**AZURE DEVOPS ENVIRONMENT SETUP**

**EXP NO: 1**

**Aim:**

Create Epic, Features, User Stories, Task

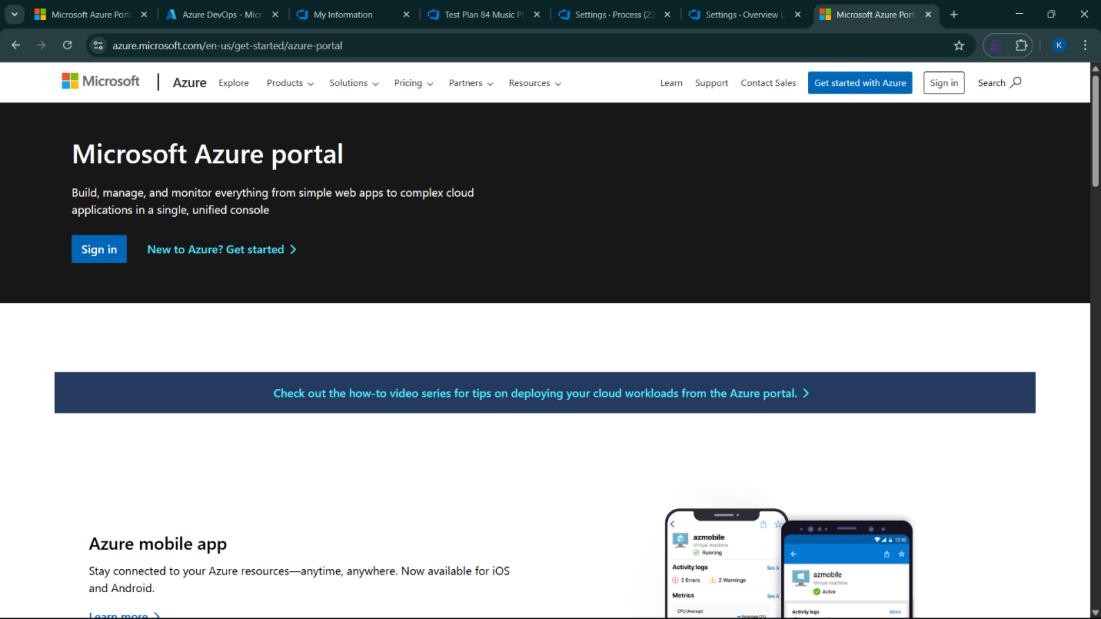
To set up and access the Azure DevOps environment by creating an organization through the Azure portal.

**INSTALLATION**

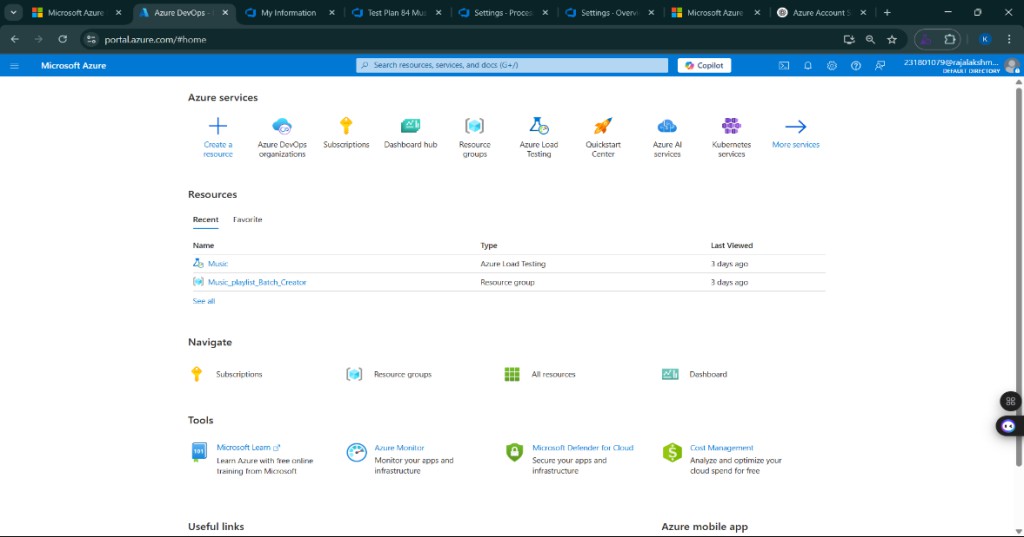
1. Open your web browser and go to the Azure website: [https://azure.microsoft.com/en-us/get-](https://azure.microsoft.com/en-us/get-started/azure-portal) [started/azure-portal](https://azure.microsoft.com/en-us/get-started/azure-portal).

Sign in using your Microsoft account credentials.

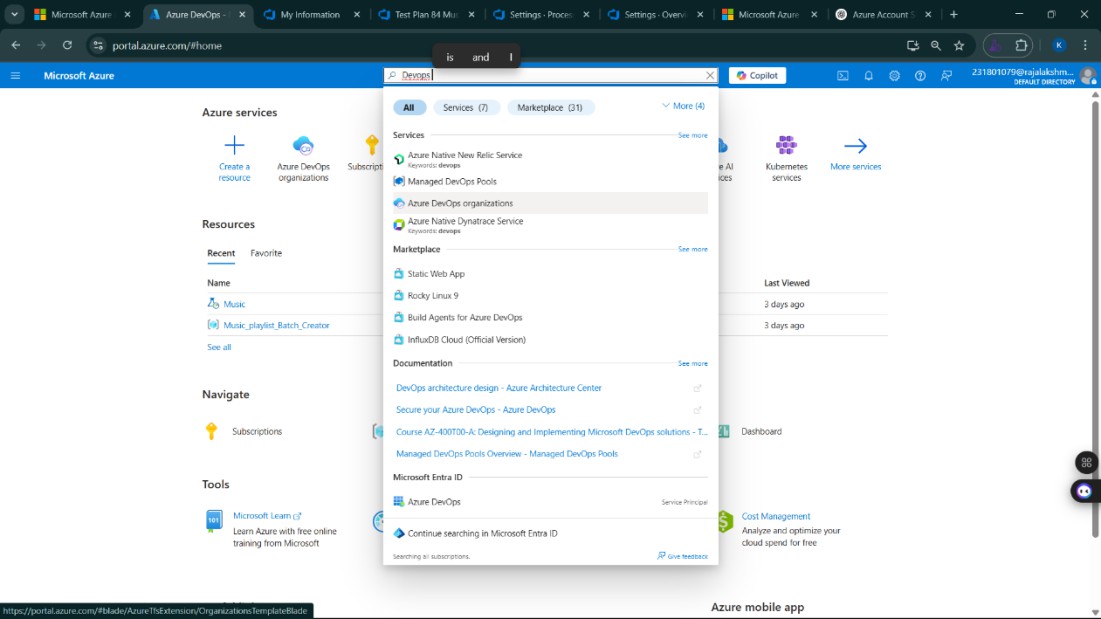
If you don't have a Microsoft account, you can create one here: <https://signup.live.com/?lic=1>

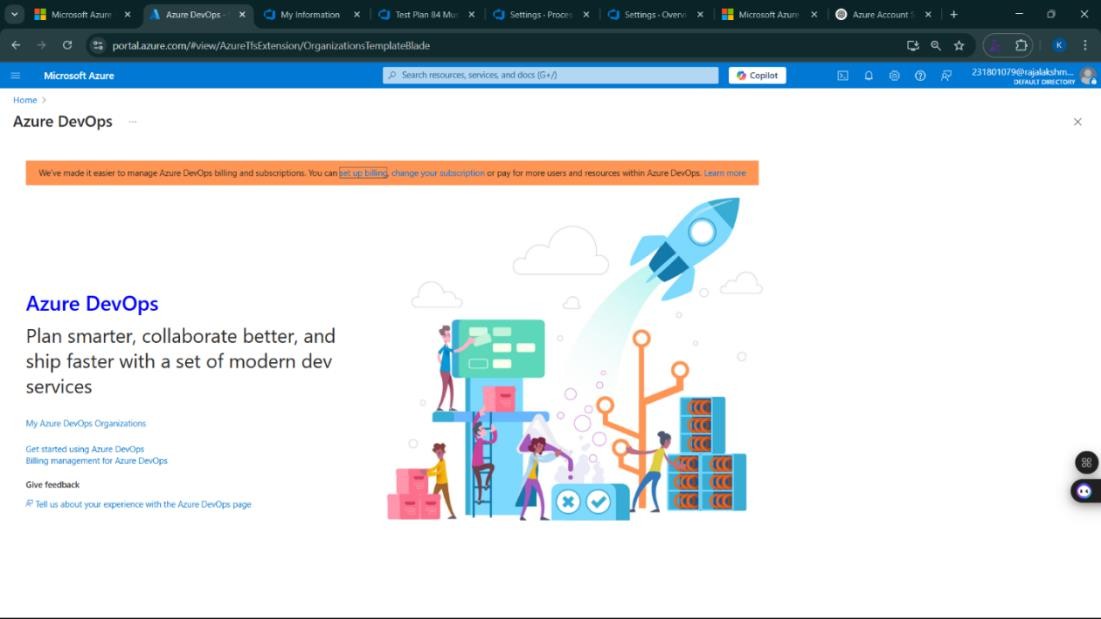


1. Azure home page



1. Open DevOps environment in the Azure platform by typing ***Azure DevOps Organizations*** in the search bar.



1. Click on the ***My Azure DevOps Organization*** link and create an organization and you should be taken to the Azure DevOps Organization Home page.

**Result:**

Successfully accessed the Azure DevOps environment and created a new organization through the Azure portal.

**AZURE DEVOPS PROJECT SETUP AND USER STORY MANAGEMENT**

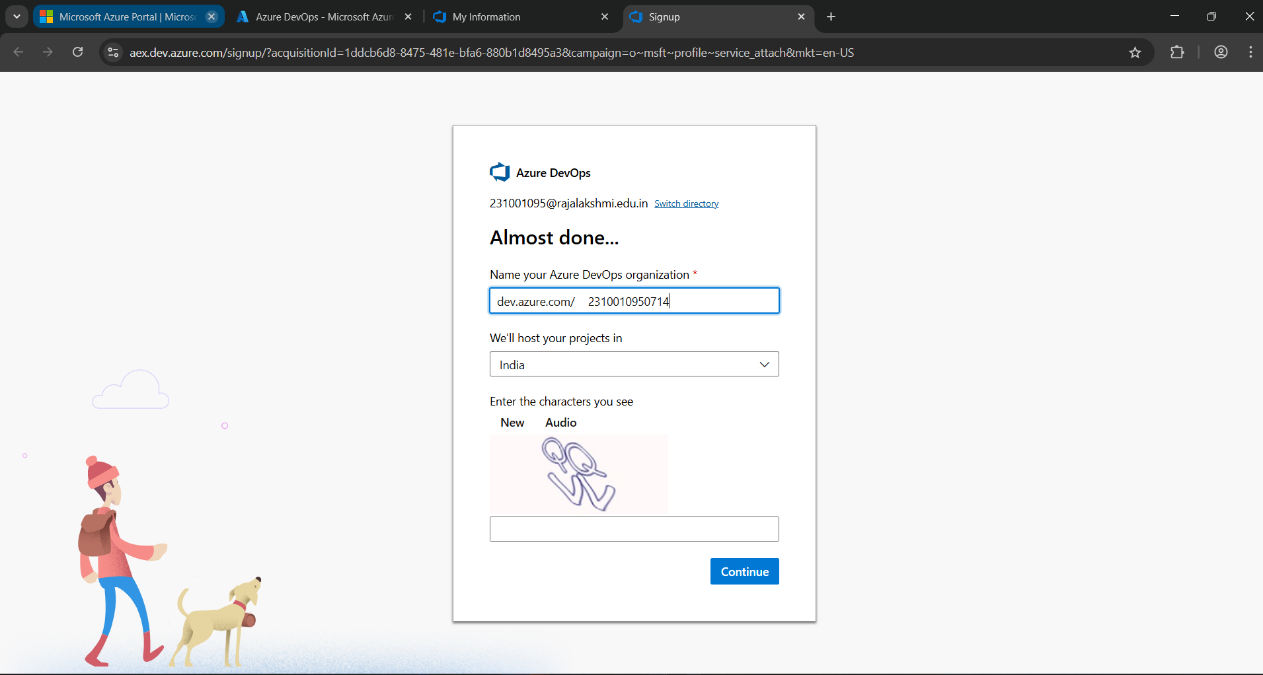
**EXP NO: 2**

**Aim:**

Create Epic, Features, User Stories, Task

To set up an Azure DevOps project for efficient collaboration and agile work management.

1. Create An Azure Account

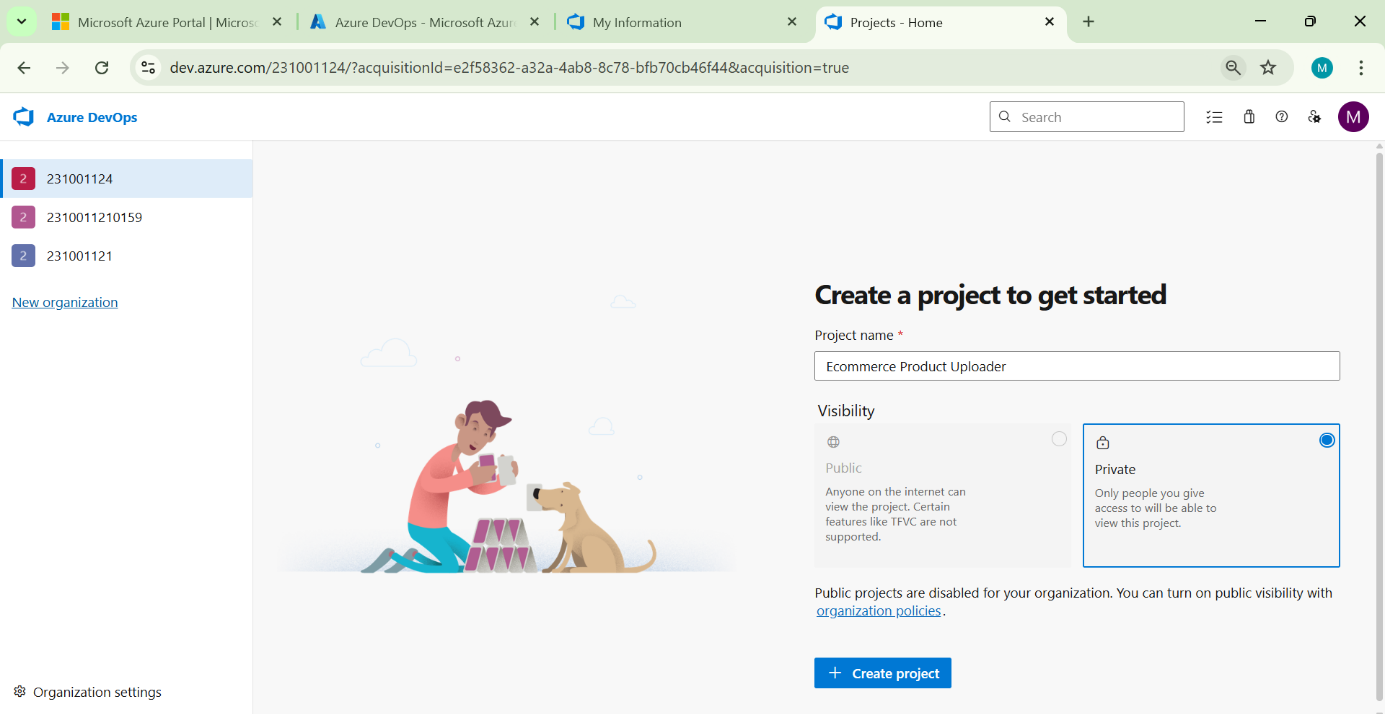


1. Create the First Project in Your Organization
   1. After the organization is set up, you’ll need to create your first **project**. This is where you'll begin to manage code, pipelines, work items, and more.
   2. On the organization’s **Home page**, click on the **New Project** button.
   3. Enter the project name, description, and visibility options:

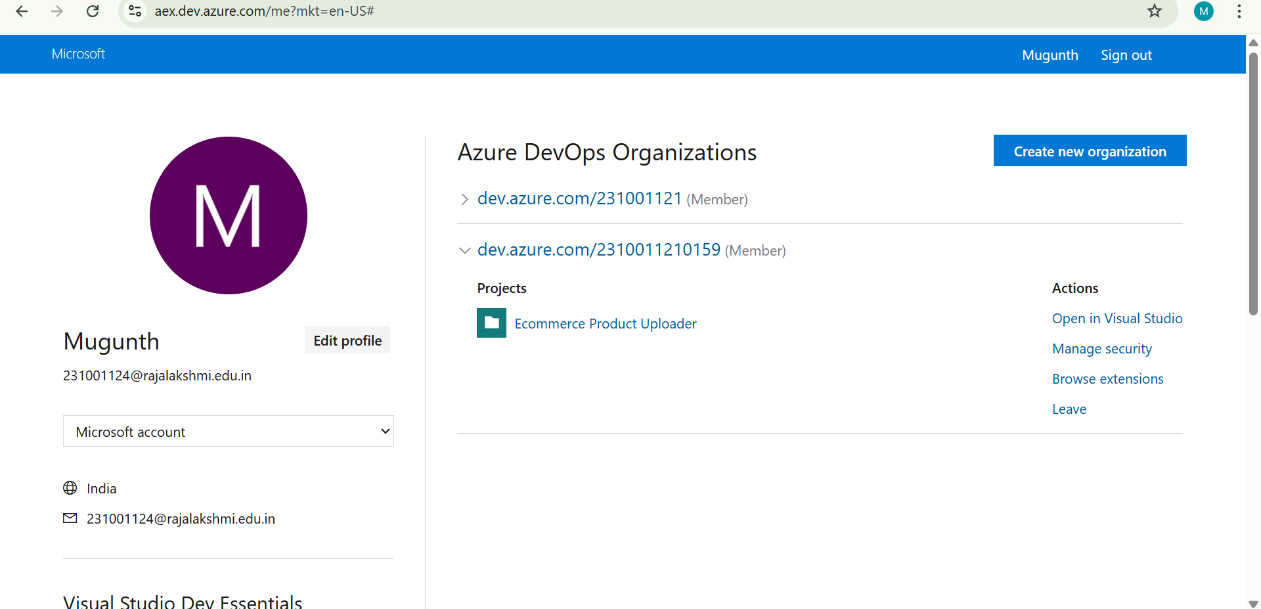
***Name****:* Choose a name for the project (e.g., Ecommerce Product Uploader).

***Description****:* Optionally, add a description to provide more context about the project. ***Visibility****:* Choose whether you want the project to be **Private** (accessible only to those invited) or **Public** (accessible to anyone).

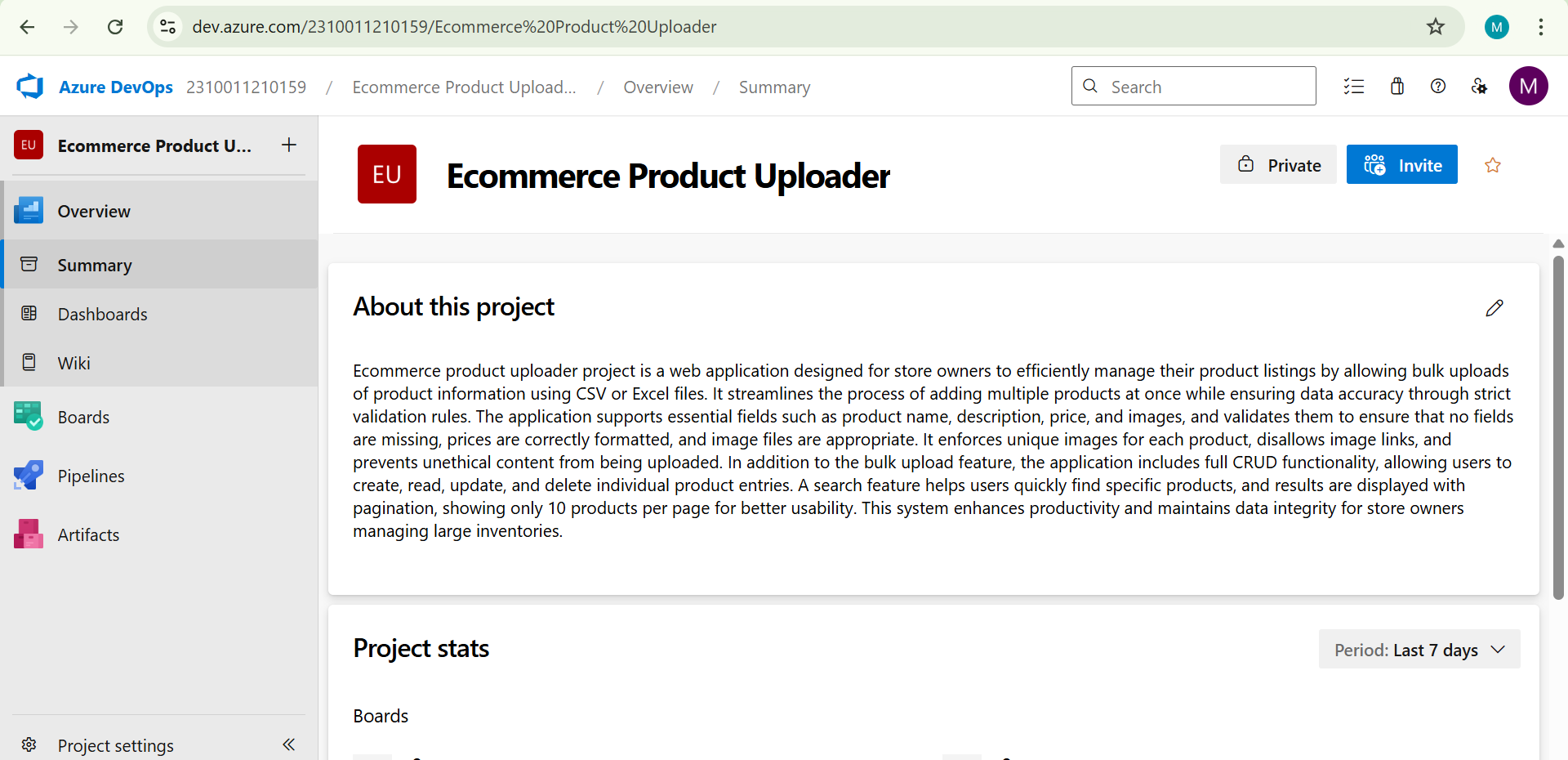
* 1. Once you’ve filled out the details, click **Create** to set up your first project.



1. Once logged in, ensure you are in the correct organization. If you're part of multiple organizations, you can switch between them from the top left corner (next to your user profile). Click on the Organization name, and you should be taken to the Azure DevOps Organization Home page.



1. Project dashboard

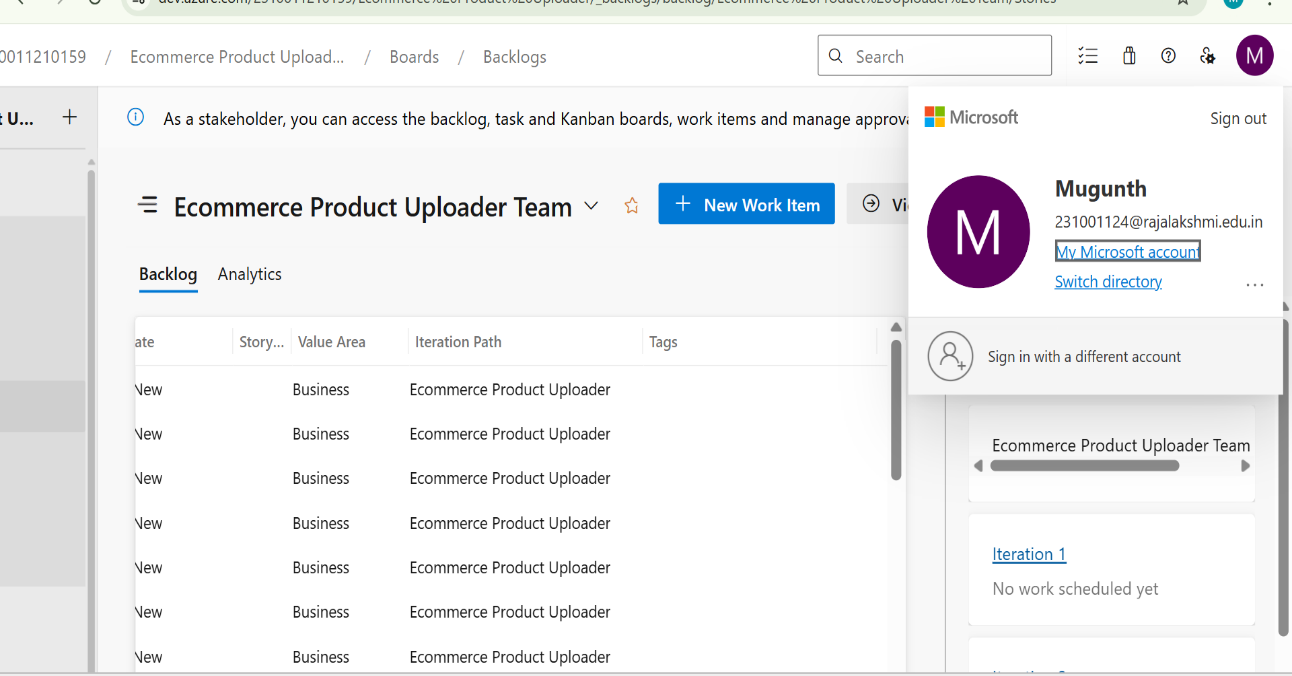


5.To manage user stories:

* 1. From the **left-hand navigation menu**, click on **Boards**. This will take you to the main **Boards**

page, where you can manage work items, backlogs, and sprints.

* 1. On the **work items** page, you'll see the option to **Add a work item** at the top. Alternatively, you can find a **+** button or **Add New Work Item** depending on the view you're in. From the **Add a work item** dropdown, select **User Story**. This will open a form to enter details for the new User Story.



**Result:**

setup.

Successfully created an Azure DevOps project with user story management and agile workflow

**SETTING UP EPICS, FEATURES, AND USER STORIES FOR PROJECT PLANNING**

**EXP NO: 3**

**Aim:**

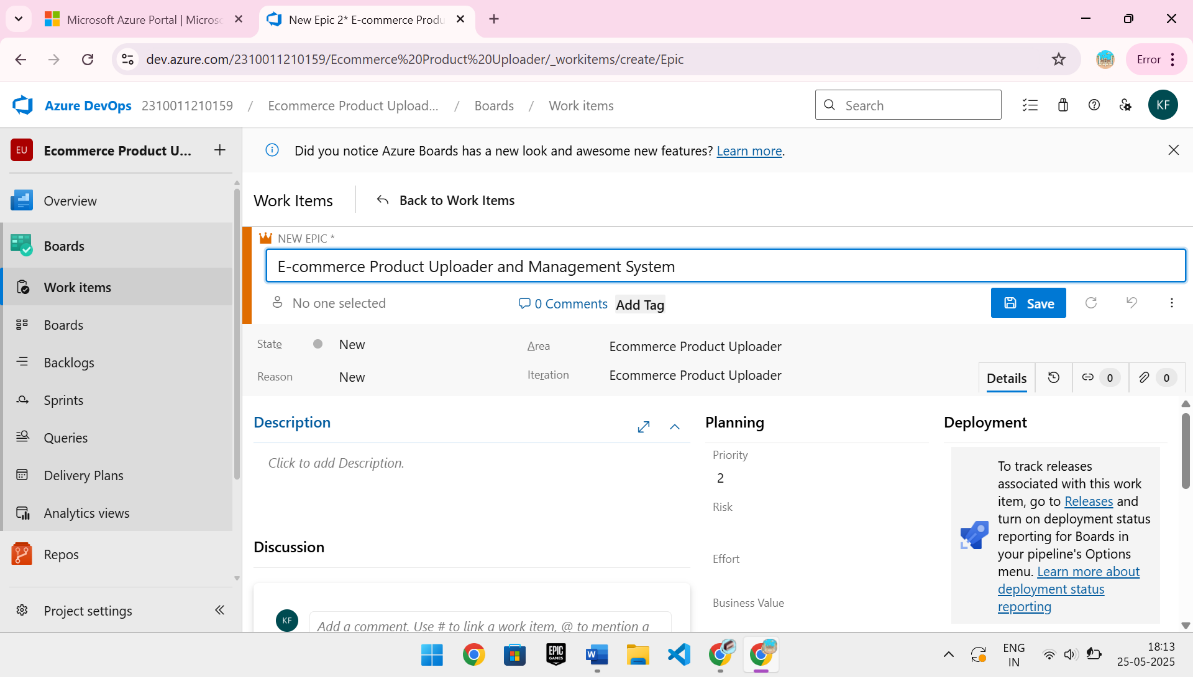
Create Epic, Features, User Stories, Task

To learn about how to create epics, user story, features, backlogs for your assigned project.

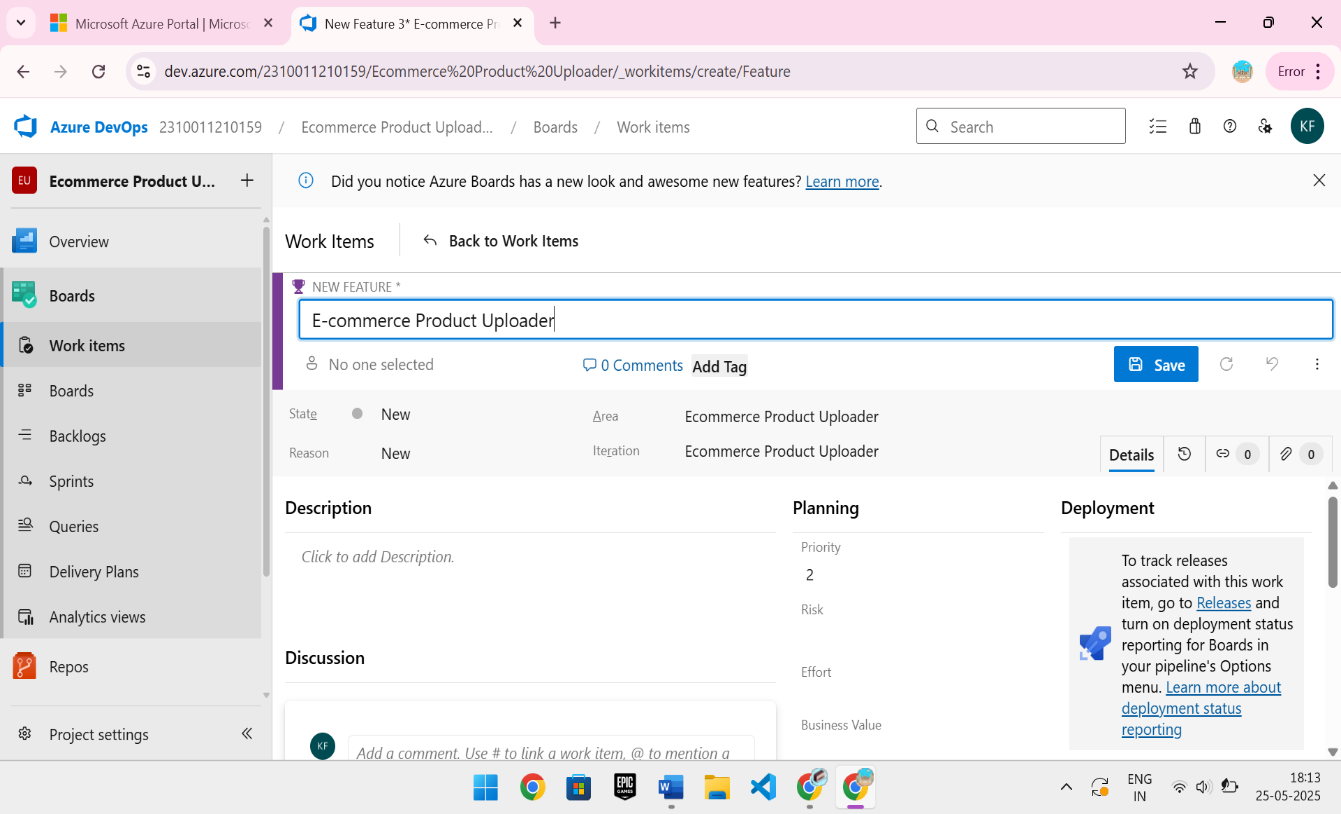
**Create Epic, Features, User Stories, Task**

****

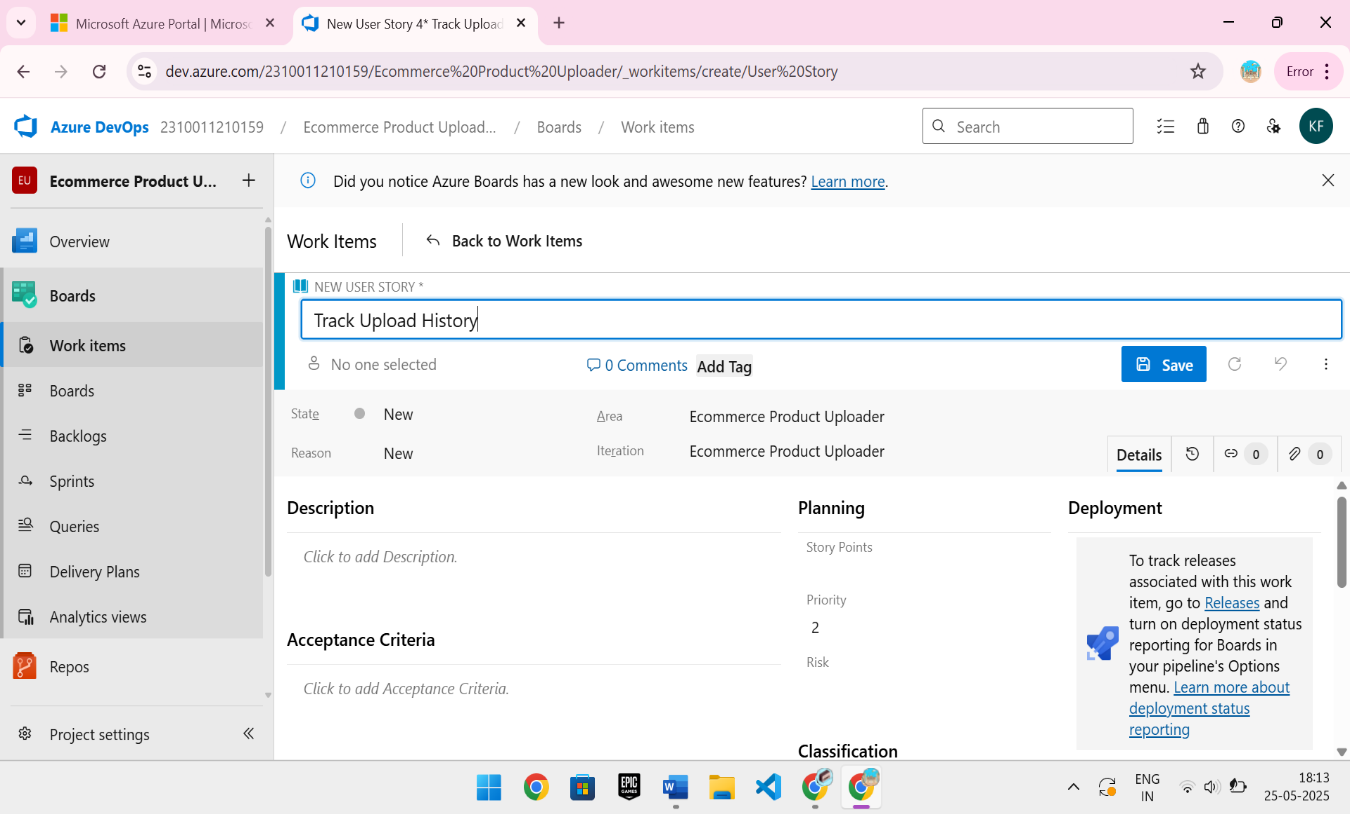
1. **Fill in Epics**

****

1. **Fill in Features**

****

1. **Fill in User Story Details**

****

**Result:**

Thus, the creation of epics, features, user story and task has been created successfully.

**SPRINT PLANNING**

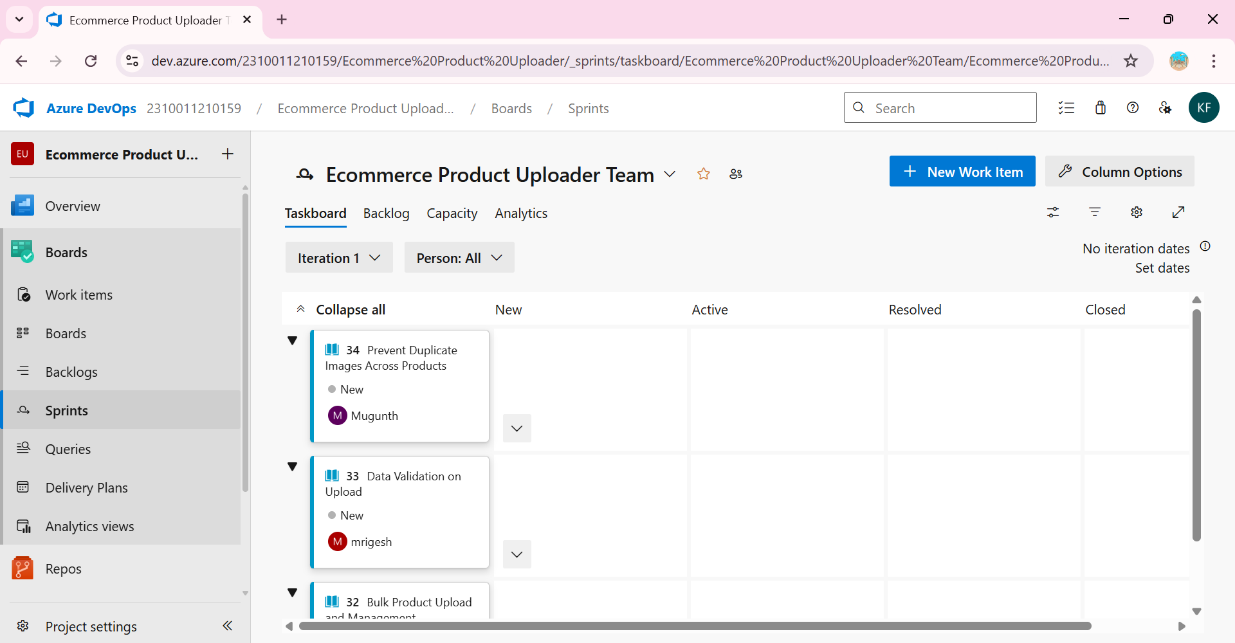
**EXP NO: 4**

**Aim:**

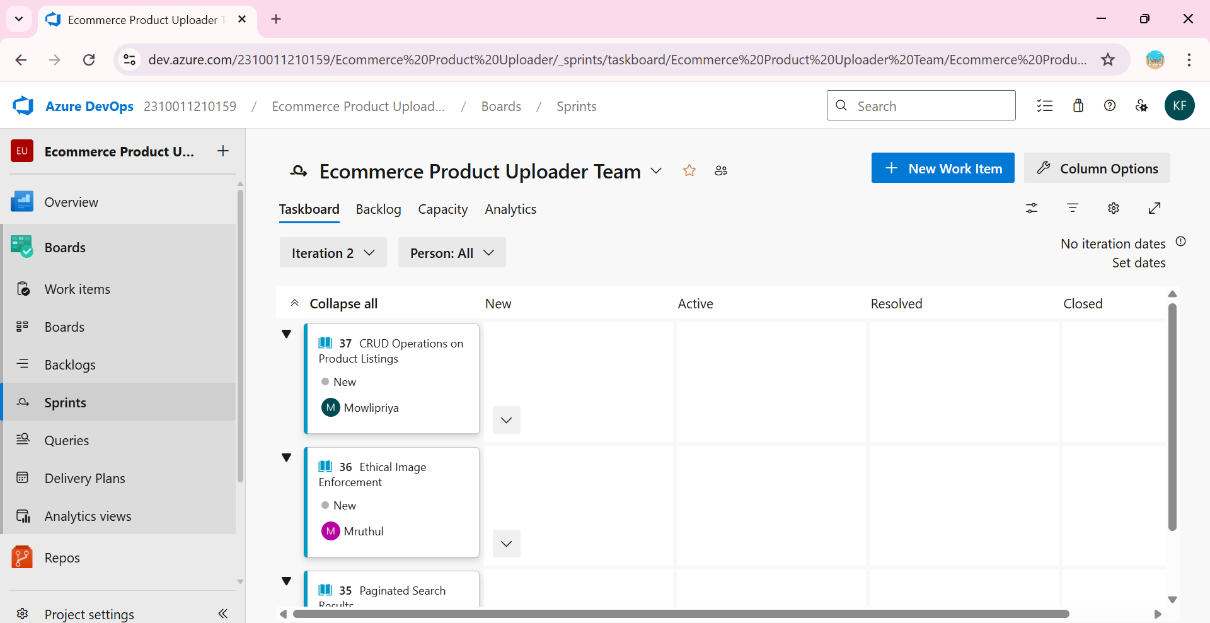
Create Epic, Features, User Stories, Task

To assign user story to specific sprint for the Ecommerce Product Uploader project.

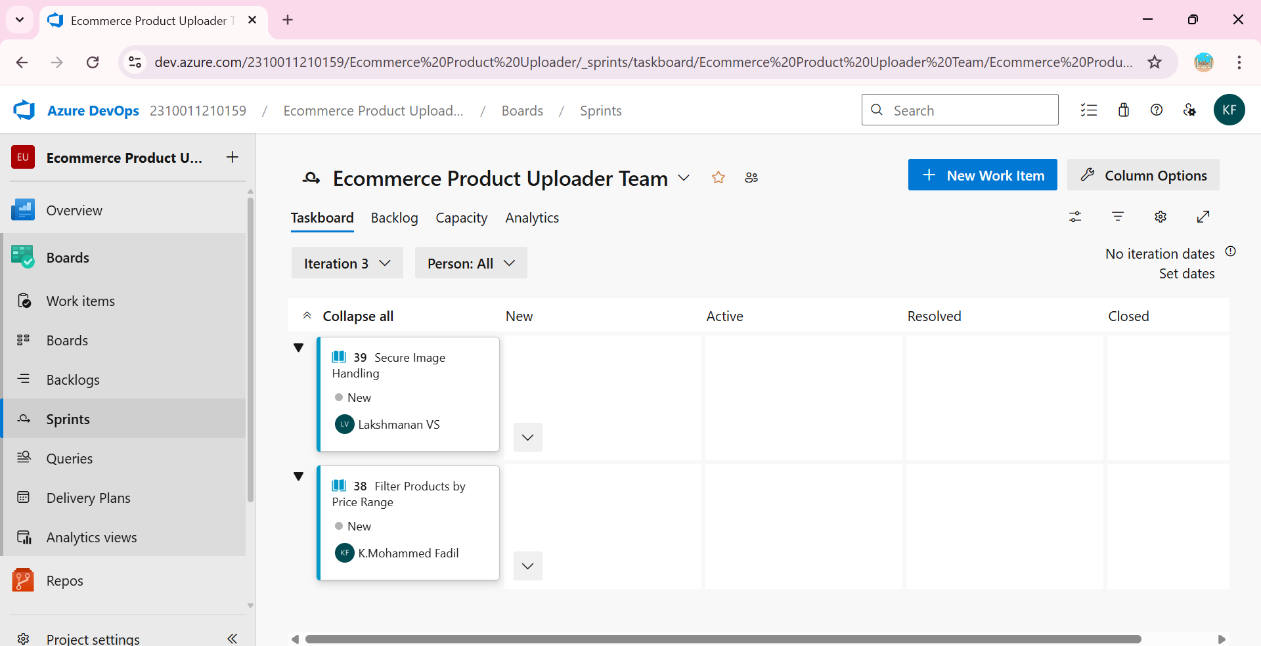
**Sprint Planning Sprint 1**

****

**Sprint 2**

****

**Sprint 3**

****

**Result:**

The Sprints are created for the Ecommerce Product Uploader project .

**POKER ESTIMATION**

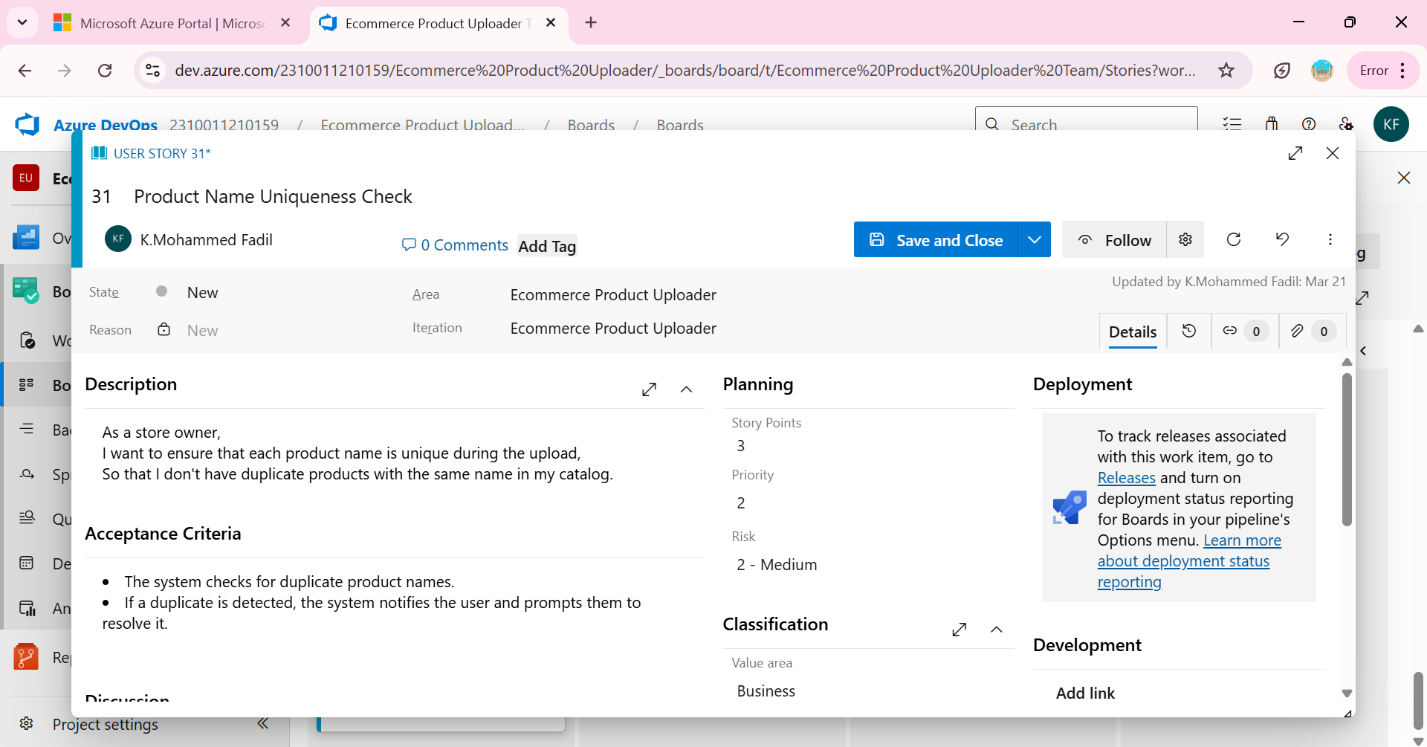
**EXP NO: 5**

**Aim:**

Create Epic, Features, User Stories, Task

Create Poker Estimation for the user stories - Ecommerce Product Uploader project.

**Poker Estimation**

****

**Result:**

The Estimation/Story Points is created for the project using Poker Estimation.

**DESIGNING CLASS AND SEQUENCE DIAGRAMS FOR PROJECT ARCHITECTURE**

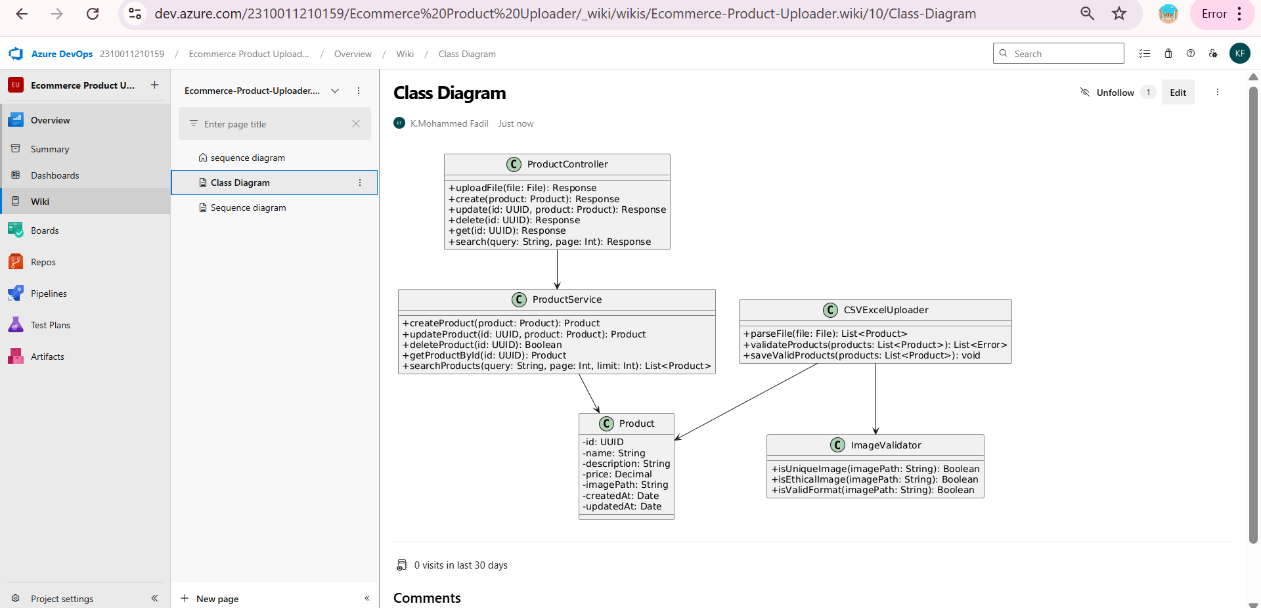
**EXP NO: 6**

**Aim:**

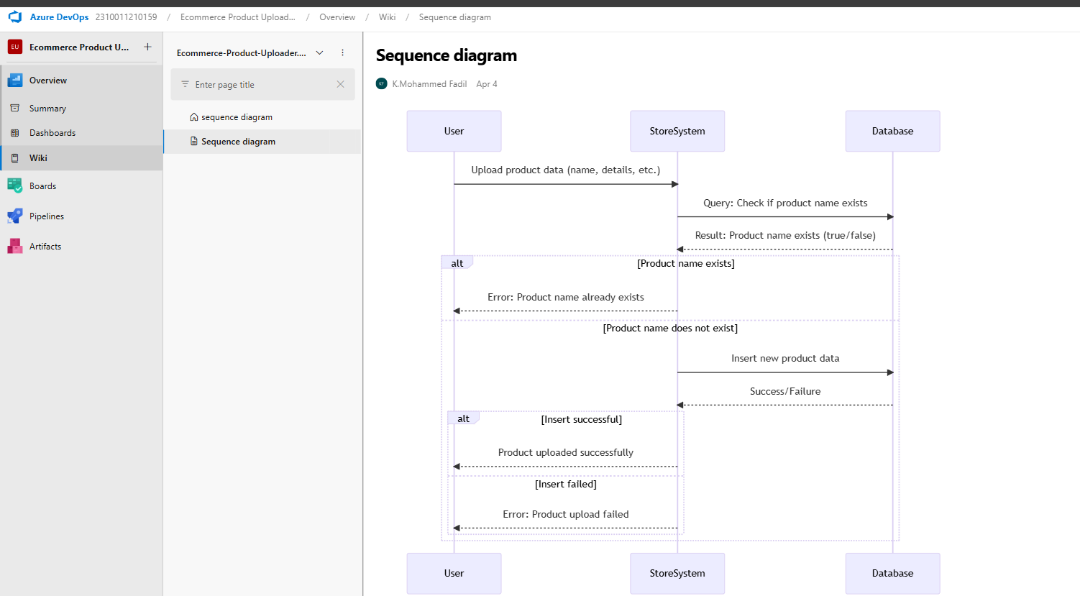
Create Epic, Features, User Stories, Task

To Design a Class Diagram and Sequence Diagram for the given Project.

**6A. Class Diagram**

****

**6B. Sequence Diagram**

****

**Result:**

The Class Diagram and Sequence Diagram is designed Successfully for the Ecommerce Product Uploader project

**DESIGNING ARCHITECTURAL AND ER DIAGRAMS FOR PROJECT STRUCTURE**

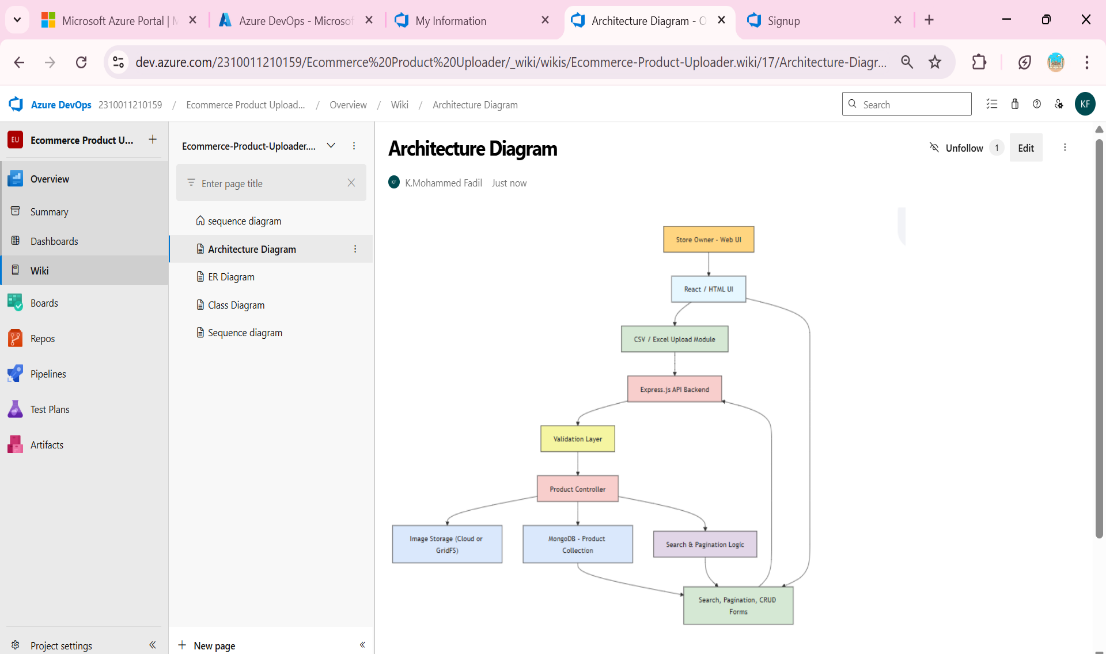
**EXP NO: 7**

**Aim:**

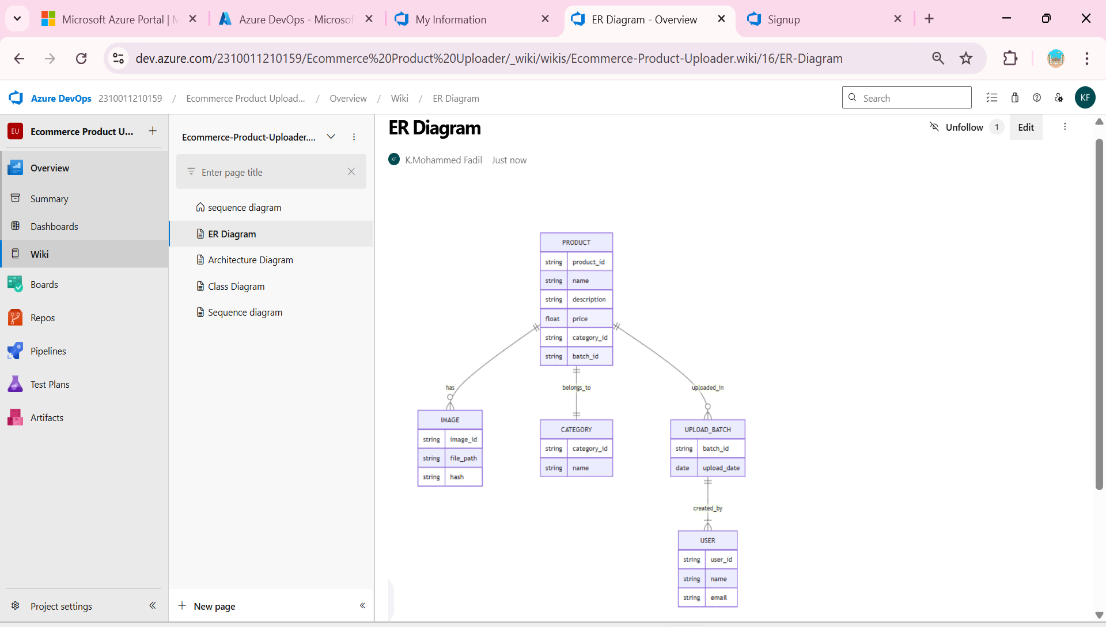
Create Epic, Features, User Stories, Task

To Design an Architectural Diagram and ER Diagram for the given Project.

**7A. Architectural Diagram**

****

**7B.ER Diagram**

****

**Result:**

The Architecture Diagram and ER Diagram is designed Successfully for Ecommerce Product Uploader project.

**TESTING – TEST PLANS AND TEST CASES**

**EXP NO: 8**

**Aim:**

Create Epic, Features, User Stories, Task

Test Plans and Test Case and write two test cases for at least five user stories showcasing the

happy path and error scenarios in azure DevOps platform.

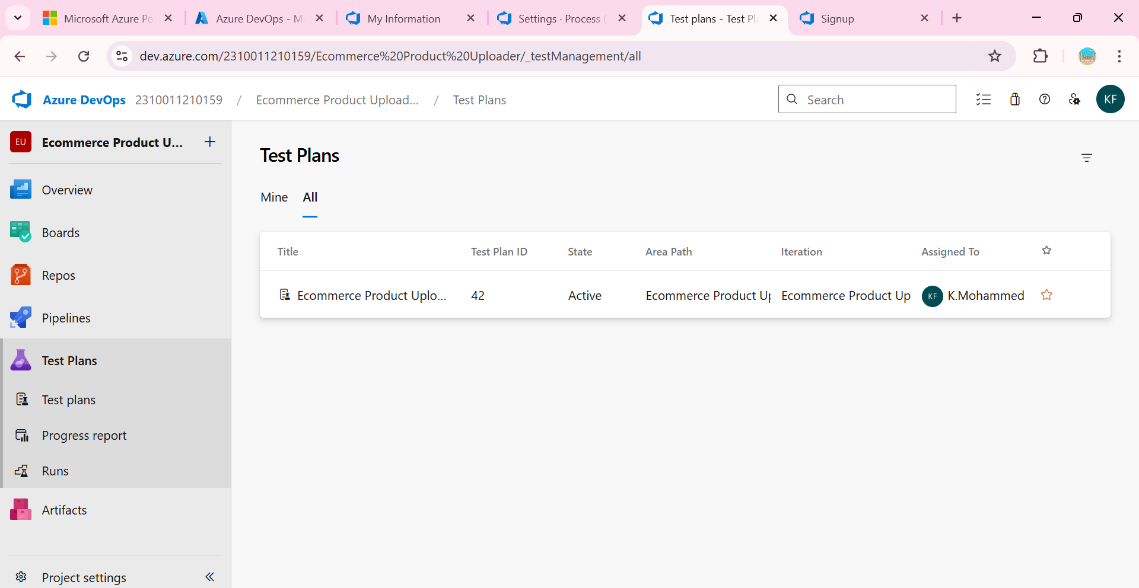
**Test Planning and Test Case Test Case Design Procedure**

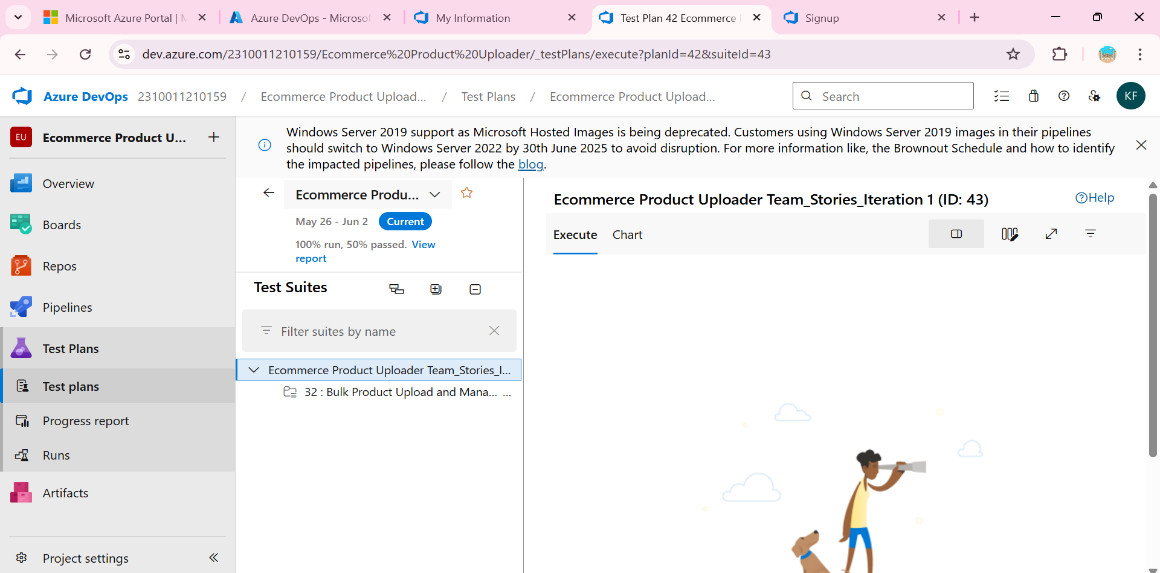
1. **Understand Core Features of the Application**
   * Bulk product upload via CSV/Excel files
   * Validate product fields: name, description, price, image
   * Prevent duplicate and unethical images
   * Create, read, update, delete (CRUD) products
   * Search products by name or description
   *  Display search results with pagination (10 per page)
2. **Define User Interactions**

Each test case simulates a real user behavior such as :

* + - * Logging in to the dashboard
      * Uploading a CSV of products
      * Editing an existing product
      * Searching for a product
      * Viewing paginated product lists

1. **Design Happy Path Test Cases**
   * Focused on validating that features work correctly under normal use.
   * Example: A valid CSV is uploaded and products appear in the list
2. **Design Error Path Test Cases**
   * Simulate failures or incorrect inputs to test error handling.
   * Example: Upload CSV missing required fields like price or image.
3. **Break Down Steps and Expected Results**
   * Each test case contains step-by-step actions and a corresponding expected outcome.
   * Ensures clarity for both testers and automation scripts.
   * Example: “Click Upload” → “File successfully validated.”  
     This makes testing repeatable and easy to automate later. Helps maintain clarity across manual and automated testing.
4. **Use Clear Naming and IDs**
   * Test cases follow naming conventions like:  
     TC01 – Upload Valid CSV, TC06 – Delete Non-existent Product  
     IDs help quickly track and link to related user stories.
5. **Separate Test Suites**
   * Grouped test cases based on functionality (e.g., Bulk Upload, Image Validation, Product CRUD,Search Functionality, Pagination Handling).
6. **Prioritize and Review**
   * Critical user actions are marked high-priority.
   * Reviewed for completeness and traceability against feature requirements.
7. **New test plan**

****

1. **Test suite**
2. **Test case**

Give two test cases for at least five user stories showcasing the happy path and error scenarios in azure DevOps platform.

Ecommerce Product Uploader– Test Plans

**USER STORIES**

* + As a store owner, I want to upload product information in bulk using a CSV or Excel file (ID: 201).
  + As a store owner, I want to ensure each product has a unique and valid image for proper display. (ID: 202).
  + As a store owner, I want to create, update, and delete product entries to manage inventory. (ID: 203).
  + As a store owner, I want to search products by name or description to find items quickly. (ID: 204).
  + As a store owner, I want to view products with pagination so I can navigate large lists easily. (ID: 205).

**Test Suites**

**Test Suit: TS01 - Bulk Product Upload and Management (ID: 44)**

**1. TC01 – Upload Valid CSV File**

* + **Action:**
    - Navigate to the Bulk Upload page from the dashboard.
    - Click **"Choose File"** and select a valid CSV with correct headers.
    - Click **"Submit"** to upload.
    - Go to the **Product List** page.
  + **Expected Results:**
    - Upload form is displayed.
    - Filename appears, file is accepted.
    - Success message: "Products uploaded successfully."
    - Products appear in the list.
  + **Type**: Happy Path

1. **TC02 – Upload CSV with Missing Fields**
   * **Action:**
     + Go to Bulk Upload page
     + Choose a CSV that lacks the "price" or "image" field
     + Click Submit
     + Check Product List page
   * **Expected Results:**
     + Upload form is displayed
     + Filename appears, but validation fails
     + Error message: “Missing required fields” is shown
     + No new products are added
   * **Type:** Error Path

**Test Suit: TS02 - Image Validation (ID: 302)**

1. **TC03 – Upload Unique Valid Image**
   * **Action:**
     + Attach image with each product.
     + All images are different and valid format.
   * **Expected Results:**
     + Upload successful with image preview enabled.
   * **Type:** Happy Path
2. **TC04 – Upload Duplicate Image**
   * **Action:**
     + Tap Assign same image to two different products.
   * **Expected Results:**
     + Error: “Same image cannot be used for different products.”
   * **Type:** Error Path

**Test Suit: TS03 - Product CRUD (ID: 303)**

1. **TC05 – Edit Product Name**
   * **Action:**
     + Go to product list.
     + Click Edit, change name, and Save.
   * **Expected Results:**
     + Product name updates and reflects immediately.
   * **Type:** Happy Path
2. **TC06 – Delete Non-existent Product**
   * **Action:**
     + Try to delete a product ID not in the list.
   * **Expected Results:**
     + Message: “Product not found or already deleted.”
   * **Type:** Error Path

**Test Suit: TS04 - Product Search (ID: 304)**

1. **TC07 – Search Product by Name**
   * **Action:**
     + Go to Search bar.
     + Type product name (e.g., “Shoes”) and search.
   * **Expected Results:**
     + Matching products appear in the result list.
   * **Type:** Happy Path
2. **TC08 – Invalid Product Search**
   * **Action:**
     + Search for a non-existing name like “$$$%%%”
   * **Expected Results:**
     + Fallback message: “No products found for the search term.”
   * **Type:** Error Path

**Test Suit: TS05 - Pagination Handling (ID: 305)**

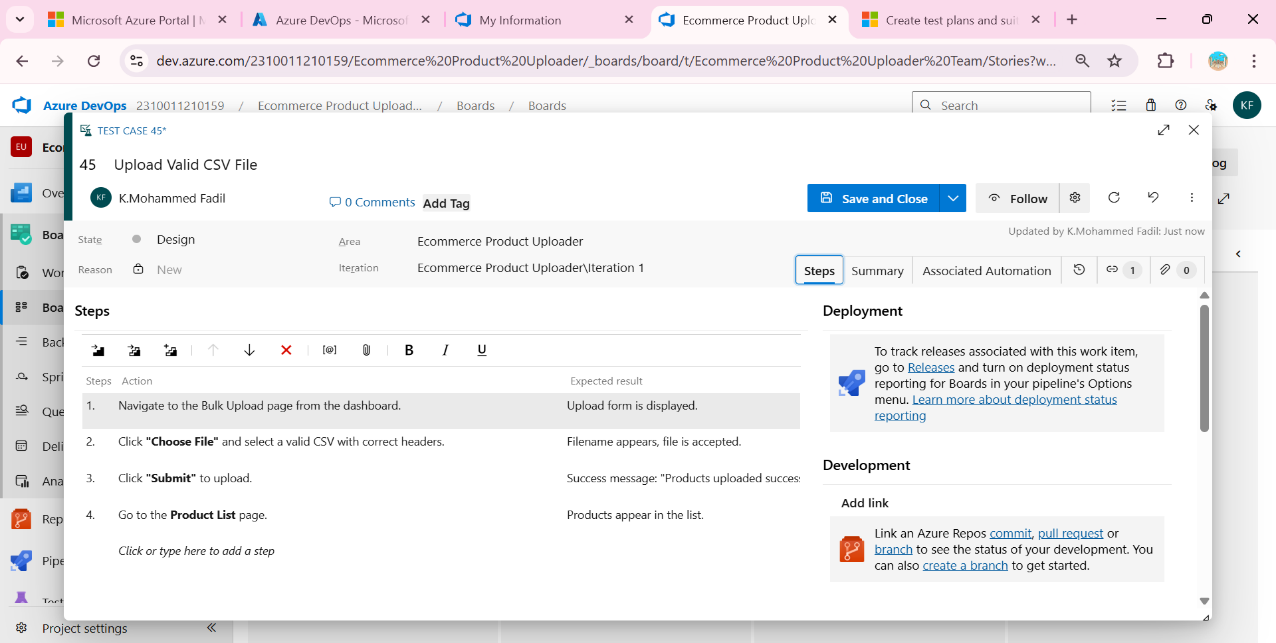
1. **TC09 – Navigate Between Pages**
   * **Action:**
     + Search products that return over 10 results.
     + Click “Next” to view second page..
   * **Expected Results:**
     + Next 10 products load on the next page.
   * **Type:** Happy Path
2. **TC10 – Empty Last Page**
   * **Action:**
     + Navigate to a page with no results (e.g., page 5).
   * **Expected Results:**
     + Error: “No more products to display.”
   * **Type:** Error Path

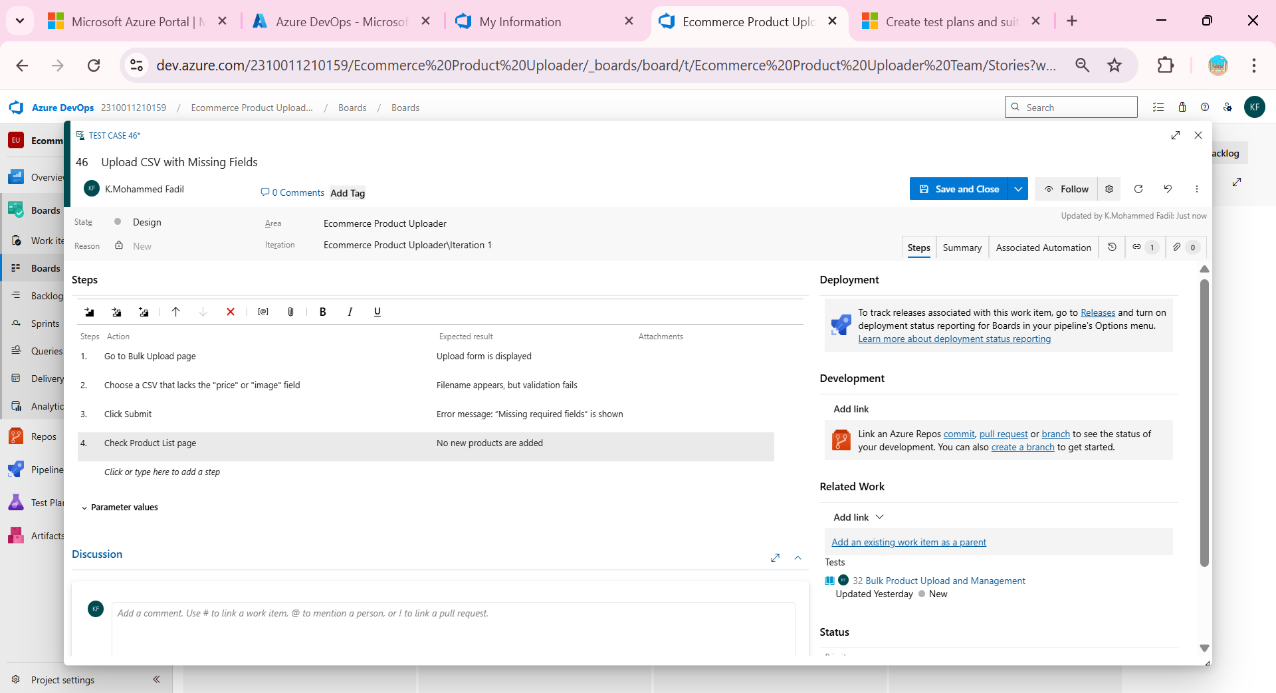
**Test Suit: TS06 - Product Image Rules Enforcement (ID: 306)**

1. **TC11 – Ethical Image Upload**
   * **Action:**
     + Select and upload clean, non-offensive product images.
     + Images are in .jpg or .png format.
   * **Expected Results:**
     + Images are accepted and linked to products.
   * **Type:** Happy Path
2. **TC12 – Inappropriate Image Upload**
   * **Action:**
     + Upload an image containing unethical or explicit content.
   * **Expected Results:**
     + Error: “Inappropriate image detected. Upload blocked.”

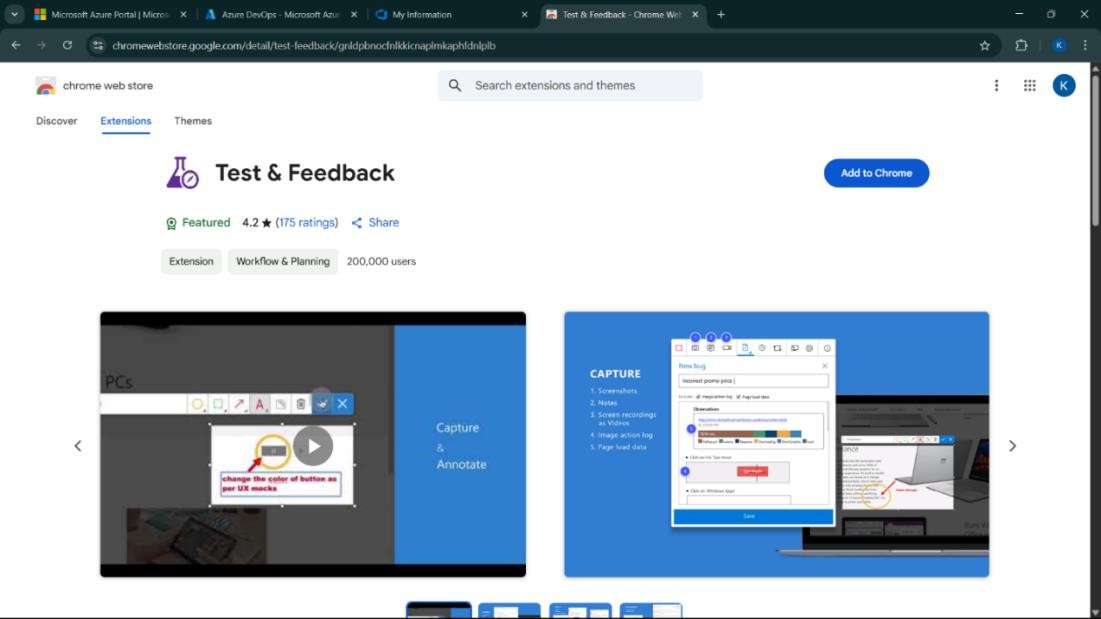
* **Type:** Error Path

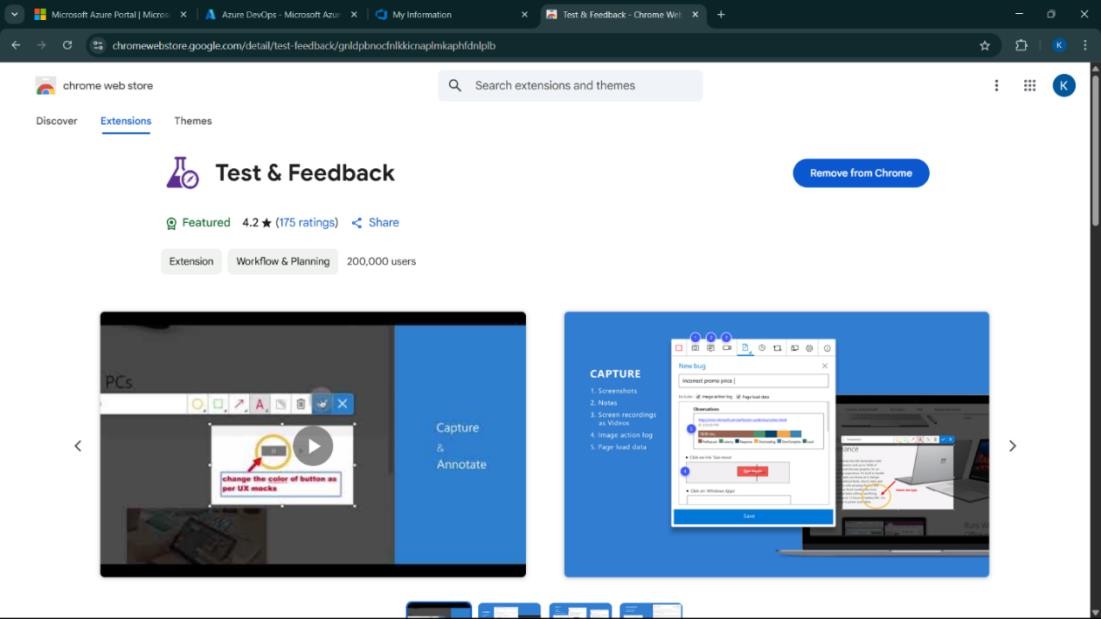
**Test Cases**

****

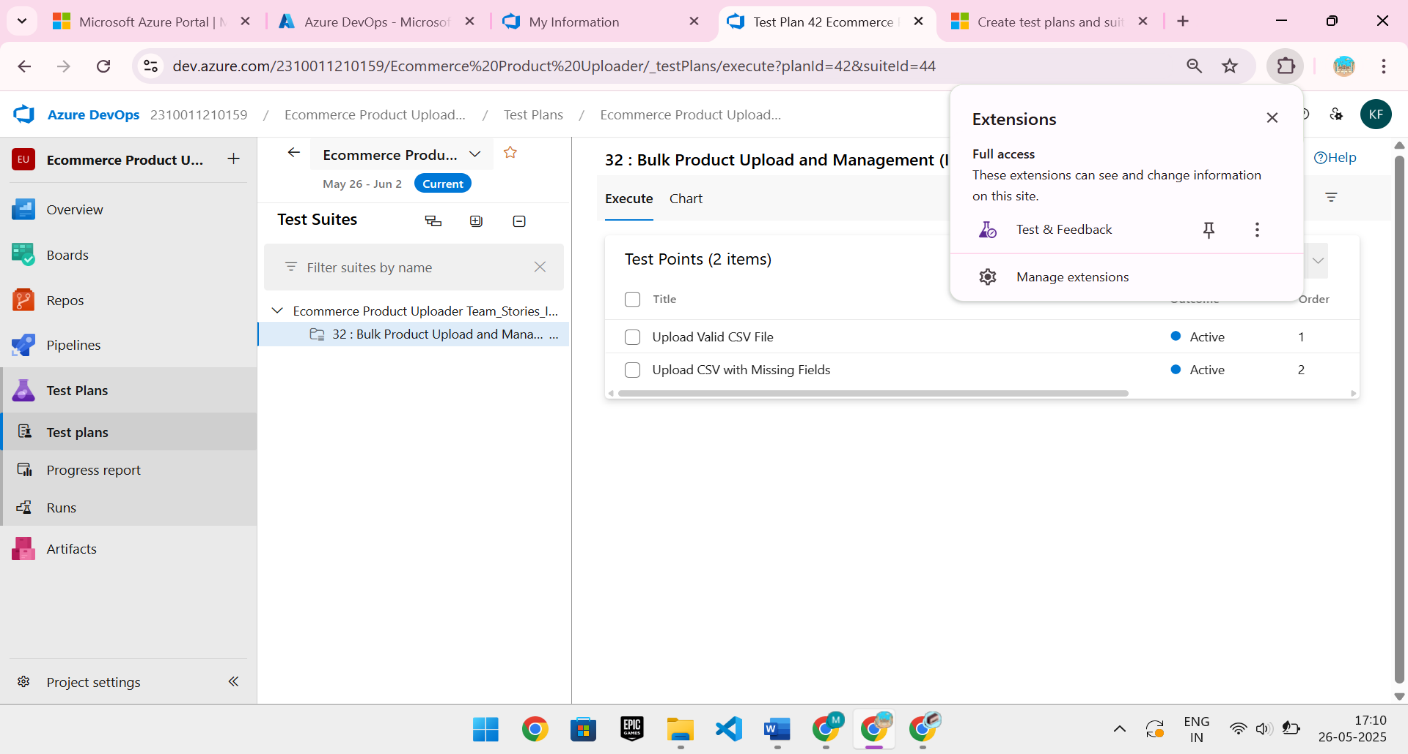


1. **Installation of test**

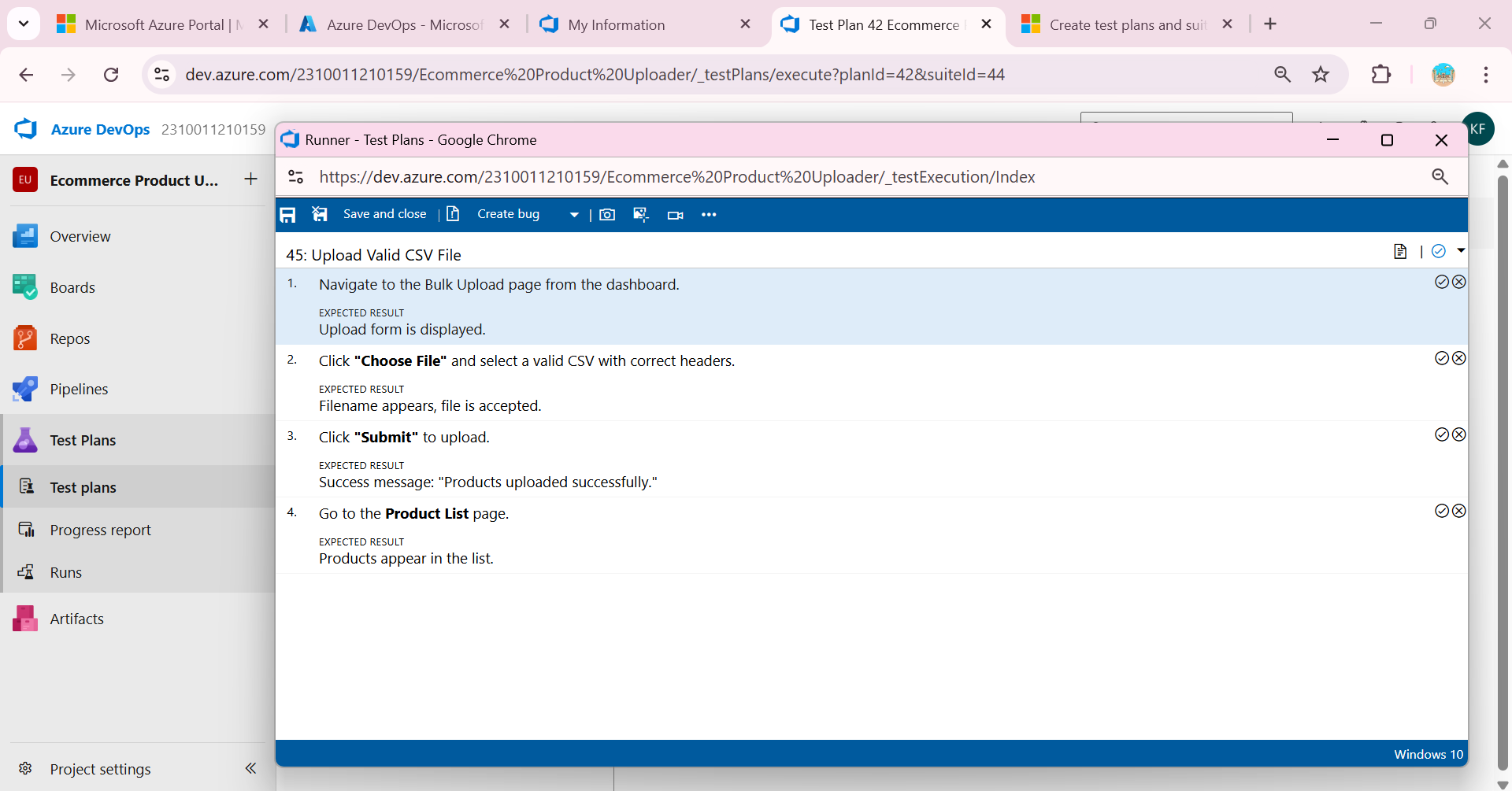
****



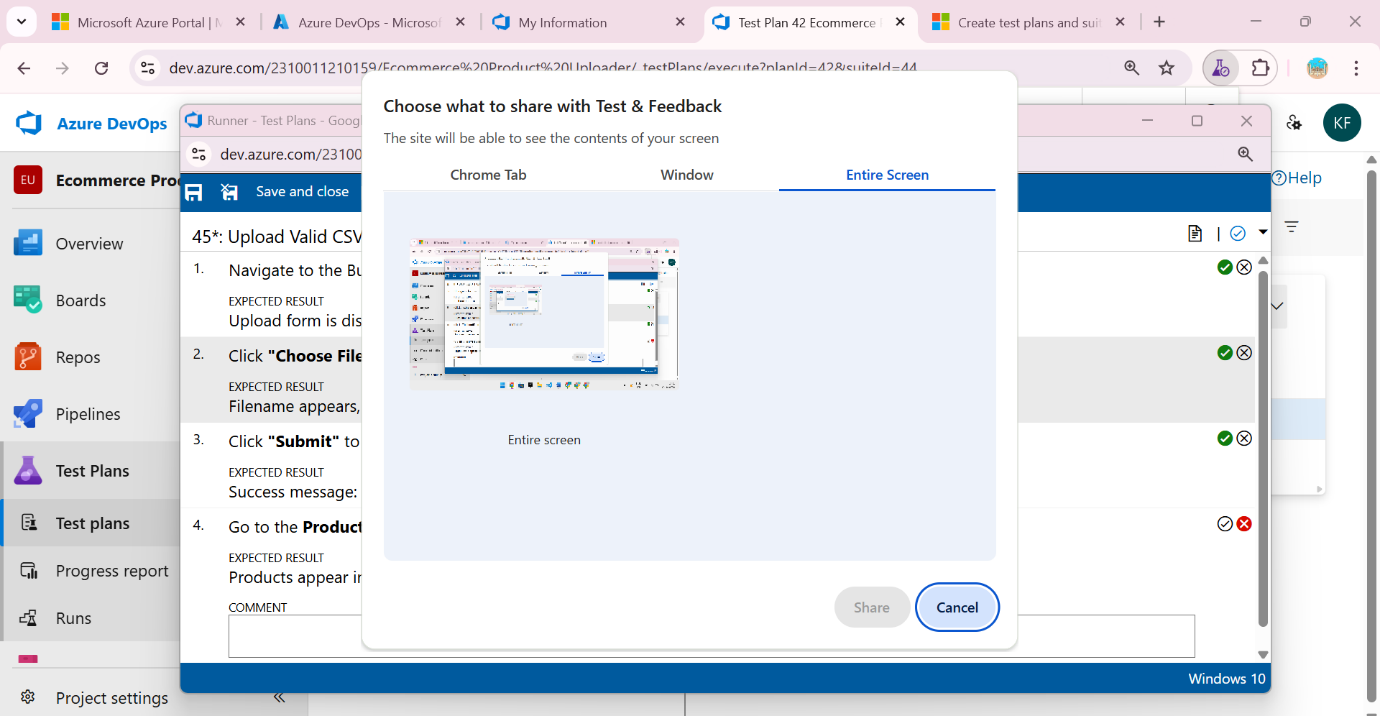
Test and feedback Showing it as an extension



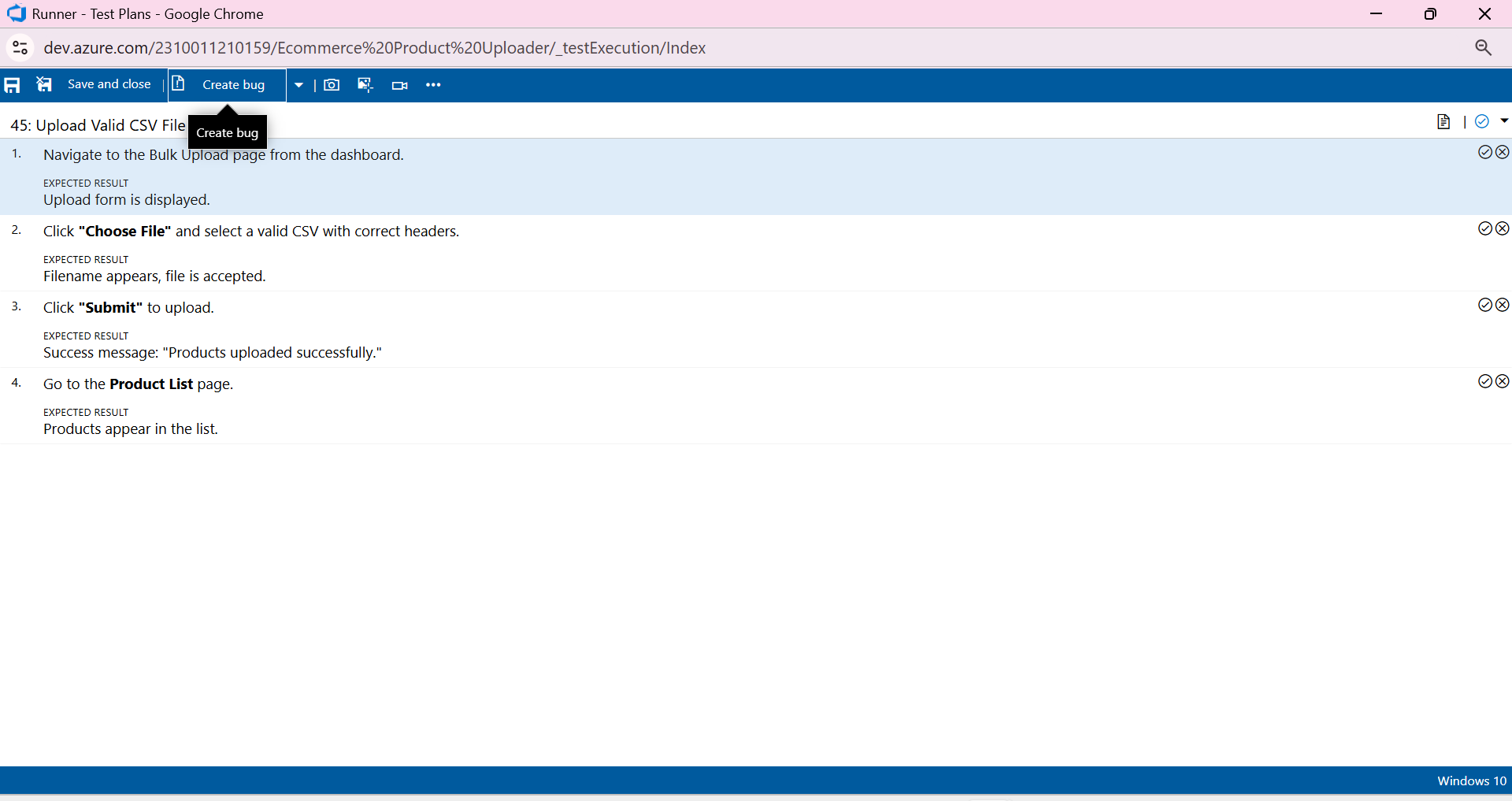
1. **Running the test cases**

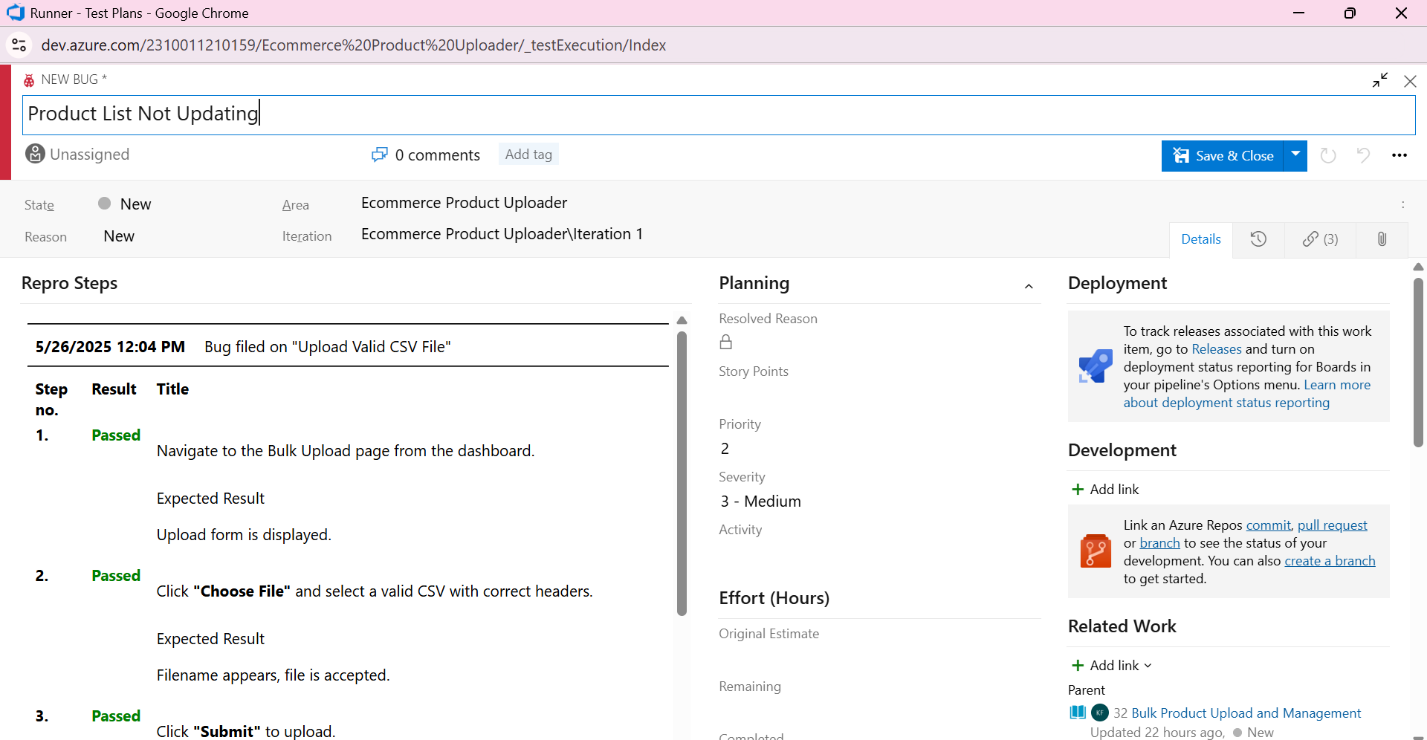
****

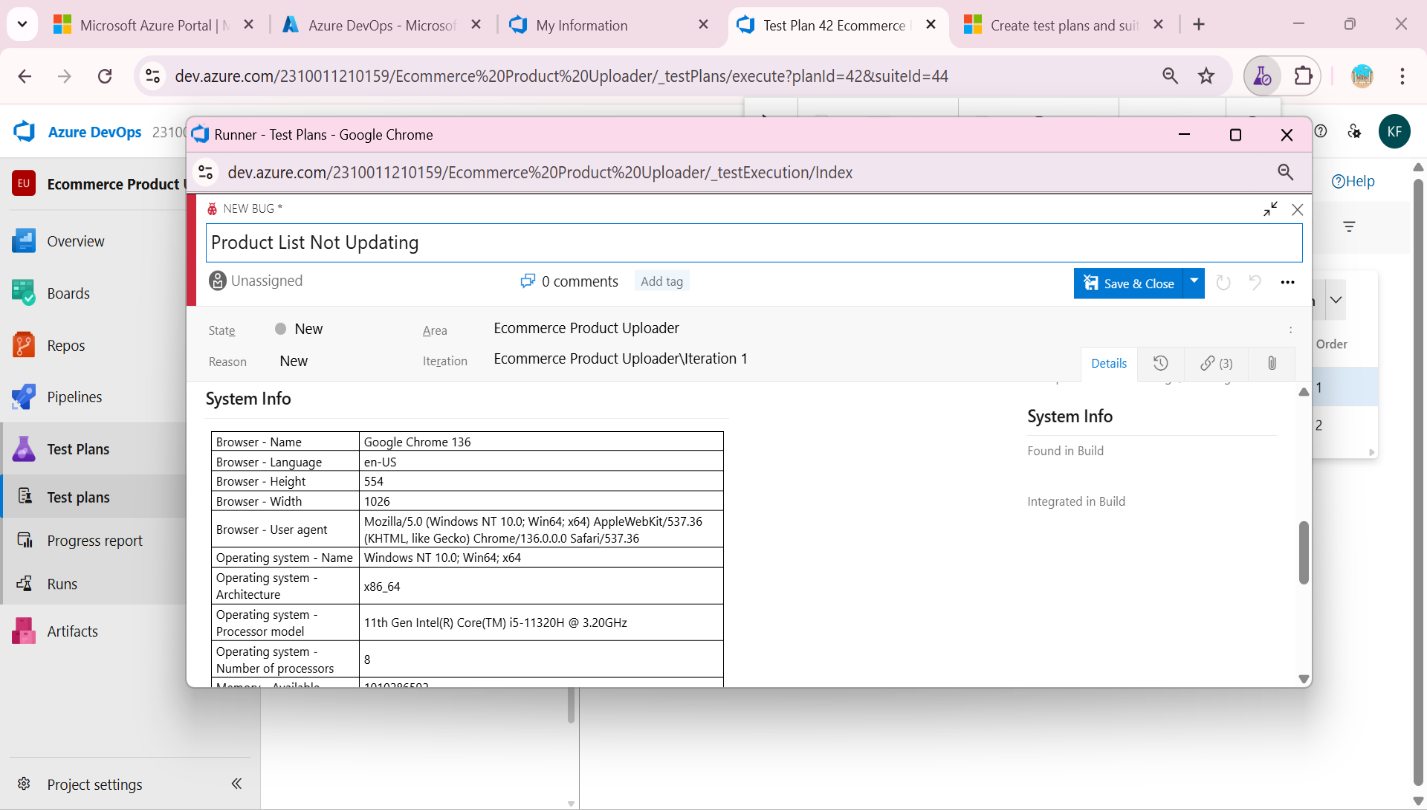
1. **Recording the test case**

****

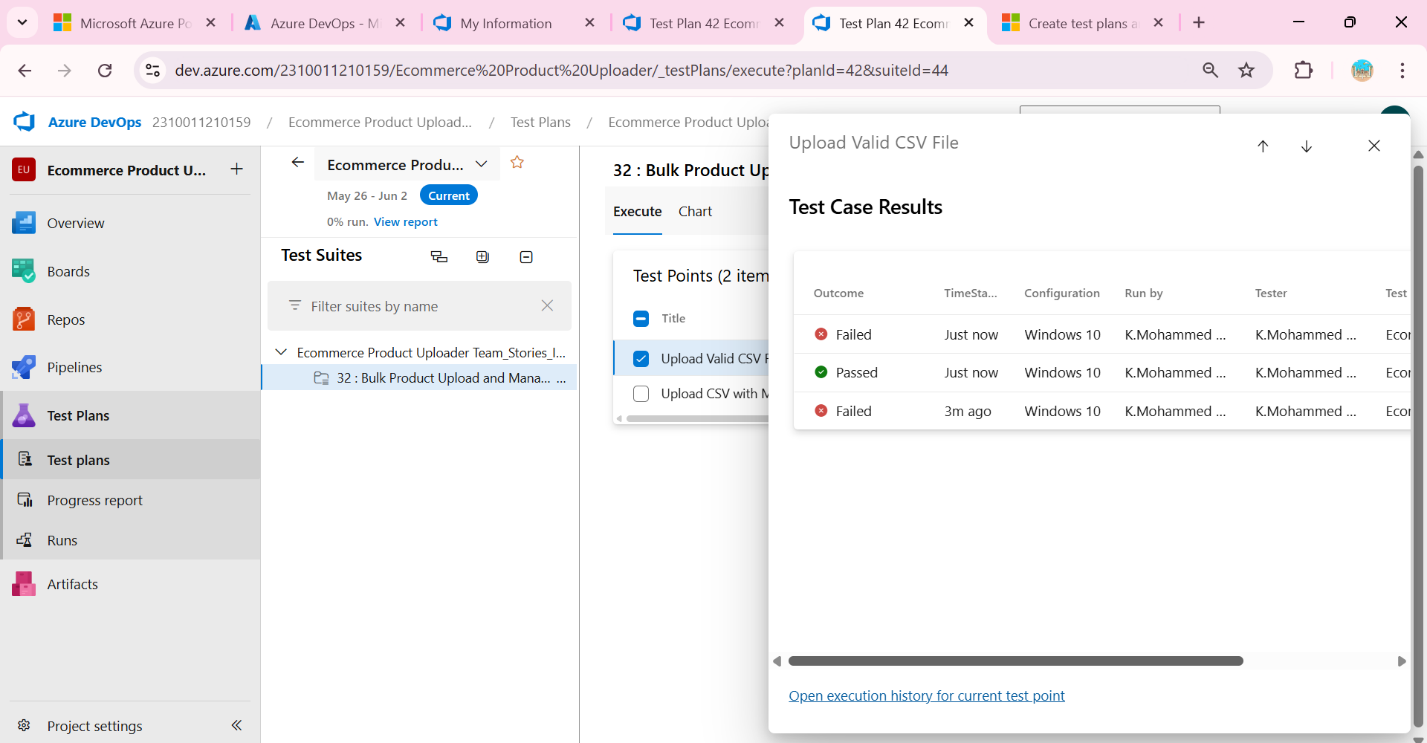
1. **Creating the bug**

****

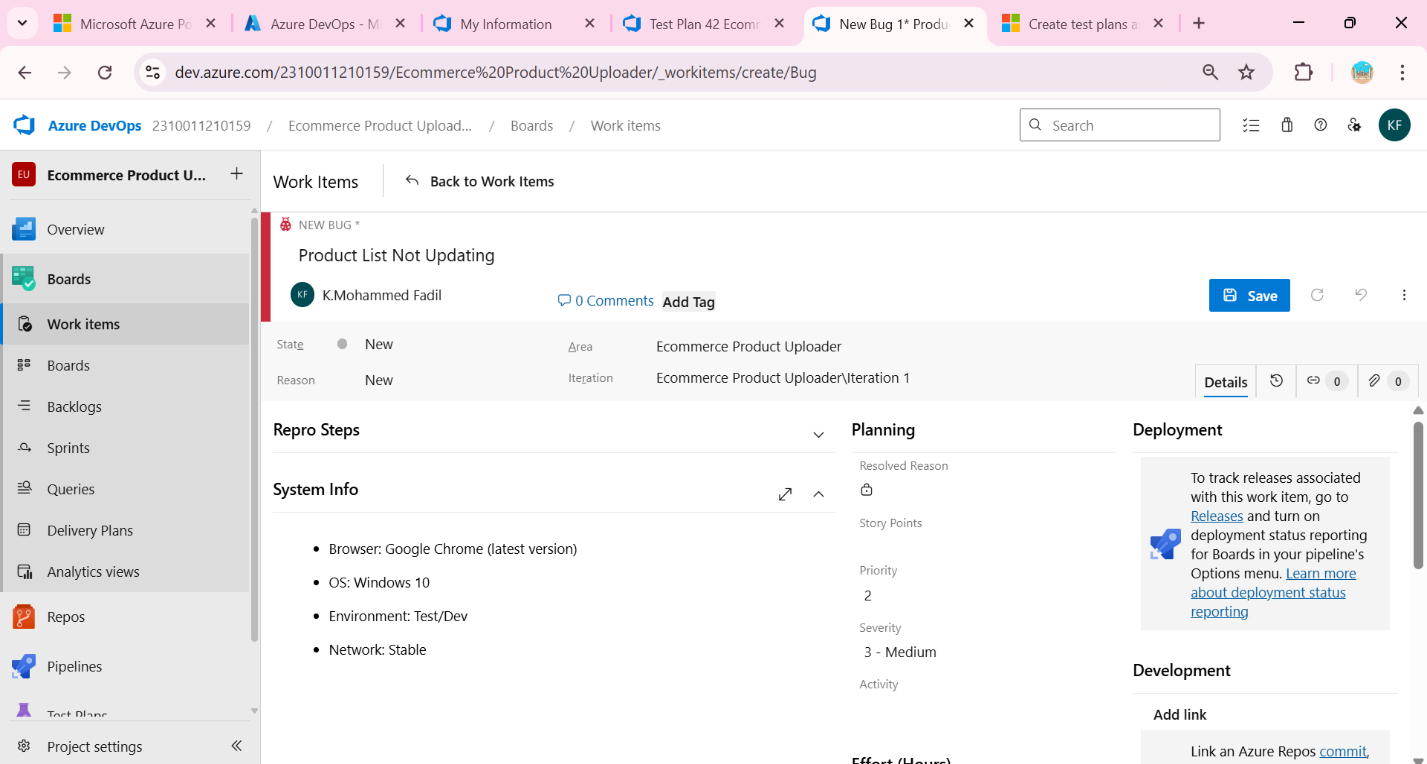


****

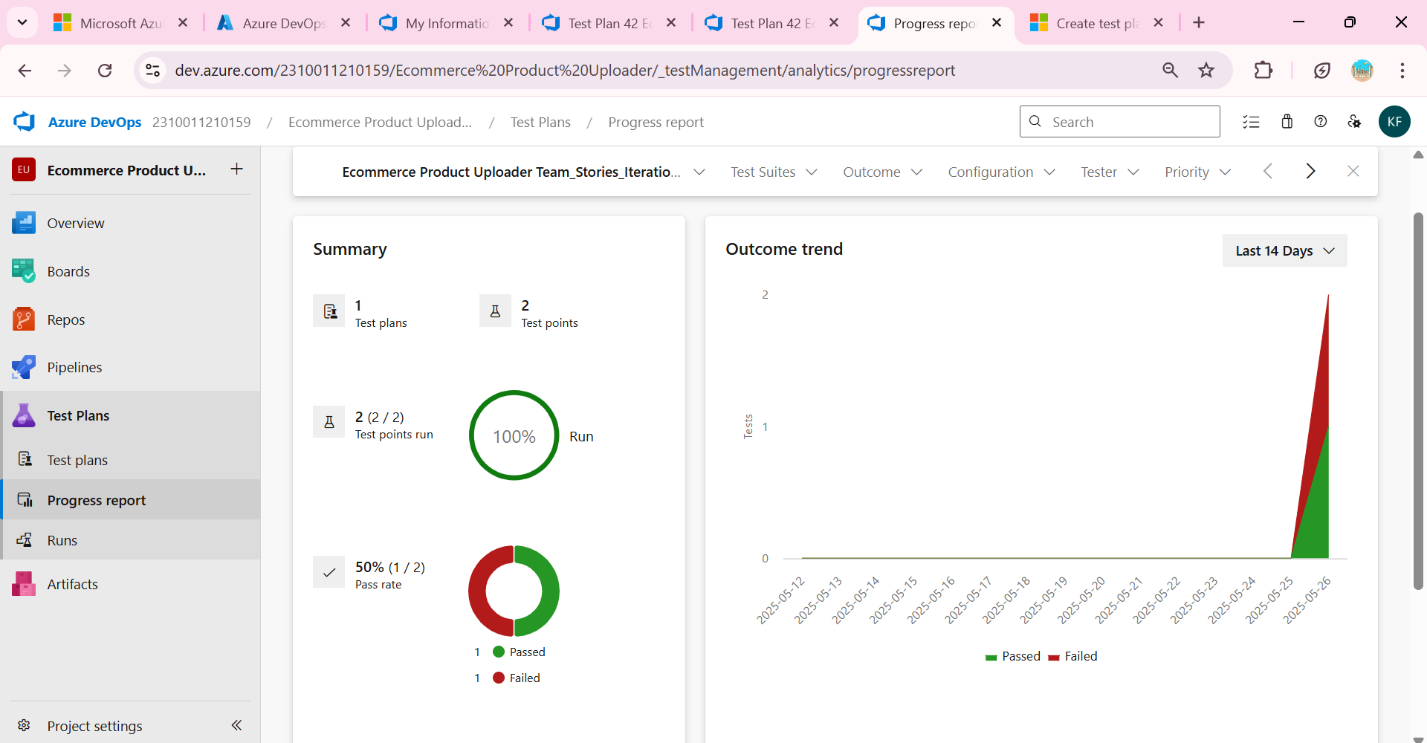
1. **Test case results**

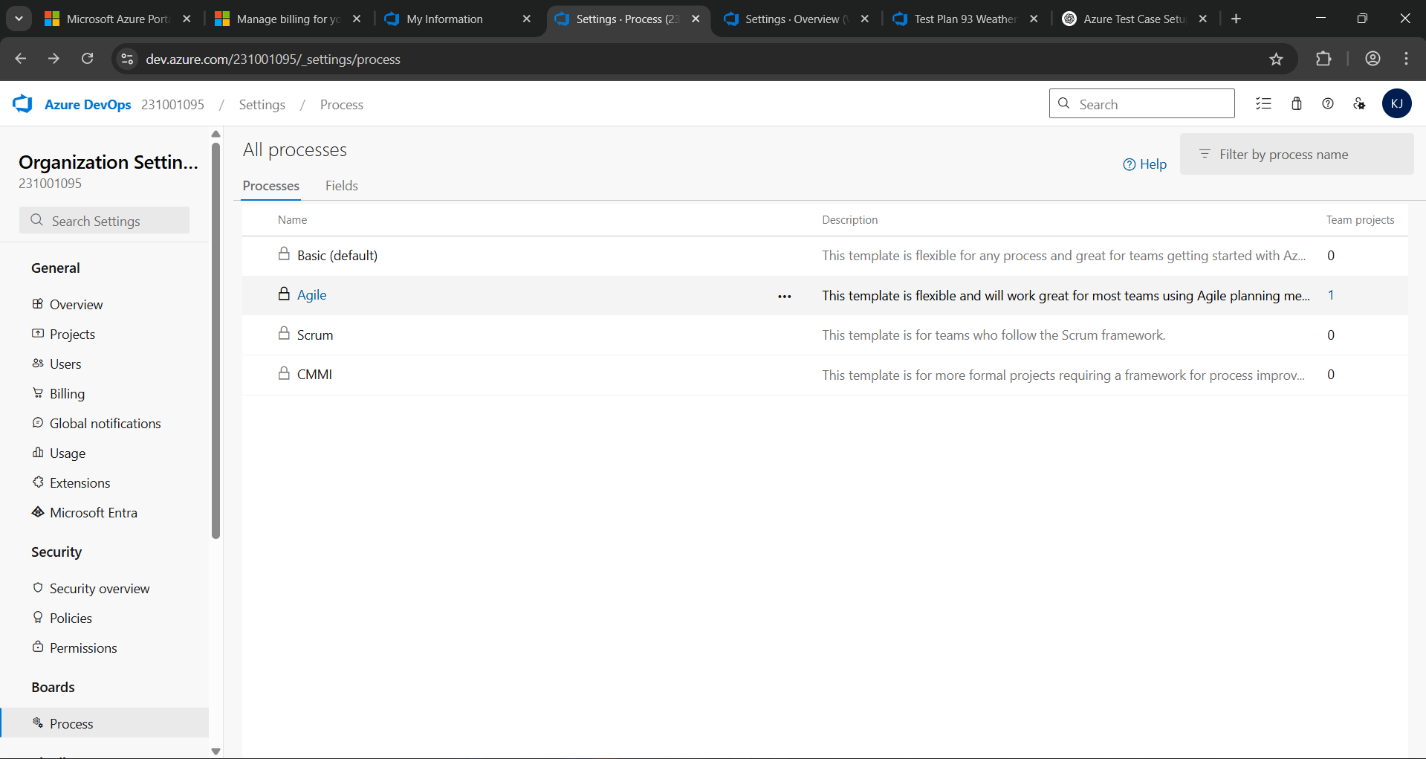
****

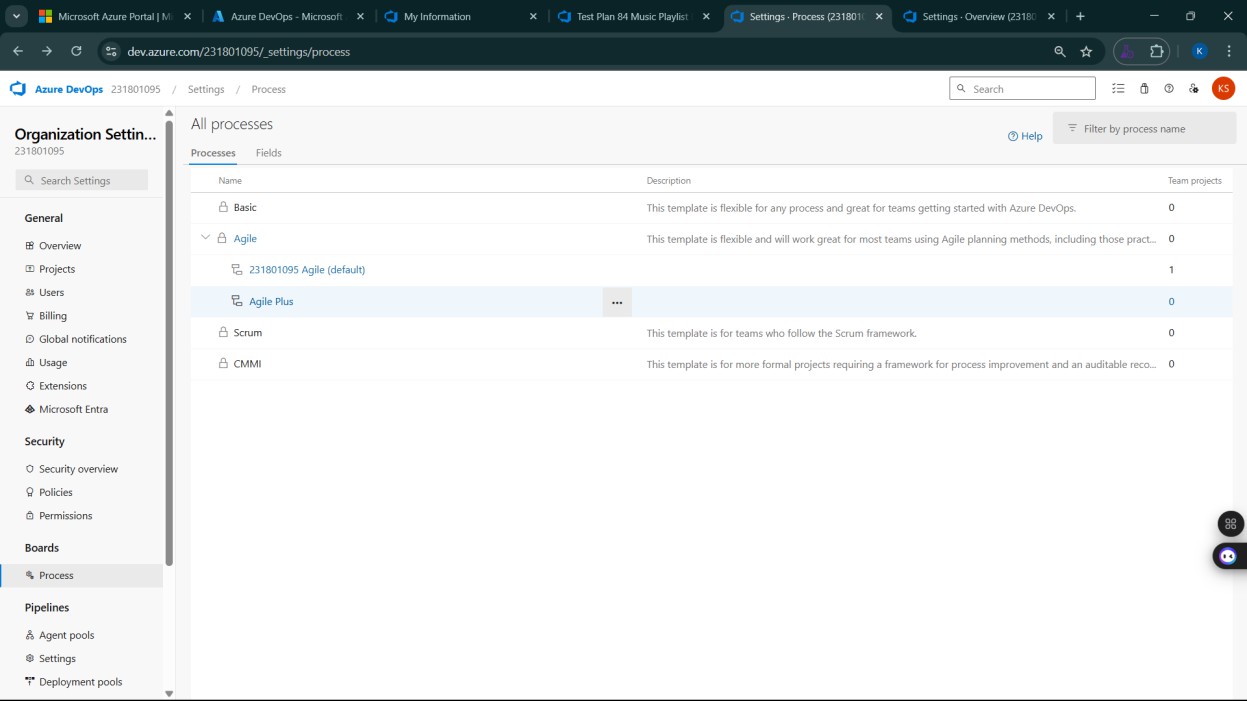
1. **Test report summary**

****

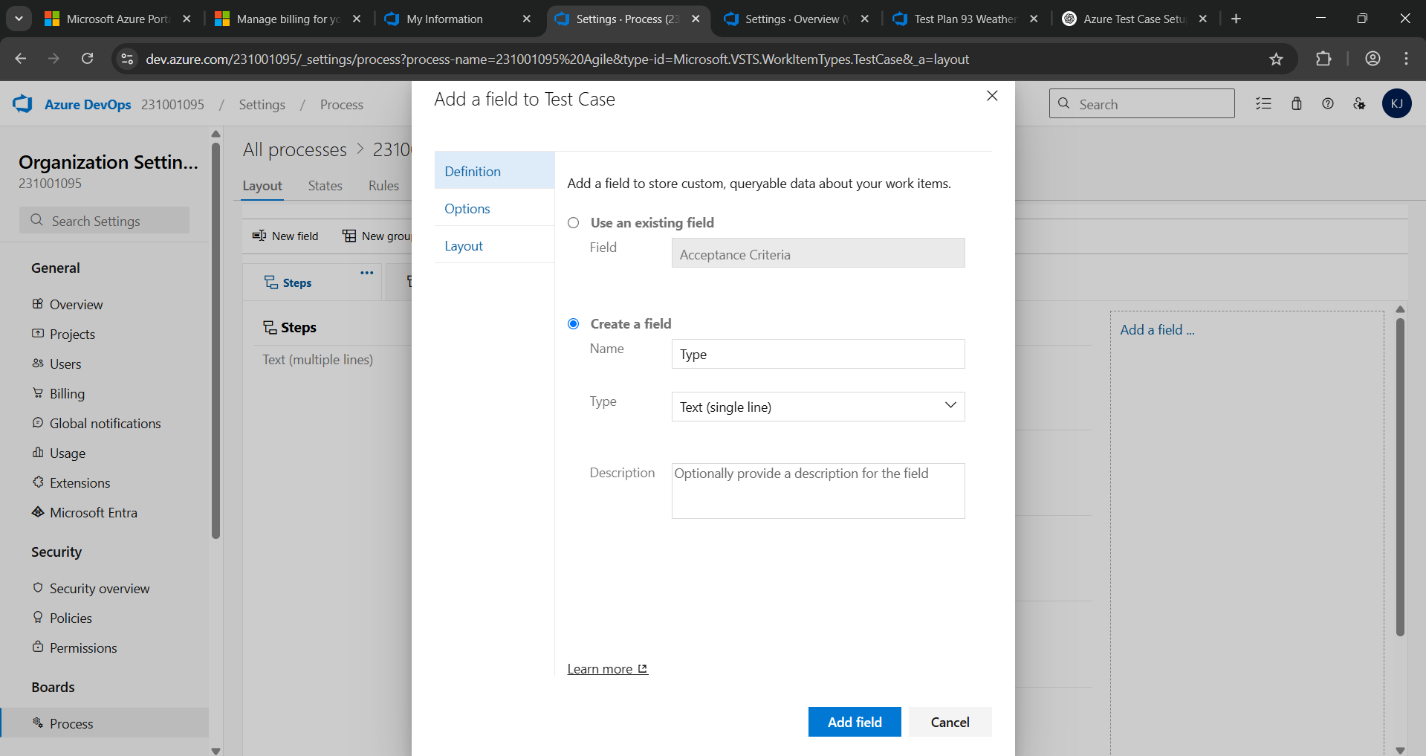
* + Assigning bug to the developer and changing state

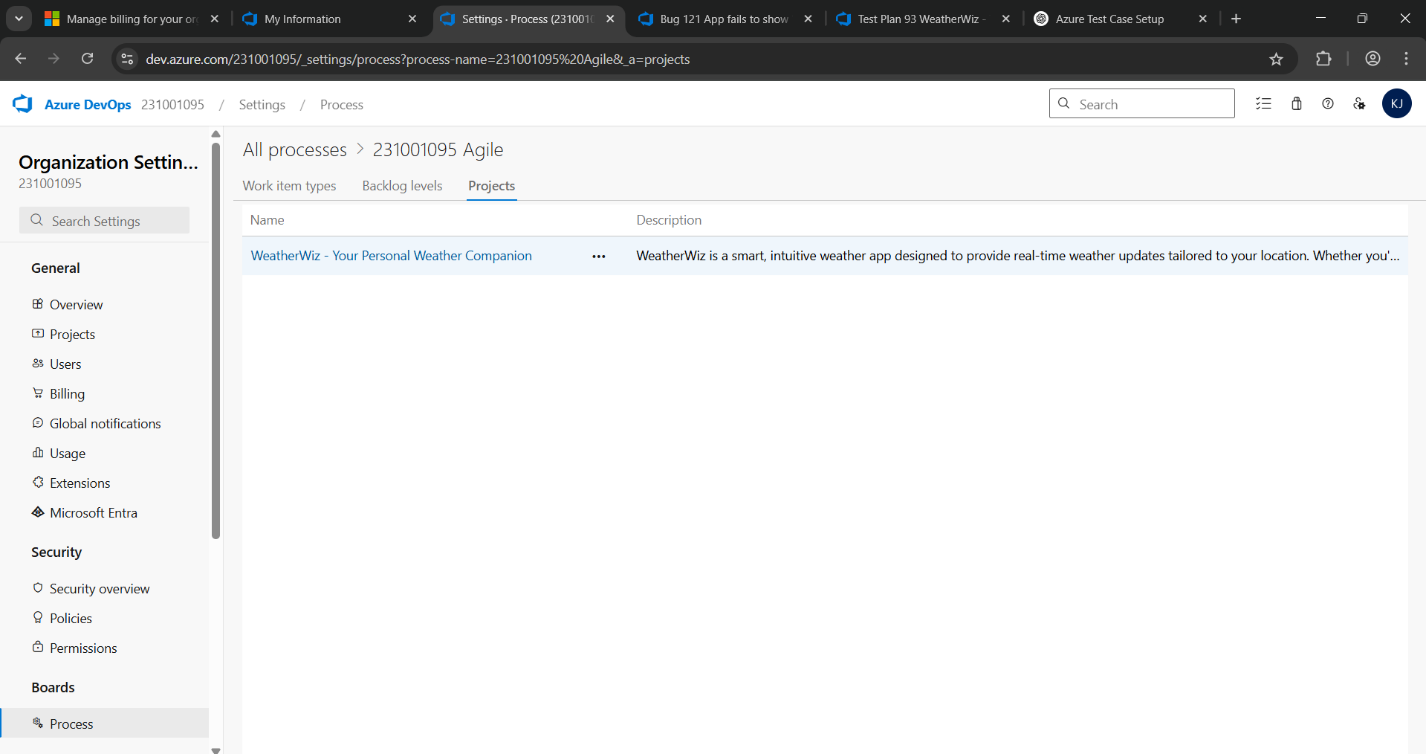
1. **Progress report**
2. **Changing the test template**

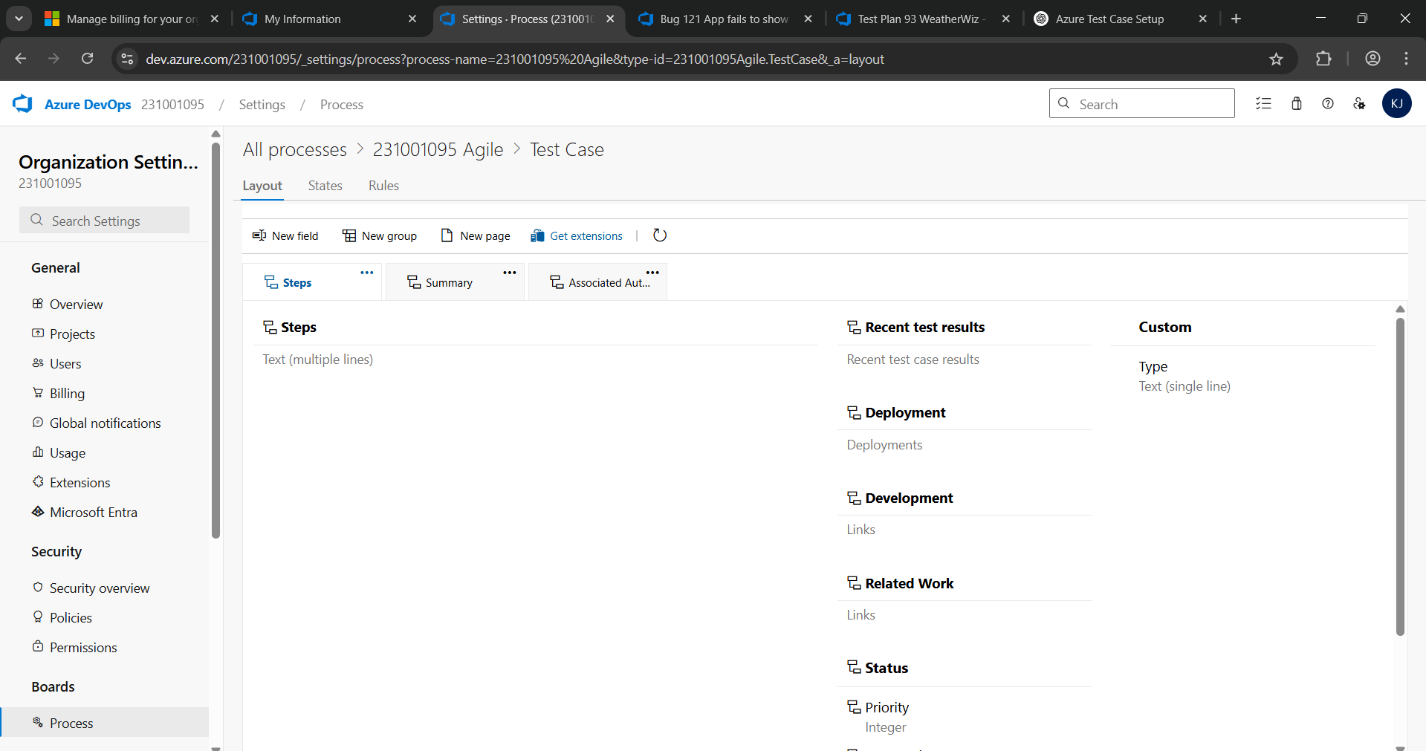
****

****

1. **View the new test case template**

****

****



**Result:**

The test plans and test cases for the user stories is created in Azure DevOps with Happy Path and

Error Path

**LOAD TESTING AND PERFORMANCE TESTING**

**EXP NO: 9**

**Aim:**

Create Epic, Features, User Stories, Task

To create an Azure Load Testing resource and run a load test to evaluate the performance of a target endpoint.

**Load Testing**

**Steps to Create an Azure Load Testing Resource:**

Before you run your first test, you need to create the Azure Load Testing resource:

1. Sign in to Azure Portal

Go to [https://portal.azure.com](https://portal.azure.com/) and log in.

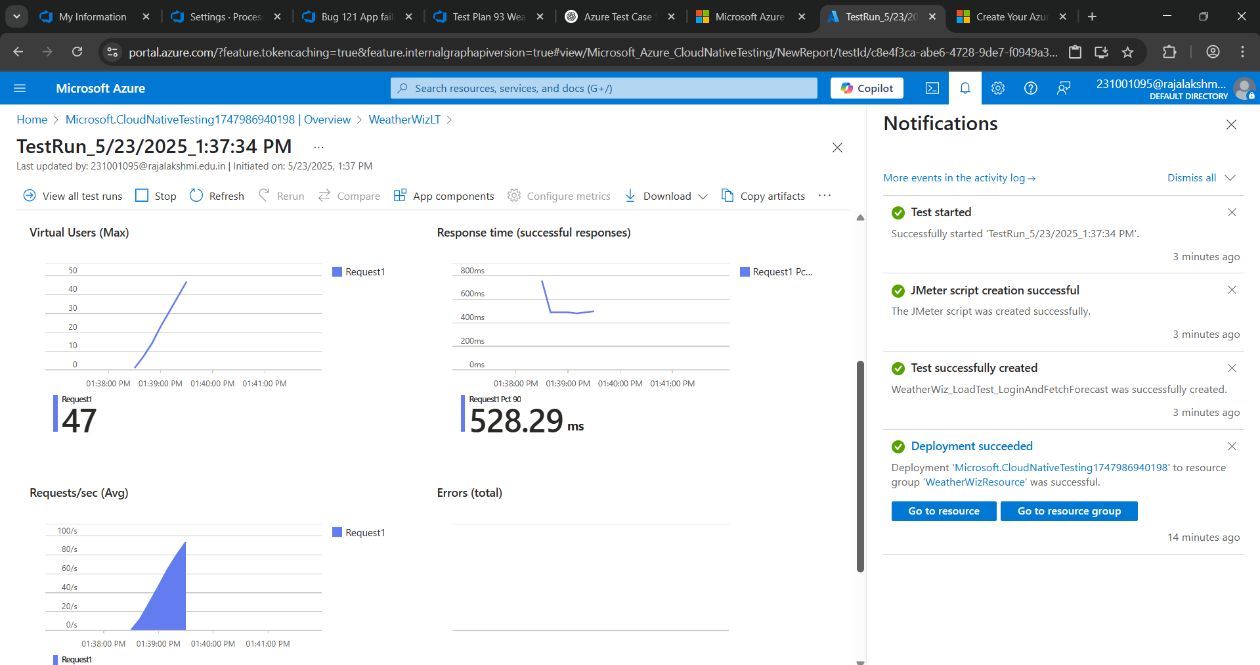
1. Create the Resource
   * Go to *Create a resource* → Search for “Azure Load Testing”.
   * Select Azure Load Testing and click Create.
2. Fill in the Configuration Details
   * Subscriptio*n:* Choose your Azure subscription.
   * *Resource Group:* Create new or select an existing one.
   * *Name:* Provide a unique name (no special characters).
   * *Location:* Choose the region for hosting the resource.
3. (Optional) Configure tags for categorization and billing.
4. Click Review + Create, then Create.
5. Once deployment is complete, click Go to resource.

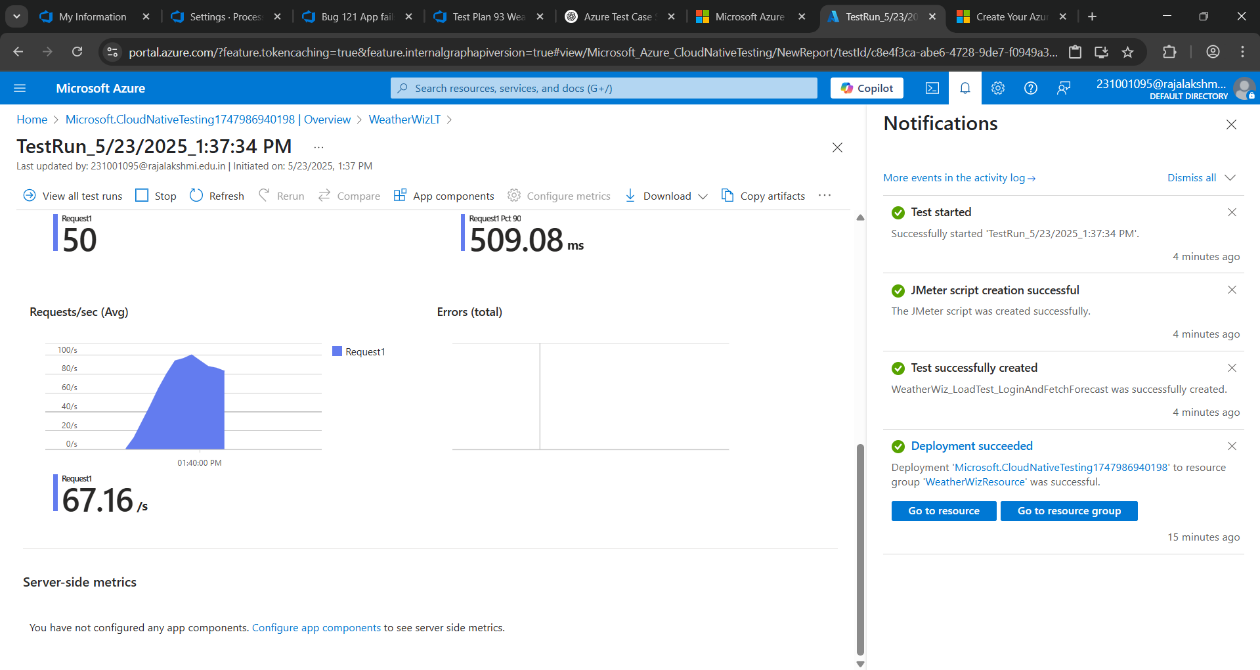
**Steps to Create and Run a Load Test:**

Once your resource is ready:

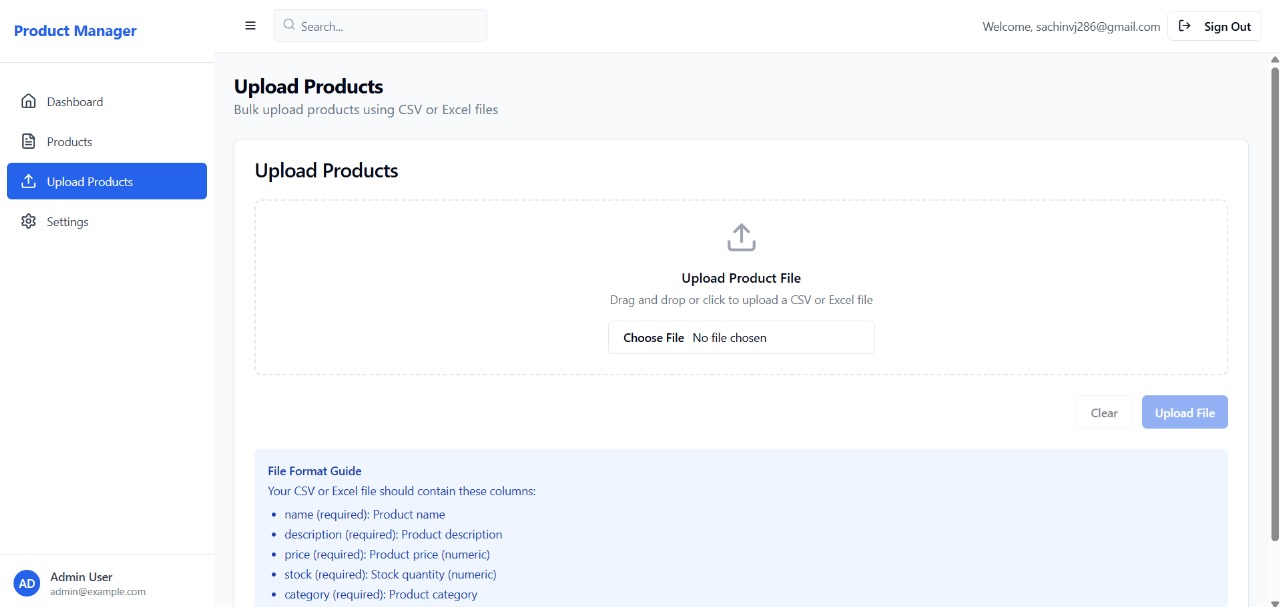
1. Go to your Azure Load Testing resource and click Add HTTP requests > Create.
2. Basics Tab
   * *Test Name:* Provide a unique name.
   * *Description:* (Optional) Add test purpose.
   * *Run After Creation:* Keep checked.
3. Load Settings
   * *Test URL:* Enter the target endpoint (e.g., https://yourapi.com/products).
4. Click Review + Create → Create to start the test.

**Load Testing**

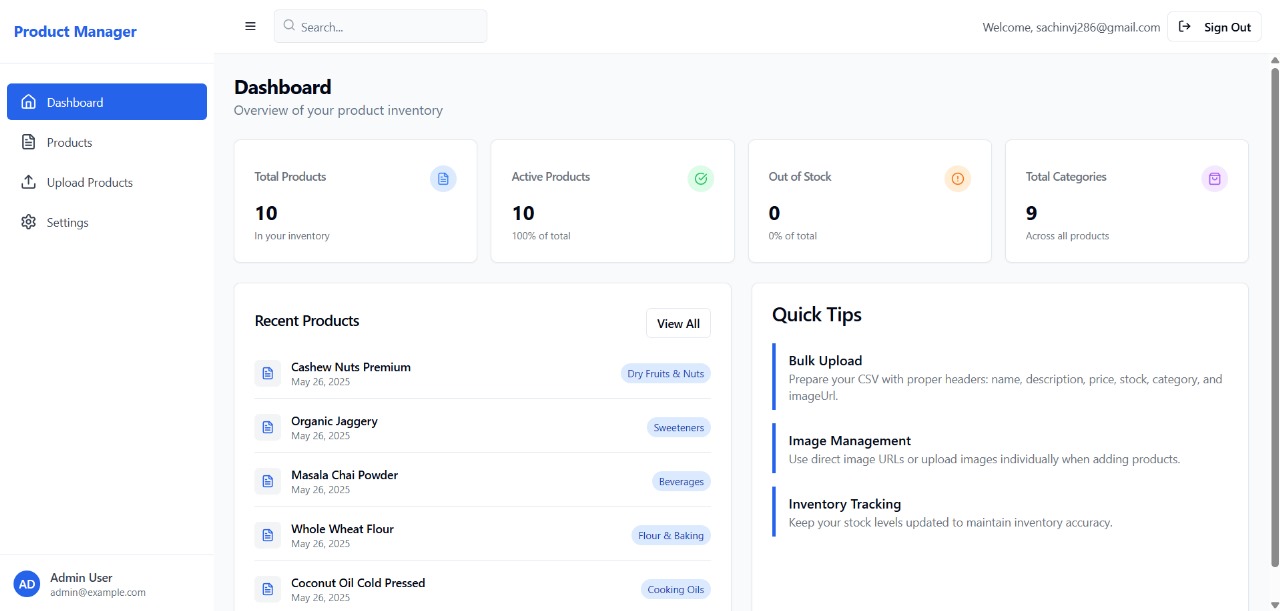
****

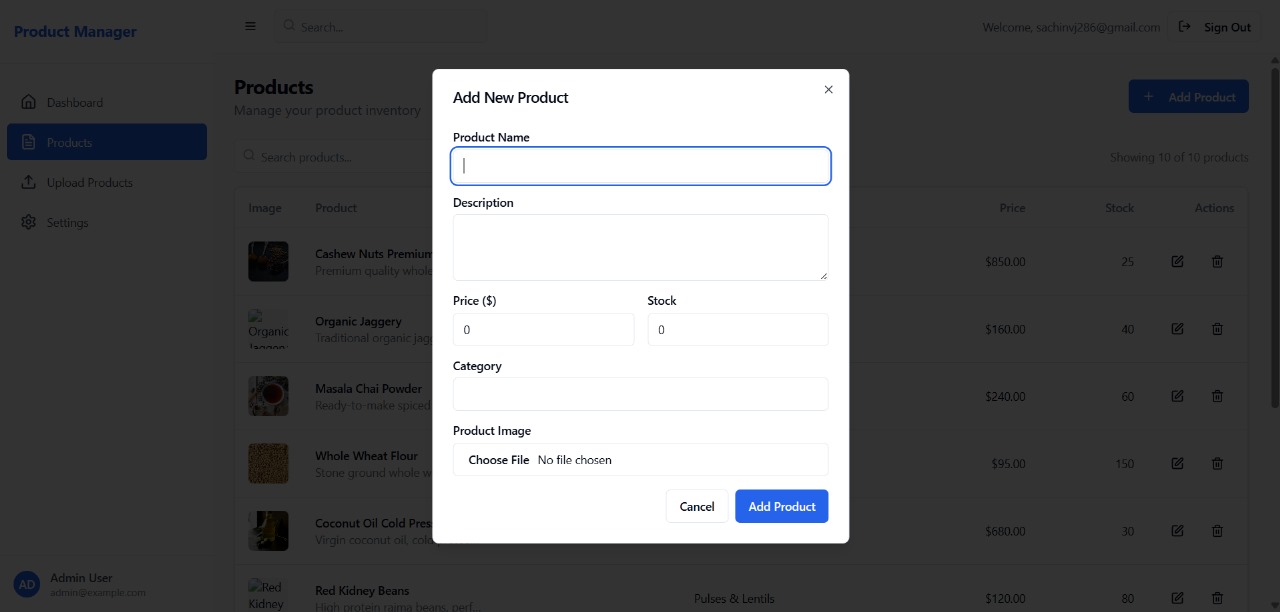
****

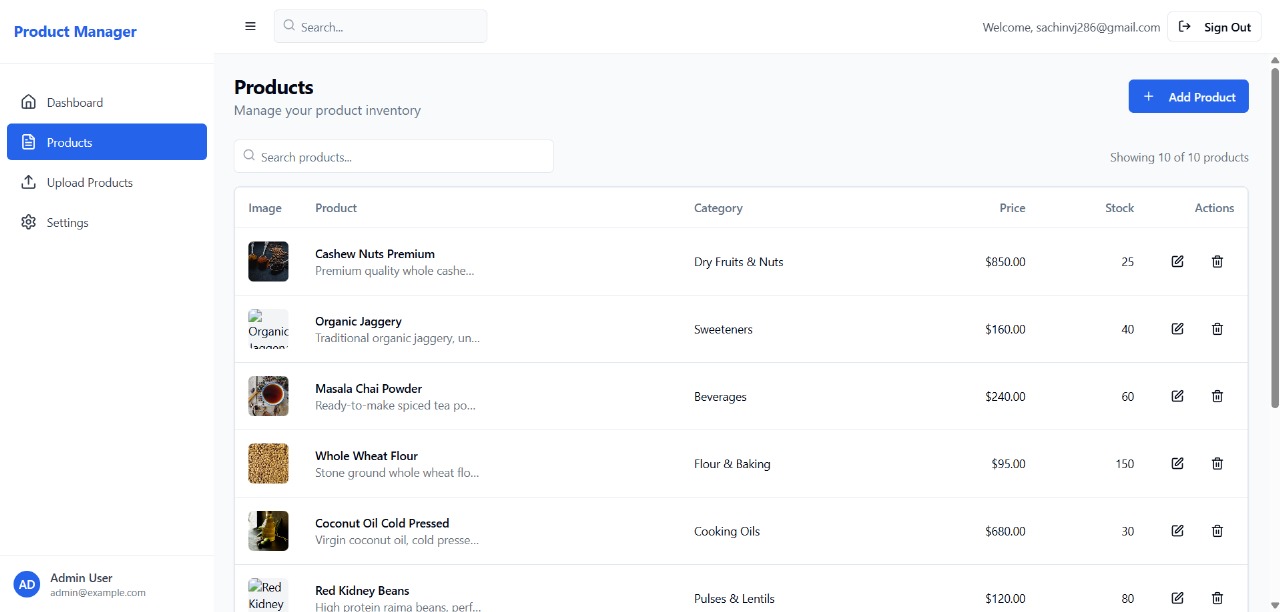
**1.Upload product file**



**2.Dashboard**

****

** 3.Add new Product**

** 4.Product Details**

**Result:**

Successfully created the Azure Load Testing resource and executed a load test to assess the

performance of the specified endpoint.

**GITHUB: PROJECT STRUCTURE & NAMING CONVENTIONS**

**EXP NO: 10**

**Aim:**

Create Epic, Features, User Stories, Task

To provide a clear and organized view of the project's folder structure and file naming conventions, helping contributors and users easily understand, navigate, and extend the Ecommerce Product Uploader project.

**GitHub Project Structure**

****

**Result:**

The GitHub repository clearly displays the organized project structure and consistent naming

conventions, making it easy for users and contributors to understand and navigate the codebase.