Movie Recommendation

Alberto Cruz Luis
https://github.com/AlbertoCruzLuis

Index

Introduction

This is a project to put into practice much of the knowledge learned in recent months.

This project is a dynamic website that will be implemented with the python - Flask micro framework. This will allow us to use a database to store all the content of the application.

The site is based on a recommendation system that will be implemented through the cosine similarity algorithm that we can find in the scikit-Learn library

This project will address issues such as Web Frameworks, Machine Learning Models, Development of both the Front and Backend of the application and its corresponding Deployment.

The entire project will be stored in a Github repository: https://github.com/AlbertoCruzLuis/MovieRecomendationFlask

The project will be displayed in heroku at the following web URL:

https://movierecomendationflask.herokuapp.com/

Keywords: Movie System Recommendation, Scikit-Learn, Python, Flask, Html, Css, Bootstrap 4, SqlAlchemy, PostgreSQL, VirtualENV, Git, Heroku…

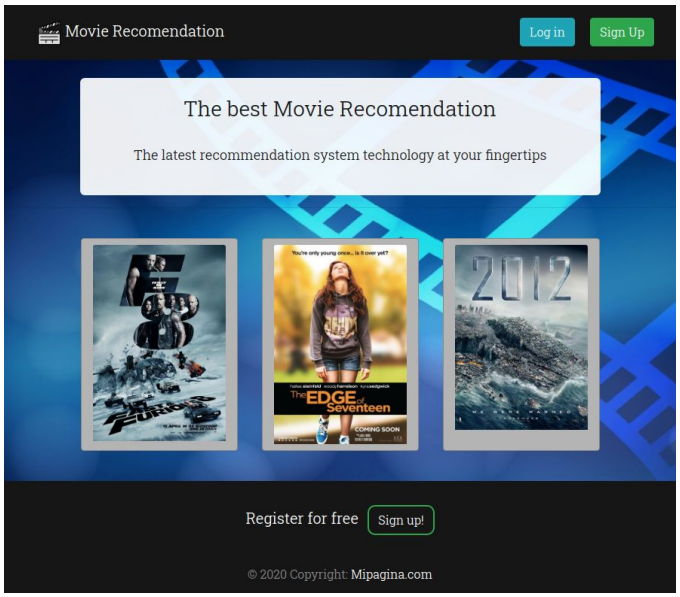
Fronted

This is the graphic part of the application. For this I used Bootstrap which is a library that incorporates many components such as buttons, navigation bars and many other things that will be useful for the project.

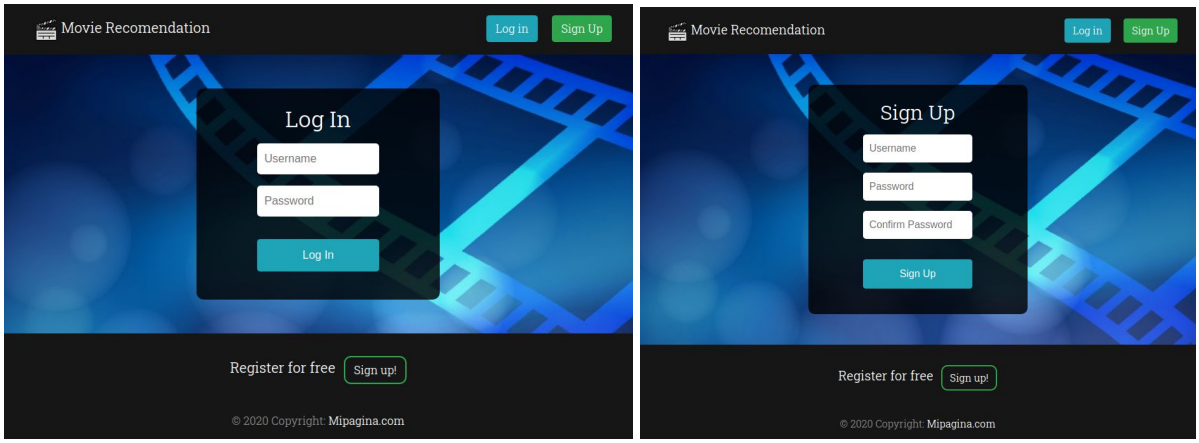
Besides that, I have used my own components that I have designed with css and html and in some cases in particular js.

The application is divided in 4 sections:

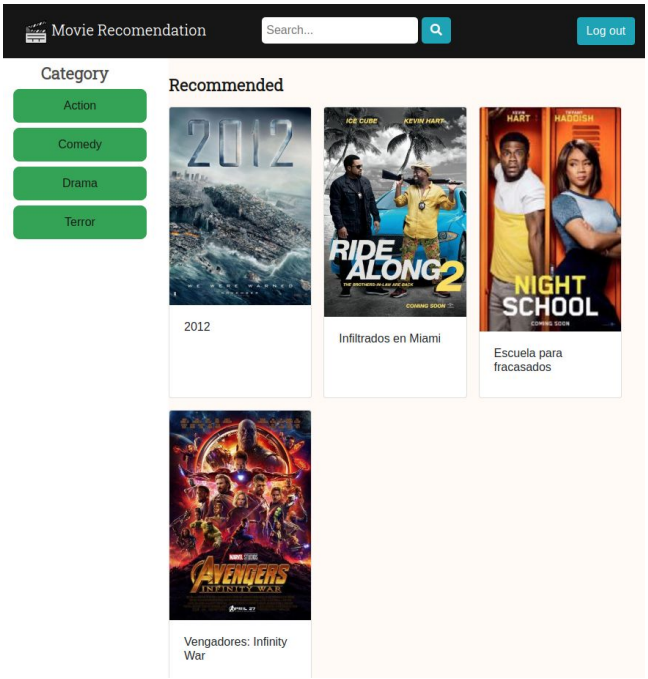
The welcome page



The logging and registration of the users



The window where the whole movie recommender system is mounted



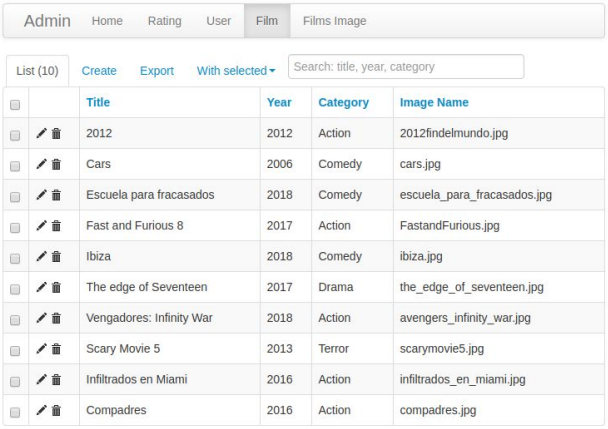
Backend

In this part we will deal with the logical part of the application such as checking the users, handling errors like the typical 404 page not found, and all that with the corresponding use of the data in the database.

All this will be implemented using the python programming language under the Flask framework.

I have also used an administrator panel provided by Flask-Admin with which I can create, update and delete all the information in the database.

This panel is restricted so that only the super user can enter.



Database

For the implementation of the database we have used Flask-sqlAlchemy which is a very useful tool since you can use it as ORM(Object-Relational mapping) in addition to being able to use the SQL database language itself.

All this is implemented first in mysql but as heroku had easy integration with the Postgresql database, what I did was migrate the data with the py-mysql2pgsql module. And from there I started to work with Postgresql

Inteligencia Artificial

This is the section that focuses on the engine of the application.

In it we will use a Recommendation System that is one of the uses given to the Machine Learning section.

We will use a simple method to recommend the films to users by rating them.

For this I have used the cosine similarity algorithm that we can find in the scikit-learn library.

Deploy

Finally to be able to launch the application on a server so that any user can visit it I have used Heroku which is a platform to create services, apps and run them on a server.

The way it works is very simple since you only have to upload 2 files:

The file with the necessary dependencies that will go in the file requirements.txt

The Procfile file from which you will tell him what service our application will run with. In our case we have used gunicorn which is a service to upload files with the HTTP protocol

Bibliography

Bootstrap: https://getbootstrap.com/

Flask: https://flask.palletsprojects.com/en/1.1.x/

Scikit-Learn: https://scikit-learn.org/stable/

Recommendation System:

https://towardsdatascience.com/cosine-similarity-for-movie-recommendation-system-e1852018cf76

Flask-SqlAlchemy: https://flask-sqlalchemy.palletsprojects.com/en/2.x/

Postgresql: https://www.postgresql.org/

Heroku: https://www.heroku.com/

Git: https://git-scm.com/

Gunicorn: https://gunicorn.org/