SRS DOCUMENTATION FOR

PROGRAMMING ASSISTANCE AI BOT

Project Objective

The objective of this project is to develop a locally-hosted Programming Assistance AI Bot using the LLaMA model to aid students in learning programming concepts, resolving coding issues, and improving their problem-solving skills. This tool aims to provide instant assistance and a hands-on learning experience in Python, Java, and C++.

Constraints

- **Local Hosting:** The application must run locally, which could limit accessibility and scalability.
- **Hardware Requirements:** Requires sufficient computational resources (16 GB RAM minimum) to handle the LLaMA model.
- **Language Support:** Currently supports Python, Java, and C++ only.
- **Response Time:** Depends on the local system's performance.
- **Model Limitations:** The accuracy and relevance of responses are constrained by the LLaMA model's capabilities.

Assumptions

- Users have basic knowledge of using a web browser to interact with the bot.
- The LLaMA model is installed and configured correctly on the local machine.
- The user inputs valid and interpretable queries or code snippets.
- Python, Flask, and all dependencies are pre-installed and functional.
- The user operates the tool on a machine meeting the minimum hardware requirements.

Dependencies

- **Python 3.8+:** Programming language used for backend development.
- **Flask Framework:** Handles the web server and API functionalities.
- **LLaMA Model:** Provides the AI-powered responses.
- **HTML, CSS**: For creating the user interface.
- **Ollama**: Facilitates local hosting of the LLaMA model.

Primary Functions

1. **Answering Programming Questions**:
   Provides explanations, examples, and insights for Python, Java, and C++ programming queries.

2. **Algorithm Explanations**:
   Explains algorithms step-by-step with code examples in Python.
3. **Debugging Code**:
   Identifies and resolves issues in the provided code snippets.
4. **Code Optimization**:
   Suggests improvements to enhance the efficiency of the given code.
5. **Practice Problems**:
   Recommends relevant practice exercises with hints and solutions.

Features

1. **Interactive Web Interface**:
   Users can easily interact with the bot via a web browser.
2. **Error Handling**:
   Provides descriptive error messages for invalid inputs or system issues.
3. **Code Examples**:
   Responses include code snippets for better clarity and understanding.
4. **Practice Problem Suggestions**:
   Curates programming exercises with difficulty levels and hints.
5. **Multi-Language Support**:
   Handles Python, Java, and C++ queries.
6. **Local Hosting**:
   Ensures complete control over data and privacy.

Requirements or Specifications

Functional Requirements

- **Query Assistance**: Answer Python, Java, and C++ questions with examples.
- **Algorithm Explanation**: Explain algorithms with code examples in Python.
- **Debugging**: Identify and fix code issues with explanations.
- **Code Optimization**: Suggest and explain code improvements.
- **Practice Problems**: Recommend problems with hints and solutions.
- **Web Interface**: Provide a simple, user-friendly interface for queries and responses.
- **Error Handling**: Display user-friendly error messages for invalid inputs or system issues.
- **API Endpoints**: Implement endpoints for all core functionalities (/query, /algorithm, /debug, etc.).

Non-Functional Requirements

- **Performance**: Respond to queries within 5–10 seconds.
- **Reliability**: Ensure consistent availability and low error rates.
- **Usability**: Maintain an intuitive interface and clear response formatting.
- **Privacy**: Operate locally without transmitting user data externally.
- **Compatibility**: Work on major browsers and modern OS platforms.
- **Security**: Restrict access to localhost and avoid sensitive data logging.

- **Maintainability**: Follow clean coding practices with sufficient comments and logs.

Conclusion

The Programming Assistance AI Bot is an innovative solution for students seeking programming help. By leveraging the capabilities of the LLaMA model and providing an intuitive web-based interface, the bot offers immediate assistance, clear explanations, and actionable insights. Its local hosting ensures data privacy and provides a cost-effective alternative to online AI tools. This project demonstrates the potential of AI-driven learning aids and sets the stage for future enhancements like additional language support, voice-based queries, and scalability through cloud deployment.