# Fast Food Project

1 - FastFood\_01>py manage.py runserver

2 - FastFood\_01>py manage.py startapp food

# Views

3 – Food => views.py

from django.http import HttpResponse

from django.shortcuts import render

def index(request):

return HttpResponse('Hello World!')

def item(request):

return HttpResponse('<h1>This is item view</h1>')

# URL Patterns

4 – food => urls.py

from django.urls import path

from . import views

urlpatterns = [

path('', views.index, name='index'),

path('item/', views.item, name='item'),

]

5 – FastFood\_01 => urls.py

from django.contrib import admin

from django.urls import path, include

urlpatterns = [

path('admin/', admin.site.urls),

path('food/', include('food.urls')),

]

6 – CHECK ----> <http://127.0.0.1:8000/food/> & <http://127.0.0.1:8000/food/item/>

# Models

7 - FastFood\_01>py manage.py migrate

8 – FastFood\_01 => settings.py

…

INSTALLED\_APPS = [

'django.contrib.admin',

'django.contrib.auth',

'django.contrib.contenttypes',

'django.contrib.sessions',

'django.contrib.messages',

'django.contrib.staticfiles',

'food.apps.FoodConfig',

]

…

9 – Food => models.py

class Item(models.Model):

item\_name = models.CharField(max\_length=200)

item\_desc = models.CharField(max\_length=200)

item\_price = models.IntegerField()

def \_\_str\_\_(self):

return self.item\_name

10 - FastFood\_01>py manage.py makemigrations food

11 - FastFood\_01>py manage.py migrate

# Using The Interactive Shell

12 - FastFood\_01>py manage.py shell

13 - >>> from food.models import Item

14 - >>> Item.objects.all()

<QuerySet []>

15 - >>> a = Item(item\_name="Pizza", item\_desc="Chessy Pizza", item\_price=20)

16 - >>> a.save()

17 - >>> a.id

1

18 - >>> a.pk

1

19 - >>> b = Item(item\_name="Burger", item\_desc="Chessy Burger", item\_price=10)

20 - >>> b.save()

21 - >>> Item.objects.all()

<QuerySet [<Item: Pizza>, <Item: Burger>]>

22 - >>> exit()

# Django Admin Panel & Creating Super User

23 - FastFood\_01>py manage.py createsuperuser

Username (leave blank to use 'ihc'): admin

Email address: admin@gmail.com

Password:123

Password (again):123

This password is too short. It must contain at least 8 characters.

This password is too common.

This password is entirely numeric.

Bypass password validation and create user anyway? [y/N]: y

Superuser created successfully.

24 - FastFood\_01>py manage.py runserver

25 – CHECK ----> <http://127.0.0.1:8000/admin/login/?next=/admin/>

26 – food => admin.py

from django.contrib import admin

from .models import Item

admin.site.register(Item)

27 – CHECK ----> <http://127.0.0.1:8000/admin/food/item/>

28 – ADD ITEM ----> Item name: Burrito Item desc: Fluffy burrito Item price: 30

# Reading Data From Database

29 – food => views.py

from django.http import HttpResponse

from django.shortcuts import render

from .models import Item

def index(request):

item\_list = Item.objects.all()

return HttpResponse(item\_list)

def item(request):

return HttpResponse('<h1>This is item view</h1>')

30 – CHECK ----> <http://127.0.0.1:8000/food/>

# Django Templates

31 – templates => food => index.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Food</title>

</head>

<body>

<h1>Food template</h1>

</body>

</html>

32 – food => views.py

from django.http import HttpResponse

from django.shortcuts import render

from .models import Item

from django.template import loader

def index(request):

item\_list = Item.objects.all()

templates = loader.get\_template('food/index.html')

context = {

}

return HttpResponse(templates.render(context, request))

def item(request):

return HttpResponse('<h1>This is item view</h1>')

# Passing Context To Templates

33 – food => views.py

from django.http import HttpResponse

from django.shortcuts import render

from .models import Item

def index(request):

item\_list = Item.objects.all()

context = {

'item\_list': item\_list,

}

return render(request, 'food/index.html', context)

def item(request):

return render('<h1>This is item view</h1>')

34 – templates => food => index.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Food</title>

<style>

table {

font-family: arial, sans-serif;

border-collapse: collapse;

width: 100%;

}

td, th {

border: 1px solid #dddddd;

text-align: left;

padding: 8px;

}

tr:nth-child(even) {

background-color: #dddddd;

}

</style>

</head>

<body>

<h1>Food template</h1>

<table>

<tr>

<th>ID</th>

<th>Item Name</th>

<th>Item Desc</th>

<th>Item Price</th>

</tr>

{% for item in item\_list %}

<tr>

<td>{{ item.id }}</td>

<td>{{ item.item\_name }}</td>

<td>{{ item.item\_desc }}</td>

<td>{{ item.item.price }}</td>

</tr>

{% endfor %}

</table>

</body>

</html>

# Creating The Detail View

35 – food => views.py

from django.http import HttpResponse

from django.shortcuts import render

from .models import Item

def index(request):

item\_list = Item.objects.all()

context = {

'item\_list': item\_list,

}

return render(request, 'food/index.html', context)

def item(request):

return render('<h1>This is item view</h1>')

def detail(request, item\_id):

return HttpResponse("This is item on/id: %s" % item\_id)

36 – food => urls.py

from django.urls import path

from . import views

urlpatterns = [

path('', views.index, name='index'),

path('item/', views.item, name='item'),

# /food/1

path('<int:item\_id>/', views.detail, name='detail'),

]

37 – CHECK ----> <http://127.0.0.1:8000/food/1/>

# Completing The Detail View

38 – food => views.py

from django.http import HttpResponse

from django.shortcuts import render

from .models import Item

def index(request):

item\_list = Item.objects.all()

context = {

'item\_list': item\_list,

}

return render(request, 'food/index.html', context)

def item(request):

return render('<h1>This is item view</h1>')

def detail(request, item\_id):

item= Item.objects.get(pk=item\_id)

context = {

'item': item,

}

return render(request, 'food/detail.html', context)

39 – food => urls.py

from django.urls import path

from . import views

app\_name = 'food'

urlpatterns = [

path('', views.index, name='index'),

path('item/', views.item, name='item'),

# /food/1

path('<int:item\_id>/', views.detail, name='detail'),

]

40 – templates => food => index.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Details</title>

<style>

table {

font-family: arial, sans-serif;

border-collapse: collapse;

width: 100%;

}

td, th {

border: 1px solid #dddddd;

text-align: left;

padding: 8px;

}

tr:nth-child(even) {

background-color: #dddddd;

}

</style>

</head>

<body>

<h1>Food Details</h1>

<table>

<tr>

<th>ID</th>

<th>Item Name</th>

<th>Item Desc</th>

<th>Item Price</th>

</tr>

{% for item in item\_list %}

<tr>

<td><a href="{% url 'food:detail' item.id %}">{{ item.id }}</a></td>

<td>{{ item.item\_name }}</td>

<td>{{ item.item\_desc }}</td>

<td>{{ item.item\_price }}</td>

</tr>

{% endfor %}

</table>

</body>

</html>

41 – templates => food => detail.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Details</title>

</head>

<body>

<h1>Food Details</h1>

<h2>Item Name: {{ item.item\_name }}</h2>

<h3>Item Desc: {{ item.item\_desc }}</h3>

<h4>Item Price: {{ item.item\_price }}$</h4>

<h5><a href="{% url 'food:index' %}">Index</a></h5>

</body>

</html>

# Static Files

42 – FastFood\_01 => settings.py

…

STATIC\_URL = '/static/'

STATICFILES\_DIRS = [

BASE\_DIR / "static",

]

…

43 – templates => food => index.html

{% load static %}

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Details</title>

<link rel="stylesheet" href="{% static 'food/style.css' %}">

</head>

<body>

<h1>Food Details</h1>

<table>

<tr>

<th>ID</th>

<th>Item Name</th>

<th>Item Desc</th>

<th>Item Price</th>

</tr>

{% for item in item\_list %}

<tr>

<td><a href="{% url 'food:detail' item.id %}">{{ item.id }}</a></td>

<td>{{ item.item\_name }}</td>

<td>{{ item.item\_desc }}</td>

<td>{{ item.item\_price }}</td>

</tr>

{% endfor %}

</table>

</body>

</html>

44 – static => food => style.css

table {

font-family: arial, sans-serif;

border-collapse: collapse;

width: 100%;

}

td, th {

border: 1px solid #dddddd;

text-align: left;

padding: 8px;

}

tr:nth-child(even) {

background-color: #dddddd;

}

# Bootstrap Navbar

45 – templates => food => index.html

{% load static %}

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Details</title>

<!-- Bootstrap CSS -->

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC" crossorigin="anonymous">

<link rel="stylesheet" href="{% static 'food/style.css' %}">

</head>

<body>

<nav class="navbar navbar-expand-lg navbar-dark bg-dark">

<div class="container-fluid">

<a class="navbar-brand" href="#">Navbar</a>

<button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse" id="navbarSupportedContent">

<ul class="navbar-nav me-auto mb-2 mb-lg-0">

<li class="nav-item">

<a class="nav-link active" aria-current="page" href="#">Home</a>

</li>

<li class="nav-item">

<a class="nav-link" href="#">Add Item</a>

</li>

<li class="nav-item">

<a class="nav-link" href="#">Delete Item</a>

</li>

<li class="nav-item">

<a class="nav-link" href="#">View Item</a>

</li>

<li class="nav-item dropdown">

<a class="nav-link dropdown-toggle" href="#" id="navbarDropdown" role="button" data-bs-toggle="dropdown" aria-expanded="false">

Dropdown

</a>

<ul class="dropdown-menu" aria-labelledby="navbarDropdown">

<li><a class="dropdown-item" href="#">Action</a></li>

<li><a class="dropdown-item" href="#">Another action</a></li>

<li><hr class="dropdown-divider"></li>

<li><a class="dropdown-item" href="#">Something else here</a></li>

</ul>

</li>

</ul>

<form class="d-flex">

<input class="form-control me-2" type="search" placeholder="Search" aria-label="Search">

<button class="btn btn-outline-success" type="submit">Search</button>

</form>

</div>

</div>

</nav>

<h1>Food Details</h1>

<table>

<tr>

<th>ID</th>

<th>Item Name</th>

<th>Item Desc</th>

<th>Item Price</th>

</tr>

{% for item in item\_list %}

<tr>

<td><a href="{% url 'food:detail' item.id %}">{{ item.id }}</a></td>

<td>{{ item.item\_name }}</td>

<td>{{ item.item\_desc }}</td>

<td>{{ item.item\_price }}</td>

</tr>

{% endfor %}

</table>

<!-- Optional JavaScript; choose one of the two! -->

<!-- Option 1: Bootstrap Bundle with Popper -->

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js" integrity="sha384-MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM" crossorigin="anonymous"></script>

<!-- Option 2: Separate Popper and Bootstrap JS -->

<!--

<script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.9.2/dist/umd/popper.min.js" integrity="sha384-IQsoLXl5PILFhosVNubq5LC7Qb9DXgDA9i+tQ8Zj3iwWAwPtgFTxbJ8NT4GN1R8p" crossorigin="anonymous"></script>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.min.js" integrity="sha384-cVKIPhGWiC2Al4u+LWgxfKTRIcfu0JTxR+EQDz/bgldoEyl4H0zUF0QKbrJ0EcQF" crossorigin="anonymous"></script>

-->

</body>

</html>

# Creating Base Template

46 – templates => base.html

{% load static %}

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>

{% block title %}

{% endblock %}

</title>

<!-- Bootstrap CSS -->

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC" crossorigin="anonymous">

<link rel="stylesheet" href="{% static 'food/style.css' %}">

</head>

<body>

<nav class="navbar navbar-expand-lg navbar-dark bg-dark">

<div class="container-fluid">

<a class="navbar-brand" href="#">Navbar</a>

<button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse" id="navbarSupportedContent">

<ul class="navbar-nav me-auto mb-2 mb-lg-0">

<li class="nav-item">

<a class="nav-link active" aria-current="page" href="#">Home</a>

</li>

<li class="nav-item">

<a class="nav-link" href="#">Add Item</a>

</li>

<li class="nav-item">

<a class="nav-link" href="#">Delete Item</a>

</li>

<li class="nav-item">

<a class="nav-link" href="#">View Item</a>

</li>

<li class="nav-item dropdown">

<a class="nav-link dropdown-toggle" href="#" id="navbarDropdown" role="button" data-bs-toggle="dropdown" aria-expanded="false">

Dropdown

</a>

<ul class="dropdown-menu" aria-labelledby="navbarDropdown">

<li><a class="dropdown-item" href="#">Action</a></li>

<li><a class="dropdown-item" href="#">Another action</a></li>

<li><hr class="dropdown-divider"></li>

<li><a class="dropdown-item" href="#">Something else here</a></li>

</ul>

</li>

</ul>

<form class="d-flex">

<input class="form-control me-2" type="search" placeholder="Search" aria-label="Search">

<button class="btn btn-outline-success" type="submit">Search</button>

</form>

</div>

</div>

</nav>

{% block body %}

{% endblock %}

<!-- Optional JavaScript; choose one of the two! -->

<!-- Option 1: Bootstrap Bundle with Popper -->

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js" integrity="sha384-MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM" crossorigin="anonymous"></script>

<!-- Option 2: Separate Popper and Bootstrap JS -->

<!--

<script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.9.2/dist/umd/popper.min.js" integrity="sha384-IQsoLXl5PILFhosVNubq5LC7Qb9DXgDA9i+tQ8Zj3iwWAwPtgFTxbJ8NT4GN1R8p" crossorigin="anonymous"></script>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.min.js" integrity="sha384-cVKIPhGWiC2Al4u+LWgxfKTRIcfu0JTxR+EQDz/bgldoEyl4H0zUF0QKbrJ0EcQF" crossorigin="anonymous"></script>

-->

</body>

</html>

47 – templates => food => index.html

{% extends "base.html" %}

{% load static %}

{% block title %} Index {% endblock %}

{% block body %}

<h1>Food Details</h1>

<table>

<tr>

<th>ID</th>

<th>Item Name</th>

<th>Item Desc</th>

<th>Item Price</th>

</tr>

{% for item in item\_list %}

<tr>

<td><a href="{% url 'food:detail' item.id %}">{{ item.id }}</a></td>

<td>{{ item.item\_name }}</td>

<td>{{ item.item\_desc }}</td>

<td>{{ item.item\_price }}</td>

</tr>

{% endfor %}

</table>

{% endblock %}

48 – templates => food => detail.html

{% extends "base.html" %}

{% load static %}

{% block title %} Index {% endblock %}

{% block body %}

<h1>Food Details</h1>

<table>

<tr>

<th>ID</th>

<th>Item Name</th>

<th>Item Desc</th>

<th>Item Price</th>

</tr>

{% for item in item\_list %}

<tr>

<td><a href="{% url 'food:detail' item.id %}">{{ item.id }}</a></td>

<td>{{ item.item\_name }}</td>

<td>{{ item.item\_desc }}</td>

<td>{{ item.item\_price }}</td>

</tr>

{% endfor %}

</table>

{% endblock %}

# Adding Image To Model

49 – food => models.py

from django.db import models

class Item(models.Model):

item\_name = models.CharField(max\_length=200)

item\_desc = models.CharField(max\_length=200)

item\_price = models.IntegerField()

item\_image = models.CharField(max\_length=500, default="https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcR-\_IXHpBmStVsAnXqctN\_3u\_ZIGLToyS-VjSWonFQRFYObK9vq0DuCscC\_LiJvzfHSdFY&usqp=CAU")

def \_\_str\_\_(self):

return self.item\_name

50 - FastFood\_01>py manage.py makemigrations

51 - FastFood\_01>py manage.py migrate

52 – templates => food => index.html

{% extends "base.html" %}

{% load static %}

{% block title %} Index {% endblock %}

{% block body %}

<div class="p-3 mb-2 bg-success text-white text-center"><h1>Menu</h1></div>

<div class="container">

<div class="row">

<table>

<tr>

<th>ID</th>

<th>Item Image</th>

<th>Item Name</th>

<th>Item Desc</th>

<th>Item Price</th>

<th>Detail</th>

</tr>

{% for item in item\_list %}

<tr>

<td>{{ item.id }}</td>

<td><img class="card" height="50px" width="50px" src="{{ item.item\_image }}"></td>

<td>{{ item.item\_name }}</td>

<td>{{ item.item\_desc }}</td>

<td>{{ item.item\_price }}</td>

<td><a class="btn btn-success" href="{% url 'food:detail' item.id %}">Detail</a></td>

</tr>

{% endfor %}

</table>

</div>

</div>

{% endblock %}

53 – templates => food => detail.html

{% extends "base.html" %}

{% load static %}

{% block title %} Detail {% endblock %}

{% block body %}

<div class="container">

<div class="row">

<div class="col-md-12">

<div class="p-3 mb-2 bg-success text-white text-center"><h1>Food Details</h1></div>

<div class="card">

<img class="card-img-top" src="{{ item.item\_image }}" alt="{{ item.item\_name }}">

<div class="card-body">

<h1 class="card-title">{{ item.item\_name }}</h1>

<p class="card-text">{{ item.item\_desc }}</p>

<h3 class="card-title">{{ item.item\_price }}$</h3>

<h5><a class="btn btn-success" href="{% url 'food:index' %}">Index</a></h5>

</div>

</div>

</div>

</div>

</div>

{% endblock %}

# Adding Form To Add Items

54 – food => views.py

from django.http import HttpResponse

from django.shortcuts import render, redirect

from .forms import ItemForm

from .models import Item

def index(request):

item\_list = Item.objects.all()

context = {

'item\_list': item\_list,

}

return render(request, 'food/index.html', context)

def item(request):

return render('<h1>This is item view</h1>')

def detail(request, item\_id):

item= Item.objects.get(pk=item\_id)

context = {

'item': item,

}

return render(request, 'food/detail.html', context)

def create\_item(request):

form = ItemForm(request.POST or None)

if form.is\_valid():

form.save()

return redirect('food:index')

return render(request, 'food/item-form.html', {'form': form})

55 – food => forms.py

from django import forms

from .models import Item

class ItemForm(forms.ModelForm):

class Meta:

model = Item

fields = ['item\_name', 'item\_desc', 'item\_price', 'item\_image']

56 – FastFood\_01 => urls.py

from django.urls import path

from . import views

app\_name = 'food'

urlpatterns = [

path('', views.index, name='index'),

path('item/', views.item, name='item'),

# /food/1

path('<int:item\_id>/', views.detail, name='detail'),

# Add Item

path('add/', views.create\_item, name='create\_item'),

]

57 – templates => food => item-form.html

<form method="post">

{% csrf\_token %}

{{ form }}

<button type="submit">Save</button>

</form>

58 – CHECK ----> http://127.0.0.1:8000/food/add/

# Adding Base Template To Form

59 – templates => food => base.html

{% load static %}

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>

{% block title %}

{% endblock %}

</title>

<!-- Bootstrap CSS -->

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC" crossorigin="anonymous">

<link rel="stylesheet" href="{% static 'food/style.css' %}">

</head>

<body>

<nav class="navbar navbar-expand-lg navbar-dark bg-dark">

<div class="container-fluid">

<a class="navbar-brand" href="#">Navbar</a>

<button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse" id="navbarSupportedContent">

<ul class="navbar-nav me-auto mb-2 mb-lg-0">

<li class="nav-item">

<a class="nav-link active" aria-current="page" href="#">Home</a>

</li>

<li class="nav-item">

<a class="nav-link" href="{% url 'food:create\_item' %}">Add Item</a>

</li>

<li class="nav-item">

<a class="nav-link" href="#">Delete Item</a>

</li>

<li class="nav-item">

<a class="nav-link" href="#">View Item</a>

</li>

<li class="nav-item dropdown">

<a class="nav-link dropdown-toggle" href="#" id="navbarDropdown" role="button" data-bs-toggle="dropdown" aria-expanded="false">

Dropdown

</a>

<ul class="dropdown-menu" aria-labelledby="navbarDropdown">

<li><a class="dropdown-item" href="#">Action</a></li>

<li><a class="dropdown-item" href="#">Another action</a></li>

<li><hr class="dropdown-divider"></li>

<li><a class="dropdown-item" href="#">Something else here</a></li>

</ul>

</li>

</ul>

<form class="d-flex">

<input class="form-control me-2" type="search" placeholder="Search" aria-label="Search">

<button class="btn btn-outline-success" type="submit">Search</button>

</form>

</div>

</div>

</nav>

{% block body %}

{% endblock %}

<!-- Optional JavaScript; choose one of the two! -->

<!-- Option 1: Bootstrap Bundle with Popper -->

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js" integrity="sha384-MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM" crossorigin="anonymous"></script>

<!-- Option 2: Separate Popper and Bootstrap JS -->

<!--

<script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.9.2/dist/umd/popper.min.js" integrity="sha384-IQsoLXl5PILFhosVNubq5LC7Qb9DXgDA9i+tQ8Zj3iwWAwPtgFTxbJ8NT4GN1R8p" crossorigin="anonymous"></script>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.min.js" integrity="sha384-cVKIPhGWiC2Al4u+LWgxfKTRIcfu0JTxR+EQDz/bgldoEyl4H0zUF0QKbrJ0EcQF" crossorigin="anonymous"></script>

-->

</body>

</html>

60 – templates => food => item-form.html

{% extends "base.html" %}

{% block body %}

<div class="p-3 mb-2 bg-success text-white text-center"><h1>Add Item</h1></div>

<div class="container">

<div class="row">

<form method="post">

{% csrf\_token %}

{{ form }}

<button class="btn btn-success" type="submit">Save</button>

</form>

</div>

</div>

{% endblock %}

# Implementing The Edit Functionality

61 – food => views.py

from django.http import HttpResponse

from django.shortcuts import render, redirect

from .forms import ItemForm

from .models import Item

def index(request):

item\_list = Item.objects.all()

context = {

'item\_list': item\_list,

}

return render(request, 'food/index.html', context)

def item(request):

return render('<h1>This is item view</h1>')

def detail(request, item\_id):

item= Item.objects.get(pk=item\_id)

context = {

'item': item,

}

return render(request, 'food/detail.html', context)

def create\_item(request):

form = ItemForm(request.POST or None)

if form.is\_valid():

form.save()

return redirect('food:index')

return render(request, 'food/item-form.html', {'form': form})

def update\_item(request, id):

item = Item.objects.get(id=id)

form = ItemForm(request.POST or None, instance=item)

if form.is\_valid():

form.save()

return redirect('food:index')

return render(request, 'food/item-form.html', {'form': form, 'item': item})

62 – food => urls.py

from django.urls import path

from . import views

app\_name = 'food'

urlpatterns = [

path('', views.index, name='index'),

path('item/', views.item, name='item'),

# /food/1

path('<int:item\_id>/', views.detail, name='detail'),

# Add Item

path('add/', views.create\_item, name='create\_item'),

# edit

path('update/<int:id>/', views.update\_item, name='update\_item'),

]

# Delete Item

63 – food => views.py

from django.http import HttpResponse

from django.shortcuts import render, redirect

from .forms import ItemForm

from .models import Item

def index(request):

item\_list = Item.objects.all()

context = {

'item\_list': item\_list,

}

return render(request, 'food/index.html', context)

def item(request):

return render('<h1>This is item view</h1>')

def detail(request, item\_id):

item= Item.objects.get(pk=item\_id)

context = {

'item': item,

}

return render(request, 'food/detail.html', context)

def create\_item(request):

form = ItemForm(request.POST or None)

if form.is\_valid():

form.save()

return redirect('food:index')

return render(request, 'food/item-form.html', {'form': form})

def update\_item(request, id):

item = Item.objects.get(id=id)

form = ItemForm(request.POST or None, instance=item)

if form.is\_valid():

form.save()

return redirect('food:index')

return render(request, 'food/item-form.html', {'form': form, 'item': item})

def delete\_item(request, id):

item = Item.objects.get(id=id)

if request.method == 'POST':

item.delete()

return redirect('food:index')

return render(request, 'food/item-delete.html', {'item': item})

64 – food => urls.py

from django.urls import path

from . import views

app\_name = 'food'

urlpatterns = [

path('', views.index, name='index'),

path('item/', views.item, name='item'),

# /food/1

path('<int:item\_id>/', views.detail, name='detail'),

# Add Item

path('add/', views.create\_item, name='create\_item'),

# edit

path('update/<int:id>/', views.update\_item, name='update\_item'),

# delete

path('delete/<int:id>/', views.delete\_item, name='delete\_item'),

]

65 – templates => food => detail.html

{% extends "base.html" %}

{% load static %}

{% block title %} Detail {% endblock %}

{% block body %}

<div class="container">

<div class="row">

<div class="col-md-12">

<div class="p-3 mb-2 bg-success text-white text-center"><h1>Food Details</h1></div>

<div class="card">

<img class="card-img-top" src="{{ item.item\_image }}" alt="{{ item.item\_name }}">

<div class="card-body">

<h1 class="card-title">{{ item.item\_name }}</h1>

<p class="card-text">{{ item.item\_desc }}</p>

<h3 class="card-title">{{ item.item\_price }}$</h3>

<a class="btn btn-success" href="{% url 'food:index' %}">Index</a>

<a class="btn btn-warning" href="{% url 'food:delete\_item' item.id %}">Delete</a>

</div>

</div>

</div>

</div>

</div>

{% endblock %}

66 – templates => food => item-delete.html

{% extends "base.html" %}

{% block body %}

<div class="p-3 mb-2 bg-success text-white text-center"><h1>Delete Item</h1></div>

<div class="container">

<div class="row">

<form method="post">

{% csrf\_token %}

<h2>Are you sure you want to delete {{ item.item\_name }}</h2>

<button class="btn btn-danger" type="submit">Confirm</button>

<a class="btn btn-secondary" href="{% url 'food:index' %}" type="button">Cancel</a>

</form>

</div>

</div>

{% endblock %}

# Creating User Registration Form

67 - FastFood\_01>django-admin startapp users

68 – FastFood\_01 => settings.py

…

'users.apps.UsersConfig',

…

69 – templates => users => register.html

{% extends "base.html" %}

{% block title %} SignUp {% endblock %}

{% block body %}

<div class="p-3 mb-2 bg-success text-white text-center"><h1>SignUp</h1></div>

<div class="container">

<div class="row">

<form method="post">

{% csrf\_token %}

{{ form }}

<button class="btn btn-success" type="submit">SignUp</button>

</form>

</div>

</div>

{% endblock %}

70 – FastFood\_01 => urls.py

from django.contrib import admin

from django.urls import path, include

from users import views as user\_views

urlpatterns = [

path('admin/', admin.site.urls),

path('food/', include('food.urls')),

path('register/', user\_views.register, name='register'),

]

71 - FastFood\_01>py manage.py runserver

72 – CHECK ----> <http://127.0.0.1:8000/register/>

# Registration Success Message

73 – users => views.py

from django.shortcuts import render, redirect

from django.contrib.auth.forms import UserCreationForm

from django.contrib import messages

def register(request):

if request.method == 'POST':

form = UserCreationForm(request.POST)

if form.is\_valid():

form.save()

username = form.cleaned\_data.get('username')

messages.success(request, f'Welcome {username}, your account is created!')

return redirect('food:index')

else:

form = UserCreationForm()

return render(request, 'users/register.html', {'form': form})

74 – templates => users => register.html

{% extends "base.html" %}

{% block title %} SignUp {% endblock %}

{% block body %}

<div class="p-3 mb-2 bg-success text-white text-center"><h1>SignUp</h1></div>

<div class="container">

<div class="row">

<form method="post">

{% csrf\_token %}

{{ form }}

<button class="btn btn-success" type="submit">SignUp</button>

</form>

</div>

</div>

{% endblock %}

75 – templates => base.html

{% load static %}

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>

{% block title %}

{% endblock %}

</title>

<!-- Bootstrap CSS -->

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC" crossorigin="anonymous">

<link rel="stylesheet" href="{% static 'food/style.css' %}">

</head>

<body>

<nav class="navbar navbar-expand-lg navbar-dark bg-dark">

<div class="container-fluid">

<a class="navbar-brand" href="#">Navbar</a>

<button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse" id="navbarSupportedContent">

<ul class="navbar-nav me-auto mb-2 mb-lg-0">

<li class="nav-item">

<a class="nav-link active" aria-current="page" href="{% url 'food:index' %}">Home</a>

</li>

<li class="nav-item">

<a class="nav-link" href="{% url 'food:create\_item' %}">Add Item</a>

</li>

<li class="nav-item">

<a class="nav-link" href="#">Resister</a>

</li>

<li class="nav-item dropdown">

<a class="nav-link dropdown-toggle" href="#" id="navbarDropdown" role="button" data-bs-toggle="dropdown" aria-expanded="false">

Dropdown

</a>

<ul class="dropdown-menu" aria-labelledby="navbarDropdown">

<li><a class="dropdown-item" href="#">Action</a></li>

<li><a class="dropdown-item" href="#">Another action</a></li>

<li><hr class="dropdown-divider"></li>

<li><a class="dropdown-item" href="#">Something else here</a></li>

</ul>

</li>

</ul>

<form class="d-flex">

<input class="form-control me-2" type="search" placeholder="Search" aria-label="Search">

<button class="btn btn-outline-success" type="submit">Search</button>

</form>

</div>

</div>

</nav>

{% if messages %}

{% for message in messages %}

<div class="alert alert-{{ message.tag }}">

<div class="p-3 mb-2 bg-info text-white text-center">

{{ message }}

</div>

</div>

{% endfor %}

{% endif %}

{% block body %}

{% endblock %}

<!-- Optional JavaScript; choose one of the two! -->

<!-- Option 1: Bootstrap Bundle with Popper -->

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js" integrity="sha384-MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM" crossorigin="anonymous"></script>

<!-- Option 2: Separate Popper and Bootstrap JS -->

<!--

<script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.9.2/dist/umd/popper.min.js" integrity="sha384-IQsoLXl5PILFhosVNubq5LC7Qb9DXgDA9i+tQ8Zj3iwWAwPtgFTxbJ8NT4GN1R8p" crossorigin="anonymous"></script>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.min.js" integrity="sha384-cVKIPhGWiC2Al4u+LWgxfKTRIcfu0JTxR+EQDz/bgldoEyl4H0zUF0QKbrJ0EcQF" crossorigin="anonymous"></script>

-->

</body>

</html>

# Adding Additional Field

76 – users => forms.py

from django import forms

from django.contrib.auth.models import User

from django.contrib.auth.forms import UserCreationForm

class RegisterForm(UserCreationForm):

email = forms.EmailField()

class Meta:

model = User

fields = ['username', 'email', 'password1', 'password2']

77 – users => views.py

from django.shortcuts import render, redirect

from django.contrib.auth.forms import UserCreationForm

from django.contrib import messages

from .forms import RegisterForm

def register(request):

if request.method == 'POST':

form = RegisterForm(request.POST)

if form.is\_valid():

form.save()

username = form.cleaned\_data.get('username')

messages.success(request, f'Welcome {username}, your account is created!')

return redirect('food:index')

else:

form = RegisterForm()

return render(request, 'users/register.html', {'form': form})

# Logging In Users

78 - FastFood\_01 => urls.py

from django.contrib import admin

from django.urls import path, include

from users import views as user\_views

from django.contrib.auth import views as authentication\_views

urlpatterns = [

path('admin/', admin.site.urls),

path('food/', include('food.urls')),

path('register/', user\_views.register, name='register'),

path('login/', authentication\_views.LoginView.as\_view(template\_name='users/login.html'), name='login'),

path('logout/', authentication\_views.LogoutView.as\_view(template\_name='users/logout.html'), name='logout'),

]

79 – templates => users => login.html

{% extends "base.html" %}

{% block title %} Login {% endblock %}

{% block body %}

<div class="p-3 mb-2 bg-success text-white text-center"><h1>Login</h1></div>

<div class="container">

<div class="row">

<form method="post">

{% csrf\_token %}

{{ form }}

<button class="btn btn-success" type="submit">Login</button>

</form>

</div>

</div>

{% endblock %}

80 – FastFood\_01 => settings.py

…

LOGIN\_REDIRECT\_URL = 'food:index'

…

81 – templates => users => logout.html

{% extends "base.html" %}

{% block title %} Logout {% endblock %}

{% block body %}

<div class="p-3 mb-2 bg-success text-white text-center"><h1>You have been Logged out</h1></div>

{% endblock %}

# Adding Login Option To Navbar

82 – templates => base.html

{% load static %}

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>

{% block title %}

{% endblock %}

</title>

<!-- Bootstrap CSS -->

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC" crossorigin="anonymous">

<link rel="stylesheet" href="{% static 'food/style.css' %}">

</head>

<body>

<nav class="navbar navbar-expand-lg navbar-dark bg-dark">

<div class="container-fluid">

<a class="navbar-brand" href="#">{{ user.username }}</a>

<button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse" id="navbarSupportedContent">

<ul class="navbar-nav me-auto mb-2 mb-lg-0">

<li class="nav-item">

<a class="nav-link active" aria-current="page" href="{% url 'food:index' %}">Home</a>

</li>

{% if user.is\_authenticated %}

<li class="nav-item">

<a class="nav-link" href="{% url 'logout' %}">Logout</a>

</li>

<li class="nav-item">

<a class="nav-link" href="{% url 'food:create\_item' %}">Add Item</a>

</li>

{% else %}

<li class="nav-item">

<a class="nav-link" href="{% url 'register' %}">Resister</a>

</li>

<li class="nav-item">

<a class="nav-link" href="{% url 'login' %}">Login</a>

</li>

{% endif %}

<li class="nav-item dropdown">

<a class="nav-link dropdown-toggle" href="#" id="navbarDropdown" role="button" data-bs-toggle="dropdown" aria-expanded="false">

Dropdown

</a>

<ul class="dropdown-menu" aria-labelledby="navbarDropdown">

<li><a class="dropdown-item" href="#">Action</a></li>

<li><a class="dropdown-item" href="#">Another action</a></li>

<li><hr class="dropdown-divider"></li>

<li><a class="dropdown-item" href="#">Something else here</a></li>

</ul>

</li>

</ul>

<form class="d-flex">

<input class="form-control me-2" type="search" placeholder="Search" aria-label="Search">

<button class="btn btn-outline-success" type="submit">Search</button>

</form>

</div>

</div>

</nav>

{% if messages %}

{% for message in messages %}

<div class="alert alert-{{ message.tag }}">

<div class="p-3 mb-2 bg-info text-white text-center">

{{ message }}

</div>

</div>

{% endfor %}

{% endif %}

{% block body %}

{% endblock %}

<!-- Optional JavaScript; choose one of the two! -->

<!-- Option 1: Bootstrap Bundle with Popper -->

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js" integrity="sha384-MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM" crossorigin="anonymous"></script>

<!-- Option 2: Separate Popper and Bootstrap JS -->

<!--

<script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.9.2/dist/umd/popper.min.js" integrity="sha384-IQsoLXl5PILFhosVNubq5LC7Qb9DXgDA9i+tQ8Zj3iwWAwPtgFTxbJ8NT4GN1R8p" crossorigin="anonymous"></script>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.min.js" integrity="sha384-cVKIPhGWiC2Al4u+LWgxfKTRIcfu0JTxR+EQDz/bgldoEyl4H0zUF0QKbrJ0EcQF" crossorigin="anonymous"></script>

-->

</body>

</html>

# Restricting Routes

83 – users => views.py

from django.contrib.auth.decorators import login\_required

from django.shortcuts import render, redirect

from django.contrib.auth.forms import UserCreationForm

from django.contrib import messages

from .forms import RegisterForm

def register(request):

if request.method == 'POST':

form = RegisterForm(request.POST)

if form.is\_valid():

form.save()

username = form.cleaned\_data.get('username')

messages.success(request, f'Welcome {username}, your account is created!')

return redirect('food:index')

else:

form = RegisterForm()

return render(request, 'users/register.html', {'form': form})

@login\_required

def profilepage(request):

return render(request, 'users/profile.html')

84 – FastFood\_01 => settings.py

…

LOGIN\_REDIRECT\_URL = 'food:index'

LOGIN\_URL = 'login'

…

85 – FastFood\_01 => urls.py

from django.contrib import admin

from django.urls import path, include

from users import views as user\_views

from django.contrib.auth import views as authentication\_views

urlpatterns = [

path('admin/', admin.site.urls),

path('food/', include('food.urls')),

path('register/', user\_views.register, name='register'),

path('login/', authentication\_views.LoginView.as\_view(template\_name='users/login.html'), name='login'),

path('logout/', authentication\_views.LogoutView.as\_view(template\_name='users/logout.html'), name='logout'),

path('profile/', user\_views.profilepage, name='profile'),

]

86 – templates => users => profile.html

{% extends "base.html" %}

{% block title %} {{ user.username }} {% endblock %}

{% block body %}

<div class="p-3 mb-2 bg-success text-white text-center"><h1>{{ user.username }}</h1></div>

{% endblock %}

87 – templates => base.html

{% load static %}

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>

{% block title %}

{% endblock %}

</title>

<!-- Bootstrap CSS -->

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC" crossorigin="anonymous">

<link rel="stylesheet" href="{% static 'food/style.css' %}">

</head>

<body>

<nav class="navbar navbar-expand-lg navbar-dark bg-dark">

<div class="container-fluid">

<a class="navbar-brand" href="{% url 'profile' %}">{{ user.username }}</a>

<button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse" id="navbarSupportedContent">

<ul class="navbar-nav me-auto mb-2 mb-lg-0">

<li class="nav-item">

<a class="nav-link active" aria-current="page" href="{% url 'food:index' %}">Home</a>

</li>

{% if user.is\_authenticated %}

<li class="nav-item">

<a class="nav-link" href="{% url 'logout' %}">Logout</a>

</li>

<li class="nav-item">

<a class="nav-link" href="{% url 'food:create\_item' %}">Add Item</a>

</li>

{% else %}

<li class="nav-item">

<a class="nav-link" href="{% url 'register' %}">Resister</a>

</li>

<li class="nav-item">

<a class="nav-link" href="{% url 'login' %}">Login</a>

</li>

{% endif %}

<li class="nav-item dropdown">

<a class="nav-link dropdown-toggle" href="#" id="navbarDropdown" role="button" data-bs-toggle="dropdown" aria-expanded="false">

Dropdown

</a>

<ul class="dropdown-menu" aria-labelledby="navbarDropdown">

<li><a class="dropdown-item" href="#">Action</a></li>

<li><a class="dropdown-item" href="#">Another action</a></li>

<li><hr class="dropdown-divider"></li>

<li><a class="dropdown-item" href="#">Something else here</a></li>

</ul>

</li>

</ul>

<form class="d-flex">

<input class="form-control me-2" type="search" placeholder="Search" aria-label="Search">

<button class="btn btn-outline-success" type="submit">Search</button>

</form>

</div>

</div>

</nav>

{% if messages %}

{% for message in messages %}

<div class="alert alert-{{ message.tag }}">

<div class="p-3 mb-2 bg-info text-white text-center">

{{ message }}

</div>

</div>

{% endfor %}

{% endif %}

{% block body %}

{% endblock %}

<!-- Optional JavaScript; choose one of the two! -->

<!-- Option 1: Bootstrap Bundle with Popper -->

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js" integrity="sha384-MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM" crossorigin="anonymous"></script>

<!-- Option 2: Separate Popper and Bootstrap JS -->

<!--

<script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.9.2/dist/umd/popper.min.js" integrity="sha384-IQsoLXl5PILFhosVNubq5LC7Qb9DXgDA9i+tQ8Zj3iwWAwPtgFTxbJ8NT4GN1R8p" crossorigin="anonymous"></script>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.min.js" integrity="sha384-cVKIPhGWiC2Al4u+LWgxfKTRIcfu0JTxR+EQDz/bgldoEyl4H0zUF0QKbrJ0EcQF" crossorigin="anonymous"></script>

-->

</body>

</html>

# Creating The Profile Model

88 – FastFood\_01>pip install pillow

89 – users => models.py

from django.db import models

from django.contrib.auth.models import User

class Profile(models.Model):

user = models.OneToOneField(User, on\_delete=models.CASCADE)

image = models.ImageField(default='profilepic.jpg', upload\_to='profile\_pictures')

location = models.CharField(max\_length=100)

def \_\_str\_\_(self):

return self.user.username

90 - FastFood\_01>py manage.py makemigrations

91 - FastFood\_01>py manage.py migrate

92 – users => admin.py

from django.contrib import admin

from .models import Profile

admin.site.register(Profile)

# Adding Path To Upload Images

93 – FastFood\_01 => settings.py

import os.path

from pathlib import Path

# Build paths inside the project like this: BASE\_DIR / 'subdir'.

BASE\_DIR = Path(\_\_file\_\_).resolve().parent.parent

TEMPLATES\_DIR = os.path.join(BASE\_DIR, 'templates')

STATIC\_DIR = os.path.join(BASE\_DIR, 'static')

MEDIA\_DIR = os.path.join(BASE\_DIR, 'media')

# Quick-start development settings - unsuitable for production

# See https://docs.djangoproject.com/en/3.2/howto/deployment/checklist/

# SECURITY WARNING: keep the secret key used in production secret!

SECRET\_KEY = 'django-insecure-qyfr(%t-h6-59kw+jq8=+8-+9ksidg)r7-d)jzxhglyhlq58@v'

# SECURITY WARNING: don't run with debug turned on in production!

DEBUG = True

ALLOWED\_HOSTS = []

# Application definition

INSTALLED\_APPS = [

'django.contrib.admin',

'django.contrib.auth',

'django.contrib.contenttypes',

'django.contrib.sessions',

'django.contrib.messages',

'django.contrib.staticfiles',

'food.apps.FoodConfig',

'users.apps.UsersConfig',

]

MIDDLEWARE = [

'django.middleware.security.SecurityMiddleware',

'django.contrib.sessions.middleware.SessionMiddleware',

'django.middleware.common.CommonMiddleware',

'django.middleware.csrf.CsrfViewMiddleware',

'django.contrib.auth.middleware.AuthenticationMiddleware',

'django.contrib.messages.middleware.MessageMiddleware',

'django.middleware.clickjacking.XFrameOptionsMiddleware',

]

ROOT\_URLCONF = 'FastFood\_01.urls'

TEMPLATES = [

{

'BACKEND': 'django.template.backends.django.DjangoTemplates',

'DIRS': [BASE\_DIR / 'templates']

,

'APP\_DIRS': True,

'OPTIONS': {

'context\_processors': [

'django.template.context\_processors.debug',

'django.template.context\_processors.request',

'django.contrib.auth.context\_processors.auth',

'django.contrib.messages.context\_processors.messages',

],

},

},

]

WSGI\_APPLICATION = 'FastFood\_01.wsgi.application'

# Database

# https://docs.djangoproject.com/en/3.2/ref/settings/#databases

DATABASES = {

'default': {

'ENGINE': 'django.db.backends.sqlite3',

'NAME': BASE\_DIR / 'db.sqlite3',

}

}

# Password validation

# https://docs.djangoproject.com/en/3.2/ref/settings/#auth-password-validators

AUTH\_PASSWORD\_VALIDATORS = [

{

'NAME': 'django.contrib.auth.password\_validation.UserAttributeSimilarityValidator',

},

{

'NAME': 'django.contrib.auth.password\_validation.MinimumLengthValidator',

},

{

'NAME': 'django.contrib.auth.password\_validation.CommonPasswordValidator',

},

{

'NAME': 'django.contrib.auth.password\_validation.NumericPasswordValidator',

},

]

# Internationalization

# https://docs.djangoproject.com/en/3.2/topics/i18n/

LANGUAGE\_CODE = 'en-us'

TIME\_ZONE = 'UTC'

USE\_I18N = True

USE\_L10N = True

USE\_TZ = True

# Static files (CSS, JavaScript, Images)

# https://docs.djangoproject.com/en/3.2/howto/static-files/

STATIC\_URL = '/static/'

STATICFILES\_DIRS = [

BASE\_DIR / "static",

]

MEDIA\_ROOT = MEDIA\_DIR

MEDIA\_URL = '/media/'

LOGIN\_REDIRECT\_URL = 'food:index'

LOGIN\_URL = 'login'

# Default primary key field type

# https://docs.djangoproject.com/en/3.2/ref/settings/#default-auto-field

DEFAULT\_AUTO\_FIELD = 'django.db.models.BigAutoField'

94 – Add Profile Pic in Admin Panel Create Automat Root in media/profile\_picture/profile.jpg

# Adding The User Profile Picture

95 – FastFood\_01 => urls.py

from django.contrib import admin

from django.contrib.staticfiles.urls import staticfiles\_urlpatterns

from django.urls import path, include

from users import views as user\_views

from django.contrib.auth import views as authentication\_views

from django.conf import settings

from django.conf.urls.static import static

urlpatterns = [

path('admin/', admin.site.urls),

path('food/', include('food.urls')),

path('register/', user\_views.register, name='register'),

path('login/', authentication\_views.LoginView.as\_view(template\_name='users/login.html'), name='login'),

path('logout/', authentication\_views.LogoutView.as\_view(template\_name='users/logout.html'), name='logout'),

path('profile/', user\_views.profilepage, name='profile'),

]

urlpatterns += staticfiles\_urlpatterns()

urlpatterns += static(settings.MEDIA\_URL, document\_root=settings.MEDIA\_ROOT)

96 – templates => users => profile.html

{% extends "base.html" %}

{% block title %} {{ user.username }} {% endblock %}

{% block body %}

<div class="container-flute">

<div class="row">

<div class="card" style="width: 18rem;">

<img class="card-img-top" src="{{ user.profile.image.url }}" alt="{{ user.username }}">

<div class="card-body">

<h5 class="card-title">{{ user.username }}</h5>

<p class="card-text">{{ user.location }}</p>

</div>

</div>

</div>

</div>

{% endblock %}

# Setting Up The Default Profile Picture

97 – users => models.py

from django.db import models

from django.contrib.auth.models import User

class Profile(models.Model):

user = models.OneToOneField(User, on\_delete=models.CASCADE)

image = models.ImageField(default='profilepic.jpg', upload\_to='profile\_pictures')

location = models.CharField(max\_length=100)

def \_\_str\_\_(self):

return self.user.username

98 – Copy ----> profilepic.jpg ----> to media FOLDER

99 -