**🧩 Project: NJBuds – New Jersey Dispensary Data Pipeline**

**🎯 Project Overview**

**Goal:** Build an automated, end-to-end data collection pipeline for all licensed cannabis dispensaries in New Jersey.  
The pipeline extracts official business information from public sources, enriches it with websites and phone numbers, and prepares the data for integration into a database and eventual product/menu aggregation system.

This project demonstrates **real-world data engineering and automation**: web scraping, data enrichment, normalization, and readiness for downstream analytics and web application use.

**🏗️ Phase 1 — Data Acquisition (Official NJ CRC Site)**

**Objective:** Collect the most accurate list of active dispensaries directly from the New Jersey Cannabis Regulatory Commission (CRC) public map.

**Process**

1. **Analyzed source structure:** The official CRC map ([https://www.nj.gov/cannabis/dispensaries/find/](https://www.nj.gov/cannabis/dispensaries/find/?utm_source=chatgpt.com)) uses an embedded Atlist map.
2. **Built Selenium scraper (scrape\_crc\_iframe.py):**
   * Opened the embedded iframe automatically.
   * Extracted dispensary names, addresses, and coordinates from dynamic map cards.
   * Exported data to nj\_dispensaries.csv.
3. **Results:**  
   ✅ 269 active dispensaries captured  
   ✅ Data included name, street, city, state, zip, and source

**🌿 Phase 2 — Recreational vs. Medicinal Segmentation**

**Objective:** Differentiate between recreational and medicinal license types.

**Key Steps**

* Attempted a second scraper focusing on the **Medicinal** layer in the Atlist map.
* Discovered both license types load from a **single underlying data source**.
* Finalized a unified scraper (scrape\_crc\_all\_sites.py) that captures all dispensaries in one run.
* Later classification will be handled via a license\_type column.

**🔍 Phase 3 — Website Discovery**

**Objective:** Automatically find the **official website** for each dispensary.

**Challenges**

* No official API or export exists for website data.
* Simple HTML search attempts (using requests + DuckDuckGo) were blocked by rate limits.

**Solution**

**Selenium-based web search automation (find\_websites\_via\_search\_selenium.py):**

* Uses DuckDuckGo search for each dispensary ("<name> <city> NJ dispensary").
* Skips social media, review, and directory sites (Weedmaps, Leafly, Yelp, etc.).
* Selects the first real business domain.
* Writes incremental checkpoints every 20 rows.
* Produces nj\_dispensaries\_with\_websites.csv.

**Outcome:**  
✅ 273 dispensaries processed  
✅ 260+ official websites discovered automatically

**☎️ Phase 4 — Phone Number Enrichment**

**Objective:** Extract phone numbers for each dispensary.

**Implementation**

**Script:** enrich\_phones\_from\_sites.py

**Workflow:**

* Reads each website from nj\_dispensaries\_with\_websites.csv.
* Crawls homepage and common subpages (/contact, /locations, /about).
* Extracts phone numbers from:
  + tel: links
  + Plain text (regex pattern for (XXX) XXX-XXXX)
* Falls back to directory pages (Weedmaps, Leafly, Dutchie, iHeartJane) when necessary.

**Output:**  
✅ nj\_dispensaries\_with\_phones.csv  
✅ 80–85% coverage achieved (≈47 records missing phones)

**🧠 Phase 5 — Platform Detection (Pre-Product Integration)**

**Objective:** Identify which online menu platform each dispensary uses.

**Script:** detect\_menu\_platforms.py

**Logic:**

* Scans HTML of each dispensary website for platform signatures:
  + Dutchie
  + iHeartJane
  + Leafly
  + Weedmaps
  + Shopify / WooCommerce (for smaller sites)
* Detects embedded iframes, API URLs, or “Shop/Menu” links.
* Outputs menu\_platform, menu\_url, and confidence score.

**Output:**  
✅ nj\_dispensaries\_with\_platforms.csv  
✅ Ready for menu/product data scraping by platform type.