Nhật ký Web pingpong manager

Đồng ý! Chúng ta bắt đầu với **Bước 1: Cài đặt môi trường làm việc**.

**Bước 1: Cài đặt môi trường làm việc**

1. **Cài Visual Studio Code (VSCode):**
   * [Tải VSCode tại đây](https://code.visualstudio.com/).
   * Cài đặt VSCode và mở ứng dụng.
2. **Cài đặt Node.js:**
   * [Tải Node.js tại đây](https://nodejs.org/).
   * Chọn phiên bản LTS (khuyên dùng).
   * Sau khi cài đặt, kiểm tra Node.js đã cài thành công bằng cách mở **Command Prompt (CMD)** hoặc **Terminal trong VSCode** và chạy:
   * node -v
   * npm -v

Kết quả sẽ hiển thị phiên bản Node.js và npm.

1. **Cài đặt Git (không bắt buộc):**
   * [Tải Git tại đây](https://git-scm.com/).
   * Cài đặt Git nếu bạn muốn quản lý mã nguồn.
   * Kiểm tra Git đã cài thành công bằng cách chạy:
   * git --version

**Hoàn thành bước này, bạn hãy báo để chúng ta tiến hành bước tiếp theo! 😊**

Tuyệt vời! Chúng ta tiếp tục với **Bước 2: Khởi tạo dự án**.

**Bước 2: Khởi tạo dự án**

1. **Tạo thư mục dự án:**
   * Tạo một thư mục trên máy tính, ví dụ: web-pingpong-manager.
   * Mở thư mục này bằng Visual Studio Code:
     + Vào **File > Open Folder...** và chọn thư mục bạn vừa tạo.
2. **Khởi tạo dự án Node.js:**
   * Trong VSCode, mở **Terminal** bằng cách nhấn Ctrl + ~ hoặc vào **View > Terminal**.
   * Chạy lệnh:
   * npm init -y

Lệnh này sẽ tạo một file package.json chứa thông tin cơ bản của dự án.

1. **Cài đặt các thư viện cần thiết:**
   * Trong terminal, lần lượt chạy các lệnh sau:
   * npm install express
   * npm install ejs
   * npm install body-parser
   * Các thư viện này sẽ được sử dụng để xây dựng server, hiển thị giao diện, và xử lý dữ liệu form.
2. **Kiểm tra lại thư mục:** Sau bước này, trong thư mục của bạn sẽ có các file và thư mục sau:
3. web-pingpong-manager/
4. ├── node\_modules/ (thư viện đã cài)
5. ├── package.json
6. ├── package-lock.json

Hoàn thành bước này, hãy báo để chúng ta tiếp tục với **Bước 3: Xây dựng cấu trúc dự án**. 😊

Tuyệt! Giờ chúng ta tiếp tục với **Bước 3: Xây dựng cấu trúc dự án**.

**Bước 3: Xây dựng cấu trúc dự án**

1. **Tạo các thư mục cần thiết:** Trong thư mục dự án (web-pingpong-manager), tạo các thư mục sau:
   * **views/**: Chứa các file giao diện HTML.
   * **public/**: Chứa các file tĩnh như CSS, JavaScript frontend, hình ảnh.
   * **routes/** *(Tùy chọn, bước này để dành cho mở rộng sau)*: Để quản lý logic xử lý URL nếu bạn muốn tách biệt code.
2. **Tạo file chính app.js:**
   * Tạo một file app.js trong thư mục chính của dự án.
   * Copy đoạn mã dưới đây vào file app.js:
   * const express = require('express');
   * const bodyParser = require('body-parser');
   * const app = express();
   * // Cấu hình EJS
   * app.set('view engine', 'ejs');
   * app.set('views', './views');
   * // Middleware
   * app.use(bodyParser.urlencoded({ extended: true }));
   * app.use(express.static('public'));
   * // Route chính
   * app.get('/', (req, res) => {
   * res.render('index', { title: 'Quản lý bóng bàn' });
   * });
   * // Lắng nghe server
   * const PORT = 3000;
   * app.listen(PORT, () => {
   * console.log(`Server đang chạy tại http://localhost:${PORT}`);
   * });
3. **Tạo file giao diện mẫu views/index.ejs:**
   * Trong thư mục views, tạo file index.ejs.
   * Copy đoạn mã dưới đây vào file index.ejs:
   * <!DOCTYPE html>
   * <html lang="en">
   * <head>
   * <meta charset="UTF-8">
   * <meta name="viewport" content="width=device-width, initial-scale=1.0">
   * <title><%= title %></title>
   * </head>
   * <body>
   * <h1>Chào mừng đến với hệ thống quản lý bóng bàn!</h1>
   * </body>
   * </html>
4. **Chạy thử ứng dụng:**
   * Mở terminal, chạy lệnh:
   * node app.js
   * Mở trình duyệt và truy cập http://localhost:3000.
   * Bạn sẽ thấy giao diện với dòng chữ "Chào mừng đến với hệ thống quản lý bóng bàn!".

Hoàn thành bước này, hãy báo để tiếp tục với **Bước 4: Phát triển thêm tính năng**! 😊

Dưới đây là hai file bạn cần: **index.ejs** và **app.js**, đã được cập nhật với tính năng **Đăng ký**.

**1. Code cho index.ejs**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Quản lý bóng bàn</title>

</head>

<body>

<h1>Chào mừng đến với hệ thống quản lý bóng bàn!</h1>

<h2>Đăng ký</h2>

<form action="/register" method="POST">

<label for="name">Họ và Tên:</label>

<input type="text" id="name" name="name" placeholder="Họ và Tên" required><br>

<label for="email">Email:</label>

<input type="email" id="email" name="email" placeholder="Email" required><br>

<label for="phone">Số Điện Thoại:</label>

<input type="text" id="phone" name="phone" placeholder="Số Điện Thoại" required><br>

<button type="submit">Đăng ký</button>

</form>

<h2>Xem danh sách người chơi</h2>

<a href="/players">Danh sách người chơi</a>

</body>

</html>

**2. Code cho app.js**

const express = require('express');

const bodyParser = require('body-parser');

const app = express();

// Cấu hình EJS

app.set('view engine', 'ejs');

app.set('views', './views');

// Middleware

app.use(bodyParser.urlencoded({ extended: true }));

app.use(express.static('public'));

// Lưu trữ thông tin người dùng (tạm thời dùng mảng)

const users = [];

// Route trang chính

app.get('/', (req, res) => {

res.render('index', { title: 'Quản lý bóng bàn' });

});

// Route xử lý Đăng ký

app.post('/register', (req, res) => {

const { name, email, phone } = req.body;

// Lưu thông tin người dùng vào mảng

const newUser = { name, email, phone };

users.push(newUser);

// Mô phỏng gửi email chúc mừng

console.log(`Email chúc mừng đã gửi tới: ${email}`);

// Chuyển hướng về trang chính

res.redirect('/');

});

// Route hiển thị danh sách người chơi

app.get('/players', (req, res) => {

res.render('players', { players: users });

});

// Lắng nghe server

const PORT = 3000;

app.listen(PORT, () => {

console.log(`Server đang chạy tại http://localhost:${PORT}`);

});

**Hướng dẫn chạy:**

1. **Copy nội dung trên vào file index.ejs và app.js tương ứng.**
2. Khởi động lại server bằng lệnh:
3. node app.js
4. Mở trình duyệt và truy cập http://localhost:3000 để kiểm tra tính năng **Đăng ký**.

Hãy thử và báo lại khi bạn hoàn tất nhé! 😊

Chúng ta sẽ triển khai tính năng **Đăng nhập (A)** tiếp theo, dựa trên quy trình bạn dự thảo.

### ****Bước 4.2: Thêm tính năng Đăng nhập (A)****

#### **1. Cập nhật giao diện Đăng nhập**

* **Cập nhật file index.ejs:**
  + Thêm form **Đăng nhập**:
  + <h2>Đăng nhập</h2>
  + <form action="/login" method="POST">
  + <label for="login-email">Email:</label>
  + <input type="email" id="login-email" name="email" placeholder="Email" required><br>
  + <button type="submit">Đăng nhập</button>
  + </form>

#### **2. Xử lý logic Đăng nhập trong** app.js

* **Cập nhật file app.js:**
  + Thêm logic kiểm tra email và gửi link đăng nhập:
  + app.post('/login', (req, res) => {
  + const { email } = req.body;
  + // Kiểm tra email trong danh sách người dùng
  + const user = users.find(u => u.email === email);
  + if (user) {
  + // Mô phỏng gửi email đăng nhập (in ra console)
  + console.log(`Link đăng nhập đã gửi tới: ${email}`);
  + res.send(`<h3>Link đăng nhập đã được gửi tới email: ${email}</h3><a href="/">Quay lại</a>`);
  + } else {
  + res.send(`<h3>Email không tồn tại trong hệ thống. Vui lòng đăng ký.</h3><a href="/">Quay lại</a>`);
  + }
  + });

### ****Hướng dẫn kiểm tra****

1. **Khởi động lại server** bằng lệnh:
2. node app.js
3. **Truy cập trang chính** tại http://localhost:3000.
4. Nhập email vào form Đăng nhập:
   * Nếu email tồn tại trong danh sách, bạn sẽ thấy thông báo "Link đăng nhập đã được gửi".
   * Nếu không tồn tại, thông báo sẽ yêu cầu đăng ký.

Hoàn thành bước này, hãy báo để chúng ta tiếp tục tính năng khác, như **quản lý dữ liệu người dùng sau đăng nhập** hoặc **xem bảng xếp hạng**. 😊

Lỗi này xảy ra vì bạn chưa tạo file players.ejs trong thư mục views. Hệ thống không tìm thấy file giao diện để hiển thị danh sách người chơi.

**Cách khắc phục**

1. **Tạo file players.ejs:**
   * Trong thư mục views, tạo file mới tên là players.ejs.
2. **Thêm nội dung sau vào players.ejs:**
3. <!DOCTYPE html>
4. <html lang="en">
5. <head>
6. <meta charset="UTF-8">
7. <meta name="viewport" content="width=device-width, initial-scale=1.0">
8. <title>Danh sách người chơi</title>
9. </head>
10. <body>
11. <h1>Danh sách người chơi</h1>
12. <ul>
13. <% players.forEach(player => { %>
14. <li><%= player.name %> - <%= player.email %> - <%= player.phone %></li>
15. <% }) %>
16. </ul>
17. <a href="/">Quay lại</a>
18. </body>
19. </html>
20. **Lưu file và khởi động lại server:**
    * Sau khi lưu, khởi động lại server bằng lệnh:
    * node app.js
21. **Kiểm tra lại:**
    * Truy cập lại đường dẫn http://localhost:3000/players để kiểm tra.

Hãy thử và cho tôi biết nếu bạn cần thêm hỗ trợ! 😊

