Project Brief: AI Testing Agent

Project Title: AI Testing Agent

Project Overview:

The AI Testing Agent is designed to assist users in evaluating the performance of their AI models by comparing expected results with actual outcomes. This tool will allow users to upload a PDF containing expected results, a screenshot of the actual results, and a text prompt of the question they want to test. Users can also provide additional project details, context, and specific commands to guide the AI in its evaluation. The AI Testing Agent will generate a detailed comparison and evaluation, as well as a concise summary based on a user-defined word count.

Objectives:

1. Develop a user-friendly interface for uploading necessary files and inputting test queries and additional information.
2. Implement an AI-based comparison engine to evaluate expected and actual results.
3. Provide detailed and summarised evaluations with easy copy functionality.
4. Ensure the system supports various file formats and handles large volumes of data efficiently.

Key Features:

1. **File Upload**: Allow users to upload PDFs (expected results), screenshots (actual results), and text prompts.
2. **Project Details**: Enable users to input their project name and additional details.
3. **Context Box**: Provide an extra context box for specific commands or additional information about the test.
4. **Comparison and Evaluation**: Generate a comparison between expected and actual results and provide a comprehensive evaluation in light of the context and test query.
5. **Copy Functionality**: Allow users to copy the full AI evaluation and a shorter summary (based on a user-defined word count) with two separate copy buttons.

Technical Specifications:

1. **Programming Language**: Python
2. **Frameworks and Libraries**:
    - Flask or Django for the web application
    - OpenCV or PIL for processing screenshots
    - Py2PDF or PDFPlumber for extracting data from PDFs
    - Natural language processing tools like NLTK or SpaCy for text analysis

3. **Database**: SQLlite for storing user inputs and test results
4. **Deployment**: Docker for containerisation and deployment on cloud services like AWS or Azure
5. **Version Control**: Git for source code management

Expected Outcomes:

- A fully functional AI Testing Agent application.
- Improved efficiency and accuracy in AI model testing and evaluation.
- Enhanced user experience through a simple and intuitive interface.

Risks and Mitigations:

1. **Data Privacy:** Ensure all user data is securely stored and accessed. Implement encryption and access control measures.
2. **Performance:** Optimize the comparison and evaluation logic to handle large datasets efficiently.
3. **Integration Challenges:** Test the integration with different file formats and ensure seamless functionality.