MBARARA UNIVERSITY OF SCIENCE AND TECHNOLOGY

Course Unit :WEB DEVELOPMENT
Date: Sat,19th August 2019

NAHURIRA COLLIN BLESSING 2021bcs052ps

WEBSITE CLONING PROJECT REPORT

1. Introduction:

In this report, I provide a comprehensive overview of my web development project, in which I cloned the Emis portal website using Django as the backend framework. This project allowed me to immerse myself in web development, from design replication to functional implementation, while integrating the power of Django.

2. Website Selection:

I selected the Emis portal website due to its modern design, interactive features, and potential to demonstrate Django's capabilities. The original website offers a platform for school and other institution to upload data about the learners, staff , infraastructure and many more.

3. Project Overview:

For this project, I utilized a combination of HTML, CSS, JavaScript, and Django to recreate the Emis portal website. Visual Studio Code served as my primary code editor, while Git facilitated version control.

4. Website Cloning Process:

Django Setup: I began by creating a Django project and apps for different sections of the website, such as login and dashboard.
Database Models: I designed Django models to mimic the structure of the original website's data, including learners, users, and school staff.
HTML Templates:Using Django's template engine, I created reusable HTML templates that dynamically generate content from the database.

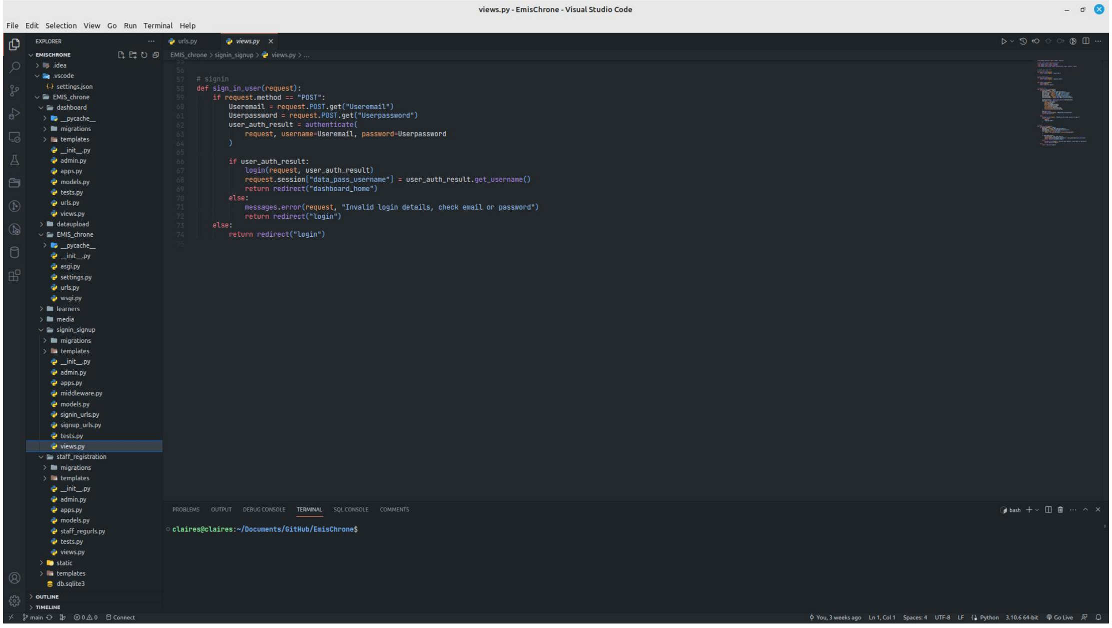
5. Functionality:

I successfully cloned and implemented the following functionalities from the original Emis portal website:

User Authentication and Registration

Utilizing Django's built-in authentication system, I created a login page where users can securely log in with their credentials. Additionally, I designed a registration page that allows new users to create accounts. This ensures data security and enables personalized experiences.

code snippet For login

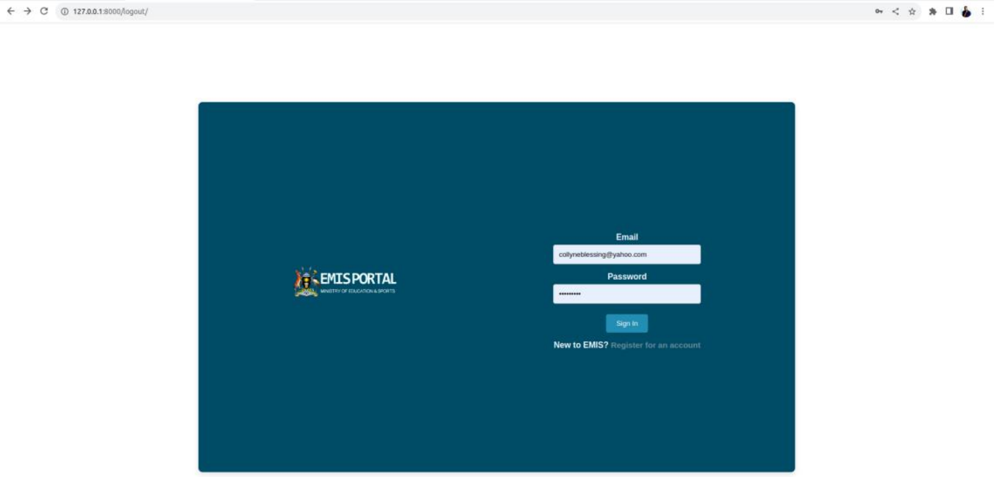


views.py - EmisChrone - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
urls.py
views.py
EMIS_chrone > signin_signup > views.py > ...
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL SQL CONSOLE COMMENTS
bash

```
# signin
def sign_in_user(request):
    if request.method == "POST":
        Useremail = request.POST.get("Useremail")
        Userpassword = request.POST.get("Userpassword")
        user_auth_result = authenticate(
            request, username=Useremail, password=Userpassword
        )

        if user_auth_result:
            login(request, user_auth_result)
            request.session["data_pass_username"] = user_auth_result.get_username()
            return redirect("dashboard_home")
        else:
            messages.error(request, "Invalid login details, check email or password")
            return redirect("login")
    else:
        return redirect("login")
```

```
claires@claires:~/Documents/GitHub/EmisChrone$
```

login_page

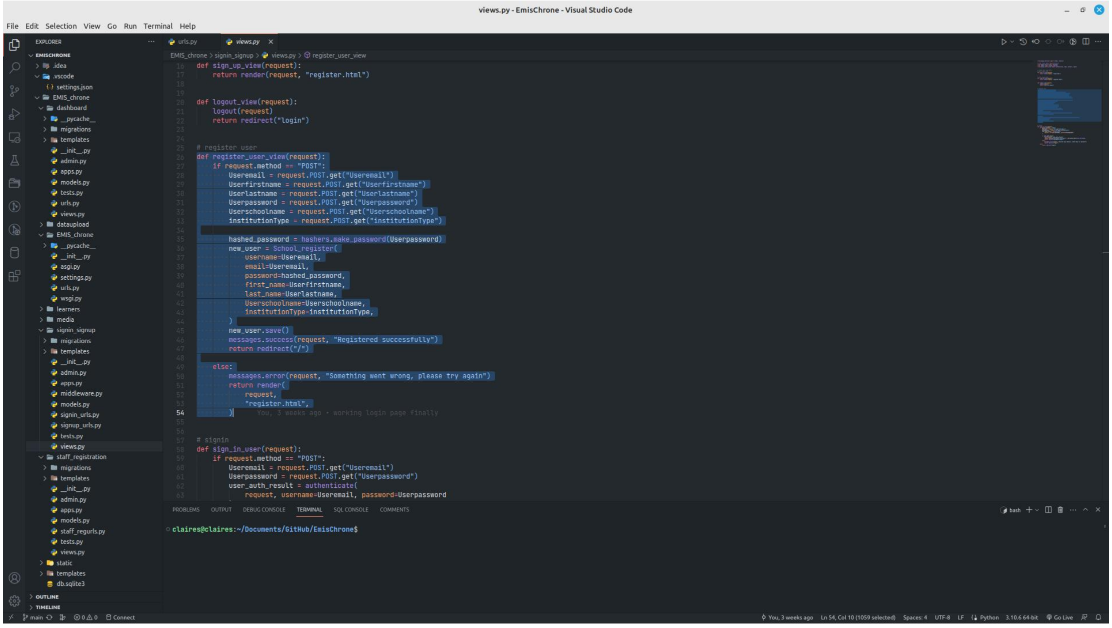


Email: collyneblessing@yahoo.com

Password: ••••••••••

New to EMIS? Register for an account

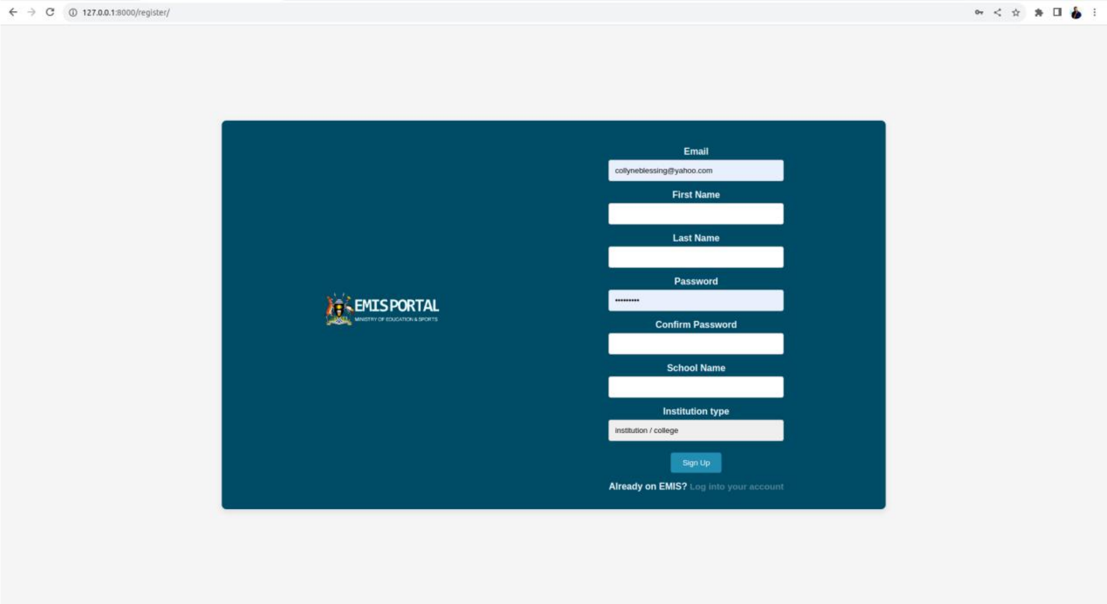
code snippet for register page


views.py - EmisChrone - Visual Studio Code
File Edit Selection View Go Run Terminal Help
urls.py
views.py
EMIS_chrone > signin_signup > views.py > register_user_view
You, 3 weeks ago • working login page finally
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
SQL CONSOLE
COMMENTS

```python
def sign_up_view(request):
    return render(request, "register.html")


def logout_view(request):
    logout(request)
    return redirect("login")


# register user
def register_user_view(request):
    if request.method == "POST":
        Useremail = request.POST.get("Useremail")
        Userfirstname = request.POST.get("Userfirstname")
        Userlastname = request.POST.get("Userlastname")
        Userpassword = request.POST.get("Userpassword")
        Userschoolname = request.POST.get("Userschoolname")
        institutionType = request.POST.get("institutionType")

        hashed_password = hashers.make_password(Userpassword)
        new_user = School_register(
            username=Useremail,
            email=Useremail,
            password=hashed_password,
            first_name=Userfirstname,
            last_name=Userlastname,
            Userschoolname=Userschoolname,
            institutionType=institutionType,
        )
        new_user.save()
        messages.success(request, "Registered successfully")
        return redirect("/")

    else:
        messages.error(request, "Something went wrong, please try again")
        return render(
            request,
            "register.html",
        )


# signin
def sign_in_user(request):
    if request.method == "POST":
        Useremail = request.POST.get("Useremail")
        Userpassword = request.POST.get("Userpassword")
        user_auth_result = authenticate(
            request, username=Useremail, password=Userpassword
```

```
claires@claires:~/Documents/GitHub/EmisChrone$
```

register_page



Email: collyneblessing@yahoo.com

First Name:

Last Name:

Password: ••••••••••

Confirm Password:

School Name:

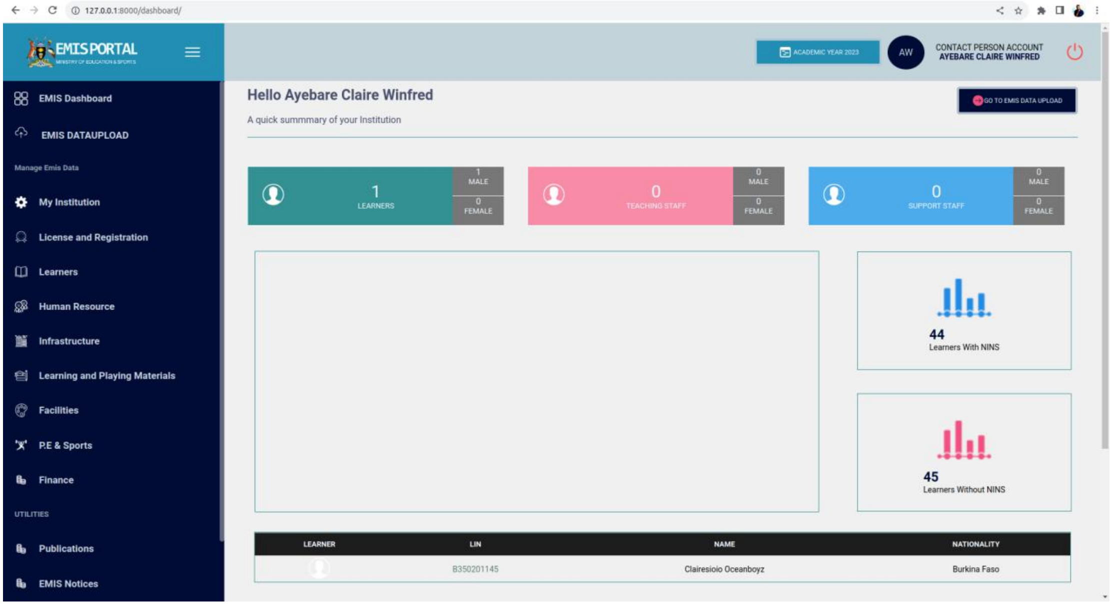
Institution type: institution / college

Already on EMIS? Log into your account

Dashboard:

I developed a personalized user dashboard that provides users with an overview of their activity on the website. The dashboard displays their recent forum activity, liked learners, and uploaded content.

dashboard



Hello Ayebare Claire Winfred

A quick summmary of your Institution

1 LEARNERS — 1 MALE, 0 FEMALE

0 TEACHING STAFF — 0 MALE, 0 FEMALE

0 SUPPORT STAFF — 0 MALE, 0 FEMALE

44 Learners With NINS

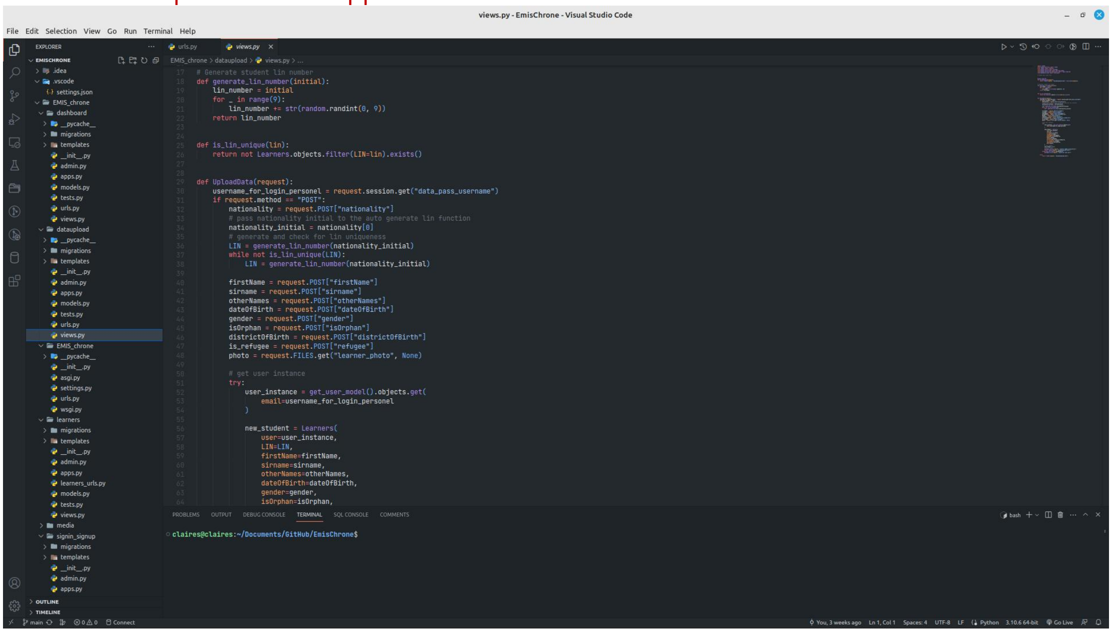
45 Learners Without NINS

LEARNER	LIN	NAME	NATIONALITY
	B350201145	Clairesioio Oceanboyz	Burkina Faso

Emis Data Upload:

Building on the foundation of Django, I integrated a feature allowing authorized users to upload educational data files in Emis format. This functionality contributes to the platform's educational resources.

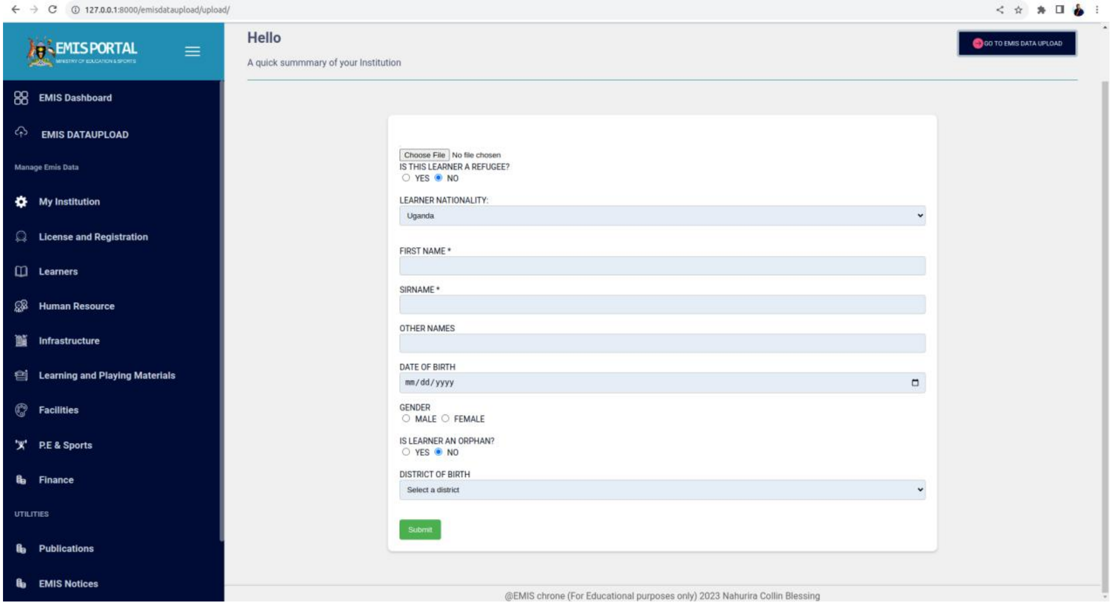
learner data upload code snippet


views.py - EmisChrone - Visual Studio Code
File Edit Selection View Go Run Terminal Help
urls.py
views.py
EMIS_chrone > dataupload > views.py > ...
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL SQL CONSOLE COMMENTS

```python
# Generate student lin number
def generate_lin_number(initial):
    lin_number = initial
    for _ in range(9):
        lin_number += str(random.randint(0, 9))
    return lin_number


def is_lin_unique(lin):
    return not Learners.objects.filter(LIN=lin).exists()


def UploadData(request):
    username_for_login_personel = request.session.get("data_pass_username")
    if request.method == "POST":
        nationality = request.POST["nationality"]
        # pass nationality initial to the auto generate lin function
        nationality_initial = nationality[0]
        # generate and check for lin uniqueness
        LIN = generate_lin_number(nationality_initial)
        while not is_lin_unique(LIN):
            LIN = generate_lin_number(nationality_initial)

        firstName = request.POST["firstName"]
        sirname = request.POST["sirname"]
        otherNames = request.POST["otherNames"]
        dateOfBirth = request.POST["dateOfBirth"]
        gender = request.POST["gender"]
        isOrphan = request.POST["isOrphan"]
        districtOfBirth = request.POST["districtOfBirth"]
        is_refugee = request.POST["refugee"]
        photo = request.FILES.get("learner_photo", None)

        # get user instance
        try:
            user_instance = get_user_model().objects.get(
                email=username_for_login_personel
            )

            new_student = Learners(
                user=user_instance,
                LIN=LIN,
                firstName=firstName,
                sirname=sirname,
                otherNames=otherNames,
                dateOfBirth=dateOfBirth,
                gender=gender,
                isOrphan=isOrphan,
```

```
claires@claires:~/Documents/GitHub/EmisChrone$
```

learners_data_upload



Hello

A quick summmary of your Institution

Choose File No file chosen

IS THIS LEARNER A REFUGEE?

○ YES ● NO

LEARNER NATIONALITY:

Uganda

FIRST NAME *

SIRNAME *

OTHER NAMES

DATE OF BIRTH

mm/dd/yyyy

GENDER

○ MALE ○ FEMALE

IS LEARNER AN ORPHAN?

○ YES ● NO

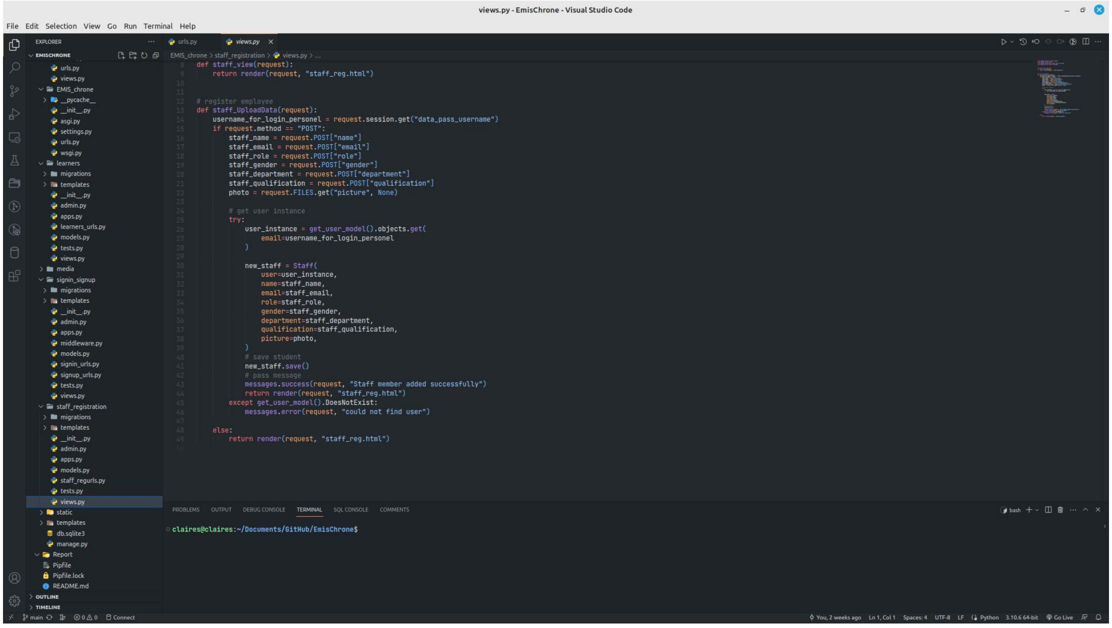
DISTRICT OF BIRTH

Select a district

Submit

@EMIS chrone (For Educational purposes only) 2023 Nahurira Collin Blessing

staff upload code snippet

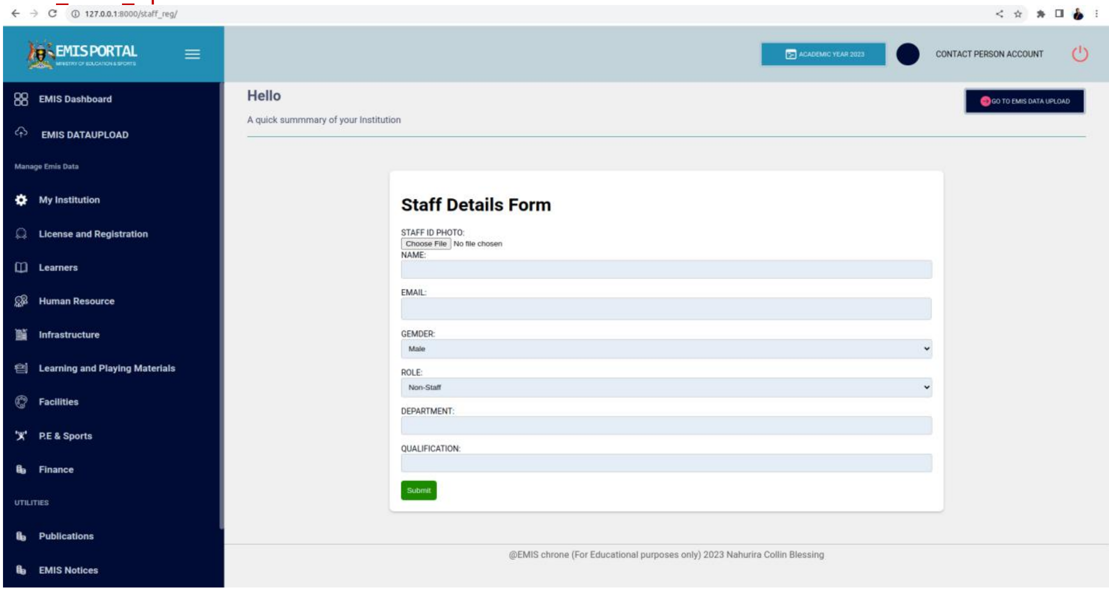


views.py - EmisChrone - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
urls.py
views.py
EMIS_chrone > staff_registration > views.py > ...
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL SQL CONSOLE COMMENTS

```python
def staff_view(request):
    return render(request, "staff_reg.html")


# register employee
def staff_UploadData(request):
    username_for_login_personel = request.session.get("data_pass_username")
    if request.method == "POST":
        staff_name = request.POST["name"]
        staff_email = request.POST["email"]
        staff_role = request.POST["role"]
        staff_gender = request.POST["gender"]
        staff_department = request.POST["department"]
        staff_qualification = request.POST["qualification"]
        photo = request.FILES.get("picture", None)

        # get user instance
        try:
            user_instance = get_user_model().objects.get(
                email=username_for_login_personel
            )

            new_staff = Staff(
                user=user_instance,
                name=staff_name,
                email=staff_email,
                role=staff_role,
                gender=staff_gender,
                department=staff_department,
                qualification=staff_qualification,
                picture=photo,
            )
            # save student
            new_staff.save()
            # pass message
            messages.success(request, "Staff member added successfully")
            return render(request, "staff_reg.html")
        except get_user_model().DoesNotExist:
            messages.error(request, "could not find user")

    else:
        return render(request, "staff_reg.html")
```

```
claires@claires:~/Documents/GitHub/EmisChrone$
```

staff_data_upload



Hello

A quick summmary of your Institution

Staff Details Form

STAFF ID PHOTO:
Choose File No file chosen

NAME:

EMAIL:

GEMDER:
Male

ROLE:
Non-Staff

DEPARTMENT:

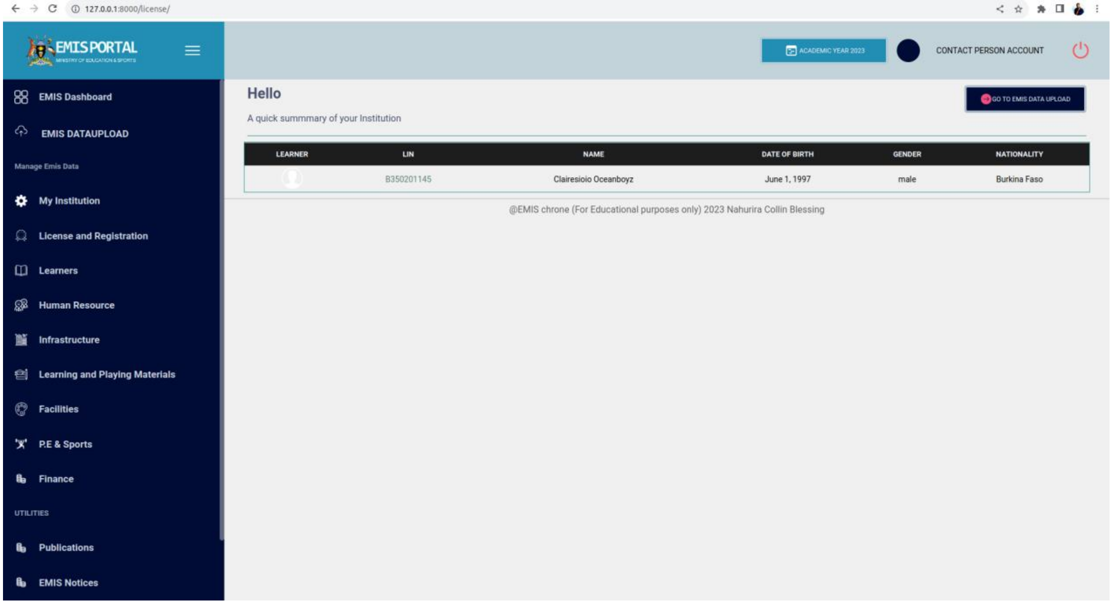
QUALIFICATION:

Submit

@EMIS chrone (For Educational purposes only) 2023 Nahurira Collin Blessing

Learners Page:
I implemented a dedicated page where users can access educational content. This page displays photos, nationality, and other information relevant to learners. The content is dynamically generated using Django's template engine.

learners_Page



Hello

A quick summmary of your Institution

LEARNER	LIN	NAME	DATE OF BIRTH	GENDER	NATIONALITY
	B350201145	Clairesioio Oceanboyz	June 1, 1997	male	Burkina Faso

@EMIS chrone (For Educational purposes only) 2023 Nahurira Collin Blessing

These functionalities were implemented in alignment with the original website's design and user experience, contributing to the overall interactive and engaging nature of the cloned website.

6. **Design and Layout:**

I aimed to retain the original design's aesthetics, focusing on responsive layout, consistent typography, and the same color palette. The responsive design adapts to different screen sizes.

7. **Code Structure and Organization:**

Django's project and app structure naturally organizes code. I maintained separation between HTML templates, CSS, and JavaScript files within each app.

8. **Testing:**

I extensively tested the cloned websites on various devices and browsers to identify and address any layout issues, broken links, or JavaScript errors. Cross-browser compatibility was achieved through targeted testing and adjustments.

9. **Challenges:**

- ➢ User Authentication and Authorization: Implementing a secure and seamless user authentication and authorization system posed challenges, especially when handling different user roles and permissions.
- ➢ Resource accessibility: I could not access some of the resources like NIRA features for utilizing full functionality of the parent identification section in the Learner registration module.

10**. Conclusion:**

Cloning the Emis portal website with Django has enriched my web development skills, deepening my understanding of HTML, CSS, JavaScript, and Django. This project underscored the value of attention to detail and responsive design in building user-friendly websites.

11. Future Improvements:

In the future, I plan to refine the animations and enhance the search functionality to provide more intuitive results.

12. References:

√ https://Emis.go.ug/

√ https://docs.djangoproject.com/

√ https://www.w3schools.com/django/

Complete project available on

√ https://github.com/collinBlessing/EmisChrone