Odisha RERA Project Scraper – Functional Documentation

📌 Purpose

This Python-based Streamlit application scrapes real estate project information from the **Odisha RERA (Real Estate Regulatory Authority)** website and presents it in a visually appealing dashboard. It allows users to:

- View a specified number of real estate projects
- Choose between a futuristic card view or a traditional styled table
- Download the data as a CSV file

⚙️ Functional Overview

1. Dashboard Interface

Technology Used:

- Streamlit – for building an interactive UI
- CSS Styling – for a futuristic look and feel
- Google Fonts – for Orbitron font animation and style

Purpose:

- Set up a responsive, web-based dashboard for displaying project data.

2. Sidebar Options

Functionality:

- `num_projects`: Slider that lets users select how many projects to fetch (1–30)
- `view_mode`: Option to select display style – `Hologram Cards` or `Colorful Table`

Purpose:

Allow users to customize data fetch quantity and view mode.

3. Web Scraping Logic

Function: `scrape_rera_projects(num_projects: int) -> pd.DataFrame`

Libraries Used:

- `selenium` – to automate browser interactions and extract dynamic data
- `pandas` – to store and process the scraped data

Steps Performed:

1. **Initialize Selenium WebDriver** in headless mode for efficiency.
2. **Navigate to the Odisha RERA portal** and wait for page elements.
3. **Iterate through projects**:
    - Click on "View Details"
    - Extract:
        - RERA Registration Number
        - Project Name
        - Promoter Name
        - Promoter Address
        - GST Number
    - Navigate back to the listing
4. Handle exceptions and alternate field labels (e.g., `Company Name` vs. `Proprietory Name`)
5. **Return data as a DataFrame**

Purpose:
Dynamically scrape a specified number of Odisha RERA project listings and return structured information.

4. Data Display Section

View Modes:

- **Hologram Cards:**
    - Styled using custom HTML and CSS
    - Shows each project in a separate futuristic "card"
- **Colorful Table:**
    - Uses Streamlit's `dataframe` with conditional formatting
    - Each column styled for readability and emphasis

Function: `style_table(df: pd.DataFrame) -> Styler`

Purpose:
Provide clear, user-friendly ways to visualize the project data.

5. CSV Export Option

Streamlit Feature Used: `st.download_button`

Functionality:

- Converts DataFrame to CSV
- Lets users download the file as `odisha_rera_projects.csv`

Purpose:
Enable users to save or share the data offline.

📂 Data Fields

Field	Description
RERA Regd. No	Unique registration number assigned by RERA
Project Name	Official name of the real estate project
Promoter Name	Company or individual responsible for the project

Promoter Address	Registered office or residence of the promoter
GST No.	Government-issued Goods and Services Tax identification

📥 Use Case Scenario

An end user, such as a regulatory analyst, property buyer, or journalist, can launch the dashboard to view and download current project data from the RERA portal for analysis or reporting.

📎 Summary of Technologies

Component	Technology Used
UI & Dashboard	Streamlit
Web Scraping	Selenium WebDriver
Data Handling	Pandas
Styling	Custom HTML, CSS, Google Fonts

✅ Benefits

- **No manual browsing** of the Odisha RERA website
- **Instant export** of structured, usable project data
- **User-friendly interface** for both technical and non-technical users
- **Visual choices** for both casual viewers and data analysts