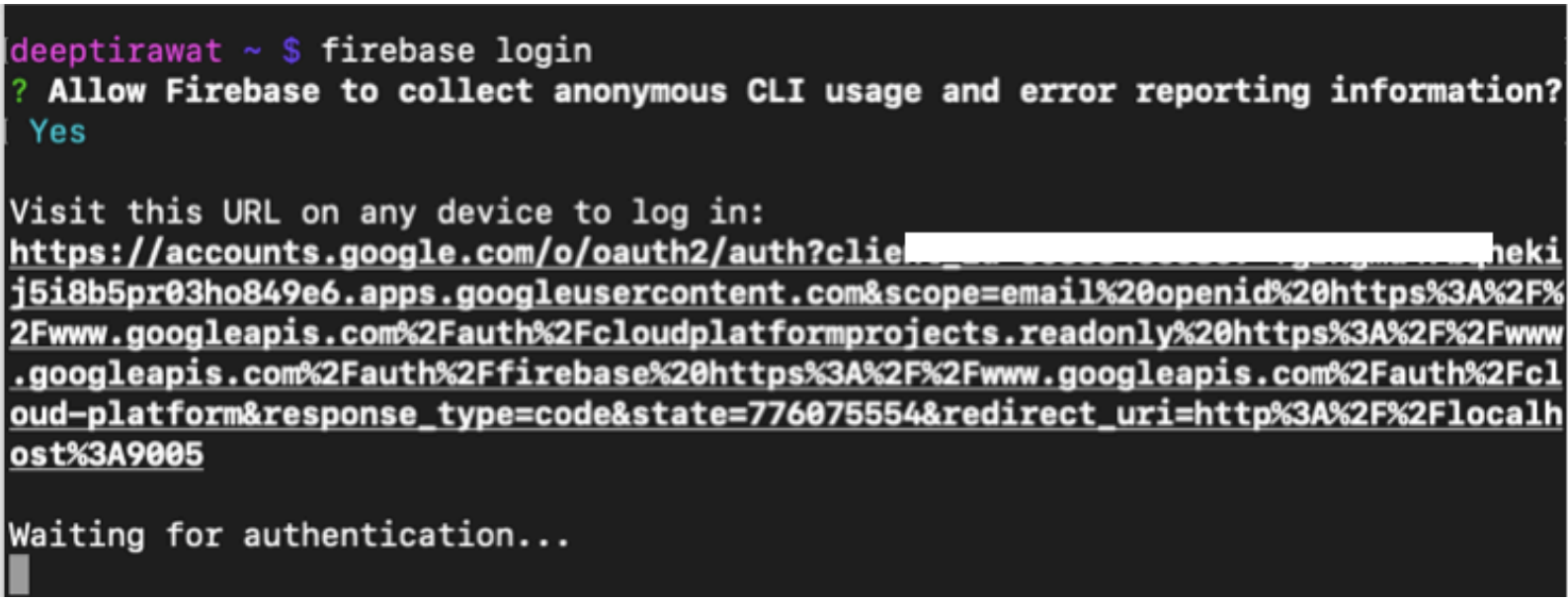
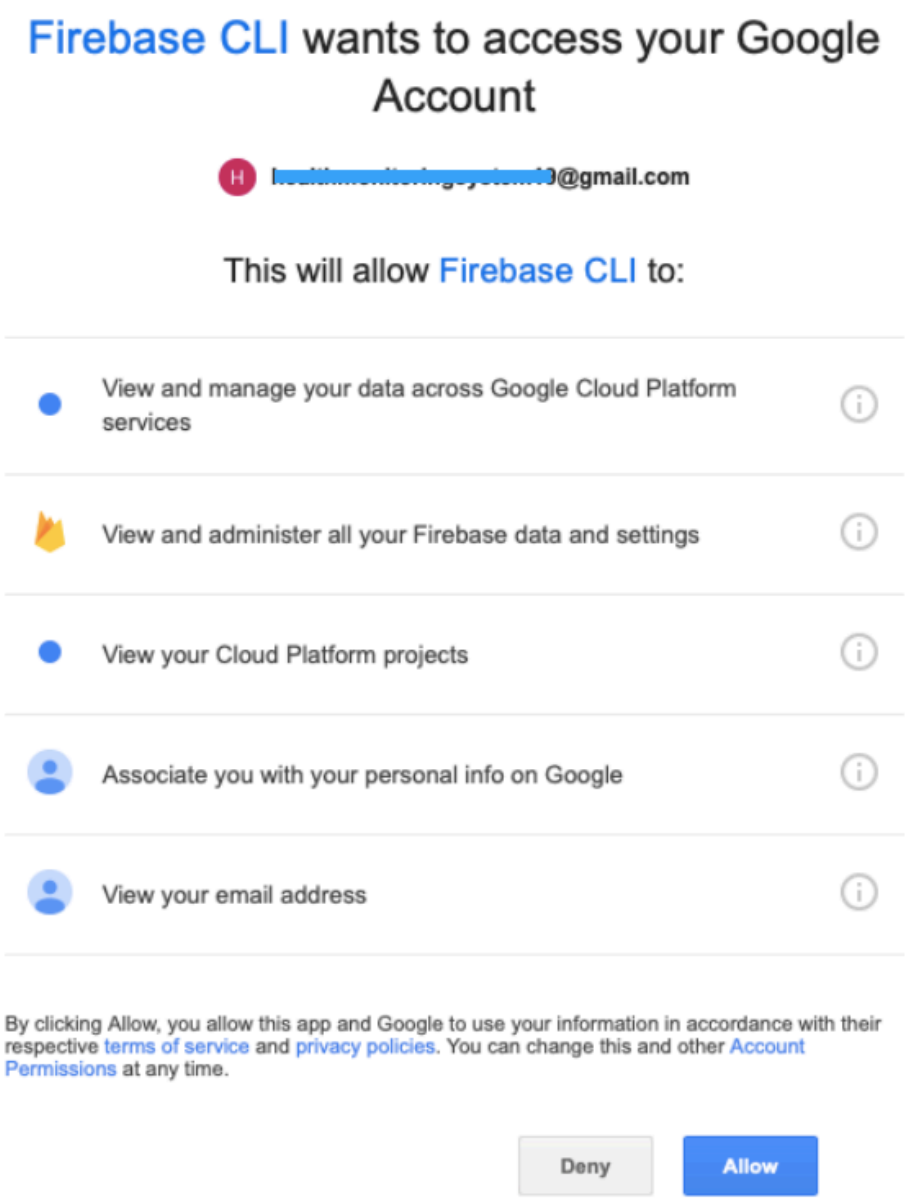
Steps to Host web application on the firebase (HealthCare_web)

- First install the firebase CLI (command line interface) on your system
- For installing the CLI in my system, I followed the following steps:
  - I downloaded nodeJs from the site - https://nodejs.org/en/download/
  - Then I type this in my command prompt - **node -v** (if it shows the version it means I have downloaded nodeJs successfully)
  - Then I type this command to install firebase CLI - **npm install -g firebase-tools**
  - After the above command execute successfully, I will login to firebase by typing - **firebase login**

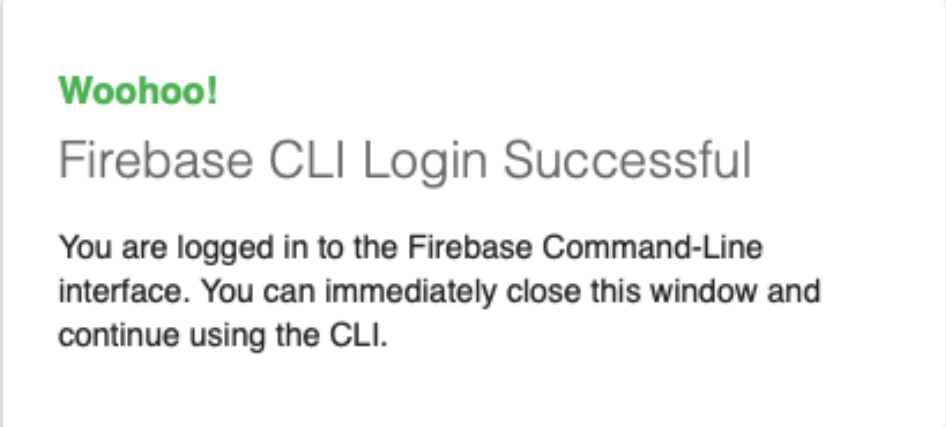
  - Then I just grant permission by typing **Y**
  - Then it takes us to another page where I choose the email from which I want to login to firebase (shown as follows)



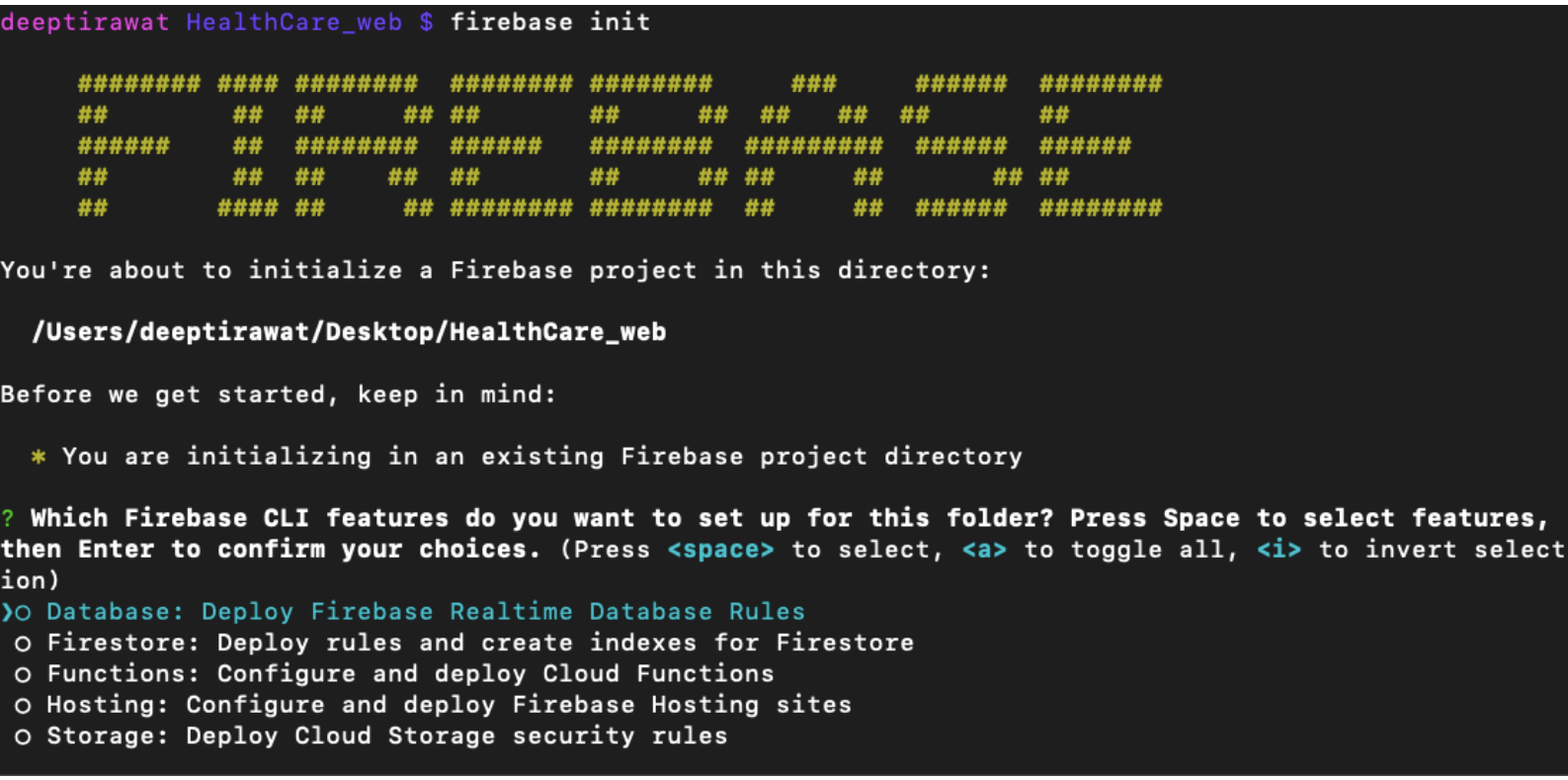
- After I click on the <email>, it asks the following permission:



- After I allow, it shows the following



- Now come back to the terminal where you are logged into firebase (with <email>)
- After installing firebase CLI, we can now run firebase from the terminal or command prompt as shown in the following example.
- Now from the terminal, go to the folder/directory of the project (in my case, the folder is named **HealthCare_web** and it is stored in the desktop)
- Now, go to the **HealthCare_web** folder from the command prompt and Initialise firebase using '**Firebase init**' command (as follows)



- Then it will ask you the firebase services you want to use in your project
- After selecting the first option ('Database'), the fourth option ('Hosting'), click 'Enter' (select these options using 'space' button on keyboard)

- Then, It will show you all the projects currently in your firebase dash-board. Select the firebase project you want to activate (in our case, the project is human-health-care) and click 'Enter'.

```
=== Project Setup

First, let's associate this project directory with a Firebase project.
You can create multiple project aliases by running firebase use --add,
but for now we'll just set up a default project.

i  .firebaserc already has a default project, using human-health-care.
```

- After the **Project Setup** is completed, it will ask a few more things for **Database Setup** and **Hosting Setup**, like ' files for database rules' and 'file for public directory', just click 'Enter' and the by-default option
- will be filled. When it asks to overwrite public directory, write N (which stands for 'No'), not doing so will re-write your code (something which we don't prefer)
- With that, you'll complete your initialisation process

```
=== Database Setup

Firebase Realtime Database Rules allow you to define how your data should be
structured and when your data can be read from and written to.

? What file should be used for Database Rules? database.rules.json
? File database.rules.json already exists. Do you want to overwrite it with the Database Rules for
? File database.rules.json already exists. Do you want to overwrite it with the Database Rules for
from the Firebase Console? Yes
✔  Database Rules for  have been downloaded to database.rules.json.
Future modifications to database.rules.json will update Database Rules when you run
firebase deploy.

=== Hosting Setup

Your public directory is the folder (relative to your project directory) that
will contain Hosting assets to be uploaded with firebase deploy. If you
have a build process for your assets, use your build's output directory.

? What do you want to use as your public directory? public
? Configure as a single-page app (rewrite all urls to /index.html)? Yes
? File public/index.html already exists. Overwrite? No
i  Skipping write of public/index.html

=== Storage Setup

Firebase Storage Security Rules allow you to define how and when to allow
uploads and downloads. You can keep these rules in your project directory
and publish them with firebase deploy.

? What file should be used for Storage Rules? storage.rules

i  Writing configuration info to firebase.json...
i  Writing project information to .firebaserc...

✔  Firebase initialization complete!
deeptirawat HealthCare_web $
```

- From the code directory (**HealthCare_web** in this case), use the '**Firebase deploy**' command to host the contents of the local machine on to the cloud.
- The deploy command, will host your site and will render you a link where you can see the website.
- In our case, the received link is: https://human-health-care.firebaseapp.-com
- Use the link to visit your site globally from anywhere!

```
deeptirawat HealthCare_web $ firebase deploy

=== Deploying to 'human-health-care'...

i  deploying database, storage, hosting
i  database: checking rules syntax...
✔  database: rules syntax for database human-health-care is valid
i  storage: checking storage.rules for compilation errors...
⚠  [W] undefined:undefined - Ruleset uses old version (version [1]). Please update to the latest ver
sion (version [2]).
✔  storage: rules file storage.rules compiled successfully
i  storage: uploading rules storage.rules...
i  hosting[human-health-care]: beginning deploy...
i  hosting[human-health-care]: found 24 files in public
✔  hosting[human-health-care]: file upload complete
i  database: releasing rules...
✔  database: rules for database human-health-care released successfully
✔  storage: released rules storage.rules to firebase.storage/human-health-care.appspot.com
i  hosting[human-health-care]: finalizing version...
✔  hosting[human-health-care]: version finalized
i  hosting[human-health-care]: releasing new version...
✔  hosting[human-health-care]: release complete

✔  Deploy complete!

Project Console: https://console.firebase.google.com/project/human-health-care/overview
Hosting URL: https://human-health-care.firebaseapp.com
deeptirawat HealthCare_web $
```

- Now, every-time, we make some changes in the code in our local machine, after making the changes, we have to simple run 'firebase deploy' command and all the changes will be deployed on the website.