**National University of Computer & Emerging Sciences, Karachi**

**Spring 2023**

**PROJECT PROPOSAL**

**WEB ENGINEERING**

**NETFLIX CLONE**

**(VIDEO STREAMING APPLICATION)**

**Monday, 20th March’23**

**BSE-6B**

**HASSAN ALI K20-1052**

**EHTESHAM ZAFAR K20-1655**

**AMNA MUBARAK K20-1695**

**PROJECT SUMMARY:**

# The Netflix clone is a web application that allows users to browse, search, and watch movies and TV shows online. The project will be built using the MERN stack, which includes MongoDB, Express, React, and Node.js. Users would be able to stream their favourite shows, movies directly from this application. Like Netflix, it would have categorization according to the genre and type of content. Furthermore, their will also be an admin dashboard from where the moderators/admins can view statistics and user engagements. They will also be able to apply crud operations on the website directly from the admin dashboard.

**PROBLEM STATEMENT:**

In this modern era where everything is shifting to internet and a downfall in cable TV users is seen around the world, our web-app plays an important role. New users everyday visits internet to stream and download movies and TV shows that they like which is quite hectic given that they first have to search the whole internet and then find the authentic download link and even after downloading, there is a high risk of computer virus. This is were Netflix Clone comes to play. This application provides all in one solution to users entertainment. With a library increasing everyday, users can stream and download their favourite movies and TV shows quite easily.

**PROJECT FEATURES:**

The application will have the following features:

**For Users**

1. **User Authentication:**

Users will be able to sign up and log in to the application. User authentication will be implemented using JSON Web Tokens (JWT).

1. **Search:**

Users will be able to search for movies and TV shows using keywords.

1. **Movie and TV Show Pages:**

The application will have dedicated pages for each movie and TV show. These pages will display the movie or TV show's details such as its rating, synopsis, trailer, cast, and crew.

1. **Watch-list:**

Users will be able to add movies and TV shows to their watch-list.

1. **Stream Content:**

Users would be able to select a movie and watch its trailer and also stream the movie or TV show.

1. **Categorization:**

All the movies and TV shows will be categorized according to the genre and the users can select various genres to stream videos.

**For Admins**

There will be an admin portal for the moderators which will have the following features:

1. **Dashboard:**

Admins will be able to monitor user engagements in the website allowing them. It will display analytic on the user interaction so that they admins can make business decisions. Admins can view all the users, their details and much more. Graphs and charts will be displayed here as well representing user engagement.

1. **CRUD Opeartions:**

Admins will be allowed to add update and delete movies directly from here which will make the operations quite easy and efficient.

**TECH STACK:**

The project will follow the standard MERN stack architecture where MongoDB will be used as the database, Express will handle the server-side logic, React will be used to create the client-side UI, and Node.js will be used to run the server.

To implement the project, the following tools and technologies will be used:

* MongoDB: A document-based database for storing movie, TV show and user’s data.
* Express: A web application framework for Node.js that will handle the server-side logic.
* React: A JavaScript library for building user interfaces that will create the client-side UI.
* Node.js: A JavaScript runtime environment that will be used to run the server.
* MongDBCompass: Compass is a free interactive tool for querying, optimizing, and analyzing the MongoDB data
* JWT: A JSON-based open standard for creating access tokens that will be used for user authentication.
* Jest: It is a JavaScript-based testing framework that lets us test both front-end and back-end applications
* Dockers: For containerizing application and making separate images of each layer for better scalability and faster deployment
* Github Actions: For creating a CI/CD pipeline that will help us automate our integration and testing process.