**ĐẠI HỌC BÁCH KHOA HÀ NỘI**

**TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**

---------\*\*\*---------



**PROJECT I**

**Đề tài: Xây dựng game Sudoku bằng ngôn ngữ JavaScript**

**PHẠM VĂN LINH**

linh.pv194094@sis.hust.edu.vn

|  |  |
| --- | --- |
| **Giảng viên hướng dẫn:** | TS. Nguyễn Tuấn Dũng |
|  |  |
| **Trường:** | Công nghệ Thông tin và Truyền thông  Đại học Bách Khoa Hà Nội |

Mục lục

[**CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI** 3](#_Toc115116024)

[**CHƯƠNG 2. CƠ SỞ LÝ THUYẾT** 4](#_Toc115116025)

[**2.1. Backend** 4](#_Toc115116026)

[2.1.1. ExpressJS 4](#_Toc115116027)

[2.1.2. RESTful API 4](#_Toc115116028)

[**2.2. Frontend** 5](#_Toc115116029)

[**CHƯƠNG 3. ỨNG DỤNG VÀO PROJECT** 7](#_Toc115116030)

[**3.1. Backend** 7](#_Toc115116031)

[3.1.1. API 7](#_Toc115116032)

[3.1.1. Thuật toán tạo câu đố 7](#_Toc115116033)

[3.1.2. Thuật toán kiểm tra lời giải 11](#_Toc115116034)

[**3.2. Frontend** 12](#_Toc115116035)

[**CHƯƠNG 4. KẾT LUẬN** 15](#_Toc115116036)

[**TÀI LIỆU THAM KHẢO** 16](#_Toc115116037)

# **CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI**

Sudoku là một tựa game câu đố với cách giải là sắp xếp các ô số theo logic. Mục tiêu của trò chơi là sắp xếp các số vào một lưới 9 x 9 sao cho mỗi cột, mỗi hàng, và mỗi lưới nhỏ 3 x 3 đều chứa các số từ 1 đến 9. Câu đố khi được sinh ra đã được điền sẵn một phần, nhiệm vụ của người chơi là điền số vào các vị trí còn lại sao cho thỏa mãn mục tiêu của trò chơi. Đề tài của em là dựa vào trò chơi trên để xây dựng game Sudoku bằng ngôn ngữ JavaScript.

Mục tiêu của đề tài này là tìm hiểu được cách xây dựng thuật toán sinh câu đố cho trò chơi Sudoku, và từ đó hiển thị câu đố để người chơi giải. Sau khi người chơi giải hết câu đố có thể kiểm tra xem câu trả lời của mình có chính xác hay không, nếu chính xác thì lưu điểm của người chơi và hiển thị danh sách những người chơi giải nhanh nhất theo các mức độ.

Để thực hiện đề tài này ngôn ngữ em lựa chọn là JavaScript để xây dựng web game Sudoku. Về fontend, em sử dụng frameword ReactJS, backend em sử dụng framework ExpressJS để sinh đề bài mới cũng như kiểm tra đáp án và lưu điểm số của người chơi. Để lưu được điểm số của người chơi em sử dụng MongoDB để lưu trữ Database.

# **CHƯƠNG 2. CƠ SỞ LÝ THUYẾT**

Trong phần này, em sẽ nêu ra những kiến thức mà em đã học được khi thực hiện Project này và em sẽ chia thành 2 phần chính là Backend và Frontend.

## **2.1. Backend**

Với backend, dựa vào những kiến thức em đã biết và tìm hiểu thêm em đã học được cách xây dựng API với ExpressJS và các nguyên tắc của RESTful API.

### 2.1.1. ExpressJS

Expressjs là một framework được xây dựng trên nền tảng của Nodejs. Nó cung cấp các tính năng mạnh mẽ để phát triển web hoặc mobile. Expressjs hỗ trợ các method HTTP và midleware tạo ra API vô cùng mạnh mẽ và dễ sử dụng.

Một số tính năng của ExpressJS:

* Thiết lập các lớp trung gian để trả về các HTTP request.
* Define router cho phép sử dụng với các hành động khác nhau dựa trên phương thức HTTP và URL.
* Cho phép trả về các trang HTML dựa vào các tham số.

### 2.1.2. RESTful API

**RESTful là gì?**

* RESful API không phải là một loại công nghệ.
* RESful API là một tiêu chuẩn dùng trong việc thiết kế API cho các ứng dụng Web (Web Service).

**RESTful hoạt động như thế nào?**

* Hoạt động chủ yếu dựa vào giao thức HTTP.
  + GET: trả về resource
  + POST: Tạo mới một Resource
  + PUT: Cập nhật một Resource
  + DELETE: Xóa một Resource
* RESTful API không sử dụng Session và Cookie. Nó sử dụng một access\_token với mỗi request.

Các nguyên tắc thiết kế trong RESTful API

1. Làm mọi thứ thật đơn giản theo nguyên tắc KISS (Keep It Simple Stupid).
2. Sử dụng danh từ, **không** dùng động từ.
3. Sử dụng phương thức phù hợp:

* GET - Để lấy tài nguyên, không thay đổi trạng thái tài nguyên.
* POST - Để tạo tài nguyên.
* PUT/PATCH - Để cập nhật tài nguyên hoặc một phần tài nguyên.
* DELETE - Để xóa tài nguyên.

1. Sử dụng danh từ số nhiều
2. **Sử dụng tham số** (parameter): Giúp cho base URL đơn giản và có thể mở rộng để phục vụ nhiều mục đích khác
3. Sử dụng mã HTTP Status Code:
   * **200** OK - Thành công.
   * **201** CREATED - Tạo dữ liệu thành công (POST method).
   * **400** BAD REQUEST - Input của Client gửi không hợp lệ.
   * **401** UNAUTHORIZED - Hệ thống chưa được ủy quyền.
   * **403** FORBIDDEN - Máy chủ từ chối yêu cầu (ngay cả khi Client đã được ủy quyền).
   * **404** NOT FOUND - Máy chủ không tìm thấy bất kỳ thứ gì với request-URL này.
   * **500** INTERNAL SERVER ERROR - Có lỗi xảy ra phía máy chủ.
4. Phiên bản (Version)
   * Rất quan trọng, hỗ trợ việc tương thích ngược trong quá trình nâng cấp sau này. Đặt trước tên tài nguyên.
   * Áp dụng nguyên tắc KISS và không chứa ký tự đặc biệt.

## **2.2. Frontend**

Về fontend em cũng đã học được cách sử dụng ReactJS để tạo một trang web đơn giản. Dưới đây là một số kiến thức em học được xoay quanh ReactJS như component, props và state cũng như Redux.

**ReactJS là gì?**

* React là một thư viện UI phát triển tại Facebook để hỗ trợ việc xây dựng những thành phần (components) UI có tính tương tác cao, có trạng thái và có thể sử dụng lại được.

**Component trong ReactJS**

* React được xây dựng xung quanh các component, chứ không dùng template như các framework khác. Trong React, chúng ta xây dựng trang web sử dụng những thành phần (component) nhỏ. Chúng ta có thể tái sử dụng một component ở nhiều nơi, với các trạng thái hoặc các thuộc tính khác nhau, trong một component lại có thể chứa thành phần khác. Mỗi component trong React có một trạng thái riêng, có thể thay đổi, và React sẽ thực hiện cập nhật component dựa trên những thay đổi của trạng thái. Mọi thứ React đều là component. Chúng giúp bảo trì mã code khi làm việc với các dự án lớn. Một react component đơn giản chỉ cần một method render. Có rất nhiều methods khả dụng khác, nhưng render là method chủ đạo.

**Props và State**

* Props: giúp các component tương tác với nhau, component nhận input gọi là props, và trả thuộc tính mô tả những gì component con sẽ render. Prop là bất biến.
* State: thể hiện trạng thái của ứng dụng, khi state thay đồi thì component đồng thời render lại để cập nhật UI.

**Redux là gì?**

* Nói một cách đơn giản, Redux là một công cụ quản lý trạng thái. Mặc dù nó được sử dụng chủ yếu với React, nhưng nó có thể được sử dụng với bất kỳ khung hoặc thư viện JavaScript nào khác.
* Với Redux, trạng thái ứng dụng của bạn được giữ trong một "store" và mỗi thành phần có thể truy cập bất kỳ trạng thái nào mà nó cần từ "store"này. Sâu hơn một chút để xem tại sao bạn có thể cần một công cụ quản lý trạng thái.

# **CHƯƠNG 3. ỨNG DỤNG VÀO PROJECT**

## **3.1. Backend**

### 3.1.1. API

Các API em sử dụng trong Project này:

- GET: /api/v1/missions/{level} : lấy câu đố mới theo level, level là các mức độ của trò chơi – 1: Dễ, 2: Trung bình, 3: Khó.

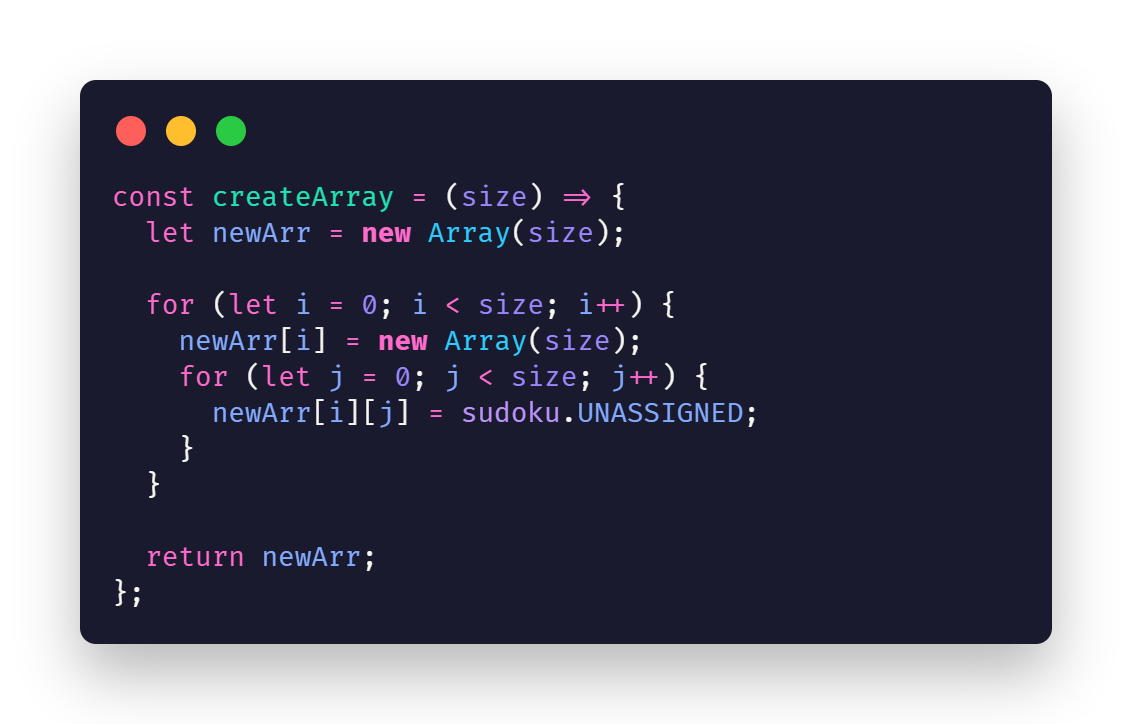
- POST: /api/v1/missions: gửi câu trả lời để kiểm tra và lưu điểm số nếu đúng.

- GET: /api/v1/scores/{level}: lấy danh sách điểm cao theo level.

### 3.1.1. Thuật toán tạo câu đố

**Bước 1:** Sinh ra một mảng 9 x 9 với các giá trị ban đầu bằng 0.

* Sinh ra mảng 2 chiều 9 x 9 và gán giá trị mặc định cho mỗi ô là 0.



**Bước 2:** Điền đầy đủ các số theo đúng nguyên tắc của game vào mảng 9 x 9.

2.1.Tìm vị trí ô có giá trị bằng 0, nếu không còn ô nào có giá trị bằng 0 thì return true.

A screenshot of a computer

Description automatically generated with medium confidence

2.2. Tạo mảng một chiều gồm các số từ 1 🡪 9 vào đảo lộn vị trí các số

Text

Description automatically generated

2.3. Duyệt mảng vừa trộn và kiểm tra số đang duyệt có phù hợp với vị trí trống hay không

2.4. Nếu phù hợp thì kiểm tra các ô đã full hay chưa, nếu đã full thì return true và dừng vòng lặp. Nếu chưa full thì đệ quy lại bước 2.1.

Text

Description automatically generated

**Bước 3:** Loại bỏ một số ô tùy theo mức độ để sinh ra câu đố hoàn chỉnh.

* Tùy vào mức độ trò chơi số ô trống sẽ khác nhau và các ô trống sẽ điền là 0.

A screenshot of a computer

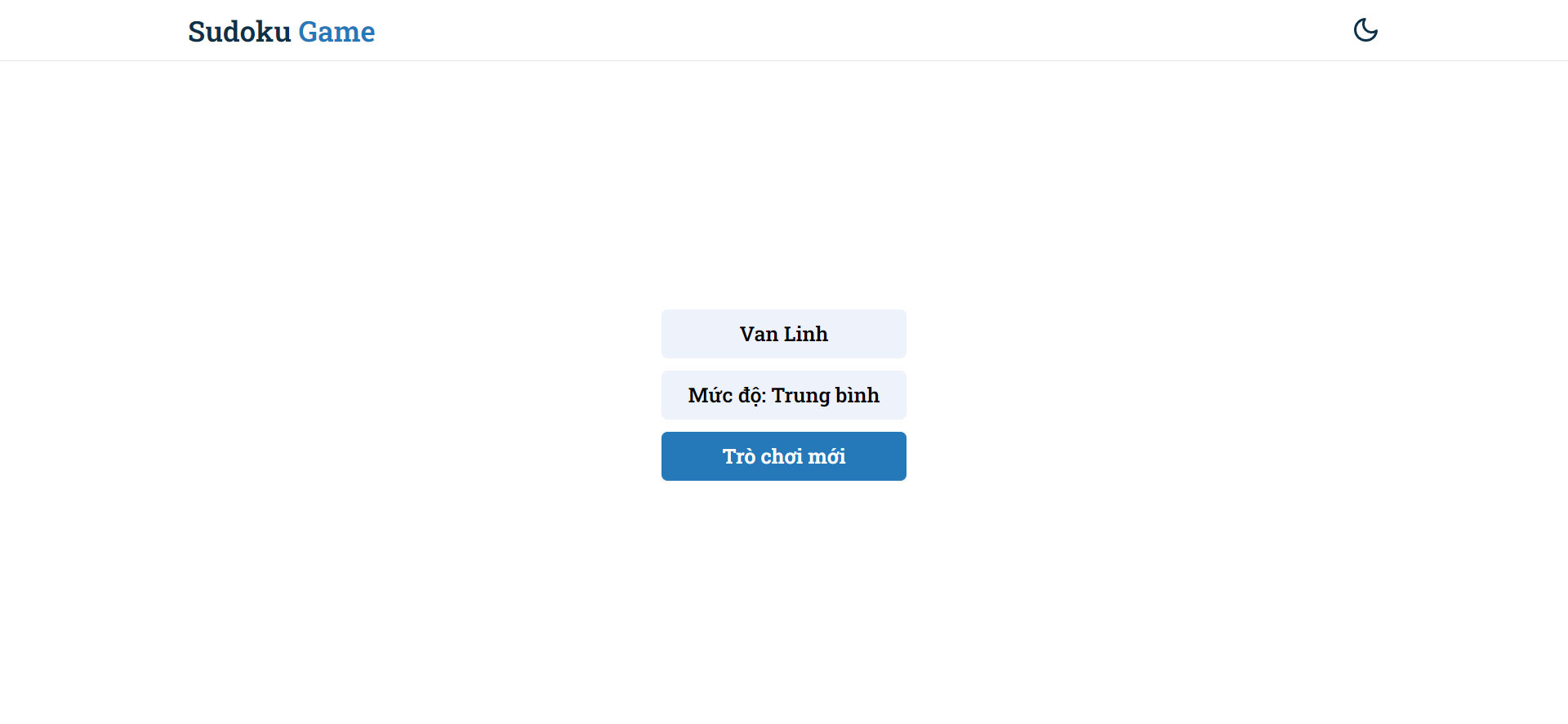
Description automatically generated with medium confidence

### 3.1.2. Thuật toán kiểm tra lời giải

Để kiểm tra lời giải của người chơi, thuật toán sẽ duyệt qua tất cả các ô số và kiểm tra các số còn lại trong hàng, cột và lưới 3 x 3 có trùng hay không. Nếu tất cả các ô số đều thỏa mãn thì lưu điểm cho người chơi.

## **3.2. Frontend**

* Với giao diện ban đầu của trò chơi người chơi có thể nhập tên và tùy chọn mức độ sau đó nhấn “Trò chơi mới”.



* Ở giao diện trò chơi người chơi có thể nhập vào các ô trống bằng bàn phím máy tính hoặc bàn phím ảo trên màn hình.

A picture containing table

Description automatically generated

* Khi đang chơi người chơi có thể tạm dừng trò chơi và thời gian chơi cũng sẽ tạm dừng cho đến khi người chơi bấm chơi trở lại.

A picture containing graphical user interface

Description automatically generated

* Sau khi giải xong người chơi có thể bấm kiểm tra để check xem đáp án đã đúng chưa. Nếu đáp án đúng sẽ hiện ra màn hình chúc mừng và thời gian giải của người chơi.

Graphical user interface, application

Description automatically generated

* Người chơi cũng có thể bật tùy chọn ghi chú nếu chưa chắc chắn với ô mình đang giải.

Graphical user interface

Description automatically generated with medium confidence

* Bên cạnh đó một số tính năng về giao diện như các ô cùng số với ô đang chọn sẽ được làm nổi bật, các ô cùng hàng cột và lưới 3 x 3 sẽ được làm nổi bật các ô điền cùng số sẽ có màu đỏ và giao diện tối giúp người chơi có trải nghiệm tốt hơn.

A picture containing text, computer, electronics, black

Description automatically generated

# **CHƯƠNG 4. KẾT LUẬN**

Đề tài “Xây dựng game Sudoku bằng ngôn ngữ JavaScript” đã được em thực hiện dưới sự hướng dẫn của thầy – TS. Nguyễn Tuấn Dũng. Đề tài này đã giúp em củng cố những kiến thức để làm trang web cũng như phát triển với nhiều kiến thức mới.

# **TÀI LIỆU THAM KHẢO**

[1] ReactJS <https://viblo.asia/p/gioi-thieu-ve-reactjs-phan-i-cac-khai-niem-co-ban-V3m5WzjblO7>

[2] Redux <https://viblo.asia/p/redux-trong-react-la-gi-1Je5EDX0lnL>

[3] RESTful API <https://hocspringboot.net/2020/10/26/mot-so-nguyen-tac-thiet-ke-api/>

[4] Các tạo game Sudoku <https://github.com/trananhtuat/javascript-sudoku>