

## **Cross-Platform Recommendation System**

Shivani Elakurthy, Lakshmi Kolluru, Srivalli Nikhitha Anantha, and Jahnavi Shakamuri

Knight Foundation School of Computing and Information Sciences, Florida International  
University

CEN5011: Advanced Software Engineering

Dr. Charlyne Walker

November 28, 2022

## Cross-Platform Recommendation System

Netflix is one of the most famous OTT platforms available in the current trend. Most people watch Netflix, Hulu, Peacock, etc. as a means to relieve stress or for fun. Once we complete watching a show, these OTT Platforms provide us recommendations for new content based on the content we finished watching. But the recommended content is only available on their own OTT Platform. Netflix doesn't recommend something from Hulu or Hulu doesn't recommend something that we finished watching on Netflix.

We're a huge fan of F.R.I.E.N.D.S. sitcom and when any of us finished the show, we always got recommended to Big Bang Theory or Young Sheldon (as they are available on HBOMax as F.R.I.E.N.D.S.) but never got recommended to Modern Family, Brooklyn Nine-Nine (which is available on Hulu). We want to build a hybrid recommendation system that considers the viewing history, likes, and ratings by the user on various OTT platforms and recommends content cross-OTT platforms. The goal is to improve the recommendations for each user, and the recommendations will include shows from multiple OTT platforms.

We want to build a website, where a user can log in and browse through the list of shows and provide ratings, likes, etc. We'll be using collaborative filtering (Navot, 2022; "Collaborative Filtering"), 2022) to find other users similar to the current user and recommend content to the current user. For example, a user 'antony' liked F.R.I.E.N.D.S. but didn't enjoy The Office. Among the users present, let's say users 'maria' and 'esha' also liked F.R.I.E.N.D.S and didn't like The Office. 'maria' also liked Brooklyn Nine-Nine, and 'esha' liked Modern Family. These two shows will be recommended to 'antony'.

Currently, these OTT platforms have no official APIs regarding content availability and user interests. We'll be using the datasets available on Kaggle (Bansal, 2021). We would also

like to use the Internet Movie Database (IMDb) API (Dadban) for the list of shows and build a recommendation system similar to Netflix's to expand the scope of recommendations.

We'll be using Python to build the recommendation system, HTML, CSS, and JavaScript for UI for the website, and PHP for server-side scripting.

### **User Story Backlog**

This section consists of all the user stories associated with this project.

**Table 1**

*User Story 1*

<b>Title</b>	Opening page of the website/application
<b>Narrative</b>	
As a user, I want the opening page quite informative so that it can tell me the purpose of this application.	
<b>Acceptance criteria</b>	
Given that as a user, when entering the website URL in any search engine then it opens the webpage.	

**Table 2**

*User Story 2*

<b>Title</b>	Registering a new account for the website
<b>Narrative</b>	
As a user, I want to be able to create an account so that I can access the features of the website.	
<b>Acceptance criteria</b>	

Given that the user navigates to the sign-up page, when the user enters the details like name, email, and password, then a verification code is sent to the user's email. Given that the user receives an email with the verification code when the user enters the verification code on the sign-up page, then he is successfully registered.

**Table 3***User Story 3*

<b>Title</b>	Logging into the user's account
<b>Narrative</b>	
As a user, I want to log in securely so that I can access the recommendations of various shows.	
<b>Acceptance criteria</b>	
Given that the user is already registered and navigates to the login page when the user enters his email and password correctly, then the user can log into his/her account.	

**Table 4***User Story 4*

<b>Title</b>	Resetting the password when the user forgot his/her password
<b>Narrative</b>	
As a user, I want to be able to reset my password so that I can set a new password when I forget my previous password.	
<b>Acceptance criteria</b>	
Given that the user is registered and navigates to the Login page when the user clicks on the Forgot Password link, then the user enters his email to receive a verification code. Given that	

the entered email is correct, and the user receives an email with a verification code, when the user enters the verification code on the website, then he is prompted to set a new password.

**Table 5***User Story 5*

<b>Title</b>	Setting preferred languages and OTT platforms for the first time
<b>Narrative</b>	
As a user who had just finished registering an account, I would like to set my preferred languages and OTT platforms, so that I can get recommendations that I would more likely watch.	
<b>Acceptance criteria</b>	
Given that the user had just finished registering an account when the user logs in the first time, then it prompts the user to set his/her preferred languages and OTT platforms.	

**Table 6***User Story 6*

<b>Title</b>	Provide ratings for a few movies/series for the first time
<b>Narrative</b>	
As a user who had just finished registering an account, I would like to provide ratings for a few movies and series so that I can get initial recommendations based on them.	
<b>Acceptance criteria</b>	

Given that the user had just finished registering an account, when the user logs in the first time, then it prompts the user to initially rate a few movies/series so that personalized recommendations can be provided to the user.

**Table 7***User Story 7*

<b>Title</b>	Recommendations based on the ratings provided by the user
<b>Narrative</b>	
As a user, I like to see recommendations for the content I haven't watched yet based on the ratings I've provided so that I don't have to research for movies/series that I might like.	
<b>Acceptance criteria</b>	
Given that the user is logged in, when the user directs to the home page, then the movies/series recommendations to the user are displayed under the 'You might like' section.	

**Table 8***User Story 8*

<b>Title</b>	To view platform-wise recommendations
<b>Narrative</b>	
As a user, I would like to see OTT platform-wise recommendations, so that I can see platform-specific movies/series.	
<b>Acceptance criteria</b>	
Given that the user is logged in, when the user navigates to the homepage, then the user can see each user preferred platform with recommendations from that platform.	

**Table 9***User Story 9*

<b>Title</b>	To view basic information for any movies/series
<b>Narrative</b>	
As a user, I would like to view basic information for any movies/series such as genres, actors, release year, description, etc. so that I know more information about them.	
<b>Acceptance criteria</b>	
Given that the user is logged in, when the user navigates to a specific movie/series page, then he/she sees the basic information for it such as genres, directors, actors, release year, content ratings, description, and which OTT platform it is available on.	

**Table 10***User Story 10*

<b>Title</b>	The search option to look for specific show
<b>Narrative</b>	
As a user, I would like to have a search field to type any movie/series' name so that I could find relevant movies/series.	
<b>Acceptance criteria</b>	
Given that the user is logged in, then the user can look for the search field on the navigation pane on any page of the website.	

**Table 11**

*User Story 11*

<b>Title</b>	Set a status for any movie/series
<b>Narrative</b>	
As a user, I would like to set a status option for any movie/series, so that I can keep track.	
<b>Acceptance criteria</b>	
Given that the user is logged in, when the user navigates to a movie/series page, then he/she can set the status option to any of the following values: ‘Currently watching’, ‘Finished’, ‘Not yet started’, ‘Dropped’, ‘Wishlist’.	

**Table 12***User Story 12*

<b>Title</b>	Provide rating for any movie/series
<b>Narrative</b>	
As a user, I want to be able to provide ratings for any movies/series so that I can share my opinion to help other users.	
<b>Acceptance criteria</b>	
Given that the user is logged in, when the user navigates to a specific movie/series page, then he/she can click on the Edit button under the ‘Your Rating’ label and provide a rating.	

**Table 13***User Story 13*

<b>Title</b>	To view similar content for any movies/series
<b>Narrative</b>	

As a user, I would like to see similar content for any movies/series that I might like so that I can see recommendations for a specific movie/series.
---

<b>Acceptance criteria</b>
----------------------------

Given that the user is logged in, when the user navigates to a specific movie/series page, then he/she can see some movies/series that are similar to the current movie/series under the ‘Similar content’ section.
---

**Table 14***User Story 14*

<b>Title</b>	To view the list of movies/series that the user is currently watching
<b>Narrative</b>	
As a user, I would like to see a list of movies/series that I'm currently watching so that I can keep track of what I'm watching.	
<b>Acceptance criteria</b>	
Given that the user is logged in and had set the status option for a couple of movies/series to ‘Currently watching’, when the user navigates to the home page, then he/she can see the list of movies/series that he/she is currently watching.	

**Table 15***User Story 15*

<b>Title</b>	To view a Wishlist that consists of movies/series that a user wants to watch
<b>Narrative</b>	

As a user, I would like to see my Wishlist so that I can jump right to it whenever I want to watch something new.
---

<b>Acceptance criteria</b>
----------------------------

Given that the user is logged in, when the user navigates to a few movies/series pages, if the user updates the status option for those movies/series to ‘Wishlist’, then the user can see his Wishlist on the homepage.
--

**Table 16***User Story 16*

<b>Title</b>	Editing personal details
<b>Narrative</b>	
As a user, I would like to edit my personal details, languages preferred, and OTT preferences in my profile so that I get up-to-date recommendations.	
<b>Acceptance criteria</b>	
Given that the user is logged in when the user navigates to the Profile page, then he/she can edit his/her basic information like name, email, password and preferred list of languages and OTT platforms.	

**Table 17***User Story 17*

<b>Title</b>	Change ratings for any movies/series
<b>Narrative</b>	

As a user, I would like to change my ratings for any content at any point in time so that I can update the ratings according to how I feel as time passes.
--

<b>Acceptance criteria</b>
----------------------------

Given that the user navigates to a specific movie/series page, when the status option is not 'Not yet started', then the user can provide or update his rating.
---

**Table 18***User Story 18*

<b>Title</b>	Logging out of the account
<b>Narrative</b>	
As a user, I would like to log out whenever I can, especially when I log in using a public computer.	
<b>Acceptance criteria</b>	
Given that the user is logged in and navigates to the Profile page when the user clicks on the Log out button, then he/she is logged out of his/her account.	

**Table 19***User Story 19*

<b>Title</b>	Preprocessing the datasets
<b>Narrative</b>	
As an admin, I would like to preprocess the datasets, so that the data can be ready for use in machine learning.	
<b>Acceptance criteria</b>	

Given that the datasets are downloadable and editable, the admin can preprocess the datasets such as data cleaning, transformation, merging, etc. before it is available for machine learning.

**Table 20***User Story 20*

<b>Title</b>	Deleting user account
<b>Narrative</b>	
As a user, I want to delete my account, so that my personal details will be deleted when I no longer want to use this service.	
<b>Acceptance criteria</b>	
Given that the user is logged in and navigates to the Profile page when the user clicks on the 'Delete my account' button and enters his password correctly, then the user's account is deleted. The user's personal details will be deleted, but the ratings provided by the user won't be deleted as they'll be used to provide recommendations to other users.	

**Table 21***User Story 21*

<b>Title</b>	Logging into the admin's account
<b>Narrative</b>	
As an admin, I want to log in securely, so that I can monitor user activity.	
<b>Acceptance criteria</b>	
Given that the admin is already registered and navigates to the login page when the admin enters his email and password correctly, then the admin can log into his/her account.	

**Table 22***User Story 22*

<b>Title</b>	Viewing all ratings
<b>Narrative</b>	
As an admin, I want to be able to view all the ratings provided by different users in the order of the time at which the rating was provided with the most recent rating at the top so that I can monitor and detect any possible spam rating by the user.	
<b>Acceptance criteria</b>	
Given that the admin is already logged in when the admin navigates to the ‘View all Ratings’ page, then the admin can view all the ratings (user’s email, content’s name, rating provided) with the recent rating at the top.	

**Table 23***User Story 23*

<b>Title</b>	Remove a rating
<b>Narrative</b>	
As an admin, I want to be able to remove a rating provided by the user so that I can remove any possible spam rating by the user.	
<b>Acceptance criteria</b>	
Given that the admin is already logged in when the admin navigates to the ‘View all Ratings’ page, then the admin can click on a rating (user’s email, content’s name, rating provided) and click on the ‘Remove Rating’ button.	

**Table 24***User Story 24*

<b>Title</b>	Viewing all users
<b>Narrative</b>	
As an admin, I want to be able to view all the users with the user with the most recent activity at the top so that I can monitor and detect any possible spam user accounts.	
<b>Acceptance criteria</b>	
Given that the admin is already logged in when the admin navigates to the ‘View all Users’ page, then the admin can view all the users.	

**Table 25***User Story 25*

<b>Title</b>	Remove a user
<b>Narrative</b>	
As an admin, I want to be able to remove a user’s account so that I can delete any possible spam user accounts.	
<b>Acceptance criteria</b>	
Given that the admin is already logged in when the admin navigates to the ‘View all Users’ page, then the admin can click on a user and click on the ‘Remove User’ button. The user will be notified via email that his account has been deleted.	

## Sprints

We'll be working on this project for four sprints with each sprint covering over two weeks. The user stories are allocated to the sprints based on factors like the order in which they should be implemented, urgency, and dependencies. These stories are assigned to the users based on their roles and capabilities.

**Table 26**

*Sprint 1 – 26<sup>th</sup> September 2022 to 9<sup>th</sup> October 2022*

User story number	Points	Name of the team person	Board status
1	1	Lakshmi Kolluru	Completed
2	3	Shivani Elakurthy	Completed
3	1	Shivani Elakurthy	Completed
4	3	Srivalli Nikhitha Anantha	Completed
5	3	Jahnavi Shakamuri	Completed
18	1	Jahnavi Shakamuri	Completed
19	3	Lakshmi Kolluru	Completed
21	1	Srivalli Nikhitha Anantha	Completed

**Table 27**

*Sprint 2 – 10<sup>th</sup> October 2022 to 23<sup>rd</sup> October 2022*

User story number	Points	Name of the team person	Board status
6	3	Srivalli Nikhitha Anantha	Completed
7	5	Shivani Elakurthy	Completed
9	3	Jahnavi Shakamuri	Completed

User story number	Points	Name of the team person	Board status
12	1	Srivalli Nikhitha Anantha	Completed
13	5	Lakshmi Kolluru	Completed

**Table 28***Sprint 3 – 24<sup>th</sup> October 2022 to 6<sup>th</sup> November 2022*

User story number	Points	Name of the team person	Board status
8	3	Lakshmi Kolluru	Completed
10	3	Shivani Elakurthy	Completed
16	3	Jahnavi Shakamuri	Completed
22	3	Srivalli Nikhitha Anantha	Completed

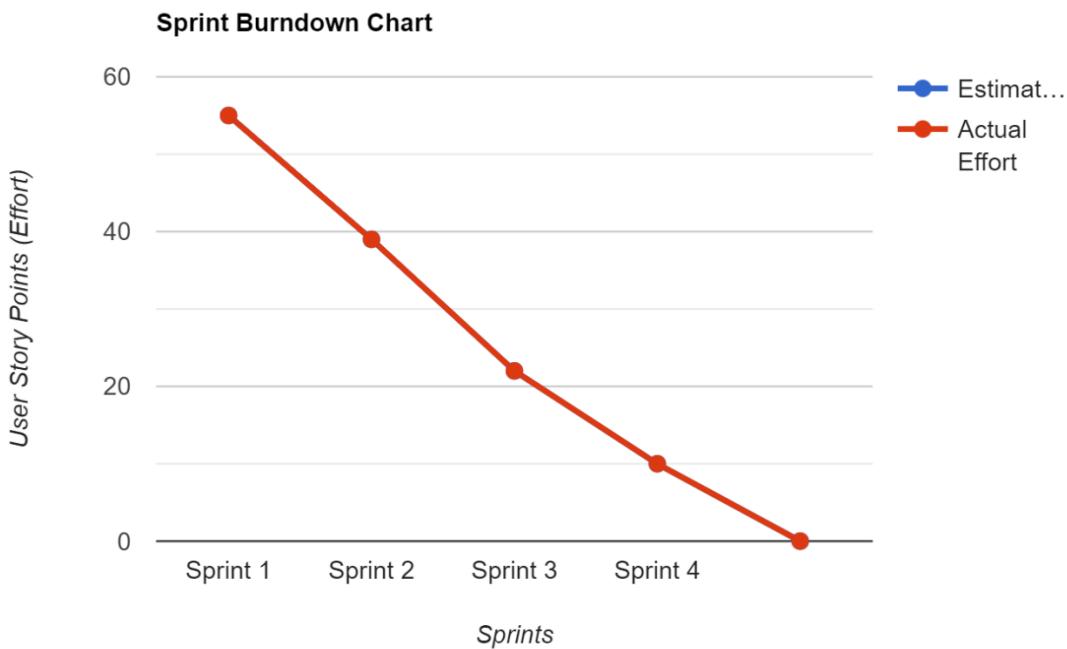
**Table 29***Sprint 4 – 7<sup>th</sup> November 2022 to 20<sup>th</sup> November 2022*

User story number	Points	Name of the team person	Board status
11	1	Srivalli Nikhitha Anantha	Completed
14	1	Shivani Elakurthy	Completed
15	1	Srivalli Nikhitha Anantha	Completed
17	1	Jahnavi Shakamuri	Completed
20	1	Shivani Elakurthy	Completed
23	1	Jahnavi Shakamuri	Completed
24	3	Lakshmi Kolluru	Completed
25	1	Shivani Elakurthy	Completed

## Sprint burndown chart

**Figure 1**

*Burndown chart for the planned sprints for the entire semester*



## Roles and Responsibilities

We are a team of four members with a mix of front-end and back-end developers. We have proposed two roles for this project: Front-end developers and Back-end developers.

### Proposed responsibilities

The proposed responsibilities for each role are as follows:

#### ***Front-end developers***

Front-end developers in our team will be designing the UI using HTML, CSS, and Javascript.

#### ***Back-end developers***

Back-end developers in our team will be working on preprocessing the datasets, building the content and collaborative filtering algorithms, connecting the front end with the data, and so on. We will be using Python for machine learning and PHP for server-side scripting. The data will be stored in .csv format.

### **Proposed roles**

**Table 30**

Name	Roles
Shivani Elakurthy	Back-end developer, Front-end developer
Lakshmi Kolluru	Back-end developer, Front-end developer
Srivalli Nikhitha Anantha	Front-end developer
Jahnavi Shakamuri	Front-end developer

### **Design**

Design can be defined as the software engineering life cycle activity in which software requirements are analyzed in order to produce a description of the software's internal structure that will serve as the basis for its construction (Bourque, P., Dupuis, R., 2004).

#### ***The design process invoked by the team***

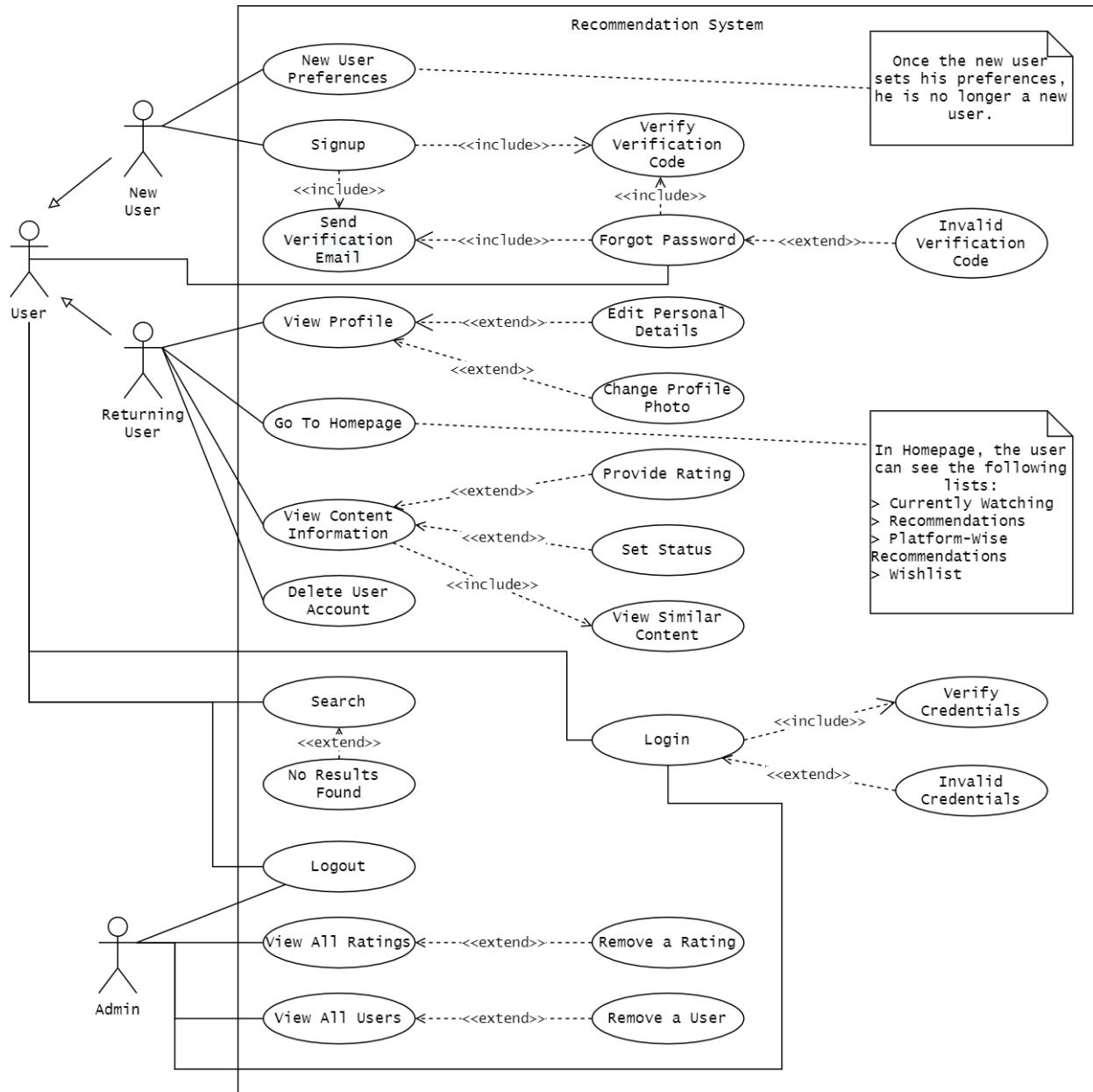
We started by looking for a few datasets that could be used in this project, datasets that contain movies/series with basic information such as genre, actors, which OTT platform they are available on, and datasets with users, the ratings they provided to different movies/series. Then, we started designing simple UI designs for all the scenarios using pen and paper to put forward all the components that are required in this project. With the help of these designs, we started designing the UML diagrams and started working on the project by working on the assigned user

stories. Next, we plan to begin the verification process by building test cases to test the working of our project.

## Use case diagram

**Figure 2**

*Use case model for Cross-Platform Recommendation System*



### ***Use cases descriptions***

**Table 31**

<i>Use case name</i>	Login
<i>Brief description</i>	This use case describes how the user or admin is logged in to the website.
<i>Participating actors</i>	Initiated by User or Admin
<i>Flow of events</i>	<ol style="list-style-type: none"> <li>1. The user/admin enters his email and password and clicks on the Login button on the Login page of the website.</li> <li>2. This use case <b>includes</b> the ‘Verify Credentials’ use case.</li> <li>3. The user/admin is now logged in to the website and is redirected to their respective Homepage of the website.</li> </ol>
<i>Entry condition</i>	The user/admin is navigated to the Login page of the website.
<i>Exit condition</i>	The user/admin is logged in and is redirected to the respective Homepage of the website.
<i>Exceptions</i>	<ul style="list-style-type: none"> <li>• The user/admin enters an invalid username or password.</li> <li>• The system could not connect to the database, hence could not validate the credentials.</li> </ul>

**Table 32**

<i>Use case name</i>	Verify Credentials
<i>Brief description</i>	This use case describes how the system verifies the user/admin credentials.
<i>Participating actors</i>	Initiated by System

<i>Use case name</i>	Verify Credentials
<i>Flow of events</i>	<ol style="list-style-type: none"> <li>1. The system verifies the email and password entered by the user or admin with the ones present in the database.</li> <li>2. The user/admin proceeds forward if the credentials are valid.</li> </ol>
<i>Entry condition</i>	The user/admin has entered his email and password on their respective Login page of the website.
<i>Exit condition</i>	The user/admin proceeds forward.
<i>Exceptions</i>	<ul style="list-style-type: none"> <li>• The user/admin enters an invalid, unregistered or non-existing email.</li> <li>• The user/admin enters the wrong password.</li> <li>• If the system could not connect to the database, hence could not validate the user credentials.</li> </ul>

**Table 33**

<i>Use case name</i>	Invalid Credentials
<i>Brief description</i>	This use case describes when the user or admin enters invalid credentials.
<i>Participating actors</i>	Initiated by System
<i>Flow of events</i>	<ol style="list-style-type: none"> <li>1. The system notifies the user or admin that the entered credentials do not match the credentials present in the database.</li> </ol>
<i>Entry condition</i>	This use case <b>extends</b> the ‘Login’ use case.

<i>Use case name</i>	Invalid Credentials
<i>Exit condition</i>	The user/admin will be notified regarding the invalid credentials.

**Table 34**

<i>Use case name</i>	Sign Up
<i>Brief description</i>	This use case describes how the user creates an account on the website.
<i>Participating actors</i>	Initiated by User
<i>Flow of events</i>	<ol style="list-style-type: none"> <li>1. The user enters his/her name, email, and password and clicks on the Next button on the Signup page of the website.</li> <li>2. This use case <b>includes</b> the ‘Send Verification Email’ use case and the ‘Verify Verification Code’ use case.</li> <li>3. The system saves the credentials in the database and creates an account for the user.</li> </ol>
<i>Entry condition</i>	The user is navigated to the Signup page of the website.
<i>Exit condition</i>	<ul style="list-style-type: none"> <li>• An account is created for the user.</li> <li>• The user is directed to the Set User Preferences page.</li> </ul>
<i>Exceptions</i>	<ul style="list-style-type: none"> <li>• The user enters an invalid or an already existing email and weak password (A password should contain a minimum of 6 characters including numbers, letters, and special characters).</li> <li>• If the system could not connect to the database, hence could not save the user credentials.</li> </ul>

**Table 35**

<i>Use case name</i>	Forgot Password
<i>Brief description</i>	This use case describes how to reset the forgotten password.
<i>Participating actors</i>	Initiated by User
<i>Flow of events</i>	<ol style="list-style-type: none"> <li>1. The user needs to enter his registered email and click on the Send Verification Code button on the Forgot Password page.</li> <li>2. This use case <b>includes</b> the ‘Send Verification Email’ use case and the ‘Verify Verification Code’ use case.</li> <li>3. The user enters a new password and clicks on the Reset button.</li> <li>4. The system updates the user’s password in the database.</li> <li>5. The user is directed to the Login page of the website.</li> </ol>
<i>Entry condition</i>	The user is navigated to the Forgot Password page of the website.
<i>Exit condition</i>	The user resets the password and is redirected to the Login page of the website.
<i>Exceptions</i>	<ul style="list-style-type: none"> <li>• The user enters an invalid, unregistered, or non-existing email.</li> <li>• If the system could not connect to the database, hence could not update the user’s password.</li> </ul>

**Table 36**

<i>Use case name</i>	Send Verification Email
<i>Brief description</i>	This use case describes how the system sends a verification email to the user’s email.
<i>Participating actors</i>	Initiated by System

<i>Use case name</i>	Send Verification Email
<i>Flow of events</i>	<ol style="list-style-type: none"> <li>1. The system takes the user's email and sends an email with a one-time verification code.</li> </ol>
<i>Entry condition</i>	The user has entered his email on the Signup page or Forgot Password page of the website.
<i>Exit condition</i>	The system sends an email to the user.
<i>Exceptions</i>	<ul style="list-style-type: none"> <li>• The system could not send an email because the server is down.</li> <li>• The user entered an email that does not exist.</li> </ul>

**Table 37**

<i>Use case name</i>	Verify Verification Code
<i>Brief description</i>	This use case describes how the system verifies the verification code during Signup or Forgot Password.
<i>Participating actors</i>	Initiated by System
<i>Flow of events</i>	<ol style="list-style-type: none"> <li>1. The user receives the verification code that is sent through an email by the system.</li> <li>2. The system verifies the verification code.</li> <li>3. The user proceeds forward if the code is valid.</li> </ol>
<i>Entry condition</i>	The user has received the email with the verification code.
<i>Exit condition</i>	The user enters the verification code and proceeds forward.
<i>Exceptions</i>	<ul style="list-style-type: none"> <li>• The user enters the wrong or expired verification code.</li> </ul>

<i>Use case name</i>	Verify Verification Code
<i>Exceptions</i>	<ul style="list-style-type: none"> <li>If the system could not connect to the database, hence could not validate the user credentials.</li> </ul>

**Table 38**

<i>Use case name</i>	Invalid Verification Code
<i>Brief description</i>	This use case describes when the user enters an invalid verification code.
<i>Participating actors</i>	Initiated by System
<i>Flow of events</i>	<ol style="list-style-type: none"> <li>The system notifies the user that the entered verification code does not match with the one sent to the user's email.</li> </ol>
<i>Entry condition</i>	This use case <b>extends</b> the 'Sign Up' and 'Forgot Password' use cases.
<i>Exit condition</i>	The user will be notified regarding the invalid verification code.

**Table 39**

<i>Use case name</i>	New User Preferences
<i>Brief description</i>	This use case describes how the new user can set language and OTT platform preferences and rate a few shows (used to provide better recommendations).
<i>Participating actors</i>	Initiated by User

<i>Use case name</i>	New User Preferences
<i>Flow of events</i>	<p>1. The user will select the preferred languages and OTT platforms, provide ratings for a few movies/series that he/she already watched, and clicks on the Get Started button on the New User Preferences page.</p> <p>2. The system stores the above preferences in the database (recommendations are provided based on these preferences).</p> <p>3. The user is directed to the Homepage page of the website.</p>
<i>Entry condition</i>	The user is a new user who has logged in to the website for the first time.
<i>Exit condition</i>	<ul style="list-style-type: none"> <li>• The user sets his/her preferences and is navigated to the Homepage.</li> <li>• The user gets recommendations according to their preferences.</li> </ul>
<i>Exceptions</i>	<ul style="list-style-type: none"> <li>• The user doesn't select any of the preferences, he/she cannot proceed forward.</li> <li>• If the system could not connect to the database, hence could not save the preferences to the database.</li> </ul>

**Table 40**

<i>Use case name</i>	Go To Homepage
<i>Brief description</i>	This use case describes what the user can expect on the homepage of the website.
<i>Participating actors</i>	Initiated by User

<i>Use case name</i>	Go To Homepage
<i>Flow of events</i>	<ol style="list-style-type: none"> <li>1. The user clicks on the website logo on the navigation pane on any page of the website.</li> <li>2. The system pulls the user's language preferences, OTT platform preferences, ratings provided, and status set by the user for various shows. These preferences are used to provide personalized recommendations using collaborative filtering.</li> <li>3. The user can see the following lists on the Homepage:             <ol style="list-style-type: none"> <li>a. The shows he has set the status to 'Currently Watching'</li> <li>b. The personalized recommendations generated by the System.</li> <li>c. Platform-wise recommendations generated by the System.</li> <li>d. The shows he has set the status to 'Wishlist'.</li> </ol> </li> </ol>
<i>Entry condition</i>	The user is logged into the website.
<i>Exit condition</i>	<ul style="list-style-type: none"> <li>• The user is directed to the Homepage.</li> <li>• The user can choose any show from the recommendations, the Wishlist, or the Currently Watching list.</li> </ul>
<i>Exceptions</i>	<ul style="list-style-type: none"> <li>• Wishlist list is not available unless the user sets the status for any show to 'Wishlist'.</li> <li>• Currently Watching list is not available unless the user sets the status for any show to 'Currently Watching'.</li> </ul>

**Table 41**

<i>Use case name</i>	View Profile
<i>Brief description</i>	This use case describes how a user can view his/her profile.
<i>Participating actors</i>	Initiated by User
<i>Flow of events</i>	<p>1. The user clicks on the Profile button on the top navigation bar on any page of the website.</p> <p>2. The system communicates with the database and pulls the user's personal details record.</p> <p>3. The user can view information like name, email, password, language preferences, profile photo, etc. on the Profile page of the website.</p>
<i>Entry condition</i>	The user is logged in to the website.
<i>Exit condition</i>	The user can now view his/her profile on the Profile page of the website.
<i>Exceptions</i>	If the system could not connect to the database, hence could not pull the user's personal information from the database.

**Table 42**

<i>Use case name</i>	Edit Personal Details
<i>Brief description</i>	This use case describes how a user can edit his/her personal details or Profile.
<i>Participating actors</i>	Initiated by User

<i>Use case name</i>	Edit Personal Details
<i>Flow of events</i>	<ol style="list-style-type: none"> <li>1. The user can edit his Name, Email, Password, Language Preferences, etc. on the Profile page of the website.</li> <li>2. The user then clicks on the Save button.</li> <li>3. The system validates the newly entered information and updates the database.</li> <li>4. The user is notified that his personal details are updated.</li> </ol>
<i>Entry condition</i>	This use case <b>extends</b> the ‘View Profile’ use case.
<i>Exit condition</i>	<ul style="list-style-type: none"> <li>• The user will see that his personal details are updated.</li> <li>• The system notifies the user that his details are updated.</li> </ul>
<i>Exceptions</i>	If the system could not connect to the database, hence could not update the user’s personal information in the database.

**Table 43**

<i>Use case name</i>	Change Profile Photo
<i>Brief description</i>	This use case describes how a user can change his profile photo.
<i>Participating actors</i>	Initiated by User
<i>Flow of events</i>	<ol style="list-style-type: none"> <li>1. The user clicks on the Change Photo button under the Profile photo on the Profile page.</li> <li>2. The system prompts the user to select an image from his PC.</li> <li>3. The user selects an image.</li> <li>4. The system updated the profile photo for the user in the database and notifies the user that his profile photo is updated.</li> </ol>

<i>Use case name</i>	Change Profile Photo
<i>Entry condition</i>	This use case <b>extends</b> the ‘View Profile’ use case.
<i>Exit condition</i>	<ul style="list-style-type: none"> <li>• The user will see that the profile photo is updated.</li> <li>• The system notifies the user that his profile photo is updated.</li> </ul>
<i>Exceptions</i>	The system could not connect to the database, hence could not update the user’s profile photo in the database.

**Table 44**

<i>Use case name</i>	Search
<i>Brief description</i>	This use case describes how the user searches for the content.
<i>Participating actors</i>	Initiated by User
<i>Flow of events</i>	<ol style="list-style-type: none"> <li>1. The user clicks on the Search button on the top navigation bar on any page of the website and enters the content/show name in the search bar.</li> <li>2. The system communicates with the database and pulls the content’s information.</li> <li>3. The user is redirected to the content’s page.</li> </ol>
<i>Entry condition</i>	The user is logged in to the website.
<i>Exit condition</i>	The user navigates to the searched content’s page.
<i>Exceptions</i>	If the system could not connect to the database, hence could not pull the content’s information from the database.

**Table 45**

<i>Use case name</i>	No Results Found
<i>Brief description</i>	This use case describes which no results are found when the user searches for a show.
<i>Participating actors</i>	Initiated by System
<i>Flow of events</i>	1. The system could not find any content matching the user's input and will display a 404 image.
<i>Entry condition</i>	This use case <b>extends</b> the 'Search' use case.
<i>Exit condition</i>	<ul style="list-style-type: none"> <li>• The user will see the 404 (File not found) image.</li> </ul>

**Table 46**

<i>Use case name</i>	View Content Information
<i>Brief description</i>	This use case describes how the user can find the basic information for any content or show.
<i>Participating actors</i>	Initiated by User
<i>Flow of events</i>	<ol style="list-style-type: none"> <li>1. The user navigates to any show's page.</li> <li>2. The system communicates with the database and pulls the content's information.</li> <li>3. The user can now see the content's basic information like genres, actors, directors, description, average rating, etc.</li> <li>4. This use case <b>includes</b> the 'View Similar Content' use case.</li> </ol>
<i>Entry condition</i>	The user is logged in to the website.
<i>Exit condition</i>	The user can see the show's information.

<i>Use case name</i>	View Content Information
<i>Exceptions</i>	If the system could not connect to the database, hence could not pull the content's information from the database.

**Table 47**

<i>Use case name</i>	Provide Rating
<i>Brief description</i>	This use case describes how a user provides a rating for any show or content.
<i>Participating actors</i>	Initiated by User
<i>Flow of events</i>	<ol style="list-style-type: none"> <li>1. The user clicks the Edit button under the Your Rating label on the movie/series or show page.</li> <li>2. The system prompts the user to enter a rating (0 to 10).</li> <li>3. The user enters a rating and clicks on the Save button.</li> <li>4. The system updates the rating provided by the user for the show in the database and notifies the user that his rating is added.</li> </ol>
<i>Entry condition</i>	This use case <b>extends</b> the ‘View Content Information’ use case.
<i>Exit condition</i>	<ul style="list-style-type: none"> <li>• The system notifies the user that his rating is added.</li> <li>• The user will see the rating he provided beside the Your Rating label.</li> </ul>
<i>Exceptions</i>	The system could not connect to the database, hence could not add the user's rating to the database.

**Table 48**

<i>Use case name</i>	Set Status
<i>Brief description</i>	This use case describes how a user can set a status for any show or content.
<i>Participating actors</i>	Initiated by User
<i>Flow of events</i>	<ol style="list-style-type: none"> <li>1. The user selects an option from the Status dropdown menu on the movie/series or show page.</li> <li>2. The system updates the status selected by the user for the show in the database and notifies the user that his status has been set.</li> </ol>
<i>Entry condition</i>	This use case <b>extends</b> the ‘View Content Information’ use case.
<i>Exit condition</i>	The system notifies the user that his status has been updated.
<i>Exceptions</i>	The system could not connect to the database, and hence could not update the user’s status for this content in the database.

**Table 49**

<i>Use case name</i>	View Similar Content
<i>Brief description</i>	This use case describes how a user can see similar content for any show.
<i>Participating actors</i>	Initiated by User
<i>Flow of events</i>	<ol style="list-style-type: none"> <li>1. The user navigates to any show’s page on the website.</li> <li>2. The system applies content-based filtering to generate a list of shows that are similar to the current show and display it under the Similar Content label.</li> </ol>

<i>Use case name</i>	View Similar Content
<i>Entry condition</i>	This use case <b>extends</b> the ‘View Content Information’ use case.
<i>Exit condition</i>	The user can see the list of shows that are similar to the current show under the Similar content label.
<i>Exceptions</i>	The system could not connect to the database, hence could not get the list of shows from the database.

**Table 50**

<i>Use case name</i>	Logout
<i>Brief description</i>	This use case describes how a user or admin can log out from the website.
<i>Participating actors</i>	Initiated by User or Admin
<i>Flow of events</i>	<ol style="list-style-type: none"> <li>1. The user/admin clicks on the Logout button on the navigation bar on any page of the website.</li> <li>2. The system communicates with the database, and updates that the user/admin has logged out of the website.</li> <li>3. The user/admin is logged out and forwarded to the opening page of the website.</li> </ol>
<i>Entry condition</i>	The user/admin is logged in to the website.
<i>Exit condition</i>	The user/admin is logged out of the website.
<i>Exceptions</i>	The system could not connect to the database, hence could not log out the user/admin from the website.

**Table 51**

<i>Use case name</i>	Delete User Account
<i>Brief description</i>	This use case describes how a user can delete his account.
<i>Participating actors</i>	Initiated by User
<i>Flow of events</i>	<ol style="list-style-type: none"> <li>1. The user clicks on the Profile button on the top navigation bar on any page of the website.</li> <li>2. The system communicates with the database, pulls the user's personal details record, and displays it to the user on the Profile page.</li> <li>3. The user clicks on the 'Delete my account' button.</li> <li>4. The system prompts the user to enter his password to verify if he/she wants to delete his/her account.</li> <li>5. The user enters his/her password.</li> <li>6. The system validates the password with the one stored in the database.</li> <li>7. The user account is deleted and navigated to the opening page of the website.</li> </ol>
<i>Entry condition</i>	The user is logged in to the website.
<i>Exit condition</i>	<p>The user's account is deleted.</p> <p>The user is navigated to the opening page of the website.</p>
<i>Exceptions</i>	<ul style="list-style-type: none"> <li>• The user enters an incorrect password.</li> <li>• The system could not connect to the database, hence could not delete the user's profile from the database.</li> </ul>

**Table 52**

<i>Use case name</i>	View All Ratings
<i>Brief description</i>	This use case describes how an admin can view user ratings of all the shows.
<i>Participating actors</i>	Initiated by Admin
<i>Flow of events</i>	<ol style="list-style-type: none"> <li>1. The admin clicks on the View All Ratings button on the navigation pane.</li> <li>2. The system retrieves all the user ratings for all the shows and redirects the admin to the View All Rating page. The page has all the user ratings in a table having the most recent rating at the top of the page.</li> </ol>
<i>Entry condition</i>	The admin is logged in to the website.
<i>Exit condition</i>	The admin can see all the user ratings on the View All Ratings page.
<i>Exceptions</i>	The system could not connect to the database and hence could not retrieve the ratings from the database.

**Table 53**

<i>Use case name</i>	Remove a Rating
<i>Brief description</i>	This use case describes how an admin can remove a rating provided by a specific user for a specific show.
<i>Participating actors</i>	Initiated by Admin

<i>Use case name</i>	Remove a Rating
<i>Flow of events</i>	<ol style="list-style-type: none"> <li>1. The admin clicks on the Remove Rating button for a rating provided by a user for a show.</li> <li>2. The system deletes this rating from the database and notifies the admin that this rating has been deleted.</li> </ol>
<i>Entry condition</i>	This use case <b>extends</b> the ‘View All Ratings’ use case.
<i>Exit condition</i>	The admin can see the rating is removed from the list of all ratings.
<i>Exceptions</i>	The system could not connect to the database and hence could not delete the rating from the database.

**Table 54**

<i>Use case name</i>	View All Users
<i>Brief description</i>	This use case describes how an admin can view all users.
<i>Participating actors</i>	Initiated by Admin
<i>Flow of events</i>	<ol style="list-style-type: none"> <li>1. The admin clicks on the View All Users button on the navigation pane.</li> <li>2. The system retrieves all the users and redirects the admin to the View All Users page. The page has all the users in a table having the user with the most recent activity at the top of the page.</li> </ol>
<i>Entry condition</i>	The admin is logged in to the website.
<i>Exit condition</i>	The admin can see all the users on the View All Users page.

<i>Use case name</i>	View All Users
<i>Exceptions</i>	The system could not connect to the database and hence could not retrieve the ratings from the database.

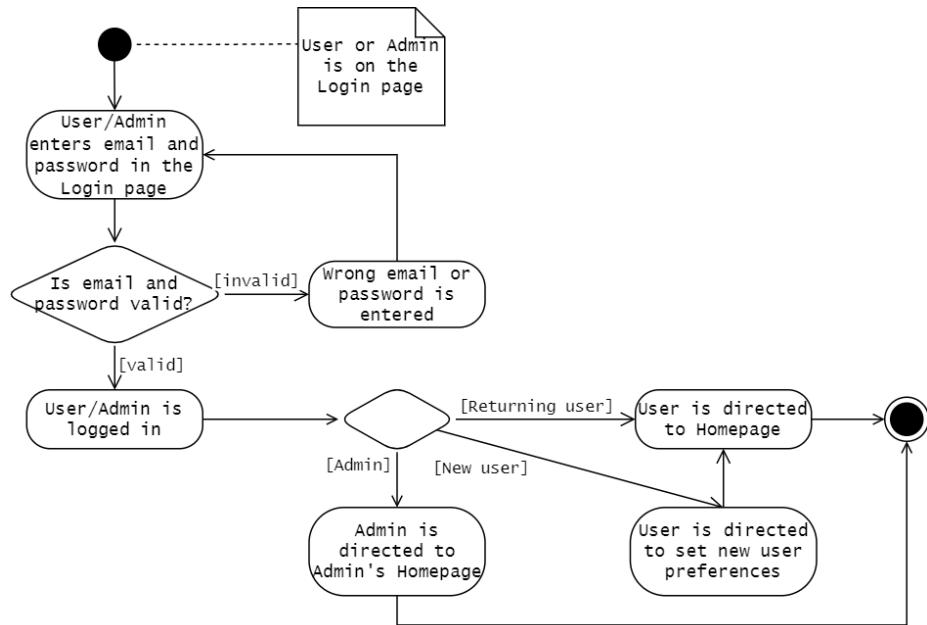
**Table 55**

<i>Use case name</i>	Remove a User
<i>Brief description</i>	This use case describes how an admin can remove a user (i.e. delete the user's account).
<i>Participating actors</i>	Initiated by Admin  Communicated with the User
<i>Flow of events</i>	<ol style="list-style-type: none"> <li>1. The admin clicks on the Remove User button on the row for the user in the table.</li> <li>2. The system deletes this user's account from the database, notifies the admin that this user's account has been deleted, and sends an email to the user notifying the deletion of his account.</li> </ol>
<i>Entry condition</i>	This use case <b>extends</b> the 'View All Users' use case.
<i>Exit condition</i>	<ul style="list-style-type: none"> <li>• The admin is notified that the user account is deleted.</li> <li>• The user is notified through email regarding the deletion of his account.</li> </ul>
<i>Exceptions</i>	The system could not connect to the database and hence could not delete the user's account.

## Activity diagrams

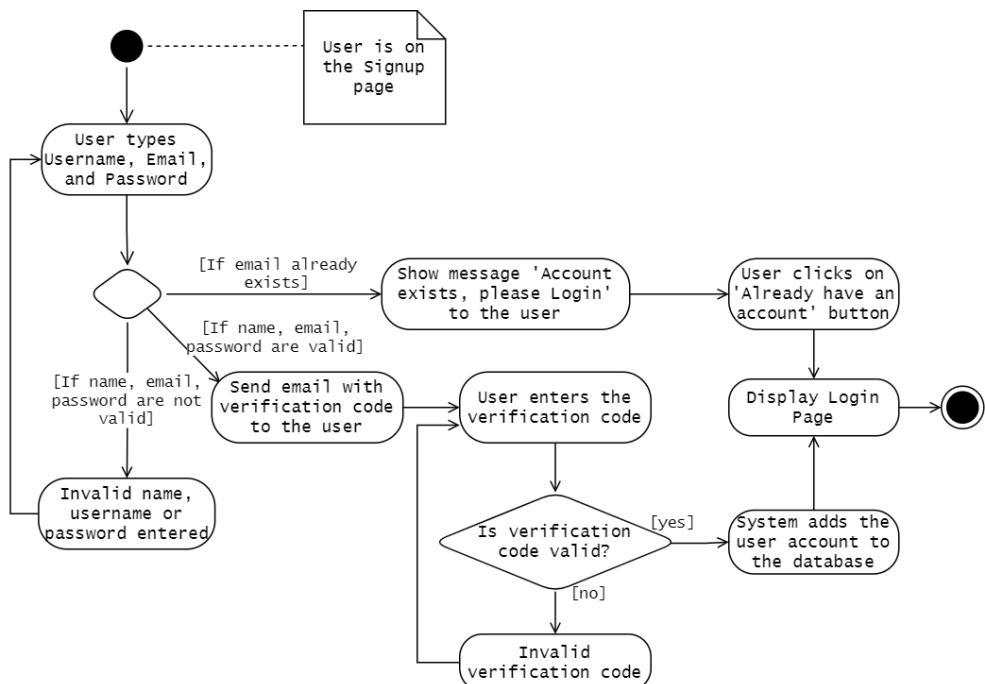
**Figure 3**

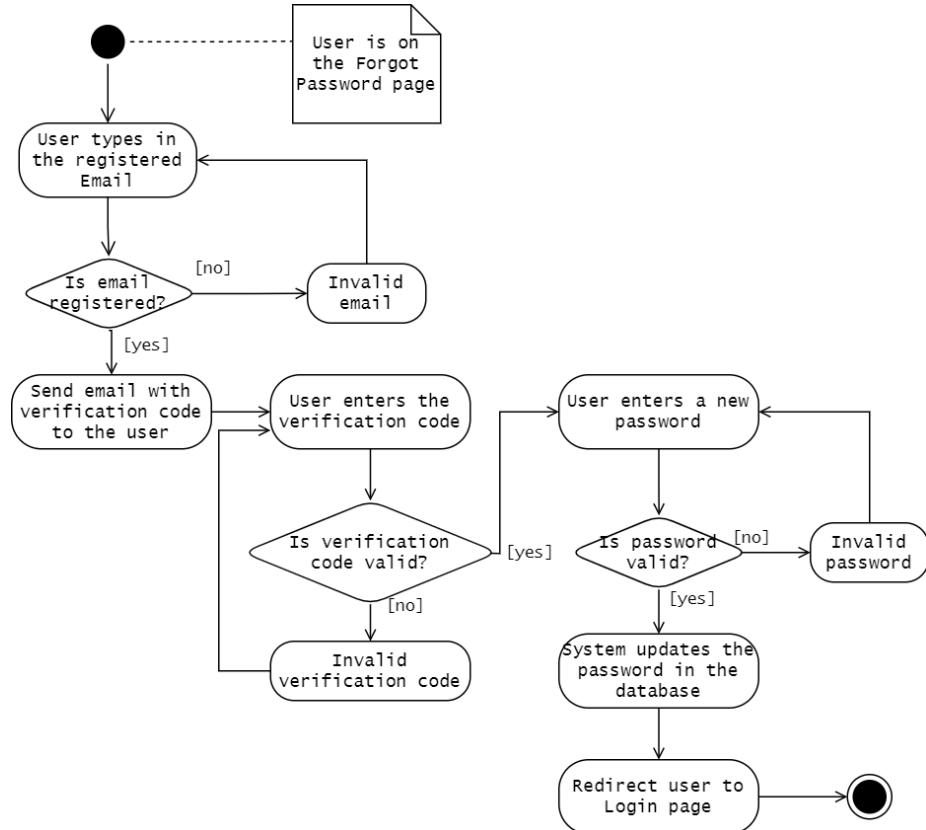
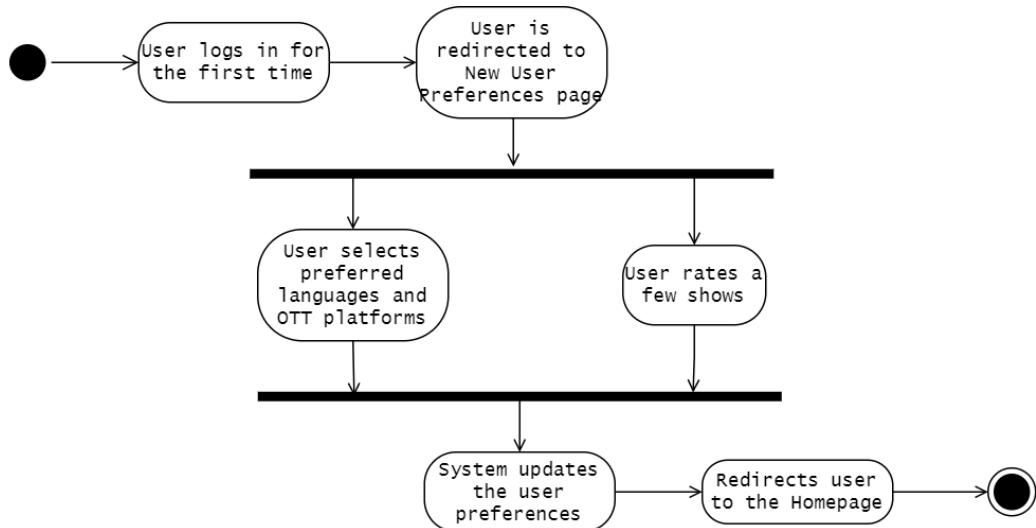
*Login Activity diagram*

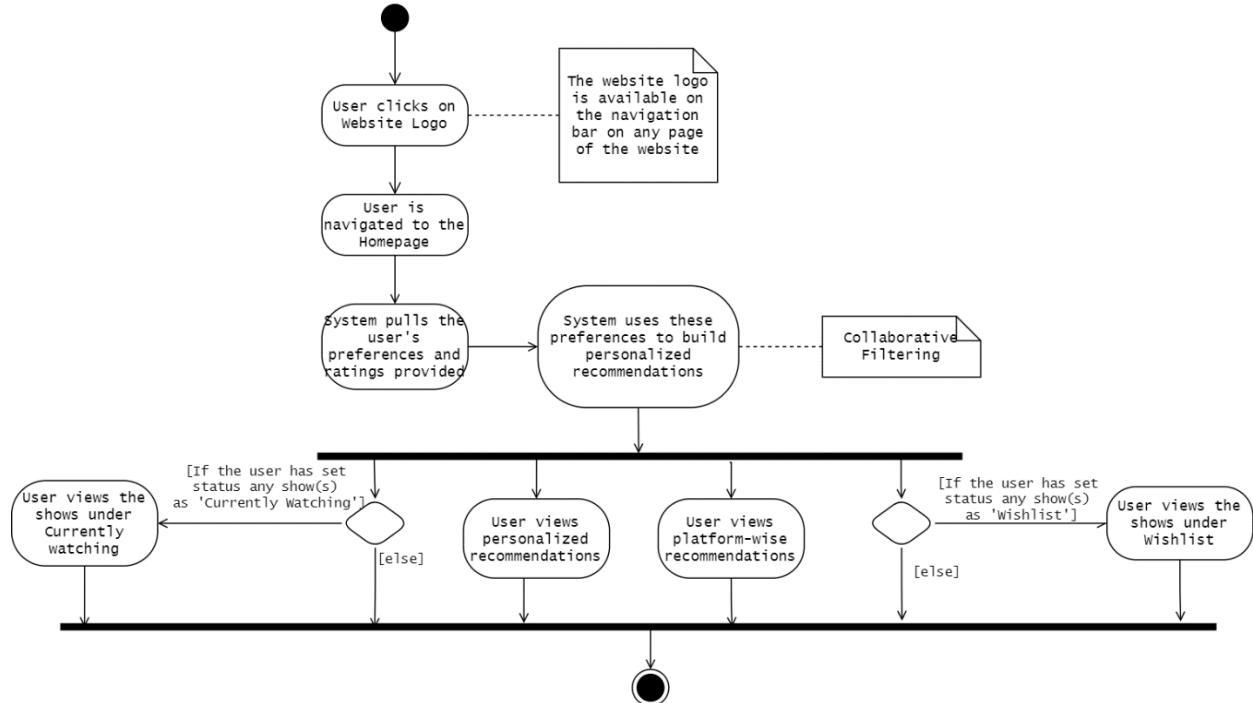
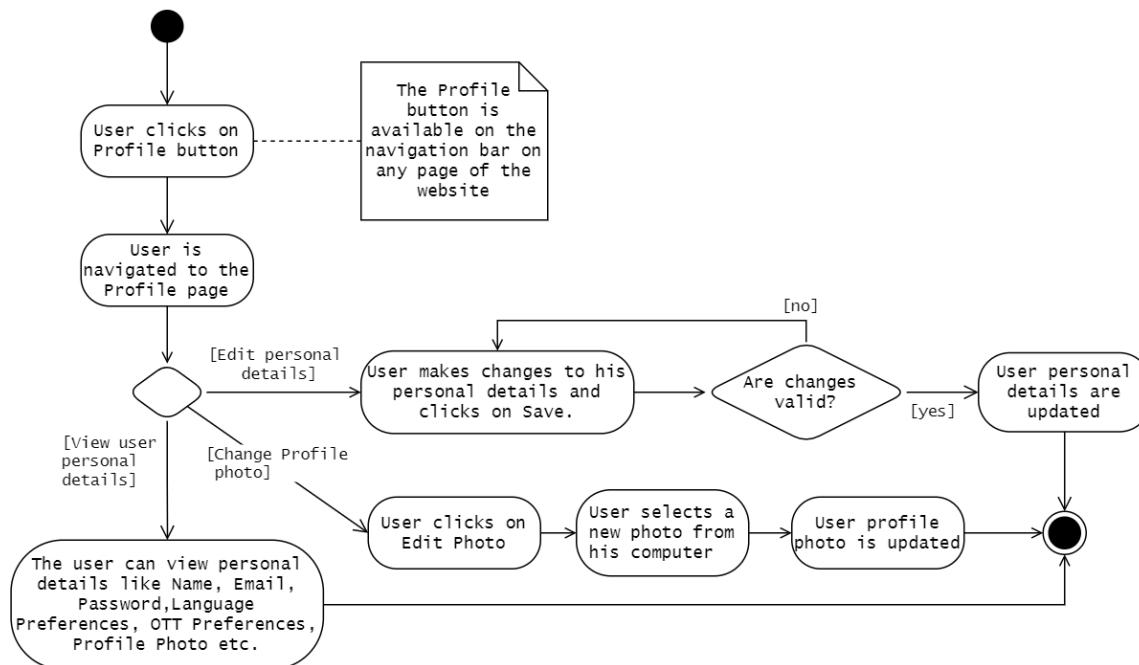


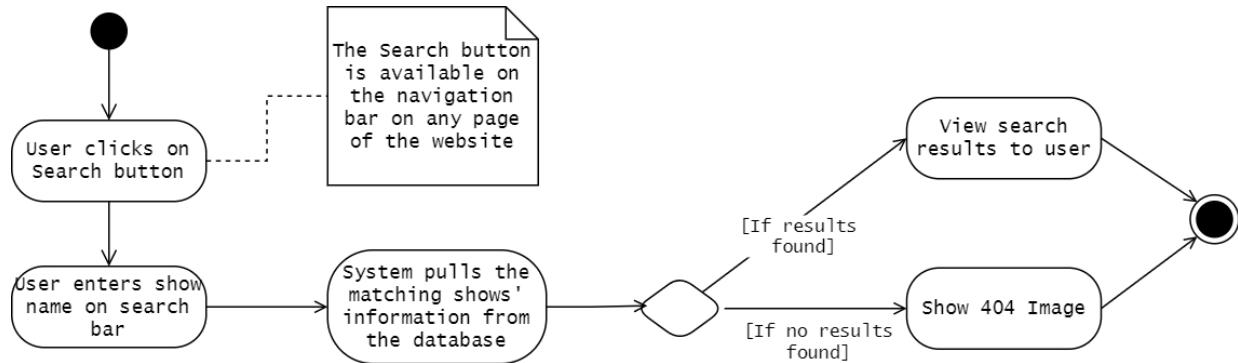
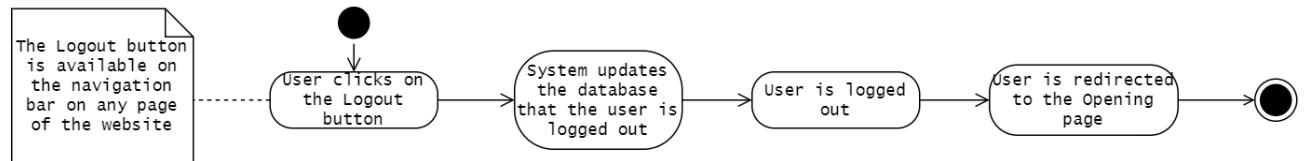
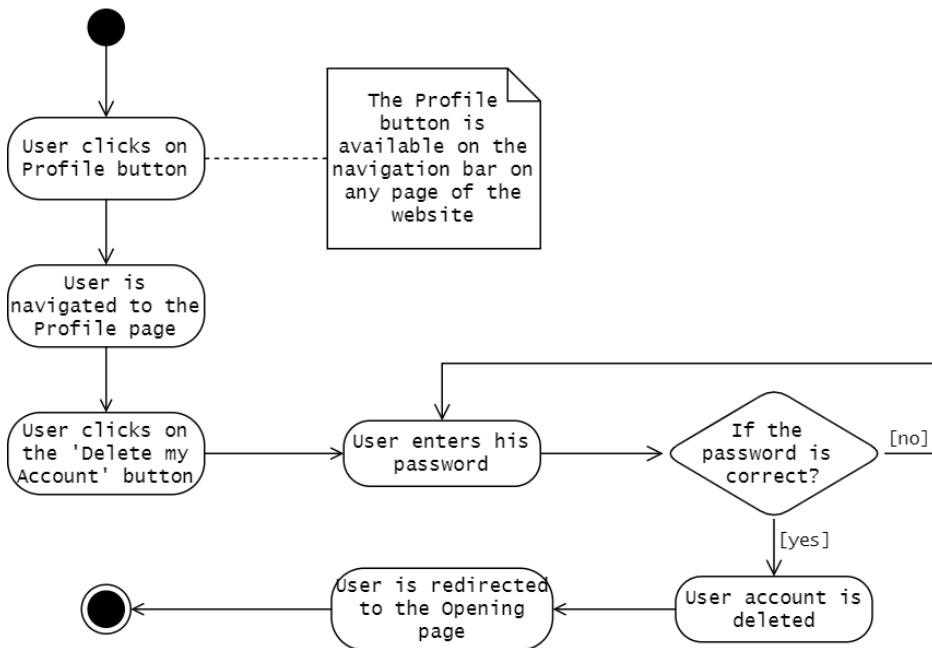
**Figure 4**

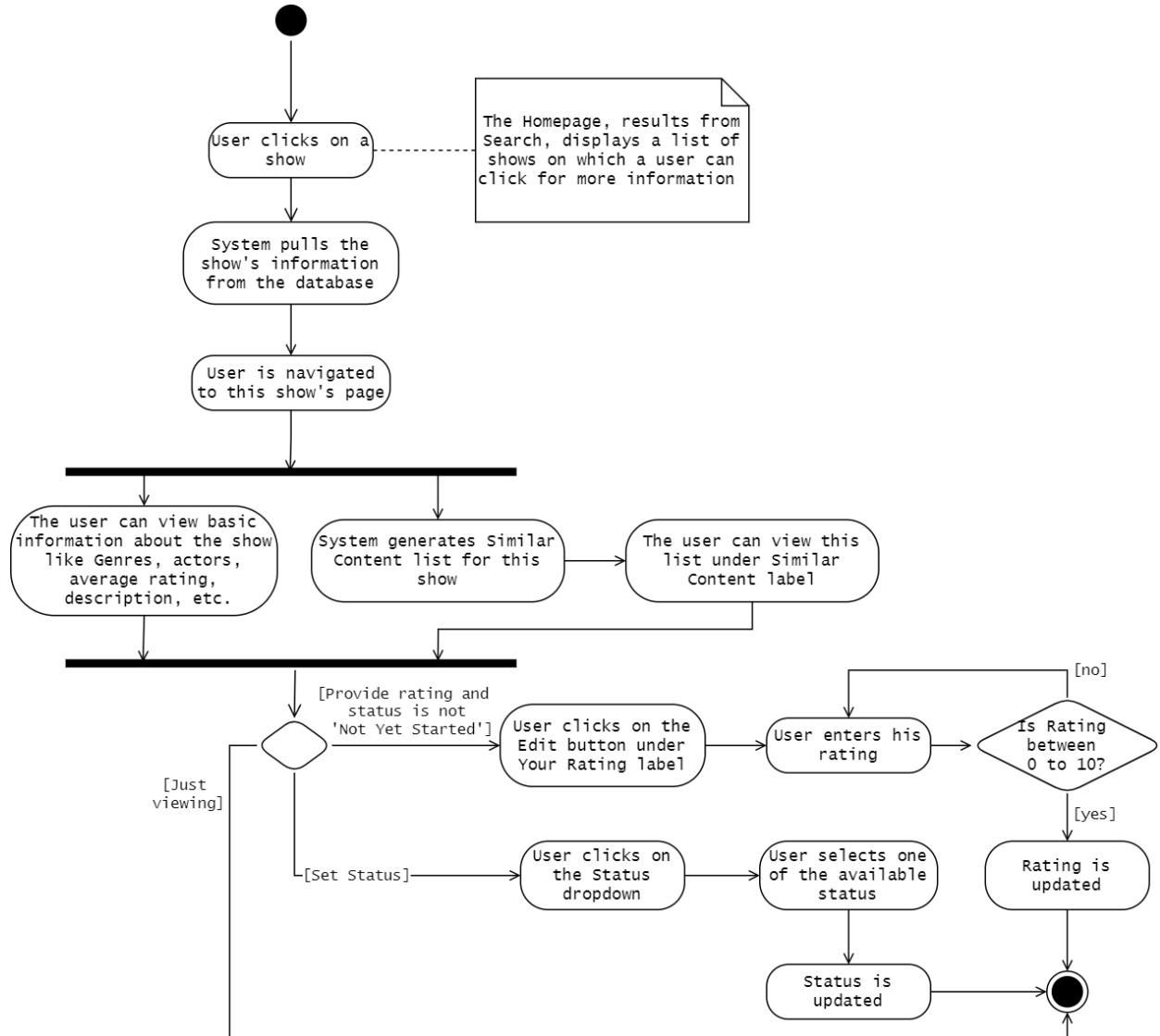
*Sign Up Activity Diagram*

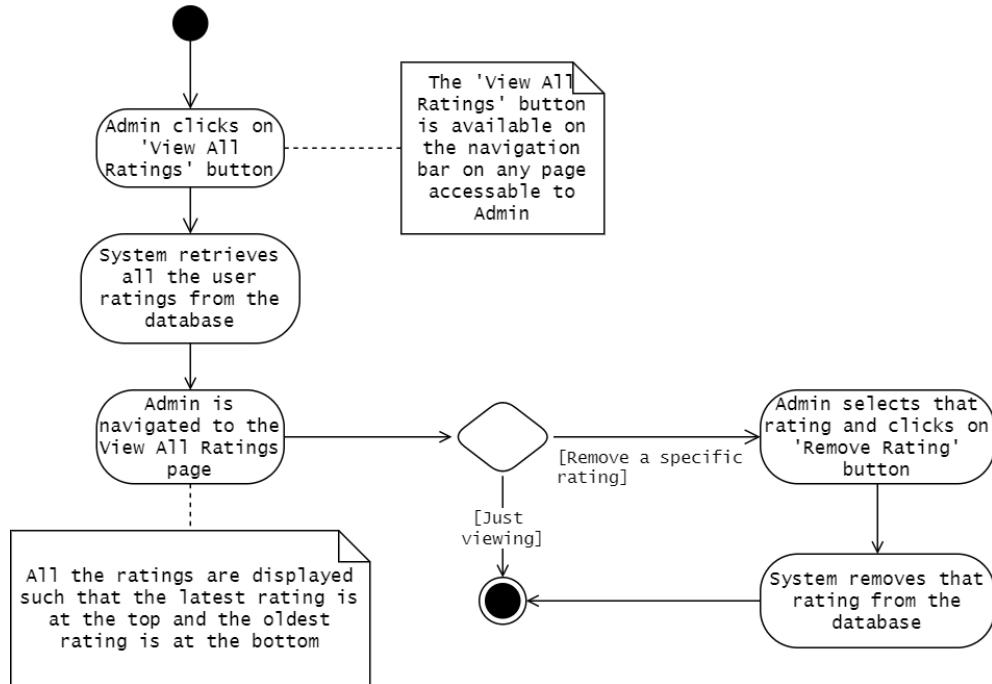
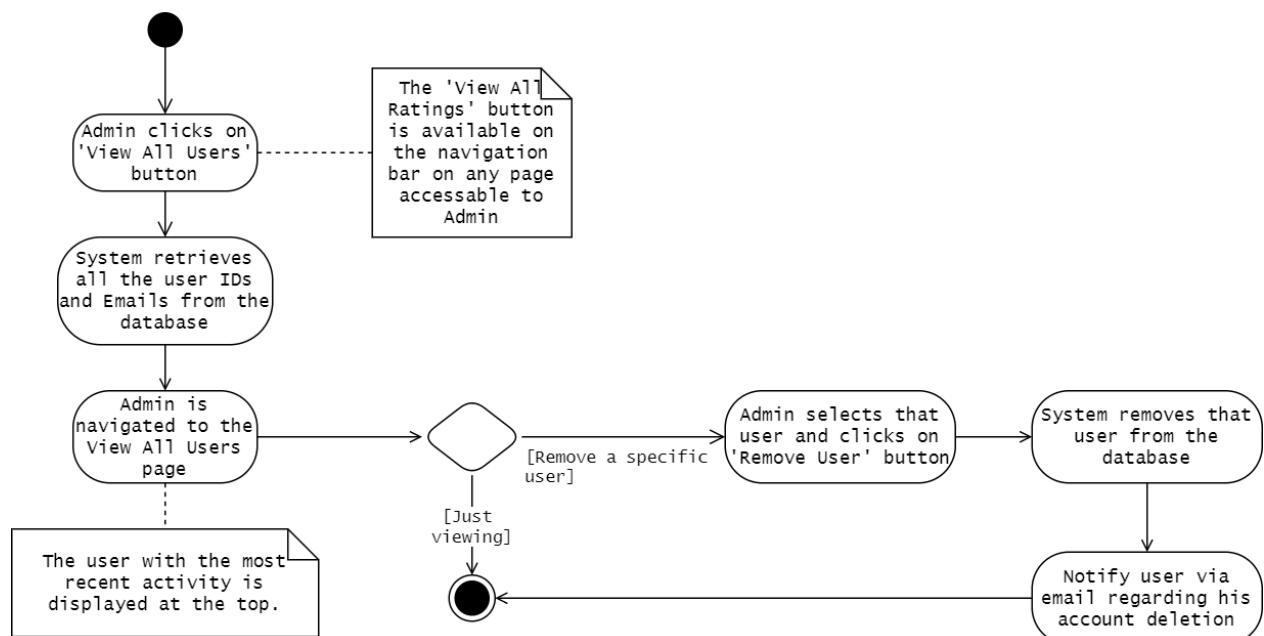


**Figure 5***Forgot Password Activity Diagram***Figure 6***New User Preferences Activity Diagram*

**Figure 7***Go To Homepage***Figure 8***View Profile*

**Figure 9***Search Activity Diagram***Figure 10***Logout***Figure 11***Delete User Account Activity Diagram*

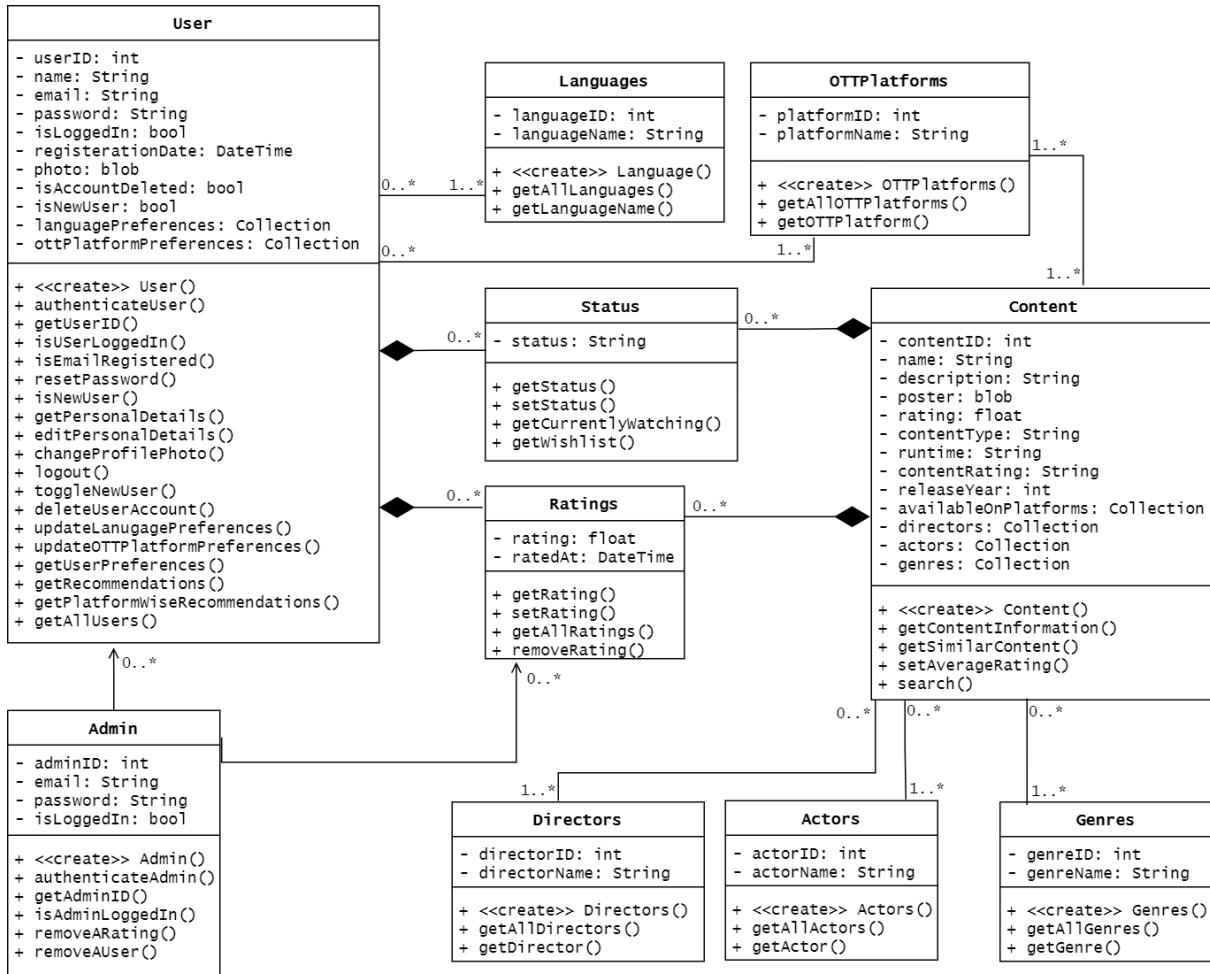
**Figure 12***View Content Information Activity Diagram*

**Figure 13***View All Ratings Activity Diagram***Figure 14***View All Users Activity Diagram*

## Class diagrams

**Figure 15**

*Class diagram for Cross-Platform Recommendation System*



## Class descriptions

**Table 56**

Class name	User
Attributes	Description
userID	A unique ID for is assigned to each user.
name	The name of the user.

---

<b>Class name</b>	User
email	The email of the user. Email is unique to each user, i.e., there cannot be two users with the same email. The user logs in using his email address.
password	The password of the user. The password is stored instead of the original password itself.
isLoggedIn	This attribute is True if the user is logged in and False otherwise.
registrationDate	The time at which the user has created his/her account.
photo	The user can set up a profile photo and this photo is stored as a BLOB.
isAccountDeleted	This attribute is True if the user deletes his account or if an admin deletes this user's account, else it is False. When this attribute is True, values for all the attributes will be cleared except for userID, isAccountDeleted. This is done to retrain this user's rating data which is used in collaborative filtering.
isNewUser	This attribute is True if this user has just created his account and has not yet set his language, or OTT preferences, else it is False.
languagePreferences	This is a collection of Languages objects, i.e., the languages that the user has selected. The user will get recommendations only in these languages.
ottPlatformPreferences	This is a collection of OTTPlatforms objects. The user will get recommendations for the content that are present only on these OTT platforms.

---

<b>Class name</b>	User
<b>Methods</b>	<b>Description</b>
User()	This constructor is used to create an object for a new User.
authenticateUser()	This method is used to authenticate the user during login. The credentials provided by the user is matched with the one stored in the database. If the credentials match, True and isLoggedIn attribute is set to True.
getUserID()	Given the email of the user, this method returns the userID attribute.
isUserLoggedIn()	This method returns True if the user is logged in, else it returns False.
isEmailRegistered()	This method returns True if there is an account already registered with the given email, else it returns False.
resetPassword()	This method updates the password given the userID and the new password.
isNewUser()	This method returns True if the user is new, i.e., the user has just created an account and has not yet set his preferences. Else it returns False.
getPersonalDetails()	This method is called when the user clicks on the Profile button on any page of the website and returns the details of the user such as personal details, preferences, etc.
editPersonalDetails()	This method is used to update the user details and returns True if the details are successfully updated.

---

<b>Class name</b>	User
changeProfilePhoto()	If the user updates his profile photo, this method is used to update the photo attribute. It returns True if the photo is updated successfully, else it returns False.
logout()	When the user clicks on Logout on the Profile page, this method is invoked where the isLoggedIn attribute is set to. This method returns True if the user is logged out successfully.
toggleNewUser()	Once the new user has set his preferences, this method will set the isNewUser attribute to False. This method returns True if the above is achieved.
deleteUserAccount()	This method is called if the user wishes to delete his account or the admin wants to delete this user's account.
updateLanguagePreferences()	This method is used to update the languagePreferences attribute for the user.
updateOTTPlatformPreferences()	This method is used to update the ottPlatformPreferences attribute for the user.
getUserPreferences()	This method returns the languagePreferences and ottPlatformPreferences attributes of the user given the user ID.
getRecommendations()	This method generates recommendations for the user based on his/her ratings provided to various movies/series and user preferences.
getPlatformwiseRecommendations()	This method generates recommendations for the user given the user ID and OTT platform name.

---

**Table 57**

<b>Class name</b>	Languages
<b>Attributes</b>	<b>Description</b>
languageID	A unique ID for is assigned to each language.
languageName	The name of the language.
<b>Methods</b>	<b>Description</b>
Languages()	This constructor is used to create an object for a new language.
getAllLanguages()	This method returns the list of all languages present.
getLanguageName()	This method returns the value of the languageName attribute given the languageID.

**Table 58**

<b>Class name</b>	OTTPlatforms
<b>Attributes</b>	<b>Description</b>
platformID	A unique ID for is assigned to each OTT platform.
platformName	The name of the OTT platform.
<b>Methods</b>	<b>Description</b>
OTTPlatforms()	This constructor is used to create an object for a new platform.
getAllOTTPlatforms()	This method returns the list of all OTT platforms present.
getOTTPlatform()	This method returns the value of the platformName attribute given the platformID.

**Table 59**

<b>Class name</b>	Content
<b>Attributes</b>	<b>Description</b>
contentID	A unique ID is assigned to each movie/series.
name	The name of the show.
description	A small description or summary of the movie/series.
poster	The poster of a show is stored as a BLOB in this attribute.
rating	The average of ratings provided by different users for the given content ID. Ratings range from 0 to 10.
contentType	This attribute has only two values: Movie, Series.
runtime	The runtime of the show.
contentRating	Content ratings are the minimum maturity level of the content. Example, TV-MA, PG, PG-13, etc.
releaseYear	The year in which a given content is released.
availableOnPlatforms	This attribute is a collection of the OTTPlatforms objects. Given the content ID, this attribute holds the objects of platforms on which this show is available on.
directors	This attribute is a collection of the Directors objects. Given the content ID, this attribute holds the objects of the directors for this show.
actors	This attribute is a collection of the Actors objects. Given the content ID, this attribute holds the objects of the actors in this show.

<b>Class name</b>	Content
genres	This attribute is a collection of the Genres objects. Given the content ID, this attribute holds the objects of the genres for this show.
<b>Methods</b>	<b>Description</b>
Content()	This constructor is used to create an object for a new show.
getContentInformation()	This method returns all the information associated with the given content ID.
getSimilarContent()	This method applies content-based filtering to find shows similar to the given content ID and returns the list of shows that are similar to the given content ID.
setAverageRating()	The rating for the given show is updated regularly as the users provide their ratings. In this method, the rating attribute is updated.
search()	Given the keywords entered by the user, this method returns the information that consists of the keywords entered.

**Table 60**

<b>Class name</b>	Status
<b>Attributes</b>	<b>Description</b>
status	This attribute can be set to any of the following values: Currently Watching, Finished, Dropped, Not Yet Started, Wishlist. This is set by a user for a show.

<b>Class name</b>	Status
<b>Methods</b>	<b>Description</b>
getStatus()	This method returns the status given the user ID and content ID.
setStatus()	This method set the status by a user ID for a content ID.
getCurrentlyWatching()	This method returns the list of shows for which the user has set the status to ‘Currently Watching’.
getWishlist()	This method returns the list of shows for which the user has set the status to ‘Wishlist’.

**Table 61**

<b>Class name</b>	Ratings
<b>Attributes</b>	<b>Description</b>
rating	This attribute is set by a user for a show. The rating can have a float value from 0 to 10.
ratedAt	This attribute saves the time at which a user ID has rated a content ID.
<b>Methods</b>	<b>Description</b>
getRating()	This method returns the rating set by a user ID for a content ID.
setRating()	This method sets the rating attribute with the value entered by the user ID for a content ID. This method also sets the ratedAt attribute to the time at which the user has given the rating.

**Table 62**

<b>Class name</b>	Directors
<b>Attributes</b>	<b>Description</b>
directorID	A unique ID for is assigned to each director.
directorName	The name of the director.
<b>Methods</b>	<b>Description</b>
Director()	This constructor is used to create an object for a new director.
getAllDirectors()	This method returns the list of all directors present.
getDirector()	This method returns the value of the directorName attribute given the directorID.

**Table 63**

<b>Class name</b>	Actors
<b>Attributes</b>	<b>Description</b>
actorID	A unique ID for is assigned to each actor.
actorName	The name of the actor.
<b>Methods</b>	<b>Description</b>
Actor()	This constructor is used to create an object for a new actor.
getAllActors()	This method returns the list of all actors present.
getActor()	This method returns the value of the actorName attribute given the actorID.

**Table 64**

<b>Class name</b>	Genres
<b>Attributes</b>	<b>Description</b>
genreID	A unique ID for is assigned to each genre.
genreName	The name of the genre.
<b>Methods</b>	<b>Description</b>
Genre()	This constructor is used to create an object for a new genre.
getAllGenres()	This method returns the list of all genres present.
getGenre()	This method returns the value of the genreName attribute given the genreID.

**Table 65**

<b>Class name</b>	Admin
<b>Attributes</b>	<b>Description</b>
adminID	A unique ID for is assigned to each admin.
email	The email of the admin. Email is unique to each admin, i.e., there cannot be two admins with the same email. The admin logs in using his email address.
password	The password of the admin. The password is stored instead of the original password itself.
isLoggedIn	This attribute is True if the admin is logged in and False otherwise.

---

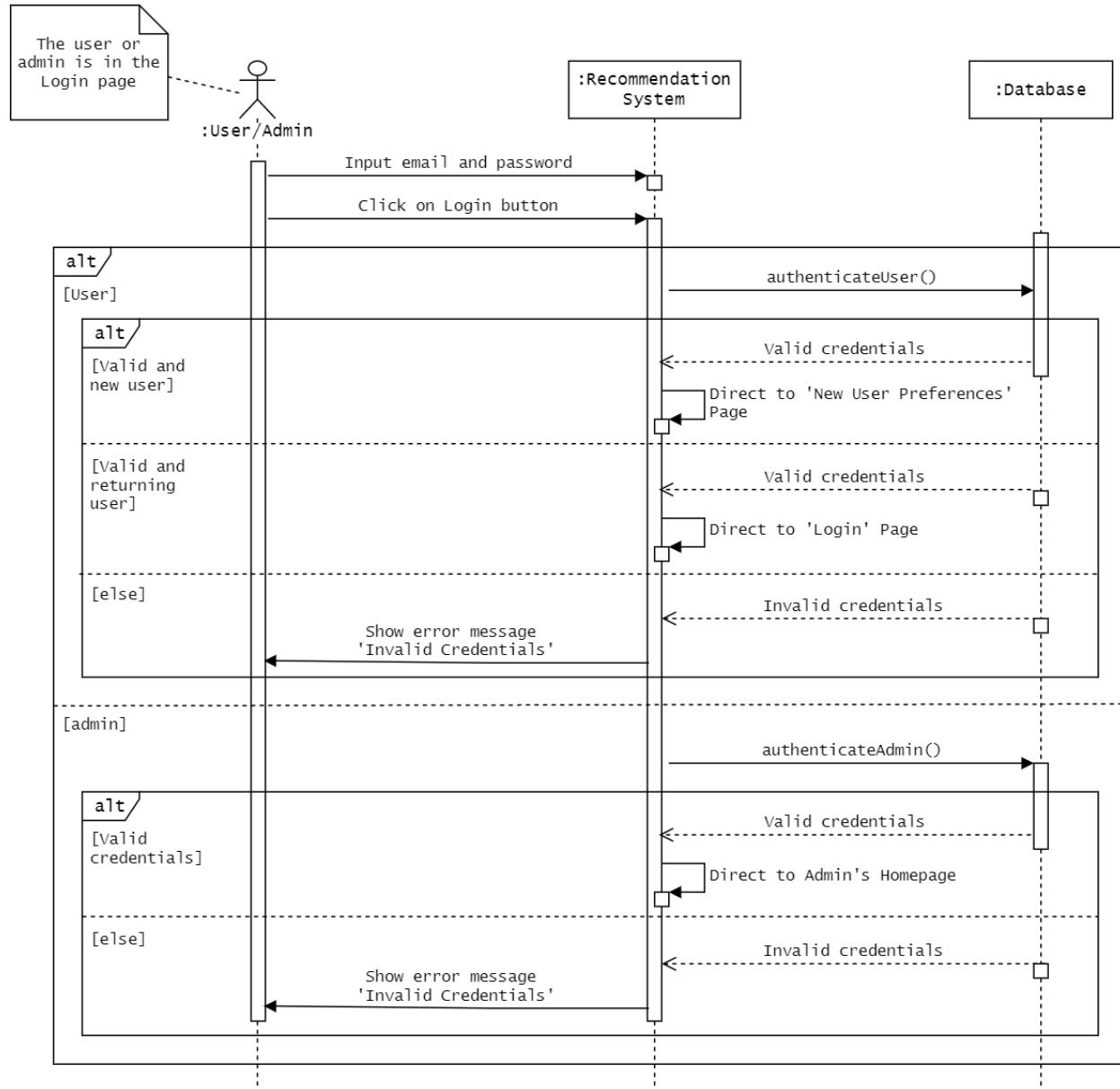
<b>Class name</b>	Admin
<b>Methods</b>	<b>Description</b>
Admin()	This constructor is used to create an object for a new Admin.
authenticateAdmin()	This method is used to authenticate the admin during login. The credentials provided by the admin is matched with the one stored in the database. If the credentials match, True and isLoggedIn attribute is set to True.
getAdminID()	Given the email of the admin, this method returns the adminID attribute.
isAdminLoggedIn()	This method returns True if the admin is logged in, else it returns False.
removeARating()	The admin can remove a rating provided by a user using this method.
removeAUser()	This method is called if the admin wants to remove or delete a user's account due to offensive activity.

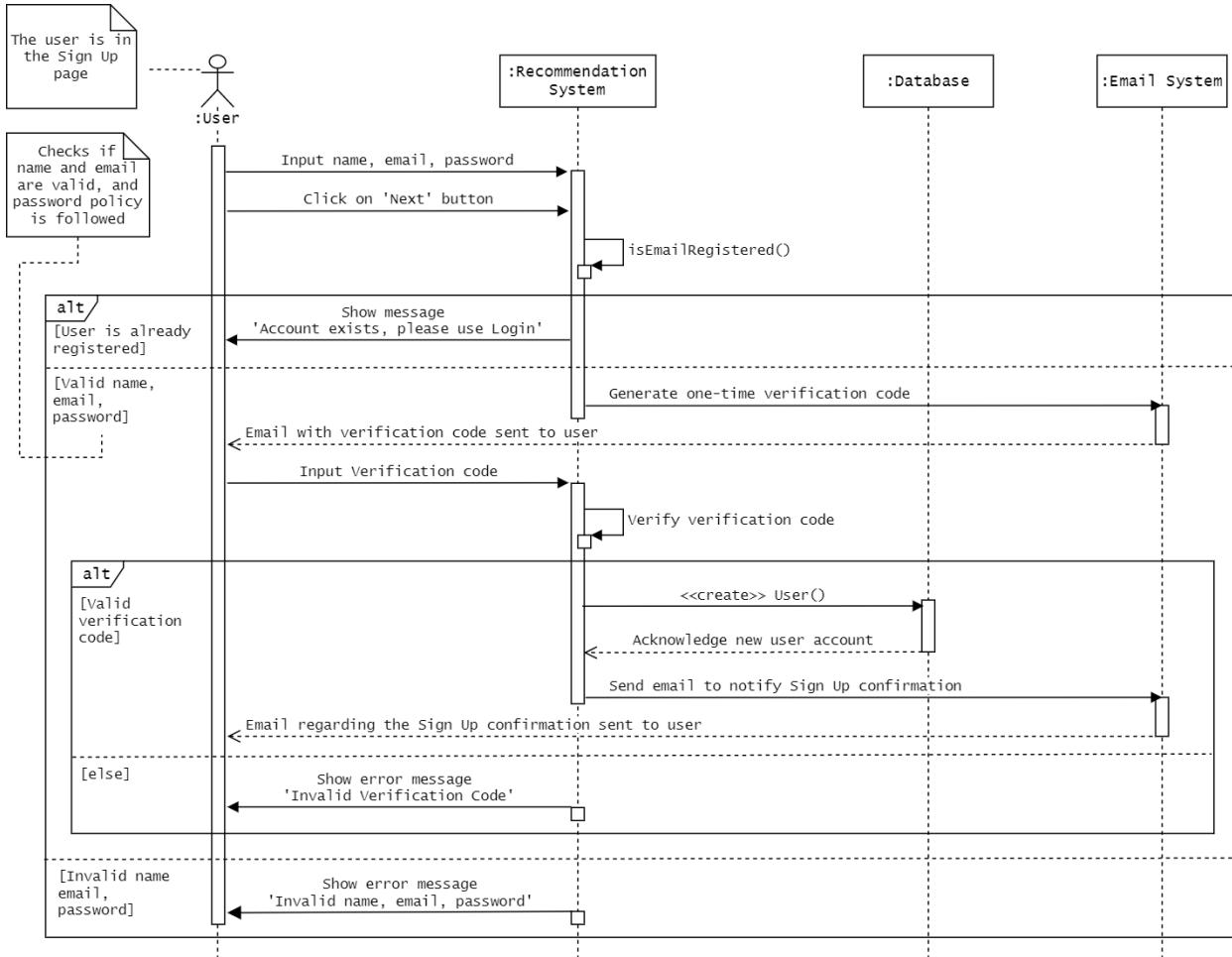
---

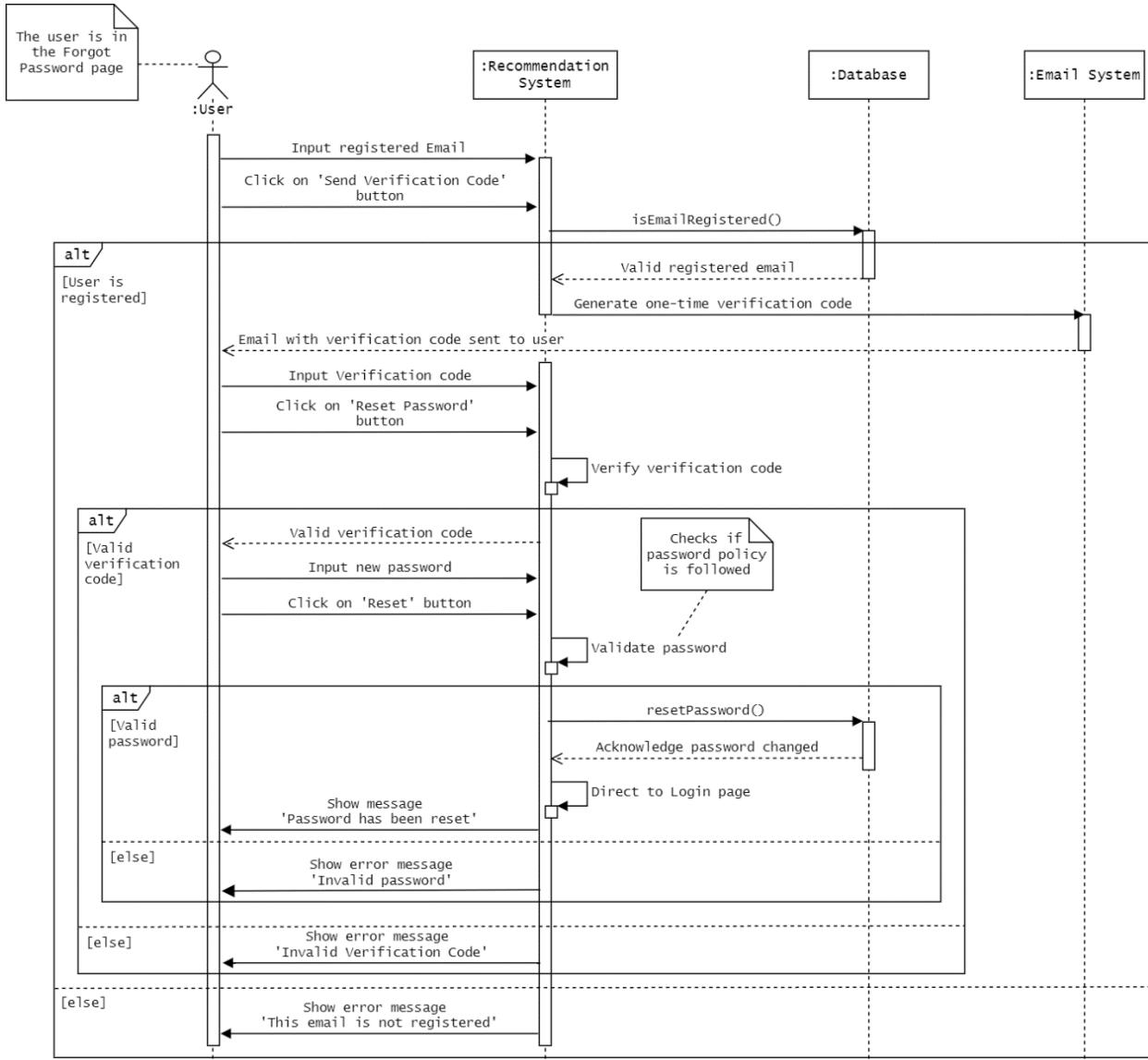
## Sequence diagrams

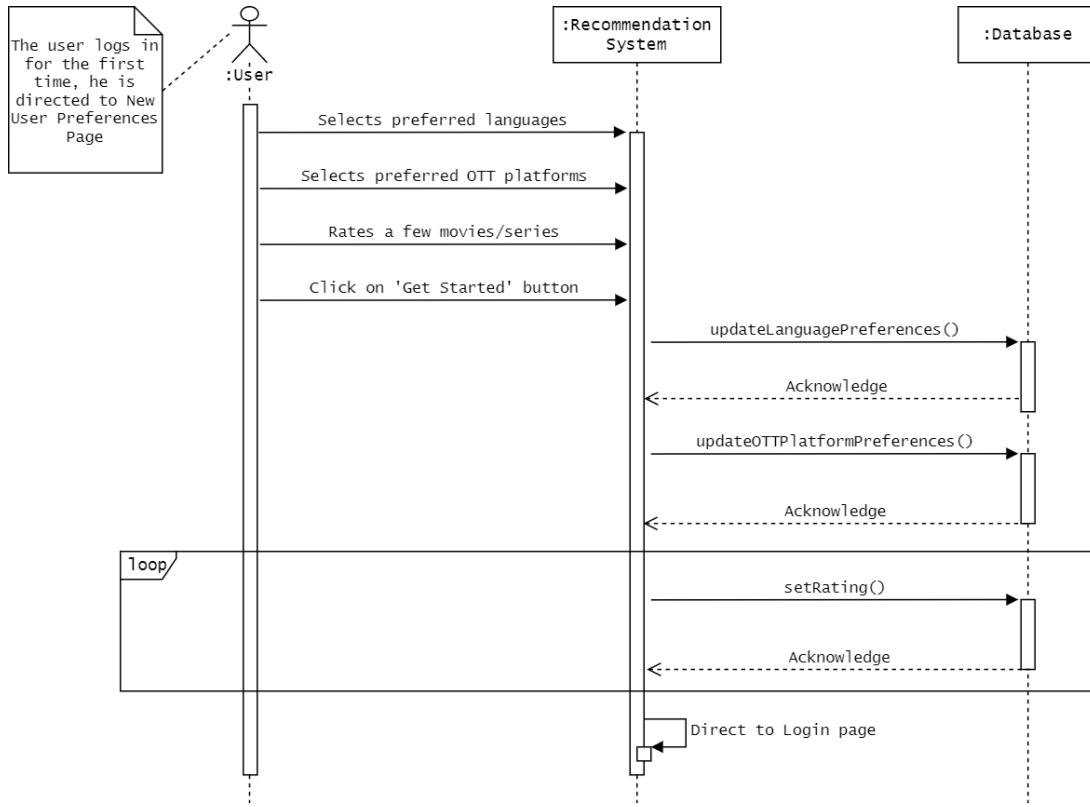
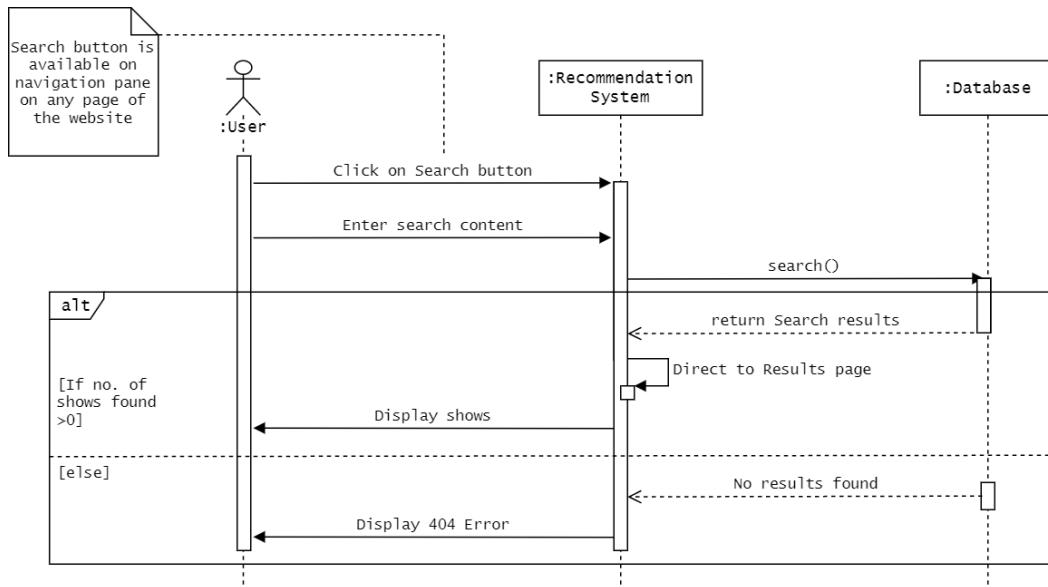
**Figure 16**

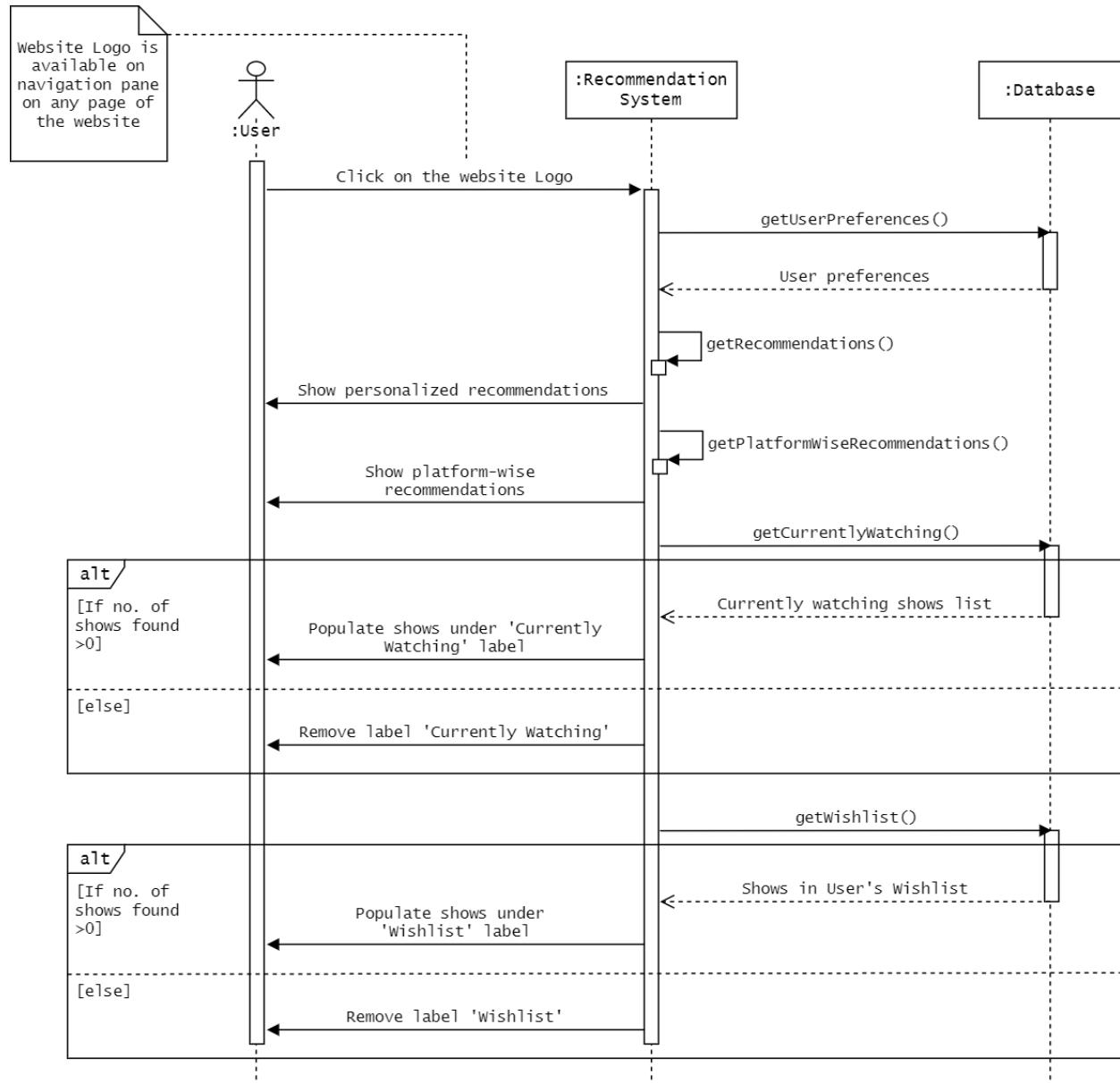
*Login Sequence Diagram*

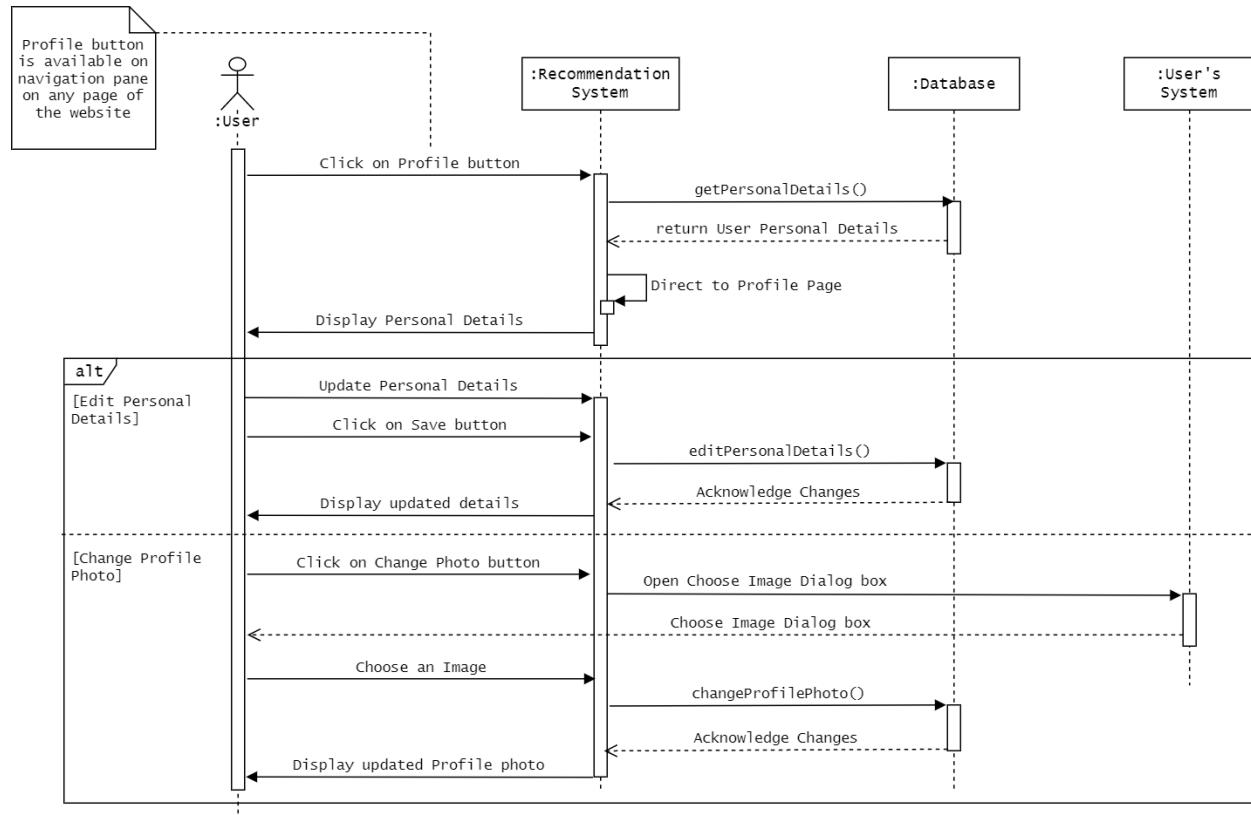
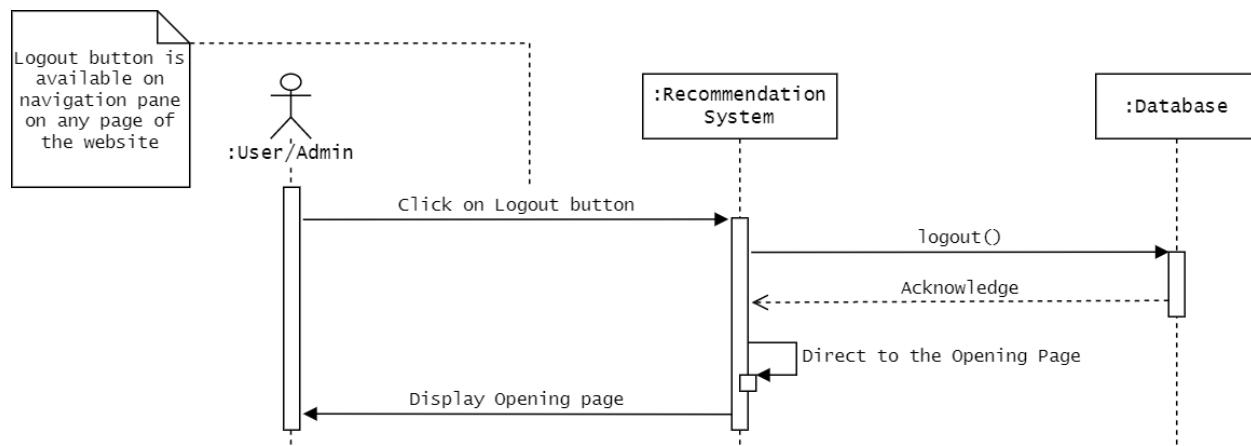


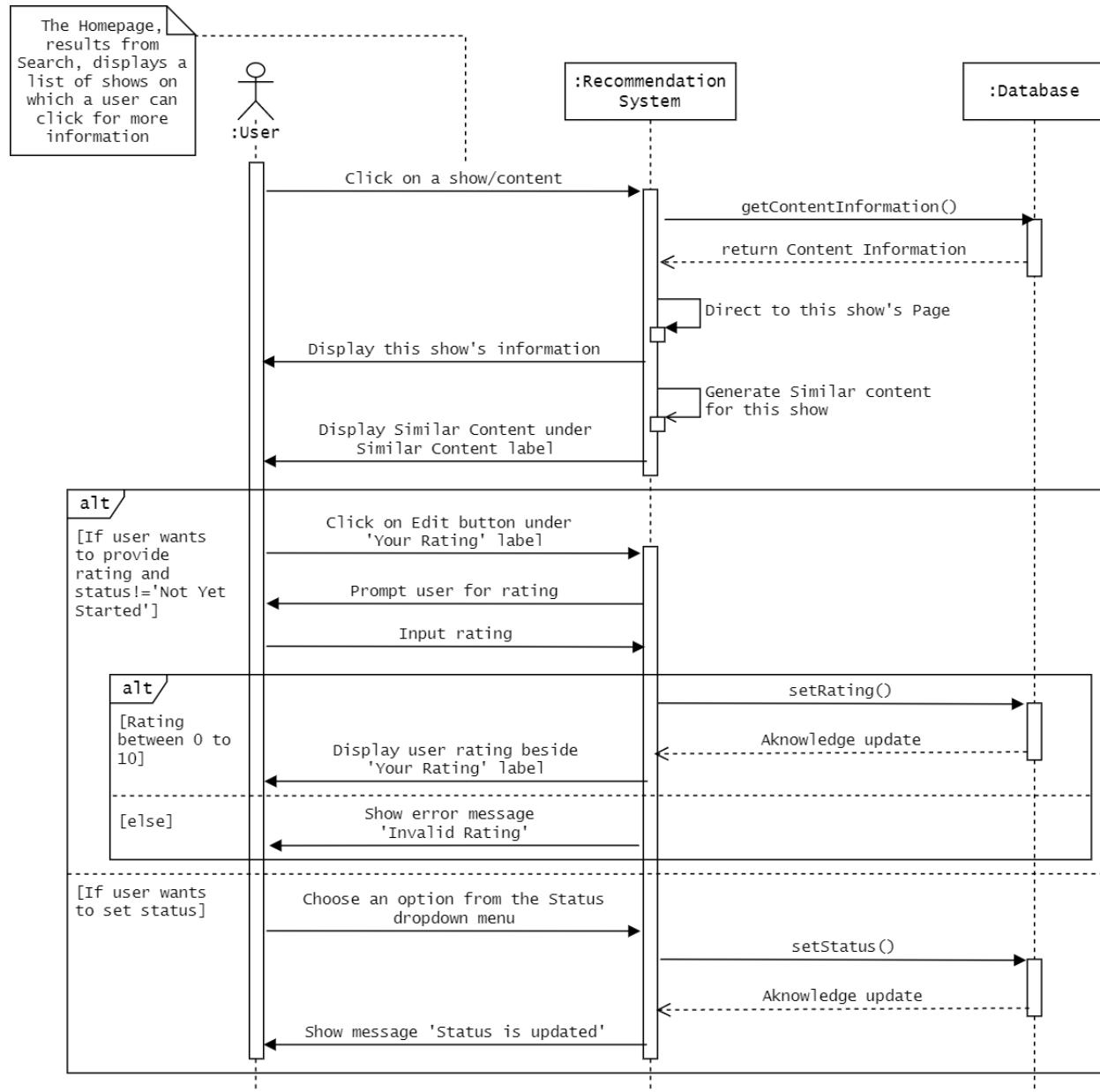
**Figure 17***Sign Up Sequence Diagram*

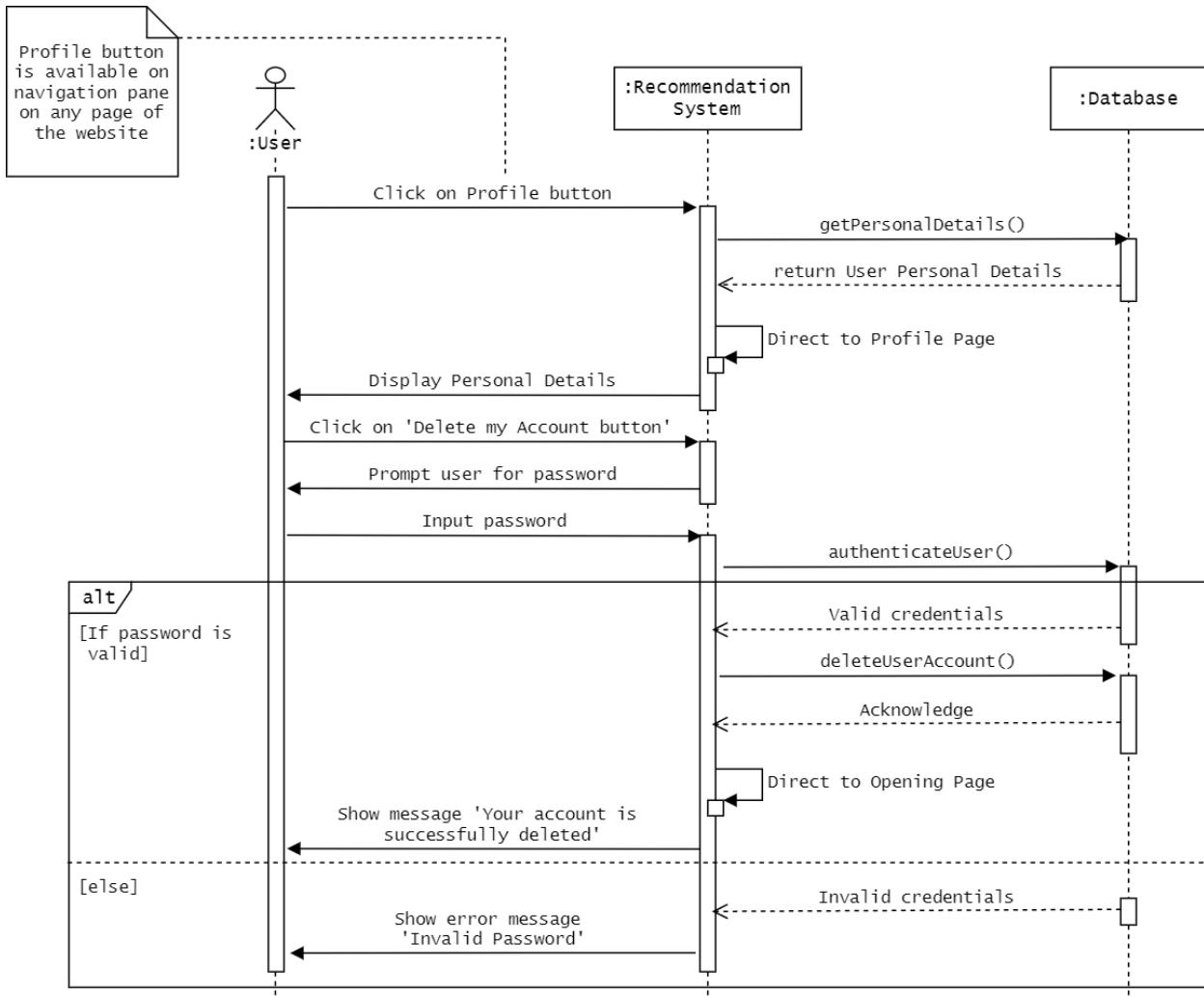
**Figure 18***Forgot Password Sequence Diagram*

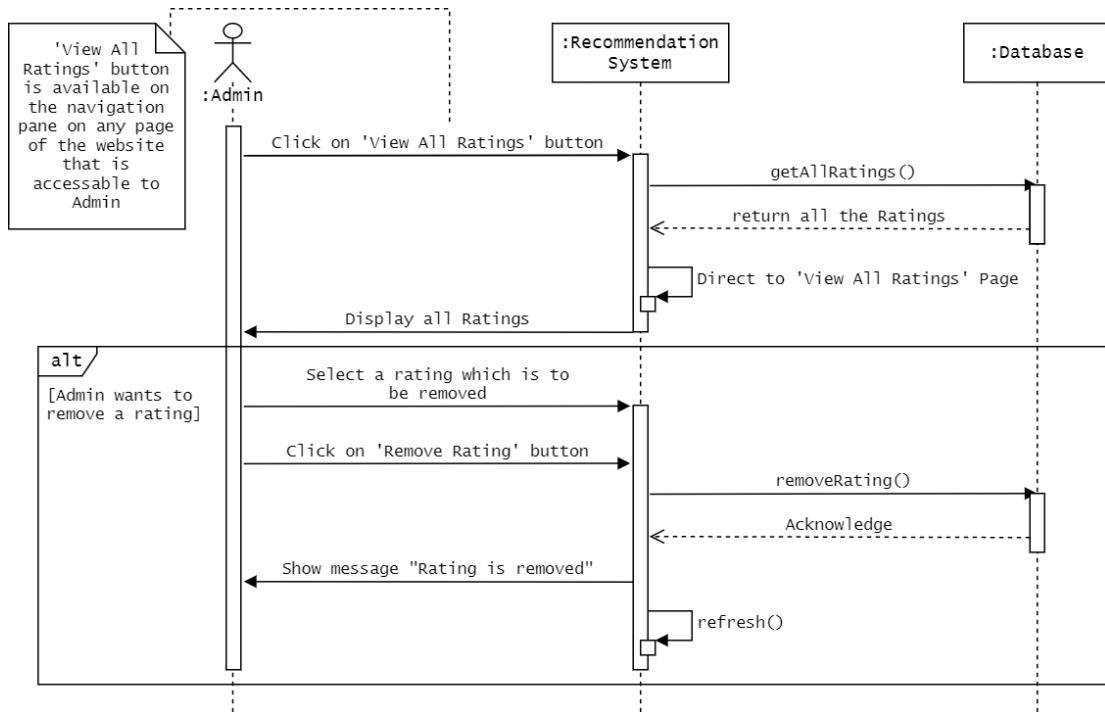
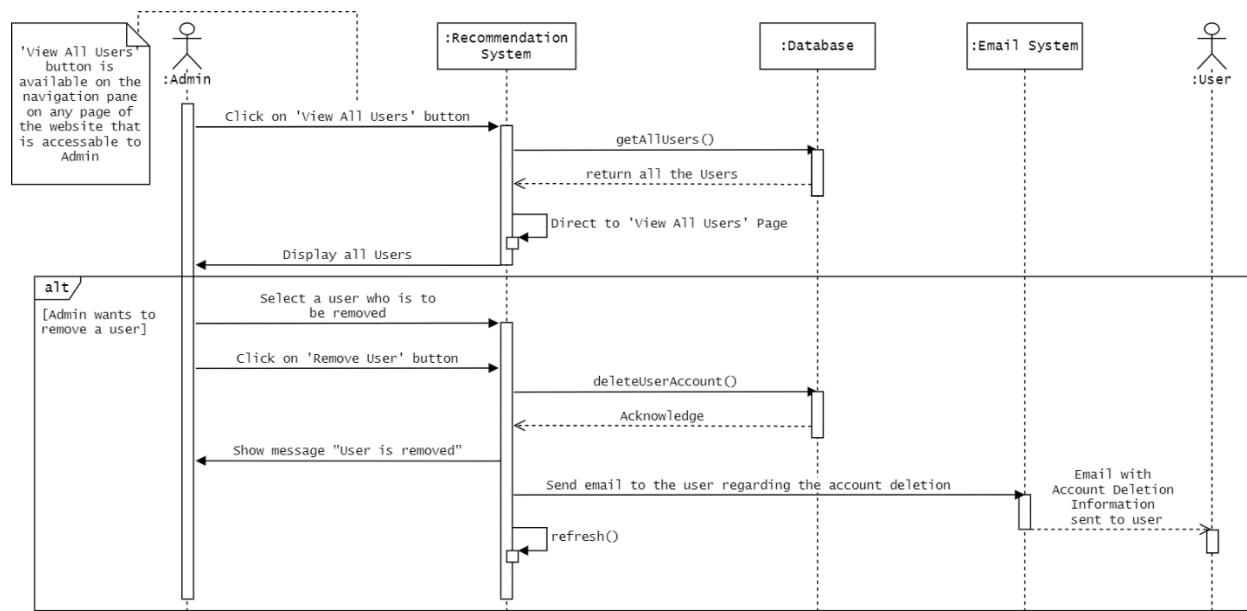
**Figure 19***New User Preferences Sequence Diagram***Figure 20***Search Sequence Diagram*

**Figure 21***Go To Homepage Sequence Diagram*

**Figure 22***View Profile Sequence Diagram***Figure 23***Logout Sequence Diagram*

**Figure 24***View Content Information Sequence Diagram*

**Figure 25***Delete User Account Sequence Diagram*

**Figure 26***View All Ratings Sequence Diagram***Figure 27***View All Users Sequence Diagram*

## Rationale Management

One of the issues we encountered in this project is that we couldn't find any APIs for any OTT platforms. So, to implement this idea, we looked for datasets in Kaggle, data.world, etc. Finding the datasets for generic information like which movies are present on which OTT platforms, genres, etc. wasn't difficult but finding the datasets that consist of different users who provided ratings to different movies/series was difficult. We couldn't find anything beyond the user ratings like user favorite movies/series, user movie lists, user watch history, etc. Since user ratings alone can be used to perform collaborative filtering, we are proceeding with the user ratings.

We do not have any information regarding the language of any show or its poster image in our datasets. We decided to use TMDb API for this purpose. We have filled out an application for an API key. We recently heard a response from the TMDb support team and received our API key. We've finished extracting the languages and poster for the content in our dataset and is ready to use.

UI is the face of the website or application. Criteria we use to guide our decisions is to provide a seamless and easy UI experience. Every decision is decided such a way that the user can easily walkthrough the website. We also explored various websites such as Airbnb, HBO Max, IMDb, etc., and Google Material Design on the internet for interesting UI designs that could be implemented in our project.

## Verification

Verification involves evaluating software during each life-cycle phase to ensure it meets the requirements set forth in the previous phase (Wallace & Fujii, 1989).

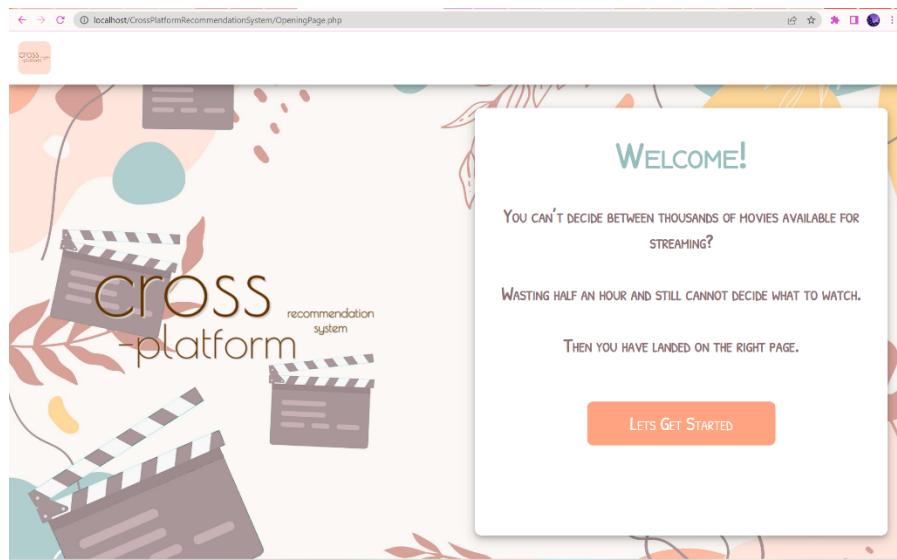
*The verification process invoked by the team*

Once we finished building the project website, we made a list of all possible system test cases for our project by first dividing the user stories into modules. For each module, we made a list of test cases. Once this list for each module is made, we assigned them to the user stories. We created new users for testing purposes on our website. We then divided the test cases among ourselves and worked on them.

## Test cases

**Table 66**

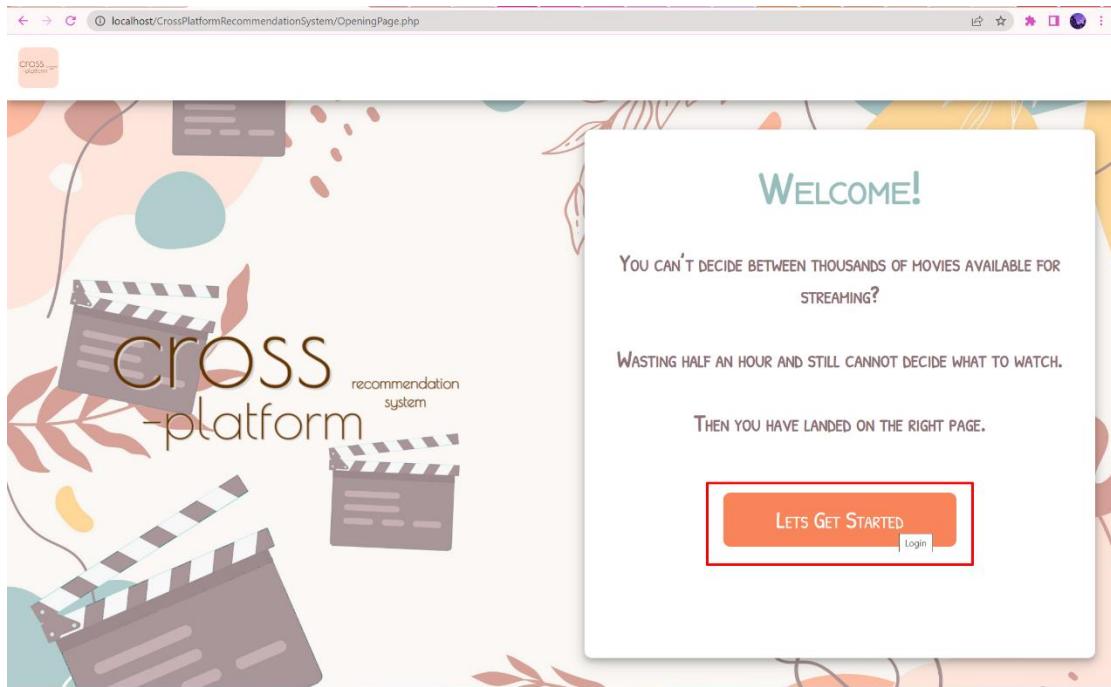
<b>Test Case ID</b>	OP_01	<b>Status</b>	Pass
<b>Module Name</b>	Opening Page		
<b>Test Case Title</b>	Opening Page		
<b>Test Case Description</b>	When the user enters the website URL, he/she is navigated to the Opening page.		
<b>Pre-conditions</b>	<b>Test Data</b>		
• Access to a browser	-		
<b>Steps</b>	1. Enter the website URL in the browser.		
<b>Expected Result</b>	<b>Actual Result</b>		
Opening page of the website will be displayed to the user (Figure 28).	Same as the Expected Result		

**Figure 28***Opening Page***Table 67**

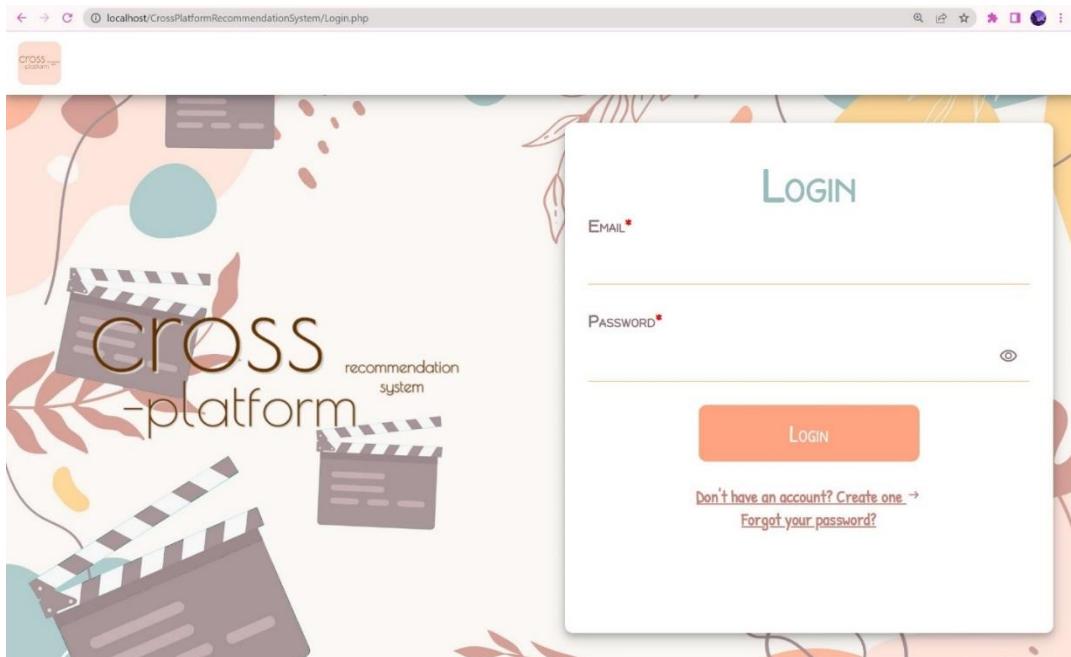
<b>Test Case ID</b>	OP_02	<b>Status</b>	Pass
<b>Module Name</b>	Opening Page		
<b>Test Case Title</b>	Opening Page – Navigate to User Login		
<b>Test Case Description</b>	When the user clicks on the 'Get Started' button, the user is navigated to the Login page.		
<b>Pre-conditions</b>	<b>Test Data</b>		
• Access to a browser	-		
<b>Steps</b>	<ol style="list-style-type: none"> <li>1. Navigate to the website's Opening page.</li> <li>2. Click on the 'Get Started' button (Figure 29).</li> </ol>		
<b>Expected Result</b>	<b>Actual Result</b>		
User is directed to the Login page (Figure 30).	Same as the Expected Result		

**Figure 29**

*Get Started button on the Opening Page*

**Figure 30**

*Login page*



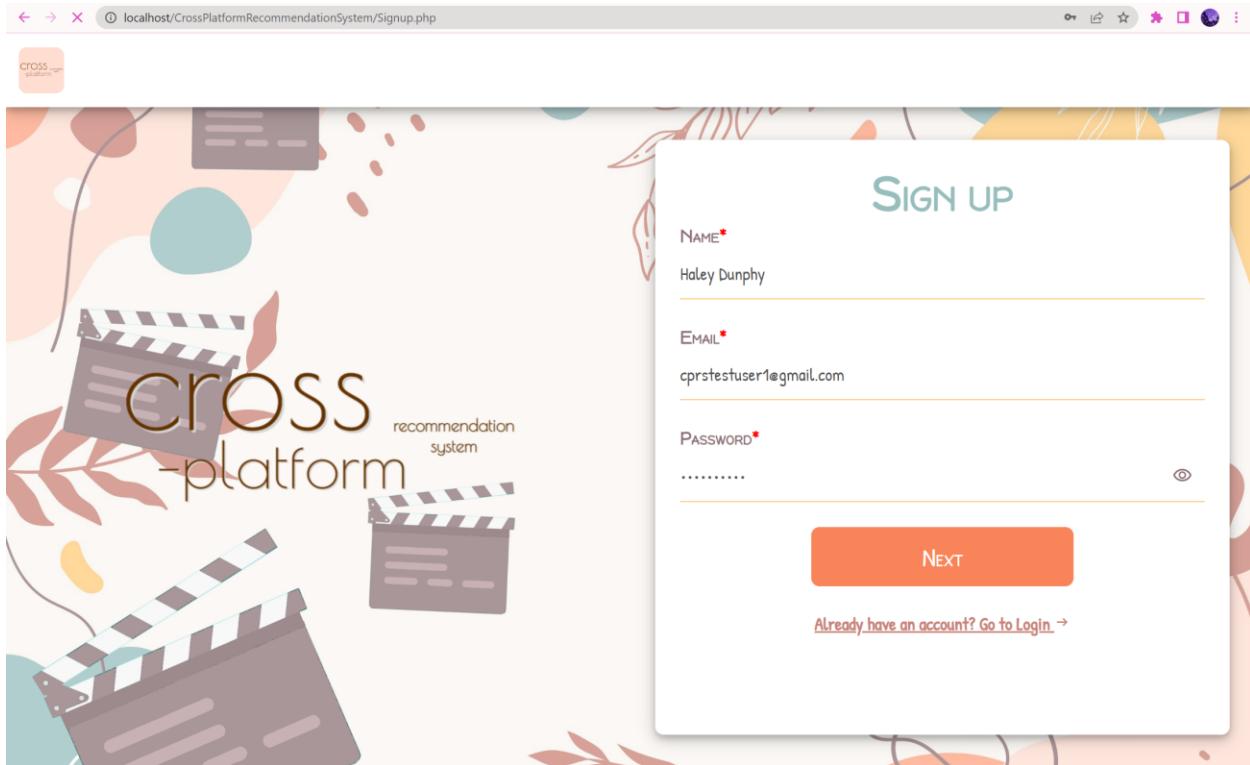
**Table 68**

<b>Test Case ID</b>	SU_01	<b>Status</b>	Pass
<b>Module Name</b>	Sign Up		
<b>Test Case Title</b>	Sign Up		
<b>Test Case Description</b>	When the user navigates to the Signup page, the user should be able to create an account.		
<b>Pre-conditions</b>	<b>Test Data</b>		
Access to a browser	name = Haley Dunphy  email = cprstestuser1@gmail.com  password = Hello@Test		
<b>Steps</b>	<ol style="list-style-type: none"> <li>1. Navigate to the Signup page.</li> <li>2. Enter the name, email, and password (Figure 31).</li> <li>3. Click on the Next button.</li> <li>4. The system sends an email with a verification code to the entered email (Figure 32).</li> <li>5. Enter the verification code (Figure 33).</li> <li>6. Click on the Verify Email button.</li> <li>7. See the ‘Your email is verified. Thank you for signing with us!’ message (Figure 34).</li> <li>8. Click on the Continue button.</li> </ol>		
<b>Expected Result</b>	<b>Actual Result</b>		
The user should be navigated to the Login page.	Same as the Expected Result		

The user will receive an confirmation email  
(Figure 35).

**Figure 31**

*Sign-up page 1*



**Figure 32**

*Verification email sent to the user*

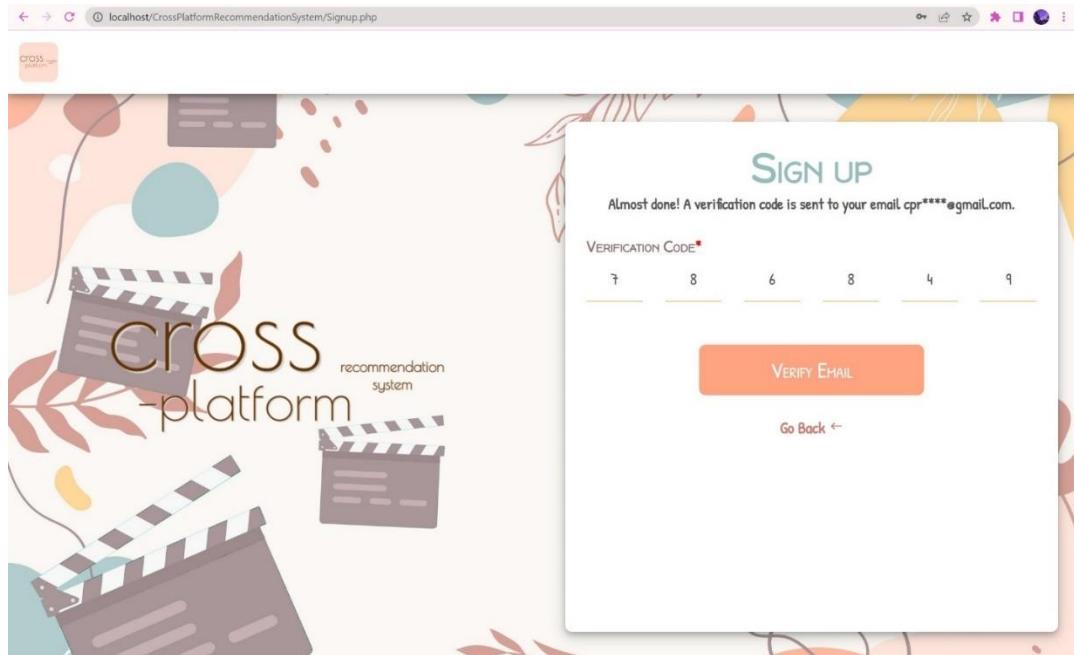
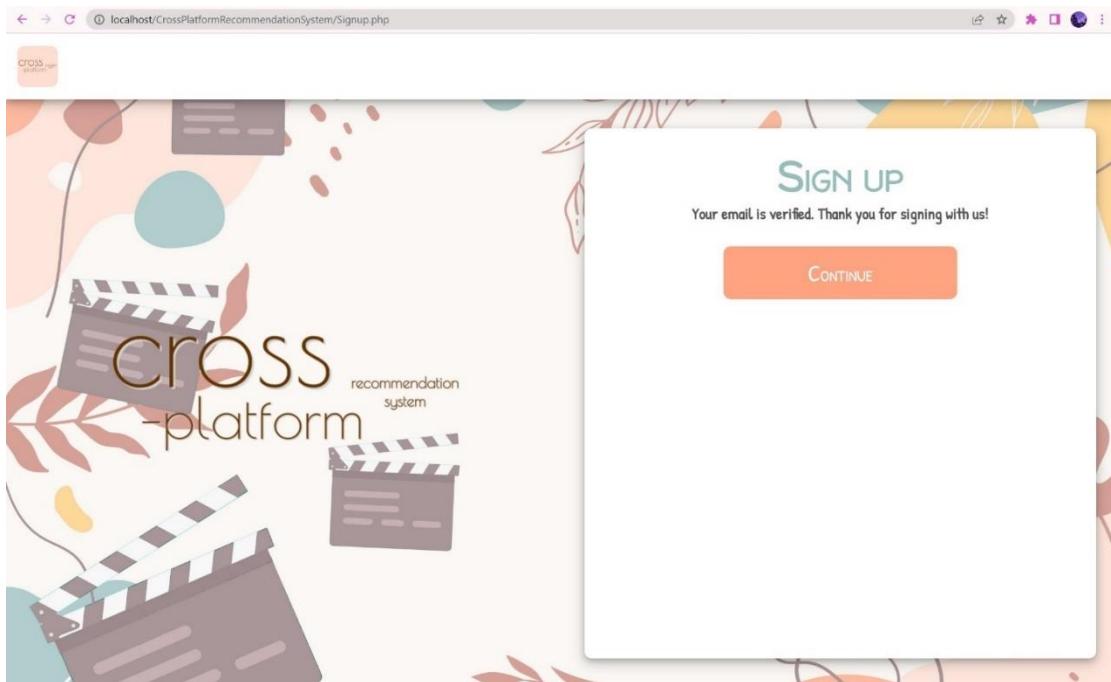
Verify your email ➤ Inbox × 🖨️ 📎

**Cross Platform Recommendation System** [crossplatformrs@gmail.com](mailto:crossplatformrs@gmail.com) via outlook.com 10:37 AM (1 minute ago) ☆ ↵ ⋮

to me ▾

Hi Haley Dunphy!

Thank you for signing up with Cross Platform Recommendation System. We need you to validate your email address to continue. Your verification code is 786849. Please enter this code in the website.

**Figure 33***Sign-up page 2***Figure 34***Sign-up page 3 – Account created*

**Figure 35**

*Confirmation email sent to the user*

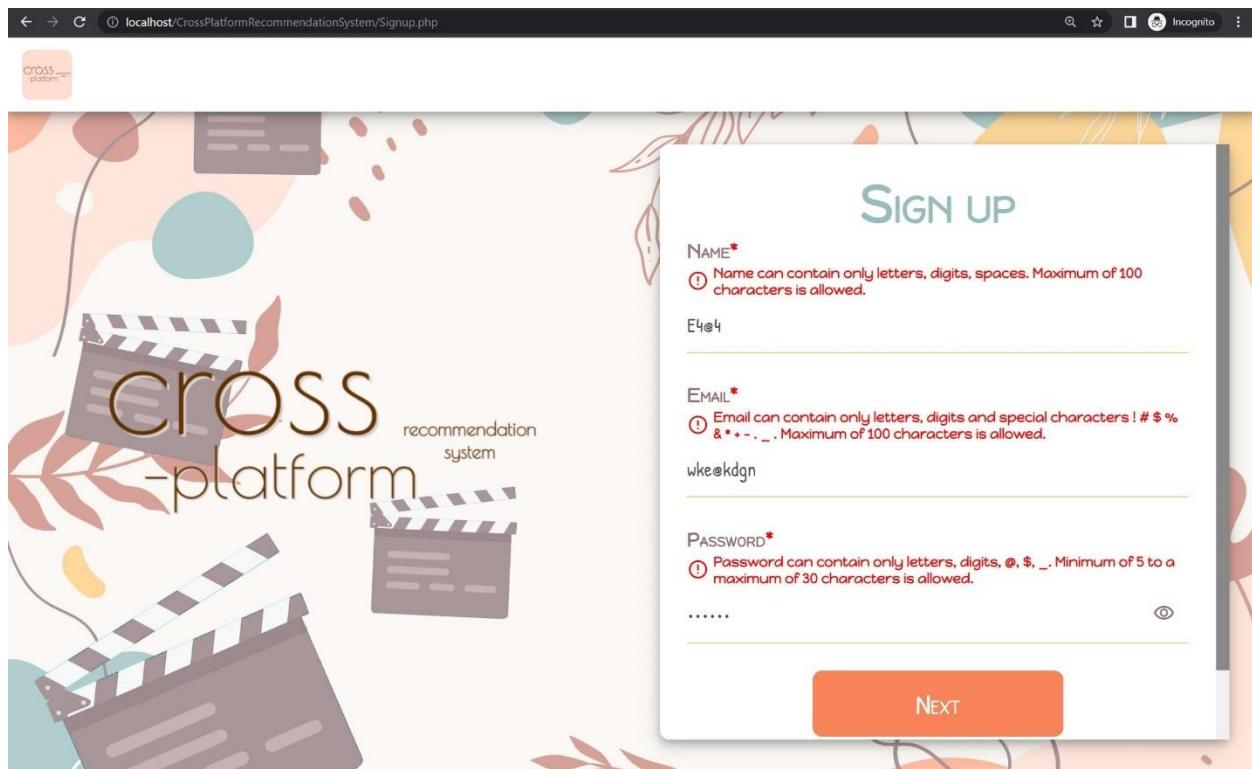
**Table 69**

<b>Test Case ID</b>	SU_02	<b>Status</b>	Pass
<b>Module Name</b>	Sign Up		
<b>Test Case Title</b>	Sign Up – Invalid User Credentials		
<b>Test Case Description</b>	When the user provides an invalid name, email, or password during Signup, the user should see the appropriate error messages.		
<b>Pre-conditions</b>	<b>Test Data</b>		
Access to a browser	name = E4@4  email = wke@kdgn  password = Hello@		
<b>Steps</b>	<ol style="list-style-type: none"> <li>1. Navigate to the Signup page.</li> <li>2. Enter the name, email, and password.</li> <li>3. Click on the Next button.</li> </ol>		
<b>Expected Result</b>	<b>Actual Result</b>		
Since the user enters an invalid name, email, and password, the user should see the appropriate error messages (Figure 36).	Same as the Expected Result		

<b>Test Case ID</b>	SU_02	<b>Status</b>	Pass
<b>Notes</b>	<ul style="list-style-type: none"> <li>• A valid name consists of only letters, numbers, and spaces and a maximum of 100 characters is allowed.</li> <li>• An email can contain only letters, digits, and special characters !#\$%&amp;*+-_. and a maximum of 100 characters is allowed.</li> <li>• A password can contain letters, digits, @, \$, _ and the length of the password should be between 5 to 30 characters.</li> </ul>		

**Figure 36**

*Errors displayed on the Sign-up page when the user enters an invalid name, email, or password*

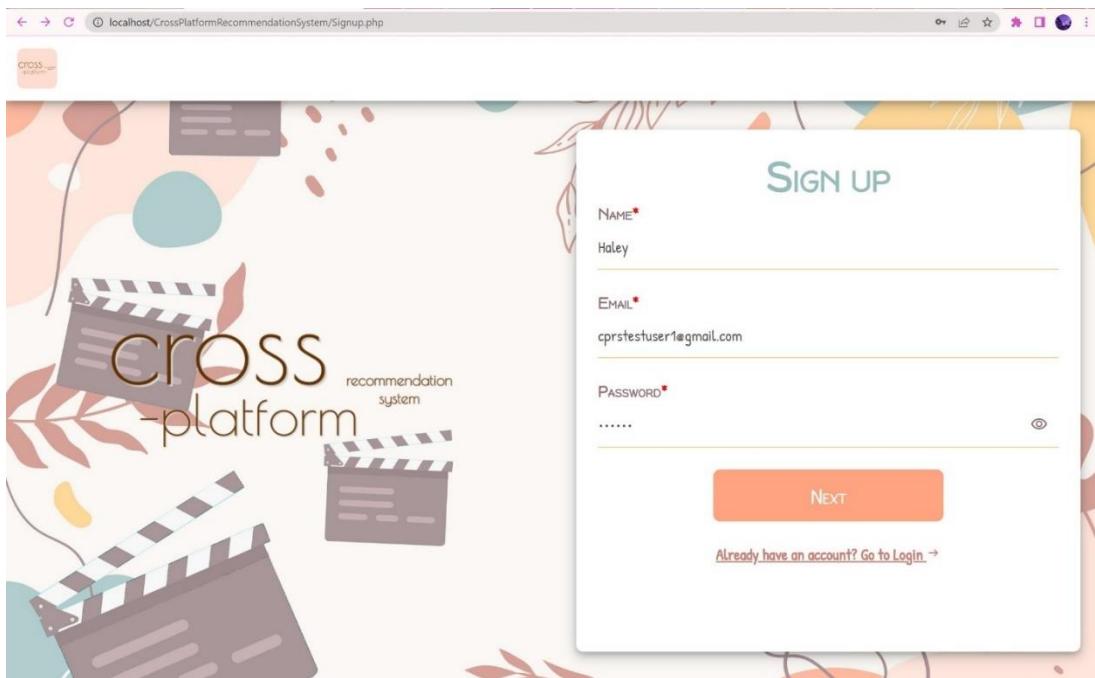


**Table 70**

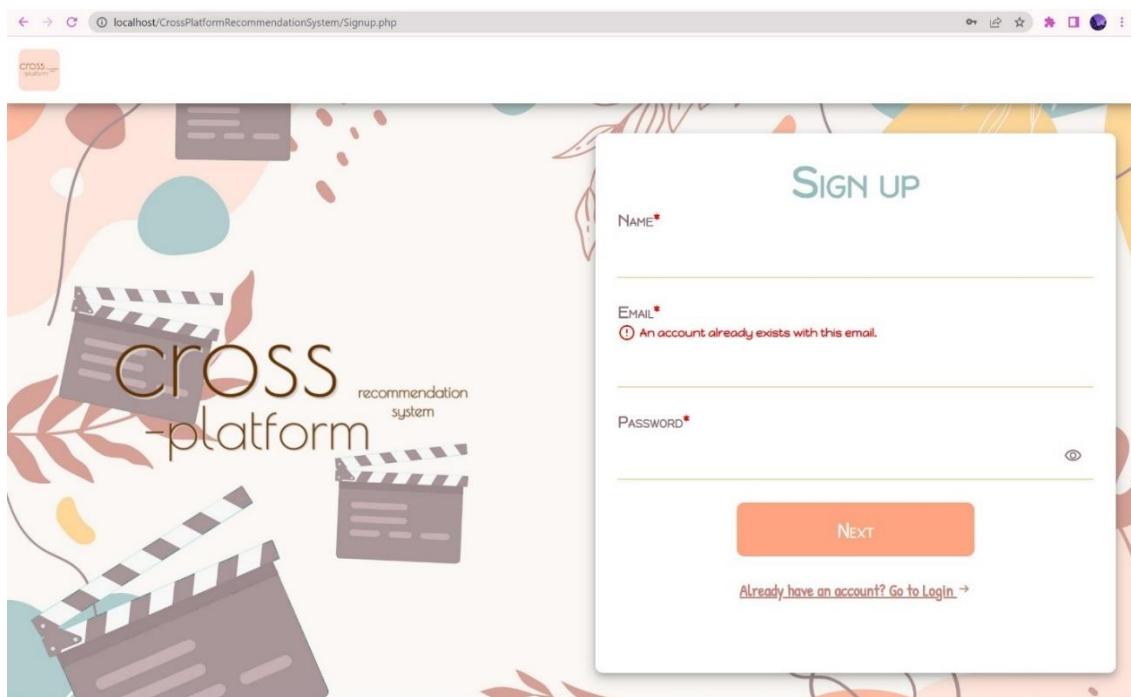
<b>Test Case ID</b>	SU_03	<b>Status</b>	Pass
<b>Module Name</b>	Sign Up		
<b>Test Case Title</b>	Sign Up – Already Registered User		
<b>Test Case Description</b>			
<b>Pre-conditions</b>	<b>Test Data</b>		
Access to a browser	name = Haley email = cprstestuser1@gmail.com password = Helloo		
<b>Steps</b>	1. Navigate to Sign-up page. 2. Enter name, email, and password (Figure 37). 3. Click on the Next button.		
<b>Expected Result</b>	<b>Actual Result</b>		
Since an account is already created on this email, the error ‘An account already exists with this email’ should be displayed (Figure 38).	Same as the Expected Result		

**Figure 37**

*Sign-up page 1 with an existing email entered*

**Figure 38**

*Error displayed when the user tries to create an account with an existing email*

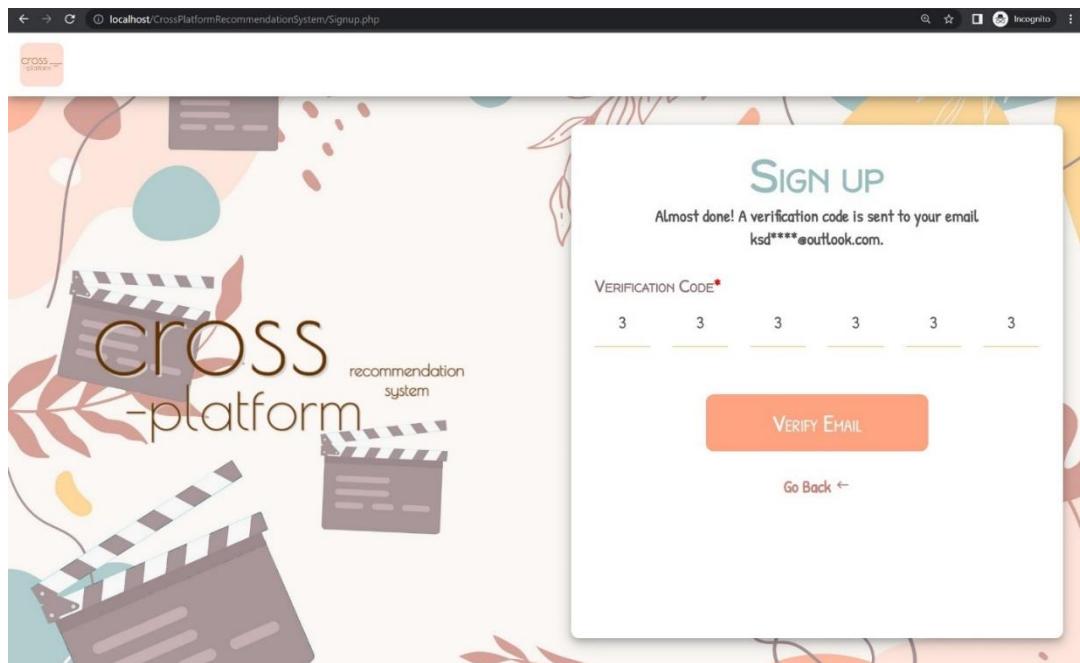


**Table 71**

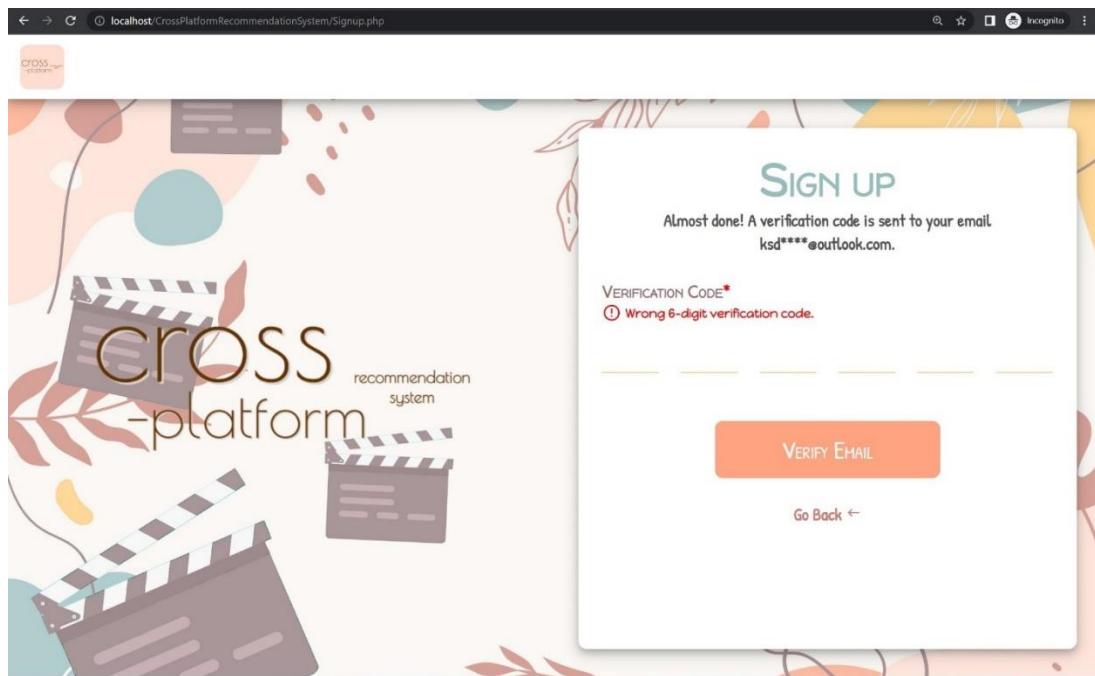
<b>Test Case ID</b>	SU_04	<b>Status</b>	Pass
<b>Module Name</b>	Sign Up		
<b>Test Case Title</b>	Sign Up – Invalid Verification Code		
<b>Test Case Description</b>	The user enters an invalid verification code in the Signup page.		
<b>Pre-conditions</b>	<p><b>Test Data</b></p> <ul style="list-style-type: none"> <li>• Access to a browser</li> <li>• The user has already entered his name, email, and password and is at the verification code page.</li> </ul>		
<b>Steps</b>	<ol style="list-style-type: none"> <li>1. Enter the verification code (Figure 39).</li> <li>2. Click on the Verify Email button.</li> </ol>		
<b>Expected Result</b>	<b>Actual Result</b>		
Since the user entered the wrong verification code, an error message ‘Wrong 6-digit verification code’ should appear to the user (Figure 40).	Same as the Expected Result		

**Figure 39**

*Incorrect verification code entered by the user*

**Figure 40**

*Error displayed when the user enters an incorrect verification code*

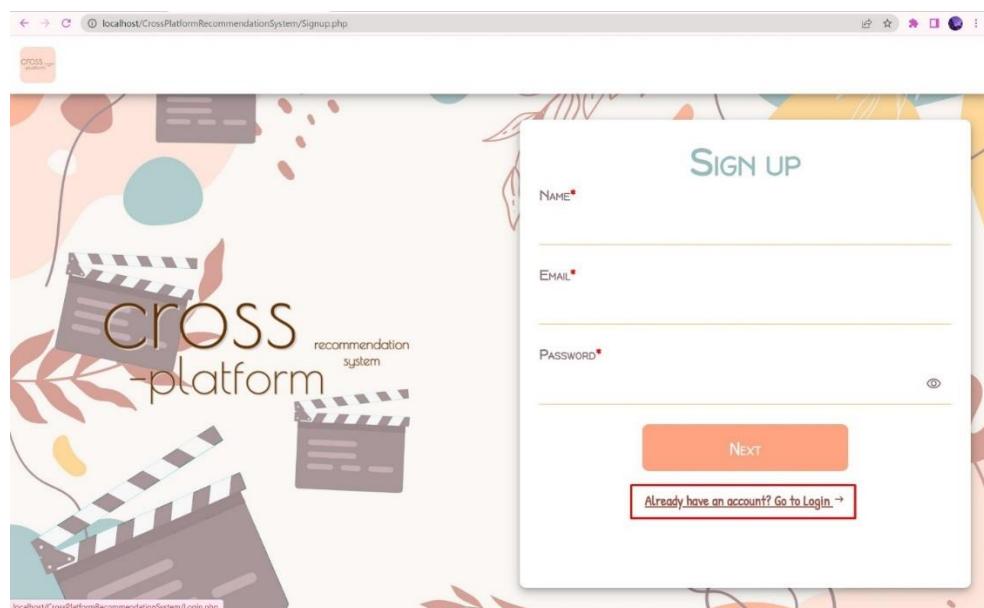


**Table 72**

<b>Test Case ID</b>	SU_05	<b>Status</b>	Pass
<b>Module Name</b>	Sign Up		
<b>Test Case Title</b>	Sign Up – Navigate to Login		
<b>Test Case Description</b>	When the user clicks on the ‘Already have an account? Go to Login →’ link, the user is navigated to the Login page.		
<b>Pre-conditions</b>	<b>Test Data</b>		
Access to a browser	-		
<b>Steps</b>	<ol style="list-style-type: none"> <li>1. The user navigates to the Signup page.</li> <li>2. The user clicks on the ‘Already have an account? Go to Login →’ link (Figure 41).</li> </ol>		
<b>Expected Result</b>	<b>Actual Result</b>		
The user is navigated to the Login page.	Same as the Expected Result		

**Figure 41**

*Navigating to Login from Sign-up page*

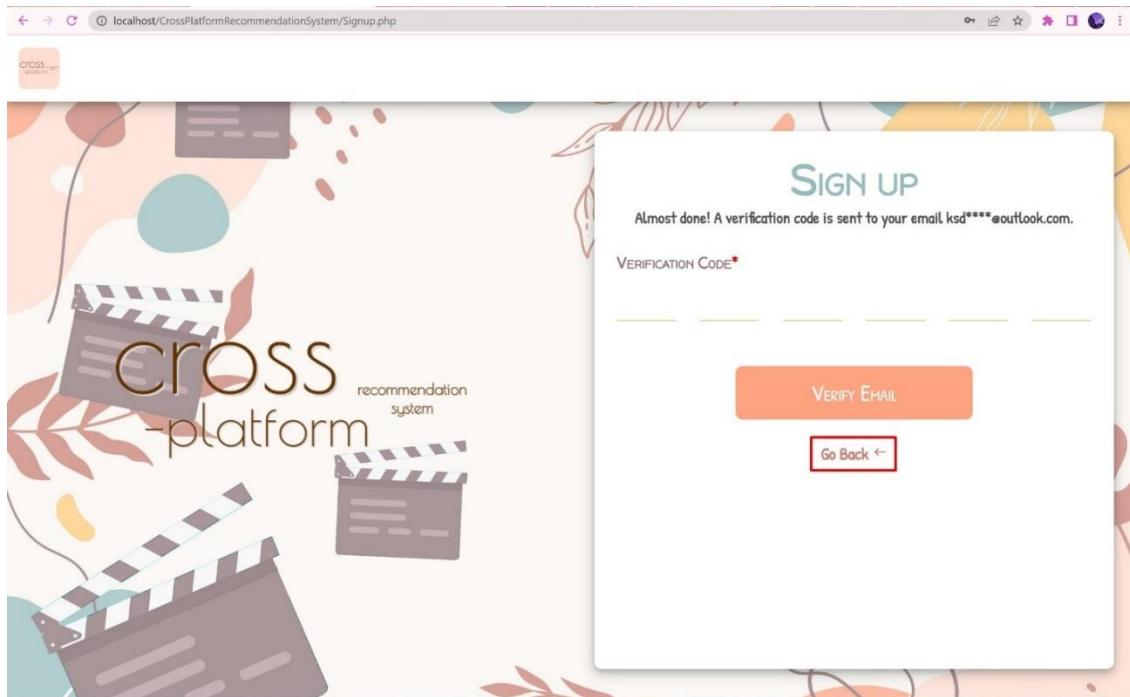


**Table 73**

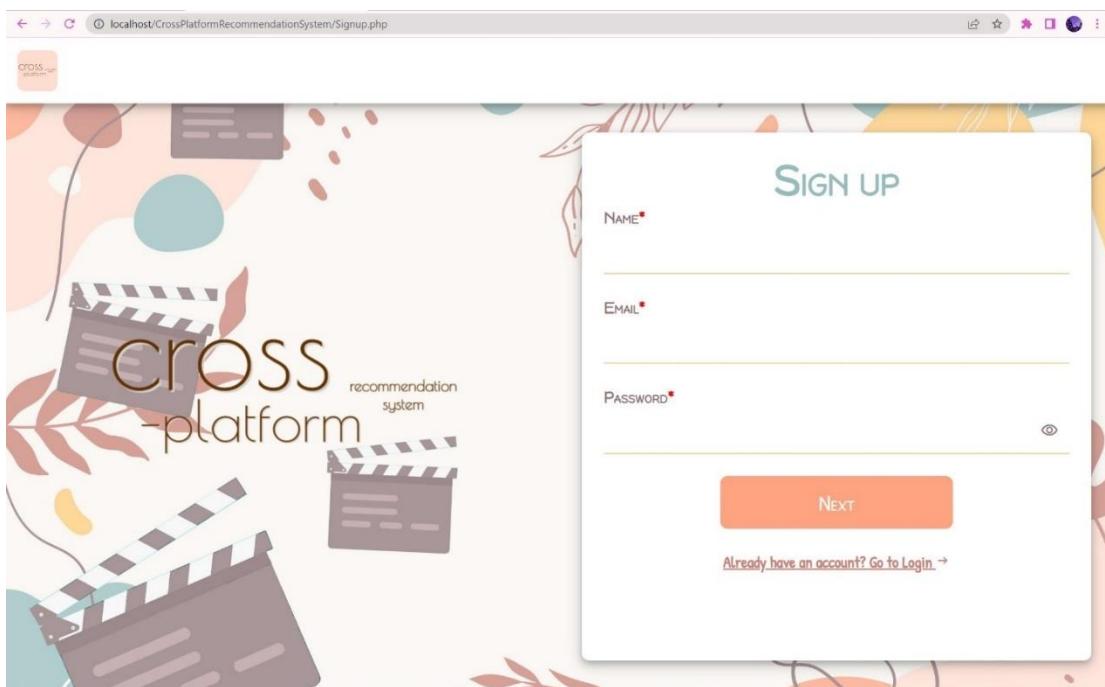
<b>Test Case ID</b>	SU_06	<b>Status</b>	Pass
<b>Module Name</b>	Sign Up		
<b>Test Case Title</b>	Sign Up – Go to Input User Credentials		
<b>Test Case Description</b>	When the user clicks on the ‘Go Back’ link on the verification page of Sign-up, the user will be navigated to the Sign-up page 1.		
<b>Pre-conditions</b>	<ul style="list-style-type: none"> <li>• Access to a browser</li> <li>• The user has already entered his name, email, and password and is at the verification code page.</li> </ul>		
<b>Steps</b>	1. Click on the ‘Go Back’ link (Figure 42).		
<b>Expected Result</b>	<b>Actual Result</b>		
The user will be navigated to the Sign-up page 1 (Figure 43).	Same as the Expected Result		

**Figure 42**

*Go Back link*

**Figure 43**

*User navigated to the Sign-up page 1 after clicking the Go Back link*

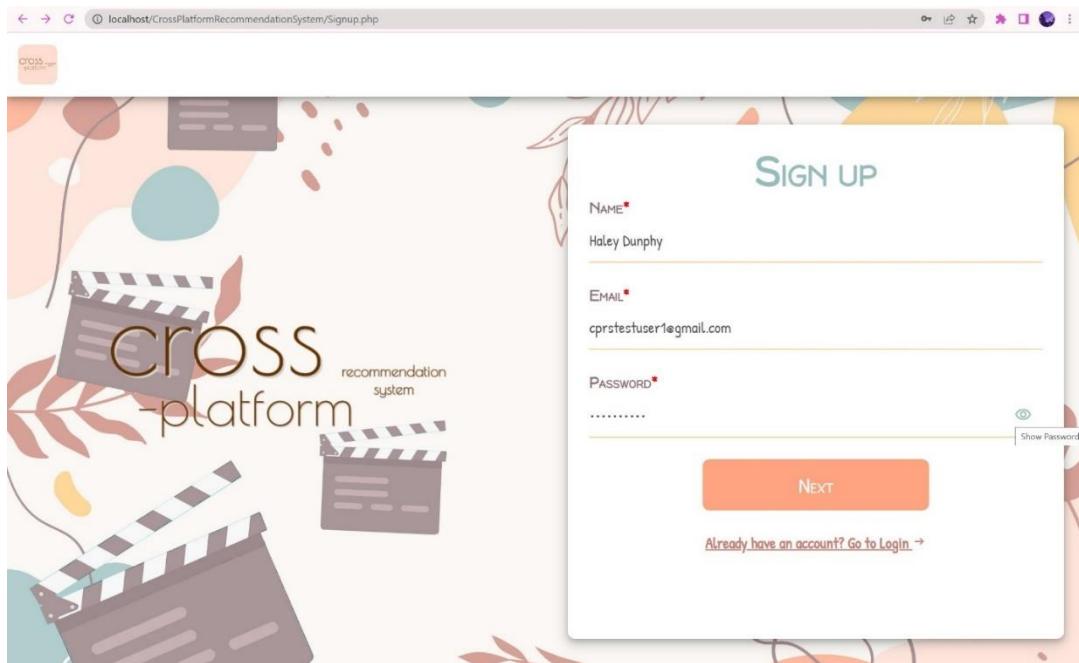


**Table 74**

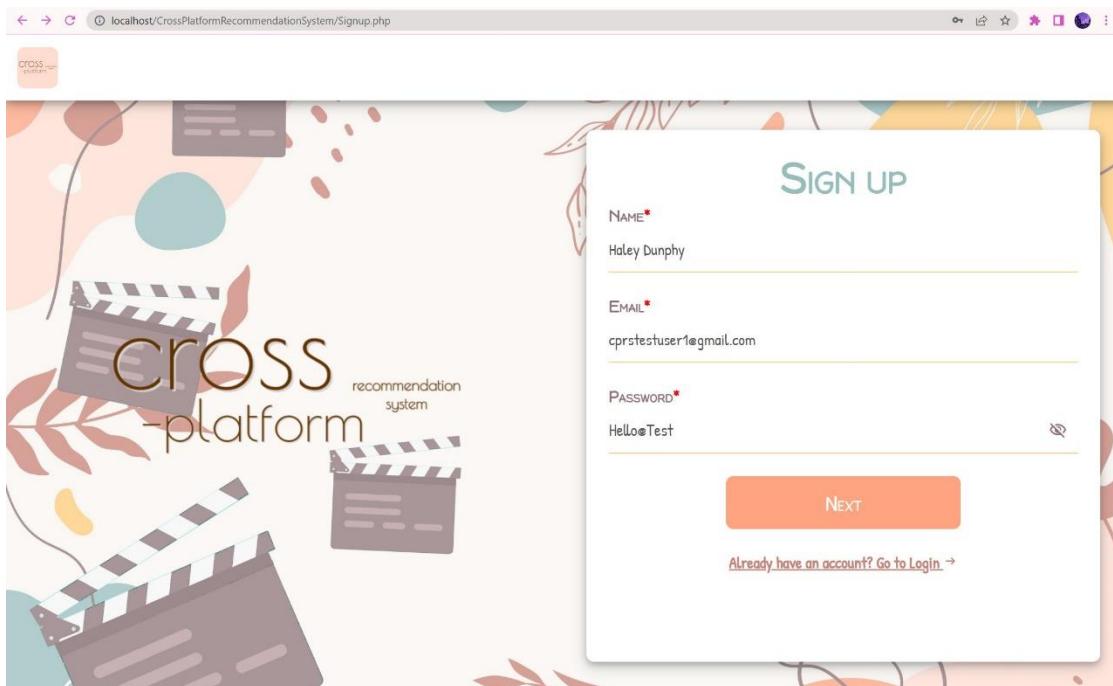
<b>Test Case ID</b>	SU_07	<b>Status</b>	Pass
<b>Module Name</b>	Sign Up		
<b>Test Case Title</b>	Sign Up – Toggle Show Password		
<b>Test Case Description</b>	When the user clicks on the Show/Hide Password button, the password should be shown or hidden.		
<b>Pre-conditions</b>	<b>Test Data</b>		
• Access to a browser	name = Haley Dunphy email = cprstestuser1@gmail.com password = Hello@Test		
<b>Steps</b>	1. Navigate to the Sign Up page. 2. Enter the name, email, and password. 3. Click on the Show Password button (Figure 44). 4. Click on the Hide Password button (Figure 45).		
<b>Expected Result</b>	<b>Actual Result</b>		
When the Show Password is clicked in step 3, the password is visible to the user (Figure 46).  When the Hide Password is clicked in step 4, the password is masked (Figure 47).	Same as the Expected Result		

**Figure 44**

*Show Password button*

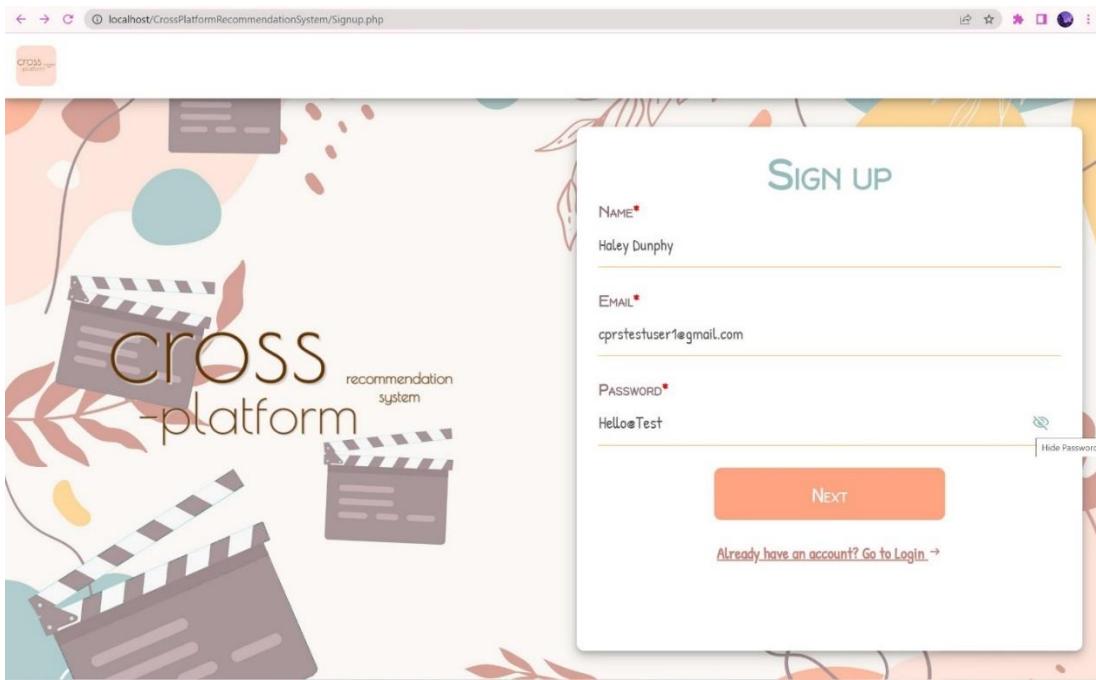
**Figure 45**

*Password is visible after clicking on the Show password button*

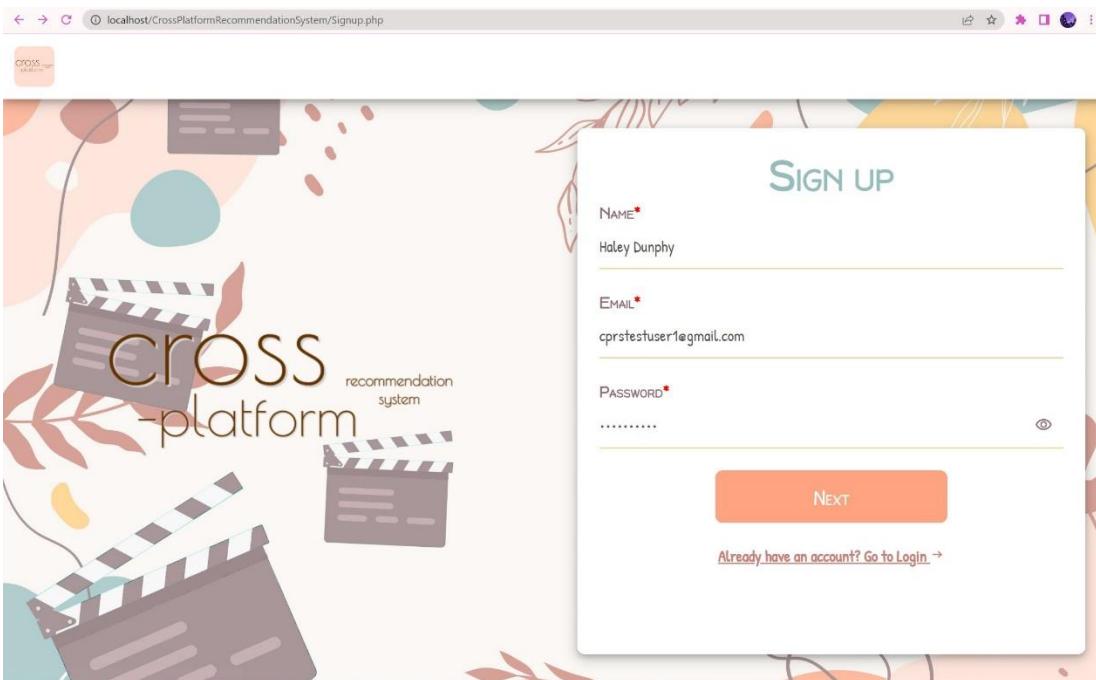


**Figure 46**

*Hide Password button*

**Figure 47**

*Password is masked after clicking on the Hide password button*

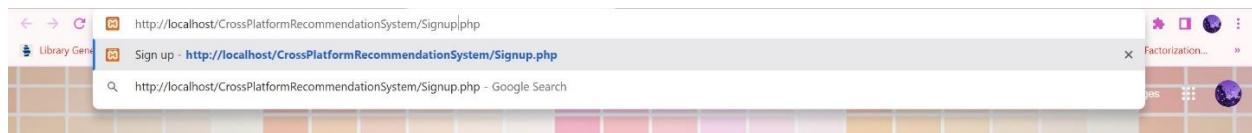


**Table 75**

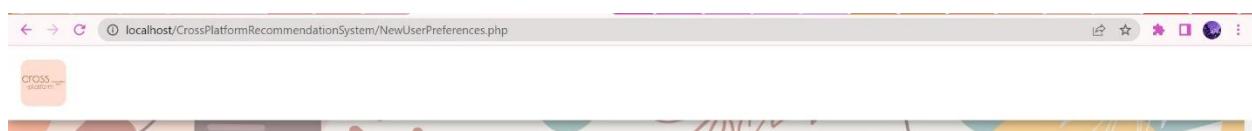
<b>Test Case ID</b>	SU_08	<b>Status</b>	Pass
<b>Module Name</b>	Sign Up		
<b>Test Case Title</b>	Sign Up – New User Already Logged In		
<b>Test Case Description</b>	When the new user is already logged in but navigates to the Sign Up page, the user is redirected to the NewUserPreferences page.		
<b>Pre-conditions</b>	<b>Test Data</b>		
<ul style="list-style-type: none"> <li>• Access to a browser</li> <li>• The user is a new user, i.e., he hasn't set his preferences yet and is logged in.</li> </ul>	email = cprstestuser1@gmail.com password = Hello@Test		
<b>Steps</b>	1. Navigate to the Sign Up page (Figure 48).		
<b>Expected Result</b>	<b>Actual Result</b>		
The user should be navigated to the NewUserPreferences page (Figure 49).	Same as the Expected Result		

**Figure 48**

New user trying to navigate to the Sign-up page

**Figure 49**

New user is navigated to the NewUserPreferences page



**Table 76**

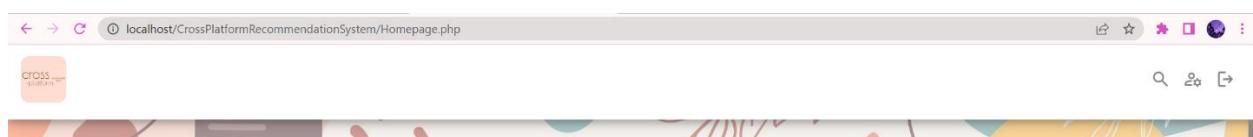
<b>Test Case ID</b>	SU_09	<b>Status</b>	Pass
<b>Module Name</b>	Sign Up		
<b>Test Case Title</b>	Sign Up – Existing User Already Logged In		
<b>Test Case Description</b>	When the existing user is already logged in but navigates to the Sign Up page, the user is redirected to the Homepage.		
<b>Pre-conditions</b>	<b>Test Data</b>		
<ul style="list-style-type: none"> <li>• Access to a browser</li> <li>• The user is an existing user, i.e., he has set his preferences yet and is logged in.</li> </ul>	email = cprstestuser2@gmail.com password = Hello@Test2		
<b>Steps</b>	1. Navigate to the Sign Up page (Figure 50).		
<b>Expected Result</b>	<b>Actual Result</b>		
The user should be navigated to the Homepage page (Figure 51).	Same as the Expected Result		

**Figure 50**

*Existing user trying to navigate to the Sign-up page*

**Figure 51**

*Existing user is navigated to the Homepage page*

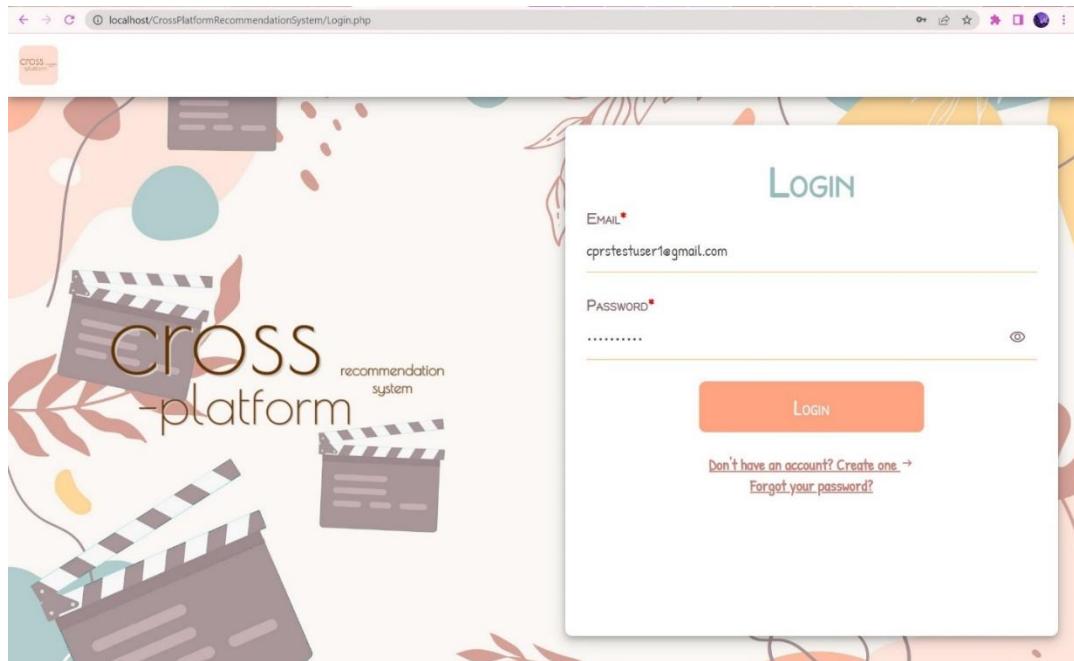


**Table 77**

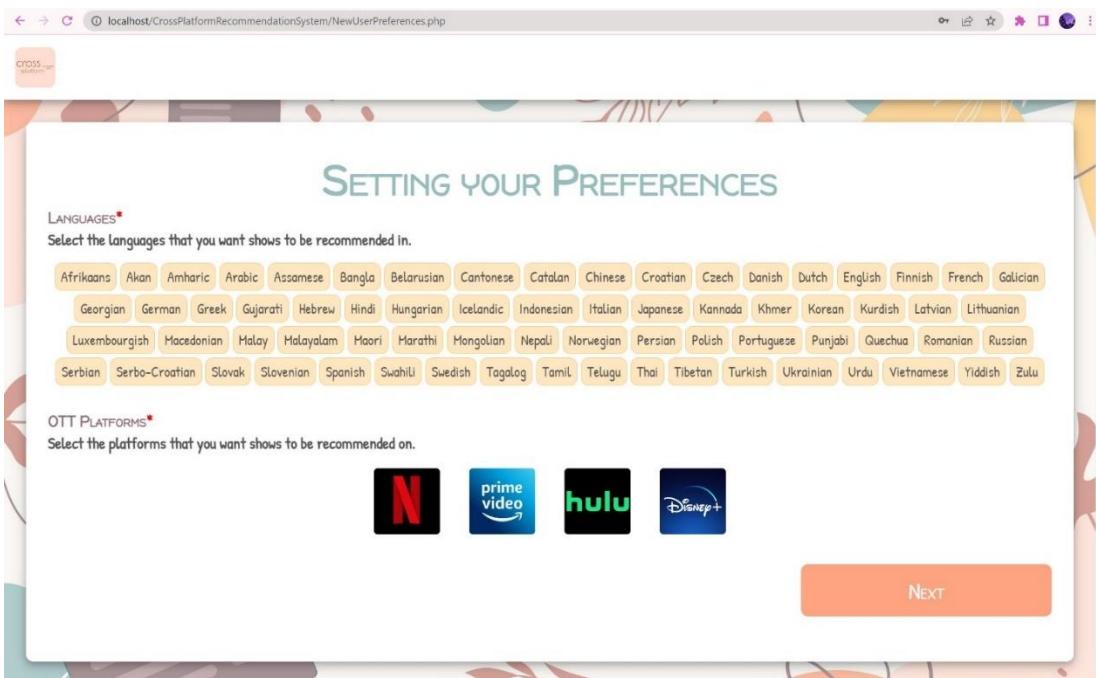
<b>Test Case ID</b>	LG_01	<b>Status</b>	Pass
<b>Module Name</b>	Login		
<b>Test Case Title</b>	User Login – Valid, New User		
<b>Test Case Description</b>	When the new user enters valid credentials, the user can log in.		
<b>Pre-conditions</b>	<p><b>Test Data</b></p> <ul style="list-style-type: none"> <li>• Access to a browser</li> <li>• The user is a new user, i.e., he hasn't set his preferences yet.</li> </ul>		
<b>Steps</b>	<ol style="list-style-type: none"> <li>1. Navigate to the Login page.</li> <li>2. Enter the email and password (Figure 52).</li> <li>3. Click on the Login button.</li> </ol>		
<b>Expected Result</b>	<b>Actual Result</b>		
The user is logged in and is directed to NewUserPreferences page (Figure 53).	Same as the Expected Result		

**Figure 52**

*Login page with new user-entered credentials*

**Figure 53**

*New user directed to NewUserPreferences page*

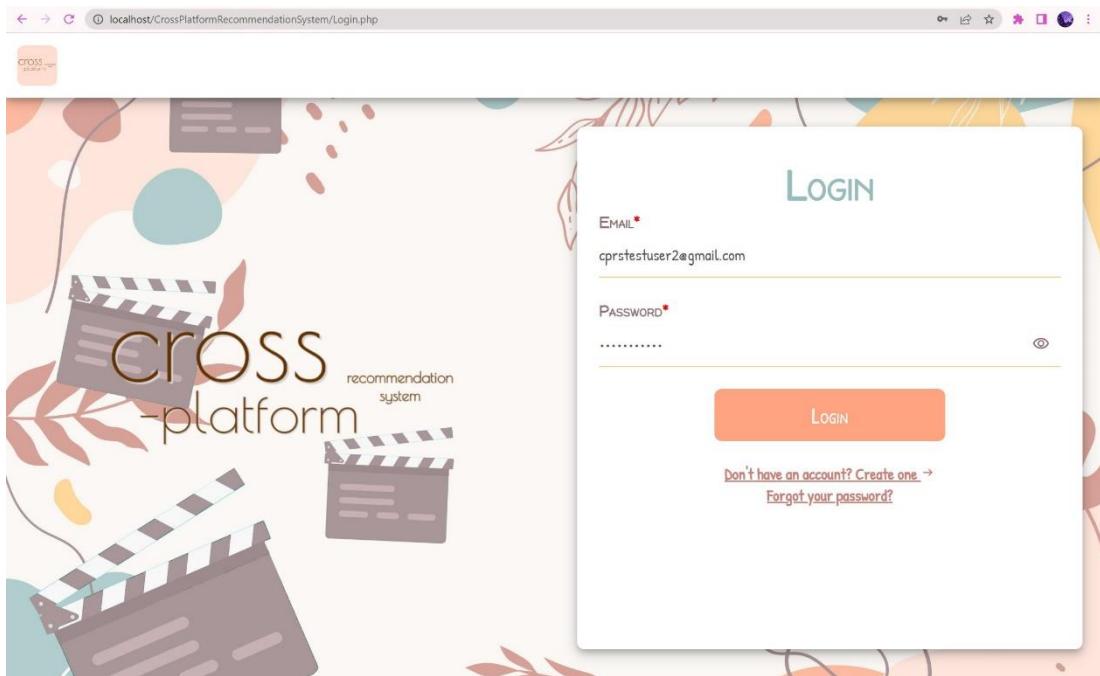


**Table 78**

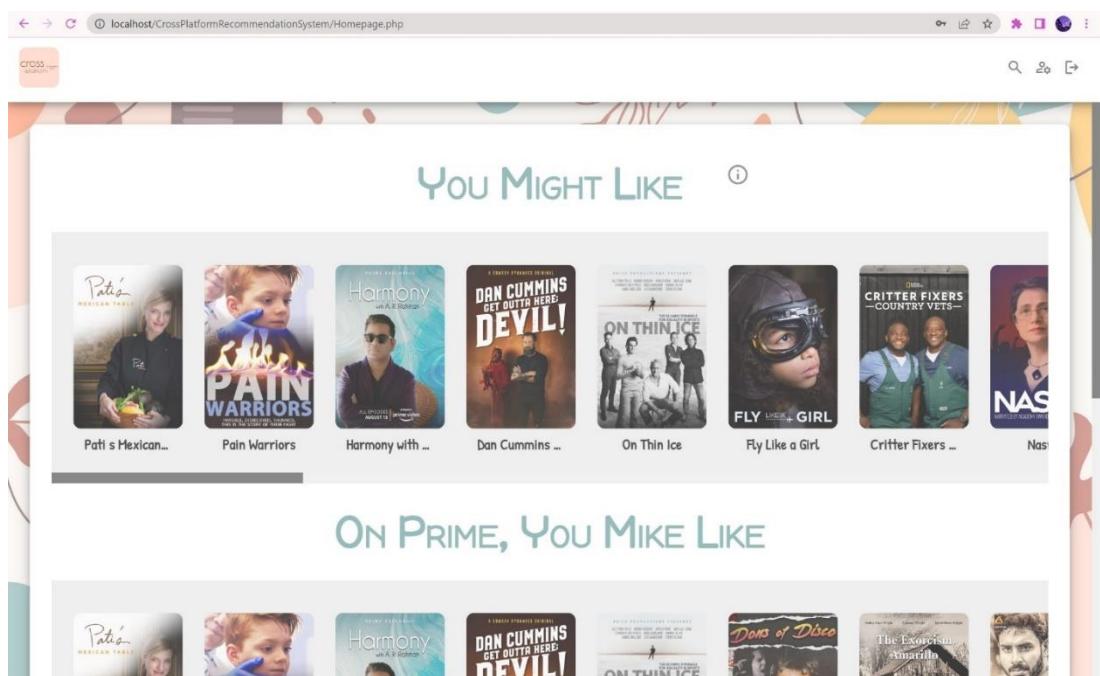
<b>Test Case ID</b>	LG_02	<b>Status</b>	Pass
<b>Module Name</b>	Login		
<b>Test Case Title</b>	User Login – Valid, Existing User		
<b>Test Case Description</b>	When the existing user enters valid credentials, the user can log in.		
<b>Pre-conditions</b>	<p><b>Test Data</b></p> <ul style="list-style-type: none"> <li>• Access to a browser</li> <li>• The user is an existing user, i.e., he has set his preferences.</li> </ul>		
<b>Steps</b>	<ol style="list-style-type: none"> <li>1. Navigate to the Login page.</li> <li>2. Enter the email and password (Figure 54).</li> <li>3. Click on the Login button.</li> </ol>		
<b>Expected Result</b>	<b>Actual Result</b>		
The user is logged in and is directed to Homepage page (Figure 55).	Same as the Expected Result		

**Figure 54**

*Login page with existing user-entered credentials*

**Figure 55**

*Existing user directed to Homepage page*

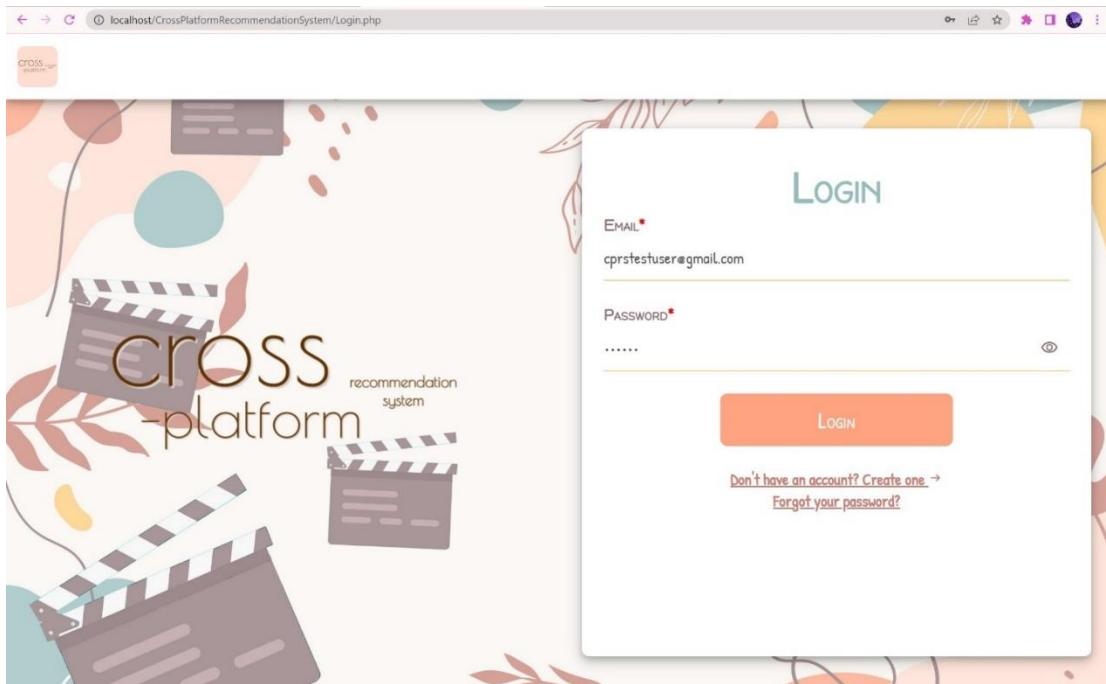


**Table 79**

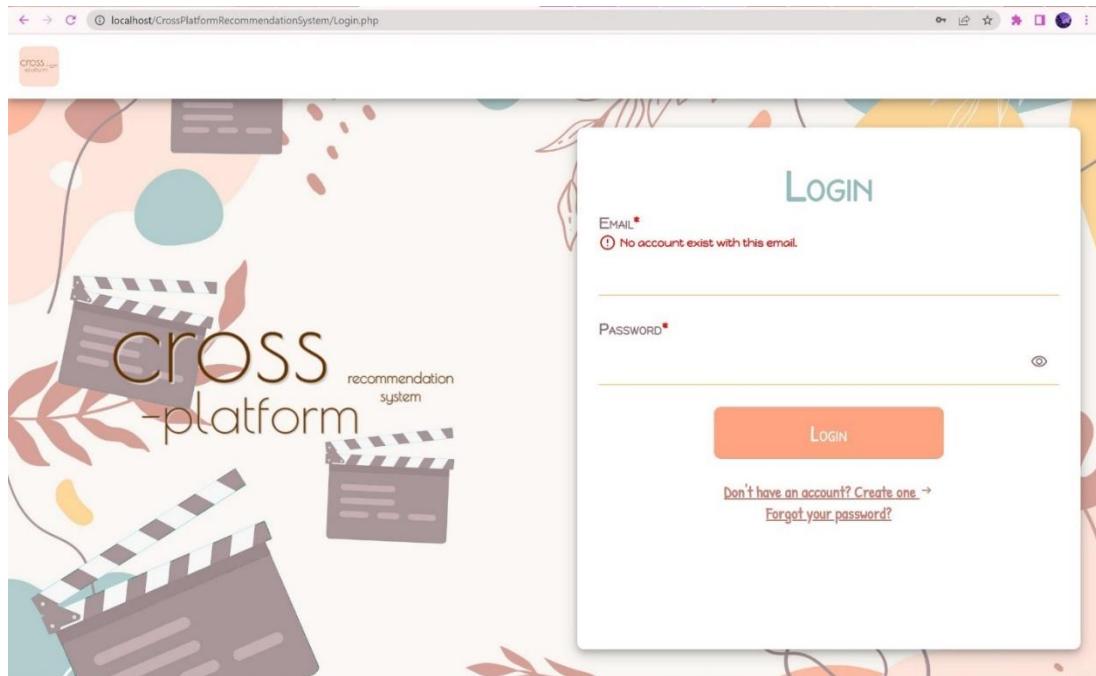
<b>Test Case ID</b>	LG_03	<b>Status</b>	Pass
<b>Module Name</b>	Login		
<b>Test Case Title</b>	User Login – Invalid User Credentials		
<b>Test Case Description</b>	When the user enters an invalid email or password, the user should see the error message.		
<b>Pre-conditions</b>	<b>Test Data</b>		
Access to a browser	email = cprstestuser@gmail.com password = Helloo		
<b>Steps</b>	<ol style="list-style-type: none"> <li>1. Navigate to the Login page.</li> <li>2. Enter the email and password (Figure 56).</li> <li>3. Click on the Login button.</li> </ol>		
<b>Expected Result</b>	<b>Actual Result</b>		
Since the user entered an invalid email, the user should see ‘No account exists with this email’ message (Figure 57).	Same as the Expected Result		

**Figure 56**

*Invalid credentials entered by the user*

**Figure 57**

*Error displayed for invalid email entered by the user*

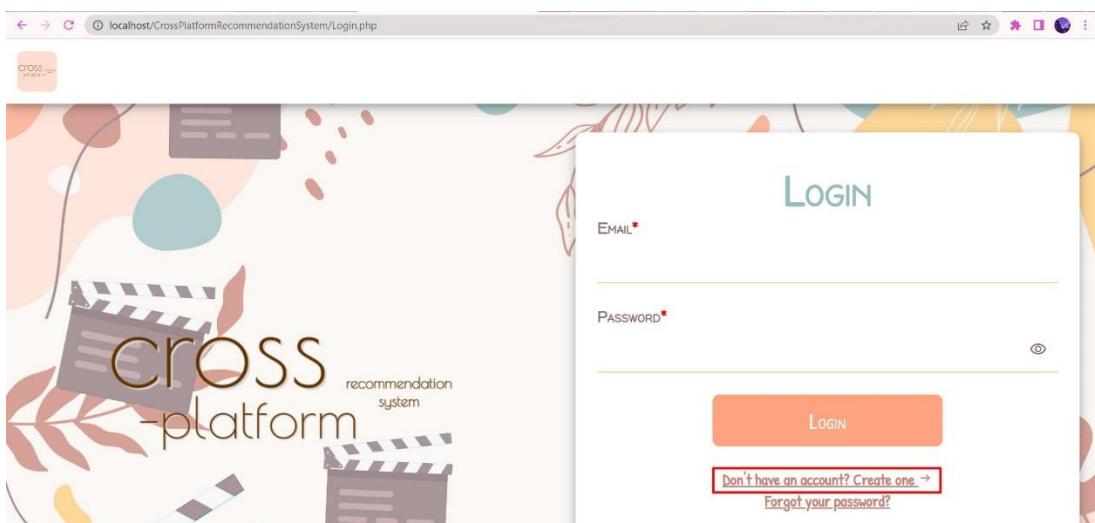


**Table 80**

<b>Test Case ID</b>	LG_04	<b>Status</b>	Pass
<b>Module Name</b>	Login		
<b>Test Case Title</b>	User Login – Navigate to Sign Up		
<b>Test Case Description</b>	When the user clicks on the “Don’t have an account? Create one → ” link, the user should be navigated to the Sign-Up page.		
<b>Pre-conditions</b>	<b>Test Data</b>		
Access to a browser	-		
<b>Steps</b>	<ol style="list-style-type: none"> <li>1. Navigate to the Login page.</li> <li>2. Click on the “Don’t have an account? Create one → ” link (Figure 58).</li> </ol>		
<b>Expected Result</b>	<b>Actual Result</b>		
The user should be navigated to the Sign Up page.	Same as the Expected Result		

**Figure 58**

“Don’t have an account? Create one” link on the Login page.

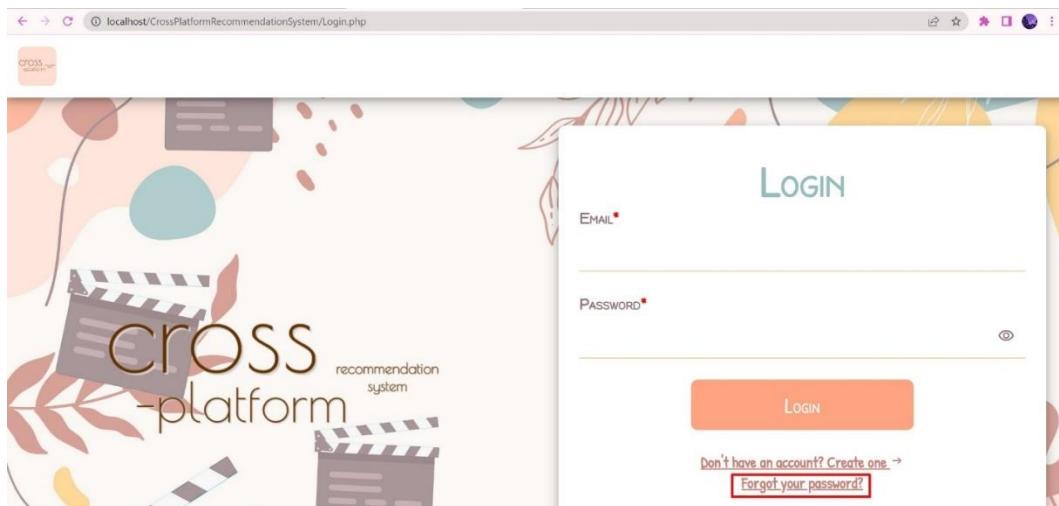


**Table 81**

<b>Test Case ID</b>	LG_05	<b>Status</b>	Pass
<b>Module Name</b>	Login		
<b>Test Case Title</b>	User Login – Navigate to Forgot Password		
<b>Test Case Description</b>	When the user clicks on the “Forgot your password?” link, the user should be navigated to the Forgot Password page.		
<b>Pre-conditions</b>	<b>Test Data</b>		
Access to a browser	-		
<b>Steps</b>	<ol style="list-style-type: none"> <li>1. Navigate to the Login page.</li> <li>2. Click on the “Forgot your password” link (Figure 59).</li> </ol>		
<b>Expected Result</b>	<b>Actual Result</b>		
The user should be navigated to the Forgot Password page.	Same as the Expected Result		

**Figure 59**

‘Forgot your password’ link on the Login page



**Table 82**

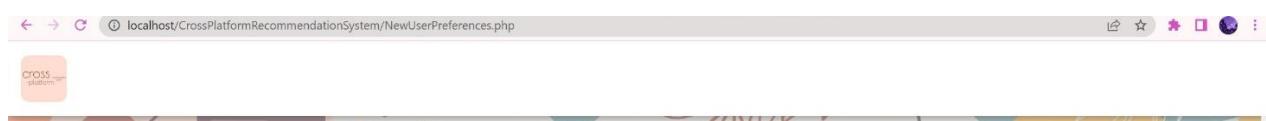
<b>Test Case ID</b>	LG_06	<b>Status</b>	Pass
<b>Module Name</b>	Login		
<b>Test Case Title</b>	User Login – New User Already Logged In		
<b>Test Case Description</b>	When the new user is already logged in but navigates to the Login page, the user is redirected to the NewUserPreferences page.		
<b>Pre-conditions</b>	<b>Test Data</b>		
<ul style="list-style-type: none"> <li>Access to a browser</li> <li>The user is a new user, i.e., he hasn't set his preferences yet and is logged in.</li> </ul>	email = cprstestuser1@gmail.com password = Hello@Test		
<b>Steps</b>	1. Navigate to the Login page (Figure 60).		
<b>Expected Result</b>	<b>Actual Result</b>		
The user should be navigated to the NewUserPreferences page (Figure 61).	Same as the Expected Result		

**Figure 60**

*New user trying to navigate to the Login page*

**Figure 61**

*New user is navigated to the NewUserPreferences page*



**Table 83**

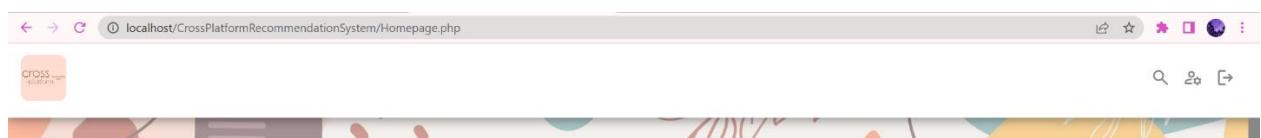
<b>Test Case ID</b>	LG_07	<b>Status</b>	Pass
<b>Module Name</b>	Login		
<b>Test Case Title</b>	User Login – Existing User Already Logged In		
<b>Test Case Description</b>	When the existing user is already logged in but navigates to the Login page, the user is redirected to the Homepage.		
<b>Pre-conditions</b>	<b>Test Data</b>		
<ul style="list-style-type: none"> <li>• Access to a browser</li> <li>• The user is an existing user, i.e., he has set his preferences yet and is logged in.</li> </ul>	-		
<b>Steps</b>	1. Navigate to the Login page (Figure 62).		
<b>Expected Result</b>	<b>Actual Result</b>		
The user should be navigated to the Homepage page (Figure 63).	Same as the Expected Result		

**Figure 62**

*Existing user trying to navigate to the Login page*

**Figure 63**

*Existing user is navigated to the Homepage page*

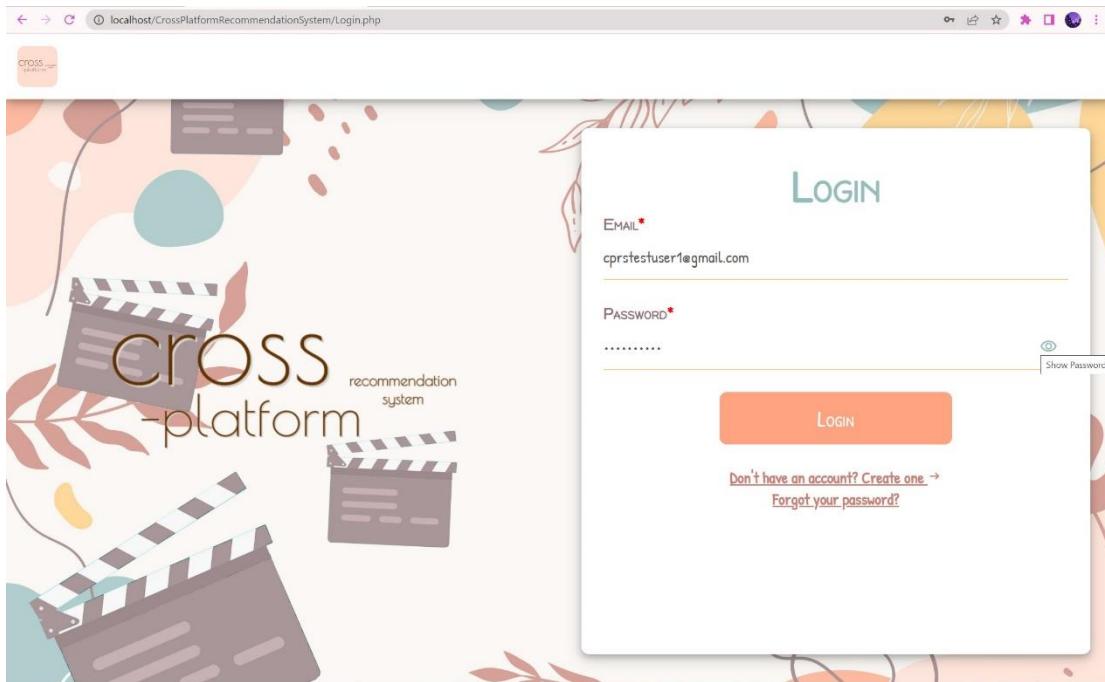


**Table 84**

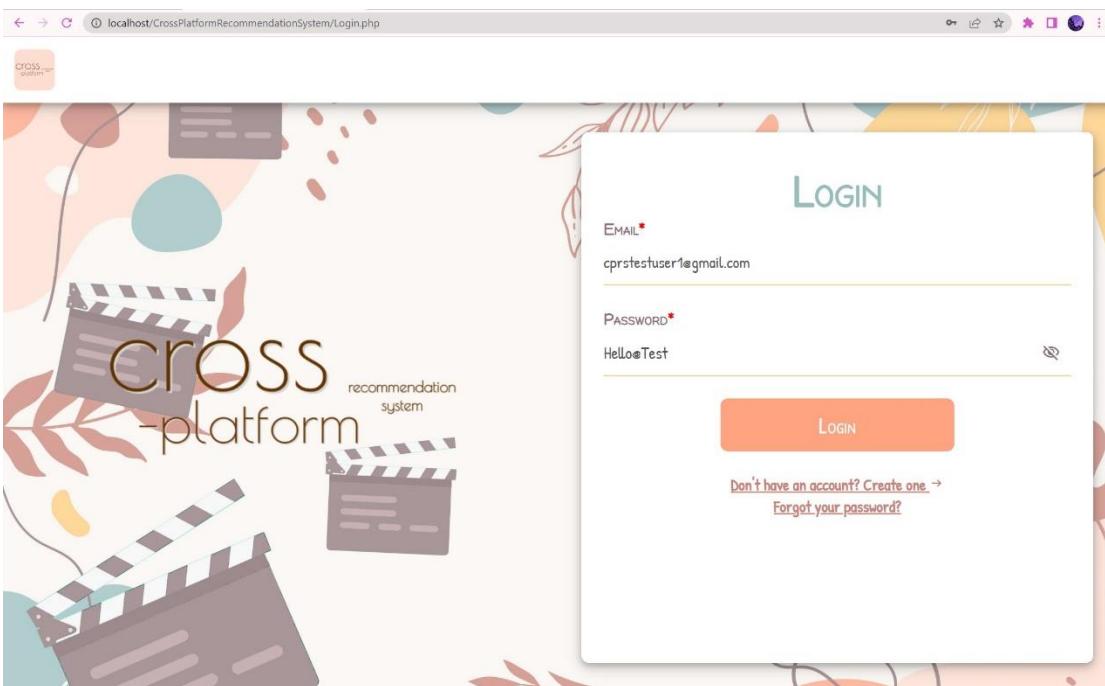
<b>Test Case ID</b>	LG_08	<b>Status</b>	Pass
<b>Module Name</b>	Login		
<b>Test Case Title</b>	User Login – Toggle Show Password		
<b>Test Case Description</b>	When the user clicks on the Show/Hide Password button, the password should be shown or hidden.		
<b>Pre-conditions</b>	<b>Test Data</b>		
• Access to a browser	email = cprstestuser1@gmail.com password = Hello@Test		
<b>Steps</b>	<ol style="list-style-type: none"> <li>1. Navigate to the Login page.</li> <li>2. Enter the email and password.</li> <li>3. Click on the Show Password button (Figure 64).</li> <li>4. Again, click on the Hide Password button (Figure 65).</li> </ol>		
<b>Expected Result</b>	<b>Actual Result</b>		
When the Show Password is clicked in step 3, the password is visible to the user (Figure 66).  When the Hide Password is clicked in step 4, the password is masked (Figure 67).	Same as the Expected Result		

**Figure 64**

Show Password button

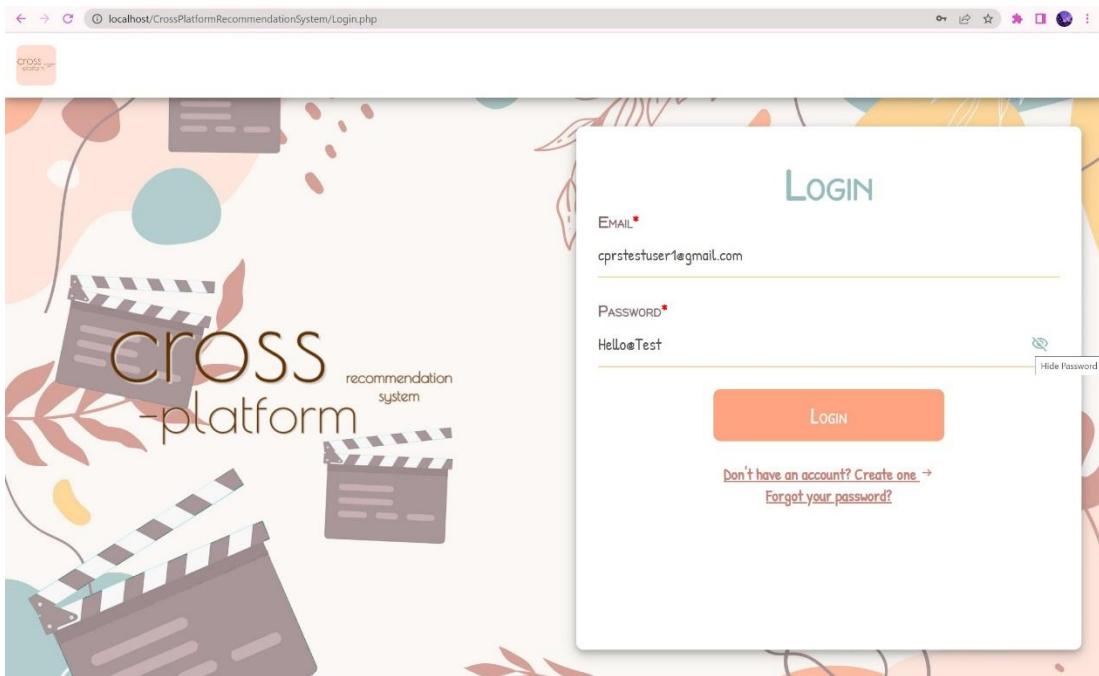
**Figure 65**

Password is visible after clicking on the Show password button

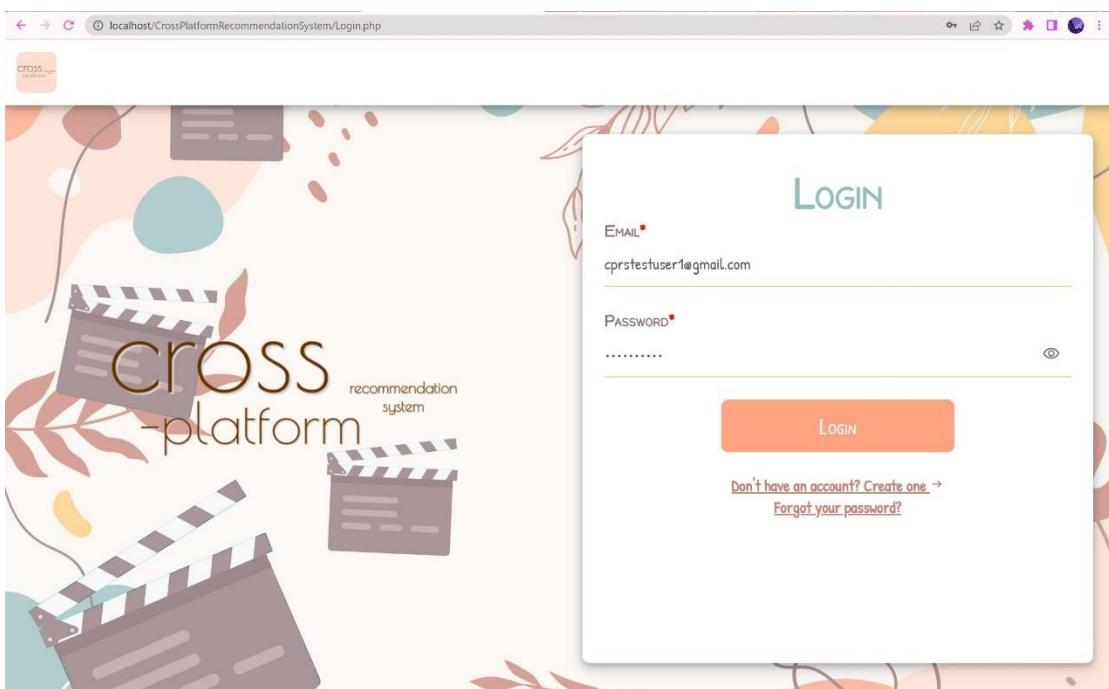


**Figure 66**

*Hide Password button*

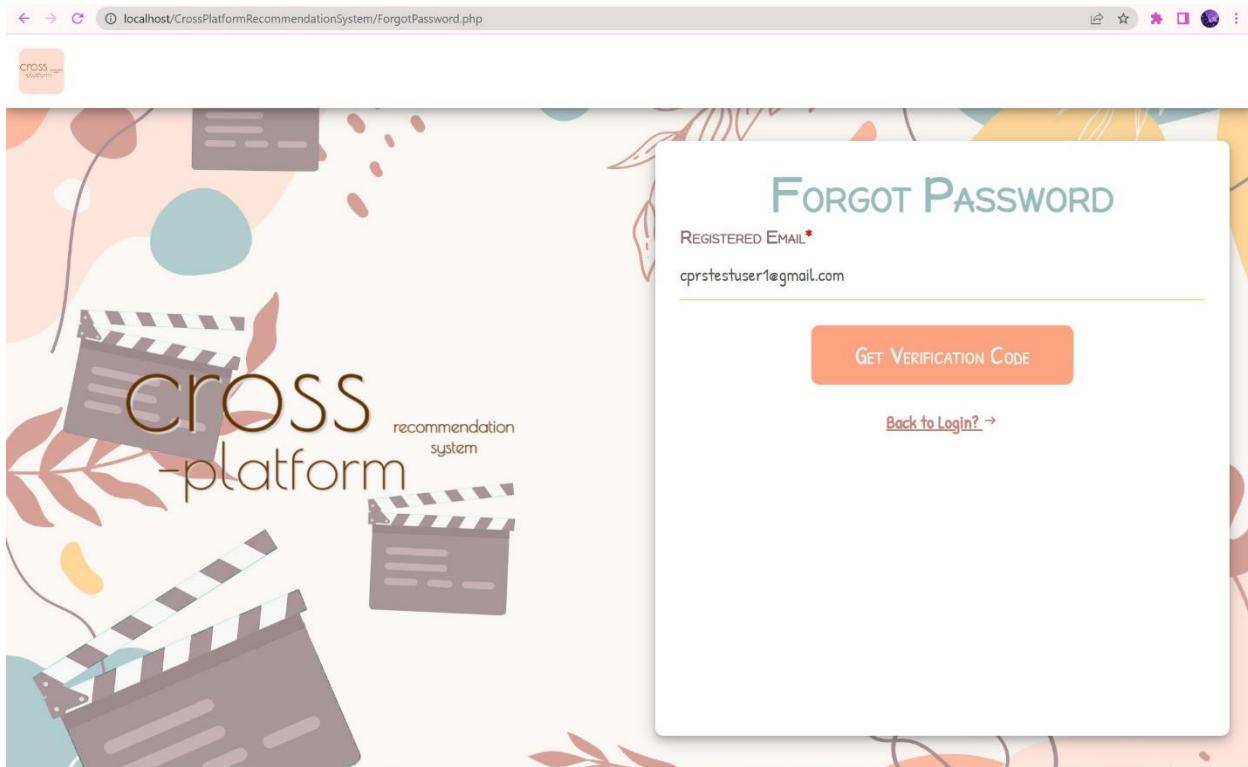
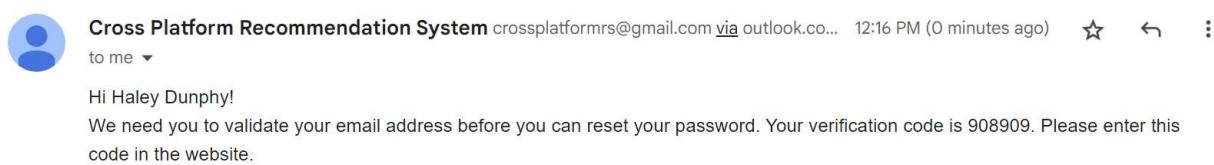
**Figure 67**

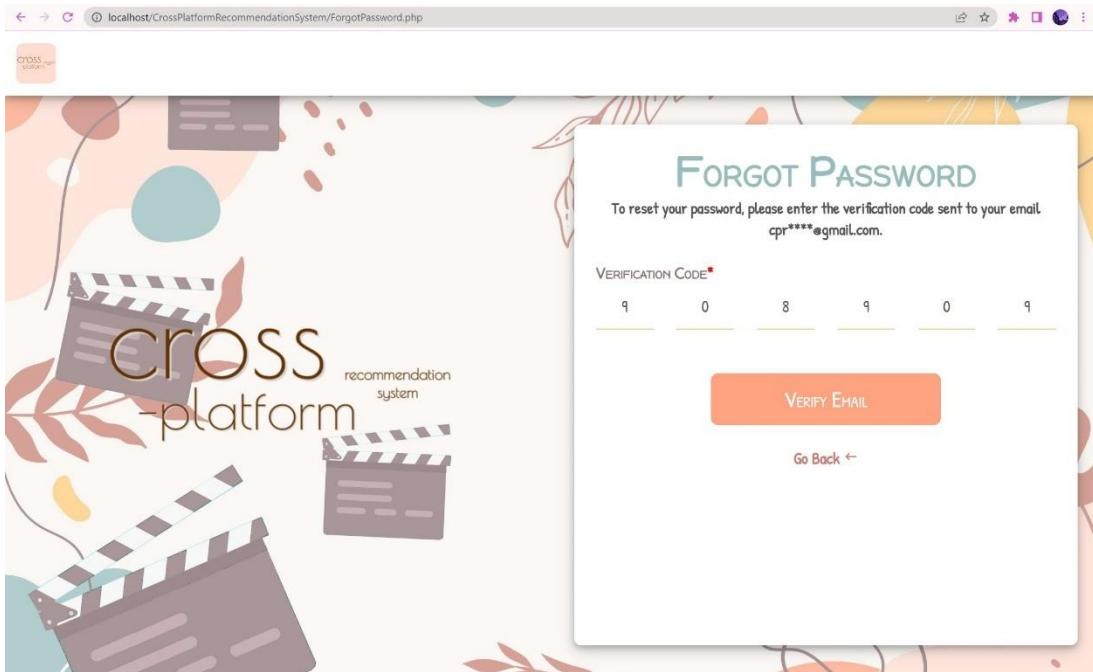
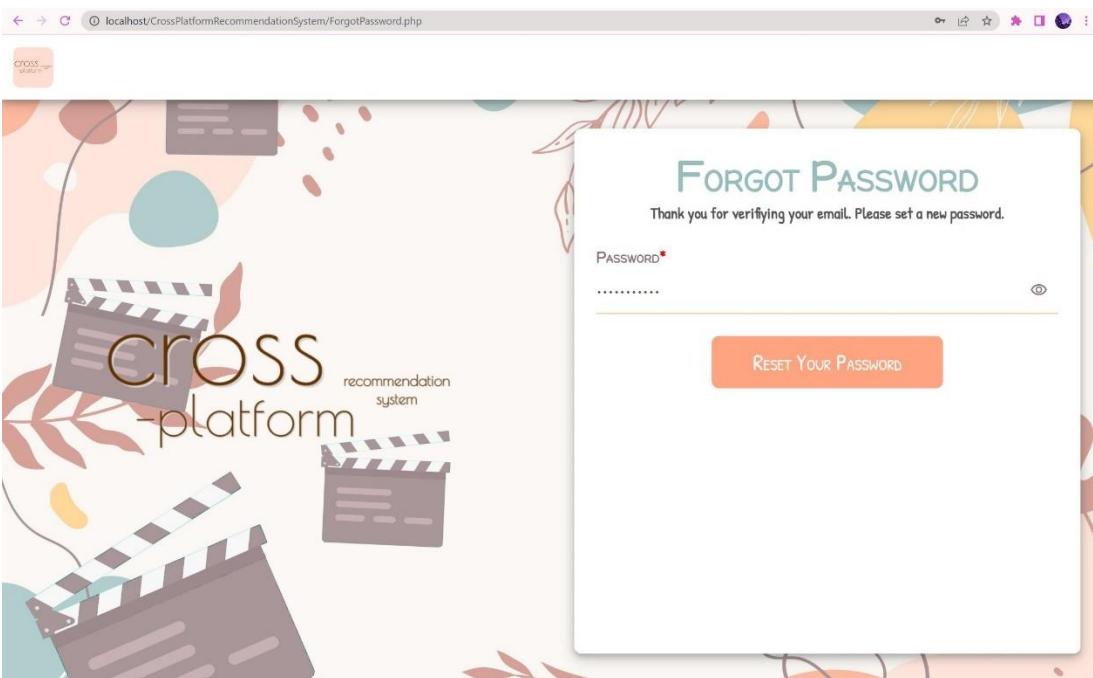
*Password is masked after clicking on the Hide password button*



**Table 85**

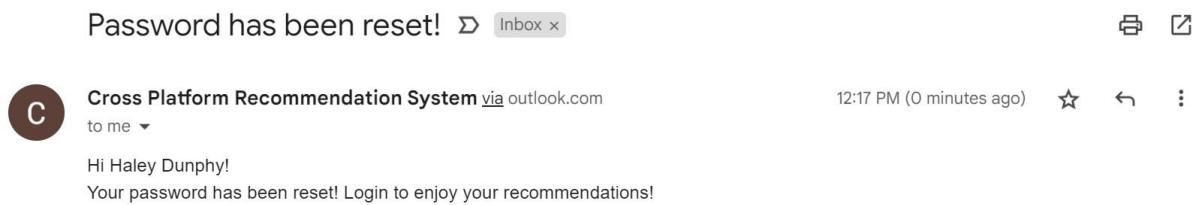
<b>Test Case ID</b>	FP_01	<b>Status</b>	Pass
<b>Module Name</b>	Forgot Password		
<b>Test Case Title</b>	Forgot Password		
<b>Test Case Description</b>	When the user navigates to the Forgot Password page, the user should be able to reset his password.		
<b>Pre-conditions</b>	<b>Test Data</b>		
Access to a browser	email = cprstestuser1@gmail.com		
<b>Steps</b>	<ol style="list-style-type: none"> <li>1. Navigate to the Forgot Password page.</li> <li>2. Enter the email (Figure 68).</li> <li>3. Click on the 'Get Verification Code' button.</li> <li>4. The system sends an email with a verification code to the entered email (Figure 69).</li> <li>5. Enter the verification code (Figure 70).</li> <li>6. Click on the Verify Email button.</li> <li>7. User enters the new password (Figure 71).</li> <li>8. Click on the Reset button.</li> </ol>		
<b>Expected Result</b>	<b>Actual Result</b>		
The user password is reset and should be navigated to the Login page.	Same as the Expected Result		
The user receives an email with confirmation (Figure 72).			

**Figure 68***Forgot Password page 1***Figure 69***Verification email sent to the user*

**Figure 70***Forgot Password page 2***Figure 71***Forgot Password page 3 – User enters the new password*

**Figure 72**

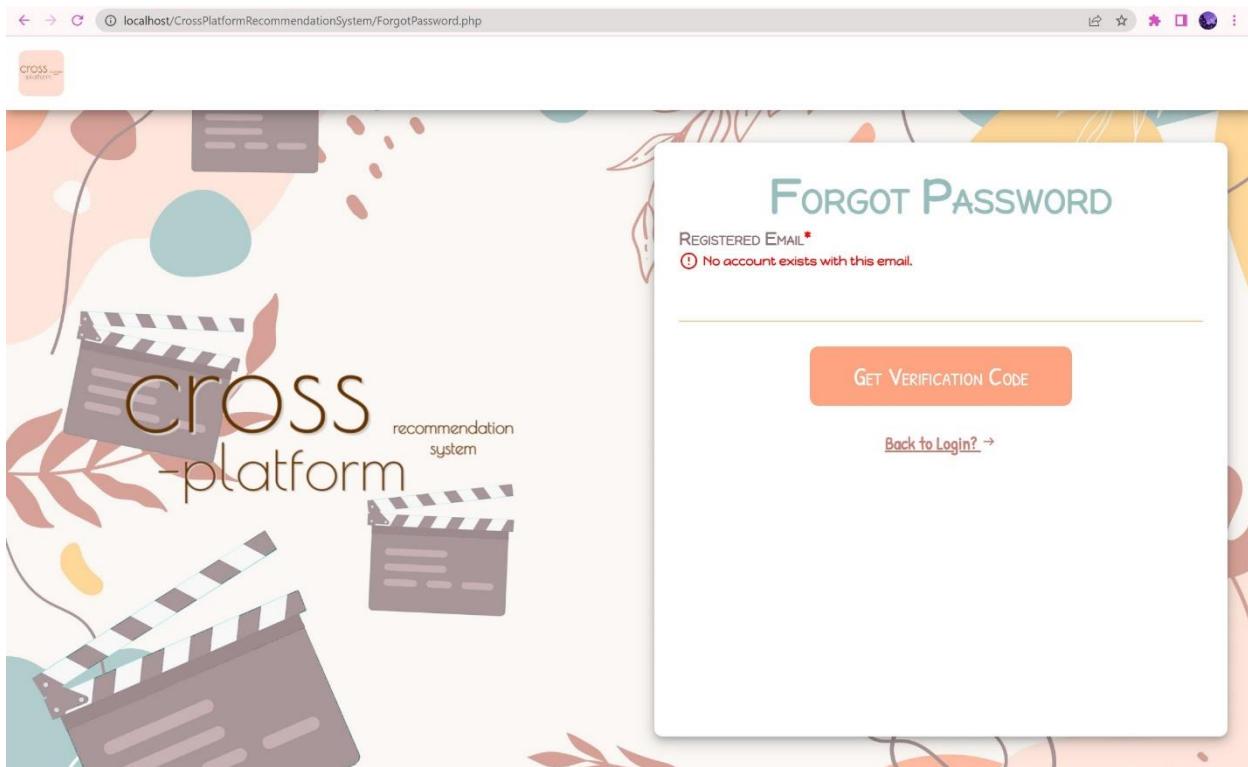
*Confirmation email sent to the user*

**Table 86**

<b>Test Case ID</b>	FP_02	<b>Status</b>	Pass
<b>Module Name</b>	Forgot Password		
<b>Test Case Title</b>	Forgot Password – Invalid Email		
<b>Test Case Description</b>	When the user provides an invalid email on the Forgot Password page, the user should see the appropriate error messages.		
<b>Pre-conditions</b>	<b>Test Data</b>		
Access to a browser	email = cprstestuser@gmail.com		
<b>Steps</b>	<ol style="list-style-type: none"> <li>1. Navigate to the Forgot Password page.</li> <li>2. Enter the email.</li> <li>3. Click on the 'Get Verification Code' button.</li> </ol>		
<b>Expected Result</b>	<b>Actual Result</b>		
Since the user enters an invalid email, he should see the appropriate error messages (Figure 73).	Same as the Expected Result		

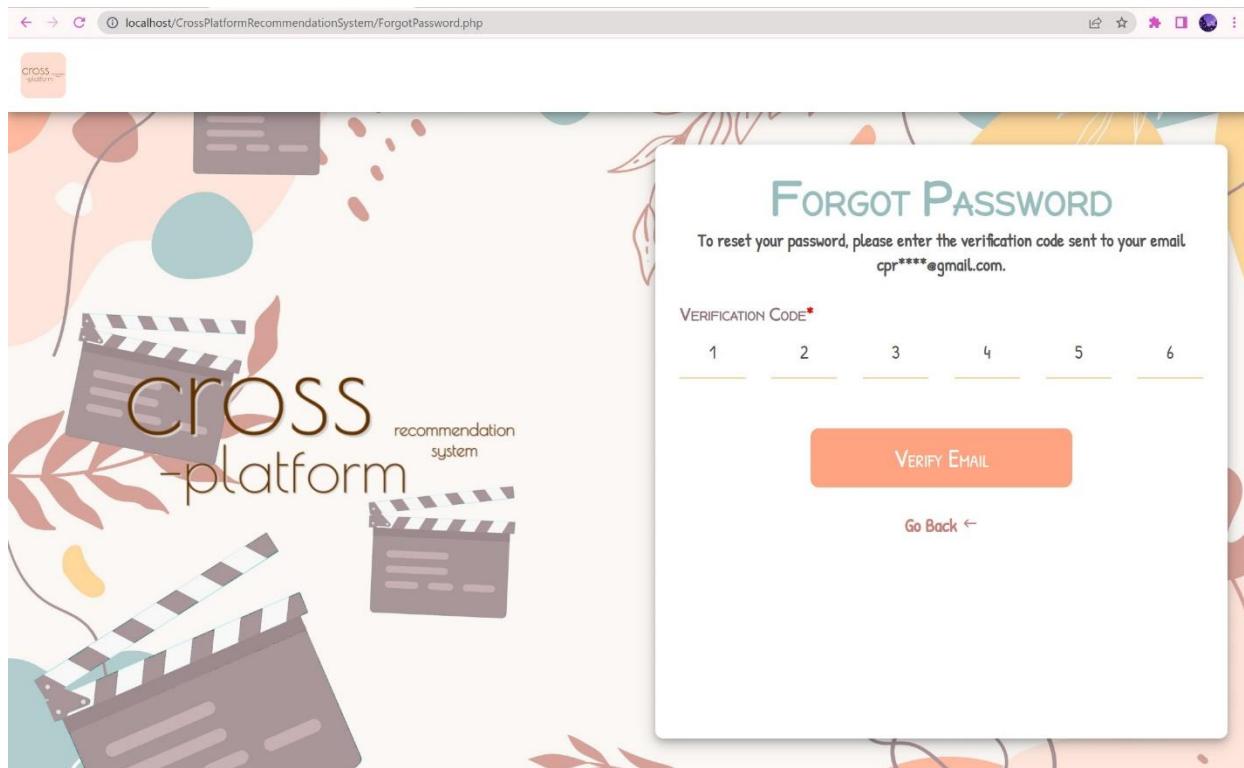
**Figure 73**

*Error displayed on the Forgot Password page when the user enters an invalid email*

**Table 87**

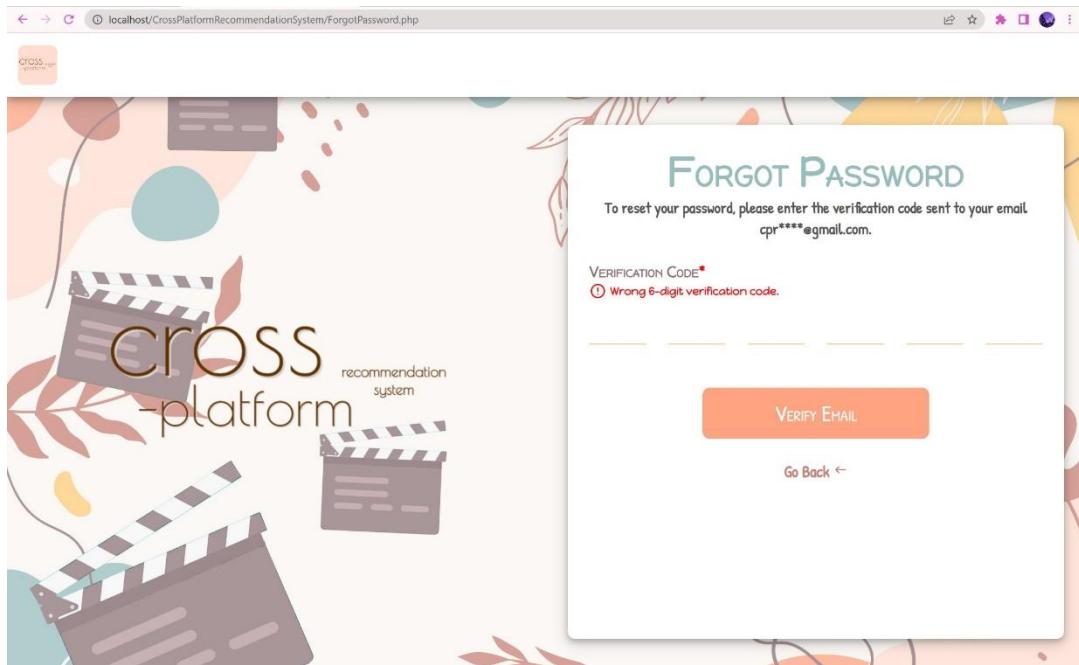
<b>Test Case ID</b>	FP_03	<b>Status</b>	Pass
<b>Module Name</b>	Forgot Password		
<b>Test Case Title</b>	Forgot Password – Invalid Verification Code		
<b>Test Case Description</b>	The user enters an invalid verification code on the Forgot Password page.		
<b>Pre-conditions</b>	<b>Test Data</b>		
<ul style="list-style-type: none"> <li>• Access to a browser</li> <li>• The user has already entered his email and is at the verification code page.</li> </ul>	email = cprstestuser1@gmail.com		

<b>Test Case ID</b>	FP_03	<b>Status</b>	Pass
<b>Steps</b>	1. Enter the verification code (Figure 74).  2. Click on the Verify Email button.		
<b>Expected Result</b>	<b>Actual Result</b>		
Since the user entered the wrong verification code, an error message 'Wrong 6-digit verification code' should appear to the user (Figure 75).	Same as the Expected Result		

**Figure 74***Incorrect verification code entered by the user*

**Figure 75**

*Error displayed when the user enters an incorrect verification code*

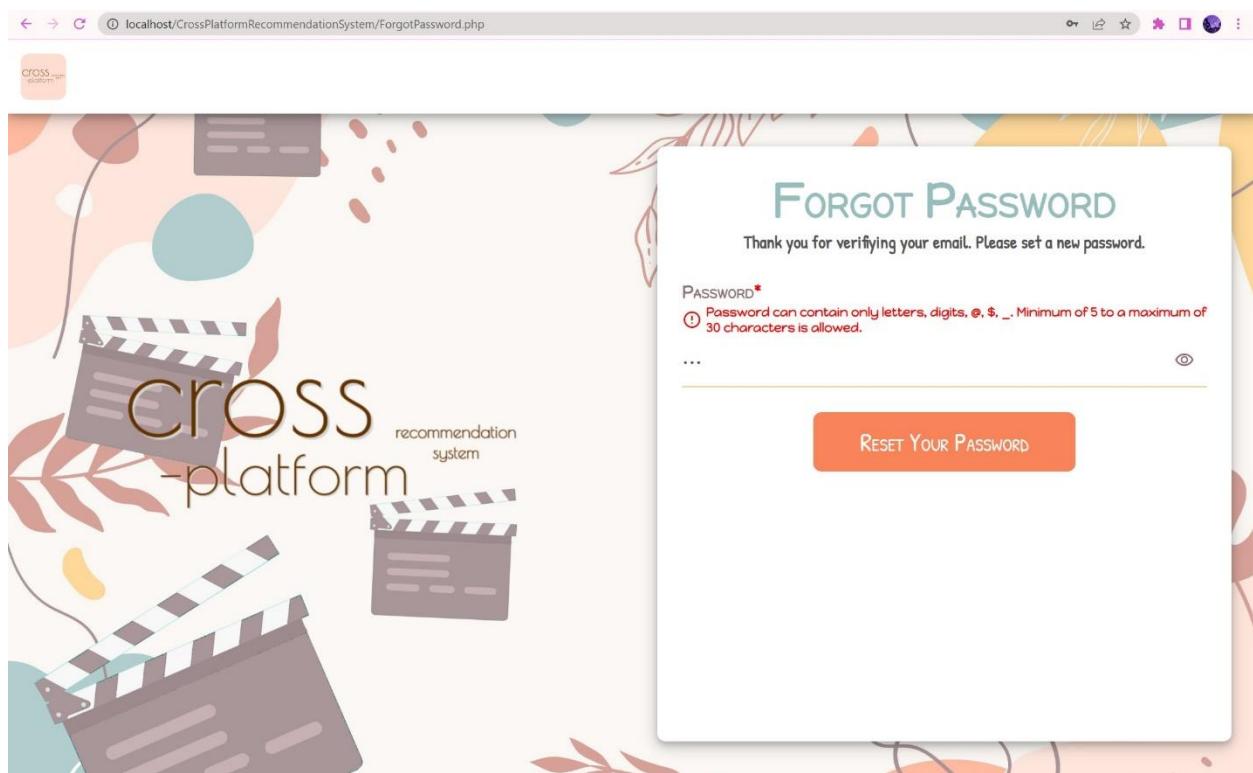
**Table 88**

<b>Test Case ID</b>	FP_04	<b>Status</b>	Pass
<b>Module Name</b>	Forgot Password		
<b>Test Case Title</b>	Forgot Password – Invalid New Password		
<b>Test Case Description</b>	The user enters an invalid new password on the Forgot Password page.		
<b>Pre-conditions</b>	<b>Test Data</b>		
<ul style="list-style-type: none"> <li>Access to a browser</li> <li>The user has already entered his email and verification code, and is at the new password page.</li> </ul>	email = cprstestuser1@gmail.com		

<b>Test Case ID</b>	FP_04	<b>Status</b>	Pass
<b>Steps</b>	3. Enter the new password. 4. Click on the 'Reset Your Password' button.		
<b>Expected Result</b>	<b>Actual Result</b>		
Since the user entered an invalid password, an error message should be displayed (Figure 76).	Same as the Expected Result		

**Figure 76**

*Error displayed when the user enters an invalid password*

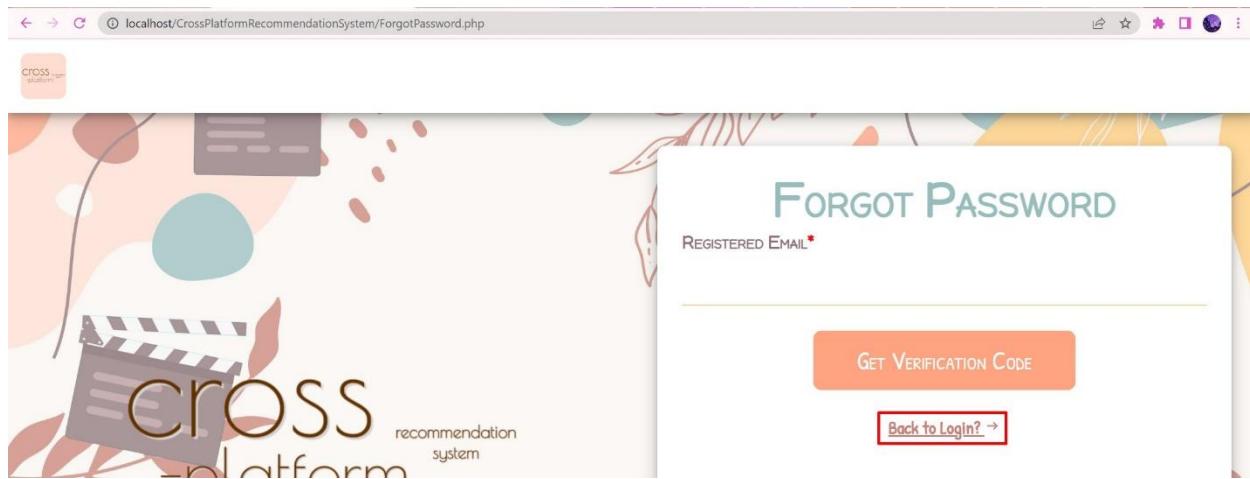


**Table 89**

<b>Test Case ID</b>	FP-05	<b>Status</b>	Pass
<b>Module Name</b>	Forgot Password		
<b>Test Case Title</b>	Forgot Password – Navigate to Login		
<b>Test Case Description</b>	When the user clicks on the ‘Back to Login →’ link, the user is navigated to the Login page.		
<b>Pre-conditions</b>	<b>Test Data</b>		
Access to a browser	-		
<b>Steps</b>	3. The user navigates to the Forgot Password page. 4. The user clicks on the ‘Back to Login →’ link (Figure 77).		
<b>Expected Result</b>	<b>Actual Result</b>		
The user is navigated to the Login page.	Same as the Expected Result		

**Figure 77**

Navigating to Login from Forgot Password page

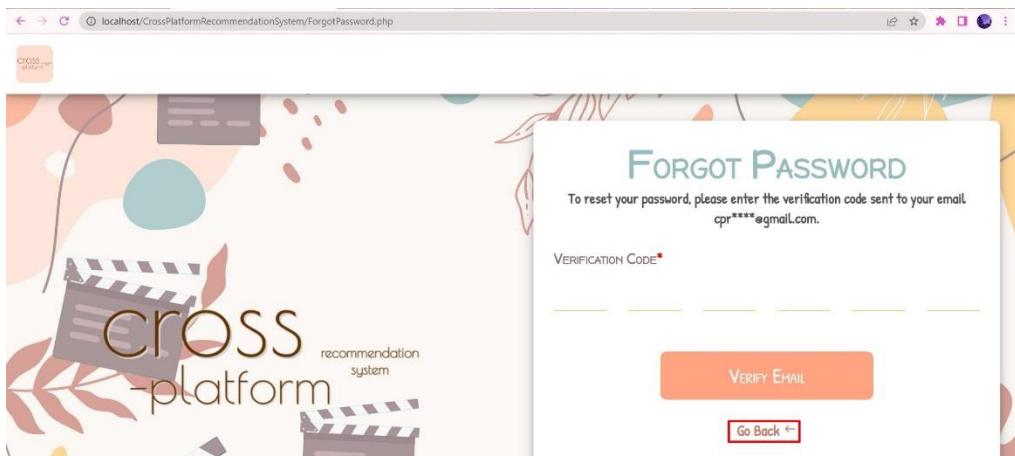


**Table 90**

<b>Test Case ID</b>	FP_06	<b>Status</b>	Pass
<b>Module Name</b>	Forgot Password		
<b>Test Case Title</b>	Forgot Password – Go to Input Email		
<b>Test Case Description</b>	When the user clicks on the ‘Go Back’ link on the verification page of Forgot Password, the user will be navigated to the Forgot Password page 1.		
<b>Pre-conditions</b>	<b>Test Data</b>		
<ul style="list-style-type: none"> <li>• Access to a browser</li> <li>• The user has already entered his email and is at the verification code page.</li> </ul>	email = cprtestuser1@gmail.com		
<b>Steps</b>	1. Click on the ‘Go Back’ link (Figure 78).		
<b>Expected Result</b>	<b>Actual Result</b>		
The user will be navigated to the Forgot Password page 1.	Same as the Expected Result		

**Figure 78**

*Go Back link*

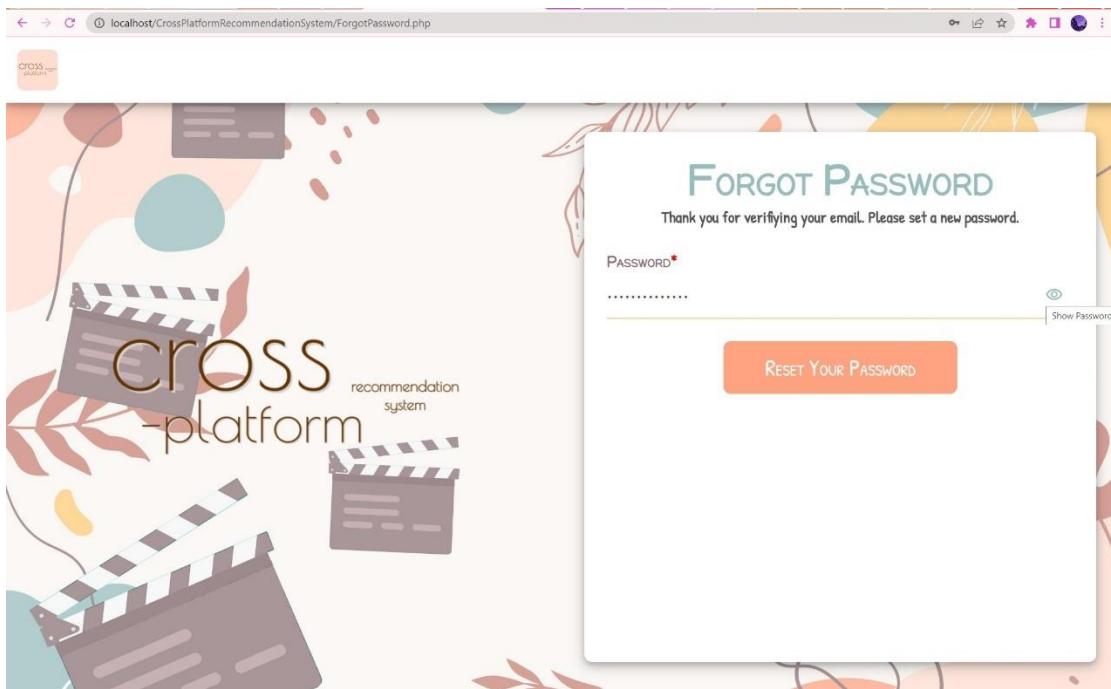


**Table 91**

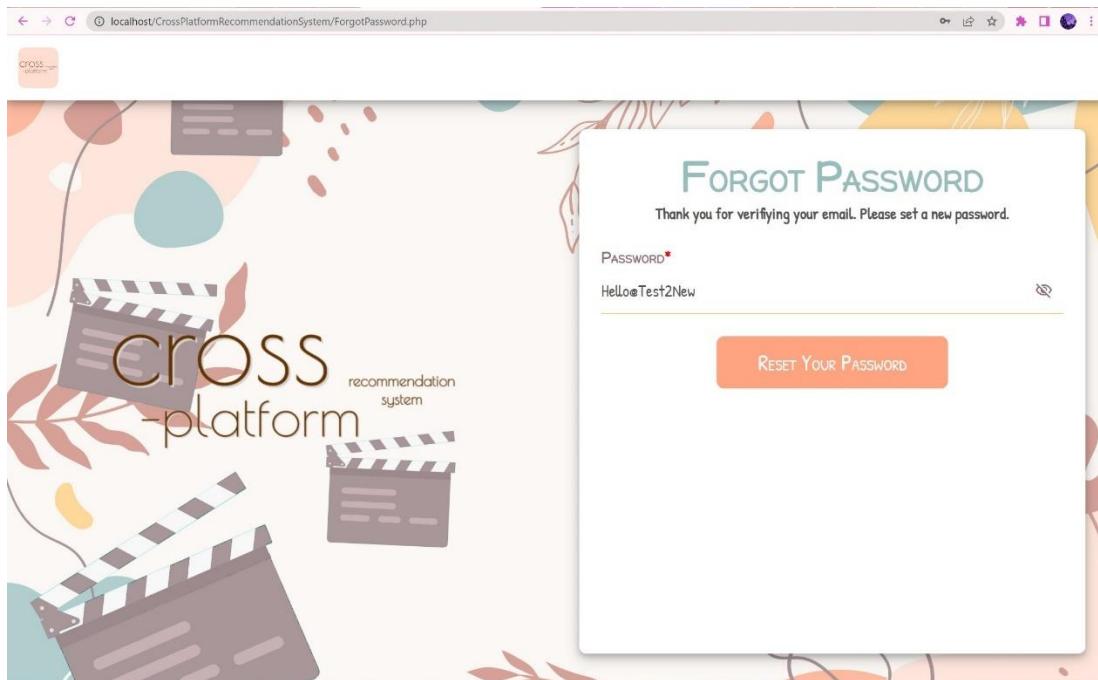
<b>Test Case ID</b>	FP_07	<b>Status</b>	Pass
<b>Module Name</b>	Forgot Password		
<b>Test Case Title</b>	Forgot Password – Toggle Show Password		
<b>Test Case Description</b>	When the user clicks on the Show/Hide Password button, the password should be shown or hidden.		
<b>Pre-conditions</b>	<p>• Access to a browser</p> <p>• The user has already entered his email and verification code and is at the new password page.</p>		
<b>Steps</b>	<ol style="list-style-type: none"> <li>1. Click on the Show Password button (Figure 79).</li> <li>2. Click on the Hide Password button (Figure 80).</li> </ol>		
<b>Expected Result</b>	<p>When the Show Password is clicked in step 3, the password is visible to the user (Figure 81).</p> <p>When the Hide Password is clicked in step 4, the password is masked.</p>		
<b>Actual Result</b>	Same as the Expected Result		

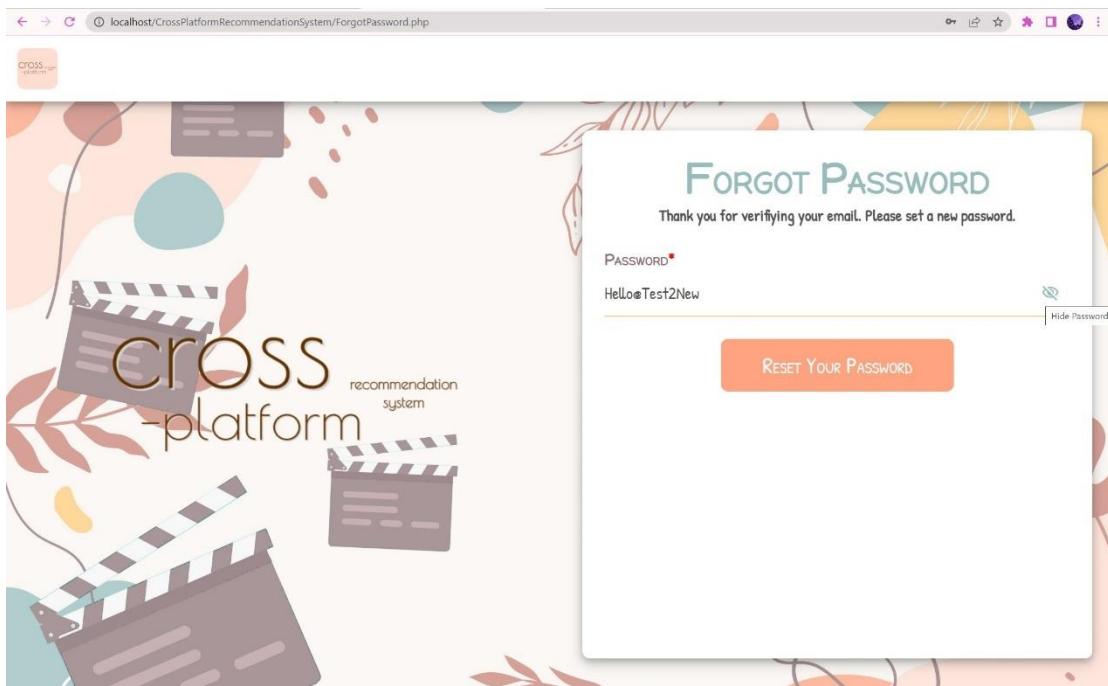
**Figure 79**

*Show Password button*

**Figure 80**

*Password is visible after clicking on the Show password button*



**Figure 81***Hide Password button***Table 92**

<b>Test Case ID</b>	FP_08	<b>Status</b>	Pass
<b>Module Name</b>	Forgot Password		
<b>Test Case Title</b>	Forgot Password – New User Already Logged In		
<b>Test Case Description</b>	When the new user is already logged in but navigates to the Forgot Password page, the user is redirected to the NewUserPreferences page.		
<b>Pre-conditions</b>	<b>Test Data</b>		
<ul style="list-style-type: none"> <li>• Access to a browser</li> <li>• The user is a new user, i.e., he hasn't set his preferences yet and is logged in.</li> </ul>	email = cprstestuser1@gmail.com  password = Hello@Test1		

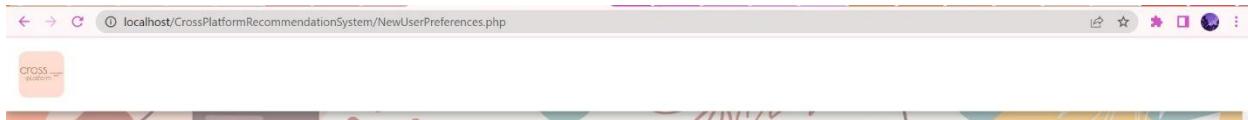
<b>Test Case ID</b>	FP_08	<b>Status</b>	Pass
<b>Steps</b>	1. Navigate to the Forgot Password page (Figure 82).		
<b>Expected Result</b>	<b>Actual Result</b>		
The user should be navigated to the NewUserPreferences page (Figure 83).	Same as the Expected Result		

**Figure 82**

*New user trying to navigate to the Forgot Password page*

**Figure 83**

*New user is navigated to the NewUserPreferences page*

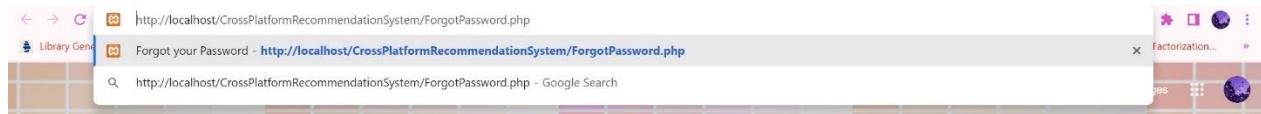
**Table 93**

<b>Test Case ID</b>	FP_09	<b>Status</b>	Pass
<b>Module Name</b>	Forgot Password		
<b>Test Case Title</b>	Forgot Password – Existing User Already Logged In		
<b>Test Case Description</b>	When the existing user is already logged in but navigates to the Forgot Password page, the user is redirected to the Homepage.		
<b>Pre-conditions</b>	<b>Test Data</b>		
<ul style="list-style-type: none"> <li>• Access to a browser</li> <li>• The user is an existing user.</li> </ul>	email = cprstestuser2@gmail.com password = Hello@Test2		

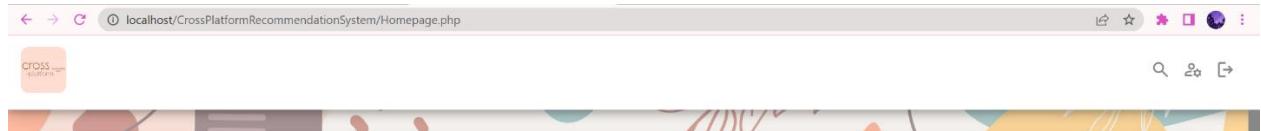
<b>Test Case ID</b>	FP_09	<b>Status</b>	Pass
<b>Steps</b>	1. Navigate to the Sign Up page (Figure 84).		
<b>Expected Result</b>	<b>Actual Result</b>		
The user should be navigated to the Homepage page (Figure 85).	Same as the Expected Result		

**Figure 84**

*Existing user trying to navigate to the Forgot Password page*

**Figure 85**

*Existing user is navigated to the Homepage page*

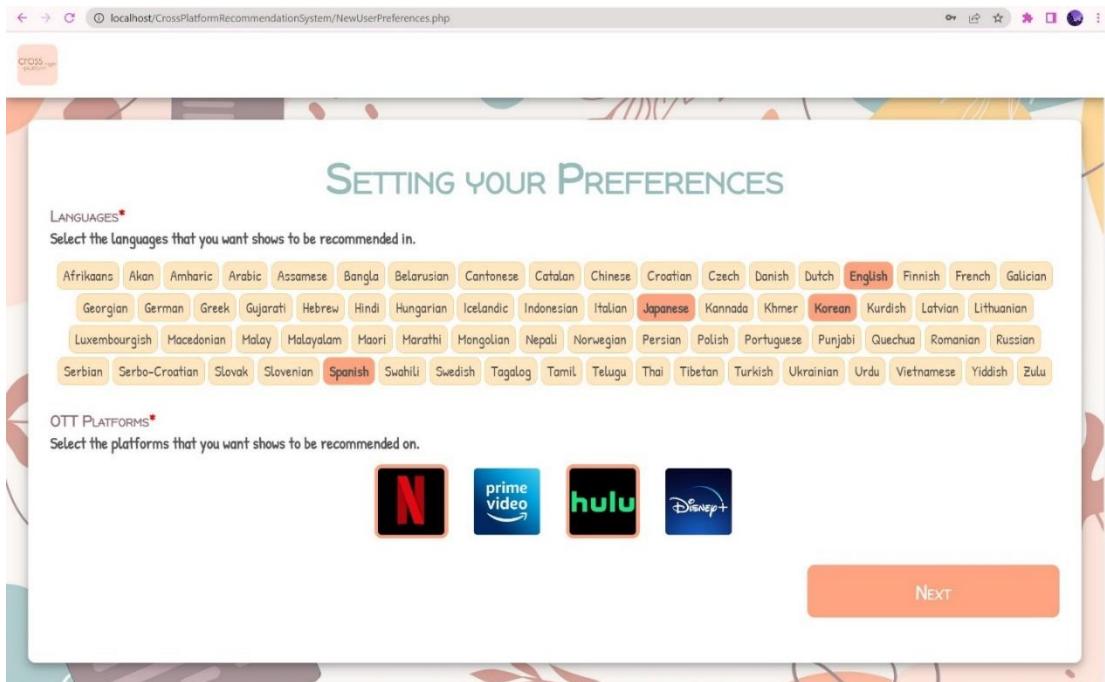
**Table 94**

<b>Test Case ID</b>	NU_01	<b>Status</b>	Pass
<b>Module Name</b>	New User Preferences		
<b>Test Case Title</b>	New User Preferences		
<b>Test Case Description</b>	When the new user navigates to the New User Preferences page, the user can set his preferences.		

<b>Test Case ID</b>	NU_01	<b>Status</b>	Pass
<b>Pre-conditions</b>	<b>Test Data</b>		
• Access to a browser  • The user is a new user, i.e., he hasn't set his preferences yet and is logged in.	email = cprstestuser1@gmail.com  password = Hello@Test1  languagesPreferred = English, Japanese, Korean, Spanish, Telugu  platformsPreferred = Netflix, Hulu  showsRated = (Save Me, 7)		
<b>Steps</b>	<ol style="list-style-type: none"> <li>1. Navigate to the NewUserPreferences page.</li> <li>2. Select the preferred languages and platforms (Figure 86).</li> <li>3. Click on the Next button.</li> <li>4. Provide ratings for a few shows (Figure 87).</li> <li>5. Click on the Get Started button.</li> </ol>		
<b>Expected Result</b>	<b>Actual Result</b>		
• The preferences and ratings provided are set for the user.  • The user should be navigated to the Homepage page (Figure 88).	Same as the Expected Result		

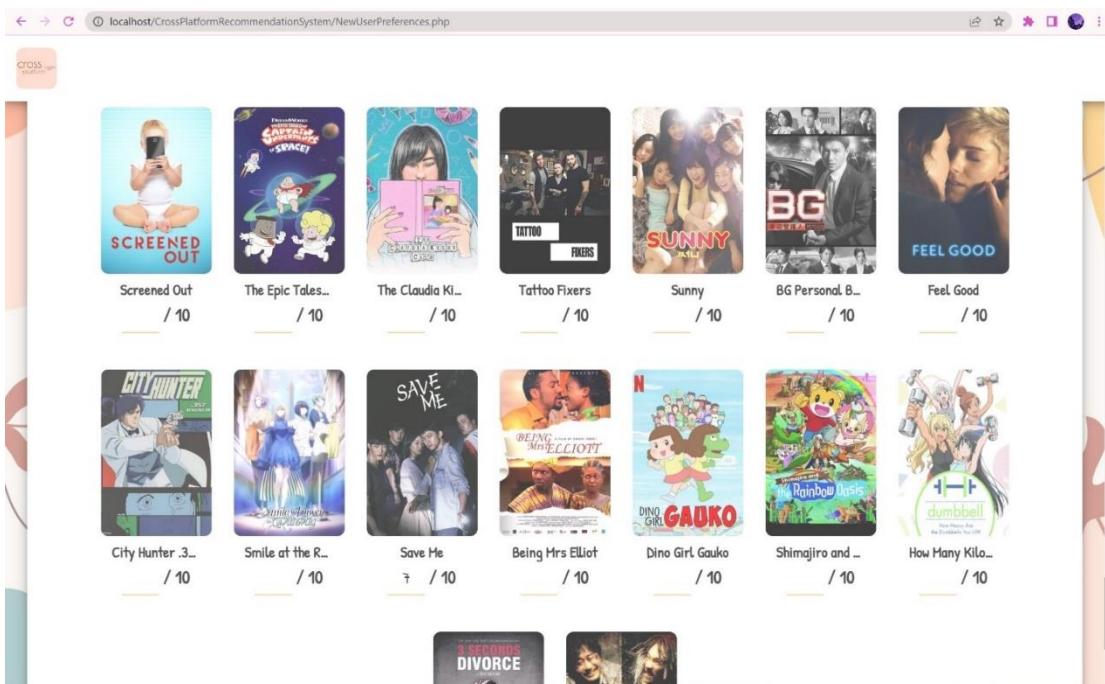
**Figure 86**

User's selected preferred languages and platforms on the NewUserPreferences page



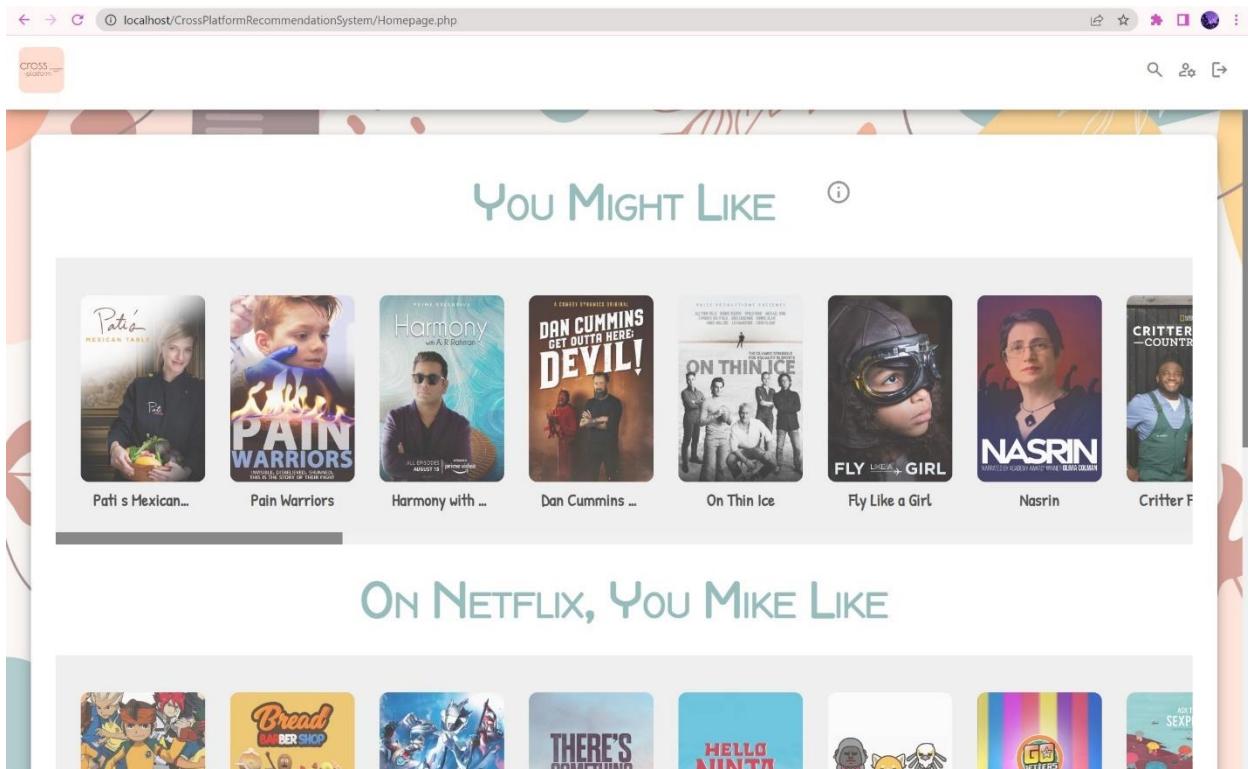
**Figure 87**

Ratings provided by the user on the NewUserPreferences page



**Figure 88**

*Homepage of the user who has just set his preferences*

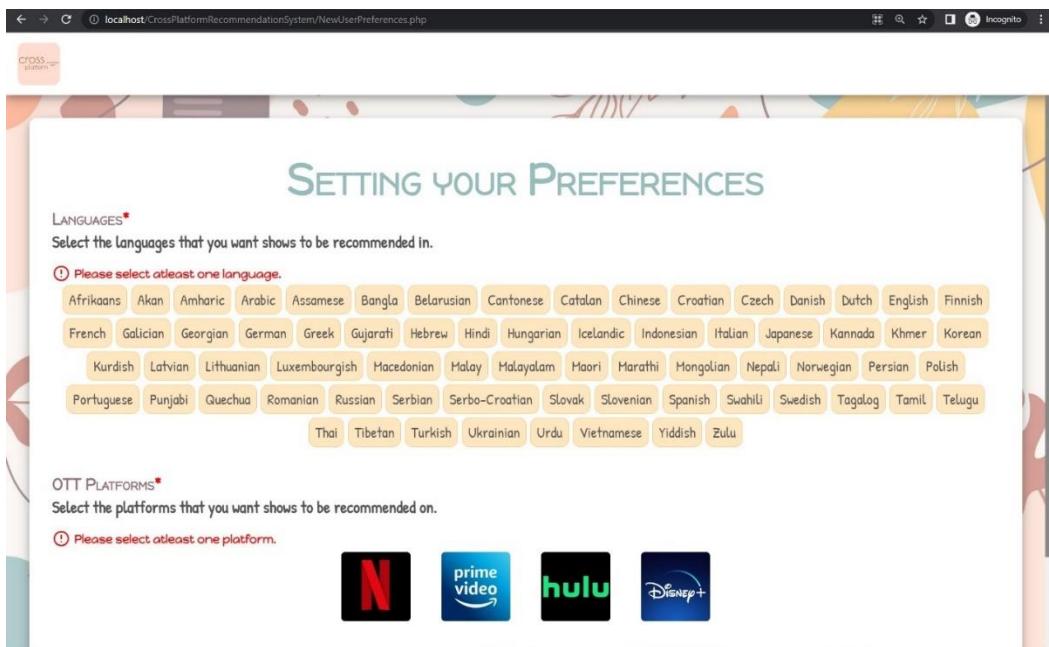
**Table 95**

<b>Test Case ID</b>	NU_02	<b>Status</b>	Pass
<b>Module Name</b>	New User Preferences		
<b>Test Case Title</b>	New User Preferences – No Preferences Selected		
<b>Test Case Description</b>	When the new user did not set his preferences, show the appropriate error messages.		
<b>Pre-conditions</b>	<b>Test Data</b>		
<ul style="list-style-type: none"> <li>Access to a browser</li> <li>The user is a new user, i.e., he hasn't set his preferences yet and is logged in.</li> </ul>	email = cprstestuser1@gmail.com password = Hello@Test1		

<b>Test Case ID</b>	NU_02	<b>Status</b>	Pass
<b>Steps</b>	<ol style="list-style-type: none"> <li>1. Navigate to the NewUserPreferences page.</li> <li>2. Click on the Next button without selecting any language or platform preferences.</li> </ol>		
<b>Expected Result</b>	<p>The message ‘Please select at least one language’ should be displayed if the user has not selected any languages (Figure 89).</p> <p>The message ‘Please select at least one platform’ should be displayed if the user has not selected any platforms (Figure 89).</p>		

**Figure 89**

*Errors displayed when the user selects no languages and platforms*

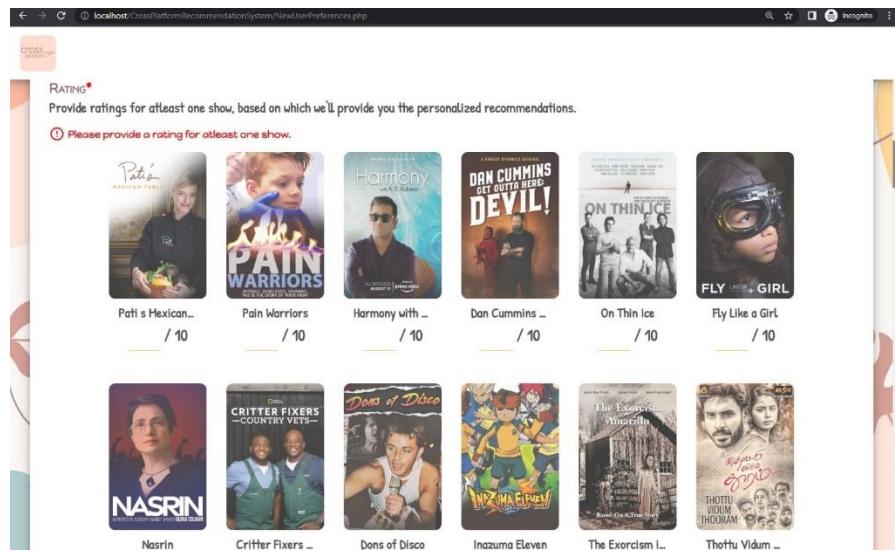


**Table 96**

<b>Test Case ID</b>	NU_03	<b>Status</b>	Pass
<b>Module Name</b>	New User Preferences		
<b>Test Case Title</b>	New User Preferences – No Shows Rated		
<b>Test Case Description</b>	When the new user did not provide a rating for any of the displayed shows, show the error message.		
<b>Pre-conditions</b>	<b>Test Data</b>		
<ul style="list-style-type: none"> <li>• Access to a browser</li> <li>• The user is a new user, i.e., he hasn't set his preferences yet and is logged in.</li> </ul>	email = cprstestuser1@gmail.com password = Hello@Test1 languagesPreferred = English, Japanese, Korean, Spanish, Telugu platformsPreferred = Netflix, Hulu		
<b>Steps</b>	<ol style="list-style-type: none"> <li>1. Navigate to the NewUserPreferences page.</li> <li>2. Select the preferred languages and platforms.</li> <li>3. Click on the Next button.</li> <li>4. Click on the Get Started button without rating for any show.</li> </ol>		
<b>Expected Result</b>	<b>Actual Result</b>		
The message 'Please provide rating for at least one show' should be displayed (Figure 90).	Same as the Expected Result		

**Figure 90**

*Error displayed when the user does not rate for any shows displayed*

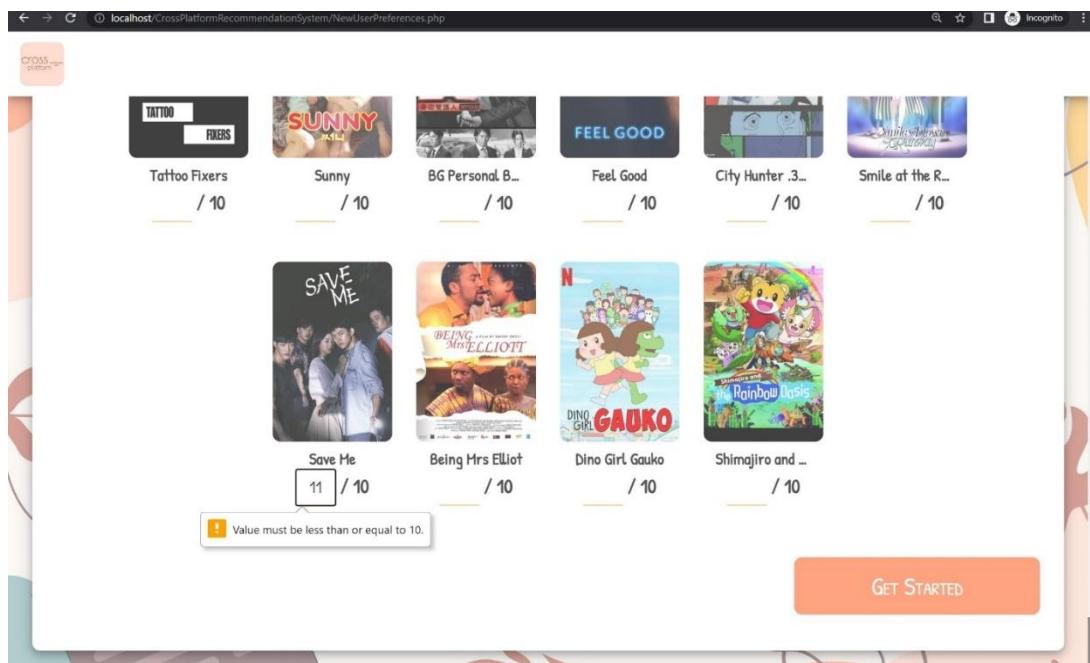
**Table 97**

<b>Test Case ID</b>	NU_04	<b>Status</b>	Pass
<b>Module Name</b>	New User Preferences		
<b>Test Case Title</b>	New User Preferences – Invalid Rating		
<b>Test Case Description</b>	When the new user provides an invalid rating for any of the displayed shows, show the error message.		
<b>Pre-conditions</b>	<b>Test Data</b>		
<ul style="list-style-type: none"> <li>Access to a browser</li> <li>The user is a new user, i.e., he hasn't set his preferences yet and is logged in.</li> </ul>	email = cprstestuser1@gmail.com password = Hello@Test1 languagesPreferred = English, Japanese, Korean, Spanish, Telugu platformsPreferred = Netflix, Hulu shows = (Save Me, 11)		

<b>Test Case ID</b>	NU_04	<b>Status</b>	Pass
<b>Steps</b>	<ol style="list-style-type: none"> <li>1. Navigate to the NewUserPreferences page.</li> <li>2. Select the preferred languages and platforms.</li> <li>3. Click on the Next button.</li> <li>4. Provide an invalid rating for a show displayed.</li> <li>5. Click on the Get Started button.</li> </ol>		
<b>Expected Result</b>	<p>Depending on the rating entered, the message          ‘Value must be less than or equal to 10’ or          ‘Value must be greater than or equal to 0’          should be displayed (Figure 91).</p>		

**Figure 91**

*Error displayed when the user enters an invalid rating*

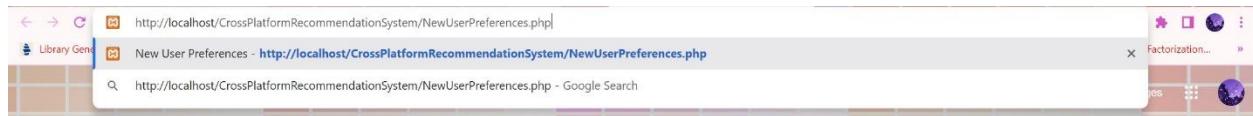


**Table 98**

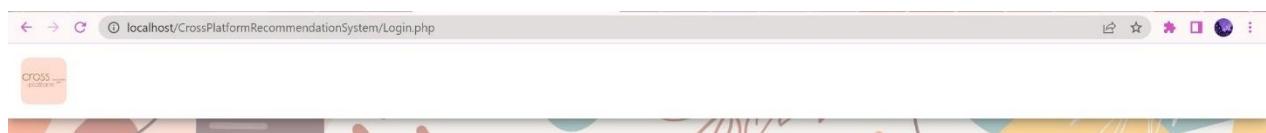
<b>Test Case ID</b>	NU_05	<b>Status</b>	Pass
<b>Module Name</b>	New User Preferences		
<b>Test Case Title</b>	New User Preferences – User Not Logged In		
<b>Test Case Description</b>	The user is not logged in and navigates to the NewUserPreferences page.		
<b>Pre-conditions</b>	<b>Test Data</b>		
Access to a browser	-		
<b>Steps</b>	1. Navigate to the NewUserPreferences page (Figure 92).		
<b>Expected Result</b>	<b>Actual Result</b>		
Since the user is not logged in, he is navigated to the Login page (Figure 93).	Same as the Expected Result		

**Figure 92**

*Existing user trying to navigate to the NewUserPreferences page*

**Figure 93**

*Existing user is navigated to the Login page*

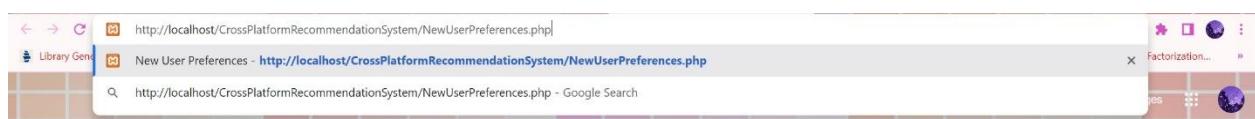


**Table 99**

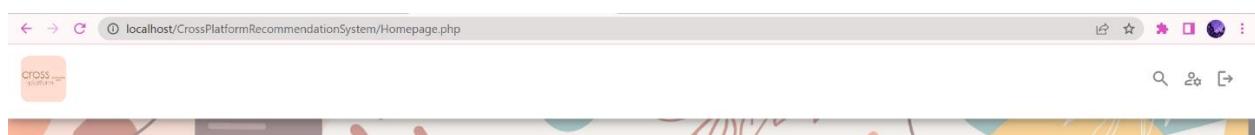
<b>Test Case ID</b>	NU_06	<b>Status</b>	Pass
<b>Module Name</b>	New User Preferences		
<b>Test Case Title</b>	New User Preferences – Existing User Logged In		
<b>Test Case Description</b>	The existing user is logged in and navigates to NewUserPreferences page.		
<b>Pre-conditions</b>	<b>Test Data</b>		
<ul style="list-style-type: none"> <li>• Access to a browser</li> <li>• The user is an existing user, i.e., he has set his preferences yet and is logged in.</li> </ul>	-		
<b>Steps</b>	1. Navigate to the NewUserPreferences page (Figure 94).		
<b>Expected Result</b>	<b>Actual Result</b>		
Since the user is an existing user, he is navigated to the Homepage page (Figure 95).	Same as the Expected Result		

**Figure 94**

*Existing user trying to navigate to the NewUserPreferences page*

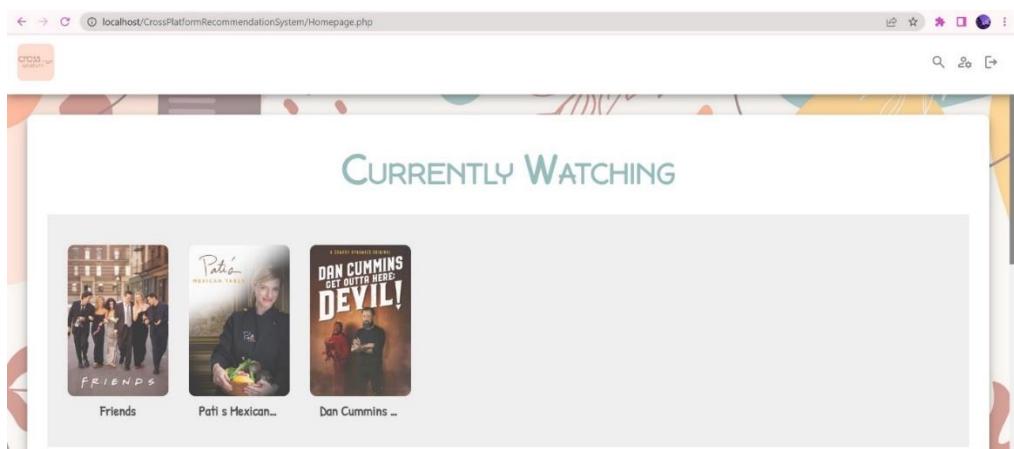
**Figure 95**

*Existing user is navigated to the Homepage page*



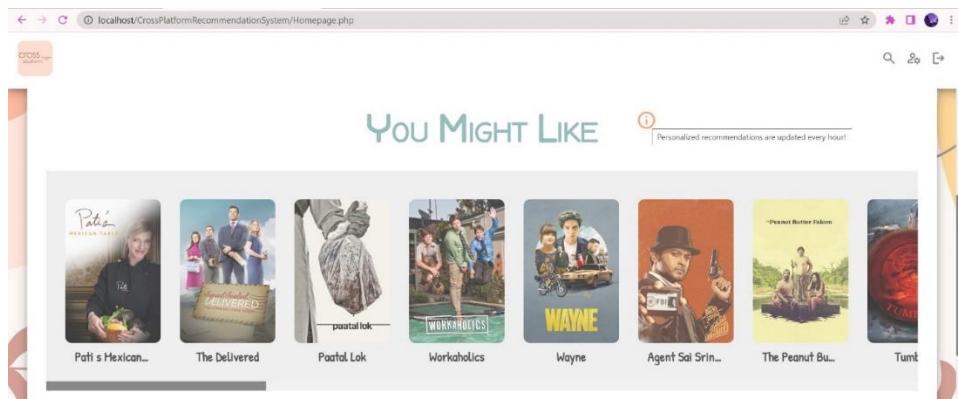
**Table 100**

<b>Test Case ID</b>	HP_01	<b>Status</b>	Pass
<b>Module Name</b>	Homepage		
<b>Test Case Title</b>	Homepage – Currently Watching		
<b>Test Case Description</b>	The user navigates to the homepage and can see the shows he set as Currently Watching.		
<b>Pre-conditions</b>	<b>Test Data</b>		
<ul style="list-style-type: none"> <li>• Access to a browser</li> <li>• The user is an existing user and is logged in.</li> </ul>	email = cprstestuser1@gmail.com password = Hello@Test1		
<b>Steps</b>	1. Navigate to the Homepage.		
<b>Expected Result</b>	<b>Actual Result</b>		
If the user has set any shows as Currently Watching, the user can see them under the ‘Currently Watching’ section (Figure 96).	Same as the Expected Result		

**Figure 96***Currently Watching*

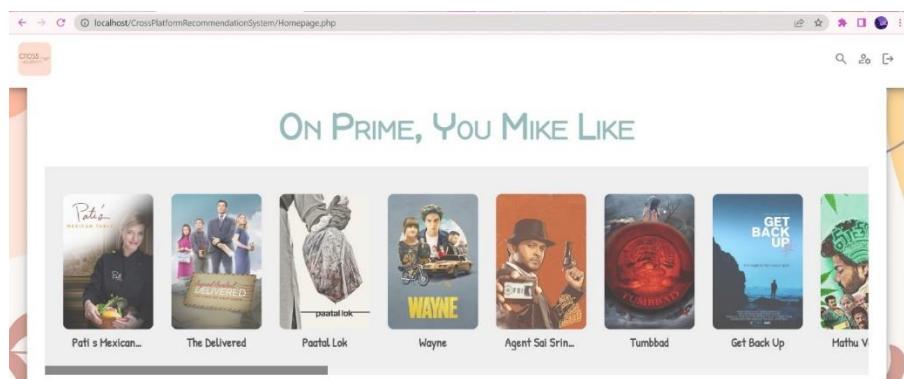
**Table 101**

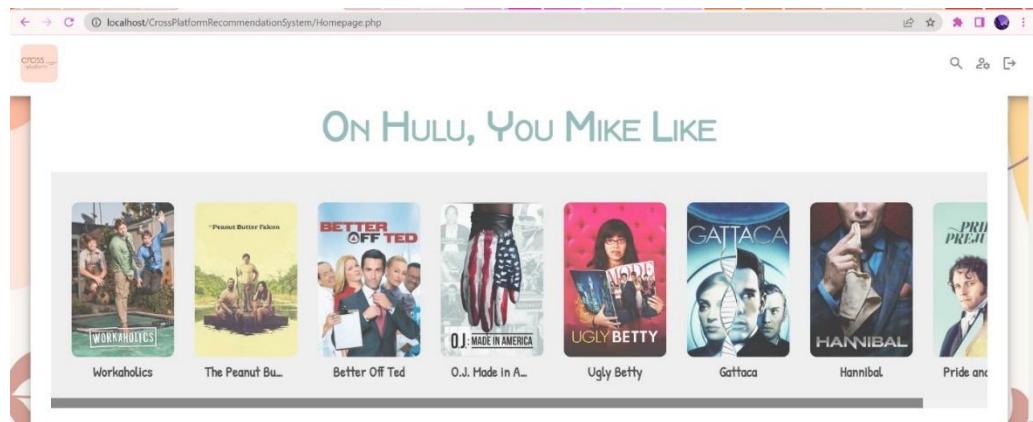
<b>Test Case ID</b>	HP_02	<b>Status</b>	Pass
<b>Module Name</b>	Homepage		
<b>Test Case Title</b>	Homepage – Personalized Recommendations		
<b>Test Case Description</b>	The user navigates to the homepage and can see the personalized recommendations generated for the user.		
<b>Pre-conditions</b>	<b>Test Data</b>		
<ul style="list-style-type: none"> <li>• Access to a browser</li> <li>• The user is an existing user and is logged in.</li> </ul>	email = cprstestuser1@gmail.com password = Hello@Test1		
<b>Steps</b>	1. Navigate to the Homepage.		
<b>Expected Result</b>	<b>Actual Result</b>		
The user can see the personalized recommendations under 'You Might Like' section (Figure 97).	Same as the Expected Result		

**Figure 97***Personalized Recommendations*

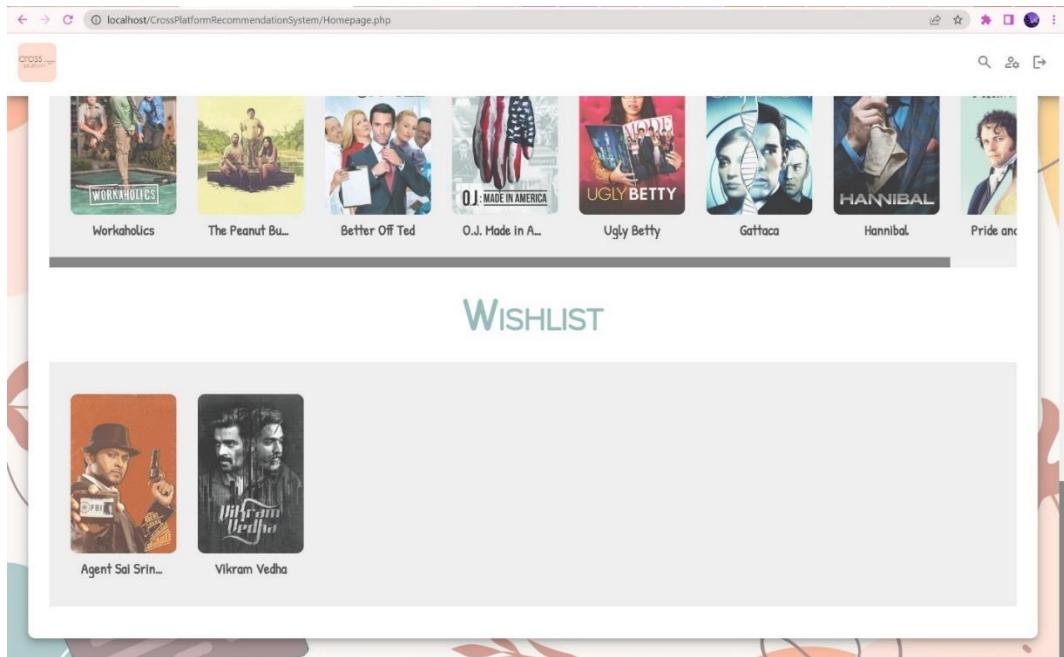
**Table 102**

<b>Test Case ID</b>	HP_03	<b>Status</b>	Pass
<b>Module Name</b>	Homepage		
<b>Test Case Title</b>	Homepage – Platform-Wise Personalized Recommendations		
<b>Test Case Description</b>	The user navigates to the homepage and can see the personalized recommendations generated for the user for each platform preferred by the user.		
<b>Pre-conditions</b>	<b>Test Data</b>		
<ul style="list-style-type: none"> <li>• Access to a browser</li> <li>• The user is an existing user and is logged in.</li> </ul>	email = cprstestuser1@gmail.com password = Hello@Test1 platformPreferences = Prime, Hulu		
<b>Steps</b>	1. Navigate to the Homepage.		
<b>Expected Result</b>	<b>Actual Result</b>		
The user can see the platform-specific personalized recommendations (Figures 98, 99).	Same as the Expected Result		

**Figure 98***Personalized Recommendations on Prime*

**Figure 99***Personalized Recommendations on Hulu***Table 103**

<b>Test Case ID</b>	HP_04	<b>Status</b>	Pass
<b>Module Name</b>	Homepage		
<b>Test Case Title</b>	Homepage – Wishlist		
<b>Test Case Description</b>	The user navigates to the homepage and can see the shows he set as Wishlist.		
<b>Pre-conditions</b>	<b>Test Data</b>		
<ul style="list-style-type: none"> <li>• Access to a browser</li> <li>• The user is an existing user and is logged in.</li> </ul>	email = <a href="mailto:elaks.edu@gmail.com">elaks.edu@gmail.com</a> password = Hello123		
<b>Steps</b>	1. Navigate to the Homepage.		
<b>Expected Result</b>	<b>Actual Result</b>		
If the user has set any shows as Wishlist, the user can see them under the ‘Wishlist’ section (Figure 100).	Same as the Expected Result		

**Figure 100***Wishlist***Table 104**

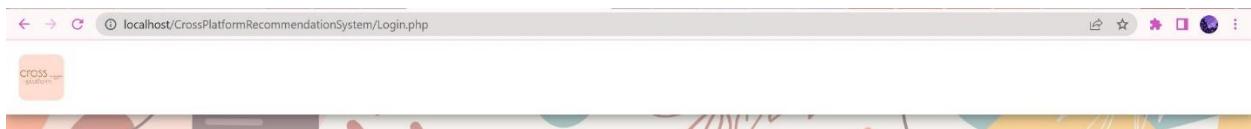
<b>Test Case ID</b>	HP_05	<b>Status</b>	Pass
<b>Module Name</b>	Homepage		
<b>Test Case Title</b>	Homepage – User Not Logged In		
<b>Test Case Description</b>	The user is not logged in and navigates to the Homepage.		
<b>Pre-conditions</b>	<b>Test Data</b>		
Access to a browser	-		
<b>Steps</b>	1. Navigate to the Homepage (Figure 101).		
<b>Expected Result</b>	<b>Actual Result</b>		
Since the user is not logged in, he is navigated to the Login page (Figure 102).	Same as the Expected Result		

**Figure 101**

*Existing user trying to navigate to the Homepage page*

**Figure 102**

*Existing user is navigated to the Login page*

**Table 105**

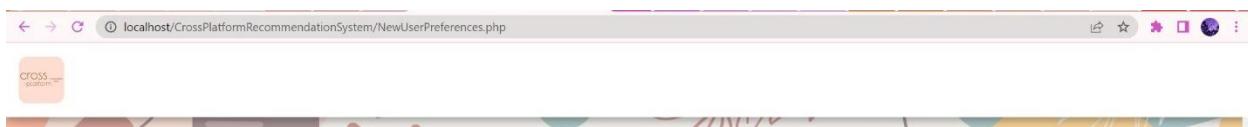
<b>Test Case ID</b>	HP_06	<b>Status</b>	Pass
<b>Module Name</b>	Homepage		
<b>Test Case Title</b>	Homepage – New User Logged In		
<b>Test Case Description</b>	The new user is logged in and navigates to Homepage.		
<b>Pre-conditions</b>	<b>Test Data</b>		
<ul style="list-style-type: none"> <li>• Access to a browser</li> <li>• The user is a new user, i.e., he hasn't set his preferences yet and is logged in.</li> </ul>	-		
<b>Steps</b>	1. Navigate to the Homepage (Figure 103).		
<b>Expected Result</b>	<b>Actual Result</b>		
Since the user is a new user, he is navigated to the NewUserPreferences page (Figure 104).	Same as the Expected Result		

**Figure 103**

*Existing user trying to navigate to the Homepage page*

**Figure 104**

*Existing user is navigated to the NewUserPreferences page*

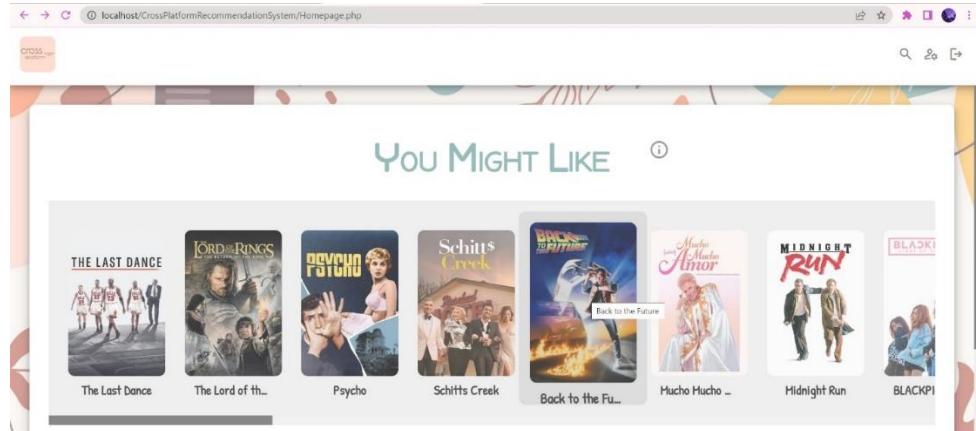
**Table 106**

<b>Test Case ID</b>	HP_07	<b>Status</b>	Pass
<b>Module Name</b>	Homepage		
<b>Test Case Title</b>	Homepage – Go To Show		
<b>Test Case Description</b>	The user navigates to the homepage, and when he clicks on a show, the user is navigated to its ShowInformation page.		
<b>Pre-conditions</b>	<b>Test Data</b>		
<ul style="list-style-type: none"> <li>Access to a browser</li> <li>The user is an existing user and is logged in.</li> </ul>	email = cprstestuser1@gmail.com password = Hello@Test1 showName = Back to the Future		
<b>Steps</b>	1. Navigate to the Homepage. 2. Click on any of the shows displayed (Figure 105).		
<b>Expected Result</b>	<b>Actual Result</b>		
The user is navigated to its ShowInformation page (Figure 106).	Same as the Expected Result		

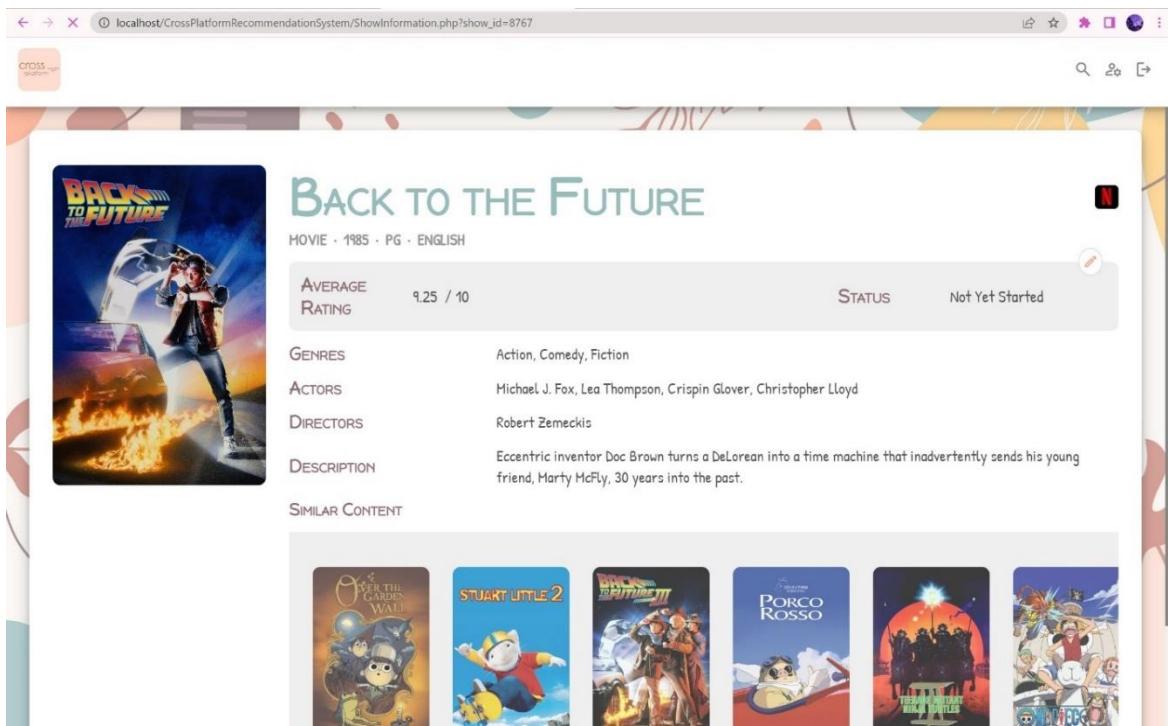
<b>Test Case ID</b>	HP_07	<b>Status</b>	Pass
<b>Notes</b>	The show_id of the clicked show is passed using GET to the ShowInformation page.		

**Figure 105**

Show clicked on the Homepage

**Figure 106**

Show Information



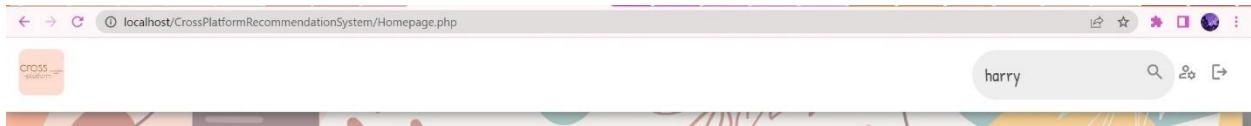
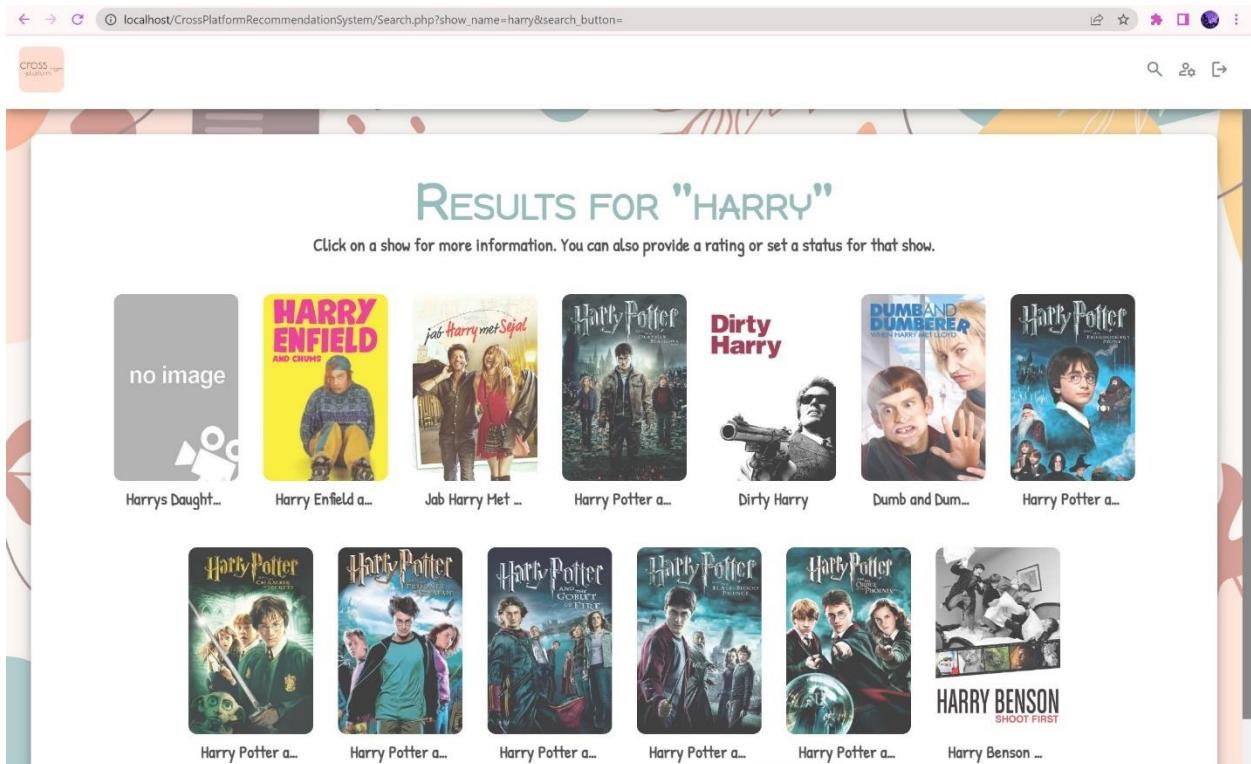
**Table 107**

<b>Test Case ID</b>	NB_01	<b>Status</b>	Pass
<b>Module Name</b>	Navigation Bar		
<b>Test Case Title</b>	Navigation Bar – Logout		
<b>Test Case Description</b>	When the user clicks on the Logout button on the navigation bar, he is logged out.		
<b>Pre-conditions</b>	<b>Test Data</b>		
<ul style="list-style-type: none"> <li>• Access to a browser</li> <li>• The user is already logged in.</li> </ul>	email = cprstestuser1@gmail.com password = Hello@Test1		
<b>Steps</b>	1. Navigate to the Homepage, Profile, ShowInformation, or Search page.  2. Click on the Logout button in the navigation bar (Figure 107).		
<b>Expected Result</b>	<b>Actual Result</b>		
The user is logged out and is navigated to the Login page.	Same as the Expected Result		

**Figure 107***Logout button*

**Table 108**

<b>Test Case ID</b>	NB_02	<b>Status</b>	Pass
<b>Module Name</b>	Navigation Bar		
<b>Test Case Title</b>	Navigation Bar – Search		
<b>Test Case Description</b>	When the user clicks on the Search button on the navigation bar, he can search for any show.		
<b>Pre-conditions</b>	<p>• Access to a browser</p> <p>• The user is an existing user and is already logged in.</p>		
	email = cprstestuser1@gmail.com password = Hello@Test1 search_keyword = harry		
<b>Steps</b>	<ol style="list-style-type: none"> <li>1. Navigate to the Homepage, Profile, ShowInformation, or Search page.</li> <li>2. Click on the Search button in the navigation bar (Figure 108).</li> <li>3. A textbox to enter the search keyword appear.</li> <li>4. Type the keyword and click on the Search button again (Figure 109).</li> </ol>		
<b>Expected Result</b>	<b>Actual Result</b>		
The user is navigated to the Search page where the results of the search are displayed (Figure 120).	Same as the Expected Result		
<b>Notes</b>	The search keyword is passed to the Search page using GET (the search keyword is passed as a value to show_name keyword).		

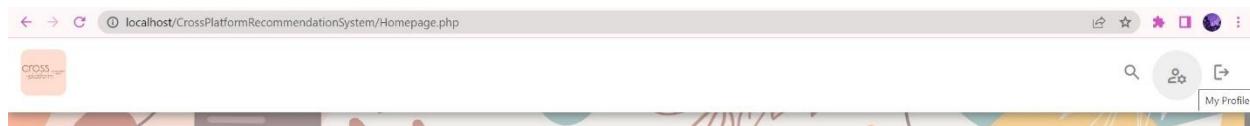
**Figure 108***Search button***Figure 109***Search keyword entered by the user***Figure 120***Search results*

**Table 109**

<b>Test Case ID</b>	NB_03	<b>Status</b>	Pass
<b>Module Name</b>	Navigation Bar		
<b>Test Case Title</b>	Navigation Bar – Go To Profile		
<b>Test Case Description</b>	When the user clicks on the Profile button on the navigation bar, he is navigated to the Profile page.		
<b>Pre-conditions</b>	<ul style="list-style-type: none"> <li>• Access to a browser</li> <li>• The user is an existing user and is already logged in.</li> </ul>		
<b>Steps</b>	<ol style="list-style-type: none"> <li>1. Navigate to the Homepage, Profile, ShowInformation, or Search page.</li> <li>2. Click on the Profile button in the navigation bar (Figure 121).</li> </ol>		
<b>Expected Result</b>	<b>Actual Result</b>		
The user is navigated to the Profile page (Figure 122).	Same as the Expected Result		

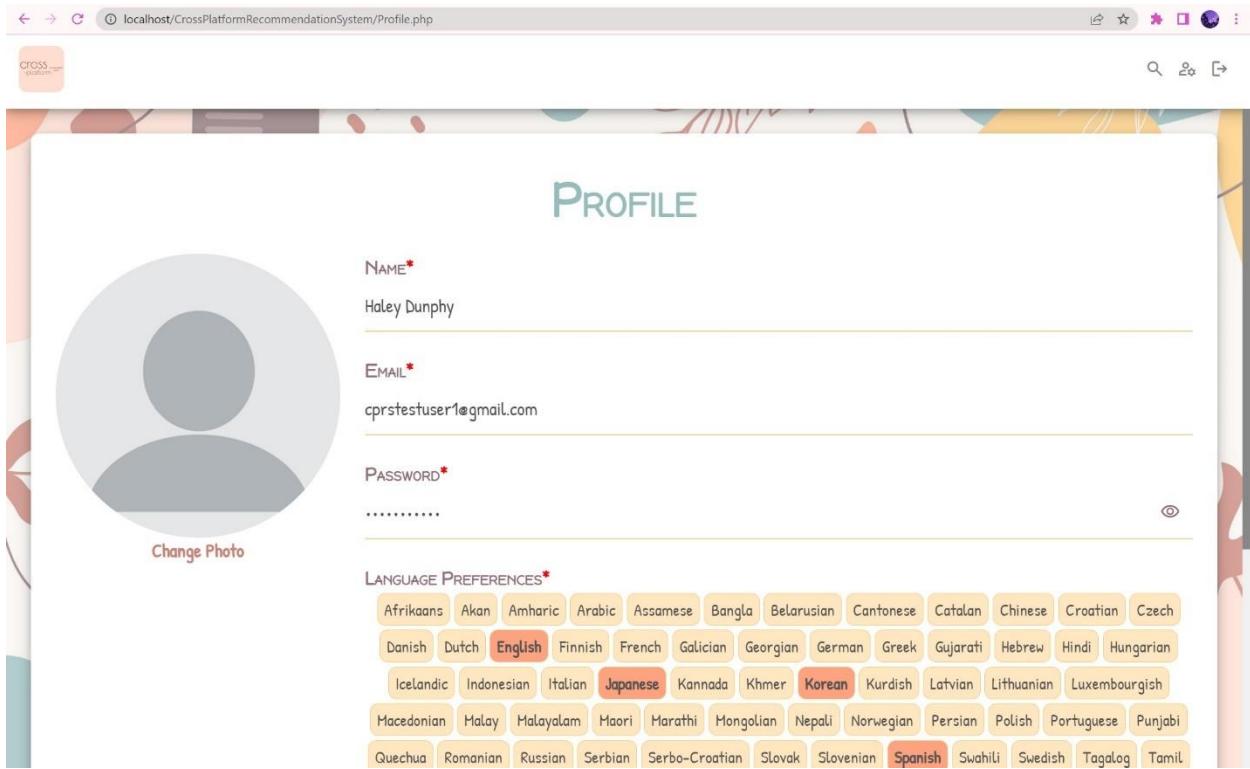
**Figure 121**

*Profile button*



**Figure 122**

*Navigating to the Profile page*

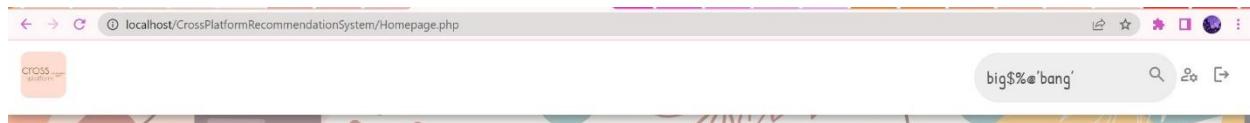
**Table 110**

<b>Test Case ID</b>	NB_04	<b>Status</b>	Pass
<b>Module Name</b>	Navigation Bar		
<b>Test Case Title</b>	Navigation Bar – Search Invalid Keyword		
<b>Test Case Description</b>	When the user clicks on the Search button on the navigation bar, he can search for any show.		
<b>Pre-conditions</b>	<b>Test Data</b>		
<ul style="list-style-type: none"> <li>• Access to a browser</li> <li>• The user is an existing user and is already logged in.</li> </ul>	email = cprstestuser1@gmail.com  password = Hello@Test1  search_keyword = big\$%@'bang'		

<b>Test Case ID</b>	NB_04	<b>Status</b>	Pass
<b>Steps</b>	<ol style="list-style-type: none"> <li>1. Navigate to the Homepage, Profile, ShowInformation, or Search page.</li> <li>2. Click on the Search button in the navigation bar.</li> <li>3. A textbox to enter the search keyword appear.</li> <li>4. Type the keyword and click on the Search button again (Figure 123).</li> </ol>		
<b>Expected Result</b>	<b>Actual Result</b>		
<ul style="list-style-type: none"> <li>• Since the keyword contains special characters, they are replaced with spaces (Figure 124).</li> <li>• The user is then navigated to the Search page where the results of the search are displayed.</li> </ul>	Same as the Expected Result		
<b>Notes</b>	<ul style="list-style-type: none"> <li>• The search keyword can consist of letters, digits, and spaces. Any other characters entered will be replaced with spaces.</li> <li>• The keyword is passed to the Search page using GET (the search keyword is passed as a value to show_name keyword).</li> </ul>		

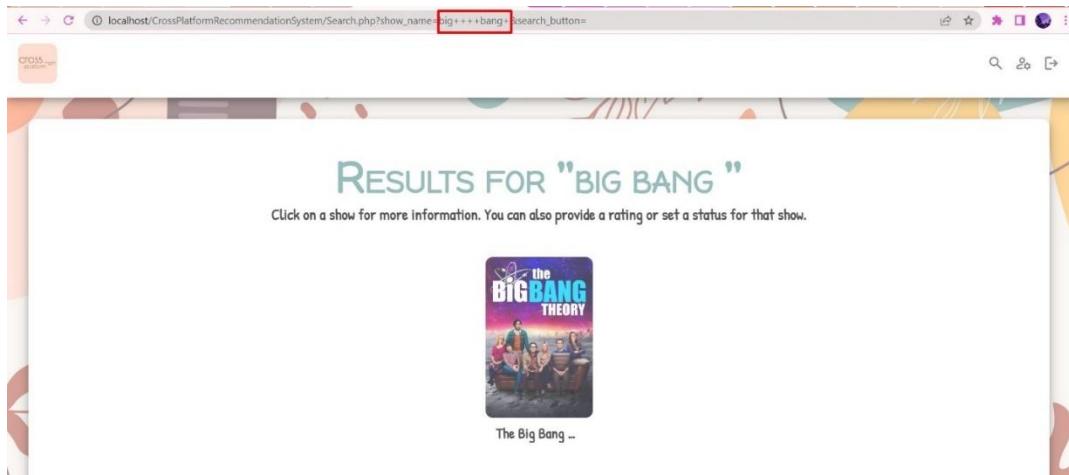
**Figure 123**

*Search keyword*



**Figure 124**

*Special characters removed in the keyword*

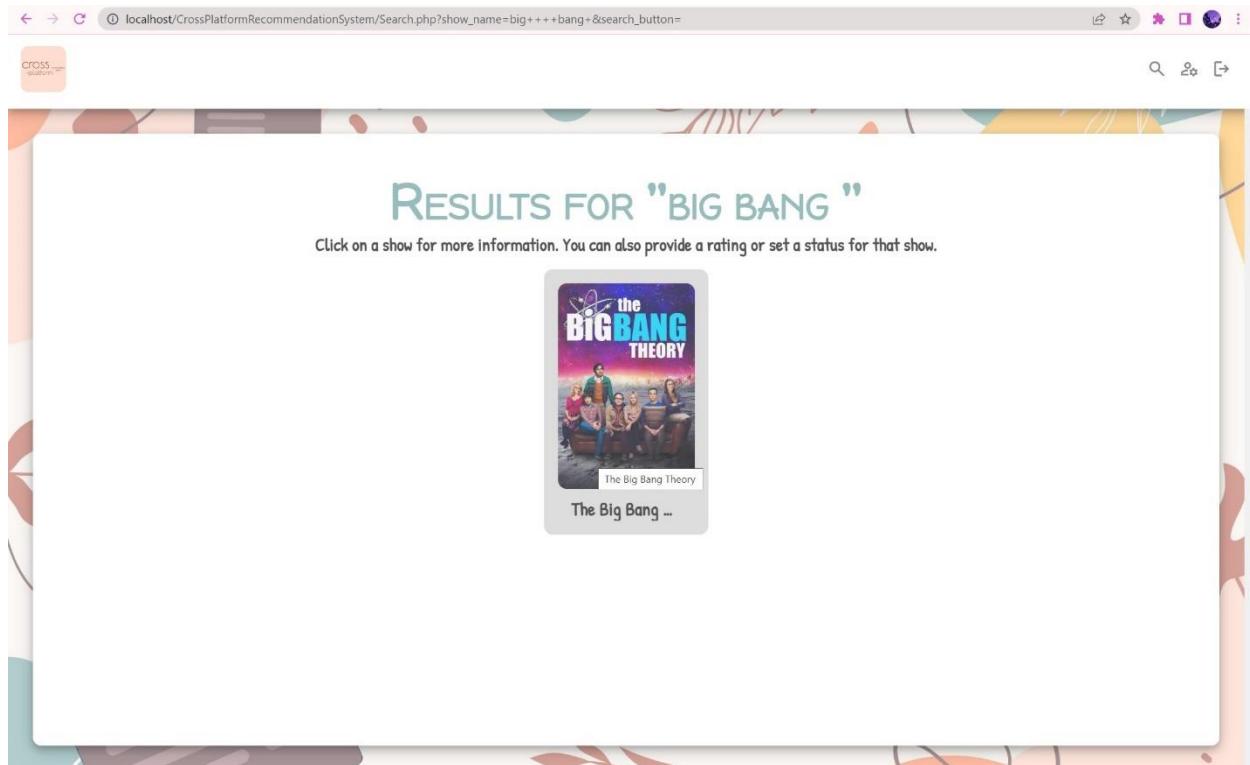
**Table 111**

<b>Test Case ID</b>	SE_01	<b>Status</b>	Pass
<b>Module Name</b>	Search		
<b>Test Case Title</b>	Search – Go To Show		
<b>Test Case Description</b>	When the user clicks on any of the show results, he is navigated to its ShowInformation page.		
<b>Pre-conditions</b>	<b>Test Data</b>		
<ul style="list-style-type: none"> <li>• Access to a browser</li> <li>• The user is an existing user and is already logged in.</li> </ul>	email = cprstestuser1@gmail.com password = Hello@Test1 show_name = big bang		
<b>Steps</b>	1. Navigate to the Search page with the value set to GET key show_name. 2. Shows matching the keyword are displayed. 3. Click on a show (Figure 125).		

<b>Test Case ID</b>	SE_01	<b>Status</b>	Pass
<b>Expected Result</b>	<b>Actual Result</b>		
The user is navigated to its ShowInformation page where the details of the show are displayed (Figure 126).	Same as the Expected Result		

**Figure 125**

*User clicks on a show*



**Figure 126**

*Show Information*

The screenshot shows a web page titled "THE BIG BANG THEORY". The page header includes the URL "localhost/CrossPlatformRecommendationSystem>ShowInformation.php?show\_id=8516". The main content area displays the following information:

- AVERAGE RATING:** xx / 10
- STATUS:** Not Yet Started
- GENRES:** Comedy
- ACTORS:** Simon Helberg, Jim Parsons, Johnny Galecki, Kaley Cuoco
- DIRECTORS:** Mark Cendrowski, Anthony Rich, Peter Chakos, Nikki Lorre
- DESCRIPTION:** Quarks fly when two geeky physicists discover that they have a new neighbor a sexy actress who inspires them to try a relationship with a real woman.
- SIMILAR CONTENT:** A row of six movie and TV show thumbnails, including "always be my Maybe", "LAUREL & HARDY", "BELOW ZERO", "CINDERELLA", "My Sweet Little Village", "Uncle P", and "Call me Kat".

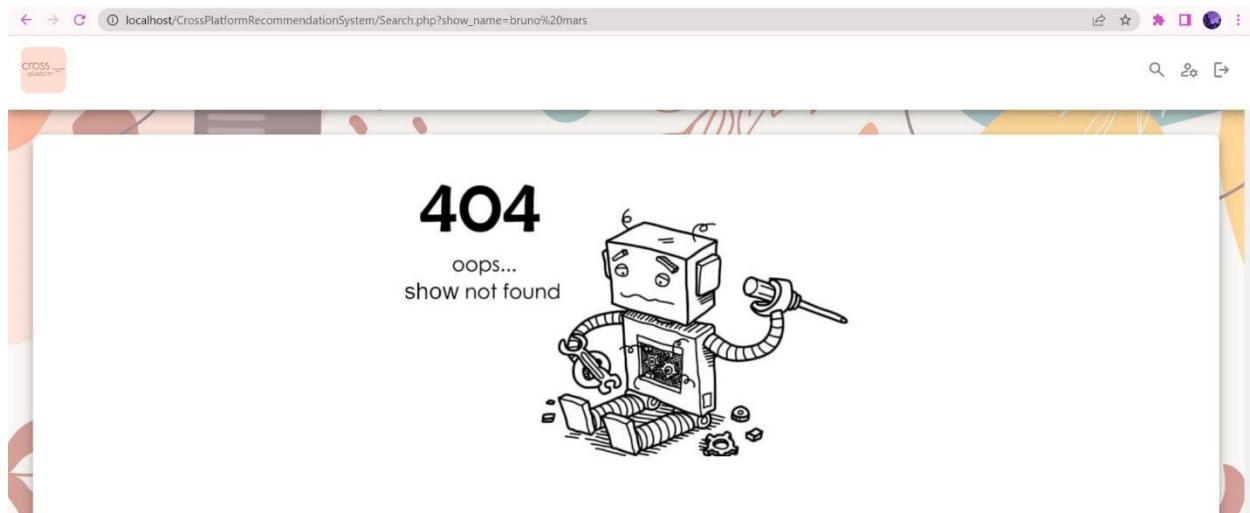
**Table 112**

<b>Test Case ID</b>	SE_02	<b>Status</b>	Pass
<b>Module Name</b>	Search		
<b>Test Case Title</b>	Search – No Shows Found		
<b>Test Case Description</b>	When there are no shows found matching the search keyword entered, a 404 error is displayed to the user.		
<b>Pre-conditions</b>	<b>Test Data</b>		
<ul style="list-style-type: none"> <li>Access to a browser</li> <li>The user is an existing user and is already logged in.</li> </ul>	email = <a href="mailto:elaks.edu@gmail.com">elaks.edu@gmail.com</a> password = Hello123 show_name = Bruno mars		

<b>Test Case ID</b>	SE_02	<b>Status</b>	Pass
<b>Steps</b>	1. Navigate to the Search page with the value set to GET key show_name.		
<b>Expected Result</b>	<b>Actual Result</b>		
Since there are no results found for the entered search keyword (show_name), a 404 error is displayed to the user (Figure 127).	Same as the Expected Result		

**Figure 127**

*No results for the entered show name in search*

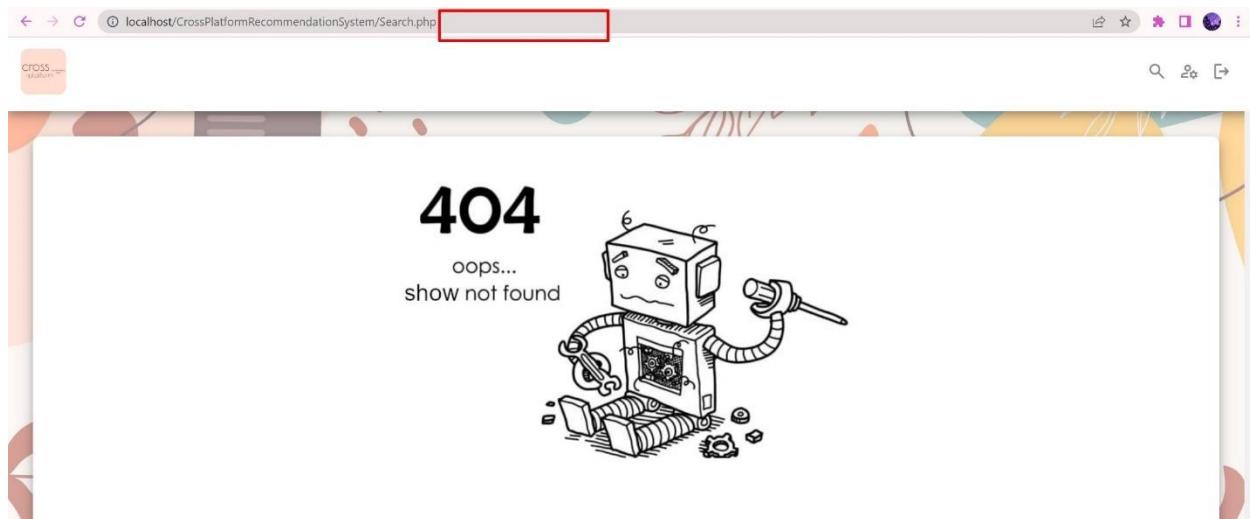
**Table 113**

<b>Test Case ID</b>	SE_03	<b>Status</b>	Pass
<b>Module Name</b>	Search		
<b>Test Case Title</b>	Search – No Search Keyword Passed		

<b>Test Case ID</b>	SE_03	<b>Status</b>	Pass
<b>Test Case Description</b>	When the user navigates to the Search page without a search keyword, a 404 error is displayed.		
<b>Pre-conditions</b>	<b>Test Data</b>		
<ul style="list-style-type: none"> <li>• Access to a browser</li> <li>• The user is an existing user and is already logged in.</li> </ul>	email = cprstestuser1@gmail.com password = Hello@Test1		
<b>Steps</b>	1. Navigate to the Search page without the GET key show_name.		
<b>Expected Result</b>	<b>Actual Result</b>		
A 404 error is displayed to the user (Figure 128).	Same as the Expected Result		

**Figure 128**

*Error displayed when there is no show\_name keyword in the Search URL*



**Table 114**

<b>Test Case ID</b>	SI_01	<b>Status</b>	Pass
<b>Module Name</b>	Show Information		
<b>Test Case Title</b>	Show Information		
<b>Test Case Description</b>	When the user is navigated to a show's ShowInformation page, the user can see the show's information displayed.		
<b>Pre-conditions</b>	<ul style="list-style-type: none"> <li>• Access to a browser</li> <li>• The user is an existing user and is already logged in.</li> </ul>		
<b>Test Data</b>	email = cprstestuser1@gmail.com password = Hello@Test1 show_id = 23297		
<b>Steps</b>	1. Navigate to the ShowInformation page with the value set to GET key show_id.		
<b>Expected Result</b>	The information belonging to that show_id is displayed to the user (Figure 129).		
<b>Actual Result</b>	Same as the Expected Result		

**Figure 129**

Show Information page for show\_id=23297

The screenshot displays the 'Show Information' page for the show with ID 23297. The main title is 'SPIDER MAN UNLIMITED'. Below it, the series information is listed as 'SERIES · 1999 · 7+ · ENGLISH'. The average rating is 5.00 / 10. The status is 'Not Yet Started'. The genres are Action, Family, Fiction. The actors include Rino Romano, Kathleen Barr, Jim Byrnes, Garry Chalk, and Paul Dobson. The director is Patrick Archibald. The description is 'Strap in to your seats and experience Spider Man Unlimited'. A 'Similar Content' section is present, showing thumbnails for other shows: ViR, Rainbow Rangers, Z, Wally Disney Slurp Symphony, Toby Tortoise Returns, and Sonic Boom.

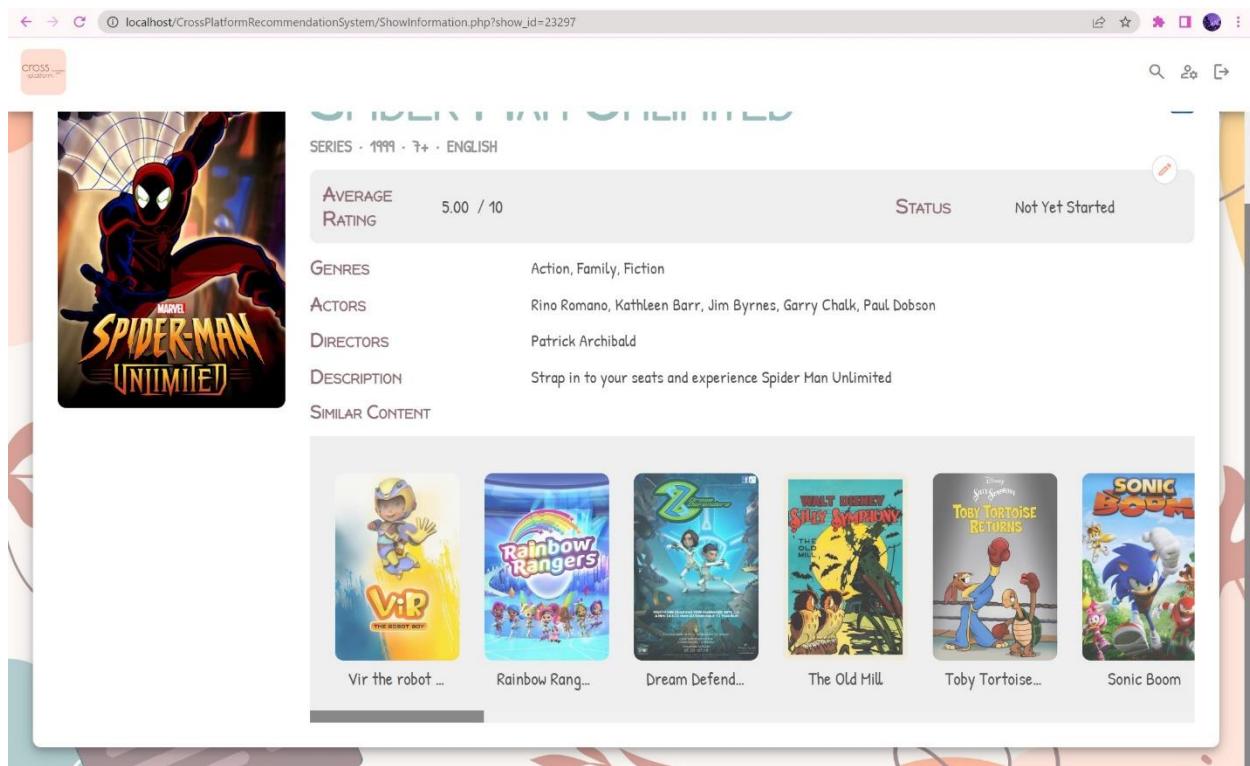
**Table 115**

<b>Test Case ID</b>	SI_02	<b>Status</b>	Pass
<b>Module Name</b>	Show Information		
<b>Test Case Title</b>	Show Information – Similar Content		
<b>Test Case Description</b>	When the user is navigated to a show's ShowInformation page, the user can see the shows that are similar to it.		
<b>Pre-conditions</b>	<b>Test Data</b>		
<ul style="list-style-type: none"> <li>Access to a browser</li> <li>The user is an existing user and is already logged in.</li> </ul>	email = cprstestuser1@gmail.com password = Hello@Test1 show_id = 23297		

<b>Test Case ID</b>	SI_02	<b>Status</b>	Pass
<b>Steps</b>	1. Navigate to the ShowInformation page with the value set to GET key show_id.		
<b>Expected Result</b>	<i>The information belonging to that show_id is displayed along with Similar Content to the user (Figure 130).</i>		

**Figure 130**

*Similar Content page for show\_id=23297*

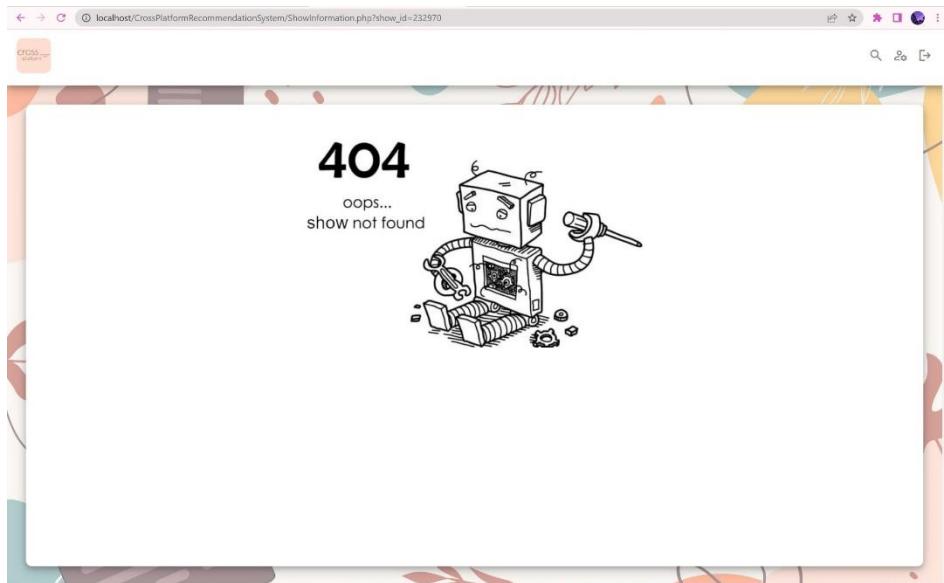


**Table 116**

<b>Test Case ID</b>	SI_03	<b>Status</b>	Pass
<b>Module Name</b>	Show Information		
<b>Test Case Title</b>	Show Information – Invalid Show ID		
<b>Test Case Description</b>	When the user is navigated to a show's ShowInformation page with an invalid show_id keyword, 404 error is displayed.		
<b>Pre-conditions</b>	<p>• Access to a browser</p> <p>• The user is an existing user and is already logged in.</p>		
<b>Test Data</b>	email = cprstestuser1@gmail.com password = Hello@Test1 show_id = 232970		
<b>Steps</b>	1. Navigate to the ShowInformation page with the value set to GET key show_id.		
<b>Expected Result</b>	Since an invalid show_id value is passed to the keyword, 404 error should be displayed (Figure 131).		
<b>Actual Result</b>	Same as the Expected Result		

**Figure 131**

*Error displayed for Show Information page for show\_id=232970*

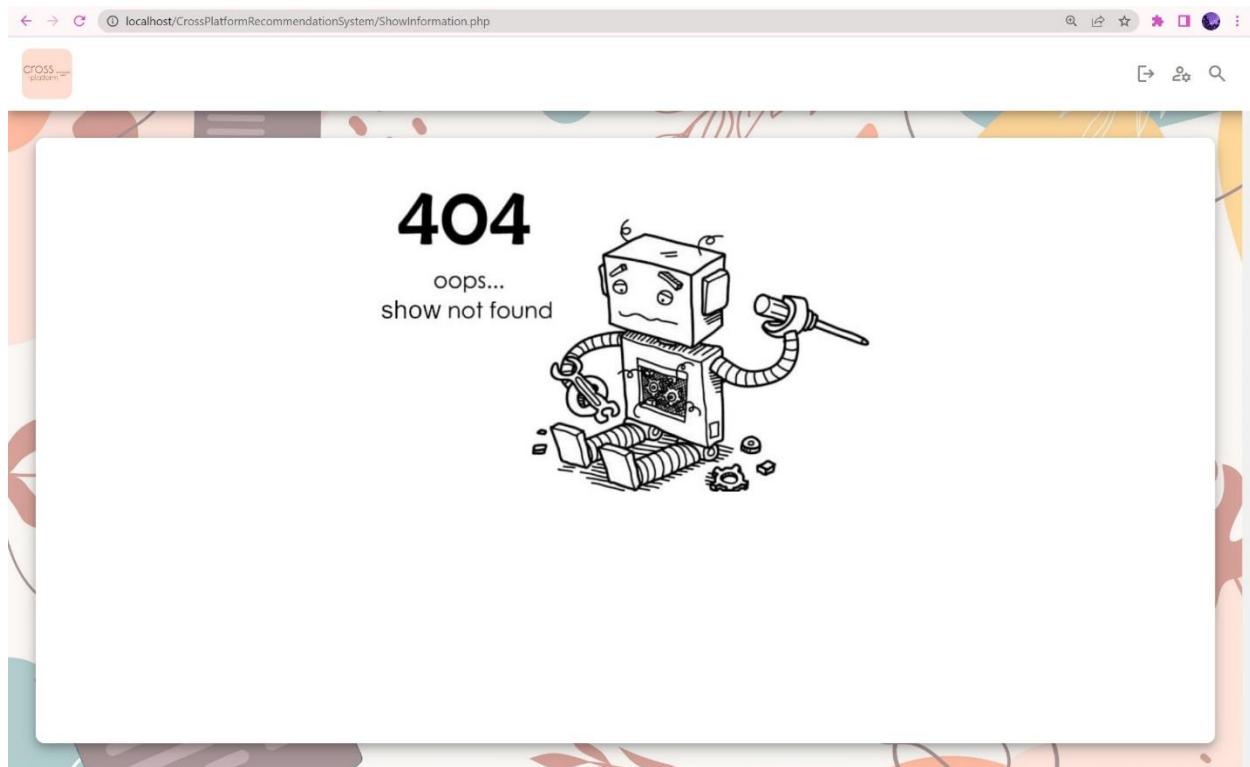
**Table 117**

<b>Test Case ID</b>	SI_04	<b>Status</b>	Pass
<b>Module Name</b>	Show Information		
<b>Test Case Title</b>	Show Information – No Show ID Passed		
<b>Test Case Description</b>	When the user is navigated to a show's ShowInformation page with no show_id keyword, 404 error is displayed.		
<b>Pre-conditions</b>	<b>Test Data</b>		
<ul style="list-style-type: none"> <li>• Access to a browser</li> <li>• The user is an existing user and is already logged in.</li> </ul>	email = cprstestuser1@gmail.com password = Hello@Test1		
<b>Steps</b>	1. Navigate to the ShowInformation page with no value set to GET key show_id.		

<b>Test Case ID</b>	SI_04	<b>Status</b>	Pass
<b>Expected Result</b>	<b>Actual Result</b>		
Since no show_id keyword is present, 404 error should be displayed (Figure 132).	Same as the Expected Result		

**Figure 132**

Error displayed for Show Information page for no show\_id keyword

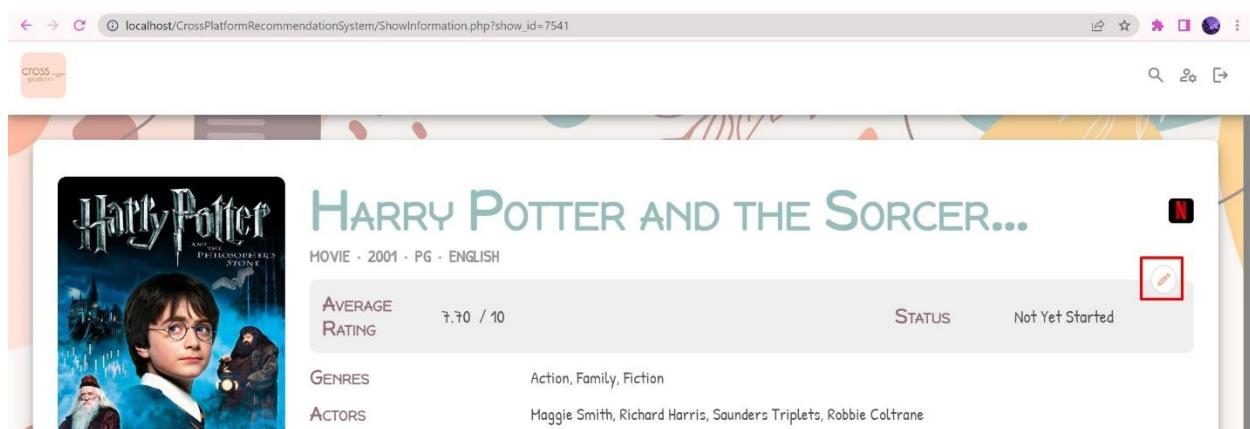
**Table 118**

<b>Test Case ID</b>	SI_05	<b>Status</b>	Pass
<b>Module Name</b>	Show Information		
<b>Test Case Title</b>	Show Information – Set Status		
<b>Test Case Description</b>	A user sets a status for a specific show in Show Information page.		

<b>Test Case ID</b>	SI_05	<b>Status</b>	Pass
<b>Pre-conditions</b>	<b>Test Data</b>		
• Access to a browser • The user is an existing user and is already logged in.	email = cprstestuser1@gmail.com password = Hello@Test1 show_id = 7541 status = Finished		
<b>Steps</b>	<ol style="list-style-type: none"> <li>1. Navigate to the ShowInformation page with the value set to GET key show_id.</li> <li>2. Click on the Edit icon near the Status (Figure 133).</li> <li>3. Change the status to the desired value (Figure 134).</li> <li>4. Click on the Save icon.</li> </ol>		
<b>Expected Result</b>	<b>Actual Result</b>		
The status for this show set by the user will be saved (Figure 135).	Same as the Expected Result		

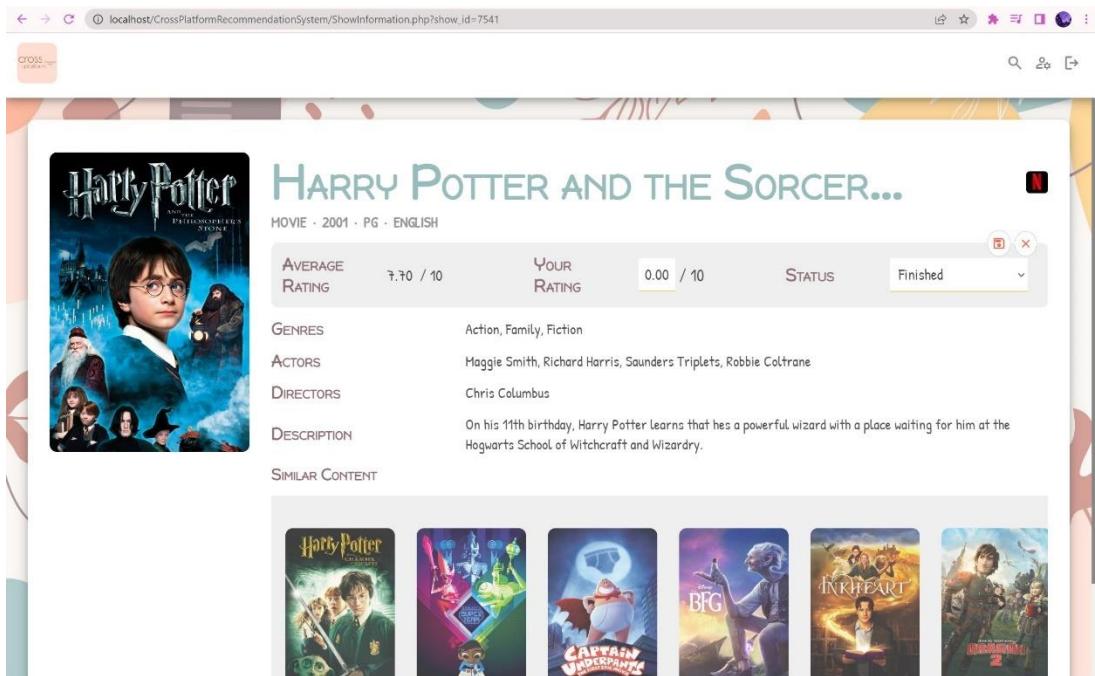
**Figure 133**

*Edit icon near the Status in ShowInformation page*

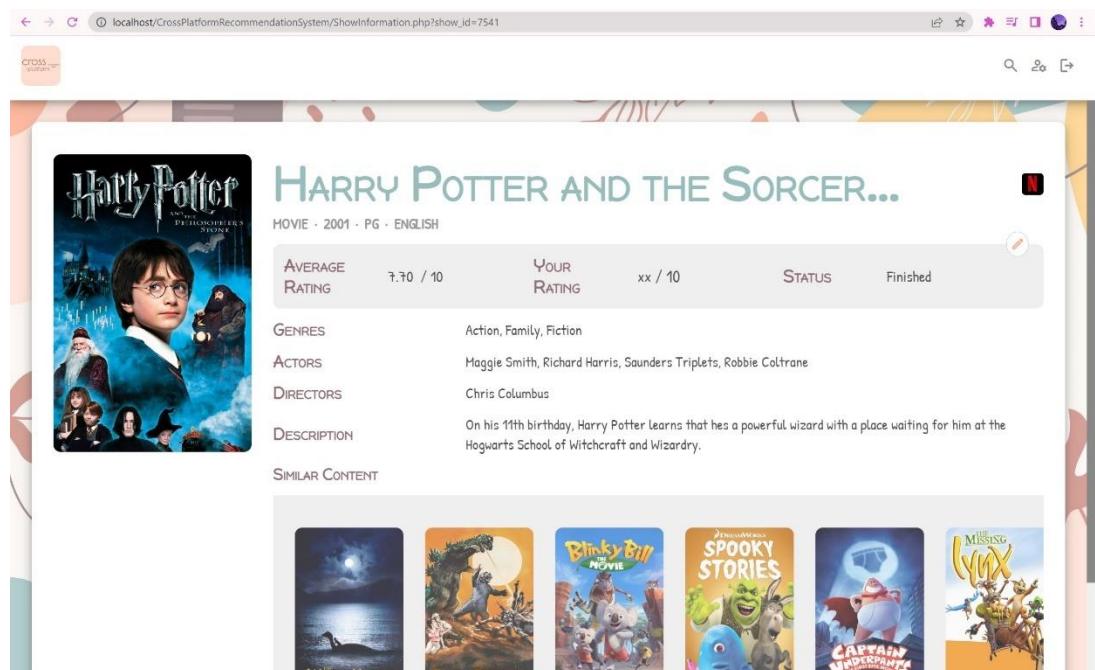


**Figure 134**

*Status edited to desired value by the user in ShowInformation page*

**Figure 135**

*Updated Status by the user for this show*

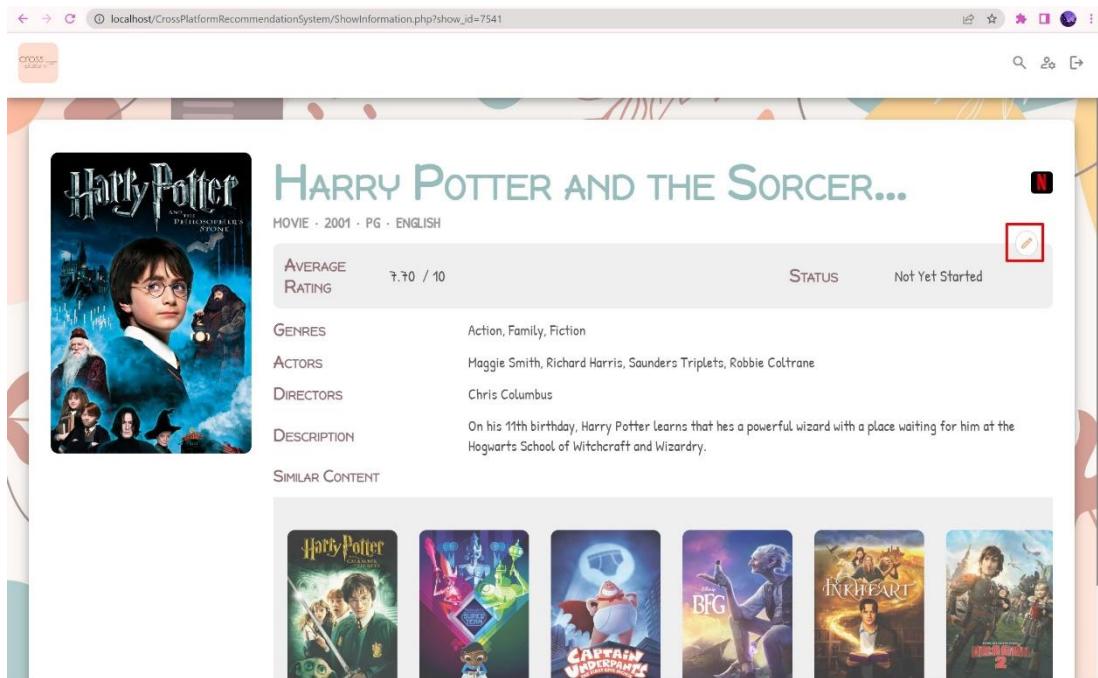


**Table 119**

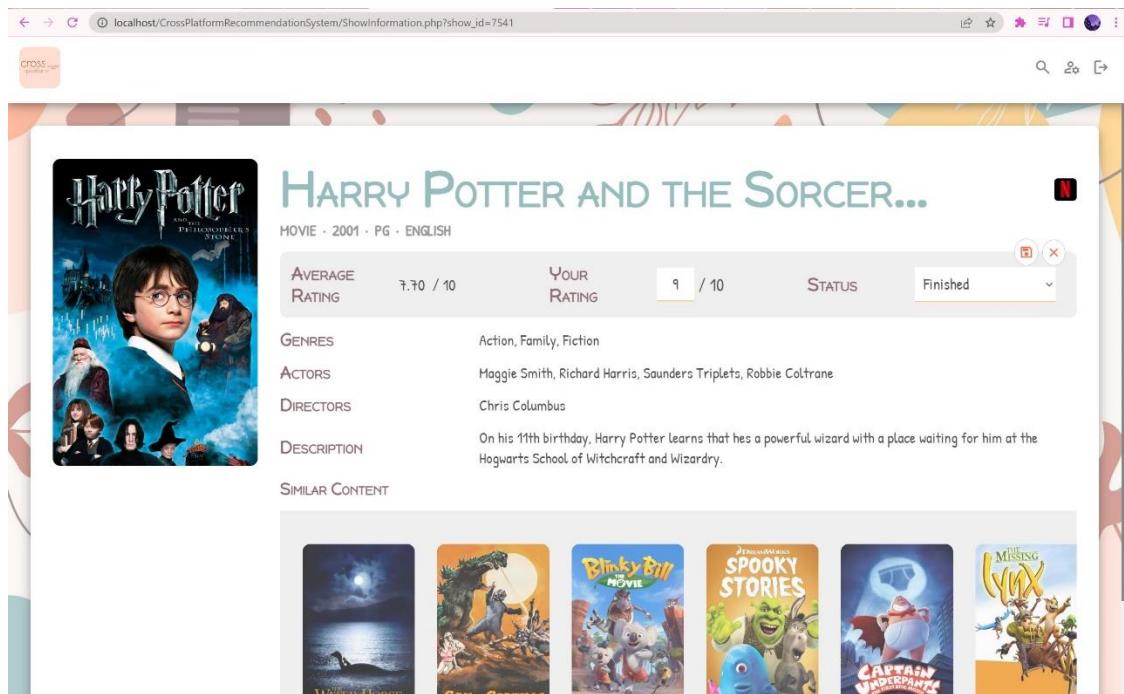
<b>Test Case ID</b>	SI_06	<b>Status</b>	Pass
<b>Module Name</b>	Show Information		
<b>Test Case Title</b>	Show Information – Set Rating		
<b>Test Case Description</b>	A user sets a rating for a specific show in Show Information page.		
<b>Pre-conditions</b>	<b>Test Data</b>		
<ul style="list-style-type: none"> <li>• Access to a browser</li> <li>• The user is an existing user and is already logged in.</li> </ul>	email = cprstestuser1@gmail.com password = Hello@Test1 show_id = 7541 rating = 9		
<b>Steps</b>	1. Navigate to the ShowInformation page with the value set to GET key show_id.  2. Click on the Edit icon near the Status (Figure 136).  3. Change the rating to the desired value between 0 and 10 (Figure 137).  4. Click on the Save icon.		
<b>Expected Result</b>	<b>Actual Result</b>		
The rating for this show set by the user will be saved (Figure 138).	Same as the Expected Result		

**Figure 136**

*Edit icon near the Status in ShowInformation page*

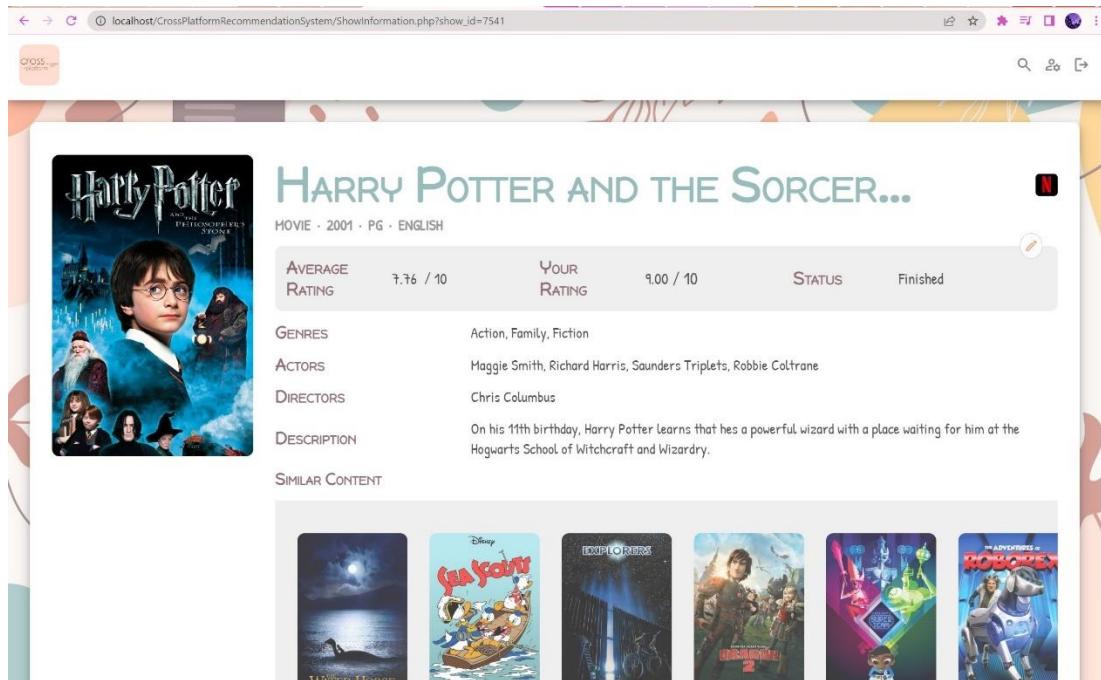
**Figure 137**

*Rating edited to desired value by the user in ShowInformation page*



**Figure 138**

*Updated Rating by the user for this show*

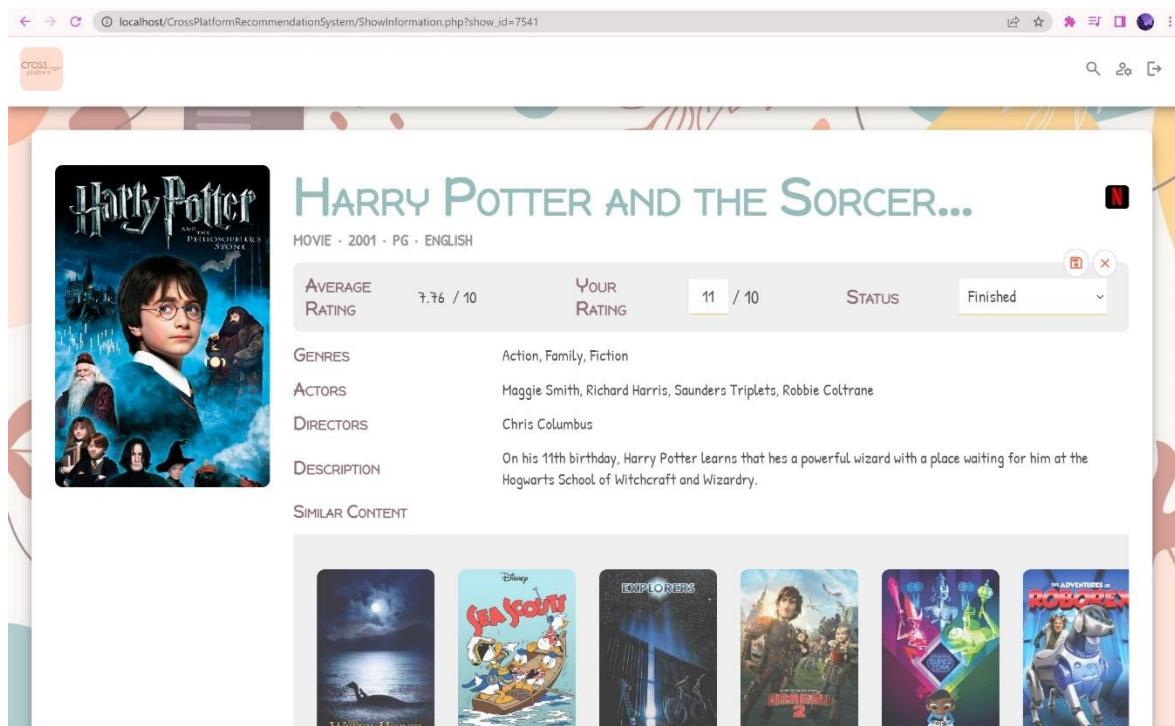
**Table 120**

<b>Test Case ID</b>	SI_07	<b>Status</b>	Pass
<b>Module Name</b>	Show Information		
<b>Test Case Title</b>	Show Information – Invalid Rating		
<b>Test Case Description</b>	When the user tries to set an invalid rating for a specific show in Show Information page, an error is displayed.		
<b>Pre-conditions</b>	<b>Test Data</b>		
<ul style="list-style-type: none"> <li>• Access to a browser</li> <li>• The user is an existing user and is already logged in.</li> </ul>	email = cprstestuser1@gmail.com  password = Hello@Test1  show_id = 7541  rating = 11		

<b>Test Case ID</b>	SI_07	<b>Status</b>	Pass
<b>Steps</b>	<ol style="list-style-type: none"> <li>1. Navigate to the ShowInformation page with the value set to GET key show_id.</li> <li>2. Click on the Edit icon near the Status.</li> <li>3. Change the rating to the desired value (Figure 139).</li> <li>4. Click on the Save icon.</li> </ol>		
<b>Expected Result</b>	<b>Actual Result</b>		
Since the rating is an invalid rating, an error will be displayed (Figure 140).	Same as the Expected Result		

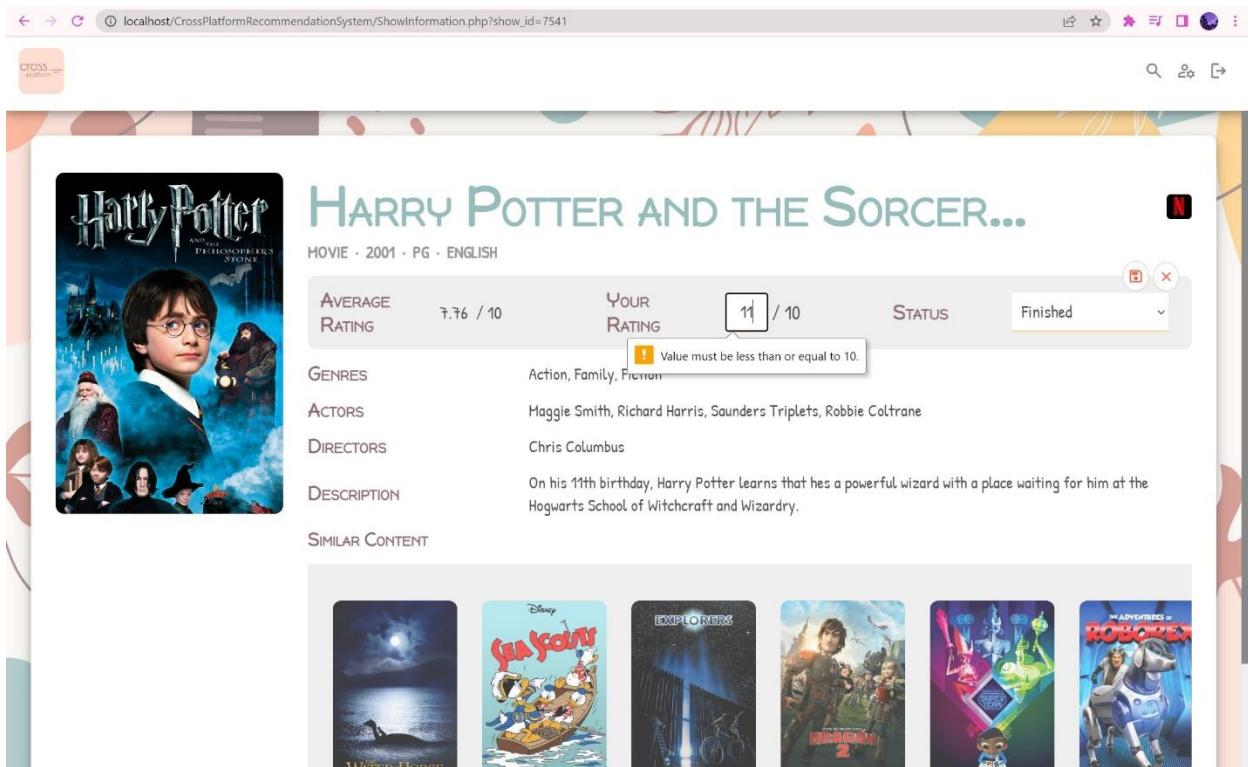
**Figure 139**

*Invalid rating entered in ShowInformation page*



**Figure 140**

*Error message displayed for the invalid rating*

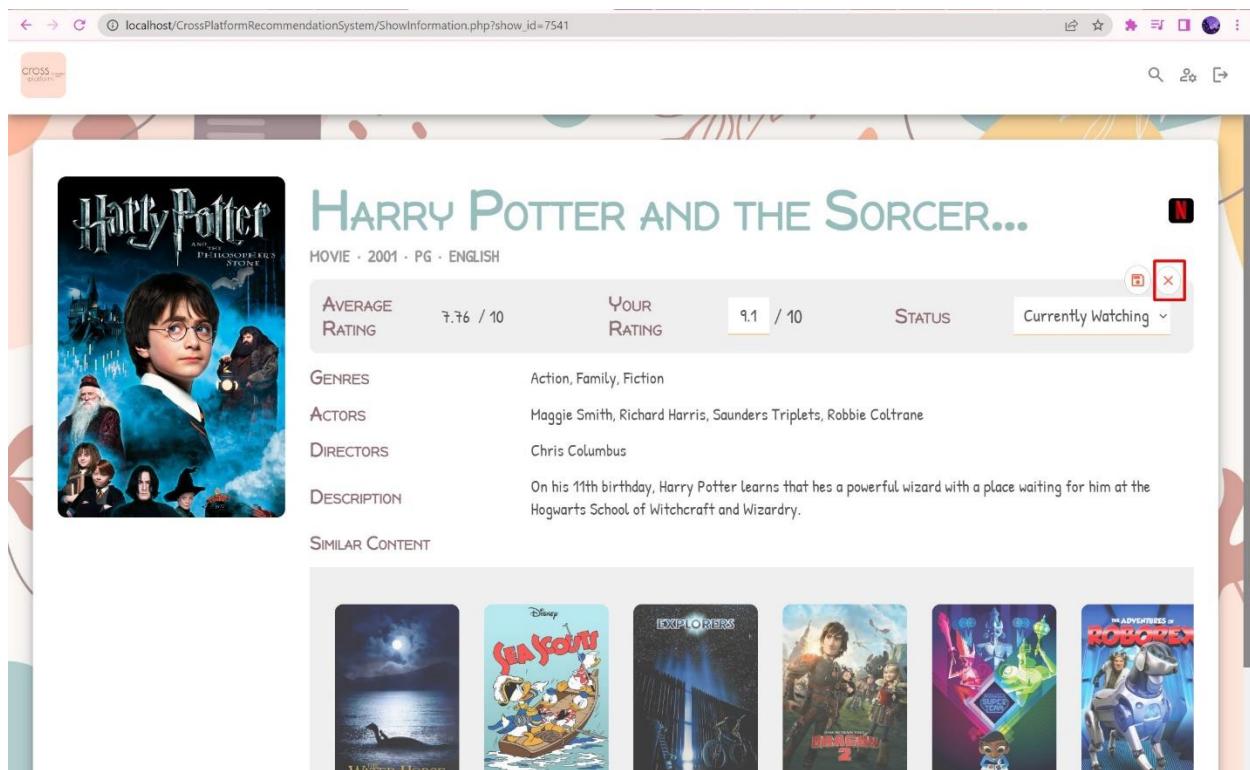
**Table 121**

<b>Test Case ID</b>	SI_08	<b>Status</b>	Pass
<b>Module Name</b>	Show Information		
<b>Test Case Title</b>	Show Information – Close Icon		
<b>Test Case Description</b>	A user can discard the edited rating or status by clicking on the Close icon.		
<b>Pre-conditions</b>	<b>Test Data</b>		
<ul style="list-style-type: none"> <li>• Access to a browser</li> <li>• The user is an existing user and is already logged in.</li> </ul>	email = cprstestuser1@gmail.com password = Hello@Test1 show_id = 7541		

<b>Test Case ID</b>	SI_08	<b>Status</b>	Pass
<b>Steps</b>	<p>5. Navigate to the ShowInformation page with the value set to GET key show_id.</p> <p>6. Click on the Edit icon near the Status.</p> <p>7. Change the rating or status to a desired value.</p> <p>8. Click on the Close icon (Figure 141).</p>		
<b>Expected Result</b>	<b>Actual Result</b>		
The new rating or status for this show edited by the user will not be saved (Figure 142).	Same as the Expected Result		

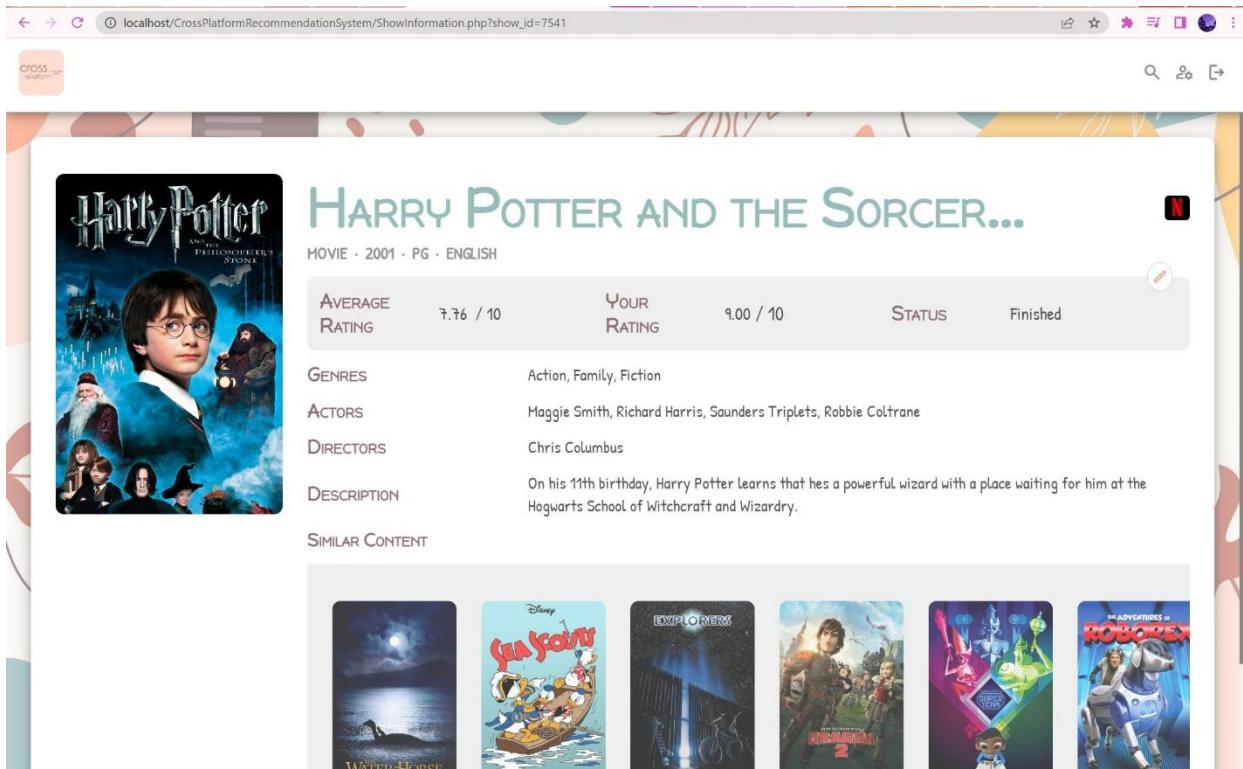
**Figure 141**

*Edited values for Rating and Status and Close icon in ShowInformation page*



**Figure 142**

ShowInformation page after clicking the Close icon

**Table 122**

<b>Test Case ID</b>	PR_01	<b>Status</b>	Pass
<b>Module Name</b>	Profile		
<b>Test Case Title</b>	Profile		
<b>Test Case Description</b>	When the user navigates to the Profile page, his personal information is displayed.		
<b>Pre-conditions</b>	<b>Test Data</b>		
<ul style="list-style-type: none"> <li>• Access to a browser</li> <li>• The user is an existing user and is already logged in.</li> </ul>	email = cprstestuser2@gmail.com password = Hello@Test2		

<b>Test Case ID</b>	PR_01	<b>Status</b>	Pass
<b>Steps</b>	1. Navigate to the Profile.		
<b>Expected Result</b>	<b>Actual Result</b>		
The personal information of the user is displayed (Figure 143).	Same as the Expected Result		

**Figure 143***Profile page*

← → ⌂ localhost/CrossPlatformRecommendationSystem/Profile.php

CROSS platform

PROFILE

NAME\*

Kamala Khan

EMAIL\*

cprtestuser2@gmail.com

PASSWORD\*

.....

LANGUAGE PREFERENCES\*

Afrikaans Akan Amharic Arabic Assamese Bangla Belarusian Cantonese Catalan Chinese Croatian Czech Danish Dutch English Finnish French Galician Georgian German Greek Gujarati Hebrew Hindi Hungarian Icelandic Indonesian Italian Japanese Kannada Khmer Korean Kurdish Latvian Lithuanian Luxembourgish Macedonian Malay Malayalam Maori Marathi Mongolian Nepali Norwegian Persian Polish Portuguese Punjabi Quechua Romanian Russian Serbian Serbo-Croatian Slovak Slovenian Spanish Swahili Swedish Tagalog Tamil

**Table 123**

<b>Test Case ID</b>	PR_02	<b>Status</b>	Pass
<b>Module Name</b>	Profile		
<b>Test Case Title</b>	Profile – Edit Details		

<b>Test Case ID</b>	PR_02	<b>Status</b>	Pass
<b>Test Case Description</b>	When the user navigates to the Profile page, his personal information can be edited.		
<b>Pre-conditions</b>	<b>Test Data</b>		
<ul style="list-style-type: none"> <li>• Access to a browser</li> <li>• The user is an existing user and is already logged in.</li> </ul>	email = cprstestuser2@gmail.com password = Hello@Test2		
<b>Steps</b>	<ol style="list-style-type: none"> <li>1. Navigate to the Profile.</li> <li>2. Edit any details in the page (Figure 144).</li> <li>3. Click on the Save button.</li> </ol>		
<b>Expected Result</b>	<b>Actual Result</b>		
The personal information of the user is updated (Figure 145).	Same as the Expected Result		

**Figure 144**

*Edited name in the Profile page*

localhost/CrossPlatformRecommendationSystem/Profile.php

cross platform

PROFILE

NAME\*

Kamala

EMAIL\*

cprstestuser2@gmail.com

PASSWORD\*

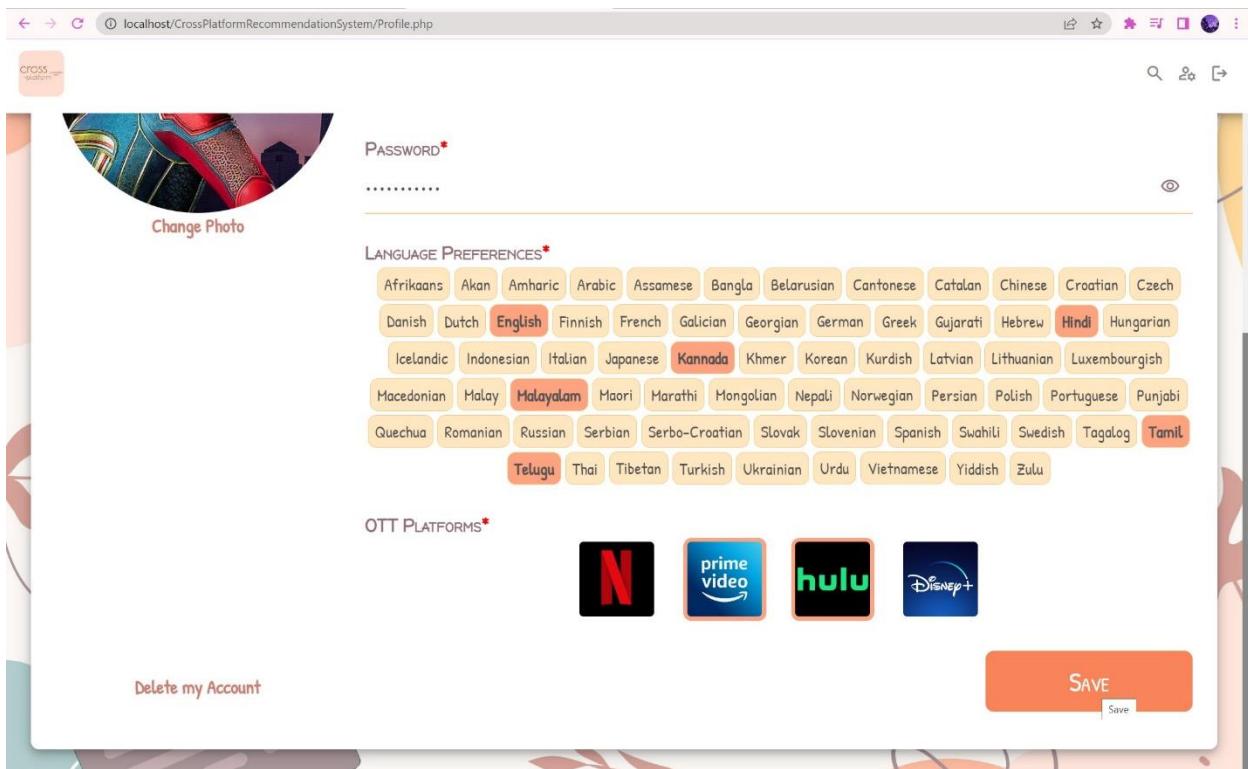
.....

LANGUAGE PREFERENCES\*

Afrikaans Akan Amharic Arabic Assamese Bangla Belarusian Cantonese Catalan Chinese Croatian Czech

**Figure 145**

*Save button in the Profile page*

**Table 124**

<b>Test Case ID</b>	PR_03	<b>Status</b>	Pass
<b>Module Name</b>	Profile		
<b>Test Case Title</b>	Profile – Change Profile Photo		
<b>Test Case Description</b>	When the user navigates to the Profile page, his profile photo can be changed.		
<b>Pre-conditions</b>	<b>Test Data</b>		
<ul style="list-style-type: none"> <li>• Access to a browser</li> <li>• The user is an existing user and is already logged in.</li> </ul>	email = cprstestuser1@gmail.com password = Hello@Test1		

<b>Test Case ID</b>	PR_03	<b>Status</b>	Pass
<b>Steps</b>	<ol style="list-style-type: none"> <li>1. Navigate to the Profile.</li> <li>2. Click on the 'Change Photo' link below the profile photo (Figure 146).</li> <li>3. Choose the new image from your computer (Figure 147).</li> <li>4. Click on the Save button.</li> </ol>		
<b>Expected Result</b>	<b>Actual Result</b>		
The profile photo of the user is updated (Figure 148).	Same as the Expected Result		

**Figure 146***Change Photo link*

localhost/CrossPlatformRecommendationSystem/Profile.php

**PROFILE**

NAME\*

Haley Dunphy

EMAIL\*

cprtestuser1@gmail.com

PASSWORD\*

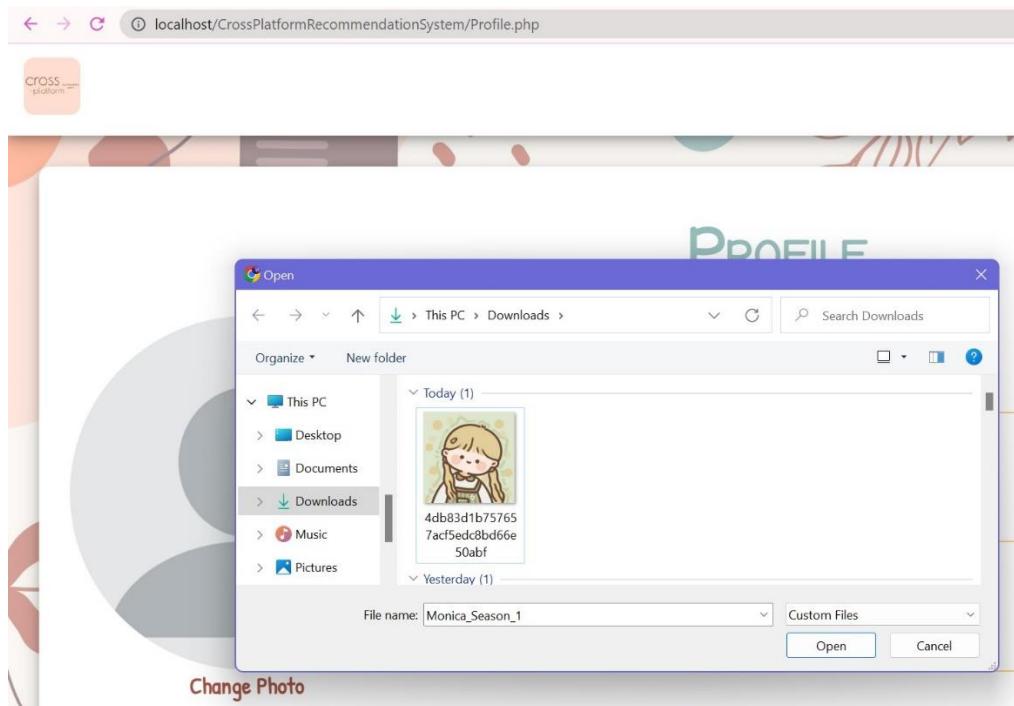
.....

LANGUAGE PREFERENCES\*

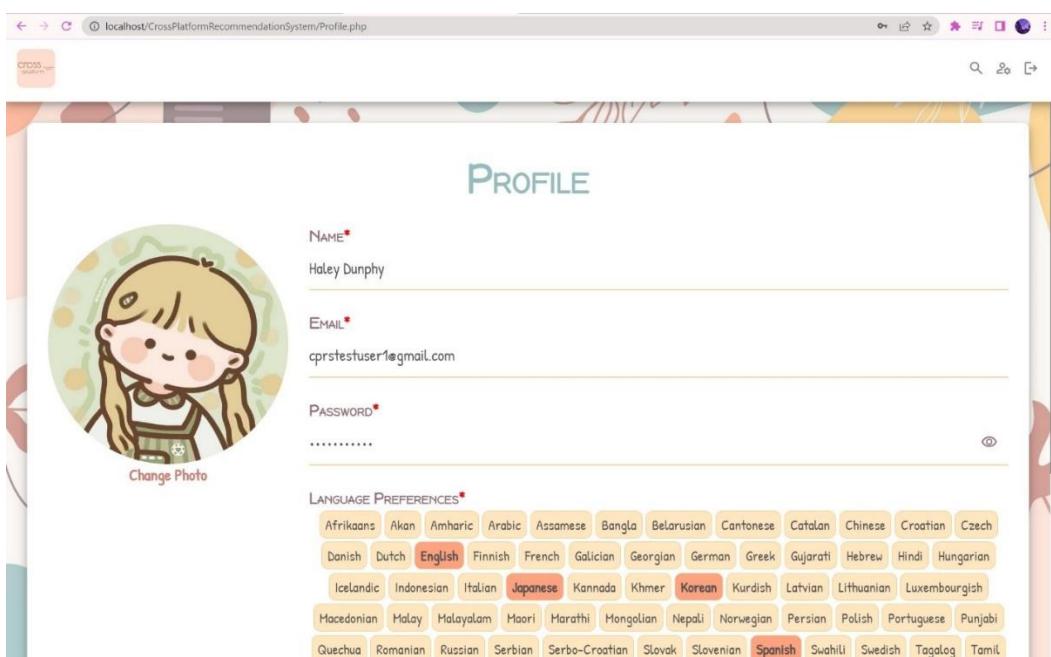
Afrikaans Akan Amharic Arabic Assamese Bangla Belarusian Cantonese Catalan Chinese Croatian Czech Danish Dutch English Finnish French Galician Georgian German Greek Gujarati Hebrew Hindi Hungarian Icelandic Indonesian Italian Japanese Kannada Khmer Korean Kurdish Latvian Lithuanian Luxembourgish Macedonian Malay Malayalam Maori Marathi Mongolian Nepali Norwegian Persian Polish Portuguese Punjabi Quechua Romanian Russian Serbian Serbo-Croatian Slovak Slovenian Spanish Swahili Swedish Tagalog Tamil

**Figure 147**

*Choose an image dialog*

**Figure 148**

*Updated Profile photo of the user in the Profile page*



**Table 125**

<b>Test Case ID</b>	PR_04	<b>Status</b>	Pass
<b>Module Name</b>	Profile		
<b>Test Case Title</b>	Profile – Invalid Details Entered		
<b>Test Case Description</b>	When the user navigates to the Profile page, if his personal information is edited to invalid values, respective errors will be displayed.		
<b>Pre-conditions</b>	<p>• Access to a browser</p> <p>• The user is an existing user and is already logged in.</p>		
<b>Steps</b>	<ol style="list-style-type: none"> <li>1. Navigate to the Profile.</li> <li>2. Edit any details in the page.</li> <li>3. Click on the Save button.</li> </ol>		
<b>Expected Result</b>	<p>Since the details are edited to invalid values, respective errors will be displayed (Figure 149).</p>		
	<b>Actual Result</b>		
	Same as the Expected Result		

**Figure 149**

*Errors displayed for the invalid edited details in the Profile page*

The screenshot shows a web browser window with the URL `localhost/CrossPlatformRecommendationSystem/Profile.php`. The page title is "PROFILE". On the left, there is a circular user icon with a cartoon character and a "Change Photo" button below it. The main form fields are as follows:

- NAME\***: Haley Dunphy " (with validation error message: "① Name can contain only letters, digits, spaces. Maximum of 100 characters is allowed.")
- EMAIL\***: cprstestuser1@gmail.com
- PASSWORD\***: ..... (with validation message: "① Password can contain only letters, digits, @, \$, \_. Minimum of 5 to a maximum of 30 characters is allowed.")
- LANGUAGE PREFERENCES\***: A grid of language names in orange rounded rectangles, including Afrikaans, Akan, Amharic, Arabic, Assamese, Bangla, Belarusian, Cantonese, Catalan, Chinese, Croatian, Czech, Danish, Dutch, English, Finnish, French, Galician, Georgian, German, Greek, Gujarati, Hebrew, Hindi, Hungarian, Icelandic, Indonesian, Italian, Japanese, Kannada, Khmer, Korean, Kurdish, Latvian, Lithuanian, and Luxembourgish.

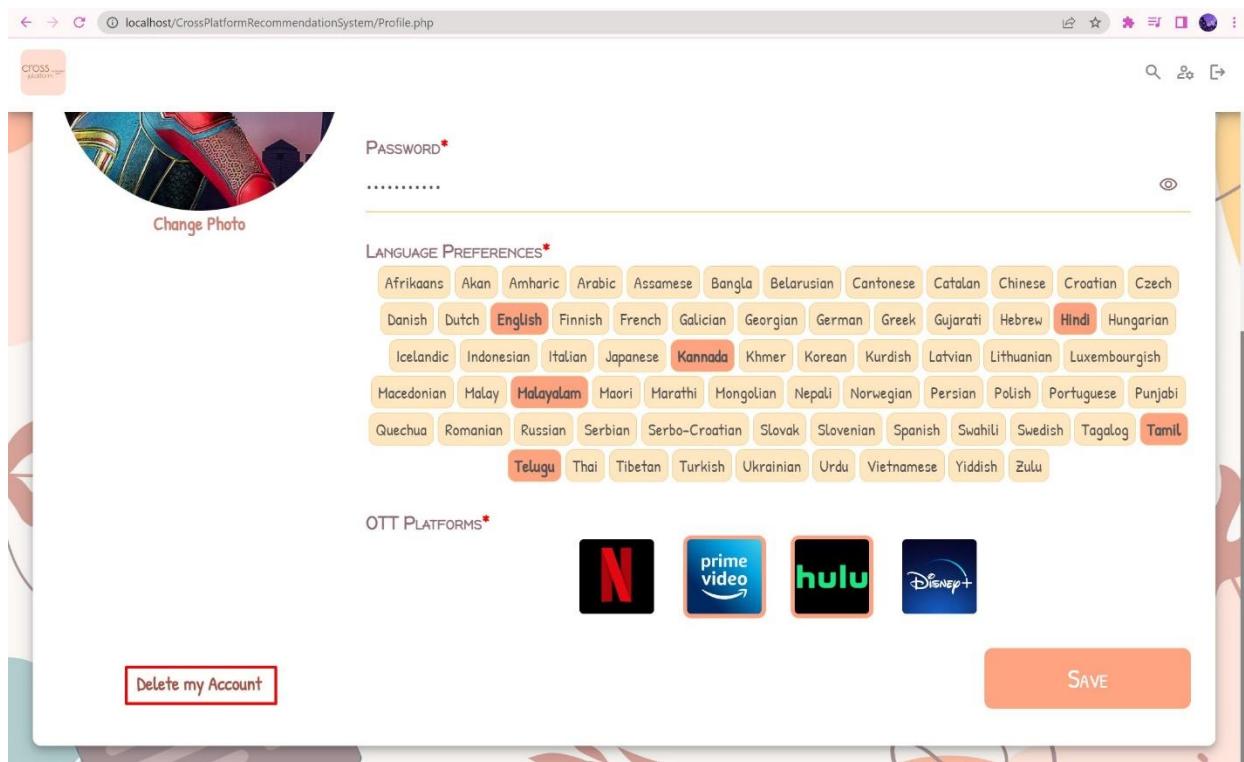
**Table 126**

<b>Test Case ID</b>	PR_05	<b>Status</b>	Pass
<b>Module Name</b>	Profile		
<b>Test Case Title</b>	Profile – Delete User Account		
<b>Test Case Description</b>	The user can delete his account when he clicks on the 'Delete My Account' link on the Profile Page.		
<b>Pre-conditions</b>	<b>Test Data</b>		
<ul style="list-style-type: none"> <li>Access to a browser</li> <li>The user is an existing user and is already logged in.</li> </ul>	email = cprstestuser2@gmail.com password = Hello@Test2		

<b>Test Case ID</b>	PR_05	<b>Status</b>	Pass
<b>Steps</b>	<ol style="list-style-type: none"> <li>1. Navigate to the Profile.</li> <li>2. Click on the 'Delete my Account' link (Figure 150).</li> <li>3. Enter the password (Figure 151).</li> <li>4. Click on the 'Delete Account' button.</li> </ol>		
<b>Expected Result</b>	<b>Actual Result</b>		
The user's account will be deleted.	Same as the Expected Result		

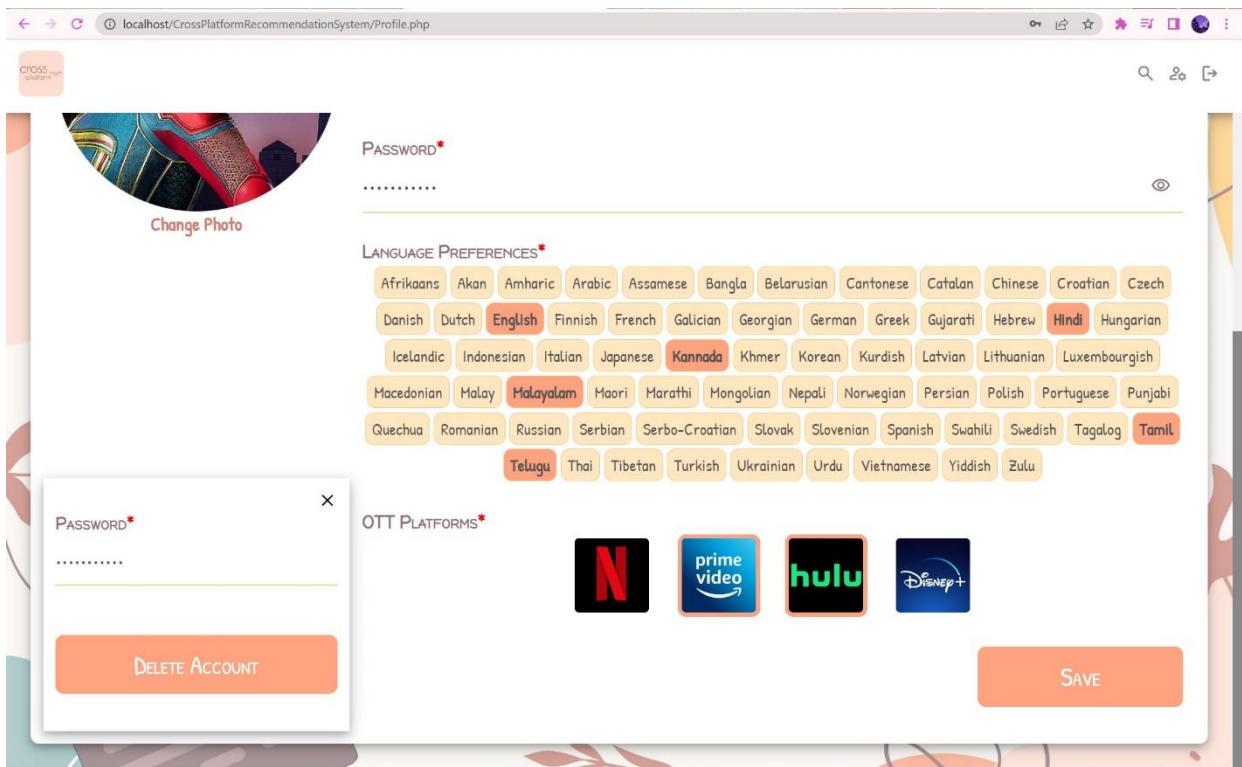
**Figure 150**

'Delete my Account' link on the Profile page



**Figure 151**

*Password entered by the user to delete his account*

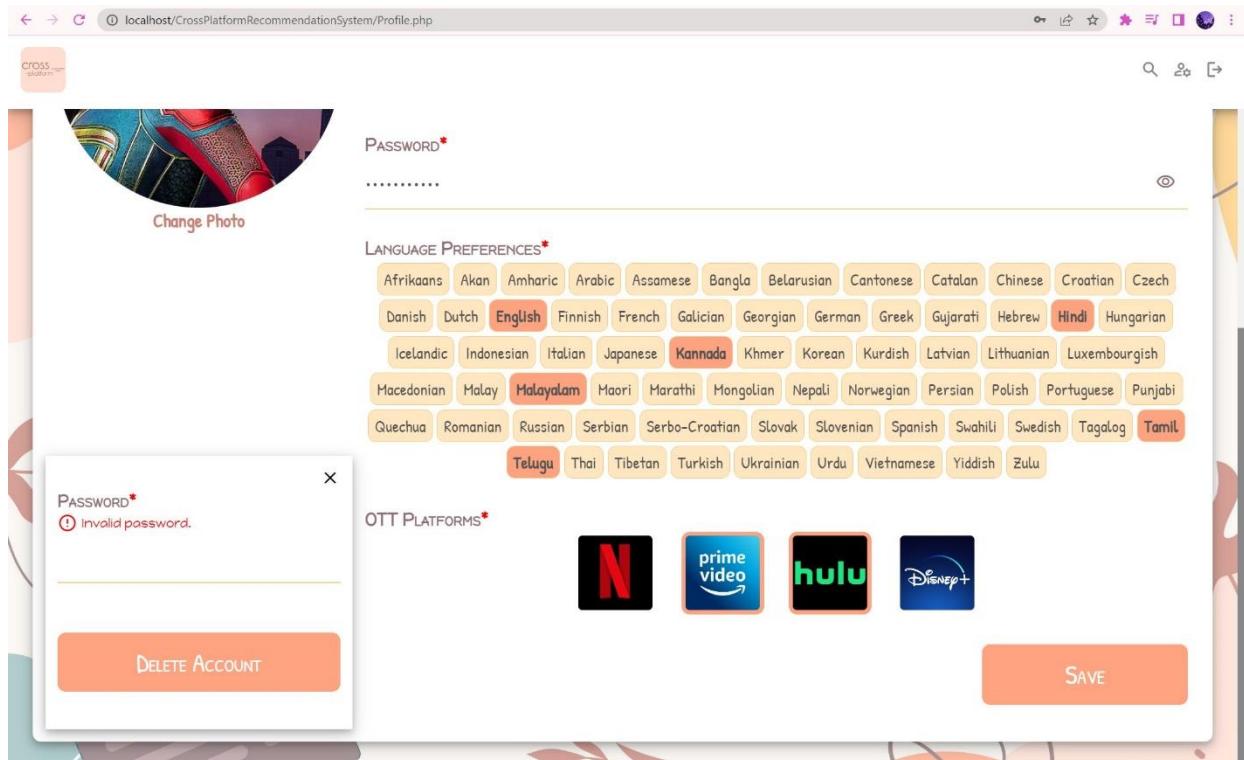
**Table 127**

<b>Test Case ID</b>	PR_06	<b>Status</b>	Pass
<b>Module Name</b>	Profile		
<b>Test Case Title</b>	Profile – Delete User, Invalid Password		
<b>Test Case Description</b>	When the user enters an invalid password to delete his, error message will be displayed.		
<b>Pre-conditions</b>	<b>Test Data</b>		
<ul style="list-style-type: none"> <li>• Access to a browser</li> <li>• The user is an existing user and is already logged in.</li> </ul>	email = cprstestuser2@gmail.com password = Hello@Test2		

<b>Test Case ID</b>	PR_06	<b>Status</b>	Pass
<b>Steps</b>	<p>5. Navigate to the Profile.</p> <p>6. Click on the 'Delete my Account' link.</p> <p>7. Enter the password.</p> <p>8. Click on the 'Delete Account' button.</p>		
<b>Expected Result</b>	<b>Actual Result</b>		
Click the user enters an invalid password, an error message will be displayed (Figure 152).	Same as the Expected Result		

**Figure 152**

*Error displayed for the invalid password entered for 'Delete my Account'.*

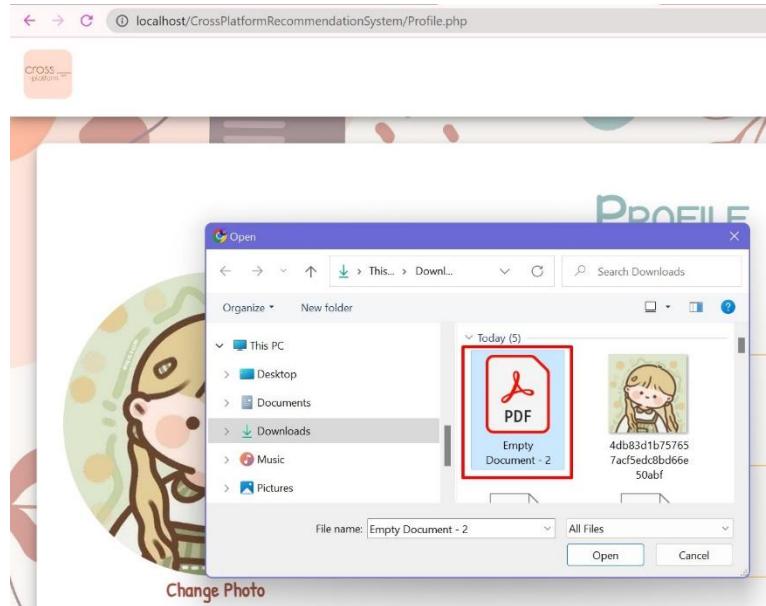


**Table 128**

<b>Test Case ID</b>	PR_07	<b>Status</b>	Pass
<b>Module Name</b>	Profile		
<b>Test Case Title</b>	Profile – Invalid File Selected		
<b>Test Case Description</b>	When the user selects an invalid file to update his profile , an error message will be displayed.		
<b>Pre-conditions</b>	<ul style="list-style-type: none"> <li>• Access to a browser</li> <li>• The user is an existing user and is already logged in.</li> </ul>		
<b>Steps</b>	<ol style="list-style-type: none"> <li>1. Navigate to the Profile.</li> <li>2. Click on the ‘Change Photo’ link below the profile photo.</li> <li>3. Choose a file from your computer (Figure 153).</li> <li>4. Click on the Save button.</li> </ol>		
<b>Expected Result</b>	<b>Actual Result</b>		
Since the user selects an invalid file for the profile photo, an error is displayed (Figure 154).	Same as the Expected Result		

**Figure 153**

*Choose an image dialog*

**Figure 154**

*Error message for selecting an invalid file as Profile photo*

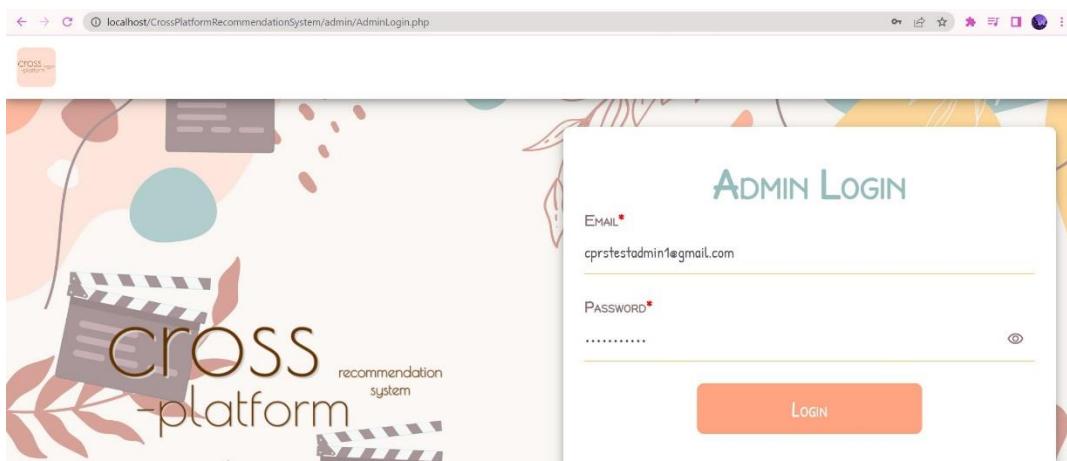
A screenshot of a 'PROFILE' edit page from 'localhost/CrossPlatformRecommendationSystem/Profile.php'. The page includes fields for NAME\*, EMAIL\*, and PASSWORD\*. A circular profile picture placeholder shows a cartoon girl. Below it, a 'Change Photo' button is visible. A red error message at the bottom left states: 'Please upload a jpeg,jpg, or png image file.' To the right, there's a 'LANGUAGE PREFERENCES\*' section with a grid of language names: Afrikaans, Akan, Amharic, Arabic, Assamese, Bangla, Belarusian, Cantonese, Catalan, Chinese, Croatian, Czech, Danish, Dutch, English, Finnish, French, Galician, Georgian, German, Greek, Gujarati, Hebrew, Hindi, Hungarian, Icelandic, Indonesian, Italian, Japanese, Kannada, Khmer, Korean, Kurdish, Latvian, Lithuanian, Luxembourgish, Macedonian, Malay, Malayalam, Maori, Marathi, Mongolian, Nepali, Norwegian, Persian, Polish, Portuguese, Punjabi, Quechua, Romanian, Russian, Serbian, Serbo-Croatian, Slovak, Slovenian, Spanish, Swahili, Swedish, Tagalog, Tamil. The 'English' button is highlighted with a red border.

**Table 129**

<b>Test Case ID</b>	AL_01	<b>Status</b>	Pass
<b>Module Name</b>	Admin Login		
<b>Test Case Title</b>	Admin Login – Valid Credentials		
<b>Test Case Description</b>	When the admin enters valid credentials, the admin can log in.		
<b>Pre-conditions</b>	<b>Test Data</b>		
Access to a browser	email = cprstestadmin1@gmail.com password = Hello@Admin		
<b>Steps</b>	<ol style="list-style-type: none"> <li>1. Navigate to the AdminLogin page.</li> <li>2. Enter the email and password (Figure 155).</li> <li>3. Click on the Login button.</li> </ol>		
<b>Expected Result</b>	<b>Actual Result</b>		
The admin is logged in and is directed to ViewAllRatings page (Figure 156).	Same as the Expected Result		

**Figure 155**

*Login page with admin-entered credentials*



**Figure 156**

New user directed to ViewAllRatings page

User ID	Content ID	Rating	Rated At
10	7541	9	2022-11-28 16:14:31
11	9098	9	2022-11-28 14:27:15
10	1962	7	2022-11-28 14:00:53
11	8771	8	2022-11-28 11:27:04
8	5668	5	2022-11-28 09:50:18
8	8771	9	2022-11-28 09:50:18
8	9365	9	2022-11-28 09:50:18
8	22144	7	2022-11-28 09:50:18
9	6473	6	2022-11-28 09:36:13
9	5925	8.6	2022-11-28 09:35:48
9	2843	6	2022-11-28 09:31:26

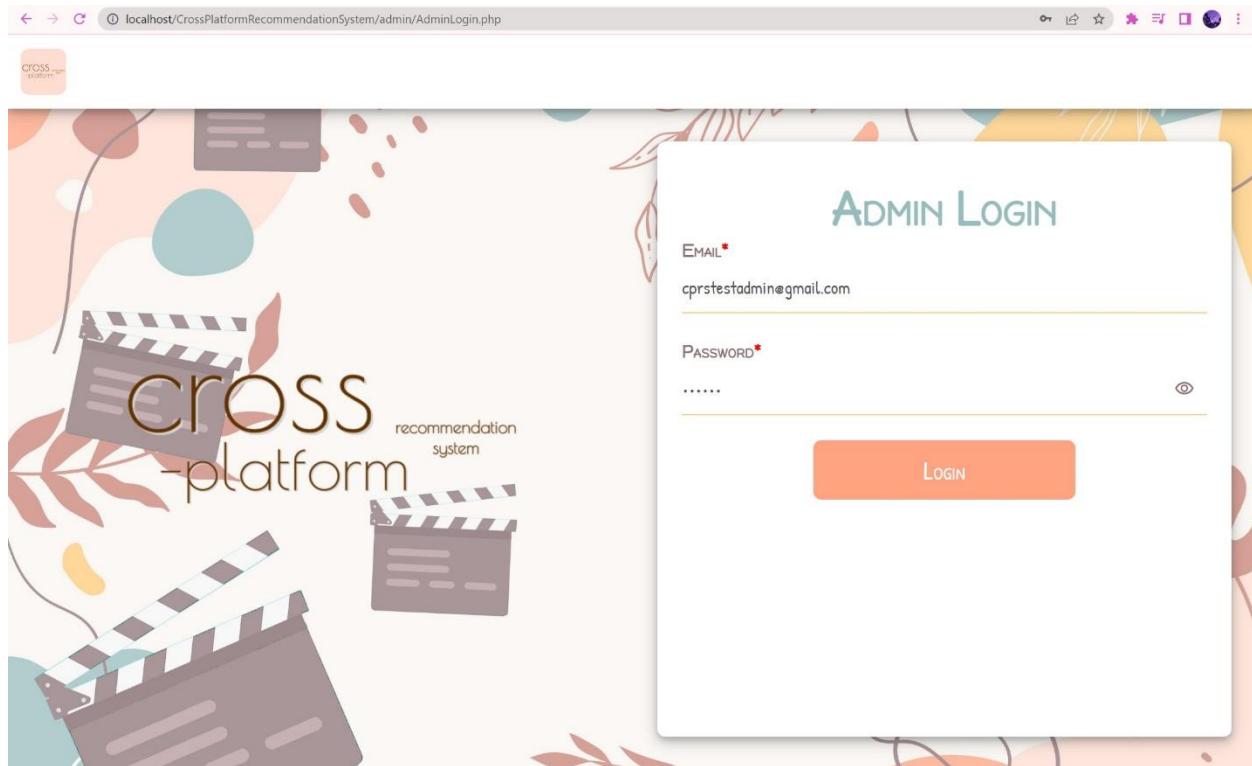
**Table 130**

<b>Test Case ID</b>	AL_02	<b>Status</b>	Pass
<b>Module Name</b>	Login		
<b>Test Case Title</b>	Admin Login – Invalid Credentials		
<b>Test Case Description</b>	When the admin enters an invalid email or password, the admin should see the error message.		
<b>Pre-conditions</b>	<b>Test Data</b>		
Access to a browser	email = cprstestadmin@gmail.com password = Helloo		

<b>Test Case ID</b>	AL_02	<b>Status</b>	Pass
<b>Steps</b>	<ol style="list-style-type: none"> <li>1. Navigate to the AdminLogin page.</li> <li>2. Enter the email and password (Figure 157).</li> <li>3. Click on the Login button.</li> </ol>		
<b>Expected Result</b>	<p>Since the admin entered an invalid email, the admin should see 'No account exists with this email' message (Figure 158).</p>		

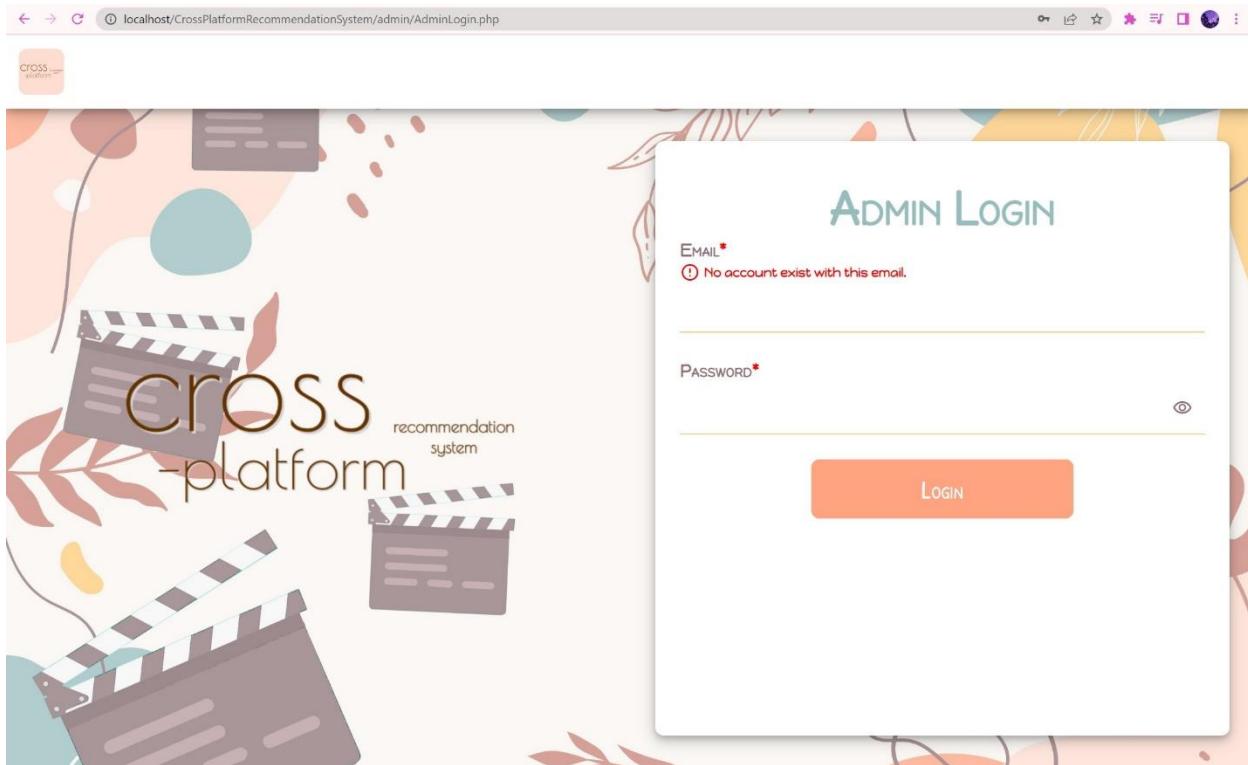
**Figure 157**

*Invalid credentials entered by the admin*



**Figure 158**

*Error displayed for invalid email entered by the admin*

**Table 131**

<b>Test Case ID</b>	AL_03	<b>Status</b>	Pass
<b>Module Name</b>	Admin Login		
<b>Test Case Title</b>	Admin Login – Admin Already Logged In		
<b>Test Case Description</b>	When the admin is already logged in but navigates to the AdminLogin page, the admin is redirected to the ViewAllRatings page.		
<b>Pre-conditions</b>	<b>Test Data</b>		
<ul style="list-style-type: none"> <li>• Access to a browser</li> <li>• The admin is already logged in.</li> </ul>	email = cprstestadmin1@gmail.com password = Hello@Admin		

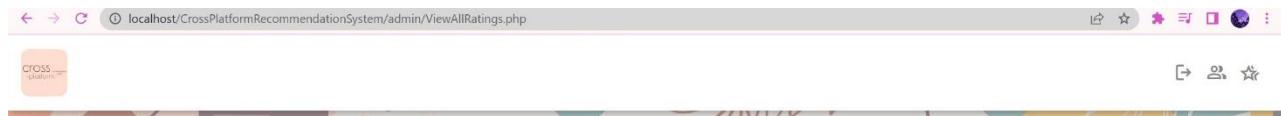
<b>Test Case ID</b>	AL_03	<b>Status</b>	Pass
<b>Steps</b>	1. Navigate to the AdminLogin page (Figure 159).		
<b>Expected Result</b>	<b>Actual Result</b>		
The admin should be navigated to the ViewAllRatings page (Figure 160).	Same as the Expected Result		

**Figure 159**

*Admin trying to navigate to the AdminLogin page*

**Figure 160**

*Admin is navigated to the ViewAllRatings page*

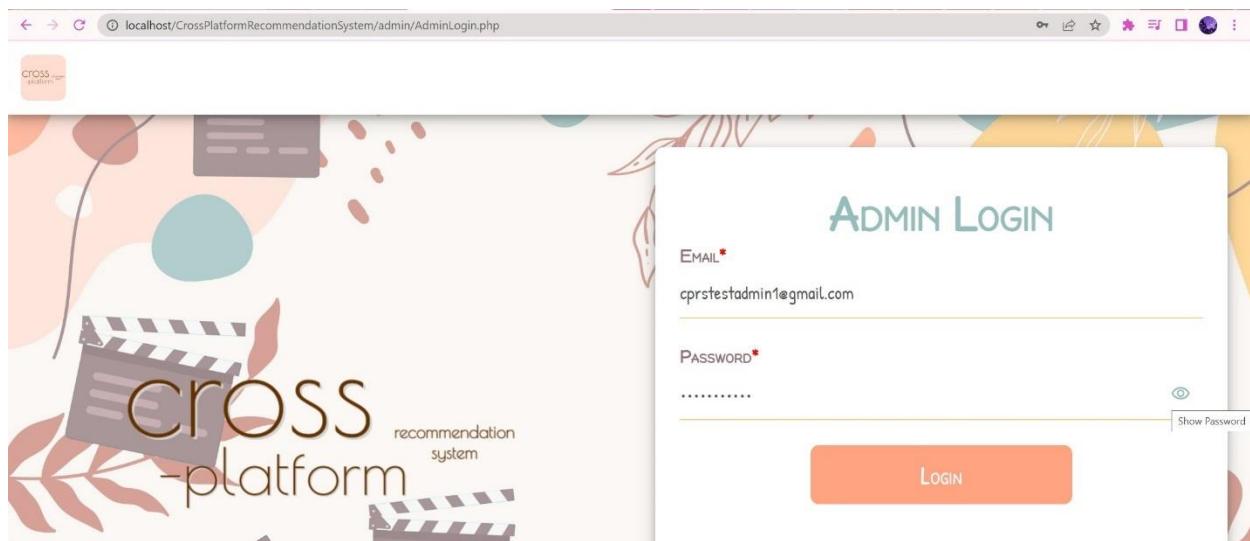
**Table 132**

<b>Test Case ID</b>	AL_04	<b>Status</b>	Pass
<b>Module Name</b>	Admin Login		
<b>Test Case Title</b>	Admin Login – Toggle Show Password		
<b>Test Case Description</b>	When the admin clicks on the Show/Hide Password button, the password should be shown or hidden.		
<b>Pre-conditions</b>	<b>Test Data</b>		
• Access to a browser	email = cprstestadmin1@gmail.com password = Hello@Test		

<b>Test Case ID</b>	AL_04	<b>Status</b>	Pass
<b>Steps</b>	<ol style="list-style-type: none"> <li>1. Navigate to the AdminLogin page.</li> <li>2. Enter the email and password.</li> <li>3. Click on the Show Password button (Figure 161).</li> <li>4. Again, click on the Hide Password button (Figure 162).</li> </ol>		
<b>Expected Result</b>	<p>When the Show Password is clicked in step 3, the password is visible to the admin (Figure 163).</p> <p>When the Hide Password is clicked in step 4, the password is masked.</p>		

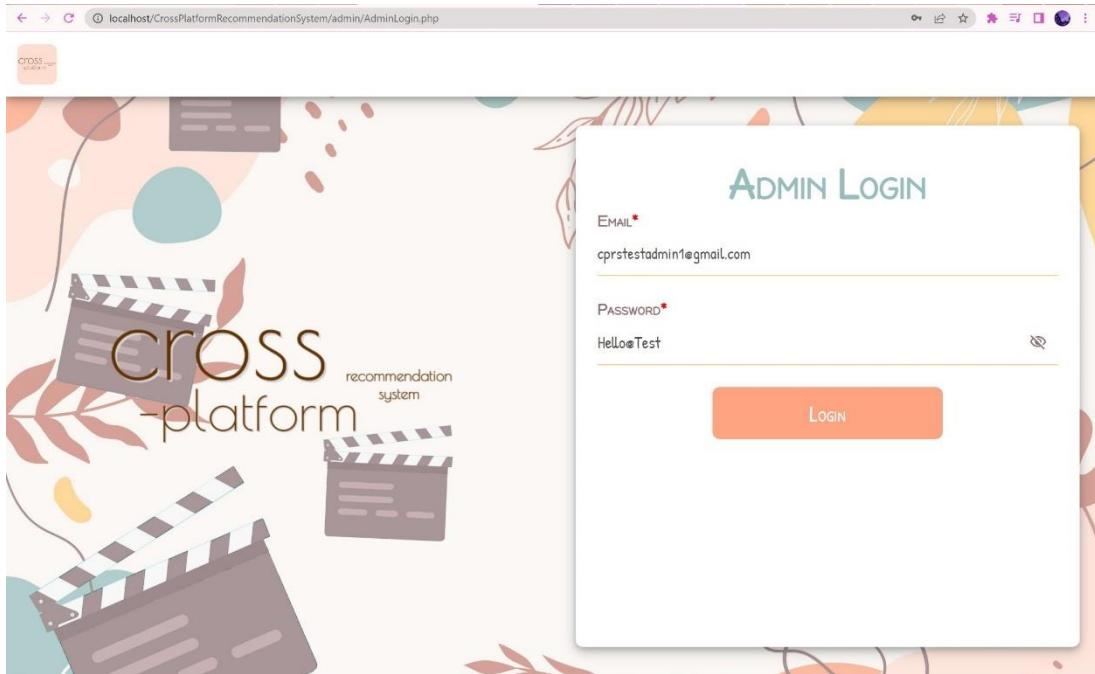
**Figure 161**

*Show Password button*

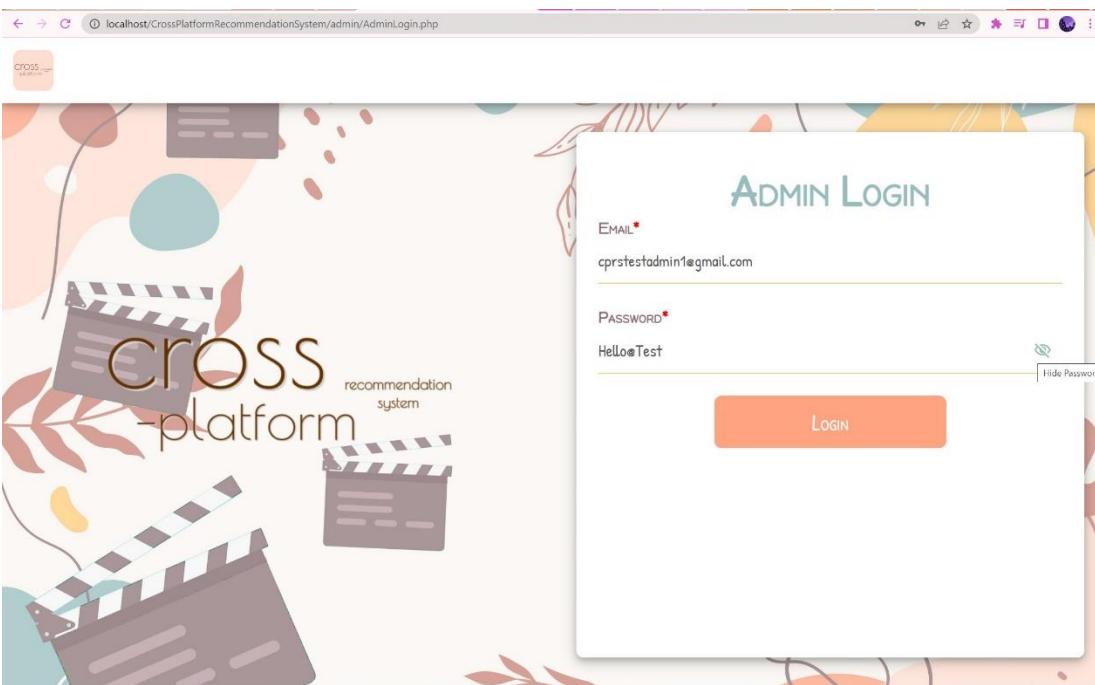


**Figure 162**

*Password is visible after clicking on the Show password button*

**Figure 163**

*Hide Password button*



**Table 133**

<b>Test Case ID</b>	VR_01	<b>Status</b>	Pass
<b>Module Name</b>	View All Ratings		
<b>Test Case Title</b>	View All Ratings		
<b>Test Case Description</b>	When the admin is navigated to the ViewAllRatings page, he can see all the ratings provided by all the users.		
<b>Pre-conditions</b>	<b>Test Data</b>		
<ul style="list-style-type: none"> <li>• Access to a browser</li> <li>• The admin is logged in</li> </ul>	email = cprstestadmin1@gmail.com password = Hello@Admin		
<b>Steps</b>	1. Navigate to the ViewAllRatings page.		
<b>Expected Result</b>	<b>Actual Result</b>		
All the ratings provided by all the users are displayed (Figure 164).	Same as the Expected Result		

**Figure 164**

*ViewAllRatings page*

User ID	Content ID	Rating	Rated At
10	7541	9	2022-11-28 16:14:31
11	9098	9	2022-11-28 14:27:15
10	1962	7	2022-11-28 14:00:53
11	8771	8	2022-11-28 11:27:04
8	5668	5	2022-11-28 09:50:18
8	8771	9	2022-11-28 09:50:18
8	9365	9	2022-11-28 09:50:18
8	22144	7	2022-11-28 09:50:18
9	6473	6	2022-11-28 09:36:13
9	5925	8.6	2022-11-28 09:35:48
9	2843	6	2022-11-28 09:31:26

**Table 134**

<b>Test Case ID</b>	VR_02	<b>Status</b>	Pass
<b>Module Name</b>	View All Ratings		
<b>Test Case Title</b>	View All Ratings – Remove a Rating		
<b>Test Case Description</b>	The admin can delete any rating by clicking on the Delete button.		
<b>Pre-conditions</b>	<b>Test Data</b>		
<ul style="list-style-type: none"> <li>• Access to a browser</li> <li>• The admin is logged in</li> </ul>	email = cprstestadmin1@gmail.com password = Hello@Admin userID = 11 contentID = 9098		
<b>Steps</b>	1. Navigate to the ViewAllRatings page.  2. Clicks on the Delete icon for a specific rating (Figure 165).		
<b>Expected Result</b>	<b>Actual Result</b>		
The rating is deleted and the remaining ratings are displayed (Figure 166).	Same as the Expected Result		

**Figure 165**

Delete button in ViewAllRatings page



**Figure 166**

*Updated ratings in ViewAllRatings page*

User ID	Content ID	Rating	Rated At
10	7541	9	2022-11-28 16:14:31
10	1962	7	2022-11-28 14:00:53
11	8771	8	2022-11-28 11:27:04
8	5668	5	2022-11-28 09:50:18
8	8771	9	2022-11-28 09:50:18
8	22144	7	2022-11-28 09:50:18
8	9365	9	2022-11-28 09:50:18
9	6473	6	2022-11-28 09:36:13
9	5925	8.6	2022-11-28 09:35:48
9	2843	6	2022-11-28 09:31:26
9	1962	7	2022-11-28 09:30:01

**Table 135**

<b>Test Case ID</b>	VU_01	<b>Status</b>	Pass
<b>Module Name</b>	View All Users		
<b>Test Case Title</b>	View All Users		
<b>Test Case Description</b>	When the admin is navigated to the ViewAllUsers page, he can see all the ratings provided by all the users.		
<b>Pre-conditions</b>	<b>Test Data</b>		
• Access to a browser	email = cprstestadmin1@gmail.com		
• The admin is logged in	password = Hello@Admin		
<b>Steps</b>	1. Navigate to the ViewAllUsers page.		

<b>Test Case ID</b>	VU_01	<b>Status</b>	Pass
<b>Expected Result</b>	<b>Actual Result</b>		
All the registered users are displayed (Figure 167).	Same as the Expected Result		

**Figure 167**

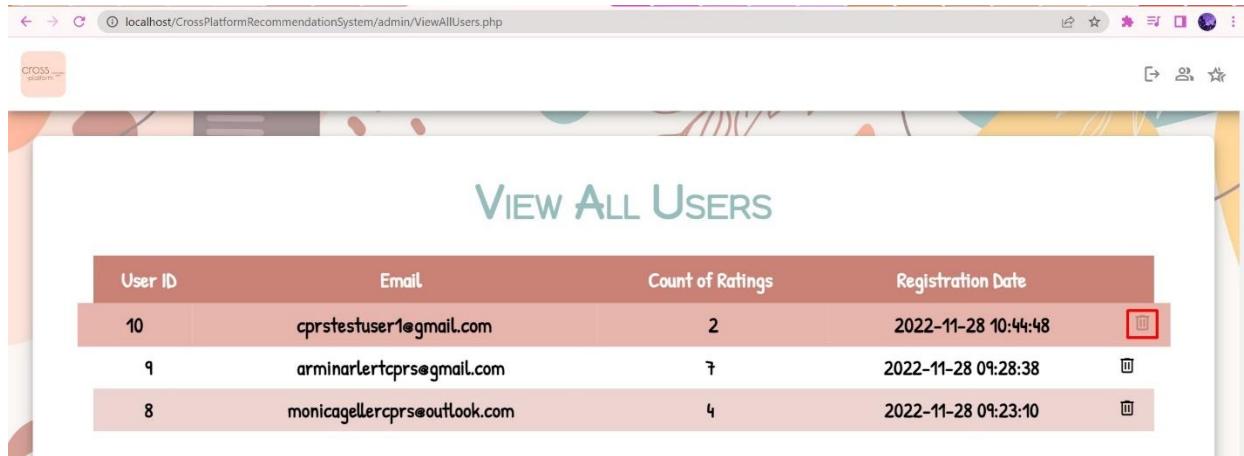
*ViewAllUsers page*

User ID	Email	Count of Ratings	Registration Date	Delete
10	cprtestuser1@gmail.com	2	2022-11-28 10:44:48	
9	arminarlertcprs@gmail.com	7	2022-11-28 09:28:38	
8	monicagellercprs@outlook.com	4	2022-11-28 09:23:10	

**Table 136**

<b>Test Case ID</b>	VU_02	<b>Status</b>	Pass
<b>Module Name</b>	View All Users		
<b>Test Case Title</b>	View All Users – Remove a User		
<b>Test Case Description</b>	The admin can delete any user by clicking on the Delete icon.		

<b>Test Case ID</b>	VU_02	<b>Status</b>	Pass
<b>Pre-conditions</b>	<b>Test Data</b>		
• Access to a browser • The admin is logged in	email = cprstestadmin1@gmail.com password = Hello@Admin userID = 10		
<b>Steps</b>	<ol style="list-style-type: none"> <li>1. Navigate to the ViewAllUsers page.</li> <li>2. Clicks on the Delete icon for a specific user (Figure 168).</li> </ol>		
<b>Expected Result</b>	<b>Actual Result</b>		
The user is deleted and the remaining users are displayed (Figure 169).	Same as the Expected Result		

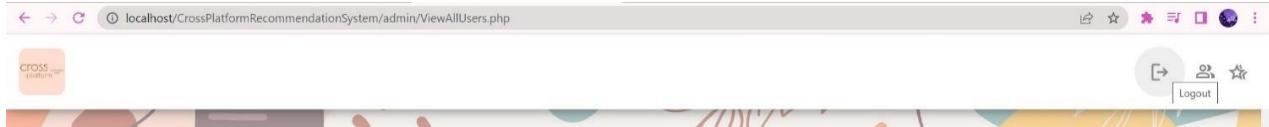
**Figure 168***Delete button in ViewAllUsers page***Figure 169***Updated users in ViewAllUsers page*

**Table 137**

<b>Test Case ID</b>	AN_01	<b>Status</b>	Pass
<b>Module Name</b>	Admin Navigation Bar		
<b>Test Case Title</b>	Admin Navigation Bar – Logout		
<b>Test Case Description</b>	When the admin clicks on the Logout button on the navigation bar, he is logged out.		
<b>Pre-conditions</b>	<b>Test Data</b>		
<ul style="list-style-type: none"> <li>• Access to a browser</li> <li>• The admin is already logged in.</li> </ul>	email = cprstestadmin1@gmail.com password = Hello@Admin		
<b>Steps</b>	3. Navigate to the ViewAllUsers or ViewAllRatings page.  4. Click on the Logout button in the navigation bar (Figure 170).		
<b>Expected Result</b>	<b>Actual Result</b>		
The admin is logged out and is navigated to the AdminLogin page.	Same as the Expected Result		

**Figure 170**

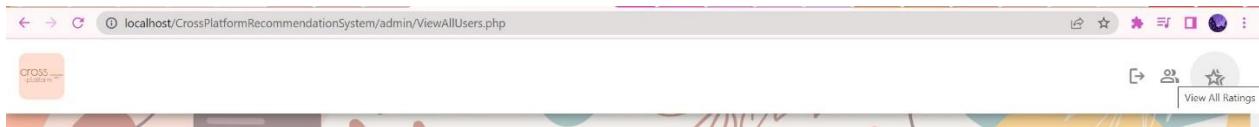
*Logout button*

**Table 138**

<b>Test Case ID</b>	AN_02	<b>Status</b>	Pass
<b>Module Name</b>	Admin Navigation Bar		
<b>Test Case Title</b>	Admin Navigation Bar – Go to View All Ratings		
<b>Test Case Description</b>	When the admin clicks on the ‘View All Ratings’ button on the navigation bar, he is navigated to the ViewAllRatings page.		
<b>Pre-conditions</b>	<b>Test Data</b>		
<ul style="list-style-type: none"> <li>• Access to a browser</li> <li>• The admin is already logged in.</li> </ul>	email = cprstestadmin1@gmail.com password = Hello@Admin		
<b>Steps</b>	1. Click on the ‘View All Ratings’ button in the navigation bar (Figure 171).		
<b>Expected Result</b>	<b>Actual Result</b>		
The admin is navigated to the ViewAllRatings page.	Same as the Expected Result		

**Figure 171**

*View All Ratings button*



**Table 139**

<b>Test Case ID</b>	AN_03	<b>Status</b>	Pass
<b>Module Name</b>	Admin Navigation Bar		
<b>Test Case Title</b>	Admin Navigation Bar – Go to View All Users		
<b>Test Case Description</b>	When the admin clicks on the ‘View All Users’ button on the navigation bar, he is navigated to the ViewAllUsers page.		
<b>Pre-conditions</b>	<b>Test Data</b> <ul style="list-style-type: none"> <li>• Access to a browser</li> <li>• The admin is already logged in.</li> </ul>		
<b>Steps</b>	1. Click on the ‘View All Users’ button in the navigation bar (Figure 172).		
<b>Expected Result</b>	<b>Actual Result</b>		
The admin is navigated to the ViewAllUsers page.	Same as the Expected Result		

**Figure 172**

*View All Users button*



### Traceability between test cases and user story

User	User Story	Test Case	Test Case Title
Story No.		No.	
1	Opening page of the website/application	OP_01	Opening Page
		OP_02	Opening Page – Navigate to Login
2	Registering a new account for the website	SU_01	Sign Up
		SU_02	Sign Up – Invalid User Credentials
		SU_03	Sign Up – Already Registered User
		SU_04	Sign Up – Invalid Verification Code
		SU_05	Sign Up – Navigate to Login
		SU_06	Sign Up – Go To Input User Credentials
		SU_07	Sign Up – Toggle Show Password
		SU_08	Sign Up – New User Already Logged In
		SU_09	Sign Up – Existing User Already Logged In
3	Logging into the user's account	LG_01	User Login
		LG_02	User Login – Valid New User
		LG_03	User Login – Valid Existing User
		LG_04	User Login – Invalid User Credentials
		LG_05	User Login – Navigate to Sign Up
		LG_06	User Login – Navigate to Forgot Password
		LG_07	User Login – New User Already Logged In

User	User Story	Test Case	Test Case Title
Story No.		No.	
		LG_08	User Login – Existing User Already Logged In
4	Resetting the password when the user forgot his/her password	FP_01	Forgot Password
		FP_02	Forgot Password – Invalid Email
		FP_03	Forgot Password – Invalid Verification Code
		FP_04	Forgot Password – Invalid New Password
		FP_05	Forgot Password – Navigate to Login
		FP_06	Forgot Password – Go To Input Email
		FP_07	Forgot Password – Toggle Show Password
		FP_08	Forgot Password – New User Already Logged In
		FP_09	Forgot Password – Existing User Already Logged In
5	Setting preferred languages and OTT platforms for the first time	NP_01	New User Preferences
		NP_02	New User Preferences – No Preferences Selected
		NP_05	New User Preferences – User Not Logged In
		NP_06	New User Preferences – Existing User Logged In

User	User Story	Test Case	Test Case Title
Story No.		No.	
6	Provide ratings for a few movies/series for the first time	NP_01 NP_03 NP_04	New User Preferences New User Preferences – No Shows Rated New User Preferences – Invalid Rating
7	Recommendations based on the ratings provided by the user	HP_02 HP_05 HP_06	Homepage – Personalized Recommendations Homepage – User Not Logged In Homepage – New User Logged In
8	To view platform-wise recommendations	HP_03	Homepage – Personalized Recommendations Platform-wise
9	To view basic information for any movies/series	SI_01 SI_03 SI_04 HP_07 SE_01	Show Information Show Information – Invalid Show ID Show Information – No Show ID Passed Homepage – Go to Show Search – Go to Show
10	The search option to look for specific show	NB_02 NB_04 SE_02 SE_03	Navigation Bar – Search Navigation Bar – Search Invalid Keyword Search – No Shows Found Search – No Search Keyword Passed
11	Set a status for any movie/series	SI_05 SI_08	Show Information – Set Status Show Information – Close Icon

User	User Story	Test Case	Test Case Title
Story No.		No.	
12	Provide rating for any movie/series	SI_06 SI_07 SI_08	Show Information – Set Rating Show Information – Invalid Rating Show Information – Close Icon
13	To view similar content for any movies/series	SI_02	Show Information – Similar Content
14	To view the list of movies/series that the user is currently watching	HP_01	Homepage – Currently Watching
15	To view a Wishlist that consists of movies/series that a user wants to watch	HP_04	Homepage – Wishlist
16	Editing personal details	NB_03 PR_01 PR_02 PR_03 PR_04 PR_07	Navigation Bar – Go to Profile Profile Profile – Edit Details Profile – Change Profile Photo Profile – Invalid Details Entered Profile – Invalid File Selected

User	User Story	Test Case	Test Case Title
Story No.		No.	
17	Change ratings for any movies/series	SI_06 SI_07	Show Information – Set Rating Show Information – Invalid Rating
18	Logging out of the account	NB_01	Navigation Bar – Logout
		AN_01	Admin Navigation Bar – Logout
20	Deleting user account	PR_05 PR_06	Profile – Delete User Account Profile – Delete User, Invalid Password
21	Logging into the admin's account	AL_01 AL_02 AL_03 AL_04	Admin Login – Valid Credentials Admin Login – Invalid Credentials Admin Login – Admin Already Logged In Admin Login – Toggle Show Password
22	Viewing all ratings	VR_01 AN_02	View All Ratings Admin Navigation Bar – Go To View All Ratings
23	Remove a rating	VR_02	View All Ratings – Remove a Rating
24	Viewing all users	VU_01 AN_03	View All Users Admin Navigation Bar – Go To View All Users
25	Remove a user	VU_02	View All Users – Remove a User

## Reflection

### Executed/Proposed schedule

Task	Planned/Proposed		Actual/Executed	
	From Date	To Date	From Date	To Date
Proposal	Aug 29, 2022	Sep 05, 2022	Same as Planned	
<b>Part 1 - Introduction</b>	Sep 07, 2022	Sep 25, 2022	Same as Planned	
User stories	Sep 07, 2022	Sep 17, 2022	Same as Planned	
Sprint	Sep 18, 2022	Sep 24, 2022	Same as Planned	
Assignment: Part 1 - Introduction	Sep 25, 2022		Same as Planned	
<b>Part 2 - Design</b>	Sep 26, 2022	Oct 30, 2022	Same as Planned	
Sprint 1	Sep 26, 2022	Oct 09, 2022	Same as Planned	
Design planning, UI planning	Sep 26, 2022	Sep 27, 2022	Same as Planned	
Use case diagrams	Sep 27, 2022	Oct 09, 2022	Sep 27, 2022	Sep 29, 2022
Class diagrams			Sep 29, 2022	Oct 02, 2022
Activity diagrams			Oct 03, 2022	Oct 05, 2022
Sequence diagrams			Oct 06, 2022	Oct 09, 2022
Sprint 2	Oct 10, 2022	Oct 23, 2022	Same as Planned	
Assignment: Part 2 - Design	Oct 24, 2022	Oct 30, 2022	Same as Planned	
<b>Part 3 - Verification</b>	Oct 31, 2022	Nov 28, 2022	Same as Planned	
Sprint 3	Oct 24, 2022	Nov 06, 2022	Oct 31, 2022	Nov 06, 2022
Sprint 4	Nov 07, 2022	Nov 20, 2022	Nov 07, 2022	Nov 13, 2022
Final Code Merge	Nov 13, 2022		Same as Planned	

Task	Planned/Proposed		Actual/Executed	
	From Date	To Date	From Date	To Date
Verification planning	Nov 10, 2022	Nov 13, 2022	Same as Planned	
Building test cases	Nov 14, 2022	Nov 16, 2022	Same as Planned	
Testing Phase I	Nov 17, 2022	Nov 23, 2022	Nov 17, 2022	Nov 20, 2022
Fixing Bugs			Nov 21, 2022	Nov 22, 2022
Testing Phase II			Nov 23, 2022	Nov 26, 2022
Assignment: Part 3 - Verification	Nov 24, 2022	Nov 28, 2022	Nov 27, 2022	Nov 28, 2022
<b>Final Report, Presentation</b>	Nov 29, 2022	Dec 06, 2022	Same as Planned	
Presentation	Nov 29, 2022	Dec 02, 2022	Same as Planned	
Final Report	Dec 05, 2022	Dec 06, 2022	Same as Planned	

### ***Deviation from the planned***

We initially planned to finish the testing phase by November 23<sup>rd</sup>. Since we had a huge number of test cases, it took us till November 20<sup>th</sup> for everyone to finish testing their test cases. We made a list of bugs on our website, and it took us till November 22<sup>nd</sup> to finish fixing them. We started our testing once again on November 24<sup>th</sup>. Originally, we had planned to start the Part 3 assignment on this day, but it had to be postponed to November 27<sup>th</sup>. Once we finished our final testing, we started working on the assignment. During our second phase of testing, each time we finished a test case, we filled its test case table and also took the screenshots for the test case. Since this part of the assignment is already done, we could finish the rest of the assignment by November 28<sup>th</sup>.

Apart from this, everything pretty much went as planned. We also finished a few tasks early so that we could spend the remaining time for the upcoming tasks.

## References

- Bansal, S. (2021, November 29). *Disney+ movies and TV shows*. Kaggle. Retrieved from  
<https://www.kaggle.com/datasets/shivamb/disney-movies-and-tv-shows>
- Bansal, S. (2021, October 12). *Amazon prime movies and TV shows*. Kaggle. Retrieved from  
<https://www.kaggle.com/datasets/shivamb/amazon-prime-movies-and-tv-shows>
- Bansal, S. (2021, October 25). *Hulu movies and TV shows*. Kaggle. Retrieved from  
<https://www.kaggle.com/datasets/shivamb/hulu-movies-and-tv-shows>
- Bansal, S. (2021, September 27). *Netflix movies and TV shows*. Kaggle. Retrieved from  
<https://www.kaggle.com/datasets/shivamb/netflix-shows>
- Bourque, P., Dupuis, R. (2004). *Guide to the software engineering body of knowledge (SWEBOK)*. IEEE Computer Society Press, Los Alamitos
- Dadban, H. (n.d.). *API for imdb, tmdb, Wikipedia and more - IMDb-API*. IMDb. Retrieved from  
<https://imdb-api.com/>
- Navot, Y. (2022, May 25). *The power of collaborative filtering*. Dynamic Yield. Retrieved from  
<https://www.dynamicyield.com/lesson/collaborative-filtering/>
- Wallace, D., Fujll, R. (1989, May). *Software Verification and validation: an overview*. IEEE Software, 6(3)
- Wikimedia Foundation. (2022, June 12). *Collaborative filtering*. Wikipedia. Retrieved from  
[https://en.wikipedia.org/wiki/Collaborative\\_filtering](https://en.wikipedia.org/wiki/Collaborative_filtering)