Technical Report

Group – 2

University of Missouri – Kansas City

COMP\_SCI5588: Data Science Capstone

Yugyung Lee

March 18, 2024

# Technical Report

**1. Introduction**

This technical report outlines the comprehensive implementation of various systems aimed at enhancing user interaction and data management within a student oriented personal assistant – UniBuddy. The implemented systems include User Authentication and Registration, Integration of Firebase with Streamlit for Chat Application and Data Export, User Authentication and Chat Application Development, and Natural Language Processing capabilities. Each system is designed to serve specific functions and contributes to the overall functionality and user experience of the application.

**2. Project Goals and Objectives**

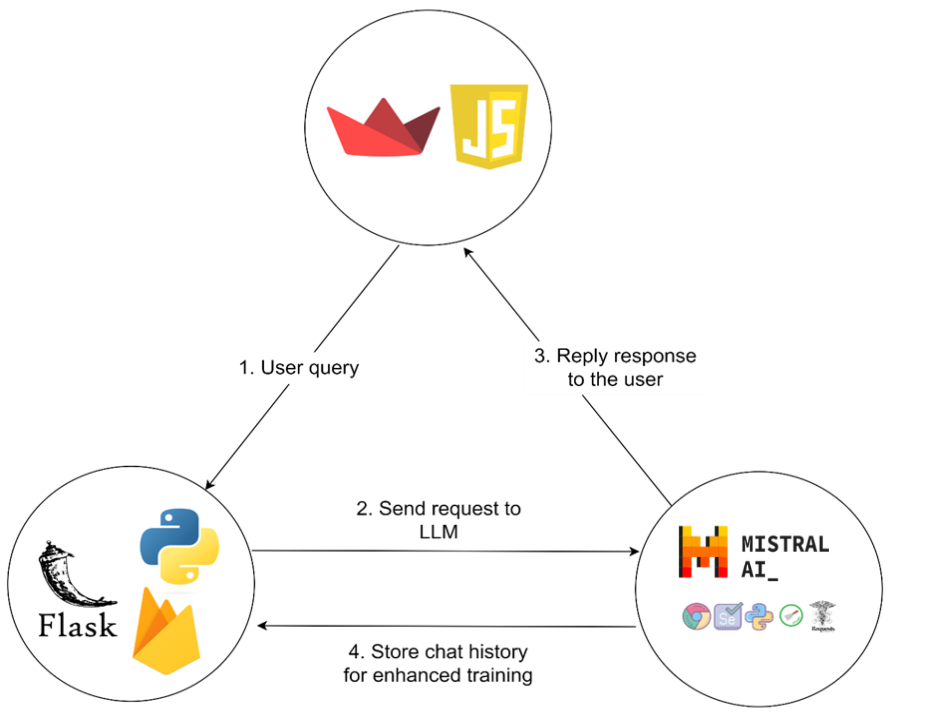
The goal of our virtual assistant is to provide students with a comprehensive and seamless platform for accessing all relevant information about universities. Our primary objective is to create an intuitive chatbot interface that enables students to easily inquire about various aspects of university life, including academics, sports, financial aid, and more. By leveraging natural language processing and advanced data retrieval techniques, our virtual assistant aims to deliver accurate and up-to-date information in real-time, helping students make informed decisions about their educational journey. Additionally, we seek to enhance user engagement and satisfaction by offering personalized recommendations and tailored responses based on individual preferences and requirements. Ultimately, our goal is to empower students with the resources and support they need to navigate the university experience confidently and successfully.

**3. Work Breakdown Structure**

The implementation of the project involves the following key tasks:

* Sai Karthik Naladala
  + Designing and implementing user authentication and registration functionalities using Flask and Firebase.
* Tarun Siga
  + Integrating Firebase with Streamlit to develop a chat application with real-time communication features.
* Deepak Ayyasamy
  + Fine-tuning an LLM for university-related queries using Llama2, Langchain, Faiss, and Firebase.

**4. Architecture**



The architecture of the implemented systems is as follows:

**By Phase – 2:**

* User Authentication and Registration System
  + Utilizes Flask for application routes and session management, with Firebase handling authentication and real-time database functionalities.
* Integration of Firebase with Streamlit
  + Employs Streamlit for the chat application interface and Firestore backend for chat history storage.
  + A separate feature for exporting chat data to CSV format is implemented using Firebase Admin SDK.
* User Authentication and Chat Application Development
  + Flask manages user registration, login, and session management, while Streamlit provides the frontend for the chat application.
  + Firebase Authentication ensures secure user authentication, and Firestore securely stores chat messages.
* Natural Language Processing
  + Llama2 serves as the core NLP engine, while Langchain enhances its capabilities by integrating external data sources and APIs.
  + Website data is scrawled using Langchain and working on converting the raw data into trainable dataset with help of GPT.
  + Working with Google Colab on fine-tuning the LLM model and testing the resultant model.
  + Faiss acts as the vector database for efficient storage and retrieval of scraped data. Firebase is used for additional functionalities and data management.

**By phase 3 – 2:**

* Enhanced Login/Register Page Design:
  + The login/register page has been revamped to offer a more professional and user-friendly interface.
  + Additional fields such as name, university name, email, etc., have been incorporated to gather comprehensive user details.
  + The design has been tailored for a custom experience, ensuring ease of navigation and accessibility for users of all backgrounds.
* Improved User Authentication and Registration Process:
* User details provided during registration are now securely stored in a real-time Firebase database, ensuring efficient data management and retrieval.
* Upon successful login, users are seamlessly redirected to the UniBuddy homepage, where personalized features and functionalities are accessible.
* Dynamic Database Handling:
* Each user's chat history is now automatically saved with their respective email ID as the database name, eliminating the need for users to specify database names manually.
* This dynamic handling of databases enhances user convenience and streamlines data organization within the application.
* Integration of Mistral AI Model:
  + Following the successful training of the Mistral AI model, it has been seamlessly integrated into the UniBuddy homepage.
  + User prompts and model responses are now captured and stored in their respective databases, facilitating ongoing training and refinement of the AI model.
  + The integration of Mistral AI enhances the application's natural language processing capabilities, enabling more accurate and contextually relevant interactions with users.

**5. Challenges**

Throughout the implementation process, several challenges were encountered, including:

* Integration complexity: Integrating multiple technologies and frameworks required careful planning and coordination to ensure seamless functionality.
* Authentication and security: Implementing secure authentication mechanisms and data handling processes posed challenges in ensuring data privacy and user security.
* Fine-tuning of the LLM: Training and fine-tuning the LLM to effectively answer user queries required significant computational resources and expertise in natural language processing techniques.

**6. Future Work**

* Enhancing user authentication and authorization functionalities to support role-based access control.
* Implementing additional features and functionalities in the chat application, such as multimedia support and sentiment analysis.
* Expanding the capabilities of the natural language processing system to handle a wider range of user queries and provide more accurate and contextually relevant responses.