

KHOA KỸ THUẬT VÀ CÔNG NGHỆ  
BỘ MÔN CÔNG NGHỆ THÔNG TIN



**THỰC TẬP ĐỒ ÁN CƠ SỞ NGÀNH**

HỌC KỲ I, NĂM HỌC 2023-2024

**NGHIÊN CỨU NODEJS VÀ  
XÂY DỰNG ỨNG DỤNG CHIA  
SẺ XE CHO GIÁNG VIÊN**

*Giáo viên hướng dẫn:*  
ThS. Nguyễn Bảo Ân

*Sinh viên thực hiện:*  
Họ tên: Sĩ Chí Thiện  
MSSV:110121107  
Lớp:DA21TTA

*Trà Vinh, tháng 01 năm 2024*

KHOA KỸ THUẬT VÀ CÔNG NGHỆ  
BỘ MÔN CÔNG NGHỆ THÔNG TIN



**THỰC TẬP ĐỒ ÁN CƠ SỞ NGÀNH**

HỌC KỲ I, NĂM HỌC 2023-2024

**NGHIÊN CỨU NODEJS VÀ  
XÂY DỰNG ỨNG DỤNG CHIA  
SẺ XE CHO GIÁNG VIÊN**

*Giáo viên hướng dẫn:*  
ThS. Nguyễn Bảo Ân

*Sinh viên thực hiện:*  
Họ tên: Sĩ Chí Thiện  
MSSV:110121107  
Lớp:DA21TTA

*Trà Vinh, tháng 01 năm 2024*

**NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

[illegible]

## LỜI CẢM ƠN

Em xin chân thành cảm ơn các thầy cô trường Đại học Trà Vinh, đặc biệt là các thầy cô khoa Công nghệ Kỹ thuật đã tạo cơ hội cho em được học tập và phát triển. Em có thể tránh được các vấn đề và nhầm lẫn trong môi trường làm việc trong tương lai.

Em xin cảm ơn Thầy Nguyễn Bảo Ân. Nhờ sự giúp đỡ và hướng dẫn tận tình của thầy từ đầu đến cuối đồ án, chúng em đã hoàn thành đồ án đúng thời hạn và tích lũy được một vốn kiến thức quý báu.

Mặc dù đã cố gắng hết sức để hoàn thành đề tài này nhưng do hạn chế về thời gian cũng như kiến thức chuyên môn nên trong quá trình nghiên cứu, đánh giá và trình bày đề tài, còn nhiều thiếu sót. Em rất mong được sự quan tâm, góp ý của các thầy cô bộ môn để đồ án cơ sở ngành của em được hoàn thiện hơn.

Sinh viên thực hiện:

110121107 – Sĩ Chí Thiện

Trà Vinh, ngày 08 tháng 01 năm 2024

## MỤC LỤC

CHƯƠNG 1 TỔNG QUAN.....	8
1.1 Mô tả bài toán.....	8
1.2 Công cụ và cách thức thực hiện .....	8
1.3 Xác định mô hình dữ liệu .....	8
CHƯƠNG 2 NGHIÊN CỨU LÝ THUYẾT .....	9
2.1 Nghiên cứu RESTfull API.....	9
2.1.1 Tổng quan về RESTfull API .....	9
2.1.2 Các thành phần của RESTfull API .....	9
2.1.3 RESTful hoạt động như thế nào? .....	10
2.1.4 Kiến thức về framework ExpressJS.....	10
2.2 Hệ quản trị cơ sở dữ liệu MongoDB .....	11
CHƯƠNG 3 HIỆN THỰC HÓA NGHIÊN CỨU .....	13
3.1 Mô tả ứng dụng .....	13
3.2 Đặc tả các chức năng hệ thống.....	13
3.3 Thiết kế hệ thống.....	13
Thiết kế dữ liệu .....	14
CHƯƠNG 4 KẾT QUẢ NGHIÊN CỨU .....	21
4.1 Sản phẩm đạt được sau quá trình nghiên cứu .....	21
4.2 Về hiệu năng .....	21
4.3 Về trải nghiệm người dùng.....	22
4.4 Kết quả thử nghiệm .....	22
Thông báo đăng nhập thành công .....	24
4.4.1 Chức năng Tạo chuyến xe.....	24
4.4.2 Chức năng Cập nhật chuyến xe .....	26
CHƯƠNG 5 KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN .....	29
5.1 Kết luận.....	29
5.1.1 Kết quả đạt được.....	29
5.1.2 Những đóng góp mới .....	29
5.1.3 Những đề xuất mới .....	29
5.2 Hướng phát triển .....	30

## DANH MỤC HÌNH ẢNH

Hình 1. Cách hoạt động của RESful API .....	10
Hình 2. Mô hình của ExpressJS .....	11
Hình 3. Cách hoạt động của MongoDB .....	12
Hình 4. Lược đồ mô tả hệ thống.....	13
Hình 5. Thiết kế dữ liệu.....	14
Hình 6. Mã nguồn của collection trips .....	16
Hình 7. Mã nguồn của collection users .....	17
Hình 8. Mã nguồn của API tạo chuyến đi .....	18
Hình 9. Mã nguồn của API update chuyến đi.....	19
Hình 10. Mã nguồn của API tìm chuyến đi .....	19
Hình 11. Mã nguồn của API xóa chuyến đi.....	20
Hình 12. Mã nguồn của API login .....	21
Hình 13. Giao diện chính.....	23
Hình 14. Form đăng nhập .....	23
Hình 15. Thông báo đăng nhập thành công.....	24
Hình 16. Nút tạo chuyến xe .....	24
Hình 17. Giao diện tạo chuyến xe.....	25
Hình 18. Thông báo tạo chuyến xe thành công .....	26
Hình 19. Nút cập nhật chuyến xe.....	26
Hình 20. Giao diện cập nhật chuyến xe.....	27
Hình 21. Thông báo cập nhật chuyến xe thành công .....	27
Hình 22. Nút xóa chuyến xe .....	28
Hình 23. Thông báo xác nhận xóa chuyến xe.....	28

## TÓM TẮT ĐỒ ÁN CƠ SỞ NGÀNH

Đề tài: Nghiên cứu Node.js và xây dựng ứng dụng chia sẻ xe cho giảng viên.

Mục tiêu: Nghiên cứu về Node.js, một nền tảng phát triển ứng dụng web và mobile dựa trên JavaScript. Xây dựng ứng dụng quản lý thực tập cho sinh viên bộ môn công nghệ thông tin dựa trên các công cụ NodeJS, framework ExpressJS và cơ sở dữ liệu MongoDB

Hướng tiếp cận: Nghiên cứu lý thuyết: Tìm hiểu các tài liệu, nghiên cứu trước đây về NodeJS, ExpressJS, MongoDB và cách thức hoạt động của một ứng dụng sử dụng NodeJS. Nghiên cứu thực nghiệm: Xây dựng ứng dụng chia sẻ xe cho giảng viên.

Cách giải quyết vấn đề: NodeJS là một nền tảng phát triển ứng dụng web và mobile hiệu quả, linh hoạt và dễ sử dụng, Express là một framework NodeJS giúp phát triển ứng dụng web back-end một cách nhanh chóng và dễ dàng. MongoDB là một hệ quản trị cơ sở dữ liệu NoSQL hiệu quả và linh hoạt.

Kết quả đạt được: Ứng dụng chia sẻ xe cho giảng viên được xây dựng thành công, đáp ứng được các yêu cầu đề ra. Ứng dụng có giao diện thân thiện, dễ sử dụng. Ứng dụng có khả năng mở rộng, dễ dàng tích hợp với các hệ thống khác.

Bài học kinh nghiệm: Cần tìm hiểu kỹ các tài liệu, nghiên cứu trước đây về đề tài nghiên cứu. Cần lập kế hoạch thực hiện chi tiết và khoa học. Cần kiên trì và nỗ lực để hoàn thành đề tài.



## MỞ ĐẦU

### Lí do chọn đề tài:

Mỗi năm, trường của chúng ta đều phải chi trả một lượng lớn tiền cho chi phí xe cộ của giảng viên khi họ thực hiện các chuyến đi liên quan đến công việc, gặp đối tác, và thực hiện những nhiệm vụ chính của mình. Điều này đã tạo ra một hao hụt nguồn lực đáng kể cho nhà trường.

### Mục đích nghiên cứu:

Với mong muốn giải quyết vấn đề này, em đã thực hiện nghiên cứu và phát triển ứng dụng "Chia Sẻ Xe Cho Giảng Viên". Ứng dụng này không chỉ giúp trường tiết kiệm được chi phí cho việc di chuyển của giảng viên mà còn tối ưu hóa các chuyến đi, giúp họ dễ dàng hơn trong việc tìm kiếm và tham gia các chuyến đi phù hợp. Điều này không chỉ giúp nhà trường tiết kiệm được tài chính mà còn tạo điều kiện thuận lợi cho giảng viên trong việc tìm kiếm các chuyến đi phù hợp.

## CHƯƠNG 1 TỔNG QUAN

### 1.1 Mô tả bài toán

Mỗi năm trường của chúng ta phải tiêu tốn rất nhiều chi phí xe cộ cho giảng viên khi thực hiện các chuyến đi liên quan đến công việc, gặp các đối tác,...khiến nhà trường hao hụt nguồn lực.

Chính vì vậy, em đã nghiên cứu và tạo ra một ứng dụng có thể giải quyết được vấn đề này, đó là ứng dụng “chia sẻ xe cho giảng viên”, giúp nhà trường tiết kiệm được chi phí cho các chuyến đi của giảng viên, có thể tối ưu hóa các chuyến đi và giúp giảng viên thuận lợi hơn trong việc tìm kiếm các chuyến đi.

### 1.2 Công cụ và cách thức thực hiện

Để có thể giải quyết được bài toán trên, em đã nghiên cứu và tìm hiểu các công cụ hợp lý nhất để có thể thực hiện ứng dụng chia sẻ xe cho giảng viên, đó chính là dùng Nodejs, Express và cơ sở dữ liệu MongoDB.

### 1.3 Xác định mô hình dữ liệu

. Mô hình dữ liệu bao gồm 2 bảng dữ liệu là trips và users

#### 1.3.1. Kết quả nghiên cứu

Sao quá trình nghiên cứu và thực hiện dự án ứng dụng chia sẻ xe cho giảng viên, em đã tiếp thu được nhiều kiến thức mới như là cách làm việc với NodeJS, ExpressJS và cơ sở dữ liệu NoSQL đó là MongoDB, điều này giúp em mở rộng được vốn kiến thức và tích lũy được nhiều kinh nghiệm quý báu, tạo tiền đề cho những dự án sau này có thể thực hiện tốt hơn.

#### 1.3.2. Kết quả đạt được

Ứng dụng sao quá trình nghiên cứu và thực hiện, đã hoàn thành với đầy đủ các chức năng thêm, sửa, xóa. Giao diện thân thiện, có thể dễ dàng làm quen và sử dụng, đáp ứng được những yêu cầu đã đề ra.

## CHƯƠNG 2 NGHIÊN CỨU LÝ THUYẾT

### 2.1 Nghiên cứu RESTfull API

RESTful API, viết tắt của Representational State Transfer, là một kiến thức cơ bản và quan trọng trong lĩnh vực phát triển ứng dụng web và mobile. Đây là một kiến thức không thể thiếu đối với các nhà phát triển muốn xây dựng các hệ thống có khả năng mở rộng, linh hoạt và tương thích trên nhiều nền tảng.

#### 2.1.1 Tổng quan về RESTfull API

Có thể nói nguyên lý **REST** và cấu trúc dữ liệu **RESTful** được biết đến rộng rãi trong giới lập trình web nói chung và lập trình ứng dụng nói riêng.

Có thể nói bản thân **REST** không phải là một loại công nghệ. Nó là phương thức tạo API với nguyên lý tổ chức nhất định. Những nguyên lý này nhằm hướng dẫn lập trình viên tạo môi trường xử lý API request được toàn diện.

#### 2.1.2 Các thành phần của RESTfull API

**API** (Application Programming Interface) là một tập các quy tắc và cơ chế mà theo đó, một ứng dụng hay một thành phần sẽ tương tác với một ứng dụng hay thành phần khác. API có thể trả về dữ liệu mà bạn cần cho ứng dụng của mình ở những kiểu dữ liệu phổ biến như JSON hay XML.

**REST** (REpresentational State Transfer) là một dạng chuyển đổi cấu trúc dữ liệu, một kiểu kiến trúc để viết API. Nó sử dụng phương thức HTTP đơn giản để tạo cho giao tiếp giữa các máy. Vì vậy, thay vì sử dụng một URL cho việc xử lý một số thông tin người dùng, REST gửi một yêu cầu HTTP như GET, POST, DELETE, vv đến một URL để xử lý dữ liệu.

**RESTful API** là một tiêu chuẩn dùng trong việc thiết kế các API cho các ứng dụng web để quản lý các resource. RESTful là một trong những kiểu thiết kế API được sử dụng phổ biến ngày nay để cho các ứng dụng (web, mobile...) khác nhau giao tiếp với nhau.

Chức năng quan trọng nhất của **REST** là quy định cách sử dụng các HTTP method (như GET, POST, PUT, DELETE...) và cách định dạng các URL cho ứng dụng

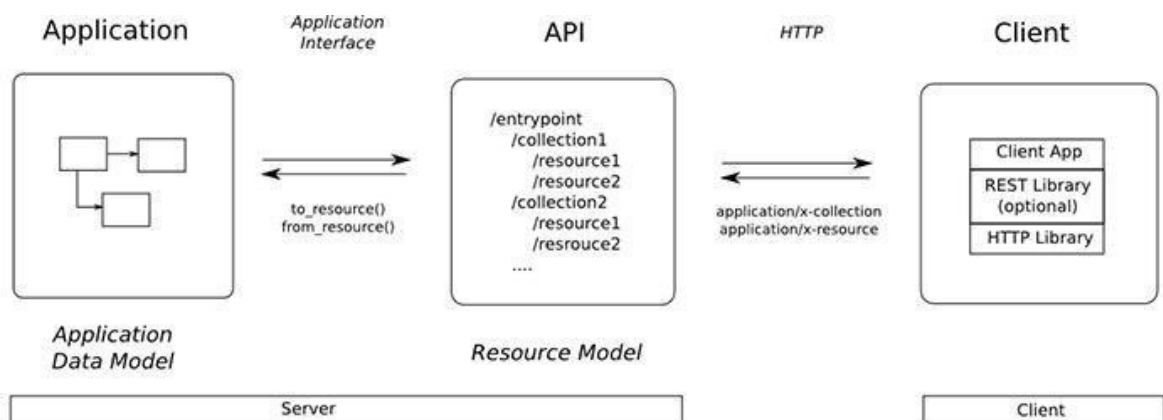
web để quản các resource. RESTful không quy định logic code ứng dụng và không giới hạn bởi ngôn ngữ lập trình ứng dụng, bất kỳ ngôn ngữ hoặc framework nào cũng có thể sử dụng để thiết kế một **RESTful API**.

### 2.1.3 RESTful hoạt động như thế nào?

REST hoạt động chủ yếu dựa vào giao thức HTTP. Các hoạt động cơ bản nêu trên sẽ sử dụng những phương thức HTTP riêng.

- GET (SELECT): Trả về một Resource hoặc một danh sách Resource.
- POST (CREATE): Tạo mới một Resource.
- PUT (UPDATE): Cập nhật thông tin cho Resource.
- DELETE (DELETE): Xoá một Resource.

Những phương thức hay hoạt động này thường được gọi là **CRUD** tương ứng với Create, Read, Update, Delete – Tạo, Đọc, Sửa, Xóa.



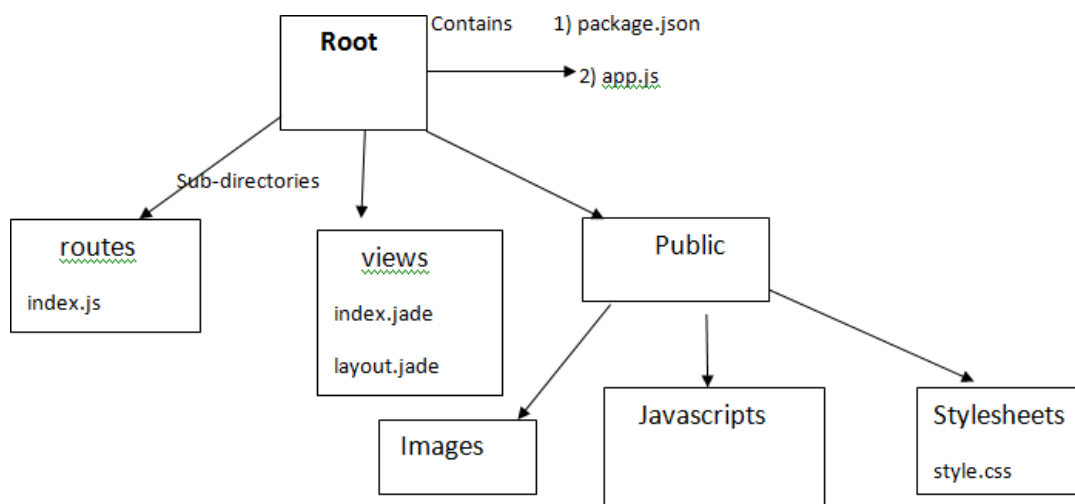
Hình 1. Cách hoạt động của RESTful API

### 2.1.4 Kiến thức về framework ExpressJS

ExpressJS là một web framework được xây dựng trên nền tảng NodeJs. **Expressjs** cung cấp các hàm HTTP và middleware để tạo ra API đơn giản và dễ sử dụng.

ExpressJS được phát hành theo giấy phép mã nguồn mở, có cộng đồng hỗ trợ lớn, được phép sử dụng cho ứng dụng có mục đích thương mại. Do vậy bạn hoàn toàn có thể yên tâm sử dụng framework này cho dự án của mình, từ dự án nhỏ tới lớn.

Cấu trúc thư mục dự án khi sử dụng ExpressJS được chia là 3 phần: routes, Views và Public. ExpressJS xây dựng ứng dụng web theo đúng mô hình MVC (Model – View – Controller).



Hình 2. Mô hình của ExpressJS

Một số chức năng chính của ExpressJS:

Hỗ trợ middleware để trả về các HTTP request

Định nghĩa route dựa trên các action của HTTP (CRUD)

Cho phép trả về các trang html sử dụng các template engine (jade, pug...)

## 2.2 Hệ quản trị cơ sở dữ liệu MongoDB

MongoDB hoạt động dưới một tiến trình ngầm service, luôn mở một cổng (Cổng mặc định là 27017) để lắng nghe các yêu cầu truy vấn, thao tác từ các ứng dụng gửi vào sau đó mới tiến hành xử lý.

Mỗi một bản ghi của MongoDB được tự động gắn thêm một field có tên “\_id” thuộc kiểu dữ liệu ObjectId mà nó quy định để xác định được tính duy nhất của bản ghi này so với bản ghi khác, cũng như phục vụ các thao tác tìm kiếm và truy vấn thông tin về sau. Trường dữ liệu “\_id” luôn được tự động đánh index (chỉ mục) để tốc độ truy vấn thông tin đạt hiệu suất cao nhất.

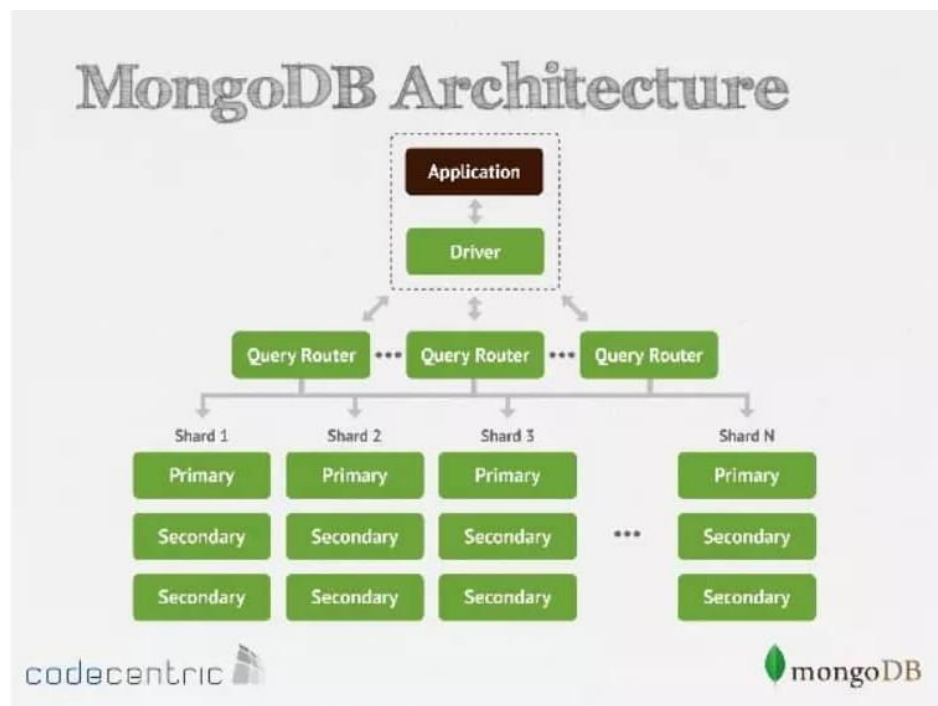
Mỗi khi có một truy vấn dữ liệu, bản ghi được cache (ghi đệm) lên bộ nhớ Ram, để phục vụ lượt truy vấn sau diễn ra nhanh hơn mà không cần phải đọc từ ổ cứng.

Khi có yêu cầu thêm/sửa/xóa bản ghi, để đảm bảo hiệu suất của ứng dụng mặc định MongoDB sẽ chưa cập nhật xuống ổ cứng ngay, mà sau 60 giây MongoDB mới thực hiện ghi toàn bộ dữ liệu thay đổi từ RAM xuống ổ cứng.

Từ đây có thể nhìn thấy *nhược điểm* của Mongoddb như sau:

Dữ liệu được caching, lấy RAM làm trọng tâm hoạt động vì vậy khi hoạt động yêu cầu một bộ nhớ RAM lớn

Như đã giới thiệu ở trên, mọi thay đổi về dữ liệu mặc định đều chưa được ghi xuống ổ cứng ngay lập tức vì vậy khả năng bị mất dữ liệu từ nguyên nhân mất điện đột xuất là rất cao.



Hình 3. Cách hoạt động của MongoDB

## CHƯƠNG 3 HIỆN THỰC HÓA NGHIÊN CỨU

### 3.1 Mô tả ứng dụng

Ứng dụng chia sẻ xe dành cho giảng viên đã được thiết kế với mục tiêu đơn giản hóa quá trình di chuyển và giảm bớt gánh nặng tài chính cho nhà trường. Giao diện người dùng thân thiện giúp giảng viên dễ dàng đăng ký và đặt chuyến xe. Giúp tối ưu hóa các chuyến xe có sẵn và tiết kiệm được nhiều chi phí.

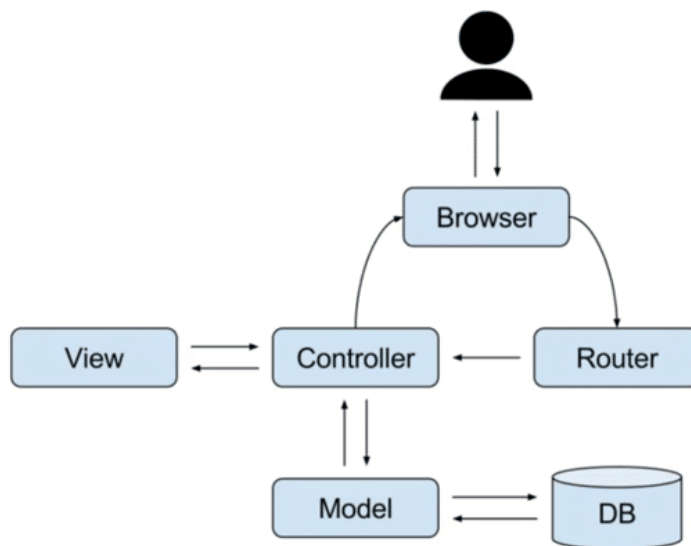
### 3.2 Đặc tả các chức năng hệ thống

Ứng dụng chia sẻ xe là một hệ thống giúp cho giảng viên dễ dàng biết được những chuyến xe đang có hiện tại và dễ dàng đăng ký chuyến xe thông qua việc liên lạc thẳng vào số điện thoại của chủ chuyến xe.

Bên cạnh đó ứng dụng cũng có chức năng đăng nhập dành cho các chủ xe có thể đăng nhập và làm những việc trong chức năng của họ, trước khi đăng nhập thì người dùng cần phải có username và mật khẩu, username và mật khẩu phải được cung cấp từ trước bởi người quản trị của ứng dụng.

### 3.3 Thiết kế hệ thống

#### 3.3.1 Lược đồ mô tả hệ thống



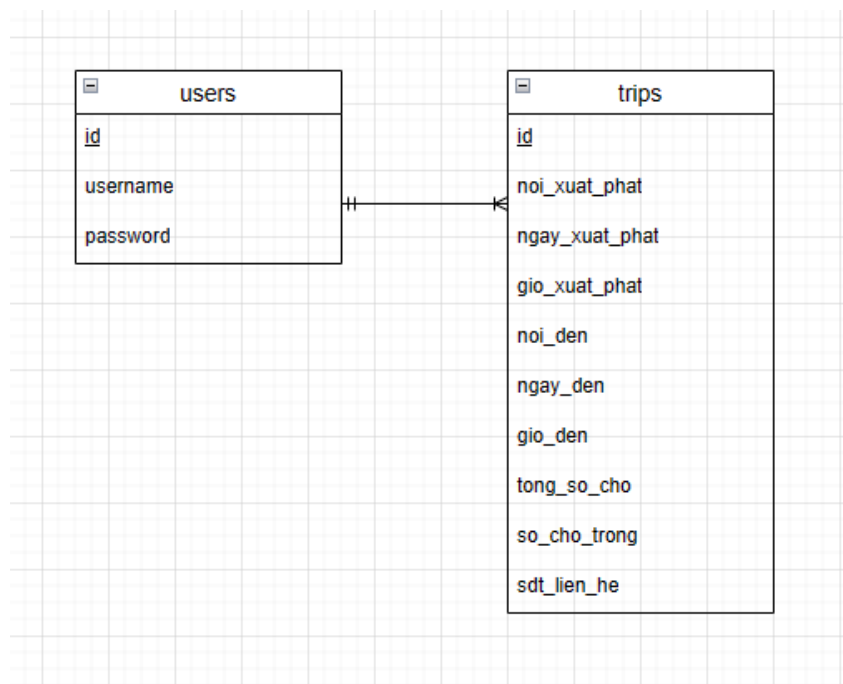
Hình 4. Lược đồ mô tả hệ thống

### 3.3.2. Cơ sở dữ liệu

Sử dụng MongoDB vì nó là một cơ sở dữ liệu NoSQL linh hoạt, có thể lưu trữ dữ liệu ở dạng JSON.

Tạo hai collections chính: users và trips.

#### Thiết kế dữ liệu



Hình 5. Thiết kế dữ liệu

#### Collection users

Lưu trữ thông tin về người dùng như username, password,...

Sử dụng một ObjectId để duy nhất xác định mỗi người dùng. Mỗi document đại diện cho một users

#### Collection trips



Mỗi chuyến đi đều gồm có các thông tin chi tiết như nơi xuất phát, ngày xuất phát, giờ xuất phát, nơi đến, ngày đến, giờ đến, tổng số chỗ, số chỗ trống, số điện thoại liên hệ. Mỗi document đại diện cho một chuyến đi.

### **Mã nguồn của các collections**

```
//./server/model/model.js
const mongoose = require('mongoose'); 486.8k (gzipped: 120.2k)

var schema = new mongoose.Schema({
  noi_xuat_phat : {
    type : String,
    required: true
  },
  ngay_xuat_phat : {
    type: String,
    required: true,
  },
  gio_xuat_phat : {
    type : String,
    required: true
  },
  noi_den : {
    type : String,
    required: true
  },
  ngay_den : {
    type : String,
    required: true
  },
  gio_den : {
    type : String,
    required: true
  },
  tong_so_cho : {
    type : Number,
    required: true
  },
  so_cho_trong: {
    type : Number,
    required: true
  },
  sdt_lien_he : {
    type : Number,
    required: true
  },
},
})

const CreateTrip = mongoose.model( name: 'createTrip', schema);

module.exports = CreateTrip;
```

Hình 6. Mã nguồn của collection trips

```
///./server/model/users.js
const mongoose = require('mongoose'); 486.8k (gzipped: 120.2k)

const userSchema = new mongoose.Schema({
  username: {
    type: String,
    required: true,
    unique: true, // Đảm bảo mỗi username là duy nhất
  },
  password: {
    type: String,
    required: true,
  },
  admin: {
    type: Boolean,
    default: false,
  }
});

const User = mongoose.model( name: 'User', schema: userSchema);

module.exports = User;
```

Hình 7. Mã nguồn của collection users

### 3.3.3. API Endpoint

#### POST/api/trips

Tạo một chuyến đi mới trong collection “trips” với các thông tin được cung cấp bởi người dùng.

```
//../server/controller/controller.js
var CreateTrip = require('../model/model');

// tạo và Lưu chuyến đi
exports.create = (req,res)=>{

  if(!req.body){
    res.status(400).send({ message : "Nội dung không được rỗng!"});
    return;
  }

  // new trips
  const trips = new CreateTrip({
    noi_xuat_phat : req.body.noi_xuat_phat,
    ngay_xuat_phat : req.body.ngay_xuat_phat,
    gio_xuat_phat: req.body.gio_xuat_phat,
    noi_den : req.body.noi_den,
    ngay_den : req.body.ngay_den,
    gio_den: req.body.gio_den,
    tong_so_cho: req.body.tong_so_cho,
    so_cho_trong: req.body.so_cho_trong,
    sdt_lien_he: req.body.sdt_lien_he,
  })

  // Lưu chuyến đi vào database
  trips
    .save(trips)
    .then(data => {
      res.redirect('/add-trip');
    })
    .catch(err =>{
      res.status(500).send({
        message : err.message || "Đã xảy ra lỗi trong khi tạo thao tác tạo"
      });
    });
}
```

Hình 8. Mã nguồn của API tạo chuyến đi

### PUT/api/trips/:id

Cập nhật chuyến xe dựa trên id chuyến xe, phải đảm bảo rằng người cập nhật chuyến xe phải là người tạo ra chuyến xe hoặc có quyền phù hợp.

```
// Cập nhật chuyến xe bằng id
exports.update = (req, res) => {
  if(!req.body){
    return res
      .status(400)
      .send({ message : "Dữ liệu cập nhật không được rỗng"})
  }

  const id = req.params.id;
  CreateTrip.findByIdAndUpdate(id, req.body, { useFindAndModify: false})
    .then(data => {
      if(!data){
        res.status(404).send({ message : `Không thể cập nhật chuyến xe với ${id}. Có thể không tìm thấy chuyến đi!` })
      }else{
        res.send(data)
      }
    })
    .catch(err => {
      res.status(500).send({ message : "Lỗi Thông Tin Chuyến Xe!" })
    })
}
}
```

Hình 9. Mã nguồn của API update chuyến đi

## GET/api/trips

Tìm thông tin chuyến đi dựa trên id.

```
//tìm chuyến xe theo id
exports.find = (req, res) => {
  if(req.query.id){
    const id = req.query.id;
    CreateTrip.findById(id)
      .then(data => {
        if(!data){
          res.status(404).send({ message : "Không tìm thấy chuyến xe có id " + id})
        }else{
          res.send(data)
        }
      })
      .catch(err => {
        res.status(500).send({ message: "Lỗi truy xuất chuyến đi với id " + id})
      })
  }else{
    CreateTrip.find()
      .then(trips => {
        res.send(trips)
      })
      .catch(err => {
        res.status(500).send({ message : err.message || "Đã xảy ra lỗi khi truy xuất thông tin chuyến đi" })
      })
  }
}
}
```

Hình 10. Mã nguồn của API tìm chuyến đi

## **DELETE/api/trips/:id**

Xóa chuyến đi dựa trên id, phải đảm bảo rằng chỉ người tạo ra chuyến đi mới được xóa.

```
// Xóa chuyến xe với id request
exports.delete = (req, res) => {
  const id = req.params.id;
  CreateTrip.findByIdAndDelete(id)
    .then(data => {
      if(!data){
        res.status(404).send({ message : `Không thể xóa chuyến xe với ${id}. Có thể không tìm thấy chuyến đi!` })
      }else{
        res.send({
          message : "Đã xóa chuyến xe thành công!!"
        })
      }
    })
    .catch(err =>{
      res.status(500).send({
        message: "Không thể xóa chuyến xe có id=" + id
      });
    });
};
```

Hình 11. Mã nguồn của API xóa chuyến đi

## **POST/api/login**

Xử lý các yêu cầu đăng nhập của người dùng

```

//./server/controller/authController.js
var User = require('../model/userDB');
const bcrypt = require('bcrypt');
const jwt = require('jsonwebtoken'); 44k (gzipped: 12.9k)

const authController = {

loginUser: async (req, res) => {
  try {
    const user = await User.findOne({ username: req.body.username });
    if (!user) {
      return res.status(404).json("Sai username!");
    }
    const validPassword = await bcrypt.compare(
      data: req.body.password,
      encrypted: user.password
    );
    if (!validPassword) {
      return res.status(404).json("Sai password");
    }
    if (user && validPassword){
      const accessToken = jwt.sign({ payload: {
        id: user.id,
        admin: user.admin
      }},
        secretOrPrivateKey: process.env.JWT_SECRET,
        options: { expiresIn: "20d" }
      );
      // Chỉ chuyển hướng nếu cả hai điều kiện đều đúng
      res.status(200).json({user, accessToken});
    }
  } catch (err) {
    console.log( message: err);
    res.status(500).json(err);
  }
}
}
module.exports = authController;

```

Hình 12. Mã nguồn của API login

## CHƯƠNG 4 KẾT QUẢ NGHIÊN CỨU

### 4.1 Sản phẩm đạt được sau quá trình nghiên cứu

Ứng dụng chia sẻ xe cho giảng viên được xây dựng thành công và đáp ứng được các yêu cầu đề ra, đầy đủ các chức năng thêm, sửa, xóa chuyển xe. Ứng dụng có khả năng xử lý nhanh chóng, giao diện thân thiện và dễ sử dụng.

### 4.2 Về hiệu năng

Node.js sử dụng mô hình xử lý không đồng bộ, giúp xử lý nhiều yêu cầu cùng một lúc mà không làm chậm quá trình thực hiện. Điều này rất hữu ích khi có nhiều người dùng thực hiện các thao tác đồng thời trên ứng dụng.

Bên cạnh đó việc tối ưu hiệu suất của ứng dụng cũng nhờ vào ExpressJS, là một framework nhẹ nhàng và nhanh chóng.

MongoDB là một cơ sở dữ liệu NoSQL, có thể cung cấp hiệu suất cao cho các truy vấn đồng thời và thêm vào đó, nó tương thích tốt với JSON-style data, phù hợp cho các ứng dụng web hiện đại.

Nhờ vào các yếu tố trên, ứng dụng chia sẻ xe cho giảng viên hoạt động một cách mượt mà và trơn tru, thời gian runtime của các tác vụ chưa đến 1 giây.

#### **4.3 Về trải nghiệm người dùng**

Giao diện rất thân thiện, tối ưu hóa việc hiển thị thông tin của các chuyến xe, nên người dùng có thể dễ dàng tiếp cận.

Hiển thị đầy đủ và dễ dàng nắm bắt thông tin của các chuyến xe.

Dễ dàng đăng ký chuyến xe.

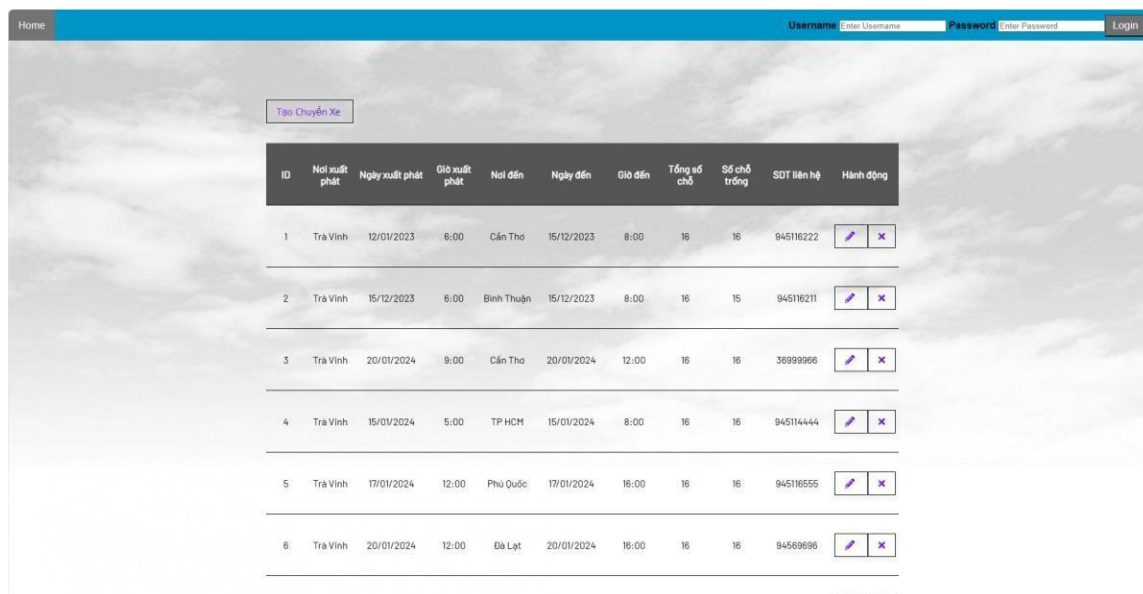
#### **4.4 Kết quả thử nghiệm**

Ứng dụng chạy thành công trên localhost với cổng 3001, chạy thử nghiệm trên máy hệ điều hành window 10 với cấu hình Intel i3-10100f, ram 16GB, GPU radeon RX 570 series, mainboard H410M-A Pro.













##### **Giao diện chính**

Trang hiển thị đầu tiên của hệ thống bao gồm thông tin của tất cả chuyến xe đang có và form đăng nhập. Giảng viên chỉ được xem thông tin của các chuyến xe sau đó liên hệ qua số điện thoại của chuyến xe đó.





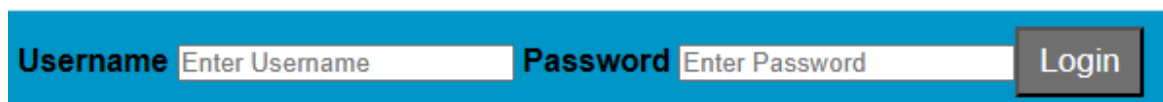
The screenshot shows a web application interface. At the top, there is a navigation bar with a 'Home' link and a login section containing 'Username' and 'Password' input fields with placeholder text 'Enter Username' and 'Enter Password', and a 'Login' button. Below the navigation bar, there is a button labeled 'Tạo Chuyến Xe'. The main content area features a table with the following columns: ID, Nơi xuất phát, Ngày xuất phát, Giờ xuất phát, Nơi đến, Ngày đến, Giờ đến, Tổng số chỗ, Số chỗ trống, SĐT liên hệ, and Hành động. The table contains six rows of data representing different vehicle trips.

ID	Nơi xuất phát	Ngày xuất phát	Giờ xuất phát	Nơi đến	Ngày đến	Giờ đến	Tổng số chỗ	Số chỗ trống	SĐT liên hệ	Hành động
1	Trà Vinh	12/01/2023	6:00	Cần Thơ	15/12/2023	8:00	16	16	945116222	 
2	Trà Vinh	15/12/2023	6:00	Bình Thuận	15/12/2023	8:00	16	15	945116211	 
3	Trà Vinh	20/01/2024	9:00	Cần Thơ	20/01/2024	12:00	16	16	36999966	 
4	Trà Vinh	15/01/2024	5:00	TP HCM	15/01/2024	8:00	16	16	945114444	 
5	Trà Vinh	17/01/2024	12:00	Phù Quốc	17/01/2024	16:00	16	16	945116555	 
6	Trà Vinh	20/01/2024	12:00	Đà Lạt	20/01/2024	16:00	16	16	94569696	 

Hình 13. Giao diện chính

### Giao diện form đăng nhập

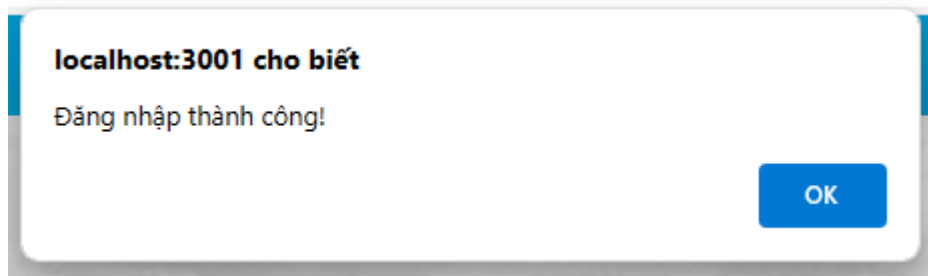
Tài khoản để đăng nhập sẽ được admin cung cấp, nếu nhập đúng username và password sẽ đăng nhập thành công, ngược lại nếu nhập sai thì sẽ đăng nhập thất bại.



The screenshot shows a login form with a blue header. It contains two input fields: 'Username' with placeholder text 'Enter Username' and 'Password' with placeholder text 'Enter Password'. To the right of these fields is a 'Login' button.

Hình 14. Form đăng nhập

### Thông báo đăng nhập thành công



Hình 15. Thông báo đăng nhập thành công

#### 4.4.1 Chức năng Tạo chuyến xe

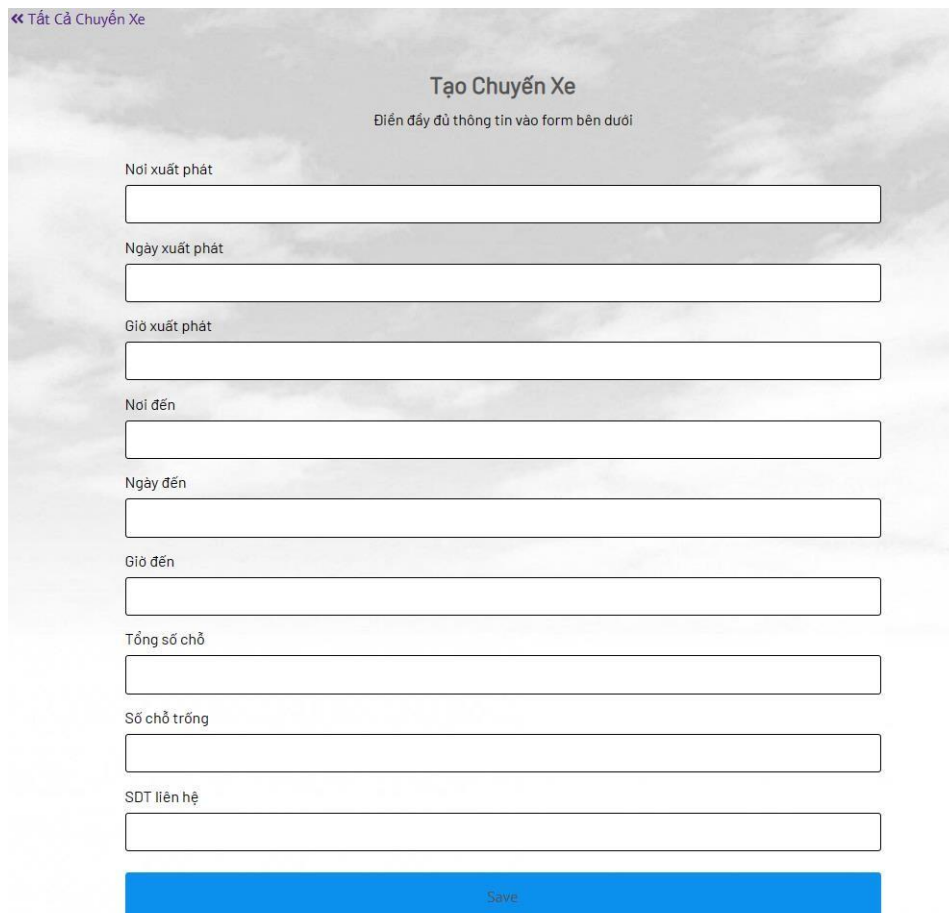
Tạo chuyến xe Chức năng tạo chuyến xe giúp người dùng tạo chuyến xe mới. Khi nhấn vào nút tạo chuyến xe ở giao diện chính sẽ dẫn đến trang tạo chuyến xe.



Hình 16. Nút tạo chuyến xe

## Giao diện tạo chuyến xe

Ở giao diện tạo chuyến xe, sẽ có các ô nhập thông tin của chuyến xe người dùng cần nhập tất cả thông tin của chuyến xe sau đó nhấn nút Save, sau khi nhấn nút Save sẽ hiện ra một thông báo tạo chuyến xe thành công.



« Tất Cả Chuyến Xe

### Tạo Chuyến Xe

Điền đầy đủ thông tin vào form bên dưới

Nơi xuất phát

Ngày xuất phát

Giờ xuất phát

Nơi đến

Ngày đến

Giờ đến

Tổng số chỗ

Số chỗ trống

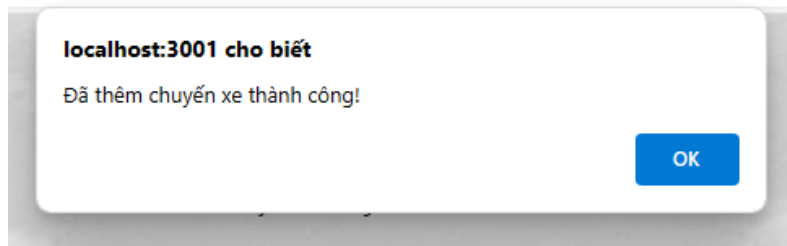
SDT liên hệ

Save

Hình 17. Giao diện tạo chuyến xe

## Thông báo tạo chuyến xe thành công



Sau khi tạo chuyến xe thành công, trở về giao diện chính sẽ thấy được chuyến xe vừa tạo.



Hình 18. Thông báo tạo chuyến xe thành công

#### 4.4.2 Chức năng Cập nhật chuyến xe

##### Nút cập nhật chuyến xe

ID	Nơi xuất phát	Ngày xuất phát	Giờ xuất phát	Nơi đến	Ngày đến	Giờ đến	Tổng số chỗ	Số chỗ trống	SĐT liên hệ	Hành động
1	Trà Vinh	12/01/2023	6:00	Cần Thơ	15/12/2023	8:00	16	16	945116222	 

Hình 19. Nút cập nhật chuyến xe

##### Giao diện cập nhật chuyến xe

Chức năng cập nhật chuyến xe giúp người dùng chỉnh sửa mọi thông tin của chuyến xe. Sau khi chỉnh sửa thông tin cần thiết và nhấn nút lưu, sẽ hiện thông báo cập nhật chuyến xe thành công.

« Tất Cả Chuyến Xe

### Cập Nhật Chuyến Xe

Hãy điền thông tin cập nhật cho chuyến xe

Nơi xuất phát  
Trà Vinh

Ngày xuất phát  
12/01/2023

Giờ xuất phát  
6:00

Nơi đến  
Cần Thơ

Ngày đến  
15/12/2023

Giờ đến  
8:00

Tổng số chỗ  
16

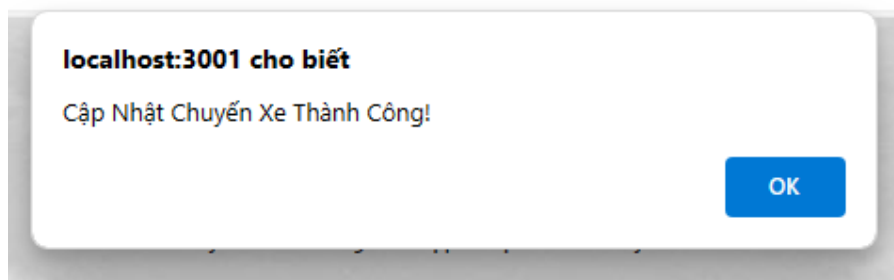
Số chỗ trống  
16

SDT liên hệ  
945116222

Cập Nhật

Hình 20. Giao diện cập nhật chuyến xe



### Thông báo cập nhật chuyến xe thành công



Hình 21. Thông báo cập nhật chuyến xe thành công

### Chức năng xóa chuyến xe

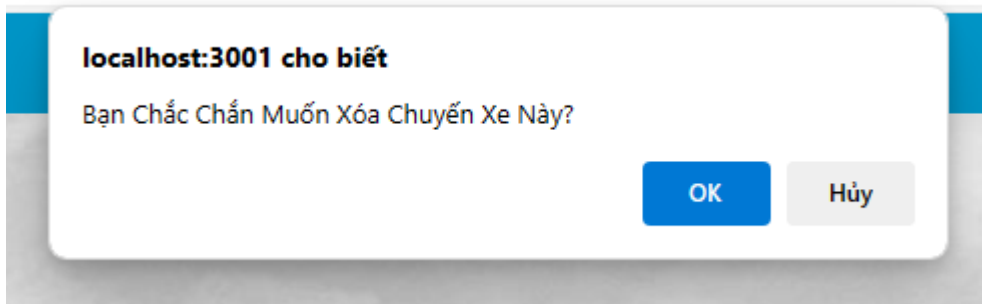
Khi nhấn vào nút xóa của chuyến xe bất kì sẽ hiện ra một thông báo để xác nhận rằng bạn đã chắc chắn xóa chuyến xe hay chưa.

ID	Nơi xuất phát	Ngày xuất phát	Giờ xuất phát	Nơi đến	Ngày đến	Giờ đến	Tổng số chỗ	Số chỗ trống	SDT liên hệ	Hành động
1	Trà Vinh	12/01/2023	6:00	Cần Thơ	15/12/2023	8:00	16	16	945116222	 

Hình 22. Nút xóa chuyến xe

### Thông báo xác nhận xóa chuyến xe

Sao khi xác nhận thì chuyến xe sẽ được xóa khỏi danh sách ban đầu.



Hình 23. Thông báo xác nhận xóa chuyến xe

## CHƯƠNG 5 KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

### 5.1 Kết luận

Sao quá trình nghiên cứu và thực hiện dự án ứng dụng chia sẻ xe cho giảng viên, em đã tiếp thu được nhiều kiến thức mới như là cách làm việc với NodeJS, ExpressJS và cơ sở dữ liệu NoSQL đó là MongoDB, điều này giúp em mở rộng được vốn kiến thức và tích lũy được nhiều kinh nghiệm quý báu, tạo tiền đề cho những dự án sau này có thể thực hiện tốt hơn.

#### 5.1.1 Kết quả đạt được

Ứng dụng chia sẻ xe cho giảng viên được xây dựng thành công và đáp ứng được các yêu cầu đề ra, có đầy đủ các chức năng thêm sửa xóa chuyển đi. Ứng dụng có khả năng xử lý nhanh chóng, giao diện thân thiện và dễ sử dụng.

#### 5.1.2 Những đóng góp mới

Ứng dụng chia sẻ xe cho giảng viên là một ứng dụng cực kì hữu ích, có thể giúp các giảng viên dễ dàng đăng ký chuyển xe cho chuyển đi của mình, giúp nhà trường giảm được chi phí đi lại của giảng viên. Ứng dụng có những đóng góp mới sau:

Sử dụng các công nghệ hiện đại như NodeJS, ExpressJS, MongoDB, giúp ứng dụng có khả năng xử lý nhanh chóng và có khả năng mở rộng cao.

Thiết kế giao diện đơn giản, thân thiện và dễ sử dụng, giúp người dùng dễ dàng tìm thấy các chức năng cần thiết.

#### 5.1.3 Những đề xuất mới

Để nâng cao hiệu quả của ứng dụng, có thể thực hiện một số đề xuất sau:

Thêm chức năng đăng kí và đăng nhập dành cho người dùng, để người dùng có thể xem được trạng thái của chuyến xe vừa đăng ký và xem được lịch sử chuyến xe đã từng đi.

Thêm chức năng tìm kiếm chuyến đi theo địa điểm hoặc theo ngày và các khung giờ, để người dùng có thể dễ dàng tìm thấy chuyến xe mong muốn.

Tối ưu hóa giao diện để người dùng để đảm bảo trải nghiệm người dùng một cách mượt mà nhất.

## **5.2 Hướng phát triển**

Ứng dụng chia sẻ xe cho giảng viên có thể được phát triển thêm theo các hướng sau:

Sử dụng các công cụ như React, Angular, hoặc Vue.js để tạo giao diện động.  
Cải thiện khả năng bảo mật của ứng dụng.

Tối ưu hóa hiệu năng của ứng dụng để có thể xử lý được nhiều yêu cầu hơn.

Cung cấp kênh hỗ trợ cho người dùng để giải quyết vấn đề và đảm bảo họ có trải nghiệm tốt nhất.

Triển khai ứng dụng trên môi trường sản xuất.

Việc phát triển ứng dụng theo các hướng trên sẽ giúp ứng dụng đáp ứng được nhu cầu sử dụng ngày càng cao của người dùng.



## DANH MỤC TÀI LIỆU THAM KHẢO

[1] CRUD với  
NodeJs, Express và  
MongoDB |  
Meditation Coding  
(huongvnq.github.io)

[2] Express.js And  
MongoDB REST  
API Tutorial |  
MongoDB

[3] Lập trình web với  
Nodejs, Express,  
MongoDB  
(gitiho.com)

[4] CRUD Là Gì?  
Lập Trình Website  
CRUD? CRUD  
Database?  
(viblo.asia)

[5] About Node.js® |  
Node.js (nodejs.org)