**GROUP PROJECT FRONT SHEET**

|  |  |  |  |
| --- | --- | --- | --- |
| **Qualification** | **BTEC Level 5 HND Diploma in Computing** | | |
| **Unit number and title** | WEBG301 - Project Web | | |
| **Submission date** | 31/10/2023 | **Date Received 1st submission** | 31/10/2023 |
| **Re-submission Date** |  | **Date Received 2nd submission** |  |
| **Student Name** | Bui Duy Hung | **Student ID** | GCH200680 |
| **Class** | GCH1108.1 | **Assessor name** | Pham Duc Tho |
| **Student declaration**  I certify that the assignment submission is entirely my own work and I fully understand the consequences of plagiarism. I understand that making a false declaration is a form of malpractice. | | | |
|  |  | **Student’s signature** | Hung |

|  |
| --- |
|  |

**Grade**

|  |  |  |
| --- | --- | --- |
| **❒ Summative Feedback: ❒ Resubmission Feedback:** | | |
| **Grade:** | **Assessor Signature:** | **Date:** |
| **Signature & Date:** | | |

Table of Contents

[1. User’ requirements (group) 7](#_Toc149473803)

[1.1. User Stories template 7](#_Toc149473804)

[1.2. Use case diagram (optional) 8](#_Toc149473805)

[2. System Design (Group) 10](#_Toc149473806)

[2.1. Site map 10](#_Toc149473807)

[2.2. Entity Relationship Diagram 12](#_Toc149473808)

[2.3. Wireframes 12](#_Toc149473809)

[3. System Implementation (Individual) 22](#_Toc149473810)

[3.1. Source code 22](#_Toc149473811)

[a. Introduction MVC design pattern in Laravel 22](#_Toc149473812)

[b. Attach project structure 22](#_Toc149473813)

[c. How can develop a Laravel project 36](#_Toc149473814)

[3.2. Web screenshots 39](#_Toc149473815)

[4. Conclusion (Individual) 48](#_Toc149473816)

[4.1. Advantage of website 48](#_Toc149473817)

[4.2. Disadvantages of website 49](#_Toc149473818)

[4.3. Lesson learnt 49](#_Toc149473819)

[4.4. Future Improvements 49](#_Toc149473820)

[5. Appendix (Group) 50](#_Toc149473821)

[5.1. A1 group member list and role (show as table) 50](#_Toc149473822)

[5.2. A2 link to code on Github (set public access) 50](#_Toc149473823)

[Figure 1: User case diagram 10](#_Toc149473856)

[Figure 2: Site map 11](#_Toc149473857)

[Figure 3: Entity Relationship Diagram 12](#_Toc149473858)

[Figure 4: Login/Register Page 13](#_Toc149473859)

[Figure 5: Homepage-feed 14](#_Toc149473860)

[Figure 6: Profile-post page 15](#_Toc149473861)

[Figure 7: Follower page 16](#_Toc149473862)

[Figure 8: Following page 17](#_Toc149473863)

[Figure 9: Create post page 18](#_Toc149473864)

[Figure 10: View post page 19](#_Toc149473865)

[Figure 11: Edit post page 20](#_Toc149473866)

[Figure 12: Avatar page 21](#_Toc149473867)

[Figure 13: MVC 24](#_Toc149473868)

[Figure 14: Model User 26](#_Toc149473869)

[Figure 15: UserController 30](#_Toc149473870)

[Figure 16: Post model 31](#_Toc149473871)

[Figure 17: PostController 33](#_Toc149473872)

[Figure 18: Follow model 34](#_Toc149473873)

[Figure 19: FollowController 35](#_Toc149473874)

[Figure 20: View 36](#_Toc149473875)

[Figure 21: Composer, php and MySQL environment 37](#_Toc149473876)

[Figure 22: .env file 38](#_Toc149473877)

[Figure 23: Database MySQL 39](#_Toc149473878)

[Figure 24: Homepage 40](#_Toc149473879)

[Figure 25: Homepage-feed 41](#_Toc149473880)

[Figure 26: Create-post page 42](#_Toc149473881)

[Figure 27: View-post page 43](#_Toc149473882)

[Figure 28: Update-post page 44](#_Toc149473883)

[Figure 29: Profile-post page 45](#_Toc149473884)

[Figure 30: Following page 46](#_Toc149473885)

[Figure 31: Following page 47](#_Toc149473886)

[Figure 32: Avatar page 48](#_Toc149473887)

[Table 1: User stories 7](#_Toc149473738)

[Table 2: Group member 49](#_Toc149473739)

# 1. User’ requirements (group)

## 1.1. User Stories template

|  |  |  |  |
| --- | --- | --- | --- |
| No | As a <type of user/personal> | I want to <global objective> | So that <benefit/result/reason> |
| 1 | Admin | Login/logout | I can login/logout my account in the blog post web. |
| 2 | Admin | Register | I can register an account in the blog post web. |
| 3 | Admin | Create a new post | I can create a new post with the information inside as title and content. |
| 4 | Admin | View all my posts and other user’s posts | I can view all my posts and other user’s post with my permission. |
| 5 | Admin | Delete my post and other user’s post | I can delete my post and other user’s post with my permission. |
| 6 | Admin | Update my post and other user’s post | I can update my post and other user’s post with the information inside as title and content. |
| 7 | Admin | Change my avatar | I can change my avatar in the blog post web. |
| 8 | Admin | Follow/unfollow other user account | I can follow/unfollow other user account. |
| 9 | User | Login/logout | I can login/logout my account in the blog post web. |
| 10 | User | Register | I can register an account in the blog post web. |
| 11 | User | Create a new post | I can create a new post with the information inside as title and content. |
| 12 | User | View all my posts | I can view all my posts created from the database. |
| 13 | User | Delete my post | I can delete my post created out the database. |
| 14 | User | Update my post | I can update my post created with the information inside as title and content. |
| 15 | User | Change my avatar | I can change my avatar in the blog post web. |
| 16 | User | Follow/unfollow other user account | I can follow/unfollow other user account. |

Table : User stories

## 1.2. Use case diagram (optional)

This blog post website system includes both Admin and User. Admin can create/view/update/delete my posts and can view/update/delete posts of other users. User also has the same features except the secondary functions of Admin. In addition, both Admin and User can follow and unfollow other users' accounts.

A diagram of a process

Description automatically generated

Figure : User case diagram

# 2. System Design (Group)

## 2.1. Site map

This is our team sitemap including for Admin and User. User belongs to the regular user group and Admin belongs to the highest user group. First, both groups of users can log in, register, and log out of the system. User accounts are logged in and registered normally, except for the Admin account, which will be upgraded based on changes in database rights. A regular user account will be able to create posts, update posts, delete posts and view all posts that have been created. For external Admin accounts, the previous functions are similar to the user's account and will be able to update and delete other users' posts. This is the authority that the Admin account is allowed to use over other user accounts. In addition, both types of accounts can follow other user accounts and there will be two pages showing information about which users are following each other. From user accounts that follow each other, there will be a home page feed that displays posts from user accounts whose accounts we have followed. Finally, there will be the avatar change feature. If the user does not change it then it will have a temporary avatar fixed there.

A diagram of a website

Description automatically generated

Figure : Site map

## 2.2. Entity Relationship Diagram

A computer screen shot of a computer

Description automatically generated

Figure : Entity Relationship Diagram

## 2.3. Wireframes

- Login/Register:

* Login: There are two fields that need to be filled in: username and password to log in
* Register: There are four fields that need to be filled in: username, email, password and password verification to be able to register a new account.

A screenshot of a login form

Description automatically generated

Figure : Login/Register Page

- Homepage-feed: Shows all posts where we have followed other user accounts.

A screenshot of a computer

Description automatically generated

Figure : Homepage-feed

- Profile-post page: Displays all the posts we created previously.

A screenshot of a computer

Description automatically generated

Figure : Profile-post page

- Follower page: Showing user accounts that are following my account.

A screenshot of a computer

Description automatically generated

Figure : Follower page

- Following page: Shows user accounts that I have followed.

A screenshot of a computer

Description automatically generated

Figure : Following page

- Create post page: There are two fields that need to be filled in: title and content to create a complete post.

A screenshot of a computer

Description automatically generated

Figure : Create post page

- View post page: A previously created post appears.

A screenshot of a computer

Description automatically generated

Figure : View post page

- Edit post page: A post page will appear with content data of the post we created previously so we can update the content.

A screenshot of a computer

Description automatically generated

Figure : Edit post page

- Avatar page: Change your profile picture to enrich your website.

A screenshot of a computer

Description automatically generated

Figure : Avatar page

# 3. System Implementation (Individual)

## 3.1. Source code

### a. Introduction MVC design pattern in Laravel

MVC is a web application architectural design model of the Laravel framework, in which the application is divided into three main components: Model, View and Controller. First Model is responsible for processing data and storing data in the database. Next, the View is responsible for displaying data to the user. Finally, the Controller is responsible for controlling the application flow and handling requests from users. This MVC design pattern is widely used in many web projects around the world. The advantages of this model include ease of maintenance, expansion, reuse of components, ease of change and upgrade. On the contrary, it can be more complicated than other models because it has many internal components.

### b. Attach project structure

- Include migrations file and MVC

A screenshot of a computer

Description automatically generated

Figure : MVC

- User code

* Model: The User model includes fields such as 'username', 'email', 'password'. This is important data information to be able to create a user account and especially in it there is an avatar() function that makes the path for your photos to go to the correct folder or the temporary photo is the photo. saved outside the 'public' folder. There are also 'hasManyThrough' relationship functions like feedPosts() to be able to get the posts of people whose accounts we have followed. The followers() function has a 'hasMany' relationship that will retrieve information about the accounts that are following our account. The followingTheseUsers() function has a 'hasMany' relationship to get information about the accounts we follow. Finally, the posts() function has a 'hasMany' relationship to get all the posts of the currently logged in user account.

A screen shot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated

Figure : Model User

* Controller: In UserController, the user will first have to log in and register an account to access the web system. The fields in the function are tightly linked and have validation to provide rules for appropriate account naming. The logout() function allows us to directly log out of the account we are using and go to the main page with login and registration. There are also functions such as showing posts of logged in user accounts profile(), showing user accounts following our account profileFollowers(), and showing user accounts we follow profileFollowing( ). Finally, the storeAvatar() function helps us change the avatar as desired.

A screen shot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screen shot of a computer program

Description automatically generated

A computer screen shot of text

Description automatically generated

Figure : UserController

- Post code

* Model: In the Post model we will have two fields 'title' and 'body' to be able to create a complete post. There is also a function user() with the 'belongsTo' relationship to determine the dependency relationship with the User table through 'user\_id'.

A screenshot of a computer program

Description automatically generated

Figure : Post model

* Controller: In PostController, there will be functions that can save the data information the user has entered to push it to the database and from there retrieve it and bring it into the user's website. In addition, there will be CRUD functions for each user's post.

A screen shot of a computer program

Description automatically generated

A screen shot of a computer program

Description automatically generated

Figure : PostController

- Follow code

* Model: In the Follow model, we will have two functions with the relationship "belongsTo" to show the dependence on the User table through 'user\_id' and 'followeduser' to be able to determine who is following us or who we are following.

A screenshot of a computer

Description automatically generated

Figure : Follow model

* Controller: In FollowController, there will be two functions representing two functions: create and remove follow other user accounts.

A screenshot of a computer program

Description automatically generated

Figure : FollowController

- View: In the view, it will represent all that the user can see and the information we have entered.

A screenshot of a computer

Description automatically generated

Figure : View

### c. How can develop a Laravel project

To be able to develop a Laravel project, it will include the following steps:

* Install Composer
* Install Composer environment in windows operating system
* Install the MySQL application
* Install MySQL environment in windows operating system
* Install php version and php environment in windows operating system

A screenshot of a computer program

Description automatically generated

Figure : Composer, php and MySQL environment

* After installing the above step, use Command Prompt to create a Laravel project: composer create-project laravel/laravel my-project
* Configure the database through the .env file include lines number 14, 15, 16A screenshot of a computer

  Description automatically generated

Figure : .env file

* Use cmd to migrate all changes into the database: php artisan migrate

A screenshot of a computer

Description automatically generated

Figure : Database MySQL

* Run web: php artisan serve

## 3.2. Web screenshots

- Login/Register

A screenshot of a computer

Description automatically generated

Figure : Homepage

- Homepage-feed

A screenshot of a computer

Description automatically generated

Figure : Homepage-feed

- Create-post page with function create a new post

A screenshot of a computer

Description automatically generated

Figure : Create-post page

- View-post page with function update, delete a post

A screenshot of a computer

Description automatically generated

Figure : View-post page

- Update-post page:

A screenshot of a computer

Description automatically generated

Figure : Update-post page

- Profile-post page

A screenshot of a computer

Description automatically generated

Figure : Profile-post page

- Follower page

A screenshot of a computer

Description automatically generated

Figure : Following page

- Following page

A screenshot of a computer

Description automatically generated

Figure : Following page

- Avatar page

A screenshot of a computer

Description automatically generated

Figure : Avatar page

# 4. Conclusion (Individual)

## 4.1. Advantage of website

- The minimalist user interface is not too colorful, bringing a pleasant feeling to the user.

- Functions operate normally without errors.

- The website's query speed is stable.

## 4.2. Disadvantages of website

- The structure of the functions on the website has not been optimized, so the user experience is not truly perfect.

- The functions are not really groundbreaking for users.

- The user interface is minimalist but does not really make users feel that the website is simple.

## 4.3. Lesson learnt

- Knowing how to use Laravel framework and MCV model in web design makes my web work easier, more convenient and scientific.

- Understand Laravel framework functions and commands.

- Learn how to connect and create database in Laravel easily with command

- Learn how to use GitHub for convenient teamwork and source code management

- Integrate security features such as authentication, encryption, etc. to protect applications from attacks.

- Regularly test and maintain the application to ensure it operates stably and safely.

## 4.4. Future Improvements

- Unique user interface while still maintaining user convenience.

- Upgrading existing functions makes the user experience feel better.

- Develop more unique features to make users more interested in the website.

- Limit potential errors that occur during web development.

# 5. Appendix (Group)

## 5.1. A1 group member list and role (show as table)

|  |  |  |
| --- | --- | --- |
| No | Name | Role |
| 1 | Bui Duy Hung | Front-end, Back-end, Entity: User page, Follow page |
| 2 | Nguyen Thanh Tung | Front-end, Back-end, Entity: Post page |

Table : Group member

## 5.2. A2 link to code on Github (set public access)

- Link github Project Web: [Click here](https://github.com/HungBuii/Laravel_project_1.git)