PROJECT REPORT

ON

NETFLIX CLONE

USING MERN STACK

BY,

AZIZA SULTANA FAROOQUI

RIZWAN MOHAMMAD

ABSTRACT :

In this project, we tried to understand the working of Netflix, how it fetches the movies data and tried to build a Netflix clone using ReactJS , TMDB API , NodeJs , MongoDB . We have signup page, signin page , browse page, and movie player. We can also fetch the details of any movie in the Netflix. It fetches the movies data from a third party API. It also has the trailer pop-ups.

INTRODUCTION:

This is a clone of Netflix website built using React.JS as a Front-end & Nodejs Expressjs as Back-end and Mongodb atlas as Database. It’s not a replica, and it doesn’t have all the features of Netflix website. it’s a similar version of Netflix with our own design touch, showing our abilities in React.JS to build something like Netflix. It contains the home page, sign-in page, sign-up page, browse page, and movie player.

TECHNOLOGIES:

NODEJS:

Node.js is an **open source, cross-platform runtime environment and library** that is used for running web applications outside the client's browser.

It is used for **server-side programming**, and primarily deployed for non-blocking, event-driven servers, such as traditional web sites and back-end API services, but was originally designed with real-time, push-based architectures in mind. Every browser has its own version of a JS engine, and node.js is built on Google Chrome's V8 JavaScript engine.

REACTJS:

The React.js framework is an open-source JavaScript framework and library developed by Facebook. It's used for building interactive user interfaces and web applications quickly and efficiently with significantly less code than you would with vanilla JavaScript.

In React, we develop our applications by creating reusable components that you can think of as independent Lego blocks. These components are individual pieces of a final interface, which, when assembled, form the application's entire user interface.

EXPRESS JS:

Express.js is a framework that works on top of Node.js web server functionality to simplify its APIs and add helpful new features. It makes it easier to organize your application's functionality with middleware and routing. It adds helpful utilities to Node.js HTTP objects and facilitates the rendering of dynamic HTTP objects.

MONGODB:

MongoDB's document model . MongoDB **stores data in flexible, JSON-like documents**, meaning fields can vary from document to document and data structure can be changed over time .The document model **maps to the objects in your application code**, making data easy to work with.