Project Proposal - Team 1

ELMO

Project Overview

In today's fast-paced world, staying informed can feel overwhelming. News is scattered across countless sources, often requiring readers to sift through multiple articles with repetitive information just to get a complete picture. That is where ELMO comes in.

The Evolving Learning Media Outlet is an AI-powered platform designed to streamline news consumption based on your preferences and interests. Leveraging AI, ELMO scans multiple sources, removes redundant details, and delivers a concise, structured and comprehensive summary, ensuring you get all the key information without repetition.

Whether you prefer quick digestible updates, standard reports or detailed insights, ELMO lets you choose your preferred level of detail. Beyond summaries, users can ask ELMO specific questions to explore topics in more depth.

With its personalized, user-friendly, and efficient approach, ELMO makes staying informed effortless, giving you exactly what you need, when you need it.

Project Scope

Core Functionalities

- User authentication and account creation system
- Explore Page featuring:
    - Dynamic trending topics customized to user preferences
    - Topic/keyword/industry tracking capabilities
    - Sub-category organization
        - Ex: Local News, College News, World News, Tech, Stocks, Sports, etc.
- Following/Saving System including:
    - Personal article library
    - Timeline view of saved topic updates
    - Distinction between cumulative and new information
- Custom Article Creation featuring:
    - Multi-source content aggregation into cohesive articles
    - Trusted source integration (CNN, BBC, ABC News, NYT, etc.)

        - Source filtering system (blacklist/whitelist functionality)
    - Content organization with subsections and source hyperlinks
    - Redundancy elimination
    - Query-based article generation
    - Article simplification feature to cater to varying audiences

Stretch Goals

- Accessibility features:
    - Text-to-speech functionality
    - Adjustable reading modes
- Visual news section with images and videos plus captions
- Multi-topic categorization with dedicated pages
- User notification system:
    - Major update alerts for followed topics
    - Weekly summary digests
- Gamified reading incentives for user engagement

Project Objectives

1. **User Authentication & Account Management:**
    - Implement a secure user authentication and account creation system using AWS Cognito to ensure data privacy and seamless access.
2. **Explorer Page Development:**
    - Design a dynamic Explorer Page that delivers tailored content based on user-defined preferences, including topics, keywords, and industries.
    - Introduce sub-categories (e.g., Local News, Tech Industry, Stocks) for a refined browsing experience.
    - Implement a Following/Saving feature to allow users to track and revisit content over time.
3. **Content Aggregation & Custom Article Creation:**
    - Develop an aggregation tool to compile information from multiple trusted sources while filtering redundant or low-quality content.
    - Allow users to blacklist/whitelist sources for enhanced customization.
    - Enable automated article curation for a streamlined reading experience.
4. **Performance & Reliability Goals:**
    - Achieve 90% accuracy in topic filtering and content aggregation.
    - Maintain an API response time under 200ms for content retrieval.
    - Ensure the Explorer Page is loaded with an abundance of information as well as load the page in a short period.

    - Have the articles be able to have real-time content updates.
5. **Deliverables & Expected Outcomes:**
    - A fully functional web application with an intuitive user experience.
    - A well-structured Explorer Page displaying trending and personalized content.
    - A secure authentication system ensuring user privacy and data protection.
    - Comprehensive documentation covering setup, usage, and maintenance.

Specifications

User Interface (UI) Design

- Platform: Web-based application
- Key Screens/Pages:
    - Login/Create Account Pages
    - Home Page with Saved Articles
    - Explorer Page for trending topics
    - Profile Page for user information
    - Personalized News Feed
- User Interaction Elements:
    - Search query interface
    - Article saving/following buttons
    - Source filtering controls
    - Profile editing tools
    - Category navigation tabs

Backend & APIs

- User authentication system with AWS Cognito
- News aggregation service
- Database structure for user profiles and saved articles
- API integration with multiple news sources
- Content merging and redundancy elimination algorithms
- Search query processing system

Machine Learning / AI (If Applicable)

- Planning to use DeepSeek locally so we can use our own data to train the model and output with minimum bias.
    - We will use the Ollama API to input a JOSN file with an engineer prompt and have it output a PDF that will be inputted straight into our article API.

    - Using the UTD servers provided by the professor.
- If DeepSeek isn't viable, we will look into OpenAI and Anthropic API to train data and do the same.
- Implementation of Retrieval Augmented Generation built on top of the designated LLM to allow for the model to utilize up-to-date information past the model cut off date, allowing for it to be able to find and utilize articles as they are released

Tech Stack

- **Frontend:** React
- **Backend:** Node.JS
- **Database:** AWS S3, DynamoDB, and RDS
- **AI/ML (if applicable):** Deepseek (alt. OpenAI, Anthropic API, Gemini)
- **Cloud & Hosting:** AWS EC2 (UTD Servers)

Hardware Requirements

- Computers/laptops capable of running development environments
- Stable internet connection for cloud services and API testing

Software Requirements

- Development Tools:
    - Visual Studio Code
        - With extensions for:
            - React
            - Node.js
            - AWS Toolkit
            - ESLint
            - Prettier
- Postman for API testing
- Git for version control
- Framework Requirements:
    - Node.js (Latest LTS version)
    - npm (Node Package Manager)
- Design Tools:
    - Figma for UX design

Project Timeline

Project Proposal - Team 1

Sprints	Duration	Frontend	Backend	General
Sprint 1	[01/31 - 02/13]	Research elements of project	Research elements of project	Define scope, research and setup
Sprint 2	[02/13 - 02/20]	Research ideas; look into color scheme, branding; work on design flow and begin low-fidelity wireframes	AWS set up, AWS Amplify, API testing	Further establish team workflow
Sprint 3	[02/20 - 02/27]	Refine UI/UX with more detailed Figma wireframes	Set up AWS Cognito authentication, set up Lambda, set up DynamoDB tables	Conduct user testing wireframes for design feedback
Sprint 4	[02/27 - 03/06]	Refine and finalize UI/UX Figma prototypes; Start frontend development	Further testing with APIs and implement that, populate tables with data during testing, have a working prototype through AWs and Postman, Continue developing the model	Begin model development, ensuring AI framework selection aligns with goals
Sprint 5	[03/06 - 03/13]	Continue frontend development; look into preparing reusable components in React; implement login/signup pages and navigation	Integrate login/logout with frontend and ensure user authentication is validated, train AI model	Valid user authentication
Sprint 6	[03/13 - 03/27]	Implement home page, explore page, article view page	Implement news aggregation logic, fetch and process data from news APIs	Validate API accuracy and latency, refine schema
Sprint 7	[03/27 - 04/03]	Implement saved article and chatbot	Develop query-based AI	Conduct unit testing for model

		page	article summarization and content filtering system	outputs
Sprint 8	[04/03 - 04/10]	Finalize frontend, refine as needed and ensure overall compatibility	Finalize testing	Conduct final debugging
Sprint 9	[04/10 - 04/17]	Polish as needed	Final testing	Finalize
Sprint 10	[04/17 - 04/24]	Polish as needed	Polish as needed	Finalize
Sprint 11	[04/24 - 05/01]	Polish as needed	Deploy final model and note documentation	Finalize
Sprint 12	[05/01 - 05/08]	Prepare for final presentation	Prepare for final presentation	Prepare for final presentation and report submission

Sprints	Duration	Frontend	Backend	General
Sprint 1	[01/31 - 02/21]	Figma Designs & Research	Research & API Testing	Define scope, research, and setup
Sprint 2	[02/21 - 03/07]	UI/UX design	API setup	Data collection
Sprint 3	[03/07 - 04/04]	Development	Development	Initial implementation
Sprint 4	[04/04 - 04/25]	Integration	Integration	Testing
Sprint 5	[04/25 - 05/09]	Final testing	Final testing	Deployment, presentation preparation, and final presentation

Team Leader Rotation

Project Proposal - Team 1

Date of Meeting	Team Leader
02/13	Shaz
02/20	Shaz
02/27	Avanthi
03/06	Avanthi
03/13	Kshitij
03/27	Kshitij
04/03	Abel
04/10	Abel
04/17	Isaac
04/24	Isaac
05/01	Shaz
05/08	Shaz

Project Team

Role	Team Member	Responsibilities
Frontend Developer	Primary: Shaz, Avanthi Secondary, Kshitij	UI development
Backend Developer	Primary: Abel, Secondary: Avanthi, Kshitij	API & Database
AI/ML Engineer	Primary: Isaac Secondary: Shaz, Kshitij	AI/ML models
QA Tester	Primary: Kshitij Secondary: Isaac, Abel	Testing & validation

Project Proposal - Team 1

Links

- **GitHub Repository**
- **Agile Board**
- **Design Document (Figma)**
- **Meeting Minutes**