Gamers Portal

IT325 Web Services Final Project

by

Mustapha Kouki

January 2024

TUNIS BUSINESS SCHOOL
UNIVERSITY OF TUNIS

Tunis Business School

Ben Arous,TUNISIA.

2023-2024

Abstract

"Gamers Portal" is a dedicated online space designed for gaming enthusiasts, focusing on user-generated game reviews and ratings. Tailored for the gaming community, the platform engages gamers of all levels, creating a dynamic hub for shared experiences. Central to "Gamers Portal" is the user-driven review system, empowering gamers to express their opinions and contribute to a collective pool of insights. Ratings provide an immediate glimpse into the community's sentiments, fostering an environment where users can discover, discuss, and celebrate their favorite games. Security is a paramount aspect, ensuring a safe and enjoyable space for all gamers. Beyond reviews, the platform seamlessly integrates features for retrieving real-time game information, offering a comprehensive experience for users seeking insights into the latest gaming trends. By prioritizing user interaction, security, and information accessibility, "Gamers Portal" seeks to redefine how the gaming community engages with and contributes to the world of digital gaming.

Contents

Chapter 1

Introduction

For my IT325 Web Services final project, I've developed a Flask-based web application focused on secure game review management. The platform allows users to perform CRUD operations, submitting, updating, and deleting game reviews securely. Leveraging technologies such as Flask, SQLite, Flask-JWT-Extended, and Flask-Bcrypt, the application ensures robust security through JWT for authentication and password hashing. The integration with the RAWG Video Games Database API enriches the user experience with real-time game information. This project showcases a blend of technology stacks and security measures, providing a user-friendly interface for gamers to explore, contribute, and interact securely within the gaming community. The report details the design decisions, architecture, and implementation.

Chapter 2

Technologies and Features

2.1 Flask Contribution

Flask serves as the foundation of my web application. It simplifies the process of building web services by providing a lightweight and modular structure. Flask handles routing, request processing, and response generation, making it an excellent choice for developing web applications.

2.2 Flask-JWT-Extended Contribution

Flask-JWT-Extended enhances the security of your application by implementing JSON Web Tokens for user authentication. JWTs are used to generate access tokens, allowing users to securely access protected routes. This extension simplifies the implementation of JWTs in Flask applications.

2.3 Flask-Bcrypt Contribution

Flask-Bcrypt adds an extra layer of security to your application by providing utilities for password hashing. User passwords are hashed using Bcrypt before being stored in the database. This helps protect sensitive user information and mitigates the risk of password-related security issues.

2.4 Postman Contribution

Postman serves as a pivotal tool for testing and validating API endpoints in the Flask project. It facilitates seamless testing of user registration, login, and access token retrieval processes. Additionally, Postman enables thorough verification of interactions with protected routes, ensuring the security of the application. Its role extends to testing functionalities such as fetching game information, adding, updating, and deleting reviews, ensuring the robustness and correctness of the Flask-based web application.

2.5 Requests Contribution

The requests library is used to make HTTP requests to external APIs. In your application, it interacts with the RAWG Video Games Database API, fetching real-time game information. This external API integration allows your users to access up-to-date data on games, enriching their experience.

2.6 SQLite Contribution

SQLite is employed as the database engine for my application. It offers a simple and file-based approach to managing data. In this case, SQLite is responsible for storing user authentication information, game reviews, and other essential data. Its lightweight nature makes it suitable for embedded systems..

2.7 GIT Contribution

Initialized the Git repository for the project using "git init" and staged all files for the initial commit. The commit, labeled "Initial commit," captured the project's starting state, including essential files such as Requirements.txt, app.py, and data.db. This version control setup enables effective tracking and collaboration during development.

2.8 All the HTTP Methods used

2.8.1 GET Requests

GET /game-info

This endpoint allows users to retrieve detailed information about a specific game. By providing the game's name in the URL path, the server responds with comprehensive details, enriching the user experience with real-time game information.

GET /reviews

This endpoint enables users to fetch reviews for a particular game. By specifying the game name in the URL path, the server returns user-generated reviews, fostering a community-driven environment for sharing gaming experiences.

GET /average rating

This endpoint provides the average rating for a given game. Users can access this information by making a GET, obtaining insights into the overall community sentiment toward the game.

2.8.2 POST Requests

POST /register

Register in the API after specifying a valid username and password in the body of the request.

POST /register

User registration is facilitated through this endpoint. By sending a POST with JSON data containing the username and password, new users can securely create accounts.

POST /login

The request body includes JSON data with the username and password, and upon successful authentication, the server responds with an access token.

POST /add review

Users can add new reviews by making a POST request. The request body should include JSON data with the game name, rating, and review. Authentication is ensured by providing the access token in the Authorization header.

2.8.3 PUT Requests

PUT /update review

This endpoint allows users to update existing reviews. By sending a PUT request, users can modify the rating and review. The request body includes the new rating and review, along with the username, password, and access token for authentication.

2.8.4 DELETE Requests

DELETE /delete review

Users can delete their reviews using this endpoint. The request body should include the username, password, and access token for authentication

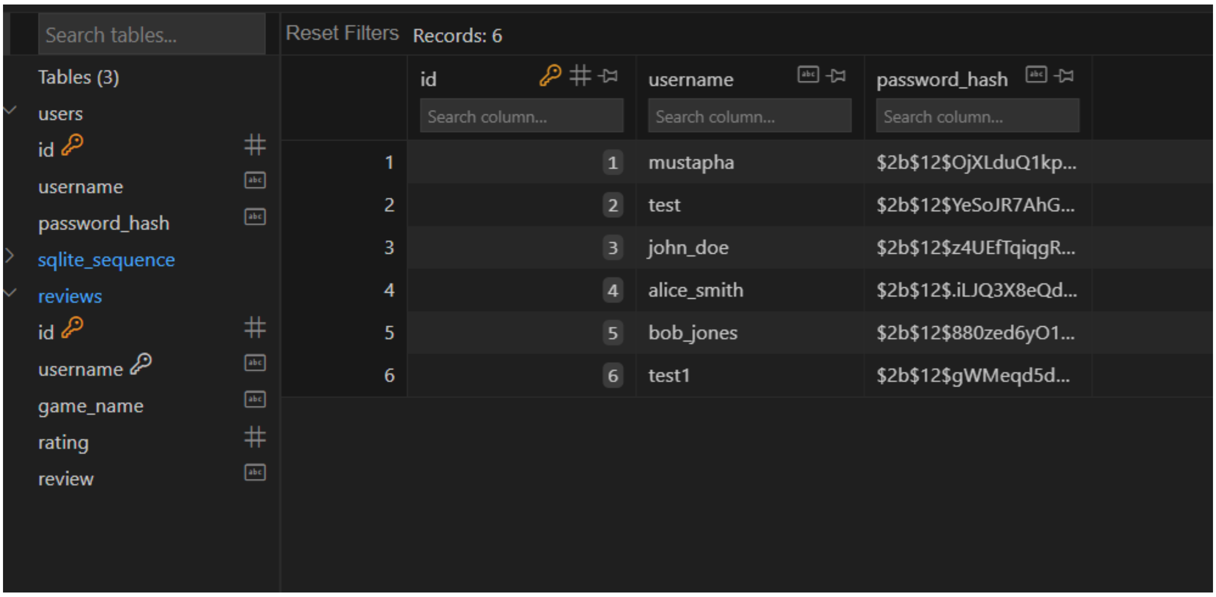
2.9 Database Tables Structure

Users Table

The users table is central to the application, storing user authentication details. The structure includes:
userid: A unique identifier for each user.
username: The chosen username for the user.
password: Hashed password for enhanced security.

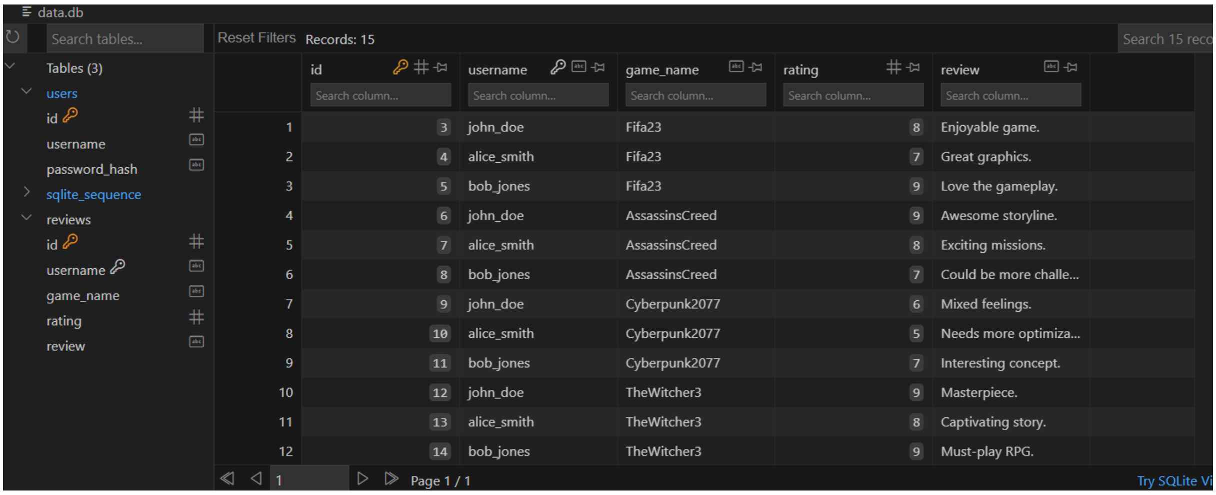
Reviews Table

Game reviews contribute significantly to the platform's dynamic content. The reviews table structure includes:
id: A unique identifier for each review.
userid: Foreign key linking to the users table.
game name: The name of the reviewed game.
rating: The user-assigned rating for the game.
review text: The textual content of the review.



	id	username	password_hash
1	1	mustapha	$2b$12$OjXLduQ1kp...
2	2	test	$2b$12$YeSoJR7AhG...
3	3	john_doe	$2b$12$z4UEfTqiqgR...
4	4	alice_smith	$2b$12$.iLJQ3X8eQd...
5	5	bob_jones	$2b$12$880zed6yO1...
6	6	test1	$2b$12$gWMeqd5d...

Figure 2.1: Users Table.



	id	username	game_name	rating	review
1	3	john_doe	Fifa23	8	Enjoyable game.
2	4	alice_smith	Fifa23	7	Great graphics.
3	5	bob_jones	Fifa23	9	Love the gameplay.
4	6	john_doe	AssassinsCreed	9	Awesome storyline.
5	7	alice_smith	AssassinsCreed	8	Exciting missions.
6	8	bob_jones	AssassinsCreed	7	Could be more challe...
7	9	john_doe	Cyberpunk2077	6	Mixed feelings.
8	10	alice_smith	Cyberpunk2077	5	Needs more optimiza...
9	11	bob_jones	Cyberpunk2077	7	Interesting concept.
10	12	john_doe	TheWitcher3	9	Masterpiece.
11	13	alice_smith	TheWitcher3	8	Captivating story.
12	14	bob_jones	TheWitcher3	9	Must-play RPG.

Figure 2.2: Reviews Table.

Chapter 3

Conclusion

In conclusion, "Gamers Portal" represents a comprehensive web application designed to enhance the gaming community's experience. By leveraging Flask and integrating technologies such as SQLite, Flask-JWT-Extended, Flask-Bcrypt, and external APIs through the Requests library, the project ensures secure user authentication, efficient data storage, and real-time game information retrieval. The user-friendly interface, coupled with features like user-generated reviews and Postman API interactions, fosters a dynamic and engaging space for gamers. Through thoughtful design decisions and careful implementation, this project not only meets the initial objectives but also sets the stage for further innovation and growth within the gaming community.

Appendix A

Some response examples

A.1 GET Requests

A.1.1 GET /game-info/fifa22

```
{"game_info":
{"genres":[{"-":"Sports"}],
"name":"FIFA 22","
platforms":[{"-":"PC"},
{"-":"PlayStation 5"},
{"-":"Xbox One"},
{"-":"PlayStation 4"},
{"-":"Xbox Series S/X"},
{"-":"Nintendo Switch"}],
"release_date":"2021-09-27"}}
```

A.1.2 GET /average-rating/Fifa23

```
{"average_rating":8.0,
"message":"The average rating for Fifa23 is 8.00. Excellent!
 Players highly appreciate the quality and enjoyment provided
 by this game."}
```

A.1.3 GET /reviews/Fifa23

```
{"reviews":
[["john_doe",8,"Enjoyable game."],
["alice_smith",7,"Great graphics."],
["bob_jones",9,"Love the gameplay."]]}
```

A.2 POST Requests

A.2.1 POST /login

```
{"access_token":"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.
eyJmcmVzaCI6ZmFsc2UsImlhdCI6MTcwNTgyOTQzMCwia
nRpIjoiODI5NWFkZDctMWYyNC00YTNjLWIzZDMtZWU
yMjA0NzZkNmFjIiwidHlwZSI6ImFjY2VzcyIsInN1YiI
6InRlc3QxIiwibmJmIjoxNzA1ODI5NDMwLCJjc3JmIjoi
OTdhZDUwY2MtZWEwNy00MTAxLWFkMTItZjYyZDAwYTJkOG
U1IiwiZXhwIjoxNzA1ODMwMzMwfQ.NmK-N3i-p
DJmk8w9FBYSDfib7YlNrCdIBaLQTzB7LtE",
"message":"Login successful!"}
```