Project Documentation

This project hosts a webapp with a database backend. Users should be able to log in to navigate their own account. If they do not have an account, they will be able to register for one by selecting “Register for an Account”. Their account should store their user information (name, email, etc) and their workout goals. Each of these can be displayed and edited on separate pages within the app. At this point, all the information is private to ensure legal standards, but ideally, this could be expanded to include sharable elements that other users could see. All of this was mostly done using Azure Active Directory B2C, dealing with all login and sign-up requests. This would not be too difficult, it would most likely involve a “post” function that would copy information from the user account into a publicly accessible model.

This is written in mostly C# and HTML, using .NET7. The webapp is deployed using Azure. The front-end of this app is a variety of razor pages that provided tools in order to use both HTML and C#. These pages are simple and efficient. The backend of the project included a MongoDB Atlas and is utilized to manage the database for the app. Models were built in this and were used to carry out the aspects of our webapp. This was also used to build a cluster database to store all user data and exercise data. In order to do this, it collects the user's usernames, passwords, and emails so that they can be referenced during the login and when looking at the profile.

One challenge we faced was getting an app development environment working. We initially started trying to get Xamarin running which allows users to build Android and iOS apps directly from the same project. This worked locally, but we had a lot of problems getting it to work with our GitHub repository. There was especially an issue with the MacBook using Visual Studio along with being able to push and pull from the GitHub repository. We switched to trying to develop a webapp to avoid building with Xamarin. Initially we wanted to use .NET Core 3.1 to build a MVC (model, view, controller) webapp but were having difficulty with the creation of a CI/CD pipeline to Azure deployment so that we can all push changes that will automatically update our deployed web app. Also, .NET Core 3.1 will not be supported after 12/13/2022 (Ironically the day the project is due) so we switched again to develop a webapp with .NET 6 using razor pages.

The challenge of getting an environment working that was compatible for everyone ate away at a lot of our time. We would spend hours trying to configure computers, download programs all for it to just not be compatible for everyone.

Once we were able to create an environment everyone was able to work on, we created a repository for our webapp. In this repository we utilized many of the aspects of GitHub. We each created a branch so that we would not overwrite each others work and so that we were always working on the most up to date version. Whenever we each made a contribution that we knew fully worked, we would push it to our individual branch and then merge it with the main branch. This helped us have seamless team work and negated anyone doing work twice. This also kept our work organized and neat.

Most of the difficulty that came after getting the environment up and running has been getting the database set up. Naturally, it has also been difficult to switch from working in C++ to working in C# and HTML, languages that we do not all know. Utilizing videos and example code has helped immensely, however it is time consuming as well.