

Project: Office ERP System

Page Tested: Admin Login and Home Page\ **Tool Used:** Selenium with Python\ **Test Type:** Functional Automation Test

1. Objective

To verify that an Admin user can log in successfully to the ERP system and is redirected to the home page displaying the welcome message. This ensures secure access and verifies the functional behavior of the login system.

2. Test Environment

- **URL:** <http://localhost:5173/>
 - **Browser:** Chrome
 - **Driver:** Chrome WebDriver
 - **Python Version:** 3.11+
 - **Libraries Used:**
 - selenium
 - time
-

3. Prerequisites

- Ensure the ERP system is running locally on port 5173.
- Chrome browser and matching ChromeDriver installed.
- Install Selenium:

```
pip install selenium
```

- Valid Admin Credentials:
 - **Email:** admin@gmail.com
 - **Password:** admin
-

4. Test Case: Admin Login and Home Page Verification

Test Steps:

1. Open the ERP login page.
2. Enter the admin email and password.
3. Click the login button.
4. Wait for the redirection to the home/dashboard page.
5. Verify the presence of the welcome message.
6. Take a screenshot of the home page after login.

Expected Result:

- Admin is successfully logged in.
 - Dashboard displays a welcome message (e.g., "Welcome, Admin").
-

5. Selenium Test Script

```
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
import time

# Launch Chrome browser
driver = webdriver.Chrome()
driver.maximize_window()
driver.get("http://localhost:5173/")

try:
    wait = WebDriverWait(driver, 10)

    # Login Fields
    email_input = wait.until(EC.presence_of_element_located((By.NAME, "email")))
    password_input = wait.until(EC.presence_of_element_located((By.NAME,
"password"))))

    email_input.send_keys("admin@gmail.com")
    password_input.send_keys("admin")

    # Click Login Button
    login_button = wait.until(EC.element_to_be_clickable((By.XPATH, "//
button[contains(text(),'Login')]")))
    login_button.click()
```

```

# Screenshot After Login
time.sleep(2)
driver.save_screenshot("after_login.png")

# Validate Welcome Message
welcome_text = wait.until(EC.presence_of_element_located((By.XPATH, "//
*[contains(text(),'Welcome')]")))
print("[PASS] Admin login test passed!")

except Exception as e:
    print("[FAIL] Test failed:", str(e))

finally:
    time.sleep(5)
    driver.quit()

```

6. Notes & Troubleshooting

- If the test fails, check the screenshot `after_login.png` to verify what page was loaded.
- Update XPath selectors if text or structure changes.
- Use browser Developer Tools to inspect elements.

7. Test Execution Report

Test Step	Expected Result	Actual Result	Status
Open login page	Login form loads	Success	Pass
Enter email and password	Fields accept input	Input entered	Pass
Click login	Navigates to home/dashboard	Navigation occurred	Pass
Wait for welcome text	"Welcome" is visible	Text found	Pass
Capture screenshot	Screenshot is saved	after_login.png saved	Pass

Final Result:  All steps passed. Admin login functionality is working correctly.

8. Next Steps

- Extend automation to cover:
 - My Profile
 - Salary Slips
 - Attendance

- Leave Requests
- Add assertions for sidebar navigation.
- Integrate with `pytest` for structured test reporting.
- Export results to HTML/CSV for CI pipelines.