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**.

**A significant value-added innovation** will be included on top of the core requirements. This innovation must be submitted to the instructor as a **proposal for approval.** Good security practices will be followed in the construction of the application. The application will be built using proper design practices to separate presentation code from business logic, and database manipulation code. The database design will be documented in a properly-formed ERD.

# Core Requirements – Site Back-End/Administrative Area (28%)

1. **User Login and Registration (4%)**

A login screen that prompts for the username and password will be presented when a user tries to access any administrative area of the website. While logged in, a logout button will be presented on every page that will log a user out from their session when clicked.

Requirement notes

* Only users defined in the Custom Feed Site database as **User Administrator** or **Theme Manager** users will be able to access the back-end administration area

1. **Send Email (Welcome Email/Email Recovery) (4%)**

Users should receive appropriate email messages when:

* Successfully Registering on the site as a new user.
* Resetting their Password.

Requirement Notes

* You employ SMTP on your local server or use an online Email Service to facilitate the sending of emails.

1. **Admin - Managing Users (4%)**

New users can be registered in the system by registering on the front-facing area of the site and they will be stored in the database with their **name**, **email** (login credential), and **password** all stored . In addition, fields for **created\_by**, **last\_modified\_by**, **created\_at**, and **updated\_at** will be maintained.

All user fields (except the REQ-0018 automatically updated ones) will be editable within the application. Input validation will ensure that no blank fields can be added and that the username defined is a valid email address.

Appropriate Input validation will ensure that all required fields are checked and any formatted fields (such as email, url, etc) will be validated.

Requirement Notes

* Only users with **User Administrator** role can manage user accounts.
* By default, you should only display users that already have one or more administrative roles assigned to them. Users with no roles assigned should not be initially listed.
* Users should be deleted using a Soft Delete and a record of who deleted the user should be recorded. This will include adding fields to the Posts table to handle when the post was Soft Deleted and who performed the Soft Delete (**deleted\_at** and **deleted\_by**). Refer to the Laravel Docs for more about soft delete.
* There should be a root site administrator account with at least **User Administrator** role assigned which **cannot be deleted**.

1. **Admin - Manage Users - Search (6%)**

When managing users, it will be necessary to locate new potential admin users by performing a search. In addition to the users list, an input field will be provided on the Manage Users page which will allow a search for users within the entire userbase. Search should be performed on the **name** and **email** fields in the users table with all results displayed as an updated list.

Requirement Notes

* Only users with **User Administrator** role can perform search for users.
* Results of the search should include all users from the user base, including those who have not been assigned an administrative role.

1. **Admin - Assigning User Roles (6%)**

User roles will be assigned in the back-end administration area by **User Administrators**.

There will be three levels of user privileges for interacting with the site: **User Administrator**, **Post Moderator**, and **Theme Manager**. These should be defined in a lookup table in your database**.**

**Roles are not inherited or contained within other higher role sets, which means that, for example, the User Administrator role does not automatically confer both Post Moderator and Theme Manager roles**. For example, if the same user needs to manage both themes and users in the back-end, then you would give them both **User Administrator** and **Theme Manager roles**.

Thus, when managing a user’s administrative permissions, they can be assigned multiple permissions (or none at all if you wish to make them simply an end-user on the site).

Requirement Notes

* Only users with **User Administrator** role can manage user administrative privilege assignments.

1. **Admin - Managing Site Themes (4%)**

Multiple Bootstrap themes can be defined in the administration area which can be applied to style the entire Social Media Site. Themes will be stored in the database with their **name**, **cdn\_url**, **description,** and **is\_default** all stored appropriately. In addition, fields for **created\_by**, **last\_modified\_by**, **created\_at**, and **updated\_at** will be maintained.

All applicable site fields will be editable within the administration area. Site Themes CAN be deleted ***except*** when the theme is currently marked as the default theme. Default status must be assigned to another theme before the theme in question can be deleted.

Appropriate Input validation will ensure that all required fields are checked and any formatted fields (such as email, url, etc) will be validated.

Requirement Notes

* Only users with **Theme Manager** role users can manage themes.
* The theme must be associated with the user who created it.
* The theme must be associated with the user who last modified it.
* Site Themes CAN be deleted ***except*** when the theme is currently marked as the default theme.

# Core Requirements – Front-end Feed Area of Site (28%)

1. **Display of Public Feed Site Page (8%)**

End-users navigating to 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. The posts should be **listed in descending order** based on the date of posting (ie. in reverse chronological order). The display of the current feed will be in a **main content area** of the page. A **sidebar area** for the page should also be present which will house a **button** which the end-user can click to **create a new post** for the feed. The sidebar should also contain the **input to change the theme** that the user wishes to use for the site.

Requirement Notes

* End-users do not have to be logged in to have access to the main feed page. (ie, they can view the feed as guest). They can scroll through and view any posts that have been submitted to the feed.
* Users cannot post to the feed via a button until they are registered users and are logged in to the site.
* The posts in the feed should be displayed in an informative manner. (ie. include any relevant information which may be deemed informative to the end-user such as when the post was made and by whom)

1. **Create new Feed Post (4%)**

When a logged-in user clicks on the button to create a new post, they will be taken to a form in which they can create a new post. THE CONTENT TYPE OF THE POST CAN BE OF YOUR CHOOSING (ex photo, video, tweet, etc). The post content however, must consist of at least three fields, one of which must be a **title**, and one of which must contain **the content or a link to the content posted**.In addition, fields for **created\_by**, **last\_modified\_by**, **created\_at**, and **updated\_at** will be maintained.

Appropriate Input validation will ensure that all required fields are checked and any formatted fields (such as email, url, etc) will be validated.

Requirement Notes

* The saved post must be associated with the user who posted it.

1. **Apply/Switch Site Styles (6 %)**

The site should be visually appealing to the end-user, so the site will implement **Bootstrap** as the foundational styling framework for the site. In addition, the site should provide a selection of Bootstrap Themes (ex **Bootswatch** **Themes**) which the user can apply by selecting from a list.

Requirement Notes

* Information about the currently chosen theme should not be stored in the database. Instead it should be stored in a site cookie which is read by every page request. That way the end-user's style will remain valid for subsequent visits to the site
* If the end-user is visiting the site for the first time or a previous cookie has been deleted and there is no chosen theme set in a cookie, the site should apply whatever default theme has been currently assigned by a **Theme Manager**.

1. **Moderating Posts (4%)**

For assigned Post Moderators, each posted item in the feed should also include a 'Delete' button. Which when clicked, prompts the Post Moderator to delete the post. If the **Post Moderator** confirms the delete, post is deleted from the feed.

Requirement Notes

* Deleting of posts by Post Moderators should be a **Soft Delete** and a record of who deleted the post should be recorded. This will include adding fields to the Posts table to handle *when* the post was Soft Deleted and *who* performed the Soft Delete (**deleted\_at** and **deleted\_by**). Refer to the Laravel Documentation regarding Soft Deletes.
* Only users assigned the **Post Moderator** role should be presented with the Delete button and should be permitted to perform the delete.

1. **Feed Auto-Update/Polling (6%)**

The main feed page, if left idle, should continuously poll the database for any new posts submitted by other users and subsequently update the feed in real time. This should be accomplished via some **AJAX** mechanism and use **Short Polling**.

Requirement Notes

* If you research a method and implement it, be sure to cite all sources and be prepared to explain, ­in detail, how the functionality works.

# Supplemental Requirements (20%)

1. **Propose (5%) and Implement (15%) Additional Custom Feature**

In addition to the core requirements listed in this document, you will also implement an additional piece of functionality that adds **SIGNIFICANT VALUE** to the existing feed site.

This functionality must be proposed and approved by your instructor by a deadline date set by the instructor. **The instructor will inform you as to how to submit this proposal and the instructor retains the final say as to whether or not your proposal is adequate to meet the requirement.**

The additional piece of functionality should **require some type of database addition/modification**.

**THE IMPLEMENTATION WILL NOT BE MARKED WITHOUT PRIOR APPROVAL FROM YOUR INSTRUCTOR ON THE REQUIRED DATE!**

# Security Requirements (10 %)

1. **Site-Wide Limiting of Access to Authorized Users (8%)**

Only users with authorized roles will be able access the back-end site’s pages and exercise their particular functionality defined for the back-end. All users will be able to access the front-end site, but only authorized Post Moderator users will be able to delete posts on the front-end.

Once logged in to either site, each page will have a Logout button, which will log the user out appropriately.

If the user has **User Administrator** or **Theme Manager** roles assigned, additional admin options should be listed in the dropdown in addition to the logout option.

Requirement Notes

* **User Administrators** – can access User Management, change users
* **Theme Managers** – can create update and delete themes.
* **Post Moderators** – can delete posts on the feed

1. **Employing Least Privilege (2%)**

All database queries will be executed on the **MySQL** database using credentials that have the minimum level of permissions possible on the database while still supporting full application functionality.

# Documentation & Database Requirements (10%)

1. **Entity Relationship Diagram for SM Feed Site Database (4%)**

A final Entity Relationship Diagram (ERD) document **will be created and presented during code review** for the MySQL database to support the custom web application. The document will use Crow’s Feet Notation. All entities and relationships will be reflected in the document. All relevant fields, data types, and keys will be appropriately indicated.

1. **MySQL and SQLite Database Implementation (6%)**

The system will be deployable on both SQLite and MySQL databases. Any and all database migrations along with any seeding of data will work on both systems.

# Development Process Requirements (4 %)

1. **Using Source Control (4%)**

The code for this project will be fully integrated with Source Control. The project will be added to a cloud-based Source Control repository from the beginning of development and the addition of every major feature (i.e. the completion of each general requirement) will be reflected in the code history. The project manager (ie: Your Instructor) will be added as a read-only participant to the repository.

# Instructions

1. **Don’t forget that a code review is part of this project. You will need to show your code to the instructor in class on or before the due date while going through an evaluation of the site’s functionality. You will need to explain how the code works and complete the code review part of the rubric. Note the group caveat marked in blue in the code review section of the rubric.**
2. ***Late submissions will be subject to the late penalties laid out in the course outline.***

**Question about ERD**

1. Do I need a Update Record for Post, User or Theme?