# Task 1: UI Component Test with MetaMask

## Approach

### Framework and Tools

- Automation Framework: Selenium with C# and NUnit  
- Browser: Google Chrome with premade profiles for VPN and non-VPN scenarios  
- Extensions: MetaMask extension for managing blockchain interactions

### Flow Adjustments for Stability

- Bug: Sometimes MetaMask doesn't open after clicking the 'Connect' button in a new browser session where the wallet is locked (info below; video attached)  
- Resolution: Pre-login into MetaMask before starting the test scenario.

## Test Scenarios

### Scenario 1: With VPN Access

- Used a Chrome profile configured for VPN.  
- Checked wallet connection, interaction, and signing.  
- Verified that the wallet address was displayed.

### Scenario 2: Without VPN Access

- Used a Chrome profile without VPN.  
- Verified that the dApp showed an error message indicating restricted access.

## Bug

### Environment on which the bug was found

- Chrome version: 131.0.6778.140 (Official Build) (64-bit)  
- MetaMask version: 12.9.1

### Steps to Reproduce

1. Open a new Chrome session with MetaMask connected.  
2. Click Connect → Agree to all → Select.  
3. A loader blinks but MetaMask doesn’t open.

### Alternative Way to Reproduce

- Same behavior occurs if MetaMask is closed without entering the password.

## Resolution

Logging into MetaMask before running tests stabilizes the flow and ensures it opens during the test.

# Task 2: End-to-End Flow Test with REST Endpoints

## Approach

- Library: RestSharp for API requests  
- API: SpaceX API (https://github.com/r-spacex/SpaceX-API)

## Scenario

The goal is to get the payload names for the second launch of Falcon 9.

## Steps

1. Get Rocket ID by Name:  
 - Retrieved the list of rockets.  
 - Filtered for the one named Falcon 9.  
 - Extracted its unique ID.  
  
2. Get Launches for the Rocket:  
 - Used the rocket ID to fetch launches.  
 - Sorted launches by date.  
 - Picked the second launch's ID.  
  
3. Get Payload IDs for the second launch:  
 - Fetched launch details by ID.  
 - Extracted payload IDs.  
  
4. Get Payload Names:  
 - Retrieved details for each payload ID.  
 - Extracted and listed payload names.

## Validation

- Ensured the names matched expectations for the second Falcon 9 launch.  
- Checked intermediate results and logged responses.  
- Example names retrieved: [ 'Cubesats', 'COTS Demo Flight 1' ]

# Task 3: Self-Learning & Technology Understanding

## Possible Test Case and Bug

### Test Case: Consensus Mechanism Validation

#### Steps

1. Prepare a Wallet with 10 COTI.  
  
2. Initiate Two Conflicting Transactions:  
 - Transaction A: Send 10 COTI to Recipient 1.  
 - Transaction B: Immediately try sending 10 COTI to Recipient 2.  
  
3. Submit Both Transactions:  
 - Triggered both transactions back-to-back.  
  
4. Validate Results:  
 - Confirmed only one transaction succeeded.  
 - Ensured the second transaction failed with 'insufficient funds'.

#### Expected Behavior

- Only the first transaction is validated.  
- Wallet balance updates after the first transaction, invalidating the second.  
- Second transaction fails with an error.

#### Potential Bug

- Issue: Both transactions get accepted, causing a double spend.  
- Impact:  
 - Users could exploit this flaw to spend the same funds multiple times.