Project Title

Personalized Recommendation Engine (Media) Web App

Sprint 2 Submission

This documentation covers progress and deliverables completed **up to Sprint 2** of the project. So far, our focus has been on developing the **web application interface** and implementing **user authentication (login system)**.
The **main recommendation engine and media suggestion features** will be developed in the **next sprints (Sprint 3)**.

Problem Statement

Students and everyday users often spend too much time searching for videos or songs they like. Our project aims to solve this by creating a web-based platform that will recommend media based on user preferences and activity (e.g., YouTube or Google account history).

Intended Users

- Students
- Music and video enthusiasts
- Anyone seeking quick and personalized recommendations

Sprint 2 Goals / Objectives

For this sprint, our team focused on building the **foundation of the web app**, including:

- Designing and developing the **frontend interface**.
- Setting up the **user login system** with secure authentication.
- Establishing basic **backend and database connections**.
- Ensuring smooth **deployment and hosting** of the web app.

Sprint 2 Deliverables

- Fully functional **web application UI** (home, login, and dashboard page).
- **User authentication system** with secure login and signup using bcrypt.
- **Database connection** via Firebase/SQLite for user data storage.
- **GitHub** repository for collaborative development.
- **Jira board** for sprint task management and progress tracking.

Planned for Next Sprint (Sprint 3)

In the next sprint, we will focus on implementing the **core recommendation engine**, including:

- Generating **personalized video and song recommendations**.
- Simulating **user media history** with mock datasets.
- Displaying **related content suggestions** in the UI.
- Integrating **basic backend logic** for recommendation processing.

Tools / Technologies

- **Frontend:** HTML, CSS, JavaScript (served through Express)
- **Backend:** Node.js with Express
- **Database:** Firebase / SQLite
- **Authentication:** bcrypt for secure password handling
- **Project Management:** Jira (task tracking), GitHub (version control)

Risks / Challenges

- **Data Access:** Real Google/YouTube history data is restricted; mock data will be used instead.
- **Cold Start Problem:** Limited recommendations for new users without activity data.
- **Team Coordination:** Managing parallel frontend and backend tasks among multiple team members.