Selenium Test Plan for Orbit - Schedule Missions Page (Static HTML)

Overview

This test plan covers automated UI verification of the provided static HTML page ("Orbit - Schedule Missions") using Selenium WebDriver with Python. The tests validate page structure, content rendering, and basic interactions on a local file. The page is purely static (no backend or dynamic JavaScript behavior beyond CSS styling), so the focus is on element presence, visible text (accounting for CSS transformations like text-transform: uppercase), icons, and stability during interactions.

Test Environment

- **OS**: macOS
- **Browser**: Chrome (latest version, managed via ChromeDriver)
- **WebDriver**: ChromeDriver (auto-managed by webdriver-manager)
- **Python**: 3.9+ (user's environment: Python 3.9)
- **Libraries**: selenium, webdriver-manager
- **Execution Mode**: Headed (browser visible for observation)
- **Test File**: test_orbit_schedule_selenium.py
- **HTML File**: orbit-schedule-missions.html (must be in the same directory)

Test Scope

- **In Scope**: Page title, header, sidebar (including uppercase rendering), main content, table (headers, 3 rows, status icons, action icons, badges), footer elements, basic interactions (clicking icons, typing in search).
- **Out of Scope**: Functional behavior (e.g., actual mission scheduling), performance, security, responsiveness across devices.

Test Scenarios (7 Total Tests)

1. test_page_title – Verify browser title is "Orbit - Schedule Missions".
2. test_header_elements – Validate logo ("Orbit™") and top-right icons (cog, user-circle).
3. test_sidebar_elements – Confirm sidebar section titles render as uppercase ("ACCOUNT", "ROBOTS", "ADMIN SETTINGS") and "Schedule Missions" is active.
4. test_main_content – Check page title, search box placeholder, and weather link.
5. test_table_content – Validate table headers, row count (3), first row data, status colors, lockout badge, and action icons.
6. test_footer_and_misc – Verify calendar download link, robot time text, and version "5.0.0".
7. test_basic_interaction – Click edit icon and type in search box; confirm no crashes and input is registered.

Execution Steps

1. Save both files in the same directory:
    - orbit-schedule-missions.html
    - test_orbit_schedule_selenium.py
2. Open Terminal and navigate to the directory:
   Bash
   ```
   cd /path/to/your/tests/folder
   ```
3. Run the test suite:
   Bash
   ```
   python3 test_orbit_schedule_selenium.py
   ```

Actual Test Run Results (December 30, 2025)

The test suite was executed successfully with the following output:

Bash
```
user@users-iMac tests % python3 test_orbit_schedule_selenium.py
----------------------------------------------------------------------
Ran 7 tests in 4.080s

OK
user@users-iMac tests %
```

Result Summary

- **Total Tests**: 7
- **Passed**: 7
- **Failed**: 0
- **Duration**: ~4.08 seconds
- **Status**: **PASS** (All tests successful – OK)

Risks and Notes

- Tests rely on visible rendered text (e.g., uppercase sidebar titles due to CSS).
- If the HTML file is modified, tests may need updates.

This test plan and execution confirm that the Orbit - Schedule Missions page renders correctly and remains stable under basic interaction in the target environment.