INET 2005 WEB APPLICATION DEVELOPMENT I

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

# Prerequisites

### Required: Completion of Phases 1 and 2. (3 optional)

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

# Summary

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

Building on the requirements you implemented in Phases 1-2, you will add more functionality to your feed site.

# Phase 4 Requirements – Site Enhancement – Theme Management and Display

1. **Modification of Database with new Migration**

A new modified ERD (in Brightspace) has added a new table to the database layout called *themes*. The new *themes* table will store information about a list of themes that will be available on the site.

* Generate the new migration for the themes table along with the accompanying Model and Controller. -cmr
* Seed the new table with a selection of themes of your choosing from the Bootswatch CDN list of themes. <https://www.bootstrapcdn.com/bootswatch/?theme=4>

Once you have completed creating and seeding your new table based on the ERD, use MySQL 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 4 video demonstration.

1. **Create Required Pages to Manage Themes**

This theme management admin page should do the following:

1. List all themes in the application.
2. Displays a button labeled “Create New Theme”
3. Provides actions to Show, Edit and Delete themes.
   1. Provide the pages to Show, Create and Update themes.
   2. Deleting themes should be accomplished via a “Soft Delete”. (Similar to deleting Users)
   3. All fields in all forms should present appropriate validation.

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

* A themes admin page that lists all users and provides buttons to perform required actions on users. (Show, Edit, Delete)
* A create theme page that provides fields for creating a new theme.
* An edit page that allows modifications to an existing theme.
* The capability to delete a theme via Soft Deleting
* All applicable pages should provide basic validation functionality to maintain valid data.

**Graphical user interface, application, table

Description automatically generated**

**Themes list page**

**Graphical user interface, application, Teams

Description automatically generated**

**Themes details page (show)**

**Graphical user interface, application

Description automatically generated**

**Create new Theme page (create)**

**Graphical user interface, application, Teams

Description automatically generated**

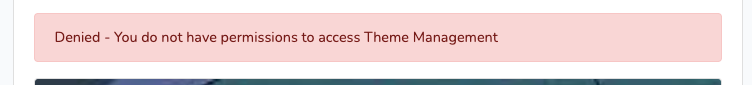
**Edit Theme page (edit)**

1. **Add necessary Middleware to Prevent unauthorized access to Theme Administration Pages**

Create and implement a Laravel Middleware which checks whether or not the currently authenticated user has been assigned the *Theme Manager* 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:



1. **Display Administration Links in NavBar**

For users who have been assigned the Theme Manager 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 Theme Manager’s admin page. This link should only be displayed for Theme Managers.

Example:

Text

Description automatically generated

1. **Add Theme Switching Functionality**

All users will be presented with a dropdown in the Navigation Bar of the application. The list is dynamically filled with data from the themes table.

|  |  |
| --- | --- |
| Graphical user interface, text, application  Description automatically generated | Graphical user interface, text, application, chat or text message  Description automatically generated |

When a user selects a theme, the current page is reloaded with the selected Bootstrap theme applied. All subsequent pages will apply the selected theme until another theme is selected from the dropdown list.

* Selecting a theme from the dropdown causes the id of the selected theme to be stored in a cookie. This cookie is subsequently read by every page load in the application and the current theme is loaded into the page.
* Users should not have to be logged in in order to change their theme.

Graphical user interface

Description automatically generated

*A page with the currently applied theme (Superhero)*

**Some resources to help with the final requirement**

1. HTML Dropdowns
   1. <https://www.w3schools.com/tags/tag_select.asp>
   2. <https://www.htmlgoodies.com/beyond/javascript/article.php/3902606/How-To-Use-JavaScript-To-Auto-Submit-Dropdowns-With-No-Submit-Button.htm>
2. Laravel – View Composers
   1. <https://laravel.com/docs/8.x/views#view-composers>
   2. <https://vegibit.com/how-to-use-view-composers/>
3. Laravel – Redirecting
   1. <https://laravel.com/docs/8.x/redirects>
   2. <https://laraveldaily.com/all-about-redirects-in-laravel-5/>
4. Setting and retrieving and deleting cookies
   1. <https://laravel.com/docs/8.x/requests#cookies>
   2. <https://coderwall.com/p/3etfgw/deleting-cookies-in-laravel>

# 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 with 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.***