**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC TÂY ĐÔ**

**KHOA KỸ THUẬT – CÔNG NGHỆ**

A logo with a green and pink leaf

Description automatically generated

**Trí tuệ - Sáng tạo - Năng động – Đổi mới**

**NIÊN LUẬN 2**

**NỀN TẢNG GIAO DỊCH XE MÁY**

**CHUYÊN NGÀNH**

**CÔNG NGHỆ THÔNG TIN**

**SINH VIÊN THỰC HIỆN**

**QUÁCH TRƯỜNG PHÚC - 227060168**

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC TÂY ĐÔ**

**KHOA KỸ THUẬT – CÔNG NGHỆ**

A logo with a green and pink leaf

Description automatically generated

**Trí tuệ - Sáng tạo - Năng động – Đổi mới**

**NIÊN LUẬN 1**

**NỀN TẢNG GIAO DỊCH XE MÁY**

**CHUYÊN NGÀNH**

**CÔNG NGHỆ THÔNG TIN**

**SINH VIÊN THỰC HIỆN**

**QUÁCH TRƯỜNG PHÚC - 227060168**

**GIÁO VIÊN HƯỚNG DẪN**

**THS. NGUYỄN CHÍ CƯỜNG**

**MỤC LỤC**

[MỤC LỤC HÌNH ẢNH iii](#_Toc195430372)

[LỜI CẢM ƠN iv](#_Toc195430373)

[LỜI CAM ĐOAN v](#_Toc195430374)

[TÓM TẮT vi](#_Toc195430375)

[ABSTRACT vii](#_Toc195430376)

[CHƯƠNG I. MỞ ĐẦU 1](#_Toc195430377)

[**1.1. Đặt vấn đề** 1](#_Toc195430378)

[**1.2. Mục tiêu cần đạt được** 1](#_Toc195430379)

[**1.3. Phạm vi nghiên cứu** 2](#_Toc195430380)

[**1.3.1. Thị trường xe máy trực tuyến** 2](#_Toc195430381)

[**1.3.2. Công nghệ phát triển hệ thống** 2](#_Toc195430382)

[**1.3.3. Quản lý sản phẩm và giao dịch** 2](#_Toc195430383)

[**1.3.4. Tìm kiếm và lọc sản phẩm** 2](#_Toc195430384)

[**1.3.5. Người dùng và tối ưu hoá** 3](#_Toc195430385)

[**1.3.6. Phạm vi địa lý và mở rộng nền tảng** 3](#_Toc195430386)

[**1.4. Phương pháp nghiên cứu** 3](#_Toc195430387)

[**1.4.1 Nghiên cứu tài liệu và cơ sở lý thuyết** 3](#_Toc195430388)

[**1.4.2 Phân tích và khảo sát hệ thống thực tế** 3](#_Toc195430389)

[**1.4.3 Thiết kế hệ thống và mô hình dữ liệu** 3](#_Toc195430390)

[**1.4.4 Kiểm thử và đánh giá** 3](#_Toc195430391)

[**1.5. Hướng giải quyết** 4](#_Toc195430392)

[**1.6. Kế hoạch thực hiện** 5](#_Toc195430393)

[CHƯƠNG II. CƠ SỞ LÝ THUYẾT 6](#_Toc195430394)

[**2.1. Mô hình kinh doanh C2C** 6](#_Toc195430395)

[**2.2. Định nghĩa Website** 7](#_Toc195430396)

[**2.3. Công nghệ được sử dụng** 8](#_Toc195430397)

[**2.2.1. Môi trường phát triển** 8](#_Toc195430398)

[**2.2.2. Công cụ phát triển** 8](#_Toc195430399)

[**2.2.3. Giao diện người dùng** 9](#_Toc195430400)

[**2.2.4. Xử lý dữ liệu** 10](#_Toc195430401)

[**2.2.5. Cơ sở dữ liệu** 11](#_Toc195430402)

[**2.2.6. Quản lý thư viện và gói phần mềm** 11](#_Toc195430403)

[CHƯƠNG III. PHÂN TÍCH HỆ THỐNG 13](#_Toc195430404)

[**3.1. Quản lý dữ liệu** 13](#_Toc195430405)

[**3.2. Thiết kế chức năng** 14](#_Toc195430406)

[**3.2.1. Đăng ký và đăng nhập** 14](#_Toc195430407)

[**3.2.1.1. Xử lý đăng ký** 14](#_Toc195430408)

[**3.2.2. Chỉnh sửa thông tin người dùng** 15](#_Toc195430409)

[**3.2.3. Đăng bán xe** 15](#_Toc195430410)

[**3.2.4. Chỉnh sửa thông tin xe đang bán** 15](#_Toc195430411)

[**3.2.5. Tìm kiếm và lọc xe** 15](#_Toc195430412)

[**3.2.6. Giao dịch** 15](#_Toc195430413)

[**3.2.7. Nhắn tin** 15](#_Toc195430414)

[**3.2.8. Đánh giá uy tín** 15](#_Toc195430415)

[CHƯƠNG V. GIAO DIỆN 15](#_Toc195430416)

[KẾT LUẬN – ĐÁNH GIÁ 15](#_Toc195430417)

[TÀI LIỆU THAM KHẢO 17](#_Toc195430418)

**MỤC LỤC HÌNH ẢNH**

[Hình 2. 1 Đồ thị trong thực tế 4](#_Toc185005247)

[Hình 2. 2 Ví dụ về đơn đồ thị vô hướng 4](#_Toc185005248)

[Hình 2. 3 Ví dụ về đa đồ thị vô hướng 5](#_Toc185005249)

[Hình 2. 5 Ví dụ về bậc định, đỉnh cô lập, đỉnh treo 6](#_Toc185005251)

[Hình 2. 7 Ví dụ về đường đi ,chu trình của đồ thị 7](#_Toc185005253)

[Hình 2. 8 Ví dụ về đồ thị liên thông 8](#_Toc185005254)

[Hình 2. 9 Ví dụ về các đỉnh cắt, cạnh cầu 8](#_Toc185005255)

[Hình 2. 10 Ví dụ về đồ thị liên thông mạnh, yếu 9](#_Toc185005256)

[Hình 2. 11 Đồ thị G 10](#_Toc185005257)

[Hình 2. 12 Kết quả ma trận kề của đồ thị G 11](#_Toc185005258)

[Hình 2. 13 Đồ thị ban đầu 13](#_Toc185005259)

[Hình 2. 14 Bước đầu tiên 13](#_Toc185005260)

[Hình 2. 15 Bước thứ hai 14](#_Toc185005261)

[Hình 2. 16 Bước thứ ba 14](#_Toc185005262)

[Hình 2. 17 Bước thứ bốn 14](#_Toc185005263)

[Hình 2. 18 Kết thúc 15](#_Toc185005264)

[Hình 2. 19 Lưu đồ thuật toán DFS 16](#_Toc185005265)

[Hình 2. 20 Bài toán 7 cây cầu 17](#_Toc185005266)

[Hình 2. 21 Đồ thị G1, G2, G3 18](#_Toc185005267)

[Hình 2. 22 Bước đầu tiên 19](#_Toc185005268)

[Hình 2. 23 Bước thứ hai 19](#_Toc185005269)

[Hình 2. 24 Bước thứ ba 20](#_Toc185005270)

[Hình 2. 25 Bước thứ tư 20](#_Toc185005271)

[Hình 2. 26 Bước thứ năm 20](#_Toc185005272)

[Hình 2. 27 Bước thứ sáu 21](#_Toc185005273)

[Hình 2. 28 Lưu đồ giải thuật Fleury 21](#_Toc185005274)

[Hình 3. 1 Giao diện chính 30](#_Toc185005275)

[Hình 3. 2 Giao diện chức năng này 30](#_Toc185005276)

[Hình 3. 3 Chức năng chọn tệp tin 30](#_Toc185005277)

[Hình 3. 4 Chức năng làm mới tệp 31](#_Toc185005278)

[Hình 3. 5 Trở về trang chủ 31](#_Toc185005279)

[Hình 3. 6 Kết quả sau khi tải lên 32](#_Toc185005280)

[Hình 3. 7 Giao diện chính của chức năng này 33](#_Toc185005281)

[Hình 3. 8 Nhận số đỉnh 33](#_Toc185005282)

[Hình 3. 9 Tạo ra các ô nhập dữ liệu 33](#_Toc185005283)

[Hình 3. 10 Nhập đỉnh kết nối vào các ô nhập liệu 34](#_Toc185005284)

[Hình 3. 11 Ví dụ khi nhập dữ liệu 34](#_Toc185005285)

[Hình 3. 12 Thêm đỉnh muốn kết nối 34](#_Toc185005286)

[Hình 3. 13 Sau khi nhấn “Thêm đỉnh” 35](#_Toc185005287)

[Hình 3. 14 Vẽ ra đồ thị mong muốn 35](#_Toc185005288)

[Hình 3. 15 Kết quả 36](#_Toc185005289)

[Hình 3. 16 Giao diện chính 37](#_Toc185005290)

[Hình 3. 17 Tiến hành tạo đồ thị ngẫu nhiên 37](#_Toc185005291)

[Hình 3. 18 Kết quả khi được tạo ngẫu nhiên 38](#_Toc185005292)

**LỜI CẢM ƠN**

Lời đầu tiên, tôi xin gửi lời cảm ơn chân thành tới Thầy Nguyễn Chí Cường, người đã tận tình hướng dẫn và chỉ bảo tôi trong suốt quá trình thực hiện bài niên luận này. Thầy đã giúp tôi vượt qua nhiều khó khăn và cung cấp những kiến thức quý báu, góp phần quan trọng vào việc hoàn thành nghiên cứu này. Bên cạnh đó, tôi cũng xin bày tỏ lòng biết ơn sâu sắc tới các thầy cô trong Khoa Kỹ Thuật Công Nghệ, Trường Đại học Tây Đô, vì đã tạo điều kiện thuận lợi trong quá trình học tập và nghiên cứu. Cảm ơn gia đình, cha mẹ và những người thân yêu đã luôn động viên, khích lệ tôi cả về tinh thần lẫn vật chất trong suốt chặng đường học tập. Cuối cùng, tôi cũng xin cảm ơn bạn bè đã luôn hỗ trợ và đóng góp ý kiến quý báu, giúp tôi hoàn thiện bài niên luận này. Những lời động viên và sự giúp đỡ của mọi người chính là nguồn động lực to lớn giúp tôi hoàn thành tốt công việc nghiên cứu. Xin chân thành cảm ơn!

**LỜI CAM ĐOAN**

Tôi xin cam đoan rằng Niên luận này là do chính tôi thực hiện, không sao chép dưới bất kỳ hình thức nào hay thuê hoặc nhờ người khác thực hiện.

Dữ liệu và kết quả phân tích trong Niên luận đảm bảo tính chính xác, khách quan và trung thực, không có bất kỳ sự ngụy tạo và điều chỉnh kết quả nghiên cứu bằng sự chủ quan của tác giả.

*Cần Thơ, ngày…..tháng……năm 20…*

Sinh viên thực hiện

*(Ký tên và ghi rõ họ và tên)*

**Quách Trường Phúc**

**TÓM TẮT**

RevNow là một nền tảng giao dịch trực tuyến chuyên nghiệp, được thiết kế với mục tiêu xây dựng một thị trường mở, minh bạch và đáng tin cậy dành riêng cho lĩnh vực mua bán xe máy – bao gồm cả xe mới lẫn xe đã qua sử dụng. Nền tảng này không chỉ phục vụ cho người dùng cá nhân mà còn hỗ trợ các cửa hàng, đại lý và doanh nghiệp dễ dàng tiếp cận người mua tiềm năng thông qua hệ thống đăng tin hiện đại và hiệu quả.

Với giao diện được thiết kế cẩn thận hiện đại, dễ sử dụng phù hợp với mọi thiết bị. Cho phép người bán đăng tin chi tiết về chiếc xe của mình, bao gồm thông tin thương hiệu, mẫu mã, phiên bản, tình trạng xe mới hay đã qua sử dụng, quãng đường mà xe đã đi, năm sản xuất của chiếc xe và các yếu tố pháp lý như có giấy tờ hay không, Và tải lên các hình ảnh thực tế của chiếc xe đó để tăng độ tin cậy và giúp người mua có cái nhìn tổng quát hơn về chiếc xe đó.

Người dùng có thể tìm kiếm chiếc xe ưu thích hoặc hợp nhu cầu của mình thông qua chức năng lọc xe máy của hệ thống. Chức năng lọc được thiết kế để lọc ra các xe hiển thị ra theo các thông tin cơ bản như hãng xe, loại xe, mẫu xe, phiên bản xe, tình trạng xe đó và các thông tin nâng cao như giá thành từ bao nhiêu đến bấy nhiêu, số quảng đường đã đi, số năm sản xuất, kích thước động cơ.

Các người dùng có thể giao tiếp với nhau thông qua hệ thống nhắn tin tích hợp, cho phép cả người mua lẫn người bán dễ dàng trao đổi trực tiếp với nhau hơn, từ đó rút ngắn thời gian giao dịch và tăng tính tương tác của cộng đồng.

**ABSTRACT**

RevNow is a professional online trading platform specifically developed to establish an open, transparent, and trustworthy marketplace for motorbike transactions, encompassing both brand-new and pre-owned vehicles. The platform caters not only to individual users but also extends its utility to dealerships, showrooms, and businesses, enabling them to reach potential buyers with ease through a streamlined and effective listing system.

The interface is thoughtfully designed to be modern, intuitive, and fully responsive across various devices. Sellers are guided through a straightforward process to post detailed listings of their vehicles, including essential information such as brand, model, version, condition (new or used), mileage, year of manufacture, and legal documentation status. To enhance trust and offer buyers a comprehensive view of the vehicle, uploading real-life images is strongly encouraged.

Buyers can conveniently search for vehicles that match their preferences or needs using an advanced filtering system. This system enables users to narrow down their search based on key attributes such as manufacturer, model type, version, and condition, as well as more refined criteria including price range, mileage, production year, and engine size.

To further facilitate the transaction process, RevNow integrates a real-time messaging feature, allowing seamless communication between buyers and sellers. This not only accelerates negotiations but also fosters a more interactive and engaging experience within the user community.

**ĐÁNH GIÁ KẾT QUẢ THỰC HIỆN**

**NIÊN LUẬN, TIỂU LUẬN, KHOÁ LUẬN**

(Học kỳ: II, Năm 2025)

**TÊN ĐỀ TÀI: Nền tảng giao dịch xe máy**

**GIÁO VIÊN HƯỚNG DẪN:**

|  |  |  |
| --- | --- | --- |
| STT | HỌ VÀ TÊN | MSCB |
| 1 | Nguyễn Chí Cường |  |

**SINH VIÊN THỰC HIỆN:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | HỌ VÀ TÊN | MSSV | THƯỞNG  *(Tối đa 1,0 điểm)* | ĐIỂM |
| 1 | Quách Trường Phúc | 227060168 |  |  |

|  |  |
| --- | --- |
| **I. HÌNH THỨC** *(Tối đa 1,5 điểm)* |  |
| **Bìa** *(tối đa 0,5 điểm)* |  |
| * Đúng mẫu 0,5 điểm |  |
| **Bố cục** *(tối đa 1 điểm)* |  |
| * Nhận xét của giảng viên hướng dẫn và giảng viên chấm * Mục lục: cấu trúc chương, mục và tiểu mục * Phụ lục (nếu có) * Tài liệu tham khảo * …………….   **Phải đúng theo quy định** *1* điểm |  |
| **II. NỘI DUNG** *(Tối đa 5 điểm)* |  |
| **Giới thiệu** *(tối đa 0.5 điểm)* |  |
| * Mô tả bài toán * Mục tiêu cần đạt, hướng giải quyết |  |
| **Lý thuyết** *(tối đa 1.0 điểm)* |  |
| * Các khái niệm * Các phương pháp * Kết quả nghiên cứu lý thuyết vận dụng vào đề tài |  |
| **Ứng dụng** *(tối đa 3 điểm)* |  |
| * Đặc tả hệ thống (0.5 điểm) * Các mô hình (MCD, MLD, PDM) và các ràng buộc toàn vẹn (1 điểm) * DFD, Sơ đồ chức năng, lưu đồ giải thuật giải quyết vấn đề (nếu có), diễn giải ý nghĩa các màn hình cập nhật (FORM) và các báo biểu (REPORT) (1 điểm) * Giới thiệu chương trình (0,5 điểm) |  |
| **Kết luận** *(tối đa 0,5 điểm)* |  |
| * Nhận xét kết quả đạt được * Hạn chế * Hướng phát triển |  |
| **III. CHƯƠNG TRÌNH DEMO** *(Tối đa 3.5 điểm)* |  |
| **Giao diện thân thiện với người dùng** *(0.5 điểm)* |  |
| **Hướng dẫn sử dụng** *(0.5 điểm)* |  |
| **Kết quả thực hiện đúng với kết quả của phần ứng dụng**   * Giải thuật đúng, thực thi chính xác, cài đặt CSDL cùng với các ràng buộc toàn vẹn, Forms cập nhật đúng, Reports hợp lý,…. *(2.5 điểm)* |  |

**Ghi chú:**

1. *Điểm trong khung “sinh viên thực hiện” là điểm kết quả cuối cùng của từng sinh viên trong quá trình thực hiện* Niên luận…/tiểu luận/khoá luận*.*
2. *Nếu sinh viên demo chương trình và trả lời vấn đáp không đạt yêu cầu của giáo viên hướng dẫn thì sinh viên sẽ nhận điểm F cho học phần này.*

Cần Thơ, ngày .......... tháng năm 20.

GIÁO VIÊN CHẤM

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

🖎🕮✍

.....................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................

|  |
| --- |
| *Cần Thơ, ngày tháng năm 2025* |
| **Giáo viên hướng dẫn** |
| **………………..** |

# CHƯƠNG I. MỞ ĐẦU

1. **Đặt vấn đề**

Trong bối cảnh công nghệ số phát triển mạnh mẽ, nhu cầu giao dịch mua bán trực tuyến ngày càng trở nên phổ biến và thiết yếu trong nhiều lĩnh vực khác nhau. Tuy nhiên, thị trường mua bán xe máy – một trong những ngành hàng có giá trị lớn và phổ biến tại Việt Nam – vẫn còn tồn tại nhiều bất cập khi thực hiện các giao dịch truyền thống như: thiếu minh bạch trong thông tin sản phẩm, rủi ro trong giao dịch trực tiếp, khó khăn trong việc xác minh danh tính người mua/bán, cũng như sự thiếu vắng một nền tảng chuyên biệt, uy tín để kết nối các bên.

Hiện nay, đa số các giao dịch xe máy cũ hoặc mới thường được thực hiện thông qua các nhóm mạng xã hội, diễn đàn hoặc chợ trực tuyến tổng hợp – nơi mà các tính năng hỗ trợ mua bán còn đơn giản, không phù hợp cho loại hình sản phẩm có tính pháp lý và kỹ thuật như xe máy. Điều này dẫn đến việc người mua khó xác minh được độ tin cậy của người bán, thông tin sản phẩm thường không đầy đủ, kém chất lượng hoặc không chính xác, từ đó tiềm ẩn nhiều rủi ro và gây mất niềm tin trong cộng đồng người dùng.

Từ thực tế đó, "RevNow – Nền tảng giao dịch xe máy trực tuyến" được xây dựng với mục tiêu tạo ra một hệ thống chuyên biệt, minh bạch, đáng tin cậy, cho phép người dùng đăng bán và tìm mua xe máy một cách dễ dàng, nhanh chóng và an toàn. RevNow không chỉ cung cấp các tính năng hỗ trợ đăng tin chi tiết, tìm kiếm nâng cao và hệ thống nhắn tin tích hợp mà còn chú trọng vào việc xác thực người dùng, đánh giá độ uy tín, đảm bảo các giao dịch diễn ra hiệu quả và minh bạch nhất có thể.

Việc xây dựng nền tảng RevNow là một bước tiến trong việc áp dụng công nghệ web hiện đại vào thực tiễn, góp phần giải quyết các vấn đề còn tồn tại trong thị trường mua bán xe máy hiện nay, đồng thời mở ra tiềm năng phát triển hệ thống trong tương lai đối với các lĩnh vực giao dịch sản phẩm có yếu tố pháp lý tương tự.

1. **Mục tiêu cần đạt được**

* Đảm bảo rằng mọi giao dịch diễn ra trên nền tảng đều minh bạch, các thông tin về sản phẩm rõ ràng và chính xác. Nền tảng phải giúp người mua và người bán dễ dàng xác thực và theo dõi các giao dịch.
* Cung cấp các tính năng cho phép người dùng đăng tin bán xe dễ dàng và chi tiết, bao gồm thông tin kỹ thuật, tình trạng xe, giá cả, và giấy tờ liên quan. Đồng thời, hệ thống tìm kiếm cần linh hoạt và mạnh mẽ, giúp người mua dễ dàng tìm kiếm sản phẩm phù hợp với các bộ lọc đa dạng như hãng xe, dòng xe, giá cả, tình trạng xe, và các tiêu chí khác.
* Tạo ra cơ chế xác thực người dùng rõ ràng và hiệu quả, giảm thiểu rủi ro gian lận, đảm bảo rằng các bên tham gia giao dịch đều là người thật và có trách nhiệm với sản phẩm hoặc dịch vụ của mình.
* Tích hợp hệ thống nhắn tin trực tuyến giúp người mua và người bán dễ dàng trao đổi thông tin, thương lượng và giải quyết các thắc mắc. Điều này không chỉ giúp nâng cao trải nghiệm người dùng mà còn đảm bảo giao dịch diễn ra thuận lợi và không bị gián đoạn.
* Cung cấp công cụ đánh giá độ uy tín của người bán sau mỗi giao dịch, giúp người mua có thể dựa vào các đánh giá này để đưa ra quyết định đúng đắn và an toàn. Đồng thời, các đánh giá sẽ giúp tăng tính minh bạch và uy tín cho nền tảng.
* Giao diện của RevNow cần được thiết kế đơn giản, dễ sử dụng, đảm bảo người dùng có thể dễ dàng thực hiện các thao tác mà không gặp phải bất kỳ khó khăn nào, từ đăng ký tài khoản cho đến giao dịch và tìm kiếm sản phẩm.
* Tạo ra cơ sở dữ liệu đáng tin cậy và hệ thống quản lý giao dịch hiệu quả, giúp theo dõi tất cả các hoạt động của người dùng và giao dịch, đồng thời cung cấp các báo cáo thống kê chính xác để hỗ trợ việc quản lý và phân tích.

1. **Phạm vi nghiên cứu**
   1. **Thị trường xe máy trực tuyến**

* Nghiên cứu về nhu cầu và thị trường xe máy trong môi trường thương mại điện tử, từ đó xác định các xu hướng tiêu dùng và các vấn đề thường gặp trong giao dịch mua bán xe máy.
* Đánh giá sự phát triển của các nền tảng giao dịch trực tuyến hiện có và xác định các điểm mạnh, yếu của chúng để cải thiện và phát triển một hệ thống tối ưu hơn.
  1. **Công nghệ phát triển hệ thống**
* Tìm hiểu về các công nghệ frontend hiện đại như React và TailwindCSS, bao gồm việc xây dựng giao diện người dùng thân thiện, dễ sử dụng, và tối ưu hóa trải nghiệm người dùng. Và nghiên cứu về Supabase để bảo mật và xác thực người dùng và triển khai hệ thống backend an toàn, hiệu quả với PostgreSQL để quản lý dữ liệu người dùng và các giao dịch mua bán xe máy.
  1. **Quản lý sản phẩm và giao dịch**
* Xây dựng và tối ưu hóa quy trình đăng bán, đăng mua sản phẩm, giúp người dùng dễ dàng thực hiện giao dịch từ khi đăng tin cho đến khi giao dịch hoàn tất.
* Đảm bảo tính minh bạch trong việc quản lý thông tin sản phẩm, các thông tin chi tiết về xe máy, và việc quản lý giao dịch giữa người mua và người bán thông qua hệ thống nhắn tin tích hợp.
  1. **Tìm kiếm và lọc sản phẩm**
* Nghiên cứu việc phát triển các tính năng tìm kiếm nâng cao và bộ lọc sản phẩm, cho phép người dùng dễ dàng tìm kiếm các sản phẩm xe máy dựa trên nhiều tiêu chí khác nhau như giá cả, hãng xe, loại xe, tình trạng, phân khối, và số km đã đi.
  1. **Người dùng và tối ưu hoá**
* Nghiên cứu và thử nghiệm các phương pháp tối ưu hóa trải nghiệm người dùng, bao gồm giao diện thân thiện, dễ sử dụng, cùng với việc tối ưu hóa tốc độ tải trang và khả năng phản hồi của hệ thống.
* Xây dựng hệ thống đánh giá độ uy tín của người bán, giúp người mua có thể đưa ra quyết định mua hàng an toàn và đáng tin cậy.
  1. **Phạm vi địa lý và mở rộng nền tảng**
* Ban đầu, nền tảng sẽ phục vụ cho thị trường xe máy trong nước, nhưng sẽ có kế hoạch mở rộng và áp dụng mô hình giao dịch cho các quốc gia khác nếu nền tảng thành công.
* Phạm vi mở rộng sẽ bao gồm việc tích hợp các ngôn ngữ khác nhau và điều chỉnh nền tảng để phù hợp với các yêu cầu và quy định địa phương của các quốc gia khác nhau.

1. **Phương pháp nghiên cứu**
2. **Nghiên cứu tài liệu và cơ sở lý thuyết**

* Tìm hiểu các tài liệu học thuật, bài báo, báo cáo và tài liệu hướng dẫn chính thức liên quan đến:
* Mô hình thương mại điện tử C2C (Customer to Customer).
* Hành vi người dùng khi mua bán xe máy trực tuyến.
* Các xu hướng thiết kế giao diện người dùng (UI/UX) trong các nền tảng hiện đại.
* Các công nghệ được sử dụng trong dự án như React, TailwindCSS, Supabase, PostgreSQL, cùng các thư viện hỗ trợ React.

1. **Phân tích và khảo sát hệ thống thực tế**

* Thực hiện khảo sát và phân tích một số nền tảng mua bán xe máy phổ biến hiện nay tại Việt Nam như Chợ Tốt, Muaban.net, OKXE, Và các nền tảng nước ngoài như Cycletrader,… nhằm:
* Rút ra điểm mạnh, điểm yếu.
* Xác định nhu cầu và hành vi người dùng.
* Tìm hiểu cách thức các nền tảng này tổ chức luồng giao dịch và giao diện người dùng.

1. **Thiết kế hệ thống và mô hình dữ liệu**

* Áp dụng các mô hình thiết kế như:
* Mô hình Use Case để xác định các chức năng chính và hành động người dùng.
* Mô hình ERD (Entity Relationship Diagram) để thiết kế cơ sở dữ liệu logic trên PostgreSQL.

1. **Kiểm thử và đánh giá**

* Kiểm thử chức năng (Functional Testing): Đảm bảo các tính năng như đăng nhập, đăng bài, tìm kiếm, nhắn tin, lọc sản phẩm,… hoạt động đúng yêu cầu.
* Kiểm thử giao diện (UI Testing): Đảm bảo tính nhất quán, thân thiện với người dùng và hoạt động tốt trên nhiều thiết bị.
* Kiểm thử hiệu năng (Performance Testing): Kiểm tra tốc độ phản hồi, khả năng xử lý dữ liệu lớn.
* Thu thập phản hồi từ người dùng thử nghiệm để tiếp tục cải tiến.

1. **Hướng giải quyết**

* Thiết kế giao diện người dùng trực quan và thân thiện, cho phép người dùng dễ dàng đăng tin, tìm kiếm, và quản lý các giao dịch. Giao diện sẽ được xây dựng trên nền tảng React kết hợp với TailwindCSS, đảm bảo tính linh hoạt, responsive và dễ sử dụng trên nhiều thiết bị. Các chức năng chính bao gồm: đăng ký/đăng nhập, đăng bán sản phẩm, tìm kiếm và lọc xe máy theo nhiều tiêu chí, cùng với hệ thống nhắn tin tích hợp để kết nối người mua và người bán.
* Sử dụng Supabase làm nền tảng backend kết hợp với cơ sở dữ liệu PostgreSQL để lưu trữ và quản lý thông tin người dùng, sản phẩm và giao dịch. Hệ thống backend sẽ đảm bảo tính bảo mật, xác thực người dùng hiệu quả (đăng nhập qua Google và email) và xử lý giao dịch nhanh chóng. Các API được xây dựng sẽ hỗ trợ các thao tác CRUD cho người dùng và sản phẩm, đồng thời tích hợp các thư viện React như @supabase/supabase-js và React Query để tối ưu hóa hiệu suất và quản lý trạng thái ứng dụng.
* Triển khai các tính năng hỗ trợ giao dịch trực tuyến hiệu quả, bao gồm:
* Hệ thống nhắn tin thời gian thực giúp người mua và người bán trao đổi thông tin, thương lượng và hẹn gặp kiểm tra xe.
* Cơ chế đánh dấu sản phẩm "đã bán" sau khi giao dịch hoàn tất, đảm bảo thông tin được cập nhật liên tục và chính xác.
* Tích hợp hệ thống đánh giá độ uy tín, cho phép người dùng phản hồi sau mỗi giao dịch, góp phần xây dựng một môi trường giao dịch an toàn và đáng tin cậy.
* Hệ thống sẽ hiển thị kết quả giao dịch và các thông tin chi tiết về sản phẩm một cách rõ ràng và dễ hiểu, hỗ trợ người dùng theo dõi lịch sử giao dịch và đánh giá sản phẩm. Các kết quả, bao gồm thông tin sản phẩm, đánh giá uy tín và trạng thái giao dịch, sẽ được lưu trữ và xuất ra dưới dạng báo cáo thống kê, giúp người dùng và quản trị viên có cái nhìn tổng quan về hoạt động của nền tảng.

1. **Kế hoạch thực hiện**

* Tuần 1: Tìm hiểu và nghiên cứu các công nghệ cần thiết cho dự án bao gồm Vite, React, TailwindCSS cho frontend, Supabase cho backend và PostgreSQL làm hệ quản trị cơ sở dữ liệu. Ngoài ra, các thư viện hỗ trợ như React Query, React Router, Axios,… cũng được xem xét và học cách tích hợp.
* Tuần 2: Tiến hành thiết kế kiến trúc tổng thể của hệ thống, phác thảo giao diện người dùng với các trang chức năng chính như: trang chủ, trang tìm kiếm, trang chi tiết sản phẩm, trang đăng bán, trang đăng nhập/đăng ký,... Đồng thời lên bố cục cơ sở dữ liệu và cách tổ chức các bảng dữ liệu trên Supabase.
* Tuần 3: Bắt đầu lập trình giao diện người dùng với React và TailwindCSS. Xây dựng các phần cơ bản như trang chủ, hệ thống điều hướng (navigation), trang đăng nhập và đăng ký người dùng.
* Tuần 4: Triển khai tính năng đăng bán sản phẩm, cho phép người dùng điền thông tin chi tiết về xe máy và tải lên hình ảnh. Xây dựng phần hiển thị danh sách sản phẩm với chức năng lọc theo các tiêu chí như hãng xe, loại xe, giá, tình trạng,...
* Tuần 5: Kết nối với Supabase để xử lý chức năng xác thực người dùng (authentication), lưu trữ và truy vấn dữ liệu sản phẩm. Tích hợp các API để hiển thị dữ liệu động từ cơ sở dữ liệu lên giao diện người dùng.
* Tuần 6: Xây dựng hệ thống nhắn tin thời gian thực giữa người mua và người bán. Phát triển chức năng đánh giá độ uy tín người dùng sau mỗi giao dịch. Bổ sung chức năng đánh dấu sản phẩm đã bán để cập nhật trạng thái sản phẩm.
* Tuần 7: Thực hiện kiểm thử hệ thống để phát hiện và sửa các lỗi phát sinh. Tối ưu hiệu năng ứng dụng, đảm bảo tính ổn định và mượt mà trong quá trình sử dụng. Kiểm tra tương thích trên nhiều thiết bị và trình duyệt.
* Tuần 8: Hoàn thiện phần báo cáo 2, trình bày lại quy trình phát triển, phân tích hệ thống, giao diện, tính năng và đánh giá kết quả đạt được. Chuẩn bị nội dung để báo cáo, và nộp sản phẩm hoàn chỉnh.

# CHƯƠNG II. CƠ SỞ LÝ THUYẾT

1. **Mô hình kinh doanh C2C**

* C2C, hay "Consumer to Consumer", là một mô hình kinh doanh cho phép các cá nhân tự mình trao đổi hàng hóa và dịch vụ thông qua các nền tảng trực tuyến. Khác với mô hình truyền thống, trong đó doanh nghiệp cung cấp sản phẩm cho người tiêu dùng, mô hình C2C tạo điều kiện cho việc giao dịch trực tiếp giữa người tiêu dùng với nhau qua các website hoặc ứng dụng chuyên dụng kết nối các cá nhân.
* **Đặc điểm của mô hình này:**
* Giao dịch trực tiếp giữa cá nhân:
* Trong mô hình này, giao dịch được thực hiện trực tiếp giữa người mua và người bán, không cần qua trung gian sản xuất hay bán lẻ. Việc trao đổi thông tin, thương lượng giá cả và thanh toán được thực hiện trực tiếp giữa các cá nhân, giúp cắt giảm chi phí và tối ưu hóa hiệu quả giao dịch.
* Nền tảng trực tuyến: Các giao dịch C2C thường diễn ra trên các sàn giao thương điện tử, trang web rao vặt hay qua các mạng xã hội. Nhờ vậy, người bán có thể tiếp cận với một lượng lớn khách hàng tiềm năng, trong khi người mua có cơ hội dễ dàng so sánh và lựa chọn sản phẩm từ nhiều nguồn khác nhau.
* Đa dạng về mặt sản phẩm: Mô hình C2C cho phép giao dịch nhiều loại sản phẩm, từ hàng mới đến hàng đã qua sử dụng, từ hàng thủ công đến sản phẩm công nghiệp. Quá trình đăng bán sản phẩm được tối giản, không cần qua nhiều thủ tục phức tạp, qua đó tạo ra sự đa dạng cho thị trường và đáp ứng được nhiều nhu cầu của người tiêu dùng.
* Giá cả cạnh tranh: Vì loại bỏ các chi phí trung gian, giá bán trong mô hình C2C thường cạnh tranh hơn so với các mô hình kinh doanh truyền thống. Người mua có thể dễ dàng so sánh giá giữa các nhà bán khác nhau, từ đó tiết kiệm chi phí và mua được sản phẩm với giá ưu đãi nhất.
* Tính mở và dễ tiếp cận: Bất kỳ ai chỉ cần có kết nối Internet cũng có thể tham gia vào mô hình C2C. Việc đăng ký tài khoản và bắt đầu bán hàng được thực hiện nhanh chóng, đơn giản và thường miễn phí, tạo ra một sân chơi bình đẳng cho mọi cá nhân với cơ hội kinh doanh rộng lớn.
* **Lợi ích mà mô hình này mang lại:**
* Mở rộng phạm vi tiếp cận khách hàng: Mô hình C2C loại bỏ rào cản địa lý, cho phép người bán dễ dàng tiếp cận khách hàng trên toàn cầu thông qua Internet. Các nền tảng thương mại điện tử như Shopee, Lazada, Tiki, … thu hút lượng truy cập lớn hàng ngày, giúp sản phẩm được phơi bày trước một lượng khách hàng tiềm năng đáng kể, từ đó tăng cơ hội giao dịch thành công.
* Tiết kiệm chi phí vận hành: Với mô hình này, người bán không phải chịu các chi phí cố định như thuê mặt bằng, quản lý cửa hàng hay chi trả tiền lương cho nhân viên. Việc bán hàng trực tuyến chủ yếu đòi hỏi một khoản đầu tư ban đầu cho website, quảng cáo và chi phí vận chuyển, giúp giảm thiểu rủi ro tài chính và tối ưu hóa quy trình giao dịch cho cả hai bên.
* Cơ hội kinh doanh cho mọi đối tượng: C2C mở ra cánh cửa cho bất kỳ ai mong muốn tham gia vào hoạt động kinh doanh mà không cần nhiều vốn hay kinh nghiệm. Chỉ cần có sản phẩm và kết nối Internet, cá nhân có thể trở thành người bán trực tuyến. Điều này tạo điều kiện cho việc tạo thêm nguồn thu nhập cho từng người cũng như giúp các doanh nghiệp nhỏ tiếp cận thị trường rộng lớn, từ đó tăng doanh số và lợi nhuận.
* Trải nghiệm mua sắm thuận tiện và đa dạng: Người mua có thể dễ dàng mua sắm bất cứ lúc nào và từ bất cứ đâu, so sánh giá và lựa chọn sản phẩm ưng ý nhất. Đồng thời, người bán cũng có thể quản lý đơn hàng và giao dịch một cách hiệu quả thông qua các công cụ trực tuyến, đồng hành cùng đó là sự đa dạng về sản phẩm với các mức giá phù hợp với nhiều nhu cầu và sở thích khác nhau.
* Thúc đẩy sự cạnh tranh và đổi mới: Mô hình C2C không giới hạn người bán chỉ hoạt động trong một lĩnh vực cụ thể, giúp họ có thể đa dạng hóa danh mục sản phẩm hoặc dịch vụ. Các nền tảng liên tục cập nhật xu hướng thị trường, tạo điều kiện cho người bán nắm bắt nhanh chóng nhu cầu của khách hàng và cung cấp các sản phẩm phù hợp, từ đó gia tăng lợi thế cạnh tranh.

1. **Định nghĩa Website**

* Website là một hệ thống thông tin số hóa được xây dựng bằng các ngôn ngữ lập trình, kịch bản, và công nghệ web (HTML, CSS, JavaScript, v.v.), hoạt động như một giao diện tương tác giữa tổ chức/cá nhân với người dùng thông qua môi trường Internet. Cấu trúc của website bao gồm nhiều trang con (web pages) liên kết với nhau bằng siêu liên kết (hyperlink), được lưu trữ trên máy chủ web (web server) và phân phối đến người dùng thông qua giao thức HTTP/HTTPS. Mỗi website được định danh duy nhất bằng tên miền (domain name) và địa chỉ IP (Internet Protocol), cho phép truy cập từ mọi thiết bị có kết nối mạng thông qua trình duyệt (Chrome, Firefox, Safari).
* Cấu trúc cơ bản của một Website bao gồm:
* Giao diện người dùng (Frontend): Đây là lớp trình bày trực tiếp tiếp xúc với người dùng cuối, đóng vai trò như cầu nối giữa hệ thống và người sử dụng thông qua các thành phần quen thuộc như HTML để xây dựng khung sườn nội dung, CSS để tạo nên diện mạo trực quan và sinh động, còn JavaScript đảm nhiệm việc xử lý các thao tác tương tác một cách linh hoạt và mượt mà. Một điểm nổi bật của giao diện người dùng chính là việc áp dụng thiết kế đáp ứng (Responsive Design) — một hệ thống thông minh có khả năng tự điều chỉnh bố cục và hiển thị sao cho phù hợp với mọi kích thước màn hình, từ máy tính để bàn cho đến điện thoại di động, dựa trên các quy tắc do nhà thiết kế định sẵn. Bên cạnh đó, giao diện còn được tối ưu bằng các kỹ thuật như tương thích trình duyệt (Cross-Browser Compatibility) nhằm đảm bảo trải nghiệm nhất quán trên các nền tảng trình duyệt khác nhau, và tối ưu hiệu năng (Performance Optimization) với các giải pháp như tải chậm (Lazy Loading) và phân tách mã nguồn (Code Splitting) để rút ngắn thời gian tải trang, nâng cao trải nghiệm người dùng.
* Xử lý dữ liệu (Backend): Đây là nơi vận hành các chức năng cốt lõi của hệ thống, đảm nhận vai trò của lớp nghiệp vụ (Business Logic Layer) và lớp dữ liệu (Data Layer), hoạt động phía sau hậu trường để xử lý các yêu cầu từ người dùng. Thành phần chính trong lớp này là máy chủ (Server) – nơi tiếp nhận và xử lý các yêu cầu được gửi đến từ giao diện người dùng thông qua các điểm cuối API (API Endpoints), từ đó phản hồi lại dữ liệu phù hợp. Hệ thống còn tích hợp cơ sở dữ liệu (Database) – bộ nhớ trung tâm lưu trữ toàn bộ thông tin quan trọng như dữ liệu người dùng, sản phẩm, đơn hàng và giao dịch. Bên cạnh đó, yếu tố xác thực và bảo mật cũng được chú trọng, với các cơ chế như quản lý đăng nhập, phân quyền truy cập, mã hóa dữ liệu và bảo vệ thông tin cá nhân, đảm bảo hệ thống vận hành an toàn và đáng tin cậy.

1. **Công nghệ được sử dụng**
2. **Môi trường phát triển**

* Hệ điều hành được sử dụng trong quá trình phát triển dự án là Windows 10 phiên bản 64-bit. Đây là nền tảng được lựa chọn nhờ khả năng tương thích rộng rãi với các công cụ lập trình hiện đại, hỗ trợ đa nhiệm mạnh mẽ, cùng tính năng bảo mật nâng cao như Windows Defender và Windows Sandbox.
* Về kiểm thử giao diện và chức năng, trình duyệt Firefox được ưu tiên sử dụng nhờ bộ công cụ dành cho nhà phát triển (DevTools) toàn diện, bao gồm chế độ Responsive Design Mode để tối ưu hóa hiển thị trên nhiều thiết bị, tính năng debug JavaScript chi tiết, và khả năng phân tích hiệu suất trang web thông qua Network Monitor và Performance Tool. Firefox cũng được đánh giá cao về việc tuân thủ các tiêu chuẩn web mới nhất (HTML5, CSS3), đảm bảo tính tương thích chéo cho ứng dụng.
* Công đoạn biên dịch và xây dựng ứng dụng được tối ưu hóa thông qua sự kết hợp giữa Node.js runtime environment và Vite build tool Node.js đóng vai trò nền tảng quản lý gói thư viện thông qua npm (Node Package Manager), đồng thời hỗ trợ thực thi các script server-side. Trong khi đó, Vite – công cụ build front-end thế hệ mới – giúp tăng tốc độ phát triển nhờ cơ chế Hot Module Replacement (HMR) cho phép cập nhật mã nguồn ngay lập tức mà không cần reload trang, cùng khả năng tối ưu hóa tài nguyên tĩnh (CSS, JavaScript, hình ảnh) thông qua tích hợp Rollup và ES Modules. Sự kết hợp này không chỉ rút ngắn thời gian build mà còn hỗ trợ phát triển các ứng dụng Single-Page Application (SPA) một cách linh hoạt.

1. **Công cụ phát triển**

* **Figma** được sử dụng làm công cụ chính để thiết kế giao diện người dùng (UI/UX). Nền tảng này cho phép tạo wireframe, prototype tương tác, và thiết kế chi tiết với tính năng Auto Layout và Design Systems, giúp duy trì tính nhất quán về màu sắc, typography, và component trên mọi màn hình. Khả năng cộng tác thời gian thực (real-time collaboration) của Figma tối ưu hóa quy trình làm việc nhóm, cho phép các thành viên cùng chỉnh sửa, comment trực tiếp, và kiểm thử usability ngay trên trình duyệt.
* **Visual Studio Code** được lựa chọn làm môi trường soạn thảo mã nguồn chính nhờ hệ sinh thái extension phong phú (ESLint, Prettier, GitLens) và tích hợp sâu với các công cụ phát triển. Editor này hỗ trợ đa ngôn ngữ (JavaScript, TypeScript, Python), debugger trực quan, và terminal tích hợp, giúp tăng tốc độ code, kiểm tra lỗi, và triển khai ứng dụng liền mạch. Tính năng Live Share còn cho phép lập trình viên pair programming từ xa, nâng cao hiệu suất làm việc nhóm.
* **Git** kết hợp **GitHub** được triển khai để quản lý phiên bản mã nguồn (version control). Với quy trình Git Flow, hệ thống được tổ chức theo các nhánh chính (main, develop), nhánh feature, và nhánh hotfix, đảm bảo kiểm soát thay đổi mã nguồn chặt chẽ. GitHub đóng vai trò trung tâm để review code (pull request), theo dõi lỗi (issues), và tích hợp CI/CD (GitHub Actions) nhằm tự động hóa kiểm thử, build, và deploy. Kho lưu trữ đám mây còn hỗ trợ backup dữ liệu an toàn và chia sẻ mã nguồn với cộng đồng mã nguồn mở.
* **PowerDesigner** được ứng dụng để thiết kế mô hình cơ sở dữ liệu (Database Modeling). Công cụ này cho phép xây dựng sơ đồ ERD (Entity-Relationship Diagram), ánh xạ quan hệ giữa các thực thể, và tự động sinh script SQL tương thích với hệ quản trị cơ sở dữ liệu như MySQL, PostgreSQL. Tính năng reverse engineering của PowerDesigner giúp phân tích cấu trúc database từ mã nguồn có sẵn, đồng thời tối ưu hóa hiệu năng thông qua phân tích ràng buộc và chỉ mục.
* **draw.io (diagrams.net)** được sử dụng để thiết kế lưu đồ thuật toán (flowchart), sơ đồ use case, và kiến trúc hệ thống. Công cụ miễn phí này cung cấp thư viện hình khối đa dạng, tích hợp trực tiếp với Google Drive, GitHub, và Confluence để lưu trữ, chia sẻ tài liệu. Nhờ giao diện trực quan và khả năng xuất file đa định dạng (PNG, PDF, SVG), draw.io trở thành giải pháp tối ưu để mô tả logic nghiệp vụ và tương tác giữa các thành phần trong hệ thống.

1. **Giao diện người dùng**

* **ReactJS** (phiên bản 19.0.0) được lựa chọn làm thư viện JavaScript trung tâm để xây dựng giao diện người dùng (UI) dựa trên mô hình component-based architecture. Mỗi component trong React là một đơn vị độc lập, kết hợp logic (JavaScript), giao diện (JSX), và styling (CSS-in-JS), giúp tái sử dụng mã nguồn và quản lý trạng thái ứng dụng thông qua React Hooks (useState, useEffect, useContext). Virtual DOM của React tối ưu hiệu suất bằng cách chỉ cập nhật những phần thay đổi trên giao diện, giảm thiểu tương tác trực tiếp với DOM thật. Đặc biệt, phiên bản 19.0.0 tích hợp Concurrent Mode và Automatic Batching, cho phép xử lý tác vụ không đồng bộ mượt mà hơn và tối ưu hóa quá trình render.
* **TailwindCSS** (phiên bản 4.0.12) được triển khai như một utility-first CSS framework, thay thế phương pháp viết CSS truyền thống. Thay vì định nghĩa lớp CSS tùy chỉnh, Tailwind cung cấp hàng trăm utility class (ví dụ: flex, pt-4, text-center) để thiết kế trực tiếp trong HTML/JSX thông qua các attribute. Framework này đi kèm với tính năng Just-in-Time (JIT) Compiler, tự động tạo ra các lớp CSS động dựa trên mã nguồn, giúp giảm kích thước file build và hỗ trợ tùy biến theme (màu sắc, spacing, breakpoint) thông qua file tailwind.config.js. Sự kết hợp giữa Tailwind và PostCSS (qua plugin @tailwindcss/vite) đảm bảo quá trình build tối ưu và loại bỏ CSS không sử dụng (PurgeCSS).
* **Vite** (phiên bản 6.2.0) đóng vai trò là công cụ build front-end thế hệ mới, thay thế Webpack truyền thống. Vite tận dụng ES Modules (ESM) để phân chia ứng dụng thành các module nhỏ, cho phép trình duyệt tải song song các tài nguyên mà không cần bundle toàn bộ. Điều này giúp thời gian khởi động dev server giảm xuống dưới 1 giây, kể cả với dự án lớn. Tính năng Hot Module Replacement (HMR) của Vite cập nhật component React ngay lập tức khi có thay đổi mã nguồn mà không làm mất trạng thái ứng dụng. Ngoài ra, Vite tích hợp sẵn TypeScript, SCSS, và hỗ trợ tối ưu hóa production build thông qua Rollup (tree-shaking, code-splitting).
* Ngoài ra còn có các thư viện React khác như:
* react-modal (3.16.3): Quản lý modal/popup với khả năng kiểm soát accessibility (ARIA attributes) và tương thích SSR.
* react-router (7.3.0): Xử lý client-side routing với cơ chế dynamic routing, hỗ trợ nested routes và lazy loading components
* @tanstack/react-query (5.67.2): Thư viện quản lý data fetching và caching, tự động xử lý stale data, retry khi gặp lỗi, và đồng bộ hóa dữ liệu giữa các tab.
* react-multi-carousel (2.8.5): Xây dựng carousel đa năng với tính năng responsive breakpoints, infinite scroll, và custom arrows/dots.
* react-paginate (8.3.0): Phân trang dữ liệu server-side/client-side với giao diện có thể tùy chỉnh bằng CSS.
* chart.js + react-chartjs-2 (4.4.8 + 5.3.0): Render biểu đồ (line, bar, pie) tương tác, hỗ trợ real-time updates và tùy chỉnh tooltip/legend.
* dompurify (3.2.5): Làm sạch HTML đầu vào (XSS sanitization) trước khi render bằng dangerouslySetInnerHTML.
* browser-image-compression (2.0.2): Nén hình ảnh trước khi upload để giảm dung lượng (định dạng WebP/AVIF).

1. **Xử lý dữ liệu**

* **Supabase** (phiên bản SDK 2.49.1) được triển khai như một nền tảng Backend-as-a-Service mã nguồn mở, cung cấp giải pháp thay thế toàn diện cho Firebase với ưu điểm vượt trội về tính linh hoạt và khả năng tùy biến. Khác với Firebase (sử dụng NoSQL Firestore), Supabase xây dựng trên nền tảng PostgreSQL hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) mạnh mẽ, hỗ trợ SQL nguyên bản, transactions ACID, và stored procedures. Kiến trúc này cho phép phát triển ứng dụng phức tạp với quan hệ dữ liệu đa chiều, đồng thời tận dụng tính năng Row Level Security (RLS) để kiểm soát truy cập dữ liệu ở cấp độ từng hàng, đảm bảo an ninh thông qua chính sách role-based.
* Các dịch vụ cốt lõi của Supabase:
* **Authentication:** Hỗ trợ đa phương thức xác thực: Email/Mật khẩu, OAuth (Google, Facebook, GitHub), Magic Links, và WebAuthn (FIDO2). Quản lý phiên người dùng (JWT tokens) tự động, tích hợp với Supabase Storage và Realtime để đồng bộ quyền truy cập.
* **PostgreSQL Database**: Tương tác dữ liệu thông qua @supabase/supabase-js SDK:

1. // Truy vấn dữ liệu với RLS

2. const { data, error } = await supabase

3. .from('users')

4. .select('id, name')

5. .eq('role', 'admin');

* **Realtime Database**: Theo dõi thay đổi dữ liệu theo thời gian thực thông qua WebSocket:

1. const channel = supabase

2. .channel('orders')

3. .on('postgres\_changes', { event: 'INSERT', schema: 'public' }, handleNewOrder)

4. .subscribe();

* **Storage:** Lưu trữ file (hình ảnh, video) trong buckets, hỗ trợ CDN tích hợp để tối ưu tốc độ tải. Upload/download file với quyền truy cập được kiểm soát bởi RLS:

1. await supabase.storage.from('avatars').upload('user1.jpg', file);

2. const url = supabase.storage.from('avatars').getPublicUrl('user1.jpg');

* **Auto-generated APIs:** REST API và GraphQL API tự động sinh từ cấu trúc database, kèm theo tài liệu Swagger/OpenAPI. Tích hợp với React Query hoặc Axios để fetch dữ liệu từ client.

1. **Cơ sở dữ liệu**

* **PostgreSQL** được Supabase tích hợp như hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) nền tảng, đóng vai trò trung tâm trong việc lưu trữ và xử lý dữ liệu có cấu trúc của ứng dụng. Khác với các database NoSQL, PostgreSQL tuân thủ nghiêm ngặt các nguyên tắc ACID (Atomicity, Consistency, Isolation, Durability), đảm bảo tính toàn vẹn dữ liệu ngay cả trong các giao dịch phức tạp. Supabase mở rộng sức mạnh của PostgreSQL thông qua hỗ trợ real-time, Row Level Security (RLS), và khả năng tích hợp trực tiếp với dịch vụ xác thực người dùng, biến nó thành một giải pháp toàn diện cho cả persistence layer và business logic.
* PostgreSQL trên Supabase được thiết kế thành các schema (public, auth, storage) và bảng quan hệ, phản ánh nghiệp vụ ứng dụng:

1. **Quản lý thư viện và gói phần mềm**

* **npm (Node Package Manager)** đóng vai trò trung tâm trong việc quản lý vòng đời thư viện của dự án. Với phiên bản 9.x+ (đi kèm Node.js 18.x), npm không chỉ cài đặt các dependency thông qua lệnh npm install, mà còn hỗ trợ version locking (tệp package-lock.json) để đảm bảo mọi thành viên trong nhóm sử dụng chính xác phiên bản thư viện đã xác định, tránh xung đột do semantic versioning (SemVer). Quy trình làm việc tuân thủ nguyên tắc deterministic build, cho phép tái tạo chính xác môi trường phát triển trên mọi hệ thống.
* Tệp package.json hoạt động như bản thiết kế tổng thể của dự án, bao gồm:
* Dependencies: Danh sách thư viện runtime (React, Axios) được khai báo dưới trường dependencies.
* DevDependencies: Công cụ hỗ trợ phát triển (ESLint, Prettier, Vite) được khai báo trong devDependencies, chỉ cài đặt trong môi trường local.
* Scripts: Tập lệnh tự động hóa (start server, build production, kiểm tra lint).
* Engines: Ràng buộc phiên bản Node.js/npm tối thiểu ("node": ">=18.0.0").
* Proxy: Cấu hình reverse proxy để tránh lỗi CORS khi gọi API từ client.

**CHƯƠNG III. PHÂN TÍCH HỆ THỐNG**

1. **Quản lý dữ liệu**

* **Thông tin người dùng**: Trong quá trình đăng ký tài khoản, mỗi người dùng bắt buộc phải cung cấp đầy đủ các thông tin cá nhân bao gồm: địa chỉ Email, họ tên, số điện thoại, số căn cước công dân, ngày sinh, lựa chọn giới tính, cùng với một mật khẩu để đảm bảo tính bảo mật cho tài khoản. Ngay sau khi hoàn tất đăng ký, hệ thống xác thực Supabase Auth sẽ tự động khởi tạo và gán cho người dùng một ID ngẫu nhiên duy nhất, được sử dụng để định danh trong toàn bộ hệ thống. Bên cạnh các thông tin cơ bản, mỗi tài khoản người dùng còn được liên kết với một số thuộc tính phụ nhằm cá nhân hóa hồ sơ người dùng, bao gồm: liên kết ảnh đại diện (avatar), liên kết banner trang cá nhân, cũng như thông tin về thành phố (city) và tiểu bang (state) nơi người dùng đang sinh sống.
* **Thông tin xe máy**: Khi người dùng tiến hành đăng bán một chiếc xe máy, họ sẽ cần cung cấp các thông tin bắt buộc như kiểu xe, hãng xe, mẫu xe, và phiên bản của xe đó. Những thông tin này được lựa chọn từ tập dữ liệu có sẵn, nhờ đó hệ thống có thể tự động bổ sung các thuộc tính phụ như kích thước động cơ mà người dùng không cần phải nhập lại. Ngoài ra, người dùng còn phải nhập thêm các thông tin chi tiết khác bao gồm: số khung, số máy, ngày sản xuất, số dặm (mile) đã di chuyển, cùng với mô tả chi tiết, giá bán và tình trạng giấy tờ pháp lý. Hệ thống cũng có cơ chế phân loại xe dựa trên số dặm: nếu người dùng nhập 0 mile, hệ thống sẽ tự động xác định đây là xe mới, còn nếu nhập một giá trị lớn hơn 5 mile, xe sẽ được phân loại là đã qua sử dụng. Mỗi chiếc xe máy trong hệ thống đều được gắn kèm một số thuộc tính phụ để hỗ trợ việc quản lý và đánh giá hiệu quả hiển thị. Trong đó, lượt xem (views) phản ánh mức độ quan tâm của người dùng đối với chiếc xe đó, trong khi trạng thái đã bán sẽ được sử dụng để đánh dấu những chiếc xe đã giao dịch thành công, giúp hệ thống ẩn khỏi danh sách bán nhằm tránh gây nhầm lẫn cho người dùng khác.
* **Thông tin giao dịch**: Mỗi khi diễn ra một giao dịch mua bán xe máy trên hệ thống, một bản ghi giao dịch sẽ được tạo ra nhằm lưu trữ đầy đủ các thông tin cần thiết. Bản ghi này bao gồm: ID giao dịch, ID của người bán, ID của người mua, và ID của chiếc xe được giao dịch – tất cả đều được liên kết thông qua các khóa định danh duy nhất để đảm bảo tính chính xác và dễ dàng truy xuất. Bên cạnh đó, mỗi giao dịch còn đi kèm với một trạng thái giao dịch, dùng để phản ánh tiến độ hoặc kết quả của giao dịch đó, ví dụ như: đang chờ xử lý, đã hoàn tất, hoặc đã hủy. Hệ thống quản lý trạng thái này không chỉ giúp theo dõi quá trình giao dịch mà còn đảm bảo tính minh bạch và rõ ràng cho cả hai bên mua – bán.
* **Tin nhắn**: Hệ thống cho phép người dùng liên lạc trực tiếp với nhau thông qua tính năng gửi tin nhắn. Mỗi tin nhắn được gửi đi sẽ bao gồm các thông tin quan trọng như: ID tin nhắn, ID của người gửi, ID của người nhận, cùng với nội dung tin nhắn. Các tin nhắn này được lưu trữ trong cơ sở dữ liệu, cho phép người dùng dễ dàng xem lại lịch sử trò chuyện, đồng thời đảm bảo việc trao đổi thông tin giữa hai bên được thực hiện một cách nhanh chóng, tiện lợi và có tổ chức. Chức năng này góp phần hỗ trợ đắc lực cho quá trình thương lượng và mua bán xe giữa người dùng với nhau.
* **Hệ thống uy tín người dùng**: Để đảm bảo môi trường giao dịch an toàn và minh bạch, hệ thống tích hợp một cơ chế đánh giá độ uy tín cho mỗi người dùng. Thông qua hệ thống này, người dùng có thể gửi đánh giá cho nhau sau mỗi giao dịch, góp phần phản ánh mức độ tin cậy và chất lượng tương tác của các bên tham gia. Mỗi đánh giá được ghi nhận sẽ bao gồm: ID của đánh giá, ID người gửi đánh giá, ID người nhận đánh giá, nội dung chi tiết của đánh giá, và một thuộc tính xác định liệu đây có phải là đánh giá tích cực hay không. Những đánh giá này sẽ được hệ thống tổng hợp lại để tính toán và thể hiện mức độ uy tín của người dùng trên hồ sơ cá nhân, giúp các bên tham khảo trước khi tiến hành giao dịch trong tương lai.

A screenshot of a computer

AI-generated content may be incorrect.

Hình 3. 1 Mô hình MCD của hệ thống

1. **Thiết kế chức năng**
   1. **Đăng ký và đăng nhập**

* Để đảm bảo an toàn và bảo mật thông tin trong quá trình truy cập, hệ thống xác thực người dùng được xây dựng dựa trên nền tảng Supabase Auth, sử dụng JSON Web Token (JWT) để quản lý phiên đăng nhập. Hệ thống hỗ trợ hai phương thức xác thực phổ biến: đăng nhập truyền thống bằng email và mật khẩu, cũng như đăng nhập thông qua tài khoản Google bằng giao thức OAuth 2.0. Tất cả quá trình xác thực đều được xử lý tự động và an toàn, với các cơ chế bảo mật như mã hóa mật khẩu, xác thực token, và đồng bộ dữ liệu người dùng theo thời gian thực.

1. **Xử lý đăng ký**

* Trong quá trình đăng ký tài khoản, người dùng được yêu cầu nhập đầy đủ các thông tin cá nhân bắt buộc bao gồm: email, mật khẩu, họ tên, số điện thoại, số căn cước công dân (CMND/CCCD), ngày sinh, và giới tính. Hệ thống sẽ thực hiện mã hóa mật khẩu bằng thuật toán băm kèm muối (hash with salt) trước khi lưu trữ vào cơ sở dữ liệu, nhằm đảm bảo rằng ngay cả khi dữ liệu bị rò rỉ, thông tin nhạy cảm của người dùng vẫn được bảo vệ an toàn.
* Sau khi đăng ký thành công, Supabase sẽ tự động tạo một ID người dùng duy nhất (UUID) và đăng nhập người dùng vào hệ thống.

1. **Xử lý đăng nhập**

* Khi người dùng tiến hành đăng nhập bằng email và mật khẩu, hệ thống sẽ kiểm tra tính hợp lệ của thông tin bằng cách đối chiếu với dữ liệu đã lưu trong hệ thống Supabase. Nếu thông tin hợp lệ, Supabase Auth sẽ khởi tạo một JWT đại diện cho phiên đăng nhập và gửi về trình duyệt của người dùng. Token này sẽ được sử dụng trong các lần truy cập kế tiếp để xác thực danh tính người dùng mà không cần đăng nhập lại.
* Ngược lại, nếu thông tin không chính xác, hệ thống sẽ trả về thông báo lỗi cụ thể như: “Email không tồn tại” hoặc “Mật khẩu không đúng”, giúp người dùng dễ dàng nhận biết và xử lý sự cố.

1. **Đăng nhập hoặc đăng ký bằng Google (OAuth)**

* Hệ thống cũng hỗ trợ người dùng đăng nhập hoặc đăng ký bằng tài khoản Google thông qua giao thức OAuth 2.0. Khi người dùng lựa chọn hình thức này, hệ thống sẽ chuyển hướng đến trang đăng nhập chính thức của Google. Sau khi người dùng xác nhận quyền truy cập, Google sẽ trả về các thông tin cần thiết như email, tên người dùng, và ảnh đại diện.
* Hệ thống sẽ kiểm tra xem tài khoản Google đó đã tồn tại trong hệ thống hay chưa. Nếu chưa, một bản ghi người dùng mới sẽ được tạo tự động dựa trên các thông tin được cung cấp từ Google. Phiên đăng nhập cũng sẽ được quản lý bằng JWT như phương thức truyền thống.

1. **Xử lý callback sau xác thực**

* Sau khi đăng nhập thành công – dù là bằng email hay Google OAuth – hệ thống sẽ chuyển hướng người dùng đến một trang trung gian (callback) để xác minh phiên làm việc. Tại đây, JWT sẽ được kiểm tra lại một lần nữa để đảm bảo hợp lệ. Sau đó, hệ thống sẽ đồng bộ các thông tin người dùng với cơ sở dữ liệu (nếu có thay đổi), và cuối cùng chuyển hướng về trang chủ của ứng dụng. Trong trường hợp có lỗi phát sinh trong quá trình xử lý, người dùng sẽ được đưa trở lại trang đăng nhập cùng với thông báo lỗi cụ thể để khắc phục.
  1. **Chỉnh sửa thông tin người dùng**
* Để nâng cao trải nghiệm người dùng và đảm bảo dữ liệu cá nhân luôn được cập nhật chính xác, hệ thống cung cấp chức năng chỉnh sửa thông tin cá nhân. Tính năng này cho phép người dùng thay đổi các thông tin đã lưu trong hệ thống, bao gồm họ tên, email, số điện thoại, giới tính, ngày sinh, cũng như địa chỉ (bang và thành phố). Quá trình xử lý được thiết kế hướng đến sự mượt mà trong thao tác, đồng thời bảo đảm tính hợp lệ và đồng bộ dữ liệu với cơ sở dữ liệu một cách hiệu quả.

1. **Khởi tạo và tải dữ liệu**

* Ngay khi giao diện chỉnh sửa được hiển thị, hệ thống sẽ tiến hành tải danh sách các bang từ tập dữ liệu địa lý có cấu trúc dạng JSON. Nếu người dùng đã từng lưu thông tin địa chỉ trước đó, hệ thống sẽ tự động khôi phục lựa chọn bang và thành phố tương ứng, đồng thời tải danh sách thành phố thuộc bang đã chọn để hỗ trợ người dùng chỉnh sửa nhanh chóng và chính xác.

1. **Xử lý thay đổi địa chỉ**

* Khi người dùng chọn bang mới, hệ thống sẽ tìm kiếm danh sách thành phố tương ứng trong dữ liệu địa lý và cập nhật giao diện để hiển thị danh sách này. Trong trường hợp không có bang nào được chọn, danh sách thành phố sẽ tự động bị xóa, đảm bảo không có giá trị không hợp lệ được chọn.
* Khi người dùng chọn thành phố, giá trị này sẽ được lưu trữ tạm thời để sử dụng trong quá trình cập nhật thông tin khi người dùng xác nhận thay đổi.

1. **Cập nhật thông tin cá nhân**

Khi người dùng nhấn nút "Submit" để xác nhận chỉnh sửa, hệ thống sẽ thực hiện một loạt các bước xử lý:

* Kiểm tra tính hợp lệ dữ liệu: Hệ thống xác minh rằng các trường bắt buộc như họ tên, email, số điện thoại, CMND/CCCD, ngày sinh, và giới tính không được để trống.
* Chuẩn hóa dữ liệu đầu vào: Các trường như số điện thoại và CMND/CCCD sẽ được chuyển đổi từ dạng chuỗi sang kiểu số nguyên để đảm bảo đúng kiểu dữ liệu trong cơ sở dữ liệu.
* Cập nhật dữ liệu vào Supabase: Sau khi dữ liệu được kiểm tra và chuẩn hóa, hệ thống sẽ sử dụng Supabase để gửi yêu cầu cập nhật vào bảng USER, dựa trên uid của người dùng hiện tại.
* Nếu cập nhật thành công, hệ thống sẽ hiển thị thông báo “Profile updated successfully!” và tự động tải lại trang để áp dụng các thay đổi mới.
* Nếu cập nhật thất bại, hệ thống sẽ hiển thị thông báo lỗi cụ thể, đồng thời giữ nguyên trạng thái dữ liệu hiện tại để người dùng có thể dễ dàng chỉnh sửa và thử lại.

1. **Đảm bảo tính nhất quán**

* Dữ liệu địa lý động: Hệ thống chỉ cho phép hiển thị danh sách thành phố khi người dùng đã chọn bang, đảm bảo không có trường hợp chọn thành phố không hợp lệ.
* Giá trị giới tính: Trường is\_man sẽ được chuyển đổi từ chuỗi "true" hoặc "false" sang kiểu boolean chính xác trước khi lưu trữ, tránh lỗi định dạng trong cơ sở dữ liệu.
* Tự động đóng form: Sau khi cập nhật thành công, form chỉnh sửa sẽ tự động đóng và trang được tải lại để phản ánh dữ liệu mới nhất của người dùng.

1. **Xử lý lỗi tiềm ẩn**

* Lỗi từ Supabase: Nếu trong quá trình cập nhật, hệ thống không thể kết nối với Supabase hoặc có lỗi xảy ra, người dùng sẽ nhận được thông báo “Update failed!”, đồng thời lỗi được ghi lại để phục vụ việc phân tích và xử lý sau này.
* Lỗi tải dữ liệu địa lý: Trong trường hợp dữ liệu địa lý (bang/thành phố) không thể tải được, danh sách địa chỉ sẽ không hiển thị, ngăn chặn khả năng người dùng chọn địa chỉ không tồn tại hoặc không hợp lệ.
  1. **Đăng bán xe**
  2. **Chỉnh sửa thông tin xe đang bán**
  3. **Tìm kiếm và lọc xe**
  4. **Giao dịch**
  5. **Nhắn tin**
  6. **Đánh giá uy tín**

**CHƯƠNG V. GIAO DIỆN**

# KẾT LUẬN – ĐÁNH GIÁ

* **Kết quả đạt được**

Bài niên luận này đã thành công trong việc giải quyết bài toán tìm chu trình Euler trên một đồ thị vô hướng. Các chức năng chính được thiết kế và triển khai hoạt động ổn định, đảm bảo tính chính xác cao và ít khả năng phát sinh lỗi trong quá trình sử dụng. Giao diện hệ thống được xây dựng trực quan, dễ thao tác, và kết quả đầu ra được xử lý chính xác, đáp ứng tốt yêu cầu của người dùng.

* **Thu hoạch chuyên môn – Kinh nghiệm**

Thông qua quá trình thực hiện, đã giúp nắm vững các khái niệm cơ bản về đồ thị, đặc biệt là các điều kiện tồn tại chu trình Euler. Ngoài ra, việc triển khai thuật toán DFS và Fleury giúp hiểu sâu hơn về cách tối ưu hóa xử lý dữ liệu đồ thị. Kỹ năng lập trình và sử dụng các công cụ hỗ trợ như Visual Studio Code, GitHub, và thư viện JavaScript cũng được cải thiện đáng kể.

* **Ưu điểm, Hạn chế - Nguyên nhân**
* **Ưu điểm**

Chương trình được thiết kế với tính ổn định cao, đảm bảo hoạt động mượt mà trong hầu hết các trường hợp sử dụng. Giao diện trực quan, rõ ràng, giúp người dùng dễ dàng thao tác và hiểu cách hoạt động của hệ thống. Đặc biệt, kết quả đầu ra chính xác, đáp ứng yêu cầu về tính toán và phân tích, là một điểm mạnh quan trọng. Ngoài ra, chương trình tích hợp ba chức năng chính trong một bài toán gồm tạo đồ thị ngẫu nhiên, vẽ đồ thị thủ công, và nhập từ file, mang lại sự linh hoạt và đa dạng trong việc sử dụng.

* **Hạn chế**

Tuy nhiên, chương trình vẫn tồn tại một số hạn chế cần cải thiện. Khi xử lý các đồ thị có kích thước quá lớn, hiệu suất có thể giảm đáng kể, làm chậm quá trình thực thi. Đối với dữ liệu nhập thủ công, người dùng phải tuân thủ các quy định định dạng cụ thể, điều này đôi lúc gây bất tiện và làm tăng nguy cơ nhập sai. Một hạn chế kỹ thuật khác là thuật toán chưa được tối ưu hóa hoàn toàn; việc sử dụng nhiều vòng lặp có thể dẫn đến tình trạng quá tải hoặc treo chương trình trong các trường hợp đặc biệt.

* **Hướng phát triển**

Một trong những hướng phát triển là bổ sung thêm các cách biểu diễn đồ thị, chẳng hạn như danh sách cạnh hoặc danh sách kề, giúp linh hoạt hơn trong việc nhập liệu và xử lý dữ liệu. Hơn nữa, việc phát triển chức năng hỗ trợ vẽ đồ thị có hướng sẽ mở rộng phạm vi ứng dụng của chương trình. Cuối cùng, tối ưu hóa các thuật toán hiện tại bằng cách giảm bớt vòng lặp không cần thiết hoặc áp dụng các thuật toán nâng cao sẽ cải thiện đáng kể hiệu suất, đặc biệt khi xử lý các đồ thị lớn và phức tạp.

# TÀI LIỆU THAM KHẢO

**Tài liệu tiếng việt:**

1. Nguyễn Hữu Danh. Toán Rời Rạc 2. Chương 1, 2, 3 và 4. Khoa Kỹ Thuật – Công Nghệ, Đại Học Tây Đô, 2023.
2. Nguyễn Chí Cường. Cấu trúc dữ liệu & Giải thuật. Khoa Kỹ thuật – Công nghệ, Đại Học Tây Đô, 2023.
3. Nguyễn Chí Cường. Lập trình căn bản. Khoa Kỹ thuật – Công nghệ, Đại Học Tây Đô, 2023.
4. Lâm Tấn Phương. Giáo trình Lý thuyết Thiết kế và lập trình Web, Khoa Kỹ thuật – Công nghệ, Đại Học Tây Đô, 2024.

**Tài liệu từ Internet:**

1. GeeksForGeeks. Fleury’s Algorithm for printing Eulerian Path or Circuit. https://www.geeksforgeeks.org/fleurys-algorithm-for-printing-eulerian-path/ (Tháng 12/2024)
2. Mozilla. JavaScript reference. https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference (Tháng 12/2024)
3. Bootstrap. Bootstrap Documents. getbootstrap.com/docs/5.3 (Tháng 12/2024)
4. OCLPhase2. Fleury Algorithm. youtube.com/watch?v=HXNyr2N3euc (Tháng 12/2024)
5. Thái Thanh Hải. Data Structure & Algorithm - Graph Algorithms - Depth First Search (DFS). viblo.asia/p/data-structure-algorithm-graph-algorithms-depth-first-search-dfs-qPoL7zyXJvk (Tháng 12/2024)
6. Thái Thanh Hải. Data Structure & Algorithm - Graph Algorithms - Breadth First Search (BFS). viblo.asia/p/data-structure-algorithm-graph-algorithms-breadth-first-search-bfs-gwd43kMM4X9 (Tháng 12/2024)
7. Viblo. Biểu diễn đồ thị trên máy tính. viblo.asia/p/bieu-dien-do-thi-tren-may-tinh-XL6lABrD5ek (Tháng 12/2024)
8. Viblo. Đồ thị Euler và Chu trình Euler. viblo.asia/p/do-thi-euler-va-chu-trinh-euler-WAyK8Bj6lxX (Tháng 12/2024)
9. Wikipedia. Đường đi Euler. vi.wikipedia.org/wiki/Đường\_đi\_Euler (Tháng 12/2024)
10. Wikipedia. Ma trận kề. vi.wikipedia.org/wiki/Ma \_trận\_kề (Tháng 12/2024)