Full Stack Project Report (2022-23)

# NETFLIX CLONE

Institute of Engineering and Technology

Submitted by:

Yash Sharma (201500825)

Harsh Agrawal (201500263)

Javvaji Vinay Venkat Sandeep (201500314)

Vutukuri Amaresh (201500808)

Supervised By:

Mrs. Ruchi Gupta Mam

Technical Trainer

Department of Computer Engineering and Application

## Declaration

We here by declare that the work which is being presented in the Full Stack Project

“Netflix Clone”, in partial fulfilment of the requirements for Full Stack Project, is an authentic record of our own work carried under the supervision of Mrs Ruchi Gupta , Technical Trainer, GLA University, Mathura.

Yash Sharma (201500825)

Sign: _____________

Harsh Agrawal (201500263)

Sign: _____________

Vutukuri Amaresh (201500808)

Sign: _____________

Javvaji Sandeep (201500314)

Sign: _____________

Course: B. Tech (CSE)
Year: 3rd
Semester: VI

Department of computer Engineering and Applications
GLA University, Mathura
17 km. Stone NH#2, Mathura -Delhi Road, P.O. – Chaumuha,
Mathura – 281406

# Certificate

This is to certify that the project entitled “Netflix Clone” carried out for Mini Project is the work done by Yash Sharma, Harsh Agrawal, Vutukuri Amaresh, Javvaji Sandeep is submitted in partial fulfilment of the requirements for the award of degree Bachelor of Technology (Computer Science and Engineering).

Signature of Supervisor:
Name of Supervisor: Mrs. Ruchi Gupta
Date: 28-April-2023

# Acknowledgement

It is our pleasure to acknowledge the assistance of a number of people without whose help this project would not have been possible.

First and foremost, We I would like to express our gratitude to Mrs. Ruchi Gupta our project mentor, for providing invaluable Encouragement, guidance and assistance. We would like to thank my co-team members for their complete support throughout in finishing the mentioned project accurately. After doing this project We can confidently say that this experience has not only enriched us with technical knowledge but also has unparsed the maturity of thought and vision, the attributes required for being a professional.

# Abstract

At Netflix, we want to entertain the world. Whatever your taste, and no matter where you live, we give you access to best-in-class TV series, documentaries, feature films and mobile games. Our members control what they want to watch, when they want it, in one simple subscription.

Netflix content varies by region and may change over time. You can watch a variety of award-winning Netflix originals, TV shows, movies, documentaries, and more.
he more you watch, the better Netflix gets at recommending TV shows and movies.

You can watch Netflix through any internet-connected device that offers the Netflix app, including smart TVs, game consoles, streaming media players, set-top boxes, smartphones, and tablets. You can also watch Netflix on your computer using an internet browser. You can review the system requirements for web browser compatibility, and check our internet speed recommendations to achieve the best performance.

# Table of Contents

# Chapter-1 Introduction

## 1.1. Overview

At Netflix, we want to entertain the world. Whatever your taste, and no matter where you live, we give you access to best-in-class TV series, documentaries, feature films and mobile games. Our members control what they want to watch, when they want it, in one simple subscription.

## 1.2. Motivation

We want our teams to do what is best for Netflix. This, in turn, generates a sense of responsibility, accountability and self-discipline that drives us to do great work. Freedom itself is not the goal; the goal is creating a strong sense of caring for Netflix so that people do what is best for the company.

## 1.3. Project Plan

### 1.3.1. Objective

At Netflix, we want to entertain the world. Whatever your taste, and no matter where you live, we give you access to best-in-class TV series, documentaries, feature films and mobile games. Our members control what they want to watch, when they want it, in one simple subscription.

### 1.3.2. Scope

Netflix is a global streaming entertainment service offering movies, TV series and games, with unlimited viewing on any internet-connected screen for an affordable, no-commitment monthly fee.

## 1.3. Drawbacks in Existing System

Since its launch, the company has been subject to numerous criticisms, the basis of which range from its business practices and workplace culture to issues with the service it provides, including content issues, lack of close captioning and pricing.

# Chapter-2 Software & Hardware Requirement Analysis

## 2.1. Hardware Requirements

- Processor: AMD Ryzen 5
- Main Memory (RAM): 256 MB
- Cache Memory: 512 KB
- Monitor: 13.5-inch Colour Monitor
- Keyboard: 108 Keys
- Mouse: Optical Mouse

## 2.2. Software Requirements

- System Software

  - Operating System: Windows 11

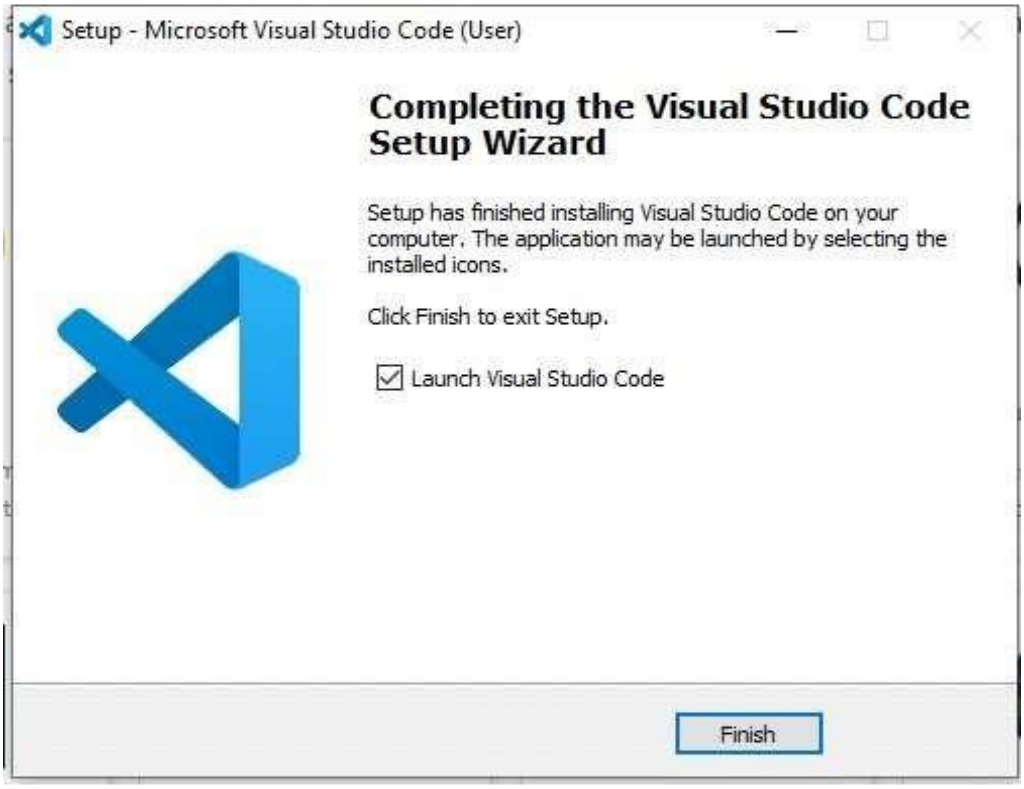
- Application Software - Tools: GitHub, VS Code - Front-end:

  □ React

## 2.3. Installation of VS Code

VS Code is a free code editor, which runs on the macOS, Linux, and Windows operating systems.

VS Code is lightweight and should run on most available hardware and platform versions. You can review the System Requirements to check if your computer configuration is supported.



Setup - Microsoft Visual Studio Code (User)

Completing the Visual Studio Code Setup Wizard

Setup has finished installing Visual Studio Code on your computer. The application may be launched by selecting the installed icons.

Click Finish to exit Setup.

☑ Launch Visual Studio Code

Fig1. Finish up Installing.

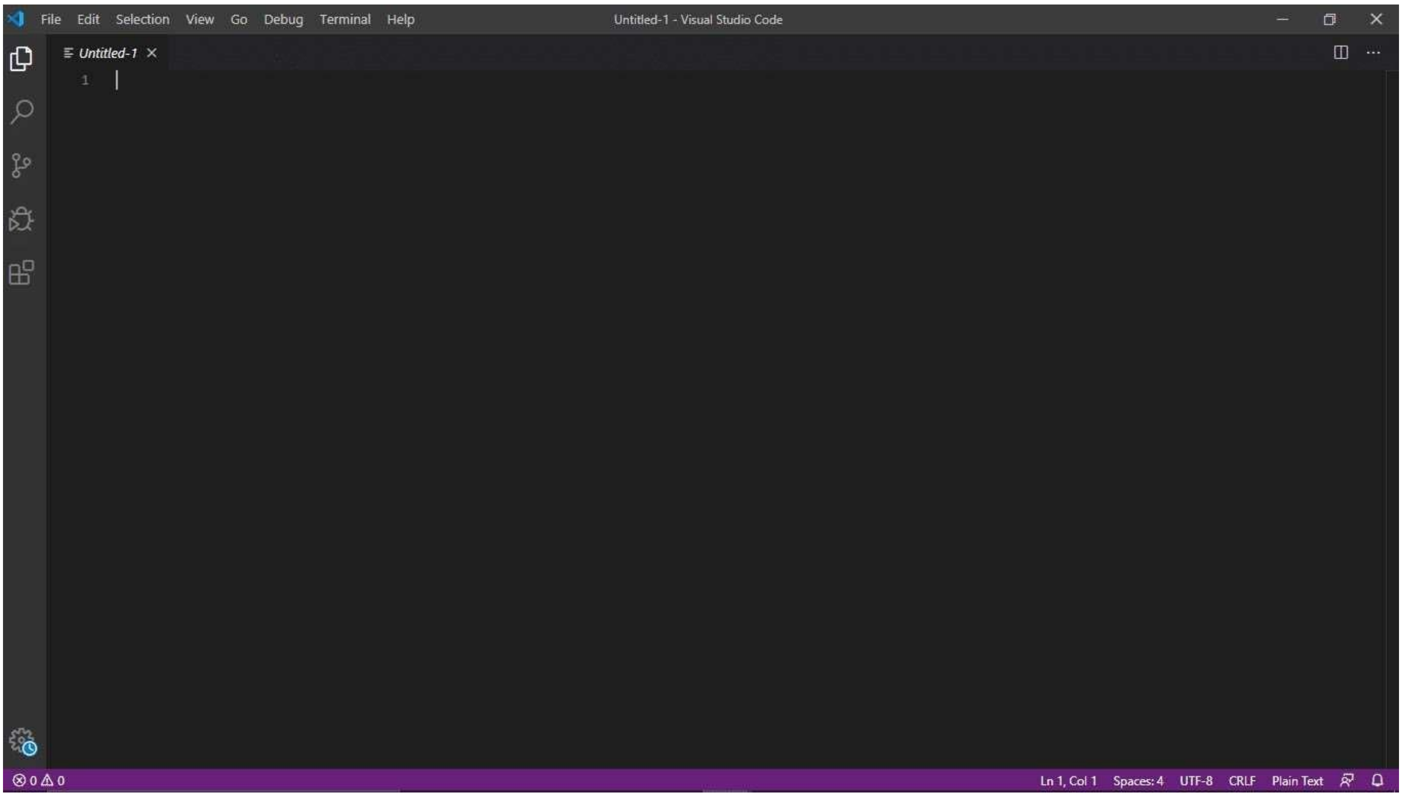


Fig2. VS Code Window.

## 2.4. Specific Requirements

### 2.4.1 Languages Used

### React:

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, you develop your 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 is a node js web application framework that provides broad features for building web and mobile applications. It is used to build a single page, multipage, and hybrid web application. It's a layer built on the top of the Node js that helps manage servers and routes.

## Node.js:

Node.js is similar in design to, and influenced by, systems like Ruby's Event Machine and Python's Twisted. Node.js takes the event model a bit further. It presents an event loop as a runtime construct instead of as a library. In other systems, there is always a blocking call to start the event-loop.

## Mongo DB:

MongoDB is a document database used to build highly available and scalable internet applications. With its flexible schema approach, it's popular with development teams using agile methodologies.

# CHAPTER -3

## Software Designs

### 3.1 Use Case diagram

A use case diagram is a dynamic or behaviour diagram in UML. Use case diagrams model the functionality of a system using actors and use cases. Use cases are a set of actions, services, and functions that the system needs to perform.

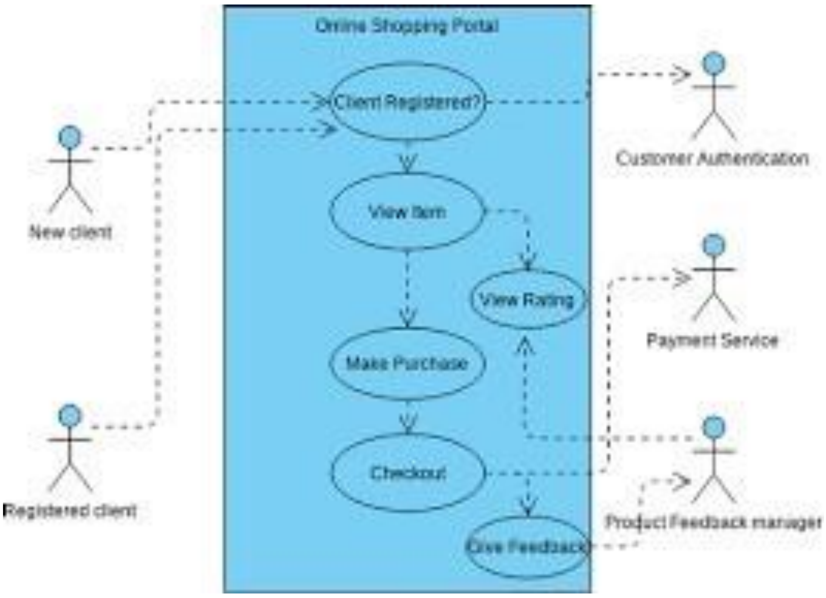
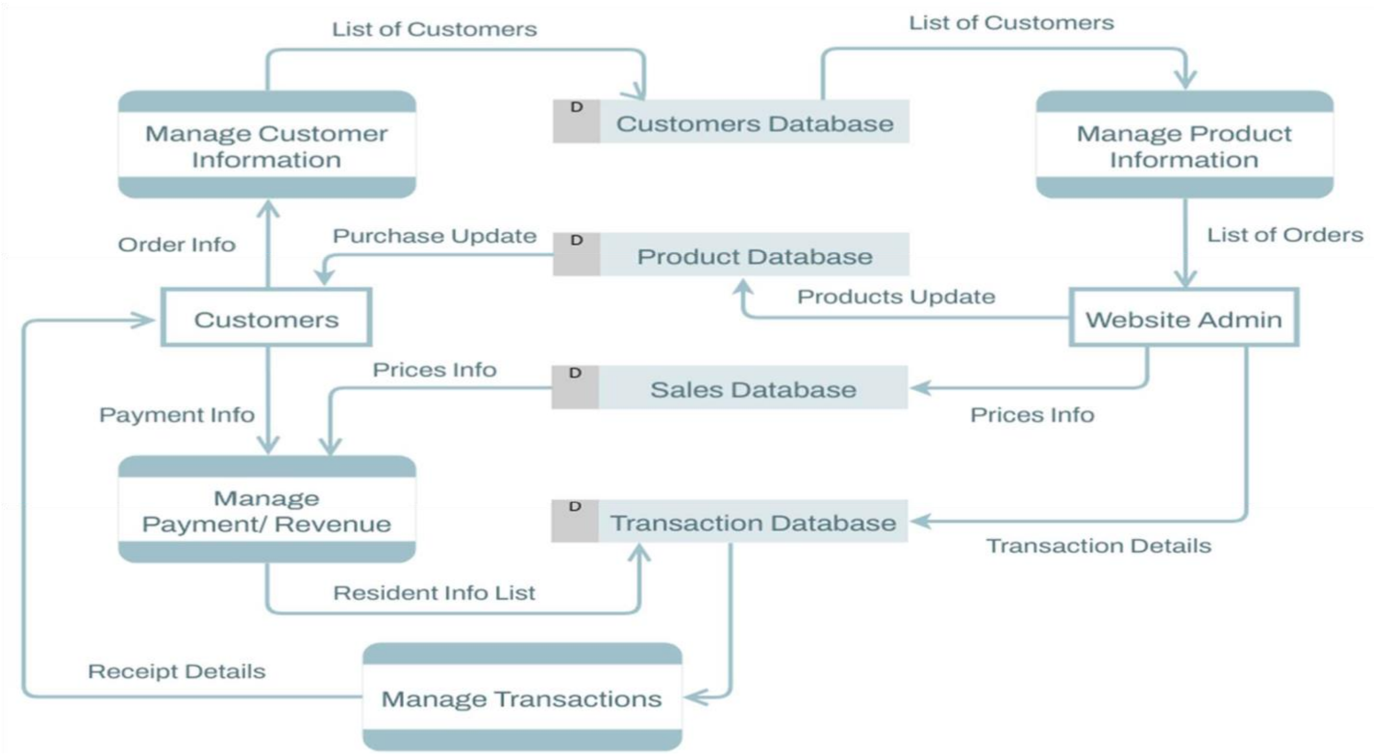


Fig 3.1 Use Case diagram

### 3.2 Data Flow Diagram

Fig. 3.2 Level-0 DFD



# Chapter-4 Testing

## 4.1 Introduction

The implementation phase of software development is concerned with translating design specification into source code. The preliminary goal of implementation is to write source code and internal documentation so that conformance of the code to its specifications can be easily verified, and so that debugging, testing and modifications are eased. This goal can be achieved by making the source code as clear and straightforward as possible. Simplicity, clarity and elegance are the hallmark of good programs, obscurity, cleverness, and complexity are indications of inadequate design and misdirected thinking.

Source code clarity is enhanced by structured coding techniques, by good coding style, by, appropriate supporting documents, by good internal comments, and by feature provided in modern programming languages.

The implementation team should be provided with a well-defined set of software requirement, an architectural design specification, and a detailed design description. Each team member must understand the objectives of implementation.

Fig5. Testing

## 4.2. Error

The term error is used in two ways. It refers to the difference between the actual output of software and the correct output, in this interpretation, error is essential a measure of the difference between actual and ideal. Error is also to used to refer to human action that result in software containing a defect or fault.

## 4.3. Fault

Fault is a condition that causes to fail in performing its required function. A fault is a basic reason for software malfunction and is synonymous with the commonly used term Bug.

## 4.4. Failure

Failure is the inability of a system or component to perform a required function according to its specifications. A software failure occurs if the behaviour of the software is the different from the specified behaviour. Failure may be caused due to functional or performance reasons. a. Unit Testing

The term unit testing comprises the sets of tests performed by an individual programmer prior to integration of the unit into a larger system.

A program unit is usually small enough that the programmer who developed it can test it in great detail, and certainly in greater detail than will be possible when the unit is integrated into an evolving software product. In the unit testing the programs are tested separately, independent of each other. Since the check is done at the program level, it is also called program teasing. b. Module Testing

A module and encapsulates related component. So can be tested without other system module. c. Subsystem Testing

Subsystem testing may be independently design and implemented common problems are subsystem interface mistake in this checking we concenton it. There are four categories of tests that a programmer will typically perform on a program unit. i Functional test

ii Performance test

iii Stress test iv

Structure test

## 4.5 Functional Test

Functional test cases involve exercising the code with Nominal input values for which expected results are known; as well as boundary values (minimum values, maximum values and values on and just outside the functional boundaries) and special values.

## 4.6 Performance Test

Performance testing determines the amount of execution time spent in various parts of the unit, program throughput, response time, and device utilization by the program unit. A certain amount of avoid expending too much effort on fine-tuning of a program unit that contributes little to the overall performance of the entire system. Performance testing is most productive at the subsystem and system levels.

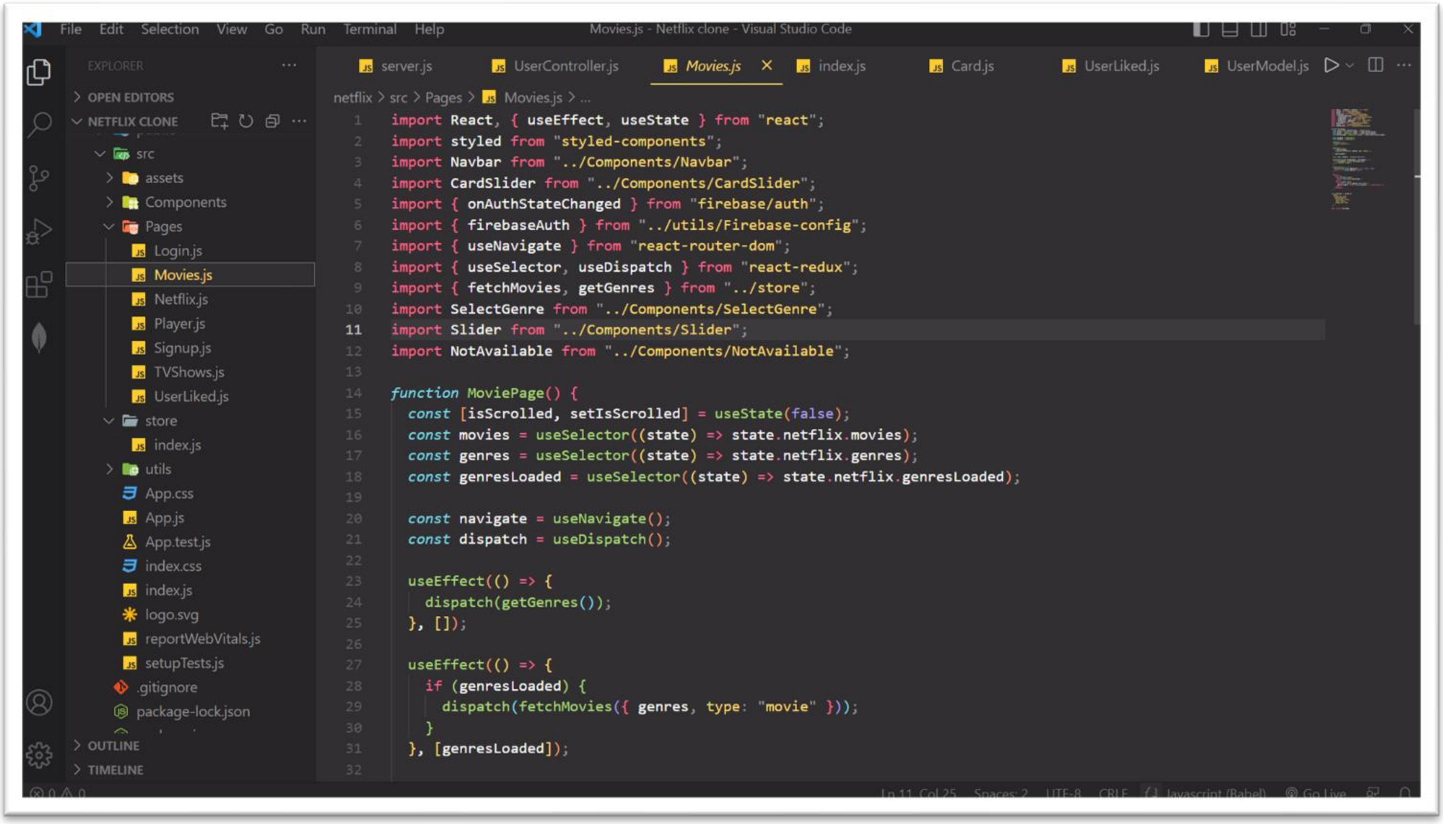
## 4.7 Stress Test

Stress test are those designed to intentionally break the unit. A great deal can be learned about the strengths and limitations of a program by examining the manner in which a program unit breaks.

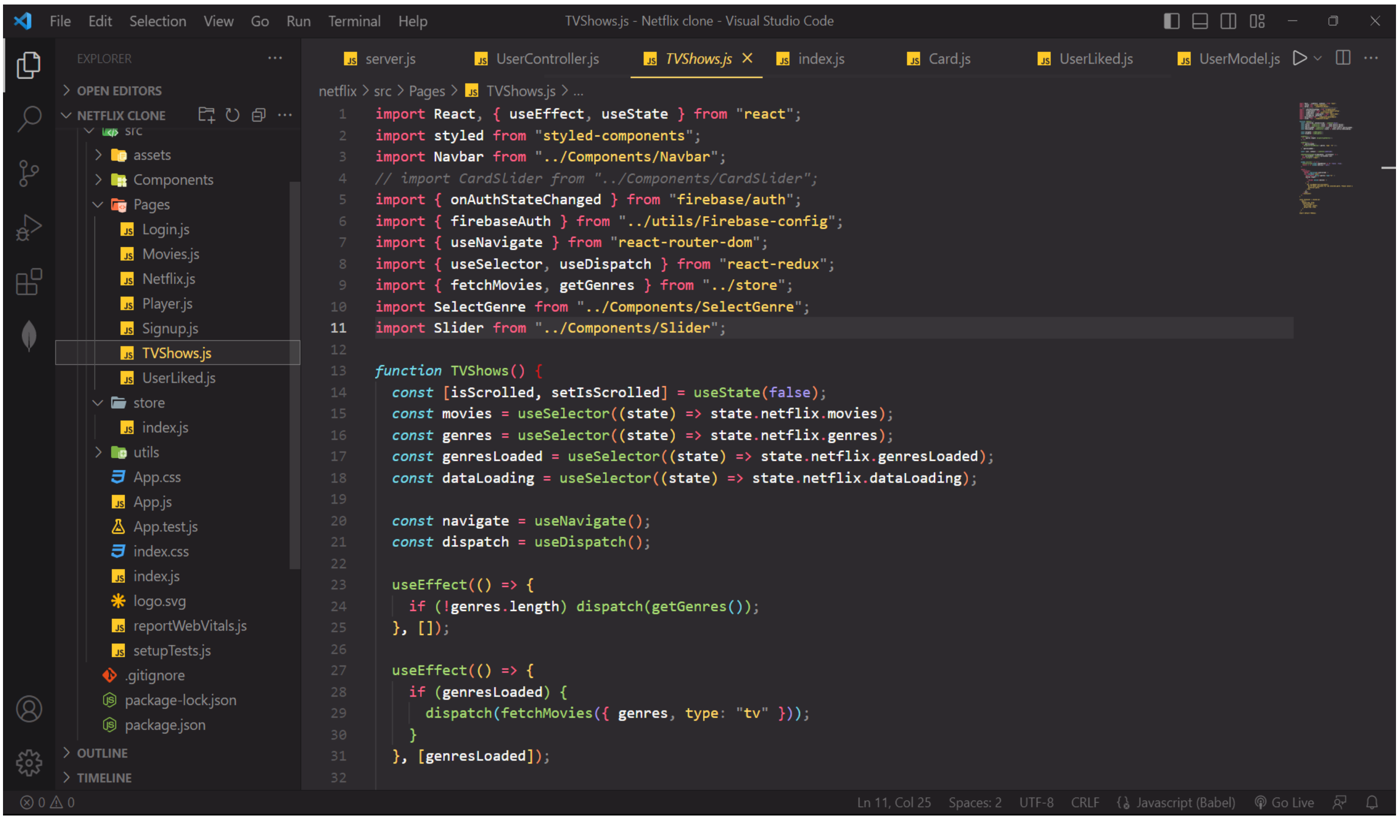
## 4.8 Structure Test

Structure tests are concerned with exercising the internal logic of a program and traversing particular execution paths. Some authors refer collectively to functional performance and stress testing as “black box” testing. While structure testing is referred to as “white box” or “glass box” testing. The major activities in structural testing are deciding which path to exercise, deriving test date to exercise those paths, determining the test coverage criterion to be used, executing the test, and measuring45 the test coverage achieved when the test cases are exercised.

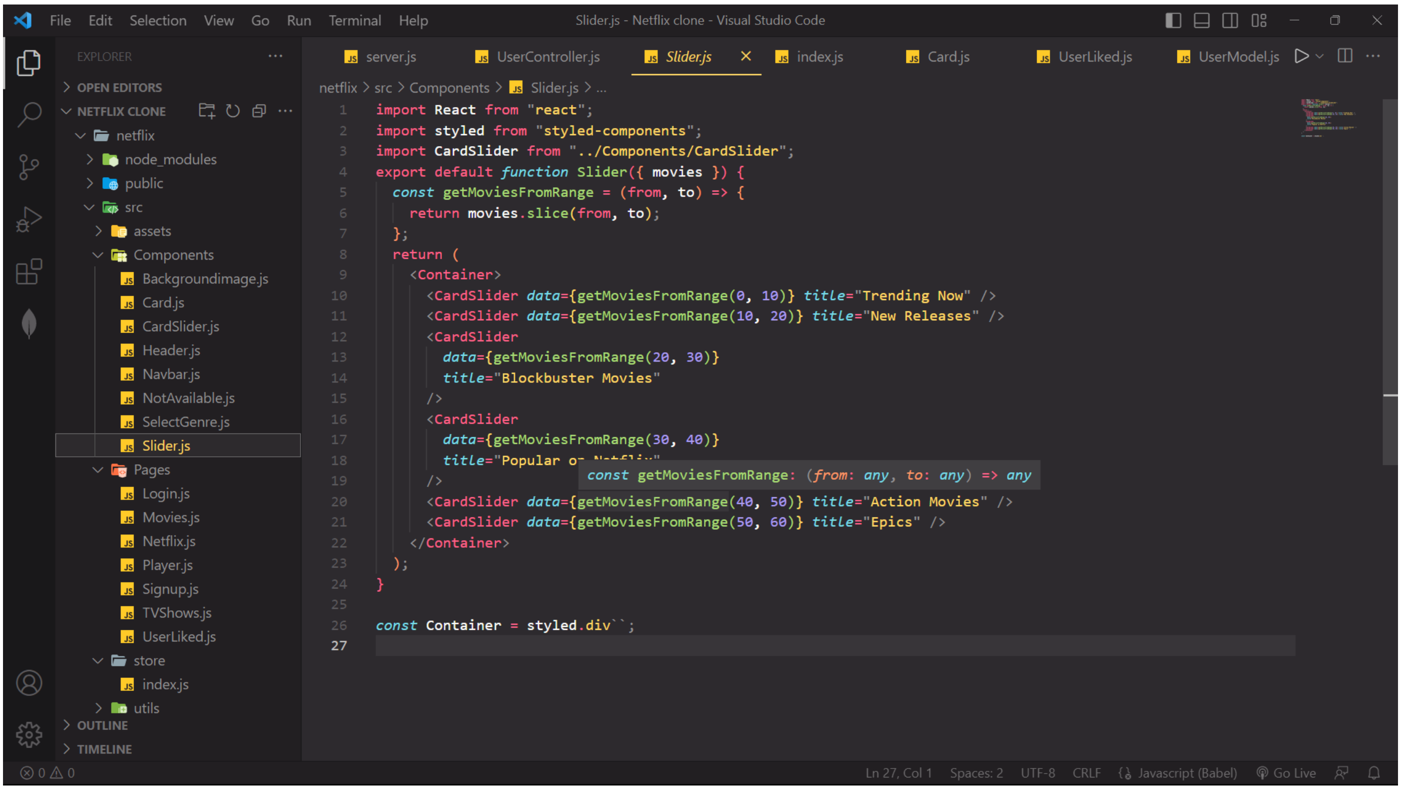
# Chapter-5 Implementation and User Interface



```javascript
import React, { useEffect, useState } from "react";
import styled from "styled-components";
import Navbar from "../Components/Navbar";
import CardSlider from "../Components/CardSlider";
import { onAuthStateChanged } from "firebase/auth";
import { firebaseAuth } from "../utils/Firebase-config";
import { useNavigate } from "react-router-dom";
import { useSelector, useDispatch } from "react-redux";
import { fetchMovies, getGenres } from "../store";
import SelectGenre from "../Components/SelectGenre";
import Slider from "../Components/Slider";
import NotAvailable from "../Components/NotAvailable";

function MoviePage() {
  const [isScrolled, setIsScrolled] = useState(false);
  const movies = useSelector((state) => state.netflix.movies);
  const genres = useSelector((state) => state.netflix.genres);
  const genresLoaded = useSelector((state) => state.netflix.genresLoaded);

  const navigate = useNavigate();
  const dispatch = useDispatch();

  useEffect(() => {
    dispatch(getGenres());
  }, []);

  useEffect(() => {
    if (genresLoaded) {
      dispatch(fetchMovies({ genres, type: "movie" }));
    }
  }, [genresLoaded]);

```



```
import React, { useEffect, useState } from "react";
import styled from "styled-components";
import Navbar from "../Components/Navbar";
// import CardSlider from "../Components/CardSlider";
import { onAuthStateChanged } from "firebase/auth";
import { firebaseAuth } from "../utils/Firebase-config";
import { useNavigate } from "react-router-dom";
import { useSelector, useDispatch } from "react-redux";
import { fetchMovies, getGenres } from "../store";
import SelectGenre from "../Components/SelectGenre";
import Slider from "../Components/Slider";

function TVShows() {
  const [isScrolled, setIsScrolled] = useState(false);
  const movies = useSelector((state) => state.netflix.movies);
  const genres = useSelector((state) => state.netflix.genres);
  const genresLoaded = useSelector((state) => state.netflix.genresLoaded);
  const dataLoading = useSelector((state) => state.netflix.dataLoading);

  const navigate = useNavigate();
  const dispatch = useDispatch();

  useEffect(() => {
    if (!genres.length) dispatch(getGenres());
  }, []);

  useEffect(() => {
    if (genresLoaded) {
      dispatch(fetchMovies({ genres, type: "tv" }));
    }
  }, [genresLoaded]);

```

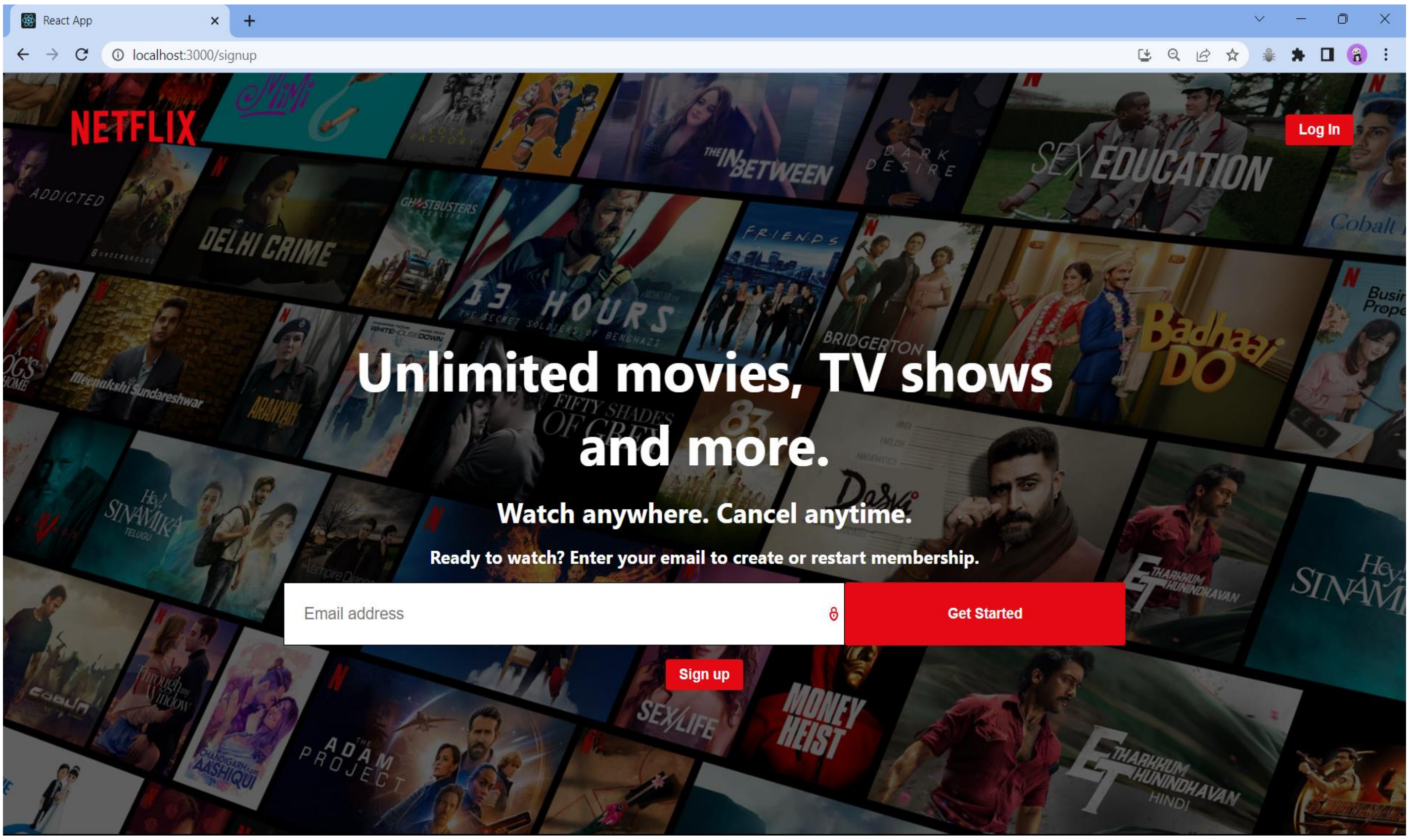


```
import React from "react";
import styled from "styled-components";
import CardSlider from "../Components/CardSlider";
export default function Slider({ movies }) {
  const getMoviesFromRange = (from, to) => {
    return movies.slice(from, to);
  };
  return (
    <Container>
      <CardSlider data={getMoviesFromRange(0, 10)} title="Trending Now" />
      <CardSlider data={getMoviesFromRange(10, 20)} title="New Releases" />
      <CardSlider
        data={getMoviesFromRange(20, 30)}
        title="Blockbuster Movies"
      />
      <CardSlider
        data={getMoviesFromRange(30, 40)}
        title="Popular on Netflix"
      />
      <CardSlider data={getMoviesFromRange(40, 50)} title="Action Movies" />
      <CardSlider data={getMoviesFromRange(50, 60)} title="Epics" />
    </Container>
  );
}

const Container = styled.div``;

```

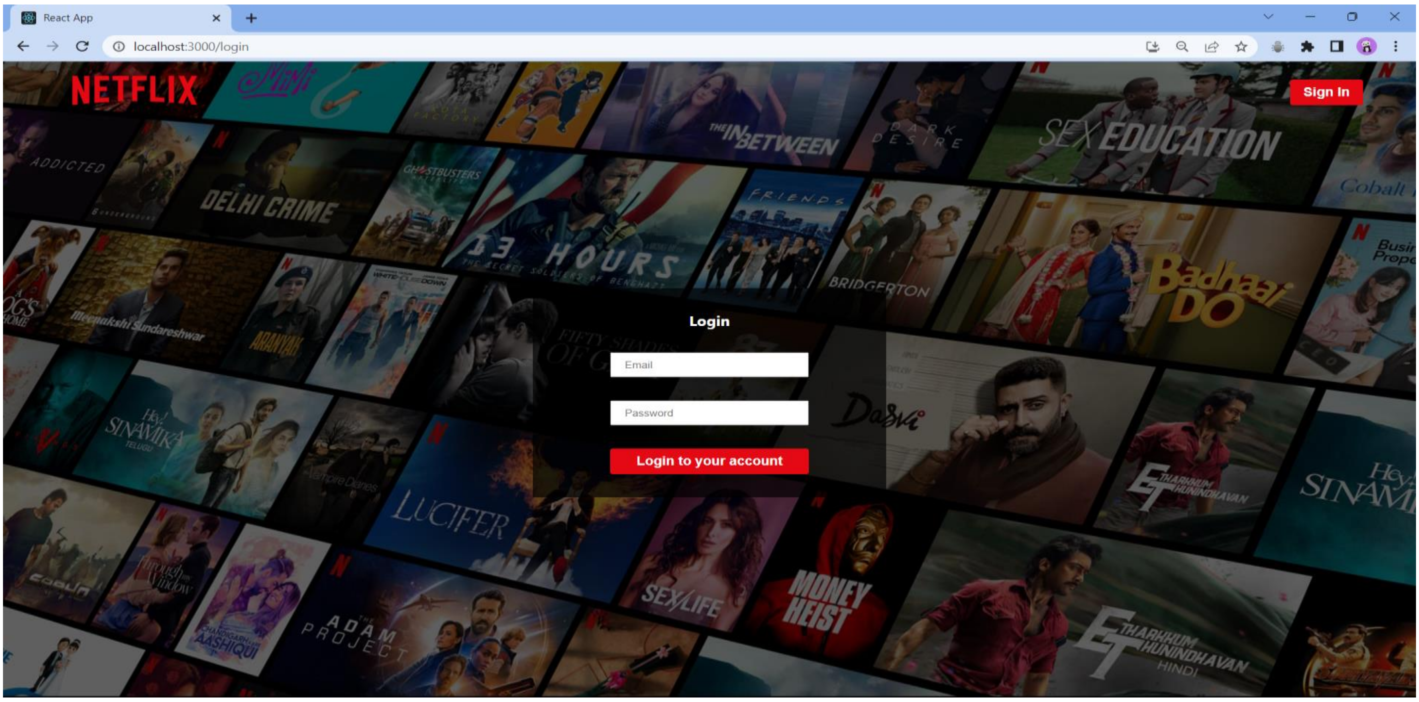
```
const getMoviesFromRange: (from: any, to: any) => any
```

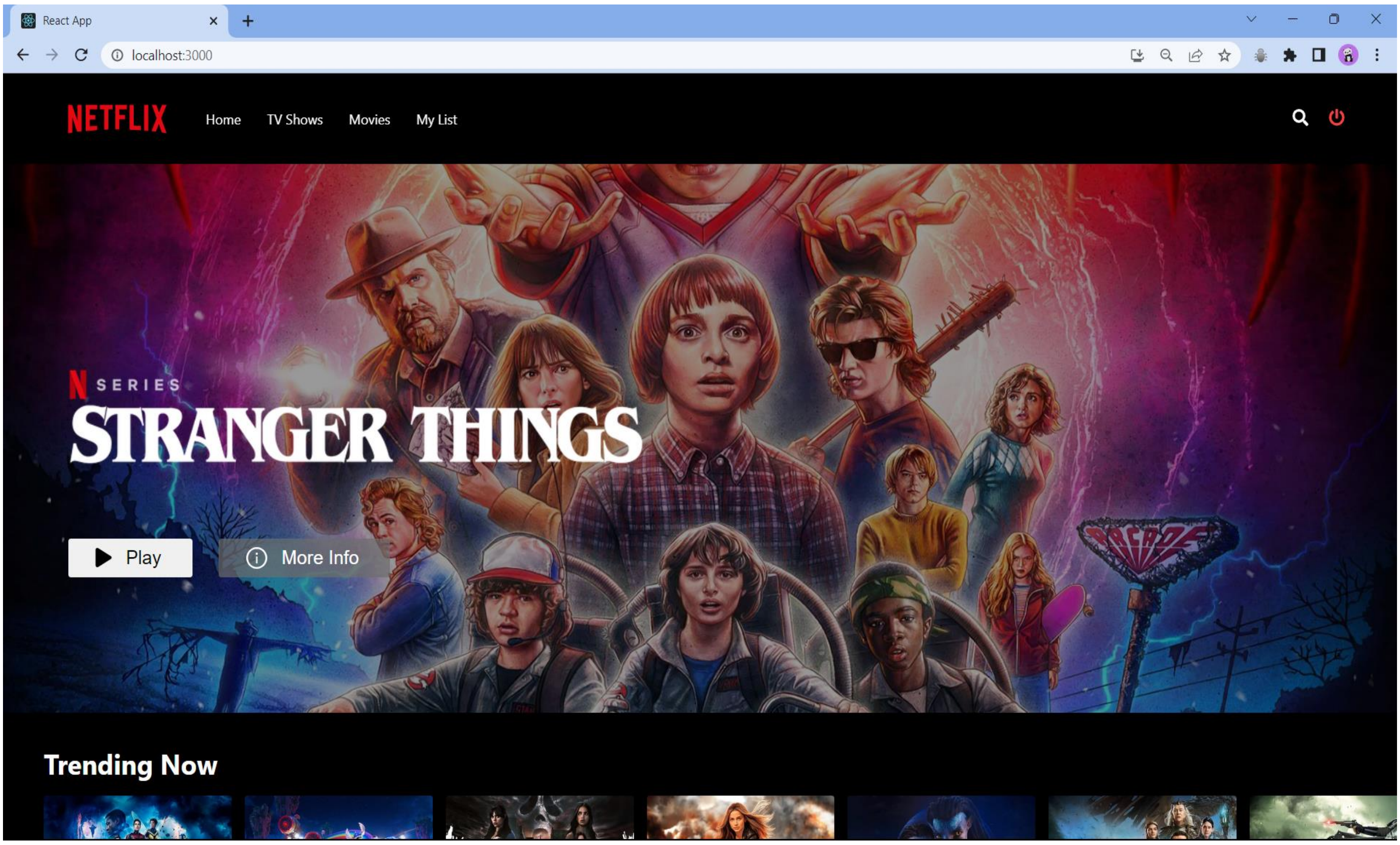
# OUTPUT

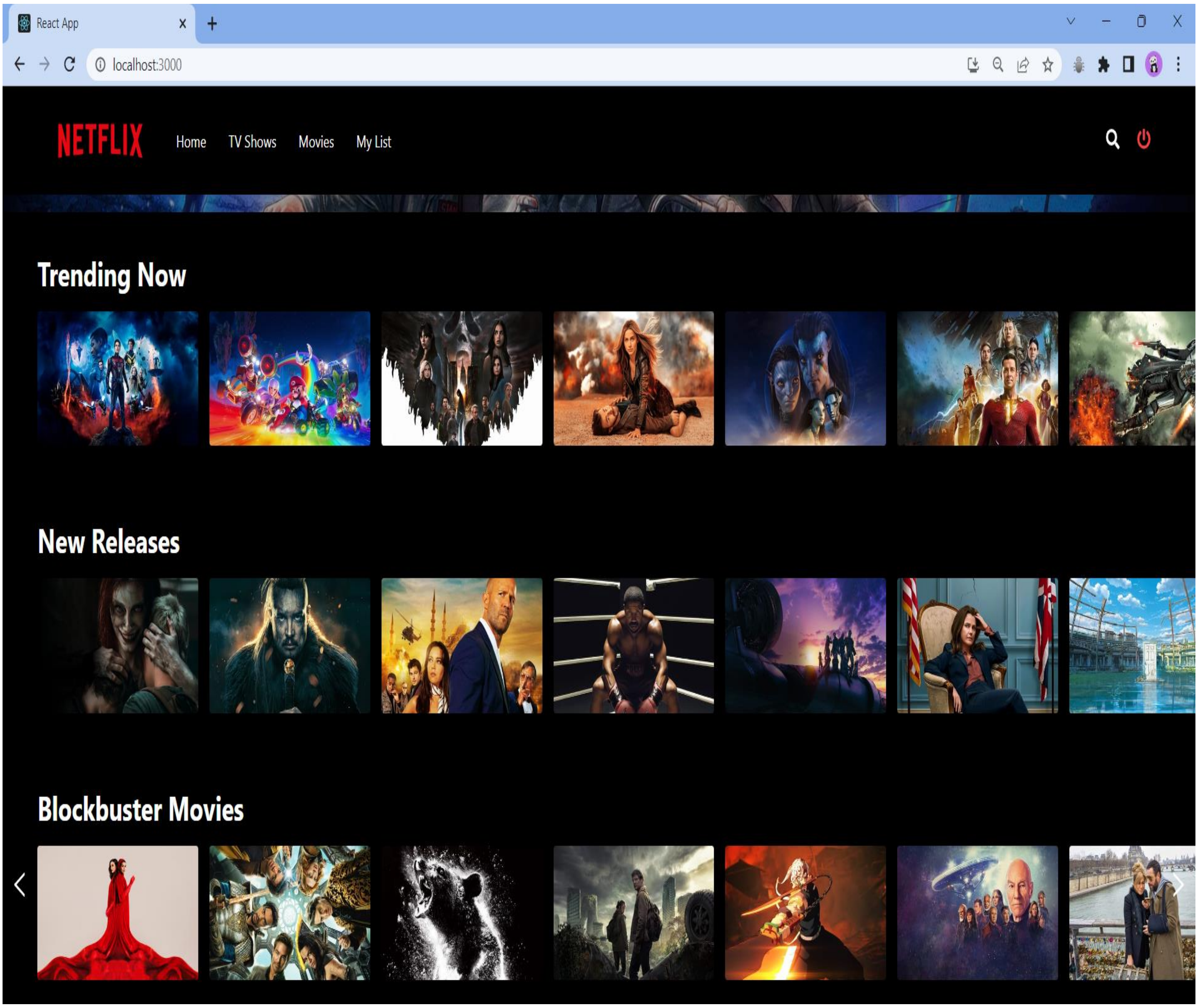
Home Page:

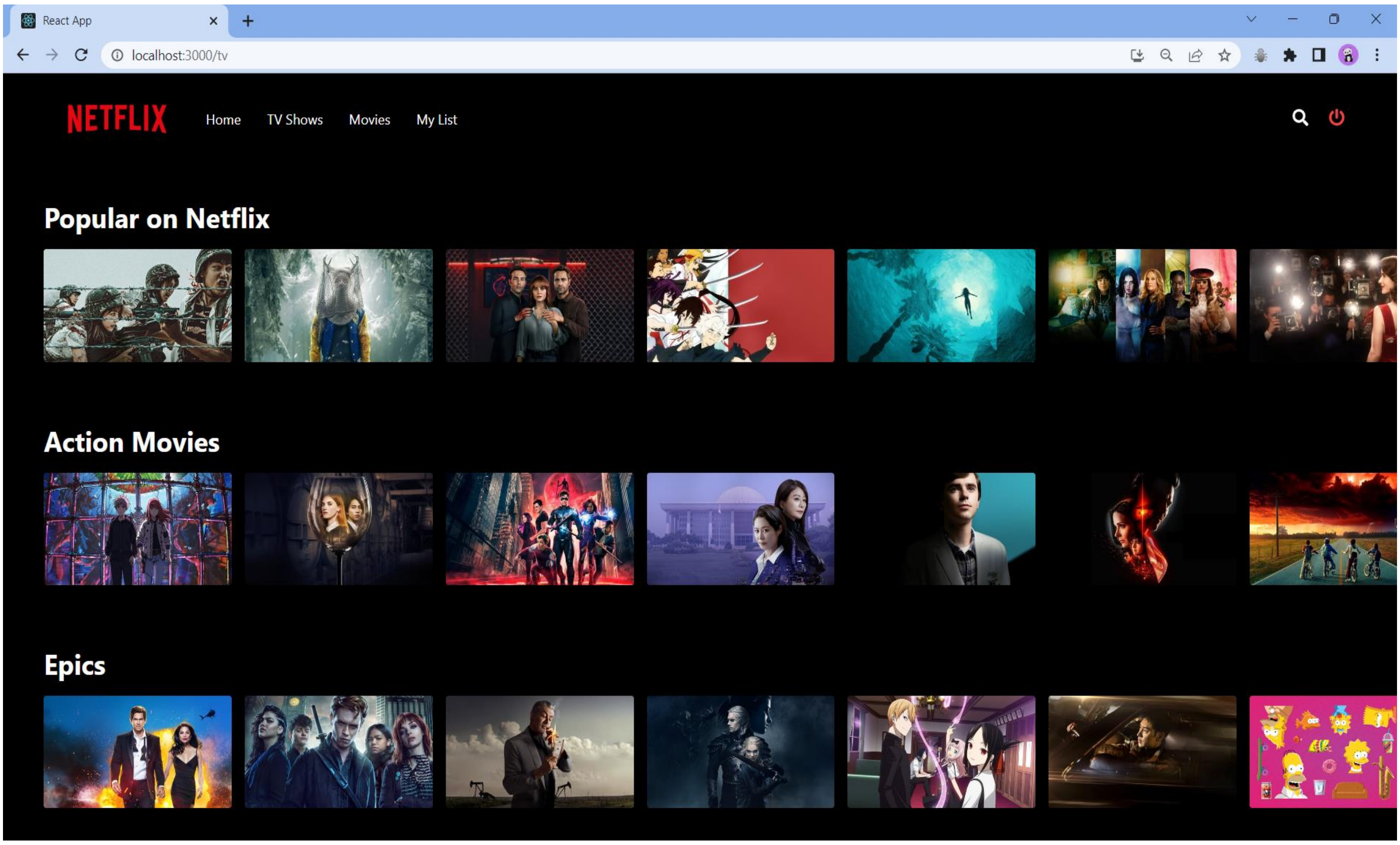

React App
localhost:3000/signup
NETFLIX
Log In
Unlimited movies, TV shows and more.
Watch anywhere. Cancel anytime.
Ready to watch? Enter your email to create or restart membership.
Email address
Get Started
Sign up

Login Page:


React App
localhost:3000/login
NETFLIX
Sign In
Login
Email
Password
Login to your account


React App
localhost:3000
NETFLIX
Home
TV Shows
Movies
My List
N SERIES
STRANGER THINGS
Play
More Info
Trending Now


React App
localhost:3000
NETFLIX
Home
TV Shows
Movies
My List
Trending Now
New Releases
Blockbuster Movies



# Chapter-6 References

□

- □ www.javatpoint.com
- □ www.w3school.com
- □ www.tutorialspoint.com
- □ www.youtube.com
- □ www.giphy.com
- □ www.beta-labs.in