Title: AI-Based Marksheet Extraction Project

---

1. Extraction Approach: - Extract raw text from PDF marksheets using Python libraries such as PyPDF2 or pdfplumber. - Pass the extracted text to OpenAI GPT-4 or GPT-3.5 models for parsing and generating structured JSON output. - JSON structure example:

```
{
  "Candidate Name": "",
  "Father's Name": "",
  "Mother's Name": "",
  "Roll No": "",
  "Registration No": "",
  "DOB": "",
  "Exam Year": "",
  "Board/University": "",
  "Institution": "",
  "Subjects": [
      {"Subject": "", "Marks": "", "Max Marks": "", "Grade": ""}
  ]
}
```

- The AI model identifies candidate details, subject marks, grades, and other relevant fields from the raw text.

2. Confidence Logic: - The model provides soft confidence scores for field extraction. - Implement checks for mandatory fields (Candidate Name, Roll No, Subjects). - If any key fields are missing or malformed, flag the output for review.

3. Design Choices: - **Front-end:** Streamlit for quick PDF upload interface and JSON display. - **Backend:** OpenAI API for text extraction and parsing. - **Security:** API keys stored in `.streamlit/secrets.toml` to prevent exposure. - Optional: FastAPI endpoint can be used to make the extraction service programmatically accessible.

4. Optional Enhancements: - Integrate OCR (Tesseract) for scanned PDFs. - Support batch PDF uploads. - Error handling for incomplete or malformed PDFs. - Provide confidence scores and highlight potential errors in extracted data.

5. Conclusion: This approach leverages AI for automated marksheet parsing, reducing manual effort and improving accuracy. The system is modular, with Streamlit front-end, AI-based parsing, and optional API endpoints for integration.

End of Document.