**Full-Stack Developer (Backend Specialist)**

**Technical Assessment**

Thank you for your interest in the Full-Stack Developer position. This assessment is designed to evaluate your skills and abilities in developing a feature for parsing and processing uploaded documents, specifically pitch decks. Please read the instructions carefully and complete the tasks within the specified timeframe.

**Instructions:**

You are required to develop a web application that parses uploaded pitch deck documents (PDF or PowerPoint) and displays relevant information in a table format on a simple dashboard. The solution should include the following functionalities:

**Backend:**

- **API Design:** Design and implement a REST API endpoint using a Python framework (Please use **Flask** for this assessment) that accepts file uploads (PDF or PowerPoint) and securely saves them to a designated storage location (use the local filesystem for this assessment).
- **Data Parsing:** Develop a module that utilizes libraries like Apache POI (for PPTX) or PyPDF2 (for PDF) to parse the uploaded documents. Extract relevant information like slide titles, text content, and any embedded metadata.
- **Data Storage:** Implement a database schema or data model (e.g., using a relational database like PostgreSQL or a NoSQL database like MongoDB) to store the extracted information for future retrieval and analysis.
- **Error Handling:** Ensure robust error handling and validation to address scenarios such as:
    - Unsupported file formats
    - Corrupted files
    - Exceeding file size limits
    - Database connection issues
- **Deployment File:** Build a deployment file using docker compose, with at least an API Gateway or Broker service, a Parsing service and a database Service. Demonstrating your knowledge of a cache service and a queuing service will be a plus.

**Submission Guidelines:**

- **Submission:** Submit your repository's link in the **submission form** by **Friday Mar 28th, 11:59 PM WAT.**
- **README File:** Update the README file with detailed instructions on how to set up, configure, run, and test your solution. Include a description of the API route as well.
- **Version Control:** Create a public Github repository and commit your code to the master branch.

**Evaluation Criteria:**

- **Code Quality:** Clean, well-commented, and well-organized code following best practices and industry standards.
- **Functionality:** Correctness and completeness of the implemented features for parsing, storing, and displaying data.
- **Error Handling:** Robust error handling to address various potential issues.
- **Documentation:** Clear and concise instructions in the README file for setup and testing.

**Additional Notes:**

- Feel free to use any libraries or frameworks you are comfortable with within the Python ecosystem.
- Consider implementing unit tests for critical functionalities.
- While user interface design isn't a major focus, a basic and responsive dashboard layout is appreciated.

Good luck with the assessment, and we look forward to reviewing your submission! If you have any questions, please reach out to **contact@vester.ai**.