

# BACKEND WEB DEVELOPMENT

DJANGO – A PYTHON FRAMEWORK

SOURABH PAL

# LIST OF CLASS WORKS (till 26<sup>th</sup> September 2023)

- Display on client side (HttpResponse)
- Display a html page from templates
- Admin panel (adding members)
- Pull data from database (admin panel added members information display)
- Creating a form (Get and post methods) – applying CSS

# DOWNLOAD VISUAL STUDIO CODE AND PYTHON

## DOWNLOAD LINKS:-

- <https://code.visualstudio.com/download>
- <https://www.python.org/downloads/>

1.

# Creating first Django project

DISPLAY CONTENT ON THE CLIENT SIDE  
(BROWSER)

# Creating project (use vs code terminal OR cmd)

## EXECUTE THE FOLLOWING CODES SEQUENTIALLY

- `mkdir <FolderName>`
- `cd <FolderName>`
- `python -m venv env`
- `cd env`
- `Scripts\activate.bat`
- `Python -m pip install Django`
- `Django-admin startproject <ProjectName>`
- `Cd <ProjectName>`
- `Python manage.py runserver`
- `Crtl + C`
- `Code .`

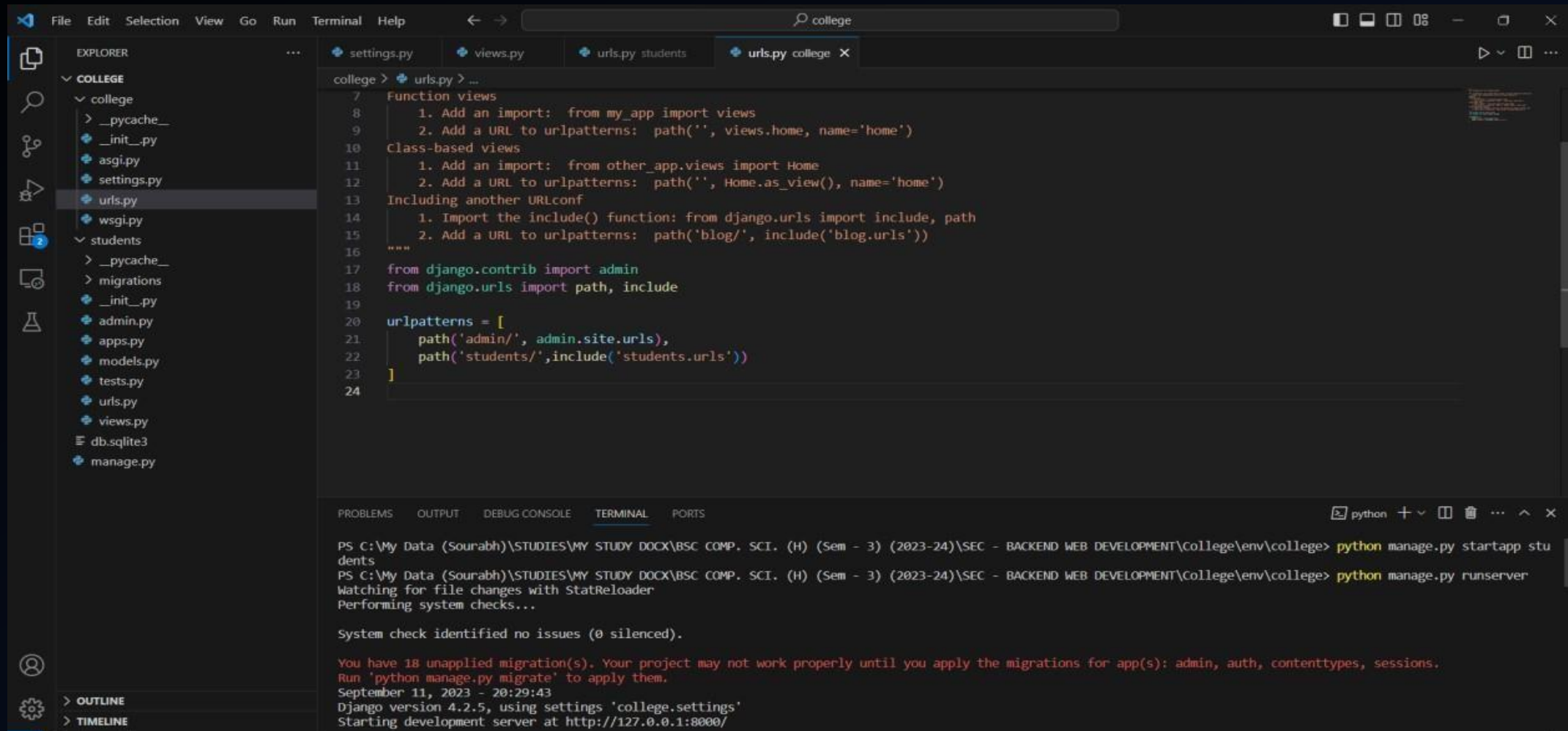
## NOTE THAT

- Runserver command will generate a url which shows if the project creation is successful or not.
- To break the connection from server, press ctrl + C.
- Code . Command will open the present directories (project) in vs code (ide)

## FOLLOW THESE STEPS

- Create app using the following command
- `Python manage.py startapp <appName>`
- Create `urls.py` in app directory
- Add the `appName` in quotes in the installed apps section of `settings.py` of the project
- Then make the following changes in `views.py`, `urls.py` (project and app).
- In the given example
- Directory name = College
- Project name = college
- App name = students

# PROJECT – URLS.PY



The screenshot displays a code editor interface with a dark theme. The Explorer panel on the left shows the project structure for a Django application named 'college'. The 'urls.py' file is selected and open in the editor. The code in 'urls.py' defines function-based and class-based views, includes another URLconf, and sets up the urlpatterns. The terminal at the bottom shows the command to start the Django development server.

```
7 Function views
8     1. Add an import: from my_app import views
9     2. Add a URL to urlpatterns: path('', views.home, name='home')
10 Class-based views
11     1. Add an import: from other_app.views import Home
12     2. Add a URL to urlpatterns: path('', Home.as_view(), name='home')
13 Including another URLconf
14     1. Import the include() function: from django.urls import include, path
15     2. Add a URL to urlpatterns: path('blog/', include('blog.urls'))
16 """
17 from django.contrib import admin
18 from django.urls import path, include
19
20 urlpatterns = [
21     path('admin/', admin.site.urls),
22     path('students/', include('students.urls'))
23 ]
24
```

PS C:\My Data (Sourabh)\STUDIES\MY STUDY DOCK\BSC COMP. SCI. (H) (Sem - 3) (2023-24)\SEC - BACKEND WEB DEVELOPMENT\College\env\college> python manage.py startapp students

PS C:\My Data (Sourabh)\STUDIES\MY STUDY DOCK\BSC COMP. SCI. (H) (Sem - 3) (2023-24)\SEC - BACKEND WEB DEVELOPMENT\College\env\college> python manage.py runserver

Watching for file changes with StatReloader

Performing system checks...

System check identified no issues (0 silenced).

You have 18 unapplied migration(s). Your project may not work properly until you apply the migrations for app(s): admin, auth, contenttypes, sessions.

Run 'python manage.py migrate' to apply them.

September 11, 2023 - 20:29:43

Django version 4.2.5, using settings 'college.settings'

Starting development server at http://127.0.0.1:8000/



# APP – URLS.PY

The image shows a Visual Studio Code editor window with a Django project named 'college'. The Explorer sidebar on the left displays the project structure, with the 'students' app folder expanded. The 'urls.py' file within the 'students' app is selected and highlighted. The main editor area shows the code for 'urls.py', which includes imports for 'path' from 'django.urls' and 'views' from the local app, and a list of URL patterns. The bottom panel shows the terminal with commands to start and run the Django application.

**EXPLORER**

- COLLEGE
  - college
    - \_\_pycache\_\_
    - \_\_init\_\_.py
    - asgi.py
    - settings.py
    - urls.py
    - wsgi.py
  - students
    - \_\_pycache\_\_
    - migrations
    - \_\_init\_\_.py
    - admin.py
    - apps.py
    - models.py
    - tests.py
    - urls.py**
    - views.py
  - db.sqlite3
  - manage.py

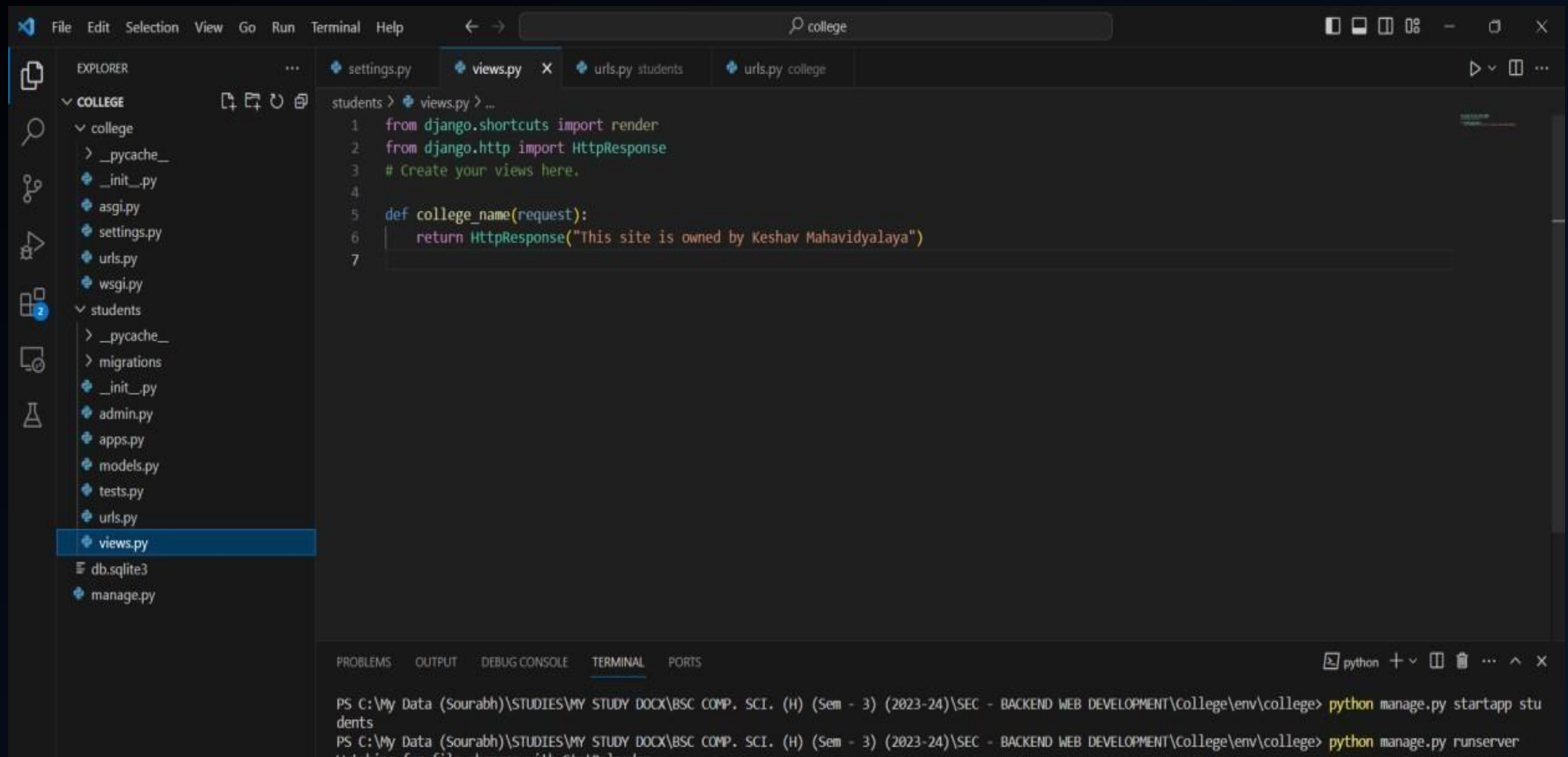
**urls.py students**

```
1 #this is urls.py of the app students
2
3 from django.urls import path
4 from . import views
5
6 urlpatterns = [
7     path('hello/', views.college_name)
8 ]
```

**TERMINAL**

```
PS C:\My Data (Sourabh)\STUDIES\MY STUDY DOCX\BSC COMP. SCI. (H) (Sem - 3) (2023-24)\SEC - BACKEND WEB DEVELOPMENT\College\env\college> python manage.py startapp stu
dents
PS C:\My Data (Sourabh)\STUDIES\MY STUDY DOCX\BSC COMP. SCI. (H) (Sem - 3) (2023-24)\SEC - BACKEND WEB DEVELOPMENT\College\env\college> python manage.py runserver
```

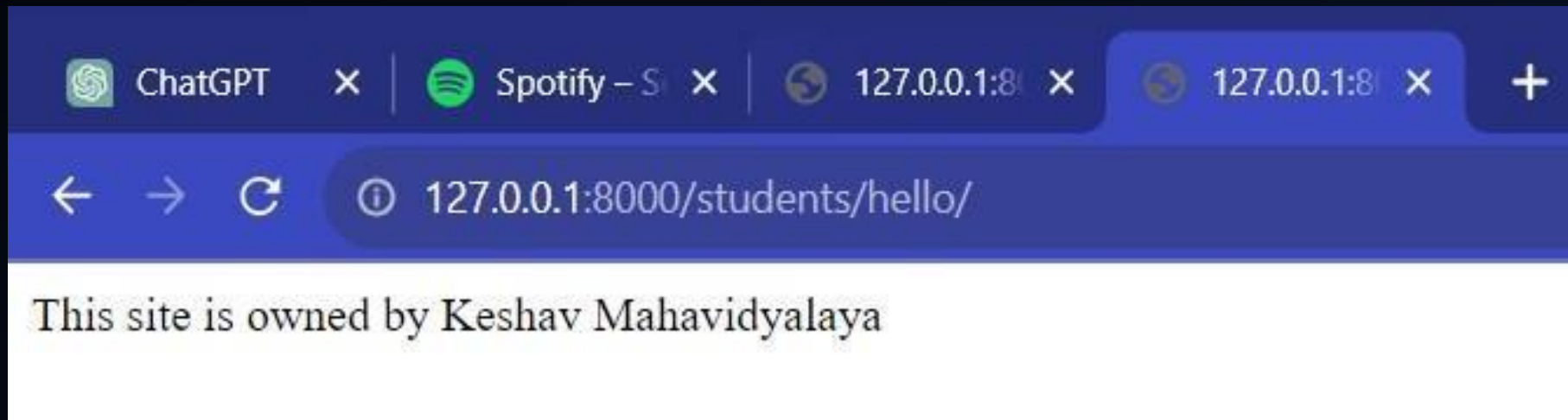
# VIEWS.PY



## BROWSER

- Run the following command
- `Python manage.py runserver`
- Then go to the url generated
- Then append the url with `/students/hello`

# RESULT



2.

# Creating templates

DISPLAY HTML PAGES ON THE CLIENT SIDE  
(BROWSER)

## TEMPLATE – HTML PAGE

- To the same project, make the following changes
- Create templates folder inside the app directory and create html page inside it.
- Then Change views.py

# VIEWS.PY

settings.py

views.py X

structure.html

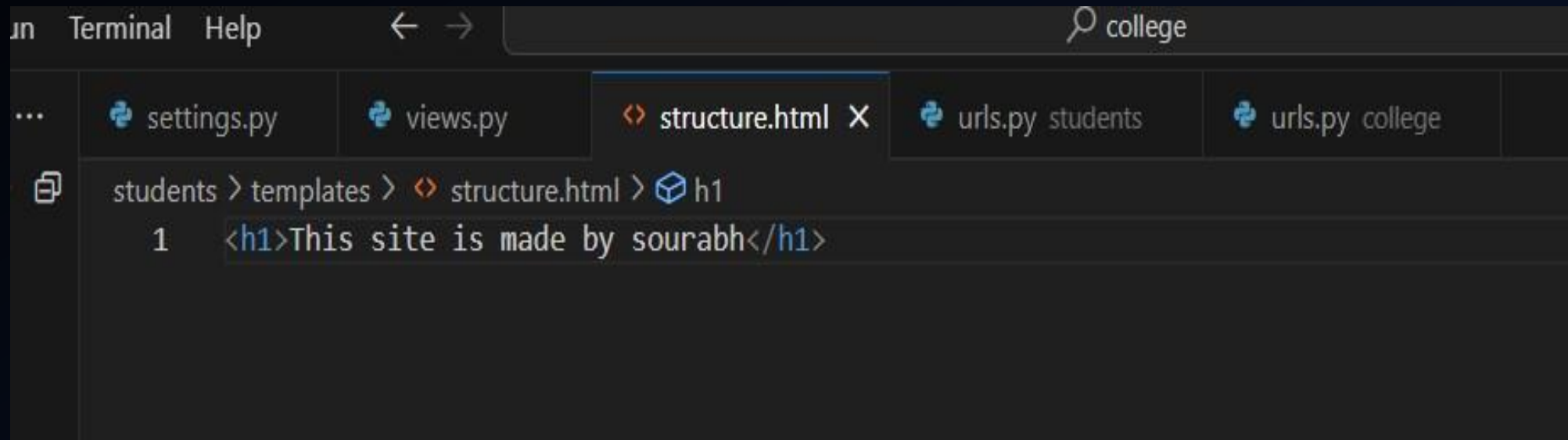
urls.py students

urls.py college

students > views.py > college\_name

```
1 from django.shortcuts import render
2 from django.http import HttpResponse
3 # Create your views here.
4
5 def college_name(request):
6     #return HttpResponse("This site is owned by Keshav Mahavidyalaya")
7     return render(request, 'structure.html')
```

# HTML PAGE INSIDE TEMPLATES



The screenshot shows a web browser window with a dark theme. The address bar displays 'college'. The browser's tab bar shows several tabs: 'settings.py', 'views.py', 'structure.html' (which is the active tab), 'urls.py students', and 'urls.py college'. The main content area of the browser shows the file path 'students > templates > structure.html > h1' and the HTML content: `1 <h1>This site is made by sourabh</h1>`. The text is displayed in a monospaced font, and the line number '1' is visible on the left side of the code block.



# BROWSER



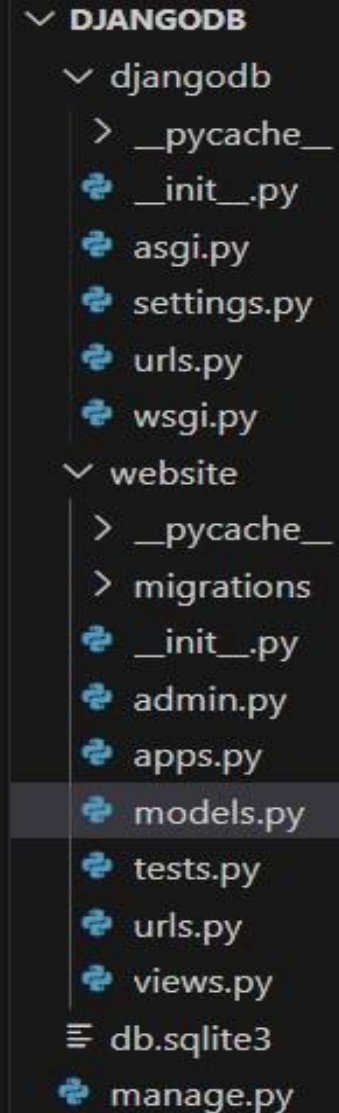
3.

# The Admin panel

USER ADDING DATA TO THE DATABASE

## INFORMATION

- Create another project
- Exampe given:
- Directory name = DjangoDb
- Project name =.djangodb
- Appname = website



A screenshot of a file explorer window showing the directory structure of a Django project. The root directory is 'DJANGODB', which contains two subdirectories: 'djangodb' and 'website'. The 'djangodb' directory contains files: '\_\_pycache\_\_', '\_\_init\_\_.py', 'asgi.py', 'settings.py', 'urls.py', and 'wsgi.py'. The 'website' directory contains files: '\_\_pycache\_\_', 'migrations', '\_\_init\_\_.py', 'admin.py', 'apps.py', 'models.py' (which is highlighted), 'tests.py', 'urls.py', 'views.py', 'db.sqlite3', and 'manage.py'.

```
▼ DJANGODB
  ▼ djangodb
    > __pycache__
    • __init__.py
    • asgi.py
    • settings.py
    • urls.py
    • wsgi.py
  ▼ website
    > __pycache__
    > migrations
    • __init__.py
    • admin.py
    • apps.py
    • models.py
    • tests.py
    • urls.py
    • views.py
    ≡ db.sqlite3
    • manage.py
```

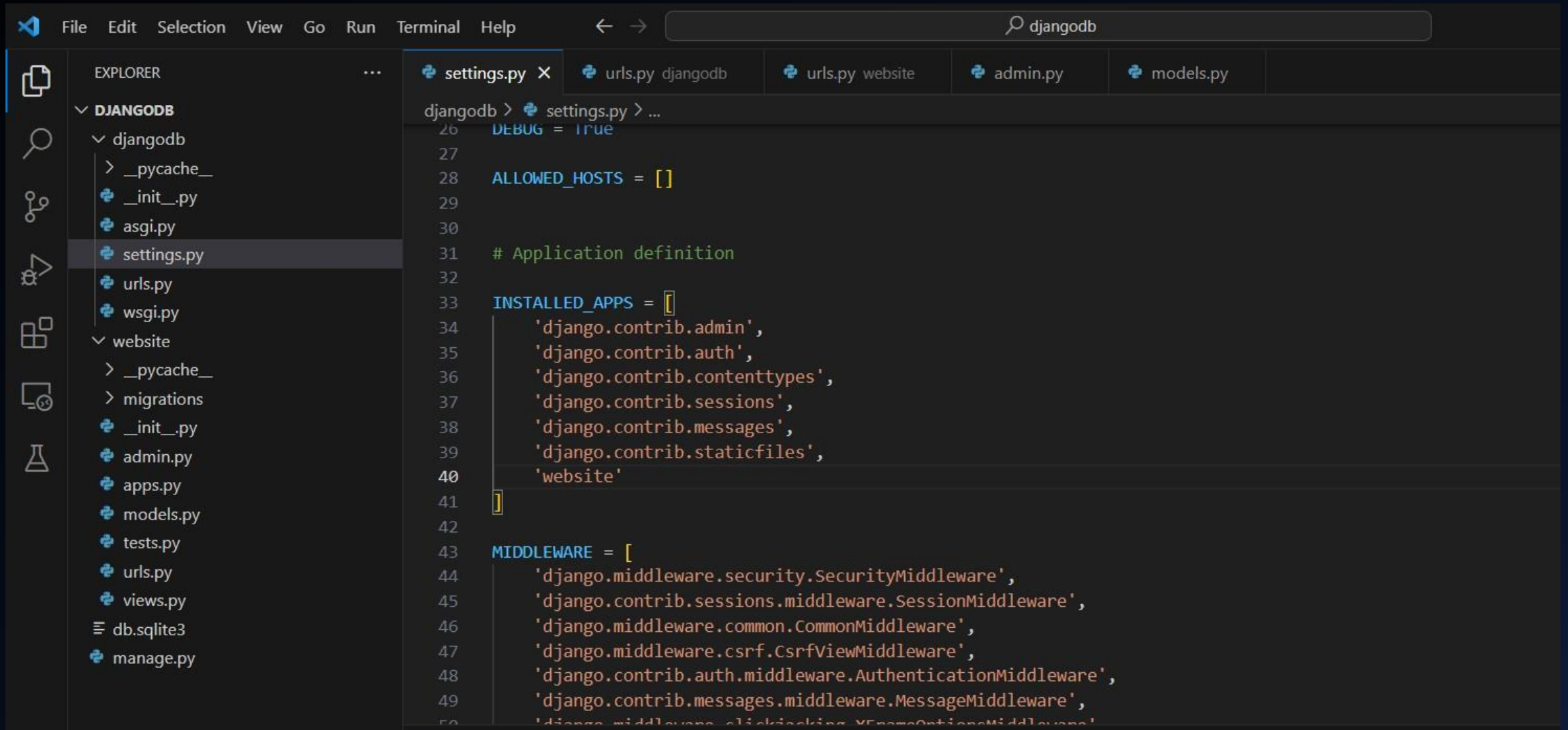
## INSTRUCTIONS

- Run -> `python manage.py migrate`
- Then make changes in the `models.py` and other files as shown in next slides
- Then create an admin user by running the command -> `python manage.py createsuperuser`
- Then create a user – remember the username and password
- Then run the command -> `python manage.py makemigrations`
- Then again -> `python manage.py migrate`
- `Python manage.py runserver`



CHANGES BEFORE  
RUNNING THE  
INSTRUCTED  
COMMANDS

# SETTINGS.PY



The image shows a code editor interface with a dark theme. On the left is the 'EXPLORER' sidebar showing a file tree for a project named 'DJANGODB'. The tree includes a 'djangodb' folder with subfolders like '\_\_pycache\_\_' and files like '\_\_init\_\_.py', 'asgi.py', 'settings.py' (which is selected), 'urls.py', and 'wsgi.py'. There is also a 'website' folder with its own '\_\_pycache\_\_', 'migrations', and '\_\_init\_\_.py' files, plus 'admin.py', 'apps.py', 'models.py', 'tests.py', another 'urls.py', and 'views.py'. At the bottom of the tree are 'db.sqlite3' and 'manage.py'.

The main editor area displays the 'settings.py' file for the 'djangodb' project. The code is as follows:

```
djangodb > settings.py > ...
26 DEBUG = True
27
28 ALLOWED_HOSTS = []
29
30
31 # Application definition
32
33 INSTALLED_APPS = [
34     'django.contrib.admin',
35     'django.contrib.auth',
36     'django.contrib.contenttypes',
37     'django.contrib.sessions',
38     'django.contrib.messages',
39     'django.contrib.staticfiles',
40     'website'
41 ]
42
43 MIDDLEWARE = [
44     'django.middleware.security.SecurityMiddleware',
45     'django.contrib.sessions.middleware.SessionMiddleware',
46     'django.middleware.common.CommonMiddleware',
47     'django.middleware.csrf.CsrfViewMiddleware',
48     'django.contrib.auth.middleware.AuthenticationMiddleware',
49     'django.contrib.messages.middleware.MessageMiddleware',
50     'django.middleware.clickjacking.XFrameOptionsMiddleware',
```

# MODELS.PY

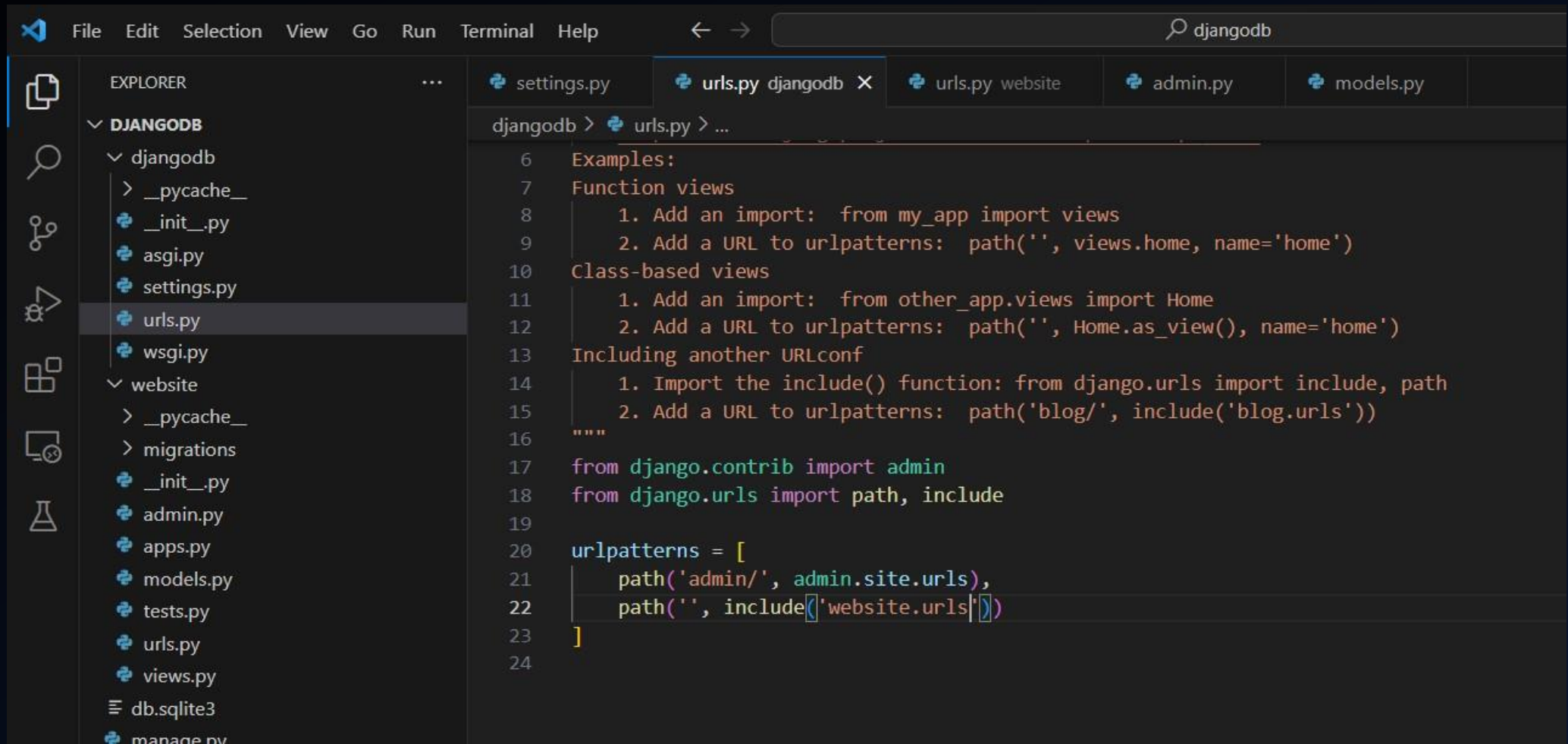
```
Terminal  Help  ← →  🔍 djangodb

⚙ settings.py  ⚙ urls.py djangodb  ⚙ urls.py website  ⚙ admin.py  ⚙ models.py ✕

website > ⚙ models.py > 📄 Members
1  from django.db import models
2
3  # Create your models here.
4  class Members(models.Model):
5      fname = models.CharField(max_length=20)
6      lname = models.CharField(max_length=20)
7      email = models.EmailField(max_length=30)
8      passwd = models.CharField(max_length=20)
9      age = models.IntegerField()
10
11
```



# PROJECT – URLS.PY



The image shows a code editor interface with a dark theme. The Explorer panel on the left displays the project structure for 'DJANGOODB'. The 'urls.py' file is selected and highlighted. The main editor area shows the content of 'urls.py', which includes examples for function and class-based views, and a configuration for including another URLconf. The file is named 'urls.py' and is located in the 'website' directory.

**EXPLORER**

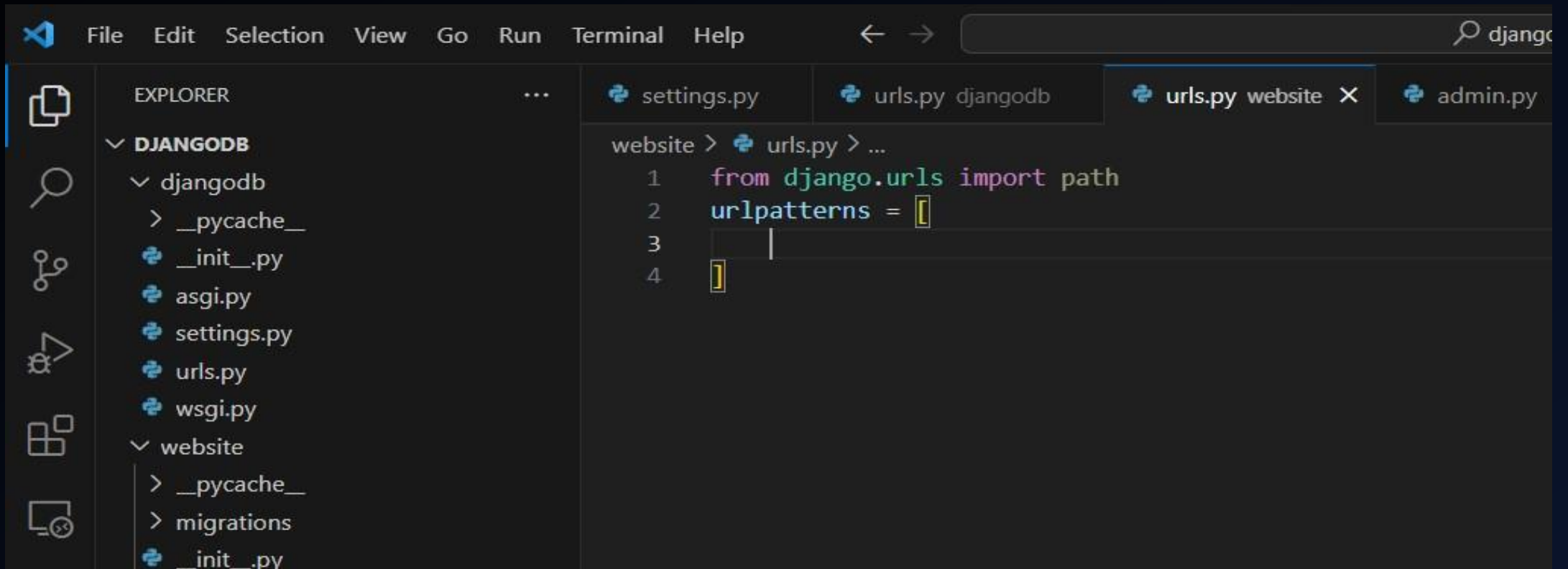
- ▼ DJANGOODB
  - ▼ djangoodb
    - > \_\_pycache\_\_
    - \_\_init\_\_.py
    - asgi.py
    - settings.py
    - urls.py**
    - wsgi.py
  - ▼ website
    - > \_\_pycache\_\_
    - > migrations
    - \_\_init\_\_.py
    - admin.py
    - apps.py
    - models.py
    - tests.py
    - urls.py
    - views.py
  - db.sqlite3
  - manage.py

**urls.py**

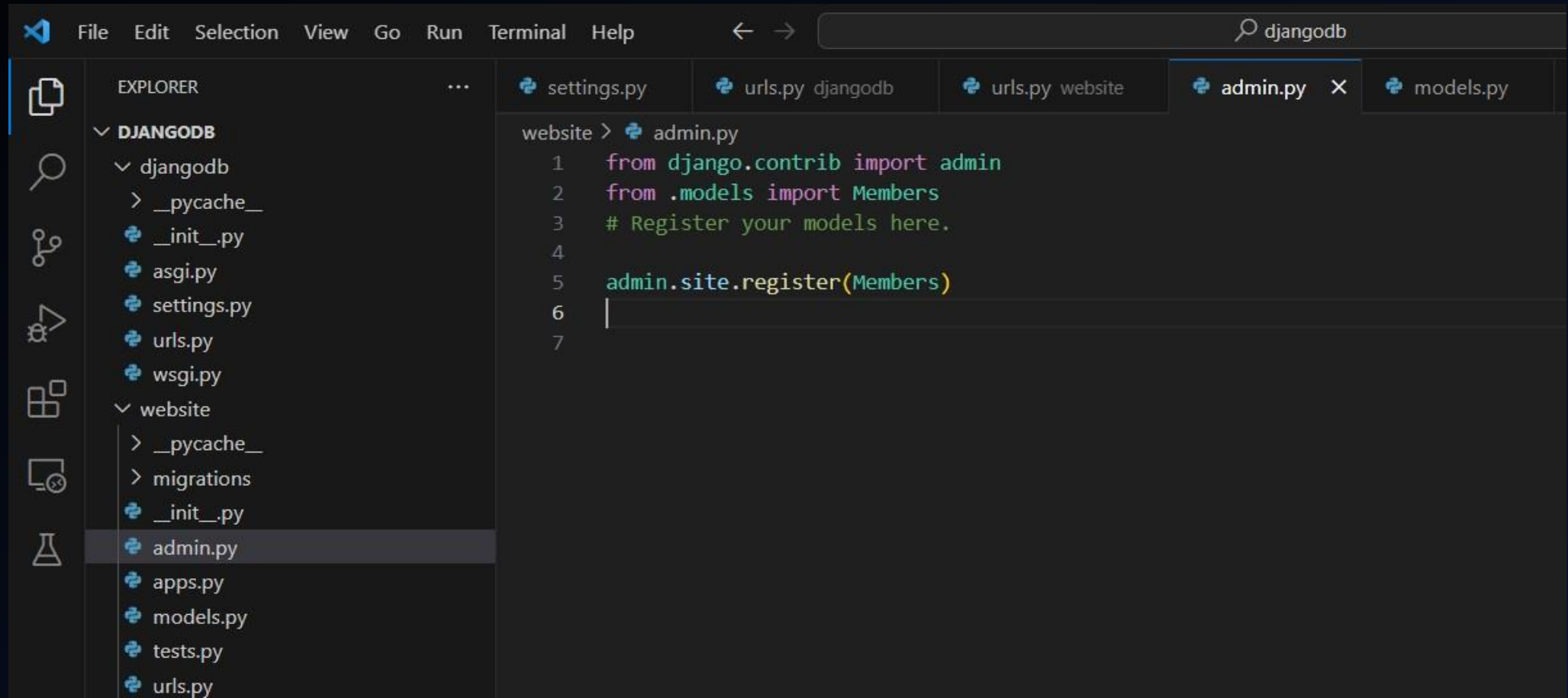
```
6 Examples:
7 Function views
8     1. Add an import:  from my_app import views
9     2. Add a URL to urlpatterns:  path('', views.home, name='home')
10 Class-based views
11     1. Add an import:  from other_app.views import Home
12     2. Add a URL to urlpatterns:  path('', Home.as_view(), name='home')
13 Including another URLconf
14     1. Import the include() function: from django.urls import include, path
15     2. Add a URL to urlpatterns:  path('blog/', include('blog.urls'))
16 """
17 from django.contrib import admin
18 from django.urls import path, include
19
20 urlpatterns = [
21     path('admin/', admin.site.urls),
22     path('', include('website.urls'))
23 ]
24
```



# APP – URLS.PY



# ADMIN.PY



The image shows a code editor interface with a dark theme. The Explorer panel on the left shows a project structure with a folder named 'DJANGODB' containing a subfolder 'django' and a file 'admin.py'. The 'admin.py' file is selected and highlighted. The main editor area shows the content of 'admin.py', which includes imports for 'admin' and 'Members', a comment, and a call to 'admin.site.register(Members)'. The file explorer on the left also shows other files like 'urls.py', 'wsgi.py', 'apps.py', 'models.py', 'tests.py', and 'urls.py'.

```
File Edit Selection View Go Run Terminal Help  
EXPLORER  
▼ DJANGODB  
  ▼ django  
    > __pycache__  
    _init_.py  
    asgi.py  
    settings.py  
    urls.py  
    wsgi.py  
  ▼ website  
    > __pycache__  
    > migrations  
    _init_.py  
    admin.py  
    apps.py  
    models.py  
    tests.py  
    urls.py  
settings.py urls.py django urls.py website admin.py models.py  
website > admin.py  
1 from django.contrib import admin  
2 from .models import Members  
3 # Register your models here.  
4  
5 admin.site.register(Members)  
6  
7
```

# ADDING MEMBERS ON THE CLIENT SIDE

The screenshot displays the Django administration interface in a web browser. The browser's address bar shows the URL `127.0.0.1:8000/admin/website/members/add/`. The page title is "Django administration".

The left sidebar contains a navigation menu with the following items:

- Start typing to filter...
- AUTHENTICATION AND AUTHORIZATION
  - Groups [Add](#)
  - Users [Add](#)
- WEBSITE
  - Members [Add](#)

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

- Fname:
- Lname:
- Email:
- Password:
- Age:

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

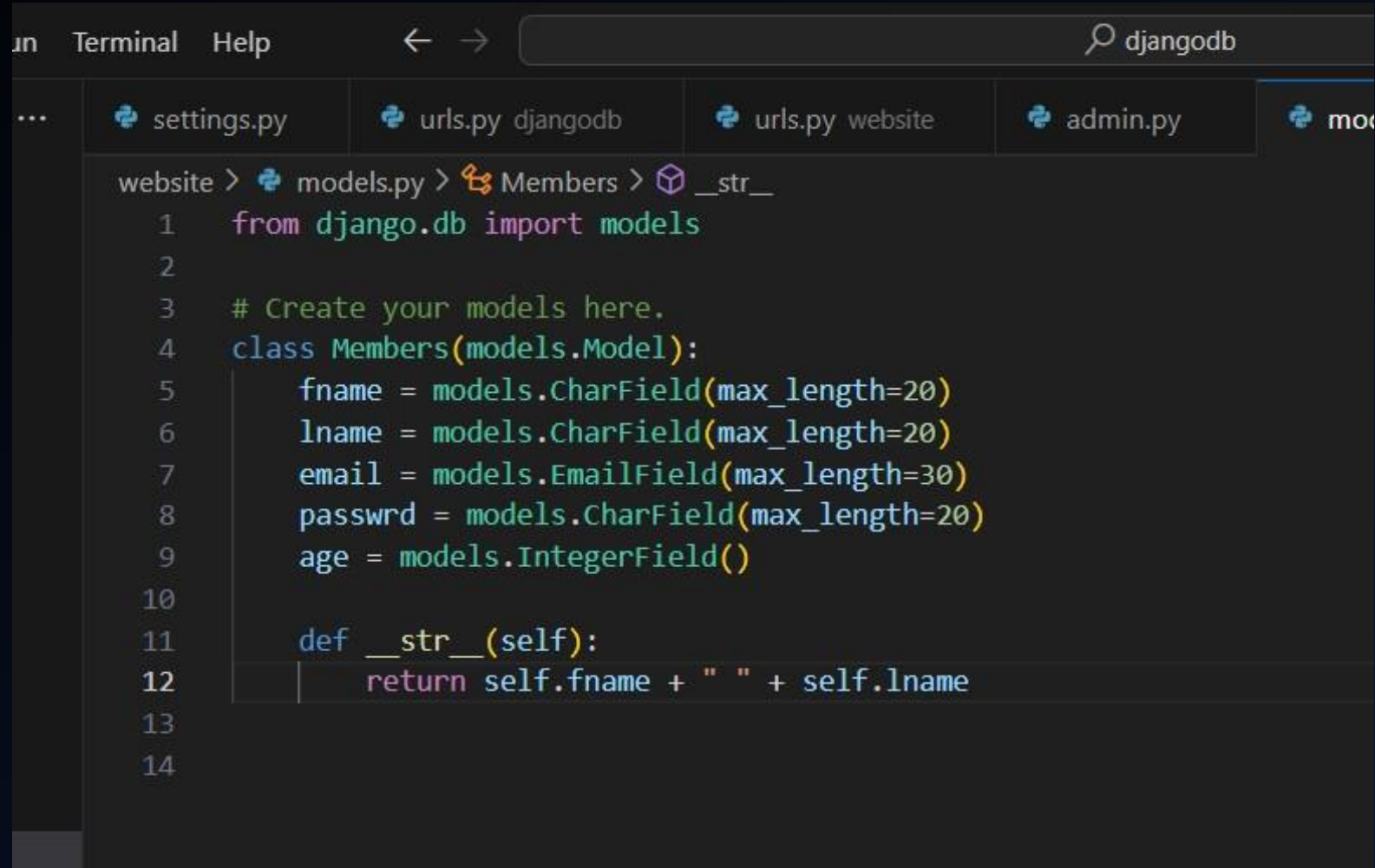
Below the form, a green message box states: "The members 'Members object (1)' was added successfully."

The bottom section of the page is titled "Select members to change" and includes an "ADD MEMBERS +" button. It features an "Action:" dropdown menu and a "Go" button, with the text "0 of 1 selected" next to it. Below this, there is a list of members:

- ☐ MEMBERS
- ☐ Members object (1)

At the bottom of this section, it says "1 members".

# UPDATED MODELS.PY – TO DISPLAY NAME LIST



```
website > models.py > Members > __str__
1  from django.db import models
2
3  # Create your models here.
4  class Members(models.Model):
5      fname = models.CharField(max_length=20)
6      lname = models.CharField(max_length=20)
7      email = models.EmailField(max_length=30)
8      passwd = models.CharField(max_length=20)
9      age = models.IntegerField()
10
11     def __str__(self):
12         return self.fname + " " + self.lname
13
14
```

4.

# Pull data from database

EXTRACT DATA FROM DATABASE AND  
DISPLAY ON THE CLIENT SIDE

## INSTRUCTIONS

- This is continuation of the previous project in which we display the content of the database on to the client screen.
- These data are entered by the user in the admin panel earlier.
- We need to create an html file for this
- Let's name it showdata.html
- Make changes to the html page in the templates

# SHOWDATA.HTML

The screenshot displays the Visual Studio Code interface with a Django project named 'django' open. The Explorer sidebar on the left shows the project structure:

- EXPLORER
  - ▼ DJANGODB
    - ▼ django
      - > \_\_pycache\_\_
      - 🔌 \_\_init\_\_.py
      - 🔌 asgi.py
      - 🔌 settings.py
      - 🔌 urls.py
      - 🔌 wsgi.py
    - ▼ website
      - > \_\_pycache\_\_
      - > migrations
      - ▼ templates
        - 🔌 showdata.html (selected)
        - 🔌 \_\_init\_\_.py
        - 🔌 admin.py
        - 🔌 apps.py

The main editor area shows the content of 'showdata.html' with the following code:

```
website > templates > 🔌 showdata.html > 📁 ul > 📁 strong
1  <h1> Members Database</h1>
2
3  <ul>
4  {% for item in all %}
5      <strong>{{ item.fname }} {{ item.lname }}</strong>
6      <br/>
7      <li>Email : {{ item.email }}</li>
8      <li>Age: {{ item.age }}</li>
9      <br/><hr>
10 {% endfor %}
11 </ul>
```

# VIEWS.PY

settings.py

showdata.html

urls.py.djangodb

urls.py.website

admin.py

views.py

website > views.py > show

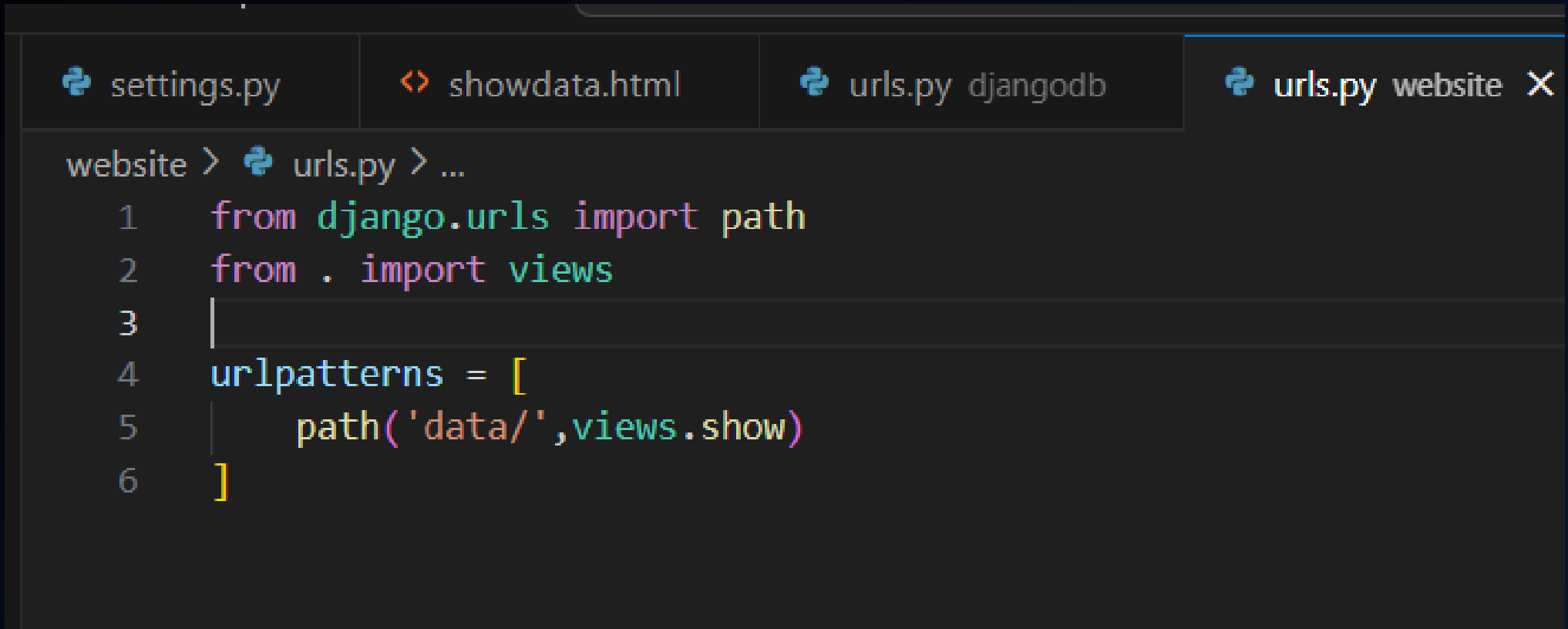
```
1 from django.shortcuts import render
2 from django.http import HttpResponse
3 from .models import Members
4 # Create your views here.
5
6 all_members = Members.objects.all
7 def show(request):
8     return render(request, 'showdata.html', {'all' : all_members})
```



## PROJECT – URLS.PY

```
15 |         2. Add a URL to urlpatterns: path('blog/', in
16 |         """
17 |         from django.contrib import admin
18 |         from django.urls import path, include
19 |
20 |         urlpatterns = [
21 |             path('admin/', admin.site.urls),
22 |             path('website/', include('website.urls'))
23 |         ]
24 |
```

## APP – URLS.PY

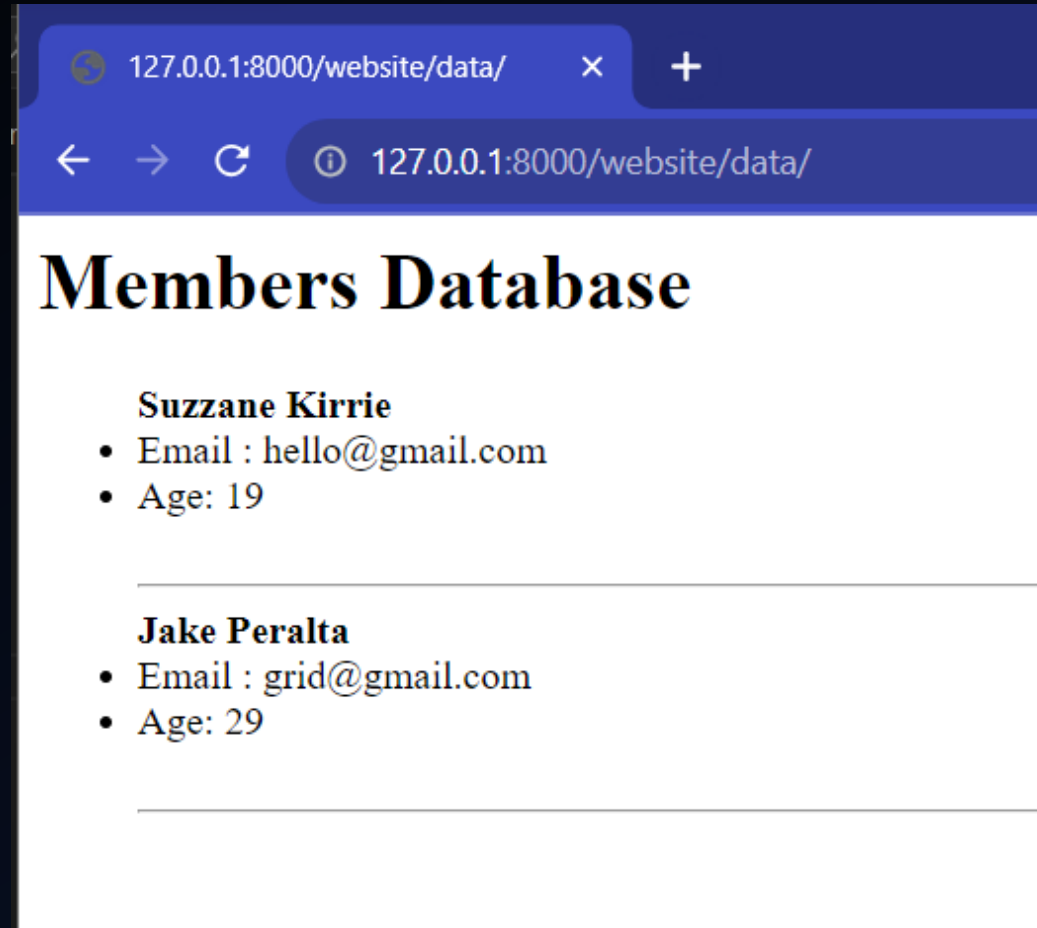


The screenshot shows a code editor with four tabs at the top: 'settings.py', 'showdata.html', 'urls.py.djangodb', and 'urls.py.website'. The 'urls.py.website' tab is active. The code in the editor is as follows:

```
website > urls.py > ...  
1  from django.urls import path  
2  from . import views  
3  
4  urlpatterns = [  
5      path('data/', views.show)  
6  ]
```

No change in admin.py and models.py

# CLIENT SIDE



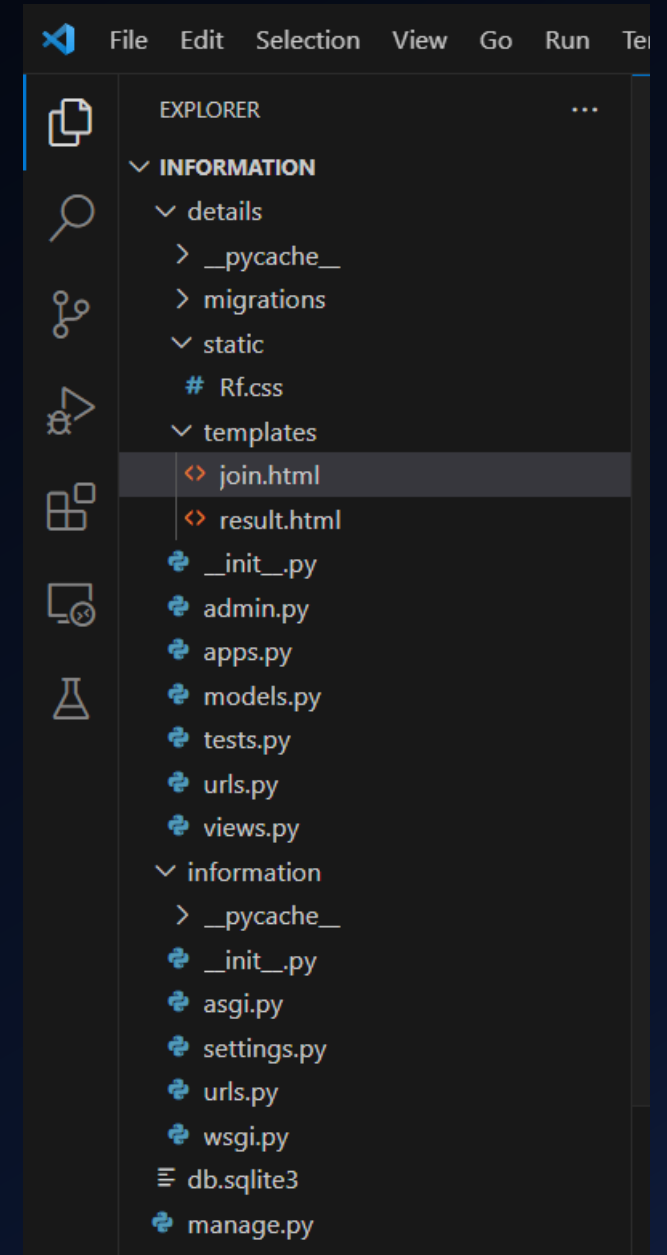
5.

# Creating Form and applying CSS

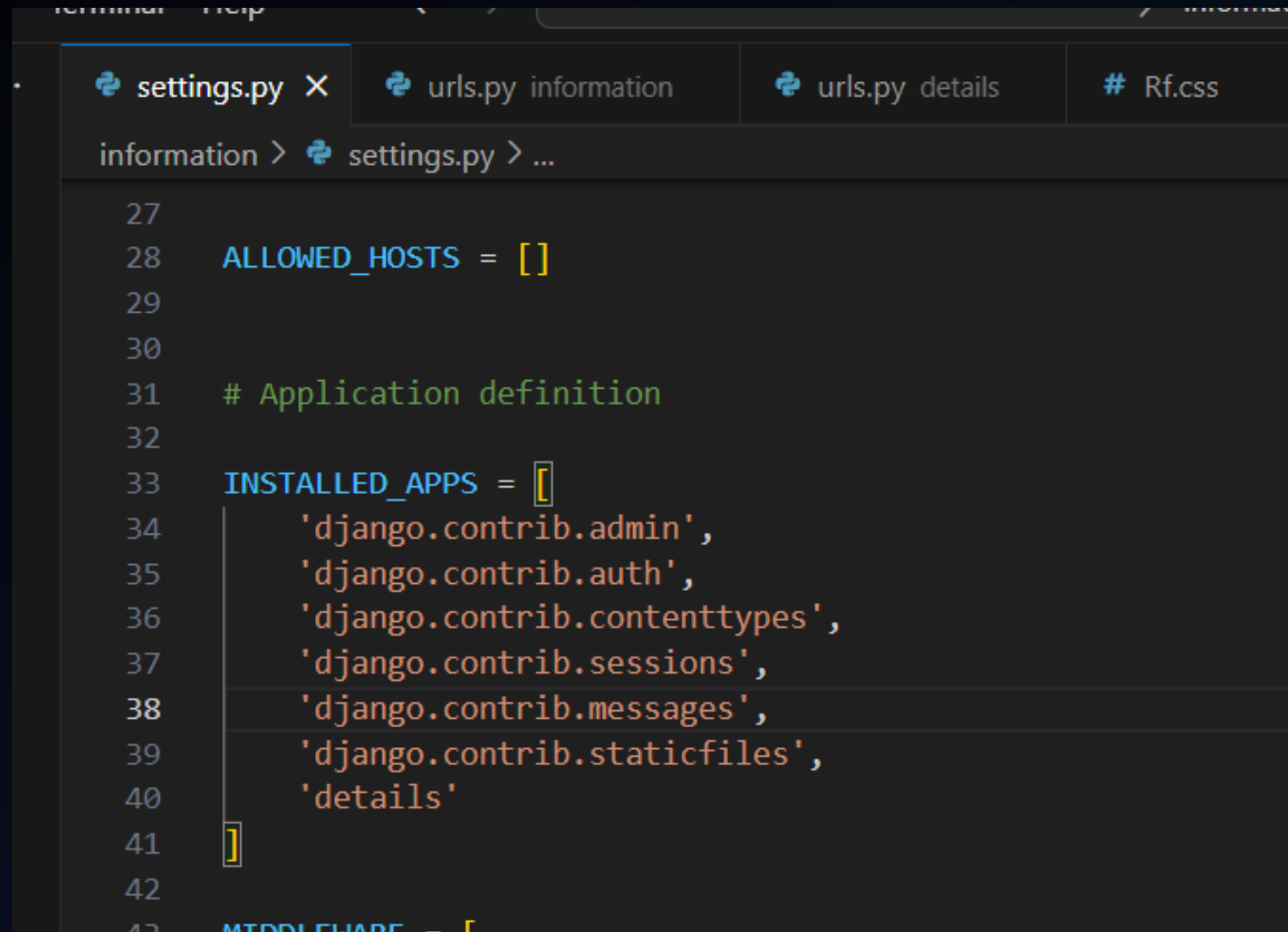
TAKING INPUT FROM USER AND  
SHOWING DESIRED OUTPUT

# INSTRUCTIONS

- Directory name = INFO
- Project name = information
- App name = details
- Html page 1 = join.html
- Html page 2 = result.html
- Views.py functions = join(), result()
- Make the following changes



# SETTINGS.PY

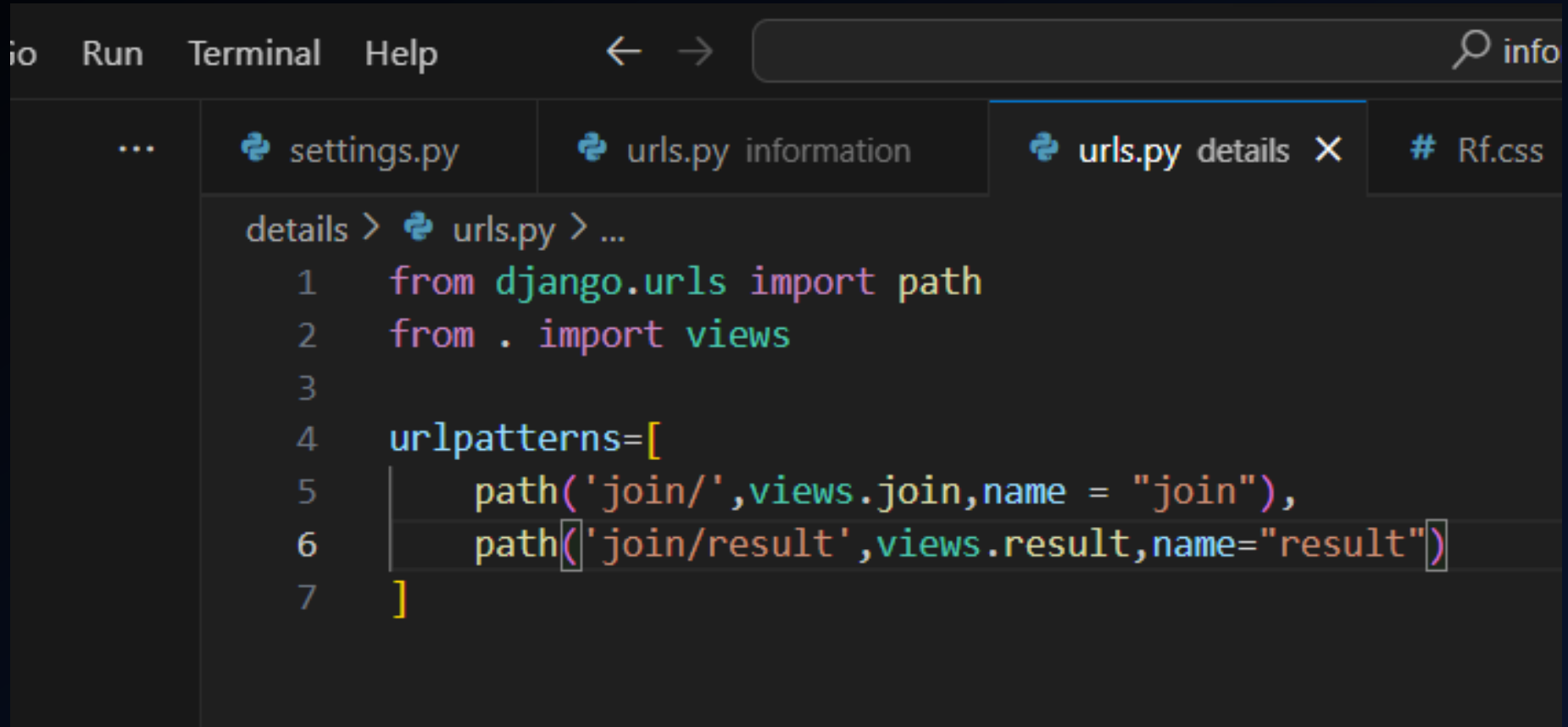


```
27
28 ALLOWED_HOSTS = []
29
30
31 # Application definition
32
33 INSTALLED_APPS = [
34     'django.contrib.admin',
35     'django.contrib.auth',
36     'django.contrib.contenttypes',
37     'django.contrib.sessions',
38     'django.contrib.messages',
39     'django.contrib.staticfiles',
40     'details'
41 ]
42
43 MIDDLEWARE = [
```

# PROJECT – URLS.PY

```
... settings.py urls.py information X urls.py details # Rf.css views.py
information > urls.py > ...
1  """
2  URL configuration for information project.
3
4  The `urlpatterns` list routes URLs to views. For more information please see:
5  | https://docs.djangoproject.com/en/4.2/topics/http/urls/
6  | Examples:
7  | Function views
8  |     1. Add an import: from my_app import views
9  |     2. Add a URL to urlpatterns: path('', views.home, name='home')
10 | Class-based views
11 |     1. Add an import: from other_app.views import Home
12 |     2. Add a URL to urlpatterns: path('', Home.as_view(), name='home')
13 | Including another URLconf
14 |     1. Import the include() function: from django.urls import include, path
15 |     2. Add a URL to urlpatterns: path('blog/', include('blog.urls'))
16 | """
17 from django.contrib import admin
18 from django.urls import path,include
19
20 urlpatterns = [
21     path('admin/', admin.site.urls),
22     path('details/', include('details.urls'))
23 ]
24
```

# APP – URLS.PY

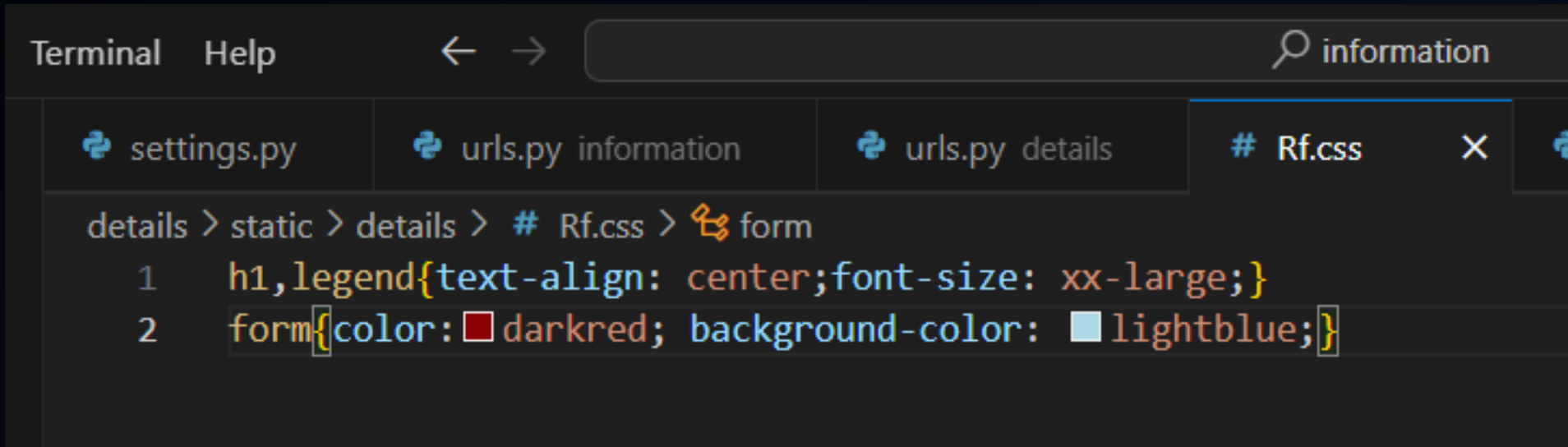


The image shows a code editor window with a dark theme. The top bar contains the menu items 'io', 'Run', 'Terminal', and 'Help', along with navigation arrows and a search icon labeled 'info'. Below the menu, there are four tabs: '...', 'settings.py', 'urls.py information', and 'urls.py details' (which is active and has a close button 'X'). To the right of the tabs is a tab labeled '# Rf.css'. The main editor area displays the following Python code:

```
details > urls.py > ...
1  from django.urls import path
2  from . import views
3
4  urlpatterns=[
5      path('join/',views.join,name = "join"),
6      path('join/result',views.result,name="result")
7  ]
```



# STYLESHEET – RF.CSS



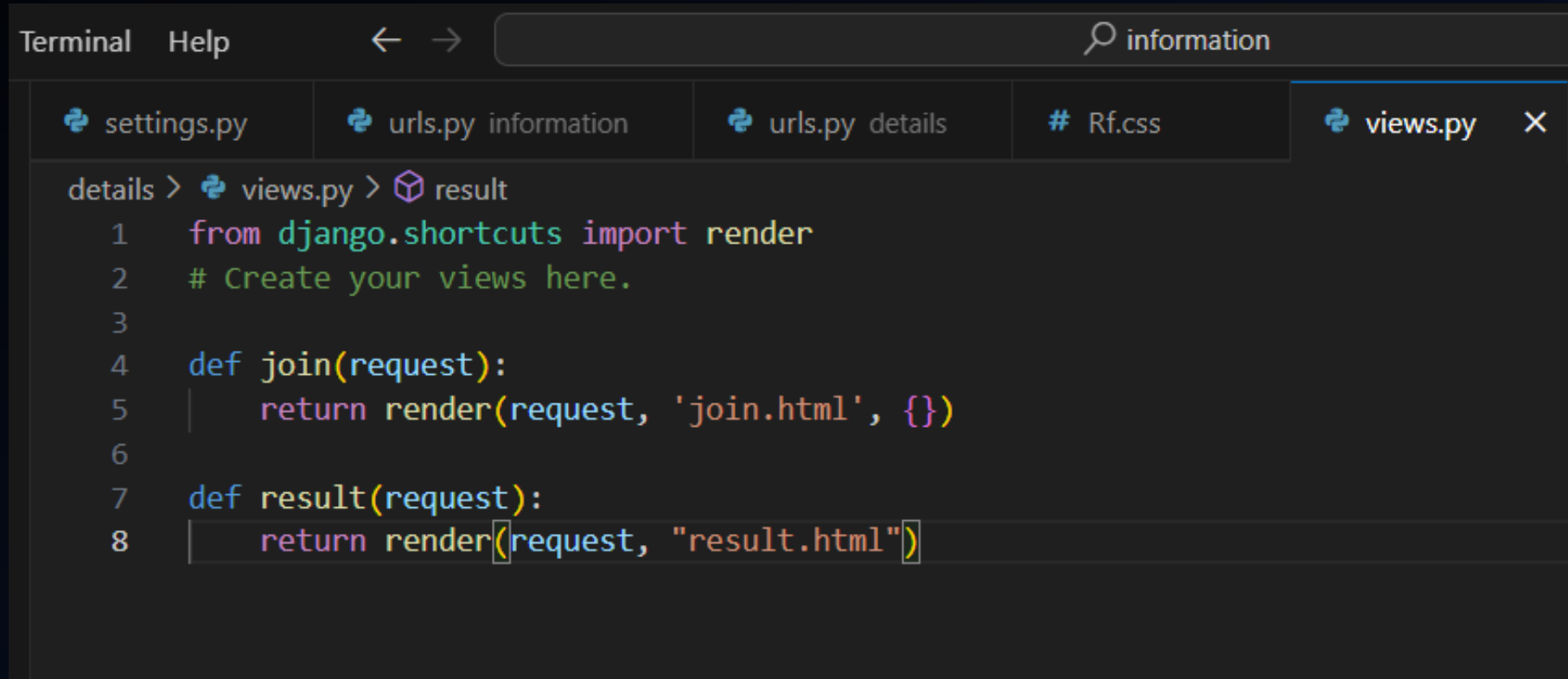
The image shows a web browser's developer tools interface. The top bar includes 'Terminal' and 'Help' menus, navigation arrows, and a search bar labeled 'information'. Below this, a tab bar shows several open files: 'settings.py', 'urls.py information', 'urls.py details', and the active tab, '# Rf.css'. The main content area displays the CSS file's path as 'details > static > details > # Rf.css > form'. The CSS code is listed with line numbers: line 1 contains 'h1,legend{text-align: center;font-size: xx-large;}' and line 2 contains 'form{color: darkred; background-color: lightblue;}'. The text 'darkred' is preceded by a small red square icon, and 'lightblue' is preceded by a small light blue square icon.

```
Terminal  Help  ←  →  information

settings.py  urls.py information  urls.py details  # Rf.css  ×

details > static > details > # Rf.css > form
1  h1,legend{text-align: center;font-size: xx-large;}
2  form{color: darkred; background-color: lightblue;}
```

# VIEWS.PY





```
Terminal  Help  <  >  information


settings.py  urls.py information  urls.py details  # Rf.css  views.py X

details > views.py > result
1  from django.shortcuts import render
2  # Create your views here.
3
4  def join(request):
5      return render(request, 'join.html', {})
6
7  def result(request):
8      return render(request, "result.html")
```


# RESULT.HTML



 settings.py



 urls.py information

 urls.py details

# Rf.css

 views.py

 result.html 

details > templates >  result.html >  h1

1 `<h1>Response Submitted</h1>`

# JOIN.HTML - 1

details > templates > join.html > html > body > script > validate

```
1  {% load static %}
2
3  <!DOCTYPE html>
4  <html>
5      <head>
6          <title>Registration Form</title>
7          <link rel="stylesheet" href="{% static 'Rf.css' %}">
8
9      </head>
10     <body>
11         <script>
12             function validate(){
13                 a = document.getElementById("naam").value;
14                 b = document.getElementById("phone").value;
15                 if (a === ""){
16                     alert("Name is compulsory");
17                 } else if (b === ""){
18                     alert("phone is compulsory");
19                 };
20             };
21
22         </script>
23         <h1>Examination Registration Form</h1>
24         <form action="result">
```

# JOIN.HTML - 2

```
20 </script>
21 <h1>Examination Registration Form</h1>
22 <form action="result">
23   <fieldset>
24     <legend>Personal Details</legend>
25
26     <label for="naam">Name:</label>
27     <input type="text" id="naam" >
28
29     <br>
30     <br>
31
32     <label for="phone">Phone Number:</label>
33     <input type="number" id="phone" >
34
35     <br>
36     <br>
37
38     <label for="email">Email::</label>
39     <input type="email" id ="email">
40
41     <br>
42     <br>
43
44     <label for="dob">Date Of Birth:</label>
45     <input type="date" id="dob">
46
47     <br>
48     <br>
49
50     <label for="add">Address:</label>
51     <textarea id="add"></textarea>
52
53     <br>
54     <br>
```

## JOIN.HTML - 3

```
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87

<br>
<br>

<label for="file">Upload your ID:</label>
<input type="file" id="file" >

</fieldset>

<hr>
<br>

<fieldset>
  <legend>Course Details</legend>

  <p>SELECT COURSE</p>
  <input type="radio" id = "op1" name="course">
  <label for="op1">B.Sc. Computer Science (Hons)</label>

  <br>

  <input type="radio" id = "op2" name="course">
  <label for="op2">B.Sc. Mathematics (Hons)</label>

  <br>

  <input type="radio" id = "op3" name="course">
  <label for="op3">BMS</label>

  <hr>
  <br>

  <p>SELECT SUBJECTS</p>

  <input type="checkbox" id = "sb1" checked>
  <label for="sb1">Mathematics</label>
```

## JOIN.HTML - 4

```
82
83 <p>SELECT SUBJECTS</p>
84
85 <input type="checkbox" id = "sb1" checked>
86 <label for="sb1">Mathematics</label>
87
88 <br>
89 <input type="checkbox" id = "sb2" checked>
90 <label for="sb2">Python</label>
91
92 <br>
93 <input type="checkbox" id = "sb3" checked>
94 <label for="sb3">Computer System Architechture</label>
95
96 <br>
97 <p>AEC:</p>
98 <input type="radio" id = "sb4a" name="hindi">
99 <label for="sb4a">Hindi - A</label>
100 <input type="radio" id = "sb4b" name="hindi">
101 <label for="sb4b">Hindi - B</label>
102 <input type="radio" id = "sb4c" name="hindi">
103 <label for="sb4c">Hindi - C</label>
104
105 <br>
106 <p>GE:</p>
107 <input type="radio" id = "sb5a" name="GE">
108 <label for="sb5a">COMMERCE</label>
109 <input type="radio" id = "sb5b" name="GE">
110 <label for="sb5b">CALCULUS</label>
111 <input type="radio" id = "sb5c" name="GE">
112 <label for="sb5c">ELECTRONICS</label>
113
```

# JOIN.HTML - 5

```
109 <input type="radio" id = "sb5b" name="GE" >
110 <label for="sb5b">CALCULUS</label>
111 <input type="radio" id = "sb5c" name="GE">
112 <label for="sb5c">ELECTRONICS</label>
113
114 <br>
115 <p>VAC:</p>
116 <input type="radio" id = "sb6a" name="VAC">
117 <label for="sb6a">VEDIC MATHEMATICS</label>
118 <input type="radio" id = "sb6b" name="VAC">
119 <label for="sb6b">DIGITAL EMPOWERMENT</label>
120 <input type="radio" id = "sb6c" name="VAC">
121 <label for="sb6c">PSYCHOLOGY</label>
122
123 <br>
124 <p>SEC:</p>
125 <input type="radio" id = "sb7a" name="SEC">
126 <label for="sb7a">FRONT-END</label>
127 <input type="radio" id = "sb7b" name="SEC">
128 <label for="sb7b">STATISTICS</label>
129 <input type="radio" id = "sb7c" name="SEC">
130 <label for="sb7c">PERSONAL FINANCE</label>
131
132
133 </fieldset>
134 <BR>
135 <BR>
136
137 <select>
138 <option value="">Select ratings</option>
139 <p>rate your experience</p>
140 <optgroup label="High ratings">
141
142 <option>10 stars</option>
143 <option>8 stars</option>
144
145 </optgroup>
```



## JOIN.HTML - 6

```
133     </fieldset>
134     <BR>
135     <BR>
136
137     <select>
138         <option value="">Select ratings</option>
139         <p>rate your experience</p>
140         <optgroup label="High ratings">
141
142             <option>10 stars</option>
143             <option>8 stars</option>
144
145         </optgroup>
146         <optgroup label="Low ratings">
147
148             <option>5 stars</option>
149             <option>3 stars</option>
150         </optgroup>
151
152
153     </select>
154     <br>
155     <br>
156
157     <button type="submit" onclick="validate()">SUBMIT</button>
158     <button type="reset">RESET</button>
159 </form>
160 </body>
161 </html>
```

# BROWSER

Spotify - Web Player: Music for € x

YouTube x

New Tab x

Registration Form x

+

← → ↻

🔒 127.0.0.1:8000/details/join/

🔍 📄 ☆ ⚙️ 🖨️ 🌐 👤 ⋮

## Examination Registration Form

### Personal Details

Name:

Phone Number:

Email:  📧

Date Of Birth:  📅

Address:  📍

Upload your ID:  No file chosen

### Course Details

SELECT COURSE

- ☐ B.Sc. Computer Science (Hons)
- ☐ B.Sc. Mathematics (Hons)
- ☐ BMS

SELECT SUBJECTS

- ☒ Mathematics
- ☒ Python
- ☒ Computer System Architecture

AEC:

- ☐ Hindi - A ☐ Hindi - B ☐ Hindi - C

GE:

- ☐ COMMERCE ☐ CALCULUS ☐ ELECTRONICS

VAC:

- ☐ VEDIC MATHEMATICS ☐ DIGITAL EMPOWERMENT ☐ PSYCHOLOGY

SEC:

- ☐ FRONT-END ☐ STATISTICS ☐ PERSONAL FINANCE

Select ratings ▼

33°C

High winds soon

🖥️

📁

📞

🌐

🔧

^

ENG IN

📶

🔊

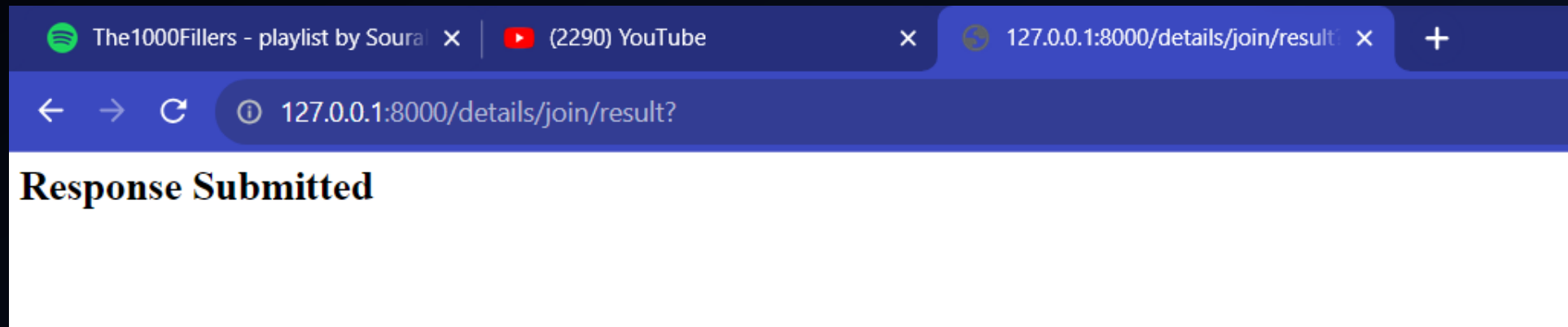
🔋

13:29

01-10-2023

1

# BROWSER – AFTER SUBMIT



6.

# Creating Evaluation Form (GET & POST methods) – with CSS

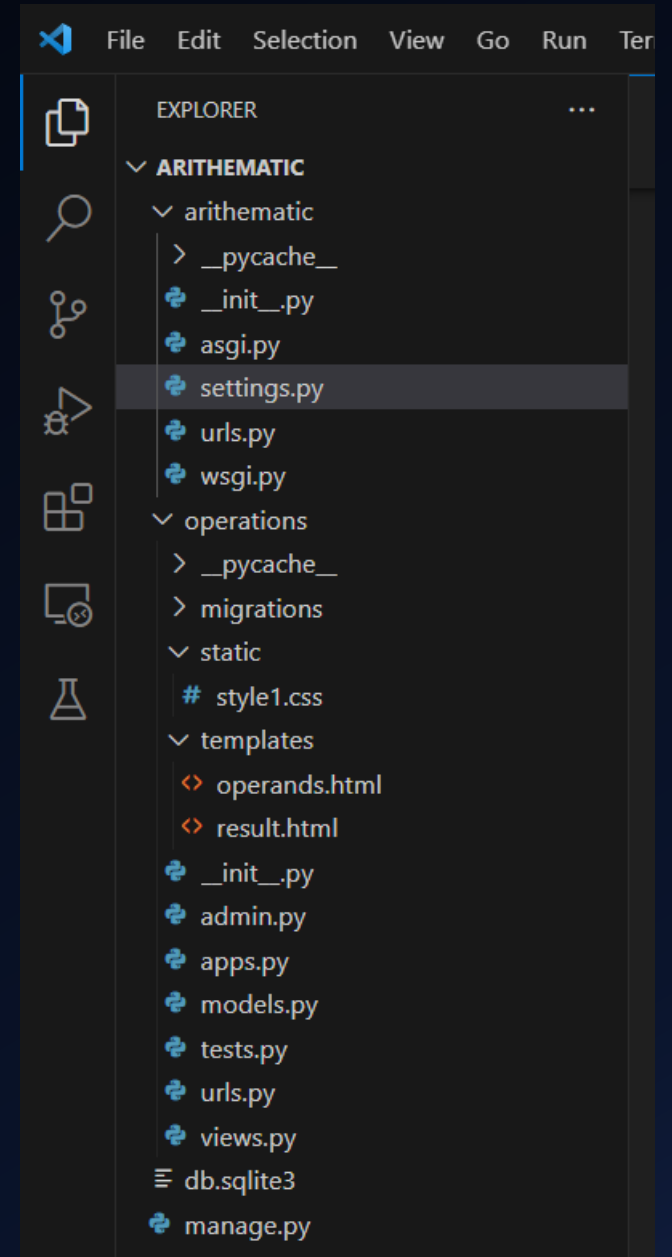
TAKING INPUT FROM USER AND  
SHOWING EVALUATED OUTPUT

## INSTRUCTIONS

- Directory name = Arithmetic
- Project name = arithmetic
- App name = operations
- Html page 1 =operands.html
- Html page 2 = result.html
- Views.py functions = enter(), add()
- Make the following changes

# SETTINGS.PY

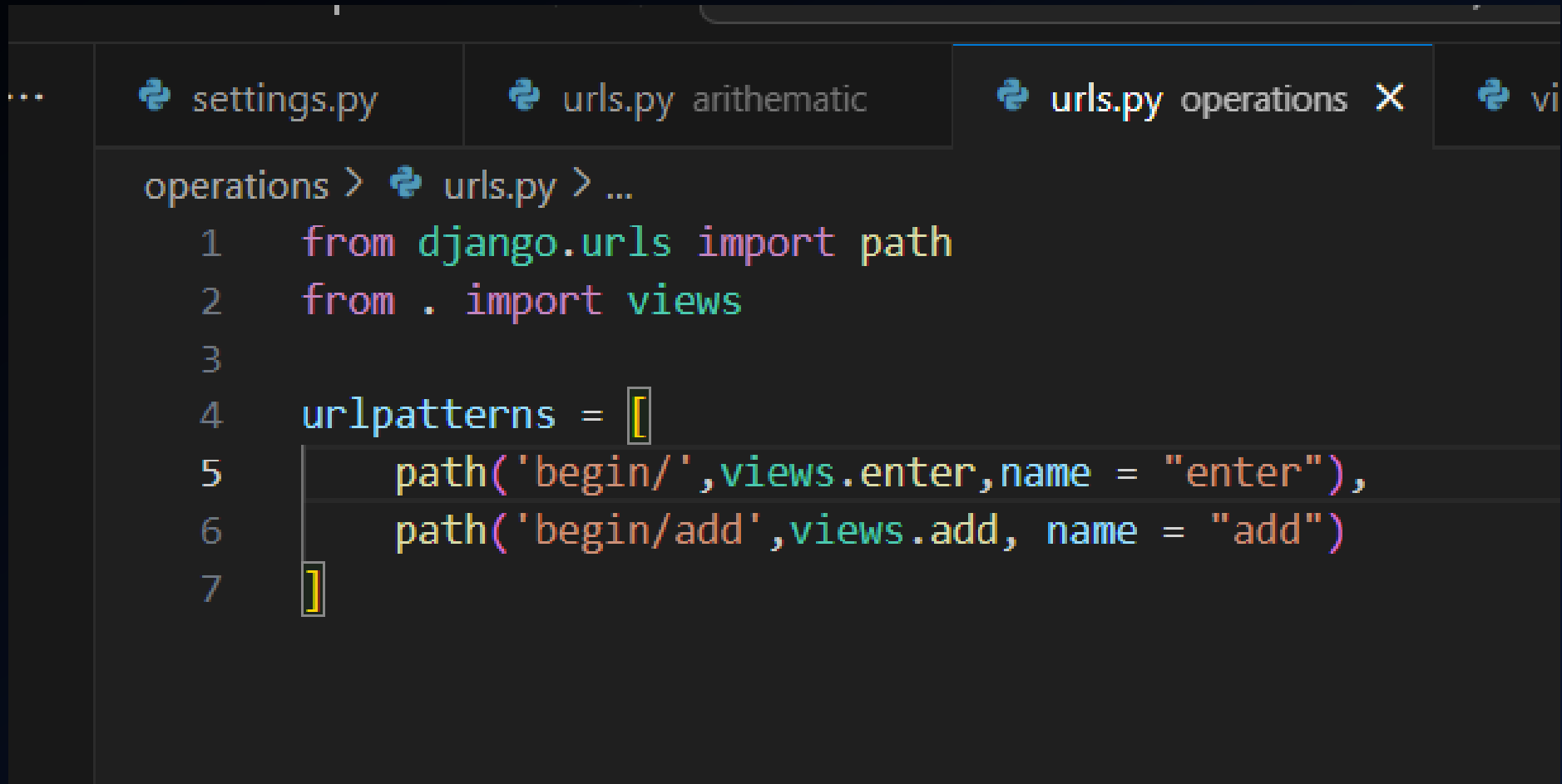
```
25  # SECURITY WARNING: don't run with debug
26  DEBUG = True
27
28  ALLOWED_HOSTS = []
29
30
31  # Application definition
32
33  INSTALLED_APPS = [
34      'django.contrib.admin',
35      'django.contrib.auth',
36      'django.contrib.contenttypes',
37      'django.contrib.sessions',
38      'django.contrib.messages',
39      'django.contrib.staticfiles',
40      'operations'
41  ]
42
43  MIDDLEWARE = [
```



## PROJECT – URLS.PY

```
16      """
17      from django.contrib import admin
18      from django.urls import path, include
19
20      urlpatterns = [
21          path('admin/', admin.site.urls),
22          path('operations/', include('operations.urls'))
23      ]
24
```

## APP – URLS.PY



The image shows a code editor window with four tabs: settings.py, urls.py arithmetic, urls.py operations (which is the active tab), and vi. The active tab displays the following Python code:

```
operations > urls.py > ...  
1  from django.urls import path  
2  from . import views  
3  
4  urlpatterns = [  
5      path('begin/', views.enter, name = "enter"),  
6      path('begin/add', views.add, name = "add")  
7  ]
```



# VIEWS.PY

```
..  settings.py  urls.py arithmetic  urls.py operations  views.py X

operations > views.py > add
1  from django.shortcuts import render
2
3  # Create your views here.
4  def enter(request):
5      return render(request, 'Operands.html',{})
6
7  def add(request):
8      val1 = int(request.GET['num1'])
9      val2 = int(request.GET['num2'])
10     res = val1+val2
11     return render(request, "result.html", {"result": res})
```

# OPERANDS.HTML

operations > templates > <> operands.html >  html

```
7    </head>
8    <body>
9    <form action="add" method="get">
10        {% csrf_token %}
11        Enter First operand: <input type="number", name="num1"><br/><br/>
12        Enter Second operand: <input type="number", name="num2"><br/>
13        <br/>
14        <input type="submit" name="new">
15        <input type="reset" name="new">
16    </form>
17 </body>
18 </html>
```

# STYLE1.CSS

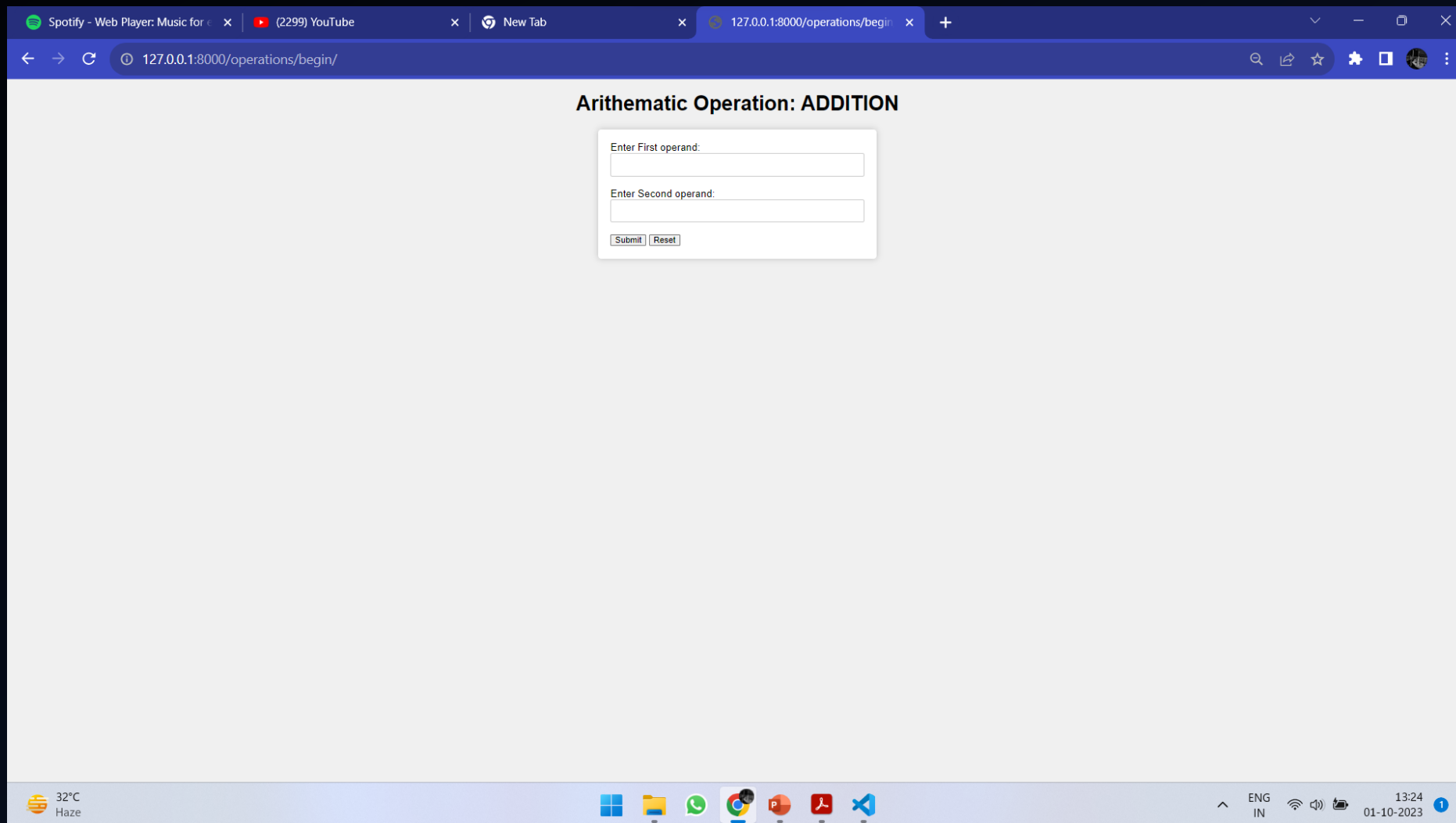
```
settings.py  urls.py arithmetic  urls.py operations  +
operations > static > # style1.css > h1
1  body {
2      font-family: Arial, sans-serif;
3      background-color: #f0f0f0;
4      margin: 0;
5      padding: 0;
6  }
7
8  h1 {
9      text-align: center;
10     margin-top: 20px;
11 }
12
13 form {
14     max-width: 400px;
15     margin: 0 auto;
16     padding: 20px;
17     background-color: #fff;
18     box-shadow: 0 0 10px rgba(0, 0, 0, 0.2);
19     border-radius: 5px;
20 }input[type="submit"]{
21     text-align: center;
22 }
23
24 .input-container {
25     margin-bottom: 100px;
26     text-align: center;
27 }
28
29 input[type="number"] {
30     width: 100%;
31     padding: 10px;
32     border: 1px solid #ccc;
33     border-radius: 3px;
34 }
35
36
```

## RESULT.HTML

```
operations > templates > <> result.html
```

```
1  Solution: {{result}}
```

# BROWSER



# BROWSER

New Tab x 127.0.0.1:8000/operations/begin x +

## Arithmetic Operation: ADDITION

Enter First operand:

Enter Second operand:

127.0.0.1:8000/operations/begin x Spotify - Web Player: Music for

← → ↻ ⓘ 127.0.0.1:8000/operations/begin/add?csrfmiddle

## Solution: 100

