Project Title	BizCardX: Extracting Business Card Data with OCR
Technologies	OCR,streamlit GUI, SQL,Data Extraction

Problem Statement:

You have been tasked with developing a Streamlit application that allows users to upload an image of a business card and extract relevant information from it using easyOCR. The extracted information should include the company name, card holder name, designation, mobile number, email address, website URL, area, city, state, and pin code. The extracted information should then be displayed in the application's graphical user interface (GUI).

In addition, the application should allow users to save the extracted information into a database along with the uploaded business card image. The database should be able to store multiple entries, each with its own business card image and extracted information.

To achieve this, you will need to use Python, Streamlit, easyOCR, and a database management system like SQLite or MySQL. The application should have a simple and intuitive user interface that guides users through the process of uploading the business card image and extracting its information. The extracted information should be displayed in a clean and organized manner, and users should be able to easily add it to the database with the click of a button. And Allow the user to Read the data, Update the data and Allow the user to delete the data through the streamlit UI

This project will require skills in image processing, OCR, GUI development, and database management. It will also require you to carefully design and plan the application architecture to ensure that it is scalable, maintainable, and extensible. Good documentation and code organization will also be important for this project.

Approach:

1. Install the required packages: You will need to install Python, Streamlit, easyOCR, and a database management system like SQLite or MySQL.
2. Design the user interface: Create a simple and intuitive user interface using Streamlit that guides users through the process of uploading the business card image and extracting its information. You can use widgets like file uploader, buttons, and text boxes to make the interface more interactive.

3. Implement the image processing and OCR: Use easyOCR to extract the relevant information from the uploaded business card image. You can use image processing techniques like resizing, cropping, and thresholding to enhance the image quality before passing it to the OCR engine.
4. Display the extracted information: Once the information has been extracted, display it in a clean and organized manner in the Streamlit GUI. You can use widgets like tables, text boxes, and labels to present the information.
5. Implement database integration: Use a database management system like SQLite or MySQL to store the extracted information along with the uploaded business card image. You can use SQL queries to create tables, insert data, and retrieve data from the database, Update the data and Allow the user to delete the data through the streamlit UI
6. Test the application: Test the application thoroughly to ensure that it works as expected. You can run the application on your local machine by running the command streamlit run app.py in the terminal, where app.py is the name of your Streamlit application file.
7. Improve the application: Continuously improve the application by adding new features, optimizing the code, and fixing bugs. You can also add user authentication and authorization to make the application more secure.

Results:

The result of the project would be a Streamlit application that allows users to upload an image of a business card and extract relevant information from it using easyOCR. The extracted information would include the company name, card holder name, designation, mobile number, email address, website URL, area, city, state, and pin code. The extracted information would then be displayed in the application's graphical user interface (GUI).

The application would also allow users to save the extracted information into a database along with the uploaded business card image. The database would be able to store multiple entries, each with its own business card image and extracted information.

The final application would have a simple and intuitive user interface that guides users through the process of uploading the business card image and extracting its information. The extracted information would be displayed in a clean and organized manner, and users would be able to easily add it to the database with the click of a button.

The project would require skills in image processing, OCR, GUI development, and database management. It would also require careful design and planning of the application architecture to ensure that it is scalable, maintainable, and extensible. Good documentation and code organization would also be important for the project. Overall, the result of the project would be a useful tool for businesses and individuals who need to manage business card information efficiently.

Dataset:

Dataset Link: Data Link

Project Evaluation metrics:

- You are supposed to write a code in a modular fashion (**in functional blocks**)
- Maintainable: It can be maintained, even as your codebase grows.
- Portable: It works the same in every environment (operating system)
- You have to maintain your code on **GitHub**.(Mandatory)
- You have to keep your **GitHub** repo public so that anyone can check your code.(Mandatory)
- Proper readme file you have to maintain for any project development(Mandatory)
- You should include basic workflow and execution of the entire project in the readme file on **GitHub**
- Follow the coding standards: https://www.python.org/dev/peps/pep-0008/
- You need to Create a Demo/Presentation video of your Project and post in **LinkedIn**(Mandatory)