CS628 Full-Stack Development – Web App

**HOS10A: Authentication**

Created by Veerendra Jagatha on Aug 17, 2023

Reviewed by Sam Chung on September 1, 2023

School of Technology & Computing (STC)

City University of Seattle (CityU)

**Before You Start**

* **Screenshots may be different from your environment.**
* The directory path shown in screenshots may be different from yours.
* Version numbers may not match the most current version at the time of writing. If given the option to choose between the stable release (long-term support) or the most recent, please select the **stable release** rather than the beta-testing version.
* There might be subtle discrepancies along with the steps. Please **use your best judgment** while going through this cookbook-style tutorial to complete each step.
* If you are not familiar with a terminal, command line, and bash scripts, check out this video: <https://youtu.be/Dp7uw9c6QH8>
* All the steps and concepts in this tutorial are from references, so if you encounter problems, please **try to read and compare the references to solve the problem**. If you still can't solve the problem, please contact your course TA.
* **Avoid copy-pasting code from the book or the GitHub repository**. Instead, type out the code yourself. Resort to copy-pasting only when you are stuck and find things not working as expected.
* Some steps may not be explained in detail. If you are not sure what to do:

1. Consult the resources from the course.
2. If you cannot solve the problem after a few tries (usually 15 -30 minutes), ask a TA for help.

#### **Readings and Examples:**

* Visit [CS 628 Repository for Examples](https://github.com/samchung0117/cs628-examples).
  + Select the related module.
  + Visit the README.md file.
  + Find examples for your practices.

**Learning Outcomes**

* Section 1: Accessing GitHub Codespaces.
* Section 2: Configuring the Backend with Passport.js Facebook Strategy.
* Section 3: Configuring the Facebook App to get the secrets.
* Section 4: Updating the Client with Facebook Login.
* Section 5: Pushing your work to GitHub.

**Section 1: Accessing GitHub Codespaces.**

Before we start the current code space, use the steps mentioned in **CopyRepo.md** in the hos-examples to move your backend code and client you worked in earlier modules to current repository. We will be adding Passport.js Facebook authentication to the same example.

Refer the steps from [TA Center](https://cityuseattle.github.io/docs/git/github_codepsace/) to get started with this week’s module GitHub Codespace.

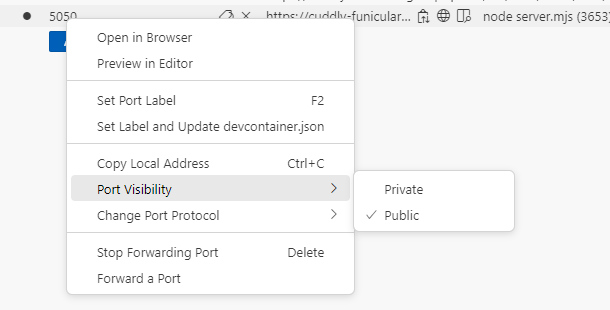
**Section 2: Configuring the Backend with Passport.js Facebook Strategy.**

Navigate to the backend directory and start the server using the below commands.  
Assume the backend directory’s name is “backend.”

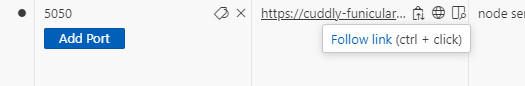
**>> cd backend/**

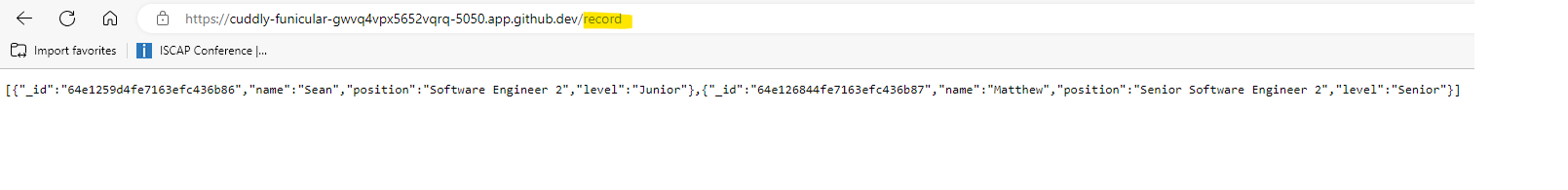
**>> node server.mjs**

Now you must make the port running your backend server public just like we did it in previous module.

****

Open the local address in the new tab and make sure the API is working,

****

****

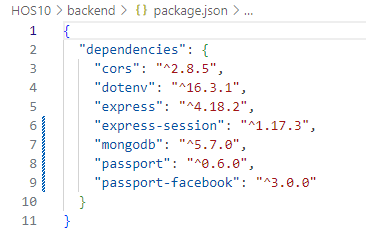
Save the backend URL from the browser. We need it for the next steps,

In my case, my URL is <https://cuddly-funicular-gwvq4vpx5652vqrq-5050.app.github.dev>. You will be having your codespace specific URL.

Now, let’s Install “express-session,” “passport,” and “passport-facebook” by running this command:

**>> npm install express-session passport passport-facebook**

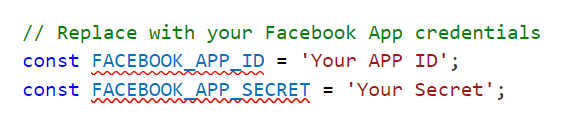
* [express-session](https://www.npmjs.com/package/express-session): create a session middleware.
* [passport](https://www.passportjs.org/docs/): create middleware for Node.js that makes it easy to implement authentication and authorization.
* [passport-facebook](https://www.passportjs.org/packages/passport-facebook/): authenticate using Facebook in your Node.js applications.



We then need to import the modules (express-session, passport, & passport-facebook) in our “server.mjs” file as shown below,



After importing the modules, let us add the placeholders for the “App\_ID” and “App\_Secret” under it. We will show you in the next section how to generate these secrets.



**Notice**:  
You will get your APP ID and Secret from. After getting them, update them.

Next step is to define the Facebook strategy.

In the “server.mjs” file, below the secrets, type the following,



The callbackURL you see in the screenshot needs to be replaced with your backend URL which is obtained earlier in this section.

**>> YourBackendURL/facebook/callback**

You can find the port number “5050” in the backend URL.

Now let s define the passport routes. Add the following code to your “server.mjs” file. You can get the entire code from the “cs628-examples.”

Again, the successRedirect and failureRedirect URL’s are your codespace specific URL’s. To obtain these, navigate to your client folder. Run the below command and capture the URL from the browser. Assume the frontend directory’s name is “frontend.”

**>>cd frontend**

**>> npm start**

**successRedirect = YourFrontEndURL/home**

**failureRedirect = YourFrontEndURL**

You can find the port number “3000” in the frontend URL.

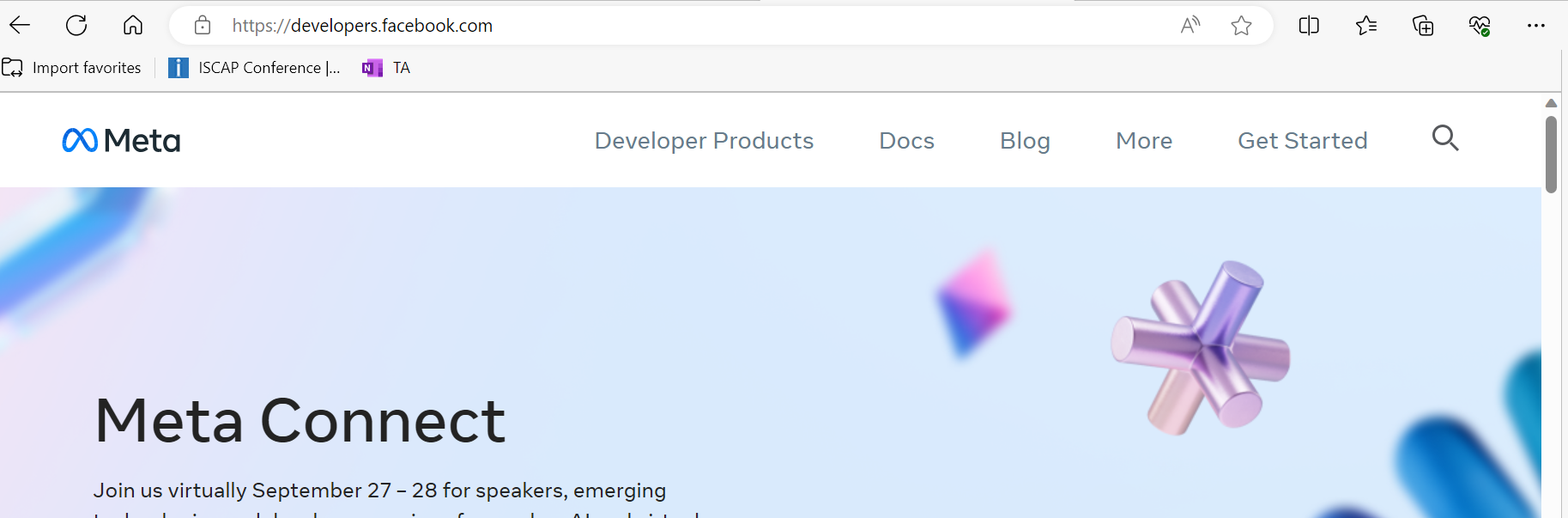
Our Backend setup is now complete.



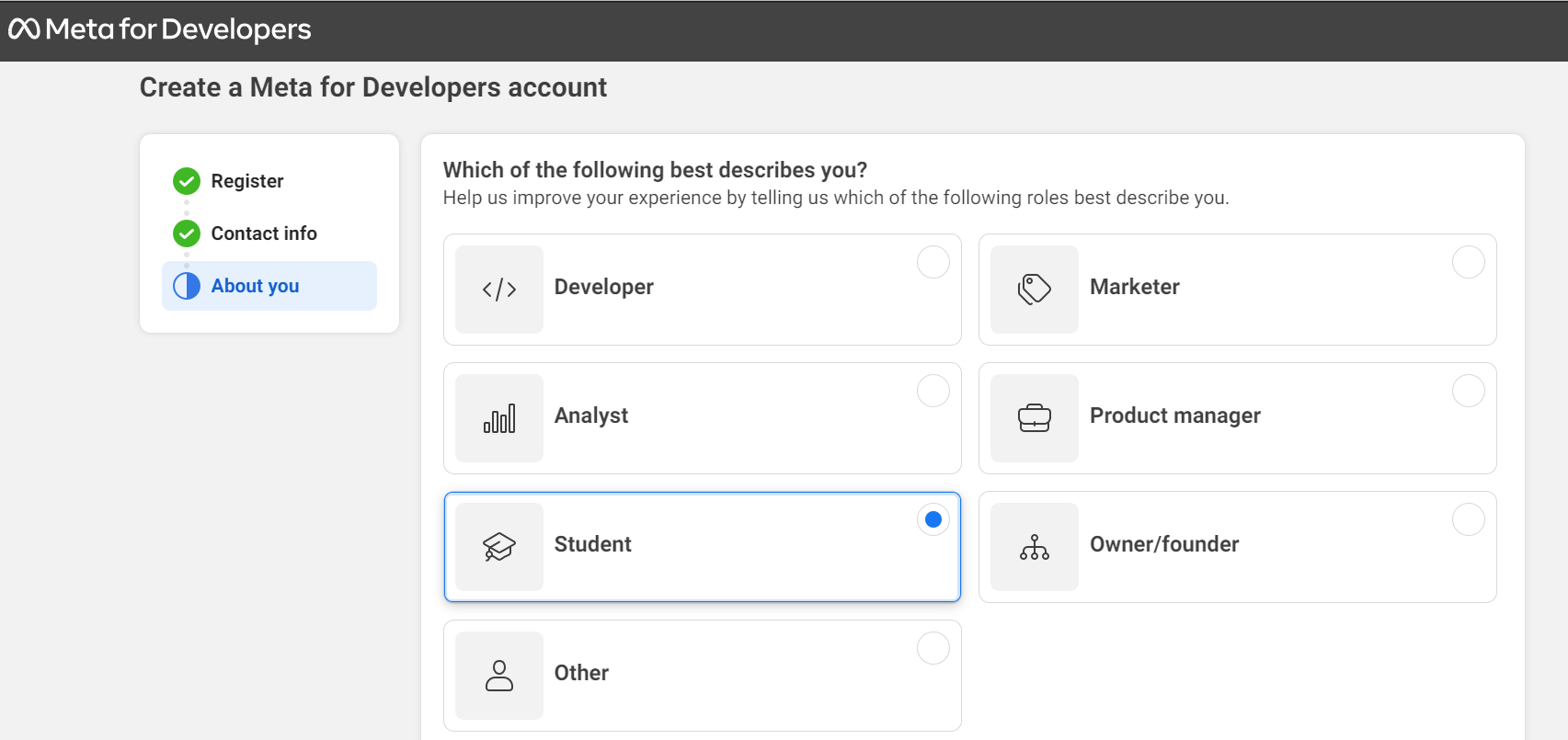
**Section 3: Configuring the Facebook App to get the secrets.**

Let us get the “App\_ID” and “App\_Secret” from the next section.

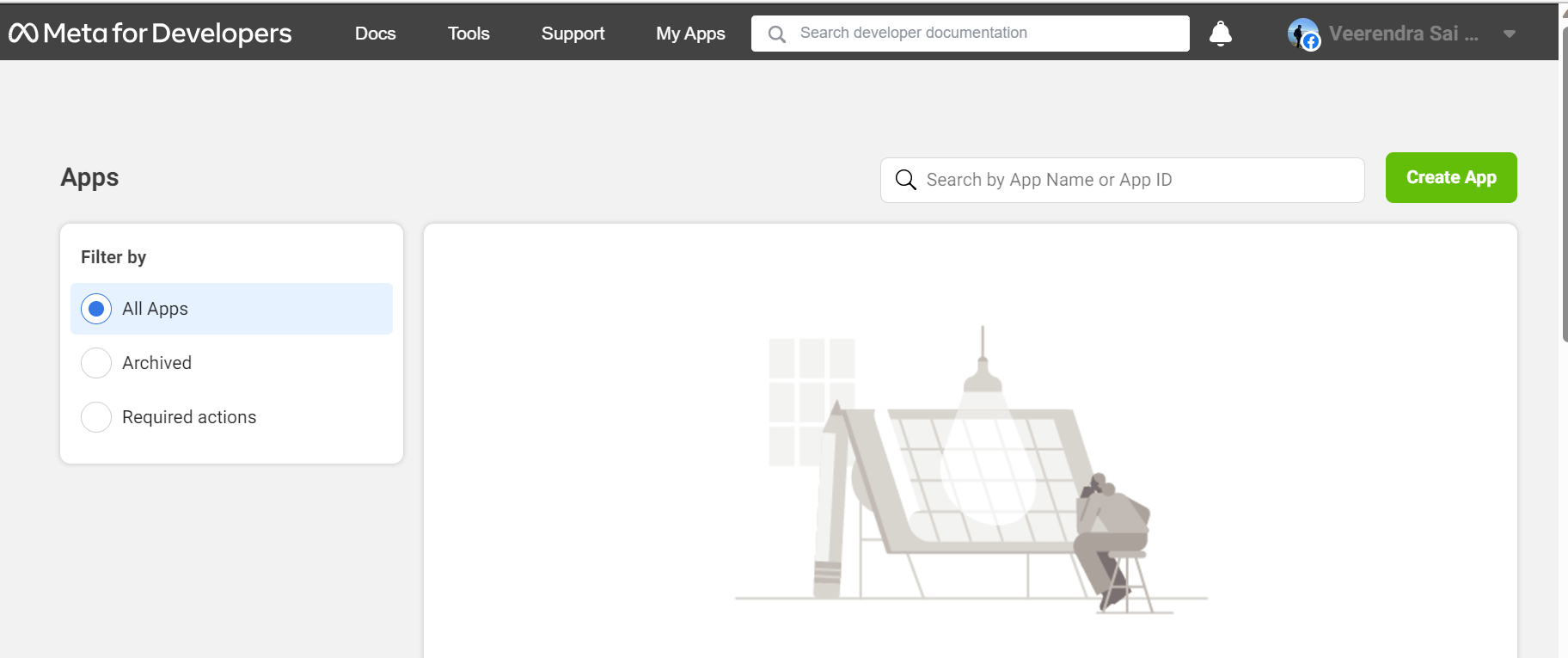
Visit the following link, <https://developers.facebook.com/>

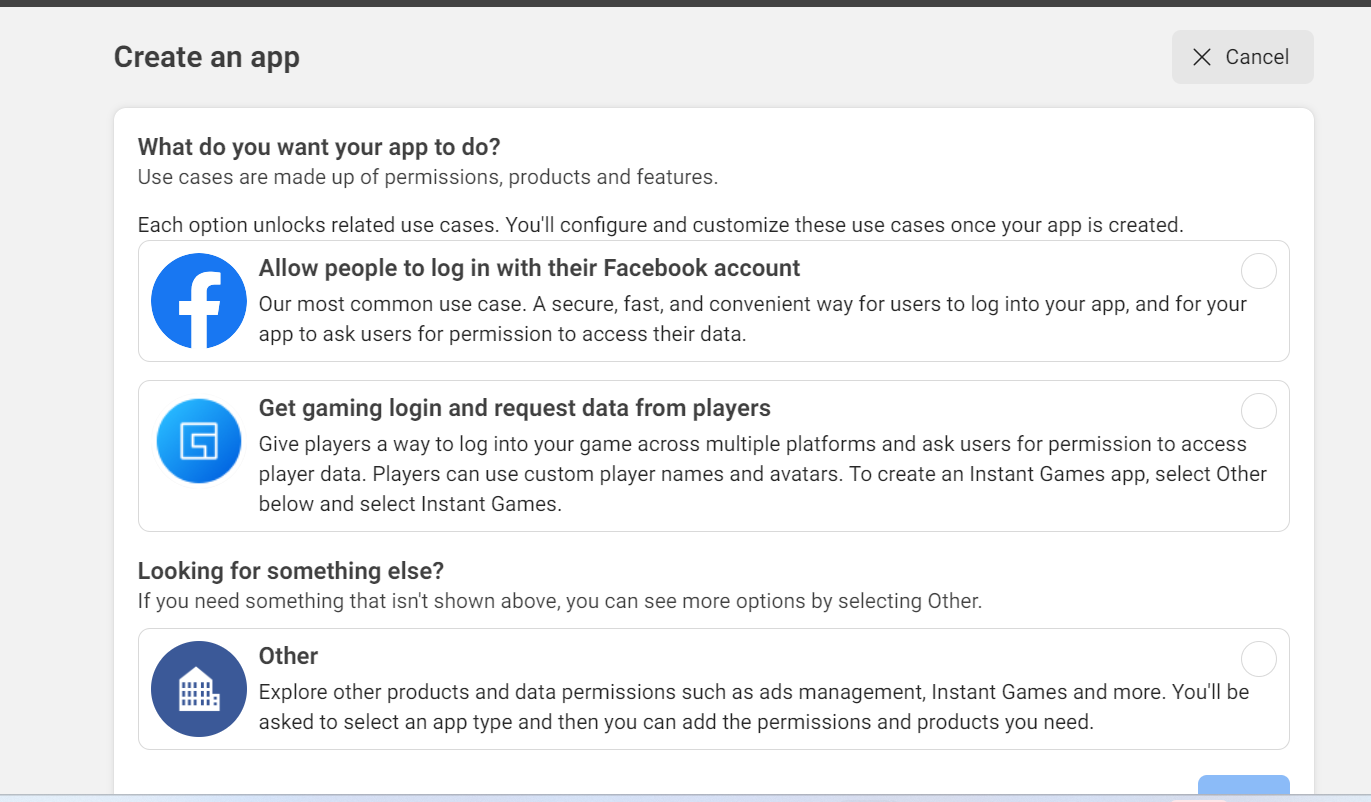


Click on Get Started and login with your Facebook account and the select Student as shown below.

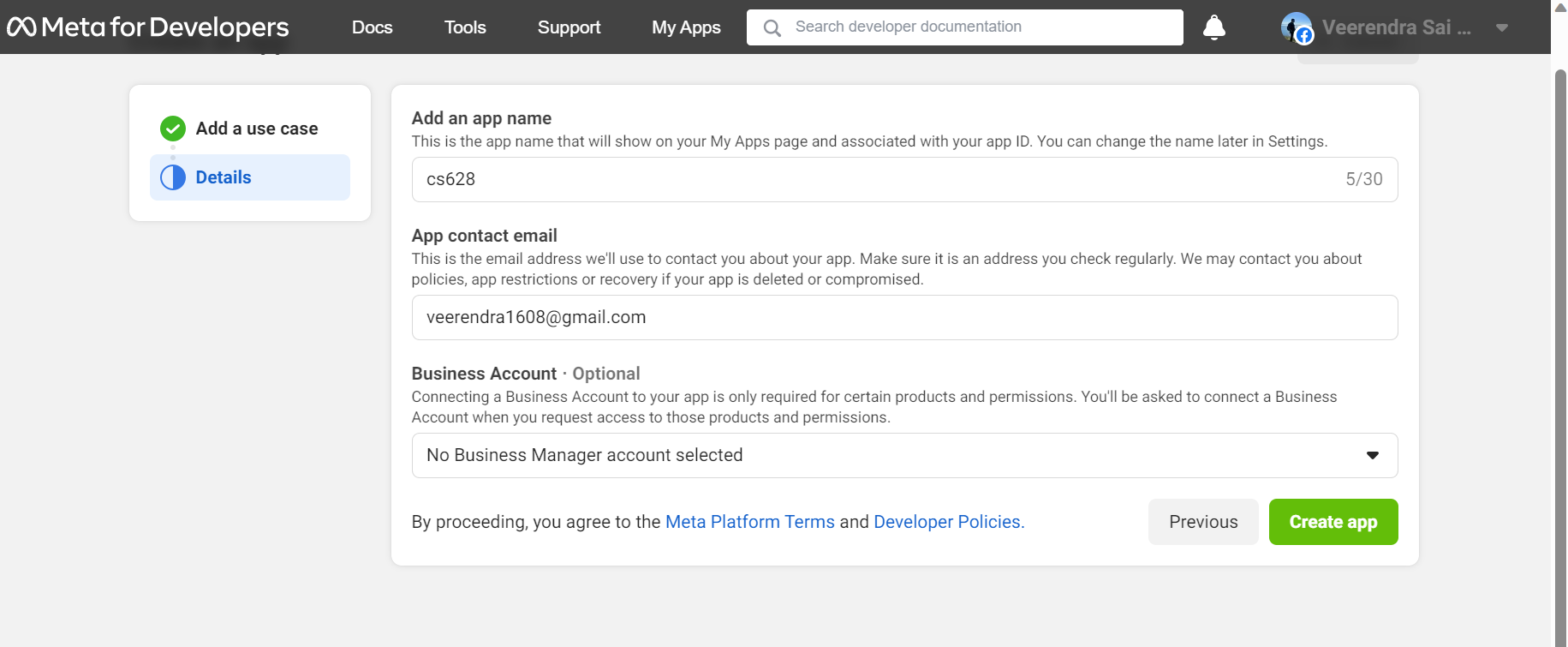


Click on Create App and select “Allow people to login with their Facebook account.”

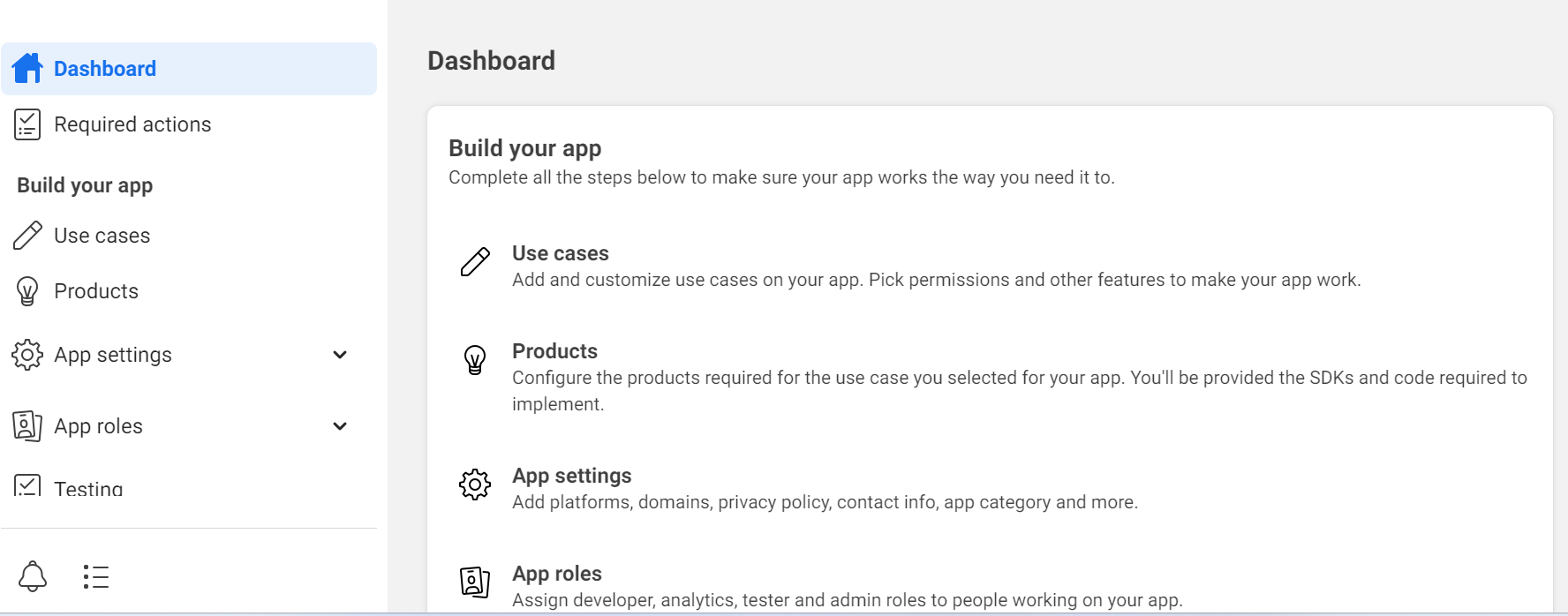


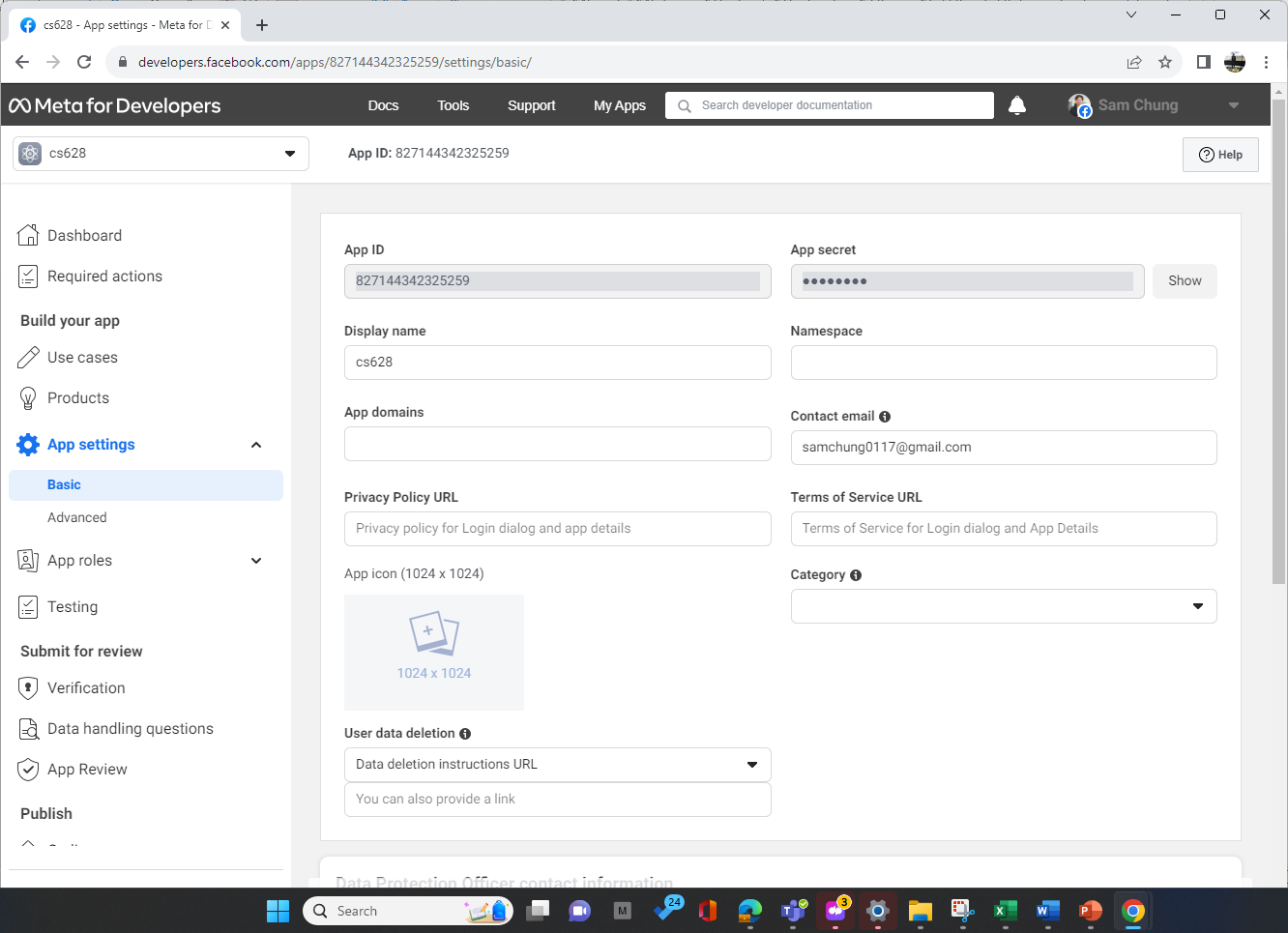


Click Next and give it a name, say “cs628.”



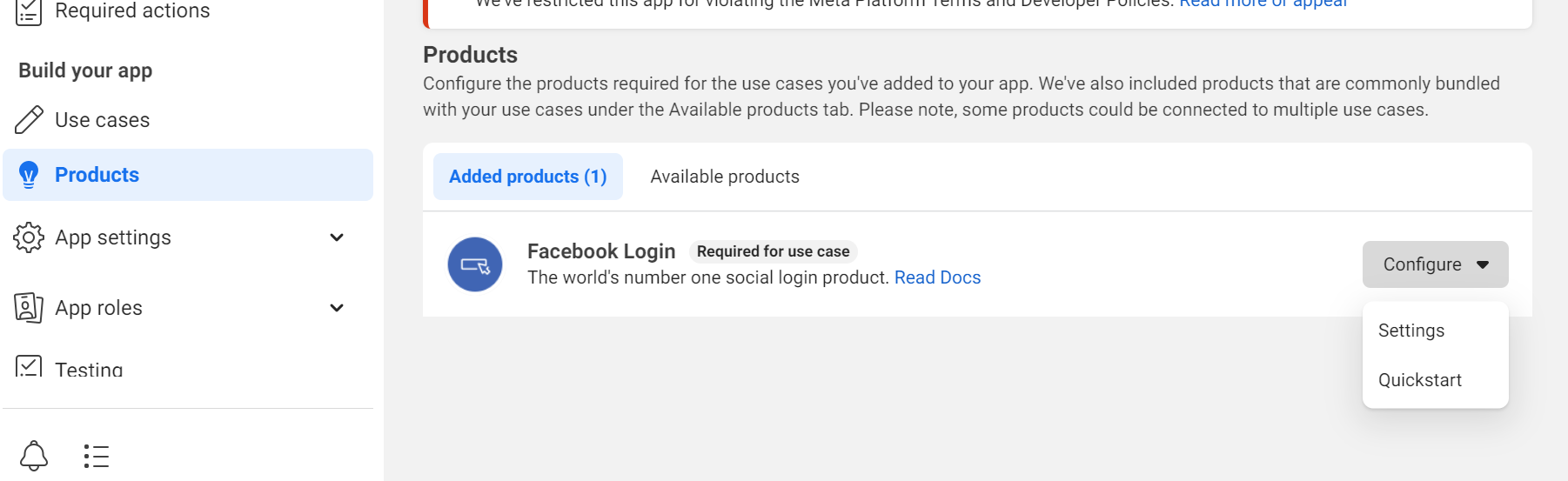
After the App is created, click on Dashboard and then “App settings -> Basic.” You will have your “App ID” and “Secret” under this screen. Save them and update those values in the “server.mjs” file.



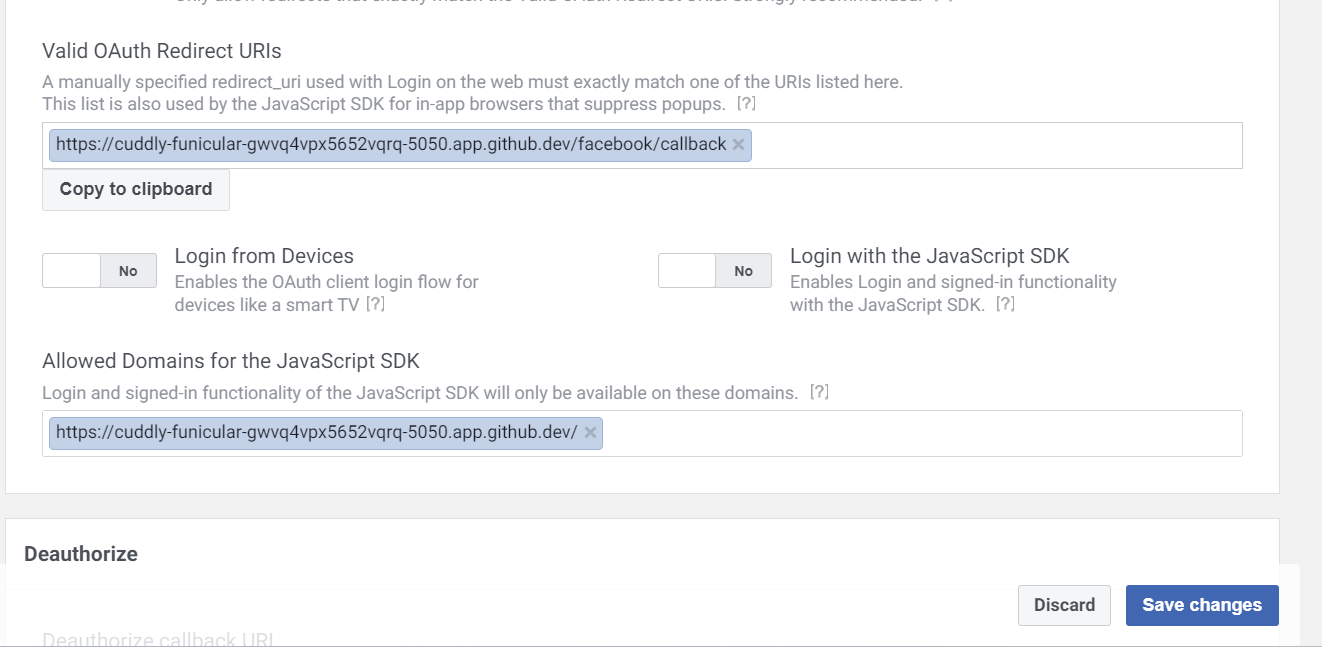


Restart your backend.

Now we need to configure the App with “Redirect URI” and “Allowed Domains.” Click on “Products -> Configure -> Settings.”



Update the “Valid OAuth Redirect URIs” with the value we had as callback URL in the “server.mjs” file. Also update the “Allowed Domains for the JavaScript SDK” section with your “BackendURL” as shown below. You should use your codespace specific URL’s. The screenshot is just for example. Save the changes with your URL’s.

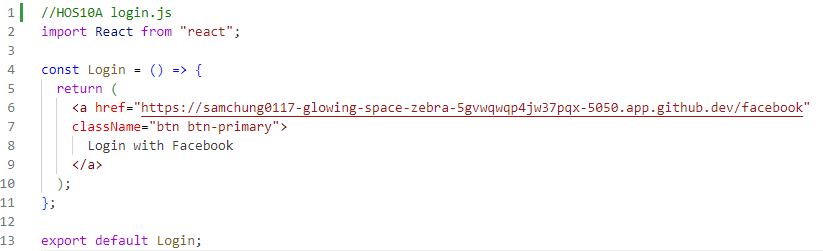


Log out the “Meta for Developers.”

**Section 4: Updating the Client with Facebook Login.**

We must make some changes to our Frontend Client to add Facebook authentication.

Add a new Login Component under components folder of the client.



The URL used here for the anchor tag should be your “backendURL/facebook”. The screenshot is showing the backend URL example.

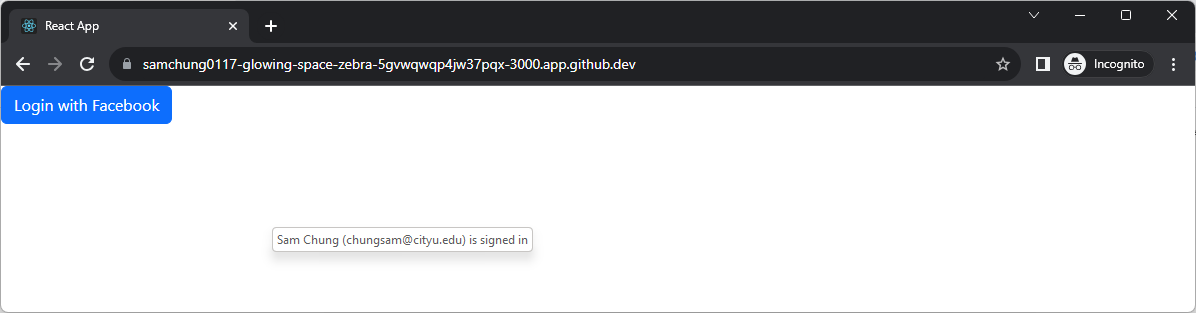
Update your “App.js” to match the following. We are changing the positioning of the Navbar Component. The “navbar” component will be used in the components – “recordList,” “create”, and “edit.” Also, the path of the RecordList component was changed from “/” to “/hone.”



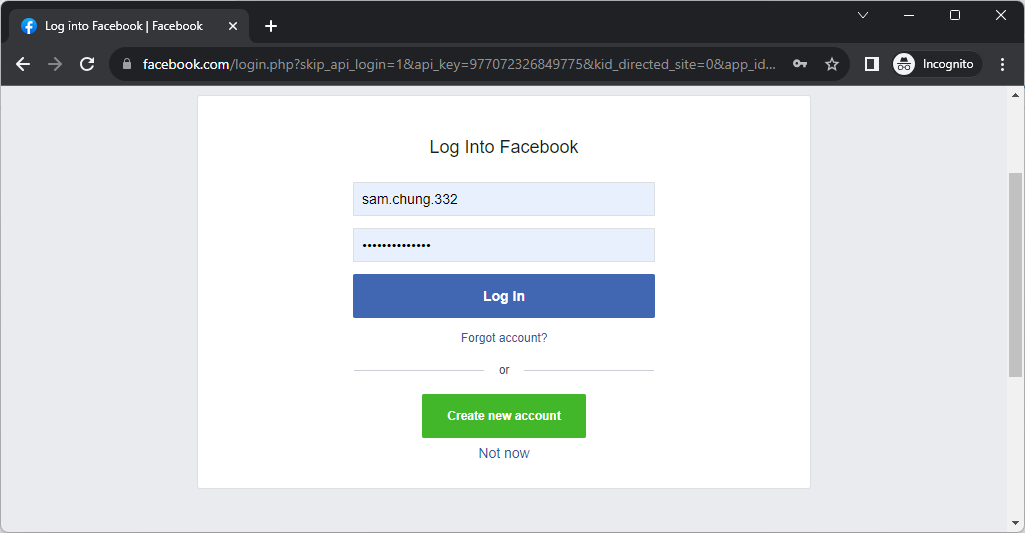
Similarly, update “create.js,” “recordList.js” and “edit.js” files with the latest content shared in the hos-examples. Only two minor changes happen: 1) “import Navbar from "./navbar";” and 2) “<Navbar />” in the return function.

After making the changes, refresh the dev browser window running the react app and you will see this screen. Make sure the backend server is also running on public port and logout from the Facebook if you are already logged in.

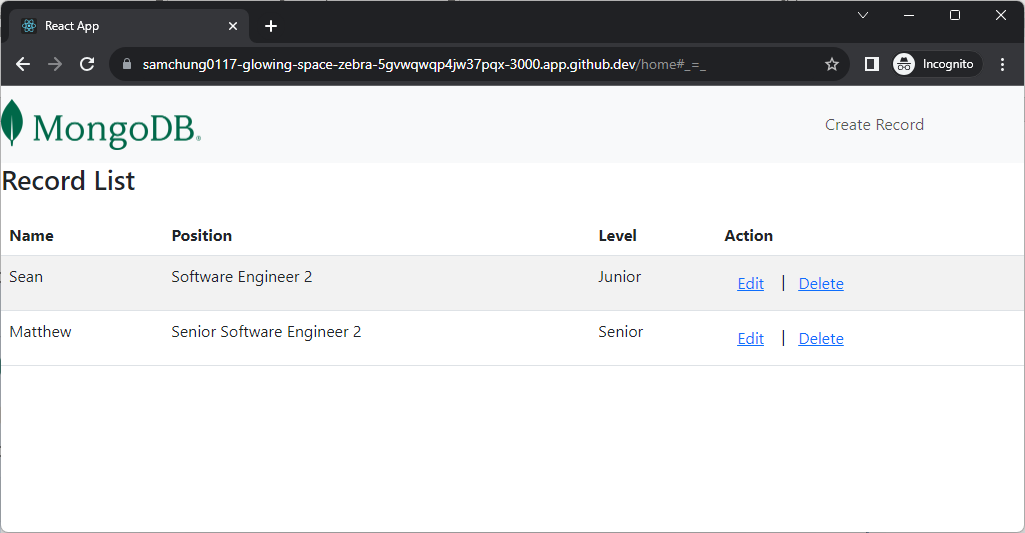
Use the private Incognito to be free from the existing cookies.



Click on Login button and then you will see this screen.



Login with your Facebook credentials, you will see the records fetched from MongoDB.



**Section 5: Pushing your work to GitHub**

* 1. Go to Source Control on your GitHub codespace and observe the pending changes.

Graphical user interface, text, application

Description automatically generated

* 1. Type the Message for your changes in the Message box on the top. For example,” **Submission for Module10 – Your Name**”
  2. Click on the dropdown beside the commit button and select **Commit & Push** to update the changes to your repository main branch.
  3. Select **Yes** when prompted.

Graphical user interface, application

Description automatically generated

**Output:**

A screenshot of a computer

Description automatically generated

A screen shot of a computer

Description automatically generated