Web Application Project III

INET2005 Web Application Development I

Evaluation : 30% of Final Grade

Assignment Description

This final project is to develop a link home page to quickly display all your important links for easy access. The Web application includes a public end for displaying all the links and an administration end for modifying the link content.

The requirements are:

- It is best to break up this web application into several controllers, models, and views within the single project folder. Three controllers are required: one for public, admin, and login. For this project, models should be shared. Views will be specific for their controllers.
- All database interaction for the public / admin end is to be implemented using the Entity Framework Core (EF Core). This will make the amount of code needed *much* smaller than otherwise. However - the login functionality *does not* need to use EF Core.
- All link and category data on the website is to be fed from a *relational* MySQL database of your own design running in its own docker container.

- PUBLIC END
  The public end must meet the following requirements:
  - The public end is made up of four categories of links displayed in four columns
  - Each column has a category title at the top
  - Each column is made up of several links, each with their favicon displayed. This can be done by using the following in the src attribute of the icon's <img>:
    `https://www.google.com/s2/favicons?domain_url=[THE URL OF YOUR LINK]`
  - The links in the column are in alphabetical order
  - At the top of each column there are a handful of links that are above a line (or equivalent). These links are "pinned" to the top of the list are also in alphabetical order. If no links are pinned then no line is displayed
  - Clicking a link should open the website in a new browser window (tab)
  - The web page must be responsive using bootstrap

- ADMINISTRATION END
  The administration end must meet the following requirements:
  - The admin end must be accessed through a login that uses a hashed and salted password stored in a database table. There should be no security holes!
  - After 18 minutes of no post-backs, a warning message must be displayed to the user that they are about to be logged out – and then at 20 minutes the user must be automatically logged out.
  - The admin end page must look the same as the public end with the inclusion of:
    - Add Link and Edit Category buttons beside each category title (top of each column)
    - Edit Link and Delete Link buttons beside each link
  - When the user clicks on an Edit Category button, a form is displayed to change the category name
  - When the user clicks on an Add Link button, a form is displayed to enter the label, URL, pinned option of the new link. The category is displayed but is not changeable since it is already determined by which Add Link button was clicked.
  - When the user clicks on an Edit Link button a form is displayed to change the link's existing category (from a dropdown), label, URL, and pinned option.
  - When the user clicks on a Delete Link button a form is displayed to delete the link or not by clicking a button

    - All forms must include a cancel button to take you back to the main admin page
    - All forms must be error checked using Data Annotations:
        - All fields are required
        - Link URLs should be checked that they are a properly formatted URL
    - The administration end must be user friendly using bootstrap
    - At the top of main admin page you must include a button / link that logs out the user *and* takes him/her back to the public end

- Comments must be included throughout your code to explain your logic
- To aid in code review testing, please have one of the categories contain only three links
- See screen samples for reference

Requirements (Marks breakdown)

Test Cases for Public End	
Application test cases *A series of test cases that test all public end requirements listed above*	6
Test Cases for Admin End	
Application test cases *A series of test cases that test all admin end requirements listed above*	19
Code Design and Publishing	
Relational Database Application Design and Structure *Model(s) class design, controller, views (razor), etc.* Commenting Git commits	1 2 1 1
TOTAL MARK	30

Other Notes

- This project will be marked during a code review on the due date. Be sure to bring your entire working project folder to the lab