Installing and setup for Shannon Application.

This document details the setup to run the Shannon Applications (IOS App and Website)

Project Description: The goal of this project is to have a system to send user's Apple watch health information to be viewed by medical professionals at Shannon.

Clone the projects GitHub repository to get started
https://github.com/bturney1/Understood-ShannonProject

Firebase setup (Ethan Clark)

Login to Firebase at firebase.google.com

Navigate to Firebase console ("Go to console button)

Create a project if one has not already been created.

Select the project.

Here is where we can set up the Firestore database, Hosting, and Authentication.

At the top, we want to click add app. We will be adding: **IOS app and Website**.

1. For **IOS app** follow these instructions https://firebase.google.com/docs/ios/setup (must have a mac with Xcode 15 or later). The "GoogleService-Info.plist" will need to replace the one we currently have (Located in PatientApp/PatientApp on GitHub) The code integrations should not need to be added or changed.

2. For **Web**, follow these instructions https://firebase.google.com/docs/web/setup

Code integrations may not need to be changed.

You may need to install **node.JS and npm**. We will need this for **Firebase Hosting** also.

*node.JS install https://nodejs.org/en/learn/getting-started/how-to-install-nodejs

*npm install https://docs.npmjs.com/downloading-and-installing-node-js-and-npm

Within the ProfessionalWebpage/Scripts from GitHub, you need to replace code in **FirebaseConfig.js** with your own information located in Firebase by clicking the **Web app** we just added (in Firebase console), scroll down. Replace the code in FirebaseConfig.js with the part that looks like this:

```
const firebaseConfig = {
  apiKey: "API KEY",
  authDomain: "AUTH DOMAIN",
  projectId: "PROJECT ID",
  storageBucket: "STORAGE BUCKET",
  messagingSenderId: "MESSAGING SENDER ID",
  appId: "APP ID",
  measurementId: "MEASUREMENT ID"
};
```

Your API information will be filled in here.

Database setup (Ethan Clark):

In Firebase console, navigate to **Firestore Database.**

The setup for the database is as follows (case sensitive naming)

Collections: Patients, Notifications, ProviderInfo

Each patient document at this time is made by hand because Shannon has patient information stored in their own database. Separate integration and implementation would be required to connect these.

Each **Patient** documents needs these fields:

BOlow (number)

HRhigh (number)

HRlow (number)

address (string)

cell (string)

email (string)

firstName (string)

lastName (string)

p_ID (string)

*Note the email of patient must match their Authentication account in order for them to be able to log in.

Additionally, each Patient document will need **sub collections** named HeartData and OxygenData

The initial document for **HeartData** will need:

HeartRate (number)

TimeStamp (timestamp)

The initial document for **OxygenData** will need:

BloodOxygen (number)

TimeStamp (string)

These documents will be created by the IOS app, so the initial document and information do not necessarily matter, but I am including the setup for reference.

For the **Notifications** collection, each document will consist of:

Message (string)

TimeStamp (timestamp)

p_ID (string)

Notifications are created via the IOS app just like HeartData and OxygenData

The **ProviderInfo** collection consists of documents containing:

Name (string)

email (string)

userID (string)

More may be added, this is a simple setup.

*Note that the documents must be made for Authentication accounts meant for the provider. The emails must match in order to log in.

Firebase Authentication Setup (Ethan Clark):

Navigate to Authentication in your project in Firebase console and select "Get started"

Select Email/Password and then enable it. Click save.

Creating users:

Switch to the "Users" tab on Authentication. To add a user click "Add user".

Emails that match the email of a Patient will be used for logging into the IOS app. All other emails will be used for logging into the Web app, and must match an email in the ProviderInfo collection.

Deploying the website to firebase hosting:

Making sure that you have **node.JS and npm** properly working, and/or **Firebase tools** installed (Firebase CLI) https://firebase.google.com/docs/cli, navigate to the root directory of the local repository (where the files like "PatientApp" and "ProfessionalWebpage" can be seen) using the command prompt OR Firebase CLI command window (You will need to login to your Firebase account). Example for Windows cmd: "cd C:/Users/username/source/repos/Understood-ShannonProject"

From here enter "firebase init"

When the console asks if you are ready to proceed, type Y then hit enter.

Using the arrow keys, move the blue selection down to "Hosting: Configure files for Firebase Hosting..." then press space to select it. Press enter.

When the console asks what you want to use as your public directory, backspace on (public) and then enter "ProfessionalWebpage"

*****IMPORTANT**

Next you will be asked "Configure as a single-page app..." Type N then press enter.

***OPTIONAL**

"Setup as automatic builds and deploys with GitHub" I do not personally use this but you may if you like. Type y or N for whichever you prefer.

If you are asked to overwrite any files, type N and enter for each.

Now that Firebase Hosting has been initialized, type "firebase deploy"

The website should successfully deploy and display the link to access the site.

Use the Hosting URL in a web browser to get to the website.

Installing the IOS App (Ben, Victor):

Pre-Requisites:

- A MacOS device running XCode version 15 or newer.
- An iOS device running iOS version 15 or newer

In XCode on the MacOS device, locate the repository of the project, and select the PatientApp directory as the active project.

Plug your iOS device into the MacOS device running XCode.

At the top bar of XCode, select the hostname of your iOS device as the build target.

In the top left of XCode, activate the “play” button to start the build process.

Once the build completes, the app can be found on the home screen of the iOS device.