

Lab 11



Django Forms (Part 2)

Student Name: Joud Alblaihed

University ID: 422205657

Subject: CS 471

Section number: 3526

Doctor Name: Bshayer Alkhalifa

Task 1

Code

The screenshot displays two separate instances of the PyCharm IDE, each showing a different project structure and code editor.

Project Structure (Left):

- File
- Edit
- Selection
- View
- Go
- Run
- ... ↗
- CSRF
- EXPLORER
- forms.py U M
- models.py M M
- tests.py
- urls.py M M
- wsgi.py
- static
- libraryproject1
- bookmodule
- task1.html
- task2.html
- task3.html
- includes
- layouts
- bookmodule
- username
- myview
- __pycache__
- int_.py
- arg.py
- wsgi.py
- urls.py
- wsgi.py
- manage.py
- myview
- OUTLINE
- TIMELINE

Code Editor (Left):

```
libraryproject > bookmodule > forms.py > ...
```

```
1 from django import forms
2 from .models import Book, Student, Address
3
4
5 class BookForm(forms.ModelForm):
6     class Meta:
7         model = Book
8         fields = ['title', 'author']
9
10
11 class StudentForm(forms.ModelForm):
12     class Meta:
13         model = Student
14         fields = '__all__'
15
16
17 class AddressForm(forms.ModelForm):
18     class Meta:
19         model = Address
20         fields = '__all__'
```

Project Structure (Right):

- File
- Edit
- Selection
- View
- Go
- Run
- ... ↗
- CSRF
- EXPLORER
- forms.py U M
- models.py M M
- list_students.html U U
- views.py M M
- urls.py M M
- wsgi.py
- static
- libraryproject1
- bookmodule
- task1.html
- task2.html
- task3.html
- includes
- layouts
- bookmodule
- username
- myview
- __pycache__
- int_.py
- arg.py
- wsgi.py
- urls.py
- wsgi.py
- manage.py
- myview
- OUTLINE
- TIMELINE

Code Editor (Right):

```
libraryproject > bookmodule > bookmodule_student > ...
```

SQL Editor:

ID	NAME	AGE	ADDRESS_ID
1	Aisha	25	1
2	Rahil	23	1
3	Sara	20	1
4	Mona	22	2
5	Yousaf	24	2
6	Ism	23	3
7	Khalid	21	3
8	Noor	22	3
9	Nur	20	3
10	Layla	19	3
11	Salem	25	4
...

```
INSERT INTO "bookmodule_student" ("name", "age", "address_id") VALUES (?, ?, ?)
```

SQL Editor Fields:

Column	Type	Nullable	Default
name	TEXT	NONNULL	NULL
age	NUMBER	NONNULL	NULL
address_id	NUMBER	NONNULL	NULL

Task 1 Code

The screenshot shows a code editor with several tabs open. The tabs include: list_students.html, addstudent.html, editstudent.html, views.py, and urls.py. The left sidebar displays the project structure under 'EXPLORER' with nodes like 'libraryproject1', 'apps', 'bookmodule', 'static', and 'templates'. The bottom navigation bar includes 'OUTLINE' and 'TIMELINE'.

The screenshot shows the 'list.html' template file. It contains HTML and CSS code for displaying a list of students. Key parts include a table structure with columns for Name, Age, Address, and Delete, and a 'thead' section with a caption and style definitions.

```
<thead>
    <tr>
        <th>Name</th>
        <th>Age</th>
        <th>Address</th>
        <th>Delete</th>
    </tr>
</thead>
<tbody>
    <tr>
        <td>{{ student.name }}</td>
        <td>{{ student.age }}</td>
        <td>{{ student.address }}</td>
        <td><a href="{{ url_for('edit_student', student.id) }}>Edit</a><br/><a href="{{ url_for('delete_student', student.id) }}>Delete</a></td>
    </tr>
</tbody>
```

The screenshot shows the 'list.html' view file. It defines a 'list' function that renders the 'list.html' template. It also handles POST requests to 'list_books' and 'list_books2' URLs, which render 'list_books.html' and 'list_books2.html' respectively.

```
def list(request):
    return render(request, 'bookmodule/list_students.html', {'students': students})

def addstudent(request):
    if request.method == 'POST':
        form = StudentForm(request.POST)
        if form.is_valid():
            form.save()
            return redirect('list_students')
        else:
            form = StudentForm(instance=student)
    return render(request, 'bookmodule/addstudent.html', {'form': form})
```

The screenshot shows the 'edit.html' template file. It contains HTML and CSS code for editing a student. It includes a form with fields for name, age, and address, and buttons for 'Edit' and 'Delete'.

```
<form method="POST">
    {{ form }}<br/>
    <input type="submit" value="Edit"/>
</form>
```

The screenshot shows the 'edit.html' view file. It defines an 'edit' function that gets a student by ID and renders the 'edit.html' template. It handles POST requests to update the student's information.

```
def editstudent(request, id):
    student = get_object_or_404(Student, id=id)
    if request.method == 'POST':
        form = StudentForm(request.POST, instance=student)
        if form.is_valid():
            form.save()
            return redirect('list_students')
        else:
            form = StudentForm(instance=student)
    return render(request, 'bookmodule/editstudent.html', {'form': form})
```

Task 1 Output

The image displays three screenshots of a web application interface, likely a Django project, showing a list of students and an edit student page.

Screenshot 1: List of Students Page

This screenshot shows a table of student records with columns: Name, Age, Address, edit, and delete. Each row contains a "edit" link and a "delete" link under the respective columns.

Name	Age	Address	edit	delete
Aisha	21	Buraidah	edit this student	delete this student
Fahad	23	Buraidah	edit this student	delete this student
Sara	20	Buraidah	edit this student	delete this student
Mona	22	Unaizah	edit this student	delete this student
Yousef	24	Unaizah	edit this student	delete this student
Lama	23	Unaizah	edit this student	delete this student
Khalid	21	Riyadh	edit this	delete this

Screenshot 2: List of Students Page

This screenshot shows a table of student records with columns: Name, Age, Address, edit, and delete. Each row contains a "edit" link and a "delete" link under the respective columns.

Name	Age	Address	edit	student	student	
Layla	19	Riyadh	edit this student	delete this student	edit this student	delete this student
Salem	25	Jeddah	edit this student	delete this student	edit this student	delete this student
Nada	22	Jeddah	edit this student	delete this student	edit this student	delete this student
Reem	23	Dammam	edit this student	delete this student	edit this student	delete this student
Majid	20	Dammam	edit this student	delete this student	edit this student	delete this student
Amal	21	Dammam	edit this student	delete this student	edit this student	delete this student
Joud	23	Buraidah	edit this student	delete this student	edit this student	delete this student

Screenshot 3: Edit Book Page

This screenshot shows an edit form for a student named Joud Abtish. The form fields are: Name (Joud Abtish), Age (23), and Address (Buraidah). There is also an "Edit" button and a note at the bottom stating "This page was created by joud".

Home List Books About

Name:

Age:

Address:

This page was created by joud

Task 1 Output

List of Students Page

127.0.0.1 8000/books/students/list/

			student	student
Layla	19	Riyadh	edit this student	delete this student
Salem	25	Jeddah	edit this student	delete this student
Nada	22	Jeddah	edit this student	delete this student
Reem	23	Dammam	edit this student	delete this student
Majid	20	Dammam	edit this student	delete this student
Amal	21	Dammam	edit this student	delete this student
Joud Alblaihed	23	Buraidah	edit this student	delete this student

This page was created by joud

List of Students Page

127.0.0.1 8000/books/students/list/

			student	student
Hassan	20	Riyadh	edit this student	delete this student
Layla	19	Riyadh	edit this student	delete this student
Salem	25	Jeddah	edit this student	delete this student
Nada	22	Jeddah	edit this student	delete this student
Reem	23	Dammam	edit this student	delete this student
Majid	20	Dammam	edit this student	delete this student
Amal	21	Dammam	edit this student	delete this student

This page was created by joud

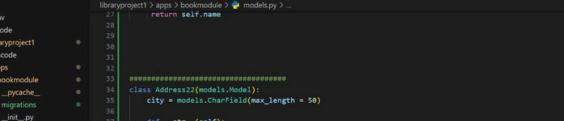
List of Students Page

127.0.0.1 8000/books/students/list/

			student	student
Layla	19	Riyadh	edit this student	delete this student
Salem	25	Jeddah	edit this student	delete this student
Nada	22	Jeddah	edit this student	delete this student
Reem	23	Dammam	edit this student	delete this student
Majid	20	Dammam	edit this student	delete this student
Amal	21	Dammam	edit this student	delete this student
Joud Alblaihed	23	Buraidah	edit this student	delete this student

This page was created by joud
127.0.0.1 8000/books/student/delete/16/

Task 2



The screenshot shows a PyCharm interface with multiple tabs open. The main code editor displays `models.py` from a Django application named `bookmodule`. The `models.py` file contains definitions for three models: `Address`, `Student`, and `Card`. Each model has attributes like `name` (CharField), `age` (IntegerField), and `address` (ManyToManyField). The `__str__` methods are overridden to return the name of the object. Below the code editor, the `views.py` file is partially visible, showing imports for `render` and `reverse`, and definitions for `list_students` and `addstudent` functions.

```
libraryproject1 | apps | bookmodule | models.py > ...
  +-- forms.py M, M
  +-- migrations
  +-- __init__.py
  +-- admin.py
  +-- apps.py
  +-- models.py M, M
  +-- tests.py
  +-- urls.py
  +-- views.py
> static
  +-- templates
    +-- bookmodule
      +-- aboutus.html
      +-- addbook.html U
      +-- addbook2.html U
      +-- addstudent.html U
      +-- student2.html U

  +-- EXPLORER
    +-- C5471
      +-- env
        +-- venv
      +-- myproject1
        +-- __pycache__
          +-- __init__.py
        +-- migrations
        +-- __init__.py
        +-- admin.py
        +-- apps.py
        +-- models.py M, M
        +-- tests.py
        +-- urls.py
        +-- views.py
      +-- static
      +-- templates
        +-- bookmodule
          +-- aboutus.html
          +-- addbook.html U
          +-- addbook2.html U
          +-- addstudent.html U
          +-- student2.html U

  +-- OUTLINE
```

The screenshot shows the PyCharm IDE interface with a Django project named 'libraryproject1'. The left sidebar displays the project structure:

- EXPLORER
- ... (with a dropdown arrow)
- CSRF
- venv
- vscode
- libraryproject1
 - __init__.py
 - bookmodule
 - forms.py
 - models.py
 - views.py
 - migrations
 - __pycache__
 - templates
 - bookmodule
 - aboutus.html
 - addbook.html
 - addbookedit.html
 - addstudent.html
- OUTLINE
- TIMELINE

The main editor window shows the following code snippets:

```
libraryproject1 > apps > bookmodule > forms.py -> ...  
class StudentForm(forms.ModelForm):  
    class Meta:  
        model = Student2  
        fields = "__all__"  
  
class AddressForm(forms.ModelForm):  
    class Meta:  
        model = Address2  
        fields = "__all__"
```

The screenshot shows the PyCharm IDE interface with the following details:

- File Structure (Left):** The project is named "CS4871". It contains a "libraryproject" folder with "lib" and "apps" subfolders. The "bookmodule" app has "tests.py", "views.py", and "static" folders. The "templates" folder for "bookmodule" contains "aboutus.html", "addbook.html", "editbook.html", "editstudent2.html", "editstudent.html", and "list_books.html".
- Code Editor (Right):** The file "bookmodule\templates\bookmodule\editstudent2.html" is open. The code is a Django template with the following content:

```
<style>
    .td {
        padding: 8px;
    }
    caption {
        font-size: 1.2em;
        margin: 10px;
        text-align: center;
        font-weight: bold;
    }
</style>
{% endblock stylesheets %}

{% block content %}
<form method="POST">
    {{ csrf_token }}
    {{ form.as_p }}
    <button type="submit">Add</button>
</form>
{% endblock content %}
```

Task 2

The screenshot shows a Windows desktop environment with a Python development setup. The taskbar at the bottom includes icons for File Explorer, Command Prompt, Python, and a browser. The main window is a code editor with the following details:

- File Tree (EXPLORER):** Shows a project structure under 'libraryproject':
 - apps
 - bookmodule
 - tests.py
 - views.py
 - static
 - bookmodule
 - about.html
 - addbook.html
 - addbook2.html
 - addstudent.html
 - editbook.html
 - editbook2.html
 - editstudent.html
 - editstudent2.html

- Terminal (POWERED BY libspike):** Displays the command line interface for Django migrations:

```
(djangope) PS C:\Users\juda\Desktop\CS471\libraryproject> python manage.py makemigrations bookmodule
Migrations for 'bookmodule':
  - Create model Address22
  + Create model Student22
(djangope) PS C:\Users\juda\Desktop\CS471\libraryproject> python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, bookmodule, contenttypes, sessions, usermodule
Migrations for 'bookmodule':
  - apps(bookmodule).migrations.V0001_address22_student22.py
  + Create model Address22
  + Create model Student22
(djangope) PS C:\Users\juda\Desktop\CS471\libraryproject> python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, bookmodule, contenttypes, sessions, usermodule
```

The screenshot shows the Microsoft Visual Studio Code interface. On the left, the Explorer sidebar displays a file tree with a 'bookmodule' folder containing several Python files (task11.py, task22.py, task33.py, task44.py) and a 'libraryproject' folder. The 'Taskbar' view shows multiple tabs for 'editstudent2.html' and 'db.sqlite3'. In the center, a terminal window is open with the command 'python bookmodule\student22.py', which has just completed successfully. On the right, the Database Explorer sidebar shows a connection to 'db.sqlite3' with a table 'bookmodule_student22'. The table has columns 'id', 'name', and 'age', with data entries: id 1, name Alice, age 20; id 2, name Bob, age 21; and id 3, name Charlie, age 22.



The screenshot shows a code editor with Python and Jinja2 template code. The code includes logic for handling student add and edit operations, including form rendering and validation.

```
models.py M.M          list_students2.html U.U      addstudent2.html U.U      editstudent2.html U.U      views.py M.P
libraryproject > apps > templates > bookmodule > addstudent2.html ...  
8 <table>  
9   <tr>  
10    <td>  
11      <input type="text" name="name" value="John Doe" />  
12    </td>  
13  </tr>  
14</table>  
15  
16 <td>  
17   <input type="text" name="age" value="25" />  
18 </td>  
19</table>  
20  
21 <td>  
22   <input type="text" name="email" value="john.doe@example.com" />  
23 </td>  
24</table>  
25  
26 <td>  
27   <input type="text" name="address" value="123 Main Street" />  
28 </td>  
29</table>  
30  
31 <td>  
32   <input type="text" name="phone" value="555-1234" />  
33 </td>  
34</table>  
35  
36 <td>  
37   <input type="text" name="course" value="Mathematics" />  
38 </td>  
39</table>  
40  
41 <td>  
42   <input type="text" name="gpa" value="3.5" />  
43 </td>  
44</table>  
45  
46 <caption>  
47   <font size="1.2em">  
48     margin: 10px;  
49     text-align: center;  
50     font-weight: bold;  
51   </font>  
52 </caption>  
53  
54 <table border="1" style="width: 100%; border-collapse: collapse;">  
55   <tr>  
56     <td style="width: 10%; padding: 5px; text-align: right;">  
57       <input type="button" value="Add New" />  
58     </td>  
59     <td style="width: 10%; padding: 5px; text-align: right;">  
60       <input type="button" value="Edit" />  
61     </td>  
62     <td style="width: 10%; padding: 5px; text-align: right;">  
63       <input type="button" value="Delete" />  
64     </td>  
65   </tr>  
66 </table>
```

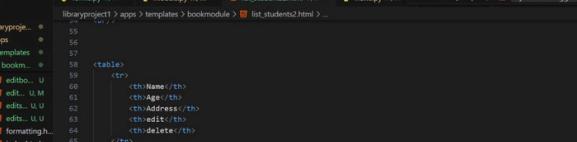
The screenshot shows the Microsoft Visual Studio Code interface with the following details:

- File Explorer:** Shows a project structure with files like `index.html`, `bookmodule.py`, and `__init__.py`.
- Database Explorer:** Shows a connection to `disqlite3` and a table `bookmodule_address22` with columns `id` and `city`.
- Code Editor:** Displays Python code for inserting data into the `bookmodule_address22` table.
- Terminal:** Shows a command-line interface with several database insert statements.

```
INSERT INTO "bookmodule_address22" ('id', 'city') VALUES (1, 'New York')
INSERT INTO "bookmodule_address22" ('id', 'city') VALUES (2, 'Los Angeles')
INSERT INTO "bookmodule_address22" ('id', 'city') VALUES (3, 'Chicago')
```

```
1 -- database: c:\Users\jouda\Desktop\CS471\libra
  1   -- database: c:\Users\jouda\Desktop\CS471\libra
  2   -- database: c:\Users\jouda\Desktop\CS471\libra
  3   -- database: c:\Users\jouda\Desktop\CS471\libra
  4   -- database: c:\Users\jouda\Desktop\CS471\libra
  5   -- database: c:\Users\jouda\Desktop\CS471\libra
  6   -- database: c:\Users\jouda\Desktop\CS471\libra
  7   -- database: c:\Users\jouda\Desktop\CS471\libra
```

Task 2



```
File Edit Selection View Go Run ... ← → ⌘ CS411 Python Debugger: Djens.vip Explorer ... forms.py M models.py M list_students2.html U views.py M Python Debugger: Djens.vip ... C:\Users\CS411\librayproject> apps> templates> bookmodule> list_students2.html <...>
libraryproject> apps> templates> bookmodule> list_students2.html <...>
55
56
57
58 <table>
59   <tr>
60     <th>Name</th>
61     <th>Age</th>
62     <th>Address</th>
63     <th>Edit</th>
64     <th>Delete</th>
65   </tr>
66   {% for student in students %}
67   <tr>
68     <td>{{ student.name }}</td>
69     <td>{{ student.age }}</td>
70     <td>
71       {% for addr in student.address.all %}
72         {{ addr.city }}{% if not forloop.last %}, {% endif %}
73       {% endfor %}
74     </td>
75     <td><a href="{% url 'edit_student' student.id %}">edit this student</a></td>
76     <td><a href="{% url 'delete_student2' student.id %}">delete this student</a></td>
77   {% endfor %}
78 </table>
79
80 </div>
81
82 <% endblock content %>
```

The screenshot shows the MySQL Workbench interface. A connection named 'CS41' is selected. In the left sidebar, there are two library projects: 'libraryproject1' and 'libraryproject2'. The 'libraryproject1' node is expanded, showing its structure: 'bookmodule' (with 'book' and 'student22_address' tables), 'usermodule' (with 'user' and 'student22_address' tables), and 'includes', 'layouts', and 'usermodule' files. The 'student22_address' table from the 'bookmodule' is currently selected in the query editor. The SQL query is:

```
INSERT INTO `bookmodule`.`student22_address` (`id`, `student22_id`, `address22_id`) VALUES (?, ?, ?)
```

The results pane shows the following data being inserted:

id	student22_id	address22_id
1	1	1
2	2	2
3	3	3

Task 2 Output

The image displays four screenshots of a web application interface, likely a Django project, showing student management features.

Add Book Page: A form for adding a new student. Fields include Name (Joud), Age (23), and Address (dropdown menu with options: New York, Los Angeles, Chicago). An "Add" button is present at the bottom.

List of Students Page: A table listing students with columns: Name, Age, Address, edit, and delete. The data is as follows:

Name	Age	Address	edit	delete
Alice	20	New York	edit this student	delete this student
Bob	21	Los Angeles	edit this student	delete this student
Charlie	22	Chicago	edit this student	delete this student

Edit Book Page: A form for editing an existing student (Joud Alhaleh). Fields include Name (Joud Alhaleh), Age (23), and Address (dropdown menu with options: New York, Los Angeles, Chicago). An "Edit" button is present at the bottom.

List of Students Page: A table listing students with columns: Name, Age, Address, edit, and delete. The data is as follows:

Name	Age	Address	edit	delete
Alice	20	New York	edit this student	delete this student
Bob	21	Los Angeles	edit this student	delete this student
Charlie	22	Chicago	edit this student	delete this student
Joud	23	New York	edit this student	delete this student

Task 2 Output

List of Students Page

127.0.0.1:8000/books/students/list2/

Home List Books About Add New Book

This page was created by joud

Name	Age	Address	edit	delete
Alice	20	New York	edit this student	delete this student
Bob	21	Los Angeles	edit this student	delete this student
Charlie	22	Chicago	edit this student	delete this student
Joud Alblaihd	23	New York	edit this student	delete this student

List of Students Page

127.0.0.1:8000/books/students/list2/

Home List Books About Add New Book

This page was created by joud

Name	Age	Address	edit	delete
Alice	20	New York	edit this student	delete this student
Bob	21	Los Angeles	edit this student	delete this student
Charlie	22	Chicago	edit this student	delete this student

List of Students Page

127.0.0.1:8000/books/students/list2/

Home List Books About Add New Book

This page was created by joud

127.0.0.1:8000/books/students/delete2/4/

Name	Age	Address	edit	delete
Alice	20	New York	edit this student	delete this student
Bob	21	Los Angeles	edit this student	delete this student
Charlie	22	Chicago	edit this student	delete this student
Joud Alblaihd	23	New York	edit this student	delete this student

Task 3

Code

The screenshot displays a dual-monitor setup for Python development, specifically using Django frameworks.

Left Monitor (CSAT1 Workspace):

- Explorer View:** Shows the project structure under 'CSAT1'. It includes a 'librappo...' folder containing 'apps' (with 'bookmodule'), 'migrations', 'pychar...', 'settings.py', 'urls.py', and 'views.py'. The 'bookmodule' folder contains 'models.py', 'views.py', and 'templat...', along with 'media' and 'static' folders.
- Code Editor:** Displays the content of 'list_profiles.html'. The code uses Django's template language to render a table of profiles, each with a 'Name' column, a 'Photo' column, an 'Edit' link, and a 'Delete' link. It also includes logic for handling profile creation and deletion.

Right Monitor (Python Debugger: Dj... Workspace):

- Code Editor:** Displays the content of 'models.py'. The code defines a 'User' model with fields for 'username', 'password', 'first_name', 'last_name', 'email', and 'is_staff'. It also includes imports for 'os.path' and 'Path' from 'pathlib'.

Task 3

Code

```
SELECT * FROM bookmodule_profile
```

```
id INTEGER NOT NULL
name varchar(100) NOT NULL
photo varchar(100) NOT NULL
```

```
INSERT INTO "bookmodule_profile" ("id", "name", "photo") VALUES (?, ?, ?)
```

id	1	AS TEXT NUMERIC BLOB NULL DEFAULT
name	t	AS TEXT NUMERIC BLOB NULL DEFAULT
photo	1	AS TEXT NUMERIC BLOB NULL DEFAULT

```
<td>
    <input type="text" value="Current Name" name="name" />
</td>
```

```
<td>
    <input type="file" name="photo" />
</td>
```

```
def list_profiles(request):
    profiles = Profile.objects.all()
    return render(request, 'bookmodule/list_profiles.html', {'profiles': profiles})
```

```
def add_profile(request):
    if request.method == 'POST':
        form = ProfileForm(request.POST, request.FILES)
        if form.is_valid():
            form.save()
            return redirect('list_profiles')
        else:
            form = ProfileForm()
    return render(request, 'bookmodule/add_profile.html', {'form': form})
```

```
def edit_profile(request, id):
    profile = get_object_or_404(Profile, id=id)
    if request.method == 'POST':
        form = ProfileForm(request.POST, request.FILES, instance=profile)
        if form.is_valid():
            form.save()
            return redirect('list_profiles')
        else:
            form = ProfileForm(instance=profile)
    return render(request, 'bookmodule/edit_profile.html', {'form': form})
```

```
def delete_profile(request, id):
    profile = get_object_or_404(Profile, id=id)
```

```
class ProfileForm(forms.ModelForm):
    class Meta:
        model = Profile
        fields = ['name', 'photo']
```

Task 3

Code

VS Code interface showing the code for `views.py` and `urls.py` in the `bookmodule` app.

`views.py` content:

```
librapp>1 apps> bookmodule > views.py M, M
librapp>2
librapp>3
librapp>4 path('lab9/task22/', views.task22),
librapp>5 path('lab9/task33/', views.task33),
librapp>6 path('lab9/task44/', views.task44),
librapp>7 path('lab9/part1/listbook', views.listbook, name="list_books"),
librapp>8 path('lab9/part1/addbook', views.addbook, name="add_book"),
librapp>9 path('lab9/part1/editbook<int:id>', views.editbook, name="edit_book"),
librapp>10 path('lab9/part2/listbook2', views.listbook2, name="list_books2"),
librapp>11 path('lab9/part2/addbook2', views.addbook2, name="add_book2"),
librapp>12 path('lab9/part2/editbook2<int:id>', views.editbook2, name="edit_book2"),
librapp>13 path('lab9/part2/deletebook2<int:id>', views.deletebook2, name="delete_book2"),
librapp>14 path('students/list/', views.list_students, name="list_students"),
librapp>15 path('students/add/', views.addstudent, name="add_student"),
librapp>16 path('students/edit<int:id>', views.editstudent, name="edit_student"),
librapp>17 path('students/delete<int:id>', views.deletestudent, name="delete_student"),
librapp>18 path('students/list2/', views.list2, name="list_students2"),
librapp>19 path('students/add2/', views.addstudent2, name="add_student2"),
librapp>20 path('students/edit2<int:id>', views.editstudent2, name="edit_student2"),
librapp>21 path('students/delete2<int:id>', views.deletestudent2, name="delete_student2"),
librapp>22 path('profiles/list', views.list_profiles, name="list_profiles"),
librapp>23 path('profiles/add/', views.add_profile, name="add_profile"),
librapp>24 path('profiles/edit<int:id>', views.edit_profile, name="edit_profile"),
librapp>25 path('profiles/delete<int:id>', views.delete_profile, name="delete_profile"),
```

`urls.py` content:

```
librapp>1
librapp>2
librapp>3
librapp>4
librapp>5
librapp>6
librapp>7
librapp>8
librapp>9
librapp>10
librapp>11
librapp>12
librapp>13
librapp>14
librapp>15
librapp>16
librapp>17
librapp>18
librapp>19
librapp>20
librapp>21
librapp>22
librapp>23
librapp>24
librapp>25
librapp>26
librapp>27
librapp>28
librapp>29
librapp>30
librapp>31
```

VS Code interface showing the code for `urls.py` in the `libraryproject`.

`urls.py` content:

```
libraryproj>1
libraryproj>2
libraryproj>3
libraryproj>4
libraryproj>5
libraryproj>6
libraryproj>7
libraryproj>8
libraryproj>9
libraryproj>10
libraryproj>11
libraryproj>12
libraryproj>13
libraryproj>14
libraryproj>15
libraryproj>16
libraryproj>17
libraryproj>18
libraryproj>19
libraryproj>20
libraryproj>21
libraryproj>22
libraryproj>23
libraryproj>24
libraryproj>25
libraryproj>26
libraryproj>27
libraryproj>28
libraryproj>29
libraryproj>30
libraryproj>31
```

Task 3 Output

This page was created by joud

Name: joud

Photo: Choose File | profile.png

Add

This page was created by joud

Name Photo edit delete

This page was created by joud

Name: Joud AlBatahi

Photo: Currently profile_photos/profile.png
Change: Choose File | No file chosen

Edit

This page was created by joud

Name Photo edit delete

This page was created by joud