\*\*AI Python Developer Assessment\*\*  
  
## 1. Project Overview  
  
This assessment evaluates a candidate's ability to develop an AI-powered system that processes document files and checks their compliance with English guidelines. The system will include:  
  
\* A Python-based API to accept input files (PDF or Word).  
\* An AI agent to assess the document against provided English guidelines.  
\* An interactive feature allowing users to request guideline-compliant modifications.  
  
## 2. Scope of Work  
  
### API Development  
  
\* Develop an API endpoint using a Python web framework (Flask, FastAPI, Django, etc.) to accept PDF or Word document uploads.  
\* Ensure secure file upload handling and validation.   
\* Process the uploaded document for further analysis.  
  
### AI Agent Implementation  
  
\* Create an AI agent that checks if the uploaded document complies with the given English guidelines.  
\* The agent should parse the document, evaluate grammar, sentence structure, clarity, and adherence to writing rules.  
\* The agent should generate a detailed report outlining compliance or violations of the guidelines.  
  
### User Interaction & Compliance Correction  
  
\* Implement a feature enabling users to request the AI agent to modify the document to ensure compliance with guidelines.  
\* Provide the modified document for download.  
  
### Testing & Validation  
  
\* Implement unit and integration tests to ensure system functionality.  
\* Validate the AI agent’s performance by testing with various document samples.  
\* Verify that API endpoints function correctly with different file formats.  
  
## 3. Technical Considerations  
  
### API Development  
  
\* Utilize a Python framework like Flask, FastAPI, or Django.  
\* Ensure efficient and secure handling of file uploads.   
\* Implement appropriate validation and error handling.  
  
### AI Model Implementation   
  
\* Use NLP models such as OpenAI GPT, spaCy, or LanguageTool for guideline checking.  
\* Process text from PDF and Word documents efficiently.  
\* Generate meaningful compliance reports based on detected errors.  
  
### Performance Optimization  
  
\* Optimize text extraction and NLP processing for large documents.  
\* Ensure API response times remain minimal.  
  
## 4. Deliverables  
  
\* A Python-based API accepting PDF/Word document uploads.  
\* An AI agent capable of assessing guideline compliance.  
\* Comprehensive test reports with validation results.   
  
## 5. Timeline   
  
Candidates are expected to complete this assessment within 3 days. Submit the project via a public GitHub repository and share the URL.   
  
## 6. Risk Assessment  
  
\* \*\*Accuracy of AI Model:\*\* The AI agent may not perfectly detect all compliance issues. Mitigation includes using well-trained NLP models and refining rule sets.  
\* \*\*Performance Issues:\*\* Processing large files may cause slow responses. Efficient algorithms and caching strategies can help optimize performance.  
  
## 7. Conclusion  
  
This assessment evaluates candidates’ expertise in Python API development, AI-powered text analysis, and interactive system design. A successful solution will be functional, efficient, and secure, demonstrating the candidate’s ability to develop AI-driven applications.