

Testing Report

Prepared by: *Karishma Mohammed*

Testing Overview

To make sure my program was stable and reliable, I tested it under different ways. The goal was to confirm that it consistently collected accurate match data while handling real-world challenges like missing data, slow networks, or API delays.

Testing Scenarios & Validation Steps

1. Multiple Runs at Different Times

I ran the program several times throughout to check if it gave consistent results. Each run successfully updated the JSON output without duplication, crashes, or data loss — showing the tool's stability.

2. Handling Empty or Missing Data

When the API didn't return any active matches, the Python handled it gracefully. Instead of stopping or showing errors, it displayed a clear message indicating no games were found. This ensures reliability even during off-peak hours.

3. Network and Timeout Testing

To simulate real-world issues, I tested the tool on slower internet connections. It retried failed requests automatically and closed the Playwright browser safely afterward. This confirmed that the program cleans up resources properly and remains stable under network problems.

4. Data Validation and Accuracy Checks

After each test run, I reviewed the JSON output to make sure it matched the expected structure. All important fields — such as id, league, start_time, and odds — were correctly filled in. This confirmed that the collected data was clean, complete, and accurate.

5. Cross-Sport Verification

The program correctly detected and recorded matches for basketball (NBA), baseball (MLB), and football (NFL). All leagues and teams were categorized properly, confirming that it works well across different sports.

Testing Results

- ✓ Consistent performance across multiple runs
- ✓ Clear handling of missing or empty data
- ✓ Stable output in valid JSON format
- ✓ No resource leaks or browser crashes
- ✓ Accurate classification across sports leagues

Summary

The program is reliable, stable, and easy to use. It handles real-time data changes, connection issues, and API updates without breaking. The final results are clean and consistent, and I'm confident it can run safely in different environments while keeping the data accurate.