CSCI 2170: ASSIGNMENT 3

Due date: March 12
Due time: 11:30pm Halifax time

EXPECTATIONS: WHAT'S OKAY & WHAT'S NOT

This is an ***individual assignment***.

- Work on it by yourself.
- Help each other in the class and/or in your groups – **do not share solutions**.
  - Restrict your help to **conceptual discussions only**.
- ***If you receive help***, include a citation in your code AND in readme.md in your submission folder with the following information:
  - **Help received from:** person, URL of website, etc.
  - **Date help received:** DD-MMM-YYYY
  - **Nature of help received:** one-two line description of the nature of the help received.

Give credit where credit is due.

This assignment tests your individual learning.

- It will be tested for plagiarism, inappropriate collaboration and violation of academic integrity principles.

Start your work early and budget your time.

- It helps you work on your assignments in a healthy manner.

If you have any questions or concerns, please email raghav@cs.dal.ca – at least 48 hours BEFORE the deadline

EXPECTATIONS: WHAT'S OKAY & WHAT'S NOT (PAGE 2)

How can your TAs help?

- Remember, this is an assignment.

  - We are assessing how you can apply what you have learned in the context that is given in the assignment.

- Your TAs can guide you in your work, but…

  - They will not give you the answer.

  - They will not be able to review your work and tell you whether it is correct or not.

  - They will not be able to review your work in the last minute, i.e., a few minutes or hours before the deadline – remember that they are students taking courses too.

  - **They cannot give you extensions** – if you want an extension for any reason, you must contact Raghav (raghav@cs.dal.ca).

OVERVIEW OF ASSIGNMENT 3 (A3)

Main goal: *The main goal of this assignment is for you to add additional features to a simple blog website.*

- This assignment tests your knowledge on form processing and database use in PHP.
- This assignment also requires knowledge in aspects of front-end development (HTML/CSS/JS).

User story for the search website:

As a blog user,

I would like to access all features of the blog,

So that I can login, read and write blog articles.

Reference video for functionality demo:

You can refer to the reference video to understand the expectations (explained in the next few slides) and to see how the search results must look and work.

NOTE

- Starter code will be available on Gitlab on Friday, February 19.

- Follow the instructions in the section named **[7] Gitlab** to configure your Gitlab to work on assignments in this course.

  - If you are unable to find the starter code on Gitlab, please email **cshelp@cs.dal.ca** immediately – folks at the CS Help Desk will need to give you access.

A3: DETAILED REQUIREMENTS (PAGE 1)

Task 1: Clone repo from Git and connect to the database

- Use **db.php** in your codebase to connect to the database.

- Verify that you can connect to the database.

Task 2: Load data into the database

- Download the SQL files from **A3-SQL.zip** for details and information about the DB tables you will use in this assignment.

- In your database setup, create a database named **2170db**.

- Import the tables named **jedilogin** and **jediblog**.

A3: DETAILED REQUIREMENTS (PAGE 2)

Task 3: Implement blog data retrieval, display and search

1. Read data from DB and simply display all blog data (from the table **jediblog**) into the **homepage** below the search form and implement search using author name and blog title.

   - Remember to limit the article content to 200 characters on the homepage, with a link to the full article in the "Read more" link (Task 4).

   - You have already implemented a part of this requirement in Assignment 2. You can re-use the code **with citation in both the PHP script AND readme.md**.

   - "*How do I cite Assignment 2?*"

     - Cite it as follows:

```
/*
This code to implement blog data retrieval, display and search has been used with some
modification from my submission for Assignment 2 in CSCI 2170 (Winter 2021).

<Your Full Name>, Assignment 2: CSCI 2170 (Winter 2021), Faculty of Computer Science,
Dalhousie University. Available online on Gitlab at [URL]:
<include_link_to_your_gitlab_A2_submission_here>.
Date accessed: <include_date_when_you_used_this_code_in_your_assignment_here>.
*/
```

A3: DETAILED REQUIREMENTS (PAGE 3)

Task 4: Implement blog content display, linked from homepage

1. Implement the functionality to link the "Read more" link on the homepage to display the full article in **post.php**

   - Refer to the reference video to see a demonstration of how this works.

   - Why is this important?
     - This is one way of passing data between pages in a website.
     - It is a great way to learn this technique so that you may be able to implement it in different applications.

A3: DETAILED REQUIREMENTS (PAGE 4)

Task 5: Implement the functionality to submit a blog post

1. The form to submit a blog post is provided in **submit-blog.php**

   - Implement the form processing script in **process-blog.php**

   - Remember to sanitize the data before you submit it to the table **jediblog**

   - You have already implemented a part of this requirement as an email submission form in Assignment 1. You can re-use the code **with citation in both the PHP script AND readme.md.**

   - "*How do I cite Assignment 1?*"

     - Cite it as follows:

```
/*
This code to implement processing form submission has been used with some
modification from my submission for Assignment 1 in CSCI 2170 (Winter 2021).

<Your Full Name>, Assignment 1: CSCI 2170 (Winter 2021), Faculty of Computer
Science, Dalhousie University. Available online on Gitlab at [URL]:
<include_link_to_your_gitlab_A1_submission_here>.
Date accessed: <include_date_when_you_used_this_code_in_your_assignment_here>.
*/
```

A3: DETAILED REQUIREMENTS (PAGE 5)

Task 5: Implement the functionality to submit a blog post (Cont'd)

2. In **submit-blog.php**, implement a functionality to check if the user wants to cancel and return to the homepage.

   - Give the user an option:
     - If they say "cancel" in the option, they should be able to stay on **submit-blog.php** and continue editing the blog post.
     - If they say "ok" in the option, they should be taken to the **homepage** and the blog entry must not be saved in the database.
   - See reference video for a demonstration of how this functionality works.

A3: DETAILED REQUIREMENTS (PAGE 6)

Task 6: Implement login and logout functionality

1. Implement the functionality to login and logout+redirect.

   - The code to logout has been provided, but you will need to appropriately redirect the user.
     - See reference video for a demonstration of how it works.
   - After the user logs in, make sure to regenerate the session ID and delete the old session.
     - There is a standard way to implement this as explained in the class and on PHP.net.
     - Implement the functionality and cite the source.
   - Use the table **jedilogin** to store and verify login information.
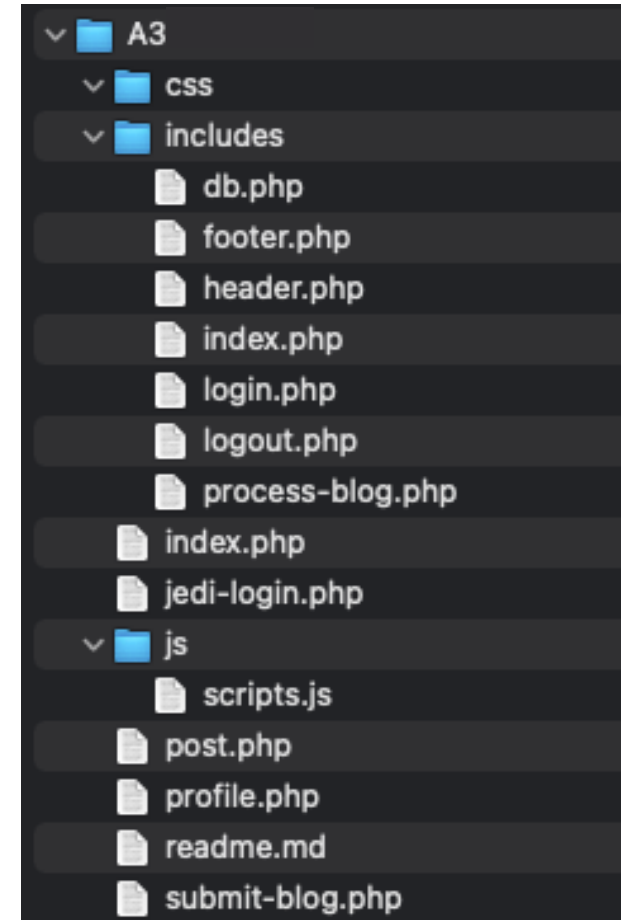
A3: DETAILED REQUIREMENTS (PAGE 7)

Task 7: Implement access control & simple profile page

1. It is important to start implementing security in your work from now.

2. Implement access control in your website as follows

   - If the user is **not** logged in,
     - They must not have access to all options in the navigation bar
     - They must not have access to the **includes** folder
     - They must not have access to any file in the **includes** folder
   - If the user is logged in,
     - The navigation menu changes with options to submit a blog, view profile and logout
   - See reference video to see how access control works

3. Implement a profile page (**profile.php**)

   - This is a simple profile page.
   - Display the user's full name as a paragraph and their username and password in a disabled form below the full name.
   - See reference video to see how the profile page works

A3: DELIVERABLE & SUBMISSION

Format your assignment deliverable as follows:

1. Use the folder that you use the folder named **A3** from Gitlab for all your work in this assignment.

   ▪ Any style or script information must be placed in **css** and **js** folders.

2. Organize **A3** as shown in the figure on the right →

3. Include details as specified in **readme.md**:

   ▪ Remember to include all citations to material studied / learned outside class and textbook and/or help received.

   ▪ If you use any images and they are your own work, include a note to indicate so.

   ▪ Include a citation for A2 and A1 work re-used in A3.

4. **Submission**

   ▪ Submit your code on **Gitlab** by pushing changes to the repository.

     ▪ Verify that your code is available on your folder on Gitlab.

   ▪ Submit your code on **Brightspace**:

     ▪ Download the folder as a ".zip" compressed folder, i.e., **A3.zip** from Gitlab (you can download from https://git.cs.dal.ca)

     ▪ Submit this downloaded folder (i.e., A3.zip) on Brightspace in the assignment submission module.

     ▪ <mark>If you submit in any format other than **.zip**, the assignment will not be graded. Sorry.</mark>

```
∨ 📁 A3
  ∨ 📁 css
  ∨ 📁 includes
        📄 db.php
        📄 footer.php
        📄 header.php
        📄 index.php
        📄 login.php
        📄 logout.php
        📄 process-blog.php
     📄 index.php
     📄 jedi-login.php
  ∨ 📁 js
        📄 scripts.js
     📄 post.php
     📄 profile.php
     📄 readme.md
     📄 submit-blog.php
```

A3: SUBMISSION POLICY

Please note the submission policy:

1. Submit what you have completed by the deadline: **March 12** (11:30pm Halifax time).

2. Remember – you are submitting **on Brightspace** AND **on Gitlab.**

3. No extensions will be provided.

   - Please submit what you complete by the deadline.

4. Email submissions or submissions through Teams will not be accepted.

A3: MARKING (PAGE 1)

This assignment will be marked with letter grades as per Dalhousie's letter grading scheme.

Grading will be based on the following components (details & rubric in the next few pages)

➢ **C1: Functionality, UI & interaction (65%)**

- Graded by executing code submitted on Gitlab and viewing the web page.
- Refer to rubric items C1-1 … C1-5 for details on marking. Implementation expectations are provided earlier in this document.

➢ **C2: Code inspection (20%)**

- Graded by reading code and giving feedback on functionality implementation. (20%)
- Include comments and organize code in a readable manner. (5%)

➢ **C3: Code organization & folder structure (5%)**

➢ **C4: Git submission (5%)**

➢ **C5: Brightspace submission (5%)**

Grade will be computed as follows:

A3 grade = C1-1 (5%) + C1-2 (10%) + C1-3 (7.5%) + C1-4 (7.5%) + C1-5 (5%) + C1-6 (25%) + C1-7 (5%)

+ C2 (20%) + C3 (5%) + C4 (5%) + C5 (5%)

A3: MARKING (PAGE 2)

Expectation	Exceeds expectations (A range)	Demonstrates clarity in concepts (B range)	Meets expectations (C range)	Does not meet expectation yet (F)
C1-1: Homepage and blog search displayed as shown in reference video. Article contents limited to 200 characters	A+: Blogs on the homepage and search results are styled to appear as shown in the reference video. Demonstrates excellent understanding of applying front-end web design concepts.	B: Demonstrates good understanding of front-end web page design. Content may not appear exactly as shown in the reference.	N/A	F: work does not meet expectations yet
C1-2: Link to and display each article in full on **post.php** as shown in reference video	A+: Article correctly displayed in full on post.php as expected.	N/A	C: PHP code to retrieve and display content implemented, but post not displayed correctly.	F: work does not meet expectations yet
C1-3: Functionality to submit blog post implemented as expected	A-: Only blog post submission implemented, and data is stored correctly in DB and user is redirected to the homepage and the new article is displayed in the homepage. A+: User is able to confirm (JS) whether they want to stay on the page or cancel and return to homepage.	N/A	C: Blog post submission implemented but is not complete. Does not work reliably.	F: work does not meet expectations yet
C1-4: Login functionality implemented as expected	A+: User is able to login and all files on the website maintain and are able to access session data correctly.	N/A	N/A	F: work does not meet expectations yet

A3: MARKING (PAGE 3)

Expectation	Exceeds expectations (A range)	Demonstrates clarity in concepts (B range)	Meets expectations (C range)	Does not meet expectation yet (F)
C1-5: User is correctly redirected after they are logged out.	**A+**: Redirection is correctly implemented as in reference video.	N/A	N/A	**F**: work does not meet expectations yet
C1-6: Access control implemented as specified and demonstrated in reference video.	**A+**: All access control features as expected are implemented correctly.	**B**: Some access control features as expected are implemented, and some may not be fully implemented.	**C**: Minimal implementation of access control, some sections of the website does not have any access control.	**F**: work does not meet expectations yet
C1-7: User profile page implemented as expected and demonstrated in reference video.	**A+**: Profile page implemented correctly as expected.	N/A	N/A	**F**: work does not meet expectations yet
Include citations as required and follow academic integrity rules.	*If no citations are needed, include a note in readme.md saying that you did not refer to any external sources for this assignment.*			**Report to AIO**: if citations or note are not included 17