Project Proposal

Software Engineering CS-360

NoteGenius

Group-35

Dr. Naveed Arshad
Zaeem Rizwan

Submitted in partial fulfillment

of the requirements of a

Software Engineering course project

3rd February, 2025

Maintain version history here

VERSION 1.0.0

Project Proposal

VERSION HISTORY				
VERSION	APPROVED BY	REVISION DATE	DESCRIPTION OF CHANGE	AUTHOR
1.0.0				

Table of Contents

1.0 Overview

1.1. Purpose

Students often struggle with switching between multiple apps for note-taking and AI-powered enhancements, disrupting their workflow. Our project eliminates this hassle by integrating AI directly into a single note-taking application. Instead of using separate tools for organization, summarization, and beautification, our app seamlessly combines conventional note-taking with AI-driven features. With automatic topic classification, summarization, and enhanced readability, students can effortlessly create structured and visually appealing notes—all within one unified platform. This ensures a smoother, more efficient learning experience without the need to toggle between different applications. Other than the AI aspect of the application, it would also provide a rich text experience, clutter free environment, and an objective view to help the user understand. The project requires basic hardware like a dual-core CPU, 4 GB RAM, and 10 GB of storage, with an optional GPU for faster AI processing. Deployment will use cloud services like AWS Free Tier or Heroku, and databases like MongoDB Atlas. Software components include a React.js frontend and a Node.js/Express or Python/Flask backend, with AI support from Hugging Face Transformers. MongoDB will manage data, while Cloudinary handles media storage.

1.2. Project Scope

1.2.1. In Scope

The software will provide a comprehensive note-taking experience, allowing users to create, edit, delete, and share notes with ease. It will support exporting notes to PDF for convenient storage and sharing. Additionally, multimedia support will enable users to include both images and text in their notes. To enhance accessibility, a text-to-speech feature will be included, allowing users to listen to their notes. The

software will also integrate AI-powered functionalities, such as auto-tagging for efficient organization, AI prompt modifications to refine user input, and AI-generated summaries to provide concise overviews

1.2.2. Out of Scope

The project explicitly excludes functionalities related to drawings and handwriting using fingers or a stylus. Additionally, automatic or AI-assisted beautification, including color and font adjustments, will not be provided. Furthermore, real-time processing of AI API requests may be difficult because hosting free models locally can cause delays, and free inference API calls often come with token limitations, restricting their efficiency.

1.3. Project description

1.3.1. Project Goals and Objectives

The primary goal of this project is to develop an AI-powered note-taking application that significantly enhances how students and professionals capture, organize, and review notes. By incorporating artificial intelligence, the application will make note-taking more efficient, structured, and accessible, addressing common challenges with manual note organization, readability, and summary creation.The app will cater to a diverse range of users, from university students and professionals to general users looking for an AI-enhanced note management system.

While there are many note-taking apps available (e.g., Evernote, Microsoft OneNote, Google Keep), few offer intelligent AI features that enhance the note-taking experience. Our app's automatic topic classification and summary generation among other features provide distinct value that existing solutions do not. Users will not only take notes but also gain AI-driven assistance that optimizes their study or work habits, making the app an indispensable tool for better information management.

1.3.2. Business Drivers

Business Driver #1: Efficient Note-Taking

Issue: Typing notes manually is a time-consuming process, and constantly switching between different apps (note-taking, research, grammar correction, etc.) adds to the inefficiency. Users need a more seamless workflow that doesn't disrupt their note-taking process.

Solution: NoteGenius’s UI integrates LLMs directly within the note-taking experience, allowing users to interact with the model and enhance their notes without having to leave the app or switch screens.

The app will incorporate multiple AI-driven features that adapt and learn to user needs without the need to switch apps. For instance, after a note is taken, AI-based auto-tagging will categorize content automatically, making it easier to locate key topics. Additionally, real-time grammar correction and contextual assistance

will be integrated to modify and enhance the content on the fly. Users can select text within the note and, with one click, send it to an AI model for revision or expansion, all while staying within the same application. This reduces the need for users to open external tools, improving productivity and the overall note-taking experience.

Moreover, AI-driven summarization will allow users to condense lengthy notes into concise summaries, so they can quickly grasp the key points without manually editing or rewriting. These features will be learned and adapted to user behavior over time, offering personalized improvements and saving time in the long run, all without the hassle of switching apps.

Business Driver #2:Accessibility

Issue: Some users prefer auditory learning over reading, which can make text-based note-taking applications less accessible for them. This preference can be a barrier for individuals with visual impairments, those who prefer multitasking, or anyone who finds listening more effective for retention and understanding.

Solution: NoteGenius offers a **Text-to-Speech (TTS)** feature, enabling users to have their notes read aloud. This auditory experience caters to different learning styles, supports multitasking, and improves accessibility for users who might struggle with traditional text-based interfaces.

1.3.3. Features

Feature #1: Basic Note-Taking

Description: Users can create, edit, save, and delete notes within the application. This feature will allow them to store and manage their notes efficiently.

Feature #2: Export Functionality

Description: Users can export their notes in PDF and Markdown formats for ease of sharing and accessibility.

Feature #3: Multimedia Support

Description: Users can insert and display images within their notes, enhancing the visual appeal and depth of their content.

Feature #4: Auto-Tagging with AI

Description: When a note is saved, it will automatically be categorized based on its content using a pre-trained NLP model (BERT). This allows for automated tagging and categorization of notes.

Feature #5: AI-Generated Summaries

Description: The app will allow users to get concise summaries of lengthy notes using a pre-trained NLP model (BERT). The summary will be displayed on a side panel within the app.

Feature #6: AI-Select and Enhance

Description: Users can select text within the note-taking app, after which they will be prompted to enter a modification request. This prompt will be sent to an AI model, which processes the request and returns a revised version of the text. The modified text will be shown on a side panel for user review.

Feature #7: Text to Speech

Description: The Text-to-Speech (TTS) feature will allow users to have their notes read aloud, providing an auditory learning experience. This will be especially useful for multitasking or accessibility. This will be implemented through the use of existing TTS libraries/APIs.

Feature #8: AI-Powered Contextual Assistance

Description: When the user saves a note, the app will analyze the text and provide additional information related to the topic. For example, writing about the “French Revolution” will prompt the app to show related topics like “Napoleon” for history notes.

Feature #9: Quiz-It

Description: This feature will quiz the user on the topic of that specific note file. A pre-trained NLP model (BERT or T5 small) will be used to generate quizzes.

Feature #10: Search Capability

Description: This feature will allow users to search through the titles, content, and tags of their notes. Code will be written to implement this feature.

Feature #11: Authentication

Description: Verifying the identity of a user before granting access to the app and its features.

1.4. Team profile

Name:Muhammad Hammad Yousaf	
ID: 26100387	
Email: 26100387@lums.edu.pk	
Interests and strengths: Artificial Intelligence, art Team Work, Problem Solving	

Name: Muhammad Tayyab Haider	
ID: 26100275	
Email: 26100275@lums.edu.pk	
Interests and strengths: teamwork, adaptability, creativity, dependability, organization, interpersonal skills, C/C++/python, designing, eager to learn	

Name: Muhammad Rayed	
ID: 26100363	
Email: 26100363@lums.edu.pk	
Interests and strengths: Just chilling,	

Name: Aleena Abbas	
ID:26100109	
Email:26100109@lums.edu.pk	
Interests and strengths: Artificial Intelligence, Research, Design,Problem Solving	

Name:Muhammad Abubakar Minhas	
ID:26100157	
Email:26100157@lums.edu.pk	
Interests and strengths: My Interest in Islam is my strength, protect me almighty allah hu allah,	

1.4.1 Expertise in a specific tool

1.5. Assumptions and Constraints

1) Text-to-Speech functionality may not be supported for all languages
2) The availability of free API calls can vary, potentially impacting the range of features offered.
3) Real-time API calls may result in delays depending on models used

1.6. Project Deliverables

Deliverables include

- Software Project Proposal
- Requirement Specifications
- Design Specifications
- Development Plan
- Test plan
- Demo + source code
- Final document
- Final presentations (showcase)

2.0 Project Organization

Clients:

Dr. Naveed Arshad

Muhammad Zaeem Rizwan

3.0 References

Gramformer : https://github.com/PrithivirajDamodaran/Gramformer
Speech-to-text: https://github.com/MahmoudAshraf97/whisper-diarization
Reference AI Note-Taking App: https://github.com/abhishekHegde2000/ai-note-app/blob/main/README.md

4.0 Definitions

LLM (Large Language Model)	An advanced AI model trained on vast datasets to understand and generate human-like text.
Inference API	A service that allows applications to interact with an AI model to process and generate outputs.
NLP (Natural Language Processing)	A field of AI that enables computers to interpret and manipulate human language.
Auto-Tagging	A feature that automatically assigns relevant tags or categories to content based on its context.
Artificial Intelligence (AI)	Technology that enables machines to perform tasks that would typically require human intelligence, such as problem-solving, learning, and decision-making.
Text-to-Speech (TTS)	A technology that converts written text into spoken words.
Markdown	A lightweight markup language used for formatting text in a structured manner.
Contextual Assistance	AI-powered suggestions that provide relevant information or modifications based on the note's content.
Token Limitation	A restriction in AI models where the number of words or characters processed in a request is capped.
Dual-Core CPU	A processor with two cores (processing units) that can handle tasks simultaneously, improving performance.
Random Access Memory (RAM)	A type of computer memory that stores data and machine code currently being used, enabling quick access and improving performance.
Cloudinary	A cloud service for storing and managing media files

	(images, videos), with features for optimization, transformation, and delivery.
Hugging Face Transformers	A popular library for natural language processing (NLP) tasks, providing pre-trained models for tasks like text generation, summarization, and translation.
Python	Python is a programming language.
Flask	Flask is a lightweight web framework that helps build server-side applications
Express	A minimal web application framework for Node.js, providing tools to manage HTTP requests, routing, and middleware.
Node.js	A JavaScript runtime environment that allows developers to run JavaScript on the server side, outside of a browser.
React.js	A JavaScript library for building user interfaces, especially for single-page applications, by creating reusable components.
Heroku	A platform-as-a-service (PaaS) that allows developers to build, run, and scale applications easily in the cloud.
MongoDB Atlas	A managed cloud database service for MongoDB, a NoSQL database that stores data in a flexible, document-based format.
AWS Free Tier	A free service offered by Amazon Web Services that provides limited resources for cloud computing and hosting.
Cloud Infrastructure	Online platforms that provide computing resources (like storage, databases, and servers) over the internet, instead of using local infrastructure.
Graphics Processing Unit (GPU)	A specialized processor designed to accelerate rendering of images and videos, often used for tasks involving complex computations like AI.

Check next page for Marking Rubric

Marking Rubric

Total marks: 45

Component	Marks
Purpose	5
Project Scope	2+2 (in scope + out of scope)
Project Goals and Objectives	5
Business Drivers	5
Features	10
Assumptions and Constraints	2
References, Definitions, Profile and Project Organization	4
Concise and to-the-point descriptions	5
Writing Quality (Descriptions should be coherent and should covers all questions asked under the heading)	3
Following the format (as mentioned under business drivers, features, team profiles and definitions)	2