🚀 Cflow Playwright Automation Report

Report generated on 14-Nov-2025 at 18:21:05 by pytest-html v4.1.1

Environment

Base URL

Summary

📄 **Project:** Cflow Playwright Automation | 🏆 📁 **Module:** Admin Module - Add User | 👤 **Tester:** Dinesh Aravinth ⚡ | 🌐 **Browser:** Chromium | 🌍 **Region:** TEST | ⏰ **Execution:** 14-Nov-2025 18:21:05

10 tests took 00:04:09.

(Un)check the boxes to filter the results.

☑ 0 Failed, ☑ **10 Passed,** ☑ 0 Skipped, ☑ 0 Expected failures, ☑ 0 Unexpected passes, ☑ 0 Errors, ☑ 0 Reruns   Show all details / Hide all details

Result	Test	Duration	Links
Passed	TestCases/Admin_add_user_testcases/test_A_Admin_add_user_positive.py::Test_01AdminAddUserPositiveCases::test_TC01_add_user_with_active_status	00:00:22	

```
-------------------------- Captured stdout setup --------------------------
```
collapse [-]
```
🌐  Launching chromium for region Test
➡️   Navigating to URL: https://testapp.cflowapps.com/cflow/login for region: Test
✅ Entered 'cflowautomation.com' into Client ID
✅ Entered 'testautomation@cflowautomation.com' into Username
✅ Entered '1234567890' into Password
✅ Clicked: Login button
🌐 Verifying Dashboard URL ...
🏆 Dashboard URL verification passed! ✅
Actual URL: 'https://testapp.cflowapps.com/cflow/dashboard'

-------------------------- Captured stdout call --------------------------

🚀 Starting Test: Add User with Active Status
✅ Clicked: Admin-side navigation link
🌐 Verifying Admin URL ...
🏆 Admin URL verification passed! ✅
Actual URL: 'https://testapp.cflowapps.com/cflow/admin/user-role-permission'
✅ Clicked: Add User button in the Admin page
✅ Entered 'User_zqtfx' into Name textbox
✅ Entered 'QA' into Department textbox
✅ Entered 'dinesh123@yopmail.com' into Email textbox
🔑 Entering Login ID: user_qehhz
✅ Entered 'user_qehhz' into Login ID textbox
✅ Login ID entered successfully: user_qehhz
🔍 Visible password validations:
1. Must be at least 12 characters long
2. Must be less than 18 characters long
3. Must have a number.
4. Must have an alphabet.
5. Must have a special character.
6. Must have an uppercase letter.
🔑 Generated valid password: 4TbS*leDRLgc
✅ Entered '4TbS*leDRLgc' into Password textbox
✅ Entered '1064' into Employee Number textbox
🎯 Selecting roles: ['User', 'Admin']
➡ Selecting role: 'User'
✅ Clicked: Roles dropdown
✅ Selected role: 'User'
➡ Selecting role: 'Admin'
✅ Clicked: Roles dropdown
✅ Selected role: 'Admin'
✅ Entered '91' into Country code input
✅ Entered '9988776655' into WhatsApp number input
✅ Element visible — 'Send welcome mail to the user?' toggle
✓ 'Send welcome mail to the user?' toggle enabled
✅ Clicked: Save button
💬 Toast detected: 'User Details Saved Successfully ×'
✅ Success message: User Details Saved Successfully ×
✅ User 'User_zqtfx' added successfully with Active status

🔍  Searching for user 'User_zqtfx' in the Active Users page...
✅ User 'User_zqtfx' search completed.
✅ User 'User_zqtfx' found in All Users Page.

🔍  Verifying user status toggle for 'User_zqtfx' under 'User Status Toggle'...
🔴 🏆  User 'User_zqtfx' status is Active (toggle ON).
```

| Passed | TestCases/Admin_add_user_testcases/test_A_Admin_add_user_positive.py::Test_01AdminAddUserPositiveCases::test_TC02_verify_user_in_employee_lookup | 00:00:04 | |

collapse [-]
```
-------------------------- Captured stdout call --------------------------
```

Result ▼ Test	Duration	Links

🔍 Verifying Employee Lookup entry for user: User_zqtfx
✅ Clicked: Lookup-side navigation link
✅ Clicked: Employees Lookup
🎯 Fetching latest employee record from lookup table...

📋 Employee Record (Full Details):
   ID: 23
   Employee No: 1064
   Employee Name: User_zqtfx
   Login ID: user_qehhz
   Email ID: dinesh123@yopmail.com
   Department: QA
   Created By: testautomation@cflowautomation.com
   Created Date: 14 Nov 2025
   Updated By: testautomation@cflowautomation.com
   Updated Date: 14 Nov 2025

📊 Comparing Employee Record Details:
✅ Employee No matches: '1064'
✅ Employee Name matches: 'User_zqtfx'
✅ Login ID matches: 'user_qehhz'
✅ Email ID matches: 'dinesh123@yopmail.com'
✅ Department matches: 'QA'

🎯 All employee details correctly reflected in Employee Lookup table.

Passed	TestCases/Admin_add_user_testcases/test_A_Admin_add_user_positive.py::Test_01AdminAddUserPositiveCases::test_TC03_reset_password_of_created_user	00:00:11	

---------------------------- Captured stdout call ----------------------------

collapse [-]

🚀 Resetting password for created user
✅ Clicked: Admin-side navigation link

🔍 Searching for user 'User_zqtfx' in the Active Users page...
✅ User 'User_zqtfx' search completed.
✅ Clicked on user 'User_zqtfx' in 'All Users list'.
✅ Clicked: Reset Password link
🖊 Trying old password: 4TbS*leDRLgc
✅ Entered '4TbS*leDRLgc' into Reset Password field
✅ Clicked: Update button (old password)
✅ Policy validation triggered correctly: 'Your password not met the password policy

×'
🔍 Visible password validations:
1. Must be at least 12 characters long
2. Must be less than 18 characters long
3. Must have a number.
4. Must have an alphabet.
5. Must have a special character.
6. Must have an uppercase letter.
🔑 Generated valid password: 8fT5G4jywT#z
✅ Entered '8fT5G4jywT#z' into Reset Password field
✅ Clicked: Update button (valid password)
✅ Password reset successful: 'Password Updated Successfully

×'

Passed	TestCases/Admin_add_user_testcases/test_A_Admin_add_user_positive.py::Test_01AdminAddUserPositiveCases::test_TC04_add_user_with_disabled_status	00:00:15	

---------------------------- Captured stdout call ----------------------------

collapse [-]

🚀 Starting Test: Add User with Disabled Status
✅ Clicked: Admin-side navigation link
🌐 Verifying Admin URL ...
🧑 Admin URL verification passed! ✅
Actual URL: 'https://testapp.cflowapps.com/cflow/admin/user-role-permission'
✅ Clicked: Add User button in the Admin page
✅ Entered 'User_egdci' into Name textbox
✅ Entered 'QA' into Department textbox
✅ Entered 'dinesh123@yopmail.com' into Email textbox
🔑 Entering Login ID: user_lwqfv
✅ Entered 'user_lwqfv' into Login ID textbox
✅ Login ID entered successfully: user_lwqfv
🔍 Visible password validations:
1. Must be at least 12 characters long
2. Must be less than 18 characters long
3. Must have a number.
4. Must have an alphabet.
5. Must have a special character.
6. Must have an uppercase letter.
🔑 Generated valid password: D#e!nHBYZ2@j
✅ Entered 'D#e!nHBYZ2@j' into Password textbox
✅ Entered '4790' into Employee Number textbox

🎯 Selecting roles: ['User']

Result	Test	Duration	Links

→ Selecting role: 'User'
✅ Clicked: Roles dropdown
✅ Selected role: 'User'
✅ Entered '91' into Country code input
✅ Entered '9988776655' into WhatsApp number input
✅ Element visible — 'Status' toggle
✓ 'Status' toggle disabled
✅ Element visible — 'Send welcome mail to the user?' toggle
✓ 'Send welcome mail to the user?' toggle enabled
✅ Clicked: Save button
💬 Toast detected: 'User Details Saved Successfully ×'
✅ Success message: User Details Saved Successfully ×
✅ User 'User_egdci' added successfully with Disabled status

🔍 Searching for user 'User_egdci' in the Active Users page...
✅ User 'User_egdci' search completed.
✅ User 'User_egdci' found in All Users Page.
🔍 Verifying that user status toggle for 'User_egdci' under 'User Status Toggle' is DISABLED...
✓ User 'User_egdci' status is Disabled (toggle OFF).

| Passed | TestCases/Admin_add_user_testcases/test_A_Admin_add_user_positive.py::Test_01AdminAddUserPositiveCases::test_TC05_enable_and_verify_in_active_users_page | 00:00:07 | |

---------------------------- Captured stdout call ----------------------------

collapse [-]

🚀 Enabling user 'User_egdci' and verifying in Active Users
🔍 Verifying that user status toggle for 'User_egdci' under 'User Status Toggle' is DISABLED...
✓ User 'User_egdci' status is Disabled (toggle OFF).
🟢 Enabling user status toggle for 'User_egdci' under 'User Status Toggle'...
🔍 Searching for user 'User_egdci' before enabling toggle...
✅ User 'User_egdci' search completed.
✅ Element visible — User_egdci Toggle
🖱 Clicked visible toggle (slider) for 'User_egdci'...
🆗 Clicked 'Yes' button on confirmation popup.
✅ User 'User_egdci' successfully enabled (toggle ON).
✅ Enabled user 'User_egdci' successfully

🔍 Searching for user 'User_egdci' in the Active Users page...
✅ User 'User_egdci' search completed.
👁 Checking if 'User_egdci' appears in Active Users list...
✅ User 'User_egdci' is displayed in Active Users list.
🎯 Verified 'User_egdci' now appears in Active Users page

| Passed | TestCases/Admin_add_user_testcases/test_A_Admin_add_user_positive.py::Test_01AdminAddUserPositiveCases::test_TC06_verify_imported_users | 00:00:38 | |

---------------------------- Captured stdout call ----------------------------

collapse [-]

✅ Clicked: Admin-side navigation link
✅ Clicked: Import button on the page
✅ File 'TestData/User_Import.xlsx' successfully uploaded using Excel file upload field
✅ Clicked: Upload button
💬 Toast detected after upload: 'Successfully imported 3 Users ×'
✅ Upload successful based on toast: Successfully imported 3 Users ×
ℹ No 'Import Summary' popup detected — continuing...
📄 Loaded 3 users from Excel for verification.

🔍 Verifying imported user: import user 19
✅ Name matches: import user 19
✅ Login ID matches: importuser29
✅ Email matches: importuser1@yopmail.com
✅ Role matches: User
✅ WhatsApp Number matches: 9994397434
✅ Employee Number matches: IM19
✅ Department matches: QA
🎯 Verification completed for 'import user 19'

🔍 Verifying imported user: import user 20
✅ Name matches: import user 20
✅ Login ID matches: importuser30
✅ Email matches: importuser1@yopmail.com
✅ Role matches: User
✅ WhatsApp Number matches: 9994397435
✅ Employee Number matches: IM20
✅ Department matches: QA
🎯 Verification completed for 'import user 20'

🔍 Verifying imported user: import user 21
✅ Name matches: import user 21
✅ Login ID matches: importuser31
✅ Email matches: importuser1@yopmail.com

✅ Role matches: User

Result ▼ Test	Duration	Links

✅ WhatsApp Number matches: 9994397436
✅ Employee Number matches: IM21
✅ Department matches: QA
🎯 Verification completed for 'import user 21'

✅ All imported users verified successfully!

Passed	TestCases/Admin_add_user_testcases/test_A_Admin_add_user_verify_login_positive.py::test_verify_new_user_login_standalone	00:00:24	

```
---------------------------- Captured stdout call ----------------------------
```
collapse [-]

✅ Entered 'cflowautomation.com' into Client ID
✅ Entered 'testautomation@cflowautomation.com' into Username
✅ Entered '1234567890' into Password
✅ Clicked: Login button
🌐 Verifying Dashboard URL ...
🏆 Dashboard URL verification passed! ✅
Actual URL: 'https://testapp.cflowapps.com/cflow/dashboard'
✅ Clicked: Admin-side navigation link
✅ Clicked: Add User button in the Admin page
✅ Entered 'User_nucow' into Name textbox
🔑 Entering Login ID: user_fbdpq
✅ Entered 'user_fbdpq' into Login ID textbox
✅ Login ID entered successfully: user_fbdpq
🔍 Visible password validations:
1. Must be at least 12 characters long
2. Must be less than 18 characters long
3. Must have a number.
4. Must have an alphabet.
5. Must have a special character.
6. Must have an uppercase letter.
🔑 Generated valid password: *SnzZRymGGEY5
✅ Entered '*SnzZRymGGEY5' into Password textbox
✅ Entered 'QA' into Department textbox
✅ Entered 'autotest@yopmail.com' into Email textbox
✅ Entered '4478' into Employee Number textbox
🎯 Selecting roles: ['User']
→ Selecting role: 'User'
✅ Clicked: Roles dropdown
✅ Selected role: 'User'
✅ Entered '91' into Country code input
✅ Entered '9988776655' into WhatsApp number input
✅ Element visible — 'Send welcome mail to the user?' toggle
✓ 'Send welcome mail to the user?' toggle enabled
✅ Clicked: Save button
💬 Toast detected: 'User Details Saved Successfully ×'
✅ Success message: User Details Saved Successfully ×

🔍 Searching for user 'User_nucow' in the Active Users page...
✅ User 'User_nucow' search completed.
✅ User 'User_nucow' found in All Users Page.

🔍 Verifying user status toggle for 'User_nucow' under 'User Status Toggle'...
🔴 🧑 User 'User_nucow' status is Active (toggle ON).
🖊 Main browser closed after user creation.

🔒 Verifying login for new user: User_nucow
✅ Entered 'cflowautomation.com' into Client ID
✅ Entered 'user_fbdpq' into Login ID
✅ Entered '*SnzZRymGGEY5' into Password
✅ Clicked: Login button
✅ Login successful for user: User_nucow
🖊 Second browser closed after login verification.
🔴 Playwright test completed without triggering asyncio loop errors.

Passed	TestCases/Admin_add_user_testcases/test_B_Admin_add_user_negative.py::Test_002_Admin_Add_User_Negative_cases::test_TC01_add_user_with_duplicate_login_id	00:00:08	

```
---------------------------- Captured stdout call ----------------------------
```
collapse [-]

🚨 Starting Negative Test: Duplicate Login ID
✅ Clicked: Admin-side navigation link
✅ Clicked: Add User button in the Admin page
✅ Entered 'Existing_User' into Name textbox
✅ Entered 'QA' into Department textbox
✅ Entered 'existinguser@yopmail.com' into Email textbox
🔑 Entering Login ID: dinesh01
✅ Entered 'dinesh01' into Login ID textbox
✅ Login ID entered successfully: dinesh01
🔍 Visible password validations:
1. Must be at least 12 characters long
2. Must be less than 18 characters long
3. Must have a number.
4. Must have an alphabet.

5. Must have a special character.
6. Must have an uppercase letter.

🚀 Cflow Playwright Automation Report

✅ Entered 'HiE@ygJWpcIA6' into Password textbox
✅ Entered '123466' into Employee Number textbox
🎯 Selecting roles: ['User']
➡️ Selecting role: 'User'
✅ Clicked: Roles dropdown
✅ Selected role: 'User'
✅ Entered '91' into Country code input
✅ Entered '9988776655' into WhatsApp number input
✅ Clicked: Save button
⚠️ Toast message detected: Username Already Exist ×
✅ Negative test passed: Toast message displayed correctly

| Passed | TestCases/Admin_add_user_testcases/test_B_Admin_add_user_negative.py::Test_002_Admin_Add_User_Negative_cases::test_TC02_add_user_with_duplicate_emp_no | 00:00:08 |

```
--------------------------- Captured stdout call ---------------------------
```
🔔 Starting Negative Test: Duplicate Employee Number
✅ Clicked: Admin-side navigation link
✅ Clicked: Add User button in the Admin page
✅ Entered 'Existing_Employee' into Name textbox
✅ Entered 'QA' into Department textbox
✅ Entered 'existingemp@yopmail.com' into Email textbox
🔑 Entering Login ID: unique_login_id_00552
✅ Entered 'unique_login_id_00552' into Login ID textbox
✅ Login ID entered successfully: unique_login_id_00552
🔍 Visible password validations:
1. Must be at least 12 characters long
2. Must be less than 18 characters long
3. Must have a number.
4. Must have an alphabet.
5. Must have a special character.
6. Must have an uppercase letter.
🔑 Generated valid password: $&!F#F0NLQ9c
✅ Entered '$&!F#F0NLQ9c' into Password textbox
✅ Entered 'E02' into Employee Number textbox
🎯 Selecting roles: ['User']
➡️ Selecting role: 'User'
✅ Clicked: Roles dropdown
✅ Selected role: 'User'
✅ Entered '91' into Country code input
✅ Entered '9988776655' into WhatsApp number input
✅ Clicked: Save button
⚠️ Toast message detected: Employee No Already Exist ×
✅ Negative test passed: Duplicate Employee Number message displayed correctly

collapse [-]

| Passed | TestCases/Admin_add_user_testcases/test_B_Admin_add_user_negative.py::Test_002_Admin_Add_User_Negative_cases::test_TC03_add_user_with_invalid_passwords | 00:01:53 |

```
--------------------------- Captured stdout call ---------------------------
```
🔔 Starting Negative Test: Invalid Password Validations
✅ Clicked: Admin-side navigation link
✅ Clicked: Add User button in the Admin page
✅ Entered 'Invalid_Password_User' into Name textbox
✅ Entered 'QA' into Department textbox
✅ Entered 'tmsjpu@yopmail.com' into Email textbox
🔑 Entering Login ID: user_hcpyb
✅ Entered 'user_hcpyb' into Login ID textbox
✅ Login ID entered successfully: user_hcpyb
✅ Entered '2434' into Employee Number textbox
🎯 Selecting roles: ['User']
➡️ Selecting role: 'User'
✅ Clicked: Roles dropdown
✅ Selected role: 'User'
✅ Entered '91' into Country code input
✅ Entered '9876543210' into WhatsApp number input

🔍 Found 6 visible password rules:
1. Must be at least 12 characters long
2. Must be less than 18 characters long
3. Must have a number.
4. Must have an alphabet.
5. Must have a special character.
6. Must have an uppercase letter.
-> Rule: 'Must be at least 12 characters long' -> invalid password length: 11 -> pwd: Xk5aDNege&M
-> Rule: 'Must be less than 18 characters long' -> invalid password length: 20 -> pwd: qlMPZw2h5YFOlRuh*WGx
-> Rule: 'Must have a number.' -> invalid password length: 12 -> pwd: Cdn%yogJwbhw
-> Rule: 'Must have an alphabet.' -> invalid password length: 12 -> pwd: 7&1^%60#3@&7
-> Rule: 'Must have a special character.' -> invalid password length: 12 -> pwd: 3GOSvWKf7J2y
-> Rule: 'Must have an uppercase letter.' -> invalid password length: 12 -> pwd: @caj8brzrpcf

🧪 Testing invalid password for rule: Must be at least 12 characters long
    ➡️ Invalid password used: Xk5aDNege&M (len=11)

✅ Clicked: Save button

collapse [-]

Result ▼	Test	Duration	Links

✅ Negative case passed — Save button still visible for rule: 'Must be at least 12 characters long'

✳️ Testing invalid password for rule: Must be less than 18 characters long
    ➡ Invalid password used: qlMPZw2h5YFOlRuh*WGx (len=20)
✅ Clicked: Save button
✅ Negative case passed — Save button still visible for rule: 'Must be less than 18 characters long'

✳️ Testing invalid password for rule: Must have a number.
    ➡ Invalid password used: Cdn%yogJwbhw (len=12)
✅ Clicked: Save button
✅ Negative case passed — Save button still visible for rule: 'Must have a number.'

✳️ Testing invalid password for rule: Must have an alphabet.
    ➡ Invalid password used: 7&1^%60#3@&7 (len=12)
✅ Clicked: Save button
✅ Negative case passed — Save button still visible for rule: 'Must have an alphabet.'

✳️ Testing invalid password for rule: Must have a special character.
    ➡ Invalid password used: 3GOSvWKf7J2y (len=12)
✅ Clicked: Save button
✅ Negative case passed — Save button still visible for rule: 'Must have a special character.'

✳️ Testing invalid password for rule: Must have an uppercase letter.
    ➡ Invalid password used: @caj8brzrpcf (len=12)
✅ Clicked: Save button
✅ Negative case passed — Save button still visible for rule: 'Must have an uppercase letter.'
✅ Completed all invalid password validation tests successfully.
-------------------------- Captured stdout teardown --------------------------

🥳 Browser will remain open for manual inspection...
❗ Close it manually when done.
Press ENTER here in terminal to close browser...