HODIT PROJECT PLAN

Section 1: Body Parts

Head
- Eyes
- Nose
- Ear
- Mouth

Neck
- Throat

Thorax
- Chest/ Breasts
- Heart (symptoms can be felt)

Spine
Shoulders
Upper Arms
Elbows
Forearms
Hands
Abdomen
- Stomach
- Gastrointestinal system

Lower Back
Hips
Pelvis (Pelvic Region)
- Genitals

Buttocks
Upper Legs
Knees
Shins
Calves
Achilles
Ankles
Feet

Section 2: Information

Common Symptoms and Treatments (27)

Allergies: Irritated Eyes, Itchy Nose, Sneezing, Runny Nose, Rashes, Stomach Cramps, Vomiting, Cough, Loss of Breath, Swelling.

Treatment: If your allergies are severe, see a doctor to identify what you are allergic to. If they are merely annoying, you may take an over-the-counter medication prescribed by a pharmacist.

Alzheimer's/Dementia: Memory Loss, Confusion, Mood Changes

Treatment: Alzheimer's is a very complex disease. Therefore, it is unlikely anything can completely cure it. You may be able to achieve results with inhibitors like Razadyne® (galantamine), Exelon® (rivastigmine), and Aricept® (donepezil).

Arthritis: Muscle Pain, Stiffness, Swelling, Redness, Decreased R.O.M.

Treatment: You can treat arthritis depending on the type. Common medications include Painkillers, NSAIDs, Counterirritants, or Counterirritants. You can also do various types of therapies to slow down the swelling inside your body and joints.

Bug Bite: Swelling, Redness, Rashes, Skin Pain, Itching

Treatment: Please immediately move to a safe area to avoid more bites or stings. Then wash the area with soap and water if necessary. Apply pressure, and if the itching is intense, take an antihistamine such as Benadryl to reduce itching. Please do not scratch or itch, as it will do more harm than good.

Blood Pressure (High): Headaches, Loss of Breath, Nosebleeds, Dizziness, Chest Pain

Treatment: Remember that blood pressure may be genetic, so this might not work for you. Make sure to lose weight, exercise, eat healthy, reduce salt intake, drink less, cut caffeine and quit smoking. If your stress levels are high, that may also be the issue.

Cancer: Weight Loss, Fatigue, Fever, Urinary Bleeding, Rectal Bleeding, Hoarseness, and Coughing Blood

Treatment: There is no cure for cancer, and you must make sure you check with your doctor. You can partake in types of cancer treatments such as chemotherapy, radiation therapy, immunotherapy, or hormone therapy.

Cardiovascular Disease: Chest Pain, Loss Of Breath, Lightheadedness, Dizziness, Fainting, Racing Heartbeat

Treatment: You can take various types of surgery such as coronary artery bypass or valve repair and even replacement surgery. You can do some lifestyle

changes such as losing weight, exercising, following a good diet and quitting destructive habits such as smoking.

Cold & Flu: Body Aches, Chills, Fatigue, Sneezing, Cough, Sore Throat, Headaches, Fever

Treatment: The best thing you can do is rest. If you rest often, then the fight against the virus will be easier. However, you can drink plenty of water, take some over-the-counter medication for congestion, and be sure to drink plenty of warm liquids.

Conjunctivitis (NEW): Irritated Eyes, Light Sensitivity, Eye Swelling, Blurred Vision

Treatment: Known as "pink eye", conjunctivitis can be treated with eye drops. Your doctor may prescribe an antibiotic, usually in the form of eye drops or ointment for the bacteria.

Coronavirus (NEW): Fever, Chills, Cough, Loss Of Breath, Fatigue, Headaches, Body Aches, Sore Throat, Runny Nose, Vomiting

Treatment: It is essential that you remain alert and cautious, as this is an ongoing pandemic. Please stay home as much as possible to avoid spreading the virus. If necessary visit your doctor to make sure you have not contracted the common cold. Be sure to be well-hydrated, and take over-the-counter-medications such as acetaminophen.

Depression (NEW): Irritability, Weight Loss, Mood Changes, Sleepiness, Hunger

Treatment: It is important to know that depression has both physical and mental implications. You should talk to a therapist, or if it is not serious, take medications such as antidepressants, SSRIs, and Antipsychotic.

Diabetes: Hunger, Fatigue, Blurred Vision, Nausea, Numbness

Treatment: Type II diabetes is genetic, so that will play a factor in your treatment. However, you can still lower blood sugar levels by consuming less sugary foods, exercising, weight reduction and dietary changes.

Fracture: Swelling, Pain, Decreased R.O.M., Difficulty Moving

Treatment: There is nothing more you can do to speed up the process of healing. Make sure to wear splints, braces, or casts, and be sure to avoid using the affected area for a while.

Gastroenteritis (NEW): Vomiting, Chills, Weight Loss, Stomach Cramps, Nausea

Treatment: Gastroenteritis is caused by the norovirus. There are many more, but norovirus is the most common. Make sure to drink plenty of fluids, take antibiotics, and make sure you go to the hospital to check if there is an

underlying condition. It is also likely this is caused by a parasite, if so take drugs designed to kill them from a pharmacy.

Golfer's Elbow: Forearm Pain, Tenderness, Stiffness, Forearm Weakness

Treatment: This is not so severe, so do not worry too much.Make sure to rest, apply ice, take pain relievers if necessary, stretching and physical therapy.

Herpes: Difficulty Urinating, Genital Pain, Genital Itching, Swollen Glands

Treatment: You can take painkillers, bathe in salted water, take warm baths, apply petroleum jelly, wear loser clothing and refrain from sexual activity until all symptoms have disappeared.

HIV/AIDS: Fever, Chills, Rashes, Sore Throat, Body Aches

Treatment: There is no cure for HIV. It takes a gradual amount of time to become AIDS. One treatment for HIV is called antiretroviral therapy where you take a combination of HIV medicines every day. Although it will not cure HIV, it will help you live a longer, happier life.

Indigestion (NEW): Nausea, Stomach Cramps, Bloating

Treatment: Indigestion is also related to heartburn, but make sure that indigestion is definitely the cause for your discomfort. Avoid foods that trigger these responses such as alcohol, peppers, chocolate, citrus fruits, or tomatoes. Eat in smaller meals of fix to six instead of three or four.

Migraine: Headache, Light Sensitivity, Nausea, Vomiting

Treatment: Firstly, see if you can take an MRI or CT scan so that you are sure that the migraine is not indicative of another underlying, more severe condition. If it is a simple migraine, you may take pain relievers, triptans, or opioid medications.

Osteoporosis: Back Pain, Height Loss, Posture Problems, Soft Bones

Treatment: You can take treatment such as bisphosphonates, monoclonal antibodies or hormone related therapy. Osteoporosis can be a result of low calcium intake, which means you should consume these foods, one great example of which is milk. You can also exercise, to strengthen both your bones and muscles.

Pregnancy(women): Vaginal Bleeding, Vaginal Cramps, Swollen Breasts, Fatigue, Mood Changes

Treatment: Pregnancy is not something to be treated as it is natural. If you are seeking to prevent pregnancy, be sure to use contraception or spermicide.

Pneumonia: Cough, Fever, Loss Of Breath, Chest Pain, Nausea, Vomiting, Fatigue

Treatment: Firstly, do not take cough medicines unless approved by your doctor. If you have a fever, you can control it with aspirin, or anti-inflammatory drugs. Make sure to drink plenty of fluids to loosen phlegm and secretions.

Sleep Apnea: Snoring, Dry Mouth, Irritability, Headache, Sleepiness

Treatment: The most effective treatment for sleep apnea is to use a CPAP (continuous positive airway device). Some lifestyle changes you should do are losing weight, trying different sleep positions, and to avoid drinking and smoking.

Sprain (Joint General): Muscle Pain, Swelling, Difficulty Moving, Joint Popping, Bruising

Treatment: Be sure to remember RICE. Rest, Ice, Compression, and Elevation. In general you should avoid using the affected area of the body until certified by a doctor.

Strain (Muscle General): Spasm, Muscle Pain, Swelling, Decreased R.O.M.

Treatment: You should protect the strained muscle from further injury. Similar to a sprain, use RICE for effective treatment. Rest, Ice, Compression, and Elevation.

Shin Splints: Soreness, Muscle Pain, Swelling

Treatment: Keep your legs elevated, use ice packs, take over the counter anti-inflammatory medications such as ibuprofen or naproxen sodium. Also wear elastic compression bandages and form rollers.

Tendinitis: Muscle Pain, Swelling, Difficulty Moving, Lumping

Treatment: Similar to sprains, strains and shin splints, use the RICE program. Rest, ice, compression, and elevation of the injured tendon. If necessary, use aspirin, ibuprofen, or other anti-inflammatory drugs to help with the pain.

Section 3: Plans (step 6)

Original Plan

1) Make a branch of symptoms for each body part, with each branch ending with a diagnosis or a no diagnosis available. Some symptoms are shared with each body part like itching, redness or swelling.
2) When a body part is clicked, a dropdown menu will appear and based on user selection, will add the symptom to the symptoms menu. Below the symptoms will be a diagnosis. I plan to make a diagnosis after 4 symptoms. What will be difficult is cross body symptoms, which will have merging branches.
3) Common Symptoms will branch off: Cough(3), Loss Of Breath(4), Sneezing(2), Pain(11),

Rashes(3), Swelling(6), Redness(2), Vomiting(3), Fatigue(4), Nausea(3), Dizziness(2), Fever(3),

4) In order to simplify the project, I will group certain muscles together, since no matter if it is the biceps, triceps, or the leg muscles, they will generally lead to the same diagnoses. Only other parts of the body, like the head or the thorax have UNIQUE symptoms.
5) I will start with parts of the body that will lead to certain symptoms.
   Head(11): Allergies, Alzheimer's/Dementia, Bug Bites, Blood Pressure(High), Cancer, Cardiovascular Disease, Cold & Flu, Diabetes, HIV/AIDS, Migraine, Pneumonia, Sleep Apnea
   Neck/Throat(4): Allergies, Bug Bites, Cold & Flu, HIV/AIDS, Pneumonia
   Thorax(4): Allergies, Blood Pressure, Cardiovascular Disease, Pneumonia
   Spine(3): Arthritis, Fracture, Osteoporosis
   Shoulders(5): Arthritis, Fracture, Sprain, Strain, Tendonitis
   Upper Arms/Legs(3): Sprain, Strain, Tendonitis
   Elbows, Knees(3): Fracture, Tendonitis, Golfer's Elbow (Elbows only)
   Forearms(3): Sprain, Strain, Tendonitis
   Hands, Feet(2): Arthritis, Bug Bites, Fracture
   Abdomen(5): Allergies, Cancer, Diabetes, Migraine, Pneumonia

Modified Plan

Instead of using the branch method, I figured out a much easier method that avoids the exponential nature of my original method. What I will do instead is implement a matching algorithm that narrows down the symptoms. My new method is below.

1) Have a complete list of symptoms for each body part. I decided this would be better than my original plan of a branch system, because this way if a user had a particular symptom they could see it right away instead of having to go through each branch.
2) Have an array of objects called diagnosesList, which has each object and its name(for reference), necessary symptoms, and the diagnosisReport which is what will show on the diagnosis area.
3) When a symptom is selected, look up each object in the array to see which of them contains that particular symptom. Chances are that if these symptoms match, it is MORE likely it is that particular disease or illness. It will add ALL of the diseases which contain this keyword to an array possibleDiagnoses. As more symptoms are added, it will filter through this new array.

4) When only one object matches all the particular symptoms, display it on the diagnosis area with the possible treatment. I will achieve this by checking if there is only one item in possibleDiagnoses.

```
diagnosesList =
 [diagnosisA = {symptoms:["A","B","C"], name:"diagnosisA", treatment:"lorem }
    diagnosisB = {symptoms:["D","E","F"], name:"diagnosisB", treatment:"lorem"},
    diagnosisC = {symptoms:["G","H","I"], name:"diagnosisC", treatment:"ipsum"}]
 for(i = 0;i<diagnosesList.length;i++)
 {if(diagnosesList[i].symptoms.includes("F")){alert(diagnosesList[i].treatment)}}
```

This will be my simplified method. There will be a list of diagnoses, and then later I will see if the diagnoses contains a specific keyword. I will not define each object with an = sign, as it is not necessary to do so.

Section 4: Log

1: Gather necessary information and make a website plan.

I will gather as much information as I can regarding different types of symptoms, and diagnoses. Since this is not an actual website, I am planning to get around 20 diagnoses to make for various unique outcomes. What will be difficult is figuring out an algorithm for the matching, but for now I will plan a layout for my page. I planned out a rough table of what I need below.

A Title/Header (Possibly set up for enhancements?)
Images with the gender buttons
A symptoms/diagnosis container
Other things to be added later.

As suggested by the project specifications, I will need to create various image maps. One helpful resource I will use will be: https://www.image-map.net/. I am unsure as to which body parts I will need to specifically section.

2: Find images, create image maps and create a layout.

I will need to find 'medical' images of the human body, so I can choose between various types of systems ranging from muscular,skeletal,nervous,or circulatory systems. I did find images online from Adobe stack which were detailed images of the muscular system. I will therefore center my project around different regions of the human body. Now, I will add the various image maps with the tool I saved yesterday. When I tested my new image maps on my images, to my dismay they did not appear. I found that image maps will be "lost" if the image itself has a different position. To resolve this I will create a container of the image called anatomyPos which will contain the image. I found that when I implemented the code below, that the area selection is still retained.

```
#anatomyContainer
{
 position:absolute;
 left:11.5vw;
 top:20vh;}
```

In addition, I learned of jQuery plugins such as imageMapster that makes highlighting possible as native CSS does not appear to be of any help at all. I now need to create a button that will change between male and female. To do this all I have to do is create a simple function which will change the image source, but also the image map and position. I also will add male and female symbols to make it more condensed instead of writing "MALE" and "FEMALE". .useMap changes the map used.

```
function changeGender(button) // Changes genders and changes the image map.
{
  anatomyPos = document.getElementById("anatomyContainer");
  if (button.innerHTML == "♂")
  {
     button.innerHTML = "♀";
     button.style.left = "47.5vw";
     button.style.backgroundColor = "#eb89b5";
     anatomyPos.style.left = "21vw";
     anatomyPos.style.top = "16.5vh";
     anatomyImg.src = "images/female.png";
     anatomyImg.useMap = "#femalebody";
     return;
  }
  if (button.innerHTML == "♀")
  {
     button.innerHTML = "♂";
     button.style.left = defaultStatus;
     button.style.backgroundColor = "#226089";
     anatomyPos.style.left = defaultStatus;
     anatomyPos.style.top = defaultStatus;
     anatomyImg.src = "images/male.png";
     anatomyImg.useMap = "#malebody";
     return;
  }
}
```

For now, I have centered the image and will not do any styling until later. I might reuse the previous menu design from my Level 5 Project. I could have used a

label selector similar to the level 5 project and avoided some Javascript, but for now I want to keep it clear and simple.

3: Create a function to show which body part was selected.

Ultimately I will have to have dynamic dropdowns that have various options within them. I considered using a custom dropdown, but that may cause some more complicated issues later down the line. I found an element called select which has a prebuilt dropdown that I wish I knew earlier. While the select tag can be styled, its option tags cannot. For now, I need a couple things to create a dynamic dropdown that (for now) shows the selected part. I will need the title of the image map area, and I need to have the position of the mouse such that I can place the select option next to it. To do this, I will access the title property by passing it through a parameter, and set the option.innerHTML equal to the title.

```
function displaySymptoms(area)
{
    selection = document.createElement("SELECT");
    myNode.appendChild(selection);
        option = document.createElement("OPTION");
        selection.append(option);
    option.innerHTML = area.title;
}
```

For the select box to be next to the mouse, I will need to use event listeners. I could technically use event handlers, however upon experimentation they capture the event one time. I need the mouse position (clientX and clientY) and place the select element in the appropriate area. In addition, I want to make sure it is placed on the side where the client mouse is. I pinpointed the exact line separating the backside and frontside of the body images and used that to determine where to position the element.

```
window.addEventListener('click',function(e)
{if(e.target.tagName == "AREA")
  {
    myNode.style.top = (((e.y)/(window.innerHeight))*100)+"vh";
    if(event.clientX > (window.innerWidth*(1/3))){myNode.style.left = "57vw";}
    else{myNode.style.left = "4vw";}
  }});
```

4: Transfer all information from the research.

Now that I have all the information from online websites, I need to figure out how to organize this information in my code. I am leaning towards creating a bunch of objects for each body part that has its initial symptoms, and then a bunch of objects in a diagnosis list that has various attributes such as the symptoms, name, and treatment.

```
Head = {symptoms: ["Blurred Vision","Body Aches","Chills","Confusion","Dizziness","Dry Mouth","Face Swelling","Fainting","Fatigue","Fever","Headaches",
  "Irritability","Irritated Eyes","Itchy Nose","Lightheadedness","Light Sensitivity","Memory Loss","Mood Changes","Nosebleeds",
  "Numbness","Rashes","Runny Nose","Sneezing","Snoring"]};
```

The body part objects only have symptoms and I will see if I need to add more properties down the line. I have to add one for every body part, however I have to group certain muscle and bone types. This way it will reduce repetition.

```
jointType = {symptoms:["Difficulty Moving","Joint Popping","Stiffness","Swelling"]};
```

I experimented with a branch system but then I realized that the complexity would be exponentially increasing. So instead I redid everything by matching the symptoms to the diagnoses. This algorithm is not only easier, but allows for expandability if I choose to add more in the future.

5: Populate initial dropdowns based on user selection.

With the selected body part I need to populate the initial select box with the symptoms in the .symptoms attribute. Previously, I already created a dynamic create box that displays the name of the body part. Now, I need to take each item in the .symptoms attribute and then add them to the options by creating it and then changing the innerHTML if possible. The best way to do this is by using a for loop. I have to do these things in order: create an option element IF there is a symptom still in the array, append this element to the select element, and then change its innerHTML. Below is a sample snippet of this algorithm.

```
for (i = -1; i<(part.symptoms.length); i++)
    {
      var option = document.createElement("OPTION");
      selection.appendChild(option);
      if(i!==-1)
        option.innerHTML = part.symptoms[i];
      else
        option.innerHTML = selectedPart;
    }
```

I changed the start of the loop to -1, such that the first option element has an innerHTML equal to the selectedPart. Since the select element inner HTML displays the text in the first option element, the first option element must have its innerHTML equal to the body part name. Now that the symptoms are added

to the select box the next part should deal with interpreting when an option is clicked to generate further symptoms.

6: Create an algorithm to match selected options with additional options.

This will be the backbone of my entire project. However, first I need to learn about eventListeners since that is the only way to recognize if the user has clicked the option in a select box. In order what must happen is: create a new select, add a RESET button, add symptoms to the table, and populate new options. I will add an event listener to the document that detects if a select option has been selected. I tried using .onchange, but that did not work. I found that .input would allow the function to work without directly referencing the select element itself. Also, to avoid any glitches or bugs, I will disable the previous select box to make sure that the user cannot change an option once they selected it. It may seem frustrating, but it will prevent any unexpected outcomes.

In the function, I will have to store the selected option in the form of a variable to be able to compare in my later function. I will then use this new variable and add it to the symptoms + diagnoses area. And create a new select box. All of this will happen separately from the narrowing down function.

```
document.addEventListener('input',function()
{
  selection.disabled = true;
  selectedSymptom = selection.options[selection.selectedIndex].text;
  myNode.firstChild.disabled = true;
  sContainer.innerHTML+= selectedSymptom + "<br/>";
  selection = document.createElement("SELECT");
  myNode.appendChild(selection);
  selection.style.display = "block";
  selection.style.marginTop = "1vh";
  addSymptoms();
},false)
```

Strangely enough, I needed the false as a parameter of the event Listener otherwise it did not work. addSymptoms() will now be the function I will work with. For this function I will narrow it down in two stages. The first checks if diagnosesList (the one that has all the diagnoses) and then the second and subsequent times check from the symptoms already narrowed down. I could have just spliced the original diagnosesList however I wanted to make it more simple and be able to refer to the

original diagnosesList. I will push the possible diagnoses as an object into tempArray from which I will have to alter later.

```
function addSymptoms()
{
  if (firstFilter == true) // Narrows down first set of symptoms.
  {
     for (i = 0; i< diagnosesList.length;i++)
     {
        if (diagnosesList[i].symptoms.includes(selectedSymptom)
              possibleDiagnoses.push(diagnosesList[i]);
     }
  }
  if (firstFilter == false) // Narrows down further set of symptoms.
  {
     for (i = 0;i < possibleDiagnoses.length;i++)
     {
        if (possibleDiagnoses[i].symptoms.includes(selectedSymptom))
           tempArray.push(possibleDiagnoses[i]);
     }
     possibleDiagnoses = tempArray;
  }
  firstFilter = false;
 }
```

7: Filter created options to make more sense.

Here is what I need to alter from the original raw array. I need to make sure there are no duplicates, or previously selected symptoms. This would create an infinite loop and also be bad user interface and experience. To achieve this, there are no built in functions for an array from which I can remove duplicates. However, I discovered a Set object that if set equal to the array would provide only unique values. I did experiment with it, but ultimately it did not work. So I created a new array called uniqueArray that would take all the values from the possibleDiagnoses and remove duplicate values. To do this, I will use the same .include function used to match the values. If the array contains the indicated element already, then remove it.

```
for (i = 0;i<possibleDiagnoses.length;i++) // Identify duplicate Values
  {
     for (j=0;j<possibleDiagnoses[i].symptoms.length;j++)
```

```
    {
        if (!(uniqueValues.includes(possibleDiagnoses[i].symptoms[j])))
            uniqueValues.push(possibleDiagnoses[i].symptoms[j]);
    }
  }
```

Next I need to make sure that previous symptoms have not been included so as to not create an infinite loop. To do this, I need to add another condition to the duplicate values if statement.

```
possibleDiagnoses[i].symptoms[j] !== selectedSymptom)
```

Now all unwanted values will not appear in the selection box. This will make it easier for me to call upon the same function without error.

8: Create a list of diagnoses, and display one with appropriate user selection.

I need to brainstorm different ways of figuring out when an amount of symptoms means that diagnosis is selected. The most straightforward way is by checking if the list of possibleDiagnoses is 1, or check if each symptom is included in the diagnosis. The first is probably the most straightforward. If possibleDiagnoses has an array length of 1, that means there must be one diagnosis that satisfies all the conditions.

```
if (possibleDiagnoses.length !== 1)
  {
    for (i = -1;i<uniqueValues.length;i++)
    {
      if(!(listSymptoms.includes(uniqueValues[i])))
      {
        uniqueValues.sort();
        var option = document.createElement("OPTION");
        selection.appendChild(option);
        if (i == -1){option.innerHTML = "Select";}
        else{option.innerHTML = uniqueValues[i];}
      }
    }
  }
  else
    showDiagnosis();}
```

Now I will work on the showDiagnosis() function which will handle showing the diagnosis and treatment. Once this happens, I need to do these things in order: access the diagnosis container, add the diagnosis into

the container, set some styling, and make sure no new select box appears. I will access it by using possibleDiagnosis[0] as there can only be one item in index 0.

```
dContainer = document.getElementById("diagnosisContainer");
  document.getElementById("finishedDiagnosis").innerHTML = possibleDiagnoses[0].name;
  dContainer.innerHTML += possibleDiagnoses[0].treatment;
  finishedDiagnosis.style.fontSize = "2.5vh";
  selection.style.display = "none";
```

9: Create the login and signup area in a new page.

Now that I have a rough draft of the minimum requirements for the website, I now need to create a login and signup area. I will create a new webpage, from which a button "Login" and "Signup" will allow access into the webpage. I need to be able to quickly switch between the two as well. I will add the database later, as for now I will work on the physical structure itself. I will take the traditional style of a login and signup tab with user input fields inside of a box. There are two static elements, which will be the login tab, signup tab, and Username & Password. The dynamic part is when the signup tab is selected which will add user inputs for Name and Email as well as remove the login button and add a signup page.

Login	Signup
1) Login Tab 2) Box Field 3) Username and Password 4) Login Button	1) Signup Tab 2) Box Field 3) Username, Password, Name and Email 4) Signup Button

I will use a label similar to that of the Level 5 Project to make this quick and simple. I will add a label tag to the signup and login tabs which will be connected to two radio buttons. Each radio button represents the state of the webpage. Since when the website loads, the login tab will be visible, that will be the default state.

```
#signup-selected:checked ~ #signup-tab{background-color:rgba(84, 35, 35, 0.804);}
#signup-area:hover{background-color:rgba(84, 35, 35, 0.29);}
#signup-selected:checked ~ #login-tab{background-color:rgba(84, 35, 35, 0.596);}
#login-area:hover{background-color:rgba(84, 35, 35, 0.29);}
#signup-selected:checked ~ #formcontainer{height: 53vh;}
#signup-selected:checked ~ #login-button{display: none;}
#signup-selected:checked ~ #signup-button{display: block;}
#signup-selected:checked ~ #inputcontainer #email{display: block;}
```

10: Add a database for handling all created accounts and login algorithms.

This will be the most difficult part of the project. To create a database, I don't really know how to use node.js or server side programming. I will instead use a localStorage that sets an array with user accounts (represented as objects). These accounts will have certain properties, so I will create a class with a username, name, email, password, diagnosis. Upon further observation and studying, I found that it is impossible to set objects using localStorage. Instead, I will have to use JSON operations such as .stringify and .parse to retrieve information on the main page.

```
function User(name, username, password, email) // User class function. Create users with properties.
{
  this.accountName = name;
  this.accountUsername = username;
  this.accountPassword = password;
  this.accountEmail = email;
  this.accountDiagnoses = [];
}
```

This will be the main function class that will be created upon signup. From here, I need to create a function for when the signup button is pressed. As previously mentioned, I will use JSON operations. I will also save the index for later use.

```
currentUser = new User(accountName,accountUsername,accountPassword,accountEmail);
  userDatabase.push(currentUser);
  localStorage.setItem('database',JSON.stringify(userDatabase));
  localStorage.setItem('user',JSON.stringify(currentUser));
  currentIndex = userDatabase.indexOf(currentUser);
  localStorage.setItem('index',currentIndex)
  location.href = "home.html";
```

However, there must also be user authentication. Else people can sign in with blank input fields. To resolve this, I created numerous handlings of errors that will return the function. For example, one such example would be a blank name input field. In addition, instead of using a plain old window.alert I will create my own custom alert that fits the theme of the rest of the website.

```
if (accountName === "" || accountName == null)
  {
```

```
    alertMessage.parentElement.style.display = "block";
    closeButton.style.display = "block";
    alertMessage.innerHTML = "Name is required."
    return;}
```

11: Make sure diagnoses are linked appropriately with the database.

The only dynamic thing that changes with each user account are the diagnoses. Every time the user is diagnosed, I must save it each and every time. And unlike the previous array of user accounts, there are various accounts affiliated with their corresponding diagnoses. I can use the index of the user account, and access its diagnoses (which is in the form of an array) and push values there and resave the database. This event will fire when the diagnosis is added to the diagnostic readout in the showDiagnosis function. These diagnoses are saved in the account information area, which will retrieve data from the database by getting the current index of the currentUser. Along with the diagnoses, the other information will be saved and will be retrieved from the database. This information, however, is static.

```
function showInfo() // Show the account information by extracting from database.
{
  document.getElementById("account-info").style.display = "block";
  document.getElementById("website-blur").style.display = "block";
  document.getElementById("account-info-name").innerHTML = currentUser.accountName;
  document.getElementById("account-info-username").innerHTML = currentUser.accountUsername;
  document.getElementById("account-info-password").innerHTML = currentUser.accountPassword;
  document.getElementById("account-info-email").innerHTML = currentUser.accountEmail;
  document.getElementById("account-info-diagnoses").innerHTML =
userDatabase[currentIndex].accountDiagnoses;
  if (document.getElementById("account-info-diagnoses").innerHTML == "")
  {document.getElementById("account-info-diagnoses").innerHTML = "Not Diagnosed.";}
}
```

12: Add quality of life changes.

Although I finished the minimum requirements for the project and the enhancement. I would like to add some changes. I think I will add a help tool, logout button, and create a popup box for the account information. The help tool was fairly easy to implement, as I just used display:none; and display:block;. The logout button was also easy, because since I am not using a database, I do not have to remove the currentUser from the website. They will have to log back in anyway, which will set a new user. The account information tool however was a bit more unique. I added a

background blur, and added some padding inside the div. The website blur also brought up and taught me an interesting property.

```
#website-blur
{
 display: none;
 backdrop-filter: blur(5px);
 z-index:2;
 position: absolute;
 left:0;
 top:0;
 width:100%;
 height:100%;
 animation: blur 0.4s;
}
```

Throughout this project, I also utilized keyframes which were very useful to create animations. And that is really all for this project. It was fairly difficult compared to previous projects, but it worked out in the end.