

# CS 510 Project Proposal

## Team 009

**Title:** Topic Modeling and Bookmarking Google Extension

**Group:**

Lahari Anne : [lanne2@illinois.edu](mailto:lanne2@illinois.edu)

Sanjay Raj Aerra : [saerra2@illinois.edu](mailto:saerra2@illinois.edu)

Sasi Pavan Chowdary Surapaneni : [ss257@illinois.edu](mailto:ss257@illinois.edu)

**Co-ordinator:** Sanjay Raj Aerra : [saerra2@illinois.edu](mailto:saerra2@illinois.edu)

**Track:** Development track

**Proposal:**

**Functions and Users:**

We propose to develop a standalone tool, a Google Chrome extension, that provides topic suggestions for bookmarking web pages based on their content. The major functions of the tool include:

- Scanning web page content to suggest relevant topics for bookmarking.
- Performing sentiment analysis to determine the tone of the page.
- Allowing users to search for bookmarks based on topics within the extension.

Our software tool/system users are individuals who frequently bookmark web pages for reference or research purposes. This tool will be particularly useful for students, researchers, professionals, and anyone who wants to organize and categorize their online content efficiently.

**Significance:**

Our tool addresses the pain point of manually categorizing and organizing bookmarked web pages. Many users need help with maintaining an organized bookmark library, leading to inefficiency in retrieving relevant information when needed. By automating the process of suggesting topics for bookmarking and performing sentiment analysis, our tool streamlines the bookmarking process and enhances user productivity. Additionally, our tool promotes better content discovery and retrieval, contributing to a more organized and efficient online experience for users.

**Approach:**

We plan to build a Google Chrome extension using JavaScript, HTML, and CSS for user interface development. For backend functionality and natural language processing tasks such as topic modeling and sentiment analysis, we will leverage Python libraries such as NLTK (Natural Language Toolkit) and scikit-learn. The processed text is also classified into a category

using a machine learning model for intelligent decision making. The extension will communicate with a backend server hosted on a cloud platform (e.g., Google Cloud Platform, AWS) for processing heavy tasks. We will also explore existing APIs and libraries for topic modeling and sentiment analysis to streamline development. Potential risks include integrating complex NLP algorithms within the extension and ensuring the scalability and efficiency of the backend server. We plan to mitigate these risks by conducting thorough testing, optimizing code performance, and seeking assistance from experts in the field if needed.

**Evaluation:**

To demonstrate the usefulness and correctness of our tool/system, we will conduct user testing and gather feedback from users on the effectiveness of topic suggestions, sentiment analysis accuracy, and overall usability of the extension. We will also evaluate the performance of the backend algorithms by comparing the suggested topics and sentiment analysis results with manual categorization and human assessment. We will also compare different model's efficiency in topic modeling. Additionally, we will measure user satisfaction and efficiency metrics such as time saved in bookmarking and retrieving web pages.

**Timeline:**

Week 1: Research and familiarization with NLP libraries, and Chrome extension development environment setup. Frontend development of the Chrome extension, implementation of topic modeling, and sentiment analysis algorithms.

Week 2-3: Backend server setup, integration with the Chrome extension, testing, and debugging. User testing and feedback collection, refinement of features based on user feedback. Final testing, optimization, and preparation for release.

Week 4: Release the Google Chrome extension to the Chrome Web Store. Report and video presentation.

**Task Division:**

Team Member 1: Frontend development of the Chrome extension, user interface design, and topic modeling implementation.

Team Member 2: Backend development, implementation of sentiment analysis algorithms, and backend server setup.

Team Member 3: Integration of frontend and backend components, testing, debugging, user testing coordination, and final optimization.