Assignment 5 - Routes, EJS, and MVC - Secure Application

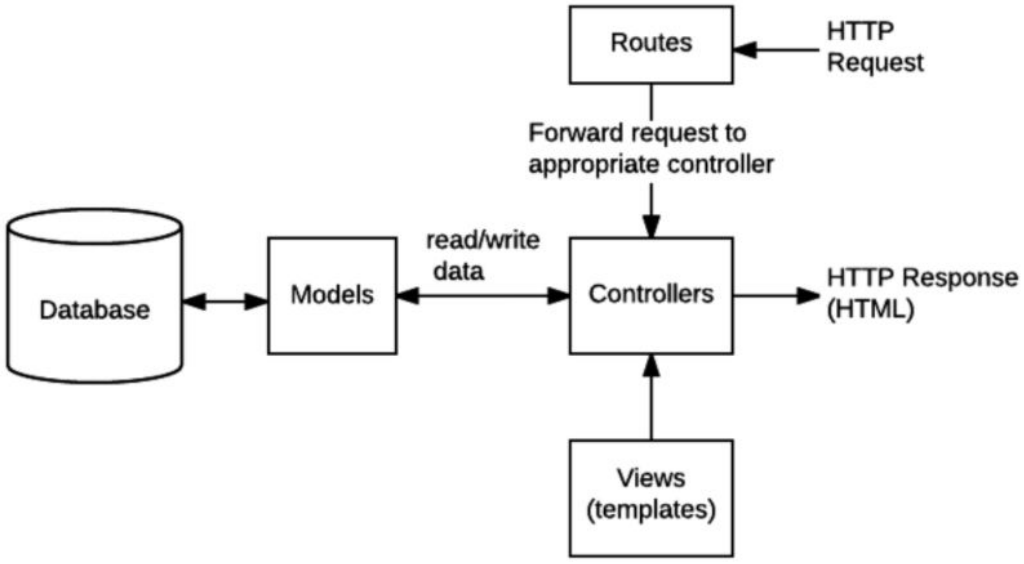
This assignment is to be completed individually. No group work is allowed.

This assignment is intended to implementing security measures to enhance the application robustness to security attacks.

Assignment Description:

This assignment is intended for implementing security measures to enhance the application robustness to security attacks

1. Correct any errors identified or partial/missing functionality from the previous assignment. If your application developed in assignment 4 has error and not complete it is your responsibility to communicate and coordinate with course staff to help fix those bugs. Keep in mind that you are building in this application in stages. It is important that you have a working application for each stage.
2. All structure, design, and content requirements from previous assignments are mandatory, unless explicitly updated in this assignment description.
3. Use JavaScript to implement the business layer of the application (**model**).
4. Use EJS pages to present the **view** to the browser.
5. Use routes to **control** the flow of the application.
6. Functionality that does not follow the assignment specifications will not receive credit.



Add Logic to Handle User Login

- When the user clicks the sign in button this should send a request to verify that username and password submitted belong to an already registered user.
- If the user credentials do not validate, the login view should be displayed showing an error message. (e.g. Either username or password are incorrect. Please try again.)
- If the user credentials are verified display the profile view.
- **Note** that this might require updating your models and database to account for username and password. Your choice on how and where to handle representing these properties in the models and the database (e.g., adding to existing models and documents or creating new models/documents specifically for login info)

Pseudocode:

- If the action parameter validates to a value of “signIn”:
    - Check the session for “theUser” attribute
    - If a “theUser” attribute is set dispatch to **profile** view (savedConnections.ejs).
    - If there is no valid User object stored in the session

- Checks the http request for a parameter called “username”
- If there is no username parameter, or it has an unknown value dispatch to the login view.
- If there is a username parameter and is valid check the http request for a parameter called “password”
- If there is no password parameter, or it has an unknown value dispatch to the login view.
- If there is a valid username and password, verify the username and password matches a valid user in the database.
- If the user credentials are valid retrieve the list of saved connections for that user from the database. If a user doesn't have any saved connections this should be empty.
- Add the list of saved connections to the session object as “currentProfile”.
- Dispatch to the **profile** view

Applying Security Measures

- Update your application to make use of the express-validator package to validate all input values read into the application (login and add connection)
- Update your application to escape dynamic output displayed in all views

Database Creation

In addition to the standard project folder for this assignment, you will submit a text file (plain text, NOT Word or RTF or PDF or any other fancy document format). This file will contain all of the MongoDB queries that you use to (1) create, and (2) populate your database. Your database script file must be called:

milestone5_create_db.txt

What to submit using Canvas (Email submissions will NOT be accepted):

1. **Milestone5.zip** - An archive of the entire web application (project) stored in a standard ZIP File, you must ensure that all the files are included as part of the archive.
2. **Milestone5_create_db.txt** – text file to create and populate your database tables.
3. **Milestone5Info.pdf** – PDF document with the following assignment information:

1. Screenshots of all views of your application displayed in your browser
2. Explanation of additional features, if any.
3. Explanation of status, stopping point, and issues if incomplete.
4. Discuss the easy and challenging parts of the assignment. How did you overcome all or some of the challenges?