Assignment for the Software Development Engineer in Test

Our objective is to create test cases and build a test automation suite that will validate functional requirements of the web app named PMTool. You can access this web app through https://pm-tool-e63fa77e3353.herokuapp.com.

Acceptance Criteria

● New users should be able to create an account giving name, email and password and optionally company and address.

● Users should be able to Add / Edit / View / Delete projects.

● Users should be able to Add / Edit / View / Delete tasks for a specific project.

● There are proper validation messages in case of missing required fields during creation/ update of user / project / task.

Deliverables

The final deliverable should contain:

● Test Scenarios that will test e2e the PMTool application.

● Test Cases that will validate each requirement and determine whether the application is working as it was specified. We expect you to include positive as well as negative test cases.

● Automated Test scripts for the test cases you decide that need to be automated. You are allowed to use a programming language of your choice.

● Instructions on how to execute your automated tests.

Notes: You are only required to test UI elements and interactions.

EXECUTION

Contents

1. Understanding of the application under test

As primary step need to understand and (if possible) categorize the functional requirements. Acceptance criteria need to be analyzed and categorize them by feature modules.

The original acceptance criteria are posted below:

1. New users should be able to create an account giving name, email and password and optionally company and address.
2. Users should be able to Add / Edit / View / Delete projects.
3. Users should be able to Add / Edit / View / Delete tasks for a specific project.
4. There are proper validation messages in case of missing required fields during creation/ update of user / project / task.

As next step acceptance criteria will be divided into modules as shown in table below:

Module	**Functionality**
User Management	User registration with required and optional fields, login, logout
Project CRUD	Create, Read, Update, Delete a project
Task CRUD	Create, Read, Update, Delete tasks within a project

The reason for dividing the criteria into modules is to organize into manageable parts. When testing a complex system-app, test cannot be performed at once and need to break to modules. Each module is a logical group of related actions, and this structure helps writing focused tests.

CRUD stands for Create, Read, Update, Delete and covers the complete lifecycle of data, this is pattern that provides a test coverage checklist and ensures that everything a user might do is been tested and nothing missed.

Module 1: User management

New users should be able to create an account giving name, email and password (required) and optionally company and address.

UI elements that will be part of testing:

1. Input fields: Name, Email, Password, company (optional), address (optional)
2. Submit button
3. Success message/ redirection to verify account
4. Error messages (missing fields, invalid email format etc)

CRUD Breakdown:

Operation	**Description**	**UI Behavior**	**Expected Result**
Create	Fill out form and submit	New user is created, redirected	Success message /User lands on verification page
Read	Check in settings tab information entered	Information to be visible in settings tab	Information must match information entered at login
Update	User editing profile optional	Edit name/email/pass etc	Changes saved and reflected
Delete improvement	User Deletes account optional	Confirmation popup	Account is removed/logged out

Module 2: Project Management

Users should be able to Add / Edit / View / Delete projects

UI elements that will be part of testing:

1. Redirect to Dashboard tab
2. Create button (Project) button inside Dashboard tab
3. Project form name and description
4. 2nd Create button after filling form/ for changes to be saved/reflected
5. Edit button for each project/ then apply button for changes to be saved/reflected
6. Delete button, for deleting the project/ Confirmation pop-up/apply for changes to be saved-reflected

CRUD Breakdown:

Operation	**UI Action**	**Expected Result**
Create	Click “Create”, fill form, Click Create again for changes to be saved-reflected	New project appears in list
Read	After project creation, a project card must appear	One card with per project with name and description
Update	Click “Edit”, change fields, click “Update” for changes to be saved-reflected	Project info updated, reflected in list
Delete	Click "Delete", confirm action	Project is removed from list

Some validations that must be performed

1. Required fields: project name cannot be blank
2. Max field length
3. Invalid inputs: only spaces

Module 3: Task Management

Users should be able to Add / Edit / View / Delete tasks for a specific project.

UI elements that will be part of testing:

1. Add Task button inside a project
2. Task form Summary and description
3. Project form name and description
4. 2nd Create button after filling form/ for changes to be saved/reflected
5. Edit button for each project/ then apply button for changes to be saved/reflected
6. Delete button, for deleting the project/ Confirmation pop-up/apply for changes to be saved-reflected

CRUD Breakdown:

Operation	**UI Action**	**Expected Result**
Create	Click “Create”, fill form, Click Create again for changes to be saved-reflected	New task appears in list
Read	After task creation, a task card must appear	One card with per task with name Description status etc
Update	Click “Edit”, change fields, click “Update” for changes to be saved-reflected	Task info updated, reflected in list
Delete	Click "Delete", confirm action	Task is removed from list

Some validations that must be performed

1. Required fields: summary, description cannot be blank
2. Max field length
3. Invalid inputs: only spaces

2. Define test scenarios

Gherkin syntax is a domain-specific language for behavior-driven development (BDD) and will be used to define clear and human-readable test scenarios. As a result, will allow both technical and non-technical stakeholders (like QA, developers, product owners) to understand and collaborate on how software should behave.

Main goal of this part is to create feature files in plain human readable language to describe how the app should behave. Files that will be created will be gathered in a feature dir and follow .feature format.

Separate ".feature" files will be made for each of 4 major modules described before

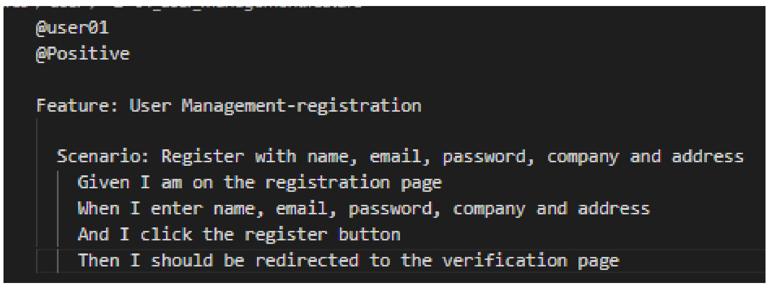
Module	**Feature files**
User management	01_user_management.feature 02_user_management.feature …..
Project Management	01_user_management.feature 02_user_management.feature …..
Task Management	01_task_CRUD.feature 02_task_CRUD.feature …..

Each file will contain scenarios for every user action.

As a first step with manually testing, need to locate the behavior of the PMtool and if it is based on requirements. Then it is time to create some Gherkin syntax test scenarios in order to later on transform them into python scripts for automated test cases. The following section presents **several example scenarios** from each feature, written using Gherkin syntax.

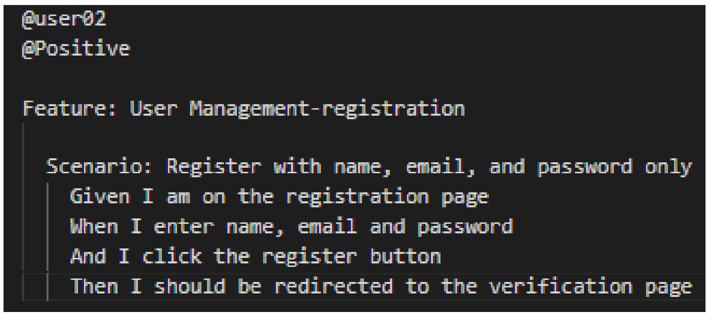
<u>Feature: User Management-registration:</u>

Positive scenarios:



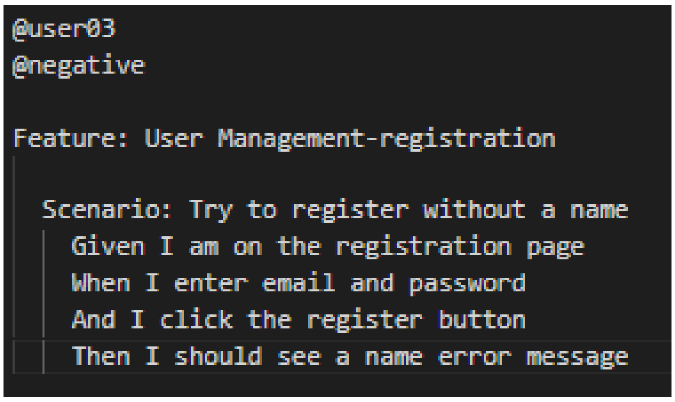
```
@user01
@Positive

Feature: User Management-registration

  Scenario: Register with name, email, password, company and address
    Given I am on the registration page
    When I enter name, email, password, company and address
    And I click the register button
    Then I should be redirected to the verification page
```

To be noted company and address are optionally

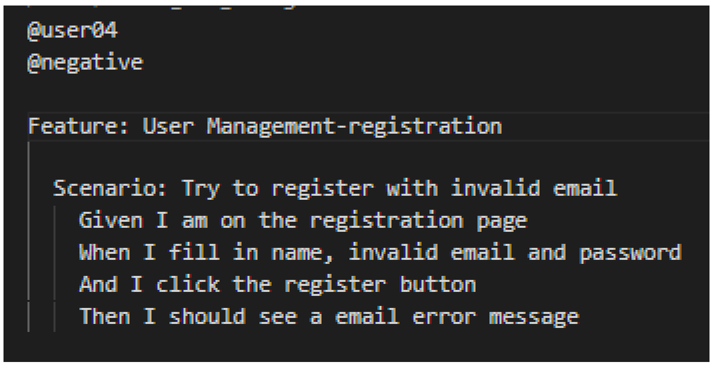


```
@user02
@Positive

Feature: User Management-registration

  Scenario: Register with name, email, and password only
    Given I am on the registration page
    When I enter name, email and password
    And I click the register button
    Then I should be redirected to the verification page
```

Negative scenarios:



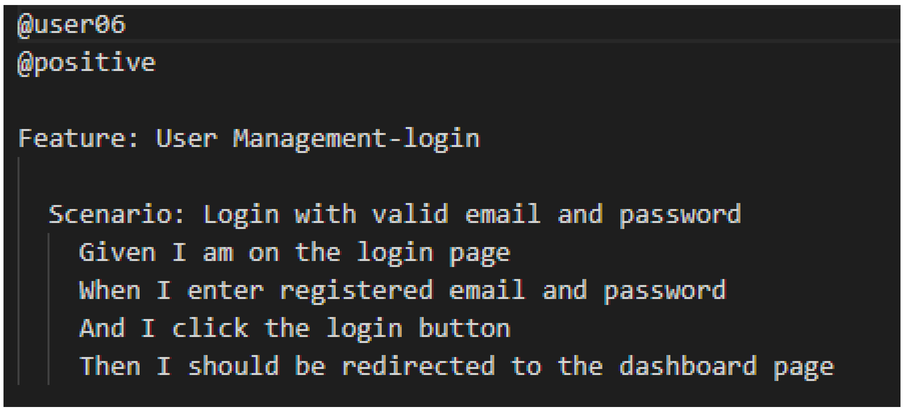
```
@user03
@negative

Feature: User Management-registration

  Scenario: Try to register without a name
    Given I am on the registration page
    When I enter email and password
    And I click the register button
    Then I should see a name error message
```



```
@user04
@negative

Feature: User Management-registration

  Scenario: Try to register with invalid email
    Given I am on the registration page
    When I fill in name, invalid email and password
    And I click the register button
    Then I should see a email error message
```

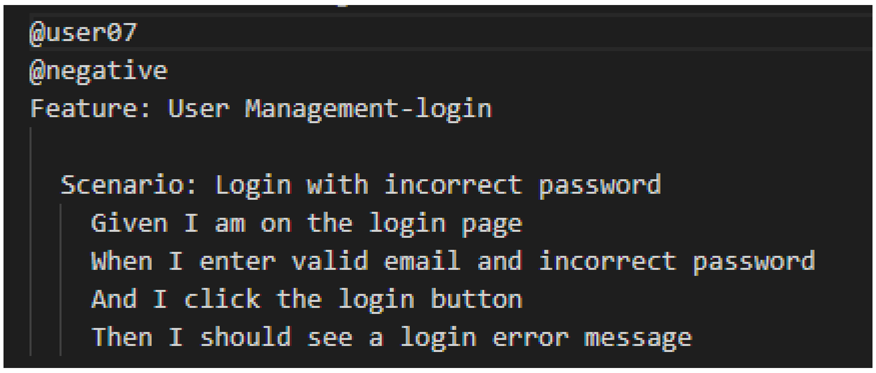
Feature: User Management-login:

Positive scenarios:

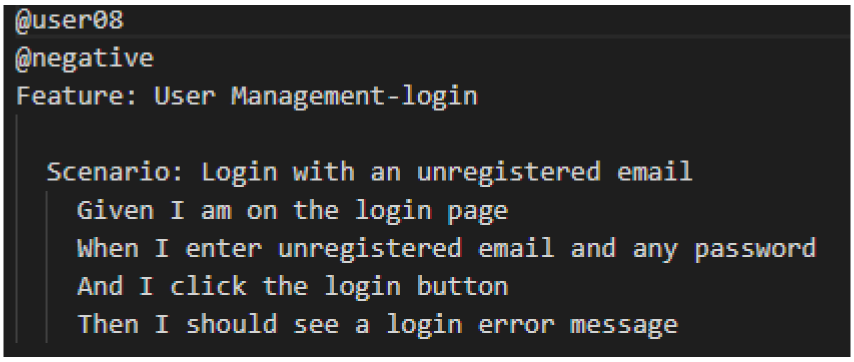


```
@user06
@positive

Feature: User Management-login

  Scenario: Login with valid email and password
    Given I am on the login page
    When I enter registered email and password
    And I click the login button
    Then I should be redirected to the dashboard page
```

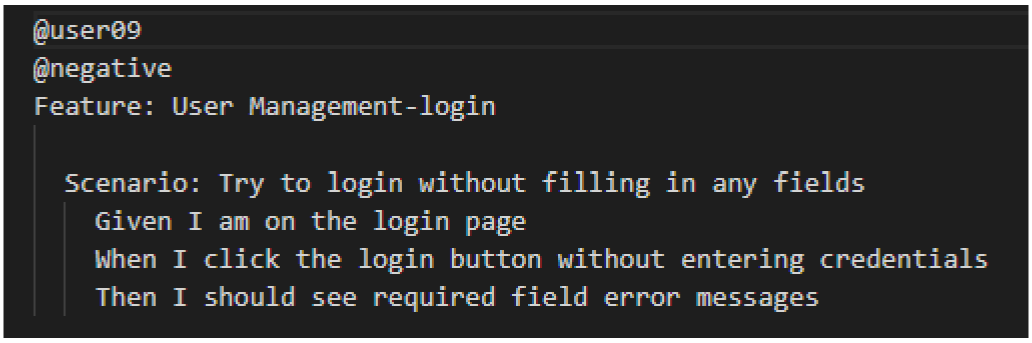
Negative scenarios:



```
@user07
@negative
Feature: User Management-login

  Scenario: Login with incorrect password
    Given I am on the login page
    When I enter valid email and incorrect password
    And I click the login button
    Then I should see a login error message
```



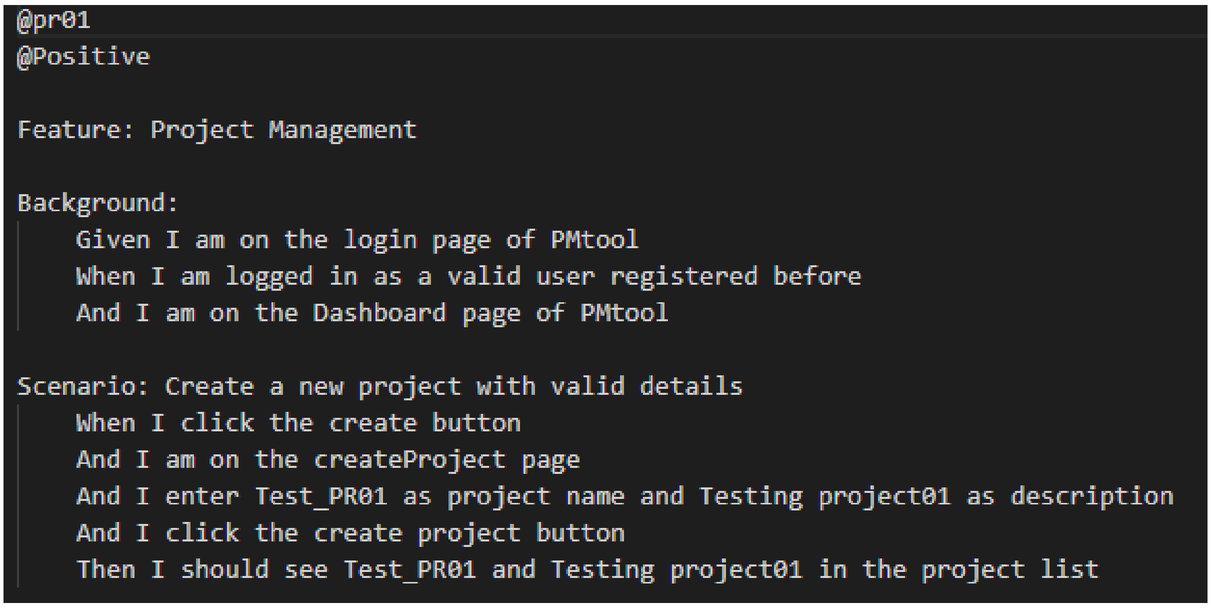
```
@user08
@negative
Feature: User Management-login

  Scenario: Login with an unregistered email
    Given I am on the login page
    When I enter unregistered email and any password
    And I click the login button
    Then I should see a login error message
```



```
@user09
@negative
Feature: User Management-login

  Scenario: Try to login without filling in any fields
    Given I am on the login page
    When I click the login button without entering credentials
    Then I should see required field error messages
```

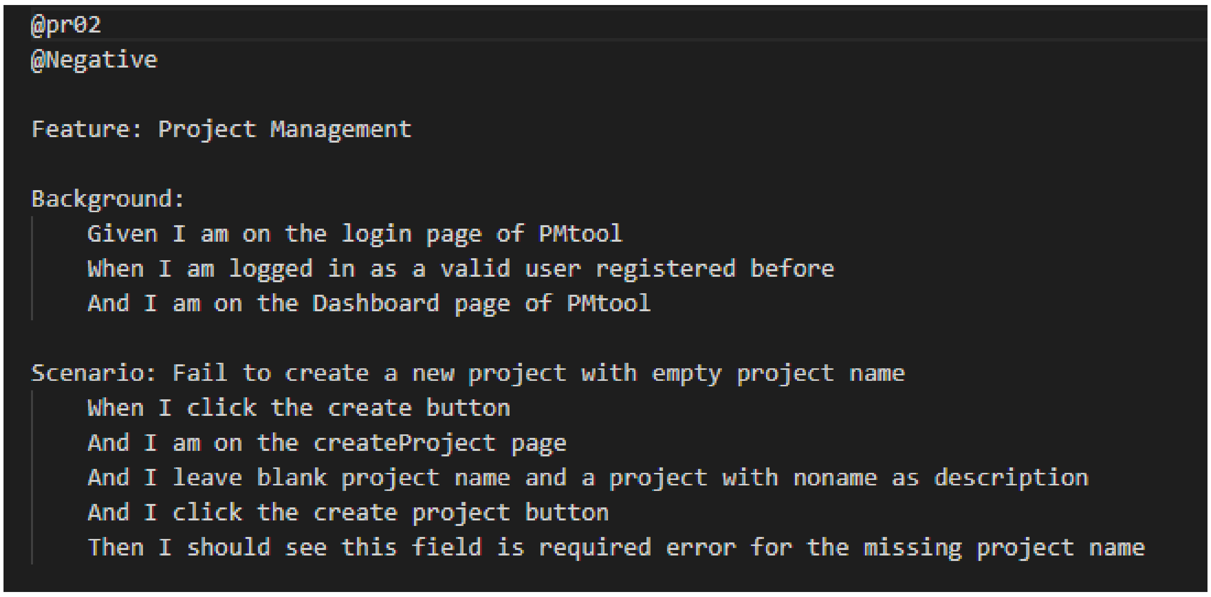
Feature: Project Management CRUD:

Positive scenarios:



```
@pr01
@Positive

Feature: Project Management

Background:
    Given I am on the login page of PMtool
    When I am logged in as a valid user registered before
    And I am on the Dashboard page of PMtool

Scenario: Create a new project with valid details
    When I click the create button
    And I am on the createProject page
    And I enter Test_PR01 as project name and Testing project01 as description
    And I click the create project button
    Then I should see Test_PR01 and Testing project01 in the project list
```

Negative scenarios:

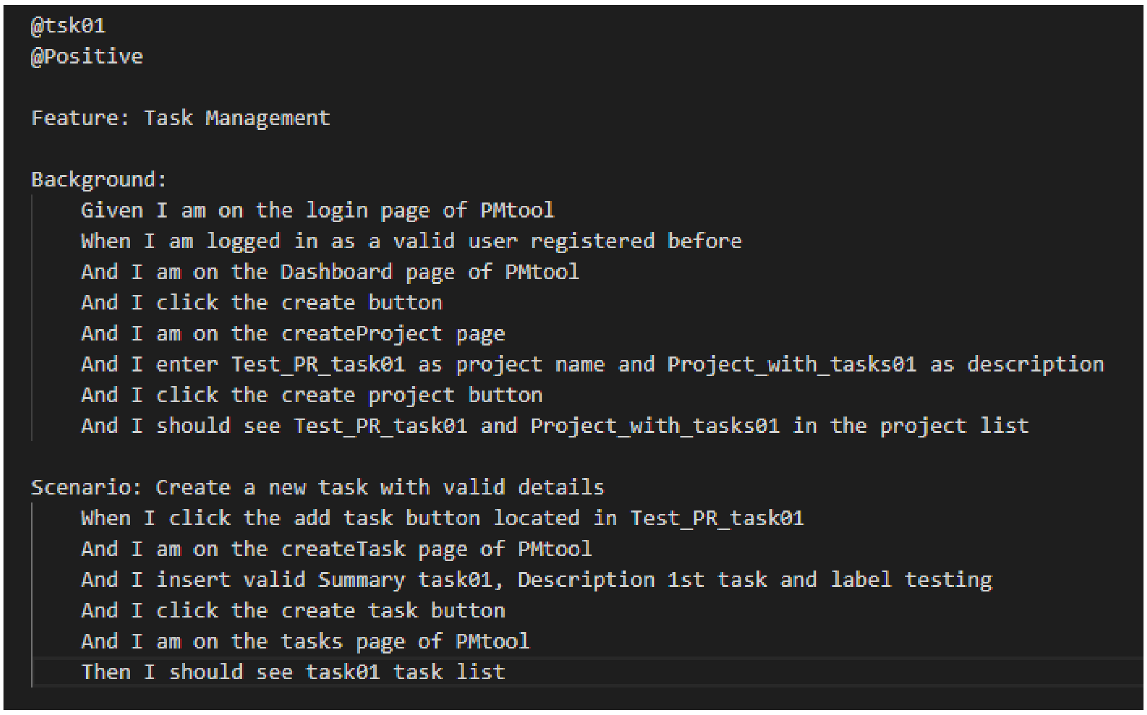


```
@pr02
@Negative

Feature: Project Management

Background:
    Given I am on the login page of PMtool
    When I am logged in as a valid user registered before
    And I am on the Dashboard page of PMtool

Scenario: Fail to create a new project with empty project name
    When I click the create button
    And I am on the createProject page
    And I leave blank project name and a project with noname as description
    And I click the create project button
    Then I should see this field is required error for the missing project name
```

More examples are included in test suite

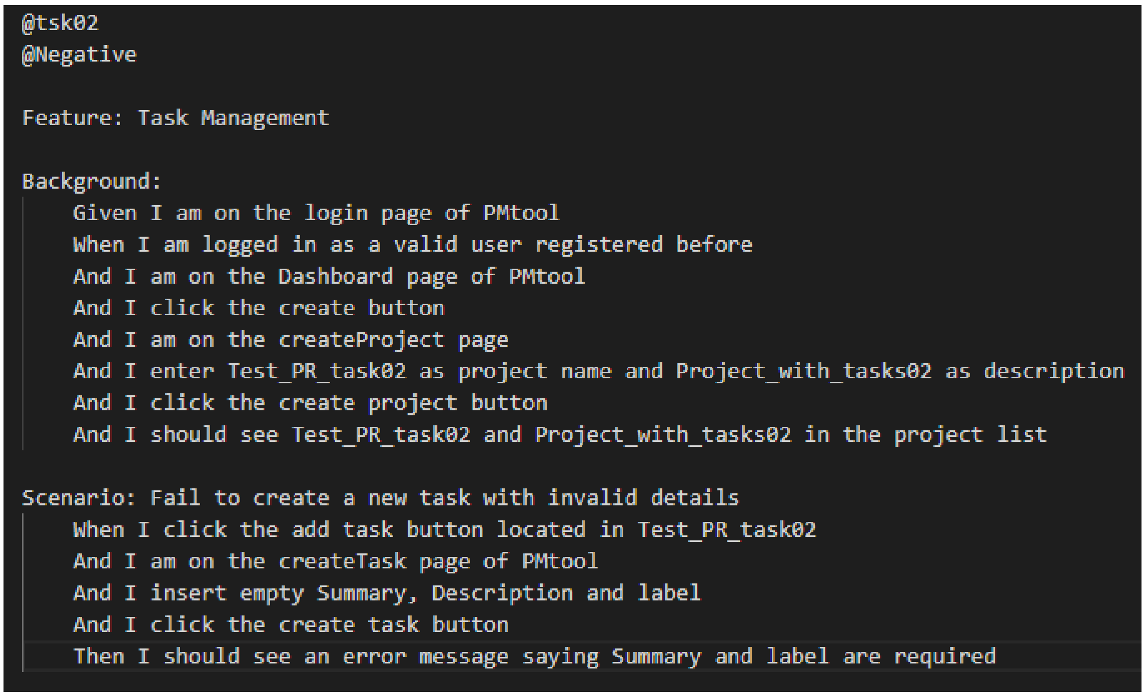
Feature: Task Management CRUD:

Positive scenarios:



```
@tsk01
@Positive

Feature: Task Management

Background:
    Given I am on the login page of PMtool
    When I am logged in as a valid user registered before
    And I am on the Dashboard page of PMtool
    And I click the create button
    And I am on the createProject page
    And I enter Test_PR_task01 as project name and Project_with_tasks01 as description
    And I click the create project button
    And I should see Test_PR_task01 and Project_with_tasks01 in the project list

Scenario: Create a new task with valid details
    When I click the add task button located in Test_PR_task01
    And I am on the createTask page of PMtool
    And I insert valid Summary task01, Description 1st task and label testing
    And I click the create task button
    And I am on the tasks page of PMtool
    Then I should see task01 task list
```

Negative scenarios:



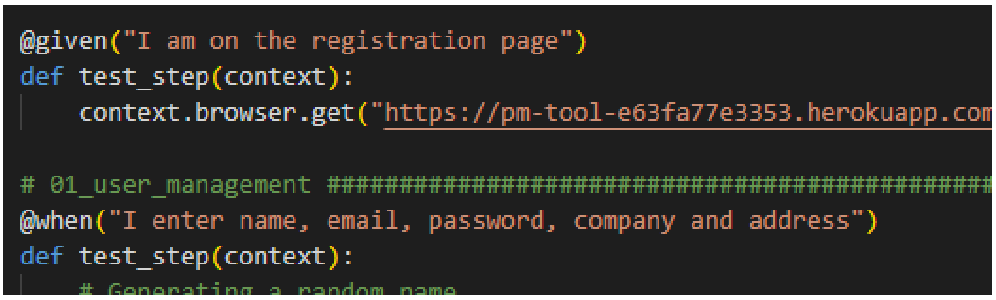
```
@tsk02
@Negative

Feature: Task Management

Background:
    Given I am on the login page of PMtool
    When I am logged in as a valid user registered before
    And I am on the Dashboard page of PMtool
    And I click the create button
    And I am on the createProject page
    And I enter Test_PR_task02 as project name and Project_with_tasks02 as description
    And I click the create project button
    And I should see Test_PR_task02 and Project_with_tasks02 in the project list

Scenario: Fail to create a new task with invalid details
    When I click the add task button located in Test_PR_task02
    And I am on the createTask page of PMtool
    And I insert empty Summary, Description and label
    And I click the create task button
    Then I should see an error message saying Summary and label are required
```

3. Implement Step Definitions

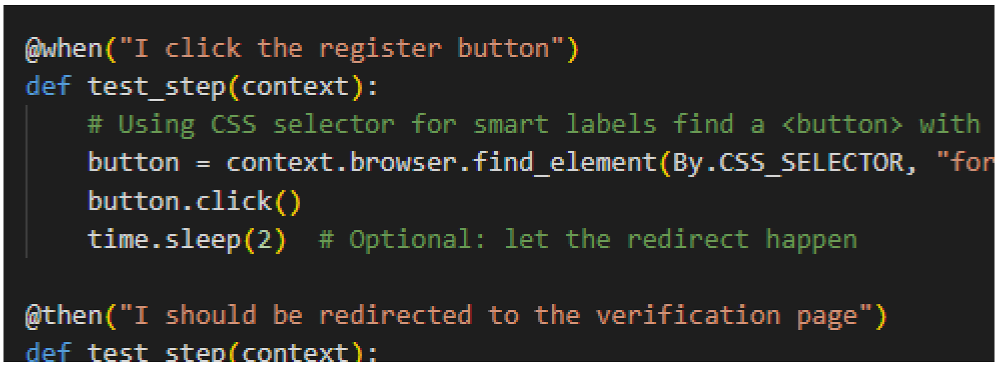
Step definition is crucial as it is the mapping process from text to a programming language functions that executes browser actions using web drivers.

For this project stack will be, python along with behave and selenium. The linking is done using decorators: @given, @when, @then

Given example below:



```
@given("I am on the registration page")
def test_step(context):
    context.browser.get("https://pm-tool-e63fa77e3353.herokuapp.com

# 01_user_management ##############################################
@when("I enter name, email, password, company and address")
def test_step(context):
    # Generating a random name
```



```
@when("I click the register button")
def test_step(context):
    # Using CSS selector for smart labels find a <button> with
    button = context.browser.find_element(By.CSS_SELECTOR, "for
    button.click()
    time.sleep(2)  # Optional: let the redirect happen

@then("I should be redirected to the verification page")
def test_step(context):
```

So, based on scenarios built in human language now there are implemented into code

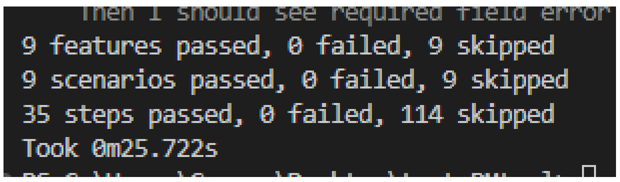
Using python, behave, selenium and some important web drivers.

To be noticed that when “And” is located in Gherkin syntax is for multiple When, so basically when someone is at specific location and when doing something then something else is happening as a result.

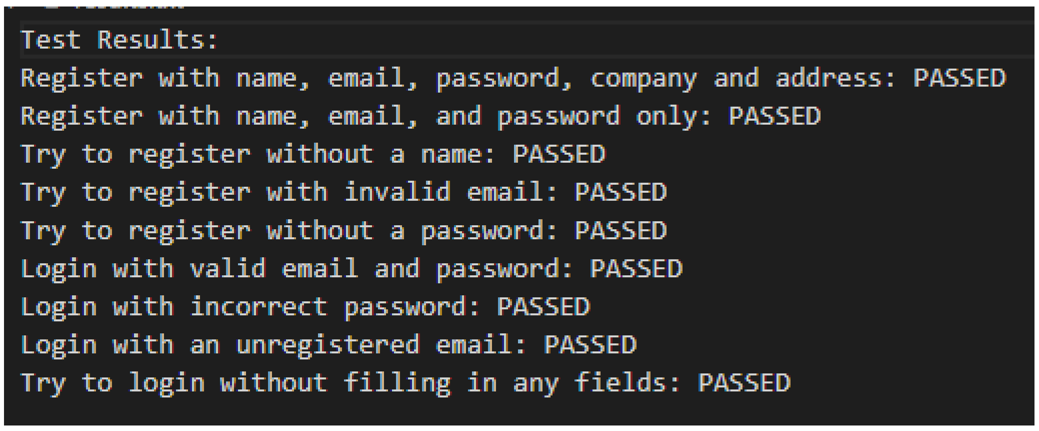
4. Results and discussion

Module 1:

For User management almost, everything seemed to work correctly, some important notes to keep is that 2fa and email was not working at all in registration verification in test environment (*I am not sure if there are not feasible to work because of test environment*). Below some results of test cases



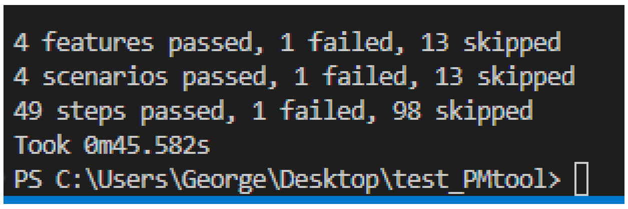
```
    Then I should see required field error
9 features passed, 0 failed, 9 skipped
9 scenarios passed, 0 failed, 9 skipped
35 steps passed, 0 failed, 114 skipped
Took 0m25.722s
```



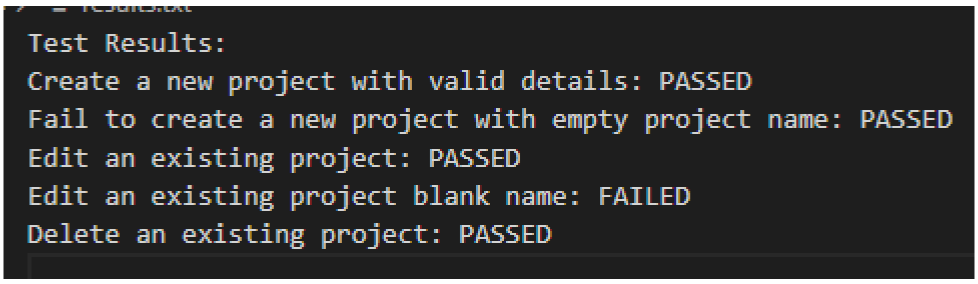
```
Test Results:
Register with name, email, password, company and address: PASSED
Register with name, email, and password only: PASSED
Try to register without a name: PASSED
Try to register with invalid email: PASSED
Try to register without a password: PASSED
Login with valid email and password: PASSED
Login with incorrect password: PASSED
Login with an unregistered email: PASSED
Try to login without filling in any fields: PASSED
```

Module 2:

For Project Management also 4/5 worked correctly except some things like creating projects with the same name and updating existing project removing name. Below results:

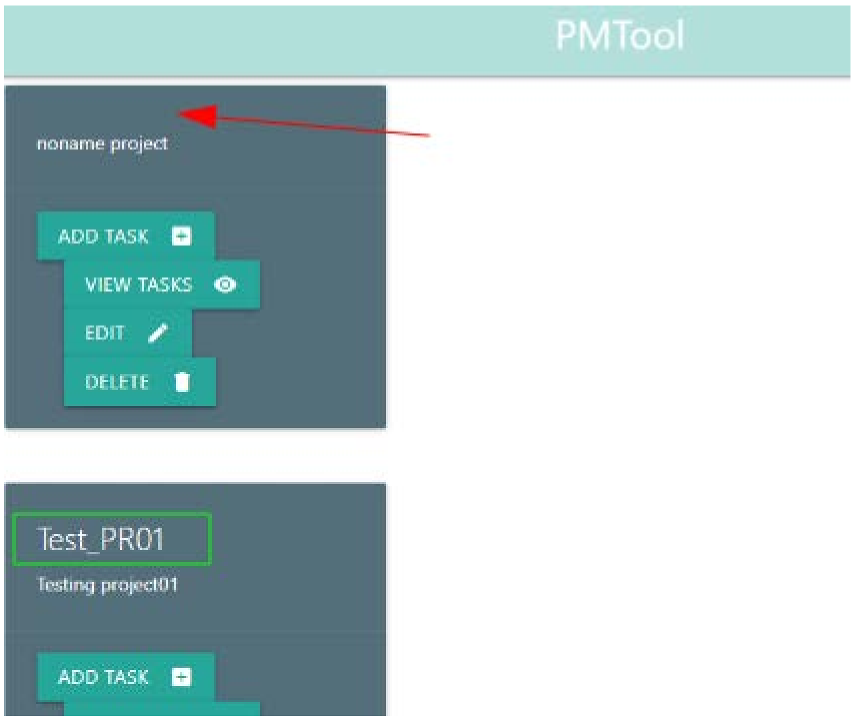


```
4 features passed, 1 failed, 13 skipped
4 scenarios passed, 1 failed, 13 skipped
49 steps passed, 1 failed, 98 skipped
Took 0m45.582s
PS C:\Users\George\Desktop\test_PMtool> 
```



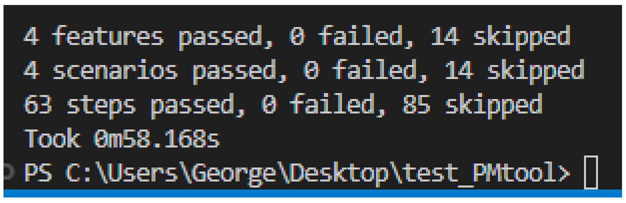
```
Test Results:
Create a new project with valid details: PASSED
Fail to create a new project with empty project name: PASSED
Edit an existing project: PASSED
Edit an existing project blank name: FAILED
Delete an existing project: PASSED
```

As for Edit an existing project

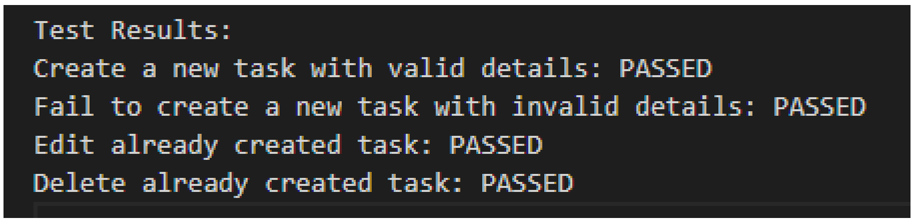


Module 3:

For Task Management checked CRUD as expected. Some issues on repeat like project management. Below some results of test cases



```
4 features passed, 0 failed, 14 skipped
4 scenarios passed, 0 failed, 14 skipped
63 steps passed, 0 failed, 85 skipped
Took 0m58.168s
PS C:\Users\George\Desktop\test_PMtool>
```



```
Test Results:
Create a new task with valid details: PASSED
Fail to create a new task with invalid details: PASSED
Edit already created task: PASSED
Delete already created task: PASSED
```

More instructions on how to run the suite will be at Readme file