

# INDEX

S.No.	Date	Title
1.	23/1/25	Azure Devops Environment Setup.
2.	30/1/25	Azure Devops Project Setup and User Story Management.
3.	06/2/25	Setting Up Epics, Features, And User Stories for Project Planning.
4.	13/2/25	Sprint Planning.
5.	20/2/25	Poker Estimation.
6.	27/2/25	Designing Class Diagram and Sequence Diagram.
7.	06/3/25	Designing Use Case Diagram and Activity Diagram.
8.	20/3/25	Testing – Test Plans and Test Cases.
9.	27/3/25	Load Testing and Pipelines.
10.	03/4/25	GitHub: Project Structure & Naming Conventions.

**EXP NO: 1**

## AZURE DEVOPS ENVIRONMENT SETUP

**Date :**

### **Aim:**

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-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>

The screenshot shows the Microsoft Azure portal homepage. At the top, there's a navigation bar with links for Microsoft, Azure, Explore, Products, Solutions, Pricing, Partners, Resources, Learn, Support, Contact Sales, Get started with Azure, Sign in, and a search bar. Below the navigation is a dark header with the text "Microsoft Azure portal" and a subtext: "Build, manage, and monitor everything from simple web apps to complex cloud applications in a single, unified console". There are two buttons: "Sign in" and "New to Azure? Get started >". A blue banner at the bottom of the page says "Check out the how-to video series for tips on deploying your cloud workloads from the Azure portal. >". On the left side, there's a section for the "Azure mobile app" with a subtext: "Stay connected to your Azure resources—anytime, anywhere. Now available for iOS and Android." To the right of this text are two screenshots of the Azure mobile app interface on a smartphone.

## 2. Azure home page

Welcome to Azure!

Don't have a subscription? Check out the following options.

**Start with an Azure free trial**  
Get \$200 free credit toward Azure products and services, plus 12 months of popular [free services](#).

**View** **Start**

**Manage Microsoft Entra ID**  
Manage access, set smart policies, and enhance security with Microsoft Entra ID.

**Learn more**

**Azure for Students**  
Get free software, Azure credit, or access Azure Dev Tools for Teaching after you verify your academic status.

**Start**

**Azure services**

[Create a resource](#) [Azure DevOps organizations](#) [Storage accounts](#) [Quickstart Center](#) [Azure AI foundry](#) [Kubernetes services](#) [Virtual machines](#) [App Services](#) [SQL databases](#) [More services](#)

**Resources**

Recent   Favorite

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

Welcome to Azure!

Don't have a subscription?

**Start with an Azure free trial**  
Get \$200 free credit toward Azure products and services, plus 12 months of popular [free services](#).

**Start**

**Azure services**

[Create a resource](#) [Azure DevOps organizations](#) [Storage accounts](#) [Quickstart Center](#) [Azure AI foundry](#) [Kubernetes services](#) [Virtual machines](#) [App Services](#) [SQL databases](#) [More services](#)

**Resources**

Recent   Favorite

Search bar: **azure devops**

Search results:

- All
- Services (99+)
- Resources

Services

- Azure DevOps organizations
- Azure Cosmos DB
- Azure Database for MySQL servers
- Azure Deployment Environments

Marketplace

- Build Agents for Azure DevOps
- Azure DevOps Auditing
- Azure DevOps Backup Tool
- Self Hosted Runner for Azure DevOps

Documentation

- Java JDKs Support for Azure Development - Java on Azure
- Frequently asked questions about the Azure Education Hub
- Describe the core architectural components of Azure - Training
- Windows Azure Developer Portal

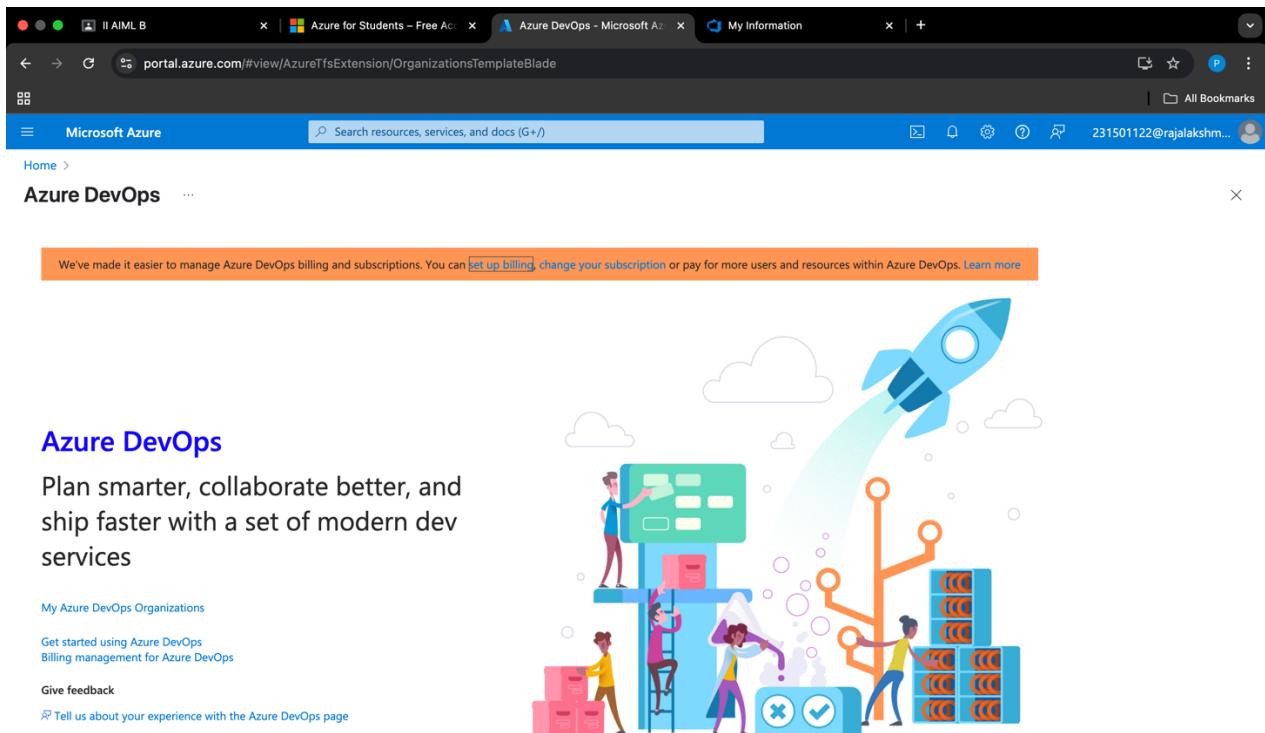
Azure for Students

Get free software, Azure credit, or access Azure Dev Tools for Teaching after you verify your academic status.

**Start**

Virtual machines App Services SQL databases More services

4. 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.

**EXP NO: 2**

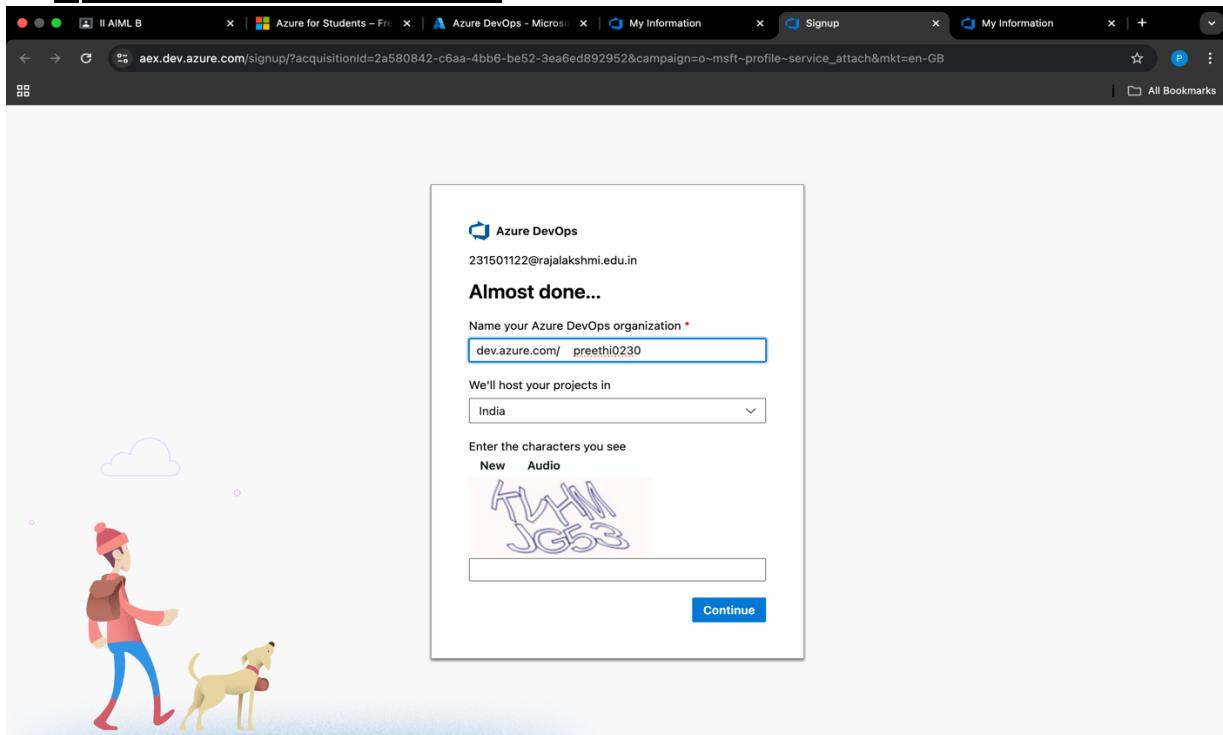
## **AZURE DEVOPS PROJECT SETUP AND USER STORY MANAGEMENT**

**Date :**

### **Aim:**

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

### **1. Create An Azure Account**



### **2. Create the First Project in Your Organization**

a. 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.

b. On the organization's **Home page**, click on the **New Project** button.

c. Enter the project name, description, and visibility options:

**Name:** Choose a name for the project (e.g., **LMS**).

**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).

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

## Create new project

X

Project name \*

Batch Data Analysis and Visualizations

Description

Visibility



Public

Anyone on the internet can view the project. Certain features like TFVC are not supported.



Private

Only people you give access to will be able to view this project.



Public projects are disabled for your organization. You can turn on public visibility with [organization policies](#).

^ Advanced

Version control [?](#)

Git

Work item process [?](#)

Agile

Cancel

Create

- 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.

A screenshot of a web browser showing the Microsoft My Information page. The URL is [aex.dev.azure.com/me?mkt=en-GB](https://aex.dev.azure.com/me?mkt=en-GB). The page displays a large red circular profile picture with the letters 'PG'. Below it, the user's name is Preethi Gopinath, email is 231501122@rajalakshmi.edu.in, and location is India. To the right, there is a list of Azure DevOps Organizations under 'dev.azure.com/231501122' (Owner). Projects listed include WORKITEMS, Assignment, Ecommerce, and ATM. Actions link to 'Open in Visual Studio'. A 'Create new organization' button is also present.

Preethi Gopinath  
231501122@rajalakshmi.edu.in

India  
231501122@rajalakshmi.edu.in

#### Visual Studio Dev Essentials

Get everything you need to build and deploy your app on any platform.

[Use your benefits](#)

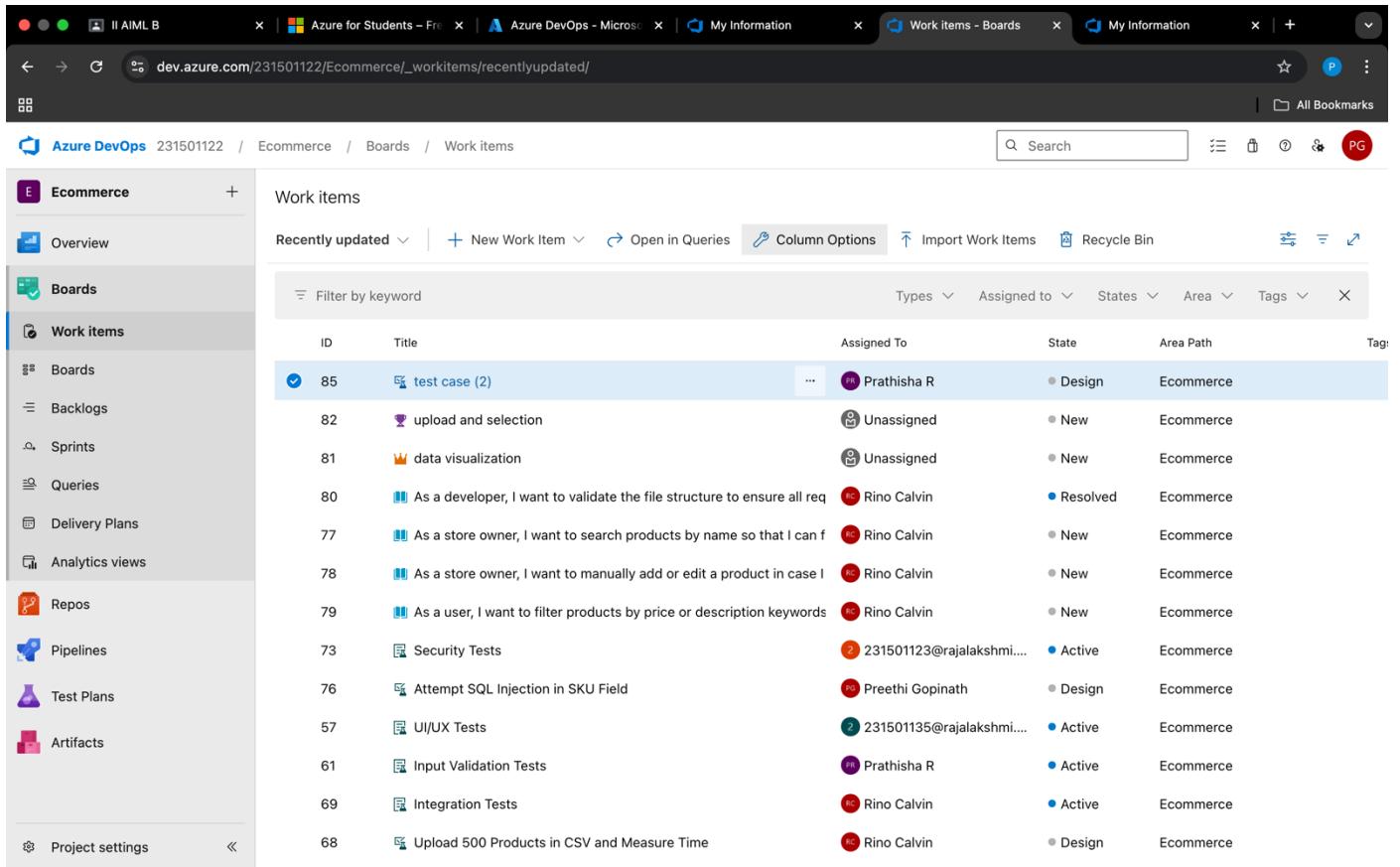
## 4. Project dashboard

A screenshot of the Azure DevOps Ecommerce project dashboard. The URL is [dev.azure.com/231501122/Ecommerce](https://dev.azure.com/231501122/Ecommerce). The left sidebar shows project navigation with 'Ecommerce' selected. The main content area displays the 'Ecommerce' project summary. It includes sections for 'About this project' (describing the E-commerce Product Uploader application), 'Project stats' (last 7 days), and 'Members' (6 members listed with icons). The 'Project stats' section provides metrics for Work items (38 created, 0 completed), Pull requests (0 opened), and Pipelines (0% succeeded).

## **5. To manage user stories:**

- a. 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.

b. 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.



The screenshot shows the Azure DevOps interface for the Ecommerce project. The left sidebar is collapsed, showing options like Overview, Boards, Work items, Boards, Backlogs, Sprints, Queries, Delivery Plans, Analytics views, Repos, Pipelines, Test Plans, Artifacts, and Project settings. The main area is titled "Work items" and displays a list of recently updated work items. The columns include ID, Title, Assigned To, State, and Area Path. One item is selected, showing its details: ID 85, Title "test case (2)", Assigned To Prathisha R, State Design, Area Path Ecommerce. Other items listed include "upload and selection", "data visualization", "As a developer, I want to validate the file structure to ensure all req", etc.

ID	Title	Assigned To	State	Area Path
85	test case (2)	Prathisha R	Design	Ecommerce
82	upload and selection	Unassigned	New	Ecommerce
81	data visualization	Unassigned	New	Ecommerce
80	As a developer, I want to validate the file structure to ensure all req	Rino Calvin	Resolved	Ecommerce
77	As a store owner, I want to search products by name so that I can f	Rino Calvin	New	Ecommerce
78	As a store owner, I want to manually add or edit a product in case I	Rino Calvin	New	Ecommerce
79	As a user, I want to filter products by price or description keywords	Rino Calvin	New	Ecommerce
73	Security Tests	231501123@rajalakshmi....	Active	Ecommerce
76	Attempt SQL Injection in SKU Field	Preethi Gopinath	Design	Ecommerce
57	UI/UX Tests	231501135@rajalakshmi....	Active	Ecommerce
61	Input Validation Tests	Prathisha R	Active	Ecommerce
69	Integration Tests	Rino Calvin	Active	Ecommerce
68	Upload 500 Products in CSV and Measure Time	Rino Calvin	Design	Ecommerce

**Result:** Successfully created an Azure DevOps project with user story management and agile workflow setup.

**EXP NO: 3**

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

**Date :**

### **Aim:**

To create epics, user stories, features, and tasks for the project, Ecommerce Product Uploader.

### **1.Create Epic, Features, User Stories, Task**

The screenshot shows the Azure DevOps Backlog board for the Ecommerce project. The backlog is organized into several epics and their associated features, user stories, and tasks. The columns include Order, Work Item Type, Title, State, Effort, Business Area, and Tags. Most items are marked as 'New'.

Order	Work Item Type	Title	State	Effort	Business Area	Tags
1	Epic	➡️ data visualization	New		Business	
2	Epic	➡️ Search and Pagination	New		Business	
	Feature	➡️ Search Functionality	New		Business	
	User Story	➡️ As a store owner, I want to search products by name...	New		Business	
	Task	➡️ Create a search bar with debounced input.	New			
	Task	➡️ Implement backend API to search by name or key...	New			
	User Story	➡️ As a user, I want to filter products by price or descr...	New		Business	
	Task	➡️ Add filter options (e.g., price range)	New			
	Task	➡️ Update query logic in backend accordingly	New			
3	Epic	➡️ product management	New		Business	
	Feature	➡️ Create and Update Products	New		Business	
	User Story	➡️ As a store owner, I want to manually add or edit a p...	New		Business	
	User Story	➡️ As a QA, I want the product edit form to auto-fill fie...	New		Business	
4	Epic	➡️ bulk product upload	New		Business	
	Feature	➡️ CSV/Excel File Upload	New		Business	

## 2. Fill in Epics

The screenshot shows the Azure DevOps Backlog page for the Ecommerce Team. The backlog is organized into three epics:

- Epic 1: Search and Pagination
  - Feature: Search Functionality
  - User Story: As a store owner, I want to search products by name...
  - Task: Create a search bar with debounced input.
  - Task: Implement backend API to search by name or key...
  - User Story: As a user, I want to filter products by price or descr...
  - Task: Add filter options (e.g., price range)
  - Task: Update query logic in backend accordingly
- Epic 2: product management
  - Feature: Create and Update Products
  - User Story: As a store owner, I want to manually add or edit a p...
  - User Story: As a QA, I want the product edit form to auto-fill fie...
- Epic 3: bulk product upload
  - Feature: CSV/Excel File Upload

## 3. Fill in Features

The screenshot shows the Azure DevOps Work Item creation dialog for a Feature work item. The work item details are as follows:

- Title:** file upload and selection
- State:** New
- Reason:** New
- Area:** Ecommerce
- Iteration:** Ecommerce\sprint3
- Description:** Click to add Description.
- Planning:**
  - Priority: 2
  - Risk:
  - Business Value:
  - Time Criticality:
  - Start Date: Select a date...
  - Target Date: Select a date...
- Deployment:** To track releases associated with this work item, go to [Releases](#) and turn on deployment status reporting for Boards in your pipeline's Options menu. [Learn more about deployment status reporting](#).
- Development:** Add link
- Classification:** Value area
- Related Work:** Add link

## 4. Fill in User Stories

The screenshot shows the 'New User Story' creation screen in Azure DevOps. The URL in the browser is [dev.azure.com/231501122/Ecommerce/\\_backlogs/backlog/Ecommerce%20Team/Epics](https://dev.azure.com/231501122/Ecommerce/_backlogs/backlog/Ecommerce%20Team/Epics). The main content area displays a user story template:

As a user I want to upload multiple csv files at once so I can analyze them together |

Below the template, there are several fields:

- State: New
- Area: Ecommerce
- Iteration: Ecommerce\sprint3
- Comments: 0
- Add Tag

On the right side, there are sections for 'Details' (with a 'Save and Close' button), 'Planning' (Story Points, Priority 2, Risk), 'Deployment' (instructions to track releases via Releases and deployment status reporting), 'Development' (Add link, linking to Azure Repos), and 'Related Work' (Add link). The sidebar on the left lists various project management sections: Backlog, Sprint, Acceptance Criteria, Description, Classification, Development, and Related Work.

**Result:** Thus, epics, features, user stories, and tasks have been created successfully.

**EXP NO: 4**

**Date :**

## **SPRINT PLANNING**

### **Aim:**

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

### **SPRINT PLANNING**

#### **Sprint 1**

The screenshot shows the Azure DevOps Taskboard for the 'Ecommerce' project under the 'Ecommerce Team' sprint. The taskboard is divided into columns: New, Active, Resolved, and Closed. Two user stories are visible in the 'New' column:

- User Story 79: As a user, I want to filter products by price or description keywords. Assigned to Rino Calvin.
- User Story 78: As a store owner, I want to manually add or edit a product in case I miss it during bulk upload. Assigned to Preethi Gopinath.

The left sidebar shows the project navigation menu with 'Sprints' selected. The top navigation bar includes links for 'Overview', 'Boards', 'Sprints', 'Queries', 'Delivery Plans', and 'Analytics views'. The URL in the browser is dev.azure.com/231501122/Ecommerce/\_sprints/taskboard/Ecommerce%20Team/Ecommerce/sprint%201.

## Sprint 2

The screenshot shows the Azure DevOps Taskboard for the Ecommerce Team in Sprint 2. The backlog is visible on the left, and the taskboard grid shows work items categorized by status: New, Active, Resolved, and Closed. The grid has columns for Sprint, Person, and Work Item details.

Sprint	Person	Work Item Details
Sprint2	All	<ul style="list-style-type: none"><li>23 As a store owner, I want to upload a CSV/Excel file so that I can add multiple products at once. New Preethi Gopinath</li><li>24 Implement file upload UI and backend endpoint. New Unassigned</li><li>25 Parse CSV/Excel file and extract product fields. New Unassigned</li><li>26 As a developer, I want to validate the file structure to ensure all required columns are present. New Preethi Gopinath</li><li>27 Check headers for required fields (name, description, price, image). New Unassigned</li><li>28 Return user-friendly error messages on Unassigned</li></ul>

## Sprint 3

The screenshot shows the Azure DevOps Taskboard for the Ecommerce Team in Sprint 3. The backlog is visible on the left, and the taskboard grid shows work items categorized by status: New, Active, Resolved, and Closed. The grid has columns for Sprint, Person, and Work Item details.

Sprint	Person	Work Item Details
sprint3	All	<ul style="list-style-type: none"><li>80 As a developer, I want to validate the file structure to ensure all required columns are present. Resolved Rino Calvin</li><li>77 As a store owner, I want to search products by name so that I can find specific items quickly. New Rino Calvin</li><li>88 As a user I want to upload multiple csv files at once so I can analyze them together New Preethi Gopinath</li></ul>

**Result:** The Sprints are created for the project, Ecommerce Product Uploader.

**EXP NO: 5**

# POKER ESTIMATION

**Date :**

## Aim:

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

## Poker Estimation

The screenshot shows the Azure DevOps interface for a User Story titled "23 As a store owner, I want to upload a CSV/Excel file so that I can add multiple products at once." The story is currently in the "Resolved" state, assigned to Preethi Gopinath, and has a priority of 2. The "Planning" section indicates 2 Story Points. The "Classification" section shows the value area as "Business". The "Development" section provides instructions on linking to Azure Repos. The "Related Work" section lists three items: "User Story 24", "User Story 25", and "User Story 26".

## Result:

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

**EXP NO: 6**

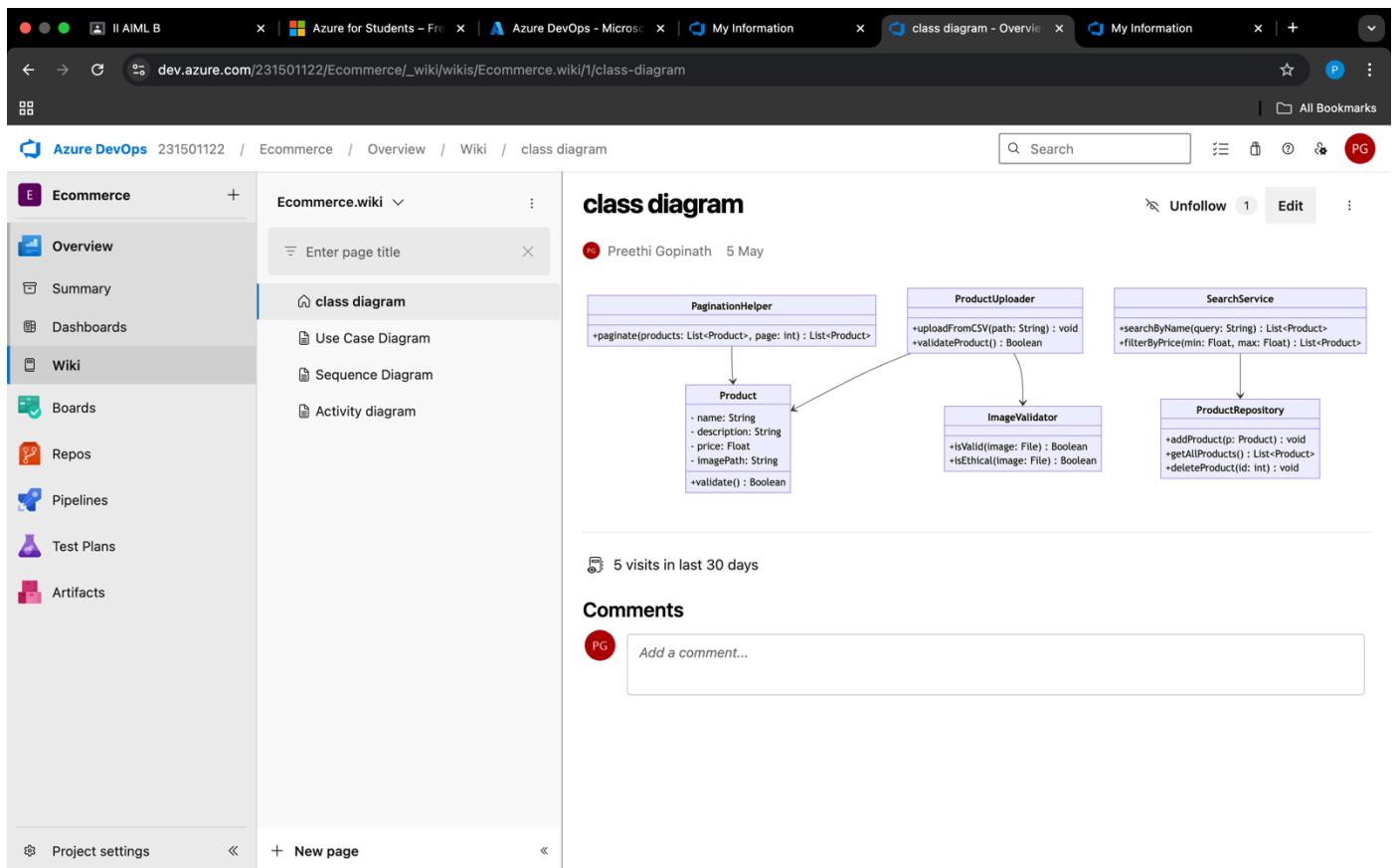
## **DESIGNING CLASS DIAGRAM AND SEQUENCE DIAGRAM**

**Date :**

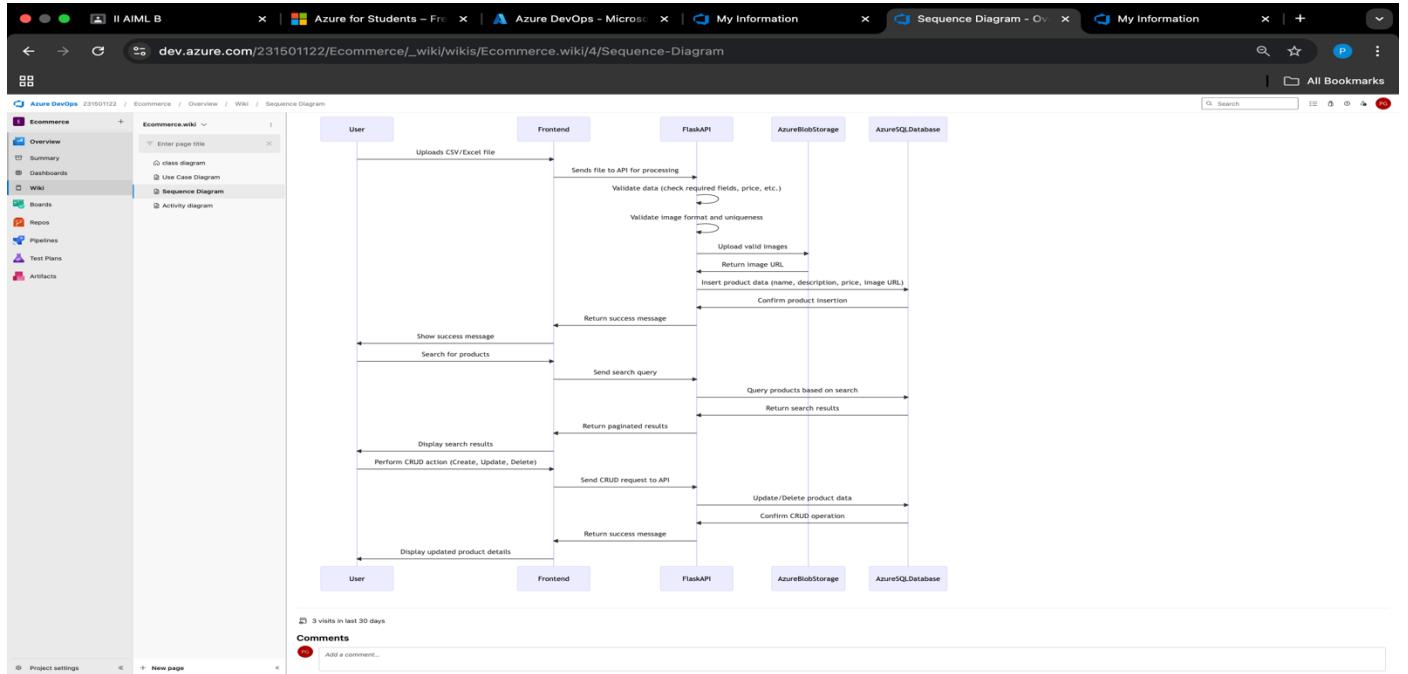
### **Aim:**

To design a Class Diagram and Sequence Diagram for the project, Ecommerce Product Uploader.

### **6A. Class Diagram**



## 6B. Sequence Diagram



**Result:** The Class and Sequence Diagrams are designed successfully for the project, Ecommerce Product Uploader.

**EXP NO: 7**

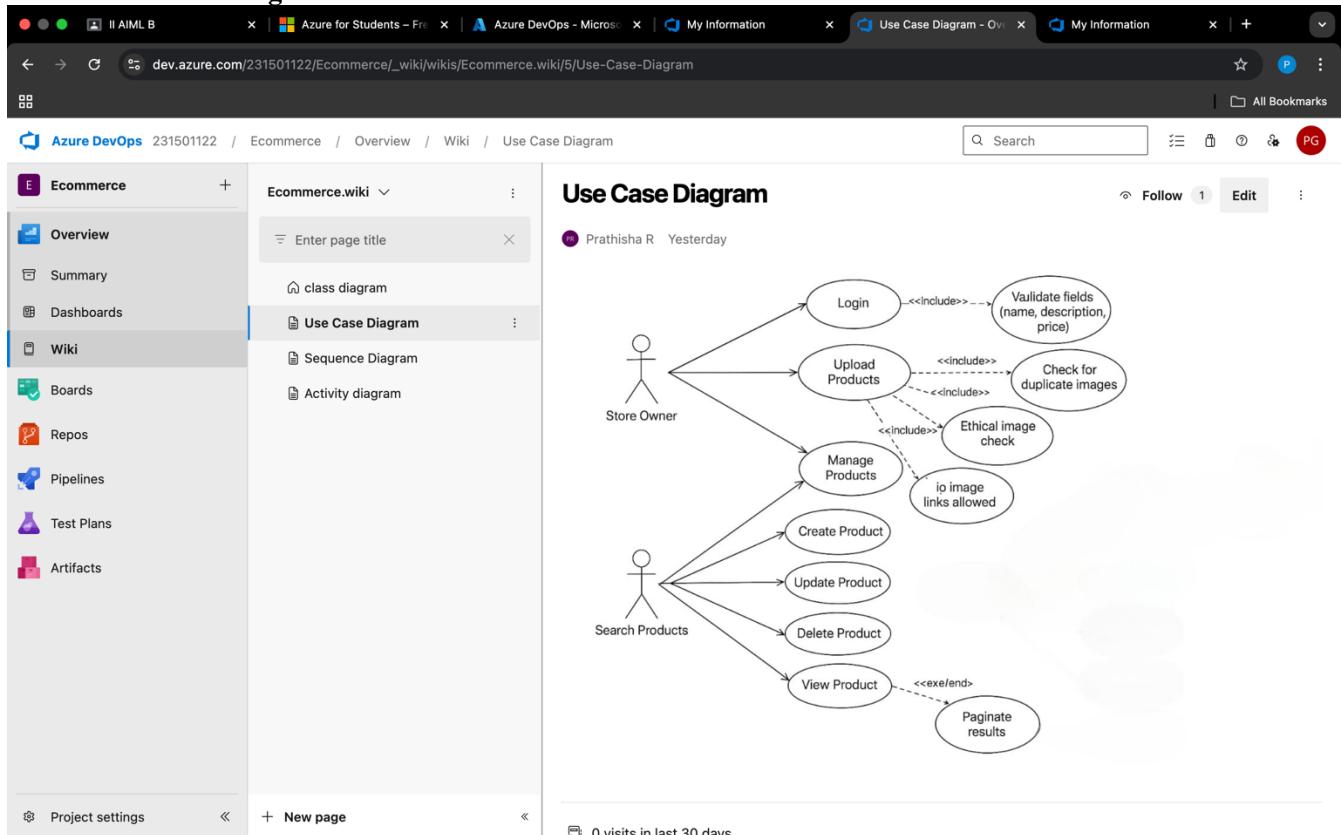
## **DESIGNING USE CASE DIAGRAM AND ACTIVITY DIAGRAM**

**Date :**

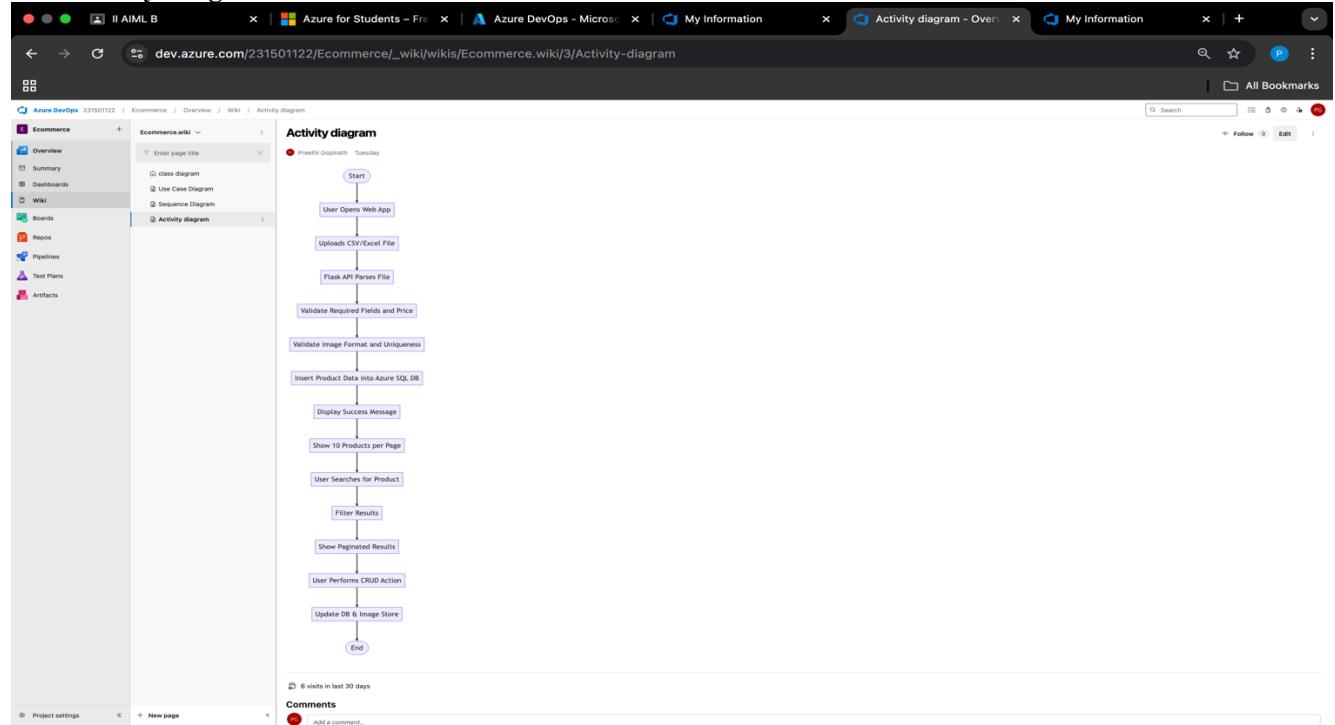
### **Aim:**

To design a Use Case Diagram and an Activity Diagram for the project, Ecommerce Product Uploader.

### **7A. Use Case Diagram**



## 7B. Activity Diagram



**Result:** The Use Case and Activity Diagrams are designed successfully for the project, Ecommerce Product Uploader.

<b>EXP NO: 8</b>	<b>TESTING – TEST PLANS AND TEST CASES</b>
<b>Date :</b>	

### **Aim:**

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**

- 1. User Authentication
- 2. Uploading and Managing Batch Data Files
- 3. Running Batch Analysis Jobs
- 4. Viewing Interactive Visualizations and Charts
- 5. Exporting Analysis Results

#### **2. Define User Interactions**

- Simulate real scenarios (e.g., upload dataset, trigger job, download result).

#### **3. Design Happy Path Test Cases**

- Validate all main functions work properly (e.g., successful login, upload, and visualization).

#### **4. Design Error Path Test Cases**

- Simulate unexpected or invalid user behavior (e.g., upload fails, unsupported file, job timeout).

#### **5. Break Down Steps and Expected Results**

- Each test case includes step-by-step actions and expected outcomes.

#### **6. Use Clear Naming and IDs**

- Example: TC01 – Successful File Upload, TC08 – Visualization Fails.

#### **7. Separate Test Suites**

- Suites grouped by modules (Login, File Upload, Job Execution, Visualization, Export).

#### **8. Prioritize and Review**

- Critical test cases marked as High Priority.

- Mapped to user stories in Azure DevOps.

## 1. New test plan

The screenshot shows the 'New Test Plan' dialog in the Azure DevOps interface. On the left, there's a sidebar with project navigation links like Overview, Boards, Repos, Pipelines, Test Plans, and Artifacts. The 'Test Plans' section is currently selected. The main area is titled 'New Test Plan' and contains fields for 'Name' (set to 'performance tests'), 'Area Path' (set to 'Ecommerce'), and 'Iteration' (set to 'Ecommerce\sprint3'). At the bottom right are 'Create' and 'Cancel' buttons.

## 2. Test suite

The screenshot shows the 'Functional tests for upload' page in the Azure DevOps interface. The sidebar shows the 'Test plans' section is selected. The main area displays a 'bulk upload(CSV) (ID: 53)' entry under 'Test Suites'. This entry has a 'Define' tab active, showing a 'Test Cases (1 item)' table. The table contains one row for 'Upload Multiple Products Using Valid CSV File' (Order 1, Test Case Id 54, Assigned to Preethi). There are also 'Execute' and 'Chart' tabs available.

### **3. Test case**

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

## USER STORIES

- As a user, I want to log in using my username and password so that I can access my account.
- As a user, I should not be able to submit the login form with empty fields so that I can provide the required data.
- As a user, I want to log out when I click the logout button so that I can end my session securely.
- As a user, I want to be redirected to the login page after logging out so that I know my session has ended and I can log in again if needed.
- As a user, I want to be able to upload multiple CSV files at once, so I can analyze them together.

## Test Suites

### Test Suite: TS01 - User Authentication (ID: 54)

1. TC01 – Successful Login
  - **Action:**
    - Navigate to the login page
    - Enter valid credentials
    - Click "Login"
  - **Expected Results:**
    - User redirected to dashboard.
  - **Type:** Happy Path

2. TC02 – Prevent Login with Empty Fields

- **Action:**
  - Navigate to the login page.
  - Leave username and/or password fields empty.
  - Click on "Login".
- **Expected Results:**
  - Validation error message is shown prompting user to fill required fields.
- **Type:** Error Path
- 

### Test Suite: TS02 - Logout Functionality (ID: 47)

1. TC03 – Successful Logout and Redirect

- **Action:**
  - Log in successfully.
  - Click the "Logout" button.
- **Expected Results:**

- User session ends.
- User is redirected to the login page.
- **Type:** Happy Path

## 2. TC04 – Access Protected Page After Logout

- **Action:**
  - Logout.
  - Attempt to navigate back to a protected page (e.g., dashboard) via browser back button or URL.
- **Expected Results:**
  - User is redirected to the login page and denied access.
- **Type:** Error Path

## Test Suite: TS03 - CSV Upload Functionality (ID: 88)

### 1. TC05 – Upload Multiple Valid CSV Files

- **Action:**
  - Log in successfully
  - Navigate to the CSV upload section
  - Select multiple valid .csv files
  - Click "Upload"
- **Expected Results:**
  - All files are uploaded successfully.
  - Files are listed and ready for analysis.
- **Type:** Happy Path

### 2. TC06 – Upload Attempt Without Selecting Files

- **Action:**
  - Navigate to the CSV upload section
  - Click "Upload" without selecting any files.
- **Expected Results:**
  - Validation message prompting the user to select at least one file.
- **Type:** Error Path

## Test Cases

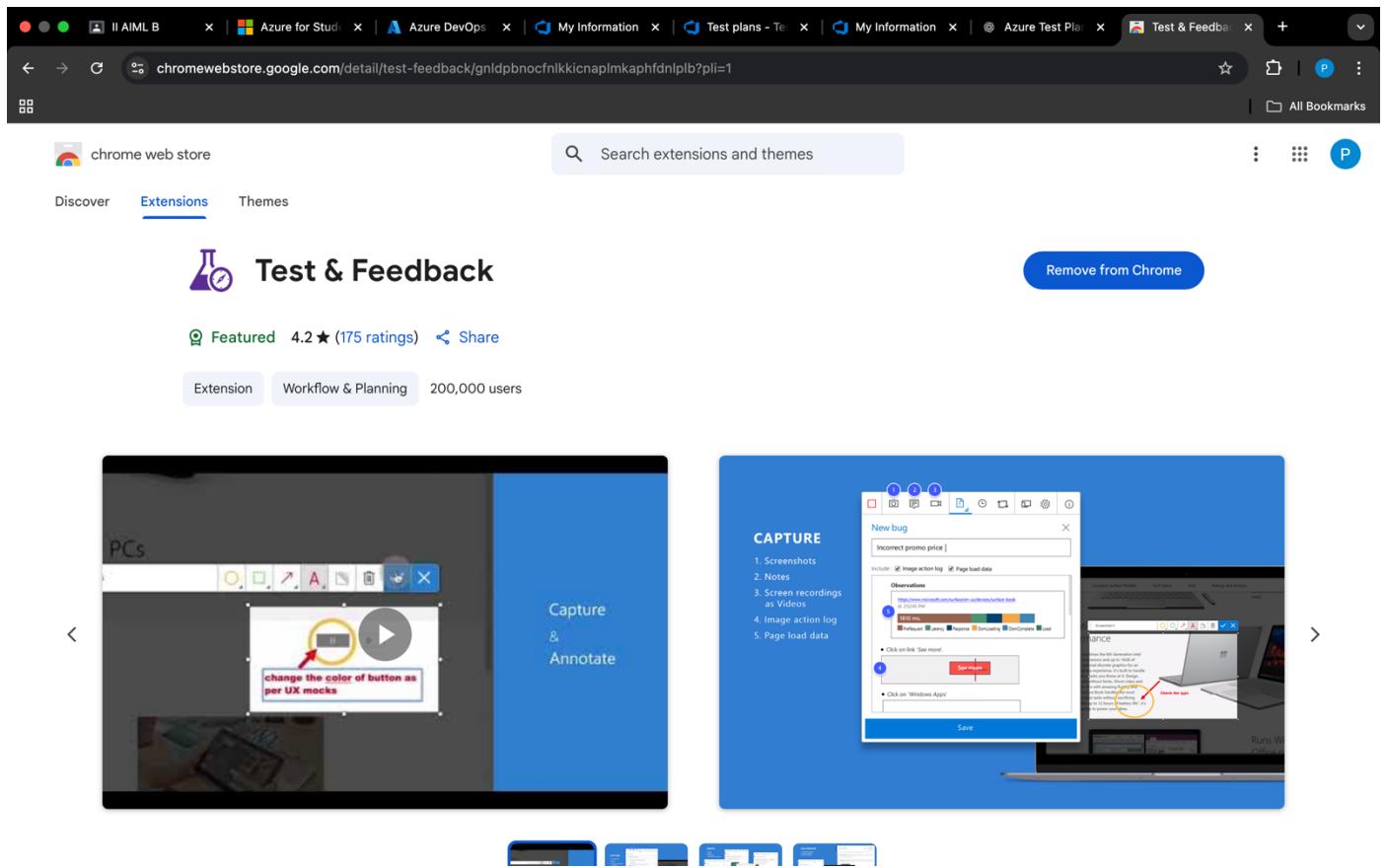
The screenshot shows the Azure DevOps interface for a project named "Ecommerce". The left sidebar is open, showing various project management and development tools like Boards, Repos, Pipelines, and Test Plans. Under "Test Plans", the "Test plans" section is selected. In the main content area, a "UI/UX Tests" plan is displayed. A single test suite named "Mobile Responsiveness (1)" is listed under "Test Suites". The details for this suite are shown in the right panel, titled "Mobile Responsiveness (ID: 59)". The "Test Cases (1 item)" table contains one entry:

	Title	Order	Test Case Id	Assign
<input type="checkbox"/>	Check Product Upload Page on Mobile Devices	1	60	Preethi

This screenshot shows the same "Ecommerce" project in Azure DevOps. The "Test plans" section is selected in the sidebar. A different test plan, "Input Validation Tests", is currently active. A single test suite named "Missing Required Fields (1)" is listed under "Test Suites". The details for this suite are shown in the right panel, titled "Missing Required Fields (ID: 63)". The "Test Cases (1 item)" table contains one entry:

	Title	Order	Test Case Id	Assign
<input type="checkbox"/>	Try Upload Without Entering Product Name or Price	1	64	Preethi

#### 4. Installation of test



## Test and feedback

Showing it as an extension

The screenshot shows the Azure DevOps interface for a project named "Ecommerce". The left sidebar is open, showing various project management sections like Overview, Boards, Repos, Pipelines, Test Plans, and Artifacts. The "Test Plans" section is currently selected. In the main content area, a test plan titled "Database Sync Check (ID: 71)" is displayed. This plan has three tabs: Define, Execute (which is selected), and Chart. Under the Execute tab, there is a section for "Test Points (1 item)". A single test point is listed with the title "Check Product Entry in DB After Upload". The status of this test point is "Active", and it is assigned to "Order 1" with a total count of 72. A tooltip for the "Test & Feedback" extension is visible, stating "Full access. These extensions can see and change information on this site." and "Manage Extensions". The URL in the browser bar is [https://dev.azure.com/231501122/Ecommerce/\\_testPlans/execute?planId=69&suiteId=71](https://dev.azure.com/231501122/Ecommerce/_testPlans/execute?planId=69&suiteId=71).

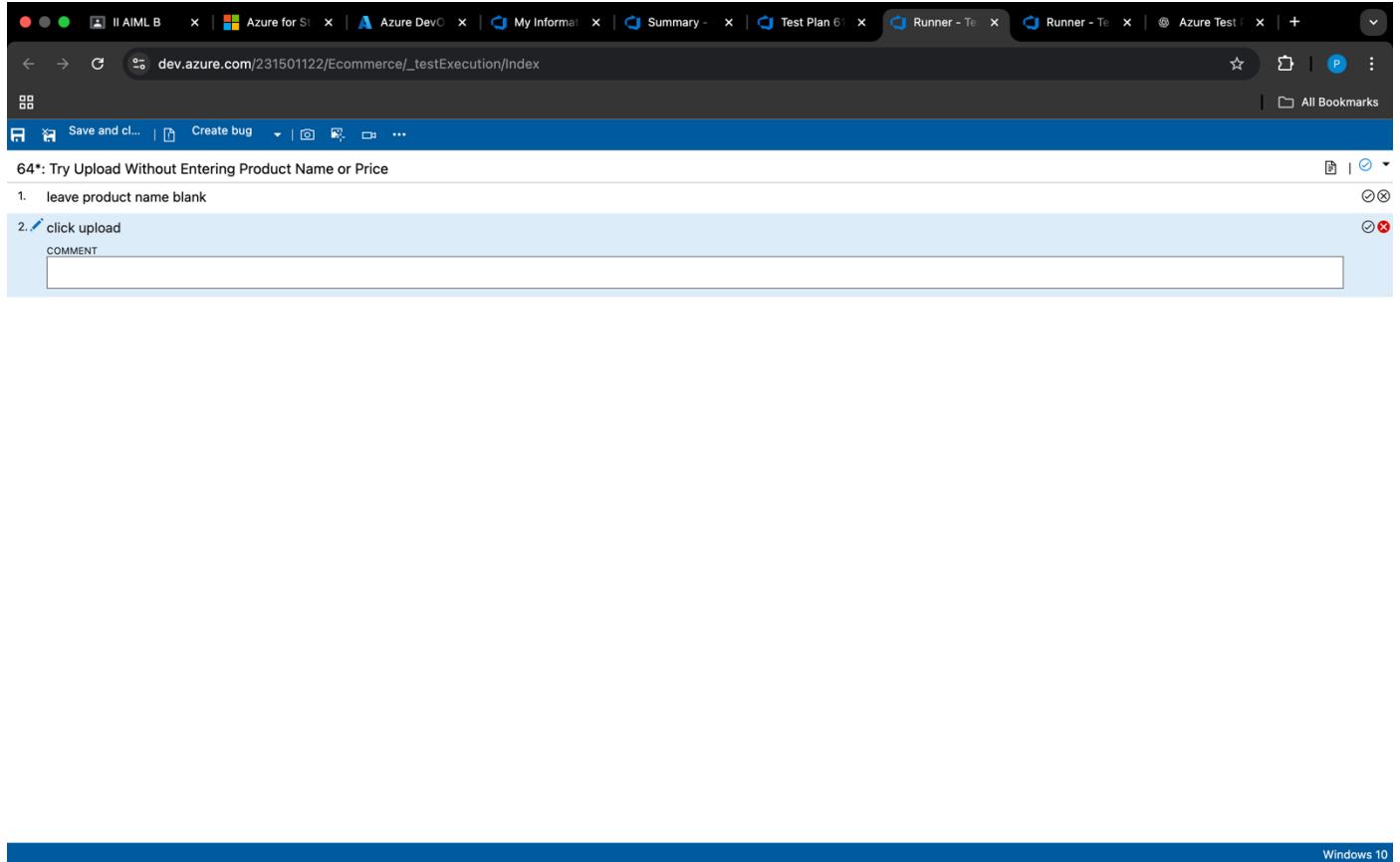
## 5. Running the test cases

The screenshot shows the Azure DevOps Test Plans interface. On the left, there's a sidebar with project navigation. In the center, a 'Test Suites' section displays a single suite named 'Database Sync Check (1)'. On the right, the main area shows a 'Test Points (1 item)' table with one row:

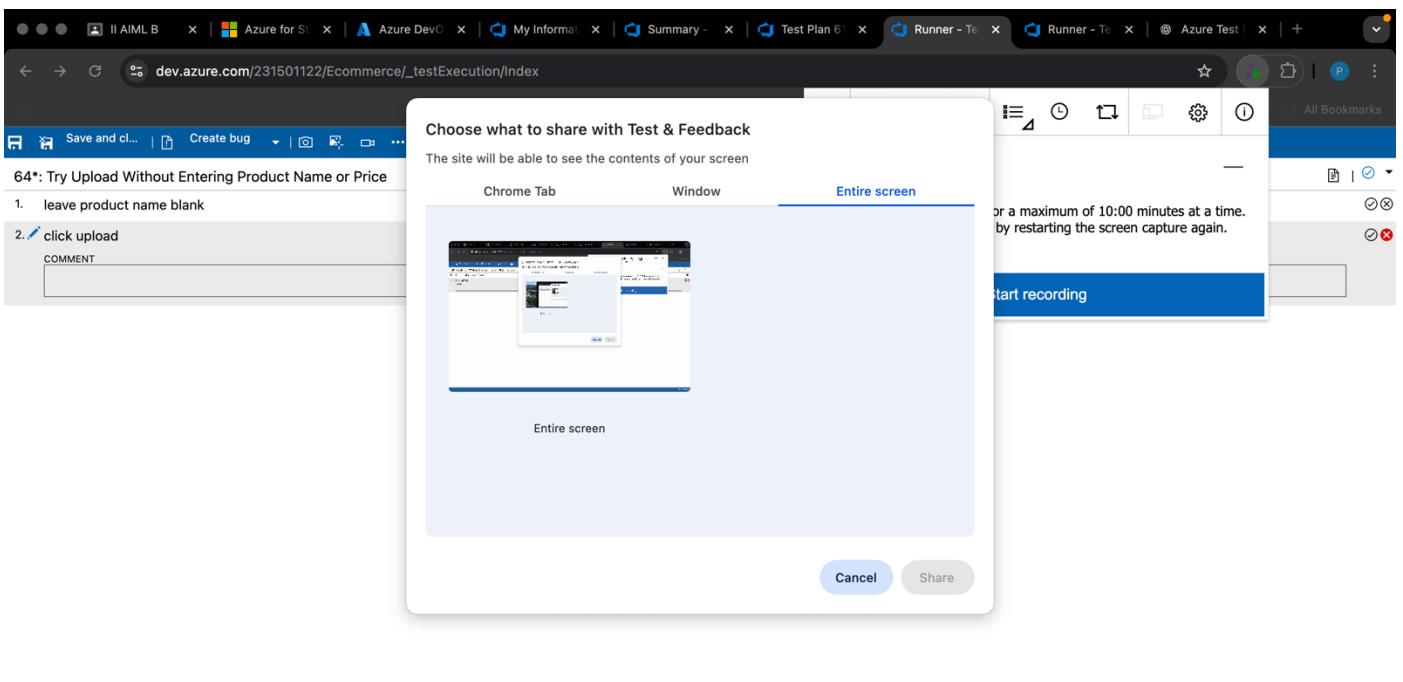
Title	Outcome	Order	Test ID
Check Product Entry in DB After Upload	Passed	1	72

A context menu is open over the first row of the table, listing options: 'View execution history', 'Mark Outcome' (with 'Run' as a submenu), 'Reset test to active', 'Edit test case', 'Assign tester', and 'View test result'. At the top of the page, there are tabs for 'Define', 'Execute' (which is selected), and 'Chart'. A search bar and a help icon are also visible at the top.

This screenshot is similar to the one above, showing the same interface elements. However, a 'Extensions' overlay is displayed on the right side of the screen. The overlay has a title 'Extensions' and a sub-section 'Full access' which states: 'These extensions can see and change information on this site.' It lists a single extension: 'Test & Feedback' (represented by a purple icon). Below this is a 'Manage Extensions' button. The rest of the interface is identical to the first screenshot, including the sidebar, test suite list, and the open context menu over the first test point.

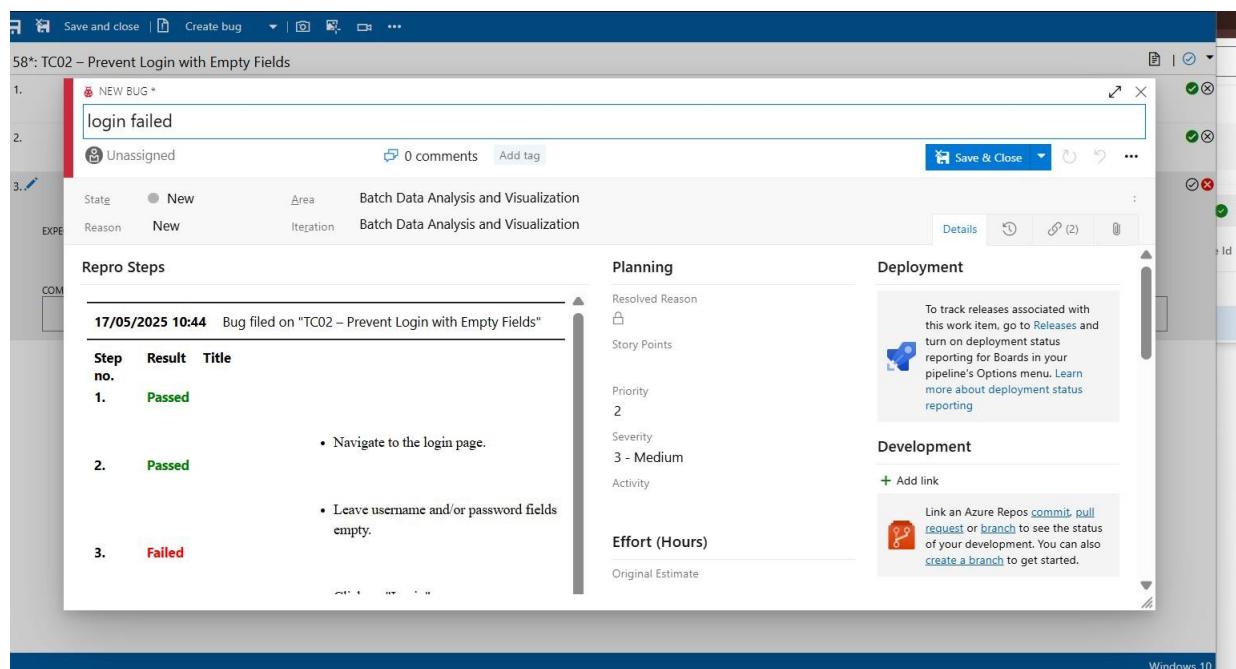


## 6. Recording the test case



Windows 10

## 7. Creating the bug



64\*: Try Upload Without Entering Product Name or Price

1. leave product name blank  
2. click upload

**NEW BUG \***

**login failed**

Unassigned 0 comments Add tag

Save & Close (⌘+Enter)

**Repro Steps**

22/05/2025 04:19 Bug filed on "Try Upload Without Entering Product Name or Price"

Step no.	Result	Title
1.	None	leave product name blank
2.	Failed	click upload

**Test Configuration:** Windows 10

**System Info**

Browser - Name	Google Chrome 136
Browser - Language	en-GB
Browser - Height	778

**Planning**

Resolved Reason  
Story Points  
Priority 2  
Severity 3 - Medium  
Activity

**Deployment**

To track releases associated with this work item, go to Releases and turn on deployment status reporting for Boards in your pipeline's Options menu. Learn more about deployment status reporting

**Development**

+ Add link Link an Azure Repos commit, pull request or branch to see the status of your development. You can also create a branch to get started.

**Effort (Hours)**

Original Estimate  
Remaining  
Completed

**Related Work**

+ Add link Add an existing work item as a parent

BUG 60

60 not logging in due to system error

No one selected 0 Comments Add Tag

Save & Close Follow Details

Updated by Shri Dharshini Just now

State: New Area: Batch Data Analysis and Visualization  
Reason: New Iteration: Batch Data Analysis and Visualization

**Repro Steps**

17/05/2025 10:51 Bug filed on "TC02 – Prevent Login with Empty Fields"

Step no.	Result	Title
1.	Passed	• Navigate to the login page.
2.	Passed	• Leave username and/or password fields empty.
3.	Failed	• Click on "Login".

**Expected Result**

- Validation error message is shown prompting user to fill required fields.

**Planning**

Resolved Reason  
Story Points  
Priority 2  
Severity 3 - Medium  
Activity

**Deployment**

To track releases associated with this work item, go to Releases and turn on deployment status reporting for Boards in your pipeline's Options menu. Learn more about deployment status reporting

**Development**

Add link Link an Azure Repos commit, pull request or branch to see the status of your development. You can also create a branch to get started.

**Effort (Hours)**

Original Estimate  
Remaining  
Completed

**Related Work**

Add link Add an existing work item as a parent

Tested By 58 TC02 – Prevent Login with Empty Fields

## 8. Test case results

The screenshot shows the Microsoft Test Manager interface. On the left, there's a navigation pane with various icons. The main area displays a 'Test Suites' list under 'batch data analysis'. One suite, 'TS01- User Authentication (2)', is expanded, showing two test points: 'TC01 – Successful Login' (Passed) and 'TC02 – Prevent Login with Empty Fields' (Not Run). To the right, a detailed view of 'TC01 – Successful Login' is shown, titled 'TC01 – Successful Login'. It includes a 'Test Case Results' table with three rows of data:

Outcome	TimeSta...	Configuration	Run by	Tester	Test
Passed	16m ago	Windows 10	Shri Dharshini	Shri Dharshini	batch
Failed	17m ago	Windows 10	Shri Dharshini	Shri Dharshini	batch
Passed	28m ago	Windows 10	Shri Dharshini	Shri Dharshini	batch

At the bottom of this window, there's a link: 'Open execution history for current test point'.

## 9. Test report summary

The screenshot shows the Azure Boards Work Item details page for a bug titled "BUG 60: not logging in due to system error". The work item is categorized under "Batch Data Analysis and Visualization" and is currently in the "New" state. The "Repro Steps" section contains three steps: 1. Passed (Navigate to the login page), 2. Passed (Leave username and/or password fields empty), and 3. Failed (Click on "Login"). The "Expected Result" is a validation error message prompting the user to fill required fields. The "Planning" section includes fields for Resolved Reason, Story Points, Priority (2), Severity (3 - Medium), and Activity. The "Deployment" section provides instructions for tracking releases. The "Development" section links to Azure Repos. The "Related Work" section allows adding links and existing work items. The work item was updated by Shri Dharshini 4m ago.

- Assigning bug to the developer and changing state

The screenshot shows the same Azure Boards Work Item details page for "BUG 60" after it has been assigned to "Shri Dharshini". The "State" is now "In Progress". The rest of the interface remains identical to the previous screenshot, showing the repro steps, expected result, planning details, deployment information, development links, and related work sections.

## 10. Progress report

The screenshot shows the Azure DevOps Test Plan Progress report interface. On the left, there's a sidebar with project navigation: Ecommerce (selected), Overview, Boards, Repos, Pipelines, Test Plans (selected), Test plans, Progress report (selected), Parameters, Configurations, Runs, and Artifacts. Below the sidebar is a 'Project settings' link. The main area has a header 'Progress report' with dropdown filters for Test Plans, Test Suites, Outcome, Configuration, Tester, Priority, and Assigned To. The main content is divided into two main sections: 'Summary' and 'Outcome trend'.

**Summary**

- 7 Test plans
- 9 Test points
- 1 (1 / 9) Test points run (11% Run)
- 100% (1 / 1) Pass rate (1 Passed)

**Outcome trend**

Last 14 Days

Date	Not run	Passed
2025-05-08	10	0
2025-05-09	9	0
2025-05-10	8	0
2025-05-11	7	0
2025-05-12	6	0
2025-05-13	5	0
2025-05-14	4	0
2025-05-15	3	0
2025-05-16	2	0
2025-05-17	1	0
2025-05-18	0	1
2025-05-19	0	1
2025-05-20	0	1
2025-05-21	0	2

## 11. Changing the test template

The screenshot shows the Azure DevOps 'Process' settings page. On the left, there's a sidebar with 'Organization Settings' and sections for General, Security, Boards, and Pipelines. The main area is titled 'All processes' and shows a list of templates under 'Processes'. The 'Agile (default)' template is selected, and its details are shown in the center: 'This template is flexible and will work great for most teams using Agile planning methods, incl...' and '0 Team projects'. Other visible templates include 'Basic', 'Agile plus', 'BatchDataAnalysis', 'Scrum', and 'CMMI'.

## 12. View the new test case template

The screenshot shows the 'Add a field to Test Case' dialog box over a background of the Azure DevOps 'Process' settings. The dialog has tabs for 'Definition', 'Options', and 'Layout'. Under 'Definition', the 'Create a field' option is selected, with 'Name' set to 'text' and 'Type' set to 'Text (single line)'. There's also a 'Description' field and a 'Learn more' link. At the bottom are 'Add field' and 'Cancel' buttons.

Azure DevOps 231501153 / Settings / Process

Organization Settings 231501153

All processes > BATCH DATA ANALYSIS > Test Case

Layout States Rules

New field New group New page Get extensions

Steps Summary Associated Aut...

Recent test results

Custom

Deployment

Development

Related Work

Status

Priority Integer

Automation status Text (single line)

Text (multiple lines)

Search Settings

General

- Overview
- Projects
- Users
- Billing
- Global notifications
- Usage
- Extensions
- Microsoft Entra

Security

- Security overview
- Policies
- Permissions

Boards

- Process

Pipelines

- Agent pools
- Settings

Azure DevOps 231501153 / Settings / Process

Organization Settings 231501153

All processes > Agile

Work item types Backlog levels Projects

Name Description

Batch data analysis and visualization ... About this project This project is a web-based application designed for batch data analysis and visualization, hosted on Microsoft Azure. It en...

Digital lending library application

digital library

SHRI DHARSHINI

Search Settings

General

- Overview
- Projects
- Users
- Billing
- Global notifications
- Usage
- Extensions
- Microsoft Entra

Security

- Security overview
- Policies
- Permissions

Boards

- Process

Pipelines

- Agent pools
- Settings

**Result:** The test plans and test cases for the user stories is created in Azure DevOps with Happy Path and Error Path.

EXP NO: 9	<b>CI/CD PIPELINES IN AZURE</b>
Date:	

### **Aim:**

To create and demonstrate an Azure DevOps pipeline for automating application builds, tests, and deployment.

### **PROCEDURE:**

#### **Steps to Create and implement pipelines in Azure:**

1. Sign in to Azure DevOps and Navigate to Your Project

Log in to [dev.azure.com](https://dev.azure.com), select your organization, and open the project where your Student Management System code resides.

2. Connect a Code Repository (Azure Repos or GitHub)

Ensure your application code is stored in a Git-based repository such as Azure Repos or GitHub. This will be the source for triggering builds and deployments in your pipeline.

3. Create a New Pipeline

Go to the Pipelines section on the left panel and click “Create Pipeline”.

Choose your source (e.g., Azure Repos Git or GitHub), and then select the repository containing your project code.

4. Choose the Pipeline Configuration

You can select either the YAML-based pipeline (recommended for version control and automation) or the Classic Editor for a GUI-based setup.

If using YAML, Azure DevOps will suggest a template or allow you to define your own.

5. Define Build Stage (CI - Continuous Integration) from YAML file

6. Install dependencies (e.g., npm install, dotnet restore)
7. Build the application (dotnet build, npm run build)
8. Run unit tests (dotnet test, npm test)
9. Publish build artifacts to be used in the release stage
10. Save and Run the Pipeline for the First Time

Save the YAML or build definition and click “Run”.

Azure will fetch the latest code and execute the defined build and test stages.

#### 11. Configure Continuous Deployment (CD)

Navigate to the Releases tab under Pipelines and click “New Release Pipeline”. Add an Artifact (from the build stage) and create a new Stage (e.g., Development, Production).

#### 12. Configure the CD stage with deployment tasks such as deploying to Azure App Service, running database migrations or scripts, and restarting services using the Azure App Service Deploy task linked to your subscription and app details.

#### 13. Set Triggers and Approvals

Enable continuous deployment trigger so the release pipeline runs automatically after a successful build.

For production environments, configure pre-deployment approvals to ensure manual verification before release.

#### 14. Monitor Pipelines and Manage Logs

View all pipeline runs under the Runs section.

Check logs for build/test/deploy stages to debug any errors.

You can also integrate email alerts or Microsoft Teams notifications for build failures.

#### 15. Review and Maintain Pipelines

Regularly update your pipeline tasks or YAML configurations as your application grows. Ensure pipeline runs are clean and artifacts are stored securely.

Integrate quality gates and code coverage policies to maintain code quality.

## Pipeline

The screenshot shows the Azure DevOps interface for the Ecommerce project. The left sidebar is visible with options like Overview, Boards, Repos, Pipelines, Test Plans, and Artifacts. The Pipelines section is selected. On the right, a pipeline run titled '#20250520.1 • Set up CI with Azure Pipelines' is displayed. The run status is green, indicating success. A note says 'This run will be cleaned up after 1 month based on your project settings.' Below the summary, there's a 'Summary' tab showing details about the individual CI run by Prathisha R, including repository information (Ecommerce, main branch, commit hash), time started and elapsed (Today at 7:00 PM, 20s), related work items (0), artifacts (0), and tests and coverage (3 commits). There are also tabs for 'Code Coverage' and 'Jobs'. The 'Jobs' section shows one job named 'Job' with a status of 'Success' and a duration of 12s.

## Result:

Successfully demonstrated pipelines in azure devops

**EXP NO: 10**

**Date :**

## **GITHUB: PROJECT STRUCTURE & NAMING CONVENTIONS**

### **Aim:**

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.

### **GitHub Project Structure**

Preethigopinat / Software-construction-

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Software-construction- Public

Pin Unwatch Fork Star 0

main Go to file + <> Code About

Preethigopinat Add files via upload 1558b15 · now

login.html.html Add files via upload 1 minute ago

uploader.html.html Add files via upload 1 minute ago

**README**

Add a README

Add a README

No description, website, or topics provided.

Activity 0 stars 1 watching 0 forks

Releases

No releases published Create a new release

Packages

No packages published Publish your first package

Languages

HTML 100.0%

Suggested workflows

Based on your tech stack

Jekyll using Docker image Configure

Package a Jekyll site using the jekyll/builder Docker image.

SLSA Generic generator Configure

Generate SLSA3 provenance for your existing release workflows

More workflows Dismiss suggestions

Terms Privacy Security Status Docs Contact Manage cookies Do not share my personal information

© 2025 GitHub, Inc.

### **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.