|  |  |
| --- | --- |
| A blue and red text on a white background  Description automatically generated | BỘ GIÁO DỤC VÀ ĐÀO TẠO  **TRƯỜNG ĐẠI HỌC CÔNG NGHỆ TP. HỒ CHÍ MINH** |

**ĐỒ ÁN MÔN HỌC: NGÔN NGỮ PHÁT TRIỂN ỨNG DỤNG MỚI**

**< Website quản lý kinh doanh billiards >**

Ngành: **CÔNG NGHỆ THÔNG TIN**

Lớp: **21DTHC1**

|  |  |  |  |
| --- | --- | --- | --- |
| Giáo viên hướng dẫn | : | GV. Nguyễn Chí Toàn |  |
| Sinh viên | : | Cao Hiếu Đạt | MSSV:2180606537 |
| Sinh viên | : | Vũ Hoàng Tấn | MSSV:2180604416 |
| Sinh viên | : | Phạm Minh Thịnh | MSSV:2180606330 |
| Sinh viên | : | Đinh Nho Bảo Nam | MSSV:2180606788 |
| Sinh viên | : | Hồ Nhật Quang | MSSV:2180602822 |
| Sinh viên | : | Đoàn Lục Quốc Duy | MSSV:2180603782 |

TP. Hồ Chí Minh, <2025>

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

TP.HCM, ngày 4 tháng 7 năm 2025

Giảng viên hướng dẫn

**LỜI NÓI ĐẦU**

Chúng em xin bày tỏ lòng biết ơn chân thành đến giảng viên **Nguyễn Chí Toàn**, là Giảng viên dạy chúng em trong quá trình học tập. Trong suốt thời gian từ khi bắt đầu học môn học “**NGÔN NGỮ PHÁT TRIỂN ỨNG DỤNG MỚI**”, chúng em đã nhận được rất nhiều sự quan tâm, giúp đỡ và hỗ trợ của thầy cùng bạn bè.

Trong thời đại 4.0 phát triển như hiện nay, công nghệ thông tin (CNTT) đã và đang đóng vai trò vô cùng quan trọng trong đời sống, kinh tế và xã hội của nhiều quốc gia, là một phần không thể thiếu của thời kỳ công nghiệp hóa - hiện đại hóa (CNH-HĐH). Việc áp dụng những sản phẩm tin học vào công việc ngày càng gia tăng, mang lại hiệu quả cao hơn và tiết kiệm thời gian.

Xuất phát từ nhu cầu thực tế đó, cùng với công việc quản lý kinh doanh BILLIARDS, việc quản lý hoạt động kinh doanh, tài sản, trang thiết bị, và dịch vụ là một việc không thể thiếu trong quá trình vận hành các cơ sở này. Nhằm thay thế một số công việc mà trước đó đạt hiệu quả không cao và mất nhiều thời gian, chúng em đã chọn đề tài “Website quản lý kinh doanh billiards” để đáp ứng nhu cầu trên và phục vụ cho các cơ sở kinh doanh bida và internet.

Do thời gian thực hiện đề tài có hạn, cùng với trình độ chuyên môn, kinh nghiệm và kiến thức của chúng em còn hạn chế nên sai sót là điều khó tránh khỏi. Vì vậy, chúng em rất mong nhận được những ý kiến đóng góp quý báu của thầy giúp chúng em có thêm kiến thức trong lĩnh vực này để đề tài nghiên cứu được hoàn thiện hơn và có thể áp dụng vào thực tế.

Chúng em xin chân thành cảm ơn!

Mục lục

[**LỜI NÓI ĐẦU** 3](#_Toc181220269)

[Mục lục 4](#_Toc181220270)

[**Mục lục hình ảnh** 5](#_Toc181220271)

[**CHƯƠNG I. TỔNG QUAN** 7](#_Toc181220272)

[**1.1** **Nhiệm vụ đồ án** 7](#_Toc181220273)

[*1.1.1* *Quản lý khách hàng* 7](#_Toc181220274)

[*1.1.2* *Quản lý máy tính và bàn bida* 7](#_Toc181220275)

[*1.1.3* *Quản lý tài chính* 7](#_Toc181220276)

[*1.1.4* *Các nhiệm vụ khác* 7](#_Toc181220277)

[*1.1.5* *Thanh toán qua nhiều hình thức* 7](#_Toc181220278)

[**1.2** **Mục tiêu đề tài** 7](#_Toc181220279)

[**1.3** **Khảo sát thực trạng** 8](#_Toc181220280)

[**CHƯƠNG II. CƠ SỞ LÝ THUYẾT** 10](#_Toc181220281)

[**2.1 Ngôn ngữ lập trình webSite** 10](#_Toc181220282)

[*2.1.1 HTML* 10](#_Toc181220283)

[*2.1.2 CSS* 12](#_Toc181220284)

[*2.1.3 Javascript* 13](#_Toc181220285)

[**2.2 Thư viện cho CSS và Javascript** 15](#_Toc181220286)

[*2.1.4 Bootstrap* 15](#_Toc181220287)

[**2.3 MongoDB** 17](#_Toc181220288)

[**2.4 Ajax** 18](#_Toc181220289)

[**2.5 Express.js** 19](#_Toc181220290)

[*2.5.1 Khái niệm* 19](#_Toc181220291)

[*2.5.2 Tính năng chính của Express.js* 19](#_Toc181220292)

[*2.5.3 Ứng dụng của Express.js* 20](#_Toc181220293)

[*2****.****5.4 Lợi ích của Express.js* 20](#_Toc181220294)

[**2.6 Node.js** 20](#_Toc181220295)

[*2.6.1 Khái niệm* 20](#_Toc181220296)

[*2.6.2 Đặc điểm chính của Node.js* 20](#_Toc181220297)

[*2.6.3 Ứng dụng của Node.js* 21](#_Toc181220298)

[*2.6.4 Lợi ích của Node.js* 21](#_Toc181220299)

[**CHƯƠNG III: ĐẶC TẢ YÊU CẦU** 23](#_Toc181220300)

[**3.1 Mô hình Use case** 23](#_Toc181220301)

[*Use case tổng quát* 23](#_Toc181220302)

[**3.2 ActivityDiagram** 24](#_Toc181220303)

[**3.3 Chức năng & giao diện** 28](#_Toc181220304)

[*3.3.1 Đăng nhập* 28](#_Toc181220305)

[*3.3.2 Thống kê* 28](#_Toc181220306)

[*3.3.3 Giao diện và chức năng bàn billiard* 30](#_Toc181220307)

[*3.3.4 Chức năng & giao diện hóa đơn* 30](#_Toc181220308)

[*3.3.5 Chức năng & giao diện đặt bàn.* 31](#_Toc181220309)

[*3.3.5 Chức năng & giao diện in hóa đơn.* 33](#_Toc181220310)

[*3.3.6 Profile* 33](#_Toc181220311)

[**3.4 Mô hình Diagam** 34](#_Toc181220312)

[**CHƯƠNG IV: KẾT LUẬN VÀ KIẾN NGHỊ** 35](#_Toc181220313)

[**4.1 Kết quả đạt được** 35](#_Toc181220314)

[**4.2 Hướng phát triển** 35](#_Toc181220315)

[**TÀI LIỆU THAM KHẢO** 37](#_Toc181220316)

# **Mục lục hình ảnh**

[Hình 2. 1 LOGO HTML 10](#_Toc181200869)

[Hình 2. 2 CSS 12](#_Toc181200870)

[Hình 2. 3 javascript 13](#_Toc181200871)

[Hình 2. 4 MongoDB 17](#_Toc181200872)

[Hình 3. 1 use case tổng quát 23](#_Toc181200852)

[Hình 3. 2 UseCase đăng nhập 23](#_Toc181200853)

[Hình 3. 3 Active diagram đặt bàn 24](#_Toc181200854)

[Hình 3. 4 ActivityDiagram chỉnh sửa thông tin bàn 25](#_Toc181200855)

[Hình 3. 5 Dịch vụ 26](#_Toc181200856)

[Hình 3. 6 Thông tin khách hàng 27](#_Toc181200857)

[Hình 3. 7 đăng nhập 28](#_Toc181200858)

[Hình 3. 8 Giao diện thống kê 29](#_Toc181200859)

[Hình 3. 9 Giao diện tổng quan 29](#_Toc181200860)

[Hình 3. 10 Hóa Đơn 30](#_Toc181200861)

[Hình 3. 11 Hóa đơn dịch vụ 31](#_Toc181200862)

[Hình 3. 12 Đặt bàn 31](#_Toc181200863)

[Hình 3. 13 Lưu thông tin khách đặt 32](#_Toc181200864)

[Hình 3. 14 Thông tin đặt bàn 32](#_Toc181200865)

[Hình 3. 15 In hóa đơn 33](#_Toc181200866)

[Hình 3. 16 Profile 33](#_Toc181200867)

[Hình 3. 17 Mô hình diagam 34](#_Toc181200868)

**CHƯƠNG I. TỔNG QUAN**

* 1. **Nhiệm vụ đồ án**
     1. *Quản lý khách hàng*
* Ghi nhận thông tin khách hàng
* Lưu trữ lịch sử sử dụng của khách hàng.
  + 1. *Quản lý máy tính và bàn bida*
* Theo dõi tình trạng hoạt động của máy tính và bàn bida (đang sử dụng, trống, ...).
* Ghi nhận thời gian sử dụng máy tính và bàn bida của khách hàng.
* In hóa đơn thanh toán.
  + 1. *Quản lý tài chính*
* Theo dõi doanh thu và chi phí của quán.
* Lập báo cáo tài chính theo ngày, tuần, tháng, ...
* Quản lý thu chi tiền mặt và thanh toán qua thẻ.
  + 1. *Các nhiệm vụ khác*
* Quản lý kho hàng (đồ uống, thức ăn, ...).
* Cung cấp các báo cáo thống kê về hoạt động kinh doanh của quán.
  + 1. *Thanh toán qua nhiều hình thức*
  1. **Mục tiêu đề tài**

Xây dựng kinh doanh dịch vụ giải trí billiards với mục tiêu thiết lập mô hình hiệu quả, thu hút khách hàng, tạo dựng thương hiệu uy tín và mang lại lợi nhuận bền vững.

* Tìm hiểu phương pháp phân tích và thiết kế hệ thống.
* Xây dựng web bán hoa.
* Đảm bảo cơ sở dữ liệu đáng tin cậy.
* Khảo sát và thu thập các thông tin từ các tài liệu và cơ sở dữ liệu có liên quan.
* Xây dựng website quản lý với giao diện thân thiện, đơn giản, dễ sử dụng.
* Đảm bảo cơ sở dữ liệu có độ tin cậy.
* Giảm bớt công việc cho nhân viên.
* Giúp cho việc báo cáo, thống kê dễ dàng và thuận tiện nhanh chóng hơn.
  1. **Khảo sát thực trạng**

Hiện tại mô hình kinh doanh billiards là mô hình kinh doanh mới ở Việt Nam nên có ít các websize hỗ trợ mô hính kinh doanh này vì thế đây được coi như là một cơ hội cũng như thách thức lớn.

Bộ phận quản lý:

* Theo dõi và kiểm tra hoạt động kinh doanh:
* Theo từng chu kỳ thời gian nhất định (theo ngày, theo tuần, theo tháng, …), bộ phận quản lý có thể theo dõi và kiểm tra tình trạng kinh doanh, doanh thu để báo cáo lãnh đạo và phân tích, triển khai chiến lược kinh doanh mới.
* Quản lý có thể theo dõi công việc của cửa hàng thông qua các báo cáo hàng tháng, quý, năm của nhân viên.
* Quản lý tài sản và trang thiết bị:
* Quản lý các thiết bị bida, máy tính và các thiết bị khác trong cửa hàng, theo dõi tình trạng và cập nhật thường xuyên trên hệ thống, thêm, sửa, xóa thông tin, cập nhật giá cả và tình trạng sử dụng.
* Quản lý sản phẩm và dịch vụ:
* Quản lý các dịch vụ cung cấp trong cửa hàng như thời gian thuê bàn bida, giờ sử dụng máy tính, theo dõi và cập nhật thông tin dịch vụ, thêm, sửa, xóa dịch vụ, cập nhật đơn giá của các dịch vụ.
* Nhân viên thu ngân:
* Chịu trách nhiệm thanh toán, tư vấn cho khách hàng.
* Kiểm kê doanh thu trong ngày, kiểm tra tiền.
* Lập hóa đơn cho khách hàng và thanh toán.
* Nhân viên phục vụ:
* Kiểm tra và xác nhận đơn đặt dịch vụ, kiểm tra tin nhắn của khách hàng và trả lời trên hệ thống.
* Lập phiếu đặt dịch vụ và xác nhận thanh toán từ khách hàng.
* Kiểm tra số lượng và tình trạng thiết bị trên hệ thống.
* Nhân viên kỹ thuật:
* Chịu trách nhiệm bảo dưỡng và sửa chữa thiết bị bida, máy tính theo yêu cầu và đơn đặt hàng.
* Đảm bảo thiết bị hoạt động đúng tiêu chuẩn và hỗ trợ kỹ thuật khi có sự cố.
* Nhân viên hỗ trợ khách hàng:
* Hỗ trợ khách hàng trong quá trình sử dụng dịch vụ, giải đáp thắc mắc và đảm bảo khách hàng hài lòng với dịch vụ.

Việc khảo sát thực trạng này sẽ giúp xác định những vấn đề tồn tại và cơ hội cải tiến trong mô hình kinh doanh của các cửa hàng bida và quán internet, từ đó đưa ra các giải pháp quản lý hiệu quả hơn, nâng cao chất lượng dịch vụ và tăng trưởng kinh doanh.

# **CHƯƠNG II. CƠ SỞ LÝ THUYẾT**

## **2.1 Ngôn ngữ lập trình webSite**

### *2.1.1 HTML*

HTML (viết tắt của từ HyperText Markup Language, hay còn gọi là "Ngôn ngữ Đánh dấu Siêu văn bản") là một [ngôn ngữ đánh dấu](https://vi.wikipedia.org/wiki/Ng%C3%B4n_ng%E1%BB%AF_%C4%91%C3%A1nh_d%E1%BA%A5u) được thiết kế ra để tạo nên các [trang web](https://vi.wikipedia.org/wiki/Website) trên [World Wide Web](https://vi.wikipedia.org/wiki/World_Wide_Web). Nó có thể được trợ giúp bởi các công nghệ như [CSS](https://vi.wikipedia.org/wiki/CSS) và các ngôn ngữ kịch bản giống như [JavaScript](https://vi.wikipedia.org/wiki/JavaScript).

Các [trình duyệt web](https://vi.wikipedia.org/wiki/Tr%C3%ACnh_duy%E1%BB%87t_web) nhận tài liệu HTML từ một [web server](https://vi.wikipedia.org/wiki/M%C3%A1y_ch%E1%BB%A7_web) hoặc một kho lưu trữ cục bộ và [render](https://vi.wikipedia.org/w/index.php?title=Browser_engine&action=edit&redlink=1) tài liệu đó thành các trang web [đa phương tiện](https://vi.wikipedia.org/wiki/%C4%90a_ph%C6%B0%C6%A1ng_ti%E1%BB%87n). HTML mô tả cấu trúc của một [trang web](https://vi.wikipedia.org/wiki/Trang_web) về mặt ngữ nghĩa và các dấu hiệu ban đầu được bao gồm cho sự xuất hiện của tài liệu.

Các [phần tử HTML](https://vi.wikipedia.org/wiki/Ph%E1%BA%A7n_t%E1%BB%AD_HTML) là các khối xây dựng của các trang HTML. Với cấu trúc HTML, hình ảnh và các đối tượng khác như biểu mẫu tương tác có thể được nhúng vào trang được hiển thị. HTML cung cấp một phương tiện để tạo tài liệu có cấu trúc bằng cách biểu thị ngữ nghĩa cấu trúc cho văn bản như headings, paragraphs, lists, [links](https://vi.wikipedia.org/wiki/Si%C3%AAu_li%C3%AAn_k%E1%BA%BFt), quotes và các mục khác. Các phần tử HTML được phân định bằng các thẻ, được viết bằng dấu ngoặc nhọn. Các thẻ như <img /> và <input /> giới thiệu trực tiếp nội dung vào trang. Các thẻ khác như <p> bao quanh và cung cấp thông tin về văn bản tài liệu và có thể bao gồm các thẻ khác làm phần tử phụ. Các trình duyệt không hiển thị các thẻ HTML, nhưng sử dụng chúng để diễn tả nội dung của trang.

HTML có thể nhúng các chương trình được viết bằng [scripting](https://vi.wikipedia.org/wiki/Ng%C3%B4n_ng%E1%BB%AF_k%E1%BB%8Bch_b%E1%BA%A3n) như [JavaScript](https://vi.wikipedia.org/wiki/JavaScript), điều này ảnh hưởng đến hành vi và nội dung của các trang web. Việc bao gồm CSS xác định giao diện và bố cục của nội dung. [World Wide Web Consortium](https://vi.wikipedia.org/wiki/World_Wide_Web_Consortium) (W3C), trước đây là đơn vị bảo trì HTML và là người duy trì hiện tại của các tiêu chuẩn CSS, đã khuyến khích việc sử dụng CSS trên HTML trình bày rõ ràng kể từ năm 1997.[[2]](https://vi.wikipedia.org/wiki/HTML#cite_note-deprecated-2)

A logo on a white background

Description automatically generated

Hình 2. 1 LOGO HTML

Lịch sử phát triển

Năm 1980, nhà khoa học máy tính [Tim Berners-Lee](https://vi.wikipedia.org/wiki/Tim_Berners-Lee), một nhà thầu tại [CERN](https://vi.wikipedia.org/wiki/CERN), đã đề xuất và tạo mẫu [ENQUIRE](https://vi.wikipedia.org/wiki/ENQUIRE), một hệ thống cho các nhà nghiên cứu CERN sử dụng và chia sẻ tài liệu. Năm 1989, Berners-Lee đã viết một bản ghi nhớ đề xuất một hệ thống [siêu văn bản](https://vi.wikipedia.org/wiki/Si%C3%AAu_v%C4%83n_b%E1%BA%A3n) dựa trên [Internet](https://vi.wikipedia.org/wiki/Internet).[1] Berners-Lee xác định rõ HTML và viết phần mềm trình duyệt và máy chủ vào cuối năm 1990. Năm đó, Berners-Lee và kỹ sư hệ thống dữ liệu CERN Robert Cailliau đã hợp tác để cùng yêu cầu tài trợ, nhưng dự án không được CERN chính thức thông qua. Trong ghi chú cá nhân của mình từ năm 1990, ông đã liệt kê "một số trong nhiều lĩnh vực mà siêu văn bản được sử dụng" và đặt một cuốn bách khoa toàn thư lên hàng đầu

Mô tả HTML công khai đầu tiên là một tài liệu có tên ["HTML Tags"](http://info.cern.ch/hypertext/WWW/MarkUp/Tags.html), lần đầu tiên được đề cập trên Internet bởi Tim Berners-Lee vào cuối năm 1991. Nó mô tả 18 phần tử bao gồm thiết kế ban đầu, tương đối đơn giản của HTML. Ngoại trừ thẻ siêu liên kết, chúng bị ảnh hưởng mạnh bởi [SGMLguid](https://vi.wikipedia.org/wiki/SGMLguid), một định dạng tài liệu dựa trên [Standard Generalized Markup Language](https://vi.wikipedia.org/wiki/Standard_Generalized_Markup_Language) (SGML) tại CERN. Mười một trong số các phần tử này vẫn tồn tại trong HTML 4.

HTML là một [ngôn ngữ đánh dấu](https://vi.wikipedia.org/wiki/Ng%C3%B4n_ng%E1%BB%AF_%C4%91%C3%A1nh_d%E1%BA%A5u) mà các trình duyệt web sử dụng để giải thích và soạn văn bản, hình ảnh và các tài liệu khác thành các trang web trực quan hoặc nghe được. Các đặc điểm mặc định cho mọi mục của đánh dấu HTML được xác định trong trình duyệt và các đặc điểm này có thể được thay đổi hoặc nâng cao bằng cách sử dụng thêm [CSS](https://vi.wikipedia.org/wiki/CSS) của nhà thiết kế trang web. Nhiều thành phần văn bản được tìm thấy trong báo cáo kỹ thuật 1988 ISO TR 9537 Techniques for using SGML, lần lượt đề cập đến các tính năng của các ngôn ngữ định dạng văn bản ban đầu, chẳng hạn như được sử dụng bởi lệnh [RUNOFF](https://vi.wikipedia.org/wiki/TYPSET_and_RUNOFF) được phát triển vào đầu những năm 1960 cho [hệ điều hành](https://vi.wikipedia.org/wiki/H%E1%BB%87_%C4%91i%E1%BB%81u_h%C3%A0nh) [CTSS](https://vi.wikipedia.org/wiki/Compatible_Time-Sharing_System) (Compatible Time-Sharing System): các lệnh định dạng này bắt nguồn từ các lệnh được sử dụng bởi các bộ sắp chữ để định dạng tài liệu theo cách thủ công. Tuy nhiên, khái niệm SGML về đánh dấu tổng quát dựa trên các phần tử (các phạm vi được chú thích lồng nhau với các thuộc tính) chứ không chỉ đơn thuần là các hiệu ứng in, với sự phân tách của cấu trúc và đánh dấu, HTML đã được chuyển dần theo hướng này với CSS.

Nguồn: <https://vi.wikipedia.org/w/index.php?title=HTML&veaction=edit&section=2>

*2.1.2 CSS*

A screen shot of a logo

Description automatically generated

Hình 2. 2 CSS

Trong môn [tin học](https://vi.wikipedia.org/wiki/Tin_h%E1%BB%8Dc), các tập tin định kiểu theo tầng – dịch từ [tiếng Anh](https://vi.wikipedia.org/wiki/Ti%E1%BA%BFng_Anh) là Cascading Style Sheets (CSS) – được dùng để miêu tả cách trình bày các tài liệu viết bằng ngôn ngữ [HTML](https://vi.wikipedia.org/wiki/HTML) và [XHTML](https://vi.wikipedia.org/wiki/XHTML).[1] Ngoài ra ngôn ngữ định kiểu theo tầng cũng có thể dùng cho [XML](https://vi.wikipedia.org/wiki/XML), [SVG](https://vi.wikipedia.org/wiki/SVG), [XUL](https://vi.wikipedia.org/wiki/XUL). Các đặc điểm kỹ thuật của CSS được duy trì bởi [World Wide Web Consortium](https://vi.wikipedia.org/wiki/W3C) ([W3C](http://www.w3c.org/)). Thay vì đặt các thẻ quy định kiểu dáng cho văn bản [HTML](https://vi.wikipedia.org/wiki/HTML) (hoặc [XHTML](https://vi.wikipedia.org/wiki/XHTML)) ngay trong [nội dung](https://vi.wikipedia.org/wiki/N%E1%BB%99i_dung) của nó, bạn nên sử dụng CSS.

Tác dụng CSS

* Hạn chế tối thiểu việc làm rối mã [HTML](https://vi.wikipedia.org/wiki/HTML) của trang [Web](https://vi.wikipedia.org/wiki/World_Wide_Web) bằng các thẻ quy định kiểu dáng (chữ đậm, chữ in nghiêng, chữ có gạch chân, chữ màu), khiến mã nguồn của trang Web được gọn gàng hơn, tách nội dung của trang [Web](https://vi.wikipedia.org/wiki/World_Wide_Web) và định dạng hiển thị, dễ dàng cho việc cập nhật nội dung.
* Tạo ra các kiểu dáng có thể áp dụng cho nhiều trang Web, giúp tránh phải lặp lại việc định dạng cho các trang [Web](https://vi.wikipedia.org/wiki/World_Wide_Web) giống nhau.

Nguồn: <https://vi.wikipedia.org/w/index.php?title=CSS&veaction=edit&section=1>

### *2.1.3 Javascript*

A yellow square with black letters

Description automatically generated

Hình 2. 3 javascript

JavaScript, theo phiên bản hiện hành, là một [ngôn ngữ lập trình](https://vi.wikipedia.org/wiki/Ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh) được phát triển từ các ý niệm [nguyên mẫu](https://vi.wikipedia.org/wiki/L%E1%BA%ADp_tr%C3%ACnh_d%E1%BB%B1a_tr%C3%AAn_nguy%C3%AAn_m%E1%BA%ABu). Ngôn ngữ này được dùng rộng rãi cho các [trang web](https://vi.wikipedia.org/wiki/Website) (phía người dùng) cũng như phía máy chủ (với Nodejs). Nó vốn được phát triển bởi [Brendan Eich](https://vi.wikipedia.org/wiki/Brendan_Eich) tại [Hãng truyền thông Netscape](https://vi.wikipedia.org/wiki/Netscape) với cái tên đầu tiên Mocha, rồi sau đó đổi tên thành LiveScript, và cuối cùng thành JavaScript. Giống [Java](https://vi.wikipedia.org/wiki/Java_(ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh)), JavaScript có [cú pháp](https://vi.wikipedia.org/wiki/C%C3%BA_ph%C3%A1p_h%E1%BB%8Dc) tương tự [C](https://vi.wikipedia.org/wiki/C_(ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh)), nhưng nó gần với Self hơn Java. .js là [phần mở rộng](https://vi.wikipedia.org/wiki/Ph%E1%BA%A7n_m%E1%BB%9F_r%E1%BB%99ng_t%C3%AAn_t%E1%BA%ADp_tin) thường được dùng cho [tập tin](https://vi.wikipedia.org/wiki/T%E1%BA%ADp_tin) [mã nguồn](https://vi.wikipedia.org/wiki/M%C3%A3_ngu%E1%BB%93n) JavaScript.

Phiên bản mới nhất của JavaScript là ECMAScript 12[1]. [ECMAScript](https://vi.wikipedia.org/wiki/ECMAScript) là phiên bản chuẩn hóa của JavaScript. [Trình duyệt](https://vi.wikipedia.org/wiki/Tr%C3%ACnh_duy%E1%BB%87t_web) [Mozilla](https://vi.wikipedia.org/wiki/Mozilla) phiên bản 1.8 [beta](https://vi.wikipedia.org/wiki/Beta) 1 có hỗ trợ không đầy đủ cho [E4X](https://vi.wikipedia.org/wiki/E4X) - phần mở rộng cho JavaScript hỗ trợ làm việc với [XML](https://vi.wikipedia.org/wiki/XML), được chuẩn hóa trong ECMA-357.

**Ứng dụng**

Cùng thời điểm Netscape bắt đầu sử dụng Java trên [trình duyệt Netscape](https://vi.wikipedia.org/wiki/Tr%C3%ACnh_duy%E1%BB%87t_Netscape), LiveScript đã được đổi tên thành JavaScript để được chú ý hơn bởi ngôn ngữ lập trình [Java](https://vi.wikipedia.org/wiki/Java_(ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh)) lúc đó đang được coi là một hiện tượng. JavaScript được bổ sung vào [trình duyệt](https://vi.wikipedia.org/wiki/Tr%C3%ACnh_duy%E1%BB%87t_web) Netscape bắt đầu từ phiên bản 2.0b3 của trình duyệt này vào tháng 12 năm [1995](https://vi.wikipedia.org/wiki/1995). Trên thực tế, JavaScript không được phát triển dựa từ Java. Do đó JavaScript chỉ dựa trên các cách đặt tên của Java. Java Script gồm 2 mảng là client-server thực hiện lệnh trên máy của end-user và web-server.

Sau thành công của JavaScript, [Microsoft](https://vi.wikipedia.org/wiki/Microsoft) bắt đầu phát triển [JScript](https://vi.wikipedia.org/wiki/JScript), một ngôn ngữ có cùng ứng dụng và tương thích với JavaScript. JScript được bổ sung vào trình duyệt [Internet Explorer](https://vi.wikipedia.org/wiki/Internet_Explorer) bắt đầu từ Internet Explorer phiên bản 3.0 được phát hành tháng 8 năm [1996](https://vi.wikipedia.org/wiki/1996).

[DOM (Document Object Model)](https://vi.wikipedia.org/wiki/DOM), một khái niệm thường được nhắc đến với JavaScript trên thực tế không phải là một phần của chuẩn ECMAScript, DOM là một chuẩn riêng biệt có liên quan chặt chẽ với XML.

JavaScript là một ngôn ngữ [lập trình dựa trên nguyên mẫu](https://vi.wikipedia.org/wiki/L%E1%BA%ADp_tr%C3%ACnh_d%E1%BB%B1a_tr%C3%AAn_nguy%C3%AAn_m%E1%BA%ABu) với cú pháp phát triển từ C. Giống như C, JavaScript có khái niệm [từ khóa](https://vi.wikipedia.org/wiki/T%E1%BB%AB_kh%C3%B3a), do đó, JavaScript gần như không thể được mở rộng.

Cũng giống như C, JavaScript không có bộ xử lý xuất/nhập (input/output) riêng. Trong khi C sử dụng [thư viện](https://vi.wikipedia.org/wiki/Th%C6%B0_vi%E1%BB%87n) xuất/nhập chuẩn, JavaScript dựa vào phần mềm ngôn ngữ được gắn vào để thực hiện xuất/nhập.

Trên [trình duyệt](https://vi.wikipedia.org/wiki/Tr%C3%ACnh_duy%E1%BB%87t_web), rất nhiều [trang web](https://vi.wikipedia.org/wiki/Trang_web) sử dụng JavaScript để thiết kế trang web động và một số hiệu ứng hình ảnh thông qua DOM. JavaScript được dùng để thực hiện một số tác vụ không thể thực hiện được với chỉ [HTML](https://vi.wikipedia.org/wiki/HTML) như kiểm tra thông tin nhập vào, tự động thay đổi hình ảnh,... Ở [Việt Nam](https://vi.wikipedia.org/wiki/Vi%E1%BB%87t_Nam), JavaScript còn được ứng dụng để làm [bộ gõ tiếng Việt](https://vi.wikipedia.org/wiki/B%E1%BB%99_g%C3%B5_ti%E1%BA%BFng_Vi%E1%BB%87t) giống như bộ gõ hiện đang sử dụng trên trang [Wikipedia tiếng Việt](https://vi.wikipedia.org/wiki/Wikipedia_ti%E1%BA%BFng_Vi%E1%BB%87t). Tuy nhiên, mỗi trình duyệt áp dụng JavaScript khác nhau và không tuân theo chuẩn [W3C](https://vi.wikipedia.org/wiki/W3C) DOM, do đó trong rất nhiều trường hợp lập trình viên phải viết nhiều phiên bản của cùng một đoạn mã nguồn để có thể hoạt động trên nhiều trình duyệt. Một số công nghệ nổi bật dùng JavaScript để tương tác với DOM bao gồm [DHTML](https://vi.wikipedia.org/wiki/HTML_%C4%91%E1%BB%99ng), [Ajax](https://vi.wikipedia.org/wiki/Ajax) và [SPA](https://vi.wikipedia.org/wiki/SPA).

Bên ngoài trình duyệt, JavaScript có thể được sử dụng trong [tập tin](https://vi.wikipedia.org/wiki/T%E1%BA%ADp_tin) [PDF](https://vi.wikipedia.org/wiki/PDF) của [Adobe Acrobat](https://vi.wikipedia.org/wiki/Adobe_Acrobat) và [Adobe Reader](https://vi.wikipedia.org/wiki/Adobe_Acrobat). Điều khiển Dashboard trên [hệ điều hành](https://vi.wikipedia.org/wiki/H%E1%BB%87_%C4%91i%E1%BB%81u_h%C3%A0nh) [Mac OS X](https://vi.wikipedia.org/wiki/Mac_OS_X) phiên bản 10.4 cũng có sử dụng JavaScript. Công nghệ [kịch bản linh động](https://vi.wikipedia.org/wiki/K%E1%BB%8Bch_b%E1%BA%A3n_linh_%C4%91%E1%BB%99ng) (active scripting) của Microsoft có hỗ trợ ngôn ngữ JScript làm một ngôn ngữ kịch bản dùng cho hệ điều hành. [JScript.NET](https://vi.wikipedia.org/wiki/JScript.NET) là một ngôn ngữ tương thích với [CLI](https://vi.wikipedia.org/wiki/CLI) gần giống JScript nhưng có thêm nhiều tính năng lập trình hướng đối tượng.

Từ khi Nodejs ra đời vào năm 2009, Javascript được biết đến nhiều hơn là một ngôn ngữ [đa nền tảng](https://vi.wikipedia.org/wiki/%C4%90a_n%E1%BB%81n_t%E1%BA%A3ng) khi có thể chạy trên cả môi trường máy chủ cũng như môi trường nhúng.

Mỗi ứng dụng này đều cung cấp mô hình đối tượng riêng cho phép tương tác với môi trường chủ, với phần lõi là ngôn ngữ lập trình JavaScript gần như giống nhau.

**2.2 Thư viện cho CSS và Javascript**

*2.1.4 Bootstrap*

A screenshot of a phone

Description automatically generated

Hình 2.4 Bootstrap

Bootstrap (trước đây là Twitter Bootstrap ) là một [khung CSS mã](https://en.wikipedia.org/wiki/CSS_framework)[nguồn mở và miễn phí](https://en.wikipedia.org/wiki/Free_and_open-source) hướng tới [phát triển web front-end đáp ứng,](https://en.wikipedia.org/wiki/Front-end_web_development)[ưu tiên thiết bị di động](https://en.wikipedia.org/wiki/Responsive_web_design#Mobile_first,_unobtrusive_JavaScript,_and_progressive_enhancement) . Nó chứa các mẫu thiết kế dựa trên [HTML](https://en.wikipedia.org/wiki/HTML) , [CSS](https://en.wikipedia.org/wiki/CSS) và (tùy chọn) [JavaScript](https://en.wikipedia.org/wiki/JavaScript) cho [kiểu chữ](https://en.wikipedia.org/wiki/Web_design#Typography) , [biểu mẫu](https://en.wikipedia.org/wiki/Form_(HTML)) , [nút](https://en.wikipedia.org/wiki/Button_(computing)#HTML) , [điều hướng](https://en.wikipedia.org/wiki/Web_navigation#Local_website_navigation) và các thành phần giao diện khác.

Tính đến tháng 5 năm 2023, Bootstrap là dự án được gắn sao nhiều thứ 17 (thư viện được gắn sao nhiều thứ 4) trên [GitHub](https://en.wikipedia.org/wiki/GitHub) , với hơn 164.000 sao. [[4]](https://en.wikipedia.org/wiki/Bootstrap_(front-end_framework)#cite_note-most-starred-4) Theo W3Techs, Bootstrap được 19,2% tổng số trang web sử dụng.

Bootstrap là một thư viện HTML, CSS và JS tập trung vào việc đơn giản hóa việc phát triển các trang web thông tin (ngược lại với [các ứng dụng web](https://en.wikipedia.org/wiki/Web_application) ). Mục đích chính của việc thêm nó vào một dự án web là áp dụng các lựa chọn về màu sắc, kích thước, phông chữ và bố cục của Bootstrap cho dự án đó. Như vậy, yếu tố chính là liệu các nhà phát triển phụ trách có tìm thấy những lựa chọn đó theo ý thích của họ hay không. Sau khi được thêm vào dự án, Bootstrap cung cấp các định nghĩa kiểu cơ bản cho tất cả [các thành phần HTML](https://en.wikipedia.org/wiki/HTML_element) . Kết quả là sự xuất hiện thống nhất cho văn xuôi, bảng biểu và các thành phần biểu mẫu trên [các trình duyệt web](https://en.wikipedia.org/wiki/Web_browser) . Ngoài ra, các nhà phát triển có thể tận dụng các lớp CSS được xác định trong Bootstrap để tùy chỉnh thêm giao diện nội dung của chúng. Ví dụ: Bootstrap đã cung cấp các bảng màu sáng và tối, tiêu đề trang, [dấu ngoặc kép](https://en.wikipedia.org/wiki/Pull_quote) nổi bật hơn và văn bản được đánh dấu.

Bootstrap cũng đi kèm với một số thành phần JavaScript không yêu cầu các thư viện khác như [jQuery](https://en.wikipedia.org/wiki/JQuery) . Chúng cung cấp các thành phần giao diện người dùng bổ sung như [hộp thoại](https://en.wikipedia.org/wiki/Dialog_box) , [chú giải công cụ](https://en.wikipedia.org/wiki/Tooltip) , thanh tiến trình, trình đơn thả xuống điều hướng và băng chuyền. Mỗi thành phần Bootstrap bao gồm một cấu trúc HTML, các khai báo CSS và trong một số trường hợp có kèm theo mã JavaScript. Chúng cũng mở rộng chức năng của một số thành phần giao diện hiện có, chẳng hạn như chức năng tự động hoàn thành cho các trường đầu vào.

Nguồn: <https://en.wikipedia.org/wiki/Bootstrap_(front-end_framework)>

**2.3 MongoDB**



Hình 2. 4 MongoDB

MongoDB là một cơ sở dữ liệu NoSQL mã nguồn mở được thiết kế để xử lý lượng dữ liệu lớn và không cấu trúc. Khác với cơ sở dữ liệu quan hệ truyền thống (như MySQL hay PostgreSQL), MongoDB lưu trữ dữ liệu dưới dạng các tài liệu JSON-like, cho phép linh hoạt hơn trong việc lưu trữ và truy vấn dữ liệu.

*Đặc điểm chính:*

* **Lưu trữ tài liệu**:
* Các tài liệu trong MongoDB được lưu trữ dưới dạng BSON (Binary JSON), giúp hỗ trợ các kiểu dữ liệu phong phú hơn.
* Mỗi tài liệu trong MongoDB có thể có một cấu trúc khác nhau, điều này giúp dễ dàng thay đổi mô hình dữ liệu mà không cần cấu trúc lại cơ sở dữ liệu.
* **Khả năng mở rộng:**
* MongoDB hỗ trợ sharding, cho phép chia nhỏ và phân phối dữ liệu trên nhiều máy chủ để tăng khả năng mở rộng.
* Điều này giúp MongoDB xử lý tốt các ứng dụng có yêu cầu lưu trữ và xử lý dữ liệu lớn.
* **Sao lưu và phục hồi**:
* MongoDB sử dụng replica sets để đảm bảo tính sẵn sàng cao và khả năng phục hồi khi có sự cố.
* Replica sets bao gồm một primary node và nhiều secondary nodes, cho phép tự động chuyển đổi khi primary node gặp sự cố.
* **Chỉ mục**:
* MongoDB hỗ trợ nhiều loại chỉ mục, bao gồm chỉ mục đơn trường, chỉ mục phức hợp và chỉ mục văn bản đầy đủ.
* Điều này giúp cải thiện hiệu suất truy vấn và tìm kiếm dữ liệu nhanh chóng hơn.
* **Tính năng không cần lược đồ cố định**:
* Không như các cơ sở dữ liệu quan hệ, MongoDB không yêu cầu xác định trước lược đồ dữ liệu, cho phép linh hoạt trong việc lưu trữ các loại dữ liệu khác nhau.
* Điều này giúp dễ dàng thay đổi cấu trúc dữ liệu mà không ảnh hưởng đến hệ thống hiện có.
* **Khả năng tổng hợp**:
* MongoDB có một framework tổng hợp mạnh mẽ, cho phép thực hiện các phép tính phức tạp và phân tích dữ liệu trực tiếp trên cơ sở dữ liệu.
* Các phép toán này bao gồm nhóm, sắp xếp, lọc và tính toán, giúp chuyển đổi và phân tích dữ liệu một cách hiệu quả

*Ứng dụng của MongoDB:*

MongoDB thường được sử dụng trong các ứng dụng cần lưu trữ và xử lý lượng dữ liệu lớn, đa dạng như:

* **Hệ thống quản lý nội dung (CMS).**
* **Ứng dụng phân tích dữ liệu lớn (Big Data).**
* **Các nền tảng truyền thông xã hội và mạng xã hội.**
* **Hệ thống quản lý người dùng và xác thực.**
* **Ứng dụng IoT (Internet of Things).**

MongoDB là một lựa chọn mạnh mẽ và linh hoạt cho các ứng dụng hiện đại yêu cầu khả năng xử lý và lưu trữ dữ liệu lớn. Nó cung cấp một bộ công cụ phong phú để làm việc với dữ liệu một cách hiệu quả và mở rộng theo nhu cầu của ứng dụng.

**2.4 Ajax**

AJAX ([tiếng Anh](https://vi.wikipedia.org/wiki/Ti%E1%BA%BFng_Anh): "Asynchronous JavaScript and XML" - nghĩa là "JavaScript và XML không đồng bộ") là một nhóm các công nghệ [phát triển web](https://vi.wikipedia.org/w/index.php?title=Ph%C3%A1t_tri%E1%BB%83n_web&action=edit&redlink=1) được sử dụng để tạo các [ứng dụng web](https://vi.wikipedia.org/wiki/%E1%BB%A8ng_d%E1%BB%A5ng_web) động hay các [ứng dụng giàu tính Internet](https://vi.wikipedia.org/w/index.php?title=%E1%BB%A8ng_d%E1%BB%A5ng_gi%C3%A0u_t%C3%ADnh_Internet&action=edit&redlink=1) (rich Internet application). Từ Ajax được ông Jesse James Garrett đưa ra và dùng lần đầu tiên vào tháng 2 năm 2005 để chỉ kỹ thuật này, mặc dù các hỗ trợ cho Ajax đã có trên các chương trình duyệt từ 10 năm trước. Ajax là một kỹ thuật phát triển [web](https://vi.wikipedia.org/wiki/World_Wide_Web) có tính tương tác cao bằng cách kết hợp các ngôn ngữ:

* [HTML](https://vi.wikipedia.org/wiki/HTML) (hoặc [XHTML](https://vi.wikipedia.org/wiki/XHTML)) với [CSS](https://vi.wikipedia.org/wiki/CSS) trong việc hiển thị thông tin
* Mô hình DOM ([Document Object Model](https://vi.wikipedia.org/wiki/DOM)), được thực hiện thông qua [JavaScript](https://vi.wikipedia.org/wiki/JavaScript), nhằm hiển thị thông tin động và tương tác với những thông tin được hiển thị
* Đối tượng XMLHttpRequest để trao đổi dữ liệu một cách không đồng bộ với máy chủ web. (Mặc dù, việc trao đổi này có thể được thực hiện với nhiều định dạng như HTML, văn bản thường, [JSON](https://vi.wikipedia.org/wiki/JavaScript_Object_Notation) và thậm chí [EBML](https://vi.wikipedia.org/w/index.php?title=EBML&action=edit&redlink=1), nhưng [XML](https://vi.wikipedia.org/wiki/XML) là ngôn ngữ thường được sử dụng).
* XML thường là định dạng cho dữ liệu truyền, mặc dù bất cứ định dạng nào cũng có thể dùng, bao gồm HTML định dạng trước, văn bản thuần (plain text), JSON và ngay cả EBML.

## **2.5 Express.js**

### *2.5.1 Khái niệm*

là một framework web dành cho Node.js, được thiết kế để xây dựng các ứng dụng web và API một cách nhanh chóng và hiệu quả. Nó rất tối giản, không áp đặt quy tắc quá chặt chẽ, và dễ tùy chỉnh theo nhu cầu.

### *2.5.2 Tính năng chính của Express.js*

* ***Middleware:***
  + Express.js sử dụng một hệ thống middleware, nơi mỗi thành phần middleware có thể xử lý yêu cầu trước khi chuyển tiếp đến thành phần tiếp theo.
  + Điều này giúp tạo ra các luồng xử lý yêu cầu phức tạp, từ xác thực, định tuyến đến xử lý lỗi.
* ***Routing:***
  + Cung cấp hệ thống định tuyến mạnh mẽ, cho phép xác định các tuyến đường (routes) và các phương thức HTTP (GET, POST, PUT, DELETE) một cách dễ dàng.
  + Hỗ trợ các biểu thức chính quy để xác định các tuyến đường động.
* ***Rendering:***
  + Hỗ trợ các công cụ render template như Pug, EJS để tạo ra các trang HTML động từ các mẫu (templates).
* ***API nhanh chóng:***
  + Express.js thường được sử dụng để xây dựng các API RESTful, nhờ vào khả năng dễ dàng xử lý các yêu cầu HTTP và JSON.
* ***Tích hợp với cơ sở dữ liệu:***
  + Dễ dàng tích hợp với các cơ sở dữ liệu khác nhau như MongoDB, MySQL, PostgreSQL, nhờ vào các module và thư viện của Node.js.

### *2.5.3 Ứng dụng của Express.js*

* **Xây dựng ứng dụng web**: Từ các trang web đơn giản đến các hệ thống quản lý nội dung phức tạp.
* **API RESTful**: Cung cấp các dịch vụ API cho các ứng dụng phía khách hàng và mobile.
* **Ứng dụng thời gian thực**: Kết hợp với WebSocket để xây dựng các ứng dụng như chat, thông báo thời gian thực.

### ***2.****5.4 Lợi ích của Express.js*

* **Đơn Giản**:  [Express.js](https://express.js/?form=MG0AV3) dễ dàng thiết lập và sử dụng, phù hợp cho cả người mới bắt đầu và các nhà phát triển có kinh nghiệm.
* **Linh Hoạt**: Cho phép các nhà phát triển lựa chọn các thành phần họ muốn sử dụng, dễ dàng tùy chỉnh.
* **Hiệu Suất Cao**: Được thiết kế để nhanh và hiệu quả, phù hợp cho các ứng dụng yêu cầu hiệu suất cao.
* **Hỗ Trợ Cộng Đồng**: Có cộng đồng lớn và sôi động, cung cấp nhiều tài nguyên, hướng dẫn và module của bên thứ ba.

**2.6 Node.js**

*2.6.1 Khái niệm*

Là một môi trường thực thi JavaScript mã nguồn mở, đa nền tảng, được phát triển để xây dựng các ứng dụng mạng nhanh và mở rộng. Được tạo ra bởi Ryan Dahl vào năm 2009, Node.js dựa trên V8 engine của Google, vốn cũng là engine chạy JavaScript trong trình duyệt Google Chrome.

*2.6.2 Đặc điểm chính của Node.js*

* **Kiến trúc không đồng bộ và sự kiện**:
  + Node.js sử dụng mô hình I/O không đồng bộ và dựa trên sự kiện, giúp xử lý nhiều kết nối đồng thời mà không gây nghẽn hiệu suất.
  + Điều này làm Node.js rất phù hợp cho các ứng dụng thời gian thực và các dịch vụ API RESTful.
* **V8 Engine**:
  + Node.js chạy trên V8 JavaScript Engine của Google, cung cấp tốc độ và hiệu suất cao.
  + V8 biên dịch mã JavaScript trực tiếp sang mã máy để thực thi nhanh hơn.
* **Quản lý gói npm (Node Package Manager)**:
  + Node.js đi kèm với npm, một hệ thống quản lý gói mạnh mẽ với hàng ngàn thư viện mã nguồn mở.
  + npm giúp dễ dàng cài đặt và quản lý các gói thư viện phục vụ cho phát triển ứng dụng.
* **Một ngôn ngữ duy nhất**:
  + Với Node.js, bạn có thể sử dụng JavaScript cho cả phía máy khách và máy chủ, giúp đồng nhất hóa mã nguồn và dễ dàng hơn trong việc quản lý dự án.
* **Khả năng mở rộng**:
  + Node.js có khả năng mở rộng cao, hỗ trợ các mô-đun và thư viện để mở rộng tính năng ứng dụng.
  + Ngoài ra, Node.js hỗ trợ microservices, giúp xây dựng các ứng dụng lớn một cách linh hoạt và quản lý dễ dàng hơn.

*2.6.3 Ứng dụng của Node.js*

* **Ứng dụng thời gian thực**: Các ứng dụng chat, công cụ cộng tác, và trò chơi thời gian thực.
* **API RESTful**: Các dịch vụ backend cho ứng dụng web và mobile.
* **Ứng dụng streaming**: Các nền tảng video, audio streaming.
* **Ứng dụng xử lý dữ liệu**: Các công cụ phân tích và xử lý dữ liệu lớn.

*2.6.4 Lợi ích của Node.js*

* **Hiệu suất cao**: Xử lý nhanh và hiệu quả với mô hình không đồng bộ.
* **Đồng nhất ngôn ngữ**: Sử dụng JavaScript cho cả phía máy khách và máy chủ.
* **Cộng đồng hỗ trợ mạnh mẽ**: Nhiều tài nguyên, thư viện và công cụ phát triển.
* **Khả năng mở rộng**: Dễ dàng mở rộng và quản lý ứng dụng.

**CHƯƠNG III: ĐẶC TẢ YÊU CẦU**

**3.1 Mô hình Use case**

*Use case tổng quát*

A diagram with text and circles

Description automatically generated with medium confidence

Hình 3. 1 use case tổng quát

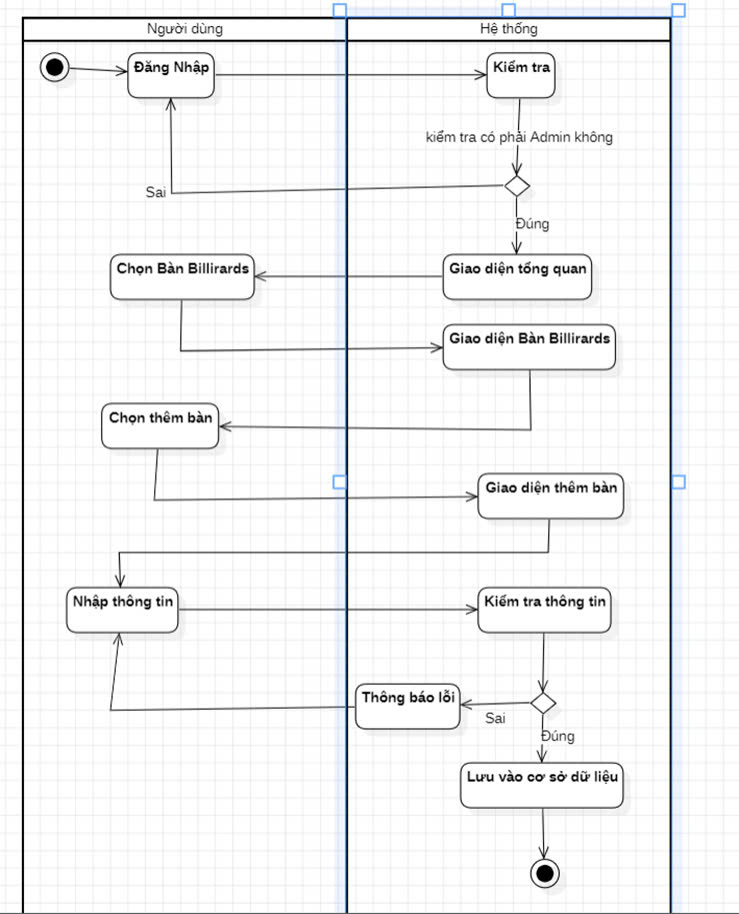
A diagram of a company

Description automatically generated

Hình 3. 2 UseCase đăng nhập

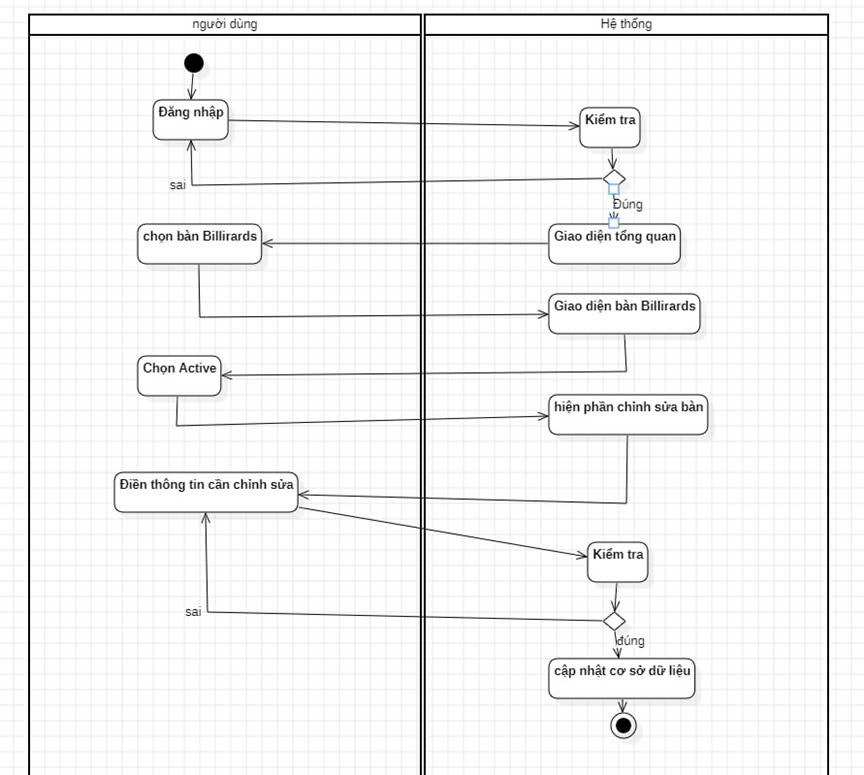
## **3.2 ActivityDiagram**

*3.2.1 Đặt bàn*



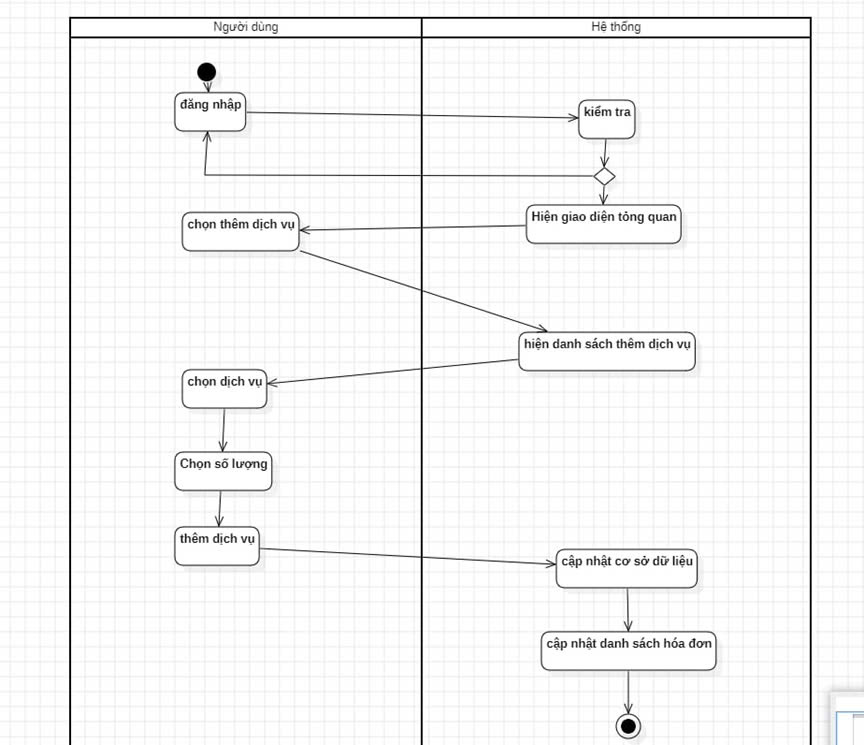
Hình 3. 3 Active diagram đặt bàn

*3.2.2 Chỉnh sửa thông tin bàn Billiards.*



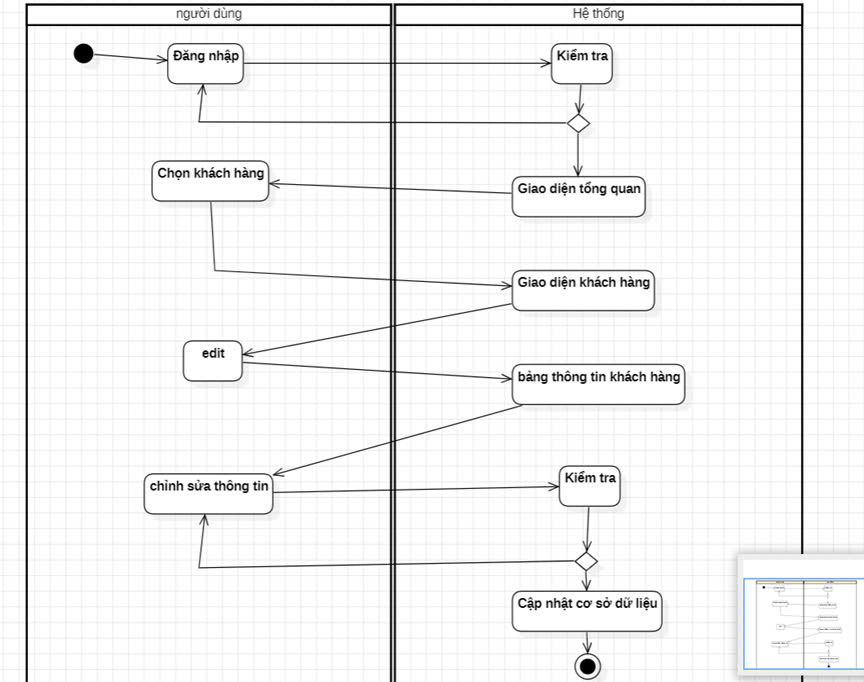
Hình 3. 4 ActivityDiagram chỉnh sửa thông tin bàn

*3.2.3 Dịch vụ*



Hình 3. 5 Dịch vụ

3.2.4 Thông tin khách hàng



Hình 3. 6 Thông tin khách hàng

**3.3 Chức năng & giao diện**

*3.3.1 Đăng nhập*

A screenshot of a computer

Description automatically generated

Hình 3. 7 đăng nhập

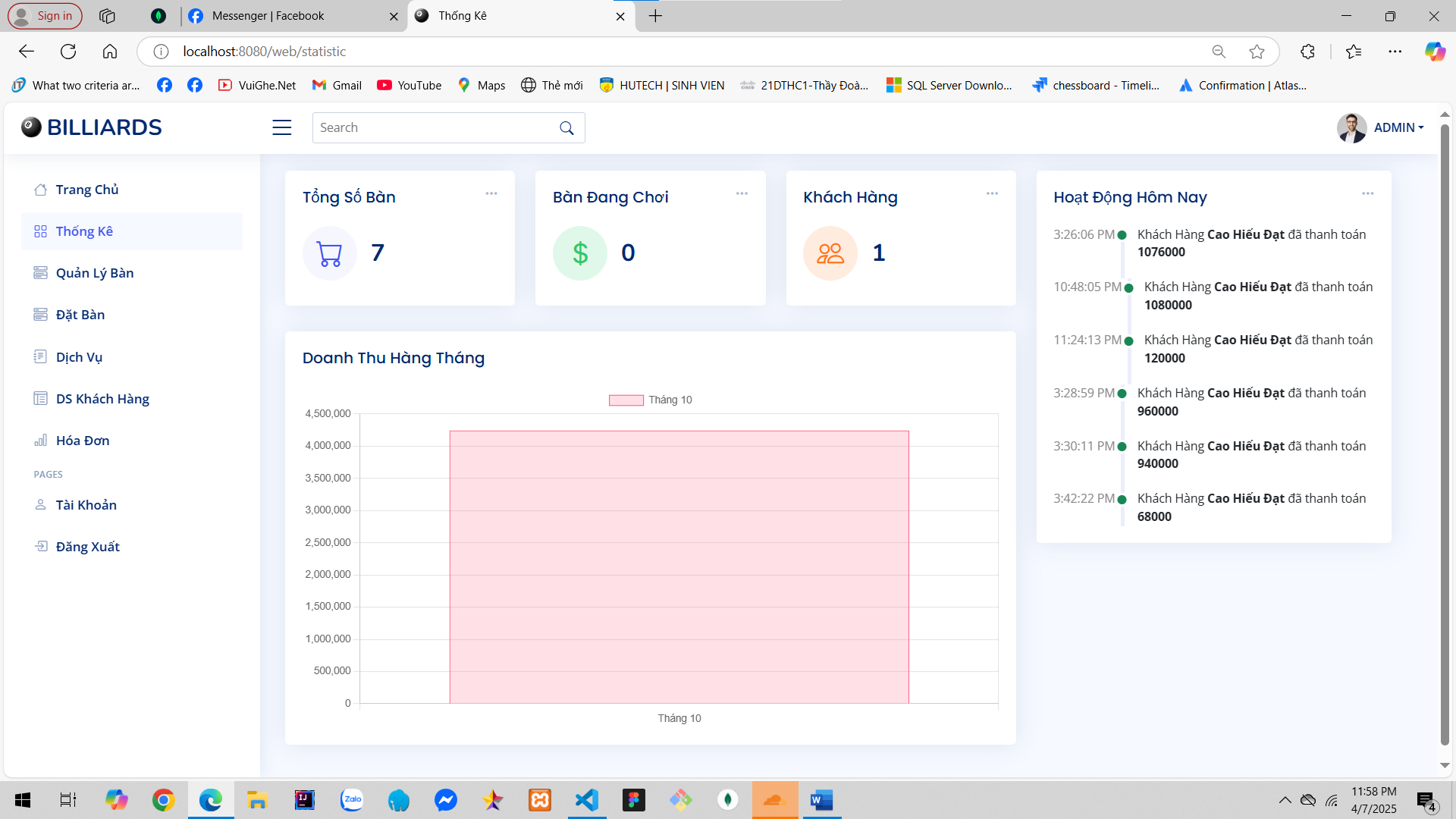
Chức năng: Khi đăng nhập với sử dụng được các chức năng khác, khi click vào các mục khác nếu chưa đăng nhập thì sẽ ở lại trang đăng nhập.

Nếu đăng nhập sai thì sẽ hiện lên thông báo cảnh báo sai tài khoản hoặc mật khẩu yêu cầu nhập lại

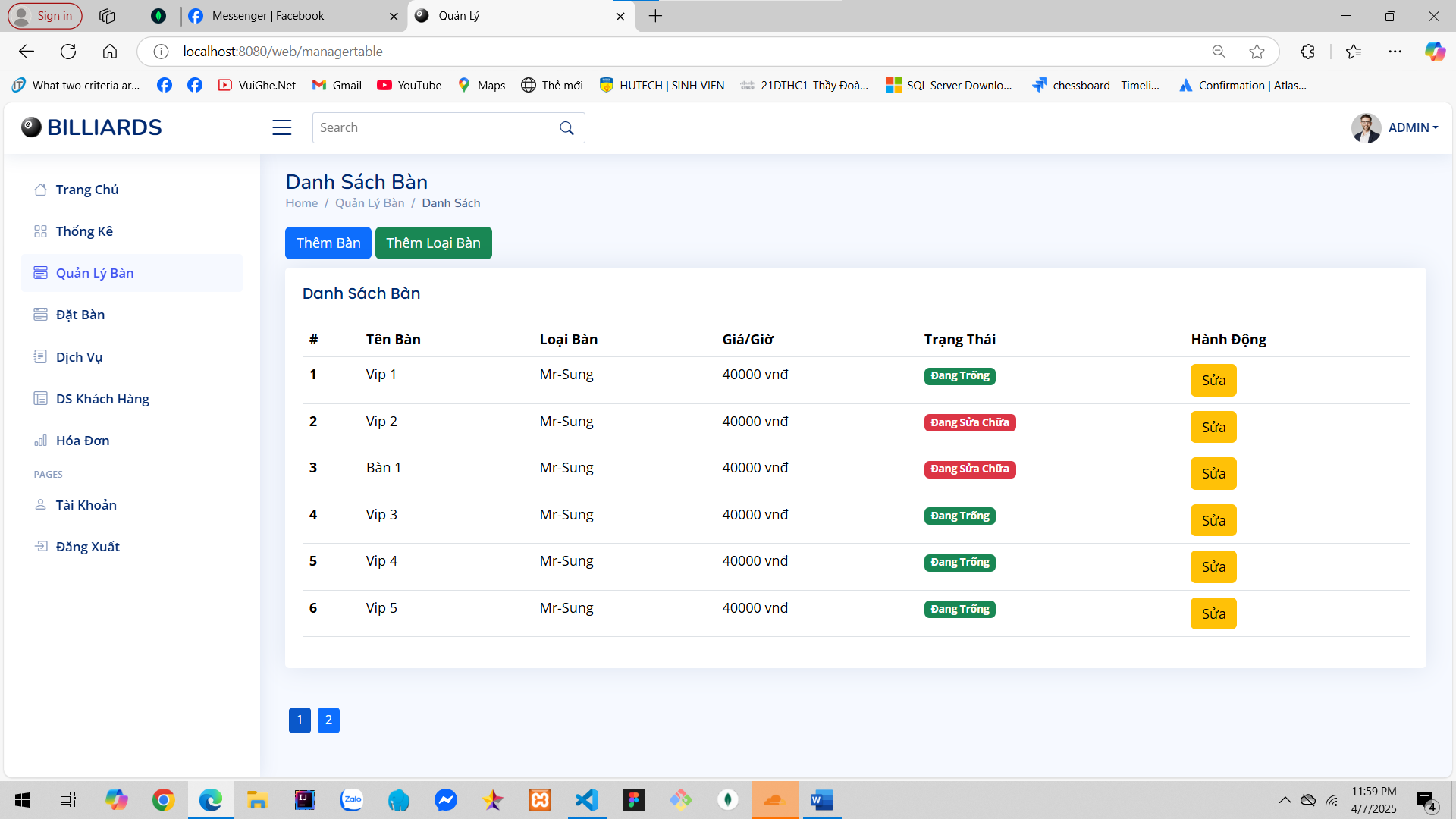
Nếu nhập đúng thông tin thì sẽ cho người dùng sử dụng các chức năng khác như xem tổng quan quản lý bàn, quản lý máy tính, xem dịch vụ doanh thu, .....

Mục đích: giúp bảo mật thông tin người dùng, ngăn chặn các hành vi tự ý đang nhập phá hoại....

*3.3.2 Thống kê*

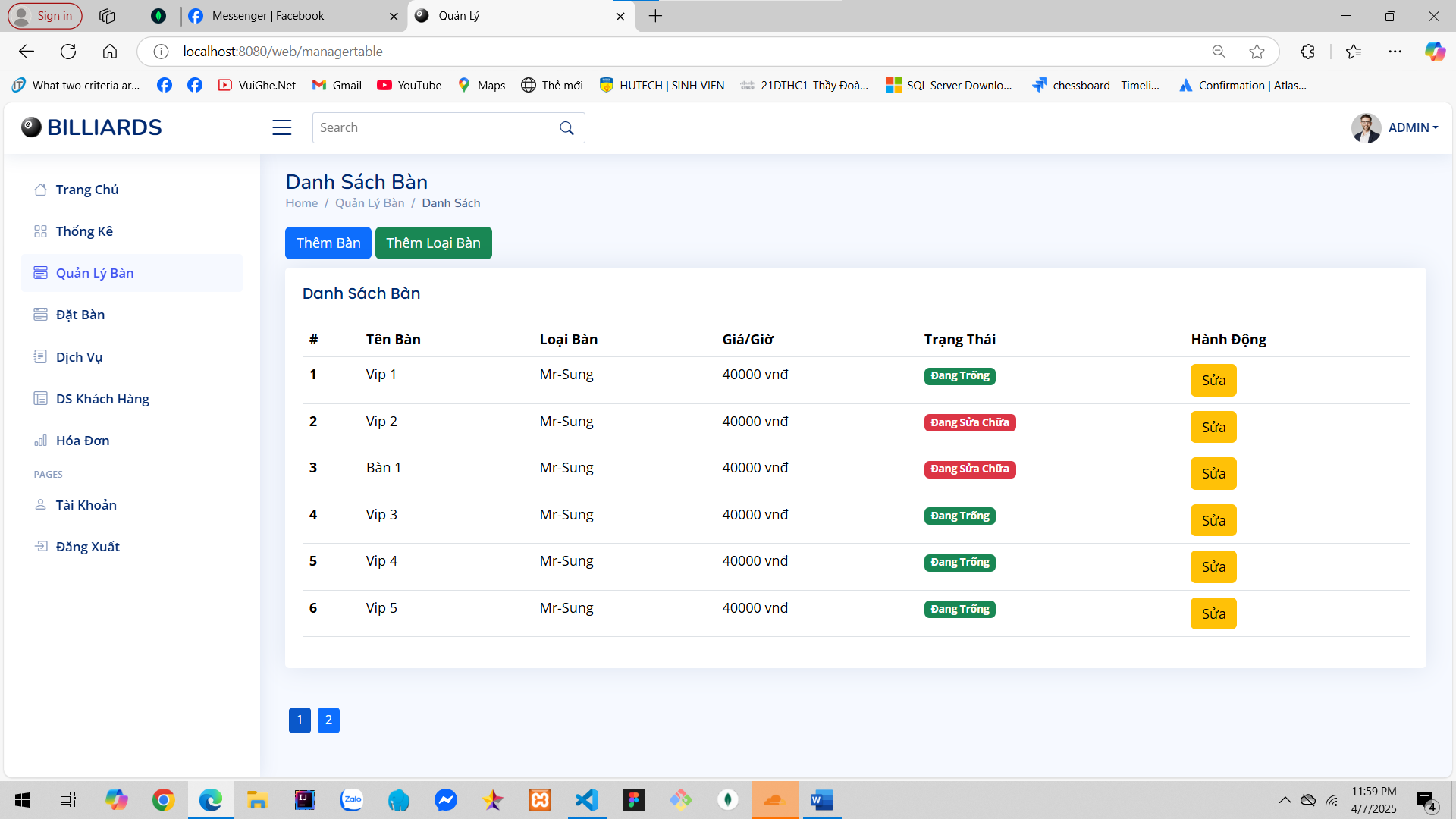


Hình 3. 8 Giao diện thống kê



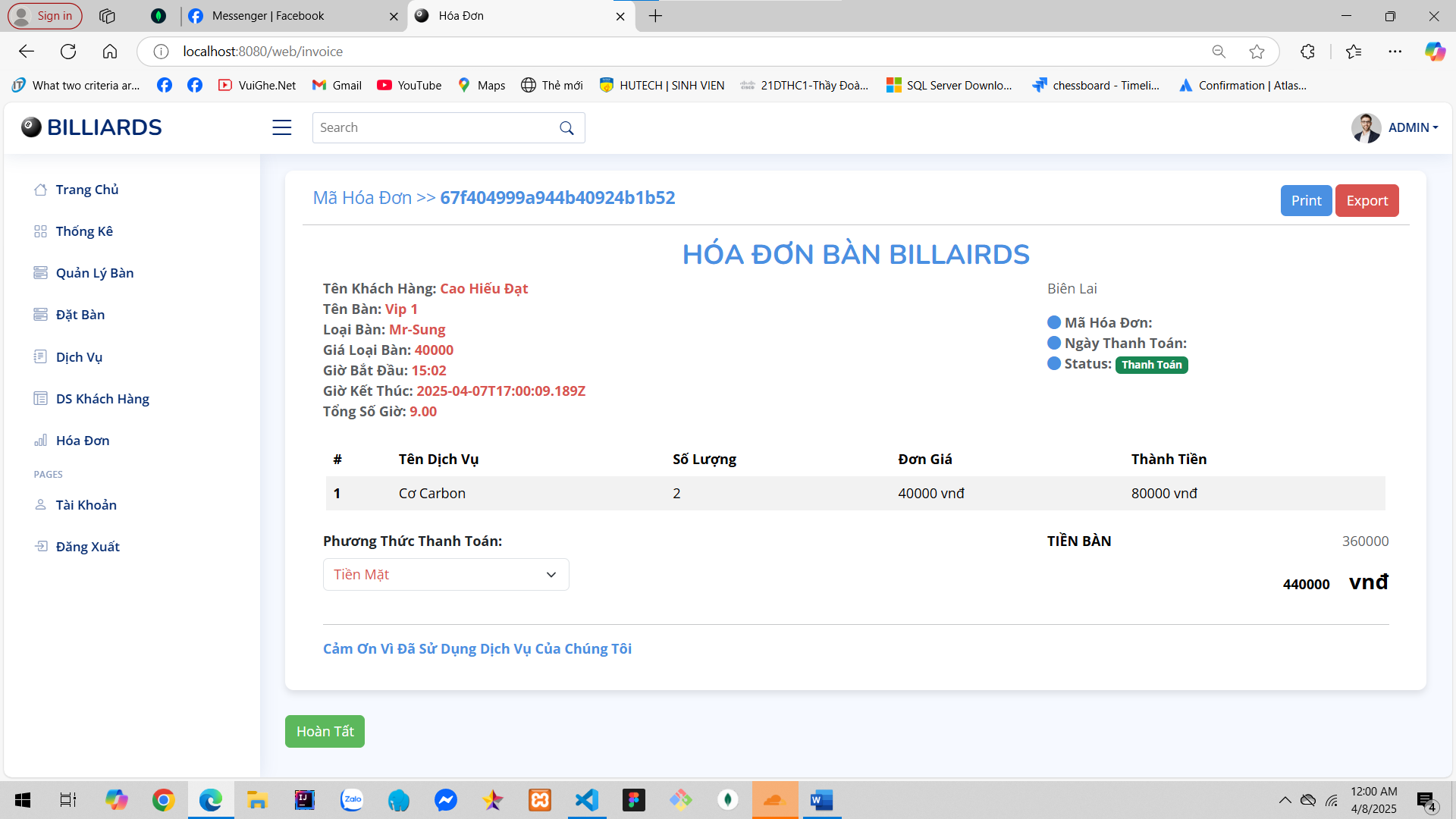
Hình 3. 9 Giao diện tổng quan

*3.3.3 Giao diện và chức năng bàn billiard*

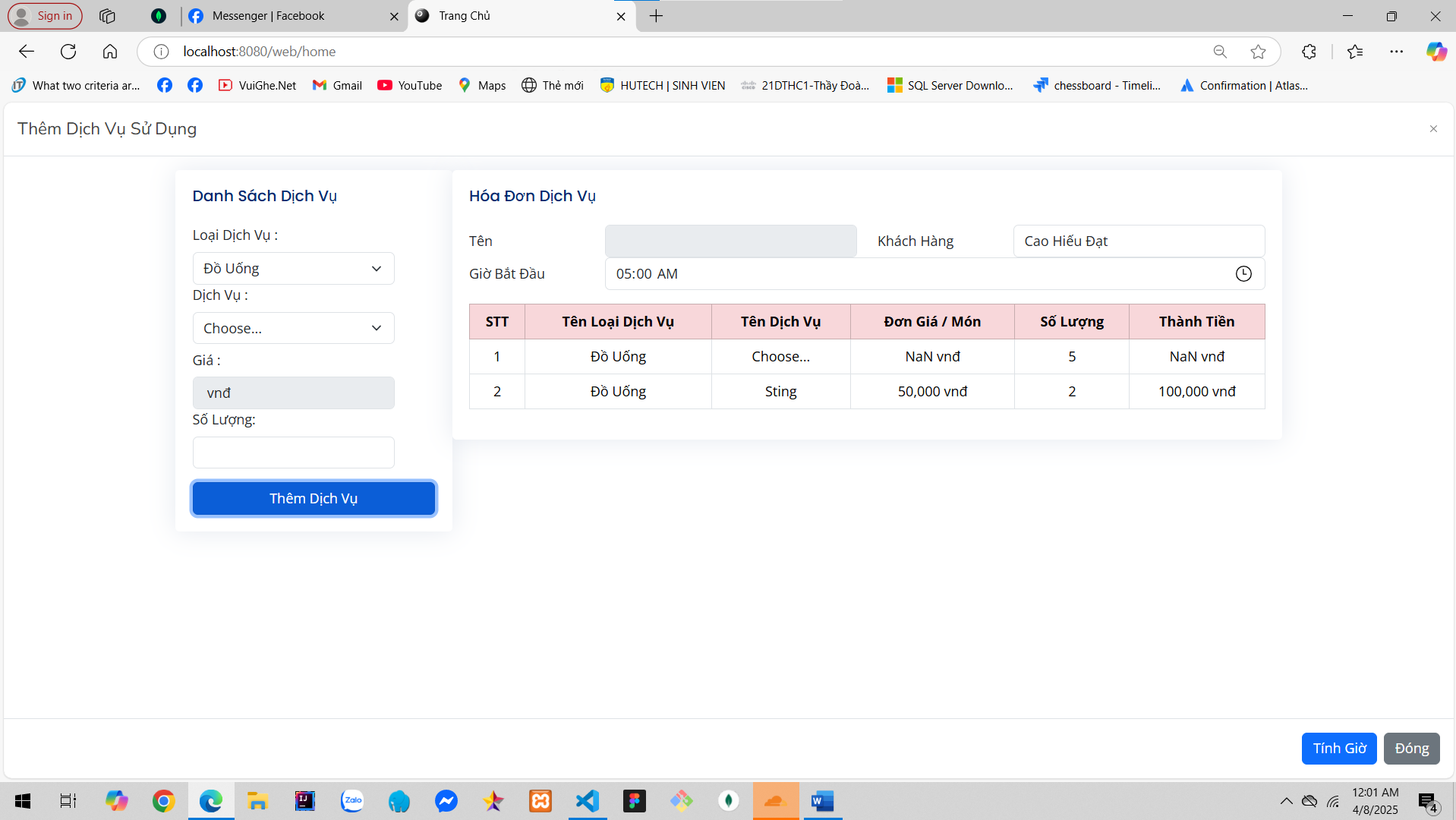


Hình 3.5 Giao diện bàn Billiards

*3.3.4 Chức năng & giao diện hóa đơn*

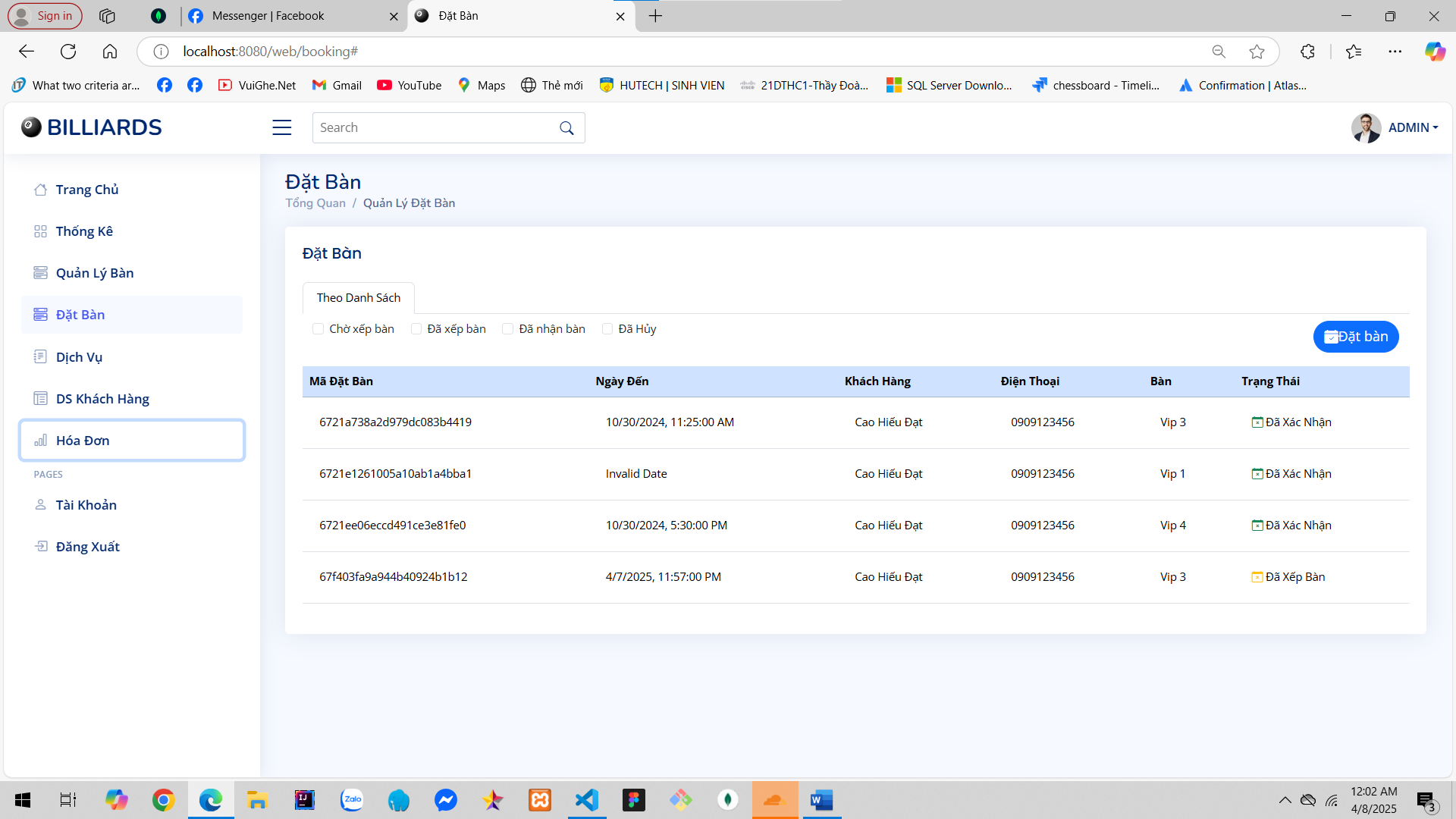


Hình 3. 10 Hóa Đơn

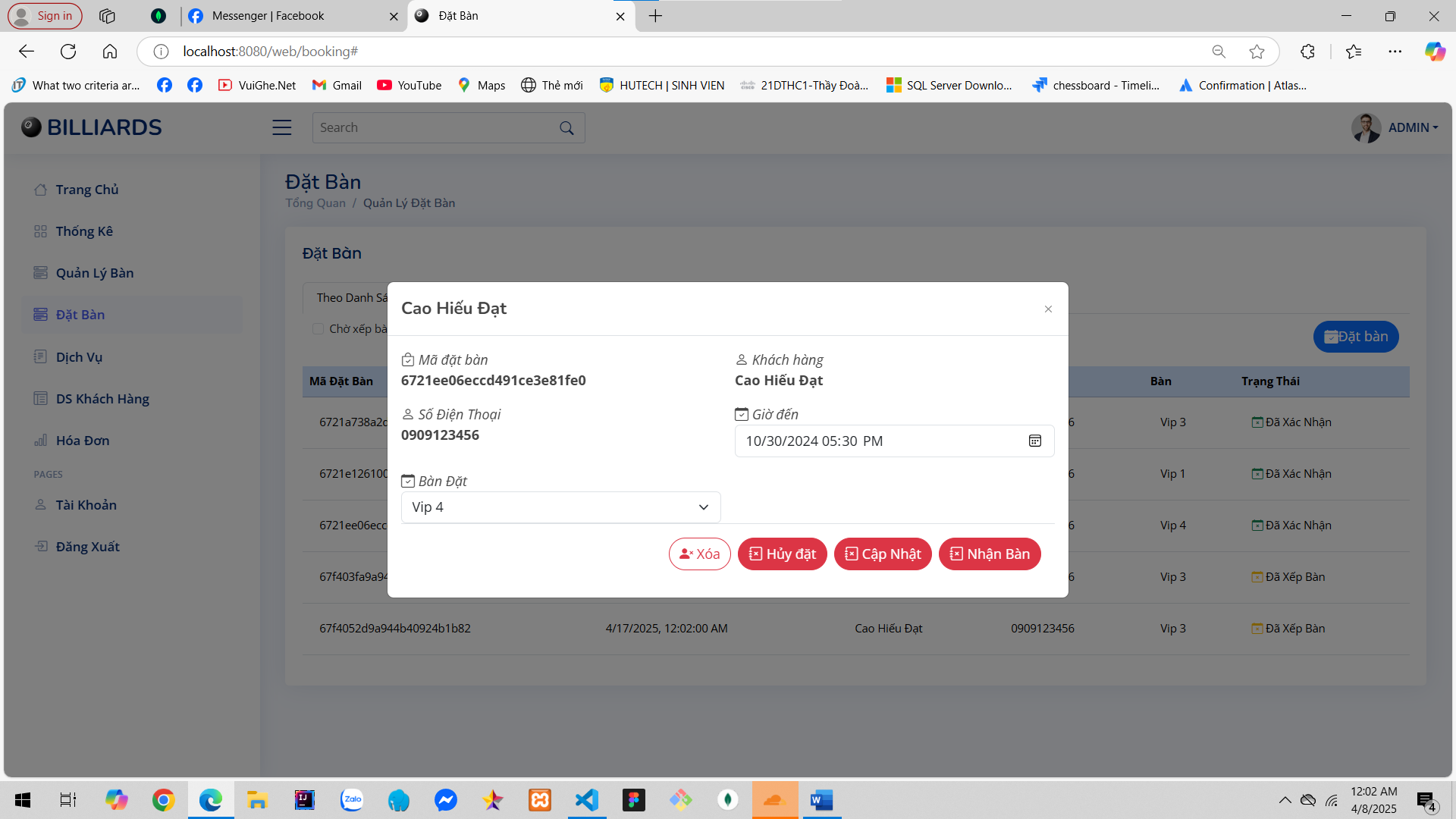


Hình 3. 11 Hóa đơn dịch vụ

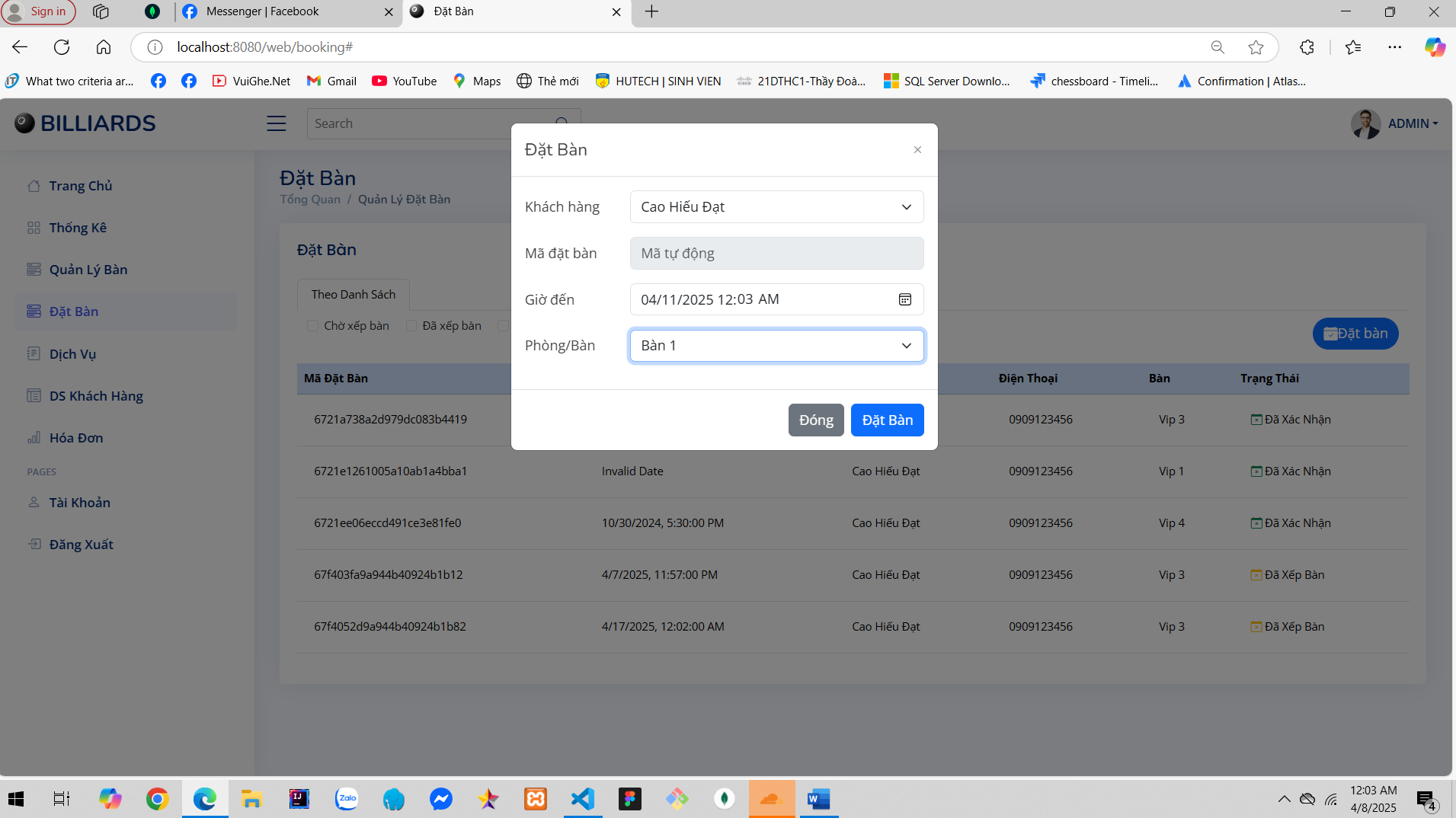
*3.3.5 Chức năng & giao diện đặt bàn.*



Hình 3. 12 Đặt bàn

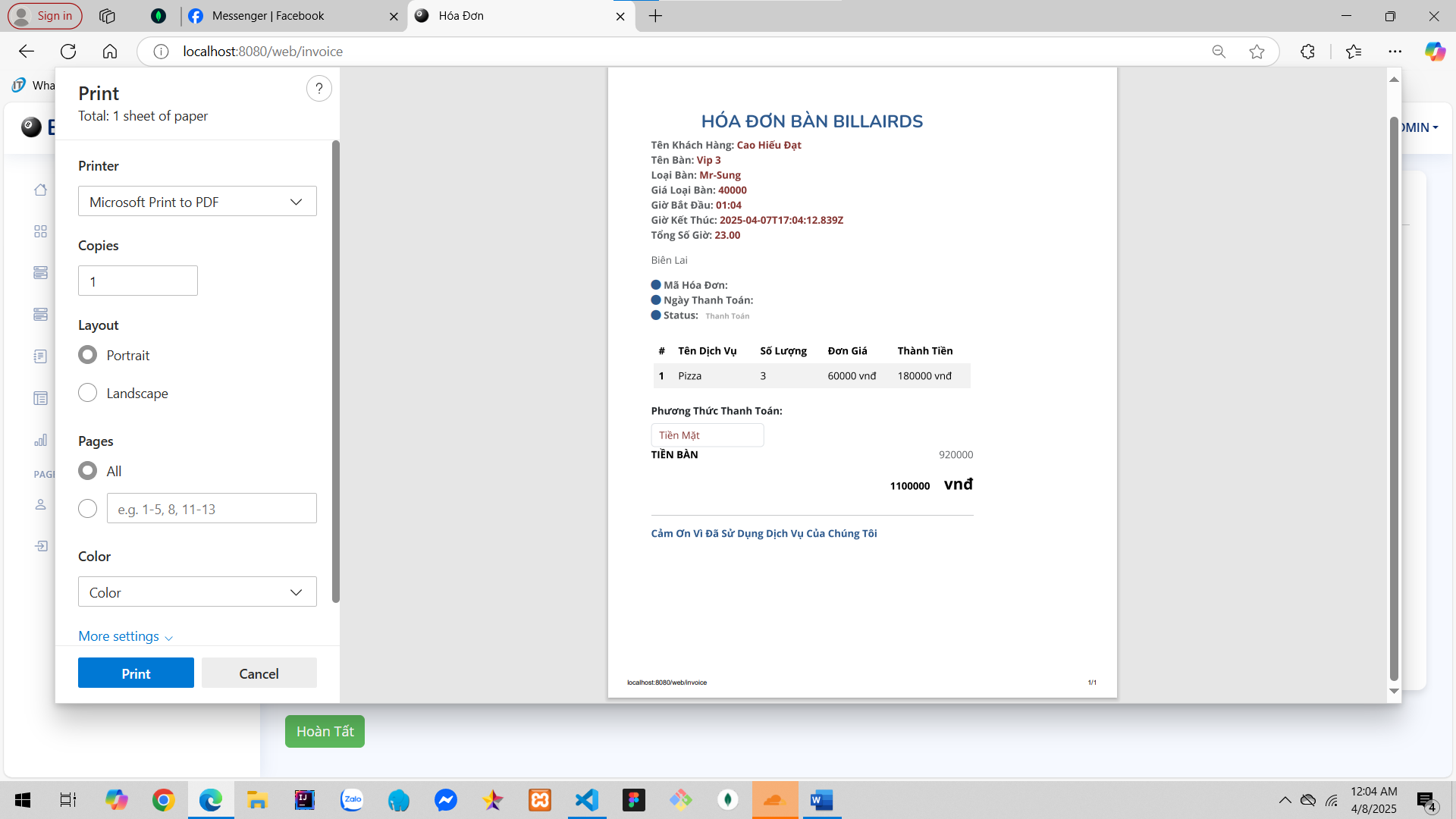


Hình 3. 13 Lưu thông tin khách đặt



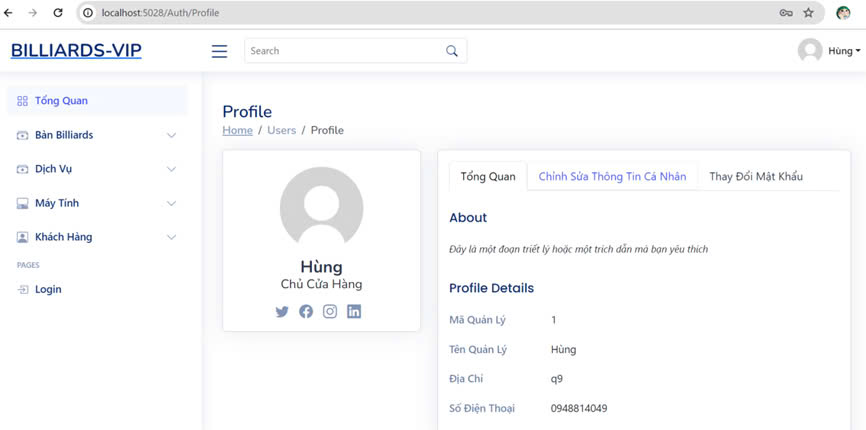
Hình 3. 14 Thông tin đặt bàn

*3.3.5 Chức năng & giao diện in hóa đơn.*



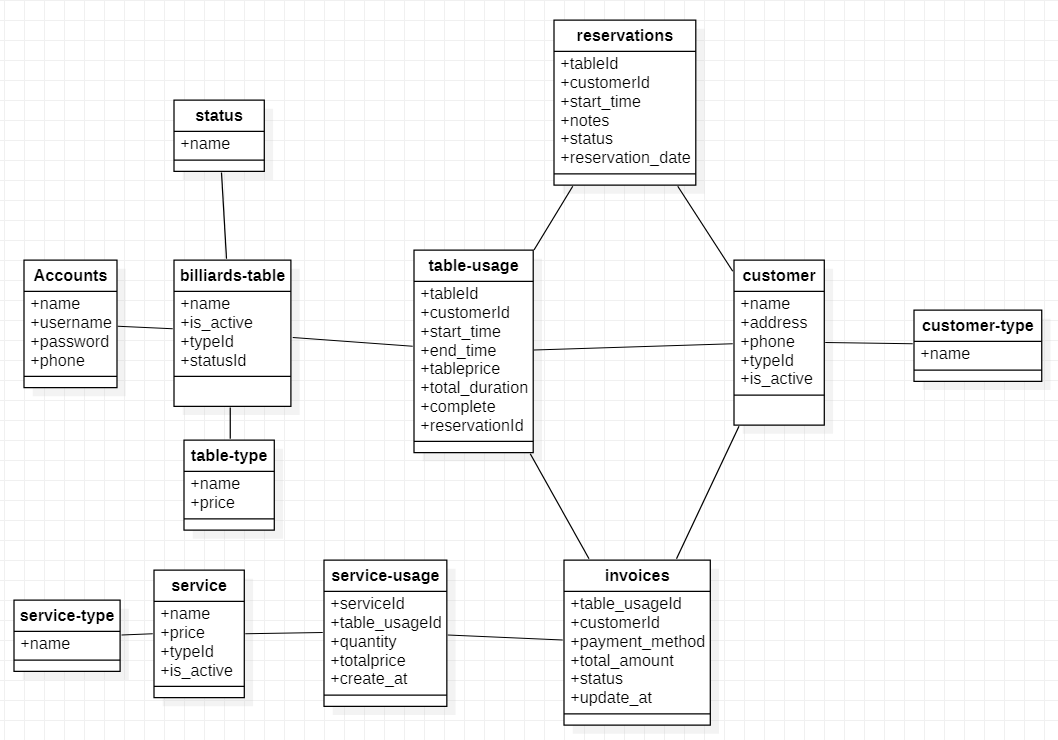
Hình 3. 15 In hóa đơn

### *3.3.6 Profile*



Hình 3. 16 Profile

**3.4 Mô hình Diagam**



Hình 3. 17 Mô hình diagram

**CHƯƠNG IV: KẾT LUẬN VÀ KIẾN NGHỊ**

**4.1 Kết quả đạt được**

Sau thời gian tìm hiểu, chọn và nghiên cứu đề tài “dịch vụ giải trí billoards”, với sự nỗ lực của cả nhóm. Đồ án cơ sở “dịch vụ giải trí billoards” của chúng em tuy còn nhiều thiếu sót, nhưng nhìn chung cũng giải quyết được phần nào nhu cầu quản lý và thanh toán đạt được những kết quả sau:

* Phân tích được quy trình hoạt động của hệ thống, các chức năng của hệ thống qua Visual Studio code và xây dựng được lớp cơ sở dữ liệu phù hợp trên mongodb.
* Xây dựng được website “dịch vụ giải trí billoards”.
* Nếu bước đầu thử nghiệm thành công chúng em sẽ tiếp tục phát triển website lên một phiên bản tốt hơn và có thể đáp ứng cho một hệ thống.
* Công việc liên quan đến việc thêm dữ liệu.
* Sau một thời gian cửa hàng phát triển và có nhu cầu mở rộng thì có thể các thông tin trước không còn hợp lí và còn tác dụng thì chúng ta có thể thực hiện các công việc như xóa, sửa.
* Trong thời gian học tập, khảo sát, phân tích - thiết kế và xây dựng hệ thống “quản lý shop hoa tươi” không thể tránh khỏi những thiếu sót nên em rất mong quý thầy và các bạn đóng góp ý kiến cho đề tài “quản lý shop hoa tươi” nhằm giúp chúng em có thể thực hiện tốt và rút ra kinh nghiệm cho những lần làm đồ án tiếp theo.

## **4.2 Hướng phát triển**

* Tiếp tục nghiên cứu, xây dựng thêm những công nghệ mới để chương trình có thể chạy tốt hơn.
* Từ những hạn chế đó, chúng em cần có những hướng phát triển để đề tài hoàn thiện hơn:
* Xây dựng các chức năng còn thiếu sót.
* Hiện thực các chức năng còn thiếu sót còn lại.
* Cập nhật thêm chương trình khuyến mãi.
* Xây dựng giao diện đẹp mắt hơn.
* Cho phép khách hàng tương tác với hệ thống nhiều hơn.
* Cập nhật các chương trình khuyến mãi.
* Nâng cao tính năng bảo mật cho hệ thống.
* Cập nhật các tính năng tìm kiếm, sắp xếp theo nhiều điều kiện của người dùng.
* Hệ thống sau khi được phát triển thành công nhóm chúng em mong muốn được đưa hệ thống vào hoạt động thực tế.

**TÀI LIỆU THAM KHẢO**

Html: <https://vi.wikipedia.org/wiki>/HTML

Css: <https://vi.wikipedia.org/wiki>/CSS

javasctript: <https://vi.wikipedia.org/wiki>/javasctipt

BootStrap: https://vi.wikipedia.org/w/index.php?title=Bootstrap&rdfrom=BootStrap

MongoDB: <https://docs.mongodb.com>

AJAX: <https://api.jquery.com/>

Express: <https://expressjs.com/>