Chatbot ANVIAN

Introduction

This project is an AI-powered chatbot designed to generate and provide code snippets in multiple programming languages. The chatbot interacts with users, understands their code requests, and delivers relevant code using LLM-powered automation. The frontend is built with React (Vite), and the backend is implemented using FastAPI.

Features

- **Multi-language support**: Generates code in Python, JavaScript, Java, C++, and TypeScript.
- **Interactive UI**: User-friendly chat interface.
- **FastAPI Backend**: API-based architecture for handling requests.
- **Autogen Integration**: Uses LLMs for intelligent code generation.
- **Cloud Deployment**: Hosted on a cloud platform.
- **Secure API Handling**: Environment variables for API key management.

System Architecture

The chatbot consists of two main components:

1. **Frontend**: Built using React and Vite.
2. **Backend**: Developed with FastAPI and Autogen.

Technologies Used

- **Frontend**: React, Vite, Framer Motion, Lucide Icons.

- **Backend**: FastAPI, Autogen, Python, Uvicorn.
- **Database**: Not applicable.
- **Hosting**: Cloud deployment (Render, Vercel, or AWS).

Installation and Setup

Prerequisites

- **Git** installed and configured.
- **Node.js** and **npm** installed.
- **Python 3.9+** installed.
- **Virtual Environment** (Optional but recommended).

Clone the Repository

```
git clone https://github.com/Chowdary24/chatbot_ANVIAN.git
cd chatbot_ANVIAN
```

Backend Setup

1. Create a virtual environment and activate it:

```
python -m venv venv
source venv/bin/activate  # For macOS/Linux
venv\Scripts\activate  # For Windows
```

2. Set up environment variables: Create a `.env` file in the root directory and add:

```
GROQ_API_KEY=your_api_key_here
```

3. Run the FastAPI server:

uvicorn main:app --host 0.0.0.0 --port 8000 --reload

Frontend Setup

1. Navigate to the frontend directory:

cd frontend

2. Install dependencies:

npm install

3. Start the development server:

npm run dev

Cloud Deployment

For cloud deployment, follow these steps:

Backend Deployment

1. **Render Deployment** (Recommended for FastAPI):
   - Create a **Render Web Service**.

Set the **Start Command** as:

 uvicorn main:app --host 0.0.0.0 --port 8000

   -
   - Add the **Environment Variable**: `GROQ_API_KEY`.

Frontend Deployment

1. **Vercel Deployment**:

Install Vercel CLI:

```
npm install -g vercel
```

- ○

Deploy:

```
vercel
```

- ○

API Endpoints

`POST /autogen-chat`

Request Body:

```
{
  "message": "Generate Python code for a calculator",
  "language": "Python"
}
```

Response:

```
{
  "response": "def calculator():\n  # Code here"
}
```

Testing

Backend Tests

Run unit tests with:

```
pytest
```

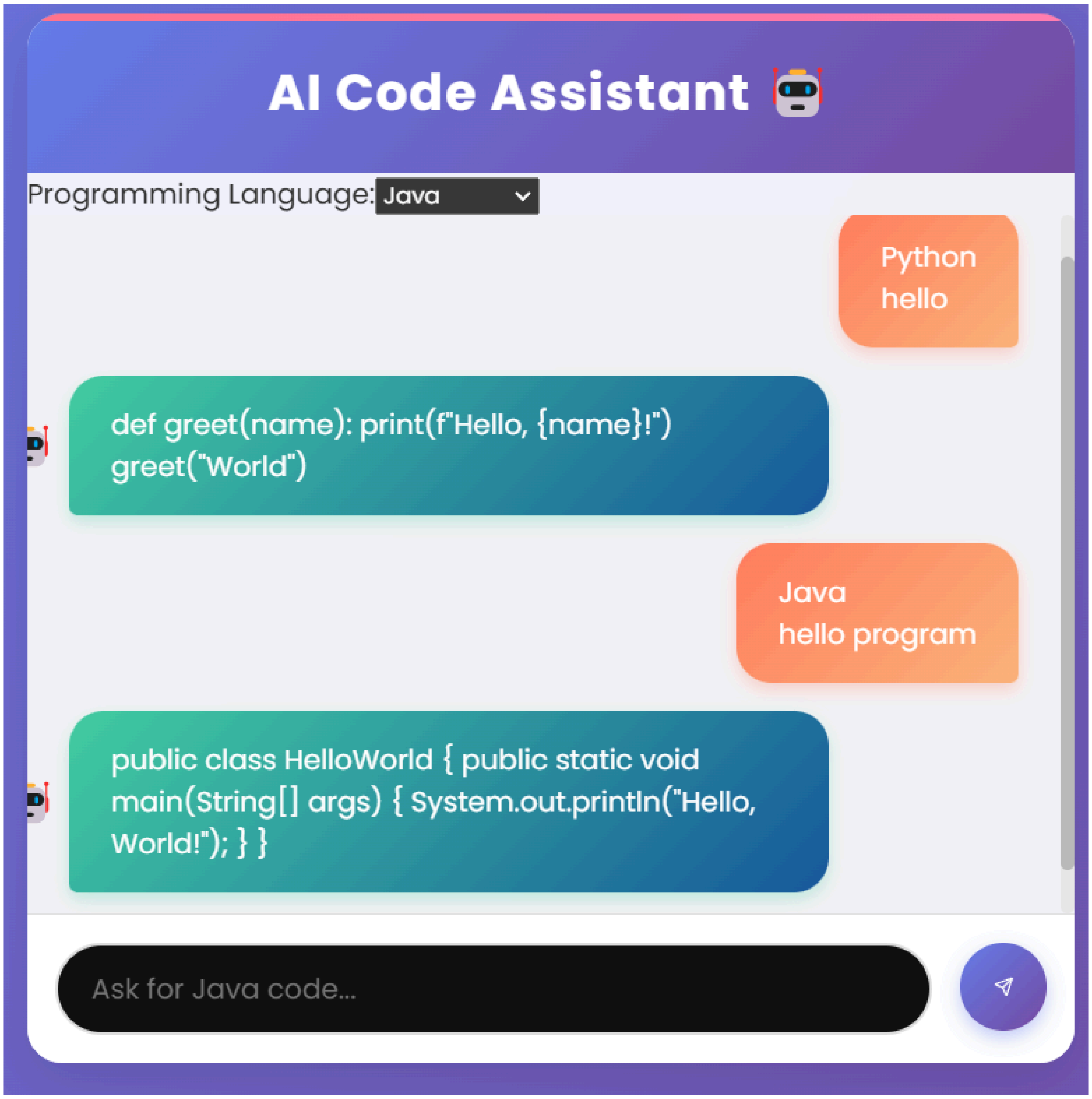
Frontend Tests

Run Jest tests:

npm test

Screenshots:

FrontEnd:



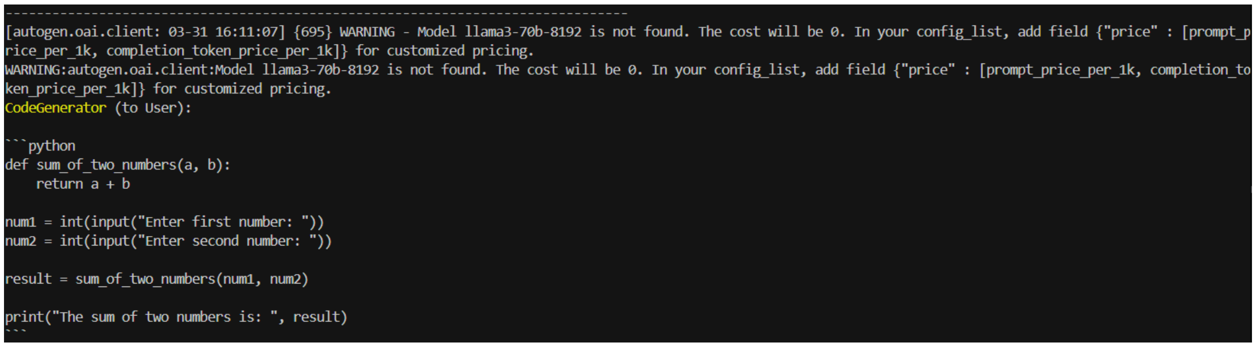
Python
hello

def greet(name): print(f"Hello, {name}!")
greet("World")

Java
hello program

public class HelloWorld { public static void main(String[] args) { System.out.println("Hello, World!"); } }

Query Response:



~~~
--------------------------------------------------------------------------------
[autogen.oai.client: 03-31 16:11:07] {695} WARNING - Model llama3-70b-8192 is not found. The cost will be 0. In your config_list, add field {"price" : [prompt_p
rice_per_1k, completion_token_price_per_1k]} for customized pricing.
WARNING:autogen.oai.client:Model llama3-70b-8192 is not found. The cost will be 0. In your config_list, add field {"price" : [prompt_price_per_1k, completion_to
ken_price_per_1k]} for customized pricing.
CodeGenerator (to User):

```python
def sum_of_two_numbers(a, b):
    return a + b

num1 = int(input("Enter first number: "))
num2 = int(input("Enter second number: "))

result = sum_of_two_numbers(num1, num2)

print("The sum of two numbers is: ", result)
```
~~~

Future Enhancements

- Add more programming languages.
- Improve UI with more interactive features.
- Implement user authentication for personalized responses.

Contributors

- **V.Venkatesh Chowdary** - Developer

  ph.no: 7337261827

  Gmail:venkateshvakalapudi24@gmail.com

  Git Repo:https://github.com/Chowdary24/InternShip24.git