

# Capstone - Project Proposal, Updated

---

## Problem Description

Everyone likes to share what they've most recently been watching or reading – even more so when someone asks for a suggestion of what they should check out next. When it comes to keeping track of these favorite movies, TV shows, or books, many resort to manually kept notes on scattered paper, notebooks, random word documents, online bookmarks, or memory alone. These currently available methods often result in lost or forgotten recommendations.

Users would benefit from an easy and intuitive way to store and organize their favorite media, readily accessible when someone asks, "What should I watch/read next?" Existing solutions either lack social integration or provide overly complex user experiences.


This project will address these issues by providing an interactive, socially connected application where users can maintain and share customized lists of their favorite media. The primary users include everyone from media enthusiasts, book clubs, casual viewers, and readers who regularly exchange recommendations with others, to pretty much anyone living in the modern age. The key value derived from solving this problem is convenience and improved social interaction around shared media interests.

My application, Recommend, will facilitate users in organizing their favorite books, TV shows, and movies. Users can follow others, explore their public lists, and discover new content organically through shared interests. Unlike simple CRUD apps or static websites, this project involves dynamic social features such as user following, public list exploration, and smart content discovery.

## New Feature Additions to Problem Scope

To further enhance the social and discovery aspects of the application, Recommend will include:

- **Genre-Based User Discovery:** Users who interact frequently with the same genres (e.g., favoriting, tagging, or saving similar media) will be recommended to each other, encouraging community-building around shared tastes.

- 
- Content Discovery via Followed Users: Users will receive recommendations and highlights of media items recently saved, highly rated, or frequently interacted with by users they follow. This promotes personalized, socially driven discovery even in the absence of direct interaction.

## Minimum Viable Product (MVP) Description

### High-level Overview


The MVP will allow users to create and organize media lists, follow other users, and explore recommendations influenced by their social network and content preferences. The core interactions include:

- User Registration and Authentication: Secure sign-up and login functionality.
- List Creation and Organization: Users can build custom lists for TV shows, movies, and books.
- User Following System: Users can follow other users to explore their lists and media.
- Public List Browsing: Users can view the public lists of people they follow.
- Media Recommendation Feed: Users see a feed of content influenced by followed users and similar genre interactions.

### Minimal Set of Features

- User registration/login.
- Creation, editing, and deletion of personalized media lists.
- Ability to categorize lists by genre, actor, author, or custom tags.
- Following/unfollowing users.
- Viewing public media lists of followed users.
- Passive recommendation feed based on followed users' media activity.
- Suggested users based on overlapping genre preferences.

### High-level Architecture

- 
- Frontend: Developed using React and Next.js to create a responsive and intuitive user interface that will handle user interactions and state management.
  - Backend Service Layer: Implemented using Next.js API routes for server-side logic, handling authentication, data manipulation, and business logic.
  - Database: Supabase, leveraging PostgreSQL, will provide persistent storage for user data, media lists, social relationships, and recommendation interactions.

## Data Management Overview

Data stored within the application will include:

- Users: User ID, name, email, password hash, profile information.
- Media Items: Item ID, type (book/movie/show), title, genre, author/actor/director, description, image URL.
- Lists: List ID, user ID (owner), title, description, tags, and associated media items.
- Follow Relationships: Follower ID, followed user ID.
- Recommendations: System-generated feed items linked to followed users or genre overlap.
- Genre Interactions: User ID, genre ID, interaction type (e.g. list addition, like), timestamp.

