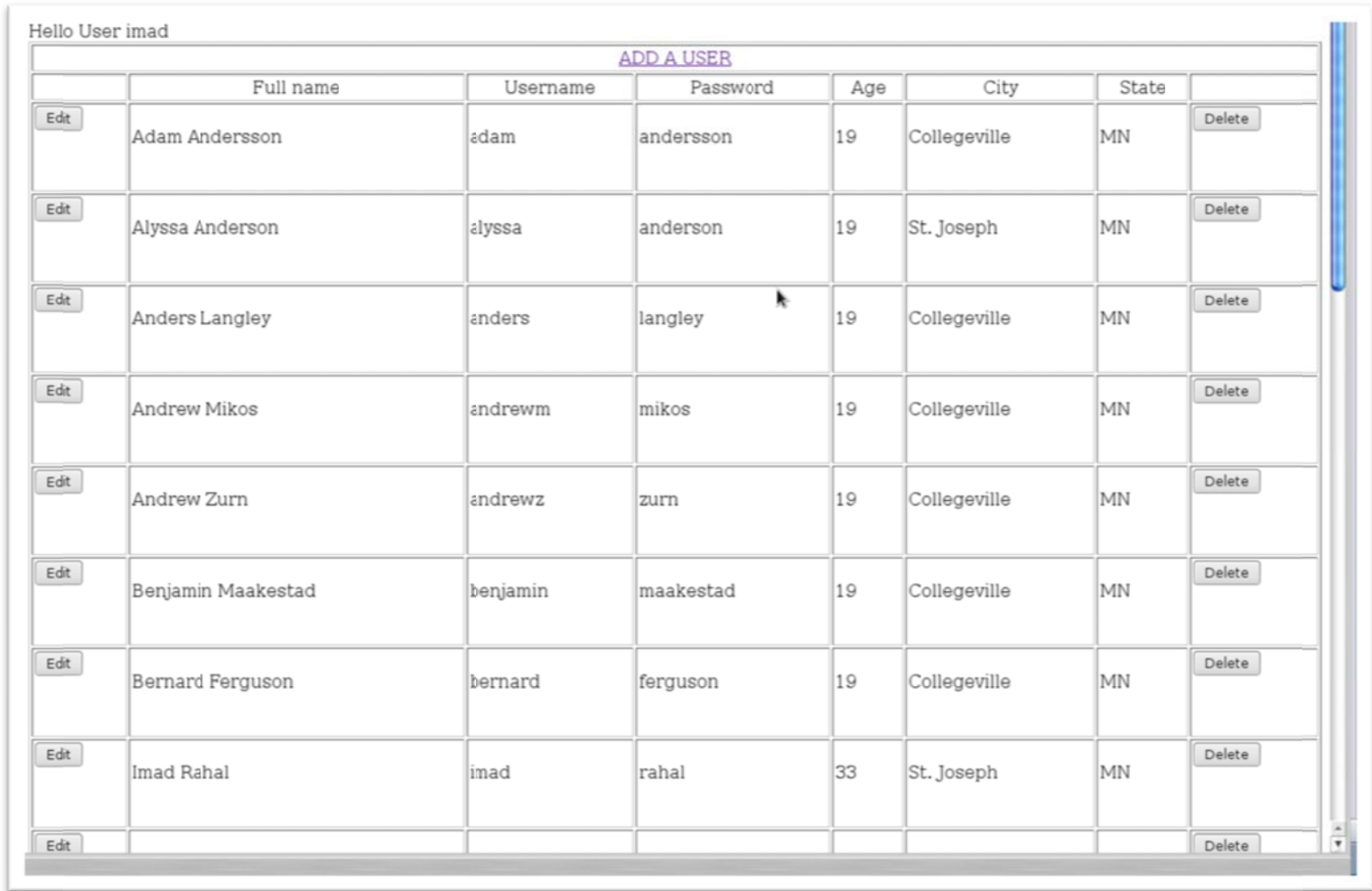


Task 3: Completing page Menu.jsp

When complete, page `Menu.jsp` will display something like the figure below which starts with a greeting for the logged-in user ("*Hello User imad*" in my case) followed by a table.

The table's first row (which spans 8 columns) contains a hyperlink to page `Add.jsp` which allows the user to add a new user. The second row contains 8 cells, the first and last of which are empty. The rest of the cells contain headers for the corresponding columns.



Hello User imad							
ADD A USER							
	Full name	Username	Password	Age	City	State	
Edit	Adam Andersson	adam	andersson	19	Collegeville	MN	Delete
Edit	Alyssa Anderson	alyssa	anderson	19	St. Joseph	MN	Delete
Edit	Anders Langley	anders	langley	19	Collegeville	MN	Delete
Edit	Andrew Mikos	andrewm	mikos	19	Collegeville	MN	Delete
Edit	Andrew Zurn	andrewz	zurn	19	Collegeville	MN	Delete
Edit	Benjamin Maakestad	benjamin	maakestad	19	Collegeville	MN	Delete
Edit	Bernard Ferguson	bernard	ferguson	19	Collegeville	MN	Delete
Edit	Imad Rahal	imad	rahal	33	St. Joseph	MN	Delete
Edit							Delete

The rest of the rows contain user information from the database with one row for every user. The first cell contains a form with two fields: an *edit submit button* which submits to page `Edit.jsp` and a *hidden field* which contains the username of the corresponding user. The reason for the hidden field is that the target page `Edit.jsp` needs to know which user was clicked in order to edit the correct one. The last cell is pretty similar but its delete button should submit to `Delete.jsp` instead. The remainder of the cells contains the user information.

Page `Menu.jsp` should use the `UserConroller` session variable created in page `Login_action.jsp`. This will allow the page to get the username of the logged-in user and call method `getAllUsers()` in `UserConroller` in order to loop over all the user records and create rows for them.

Task 4: Completing pages Add.jsp & Add_action.jsp

Page `Add.jsp` contains an empty form which allows us to enter information for a new user and submits the form info to page `Add_action.jsp`. Page `Add_action.jsp` should then create a new `User` object using the submitted form information and use the `UserConroller` session variable to call method `addUser` by passing to it the newly created `User` object (*PS: this method will fail if the provided username is not unique*). Finally, page `Add_action.jsp` should then forward page to `Menu.jsp`.

Task 5: Completing page Delete.jsp

Page `Delete.jsp` should use the `UserConroller` session variable to call method `getSpecificUser(String username)` in `UserConroller` using the provided username. It should then create a new `User` object using the array returned by method `getSpecificUser` and pass the created the `User` object to method `deleteUser` in class `UserConroller`. The page should then forward page to `Menu.jsp`.

Task 6: Completing page Edit.jsp & Edit_action.jsp

Page `Edit.jsp` should show a user form similar to the one in `Add.jsp` but this time it will be a complete form containing information on the selected user. It should use the `UserConroller` session variable to call method `getSpecificUser(String username)` in `UserConroller` using the provided username. Page `Edit.jsp` will then load the user's information into the form for display. Since the username field is not editable, it has been marked as `readonly`. The edit form will submit the updated information to page `Edit_action.jsp` which should, in turn, create a new `User` object using the submitted form information and use the `UserConroller` session variable to call method `editUser` by passing to it the newly created `User` object. Finally, page `Edit_action.jsp` should then forward page to `Menu.jsp`.