rem 1rem;  
}  
  
.nav-link.active, .nav-link:focus {  
    color: #fff;  
}  
  
.navbar-toggler {  
    padding: 1px 5px;  
    font-size: 18px;  
    line-height: 0.3;  
    background: #fff;  
}  
  
.navbar-light .navbar-nav .nav-link:focus, .navbar-light .navbar-nav .nav-link:hover {
```



```

    color: #fff;
}

.w-100 {
    height: 100vh;
}

@media only screen and (max-width: 767px) {
    .navbar-nav.ml-auto {
        background: rgba(0, 0, 0, 0.5);
    }
    .navbar-nav a {
        font-size: 14px;
        font-weight: normal;
    }
}

/* CAROUSEL */

```

Receipt.html

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Receipt</title>
    <link href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.0/css/bootstrap.min.css" rel="stylesheet">
    <link href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.0/js/bootstrap.min.js" rel="stylesheet">
    <link href="https://use.fontawesome.com/releases/v5.8.1/css/all.css" rel="stylesheet">
    <link href="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.2.1/jquery.min.js" rel="stylesheet">

```

```

    <link rel="stylesheet" href="/static/css/receipt.css">
</head>
<body>

{% include "header.html" %}
<div class="container">
    <div class="page-header">
        <h1>Shopping Receipt </h1>
    </div>
    <div class="container">
        <div class="row">
            {% for i in items %}
            <div class="col-md-6 col-md-offset-3 body-main">
                <div class="col-md-12">
                    <div class="row">
                        <div class="col-md-
4">  </div>
                        <div class="col-md-8 text-right">
                            <h4 style="color: #233c53;"><strong>{{ i.address.name }}</strong></h4>
                            <p>{{ i.address.address }}</p>
                            <p>{{ i.address.phno }}</p>
                        </div>
                    </div> <br />
                    <div class="row">
                        <div class="col-md-12 text-center">
                            <h2><strong>Shoppy</strong></h2><br>
                            <h5>Here is your bill!</h5>
                        </div>
                    </div> <br />
                </div>
            </div>
        </div>
    </div>
</div>

```

```

<table class="table">
  <thead>
    <tr>
      <th>

        <h5>Product</h5>
      </th>
      <th>

        <h5>Amount</h5>
      </th>
    </tr>
  </thead>
  <tbody>

    <tr>
      <td class="col-md-9">{{ i.pid.title }}</td>
      <td class="col-md-3"><i class="fas fa-rupee-sign" area-
hidden="true"></i> {{ i.pid.price }} </td>
    </tr>

    <tr style="color: #233c53;">
      <td class="text-right">
        <h4><strong>Total:</strong></h4>
      </td>
      <td class="text-left">
        <h4><strong><i class="fas fa-rupee-sign" area-
hidden="true"></i> {{ i.price }} </strong></h4>
      </td>
    </tr>

  </tbody>
</table>

```

```

</div>

    <div>
        <div class="col-md-12">
            <p><b>Date :</b>{{ i.created_at }}</p> <br />
            <p><b>Signature:</b>Shoppyyyyy/-</p>

        </div>
    </div>
</div>
<div>
    {% endfor %}
</div>
</div>

    <!-- <button class="receipt_btn">Confirm</button> -->
</div>
</body>
</html>

```

Receipt.css

```

.body-main {
    background: #ffffff;
    border-bottom: 15px solid #1E1F23;
    border-top: 15px solid #1E1F23;
    margin-top: 3px;
    margin-bottom: 30px;
    padding: 40px 30px !important;
    position: relative;
    box-shadow: 0 1px 21px #808080;
    font-size: 10px
}

.main thead {
    background: #1E1F23;

    color: #fff
}

.img {
    height: 100px
}

h1 {

```

```
        text-align: center
    }
    .receipt_btn{
        width: 25%;
        padding: 8px 10px;
        font-size: 20px;
        border: 0px;

background-color: #c0cc17; /* For browsers that do not support gradients */
        background-image: linear-gradient(to top right, #e6d816 12%, #f14fb3);
        color: #fff;
        cursor: pointer;
        border-radius: 0px;
        outline: none;
        margin-bottom: 2%;
        margin-left: 40%;
    }
    .receipt_btn:hover{
        background: #ffd658;
    }
}
```

E-Commerce Website

The screenshot shows the Django administration interface for an e-commerce website. The browser address bar indicates the URL is `127.0.0.1:8000/admin/webapp/orders/30/change/`. The page title is "Django administration". The sidebar on the left contains navigation links for "AUTHENTICATION AND AUTHORIZATION" (Groups, Users), "PYTHON SOCIAL AUTH" (Associations, Nonces, User social auths), and "WEBAPP" (Addresses, Buyers, Carts, Orders, Products, Sellers). The "Orders" link is highlighted. The main content area is titled "Change orders" and contains a form with the following fields:

- Bid:** A dropdown menu showing "buyer object (amruthaanuanand@gmail.com)".
- Pid:** A dropdown menu showing "set of pens(10)".
- Sid:** A dropdown menu showing "seller object (2)".
- Created at:** Two fields: "Date" (2020-12-30) and "Time" (10:01:27). A note below states "Note: You are 5.5 hours ahead of server time."
- Price:** A text input field containing "50".
- Address:** A dropdown menu showing "address object (37)".

At the bottom of the form, there are four buttons: "Delete", "Save and add another", "Save and continue editing", and "SAVE".

Fig 5: Orders Table

Footer.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Shoppy</title>
  <link href="https://fonts.googleapis.com/css2?family=Merienda&display=swap" rel="stylesheet">
  <link href="https://fonts.googleapis.com/css2?family=Poppins:wght@400;700&display=swap" rel="stylesheet">
  <link href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css" rel="stylesheet">
  <link href="https://cdnjs.cloudflare.com/ajax/libs/animate.css/3.7.2/animate.min.css" rel="stylesheet">
  <link rel="stylesheet" href="/static/css/footer.css">
</head>
```

```

<body>
  <footer class="panel-footer">
    <div class="container">
      <div class="row">
        <section id="hours" class="col-sm-4">
          <span>About Us</span><br>
          Shoppy<br>
          We have quality products. <br>

          <hr class="visible-xs">
        </section>
        <section id="address" class="col-sm-4">
          <span>Contact</span><br>
          Number : 9876543210<br>
          Email : Shoppy@gmail.com
          <hr class="visible-xs">
        </section>
        <section id="testimonials" class="col-sm-4">
          <p>"The thing about shopping is that you never know exactly what you are going to
find. A shopping experience can be filled with joyous surprises or unexpected pitfalls.
          No matter how hard you try, you cannot plan every detail of a shopping trip, and you
          never know exactly how it will end up.!"</p>
        </section>
      </div>

      <div class="text-center">&copy; Copyright Shoppy 2020</div>
    </div>
  </footer>
</body>
</html>

```

Footer.css

```
/* FOOTER */
```

```

.panel-footer {
  margin-top: 0px;
  padding-top: 35px;
  padding-bottom: 30px;
  background-color: #233c53;
  border-top: 0;
}

.panel-footer div.row {
  margin-bottom: 35px;
}

#hours, #address {
  line-height: 2;
  color: gray;
  font-family: sans-serif;
}

#hours > span, #address > span {
  font-size: 1.3em;
}

#hours span{
  font-size:30px;
  font-family: serif;
  color: #fff;
}

#address span{
font-size:30px;
  font-family: serif;
  color:#fff;
}

#address p {
  color: #e6d816;
  font-size: .8em;

  line-height: 1.8;

```



```

}

#testimonials {
    font-style: italic;
}

#testimonials p:nth-child(2) {
    margin-top: 25px;
}

.text-center{
    color: #fff;
    font-family: serif;
    font-size: 20px;
}

#testimonials{
    color: gray;
    font-family: sans-serif;
    font-size: 15px;
    margin-top: 20px;
}

/* END FOOTER */

```

Address_form.html

```

<!DOCTYPE html>
<html lang="en">
<head>

<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Registration Form</title>
<link rel="stylesheet" href="/static/css/address_form.css">
</head>
<body>

    {% include "header.html" %}

    <div class="wrapper">

```

```

<div class="content">
  <h1> ADD YOUR ADDRESS</h1>
</div>
<form method="POST"
  class="needs-validation"
  novalidate
  action="/add_address/"
  enctype="multipart/form-data">
  {% csrf_token %}
  <div class="top-form">
    <div class="inner-form">
      <div class="label">name</div>
      <input
        required
        name="name"
        class="form-control"
        id="name"
        placeholder="name"/>
    </div>
    <div class="inner-form">
      <div class="label">phone</div>
      <input
        required
        name="phone"
        class="form-control"
        id="phone"
        placeholder="phone"/>
    </div>
  </div>
  <div class="middle-form">
    <div class="inner-form">
      <div class="label">Address</div>

```

```

    <input
      required
      name="address"
      class="form-control"
      id="address"
      placeholder="address"/>
  </div>
</div>
<button class="address_form-btn"
  type="submit">
Checkout
</button>
<script>
$(document).on("change", ".custom-file-input", function (event) {
  $(this)
    .next(".custom-file-label")
    .html(event.target.files[0].name);
});
// Example starter JavaScript for disabling form submissions if there are invalid fields
(function () {
  "use strict";
  window.addEventListener(
    "load",
    function () {

      // Fetch all the forms we want to apply custom Bootstrap validation styles to
      var forms = document.getElementsByClassName(
        "needs-validation"
      );

      // Loop over them and prevent submission
      var validation = Array.prototype.filter.call(forms, function (
        form
      ) {

```

```

        form.addEventListener(
            "submit",
            function (event) {
                if (form.checkValidity() === false) {
                    event.preventDefault();
                    event.stopPropagation();
                    $(form).find(".form-control:invalid").focus();
                }
                form.classList.add("was-validated");
            },
            false
        );
    });
},
false
);
})0;
</script>
</div>
</div>
</form>
</body>
</html>

```

Address_form.css

```

@import url("https://fonts.googleapis.com/css?family=Kavivanar");

* {
    margin: 0;
    padding: 0;
    user-select: none;
}

```

```
body {  
  background: rgb(177, 175, 175);  
}
```

```
.wrapper {  
  margin: 0px auto 0;  
  width: 100%;  
  max-width: 680px;  
  padding: 20px;  
  box-sizing: border-box;  
}
```

```
/* content */
```

```
.content {  
  text-align: center;  
}
```

```
.content h1 {  
  letter-spacing: 1px;  
}
```

```
/* form */
```

```
.form {  
  
width: 100%;  
  margin: 25px 0;  
  
}
```

```
.top-form,  
.middle-form,  
.bottom-form {
```

```
width: 100%;
min-height: 65px;
margin: 10px 0;
}

.form input[type="text"],
.form textarea {
border: 2px solid #fff;
padding: 10px 5px;
outline: none;
border-radius: 2px;
width: 100%;
transition: all 0.2s ease;
}

.form input:focus,
.form textarea:focus {
border-color: #4ca1af;
outline: none;
box-shadow: inset 0 1px 1px rgba(0, 0, 0, 0.0125),
0 0 8px rgba(76, 161, 175, 0.5);
}

.form .label {
margin-bottom: 5px;

text-transform: capitalize;
}

/* top-contact */
.top-form .inner-form {
width: 29.9%;
float: left;
```

```
margin-right: 5%;
}

.top-form .inner-form:last-child {
margin-right: 0;
}

/* middle-form */
.middle-form {
clear: both;
}

/* bottom-form */
.bottom-form textarea {
height: 80px;
}

.btn{
width: 30%;
padding: 8px 10px;
font-size: 20px;
border: 0px;
background-color: #c0cc17; /* For browsers that do not support gradients */
background-image: linear-gradient(to top right, #e6d816 12%, #f14fb3);
color: #fff;

cursor: pointer;
border-radius: 0px;
outline: none;
margin-bottom: 2%;
}

.btn:hover{
```

```
background: #ffd658;
}

::-webkit-input-placeholder {
  /* Chrome/Opera/Safari */
  font-family: "Kavivanar", cursive;
}

::-moz-placeholder {
  /* Firefox 19+ */
  font-family: "Kavivanar", cursive;
}

:-ms-input-placeholder {
  /* IE 10+ */
  font-family: "Kavivanar", cursive;
}

}

@media screen and (max-width: 460px) {
  .top-form .inner-form {
    width: 100%;
    margin: 5px 0;
  }

  .top-form,
  .middle-form,
  .bottom-form {
    margin: 5px 0;
  }

  .form {

    margin-top: 10px;
  }
}

.address_form-btn{
  width: 15%;
  padding: 8px 10px;
```


E-Commerce Website

```
font-size: 20px;
border: 0px;
background-color: #c0cc17; /* For browsers that do not support gradients */
background-image: linear-gradient(to top right, #e6d816 12%, #f14fb3);
color: #fff;
cursor: pointer;
border-radius: 0px;
outline: none;
margin-bottom: 2%;
margin-left: 40%;
margin-top: 5%;
}

.address_form-btn:hover{
background: #ffd658;
}
```

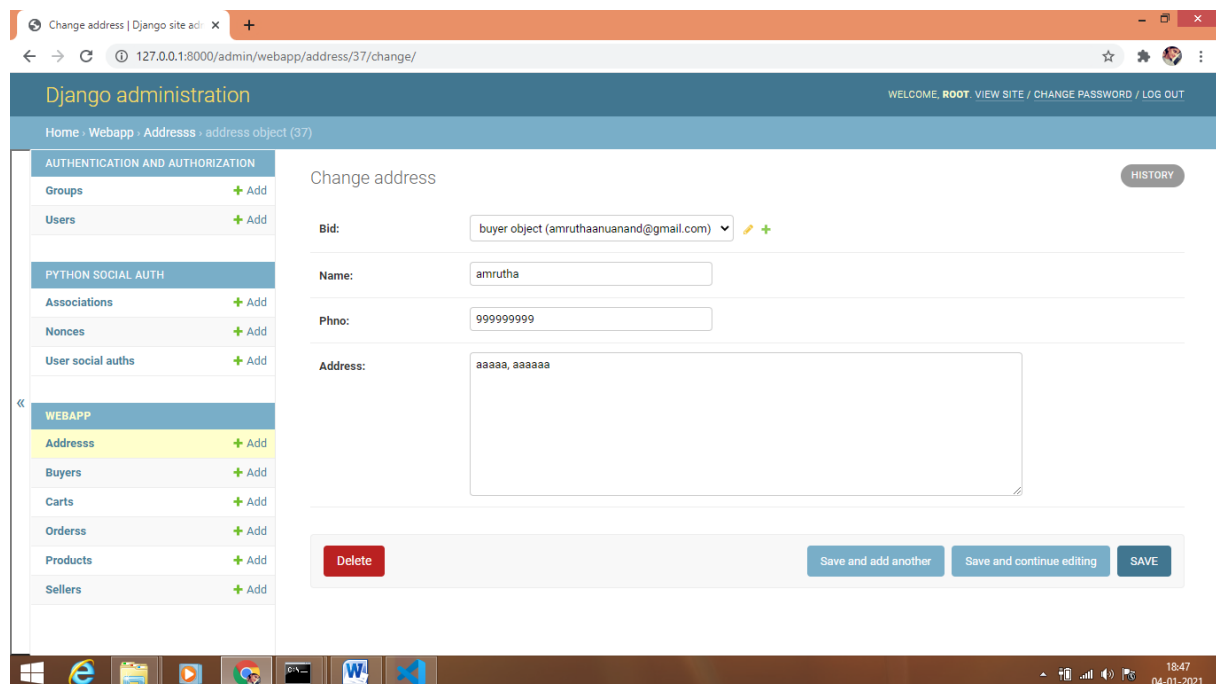


Fig 6: Address Table

Buyer_tq.html

```
<html lang="en">
<head>
<meta charset="utf-8" />
```

```

<meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1" />
<meta name="viewport" content="width=device-width, initial-scale=1">
<title></title>
<link href='https://fonts.googleapis.com/css?family=Lato:300,400|Montserrat:700' rel='style
sheet' type='text/css'>
  <link rel="stylesheet" href="https://2-22-4-dot-lead-
pages.appspot.com/static/lp918/min/default_thank_you.css">
  <link rel="stylesheet" href="/static/css/buyer_tq.css">
  <script src="https://2-22-4-dot-lead-pages.appspot.com/static/lp918/min/jquery-
1.9.1.min.js"></script>
  <script src="https://2-22-4-dot-lead-
pages.appspot.com/static/lp918/min/html5shiv.js"></script>
</head>
<body>
  <div class="jumbotron text-center">
    <h1 class="display-3">Thank You!</h1>

    <p class="lead"><strong>Your order is Placed</strong> for further updates and trackage che
ck your whatsapp </p>
    <p class="lead">
      <a class="btn btn-primary btn-
sm" href="/products" role="button" style="color: blue;">Continue to Shopping with Shoppy<
/a>
    </p>
  </div>
</body>
</html>

```

Buyer_tq.css

```

.display-3{
  margin-top:10%;
}

```

Cart.html

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Shoppy</title>
  <link href="https://fonts.googleapis.com/css2?family=Merienda&display=swap" rel="stylesheet">
  <link href="https://fonts.googleapis.com/css2?family=Poppins:wght@400;700&display=swap" rel="stylesheet">
  <link href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css" rel="stylesheet">
  <link href="https://cdnjs.cloudflare.com/ajax/libs/animate.css/3.7.2/animate.min.css" rel="stylesheet">

  <link rel="stylesheet" href="/static/css/cart.css">
</head>
<body>
  { % include "header.html" % }
  <div class="cart">
    <div class="cart_1">
      <div class="cart_2">
        
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
```



```

<title></title>

{% load static %}

<meta name="viewport" content="width=device-width, initial-scale=1">

<link rel="stylesheet" href="/static/css/product.css">

<!-- font awesome -->

<script src="https://kit.fontawesome.com/dbed6b6114.js" crossorigin="anonymous"></s
cript>

</head>

<body>

    {% include "header.html" %}

    <br>

<div class = "products">

    <div class = "container">

        <div class = "product-items">

            {% for p in products %}

                <!-- single product -->

                <div class = "product">

                    <div class = "product-content">

                        <div class = "product-img">

                            <img src = "{{ p.image.url }}" alt = "product image">

                        </div>

                        <div class = "product-btns">

                            <button type = "button" class = "btn-
cart"> <a href="/add_to_cart{{ p.getabsoluteurl }}">add to cart</a>

                                <span><i class = "fas fa-plus"></i></span>

                            </button>

                        </div>

                    </div>

                </div>

                <div class = "product-info">

                    <p>{{ p.title }}</p>

```

```

        <p>{{p.description}}</p>
        <p>Rs.{{p.price}}</p>
    </div>

</div>
<!-- end of single product -->
{% endfor %}
</div>

</div>

</div>

</div>

</div>
</body>
</html>

```

Product.css

```
@import url('https://fonts.googleapis.com/css2?family=Quicksand:wght@300;400;500;600;700&family=Roboto:wght@300;400;500;700;900&display=swap');

:root{
  --white-light: rgba(255, 255, 255, 0.5);
  --alice-blue: #f8f9fa;
  --carribean-green: #40c9a2;
  --gray: #ededed;
}

*{
  padding: 0;
  margin: 0;
  box-sizing: border-box;
```

```
}  
  
body{  
    font-family: 'Quicksand', sans-serif;  
}  
  
/* Utility stylings */  
img{  
    width: 100%;  
    display: block;  
  
}  
  
.container{  
    width: 88vw;  
    margin: 0 auto;  
}  
  
.lg-title,  
.md-title,  
.sm-title{  
    font-family: 'Roboto', sans-serif;  
    padding: 0.6rem 0;  
    text-transform: capitalize;  
}  
  
.lg-title{  
    font-size: 2.5rem;  
    font-weight: 500;  
    text-align: center;  
    padding: 1.3rem 0;  
    opacity: 0.9;  
}  
  
.md-title{  
    font-size: 2rem;  
    font-family: 'Roboto', sans-serif;  
}  
  
.sm-title{
```

```
font-weight: 300;
font-size: 1rem;
text-transform: uppercase;
}
.text-light{
font-size: 1rem;
font-weight: 600;
line-height: 1.5;

opacity: 0.5;
margin: 0.4rem 0;
}

/* product section */
.products{
background: var(--alice-blue);
padding: 3.2rem 0;
}
.products .text-light{
text-align: center;
width: 70%;
margin: 0.9rem auto;
}
.product{
margin: 2rem;
position: relative;
}
.product_h1{
color: black;
font-size: 40px;
margin-left: 40%;
margin-bottom: 3%;
}
.product-content{
```

```
background: var(--gray);
padding: 3rem 0.5rem 2rem 0.5rem;
cursor: pointer;
}

.product-img{
background: var(--white-light);
box-shadow: 0 0 20px 10px var(--white-light);

width: 200px;
height: 200px;
margin: 0 auto;
border-radius: 50%;
transition: background 0.5s ease;
}

.product-btns{
display: flex;
justify-content: center;
margin-top: 1.4rem;
opacity: 0;
transition: opacity 0.6s ease;
}

.btn-cart, .btn-buy{
background: transparent;
border: 1px solid black;
padding: 0.8rem 0;
width: 125px;
font-family: inherit;
text-transform: uppercase;
cursor: pointer;
border: none;
transition: all 0.6s ease;
}

.btn-cart{
background: black;
```

```
    color: white;
}
.btn-cart:hover{
    background: var(--carribean-green);
}
.btn-buy{
    background: white;
}
.btn-buy:hover{
    background: var(--carribean-green);
    color: #fff;
}
.product-info{
    background: white;
    padding: 2rem;
}
.product-info-top{
    display: flex;
    justify-content: space-between;
    align-items: center;
}
.rating span{
    color: var(--carribean-green);
}
.product-name{
    color: black;
    display: block;
    text-decoration: none;
    font-size: 1rem;
    text-transform: uppercase;
    font-weight: bold;
}
.product-price{
```

```
padding-top: 0.6rem;
padding-right: 0.6rem;
display: inline-block;
}

.product-price:first-of-type{

    text-decoration: line-through;
    color: var(--carribean-green);
}

.product-img img{
    transition: transform 0.6s ease;
}

.product:hover .product-img img{
    transform: scale(1.1);
}

.product:hover .product-img{
    background: var(--carribean-green);
}

.product:hover .product-btns{
    opacity: 1;
}

.off-info .sm-title{
    background: var(--carribean-green);
    color: white;
    display: inline-block;
    padding: 0.5rem;
    position: absolute;
    top: 0;
    left: 0;
    writing-mode: vertical-lr;
    transform: rotate(180deg);
    z-index: 1;
    letter-spacing: 3px;
    cursor: pointer;
```

```
}

/* product collection */
.product-collection{

    padding: 3.2rem 0;
}

.product-collection-wrapper{
    padding: 3.2rem 0;
}

.product-col-left{
    background: linear-gradient(rgba(0, 0, 0, 0), rgba(0, 0, 0, 0.3)), url("images/fashion-img-1.jpg") center/cover no-repeat;
}

.product-col-r-top{
    background: linear-gradient(rgba(0, 0, 0, 0), rgba(0, 0, 0, 0.3)), url("images/fashion-img-2.png") center/cover no-repeat;
}

.flex{
    display: flex;
    justify-content: center;
    align-items: flex-end;
    height: 50vh;
    padding: 2rem 1.5rem 3.2rem;
    margin: 5px;
}

.product-col-r-bottom > div:first-child{
    background: #eaa001;
}

.product-col-r-bottom > div:last-child{
    background: #0090ff;
}

.product-col-content{
    text-align: center;
}
```



```
    color: white;
}

.product-collection .text-light{

    opacity: 1;
    font-size: 0.8;
    font-weight: 400;
    line-height: 1.7;
}

.btn-dark{
    background: black;
    color: white;
    outline: 0;
    border-radius: 25px;
    padding: 0.7rem 1rem;
    border: 0;
    margin-top: 1rem;
    cursor: pointer;
    transition: all 0.6s ease;
    font-size: 1rem;
    font-family: inherit;
}

.btn-dark:hover{
    background: var(--carribean-green);
}

/* Media Queries */

@media screen and (min-width: 992px){
    .product-items{
        display: grid;
        grid-template-columns: repeat(2, 1fr);
    }

    .product-col-r-bottom{
        display: grid;
    }
}
```

```
        grid-template-columns: repeat(2, 1fr);
    }
}

@media screen and (min-width: 1200px){
    .product-items{
        grid-template-columns: repeat(3, 1fr);
    }
    .product{
        margin-right: 1rem;
        margin-left: 1rem;
    }
    .products .text-light{
        width: 50%;
    }
}

@media screen and (min-width: 1336px){
    .product-items{
        grid-template-columns: repeat(4, 1fr);
    }
}
```

```
.product-collection-wrapper{
    display: grid;
    grid-template-columns: repeat(2, 1fr);
}
.flex{
    height: 60vh;
}
.product-col-left{
    height: 121.5vh;
}
}
```

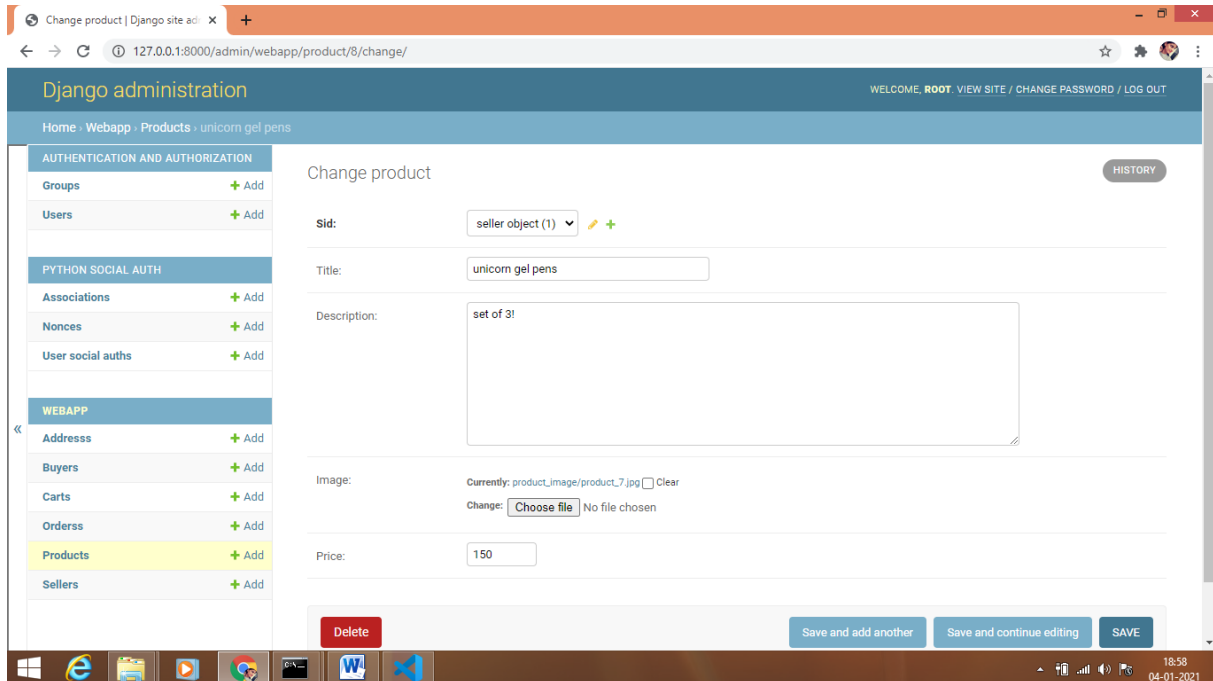


Fig 7: Product Table

Orders.html

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Shoppy</title>
  <link href="https://fonts.googleapis.com/css2?family=Merienda&display=swap" rel="stylesheet">
  <link href="https://fonts.googleapis.com/css2?family=Poppins:wght@400;700&display=swap" rel="stylesheet">
  <link href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css" rel="stylesheet">

```

```

<link href="https://cdnjs.cloudflare.com/ajax/libs/animate.css/3.7.2/animate.min.css" rel="stylesheet">

<link rel="stylesheet" href="/static/css/my_order.css">
</head>

<body>
  {% include "header.html" %}

  <div class="my_order">
    <div class="my_order-1">
      <div class="my_order-2">
        
      </div>
      <div class="my_order-3">
        <h2 class="my_order_h2">You are not yet order</h2>
        <center><p>Lets start shopping! By logging in!</p></center>
        <!--
        <a href="/product.html"><button class="my_order_btn">Shop Now</button></a> -->
      </div>
    </div>
  </div>
  </div>

  <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"></script>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.3/umd/popper.min.js"></script>

  <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/js/bootstrap.min.js"></script>

  {% include "footer.html" %}
</body>
</html>

```

My_order.css

```
.my_order{
  background-color: rgb(133, 131, 131);
  margin-left:20%;
  margin-right: 20%;

  height: 550px;
  align-items: center;
  margin-top: 2%;
  margin-bottom:2%;
  border-radius: 2%;
  padding-top:2%;
}

.my_order-1{
  position: relative;
  width: 50%;
  height: 475px;
  background-color: white;
  box-shadow: -15px 0 35px rgba(8, 6, 6, 0.1);
  transition: .10s;
  margin-left:26%;
  border-radius: 5%;
}

.my_order_btn{
  width: 30%;
  padding: 10px 10px;
  font-size: 20px;
  margin-left: 35%;
  border: 0px;
  background-color: #c0cc17; /* For browsers that do not support gradients */
  background-image: linear-gradient(to top right, #e6d816 12%, #f14fb3);
  color: #fff;
  cursor: pointer;
  border-radius: 20px;
  outline: none;
```

```
text-align: center;
}
.my_order_btn:hover{

background: #ffd658;
}
.my_order_h2{
text-align: center;
}
.my_order_h1{
color: black;
font-size: 40px;
margin-left: 45%;
margin-bottom: 1%;
}
.my_order-3 p{
font-size: 20px;
color:rgb(14, 12, 12);
margin-left: 6%;
}
.my_order-2 img{
width: 70%;
height: 2%;
margin-left:15%;
}
```

E-Commerce Website

The screenshot shows the Django administration interface for an e-commerce website. The browser tab is titled "Change seller | Django site admin". The URL is "127.0.0.1:8000/admin/webapp/seller/2/change/". The page header includes "Django administration" and a welcome message for "root". The breadcrumb trail is "Home > Webapp > Sellers > seller object (2)".

On the left sidebar, there are two main sections: "AUTHENTICATION AND AUTHORIZATION" and "WEBAPP". The "WEBAPP" section is expanded, showing a list of models: Addresss, Buyers, Carts, Orderss, Products, and Sellers. The "Sellers" model is highlighted in yellow.

The main content area is titled "Change seller" and contains a form with the following fields:

- S email: A dropdown menu showing "buyer object (bbb@gmail.com)".
- S name: A text input field containing "bbb".
- S address: A text input field containing "sample address 2".
- S phno: A text input field containing "8888888888".

At the bottom of the form, there are four buttons: "Delete" (red), "Save and add another", "Save and continue editing", and "SAVE". A "HISTORY" button is located in the top right corner of the form area.

Fig 8: Seller Table

The screenshot shows the Django administration interface for an e-commerce website. The browser tab is titled "Change buyer | Django site admin". The URL is "127.0.0.1:8000/admin/webapp/buyer/bbb_40gmail.com/change/". The page header includes "Django administration" and a welcome message for "root". The breadcrumb trail is "Home > Webapp > Buyers > buyer object (bbb@gmail.com)".

On the left sidebar, there are two main sections: "AUTHENTICATION AND AUTHORIZATION" and "WEBAPP". The "WEBAPP" section is expanded, showing a list of models: Addresss, Buyers, Carts, Orderss, Products, and Sellers. The "Buyers" model is highlighted in yellow.

The main content area is titled "Change buyer" and contains a form with the following fields:

- B email: A text input field containing "bbb@gmail.com".
- B name: A text input field containing "bbb".
- B phno: A text input field containing "8888888888".

At the bottom of the form, there are four buttons: "Delete" (red), "Save and add another", "Save and continue editing", and "SAVE". A "HISTORY" button is located in the top right corner of the form area.

Fig 9: Buyer Table

TESTING

This chapter gives the outline of all testing methods that are carried out to get a bug free system. Quality can be achieved by testing the product using different techniques at different phases of the project development. The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components sub assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

6.1 TESTING PROCESS

Testing is an integral part of software development. Testing process certifies whether the product that is developed compiles with the standards that it was designed to. Testing process involves building of test cases against which the product has to be tested.

6.2 TESTING OBJECTIVES

The main objectives of testing process are as follows.

- Testing is a process of executing a program with the intent of finding an error.
- A good test case is one that has high probability of finding undiscovered error.
- A successful test is one that uncovers the undiscovered error.

6.3 TEST CASES

The test cases provided here test the most important features of the project.

6.3.1 Test cases for the project

Table 6.1 ----- Test Case for E-Commerce Website

Sl No	Test Input	Expected Results	Observed Results	Remarks
1	Insert a record	New tuple should be inserted	Query OK 1 row inserted or affected	PASS
2	Delete a record	Delete the record	Query OK 1 row deleted or affected	PASS
3	Update a record	Update the record	Query OK 1 row updated or affected	PASS
4	Create a trigger	Trigger created	Query OK trigger created	PASS
5	Create a stored procedure	Stored procedure created	Query OK stored procedure created	PASS

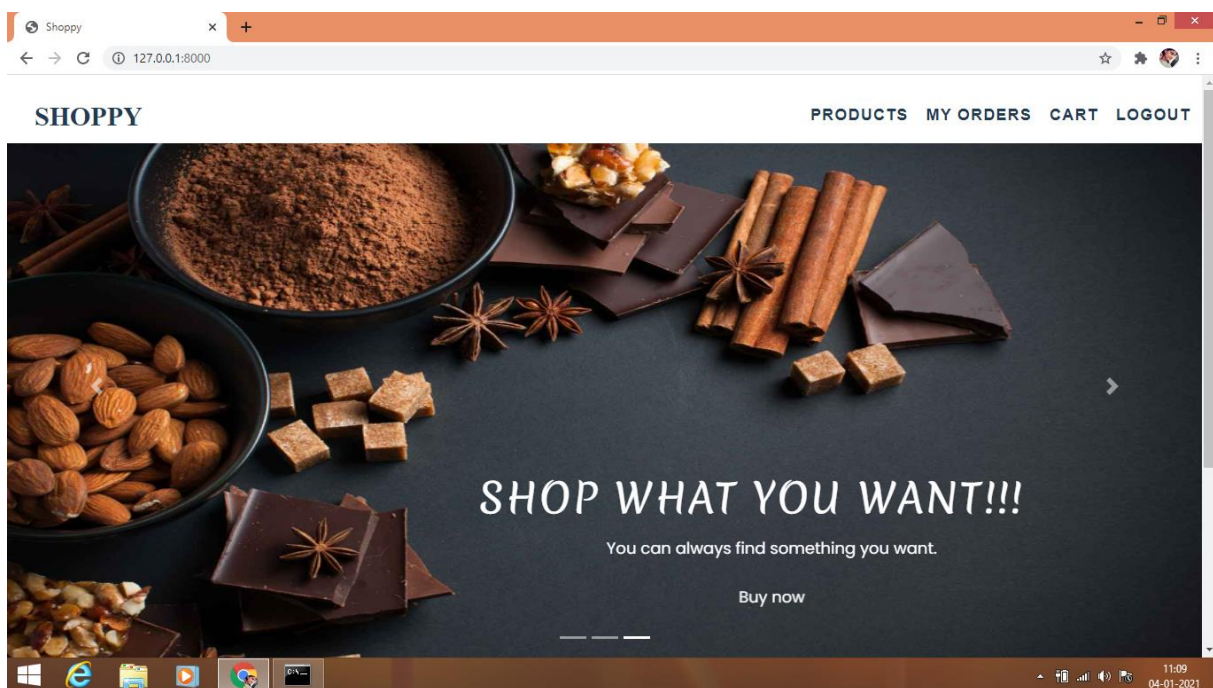
CHAPTER 7

RESULTS

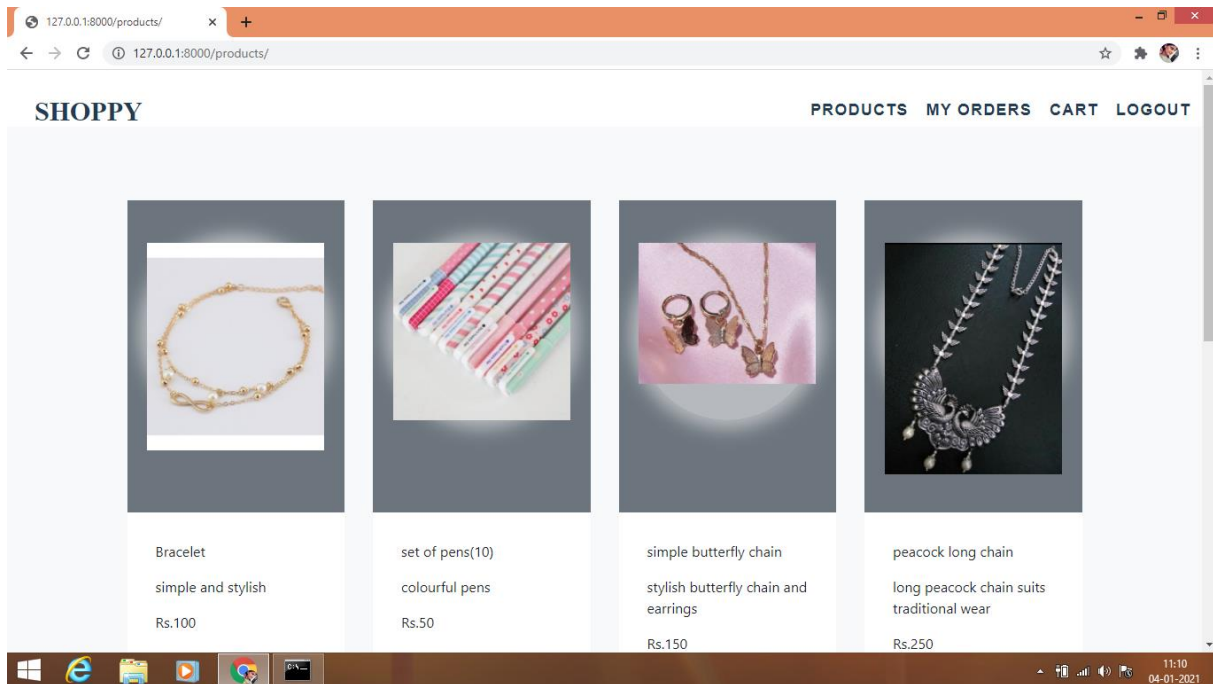
This section describes the screens of the “E-Commerce Website”. The snapshots are shown below for each module.

7.1 SNAPSHOTS

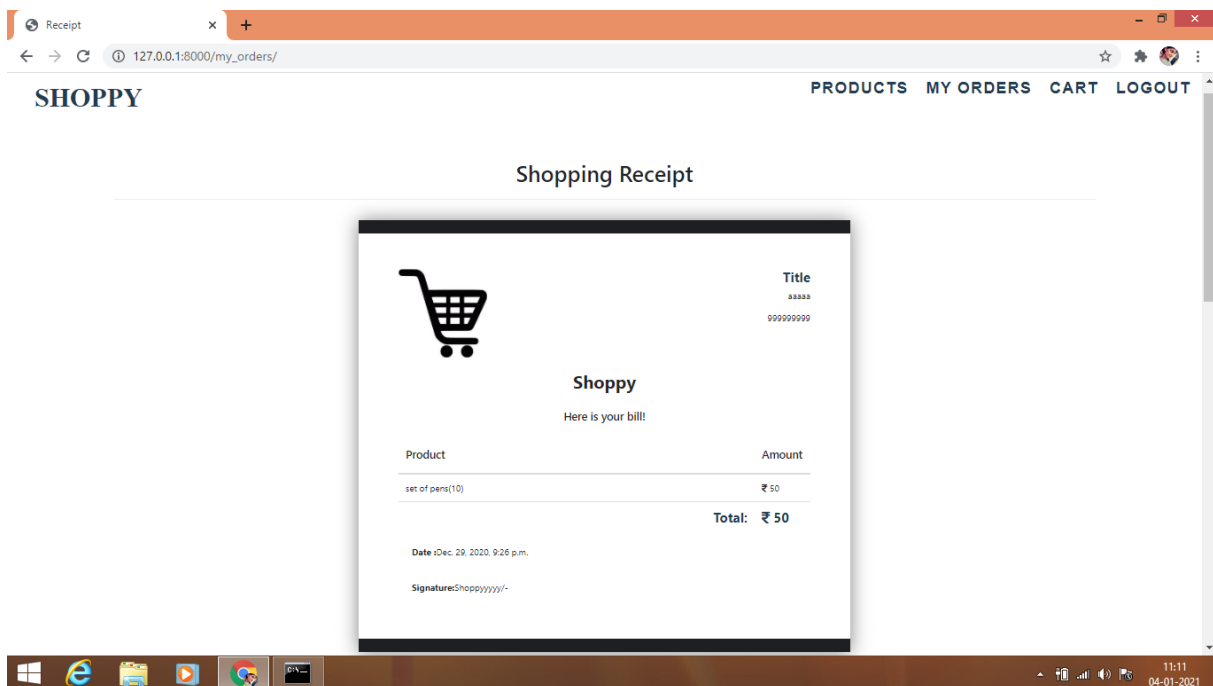
Snapshot 1 : Front Layout



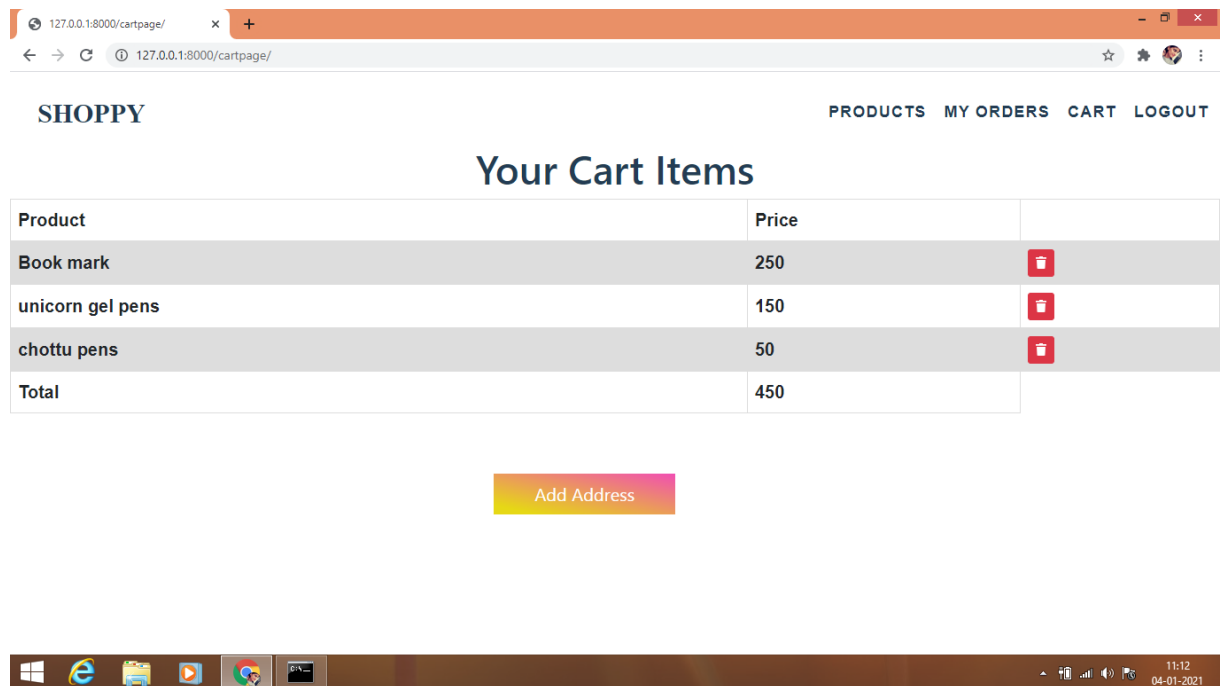
Snapshot 2 : Product-page Layout



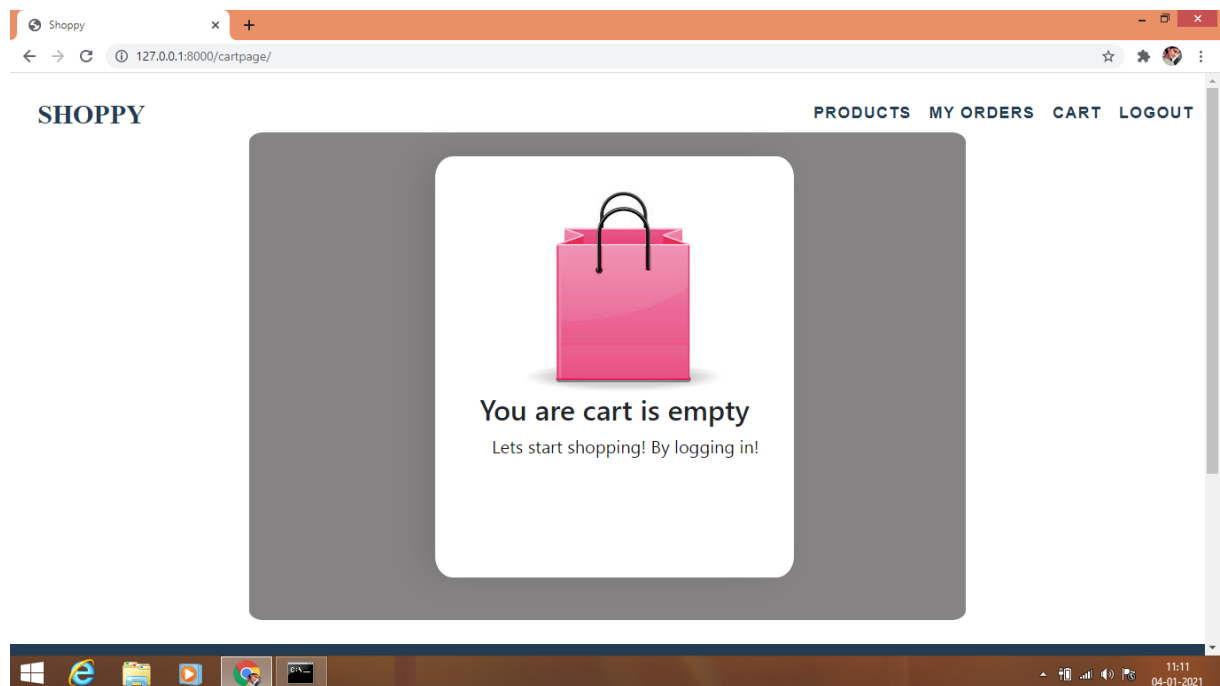
Snapshot 3 : My-Orders page Layout



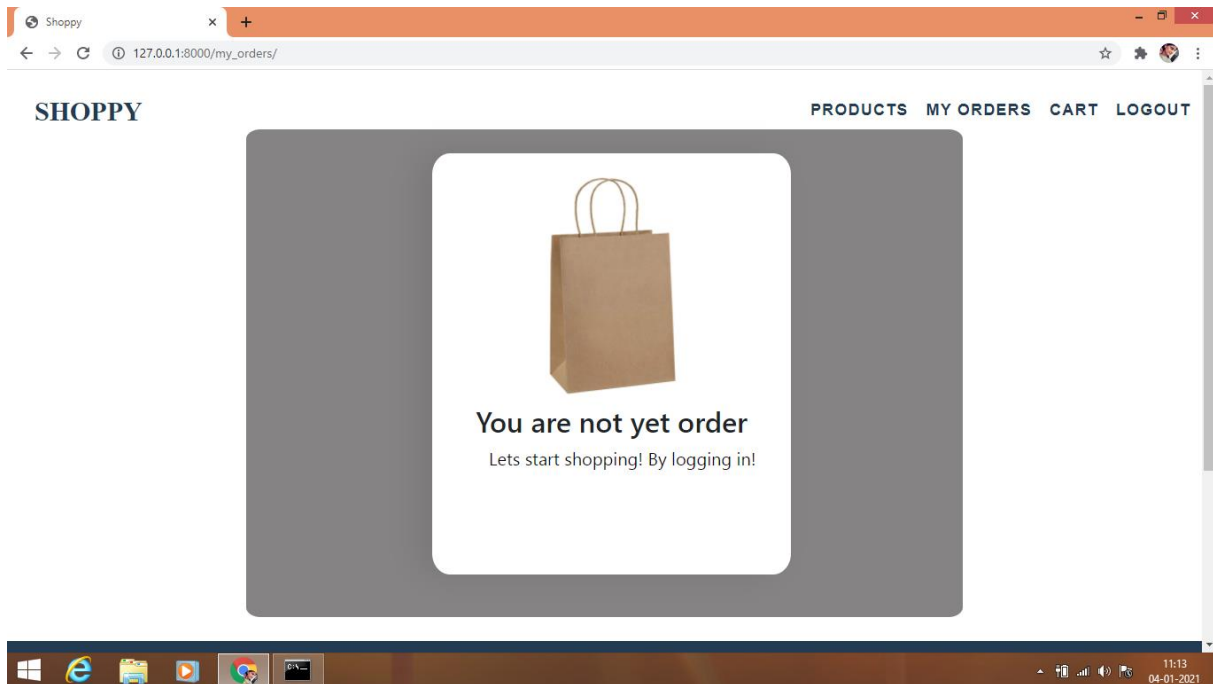
Snapshot 4 : Cart-Page Layout



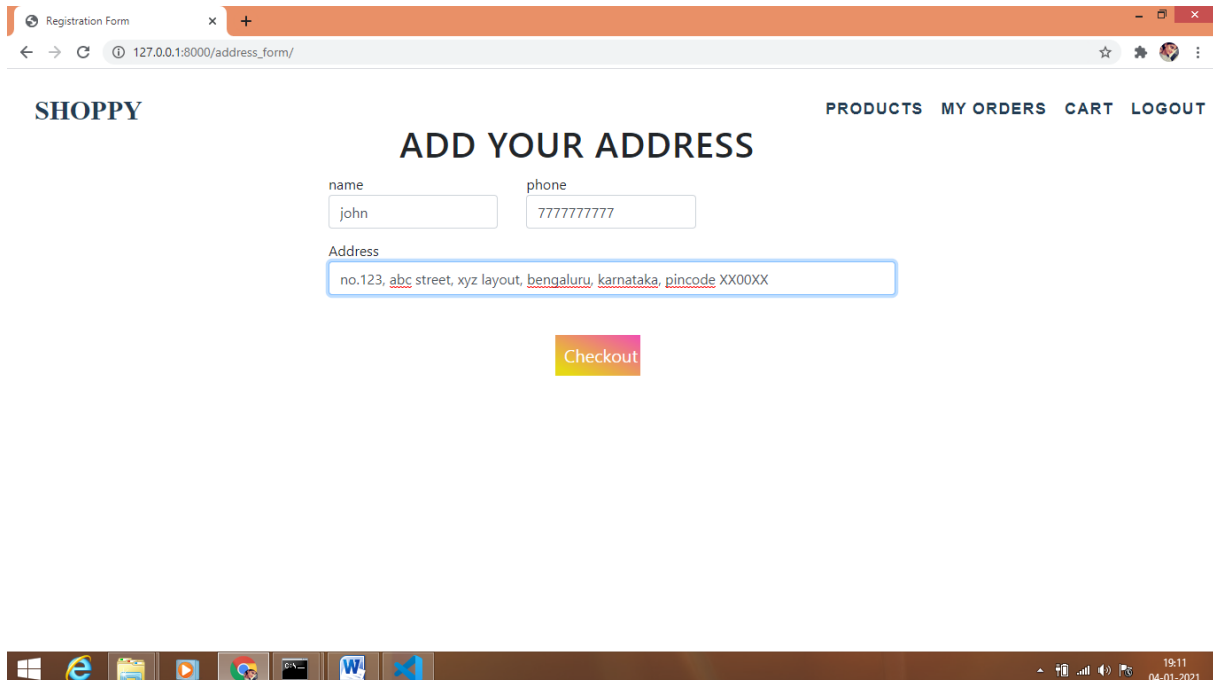
Snapshot 5 : Empty-Cart Page Layout



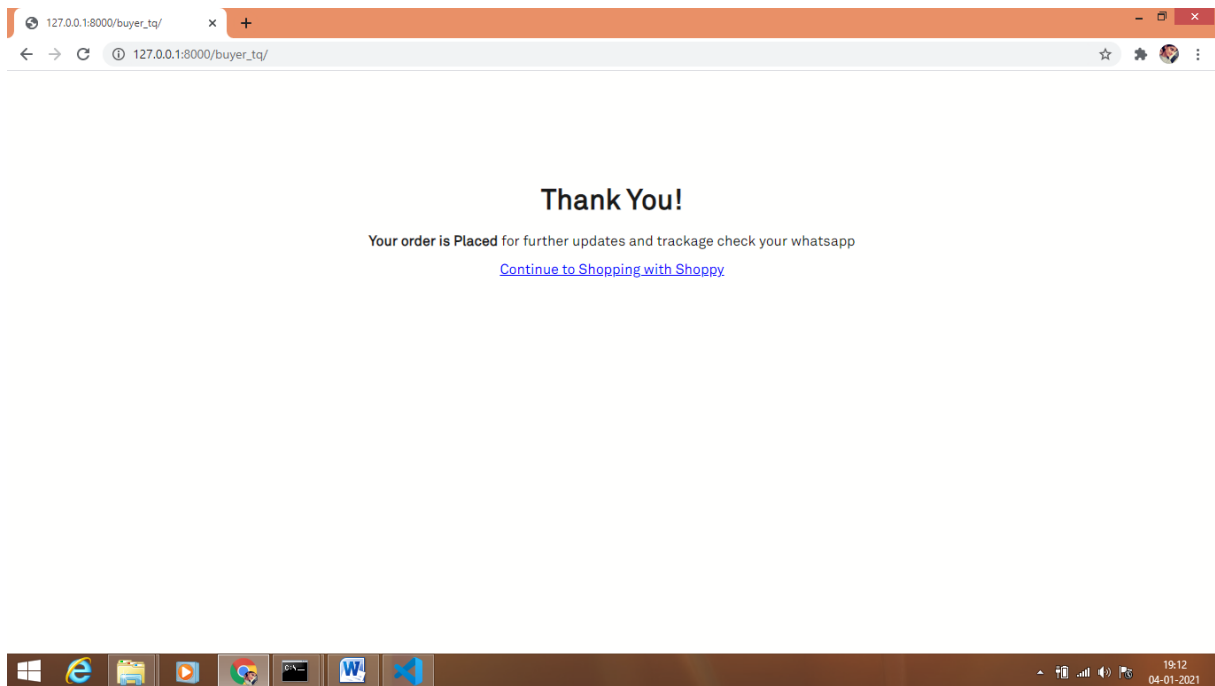
Snapshot 6 : Empty-Order Page Layout



Snapshot 7 : Address_form Page Layout



Snapshot 8 : Thank you Page Layout



CONCLUSION

E-Commerce is not just about conducting business transactions via the Internet. Its impact will be far-reaching, and more prominent than we know currently. This is because the revolution in information technology is happening simultaneously with other developments, especially the globalization of the business. The new age of global e-commerce is creating entirely new economy and that will tremendously change our lives, will reshape the competition in various industries, and alter the economy globally. As companies are gaining high profits, more and more other companies are developing their websites to increase their profits. Since more businesses are being held online resulting in high economy development and emergence of a more innovative and advanced technology.

REFERENCES

- [1] <https://docs.djangoproject.com/en/3.1/>
- [2] <https://dev.mysql.com/doc/>
- [3] <https://getbootstrap.com/>
- [4] <https://docs.python.org/3/>
- [5] https://www.w3schools.com/jsref/dom_obj_document.asp
- [6] <https://stackoverflow.com/documentation>