# 1. Project Proposal: Movie Streaming Platform

### 1. Overview

The **Movie Streaming Platform** is a web-based application designed to provide users with an experience for discovering, streaming, and managing movies. Inspired by platforms like Netflix and Amazon Prime, this project aims to offer a user-friendly interface with personalized recommendations, high-quality streaming, and a secure authentication system.

### 2. Objectives

The primary goals of this project are:

* **Develop a scalable and responsive web application** for streaming movies.
* **Implement user authentication and authorization** to manage subscriptions and user profiles.
* **Provide multiple streaming qualities** (e.g., SD, HD, and 4K) with adaptive bitrate streaming.
* **Optimize performance and scalability** to handle high traffic efficiently.

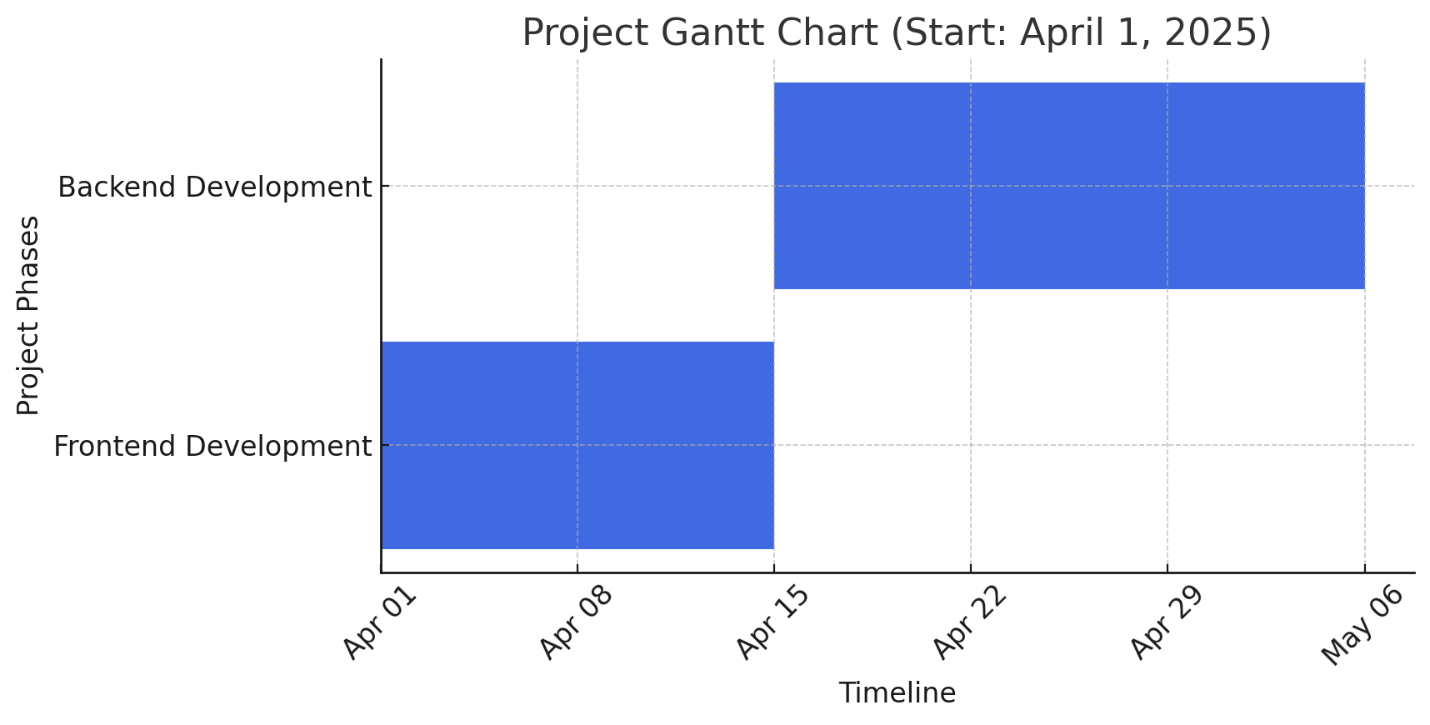
### 3. Scope

* User registration, login, and profile management.
* Movies browsing with filtering and search capabilities.
* Watchlist and favorites management.
* Streaming functionality with adaptive quality settings.
* Reviews and ratings system.
* Admin panel for content management.

# 

# 2. Project Plan

### 1. Grantt Chart



### 

### 2. Timeline & Milestones (5 Weeks)

| **Week** | **Phase** | **Tasks** | **Milestone** |
| --- | --- | --- | --- |
| **1-2** | Frontend Dev | Design UI, implement homepage, movie listing, and details pages | Basic UI ready & responsive |
| **3-5** | Backend Dev | Develop API (.NET), implement authentication & database | API functional & project live |

### 3. Deliverables

* **Week 2:** Fully responsive frontend with movie listing & details
* **Week 5:** Working backend with authentication & API endpoints

# 3. Task Assignment & Roles

### 1. Frontend Development

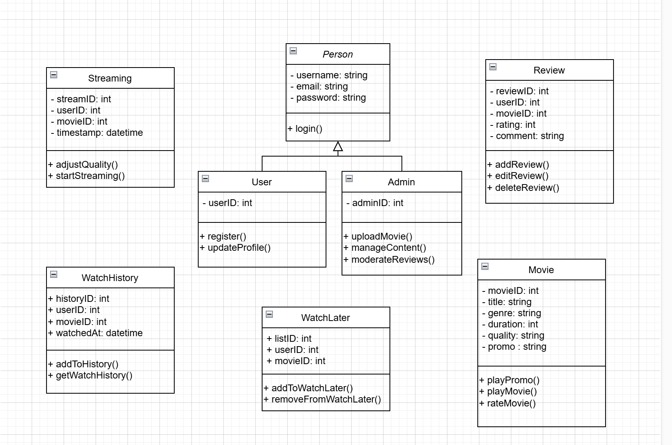
* Develop the user interface using HTML5, CSS3, Bootstrap and JavaScript.
* Ensure responsive and interactive.
* Implement search, filtering, and navigation functionalities.

### 2. Backend Development

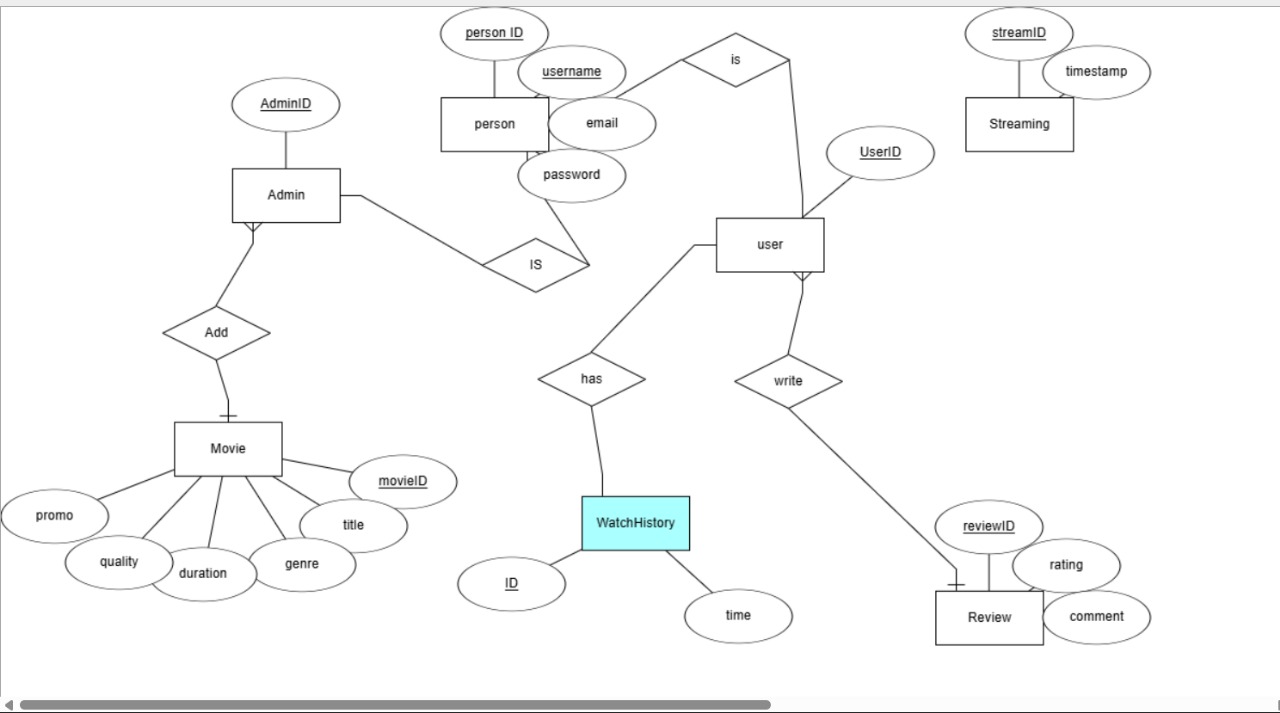
* Develop RESTful APIs for movie streaming, user authentication, and data management.
* Implement business logic and database interactions.
* Ensure security, scalability, and performance optimization.

**4. System Analysis & Design**

**class diagram:**



**ERD:**



# 

# 4. Risk Assessment & Mitigation Plan

### Risk: Difficulty in Backend Development

* Impact: Slow progress in building the API and challenges in managing databases.
* Mitigation:
  + Use **.NET Web API with Entity Framework** to simplify database operations.
  + Searching and getting the advantage of AI tools.

### Risk: Frontend Development Challenges

* Impact: Struggles with implementing UI components and responsiveness.
* Mitigation:
  + Use **Bootstrap** for responsiveness instead of writing custom CSS from scratch.

### Risk: Time Management & Burnout

* Impact: Delays in project completion due to lack of time or motivation.
* Mitigation:
  + Use **Notion** to break tasks into smaller steps.
  + Set **realistic daily or weekly goals** instead of rushing.

**Risk:** **Bugs and Debugging Struggles**

* Impact: Wasting time trying to fix issues without understanding them.
* Mitigation:
  + Use **debugging tools** (e.g., Visual Studio Debugger, Postman for API testing).
  + Search for solutions on **Stack Overflow** and Benefit from **AI** Tools.

### Risk: Lack of Experience in Team Collaboration

* Impact: Difficulty in merging code changes and tracking progress.
* Mitigation:
  + Use **GitHub** for version control (learn how to commit, push, pull, and merge).
  + Communicate daily to avoid conflicts and stay aligned.

### ****3. Requirements Gathering****

#### **Stakeholder Analysis**

#### The main stakeholders include:

* **End Users**: Individuals who will browse and stream movies. They need a user-friendly interface, high-quality streaming, and a smooth search experience.
* **Content Providers**: Movie studios and distributors who want a secure and well-organized platform to showcase their content.
* **Administrators**: Responsible for managing content, monitoring user activity, and handling technical issues.
* **Developers & Designers**: Team members working on building and maintaining the platform, ensuring performance, scalability, and security.

#### **User Stories & Use Cases**

User stories and use cases describe how different users will interact with the platform.

* **User Story 1** (New User Registration): As a new user, I want to sign up using my email or social media so that I can access the platform.
* **User Story 2** (Browsing & Searching): As a user, I want to search for by genre, rating, or release date so that I can quickly find content I like.
* **User Story 3** (Watching Content): As a user, I want to stream videos in different quality options so that I can adjust based on my internet speed.
* **User Story 4** (Watch Later): As a user, I want to save movies to a "Watch Later" list so that I can find them easily when I have time to watch.
* **User Story 5** (Admin Content Management): As an administrator, I want to upload and update movies so that I can keep the platform’s content fresh and relevant.

#### **Functional Requirements**

These define the core functionalities of the platform:

* **User Management**: Registration, login and profile management.
* **Content Browsing**: Search, filtering, and categorization of movies.
* **Streaming**: Video playback with adaptive quality selection (SD, HD, 4K).
* **User Engagement**: reviews, and rating system.
* **Watch History**: Tracks recently watched movies for easy access.
* **Watch Later List**: Allows users to save content for future viewing.
* **Admin Panel**: Content upload, management, and moderation.
* **Security**: Secure authentication, access control, and data encryption.

#### **Non-functional Requirements**

These ensure performance, security, and usability:

* **Performance**: Fast loading times and seamless video playback.
* **Scalability**: Ability to handle a large number of users and content.
* **Security**: Data encryption, secure authentication, and protection against cyber threats.
* **Usability**: Simple, intuitive, and visually appealing UI for a smooth user experience.
* **Reliability**: The platform should be available 24/7 with minimal downtime.