Frontend is the user facing component of the application, which will allow people to interact with API, in out case new nodes to sign in.

asgiref

Django

django-cors-headers // this allow us to fix the cross origin request issue

djangorestframework //

djangorestframework-simplejwt //for handling jwt authentication request

PyJWT

pytz

sqlparse

psycopg2-binary //postgrep sql database

python-dotenv //for loading in some env. variables

CREATE DJANGO PROJECT

**django-admin startproject backend**



**JSON WEB TOKEN (JWT)**

Json web token will act as a the permissions or authentication everytime we access a website.

The idea is, we have a frontend and our frontend is separate from the backend. So, everytime we make a request to the backend, the backend needs to know who we are, & permissions to do. So, it will include a token along with our request to our backend. And that token can be decoded and understood to represent a certain set of permissions.

We have frontend and frontend is communicating with the backend. Its doing it through a request. It send a request to the backend and then it will receive a response from the backend to the frontend.

Once we pass our login credentials to the backedn throught the frontend, assuming our credentials are current, the back will send 2 tokens to the frontend.

1. Access token 2. Refresh token

Access token is once which we will use with all of our request

Refresh token we will use to refresh the access token



------------- -----

To grant an access token, we need to have a set of credentials.

Step1 : figure out a way to create a new user

(we can create a new user & then store their username & password, & then use their credentials to login and access the access token for that user.)

**Django use an ORM.**

ORM is object relational mapping.

It maps python objects to the corresponding code that needs to be executed to make a change in the database.

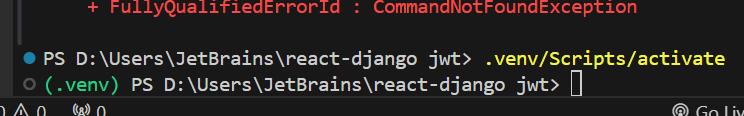
Serializer take a python object and convert it into json data that can be used in the communication with other applications.

Command to create a new environment

**python -m venv .venv**

Command to activate the python environment

**.venv/Scripts/activate**



Command to create Django backend project app

**django-admin startproject backend**

**python manage.py makemigrations**

* What makemigration will do, is, it will create the file that specifies the migrations that need to be perform
* So, now once u make the migrations u will actually migrate.

**python manage.py migrate**

to run the Django application

**python manage.py runserver**

**npm create vite@latest frontend -- --template react**

**npm install axios react-router-dom jwt-decode**

Once token is created

Now we will start writing the code for authenticated users

**Interceptor** :

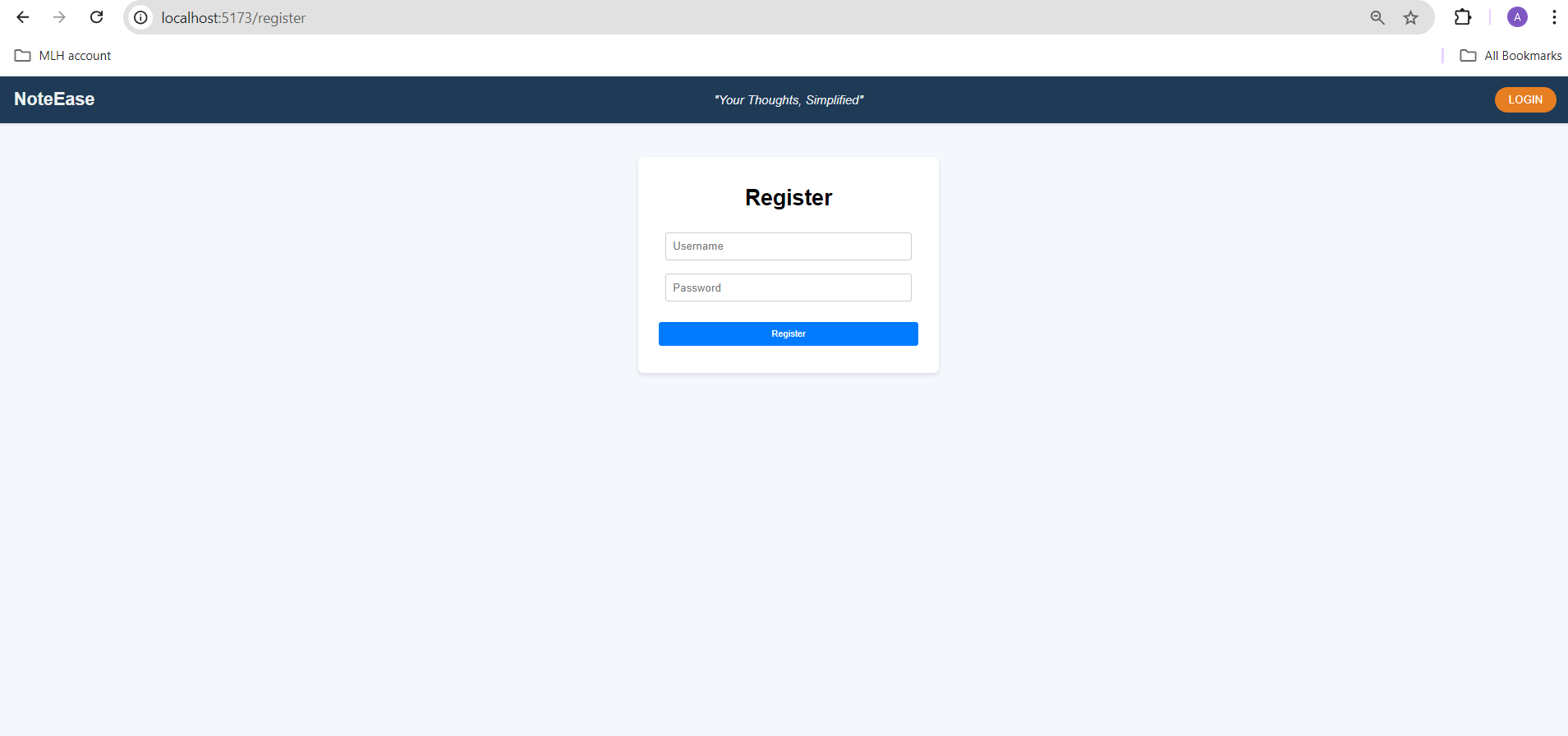
The interceptor will essentially intercept any request that we are gonna send and it will automatically add the correct header so that we don’t need to write it manually a bunch of times or repetitively in our code.

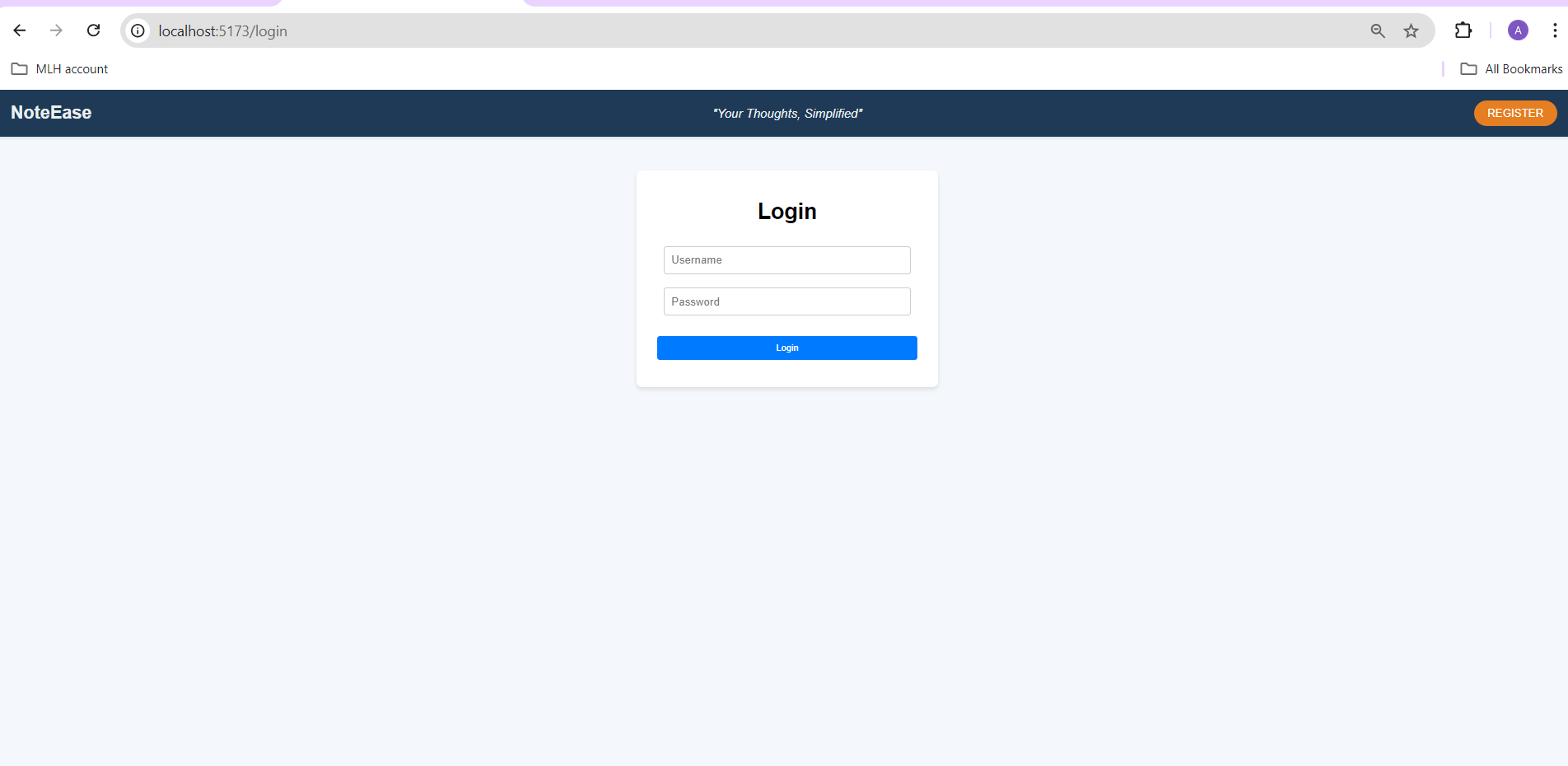
**Axios** is a very clear way to send a network request.

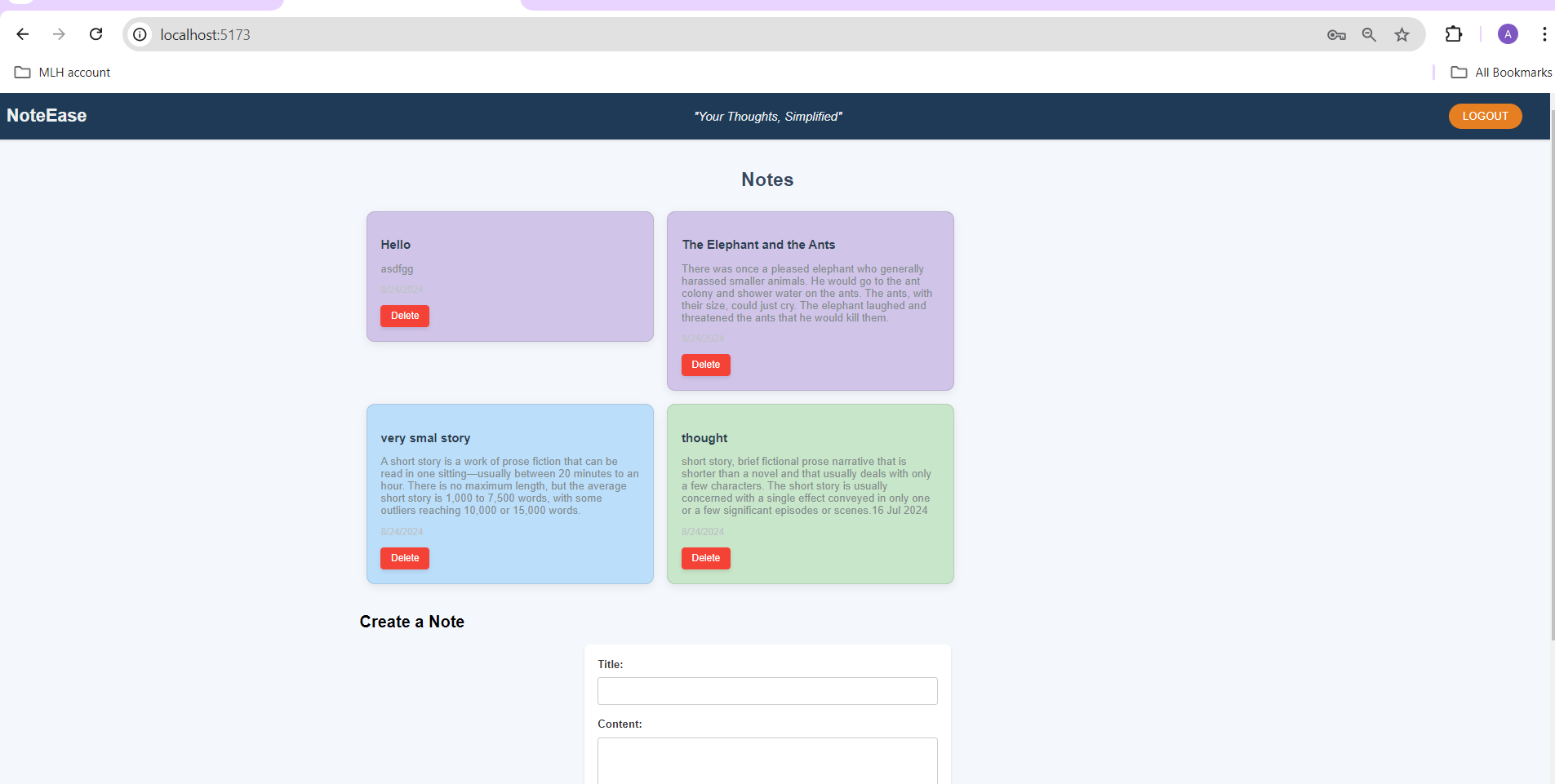
**So, we are going to setup axios interceptor so any time we**  send a request, its gonna check if we have a access token and if do, it will automatically add it to that request so that we don’t need to think about it, once we write this code.

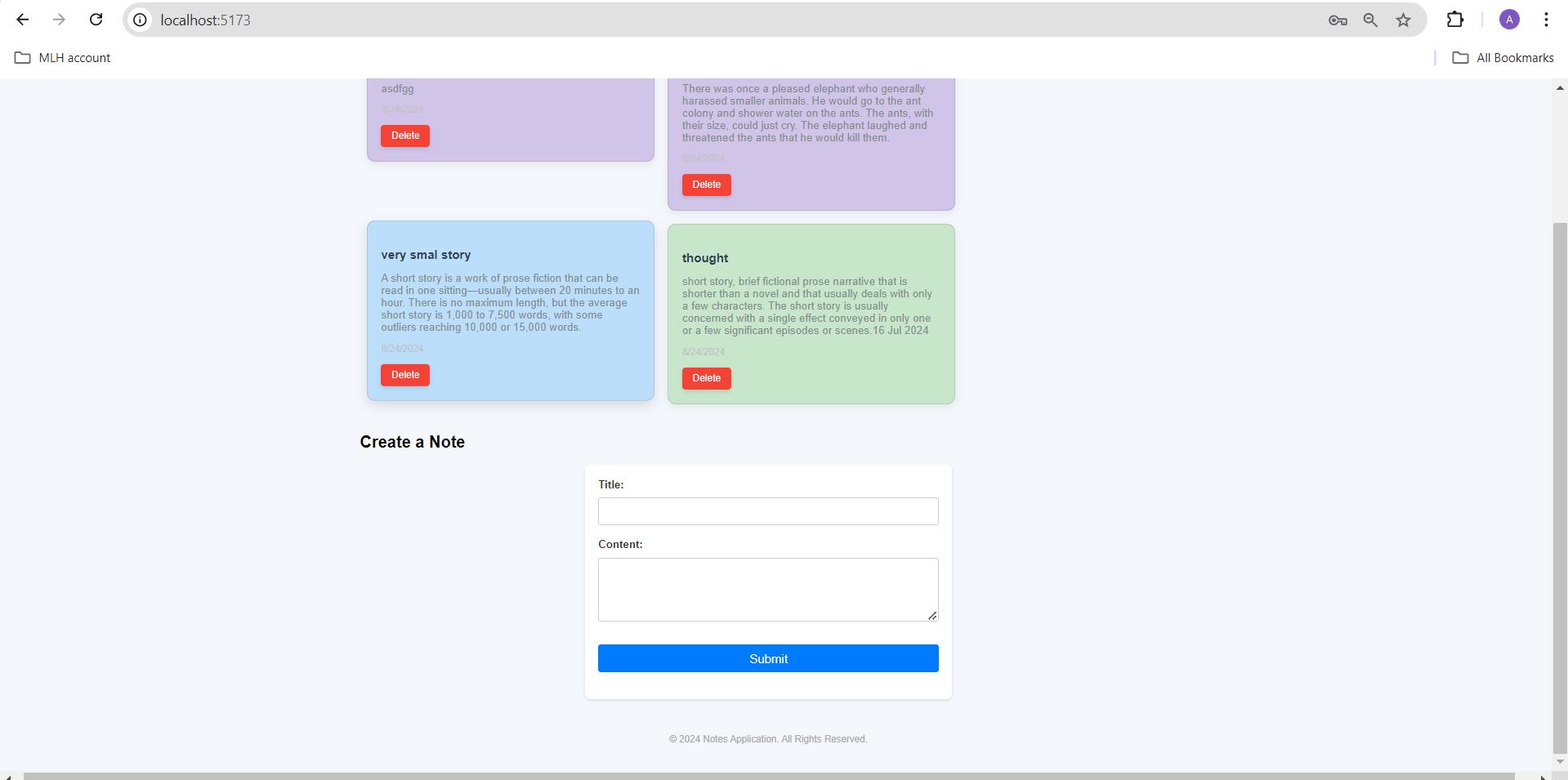
**ProtectedRoute.jsx**

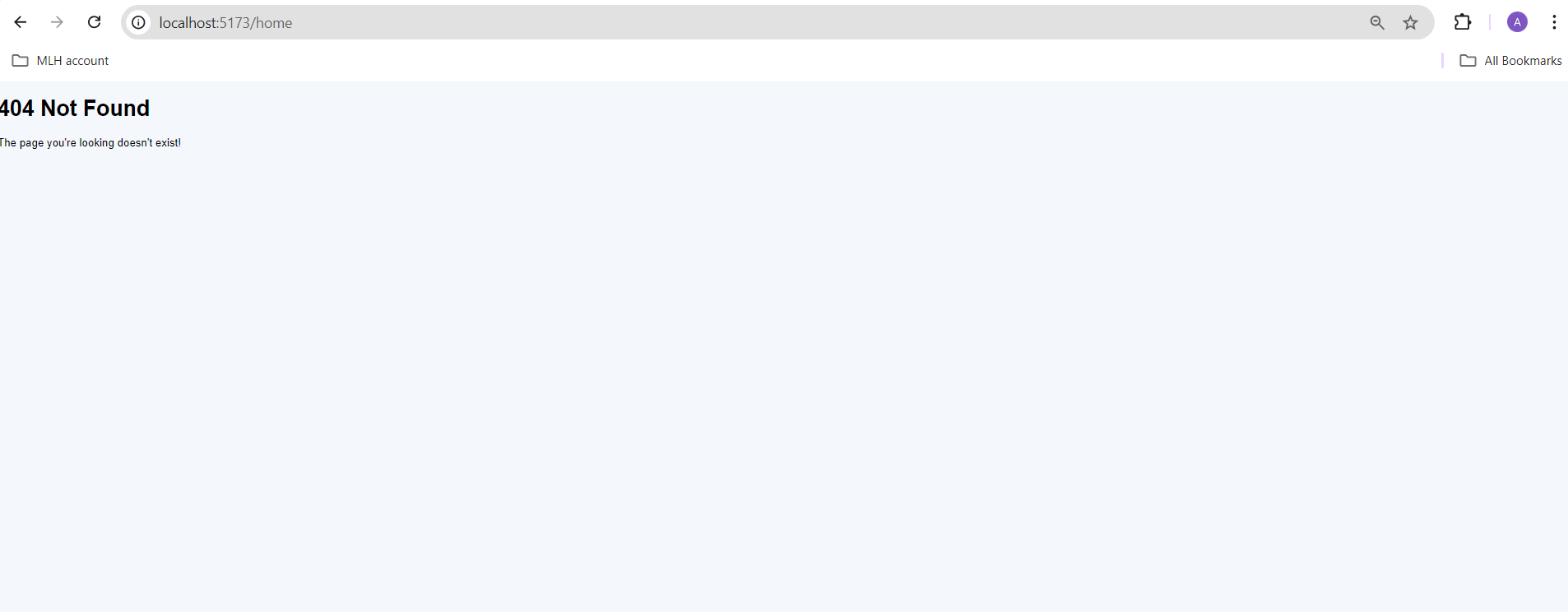
This is gonna act as a wrapper for a protected route.







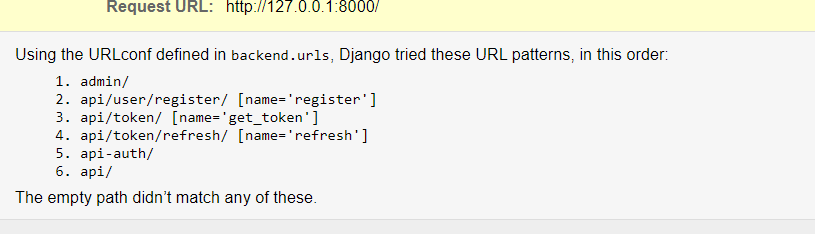




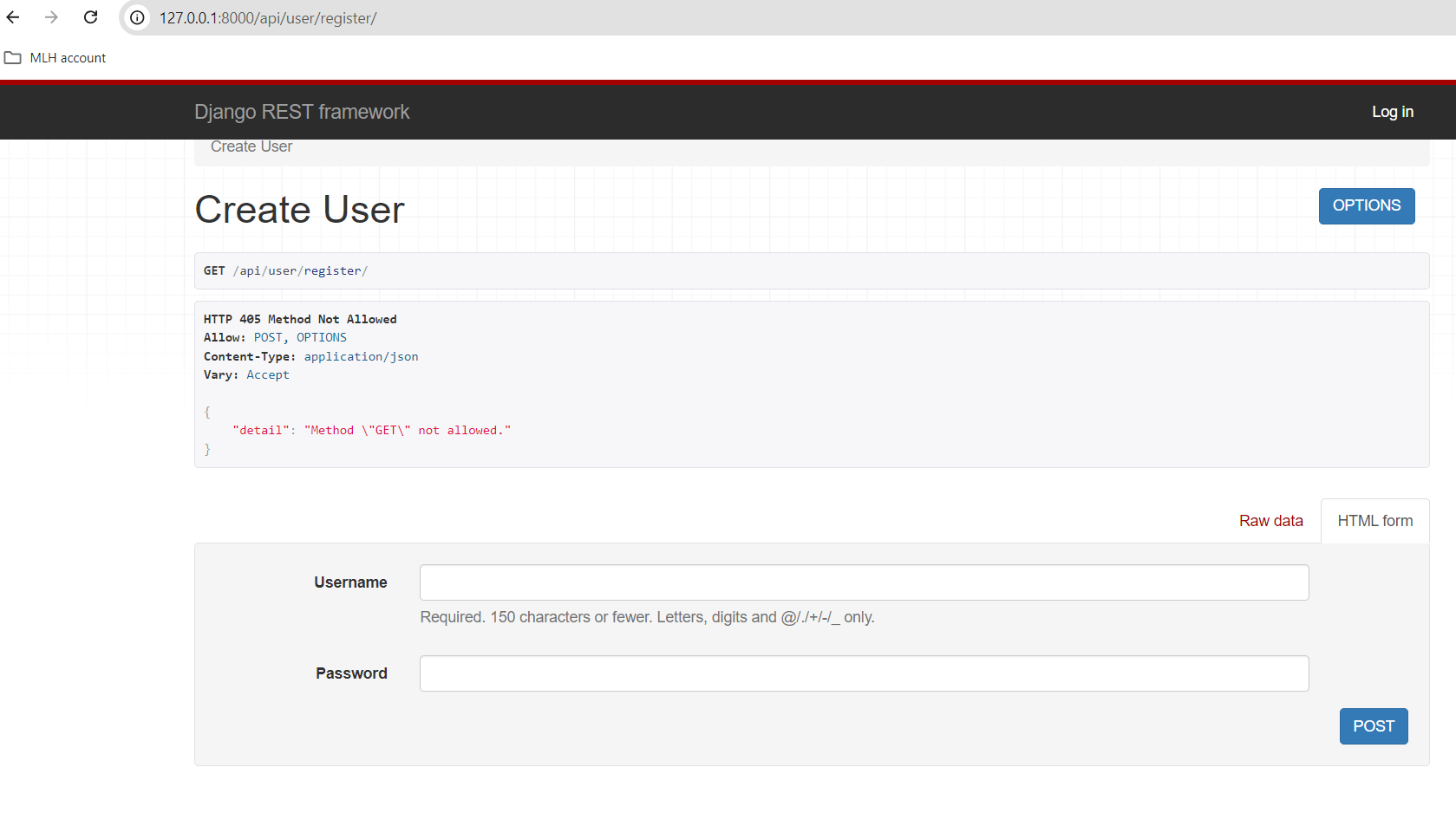
**Functionality**

1. Once the user registers (http://localhost:5173/register). A jwt “response token” \* “refresh token” is generated and is stored in the database.
2. When user login with the correct credentials the response token is used.

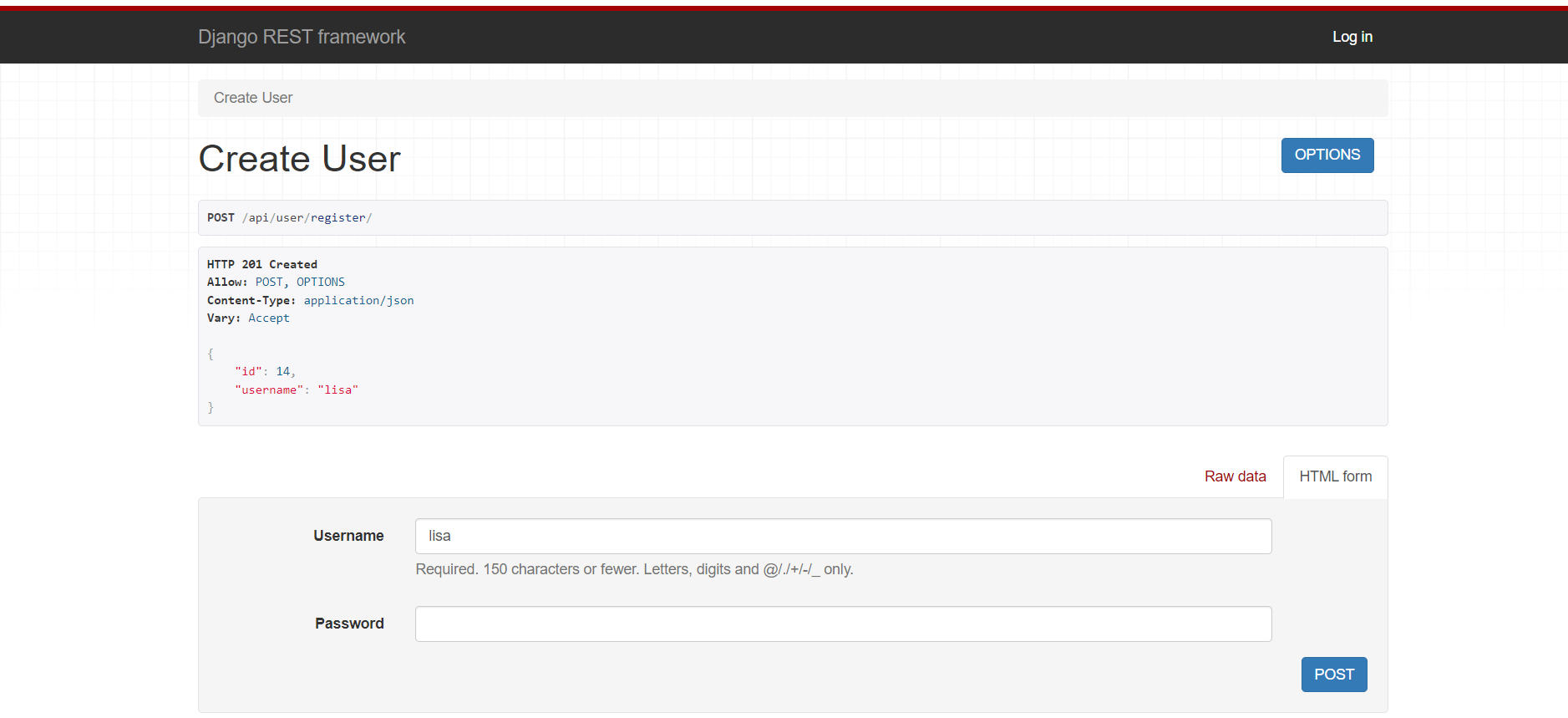
**Following are the urls in the Django**



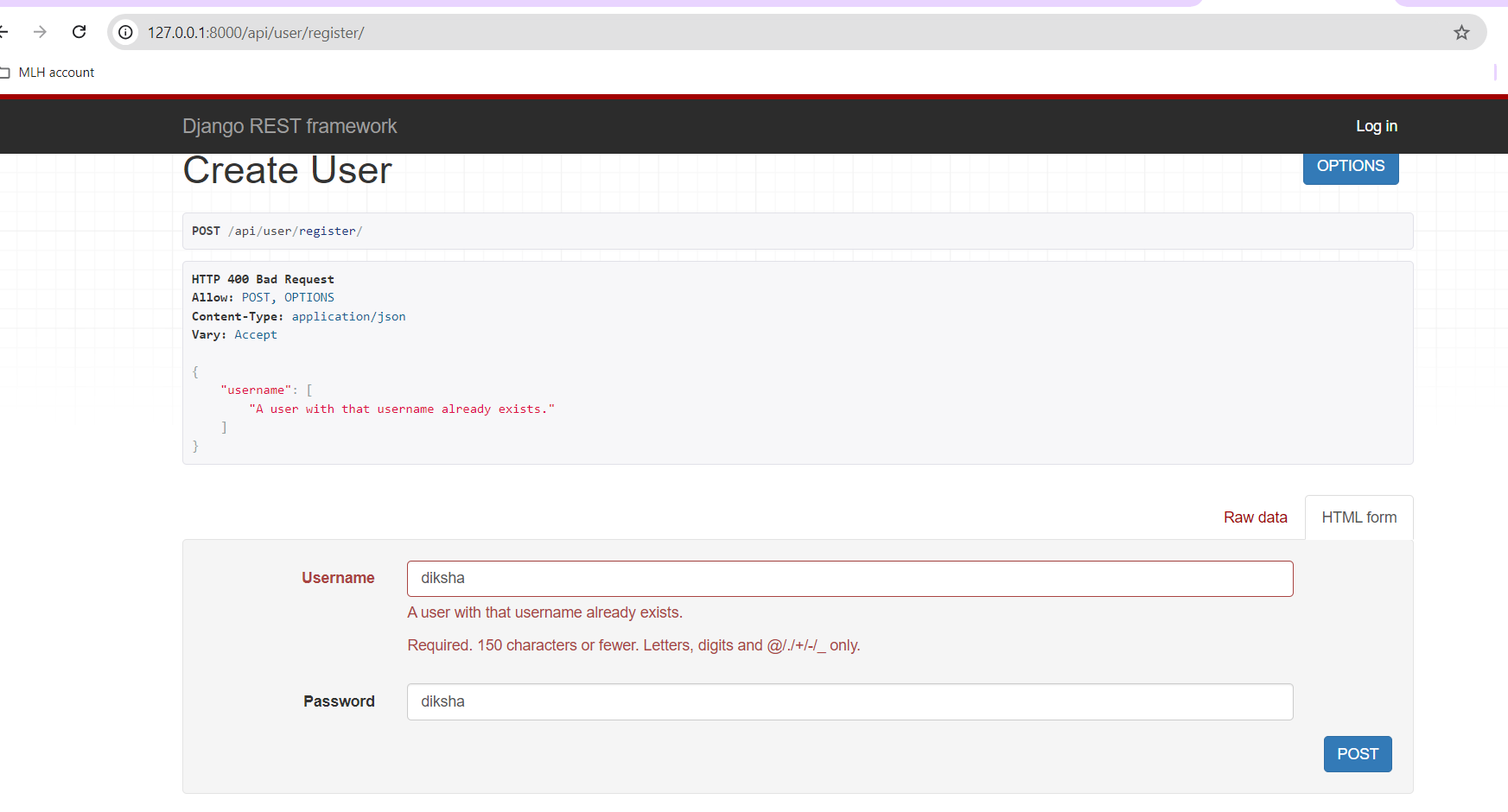
1. FOR REGISTRATION : http://127.0.0.1:8000/api/user/register/



1. Once user enters the details, the user gets register

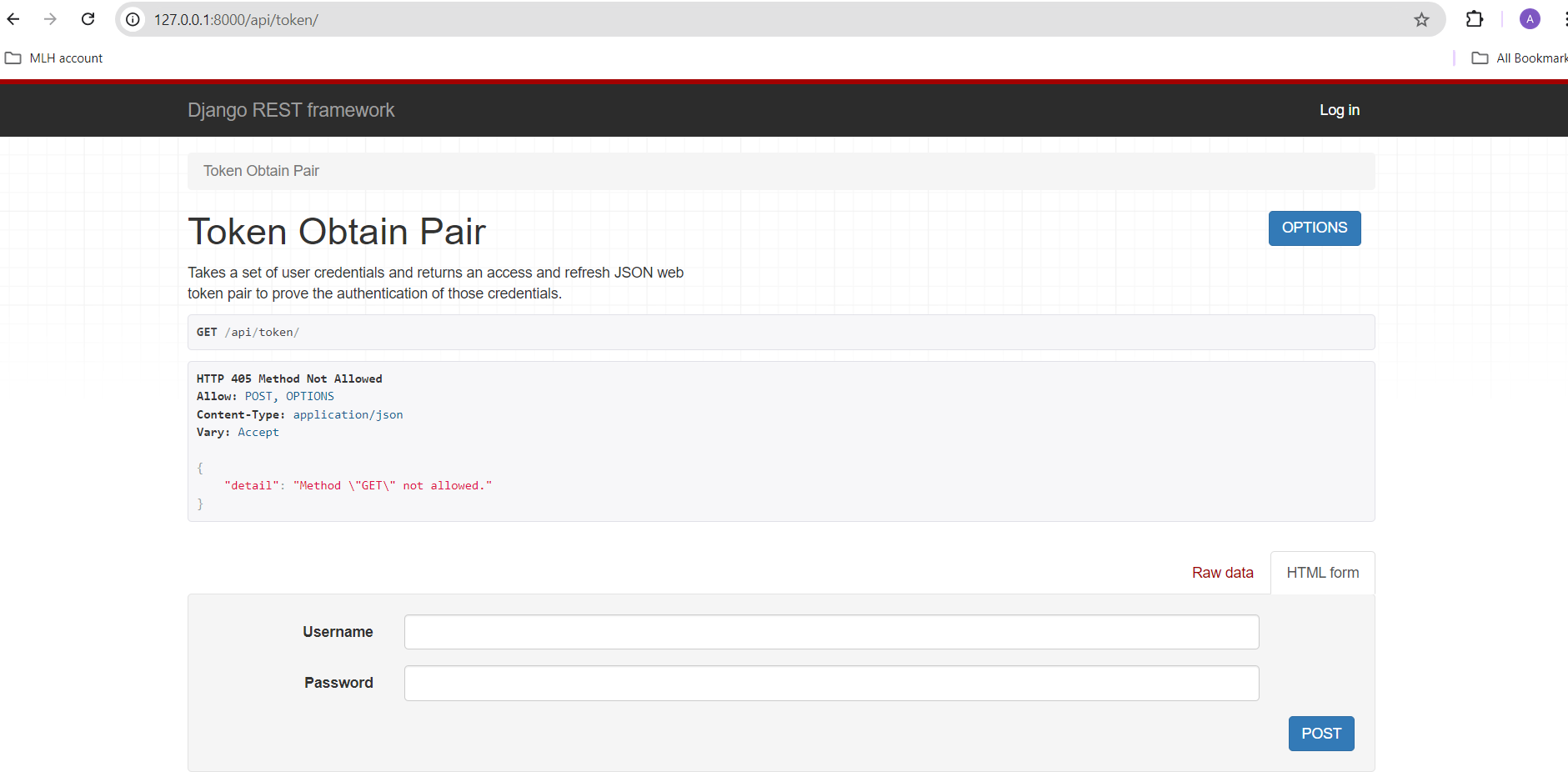


1. Do not allow duplication username

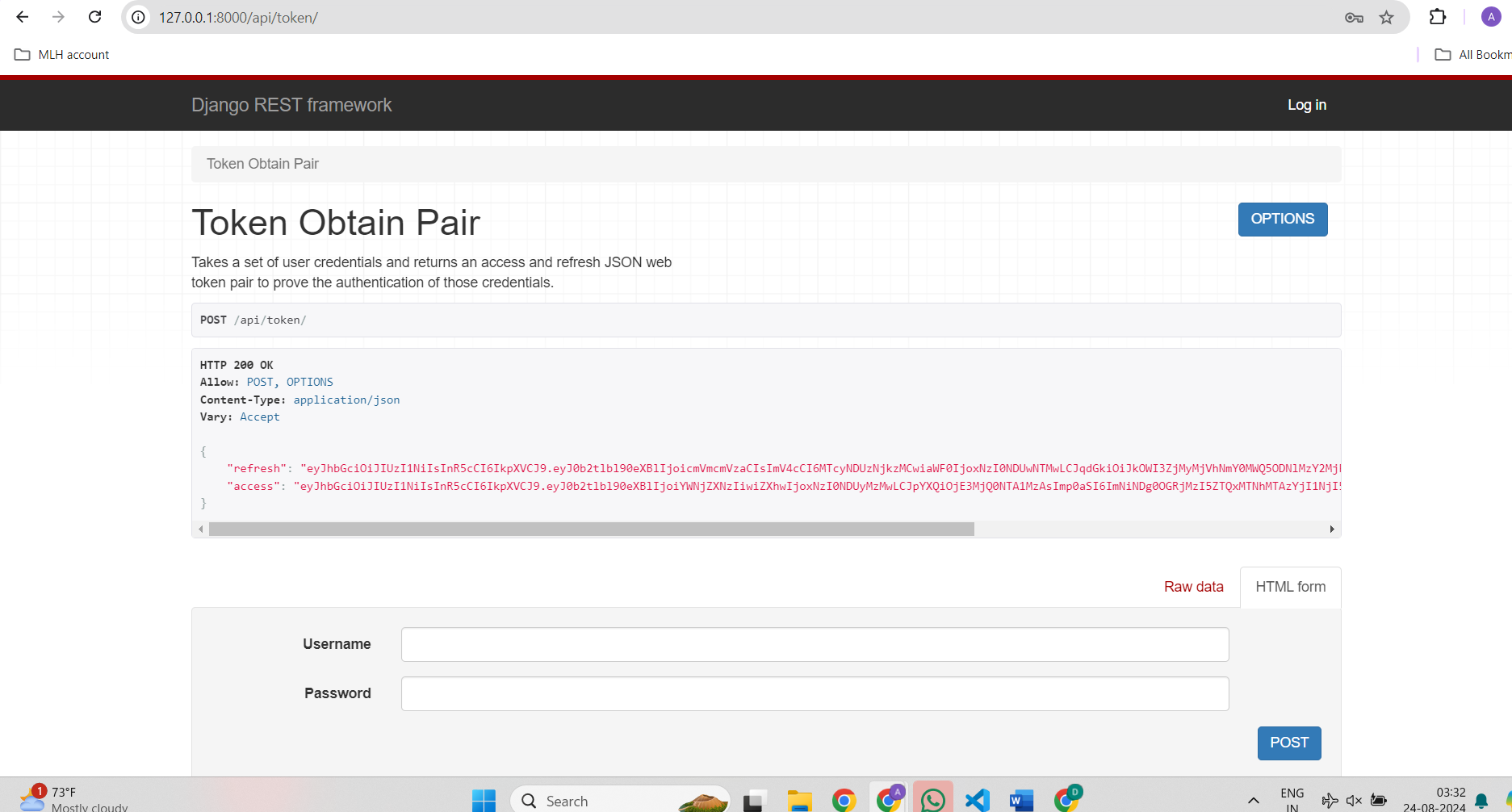


1. Token url

<http://127.0.0.1:8000/api/token/>



1. Token details



1. Refersh token link (http://127.0.0.1:8000/api/token/refresh/)

