****

**LA GRANDEE INTERNATIONAL COLLEGE**

**Simalchaur, Pokhara Nepal**

**Final Report Defence**

**On**

**“Pop Corn Box”**

**Submitted to:**

**Bachelor of Computer Application (BCA) Program**

**In partial fulfilment of the requirements for the degree of BCA under**

**Pokhara University**

**Submitted by:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name:** | **Course** | **Semester** | **P.U. Registration Number** |
| **Krishna Gurung** | **BCA** | **6th** | **2021-1-53-0353** |
| **Prajal Gurung** | **BCA** | **6th** | **2021-1-53-0359** |
|  |  |  |  |

**Date:16/02/2025**

# 

# Acknowledgement

We would like to express our gratitude to our BCA coordinator Mr**. Kundan Chaudhari**, Project supervisor **Mr.** **Sunil Sapkota** and LA Grandee International College for their support and contributions to the development of Pop Corn Box.

This project is done for the in partial fulfilment of the requirements for BCA (Bachelor of Computer Application) program under Pokhara University. Our project was made possible by the effort and dedication of our team members. We thank our dedicated team for their hard work and contributions to the game. We are grateful for the guidance and mentorship provided by our respected sir **Mr. Sunil Sapkota.**

Sincerely,

Krishna Gurung

Prajal Gurung

**Declaration for**

**“Pop Corn Box”**

# Student’s Declaration

We, **Prajal Gurung** and **Krishna Gurung** being students of the sixth semester at **LA GRANDEE International College**, Faculty of Science and Technology ‘kha’, Pokhara University, do hereby declare that the project proposal submitted to the aforementioned institution is an original work completed by us in partial fulfilment of the requirements for the Bachelor of Computer Application (BCA) program, under the supervision of Sir **Mr. Sunil Sapkota**. We further state that no resources other than those specifically listed have been utilized in the completion of this project.

Name: Prajal Gurung Name: Krishna Gurung

Class Roll No.: 15 Class Roll No.: 9

PU-Registration No.: 2021-1-53-0359 PU-Registration No.: 2021-1-53-0353

Semester: 6th Semester Semester: 6th Semester

**Date: 16/02/2025 Date: 16/02/2025**

Signature: ............... Signature: ...............

# Supervisor’s Declaration

I hereby recommend that this project entitled “Pop Corn Box” is done under my supervision by **Prajal Gurung & Krishna Bahadur Gurung** during their sixth Semester in partial fulfilment of the requirements for the degree of BCA under Pokhara University is completed to my satisfaction and be processed for final evaluation.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Mr. Sunil Sapkota**

**Date:16/02/2025**

# Abstract

The “PopCornBox” project is focused on developing a web-based application that aims to simplify movie discovery, subscription management, and payment handling, creating a seamless and user-friendly environment for movie lovers. By addressing usual challenges faced by current movie streaming platforms, such as complex user interfaces, intrusive ads, and inefficient watchlist management, “PopCornBox” enhances the overall movie-watching experience.

**TABLE OF CONTENTS**

[Acknowledgement ii](#_Toc1249147584)

[Student’s Declaration iii](#_Toc2120958622)

[Supervisor’s Declaration iv](#_Toc551833447)

[Abstract v](#_Toc824046059)

[1. INTRODUCTION 1](#_Toc2062674307)

[2. PROBLEM STATEMENT 2](#_Toc997183692)

[3. OBJECTIVES 3](#_Toc305573993)

[4. BACKGROUND STUDY 4](#_Toc107757795)

[5. Requirement Document 5](#_Toc1781033958)

[5.1 Functional requirements 5](#_Toc2000869632)

[5.1.1 User Features 5](#_Toc1263777748)

[5.1.2 Admin Features 5](#_Toc951517853)

[5.2 Non-Functional Requirements 6](#_Toc198341010)

[6. SYSTEM DESIGN 7](#_Toc65188162)

[6.1 ER-DIAGRAM 7](#_Toc88744855)

[6.2 DATAFLOW DIAGRAM 8](#_Toc1571626152)

[7. METHODOLOGY 11](#_Toc1712005545)

[8. PROJECT GANTT CHART 14](#_Toc1529566804)

[9. Work Assigned 15](#_Toc83524807)

[10. TESTING 16](#_Toc2025476813)

[11. PROJECT RESULTS 17](#_Toc891577666)

[12. FUTURE ENHANCEMENTS 18](#_Toc1244168708)

[13. CONCLUSION 19](#_Toc707430044)

[14. ANNEXURES 20](#_Toc1685454798)

[15. Reference: 24](#_Toc988577468)

**LIST OF FIGURES**

[Figure 6.1 ER- Diagram 7](#_Toc188170694)

[Figure 6.2 Level 0, DFD 8](#_Toc188170695)

[Figure 6.3 Level 1 DFD 9](#_Toc188170696)

[Figure 6.4 Level 2, DFD 10](#_Toc188170697)

[Figure 7.1 Agile Model 11](#_Toc188170698)

[Figure 8.1: Timeline chart 14](#_Toc188170699)

**Abbreviations**

|  |  |
| --- | --- |
| **DFD** | **Data Flow Diagram** |
| **PCB** | **Pop Corn Box** |
| **DB** | **Database** |
| **ERD** | **Entity Relational Diagram** |
| **UI** | **User Interface** |
| **CSS** | **Cascading Style Sheet** |
| **JS** | **Java Script** |

# 

# INTRODUCTION

A **Movie Streaming App** is a specialized software application designed to allow users to watch movies over the internet. This platform aims to provide an on-demand entertainment service, where users can access a vast library of movies without needing to download its files. As viewers increasingly seek personalized and user-friendly experiences, the demand for innovating and streaming application has grown.

The primary goal of this project is to efficiently manage and deliver a vast array of video content to users while ensuring a high-quality, uninterrupted streaming experience. Additionally, it simplifies administrative tasks like user subscription management and payment handling, enabling smooth business operations for the platform provider.

The motivation behind creating a “PopCornBox” is to address the evolving needs of movie enthusiast by providing a seamless and enriching platform for discovering and enjoying required movie contents. This proposal outlines the vision of objectives and development plan for “PopCornBox”, highlighting streaming experience for normal movie watcher.

Throughout the project, various software design, including ER diagrams, DFD, project Gantt chart will be employed to support its development. The coding will do in the web technologies like for frontend: HTML, CSS, JS and for backend: we have-not decided till now. The project tasks will be evenly and equally distributed among team members according to their skills and knowledge.

# PROBLEM STATEMENT

* Complicated and Unmanaged User Interfaces (UI)
* Commercial Interruptions
* Inefficient Watchlist Management

# OBJECTIVES

* Billing and Payment Management
* Membership Management
* Simplify User Interface (UI)

# BACKGROUND STUDY

We initiated our investigation by recognizing the necessity for a Movie Streaming Web Application. Initially, our research was focused on identifying the underlying reasons that necessitate the implementation of “PopCornBox”. We gathered various project requirements through visiting website like Netflix, Amazon Prime Video, Hulu, Disney+, etc.

During our analysis, we investigate the common problems faced by existing movie application. The primary goal of the project was to create a system that could be easily managed and provide security while covering all the key aspects of “PopCornBox”, such as membership management, subscription, and simple user interface.

However, it became apparent that the project had limitations, primarily due to complexity of user interface, commercial interruption and inefficient watchlist management. While visiting “Amazon Prime” (https://www.primevideo.com), one of the popular movie streaming apps, we noticed/detected that its UI has been widely criticized for its cluttered and confusing layout. We also noticed numerous “Hulu” (https://www.hulu.com) users frequently about the high frequency and repetitive ads which ultimately disrupts the flow of movie-watching experience, making it less enjoyable. Similarly, Efficient watchlist management is crucial for users who want to keep track of movies which enables users to organize and prioritizes watchlist which we found lacking various trending movie streaming sites like Amazon Prime, Netflix (https://www.netflix.com), Hulu, etc.

After having gone through the analysis, we have tried to coverup these problems or drawbacks of existing movie streaming sites. From this analysis, we will be developing the “PopCornBox”, movie app designed to tackle these issues and provide more streamlined and enjoyable movie-watching experience.

# 5. Requirement Document

## 5.1 Functional requirements

### **5.1.1 User Features**

1. **User Registration and Login**
   1. Users must be able to create an account, log in, and log out securely.
   2. Support for password recovery via email.
2. **Membership and Subscription Management**
   1. Users can choose between subscription plans (monthly, yearly, etc.).
   2. Option for upgrading or downgrading membership plans.
3. **Movie Browsing and Search**
   1. A search bar to allow users to find movies
4. **Watchlist Management**
   1. Allow users to create, view, update, and delete their personalized watchlists.
5. **Video Streaming**
   1. Stream movies directly in-app with adaptive streaming based on internet speed.
6. **Billing and Payment Management**
   1. Secure payment options

### **5.1.2 Admin Features**

1. **User Management**
   1. View, edit user accounts.
2. **Subscription Monitoring**
   1. Track user subscriptions and payments.
3. **Activity Logs**
   1. Maintain logs for admin actions for accountability.

## 5.2 Non-Functional Requirements

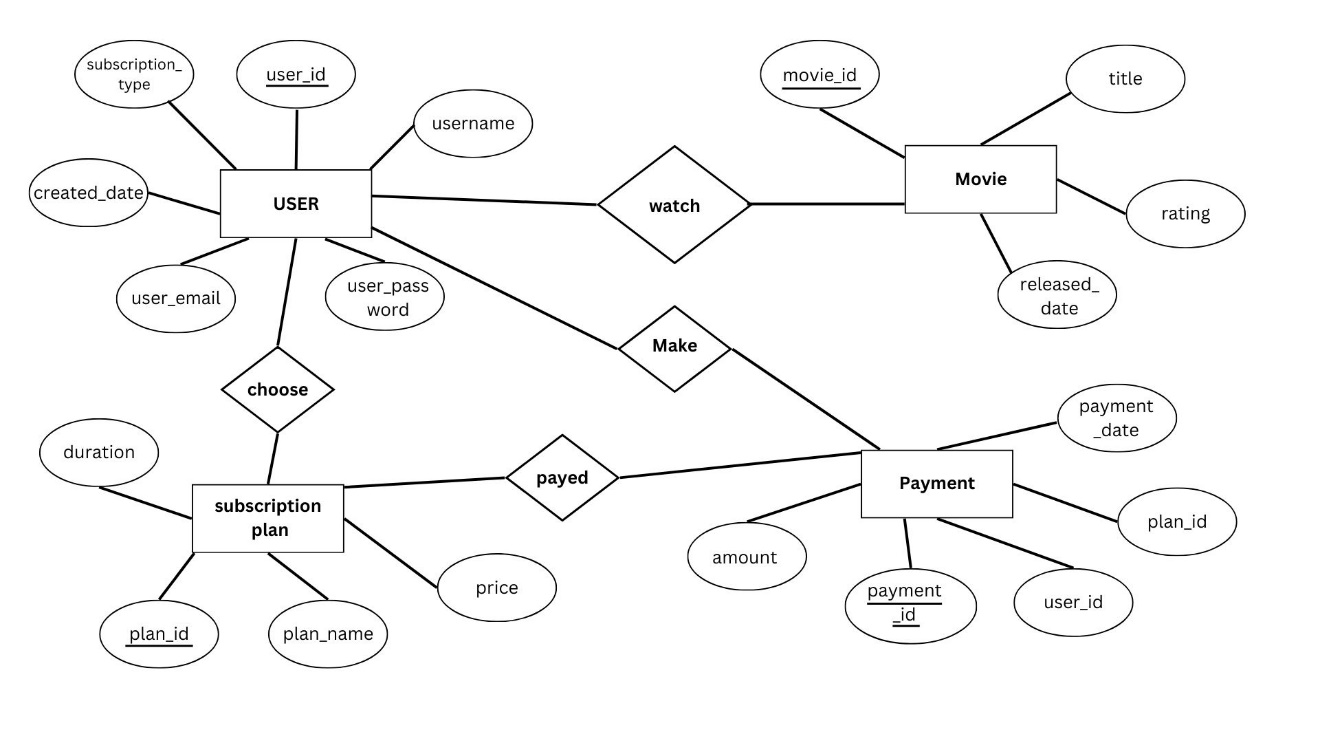
* Usability: The application must have a user-friendly interface with easy navigation.
* Performance**:** The application must meet certain performance requirements such as fast loading time and low memory usage.
* Security and privacy: The application must have security and privacy features which are authentication and authorization.

# 6. SYSTEM DESIGN

Dataflow, Algorithm and Flowchart are used for understanding the system's design and its functionalities, and both are important for creating proper documentation.

## 6.1 ER-DIAGRAM

An Entity-Relationship (ER) diagram is a visual representation of a database's structure. It uses entities (objects or concepts) and their relationships to illustrate how data is organized and connected within a database system. ER diagrams are widely used in database design and modelling to help understand and plan data relationships.

****

**Figure 0.1 ER- Diagram**

## 6.2 DATAFLOW DIAGRAM

It is a diagrammatic representation that portrays the flow of data in a system or a process. Helps communicate the general data flow structure of a proposed system to the system designer, programmer, and end-users.

**A diagram of a movie application

Description automatically generated**

**Figure 0.2 Level 0, DFD**

**A diagram of a application

Description automatically generated**

**Figure 0.3 Level 1 DFD**

**A diagram of a login access

Description automatically generated**

**Figure 0.4 Level 2, DFD**

# METHODOLOGY

**A diagram of a process

Description automatically generated**

**Figure 7.1 Agile Model**

* The discussions of 6 steps of Agile Methodology are given below.

1. Sprint 0: Project Initiation and Planning

This is the foundational phase where the groundwork for the project is laid. Although not always considered a "sprint" in traditional Agile, it is essential for setting up the project.

Activities:

* Project Name : Define the name of the project.
* Objectives : Clearly outline the goals and objectives of the project.
* Problem Identification : Identify the problem the project aims to solve.
* Background Study : Conduct research to understand the context and existing solutions.
* Project Gantt Chart : Create a high-level timeline (Gantt chart) to visualize the project phases and milestones.
* Requirement Gathering : Start gathering initial requirements from stakeholders.

Deliverables:

* Project charter or initiation document.
* High-level Gantt chart.
* Initial list of requirements.

2. Sprint 1: Frontend Development - Login and Registration

In this sprint, the focus is on building the frontend components for user authentication.

Activities:

* Coding : Develop the frontend for login and registration pages.
* UI/UX Design : Ensure the design aligns with user experience best practices.
* Review Requirements : Revisit the requirement documents to ensure alignment with the project goals.
* System Design : Begin drafting the system architecture, focusing on the frontend components.

Deliverables:

* Functional login and registration pages.
* Updated system design documentation.
* Revised project Gantt chart reflecting progress.

3. Sprint 2: Frontend Development - Dashboard

The second sprint focuses on developing the dashboard, which serves as the central hub for users.

Activities:

* Coding : Build the frontend for the dashboard, including key features like navigation, widgets, and data visualization.
* Requirement Documents : Study the updated requirement documents to ensure all features are covered.
* System Design : Refine the system design to include the dashboard's integration with backend services.
* Project Gantt Chart : Update the timeline to reflect progress and adjust for any changes in scope.

Deliverables:

* Functional dashboard interface.
* Updated system design and requirement documents.
* Revised Gantt chart.

4. Sprint 3: Backend Development - Initial Setup

This sprint marks the transition to backend development, laying the foundation for server-side logic and database interactions.

Activities:

* Coding : Begin coding the backend, focusing on core functionalities like user authentication, API endpoints, and database schema design.
* Methodology : Document the development methodology being used (e.g., Agile, Scrum).
* References : Compile references for technologies, frameworks, and libraries used in the project.
* Project Gantt Chart : Update the timeline to reflect backend development progress.

Deliverables:

* Initial backend setup with basic functionalities.
* Documentation of methodology and references.
* Updated Gantt chart.

5. Sprint 4: Backend Development - Full Implementation

This sprint focuses on completing the backend development and ensuring seamless integration with the frontend.

Activities:

* Coding : Finalize the backend implementation, including advanced features like data processing, security measures, and scalability considerations.
* Testing : Conduct unit testing and integration testing to identify and fix bugs.
* Project Chart : Update the project chart to reflect the status of backend development.
* References : Add any new references or resources used during this phase.

Deliverables:

* Fully functional backend.
* Test cases and results.
* Updated project chart and references.

6. Sprint 5: Testing, Deployment, and Review

The final sprint involves comprehensive testing, deployment, and a review of the entire project.

Activities:

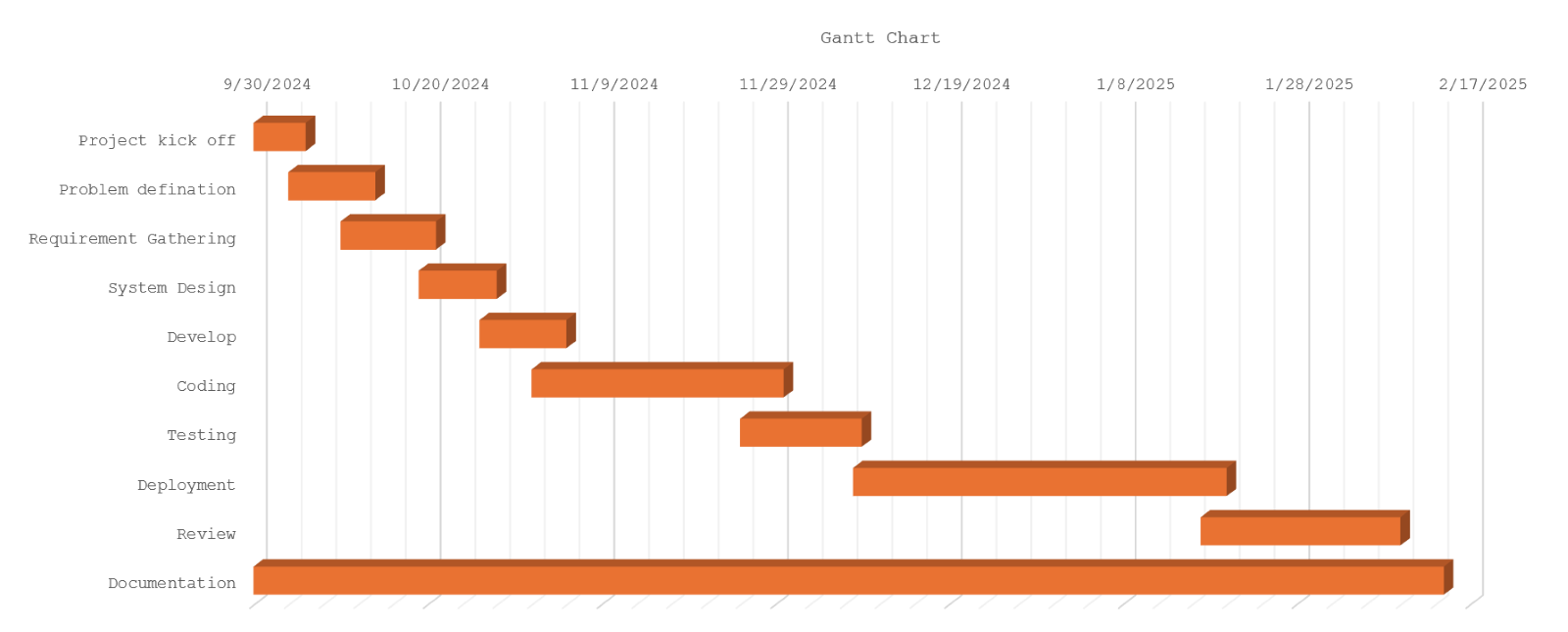
* Testing : Perform end-to-end testing, including user acceptance testing (UAT), to ensure the application meets all requirements.
* Deployment : Deploy the application to the production environment.
* Review : Conduct a retrospective meeting to evaluate what went well, what could be improved, and lessons learned.
* Documentation : Finalize all documentation, including user manuals, technical guides, and references.

Deliverables:

* Fully tested and deployed application.
* Comprehensive documentation.
* Retrospective report.

# PROJECT GANTT CHART

A Gantt chart is a popular project management tool used to visualize the schedule of a project. It displays tasks or activities against time, allowing project managers to track progress, manage dependencies, and allocate resources efficiently.

****

**Figure 8.1: Timeline chart**

|  |  |  |
| --- | --- | --- |
| **INDEX** | | |
| **S. No.** | **Color** | **Work Status** |
| **1** | **Orange** | **Completed** |
| **2** | **Blue** | **Not completed** |

# Work Assigned

The different task identified for the compilation of the project were divided among the team members, with accordance to their talent and capabilities, and performed accordingly. Later they were integrated together to form a single unit. The division of task between four of us is tabulated below.

|  |  |  |  |
| --- | --- | --- | --- |
| **S.N.** | **Name of the member** | **Work assigned** | **Remarks** |
| **1.** | **Krishna Gurung** | **Coding**  **Documentation**  **Problem Identification**  **Coding of Major Module** |  |
| **2.** | **Prajal Gurung** | **Documentation**  **UI / UX Design**  **Support in Coding**  **System Design**  **Gantt Chart** |  |

# TESTING

Software testing is the process of evaluating a software product to ensure it function as intended. It involves running test cases to identify bugs and verify that the software meets expectations.

Purpose To Find issues early in the development process, To reduce the risk of software failure, To improve the performance, and to prevent bugs.

We had performed following test cases:

**Test Cases**

**Test Case 1**

**Title**: Login Testing

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case Id** | **Test Case** | **Test Steps** | **Test Data** | **Expected Results** |
| **1** | Invalid Login | 1.Navigate to the login page.  2.Enter valid credentials.  3. Click “login”. | Email = abc  Password = v78 | Incorrect email and password |
| **2** | Valid login | 1.Navigate to the login page.  2.Enter valid credentials.  3. Click “login”. | Email = [gprajal88@gmail.com](mailto:gprajal88@gmail.com)  Password = 12345 | User is redirected to the dashboard or home page. |
| **3** | Empty Field | 1.Navigate to the login page.  2.Enter valid credentials.  3. Click “login”. |  | Please fill out this field |

Test Case 2:

Title: Registration Testing

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case Id** | **Test Case** | **Test Steps** | **Test Data** | **Expected Results** |
| **1** | Duplicate Registration | 1.Navigate to the register page.  2.Enter valid credentials.  3. Click “register”. | First name=Prajal  Last name=Gurung  [Email=gprajal88@gmail.com](mailto:Email=gprajal88@gmail.com)  Password = 12345 | Email already exist |
| **2** | Valid register | 1.Navigate to the register page.  2.Enter valid credentials.  3. Click “register”. | First name = Prajal  Last name = Gurung  Email = [gprajal88@gmail.com](mailto:gprajal88@gmail.com)  Password = 12345 | User is redirected to the login page |
| **3** | Empty Field | 1.Navigate to the register page.  2.Enter valid credentials.  3. Click “register”. |  | Please fill out this field |

Test Case 3:

Title: Home Page Testing

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case Id** | **Test Case** | **Test Steps** | **Test Data** | **Expected Results** |
| **1** | Homepage load | 1.Login with valid credentials.  2. redirect to the homepage |  | Payed user can only load to homepage |
| **2** | Logout functionality | 1.Login with valid credentials.  2. Click “logout”. |  | User is redirected to the login page and session is deleted |

Test Case 4:

Title: Subscription Testing

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case Id** | **Test Case** | **Test Steps** | **Test Data** | **Expected Results** |
| **1** | Verify subscription plans are displayed dynamically | 1.Navigate to the payment page.  2.Check the order of plans. |  | Plans are displayed |
| **2** | Verify user can select a subscription tier | 1.Click on a subscription tier (eg : “yearly”).  2.Proceed to payment | Selected:900 | User is redirected to payment gateway |

Test Case 5:

Title: Forget Password Testing

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case Id** | **Test Case** | **Test Steps** | **Test Data** | **Expected Results** |
| **1** | Forget Password | 1.Navigate to the forget page.  2.Enter valid credentials.  3.Click “Send reset link”. | Email = gprajal88@gmail.com | Redirect to verify otp and reset password |
| **2** | Verify Forget Password | 1.Enter valid credentials  2.Click “set new password.” | Otp = 193402  Password = 54321 | Password reset successfully and redirect to login page |
| 3 | Verify forget password with invalid detail | 1.Enter invalid credentials.  2.Click “Set new password” | Otp = 123456  Password = 54321 | Invalid otp. |

# PROJECT RESULTS

The **Popcorn Box** movie management system project successfully delivered a robust platform for users to browse, book, and review movies while enabling administrators to manage movie listings and showtimes seamlessly. The system featured an intuitive user interface, allowing users to search movies by title, genre, or language, view detailed information such as trailers and cast, and book tickets with secure payment integration.

Administrators could efficiently handle tasks like adding or editing movies, configuring showtimes, and monitoring bookings through a dedicated dashboard. Built with modern technologies like Node.js, and MongoDB, Popcorn Box ensured a smooth and responsive experience for all users. The platform received positive feedback for its simplicity and efficiency, making movie discovery and ticket booking more convenient. Future enhancements include adding personalized movie recommendations, expanding payment options, and developing a mobile application to enhance accessibility further.

* Functional Movie Streaming App: Responsive design that enhances user experience across various devices, including desktops, tablets, and smartphones.
* Security: It secures the user profile’s by using authentication and authorization like username and password.
* Subscription & Payment Management: It tracks membership details such as membership plan, start date, expiry date, and payment status.

# FUTURE ENHANCEMENTS

Below, we have listed enhancements which can range from improving user experience to adding new features that enhance functionality of our PopCornBox:

* Download Movie to watch offline.
* Add multi-language audio tracks and subtitles for accessibility.
* Allow users to purchase premium content or rent movies not included in their subscription.
* Notify users about subscription expiry, new movies, or updates to their watchlists.
* Filters for categorization (e.g., trending, genre, language, ratings).

# CONCLUSION

In conclusion, PopCornBox is designed to provide a smooth and enjoyable movie streaming experience for users, while making it easier for administrators to manage their operations. By addressing common issues faced by other streaming platforms—such as complicated interfaces, intrusive ads, and poor watchlist management—PopCornBox offers a cleaner, more user-friendly environment.

We have developed the system using a combination of modern web technologies and tools, ensuring that it is secure, reliable, and easy to navigate. The focus is on making the platform both efficient for users and easy to manage for admins, streamlining tasks like subscription management and content updates.

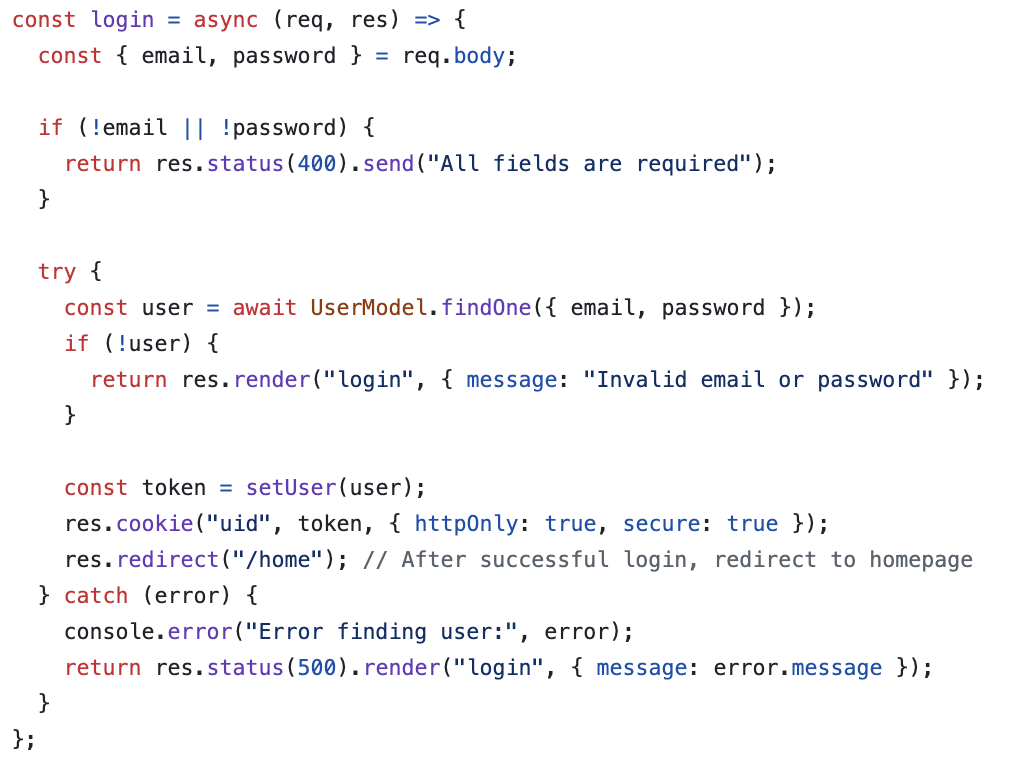
Ultimately, PopCornBox will allow movie enthusiasts to enjoy uninterrupted viewing, while gym owners and staff can rely on the system to handle administrative tasks with ease. With its simple interface and thoughtful features, PopCornBox is set to become an efficient and trusted solution for movie streaming needs.

# ANNEXURES

**Code Snippet for User Authentication:**

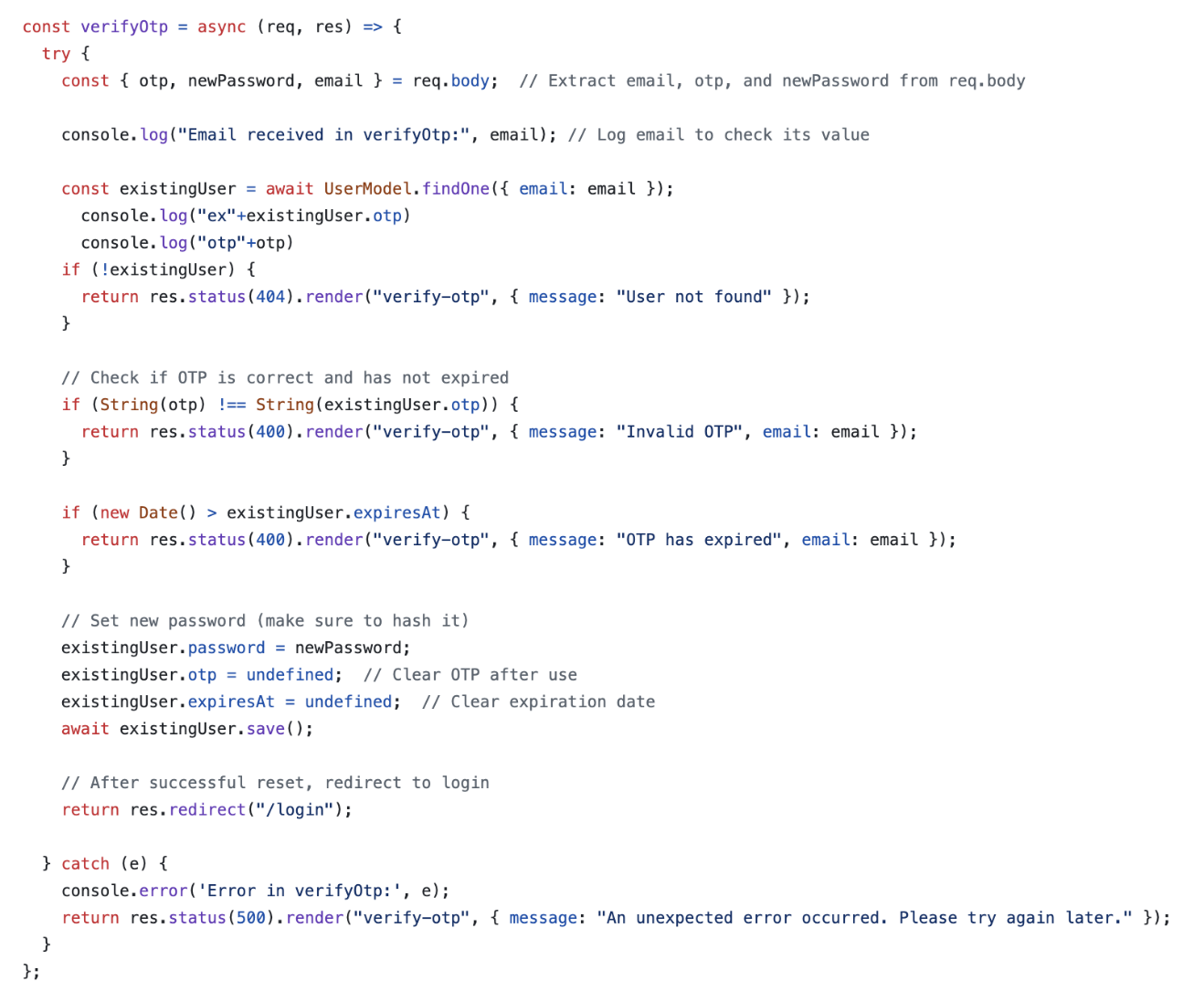
1. **Register**



1. **Login**
2. **Forget-Password**



1. **Verify-otp**



# Reference:

Netflix. (1997, August 29). Retrieved from Netlfix: https://www.netflix.com

Hulu. (2007). Retrieved from Hulu: https://www.hulu.com

Asana. (2008). Retrieved from Asana: https://asana.com/resources/agile-methodology

Prime, Amazon. (2005, February 2). Retrieved from Prime Video: https://www.primevideo.com