Criminal Records Automation Tool

Project Overview

Create an automated tool for criminal records from eclerksla.com and build a Django web application to store and display the scraped data.

Required Technology Stack

- **Backend**: Django Framework
- **Web Scraping**: Selenium WebDriver
- **Database**: SQLite (development) / PostgreSQL (production)
- **Frontend**: HTML, CSS, Bootstrap
- **Python Libraries**: selenium (Undetected_chromedriver)

Task Requirements

Phase 1: Django Project Setup

1. Create new Django project with proper structure
2. Set up virtual environment with required dependencies
3. Configure Django settings for database and installed apps
4. Create a new Django app called "scraper"

Phase 2: Database Design

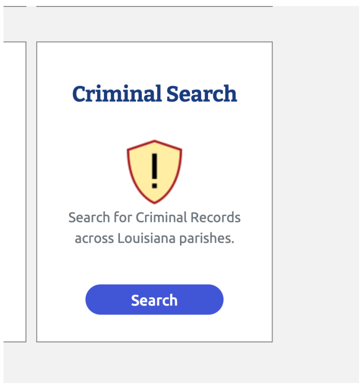
1. Create Django model for criminal records with following fields:
   - Defendant Name
   - Birth Date
   - Sex
   - Race
   - Case Number (unique identifier)
   - Date Filed
   - Charges (text field for multiple charges)
   - Arrest/Citation Date
   - Parish
   - Alert Available (boolean)
   - Scraped timestamp
2. Run migrations to create database tables

Phase 3: Web Scraping Implementation

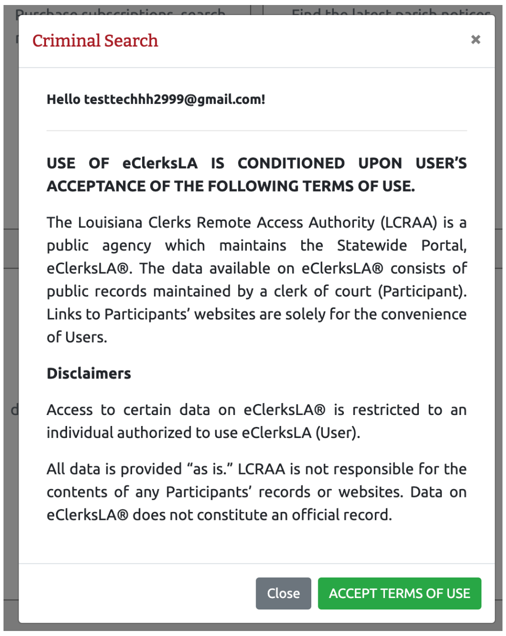
3.1 Selenium Automation Workflow

1. **Initial Navigation**
   - Go to website: https://eclerksla.com/Home
   - Handle any required login process

2. **Criminal Search Access**
   - Locate and click the "Search" button under the Criminal Search section



   - Wait for and handle terms of service popup
   - Click "Accept" on terms popup



Criminal Search

Hello testtechhh2999@gmail.com!

USE OF eClerksLA IS CONDITIONED UPON USER'S ACCEPTANCE OF THE FOLLOWING TERMS OF USE.

The Louisiana Clerks Remote Access Authority (LCRAA) is a public agency which maintains the Statewide Portal, eClerksLA®. The data available on eClerksLA® consists of public records maintained by a clerk of court (Participant). Links to Participants' websites are solely for the convenience of Users.

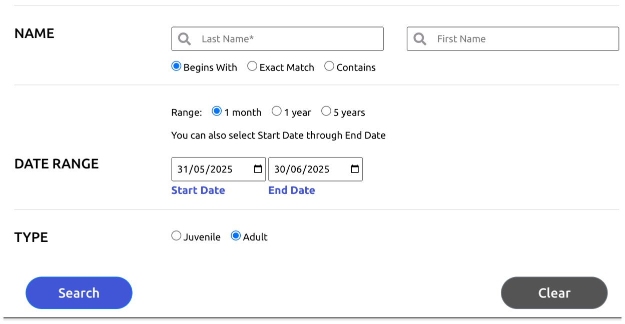
Disclaimers

Access to certain data on eClerksLA® is restricted to an individual authorized to use eClerksLA (User).

All data is provided "as is." LCRAA is not responsible for the contents of any Participants' records or websites. Data on eClerksLA® does not constitute an official record.

3. **New Tab Handling**
   - Detect when new tab opens
   - Switch Selenium focus to the new tab

4. **Date Range Configuration**
   - Locate the two date range input boxes
   - Set "From Date" to: 01/01/2020
   - Set "To Date" to: 01/07/2025



NAME
Last Name* | First Name
[✓ Begins With] Exact Match Contains

DATE RANGE
Range: [✓ 1 month] 1 year 5 years
You can also select Start Date through End Date
Start Date: 31/05/2025
End Date: 30/06/2025

TYPE
Juvenile [✓ Adult]

5. **Search Execution**
   - Click the search button
   - Wait for results table to load
6. **Data Extraction**
   - Scrape all columns from the results table:
     - Defendant
     - Birth Date
     - Sex
     - Race
     - Case
     - Date Filed
     - Charges
     - Arrest/Citation
     - Parish
     - Alert (icon indicator)
   - Extract all rows of data
   - Handle pagination if multiple pages exist

3.2 Data Processing

1. Parse and clean scraped data
2. Convert date strings to proper date format
3. Handle missing or malformed data
4. Save records to Django database
5. Implement duplicate prevention using case numbers

Phase 4: Django Backend Development

4.1 Views Implementation

1. **Records List View**
   - **Create a view with frontend template**
   - Display all criminal records in a table format
   - Implement search functionality (by name, case number, parish)
   - Add pagination for large datasets

    - Show total record count
2. **Record Detail View**
    - Display complete information for a single record
    - Format charges text for readability
    - Include navigation back to list

4.2 Django Admin Integration

1. Register criminal record model in admin
2. Configure admin interface with:
    - List display with key fields
    - Search functionality
    - Filtering options by parish, date, etc.
    - Read-only fields for scraped data

Phase 5: Frontend Development

5 Template Design

1. **Base Template**
    - Create responsive layout using Bootstrap
    - Include navigation header
    - Set up consistent styling
2. **Records List Template**
    - Responsive table design
    - Search form with clear/reset options
    - Pagination controls
3. **Record Detail Template**
    - Organized information display
    - Proper formatting for charges section
    - Back navigation

Phase 6: Management Commands

1. Create Django management command for running scraper
2. Add command-line options for date ranges
3. Include progress indicators and error reporting
4. Schedule capability for automated runs

Phase 7: Error Handling & Logging

1. Implement comprehensive error handling for:
    - Network connectivity issues
    - Website structure changes
    - Data parsing errors
    - Database connection problems
2. Add logging for scraping activities
3. Create error notification system

Deliverables

1. Codebase

- Complete Django project with scraper app
- Selenium scraping script
- Database models and migrations
- Frontend templates and views
- Admin interface configuration

2. Documentation

- Setup and installation instructions
- How to run the scraper
- Database schema documentation
- User guide for web interface

3. Testing

- Test scraper functionality end-to-end
- Verify data accuracy and completeness
- Test frontend functionality across devices
- Validate error handling scenarios

Success Criteria

1. Scraper successfully extracts first 20 row's data from criminal records table
2. Data is accurately stored in Django database
3. Web interface displays records with search and pagination
4. System handles errors gracefully
5. Code is well-documented and maintainable
6. Application is responsive and user-friendly