

**LA GRANDEE INTERNATIONAL COLLEGE**

**Simalchaur, Pokhara Nepal**

Project Proposal Report On

**“PopCornBox”**

**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** |
| Kailash Gurung | BCA | 6th | 2021-1-53-0352 |
| Krishna Bahadur Gurung | BCA | 6th | 2021-1-53-0353 |
| Prajal Gurung | BCA | 6th | 2021-1-53-0359 |

**Date: 12/12/2024**

**Table of Contents**

[Table of Figures iii](#_Toc184937814)

[Abbreviations iv](#_Toc184937815)

[1. Introduction 1](#_Toc184937816)

[2. Background Study 2](#_Toc184937817)

[3. Problem Statement 3](#_Toc184937818)

[4. Objectives 4](#_Toc184937819)

[5. Methodology 5](#_Toc184937820)

[6. Completed 7](#_Toc184937821)

[7. Needs to be done 8](#_Toc184937822)

[8. Issues that is hindering the project 9](#_Toc184937823)

[9. Project Gantt chart 10](#_Toc184937824)

[10. Deliverables 11](#_Toc184937825)

[11. Reference 12](#_Toc184937826)

# Table of Figures

[Figure 5.1: Agile Model 5](#_Toc184937873)

[Figure 6.1: Project Gantt chart 10](#_Toc184937874)

# Abbreviations

|  |  |
| --- | --- |
| CSS | Cascading Style Sheet |
| ERD | Entity Relational Diagram |
| HTML | Hypertext Markup Language |
| JS | JavaScript |

# 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.

# 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.

# Problem Statement

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

# Objectives

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

# Methodology

A diagram of a process

Description automatically generated

Figure 5.1: Agile Model

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

**1**. **Plan**

* **Continuous Planning**: Planning is not a one-time phase but an ongoing activity. Agile teams create a product backlog of features and prioritize them at the start of each iteration (e.g., sprint).
* **Short-Term Focus**: Plans are kept flexible and only detailed for the immediate iteration to accommodate changes.

2. Design

* **Just-in-Time Design**: Agile uses lightweight, iterative design. Instead of designing everything upfront, design evolves with the project.
* **Collaborative**: Teams and stakeholders work together to refine design aspects in real-time, often using user stories or prototypes.

3. **Develop**

* **Incremental Development**: Development focuses on building small, functional parts of the product during each sprint.
* **Cross-Functional Collaboration**: Developers work closely with designers, testers, and stakeholders to ensure alignment.

4. **Test**

* **Continuous Testing**: Testing happens alongside development. Test-driven development (TDD) and behavior-driven development (BDD) are common practices.
* **Automation**: Automated testing ensures rapid validation of changes, minimizing the risk of regressions.

5. **Deploy**

* **Frequent Deployment**: Agile encourages deploying working software frequently, often at the end of each sprint or even more often (e.g., continuous delivery).
* **Incremental Deployment**: Features are deployed incrementally, ensuring users can benefit from new updates without waiting for the entire project to be completed.

6. Review

* **Iterative Feedback Loops**: Agile incorporates regular reviews, including sprint reviews and retrospectives.
* **Stakeholder Involvement**: Customers, users, and other stakeholders are actively involved in reviewing the progress and providing feedback.

# Completed

We have completed the following features of our system till now:

* Basic Layout Design:
  + Homepage layout displays trending and top movies using TMDB API.
  + Includes a navigation menu and footer design
* Search Functionality:
  + Users can search for movies using keywords and the results are displayed dynamically.
* Technology Used:
  + Frontend: HTML, CSS ,JavaScript
  + Backend: Node.js , Express.js, MongoDB

# Needs to be done

As we are still in the process of developing the project, the following features remain to be implemented:

* Billing and Payment Management
* Membership Management
* User Authentication

# Issues that is hindering the project

Several issues are currently hindering our project’s progress:

* Learning Curve:
  + We chose Node.js as the backend technology and have been learning it through online books and YouTube tutorials. However, self-learning has made it challenging to keep up with project’s demands.
* API Access:
  + Finding free APIs for movie details has been difficult. Additionally, some APIs require an API key, which we have been unable to obtain due to restrictions. But, we somehow got the API and its key.
* Technology Used:
  + Frontend: HTML, CSS ,JavaScript
  + Backend: Node.js , Express.js, MongoDB
* API integration
  + Working with external APIs involves asynchronous operations, other functions and potentials failures. It was complex to manage multiple asynchronous APIs call and ensure smooth operations.

# Project Gantt chart

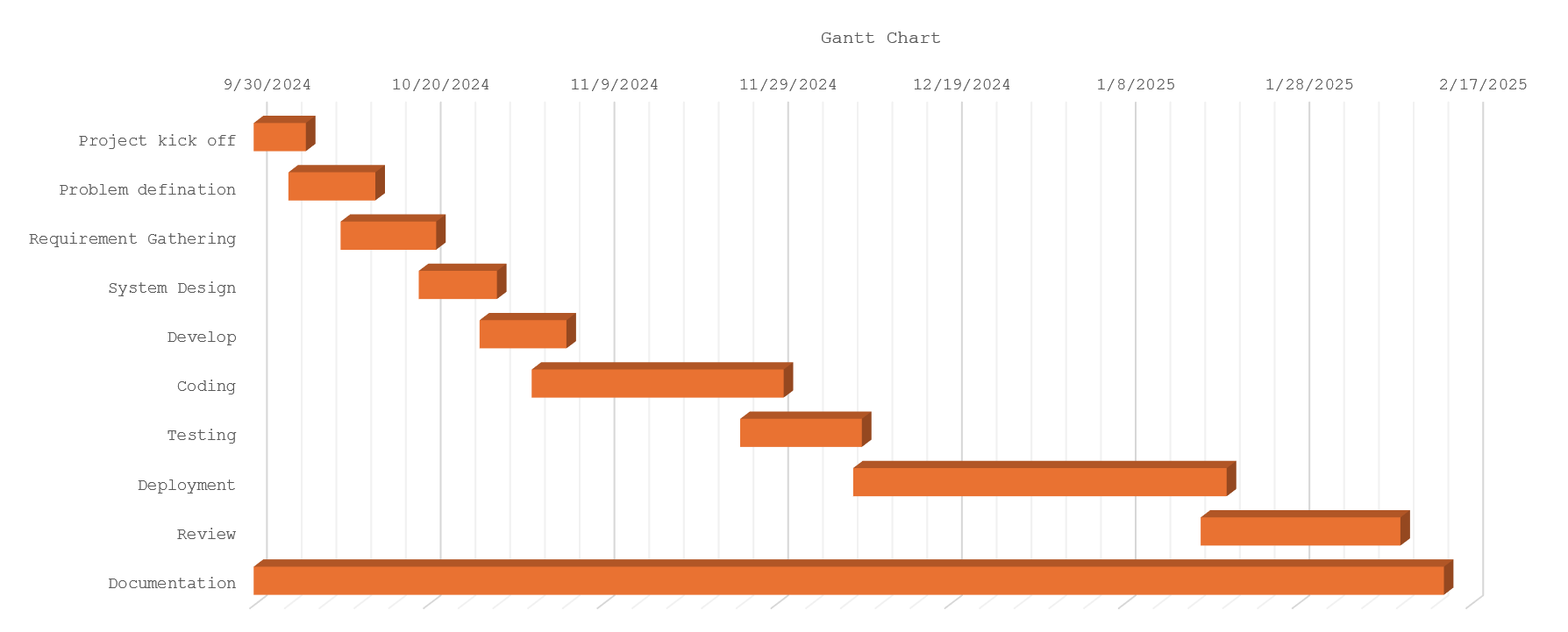


Figure 6.1: Project Gantt chart

# Deliverables

* **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.

# Reference

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

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

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

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