

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.

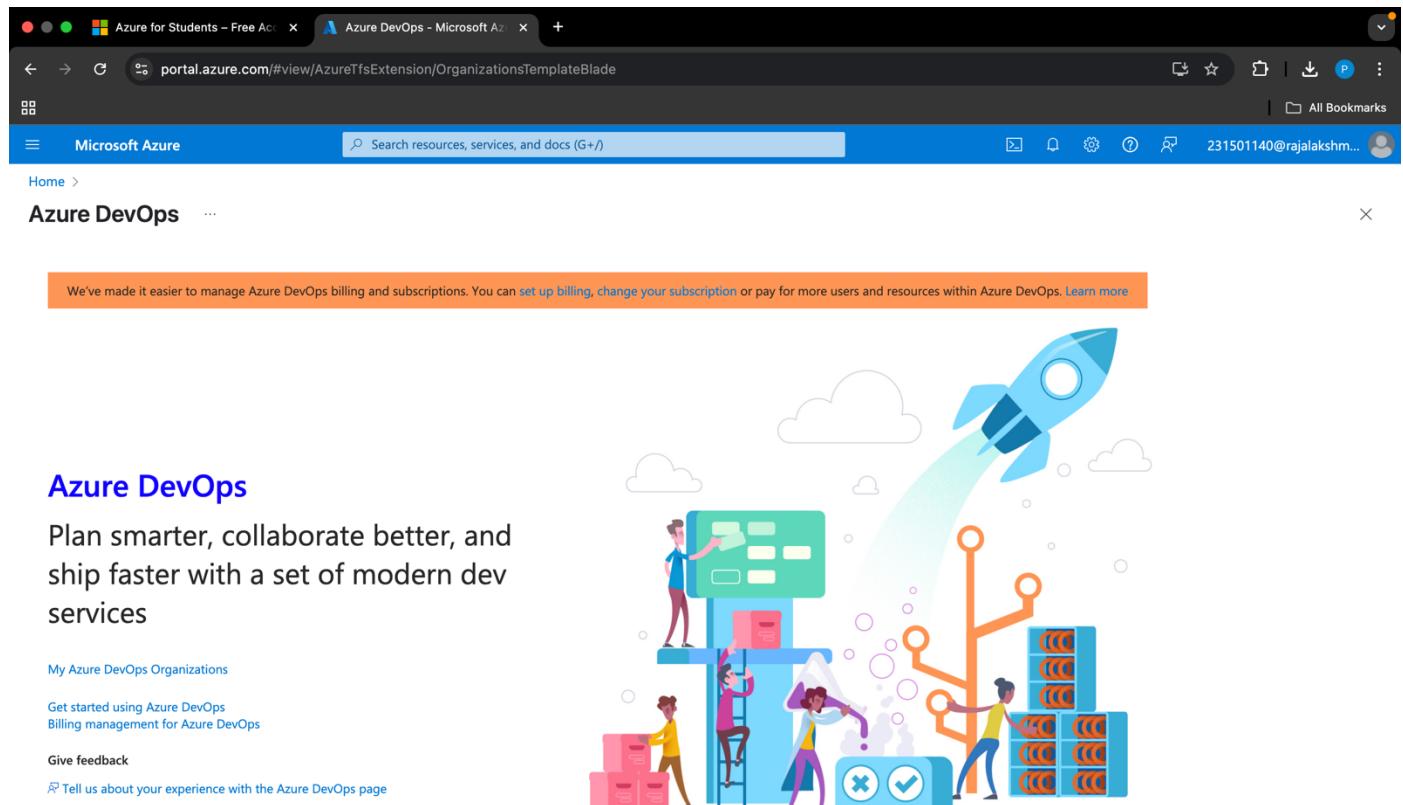
2. Azure home page

The screenshot shows the Microsoft Azure home page. At the top, there's a search bar with the placeholder "Search resources, services, and docs (G+/-)". Below the search bar, there's a navigation bar with icons for Home, Notifications, Settings, and Profile. The main content area is titled "Azure services" and features a grid of service icons: Create a resource, Azure DevOps organizations, Quickstart Center, Azure AI foundry, Kubernetes services, Virtual machines, App Services, Storage accounts, SQL databases, and More services. Below this, there's a section titled "Resources" with tabs for "Recent" and "Favorite". A table lists resources by Name, Type, and Last Viewed. A message says "No resources have been viewed recently" with a "View all resources" button. At the bottom, there are sections for "Navigate" (Subscriptions, Resource groups, All resources, Dashboard) and "Tools".

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

The screenshot shows the Microsoft Azure home page with a search overlay. The search bar at the top contains the text "azure devops". The search results are displayed in a modal window. The "All" tab is selected, showing "Services (99+)" and "Resources". Under "Services", there are icons for Azure DevOps organizations, Azure Cosmos DB, Azure Database for MySQL servers, and Azure Deployment Environments. There are also sections for "Marketplace", "Documentation", and "Last Viewed". The "Resources" tab is also visible. At the bottom of the search overlay, there's a "Continue searching in Microsoft Entra ID" link and a "give feedback" button. The rest of the page shows the standard Azure services dashboard.

- 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

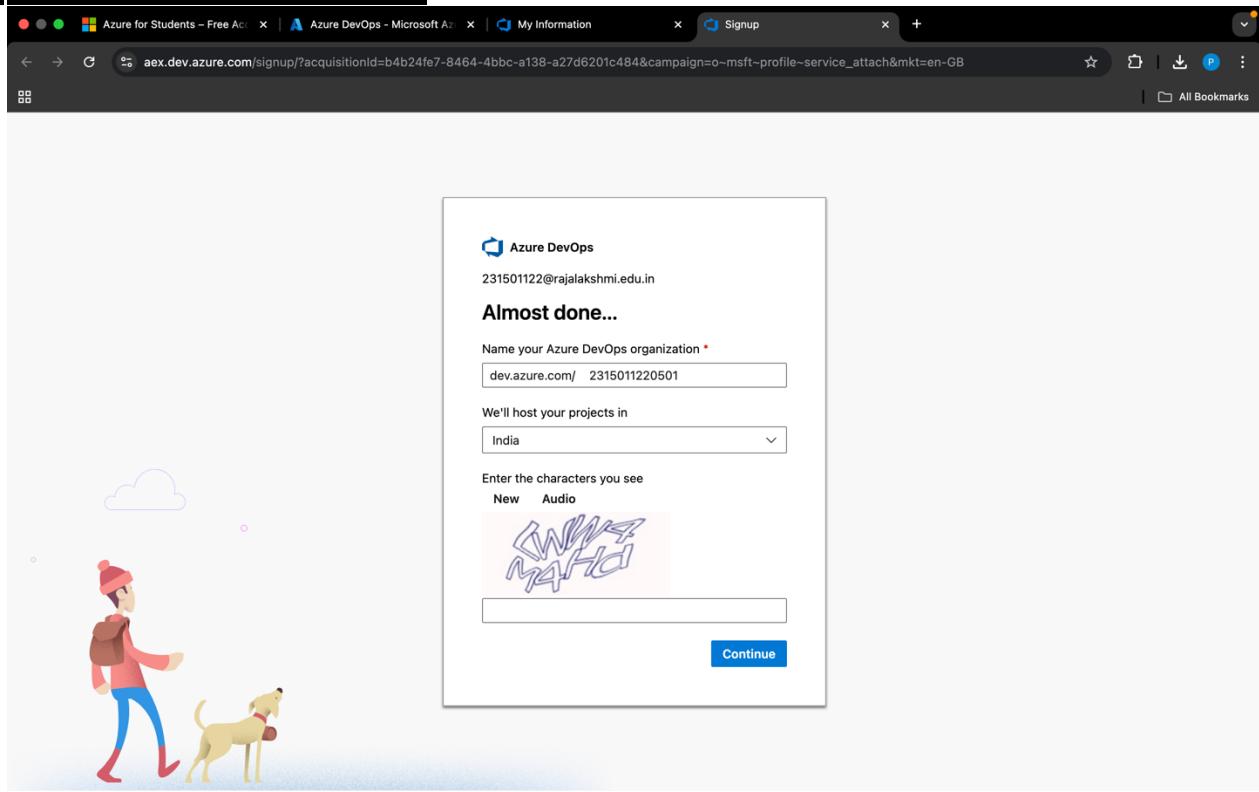
AZURE DEVOPS PROJECT SETUP AND USER STORY MANAGEMENT

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.

The screenshot shows the Azure DevOps Project dashboard for project 231501122. The dashboard includes a sidebar with organization links, a top navigation bar with tabs like Projects, My work items, and My pull requests, and a search bar. The main area displays four cards: Ecommerce, Assignment, ATM, and WORKITEMS. The Ecommerce card is expanded, showing its description and five small icons. The other cards are collapsed.

231501122

Projects My work items My pull requests

Ecommerce

The E-commerce Product Uploader is a web-based application designed to help store owners efficiently manage bulk product uploads using CSV or Excel files. The system allows uploading essenti...

A Assignment

A ATM

E Ecommerce

WORKITEMS

4. Project dashboard

The screenshot shows the Azure DevOps interface for the 'Ecommerce' project. The left sidebar includes links for Overview, Summary, Dashboards, Wiki, Boards, Repos, Pipelines, Test Plans, and Artifacts. The main content area displays the 'About this project' section, which describes the E-commerce Product Uploader application. It highlights features like bulk product uploads using CSV or Excel files, full CRUD operations, product search, and paginated views. The application is integrated into Azure DevOps for development, testing, and deployment. The right side features a 'Project stats' card showing metrics for work items, pull requests, commits, pipelines, and members.

About this project

The E-commerce Product Uploader is a web-based application designed to help store owners efficiently manage bulk product uploads using CSV or Excel files. The system allows uploading essential product details such as name, description, price, and images, with built-in validation rules to ensure data accuracy, image integrity, and compliance with ethical standards.

The project incorporates robust backend and frontend components, supporting full CRUD operations, product search, and paginated views for enhanced usability. Images are locally verified to prevent duplication across products and ensure only appropriate content is uploaded—excluding external links or unethical visuals.

The application is integrated into Azure DevOps for streamlined development, testing, and deployment. Within Azure, detailed test plans, test suites, and test cases have been created to verify system behavior under both normal (happy path) and error (negative) conditions. This enables automated and manual test execution, bug tracking, and status monitoring for high-quality delivery.

This project serves as a complete example of how to manage a full-stack software system lifecycle using Azure DevOps, emphasizing collaborative development, test-driven delivery, and continuous quality assurance.

Project stats Period: Last 7 days

Boards

- Work items created: 42
- Work items completed: 0

Repos

- Pull requests opened: 0
- Commits by 1 authors: 11

Pipelines

- Builds succeeded: 0%

Members 6

- SS
- PR
- RC
- PK
- PG
- SS

5. To manage user stories:

- 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.

The screenshot shows the Azure DevOps interface for the Ecommerce project. The left sidebar is collapsed, showing the main navigation bar at the top with tabs for 'Azure for Students', 'Azure DevOps - Micro...', 'My Information', 'Work items - Boards', and 'DevOps Solutions | Micro...'. Below the navigation bar, the URL is dev.azure.com/231501122/Ecommerce/_workitems/recentlyupdated/. The main content area is titled 'Work items' and shows a list of recently updated work items. The list includes columns for ID, Title, Assigned To, State, Area Path, and Tag. The first item in the list is highlighted.

ID	Title	Assigned To	State	Area Path	Tag:
64	Try Upload Without Entering Product Name or Price	Preethi Gopinath	Design	Ecommerce	
90	login failed	Unassigned	New	Ecommerce	
89	login failed	Unassigned	New	Ecommerce	
23	As a store owner, I want to upload a CSV/Excel file so that I can add...	Preethi Gopinath	Resolved	Ecommerce	
78	As a store owner, I want to manually add or edit a product in case I...	Preethi Gopinath	New	Ecommerce	
26	As a developer, I want to validate the file structure to ensure all req...	Preethi Gopinath	New	Ecommerce	
88	As a user i want to upload multiple csv files at once so I can analyz...	Preethi Gopinath	New	Ecommerce	
87	file upload and selection	Unassigned	New	Ecommerce	
86	data visualisation	Unassigned	New	Ecommerce	
85	test case (2)	Prathisha R	Design	Ecommerce	
82	upload and selection	Unassigned	New	Ecommerce	
80	As a developer, I want to validate the file structure to ensure all req...	Rino Calvin	Resolved	Ecommerce	

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.

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

EXP NO: 3

Date :

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

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 left sidebar includes 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 displays a backlog of work items under the Ecommerce Team. The backlog is organized by epic, feature, user story, and task. Epics include 'data visualisation' and 'product management'. Features include 'Search and Pagination' and 'Create and Update Products'. User stories and tasks provide more detail for each feature. To the right, there's a 'Planning' section showing sprints: sprint3 (5/20/2025 - 5/26/2025), sprint4 (5/27/2025 - 6/2/2025), and sprint1 (6/3/2025 - 6/9/2025). A 'New Sprint' button is also present.

2. Fill in Epics

The screenshot shows the Azure DevOps interface for the Ecommerce project. On the left, the navigation bar includes 'Overview', 'Boards', 'Work items', 'Backlogs' (which is selected), 'Sprints', 'Queries', 'Delivery Plans', 'Analytics views', 'Repos', 'Pipelines', 'Test Plans', and 'Artifacts'. The main area displays the 'Backlog' tab under the 'Ecommerce Team' board. A search bar at the top right has 'data visualization' typed into it. Below the search bar are buttons for '+ New Work Item', 'View as Board', and 'Column Options'. The backlog table lists work items with columns for Order, Work item Ty..., Title, State, and Effort. The first epic is titled 'data visualization' and contains two user stories: 'Search and Pagination' and 'Search Functionality'. The second epic is titled 'product management' and contains three user stories: 'As a store owner, I want to search products by name...', 'As a user, I want to filter products by price or descr...', and 'Update query logic in backend accordingly'. To the right of the backlog, there's a 'Planning' section with a drag-and-drop interface for work items and a list of sprints: 'sprint3' (5/20/2025 - 5/26/2025, planned effort 0 days), 'sprint4' (5/27/2025 - 6/2/2025, no work scheduled yet), and 'sprint1' (6/3/2025 - 6/9/2025, planned effort 0 days). A '+ New Sprint' button is also present.

3. Fill in Features

The screenshot shows the 'New User Story' dialog for the Ecommerce project. The dialog has a 'file upload' input field with 'file upload|' typed into it. It includes sections for 'Description' (with a placeholder 'Click to add Description.'), 'Acceptance Criteria' (with a placeholder 'Click to add Acceptance Criteria.'), 'Discussion' (with a placeholder 'Add a comment. Use # to link a work item, @ to mention a person, or ! to link a pull request.'), and 'Classification' (with 'Value area' set to 'Business'). On the right side, there are tabs for 'Planning' (Priority: 2, Risk), 'Deployment' (with a note about tracking releases and deployment status reporting), 'Development' (with a note about linking to Azure Repos commits), and 'Related Work' (with an 'Add link' button). At the bottom, there are buttons for 'Save and Close' and 'Details'.

4. Fill in User Stories

The screenshot shows the Azure DevOps interface for creating a User Story. The URL in the browser is https://dev.azure.com/231501122/Ecommerce/_backlogs/backlog/Ecommerce%20Team/Epics?workitem=88. The page title is "USER STORY 88". The user story description is: "88 As a user i want to upload multiple csv files at once so I can analyze them together". The author is Preethi Gopinath, with 0 comments and 0 tags. The state is New, area is Ecommerce, reason is New, and iteration is Ecommerce\sprint3. The "Details" section includes fields for Story Points (empty), Priority (2), and Risk (empty). The "Deployment" section notes that releases are tracked via the Releases tab. The "Development" section includes an "Add link" button for linking to Azure Repos. The "Related Work" section has an "Add link" button. The sidebar on the left lists various project components like Boards, Backlogs, and Artifacts. The bottom navigation bar shows "Project settings", "User Story", and "As a store owner, I want to manually add or edit a p...".

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. The left sidebar navigation includes 'Overview', 'Boards', 'Work items', 'Backlogs', 'Sprints' (selected), 'Queries', 'Delivery Plans', and 'Analytics views'. The top navigation bar shows the URL as dev.azure.com/231501122/Ecommerce/_sprints/taskboard/Ecommerce%20Team/Ecommerce/sprint%201. The main area displays the Taskboard for 'sprint 1' with filter 'Person: All'. The board has columns for '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.

At the bottom of the Taskboard, there are buttons for '+ New Work Item' and 'Column Options'.

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 ID/Title.

Sprint	Person	Work Item ID	Title
Sprint2	All	23	As a store owner, I want to upload a CSV/Excel file so that I can add multiple products at once.
Sprint2	All	24	Implement file upload UI and backend endpoint.
Sprint2	All	25	Parse CSV/Excel file and extract product fields.
Sprint2	All	26	As a developer, I want to validate the file structure to ensure all required columns are present.
Sprint2	All	27	Check headers for required fields (name, description, price, image).
Sprint2	All	28	Return user-friendly error messages on

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 ID/Title.

Sprint	Person	Work Item ID	Title
sprint3	All	80	As a developer, I want to validate the file structure to ensure all required columns are present.
sprint3	All	77	As a store owner, I want to search products by name so that I can find specific items quickly.
sprint3	All	88	As a user I want to upload multiple csv files at once so I can analyze them together

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 work item titled "USER STORY 78". The story is described as: "As a store owner, I want to manually add or edit a product in case I miss it during bulk upload." It is assigned to "sai senthil" and has 0 comments and 0 tags. The status is "Resolved" (selected), reason is "New", and iteration is "Ecommerce\Iteration 1". The description is "The iteration within which this user story will be implemented [Field Name: Iteration Path]". The story has 2 priority points and is in the "Planning" phase. The classification is "Business". The development section includes a note about linking to Azure Repos commits, pull requests, or branches. The deployment section notes that deployment status reporting is available for Boards in the pipeline's Options menu.

Result:

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

EXP NO: 6

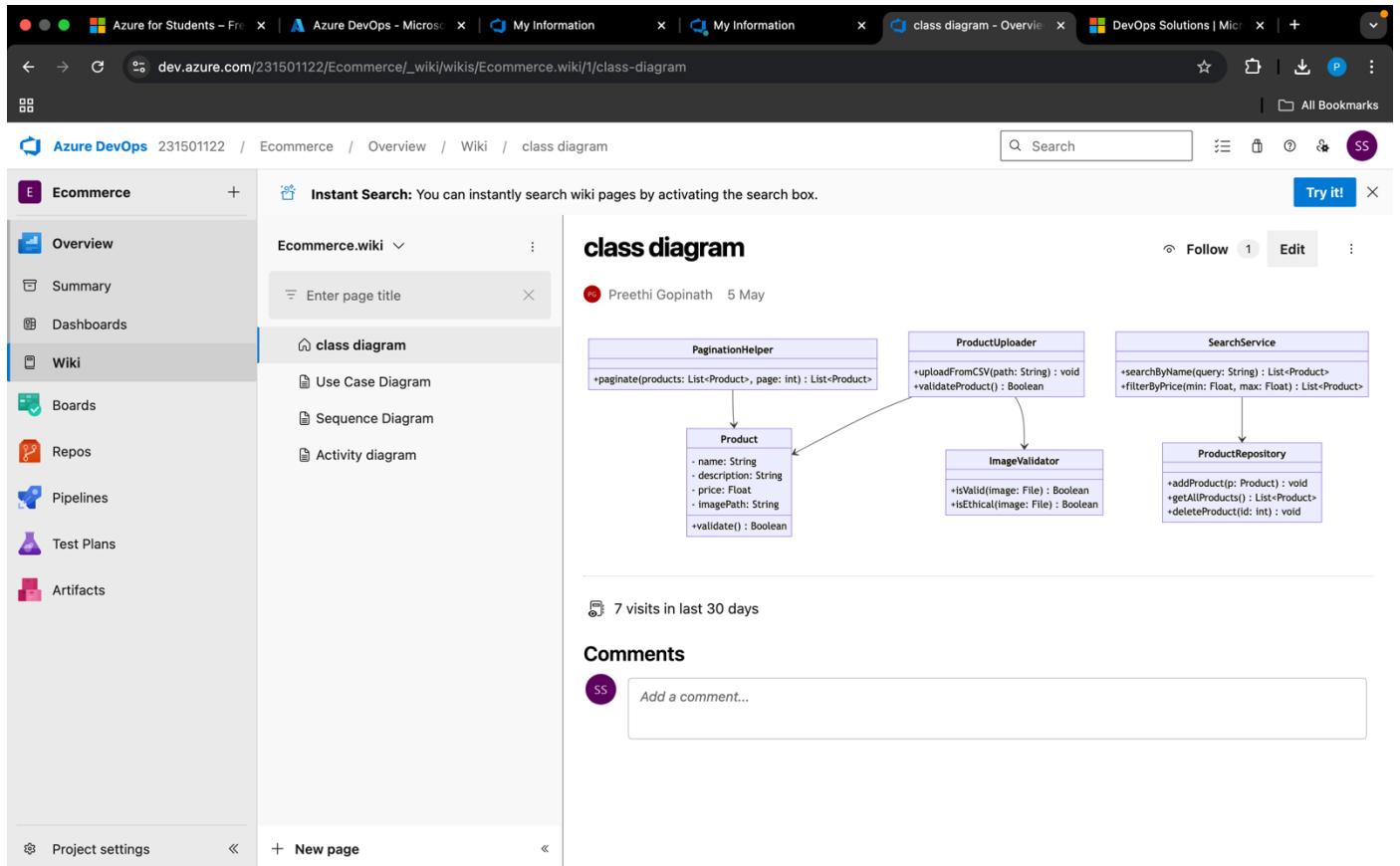
DESIGNING CLASS DIAGRAM AND SEQUENCE DIAGRAM

Date :

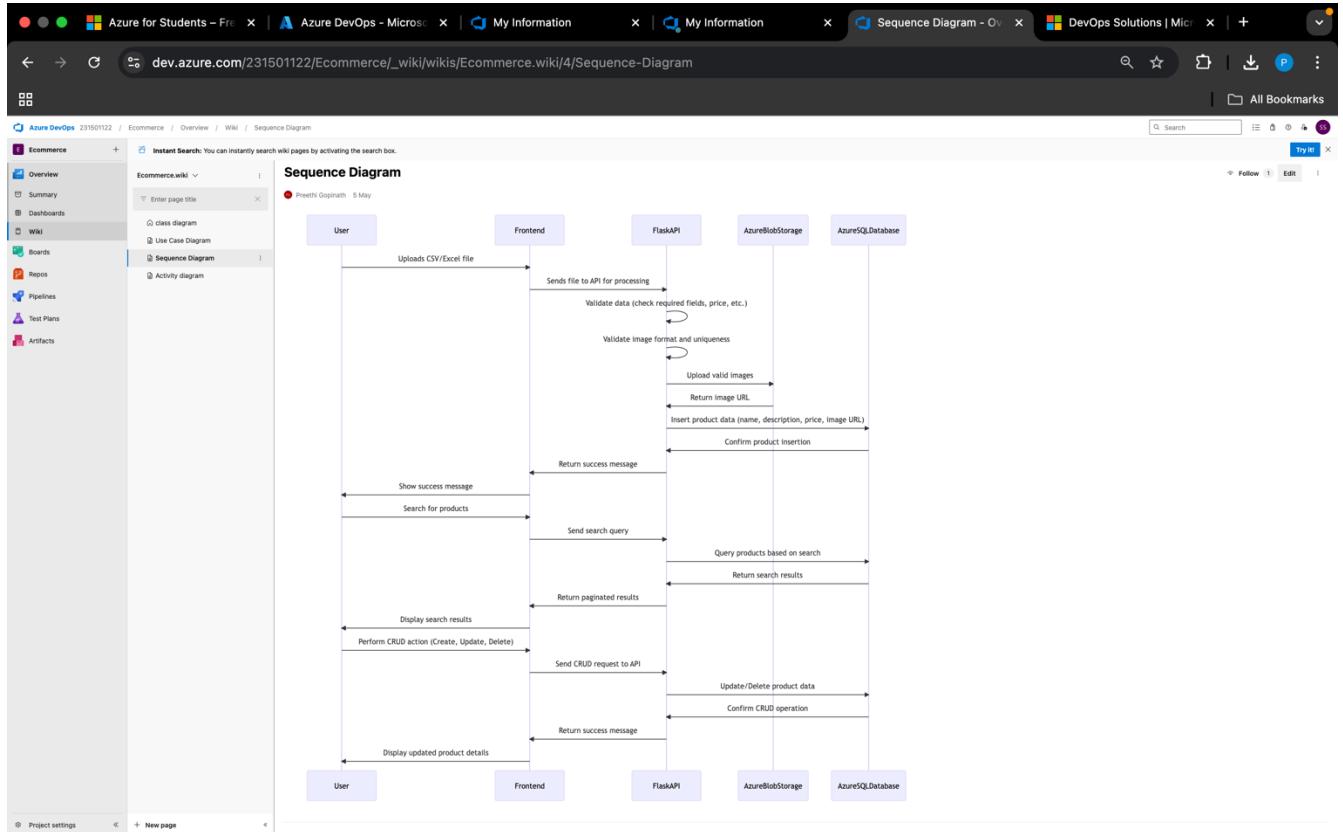
Aim:

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

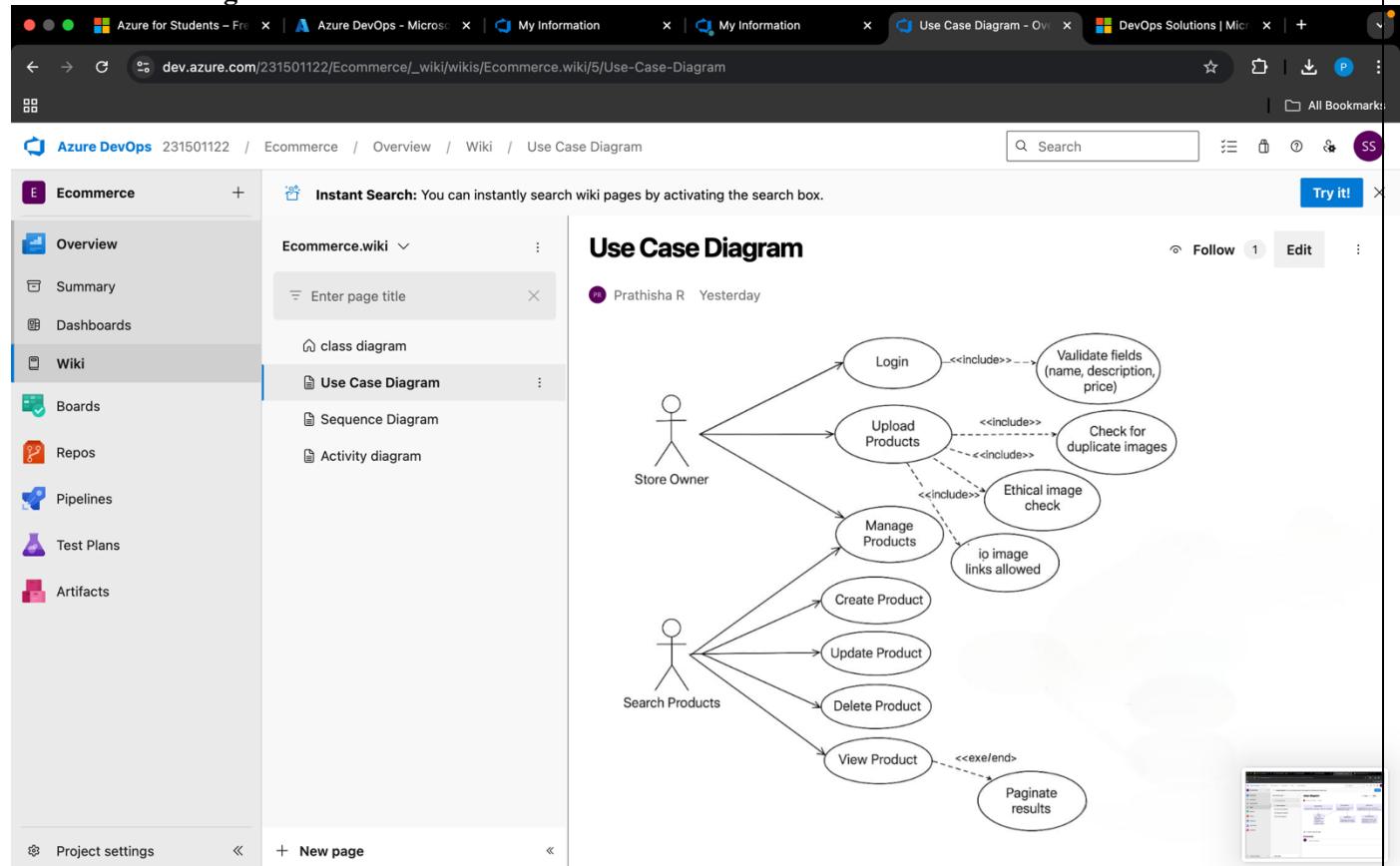
DESIGNING USE CASE DIAGRAM AND ACTIVITY DIAGRAM

Date :

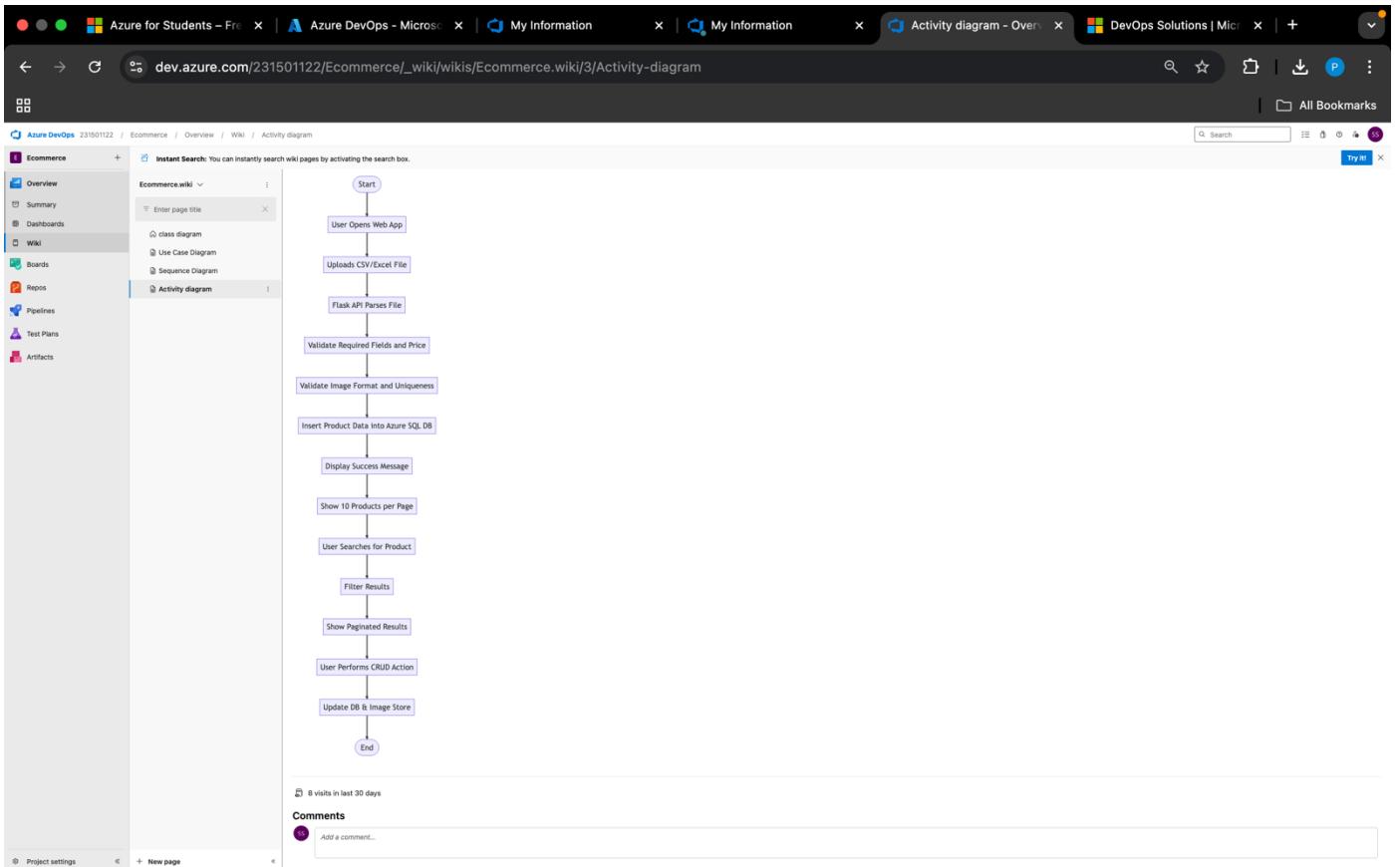
Aim:

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.

EXP NO: 8	TESTING – TEST PLANS AND TEST CASES
Date :	

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

New Test Plan

Name: performance tests

Area Path: Ecommerce

Iteration: Ecommerce\sprint3

20/05/2025 - 26/05/2025

Create Cancel

2. Test suite

Functional tests ... Past May 13 - May 19 0% run. View report

Test Suites

bulk upload(CSV) (ID: 53)

Define Execute Chart

Test Cases (1 item)

	Title	Order	Test Case Id	Assign
<input type="checkbox"/>	Upload Multiple Products Using Valid CSV File	1	54	Preethi

Upload Single Product (1)

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

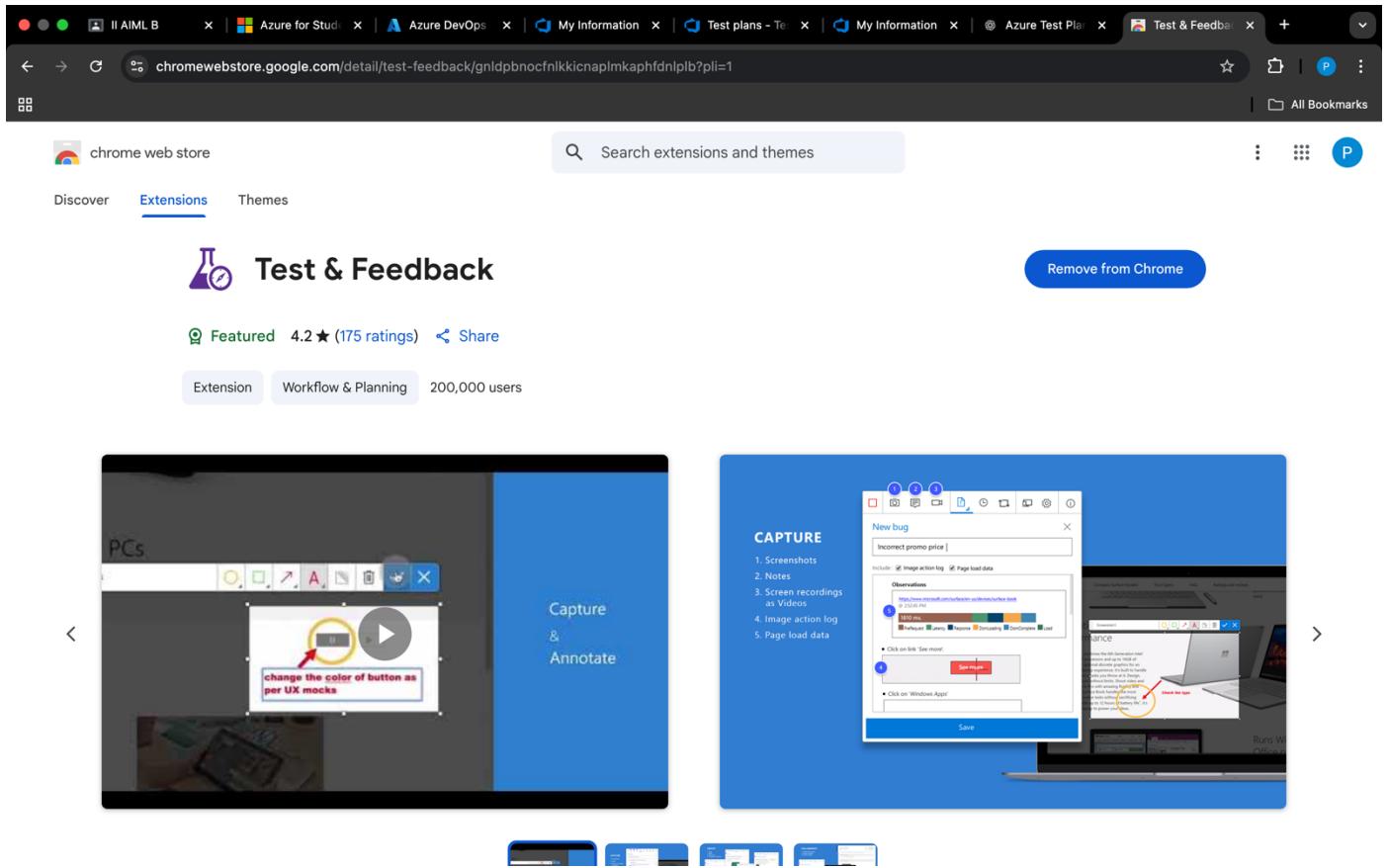
The screenshot shows the Azure DevOps interface for a project named "Ecommerce". The left sidebar is open, showing various project management and development tools like Boards, Repos, Pipelines, and Test Plans. Under "Test Plans", the "Test plans" section is selected. In the main content area, a "UI/UX 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 lists one entry:

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

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, an "Input Validation Tests" plan is displayed. 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 lists 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

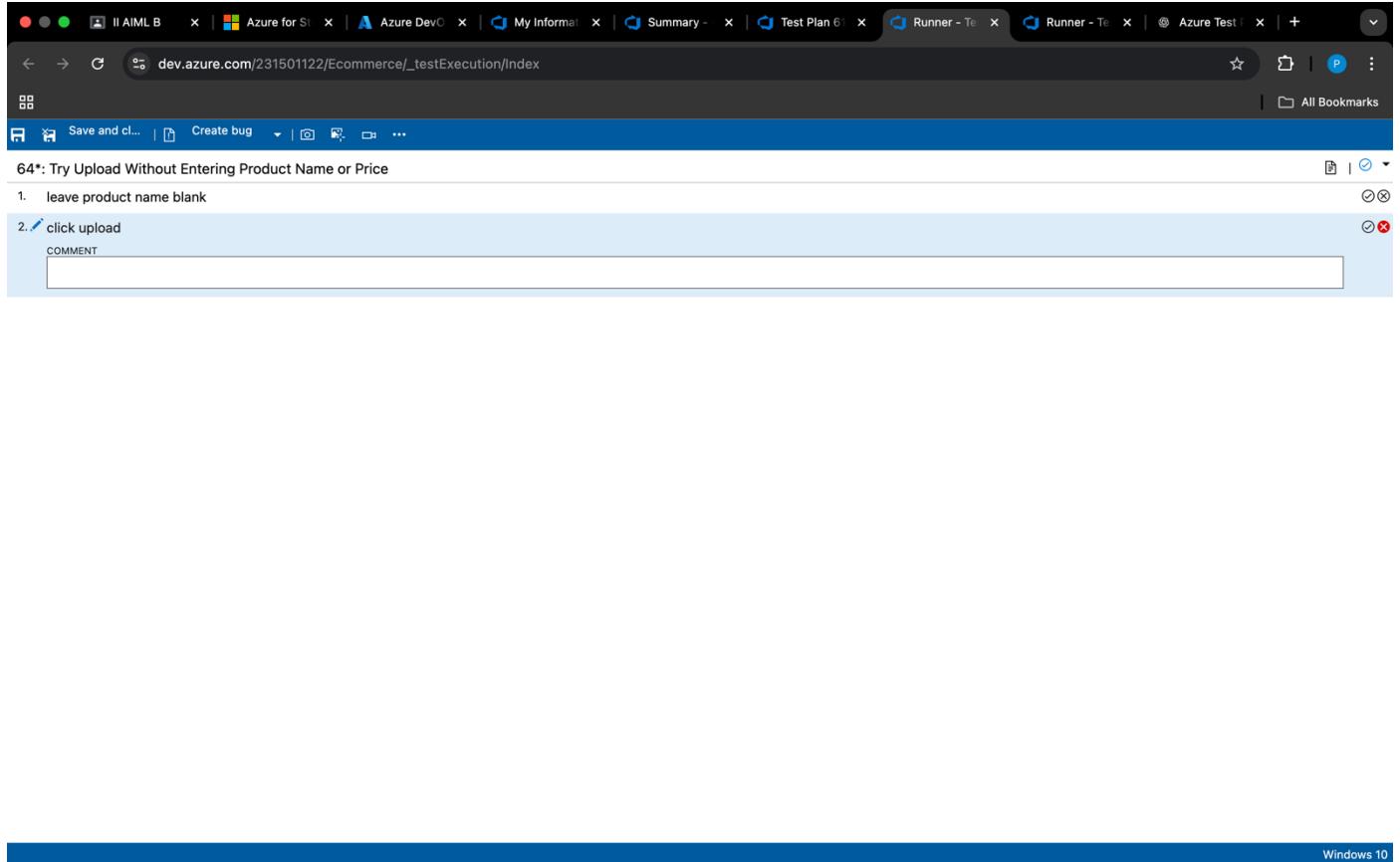
Showing it as an extension

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.

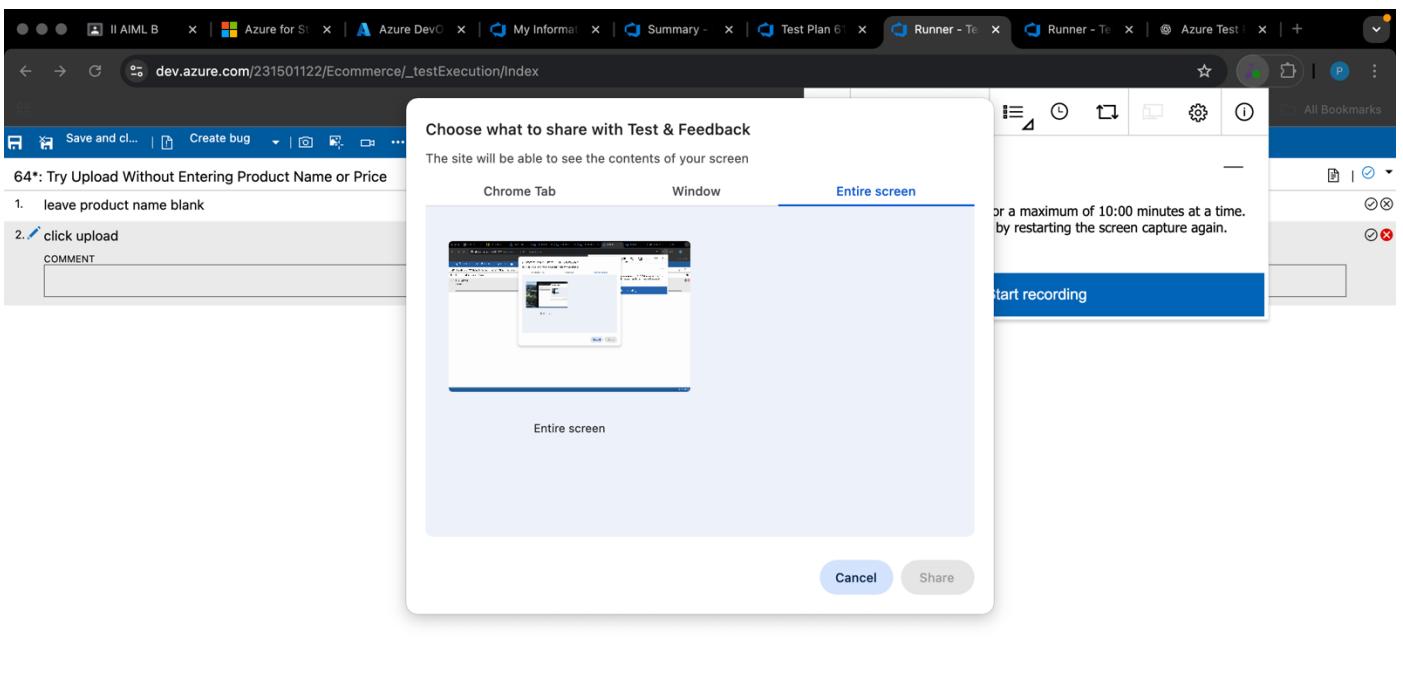
5. Running the test cases

The screenshot shows the Azure DevOps Test Plans interface. On the left, there's a sidebar with project navigation. In the center, a 'Database Sync Check (ID: 71)' test plan is displayed under the 'Execute' tab. A context menu is open over a test point named 'Check Product Entry in DB After Upload'. The menu includes options like 'View execution history', 'Mark Outcome' (with 'Passed' selected), 'Run' (with 'Run for web application' selected), 'Reset test to active', 'Edit test case', 'Assign tester', and 'View test result'.

This screenshot is similar to the one above, but it includes an 'Extensions' overlay on the right side. The overlay lists 'Test & Feedback' as having 'Full access' and provides a link to 'Manage Extensions'. The main interface shows the same 'Database Sync Check (ID: 71)' test plan and its details.

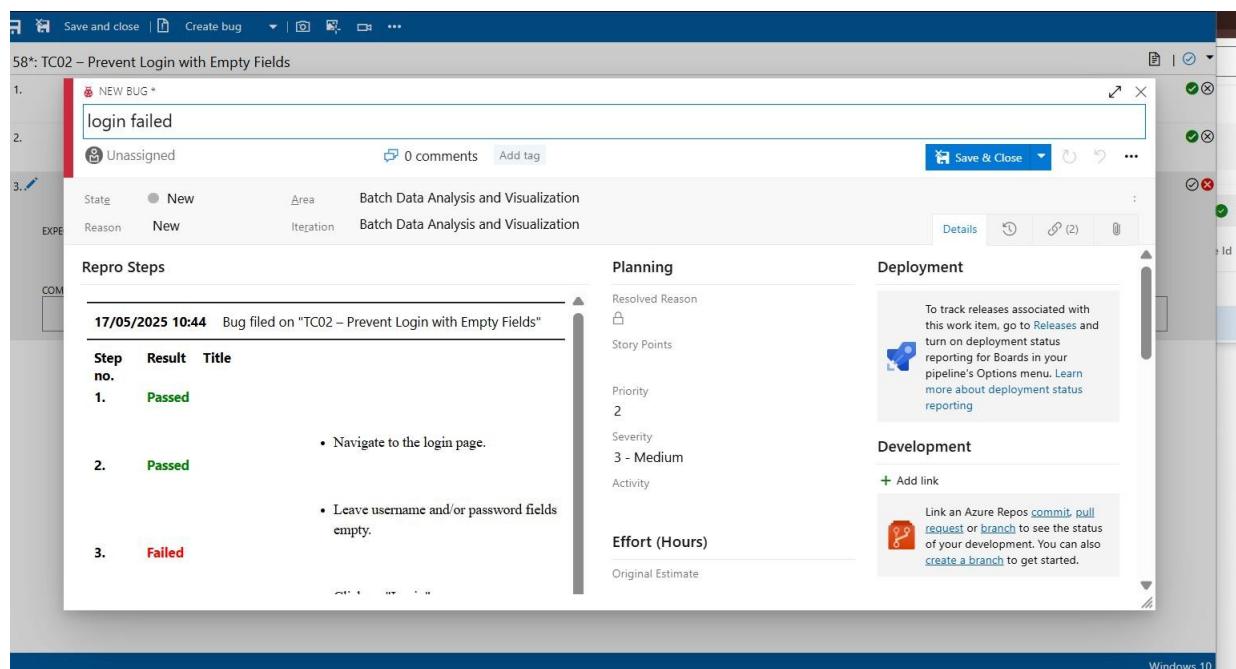


6. Recording the test case



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 plan named 'batch data analysis' with a status bar indicating 'May 17 - May 24' and '50% run, 50% passed'. Below this, under 'Test Suites', two suites are listed: 'TS01- User Authentication (2)' and 'TS02- Logout Functionality (2)'. The 'TS01- User Authentication' suite is expanded, showing two test points: 'TC01 – Successful Login' (Passed) and 'TC02 – Prevent Login with Empty Fields' (Not Run). A modal window titled 'TC01 – Successful Login' is open, showing the 'Test Case Results' table:

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 the modal, a link reads '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" for step 3 is "Validation error message is shown prompting user to fill required fields". The "Planning" section shows a priority of 2 and a severity of 3 - Medium. The "Deployment" section includes a note about tracking releases and a link to Azure Repos. The "Development" section provides instructions for linking commits or pull requests. The "Related Work" section allows for adding links to existing work items.

- Assigning bug to the developer and changing state

This 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 listed as "New" and "Assigned" to "Shri Dharshini". The rest of the page content, including the repro steps, planning details, deployment information, development links, and related work sections, remains identical to the previous screenshot.

10. Progress report

The screenshot shows the Azure DevOps interface for a project named "Ecommerce". The left sidebar contains navigation links for Overview, Boards, Repos, Pipelines, Test Plans, Test plans, Progress report (which is selected), Parameters, Configurations, Runs, and Artifacts. Below the sidebar is a "Project settings" link. The main content area is titled "Progress report" and includes a summary of test plans and points, a circular progress bar for runs, and a chart showing the outcome trend over the last 14 days. The chart indicates 11% completion with 1 run and 1 passed test point.

Summary

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

Outcome trend

Last 14 Days

The chart displays the number of tests run per day. The legend indicates that grey areas represent "Not run" and green areas represent "Passed". The data shows a significant increase in test runs starting around May 18th, reaching approximately 8 tests by May 21st, with all tests passing.

Date	Not run	Passed
2025-05-08	0	0
2025-05-09	0	0
2025-05-10	0	0
2025-05-11	0	0
2025-05-12	0	0
2025-05-13	0	0
2025-05-14	0	0
2025-05-15	0	0
2025-05-16	0	0
2025-05-17	0	0
2025-05-18	0	0
2025-05-19	0	0
2025-05-20	0	0
2025-05-21	0	8

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 available process templates. The 'Agile (default)' template is selected, indicated by a blue border around its name. Other visible templates include 'Basic', 'Agile plus', 'BatchDataAnalysis', 'Scrum', and 'CMMI'. Each template has a brief description and a 'Team projects' count.

12. View the new test case template

The screenshot shows the 'Add a field to Test Case' dialog box. It's part of the 'Agile' process template configuration. The dialog has three tabs: '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. In the background, the Azure DevOps interface shows the 'All processes > Agile' path and some other settings like 'Steps' and 'Layout'.

The screenshot shows the Azure DevOps Organization Settings page for a workspace named '231501153'. The left sidebar is titled 'Organization Settings' and includes sections for General, Security, Boards, Pipelines, and Process. The 'Process' section is currently selected. The main content area displays the 'All processes > BATCH DATA ANALYSIS > Test Case' configuration. It features a 'Layout' tab selected, showing tabs for 'Steps', 'Summary', and 'Associated Aut...'. Below these are sections for 'Recent test results', 'Deployment', 'Development', 'Related Work', and 'Status'. Each section contains a 'Text (multiple lines)' input field. A search bar at the top right and a navigation bar with icons for Home, Dashboards, Pipelines, and Boards are also visible.

Azure DevOps 231501153 / Settings / Process

Organization Sett...
231501153

Search Settings

General

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

Security

- Security overview
- Policies
- Permissions

Boards

Process

Pipelines

Agent pools

Settings

System processes cannot be customized. To add customization create an inherited process.

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	

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	CI/CD PIPELINES IN AZURE
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, 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 was triggered by Prathisha R and is associated with the Ecommerce repository and commit 93ebaf40. The run started at 7:00 PM today and completed successfully in 20 seconds. A summary table shows one job named 'Job' with a success status and a duration of 12 seconds. A note indicates that the run will be cleaned up after 1 month based on project settings.

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

The screenshot shows a GitHub repository page for 'Software-construction-project'. The repository is public and has one branch, 'main'. There are two commits: one by 'senthilsgit22' renaming 'uploader.html.html' to 'uploader.html' and another by 'abbe0eb' renaming 'login.html.html' to 'login.html'. A 'README' file is present. The repository has no releases, packages, or forks. It has 1 watcher and 0 stars.

Code / Software-construction-project

Issues **Pull requests** **Actions** **Projects** **Wiki** **Security** **Insights** **Settings**

Software-construction-project **Public**

main **Code** **About**

senthilsgit22 Rename uploader.html.html to uploader.html

login.html Update and rename login.html.html to ...

uploader.html Rename uploader.html.html to upload...

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

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.