Test Report

To: EC464

From: James Knee, Anish Sinha, Howell Xia, Jilin Zheng

Team: 2 – ExerSights(AI Coach/PT)

Date: 04/13/25

Subject: Final Test Report

---

1 Required Materials

1.0 Hardware

- A computer and mobile phone
- Integrated and/or connected camera
- Spacious and flat area

1.1 Software

- Internet connection
- Web browser that can run JavaScript

2 Background Information

We have configured our program to be a web app accessible via the URL which can be found here: https://exersights.web.app/. We have developed several different interactive webpages using React. The user will click on the URL to launch the program, which will run locally on the client-side.

Once the program is running, the user can navigate between the different pages. The *Home* page serves as a welcome to users and contains a video tutorial and disclaimer. The *Catalog* page lists the current exercises and allows users to navigate to them. The *Program* page allows users to create custom workout programs. The *FAQ* page contains answers to commonly asked questions, limitations of the app, and a contact form allowing users to submit feedback or requests for new exercises. The *About* page contains information on the developer team. The *LogIn/LogOut* button allows users to link a gmail account to the current session. We have incorporated user-saved exercise settings, workout programs, and pinned exercises to the user accounts.

Navigating to an exercise page from the Catalog will request access to a camera. Once given permission, the center of the page will display the camera feed, the computer vision model, and a feedback/configuration panel. When the user is in frame and attempts the exercise, the panel will display exercise-specific feedback based on joint and limb angles and location. The user can configure the exercise and feedback settings to adjust the exercise to their liking.

3 Setup Procedure

3.0 Connect one of our laptops and phones to the internet.
3.1 Place the laptop onto a desk or bench, with the integrated webcam facing forward.
3.2 Connect an external webcam to the laptop in addition to the integrated.
3.3 Prop phone up on the desk. Perhaps use a water bottle to keep the phone vertical.
3.4 Open the URL found here: https://exersights.web.app/

4 Testing Procedure

4.0 Load application and show the different web pages focusing on single page implementation.
4.1 Open an exercise page of the tester's choice, we suggest squat as it is easiest to demonstrate.
4.2 Demonstrate user login. Show the ability to switch between multiple cameras.
4.3 Show the computer vision model landmarks within the webcam feed.
4.4 Perform an exercise which will demonstrate audio/visual feedback (test accuracy for 10 reps).
4.5 Demonstrate the help modal, timer, and video upload feature.
4.6 Demonstrate the settings modal. Have the user change the targets and retest the exercise. The user can also test the voice select, angle display toggle, and skeleton display toggle.
4.7 Switch to a mobile device and demonstrate how you can add the Progressive Web App (PWA) and once downloaded use without internet.
4.8 Navigate to the Frequently Asked Questions (FAQ) page and demonstrate the feedback form. Show survey responses.
4.9 Demonstrate exercise program feature.
4.10 Present the Firestore database and its stored user information.
4.11 Demonstrate the GitHub Continuous Integration/Continuous Development (CI/CD) pipeline, specifically the security scanners.
4.12 Show/perform system latency tests, angle accuracy measurements, and website availability.

5 Target Metrics

5.0 The feedback will be real-time based on the camera feed, with less than 200ms of delay.
5.1 The system will accurately calculate/detect joint angles, with less than 5^{o} margin of error.
5.2 The app must be available to users 24/7.
5.3 The app must ensure the visual feedback is visible to users from a distance of at least 5 feet.
5.4 The repetition count accuracy must be at least 90% given the model is stable.

6 Target Usability Objectives

6.0 The app should be able to run from any chosen computer or mobile device with a responsive user interface (UI).
6.1 The UI must be intuitive. Instructions must be detailed and easy to follow.
6.2 The app must provide personalized user accounts with Google login, storing saved exercise settings, workout programs, and pinned exercises.
6.3 The default “correct” form provided by the app will be based on physiological research literature and other reputable sources. References must be listed for each exercise.
6.4 The team must ensure code security by incorporating security scanning into the CI/CD pipeline.

7 Measured Metrics

7.0 The feedback latency was measured to be approximately 43ms, well within the original requirement of 200ms.

- The latency of our system is dependent on how quickly the computer vision model can calculate the human body landmarks and pass them to the detectPose method. Everytime the detectPose method gets called, the feedback updates. Thus, we were able to calculate the feedback latency by time stamping/logging the detectPose method and measuring the average time difference between each call.
- We measured the latency for both the TasksVision model (which we currently use for our app) and the older legacy Pose model (which we used in the first semester). The results are shown below.

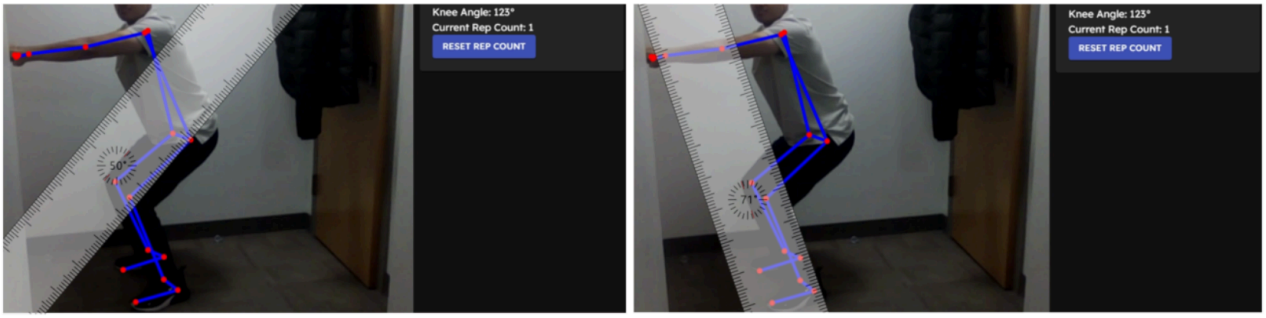
Trial #	Pose Model Latency (ms)	Tasks Vision Model Latency (ms)
0	33.35954198491482	40.17283236980438
1	33.33729729716842	42.24635761620983
2	33.271631206181034	44.49808917227824
3	33.20324675293712	42.09617224834752
4	33.38036809815951	42.26687116564417
5	33.434666666984555	42.41165644186406
6	33.309248554913296	44.10306748437004
7	33.39238095283508	42.44832214813104
8	33.53850931632593	42.04844720467277
9	33.26433566483584	46.52785388138741
AVERAGE	33.34912264952557	42.88196697327094

- As shown the average latency for both models was calculated to be well below the maximum 200ms requirement. The hosted app uses the TasksVision model which has an average latency of 42.88ms.

7.1 The system calculated joint angles with an average margin of error at 2.8°, which is well below the maximum requirement of 5° margin of error.

- We calculated the actual angle accuracy by recording a video of the squat exercise. We then chose 5 frames at random where we manually calculated the knee angle using a protractor software. To determine the margin of error we calculated the difference between the actual angle from the protractor and the angle shown in the feedback panel. This methodology is shown in the images below.

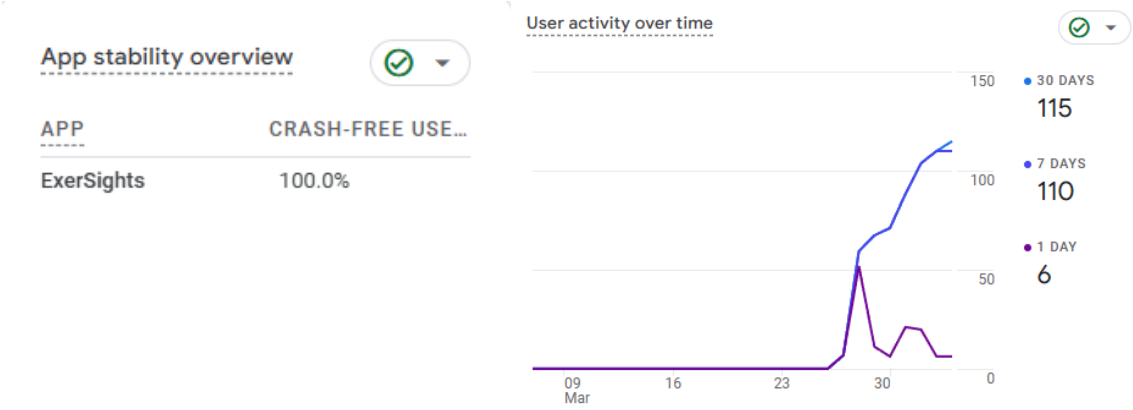
- We measured the angles from the horizontal line and obtained an angle of 121° (50° + 71°). Our application gives a measurement of 123°, which is a 2° margin of error.



- The margin of error data for the 5 randomly chosen frames is shown below. The average margin of error was calculated to be 2.8° which is below the maximum 5° requirement.

	Frame 1	Frame 2	Frame 3	Frame 4	Frame 5	Avg
Margin of Error (°)	2°	4°	2°	2°	4°	2.8°

7.2 The web app has been available to users 24/7. Ever since the site was first hosted in October 2024, there have been no outages. The image below from the Firebase Hosting Analytics page demonstrates 100% crash-free users for our application.



7.3 The visual feedback feature allows feedback to be seen further away via a colored border around the webcam output. Previously, we were only able to read the text feedback standing 1.5 feet away from the screen. With the new colored border, it is possible to stand 5 feet away from the screen and still be able to see the border color reflecting the state of the user, e.g., completed a repetition in green, in the process of completing a repetition in yellow. The repetition count also briefly shows in larger font in the middle of the webcam output whenever a repetition is successfully performed. This was shown during the both the second prototype and final tests, where the user could view the feedback from 5 feet. However, we do acknowledge that the exact visibility distance varies with external factors such as screen size, lighting, user eyesight, etc.

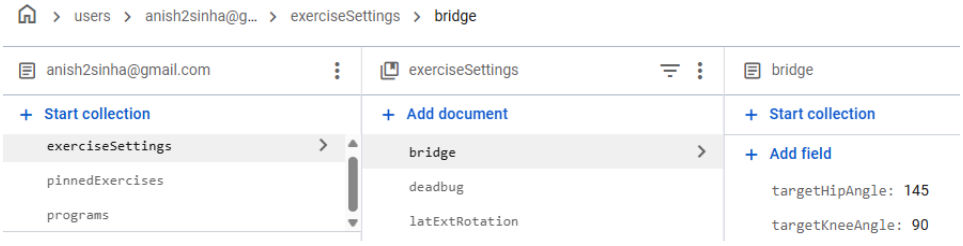
7.4 The repetition count accuracy is well above 90% given the model is stable. During the final test, we demonstrated 100% accuracy on the squat exercise as all of the repetitions were registered (10 were performed in total). We individually tested the push up exercise for 50 repetitions with all 50 being registered, yielding 100% accuracy. We also tested the lateral external rotation exercise for 50 repetitions with 47 being registered, yielding 94% accuracy. A few of the exercises (such as the lateral external rotation) have slightly lower accuracy due to the joint positions during the movement (based on the exercise, some joints can get covered by others), but the majority of our 15 exercises do not have this issue and have close to 100% accuracy.

8 Final Status of Usability Objectives

8.0 The app has a responsive UI and can be run from any laptop or mobile device. On mobile devices, the web app provides a PWA option which allows users to download it to their home screen. The PWA functions similar to a regular iOS/Android native mobile app. The PWA also caches the app to the device, allowing for offline use. A video demo of the PWA can be accessed at the following link: https://www.youtube.com/shorts/1x5IxiZRYbU

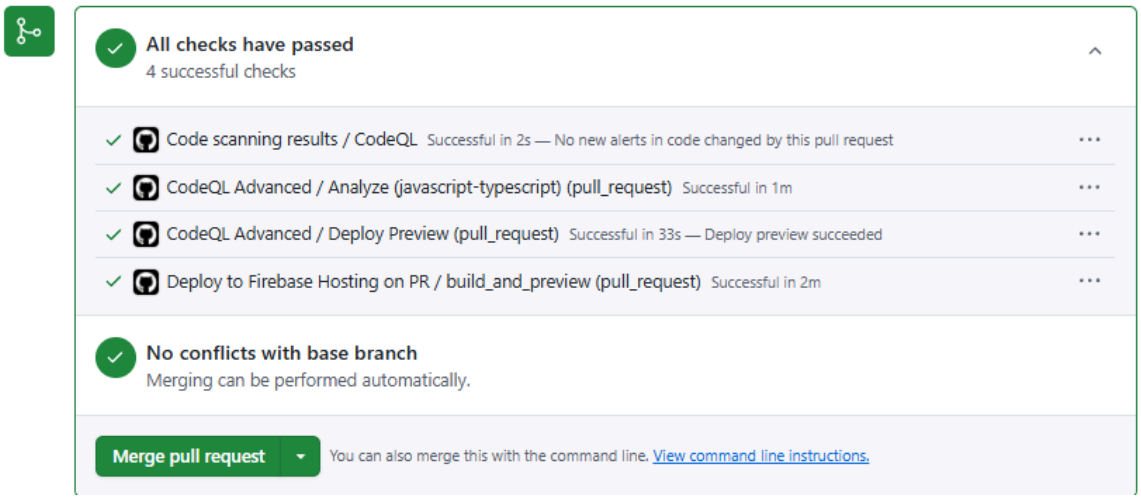
8.1 To measure the ease of use of our app, we started a survey. The survey seeks feedback and satisfaction ratings from users through a series of questions, including one on ease-of-use. The majority of users reported that the app was “extremely easy” to use with the remaining users reporting it to be “somewhat easy”. No users reported the app to be difficult to use. We received 9 responses in total (as of writing this report). A snapshot of the survey results are shown below.

8.2 The app provides personalized user accounts with Google login, storing saved exercise settings, workouts, and pinned exercises. We recently added a new feature which gives users the option to view and save a summary of their exercise after completion. These summaries are then available in the “My ExerSights” page. All of this data is securely stored in our Firestore Cloud database.



8.3 The default “correct” form is based on physiological research literature and other reputable sources. References for each exercise are listed on our README at the following link: https://github.com/AI-Coach-PT/ExerSights#references

8.4 We ensure the security of our codebase by incorporating CodeQL security scanners into our CI/CD pipeline using GitHub actions. These scanners check for vulnerable imports, memory leaks, code malpractice, etc.



9 Conclusion

Our application meets all of the MVP requirements promised in the PDRR (Fall 2024) as well as those listed in the CDR (Spring 2025). The system provides low-latency real-time feedback to users with high accuracy on repetition count and joint angle calculations. The colorful visual effects allow the feedback to be visible to users at far distances (>5 feet). Since the site was hosted in October 2024, there have been 0 site crashes, ensuring 24/7 availability with 100% crash-free users.

Based on the survey we conducted, satisfaction amongst users is very high. Most users report the UI to be intuitive and easy to use. The UI is also responsive to all screen sizes and provides mobile users with a PWA option, which can be downloaded onto the homescreen and used offline. Users have access to personalized accounts (through Google login), which store their custom exercise settings, workouts, and pinned exercises. All exercises are based on physiological research/articles, and the references for each are listed on the README. We ensure code and data security by incorporating security scanners and hosting the database on Firestore Google Cloud, a reputed and secure provider.

All of these functional requirements and features provide an enjoyable experience for our users. As indicated by the recent survey results (link), the ExerSights app is successfully helping its users improve their exercise forms for safer and more effective workouts.