CST2355 302 Group3

Assignment 2

Part I: our team

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Raised hand with solid fill | Group 3 members:   |  |  | | --- | --- | | **Saebyul Yun** | **#** | | **Kuang-I Ho** | **#** | | **Shang-Yuan Chang** | **# 041091887** | |

Part II: Choosing your projecct

|  |  |  |
| --- | --- | --- |
| Raised hand with solid fill | Project Topic:   |  | | --- | | **Blog Platform**  [**Github Repository**](https://github.com/aien0236/CST8285_302_Group3_Assignment2_BlogPlatform.git) | |

Part III: project submission

## Breakdown of Tasks

A breakdown of the tasks assigned to each team member.

|  |  |  |
| --- | --- | --- |
| Tasks | Related Files | Assigned Member |
| General Style Design | Style.css / main.js | Saebyul Yun |
| Homepage Design | Home.html / home.css / home.js | Saebyul Yun |
| Signup Page Design | Signup.html / signup.css / signup.js | Kuang-I Ho |
| Posts and Comment Page | Post.html / comment.html / post.css / post.js | Shang-Yuan Chang |
| Functionality Guide | Functionality Guide.docx | Kuang-I Ho |
| Database Description | Database Description.docx | Shang-Yuan Chang |
| Wireframe | Wireframe.docx | Kuang-I Ho |
| Sitemap | sitemap.png | Shang-Yuan Chang |
| Database Design / Default Data Generation / PHP Scripts | Add\_comment.php / add\_post.php / database.php / delete\_post.php / filter\_posts.php / db.sql | Saebyul Yun, Kuang-I Ho, Shang-Yuan Chang |

Part IV: Documentations

Functionality Guide

## The Welcome Page

|  |  |
| --- | --- |
| Raised hand with solid fill | Currently, there are some features that haven't been implemented yet, but overall, we've provided a solid default framework. This ensures that no matter how you want to utilize the homepage in the future, it won't be an issue. It serves as the facade of the entire website. |

* Having a basic default framework, with the top featuring a carousel of popular news.
* The default framework below is designed to provide some simple explanations, helping users get familiar quickly.
* Lastly, there are some image frames that can be filled as needed.

## The Signup Page

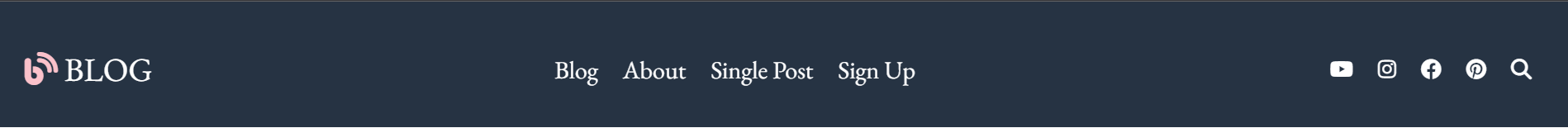
|  |  |
| --- | --- |
| Raised hand with solid fill | The registration functionality is not fully implemented in terms of backend-user binding, but the overall framework has been constructed. |

* Providing user registration functionality.
* Having input validation to ensure users provide correct information.
* Subsequently, the ability to link users with the backend database.

## The Posts and Comment Page

|  |  |
| --- | --- |
| Raised hand with solid fill | Leave some empty space on both sides for future use, whether it's for advertisers or other functions. |

* The main function is to search for posts from the database and display them.
* Providing basic functions for adding, deleting, modifying, and querying.
* You can click on a post to leave comments!



1. To register, please use the navigation bar to go to the "Sign Up" page.

A screen shot of a login form

Description automatically generated

1. Fill out the sign-up page without receiving any error messages regarding invalid information.

A screenshot of a computer screen

Description automatically generated

1. Click on the "search" button to easily find any article on the single page.
2. To create a blog post, simply click on the "+" button. Then you will see below page.

A screenshot of a chat

Description automatically generated

1. If you want to delete the article you wrote, you can click the article then you will see the “Delete” button on the top.

A person standing on a rock

Description automatically generated

1. To leave a comment, click on the article and enter your comment. (In the red rectangle.)

A person standing on a rock

Description automatically generated

Database Description

|  |  |
| --- | --- |
| Raised hand with solid fill | Currently, there are only two simple tables for storing posts. In the future, we can add a user account table to link user accounts and control content visibility. |

## Table: blog\_posts

1. Primarily used to store user posts.
2. The primary key is automatically incremented.
3. Images are stored as file paths, not the actual image files.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| id | title | content | category | Image\_path | author | date |
| PK  INT (11) | DEFAULT NULL  VARCHAR (100) | DEFAULT NULL  TEXT | DEFAULT NULL  VARCHAR (50) | DEFAULT NULL  VARCHAR (50) | DEFAULT NULL  VARCHAR (50) | DEFAULT NULL  DATETIME |

## Table: comment

1. Primarily used to store user comments.
2. Linked to the "blog\_post" table using the foreign key "post\_id".

|  |  |  |  |
| --- | --- | --- | --- |
| id | comment | author | post\_id |
| PK  INT (11) | DEFAULT NULL  VARCHAR (100) | DEFAULT NULL  VARCHAR (30) | FK  INT (11) |