|  |  |
| --- | --- |
|  |  |

**GRADUATION PROJECT**

**NEWS WEBSITE FOR G37 GENERAL HOSPITAL**

Subject: **GRADUATION PROJECT**

Specialty: **INFORMATION TECHNOLOGY**

**Supervisor:** Bùi Mạnh Toàn

**Student’s name:**

1. Nguyễn Hoàng Duy Student’s ID: 2082000116 Class: 20DTHQB1

|  |  |
| --- | --- |
| Ảnh có chứa văn bản, Phông chữ, biểu tượng, Đồ họa  Mô tả được tạo tự động | MINISTRY OF EDUCATION & TRAINING  **HOCHIMINH CITY UNIVERSITY OF TECHNOLOGY** |

**GRADUATION PROJECT**

**NEWS WEBSITE FOR G37 GENERAL HOSPITAL**

Subject: **GRADUATION PROJECT**

Specialty: **INFORMATION TECHNOLOGY**

**Supervisor:** Bùi Mạnh Toàn

**Student’s name:**

1. Nguyễn Hoàng Duy Student’s ID: 2082000116 Class: 20DTHQB1

Ho Chi Minh City, 2024

# **Declaration of the report**

I’m hereby declare that this report has been entirely written by me and is based on accurate and comprehensive research, analysis, and discussion. All reference materials used in this thesis are listed in the reference list.

Furthermore, I’m declare that this report has not been plagiarized or copied from any other sources without proper citation. I’m confirm that the project is an independent research work of ours.

If any violation related to the truthfulness of this declaration is found, i will take full responsibility for the legal and criminal consequences that may arise.

Sincerely,

Nguyễn Hoàng Duy

**HUTECH Institute of International Education**

# **REGISTRATION FORM**

**GRADUATION PROJECT**

*Program: International Standard Program*

1. **Name of student/ group** Number of members in group: 1
2. Nguyễn Hoàng Duy ID: 2082000116… Class: 20DTHQB1

Subject : Graduation Project

Specialty : Information Technology

1. **Supervisor/Mentor**: Bùi Mạnh Toàn
2. **Registered topic**: News Website for G37 General Hospital

Students understood the thesis’s requirements and commit to implement the project in accordance with schedule and meet the deadline.

|  |  |
| --- | --- |
| **Supervisor (Lecturer)** | *Ho Chi Minh city, …..........................2024*  **Signature**  *(Student’s full name)* |

**HUTECH Institute of International Education**

# **PROGRESS TRACKING FORM FOR GRADUATION PROJECT**

*(Recorded by supervisor and transferred to the students to file with the project at the end)*

1. **Topic**: News Website for G37 General Hospital

1. **Supervisor/Mentor**: Bùi Mạnh Toàn
2. **Name of students/ group** Number of members in group:

(1) Nguyễn Hoàng Duy ID: 2082000116… Class: 20DTHQB1

Subject : Graduation Project

Specialty : Information Technology

| **Week** | **Date** | **Contents** | **Comments of Supervisor/Mentor**  *(Signature)* | |
| --- | --- | --- | --- | --- |
| 1 | |  | | --- | | 19/2/2024 | | Discuss, identify project topics and development directions with instructors. |  | |
| 2 | 26/2/2024 | Draw use case diagram. |  | |
| 3 | |  | | --- | | 4/3/2024 | | Draw activity diagram and sequence diagram. |  | |
| 4 | 11/3/2024 | Start coding front end of the website. |  | |
| 5 | 18/3/2024 | Fix bug, improve the front end to look as good as possible, make the front end easy to use. Complete the front end. |  | |
| 6 | 25/3/2024 | Learn about PHP and connect front end with back end (using general database) |  | |
| 7 | 1/4/2024 | Coding CRUD for categories function |  | |
| Checking date: | | Progressive Assessment: …………..%  On-going: 🞎 Stop: 🞎 | | |
| 8 | 8/4/2024 | Coding CRUD for user function. |  | |
| 9 | 15/4/2024 | Coding CRUD for news function. |  | |
| 10 | 22/4/2024 | Improve dashboard part (admin approve news before putting on website, author can see the own news). |  | |
| 11 | |  | | --- | | 29/4/2024 | | Improve website interface. |  | |
| 12 | |  | | --- | | 6/5/2024 | | Coding like and comment section. |  | |
| 13 | |  | | --- | | 13/5/2024 | | Coding sort news by view and like function. |  | |
| 14 | |  | | --- | | 20/5/2023 | | Check everything of the website to start writing report, add news to website to look more like real-world project. |  |
| 15 | 27/5/2023 | Write and edit report with mentor support. |  |
| 16 | 3/6/2024 | Complete all project and report then give to mentor. |  |

|  |  |
| --- | --- |
| **Supportive Supervisor/Mentor** (if any)  *(Instructor’s full name)* | *Ho Chi Minh city, …..........................2024*  **Supervisor/Mentor**  *(Supervisor/Mentor’s full name)* |

**Table of Contents**

[**Declaration of the report** III](#_Toc168082931)

[**REGISTRATION FORM** IV](#_Toc168082932)

[**PROGRESS TRACKING FORM FOR GRADUATION PROJECT** V](#_Toc168082933)

[**Table of Contents** VIII](#_Toc168082934)

[**List of drawings, graphs** X](#_Toc168082935)

[**Introduction** XI](#_Toc168082936)

[**CHAPTER 1: OVERVIEW** 1](#_Toc168082937)

[**1.1.** **The goal of this project** 2](#_Toc168082938)

[**1.2.** **The structure of the project** 3](#_Toc168082939)

[**CHAPTER 2: THEORETICAL BASIS** 4](#_Toc168082940)

[**2.1.** **Overview of the website-making process** 4](#_Toc168082941)

[**2.2.** **Diagrams** 5](#_Toc168082942)

[**2.2.1.** **ER diagrams** 6](#_Toc168082943)

[**2.2.2.** **Use Case diagrams** 7](#_Toc168082944)

[**2.2.3.** **Activity Diagrams** 8](#_Toc168082945)

[**2.2.4.** **Sequence Diagram** 9](#_Toc168082946)

[**2.3.** **Front-end part** 11](#_Toc168082947)

[**2.3.1.** **Developed a front-end website** 11](#_Toc168082948)

[**2.3.2.** **Main languages to coding front-end** 11](#_Toc168082949)

[**2.4.** **Back-end technical** 15](#_Toc168082950)

[**2.4.1.** **PHP** 15](#_Toc168082951)

[**2.4.2.** **Laragon** 16](#_Toc168082952)

[**2.5.** **Connect Front-end and Back-end** 18](#_Toc168082953)

[**2.6.** **Conclusion** 19](#_Toc168082954)

[**CHAPTER 3: EXPERIMENTAL RESULTS** 20](#_Toc168082955)

[**3.1.** **Diagrams** 20](#_Toc168082956)

[**3.1.1.** **ER Diagrams** 20](#_Toc168082957)

[**3.1.2.** **Use Case Diagrams** 24](#_Toc168082958)

[**3.2.** **User interface and admin dashboard of the website** 36](#_Toc168082959)

[**3.2.1.** **User interface** 36](#_Toc168082960)

[**3.3.** **Admin Dashboard** 42](#_Toc168082961)

[**3.4.** **Website services** 51](#_Toc168082962)

[**CHAPTER 4: CONCLUSIONS AND FUTURE WORKS** 53](#_Toc168082963)

[**4.1.** **Conclusions** 53](#_Toc168082964)

[**4.2.** **Future works** 53](#_Toc168082965)

[**REFERENCES** 54](#_Toc168082966)

# **List of drawings, graphs**

[**Figure** 3.1: ER diagram of News Website 20](#_Toc168141505)

[**Figure** 3.2: Cut ER diagram of Dental Clinic Website 1 21](#_Toc168141506)

[**Figure** 3.3: Cut ER diagram of Dental Clinic Website 2 22](#_Toc168141507)

[**Figure** 3.6: Bills Management UC for Admin 25](#_Toc168141508)

[**Figure** 3.7: Health record Management UC for Admin 26](#_Toc168141509)

[**Figure** 3.8: Appointment Management UC for Admin 27](#_Toc168141510)

[**Figure** 3.9: Salary Management UC for Admin 28](#_Toc168141511)

[**Figure** 3.10: User Management UC for Admin 29](#_Toc168141512)

[**Figure** 3.11: Allot Management UC for Admin 30](#_Toc168141513)

[**Figure** 3.12: Medicine Management UC for Admin 31](#_Toc168141514)

[**Figure** 3.13: Prescription Management UC for Admin 32](#_Toc168141515)

[**Figure** 3.14: Management System UC for Admin 33](#_Toc168141516)

[**Figure** 3.15: Doctor UC 34](#_Toc168141517)

[**Figure** 3.16: Homepage of the Website 1 35](#_Toc168141518)

[**Figure** 3.17: Appointment Arrange 36](#_Toc168141519)

[**Figure** 3.18: Homepage of the Website 2 37](#_Toc168141520)

[**Figure** 3.19: Dentist's details 38](#_Toc168141521)

[**Figure** 3.20: Login page 39](#_Toc168141522)

[**Figure** 3.21: Register page 40](#_Toc168141523)

[**Figure** 3.22: Server Homepage 41](#_Toc168141524)

[**Figure** 3.23: User information page 41](#_Toc168141525)

[**Figure** 3.24: User Create page 42](#_Toc168141526)

[**Figure** 3.25: Update User Page 43](#_Toc168141527)

[**Figure** 3.26: Appointment Information Page 43](#_Toc168141528)

[**Figure** 3.27: Create Appointment Page 44](#_Toc168141529)

[**Figure** 3.28: Appointment Details Page 45](#_Toc168141530)

[**Figure** 3.29: Decentralization Page 46](#_Toc168141531)

[**Figure** 3.30: Service Create Page 47](#_Toc168141532)

[**Figure** 3.31: Service Update Page 48](#_Toc168141533)

[**Figure** 3.32: Shift Information Page 49](#_Toc168141534)

[**Figure** 3.33: Shift Create Page 49](#_Toc168141535)

[Figure 3.34: Website services 50](#_Toc168141536)

[**Figure** 3.35: Website services detail 51](#_Toc168141537)

# **Introduction**

In the current era of Industry 4.0, information technology is undoubtedly one of the leading sectors in development. Everything that humans see, use, and beyond bears the mark and influence of information technology. Acknowledging this fact, HUTECH University and I, personally, have aspired to immerse ourselves in this field to contribute to the development of information technology in Vietnam. Thus, this project was born.

During my academic studies and efforts to enhance my knowledge at the university, as well as while working on this project, I realized that my knowledge was quite limited compared to the scope of large-scale projects in the real world. Consequently, I had to devote a significant amount of time and effort to relearn everything.

However, this process did not go entirely as planned; there were numerous challenges and difficulties, as well as "bugs". Therefore, I would like to extend my deepest and most sincere gratitude to Mr. Bùi Mạnh Toàn for his wholehearted assistance throughout the lengthy duration of this project. Despite the difficulties, he never hesitated to meet with me, patiently correcting my errors and helping me complete this project in the most thorough manner possible. Once again, thank you very much, sir.

# **CHAPTER 1: OVERVIEW**

The project I chose is making News Website for G37 General Hospital, a website that allows people to access the latest, hottest, and fastest information about hospitals, healthcare, or the achievements that Vietnam has made in this field.

From the perspective of website staff such as authors or admins, they can post news quickly and conveniently using the provided tools. They can also edit or delete news if there are any errors, but such cases will be rare because the website has an approval function that allows admins to thoroughly review the news before they are officially published. Additionally, various topics can be added to make the website more diverse. In conclusion, they can use all the functions that other websites have, with some minor upgrades.

Before I started this project, I’m already had some knowledge about making a website from Website Design course at HUTECH University. I’m have good skills in HTML, CSS and JavaScript to make our website look good enough compared to other professional websites on the internet. I’m also have knowledge about the backend skill called PHP, a popular technique uses for the backend in websites out there.

To pass the web programming course, I need to create a beautiful and professional website. Of course, I thought I could easily achieve this with a high score and that the graduation project wouldn't be challenging for me. However, when I started working on it, I realized how difficult it was. I had to sit down with my team members to share knowledge and help each other to get through the graduation project as smoothly as possible. Therefore, this might not be an ideal news website, but it will have all the basic functions of a real website out there.

## **The goal of this project**

Before talking about my news website, I need to give an overview of this project. This news page is just a part of a multidisciplinary website that my team and I are working on. The website focuses on the needs of people to check their health quickly and conveniently without much effort. My news page is no exception; it aims to provide users with the latest and most comprehensive information on domestic and international health situations to prevent potential health issues. This is one way that a health news website can help people "Prevent Disease."

Although there are numerous websites out there that provide news to users, such as Sức Khỏe Đời Sống, the website I used as a model for my own, or Tuổi Trẻ, I find these websites quite cluttered with an overwhelming amount of information and textures that can confuse readers and make them look unorganized. Therefore, such websites tend to attract older readers rather than the younger generation, who have more modern ways of accessing news through platforms like Facebook or TikTok. However, these platforms rarely provide health news.

Speaking of young people, they currently only care about drama news or entertainment news and have little to no interest in health or political news. Therefore, health news websites or health television channels naturally do not attract much attention from them.

Aware of this issue, I aim to create a website that, while it may not look highly professional, can still provide comprehensive health content. It will feature an interface that is not only visually appealing and aligned with the tastes of the younger generation but also tidy and easy to use for older adults, who are often considered less tech-savvy.

## **The structure of the project**

In this project, the structure is very simple. First is the preliminary page with page cover, it’s about the name of the project, the name of the advisor, the student’s name, id and class, the declaration page, and finally is the index.

Next is the main text with four main chapters. Chapter 1 is the overview, introduces shortly about the project, and summary the theory. Structure of the report, Present the structure of the project consisting of chapters and summarize each chapter, and the missions of the project, State the urgency and reasons for the formation of the project, scientific and practical significance, research objectives, objects, limited scope.

Chapter 2 is Theoretical Basis, in this chapter I will present the concepts and methods of the problem solving by the team including the description of technologies, systems, constraints or new solutions, mathematical models, model-building explanation.

Chapter 3 is Reporting Result, this chapter is simply showing the results of the project. For the application topics, if the result is the software product must have design documents, settings, interfaces, follow one of the learned models like UML.

And finally, is the chapter 4, Conclusion. List general conclusions, affirm the achieved results, new contributions, proposals, and recommendations (if any).

# **CHAPTER 2: THEORETICAL BASIS**

1. **Overview of the website-making process**

First, let’s talk about what a website is, a website is a collection of publicly accessible, interlinked Web pages that share a single domain name. Websites can be created and maintained by an individual, group, business or organization to serve a variety of purposes [1].

To make, create, and design a website, it’s primarily coded in Hypertext Markup Language (HTML); Cascading Style Sheets (CSS) are used to control appearance beyond basic HTML. It’s taken tons of code in HTML and CSS language to design and make the web look more professional and cooler in the eyes of who visit the website. Besides that, there are other languages that help HTML and CSS to make the website look more beautiful and more functional, optimize the speed to connect to the website or to use the function of the website like JavaScript, ReactJS, and Bootstrap. Those coding languages will be talked about more later in chapter 2.

A website often has two sides of it called Front-end and Back-end, so what is the different of those two. Although they are very different, the frontend and backend are come together to create a functional website.

Frontend development focuses on the graphical aspects of a website or app, AKA the client or user-side. Backend development concentrates on the server-side. Together, they develop a dynamic website that lets users make purchases, use contact forms, and engage in interactive activities. Examples of dynamic websites are Facebook, Netflix, and PayPal.

Frontend and backend also use different programming languages. Frontend is developed with CSS, HTML, and JavaScript, while backend is developed with Python, Ruby, PHP, C++, Node.js, and also JavaScript.

Backend and frontend developers also use different frameworks. Frontend developers use frameworks and libraries like AngularJS, React.js, jQuery, and Sass. Meanwhile, backend developers work with Express, Django, Rails, Spring, and Laravel.

In the first place, front-end and back-end of the website is not connected to each other because both use different kinds of coding languages, so that to connect front and back-end to each other require other techniques, most popular techniques is Templating and API. Template engines are used when you want to rapidly build web applications that are split into different components. Templates also enable fast rendering of the server-side data that needs to be passed to the application. [2]. This approach makes it easier to design an HTML page. And with API, API is the acronym for application programming interface — a software intermediary that allows two applications to talk to each other. APIs are an accessible way to extract and share data within and across organizations. APIs are all around us. Every time you use a rideshare app, send a mobile payment, or change the thermostat temperature from your phone, you’re using an API. [3].

Finally, if a website being made, then it must be uploaded on the internet for people to use. And to do that required a lot of work, not just simply put it on the internet without anything trouble. And the way to do it will be discussed later in this chapter.

In conclusion, to make a website is not a piece of cake work, it’s requiring lots of work, sweat, pressure and hours to make a complete and good-looking website. That why it’s always come with a team to create a website to share the hard work.

1. **Diagrams**

To create a website, a group of coders not just create it from scratch, out of nowhere with no destination, they have to draw diagram. Diagrams just like plan of the project, what coder should put in this website, what is a user can do, what is a staff can do and what is an administrator can do, how many functions this project can have, give this detail to this place and that detail to that place. In general, to create something, website or application or anything else, everything needs a diagram, a plan and after all set, then coders will start to create the project.

This project is not an exception, to create this website I must follow this rule and create many diagrams to orientation what I will do with this website. I have ER diagrams, Use Case diagrams, Activity Diagrams and Sequence Diagrams are the main diagrams in this project to create my website.

Let’s talk about what is diagram first. According to Oxford Dictionary, a diagram is a simplified drawing showing the appearance, structure, or workings of something, a schematic representation. In this project, this definition true enough.

A lot of diagrams to use to plan process what I can do to complete my project, but like I said before, in this project just have four main diagrams to complete this project: ER Diagrams, Use Case Diagrams, Activity Diagrams and Sequence Diagrams.

1. **ER diagrams**

An entity relationship diagram (ERD), also known as an entity relationship model, is a graphical representation that depicts relationships among people, objects, places, concepts or events within an information technology (IT) system. An ERD uses [data modeling](https://www.techtarget.com/searchdatamanagement/definition/data-modeling) techniques that can help define business processes and serve as the foundation for a [relational database](https://www.techtarget.com/searchdatamanagement/definition/relational-database).[4].

Entity relationship diagrams provide a visual starting point for database design that can also be used to help determine information system requirements throughout an organization. After a [relational database](https://www.techtarget.com/searchdatamanagement/definition/relational-database) is rolled out, an ERD can still serve as a reference point, should any debugging or business process re-engineering be needed later.

However, while an ERD can be useful for organizing data that can be represented by a relational structure, it can't sufficiently represent semi-structured or unstructured data. It's also unlikely to be helpful on its own in integrating data into a pre-existing information system.[4]

A one to one (1:1) relationship is the relationship of one entity to only one other entity, and vice versa. It should be rare in any relational database design, because very little chance that have one-to-one relationships between things [5].

One-to-many relationships are used when a single entity is associated with any number of other entities. For example, a Blog can have many associated Posts, but each Post is associated with only one Blog [6].

Many-to-many relationships are not ideal. If left as it is in the above example, the data would be duplicated. For instance, if there’s a professor that teaches six subjects, you would have him or her listed in the table six times, every time for a different subject. This is quite inefficient. So, how would you resolve this many-to-many relationship between these two entities? By introducing a junction table into your model. It will resolve the many-to-many relationship into multiple one-to-many relationships [7].

1. **Use Case diagrams**

A UML use case diagram is the primary form of system/software requirements for a new software program underdeveloped. Use cases specify the expected behavior (what), and not the exact method of making it happen (how). A key concept of use case modeling is that it helps us design a system from the end user's perspective. It is an effective technique for communicating system behavior in the user's terms by specifying all externally visible system behavior [8].

A use case diagram is usually simple. It does not show the detail of the use cases: It only summarizes some of the relationships between use cases, actors, and systems. It does not show the order in which steps are performed to achieve the goals of each use case.

As said, a use case [8] diagram should be simple and contains only a few shapes. If what project contain more than 20 use cases, it is probably misusing use case diagram.

Use case diagrams are typically developed in the early stage of development and people often apply use case modeling for the following purposes:

- Represent the goals of systems and users.

- Specify the context a system should be viewed in.

- Specify system requirements.

- Provide a model for the flow of events when it comes to user interactions.

- Provide an outside view of a system [8].

1. **Activity Diagrams**

Activity Diagrams are used to illustrate the flow of control in a system and refer to the steps involved in the execution of a use case. We can depict both sequential processing and concurrent processing of activities using an activity diagram ie an activity diagram focuses on the condition of flow and the sequence in which it happens.

* We describe what causes a particular event using an activity diagram.
* An activity diagram portrays the control flow from a start point to a finish point showing the various decision paths that exist while the activity is being executed.
* They are used in business and process modeling where their primary use is to depict the dynamic aspects of a system. Sequence Diagrams [9]

Activity diagrams present a number of benefits to users. Consider creating an activity diagram to:

* Demonstrate the logic of an algorithm.
* Describe the steps performed in a UML use case.
* Illustrate a business process or workflow between users and the system.
* Simplify and improve any process by clarifying complicated use cases.
* Model software architecture elements, such as method, function, and operation. [10]

Before you begin making an activity diagram, you should first understand its makeup. Some of the most common components of an activity diagram include:

* Action: A step in the activity wherein the users or software perform a given task. In Lucidchart, actions are symbolized with round-edged rectangles.
* Decision node: A conditional branch in the flow that is represented by a diamond. It includes a single input and two or more outputs.
* Control flows: Another name for the connectors that show the flow between steps in the diagram.
* Start node: Symbolizes the beginning of the activity. The start node is represented by a black circle.
* End node: Represents the final step in the activity. The end node is represented by an outlined black circle. [10]

1. **Sequence Diagram**

UML Sequence Diagrams are interaction diagrams that detail how operations are carried out. They capture the interaction between objects in the context of a collaboration. Sequence Diagrams are time focus and they show the order of the interaction visually by using the vertical axis of the diagram to represent time what messages are sent and when. [11]

**Sequence Diagrams captures:**

the interaction that takes place in a collaboration that either realizes a use case or an operation (instance diagrams or generic diagrams)

high-level interactions between user of the system and the system, between the system and other systems, or between subsystems (sometimes known as system sequence diagrams). [11]

**Purpose of Sequence Diagram**

* Model high-level interaction between active objects in a system.
* Model the interaction between object instances within a collaboration that realizes a use case.
* Model the interaction between objects within a collaboration that realizes an operation.
* Either model generic interactions (showing all possible paths through the interaction) or specific instances of a interaction (showing just one path through the interaction). [11]

**Sequence Diagrams at a Glance**

Sequence Diagrams show elements as they interact over time and they are organized according to object (horizontally) and time (vertically):

**Object Dimension**

The horizontal axis shows the elements that are involved in the interaction. Conventionally, the objects involved in the operation are listed from left to right according to when they take part in the message sequence. However, the elements on the horizontal axis may appear in any order.

**Time Dimension**

The vertical axis represents time proceedings (or progressing) down the page. [11]

1. **Front-end part**

Let’s go deeper into what is called a Front-end of a website, and what are the requirement to make a good front-end to a website.

The front end is everything a user sees and interacts with when they click on a link or type in a web address. The web address is also known as at URL, or Uniform Resource Locator, and it tells what webpage should load and appear in your browser.

1. **Developed a front-end website**

Creating a front end experience isn’t a one-person job. In fact, it takes a group of people to create a website that looks as good as it feels to use.

Most often, this group will be comprised of a front end developer — whose job it is to actually write the code for the functions of the website — and a UX or UI developer, who will work on the visuals of the website.

The front end experience usually goes through a few different stages of development, including the creation of wireframes (rough outlines of the user flow), prototypes (working examples of the site), and finally user testing [12].

1. **Main languages to coding front-end**

There are 7-8 languages to coding front-end, but in this case just four main languages using in this project, it is Hypertext Markup Language (HTML), Cascading Style Sheets (CSS), JavaScript and Bootstrap.

First, we are going to talk about Hypertext Markup Language or HTML for short. HTML is a text-based approach to describing how content contained within an HTML file is structured. This markup tells a web browser how to display text, images and other forms of multimedia on a webpage.

HTML stands for Hypertext Markup Language. Itis a standard markup language for web page creation. It allows the creation and structure of sections, paragraphs, and links using HTML elements (the building blocks of a web page) such as tags and attributes.

HTML has a lot of use cases, namely:

* Web development. Developers use HTML code to design how a browser displays web page elements, such as text, hyperlinks, and media files.
* Internet navigation. Users can easily navigate and insert links between related pages and websites as HTML is heavily used to embed hyperlinks.
* Web documentation. HTML makes it possible to organize and format documents, similarly to Microsoft Word. [13].

Next Language make the website look cooler and more professional compared to HTML code only, and why this code file always call style. Cascading Style Sheets (CSS).

Cascading Style Sheets, fondly referred to as CSS, is a simple design language intended to simplify the process of making web pages presentable.

CSS [14] handles the look and feel part of a web page. Using CSS, you can control the color of the text, the style of fonts, the spacing between paragraphs, how columns are sized and laid out, what background images or colors are used, layout designs, variations in display for different devices and screen sizes as well as a variety of other effects.

CSS is easy to learn and understand but it provides powerful control over the presentation of an HTML document. Most commonly, CSS is combined with the markup languages HTML or XHTML. CSS is a language for specifying how documents are presented to users — how they are styled, laid out, etc. A document is usually a text file structured using a markup language — HTML is the most common markup language, but you may also come across other markup languages such as SVG or XML.

Presenting a document to a user means converting it into a form usable by your audience. Browsers, like Firefox, Chrome, or Edge, are designed to present documents visually, for example, on a computer screen, projector, or printer.

Next is JavaScript. JavaScript [15] is a scripting language for creating dynamic web page content. It creates elements for improving site visitors’ interaction with web pages, such as dropdown menus, animated graphics, and dynamic background colors.

Today, JavaScript has plenty of frameworks and libraries to simplify complex projects, such as AngularJS, jQuery, and ReactJS.

Originally run on the client-side, the JavaScript implementation has branched out to the server-side after the Node.js development ‒ a cross-platform server environment built on the Google Chrome JavaScript V8 engine.

While it caters to web-based programs the most, JavaScript programming features have other implementations in different areas. The following are several basic uses of JavaScript.

And last but not least, Bootstrap. Bootstrap [16] is a free, open-source front-end development framework for the creation of websites and web apps. Designed to enable responsive development of mobile-first websites, Bootstrap provides a collection of syntax for template designs.

As a framework, Bootstrap includes the basics for responsive web development, so developers only need to insert the code into a pre-defined grid system. The Bootstrap framework is built on Hypertext Markup Language (HTML), cascading style sheets (CSS) and JavaScript. Web developers using Bootstrap can build websites much faster without spending time worrying about basic commands and functions.

Bootstrap makes responsive web design a reality. It makes it possible for a web page or app to detect the visitor's screen size and orientation and automatically adapt the display accordingly. The mobile-first approach assumes smartphones, tablets and task-specific mobile apps are employees' primary tools for getting work done. Bootstrap addresses the requirements of those technologies in design and includes UI components, layouts, JavaScript tools and the implementation framework. The software is available precompiled or as source code.

Mark Otto and Jacob Thornton developed Bootstrap at Twitter to improve the consistency of tools used on the site and to reduce maintenance. The software was formerly known as Twitter Blueprint and is sometimes referred to as Twitter Bootstrap.

In conclusion, front-end main languages create, design, the interface of the website to look as good looking as possible to help the visitors of the website more comfortable when using the website.

And that all for front-end, now let’s get to back-end of the website. If front-end called the client-side or user-side, then back-end is what call the server-side.

1. **Back-end technical**

Back end is a part that users can’t see, only admin and who have authority can access because it’s the server of a website.

There are a lot of back-end coding languages and frameworks out there but in this project the back-end will be coding in PHP with a little bit support from Laragon

1. **PHP**

PHP [17] (recursive acronym for PHP: Hypertext Preprocessor) is a widely used open-source general-purpose scripting language that is especially suited for web development and can be embedded into HTML.

What distinguishes PHP from something like client-side JavaScript is that the code is executed on the server, generating HTML which is then sent to the client. The client would receive the results of running that script, but would not know what the underlying code was. You can even configure your web server to process all your HTML files with PHP, and then there's really no way that users can tell what you have up your sleeve.

The best part about using PHP is that it is extremely simple for a newcomer, but offers many advanced features for a professional programmer. Don't be afraid to read the long list of PHP's features. You can jump in, in a short time, and start writing simple scripts in a few hours.

PHP is mainly focused on server-side scripting, so you can do anything any other program can do, such as collect form data, generate dynamic page content, or send and receive cookies. But PHP can do much more. There are three main areas where PHP scripts are used:

- Server-side scripting. This is the most traditional and main target field for PHP. You need three things to make this work: the PHP parser, a web server, and a web browser.

- Command line scripting. You can make a PHP script to run it without any server or browser. You only need the PHP parser to use it this way.

- Writing desktop applications. PHP is probably not the very best language to create a desktop application with a graphical user interface, but if you know PHP very well, and would like to use some advanced PHP features in your client-side applications you can also use PHP-GTK to write such programs [18].

1. **Laragon**

Laragon is a portable, isolated, fast & powerful universal development environment for PHP, Node.js, Python, Java, Go, Ruby. It is fast, lightweight, easy-to-use and easy-to-extend.

Laragon is great for building and managing modern web applications. It is focused on performance - designed around stability, simplicity, flexibility and freedom.

Laragon is very lightweight and will stay as lean as possible. The core binary itself is less than 2MB and uses less than 4MB RAM when running.

Laragon doesn’t use Windows services. It has its own service orchestration which manages services asynchronously and non-blocking so you’ll find things run fast & smoothly with Laragon. [19]

**Features:**

* Pretty URLs: Use app.test instead of localhost/app.
* Portable: You can move Laragon folder around (to another disks, to another laptops, sync to Cloud,…) without any worries.
* Isolated: Laragon has an isolated environment with your OS - it will keep your system clean.
* Easy Operation: Unlike others which pre-config for you, Laragon auto-configs all the complicated things. That why you can add another versions of PHP, Python, Ruby, Java, Go, Apache, Nginx, MySQL, PostgreSQL, MongoDB,… effortlessly.
* Modern & Powerful: Laragon comes with modern architect which is suitable to build modern web apps. You can work with both Apache & Nginx as they are fully-managed.

Also, Laragon makes things a lot easier:

Wanna have a Wordpress CMS? Just 1 click.

Wanna show your local project to customers? Just 1 click.

Wanna enable/disable a PHP extension? Just 1 click. [19]

1. **Connect Front-end and Back-end**

Because front-end and back-end of the website use different types of code languages so to connect those two parts we have to use some specials method.

There are a lot of method on the internet to connect front-end part and back-end part, connect two different code languages. There are lots of choice, but the choice is how the coders like to choose, old way or modern way, long way but easy or short way but hard, it’s all just the choice. And in this project will use Laragon.

To explain it simply, Laragon is basically similar to XAMPP, as it comes with MySQL or MariaDB and a management tool like phpMyAdmin or Adminer, making database management and usage quite easy. I have implemented AJAX requests to the back-end server to fetch data from the database, and that is how I link the front-end and back-end together.

1. **Conclusion**

Finally, all the information above consumes the theories I use in this project to develop, create, and design the News Website for. To put this project website on the internet and can put it to use, this project lacks lots of modern techniques like REACT to make the code run smoother or some framework like NodeJS or something else. But with just the languages above the website look good enough and can put on the internet.

# **CHAPTER 3: EXPERIMENTAL RESULTS**

* 1. **Diagrams**

1. **ER Diagrams**

The definition of diagrams and what kind of diagrams this project use already said in chapter 2, so let’s get to point and show the result I archive in this project.

Ảnh có chứa văn bản, ảnh chụp màn hình, biểu đồ, Song song

Mô tả được tạo tự động

**Figure** 3.1: ER diagram of News Website

This figure shows all the ER diagram but with the size it is very hard to read and explain so I am going to cut it to two part to explain the diagram easier.

Ảnh có chứa văn bản, biểu đồ, ảnh chụp màn hình, Song song

Mô tả được tạo tự động

**Figure** 3.2: Cut ER diagram of News Website 1

Hãy bắt đầu với phần users trước,

Ảnh có chứa văn bản, ảnh chụp màn hình, biểu đồ, Song song

Mô tả được tạo tự động

**Figure** 3.3: Cut ER diagram of News Website 2

Moving on to the second part of the ER diagram, I will use the posts table as the center and start explaining from it. The posts table has a foreign key connected to the users table, which is staff\_uuid. This is a one-to-one relationship because a specific post can only be published by one person. Conversely, a person (Author, Admin) can post many articles, so this will be a zero-to-many relationship because there might be people who do not post any news at all.

For the news\_categories table, since this news website requires users to select a specific category before reading news, the posts table will have a foreign key, categories\_id, connected to the id of the news\_categories table. The posts table will connect to the news\_categories table with a one-to-one relationship because a post can only belong to one specific category. Conversely, similar to the users table, a category can contain many or no news articles, making it a zero-to-many relationship.

Next, since the comments and likes tables are quite similar, I will explain both together. Both tables have two foreign keys: user\_uuid and post\_id. Firstly, with the users table, both are connected by a one-to-one relationship, meaning a like or a comment only belongs to a specific person. Conversely, a person can have many likes and comments by liking and commenting on different posts or none at all, making it a zero-to-many relationship. As for the posts table, both are connected to it in the same way as the users table: a like and a comment can only belong to a specific post. However, a post can have no likes and comments or many, thus creating a zero-to-many relationship.

In summary, there are two main relationships frequently used in this ER diagram: one-to-one and zero-to-many. Although one-to-many is a commonly used relationship, it does not exist in this ERD.

1. **Use Case Diagrams**

Skip all of that, now we come to next type of diagram used in this project, Use Case diagrams. There are two types of UC diagram in this project, it is Admin part UC and Doctor part UC.

Let’s show the general use case first so people can imagine how the website work.

Ảnh có chứa bản phác thảo, hình vẽ, biểu đồ, Nghệ thuật vẽ nét đơn

Mô tả được tạo tự động

**Figure** 3.4: General Use Case of the website

First, let's discuss how the website operates for all three actors: Users (ordinary news readers), Authors, and Admins of this news website. For ordinary users, when they visit the website, they can choose specific topics to read, such as international health issues or local medical advancements. Then, they can select the news articles they want to read. They can like articles if they find them interesting or leave comments if they have any opinions on the news. However, they need to log in to perform these actions. Users can also filter articles by views or likes and provide feedback to the hospital if they have any contributions to make.

On the Author and Admin side, they have access to all the functions available to regular users. Additionally, they can access the admin dashboard to publish articles, create user accounts, or add topics for news. However, Authors are limited in some tasks that only Admins can perform, which will be further detailed in the subsequent sections of the report, including detailed use cases and specifications.

Ảnh có chứa biểu đồ, vòng tròn, văn bản, hàng

Mô tả được tạo tự động

**Figure** 3.5: Login UC for all actors.

|  |  |
| --- | --- |
| Use Case Name | Login |
| Use Case Description | This use case describes the process by which a user (Author, User, or Admin) logs into the system. The login process includes authentication, saving user login credentials, and providing an option to stay signed in. The user can also log out from the system. |
| Actors | Users, Author, Admin |
| Preconditions | All users have to have an account in the system |
| Postconditions | - Go to pick topics page when login successful.  - The user's login credentials are saved. |
| Main Flow | - The user opens the login page.  - The user enters their username and password.  - The system validates the entered credentials (include Authentication).  - If the credentials are valid:  + The system saves the user's login credentials (include Save User Login Credentials).  + The system provides stay signed in.  + The user is authenticated and redirected to the main page.  If the credentials are invalid, the system displays an error message and prompts the user to re-enter their credentials. |
| Alternate Flows | none |
| Special Requirements | - The system must encrypt passwords before storing them.  - The login process should not take more than 2 seconds.  - The "Stay Signed In" functionality should use secure cookies. |

Ảnh có chứa biểu đồ, bản phác thảo, hình vẽ, vòng tròn

Mô tả được tạo tự động

**Figure** 3.6: Watch news UC for all actors.

|  |  |
| --- | --- |
| Use Case Name | Watch news |
| Use Case Description | This use case describes the process by which a user (Author, User, or Admin) selects a news topic, searches for news, sorts news by views or likes, and picks a specific news article to view. The user can also like, comment on, and watch the news. |
| Actors | Users, Author, Admin |
| Preconditions | - The user must be logged into the system to like or comment on the news.  - The system must have news articles available for viewing. |
| Postconditions | - The user can view the selected news article.  - If the user likes or comments on the news, their interactions are saved.  - Feedback is sent if the user chooses to send it. |
| Main Flow | - The user selects a news topic (Pick News Topic).  - The user may search for specific news articles (Search).  - The user may sort news by views or likes (Sort News by Views or Likes).  - The user picks a specific news article to view (Pick News to See).  - The user views the news article (Watch News). |
| Alternate Flows | - Likes news: The user likes the news article (Like News), The system saves the like.  - Comment on News: The user comments on the news article (Comment News), The system saves the comment.  - Send Feedback: The user chooses to send feedback (Send Feedback), The system sends the feedback to the hospital email (Send to Hospital Email). |
| Special Requirements | - The system must ensure that only logged-in users can like or comment on news articles.  - The system should provide a responsive interface for searching and sorting news articles. |

Ảnh có chứa biểu đồ, vòng tròn, văn bản, bản phác thảo

Mô tả được tạo tự động

**Figure** 3.7: Admin dashboard UC for Author

|  |  |
| --- | --- |
| Use Case Name | Admin dashboard UC for Author |
| Use Case Description | This use case describes the process by which an author selects a news topic from the available options. This is a first step before the author can proceed to the Admin Dashboard to manage posts or watch news content. |
| Actors | Author |
| Preconditions | The author must be logged into the system. |
| Postconditions | Author can go to admin dashboard after login successful. |
| Main Flow | - The Author logs into the system (this includes the "Login" use case).  - The Author navigates to the "Pick News Topic" section.  - The Author selects a desired news topic from the list of available topics.  - The system records the selected news topic.  - The system redirects the author to the Admin Dashboard. |
| Alternate Flows | - If the author is not logged in: The system does not show the button to go to admin dashboard, upon successful login, the author is redirected to the "Pick News Topic" section. |
| Special Requirements | - The system should provide a user-friendly interface for selecting news topics.  - The Author can go to admin-dashboard easily. |

Ảnh có chứa văn bản, biểu đồ, vòng tròn

Mô tả được tạo tự động

**Figure** 3.8: Posts Management UC for Author

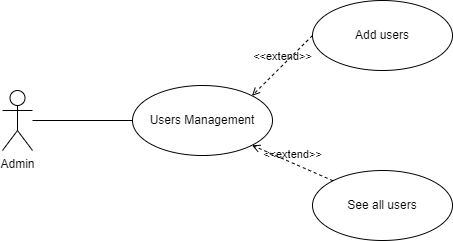
|  |  |
| --- | --- |
| Use Case Name | Posts Management UC for Author |
| Use Case Description | This use case describes the process by which an author manages posts, including adding new posts, editing existing posts, and see their own posts. The posts must be approved by an admin before they are published on the website. |
| Actors | Author |
| Preconditions | The author must be logged into the system. |
| Postconditions | - Posts are added, edited, or see own posts as specified by the author.  - Posts are submitted for admin approval before being published on the website. |
| Main Flow | - The Author logs into the system.  - The Author go to Admin dashboard.  - The Author can choose to add post or edit posts after going to their own posts section.  - For adding a new post:  1: Author selects "Add Post."  2: Author enters the post details and submits it.  3: The system includes the "Admin Approved" use case.  4: If approved by the admin, the post is put on the website. |
| Alternate Flows | - If the post is not approved by the admin:  + The author can choose to edit the post and resubmit it for approval. |
| Special Requirements | - The system should provide a user-friendly interface for managing posts.  - The system should ensure that the posts are saved and submitted accurately.  - The admin approval process should be efficient to avoid delays in publishing posts. |

Ảnh có chứa biểu đồ, văn bản, vòng tròn

Mô tả được tạo tự động

**Figure** 3.9: Posts Management UC for Admin

|  |  |
| --- | --- |
| Use Case Name | Posts Management UC for Admin |
| Use Case Description | This use case describes the functionality that allows an admin to manage posts on a website. The admin can view all posts, approve posts, edit posts, and delete posts. |
| Actors | Admin |
| Preconditions | The admin must be logged into the system. |
| Postconditions | - Posts are added, edited, approved, or deleted as specified by the admin.  - Approved posts are published on the website. |
| Main Flow | - The Author logs into the system.  - The Author goes to Admin dashboard.  - Admin can choose to:  + Add a new post.  + Edit an existing post.  + Approve posts.  + Delete posts from the website.  - For adding a new post:  + Admin selects "Add Posts."  + Admin enters the post details and submits it.  + The system includes the "Admin Approved" use case.  + If approved by the admin, the post is put on the website.  - For editing an existing post:  + Admin selects "Edit Posts." after click in “My Posts” or “Manage Posts”.  + Admin modifies the post details and submits it.  - For approving posts:  + Admin selects "Approve Posts."  + Admin reviews the post details.  + If approved, the post is put on the website.  - For deleting posts:  + Admin selects Delete button in Manage Posts.  + Admin confirms the deletion.  + The post is removed from the website but not in the database. |
| Alternate Flows | If admin not login then admin can’t go to admin-dashboard. |
| Special Requirements | - The system should provide a user-friendly interface for managing posts.  - The system should ensure that the posts are saved and submitted accurately.  - The admin approval process should be efficient to avoid delays in publishing posts.  - The add post process should include validation to ensure that all necessary user details are provided and correct. |



**Figure** 3.10: Users Management UC for Admin

|  |  |
| --- | --- |
| Use Case Name | Users Management UC for Admin |
| Use Case Description | This use case describes the process by which an admin manages users, including adding new users and viewing a list of all users within the system. The admin must be logged into the system to manage users. |
| Actors | Admin |
| Preconditions | The admin must be logged into the system. |
| Postconditions | - Users are added to the system or viewed as specified by the admin.  - The system reflects the current list of users accurately. |
| Main Flow | - The Author logs into the system.  - The Author goes to Admin dashboard.  - Admin can choose to:  + Add a new user.  + See all users.  - For adding a new user:  + Admin selects "Add Users."  + Admin enters the user details and submits the form.  + The system validates and saves the new user information.  - For seeing all users:  + Admin selects "See All Users."  + The system retrieves and displays the list of all users. |
| Alternate Flows | - If admin not login, then admin can’t go to admin-dashboard.  - If the user details are invalid during the add user process:  + The system displays an error message.  + The admin corrects the user details and resubmits the form. |
| Special Requirements | - The system should provide a user-friendly interface for managing users.  - The system should ensure that the users information are saved and submitted accurately.  - The add user process should include validation to ensure that all necessary user details are provided and correct. |

.

Ảnh có chứa biểu đồ, văn bản, vòng tròn, hàng

Mô tả được tạo tự động

**Figure** 3.11: Categories Management UC for Admin

|  |  |
| --- | --- |
| Use Case Name | Categories Management |
| Use Case Description | This use case describes the process by which an admin manages categories within the system. This includes adding new categories, editing existing categories, and deleting categories. |
| Actors | Admin |
| Preconditions | The admin must be logged into the system. |
| Postconditions | - Categories are added, edited, or deleted as specified by the admin.  - The system reflects the current list of categories accurately. |
| Main Flow | - The Author logs into the system.  - The Author goes to Admin dashboard.  - Admin can choose to:  + Add a new category.  + Manage existing categories (edit or delete)  - For adding a new category:  + Admin selects "Add Category."  + Admin enters the category details and submits the form.  + The system validates and saves the new category information.  - For managing existing categories:  + Admin selects "Manage Categories."  + The system displays a list of existing categories.  + Admin can choose to:  ++ Edit a category.  ++ Delete a category.  - For editing a category:  + Admin click category button to edit.  + Admin updates the category details and submits the form.  + The system validates and saves the updated category information.  - For deleting a category:  + Admin selects a category to delete.  + The system prompts for confirmation.  + Admin confirms the deletion.  + The system deletes the category from the website and updates the list of categories.  + Change all news in deleted category to uncategorized. |
| Alternate Flows | - If admin not login, then admin can’t go to admin-dashboard.  - If the category details are invalid during the add or edit process:  + The system displays an error message.  + The admin corrects the category details and resubmits the form.  - If the admin cancels the add, edit, or delete action:  + The system discards the changes and returns to the previous state. |
| Special Requirements | - The system should provide a user-friendly interface for managing categories.  - The system should ensure that the category information are saved and submitted accurately.  - The add and edit category processes should include validation to ensure that all necessary category details are provided and correct. |

* 1. **User interface and admin dashboard of the website**

1. **User interface**

Ảnh có chứa văn bản, ảnh chụp màn hình, thiết kế đồ họa, người

Mô tả được tạo tự động

Figure 3.12: Pick Topic to watch news page.

First, anyone, whether they are a regular news reader, an Author, or an Admin, must visit this page to select the specific news topic they want to view.

Ảnh có chứa văn bản, phần mềm, Phần mềm đa phương tiện, Website

Mô tả được tạo tự động

Figure 3.13: News Section 1.

When you have selected a preferred topic to start exploring medical news (for example, the hot medical news topic shown in the image), this page will appear with all the features of a typical news website. The largest image that immediately catches the user's attention will be the latest news just posted. On the right-hand side, users can choose to view either the 4 most viewed news articles or the 4 latest news articles. Also the header will look a little bit different between the user not login with user already login.

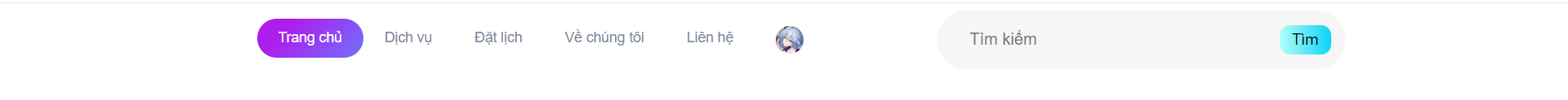


Figure 3.14: Header that user already login.

As you can see, when the user logs in, the header will display the user's avatar instead of just the login text as before.

Ảnh có chứa văn bản, ảnh chụp màn hình

Mô tả được tạo tự động

Figure 3.15: News section 2

When users continue to scroll down, they will reach the section showcasing news articles that have garnered a lot of views, indicating their popularity. Additionally, by clicking the "See More" button, users will be taken to a page where they can view all news articles sorted by view count or popularity. On the right-hand side, there will be an introduction to G37 General Hospital.

Ảnh có chứa văn bản, ảnh chụp màn hình, thiết kế

Mô tả được tạo tự động

Figure 3.16: Sort news by like or views 1.

Ảnh có chứa văn bản, ảnh chụp màn hình, Website, Trang web

Mô tả được tạo tự động

Figure 3.16: Sort news by like or views 2.

After clicking the "See More" button mentioned above, users will be taken to this page. Here, users can easily switch between the most viewed and most liked news articles, which are sorted in descending order.

Ảnh có chứa văn bản, ảnh chụp màn hình, Website, Quảng cáo trực tuyến

Mô tả được tạo tự động

Figure 3.17: News section 3.

Next, on the left-hand side is a tool for users to quickly and conveniently select different news topics to view without having to go back to the topic selection page. On the right-hand side is a table consisting of the top four articles with the most likes for users to refer to.

Ảnh có chứa văn bản, ảnh chụp màn hình, Website, Quảng cáo trực tuyến

Mô tả được tạo tự động

Figure 3.18: Main news 1.

Ảnh có chứa văn bản, ảnh chụp màn hình, Mặt người, Website

Mô tả được tạo tự động

Figure 3.19: Main news 2

Ảnh có chứa văn bản, máy tính, ảnh chụp màn hình, Website

Mô tả được tạo tự động

Figure 3.20: Main news 3.

Because this is the main news section of the website, this part features all the news articles arranged in chronological order, starting with the most recent ones. Furthermore, as shown in Figure 3.19, initially, only a few news articles are displayed. If users want to see more, they can click on the "View more" button to reveal additional articles, as depicted in Figure 3.20. I implemented this to limit the need for loading hundreds of news articles every time someone visits the page, thus preventing unnecessary strain on page loading speed.

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, tài liệu

Mô tả được tạo tự động

Figure 3.21: Content of the news.

When users select an interesting news article, they will be directed to a page where they can read the entire article. This page includes the article title, the article content, the author's name, and the time the article was posted.

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm

Mô tả được tạo tự động

Figure 3.22: Likes and Comments Section 1.

At the end of each article, there's a section for likes, where anyone who finds the content enjoyable or beneficial can express their appreciation. Furthermore, users can also share their thoughts by commenting on the article and discussing its content.

Ảnh có chứa văn bản, ảnh chụp màn hình, màn hình, Phông chữ

Mô tả được tạo tự động

Figure 3.23: Likes and Comments Section 2.

If not logged in, users will be notified to log in to comment or like the content.

And that concludes the user-side portion. Now, let's move on to the admin-dashboard section, where articles and categories are managed and created.

1. **Admin dashboard**

Due to the distinction between authors and administrators, this section will be divided into two parts: the author's section and the admin’s section.

**3.2.2.1: Author’s section.**

Ảnh có chứa văn bản, phần mềm, số, ảnh chụp màn hình

Mô tả được tạo tự động

Figure 3.24: Add news.

Both authors and admins can easily add news using this tool.

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Trang web

Mô tả được tạo tự động

Figure 3.25: My Posts section.

Here, authors can review all the articles they've posted. They can also check the status of their articles to see if they have been approved by the admin or not.

**3.2.2.1: Admin’s section.**

Ảnh có chứa văn bản, phần mềm, Biểu tượng máy tính, Trang web

Mô tả được tạo tự động

Figure 3.26: Approval section.

Here, all the posts awaiting approval will appear for the admin to review before publishing them on the official website. The admin can preview the posts to check, and if everything is okay, just click on the approve button, and the post will be published immediately.

Ảnh có chứa văn bản, ảnh chụp màn hình, số

Mô tả được tạo tự động

Figure 3.27: Add user tool.

Ảnh có chứa văn bản, phần mềm, số, ảnh chụp màn hình

Mô tả được tạo tự động

Figure 3.28: Add categories tool.

Because the interface and usage are similar to the news section introduced in the Author section, I won't elaborate much on this part.

Ảnh có chứa văn bản, ảnh chụp màn hình, Trang web, Website

Mô tả được tạo tự động

Figure 3.29: Show all news from all author and administrator.

Ảnh có chứa văn bản, phần mềm, Trang web, Website

Mô tả được tạo tự động

Figure 3.29: Show all categories from all author and administrator.

In summary, the admin dashboard of my website doesn't really stand out, but it does have all the necessary features of a news website. It provides all the essential tools for users to post articles, update user information, or add a news topic.

# **CHAPTER 4: CONCLUSIONS AND FUTURE WORKS**

1. **Conclusions**

It must be said, before starting this graduation project, I was truly confused and thought that I might not be able to complete it. However, week after week went by, with meetings with the supervising lecturer and group members. The time spent researching and exploring various websites gradually led to the formation and completion of this website.

Although I ambitiously claimed that it would revolutionize the healthcare news industry for young people in Vietnam, it was truly difficult, and this website is far from being as professional as a typical one. Nonetheless, this is a labor of love that I have poured sweat and effort into for over three months, and I am extremely proud of it as well as of myself.

1. **Future works**

In the future, I want to improve this website in many ways. I want the topic selection section to look better, the interface to be cleaner, and to add more features, such as more hashtags for filtering news like #covid-19 or #chronicdisease. I even envision this project as a 3D news website, providing users with a better and more enjoyable experience when reading boring news.

However, my time and skills are limited, so there are many ideas I haven't been able to implement yet. But it is only a matter of time before I roll up my sleeves and complete a news website that surpasses the current ones.

# **REFERENCES**

[1]. <https://www.techopedia.com/definition/5411/website>

Margaret Rouse, What does a website mean , August 2020.

[2]. <https://www.educative.io/answers/what-are-template-engines>

Theodore Kelechukwu Onyejiaku, What are template engines, October 2022.

[3]. <https://www.mulesoft.com/resources/api/what-is-an-api>

SEO Content Writer at MuleSoft, What is an API?, [Ma-Keba Frye](https://blogs.mulesoft.com/author/Ma-Keba-Frye/) - 2019.

[4]. <https://www.techtarget.com/searchdatamanagement/definition/entity-relationship-diagram-ERD>

[Jacqueline Biscobing](https://www.techtarget.com/contributor/Jacqueline-Biscobing), Entity Relationship Diagram (ERD) , 2020.

[5]. <https://opentextbc.ca/dbdesign01/chapter/chapter-8-entity-relationship-model/> Adrienne Watt, Chapter 8 The Entity Relationship Data Model , 2019.

[6]. <https://learn.microsoft.com/en-us/ef/core/modeling/relationships/one-to-many>

Lisandro Fernigrini, One-to-many relationships , march 30th 2023.

[7]. <https://vertabelo.com/blog/many-to-many-relationship/>

Tihomir Babic, What Is a Many-to-Many Relationship in a Database? An Explanation with Three Examples , December 3 2020.

[8]. <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-use-case-diagram/>

Christopher Alan, What is Use Case Diagram, 2022.

[9]. <https://www.geeksforgeeks.org/unified-modeling-language-uml-activity-diagrams/>

Geeksforgeeks, Activity Diagrams | Unified Modeling Language (UML), January 15, 2024.

[10]. <https://www.lucidchart.com/pages/uml-activity-diagram>

Barry Allen, UML Activity Diagram Tutorial, 2022.

[11]. <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-sequence-diagram/>

Arthur Morgran, What is Sequence Diagram?, July 5 2021.

[12]. <https://www.freecodecamp.org/news/frontend-vs-backend-whats-the-difference/> [Dionysia Lemonaki](https://www.freecodecamp.org/news/author/dionysia/), Frontend VS Backend – What's the Difference, March 18 2022.

[13]. <https://www.hostinger.com/tutorials/what-is-html>

Astari S, What Is HTML? Hypertext Markup Language Basics Explained, May 02, 2023

[14]. <https://www.hostinger.com/tutorials/what-is-css>

Domantas G, What Is CSS and How Does It Work, May 15, 2023.

[15]. <https://www.hostinger.com/tutorials/what-is-javascript#What_Is_JavaScript> Jordana A, What Is JavaScript, A Basic Introduction to JS for Beginners , Jan 31, 2024

[16]. <https://www.techtarget.com/whatis/definition/bootstrap>

Andrew Zola, Definition of Bootstrap, August 2022.

[17]. <https://www.php.net/manual/en/intro-whatis.php>

What is PHP,[Copyright © 2001-2024 The PHP Group](https://www.php.net/copyright.php).

[18]. <https://www.php.net/manual/en/intro-whatcando.php>

What can PHP do, [Copyright © 2001-2024 The PHP Group](https://www.php.net/copyright.php).