FOOD ORDER SYSTEM WITH FEEDBACK

Project submitted to the

APSSDC

Bachelor of Technology

In

Computer Science and Engineering

Submitted By

KOMATI ANKITHA (24P35A0522)

YARRA SATYA KEERTHANA (23P31A05J8)

KADALI RAMYASRI (23P31A05G3)

YARRA BHAVANA LAKSHMI (23P31A05J7)

JAGARAPU TEJASRI (24P35A0518)

Under the guidance of

SRIKANYA TAMADA

TABLE OF CONTENTS

ABSTRACT

The Food Order System with Feedback is a web-based application designed to streamline the process of ordering food from restaurants and collecting customer feedback to enhance service quality. This system enables users to browse menus, place orders, and make online payments efficiently through a user-friendly interface.

Restaurants can manage their orders, update menu items, and monitor customer feedback in real time. The integrated feedback module allows customers to rate their food and service, submit reviews, and suggest improvements. This feedback is analysed to help restaurants maintain quality standards and improve customer satisfaction.

By combining digital ordering with responsive feedback collection, the system aims to create a seamless and interactive food ordering experience for both customers and restaurant owners.

The system consists of four main modules:

- User Module
- Admin Module
- Order Module
- Feedback Module

The primary objectives of this analysis are as follows:

- To develop a user-friendly platform for customers to easily browse restaurant menus, place food orders, and make payments online.

- To streamline the food ordering process for restaurants by providing a centralized system to receive, manage, and update orders in real time.

- To implement a feedback and rating system that allows customers to share their dining experiences, which helps restaurants improve their service quality.

- To enable real-time tracking of order status by customers, improving transparency and reducing customer service issues.

- To allow restaurant administrators to manage their menu items, pricing, availability, and view customer feedback through a dedicated admin panel.

- To maintain a secure and reliable database for storing customer profiles, order history, and feedback data, ensuring data integrity and privacy.

- To analyse feedback data for generating useful insights such as top-rated dishes, service improvement areas, and customer satisfaction levels.

- To support scalability and future enhancements such as promo codes, loyalty rewards, multi-language support, and integration with third-party delivery services.

INTRODUCTION

In recent years, the food industry has experienced a significant shift towards digital platforms, driven by the increasing demand for convenience, speed, and customer-centric services. Traditional food ordering methods, such as phone calls or walk-ins, or time-consuming and often lead to errors in communication or order handling. To address these challenges, modern food ordering systems are being developed to automate and simplify the enter process from menu browsing to delivery and customer feedback.

The Food Order System with Feedback is an innovative solution design to enhance the dining experience for customers while helping restaurants manage their operations efficiently. This system allows users to register, view restaurant menus, place customized orders, and make secure payments through an intuitive web or mobile interface. After receiving their orders, customers can also provide feedback on various aspects such as food quality, delivery time, hygiene, and overall satisfaction.

Restaurants, in turn, benefit from an organised platform that not only helps them process orders efficiently but also gives them access to valuable customer insights through the integrated feedback system. By analysing customer reviews and ratings, restaurants can identify strengths and weaknesses in their service, enabling them to take informed actions for improvement.

By integrating a responsive feedback mechanism, the Food Order System goes beyond just processing orders it contributes to building trust, enhancing service quality, and promoting long-term customer satisfaction.

PROJECT MODULES:

1. User Module:

- Register/Login/Logout
- View food menu
- Add to cart
- Place order
- Submit feedback

2. Admin Module:

- Login/Logout
- Add/Edit/Delete food items
- View orders
- Manage order statuses
- View customer feedback

3. Order Module:

- Cart management
- Checkout process
- Order history
- Order status tracking

4. Feedback Module:

- Feedback form(After order completion)
- Star rating and comments
- Admin review interface

TECHNOLOGY STACK USED

1. FRONTEND:

- HTML – For structuring web pages
- CSS – For styling and layout
- Bootstrap – For responsive design and faster development

2. BACKEND:

- Python(Django) – Robust backend framework with built-in admin panel

3. DATABASE:

- SQLite – Lightweight option for small projects or local development

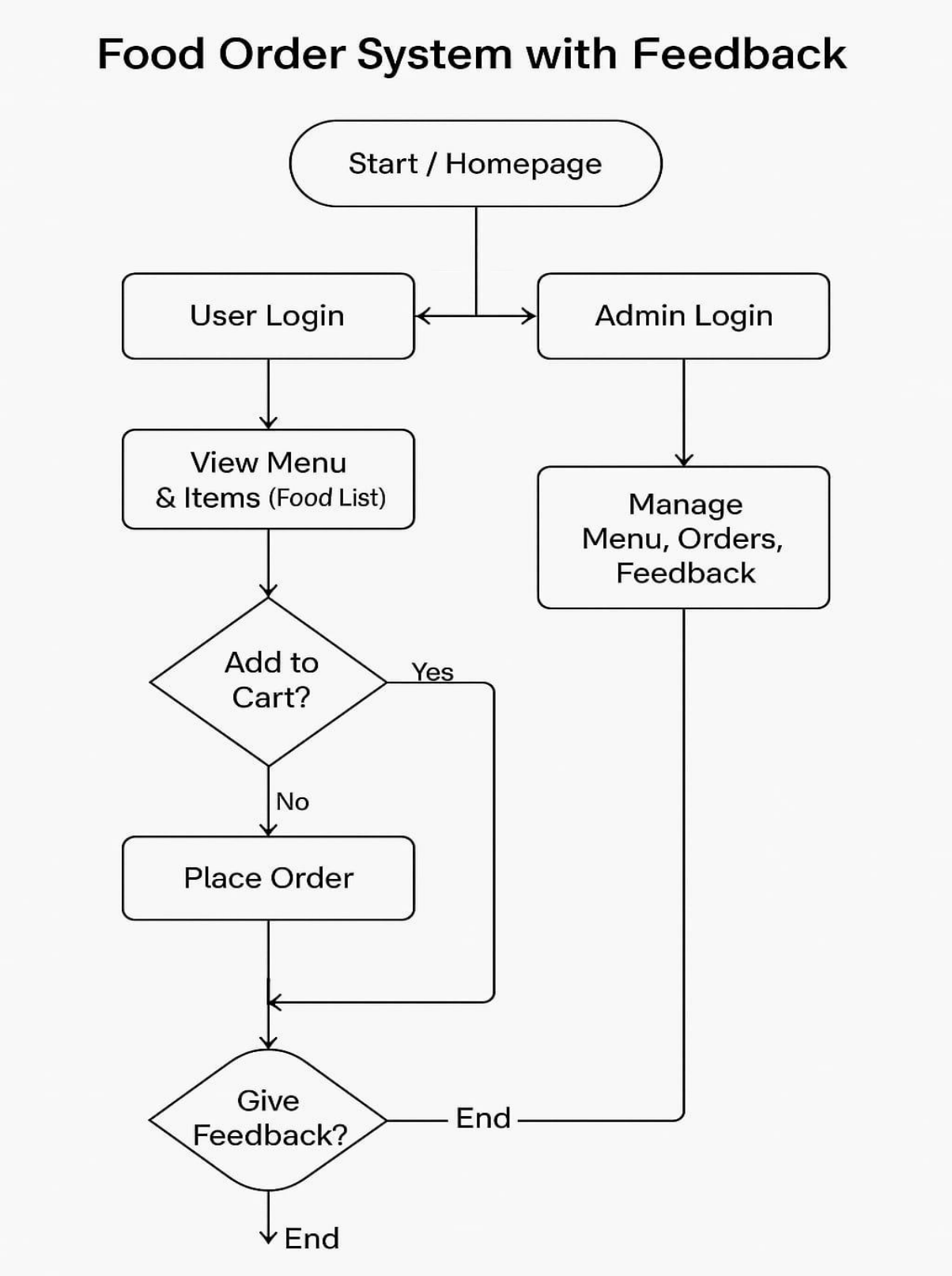
4. Authentication & Security:

- Django Authentication System(if using django)
- Django built-in password hashing - for storing passwords securely

5. Additional tools & Features:

- Admin dashboard
- Feedback analysis – Using simple data analytics

System design and architecture



System design layers:

1. Presentation Layer(Frontend):

- Acts as a interface between the user and the system.
- Allows customers to register, login, browse menus, place orders and give feedbacks.
- Technology : HTML, CSS, Bootstrap

2. Application Layer(Backend):

- Core business logic : Mnages authentication, order processing, and feedback connection.
- Responsible for vadidating user input, handling HTTP requests/responses, and accessing the database.
- Technologies : Django

3. Data Layer(Database):

- Stores persistent data : Users, orders, feedback, restaurant menus.
- Ensures data integrity and relational consistency.
- Technologies : SQLite

ADVANTAGES

1. User-Friendly Interface:

- Simple, intuitive design for easy food ordering and feedback submission.

2. Contactless Ordering:

- Reduces the need for physical interaction, ideal for modern hygiene standards.

3. Quick Order Placement:

- Speeds up the ordering process with a digital cart and one-click checkout.

4. 24/7 Availability:

- Customers can place orders at any time, unlike manual systems.

5. Accurate Order Processing:

- Minimizes human error in taking and processing orders.

6. Real-Time Order Updates:

- Customers can track their order status live(e.g., Pending, Preparing, Delivered).

7. Secure Login & Payment:

- User accounts and online payments are handled securely using standard protocols.

8. Feedback Improves Quality:

- Customer reviews help restaurants fix issues and improve food/service.

9. Admin Control Panel:

- Restaurant admins can manage menu, prices, feedback, and orders efficiently.

10. Digital Record Keeping:

- Automatic storage of order history, user data, and reviews – useful for reports and audits.

11. Scalable Architecture:

- Easily expandable to support new features like promo codes, delivery tracking, or mobile apps.

12. Promotes Customer Loyalty:

- Positive experiences and active feedback loops build customer trust and retention.

PROJECT CODE

Settings.py:

```
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'FoodTime',
]
```

Urls.py:

```
from django.contrib import admin
from django.urls import path
from FoodTime import views
from django.conf import settings
from django.conf.urls.static import static

urlpatterns = [
    path('admin/', admin.site.urls),
    path('index/',views.index,name="index"),
    path('home/',views.home,name="home"),
    path('food/',views.food,name="food"),
    path('order/<int:no>/', views.order, name='order'),
```

```
    path('delete/',views.delete,name="delete"),
    path('register/',views.register,name="register"),
    path('login/',views.login,name="login"),
    path('feed/',views.feed,name="feed"),
    path('review/',views.review,name="review")

]
```

Views.py:

```
from django.shortcuts import render,redirect
from django.http import HttpResponse
from .models import FoodItem,Registers,Feedback,Order
from django.contrib.auth import authenticate,login
from FoodProject import settings
# Create your views here.


def index(request):
        return render(request,'index.html')

def home(request):
        return render(request,'home.html')

def food(request):
    items = FoodItem.objects.all()
    return render(request, 'food.html', {'items': items})
```

```
def order(request,no):
    try:
        item = FoodItem.objects.get(id=no)
    except FoodItem.DoesNotExist:
        raise Http404("Food item does not exist")

    if request.method == 'POST':
        add=request.POST.get('add')
        od = Order(item=item.id, add=add)
        od.save()
        return render(request, 'success.html', {'item': item})

    return render(request, 'order.html', {'item': item})

def delete(request):
        return render(request,'delete.html')

def register(request):
        if request.method=="POST":
                fna=request.POST['fname']
                lna=request.POST['lname']
                una=request.POST['uname']
                mb=request.POST['mbl']
                e=request.POST['em']

        Registers.objects.create(first_name=fna,last_name=lna,username=una,mo
bile=mb,email=e)
                return redirect('home')
```

```
    return render(request,'register.html',{})

def login(request):
    if request.method=="POST":
        username=request.POST['uname']
        password=request.POST['pswd']
        u=authenticate(username=username,password=password)
        if u:
            return HttpResponse("<h2>Authenticated User!</h2>")
            return redirect('admin')
        else:
            return HttpResponse("<h2>Invalid User !</h2>")
    return render(request,'login.html',{})

def feed(request):
    if request.method == 'POST':
        customer_name = request.POST.get('customer_name')
        email = request.POST.get('email')
        food_rating = request.POST.get('food_rating')
        delivery_rating = request.POST.get('delivery_rating')
        comments = request.POST.get('comments')
        Feedback.objects.create(
     customer_name=customer_name,
     email=email,
     food_rating=food_rating,
     delivery_rating=delivery_rating,
     comments=comments)
```

```
            return redirect('home')
        return render(request,'feed.html')

def review(request):
        info=Feedback.objects.all()
        return render(request,"review.html",{'re':info})
```

Models.py:

```
from django.db import models
# Create your models here.
class FoodItem(models.Model):
    name = models.CharField(max_length=100)
    image = models.ImageField(upload_to='food_images/')
    price = models.DecimalField(max_digits=6, decimal_places=2)

    def __str__(self):
        return self.name

class Registers(models.Model):
        first_name=models.CharField(max_length=30)
        last_name=models.CharField(max_length=30)
        username=models.CharField(max_length=30)
        mobile=models.CharField(max_length=10)
        email=models.EmailField(max_length=30)

        def _str_(self):
            return self.name+" "+self.email
```

```
class Feedback(models.Model):
    customer_name = models.CharField(max_length=100, blank=True)
    email = models.EmailField(blank=True)
    food_rating = models.IntegerField()
    delivery_rating = models.IntegerField()
    comments = models.TextField(blank=True)
    submitted_at = models.DateTimeField(auto_now_add=True)

    def __str__(self):
        return f"Feedback {self.id} - {self.customer_name or 'Anonymous'}"


class Order(models.Model):
    item=models.IntegerField()
    add=models.TextField(blank=True)

    def __str__(self):
        return str(self.item)+" "+self.add
```

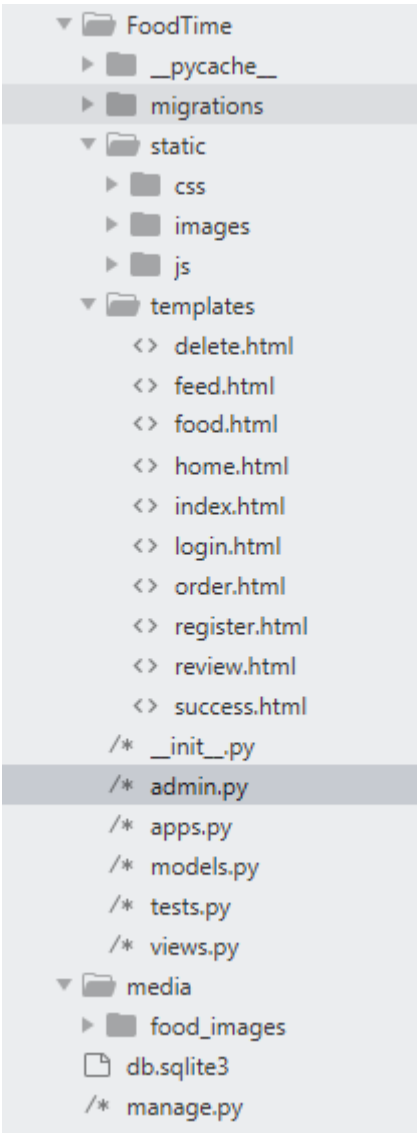
Apps.py:

```
from django.apps import AppConfig

class FoodtimeConfig(AppConfig):
    default_auto_field = 'django.db.models.BigAutoField'
    name = 'FoodTime'
```

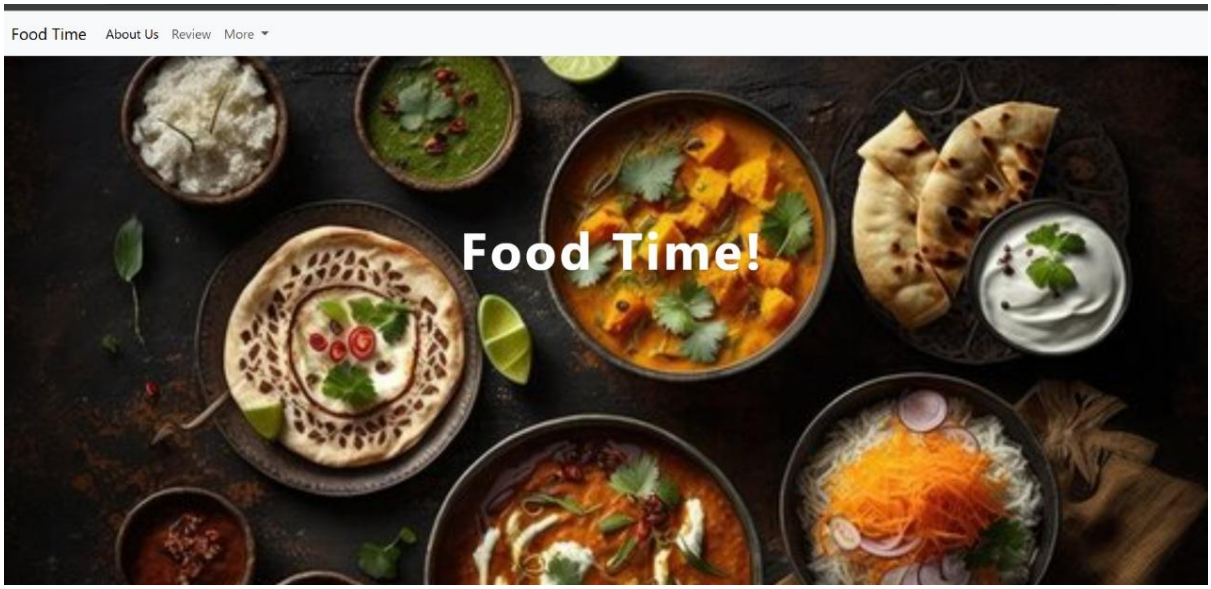
Admin.py:

```
from django.contrib import admin

# Register your models here.
from .models import FoodItem,Registers,Feedback,Order

admin.site.register(FoodItem)

admin.site.register(Registers)

admin.site.register(Feedback)

admin.site.register(Order)
```

FILE STRUCTURE:



OUTPUTS:


Food Time
About Us
Review
More
Food Time!


Home
Food Menu
Feedback

Registeration Form

First_name:

Enter Your firstname

Last_name:

Enter Your lastname

Username:

Enter Your Username

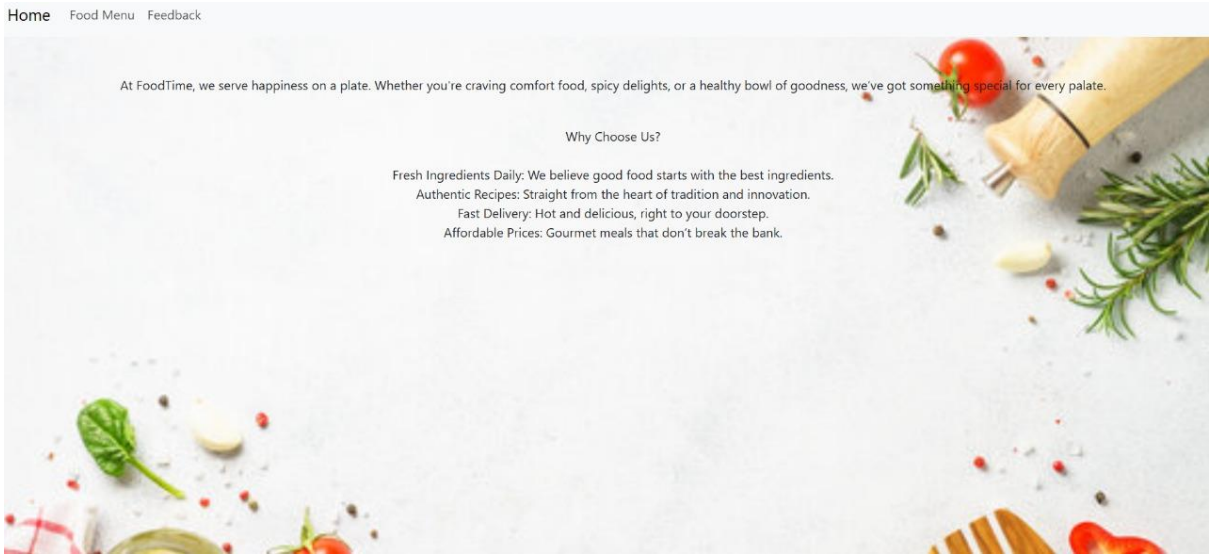
mobile:

Enter Your Mobile number

Email:

Enter Your Email

Register



Food Menu

Search for Fodd

Food Menu

Chicken Curry

₹100.00

Paneer curry

₹150.00

Tiger Rice

₹70.00

Chicken Biryani

₹150.00

Potato Fry

Mutton Curyy

Brinjal Curyy

Potato Curyy

Order: Chicken Curry

Price: ₹100.00

Enter Items

Enter no of items

Enter Address

Address.....

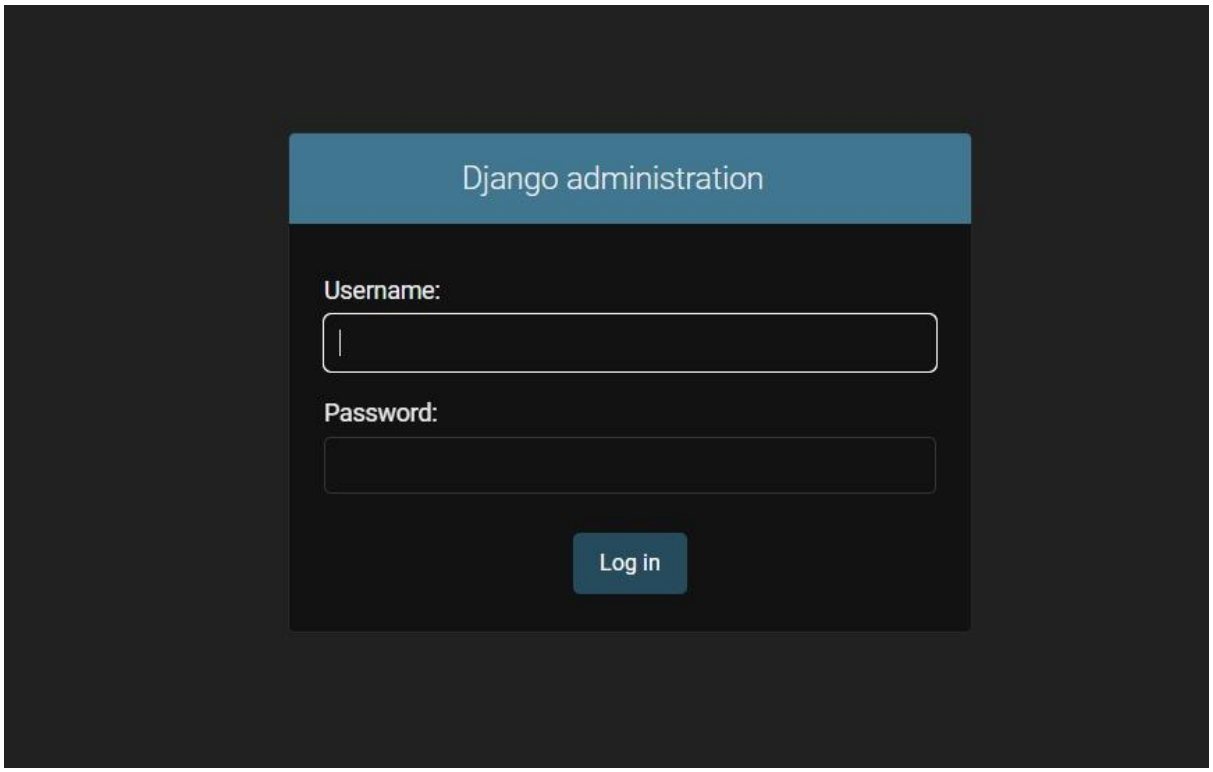
Cancel

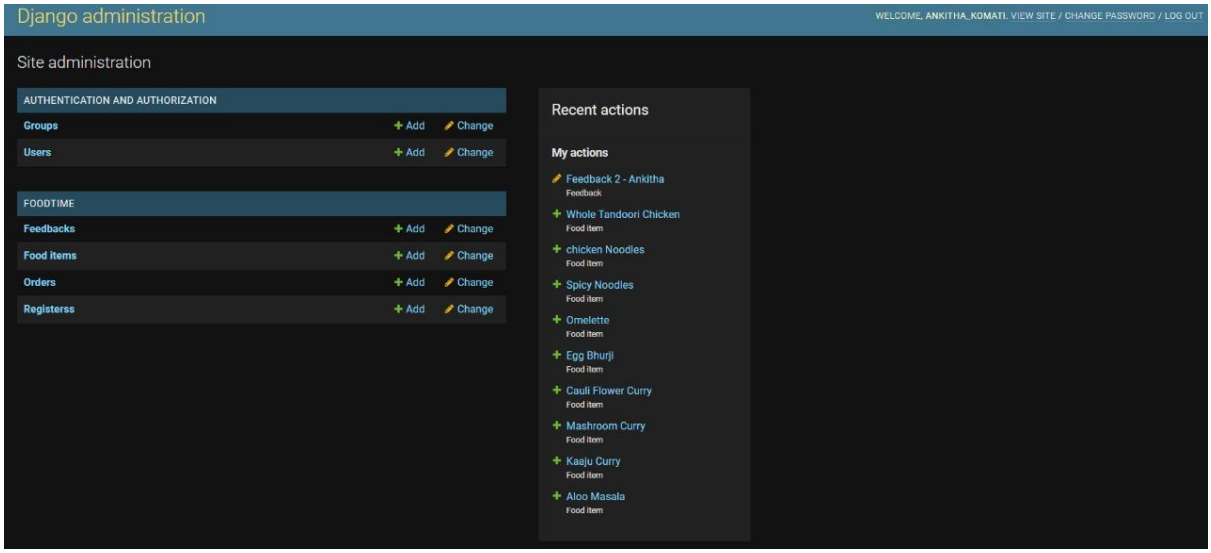
Confirm Order

Thank you for ordering Chicken Curry!

Enjoy Your Day

Back to menu




Django administration
WELCOME, ANKITHA_KOMATI. VIEW SITE / CHANGE PASSWORD / LOG OUT
Site administration
AUTHENTICATION AND AUTHORIZATION
Groups + Add Change
Users + Add Change
FOODTIME
Feedbacks + Add Change
Food items + Add Change
Orders + Add Change
Registerss + Add Change
Recent actions
My actions
Feedback 2 - Ankitha
Feedback
Whole Tandoori Chicken
Food item
chicken Noodles
Food item
Spicy Noodles
Food item
Omelette
Food item
Egg Bhurji
Food item
Cauli Flower Curry
Food item
Mashroom Curry
Food item
Kaaju Curry
Food item
Aloo Masala
Food item

CONCLUSION

The Food Order System with Feedback is a complete digital solution that addresses the limitations of traditional food ordering methods. In today's fast-paced world, where convenience, speed, and customer satisfaction are key priorities, this system provides an effective platform for both customers and restaurant administrators. By offering a user-friendly interface, secure order placements, and real-time communication, it bridges the gap between food service providers and their customers.

From the customers perspectives, the system brings great value. It eliminates the need for calling restaurants, waiting in lines, or worrying about incorrect orders. Customers can explore the digital menu, select their favourite dishes, and place orders with just a few clicks. The system also ensures transparency by allowing users to track their order status at every step – from confirmation to delivery. Most importantly, it empowers users to give genuine feedback through ratings and comments. This makes the user feel heard and involved in improving the service.

For restaurant owners or admins, the system simplifies day-to-day operations. Orders are received instantly and stored in an organized manner. Menu items can be updated anytime, and customer reviews can be viewed to understand service quality. Admins can easily monitor customer preferences, sales patterns, and the most popular dishes. This not only improves decision-making but also builds stronger customer trust through accountability and quality assurance.

The feedback feature is one of the most impactful parts of this system. It enables continuous improvement by collecting real-time responses. These insights help in identifying issues, improving food quality, refining service speed, and enhancing packaging and delivery. In the long run, feedback can shape customer loyalty and business growth.

From a technical point of view, the project uses a structured and scalable architecture. It can be extended in the future to include mobile apps, promotional offers, delivery tracking, and advanced analytics. It supports integration with payment gateways, notification services, and map APIs, making it suitable for real-world deployment.

In conclusion, the Food Order System with Feedback is more than just a digital ordering tool – it is a complete ecosystem that improves communication, reduces human error, increases satisfaction, and enhances the overall experience for all stakeholders involved. It reflects how technology can simplify life and elevate service quality in the food industry, and it holds great potential for expansion and real-time application in today’s digital era.