Card Status API with Docker

This project provides a Flask-based API for retrieving the status of a user's card. The API interacts with a SQLite database and loads card status data from CSV files provided by partner companies.

Project Structure

- **app.py**: Main Flask application file containing API endpoints and database interaction logic.
- **Dockerfile**: Instructions for building a Docker image for the Flask application.
- **requirements.txt**: List of Python dependencies required by the Flask application.
- **data/**: Directory containing sample CSV files with card status information.

Prerequisites

- Docker installed on your system
- Postman installed in system(optional)

Steps for Running docker Image

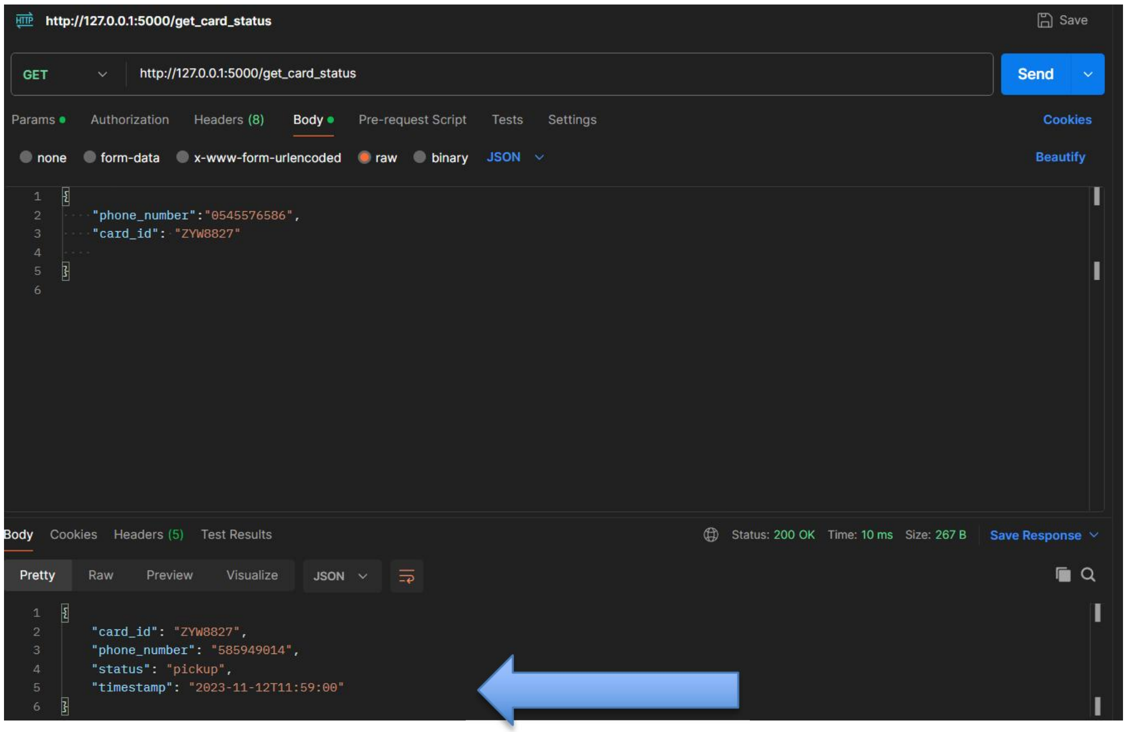
Step1: Pull image from docker Hub using below command
Command:- **docker pull anubhav670/card_status_img**

Step2: Running docker image with port no 5000
Command:- **docker run -dit --name card_container -p 5000:5000 anubhav670/card_status_img**

Step3: Open **http://127.0.0.1:5000/get_card_status** in browser or postman

Output

1. Sending a data in card status request in postman and we get the status of delivery



http://127.0.0.1:5000/get_card_status

GET http://127.0.0.1:5000/get_card_status

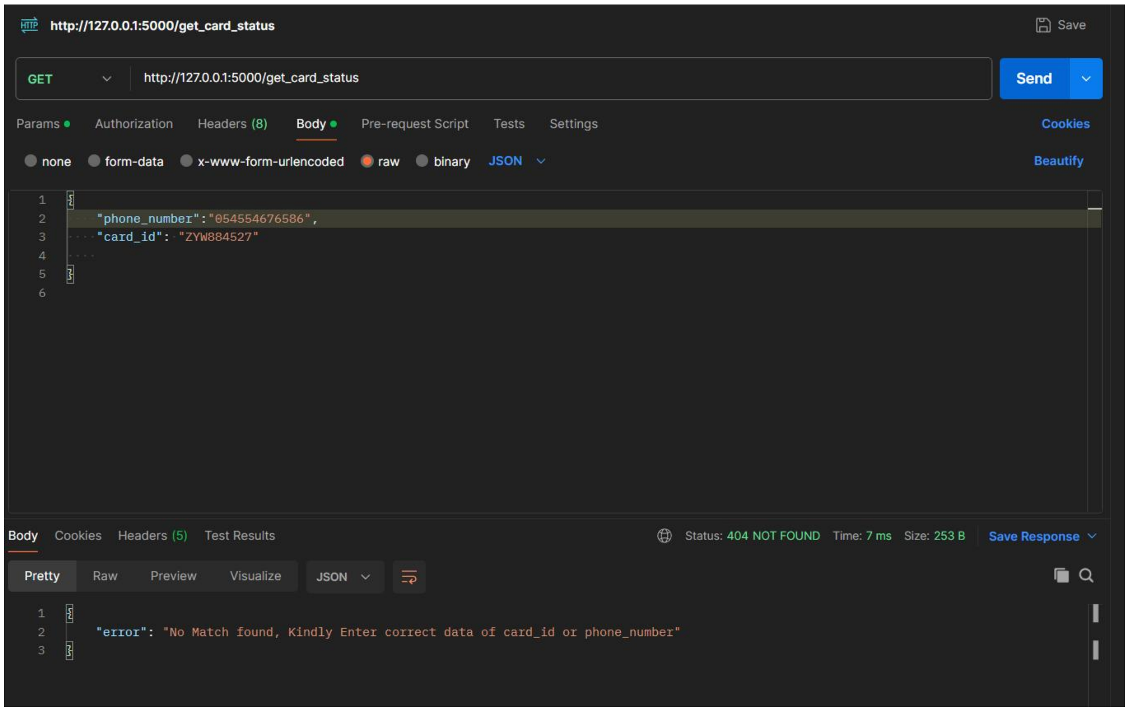
Body: raw JSON

```
{
    "phone_number":"0545576586",
    "card_id": "ZYW8827"
    
}
```

Status: 200 OK Time: 10 ms Size: 267 B

```
{
    "card_id": "ZYW8827",
    "phone_number": "585949014",
    "status": "pickup",
    "timestamp": "2023-11-12T11:59:00"
}
```

2. Client sending a Wrong data to api and getting error response



http://127.0.0.1:5000/get_card_status

GET http://127.0.0.1:5000/get_card_status

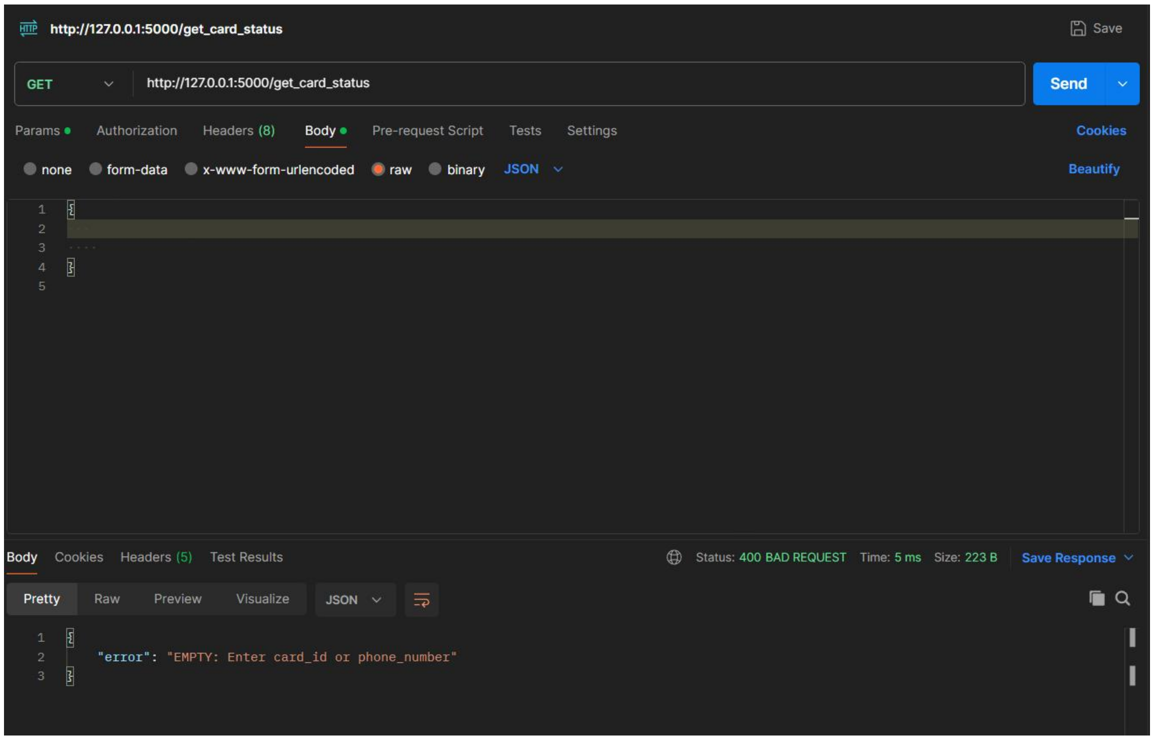
Body: raw JSON

```
{
    "phone_number":"054554676586",
    "card_id": "ZYW884527"
    
}
```

Status: 404 NOT FOUND Time: 7 ms Size: 253 B

```
{
    "error": "No Match found, Kindly Enter correct data of card_id or phone_number"
}
```

3. If Client send no data to api, it will result in error generation



URL: http://127.0.0.1:5000/get_card_status

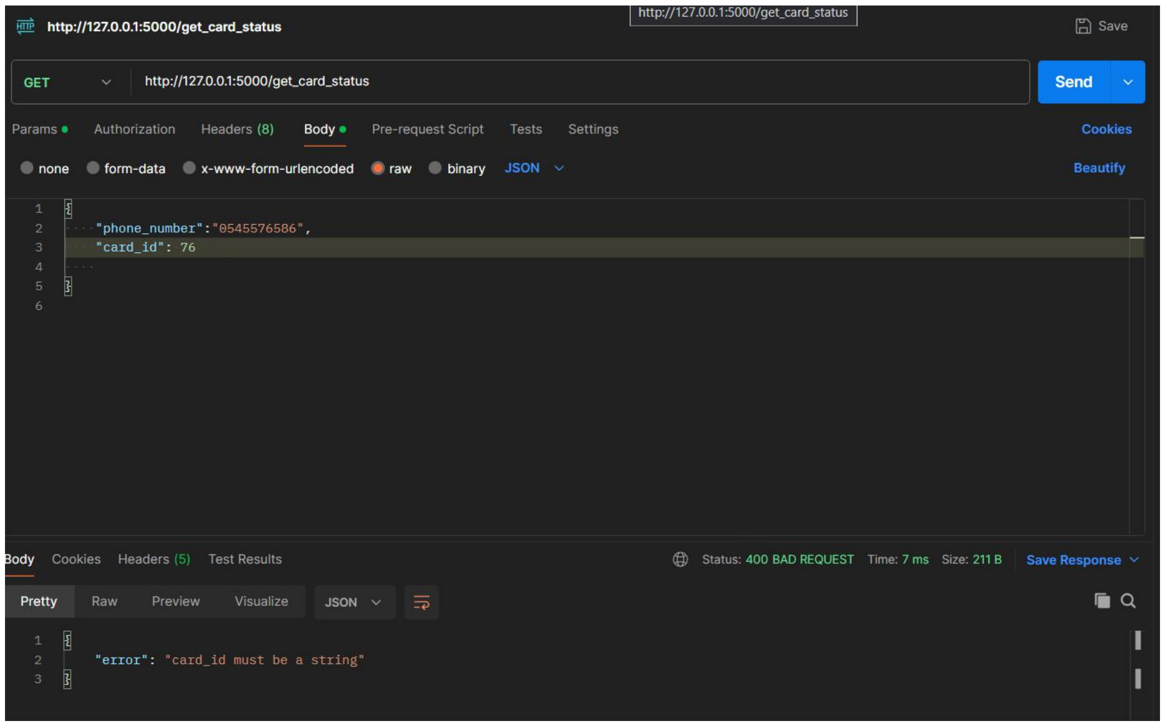
Body: raw, JSON

```
{

}
```

Status: 400 BAD REQUEST Time: 5 ms Size: 223 B

```
{
    "error": "EMPTY: Enter card_id or phone_number"
}
```

4. If client enter a wrong format then also it will generate error.



URL: http://127.0.0.1:5000/get_card_status

Body: raw, JSON

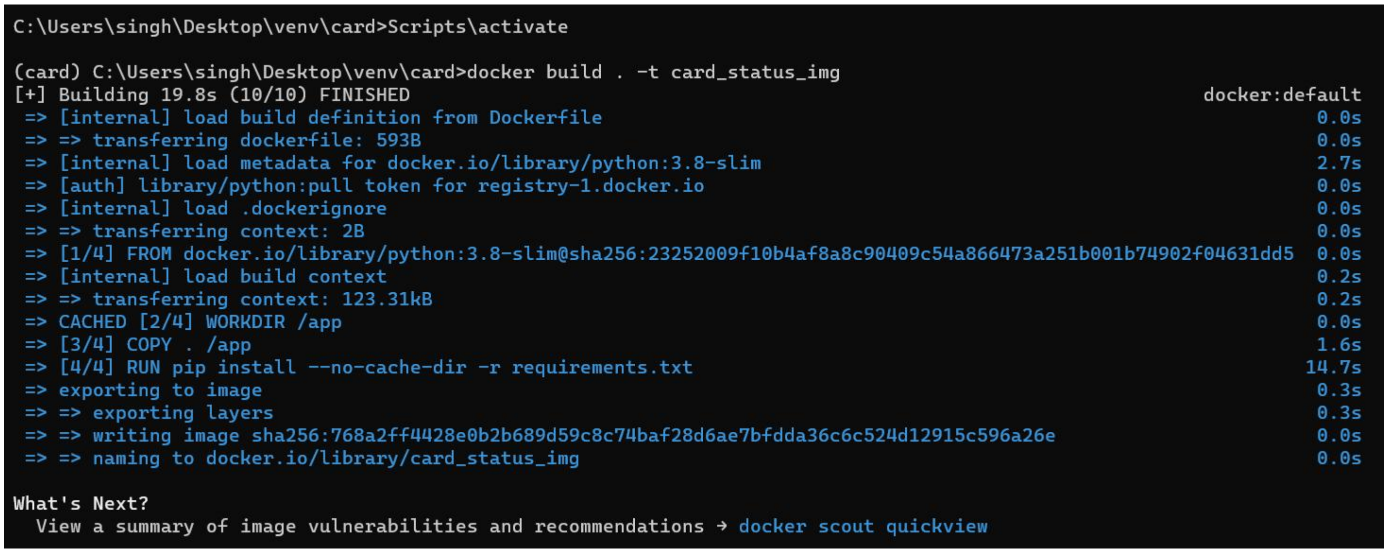
```
{
    "phone_number":"0545576586",
    "card_id": 76

}
```

Status: 400 BAD REQUEST Time: 7 ms Size: 211 B

```
{
    "error": "card_id must be a string"
}
```

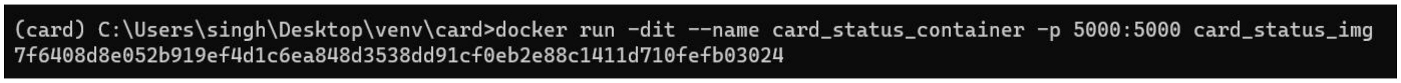
Steps followed for Creating docker Image

Step1: First we create a virtual environment and then created a docker Image using command shown below in image



```
C:\Users\singh\Desktop\venv\card>Scripts\activate

(card) C:\Users\singh\Desktop\venv\card>docker build . -t card_status_img
[+] Building 19.8s (10/10) FINISHED                                                           docker:default
 => [internal] load build definition from Dockerfile                                                    0.0s
 => => transferring dockerfile: 593B                                                                    0.0s
 => [internal] load metadata for docker.io/library/python:3.8-slim                                      2.7s
 => [auth] library/python:pull token for registry-1.docker.io                                           0.0s
 => [internal] load .dockerignore                                                                       0.0s
 => => transferring context: 2B                                                                         0.0s
 => [1/4] FROM docker.io/library/python:3.8-slim@sha256:23252009f10b4af8a8c90409c54a866473a251b001b74902f04631dd5  0.0s
 => [internal] load build context                                                                       0.2s
 => => transferring context: 123.31kB                                                                   0.2s
 => CACHED [2/4] WORKDIR /app                                                                           0.0s
 => [3/4] COPY . /app                                                                                   1.6s
 => [4/4] RUN pip install --no-cache-dir -r requirements.txt                                           14.7s
 => exporting to image                                                                                  0.3s
 => => exporting layers                                                                                 0.3s
 => => writing image sha256:768a2ff4428e0b2b689d59c8c74baf28d6ae7bfdda36c6c524d12915c596a26e           0.0s
 => => naming to docker.io/library/card_status_img                                                      0.0s

What's Next?
  View a summary of image vulnerabilities and recommendations → docker scout quickview
```

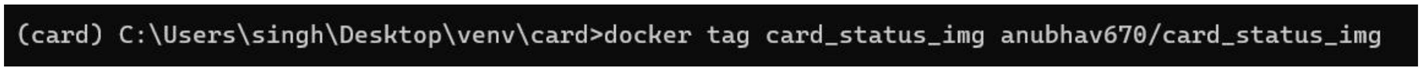
Step2: Once the Docker image is built successfully, you can run a Docker container based on the image using the following command:



```
(card) C:\Users\singh\Desktop\venv\card>docker run -dit --name card_status_container -p 5000:5000 card_status_img
7f6408d8e052b919ef4d1c6ea848d3538dd91cf0eb2e88c1411d710fefb03024
```

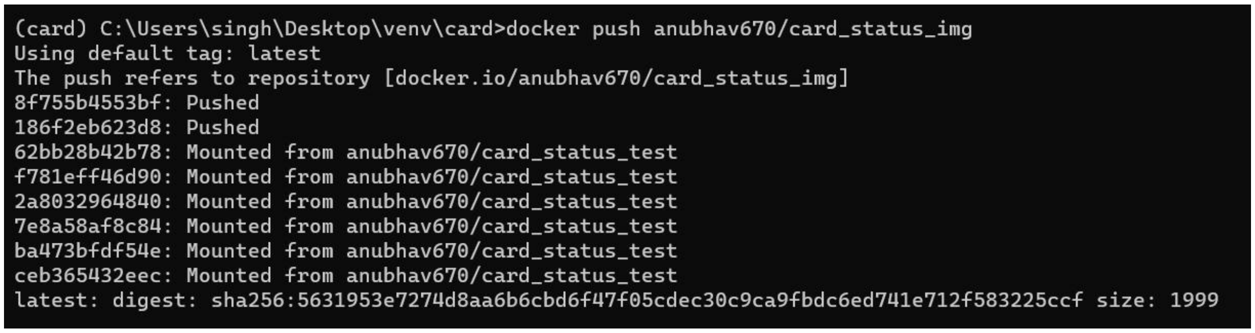
This command maps port 5000 of the Docker container to port 5000 on your host machine, allowing you to access the Flask application running inside the container.

Step3: Creating a docker tag Image using a below command.



```
(card) C:\Users\singh\Desktop\venv\card>docker tag card_status_img anubhav670/card_status_img
```

Step4: Pushing a docker tag image into a docker Hub



```
(card) C:\Users\singh\Desktop\venv\card>docker push anubhav670/card_status_img
Using default tag: latest
The push refers to repository [docker.io/anubhav670/card_status_img]
8f755b4553bf: Pushed
186f2eb623d8: Pushed
62bb28b42b78: Mounted from anubhav670/card_status_test
f781eff46d90: Mounted from anubhav670/card_status_test
2a8032964840: Mounted from anubhav670/card_status_test
7e8a58af8c84: Mounted from anubhav670/card_status_test
ba473bfdf54e: Mounted from anubhav670/card_status_test
ceb365432eec: Mounted from anubhav670/card_status_test
latest: digest: sha256:5631953e7274d8aa6b6cbd6f47f05cdec30c9ca9fbdc6ed741e712f583225ccf size: 1999
```