Spring 2020 CSCI 4131 – Internet Programming
Homework Assignment 7
Due Saturday, May 1ˢᵗ 12:00 noon
Late Submissions, with Penalty, accepted through Sunday May 2nd, 6pm

1 Introduction

In this assignment, you will continue to add more functionality to the website you developed in the Homework 6. Your task is to upgrade the previous assignment to develop an updated contact's page with management features that enable you to add new contacts, deleting contacts, or update information on existing contacts of your website. Adding this functionality will require you to compose and submit delete and update SQL queries to the MySQL database from your node.js/Express server with information gathered from the user via HTML pages that you will create.

You will also work with XML (Extensible Markup Language) in this assignment. To enable flexibility and increase security, the database configuration will be saved in an XML file. Your server will read the configuration information from the XML file, and that information (which is the same information that you used in assignment 6) to connect to your MySQL database.

Most of your work (except the XML part) in this assignment will be focused on the Contacts page.

This assignment specification is 9 pages long.

2 Preparation and Provided Files

1. The setup for Node.js and MySQL remains the same as the last assignment.
2. The code for this assignment should be placed in a directory named: ***yourx500id_hw07***
3. The code from the last assignment can serve as a base for this assignment. Copy the code from the last assignment and place it in ***yourx500id_hw07*** directory.
4. You can use any *npm* module that you require for this assignment. You should use the modules you used in previous assignments, and you will need a module for parsing XML, Thus, you will need to install the following *npm* module: xml2js (npm install xml2js –save)
5. You are free to determine the project structure (where you will store the various html, css and JavaScript files) for this assignment on your own.

Note: *We have provided the following sample XML file which stores database configuration:*
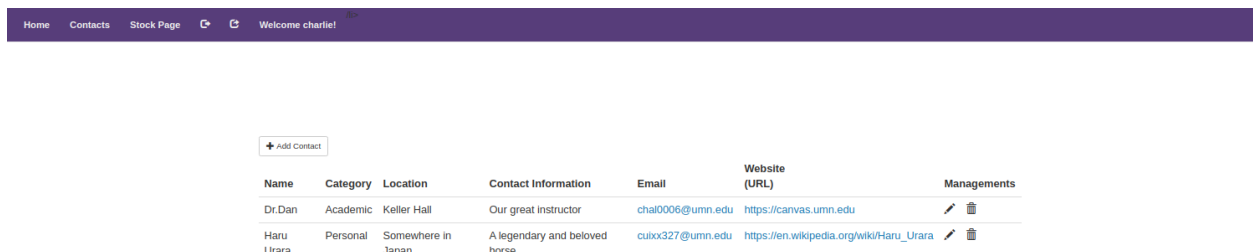
sample_dbconfig.xml

*Replace the values in the user, password, and database tags with the user/database id and numeric password that was provided via Canvas assignment (**MYSQL Database Account and Password**)*

3 Functionality

Your solution should retain all functionality from Homework assignment 6, and add the functionality discussed in the Contacts page (build/refactor your existing code).

The existing navigation bar should now also provide a button which triggers the functionality *to* export the data in the contacts table to a csv file (the icon to left of the message: "Welcome Charlie"). Finally, you will refactor the contacts page so that the functionality to add a contact is initiated by clicking on a button named "Add Contact" on the home page. So, the updated Contacts page should look similar to the image below:

Contacts page - Edit Contact functionality

- Each row in the table should have an **Edit** button associated with it (see the pencil icon displayed the just to the left of the garbage can on the images of the Contacts page above).
- Clicking **Edit** button should activate edit mode and display the row being edited in the list of contacts table as illustrated below.

- Once edit mode is entered for a contact, the row with the contact information being edited will display two buttons: **Update** and **Cancel**. You can use buttons with icons as displayed in the image illustrating the functionality above, or just create buttons with labels "Update" and "Cancel").

- You should be able to update the **Category**, **Location**, **Contact Information, Email** and **Website** in edit mode. *But the Name of a Contact cannot be modified.* If a user attempts to modify the **Name** of a contact, when the **Update** button is clicked (in the image below,

the update button is the disk icon to the left of the cancel icon), the webpage should generate an error message and stop the user from updating the Name. See the screenshots below for an illustration of this behavior.

- In the image above, a user attempts to change the name of the contact to Tianming, and in the image below, an error message is displayed after the update button is clicked.

However, note that the **Category**, **Location**, **Contact Information, Email** and **Website** information can be modified and stored without error when the user clicks on the **Update** button. The updated information will then be displayed on the contacts page (this is not shown)

- **Cancel** should **discard** the changes and exit edit mode, and the display the information that was present before edit mode was entered (see image below).

- The **Update** button should validate the data entered in the row being edited on the contact page as follows:
  - The html used to enter/accept the data should detect invalid input and, as done with the addContact functionality (specified below), if the user enters invalid input (for example, if there is empty input field and/or a syntactically incorrect URL is entered) - an error message should be displayed on the webpage.
  - If all the data entered by the user is valid, the data entered by the user should be sent (posted) to the server, and the updated information should be displayed on the contacts page.

Contacts page - Add Contact functionality

- As displayed in the picture at the top of page 2, the Contacts page should have a button to add a new contact.
- Upon pressing the **Add Contact** button, a new row with blank input text fields should be displayed the table. Then as shown in the screenshot below, the user should add a new contact by entering the required data in the input text fields. However, as specified in the discussion of the Edit functionality, your page should validate the input, and if the user has failed to enter all the required input and/or entered invalid input, an error message should be displayed if the **Save** button is clicked (as illustrated below):

Home	Contacts	Stock Page	↪	⬀	Welcome charlie!			

Error: missing required information.
Website field is required.

✚ Add Contact

Name	Category	Location	Contact Information	Email	Website (URL)	Managements
Dr.Dan	Academic	Keller Hall	Our great instructor	chal0006@umn.edu	https://canvas.umn.edu	✏ 🗑
Haru Urara	Personal	Somewhere in Japan	A legendary and beloved horse	cuixx327@umn.edu	https://en.wikipedia.org/wiki/Haru_Urara	✏ 🗑
Tianming	Personal ⌄	Keller Hall	I am just a poor boy	cuixx327@umn.edu		⏭ ✖

- If the user clicks **Save** with valid input, the data entered by the user should be posted to the server.
  - **However**, the server should validate that no other user in the database has the same name. If there is no other user with the same name in table *tbl_contacts*, the information about the new contact should be inserted in the table *tbl_contacts*.
  - Upon success new contact should be displayed in the table on the contacts page.
  - Otherwise, an error message should be displayed (see the screenshots below).
- If user clicks **Cancel (the x)**, the new row should be removed, and the display should revert back to the state before the **Add User** button was pressed.
- **Note**: Review screenshots I. and II on the next page for an illustration of how the **Add User** functionality should operate.

Screenshot: if the user enters valid input into all input fields, when the Save button is clicked the new contact should be added into the database, and then displayed on the contacts page:

					Website		
Name	**Category**	**Location**	**Contact Information**	**Email**	**(URL)**		**Managements**
Dr.Dan	Academic	Keller Hall	Our great instructor	chal0006@umn.edu	https://canvas.umn.edu		✏ 🗑
Haru Urara	Personal	Somewhere in Japan	A legendary and beloved horse	cuixx327@umn.edu	https://en.wikipedia.org/wiki/Haru_Urara		✏ 🗑
Tianming	Personal	Keller Hall	I am just a poor boy	cuixx327@umn.edu	https://cse.umn.edu		✏ 🗑

_**Screenshot**: The server should ensure the Name of the new contact is not a duplicate of a name that is already in the **tbl_contacts**. If it is, it should not be added into the database, and an error message shall be displayed:_

<div align="center">Error: Name of contacts cannot be duplicated!</div>

+ Add Contact

Name	**Category**	**Location**	**Contact Information**	**Email**	Website **(URL)**	**Managements**
Dr.Dan	Academic	Keller Hall	Our great instructor	chal0006@umn.edu	https://canvas.umn.edu	✏ 🗑
Haru Urara	Personal	Somewhere in Japan	A legendary and beloved horse	cuixx327@umn.edu	https://en.wikipedia.org/wiki/Haru_Urara	✏ 🗑
Tianming	Personal	Keller Hall	I am just a poor boy	cuixx327@umn.edu	https://cse.umn.edu	✏ 🗑
Tianming	Personal ▾	Keller Hall	Another one called Tianming	cuixx327@umn.edu	https://cse.umn.edu	💾 ✖

Contacts page - Delete Contacts functionality

- Each row in the table should have a **Delete** button (icon) associated with it (see the garbage can in the rightmost column).
- The **Delete** button should delete the contact from the database and from the HTML table. To achieve this, the server should provide a delete API which can be used by the client.

Screenshot: before clicking Delete button of the row with the Name: "Tianming".

+ Add Contact

					Website		
Name	**Category**	**Location**	**Contact Information**	**Email**	**(URL)**		**Managements**
Dr.Dan	Academic	Keller Hall	Our great instructor	chal0006@umn.edu	https://canvas.umn.edu		✏ 🗑
Haru Urara	Personal	Somewhere in Japan	A legendary and beloved horse	cuixx327@umn.edu	https://en.wikipedia.org/wiki/Haru_Urara		✏ 🗑
Tianming	Personal	Keller Hall	I am just a poor boy	cuixx327@umn.edu	https://cse.umn.edu		✏ 🗑

Screenshot: after clicking Delete button on the row with the Name: "Tianming". The row is removed from the *tbl_contacts* in the MySQL database and no longer displayed in the contacts table.
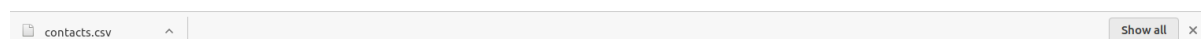

Navigation bar

- The navigation bar displays the user which is currently logged in as shown in previous screenshots, it shall also provide a new button "Export", this button will only be displayed on the Contacts page. You should also remove the "addContacts" page from the navigation bar.

Export functionality (bonus)

- Once the Export button is clicked, it will automatically generate a contacts.csv file and begin a download:

The contacts.csv should contains the information from the contact table, shown as the following screenshot.

	A	B	C	D	E	F	G
1	Name	Category	Location	Contact_Infomation	Email	Website	
2	Dr.Dan	Academic	Keller Hall	Our great instructor	chal0006@umn.edu	https://canvas.umn.edu	
3	Haru Urara	Personal	Somewhere in Japan	A legendary and beloved horse	cuixx327@umn.edu	https://en.wikipedia.org/wiki/Haru_Urara	
4	Tianming	Personal	Keller Hall	I am just a poor boy	cuixx327@umn.edu	https://cse.umn.edu	
5							
6							
7							
8							
9							

Updating the Database Configuration Functionality:

1. Your server should now read the values required to connect and login to your MySQL database from the file: `sample_dbconfig.xml` (*which has been provided, and YOU MUST UPDATE*) and use them to establish a database connection. **Note**, you can only run the xml2js parser from node.js/Express (the server).

4 Submission Instructions

PLEASE ENSURE TO TEST YOUR CODE ON CSE LAB MACHINES.

You will need to submit all the files used to develop the website. This includes all the HTML, CSS, JavaScript, package.json, index.js and any other files.

** **Make sure you have this account information in your accounts table. Violation to this will result in penalty. (-10 points, at a minimum)**

Login: admin$
Password: admin%

We will use this login and password values to login to your website. Your assignment should be submitted in a single compressed file with the name *yourx500id_hw07*. The server file should be named as *index.js*.

5 Evaluation

Your submission will be graded out of 110 (100+10) points on the following items:

- **Submission instructions are met.** This time we will no longer give [-0] for this. (5 points)
- Export functionality works correctly. (10 bonus points)
- Navigation bar in the Contact page displays the currently logged user. (5 points)
- On the Contacts page, the DELETE button works for every row correctly. (15 points)
- On the Contacts page, the EDIT button works correctly for every row and switches the view to EDIT mode when the EDIT button is selected. (10 points)
- On the Contacts page, the UPDATE button works correctly in EDIT mode. It validates the input, and updates the location, contact info, email and webpage accordingly. (10 points)
- On the Contacts page, the CANCEL button works correctly from EDIT mode. (5 points)
- On the Contacts page, the ADD CONTACT button works correctly and switches the view to ADD mode. (10 points)
- On the Contacts page, the SAVE button works correctly in ADD mode. It validates the input, and information accordingly. (10 points)
- On the Contacts page, the CANCEL button works correctly from ADD mode. (5 points)
- The name of the contacts cannot be duplicated, so ADD CONTACT should not allow the user to add a new contact with an existed name. Also, the name of a contacts cannot be modified by the EDIT mode. Corresponding error messages should be correctly displayed on the Contact page. (5 points)
- The Server reads database configuration from XML file and establishes connection to database. (5 points)
- All the functionality remaining from assignment 6 works correctly. (15 points)

6 Assignment Guidelines and Penalties – we reserve the right to grade subjectively.

1. The illustrations in the lab assignments and demos, though not stated, are implied requirements - though the style can differ, it shouldn't cause your output to look worse than the example given. **Therefore, though the CSS and page elements can vary but they should function the same as illustrated in the assignment write-up and demos**, and by our judgment, they should not negatively impact the functionality described in the write-up and demos.

2. Unless stated otherwise, for assignments that build on other assignments, the functionality developed in the previous assignment(s) **should** still work. Some examples are as follows (**note, this is not a complete list**):

    a. The login page still works, and the Contact page can be accessed only after logging in.

    b. The logout button still works.

You should follow these guidelines while working on your assignments. Failure to do so may result in penalties assessed against your score that are not explicitly stated in the rubric.