**Report: Input Validation assignment for CSE-5382**

**Name: Mrunmai Nitin Magar**

**UTA ID: 1002092125**

### **Instructions for Building and Running Software and Unit Tests**

* **Build Instructions:**
  + Ensure that you have Docker installed on your system.
  + Navigate to the project directory.
  + Build the Docker image using the provided Dockerfile:

docker build -t phonebook-app .

* **Running Software:**
  + Run the Docker container:

docker run -p 8000:8000 phonebook-app

* + The FastAPI application will be accessible at [http://localhost:8000](http://localhost:8000/).
* **Running Unit Tests:**
  + Ensure that the Docker container is running.
  + Open a new terminal.
  + Run the following command to execute the unit tests:

docker exec -it CONTAINER\_ID pytest testmain.py

1. **Description of How the Code Works**

#### **FastAPI Application Structure:**

The FastAPI application provides REST API endpoints for managing a phonebook. The key components include:

* Main Application (app.py):
  + Defines FastAPI app, SQLAlchemy models (PhoneBook, AuditLog), and API endpoints.
  + Uses Pydantic models for request and response validation.
* Database Models (app.py):
  + PhoneBook: Represents a person's entry in the phonebook.
  + AuditLog: Represents log entries for add, remove, or list operations.
* Session Management (main.py):
  + Utilizes SQLAlchemy to create an SQLite database and manage sessions.
* Validation (app.py):
  + Implements regular expressions for name and phone number validation.
  + Validates inputs before database operations.
* Audit Logging (app.py):
  + Adds audit log entries for add, remove, or list operations.

As per the bonus point requirement,

* Storing the data in phonebook.db using sqlite
* Parameterized api is used for put operations (deletebynumber and deletebyaname)

### A screenshot of a computer Description automatically generatedA screenshot of a computer Description automatically generatedA screenshot of a computer Description automatically generatedA screenshot of a computer Description automatically generated**Outputs:**

### **Assumptions Made**

* Database Backend:
  + SQLite is used as the database backend for simplicity and portability.
* Logging Requirements:
  + Assumes basic audit logging is sufficient; more extensive logging could be added based on specific needs.
* Input Validation:
  + Assumes basic validation using regular expressions; additional validation rules may be needed for specific use cases.

### **Pros/Cons of the Approach**

#### **4.1 Pros:**

* FastAPI and SQLAlchemy:
  + Leveraging FastAPI for efficient API development.
  + Utilizing SQLAlchemy for robust database interactions.
* Dockerized Application:
  + Easy deployment and testing using Docker containers.
* Input Validation:
  + Implemented input validation using regular expressions to ensure data integrity.
* Audit Logging:
  + Integrated audit logging to track key operations for accountability.

#### **4.2 Cons:**

* Database Choice:
  + SQLite is used for simplicity; might not be suitable for large-scale production use.
* Validation Scope:
  + Basic input validation; additional rules might be required based on specific business requirements.
* Logging Depth:
  + Audit logging is basic; more detailed logging could be added for in-depth analysis.

### **Conclusion**

The implemented FastAPI application provides a functional phonebook management system with basic input validation and audit logging. The use of Docker ensures easy deployment and testing. Further enhancements can be made based on specific requirements and scalability considerations.