Login Test Script

Overview

This repository contains a Python script that uses Selenium WebDriver to test the login functionality of a web application. The script navigates to a login page, performs login actions, and verifies if the login was successful.

Prerequisites

- Python 3.x installed on your machine.
- `pip` (Python package installer) for installing required packages.

Setup

1. Install Python

Ensure that Python 3.x is installed on your machine. You can download Python from [the official Python website].

2. Install Required Packages

You need to install the following Python packages:

- `selenium`: For interacting with the web browser.
- `webdriver_manager`: For managing WebDriver binaries automatically.

You can install these packages using pip. Run the following command in your terminal or command prompt:

```
bash
Copy code
pip install selenium webdriver_manager
```

3. WebDriver

The script uses `webdriver_manager` to handle WebDriver binaries. It will automatically download the correct WebDriver for your browser. Currently, the script is set up for Google Chrome. If you want to use a different browser, you will need to modify the script accordingly and ensure the corresponding WebDriver is installed.

Configuration

1. **Edit the Script**: Open the `login_test.py` file and adjust the following placeholders:
    - `https://your-web-application-url.com/login`: Replace with the URL of your web application's login page.
    - `username_field`, `password_field`, `login_button`: Update the `By.ID` selectors to match the IDs or other attributes of your login form elements.
    - `your_username` and `your_password`: Replace with the credentials you want to test.
2. **Save the File**: Ensure you save any changes made to the script.

Running the Script

1. **Navigate to the Script Directory**: Open your terminal or command prompt and navigate to the directory containing the `login_test.py` file.
2. **Run the Script**: Execute the script by running the following command:

    ```
    bash
    Copy code
    python login_test.py
    ```

3. **View Results**: The script will print `Login successful!` or `Login failed!` based on the result of the login attempt. Check the browser window that opens to see the automated login process.

Troubleshooting

- **Browser Compatibility**: Ensure you have the correct WebDriver version for your browser. The script uses Chrome by default; if you need Firefox or another browser, update the script accordingly.
- **Element Selectors**: Make sure that the element selectors (IDs) used in the script match those in your actual web application. You might need to adjust the selectors if the IDs or attributes are different.

Key Points:

- **WebDriver**: This script uses `webdriver_manager` to handle the WebDriver installation and setup automatically. You can also manually download the WebDriver and set its path if you prefer.
- **Selectors**: Adjust the `By.ID` selectors to match the actual IDs or other attributes of the input fields and buttons in your login form.
- **URL**: Replace `"https://your-web-application-url.com/login"` with the URL of your web application’s login page.
- **Waits**: Depending on your application’s performance, you may need to add explicit waits (using `WebDriverWait`) to handle asynchronous content loading.