Install virtual work venv

Install user-agent

pip install django-user-agents

Python -m venv venv (sur le cmd du dossier)

venv\scripts\activate (activate virtual work)

(to install django dans le venv)

pip install Django

(to install djangorestframework)

pip install djangorestframework

pour acceder a la page on passse par le project créer avec le code

django-admin startproject \*nom\*

(creer une application sur python)

python manage.py startapp allan20

et le cmd et on ecrit

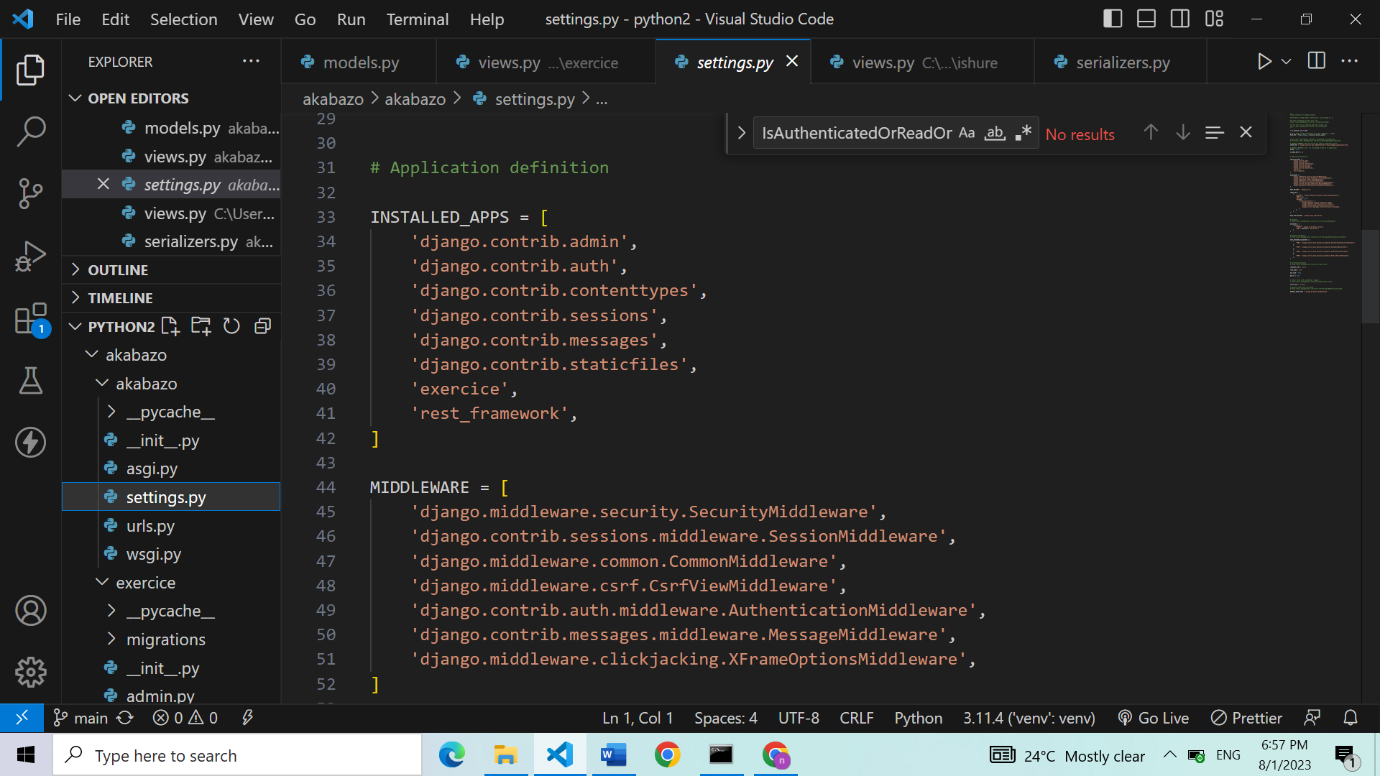
python manage.py runserver

python manage.py makemigrations

python manage.py migrate

python manage.py createsuperuser (allan20 code ntwari2032002) (kenallan kenallan) (ken ken123)(levin levin 123)

inserer dans le projet fichier « setting » le nom de l’application



appliquer les code suivant dans le code « exemple1 »

from django.contrib import admin

from django.urls import path,include

urlpatterns = [

    path('admin/', admin.site.urls),

    path('allan20/', include('allan20.urls')),

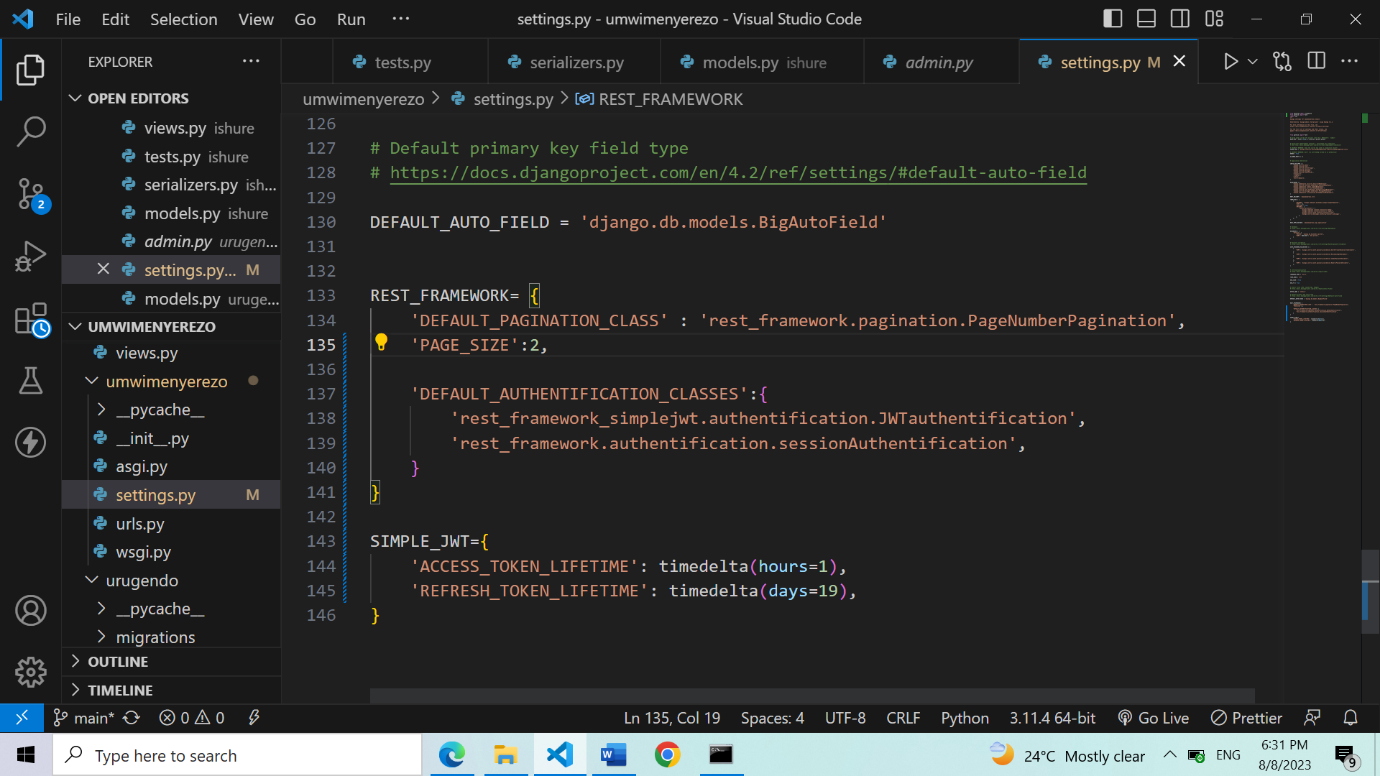
]

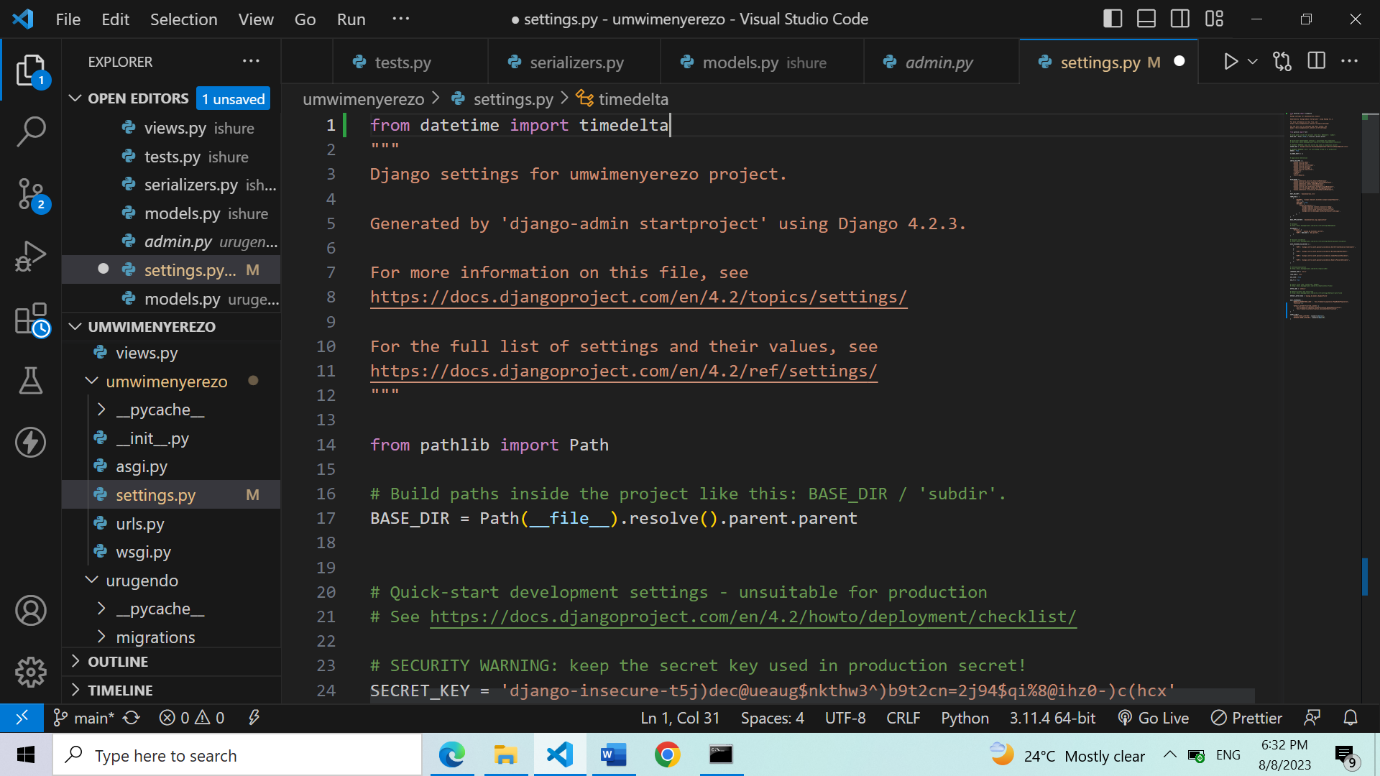
copier le fichier “urls” dans le projet « exemple1 » et le mettre dans l’aplication « allan20 »

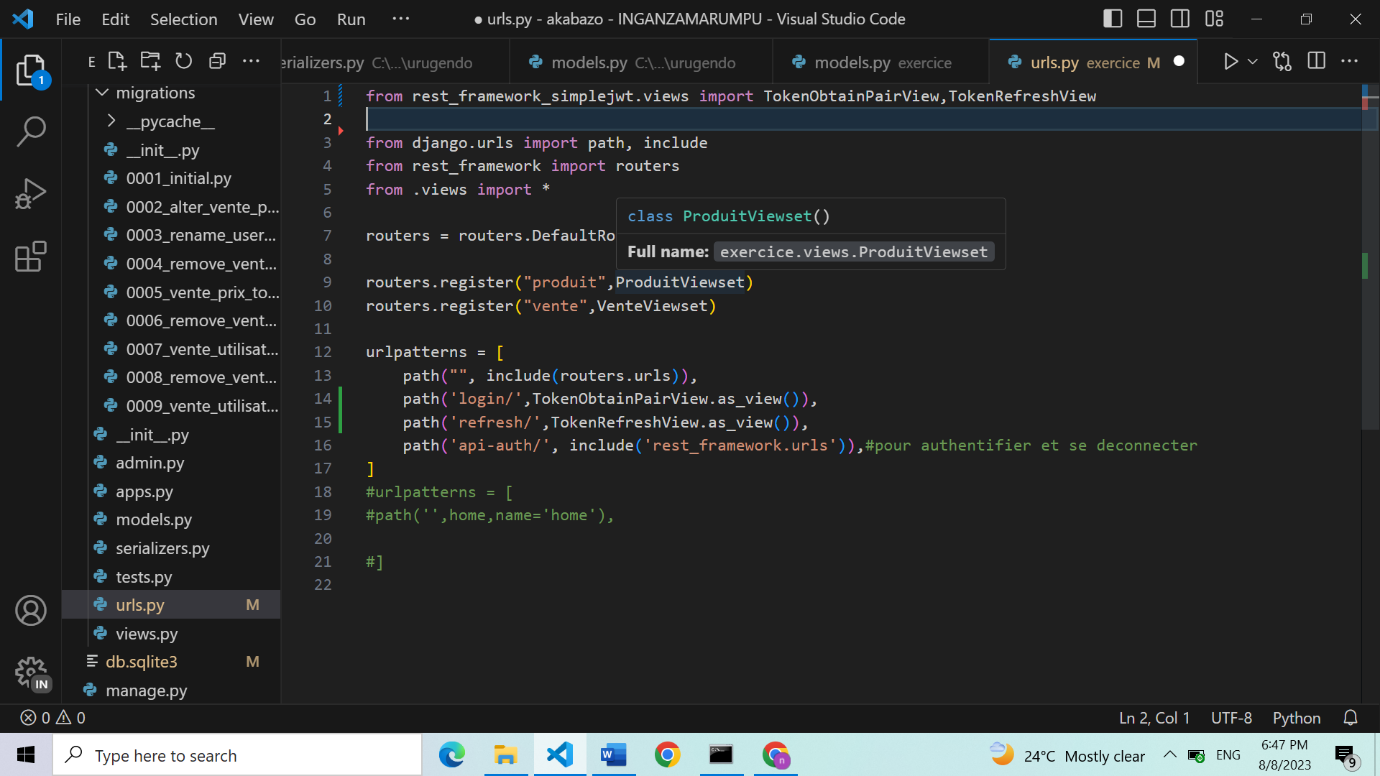
(install jwt in django) et la pagination dans une donner django

pip install djangorestframework-simplejwt

pour acceder on ecrit sur tableau de recherche 123 :800/login ou refresh



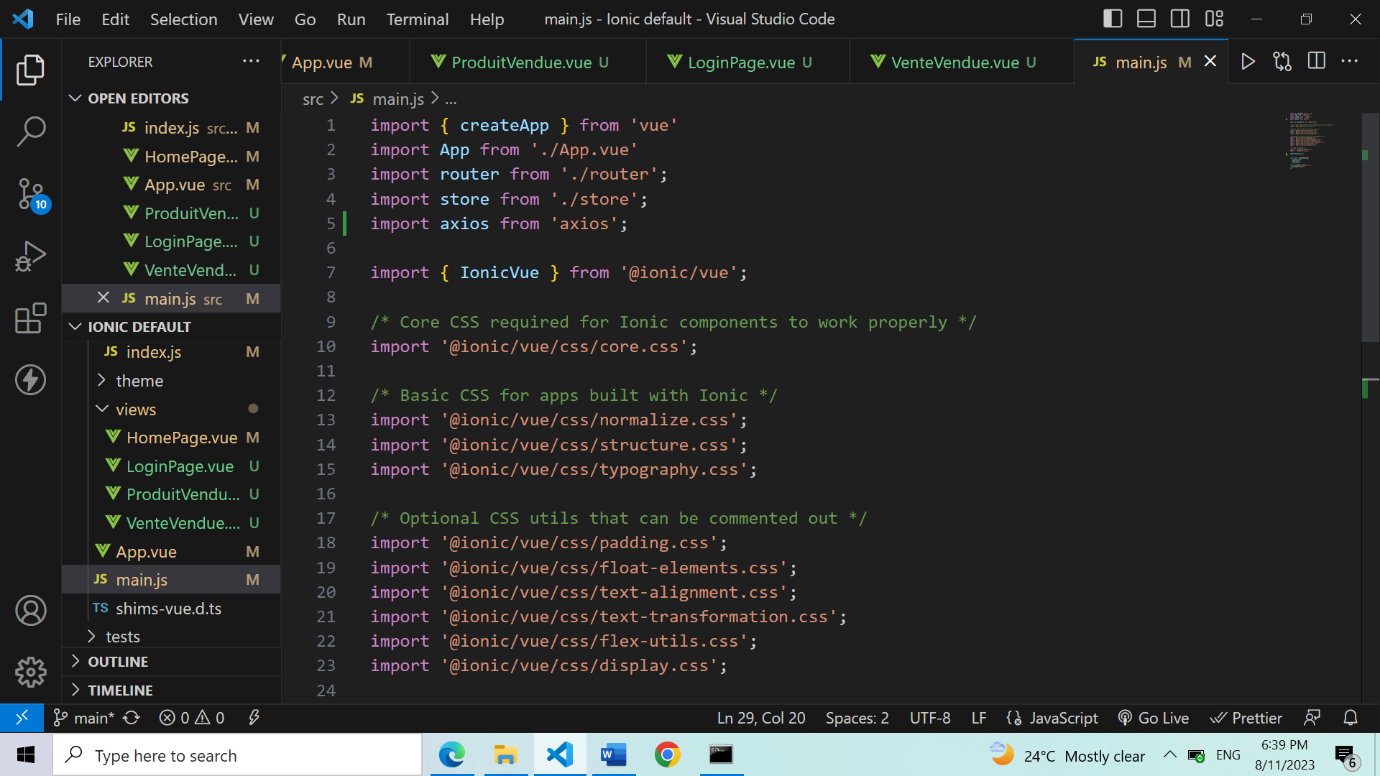


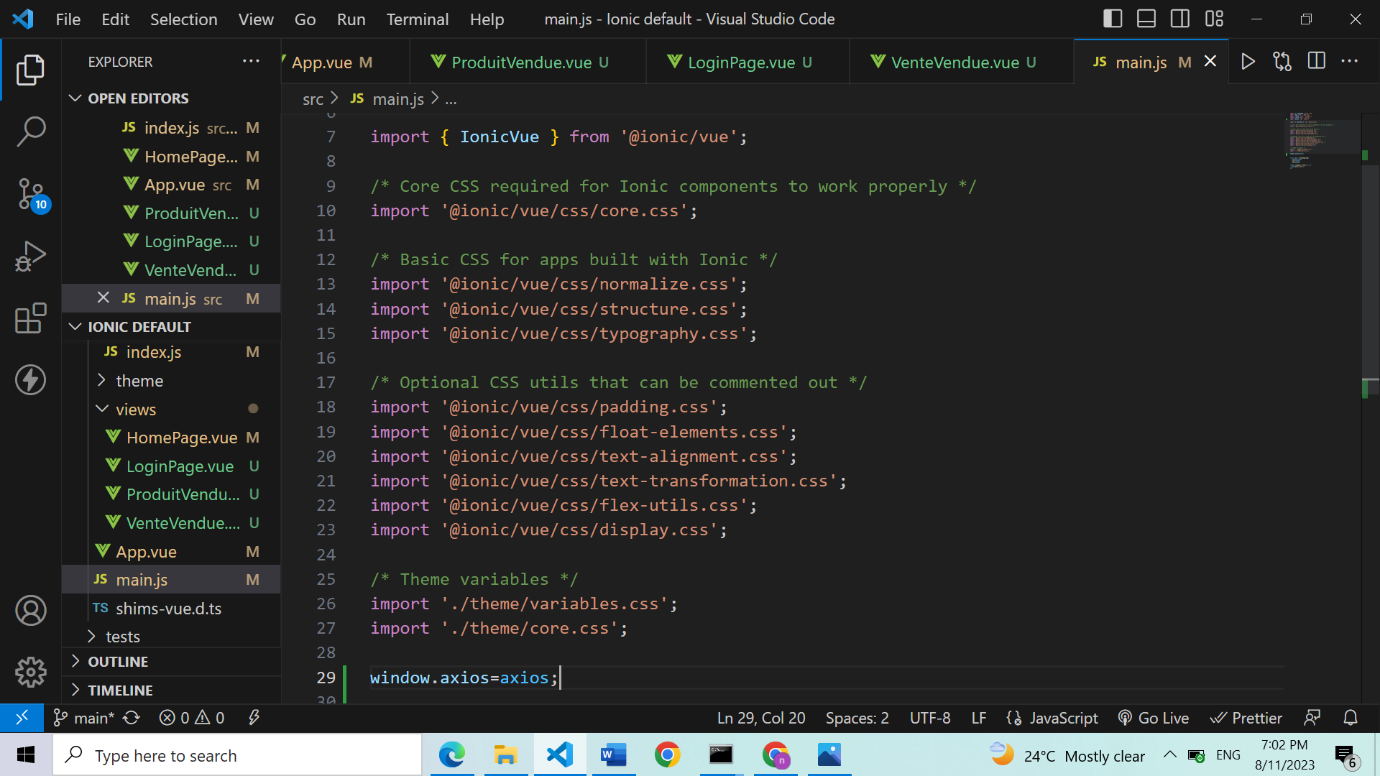


« npm install axios » dans le dossier ionic (connecter avec django sur Ionic)

Et s’assurer que le « npm install vuex » est installer pour traiter les infos

Ecrit « axios.get » pour voir si ca marche dans le console du projet ionic

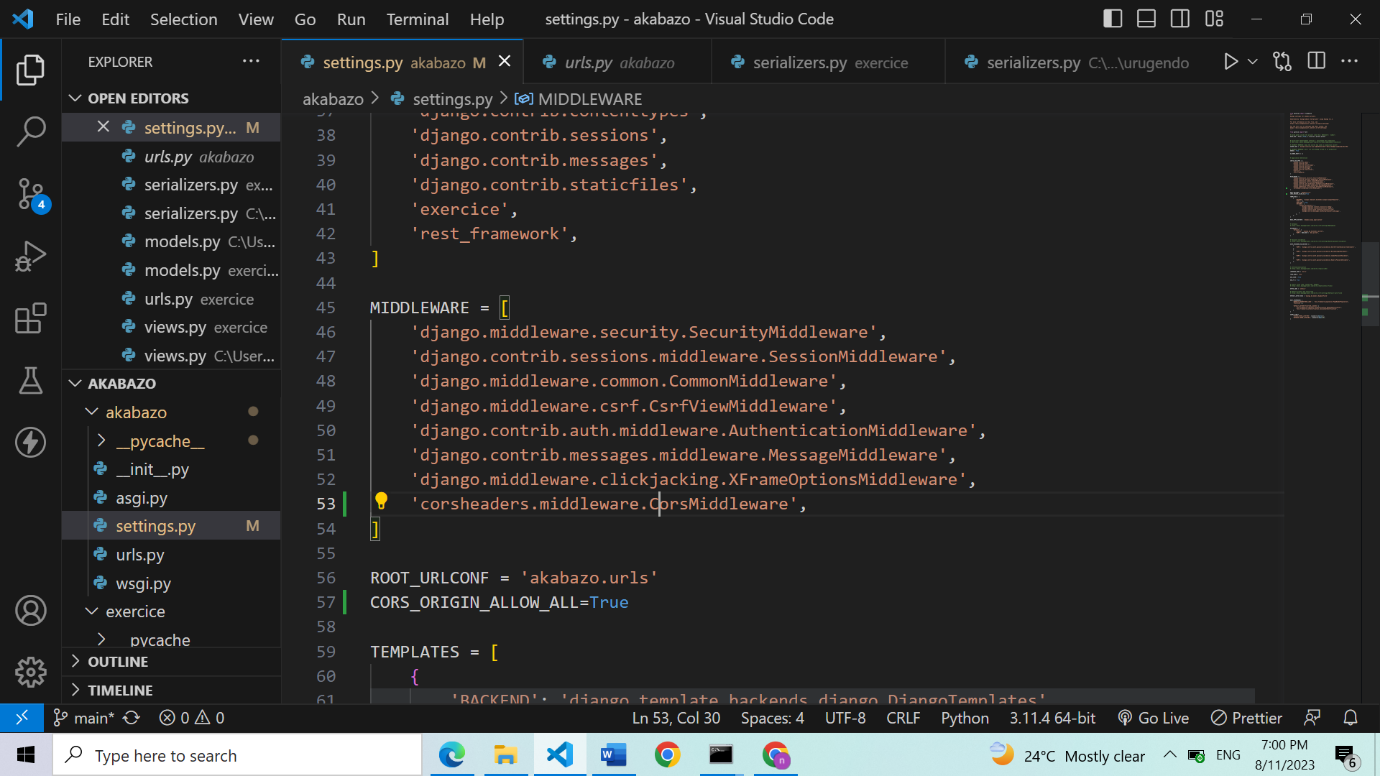




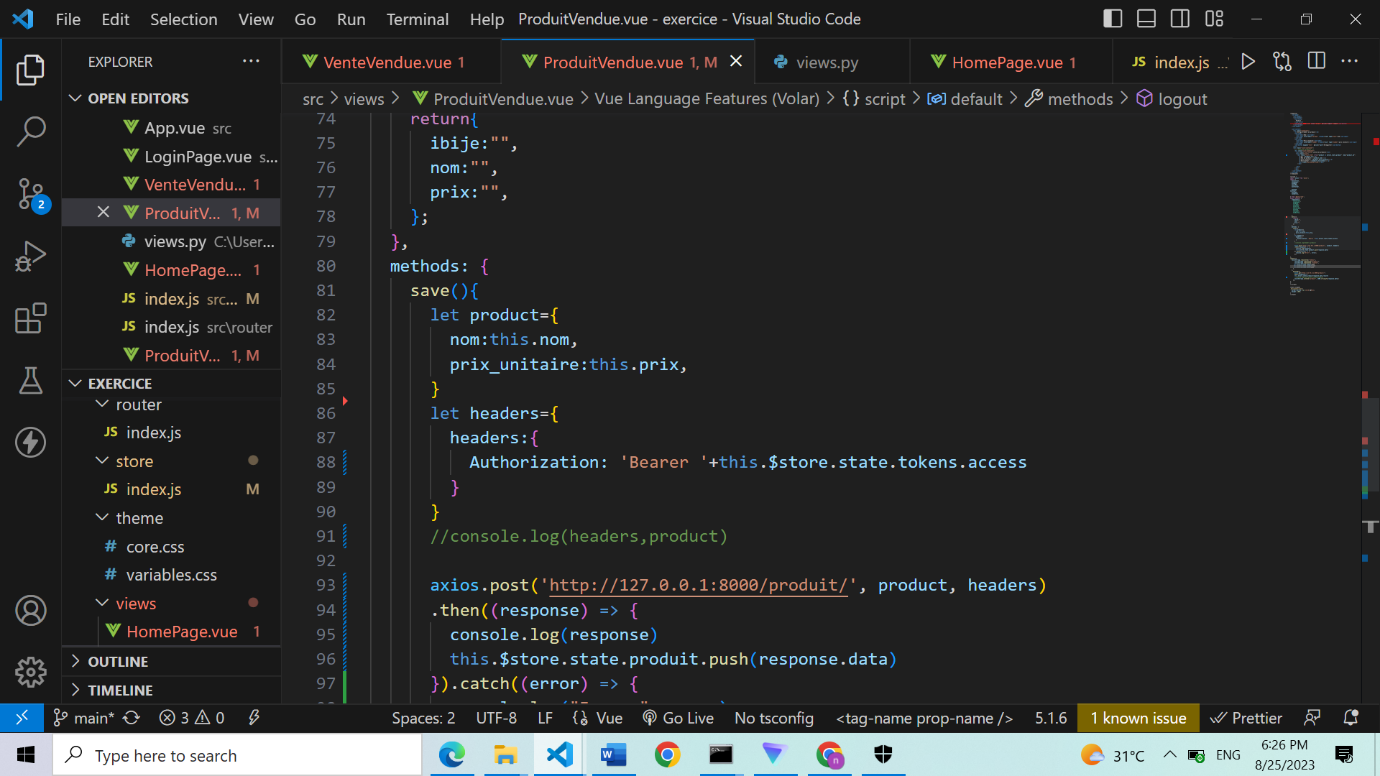
Pour connecter les deux pages ionic et django if faut installer dans le projet pyhton dans le meme racine que venv

« pip install django-cors-headers “

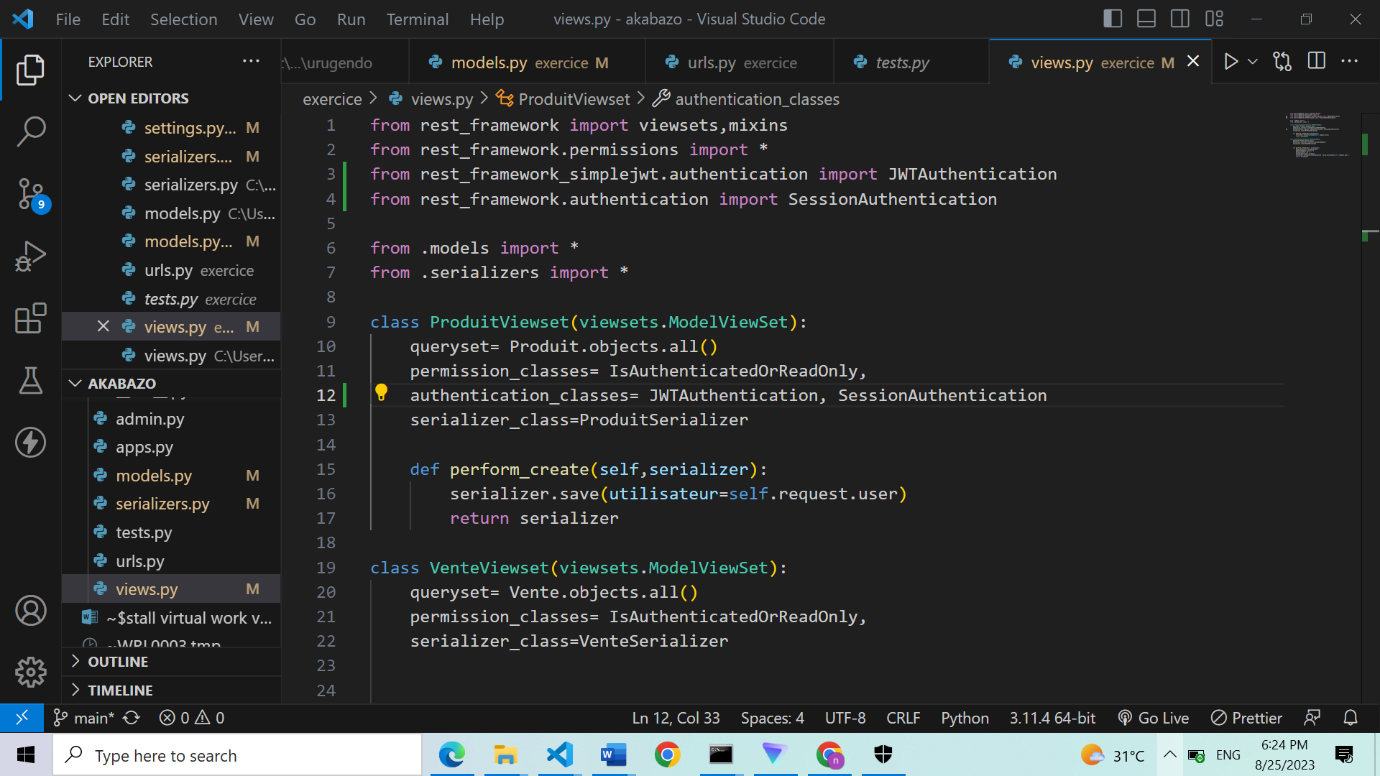
Install dans le setting



Pour envoyer sur le back end il faut pour le ionic



Et sur le backend il faut



Pour supprimer un data sur ionic provenant de djnago il faut sur ionic

    handleCheckboxChange(selectedProduct) {

      this.prod.forEach(product => {

        if (product !== selectedProduct) {

          product.isSelected = false;

        }

      });

      if (selectedProduct) {

        console.log("Vente sélectionnée. ID:", selectedProduct.id);

        if (confirm("Êtes-vous sûr de vouloir supprimer ce produit ?")) {

          axios.delete(`http://127.0.0.1:8000/produit/${selectedProduct.id}/`, {

              headers: {

                Authorization: 'Bearer ' + this.$store.state.tokens.access,

              },

            })

         .then(response => {

         // mise a jour de la liste

               axios.get("http://127.0.0.1:8000/produit/")

                    .then((response)=>{

      this.$store.state.produit=response.data.results

      this.prod=this.$store.state.produit

      localStorage.setItem("produit", JSON.stringify(response.data))

    });

              console.log(response);

              // Actualiser la liste des produits

              this.prod = this.prod.filter(product => product.id !== selectedProduct.id);

              console.log('Ressource supprimée avec succès');

            })

            .catch(error => {

              console.error('Erreur lors de la suppression de la ressource:', error);

            });

          }

      } else {

        console.log("Vente désélectionnée. ID:", selectedProduct.id);

      }

    },

Sur django aller sur model pour s’assurer que les foreign key n’empeche pas de supprimer

   nom=models.ForeignKey(Produit,on\_delete=models.SET\_NULL, null=True)pour supprimer le foreign key tout en gardant les les donner déjà sauvegarder

utilisateur=models.ForeignKey(User,on\_delete=models.PROTECT)

pour lier les deux model a savoir lorsque le foreign key est supprimer soit bloquer