**PDF Document Outline**

**1. Overall Approach**

**Project Overview:** This project is a chatbot application designed to provide answers to predefined questions from a corpus. The chatbot maintains conversation history to enhance user interaction. It is implemented using Flask for the backend and HTML/JavaScript for the frontend.

**Approach:**

* **Backend**: Developed using Flask to handle API requests and serve responses based on a predefined corpus.
* **Frontend**: Built with HTML and JavaScript to provide a minimalistic and interactive user interface. Conversation history is managed client-side and sent to the server for processing.

**2. Frameworks/Libraries/Tools Used**

**Backend:**

* **Flask**: A micro web framework for Python used to create the web server and handle HTTP requests.
* **NLTK**: The Natural Language Toolkit used for tokenizing user input to match questions in the corpus.

**Frontend:**

* **HTML**: Used to structure the chatbot interface.
* **JavaScript**: Used to handle user interactions, manage conversation history, and communicate with the Flask backend.

**Tools:**

* **Python**: Programming language used for the backend implementation.
* **pip**: Package installer for Python used to install Flask and NLTK.

**3. Problems Faced and Solutions**

**Problem 1: Handling Missing Keys**

* **Issue**: KeyError occurred when the conversation key was missing in the request data.
* **Solution**: Updated the backend code to use data.get('key', default\_value) to provide default values when keys are missing.

**Problem 2: Conversation History Management**

* **Issue**: Ensuring that the conversation history was maintained correctly between the client and server.
* **Solution**: Implemented client-side management of conversation history and ensured that the entire conversation history was sent with each request to update and display it correctly.

**Problem 3: NLTK Data Download Issues**

* **Issue**: Difficulty in downloading NLTK data required for tokenization.
* **Solution**: Provided a separate script (download\_nltk\_data.py) to download the necessary NLTK data before running the main application.