Wilson McGill

Project 1: Ears Project

Documentation:

**Overview**

Original Goal: Rebuild the old application into a new web application using Ruby on Rails. The original software was having licensing issues (fixed) and was needing to be redone, so we were tasked with creating a scalable web app to replace it.

What is done as of end of Spring 2023: There is a basic working version of the new web app. It contains all the base functionality, signing in, adding clients to test, performing test on said clients and submitting tests. There are some minor visual discrepancies, but the backend is up and working and the webapp still performs the set-out goals.

What needs to be done: The webapp still needs the remaining tests to be implemented, including RDDT and others along with the lists for those other tests and the working test. There is 5 lists per test type and currently only 1 test and list work for 1 type of test. The app also needs much more rigorous testing and ensuring of no minor bugs / visual problems. This may require some reformatting of design choices.

**Install instructions**

To install the old software:

1. Request keys from Deborah
2. Visit <https://www.dichoticsinc.com/> and download the software
3. Input the license keys and begin

To install the new web app:

1. Set up a Ruby on Rails environment on your system. Its in Ruby on rails 6 with ruby version 3.1.1 (make sure the ruby version is correct in gemfile on your local branch)
2. Get access to the Github through Deborah and create your own local branch to make changes from
3. Begin working!

**Functional description and known problems**

The system is effectively just a standard Ruby web app like you have created in software engineering. It uses ruby’s built in data base and stores users/clinicians(these are the same, basically just the people running the test), clients(people being tested), and tests(results of the test). It should work properly and uses bootstrap 4 for the styling.

There has been issues with bootstrap features not working and this is why some pages contain direct links to features at the top, so if additional bootstrap features do not work try that. There has been issues with the test page not updating for certain people and no reason why, the javascript script on the test page will sometimes not provide errors if changes made do not work, it will just stop outputting the values. ENSURE ANY CHANGES TO THIS PAGE ARE CORRECT BEFORE APPLYING or just implement tiny changes at a time to the test page. This will save huge headaches when troubleshooting the script. You may need to revert github commits if you cannot diagnose why the javascript is not working (this was necessary in initial development).

**General tips for success**

Do not commit large amounts at once. Commit as you go as soon as features are enabled. This creates less crunch before deadlines and smooths out any reverting necessary if features do not work. Use all tools at your disposal, team members, ai, professors, etc. Ensure your team feels comfortable pointing out problems with code and ensure no one gets butthurt about suggestions or changes. Always outcome oriented, do not sit on options or problems, pick the agreed upon direction and move towards it.

Use the Ruby on Rails Demos and Deets (google it) for basic examples of setting up databases, relations, etc. Work based off these for general Ruby on Rails knowledge if you do not have it yet.

**Extra Information**

The code contains documentation for all major features done by each group member, continue adding to this, ESPECIALLY WHEN CHANGING THINGS IN AREAS OF CODE YOU DO NOT NORMALLY WORK IN. I.e., if you tend to work on the test pages, and you change something in the navbar PUT A QUICK COMMENT, this will save your team members’ lives.