1. **Introduction:**

Our group application is a documents auto-correction using AI integration. Which similarly with other application on the market such as Grammarly, Quillbot,… We have chosen the name of our application to be **Language Sculptor.**

Our application using **React JS** for frontend development. **Flask-Python** for backend endpoints, Authentication and File storage using **Firebase, Firestore**.

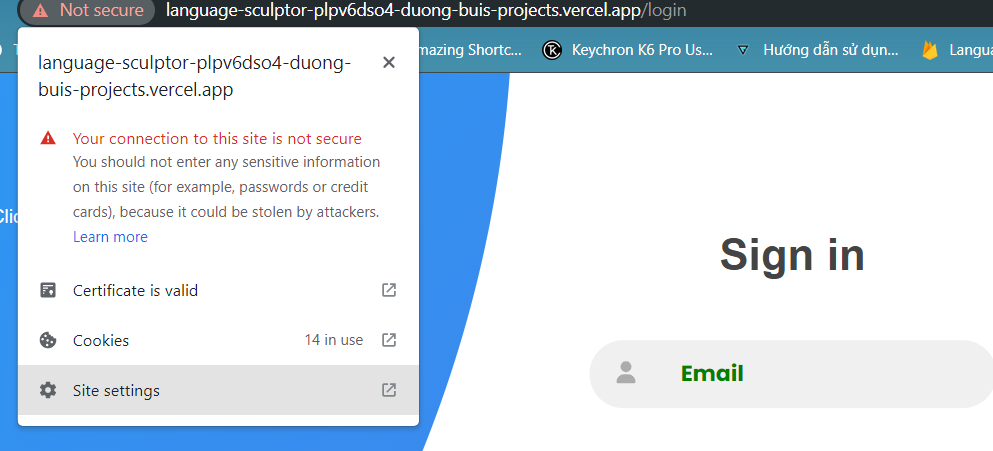
OAuth2 sign in features with **Google, Facebook and Microsoft.**

Payment implementation, we using **PayPal** with sandbox account.

For deployment, we will use **Vercel** for Frontend, and **Amazon Web Service server** to run our Flask Python backend end points. Here is the link for live deployment:

<https://language-sculptor-ai.vercel.app/login>

**Instruction on how to run the deployment link:** Due to the difference of **Vercel’s HTTPS** and the current **AWS** backend link that we are using, which is **HTTP,** you will encounter the **Mixed Content Error,** which should be display on your Web Console. Consider following these instructions:



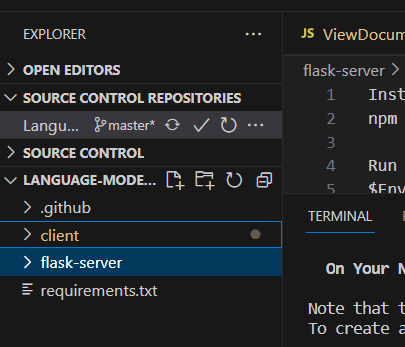
* Go and click on the **Site setting** , it is advisable to use **Chrome** to see this setting easier.



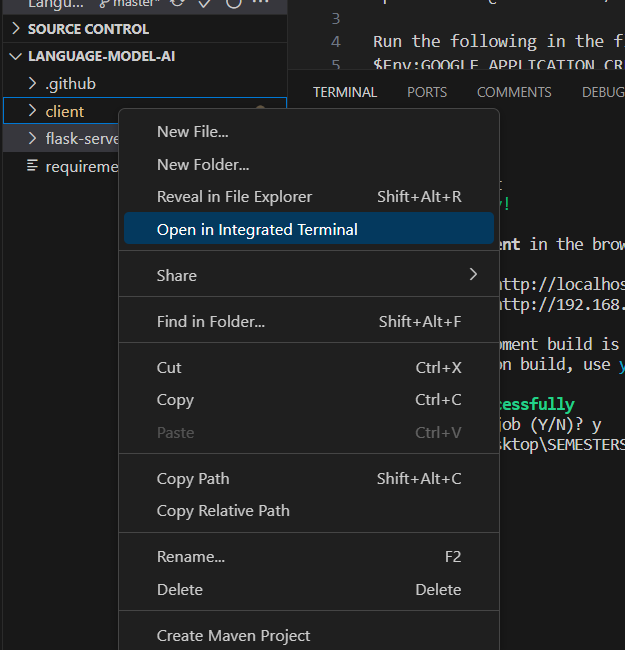
* After that, search for **Insecure content** option, and click **Allow.**
* This is a security risk issue, so it is advise to only turn this on when trying to test our **Vercel** Link above !

1. **Running Instruction and Important Notice (LOCAL RUN):**

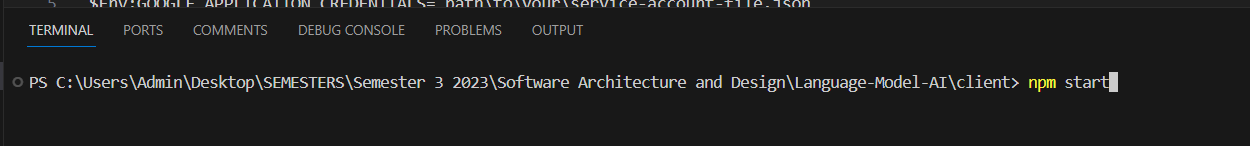
* The file that we uploaded on Canvas can be only running on **Local Host**, the one that was deployed onto Vercel and AWS are different. As those environments required some code modification that will affect the ability to running locally significantly due to their complexity. We have decided the quickest way for you to test our application is to run it **locally on your own computer.**
* Here is some Instruction to running our code:

****

* Here on the VS Code display, you will see 2 important folders, **client** is for running our **Frontend,** and **flask-server** is for running our **Backend**.

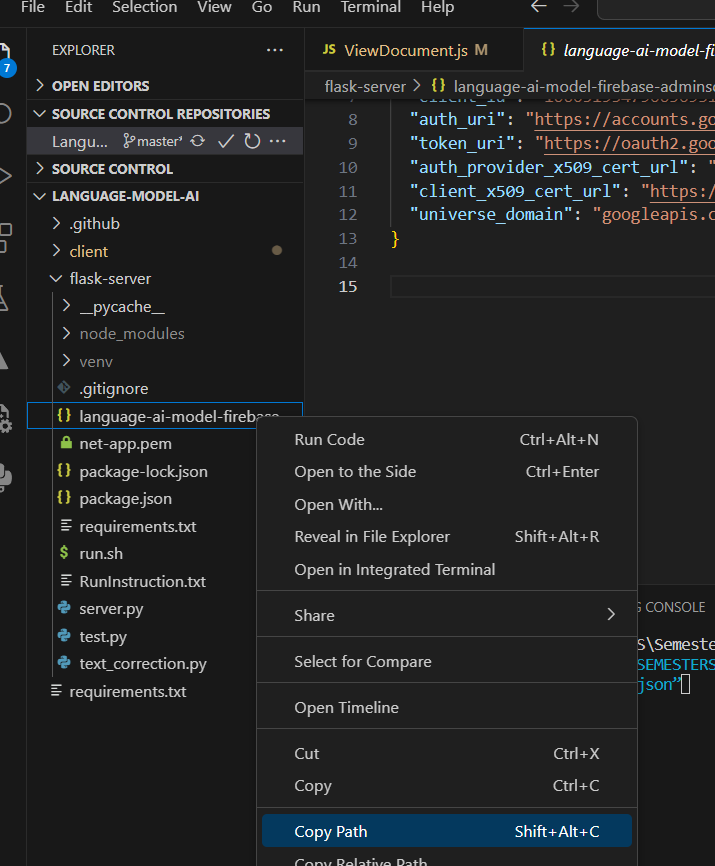


* Click on the **Open in Integrated Terminal** for **each** Folder.

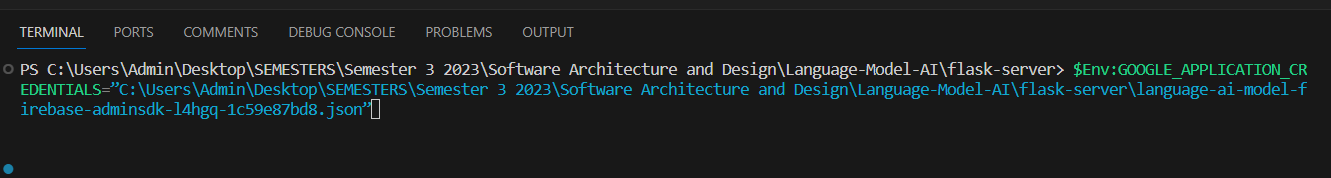


* You will see the open terminal, for this picture is the **client** folder, here you just simply run **npm start**
* After that it will automatically open a local host website on your default browser.

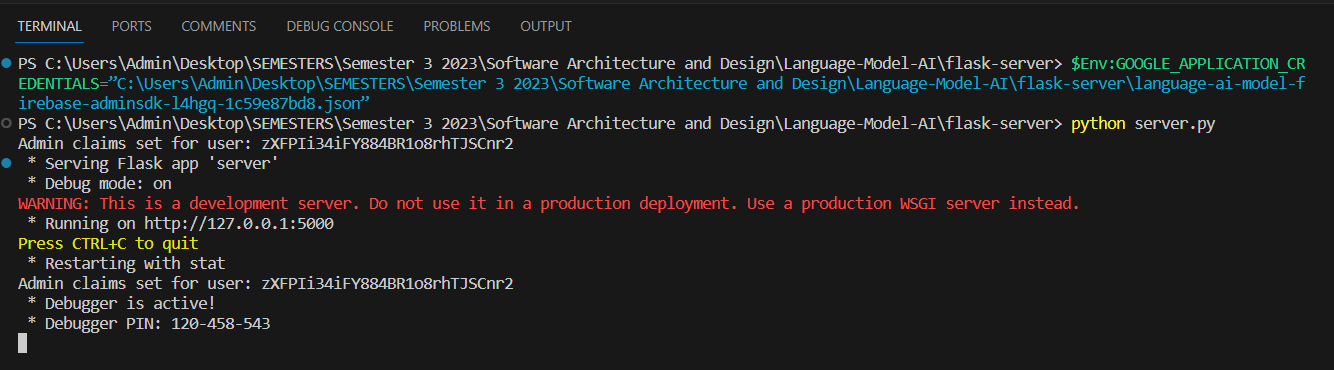
For running the **Backend,** things will require some little more steps:



* Here you will need to open the **flask-server** folder, and choose the file name **language-ai-model-firebase-adminsdk-l4hgq-1c59e87bd8.json**
* Click on **Copy Path**
* Copy this **important line**: $Env:GOOGLE\_APPLICATION\_CREDENTIALS=””
* Paste the path that you have copied earlier into the “” bracket of the **important line** above.



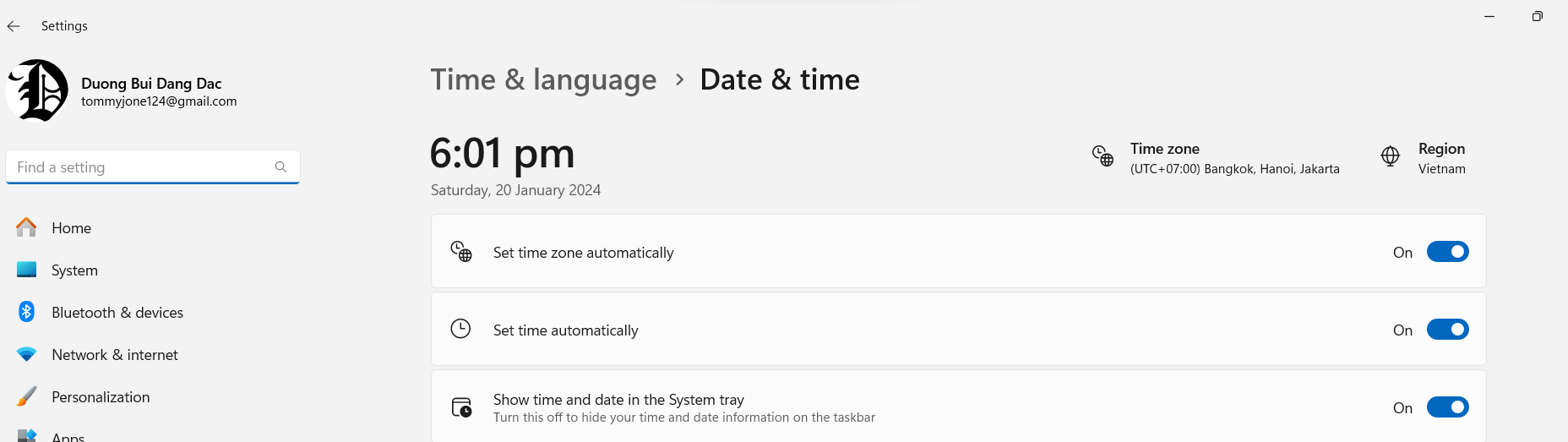
* Press **enter**, then you should see that it not display anything on the console, which mean you are on the right track. Please notice that on this above picture, my file path in the blue text are having a different path than yours, which could be vary on the location that you store our file on your PC !
* After this step, you can then run: **python server.py** or **py server.py.** You then should see the following picture:



This mean that our **backend** has been successfully running !

* At this point, everything has been set! You can then view and test our product, please see the next part of our report: **Features.** In order to see and test all of the features we have!
* **Important note:** You might sometime not able to be login, due to this error showing on the backend:

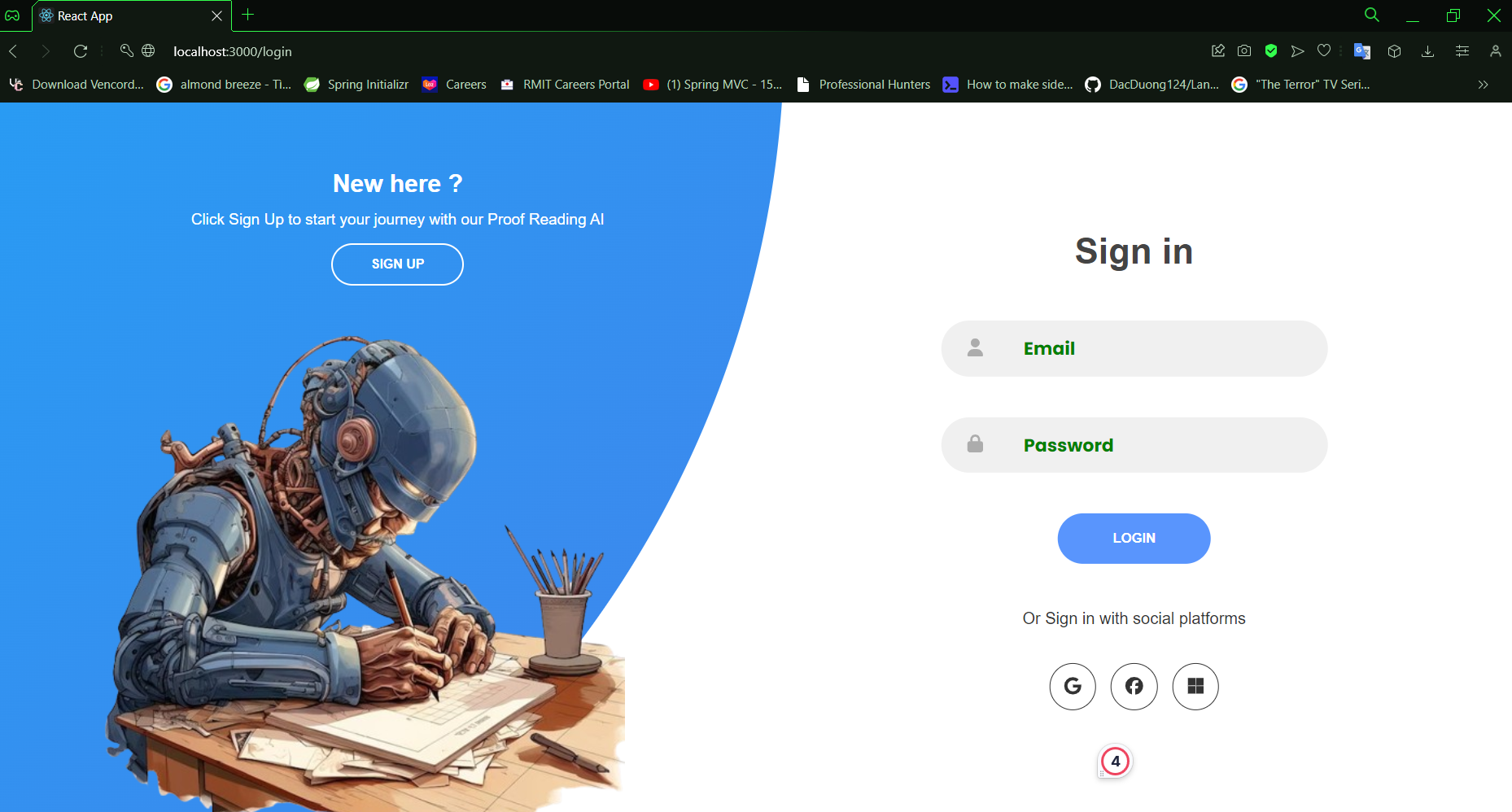


This mean that the PC clock on your computer could be a few second, or even a millisecond slower than the current **Firebase** server. Since our login feature require this token for security, it is advise to troubleshoot this problem as following:  
  


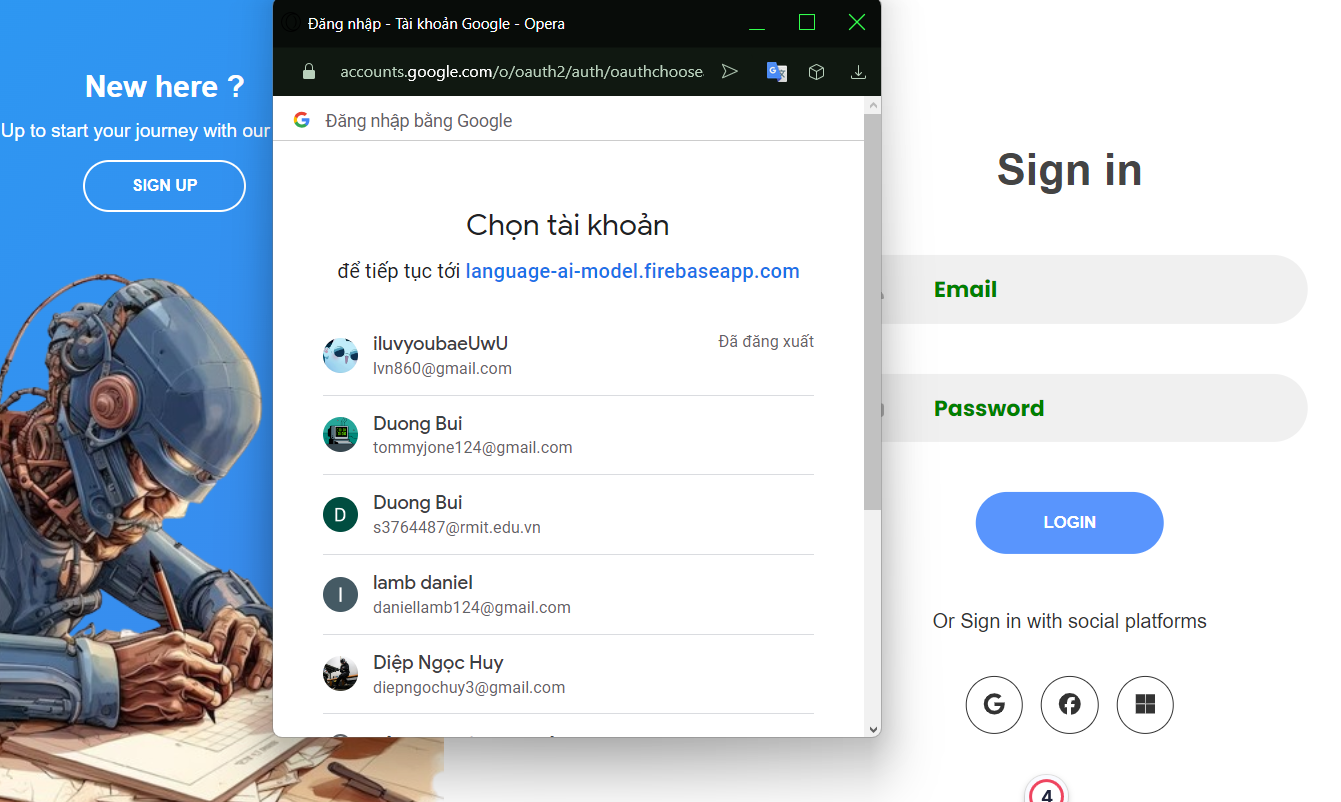
Go to your date and time section, to manually turn on and off **both of** the set time automatically. Then try our application again, after that, everything should be working fine!

1. **Features:**

* **Section 1:**  
  Create account, Login-logout, and OAuth2 Single Sign-in using Google, Facebook and Microsoft. Each of these login will provide user a secured **jwtToken** (Can be check on Application Local Storage section). When log-out, this token will be deleted or manually delete this token on Local Storage, the user will be sign-out completely and cannot manually access any other endpoints without having to sign in again. Ensure security for user !
* Screenshots for details:

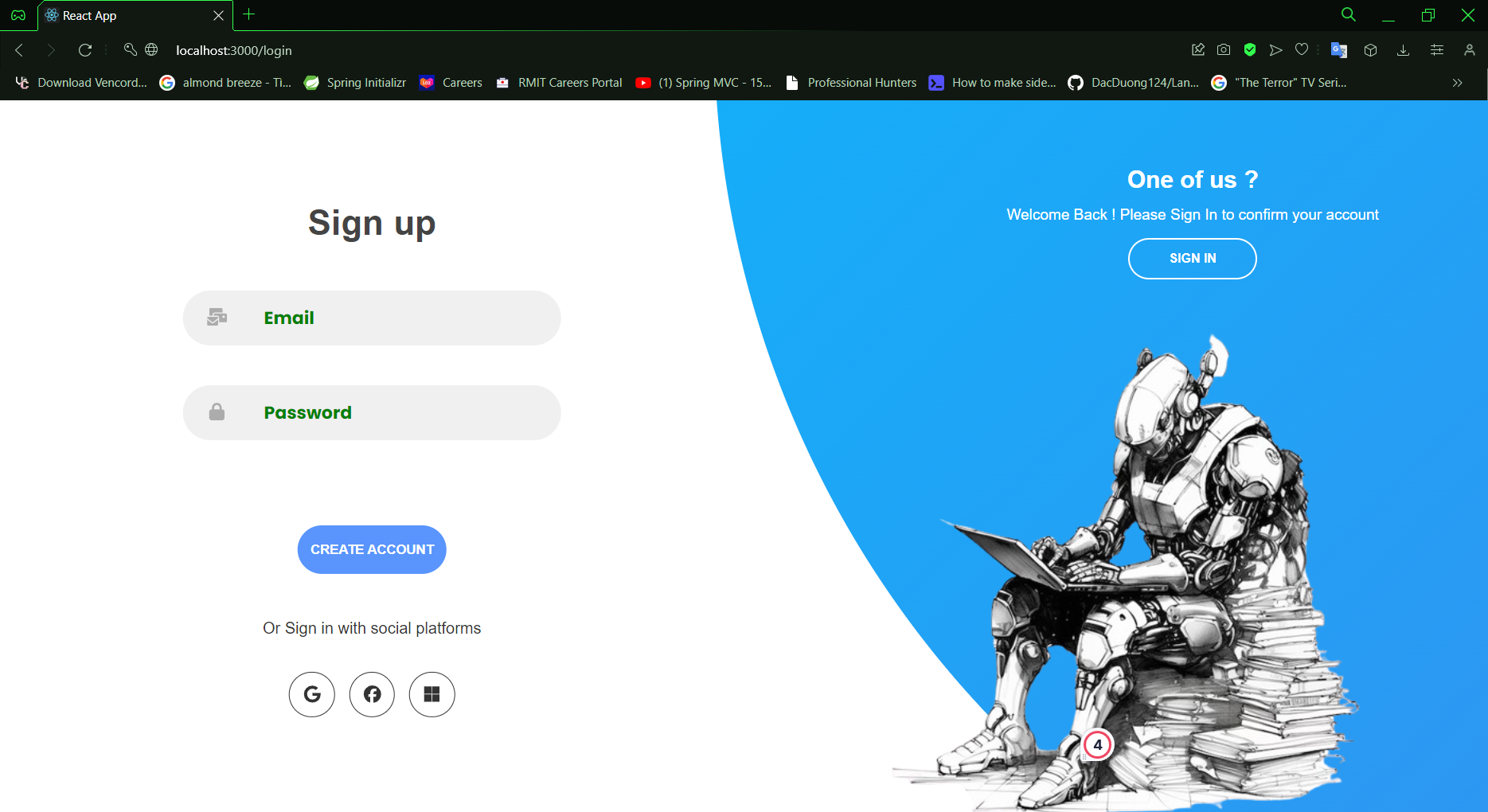


*Figure 1: Sign-in page*



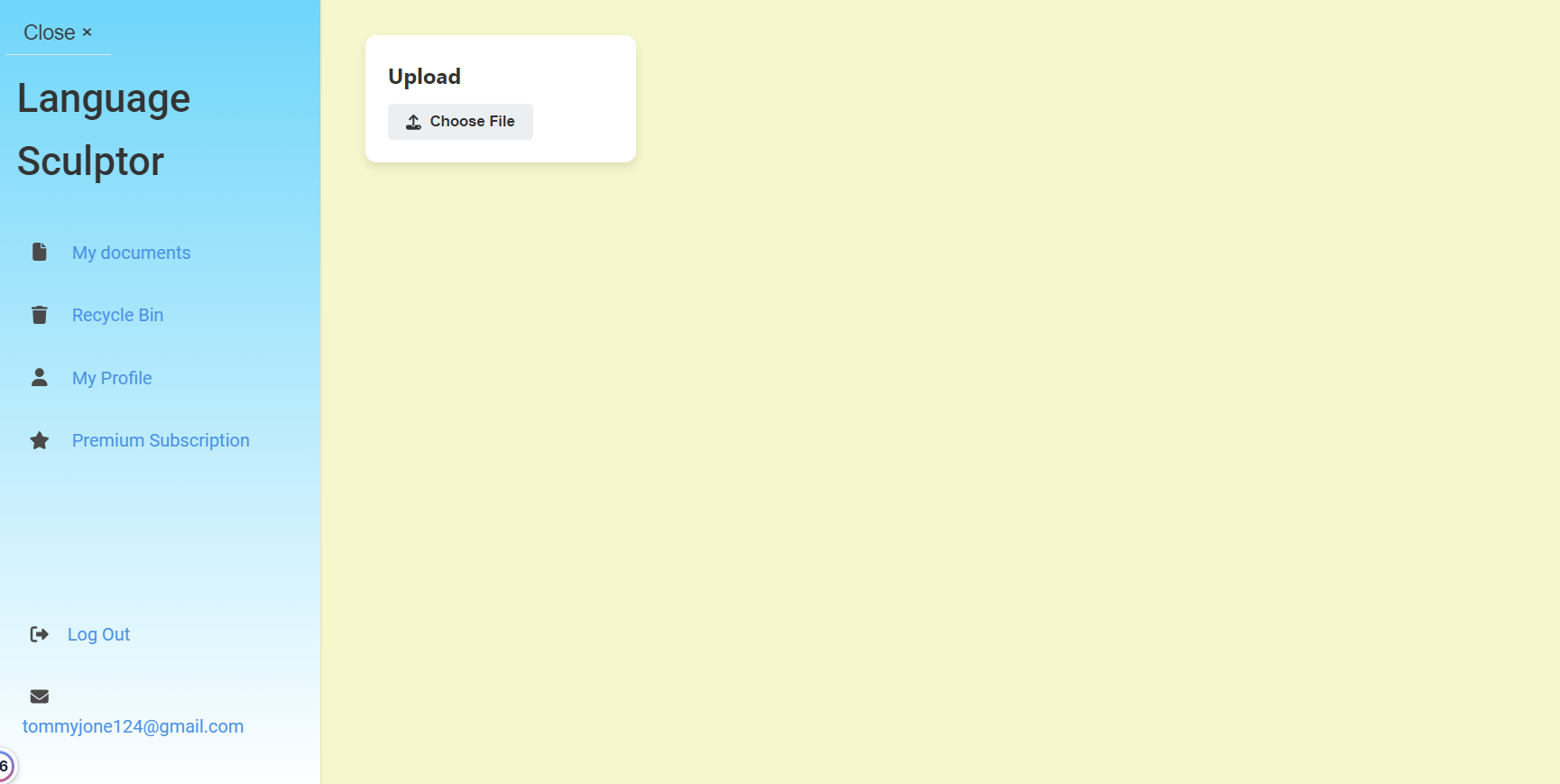
*Figure 2: Sign-in page (With Google example)*

* **Important note:** Please note that if you have sign-in **first** using Google, be sure that you should use Google sign in for the rest of your journey, as our app encounter an issue on Firebase authentication that if a user sign-in using the same email with other platforms (Such as Facebook and Microsoft). It will encounter an error that the email has already been used for this platform. So be sure to stay consistence of using 1 platform to sign in only. Don’t use other platforms that has the **same** email as the platform that you sign in at **first** time.

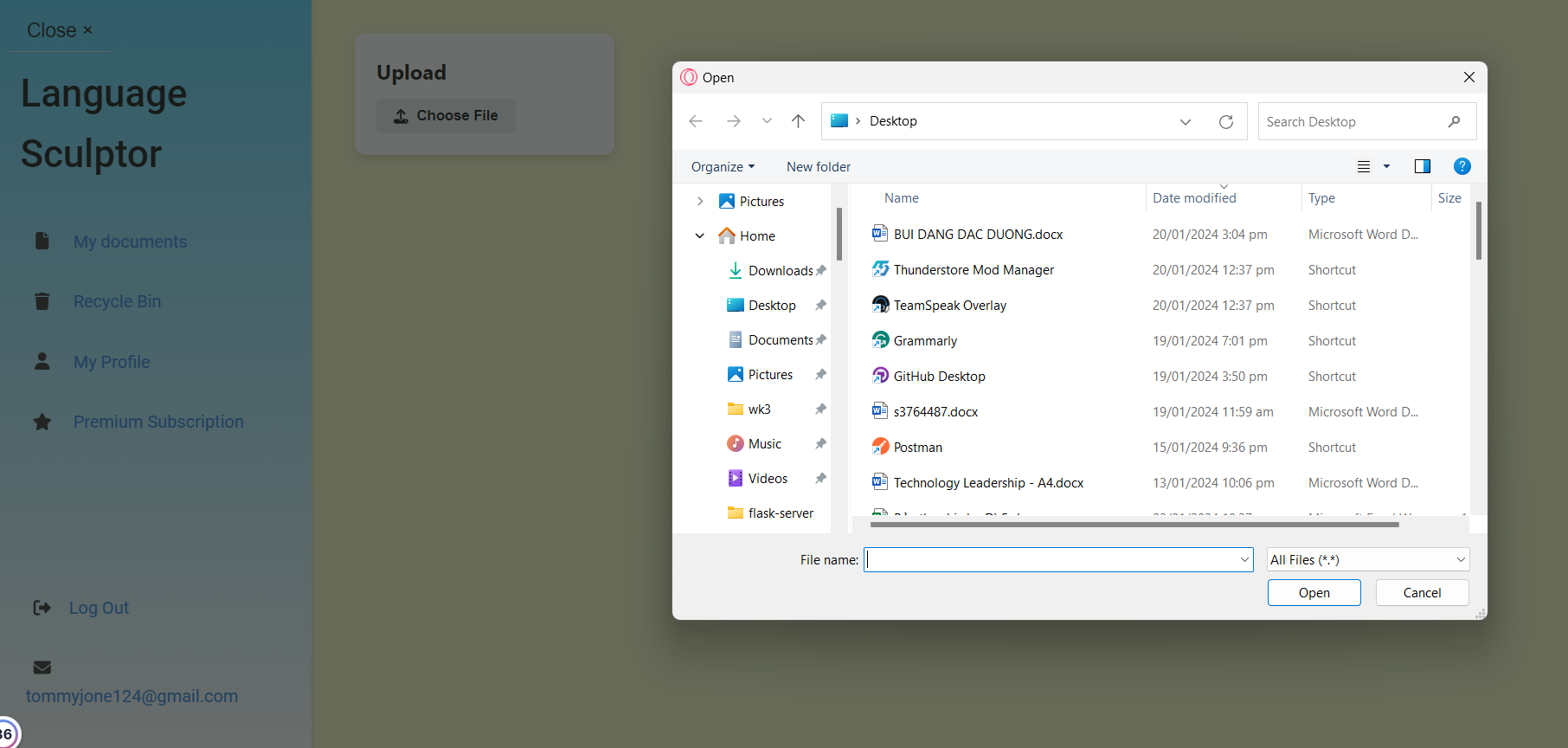


*Figure 3: Sign-up page*

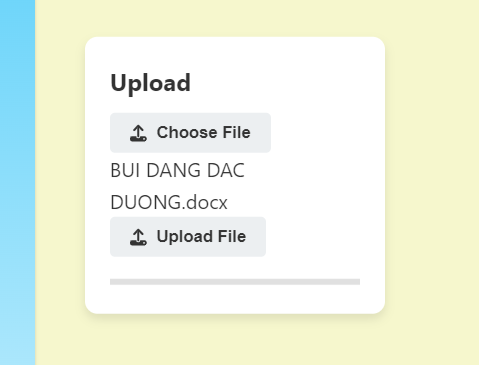
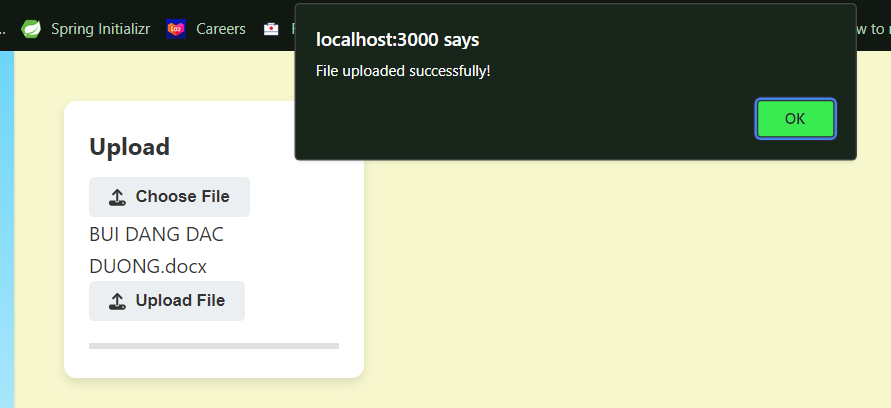
* Here user can sign up either using their own email, or also Sign-up using other platforms (Which in this case is also the same with sign-in using other platforms). Create your account, and move back to the sign in screen to continue !
* **Section 2:**
* The user should be able to signed-in to the home page, with sidebar to navigate to other section of the website. Here they can upload their files, correct their documents, download those corrected documents, move their documents to Recycle Bin or Delete them permanently. They can also view their profile information, and Upgrade their profile using Paypal payment feature.
* Screenshots:



*Figure 4: User Home Page (My Document section)*

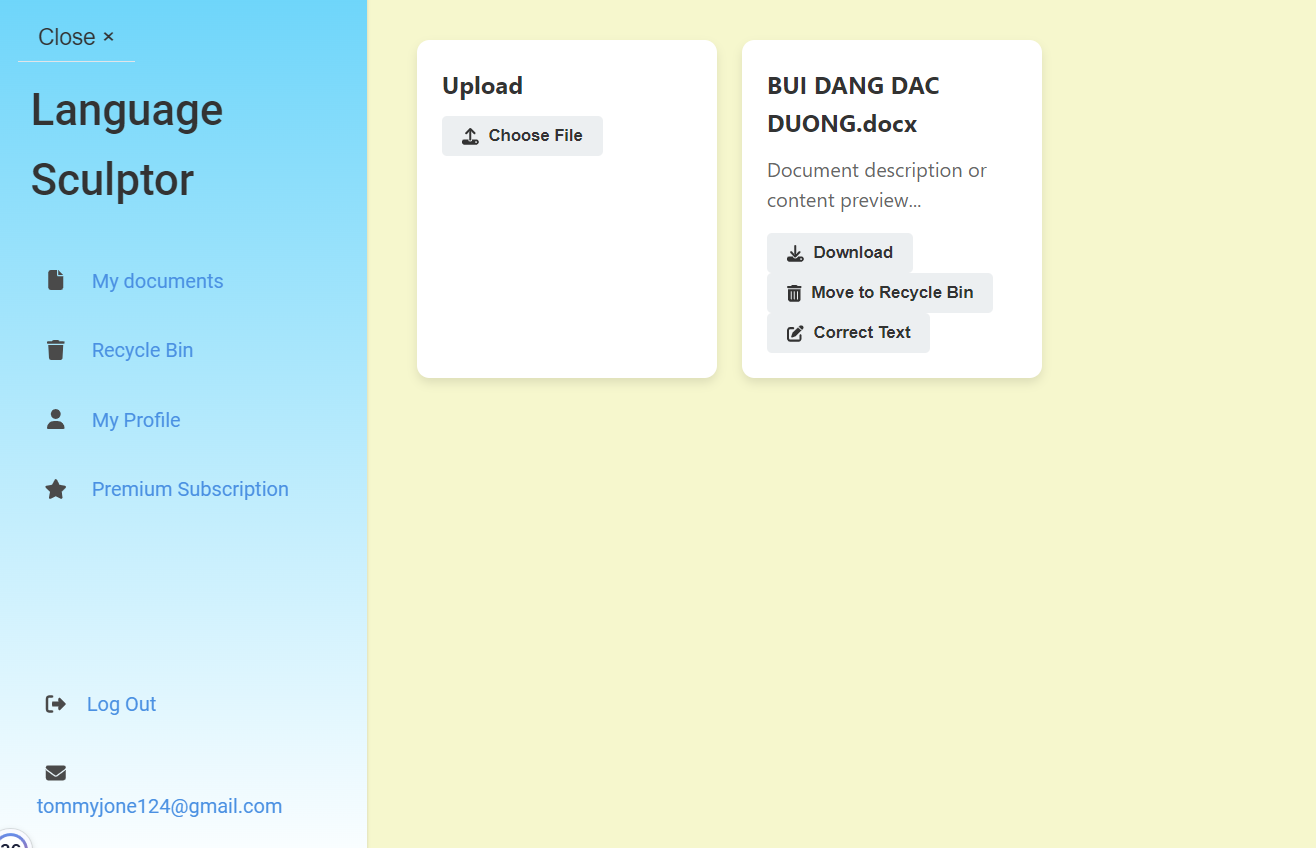
**

*Figure 5: Click the “Choose File” button to choose a file to upload*

* *

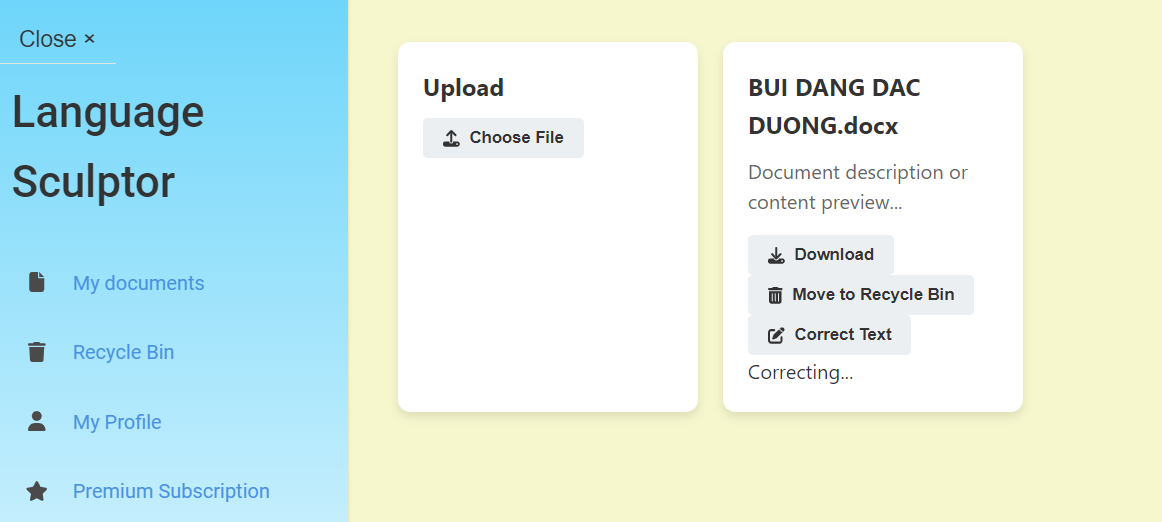
*Figure 6 + 7: Press the “Upload File” button and then press OK*

* **Important Note:** The user needs to refresh the page to see the uploaded file.

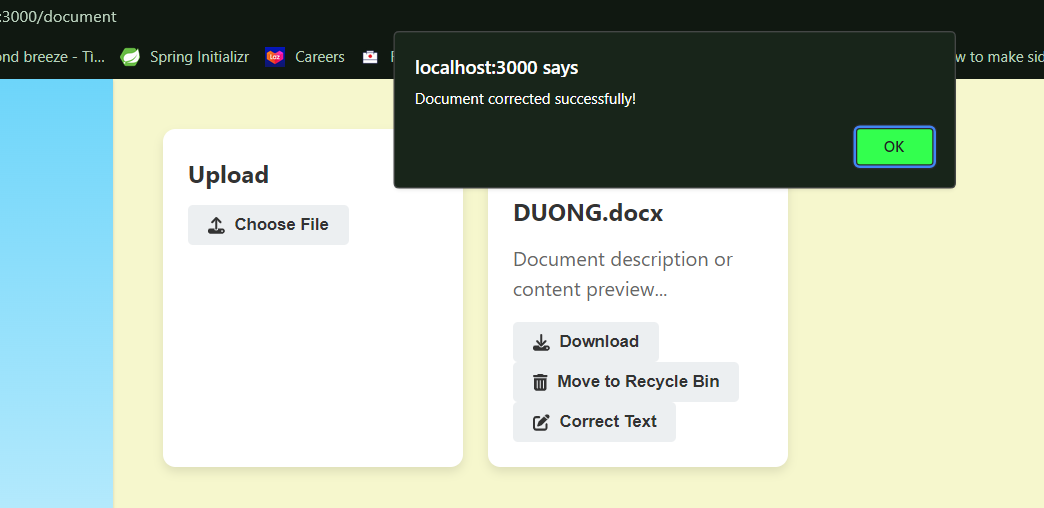
**

*Figure 8: Here user can see the file get uploaded (Remember to refresh the page)*

* At this point, the user can either choose to Download the file again to their computer, Move this file to Recycle Bin, or using one of our core feature, The “Correct Text” button.

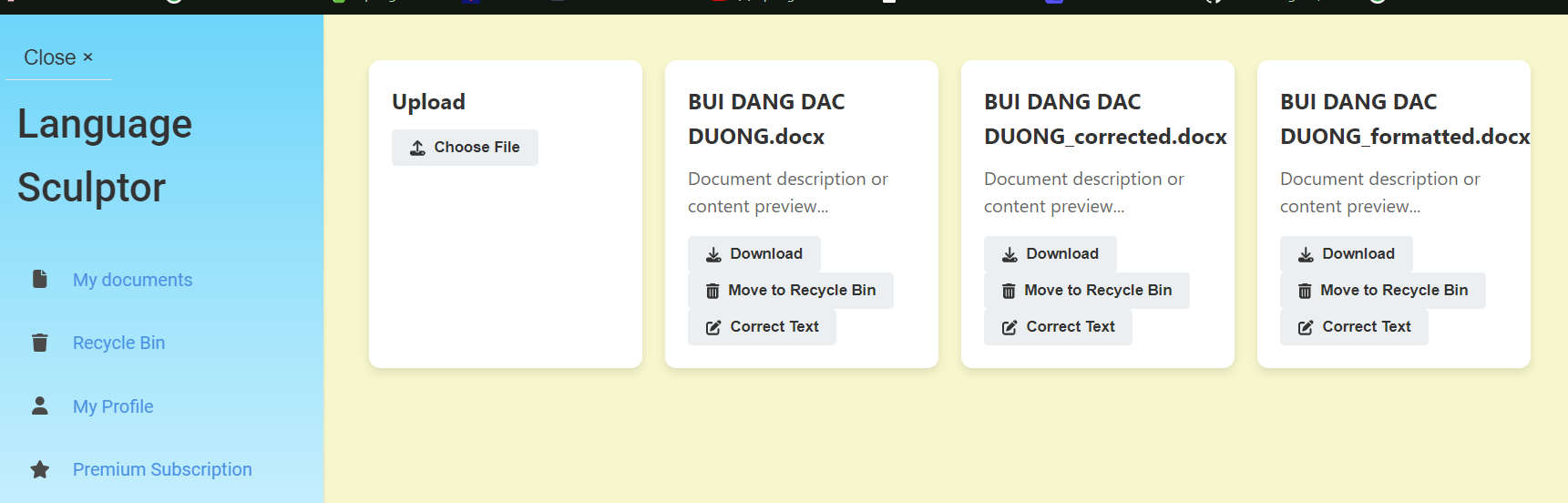


*Figure 9: Here user can see the file getting “Correcting” after choose “Correct Text”*



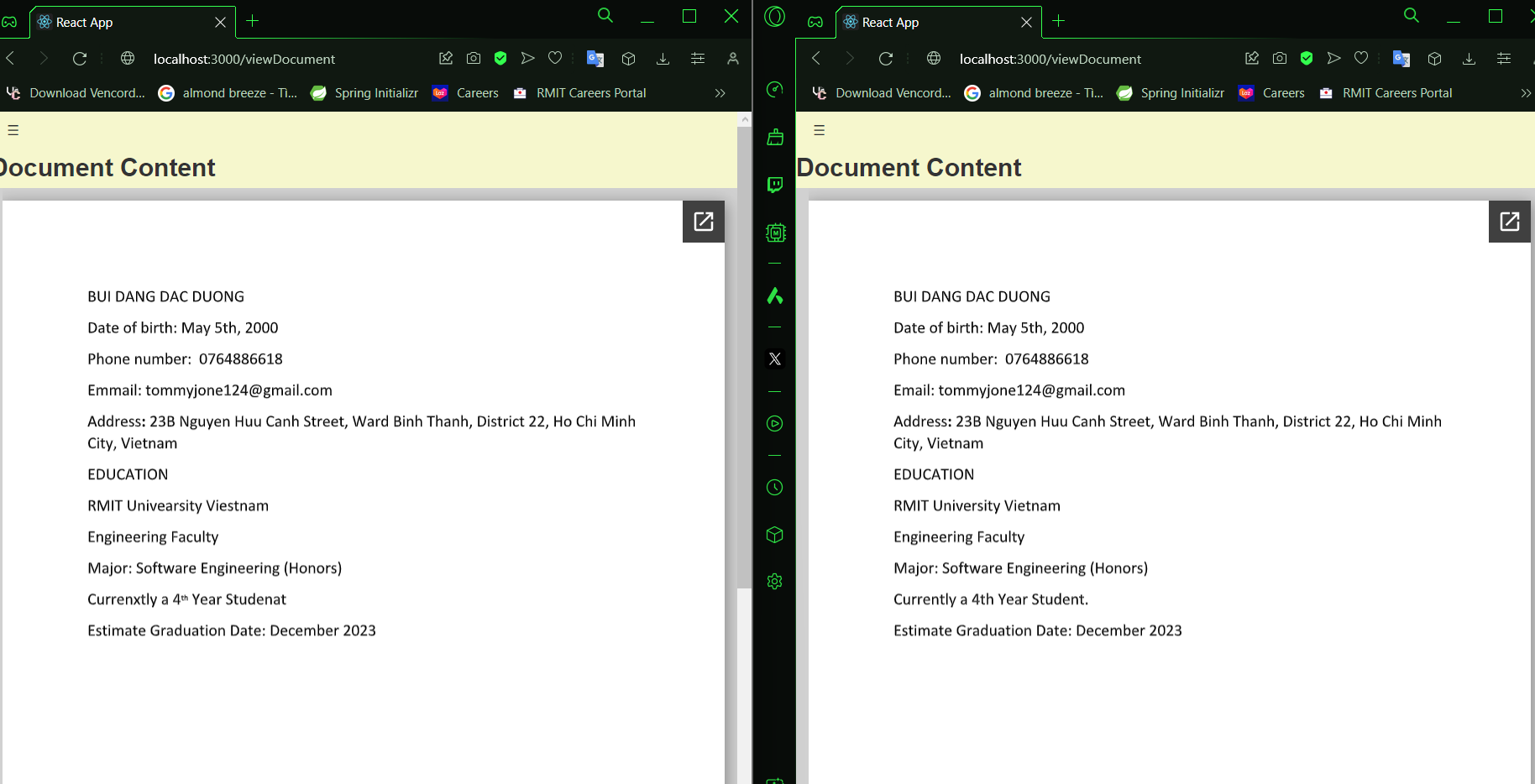
*Figure 10: The document has been corrected successfully! Press the Ok button to continue*

* **Important note:** This process of correcting can be dependent on how large the document file is, the larger and many more pages it has, the longer the process will take. In our testing, a 30 pages document might take up to 15-20 minutes! Please remember to refresh the pages after pressing the Ok button to see the changes.



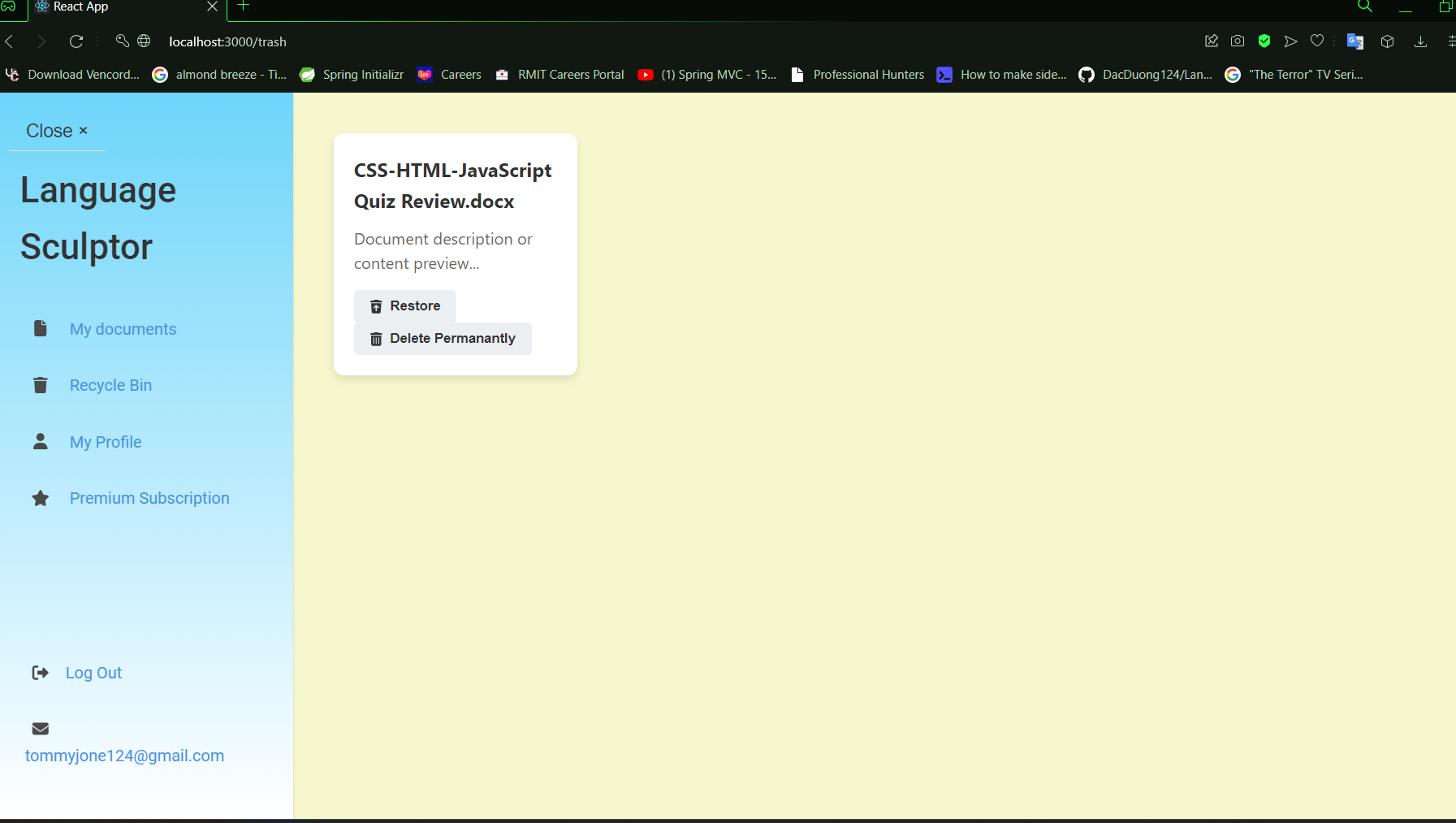
*Figure 11: Here we can see that after the correction, it will give us 2 extra files, corrected, and formatted*

* The difference between these two files are that the Corrected one are just corrected the text, but the Formatted one not only just corrected the text, but also formatted the document too.



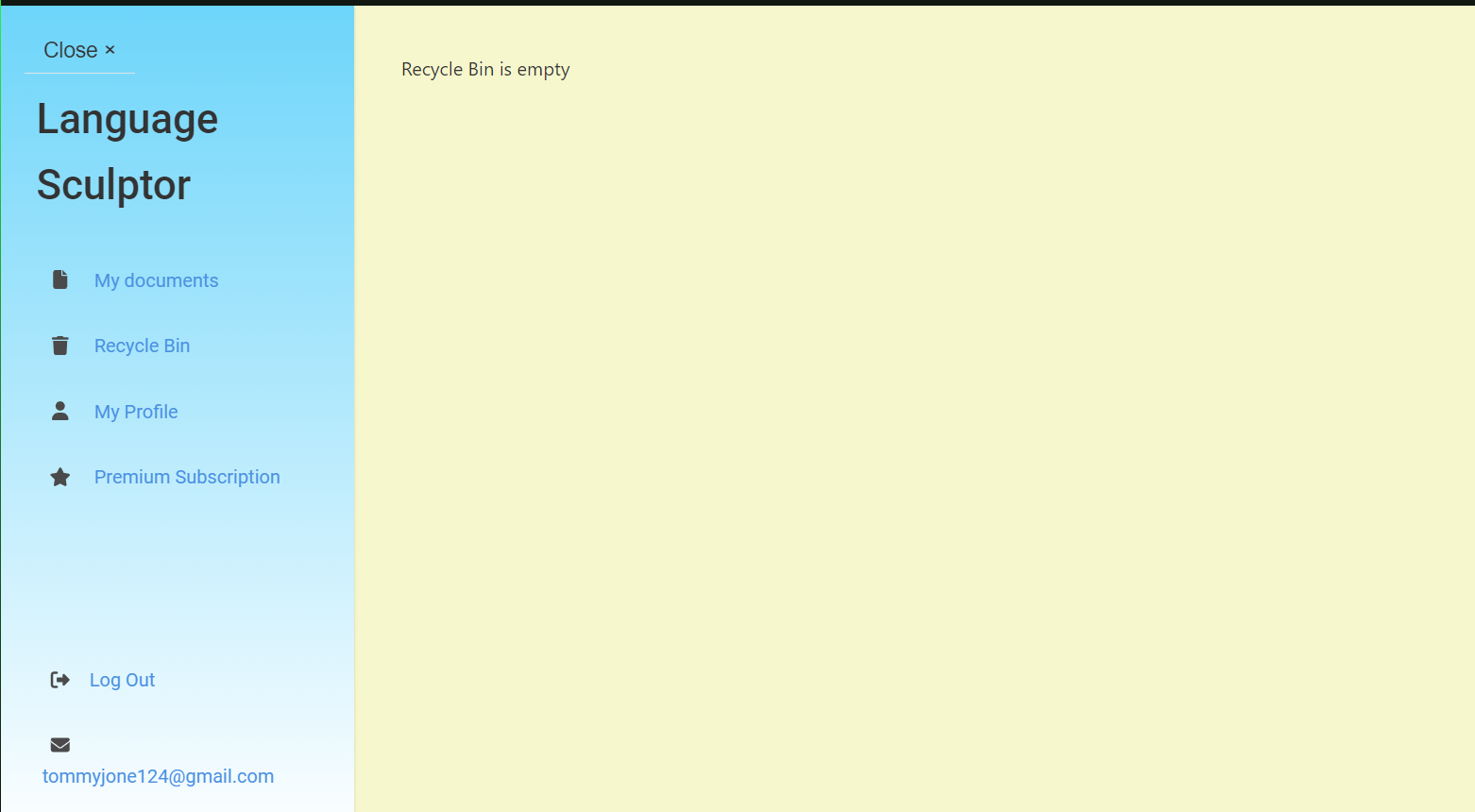
*Figure 12: Side by side comparision*

* **Important note:** If you want to directly view the document on the website, simply just click onto the document card, and it will move you into a separate page to view the document content. Please note that sometime the document might take time to be shown up, or it might need to be refresh the pages a few time in order for it to be shown up, which in this case our team consider this feature to be a little unstable at this time.

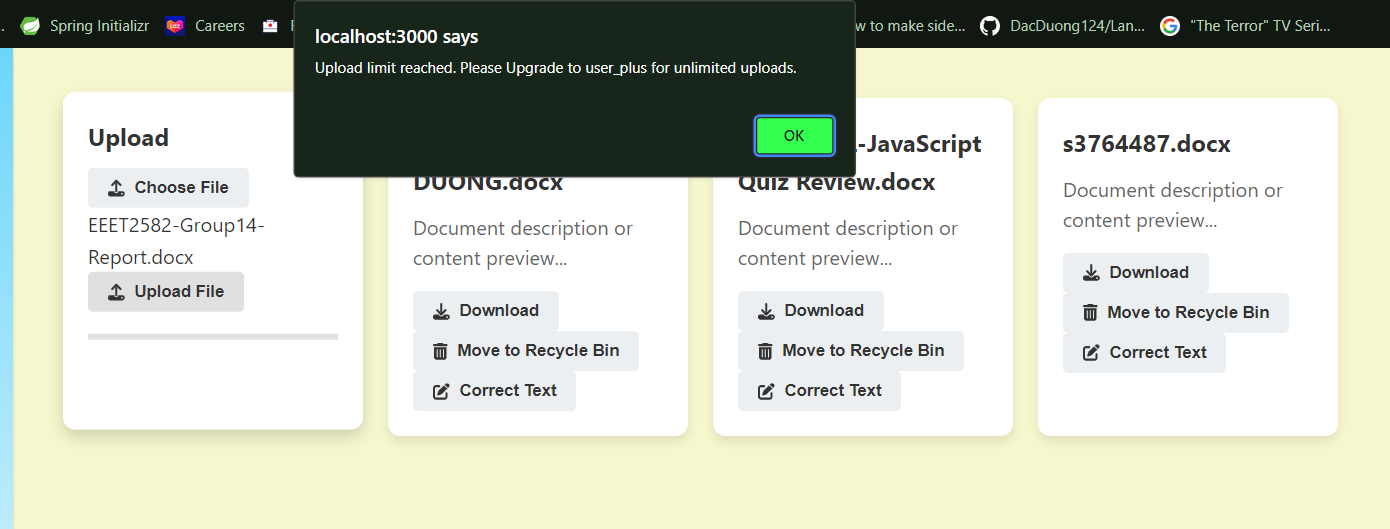


*Image 13: Recycle Bin section*

* Here in the Recycle Bin, the user can either choose to restore their file back to My Document section, or choose to permanently delete their files.

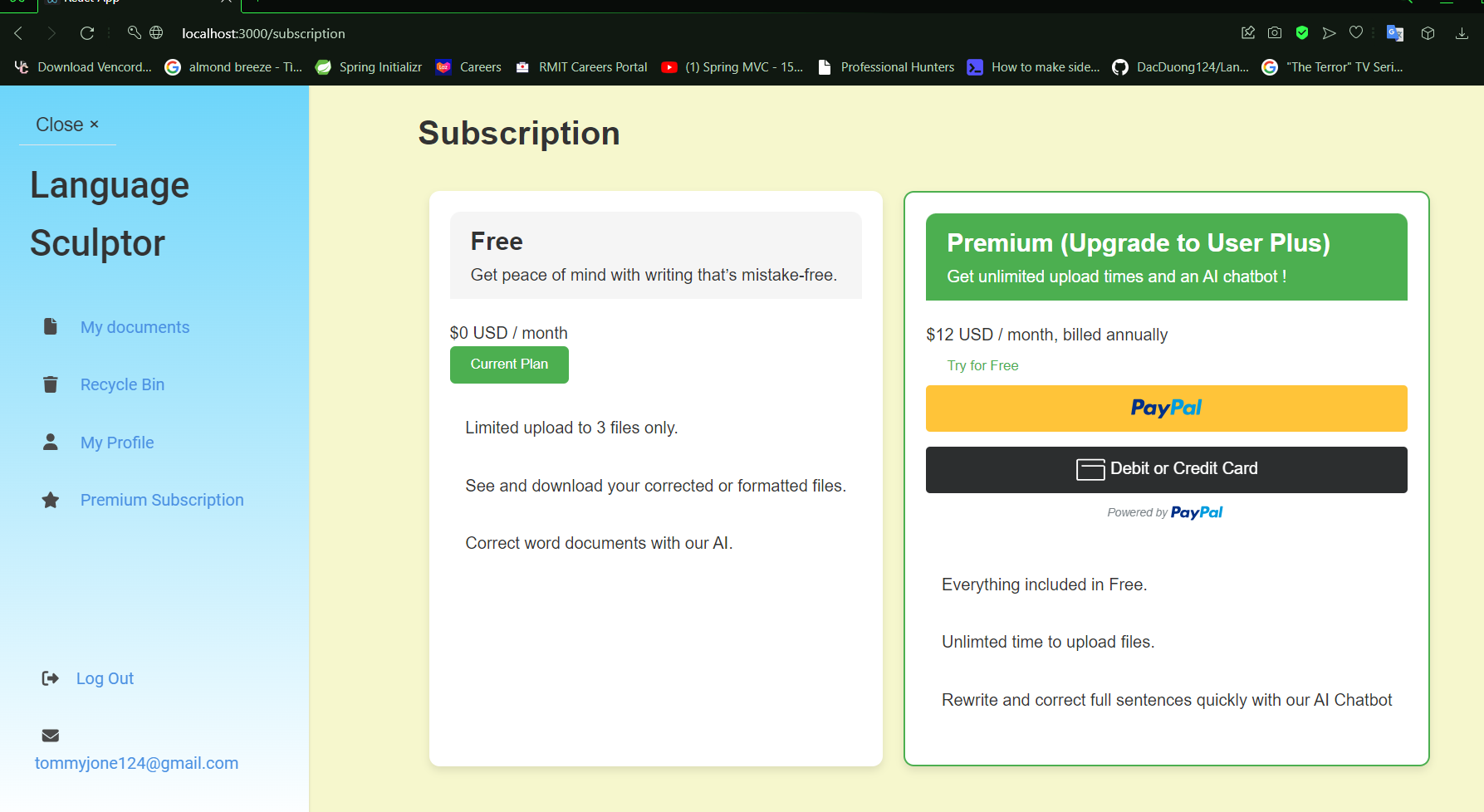


*Image 14: Recycle Bin section (Empty)*



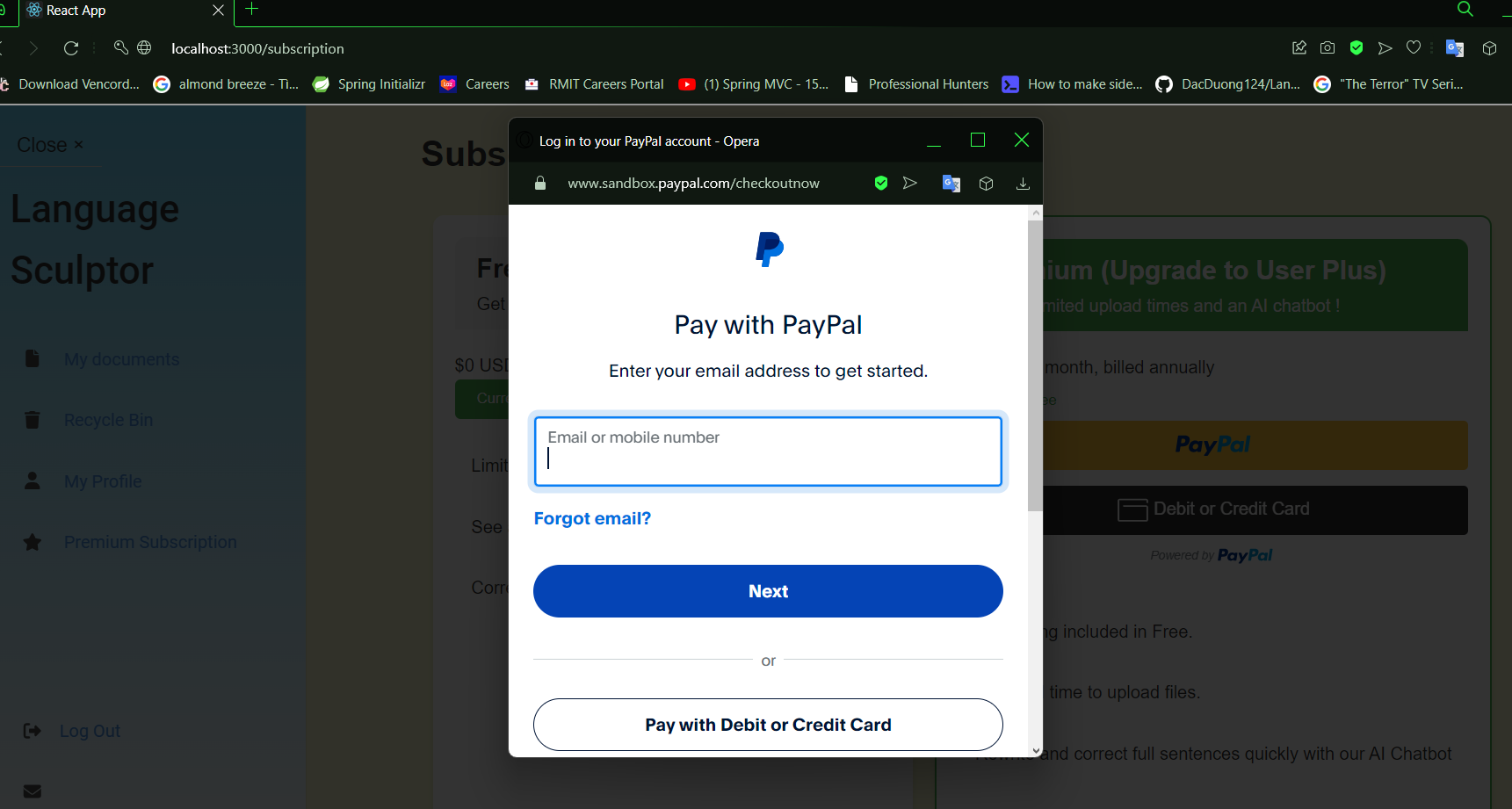
*Figure 15: Normal free User file uploading limitation*

* For normal user that using our application without paying, we only allow them to upload 3 files at a time. So if the user wanted to upload infinitely, they will need to pay 12$ to do so.

****

*Figure 16: Premium Subscription Section*

* In this section, the user can choose to upgrade their account into Premium. By either using Paypal account, or directly paying with their credit card.

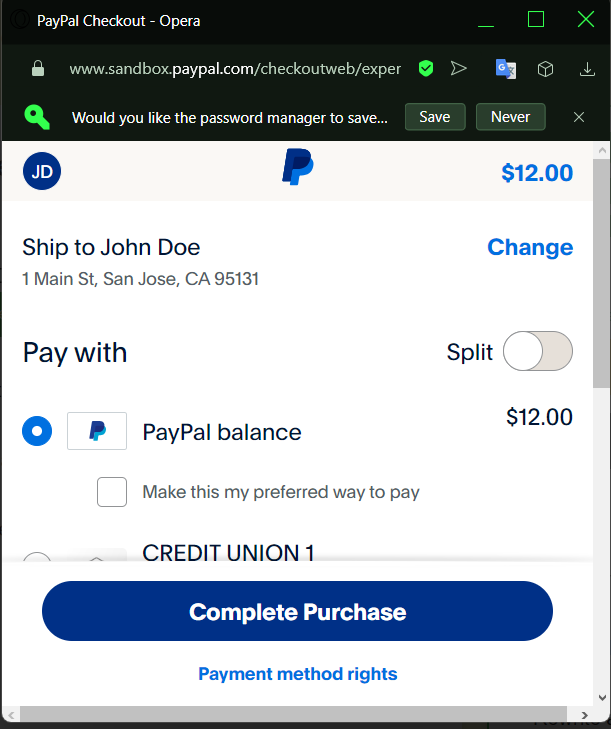


*Figure 17: Premium Subscription Section (Paypal)*

* Here the user will enter their paypal account. Note that this is only a testing product, so we only using sandbox account, so in order to access into this feature, you have to enter the following information:

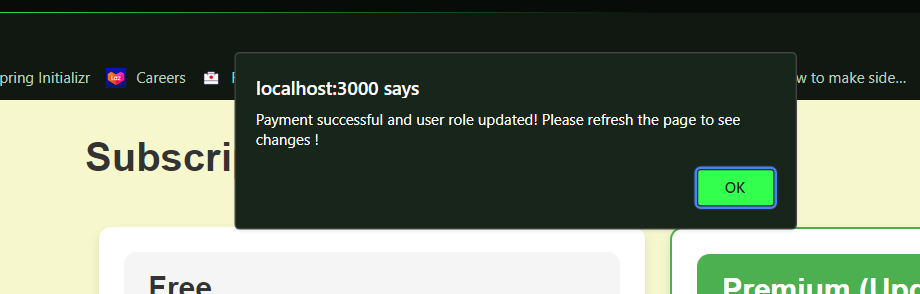
**Email:** sb-swahw27326949@personal.example.com

**Password:** FL7$6q-y

****

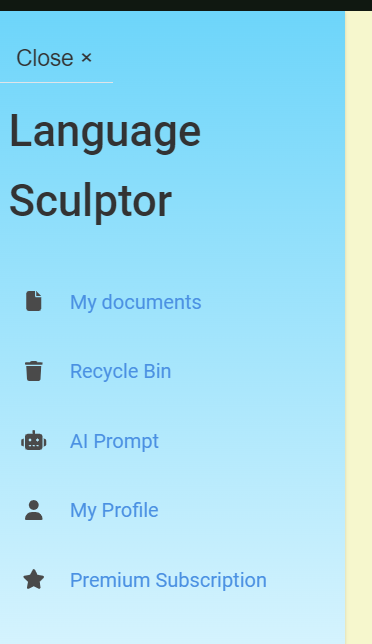
*Figure 18: Premium Subscription Section (Paypal)*

* After being able to access into the account provided above, simply press the Complete Purchase button.



*Figure 19: Premium Subscription Section (Payment Successful Alert)*

* Here after purchase, you will receive a notification said that the payment was successful and the user role has been updated !

****

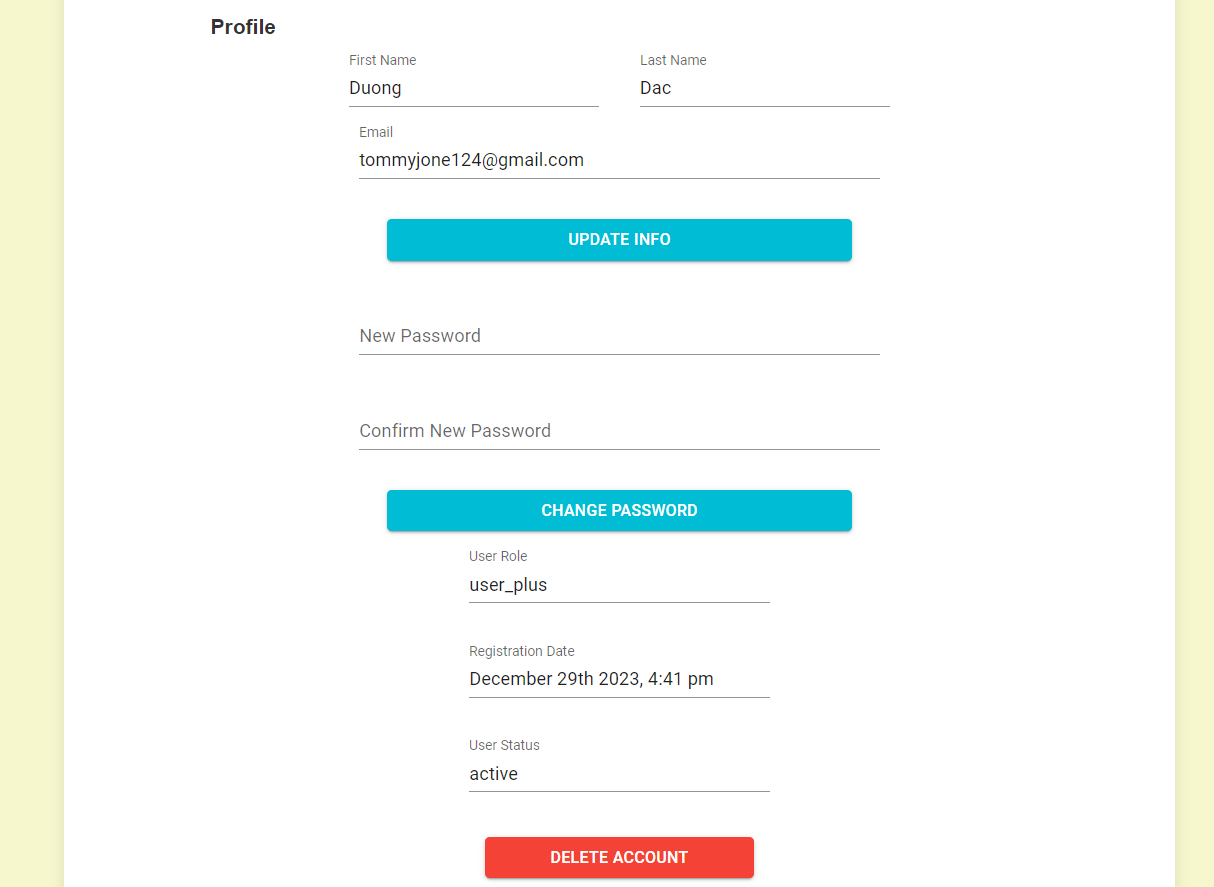
*Figure 20: Extra feature added into the sidebar after user role updated (AI Prompt added)*

* The sidebar will then automatically update a new section for user to enter, which is an AI Prompt to enter sentences or text that you want to correct.

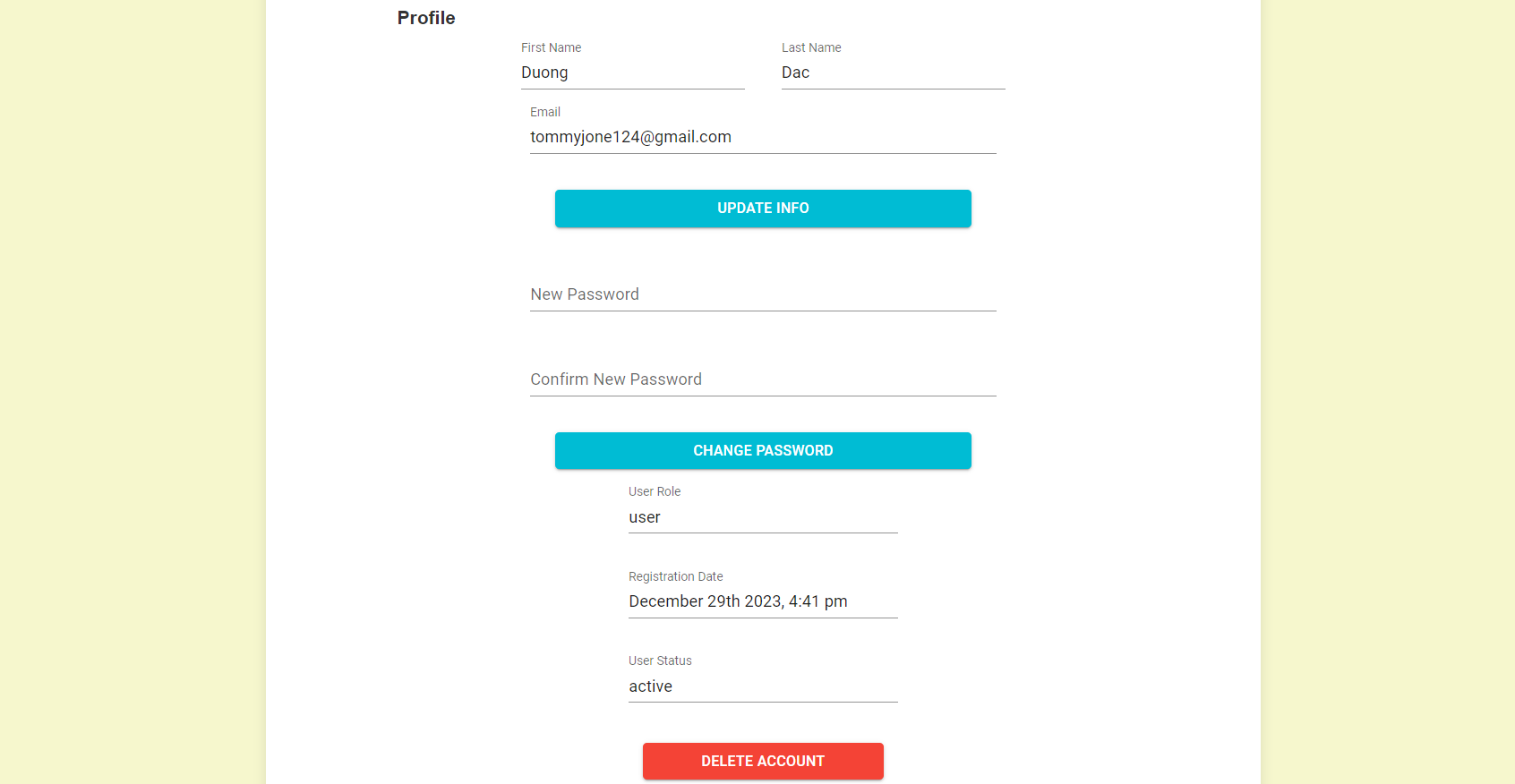
**

*Figure 21: AI Prompt Section*

* As you can see in the picture, after entering a sentence I want to correct, and press Generate Response, the AI will give me the corrected sentence.

****

*Figure 22: User Profile Section (Updated user role to: user\_plus)*

****

*Figure 23: User Profile Section (User Role: user) This is before upgrading their account*

* Here you can see on the above Figure 22 + 23, This is the user profile section, in here they can view their own user name, note that this user name is also corresponding with their OAuth2 account, such as Google, Facebook or Microsoft. Or if they sign up using the application Sign Up using tradition sign up method instead, their username will be empty and the user can update it via button Update Info.
* They can also change their own password too, however, they can only change their own password if they using traditional account, not by using OAuth2 account.
* The last 3 section is the Role of the User, which can either be “user” or “user\_plus”, depending on whether they have updated their account or not. The Date section that the user registered their account on, and finally the status to see if their account are currently active or not.

That’s all for the features we have for our application ! Thank you for reading if you have made it this far !