TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÀI BÁO CÁO GIỮA KỲ MÔN LẬP TRÌNH WEB VÀ ỨNG DỤNG**

**TÌM HIỂU TOÀN DIỆN VỀ SPA**

*Người hướng dẫn*: **TRẦN BẢO TÍN**

*Người thực hiện*: **NGUYỄN XUÂN BÌNH – 52000018**

**NGUYỄN CÔNG HIỀN – 52000444**

Lớp **: 20050201**

Khoá  **: 24**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2022**

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÀI BÁO CÁO GIỮA KỲ MÔN LẬP TRÌNH WEB VÀ ỨNG DỤNG**

**TÌM HIỂU TOÀN DIỆN VỀ SPA**

*Người hướng dẫn*: **TRẦN BẢO TÍN**

*Người thực hiện*: **NGUYỄN XUÂN BÌNH – 52000018**

**NGUYỄN CÔNG HIỀN – 52000444**

Lớp **: 20050201**

Khoá  **: 24**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2022**

LỜI CẢM ƠN

Đầu tiên, chúng em xin cảm ơn Trường Đại học Tôn Đức Thắng đã tạo điều kiện để phục vụ tốt cho việc học tập của chúng em. Với tình hình cực kì căng thẳng do dịch covid, nhà trường đã nhanh chóng chuyển sang hình thức học trực tuyến để việc tiếp thu kiến thức của chúng em không bị gián đoạn.

Tiếp theo em xin chân thành cảm ơn sự giảng dạy nhiệt tình, tận tâm của thầy Trần Bảo Tín giảng dạy nhóm chúng em đã tạo điều kiện hỗ trợ, giúp đỡ chúng em trong suốt quá trình học online. Trong suốt quá trình học, chúng em đã nhận được rất nhiều sự quan tâm, giúp đỡ và chỉ dẫn của thầy. Thầy luôn tận tình chỉ bảo khi chúng em có những thắc mắc chưa hiểu rõ.

Nhờ có những lời hướng dẫn, dạy bảo của thầy nên chúng em mới có thể hoàn thiện tốt đẹp đề tài nghiên cứu. Cảm ơn thầy vì đã giúp đỡ chúng em trong quá trình học tập vừa qua ạ.

Chúng em kính chúc thầy nhiều sức khỏe, thành công và hạnh phúc.

TÓM TẮT

Bài báo cáo nghiên cứu về nội dung tổng quát của Single Page Application như khái niệm, ưu nhược điểm, ứng dụng và so sánh với web truyền thống.

Áp dụng thư viện React và ngôn ngữ php để tạo một trang web SPA demo có sử dụng Rest Api.

MỤC LỤC

[LỜI CẢM ƠN i](#_Toc104926414)

[TÓM TẮT ii](#_Toc104926415)

[MỤC LỤC 1](#_Toc104926416)

[DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ 3](#_Toc104926417)

[CHƯƠNG 1 – GIỚI THIỆU ĐỀ TÀI 4](#_Toc104926418)

[1.1 Giới thiệu đề tài 4](#_Toc104926419)

[1.2 Lý do chọn đề tài 4](#_Toc104926420)

[CHƯƠNG 2 – CƠ SỞ LÝ THUYẾT 5](#_Toc104926421)

[2.1 Đặc điểm của Client Side Rendering 5](#_Toc104926422)

[2.2 Khái niệm về SPA 5](#_Toc104926423)

[2.3 Cấu trúc của một Single Page Application 5](#_Toc104926424)

[2.3 Nguyên lý hoạt động của Single Page Application 7](#_Toc104926425)

[2.4 Ưu điểm của Single Page Application 7](#_Toc104926426)

[2.5 Nhược điểm của Single Page Application 8](#_Toc104926427)

[2.6 So sánh Single Page Application với trang Web truyền thống 8](#_Toc104926428)

[2.4.1 SPA 8](#_Toc104926429)

[2.4.2 Trang Web truyền thống 9](#_Toc104926430)

[CHƯƠNG 3 – PHÂN TÍCH THIẾT KẾ VÀ THỰC NGHIỆM 11](#_Toc104926431)

[3.1 Mô tả web minh họa 11](#_Toc104926432)

[3.2 Cấu trúc web minh họa 11](#_Toc104926433)

[3.3 Các chức năng của web minh họa 13](#_Toc104926434)

**DANH MỤC CÁC CHỮ VIẾT TẮT**

**CÁC CHỮ VIẾT TẮT**

SPA: Single Page Application

API: Application Programming Interface

SEO: Search Engine Optimization

DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ

**DANH MỤC HÌNH**

[Hình 2.1: Ví dụ về giao diện của một SPA cơ bản 6](#_Toc104926523)

[Hình 2.2: Hình ảnh minh họa nguyên lý của SPA 7](#_Toc104926524)

[Hình 2.3. Cách Single Page Application hoạt động. Nguồn: Monocubed 9](#_Toc104926525)

[Hình 3.1: Cấu trúc trang web 12](#_Toc104926526)

[Hình 3.2: Giao diện chọn vé 13](#_Toc104926527)

[Hình 3.3 Giao diện nhập thông tin đặt vé 14](#_Toc104926528)

[Hình 3.5: Giao diện thanh toán vé đã đặt 15](#_Toc104926529)

[Hình 3.6 Giao diện đặt vé khi xóa người đặt Nguyễn Xuân Bình 16](#_Toc104926530)

[Hình 3.8 Giao diện thanh toán vé sau khi nhấn nút purchase 17](#_Toc104926531)

[Hình 3.9 Database sau khi người dùng thực hiện thanh toán 17](#_Toc104926532)

CHƯƠNG 1 – GIỚI THIỆU ĐỀ TÀI

1.1 Giới thiệu đề tài

Hiện nay, Single Page Application (SPA) là một xu hướng cực kì nóng hổi trong lĩnh vực phát triển web. Mặc dù ra đời từ rất lâu những vẫn có rất nhiều trang web nổi tiếng áp dụng SPA. Tuy nhiên để làm ra một SPA tốt là không hề dễ dàng và đồng thời đòi hỏi rất nhiều công sức.

1.2 Lý do chọn đề tài

Với việc hiện nay SPA đang là xu hướng của ngành thiết kế web thì đây là một trong những lý do chính bọn em muốn chọn đề tài này. Một phần khác chúng em muốn tìm hiểu thêm về một lĩnh vực khác trong thiết kế web để giúp trau dồi thêm kiến thức có thể sử dụng trong tương lai.

**1.3 Ý nghĩa của đề tài**

Cùng với sự chuyển dịch mô hình web từ server-side rendering sang client-side rendering thì vấn đền tìm hiểu toàn diện về SPA đối với chúng em là một đề tài vô cùng cần thiết và bổ ích cho tương lai của chúng em. Qua việc tìm hiểu về SPA chúng em biết được thêm về một xu hướng mới trong lĩnh vực thiết kế web. Cùng với đó việc yêu cầu thiết kế một web SPA có sử dụng framework/ library, chúng em đã được biết cách sử dụng của thư viện ReactJs đồng thời áp dụng Web API và có thể viết được Rest API của riêng mình.

CHƯƠNG 2 – CƠ SỞ LÝ THUYẾT

2.1 Đặc điểm của Client Side Rendering

Cơ sở để hình thành 1 web SPA dựa trên cơ chế Client Side Redering. Vì vậy vấn đề về hiểu biết rõ về đặc điểm của CSR rất quan trọng:

Các đặc điểm của CSR:

* Những logic đơn giản (validation, đọc dữ liệu, sorting, filtering) nằm ở client side
* Logic để routing (chuyển trang), render (hiển thị) dữ liệu thì 96.69% là nằm ở client side
* Logic phức tạp (thanh toán, phân quyền) hoặc cần xử lý nhiều (data processing, report) **vẫn nằm ở server side.**

2.2 Khái niệm về SPA

Single Page Application (SPA) là một ứng dụng web hay là một trang web giúp nâng cao trải nghiệm người dùng bằng cách sử dụng HTML5 và AJAX. Về cách hoạt động, đầu tiên khi tải một trang web bất kỳ, SPA sẽ tải một trang HTML đơn, sau khi người dùng request, SPA sẽ tiếp tục tải các HTML khác trong cùng một trang đó, SPA có thể sử dụng một số thư viện Javascript như Angular JS, ReactJs, VueJs,... và chúng em đã sử dụng thư viện ReactJs cho web minh họa.

Vì SPA rất phù hợp cho những trang web **cần tương tác nhiều**, hầu hết web của các công ty công nghệ, công ty startup đều đùng cơ chế này như: Facebook, Youtube, Gmail, … SPA mang lại trải nghiệm cực kì tốt và mượt mà đặc biệt là khi điều hướng trang web

2.3 Cấu trúc của một Single Page Application

SPA thì toàn bộ resource (tài nguyên) của web chỉ bao gồm các file CSS, Javascript. Cấu trúc web page sẽ được load lần đầu tiên khi chúng ta bắt đầu duyệt một website nào đó. Ở những lần sau, khi chuyển đến một trang khác, client sẽ gửi đi những ajax request để get (lấy) dữ liệu cần thiết (thường sẽ là nội dung).

Một số thành phần sẽ được giữ nguyên ở web SPA khi người dùng điều hướng trong web như header, footer, logo, navigation bar…, một số thì chỉ nằm ở những vị trí cố định filter bar, banner, và còn rất nhiều layout và template được lặp lại (blog, self-service, thiết lập của google mail). SPA sử dụng hiệu quả những sự lặp lại như này và nhờ vào việc linh động viết lại những thành phần cần thiết (có thay đổi) trong trang hiện tại thay vì tải lại toàn bộ 1 trang mới từ Server, SPA loại bỏ sự ngắt quãng (tải lại trang) khi người dùng đang trải nghiệm những trang nối tiếp nhau, tạo cho người dùng cảm giác như đang sử dụng 1 ứng dụng trên máy tính.

Graphical user interface, application

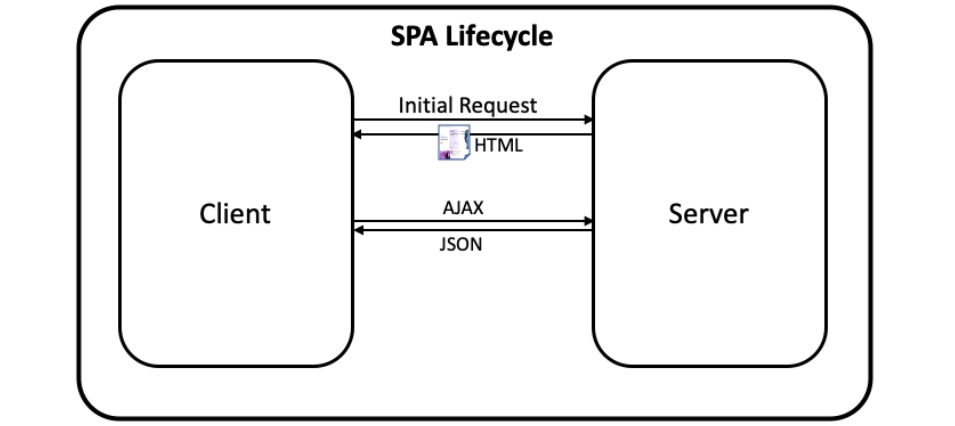
Description automatically generated

Hình 2.1: Ví dụ về giao diện của một SPA cơ bản

Như ví dụ về SPA ta có :

* Vùng màu xanh là khung trang hay còn được gọi là master page của website
* Vùng màu đỏ là phần hiển thị content của website sẽ hiển và thay đổi nội dung theo thao tác của người dùng , phần màu xanh không vị ảnh hưởng.

2.3 Nguyên lý hoạt động của Single Page Application



Hình 2.2: Hình ảnh minh họa nguyên lý của SPA

Cách hoạt động của [frontend](https://itnavi.com.vn/blog/frontend-la-gi/) và backend theo SPA có thể được mô tả như sau:

* Khi người dùng truy cập trang web, frontend sẽ tiếp nhận request chứ không phải backend. Web SPA thực hiện routing ở frontend chứ không
* Sau khi tiếp nhận request, frontend sẽ xác nhận người dùng muốn sử dụng tính năng nào và cần những dữ liệu gì sau đó mới gửi request tới backend. Cuối cùng, backend sẽ thực hiện yêu cầu và trả về những dữ liệu mong muốn.
* Frontend nhận dữ liệu từ backend và dựa vào dữ liệu này để render ra nội dung trang web một cách hoàn chỉnh.

2.4 Ưu điểm của Single Page Application

Các ưu điểm của SPA:

* Trang web chỉ cần **load một lần duy nhất**. Khi user chuyển trang hoặc thêm dữ liệu, JavaScript sẽ lấy và gửi dữ liệu từ server qua AJAX. User có thể thấy dữ liệu mới mà không cần chuyển trang.
* Chuyển logic sang client nên giảm tải được một phần cho server.
* Giảm được băng thông do chỉ cần lấy JSON và dữ liệu cần thiết, thay vì phải lấy toàn bộ trang
* Với việc chỉ xử lý những gì thay đổi thì sẽ không tốn thời gian nhiều trong việc xử lý dữ liệu. Vì vậy, SPA sẽ cải thiện đáng kể trải nghiệm người dùng, đặc biệt là người dùng mobile.
* Với các trang web cần tương tác nhiều, SPA **hoạt động mượt mà hơn** vì code chạy trên browser, không cần load đi loại lại nhiều

2.5 Nhược điểm của Single Page Application

Đồng thời SPA cũng có một số nhược điểm riêng:

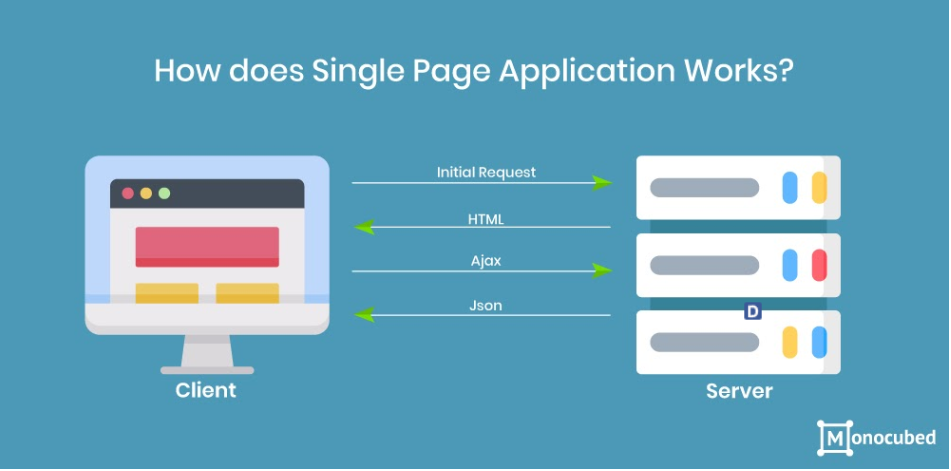
* [Lần tải trang web đầu tiên **sẽ chậm hơn**](https://medium.com/@addyosmani/the-cost-of-javascript-in-2018-7d8950fbb5d4) nếu không biết xử lý code hiệu quả. Vì trang web phải tải toàn bộ JavaScript về (khá nặng), và chạy JS, gọi API để lấy dữ liệu từ server (chậm), sau đó render dữ liệu
* Đòi hỏi project phải chia làm 2 phần riêng là **back-end (REST api) và front-end** nên khó code hơn
* Không chạy được ở các trình duyệt cũ không nhận [JavaScript ES6](https://toidicodedao.com/2016/04/05/series-javascript-sida-cung-lam-quen-va-chich-nham-nghich-es6/)
* SEO không tốt bằng website truyền thống vì người dùng sẽ phải cuộn trang web khá nhiều lần mới tìm thấy nội dung mà họ cần.
* Nếu client sử dụng điện thoại, thiết bị yếu thì khi tải trang web sẽ bị chậm

2.6 So sánh Single Page Application với trang Web truyền thống

2.4.1 SPA

Khi người dùng truy cập vào một thành phần bất kỳ, trang web chỉ tải lại nội dung đó mà không tải lại hết cả một trang web. Vì vậy, người dùng có thể truy cập nhiều trang con mà không ảnh hưởng gì đến trang gốc.

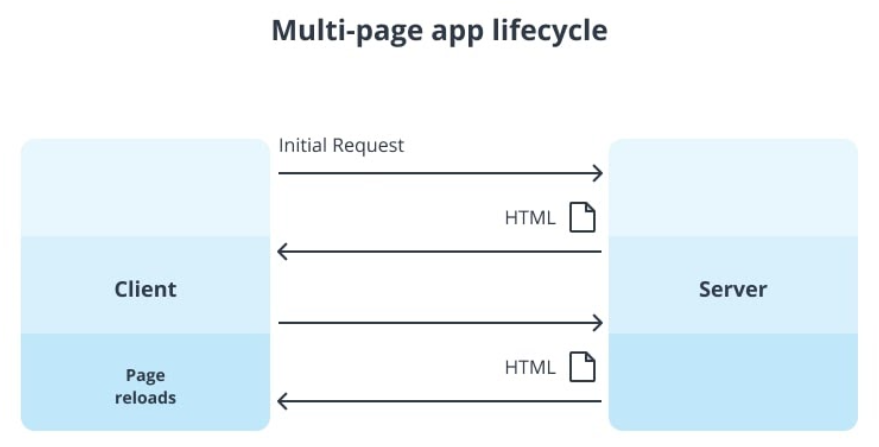
Ví dụ như Google : khi thực hiện chức năng tìm kiếm, những gì thay đổi là kết quả tìm kiếm được, còn thanh Google thì không có thay đổi. Tương tự như Facebook khi lướt News feed, hãy thử load lại trang, Facebook chỉ load lại phần nội dung của trang. Phần khung trang (master page) sẽ không thay đổi.

****

Hình 2.3. Cách Single Page Application hoạt động. Nguồn: Monocubed

2.4.2 Trang Web truyền thống

Khi người dùng thực hiện gửi một yêu cầu từ trang web, server sẽ nhận yêu cầu đó, tính toán và gửi lại toàn bộ trang web cho người dùng dưới dạng mã HTML. Ở phía client (phía người dùng) sẽ hiển thị kết quả trả về. Điều này khiến server thì tốn tải để tính toán, xử lý và phải gửi đi những thứ không thay đổi. Phía client lại chỉ đợi và hiển thị kết quả trả về.

****Hình 2.4. Cách website truyền thống hoạt động. Nguồn: Azoft

CHƯƠNG 3 – PHÂN TÍCH THIẾT KẾ VÀ THỰC NGHIỆM

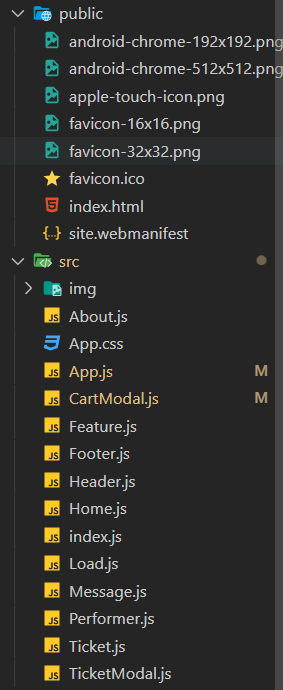
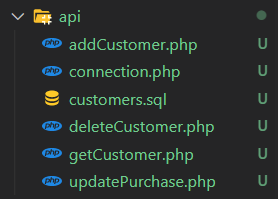
3.1 Mô tả web minh họa

Về chức năng của SPA, trang web minh họa của chúng em được sử dụng để đặt vé ca nhạc mang tên Melody. Trang web sử dụng thư viện React để xây dựng phần front end và dùng ngôn ngữ php để xây dựng các Rest API . Về tính năng để áp dụng được Rest API do bọn em viết thì web có một chức năng là đặt vé xem phim. Người dùng có thể nhập thông tin cá nhân của họ như tên, email và số điện thoại. Sau đó thông tin người dùng sẽ được lưu trữ ở database và hiển thị lên trang web.

3.2 Cấu trúc web minh họa

Về chức năng của SPA, trang web minh họa của chúng em là một website giới thiệu về một buổi trình diễn ca nhạc sử dụng thư viện ReactJs. Về tính năng để áp dụng được Rest API do bọn em viết thì web có một chức năng là đặt vé online . Người dùng có thể lựa chọn loại vé và nhập thông tin cá nhân của họ như tên, email. Sau đó thông tin đặt vé của người dùng sẽ được lưu trữ ở database.

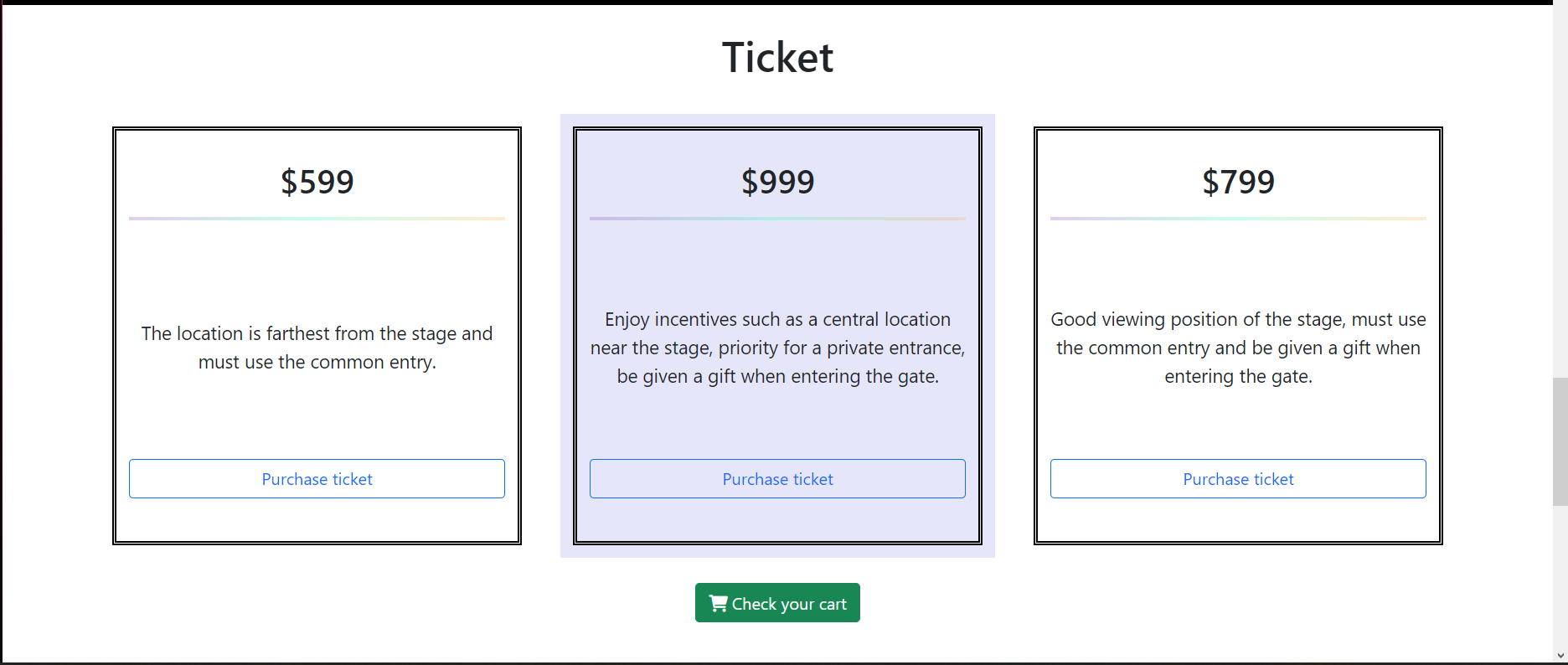
Về cấu trúc của trang web minh họa:



Hình 3.1: Cấu trúc trang web

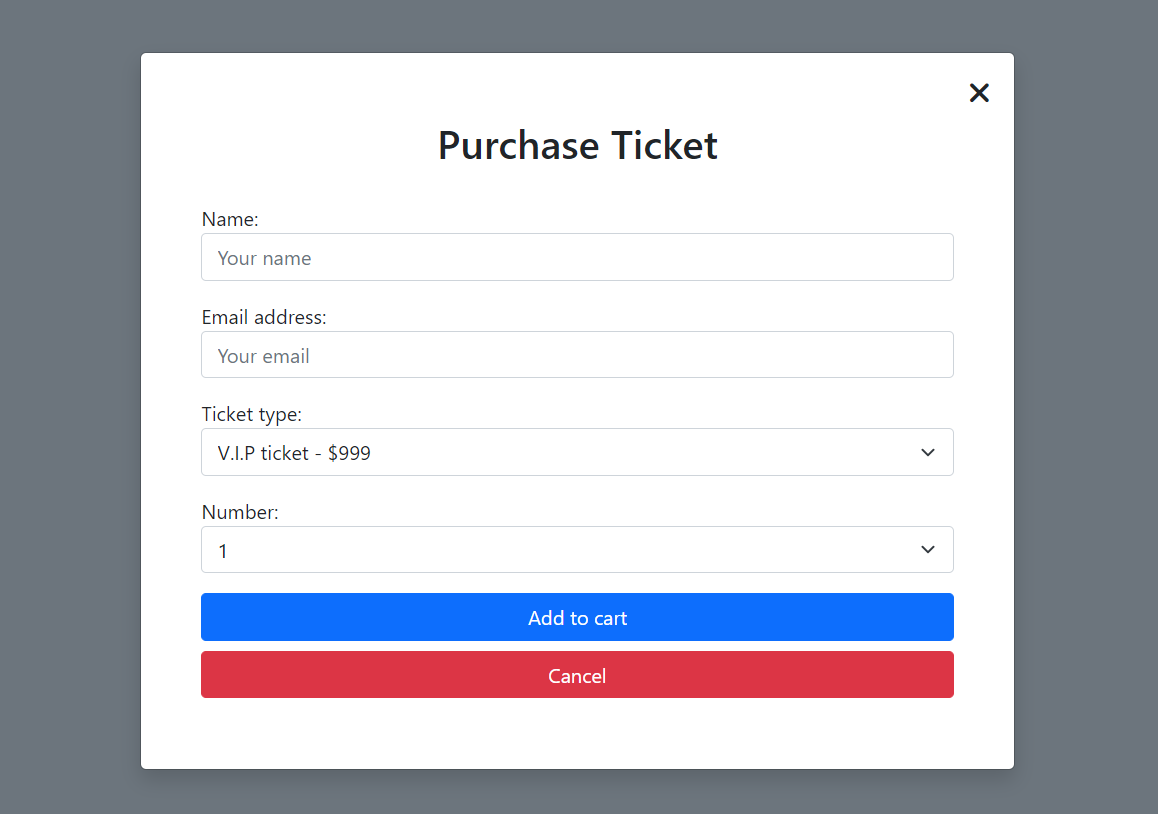
Resource của trang web bao gồm folder public chứa các file ảnh favicon và file index.html làm khung sườn của website. Folder src chứa các file component function, file App.css làm cho trang web thêm sinh động, folder img chứa các file ảnh của trang web. Cuối cùng là folder api bao gồm các file addCustomer.php thêm thông tin đặt vé của khách hàng, file connection.php để kế nối database, file customers.sql chứa cơ sỡ dữ liệu của trang web, file deleteCustomer.php để xóa thông tin đặt vé của khách hàng, file getCustomer.php để lấy thông tin đặt vé của khách hàng trong database, file updatePurchase.php để cập nhật thông tin đặt vé của khách hàng trong database.

3.3 Các chức năng của web minh họa



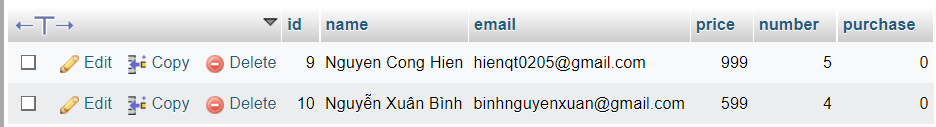
Hình 3.2: Giao diện chọn vé

Ở giao diện chọn vé, có ba loại vé cho khác hàng lựa chọn với 3 mức giá khác nhau, và thông tin chi tiết của từng loại vé được hiển thị một cách rõ ràng giúp khách hàng lựa chọn một cách dễ dàng hơn. Nếu muốn đặt loại vé nào thì click vào nút purchase ticket ở loại thẻ tương ứng. Nút Check your cart để hiện thị các loại vé của khách đã đặt nhưng chưa thực hiện thanh toán.



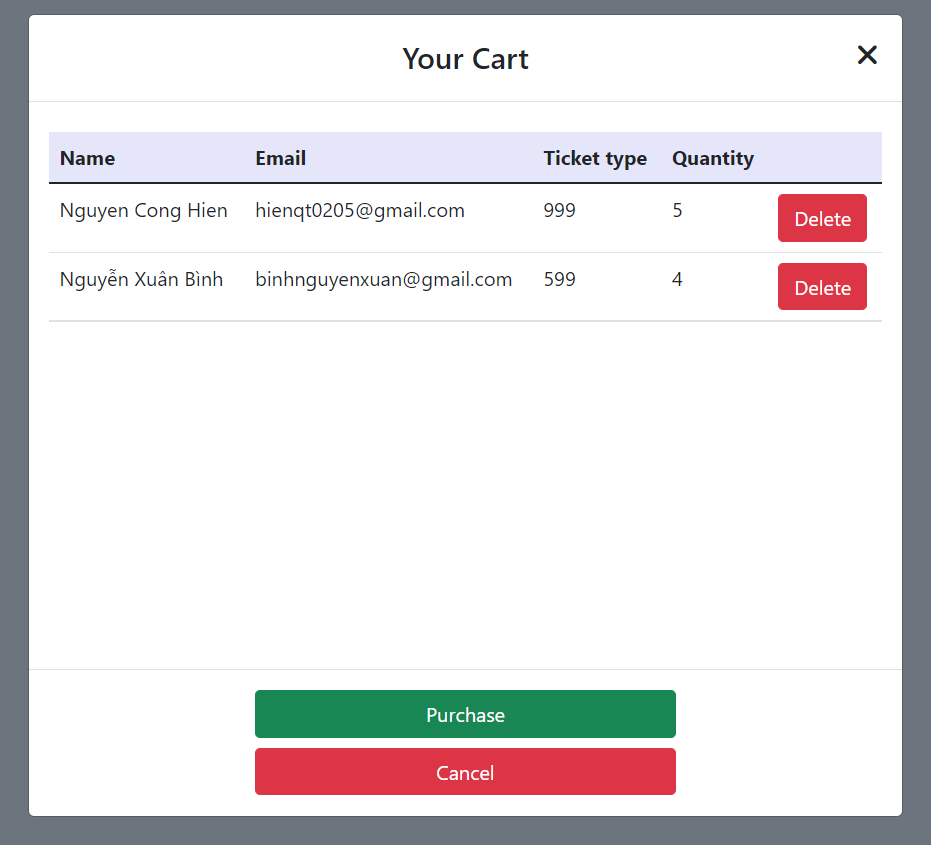
Hình 3.3 Giao diện nhập thông tin đặt vé

Ở giao diện nhập thông tin đặt vé, khách hàng cần nhập đầy đủ các thông tin như tên, địa chỉ email, loại vé có thể thay đổi tại giao diện này và số lượng. Nếu người dùng muốn thêm vào giỏ hàng chỉ cần nhấn nút Add to cart. Ngược lại, khách hàng cần nhấn nút cancel hoặc biểu tượng x ở góc phải trên để thoát khỏi giao diện Purchase ticket.



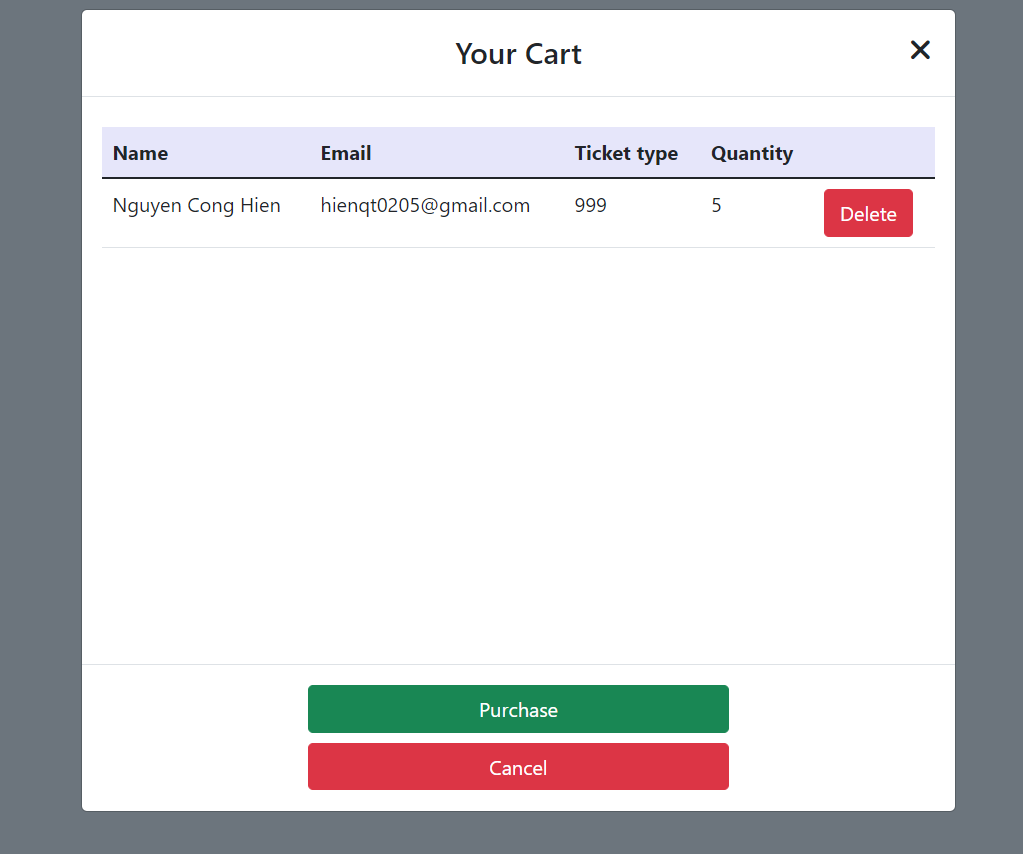
Hình 3.4 Thông tin đặt vé của khách hàng được thêm vào database

Sau khi khách hàng nhấn nút Add to cart thì thông tin đặt vé của khách hàng được thêm vào database. Trong đó có thêm cột purchase thể hiện trạng thái thanh toán với 0 là chưa thanh toán và 1 là đã thanh toán.

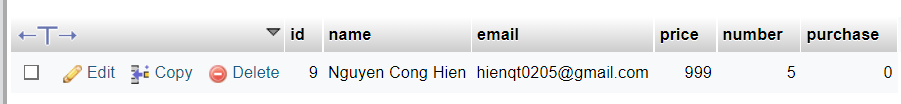


Hình 3.5: Giao diện thanh toán vé đã đặt

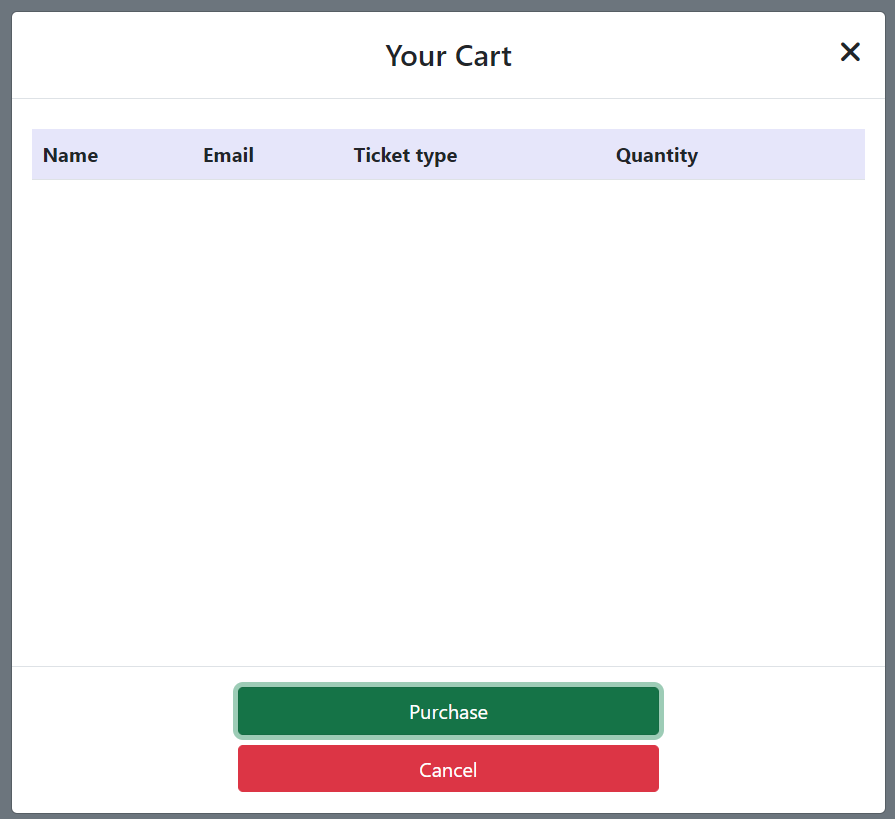
Ở giao diện thanh toán vé, ở đây sẽ hiển thị thông tin các vé mà người dùng đã đặt nhưng chưa thanh toán. Khách hàng có thể hủy đặt vé bằng cách nhấn nút delete bên phải vé tương ứng. Nếu khách hàng muốn thực hiện thanh toán thì nhấn nút Purchase bên dưới. Nếu khách hàng muốn thoát khỏi giao diện thanh toán vé thì nhấn nút Cancel hoặc nút x ở góc phải trên.



Hình 3.6 Giao diện đặt vé khi xóa người đặt Nguyễn Xuân Bình

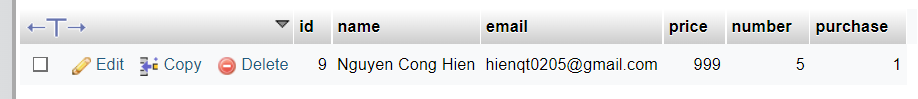


Hình 3.7 Database sau khi xóa thông tin vé người đặt Nguyễn Xuân Bình

****

Hình 3.8 Giao diện thanh toán vé sau khi nhấn nút purchase

Sau khi nhấn nút Purchase thì các loại vé trong giao diện thanh toán vé sẽ bị xóa hết và hiển thị thông báo thanh toán thành công. Cột purchase trong database sẽ đổi thành 1.



Hình 3.9 Database sau khi người dùng thực hiện thanh toán

**TÀI LIỆU THAM KHẢO**

1. Single Page Application là gì? – stdio.vn – Truy cập ngày 25/5/2022

[SPA - Single Page Application Là Gì? — JavaScript (stdio.vn)](https://www.stdio.vn/javascript/spa-single-page-application-la-gi-3jnH51)

2. Sự khác biệt giữa server-side rendering và client-side rendering – toidicodedao – Truy cập ngày 25/5/2022

[Sự khác biệt giữa server-side rendering và client-side rendering | Từ coder đến developer – Tôi đi code dạo (toidicodedao.com)](https://toidicodedao.com/2018/09/11/su-khac-biet-giua-server-side-rendering-va-client-side-rendering/)

3. Single Page Application – ITNavi – Truy cập ngày 25/5/2002

[Single Page Application là gì? Đây có phải là xu hướng lập trình Web hiện nay? (itnavi.com.vn)](https://itnavi.com.vn/blog/single-page-application-la-gi)