Assignment: Data Wrangling, Visualization, and Web Presentation

Data Wrangling and Visualization Course

February 17, 2025

1 Introduction

In this assignment, you will integrate multiple skills: parsing real-world data, designing and populating a database, and building a modern, interactive web page to visualize your data. This task is designed to test your ability to extract meaningful information from unstructured sources and then communicate that information through a clean, responsive user interface. We expect deep thought in both your design decisions and implementation strategies.

2 Assignment Overview

You are tasked with the following major components:

1. **Data Extraction and Database Creation:** Parse a Wikipedia page, extract relevant data, and store the information in a relational database.
2. **Web Page Development and Hosting:** Create an interactive and visually appealing web page using HTML, CSS, and JavaScript to display the extracted data. This page must be hosted on GitHub Pages.

3 Part 1: Data Extraction and Database Design

3.1 Topic Selection

For this assignment, you will extract data from the Wikipedia page on **Highest-Grossing Films**. Use the following URL as your primary data source:

`https://en.wikipedia.org/wiki/List_of_highest-grossing_films`

3.2 Data Extraction Requirements

Your task is to parse the target Wikipedia page and extract the following information for each film listed:

- **Film Title** (string)
- **Release Year** (integer)
- **Director(s)** (string or list of strings)
- **Box Office Revenue** (numeric value; you may include currency symbols as necessary)
- **Country of Origin** (string)
- **Additional Attributes (Optional):** You may also extract other data (e.g., genre, production company) if available.

3.3 Database Structure

You have to store the data in a relational database. You may use SQLite, MySQL, PostgreSQL, or MongoDB. Below is the schema using a single table design.

Table: `films`

- `id` — **INTEGER PRIMARY KEY AUTOINCREMENT**: Unique identifier for each film.
- `title` — **TEXT NOT NULL**: The title of the film.
- `release_year` — **INTEGER**: Year of release.
- `director` — **TEXT**: Name(s) of the director(s).
- `box_office` — **REAL** or **TEXT**: Box office revenue (consider data cleaning if using currency symbols).
- `country` — **TEXT**: Country of origin.

3.4 Implementation Instructions

1. Develop a Python script within a Jupyter Notebook that uses libraries such as `BeautifulSoup`, `requests`, or `Scrapy` to parse the Wikipedia page.
2. Clean and structure the data appropriately.
3. Insert the cleaned data into your chosen database.
4. Document your code with clear comments and markdown cells explaining your approach.

4 Part 2: Web Page Development and GitHub Pages Hosting

4.1 Web Page Requirements

Design and develop a web page that presents a subset of the information from your database. The web page should:

- Be visually appealing with a modern design using HTML5 and CSS3.
- Use JavaScript to dynamically display and manipulate data (e.g., filtering, sorting).
- Present key information such as film titles, release years, directors, and box office numbers.

4.2 Data Integration Strategy

Since GitHub Pages is static hosting and does not support server-side code, you should:

1. Export your database content to a JSON file.
2. Use JavaScript (or a library such as `fetch API`) to load and display the JSON data.

4.3 GitHub Pages Deployment Instructions

Follow these steps to host your webpage on GitHub Pages:

1. **Repository Creation:** Create a new repository on GitHub (e.g., `highest-grossing-films`).
2. **Add Files:** Commit your HTML, CSS, and JavaScript files to the repository.
3. **Enable GitHub Pages:**
   (a) Go to your repository's `Settings`.
   (b) Scroll down to the **GitHub Pages** section.
   (c) Under **Source**, select the branch you want to deploy (typically `main`) and choose the root folder (or `docs/` if you prefer).
   (d) Click **Save**. GitHub will provide a URL (e.g., `https://<username>.github.io/<repository>`).
4. **Verify:** Open the provided URL in your browser to ensure your page is live.

5 Submission Requirements

You must submit the following:

1. A link to the **hosted webpage** (the GitHub Pages URL).
2. A link to the **GitHub repository** containing your front-end code (HTML, CSS, JavaScript).
3. The **Jupyter Notebook** containing your Python code for parsing and crawling the Wikipedia page and populating your database.

6 Evaluation Criteria

Your work will be evaluated based on:

- **Database Design 30**
- **Wikipedia Parsing 20**
- **Simple Web Page Design 30**
- **Deployment 5**
- **Interactive features in the webpage 15**

Good luck and happy coding!