

Court Data Fetcher & Mini-Dashboard

Project Overview

This project is a small web application that allows users to search for court case details from the **Delhi High Court** website by entering the case type, case number, and year. It programmatically requests the official court site, assists manual CAPTCHA solving, extracts available case details, stores the data in a PostgreSQL database, and displays the results in a user-friendly web interface.

Features

- User-friendly form for entering case search parameters.
- Automated browser session to navigate to the official Delhi High Court website and autofill the search form.
- Manual CAPTCHA handling: The browser opens for the user to solve CAPTCHA before scraping proceeds.
- Scrapes the following case details:
 - Parties' names
 - Next hearing date (or last hearing date if next date is not available)
 - Case status
- Stores search parameters and scraped data in PostgreSQL.
- Displays the extracted case data in the frontend results page.

Technical Details

- Backend framework: FastAPI
- Database: PostgreSQL (using SQLAlchemy ORM)
- Scraper: Playwright (with headless browser disabled for manual CAPTCHA solving)
- Templates: Jinja2 for HTML rendering

CAPTCHA Handling Approach

Due to legal and technical restrictions around CAPTCHA automation:

- The Playwright browser launches with a visible window.
- User manually solves the CAPTCHA on the official website.
- After successful CAPTCHA submission, the user confirms via terminal input to continue scraping.
- This manual step is necessary to respect the site's terms and avoid automated CAPTCHA breaking.

Data Extraction

The scraper extracts data from the search results table, including:

- **Parties' Names:** Extracted from the "Petitioner Vs. Respondent" column.
- **Next Hearing Date:** Extracted by parsing "NEXT DATE" or fallback "Last Date" from the listing info.
- **Case Status:** From the case number/status column.

Known Limitations

- **Filing Date:** The official Delhi High Court search page does not display the filing date; hence this data is not extracted.
- **Order/Judgment PDF Links:** The search results page does not provide direct PDF links to orders or judgments, so this data is not available.
- To access full orders or judgments PDFs, additional scraping of detailed case pages may be required, which is beyond the current scope.

Setup Instructions

1. **Clone the repository**
2. **Create and activate a virtual environment**

```
python -m venv venv
```

```
source venv/bin/activate # Linux/macOS
```

```
venv\Scripts\activate # Windows
```

3. **Install dependencies**

```
pip install -r requirements.txt
```

Requirements:

```
fastapi
uvicorn
jinja2
sqlalchemy
asyncpg
python-dotenv
playwright
psycpg2-binary
python-multipart
httpx pytest
```

4. Set up PostgreSQL database

- Create a database named court_data.
- Update the .env file with your PostgreSQL connection URL,
- e.g.
DATABASE_URL=postgresql://username:password@localhost:5432/court_data

5. Run the FastAPI app

```
uvicorn main:app --reload
```

6. Access the app

Open your browser at [http://127.0.0.1:8000/form\(my local_host\)](http://127.0.0.1:8000/form(my local_host)) to search for case details.

How to Use

- Fill the case type, number, and year in the search form.
- On submission, the Playwright browser opens the official court site and autofills your inputs.
- Manually solve the CAPTCHA and submit the form in the opened browser.
- Once submitted, press Enter in the terminal to continue scraping.
- Scraped data will be saved in the PostgreSQL database and displayed on the results page.

Database Schema

Table: court_queries

Column	Type	Description
id	Integer	Primary key
case_type	String	Case type (e.g., W.P.(C), BAIL)
case_number	String	Case number
case_year	String	Case year
diary_no_status	String	Diary number and case status
petitioner_vs_respondent	String	Parties involved in the case
listing_date_court_no	String	Next hearing date / listing info

Future Improvements

- Automate CAPTCHA solving if legally allowed.
- Extract and save filing dates if available.
- Scrape detailed case pages for order/judgment PDF links.
- Add authentication and user management for the web app.