CSCI 4131 – Internet Programming Homework Assignment 7 (Version 1, 11/28)

Due Monday, December 10th at 11:55pm (10 point bonus)

Final Submission Date (with no penalty): **Friday, December 14th at 2pm**

NOTE: No HW7 submissions will be accepted after Friday, December 14th at 2pm

1 Description

In this assignment, you will continue to add more functionality to the website you developed in the last assignment (HW 6) . Your task is to upgrade the previous assignment by adding a new user administration page with user management features that include the capability to add new users, delete existing users, and update information about existing users.

You will also use XML (Extensible Markup Language) in this assignment. Your database configuration and login information will be saved in an XML file and your server will read the configuration details from the XML file and use it to establish a connection to the database.

2 Preparation and Files Provided

1. The setup for Node.js and MySQL remain the same as the last assignment (you will need to go through the setup for the npm modules (Steps 8 and 9 from the HW Assignment 6 write-up) again in your HW 7 directory / folder).
2. The code for this assignment should be placed in a directory named: ***yourx500id_hw07***
3. The code from last assignment can serve as a base for this assignment. Copy the code from last assignment and place it in ***yourx500id_hw07*** directory.
4. You can use any npm module that you deem fit for this assignment. You will find the following npm module useful: ***xml2js*** ***(****add it to package.json with the npm install command with the save option****).***
5. You are free to decide your own project (i.e., directory) structure for this assignment.

Note: *We have provided the following sample XML file which should be used to store your database configuration:*

dbconfig.xml

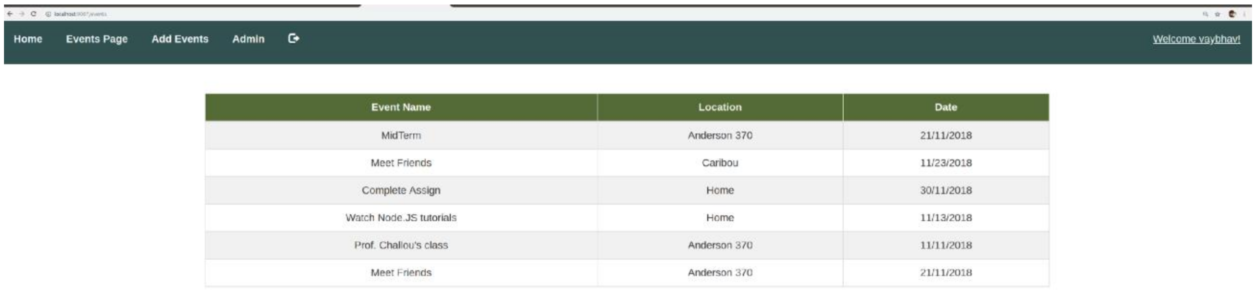
Replace the values in the user, password, and database tags with your database user, password, and database information.

3 Functionality

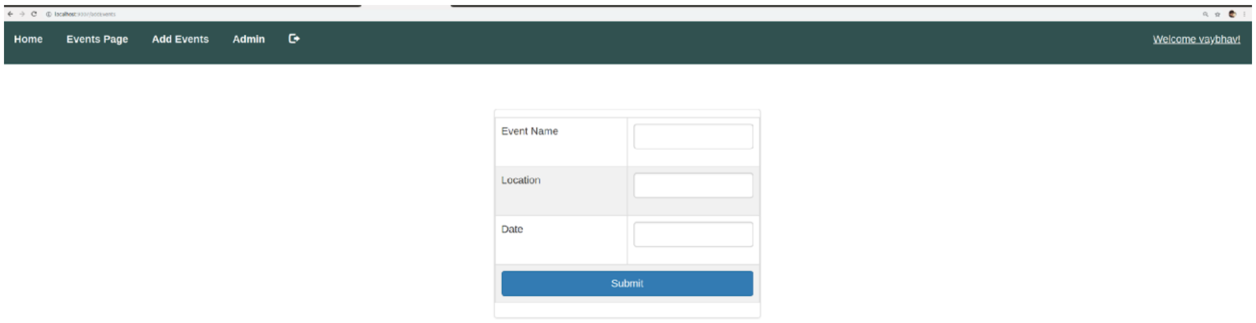
This assignment you to retains all the functionality from Assignment 6 and build on it

The navigation bar in **Events** page and **Add Events** page should provide a link to the **Admin page** .

See the two screenshots below:



Event Name	Location	Date
MidTerm	Anderson 370	21/11/2018
Meet Friends	Caribou	11/23/2018
Complete Assign	Home	30/11/2018
Watch Node.JS tutorials	Home	11/13/2018
Prof. Challou's class	Anderson 370	11/11/2018
Meet Friends	Anderson 370	21/11/2018



*<u>Notice that the navigation bar in both pages has a link to the **Admin page**.</u>*

Admin page - *List Users* functionality

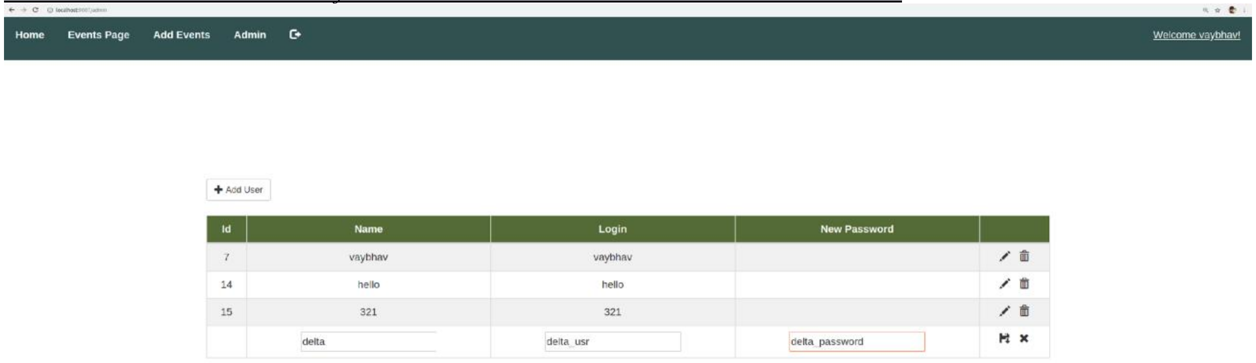
- If a user tries to access this page without a valid login, the user should be routed to "Login" page.
- The page should have a navigation bar with logout button.
- The table in this page should be dynamically populated with the list of users along their corresponding **Id**, **Name**, and **Login** (illustrated in the screen-shot on the top of the next page).
- To achieve this, the server should provide a GET API which returns the list of users and the information associated with each user. The mapping between the table headers and columns in `tbl_accounts` is as follows:
  - **Id**: `acc_id`
  - **Name**: `acc_name`
  - **Login**: `acc_login`
- The client can call the API and then build table a table populated with the data received from the server.

Admin page - *Add User* functionality

- This page should have a button to add new user.
- Upon pressing the **Add User** button, a new row should be populated in the table.
- **Name**, **Login**, and **New Password** columns of this new row should be editable.
- The user has the option to enter the details of the new user and then click either **Save** or **Cancel** button.
- If 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 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 be reverted to the state before **Add User** button was pressed.

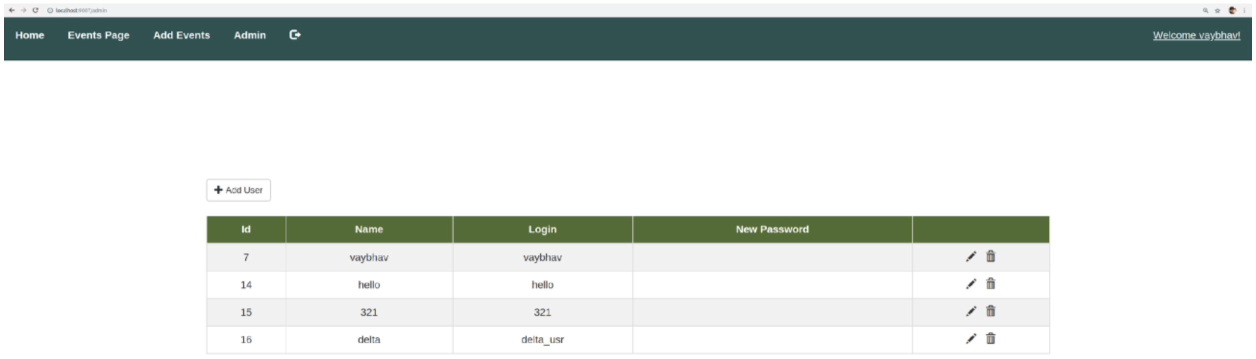
Note: Review the screenshots on the next page related to **Add User** functionality.

Admin enters the details of a new user named “delta” and clicks ***Save***



Id	Name	Login	New Password	
7	vaybhav	vaybhav		
14	hello	hello		
15	321	321		
	delta	delta_usr	delta_password	

The new user is successfully added.



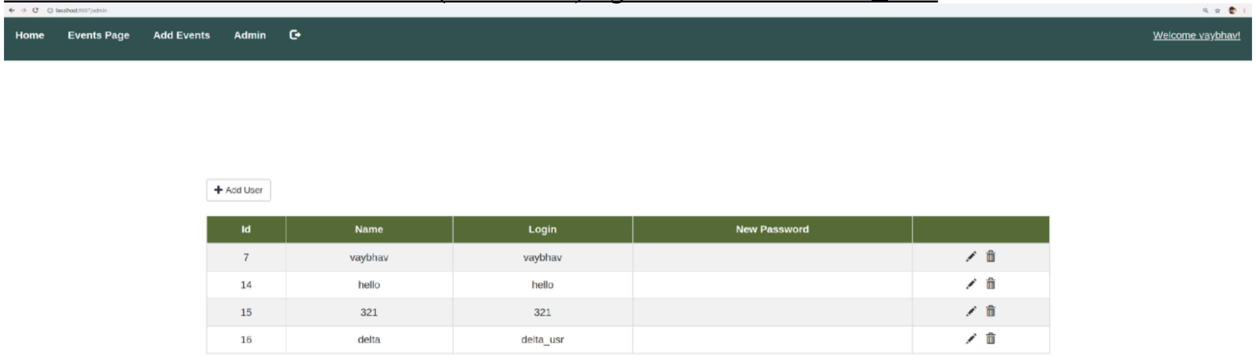
Id	Name	Login	New Password	
7	vaybhav	vaybhav		
14	hello	hello		
15	321	321		
16	delta	delta_usr		

Admin page - *Delete User* functionality

- Each row in the table should have **Delete** button associated with it.
- 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, server should provide a delete API which can be used by the client.
- As mentioned in the bulleted item above your server should not allow a user that is logged in to be deleted. If a user tries to delete a user that is logged in, the following error message should be displayed: Can not delete the user that is logged in

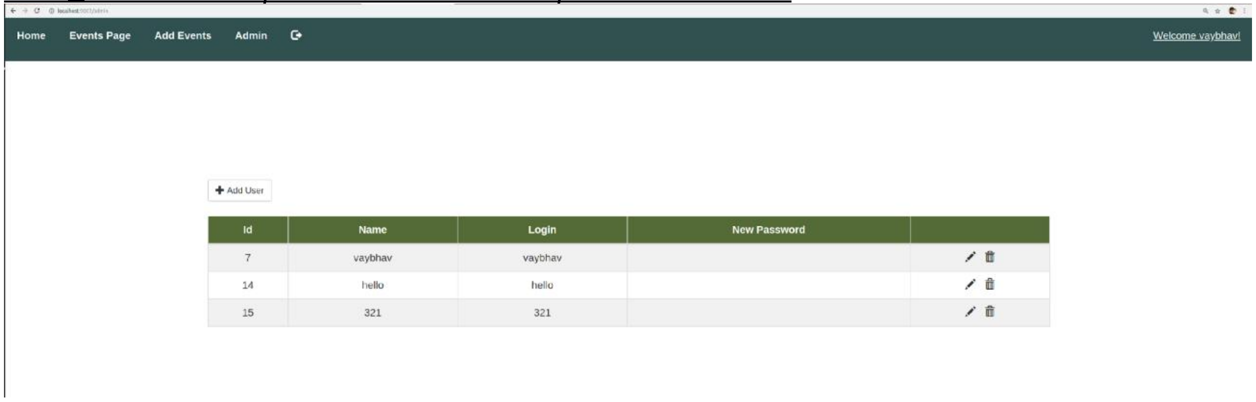
Note: Review the screenshots on the next page related to **Delete User** functionality.

Admin clicks on the delete button (trash icon) against the user delta_usr.



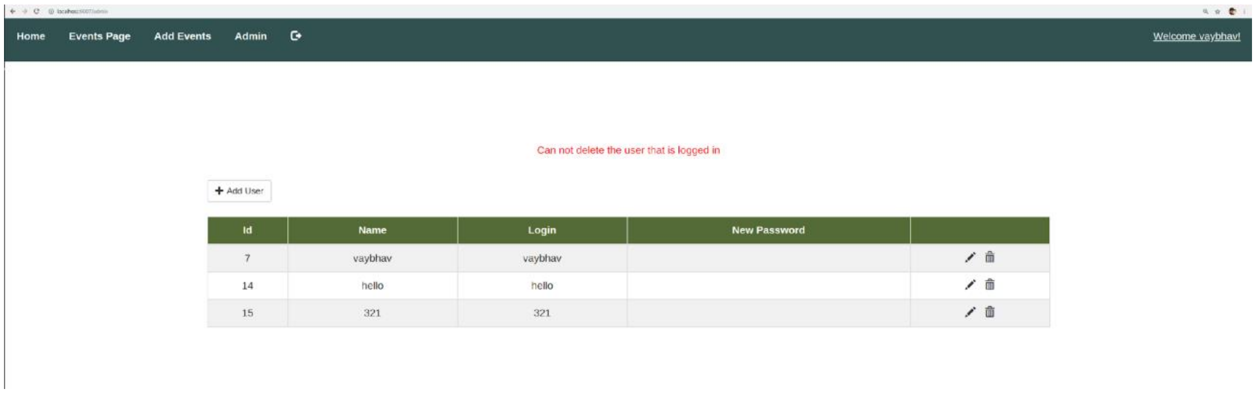
Id	Name	Login	New Password	
7	vaybhav	vaybhav		
14	hello	hello		
15	321	321		
16	delta	delta_usr		

delta_usr is deleted from the database and from HTML table.



Id	Name	Login	New Password	
7	vaybhav	vaybhav		
14	hello	hello		
15	321	321		

Admin tries to delete the user vaybhav, whom is currently logged in. The user is not deleted and an error message is displayed.



Can not delete the user that is logged in

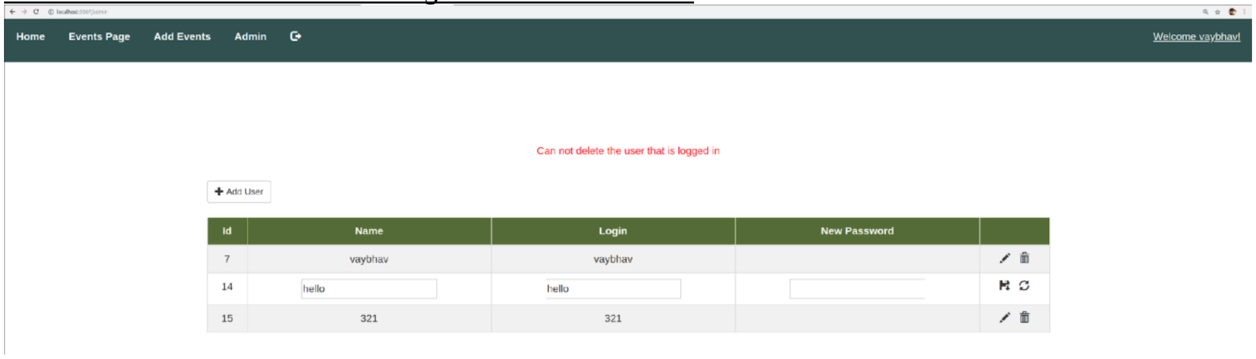
Id	Name	Login	New Password	
7	vaybhav	vaybhav		
14	hello	hello		
15	321	321		

Admin page - *Edit User* functionality

- Each row in the table should have edit button associated with it.
- Clicking the **Edit** button should activate edit mode, and display an editable row in the users table.
- You should be able to update the **Name**, **Login**, and **New Password** in edit mode.
- 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 database table: ***tbl_accounts***. The updated information should also be displayed in the HTML table displayed in the browser.
  - In case the validation fails, the following error message should be displayed: This login is used by another user

Note: Review the screenshots below and on the next page related to **Edit User** functionality.

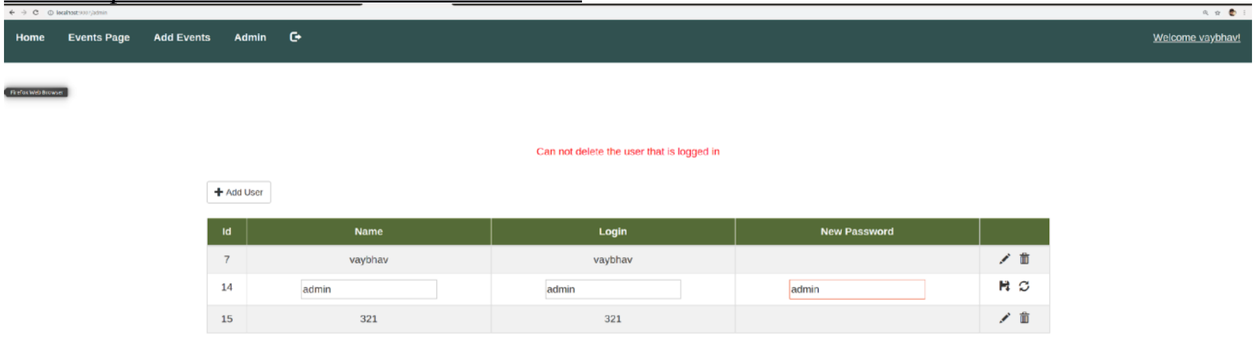
Admin clicks on the edit button against the user hello.



Can not delete the user that is logged in

Id	Name	Login	New Password	
7	vaybhav	vaybhav		
14	hello	hello		
15	321	321		

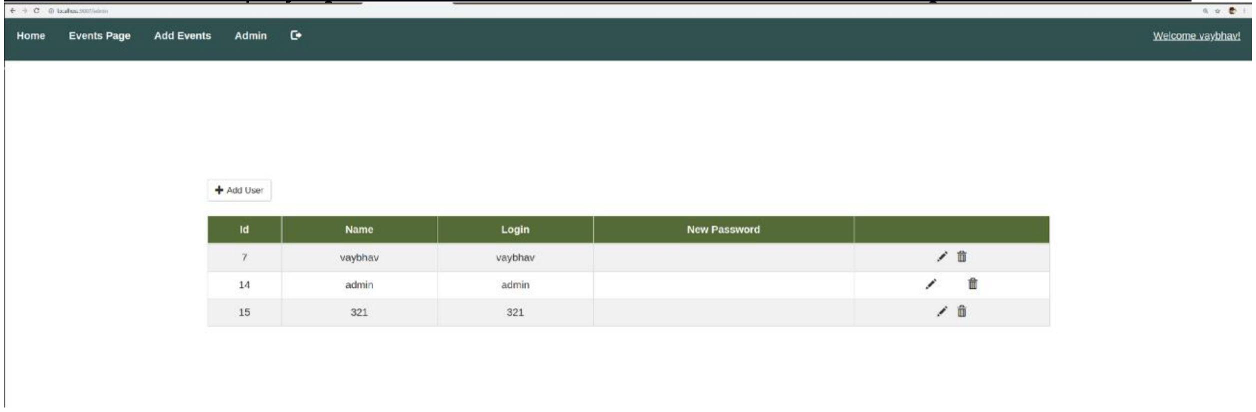
Admin updates the values and clicks on save.



Can not delete the user that is logged in

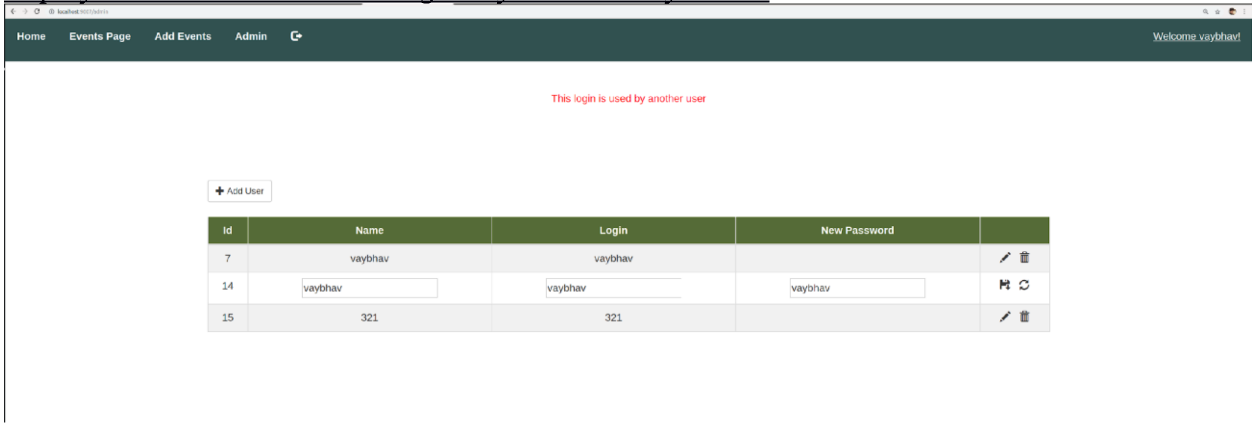
Id	Name	Login	New Password	
7	vaybhav	vaybhav		
14	admin	admin	admin	
15	321	321		

Values are successfully updated in database and the result is shown in updated HTML table.



Id	Name	Login	New Password	
7	vaybhav	vaybhav		
14	admin	admin		
15	321	321		

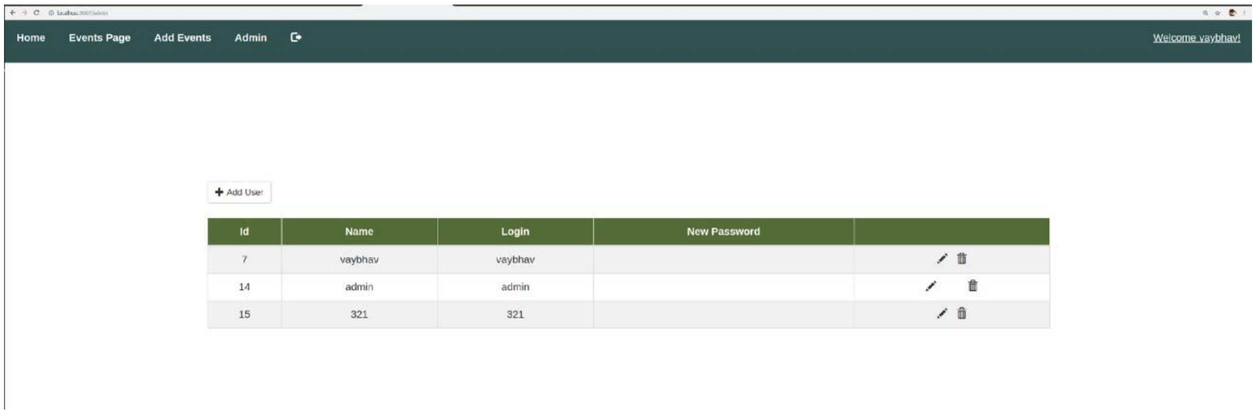
Admin edits the details of user admin and updates the login to vaybhav. Error message is displayed because a user with login vaybhav already exists.



This login is used by another user

Id	Name	Login	New Password	
7	vaybhav	vaybhav		
14	vaybhav	vaybhav	vaybhav	
15	321	321		

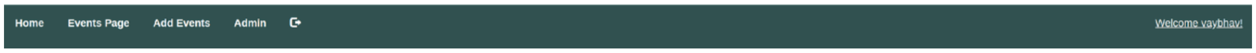
*Admin clicks on **Cancel** button to discard the changes.*



Id	Name	Login	New Password	
7	vaybhav	vaybhav		
14	admin	admin		
15	321	321		

Navigation bar

- Note, the navigation bar should display the user that is currently logged in.



Database Configuration

Your server should read the values in the file: **dbconfig.xml** to establish a database connection. You must edit the file and enter your *database user*, *password,* and *database* information.

Obtaining ICONs used on the web for logout, edit (the pencil), delete (the garbage can), etc.

You can obtain the icons shown in the pictures above for use in your solution at the following link:
https://www.w3schools.com/bootstrap/bootstrap_ref_comp_glyphs.asp

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:

Login: admin

Password: admin

Create a **README** file inside **yourx500id_hw07** directory. This README file should include: Your x500id, login, and password for your website.

Compress the **yourx500id_hw07** directory and submit it**.**

We will use the login and password values provided by you in README file to login to your website. Ensure that these values are correct and can be used to access your website.

5 Evaluation

Your submission will be graded out of 100 points on the following items:

- Submission instructions are not met (-10 points).
- The Navigation bar displays the currently logged user. (5 points)
- The Admin page displays the correct list of users in the system. (10 points)
- In the Admin page, the DELETE button works for every row and displays error messages correctly. (10 points)
- In the Admin page, the EDIT button works for every row and switches the view to EDIT mode when selected. (10 points)
- In the Admin page, UPDATE button works in EDIT mode. It validates the input, and updates the name, login, password accordingly. (10 points)
- In the Admin page, CANCEL button works in EDIT mode. (5 points)
- In the Admin page, ADD USER works and switches the view to ADD mode. (10 points)
- In the Admin page, SAVE button works in ADD mode. It validates the input, and saves the name, login, password accordingly. (10 points)
- In the Admin page, CANCEL button works in ADD mode. (5 points)
- The server reads database configuration from XML file and uses it to connect to the database. (10 points)
- All functionality from assignment 6 works. (15 points)
- **Bonus** – Submit your final version of this assignment by 11:55pm, Monday December 10th (10 points)