CSCI 4131 – Internet Programming
Homework Assignment 7

Posted April 8th, 2018

Due: Friday April 20 at 11:55 PM

Submissions Accepted (with Penalty) through Saturday April 21 at 11:55AM

1 Description

In this assignment, you will continue to add more functionality to the website you developed in the last assignment. Your task is to upgrade previous assignment by adding a user administration page with user management functionality for adding new users; updating information for existing users, and deleting existing users.

You will also work with XML (Extensible Markup Language) in this assignment. The database configuration for your website will be saved in an XML file and your server will read the configuration details from the XML file and use the information to stored in the XML file to establish a connection to the database. Note, this assignment description has 13 pages (and most of them are pictures to describe the functionality required for this assignment).

2 Preparation and Provided Files

1. The setup for Node.js and MySQL remain 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 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 might find the following npm module useful: ***xml2js (which is useful for parsing XML and storing it in a javaScript-friendly format.***
5. You are free to formulate your own project structure for this assignment.

Note: *We have provided the following sample XML file which stores database configuration:* ***sample_dbconfig.xml***
Replace the values in user, password, and database tags with the value you used for HW 6.

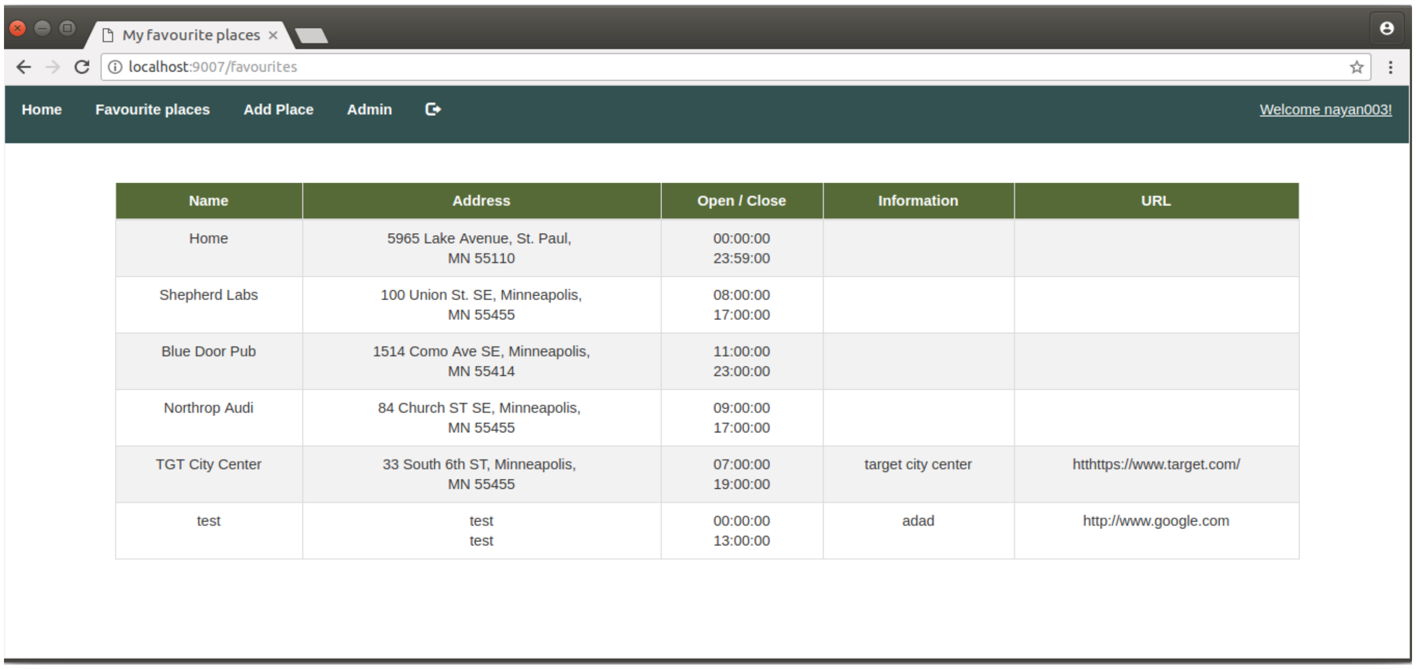
3 Functionality

This assignment will retain all the functionality from Assignment 6 and build on existing code.

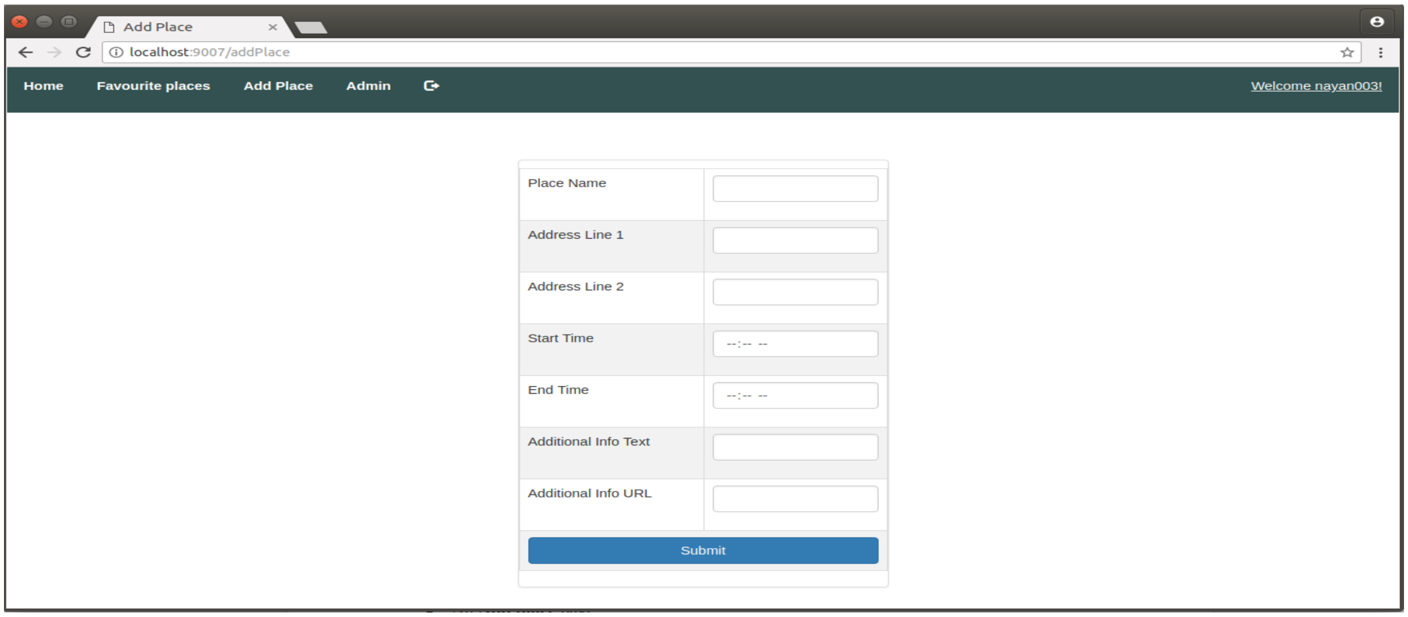
The navigation bar in **Favourite places** page and **Add place** page should provide a menu item to access the **Admin page** . See the screen shots that follow.

NOTE: you do not need the Home menu item (or to implement a page associated with the Home menu it) pictured in the screenshots that follow.

*Notice that the navigation bar in both pages has link to **Admin page**.*



Name	Address	Open / Close	Information	URL
Home	5965 Lake Avenue, St. Paul, MN 55110	00:00:00 23:59:00		
Shepherd Labs	100 Union St. SE, Minneapolis, MN 55455	08:00:00 17:00:00		
Blue Door Pub	1514 Como Ave SE, Minneapolis, MN 55414	11:00:00 23:00:00		
Northrop Audi	84 Church ST SE, Minneapolis, MN 55455	09:00:00 17:00:00		
TGT City Center	33 South 6th ST, Minneapolis, MN 55455	07:00:00 19:00:00	target city center	hthttps://www.target.com/
test	test test	00:00:00 13:00:00	adad	http://www.google.com

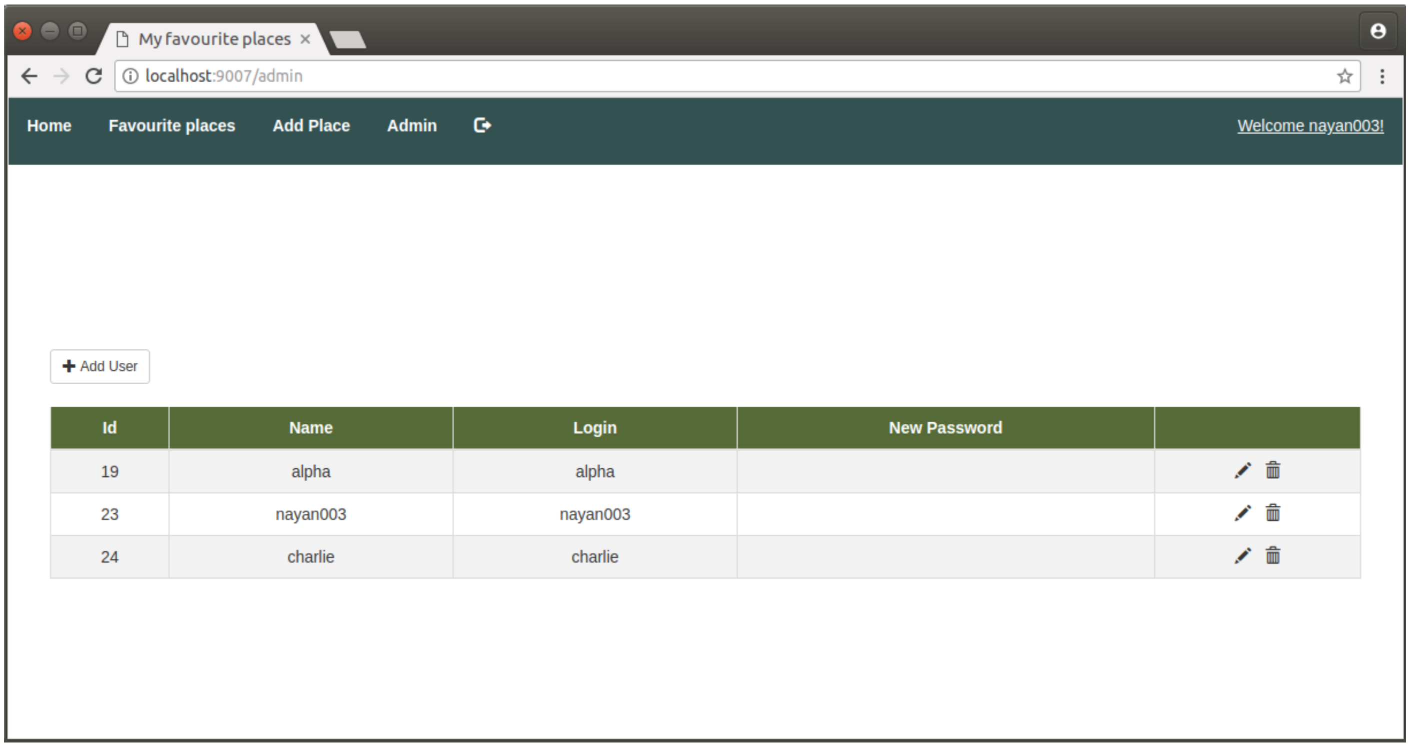


Place Name	
Address Line 1	
Address Line 2	
Start Time	--:-- --
End Time	--:-- --
Additional Info Text	
Additional Info URL	

Submit

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 following screen-shot).
- To achieve this, you could create a GET API on your server 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
- The client (HTML/CSS/JavaSrcript page) could call the API and populate the table using the data received from the server.



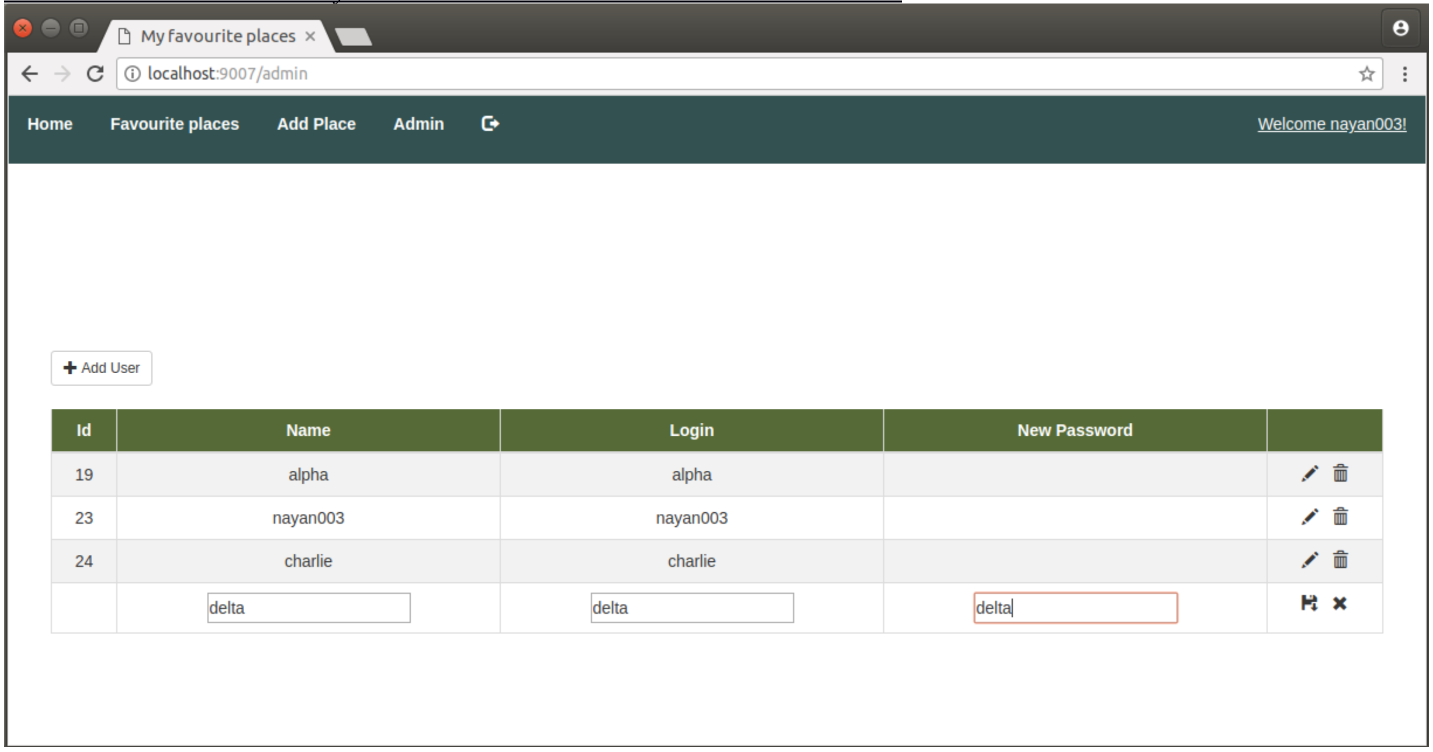
Id	Name	Login	New Password	
19	alpha	alpha		
23	nayan003	nayan003		
24	charlie	charlie		

Admin page - Add User functionality

- This page should have a button to add new user.
- After pressing the **Add User** button, a new row should be displayed in the table.
- The **Name**, **Login**, and **New Password** columns of this new row should be editable.
- The should be able to enter the details for a new user (Name, Login, and Password) and then click either **Save** or **Cancel** button.
- If user clicks **Save**, the data entered by the user should be posted (i.e., sent via a post message) to the server.
    - The server should ensure that no other user in the database has the same login.
    - If no existing user has the same login, validation passes, and the information about the new user should be inserted in the following table: ***tbl_accounts***.
    - A row with new user information should be displayed in the Admin table.
    - In case the validation fails, the following error message should be displayed: This login is used by another user, and the information entered on the page before the Save button was clicked should remain on the page.
- If user clicks **Cancel**, the new row should be removed and the table displayed should revert back to the state it was in before **Add User** button was pressed.

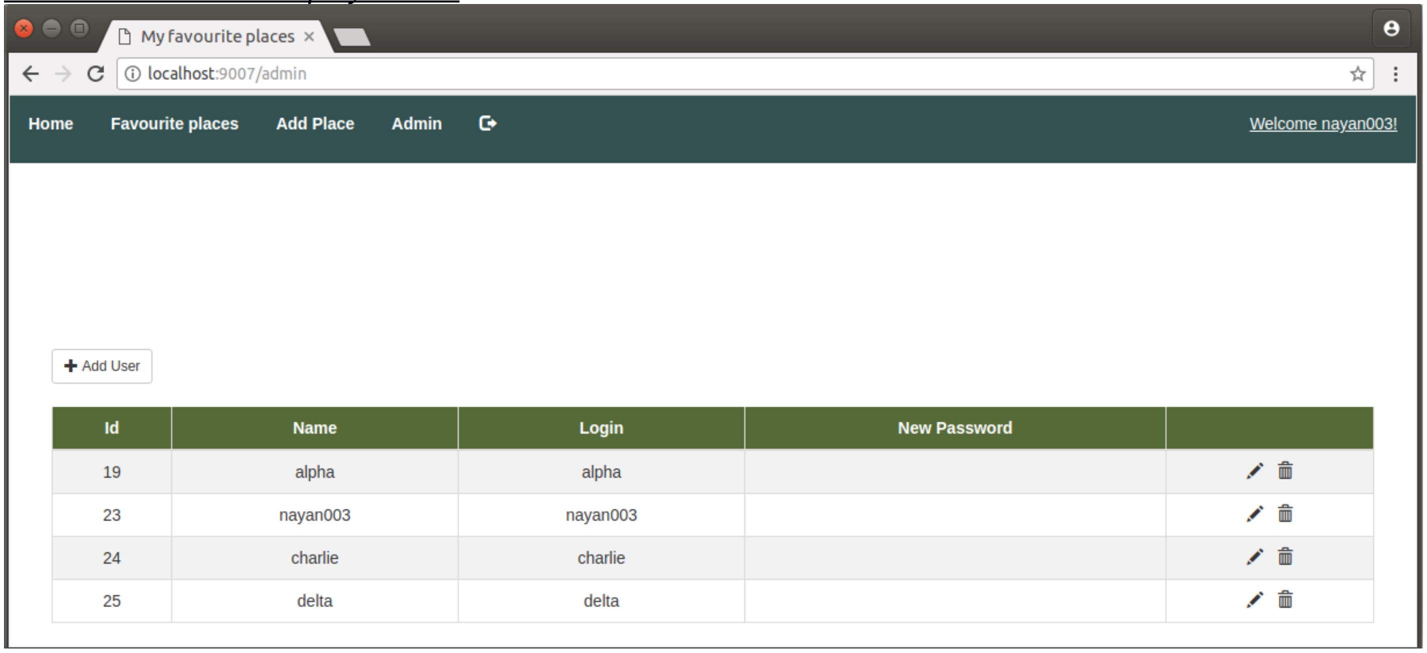
Note: Please review the following screenshots related to **Add User** functionality.

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



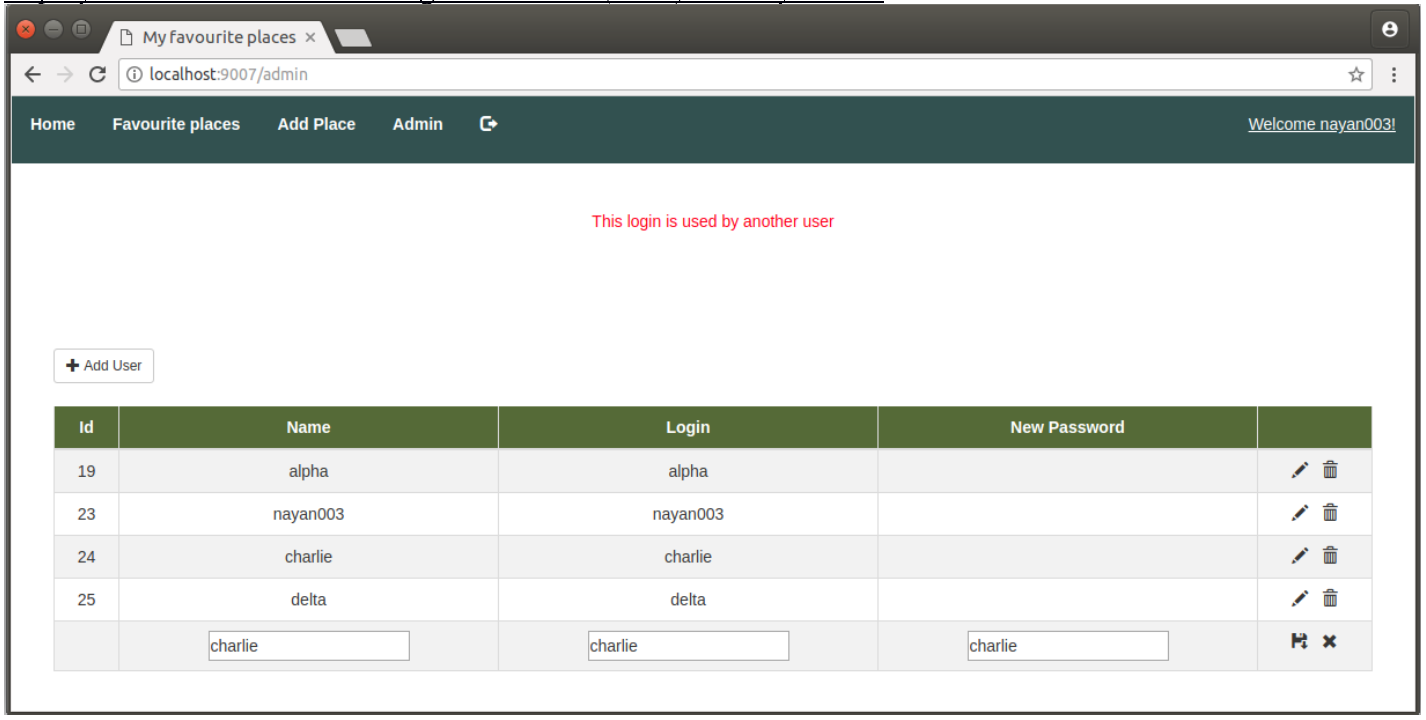
Id	Name	Login	New Password	
19	alpha	alpha		
23	nayan003	nayan003		
24	charlie	charlie		
	delta	delta	delta	

The new user is successfully added.



Id	Name	Login	New Password	
19	alpha	alpha		
23	nayan003	nayan003		
24	charlie	charlie		
25	delta	delta		

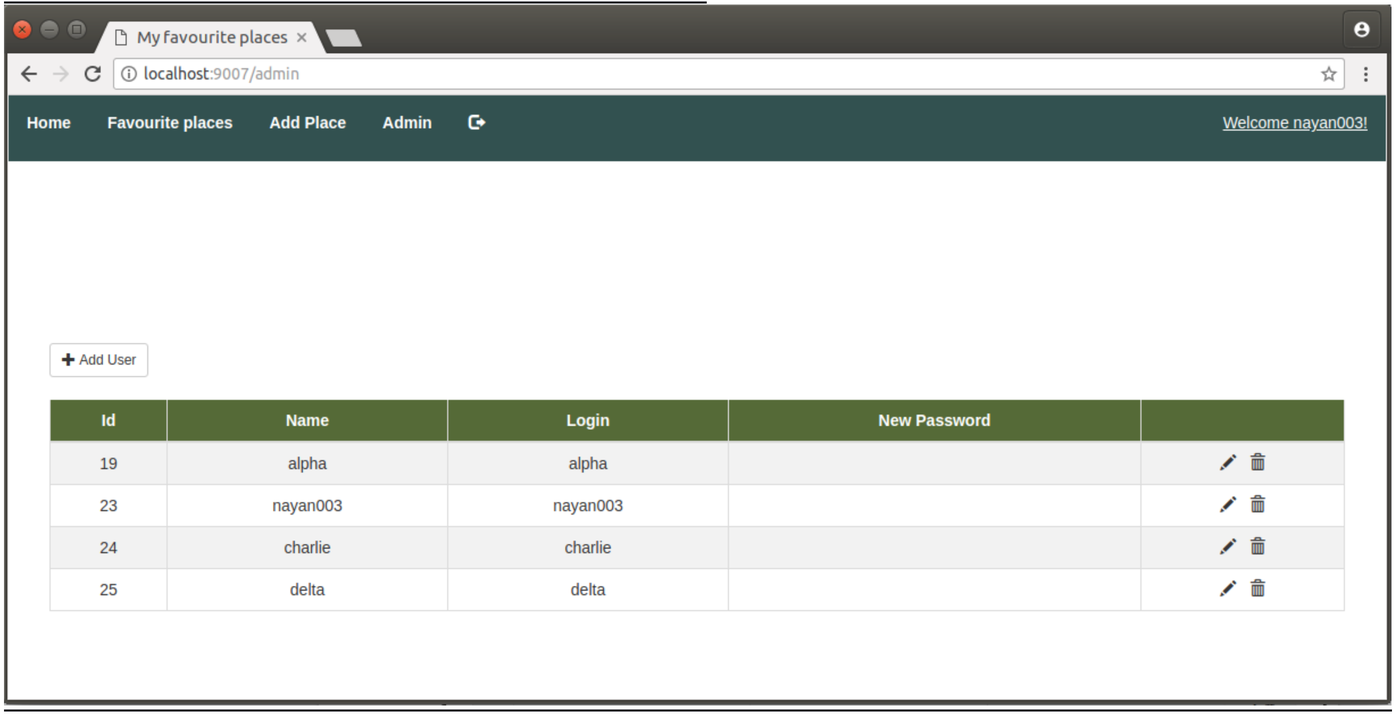
Admin enters the details of a new user with login "charlie" and clicks ***Save.*** *Error message is displayed because a user with login "charlie" (id 24) already exists.*



This login is used by another user

Id	Name	Login	New Password	
19	alpha	alpha		
23	nayan003	nayan003		
24	charlie	charlie		
25	delta	delta		
	charlie	charlie	charlie	

*Admin clicks on **Cancel** button to remove the new row.*



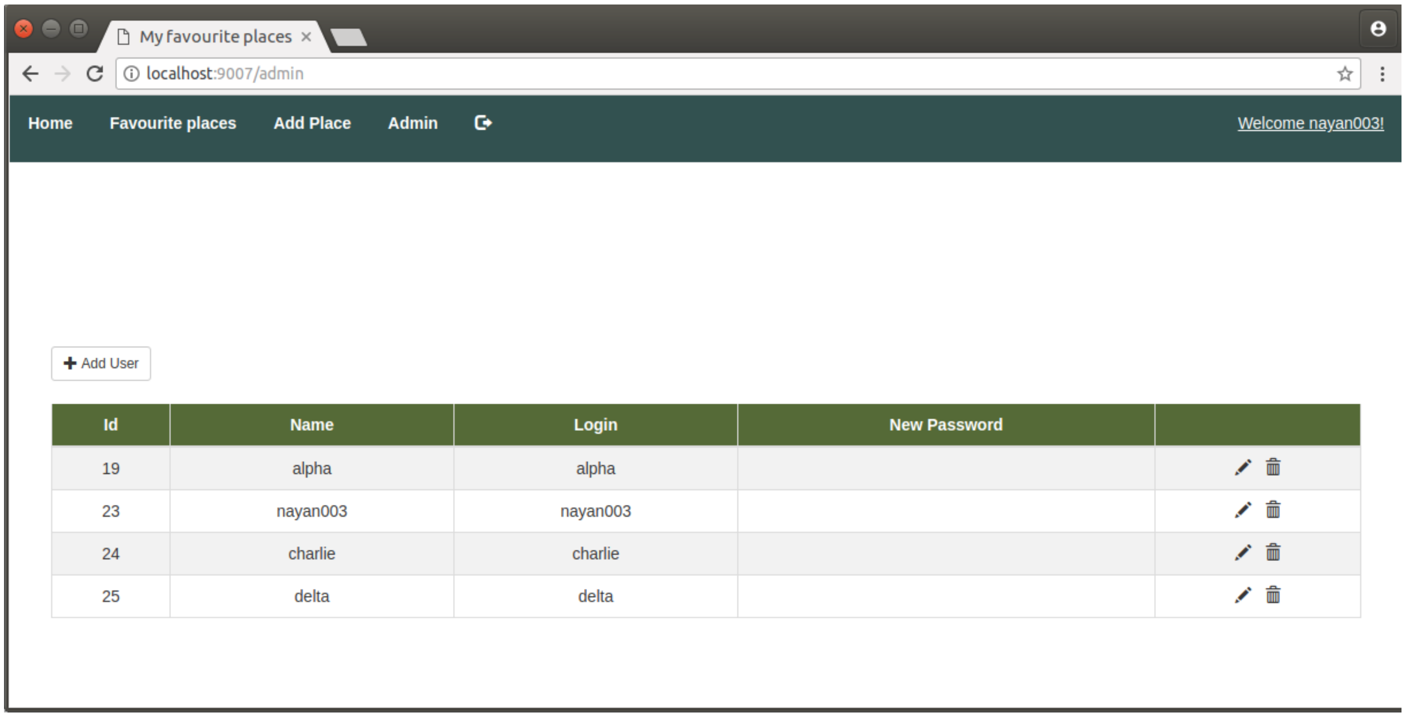
Id	Name	Login	New Password	
19	alpha	alpha		
23	nayan003	nayan003		
24	charlie	charlie		
25	delta	delta		

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 table displayed only if the user being deleted is not the one currently logged in. To achieve this, your server should provide a delete API which can be used by the client.
- The system should not allow a user to delete a user that is currently logged in. The following error message should be displayed if the user attempts to delete themselves: Can not delete the user that is logged in

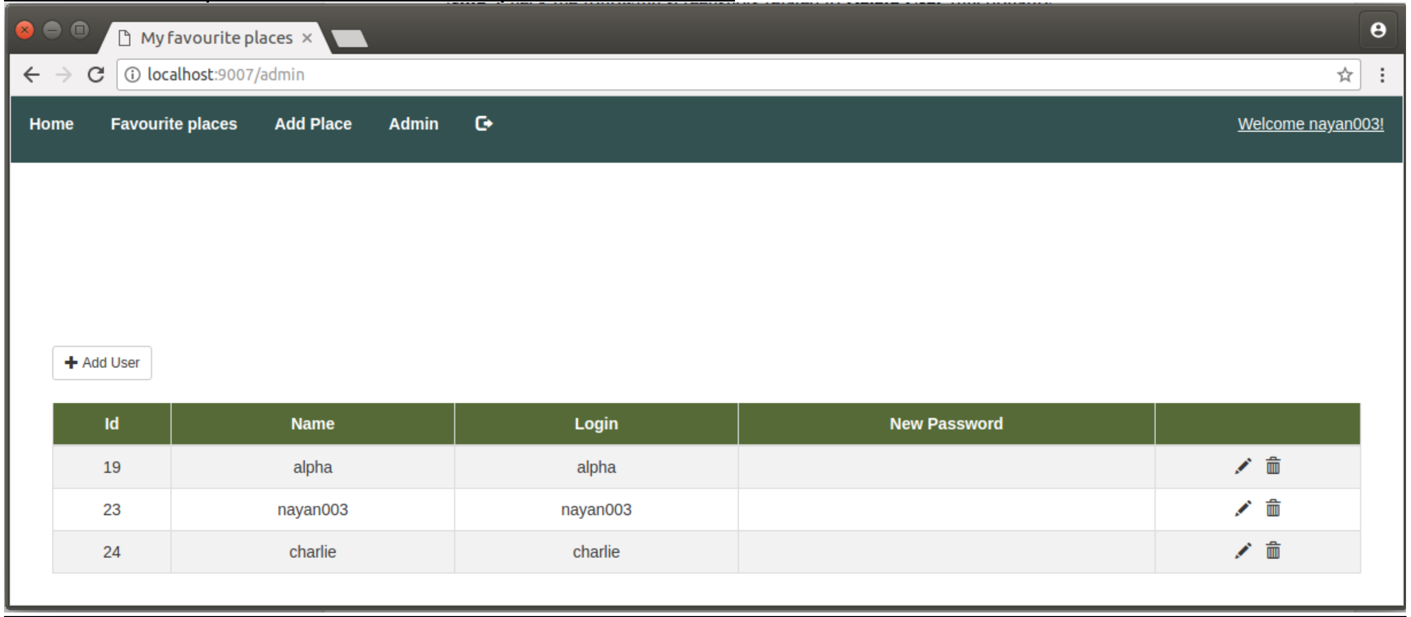
Note: Check the following screenshots depicting the **Delete User** functionality.

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



Id	Name	Login	New Password	
19	alpha	alpha		
23	nayan003	nayan003		
24	charlie	charlie		
25	delta	delta		

delta is deleted from the database and from HTML table.



My favourite places

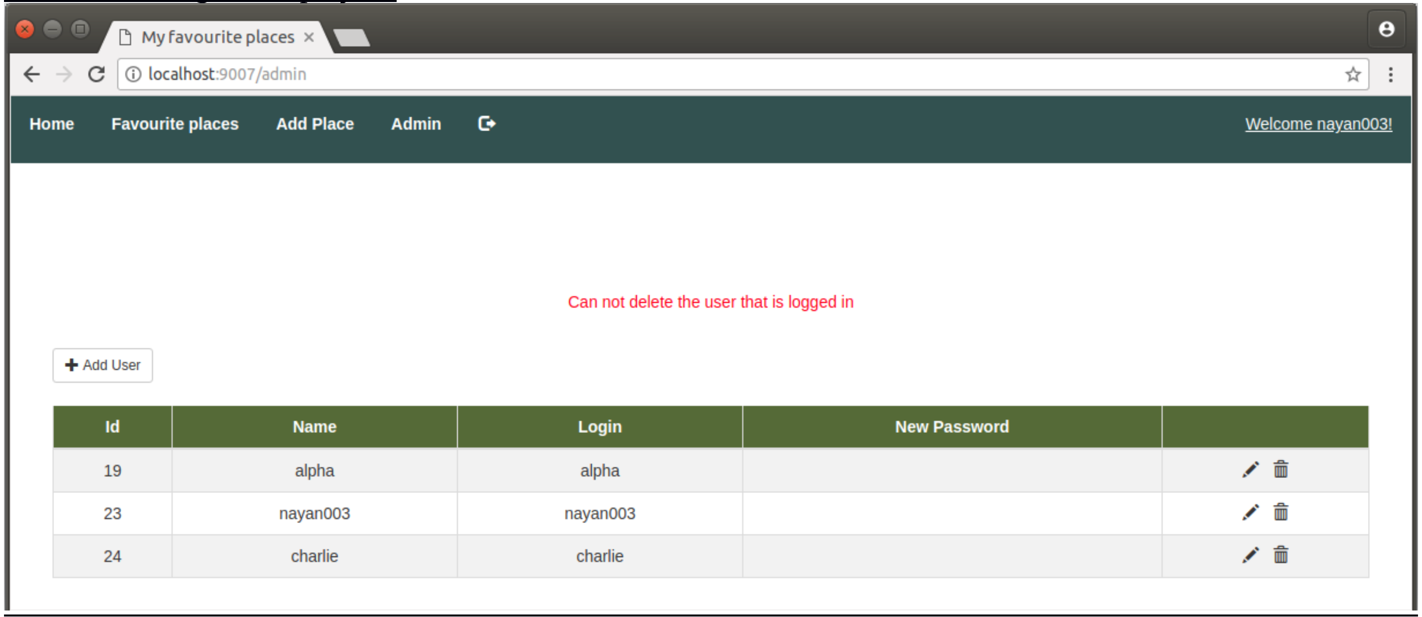
localhost:9007/admin

Home | Favourite places | Add Place | Admin | Welcome nayan003!

+ Add User

Id	Name	Login	New Password	
19	alpha	alpha		
23	nayan003	nayan003		
24	charlie	charlie		

Admin tries to delete the user nayan003 which is currently logged in. The user is not deleted and an error message is displayed.



My favourite places

localhost:9007/admin

Home | Favourite places | Add Place | Admin | Welcome nayan003!

Can not delete the user that is logged in

+ Add User

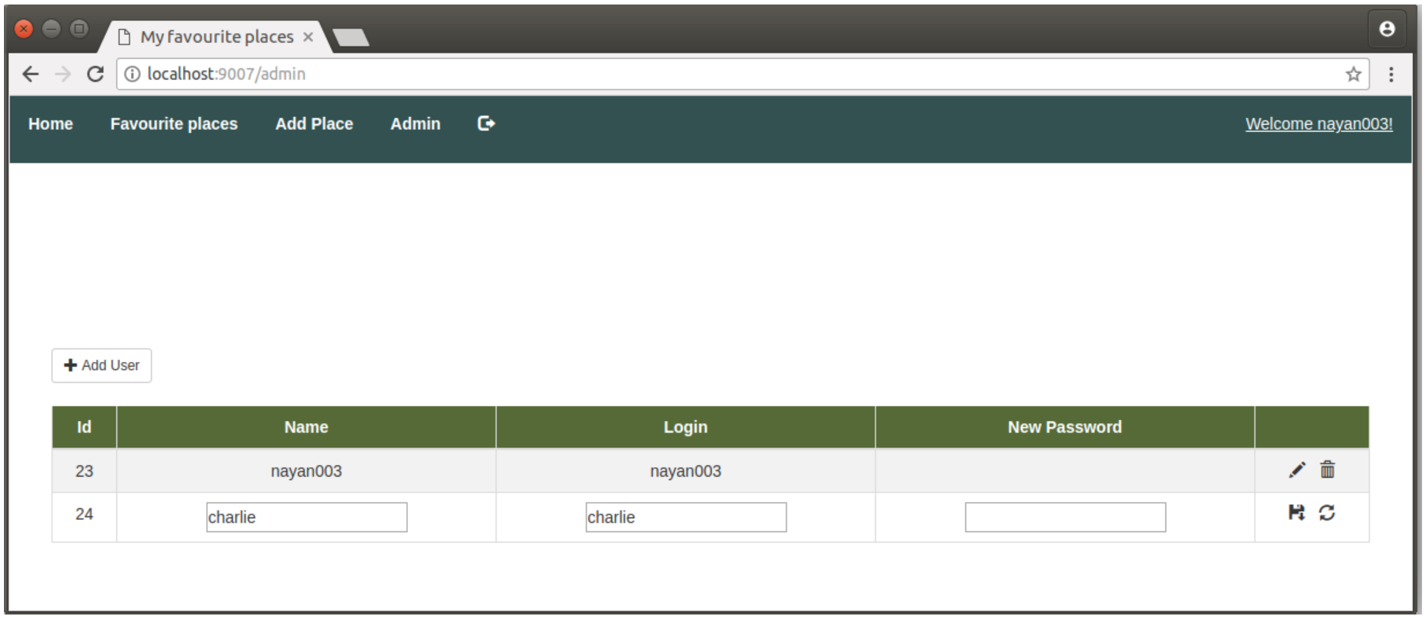
Id	Name	Login	New Password	
19	alpha	alpha		
23	nayan003	nayan003		
24	charlie	charlie		

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 it in the “list of 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.
    - If the validation succeeds, the newly entered information about the user being edited 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: This login is used by another user

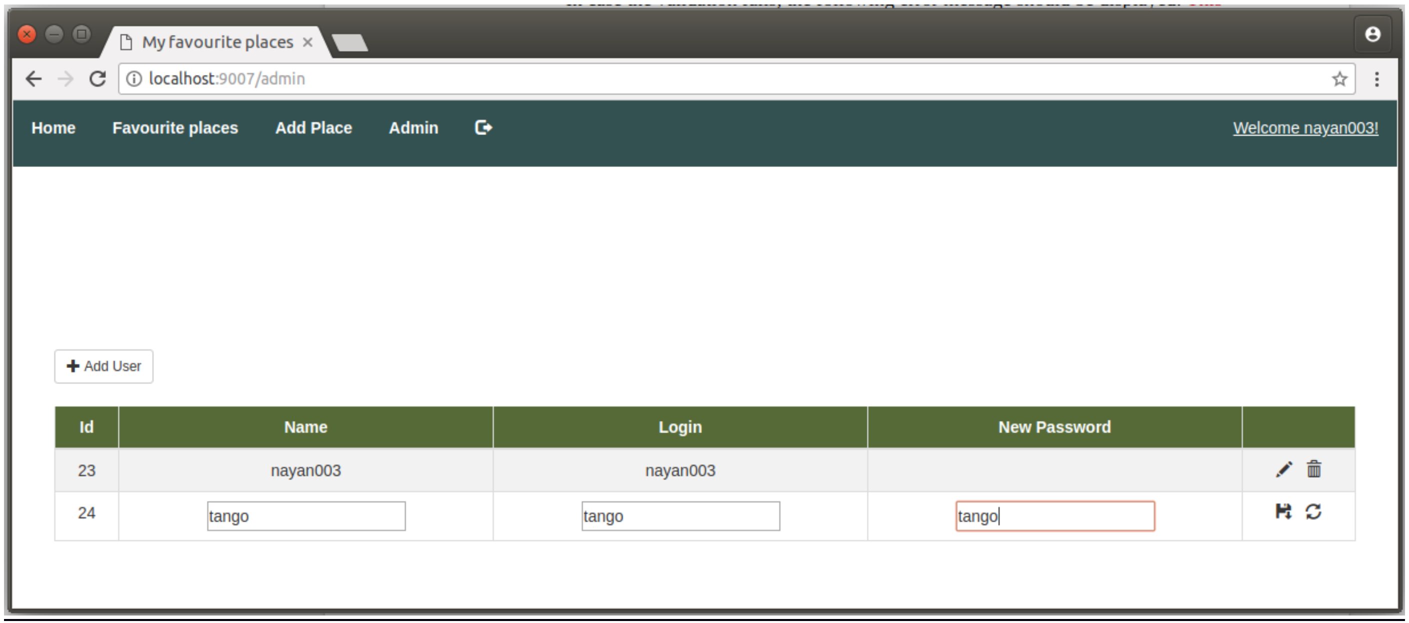
<u>**Note**</u>: Please review the following screenshots depicting the **Edit User** functionality.

<u>*Admin clicks on the edit button against the user charlie.*</u>



Id	Name	Login	New Password	
23	nayan003	nayan003		
24	charlie	charlie		

Admin updates the values and clicks on save.



My favourite places

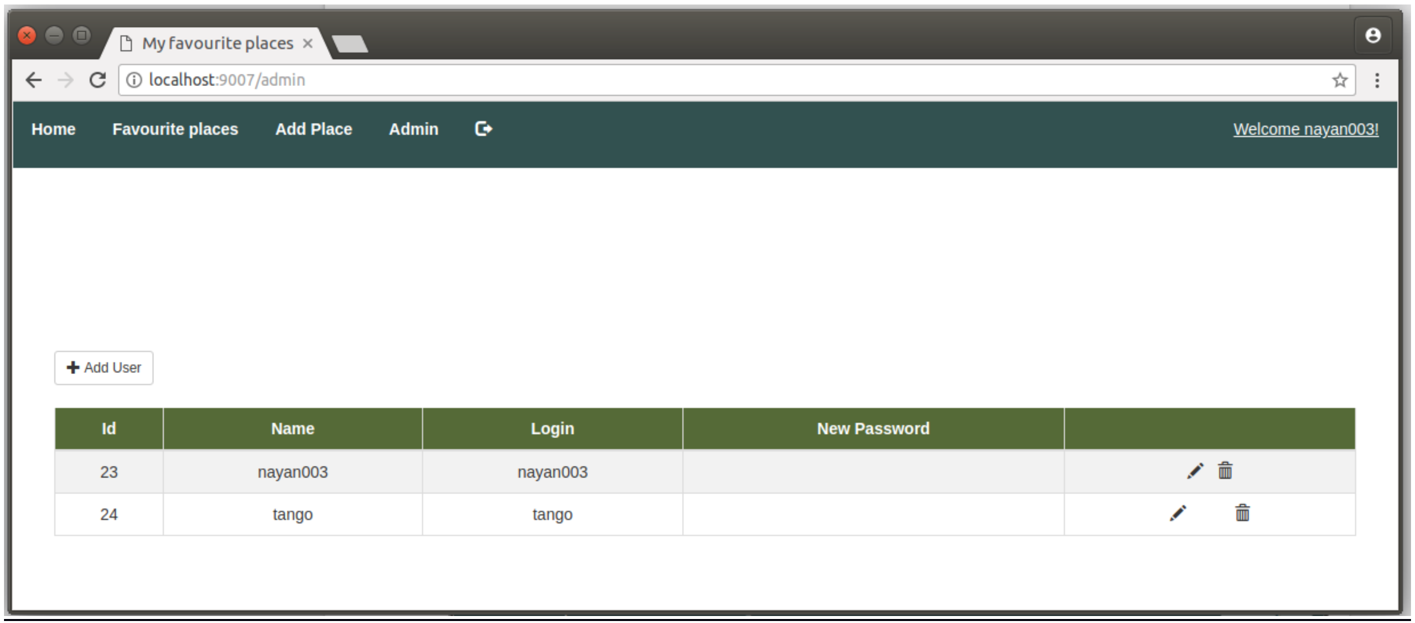
localhost:9007/admin

Home | Favourite places | Add Place | Admin | Welcome nayan003!

+ Add User

Id	Name	Login	New Password	
23	nayan003	nayan003		
24	tango	tango	tango	

Values are successfully updated in database and the same is reflected in HTML table.



My favourite places

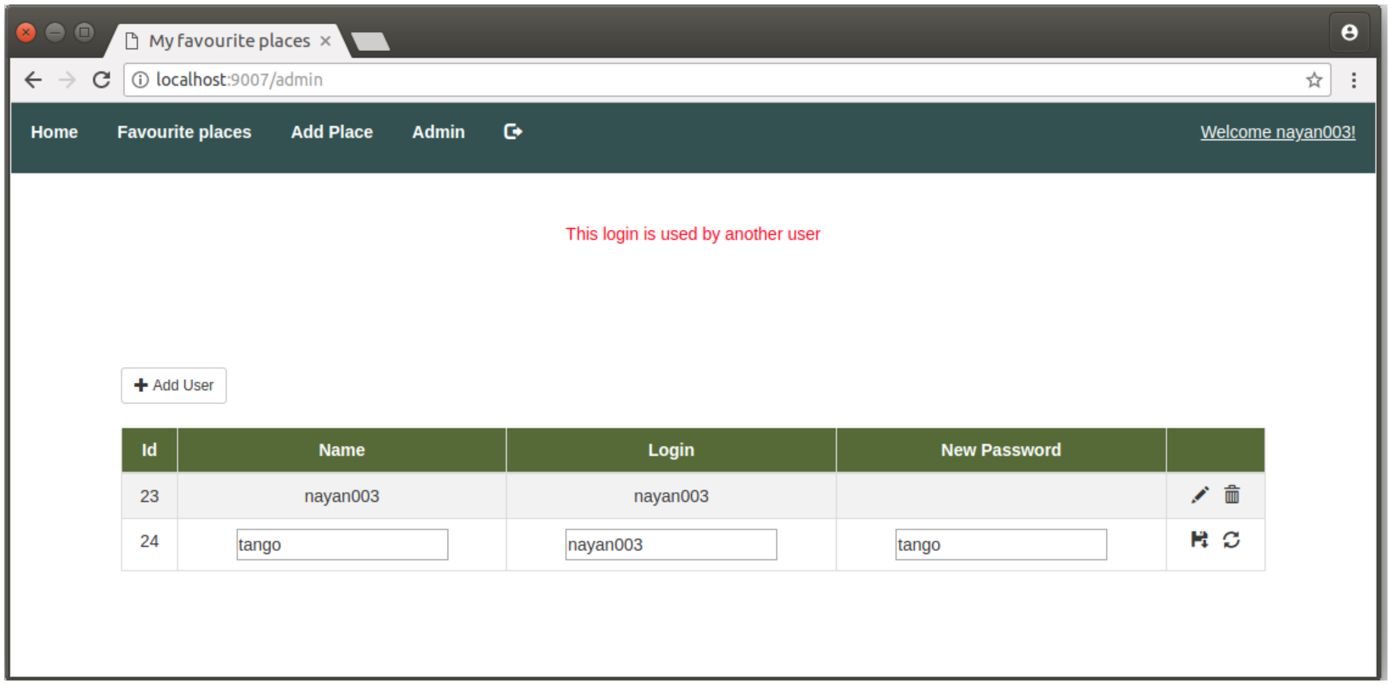
localhost:9007/admin

Home | Favourite places | Add Place | Admin | Welcome nayan003!

+ Add User

Id	Name	Login	New Password	
23	nayan003	nayan003		
24	tango	tango		

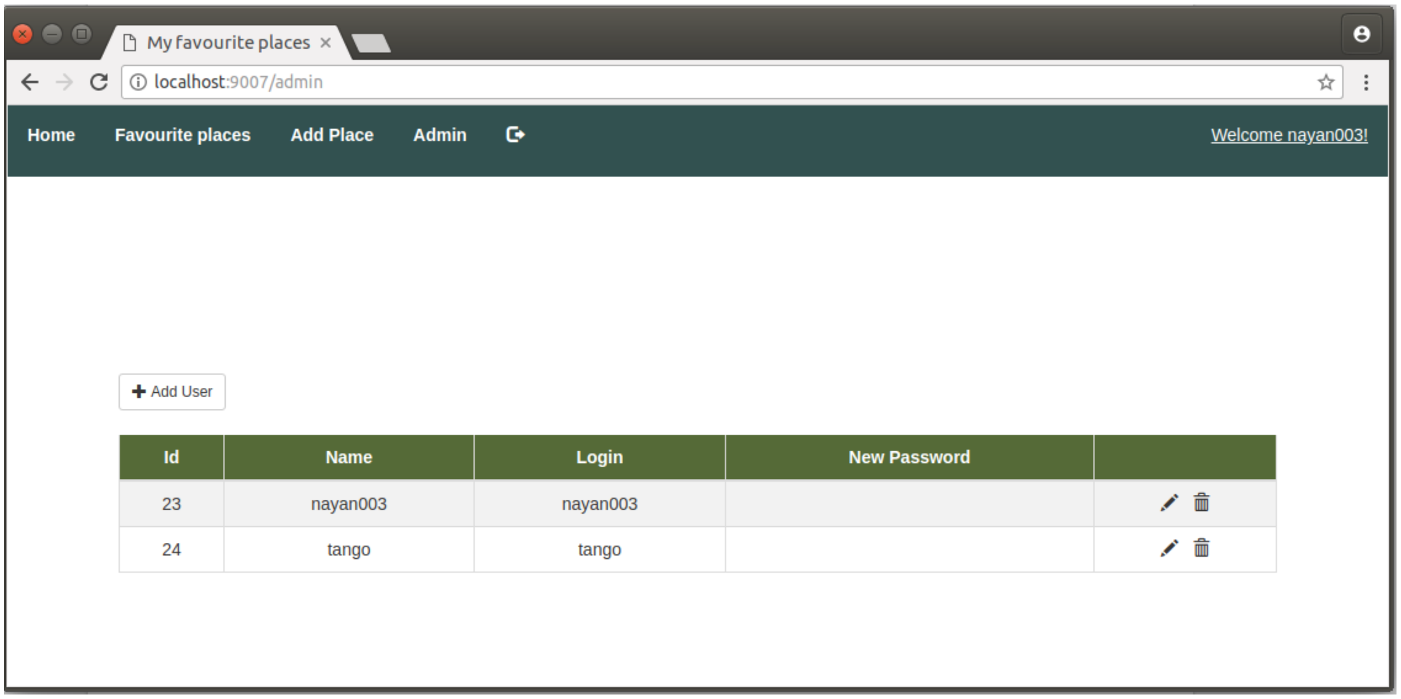
An Admin edits the details of user tango and updates the login to nayan003. An error message is displayed because a user with login nayan003 already exists.



This login is used by another user

Id	Name	Login	New Password	
23	nayan003	nayan003		
24	tango	nayan003	tango	

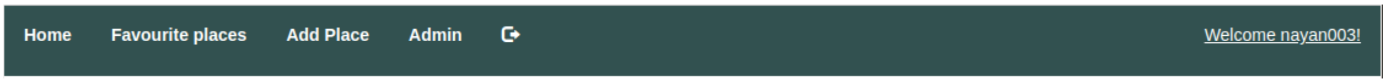
*<u>Admin clicks on **Cancel** button to discard the changes.</u>*



Id	Name	Login	New Password	
23	nayan003	nayan003		
24	tango	tango		

Navigation bar

- The Navigation bar should display the login id associated with the user that is currently logged in.



Database Configuration

- Your server should read the values (that you insert) in the file **`sample_dbconfig.xml`** file to establish database connection.

4 Submission Instructions

PLEASE ENSURE TO TEST YOUR CODE ON CSE LAB MACHINES.

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

Delete the **node_modules** folder present inside the project directory (**yourx500id_hw07**).

Create a **README** file inside **yourx500id_hw07** directory. This README file should include the following information for your website:

- **`Your x500id,`**
- **`login, and`**
- **`password.`**

Compress the **yourx500id_hw07** directory and submit it (use tar or zip)**.**

We will use the login and password values provided by you in README file to login to your website. Please ensure that these values are correct and can be used to access your website or we may have to assign you a grade of Zero.

5 Evaluation

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

- Submission instructions are met (5 points).
- Navigation bar displays the currently logged user. (5 points)
- Admin page displays the correct list of users in the system. (10 points)
- In Admin page, the DELETE button works for every row and displays error messages correctly. (10 points)
- In Admin page, the EDIT button works for every row and switches the view to EDIT mode as specified in this assignment. (10 points)
- In Admin page, UPDATE button works in EDIT mode. It validates the input, and updates the name, login, password as specified in this assignment. (10 points)
- In Admin page, CANCEL button works in EDIT mode. (5 points)
- In Admin page, ADD USER works and switches the view to ADD mode. (10 points)
- In Admin page, SAVE button works in ADD mode. It validates the input, and saves the name, login, password accordingly. (10 points)
- In Admin page, CANCEL button works in ADD mode. (5 points)
- Server reads database configuration from XML file and establishes connection to database. (10 points)
- All functionality from assignment 6 works. (10 points)

If you need help getting your HW 6 functioning properly, come to an office hour as soon as possible.