# Import Flask modules  
from flask import Flask, render\_template, request  
from flask\_sqlalchemy import SQLAlchemy

# Create an object named app  
app = Flask(\_\_name\_\_)

# Configure sqlite database  
app.config['SQLALCHEMY\_DATABASE\_URI'] = 'sqlite:///./email.db'  
app.config['SQLALCHEMY\_TRACK\_MODIFICATIONS']= False  
db = SQLAlchemy(app)# Execute the code below only once.

# Write sql code for initializing users table..  
drop\_table = 'DROP TABLE IF EXISTS users;'  
users\_table = """  
CREATE TABLE users(  
username VARCHAR NOT NULL PRIMARY KEY,  
email VARCHAR);  
"""  
data = """  
INSERT INTO users  
VALUES  
    ("Buddy Rich", "buddy@clarusway.com" ),  
    ("Candido", "candido@clarusway.com"),  
    ("Charlie Byrd", "charlie.byrd@clarusway.com");  
"""

db.session.execute(drop\_table)  
db.session.execute(users\_table)  
db.session.execute(data)  
db.session.commit()

# Write a function named `find\_emails` which find emails using keyword from the user table in the db,

# and returns result as tuples `(name, email)`.  
def find\_emails(keyword):  
    query=f"""  
    SELECT \* FROM users WHERE username LIKE '%{keyword}%';  
    """  
    result = db.session.execute(query)  
    user\_emails = [(row[0], row[1]) for row in result]  
    if not any(user\_emails):  
        user\_emails=[('Not Found', 'Not Found')]  
    return user\_emails

# Write a function named `insert\_email` which adds new email to users table the db.  
def insert\_email(name, email):  
    query=f"""  
    SELECT \* FROM users WHERE username LIKE '{name}';  
    """  
    result = db.session.execute(query)  
    #default response  
    response = 'Error occured...'  
    # if user input are None (null) give warning  
    if name == None or email == None:  
        response = 'Username or email can not be empty!!'  
    # if there is no same user name in the db, then insert the new one  
    elif not any(result):  
        insert =f"""  
        INSERT INTO users  
        VALUES ('{name}', '{email}')  
        """  
        result = db.session.execute(insert)  
        db.session.commit()  
        response= f'User {name} added successfully'  
    # if there is user with same name, then give warning  
    else:  
        response = f'User {name} already exits.'    return response# Write a function named `emails` which finds email addresses by keyword using `GET` and `POST` methods,  
# using template files named `emails.html` given under `templates` folder  
# and assign to the static route of ('/')  
@app.route('/', methods = ['GET', 'POST'])  
def emails():  
    if request.method == 'POST':  
        user\_name = request.form['username']  
        user\_emails = find\_emails(user\_name)  
        return render\_template('emails.html', name\_emails=user\_emails, keyword=user\_name, show\_result=True)  
    else:  
        return render\_template('emails.html', show\_result=False)# Write a function named `add\_email` which inserts new email to the database using `GET` and `POST` methods,  
# using template files named `add-email.html` given under `templates` folder  
# and assign to the static route of ('add')  
@app.route('/add', methods=['GET', 'POST'])  
def add\_email():  
    if request.method == 'POST':  
        user\_name = request.form['username']  
        user\_email = request.form['useremail']  
        result = insert\_email(user\_name, user\_email)  
        return render\_template('add-email.html', result=result, show\_result=True)  
    else:  
        return render\_template('add-email.html', show\_result=False)# Add a statement to run the Flask application which can be reached from any host on port 80.  
if \_\_name\_\_ == '\_\_main\_\_':  
    app.run(debug=True)  
    # app.run(host='0.0.0.0', port=80)