



Tribhuwan University
Institute of Engineering
Thapathali Campus

Subject: Web Application Programming

LAB #3

Lab Report on: Server Side Programming and Django

Submitted By

Name: Srijal Basnet

Roll No: THA080BCT044

Submitted To

Department Of Electronics and Computer Engineering

5th February 2026

OBJECTIVES

To understand the concept and workflow of server-side programming.

To learn the fundamental structure and architecture of the Django framework.

To build a basic CRUD web application using Django.

THEORY

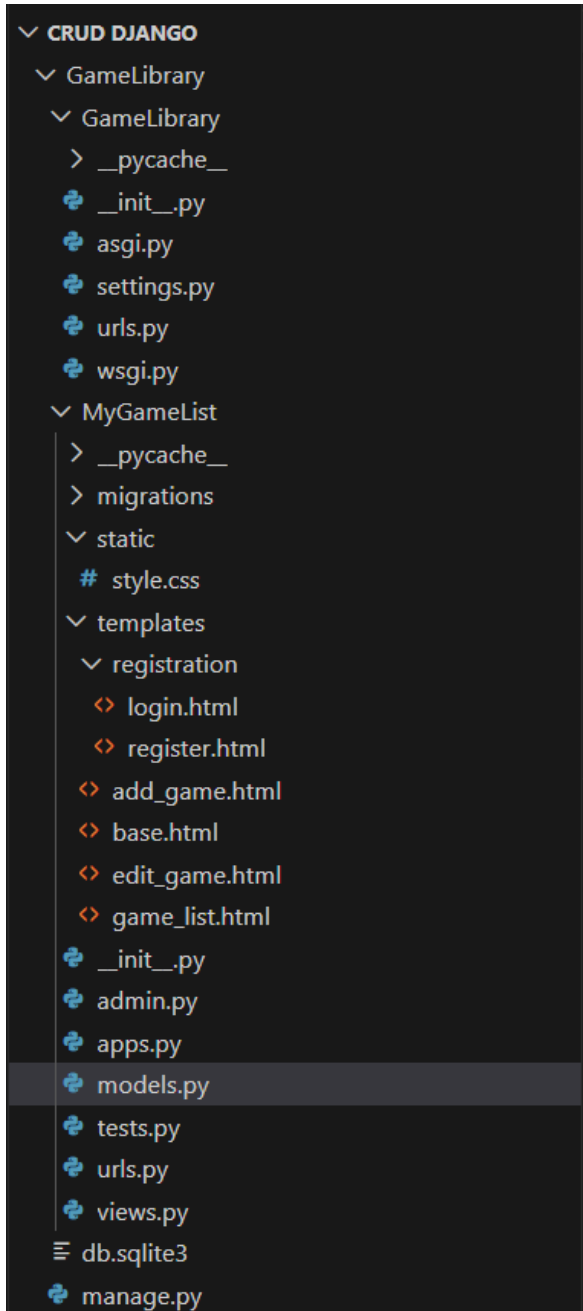
Server-side programming refers to backend development where application logic runs on a web server instead of the user's browser. When a user sends a request through HTTP methods such as GET or POST, the server processes the request, interacts with the database if required, and sends back a response like HTML or JSON. This allows web applications to generate dynamic content based on user input or stored data.

Server-side systems also provide improved security because sensitive logic and database operations remain hidden from the client side. Backend frameworks handle data processing, authentication, and CRUD operations, ensuring structured data management and persistence.

Django is a high-level Python web framework designed to simplify web development by providing built-in tools for routing, database management, authentication, and templating. It follows a structured architecture that separates data models, business logic, and presentation layers, allowing developers to build scalable applications quickly.

In this lab project, I created a simple CRUD Django application called 'My Game List'. The application allows users to add, edit, view, and delete games. Each game entry includes fields such as rating, review, and play status. The application integrates with the RAWG API to retrieve game database information, demonstrating how external APIs can be used alongside Django to enrich application data while maintaining a local database.

Project Folder Structure

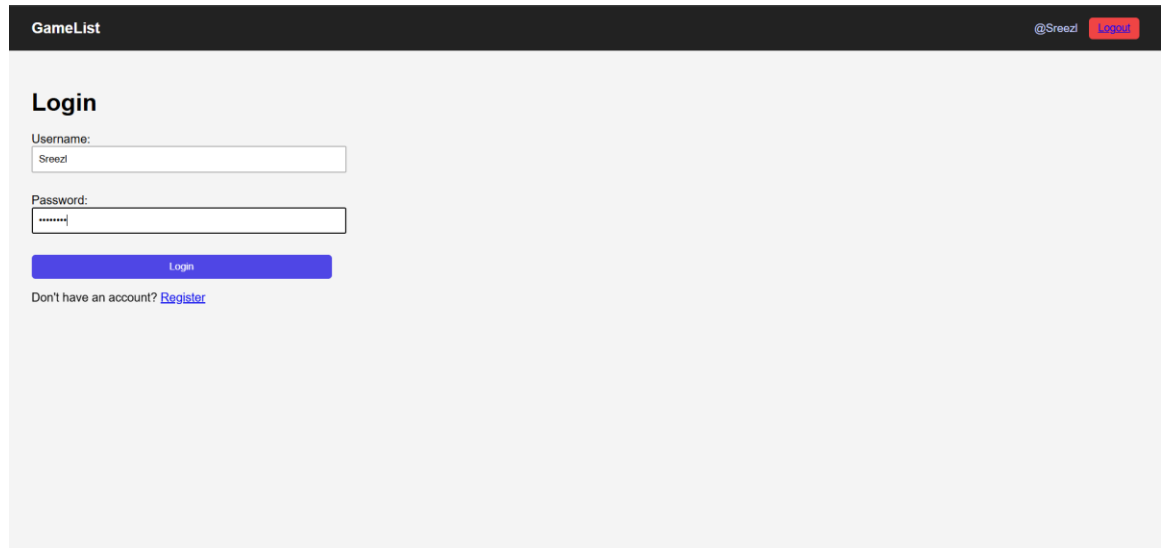


SOURCE CODE

<https://github.com/Srijal-Basnet/My-Game-List-App.git>

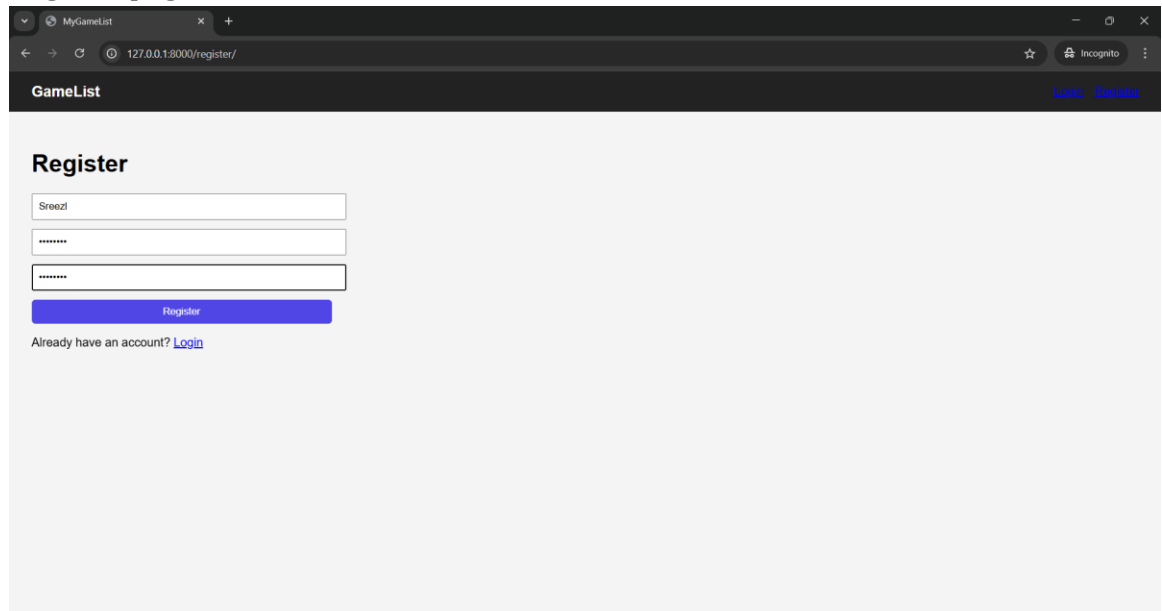
OUTPUT

1. Login page



The screenshot shows the login page of the GameList application. The page has a dark header with the text "GameList" on the left and "@Sreezl Logout" on the right. The main content area is light gray and contains the "Login" section. It features two input fields: "Username:" with the value "Sreezl" and "Password:" with masked characters. Below these fields is a blue "Login" button. At the bottom of the section, there is a link that says "Don't have an account? Register".

2. Register page



The screenshot shows the register page of the GameList application. The browser's address bar indicates the URL "127.0.0.1:8000/register/". The page has a dark header with "GameList" on the left and "Login Register" on the right. The main content area is light gray and contains the "Register" section. It features three input fields: "Username:" with the value "Sreezl", and two "Password:" fields, both with masked characters. Below these fields is a blue "Register" button. At the bottom of the section, there is a link that says "Already have an account? Login".


3. Home Page/ Game list

GameList


@Sreezi [Logout](#)

My Game List


[+ Add Game](#)




Grand Theft Auto V
Action
★ 8/10
Completed
[Edit](#) | [Delete](#)




God of War (2018)
Action
★ 9/10
Completed
[Edit](#) | [Delete](#)




God of War: Ragnarök
Action
★ 10/10
Completed
[Edit](#) | [Delete](#)




Clair Obscur: Expedition 33
RPG
★ 10/10
Playing
[Edit](#) | [Delete](#)




Marvel's Spider-Man Remastered
Casual
★ 10/10
Completed
[Edit](#) | [Delete](#)




Marvel Rivals
Multiplayer
★ 8/10
Playing
[Edit](#) | [Delete](#)




Action
★ 8/10
Completed
[Edit](#) | [Delete](#)




Action
★ 9/10
Completed
[Edit](#) | [Delete](#)




Action
★ 10/10
Completed
[Edit](#) | [Delete](#)




Clair Obscur: Expedition 33
RPG
★ 10/10
Playing
[Edit](#) | [Delete](#)




Casual
★ 10/10
Completed
[Edit](#) | [Delete](#)




Multiplayer
★ 8/10
Playing
[Edit](#) | [Delete](#)




PEAK.
Multiplayer
★ 7/10
Completed
[Edit](#) | [Delete](#)




The Last of Us Part I
Story
★ 9/10
Completed
[Edit](#) | [Delete](#)




Ghost of Tsushima Director's Cut
Adventure
★ 8/10
On Hold
[Edit](#) | [Delete](#)



Red Dead Redemption 2
Action
★ 0/10
Planned
[Edit](#) | [Delete](#)



Minecraft
Multiplayer
★ 10/10
Playing
[Edit](#) | [Delete](#)



DOOM (2016)
Shooter
★ 8/10
Completed
[Edit](#) | [Delete](#)

4. Edit Game Page

GameList

@Sreezi [Logout](#)

Edit Game

Marvel's Spider-Man Remastered

Casual

Rating (0-10)
9

Status
Playing

Update

5. Add Game Page

Add Game

Grand Theft Auto V (Action) ★ 4

Gears 5 (Shooter) ★ 3

Tekken 5 (Fighting) ★ 4

DIRT 5 (Racing) ★ 3

Persona 5 (Adventure) ★ 4

Title

Genre

Review

Rating (0-10)

Status

Playing

Add Game

DISCUSSION AND CONCLUSION

The development of this app demonstrated how server-side programming enables dynamic and interactive web applications. Using Django, the application successfully implemented CRUD functionality for managing game entries while maintaining a clear separation between backend logic and frontend templates. The integration of the RAWG API showed how external services can be used to fetch structured data and enhance user experience without manually maintaining a large dataset.

Through this lab, key concepts such as request-response cycles, database interaction using Django ORM, and API integration were explored. Django's built-in features simplified authentication, routing, and data handling, making development faster and more organized. Overall, the project demonstrates that Django provides a reliable and efficient framework for building secure server-side applications with clean architecture and scalable design.