**Report on.**

**NETFLIX CLONE USING DJANGO FRAME WORK**

**Su**bmitted by:

Tumwebaze Anthony

Under the esteemed guidance of:

Mr. Tonny Engwau



MBARARA UNIVERSITY OF SCIENCE AND TECHNOLOGY

FACULTY OF COMPUTING AND INFORMATICS

DEPARTMENT OF SCIENCE IN COMPUTER SCIENCE

**WEB APPLICATION DEVELOPMENT**

**(NETFLIX CLONE REPORT IN DJANGO)**

TUMWEBAZE ANTHONY 2021/BCS/083/PS

Friday, August 25 2023

Declaration

I, the undersigned, hereby certify that this report titled “**NETFLIX CLONE USING DJANGO**” is my original research work project and I present it without any reservation for external examination. I declare that the work is a result of continuous and consistent efforts .I also acknowledge the guidance and support of my supervisor, Tonny Engwau, from the Department of Computer Science at Mbarara University of Science and Technology.

TUMWEBAZE ANTHONY

2021/BCS/083/PS

**Abstract:**

The purpose of this project was to develop a Netflix clone using the Django framework, replicating key features and functionalities of the popular streaming platform. The project aimed to showcase proficiency in web development with Django, incorporating user authentication, media streaming, content categorization, and user-friendly interface.

**Introduction:**

**1. Research Background**

Creating a Netflix clone using Django is a substantial project that involves various components such as user authentication, content management, video streaming. To develop a comprehensive understanding of the background research needed for this project, I had to first consider the following aspects:

Such as Streaming Technology, Video hosting and CDN(Content Delivery Networks), Database Design , User Authentication and Authorization, Payment Integration , Content Management, User interface and experience , Search and Recommendation Algorithms , Scalability and Performance , Content licensing and Copyright , Deployment and Hosting , Testing and Assurance , Security Considerations Monitoring and Analytics.

**2. The Purpose and Significance of the Project**

The rise of online streaming platforms has transformed the way we consume media content. This project aimed to create a clone of the Netflix platform using the Django web framework. The primary objectives were to implement user registration and authentication, display and categorize media content, and enable streaming of videos.

**3. Technology and Development Environment**

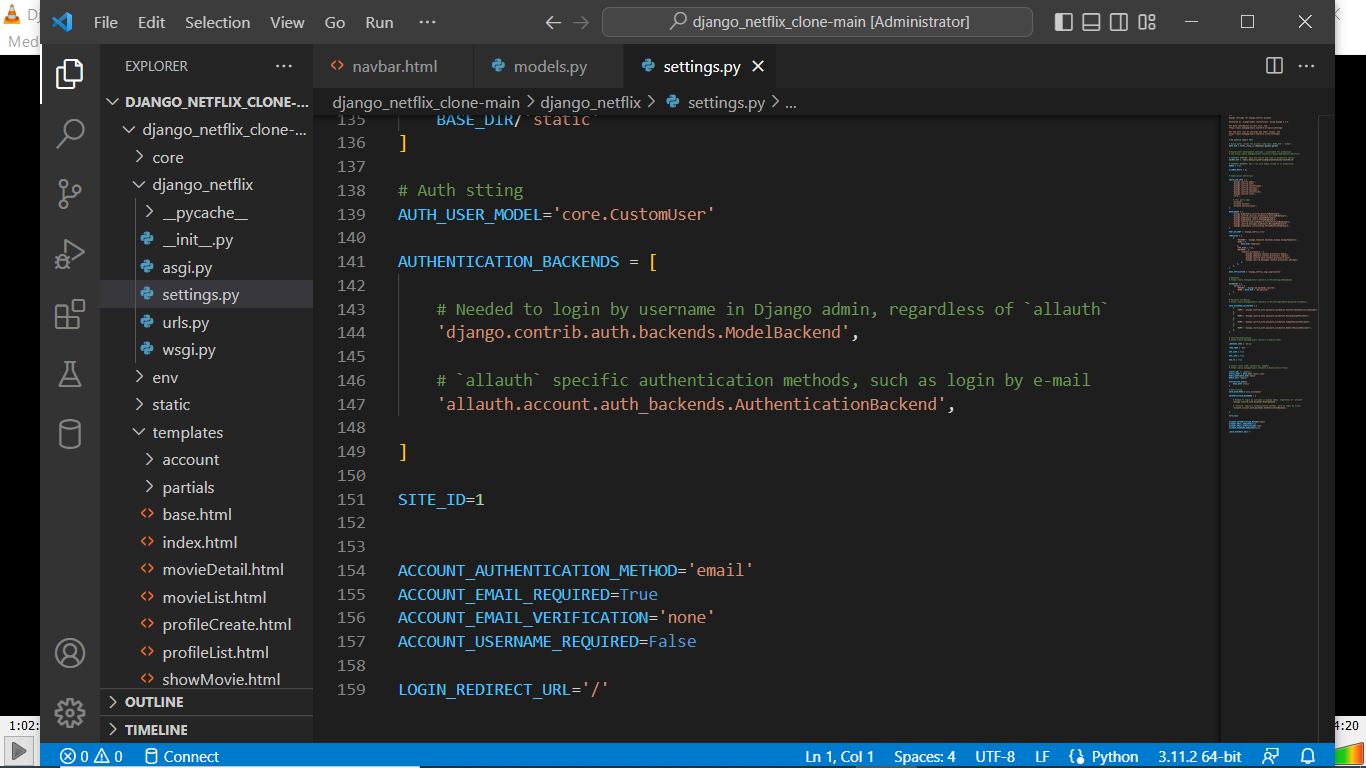
Used Django (A high level Python web frame work that promotes rapid development and clean, pragmatic design) as a web application framework that can create a dynamic and interactive website for the Netflix clone. As the framework should support the creation of web pages, forms, databases, user authentication, etc., that are essential for the functionality of the Netflix clone web application.

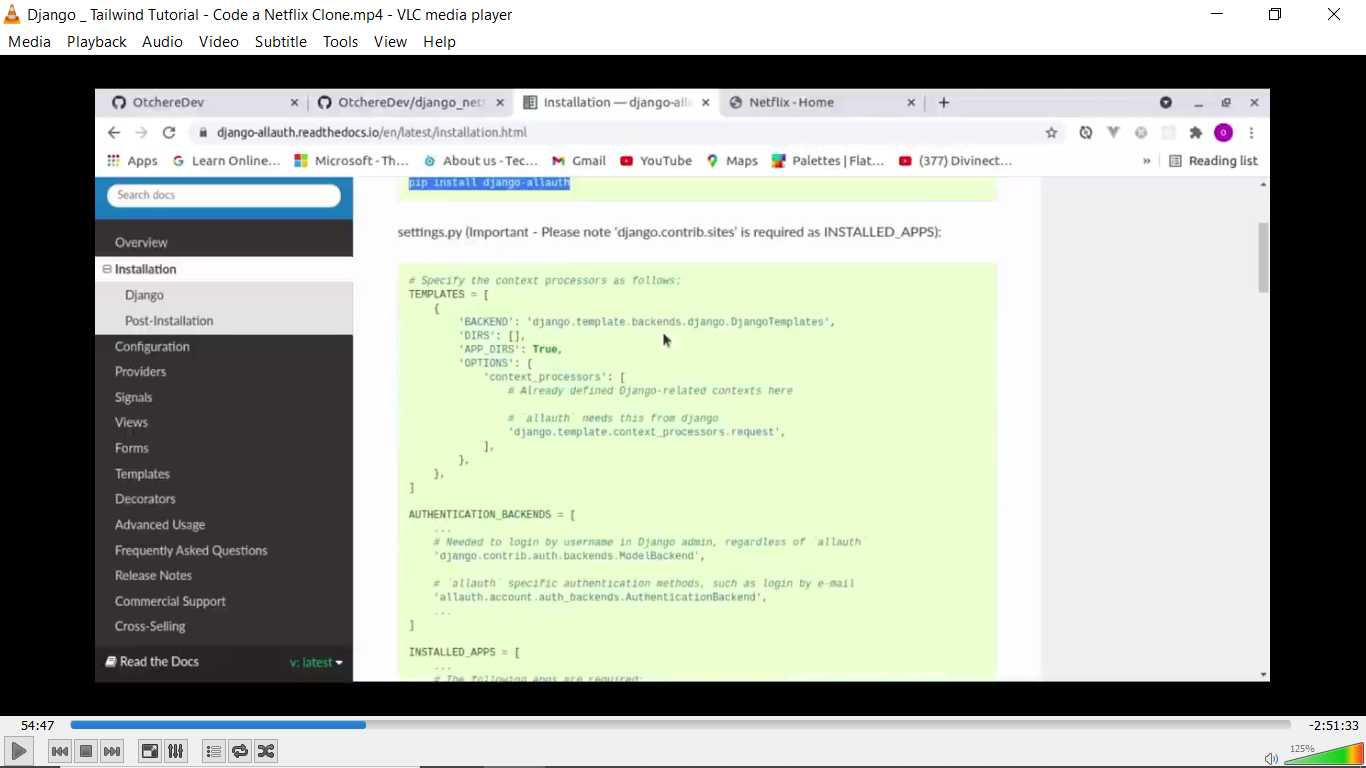
HTML, CSS, JavaScript: used for front end development and enhancing user interface.

SQLite: Default database system for django used to store application data

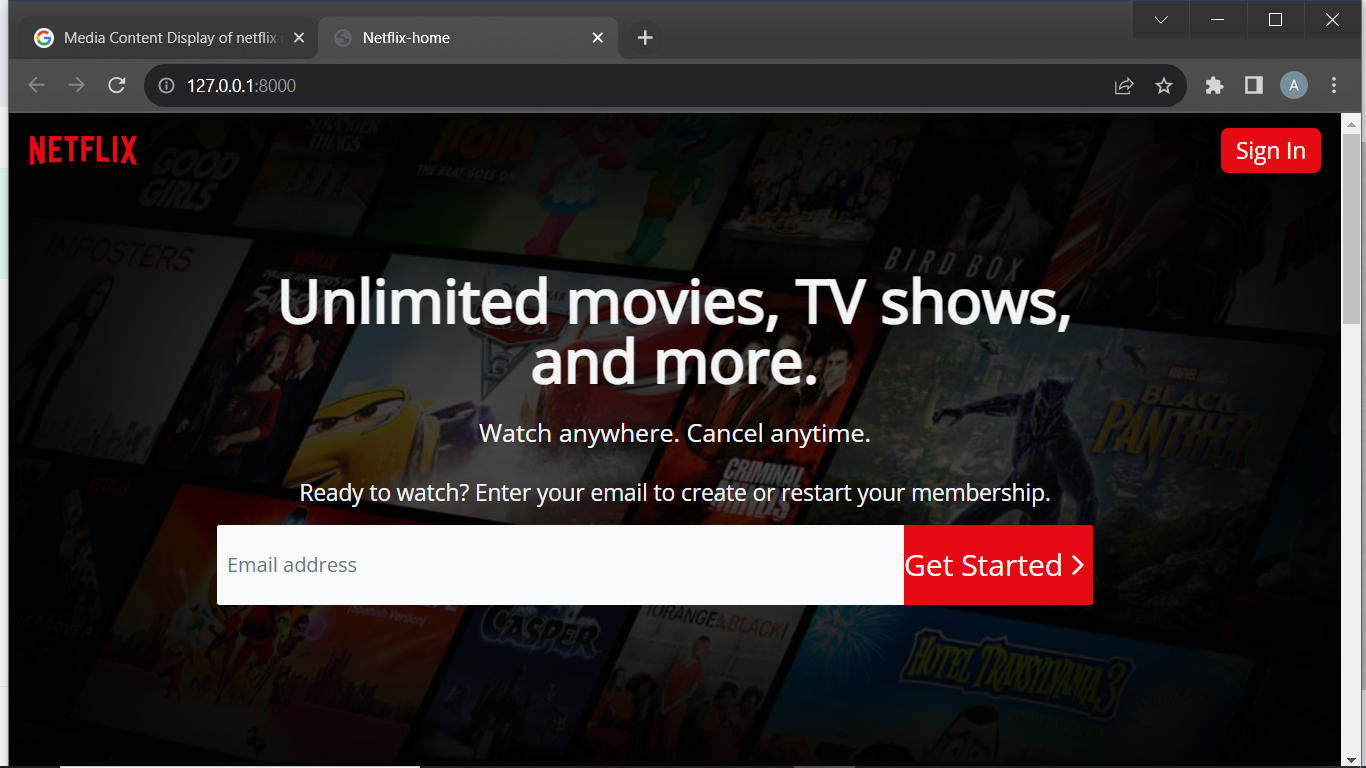
**4. Features implemented**

**4.1 User Authentication:** User registration and authentication were implemented using Django’s built in authentication system.

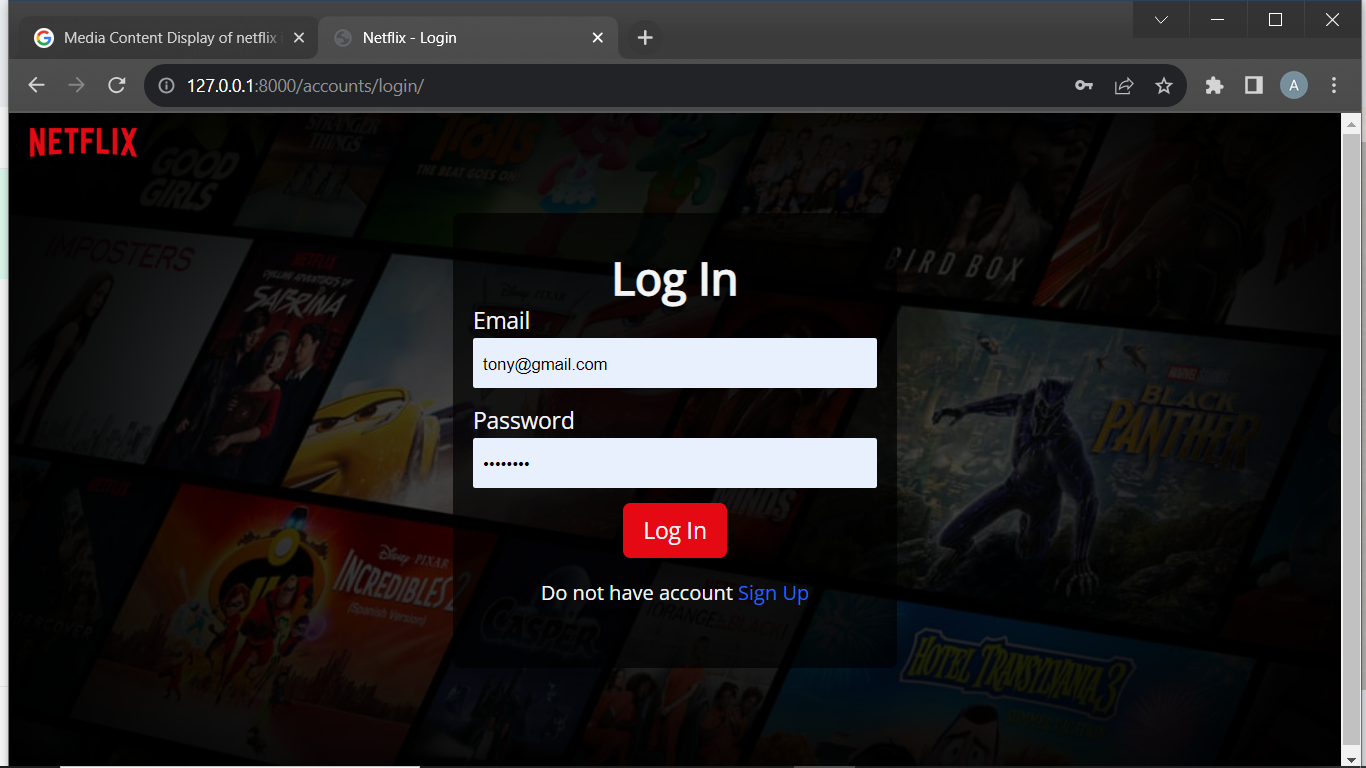




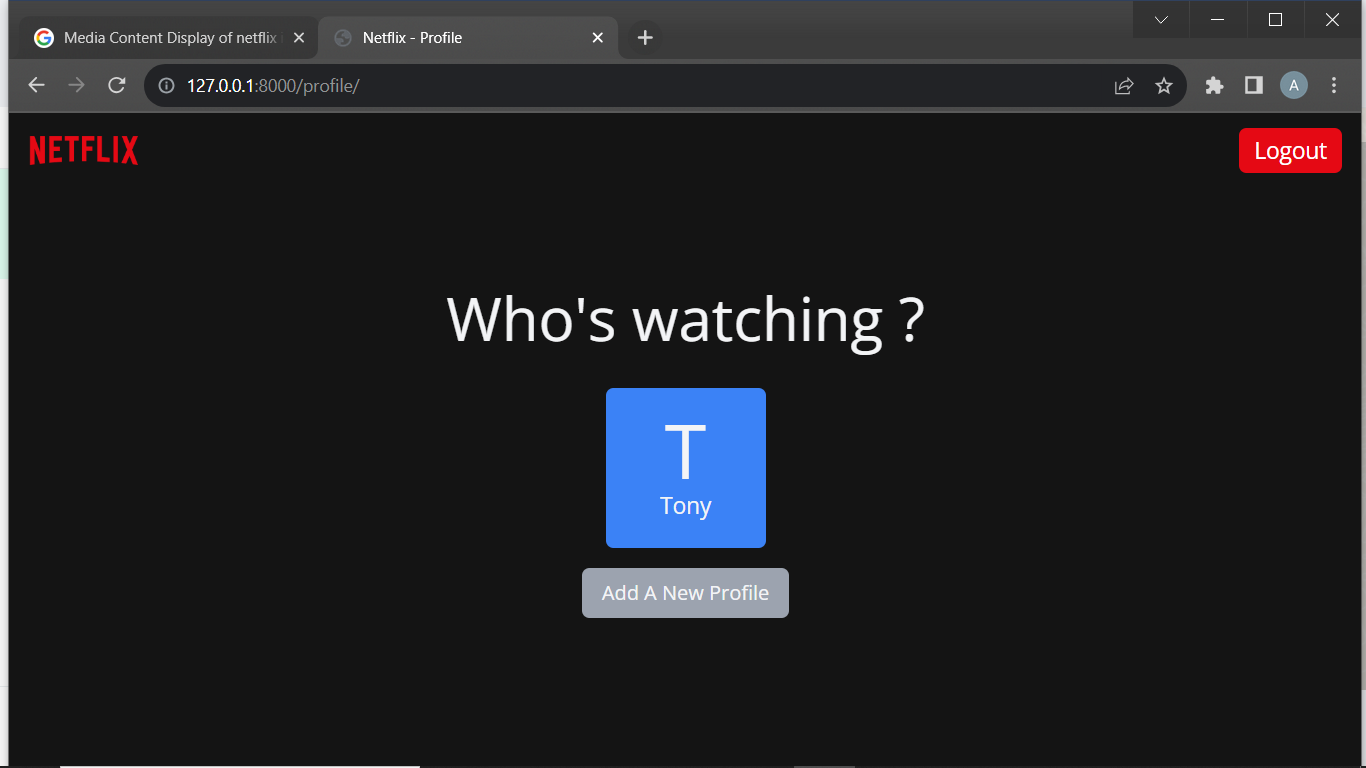
Users can register, login and logout. Passwords were securely hashed before storage.

****

On the login page, the user types in his or her email address (used as username) and password. Once the credentials submitted don’t match, an error message is displayed on top of the page indicating that username and password did not match with what was registered in the database while signing up

****

If the username (email) and password match with the record in the database, the user is redirected to his/her respective dashboard. Below is a ‘Tony’ a user whose credentials where correct and has logged in successfully.

****

**Media Content Display:**

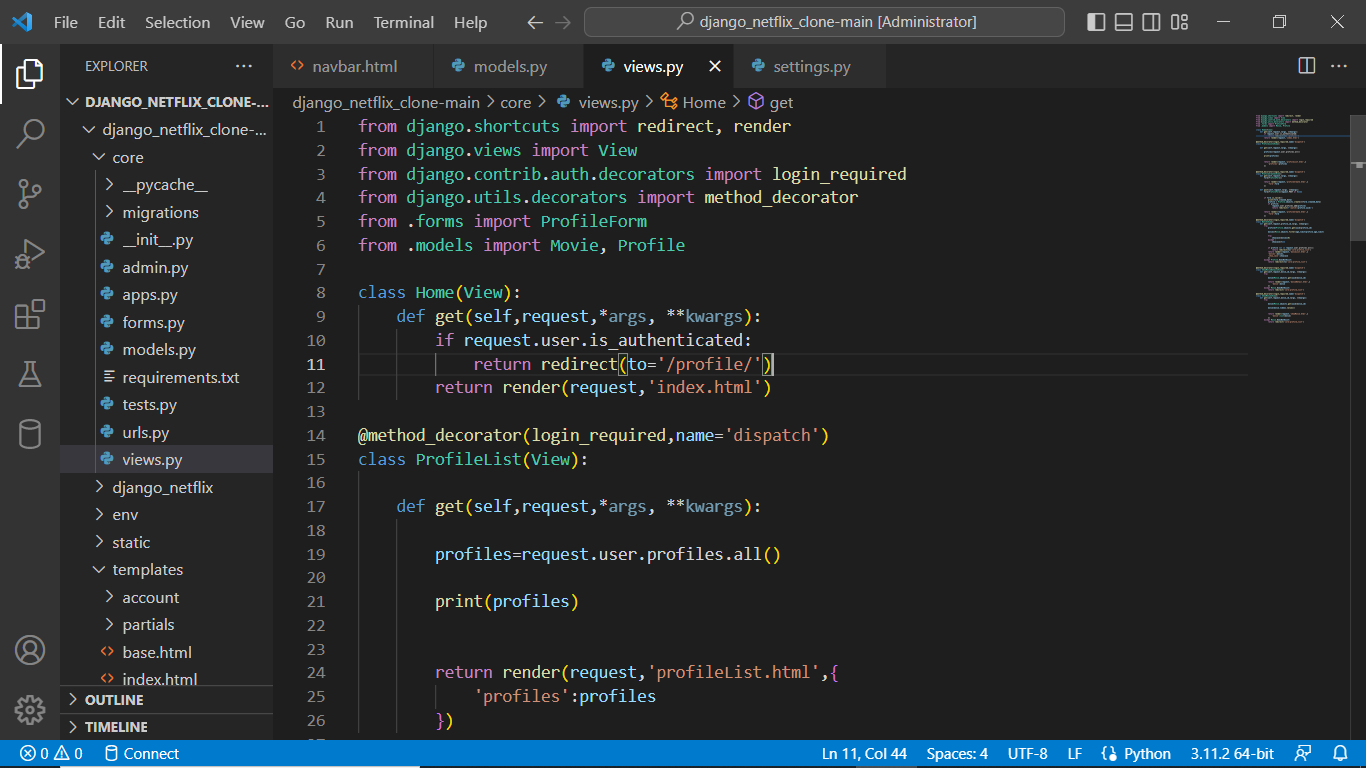
Media content including movies and TV shows were displayed on the platform. Each item had a title, description, genre and a cover image :

**Content Categorization:**

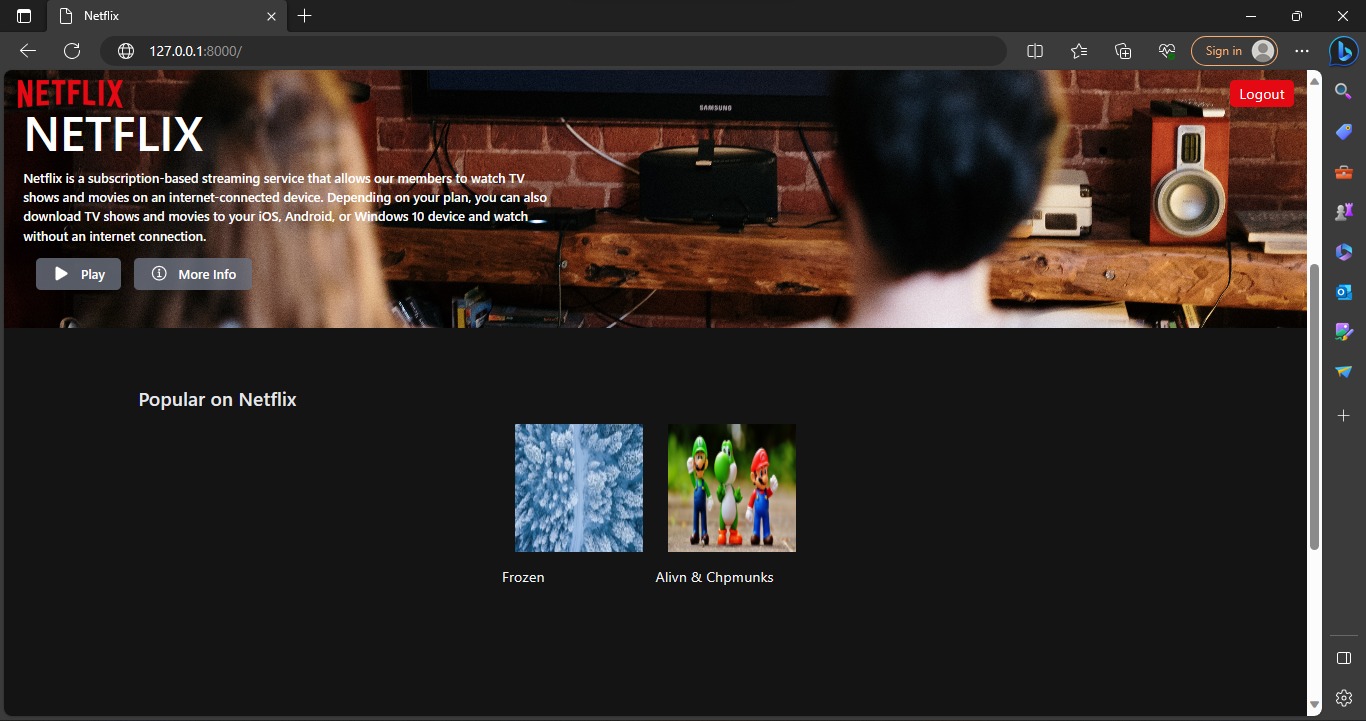
Media content was categorized into different genres (e.g. Action , Drama , Comedy ) to help users discover content of their preference.

**Database:** Utilised Django’s ORM that is used to send data between a database and models in an application. It maps a relation between the database and a model. So, ORM maps object attributes to fields of a table. ORM makes the entire development process fast and error-free. Essentially, it eliminates the need to write SQL code.

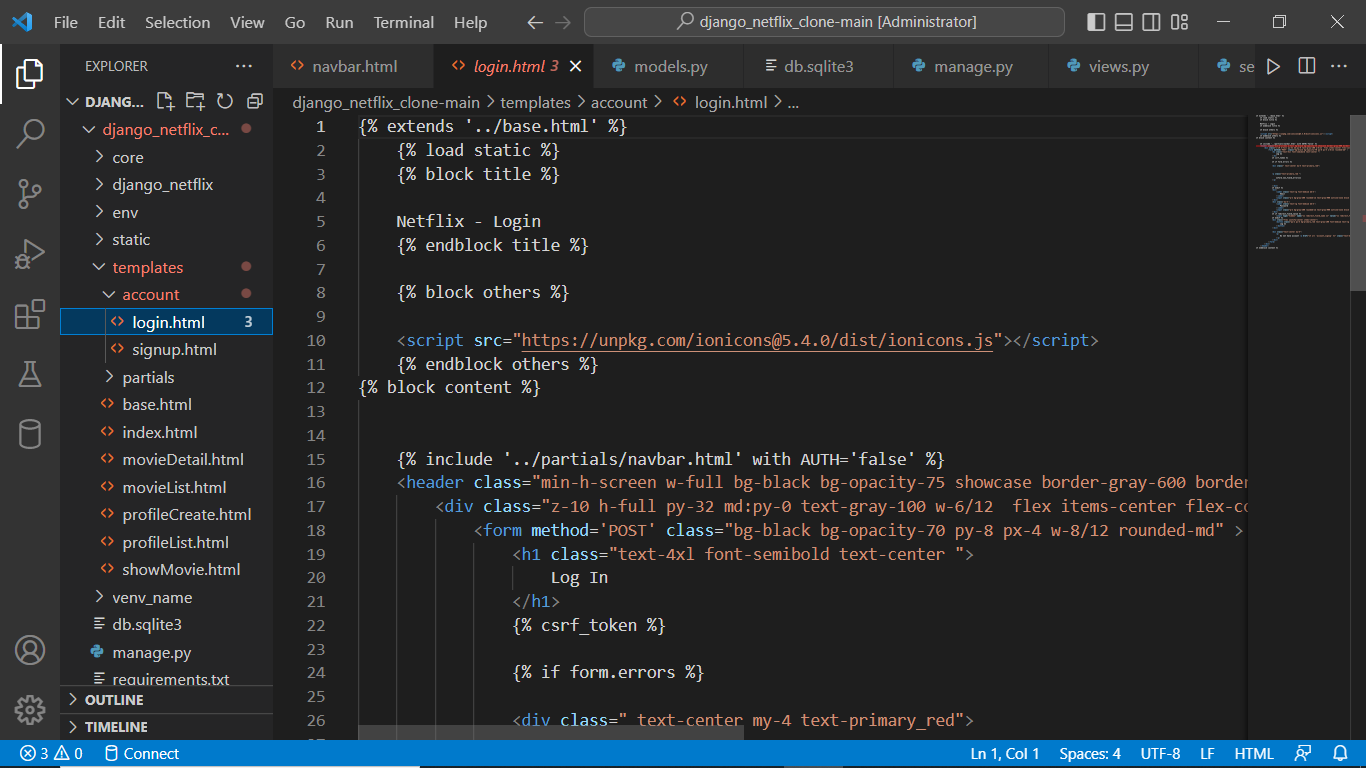
I used the Django views to render and redirect to different web pages on request by the user. More so, they have been used with the help of “POST” method to transfer data from a webpage (in put fields) to the models for storage and retrieving using query sets. Below is a sample of a view used to display the home page and profile list.

Users can be able to play a movie of their preffered choice once their credentials have been verified as the screenshot below shows





To achieve consistency throughout the whole application added a folder **“Partials ”** in the **Templates** folder with an html file named **‘nav.html’** that’s included within all templates so as not to lose consistency.



**Challenges faced**

Developing a Netflix clone using Django presents a set of challenges that need to be addressed for a successful and smooth implementation. Some of the key challenges include:

**Scalability**: As your platform gains popularity, the number of users and concurrent requests will increase. Ensuring that your application can scale horizontally to accommodate higher traffic is essential. Django does offer scalability options, but planning and implementing them correctly can be challenging.

**Performance**: Streaming high-quality videos demands efficient handling of media files. Django might not be the best option for serving media files directly due to its synchronous nature. Implementing a content delivery network (CDN) or using specialized media servers can be complex.

**Media Storage**: Storing and serving multimedia files like images and videos efficiently is crucial for a streaming platform. Integrating with cloud storage services like Amazon S3 requires proper configuration and handling of permissions and access keys.

**Security**: Security is paramount, especially when handling user data and payments. Implementing robust user authentication and authorization mechanisms, protecting against SQL injection, and ensuring secure communication through HTTPS are critical aspects.

Addressing these challenges requires careful planning, architectural decisions, and a deep understanding of both Django and the specific requirements of a streaming platform.

**Conclusion**

In conclusion, the creation of a Netflix clone using Django has been a journey of innovation, challenges, and accomplishments. Throughout this project, i aimed to replicate the essence of the renowned streaming platform while leveraging the capabilities of the Django web framework. The outcome of this endeavor underscores the power of modern technologies in shaping immersive user experiences and robust backend systems.

The project's success hinged on a meticulous combination of backend and frontend technologies. Django, with its robust architecture and extensive ecosystem, provided a solid foundation for crafting a feature-rich backend. The utilization of Django Rest Framework facilitated the creation of seamless APIs for data interaction, allowing users to register, log in, explore content, and enjoy seamless streaming.

On the frontend, React emerged as an invaluable asset, enabling the creation of dynamic user interfaces that adapt gracefully across devices and screen sizes.

Looking forward, the roadmap for this Netflix clone extends beyond its current state. The groundwork laid in this project opens avenues for further enhancements, including the introduction of user profiles, synchronized multi-device streaming, content reviews and ratings, payment gateways for premium subscriptions, and seamless social media integration. These additions would elevate the clone's engagement and interaction, contributing to its evolution as a comprehensive streaming solution.

In closing, the journey of building a Netflix clone using Django has been a testament to the capabilities of modern web development technologies. The project's successful implementation serves as an inspiration for aspiring developers and underscores the significance of thoughtful architecture, meticulous planning, and continuous innovation in shaping user-centric digital experiences.