

Blackjack Game development
CSC 648
Section 01 Team 05
Milestone 4 - Testing

Team members

Seng Maw – Team leader & Scrum Master
Pritam Gautam – Frontend lead
Sneha Shrestha – Frontend lead
Lane Maimone – Backend lead
Nathaniel Miller – Backend lead
Tyler Hsieh – Backend lead & GitHub Master

Product Summary

Blackjack webapp game

<https://csc-648-group5-babyblackjack.web.app/>

Committed Functions

- User should be able to Sign Up
- User should be able to Login
- User should be able to play game in single player
- User should be able to play the game multiple times until the wallet is empty
- User should be able to see the score

Unique Features of the product

The purpose of our product is to have access to the casino experience from one click away without having to worry about losing the actual money. The user interface for the game is very easy to explore and play. Our team has put the time into choosing the right shade and color scheme for the theme of the game. The commands we used in the game are phrases which give the user a real feel of the actual casino.

QA testing

Unit Testing

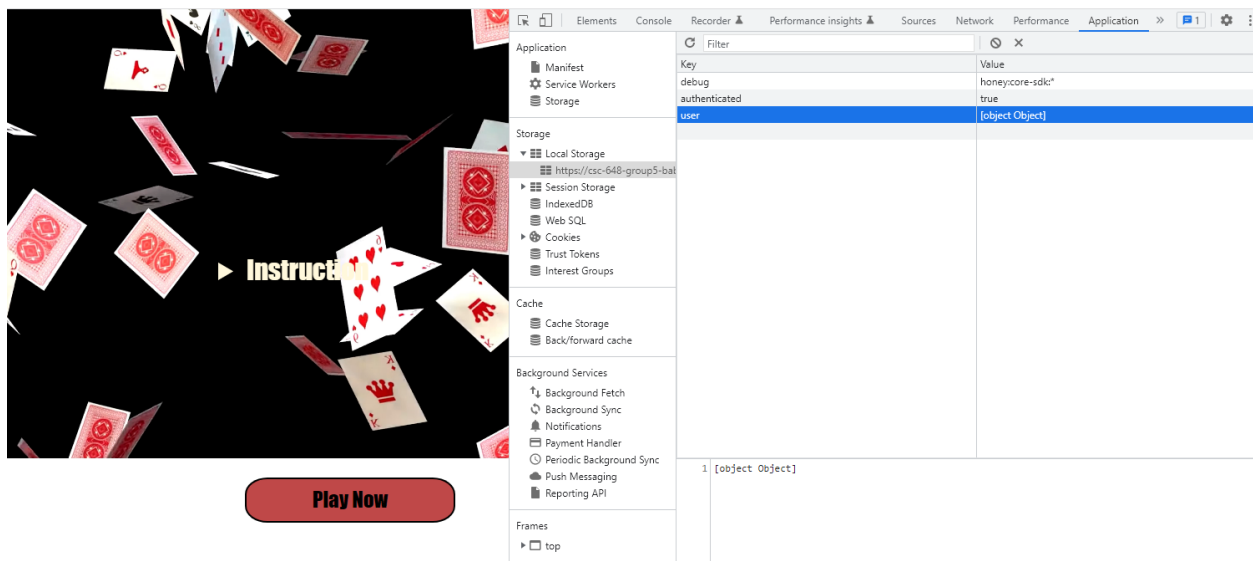
Unit testing is focused on the each of the components of the whole web application work well independently as intended for both front-end and back-end features such as:

- User signing up and account creation (Front-end Test)
- Saving the user info into database for logging in purpose (Back end Test)
- Responsive displaying for different browsers and end systems (Front end)
- Game working as intended (sub features included) (Back end test)
- Display the score for user (Back end test)

The log in test is passed.

Logging in with invalid login: credentials (email: notanaccount@mail.com, password: fakepass) -> stays on login page -> passed)

Logging in with valid login: credentials (email: test@gmail.com, password: test) -> redirects to /main, stores "authenticated" with value true, and user object (contains username, email, and password) in browser localStorage -> passed



Game Feature Test (Back end) – Functions , Classes

Integration Testing

We pushed the tests we did for the functionality of the game in github and below is the link.

<https://github.com/CSC-648-SFSU/csc-648-project-csc648-01-fall22-Team05/tree/wip/applications/server/game>

The below are the different test cases for features of the game.

Test Case ID	test01	Test Case Description	Check the natural blackjack cases for playerr			
Created By	Lane Maimon	Reviewed By	Lane Maimone	Version	1.0	
QA Tester's Log						
Tester's Name	Lane	Date Tested	December 4, 2022	Test Case (Pass/Fail/Not	Fail	
S #	Prerequisites:		S #	Test Data Requirement		
1	A new game must begin.		1	dealerCard[1] = K, dealerCard[2] = A		
2			2	playerCard[1] = A, playerCard[2] = 10		
3			3			
4			4			
Test Conditions						
Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended		
1	Player has natural 21	player loses	doesn't run entirely	Not executed		
2						
3						
4						
5						
Test Case ID	test03	Test Case Description	Tests that the game can build a deck of cards			
Created By	Lane Maimone	Reviewed By	Lane Maimone	Version	1.0	
QA Tester's Log						
Tester's Name	Lane Maimone	Date Tested	December 4, 2022	Test Case (Pass/Fail/Not	Fail	
S #	Prerequisites:		S #	Test Data Requirement		
1	There can't be a deck already made		1	None just call the function to build a deck.		
2			2			
3			3			
4			4			
Test Conditions						
Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended		
1	Call the deck build function	deck with every card	When ran individually it works otherwise	Not executed		

Test Case ID	test02	Test Case Description	Check player win with non blackjack conditions			
Created By	Lane Maimone	Reviewed By	Lane Maimone	Version	1.0	
QA Tester's Log						
Tester's Name	Lane	Date Tested	December 4, 2022	Test Case (Pass/Fail/Not	Fail	
S #	Prerequisites:		S #	Test Data Requirement		
1	A new game must begin		1	player as better than 17 and dealer has 17		
2			2			
3			3			
4			4			
Test Conditions						
Step #	Step Details	Expected Results	Actual Results		Pass / Fail / Not executed / Suspended	
1	set dealer and player to 17 and 18	player wins	it fails to run		Not Executed	
2						
3						
1						
2						
3						
Test Case ID	test01	Test Case Description	Check the natural blackjack cases for dealer			
Created By	Lane Maimone	Reviewed By	Lane Maimone	Version	1.0	
QA Tester's Log						
Tester's Name	Lane	Date Tested	December 4, 2022	Test Case (Pass/Fail/Not	Fail	
S #	Prerequisites:		S #	Test Data Requirement		
1	A new game must begin.		1	dealerCard[1] = K, dealerCard[2] = A		
2			2	playerCard[1] = A, playerCard[2] = 10		
3			3			
4			4			
Test Conditions						
Step #	Step Details	Expected Results	Actual Results		Pass / Fail / Not executed / Suspended	
1	Dealer has natural 21	player loses	doesn't run entirely		Not executed	
2						
3						
4						
5						
6						

Code Reviews

We created different branches and several pull requests regarding the different features that we implemented for the review purpose within the group members. Below is the link for the code review we did with pull requests.

<https://github.com/CSC-648-SFSU/csc-648-project-csc648-01-fall22-Team05/pulls>