VISVESVARAYA TECHNOLOGICAL UNIVERSITY

JNANA SANGAMA, BELAGAVI-590018

MINI PROJECT SOURCE CODE
(21ADMP67)
ON

"AI-CHATBOT FOR DEPARTMENT"

Submitted in partial fulfilment of the requirements for the award of the degree of

BACHELOR OF ENGINEERING
IN
ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

Submitted by

HARSHITHA N	**(1KG21AD012)**
MANYASHREE U J	**(1KG21AD028)**
PRAJWAL C L	**(1KG21AD036)**
JYOTHISH K S	**(1KG22AD401)**

Under the Guidance of

Prof. Harshitha H S
Assistant Professor
Department of AI&DS
K.S.S.E.M, Bengaluru

Department of Artificial Intelligence and Data Science

K. S. School of Engineering and Management
No 15, Mallasandra, off. Kanakapura Road, Bengaluru – 560109
2023-2024

PROJECT MANUAL

1. Introduction

This manual provides instructions to set up and run the AI Chatbot project, with pseudocode for understanding the flow.

2. Prerequisites

- Python 3.6 or higher
- Flask
- Flask-CORS
- Google Generative AI Python package
- A web browser

3. Project Structure

```
AI_Chatbot_Project/
│
├── app/
│   ├── __init__.py
│   ├── routes.py
│
├── templates/
│   ├── index.html
│
├── static/
│   ├── css/
│   │   └── style.css
│   ├── js/
│       └── script.js
│
├── main.py
├── requirements.txt
└── README.md
```

4. Setting Up the Project

4.1 Clone the Repository

```
git clone <repository-url>
cd AI_Chatbot_Project
```

4.2 Create a Virtual Environment

```
python -m venv venv
source venv/bin/activate   # On Windows: venv\Scripts\activate
```

4.3 Install Dependencies

```
pip install -r requirements.txt
```

requirements.txt should contain:

```
Flask
Flask-CORS
google-generativeai
```

5. Configuring the Google API Key

Update routes.py with your Google API key:

- GOOGLE_API_KEY = "YOUR_GOOGLE_API_KEY"

6. Running the Project

Pseudocode

Backend (Flask Application)

plaintext

Copy code

- Import necessary libraries
- Initialize Flask app and CORS
- Set Google API key and configure generative AI model

- Define function to interact with the AI model:
    - Prepare a knowledge base and user query
    - Generate content using the AI model
    - Check for specific responses and return appropriate reply

- Define route for chat:
    - Get user input from request
    - Use the AI model to generate a response
    - Return response as JSON

- Run Flask app

Frontend (HTML, CSS, JavaScript)

- **HTML**
- Create basic HTML structure
- Add a container for chat messages
- Add an input field for user messages
- Include CSS and JavaScript files

CSS

- Define styles for body and chat container
- Style chat messages, user messages, and bot messages
- Style chat box and input field

JavaScript

- Define function to append messages to chat box
- Define function to send user messages to the server
    - Get user input and display it
    - Send input to the server via POST request
    - Display the server's response

This pseudocode provides a high-level overview of how the backend and frontend components work together in this project. Let me know if you need any additional details or assistance.

7. Accessing the Application

Open your web browser and go to http://127.0.0.1:5000 to access the chatbot interface.

8. Troubleshooting

- Ensure all dependencies are installed.
- Verify the Google API key is correctly set.
- Check Flask server logs for any errors.

9. Conclusion

By following this manual, you should be able to successfully set up and run the AI Chatbot project. For further assistance, consult the project documentation or contact the project maintainers.