CSCI 5410

Assignment 2 – Part A

Name: Benny Daniel Tharigopala

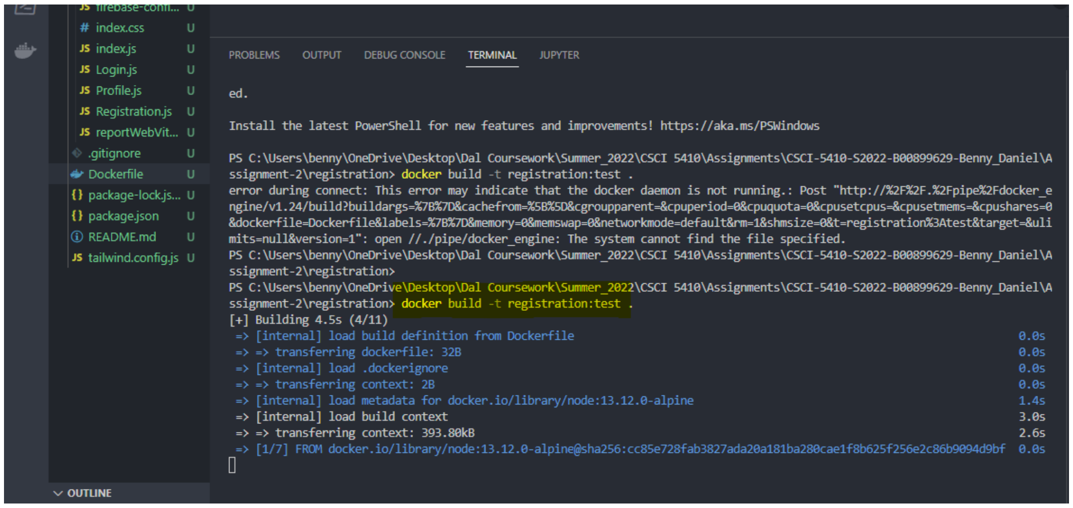
Banner ID: B00899629

GitLab URL: https://git.cs.dal.ca/benny/csci5410_B00899629_Benny_Tharigopala

Cloud Run URLs:

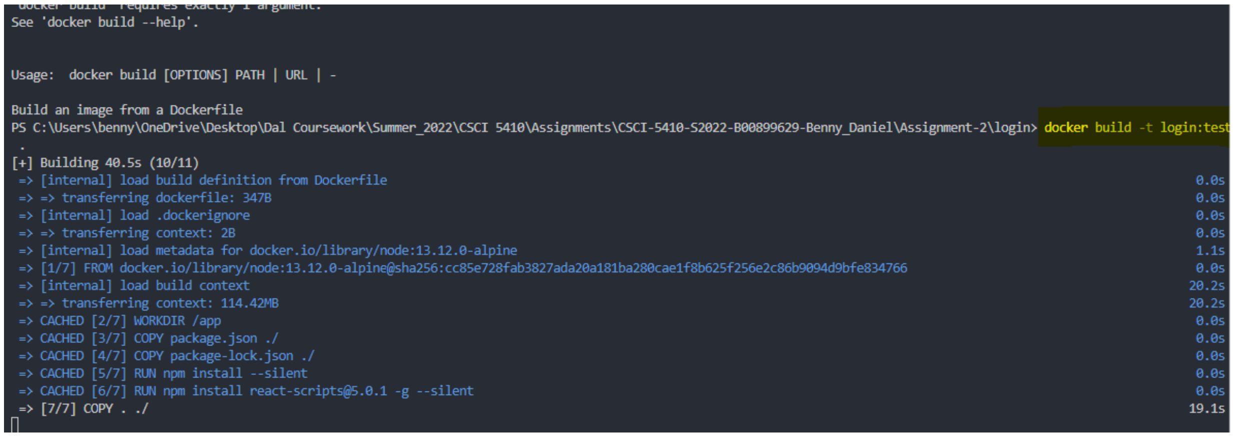
1. **https://registration-qsqervj6va-ue.a.run.app**
2. **https://login-qsqervj6va-ue.a.run.app**
3. **https://profile-qsqervj6va-ue.a.run.app/Benny Daniel**

Development, Execution & Deployment of a Containerized Application



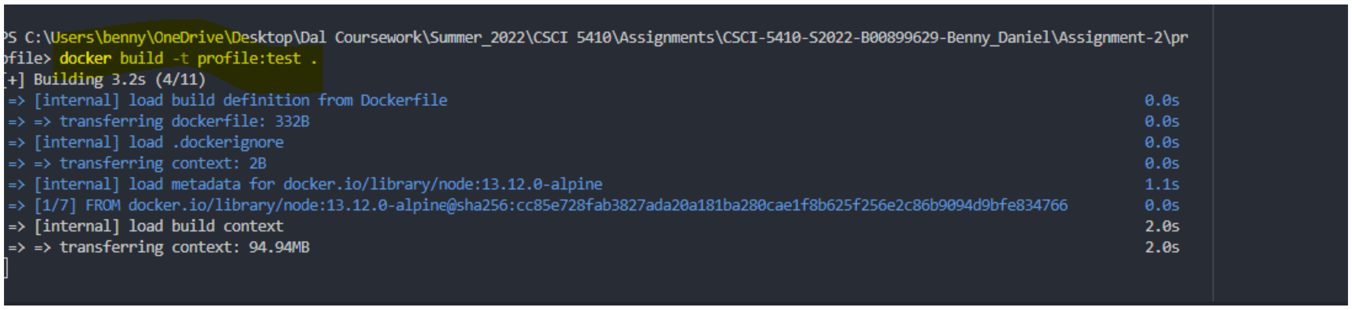
```
ed.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\benny\OneDrive\Desktop\Dal Coursework\Summer_2022\CSCI 5410\Assignments\CSCI-5410-S2022-B00899629-Benny_Daniel\A
ssignment-2\registration> docker build -t registration:test .
error during connect: This error may indicate that the docker daemon is not running.: Post "http://%2F%2F.%2Fpipe%2Fdocker_e
ngine/v1.24/build?buildargs=%7B%7D&cachefrom=%5B%5D&cgroupparent=&cpuperiod=0&cpuquota=0&cpusetcpus=&cpusetmems=&cpushares=0
&dockerfile=Dockerfile&labels=%7B%7D&memory=0&memswap=0&networkmode=default&rm=1&shmsize=0&t=registration%3Atest&target=&uli
mits=null&version=1": open //./pipe/docker_engine: The system cannot find the file specified.
PS C:\Users\benny\OneDrive\Desktop\Dal Coursework\Summer_2022\CSCI 5410\Assignments\CSCI-5410-S2022-B00899629-Benny_Daniel\A
ssignment-2\registration>
PS C:\Users\benny\OneDrive\Desktop\Dal Coursework\Summer_2022\CSCI 5410\Assignments\CSCI-5410-S2022-B00899629-Benny_Daniel\A
ssignment-2\registration> docker build -t registration:test .
[+] Building 4.5s (4/11)
 => [internal] load build definition from Dockerfile                                                            0.0s
 => => transferring dockerfile: 32B                                                                             0.0s
 => [internal] load .dockerignore                                                                               0.0s
 => => transferring context: 2B                                                                                 0.0s
 => [internal] load metadata for docker.io/library/node:13.12.0-alpine                                          1.4s
 => [internal] load build context                                                                               3.0s
 => => transferring context: 393.80kB                                                                           2.6s
 => [1/7] FROM docker.io/library/node:13.12.0-alpine@sha256:cc85e728fab3827ada20a181ba280cae1f8b625f256e2c86b9094d9bf  0.0s
```

Figure 1: Build the Image for the Registration Page



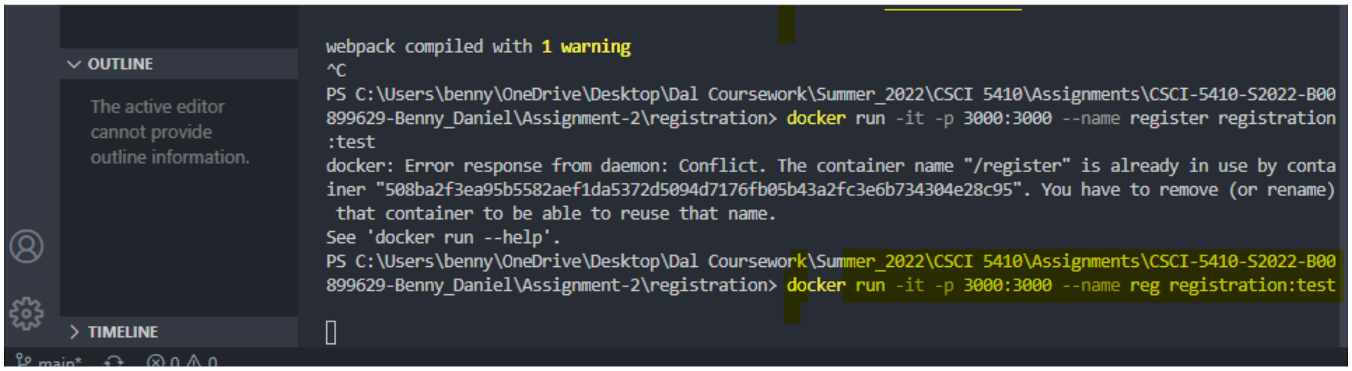
```
"docker build" requires exactly 1 argument.
See 'docker build --help'.


Usage:  docker build [OPTIONS] PATH | URL | -

Build an image from a Dockerfile
PS C:\Users\benny\OneDrive\Desktop\Dal Coursework\Summer_2022\CSCI 5410\Assignments\CSCI-5410-S2022-B00899629-Benny_Daniel\Assignment-2\login> docker build -t login:test
 .
[+] Building 40.5s (10/11)
 => [internal] load build definition from Dockerfile                                                                   0.0s
 => => transferring dockerfile: 347B                                                                                   0.0s
 => [internal] load .dockerignore                                                                                      0.0s
 => => transferring context: 2B                                                                                        0.0s
 => [internal] load metadata for docker.io/library/node:13.12.0-alpine                                                 1.1s
 => [1/7] FROM docker.io/library/node:13.12.0-alpine@sha256:cc85e728fab3827ada20a181ba280cae1f8b625f256e2c86b9094d9bfe834766   0.0s
 => [internal] load build context                                                                                     20.2s
 => => transferring context: 114.42MB                                                                                 20.2s
 => CACHED [2/7] WORKDIR /app                                                                                          0.0s
 => CACHED [3/7] COPY package.json ./                                                                                  0.0s
 => CACHED [4/7] COPY package-lock.json ./                                                                             0.0s
 => CACHED [5/7] RUN npm install --silent                                                                              0.0s
 => CACHED [6/7] RUN npm install react-scripts@5.0.1 -g --silent                                                       0.0s
 => [7/7] COPY . ./                                                                                                   19.1s
```

Figure 2: Build the Image for the Login Page



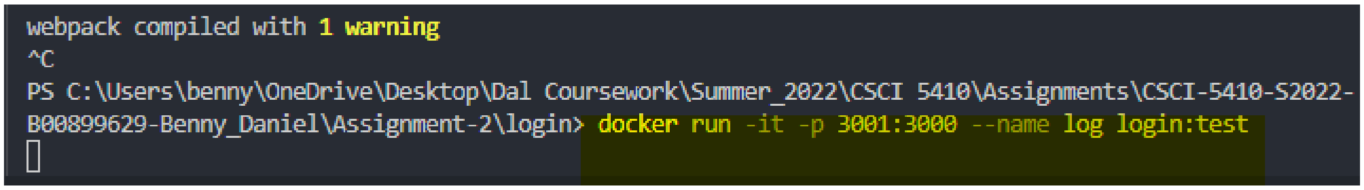
```
S C:\Users\benny\OneDrive\Desktop\Dal Coursework\Summer_2022\CSCI 5410\Assignments\CSCI-5410-S2022-B00899629-Benny_Daniel\Assignment-2\pr
file> docker build -t profile:test .
+] Building 3.2s (4/11)
 => [internal] load build definition from Dockerfile                                                         0.0s
 => => transferring dockerfile: 332B                                                                         0.0s
 => [internal] load .dockerignore                                                                            0.0s
 => => transferring context: 2B                                                                              0.0s
 => [internal] load metadata for docker.io/library/node:13.12.0-alpine                                       1.1s
 => [1/7] FROM docker.io/library/node:13.12.0-alpine@sha256:cc85e728fab3827ada20a181ba280cae1f8b625f256e2c86b9094d9bfe834766   0.0s
 => [internal] load build context                                                                            2.0s
 => => transferring context: 94.94MB                                                                         2.0s
```

Figure 3: Build the Image for the Profile Page



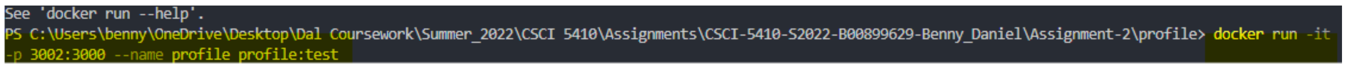
```
webpack compiled with 1 warning
^C
PS C:\Users\benny\OneDrive\Desktop\Dal Coursework\Summer_2022\CSCI 5410\Assignments\CSCI-5410-S2022-B00
899629-Benny_Daniel\Assignment-2\registration> docker run -it -p 3000:3000 --name register registration
:test
docker: Error response from daemon: Conflict. The container name "/register" is already in use by conta
iner "508ba2f3ea95b5582aef1da5372d5094d7176fb05b43a2fc3e6b734304e28c95". You have to remove (or rename)
 that container to be able to reuse that name.
See 'docker run --help'.
PS C:\Users\benny\OneDrive\Desktop\Dal Coursework\Summer_2022\CSCI 5410\Assignments\CSCI-5410-S2022-B00
899629-Benny_Daniel\Assignment-2\registration> docker run -it -p 3000:3000 --name reg registration:test
```

Figure 4: Execute a container for the "registation" Image



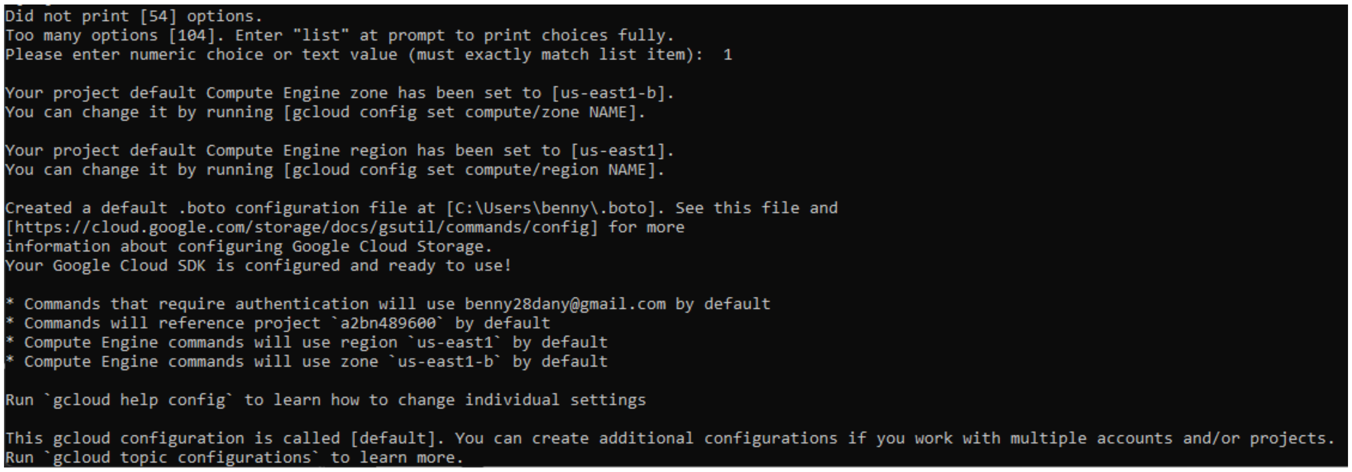
```
webpack compiled with 1 warning
^C
PS C:\Users\benny\OneDrive\Desktop\Dal Coursework\Summer_2022\CSCI 5410\Assignments\CSCI-5410-S2022-
B00899629-Benny_Daniel\Assignment-2\login> docker run -it -p 3001:3000 --name log login:test
```

Figure 5: Execute a container for the "login" Image



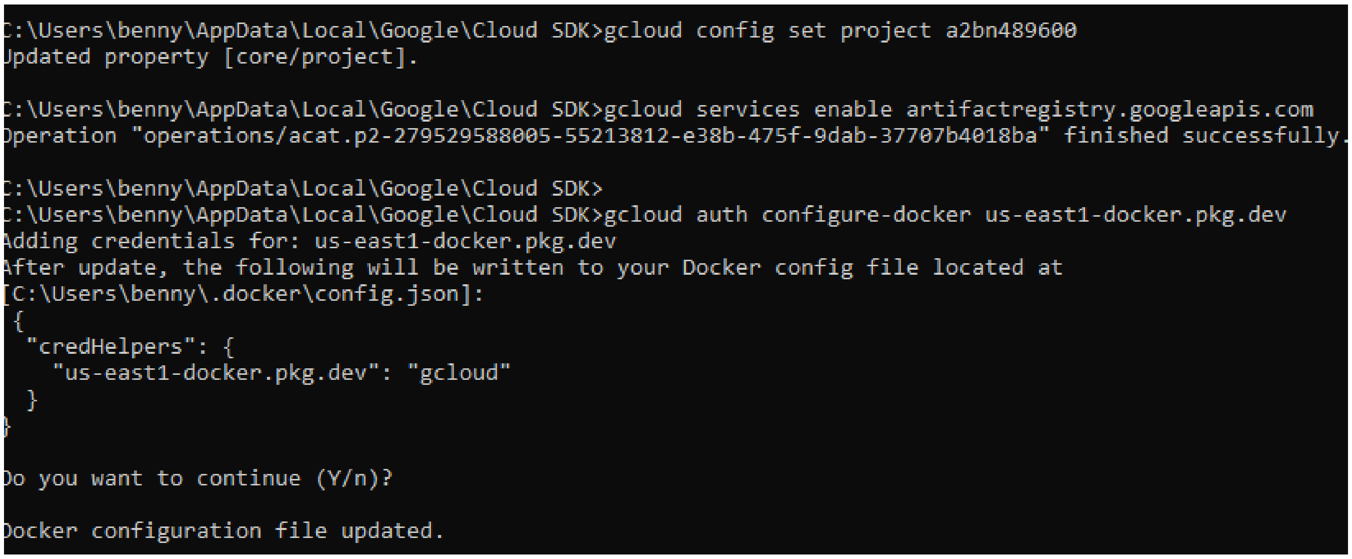
```
See 'docker run --help'.
PS C:\Users\benny\OneDrive\Desktop\Dal Coursework\Summer_2022\CSCI 5410\Assignments\CSCI-5410-S2022-B00899629-Benny_Daniel\Assignment-2\profile> docker run -it
-p 3002:3000 --name profile profile:test
```

Figure 6: Execute a container for the "profile" Image



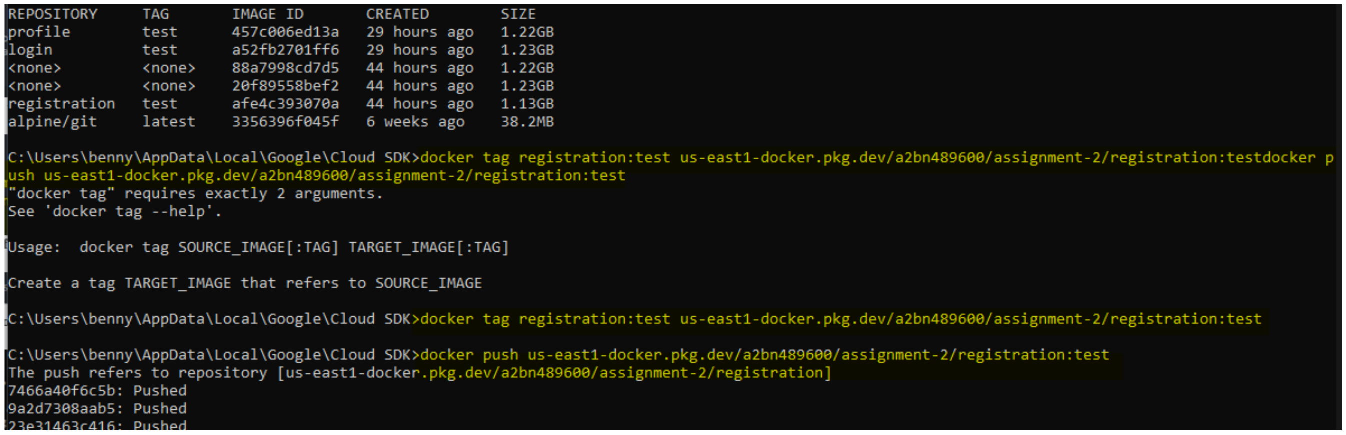
```
Did not print [54] options.
Too many options [104]. Enter "list" at prompt to print choices fully.
Please enter numeric choice or text value (must exactly match list item):  1

Your project default Compute Engine zone has been set to [us-east1-b].
You can change it by running [gcloud config set compute/zone NAME].

Your project default Compute Engine region has been set to [us-east1].
You can change it by running [gcloud config set compute/region NAME].

Created a default .boto configuration file at [C:\Users\benny\.boto]. See this file and
[https://cloud.google.com/storage/docs/gsutil/commands/config] for more
information about configuring Google Cloud Storage.
Your Google Cloud SDK is configured and ready to use!

* Commands that require authentication will use benny28dany@gmail.com by default
* Commands will reference project `a2bn489600` by default
* Compute Engine commands will use region `us-east1` by default
* Compute Engine commands will use zone `us-east1-b` by default

Run `gcloud help config` to learn how to change individual settings

This gcloud configuration is called [default]. You can create additional configurations if you work with multiple accounts and/or projects.
Run `gcloud topic configurations` to learn more.
```

Figure 7: Select a Region [1]



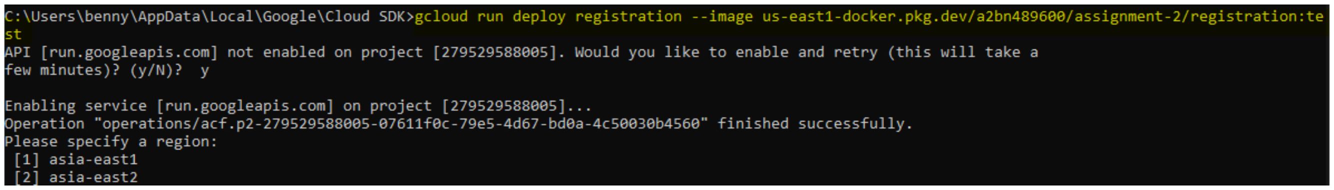
```
C:\Users\benny\AppData\Local\Google\Cloud SDK>gcloud config set project a2bn489600
pdated property [core/project].

C:\Users\benny\AppData\Local\Google\Cloud SDK>gcloud services enable artifactregistry.googleapis.com
peration "operations/acat.p2-279529588005-55213812-e38b-475f-9dab-37707b4018ba" finished successfully.

C:\Users\benny\AppData\Local\Google\Cloud SDK>
C:\Users\benny\AppData\Local\Google\Cloud SDK>gcloud auth configure-docker us-east1-docker.pkg.dev
dding credentials for: us-east1-docker.pkg.dev
fter update, the following will be written to your Docker config file located at
[C:\Users\benny\.docker\config.json]:
 {
  "credHelpers": {
    "us-east1-docker.pkg.dev": "gcloud"
  }
}

o you want to continue (Y/n)?

ocker configuration file updated.
```

Figure 8: Select a Project



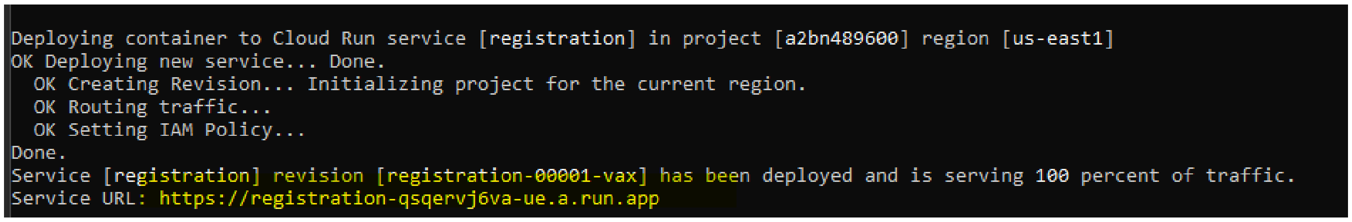
```
REPOSITORY     TAG       IMAGE ID       CREATED         SIZE
profile        test      457c006ed13a   29 hours ago    1.22GB
login          test      a52fb2701ff6   29 hours ago    1.23GB
<none>         <none>    88a7998cd7d5   44 hours ago    1.22GB
<none>         <none>    20f89558bef2   44 hours ago    1.23GB
registration   test      afe4c393070a   44 hours ago    1.13GB
alpine/git     latest    3356396f045f   6 weeks ago     38.2MB

C:\Users\benny\AppData\Local\Google\Cloud SDK>docker tag registration:test us-east1-docker.pkg.dev/a2bn489600/assignment-2/registration:testdocker p
ush us-east1-docker.pkg.dev/a2bn489600/assignment-2/registration:test
"docker tag" requires exactly 2 arguments.
See 'docker tag --help'.

Usage:  docker tag SOURCE_IMAGE[:TAG] TARGET_IMAGE[:TAG]

Create a tag TARGET_IMAGE that refers to SOURCE_IMAGE

C:\Users\benny\AppData\Local\Google\Cloud SDK>docker tag registration:test us-east1-docker.pkg.dev/a2bn489600/assignment-2/registration:test

C:\Users\benny\AppData\Local\Google\Cloud SDK>docker push us-east1-docker.pkg.dev/a2bn489600/assignment-2/registration:test
The push refers to repository [us-east1-docker.pkg.dev/a2bn489600/assignment-2/registration]
7466a40f6c5b: Pushed
9a2d7308aab5: Pushed
23e31463c416: Pushed
```

Figure 9: Create a tag for the registration image, push it to the artifact registry [2-3]



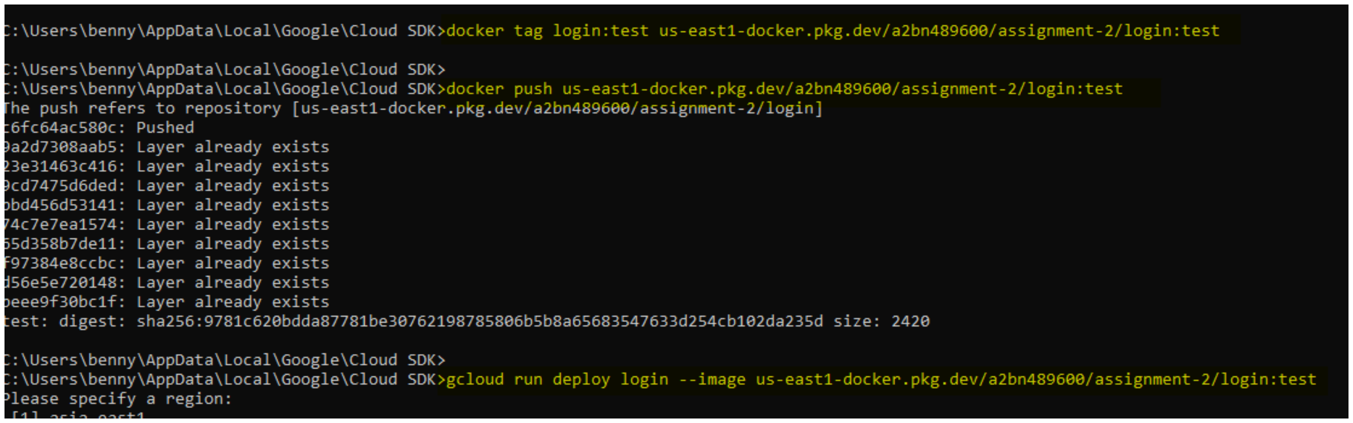
```
C:\Users\benny\AppData\Local\Google\Cloud SDK>gcloud run deploy registration --image us-east1-docker.pkg.dev/a2bn489600/assignment-2/registration:te
st
API [run.googleapis.com] not enabled on project [279529588005]. Would you like to enable and retry (this will take a
few minutes)? (y/N)?  y

Enabling service [run.googleapis.com] on project [279529588005]...
Operation "operations/acf.p2-279529588005-07611f0c-79e5-4d67-bd0a-4c50030b4560" finished successfully.
Please specify a region:
 [1] asia-east1
 [2] asia-east2
```

Figure 10: Execute the image on Cloud Run



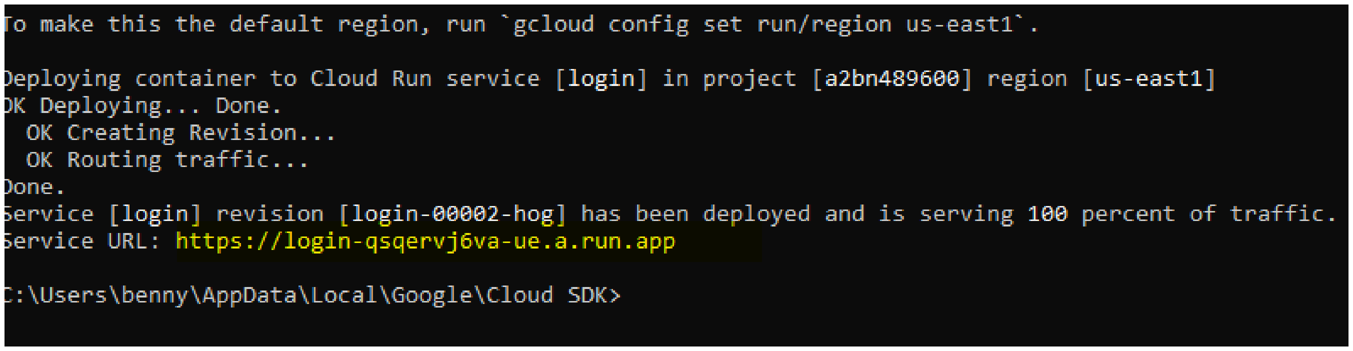
```
Deploying container to Cloud Run service [registration] in project [a2bn489600] region [us-east1]
OK Deploying new service... Done.
  OK Creating Revision... Initializing project for the current region.
  OK Routing traffic...
  OK Setting IAM Policy...
Done.
Service [registration] revision [registration-00001-vax] has been deployed and is serving 100 percent of traffic.
Service URL: https://registration-qsqervj6va-ue.a.run.app
```

Figure 11: URL for the Registration page on Cloudrun



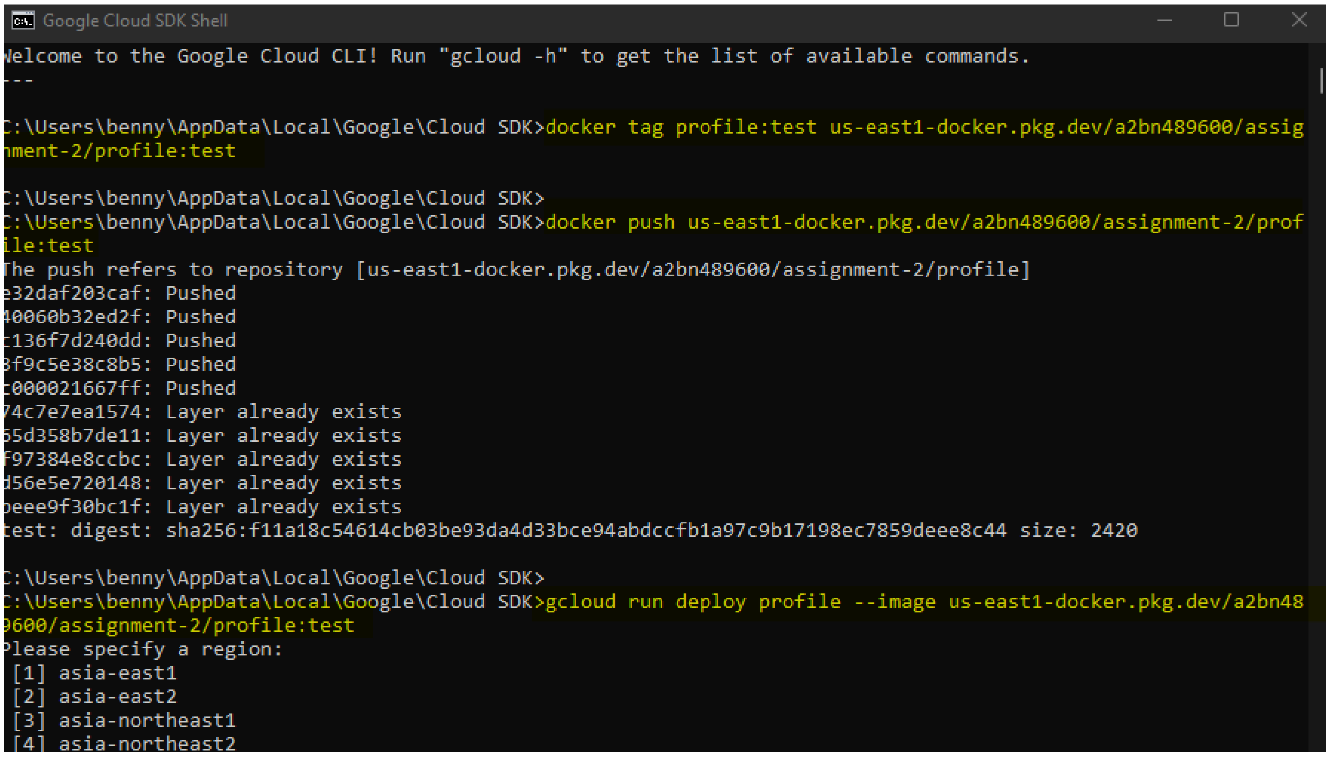
```
:\Users\benny\AppData\Local\Google\Cloud SDK>docker tag login:test us-east1-docker.pkg.dev/a2bn489600/assignment-2/login:test

:\Users\benny\AppData\Local\Google\Cloud SDK>
:\Users\benny\AppData\Local\Google\Cloud SDK>docker push us-east1-docker.pkg.dev/a2bn489600/assignment-2/login:test
The push refers to repository [us-east1-docker.pkg.dev/a2bn489600/assignment-2/login]
:6fc64ac580c: Pushed
9a2d7308aab5: Layer already exists
23e31463c416: Layer already exists
9cd7475d6ded: Layer already exists
bbd456d53141: Layer already exists
74c7e7ea1574: Layer already exists
55d358b7de11: Layer already exists
f97384e8ccbc: Layer already exists
d56e5e720148: Layer already exists
beee9f30bc1f: Layer already exists
test: digest: sha256:9781c620bdda87781be30762198785806b5b8a65683547633d254cb102da235d size: 2420

:\Users\benny\AppData\Local\Google\Cloud SDK>
:\Users\benny\AppData\Local\Google\Cloud SDK>gcloud run deploy login --image us-east1-docker.pkg.dev/a2bn489600/assignment-2/login:test
Please specify a region:
```

Figure 12: Create a tag for the login image, push it to the artifact registry



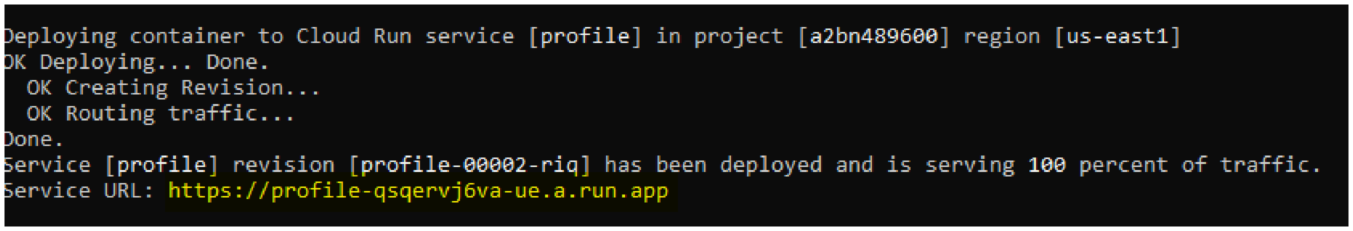
```
To make this the default region, run `gcloud config set run/region us-east1`.

Deploying container to Cloud Run service [login] in project [a2bn489600] region [us-east1]
OK Deploying... Done.
  OK Creating Revision...
  OK Routing traffic...
Done.
Service [login] revision [login-00002-hog] has been deployed and is serving 100 percent of traffic.
Service URL: https://login-qsqervj6va-ue.a.run.app

C:\Users\benny\AppData\Local\Google\Cloud SDK>
```

Figure 13: Obtain the URL for the Login page



```
Welcome to the Google Cloud CLI! Run "gcloud -h" to get the list of available commands.
---

C:\Users\benny\AppData\Local\Google\Cloud SDK>docker tag profile:test us-east1-docker.pkg.dev/a2bn489600/assig
nment-2/profile:test

C:\Users\benny\AppData\Local\Google\Cloud SDK>
C:\Users\benny\AppData\Local\Google\Cloud SDK>docker push us-east1-docker.pkg.dev/a2bn489600/assignment-2/prof
ile:test
The push refers to repository [us-east1-docker.pkg.dev/a2bn489600/assignment-2/profile]
e32daf203caf: Pushed
40060b32ed2f: Pushed
c136f7d240dd: Pushed
3f9c5e38c8b5: Pushed
c000021667ff: Pushed
74c7e7ea1574: Layer already exists
55d358b7de11: Layer already exists
f97384e8ccbc: Layer already exists
d56e5e720148: Layer already exists
beee9f30bc1f: Layer already exists
test: digest: sha256:f11a18c54614cb03be93da4d33bce94abdccfb1a97c9b17198ec7859deee8c44 size: 2420

C:\Users\benny\AppData\Local\Google\Cloud SDK>
C:\Users\benny\AppData\Local\Google\Cloud SDK>gcloud run deploy profile --image us-east1-docker.pkg.dev/a2bn48
9600/assignment-2/profile:test
Please specify a region:
 [1] asia-east1
 [2] asia-east2
 [3] asia-northeast1
 [4] asia-northeast2
```

Figure 14: Create a tag for the Profile image, push it to the artifact registry



```
Deploying container to Cloud Run service [profile] in project [a2bn489600] region [us-east1]
OK Deploying... Done.
  OK Creating Revision...
  OK Routing traffic...
Done.
Service [profile] revision [profile-00002-riq] has been deployed and is serving 100 percent of traffic.
Service URL: https://profile-qsqervj6va-ue.a.run.app
```

Figure 15: Obtain the URL for the Profile page

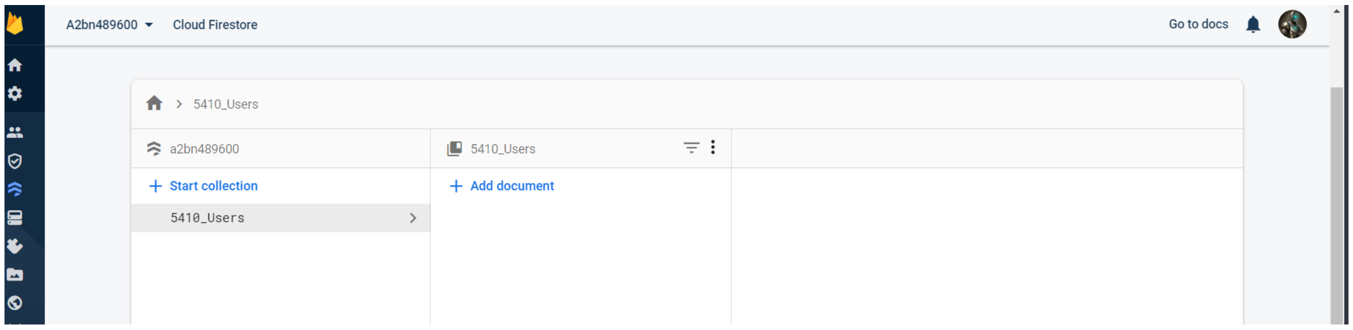


Figure 16: Empty User Details Table

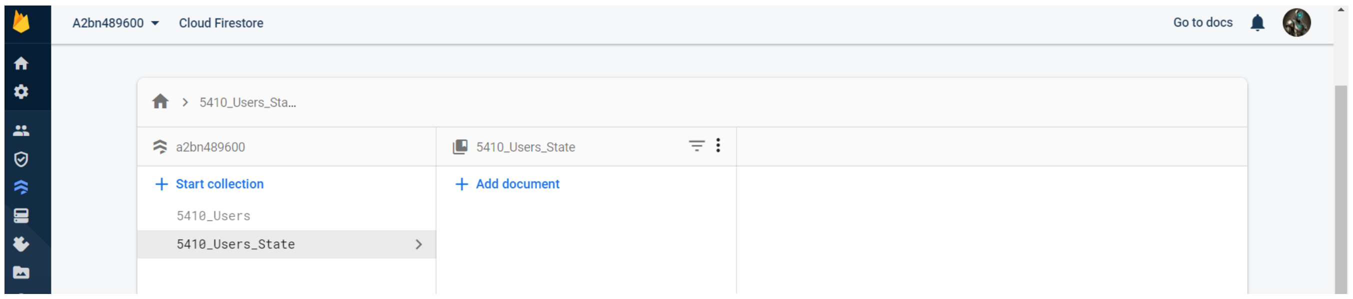
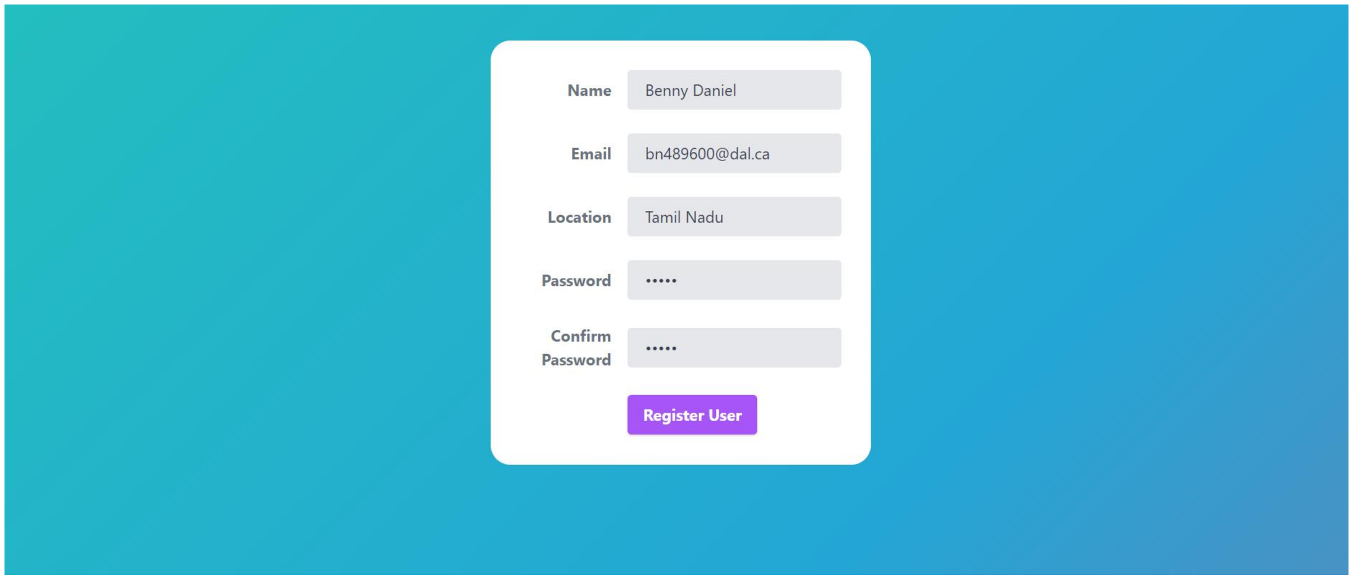
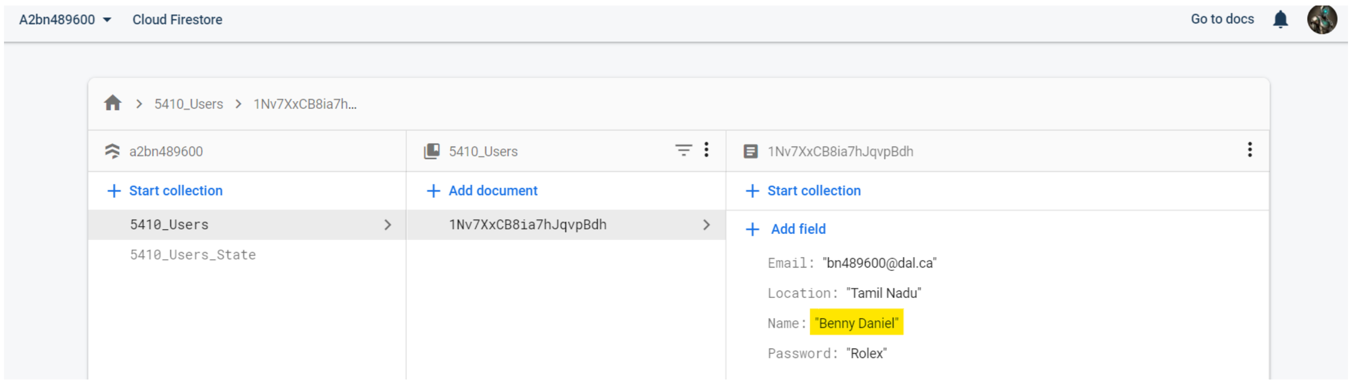


Figure 17: Empty User State Table



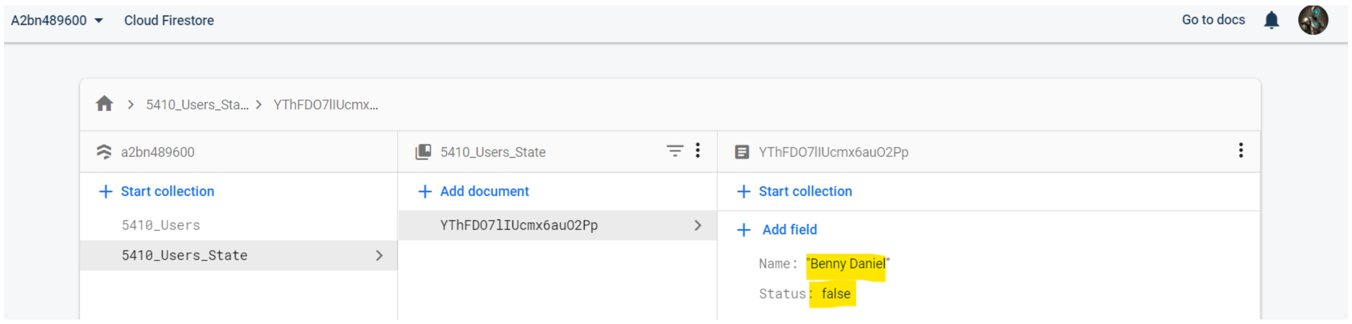
Name	Benny Daniel
Email	bn489600@dal.ca
Location	Tamil Nadu
Password	•••••
Confirm Password	•••••

Figure 18: Registering a User - Benny Daniel"



Email: "bn489600@dal.ca"
Location: "Tamil Nadu"
Name: "Benny Daniel"
Password: "Rolex"

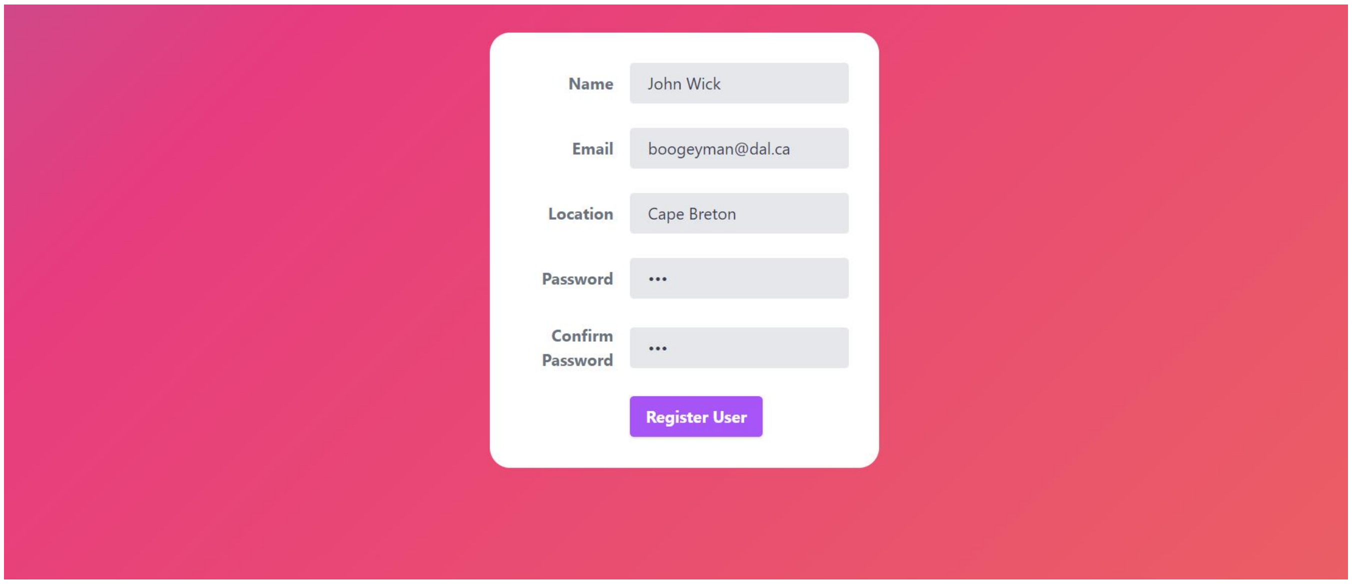
Figure 19: New Document added to the Collection upon registration



Name: "Benny Daniel"
Status: false

Figure 20: New Document added to the "State" Collection upon registration

Similarly, register 2 other Users:



Name	John Wick
Email	boogeyman@dal.ca
Location	Cape Breton
Password	•••
Confirm Password	•••

Figure 21: Registering a User - "John Wick"

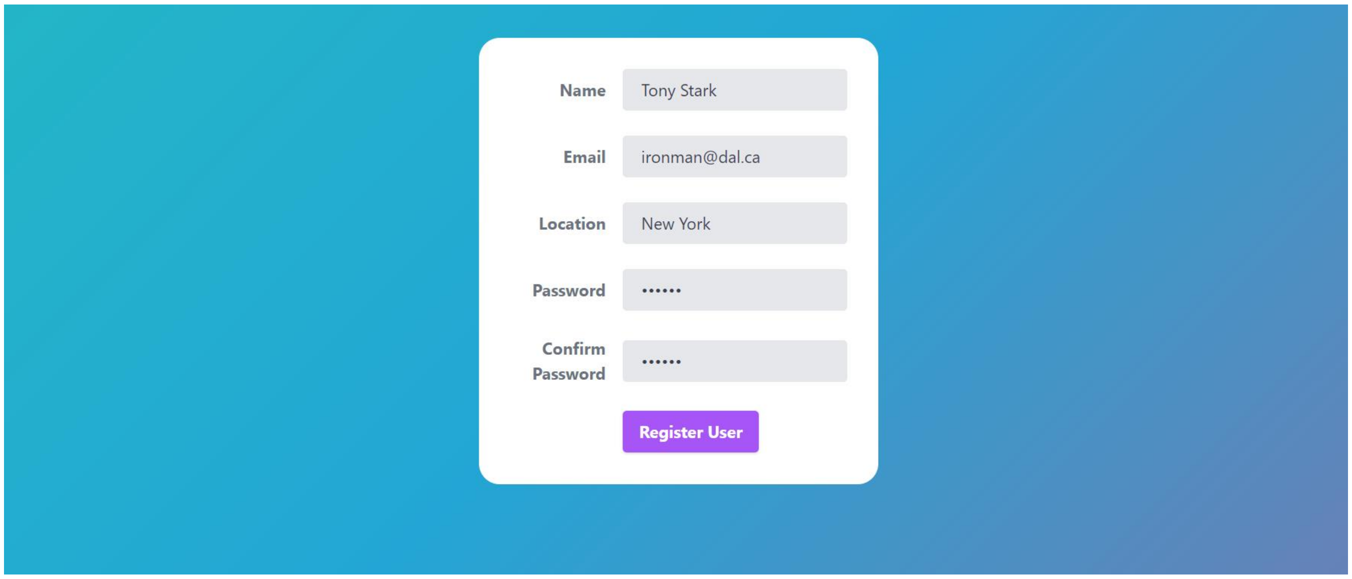
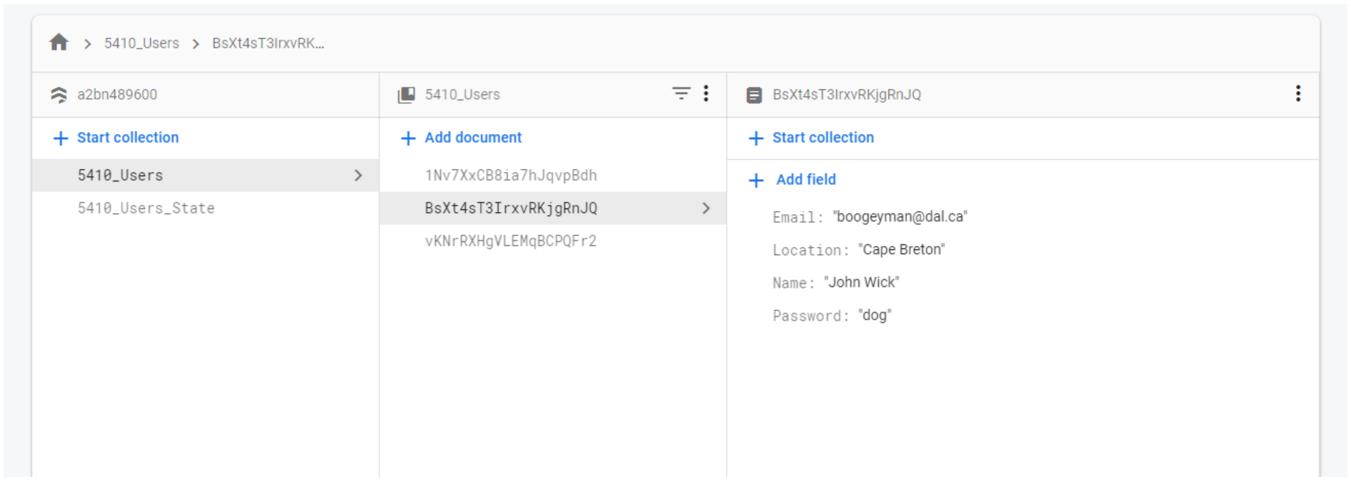
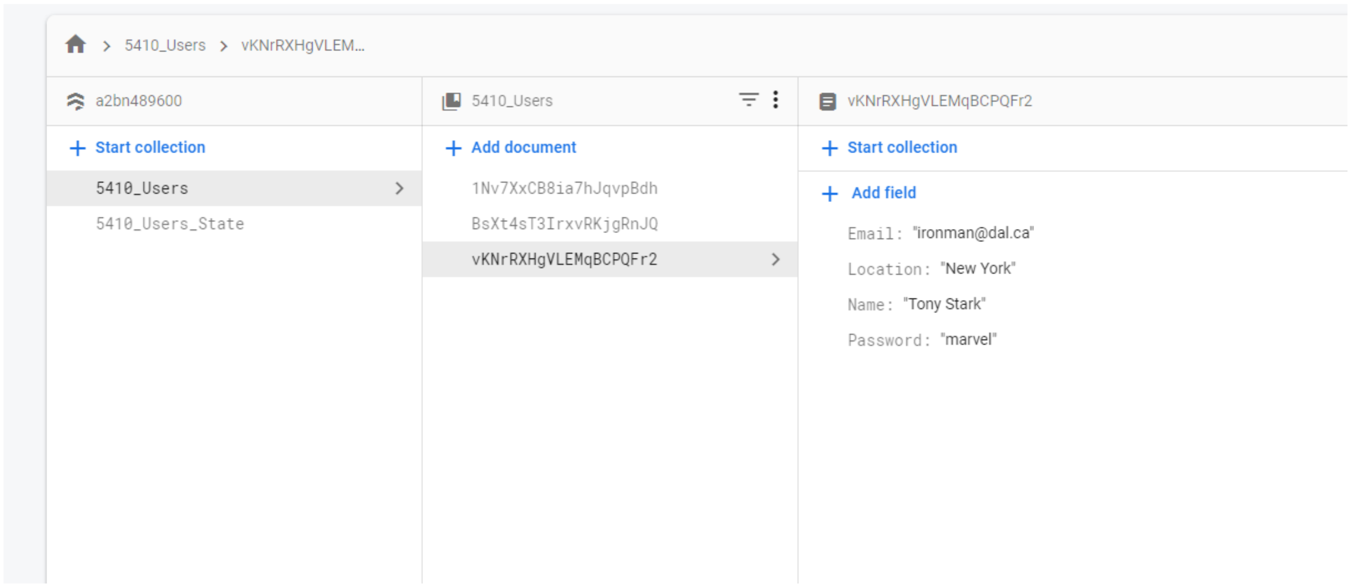

Name	Tony Stark
Email	ironman@dal.ca
Location	New York
Password	••••••
Confirm Password	••••••

Figure 22: Registering a User - "Tony Stark"



```
Email: "boogeyman@dal.ca"
Location: "Cape Breton"
Name: "John Wick"
Password: "dog"
```

Figure 23: Firebase Document after registering John Wick

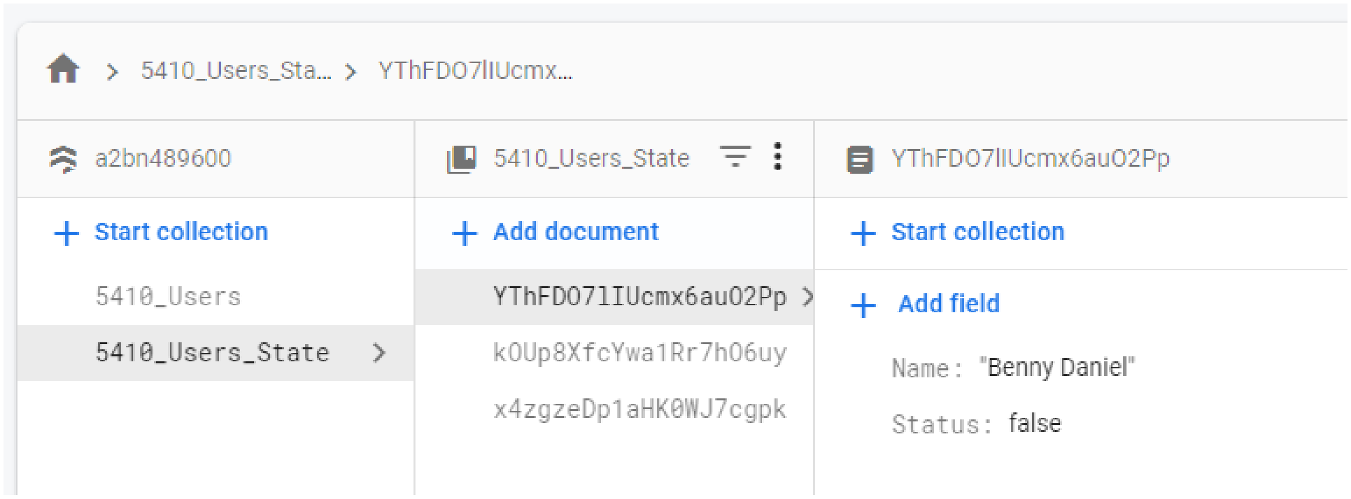

5410_Users > vKNrRXHgVLEM...

a2bn489600
+ Start collection
5410_Users
5410_Users_State

5410_Users
+ Add document
1Nv7XxCB8ia7hJqvpBdh
BsXt4sT3IrxvRKjgRnJQ
vKNrRXHgVLEMqBCPQFr2

vKNrRXHgVLEMqBCPQFr2
+ Start collection
+ Add field
Email: "ironman@dal.ca"
Location: "New York"
Name: "Tony Stark"
Password: "marvel"

Figure 24: Firebase Document after registering Tony Stark


5410_Users_Sta... > YThFDO7lIUcmx...

a2bn489600
+ Start collection
5410_Users
5410_Users_State

5410_Users_State
+ Add document
YThFDO7lIUcmx6auO2Pp
kOUp8XfcYwa1Rr7h06uy
x4zgzeDp1aHK0WJ7cgpk

YThFDO7lIUcmx6auO2Pp
+ Start collection
+ Add field
Name: "Benny Daniel"
Status: false

Figure 25:State of Benny Daniel before logging in

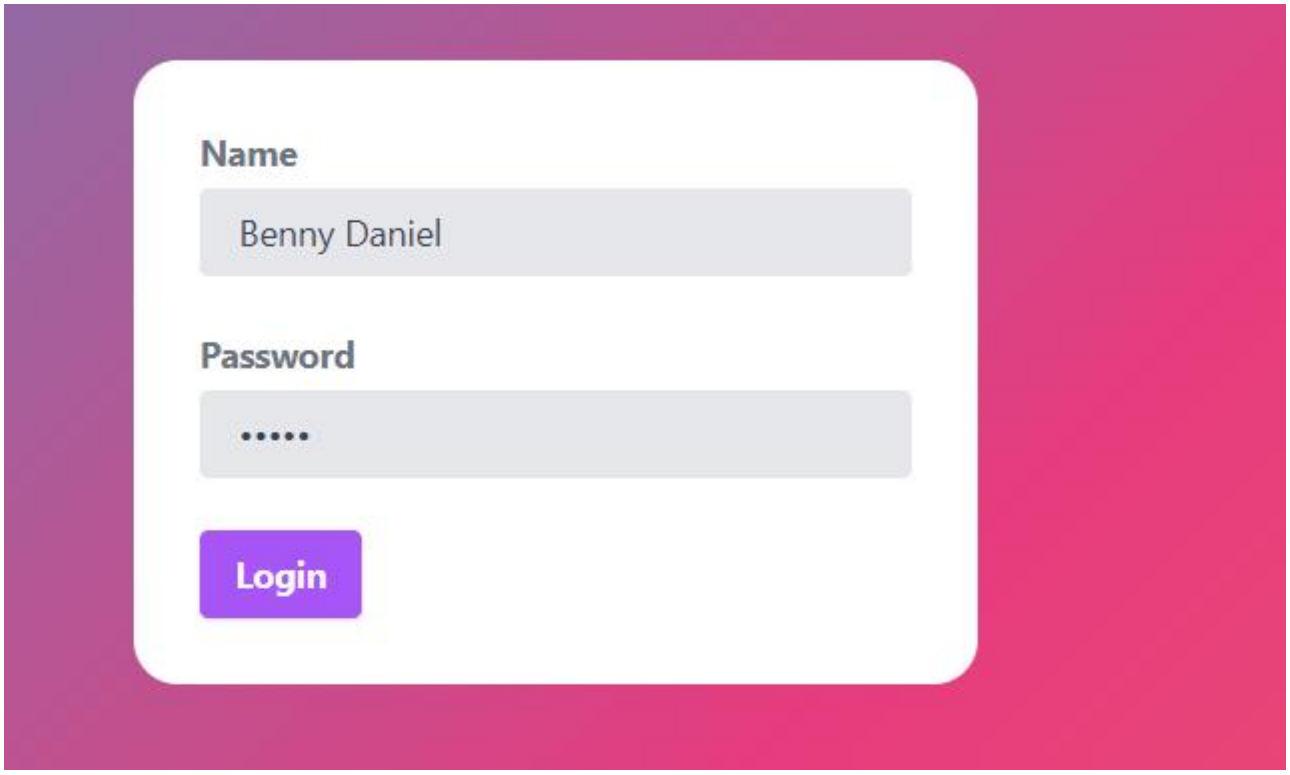
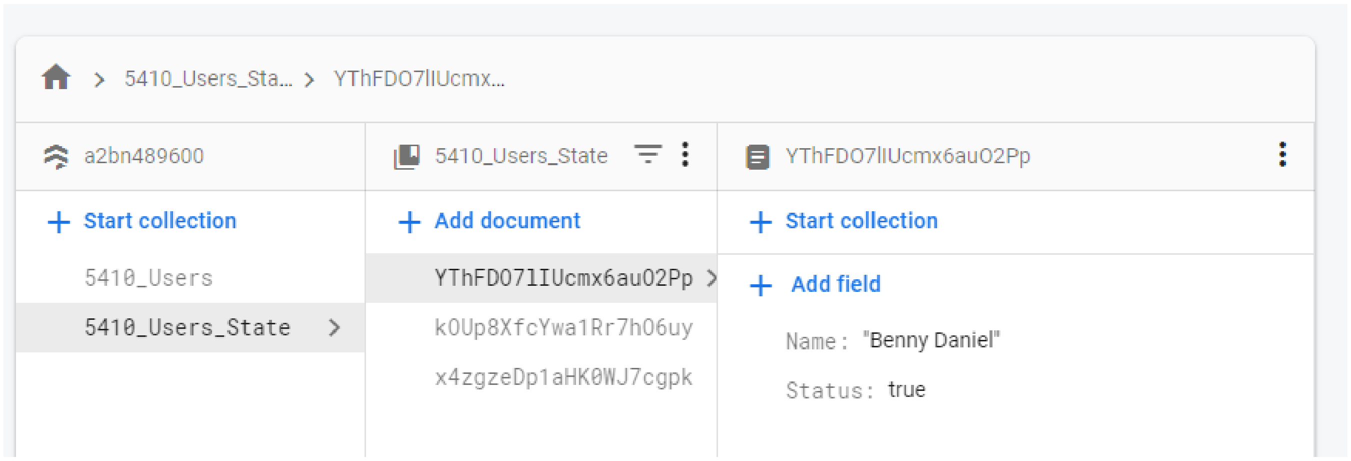


Figure 26: Benny Daniel Logs into the system

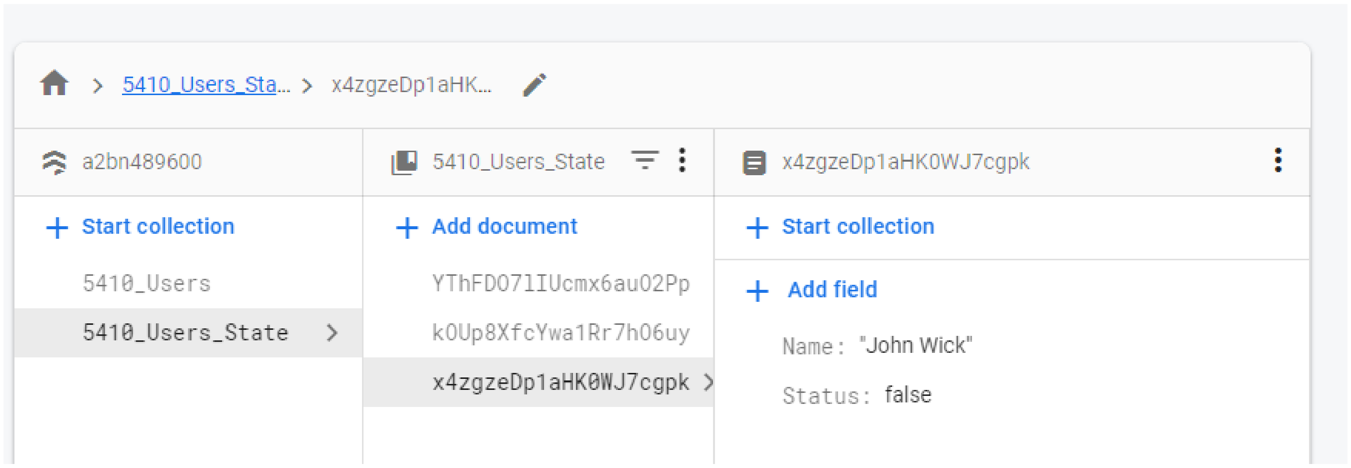


5410_Users
5410_Users_State

YThFDO7lIUcmx6au02Pp
kOUp8XfcYwa1Rr7h06uy
x4zgzeDp1aHK0WJ7cgpk

Name: "Benny Daniel"
Status: true

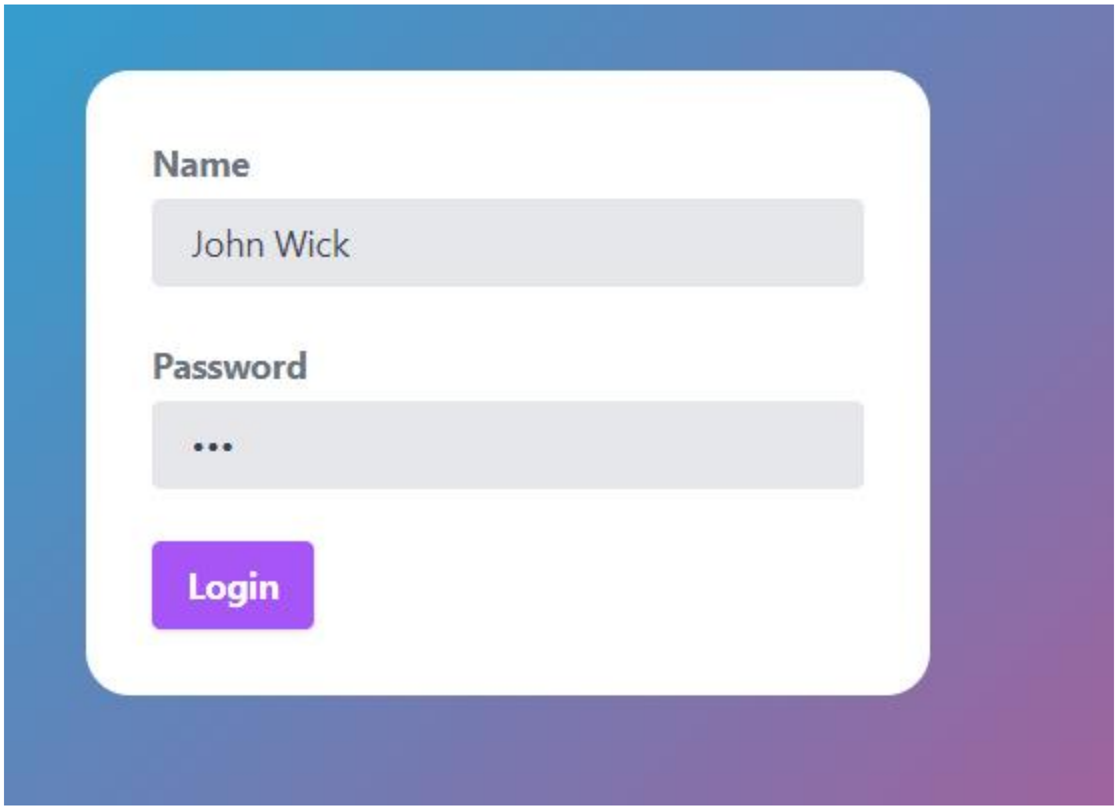
Figure 27:State of Benny Daniel after logging in



5410_Users_Sta... > x4zgzeDp1aHK...

a2bn489600	5410_Users_State	x4zgzeDp1aHK0WJ7cgpk
+ Start collection	+ Add document	+ Start collection
5410_Users	YThFD071IUcmx6au02Pp	+ Add field
5410_Users_State	k0Up8XfcYwa1Rr7h06uy	Name: "John Wick"
	x4zgzeDp1aHK0WJ7cgpk	Status: false

Figure 28:State of John Wick before logging in



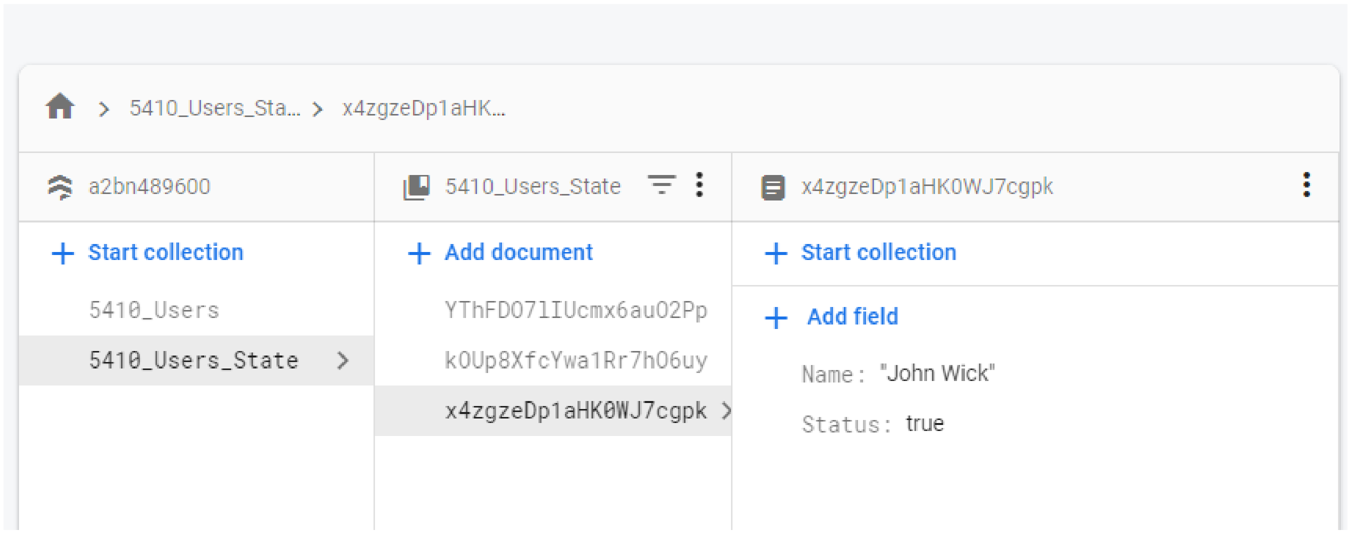
Name

John Wick

Password

•••

Figure 29: John Wick logs into the system



5410_Users	YThFD071IUcmx6au02Pp	Name : "John Wick"
5410_Users_State	k0Up8XfcYwa1Rr7h06uy	Status : true
	x4zgzeDp1aHK0WJ7cgpk	

Figure 30:State of John Wick after logging in

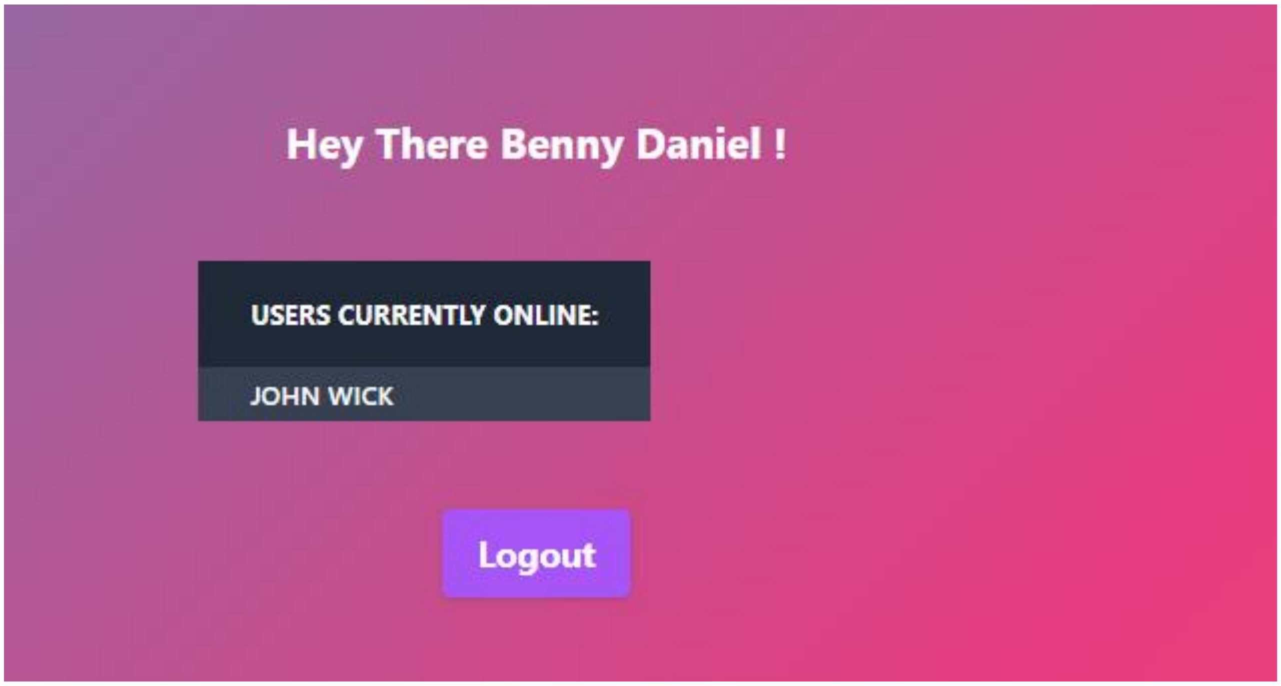


Figure 31: Profile page for Benny when Benny and John are online

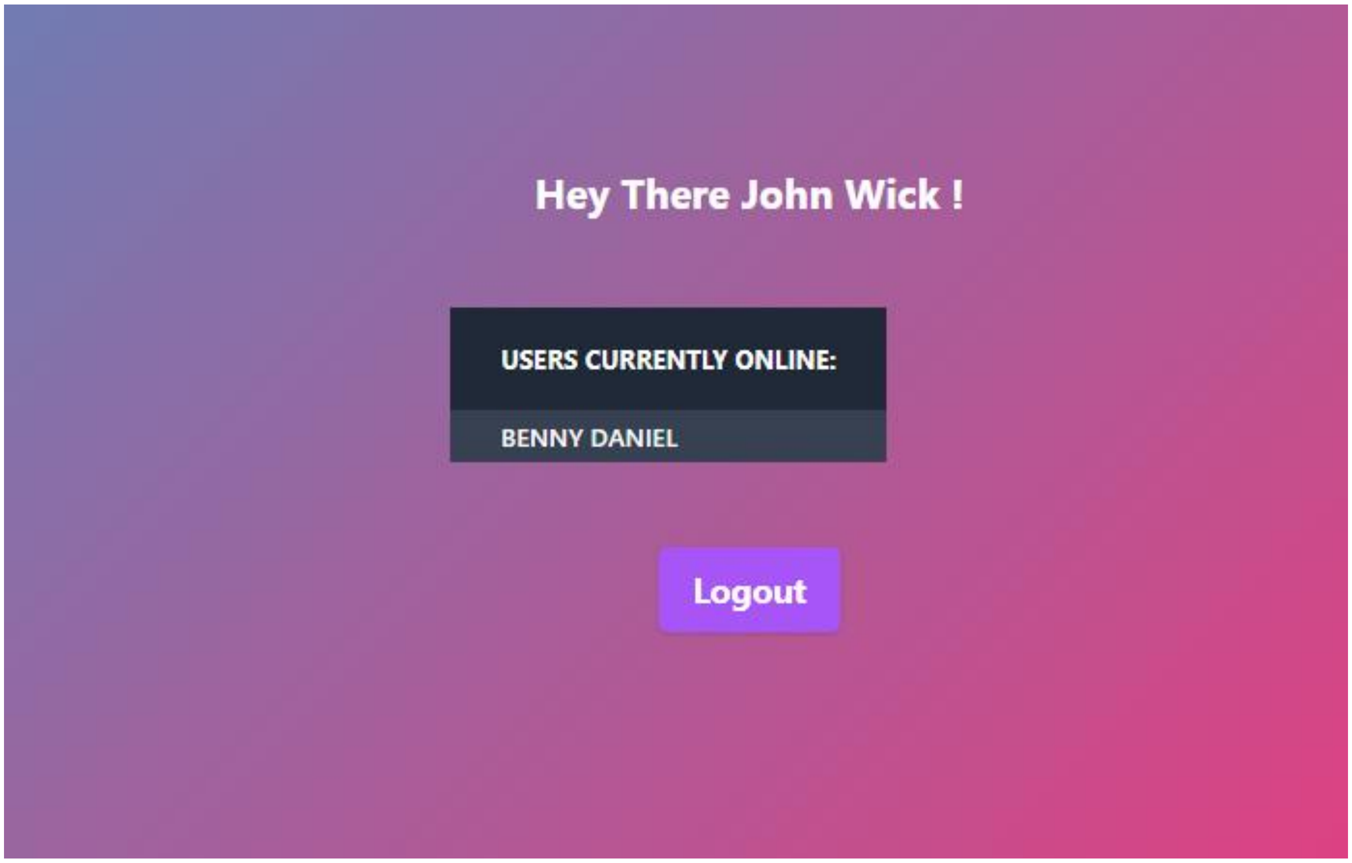
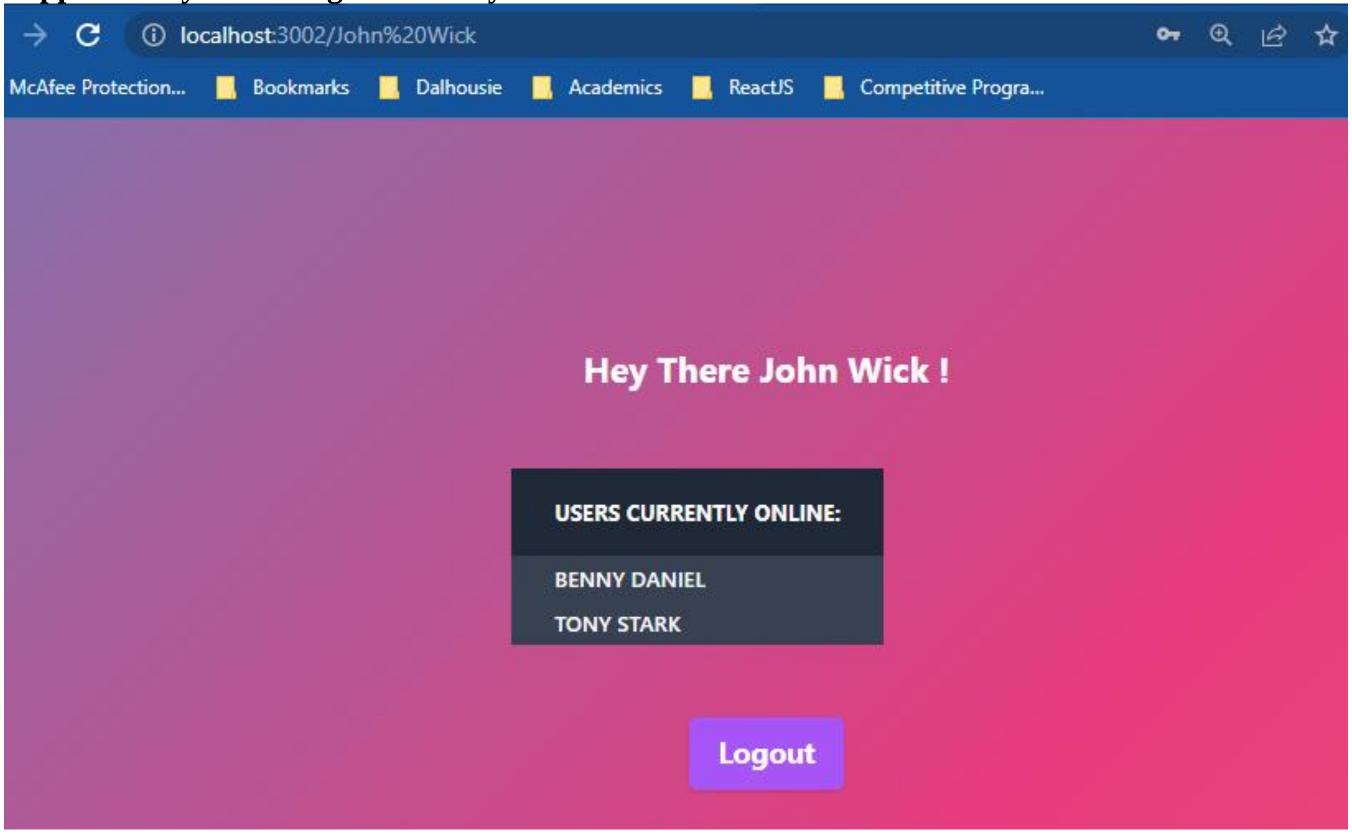


Figure 32: Profile page for John when Benny and John are online

Suppose Tony Starks logs into the system:



Hey There John Wick !

USERS CURRENTLY ONLINE:

BENNY DANIEL

TONY STARK

Figure 33: Profile page for John when Benny, Tony and John are online

Development & Deployment Process

The ReactJS web application library was used for Part A of this assignment. First, I created a React component (application) for the Registration page, as shown in Figure 9. Subsequently, I developed 2 more React components, one for the Login page and another for the Profile page, as shown in Figures 17 and 22, respectively.

Registration Page:

Using the Firebase API, I connected the Registration page to a project on the Cloud. Subsequently, a connection between the page and 2 Collections, namely, **5410_Users** & **5410_Users_State**, was established so that whenever Users furnish valid information and register themselves in the System, a new Documents corresponding to User details and Online Status are created in the aforementioned collections. After registration, Users will be redirected to the Login page. The fields are validated using HTML5 validation, namely **required** & **pattern** properties.

Example:

```
<input
              class='bg-gray-200 appearance-none border-2 border-gray-200 rounded w-
full py-2 px-4 text-gray-700 leading-tight focus:outline-none focus:bg-white
focus:border-purple-500'
              name='Password'
              type='password'
              placeholder='Password...'
            pattern='^[A-Za-z0-9$%^&*/(/)]{5,50}$'
              required='true'
              onChange={(event) => {
                setNewPassword(event.target.value);
              }}
            />
```

Code Snippet:

```
const usersCollectionRef = collection(db, '5410_Users');
  const usersStateRef = collection(db, '5410_Users_State');

  //Add Data to the 2 collections, setting the Online status to False by Default

  const createUser = async () => {
    await addDoc(usersCollectionRef, {
      Name: newName,
      Password: newPassword,
      Email: newEmail,
      Location: newLocation,
    });

    await addDoc(usersStateRef, {
      Name: newName,
      Status: false,
    });
    window.location.href = 'http://localhost:3001/';
  };
```

Login Page:

When a valid User (one whose details are present in the Firebase collection) attempts to log into the system, the script validates if the username and password combination exist in the database. If true, the Online status of the User is set to **True** in the “**5410_Users_State**” collection and the User is redirected to the Profile page.

Code Snippet:

```
const [users, setUsers] = useState([]);

const getUsers = async () => {
    const data = await getDocs(usersCollectionRef);
    setUsers(data.docs.map((doc) => ({ ...doc.data(), id: doc.id })));
  };
  getUsers();

const handleSubmit = (e) => {
    e.preventDefault();
    console.log(users);
    for (let user of users) {
      const userDetails = { ...user };
      if (userDetails.Name == newName && userDetails.Password == newPassword) {
        const nm = userDetails.Name;
        console.log('USER EXISTS IN THE DB - VALIDATION SUCCESSFUL');
        console.log(newName);
        updateFirebase();
        // navigate('/Profile', { Name: nm });
        window.location.href = 'http://localhost:3002/' + nm;
      }
    }
  };
```

Profile Page:

When Users are redirected to their Profile page, they are greeted with a personalized Welcome message along with the information of other Users who are “Online”. This is achieved by analyzing the Users whose Online Status is set to True in Firebase and retrieving their names.

Code Snippet:

```
let { loggedInUser } = useParams();

const getUsers = async () => {
    const data = await getDocs(usersCollectionRef);
    setUsers(data.docs.map((doc) => ({ ...doc.data(), id: doc.id })));
  };
  getUsers();
  const onlineUsers = users.filter((user) => {
    return user.Status == true; //AND User.name!= loggedInUser
  });

  const listItems = onlineUsers.map((user) => (
    <tr>
      {' '}
      <td
        scope='row'
        class='px-6 py-1 font-medium text-gray-900 dark:text-white whitespace-nowrap'
      >
        {user.Name}
      </td>
    </tr>
  ));

const handleLogout = async () => {
    updateFirebase();
    window.location.href = 'http://localhost:3001/';
  };

  const updateFirebase = async () => {
    const currentUser = users.filter((user) => {
      return user.Name == loggedInUser;
    });
    const logOffUserDoc = doc(db, '5410_Users_State', currentUser[0].id);
    const onlineStatus = { Status: false };
    await updateDoc(logOffUserDoc, onlineStatus);
  };


<table class='px-10 py-10'>
          <thead class='text-xs text-gray-700 uppercase bg-gray-50 dark:bg-gray-700
dark:text-gray-400'>
            <tr class='border-b dark:bg-gray-800 dark:border-gray-700 odd:bg-white
even:bg-gray-50 odd:dark:bg-gray-800 even:dark:bg-gray-700'>
              <th class='px-6 py-4 font-large text-gray-900 dark:text-white whitespace-
nowrap'>
                Users Currently Online:
              </th>
            </tr>
            {listItems}
          </thead>
        </table>
```

Docker Images and Containers:

After the apps were developed and tested locally, the three apps were built into 3 Docker images. Subsequently target image tags were created for all 3 images and pushed to Google's container registry. Finally, the 3 images were executed on Cloud Run to generate URLs corresponding to the three pages. The concept of dockers, images and Cloud Run can be compared to Code management with GitHub. Similar to how we store code on remote repositories we store images on Google's artifact registry. These images are application scripts built into images. When these images are containerized, the scripts which compose the image, are executed sequentially, thereby resulting in a web application, in this scenario.

Citations

[1] "Authentication Methods | Container Registry Documentation." *Google Cloud*, 13 Mar. 2016, cloud.google.com/container-registry/docs/advanced-authentication. Accessed 8 June 2022.

[2] "Quickstart: Install the Google Cloud CLI | Google Cloud CLI Documentation." *Google Cloud*, 4 Nov. 2018, cloud.google.com/sdk/docs/install-sdk. Accessed 5 June 2022.

[3] "Pushing and Pulling Images | Container Registry Documentation." *Google Cloud*, 14 Mar. 2017, cloud.google.com/container-registry/docs/pushing-and-pulling. Accessed 8 June 2022.