Documentation for CPE 315

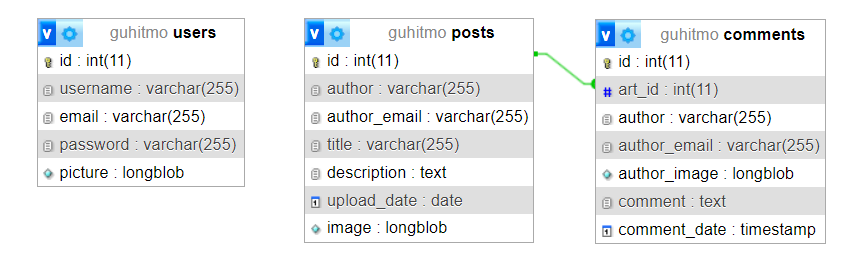
1. Introduction:

* Brief overview of the backend system.
  + PHP and MySQL was used as the backend part of the system. This was used in the login and registration page, profile editing, art creation, and comment section.
* Purpose and goals of the backend.
  + Our backend aims to handle user authentication, art content management, and art interactions (comments).

1. Getting Started:

* Prerequisites for setting up the development environment.
  + Install any text editor, but for us we used VSCode with an extension necessary for running PHP application. Install xampp for the database connection and toggle the Apache and MySQL.
* Instructions for installing necessary dependencies.
  + Download PHP Intelliphense and PHP Server for running our PHP Files in the VSCode.
* Steps to set up the project locally.
  + Download the project and import the sql to the database. We can use the VSCode as our text editor with proper PHP extenstions and xampp for the local database. For the PHP Extension, we can download PHP Intelliphense and PHP Server for running our PHP Files. In the xampp application, we must toggle Apache and MySQL before running our project.

1. Architecture: (Pwede sa SQL nalang, para auto generated na yung erd diagram)

* Overview of the backend architecture.
  + 
* Explanation of the design patterns and principles used.
  + The ‘users’ are used to store all the registered accounts which can be used for the author, author\_email, and author\_image part to reflect the user accordingly. The ‘posts’ table contains all the posted art information which retrieves the user information from the ‘users’ table. The same applies to ‘comments’ table which retrieves both from the ‘users’ and ‘posts’ tables. The id from the ‘posts’ table is used in art\_id from the ‘comments’ table to determine which post is commented on.
* Diagrams illustrating the system architecture.
  + Pwede nyo na tong iexplore or ma-fillout gamit yung files na nasa gdrive

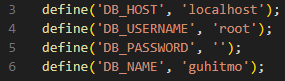
1. Technologies Used:

* List of programming languages, frameworks, libraries, and tools used in the backend.
  + Programming languages: PHP, SQL, JavaScript
  + Markup Language: HTML
  + Stylesheet Language: CSS
  + Frameworks: Bootstrap
  + Tools: VSCode, Hostinger
* Version numbers of each technology.
  + Check nyo nalang kung anung version nung mga ginamit if applicable

1. Directory Structure:

* Explanation of the directory structure.
  + We have 4 folders alongside to all the pages of the website. The 4 folders are ‘assets’ containing the css and js codes, ‘backend’ that contains the config.php, create-post.php, logout.php, ‘components’ that contains the header, footer, and navbar php, ‘images’ that contains all the images used in the website.
* Description of each directory and its purpose.
  + Basically, the ‘asset’ folder contains all the logic and the styling for the front end of the website, in here we can see the bootstrap framework. The ‘backend’ folder have config.php for the connection between our website to the database, create-post.php for the creation of each posts, logout.php for logging out of the session. The ‘components’ have header.inc.php which means that it cannot be run as a standalone that’s why its extension is .inc.php and contains all the initial setup for the html. The navbar.inc.php contains the navbar that can be seen at the top section of each page except login and registration page, this is use to navigate around the website and reflect whether you’re logged in or not. The footer.inc.php is basically the footer of the website. The ‘components’ are created for easier changes in each component so that we can avoid any human errors when change/s are applied. The ‘images’ just contains all the images used in the website. The php pages are created for the pages used in the website.

1. Configuration:

* Explanation of configuration files.
  + The config.php is the connection between our website and the database. This is where we setup the database host, database username, database password, and database name. When we transfer our website to online, the config.php information was changed according to the online database that we used.
* Instructions for modifying configurations for different environments (development, staging, production).
  + This is the sample code in our config.php that will be changed:
  + 
  + When modifying, these will be the only lines that will have any changes. The information that will be used is provided from the online database that will be used.

1. Database Schema:

* Description of the database schema.
  + The database is named ‘guhitmo’ locally and ‘u631567875\_guhitmo’ for the online database which contains 3 tables.
* Explanation of each table, column, and their relationships.
  + The table names are comments, posts, and users. The ‘comments’ table contains all the comments in each art detail pages. The ‘posts’ handles all the posted arts in the website. The ‘users’ table contains all the registered users in the website.
* Sample data if applicable.
  + Kayo nalang maglagay ng sample data dito, yung sakin kasi, puro pangalan ko nakalagay haha

1. Data Models:

* Documentation of data models used in the backend.
* Explanation of each model's attributes and relationships.

1. Testing:

* Instructions for running tests.
  + Pwede nyo na rin to masagot using the website itself
* Description of test suites and test coverage.
  + Same here
* Guidelines for writing new tests.
  + Same here

1. Deployment:

* Instructions for deploying the backend to different environments.
  + Same na rin to ng explanation sa taas na babaguhin yung config.php for the online database
* Configuration management for deployment.
  + Check the necessary information of the online database and change the config.php accordingly
* Monitoring and logging setup.

1. Troubleshooting:

* Common issues and their resolutions.
  + Database setup, the config.php should be changed to a working database whether locally or online before running the website. Also, if it will be run locally, don’t forget to run the xampp application and toggle the Apache and MySQL
* Troubleshooting tips for developers.
  + Always check the config.php and the database tables, it should be like the database that we used, especially the tables and its columns inside them.

1. Contributing:

* Guidelines for contributing to the backend codebase.
* Code style conventions.
  + We utilized bootstrap in our css and js and used some inline styling with custom css files too.
* Contribution workflow.

1. Changelog:

* Record of changes made to the backend over time.
  + For the PHP side, the first thing that were created was the authentication form, then the posting, then commenting, then the profile editing. For the database side, first is the users, then posts, then comments. For the config.php, it is initially setup for the local database, when we transfer it online, it was updated using the online database details.
* Version history with release dates and notable changes.
  + Parang wala naman, pwede nyo na siguro to lagyan ng eme eme lang na sentences