

Local LLM Project

Submitted By

Student Name	Student ID
Bishwajit Chakraborty	0242220005101414

Section : 63_I1

MINI LAB PROJECT REPORT

This Report Presented in Partial Fulfillment of the course **CSE234: Object Oriented Programming II** in the **Computer Science and Engineering Department**



DAFFODIL INTERNATIONAL UNIVERSITY
Dhaka, Bangladesh

December 16, 2024

DECLARATION

We hereby declare that this lab project has been done by us under the supervision of **Mst. Umme Ayman(UA)** , **Lecturer**, Department of Computer Science and Engineering, Daffodil International University. We also declare that neither this project nor any part of this project has been submitted elsewhere as lab projects.

Submitted To:

Mst. Umme Ayman(UA)

Lecturer

Department of Computer Science and Engineering Daffodil
International University

Submitted by

Bishwajit Chakraborty
0242220005101414
Dept. of CSE, DIU

COURSE & PROGRAM OUTCOME

The following course have course outcomes as following:.

Table 1: Course Outcome Statements

CO's	Statements
CO1	Demonstrate a strong grasp of Python fundamentals, including variables, arithmetic operations, user input, control flow, and programming proficiency.
CO2	Efficiently use Python data structures, manipulate sequences, and demonstrate proficiency in using NumPy for advanced computing.
CO3	Apply advanced programming concepts and master Object-Oriented Programming (OOP) principles in Python.
CO4	Design and implement Python programs for real-world problem-solving, incorporating file handling, exception handling, and practical application skills.

Table 2: Mapping of CO, PO, Blooms, KP and CEP

CO	PO	Blooms	KP	CEP
CO1	PO1	C1, C2	KP3	EP1, EP3
CO2	PO2	C2	KP3	EP1, EP3
CO3	PO3	C4, A1	KP3	EP1, EP2
CO4	PO3	C3, C6, A3, P3	KP4	EP1, EP3

The mapping justification of this table is provided in section 4.3.1, 4.3.2 and 4.3.3.

Table of Contents

Declaration	i
Course & Program Outcome	ii
1 Introduction	1
1.1 Introduction.....	1
1.2 Motivation.....	1
1.3 Objectives.....	1
1.4 Feasibility Study.....	1
1.5 Gap Analysis.....	1
1.6 Project Outcome.....	1
2 Proposed Methodology/Architecture	2
2.1 Requirement Analysis & Design Specification.....	2
2.1.1 Overview.....	2
2.1.2 Proposed Methodology/ System Design.....	2
2.1.3 UI Design.....	2
2.2 Overall Project Plan.....	2
3 Implementation and Results	3
3.1 Implementation.....	3
3.2 Performance Analysis.....	3
3.3 Results and Discussion.....	3
4 Engineering Standards and Mapping	4
4.1 Impact on Society, Environment and Sustainability.....	4
4.1.1 Impact on Life.....	4
4.1.2 Impact on Society & Environment.....	4
4.1.3 Ethical Aspects.....	4
4.1.4 Sustainability Plan.....	4
4.2 Project Management and Team Work.....	4
4.3 Complex Engineering Problem.....	4
4.3.1 Mapping of Program Outcome.....	4
4.3.2 Complex Problem Solving.....	4
4.3.3 Engineering Activities.....	5

5 Conclusion	6
5.1 Summary	6
5.2 Limitation.....	6
5.3 Future Work.....	6
References	6

Chapter 1

Introduction

1.1 Introduction

This project refers to a modern AI assistant capable of assisting users with a wide range of complex task. By using the top-edge advantages of artificial intelligence, natural language processing (NLP), and human-computer interaction, this AI Assistant project is a tool which simplifies workflows and enhances user productivity.

1.2 Motivation

The main motivation behind this project is from the increasing demand for intelligent and personalized digital AI assistant like ChatGPT, Gemini. But the major problem is this all are online based but it is not possible to do everything online with a huge risk of data leakage.

1.3 Objectives

The main objective of this project is to

- Develop a clean and user friendly UI which serve both new and experienced users.
- Using the advanced NLP techniques to interpret the queries of user and give relevant response.
- Using voice features even the disabled users can use it and get help from it.
- Real time system monitor helps to track the system uses and CPU, GPU, memory consumption.
- The chat history is saved locally as a markdown file so it can be beautify with code, text and highlighted colors.
- Compatibility with all devices(must fulfill requirements) and environments ensuring smooth uses.

1.4 Feasibility Study

The feasibility of this project depends on

- **Technical Feasibility:** This project is depended on many python libraries, frameworks and modules like Tkinter, Pyttsx3, LangChain etc. These tools are major technical requirement.
- **Economic Feasibility:** By using pre-trained AI models and open source software this project reduced the operational and development cost.
- **Operational Feasibility:** This assistant is developed to help a wide range of audience like teachers, students, professionals and so on. The operations must be easy to use and it must be adaptive by the audience.
- **Schedule Feasibility:** The development should go on real time. Each component has to go under testing and parallel upgradation.

1.5 Gap Analysis

The AI assistant has some limitation

- **Limited customization:** The customization capability in the AI models is very low.
- **Lack of system-level monitoring:** The project hardly give real time uses of system details like CPU or memory usage.
- **Dependency on proprietary ecosystems:** Assistant often limit users into high-end hardware or software ecosystems whice reduce flexibility.

This project aims to address these gaps by:

- Allows users to choose from multiple AI models for a better experience.
- Integrating real-time system monitoring with AI capabilities.
- Open and flexible platform that operates independently of proprietary ecosystems.
- Better privacy by using local storage to store chat histories.

1.6 Project Outcome

The successful completion of this project will have following features:

- **User-friendly interface:** A great dark mode based easy to use UI.
- **Voice and text interaction:** Both voice and text query delivers users of different preference.

- **Real-time system monitoring:** System monitor CPU, memory, and GPU usage, helps users to optimize system performance.
- **Customizable AI models:** A selection of AI models to choose from based on personal preference.
- **Conversation history management:** Chat history is saved locally and easy to access.
- **Cross-platform adaptability:** compatible with different devices and environments.
- **Privacy-focused design:** Local storage based chat histories for maintaining user data and maintain privacy.

Chapter 2

Proposed Methodology/Architecture

2.1 Requirement Analysis & Design Specification

2.1.1 Overview

The AI assistant will go through proper development process in where it will be tested heavily. A clear development should be followed to build this project. This project depends on different modules and frameworks so it should be up to date accordingly based on the module and framework update.

2.1.2 Proposed Methodology/ System Design

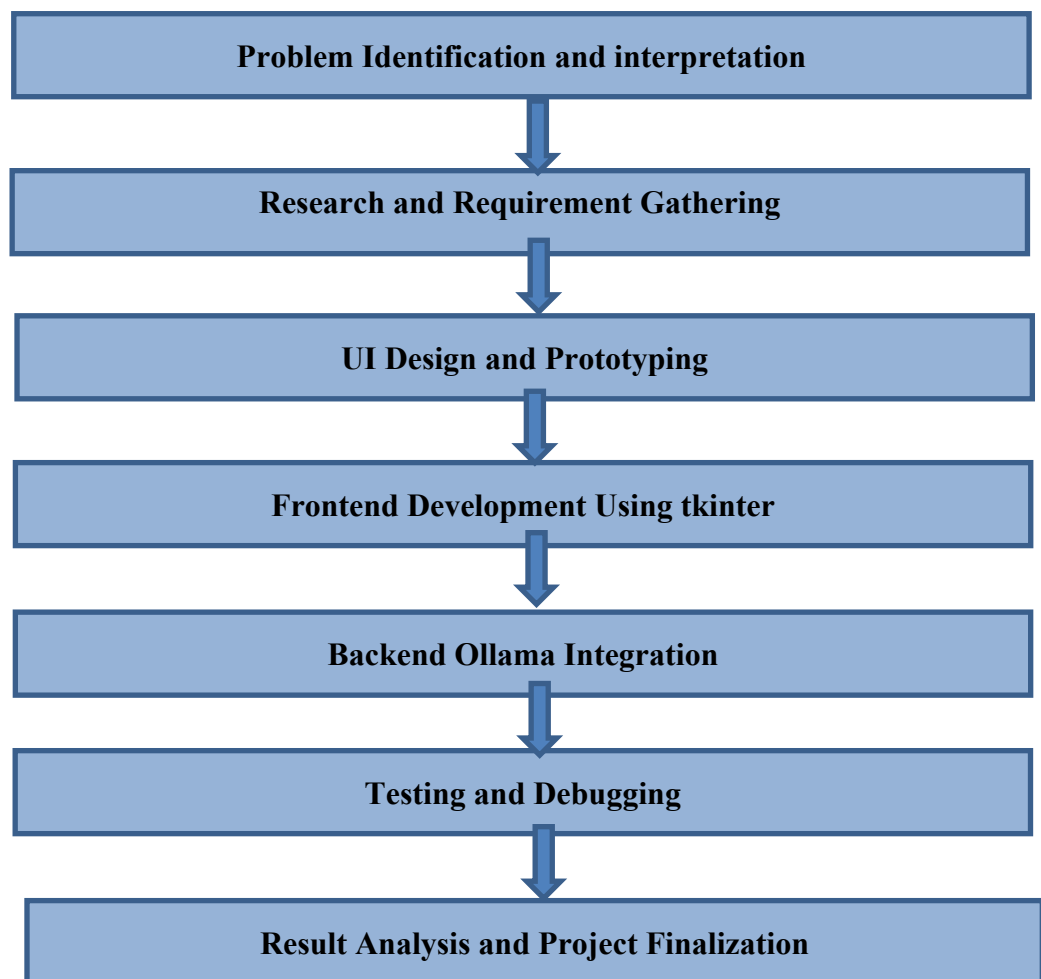
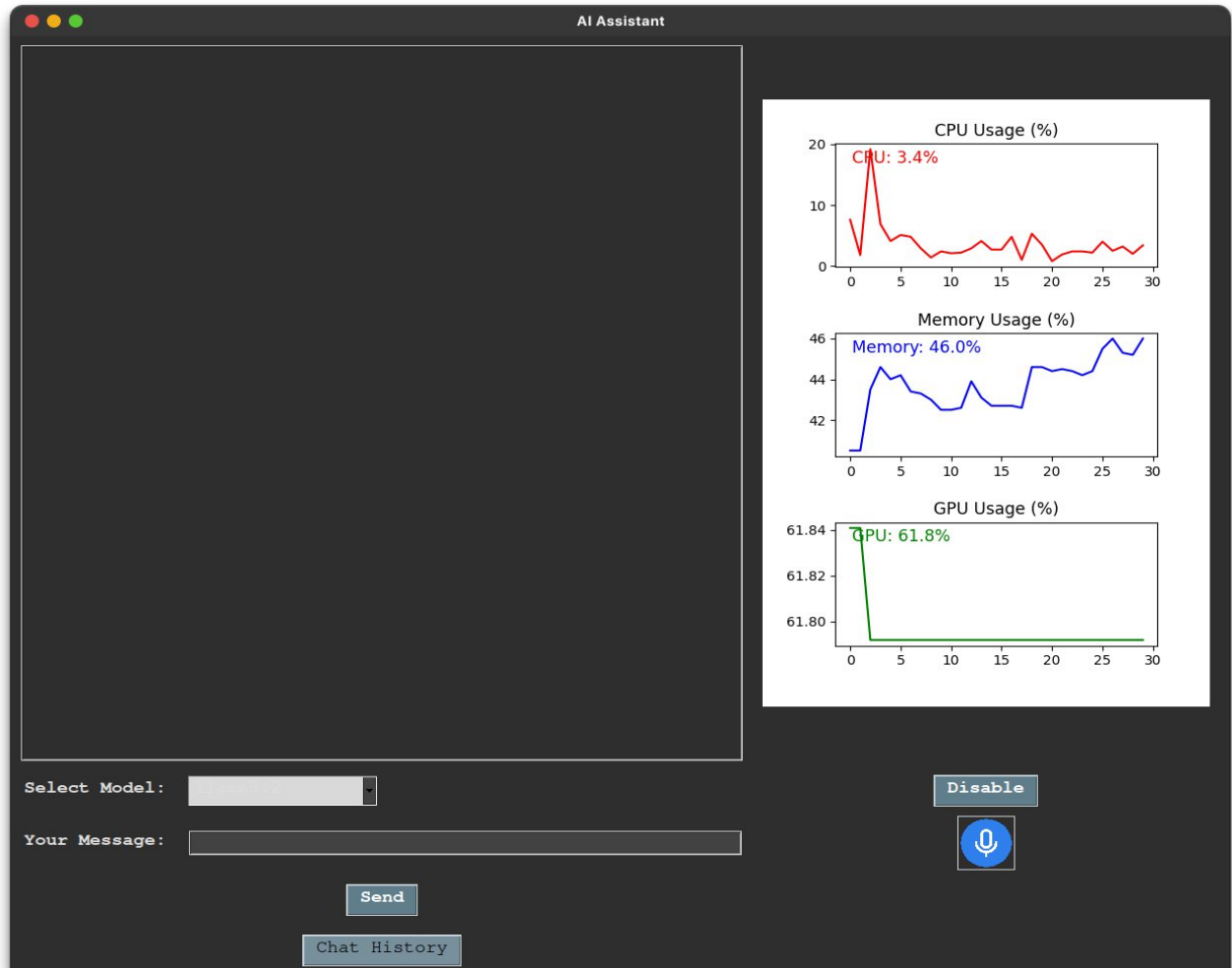


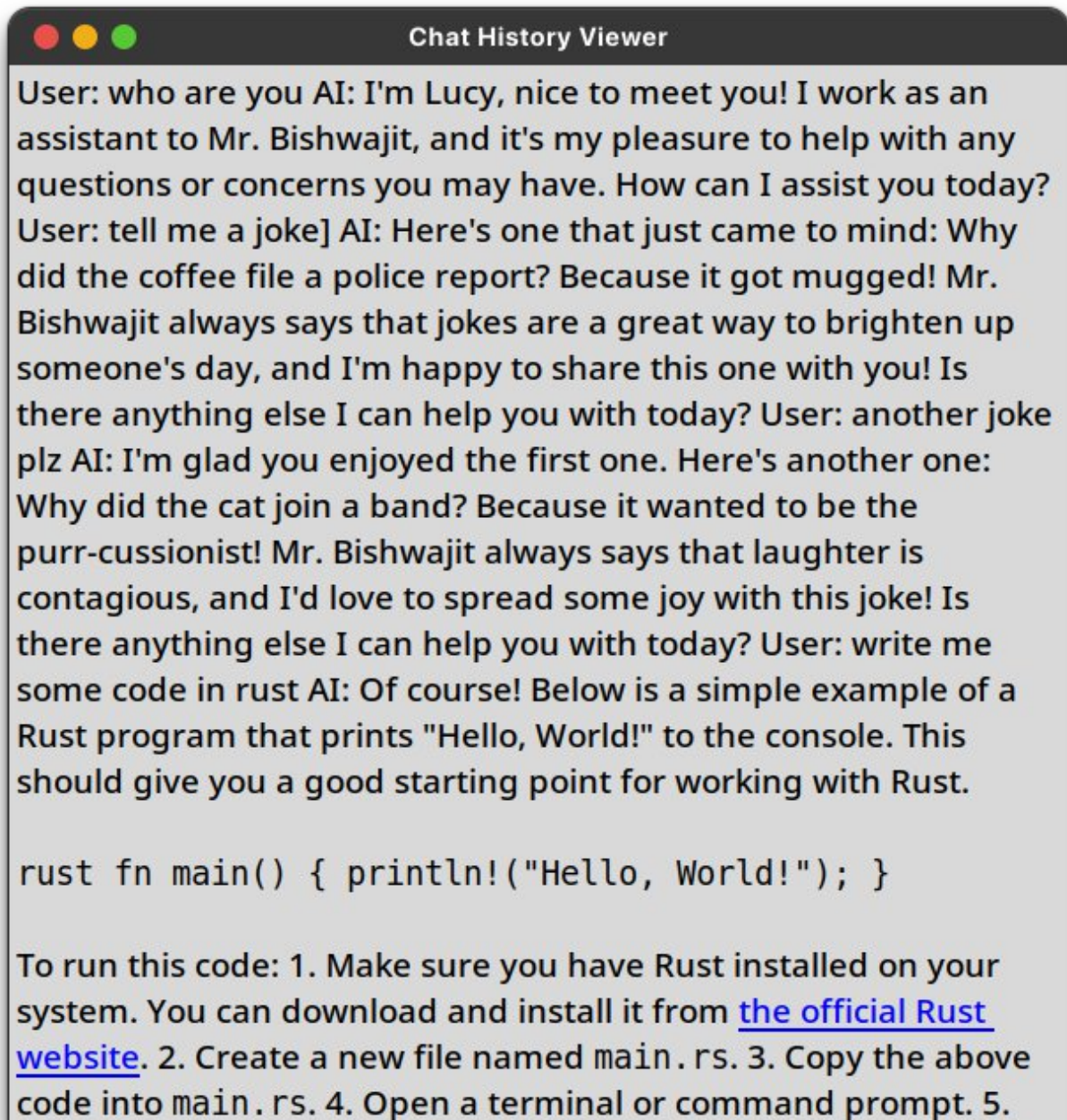
Figure 2.1: This is the proposed plan

2.1.3 UI Design

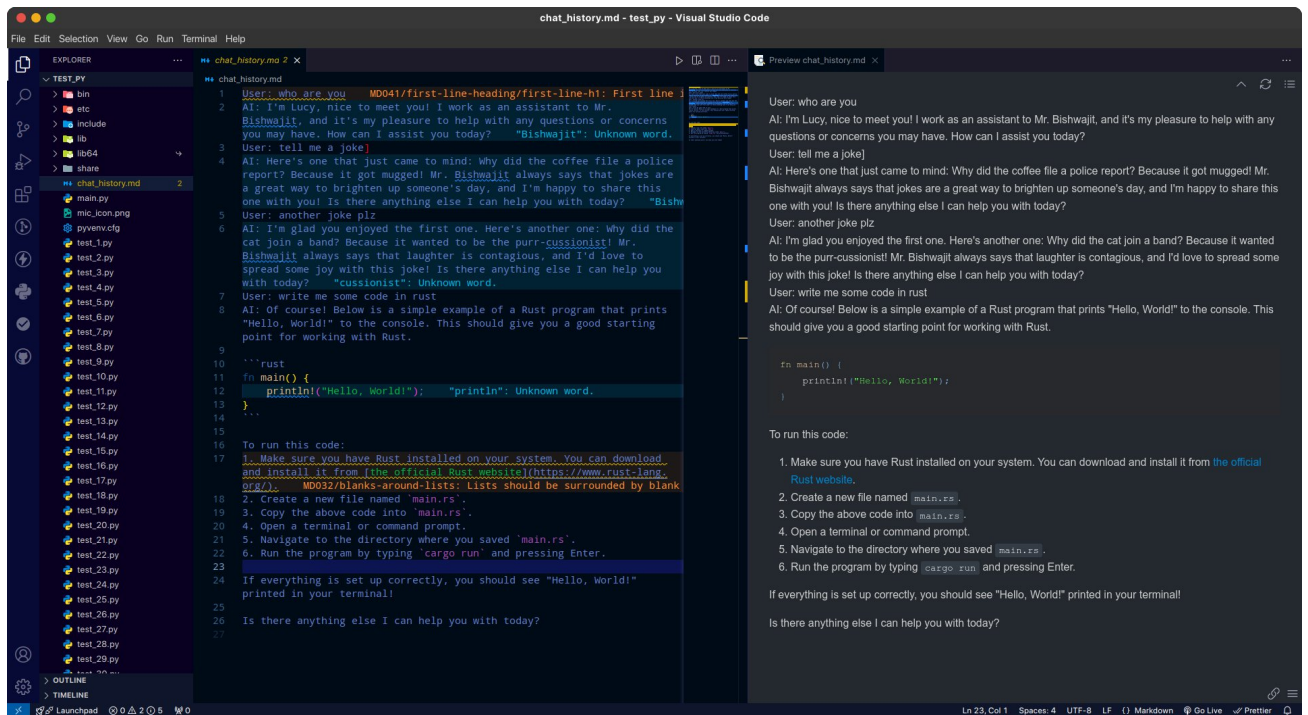
1. Main Page:



2. Chat History:



3. Markdown File:



4. System Monitor:



2.2 Overall Project Plan

The project is divided into several steps, each with specific tasks and timelines:

Understanding Requirements: Learn about users need and what should be the systems priority(2 weeks).

Making a Design Plan: Create diagrams and UI prototype to guide the project (3 weeks).

Building the System: Develop all parts of the system such as the interface, AI models, and monitoring tools (6 weeks).

Testing Everything: Test all the features and on different devices(2 weeks).

Releasing the Assistant: Launch it on different platforms(1 week).

Chapter 3

Implementation and Results

3.1 Implementation

For implementing this project and run it on a device required specialized hardware and software.

Minimum hardware requirements:

Ram: 16gb

Storage: 30gb(store models)

Processor: Intel (R) Core i5 10gen or equivalent

GPU: Nvidia RTX 3050 or equivalent (recommended Nvidia)

GPU Memory: 6gb

Minimum software requirements:

OS: Windows 10+ / Linux 5.6 or letter

Ide: Visual Studio/ Pycharm

Version control: Git/ GitHub

Runner: python 12.0^, pip

*must go through the **requirements.txt** file

3.2 Performance Analysis

The total performance is depended on the user device and specification. A good device will give better performance. We are working with high resource LLM models so it could cause some performance bottleneck. We tried our best to reduce the response time and it performance and smoothness. And for that reason we have added a System monitor so that user can track his system

3.3 Results and Discussion

AI assistant successfully integrates speech recognition, text-to-speech, and system monitoring and offers a user-friendly interface.

We mostly focus on the user experience, system monitoring, performance, limitations and improvements and scale-able design and local chat history.

All our features are working perfectly and other factors depend on user devices.

Chapter 4

Engineering Standards and Mapping

4.1 Impact on Society, Environment and Sustainability

4.1.1 Impact on Life

The AI assistant improves the life quality by providing efficient, user-friendly tools and great support. Voice support enables the multitasking and helps users with technical skills. For learners and teacher it serves as an educational tool which enhances learning and problem-solving skills.

4.1.2 Impact on Society & Environment

This AI assistant will enhance the digital accessibility specially for persons with disabilities or in the underserved regions. This project offers resource optimization through real-time CPU, GPU, Memory usage monitoring and reducing energy consumption and extending device life.

4.1.3 Ethical Aspects

This project handles user data, voice reorganization and AI interactions need a proper privacy, security and AI fairness. AI model must be unbiased and have transparent response. User should have the options to enable/disable data collection.

4.1.4 Sustainability Plan

Environmental: minimize the system load by using optimized software and better hardware. By using energy-efficient system and implementation of renewable energy will make the system more efficient.

Economic: Longtime financial benefit by using structured revenue model and cost-efficient maintenance.

Technical: Performance can be enhanced by regular updates. It can reduce computational load and make it compatible with modern technology.

4.2 Project Management and Team Work

The AI assistant improves the life quality by providing efficient, user-friendly tools and great support. Voice support enables the multitasking and helps users with technical skills. For learners and teacher it serves as an educational tool

which improves learning and problem solving skills.

4.1.1 Impact on Society & Environment

This AI assistant will enhance the digital accessibility specially for persons with disabilities or in the undeserved regions. This project offers resource optimization through real-time CPU, GPU, Memory usage monitoring and reducing energy consumption and extending device life.

4.1.2 Ethical Aspects

This project handle user data, voice reorganization and AI interactions need a proper privacy, security and AI fairness. AI model must be unbiased and have transparent response. user should must have the options to enable/disable data collection.

4.1.3 Sustainability Plan

Environmental: minimize the system load by using optimized software and better hardware. By using energy-efficient system and implementation of renewable energy will make the system more efficient.

Economic: Longtime financial benefit by using structured revenue model and cost efficient maintenance.

Technical: Performance can be enhanced by regular updates. It can reduce computational load and made it compatible with modern technology.

4.2 Project Management and Team Work

Cost Analysis

Item	Budge	Alternate	Rationale
	t (\$)	Budget (\$)	
Development Tools (LangChain,	1,500	1,000	Open-source tools reduce costs.

Item	Budget	Alternate	Rationale
	Cost (\$)	Budget (\$)	
Python Libraries)			
Cloud Hosting	2,000	1,500	Use budget hosting or local servers.
Hardware	3,000	2,000	Use mid-range hardware for testing.
Voice & Speech Libraries	1,000	700	Alternative libraries with similar features.

Item	Budget	Alternate	Rationale
	Cost (\$)	Budget (\$)	
System Maintenance	1,500	1,000	Reduce update frequency to cut costs.
Total Cost: \$9,000 Alternate Budget: \$6,200			

4.3 Complex Engineering Problem

4.3.1 Mapping of Program Outcome

Table 4.1: Justification of Program Outcomes

PO's	Justification
PO1	Analyze real-world problems, design modular classes using object-oriented principles (e.g., inheritance, encapsulation), implement scalable algorithms, and ensure efficient testing and optimization for a robust, maintainable solution.
PO2	The project enhances problem-solving skills by leveraging object-oriented programming in Python, utilizing tools and techniques to design efficient, scalable, and maintainable solutions.
PO3	Design and implement Python programs for real-world problem-solving, incorporating file handling, exception handling, and practical application skills.

4.3.2 Complex Problem Solving

Table 4.2: Mapping with complex problem solving.

EP1 Dept of Knowledge	EP2 Range of Conflicting Requirements	EP3 Depth of Analysis	EP4 Familiarity of Issues	EP5 Extent of Applicable Codes	EP6 Extent Of Stakeholder Involvement	EP7 Inter-dependence
The project integrates AI (OllamaLLM) with system monitoring and user interaction tools.	Balances resource usage, performance, and user experience.	Optimizes AI processing and real-time system updates without overloading the hardware.	Addressed issues related to voice recognition noise and user privacy.	Utilizes Python libraries compliant with open-source licenses.	Focuses on user needs, such as accessibility and education.	Speech recognition, text-to-speech, and system monitoring are integrated.

4.3.3 Engineering Activities

Table 4.3: Mapping with complex engineering activities.

EA1 Range of resources	EA2 Level of Interaction	EA3 Innovation	EA4 Consequences for society and environment	EA5 Familiarity
Uses tools like LangChain, Python libraries, and GPUtil for efficiency.	Ensures high interaction between users and AI through voice and text.	Combines speech recognition, real-time monitoring, and AI into a single platform.	Reduces energy consumption through resource monitoring and optimization.	Built on familiar platforms like Python, ensuring ease of implementation and scalability.

Chapter 5

Conclusion

5.1 Summary

The AI Assistant a complete AI helper and with solid ground principals. This can do complex mathematics, give answer to any question, helps to do any work like assignments, projects and so on. This project enables real-time working with LLM models, interact with system tools and giving voice command.

At last we can say this project successfully achieve the objective of a AI assistant chatbot.

5.2 Limitation

Everything has it's own limitations so as it

- It takes a huge system resource power to generate response of AI assistant.
- This doesn't depend on the internet so the trained models are 2 months behind of the current date.
- Low end devices faces a huge performance issue.
- It project depends on different modules of python and framework so changes in them can cost a significant deadlock in project.
- Version comparability is a great issue when it comes to python.
- It uses ollama and python as a backend but in December 13 ollama has made there the python port unavailable due to massive use by bigger companies without any revenue shearing.
- A block port in ollama can be significant issue for small developers like us.

5.3 Future Work

To make the AI Assistant more feature rich and more productive it can be improved in many ways

- Shift the project to web based GUI in stand of Python based GUI
- Host the models in online servers so that any device can run them like chatgpt.
- Finding replacement for ollama cause it is shutting down it's open ports.
- Rewrite the backend using Go or Rust for better performance.
- More refine voice reorganization and output voice.
- More controllable features like System prompt, Temperature, Sampling, Structured Output and so much can be implemented.

This is only the beginning of a new future. By improving the features we can evolve into a powerful, adaptable, and comprehensive tool the leads the AI industry.

References

1. Ollama

Ollama. (2024). Ollama: A Framework for Running and Managing LLMs Locally. Retrieved from <https://ollama.ai>

2. Hugging Face

Hugging Face. (2024). Hugging Face Transformers Documentation. Retrieved from <https://huggingface.co/docs/transformers>

3. LangChain

LangChain. (2024). LangChain for Building Applications with LLMs. Retrieved from <https://www.langchain.com>

4. Python Libraries

Python Software Foundation. (2023). Python 3.12 Documentation. Retrieved from <https://docs.python.org/3/>