Project Proposal

for

Mirror News Summarizer

Student Name: Zijun Li

Student ID: 2272583

Supervisor Name: Jizheng Wan

Project Category/Topic: AI

1 Project Aim:

- The goal is to develop 'Mirror News Summarizer', an advanced platform that significantly enhances the user experience of consuming news online. It extracts content from selected news sites and, while maintaining a similar format reminiscent of the original website, eliminates advertisements and replaces full articles with succinct summaries to present on a mirror news platform. Furthermore, it offers a dedicated interface for tailored recommendations and a favorites section, ensuring personalized and engaging content for users. This system is meticulously designed to streamline the process of information acquisition, making news reading more efficient and user-friendly.

- **Significance**: In today's digital age, users grapple with the deluge of information, often finding it challenging to decipher extensive content swiftly. The 'Mirror News Summarizer', with its mirror-style approach, elegantly addresses this by presenting concise news articles that encapsulate the essence of the original. This mirroring feature offers users a sense of familiarity and consistency, enhancing readability and reducing decision fatigue. By reducing the often dizzying array of design choices, it enhances readability and combats decision fatigue. Coupled with tailored recommendations and a favorites section, the platform not only ensures efficient news consumption but also elevates the user experience, crafting a more immersive and customized reading journey.

- **Relevance to AI**: The 'Mirror News Summarizer' leverages cutting-edge AI technologies and methodologies for several core functionalities:

  Summarization: Employs Summary APIs, capitalizing on advanced NLP models like GPT (Generative Pre-trained Transformer), to generate precise and coherent news summaries. This AI-driven approach understands the context and extracts essential details, contributing to a succinct yet comprehensive summary.

  Recommendation System: Integrates existing AI recommendation models that analyze user behavior, preferences, and interaction history to provide personalized news content, enhancing user engagement and satisfaction.

  Text Similarity Analysis: Implements AI algorithms for text similarity checks to identify and eliminate duplicate content from different sources, presenting unique content to the users.

2 Literature Review:

In this chapter, we delve into an insightful comparison of some of the leading news recommendation and summarization platforms in the digital realm.

Inshorts[1] has positioned itself as a forerunner in the concise news space. Its backend, fortified by a rich tech stack of 22 products, from HTML5 to Google Analytics, ensures a seamless and responsive experience for its users.[4] On the frontend, its minimalist yet impactful design delivers 60-word summaries that cut through the noise, making news consumption swift and straightforward. The user-centric mobile interface ensures quick access and readability, vital in today ’ s fast-paced world. However, its primary challenge remains its concentrated regional appeal, especially with India as its prime audience. Venturing into this space, the *Mirror News Summarizer* encompasses advanced AI-driven personalization algorithms. Not only does it aggregate news, but it also tailors content based on individual preferences, enhancing global reach and user engagement.

Feedly[2] is a master curator, with a technologically robust foundation that comprises 30 essential products, ensuring optimal content delivery from varied sources.[5] Its interface is a blend of simplicity and categorization, allowing users to tailor their feeds based on interests. Yet, where it excels in aggregation, there’s room for improvement in content condensation. As users demand both breadth and brevity, the *Mirror News Summarizer* bridges this gap. By employing advanced algorithms, it not only collates content but also presents it in easily digestible chunks, ensuring that users receive depth without the associated reading time.

Flipboard[3] is the epitome of visual storytelling in the digital news domain. Supported by an expansive tech stack and 182 website technologies, it offers a fluid, magazine-esque reading experience.[6] The emphasis on visual elements, combined with its robust technological deployment, ensures users are always engaged. However, the focus on visuals sometimes comes at the cost of brevity. Recognizing this, the *Mirror News Summarizer* marries visual appeal with content succinctness. The result is a platform that offers visual richness, but without compromising on the essence of the news.

Lastly, the *Mirror News Summarizer* emerges as a holistic solution. It’s not just about presenting news but doing so in a manner that resonates with the user’s preferences and pace. Advanced AI algorithms, combined with user feedback loops, ensure that the platform learns and evolves with each user interaction. The addition of features like a ’favorites’ section adds a layer of personal touch, ensuring users always have their preferred content at their fingertips.

3 Project Objectives

1. **Interview & User Persona Analysis**

   Understand the unique needs, preferences, and pain points of potential users of the Mirror News Summarizer through comprehensive interviews and persona analysis.

   - **Feature List Compilation:**
     - Quick News Summaries
     - Personalized User Interface
     - Aesthetic and User-Friendly Design
     - Intelligent Content Filtering
     - Distraction-Free Reading Mode
     - Reliable News Sources Integration

2. **Agile Development Process**

   - **MVP (Minimum Viable Product)**
     - Develop a news summary tool that offers concise overviews of major news topics by mirroring a specific website and successfully implementing web scraping for that site.
     - Implement a reliable and accurate news aggregation algorithm.
     - Establish a basic recommendation system to curate news articles for individual users.
   - **Iterative Enhancements:**
     - Personalized news recommendation engine that evolves based on users' reading habits.
     - Bookmarking feature, allowing users to save and return to articles at their convenience.
     - Ad-free and distraction-free user interface to enhance readability and user experience.
     - Categorization and sorting options for saved articles, providing better organization.
   - **Feedback & Iteration:**
     - Continually gather feedback from the user personas (and broader user base) to refine and improve features.
     - Prioritize feature developments based on both user feedback and technological advancements in the field.

3. **Test & Evaluation**

    - **Continuous Integration (CI):**
        - Implement CI tools to ensure code quality and to detect issues early.
        - Regularly test new features in isolation before merging them with the main application.
    - **Continuous Deployment (CD):**
        - Automated deployment of the application to a production environment after passing CI tests.
    - **User Acceptance Testing (UAT):**
        - Before major releases, conduct UAT with real users to ensure the application meets their needs and expectations.
        - Adjust features based on feedback from UAT.
    - **Debugging:**
        - Regularly monitor application logs and user feedback to identify and fix bugs.
        - Prioritize bugs based on their impact on user experience and application stability.

4 Project Deliverables

1. User-Centric News Access and Customization

    - Develop a user authentication and login system, ensuring secure and consistent access to personalized news settings across different devices.
    - Develop a user interface within the 'Mirror News Summarizer' that allows individuals to choose their preferred news sources and categories, enhancing personalized interaction.
    - Implement a content filtration and retrieval system that curates and presents news from selected mirror news websites, optimizing the influx of information based on user preferences.

2. Enhanced Content Presentation

    - Design an intuitive layout to display summarized news content, incorporating essential details like the headline, source, publication date, and core points, all within a user-friendly, visually appealing framework.

- Establish a mechanism for users to customize the news summary display, including aspects like theming (dark mode, font choices) and organization (list view, card view), further improving user experience and accessibility.

3. Intelligent Content Summarization and Duplication Check

   - Execute an AI-driven process for dynamic news content summarization, ensuring that the essence and critical points of the full articles are cohesively maintained in the shortened version.
   - Integrate a text similarity analysis feature that identifies and filters out repetitive news from different sources, thereby avoiding content redundancy and enhancing the quality of content presented to the user.

4. Adaptive Recommendation and Interaction System

   - Construct a recommendation engine that analyses user interactions, preferences, and trending news to suggest relevant content, thereby personalizing the user's news feed and improving content relevancy.
   - Develop interactive components within the platform, such as feedback options for summaries and bookmarking capabilities for preferred news articles, fostering user engagement and platform loyalty.

5. Voice Integration and Multilingual Support

   - Integrate a text-to-speech feature, enabling users to listen to news summaries, thereby accommodating different user needs and enhancing accessibility.
   - Incorporate a multilingual interface, allowing the platform to serve a broader demographic, catering to users from various linguistic backgrounds, and expanding the platform's reach.

6. User Engagement and Analytic Reports

   - Implement a system for tracking user engagement metrics, preferences, and feedback, utilized for continuous improvement of the platform's features and user experience.

5 Methodologies

1. **User Authentication and Login**: Develop a secure user registration and login mechanism using technologies like JWT (JSON Web Tokens) and OAuth. This step ensures a protected, personalized space for each user and sets the foundation for subsequent personalized interactions.

2. **User Preferences and Customization Setup**: Initiate user interaction by offering a setup interface where they can select preferred news categories, sources, and UI/UX elements (e.g., dark mode, text size). This step personalizes the subsequent content curation.

3. **Content Aggregation and Filtration**: Utilize web scraping tools (like requests) and APIs to aggregate news content from various mirror websites selected by the user. This process includes filtering out ads and non-essential elements, focusing solely on the actual news content.

4. **AI-driven News Summarization**: Implement an AI model API to generate concise, coherent summaries of the aggregated news articles. This step requires maintaining the integrity and critical nuances of the original news content while reducing the length.

5. **Duplication and Redundancy Removal**: Apply text similarity algorithms to identify and eliminate duplicate content across multiple news sources, ensuring a unique, clutter-free user feed.

6. **Dynamic Content Presentation**:Display the summarized content to users in a structured, customizable UI, allowing for different viewing preferences (e.g., list view, card view). This stage involves frontend technologies like React for a responsive design.

7. **User-Interactive Recommendation System**: Integrate a recommendation system that tracks user engagement (clicks, reading time) and utilizes existing models to suggest relevant news, enhancing the personalization aspect of content delivery.

8. **User Feedback and Continuous Improvement**: Allow users to provide feedback on news summaries and recommendations, which the system will use for continuous learning and improvement. This iterative process is crucial for maintaining content relevance and accuracy.

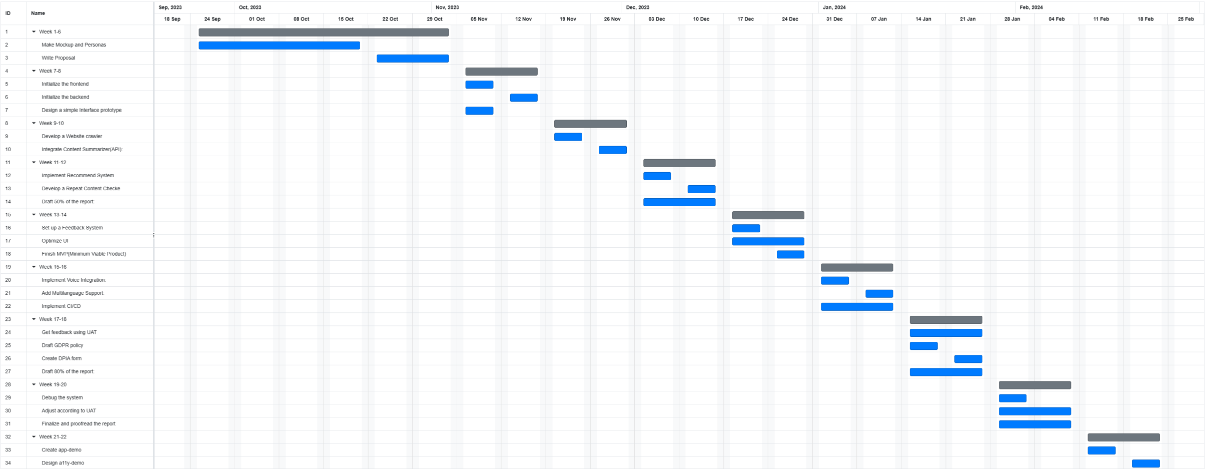
9. **Text-to-Speech (TTS) and Multilingual Support Integration**:Incorporate TTS functionality for users to listen to summaries, enhancing accessibility, and convenience. Additionally, implement multilingual support to cater to a diverse user base.

10. **Backend Support and Data Management**: Use backend technologies like Flask for managing data transactions, user requests, and system responses, ensuring a seamless, efficient backend operation.

11. **Evaluation and User Testing**: Conduct comprehensive user testing to evaluate the system's performance, particularly the accuracy of news summaries, effectiveness of the recommendation engine, and overall user satisfaction. Collect feedback for future enhancements.

6 Project Plan

Week	Dates	Tasks
Week 1-6	9.25 - 11.5	- Make Mockup and Personas - Write Proposal
Week 7-8	11.6 - 11.19	- Initialize the frontend: Set up tools and frameworks - Initialize the backend: Set up server and database - Design a simple Interface prototype
Week 9-10	11.20 - 12.3	- Develop a Website crawler: Define sources and extractors - Integrate Content Summarizer(API): Set up API calls and responses
Week 11-12	12.4 - 12.17	- Implement Recommend System: Design algorithms and interfaces - Develop a Repeat Content Checker: Set up detection algorithms - Draft 50% of the report: Define structure and content
Week 13-14	12.18 - 12.31	- Set up a Feedback System: Design UI and backend handlers - Optimize UI: Improve navigation menu, Redesign key pages - Finish MVP(Minimum Viable Product)
Week 15-16	1.1 - 1.14	- Implement Voice Integration: Choose voice API and integrate - Add Multilanguage Support: Define languages and integrate translation APIs - Implement CI/CD: Set up pipelines and deployment strategies
Week 17-18	1.15 - 1.28	- Get feedback using UAT - Draft GDPR policy: Define data collection and usage - Create DPIA form: Define risk assessment criteria - Draft 80% of the report: Refine content and gather feedback
Week 19-20	1.29 - 2.11	- Debug the system: Set up testing environments and fix issues - Adjust according to UAT - Finalize and proofread the report
Week 21-22	2.12 - 2.25	- Create app-demo: Showcase key features and functionalities - Design a11y-demo: Highlight accessibility features and improvements



ID	Name
1	Week 1-6
2	Make Mockup and Personas
3	Write Proposal
4	Week 7-8
5	Initialize the frontend
6	Initialize the backend
7	Design a simple Interface prototype
8	Week 9-10
9	Develop a Website crawler
10	Integrate Content Summarizer(API):
11	Week 11-12
12	Implement Recommend System
13	Develop a Repeat Content Checke
14	Draft 50% of the report:
15	Week 13-14
16	Set up a Feedback System
17	Optimize UI
18	Finish MVP(Minimum Viable Product)
19	Week 15-16
20	Implement Voice Integration:
21	Add Multilanguage Support:
22	Implement CI/CD
23	Week 17-18
24	Get feedback using UAT
25	Draft GDPR policy
26	Create DPIA form
27	Draft 80% of the report:
28	Week 19-20
29	Debug the system
30	Adjust according to UAT
31	Finalize and proofread the report
32	Week 21-22
33	Create app-demo
34	Design a11y-demo

7 Risks and contingency plan

- **Risk**: Web scraping tools often depend on the consistent structure of a webpage. If major news sites undergo a significant design overhaul, the scraping tools developed for the 'Mirror News Summarizer' might fail to aggregate content effectively.

  Contingency: Should our primary targeted news sources change their webpage structures, we plan to quickly adapt our scraping scripts. Furthermore, we'll set up periodic monitoring tools to check the efficiency of our scrapers and get alerts for potential failures.

- **Risk**: News websites frequently employ anti-scraping technologies and techniques to prevent automated bots from accessing their content. This can hinder the consistent inflow of news data to the platform.

  Contingency: In the event we encounter difficulties accessing content from a particular source, we'll pivot to integrate RSS feeds or official news APIs, which offer structured and consistent access to news articles.

- **Risk**: Third-party APIs often have usage limits or rate limits that could restrict the extent and frequency of operations, affecting the performance of our summarization system if the project scales up.

  Contingency: If we approach the API rate limits, our plan is to integrate caching mechanisms to store frequently accessed summaries. Additionally, we'll explore the possibility of incorporating other summarization algorithms or tools to diversify our backend and reduce dependency on a single API.

8 Hardware/Software Resource

1. Hardware:
   - RAM: 4 GB
   - Storage: 50 GB
2. Software Requirements:
   - OS: Windows
   - Python 3.8 or later
   - Text Editor or IDE: VS Code/PyCharm
   - Browser: Google Chrome/Edge
   - Database Management System: MySQL/PostgreSQL/SQLite
   - Backend: Flask
   - Frontend: React.js
   - Web scraping tools: Beautiful Soup
   - Summary API key: OpenAI API
   - Version Control: Git

9 Data

- The primary data for the project comprises news articles and content fetched from various online news websites through web scraping techniques.
- The data includes textual content, publication dates, author names, and other relevant metadata associated with each news article.
- User interaction data, such as reading preferences, feedback on summaries, and browsing history, will be collected through the platform to enhance the recommendation system.
- Ethical considerations and privacy compliance measures are in place for handling user data, with protocols established for secure data storage, retrieval, and processing.

References

[1] Inshorts. (n.d.). Retrieved from https://www.inshorts.com/en/read

[2] Feedly: Discover. (n.d.). Retrieved from https://feedly.com/i/discover

[3] Flipboard. (n.d.). Retrieved from https://flipboard.com/

[4] Inshorts - Technology. (n.d.). Crunchbase. Retrieved from https://www.crunchbase.com/organization/news-in-shorts/technology

[5] DevHD - Technology. (n.d.). Crunchbase. Retrieved from https://www.crunchbase.com/organization/devhd/technology

[6] Flipboard - Technology. (n.d.). Crunchbase. Retrieved from https://www.crunchbase.com/organization/flipboard/technology