What is the intended purpose of your application?

IO game themed around a warehouse with a store as well  
● What work has been completed for this milestone?

A bunch of react components have been created  
● What work is left, and how do you plan to complete it?

The whole game aspect. I plan on using either p5.js or pixi.js for the game. Then I will also be using web sockets  
○ What does your timeline/roadmap look like to finish on time?

Work on this for the entire break for the most part and then  
● How are you using React?

Right now just for components from the database but I plan on implementing an admin role for the user so they can’t see certain things on the page  
○ What components have you made?

Login/signup/Account info page/Shop item/Shop item details  
○ What components do you still plan to add?

Not sure need to brainstorm.  
● What data are you storing in MongoDB?

Account and the store items. Account has had a lot of variables added to it for game purposes  
○ What data do you still need to store?

For account, currency, the skins owned, admin, high score and games played  
● What is your profit model?

Store to buy skins for your character  
○ Have you implemented it yet?

No only visually  
■ If so, how?  
■ If not, what is your plan to implement it?

Going to have a store page and there is either going to be a shopping cart element or like a confirm page to pad out the react requirements  
● Do you have a plan for going above and beyond? If so, what is it?

Finish the assignment  
● If you used any borrowed code or code fragments, where did you get them from?  
What do the code fragments do? Where are they in your code?

A lot of the Account/login functionality is the same from domo. Also I have a websocket IO that is ripped straight out of the demo but is not currently in use at all.

app.get('/', mid.requiresSecure, mid.requiresLogout, controllers.Account.loginPage);

Return user to the login page. Returns nothing

  app.get('/getToken', mid.requiresSecure, controllers.Account.getToken);

Retrieves a csrf token, no params

  app.get('/login', mid.requiresSecure, mid.requiresLogout, controllers.Account.loginPage);

Returns the user to the login page

  app.post('/login', mid.requiresSecure, mid.requiresLogout, controllers.Account.login);

Logs the user in, parameters username, password, returns nothing

  app.post('/signup', mid.requiresSecure, mid.requiresLogout, controllers.Account.signup);

Signs the user up, parameters Username, password1 and 2, returns nothing

  app.get('/logout', mid.requiresLogin, controllers.Account.logout);

Closes the session

//app.get(‘/purchase’, mid.requiresLogin, controllers.ShopItem.buyItem);

Currently left out but, will add the item to the user’s array of available skins and subtract currency. Parms are the account ID and account currency along with shopitem ID.