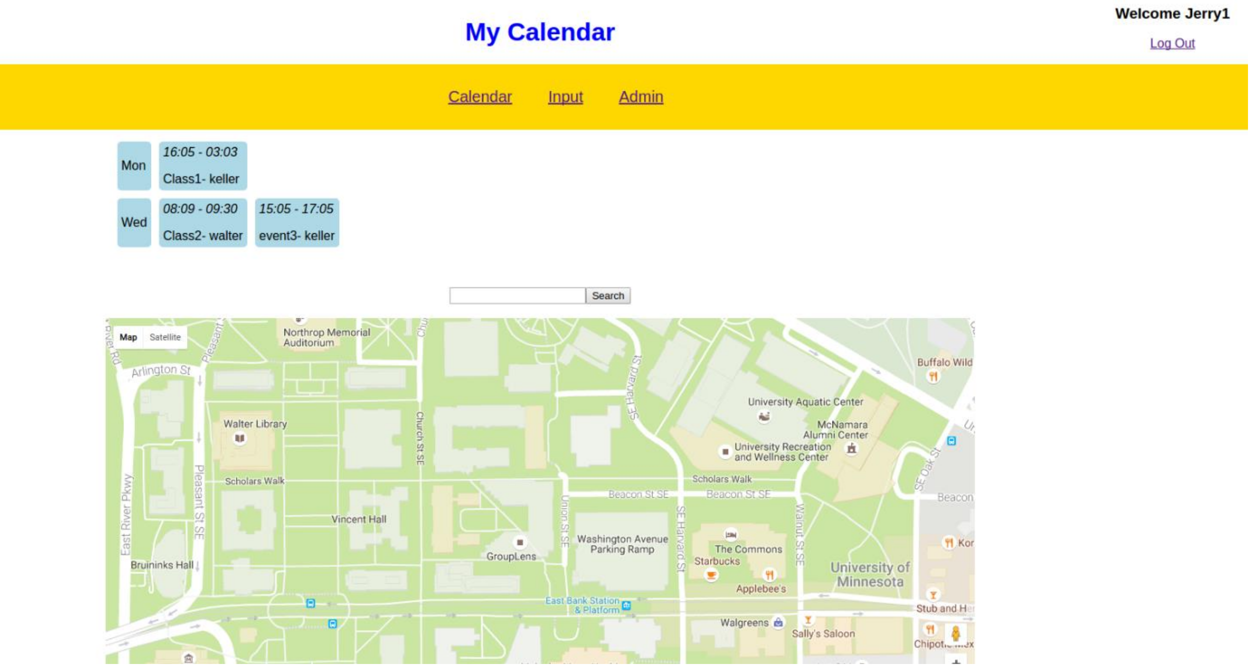
CSci-4131 Internet Programming
Fall 2016
Assignment 9

Issued: Monday November 28th
Due Date and Time: ***Monday , December 12th at 11:55pm***
Late Due Date: ***12:00 noon, Friday, December 16th at 11:55am***

Task:
In this assignment you will continue adding server side functionality to Assignment 8 using PHP and MySQL. Your task is to upgrade previous assignment to add a user Administration page with user management features **Edit, Update, Add, and Delete** to your Website.

Functionality

- Retain all the functionality from Assignment 8 and build on existing code – your login page from assignment 8 should be the first page a user trying to login to your website should see and interact with.
- Create an admin page (admin.php) linked to calendar page and input page.



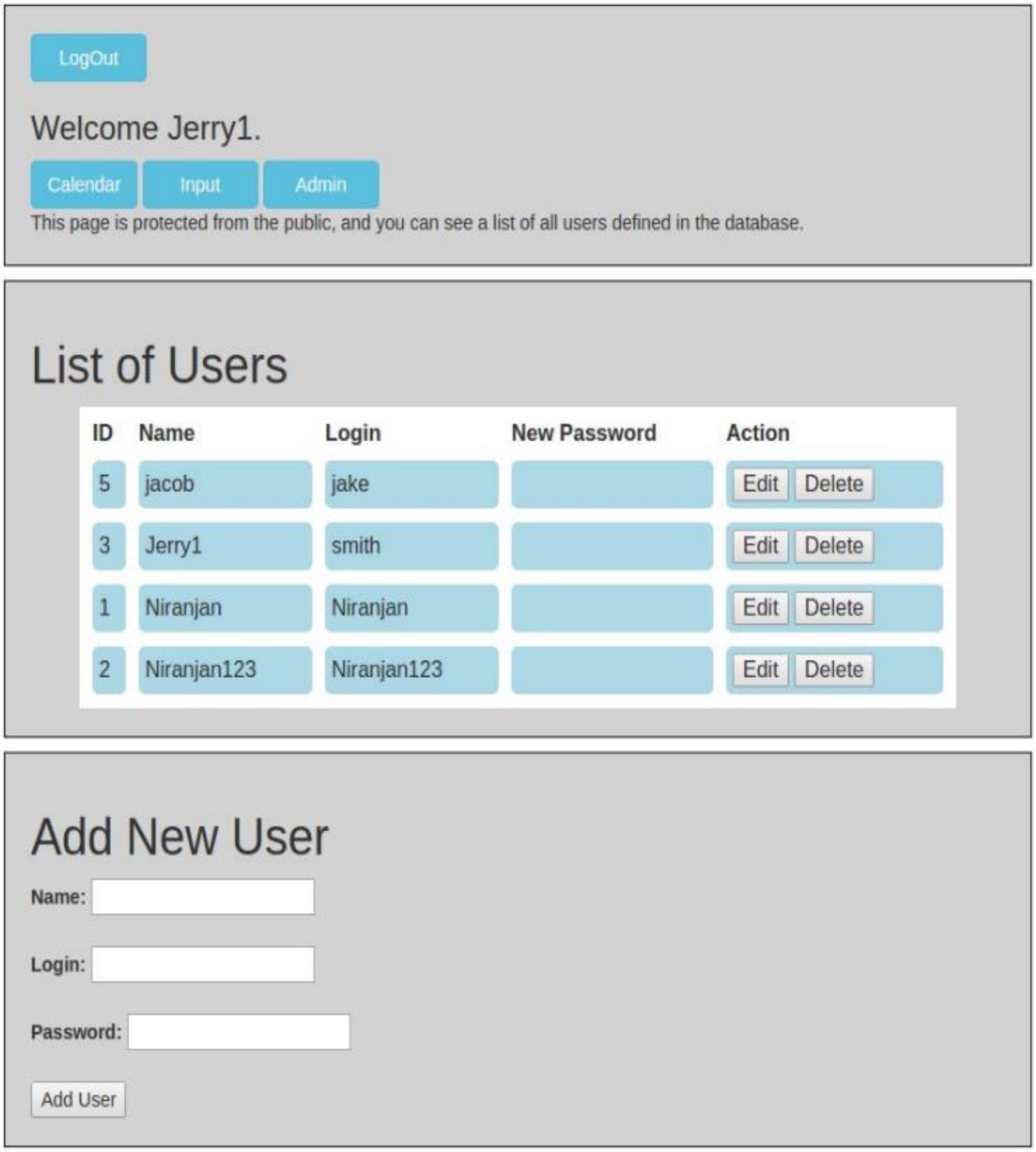
My Calendar

Welcome Jerry1

Log Out

Mon	16:05 - 03:03 Class1- keller	
Wed	08:09 - 09:30 Class2- walter	15:05 - 17:05 event3- keller

- The `Admin` page should display list of users, as illustrated in the screenshot on page 2, with their corresponding `Id`, `Name`, and `Login` id's. The values are stored in each row of the `tbl_accounts` table in the `acc_id, acc_name, acct_login` and `acct_password` columns.



Welcome Jerry1.

This page is protected from the public, and you can see a list of all users defined in the database.

List of Users

ID	Name	Login	New Password	Action
5	jacob	jake		Edit Delete
3	Jerry1	smith		Edit Delete
1	Niranjan	Niranjan		Edit Delete
2	Niranjan123	Niranjan123		Edit Delete

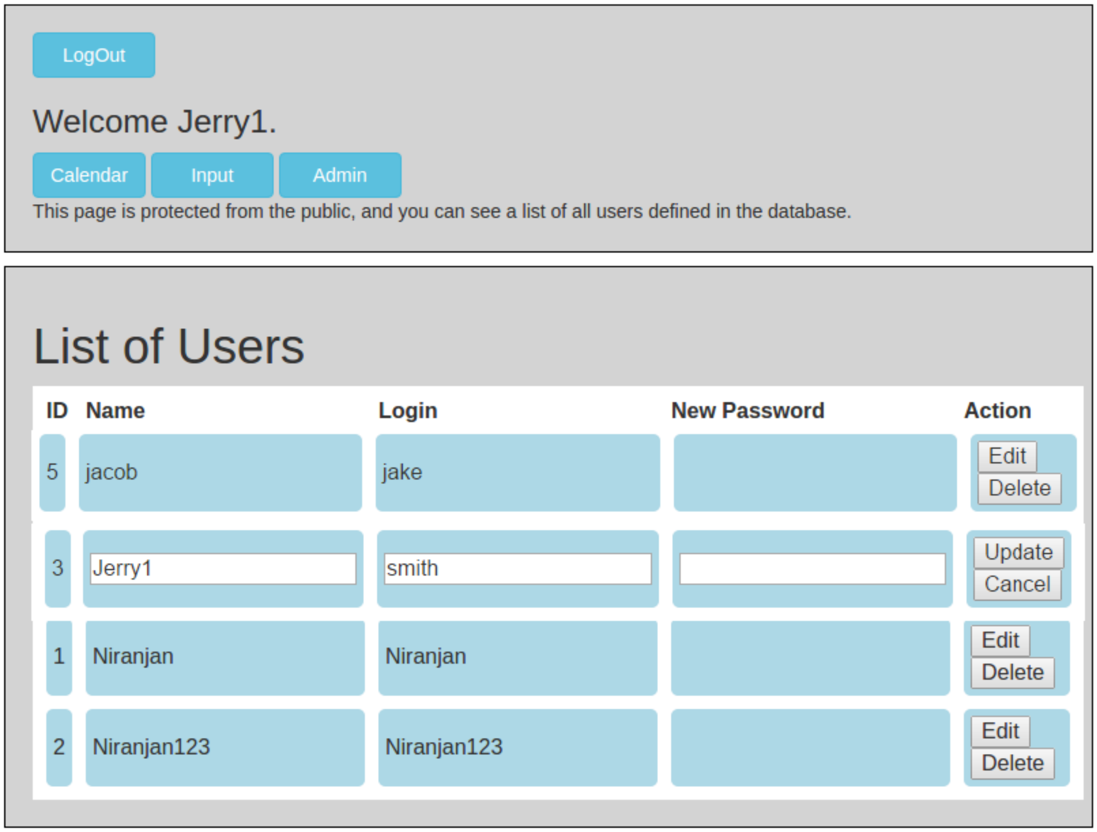
Add New User

Name:

Login:

Password:

- The `Edit` button will activate edit mode, and display it in the **List of Users** table. You should be able to update the user name, login, and password in edit mode. Consider the user **Jerry1** depicted in the screenshot above. If the `Edit` button is selected, the table will be updated and the table in the screenshot at the top of the next page will be displayed.
- Edit mode will have two different buttons: **`Update`** and **`Cancel`**. **`Cancel`** should discard the changes and exit edit mode. The **`Update`** button should validate the input, check to make sure that no other user in the database has the same login, and use SHA1 to hash the password and store it (via an update) in the database. If the update is not successful, your program should print an error message and return to edit mode. Finally, if the update is successful your program should exit edit mode, display the list of users, and display a message that user has been successfully updated. This is depicted in the screenshots on page 3, which change the user Jerry1 to ***Jerry***.

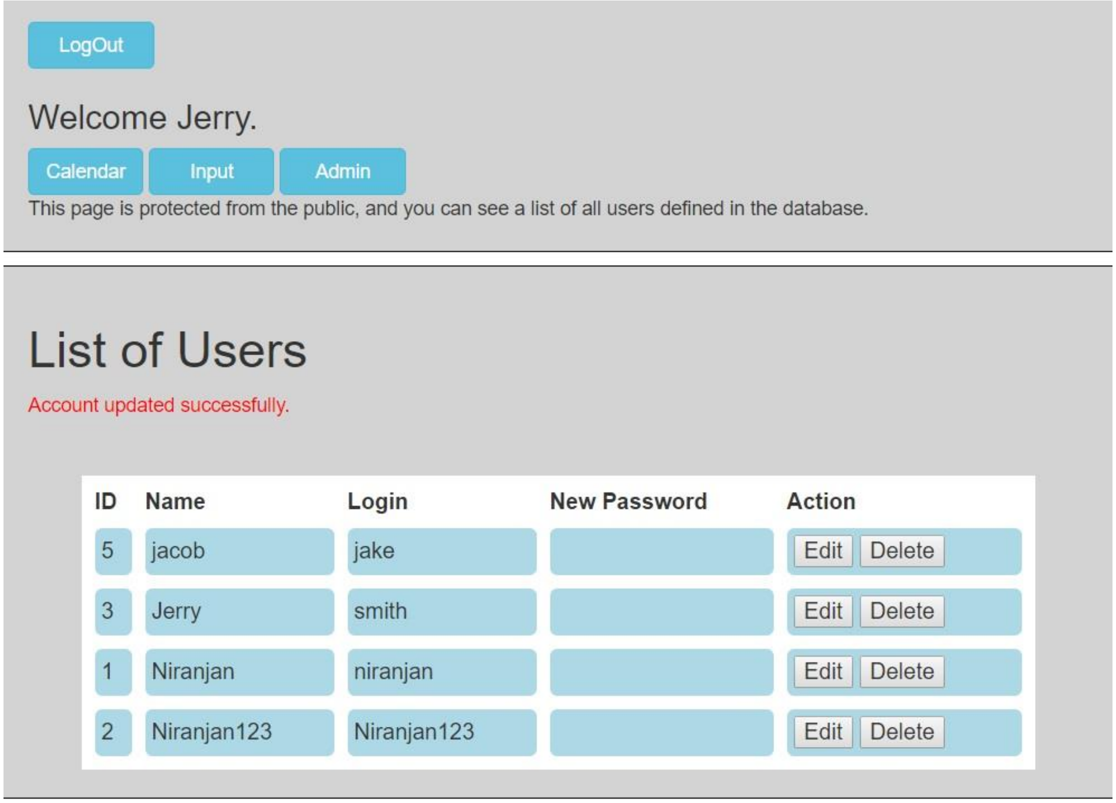


Welcome Jerry1.

This page is protected from the public, and you can see a list of all users defined in the database.

List of Users

ID	Name	Login	New Password	Action
5	jacob	jake		Edit Delete
3	Jerry1	smith		Update Cancel
1	Niranjan	Niranjan		Edit Delete
2	Niranjan123	Niranjan123		Edit Delete



Welcome Jerry.

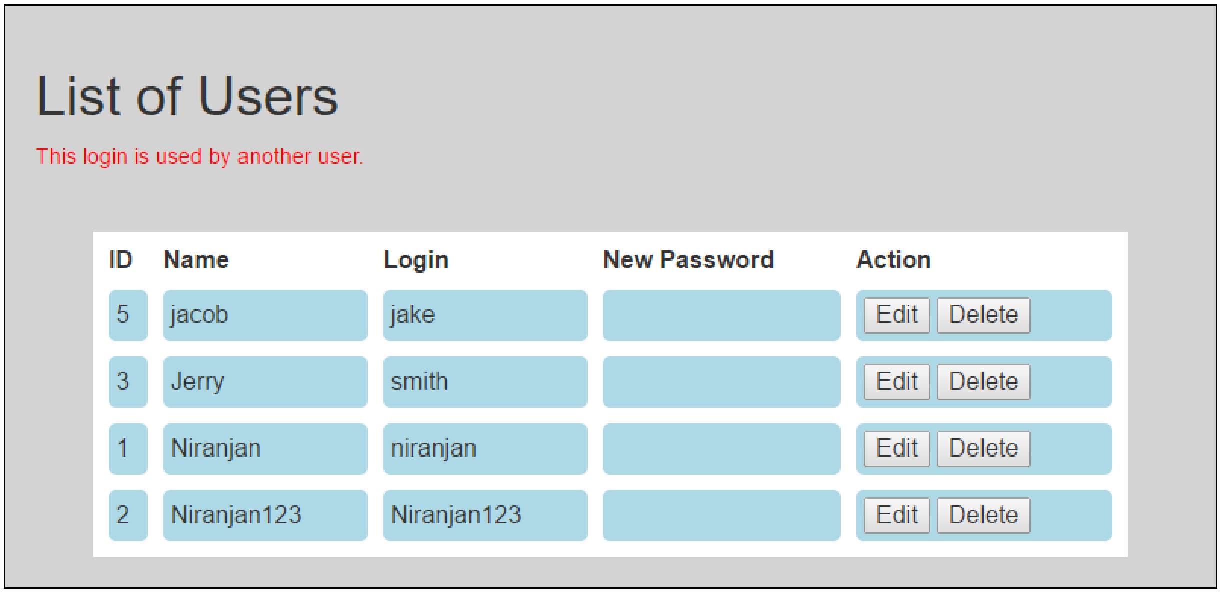
This page is protected from the public, and you can see a list of all users defined in the database.

List of Users

Account updated successfully.

ID	Name	Login	New Password	Action
5	jacob	jake		Edit Delete
3	Jerry	smith		Edit Delete
1	Niranjan	niranjan		Edit Delete
2	Niranjan123	Niranjan123		Edit Delete

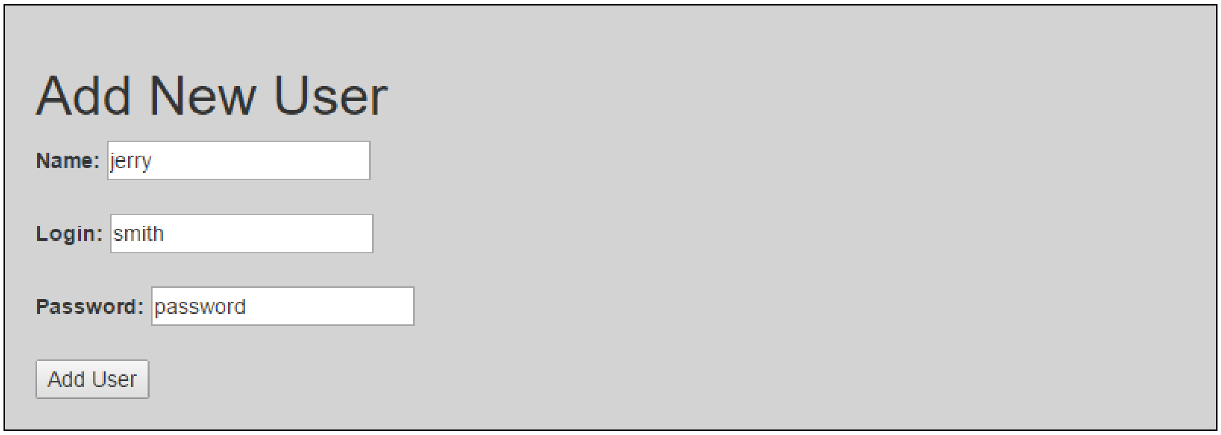
- The Admin page also has a new section: **Add New User**. If you press the **`Add User`** button, the controller should to validate the input, check to ensure no other user in the database has the same login, and use SHA1 to hash the password and add the new user information, including the hashed password, to the database. It should display a message if the new user is successfully added (or if an error occurs). An error scenario is depicted in the screenshot below, and a scenario that is successful is depicted on the top of Page 5.



List of Users

This login is used by another user.

ID	Name	Login	New Password	Action
5	jacob	jake		Edit Delete
3	Jerry	smith		Edit Delete
1	Niranjan	niranjan		Edit Delete
2	Niranjan123	Niranjan123		Edit Delete

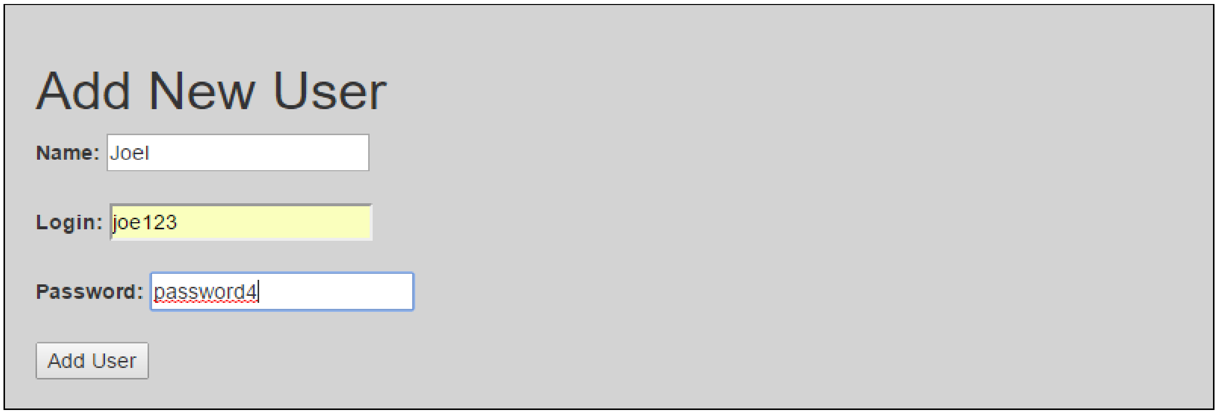


Add New User

Name: jerry

Login: smith

Password: password

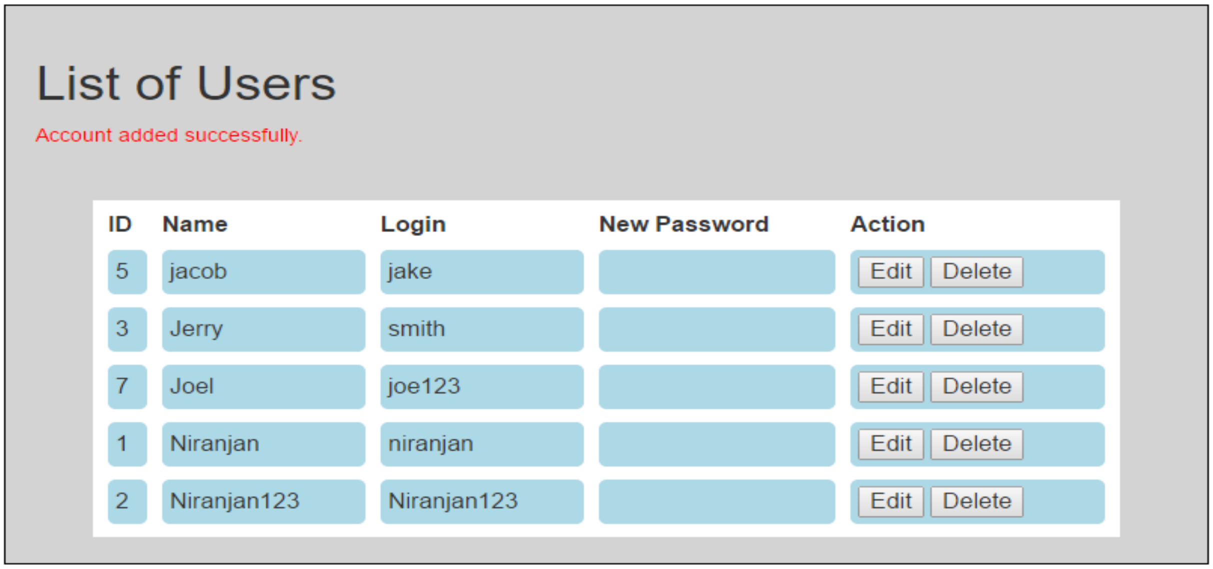


Add New User

Name: Joel

Login: joe123

Password: password4

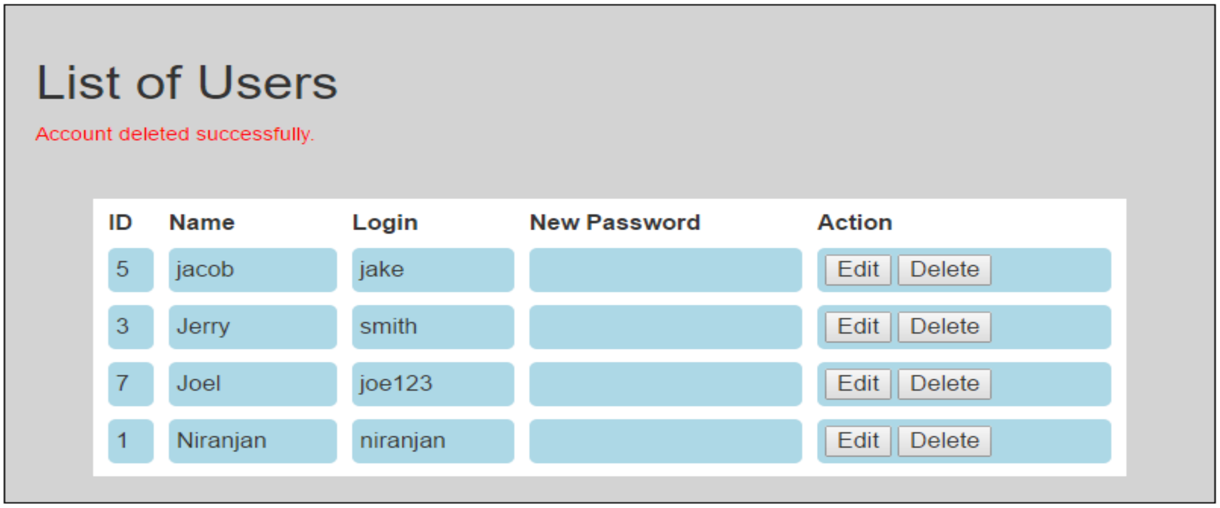


List of Users

Account added successfully.

ID	Name	Login	New Password	Action
5	jacob	jake		Edit Delete
3	Jerry	smith		Edit Delete
7	Joel	joe123		Edit Delete
1	Niranjan	niranjan		Edit Delete
2	Niranjan123	Niranjan123		Edit Delete

- The `Delete` button should delete the user from the database, and display a message indicating the operation was successful. The screenshot below depicts the result when the user Niranjan123 is deleted successfully.



List of Users

Account deleted successfully.

ID	Name	Login	New Password	Action
5	jacob	jake		Edit Delete
3	Jerry	smith		Edit Delete
7	Joel	joe123		Edit Delete
1	Niranjan	niranjan		Edit Delete

- One important requirement for input validation for this assignment is that no two different users can have same login. Your code should check this before adding or updating users and show proper message should that occur (something like, the password you have selected is already in use).

Using a Model View Controller (MVC) pattern in your solution – 25 Points
In MVC, you have to separate the Model from the View and Controller . There are a variety of ways to accomplish this, but the easiest will entail either implementing one controller that handles every page, or alternately, you can create one controller for each page. Creating a separate controller for each page is a little bit easier to understand and simpler to implement, but either approach will work.
The key is to ensure your program separates the View and the Model (the Model is the database and the routines associated with initializing it, updating, inserting, and deleting from it). The View should not directly update the database in any case. See Chapters 4 and 5 in the textbook PHP and MySQL by Murach for an example of using the MVC pattern with PHP (and MySQL). To obtain these points, at a minimum, the Model should be separated from the view and the controller.

Submission Instructions. You should submit all the files required to execute and test your program in a tar file. Your files may look something like this:

- database.php: Your MySQL database credentials.
- Calendar.php – your calendar page(s) and functionality from previous assignments
- Form.php – your form page(s) and functionality from previous
- users.php: Users Controller
- login.php: Login Controller
- logout.php: Log out page.
- models/account.php: Account Model
- models/users.php: Users Model
- views/login view.php: Login View
- views/users view.php: Users View
- style.css: (optional)
- A README file with your login and password and any additional instructions

Grading Criteria (200 points):

- admin.php page displays the appropriate User name at the top - 10 points
- admin.php page displays correct list of users in the system - 10 points
- admin.php page EDIT button works for every row - 20 points
- admin.php page DELETE button works for every row - 10 points
- admin.php page UPDATE button works for every row, validates the input, and updates the user name, login, and password accordingly - 20 points
- admin.php page CANCEL button works for every row - 5 points
- admin.php page Add New User works and displays error messages correctly - 15 points
- You can navigate between the pages of your website and not have to re-login to use the admin page. – 10 points
- All functionality from the Calendar Page from previous assignments work correctly (Search on Google Maps, Calendar is displayed correctly from calendar.txt file) – 35 points
- All functionality on the Input Form on the Input Page from previous assignments works correctly (new calendar events, or completely clear the calendar – 35 points
- Model separated from view and controller, encapsulated in a Php Class (25 points)
- Logout functionality works correctly - 5 points

Note: Failure to correctly follow the submission instructions may result in a grade of ZERO for the assignment.
Also, if your code fails to execute correctly on the CSELabs computers, we will deduct points accordingly.