COMP2004 – Final Project – Week 14

Due date: Friday, Dec. 13th @11:59 pm

Grade: 40%

AI is not permitted for this project. Using AI or external help will result in a Zero grade and an Academic Integrity report.

Project Description:

Welcome to the final project of Full Stack Development for Fall 2024. For this project, you and your team will continue to finalize the Groceries App that we started in Project One and enhanced in Project 2. This time, we want to develop the project into an alpha version that is ready for deployment. This will require us to enhance the concept of separation of concerns, pagination, authorization, and private routing.

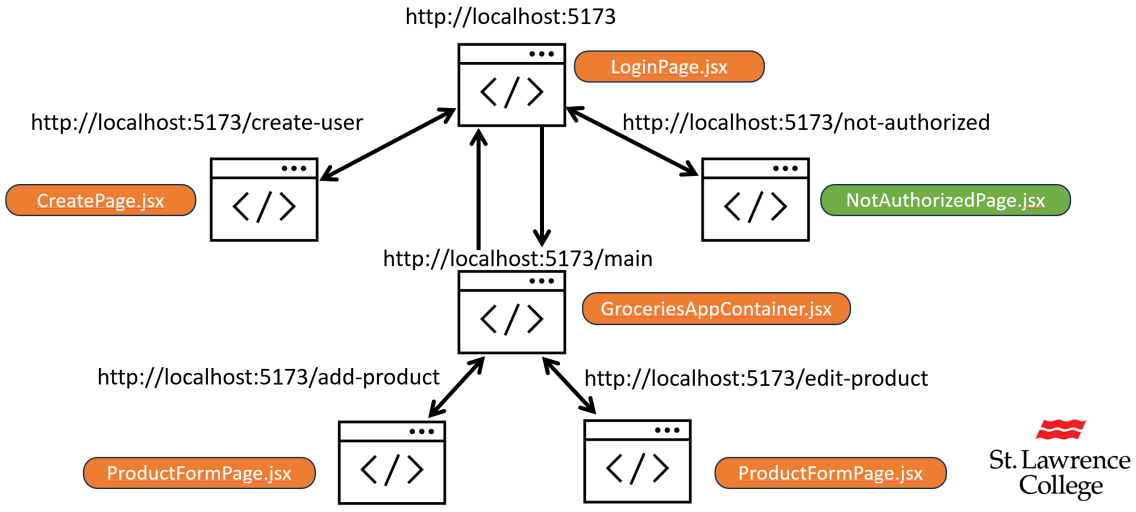
***Note**: You can continue building version 3 on your previous submission for Project 2 or use my starter code provided on Blackboard.*

The alpha version of the app consists of the following:

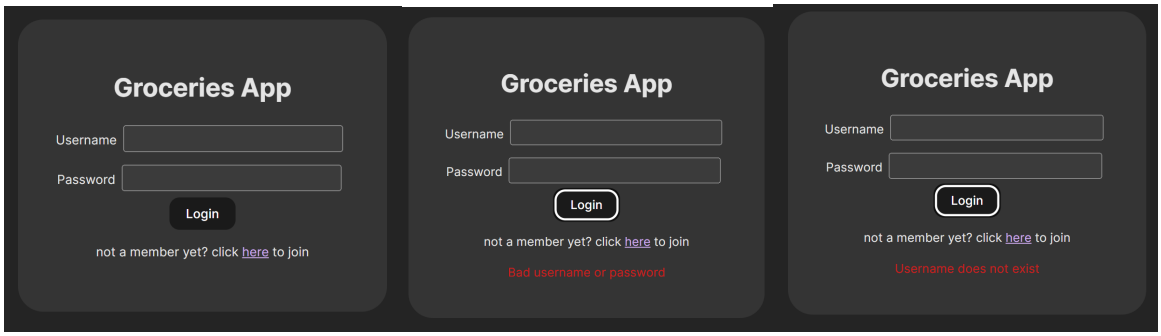
<u>Part A: Separation of concerns and pagination:</u>

In this project, we will utilize this concept we learned at the start of the semester. By separating every component in their file, reusability and refactoring become easier. The following diagram shows the new web application map of pages and routes.

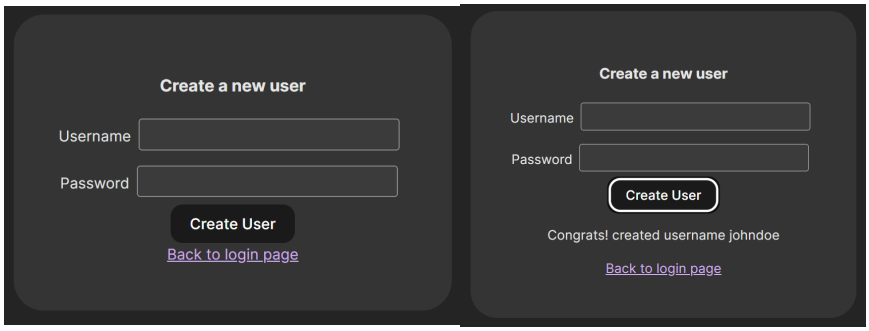
Groceries App Alpha web map



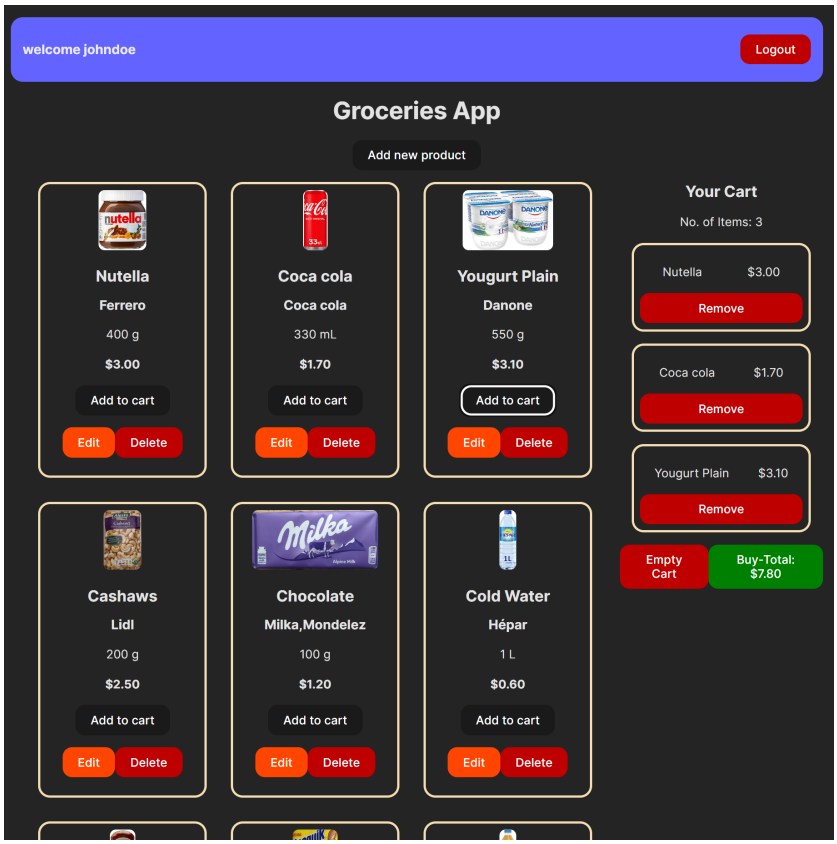
1. **Route (“/”):** The landing page will be a login form expecting a username and password to enter. If the credentials are valid, the page will automatically navigate to Route (“/main”) where the main GroceriesApp exists. If the credentials are invalid, the form should display a message that the username and password are incorrect. If the username does not exist on the database, you should display a message stating that. There is also a link to Route (“/create-user) to create a new user on the database. (Check the below screenshots to learn more.)



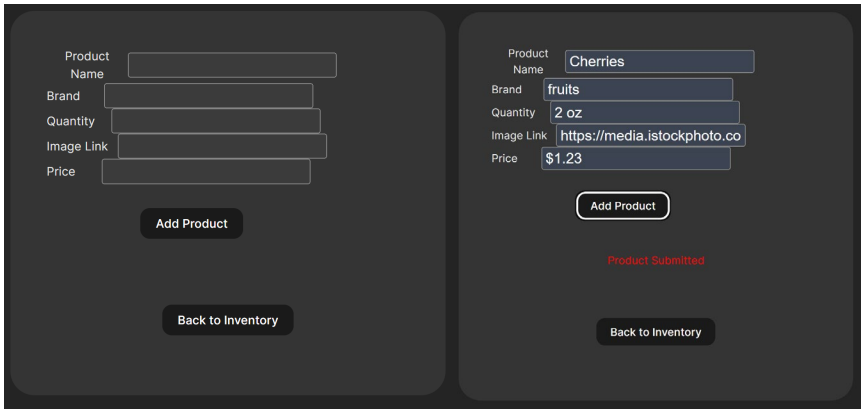
2. **Route (“/create-user”):** If the user is new and wants to join, the user will click on the link on the login page to go to Route(“/create-user). This route will land on a new page that will load the same login component with a different message for the title and button. Once the user is created, the form will display a message confirming the creation. Also, the user must navigate back to the login page on Route (“/”) to log in with the new user. (Check the screenshots below to learn more.)



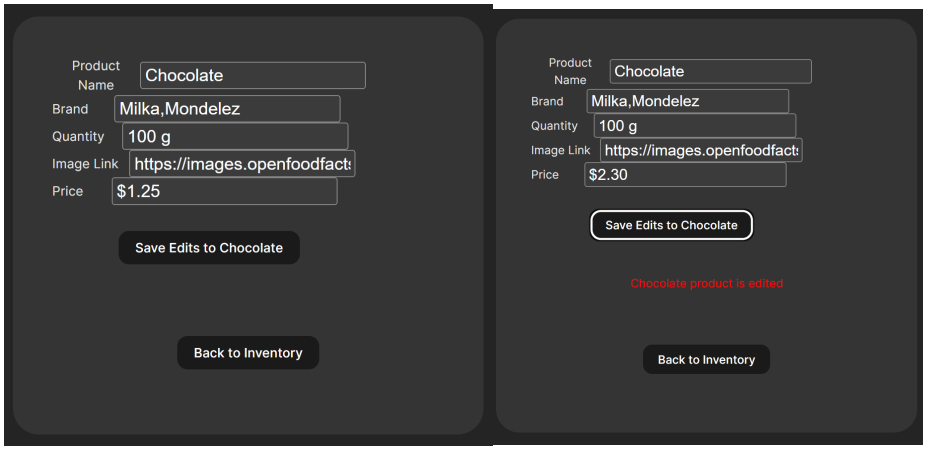
3. **Route (“/main”):** This is the primary app route. Once the username is authenticated, the page will display the InventoryCards component. The page should operate as expected, like the last project. You need to add a new navigation bar at the top of the page containing a welcome message with the currently logged username added to it and a logout button. If the user logs out, the page will navigate to Route(“/”) and back to the login page. Also, the add and edit form that used to be at the app's top will be moved to their pages. A new button will be added to direct to the Route(“/add-product”) page, and the edit buttons should navigate to the Route (“/edit-product”) page. (check the screenshot to learn more.)



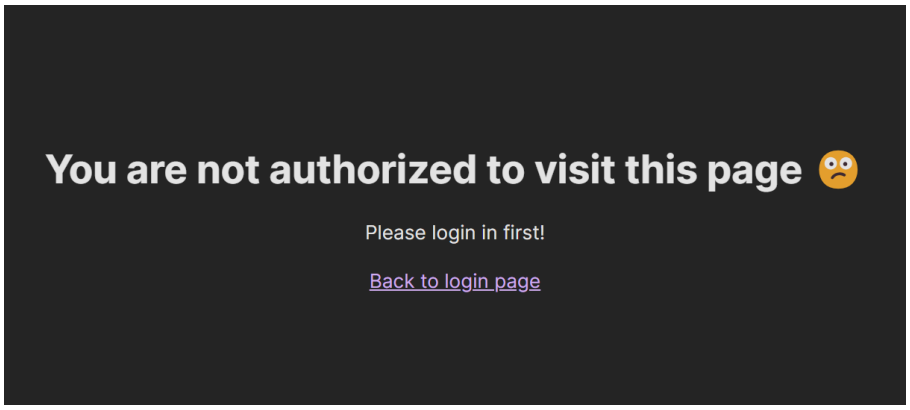
4. **Route (“/add-product”):** This route will display the InventoryForm component. The inputs should be empty to add information about a new product to be added. There will be a button to add a new product and another to return to Route(“/main”). Once a new product is added, a confirming message is displayed. (check the screenshots to learn more; please ignore my styling 😊.)



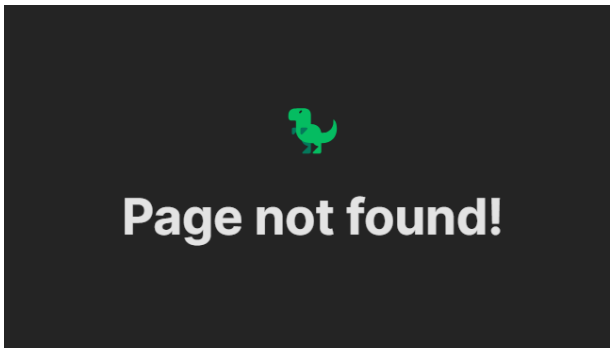
5. **Route (“/edit-product”):** This route will display the InventoryForm component. The inputs should be filled out with the product information that needs to be edited. Also, this form will have a save to inventory button, with the product name on the button, and a back to inventory button. (check the screenshots to learn more; once again, please ignore my styling 😊.)



6. **Route (“/not-authorized”):** This route will display a new page with a message that the page the user wants to reach cannot be found unless the user is logged in and authenticated. To test this, log out of the app and add one of the routes to either the main, add, or edit pages in the address bar. There will be a link to the login page (check the screenshot to learn more).



7. **Route (“*”):** This is any route other than the ones mentioned above. This will display a message that the page you want to reach is not found. (check the screenshot to learn more.)



Part B: Authentication:

Every user needs to be authenticated to use the app. The credentials should be saved in the database, and authentication will occur on the backend server. Once a user is authenticated, a cookie token is created and stored on the browser. This token will confirm authentication to enable routing to the locked pages, which we will call private routing. Once a user logs out, their authentication cookie must be deleted from the browser, and the app's private route should be locked.

Part C: Private Routing:

Routes ("/main," "add-product," and "edit-product") will be considered private routes. As explained above, you need to be authenticated first to access them. Therefore, you will need to design your routing with this in mind.

Submission:

Each team should submit a GitHub link to the project on the submission page on Blackboard.

Grading rubric:

Key Concept	**Extensive Evidence**	**Convincing Evidence**	**Limited Evidence**	**No Evidence**
Functioning Code	The code functions with no errors and passes all test cases. (15%)	The code functions without errors, but not all test cases pass. (8-14%)	The code functions with errors, and some of the tests pass (1 - 7%)	The code is not functional. (0%)
Program Logic and Correctness	The program's output is as expected. (20%)	The program's output produces minor differences than expected. (10-19%)	The program's output produces significant differences than expected. (1 - 9%)	The program does not output. (0%)
Commenting and annotations	The code contains extensive comments and annotations describing the code functions. (3%)	The code contains comments and annotations describing the code functions. (2%)	The code contains some comments and annotations describing the code functions. (1%)	The code is without comments or annotations (0%)
Variable and function naming	The variables and functions' naming are descriptive. (2%)	The variables and functions' naming are somehow descriptive. (1%)		The variables and functions' naming are not descriptive (0%)