1.) While developing the web application using Ionic Framework we ran into a major problem. That problem was that webgazer technology for eye tracking was incompatible with ionic. As a team we decided to develop a web based site that contains the examinations of visual acuity, eye movement, and pupillary reflexes. We developed a website using HTML5 and JavaScript, this web site will interact with the patient and give the doctor access to patient statistics. The patient and doctor have separate login screens, the patient will have access to webgazer for the pupillary reflexes and eye movement examinations, for the visual acuity they will be shown different letters and need to differentiate between them.

2.) Github new link https://github.com/gmejia8/ValleyChildrenHospital

This is the code that will create a connection to the backend database. We added all the configurations that are needed for the database. Also, allowing the html file to read from the javascript file in order to access the database.

```
+            <!-- The core Firebase JS SDK is always required and must be listed first -->
+
+    <!-- TODO: Add SDKs for Firebase products that you want to use
+        https://firebase.google.com/docs/web/setup#available-libraries -->
+        <script src="https://www.gstatic.com/firebasejs/7.2.1/firebase-analytics.js"></script>
+
+        <script>
+        // Your web app's Firebase configuration
+        var firebaseConfig = {
+            apiKey: "AIzaSyABQ7QKlUS-jd9VMd7r8S7PjxPljicMfSw",
+            authDomain: "valleychildrenhospital.firebaseapp.com",
+            databaseURL: "https://valleychildrenhospital.firebaseio.com",
+            projectId: "valleychildrenhospital",
+            storageBucket: "valleychildrenhospital.appspot.com",
+            messagingSenderId: "592627206728",
+            appId: "1:592627206728:web:92a50c43a60cd02abde67d",
+            measurementId: "G-5SZ4K0HP0P"
+        };
+        // Initialize Firebase
+        firebase.initializeApp(firebaseConfig);
+        firebase.analytics();
+
+        const db = firebase.firestore();
+        db.settings({timestampsInSnapshots: true });
+
+        </script>
+
+            <script src="app.js"></script>
+        </body>
+    </html>
```

Added the addition of a general webgazer execute function, so when the button that is currently connected the the webgazer function. Webgazer will open when pressed, currently still under testing and this is a new addition to the main function of the code.

```
$(document).ready(function () {


    $("#btn").click(function (event) {

    webgazer.setRegression('ridge').setTracker('clmtrackr')
    .setGazeListener(function (data, elapsedTime) {
              if (data == null) {
                  return;
              }
              xprediction = data.x; //these x coordinates are relative to the viewport
              yprediction = data.y; //these y coordinates are relative to the viewport
          }).begin().showPredictionPoints(true);

        })

      });
```

This new addition allows the successful integration of the front end with the database, and allows us to execute the functions getting data, and saving data. There is still some more work needed in order to compare with the usernames being compared. We are currently more focused on how to successfully integrate webgazer.

```
12  + function renderPatientLogIn(doc){
13  +
14  +     let li = document.createElement('li');
15  +     let PatientUserName = document.createElement('span');
16  +     let PatientPassword = document.createElement('span');
17  +     let cross = document.createElement('div');
18  +
19  +     li.setAttribute('date-id', doc.id);
20  +     PatientUserName.textContent = doc.data().Username;
21  +     PatientPassword.textContent = doc.data().Password;
22  +
23  +
24  +     li.appendChild(PatientUserName);
25  +     li.appendChild(PatientPassword);
26  +
27  +
28  +     PatientLogIn.appendChild(li);
29  +
30  +  }
```

In developing the front end using bootstrap and chart.js to create charts for statistics. The login screen was connected to the backend using firebase. A modal was used for the login screen wich required javascript functions and event handlers.

```
var modal = document.getElementById('id01');
const PatientLogIn = document.querySelector('id01');
const form = document.querySelector('#add-PatientLogIn-form');

window.onclick = function(event){
    if(event.target == modal){
        this.modal.style.display = "none";
    }
    console.log('start');
}
```

Changing from out old method of creating and application, we change our field of direction and pivoted to creating a web application. This was necessary do to our primary concern being the administration of the visual exams. Our focus was to integrate webgazer as the eye tracker for the visual exams, but since the platform we were using will not support the third party library of webgazer, we had to shift into a web application. The changes you see above show the additions to the html files that create the web application.

All of our code is contained within the index.html and the myscript.js files. The main functions are currently the webgazer function, that takes in the value of the press of a button. This will tell the function to activate and open webgazer. We then added the rendering function that will append all the data from within the database and make it usable for the manipulation of the front end. In the app that we first tried to create, there were problems of connectivity with the front end. These issues have been resolved and have connected the frontend to the backend. Another big change was the addition of a working webgazer. Since ionic and angular did not support webgazer, the functionality became much easier to implement within just html and javascript code. The only functionalities left to implement for the frontend to access from the database, is some comparison queries that allow the doctor/patient to look for either specific days for exams, or for the doctor to look for specific patients. In short, the implementation of a search bar that searches through the specific doc (tables, and goes through each tuple) in the database.

A functionality that was added, was the ability for the patient to logout of the page, this was implemented as a side menu.

```
    <script>
        $("#menu-toggle").click(function(e) {
          e.preventDefault();
          $("#wrapper").toggleClass("toggled");
        });
```

```
        </script>
```