Project Report and Reflection: Movie Review API

Deployed Web APP Link: Movie Review API

https://hashimaziz88.pythonanywhere.com/

View the Slide Deck: Movie_Review_API Slides Hashim

https://docs.google.com/presentation/d/1bTdwqfRYv3OmdvDa6qfXhc6zR07JWXpJhn3UcswmrPA/edit?usp=sharing

Project Overview

The Movie Review API project was conceived as a way to combine my passion for cinema with my growing skills in software development. As an avid movie enthusiast and anime fan, I wanted to create a platform that would allow users to search for movies, read and write reviews, and rate films. This project served as both a technical challenge and a personal endeavor to build something I'm genuinely excited about and would use myself.

Technical Implementation

Technologies Used

- **Backend Framework with Views:** Django with Django REST Framework
- **Database:** SQLite
- **External API:** OMDb API for movie data
- **Authentication:** Django REST Framework SimpleJWT (JSON Web Tokens)
- **Deployment:** PythonAnywhere

Key Features Implemented

1. **User Authentication and Registration:** Implemented a secure registration and login system using Django REST Framework's built-in authentication and SimpleJWT for token-based authentication.
2. **Movie Search:** Integrated the OMDb API to allow users to search for movies and retrieve detailed information.
3**. Review System:** Developed CRUD operations for users to create, read, update, and delete movie reviews.
4. **Rating System:** Implemented a star-based rating system for movies.

5. **API Endpoints:** Designed and implemented RESTful API endpoints for all major functionalities.

Challenges Faced

1. **External API Integration:** Integrating the OMDb API presented challenges in handling rate limits and ensuring data consistency between our database and the external source.

Solution: Created a robust error-handling system to manage API failures gracefully.

2. **Authentication System:** Implementing JWT authentication using Django REST Framework SimpleJWT required a deep dive into security best practices and token management. Additionally, handling registration and login with Django REST Framework's built-in authentication system required some adjustments for token-based workflows.

Solution: Thoroughly researched SimpleJWT implementation in Django and utilized Django REST Framework's built-in features for handling user authentication and registration, ensuring security best practices for token storage and transmission.

3. **Database Design:** Designing an efficient database schema that could handle complex relationships between users and reviews was challenging.

Solution: Spent considerable time on the ERD design phase, iterating through different models until finding an optimal structure that balanced performance and flexibility.

4. **Deployment:** Deploying a Django application with a SQLite database on PythonAnywhere presented some initial hurdles.

Solution: Carefully followed PythonAnywhere's documentation as well the error logs and reached out to their support forum when needed. Successfully deployed a fully functional application.

Key Learnings

1. **API Design:** Gained valuable experience in designing a RESTful API, including best practices for endpoint naming, request/response structures, and error handling.

2. **Authentication:** Deepened my understanding of token-based authentication, particularly through Django REST Framework SimpleJWT, and the use of Django REST Framework's built-in authentication and registration features.

3. **External API Integration:** Learned strategies for effectively integrating and managing external API dependencies in a production environment.

4. **Database Optimization:** Improved skills in database design and optimization, particularly in handling relationships and queries in Django ORM.

5. **Deployment:** Enhanced my knowledge of deploying Django applications in a production environment, including managing environment variables and database configurations.

6. **Project Management:** Improved my ability to plan and execute a complex project from conception to deployment, including breaking down tasks and managing timelines.

Reflection on Personal Growth

This project has been a significant milestone in my journey as a software developer. It allowed me to apply the skills I've learned through various courses and programs, including the HyperionDev Full Stack Web Developer program and the ALX Back-End Web Development program.

Working on a project that aligns with my personal interests in movies and anime made the development process both challenging and enjoyable. It reinforced my belief that combining passion with technical skills can lead to more engaging and meaningful work.

The process of building this API has also highlighted areas where I need to improve. For instance, I realized the importance of writing clean, maintainable code from the start, as refactoring became necessary in later stages of the project. This experience has motivated me to focus more on code quality and documentation in future projects.

Future Improvements

While the current version of the Movie Review API is functional and deployed, there are several areas I'd like to improve in the future:

1. Implement more advanced filtering and search capabilities for movies and reviews.
2. Add social features, such as following other users and sharing reviews.
3. Enhance the user interface with a React-based frontend to create a full-stack application.
4. Implement more comprehensive test coverage, including integration and end-to-end tests.
5. Optimize database queries for improved performance with larger datasets.

Conclusion

Developing the Movie Review API has been a rewarding experience that has significantly contributed to my growth as a software developer. It has reinforced my passion for both technology and cinema and has given me a tangible product that showcases my skills and interests.

This project has enhanced my technical abilities, as well as my problem-solving skills and perseverance. The challenges faced and overcome have boosted my confidence in tackling complex software development tasks.

Moving forward, I am excited to continue refining this project and to apply the lessons learned to future endeavors. The experience gained from this project will be invaluable as I progress in my software development career, particularly in my current role as a software engineering recruit at WeThinkCode_.

The journey of creating this API has reaffirmed my belief in the power of combining personal interests with professional skills. It has shown me that with dedication, continuous learning, and a passion for the subject matter, it's possible to create meaningful and useful applications that can benefit not just myself but potentially many others as well.