CPSC 473

Web Front-End Engineering for Internet Applications

CALIFORNIA STATE UNIVERSITY
FULLERTON™

Department of Computer Science,

California State University, Fullerton

PROJECT – 2

BlackJack

Submitted to:
Professor Kenytt Avery

Submitted By:
Chris Anderson
Natalie Ceballos
Anika Corpus
Luke Higgott
Felipe Perez
Pragya Tak

Find the source code at https://github.com/codepoetchris/CPSC473-Project-2

May 15, 2017

Table of Contents

I. Motivation

BlackJack is an online web application where a user can play the popular card game and socialize with friends. Want to enhance your blackjack skills without breaking the bank? BlackJack give the user the opportunity to play the game and increase their techniques without the risk of gambling their savings away. This application allows the user to play their favorite card game and socialize with others at their convenience.

II. Introduction

BlackJack application is a platform where a user can socialize with others and play a leisurely game of blackjack.

To implement the project, we used the following technologies:

1. Ember CLI
2. Bower and NPM
3. JavaScipt
4. Ember Server and Node.js.
5. PhantomJS

III.Scope

This document describes the functionalities of our BlackJack web application. It also describes the configurations and required settings to run it on any system. Upon completing this application, features such as sign in, playing the game, and chat will be available for every user.

IV. Setup and Installation

This application is currently running on local machines. To use the application and its' features, the user must download the following requirements and follow the listed instructions.

Windows Users:
Clone the project from the following github url:

(https://github.com/codepoetchris/CPSC473-Project-2.git)

- Download and install node.js
- “npm install” for installing all the node modules.
- Download and install bower
- “bower install” for installing all the bower modules.
- “ember server” to run ember
- “node websockets-server.js” to run the websocket server
- Run the application “BlackJack” using the following url: (http://localhost:4200)

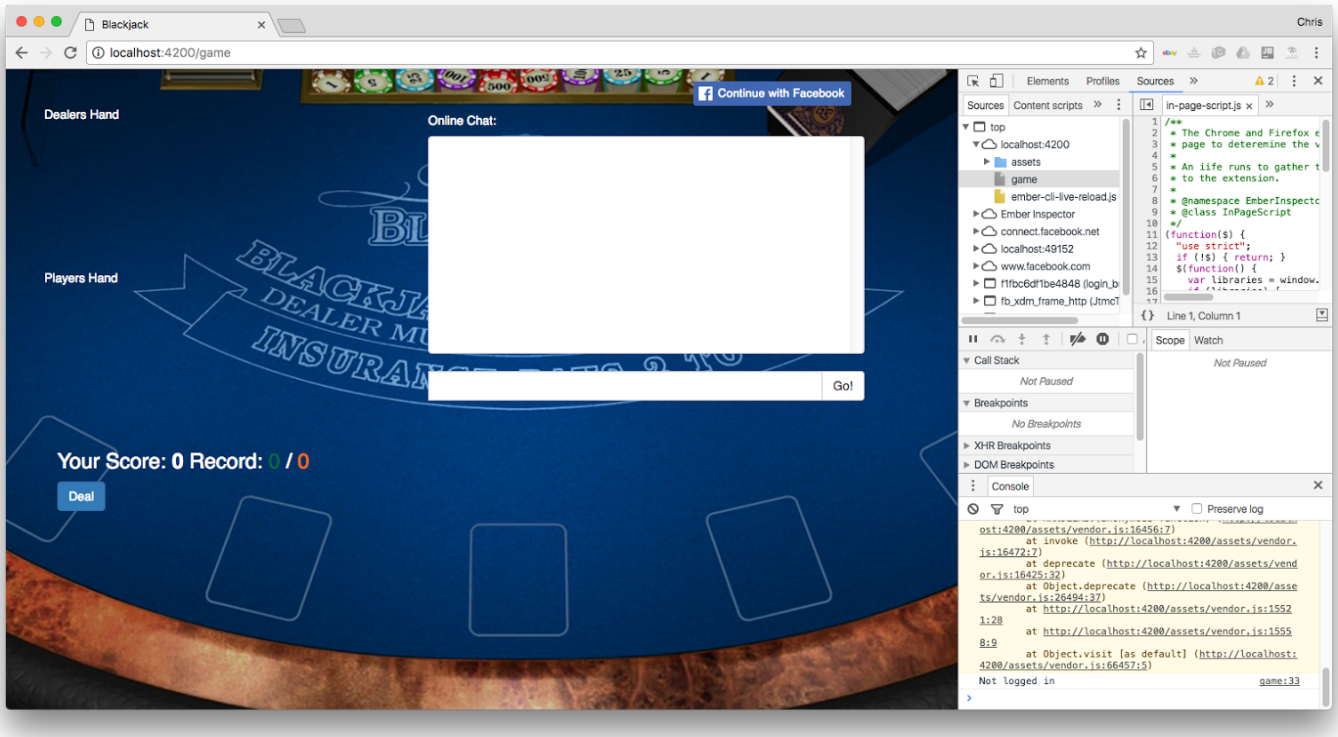
V. Functionalities

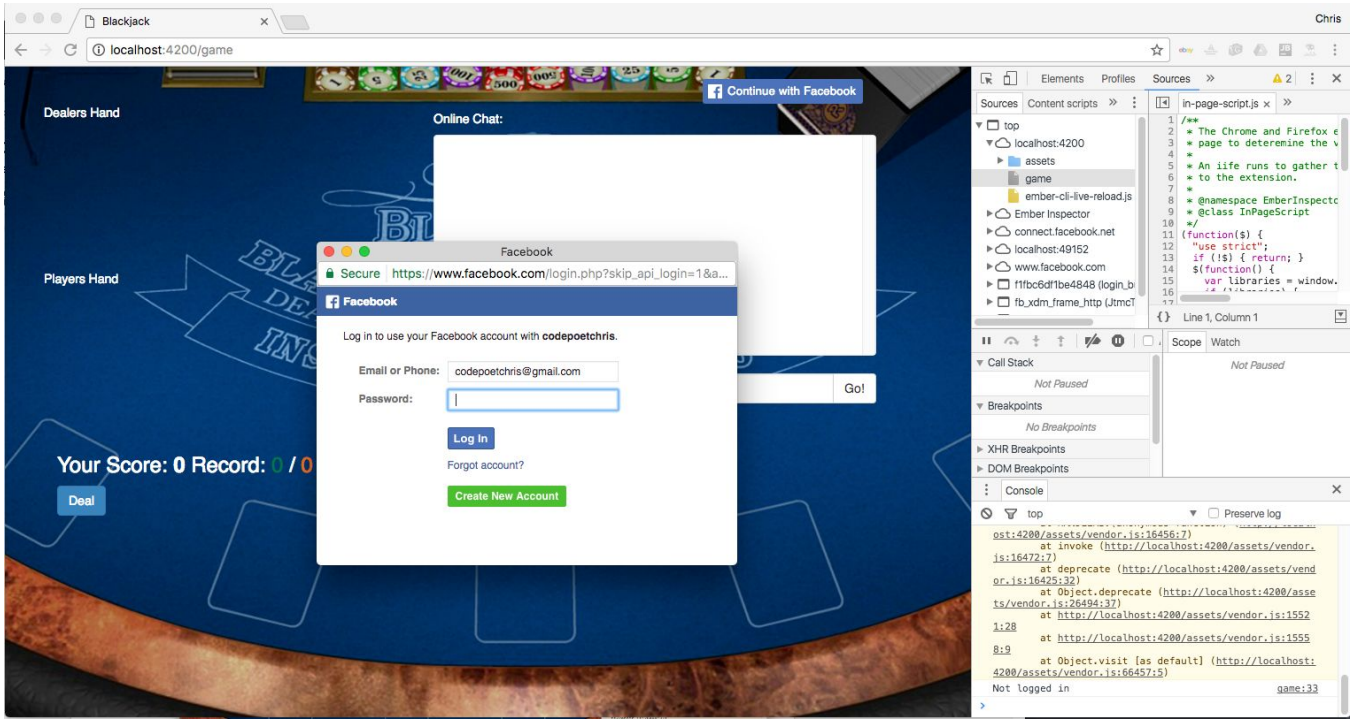
1. Login to game
2. Play interactive blackjack game
3. Chat with other users logged into game

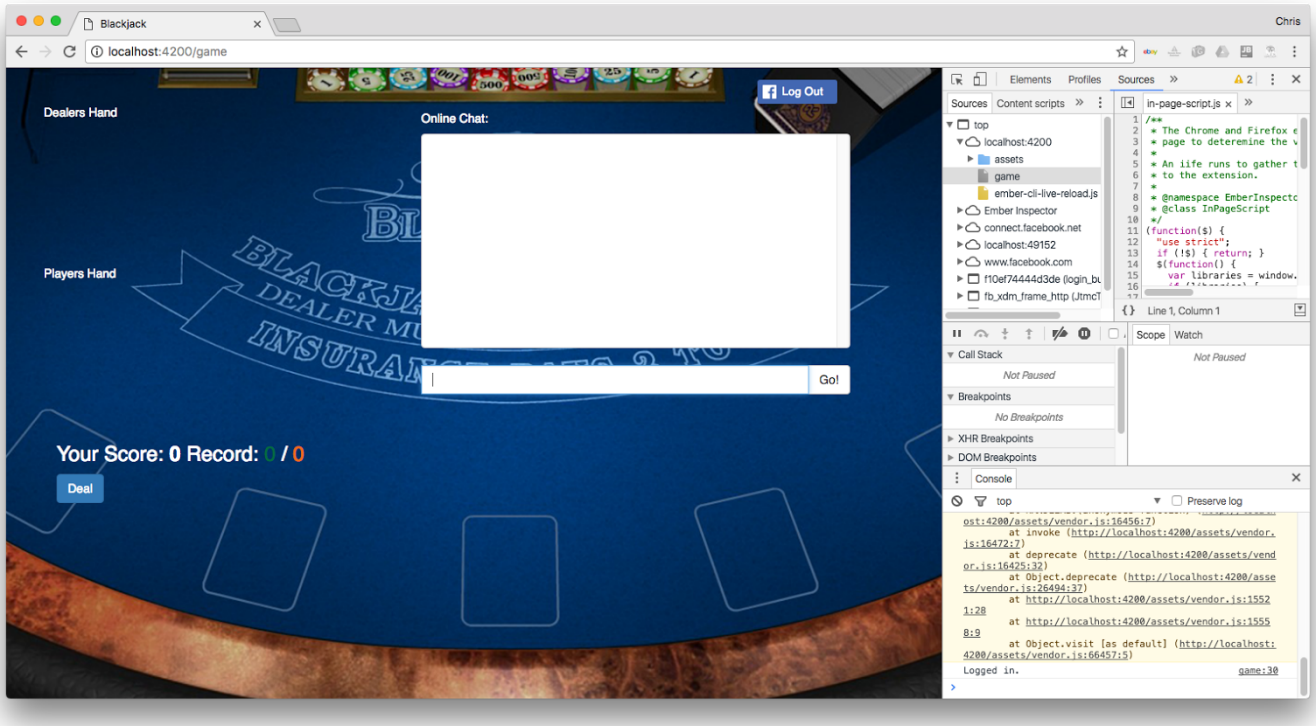
VI. Application In Action

1. **Login** - To fulfill the 3rd party webservice API requirement we decided to implement Facebook Authentication. Upon launching the application the authenticator will make a call to Facebook to determine if the user is logged in. The users status is logged to the console. If not logged in, clicking the “Continue with Facebook” button in the top-right of the screen will prompt the user for their facebook login and perform the login. At this point the “Continue with Facebook” button will change to a “Log Out” button and authentication is complete.

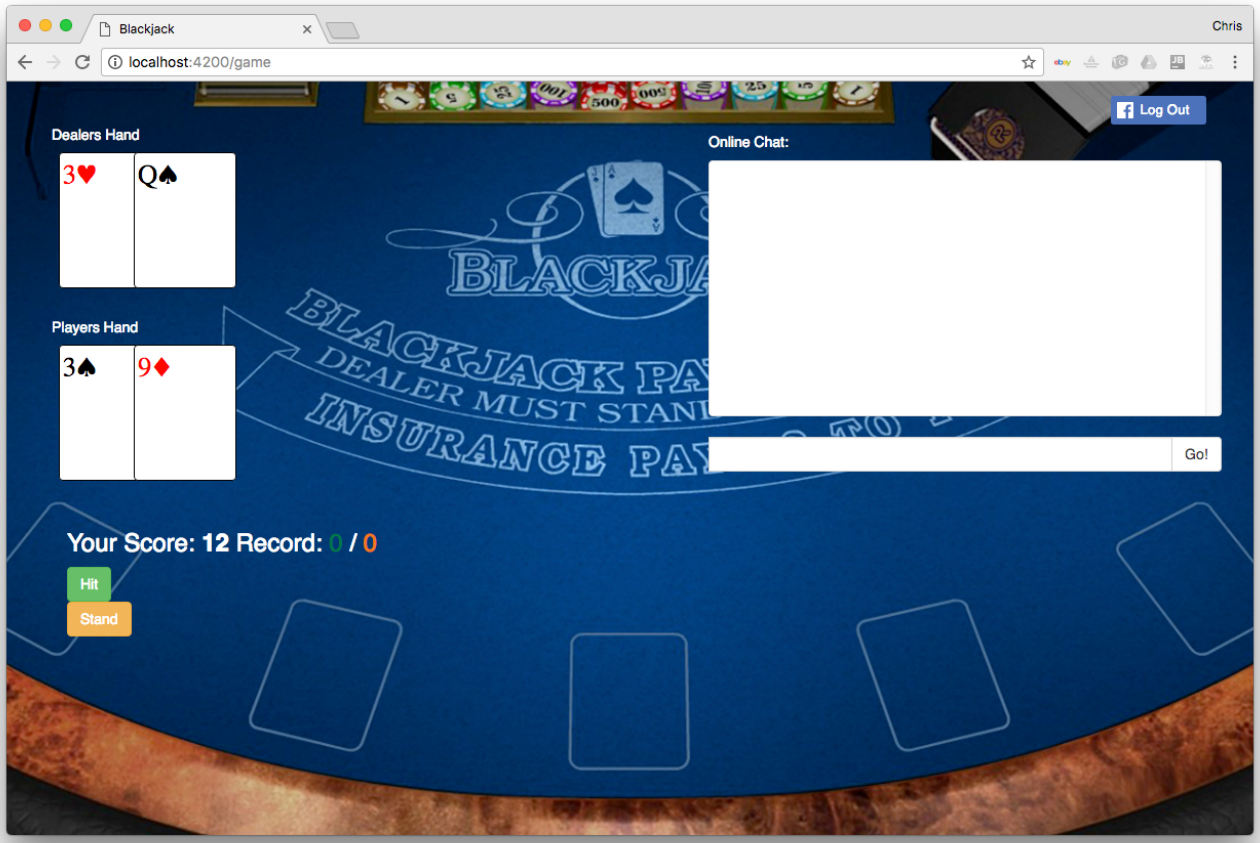
   Our intent was to create a landing page that prompts the user to login prior to playing the game. This way we could setup a profile for the user with information such as bank account numbers and game play statistics.

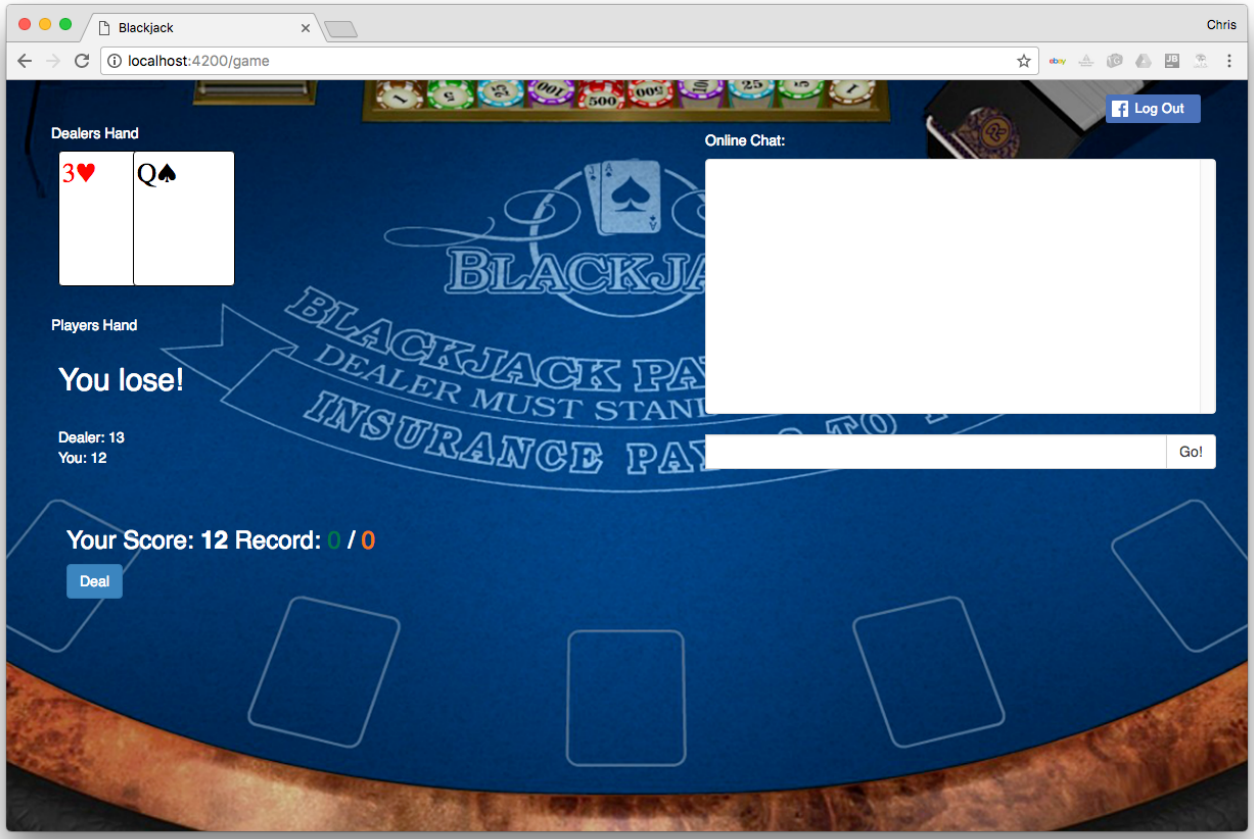

Blackjack
Chris
localhost:4200/game
Dealers Hand
Online Chat:
Continue with Facebook
Players Hand
Go!
Your Score: 0 Record: 0 / 0
Deal
Elements
Profiles
Sources
Sources
Content scripts
in-page-script.js
top
localhost:4200
assets
game
ember-cli-live-reload.js
Ember Inspector
connect.facebook.net
localhost:49152
www.facebook.com
Line 1, Column 1
Scope
Watch
Not Paused
Call Stack
Not Paused
Breakpoints
No Breakpoints
XHR Breakpoints
DOM Breakpoints
Console
Preserve log
Not logged in
game:33


Blackjack
Chris
localhost:4200/game
Dealers Hand
Online Chat:
Continue with Facebook
Players Hand
Facebook
Secure
https://www.facebook.com/login.php?skip_api_login=1&a...
Log in to use your Facebook account with codepoetchris.
Email or Phone:
codepoetchris@gmail.com
Password:
Log In
Forgot account?
Create New Account
Go!
Your Score: 0 Record: 0 / 0
Deal
Not logged in
game:33



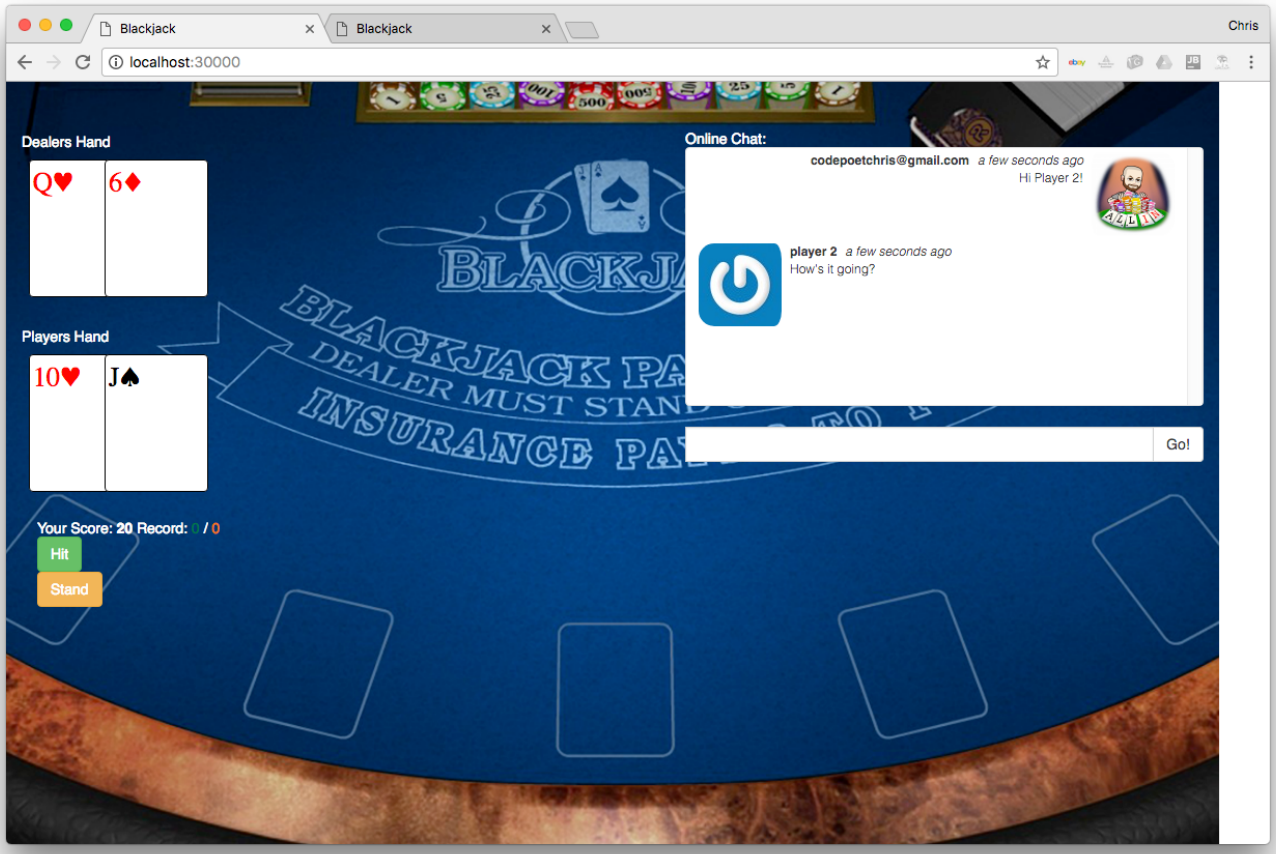
2. **Playing the Game** - You first start out by hitting the “Deal” button. You then make a decision on if you want to “Hit” or “Stay.” These are the only two actions implemented in this version of the code. Future versions would implement betting, splitting, surrender, insurance, etc.

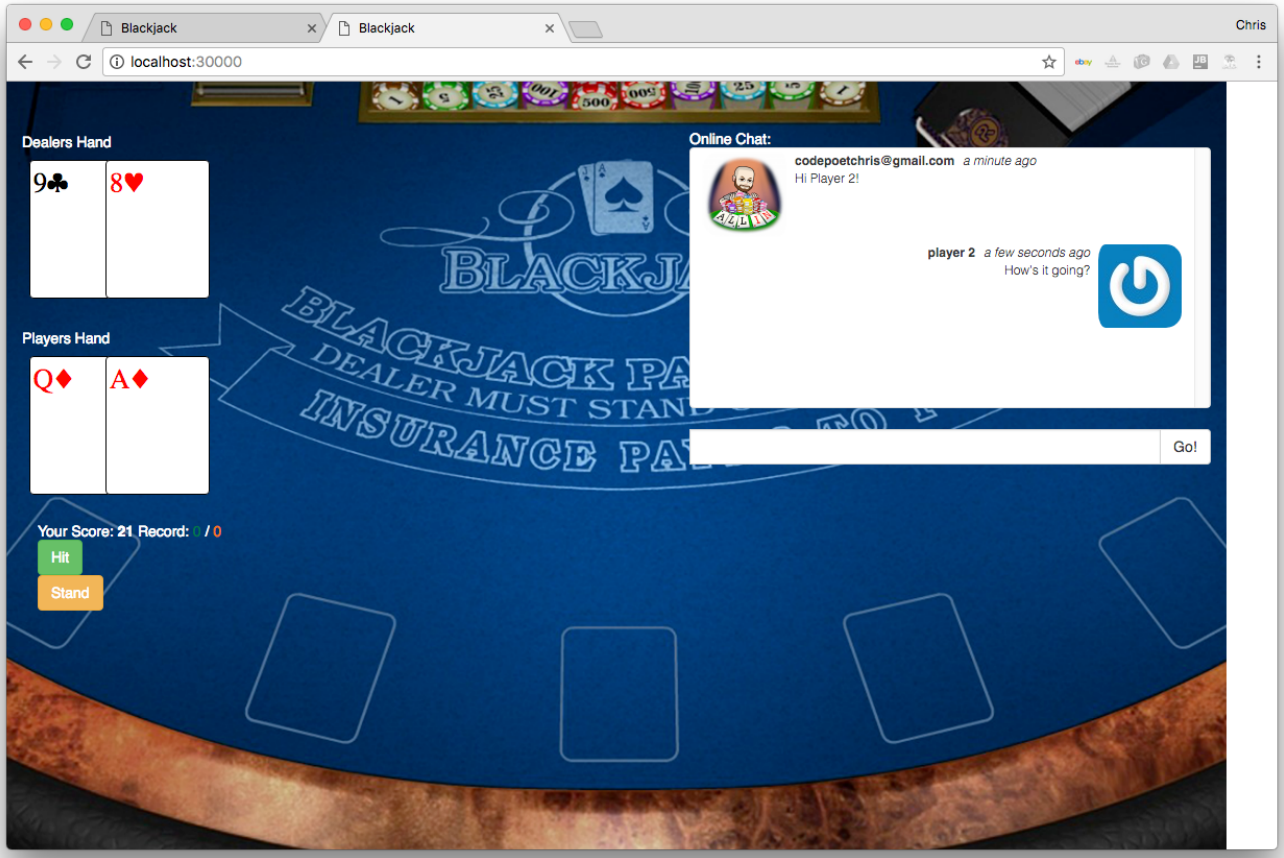

Blackjack
Chris
localhost:4200/game
Log Out
Dealers Hand
3♥
Q♠
Online Chat:
Players Hand
3♠
9♦
Go!
Your Score: 12 Record: 0 / 0
Hit
Stand



3. **Websockets** - Our intention was to make the Blackjack game multiplayer via websockets but we struggled greatly with the implementation in Ember. You'll notice in our github repository that there are several branches devoted to trying to get websockets working. The "BlackJackDeal-UserFunctions" branch contains a fully functional chat running on node as demonstrated in the screen captures below. You'll need to execute *node index.js* to run this branch. The site is hosted at http://localhost:30000.

   The "Implement-ws-instead-of-socket.io" branch contains a working websocket server (node websockets-server.js), however, we were not able to completely port the code for the chat nor implement the multiplayer functionality for the actual game.


Blackjack
Blackjack
Chris
localhost:30000
Dealers Hand
Q♥
6♦
Players Hand
10♥
J♠
Your Score: 20 Record: 0 / 0
Hit
Stand
Online Chat:
codepoetchris@gmail.com a few seconds ago
Hi Player 2!
player 2 a few seconds ago
How's it going?
Go!



VII. References

- Font-End Web Development - The Big Nerd Ranch Guide by Chris Aquino and Todd Gandee
- https://ember-cli.com/
- https://emberjs.com/