Absolutely! Let's break down how to apply Gherkin feature files to your automated Cypress tests in detailed steps:

**1. Install Cucumber Plugin:**

* **Install:** In your terminal, run npm install --save-dev cypress-cucumber-preprocessor to install the Cucumber plugin for Cypress.
* **Configuration:** Update your cypress.config.js file to include the following:

JavaScript

const cucumber = require('cypress-cucumber-preprocessor').default;

module.exports = defineConfig({

e2e: {

// ... other configuration

setupNodeEvents(on, config) {

on('file:preprocessor', cucumber());

},

},

});

Use code [with caution.](/faq#coding)

content\_copy

**2. Create Feature Files:**

* **Location:** Create a new folder called features in your cypress/e2e directory.
* **Feature File:** Inside the features folder, create a file with the .feature extension (e.g., profileCreation.feature).
* **Write Scenarios:** Use Gherkin syntax to describe your test scenarios. Here's an example for your TC1:

Gherkin

Feature: Profile Creation

Scenario: Save profile without entering any data

Given I am on the profile creation page

When I click the "Save" button

Then I should see an error message for the first name field

And I should see an error message for the last name field

And I should see an error message for the email field

Use code [with caution.](/faq#coding)

content\_copy

**3. Implement Step Definitions:**

* **Location:** Create a step\_definitions folder in your cypress/e2e directory.
* **Step Definition File:** Create a JavaScript file (e.g., profileCreationSteps.js) inside the step\_definitions folder.
* **Define Steps:** Write Cypress code to implement each step from your feature file. Example for the "Given" step:

JavaScript

import { Given } from 'cypress-cucumber-preprocessor/steps';

import ProfileCreatePage from '../pages/profileCreate.page.js';

Given('I am on the profile creation page', () => {

cy.visit('https://testingtasks.kwentra.com/frontoffice/#/profileslist/add');

});

Use code [with caution.](/faq#coding)

content\_copy

**4. Run Tests:**

* **Command:** Run your tests using the command:

Bash

npx cypress run --spec "cypress/e2e/features/\*\*/\*.feature"

Use code [with caution.](/faq#coding)

content\_copy

**Structure Example:**

cypress/

├── e2e/

│ ├── features/

│ │ └── profileCreation.feature

│ ├── pages/

│ │ ├── profileCreate.page.js

│ │ └── signInPage.page.js

│ ├── step\_definitions/

│ │ └── profileCreationSteps.js

│ ├── support/

│ └── plugins/

├── fixtures/

├── support/

└── cypress.config.js

**Advantages of Using Gherkin Feature Files:**

* **Readability:** Test scenarios are written in plain English, making them easy for non-technical stakeholders to understand.
* **Collaboration:** Gherkin promotes collaboration between developers, testers, and product owners.
* **Living Documentation:** Feature files serve as living documentation of your application's behavior.
* **Reusability:** Steps can be reused across different scenarios, reducing code duplication.
* **BDD Alignment:** Gherkin aligns well with Behavior-Driven Development (BDD) practices.

If you face any specific issues during implementation, feel free to ask for further assistance.

tuneshare

more\_vert

**Upload an image**

This prompt requires you to add an image. To add an image, select add\_circle.

Got it

**Do you need a little help with this prompt?**

Power up your prompt and Gemini will expand it to get you better results

Got it

expand\_content

add\_circle

mic

send

Gemini may display inaccurate