FINAL ASSIGNMENT – SIMPLE, CUSTOM SOCIAL MEDIA FEED IN PHP & MYSQL

Summary

NOTE: YOU WILL BUILD YOUR OWN SIMPLE CUSTOM SOCIAL MEDIA FEED SITE FROM SCRATCH.

You will create a web application in PHP that will serve as a custom Social Media Feed site. The site will consist of two areas:

1) An **administrative back-end area**, the basic structure (i.e. users, posts, themes) of a website can be dynamically created and modified by only authorized users with the appropriate permissions. The additions and changes to the site's administrative structure will all be stored in the custom SQLite/MySQL databases that you will create to support your Social Media Feed site.

2) A **front-end site area** which will dynamically display posts based on the structure stored in the database. The front-end site will be viewable by all users including guests, but users can register and on subsequent visits log in in order to be able to contribute and make some changes to feed content.

The site will be built using the **LARAVEL (Version 10)** Application Framework and styled using **BOOTSTRAP**.

THE PROJECT WILL BE IMPLEMENTED IN 4 PHASES:

Phase 1 Requirements – Site Back-End/Administrative Area

REQ-001 USER LOGIN AND REGISTRATION STYLED WITH BOOTSTRAP

Generate a new Laravel 10 application using the steps that were used in class for the corresponding labs:

Initialize your newly created Laravel app to be a Git repository and tie it to your FinalAssignment repository by adding the appropriate remote. Confirm that you can commit and push your app to your Github Classroom repository so that you've backed it up before proceeding.

After you have generated a new Laravel application, be sure to enable authentication and styling using the Bootstrap CSS framework. These would be the same steps as performed in the in-class Lab 7.

At the end of this requirement, you should have an empty Laravel application styled using Bootstrap 5 and have the Login and Register links available in the upper right corner of your application.

REQ-002 Generating the Required Database Tables and Seed with initial Data

Using Laravel's migration strategies, generate the required tables for Phase 1 based on the provided ERD in the Phase 1 section of Brightspace. Use MySql as your database backend. Be sure to identify in the ERD which tables will require models and controllers to be generated and which ones will not. (Remember that you can always delete artisan-generated files later if you end up not needing them).

When seeding your database be sure to include the following:

1 – roles Table – Seed your database with the following roles: **User Administrator**, **Moderator**, and **Theme Manager**

2 - users Table – Seed your database with following users:

- name: Jane UserAdmin, email: jane@example.com, password: [provide a hashed password]
- name: Bob Moderator, email: bob@example.com, password: [provide a hashed password]
- name: Susan ThemeAdmin, email: susan@example.com., password: [provide a hashed password]

3 - role_user Table – Seed your table so that the appropriate role in the roles table is tied to the appropriate user in the users table.

Once you have completed creating and seeding your database based on the ERD, use SQL Workbench to reverse engineer your database so that it generates an ERD diagram that looks similar to the provided ERD document. https://dataedo.com/kb/tools/mysql-workbench/how-to-reverse-engineer-database You will need to demonstrate this generated ERD as part of your Phase 1 video demonstration.

REQ-003 Create Required pages to Manage Users

This main users management admin page should do the following:

1- List all users who have been assigned one or more roles in the application. (This does not include users who have registered in the application as end-users)
2- Displays a button labeled "Create New Admin User" (In the tutorial screenshot it only shows "Create New User")
3- Provides actions to Show, Edit and Delete administrative users.
    a. Provide the pages to Create and Update users.
    b. Deleting users should be accomplished via a "Soft Delete". View the following video (https://coderstape.com/lesson/187-soft-deletes) and the Laravel documentation (https://laravel.com/docs/10.x/eloquent#soft-deleting) for more information on how to implement Soft Deletes.

At the end of this requirement, you should have the following admin functionality created in your app:

- A users admin page that lists all users and provides buttons to perform required actions on users. (Show, Edit, Delete)
- A create user page that provides fields for creating a new user as well as a field to tie a user to one or more of the available roles.
- An edit page that allows modifications to an existing user.
- The capability to delete a user via Soft Deleting
- All applicable pages should provide basic validation functionality to maintain valid data.

Phase 2 Requirements – Site Front-End/Feed Area

REQ-001 Display of Public Feed Page (Homepage)

End-users navigating to the website in a web browser will be presented with a **main feed page** listing all of the **current posts** submitted by the various users registered on the site.

- It is strongly suggested that you use the available home page that was generated when you installed and ran the Laravel/ui package to show your posts. The view is home.blade.php.
- The posts in the feed should be listed in descending chronological order based on the date and time of posting. (ie, most recent listed first)
- In addition to the desired content to be displayed, each post in the feed should also display the date and time that the post was created. The date format can take one of many forms. The choice is yours.
- A 'Create New Post' button should be presented somewhere on the page when the end-user is logged in but should not be visible when an anonymous user visits the page.
- If the user is currently logged in to the feed, each post in the feed should also display an 'Edit' button, but only for those posts that were created by the logged-in user.
- If the user is currently logged in to the feed, each post in the feed should also display a 'Delete' button, but only under the following conditions:
  - The authenticated user owns the post, or the authenticated user is a Moderator. (Moderators can delete any post)

REQ-002 Creation of Create Post and Edit Post pages, Soft Deletion of Post

As mentioned in REQ-001, users who are authenticated should be presented with buttons on the main feed page to create, edit, and delete certain posts belonging to the user.

Create Post Page

When the authenticated user clicks on the ‘Create New Post’ button, they should be presented with a form that will enable them to create a new post.

- The page should present all fields to create the new post
- The fields should present appropriate validation.

Edit Post Page

When the authenticated user clicks on the ‘Edit’ button for a specific post, they should be presented with a form that will enable them to edit the post.

- The page should present all fields to create the new post
- The fields should present appropriate validation.

Post Soft Deletion

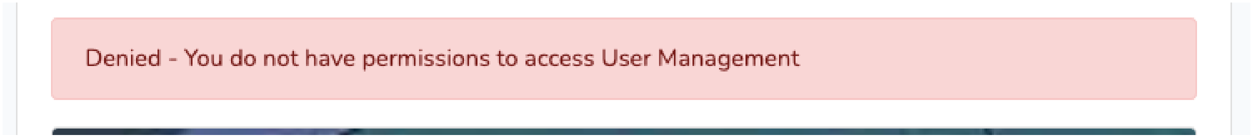
- When the authenticated user clicks on the ‘Delete’ button for a specific post, the post should be soft deleted.

REQ-003 Add necessary Middleware to Prevent unauthorized access to User Administration Pages

Create and implement a Laravel Middleware which checks whether or not the currently authenticated user has been assigned the User Administrator role. (Note: there is already an existing Middleware which checks where the user is authenticated)

If the user is deemed not to have the required role, then the user should be redirected back to the previous page from where they came, and a Session Flash message should be displayed informing the user that they do not have access to the requested page.

Example:

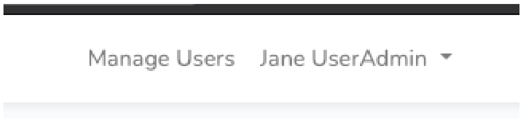


Denied - You do not have permissions to access User Management

REQ-004 Display Administration Links in NavBar

For users who have been assigned the User Administrator role, a link should be displayed in the Navigation bar next to the logged in user’s name which will take the user to the User Administrator’s admin page.

Example:



Phase 3 - Deployment - Dockerize the laravel app

Phase 4 - Finalize the app with more advanced features

- Retrieve the commit number of your last Github commit. This will be considered your final submission for this phase of the assignment.

- Record and submit a screen recording with audio which walks through the requirements for this assignment phase. You should demonstrate that:

1 - You have generated the new themes table as per the provided ERD and you have seeded the table with data. Be sure to include the ERD of your database that you generated using MySQL Workbench reverse engineering.

2 - You have created the required Theme admin pages to List, Show, Create, Update and Delete users. You should also demonstrate an example of validation for one of the pages.

3 - You have only allowed access to Theme Management pages to Theme Managers. All other users are prohibited from accessing the pages.

4 - You display a Manage Themes link in the navigation bar for Theme Managers

5 - You are able to switch themes.