Language Learning Chatbot Documentation

- Abhipsha Bhatta

Overview

The Language Learning Chatbot is a web app that helps users learn new languages by chatting with an AI assistant. The chatbot answers questions about languages, provides translations, explains grammar, and helps with pronunciation.
The app has two main parts:

- Frontend: Built with React, where users can sign up, log in, and chat with the AI.
- Backend: Built with Flask, which manages users, handles chats, and connects to databases.

Features

- User Registration and Login: Securely sign up and log in.
- Language Learning Help: The AI chatbot helps users learn languages.
- Save Chat History: Store chat messages to track learning (planned for the future).

Technologies Used

- Frontend: React, JavaScript
- Backend: Flask (Python), Flask-JWT-Extended, Flask-Bcrypt, Flask-CORS
- Databases: PostgreSQL (user information) and MongoDB (chat messages)
- API: OpenAI API for AI responses

How to Start

Requirements:

- Python 3.8+ for backend
- Node.js and npm for frontend
- PostgreSQL and MongoDB for databases
- OpenAI API Key

Setup Instructions

1. Clone the Project:
   git clone https://github.com/abhipshabhatta/language-learning-chatbot.git
   cd language-learning-chatbot

Backend Setup:

- Go to the backend folder: cd backend
- Create a virtual environment and install dependencies: python3 -m venv venv
- source venv/bin/activate
- pip install -r requirements.txt

Frontend Setup:

- Go to the frontend folder: cd frontend
- Install frontend dependencies: npm install

Running the App

1. Backend:
   Create a .env file with environment variables
   JWT_SECRET_KEY= “your jwt secret key”
   DATABASE_URL=postgresql://username:password@localhost:5432/your_database
   MONGO_URI=mongodb://localhost:27017
   OPENAI_API_KEY=your_openai_key

   - Start the backend: python app.py
2. Frontend:
   - Start the frontend: npm start

Backend Overview

Database Setup

- PostgreSQL: Stores user accounts and passwords.
- MongoDB: Stores chat messages.

User Authentication

- Users sign up and log in using endpoints: POST /auth/signup and POST /auth/signin
- Passwords are securely stored using bcrypt.

Chatbot Functionality

- Users can chat with the AI using the POST /chat/ask endpoint.
- The backend uses OpenAI's API to generate the chatbot’s responses.

Frontend Overview

- Signup Page (Signup.jsx): Users register by providing their details.
- Login Page (SignIn.jsx): Users log in to access the chat.
- Chat Page (Chat.jsx): Users chat with the AI, ask questions, and learn languages.

API Details

Authentication API

- POST /auth/signup: Allows new users to create an account.
  Request data: { "username": "user123", "password": "password123"}

Response:

- 201 Created: User created.
- 400 Bad Request: User already exists.

POST /auth/signin: Logs users in and provides a JWT token.

Response:

- 200 OK: Token provided.
- 401 Unauthorized: Incorrect credentials.

Chatbot API

- POST /chat/ask: Sends user’s question to the AI and gets a response.

Request: {"question": "How do I say 'hello' in French?"}

Response: {"answer": "In French, you say 'bonjour'."}

How to Use the Chatbot

3. Sign Up: Register with your details on the signup page.
4. Log In: Use your credentials to log in.
5. Chat: On the chat page, select the language you want to learn and start asking questions.

Environment Settings

In your .env file, set:

- JWT_SECRET_KEY: Key to create secure user tokens.
- DATABASE_URL: Connection for PostgreSQL.
- MONGO_URI: Connection for MongoDB.
- OPENAI_API_KEY: OpenAI key to access AI services.

Testing

- Manual Testing: Use Postman to test each API endpoint (signup, login, chat).
- Unit Testing: Write tests to check each function works correctly, especially for user signup/login and chatbot interactions.

Future Improvements

Chat History:

- Add the ability for users to view their past conversations for learning reference.

Quizzes and Practice:

- Create quizzes to test users' knowledge of the language they are learning.

Speech Features:

- Add speech-to-text for input and text-to-speech for output to improve interactivity.

Conclusion

The Language Learning Chatbot is built to help people learn new languages in an interactive way. It uses a combination of secure user login, real-time AI chat, and language-specific help to create a fun and educational experience.