

# Spring 2022 CSCI 4131 – Internet Programming

## Homework Assignment 7

**Due Friday, April 29 11:59pm**  
**Late Submissions, with Penalty, accepted through Sunday May 1st, 12:00 noon**

### 1 Description

In this assignment, you will continue to add more functionality to the website developed in the last assignment. Your task is to upgrade the previous assignment to add a new user administration page with user management features that enable you to add new users, deleting users, or update information on existing users of your website, (and some other interesting features for the BONUS). 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.

### 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 might find the following *npm* module useful: ***xml2js***
5. You should 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 the assignment on the class Canvas site named **MYSQL Database Account and Password***

### 3 Functionality

Your solution should retain all functionality from Homework assignment 6, and add the functionality discussed in the following pages (build on your existing code).

The existing navigation bar should now also provide a link to the **Admin** page and show a welcome information for the **logged in User**. Check the following screenshot as an example (for other pages, you also need to add this new link to the Admin page and the welcome information):

The screenshot shows the Admin page with a navigation bar at the top. The navigation bar includes links for Home, Admin Page (highlighted), All Contacts, My Contacts, Add Contact, Stocks Page, and LOG OUT. A welcome message "Welcome user!" is displayed on the right. Below the navigation bar, there is a button labeled "+ Add User". The main content area features a table with the following columns: Id, Name, Login, New Password, and a set of icons (edit, delete, and a right arrow). The table contains five rows of user data.

Id	Name	Login	New Password	
1	user	user		
123456	123456	123456		
123458	qqq	qqq		
123459	zejun	zejun		
123460	zejun	zejun1018		

Below the table, there is another navigation bar with links for Home, Admin Page, All Contacts (highlighted), My Contacts, Add Contact, Stocks Page, and LOG OUT. A welcome message "Welcome user!" is displayed on the right. The main content area features a table with the following columns: Category, Name, Location, Contact Information, Email, and Website. The table contains two rows of contact data.










Category	Name	Location	Contact Information	Email	Website
academic	1111	2929	1111	zhan7565@umn.edu	Google
academic	Zejun	2929	1111	zhan7565@umn.edu	Google

## Admin page (New)

- If a user tries to access this page without a valid login, the user should be routed to the "Login" page.
- The page should have a navigation bar with a logout button.
- The table in this page should be dynamically populated with the list of users along with their corresponding Id, Name, and Login (illustrated in the following screen-shot).
- To achieve this, the server could provide a GET API which returns the list of users. The mapping between the table headers and columns in tbl\_accounts is as follows:
  - Id: acc\_id
  - Name: acc\_name
  - Login: acc\_login
  - New Password: acc\_password
- The client (the page you will construct) could call the API (using **AJAX**) and populate the table using the data received from the server.

+

Add User

Id	Name	Login	New Password	
1	user	user		  
123456	123456	123456		  
123458	qqq	qqq		  
123459	zejun	zejun		  
123460	zejun	zejun1018		  






Note: acc\_login should be the thing that really matters – when you tried to log in with an account, you will use acc\_login (instead of acc\_name) and acc\_password, it is also acc\_login which will be shown on the welcome information, as shown in the screenshot. acc\_name is just a name (or marker) for a name that the person who owns the account provides.

## Admin page - Add User functionality

- As displayed in the picture on the above, the Admin page should have a button to add a new user.
- Upon pressing the **Add User** button, a new row with blank fields (should be added to the table) as shown in the screenshot I on the next page.
- Name, Login, and New Password** columns of this new row should be editable. **All of these three fields are required.** You can check if these fields are empty either on the admin page before submission or on the server side before saving the record to the database. **Your database should not contain any user record with empty name, login, or password.**
- A user has the option to enter the details of the new user and then click either **Save** or **Cancel** button.
- If the user clicks **Save**, the data entered by the user should be posted to the server.
  - The server should validate that no other user in the database has the same acc\_login.
  - In case the validation passes, the information about the new user should be inserted in the following table: *tbl\_accounts*.
  - The new user should also be added to the HTML table.
  - In case the validation fails, the following error message should be displayed: **This login is used by another user**
- If user clicks **Cancel**, the new row should be removed and display should revert back to the state before **Add User** button was pressed
- Note:** Review screenshots I. and II below for an indication of how the **Add User** functionality should operate.







*I. Admin enters the details of a new user and clicks the **Save button***

+ Add User

Id	Name	Login	New Password	
1	user	user		  
	<input type="text" value="Zejun"/>	<input type="text" value="Zejun"/>	<input type="text" value="123456"/>	 

*II. The new user is successfully added.*









+ Add User

Id	Name	Login	New Password	
1	user	user		  
123465	Zejun	Zejun		  

*III. Error messages shown when there is insufficient information:*

Error: missing required information.  
Login field is required.  
Password field is required.

+ Add User

Id	Name	Login	New Password	
1	user	user		  
123465	Zejun	Zejun		  
	<input type="text" value="qqqqq"/>	<input type="text"/>	<input type="text"/>	 

IV. Error messages shown when trying to add a user whose acc\_login is already used by another user (Note: check the screenshots, the Name can be duplicated, but the acc\_login cannot).

Error: missing required information.  
Password field is required.

+ Add User

Id	Name	Login	New Password	
1	user	user		✎ 🗑 ▶
123465	Zejun	Zejun		✎ 🗑 ▶
	<input type="text" value="qqqqq"/>	<input type="text" value="Zejun"/>	<input type="password"/>	⌂ ✕

## Admin page - Delete User 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 user from the database and from the HTML table only if the user being deleted is not the one currently logged in. To achieve this, the server should provide a delete API which can be used by the client.
- The server should not allow the user to delete the currently logged user. The following error message should be displayed on the Admin page in the event that a user attempts to do this: **Error: Cannot delete logged in User!**
- **Note:** Review screenshots A, B, and C below depicting the **Delete User** functionality.

A. User clicks on the delete button (trash icon) next to the user **Zejun\_option** option.

Home
Admin Page
All Contacts
My Contacts
Add Contact
Stocks Page
LOG OUT
Welcome user!

+ Add User

Id	Name	Login	New Password	
1	user	user		✎ 🗑 ▶
123465	Zejun	Zejun		✎ 🗑 ▶
123466	Zejun	Zejun_option		✎ 🗑 ▶

B. Zejun\_option is deleted from the database and from the HTML table.

+ Add User

Id	Name	Login	New Password	
1	user	user		  
123465	Zejun	Zejun		  

C. User tries to delete the user named **user**, who is currently logged in. The user is not deleted and an error message is displayed.

Home

Admin Page

All Contacts

My Contacts

Add Contact







Stocks Page

LOG OUT

Welcome user!

Error: This login is used by another user!

+ Add User

Id	Name	Login	New Password	
1	user	user		  
123465	Zejun	Zejun		  

## Admin page - Edit User functionality

- Each row in the table should have an **Edit** button associated with it (see the pencil icon in the just to the left of the garbage can).
- Clicking **Edit** button should activate edit mode and display it in the list of users table.
- You should be able to update the **Name**, **Login**, and **New Password** in edit mode. **Notice that your database should not contain records with any empty fields. But for the update mode, even though you cannot save an empty password into the database, you should still allow the empty password field for submission. An empty password field implies that only the name and login may be changed and the password should not be changed. You can identify such cases on the server side to avoid saving an empty password into the database.**
- Edit mode will have two different buttons: **Update** and **Cancel**.
- **Cancel** should **discard** the changes and exit edit mode.
- Upon clicking the **Update** button, the data entered by the user should be posted to the server.
  - The server should validate that no other user in the database (other than the one being edited) has the same login.
  - In case the validation passes, the information about the edited user should be updated in the following table: **tbl\_accounts**. The updated information should also reflect in the HTML table.
  - In case the validation fails, the following error message should be displayed: **Error: This login is used by another user.**







**Note:** Review the screenshots i, ii, ii, iv, v and vi depicting the **Edit User** functionality. Here you can decide whether it is possible to modify the “current login user”, both will be OK.

i) Admin clicks on the edit button next to the user **Zejun**.

Home	Admin Page	All Contacts	My Contacts	Add Contact	Stocks Page	LOG OUT	Welcor
------	------------	--------------	-------------	-------------	-------------	---------	--------

+ Add User

Id	Name	Login	New Password	
1	user	user		  
123465	Zejun	Zejun		  

ii) Admin updates the values to **Zejun** and clicks on save (the disk icon).

+ Add User

Id	Name	Login	New Password	
1	user	user		  
123465	<input type="text" value="Zejun_new"/>	<input type="text" value="Zejun_new"/>	<input type="text" value="123456"/>	  

iii) The values are successfully updated in database and displayed in the user table.







+ Add User

Id	Name	Login	New Password	
1	user	user		  
123465	Zejun_new	Zejun_new		  

iv) The user edits the details of user **Zejun** and updates the acc login to **user**. Error message is displayed because a user with login **user** already exists.

Error: This login is used by another user!






+ Add User

Id	Name	Login	New Password	
1	user	user		  
123465	<input type="text" value="Zejun_new"/>	<input type="text" value="user"/>	<input type="text"/>	  

v) Admin clicks on **Cancel** button to discard the changes (the circular icon next to the disk icon).

Error: This login is used by another user!

+ Add User

Id	Name	Login	New Password	
1	user	user		  
123465	<input type="text" value="Zejun_new"/>	<input type="text" value="user"/>	<input type="text"/>	 






+ Add User

Id	Name	Login	New Password	
1	user	user		  
123465	Zejun_new	Zejun_new		  

vi) Error message of missing values, similar to add user part.

Error: missing required information.  
Name field is required.

+ Add User

Id	Name	Login	New Password	
1	user	user		  
123465	<input type="text"/>	<input type="text" value="Zejun_new"/>	<input type="text"/>	 







## Admin page - “Switch to” functionality (BONUS)

Notice the welcome information, currently user is logged in.

Home Admin Page All Contacts My Contacts Add Contact Stocks Page LOG OUT

Welcome user!




+ Add User

Id	Name	Login	New Password	
1	user	user		  
123465	Zejun	Zejun		  







Once the switch to button next to Zejun is clicked, the logged-in user is switched to Zejun:



+ Add User

Id	Name	Login	New Password	
1	user	user		  
123465	Zejun	Zejun		  

+ Add User

Id	Name	Login	New Password	
1	user	user		  
123465	Zejun	Zejun		  

## Navigation bar

- The navigation bar displays the user which is currently logged in.

## Export functionality (BONUS)

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

Category	Name	Location	Contact Information	Email	Website
academic	1111	2929	1111	zhan7565@umn.edu	<a href="#">Google</a>
academic	Zejun	2929	1111	zhan7565@umn.edu	<a href="#">Google</a>
industry	Tianming	keller Hall	12345	1234@umn.edu	<a href="#">Google</a>
personal	Zejun	keller Hall	TA	zhan7565@umn.edu	<a href="#">Google</a>

- The contacts.csv should contain at least the following information from the contact table, shown as the following screenshot. You can also include the data in the info field of the contact table (but it is not required).

	A	B	C	D	E	F	
1	Name	Category	Location	Contact Information	Email	Website	
2	1111	academic	2929	1111	zhan7565@umn.edu	https://www.google.com	
3	Zejun	academic	2929	1111	zhan7565@umn.edu	https://www.google.com	
4	Tianming	industry	keller Hall	12345	1234@umn.edu	https://www.google.com	
5	Zejun	personal	keller Hall	TA	zhan7565@umn.edu	https://www.google.com	
6							

## Database Configuration

- Your server should read the values in the file: sample\_dbconfig.xml (**which was provided, and YOU MUST UPDATE**) and use them to establish a database connection. Note, you can only run the parser from node.js/Express, so make sure to insert ample calls to console.log to check to make sure your server is functioning as you expected as you develop your solution.

## 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 a penalty.**

**Login: admin**

**Password: admin**

**And as specified in Programming Homework Assignment 6, Please remove the node\_modules/ folder from your submission! Do not leave it in your submission!**

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 **index.js**.

## 5 Evaluation

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

- **Submission instructions are met (5 points).**
- Navigation bar in all the pages displays the currently logged user. (5 points)
- Admin page correctly displays the list of users in the system. The Admin page should display at least two other users besides the admin user when submitted. (10 points)
- On the Admin page, the DELETE button works for every row and displays error messages correctly. (10 points)
- On the Admin page, the EDIT button works for every row and switches the view to EDIT mode. (10 points)
- On the Admin page, the UPDATE button works in EDIT mode. It validates the input, and updates the name, login, password accordingly. All the test cases should work (10 points)
- On the Admin page, the CANCEL button works in EDIT mode. (5 points)
- On the Admin page, ADD USER works and switches the view to ADD mode. (10 points)
- On the Admin page, the SAVE button works in ADD mode. It validates the input, and saves the name, login, password accordingly. (10 points)
- On the Admin page, the CANCEL button works in ADD mode. (5 points)
- The Server reads database configuration from XML file and establishes connection to database. (10 points)
- All functionality from assignment 6 works. (10 points)
- Switch-to functionality (5-point BONUS)
- “Export” functionality (5-point BONUS)

## **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 addEvent page should still accept the values for event name, event location, event day, etc. required by previous assignments and no null values should be accepted or displayed once the event is added.
- b. The extra info url and text should be the same field and not two different columns on the events page.

**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.**