# Court-Data Fetcher & Mini-Dashboard (Delhi High Court Scraper)

## About This Project

This project was developed as part of a technical assignment from Internshala.

**The task required building a mini web app capable of:**

- Fetching public case information from an official Indian court
- Handling manual CAPTCHA legally
- Saving results in a PostgreSQL database
- Displaying the case information in a clean, simple HTML dashboard

### The official assignment title was:

**Task — "Court-Data Fetcher & Mini-Dashboard"**

For this assignment, I selected the Delhi High Court website:

— https://delhihighcourt.nic.in/

It is stable, public, structured, and ideal for scraping with Playwright.

## What This Project Demonstrates

- FastAPI backend development
- Web scraping using Playwright
- Manual CAPTCHA handling (legal and required)
- HTML form-based frontend for case search

- PostgreSQL database integration
- Clean result rendering
- Storing raw HTML + search history

# How the System Works (Simple Flow)

1. User opens the web app at:
   http://127.0.0.1:8000/form
2. User fills in the case details (type, number, year).
3. Playwright launches a browser automatically.
4. The scraper autofills the official Delhi High Court search form.
5. Browser waits for the user to **solve CAPTCHA manually**.
6. User submits the form on the website.
7. User presses Enter in the terminal to continue.
8. Scraper extracts:
   - Parties' names
   - Case status
   - Next/Last hearing date
9. Data is:
   - Shown on the results HTML page
   - Stored in PostgreSQL

# Features

## Web Application Features

- Clean HTML form for entering case details
- Auto-navigation and auto-filling of court form
- Manual CAPTCHA step → ensures legal scraping
- Stores:
  - Search parameters

- Extracted case details
- Raw HTML page

# Scraped Case Details

- Petitioner vs Respondent
- Next hearing date (or last hearing if next is unavailable)
- Case status
- Diary number information

# Database Storage

Every search entry is saved in **PostgreSQL** using SQLAlchemy ORM.

# Technical Stack

- 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 (Important)

CAPTCHA is **not bypassed** (as that is illegal and violates terms).
Instead, this system uses:

1. Playwright browser opens visibly
2. User manually enters CAPTCHA
3. Scraper continues after confirmation in terminal

This approach is **safe, legal, and recommended**.

# Data Extraction Details

The scraper reliably extracts:

- **Parties' Names**
- **Next Hearing Date** (or Last Date)
- **Case Status**
- **Diary Number & Listing Info**

# Limitations (From Court Website Itself)

- **Filing Date** is not available on search results page
- **Order/Judgment PDF links** are not provided
- To fetch PDFs, deeper scraping into case-detail pages is required

# Database Schema

## Table: court_queries

| Column | Type | Description |
|---|---|---|
| id | Integer | Primary key |
| case_type | String | e.g., W.P.(C), BAIL |
| case_number | String | Case number |
| case_year | String | Year of filing |
| diary_no_status | String | Diary number + case status |
| petitioner_vs_respondent | String | Parties involved |
| listing_date_court_no | String | Next/Last listing date |

# Setup Instructions

**1. Clone the repository**

https://github.com/NAGASIVA-JALLA/court-data-fetcher.git

**2.Create a virtual environment**

python -m venv venv

venv\Scripts\activate  # Windows

source venv/bin/activate  # Mac/Linux

**3. Install dependencies**

pip install -r requirements.txt

**4. Install Playwright browsers**

playwright install

**5. Configure PostgreSQL**

**Create a database:** court_data

**Update your .env file:**

DATABASE_URL=postgresql://username:password@localhost:5432/court_data

**6. Run FastAPI**

uvicorn main:app

**7.Open the app**

— http://127.0.0.1:8000/form

# How to Use

1. Enter case type, number, and year
2. Submit the form
3. Playwright opens Delhi High Court website
4. Solve CAPTCHA manually
5. Submit the court form
6. Return to terminal and press **Enter**
7. View results on the frontend
8. Data is automatically saved in PostgreSQL

# Future Improvements

- Automate CAPTCHA if legally permitted
- Add case-order/judgment PDF scraper
- Create complete dashboard with charts
- Add user login system
- Error reporting + logs page