

Code for Backend development for the company Swiggy

Create folder:

```
>mkdir comp
```

```
>cd comp
```

install Django

```
>pip install Django
```

Create a project

```
django-admin startproject swiggy_project
```

```
cd swiggy_project
```

Startup a app(Swiggy)

```
python manage.py startapp swiggy_app
```

Install DRF

```
pip install djangorestframework
```

Install Postgresql

```
>brew install postgresql
```

Install psycopg2

```
pip install psycopg2
```

Under Swiggy_project the files are exist

▼ python

DLLs

> Doc

> include

> Lib

libs

New folder

> PyCharm 2023.3.3

Scripts

▼ swiggy_project

> .idea

▼ swiggy_app

migrations

template

▼ swiggy_project

__pycache__

1.Setting.py

```
DATABASES = {  
    'default': {  
        'ENGINE': 'django.db.backends.postgresql',  
        'NAME': 'swiggy_db',  
        'USER': 'your_postgres_user',  
        'PASSWORD': 'your_postgres_password',  
        'HOST': 'localhost',  
        'PORT': '5432',  
    }  
}
```

2.Url.py:

```
from django.contrib import admin  
from django.urls import path, include
```

```
urlpatterns = [  
    path('admin/', admin.site.urls),  
    path('api/', include('swiggy_app.urls')),  
]
```

Under swiggy_app:

1.Models.py:

```
from django.db import models  
from django.contrib.auth.models import User  
  
class Milestone(models.Model):  
    user = models.OneToOneField(User, on_delete=models.CASCADE)  
    points = models.IntegerField(default=0)  
  
class Subscription(models.Model):  
    user = models.OneToOneField(User, on_delete=models.CASCADE)  
    swiggy_plus = models.BooleanField(default=False)
```

2.Serializers.py:

```
from rest_framework import serializers  
from .models import Milestone, Subscription
```

```
class MilestoneSerializer(serializers.ModelSerializer):  
    class Meta:  
        model = Milestone  
        fields = ('user', 'points')
```

```
class SubscriptionSerializer(serializers.ModelSerializer):  
    class Meta:  
        model = Subscription  
        fields = ('user', 'swiggy_plus')
```

3.Views.py:

```
from django.shortcuts import render  
from django.contrib.auth.decorators import login_required  
from .models import Milestone, Subscription
```

@login_required

```
def milestone_view(request):  
    milestone = Milestone.objects.get_or_create(user=request.user)[0]  
    subscription = Subscription.objects.get_or_create(user=request.user)[0]  
    return render(request, 'index.html', {'user': request.user, 'milestone': milestone, 'subscription': subscription})
```

4.Url.py:

```
from django.urls import path
```

```
from .views import milestone_view
```

```
urlpatterns = [  
    path("", milestone_view, name='milestone'),  
]
```

5.Template:

□5.1 index.html:

```
□□<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
    <meta charset="UTF-8">
```

```
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
    <title>Swiggy Milestones</title>
```

```
</head>
```

```
<body>
```

```
    <h1>Welcome to Swiggy Milestones</h1>
```

```
    {% if user.is_authenticated %}
```

```
        <p>Hello, {{ user.username }}!</p>
```

```
        <p>Your Milestone Points: {{ milestone.points }}</p>
```

```
        <p>Swiggy Plus Subscription: {% if subscription.swiggy_plus %}Active{% else %}Inactive{% endif %}</p>
```

```
    {% else %}
```

```
        <p>Please log in to view your milestone points and subscription status.</p>
```

```
    {% endif %}
```

```
    <script>
```

```
        // You can add any JavaScript logic here if needed
```

```
    </script>
```

```
</body>
```

```
</html>
```

After saving the files

To make Migrations:

```
□>python manage.py makemigrations
```

To make Migrate:

```
□>python manage.py migrate
```

To Run server:

```
□python manage.py runserver
```

Open at the chrome and type the ip address which we get in the terminal and the output is:

Introducing Swiggy Milestone!

At Swiggy, we care about your loyalty. That's why we're excited to introduce Swiggy Milestone, our new gamification feature designed to enhance your experience and reward your loyalty.

Unlock Exciting Benefits:

- Free deliveries on eligible orders
- Exclusive discounts
- Premium customer support

Subscribe to Swiggy Plus:

For even more perks, consider subscribing to Swiggy Plus! With Swiggy Plus, you'll enjoy:

- Unlimited free deliveries
- Additional discounts and promotions
- Priority customer support

Don't miss out on the ultimate Swiggy experience. Upgrade to Swiggy Plus today and start enjoying the benefits!

Thank you for choosing Swiggy – where your satisfaction is our priority.