# VIDEO STREAMING WEBSITE

## PROJECT REPORT

Submitted by

Varun Chikkala 19BCB0062

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

## SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

# ABSTRACT

Cutting edge data set frameworks should uphold both literary information and different sorts of sight and sound information. These two kinds of information vary in their association and the executives. Sight and sound information includes static media like content and pictures, that don't change with time, and dynamic or ceaseless media like sound and video, that differs with time and requires an ensured move rate. Ordinary data set frameworks are intended for overseeing literary and mathematical information, and recovering such information is frequently founded on basic correlations of text/mathematical qualities. Be that as it may, this basic strategy for recovery is longer satisfactory for the sight and sound information, since the digitized portrayal of pictures, video, or information itself doesn't pass on the truth of these media things. Sight and sound information has certain attributes that muddle their capacity in regular data sets or record frameworks. In the first place, sight and sound information will in general be voluminous. Second, constant media information, for example, video and sound have timing attributes related with them. The proposed technique is to store media objects in a social data set, which offers a few benefits over record framework stockpiling, as secure back up, simultaneous login, and quicker recovery. utilizing a social data set framework, for example, MySQL and the use of continuous video real time in conveying such information. In view of the client question, the recovered sound/video information is transferred over the Internet utilizing Genuine Framework G2 which gives great sound and video at all piece rates and surprisingly under lossy organization conditions.

Due to the explosive growth of the Internet and increasing demand for multimedia information on the Web, streaming video over the Internet has received tremendous attention from academia and industry. During the COVID 19 pandemic, many people have started using streaming services. Streaming services have profited enormously in this pandemic time as most of people are in their homes and watch movies, TV shows etc. Video Streaming websites help users around the globe download and watch large video files from the comfort of their homes. For today's audiences it's all about immediacy and mobility, the content they are looking for must be just a click away to fit their needs and streaming websites is the best option for it. This project is to create a streaming website where most of the TV shows, movies etc are available for the users to watch and have a good time.

# INTRODUCTION

## 1.1 SYSTEM OVERVIEW

Video Streaming and broadband connections help users around the globe download and watch large video files from the comfort of their homes. Taking advantage of this technology, the American company Netflix launched a video streaming website on 2009 where users could watch the most recent Television episodes and Hollywood Blockbusters. Netflix changed content consumption models in the entertainment industry and led to the disappearance of the mainstream video rental store in North America. For today’s audiences it’s all about immediacy and mobility, the content they are looking for must be just a click away to fit their needs. This project is to create a streaming website where most of the TV shows, movies etc are available for the users to watch with just a click away and have a good time.

## 1.2 OBJECTIVE

Due to the explosive growth of the Internet and increasing demand for multimedia information on the Web, streaming video over the Internet has received tremendous attention from academia and industry. During the COVID 19 pandemic, many people have started using streaming services. Streaming services have profited enormously in this pandemic time as most of people are in their homes and watch movies, TV shows etc. Video Streaming websites help users around the globe download and watch large video files from the comfort of their homes. For today’s audiences it’s all about immediacy and mobility, the content they are looking for must be just a click away to fit their needs and streaming websites is the best option for it. This project is to create a streaming website where most of the TV shows, movies etc are available for the users to watch and have a good time.

## 1.3 APPLICATIONS

- For watching videos of different genres at one place
- To watch vast content of different countries at one place

## 1.4 LIMITATIONS

- Need constant maintenance to make sure the videos are available in a click away even though more and more content is added.
- Need wifi connection to make sure the buffering won’t take long time as usual.

# SYSTEM ANALYSIS

## 2.1 EXISTING SYSTEM

- Video streaming can’t be done without without a CDN with a streaming support. If you have breaking concerns news or live events, you’re probably going to want to live stream it.
- Outages,slow clouds, and CDN issues — what looks like buffering or stalled screens to viewers —is the thing that will annoy users and it will probably cause user abandonment and loss of revenue.
- As audiences grow, content distributors need a way to ensure seamless, high-quality delivery. To scale globally while improving customer satisfaction, you need a way to optimize traffic management intelligently with user-by-user, second-by-second granularity.
- Real user measurement (real-time pulse of user experience, even in sparsely-served regions), real-time server health, and third party metrics (to guide and monitor efficient use of bandwidth resources) give you an essential level of control.

## 2.2 PROPOSED SYSTEM

In this project, we are using MySql database management system to keep data. As we know, we should keep information about videos and users into RDBMS. Other information about videos called metadata should be kept too.

In this project, we have three main tables to store data

1. User
2. Video
3. Video Comment

**User Table**

- UserID (primary key)
- firstName(varchar)
- lastName(varchar)
- username(varchar)
- email(varchar)
- password(varchar)
- signupDate(date time)

**Video Table**

- VideoId (primarykey)
- Title (varchar)
- description(varchar)
- filepath(varchar)
- isMovie(boolean)
- uploadDate(date time)
- releaseDate(date time)
- views(int)
- duration(time)
- season(int)
- episode(int)
- entityId(foreign key)

**Video Progress Table**

- id(primary key)
- username(foreign key)
- videoId(foreign key)
- progress(varchar)
- finished(boolean)
- dateModified(date time)

**Categories**

- Id (primary key)
- GenreName(varchar)

**Entities**

- id(primary key)
- Name (varchar)
- thumbnail(varchar)
- preview(varchar)
- categoryId(foreign key)

**Billing Details**

- id(primary key)
- agreementId(number)
- nextBillingDate(Date)
- token(varchar)
- username(foreign key)

# REQUIREMENT SPECIFICATION

## 3.1 HARDWARE REQUIREMENTS

- Chrome 73.0 or higher, Safari 8.0 or higher
- Internet speed of 1MBPS or higher is required for smooth functioning.

## 3.2 SOFTWARE REQUIREMENTS

**i) Front end development**

- **HTML**

Hyper Text Markup Language (HTML) is the backbone of any website development process, without which a web page doesn't exist. Hypertext means that text has links, termed hyperlinks, embedded in it. A markup language indicates text can be turned into images, tables, links, and other representations. It is the HTML code that provides an overall framework of how the site will look. HTML was developed by Tim Berners-Lee. The latest version of HTML is called HTML5 and was published on October 28, 2014 by the W3 recommendation.

- **CSS**

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language like HTML. CSS is designed to enable the separation of presentation and content, including layout, colours, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file, and reduce complexity and repetition in the structural content.

- **JavaScript**

JavaScript often abbreviated as JS, is a high-level, interpreted programming language. Alongside HTML and CSS, JavaScript is one of the three core technologies of the World Wide Web. JavaScript enables interactive web pages and thus is an essential part of web applications. The vast majority of websites u se it, and all major web browsers have a dedicated JavaScript engine to execute it. JavaScript code can use the Document Object Model (DOM), provided by the HTML standard, to manipulate a web page in response to events, like user input. Using a technique called AJAX, JavaScript code can also actively retrieve content from the web and also react to server-side events as well, adding a truly dynamic nature to the web page experience.

**ii) Server-Side Programming**

- **PHP**

Hypertext Preprocessor (or simply PHP) is a server-side scripting language designed for Web development. PHP code may be embedded into HTML code, or it can be used in combination with various web template systems, web content management systems, and web frameworks. PHP code is usually processed by a PHP interpreter implemented as a module in the web server or as a Common Gateway Interface (CGI) executable. The web server combines the results of the interpreted and executed PHP code, which may be any type of data, including images, with the generated web page. PHP code may also be executed with a command-line interface(CLI) and can be used to implement standalone graphical applications.

- **MySQL**

MySQL is an open-source relational database management system (RDBMS). It is particularly useful in handling structured data where there are relations between different entities/variables of the data. It uses the standard SQL. MySQL is the de-facto standard database system for web sites with huge volumes of both data and end-users (like Facebook, Twitter, and Wikipedia). Another great thing about MySQL is that it can be scaled down to support embedded database applications.

- Advanced Technologies used : **AJAX**

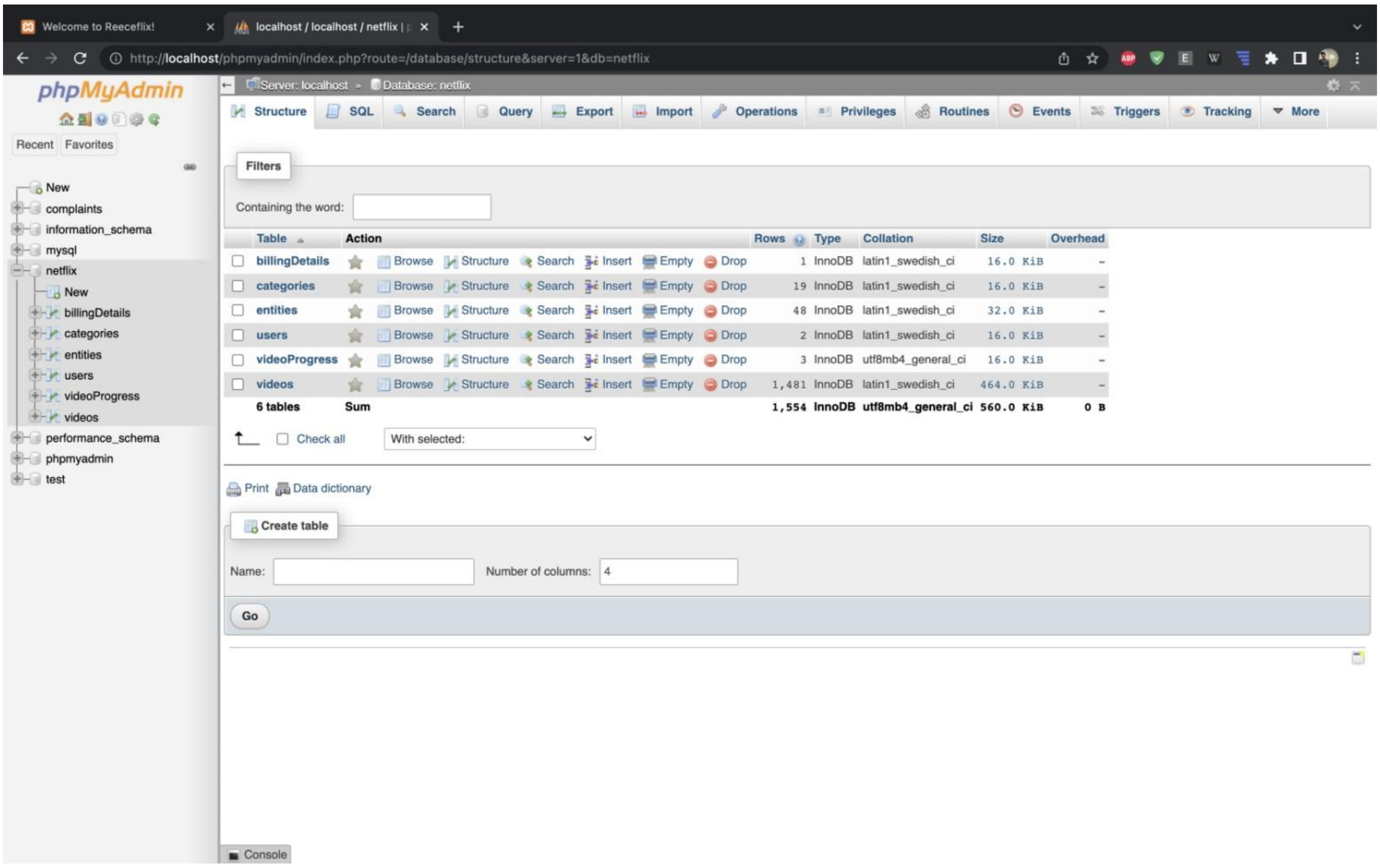
AJAX = Asynchronous JavaScript and XML. AJAX is not a programming language.
AJAX uses only a combination of:
Built-in browser XMLHttpRequest item (requesting data from web server)
JavaScript and HTML DOM (display or use data)

AJAX is a misleading word. AJAX applications may use XML to move data, but it is equally common to move data such as plain text or JSON text. AJAX allows web pages to be updated automatically by data exchange with a web server in secret. This means that you may have to refresh portions of a web page, without having to reload the entire page.

AJAX is a developer's dream, because you can:

- Update a web page without reloading the page
- Request data from a server - after the page has loaded
- Receive data from a server - after the page has loaded
- Send data to a server - in the background

## 4.3 DATABASE DESIGN



http://localhost/phpmyadmin/index.php?route=/database/structure&server=1&db=netflix

Containing the word:

| | Table | Action | | | | | | | Rows | Type | Collation | Size | Overhead |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| □ | billingDetails | ☆ | Browse | Structure | Search | Insert | Empty | Drop | 1 | InnoDB | latin1_swedish_ci | 16.0 KiB | - |
| □ | categories | ☆ | Browse | Structure | Search | Insert | Empty | Drop | 19 | InnoDB | latin1_swedish_ci | 16.0 KiB | - |
| □ | entities | ☆ | Browse | Structure | Search | Insert | Empty | Drop | 48 | InnoDB | latin1_swedish_ci | 32.0 KiB | - |
| □ | users | ☆ | Browse | Structure | Search | Insert | Empty | Drop | 2 | InnoDB | latin1_swedish_ci | 16.0 KiB | - |
| □ | videoProgress | ☆ | Browse | Structure | Search | Insert | Empty | Drop | 3 | InnoDB | utf8mb4_general_ci | 16.0 KiB | - |
| □ | videos | ☆ | Browse | Structure | Search | Insert | Empty | Drop | 1,481 | InnoDB | latin1_swedish_ci | 464.0 KiB | - |
| | 6 tables | Sum | | | | | | | 1,554 | InnoDB | utf8mb4_general_ci | 560.0 KiB | 0 B |

Name:
Number of columns: 4

Fig 4.3 : Design of the database

# SYSTEM IMPLEMENTATION

## 5.1 MODULE DESCRIPTION

**Accounts :** In this module user can register by entering their profile information. After registration user can access accounts page by entering login id and password.

**Video manager :** This module allows administrator to upload videos and they can edit uploaded video description and its contents.

**Dashboard module :** The user can view the available movie titles or series titles to watch.

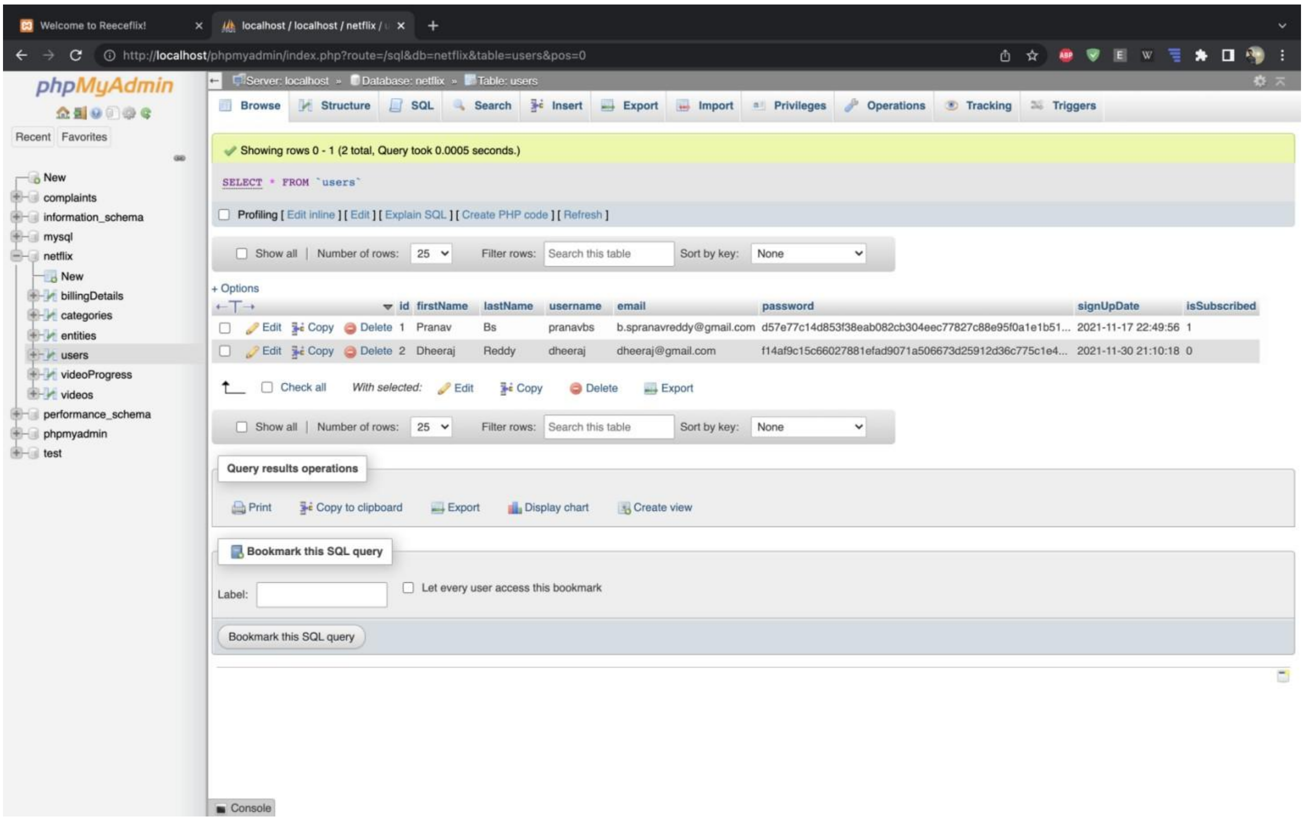
**Content module :** This module allows the user to watch any shows on the website.

**Subscription :** In this module user can subscribe for using the website. After confirmation of the subscription the user can watch videos. The subscribed member can download the videos by clicking download button.

**Comments and likes :** The registered user can post their comments and like the uploaded videos.

# SQL Tables (as seen in phpMyAdmin)

## i) Users table



http://localhost/phpmyadmin/index.php?route=/sql&db=netflix&table=users&pos=0

Server: localhost » Database: netflix » Table: users

✓ Showing rows 0 - 1 (2 total, Query took 0.0005 seconds.)

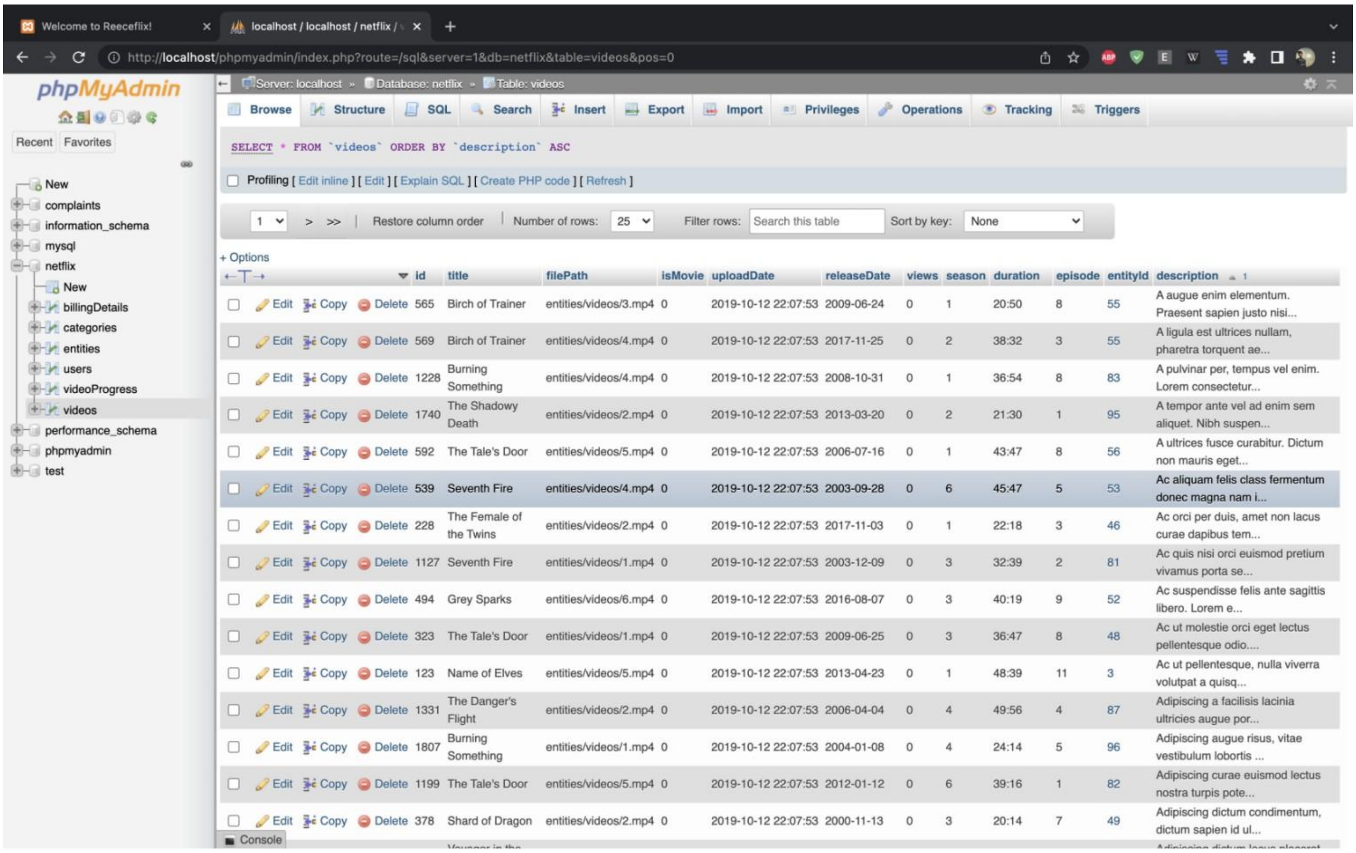
```
SELECT * FROM `users`
```

Number of rows: 25 | Sort by key: None

| id | firstName | lastName | username | email | password | signUpDate | isSubscribed |
|---|---|---|---|---|---|---|---|
| 1 | Pranav | Bs | pranavbs | b.spranavreddy@gmail.com | d57e77c14d853f38eab082cb304eec77827c88e95f0a1e1b51... | 2021-11-17 22:49:56 | 1 |
| 2 | Dheeraj | Reddy | dheeraj | dheeraj@gmail.com | f14af9c15c66027881efad9071a506673d25912d36c775c1e4... | 2021-11-30 21:10:18 | 0 |

Fig 5.A Table which records users' information when they sign in

## ii) Videos Table



http://localhost/phpmyadmin/index.php?route=/sql&server=1&db=netflix&table=videos&pos=0

Server: localhost » Database: netflix » Table: videos

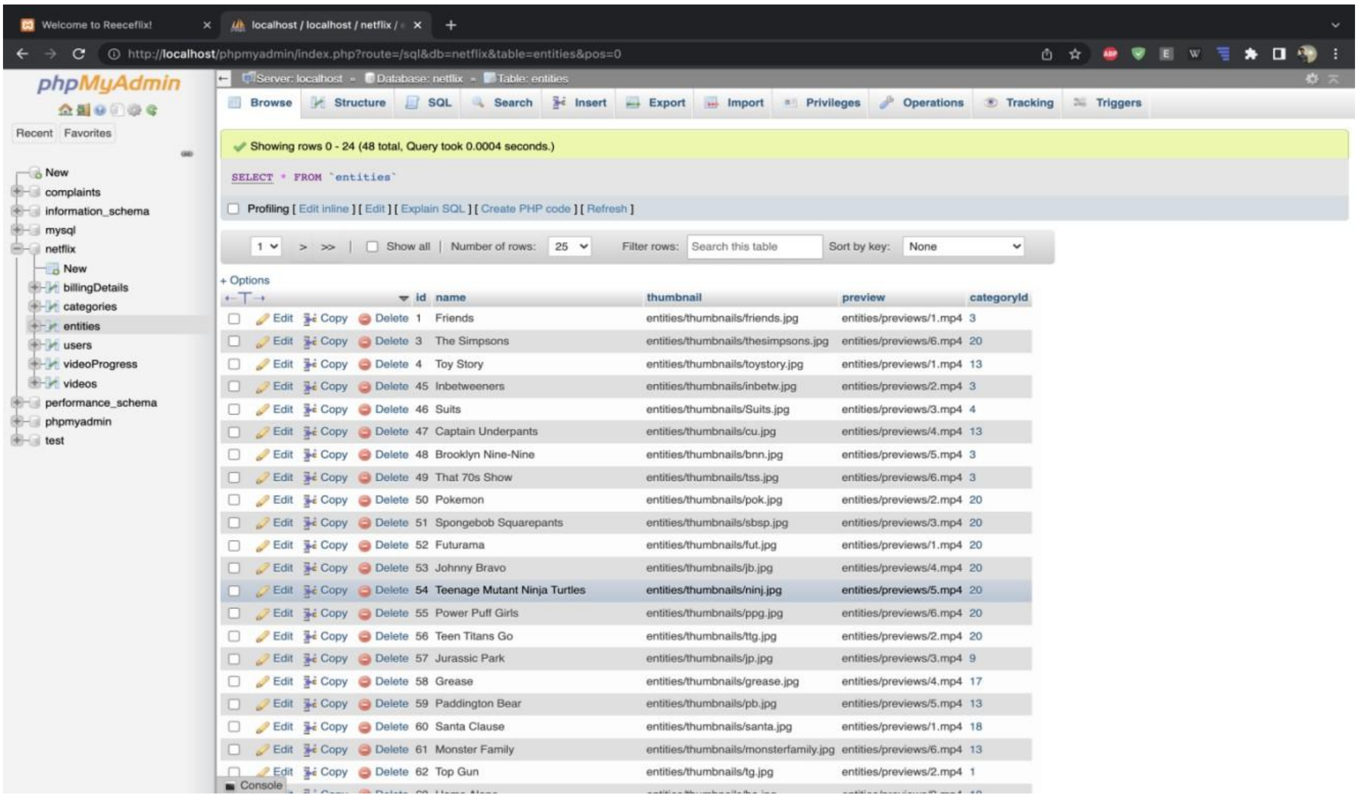
```
SELECT * FROM `videos` ORDER BY `description` ASC
```

Number of rows: 25 | Sort by key: None

| id | title | filePath | isMovie | uploadDate | releaseDate | views | season | duration | episode | entityId | description |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 565 | Birch of Trainer | entities/videos/3.mp4 | 0 | 2019-10-12 22:07:53 | 2009-06-24 | 0 | 1 | 20:50 | 8 | 55 | A augue enim elementum. Praesent sapien justo nisi... |
| 569 | Birch of Trainer | entities/videos/4.mp4 | 0 | 2019-10-12 22:07:53 | 2017-11-25 | 0 | 2 | 38:32 | 3 | 55 | A ligula est ultrices nullam, pharetra torquent ae... |
| 1228 | Burning Something | entities/videos/4.mp4 | 0 | 2019-10-12 22:07:53 | 2008-10-31 | 0 | 1 | 36:54 | 8 | 83 | A pulvinar per, tempus vel enim. Lorem consectetur... |
| 1740 | The Shadowy Death | entities/videos/2.mp4 | 0 | 2019-10-12 22:07:53 | 2013-03-20 | 0 | 2 | 21:30 | 1 | 95 | A tempor ante vel ad enim sem aliquet. Nibh suspen... |
| 592 | The Tale's Door | entities/videos/5.mp4 | 0 | 2019-10-12 22:07:53 | 2006-07-16 | 0 | 1 | 43:47 | 8 | 56 | A ultrices fusce curabitur. Dictum non mauris eget... |
| 539 | Seventh Fire | entities/videos/4.mp4 | 0 | 2019-10-12 22:07:53 | 2003-09-28 | 0 | 6 | 45:47 | 5 | 53 | Ac aliquam felis class fermentum donec magna nam i... |
| 228 | The Female of the Twins | entities/videos/2.mp4 | 0 | 2019-10-12 22:07:53 | 2017-11-03 | 0 | 1 | 22:18 | 3 | 46 | Ac orci per duis, amet non lacus curae dapibus tem... |
| 1127 | Seventh Fire | entities/videos/1.mp4 | 0 | 2019-10-12 22:07:53 | 2003-12-09 | 0 | 3 | 32:39 | 2 | 81 | Ac quis nisi orci euismod pretium vivamus porta se... |
| 494 | Grey Sparks | entities/videos/6.mp4 | 0 | 2019-10-12 22:07:53 | 2016-08-07 | 0 | 3 | 40:19 | 9 | 52 | Ac suspendisse felis ante sagittis libero. Lorem e... |
| 323 | The Tale's Door | entities/videos/1.mp4 | 0 | 2019-10-12 22:07:53 | 2009-06-25 | 0 | 3 | 36:47 | 8 | 48 | Ac ut molestie orci eget lectus pellentesque odio.... |
| 123 | Name of Elves | entities/videos/5.mp4 | 0 | 2019-10-12 22:07:53 | 2013-04-23 | 0 | 1 | 48:39 | 11 | 3 | Ac ut pellentesque, nulla viverra volutpat a quisq... |
| 1331 | The Danger's Flight | entities/videos/2.mp4 | 0 | 2019-10-12 22:07:53 | 2006-04-04 | 0 | 4 | 49:56 | 4 | 87 | Adipiscing a facilisis lacinia ultricies augue por... |
| 1807 | Burning Something | entities/videos/1.mp4 | 0 | 2019-10-12 22:07:53 | 2004-01-08 | 0 | 4 | 24:14 | 5 | 96 | Adipiscing augue risus, vitae vestibulum lobortis ... |
| 1199 | The Tale's Door | entities/videos/5.mp4 | 0 | 2019-10-12 22:07:53 | 2012-01-12 | 0 | 6 | 39:16 | 1 | 82 | Adipiscing curae euismod lectus nostra turpis pote... |
| 378 | Shard of Dragon | entities/videos/2.mp4 | 0 | 2019-10-12 22:07:53 | 2000-11-13 | 0 | 3 | 20:14 | 7 | 49 | Adipiscing dictum condimentum, dictum sapien id ul... |

Fig 5.B Loading the database with the videos for the user

## iii) Entities Table



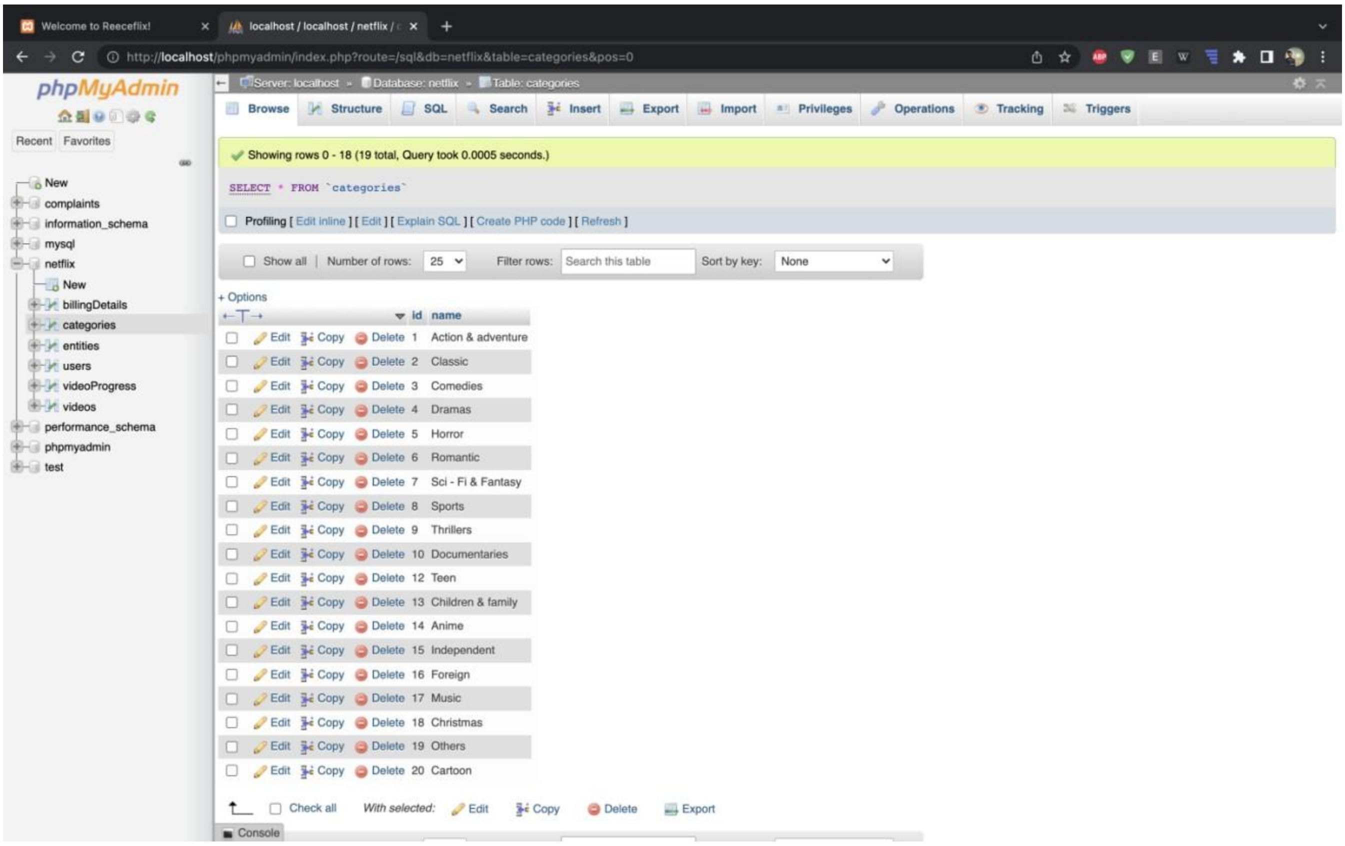
Showing rows 0 - 24 (48 total, Query took 0.0004 seconds.)

```
SELECT * FROM `entities`
```

| id | name | thumbnail | preview | categoryId |
|---|---|---|---|---|
| 1 | Friends | entities/thumbnails/friends.jpg | entities/previews/1.mp4 | 3 |
| 3 | The Simpsons | entities/thumbnails/thesimpsons.jpg | entities/previews/6.mp4 | 20 |
| 4 | Toy Story | entities/thumbnails/toystory.jpg | entities/previews/1.mp4 | 13 |
| 45 | Inbetweeners | entities/thumbnails/inbetw.jpg | entities/previews/2.mp4 | 3 |
| 46 | Suits | entities/thumbnails/Suits.jpg | entities/previews/3.mp4 | 4 |
| 47 | Captain Underpants | entities/thumbnails/cu.jpg | entities/previews/4.mp4 | 13 |
| 48 | Brooklyn Nine-Nine | entities/thumbnails/bnn.jpg | entities/previews/5.mp4 | 3 |
| 49 | That 70s Show | entities/thumbnails/tss.jpg | entities/previews/6.mp4 | 3 |
| 50 | Pokemon | entities/thumbnails/pok.jpg | entities/previews/2.mp4 | 20 |
| 51 | Spongebob Squarepants | entities/thumbnails/sbsp.jpg | entities/previews/3.mp4 | 20 |
| 52 | Futurama | entities/thumbnails/fut.jpg | entities/previews/1.mp4 | 20 |
| 53 | Johnny Bravo | entities/thumbnails/jb.jpg | entities/previews/4.mp4 | 20 |
| 54 | Teenage Mutant Ninja Turtles | entities/thumbnails/ninj.jpg | entities/previews/5.mp4 | 20 |
| 55 | Power Puff Girls | entities/thumbnails/ppg.jpg | entities/previews/6.mp4 | 20 |
| 56 | Teen Titans Go | entities/thumbnails/ttg.jpg | entities/previews/2.mp4 | 20 |
| 57 | Jurassic Park | entities/thumbnails/jp.jpg | entities/previews/3.mp4 | 9 |
| 58 | Grease | entities/thumbnails/grease.jpg | entities/previews/4.mp4 | 17 |
| 59 | Paddington Bear | entities/thumbnails/pb.jpg | entities/previews/5.mp4 | 13 |
| 60 | Santa Clause | entities/thumbnails/santa.jpg | entities/previews/1.mp4 | 18 |
| 61 | Monster Family | entities/thumbnails/monsterfamily.jpg | entities/previews/6.mp4 | 13 |
| 62 | Top Gun | entities/thumbnails/tg.jpg | entities/previews/2.mp4 | 1 |

Fig 5.C Loading up thumbnails and previews for the user

## iv) Categories Table



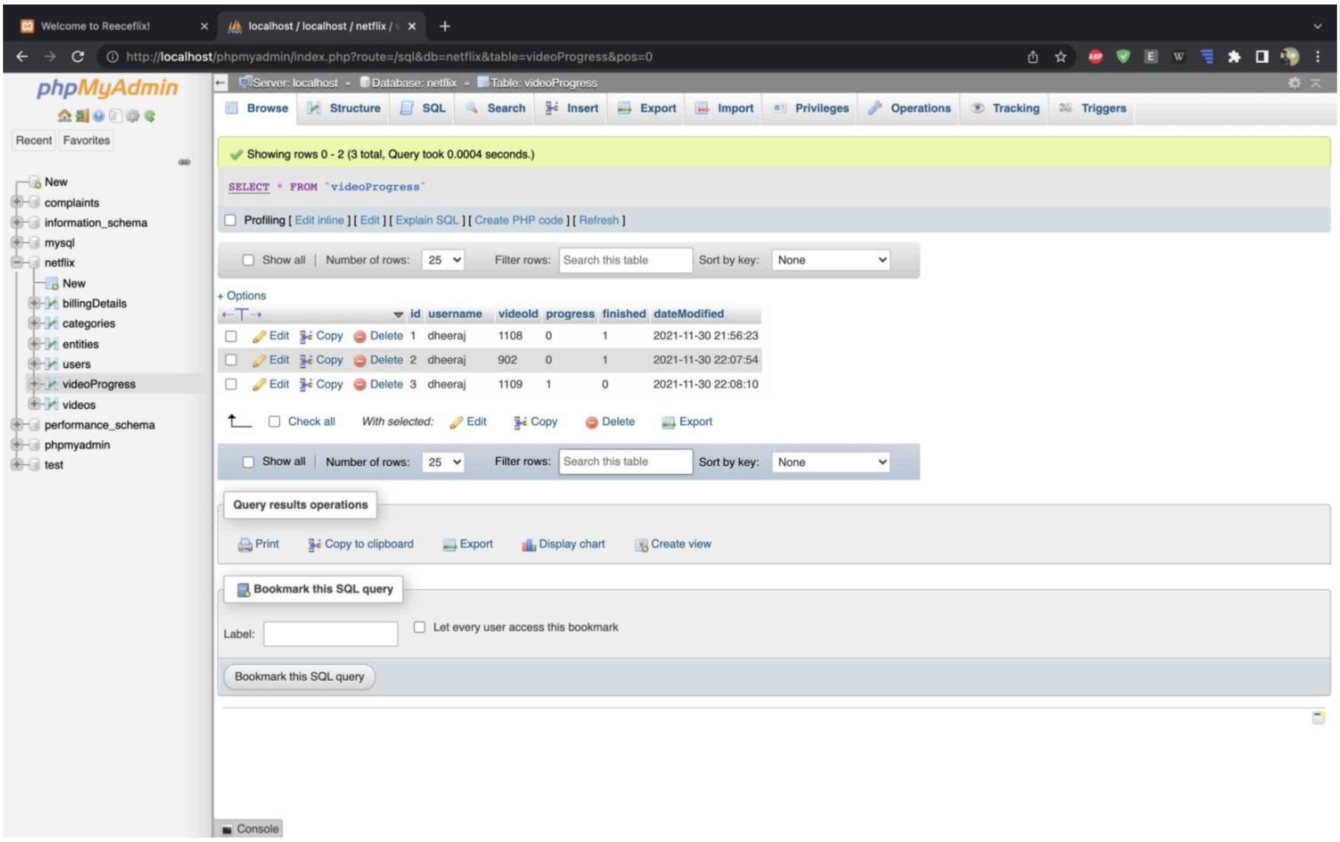
Showing rows 0 - 18 (19 total, Query took 0.0005 seconds.)

```
SELECT * FROM `categories`
```

| id | name |
|---|---|
| 1 | Action & adventure |
| 2 | Classic |
| 3 | Comedies |
| 4 | Dramas |
| 5 | Horror |
| 6 | Romantic |
| 7 | Sci - Fi & Fantasy |
| 8 | Sports |
| 9 | Thrillers |
| 10 | Documentaries |
| 12 | Teen |
| 13 | Children & family |
| 14 | Anime |
| 15 | Independent |
| 16 | Foreign |
| 17 | Music |
| 18 | Christmas |
| 19 | Others |
| 20 | Cartoon |

Fig 5.D Providing the genres with different id

## v) Video Progress Table



http://localhost/phpmyadmin/index.php?route=/sql&db=netflix&table=videoProgress&pos=0

Server: localhost » Database: netflix » Table: videoProgress

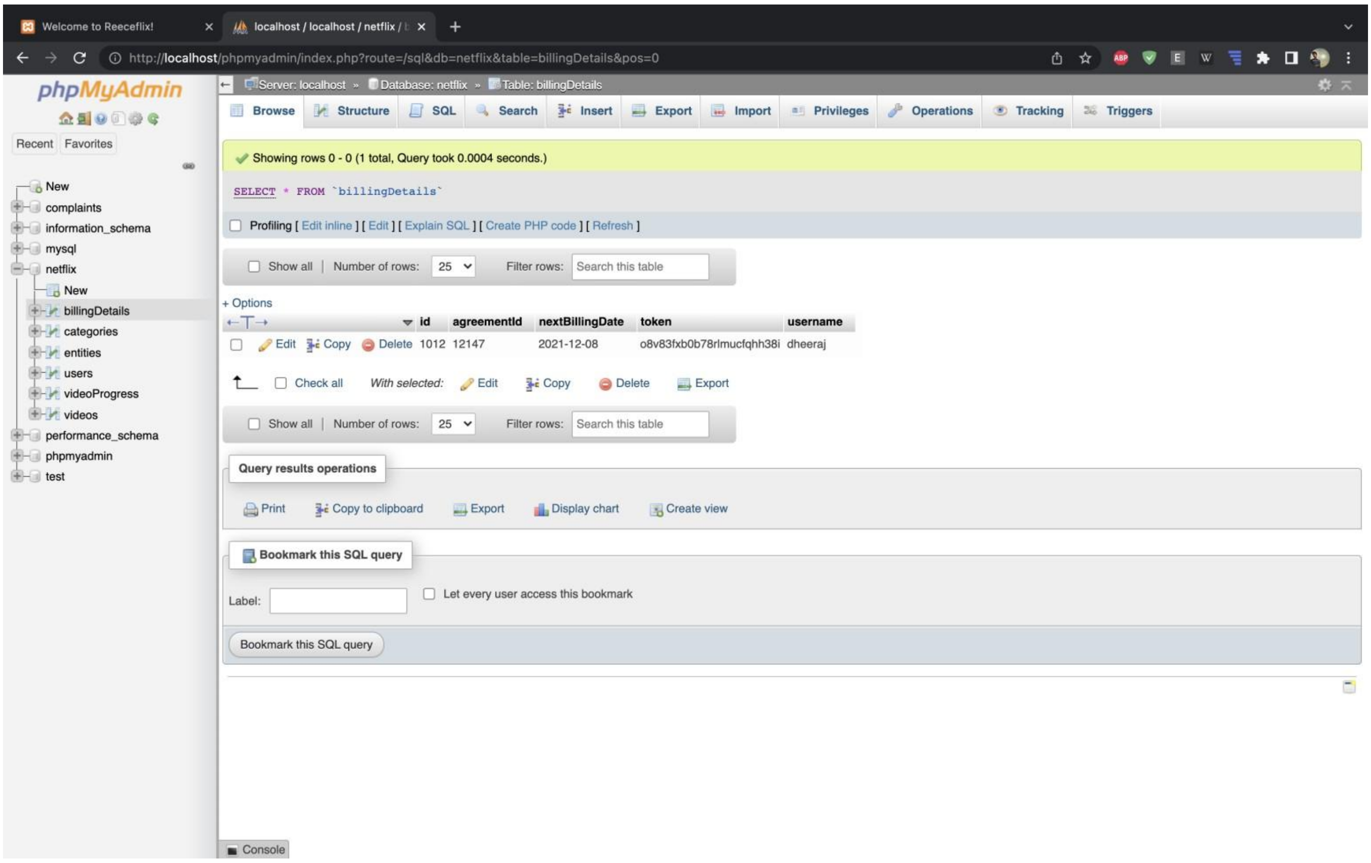
Showing rows 0 - 2 (3 total, Query took 0.0004 seconds.)

```
SELECT * FROM `videoProgress`
```

| id | username | videoId | progress | finished | dateModified |
|---|---|---|---|---|---|
| 1 | dheeraj | 1108 | 0 | 1 | 2021-11-30 21:56:23 |
| 2 | dheeraj | 902 | 0 | 1 | 2021-11-30 22:07:54 |
| 3 | dheeraj | 1109 | 1 | 0 | 2021-11-30 22:08:10 |

Fig 5.E Table to record user viewing status of the videos.

## vi) Billing Table



http://localhost/phpmyadmin/index.php?route=/sql&db=netflix&table=billingDetails&pos=0

Server: localhost » Database: netflix » Table: billingDetails

Showing rows 0 - 0 (1 total, Query took 0.0004 seconds.)

```
SELECT * FROM `billingDetails`
```

| id | agreementId | nextBillingDate | token | username |
|---|---|---|---|---|
| 1012 | 12147 | 2021-12-08 | o8v83fxb0b78rlmucfqhh38i | dheeraj |

Fig 5.F Table to record user billing details.

# CONCLUSION AND FUTURE WORKS

This project is being built across different development areas, the primary features included are uploading videos by users, sharing of videos through social media and viewing of videos uploaded by others. The videos can also be downloaded, edited and uploaded online again. The project has different modules in the development work, which are divided among administrators, users, web registrations and search option. Being an online platform, it provides the flexibility of working on the project from remote locations with proper access codes.

Some additional features which can be added to make the project more interactive are: Uploading and sharing videos using embedded links, Setting up channels to categorize videos, broadcasting events/ meetings using invitation links and setting up advertisement avenues to gain income.

# APPENDICES

## 7.1 APPENDIX 1

**User login page written in php**

```
<?php
require_once("includes/config.php");
require_once("includes/classes/FormSanitizer.php");
require_once("includes/classes/Account.php"); require_once("includes/classes/Constants.php");

$account = new Account($con);

if (isset($_POST["submitButton"])) {
    $username = FormSanitizer::sanitizeFormUsername($_POST["username"]);
    $password = FormSanitizer::sanitizeFormPassword($_POST["password"]);

    $success = $account->login($username, $password);
    if ($success) {
        $_SESSION["userLoggedIn"] = $username; header("Location: index.php");
    }
}

function getInputValue($name)
  {
      if (isset($_POST[$name]))
         { echo $_POST[$name];
      }
}
?>
```

```
<!DOCTYPE html>
<html lang="en" dir="ltr">
<head>
    <meta charset="utf-8">
    <title>Welcome to Reeceflix!</title>
    <link rel="stylesheet" type="text/css" href="assets/style/style.css" />
</head>
<body>
    <div class="signInContainer">
    <div class="column">
    <div class="header">
    <img src="assets/images/logo.png" title="Logo" alt="Site logo">
    <h3>Sign In</h3>
    <span>to continue to Reeceflix</span>
    </div>
    <form class="" method="POST">
    <?php echo $account->getError(Constants::$loginFailed); ?>
    <input type="text" name="username" placeholder="Username" value="<?php
       getInputValue("username");?>" required>
    <input type="password" name="password" placeholder="Password" required>
    <input type="submit" name="submitButton" value="SUBMIT">
    </form>
    <a href="register.php" class="signInMessage">Need an account? Sign in here</a>
    </div>
    </div>
</body>
</html>
```

### Connection to the database

config.php

```
<?php
ob_start(); // Turns on output buffering.
session_start(); // We are able to use sessions.
date_default_timezone_set("Europe/Budapest"); // Setting the time zone.
try {
    $con = new PDO("mysql:dbname=video_player;host=localhost", "root", " ");
    $con->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_WARNING);
} catch (PDOException $e) {
    exit("Connection failed: " . $e->getMessage());
 }
?>
```

### AJAX scripts

addDuration.php

```
<?php
require_once("../includes/config.php");
if (isset($_POST["videoId"]) && isset($_POST["username"])) {
    $query = $con->prepare("SELECT * FROM videoProgress WHERE username=:username
        AND videoId=:videoId");
    $query->bindValue(":username", $_POST["username"]);
    $query->bindValue(":videoId", $_POST["videoId"]);
    $query->execute();
if ($query->rowCount() == 0) {
    $query = $con->prepare("INSERT INTO videoProgress (username, videoId) VALUES
        (:username, :videoId)");
    $query->bindValue(":username", $_POST["username"]);
    $query->bindValue(":videoId", $_POST["videoId"]);
    $query->execute();
}
} else {
    echo "No videoId or username passed into file.";
 }
?>
```

**getProgress.php**

```
<?php
require_once("../includes/config.php");
if (isset($_POST["videoId"]) && isset($_POST["username"])) {
    $query = $con->prepare("SELECT progress from videoProgress WHERE
        username=:username AND videoId=:videoId");
    $query->bindValue(":username", $_POST["username"]);
    $query->bindValue(":videoId", $_POST["videoId"]);
    $query->execute();

echo $query->fetchColumn();
} else {
    echo "No videoId or username passed into file.";
 }
?>
```

**getSearchresults.php**

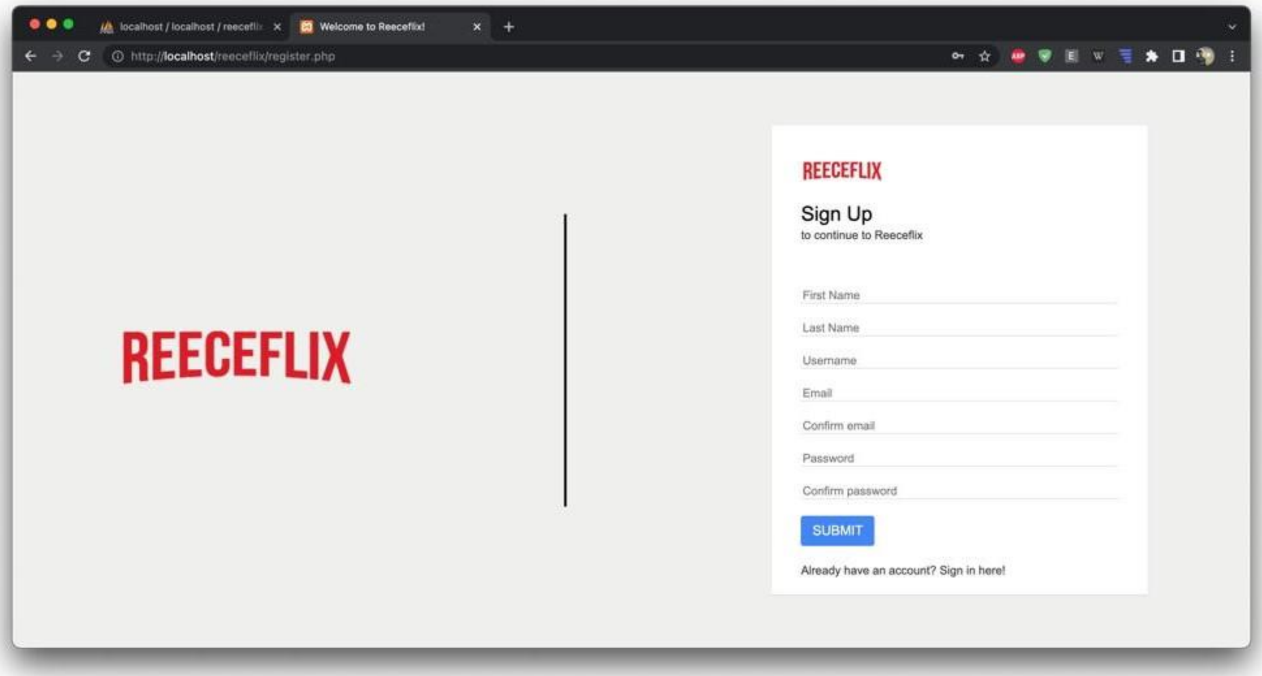
```
<?php
require_once("../includes/config.php");
require_once("../includes/classes/SearchResultsProvider.php");
require_once("../includes/classes/EntityProvider.php");
require_once("../includes/classes/Entity.php");
require_once("../includes/classes/PreviewProvider.php");

if (isset($_POST["term"]) && isset($_POST["username"])) {
    $srp = new SearchResultsProvider($con, $_POST["username"]); echo $srp-
        >getResults($_POST["term"]);
} else {
    echo "No term or username passed into file.";
 }
?>
```

# 7.2 APPENDIX 2

## IMPLEMENTATION SCREENSHOTS

### User Registration


localhost / localhost / reecefli
Welcome to Reeceflix!
http://localhost/reeceflix/register.php
REECEFLIX

REECEFLIX

Sign Up
to continue to Reeceflix

First Name

Last Name

Username

Email

Confirm email

Password

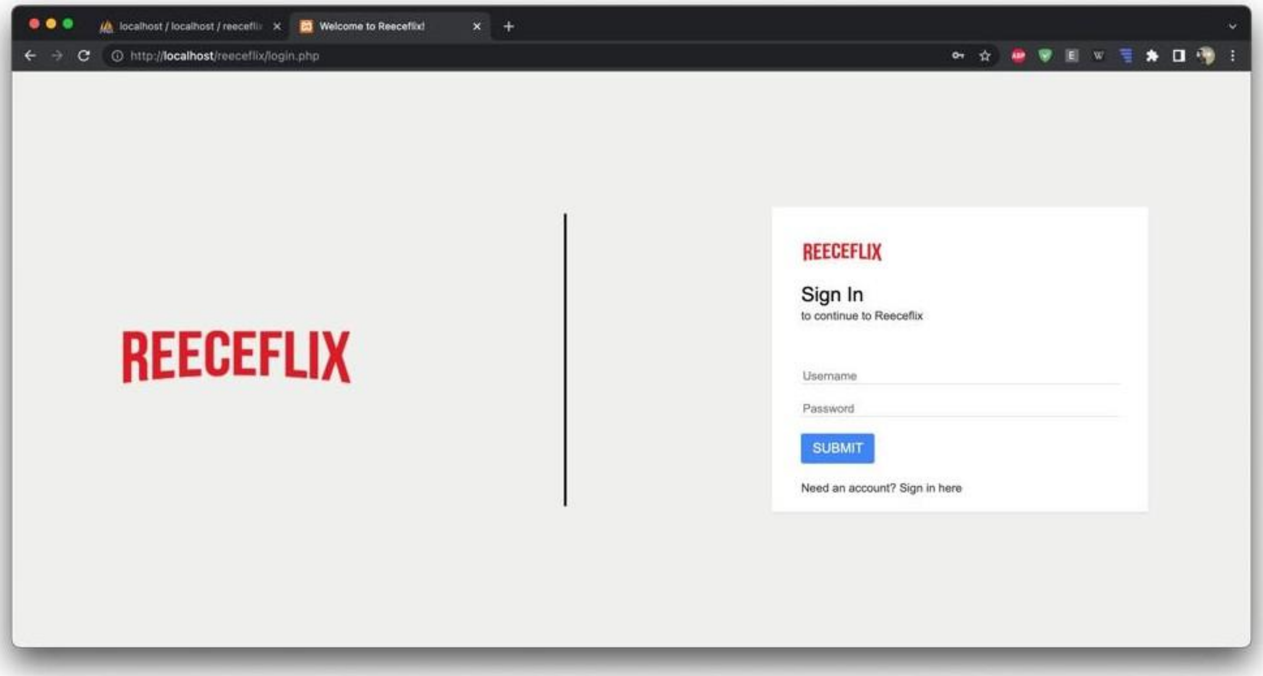
Confirm password

SUBMIT

Already have an account? Sign in here!

**Fig 7.2.A** The image shows the user registration form accessed from site's login page. The user fills up the form to register with the website and information is stored in the users table.

### User Login


localhost / localhost / reecefli
Welcome to Reeceflix!
http://localhost/reeceflix/login.php
REECEFLIX

REECEFLIX

Sign In
to continue to Reeceflix

Username

Password

SUBMIT

Need an account? Sign in here

**Fig 7.2.B** This is a login form if the user has already has an account registered with the database.

## User Homepage

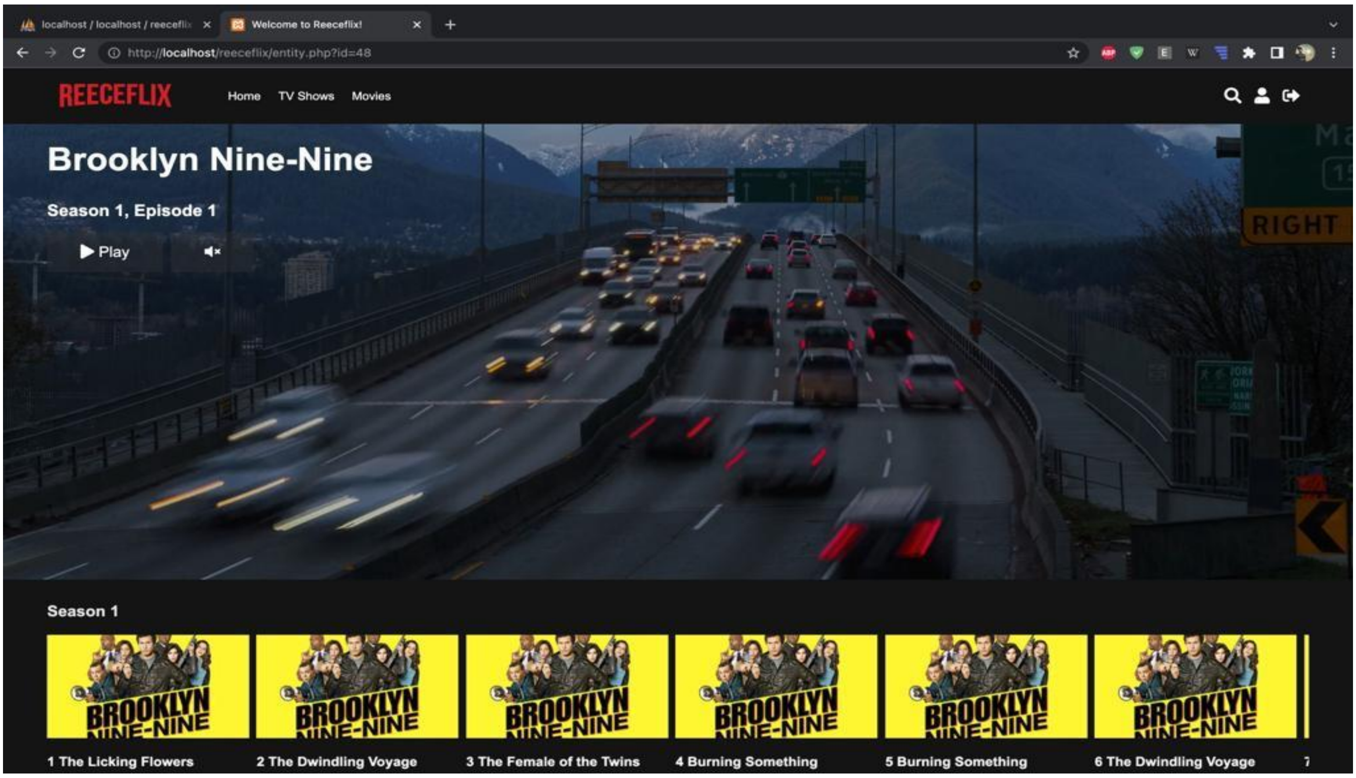


**Fig 7.2.C** The recent viewed video/series's preview will be shown

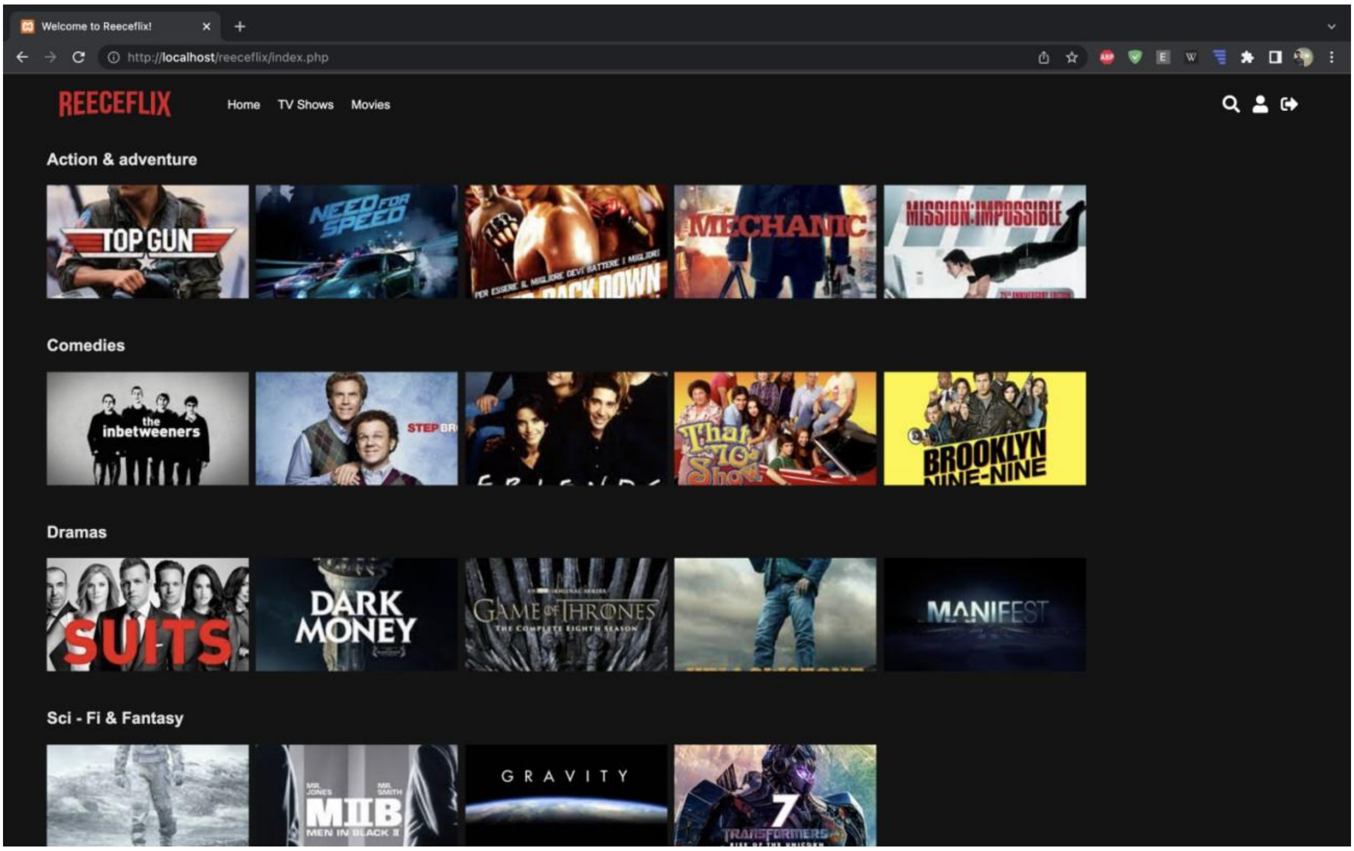


**Fig 7.2.D** Different types of genres are available for the users

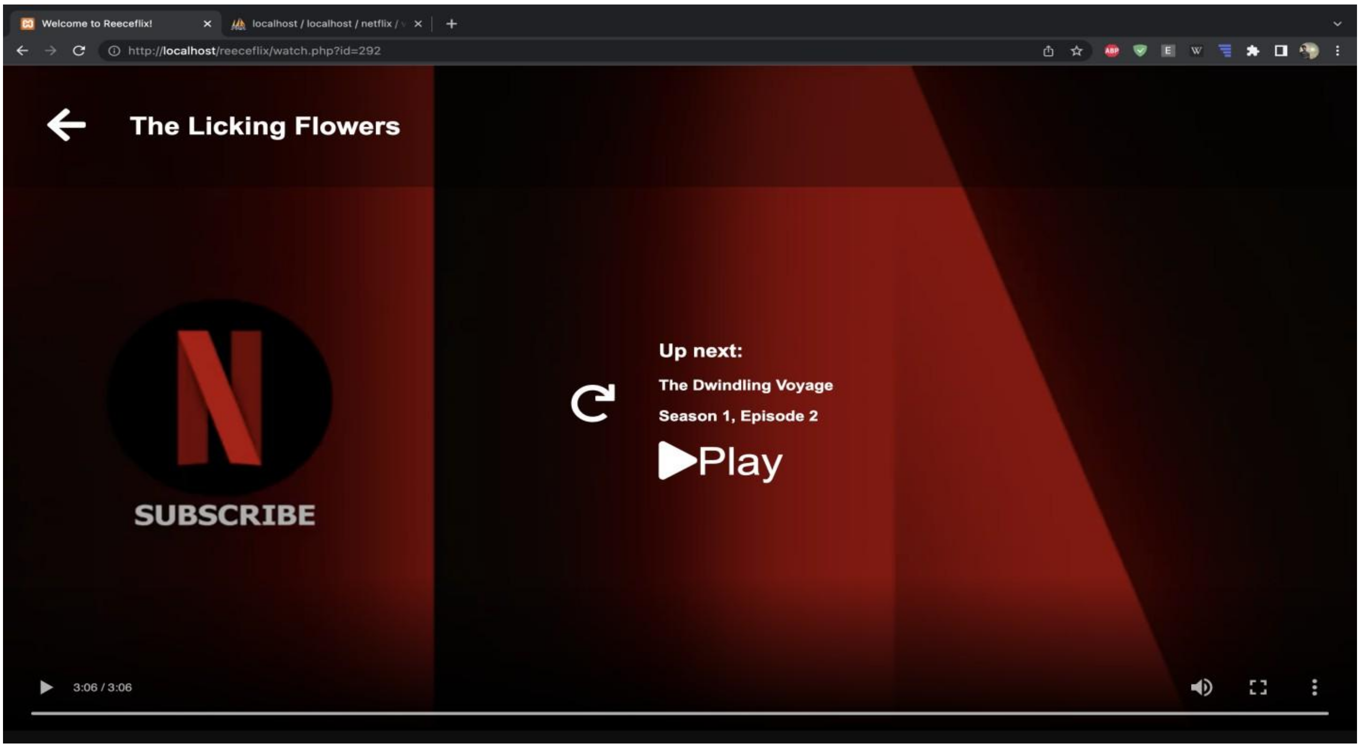


**Fig 7.2.E** A video is played and the user's view is recorded.

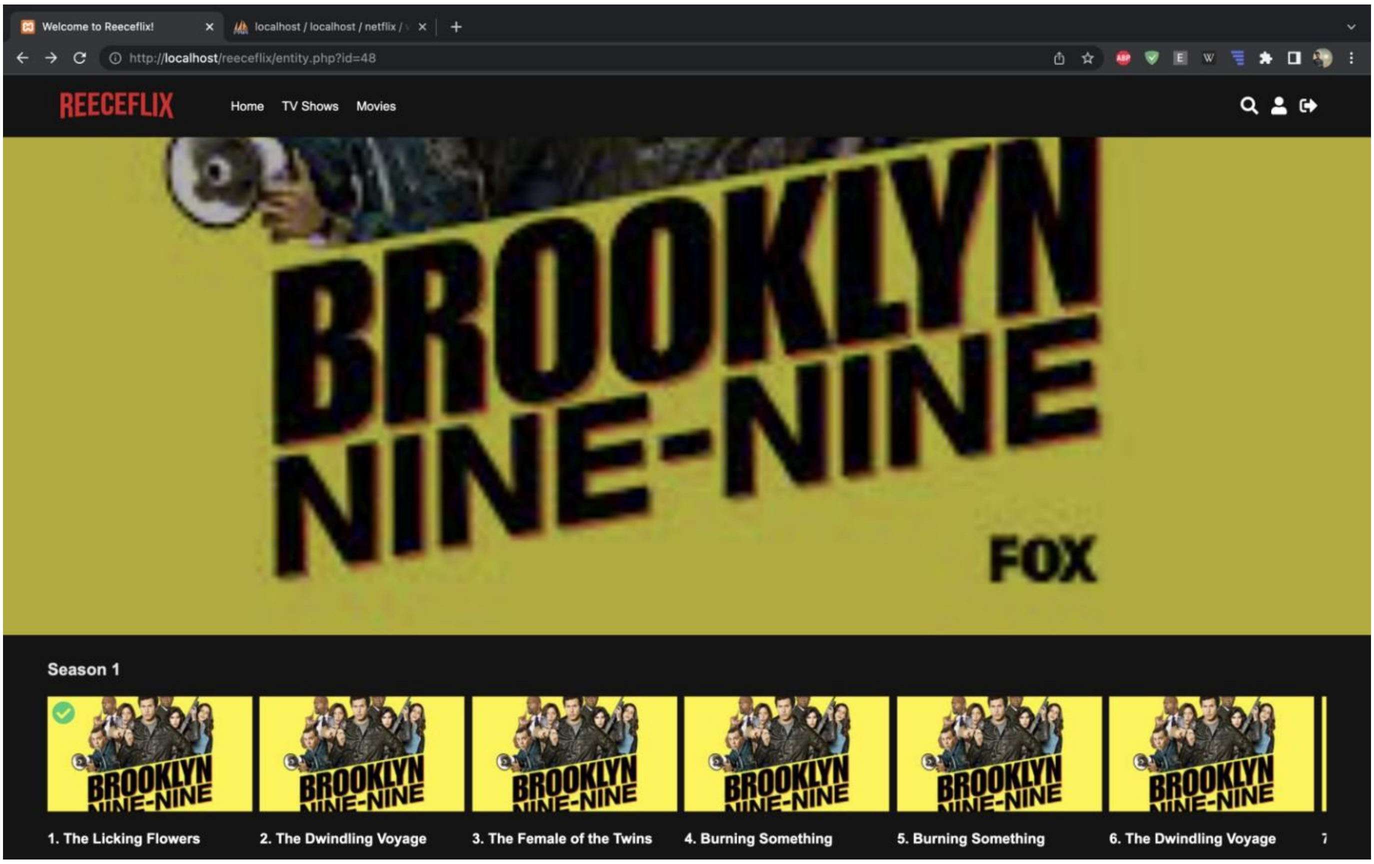
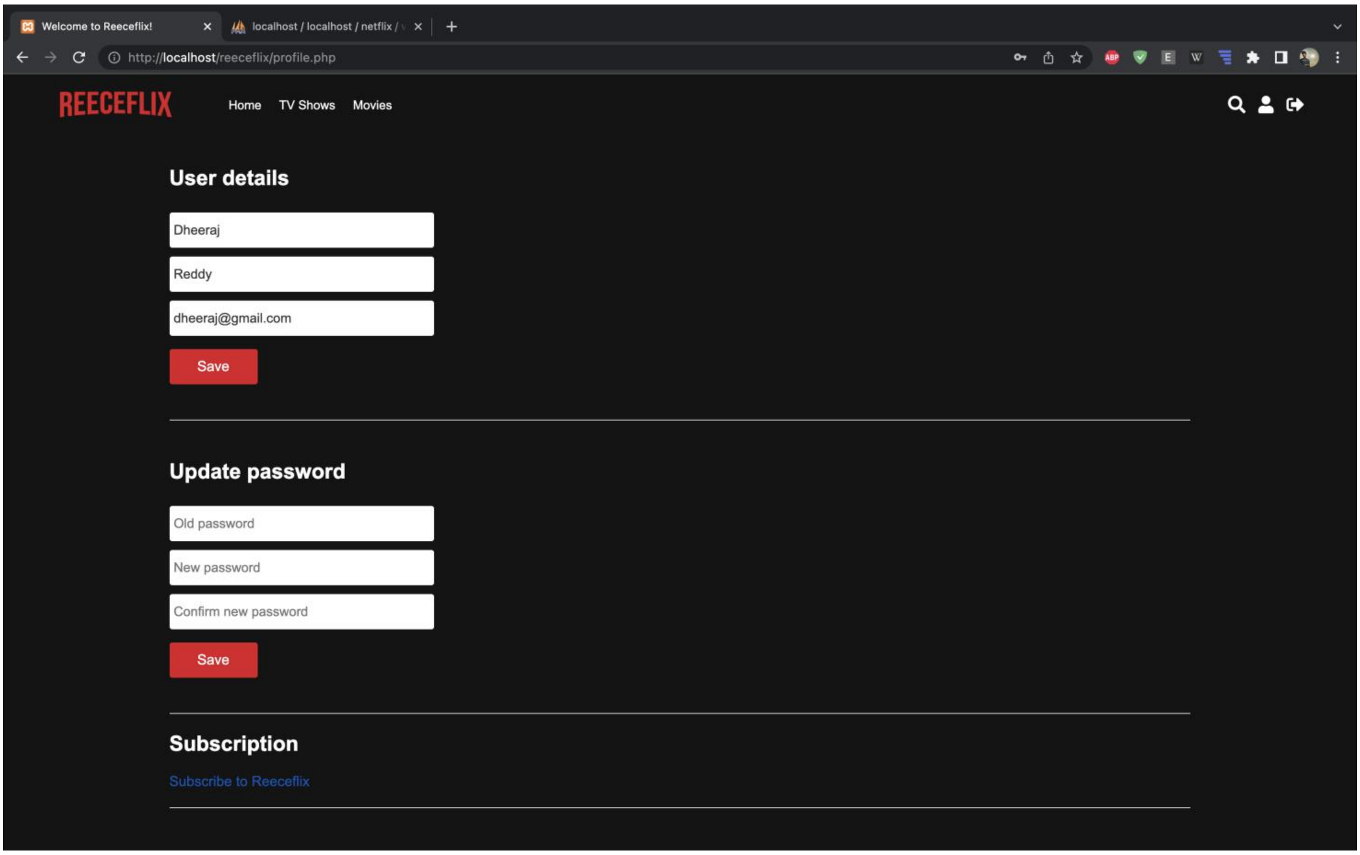


**Fig 7.2.F** All the seasons' episodes are displayed to the user



Welcome to Reeceflix!
localhost / localhost / netflix / ...
http://localhost/reeceflix/profile.php
REECEFLIX
Home
TV Shows
Movies

User details

Dheeraj

Reddy

dheeraj@gmail.com

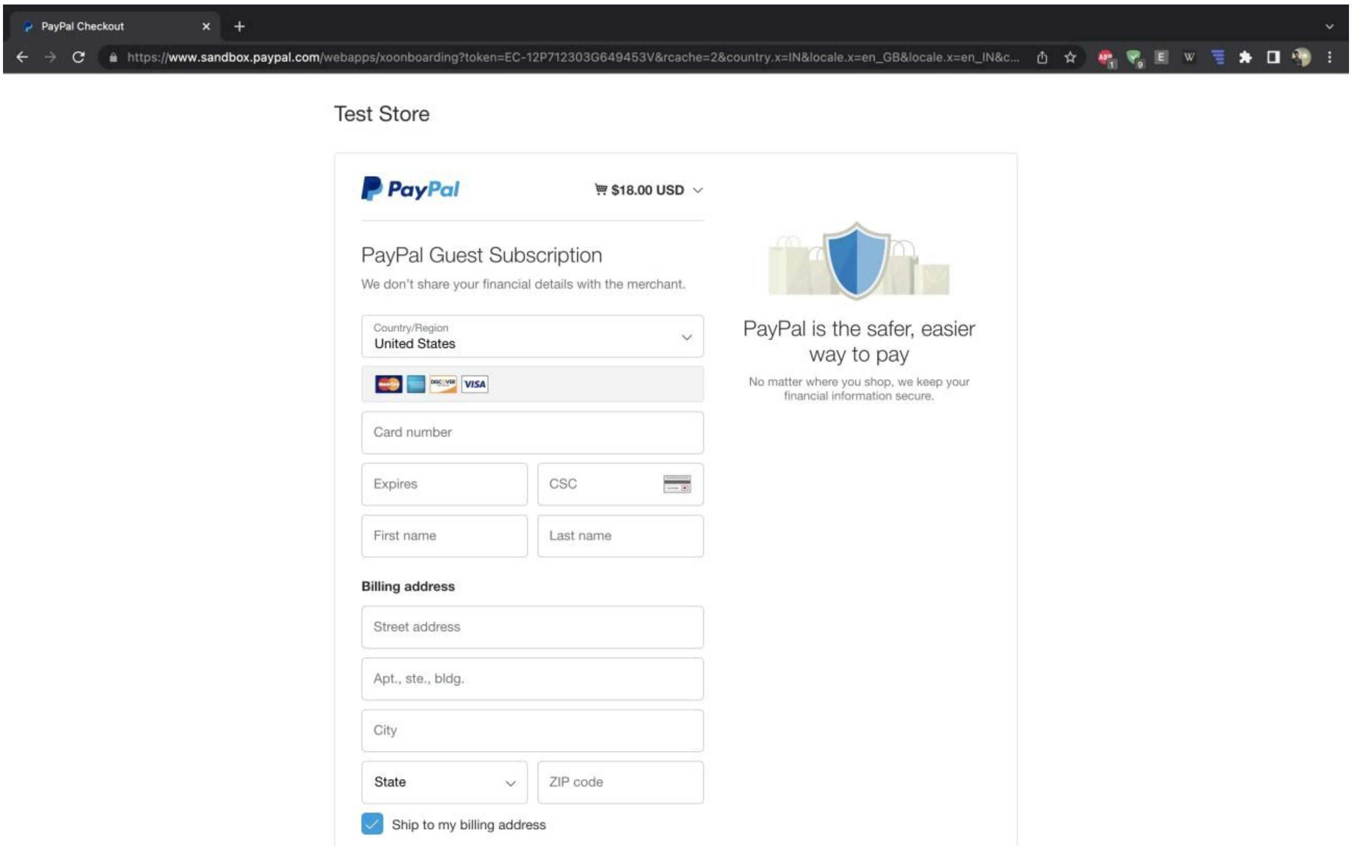
Update password

Subscription

Subscribe to Reeceflix

**Fig 7.2.G** User Profile

## Payment Page



PayPal Checkout
https://www.sandbox.paypal.com/webapps/xoonboarding?token=EC-12P712303G649453V&rcache=2&country.x=IN&locale.x=en_GB&locale.x=en_IN&c...

Test Store

PayPal Guest Subscription

We don't share your financial details with the merchant.

Country/Region: United States

VISA
Card number
Expires
CSC
First name
Last name

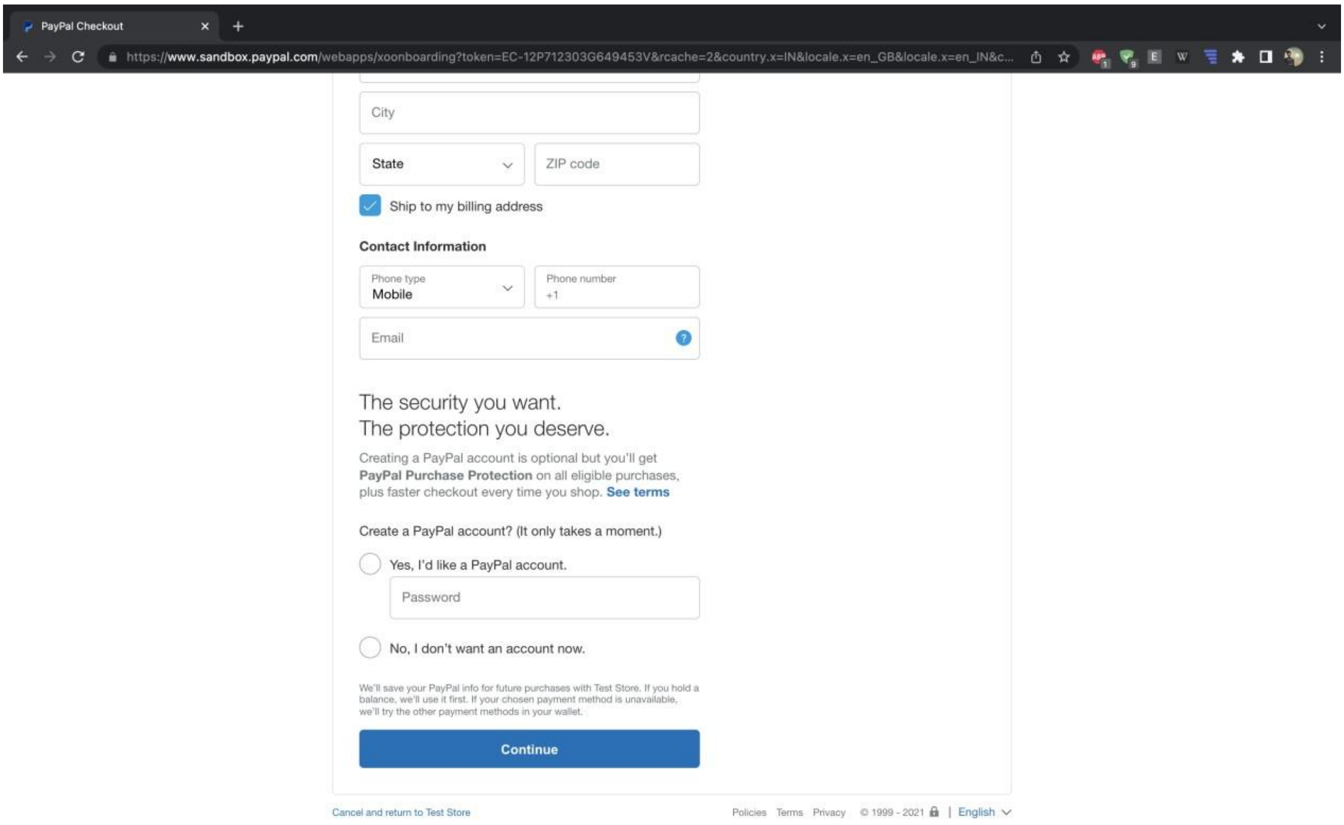
Billing address

Street address
Apt., ste., bldg.
City
State
ZIP code

☑ Ship to my billing address

PayPal is the safer, easier way to pay

No matter where you shop, we keep your financial information secure.



https://www.sandbox.paypal.com/webapps/xoonboarding?token=EC-12P712303G649453V&rcache=2&country.x=IN&locale.x=en_GB&locale.x=en_IN&c...

City

State

ZIP code

☑ Ship to my billing address

**Contact Information**

Phone type: Mobile

Phone number: +1

Email

The security you want.
The protection you deserve.

Creating a PayPal account is optional but you'll get **PayPal Purchase Protection** on all eligible purchases, plus faster checkout every time you shop. **See terms**

Create a PayPal account? (It only takes a moment.)

○ Yes, I'd like a PayPal account.

Password

○ No, I don't want an account now.

We'll save your PayPal info for future purchases with Test Store. If you hold a balance, we'll use it first. If your chosen payment method is unavailable, we'll try the other payment methods in your wallet.

Continue
Cancel and return to Test Store
Policies Terms Privacy © 1999 - 2021 | English

# REFERENCES

1. Fonts used in website: https://fonts.google.com/

2. CSS help: https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Selectors

3. General Implementation and Debugging help: https://www.stackoverflow.com

4. Syntax help (for all languages): https://www.w3schools.com

5. MySQL Syntax help: https://dev.mysql.com/doc/refman/8.0/en/

6. PHP help: Videos on https://www.youtube.com

# CODE LINK

https://bit.ly/3DhOUgz