Development Documentation

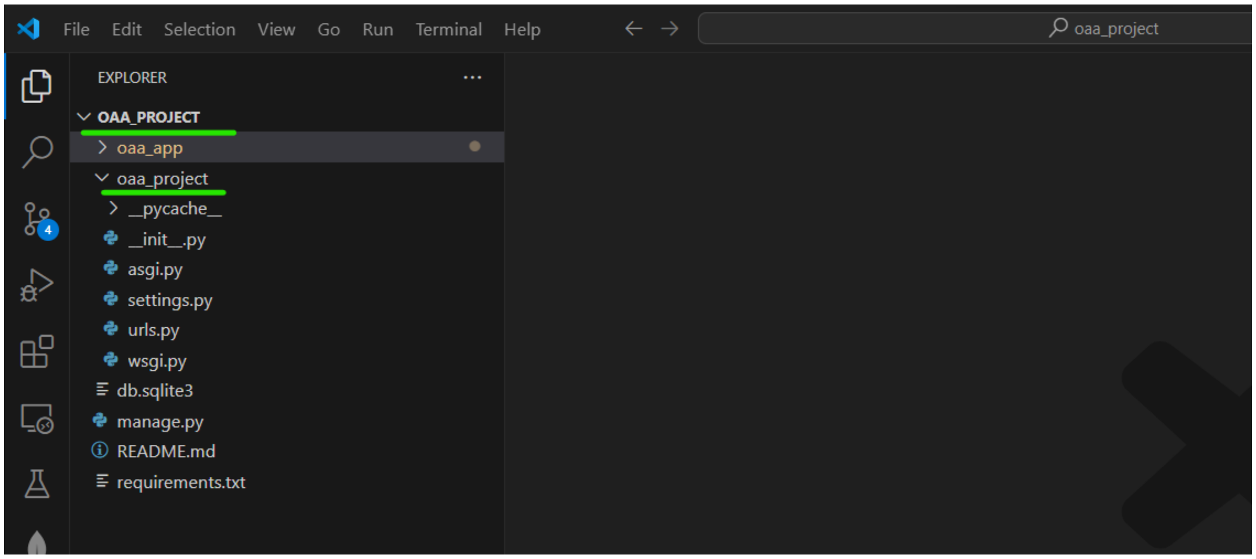
Ontario Academic Atlas

Introduction:

The OAA project is a Django web application designed to offer educational resources for students and educators in Ontario. This document provides an overview of the project setup and configuration, including initialization, settings.py, urls.py, and the project directory structure.

Initialize Django Project:

The oaa_project was initialized using the `django-admin startproject` command, creating a new Django project named 'oaa_project'. This command establishes the project directory structure and generates essential configuration files.



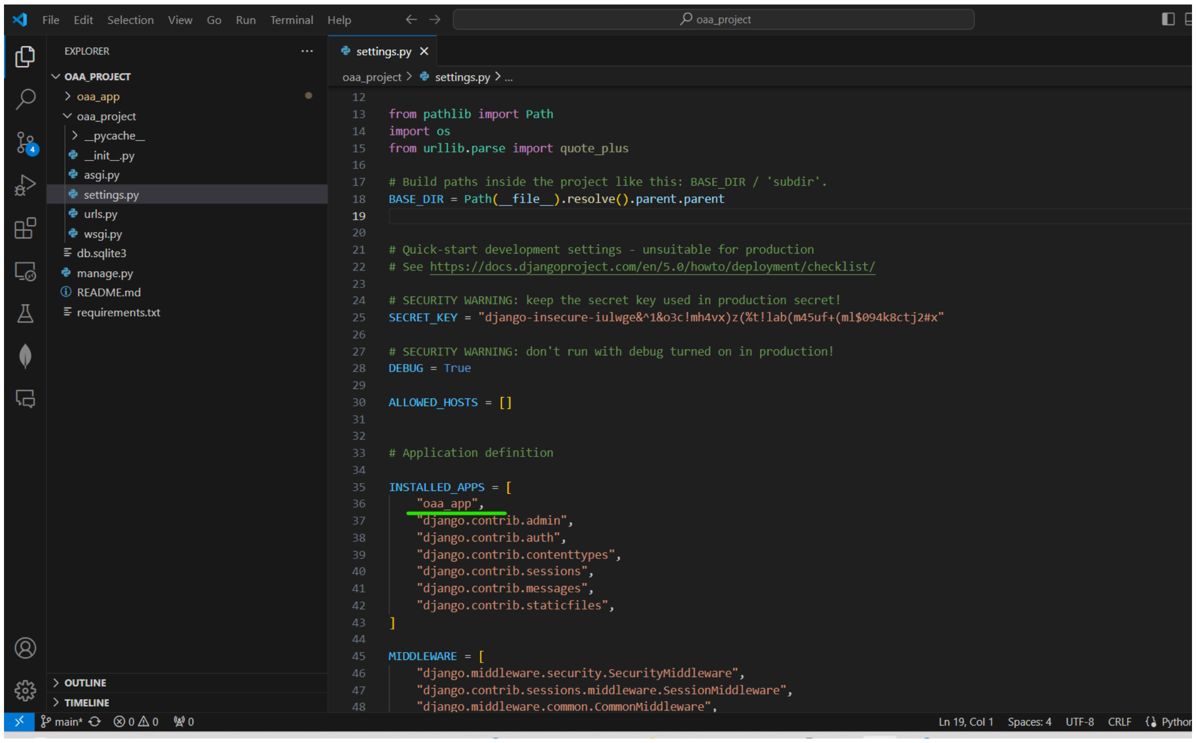
```
OAA_PROJECT
  oaa_app
  oaa_project
    __pycache__
    __init__.py
    asgi.py
    settings.py
    urls.py
    wsgi.py
  db.sqlite3
  manage.py
  README.md
  requirements.txt
```

Settings Configuration:

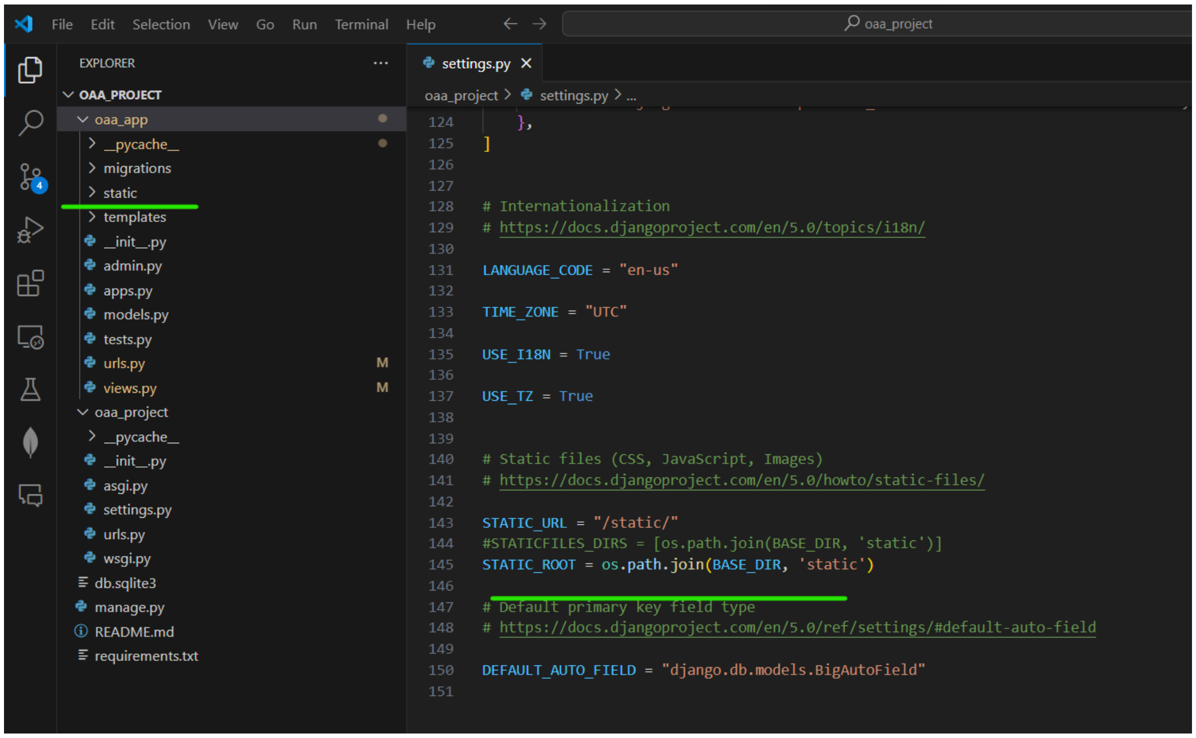
In the **settings.py** file, various settings for the Django project have been configured:

- DEBUG mode is enabled for development, facilitating detailed error pages and debugging tools.
- Installed apps include 'oaa_app' and other default Django apps required for basic project functionality.

- Database settings are configured for SQLite3, the default database engine provided by Django for development.
- Static files settings are defined to specify the location for serving CSS, JavaScript, and images. STATICFILES_DIRS is utilized to specify directories where Django will search for static files.



```python
from pathlib import Path
import os
from urllib.parse import quote_plus

# Build paths inside the project like this: BASE_DIR / 'subdir'.
BASE_DIR = Path(__file__).resolve().parent.parent


# Quick-start development settings - unsuitable for production
# See https://docs.djangoproject.com/en/5.0/howto/deployment/checklist/

# SECURITY WARNING: keep the secret key used in production secret!
SECRET_KEY = "django-insecure-iulwge&^1&o3c!mh4vx)z(%t!lab(m45uf+(ml$094k8ctj2#x"

# SECURITY WARNING: don't run with debug turned on in production!
DEBUG = True

ALLOWED_HOSTS = []


# Application definition

INSTALLED_APPS = [
    "oaa_app",
    "django.contrib.admin",
    "django.contrib.auth",
    "django.contrib.contenttypes",
    "django.contrib.sessions",
    "django.contrib.messages",
    "django.contrib.staticfiles",
]

MIDDLEWARE = [
    "django.middleware.security.SecurityMiddleware",
    "django.contrib.sessions.middleware.SessionMiddleware",
    "django.middleware.common.CommonMiddleware",
```

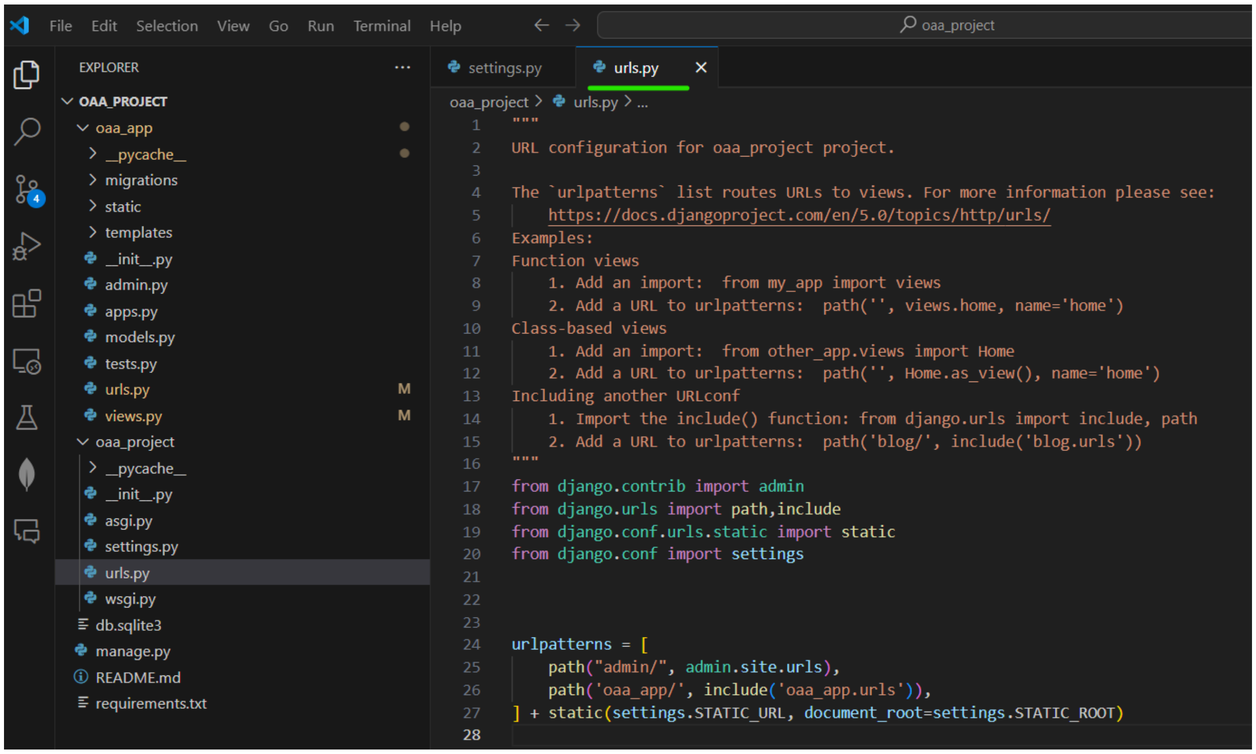


```python
    },
]


# Internationalization
# https://docs.djangoproject.com/en/5.0/topics/i18n/

LANGUAGE_CODE = "en-us"

TIME_ZONE = "UTC"

USE_I18N = True

USE_TZ = True


# Static files (CSS, JavaScript, Images)
# https://docs.djangoproject.com/en/5.0/howto/static-files/

STATIC_URL = "/static/"
#STATICFILES_DIRS = [os.path.join(BASE_DIR, 'static')]
STATIC_ROOT = os.path.join(BASE_DIR, 'static')

# Default primary key field type
# https://docs.djangoproject.com/en/5.0/ref/settings/#default-auto-field

DEFAULT_AUTO_FIELD = "django.db.models.BigAutoField"

```

URL Configuration:

The **urls.py** file in the project directory contains URL configuration for the project:

- URL patterns are mapped to views using the path() function from the Django.urls module.
- The admin URL pattern is included for accessing the Django admin interface to manage project data.
- URL patterns from the oaa_app application is included to define custom endpoints for the application's functionality.

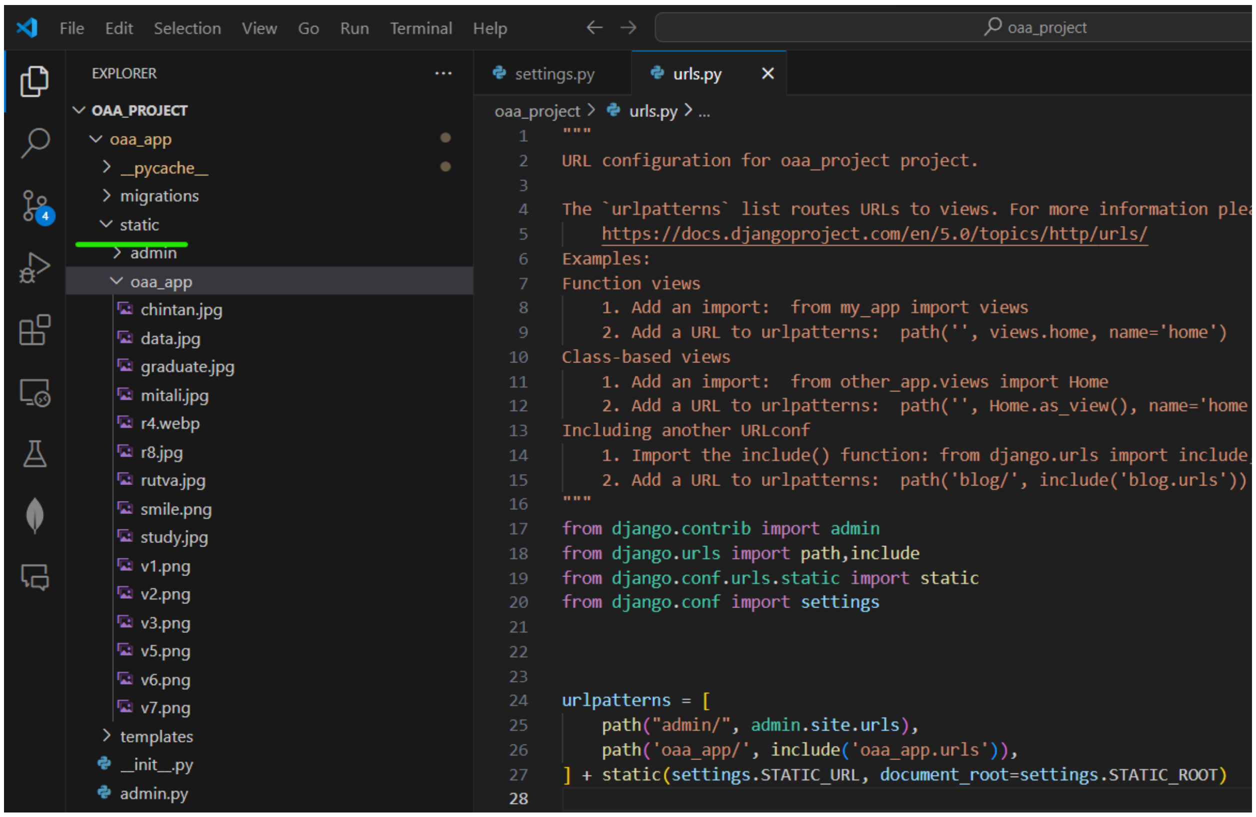


```python
"""
URL configuration for oaa_project project.

The `urlpatterns` list routes URLs to views. For more information please see:
    https://docs.djangoproject.com/en/5.0/topics/http/urls/
Examples:
Function views
    1. Add an import:  from my_app import views
    2. Add a URL to urlpatterns:  path('', views.home, name='home')
Class-based views
    1. Add an import:  from other_app.views import Home
    2. Add a URL to urlpatterns:  path('', Home.as_view(), name='home')
Including another URLconf
    1. Import the include() function: from django.urls import include, path
    2. Add a URL to urlpatterns:  path('blog/', include('blog.urls'))
"""
from django.contrib import admin
from django.urls import path,include
from django.conf.urls.static import static
from django.conf import settings



urlpatterns = [
    path("admin/", admin.site.urls),
    path('oaa_app/', include('oaa_app.urls')),
] + static(settings.STATIC_URL, document_root=settings.STATIC_ROOT)
```

Project Directory Structure:

The project directory comprises two additional folders:

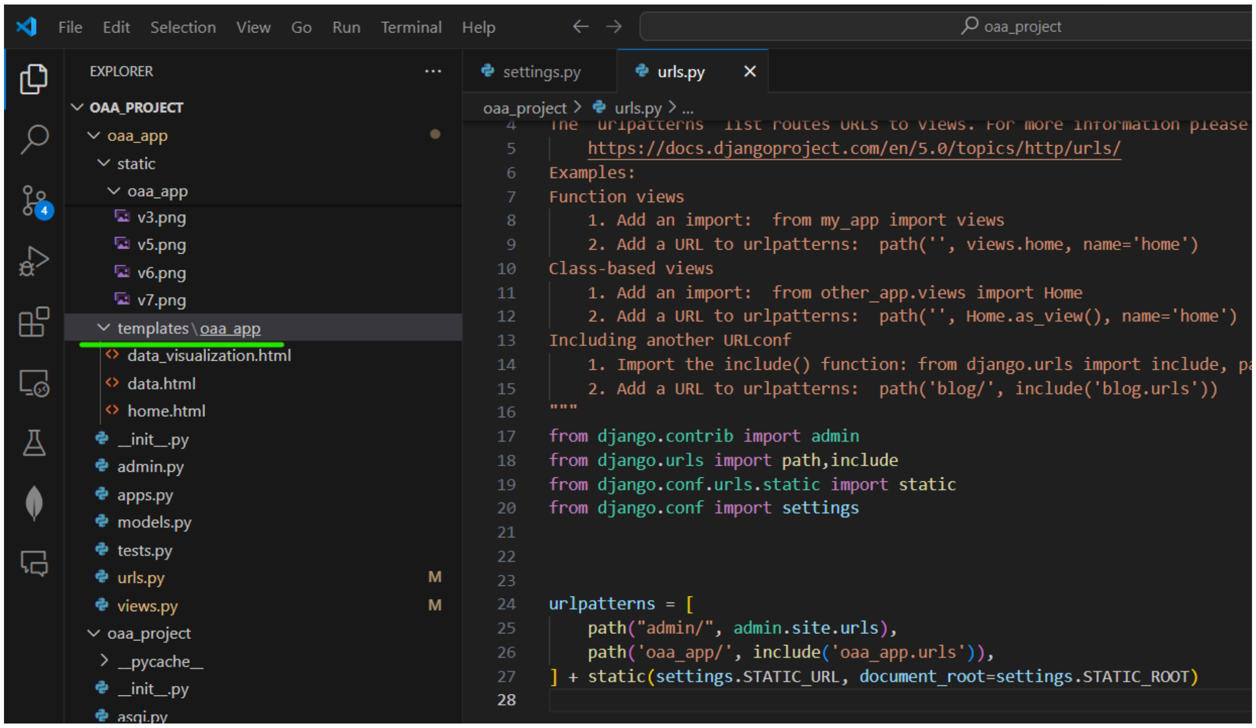
- **Static folder:** This directory houses all static files such as images, CSS, and JavaScript files utilized in the project.


File Edit Selection View Go Run Terminal Help
oaa_project
EXPLORER
settings.py
urls.py
oaa_project > urls.py > ...

```
OAA_PROJECT
  oaa_app
    __pycache__
    migrations
    static
      admin
      oaa_app
        chintan.jpg
        data.jpg
        graduate.jpg
        mitali.jpg
        r4.webp
        r8.jpg
        rutva.jpg
        smile.png
        study.jpg
        v1.png
        v2.png
        v3.png
        v5.png
        v6.png
        v7.png
    templates
    __init__.py
    admin.py
```

```python
"""
URL configuration for oaa_project project.

The `urlpatterns` list routes URLs to views. For more information plea
    https://docs.djangoproject.com/en/5.0/topics/http/urls/
Examples:
Function views
    1. Add an import:  from my_app import views
    2. Add a URL to urlpatterns:  path('', views.home, name='home')
Class-based views
    1. Add an import:  from other_app.views import Home
    2. Add a URL to urlpatterns:  path('', Home.as_view(), name='home
Including another URLconf
    1. Import the include() function: from django.urls import include
    2. Add a URL to urlpatterns:  path('blog/', include('blog.urls'))
"""
from django.contrib import admin
from django.urls import path,include
from django.conf.urls.static import static
from django.conf import settings



urlpatterns = [
    path("admin/", admin.site.urls),
    path('oaa_app/', include('oaa_app.urls')),
] + static(settings.STATIC_URL, document_root=settings.STATIC_ROOT)

```

- **Template folder:** This directory contains all HTML template files used to render different pages of the web application. These templates are dynamically rendered with data from the views.


File Edit Selection View Go Run Terminal Help
oaa_project
EXPLORER
settings.py
urls.py
oaa_project > urls.py > ...

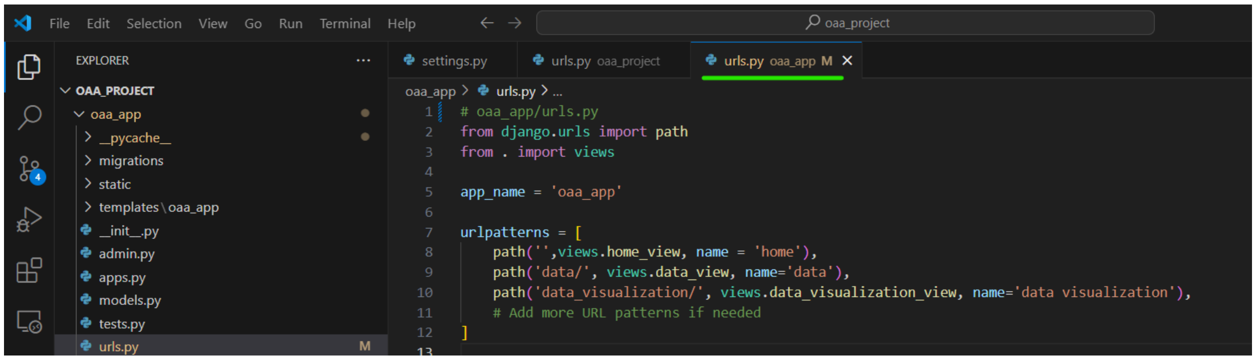
```
OAA_PROJECT
  oaa_app
    static
      oaa_app
        v3.png
        v5.png
        v6.png
        v7.png
    templates \ oaa_app
      data_visualization.html
      data.html
      home.html
    __init__.py
    admin.py
    apps.py
    models.py
    tests.py
    urls.py          M
    views.py         M
  oaa_project
    __pycache__
    __init__.py
    asgi.py
```

```python
The `urlpatterns` list routes URLs to views. For more information please
    https://docs.djangoproject.com/en/5.0/topics/http/urls/
Examples:
Function views
    1. Add an import:  from my_app import views
    2. Add a URL to urlpatterns:  path('', views.home, name='home')
Class-based views
    1. Add an import:  from other_app.views import Home
    2. Add a URL to urlpatterns:  path('', Home.as_view(), name='home')
Including another URLconf
    1. Import the include() function: from django.urls import include, pa
    2. Add a URL to urlpatterns:  path('blog/', include('blog.urls'))
"""
from django.contrib import admin
from django.urls import path,include
from django.conf.urls.static import static
from django.conf import settings



urlpatterns = [
    path("admin/", admin.site.urls),
    path('oaa_app/', include('oaa_app.urls')),
] + static(settings.STATIC_URL, document_root=settings.STATIC_ROOT)

```

oaa_app Configuration:

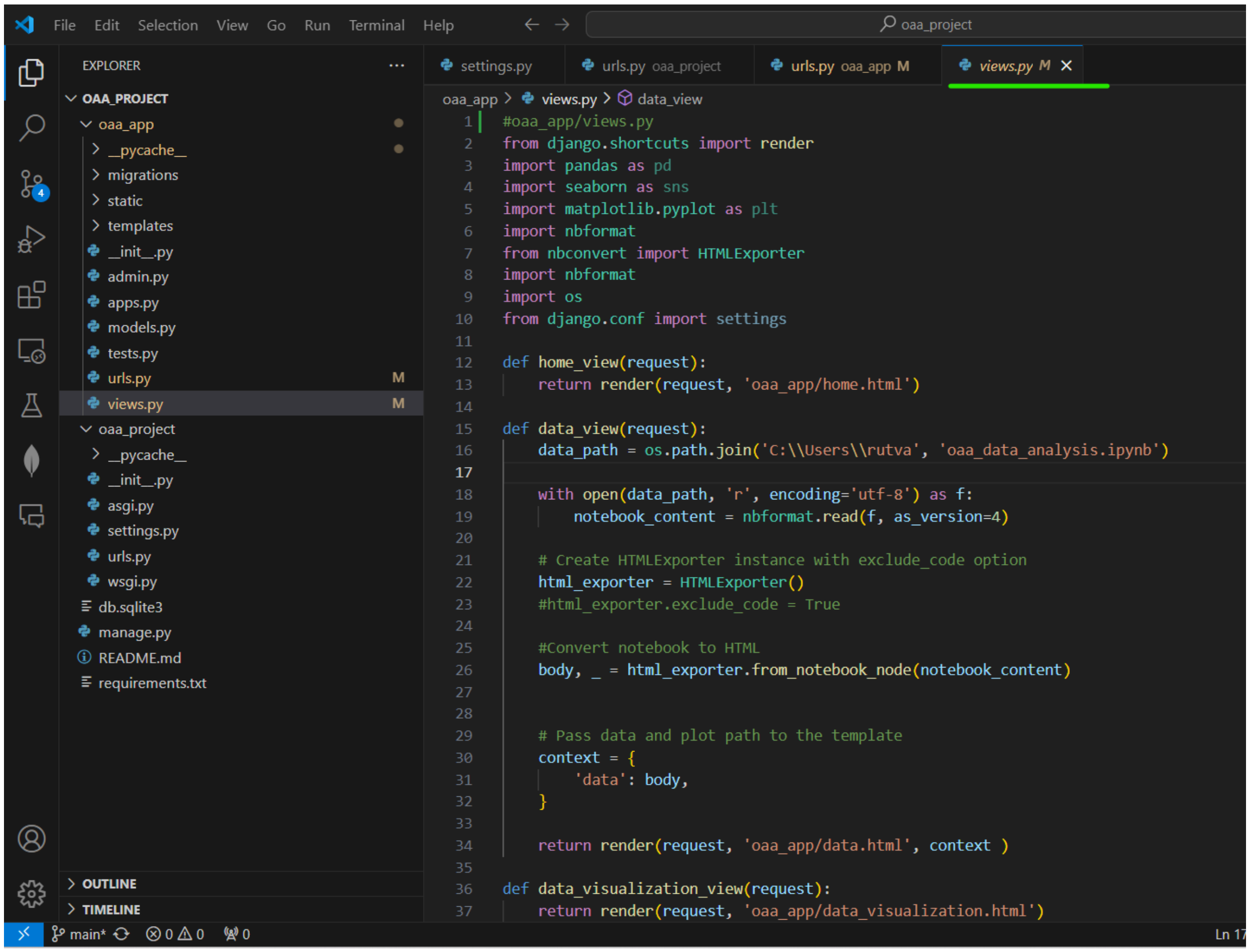
The oaa_app directory within the project contains application-specific files:

- **urls.py:** This file defines URL patterns specific to the oaa_app application, mapping them to corresponding views.



```python
# oaa_app/urls.py
from django.urls import path
from . import views

app_name = 'oaa_app'

urlpatterns = [
    path('',views.home_view, name = 'home'),
    path('data/', views.data_view, name='data'),
    path('data_visualization/', views.data_visualization_view, name='data visualization'),
    # Add more URL patterns if needed
]
```

- **views.py:** This file holds the view functions responsible for processing requests and returning responses for different application URLs. It includes functions like home_view(), data_view(), and data_visualization_view(), each rendering a specific HTML template.



```python
#oaa_app/views.py
from django.shortcuts import render
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import nbformat
from nbconvert import HTMLExporter
import nbformat
import os
from django.conf import settings

def home_view(request):
    return render(request, 'oaa_app/home.html')

def data_view(request):
    data_path = os.path.join('C:\\Users\\rutva', 'oaa_data_analysis.ipynb')

    with open(data_path, 'r', encoding='utf-8') as f:
        notebook_content = nbformat.read(f, as_version=4)

    # Create HTMLExporter instance with exclude_code option
    html_exporter = HTMLExporter()
    #html_exporter.exclude_code = True

    #Convert notebook to HTML
    body, _ = html_exporter.from_notebook_node(notebook_content)


    # Pass data and plot path to the template
    context = {
        'data': body,
    }

    return render(request, 'oaa_app/data.html', context )

def data_visualization_view(request):
    return render(request, 'oaa_app/data_visualization.html')
```

Review Code:

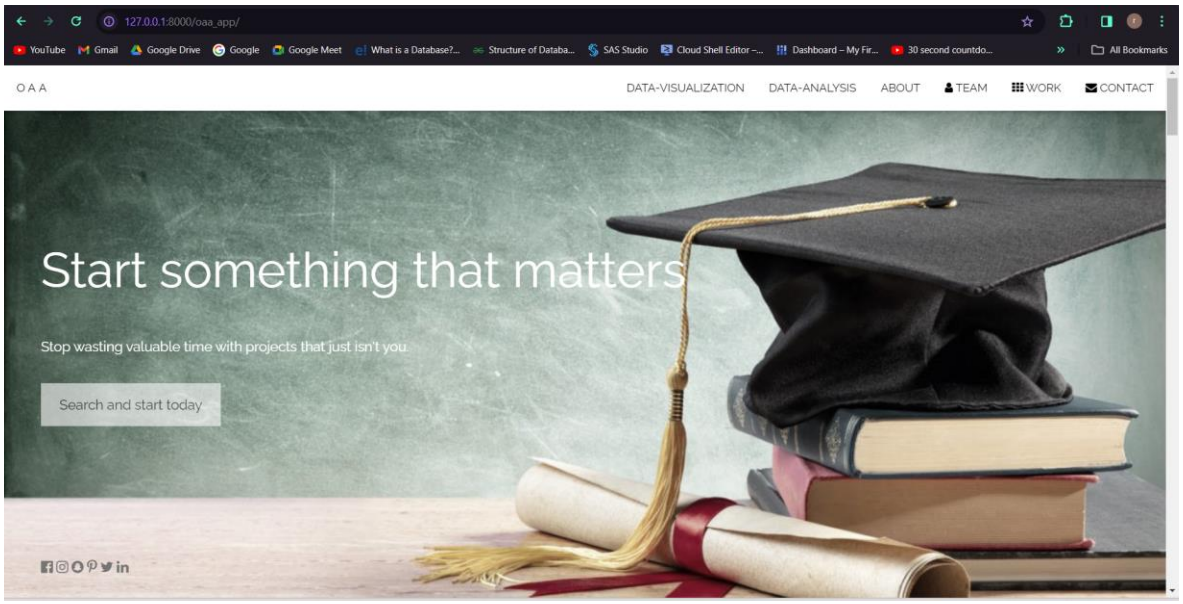
To review the complete codebase, visit our GitHub repository: https://github.com/rutva1605/oaa

Review All Documents:

https://github.com/rutva1605/oaa_project

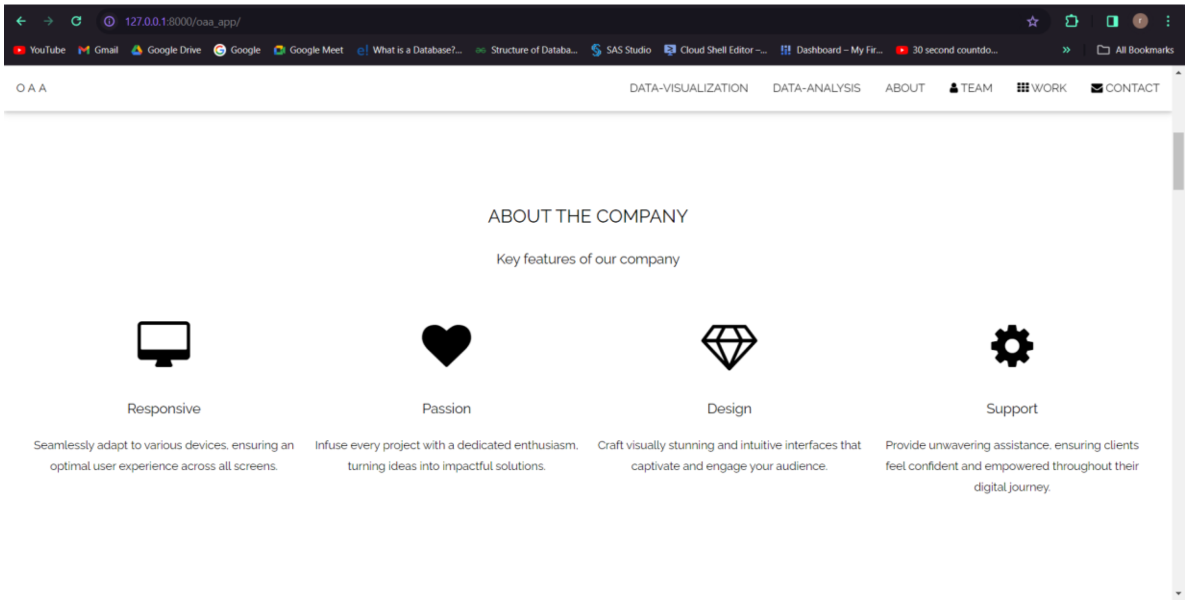
Final Website View:

Home Page:



Start something that matters

Stop wasting valuable time with projects that just isn't you.



ABOUT THE COMPANY

Key features of our company

Responsive

Seamlessly adapt to various devices, ensuring an optimal user experience across all screens.

Passion

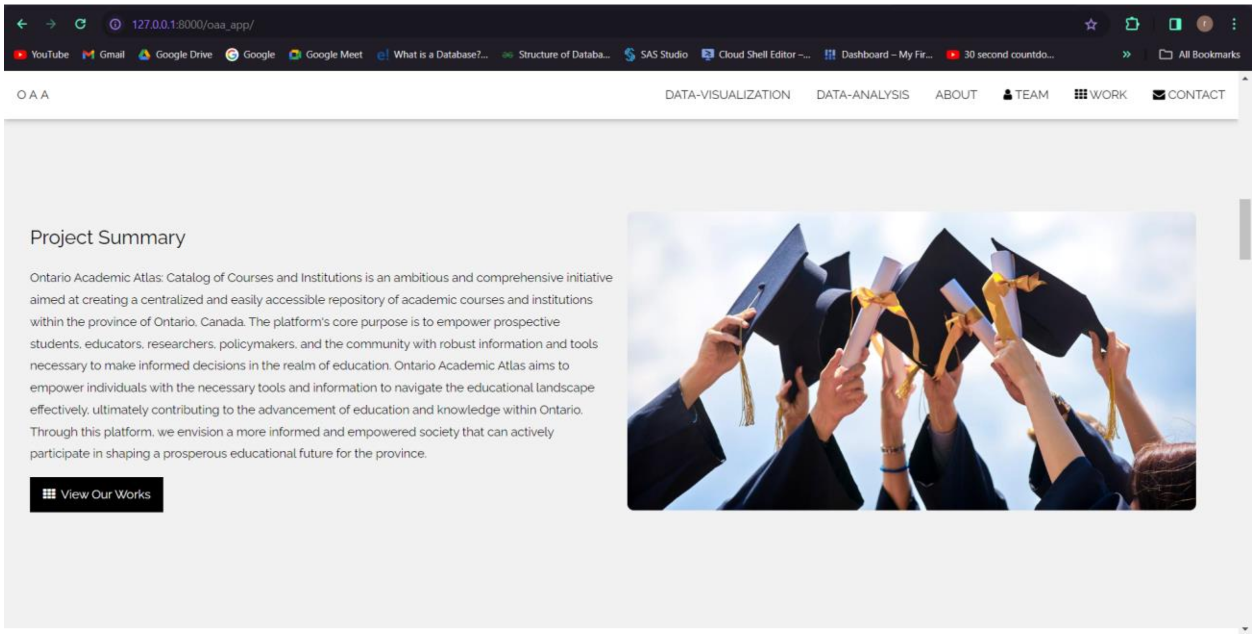
Infuse every project with a dedicated enthusiasm, turning ideas into impactful solutions.

Design

Craft visually stunning and intuitive interfaces that captivate and engage your audience.

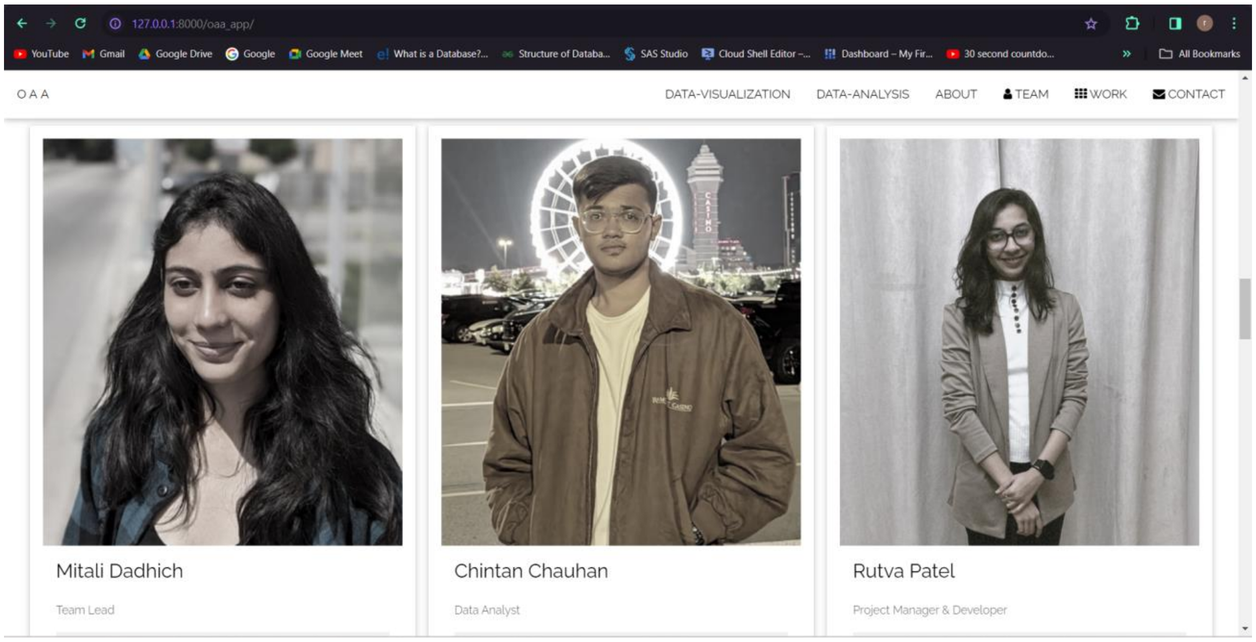
Support

Provide unwavering assistance, ensuring clients feel confident and empowered throughout their digital journey.



Project Summary

Ontario Academic Atlas: Catalog of Courses and Institutions is an ambitious and comprehensive initiative aimed at creating a centralized and easily accessible repository of academic courses and institutions within the province of Ontario, Canada. The platform's core purpose is to empower prospective students, educators, researchers, policymakers, and the community with robust information and tools necessary to make informed decisions in the realm of education. Ontario Academic Atlas aims to empower individuals with the necessary tools and information to navigate the educational landscape effectively, ultimately contributing to the advancement of education and knowledge within Ontario. Through this platform, we envision a more informed and empowered society that can actively participate in shaping a prosperous educational future for the province.



Mitali Dadhich

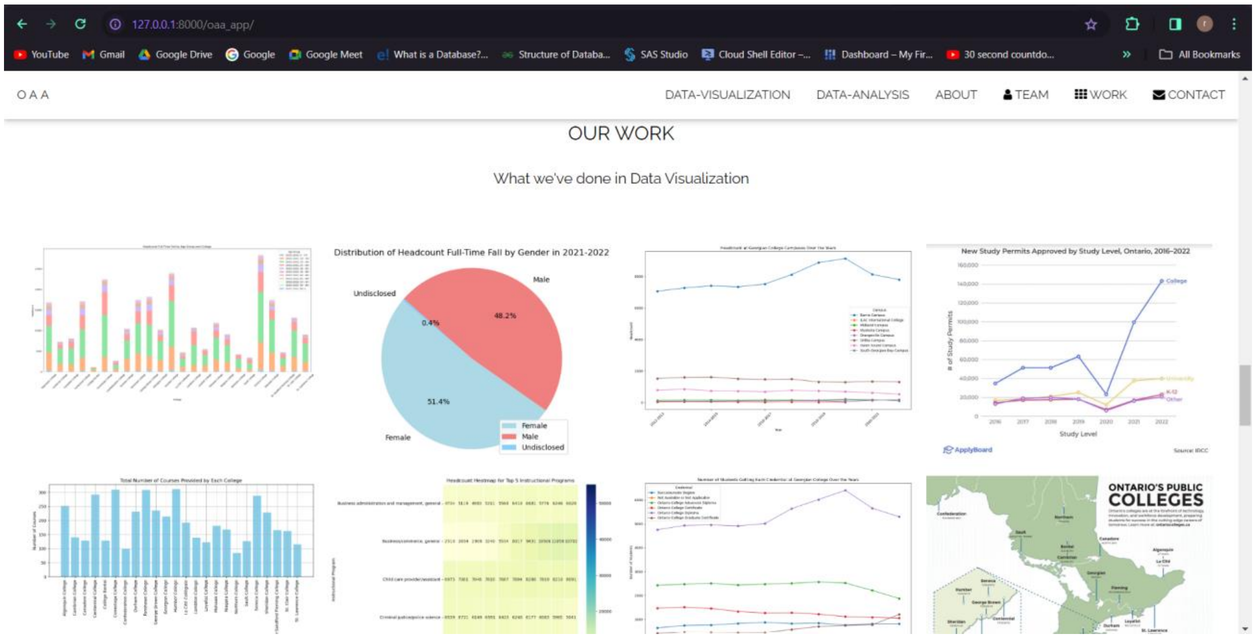
Team Lead

Chintan Chauhan

Data Analyst

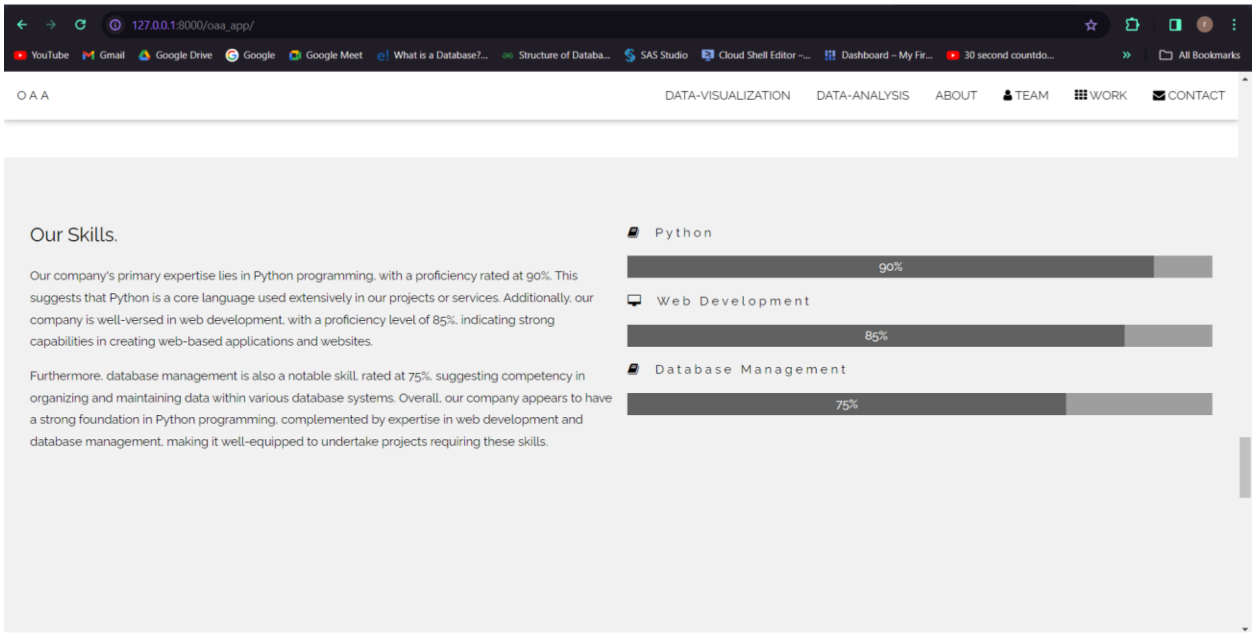
Rutva Patel

Project Manager & Developer



OUR WORK

What we've done in Data Visualization



Our Skills.

Our company's primary expertise lies in Python programming, with a proficiency rated at 90%. This suggests that Python is a core language used extensively in our projects or services. Additionally, our company is well-versed in web development, with a proficiency level of 85%, indicating strong capabilities in creating web-based applications and websites.

Furthermore, database management is also a notable skill, rated at 75%, suggesting competency in organizing and maintaining data within various database systems. Overall, our company appears to have a strong foundation in Python programming, complemented by expertise in web development and database management, making it well-equipped to undertake projects requiring these skills.

Python

90%

Web Development

85%

Database Management

75%

Conclusion:

In conclusion, the oaa_project is a Django web application tailored to provide educational resources for students and educators in Ontario. Throughout this development document, we have outlined the setup and configuration of the project, including initialization, settings.py configuration, URL configuration, and the structure of the project directories.

By utilizing the django-admin startproject command, we initialized the oaa_project, establishing the necessary directory structure and generating essential configuration files. In the settings.py file, we configured various settings such as DEBUG mode, installed apps, database settings, and static files settings to ensure smooth development and deployment.

The URL configuration in the urls.py file maps URL patterns to views, enabling the creation of custom endpoints for the project. Additionally, we discussed the project directory structure, highlighting the static and template folders, which contain static files and HTML templates, respectively.

Furthermore, we provided insights into the oaa_app configuration, detailing the specific files within the application directory responsible for defining URL patterns and view functions.

To facilitate code review and collaboration, the complete codebase of the oaa_project is available on our GitHub repository. This document serves as a comprehensive guide for developers contributing to the project and stakeholders interested in understanding its architecture and configuration.