

Project Report

Introduction:

The provided code is a simple Flask API for managing contacts in a phone book. It allows users to add, delete, and list contacts, as well as view logs of these actions. Authentication is implemented using JWT tokens.

Code Description:

The code consists of three main components:

- **app.py:** The Flask application code that defines the endpoints, their logic, and their routes.
- **database.py:** Contains functions for creating a SQLite database, inserting, and deleting contacts, and inserting logs of actions taken by the user.
- **Dockerfile:** Defines the Docker image used to run the application.

The API has the following endpoints:

- `/login`: A POST endpoint that allows a user to log in and receive a JWT token.
- `/PhoneBook/list`: A GET endpoint that returns a list of all contacts in the phone book.
- `/PhoneBook/add`: A POST endpoint that allows a user to add a new contact to the phone book.
- `/PhoneBook/deleteByName`: A PUT endpoint that allows a user to delete a contact by name.
- `/PhoneBook/deleteByNumber`: A PUT endpoint that allows a user to delete a contact by phone number.
- `/PhoneBook/logs`: A GET endpoint that allows a user to view a log of all actions taken on the phone book.

Design of Regular Expressions:

The code uses regular expressions to validate the name and phone number fields entered by the user when adding a new contact. These regular expressions are defined as follows:

Name regex:

```
r'^[a-zA-Z ,'\-]+(?:[a-zA-Z']*)?(?:-[a-zA-Z']*)?(?:,[a-zA-Z ,'\-]+(?:[a-zA-Z']*)?(?:-[a-zA-Z']*)?)?$'
```

Explanation:

- The name must start with one or more alphabetic characters (upper or lower case).
- It can contain spaces, commas, periods, apostrophes, and hyphens.
- If the name contains an apostrophe or hyphen, it must be followed by at least one alphabetic character.
- If the name contains a comma, there must be at least one alphabetic character after the comma.

Phone number regex:

```
r'^(?!+01\d{3}\d{3}\d{3}-\d{4}$)(?!+1234\d{3}\d{3}\d{3}-\d{4}$)(?!7031111234)(\d{3}\d{3}\d{3}-\d{4}|+\d{1,4}[-.\s]?*\d{1,3}?\d{1,3}?\d{1,4}[-.\s]?*\d{1,4}[-.\s]?*\d{1,9})$'
```

Explanation:

- The phone number can start with a plus sign (+), an opening parenthesis followed by three digits, or any combination of digits and hyphens or dots.
- If the phone number starts with a plus sign, it can be followed by up to four digits, an optional separator (hyphen, dot, or space), and then any combination of digits and separators.

Name: Dishant Krishna Koli

Subject: 2232-CSE-5382-001-SECURE PROGRAMMING

UID: 1002067615

Topic: Phonebook APP - (REST API)

- If the phone number starts with an opening parenthesis followed by three digits, it must be followed by a closing parenthesis and then six digits with an optional separator (hyphen or space) after the third digit.
- The phone number cannot be equal to "7031111234".
- The phone number cannot match specific patterns: "+01 (XXX) XXX-XXXX" and "+1234 (201) XXX-XXXX".

Assumptions: The code assumes that the SQLite database is already created and will not attempt to create it again.