Online Organic Food Ordering website

CSE3002 – INTERNET & WEB PROGRAMMING

J COMPONENT PROJECT DOCUMENT FALL

2020 - 2021

INDIVIDUAL PROJECT

1. PRAJEETH KUMAR M.J - 18BCE2354

UNDER THE GUIDENCE OF PROF.
NAVEEN KUMAR N SCOPE

School of Computer Science and Engineering

DECLARATION

I/We hereby declare that the J Component project entitled **"Online Organic Food Ordering Website(E-commerce)"** submitted by me/us to the School of Computer Science and Engineering, VIT University, Vellore-14 in partial fulfillment of the requirements for the **Internet and Web Programming (CSE3002)** course, is a record of bonafide work carried out by me/us under the supervision of **Naveen Kumar N, Assistant Professor (Sr).** I further declare that the work reported in this project has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma of this institute or of any other institute or university.

Reg_no: 18BCE2354

Name: Prajeeth Kumar M.J

Chapter Title	Page
Title Page	1
Declaration	2
Table of Contents	5
List of Figures	11
List of Abbreviations	28
Abstract	4

1. Introduction
    - Aim
    - Objective
2. System Requirements
    - (Software requirements)
3. Solution for Existing problems
    - Solution-1
        - Problem description and its Solution
        - Sample code
        - Sample Screenshot
    - Solution-2
        - Problem description and its Solution
        - Sample code
        - Sample Screenshot
4. Recent Technology
    - Technology-1
        - Importance of the technology
        - Sample code
        - Sample Screenshot
    - Technology-2
        - Importance of the technology
        - Sample code
        - Sample Screenshot

5. Screenshots

   Selected frontend design - excluding recent technology and existing problem solution screenshots.

6. Conclusion

Table of contents

Abstract:

Organic food products and other organic ingredients are grown without the use of pesticides, synthetic fertilizers, sewage sludge, or ionizing radiation. Conventional fruits and vegetables are often sprayed with pesticides. When you buy such fruits and vegetables, these stubborn chemicals remain on the food. The second big difference between conventional and organic food is that many conventional foods are genetically modified or contain genetically modified organisms. Organic food is not easily available in the market. There are only some particular shops where organic food is available. E-commerce is a process of doing business through computer network. Online shopping is a form of electronic shopping store where the buyer is directly online to the seller's computer usually via the internet.

To overcome the difficulty of organic food shopping. I decided to propose a online organic store which provides organic foods by just sitting at home and follows E-commerce mode of shopping. This system has two modules namely, Admin and Customers. Admin has authority to add organic food list on the website, view products uploaded, view customers and view the customer's order. Customers can register and login using credentials. The advantages are:

- Saves time.
- Easy to access the system from anywhere and anytime.

This organic food website is the main branch of the city which has many stores or hotels in it , so that users can choose any stores they wish and order their organic products. It is also been tested and certified by FDA.

I chose this topic as Organic food products and other organic ingredients are grown without the use of pesticides, synthetic fertilizers, Conventional fruits and vegetables are often sprayed with pesticides. There are very few online websites for ordering organic foods. Most of the people opt for fast food ordering. So i decided to make this website to create awarness among the people how important organic foods are and also by buying these organic foods , it would be some kind of respect to the farmers who are working hardly for us.

1. Introduction

1.1. Aim

To design and develop a website for the users to choose any organic food hotel or store among many stores and buy organic foods using html, css, javascript, sql , django other recent web technologies and implement solutions for existing problems.

The main aim is to create awareness among the people about the organic foods and I want many people to opt for organic foods rather than fast foods.

1.2. Objective

The objective of this project is to develop an e-commerce website (organic food store) with a user-friendly and responsive interface where users can buy good organic foods certified by the FDA(Food And Drug Administration). The website is to be created using recent technologies like django to solve problems in existing websites. The functionality of the website includes

i. A user can sign in by creating an account

ii. The registered user can login using a username and a password

iii. They can choose any hotels they wish.

iv. The users can view available organic foods. They can search for the restaurants they want.

v. The users can choose what products they want and add to their cart.

vi. They can choose the mode of online payment they want.

vii. Users can also rate the website and give feedback about it.

viii. They can also ask queries to the admin.

2. System Requirements

• python(3) and pip(3) to install django and its requirements.

Node js to run or execute angularjs in local server.

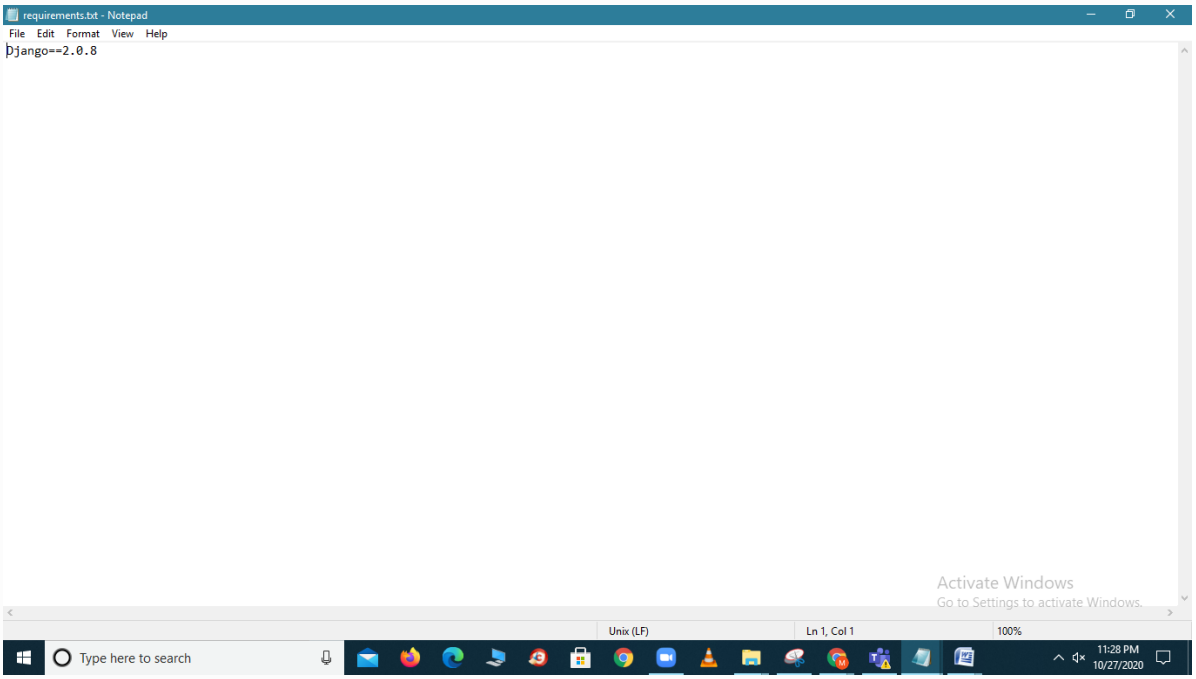
- 1 GB RAM
- 1 CPU Core
- 24 GB SSD Storage
- 2 TB Transfer
- 40 Gbit Network In
- 125 Mbit Network Out

• A web browser

- Python 3.5 to 3.9 supported.
- Django 2.2 to 3.1 supported.
- MySQL 5.6 to 8.0 supported.
- MariaDB 10.1 to 10.5 supported.

- mysqclient 1.3 to 1.4 supported.

Requirements.txt file:



3. Solution for Existing problems

3.1. Solution-1

3.1.1. Problem description and its Solution

This organic food website is the main branch of the city which has many stores or hotels in it , so that users can choose any stores they wish and order their organic products. It is also been tested and certified by FDA. Having many hotels for a website, it can help the users to have a wide range of options for selecting the hotels.

This module is to search restaurants for the users. The users can search the hotels they want instead of seeing the hotels one by one. For example, if there are 10-15 hotels , the users can easily search the restaurants. This reduces the time of the user using this website.(user-friendliness). There is no option like this in the existing website. Therefore this feature is one of the solution for the existing website.

3.1.2. Sample code

In view.py:

```
def restuarent(request):
    r_object = Restaurant.objects.all()
    query   = request.GET.get('q')
    if query:
            r_object=Restaurant.objects.filter(Q(rname__icontains=query)).distinct()
            return render(request,'webapp/restaurents.html',{'r_object':r_object})
    return render(request,'webapp/restaurents.html',{'r_object':r_object})
```

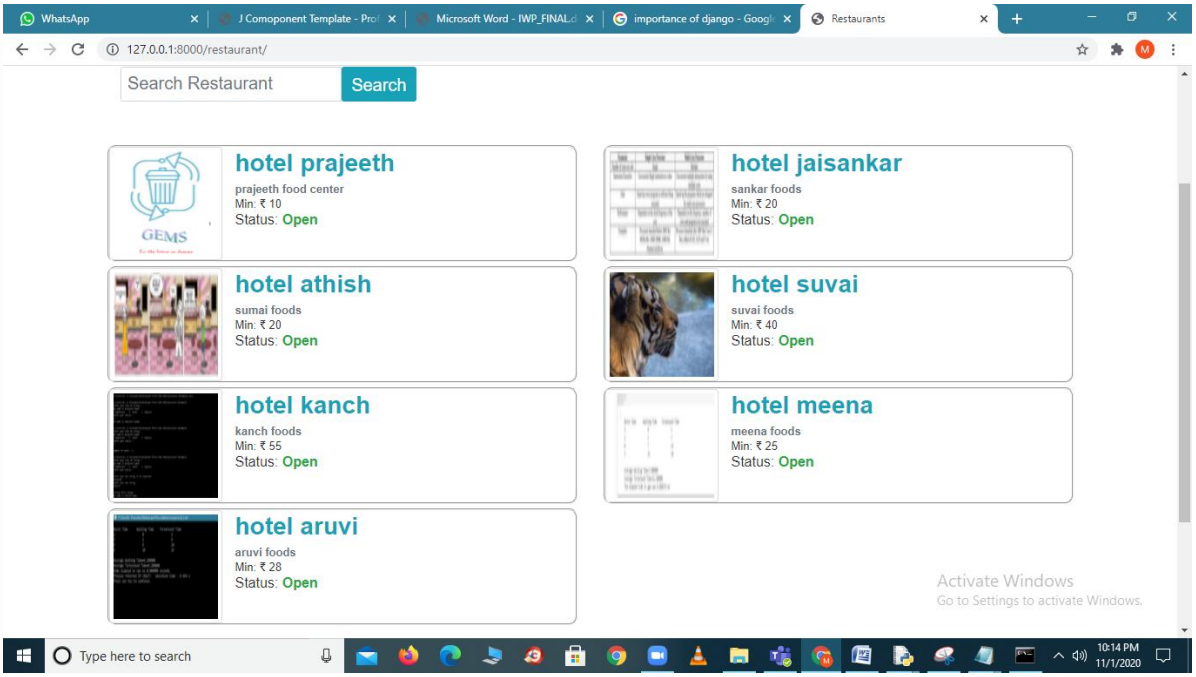
restaurents.html:

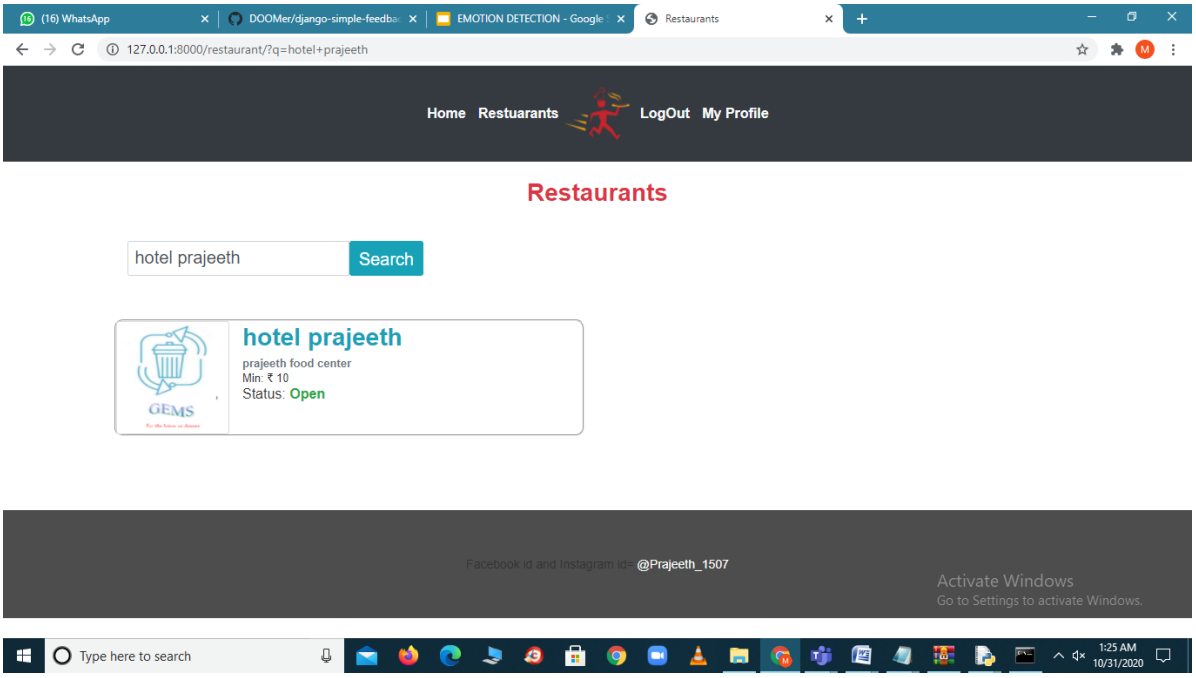
```
{% extends 'webapp/base.html'%}
{% block title %}Restaurants{% endblock %}

{% block body %}
<div class="container-fluid">
    <h1 class="text-center text-danger"><strong>Restaurants</strong></h1>
</div>
<br>
<div class="container">
  <div class="col-sm-4 col-sm-offset-3">
   <form class="form-horizontal" role="search" method="get" action="">
     <div class="form-group ">
     <div class="input-group">
```

```
          <input type="text" class="form-control input-lg " name="q" value="{{
request.GET.q }}" placeholder= "Search Restaurant " style="height: 40px; width: 50%;
font-size: 20px;" />

          <span class='input-group-btn'>

           <button class='btn btn-info btn-lg'style="height:40px; font-size: 20px;"
type='submit'>Search

           </button>

         </span>

        </div>

     </div>

   </form>



 </div>

</div>

<br>

<br>

          <div class="container">

                    <div class="row">

                    {% for rest in r_object %}

       <div class="col-xs-7 col-md-6">

     <div class="thumbnail">

                      <div class="media mb-2 ml-8" style="border-radius: 10px; border-style:
groove; border-width: 2px;">

       <a href="{% url 'menu' rest.id %}" class="pull-left ">

          <img src="{{rest.r_logo.url}}" class="img-thumbnail img-responsive "
style="height: 130px; width: 130px;">

       </a>
```

```
      <div class="media-body pl-4">

        <a href="{% url 'menu' rest.id %}" style="text-decoration: none !important;"><h1
class="media-heading text-info mt-1" style="height:
20px;"><strong>{{rest.rname}}</strong></h1></a>

          <h6 class="text-secondary"><font
size="2"><strong>{{rest.info}}</strong></font></h6>

          <h4><font size="2">Min: &#x20b9; {{rest.min_ord}}</font></h4>

          <h5><font size="3">Status: <strong class="text-
success">{{rest.status}}</strong></font></h5>



      </div>


   </div>


 </div>

</div>


{% endfor%}

</div>

</div>


{% endblock %}
```

3.1.3. Sample Screenshot





3.2. Solution-2

3.2.1. Problem description and its Solution

This module is the feedback form. It asks the feedback from the user to give the rating and review of their overall experience of the website. They can also ask any queries in the form.

This feedback form is very essential for the healthy interaction or relationship between the user and the admin. By getting those feedbacks continuously, the admin could get to know the positive and negative feedbacks of their website and they would try to improve the user requirements.

Usually the form will be at the end in any website, but here it is little before, so that all the users accessing this website would give the feedback and this would be useful for the administrator. Only if the feedback form is filled the user can proceed further and buy their products. It is made compulsory for all the users to give the feedback form.

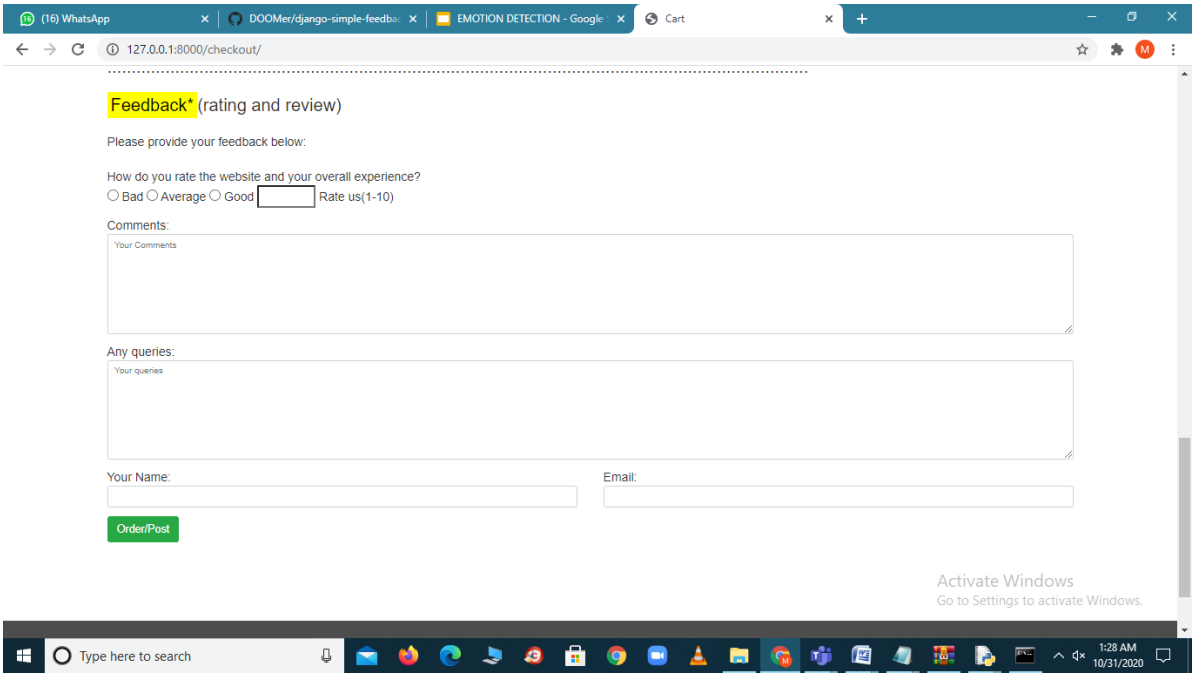
This feature is not in any existing organic store websites. So this is one of the solutions for the existing website.

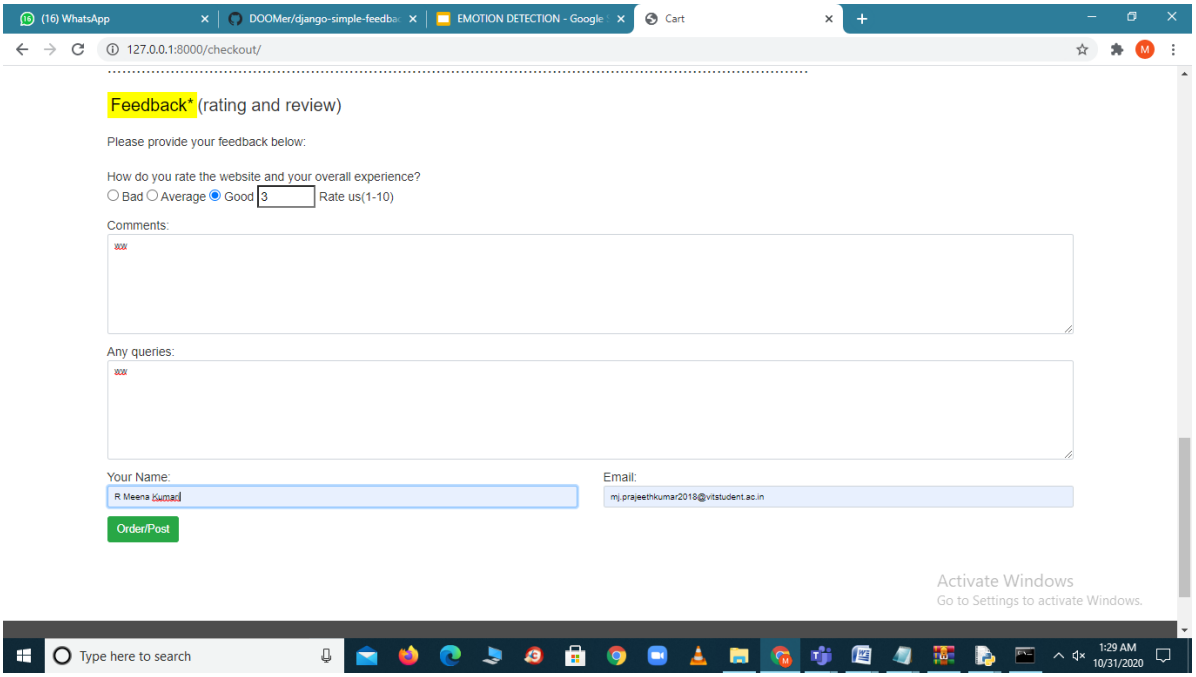
3.2.2. Sample code

```
<h2><mark>Feedback*</mark>(rating and review)</h2>
            <br>
               <p> Please provide your feedback below: </p>
              <br>
               <form role="form" method="post" id="reused_form">
                  <div class="row">
                     <div class="col-sm-12 form-group">
                        <label>How do you rate the website and your overall
experience?</label>
                        <p>
                           <label class="radio-inline">
                              <input type="radio" name="experience" id="radio_experience"
value="bad" >
                              Bad
                           </label>
                           <label class="radio-inline">
                              <input type="radio" name="experience" id="radio_experience"
value="average" >
                              Average
                           </label>
                           <label class="radio-inline">
                              <input type="radio" name="experience" id="radio_experience"
value="good" >
                              Good
                           </label>
                           <label class="radio-inline">
                              <input type="number" name="experience" id="radio_experience"
value="rate us" min="1" max="10">
                              Rate us(1-10)
                           </label>

                        </p>
                     </div>
                  </div>
                  <div class="row">
                     <div class="col-sm-12 form-group">
                        <label for="comments"> Comments:</label>
                        <textarea class="form-control" type="textarea" name="comments"
id="comments" placeholder="Your Comments" maxlength="6000" rows="7"></textarea>
                     </div>
                      <br>
                     <div class="col-sm-12 form-group">
```

```
                        <label for="comments"> Any queries:</label>
                        <textarea class="form-control" type="textarea" name="comments"
id="comments" placeholder="Your queries" maxlength="6000" rows="7"></textarea>
                     </div>
                  </div>
                  <div class="row">
                     <div class="col-sm-6 form-group">
                        <label for="name"> Your Name:</label>
                        <input type="text" class="form-control" id="name" name="name"
required>
                     </div>
                     <div class="col-sm-6 form-group">
                        <label for="email"> Email:</label>
                        <input type="email" class="form-control" id="email" name="email"
required>
                     </div>
                  </div>
                  <div class="row">
                     <div class="col-sm-12 form-group">


                        <input type="submit" class="btn btn-success btn-lg"
value="Order/Post" name="submit">
                     </div>
                  </div>
               </form>
               <div id="success_message" style="width:100%; height:100%; display:none; ">
<h3>Posted your feedback successfully!</h3> </div>
               <div id="error_message" style="width:100%; height:100%; display:none; ">
<h3>Error</h3> Sorry there was an error sending your form. </div>
            </div>
         </div>
      </div>




</div>
```

3.2.3. Sample Screenshot





4. Recent Technology

4.1. Technology 1 – Django

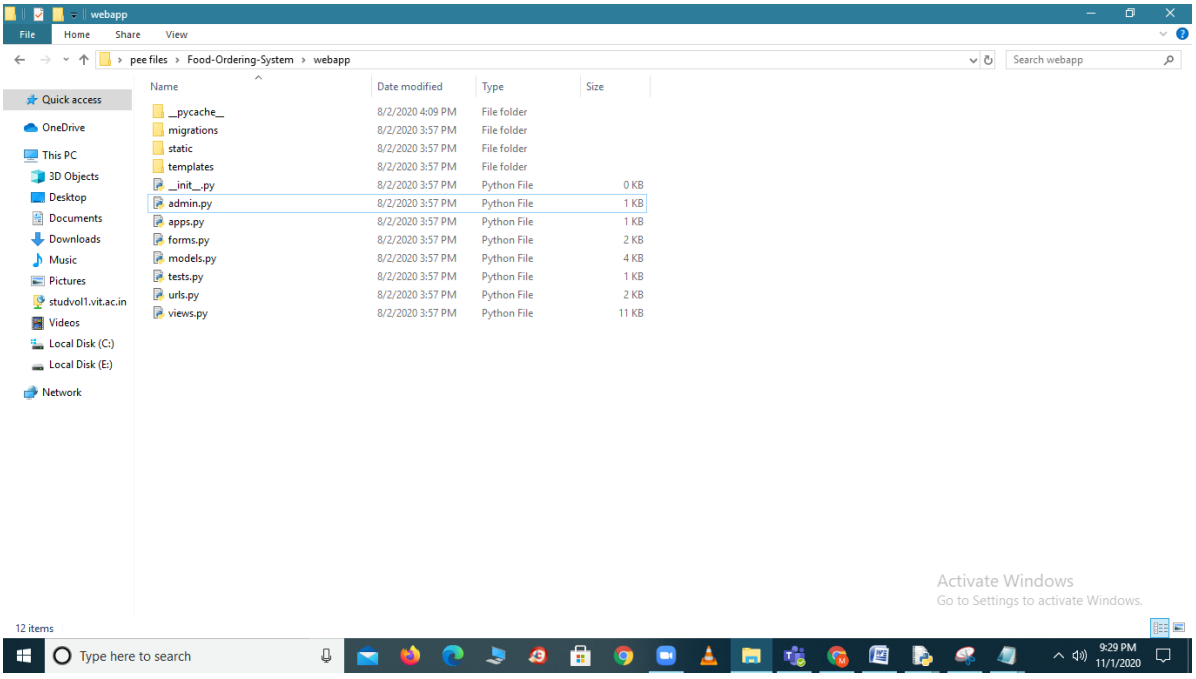
4.1.1. Importance of the technology

Django is an open-source python web framework **used** for rapid development, pragmatic, maintainable, clean design, and secure websites. The main goal of the **Django** framework is to allow developers to focus on components of the application that are new instead of spending time on already developed components. It has an automatic admin interface.

While speaking about the web apps **future**, **Django has** a lot to offer and **is** capable to serve any modern web application structures. Using the **Django** framework **is** one of the best

ways to develop modern web applications. Learn the framework that **is** in-demand if you want to build a successful career.

4.1.2. Sample code:



admin.py:

```
from django.contrib import admin
from .models import Customer,Restaurant,Item,Menu,Order,orderItem,User

admin.site.register(User)
admin.site.register(Customer)
admin.site.register(Restaurant)
admin.site.register(Item)
admin.site.register(Menu)
admin.site.register(Order)
admin.site.register(orderItem)
```

models.py

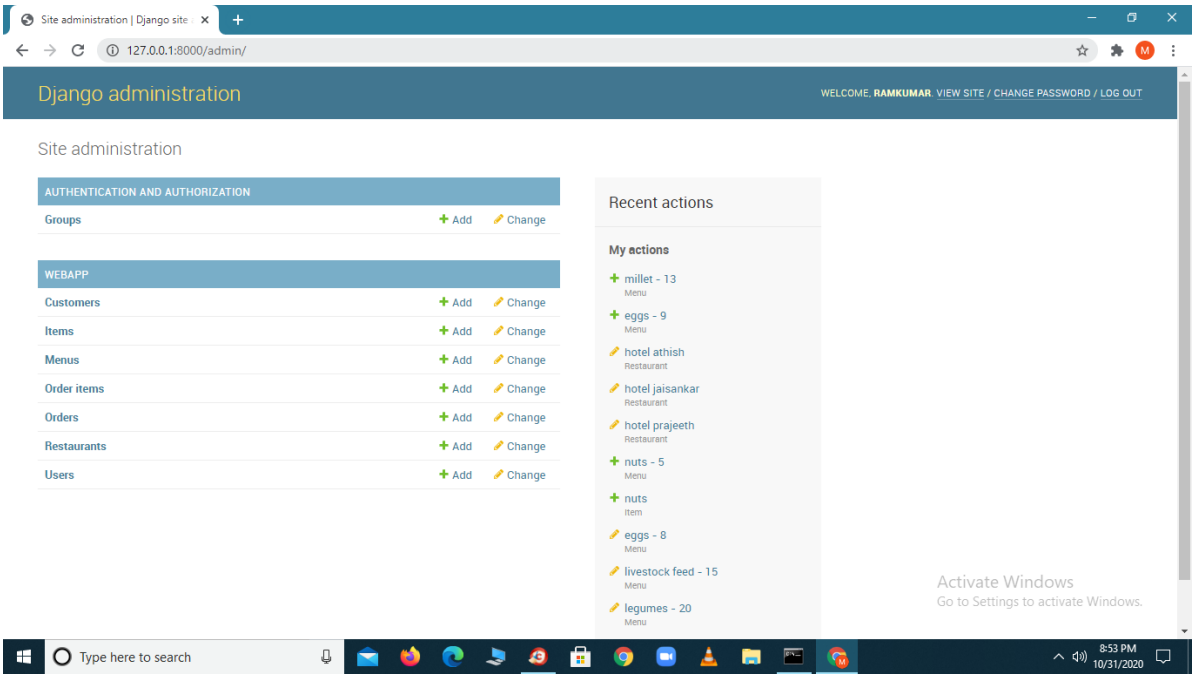
```
from django.db import models
from django.contrib.auth.models import AbstractUser
from django.conf import settings
from django.db.models.signals import post_save

class User(AbstractUser):
        is_customer   = models.BooleanField(default=False)
        is_restaurant = models.BooleanField(default=False)


class Customer(models.Model):
        user                =
```

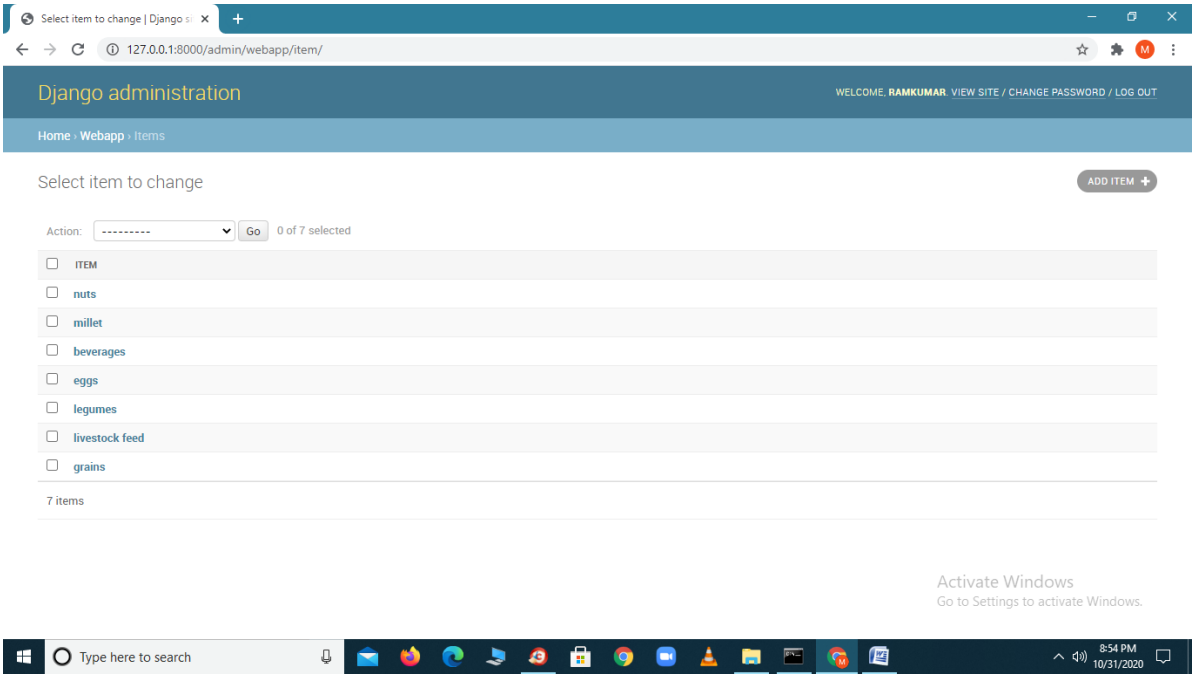
```
models.OneToOneField(settings.AUTH_USER_MODEL,on_delete=models.CASCADE)
        f_name          = models.CharField(max_length=20,blank=False)
        l_name          = models.CharField(max_length=20,blank=False)
        city            = models.CharField(max_length=40,blank=False)
        phone           = models.CharField(max_length=10,blank=False)
        address         = models.TextField()

        def __str__(self):
                return self.user.username

class Restaurant(models.Model):
        user       =
models.OneToOneField(settings.AUTH_USER_MODEL,on_delete=models.CASCADE)
        rname           = models.CharField(max_length=100,blank=False)
        info            = models.CharField(max_length=40,blank=False)
        min_ord                 = models.CharField(max_length=5,blank=False)
        location    = models.CharField(max_length=40,blank=False)
        r_logo      = models.FileField(blank=False)

        REST_STATE_OPEN    = "Open"
        REST_STATE_CLOSE   = "Closed"
        REST_STATE_CHOICES =(
                        (REST_STATE_OPEN,REST_STATE_OPEN),
                        (REST_STATE_CLOSE,REST_STATE_CLOSE)
                )
        status   =
models.CharField(max_length=50,choices=REST_STATE_CHOICES,default=REST_STA
TE_OPEN,blank=False)
        approved = models.BooleanField(blank=False,default=True)

        def __str__(self):
                return self.rname

class Item(models.Model):
        id                      = models.AutoField(primary_key=True)
        fname           = models.CharField(max_length=30,blank=False)
        category        = models.CharField(max_length=50,blank=False)

        def __str__(self):
                return self.fname

class Menu(models.Model):
        id                = models.AutoField(primary_key=True)
        item_id  = models.ForeignKey(Item,on_delete=models.CASCADE)
        r_id     = models.ForeignKey(Restaurant,on_delete=models.CASCADE)
        price    = models.IntegerField(blank=False)
        quantity = models.IntegerField(blank=False,default=0)

        def __str__(self):
                return self.item_id.fname+' - '+str(self.price)
```

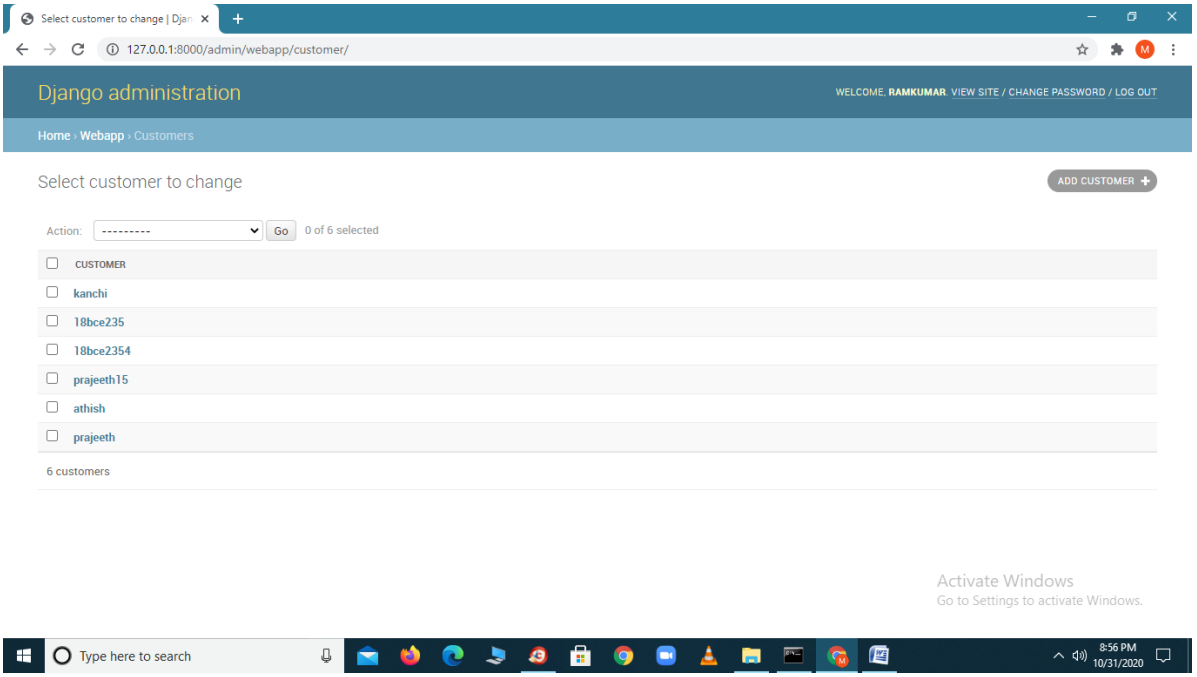
```
class Order(models.Model):
        id                              = models.AutoField(primary_key=True)
        total_amount    = models.IntegerField(default=0)
        timestamp       = models.DateTimeField(auto_now_add=True)
        delivery_addr   = models.CharField(max_length=50,blank=True)
        orderedBy       = models.ForeignKey(User ,on_delete=models.CASCADE)
        r_id                    = models.ForeignKey(Restaurant
,on_delete=models.CASCADE)

        ORDER_STATE_WAITING             = "Waiting"
        ORDER_STATE_PLACED              = "Placed"
        ORDER_STATE_ACKNOWLEDGED = "Acknowledged"
        ORDER_STATE_COMPLETED    = "Completed"
        ORDER_STATE_CANCELLED    = "Cancelled"
        ORDER_STATE_DISPATCHED   = "Dispatched"

        ORDER_STATE_CHOICES = (
                (ORDER_STATE_WAITING,ORDER_STATE_WAITING),
           (ORDER_STATE_PLACED, ORDER_STATE_PLACED),
           (ORDER_STATE_ACKNOWLEDGED,
ORDER_STATE_ACKNOWLEDGED),
           (ORDER_STATE_COMPLETED, ORDER_STATE_COMPLETED),
           (ORDER_STATE_CANCELLED, ORDER_STATE_CANCELLED),
           (ORDER_STATE_DISPATCHED, ORDER_STATE_DISPATCHED)
        )
        status =
models.CharField(max_length=50,choices=ORDER_STATE_CHOICES,default=ORDER_
STATE_WAITING)

        def __str__(self):
                return str(self.id) +' '+self.status


class orderItem(models.Model):
        id                              = models.AutoField(primary_key=True)
        item_id         = models.ForeignKey(Menu ,on_delete=models.CASCADE)
        ord_id          = models.ForeignKey(Order,on_delete=models.CASCADE)
        quantity        = models.IntegerField(default=0)

        def __str__(self):
                return str(self.id)
```

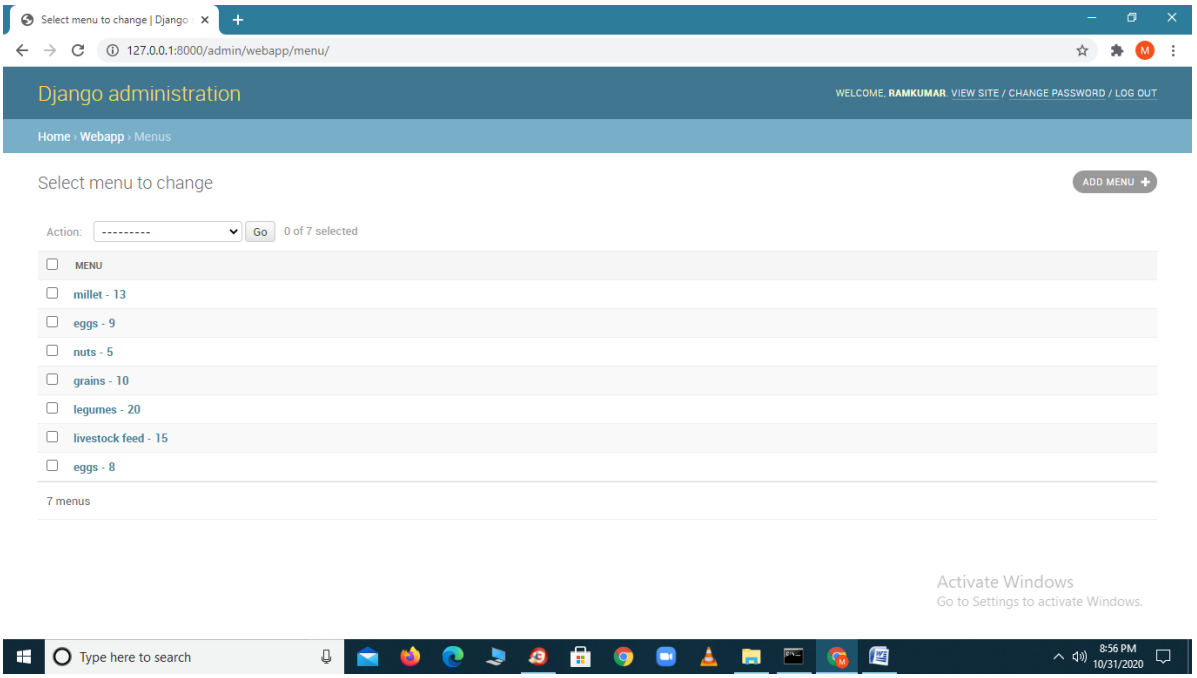
4.1.3. Sample Screenshot

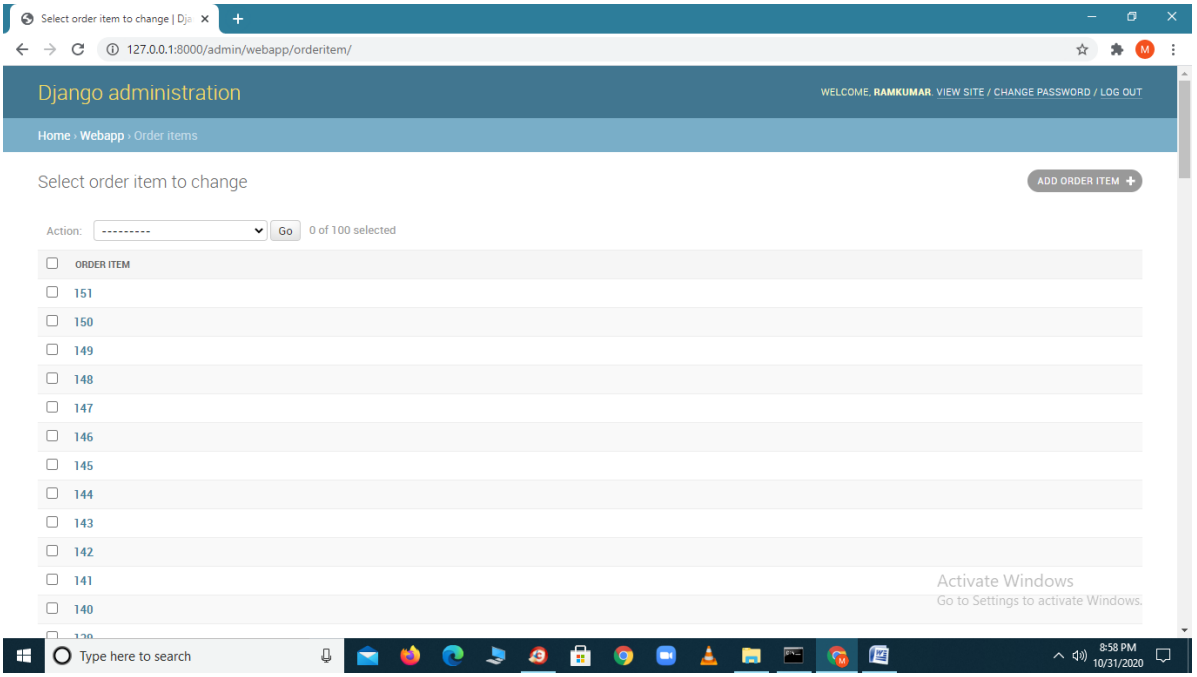
This is the django admin interface. We can insert,edit and delete the modules(customers,items,menus,order items,orders,restaurants,users).


Site administration | Django site
127.0.0.1:8000/admin/
Django administration
WELCOME, RAMKUMAR. VIEW SITE / CHANGE PASSWORD / LOG OUT
Site administration
AUTHENTICATION AND AUTHORIZATION
Groups + Add Change
WEBAPP
Customers + Add Change
Items + Add Change
Menus + Add Change
Order items + Add Change
Orders + Add Change
Restaurants + Add Change
Users + Add Change
Recent actions
My actions
millet - 13 Menu
eggs - 9 Menu
hotel athish Restaurant
hotel jaisankar Restaurant
hotel prajeeth Restaurant
nuts - 5 Menu
nuts Item
eggs - 8 Menu
livestock feed - 15 Menu
legumes - 20 Menu
Activate Windows
Go to Settings to activate Windows.
Type here to search
8:53 PM
10/31/2020

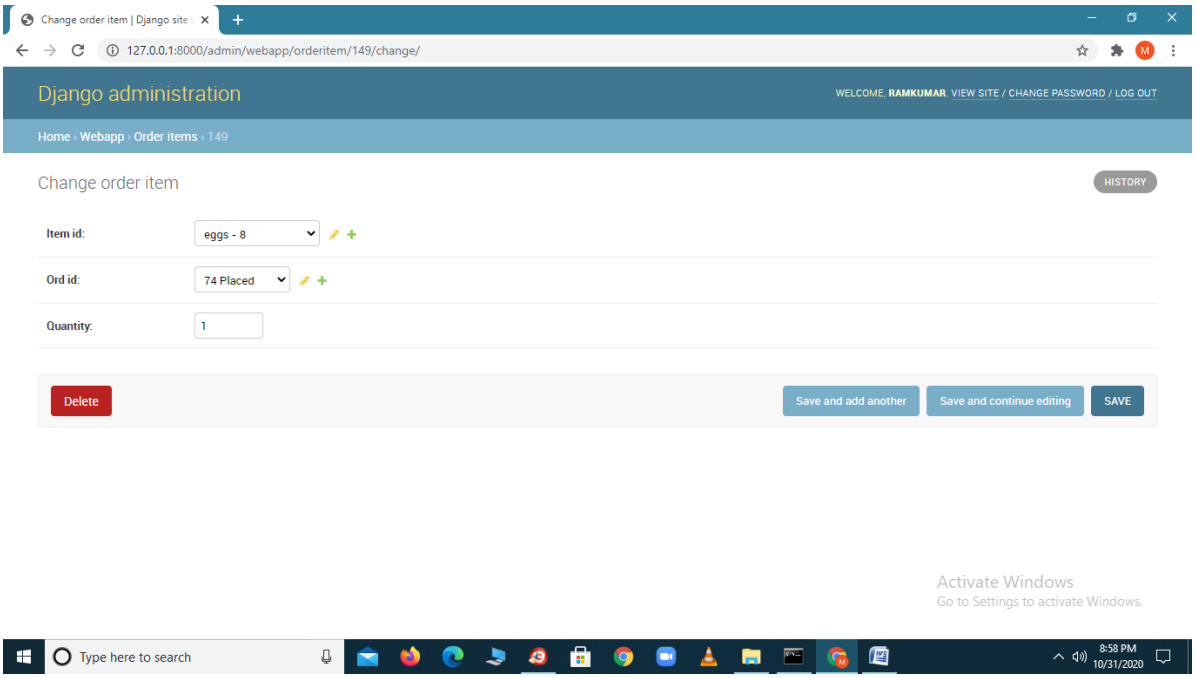

Select item to change | Django si
127.0.0.1:8000/admin/webapp/item/
Django administration
WELCOME, RAMKUMAR. VIEW SITE / CHANGE PASSWORD / LOG OUT
Home › Webapp › Items
Select item to change
ADD ITEM +
Action: --------- Go 0 of 7 selected
ITEM
nuts
millet
beverages
eggs
legumes
livestock feed
grains
7 items
Activate Windows
Go to Settings to activate Windows.
Type here to search
8:54 PM
10/31/2020

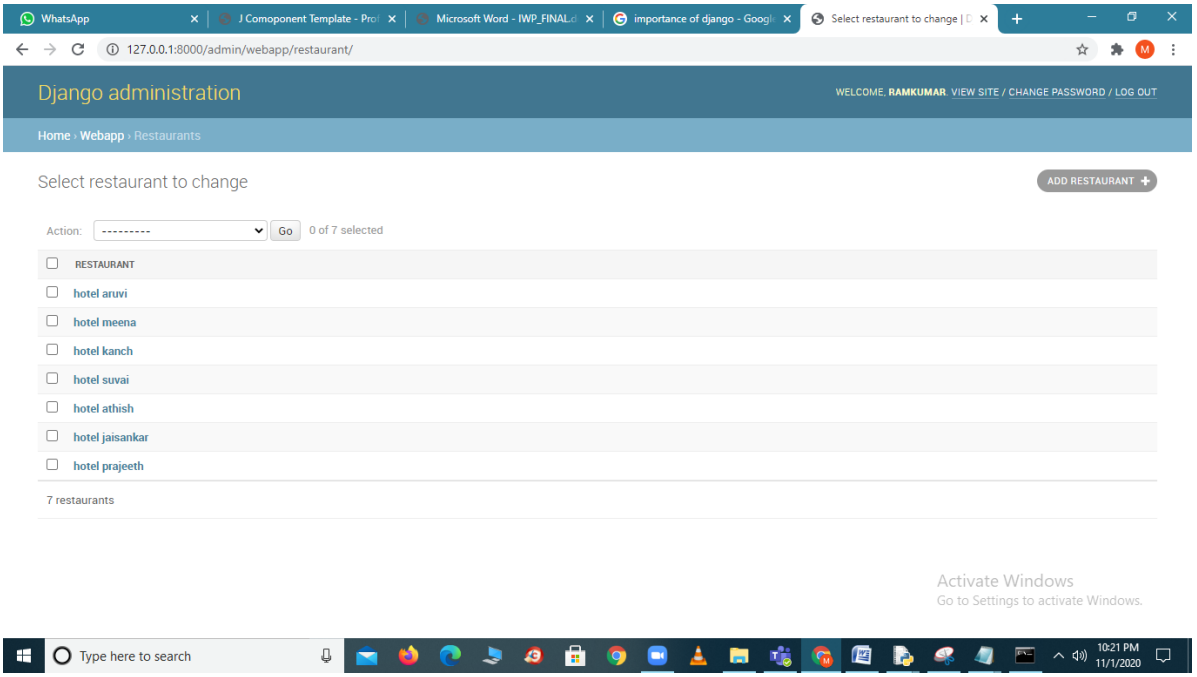

Select customer to change | Djan
127.0.0.1:8000/admin/webapp/customer/
Django administration
WELCOME, RAMKUMAR. VIEW SITE / CHANGE PASSWORD / LOG OUT
Home › Webapp › Customers
Select customer to change
ADD CUSTOMER
Action: --------- Go 0 of 6 selected
CUSTOMER
kanchi
18bce235
18bce2354
prajeeth15
athish
prajeeth
6 customers
Activate Windows
Go to Settings to activate Windows.
Type here to search
8:56 PM
10/31/2020

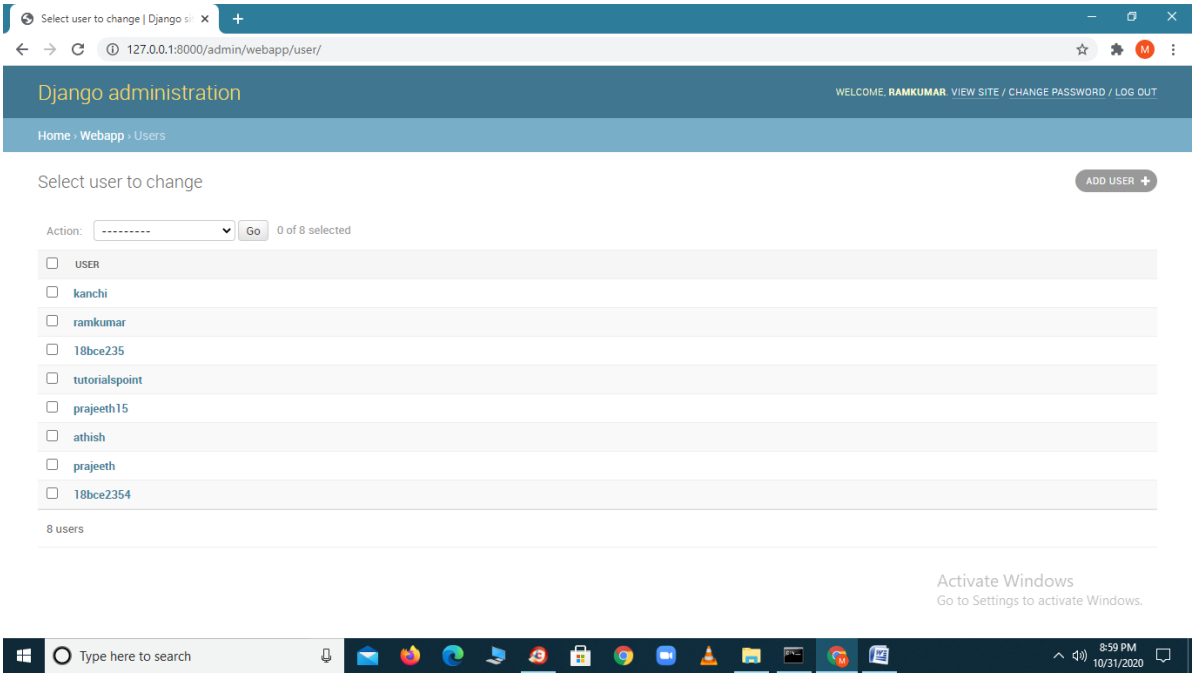

Select menu to change | Django
127.0.0.1:8000/admin/webapp/menu/
Django administration
WELCOME, RAMKUMAR. VIEW SITE / CHANGE PASSWORD / LOG OUT
Home › Webapp › Menus
Select menu to change
ADD MENU
Action: --------- Go 0 of 7 selected
MENU
millet - 13
eggs - 9
nuts - 5
grains - 10
legumes - 20
livestock feed - 15
eggs - 8
7 menus
Activate Windows
Go to Settings to activate Windows.
Type here to search
8:56 PM
10/31/2020



This is the list of order items ordered by the customer.




WhatsApp
J Comoponent Template - Prof
Microsoft Word - IWP_FINAL.d
importance of django - Google
Select restaurant to change | D
127.0.0.1:8000/admin/webapp/restaurant/
Django administration
WELCOME, RAMKUMAR. VIEW SITE / CHANGE PASSWORD / LOG OUT
Home › Webapp › Restaurants
Select restaurant to change
ADD RESTAURANT
Action: --------- Go 0 of 7 selected
RESTAURANT
hotel aruvi
hotel meena
hotel kanch
hotel suvai
hotel athish
hotel jaisankar
hotel prajeeth
7 restaurants
Activate Windows
Go to Settings to activate Windows.
Type here to search
10:21 PM
11/1/2020


Select user to change | Django si
127.0.0.1:8000/admin/webapp/user/
Django administration
WELCOME, RAMKUMAR. VIEW SITE / CHANGE PASSWORD / LOG OUT
Home › Webapp › Users
Select user to change
ADD USER
Action: --------- Go 0 of 8 selected
USER
kanchi
ramkumar
18bce235
tutorialspoint
prajeeth15
athish
prajeeth
18bce2354
8 users
Activate Windows
Go to Settings to activate Windows.
Type here to search
8:59 PM
10/31/2020

4.2. Technology 2 – Angular Js

4.2.1. Importance of the technology

AngularJS is the most preferred framework for building creative components of the website. As it is the type of JavaScript, it was designed as a full-featured JavaScript framework to enhance simplicity and efficiency.

Developers find AngularJS very effective in terms of creating dynamic, single page application, and supporting MVC(Model-View-Controller) programming structure. At a present span of time, in the competitive business environment, AngularJs has emerged as the superhero due to its intuitiveness and rich features.

4.2.2. Sample Code

Login.html:

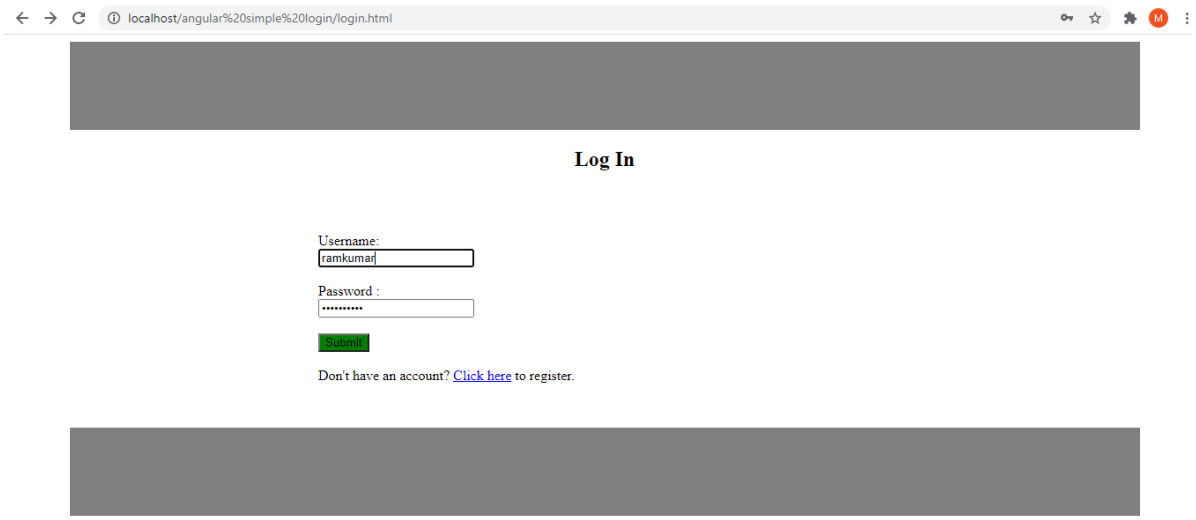
```
<!DOCTYPE html>
<html>
<head>
  <title>login</title>

</head>
<link href="style.css" rel='stylesheet' type='text/css'>
<script type="text/javascript"
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.4/angular.min.js">
</script>
<body >

<header ng-include="'header.html'"></header>

  <h2 align="center">Log In</h2>
<div  style=" margin-left:300px;margin-right: 300px; padding: 50px;">

  <form ng-app="myApp" ng-controller="myCtrl"
name="myForm" >
  <label class="control-label col-sm-2" for="id_username">
        Username:
                    </label>
                    <div class="col-sm-6">
                       <input ng-moddel="pass" name="pass" type="text">
                    </div>
<br>
  Password :
<br>
<input type="password" ng-model="pass" name="pass"><br><br>

<button type="button" name="login" ng-click="submit()" style="background-
color:green">Submit</button>
<br>
<br>
   Don't have an account? <a href="register.html">Click here</a> to register.
```

```
</form>

</div>
<footer ng-include="'footer.html'"></footer>
<script type="text/javascript">
  var app = angular.module('myApp',[]);
  app.controller('myCtrl',function($scope){
    $scope.user = 'ramkumar';
    $scope.pass = 'prajeeth15';
    $scope.submit=function(){
      if($scope.user == 'ramkumar' )
    {
      if($scope.pass == 'prajeeth15')
        {location="loggedin.html";}
      else{alert("Wrong Password");}
    }
    else{alert("Wrong Username");}

    };
  });
</script>
</body>
</html>
```

Styel.css:

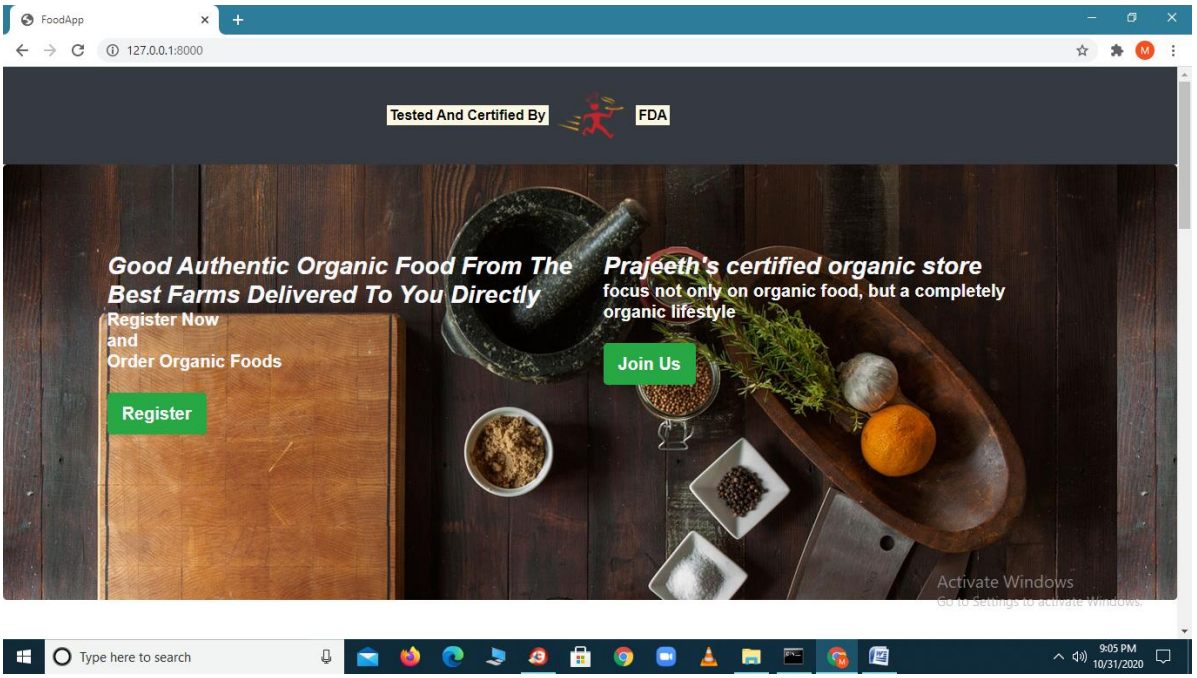
```
/* header */
header{
 width: 90%;
 padding-top: 50px;
 padding-bottom: 50px;
 text-align: center;
 margin: 0 auto;
 background-color: grey;
 color: white;

}

/* footer */
footer{
 width: 90%;
 padding-top: 50px;
 padding-bottom: 50px;
 text-align: center;
 margin: 0 auto;
 background-color: grey;
 color: white;
}
```

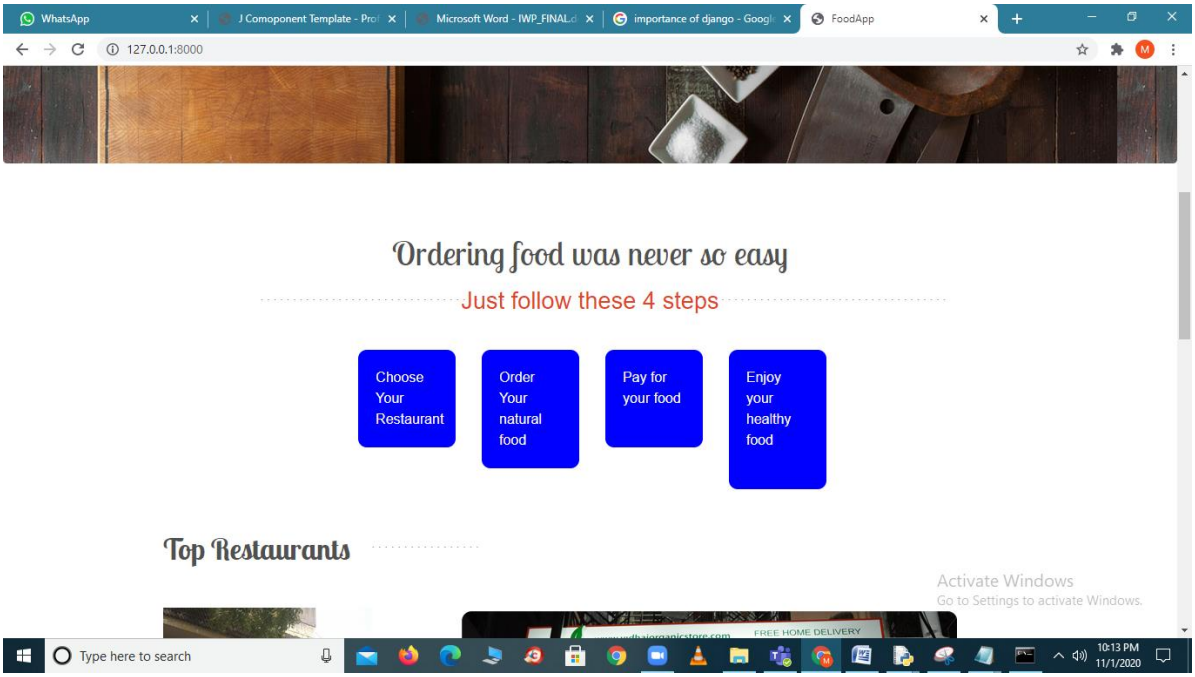
4.2.2. Screenshots

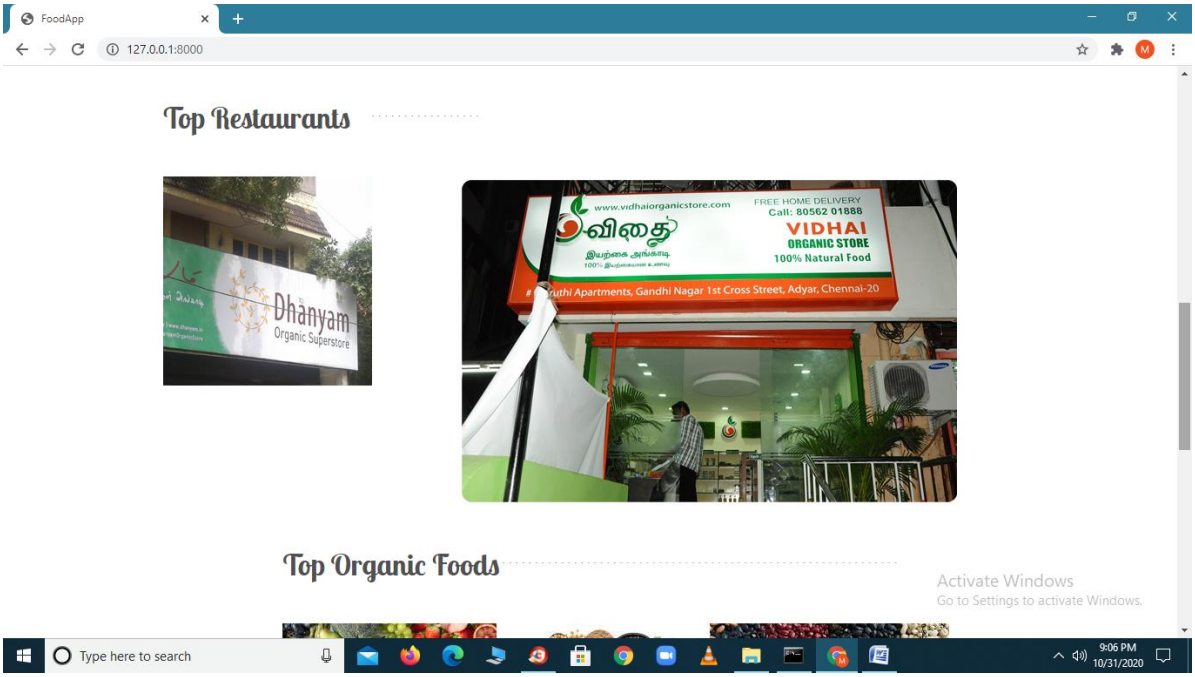


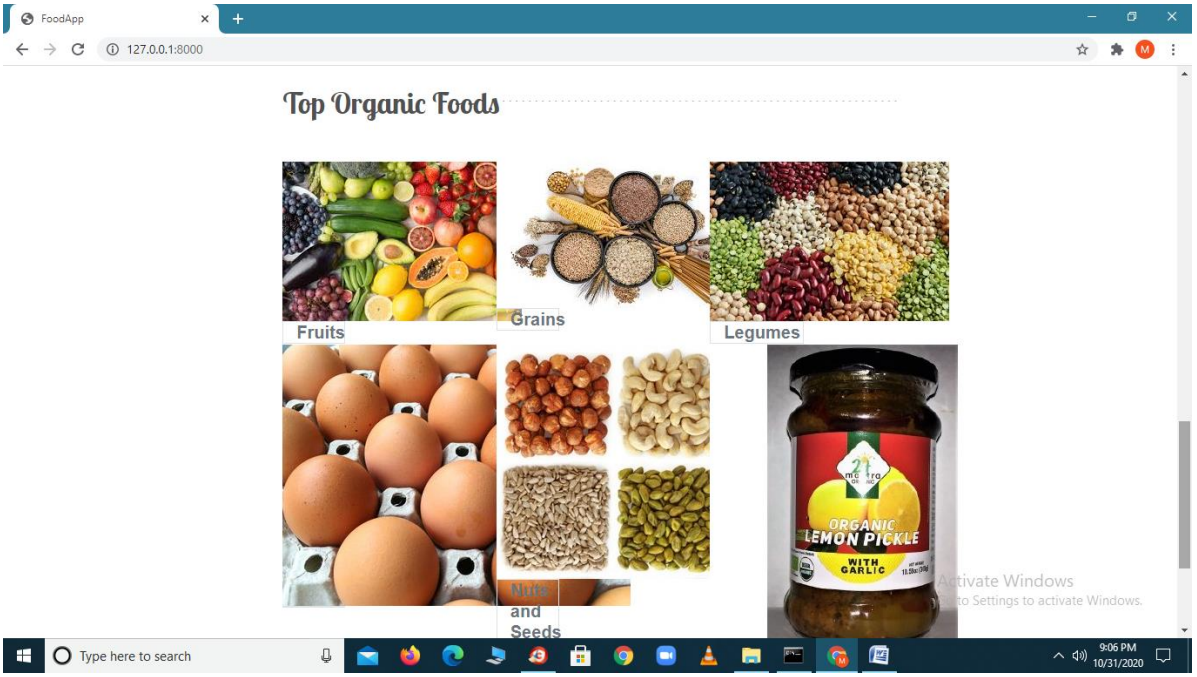
5. Screenshots

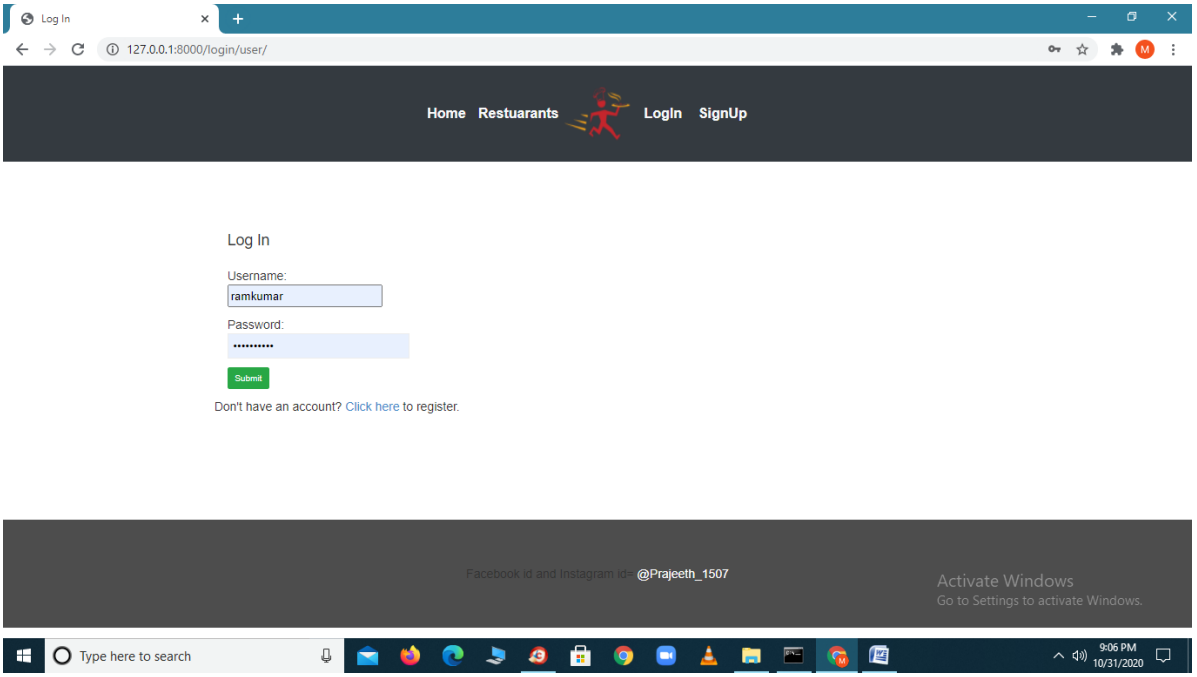
Selected frontend design

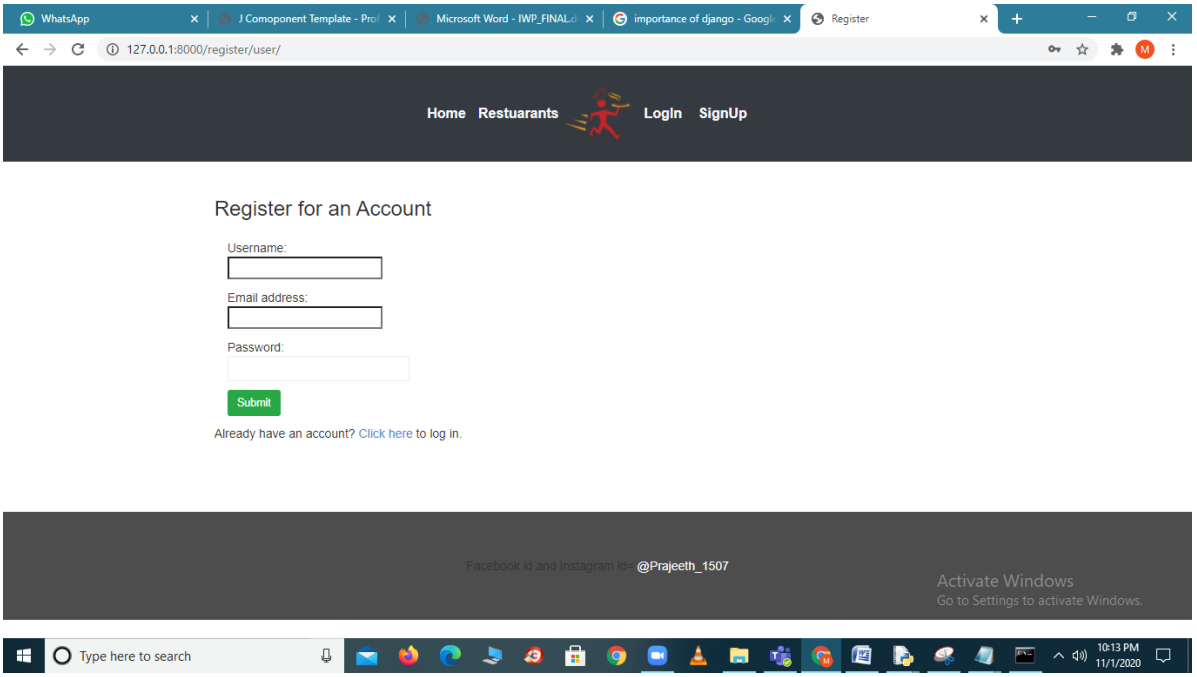


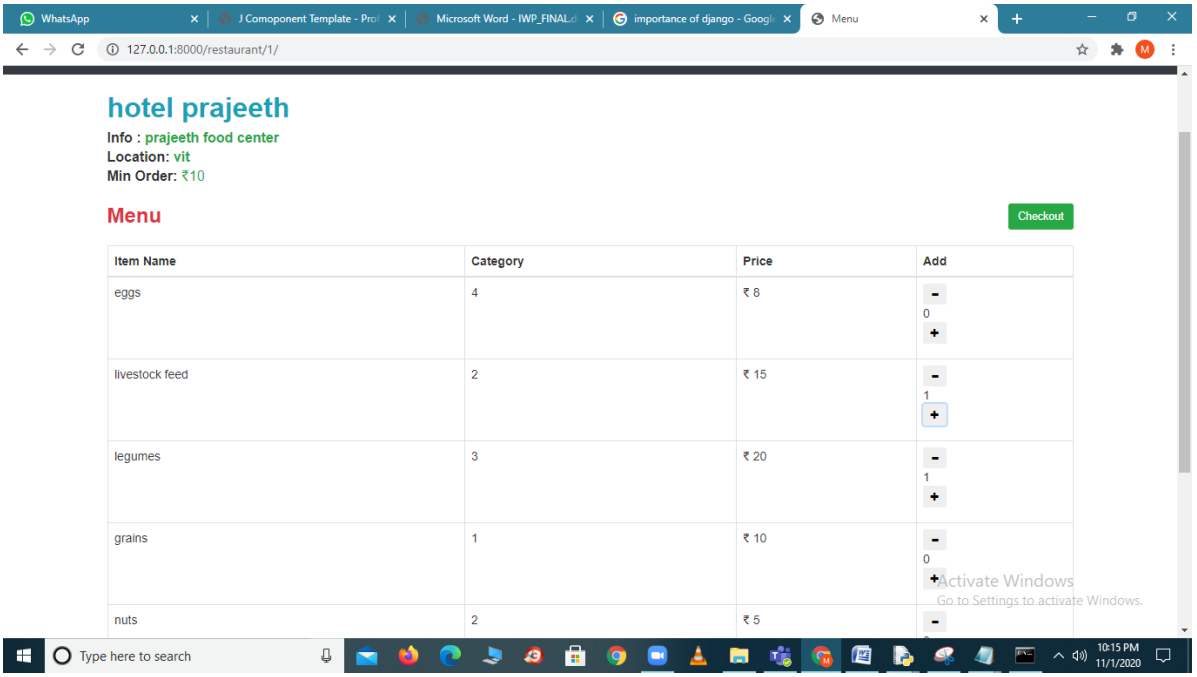

FoodApp
127.0.0.1:8000
Ordering food was never so easy
Just follow these 4 steps
Choose Your Restaurant
Order Your natural food
Pay for your food
Enjoy your healthy food
Top Restaurants
Activate Windows
Go to Settings to activate Windows.
Type here to search
10:13 PM
11/1/2020

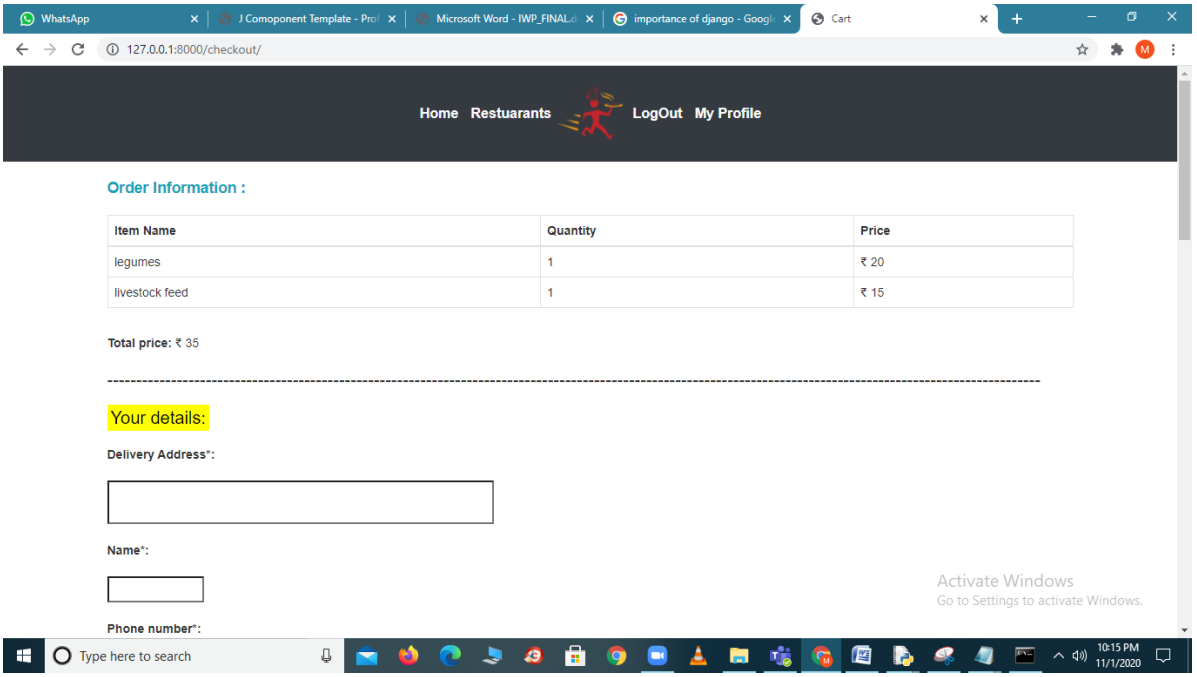

FoodApp
127.0.0.1:8000
Top Restaurants
Dhanyam
Organic Superstore
www.vidhaiorganicstore.com
FREE HOME DELIVERY
Call: 80562 01888
VIDHAI
ORGANIC STORE
100% Natural Food
Top Organic Foods
Activate Windows
Go to Settings to activate Windows.
Type here to search
9:06 PM
10/31/2020

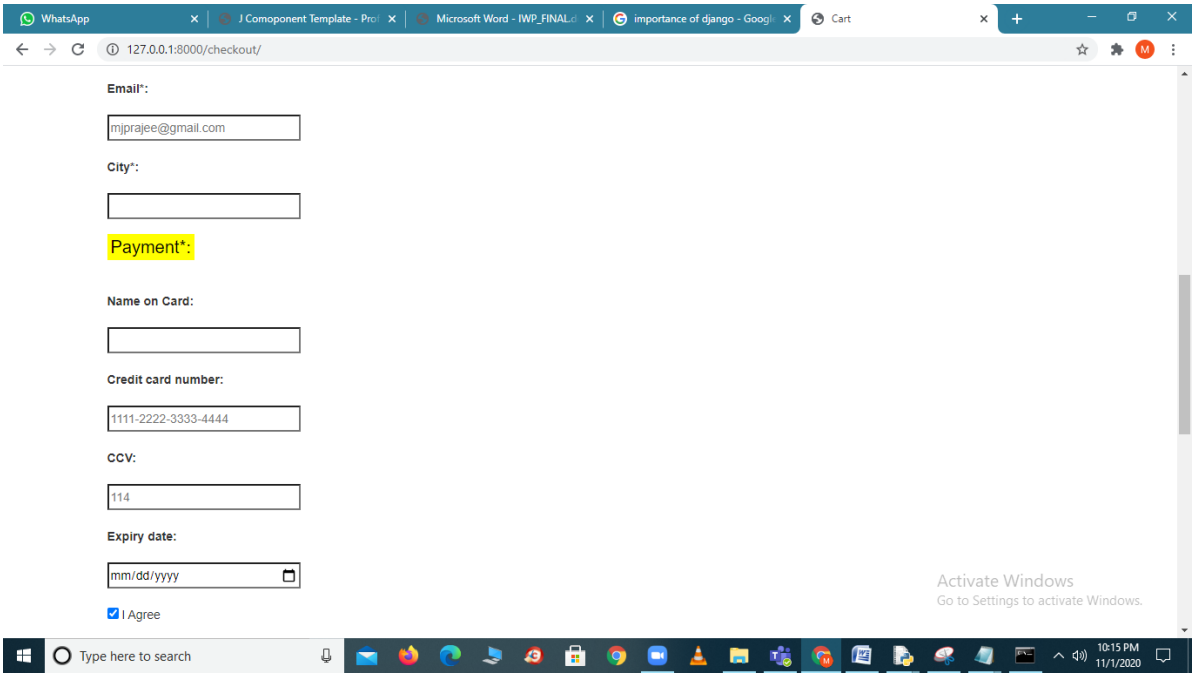

FoodApp
127.0.0.1:8000
Top Organic Foods
Fruits
Grains
Legumes
Nuts and Seeds
ORGANIC LEMON PICKLE
WITH GARLIC
Activate Windows
Type here to search
9:06 PM
10/31/2020

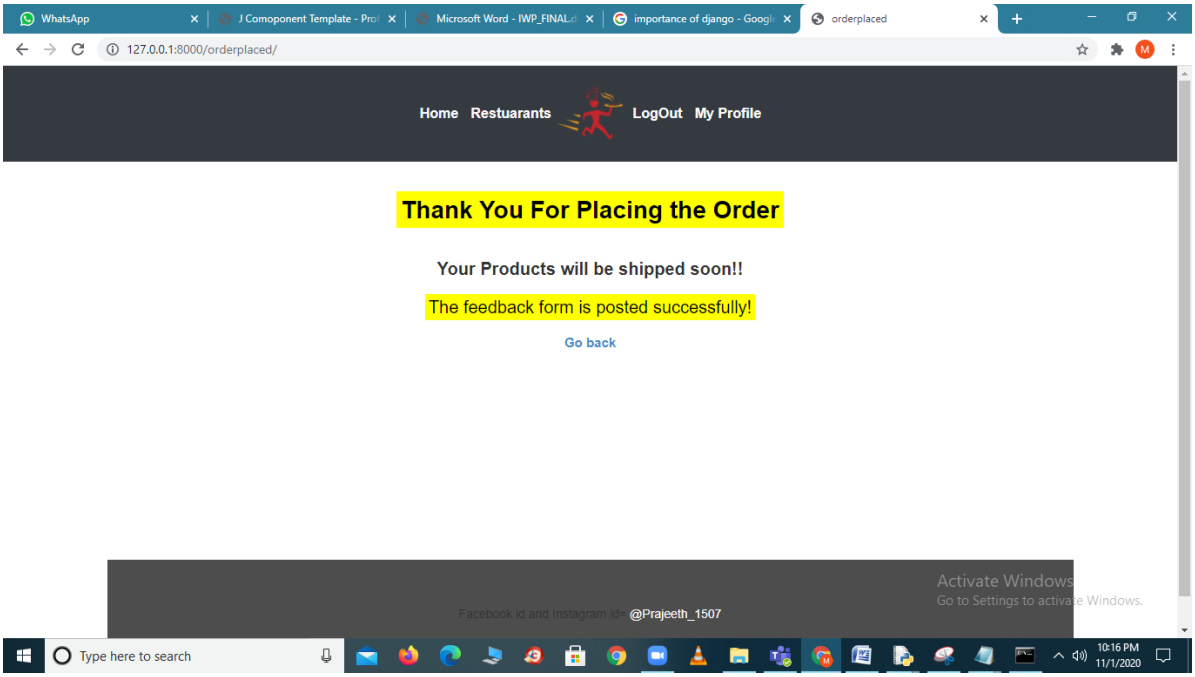

Log In
127.0.0.1:8000/login/user/
Home
Restuarants
LogIn
SignUp
Log In
Username:
ramkumar
Password:
Submit
Don't have an account? Click here to register.
Facebook id and Instagram id= @Prajeeth_1507
Activate Windows
Type here to search
9:06 PM
10/31/2020


WhatsApp
J Comoponent Template - Prof
Microsoft Word - IWP_FINAL.d
importance of django - Googl
Register
127.0.0.1:8000/register/user/
Home
Restuarants
LogIn
SignUp
Register for an Account
Username:
Email address:
Password:
Submit
Already have an account? Click here to log in.
Facebook id and Instagram id= @Prajeeth_1507
Activate Windows
Go to Settings to activate Windows.
Type here to search
10:13 PM
11/1/2020


WhatsApp
J Comoponent Template - Prof
Microsoft Word - IWP_FINAL.d
importance of django - Googl
Menu
127.0.0.1:8000/restaurant/1/
hotel prajeeth
Info : prajeeth food center
Location: vit
Min Order: ₹10
Menu
Checkout
Item Name
Category
Price
Add
eggs
4
₹ 8
livestock feed
2
₹ 15
legumes
3
₹ 20
grains
1
₹ 10
nuts
2
₹ 5
Activate Windows
Go to Settings to activate Windows.
Type here to search
10:15 PM
11/1/2020


WhatsApp
J Comoponent Template - Prof
Microsoft Word - IWP_FINAL.d
importance of django - Googl
Cart
127.0.0.1:8000/checkout/
Home Restuarants LogOut My Profile
Order Information :
Item Name
Quantity
Price
legumes
1
₹ 20
livestock feed
1
₹ 15
Total price: ₹ 35
Your details:
Delivery Address*:
Name*:
Phone number*:
Activate Windows
Go to Settings to activate Windows.
Type here to search
10:15 PM
11/1/2020


WhatsApp
J Comoponent Template - Prof
Microsoft Word - IWP_FINAL.d
importance of django - Googl
Cart
127.0.0.1:8000/checkout/
Email*:
mjprajee@gmail.com
City*:
Payment*:
Name on Card:
Credit card number:
1111-2222-3333-4444
CCV:
114
Expiry date:
mm/dd/yyyy
I Agree
Activate Windows
Go to Settings to activate Windows.
Type here to search
10:15 PM
11/1/2020



6. Conclusion:

A secure, responsive, user friendly online organic food ordering website was created successfully using the technologies such as Html, Css, Javascript, Django, Angularjs where users can choose their favourite restaurants (ie. Stores) and order their organic foods quickly and cheaply which are good for their health. The website is also tested and certified by FDA(Food and Drug Administration) to guarantee that the foods are purely organic and pesticides-free.

LIST OF ABBREVATIONS:

FDA- Food and Drug Administration