



Sudarshangjuri- Assignment – User Documentation

1 Project Overview

Project Name: Sudarshangjuri- Assignment

Tech Stack:

- **Frontend:** ReactJS + Vite + Google OAuth
- **Backend:** Python + Django + Django REST Framework (DRF)
- **Web Scraping:** BeautifulSoup + Requests
- **Authentication:** Google OAuth 2.0 + JWT tokens
- **Database:** SQLite

Purpose:

A web app to:

1. Sign in securely using Google OAuth
2. Enter a website URL to scrape headlines/titles
3. Display scraped data on a responsive dashboard
4. Optional: Refresh data in real-time

GOOGLE_CLIENT_ID= 678138623677-ld07o0f665fs7eecbc4hl7hifv2h0h9j.apps.googleusercontent.com

3.Backend Setup (Django + DRF)

3.1 Create Project Folder & Virtual Environment

```
cd Sudarshangjuri- Assignment/backend
python -m venv venv # 🔥 Create virtual environment
# Activate venv:
# Windows
venv\Scripts\activate
# Mac/Linux
source venv/bin/activate
```

Install Dependencies

```
pip install --upgrade pip
pip install django djangorestframework djangorestframework-
simplejwt google-auth requests beautifulsoup4 python-dotenv
django-cors-headers whitenoise gunicorn
pip freeze > requirements.txt
```

Create .env file

```
SECRET_KEY=your_django_secret_key
DEBUG=True
GOOGLE_CLIENT_ID=your-google-client-
id.apps.googleusercontent.com
```

4. Database Setup Run migrations:

```
python manage.py makemigrations
```

```
python manage.py migrate
```

Create a superuser (admin): python manage.py createsuperuser

Run the backend server:

```
python manage.py runserver 8000
```

Frontend Setup

1. Open a new terminal and navigate to frontend:

```
2.cd .../frontend # from project root
```

```
3.npm create vite@latest react-scraper -- --template react
```

```
4.cd react-scraper
```

```
5.npm install
```

```
6.# install additional deps
```

```
7.npm install axios @react-oauth/google react-router-dom
```

Create frontend/react-scraper/.env :

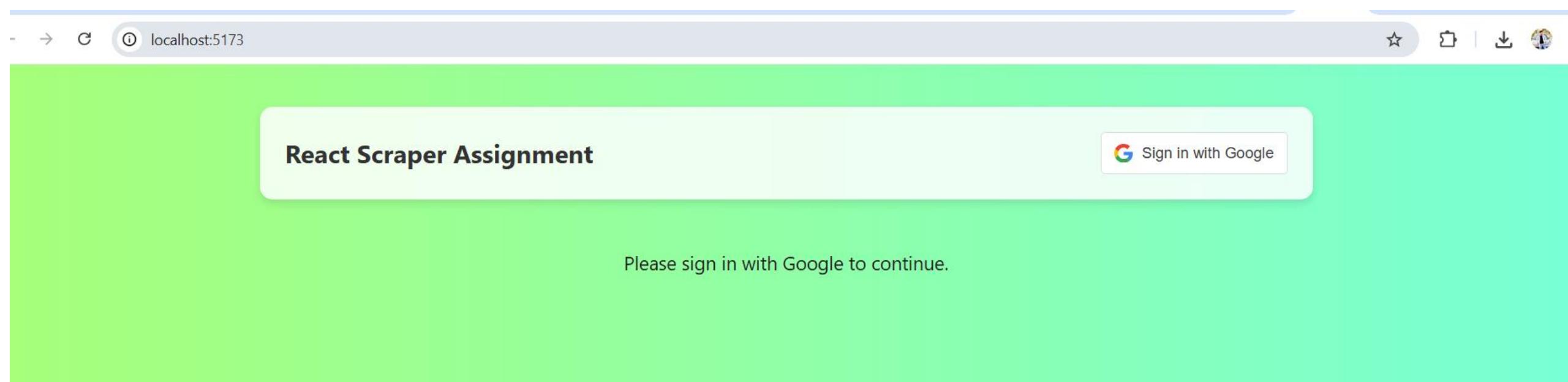
VITE_GOOGLE_CLIENT_ID=your-google-client-id.apps.googleusercontent.com

VITE_API_BASE_URL=http://localhost:8000

5. Using the Application

5.1 Login with Google

1. Open the frontend URL (<http://localhost:5173>)
2. Click on the **Google Sign-in button**.
3. Accept consent in the Google pop-up.
4. On success, you are redirected to the dashboard.



5.2 Scraping a Website

1.In the dashboard, enter the **URL** of a public website you want to scrape.

•Click **Fetch Latest**.

•Wait for the spinner/loading indicator to complete.

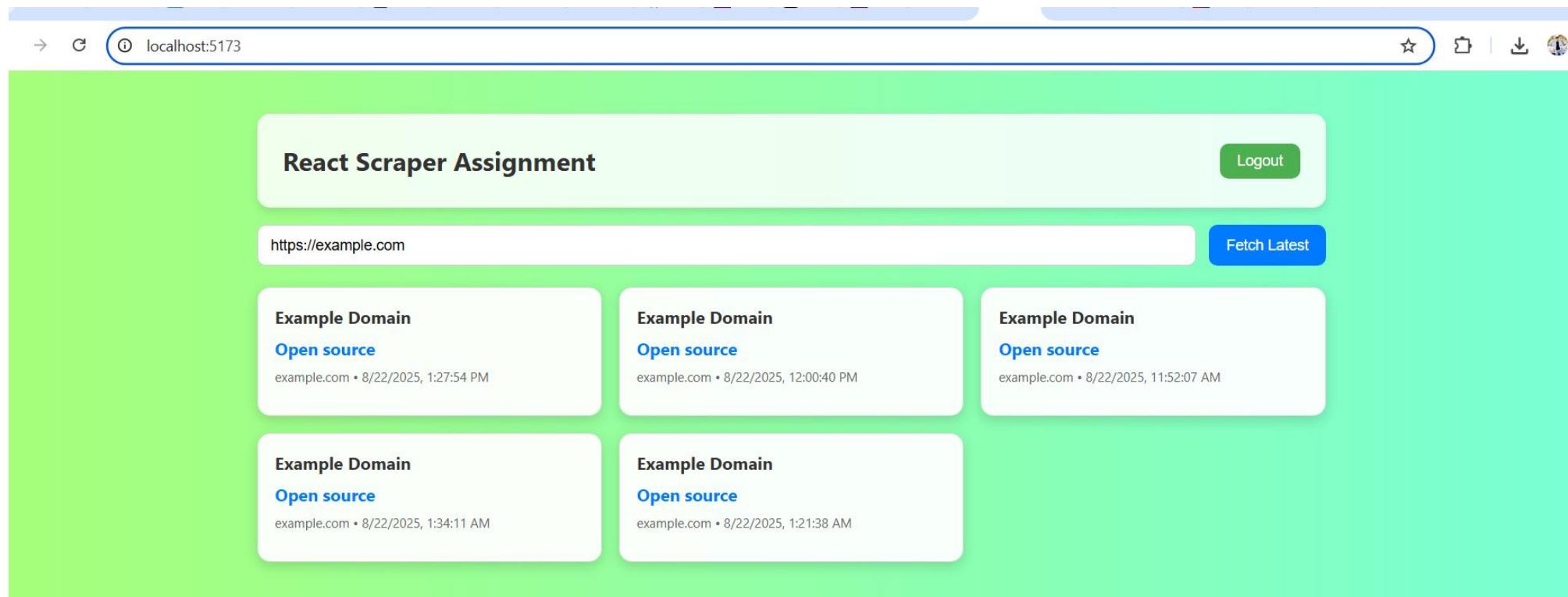
•Title

•URL link

•Source

•Timestamp

•The scraped data will appear as **cards** showing:



5.3 Viewing Previously Scraped Items

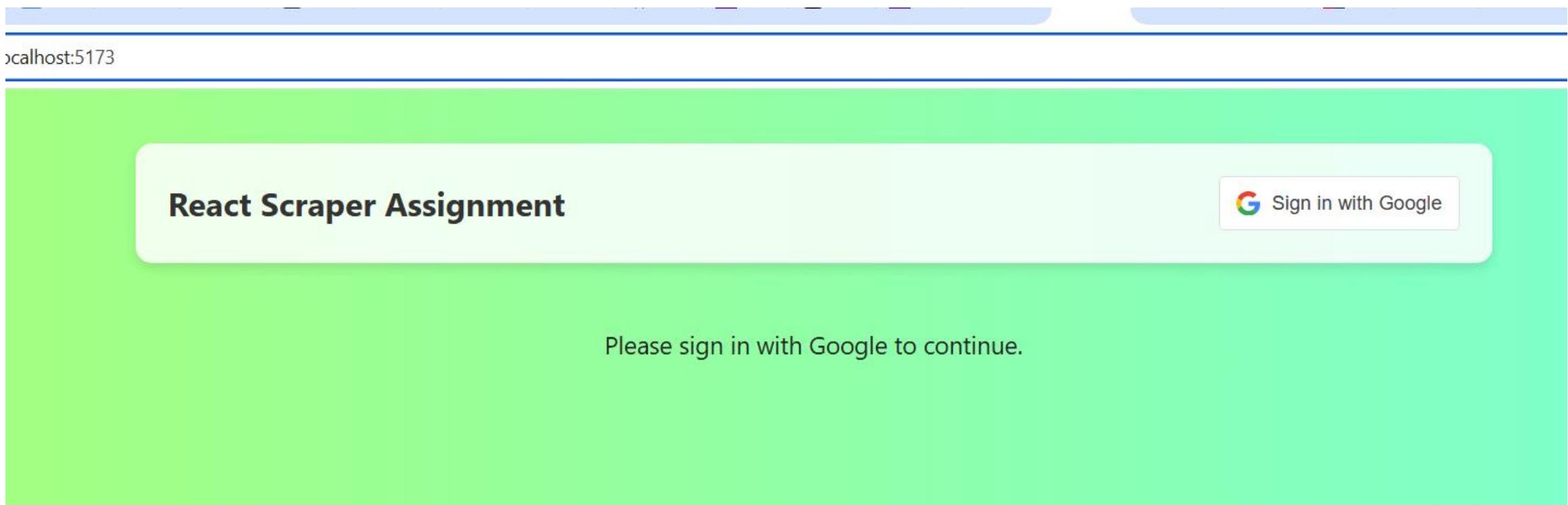
- 1.The dashboard automatically shows the last 200 items fetched from any website.
- 2.Scroll through cards to see all previous scrapes.

The screenshot displays a web browser window with the URL `localhost:5173` in the address bar. The page title is "React Scraper Assignment". On the left, there is a search bar containing the URL `https://example.com` and a blue button labeled "Fetch Latest". The main content area contains five white cards, each representing a scraped item from the example domain. Each card has a title "Example Domain", a link "Open source", and a timestamp. The cards are arranged in two rows: three in the top row and two in the bottom row.

Card 1	Card 2	Card 3	Card 4	Card 5
Example Domain Open source example.com • 8/22/2025, 1:27:54 PM	Example Domain Open source example.com • 8/22/2025, 12:00:40 PM	Example Domain Open source example.com • 8/22/2025, 11:52:07 AM		
Example Domain Open source example.com • 8/22/2025, 1:34:11 AM	Example Domain Open source example.com • 8/22/2025, 1:21:38 AM			

5.4 Logout

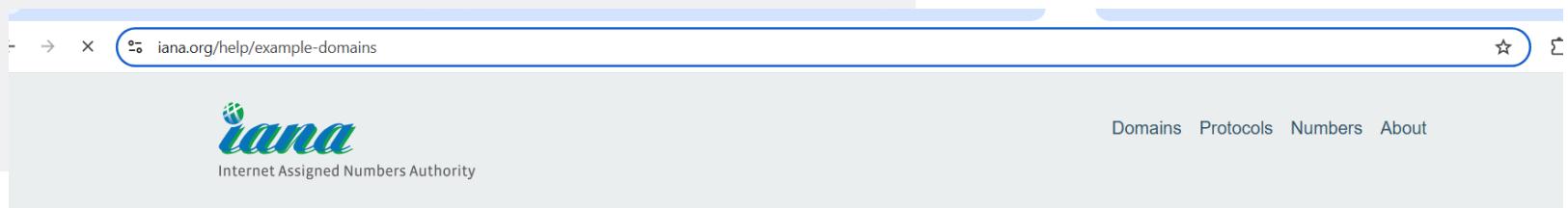
- 1.Click the **Logout** button in the header.
- 2.You will be logged out and returned to the login screen.



Example Domain

This domain is for use in illustrative examples in documents. You may use this domain in literature without prior coordination or asking for permission.

[More information...](#)



Example Domains

As described in [RFC 2606](#) and [RFC 6761](#), a number of domains such as example.com and example.org are maintained for documentation purposes. These domains may be used as illustrative examples in documents without prior coordination with us. They are not available for registration or transfer.

We provide a web service on the example domain hosts to provide basic information on the purpose of the domain. These web services are provided as best effort, but are not designed to support production applications. While incidental traffic for incorrectly configured applications is expected, please do not design applications that require the example domains to have operating HTTP service.

Further Reading

- [IANA-managed Reserved Domains](#)

Last revised 2017-05-13.