

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



THỰC TẬP ĐỒ ÁN CHUYÊN NGÀNH  
HỌC KỲ I, NĂM HỌC 2024-2025

# XÂY DỰNG HỆ THỐNG KHỞI TẠO THIỆP CƯỜI ONLINE

*Giảng viên hướng dẫn:*  
ThS. Đoàn Phước Miền

*Sinh viên thực hiện:*  
Họ tên: Lê Thị Nguyễn Xuyên  
MSSV: 110121174  
Lớp: DA21TTC

*Trà Vinh, tháng 01 năm 2025*

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



THỰC TẬP ĐỒ ÁN CHUYÊN NGÀNH  
HỌC KỲ I, NĂM HỌC 2024-2025

# XÂY DỰNG HỆ THỐNG KHỞI TẠO THIỆP CƯỜI ONLINE

*Giảng viên hướng dẫn:*  
ThS. Đoàn Phước Miền

*Sinh viên thực hiện:*  
Họ tên: Lê Thị Nguyễn Xuyên  
MSSV: 110121174  
Lớp: DA21TTC

*Trà Vinh, tháng 01 năm 2025*

## This image shows a full page of blank handwriting practice paper. It features multiple rows of horizontal lines. Each row consists of three lines: a solid top line, a dashed middle line, and a solid bottom line. The rows are evenly spaced across the entire page, providing a guide for letter height and placement. There are no margins, text, or other markings on the paper.

(Ký tên và ghi rõ họ tên)

[illegible]

**Thành viên hội đồng**  
(Ký tên và ghi rõ họ tên)

## LỜI CẢM ƠN

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

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

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

Em xin chân thành cảm ơn.

**Lê Thị Nguyễn Xuyên**

## MỤC LỤC

CHƯƠNG 1: TỔNG QUAN.....	3
CHƯƠNG 2: NGHIÊN CỨU LÝ THUYẾT .....	4
2.1    Khái niệm và vai trò của thiệp cưới .....	4
2.2    Tổng quan về hệ thống khởi tạo và thiết kế trực tuyến.....	4
2.3    Các công nghệ sử dụng .....	4
2.3.1    Angular.....	4
2.3.2    HTML .....	6
2.3.3    CSS.....	7
2.3.4    JavaScript.....	8
2.3.5    NodeJS .....	9
2.3.6    NoSQL .....	11
2.3.7    MongoDB .....	12
2.3.8    API .....	14
CHƯƠNG 3: HIỆN THỰC HÓA NGHIÊN CỨU .....	15
3.1    Mô tả bài toán.....	15
3.2    Đặc tả bài toán.....	15
3.3    Sơ đồ usecase tổng quát .....	17
3.3.1    Sơ đồ use case quản lý mẫu thiệp .....	17
3.4    Các bước để xây dựng hệ thống .....	18
3.4.1    Cài NodeJS.....	18
3.4.2    Cài đặt Angular CLI.....	19
3.5    Tạo project Angular.....	20
3.5.1    Tạo dự án Angular .....	20
3.5.2    Tạo các Component .....	21
3.6    Cài đặt Backend với NodeJS và MongoDB .....	22
3.6.1    Cài đặt NodeJS và Express .....	22
3.6.2    Kết nối MongoDB.....	22
3.6.3    Cơ sở dữ liệu .....	25
3.7    Kiểm tra và chạy ứng dụng .....	26
CHƯƠNG 4: KẾT QUẢ NGHIÊN CỨU .....	27
4.1    Giao diện trang chủ .....	27

4.2	Giao diện chỉnh sửa thiệp cưới.....	28
4.3	Giao diện đăng nhập/đăng ký.....	28
4.4	Giao diện lưu thiệp cưới đã chỉnh sửa.....	29
4.5	Giao diện lưu thiệp cưới yêu thích.....	30
4.6	Giao diện đăng nhập Admin.....	30
4.7	Giao diện thống kê .....	31
4.8	Giao diện thêm thiệp cưới .....	31
4.9	Giao diện danh sách thiệp cưới .....	32
CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN.....		33
5.1	Kết luận .....	33
5.2	Hướng phát triển .....	33
DANH MỤC TÀI LIỆU THAM KHẢO .....		34

## DANH MỤC HÌNH ẢNH

Hình 2. 1 Angular.....	5
Hình 2. 2 Ngôn ngữ HTML .....	6
Hình 2. 3 Ngôn ngữ CSS.....	7
Hình 2. 4 Ngôn ngữ JavaScript.....	8
Hình 2. 5 NodeJS .....	9
Hình 2. 6 MongoDB.....	12
Hình 3. 1 Biểu đồ use case tổng quát.....	17
Hình 3. 2 Biểu đồ use case quản lý thiệp cưới.....	17
Hình 3. 3 Biểu đồ use case xem các mẫu thiệp cưới.....	18
Hình 3. 4 Biểu đồ use case tùy chỉnh thiệp cưới.....	18
Hình 3. 5 Biểu đồ use case lưu thiệp cưới .....	18
Hình 4. 1 Giao diện trang chủ.....	27
Hình 4. 2 Giao diện chỉnh sửa thiệp cưới .....	28
Hình 4. 3 Giao diện đăng nhập/ đăng ký.....	29
Hình 4. 4 Giao diện lưu thiệp cưới đã chỉnh sửa .....	29
Hình 4. 5 Giao diện lưu thiệp cưới yêu thích.....	30
Hình 4. 6 Giao diện đăng nhập admin .....	30
Hình 4. 7 Giao diện thống kê .....	31
Hình 4. 8 Giao diện thêm thiệp cưới.....	31
Hình 4. 9 Giao diện danh sách thiệp cưới .....	32



## TÓM TẮT ĐỒ ÁN CHUYÊN NGÀNH

### 1. Vấn đề nghiên cứu

Hiện nay, công nghệ thông tin phát triển mạnh mẽ đã làm cho việc làm thiệp cưới trực tuyến trở nên phổ biến. Tuy nhiên, việc tìm kiếm các dịch vụ thiết kế thiệp cưới tùy chỉnh thường đòi hỏi nhiều thời gian và chi phí. Vì vậy, hệ thống khởi tạo thiệp cưới online này được xây dựng nhằm giúp người dùng tự thiết kế và tùy chỉnh thiệp cưới nhanh chóng, tiện lợi và tiết kiệm, đáp ứng nhu cầu cá nhân hóa của người sử dụng.

### 2. Các hướng tiếp cận

Phân tích yêu cầu người dùng: Tìm hiểu các tính năng cần thiết cho hệ thống từ góc nhìn của người sử dụng như khả năng chọn mẫu thiệp, tùy chỉnh màu sắc, nội dung, hình ảnh.

Xây dựng hệ thống thiết kế: Phát triển giao diện thân thiện, dễ sử dụng, đơn giản giúp người dùng có thể tự thiết kế thiệp cưới theo ý muốn một cách dễ dàng.

Công nghệ sử dụng: Kết hợp các công nghệ web hiện đại như HTML5, CSS3, JavaScript để tạo giao diện và backend Node.js để xử lý dữ liệu và quản lý thông tin.

### 3. Kết quả đạt được

- Xây dựng thành công một hệ thống cho phép người dùng tạo và lưu thiệp cưới của riêng mình.
- Nâng cao kỹ năng sử dụng các ngôn ngữ lập trình và công nghệ web hiện đại trong việc xây dựng và triển khai hệ thống online.

## MỞ ĐẦU

### 1. Lý do chọn đề tài

Trong thời đại công nghệ phát triển, nhu cầu cá nhân hóa thiệp cưới đang trở thành xu hướng nhưng việc thiết kế tùy chỉnh thường mất nhiều thời gian và chi phí. Hệ thống thiệp cưới online giúp giải quyết vấn đề này bằng cách cung cấp nền tảng tiện lợi cho người dùng tự thiết kế thiệp theo sở thích.

Tuy nhiên, việc tìm kiếm dịch vụ thiết kế thiệp cưới tùy chỉnh có thể là một quá trình tốn kém, phức tạp và mất nhiều thời gian, đặc biệt khi phải làm việc trực tiếp với các đơn vị thiết kế hoặc in ấn. Xu hướng làm thiệp cưới trực tuyến ra đời như một giải pháp tối ưu, đáp ứng nhu cầu về sự tiện lợi và tiết kiệm chi phí. Thay vì phải đến các cửa hàng, người dùng có thể dễ dàng tạo mẫu thiệp cưới của riêng mình thông qua hệ thống trực tuyến với đầy đủ công cụ và tùy chọn cá nhân hóa.

Vì vậy, đề tài xây dựng hệ thống khởi tạo thiệp cưới trực tuyến được chọn nhằm mang đến một nền tảng hỗ trợ người dùng tự thiết kế thiệp cưới một cách linh hoạt, tiện lợi và tiết kiệm thời gian.

### 2. Mục tiêu nghiên cứu

- Nghiên cứu về giao diện và tính năng đáp ứng người dùng.
- Phát triển công nghệ với Angular, Node.js và MongoDB để xây dựng hệ thống.

### 3. Đối tượng nghiên cứu

- Người dùng chuẩn bị kết hôn và có nhu cầu tự thiết kế thiệp cưới theo phong cách riêng.
- Các công cụ và ngôn ngữ lập trình dùng trong quá trình phát triển hệ thống, như Angular cho giao diện, Node.js cho backend, và MongoDB cho cơ sở dữ liệu.
- Hệ thống khởi tạo thiệp cưới online với các tính năng cá nhân hóa đa dạng.

### 4. Phạm vi nghiên cứu

- Nghiên cứu và phát triển dựa trên các công nghệ web hiện đại bao gồm Angular, Node.js và MongoDB.

- Tập trung vào các tính năng chính của hệ thống nhằm đáp ứng nhu cầu của người dùng trong việc tự thiết kế và cá nhân hóa thiệp cưới.

## **CHƯƠNG 1: TỔNG QUAN**

### **1.1 Đặt vấn đề**

Hiện nay, cá nhân hóa và tự thiết kế các sản phẩm cưới, đặc biệt là thiệp cưới, đã trở thành xu hướng được nhiều cặp đôi yêu thích. Thiệp cưới không chỉ đơn thuần là một lời mời mà còn là biểu tượng thể hiện phong cách và cá tính riêng của cặp đôi, tạo nên dấu ấn đặc biệt cho ngày trọng đại của họ. Tuy nhiên, việc sử dụng các dịch vụ thiết kế thiệp cưới truyền thống thường đòi hỏi nhiều thời gian, chi phí và không cho phép người dùng tự do sáng tạo theo ý tưởng của mình. Vì vậy, xây dựng một hệ thống khởi tạo thiệp cưới online là điều cần thiết để đáp ứng nhu cầu này.

Hệ thống khởi tạo thiệp cưới online sẽ giúp các cặp đôi dễ dàng sáng tạo, thay đổi nội dung, màu sắc, hình ảnh, mang đến trải nghiệm thiết kế tiện lợi mà không yêu cầu kiến thức chuyên sâu về thiết kế. Với giao diện thân thiện, đơn giản, hệ thống sẽ hỗ trợ người dùng sử dụng dễ dàng, giúp tiết kiệm thời gian và chi phí.

### **1.2 Mục đích nghiên cứu**

- Xây dựng hệ thống khởi tạo thiệp cưới online, giúp các cặp đôi dễ dàng tạo ra thiệp cưới theo sở thích cá nhân, thể hiện phong cách riêng của họ qua từng thiết kế, biến mỗi thiệp mời thành dấu ấn độc đáo và ý nghĩa.
- Giúp người dùng tiết kiệm thời gian và chi phí trong việc tạo thiệp cưới mà vẫn đảm bảo tính cá nhân hóa và sự khác biệt cho mỗi sản phẩm.

## CHƯƠNG 2: NGHIÊN CỨU LÝ THUYẾT

### 2.1 Khái niệm và vai trò của thiệp cưới

Thiệp cưới là một phần không thể thiếu trong lễ cưới, không chỉ đóng vai trò như một lời mời mà còn là phương tiện truyền tải thông điệp, phong cách, và cá tính của cặp đôi đến với khách mời. Thiệp cưới có ý nghĩa tượng trưng cho sự trang trọng và quan trọng của ngày lễ, đồng thời thể hiện văn hóa và truyền thống gia đình.

Với sự phát triển của công nghệ và nhu cầu thể hiện cái tôi cá nhân, việc cá nhân hóa các sản phẩm, đặc biệt là thiệp cưới, ngày càng được ưa chuộng. Việc tự tay thiết kế thiệp cưới cho phép cặp đôi tạo ra những mẫu thiệp độc đáo, phản ánh phong cách và cá tính của họ, tạo dấu ấn riêng biệt cho ngày trọng đại.

### 2.2 Tổng quan về hệ thống khởi tạo và thiết kế trực tuyến

Hệ thống thiết kế trực tuyến giúp “người dùng dễ dàng tùy chỉnh và sáng tạo sản phẩm theo sở thích cá nhân mà không cần sự hỗ trợ chuyên môn.

### 2.3 Các công nghệ sử dụng

#### 2.3.1 Angular

Angular là một mã nguồn mở viết bằng TypeScript và được sử dụng để thiết kế giao diện web (front – end). Angular được xây dựng, phát triển từ những năm 2009 và đang duy trì cho đến nay bởi Google. Đây được xem là framework front end mạnh mẽ và chuyên dụng dành cho các lập trình viên sử dụng HTML cao cấp.

Angular được ứng dụng rộng rãi để xây dựng các project Single Page Application (ứng dụng trang đơn) [1].



Hình 2. 1 Angular

Phần mềm này được xây dựng, phát triển và cho ra đời vào năm 2009 bởi Misko Hevery cùng với một người bạn khác là Adam Abrons. Angular được coi là một dự án riêng cho đến khi Misko Hevery tham gia vào dự án Google Feedback với tư cách là lập trình viên bán thời gian. Trong thời gian khoảng 6 tháng, Misko cùng 2 người khác nữa đã viết lên 17.000 dòng mã khác nhau cho dự án Google Feedback.

#### **Ưu điểm:**

- Angular được các chuyên gia đánh giá cao, mã nguồn này giúp các Single Page Application làm việc dễ dàng, nhanh chóng.
- Nhờ khả năng Binding data lên trên các nền tảng HTML nên code front-end thường rất thân thiện với người dùng.
- Bạn có thể thuận tiện Unit Test.
- Component có thể tái sử dụng dễ dàng hơn.
- Angular có khả năng hỗ trợ cho các lập trình viên có thể viết code được ít hơn cùng với nhiều chức năng hơn. Từ đó giúp tiết kiệm thời gian lập trình và tăng hiệu suất công việc.
- AngularJS tương thích với nhiều nền tảng khác nhau. Bạn có thể dùng được trên nhiều loại trình duyệt khác nhau cả trong máy tính và thiết bị điện thoại di động.

### 2.3.2 HTML



Hình 2. 2 Ngôn ngữ HTML

HTML là viết tắt của cụm từ Hypertext Markup Language (tạm dịch là Ngôn ngữ đánh dấu siêu văn bản) [2]. HTML được sử dụng để tạo và cấu trúc các thành phần trong trang web hoặc ứng dụng, phân chia các đoạn văn, heading, titles, blockquotes,...

HTML được tạo ra bởi Tim Berners-Lee, một nhà vật lý học của trung tâm nghiên cứu CERN ở Thụy Sĩ. Vào thời điểm hiện tại, HTML đã trở thành một chuẩn Internet được tổ chức W3C (World Wide Web Consortium) vận hành và phát triển. Bạn có thể tự tìm kiếm trạng thái mới nhất của HTML tại bất kỳ thời điểm nào trên Website của W3C.

Chỉ được áp dụng trọng điểm cho Website tĩnh. Nếu muốn tạo các tính năng động, lập trình viên phải sử dụng thêm JavaScript hoặc ngôn ngữ backend của bên thứ 3 (ví dụ như: PHP).

HTML có thể được hỗ trợ bởi các ngôn ngữ khác như CSS hoặc JavaScript.

Một tài liệu HTML được hình thành bởi các phần tử HTML (HTML Elements) được quy định bằng các cặp thẻ (tag và attributes). Các cặp thẻ này được bao bọc bởi một dấu ngoặc nhọn (ví dụ <html>) và thường là sẽ được khai báo thành một cặp, bao gồm thẻ mở và thẻ đóng (ví dụ <html> ... </html>).

Các trình duyệt web nhận tài liệu HTML từ một web server hoặc một kho lưu trữ cục bộ và render tài liệu đó thành các trang web đa phương tiện. HTML mô

tả cấu trúc của một trang web về mặt ngữ nghĩa và các dấu hiệu ban đầu được bao gồm cho sự xuất hiện của tài liệu.

Mỗi trang HTML chứa một bộ các tag (cũng được gọi là elements). Mỗi thẻ sẽ có những tác dụng nhất định, giúp xây dựng nên một cấu trúc hoàn chỉnh cho Website. Bạn có thể xem như là việc xây dựng từng khối của một trang web. Nó tạo thành cấu trúc cây thư mục bao gồm section, paragraph, heading, và những khối nội dung khác.

### 2.3.3 CSS



Hình 2. 3 Ngôn ngữ CSS

CSS là viết tắt của Cascading Style Sheets, nó là một ngôn ngữ được sử dụng để tìm và định dạng lại các phần tử được tạo ra bởi các ngôn ngữ đánh dấu HTML. Nói ngắn gọn hơn là ngôn ngữ tạo phong cách cho trang web. Bạn có thể hiểu đơn giản rằng, nếu HTML đóng vai trò định dạng các phần tử trên website như việc tạo ra các đoạn văn bản, các tiêu đề, bảng,...thì CSS sẽ giúp chúng ta có thể thêm style vào các phần tử HTML đó như đổi bố cục, màu sắc trang, đổi màu chữ, font chữ, thay đổi cấu trúc...

CSS được phát triển bởi W3C (World Wide Web Consortium) vào năm 1996, vì HTML không được thiết kế để gắn tag để giúp định dạng trang web.



Phương thức hoạt động của CSS là nó sẽ tìm dựa vào các vùng chọn, vùng chọn có thể là tên một thẻ HTML, tên một ID, class hay nhiều kiểu khác. Sau đó là nó sẽ áp dụng các thuộc tính cần thay đổi lên vùng chọn đó.

Mối tương quan giữa HTML và CSS rất mật thiết. HTML là ngôn ngữ markup (nền tảng của site) và CSS định hình phong cách (tất cả những gì tạo nên giao diện website), chúng là không thể tách rời [3].

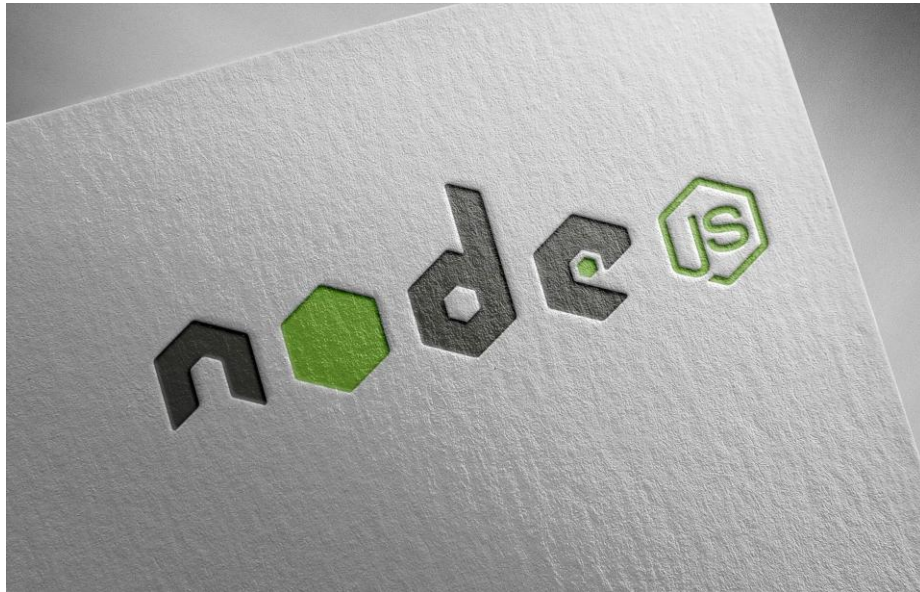
#### **2.3.4 JavaScript**

Javascript được biết đến lần đầu tiên vào tháng 9 năm 1995 được tạo ra bởi một nhân viên Netscape tên là Brendan Eich . Javascript là một ngôn ngữ lập trình website, được tích hợp và nhúng trong HTML và giúp cho website trở nên sống động hơn. Đây là dạng ngôn ngữ theo kịch bản, được hình thành dựa trên chính đối tượng phát triển có sẵn hoặc đơn giản là tự định nghĩa ra. Javascript cho phép bạn kiểm soát các hành vi của trang web tốt hơn so với việc chỉ sử dụng mỗi HTML. Có thể kể đến một vài ứng dụng thực tiễn cực kỳ quen thuộc của Javascript như slideshow, pop-up quảng cáo hay tính năng autocomplete của Google,... chúng đều được viết bằng ngôn ngữ Javascript.



Hình 2. 4 Ngôn ngữ JavaScript

### 2.3.5 NodeJS

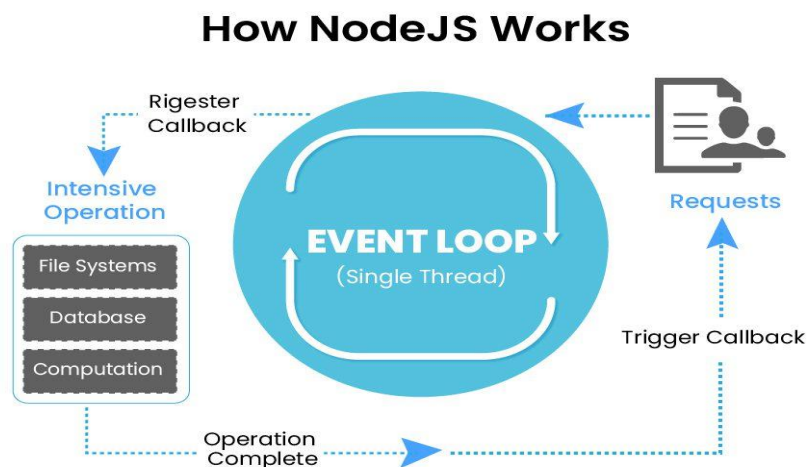


Hình 2. 5 NodeJS

NodeJS là một nền tảng được xây dựng trên “V8 Javascript engine” được viết bằng c++ và Javascript. Nền tảng này được phát triển bởi Ryan Lienhart Dahl vào năm 2009 [4].

NodeJS ra đời khi các developer đòi đầu của JavaScript mở rộng nó từ một thứ bạn chỉ chạy được trên trình duyệt thành một thứ bạn có thể chạy trên máy của mình dưới dạng ứng dụng độc lập.

**NodeJS hoạt động ra sao:**



NodeJS hoạt động dựa trên một số nguyên tắc cơ bản giúp nó hiệu quả trong việc xử lý các ứng dụng có nhiều hoạt động nhập/xuất (I/O) mà không bị chặn,

đồng thời giảm đáng kể sự phức tạp trong quản lý các luồng thực thi. Dưới đây là một số thành phần chính giải thích cách thức hoạt động của NodeJS:

### Kiến trúc Non-blocking I/O và Event-Driven:

NodeJS sử dụng một mô hình non-blocking I/O (input/output) và event-driven, nghĩa là các hoạt động như đọc file, truy vấn cơ sở dữ liệu, hoặc giao tiếp mạng được thực hiện mà không chặn tiến trình chính. Điều này cho phép xử lý nhiều yêu cầu cùng lúc mà không cần tạo nhiều luồng (thread), giúp giảm bớt chi phí liên quan đến quản lý luồng và tối ưu hóa hiệu suất.

Khi một hoạt động I/O được khởi tạo, nó sẽ được gửi đến thực thi trong hệ thống hoặc cơ sở dữ liệu mà không làm chậm tiến trình chính. Sau khi hoạt động hoàn tất, một sự kiện sẽ được phát đi và xử lý bằng các hàm gọi lại (callback).

### V8 JavaScript Engine

NodeJS được xây dựng trên động cơ JavaScript V8 của Google Chrome, đây là một động cơ rất nhanh cho phép biên dịch mã JavaScript thành mã máy để thực thi trực tiếp trên phần cứng, làm tăng hiệu suất thực thi.

### Single-Threaded

Mặc dù NodeJS hoạt động trên một luồng duy nhất cho logic ứng dụng của người dùng, nó vẫn sử dụng nhiều luồng ở tầng thấp hơn thông qua thư viện libuv để xử lý các hoạt động I/O. Tuy nhiên, những chi tiết này được ẩn giấu khỏi người dùng, giúp việc lập trình đơn giản hơn mà vẫn đảm bảo hiệu suất.

### Event Loop

Trái tim của NodeJS là “event loop”. Đây là vòng lặp sự kiện mà ở đó NodeJS tiếp tục lắng nghe sự kiện và thực hiện các hàm gọi lại khi một sự kiện được kích hoạt. Vòng lặp sự kiện cho phép NodeJS xử lý hàng nghìn kết nối đồng thời mà không cần phải tạo ra chi phí quản lý luồng.

### Trigger Callback

Khi thao tác I/O hoàn tất, hệ điều hành thông báo cho NodeJS, và NodeJS sau đó thực thi hàm callback tương ứng để xử lý kết quả hoặc tiếp tục xử lý logic.

### NPM (Node Package Manager)

NPM là hệ thống quản lý gói cho NodeJS, cho phép các nhà phát triển dễ dàng chia sẻ và sử dụng mã nguồn từ nhau. NPM là một trong những kho lưu trữ mã nguồn mở lớn nhất thế giới và chứa hàng ngàn module có thể được tích hợp vào ứng dụng của bạn.

Tổng hợp lại, Node.js mang đến một mô hình hiệu quả và mạnh mẽ cho các ứng dụng web và máy chủ, nhờ khả năng xử lý đồng thời nhiều hoạt động I/O mà không bị chặn, và qua đó tối ưu hóa việc sử dụng tài nguyên và cải thiện hiệu suất.

### **2.3.6 NoSQL**

Cơ sở dữ liệu NoSQL chuyên dành cho các mô hình dữ liệu cụ thể và lưu trữ dữ liệu trong các sơ đồ linh hoạt dễ dàng điều chỉnh quy mô cho các ứng dụng hiện đại [5]. Cơ sở dữ liệu NoSQL được công nhận rộng rãi vì khả năng dễ phát triển, chức năng cũng như hiệu năng ở quy mô lớn. Trang này có các tài nguyên giúp bạn hiểu rõ hơn về cơ sở dữ liệu NoSQL và bắt đầu sử dụng.

#### **Ưu điểm:**

Các ứng dụng hiện đại phải đối mặt với một số thách thức có thể được giải quyết bằng cơ sở dữ liệu NoSQL. Ví dụ, các ứng dụng xử lý một khối lượng dữ liệu lớn từ các nguồn khác nhau như phương tiện truyền thông xã hội, cảm biến thông minh và cơ sở dữ liệu của bên thứ ba. Tất cả dữ liệu khác nhau này không phù hợp với mô hình quan hệ. Việc thực thi các cấu trúc dạng bảng có thể dẫn đến dư thừa, sao chép dữ liệu và các vấn đề về hiệu suất trên quy mô lớn.

Cơ sở dữ liệu NoSQL chuyên dành cho các mô hình dữ liệu phi quan hệ và có sơ đồ linh hoạt để xây dựng các ứng dụng hiện đại. Chúng được công nhận rộng rãi vì khả năng dễ phát triển, chức năng cũng như hiệu năng ở quy mô lớn. Lợi ích của cơ sở dữ liệu NoSQL được liệt kê dưới đây.

#### **Tính linh hoạt:**

Cơ sở dữ liệu NoSQL thường cung cấp các sơ đồ linh hoạt giúp công đoạn phát triển nhanh hơn và có khả năng lặp lại cao hơn. Mô hình dữ liệu linh hoạt biến cơ sở dữ liệu NoSQL thành lựa chọn lý tưởng cho dữ liệu không được tổ chức thành cấu trúc hoặc có cấu trúc chưa hoàn chỉnh.

**Khả năng mở rộng:**

Cơ sở dữ liệu NoSQL thường được thiết kế để tăng quy mô bằng cách sử dụng các cụm phần cứng được phân phối thay vì tăng quy mô bằng cách bổ sung máy chủ mạnh và tốn kém. Một số nhà cung cấp dịch vụ đám mây xử lý các hoạt động này một cách không công khai dưới dạng dịch vụ được quản lý đầy đủ.

**Hiệu năng cao:**

Cơ sở dữ liệu NoSQL được tối ưu hóa cho các mô hình dữ liệu cụ thể và các mẫu truy cập. Chúng cho phép hiệu suất cao hơn so với khi bạn đang cố gắng thực hiện chức năng tương tự với cơ sở dữ liệu quan hệ.

### 2.3.7 MongoDB



Hình 2. 6 MongoDB

MongoDB lần đầu ra đời bởi MongoDB Inc., tại thời điểm đó là thế hệ 10, vào tháng Mười năm 2007, nó là một phần của sản phẩm PaaS (Platform as a Service) tương tự như Windows Azure và Google App Engine. Sau đó nó đã được chuyển thành nguồn mở từ năm 2009 [6].

MongoDB là một hệ quản trị cơ sở dữ liệu mã nguồn mở, là CSDL thuộc NoSql và được hàng triệu người sử dụng.

MongoDB là cơ sở dữ liệu NoSQL, lưu trữ dữ liệu linh hoạt theo dạng JSON. Với nhu cầu lưu trữ các mẫu thiệp cưới đa dạng, MongoDB là lựa chọn phù hợp giúp quản lý và truy vấn dữ liệu nhanh chóng, dễ mở rộng khi hệ thống phát triển.

Với CSDL quan hệ chúng ta có khái niệm bảng, các cơ sở dữ liệu quan hệ (như MySQL hay SQL Server...) sử dụng các bảng để lưu dữ liệu thì với MongoDB chúng ta sẽ dùng khái niệm là collection thay vì bảng.

Các collection trong MongoDB được cấu trúc rất linh hoạt, cho phép các dữ liệu lưu trữ không cần tuân theo một cấu trúc nhất định.

Thông tin liên quan được lưu trữ cùng nhau để truy cập truy vấn nhanh thông qua ngôn ngữ truy vấn MongoDB.

- **Ưu điểm:**

Do MongoDB sử dụng lưu trữ dữ liệu dưới dạng Document JSON nên mỗi một collection sẽ có các kích cỡ và các document khác nhau, linh hoạt trong việc lưu trữ dữ liệu, nên bạn muốn gì thì cứ insert vào thoải mái.

Dữ liệu trong MongoDB không có sự ràng buộc lẫn nhau, không có join như trong RDBMS nên khi insert, xóa hay update nó không cần phải mất thời gian kiểm tra xem có thỏa mãn các ràng buộc dữ liệu như trong RDBMS.

MongoDB rất dễ mở rộng (Horizontal Scalability). Trong MongoDB có một khái niệm cluster là cụm các node chứa dữ liệu giao tiếp với nhau, khi muốn mở rộng hệ thống ta chỉ cần thêm một node vào cluster:

Trường dữ liệu “\_id” luôn được tự động đánh index (chỉ mục) để tốc độ truy vấn thông tin đạt hiệu suất cao nhất.

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

Hiệu năng cao: Tốc độ truy vấn (find, update, insert, delete) của MongoDB nhanh hơn hẳn so với các hệ quản trị cơ sở dữ liệu quan hệ (RDBMS). Với một lượng dữ liệu đủ lớn thì thử nghiệm cho thấy tốc độ insert của MongoDB có thể nhanh tới gấp 100 lần so với MySQL.

- **Nhược điểm:**

Một ưu điểm của MongoDB cũng chính là nhược điểm của nó. MongoDB không có các tính chất ràng buộc như trong RDBMS nên khi thao tác với mongoDB thì phải hết sức cẩn thận.

Tồn bộ nhớ do dữ liệu lưu dưới dạng key-value, các collection chỉ khác về value do đó key sẽ bị lặp lại. Không hỗ trợ join nên dễ bị dư thừa dữ liệu.

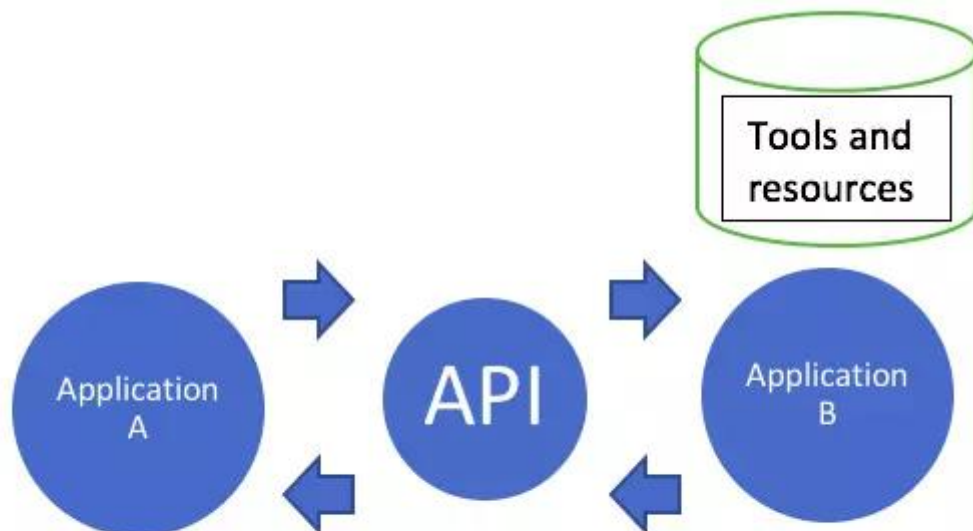
Khi insert/update/remove bản ghi, MongoDB sẽ chưa cập nhật ngay xuống ổ cứng, mà sau 60 giây MongoDB mới thực hiện ghi toàn bộ dữ liệu thay đổi từ RAM xuống ổ cứng điều này sẽ là nhược điểm vì sẽ có nguy cơ bị mất dữ liệu khi xảy ra các tình huống như mất điện...

### 2.3.8 API

Về mặt kỹ thuật, API là viết tắt của Giao diện lập trình ứng dụng (Application Programming Interface).

API là một trung gian phần mềm cho phép hai ứng dụng giao tiếp với nhau, có thể sử dụng cho web-based system, operating system, database system, computer hardware, or software library.

Ở dạng đơn giản nhất, API là giao diện cho phép một ứng dụng giao tiếp với ứng dụng khác thông qua các lệnh đơn giản và cách các lệnh này được gửi và định dạng mà dữ liệu được truy xuất thông qua API có thể khác với API SOAP hoặc REST [7].



## CHƯƠNG 3: HIỆN THỰC HÓA NGHIÊN CỨU

### 3.1 Mô tả bài toán

Bài toán của hệ thống khởi tạo thiệp cưới online là xây dựng một nền tảng trực tuyến cho phép người dùng thiết kế và tạo ra thiệp cưới tùy chỉnh. Hệ thống sẽ cung cấp các mẫu thiệp cưới có sẵn được phân loại theo các phong cách khác nhau như cổ điển, hiện đại, lãng mạn, tối giản,... . Người dùng có thể tùy chỉnh các thông tin cơ bản như tên cô dâu, chú rể, ngày cưới, địa điểm tổ chức tiệc cưới, lời mời, và các chi tiết khác.

Ngoài việc thay đổi văn bản, người dùng còn có thể thay đổi hình ảnh nền, thêm hình ảnh cá nhân và chỉnh sửa kích thước, vị trí của chúng trên thiệp. Hệ thống cũng sẽ cho phép người dùng đăng ký và đăng nhập tài khoản để lưu lại các thiết kế yêu thích và tiếp tục chỉnh sửa sau khi đăng nhập.

Hệ thống với các chức năng chính như sau:

Hiển thị các mẫu thiệp cưới có sẵn: Các mẫu thiệp sẽ được phân loại theo các chủ đề phong cách.

Tùy chỉnh văn bản: Cho phép người dùng thay đổi nội dung thiệp, bao gồm các thông tin về cô dâu, chú rể, ngày cưới, v.v.

Tùy chỉnh hình ảnh và nền: Người dùng có thể thay đổi hình nền, thêm hình ảnh cá nhân và chỉnh sửa ảnh.

Đăng ký/Đăng nhập người dùng: Hệ thống cần có chức năng đăng ký và đăng nhập để lưu trữ và truy xuất thiết kế thiệp.

### 3.2 Đặc tả bài toán

#### **Yêu cầu chức năng:**

Hiển thị danh mục thiệp cưới: Trang chủ của hệ thống sẽ hiển thị danh sách các mẫu thiệp cưới có sẵn, phân loại theo các phong cách như cổ điển, hiện đại, lãng mạn, tối giản.

Người dùng có thể duyệt qua các mẫu thiệp theo từng loại phong cách.

Mỗi mẫu thiệp sẽ có hình ảnh đại diện, mô tả ngắn gọn và giá tham khảo, người dùng có thể chọn mẫu thiệp để bắt đầu tùy chỉnh.



Tùy chỉnh văn bản trên thiệp: Người dùng có thể thay đổi thông tin văn bản trên thiệp cưới như tên cô dâu, chú rể, ngày cưới, địa điểm tổ chức tiệc cưới, lời mời.

Người dùng có thể chỉnh sửa các trường văn bản thông qua các ô input.

Cung cấp các tùy chọn như thay đổi phông chữ, kích thước chữ, màu sắc và kiểu chữ.

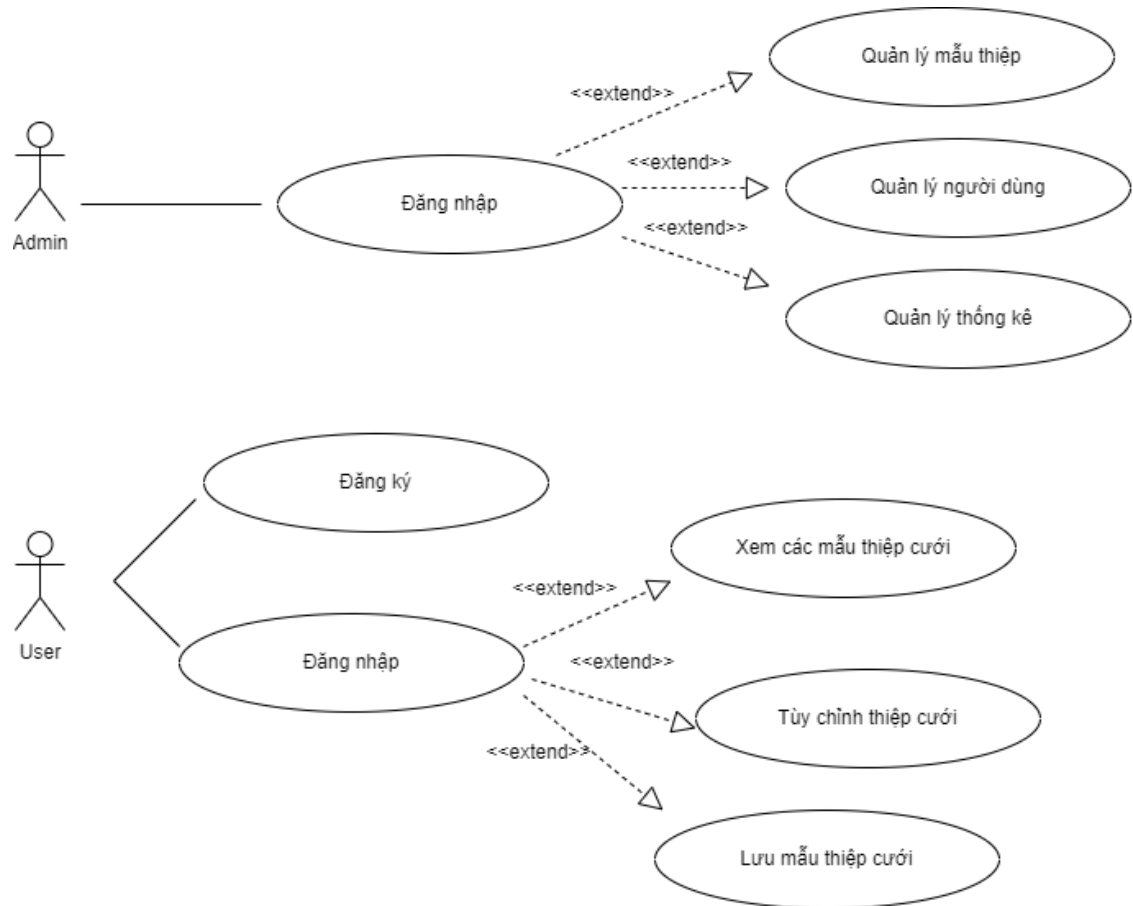
Tùy chỉnh hình ảnh và nền: Người dùng có thể thay đổi hình nền của thiệp, thêm hình ảnh cá nhân hoặc chọn hình ảnh từ thư viện có sẵn.

Đăng ký/Đăng nhập người dùng: Người dùng có thể đăng ký tài khoản hoặc đăng nhập để lưu lại các mẫu thiệp yêu thích và các thiết kế đang trong quá trình hoàn thiện.

Hệ thống yêu cầu người dùng đăng ký tài khoản bằng email và mật khẩu, người dùng có thể đăng nhập để truy cập các thiết kế thiệp đã lưu.

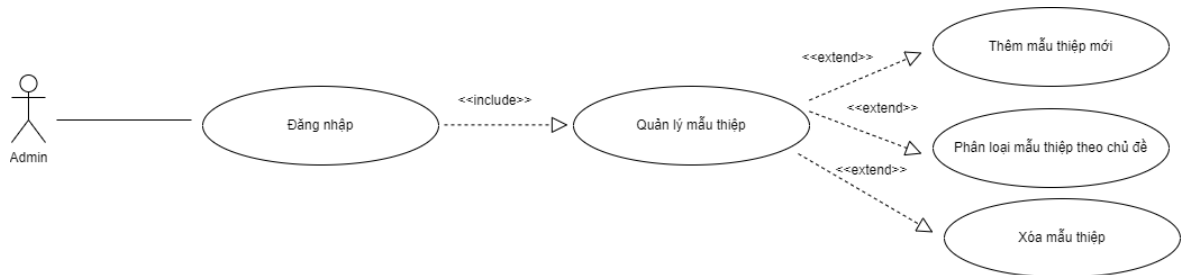
Lưu trữ các thiết kế thiệp cưới của người dùng trong tài khoản cá nhân.

### 3.3 Sơ đồ usecase tổng quát



Hình 3. 1 Biểu đồ use case tổng quát

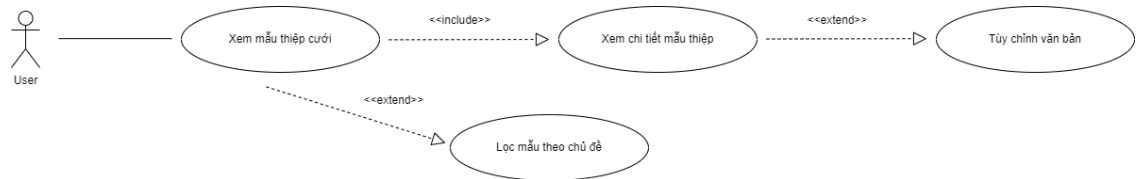
#### 3.3.1 Sơ đồ use case quản lý mẫu thiệp



Hình 3. 2 Biểu đồ use case quản lý thiệp cưới

Tác nhân quản trị khi thực hiện thao tác quản lý mẫu thiệp phải đăng nhập thì tác nhân này có quyền thêm xóa mẫu thiệp, phân loại mẫu thiệp theo chủ đề.

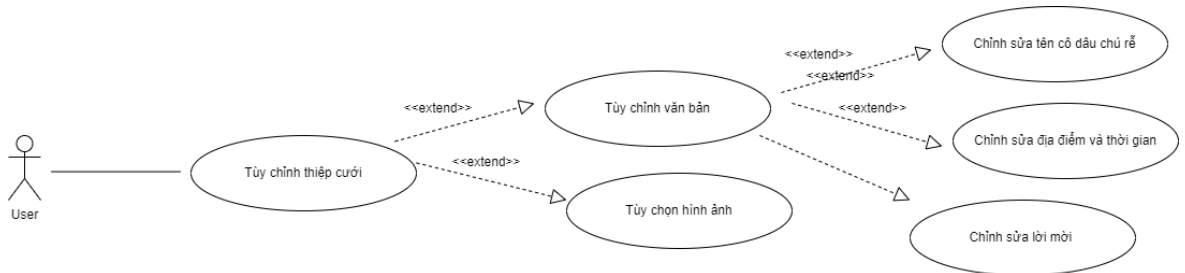
**Sơ đồ use case xem các mẫu thiệp cưới:**



Hình 3. 3 Biểu đồ use case xem các mẫu thiệp cưới

Khi tác nhân người dùng truy cập vào hệ thống có thể thực hiện thao tác: xem mẫu thiệp cưới, xem chi tiết mẫu thiệp và lọc theo chủ đề.

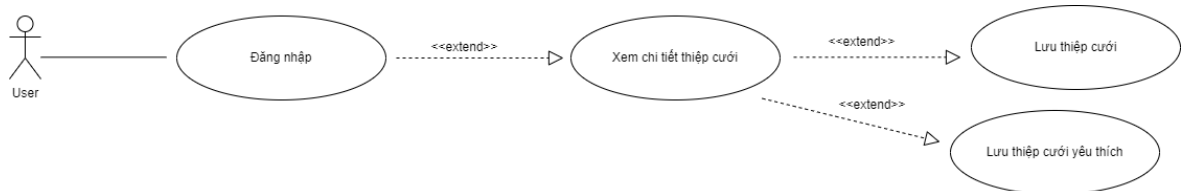
**Sơ đồ use case tùy chỉnh thiệp cưới:**



Hình 3. 4 Biểu đồ use case tùy chỉnh thiệp cưới

Khi tác nhân người dùng truy cập vào hệ thống có thể thực hiện thao tác: tùy chỉnh văn bản, chỉnh sửa tên cô dâu chú rể, chỉnh sửa ngày cưới, chỉnh sửa địa điểm và thời gian.

**Sơ đồ use case lưu thiệp cưới:**



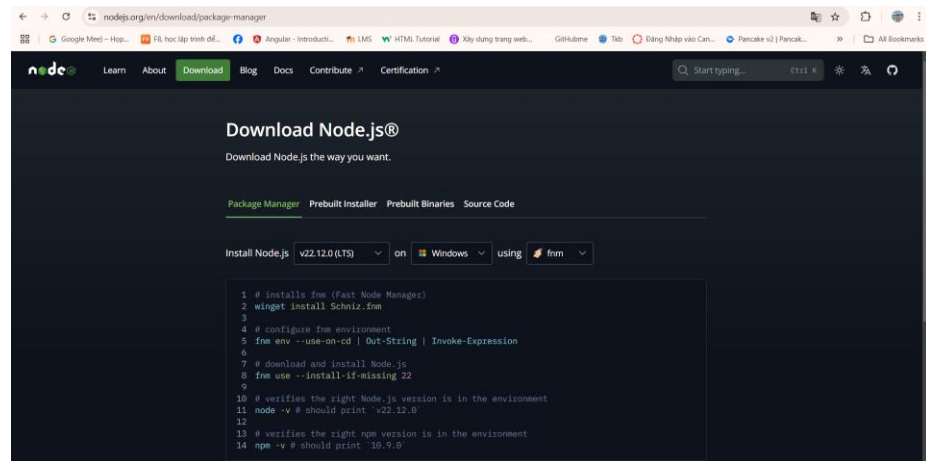
Hình 3. 5 Biểu đồ use case lưu thiệp cưới

Khi tác nhân người dùng truy cập vào hệ thống có thể xem chi tiết thiệp cưới và lưu thiệp cưới yêu thích.

### 3.4 Các bước để xây dựng hệ thống

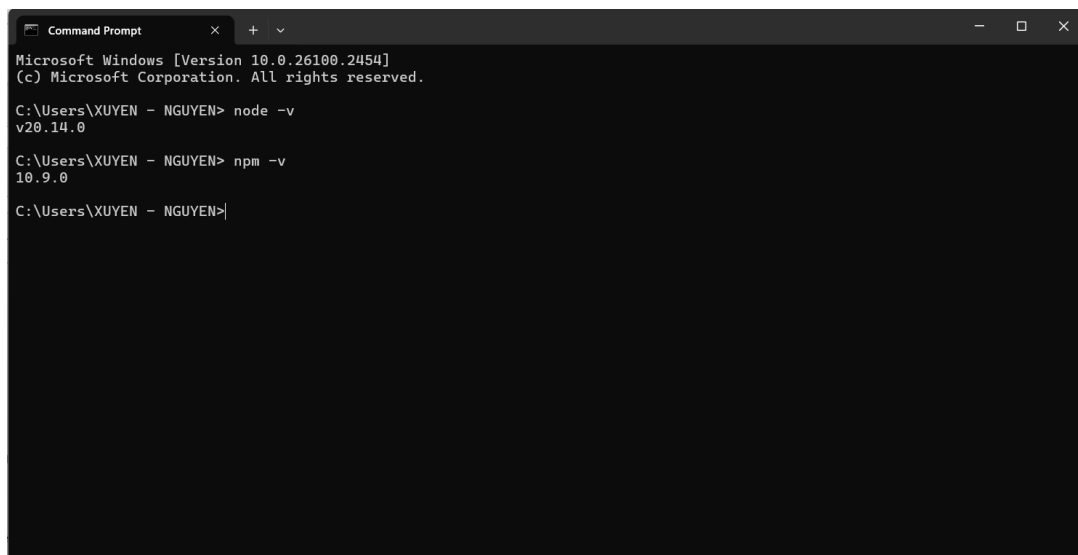
#### 3.4.1 Cài NodeJS

- Tải Node.js từ <https://nodejs.org/en> và cài đặt phiên bản LTS.



Hình 3. 4 Cài đặt NodeJS

- Kiểm tra việc cài đặt NodeJS đã thành công hay chưa bằng lệnh:



Hình 3. 5 Kiểm tra NodeJS và NPM bằng các dòng lệnh

### 3.4.2 Cài đặt Angular CLI

- Sử dụng lệnh sau để cài đặt Angular CLI

```
npm install -g @angular/cli
```

- Kiểm tra phiên bản Angular CLI

```
ng version|
```

## 3.5 Tạo project Angular

### 3.5.1 Tạo dự án Angular

1. Mở Terminal và tạo một ứng dụng Angular mới bằng dòng lệnh sau:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

XUYEN - NGUYEN@LAPTOP-BKA0QPBA MINGW64 ~
$ ng new wedding-card-project
```

2. Chọn các tùy chọn mặc định và tiếp tục

```
XUYEN - NGUYEN@LAPTOP-BKA0QPBA MINGW64 ~
$ ng new wedding-card-project
? Which stylesheet format would you like to use? (Use arrow keys)
> CSS [ https://developer.mozilla.org/docs/Web/CSS ]
  Sass (SCSS) [ https://sass-lang.com/documentation/syntax#scss ]
  Sass (Indented) [ https://sass-lang.com/documentation/syntax#the-indented-syntax ]
  Less [ http://lesscss.org ]

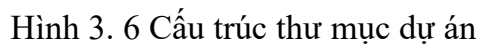
XUYEN - NGUYEN@LAPTOP-BKA0QPBA MINGW64 ~
$ ng new wedding-card-project
✓ Which stylesheet format would you like to use? CSS [ https://developer.mozilla.org/docs/Web/CSS ]
? Do you want to enable Server-Side Rendering (SSR) and Static Site Generation (SSG/Prerendering)? (y/n) n
```

3. Vào thư mục của dự án

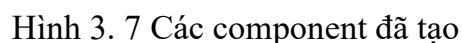
```
XUYEN - NGUYEN@LAPTOP-BKA0QPBA MINGW64 ~
$ cd wedding-card-project
● code .
```

Dự án sẽ có cấu trúc như sau:

---



- [illegible]



## 3.6 Cài đặt Backend với NodeJS và MongoDB

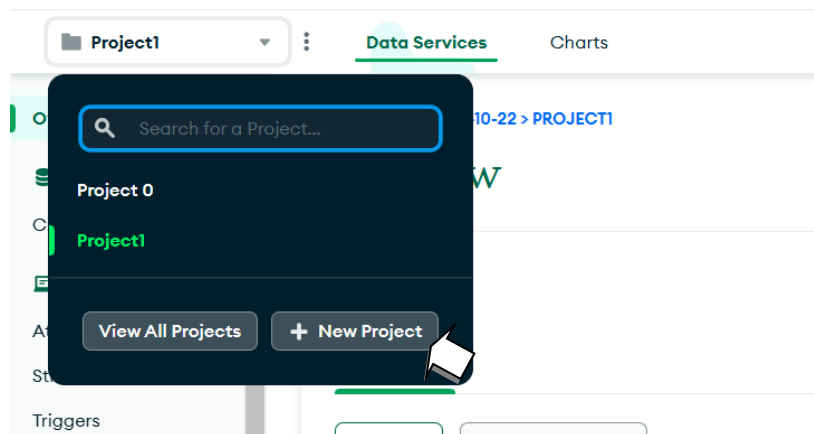
### 3.6.1 Cài đặt NodeJS và Express

Tạo thư mục backend trong dự án

```
XUYEN - NGUYEN@LAPTOP-BKA0QPBA MINGW64 ~/wedding-card-project (master)
$ mkdir backend
$ cd backend
$ npm init -y
$ npm install express mongoose body-parser cors
```

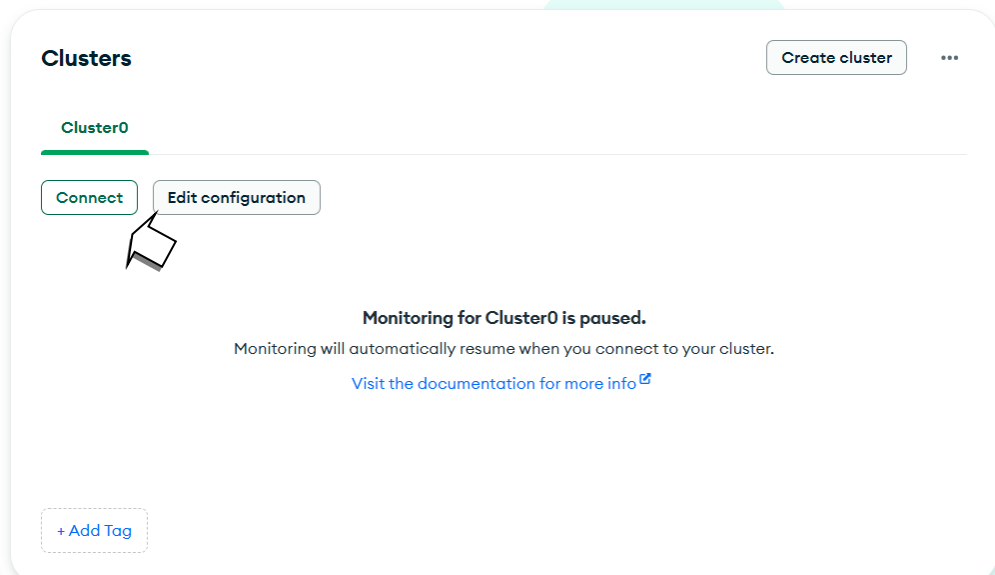
### 3.6.2 Kết nối MongoDB

- Đăng nhập tài khoản vào <https://cloud.mongodb.com/>
- Tiếp theo và đặt tên bấm chọn new project

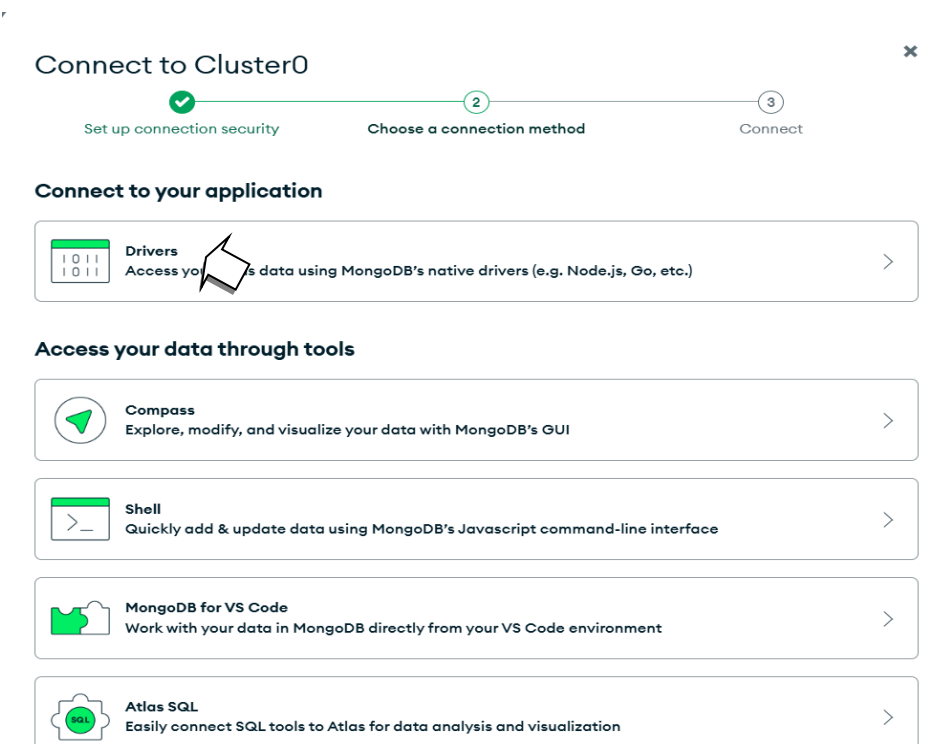


- Sau đó chọn connect

## Overview



### - Chọn Drivers



### - Chọn Driver dạng Node.js và copy dòng code kết nối vào file .env



Driver

Node.js

Version

6.7 or later

### 2. Install your driver

Run the following on the command line

```
npm install mongodb
```

[View MongoDB Node.js Driver installation instructions.](#)

### 3. Add your connection string into your application code

Use this connection string in your application

☐ View full code sample

```
mongodb+srv://lethinguyenxuyen2003:<db_password>@cluster0.wckag.mongodb.net/?retryWrites=true&w=majority&appName=Cluster0
```

Replace `<db_password>` with the password for the `lethinguyenxuyen2003` database user. Ensure any option params are [URL encoded](#).

RESOURCES

[Get started with the Node.js Driver](#)[Node.js Starter Sample App](#)[Access your Database Users](#)[Troubleshoot Connections](#)

Go Back

Done

```
.env  x  JS mongodb.js  <> product.component.html U  TS product.component.ts U  JS thiepcuoiModel.js  JS hi
wedding-invitation-backend > .env
1  MONGODB_URI = "mongodb+srv://lethinguyenxuyen2003:12292003@cluster0.wckag.mongodb.net"
```

- Tạo file mongodb.js trong thư mục backend

```
JS mongodb.js  x  <> product.component.html  TS product.component.ts  JS thiepcuoiModel.js  JS hi
wedding-invitation-backend > config > JS mongodb.js > [e] default
1  import mongoose from 'mongoose'
2
3  const connectDB = async () => {
4
5      mongoose.connection.on('connected', () => {
6          console.log("DB connected");
7      })
8  }
9
10  await mongoose.connect(`${process.env.MONGODB_URI}/thiepcuoi`)
11  }
12
13  export default connectDB
```

Gọi hàm connectDB() trong file server.js

```
wedding-invitation-backend > JS server.js > ...
1  import connectDB from "../config/mongodb.js";
2  import cors from 'cors'
3  import 'dotenv/config'
4  import express from 'express'
5  import userRouter from "../routes/userRoute.js";
6
7  //config
8  const port = process.env.PORT || 5000;
9  const app = express();
10 connectDB();
11
```

- Khởi động server NodeJS

```
○ $ node server.js
```

Tiếp theo thiết kế giao diện cho từng trang, cập nhật các component Angular với các giao diện người dùng tương tác.

### 3.6.3 Cơ sở dữ liệu

Cơ sở dữ liệu : WeddingCardSchema

```
// Định nghĩa Schema cho thiệp cưới
const WeddingCardSchema = new mongoose.Schema(
{
  id:{ type: String, required: true },
  name: { type: String, required: true }, // Tên thiệp cưới (e.g., "Thiệp Cổ Điển 1")
  category: { type: String, required: true }, // Danh mục (e.g., "Cổ điển", "Hiện đại")
  image: { type: String, required: true }, // Đường dẫn tới hình ảnh đại diện của thiệp
  image2: { type: String, required: true }, // Đường dẫn tới hình ảnh đại diện của thiệp
  mota: { type: String, required: true },
  gia: { type: Number, required: true },
  boxes: { type: [BoxSchema], default: [] }, // Danh sách các hộp văn bản
  createdAt: { type: Date, default: Date.now }, // Ngày tạo thiệp
  updatedAt: { type: Date, default: Date.now }, // Ngày cập nhật thiệp
},
);
```

- name, category, mota, gia: Lưu trữ tên, danh mục, mô tả và giá tiền của thiệp cưới.
- image, image2: Đường dẫn URL của các hình ảnh đại diện.

### Ứng dụng:

Schema này quản lý dữ liệu của thiệp cưới, bao gồm cả nội dung văn bản và hình ảnh.

Cơ sở dữ liệu : userSchema

```
const userSchema = new mongoose.Schema({
  username: { type: String, required: true, unique: true },
  email: { type: String, required: true, unique: true },
  password_hash: { type: String, required: true },
  favorites: [{ type: mongoose.Schema.Types.ObjectId, ref: 'WeddingCard' }],
  created_at: { type: Date, default: Date.now },
  updated_at: { type: Date, default: Date.now },
  customizedCards: [
    {
      cardId: { type: mongoose.Schema.Types.ObjectId, ref: 'WeddingCard' },
      boxes: [
        {
          name: String,
          position: { x: Number, y: Number },
          style: {
            font: String,
            fontSize: Number,
            color: String,
            bold: Boolean,
            italic: Boolean,
          },
        },
        {
          text: String,
        },
      ],
    },
  ],
}, { minimize: false } // Giữ lại các trường rỗng trong dữ liệu
);
```

### 3.7 Kiểm tra và chạy ứng dụng

- Chạy backend NodeJS với câu lệnh

```
$ node server.js
```

- Chạy thành công

```
XUYEN - NGUYEN@LAPTOP-BKA0QPBA MINGW64 /f/rua/wedding-invitation-backend node server.js
Server running on port : 5000
DB connected
```

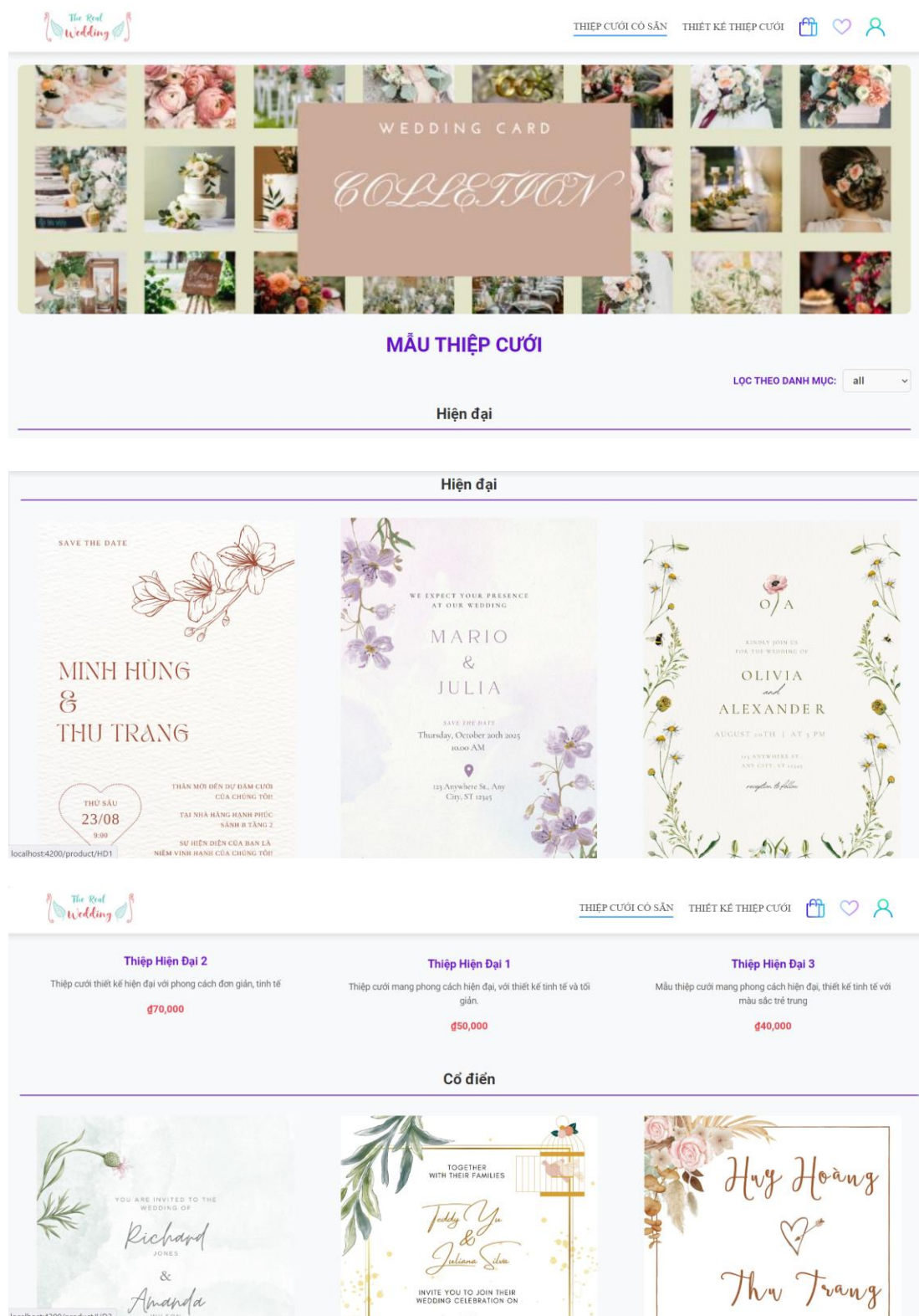
- Chạy frontend Angular:

```
$ ng serve
```

Sau đó truy cập vào <http://localhost:4200> để xem kết quả.

## CHƯƠNG 4: KẾT QUẢ NGHIÊN CỨU

### 4.1 Giao diện trang chủ

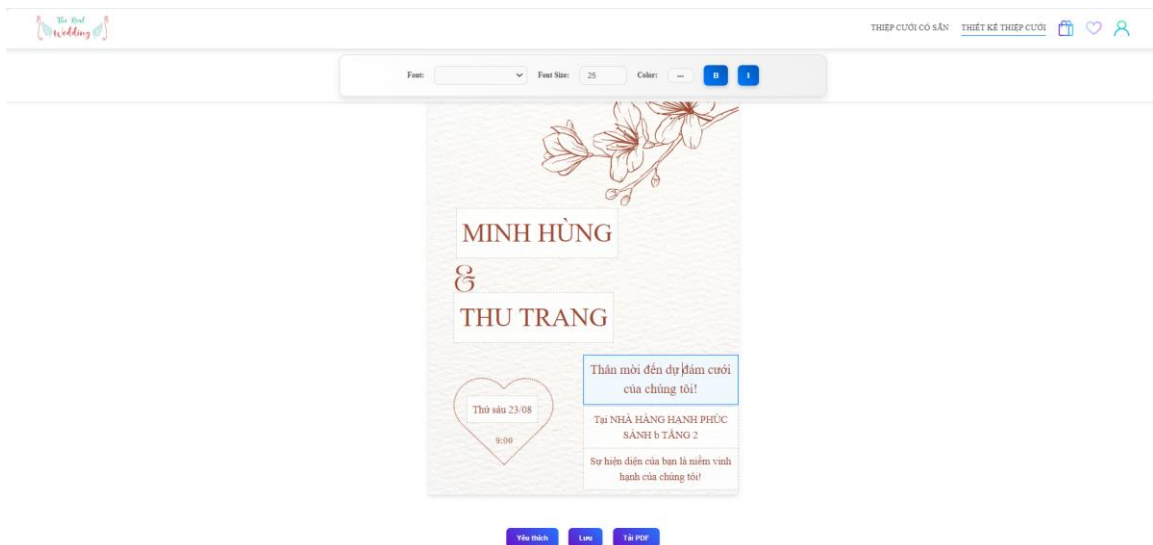


Hình 4. 1 Giao diện trang chủ

Trang chủ hệ thống thiệp cưới nổi bật với banner lớn, danh mục sản phẩm thiệp cưới đa dạng (Hiện đại, Cổ điển, Lãng mạn) có đầy đủ mô tả ngắn, giá tiền và

tên cụ thể tính năng lọc tiện lợi, người dùng có thể click vào từng thiệp cưới để chỉnh sửa.

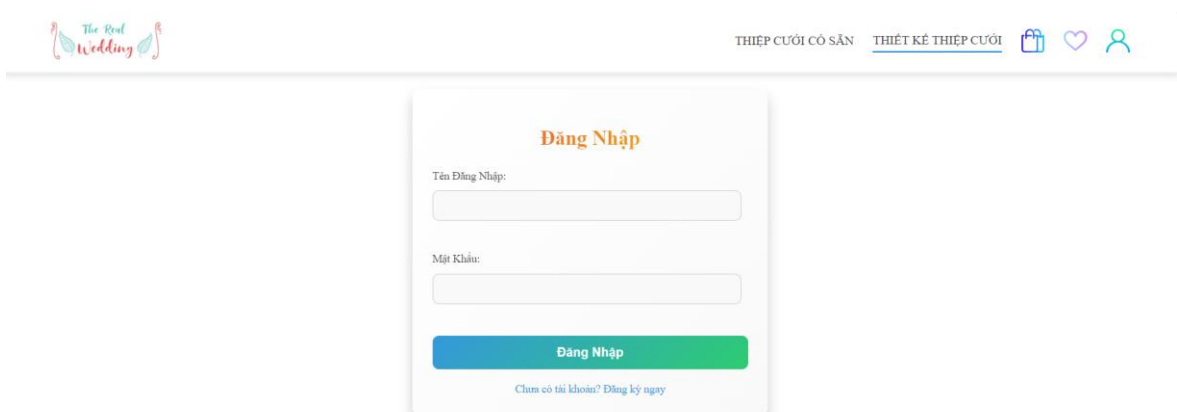
### 4.2 Giao diện chỉnh sửa thiệp cưới



Hình 4. 2 Giao diện chỉnh sửa thiệp cưới

Trang chỉnh sửa thiệp cưới cho phép người dùng tùy chỉnh văn bản thay đổi font chữ, màu sắc, định dạng, kích thước chữ. Người dùng có thể đăng nhập để lưu thiệp cưới đã chỉnh sửa, thêm vào danh sách yêu thích và tải thiệp về dưới dạng PDF

### 4.3 Giao diện đăng nhập/đăng ký



Hình 4. 3 Giao diện đăng nhập/ đăng ký

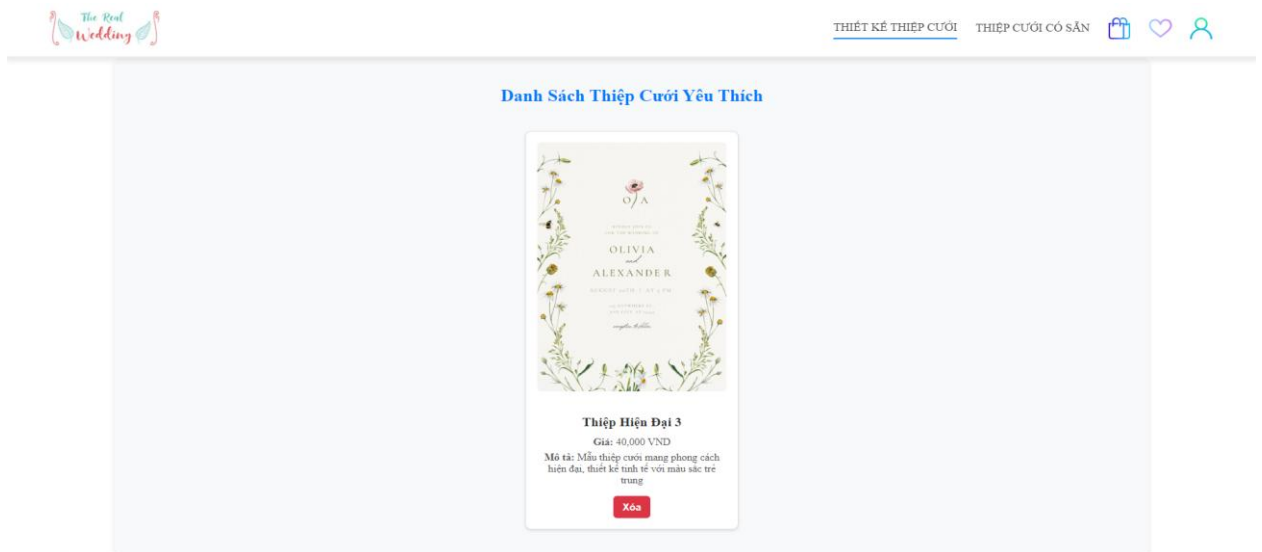
Người dùng có thể đăng ký để tạo tài khoản mới sau khi đăng ký người dùng có thể đăng nhập vào hệ thống để lưu lại các thiệp cưới đã chỉnh sửa, quản lý danh sách yêu thích.

#### 4.4 Giao diện lưu thiệp cưới đã chỉnh sửa

Hình 4. 4 Giao diện lưu thiệp cưới đã chỉnh sửa

Giao diện lưu thiệp cưới được thiết kế để hiển thị toàn bộ danh sách các thiệp cưới mà người dùng đã chỉnh sửa và lưu lại. Người dùng có thể xem lại các thiệp cưới này, kiểm tra chi tiết về nội dung thiết kế đã chỉnh sửa và tiếp tục chỉnh sửa nếu cần. Giao diện này cũng hỗ trợ tải thiệp về dưới dạng PDF để sử dụng ngay lập tức.

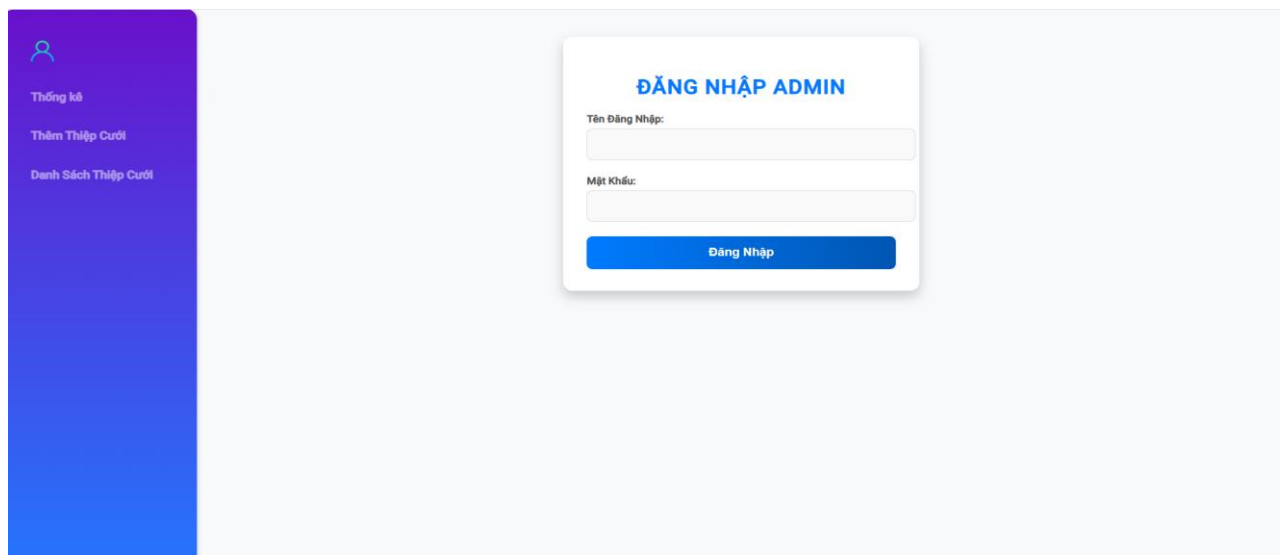
## 4.5 Giao diện lưu thiệp cưới yêu thích



Hình 4. 5 Giao diện lưu thiệp cưới yêu thích

Giao diện hiển thị danh sách các thiệp cưới mà người dùng đã thêm vào mục yêu thích, giúp họ dễ dàng xem lại những thiết kế ấn tượng. Tại đây, người dùng có thể quản lý danh sách yêu thích của mình bao gồm tùy chọn xóa các thiệp không còn phù hợp chỉ bằng một nút xóa.

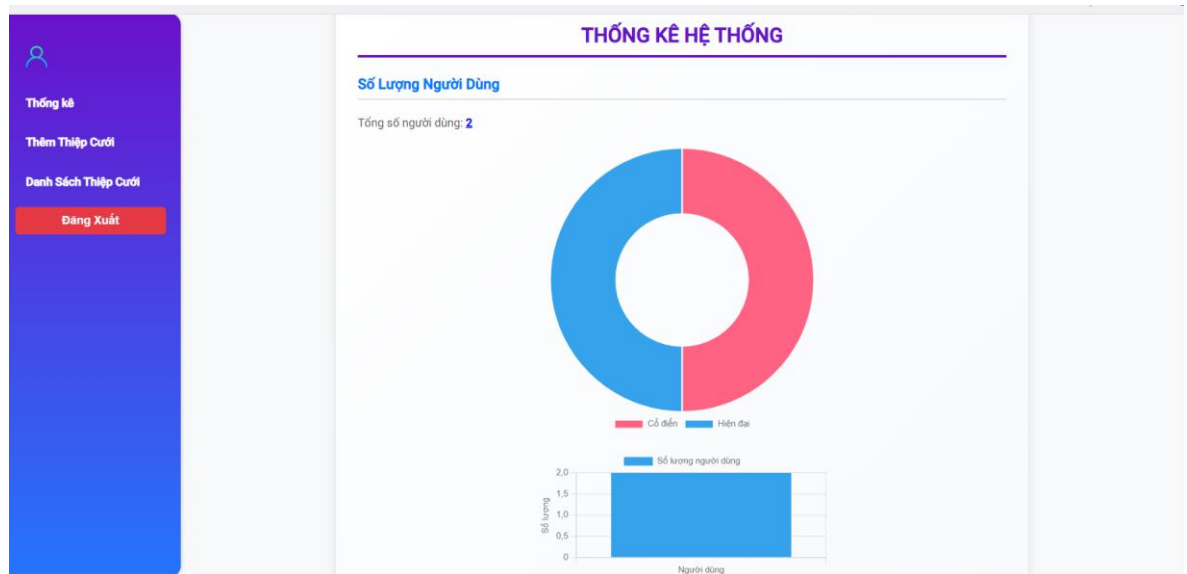
## 4.6 Giao diện đăng nhập Admin



Hình 4. 6 Giao diện đăng nhập admin

Trang dành riêng cho admin với giao diện đăng nhập bảo mật cho phép truy cập vào hệ thống quản lý. Admin có thể kiểm soát số lượng người dùng, theo dõi các thiệp cưới đã được thêm vào hệ thống và xem các báo cáo thống kê chi tiết giao diện được thiết kế để hỗ trợ quản lý hiệu quả và dễ dàng.

## 4.7 Giao diện thống kê



Hình 4. 7 Giao diện thống kê

Trang thống kê cung cấp thông tin chi tiết về số lượng thiệp cưới theo từng danh mục và số lượng người dùng truy cập hệ thống. Giao diện trực quan với các biểu đồ và số liệu dễ hiểu hỗ trợ admin theo dõi hoạt động hệ thống và đưa ra quyết định quản lý hiệu quả.

## 4.8 Giao diện thêm thiệp cưới

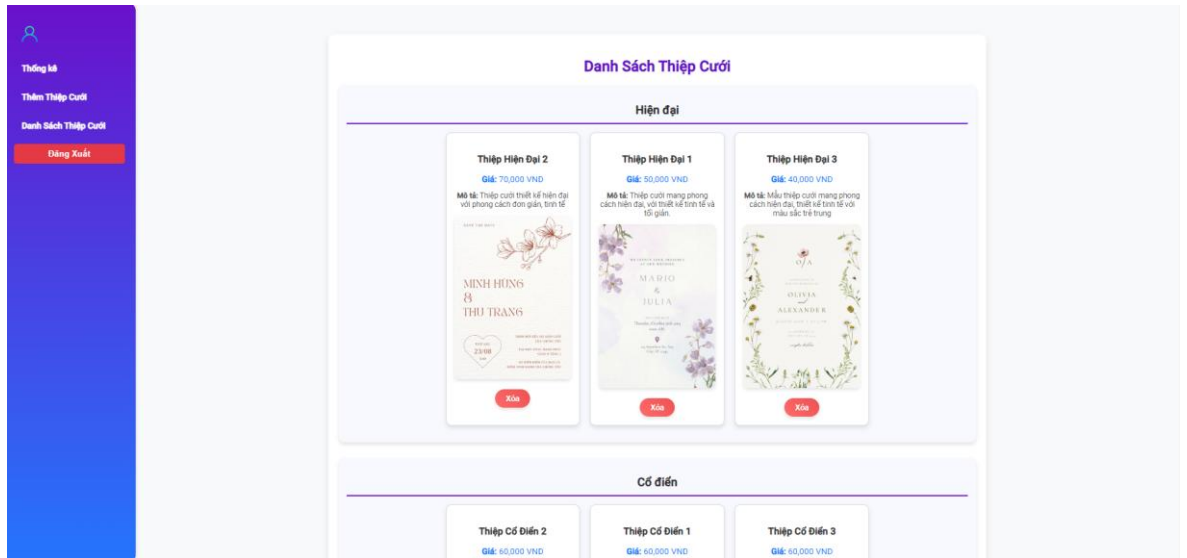
The screenshot displays the 'Thêm Thiệp Cưới' (Add Wedding Card) form. The form includes the following fields and sections: 'ID Thiệp:', 'Tên Thiệp:', 'Danh mục:' (a dropdown menu), 'Hình Ảnh 1:' (with a 'Chọn tệp' button and a note 'Chưa có tệp nào được chọn'), 'Hình Ảnh 2:' (with a 'Chọn tệp' button and a note 'Chưa có tệp nào được chọn'), 'Mô Tả:', 'Giá:', and a section titled 'Hộp Văn Bản' (Text Box) containing 'Hộp 1' and 'Tên Bưu:'. At the bottom right of the form is a 'Lưu Dữ Liệu' (Save Data) button.

Hình 4. 8 Giao diện thêm thiệp cưới

Trang thêm thiệp cưới cho phép admin nhập và quản lý các thông tin chi tiết của thiệp cưới như ID, tên thiệp, hình ảnh minh họa, giá cả và các thông tin liên quan khác. Giao diện dễ sử dụng hỗ trợ admin cập nhật nhanh chóng danh mục sản phẩm.



## 4.9 Giao diện danh sách thiệp cưới



Hình 4. 9 Giao diện danh sách thiệp cưới

Giao diện danh sách thiệp cưới cung cấp cho admin khả năng quản lý các thiệp cưới đã được thêm vào hệ thống. Admin có thể xem thông tin chi tiết và xóa các thiệp cưới một cách dễ dàng và nhanh chóng.

## **CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN**

### **5.1 Kết luận**

- Hệ thống khởi tạo thiệp cưới online đã được xây dựng thành công, tích hợp đầy đủ các chức năng như chỉnh sửa, lưu, tải thiệp cưới dưới dạng PDF, và quản lý thiệp cưới.
- Giao diện người dùng được thiết kế trực quan, dễ sử dụng, mang lại trải nghiệm linh hoạt và tiện lợi khi tùy chỉnh thiệp cưới.
- Hệ thống quản lý dành riêng cho Admin hoạt động hiệu quả, hỗ trợ quản lý danh sách thiệp cưới, kiểm soát người dùng và cung cấp báo cáo thống kê chi tiết.

### **5.2 Hướng phát triển**

- Phát triển thêm chức năng chia sẻ thiệp cưới qua các mạng xã hội như Facebook, Instagram, hoặc gửi trực tiếp qua email để tăng tính tiện lợi cho người dùng.
- Tăng cường các biện pháp bảo mật dữ liệu để bảo vệ thông tin người dùng, đặc biệt là trong các giao dịch trực tuyến.
- Nghiên cứu và triển khai thêm các tính năng như đặt in thiệp cưới trực tuyến, giao hàng tận nơi để hoàn thiện chuỗi dịch vụ.
- Tối ưu hóa hiệu năng hệ thống, đặc biệt trên các thiết bị di động, nhằm cải thiện tốc độ xử lý và nâng cao trải nghiệm người dùng trong môi trường mạng yếu.

## DANH MỤC TÀI LIỆU THAM KHẢO

- [1] Angular, “Angular là gì?”: <https://vietnix.vn/angular-la-gi/> ngày truy cập 13/11/2024.
- [2] HTML, “HTML là gì?”: <https://topdev.vn/blog/html-la-gi/> ngày truy cập 13/11/2024.
- [3] CSS, “CSS là gì?”: <https://topdev.vn/blog/css-la-gi/> ngày truy cập 13/11/2024
- [4] NodeJS, “NodeJS là gì?”: <https://topdev.vn/blog/node-js-la-gi/> Ngày truy cập 13/11/2024
- [5] NoSQL, “NoSQL là gì?”: <https://aws.amazon.com/vi/nosql/> Ngày truy cập 13/11/2024
- [6] MongoDB, “MongoDB là gì?”: <https://topdev.vn/blog/mongodb-la-gi/> Ngày truy cập 13/11/2024
- [7] API, “API là gì?”: <https://viblo.asia/p/tim-hieu-kien-thuc-co-ban-ve-api-maGK7A4Mlj2> Ngày truy cập 04/01/2025