

Date: 4/11/2025

1. Decisions & Tasks

- **Decisions Made:**

- The project will focus exclusively on movies, with the core feature being the upload of a movie collection list that will be sent via API to ChatGPT for generating personalized recommendations.
- The community and social interaction features will be excluded from the current version to keep the design simple and streamline development.
- The upload functionality will support multiple file formats (e.g., CSV, JSON).
- Security and compliance requirements (such as GDPR and copyright compliance) must be integrated into the system design from the start.

- **Action Items:**

- **Developer:**

- Implement the movie collection list upload feature (supporting CSV/JSON).
- Integrate the ChatGPT API to process the uploaded data and return a recommendation list.
- Develop error-handling mechanisms to manage data inconsistencies or incomplete data.

- **Client:**

- Provide sample movie collection files for testing. *(Due: [Insert Due Date])*
- Review and approve the API prompt design and recommendation output format. *(Due: [Insert Due Date])*

2. Requested Features, Changes, & Clarifying Questions

- **Requested Features/Changes:**

- A user-friendly interface to upload movie collection lists.
- A recommendation engine powered by ChatGPT that generates personalized movie lists based on uploaded data.
- Support for different file formats to accommodate user preferences.
- A streamlined UI that focuses on movie management and recommendation without the complexities of a community forum.

- **Clarifying Questions Discussed:**

- **File Format & Data:**

- What specific movie attributes (e.g., title, rating, genre, viewing date) should be extracted from the upload file?
- How should the system handle missing or malformed data in the file?

- **API Integration:**

- What level of detail is expected in the recommendation output (e.g., brief descriptions, genre matches, viewing suggestions)?
- Are there any customization options required in the API prompt to refine the recommendations?

- **User Experience:**

- Should users have the ability to re-upload or update their movie collection for periodic recommendations?
- How will the recommendation results be displayed—within the web interface or via email notifications?

3. Expected Timeline & Challenges

- **Expected Timeline:**

- **Week 1:**

- Finalize the project requirements and sample movie list formats.
 - Review and approve the initial API prompt design with the client.

- **Week 2-3:**

- Develop the file upload and parsing module.
 - Begin integrating the ChatGPT API into the backend for processing recommendations.

- **Week 4:**

- Testing the complete workflow: upload, data parsing, API call, and recommendation output.
 - Iterations based on feedback and bug fixes.

- **Week 5:**

- Final testing and deployment of the initial version.

- **Challenges and Potential Issues:**

- **Data Consistency:** Ensuring all movie collection files (across multiple formats) are properly parsed without errors.
- **API Response Time and Accuracy:** Managing potential latency or inconsistencies in the recommendations provided by the ChatGPT API.
- **Compliance and Security:** Implementing all necessary data protection measures and ensuring the application meets regulatory standards from the outset.
- **User Feedback Integration:** Rapidly adapting the system in response to user feedback during early-stage testing.

Next Steps

- **Immediate:**
 - Developer to start prototype development for the file upload module and initial API integration.
 - Client to provide sample data and confirm the recommendation output format.
- **Follow-Up Meeting:** Scheduled for [Insert Next Meeting Date] to review progress and address any emerging issues.