# Real Estate Data Collection and Cleaning Project

## Important Notes (Read Carefully)

* **Individual Project** – This is strictly an individual assignment. Collaboration is not allowed.
* **Internet Use** – You may search for concepts or documentation online. However, **copying code directly from external sources is prohibited**.
* **Evaluation Format** – You will participate in a **one-on-one review**, where you must explain your process, decisions, and code functionality.
* **Timeline**
  + **Start:** Wednesday morning
  + **Deadline:** End of day Thursday
* **Early Submission** – If you complete and push your project to GitHub early, you are free to leave the workspace.

## Project Objective

This project is designed to simulate a real-world data science task. You will scrape property data from two real estate websites, clean and integrate the data, engineer features, and frame a predictive modeling challenge based on pricing.

The final deliverable will be a GitHub repository containing all project artifacts.

## Assigned Websites

You will be assigned two real estate websites. Each site contains listings for properties for rent and properties for sale.

* Website 1: https://www.dubizzle.com.om/en/properties/
* Website 2: https://om.opensooq.com/en/property

Your task is to collect data relevant to the pricing and characteristics of listed **properties for sale**.

## Project Requirements

### 1. Web Scraping

* Scrape relevant data (e.g., property title, location, number of rooms, price, size, listing type).
* Save the raw data in a structured format (e.g., CSV or JSON).
* Handle pagination and dynamic content if necessary.

### 2. Data Cleaning & Integration

* Use **Python functions** to clean the data (missing values, inconsistent formatting, duplicates, etc.).
* Combine data from both websites into **a single cleaned CSV file**.
* Ensure consistency in column naming and data types.

### 3. Feature Engineering

* Generate new features to support later modeling (e.g., price per square meter, total rooms, encoded categorical values).
* Apply appropriate **feature scaling** (e.g., MinMaxScaler, StandardScaler).

### 4. Predictive Modeling (Challenge)

* Define a **modeling objective**, such as predicting property price.
* Prepare your dataset accordingly (split features/target, handle categorical variables, etc.).
* Suggest and briefly implement **Scikit-learn models** suitable for the task (e.g., Linear Regression, Decision Tree, Random Forest).

This is a bonus challenge. It is not required but will earn additional credit if well-executed.

### 5. GitHub Repository

Each student must create a **public GitHub repository** with the following:

* Web scraping scripts or notebooks
* Data cleaning functions
* Final combined CSV file
* Feature engineering and modeling code
* A brief README.md file explaining:
  + Your objectives
  + Websites used
  + Steps taken in data collection and cleaning
  + Feature engineering strategy
  + Modeling approach (if applicable)

## Deliverables Summary

| Deliverable | Format |
| --- | --- |
| Raw scraped data | CSV or JSON |
| Cleaning code | .py script or notebook |
| Final cleaned & combined dataset | CSV |
| Feature engineering & scaling | Notebook or script |
| (Optional) Modeling code | Notebook or script |
| GitHub repository with all files | Public URL |

## Evaluation Criteria

| Component | Weight |
| --- | --- |
| **Web Scraping** | 40% |
| **Data Cleaning & Integration** | 30% |
| **Feature Engineering** | 20% |
| **Code Quality & GitHub Setup** | 10% |
| **Challenge Task (Modeling)** | +15% Bonus |

The bonus challenge will **not affect your core score** but can significantly boost your final evaluation if completed thoughtfully.

## Final Reminders

* Structure your code professionally and document your functions.
* Keep your GitHub repo clean and organized.
* Be prepared to explain your choices in the evaluation interview.