# Mobile Automation Script – Documentation

## Approach

The automation script was developed using Python with Appium and Selenium WebDriver, targeting a mobile browser environment on an Android emulator. The goal was to automate user interaction with the Motorola Solutions PSCore Field Response & Reporting webpage.  
  
Key Steps:  
1. Browser Launch: Chrome is opened using Appium and matched with the appropriate chromedriver version.  
2. Website Navigation: The script navigates to the PSCore product page.  
3. Scroll and Interaction: `scrollIntoView()` is used to bring the “Informed Tasking” tab into view, and JavaScript-based `click()` is used to interact with UI elements.  
4. Wait Strategy: `WebDriverWait` and expected conditions are used to ensure elements are fully loaded before actions are taken.  
5. Cleanup: Chrome browser is closed at the end of the test.

## Challenges Faced & Solutions

|  |  |  |
| --- | --- | --- |
| Challenge | Cause | Solution |
| ChromeDriver mismatch | Emulator had Chrome v133, Appium expected v136 | Downloaded correct chromedriver and set path via `chromedriverExecutable`. |
| ElementClickInterceptedException | Element hidden or overlapped by headers | Used `scrollIntoView()` and JavaScript-based click. |
| Appium timeout | Emulator took longer to launch instrumentation | Increased `uiautomator2ServerLaunchTimeout` to 60000ms. |
| Emulator screen locked | Appium couldn’t interact with locked screen | Unlocked emulator using `adb shell input keyevent 82`. |
| Pixel-based scroll failed | Element location varied by screen size | Replaced hardcoded scroll with `element.scrollIntoView()`. |