Generate 10 random numbers using numpy Close

```
import sqlite3

def init_db():
    conn = sqlite3.connect('tasks.db')
    cursor = conn.cursor()
    cursor.execute('''
        CREATE TABLE IF NOT EXISTS tasks (
            id INTEGER PRIMARY KEY AUTOINCREMENT,
            description TEXT NOT NULL,
            status INTEGER NOT NULL
        )
    ''')
    conn.commit()
    conn.close()

init_db()



from flask import Flask, render_template, request, redirect
import sqlite3

app = Flask(__name__)

# Database functions
def get_db_connection():
    conn = sqlite3.connect('tasks.db')
    conn.row_factory = sqlite3.Row
    return conn



@app.route('/')
def home():
    conn = get_db_connection()
    tasks = conn.execute('SELECT * FROM tasks').fetchall()
    conn.close()
    return render_template('home.html', tasks=tasks)

@app.route('/create', methods=('GET', 'POST'))
def create_task():
    if request.method == 'POST':
        description = request.form['description']
        conn = get_db_connection()
        conn.execute('INSERT INTO tasks (description, status) VALUES (?, ?)', (description, 0))
        conn.commit()
        conn.close()
        return redirect('/')
    return render_template('create.html')

@app.route('/update/<int:task_id>', methods=('GET', 'POST'))
def update_task(task_id):
    conn = get_db_connection()
    task = conn.execute('SELECT * FROM tasks WHERE id = ?', (task_id,)).fetchone()

    if request.method == 'POST':
        description = request.form['description']
        status = request.form['status']
        conn.execute('UPDATE tasks SET description = ?, status = ? WHERE id = ?', (description, status, task_id))
        conn.commit()
        conn.close()
        return redirect('/')

    conn.close()
    return render_template('update.html', task=task)

@app.route('/delete/<int:task_id>', methods=('POST',))
def delete_task(task_id):
    conn = get_db_connection()
    conn.execute('DELETE FROM tasks WHERE id = ?', (task_id,))
    conn.commit()
    conn.close()
    return redirect('/')
```

```
<!DOCTYPE html>
<html>
<head>
    <title>Task List</title>
</head>
<body>
    <h1>Task List</h1>
    <a href="/create">Create New Task</a>
    <ul>
        {% for task in tasks %}
        <li>
            {{ task.description }} - Status: {{ 'Complete' if task.status == 1 else 'Incomplete' }}
            <a href="/update/{{ task.id }}">Edit</a>
            <form action="/delete/{{ task.id }}" method="post" style="display:inline;">
                <button type="submit">Delete</button>
            </form>
        </li>
        {% endfor %}
    </ul>
</body>
</html>
```

```
  File "<ipython-input-4-b79317ef3cf3>", line 1
    <!DOCTYPE html>
    ^
SyntaxError: invalid syntax
```

Next steps: Fix error

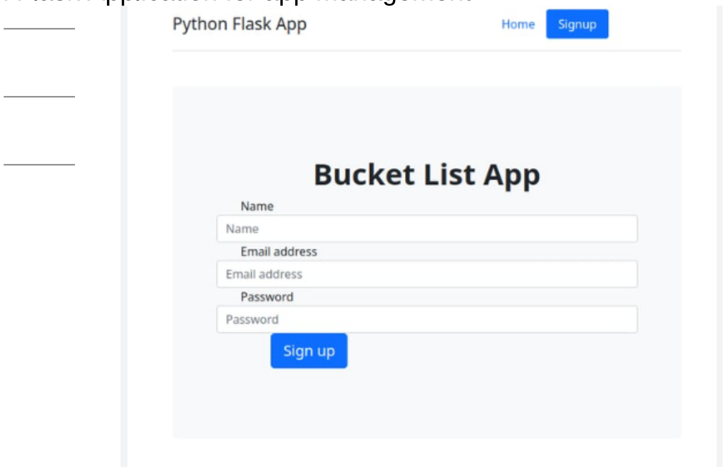
```
if __name__ == '__main__':
    app.run(debug=True)
```

```
 * Serving Flask app '__main__'
 * Debug mode: on
INFO:werkzeug:WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:5000
INFO:werkzeug:Press CTRL+C to quit
INFO:werkzeug: * Restarting with stat
```

Generate print hello world using rot13 Close

```
Waiting...
```

Python Flask Application for app management


Python Flask App
Home
Signup
Bucket List App
Name
Name
Email address
Email address
Password
Password
Sign up