

**CS-321: Server-Side Web Development**  
**Spring 2021**  
**Northeastern Illinois University**  
**Homework #8: Persistence, Part II**  
**Due Date: Thursday, 03/25/21 by 6:00 p.m. CDT**

**IMPORTANT:** Every step in this homework assignment refers to materials that was presented in the lecture videos and PowerPoints, so you should be actively using the videos, PowerPoints, and referenced materials.

**Homework Goals:**

1. Add View functionality for a single item.
  2. Add Edit functionality for a single item.
  3. Add Delete functionality for a single item.
  4. Handle validation for all data submissions.
  5. Maintain consistent styling for all pages in your website.
- 

**Part I: View functionality and Uniqueness**

- Is there a particular aspect of your model that should be unique other than the id? If so, make sure to add a column constraint. However, this may not be relevant for your particular model.
- On the page that allows a user to view all the items, modify the list so that each item is represented by a header. Do not show the details for that item.
- For each item, add in a button, link, or icon that will ultimately link to viewing the details for that item.
- Options: You can do one of the following:
  1. The view button/link/icon can use CSS to expand the details in the page.
  2. Create a separate HTML page that allows a user to view and edit specific functionality for an item. This page should have consistent navigation (i.e. it should have the navbar), but should not be a navigation link in the navigation bar.
- If you created a separate page for viewing the details, decide which fields for an item should be editable.
- If you had expandable functionality for viewing the details, you will need an edit icon that will link to an edit form for the item.
- Make sure that you create the corresponding controller method to handle the view request if you created a separate page for viewing the details.

- Run your code and test it. Push to GitHub.
- 

### **Part II: Edit functionality**

- Depending on how you set up Part I, you may need to create a link/icon/button that goes to another HTML page that contains a form for editing a specific item.
  - The other option was to have the view/edit functionality be on the same page - in other words, a user can view their functionality in a form and edit it.
  - The form should be pre-populated with the item information and you need to decide which fields are editable (and why). Don't just make things uneditable because you think it's easier!
  - The form should have a submit button and a cancel button.
  - The cancel button should link back to the view all items page.
  - The submit button should be associated with a controller method that UPSERTs the information into the DB. You must perform validation and handle uniqueness.
  - You should display helpful error messages to the user in the event that the fields are entered incorrectly using both server-side and client-side validation.
  - Upon successfully upserting, you should redirect back to the same page (with the user information displayed) or back to the view all list. What makes the most sense from a user-perspective?
  - Run and test your code.
  - Push your code to GitHub.
- 

### **Part III: Delete functionality**

- In the view all page, include a link/button/icon that can be used to delete a particular item.
  - Prior to deleting the item, make sure to check that the user wants to delete that item (for sure). You can use a Bootstrap modal to do this.
  - This should then be associated with a controller method (GET) that deletes that particular item and redirects to the view all page.
  - Run and test your code.
  - Push your code to GitHub.
- 

### **Submitting your code to D2L:**

1. Zip up the project folder and submit to the D2L assignment folder. I look at your code in GitHub and also use D2L as a way to know when your GitHub repo is ready to be reviewed.
2. As a comment in D2L (there's a place for students to comment when they upload files), please provide the database name, the username, and the password for your database.