TRƯỜNG ĐẠI HỌC KỸ THUẬT CÔNG NGHIỆP

KHOA ĐIỆN TỬ

Bộ môn: Công nghệ thông tin

BÀI TẬP LỚN LẬP TRÌNH PYTHON

Tên đề tài: WEB GAME CARO

Sinh viên thực hiện: **Phạm Tuấn Dương**

MSSV: **K205480106044**

Lớp: **56KMT.01**

Thái Nguyên – 2024

TRƯỜNG ĐH KỸ THUẬT CÔNG NGHIỆP
KHOA ĐIỆN TỬ
BỘ MÔN CÔNG NGHỆ THÔNG TIN

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập - Tự do - Hạnh phúc
----o0o----

Ngày hoàn thành, ngày 26 tháng 05 năm 2024

BÀI TẬP LỚN LẬP TRÌNH PYTHON

BỘ MÔN CÔNG NGHỆ THÔNG TIN

Sinh viên: Đỗ Văn Hiếu MSSV: K205480106044

Ngành: Kỹ thuật máy tính Lớp học phần: 56KMT.01

Tên đề tài: **Website chơi game caro**

I. Mục tiêu

- Xây dựng website cho phép nhiều người chơi cùng truy cập
- Trong web sẽ tạo được các phòng chơi để hai người chơi thi đấu
- Phòng chơi sẽ được người chơi tùy ý tạo
- Người chơi có thể rời phòng, vào phòng
- Xây dựng một ứng dụng chơi game vui vẻ và công bằng

IV. Thực hiện

- Công nghệ sử dụng: flask, python-socketio
- Lưu trữ dữ liệu trên bộ nhớ server

GIÁO VIÊN HƯỚNG DẪN

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Thái Nguyên, ngày….tháng…..năm 20....

GIÁO VIÊN HƯỚNG DẪN

(Ký ghi rõ họ tên)

LỜI NÓI ĐẦU

Trong thời đại công nghệ số phát triển mạnh mẽ như hiện nay, các ứng dụng web ngày càng trở nên phổ biến và đóng vai trò quan trọng trong nhiều lĩnh vực, từ giáo dục, giải trí đến kinh doanh. Việc phát triển các trò chơi trực tuyến không chỉ mang lại sự giải trí mà còn giúp rèn luyện tư duy, khả năng lập luận và kỹ năng giải quyết vấn đề.

Đề tài "Phát triển ứng dụng web chơi cờ caro" nhằm xây dựng một trò chơi cờ caro có thể chơi trực tuyến qua web. Đây là một bài tập lớn trong khuôn khổ môn học phát triển phần mềm, giúp sinh viên ứng dụng kiến thức đã học vào thực tế, từ đó hiểu rõ hơn về quy trình phát triển một ứng dụng web hoàn chỉnh. Trò chơi cờ caro không chỉ quen thuộc với mọi lứa tuổi mà còn có tính cạnh tranh và chiến thuật cao.

Trong bài báo cáo này, chúng tôi sẽ trình bày chi tiết quá trình phân tích, thiết kế, và triển khai ứng dụng web chơi cờ caro. Bao gồm việc xác định yêu cầu hệ thống, thiết kế giao diện người dùng, xây dựng cơ sở dữ liệu, lập trình logic trò chơi, và kiểm thử ứng dụng. Qua đó, bài tập không chỉ giúp chúng tôi củng cố kiến thức mà còn phát triển kỹ năng làm việc nhóm, giải quyết vấn đề và quản lý dự án.

Hy vọng rằng qua bài báo cáo này, tôi có thể trình bày rõ ràng và chi tiết về quá trình phát triển ứng dụng web chơi cờ caro, đồng thời chia sẻ những kinh nghiệm và bài học quý giá đã thu được trong suốt quá trình thực hiện đề tài

Do kiến thức còn hạn hẹp nên báo cáo không tránh khỏi sai sót, rất mong sự đóng góp ý kiến của thầy cô và các bạn để báo cáo được hoàn chỉnh hơn.

CHƯƠNG I: GIỚI THIỆU ĐỀ TÀI

1.1. Giới thiệu

Ứng dụng web chơi cờ caro sẽ cung cấp một nền tảng trực tuyến cho phép người dùng có thể chơi trò chơi cờ caro một cách tiện lợi và hiệu quả. Ứng dụng này không chỉ đáp ứng nhu cầu giải trí mà còn thúc đẩy sự giao lưu, kết nối giữa các người chơi thông qua mạng internet. Đề tài này sẽ tập trung vào việc phát triển ứng dụng với các tiêu chí cụ thể như sau:

1.2. Mục tiêu

- Phát triển ứng dụng trên nền website: Tạo ra một ứng dụng web có thể chạy trên mọi trình duyệt, giúp người dùng dễ dàng truy cập từ bất kỳ thiết bị nào có kết nối internet.
- Cho phép nhiều người chơi cùng truy cập: Ứng dụng phải hỗ trợ nhiều người chơi cùng tham gia, đảm bảo hiệu suất và trải nghiệm người dùng tốt.
- Người chơi có thể tự tạo, lựa chọn phòng, đối thủ: Cung cấp chức năng cho phép người dùng tạo phòng chơi riêng, lựa chọn hoặc mời đối thủ tham gia trận đấu.
- Đảm bảo sự công bằng trong trò chơi: Sử dụng các thuật toán và biện pháp kỹ thuật để đảm bảo tính công bằng, không để người chơi gian lận hoặc thao túng kết quả trận đấu.

1.3. Nội dung thực hiện

Công nghệ sử dụng:

- Frontend: HTML, CSS, JavaScript.
- Backend: Python - flask

Cho phép nhiều người chơi cùng truy cập:

- Sử dụng WebSocket:Sử dụng WebSocket để xử lý kết nối thời gian thực, đảm bảo người chơi có thể nhận và gửi dữ liệu nhanh chóng.

Quản lý phiên chơi:

- Triển khai hệ thống quản lý phiên chơi để theo dõi và duy trì trạng thái của các trận đấu.

Chức năng tạo phòng chơi:

- Người dùng có thể tạo phòng chơi riêng, thiết lập các tùy chọn như tên phòng, số người chơi tối đa.

Chức năng lựa chọn phòng chơi:

- Hiển thị danh sách các phòng chơi đang có sẵn để người dùng tham gia.
- Chức năng mời đối thủ:
- Cho phép người chơi mời bạn bè hoặc người chơi khác thông qua liên kết hoặc danh sách bạn bè.

Đảm bảo sự công bằng trong trò chơi

- Thuật toán kiểm tra nước đi hợp lệ:
- Phát triển và triển khai thuật toán kiểm tra nước đi để đảm bảo mọi nước đi đều hợp lệ theo quy tắc của cờ caro.

CHƯƠNG II: THỰC HIỆN ĐỀ BÀI

2.1 Thực hiện cấu hình và lập trình

Cấu hình môi trường và cài đặt các thư viện cần thiết

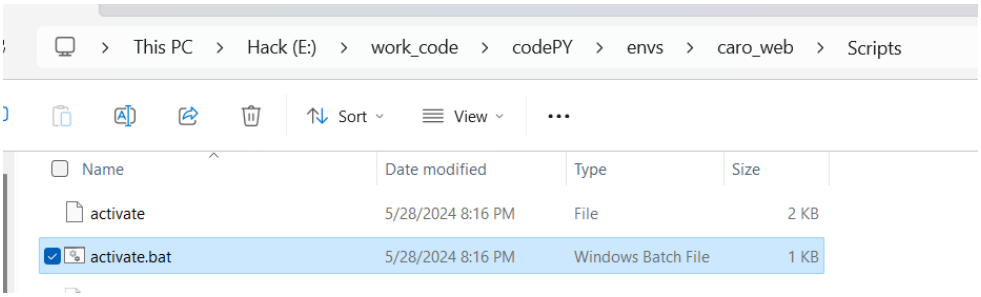
Với bài tập này em sử dụng python làm web server vậy nên yêu cầu máy tính cần có python:

```
C:\Users\DkidWin11>python --version
Python 3.11.5
```

Tiếp theo là cài đặt môi trường ảo cho project, tới forder muốn đặt môi trường ảo và chạy lệnh:

```
E:\work_code\codePY\envs>python -m venv caro_web
```

Khi đó môi trường có tên là caro_web sẽ được tạo, đi tới forder và tìm tới Scripts\activate.bat



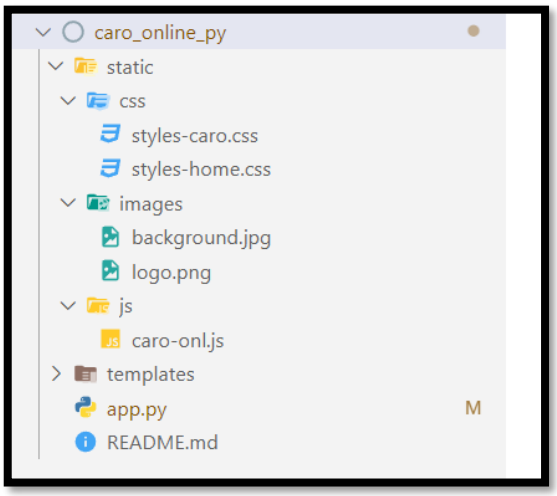
Tiếp theo là tạo foder chứa dự án và kích hoạt môi trường ảo bằng cách call file activate.bat:

```
E:\work_code\codePY\caro_online_py>call E:\work_code\codePY\envs\caro_web\Scripts\activate.bat

(caro_web) E:\work_code\codePY\caro_online_py>
```

Sau đó cài đặt các thư viện cần sử dụng qua pip bao gồm: flask, flask_socketio qua công cụ pip

Tiếp theo là mở công cụ lập trình (vs code), và bắt đầu dự án. Bài tập sẽ sử dụng flask với server tên là app.py và cơ chế "render_template" để trả về trang trang html hiển thị giao diện, thư mục static phục vụ ảnh, file css, file js. Vậy nên cấu trúc thư mục dự án:



Cấu hình server khai báo các thư viện, đối tượng của socket và trả về hai đường dẫn trang home và trang chơi game:

app.py M ×

caro_online_py > app.py

```python
# -*- coding: utf-8 -*-
from flask import Flask, render_template, request
from flask_socketio import SocketIO, emit, join_room, leave_room, disconnect
from flask import jsonify

app = Flask(__name__)
socketio = SocketIO(app)
rooms = {}

# Trang home
@app.route('/')
def index():
    return render_template('index.html')

# Trang chơi caro online 2 người
@app.route('/caro-onl')
def onl():
    return render_template('caro-onl.html')

```

Sau khi đã cấu hình song project như trên và trả về trang html, liên kết hình ảnh, css, javascript thì phần còn lại là giao tiếp giữa server và ajax từ js client gửi về. Tổng quan về quá trình đó như sau:

WebSocket là một giao thức mạng tiên tiến được thiết kế để tạo ra kết nối hai chiều (full-duplex) giữa client và server thông qua một kết nối TCP duy nhất. Nó thường được sử dụng trong các ứng dụng thời gian thực như chat, game trực tuyến, và các ứng dụng web thời gian thực khác.

- **Khởi tạo kết nối**: Client gửi một yêu cầu handshake đặc biệt đến server để bắt đầu kết nối WebSocket. Nếu server chấp nhận, kết nối WebSocket được thiết lập.
- **Giao tiếp hai chiều**: Một khi kết nối WebSocket được thiết lập, cả client và server có thể gửi tin nhắn cho nhau bất cứ lúc nào. Không cần phải tạo mới kết nối cho mỗi yêu cầu.
- **Đóng kết nối**: Khi không còn cần thiết, cả client và server có thể đóng kết nối WebSocket.

Cụ thể trong bài tập:

Tạo Phòng Chơi (/create route)

Phương thức: POST

Mô tả: Tạo một phòng chơi mới với mã phòng được cung cấp bởi người dùng.

Quá trình:

- Người dùng gửi yêu cầu POST với room_code để tạo phòng chơi.
- Server kiểm tra xem phòng chơi với mã đó đã tồn tại chưa.
- Nếu phòng chơi đã tồn tại, server trả về lỗi 400.
- Nếu phòng chưa tồn tại, server tạo một phòng mới và trả về thông báo thành công.

```python
global_player_count = 0
# Tạo phòng chơi để join vào phòng
@app.route('/create', methods=['POST'])
def create():
    room_code = request.form.get('room_code')
    if room_code in rooms:
        return 'Room already exists', 400
    rooms[room_code] = {'players': []}
    player_count = 0
    return 'Room created successfully'
```

Tham Gia Phòng Chơi (join event)

Phương thức: WebSocket

Mô tả: Cho phép người chơi tham gia vào phòng chơi.

Quá trình:

- Người dùng gửi sự kiện join với room_code.
- Server kiểm tra điều kiện của phòng (tồn tại và chưa đầy).
- Người chơi được gán biểu tượng ('X' hoặc 'O') và tham gia phòng.
- Server thông báo đến tất cả người chơi trong phòng về việc người chơi mới tham gia.

```python
# Join phòng chơi (check các điều kiện)
@socketio.on('join')
def on_join(data):
    global global_player_count
    room_code = data['room_code']

    if room_code not in rooms or len(rooms[room_code]['players']) >= 2:
      return 'Room is full', 400

    symbol = 'X' if global_player_count == 1 else 'O'
    player = {'id': request.sid, 'symbol': symbol}
    rooms[room_code]['players'].append(player)

    join_room(room_code)

    # Gửi thông tin của cả hai người chơi về client
    emit('join', {'room_code': room_code, 'player': player, 'request_sid': request.sid, 'players': rooms[roo

    global_player_count += 1
```

Xử lý Mất Kết Nối (disconnect event)

Phương thức: WebSocket

Mô tả: Xử lý khi một người chơi bị mất kết nối.

Quá trình:

- Server phát hiện sự kiện mất kết nối.
- Cập nhật danh sách người chơi trong phòng.
- Thông báo đến các người chơi còn lại trong phòng về việc mất kết nối của đối thủ.

```python
# Xử lý sự kiện disconnect
@socketio.on('disconnect')
def handle_disconnect():
    global global_player_count

    # Lấy thông tin người chơi mất mạng
    disconnected_player = None
    room_code = None

    for room_code, room_data in rooms.items():
        for player in room_data['players']:
            if player['id'] == request.sid:
                disconnected_player = player
                room_data['players'].remove(player)
                break

    # Nếu có người chơi mất mạng, cập nhật thông tin và gửi sự kiện 'leave' đến các client trong phòng
    if disconnected_player:
        leave_room(room_code)
        emit('leave', {
            'room_code': room_code,
            'player': disconnected_player,
            'request_sid': request.sid,
            'players': room_data['players']
        }, room=room_code)

        global_player_count -= 1
```

Rời Phòng Chơi (leave event)

Phương thức: WebSocket

Mô tả: Cho phép người chơi rời khỏi phòng chơi.

Quá trình:

- Người dùng gửi sự kiện leave với room_code.
- Server cập nhật danh sách người chơi trong phòng.
- Thông báo đến các người chơi còn lại trong phòng về việc người chơi rời khỏi phòng.

```python
# Rời phòng chơi
@socketio.on('leave')
def on_leave(data):
    global global_player_count
    room_code = data['room_code']

    if room_code not in rooms:
        return 'Room does not exist', 400

    player = next((player for player in rooms[room_code]['players'] if player['id'] == request.sid), None)
    if player is None:
        return 'Player not in room', 400

    rooms[room_code]['players'].remove(player)
    leave_room(room_code)

    # Gửi thông tin về việc rời phòng về client
    emit('leave', {'room_code': room_code, 'player': player, 'request_sid': request.sid, 'players': rooms[roo

    global_player_count -= 1
```

Cập Nhật Bước Đi của Người Chơi (playerMove event)

Phương thức: WebSocket

Mô tả: Xử lý và truyền thông tin về các bước đi của người chơi.

Quá trình:

- Người dùng gửi sự kiện playerMove với dữ liệu bước đi.
- Server kiểm tra tính hợp lệ và thông báo bước đi đến tất cả người chơi trong phòng.
- Xử lý và thông báo bước đi của đối thủ nếu có.

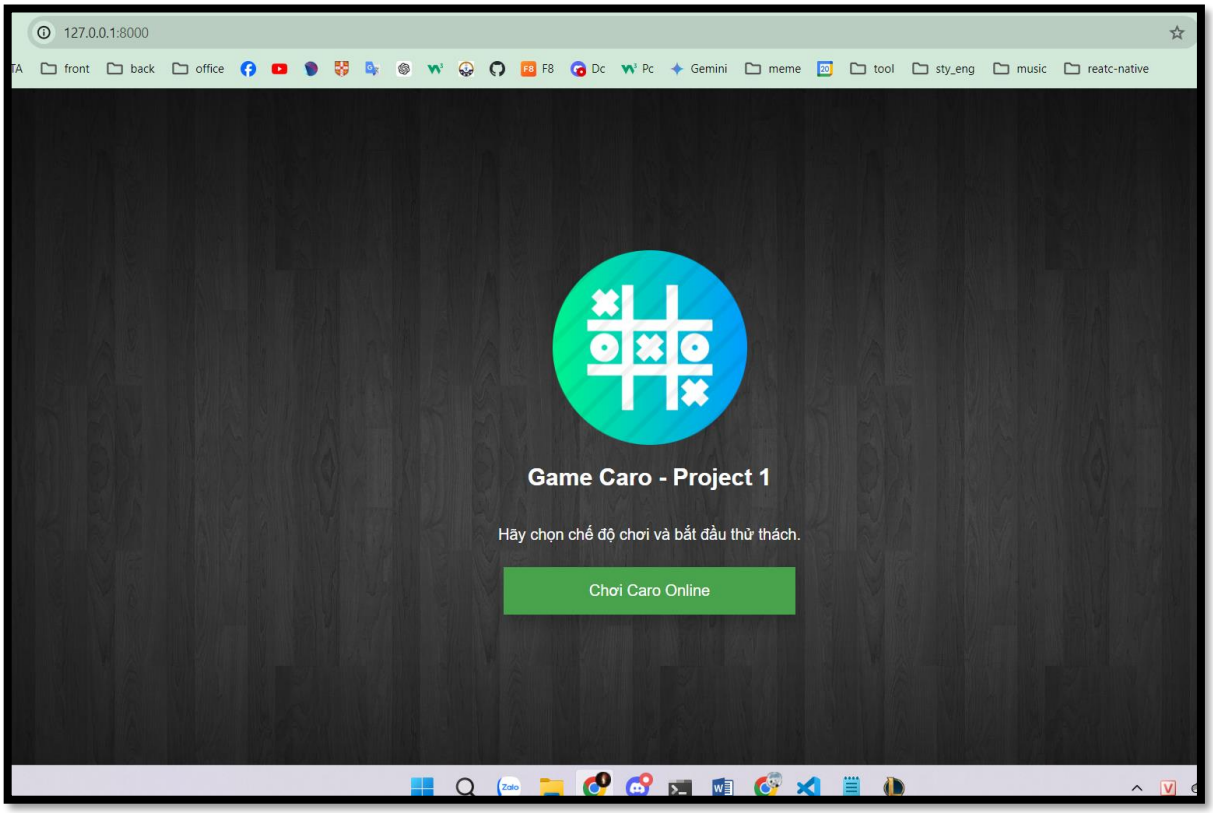
```python
# Cập nhật các bước đánh của người chơi
@socketio.on('playerMove')
def on_move(data):
  room_code = data['room_code']
  player = next((player for player in rooms.get(room_code, {}).get('players', []) if player['id'] == request.s
  if player:
      # Gửi bước đánh của người chơi hiện tại đến tất cả người chơi trong phòng
      emit('playerMove', data, room=room_code, include_self=True)
      # Xác định đối thủ của người chơi hiện tại
      opponent = next((p for p in rooms[room_code]['players'] if p['id'] != request.sid), None)
      if opponent and player['symbol'] != opponent['symbol']:
        # Gửi bước đánh của đối thủ đến toàn bộ phòng, tránh gửi lại cho chính người chơi hiện tại
        emit('opponentMove', data, room=room_code, skip_sid=request.sid)
      else:
          # Handle nếu không phải lượt của người chơi hoặc không xác định được đối thủ
          emit('invalidMove', {'message': 'It is not your turn or opponent not found'}, room=request.sid)
  else:
      # Handle nếu không phải người chơi trong phòng
      emit('invalidMove', {'message': 'You are not in this room'}, room=request.sid)
```

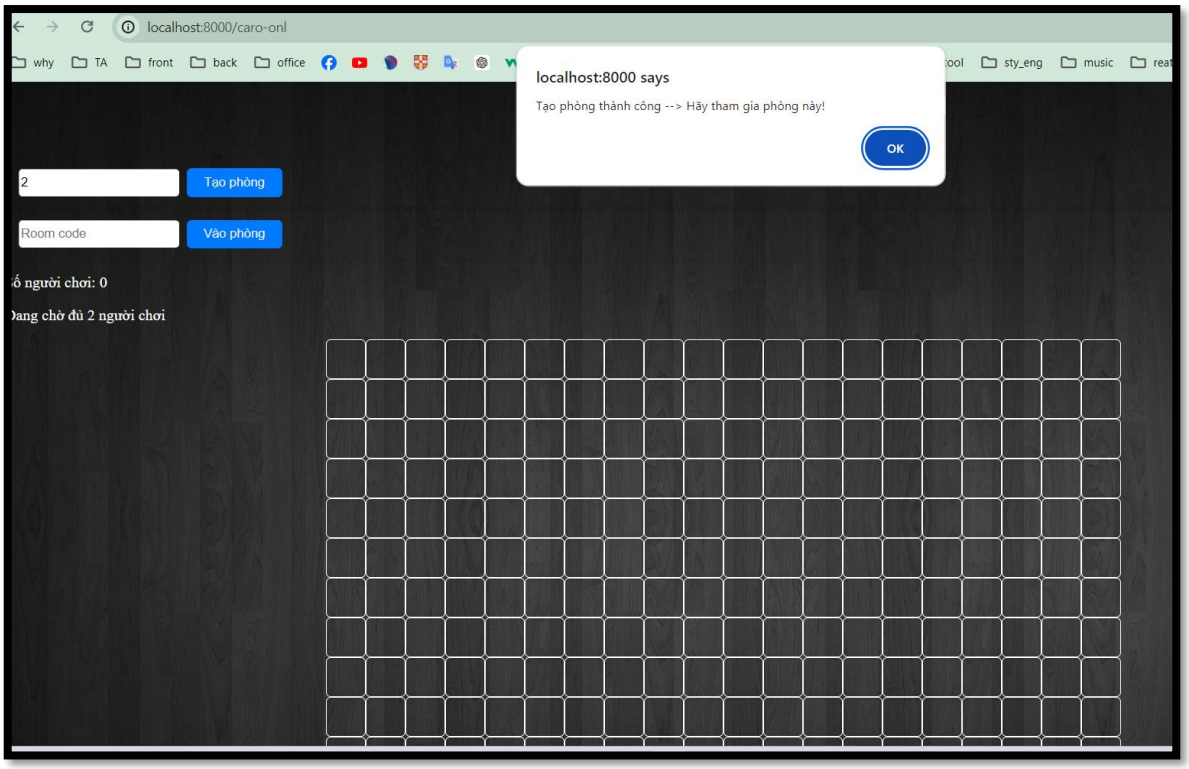
Trên đây là cách server nhận yêu cầu và xử lya khi có dữ liệu từ ajax từ phía client gửi về, Ví dụ về client khi cần tạo phòng mới:

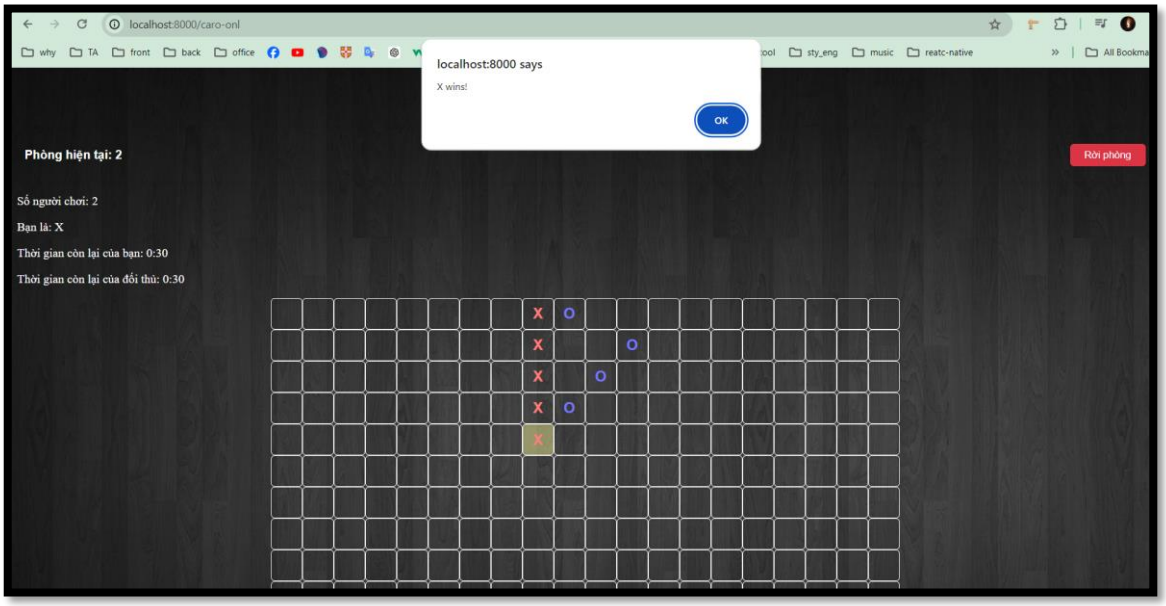
```
// Tạo mã phòng để bắt đầu chơi
document.getElementById('create-room-form').addEventListener('submit', function(e) {
    e.preventDefault();
    let roomCode = document.getElementById('create-room-code').value;
    fetch('/create', {
        method: 'POST',
        body: new URLSearchParams({
            'room_code': roomCode,
        })
    }).then(response => {
        if (!response.ok) {
            alert('Tạo phòng không thành công! (Phòng đã tồn tại)');
        } else {
            alert('Tạo phòng thành công --> Hãy tham gia phòng này!');
        }
    });
});
```

2.2 Kết quả

Kết quả đạt được đã tạo một server cho phép nhiều người chơi online và thi đấu game caro:




localhost:8000/caro-onl
why
TA
front
back
office
sty_eng
music
localhost:8000 says
Tạo phòng thành công --> Hãy tham gia phòng này!
OK
2
Tạo phòng
Room code
Vào phòng
ố người chơi: 0
)ang chờ đủ 2 người chơi


localhost:8000/caro-onl
why
TA
front
back
office
sty_eng
music
reatc-native
localhost:8000 says
X wins!
OK
Phòng hiện tại: 2
Rời phòng
Số người chơi: 2
Bạn là: X
Thời gian còn lại của bạn: 0:30
Thời gian còn lại của đối thủ: 0:30

CHƯƠNG III: KẾT LUẬN

Sau khi hoàn thành đề tài "Phát triển ứng dụng web chơi cờ caro", chúng ta đã đạt được những mục tiêu cơ bản đề ra và học được nhiều điều quan trọng trong quá trình phát triển ứng dụng web. Ứng dụng đã được xây dựng với các tính năng cơ bản như tạo phòng chơi, tham gia phòng chơi, cập nhật và hiển thị các nước đi của người chơi, cũng như xử lý các tình huống mất kết nối và rời phòng.

Xây dựng ứng dụng trên nền website

Chúng ta đã xây dựng thành công một ứng dụng web chơi cờ caro, giúp người dùng dễ dàng truy cập từ mọi thiết bị có kết nối internet. Giao diện người dùng (UI) được thiết kế thân thiện, trực quan, đảm bảo trải nghiệm người dùng tốt.

Hỗ trợ nhiều người chơi cùng tham gia

Ứng dụng hỗ trợ nhiều người chơi cùng tham gia và chơi cờ caro trực tuyến. Chúng ta đã triển khai thành công việc sử dụng WebSocket để xử lý kết nối thời gian thực giữa các người chơi, đảm bảo mọi người có thể nhận và gửi dữ liệu một cách nhanh chóng và hiệu quả.

Tạo và quản lý phòng chơi

Ứng dụng cho phép người dùng tự tạo và lựa chọn phòng chơi, cũng như mời đối thủ tham gia trận đấu. Chức năng này giúp tăng tính tương tác và sự linh hoạt cho người chơi khi họ có thể dễ dàng tạo các trận đấu riêng tư hoặc công khai.

Đảm bảo công bằng trong trò chơi

Chúng ta đã phát triển và triển khai các thuật toán để kiểm tra tính hợp lệ của các nước đi, đảm bảo mọi nước đi đều tuân theo quy tắc của trò chơi cờ caro.

Đồng thời, các biện pháp bảo mật đã được áp dụng để ngăn chặn các hành vi gian lận.

Các Bài Học Kinh Nghiệm

Sử dụng websocket

Việc sử dụng WebSocket đã giúp chúng ta hiểu rõ hơn về cách thức giao tiếp thời gian thực giữa client và server. Chúng ta đã học được cách thiết lập và quản lý kết nối WebSocket, xử lý các sự kiện và truyền tải dữ liệu một cách hiệu quả.

Thiết kế và triển khai Backend

Chúng ta đã học cách thiết kế và triển khai backend sử dụng các framework phổ biến như Flask hoặc Django, kết hợp với các thư viện hỗ trợ WebSocket như Socket.IO. Việc quản lý phiên chơi, lưu trữ dữ liệu và xử lý logic trò chơi đã giúp chúng ta nắm vững hơn về lập trình backend.

Đề tài "Phát triển ứng dụng web chơi cờ caro" đã giúp chúng ta đạt được nhiều kết quả tích cực và học hỏi được nhiều kiến thức quan trọng trong lĩnh vực phát triển ứng dụng web. Những kinh nghiệm này sẽ là nền tảng vững chắc để chúng ta tiếp tục phát triển và mở rộng các dự án trong tương lai, đáp ứng tốt hơn nhu cầu của người dùng.