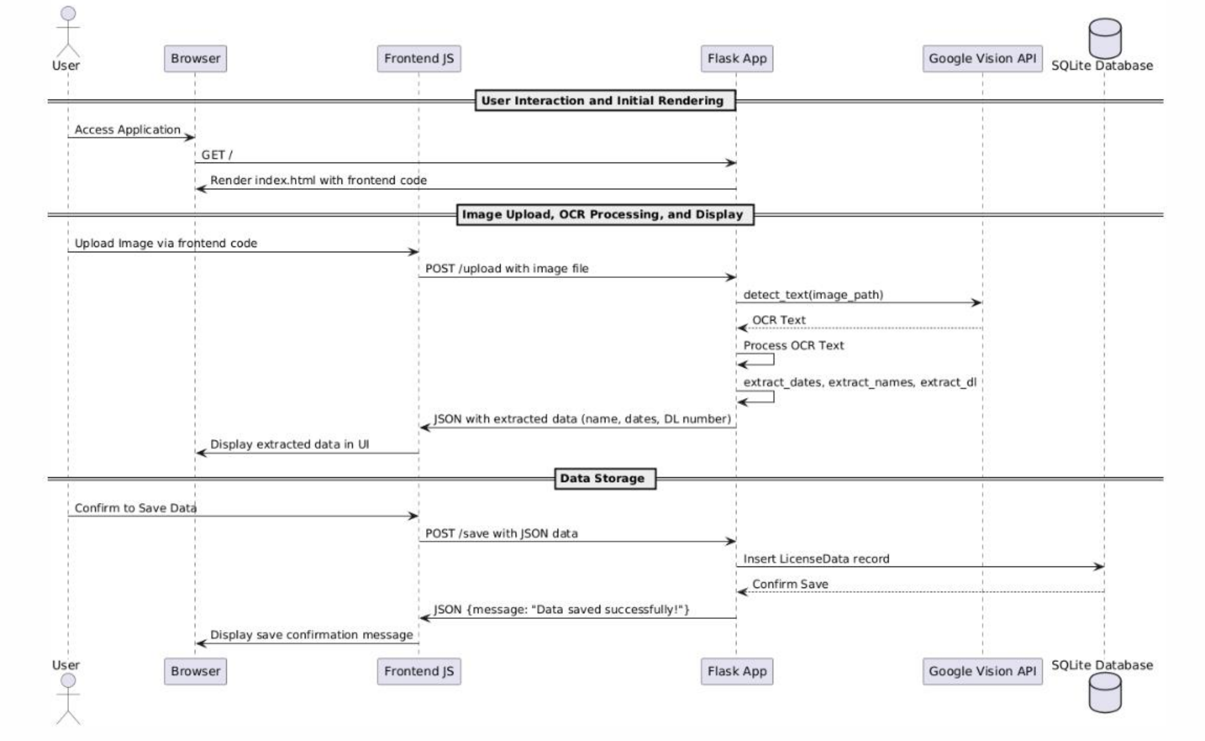
# Software Engineer- Technical Assessment

## Problem Statement:

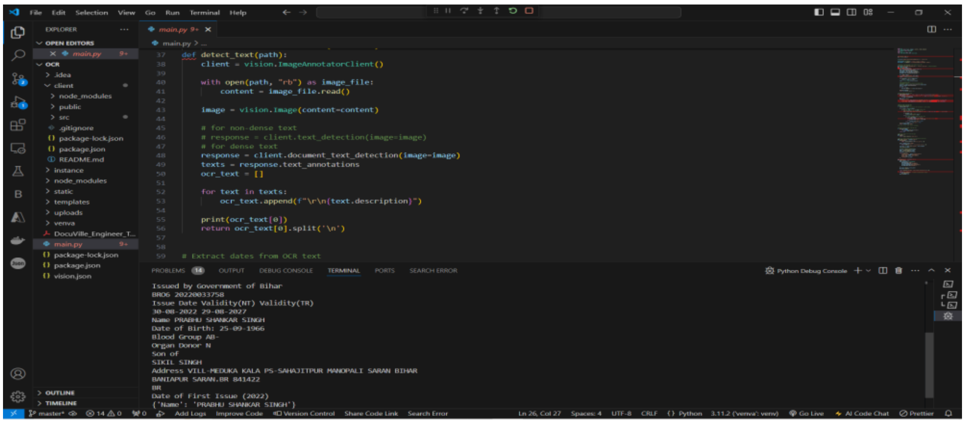
You are required to capture key information from one of the following important documents: passport, driver's license, or any other government-issued ID. You may choose one document to proceed with. Ensure that you clearly identify and extract key details such as name, document number, issue and expiration dates, and any other relevant information.

## System Design:


User
Browser
Frontend JS
Flask App
Google Vision API
SQLite Database
User Interaction and Initial Rendering
Access Application
GET /
Render index.html with frontend code
Image Upload, OCR Processing, and Display
Upload Image via frontend code
POST /upload with image file
detect_text(image_path)
OCR Text
Process OCR Text
extract_dates, extract_names, extract_dl
JSON with extracted data (name, dates, DL number)
Display extracted data in UI
Data Storage
Confirm to Save Data
POST /save with JSON data
Insert LicenseData record
Confirm Save
JSON {message: "Data saved successfully!"}
Display save confirmation message
User
Browser
Frontend JS
Flask App
Google Vision API
SQLite Database

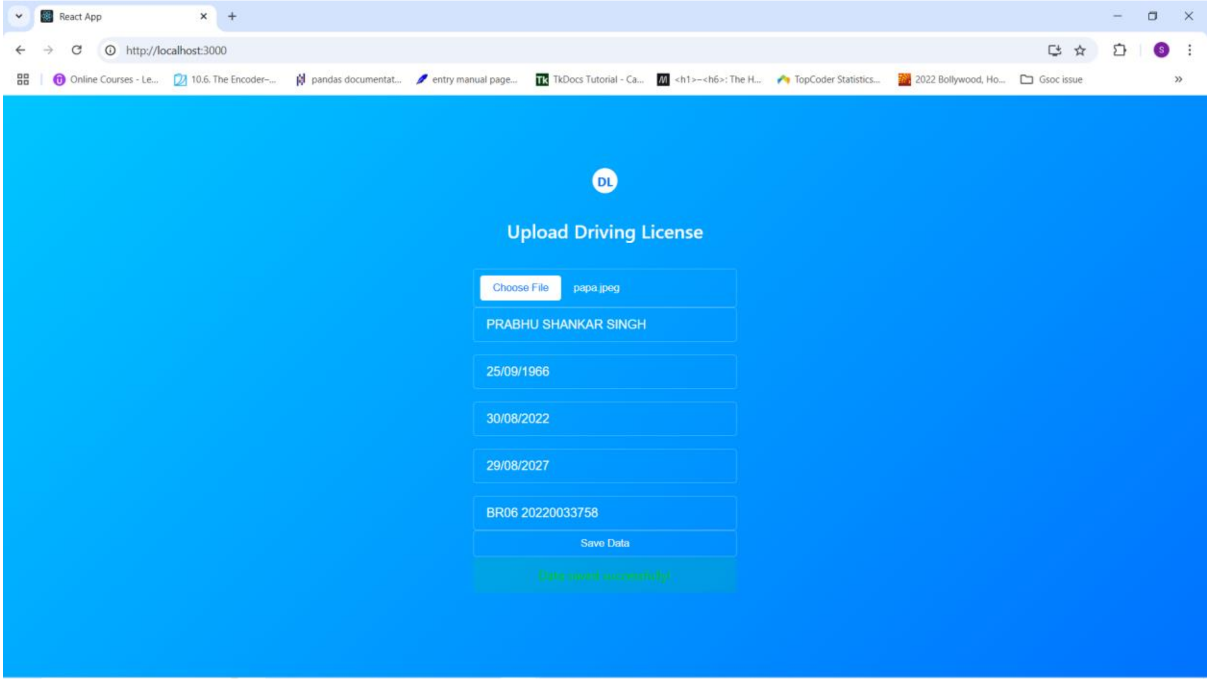
## Screenshots or Working Version:

This is the working image of this Assignment. I will upload a video of the demo as well.



```python
def detect_text(path):
    client = vision.ImageAnnotatorClient()

    with open(path, "rb") as image_file:
        content = image_file.read()

    image = vision.Image(content=content)

    # for non-dense text
    # response = client.text_detection(image=image)
    # for dense text
    response = client.document_text_detection(image=image)
    texts = response.text_annotations
    ocr_text = []

    for text in texts:
        ocr_text.append(f"\r\n{text.description}")

    print(ocr_text[0])
    return ocr_text[0].split('\n')


# Extract dates from OCR text
```

```
Issued by Government of Bihar
BR06 20220033758
Issue Date Validity(NT) Validity(TR)
30-08-2022 29-08-2027
Name PRABHU SHANKAR SINGH
Date of Birth: 25-09-1966
Blood Group AB-
Organ Donor N
Son of
SIKIL SINGH
Address VILL-MEDUKA KALA PS-SAHAJITPUR MANOPALI SARAN BIHAR
BANIAPUR SARAN.BR 841422
BR
Date of First Issue (2022)
{'Name': 'PRABHU SHANKAR SINGH'}
```



Upload Driving License

papa.jpeg

PRABHU SHANKAR SINGH

25/09/1966

30/08/2022

29/08/2027

BR06 20220033758

Data saved successfully!

**Feedback and Opportunities for Iteration**

This exercise involved creating a Flask application that utilizes the Google Vision API for Optical Character Recognition (OCR) to extract information from images of driver's licenses. Overall, it was an engaging and educational challenge that allowed me to apply my knowledge of web development, API integration, and data handling.

Enjoyable Aspects:

- Integration of Technologies: I enjoyed the process of integrating various technologies, such as Flask for the web framework, SQLAlchemy for database management, and Google Vision API for OCR and React. It was satisfying to see them work together to achieve the desired functionality.
- Problem-Solving: Tackling the extraction of specific data (like names, dates, and license numbers) from potentially unstructured text was a stimulating problem. It required thoughtful consideration of regular expressions and string manipulation techniques.

Difficulties Encountered:

- Regex Complexity: Crafting the correct regular expressions to accurately capture dates and license numbers was challenging. It required multiple iterations and testing to ensure they worked across various formats.
- Error Handling: Implementing robust error handling for file uploads and OCR processing was somewhat tricky. Ensuring the application gracefully handles different types of user input and unexpected errors is crucial for a good user experience.

Improvement Opportunities:

Given more time, I would consider the following improvements:

1. Scalability:
   - Database Optimization: If the application were to handle a large volume of data, I would explore using a more scalable database solution (like PostgreSQL or MongoDB) instead of SQLite. This would allow for better performance and concurrent access.
   - Load Balancing: Implementing load balancing and deploying the application on a cloud platform could help manage increased traffic.
2. Security:
   - File Upload Security: I would enhance security around file uploads by implementing checks for file types and sizes, as well as sanitizing file names to prevent directory traversal attacks.
   - Data Protection: Implementing encryption for sensitive data stored in the database, such as personal information from driver's licenses, would help protect user privacy.
3. User Experience:
   - Frontend Enhancements: Improving the frontend with a more user-friendly interface, including better error messages.
   - Feedback Mechanism: Adding a feedback mechanism for users to report issues or suggest features could help in refining the application based on user needs.
4. Testing and Documentation:
   - Unit Testing: Developing a suite of unit tests for the backend logic to ensure reliability and facilitate future changes would be beneficial.
   - Comprehensive Documentation: Providing detailed documentation for both users and developers would help in onboarding and maintenance of the application.

In conclusion, this exercise provided a valuable opportunity to develop practical skills in web application development and data processing. The challenges faced were rewarding to overcome, and the potential for improvement offers exciting avenues for future development.