BackEnd Workshop-3

Clarusway

Subject: Django CRUD Operations

Learning Goal

- Practice Django to implement CRUD (Create, Read, Update, Delete) operations.

Introduction

In this workshop, we will create a "Student Register" Django app.

- The root url will be a Student Register form. Users can submit a new student using this form.
- After submission, the user will be linked to "list" page. Or they can go to the list page with a direct link.
- In list page, users will see the "Full Name" and the "Path" of the student. Here there will be "Add New", "Update", and "Delete" buttons.
    - "Add New" will go to the "Student Register" form.
    - "Update" will go to the related students filled form, which can be updated.
    - "Delete" will erase the related student, and continue to show the list page.

Model:

In this project, you need to create a Student model with fields:

- fullname
- number
- mobile
- email

- gender
- path

You can use choices option with gender and path.

Form :

In this project, you need to create a forms.py file including Student form using Student model.

Views:

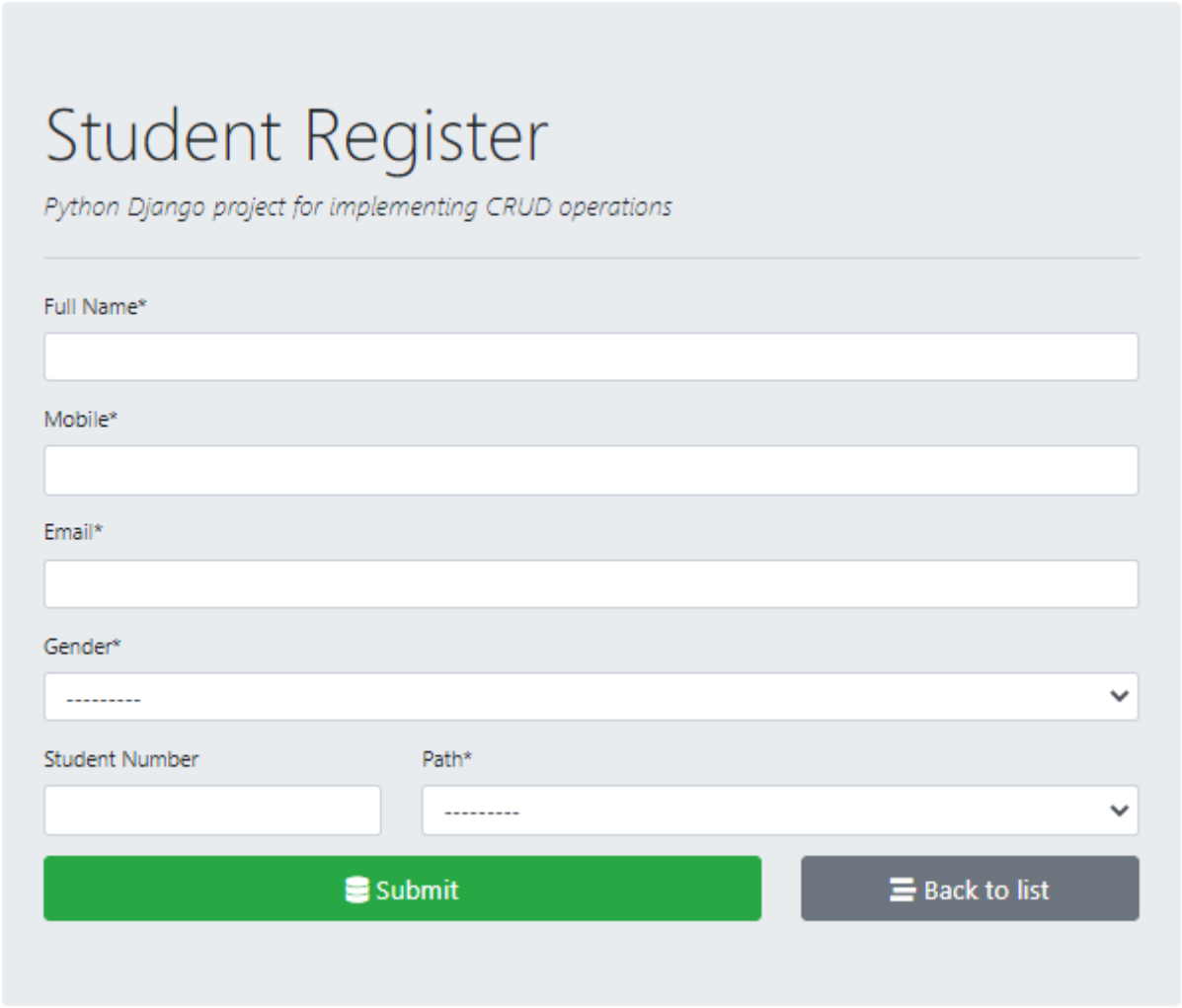
In this project, you need to create three views:

- student_add_update # Including Student form, to create and update student object.
- student_list # To read all student objects on the template.
- student_delete # To delete a student object.

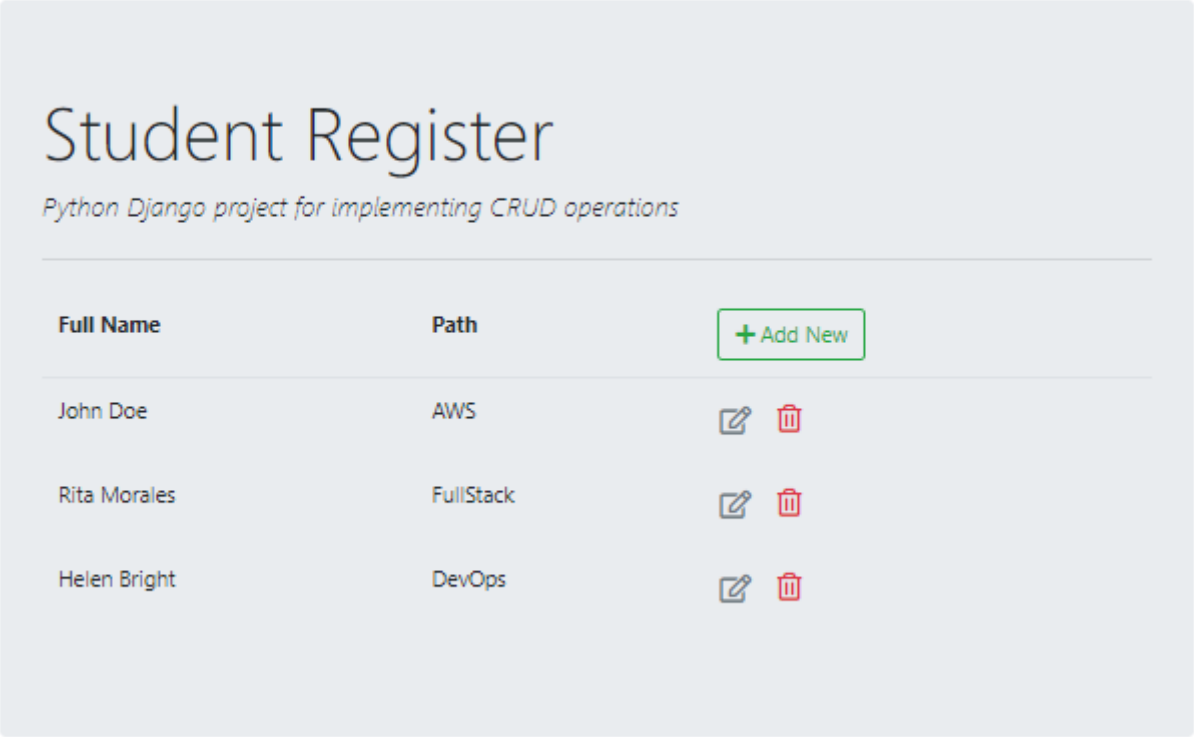
Templates:

In this project, you need to create two templates:

- student_form.html



- student_list.html



And one optional template for base.html.

URLs:

In this project, you need to create three urls:

- '' : root path will return main page with student form.
- '<int:id>/' : While update, the same form will be displayed with student info of related id.
- 'delete/<int:id>/' : delete student.
- 'list/' : list view to show all student objects.

Project Folder Structure

At the end of the project, the folder sturcture will be like:

Code:

Part 1 - Create first project and app

1. Create a working directory, cd to new directory.

2. Create virtual environment as a best practice:

3. Activate scripts:

4. Install django.

5. (Optional) See installed python packages.

6. Create the requirements.txt on your working directory, send your installed packages to this file, requirements file must be up to date.

7. Create project.

8. (Optional) If you created your project wihthout "." at the end, change the name of the project main directory as src to distinguish from subfolder with the same name.

9. Create the application.

10. Go to settings.py and add another line as below under INSTALLED_APPS.

11. Run your project.

Go to http://localhost:8000/ in your browser, and you should see Django rocket, which means you successfully created project.

The install worked successfully! Congratulations!

Part 2 - Create Student Model

1. Go to models.py and create your student model.

2. See how to use choices option.

3. Discuss using field options like EmailField.

4. Manage migrations.

Part 3 - Create Student Form

1. Create forms.py under student_register app and create your student form.

2. Discuss about __init__ method. How we can overwrite ModelForm?

Part 4 - Create Views and URLs

1. Go to views.py under "student_register" directory, and create your views.

2. Discuss about redirect function.

3. Include URL path of the new app to the project url list, go to urls.py

4. Add urls.py file under student_register directory.

Part 5 - Install Crispy Forms and Create Templates

1. Install crispy forms according to this link:

Crispy Forms Installation

2. Create templates/student_register folder and under that folder, create base.html file as a best practice. Create this base template including bootstrap like below.

3. Discuss using block tag.

4. Under templates/student_register folder create student_form.html as the second template, this will serve Student Form.

5. Under templates/student_register folder create student_list.html as the third template, this will show all students.

6. Run our project again to see our view.

7. Go to http://localhost:8000/ in your browser, and check the functionality of your website. To stop the server use "CTRL + C"

8. If you want to send this project to your Github repo, do not forget to add .gitignore file, and secure your sensitive information like keys storing them locally in .env file and using python-decouple module.

9. Update the requirements.txt file after all your installations!

☺ **Thanks for Attending** ✍

Clarusway