**NGÀY 09/01/2025**

1. Tớ đã Cài đặt xong Python, **Git** (để quản lý code), IDE, Cài đặt Virtual Environment (Môi trường ảo) và đã kiểm tra phiên bản tốt rồi. Bây giờ tớ làm gì tiếp?

from flask import Flask

app = Flask(\_\_name\_\_)

@app.route("/")

def home():

    return "Chào mừng bạn đến với Website Bóng Bàn!"

if \_\_name\_\_ == "\_\_main\_\_":

    app.run(debug=True)

### 2. ****Bước 2: Thêm giao diện HTML.**** Bây giờ, chúng ta sẽ cải thiện giao diện trang web bằng cách sử dụng HTML và CSS.

### 

### 

### 3. Tiếp theo

Nếu giao diện hoạt động tốt, chúng ta sẽ tiếp tục:

1. **Tạo trang đăng ký để người dùng nhập thông tin.**
2. **Tích hợp dữ liệu từ form đăng ký vào Flask.**

### 

**4. Tiếp theo**

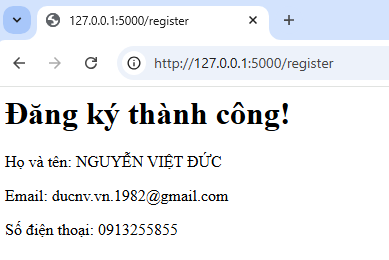
Nếu bạn đã hoàn thành bước này, chúng ta có thể:

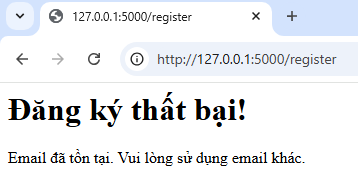
1. Lưu thông tin người dùng vào cơ sở dữ liệu.(done)
2. Thêm thông báo lỗi nếu thông tin nhập không hợp lệ. (done)
3. Hiển thị danh sách người dùng đã đăng ký. (done)

Bạn muốn tiếp tục với phần nào trước?

***4.1. Lưu thông tin người dùng vào cơ sở dữ liệu.***

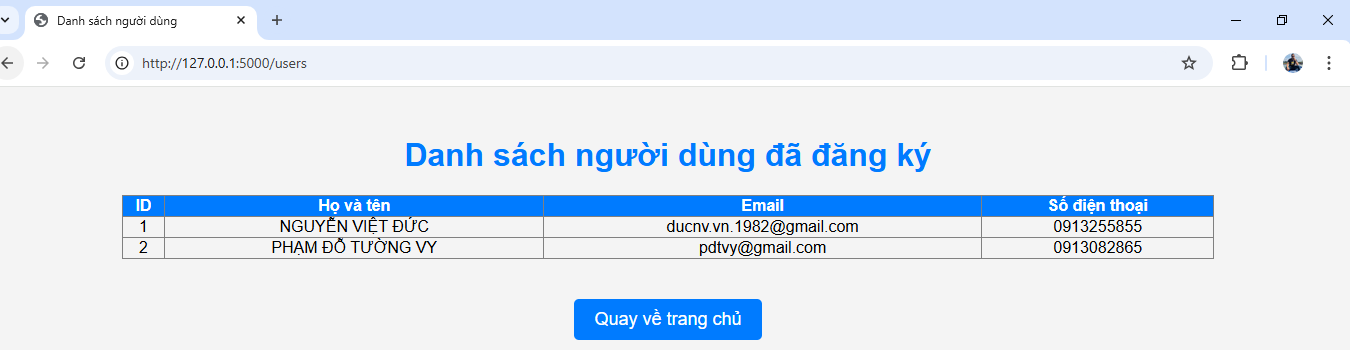
Để lưu thông tin người dùng vào cơ sở dữ liệu, chúng ta sẽ sử dụng **SQLite**, một cơ sở dữ liệu nhẹ được tích hợp sẵn trong Python. Dưới đây là từng bước chi tiết

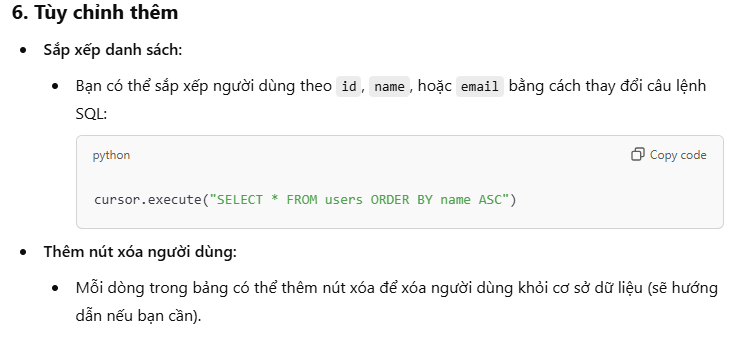




Nếu mọi thứ đã hoạt động ổn, chúng ta có thể:

**4.1.1. Hiển thị danh sách người dùng đã đăng ký.**



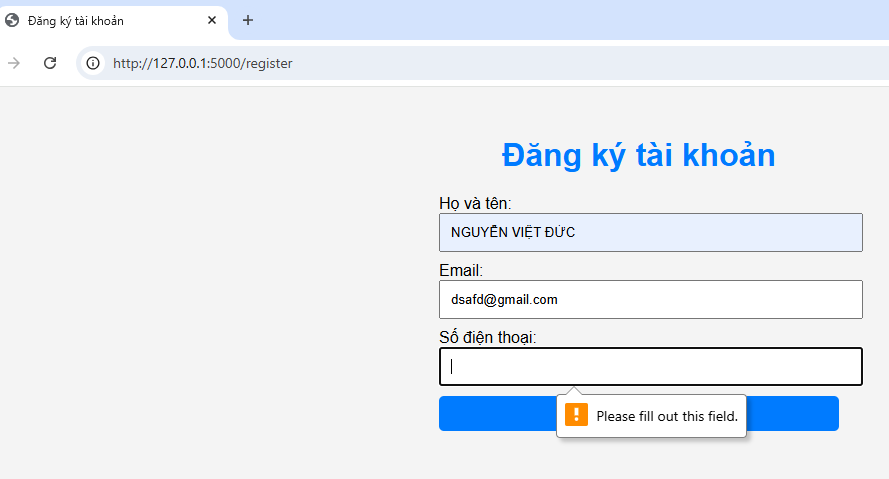


**4.1.2. Tích hợp thêm thông báo lỗi khi thông tin nhập không hợp lệ.**

**Kiểm tra dữ liệu nhập vào**

Chúng ta sẽ kiểm tra dữ liệu người dùng nhập vào để đảm bảo:

1. **Họ và tên không được để trống.**
2. **Email phải đúng định dạng.**
3. **Số điện thoại chỉ chứa số và không được để trống.**



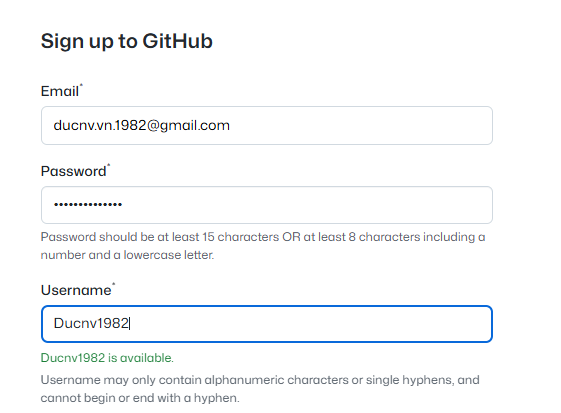
**5. Tiếp theo**

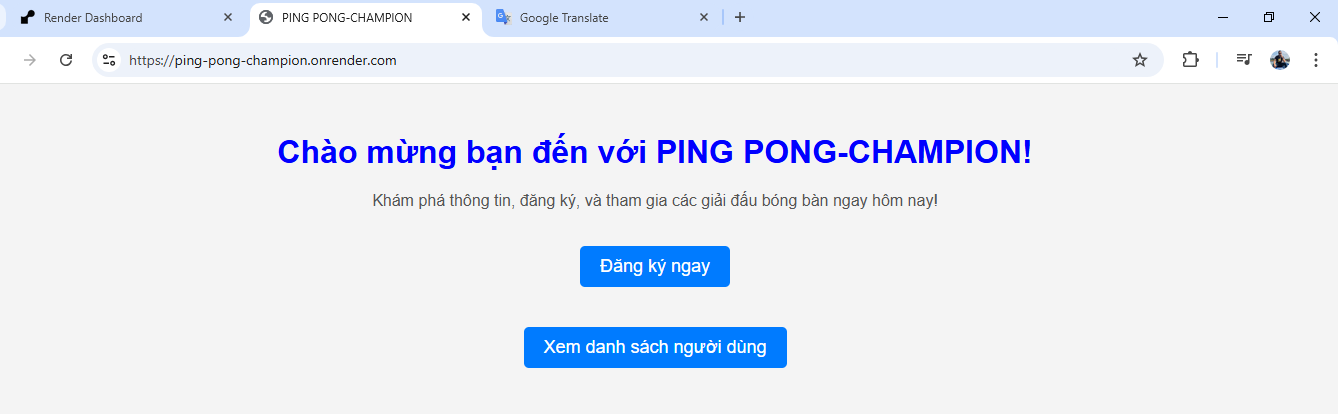
Nếu bạn đã hoàn thành bước này, chúng ta có thể tiếp tục:

1. **~~Thêm chức năng chỉnh sửa thông tin người dùng.~~**
2. **Thêm chức năng xóa người dùng.**

6. Tốt lắm, nội dung cơ bản cũng ok rồi để thử bước publish, hướng dẫn mình nhé

Để **publish (đưa website lên Internet)**, dùng **Render**, một nền tảng miễn phí, dễ sử dụng và hiện đang rất phổ biến. Dưới đây là hướng dẫn chi tiết để đưa ứng dụng của bạn lên **Render**





**HOÀN THÀNH XUẤT SẮC**

**Nội dung hướng dẫn dưới đây**

|  |
| --- |
| Dưới đây là hướng dẫn **từ đầu đến cuối** để bạn publish ứng dụng Flask của mình lên **Render**. Tôi sẽ hướng dẫn chi tiết từng bước để bạn có thể thực hiện mà không gặp lỗi. **1. Chuẩn bị mã nguồn******Cấu trúc thư mục dự án**** Hãy đảm bảo mã nguồn của bạn được sắp xếp theo cấu trúc sau:  css  Copy code  Website\_BongBan/  ├── app.py # File chính chứa ứng dụng Flask  ├── requirements.txt # Danh sách các thư viện cần cài đặt  ├── start.sh # File khởi động ứng dụng  ├── templates/ # Chứa các file HTML  │ ├── index.html  ├── static/ # Chứa các file CSS, JS, hình ảnh  │ ├── style.css ****File quan trọng******a.** app.py Đây là file Python chính của ứng dụng Flask. Ví dụ:  python  Copy code  from flask import Flask, render\_template  app = Flask(\_\_name\_\_)  @app.route("/")  def home():  return render\_template("index.html")  if \_\_name\_\_ == "\_\_main\_\_":  app.run(debug=True) **b.** requirements.txt Liệt kê các thư viện cần thiết để Render cài đặt tự động. Tạo file này bằng lệnh:  bash  Copy code  pip freeze > requirements.txt  Ví dụ nội dung:  makefile  Copy code  Flask==2.2.3  gunicorn==20.1.0 **c.** start.sh File này dùng để khởi động ứng dụng trên Render:  bash  Copy code  #!/bin/bash  gunicorn app:app --bind 0.0.0.0:$PORT **d.** templates/index.html Tạo giao diện cơ bản:  html  Copy code  <!DOCTYPE html>  <html lang="en">  <head>  <title>Website Bóng Bàn</title>  </head>  <body>  <h1>Chào mừng bạn đến với Website Bóng Bàn!</h1>  </body>  </html> **2. Đưa mã nguồn lên GitHub******a. Tạo repository trên GitHub****  1. Đăng nhập vào [GitHub](https://github.com). 2. Tạo một repository mới (ví dụ: website-bongban).  ****b. Đẩy mã nguồn từ máy lên GitHub****  1. Mở terminal và điều hướng đến thư mục dự án:   bash  Copy code  cd Website\_BongBan   1. Khởi tạo Git:   bash  Copy code  git init   1. Thêm remote repository:   bash  Copy code  git remote add origin https://github.com/<username>/website-bongban.git   * + Thay <username> bằng tên tài khoản GitHub của bạn.  1. Thêm tất cả các file vào Git:   bash  Copy code  git add .   1. Commit thay đổi:   bash  Copy code  git commit -m "Initial commit"   1. Đẩy mã nguồn lên GitHub:   bash  Copy code  git push -u origin main **3. Tạo Web Service trên Render******a. Truy cập Render****  1. Đăng nhập vào [Render.com](https://render.com). 2. Vào **Dashboard** và chọn **New > Web Service**.  ****b. Cấu hình Web Service****  1. **Kết nối với GitHub:**    * Render sẽ yêu cầu kết nối với tài khoản GitHub của bạn.    * Chọn repository website-bongban. 2. **Cấu hình dịch vụ:**    * **Name:** Đặt tên ứng dụng, ví dụ: website-bongban.    * **Environment:** Chọn Python.    * **Branch:** Chọn main.    * **Root Directory:** Để trống (nếu mã nguồn nằm trong thư mục gốc).    * **Build Command:** Render tự nhận diện Python, để trống hoặc nhập:   bash  Copy code  pip install -r requirements.txt   * + **Start Command:** Nhập:   bash  Copy code  ./start.sh   1. **Nhấn "Create Web Service"** để tạo và triển khai ứng dụng.  **4. Theo dõi quá trình triển khai**  1. Render sẽ tự động build và deploy ứng dụng của bạn. 2. Theo dõi **Logs** trong Render Dashboard để kiểm tra quá trình cài đặt. 3. Nếu không có lỗi, Render sẽ cung cấp link ứng dụng (ví dụ: https://website-bongban.onrender.com).  **5. Kiểm tra ứng dụng**  1. Mở link ứng dụng Render cung cấp trên trình duyệt. 2. Kiểm tra xem ứng dụng Flask đã chạy đúng chưa.  **6. Cập nhật mã nguồn** Nếu bạn cần chỉnh sửa ứng dụng, hãy làm như sau:   1. Thực hiện thay đổi trên máy tính. 2. Commit thay đổi:   bash  Copy code  git add .  git commit -m "Update ứng dụng"   1. Đẩy thay đổi lên GitHub:   bash  Copy code  git push origin main   1. Render sẽ tự động build và cập nhật ứng dụng.  **7. Nếu gặp lỗi**  1. **Kiểm tra log:** Vào **Logs** trên Render Dashboard để xem chi tiết lỗi. 2. **Hỗ trợ:** Gửi lỗi hoặc log để tôi hỗ trợ xử lý.   Hãy làm thử và cho tôi biết kết quả nhé! 😊 |

**NGÀY 10/01/2025**

1. Kiểm tra lại Database và xây dựng nội dung

|  |  |  |  |
| --- | --- | --- | --- |
| Hiện tại | | | |
| ID | Họ và tên | Email | Số điện thoại |
| Hệ thống tự cập nhật, là số đếm theo thứ tự đăng ký | Member nhập | Member nhập | Member nhập |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Chỉnh sửa lại | | | | | | | | |
| ID | Họ và tên | Email | Số điện thoại | Số điểm Elo hiện tại | Số trận đấu chính thức đã thực hiện | Tỉ lệ Thắng/Hòa/Thua | Trận đấu gần đây nhất | Phân hạng từ hệ thống |
| Hệ thống tự cập nhật, là số đếm theo thứ tự đăng ký | Member nhập | Member nhập | Member nhập | Hệ thống tính toán dựa trên hàm cụ thể được cập nhật bằng thủ tục xác thực. Theo đõ điểm Elo ban đầu khi tham gia hệ thống đều là 1,000 | Hệ thống tính toán dựa trên cập nhật tổng số trận đấu đã được xác thực | Hệ thống tính toán dựa trên cập nhật kết quả trận đấu đã được xác thực | Hệ thống cập nhật bằng ngày của trận đấu gần đây nhất | Tương ứng với điểm Elo, dự kiến sẽ phân ra 4 hạng (Hạng A, B, C, D) mỗi hạng gắn với một thang điểm cụ thể. Phân hạng chỉ được xếp khi trận đấu gần nhất dưới 30 ngày. |