William Wantuch, Liam Walsh, Alan Yao

CSC 2053 - 001

Project 3 Project Description and Team Members

Our project uses a login system and prompts a user to create a username and password, then brings them to the main blackjack game page. William used the Deck of Cards API online. Pressing the new game button calls the API and retrieves the deckID as well as drawing 2 cards for the player and 2 cards for the dealer, adding their card's code to the appropriate array. All of the buttons are disabled until the API finishes retrieving and displaying the cards. The images are then retrieved from the API by using the card's code. You have the option to hit, which calls the API and adds another card to the player's card array and displays that card on the screen. When the player reaches 21, the hit button becomes disabled. When the player stands, the dealer's hole card is turned over and he draws cards until he reaches 17, then his cards' value is compared to the value of the player's cards, displaying a statement based on the result. All of the buttons become disabled once the user clicks stand and the new game button becomes enabled once the API finishes its calls, which usually takes about 12 seconds. William encountered issues with the comparison functions being called before the API finished receiving its call, which caused incorrect conclusions. To fix this issue William added setTimeout functions so the program waits until the API call is completed.

Alan created the login system with PHP, HTML and MySql. The login system consisted of a login.html, register.html, authenticate.php, and register.php. All php pages automatically connect to the MySql database before anything else. Both the html pages were just forms for the user to input their credentials. After they were inputted, the php would automatically compare

the login inputs with the login database, and kick the user back to the login page if the username or password was wrong. For registration, if the username was already taken, then the user was also informed that the username was taken and redirected to the registration page. If registration was successful, then the username and the hashed password was uploaded to the MySql server. Once there was successful login, the user would automatically be redirected to the Blackjack main page.

The Blackjack and Leaderboard page both check whether or not there is a login session. If there isn't, they are kicked to the login page. Once there is a Blackjack player logged in, then they can start playing Blackjack. At the top of the page, their wins and losses are displayed, and pulled up through the MySql database, so that it isn't session specific. When you refresh the page, the wins and losses do not disappear. Wins and losses are automatically recorded in the MySql database when a player starts a new game through Ajax. Wins and losses are also automatically updated at the top of the Blackjack page without refreshing through passing the wins and losses to a Javascript variable and updating them each game.

The leaderboard page displays every player's wins and losses, and displays the player with the most wins down to the player with the least amount of wins. These wins and losses are pulled through the MySql database and printed out automatically in a table.

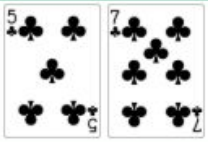
William created the blackjack code, which integrated the blackjack API. Liam created the CSS style sheets. Alan created the user login system, leaderboards, the navigation bar style sheet, and database using HTML, PHP, CSS and MySql.

BLACKJACK

Wins: 1 Losses: 2 Ties: 0

DEALER: 8

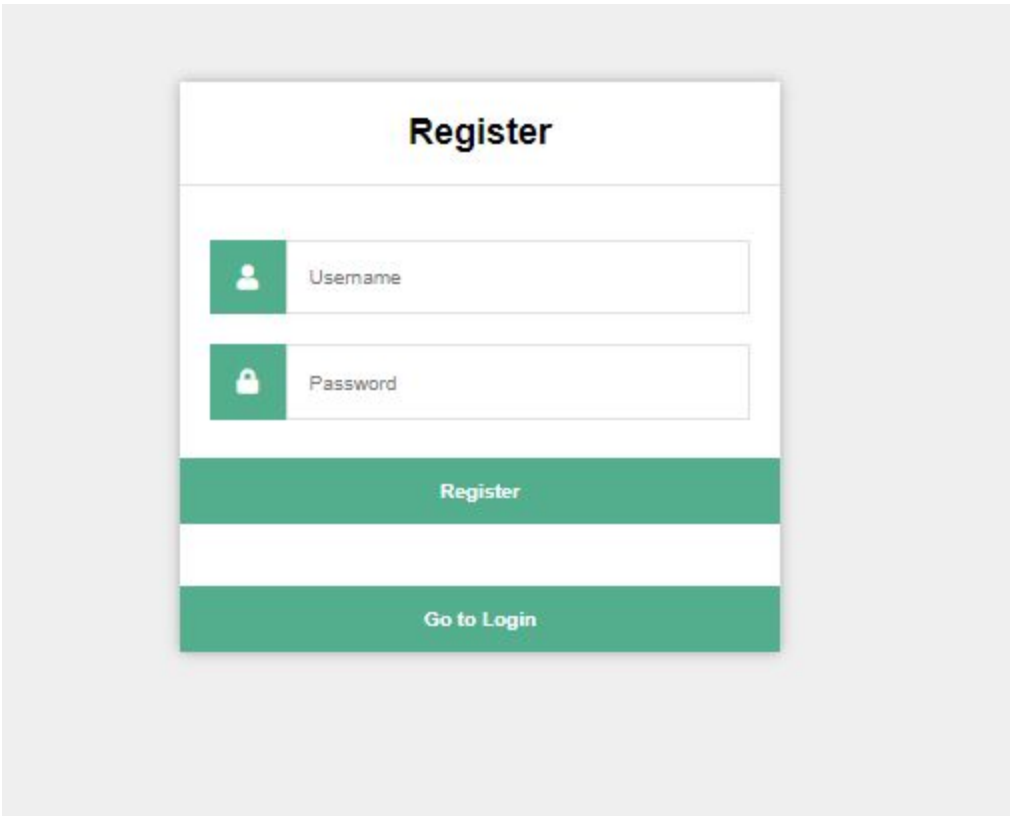
PLAYER: 12



HIT STAND

NEW GAME

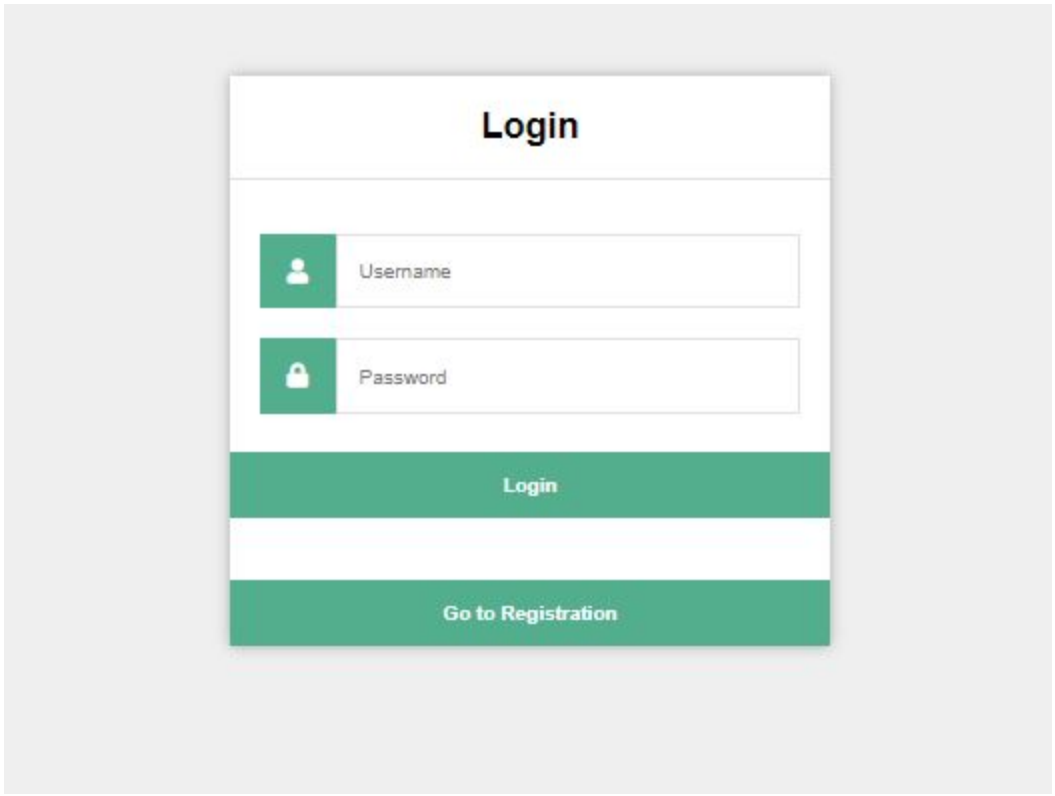
PROFILE LOGOUT



LOGOUT HOME

Welcome alan
You have: 1 wins, 2 losses, and 0 ties

Username	Wins	Loss	Tie
Bob	59	19	4
Liam	33	0	0
William	23	0	0
Wildcats	2	0	0
alan	1	2	0
bob1	1	0	0
bobb	1	1	0
Test	0	0	0
swag	0	1	0


Login
Username
Password
Login
Go to Registration