INET 2005 WEB APPLICATION DEVELOPMENT I

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

# Prerequisites

### Strongly Recommended: Lab 6 (MVC), & Labs 7-10 (Framework).

# Summary

NOTE: YOU WILL BUILD YOUR OWN 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** Application Framework and styled using **BOOTSTRAP**.

# Phase 1 Requirements – Site Back-End/Administrative Area

1. **User Login and Registration styled with Bootstrap**

After you have generated a new Laravel application using the laravel new command, be sure to enable authentication and styling using the Bootstrap 4 CSS framework. You can follow the instructions in this article to enable these features. <https://www.itsolutionstuff.com/post/laravel-8-install-bootstrap-example-tutorialexample.html>

At the end of this requirement, you should have an empty Laravel application styled using Bootstrap 4 and have the Login and Register links available in the upper right corner of your application.

1. **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](mailto:jane@example.com), password: [provide a hashed password]
* name: Bob Moderator, email: [bob@example.com](mailto:bob@example.com), password: [provide a hashed password]
* name: Susan ThemeAdmin, email: [susan@example.com](mailto: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 you’re your Phase 1 video demonstration.

1. **Create Required pages to Manage Users**

Using the following article as a guide: <https://www.itsolutionstuff.com/post/laravel-8-user-roles-and-permissions-tutorialexample.html>, create a page to list users in your application. Use the ‘Users Management’ screenshot as an example.

Note: You are NOT being asked to perform the tutorial in the above provided link. The article is provided solely as a guide to accomplish your actions. Some of the steps you have already completed as part of the previous requirements.

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.
   1. Provide the pages to Create and Update users.
   2. 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/8.x/eloquent#soft-deleting>)for more information on how to perform 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.

# Instructions

1. **Don’t forget that a code review is part of this project. You will need to show your code to the instructor as part or your submission either in person or via a screen recording that you will provide. You will need to explain how the code works and complete the code review part of the rubric.**
2. ***Late submissions will be subject to the late penalties laid out in the course outline.***