**Step-by-Step Process**

**1. Initial Research:**  
I began by searching for websites that display live Fortnite player count apart from [Fortnite.gg](https://fortnite.gg/).

**2. Alternative Sources:**  
The only other notable site was TRN (Tracker Network), but I knew their API was deprecated due to prior attempts to scrape Valorant data.

**3. FortniteAPI.io Exploration:**  
I discovered [FortniteAPI.io](https://fortniteapi.io/) and obtained access to their API key. However, it only offers map-related data, not player counts. The servers are a little wonky but they work.

**4. Network Traffic Analysis:**  
I inspected the network logs on Fortnite.gg to determine where the site was retrieving its data from.

**5. Server-Side Data Injection:**  
It became clear the data was being injected server-side, and no usable hidden API was found despite multiple attempts.

**6. Map Code Embedding:**  
I explored various map codes and observed that each was linked to a specific server-side ID, which was tied to the player count table.

**7. Real-Time Data Retrieval Strategy:**  
I shifted focus to find a method for real-time data access.

**8. Browser Automation Attempts:**  
Initial attempts at browser automation proved too slow. Moreover, Fortnite.gg has robust Cloudflare protections that block most proxy-based scraping.

**9. CSS Skeleton Strategy with crawl4ai:**  
I noticed the site maintains a consistent skeleton CSS structure throughout its flows. This allowed me to:

* Easily isolate the data
* Convert the content to Markdown
* Parse the Markdown to JSON

I used crawl4ai for this purpose.

**Advantages of This Approach**

* **Speed:** Efficient extraction with minimal latency
* **Real-Time:** Captures live data accurately
* **Customizable:** Flexible schema to extract additional fields
* **Rate-Limiting Friendly:** Requires fewer requests compared to conventional scraping

**Predictive Modelling with Prophet**

For forecasting player counts, I chose **Prophet** for the following reasons:

* Designed for time series data with strong seasonality and trend components
* Handles missing or irregular data gracefully
* Simple and intuitive configuration
* Produces interpretable forecasts with confidence intervals
* Ideal for demonstration and experimentation in real-world scenarios involving live scraped data