ASSIGNMENT 2: PART A

TASK A: Build, deploy, and run a Containerized Application using GCP.

LINK TO GITLAB: https://git.cs.dal.ca/rtyagi/csci5410_b00872269_ruhirajnish_tyagi/-/tree/main/assignment-2

Step 1: Firestore database before running the apps and image and container snapshots on Docker.

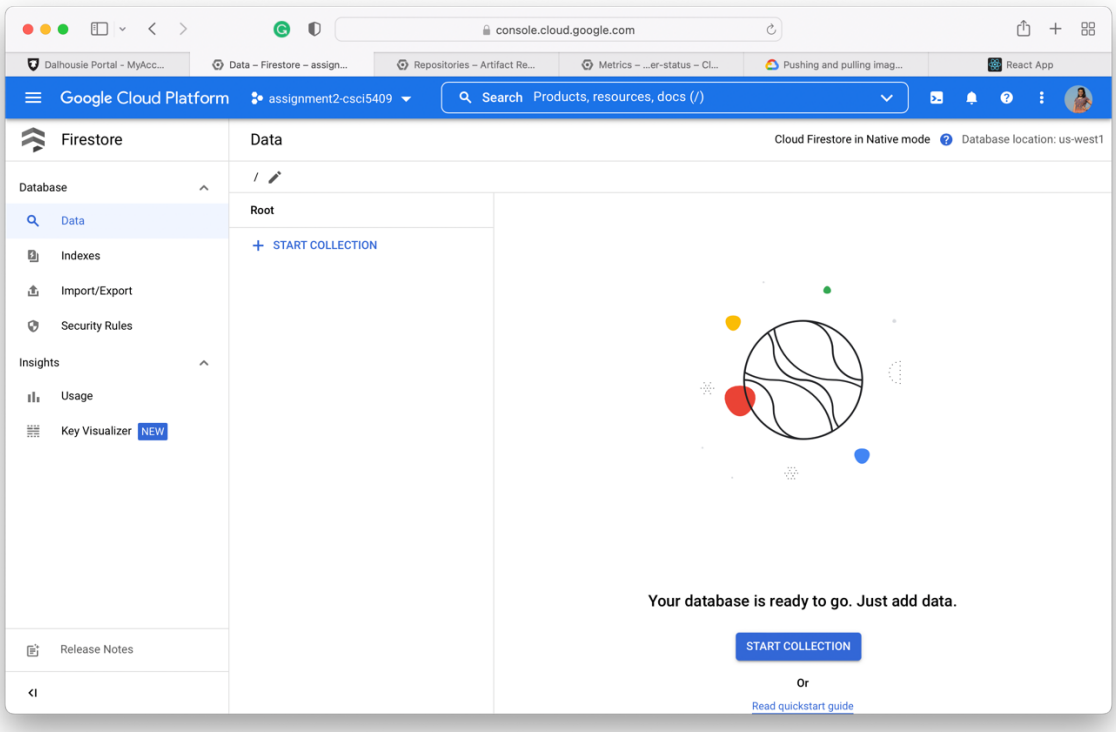

Figure 1. *Screenshot of empty Firestore on GCP.*

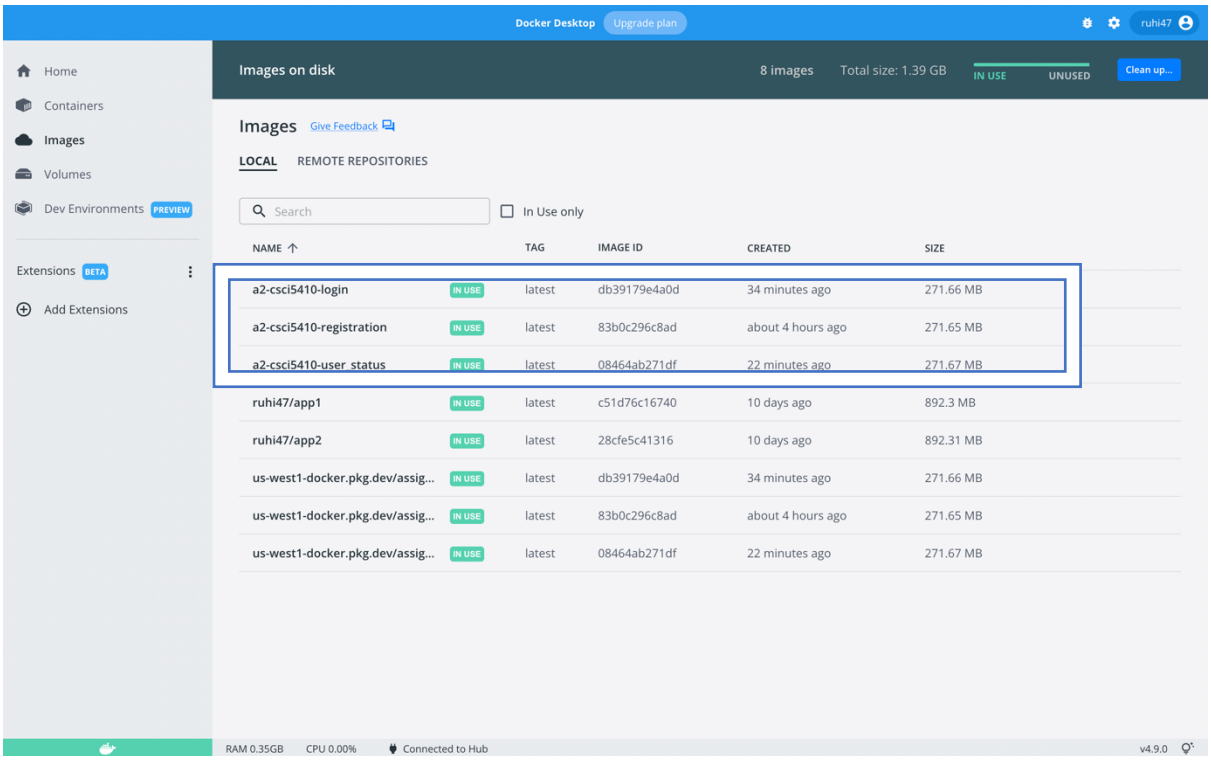

Figure 2. *Screenshot of images created of login, registration, and user status on Docker.*

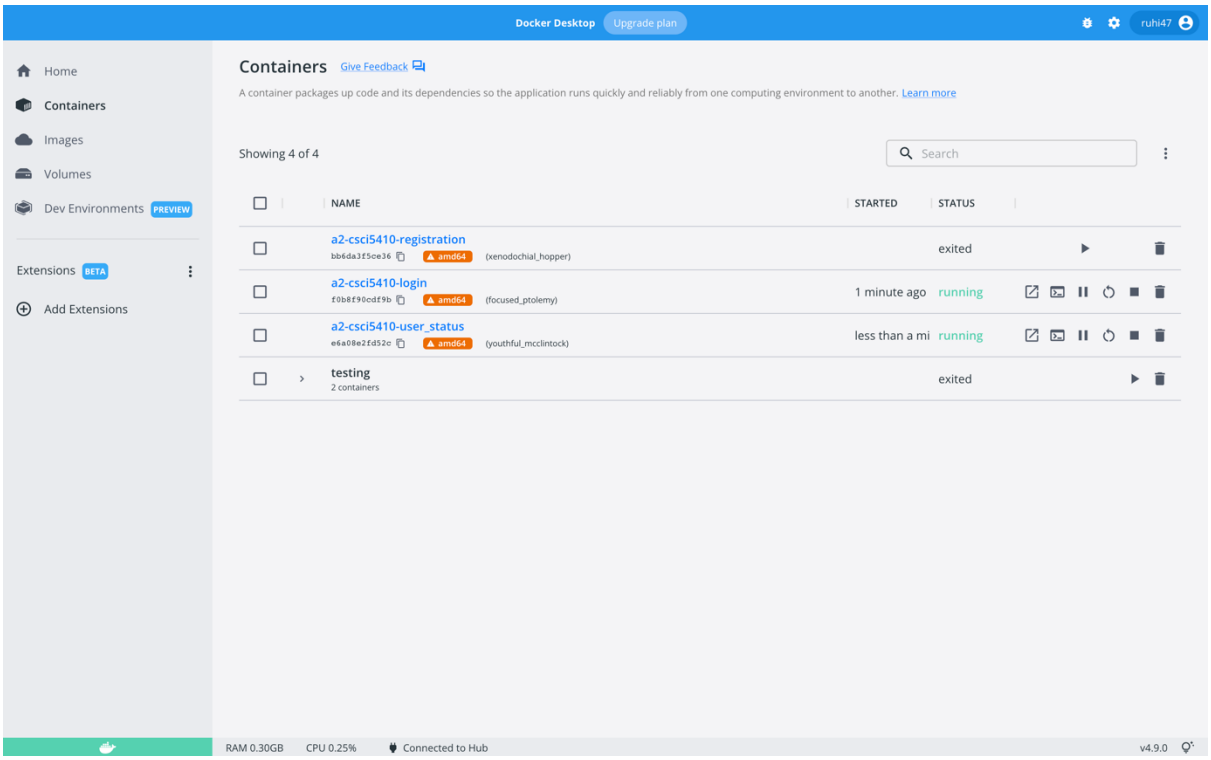

Figure 3. *Screenshot of respective containers of images on Docker.*

Step 2: Test on registration module demonstrating successful working of the module.

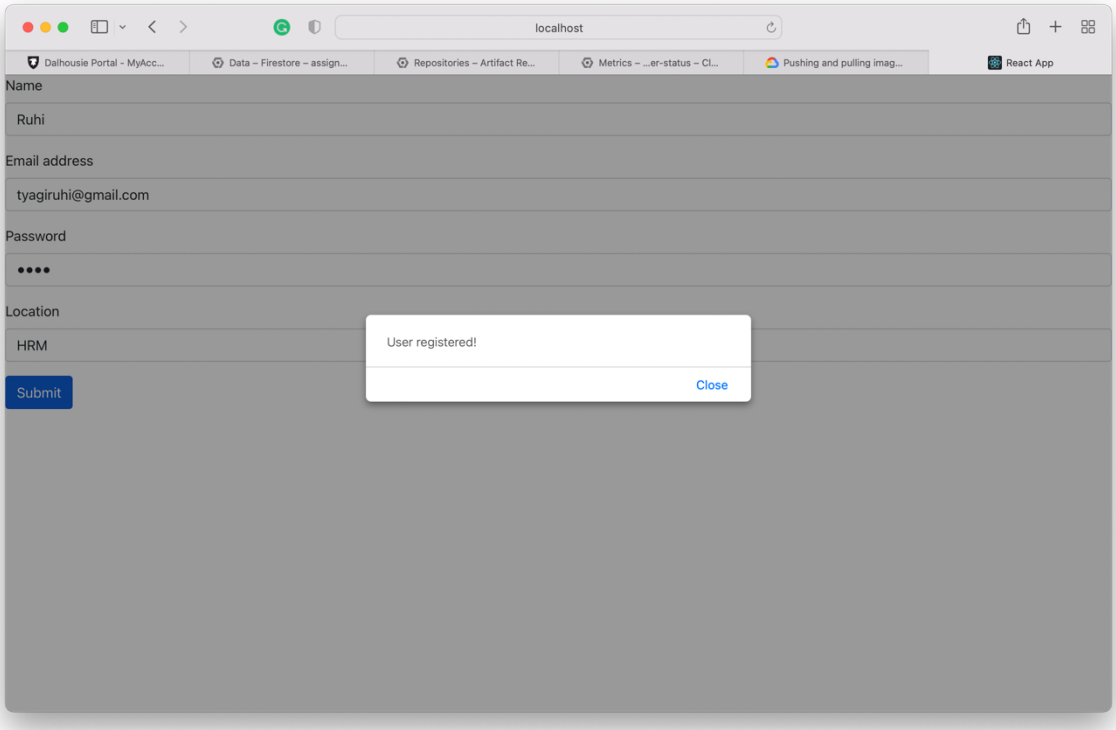

Figure 4. *Screenshot of SUCCESSFUL user registration.*

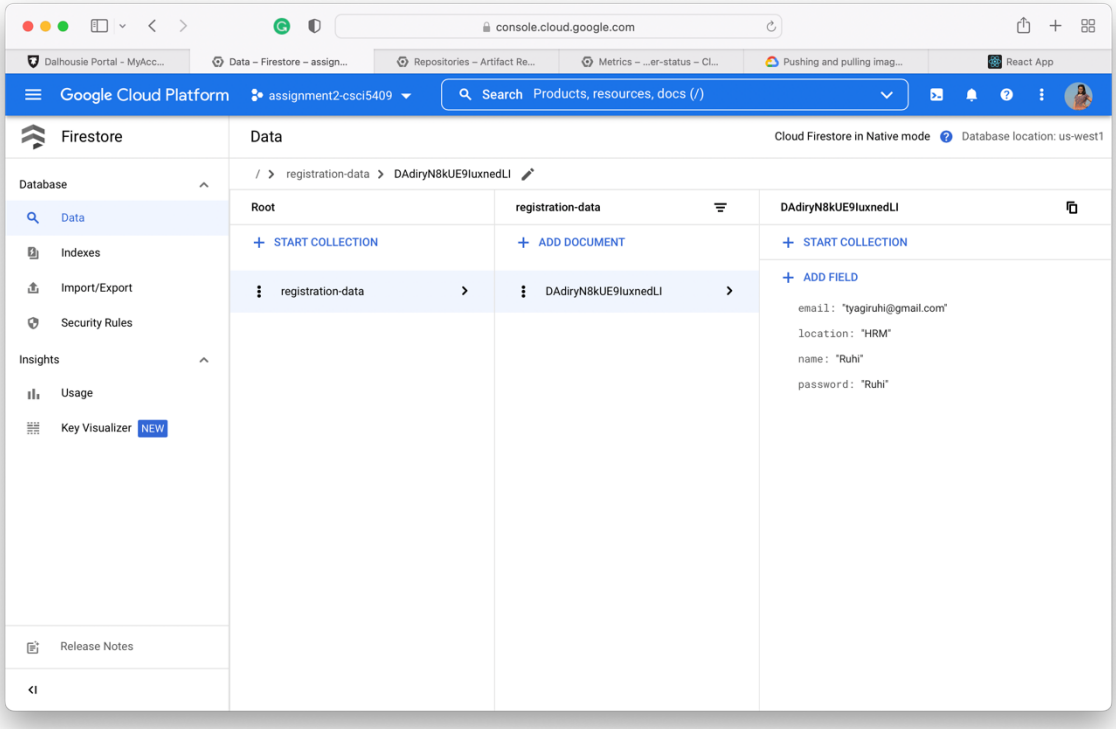

Figure 5. *Screenshot of updated Firestore on GCP after user registration.*

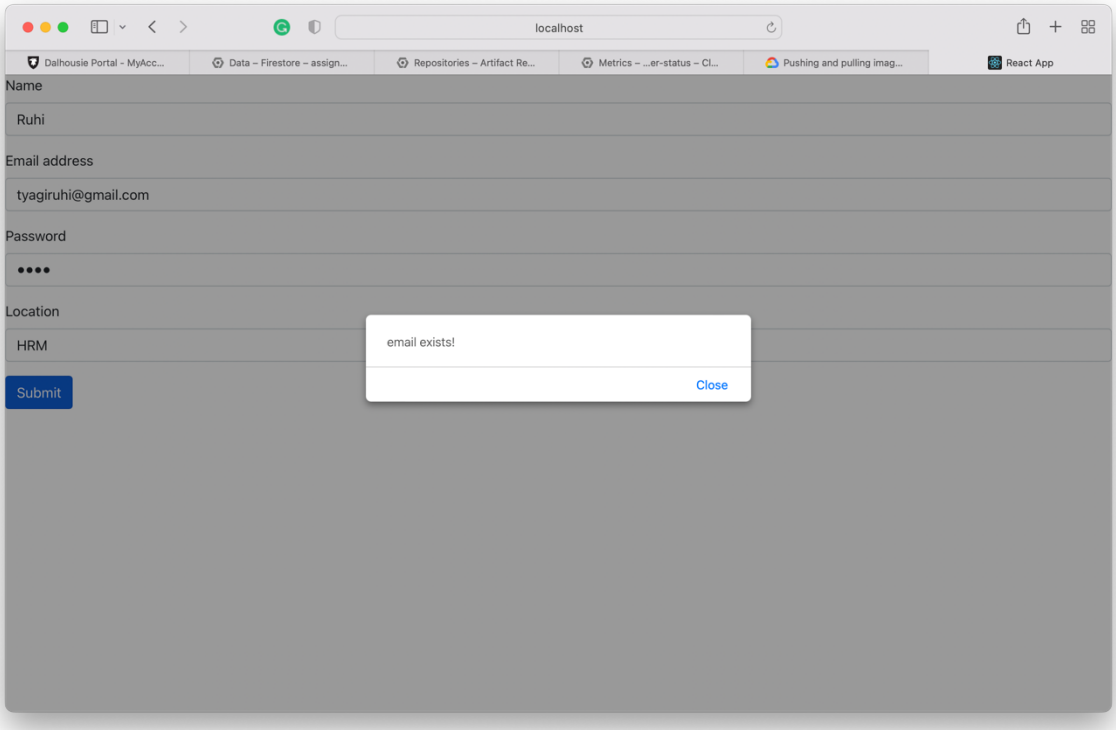

Figure 6. *Screenshot of FAILED user registration with an existing email address.*

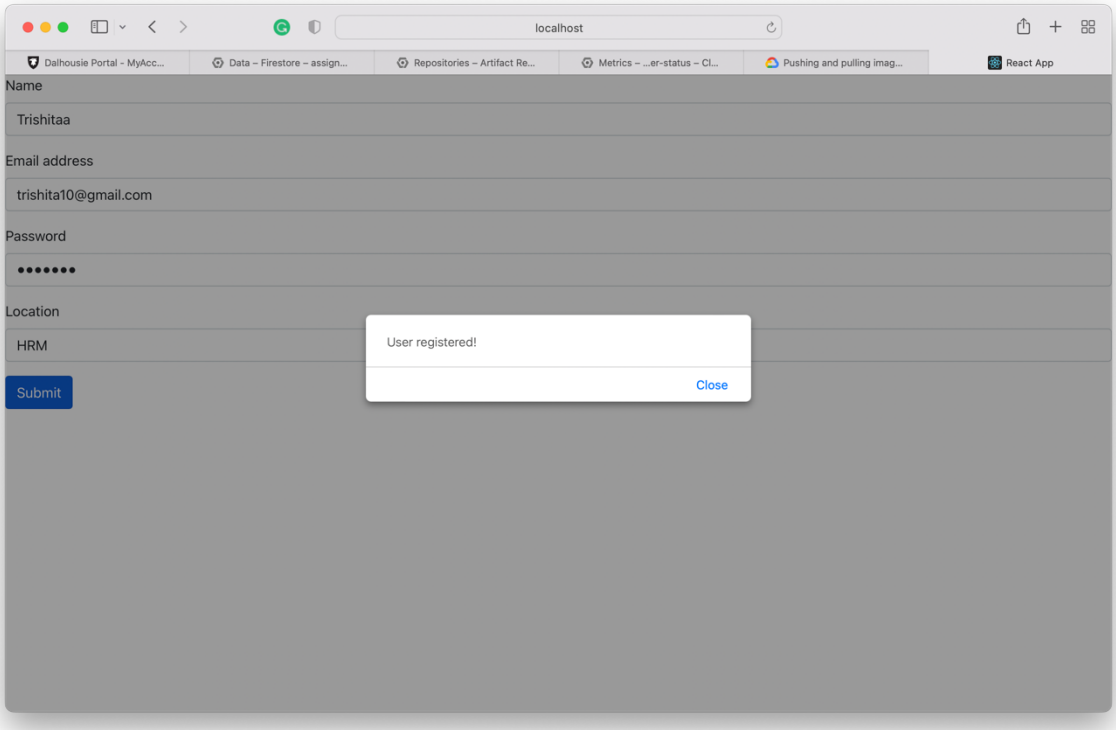

Figure 7. *Screenshot of another SUCCESSFUL user registration.*

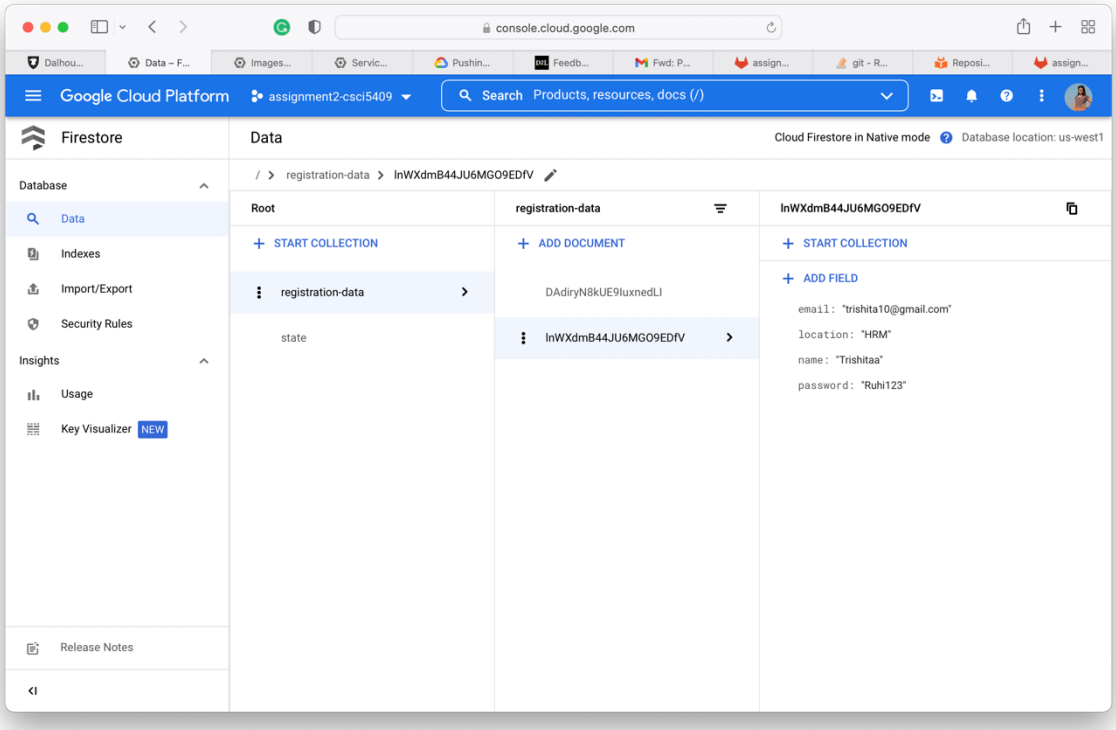

Figure 8. *Screenshot of updated Firestore on GCP after user registration.*

Step 3: Test on login module demonstrating successful working of the module.

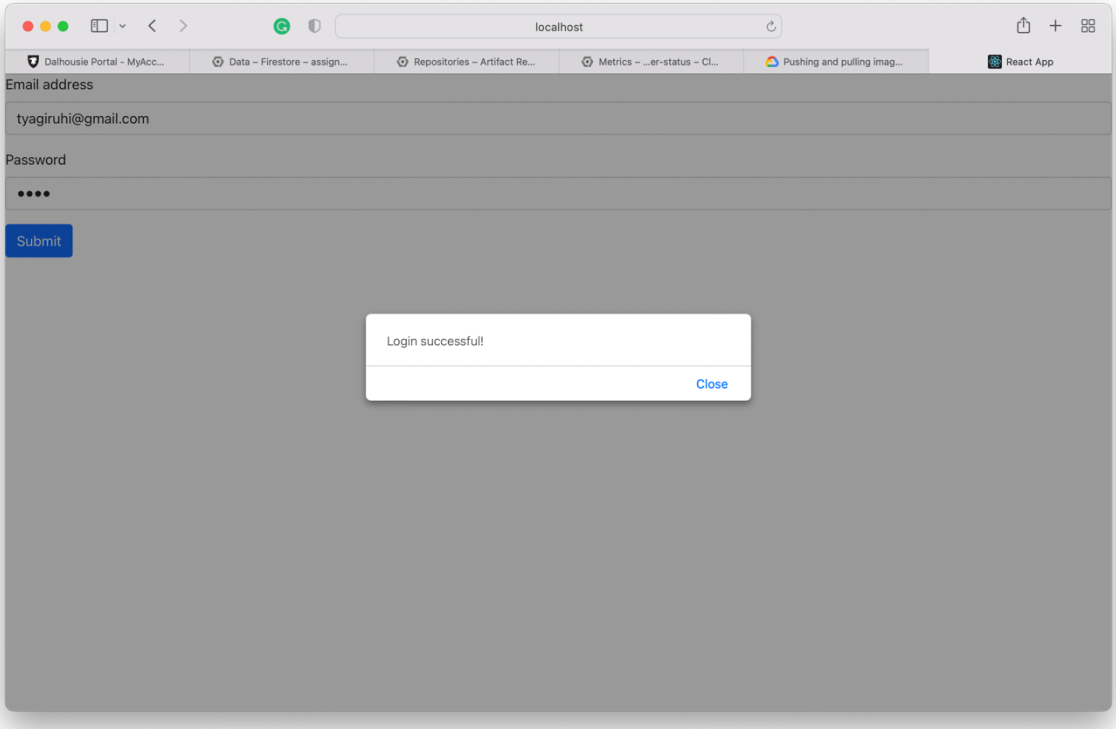

Figure 9. *Screenshot of SUCCESSFUL user login.*

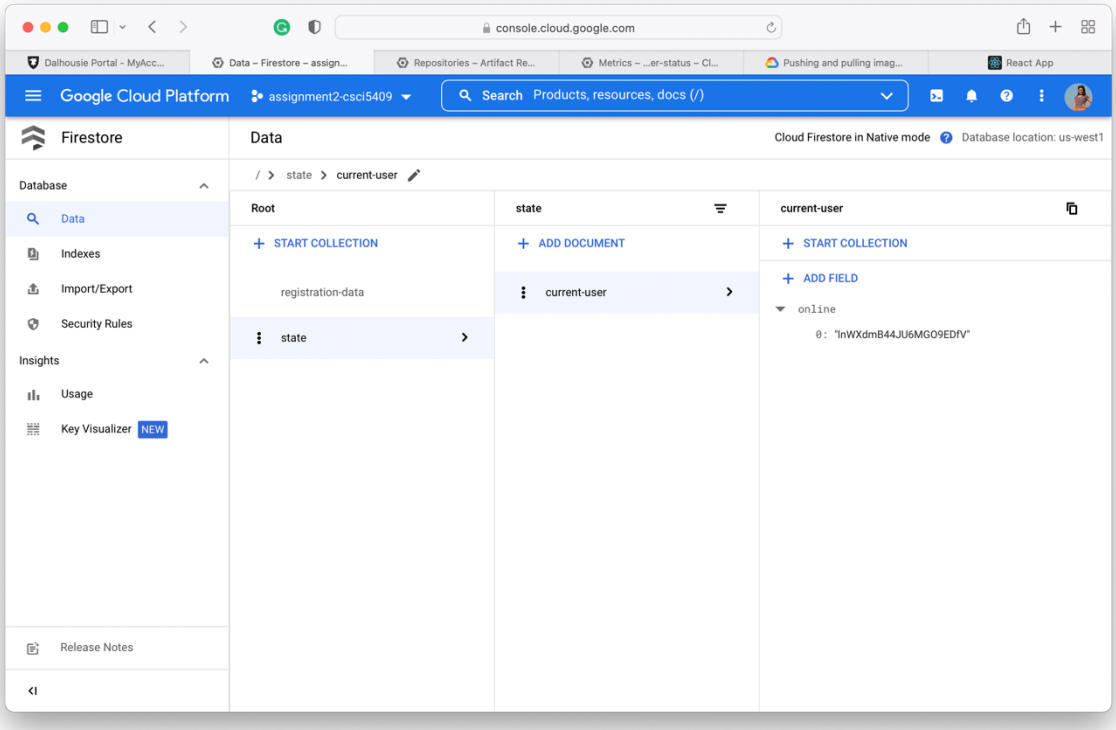

Figure 10. *Screenshot of updated Firestore showing user login and status change to ONLINE.*

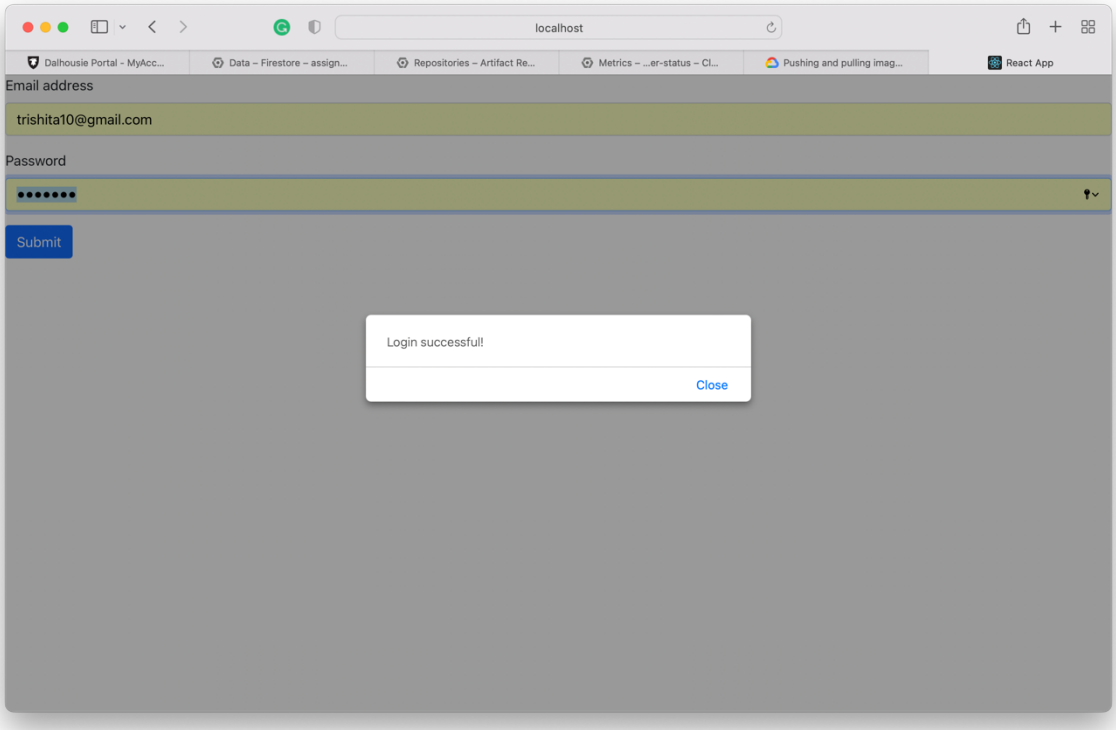

Figure 11. *Screenshot of another SUCCESSFUL user login.*

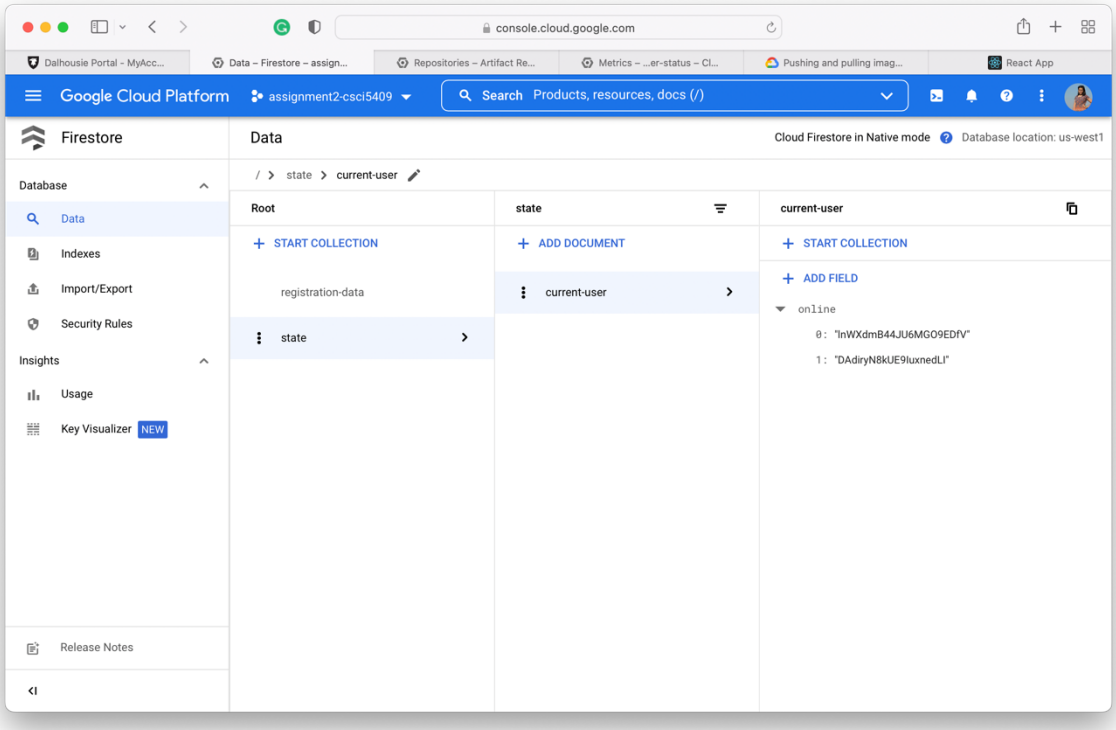

Figure 12. *Screenshot of updated Firestore showing user login and status change to ONLINE.*

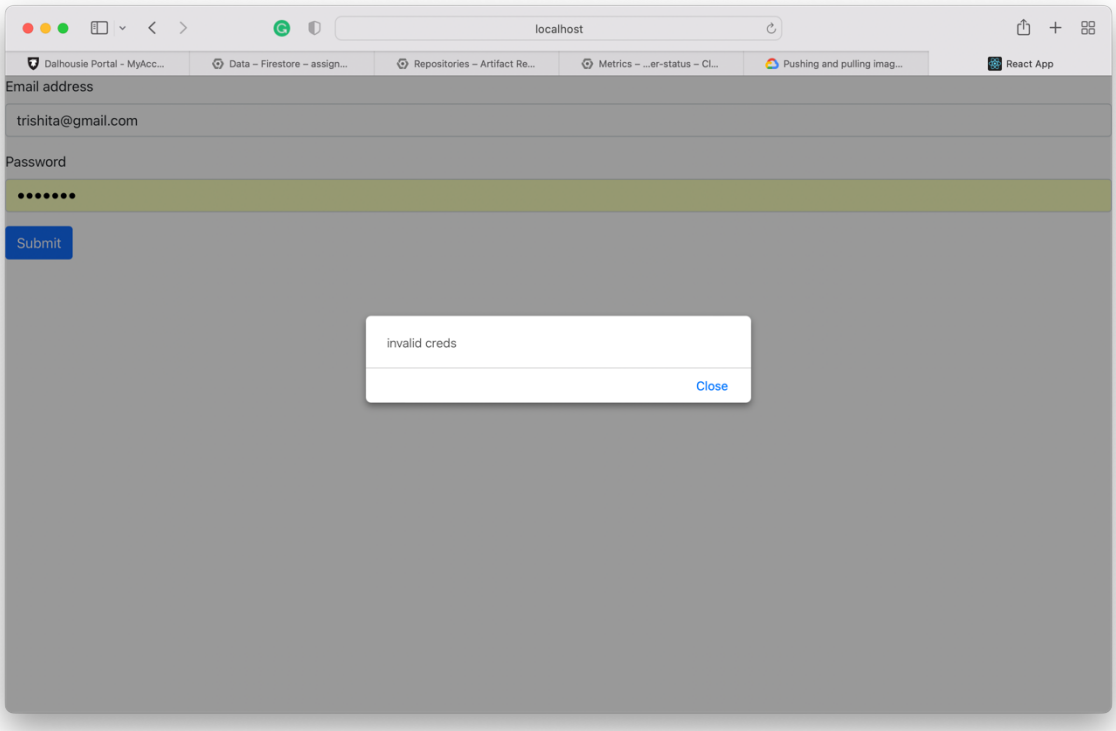

Figure 13. *Screenshot of FAILED user login with an non-existing email address.*

Step 4: Test on user status demonstrating successful working of the module.

As it can be seen in Figure 9 and 11, the users are online in order they logged in shown in Figure 14.

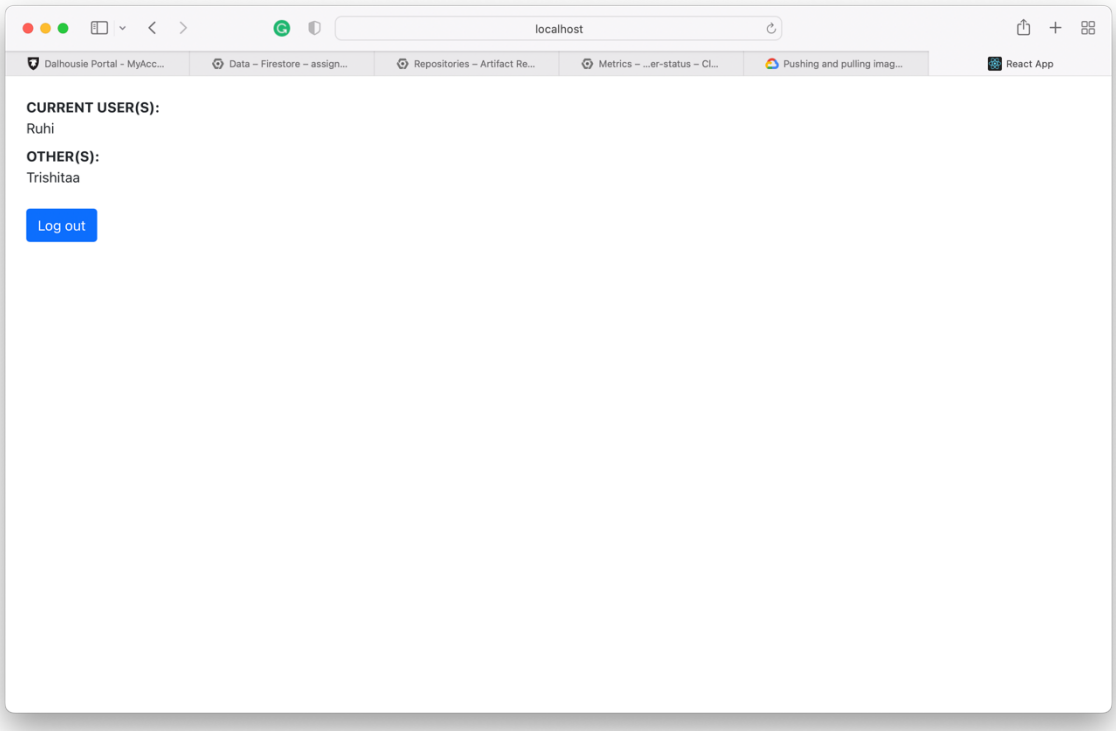

Figure 14. *Screenshot of users that are ONLINE.*

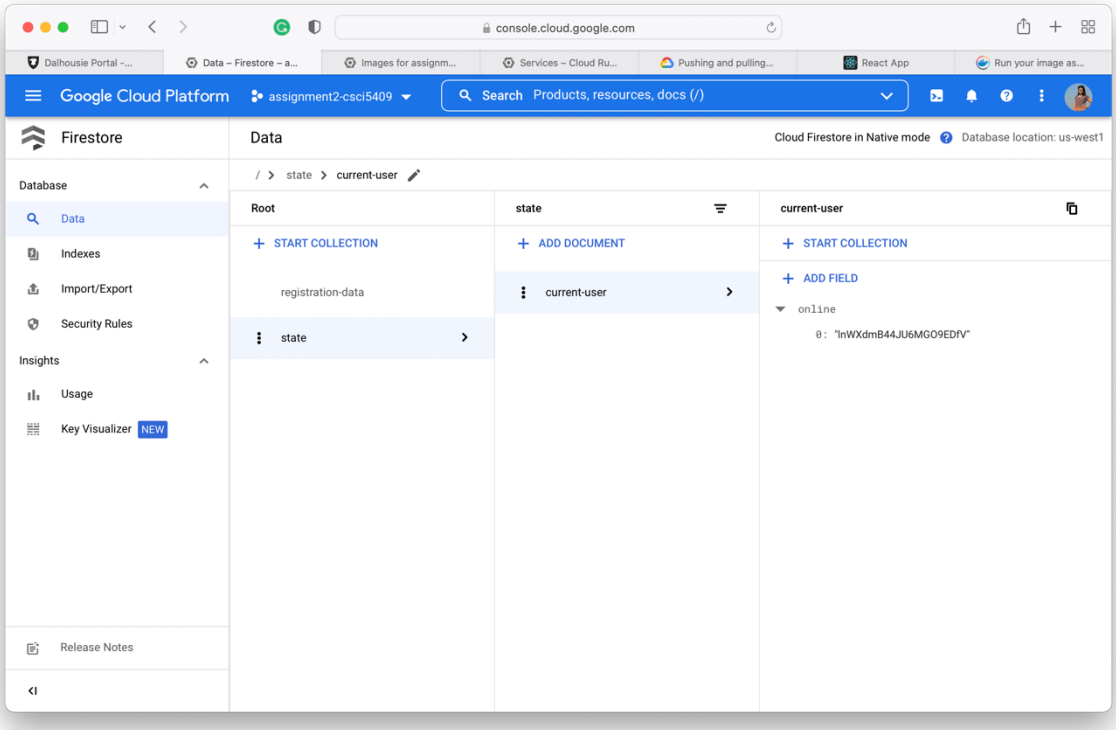

Figure 15. *Screenshot of Firestore showing the ONLY online user after logout.*

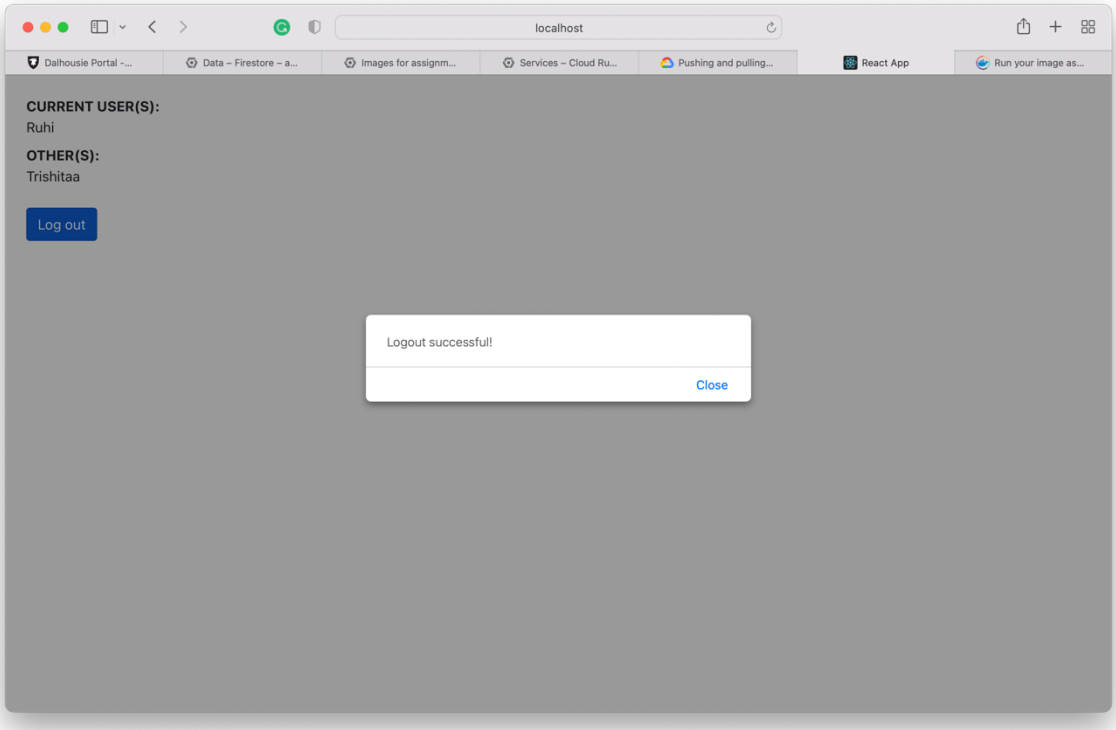

Figure 16. *Screenshot of SUCCESSFUL logout of the current user.*

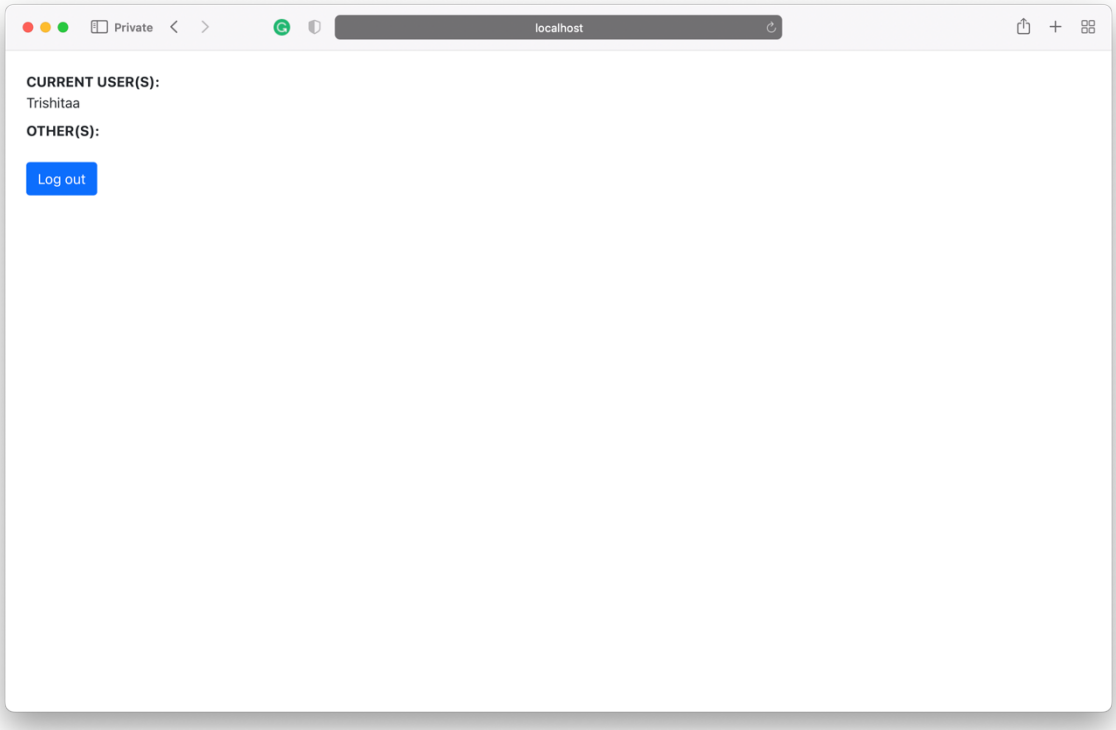

Figure 17. *Screenshot of SUCCESSFUL logout of the current user and showing the ONLY user online.*

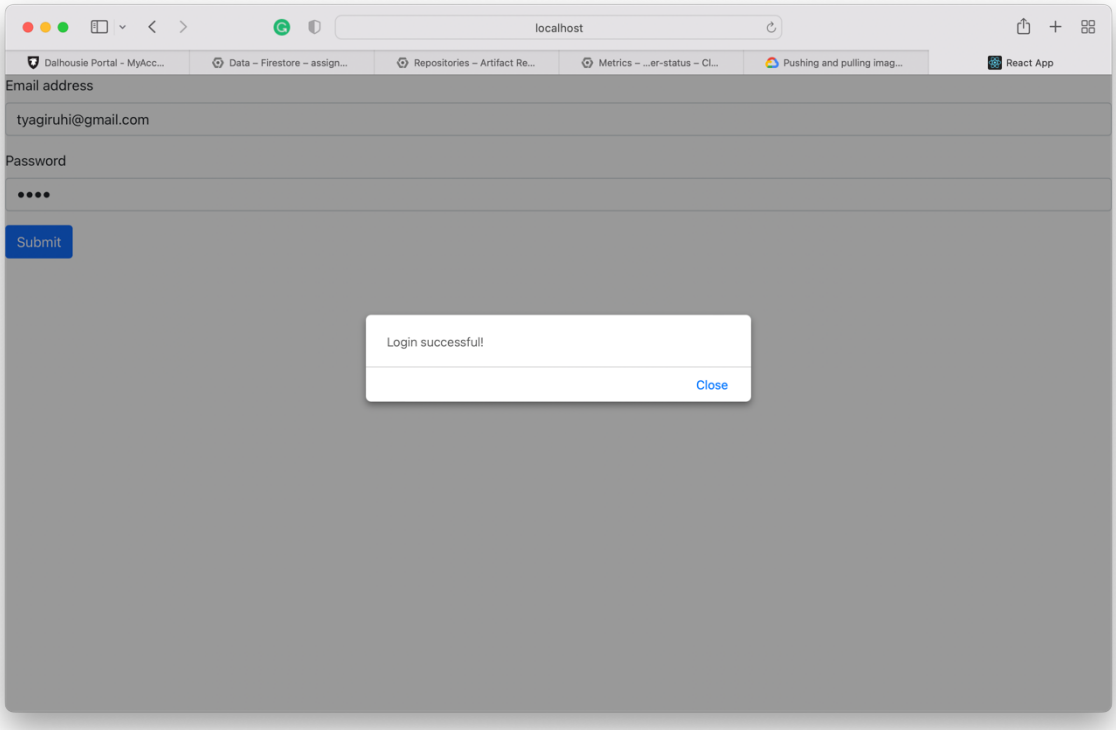

Figure 18. *Screenshot of login to show that this user goes to OTHER(S) category on user status.*

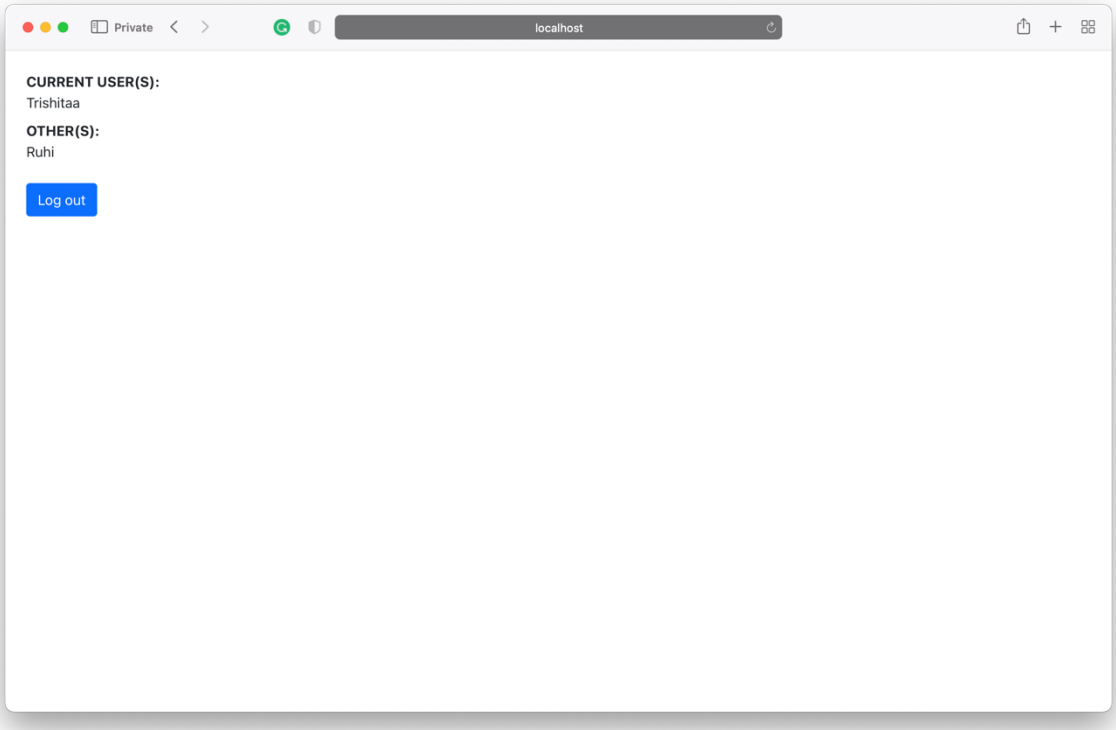

Figure 19. *Screenshot of users that are ONLINE as in order they logged in after recent login shown in figure 18.*

Step 5: Lastly, images of docker on Google Cloud Platform.

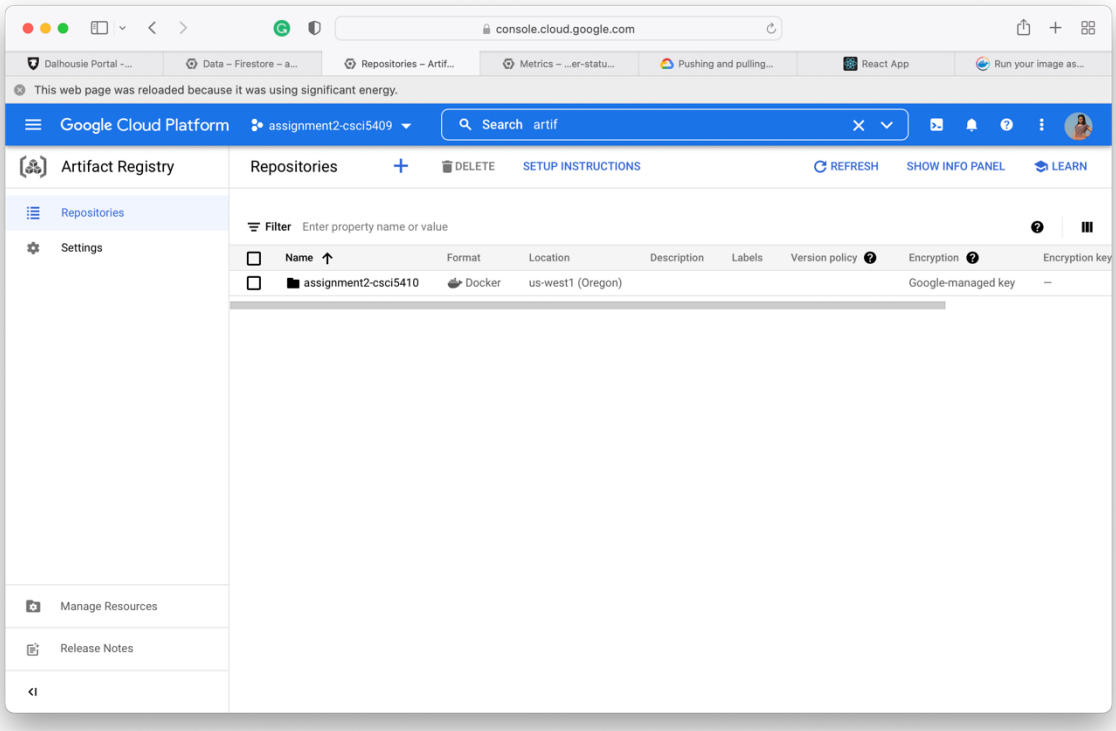

Figure 20. *Screenshot of repository in Artifact Registry on GCP.*

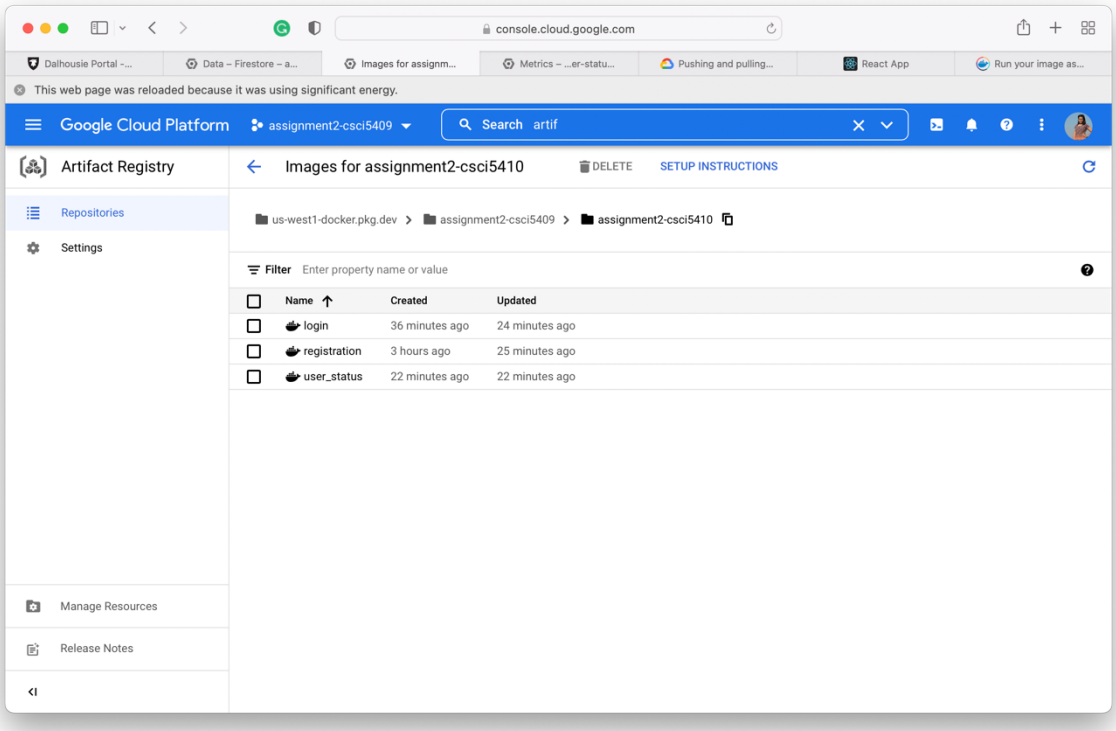

Figure 19. *Screenshot of images inside the repository.*

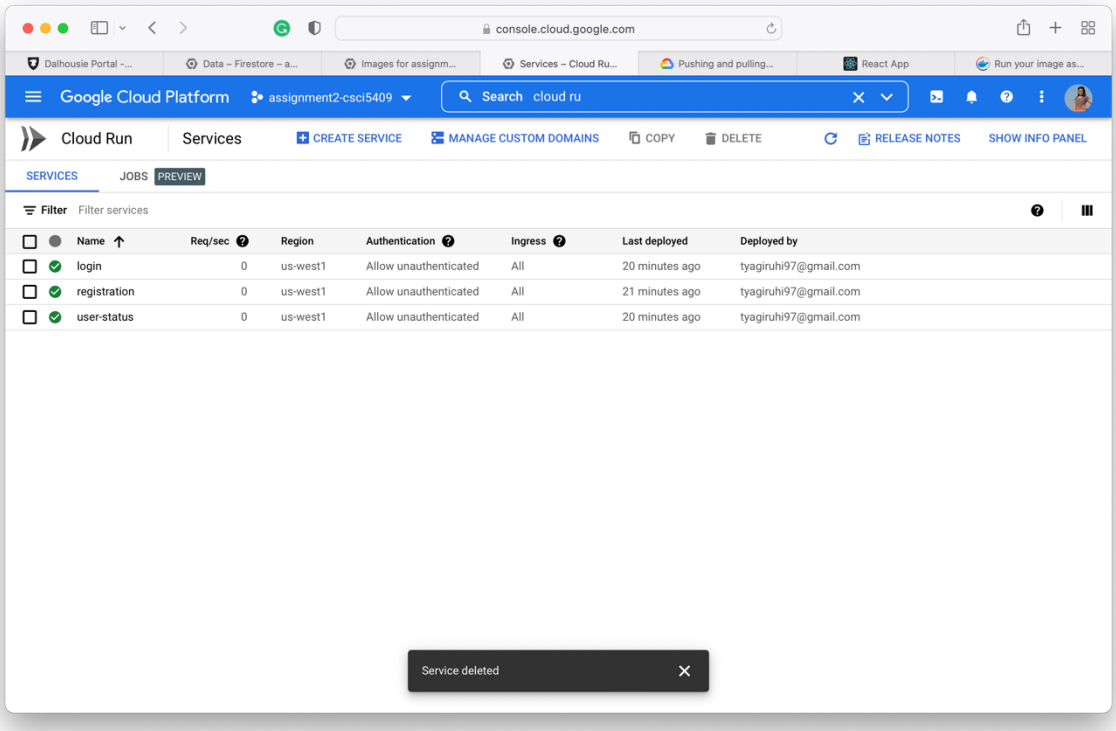

Figure 19. *Screenshot of services on Cloud Run.*

Step 6: *Stepwise detail of how I performed the assignment.*

1. **WebStorm:**
   WebStorm is a JavaScript and associated technologies integrated development environment [1]. It improves your development experience by automating routine operations and assisting you with complex tasks, just like other JetBrains IDEs [1].
   I used this IDE to create three applications: register_module, login_modul and user_status_module using Node.js. for backend. And for frontend, I created respective frontend modules using React technology.

2. **Docker:**
   Docker has been used to produce images after reading the docker documentation [2]. Docker images are created in the tasks above using the docker build command with the platform type linux/amd64, as seen below.

   docker build -t <imagename> . --platform linux/amd64

   The images are then containerized, as shown below, using the docker run command.

   docker run -dp <imagename>

   Each container runs independently of the others, and a single image can be used to produce numerous containers. Images and containers must be pushed to Google Cloud after they have been created.

3. **Artifact Registry:**
   Google Cloud provides a service called Artifact Registry[3] for managing docker images and containers in the cloud, where docker images may be pulled or pushed. The gcloud client is initially installed in order to access all of the gcloud services and to authenticate with gcloud. Before pushing the docker images, an oauth2accesstoken is generated for the project in Google Cloud for the above actions. The token will expire in 60 minutes after it is generated. The docker images are initially labelled using the token created.

   docker tag <name> <region>-docker.pkg.dev/<project-id> /<repo-name>/<name>

   After tagging the image, the image is pushed to the cloud using the command:

   docker push <region>.pkg.dev/<project-id>/<repo-name>/<name>

4. **Cloud Run:**

Three services are established and deployed in the cloud run for the three images pushed in the artefact registry once docker images are created in artefact registries. Each service provides an HTTP/HTTPS endpoint that may be used to call any of the microservices that have been deployed. I familiarized myself with working of Cloud Run using this [4] documentation.

5. **Backend Technologies:**
   For all of the backend microservices, NodeJS is used to design APIs, ExpressJS is used to create API routes, and google-cloud/firestore is used to interface with the Firestore database.

6. **Frontend Technologies:**
   Only one website is being developed for the frontend, with routing between all of the pages such as Registration, Login, and User Status. ReactJS is being utilised for the core application, and ReactRouter is being used for frontend routing.

REFERENCES:

[1] "WebStorm: The Smartest JavaScript IDE, by JetBrains", *JetBrains*, 2022. [Online]. Available: https://www.jetbrains.com/webstorm/. [Accessed: 11- Jun- 2022]

[2] "Docker Documentation", Docker Documentation, 2022. [Online]. Available: https://docs.docker.com/. [Accessed: 11- Jun- 2022]

[3] "Working with container images | Artifact Registry documentation | Google Cloud", Google Cloud, 2022. [Online]. Available: https://cloud.google.com/artifactregistry/
docs/docker. [Accessed: 10- Jun- 2022]

[4] "Quickstart: Deploy a container to Cloud Run | Cloud Run Documentation | Google Cloud", Google Cloud, 2022. [Online]. Available: https://cloud.google.com/run/docs/quickstarts/deploy-container. [Accessed: 10- Jun-2022]