**1st qst**

```
CREATE TABLE sim_cards (
    sim_number TEXT PRIMARY KEY,
    phone_number TEXT NOT NULL,
    status TEXT NOT NULL DEFAULT 'inactive',
    activation_date TIMESTAMP
);
```

2nd python

```
from flask import Flask, request, jsonify
import sqlite3
from datetime import datetime
app = Flask(__name__)
# Connect to the database
def get_db_connection():
    conn = sqlite3.connect('sim_cards.db')
    conn.row_factory = sqlite3.Row
    return conn
# Activate SIM Card
@app.route('/activate', methods=['POST'])
def activate_sim_card():
    data = request.json
    if 'sim_number' not in data:
        return jsonify({'error': 'SIM number is required'}), 400
    sim_number = data['sim_number']
    conn = get_db_connection()
    sim_card = conn.execute('SELECT * FROM sim_cards WHERE sim_number = ?', (sim_number,)).fetchone()
    if sim_card is None:
        return jsonify({'error': 'SIM card not found'}), 404
    if sim_card['status'] == 'active':
        return jsonify({'error': 'SIM card is already active'}), 400

    conn.execute('UPDATE sim_cards SET status = ?, activation_date = ? WHERE sim_number = ?', ('active', datetime.now(), sim_number))
    conn.commit()
    conn.close()
    return jsonify({'message': 'SIM card activated successfully'}), 200
```

```
# Deactivate SIM Card
@app.route('/deactivate', methods=['POST'])
def deactivate_sim_card():
    data = request.json
    if 'sim_number' not in data:
        return jsonify({'error': 'SIM number is required'}), 400

    sim_number = data['sim_number']
    conn = get_db_connection()
    sim_card = conn.execute('SELECT * FROM sim_cards WHERE sim_number = ?', (sim_number,)).fetchone()
    if sim_card is None:
        return jsonify({'error': 'SIM card not found'}), 404
    if sim_card['status'] == 'inactive':
        return jsonify({'error': 'SIM card is already inactive'}), 400

    conn.execute('UPDATE sim_cards SET status = ? WHERE sim_number = ?', ('inactive', sim_number))
    conn.commit()
    conn.close()
    return jsonify({'message': 'SIM card deactivated successfully'}), 200

# Get SIM Details
@app.route('/sim-details/<sim_number>', methods=['GET'])
def get_sim_details(sim_number):
    conn = get_db_connection()
    sim_card = conn.execute('SELECT * FROM sim_cards WHERE sim_number = ?', (sim_number,)).fetchone()
    if sim_card is None:
        return jsonify({'error': 'SIM card not found'}), 404
    conn.close()
    return jsonify(dict(sim_card)), 200

if __name__ == '__main__':
    app.run(debug=True)
```

3rd  Error Handling

The API endpoints handle the following error scenarios:

- SIM card not found (404)
- SIM card already in the desired state (400)
- Required input fields missing or invalid (400)

5th github Here's an example README file:

# SIM Card Activation Service

This is a simple SIM card activation service built using Flask and SQLite.

## Setup and Run

1. Clone the repository: `git clone https://github.com/your-username/sim-card-activation-service.git`
2. Install dependencies: `pip install -r requirements.txt`
3. Create the database: `sqlite3 sim_cards.db < schema.sql`
4. Run the application: `python app.py`

## API Endpoints

### Activate SIM Card

* Endpoint: `/activate`
* Method: `POST`
* Input: `{"sim_number": "your_sim_number"}`
* Response: `{"message": "SIM card activated successfully"}`

### Deactivate SIM Card

* Endpoint: `/deactivate`
* Method: `POST`
* Input: `{"sim_number": "your_sim_number"}`
* Response: `{"message": "SIM card deactivated successfully"}`

### Get SIM Details

* Endpoint: `/sim-details/<sim_number>`
* Method: `GET`
* Response: `{"sim_number": "your_sim_number", "phone_number": "your_phone_number", "status": "active/inactive", "activation_date": "activation_date"}`

## Testing

You can test the API endpoints using a