



**RAJALAKSHMI**  
**ENGINEERING COLLEGE**  
An AUTONOMOUS Institution  
Affiliated to ANNA UNIVERSITY, Chennai

**DEPARTMENT OF COMPUTER SCIENCE AND  
ENGINEERING LAB MANUAL**

**CS23432 – Software Construction**

**(REGULATION 2023)**

**RAJALAKSHMI ENGINEERING COLLEGE**  
**Thandalam, Chennai-602015**

Name:Janani V

Register No: 231801064

Year / Branch / Section: 2<sup>nd</sup> / AI&DS / FA

Semester: IV

Academic Year: 2024 - 2025

# INDEX

S.No.	Date	Title
1.	21/1/25	Azure Devops Environment Setup.
2.	21/1/25	Azure Devops Project Setup and User Story Management.
3.	28/1/25	Setting Up Epics, Features, And User Stories for Project Planning.
4.	11/2/25	Sprint Planning.
5.	18/2/25	Poker Estimation.
6.	25/2/25	Designing Class and Sequence Diagrams for Project Architecture.
7.	04/3/25	Designing Architectural and ER Diagrams for Project Structure.
8.	25/3/25	Testing – Test Plans and Test Cases.
9.	15/4/25	Load Testing and Pipelines.
10.	22/4/25	GitHub: Project Structure & Naming Conventions.

**EXP NO: 1**

## **AZURE DEVOPS ENVIRONMENT SETUP**

### **Aim:**

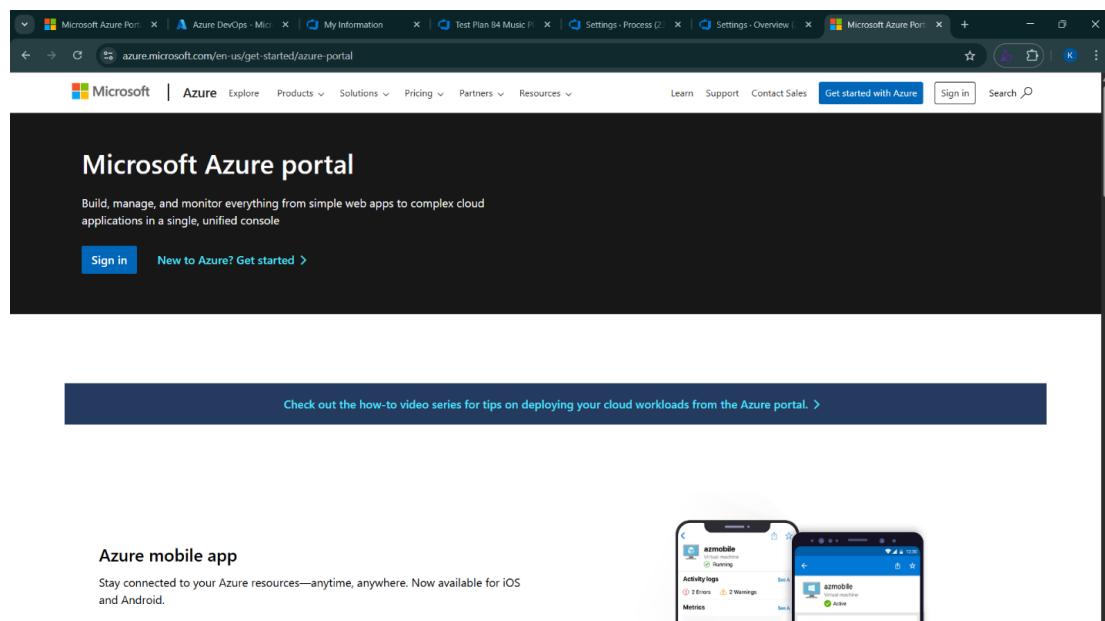
To set up and access the Azure DevOps environment by creating an organization through the Azure portal.

### **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>



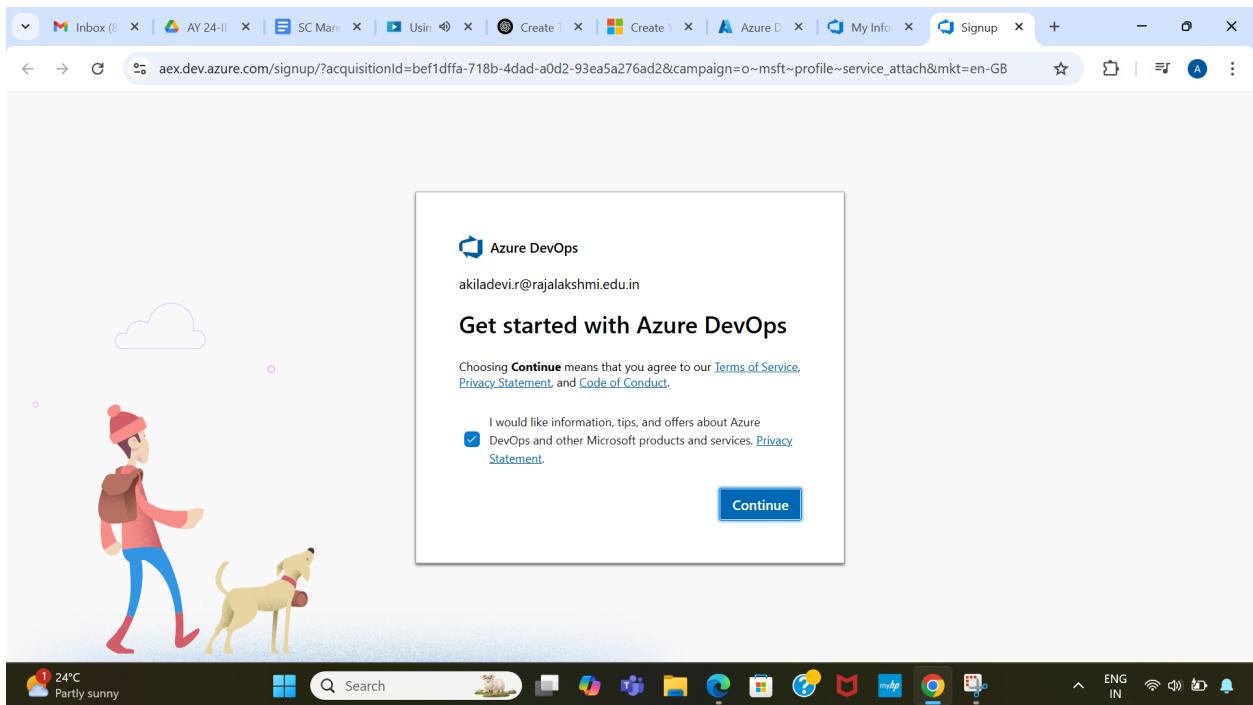
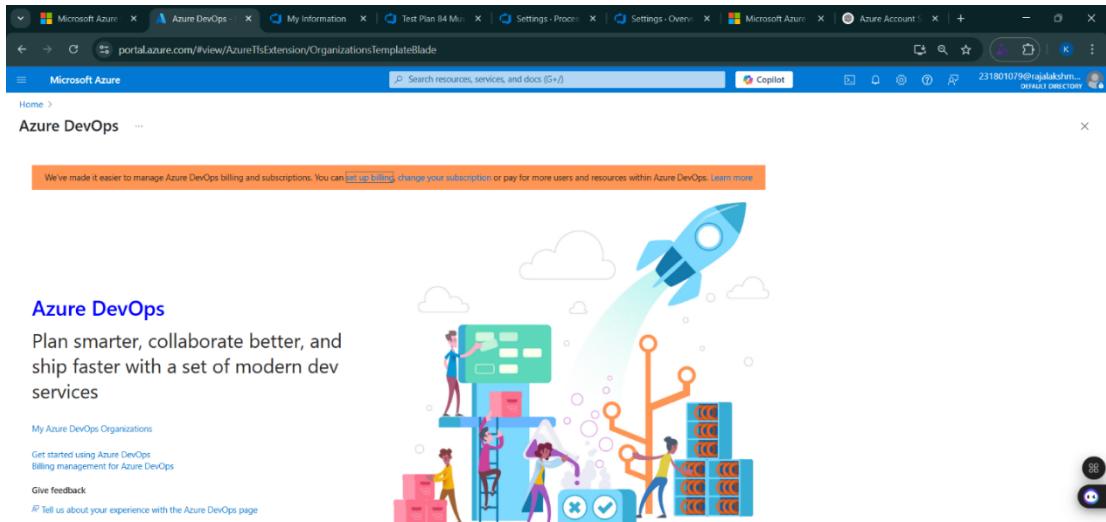
## 2. Azure home page

The screenshot shows the Microsoft Azure home page. At the top, there's a navigation bar with tabs like Microsoft Azure, Azure DevOps, My Information, Test Plan 84 M... (highlighted), Settings - Process, Settings - Overview, Microsoft Azure, and Azure Account. Below the navigation bar is a search bar with placeholder text "Search resources, services, and docs (S+)" and a Copilot button. The main content area is titled "Azure services" and includes a "Create a resource" button and icons for Azure DevOps organizations, Subscriptions, Dashboard hub, Resource groups, Azure Load Testing, Quickstart Center, Azure AI services, Kubernetes services, and More services. Below this is a "Resources" section with a "Recent" tab showing two items: "Music" (Azure Load Testing) and "Music\_playlist\_Batch\_Creator" (Resource group). A "See all" link is also present. To the right of the resources is a vertical sidebar with a profile picture and the email address "231801079@rajalakshmi...". The sidebar also includes sections for "Navigate" (Subscriptions, Resource groups, All resources, Dashboard), "Tools" (Microsoft Learn, Azure Monitor, Microsoft Defender for Cloud, Cost Management), and "Useful links" (Azure mobile app).

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

This screenshot is similar to the previous one but with a search bar at the top containing the text "DevOps". The search results are displayed below, under the "Services" and "Marketplace" tabs. Under "Services", items include Azure Native New Relic Service, Managed DevOps Pools, Azure DevOps organizations, and Azure Native Dynatrace Service. Under "Marketplace", items include Static Web App, Rocky Linux 9, Build Agents for Azure DevOps, and InfluxDB Cloud (Official Version). The rest of the interface is identical to the first screenshot, including the sidebar with "My Azure DevOps Organization" highlighted.

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.

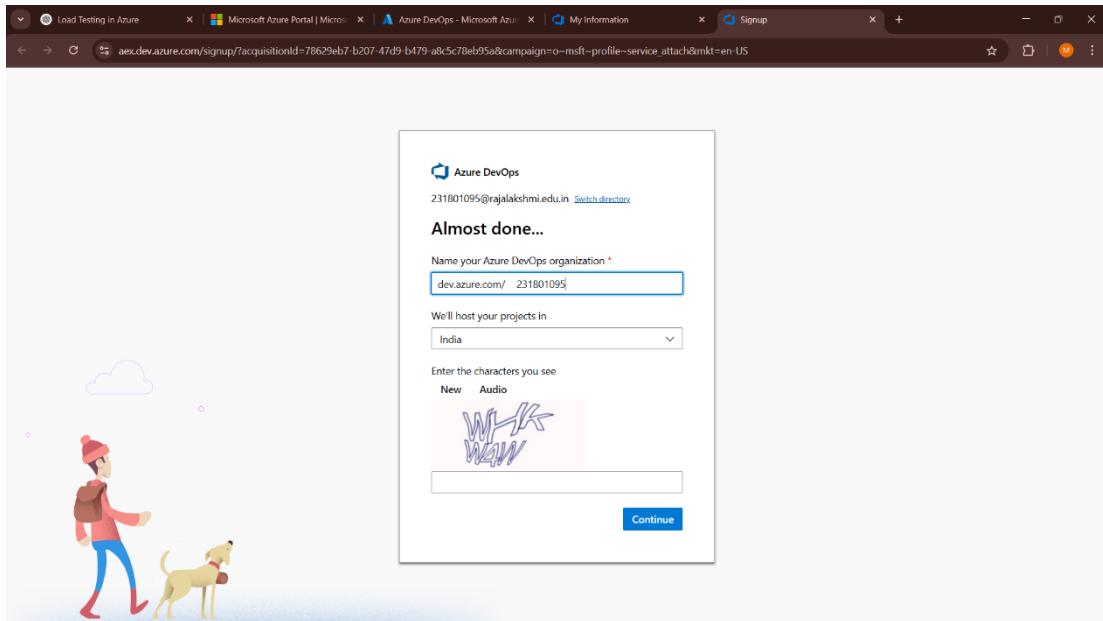
**EXP NO: 2**

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

### **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

Project name \*

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 [?](#)

Work item process [?](#)

3. 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 Organizations dashboard. At the top right, it displays the user's name "Krithika.M.A" and a "Sign out" link. Below the header, there's a large circular profile picture with a purple "K" on it. To the right of the profile picture, the title "Azure DevOps Organizations" is shown, along with a "Create new organization" button. Under the "Projects" section, there is one project listed: "E-commerce product uploader". To the right of the project, there are "Actions" like "Open in Visual Studio". Below the projects, there's a section for "New project". Further down, another user "dev.azure.com/231801065 (Member)" is listed. On the left side of the dashboard, there's a sidebar with sections for "Visual Studio Dev Essentials" (with a "Use your benefits" link) and "Krithika.M.A" (including email "krithika7604@gmail.com" and location "India"). The bottom of the screen shows the Windows taskbar with various pinned icons.

#### 4. Project dashboard

The screenshot shows the "E-commerce product uploader" project dashboard. The top navigation bar includes the project name "E-commerce product uploa...", a "Search" bar, and a "Private" button. The left sidebar has a "Summary" tab selected, along with other options like "Overview", "Dashboards", "Wiki", "Boards", "Repos", "Pipelines", "Test Plans", and "Artifacts". The main content area features a welcome message "Welcome to the project!" and a "Project stats" section which states "No stats are available at this moment. Setup a service to see project activity." Below the stats, there's a "Members" section showing 1 member. The bottom of the screen shows the Windows taskbar.

## 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 managing work items. The left sidebar has 'Work items' selected. The main area displays a table of work items with columns for ID, Title, Assigned To, State, and Area Path. A tooltip for 'User Story' is visible over the 'Title' column. The table includes rows for various items like 'Bug', 'Epic', 'Feature', 'Issue', and 'Task'. The 'Assigned To' column shows names like 'madhusha', 'Janani Vasu', 'Kritika.MA', 'Janani Rajan', and 'Mahalakshmi231801093'. The 'Area Path' column shows paths like 'E-commerce product uploader'.

The screenshot shows the Microsoft sign-in page. It features a large purple circular profile picture with a white letter 'K'. The user's name 'Krithika.M.A' is displayed in bold, along with their email address 'krithika7604@gmail.com'. Below this, there are links for 'My Microsoft account' and 'Switch directory'. The Microsoft logo is at the top left, and a 'Sign out' link is at the top right. The bottom of the page has a 'Sign in with a different account' link.

**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**

### **Aim:**

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

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

The screenshot shows the Azure DevOps Boards Backlog view for the 'E-commerce product uploader' team. On the left, the navigation menu is visible with 'Backlogs' selected. The main area displays a backlog board with columns for Order, Work Item Type, Title, State, Effort, Business Area, and Tags. A hierarchical structure is shown under an 'Epic' item:

- Epic: User Authentication and Authorization (State: New, Business)
- Feature: Login page for users (State: New, Business)
- User Story: As a user, I want to log in using my email and password... (State: New, Business)
  - Task: Design the Login Page UI (State: New)
  - Task: Add Validation for Input Fields. (State: New)
  - Task: Handle Incorrect Login Attempts. (State: New)
- User Story: As a user, I want to reset my password if I forget it, so ... (State: New, Business)
- User Story: As a user, I want to authenticate using my social medi... (State: New, Business)
- Feature: Login Page for Product Uploaders (State: New, Business)

### **1.Fill in Epics**

The screenshot shows the Azure DevOps Work items screen for creating a new Epic. The navigation menu on the left has 'Work items' selected. The main area is titled 'NEW EPIC' and contains fields for 'Description' and 'Discussion'. The 'Description' field has the placeholder 'Click to add Description.' The 'Discussion' field has a note: 'Add a comment. Use # to link a work item, @ to mention a person, or ! to link a pull request.' The 'Planning' section includes fields for 'Priority' (set to 2), 'Risk', and 'Effort'. The 'Deployment' section includes a note: '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.' The 'Development' section includes a note: '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.'

## 2.Fill in Features

The screenshot shows the 'Work Items' screen in Azure DevOps for a project titled 'E-commerce product uploader'. A new feature work item is being created, titled 'Login page for users'. The 'Description' section is empty. In the 'Planning' section, the state is 'New', area is 'E-commerce product uploader', reason is 'New', priority is '2', and risk is 'Low'. The 'Deployment' section includes instructions for tracking releases and deployment status reporting. The 'Development' section includes links for adding a branch and viewing commit status. The browser taskbar at the bottom shows various open tabs and system icons.

## 3.Fill in User Story Details

The screenshot shows the 'Work Items' screen in Azure DevOps for a project titled 'LMS'. A new user story work item is being created, titled 'Login'. The 'Description' section contains the text: 'As a user I can login using regno and password so I can access my account.' The 'Acceptance Criteria' section is empty. In the 'Planning' section, the state is 'New', area is 'LMS', reason is 'New', story points are '5', priority is '2', and risk is '2 - Medium'. The 'Classification' section indicates the value area is 'Business'. The 'Deployment' section includes instructions for tracking releases and deployment status reporting. The 'Development' section includes links for adding a branch and viewing commit status. The browser taskbar at the bottom shows various open tabs and system icons.

**Result:**

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

**EXP NO: 4**

# **SPRINT PLANNING**

## **Aim:**

To assign user story to specific sprint for the E Commerce Product UploaderProject.

## **Sprint Planning**

The screenshot shows the Azure DevOps Boards interface. On the left, there's a sidebar with options like Overview, Boards, Work items, Backlogs, Sprints, Queries, Delivery Plans, Pipelines, and Artifacts. The 'Sprints' option is selected. In the center, under the 'E-commerce product uploader Team' section, the 'Backlog' tab is active. At the top, it says 'Did you notice Azure Boards has a new look and awesome new features? Learn more.' Below that, it shows 'E-commerce product uploader Team' and 'Taskboard Backlog Analytics'. A 'Sprint 1' dropdown is open, showing a table of tasks. The table has columns for Order, Title, State, Assigned To, and Remaining. There are five tasks listed:

Order	Title	State	Assigned To	Remaining
1	As a user, I want to authenticate using my social media account so that I can regain access to my account.	New	Janani Rajan	
2	As a user, I want to reset my password if I forget it, so that I can regain access to my account.	New	Mahalakshmi23...	
3	As a user, I want to log in using my email and password so that I can regain access to my account.	New	Krithika.M.A	
4	As a product uploader, I want to log in using my credentials so that I can upload products.	New	231801064@raj...	
5	As a product uploader, I want to authenticate using a role-based authentication system so that I can upload products.	New	231801091@raj...	

At the bottom right, it says 'February 17 - February 28 10 work days remaining'.

## **Sprint 1**

The screenshot shows the Azure DevOps Boards interface, similar to the previous one but with the 'Taskboard' tab active. The left sidebar is identical. The 'Backlog' tab is now inactive, indicated by a grey background. The 'Taskboard' tab is active, showing the 'Sprint 1' tasks. The tasks are grouped by state: Active, Resolved, and Closed. Task 5 is selected and highlighted with a blue background. The task details are shown in a callout box:

As a user, I want to authenticate using my social media account so that I can regain access to my account.  
Mahalakshmi2318010...

The 'Active' column contains tasks 58, 59, and 60. The 'Resolved' column contains task 58. The 'Closed' column is empty.

## Sprint 2

The screenshot shows the Azure DevOps Taskboard for the 'E-commerce product uploader Team'. The left sidebar is open, showing 'Sprints' selected. The main area displays a taskboard for 'Sprint 2' from March 3 to March 14, which spans 10 work days. The board has four columns: New, Active, Resolved, and Closed. There are three tasks visible:

- 12** As a product uploader, I want to enter product details (name, description, price) so that I can list the product on the website.
  - New
  - Unassigned
- 13** As a product uploader, I want to upload product images so that customers can view the product visually.
  - New
  - Unassigned
- 14** As a product uploader, I want to assign categories and tags to the product so that it is listed correctly.
  - New
  - Unassigned

## Sprint 3

The screenshot shows the Azure DevOps Taskboard for the 'E-commerce product uploader Team'. The left sidebar is open, showing 'Sprints' selected. The main area displays a taskboard for 'Sprint 3' from March 17 to March 28, which spans 10 work days. The board has four columns: New, Active, Resolved, and Closed. There are three tasks visible:

- 22** As an admin, I want to review all newly uploaded products before they are displayed on the website so that only quality products are listed.
  - New
  - Unassigned
- 23** As an admin, I want to provide feedback on rejected products to the uploader, so they can make necessary changes.
  - New
  - Unassigned
- 25** As a product uploader, I want to receive notifications when my product is approved or rejected.
  - New
  - Unassigned

**Result:**

The Sprints are created for the E-Commerce Product Uploader Project

**EXP NO: 5**

## **POKER ESTIMATION**

### **Aim:**

Create Poker Estimation for the user stories - E Commerce Product Uploader Project.

### **Poker Estimation**

As a user, I want to reset my password if I forget it, so that I can regain access to my account.

Mahalakshmi231801093    0 Comments    Add Tag

New	Area	E-commerce product uploader
New	Iteration	E-commerce product uploader\Sprint 1

Description

• **Password Reset Request:** The user enters their email or username to request a password reset.  
• **Verification Process:** A verification link or code is sent to confirm the user's identity.  
• **Password Update:** The user sets a new password after identity verification.  
• **Confirmation and Login:** The user is notified and can log in with the new password.

Planning

Story Points: 5  
Priority: 2  
Risk:

ance Criteria

1. **Password Reset Request:**  
User can request a reset by entering their email/username.

2. **Verification Link/Code:**  
A verification link/code is sent and expires within a set time.(eg: within 15 mins)

3. **New Password Set:**

Classification

Value area: Business

### **Result:**

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

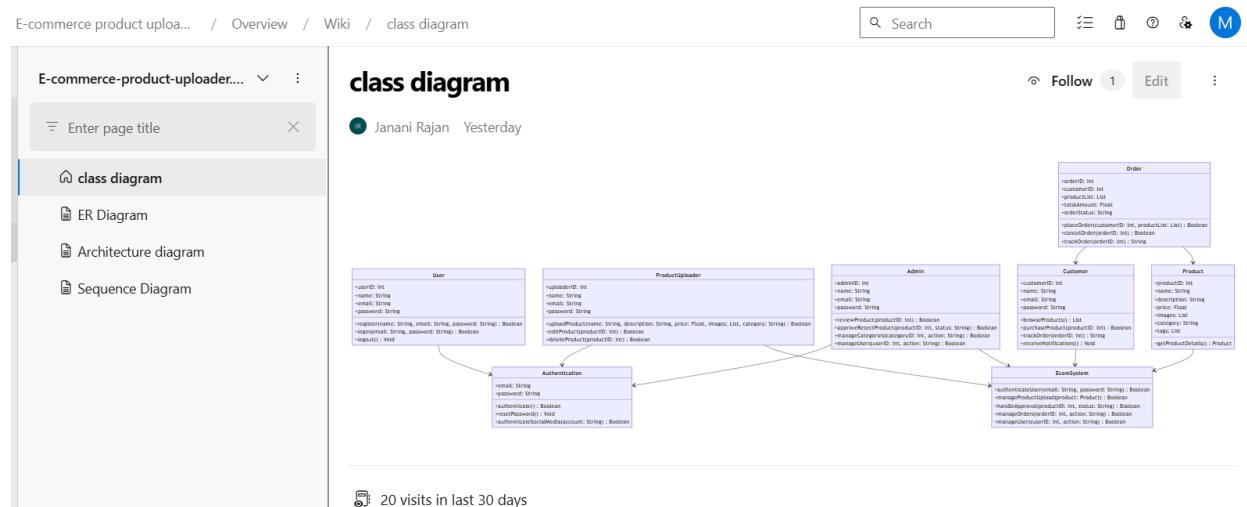
**EXP NO: 6**

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

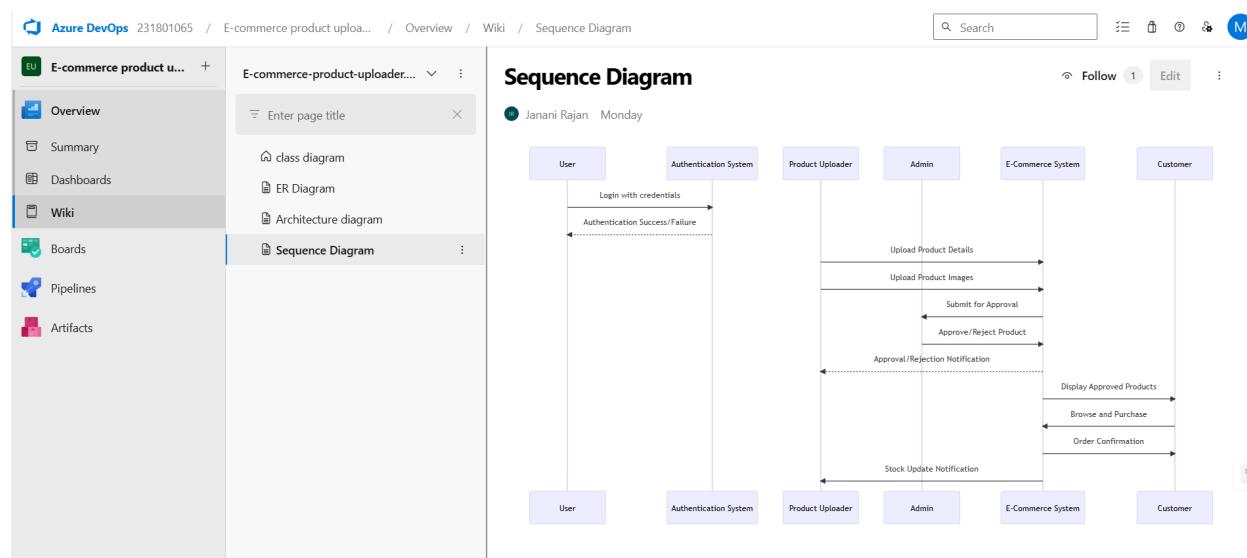
### **Aim:**

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 E-Commerce Product Uploader.

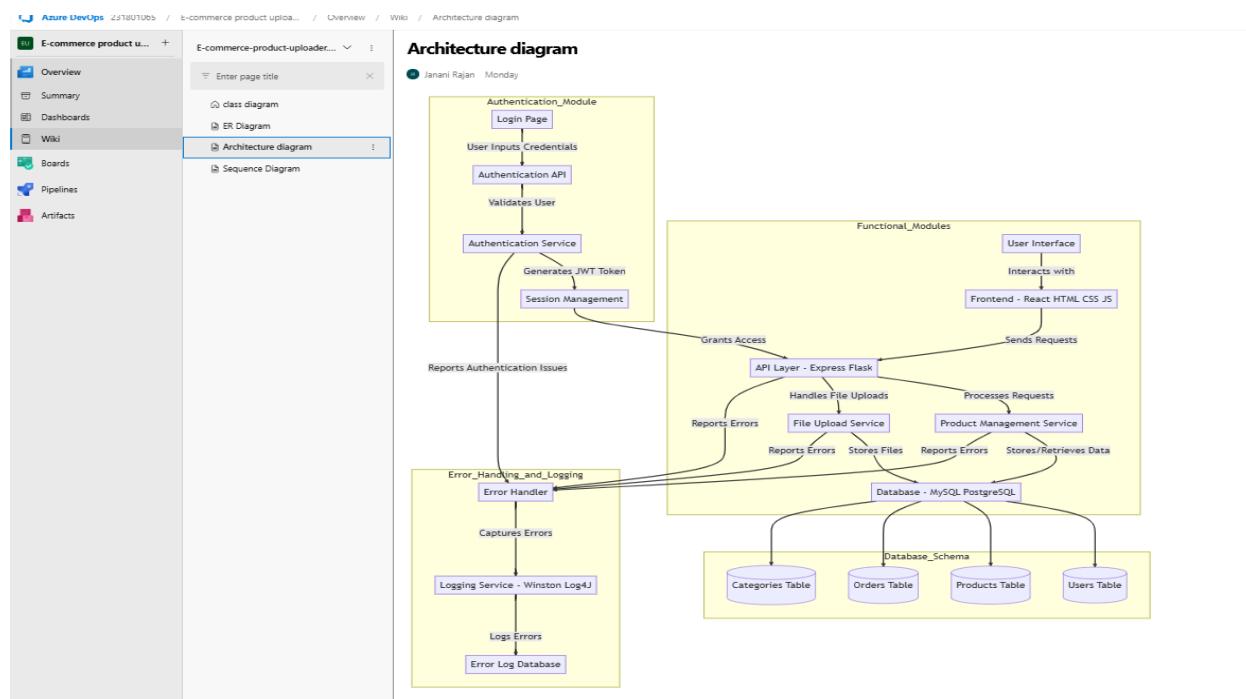
**EXP NO: 7**

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

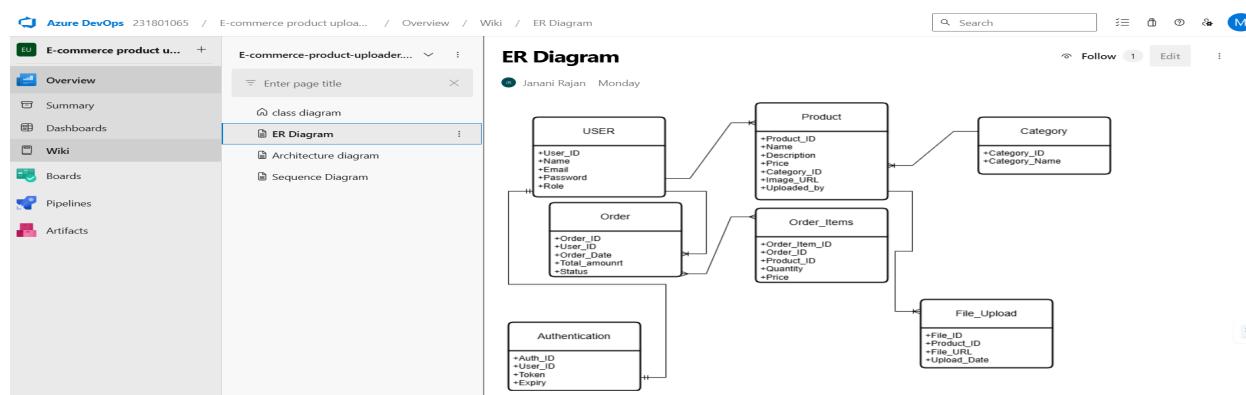
## **Aim:**

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 the E-Commerce Product Uploader.

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

- User Signup & Login
- Viewing and Managing Playlists
- Fetching Real-time Metadata
- Editing playlists (rename, reorder, record)
- Creating smart audio playlists based on categories (mood, genre, artist, etc.)

**2. Define User Interactions**

- Each test case simulates a real user behaviour (e.g., logging in, renaming a playlist, adding a song).

**3. Design Happy Path Test Cases**

- Focused on validating that all features function as expected under normal conditions.
- Example: User logs in successfully, adds item to playlist, or creates a category-based playlist.

**4. Design Error Path Test Cases**

- Simulate negative or unexpected scenarios to test robustness and error handling.
- Example: Login fails with invalid credentials, save fails when offline, no recommendations found.

**5. 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.

**6. Use Clear Naming and IDs**

- Test cases are named clearly (e.g., TC01 – Successful Login, TC10 – Save Playlist Fails).
- Helps in quick identification and linking to user stories or features.

**7. Separate Test Suites**

- Grouped test cases based on functionality (e.g., Login, Playlist Editing, Recommendation System).

- Improves organization and test execution flow in Azure DevOps.

## 8. Prioritize and Review

- Critical user actions are marked high-priority.
- Reviewed for completeness and traceability against feature requirements.

### 1. New test plan

New Test Plan

Name \*: E-commerce product uploader - Test plan1

Area Path \*: E-commerce product uploader

Iteration \*: E-commerce product uploader\Sprint 1

Create Cancel

### 2. Test suite

ProjectTestSuite (ID: 91)

Test Points (6 items)

Title	Outcome	Order	Test Case Id
Check product categories are displayed in the left	Passed	1	92
Check product categories are displayed in the left	Passed	1	92
Static suite	Passed	2	93
Requirement based suite	Passed	2	93
Query based suite	Failed	3	99
Validate Login with parameters	Active	3	99

Run for web application

### **3.Test case**

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

E Commerce Product Uploader– Test Plans

#### **USER STORIES**

- As a user, I want to sign up and log in securely so that I can access my playlists (ID: 79).
- As a user, I need to see my playlist in one place (ID: 76).
- As a user, I should be able to create an audio playlist as needed (ID: 73).
- As a user, I should be able to rename, record, and change the playlist (ID: 68).
- As a user, I need to have real-time metadata (ID: 65).

#### **Test Suites**

##### **Test Suit: TS01 - User Login (ID: 86)**

###### **1. TC01 – Successful Sign Up**

- **Action:**
  - Go to the Sign-Up page.
  - Enter valid name, email, and password.
  - Click "Sign Up".
- **Expected Results:**
  - Sign-Up form is displayed.
  - Fields accept values without error.
  - Account is created, and the user is redirected to the dashboard.
- **Type:** Happy Path

###### **2. TC02 – Secure Login**

- **Action:**
  - Go to the Login page.
  - Enter valid email and password.
  - Click on "Login".
- **Expected Results:**
  - Login form is displayed.
  - Fields accept data without error.
  - User is logged in and redirected to the dashboard.
- **Type:** Happy Path

###### **3. TC03 – Sign Up with Existing Email**

- **Action:**
  - Go to the Sign-Up page.
  - Enter a name and an already registered email.
  - Click on "Sign Up".
- **Expected Results:**

- Fields accept data.
- Error message "Email already registered" is displayed.
- **Type:** Error Path

#### 4. TC04 – Login with Wrong Password

- **Action:**
  - Go to the Login page.
  - Enter valid email and incorrect password.
  - Click on "Login".
- **Expected Results:**
  - Input is accepted.
  - Error message "Invalid username or password" is shown.
- **Type:** Error Path

### Test Suit: TS02 - View Playlists (ID: 87)

#### 1. TC05 – View Playlist Page

- **Action:**
  - Log in successfully.
  - Navigate to "My Playlists" section.
- **Expected Results:**
  - All created playlists are displayed clearly.
- **Type:** Happy Path

#### 2. TC06 – Playlist Loading Failure

- **Action:**
  - Disconnect from the internet.
  - Navigate to "My Playlists".
- **Expected Results:**
  - Network is offline.
  - Error message "Unable to load playlists" is shown.
- **Type:** Error Path

### Test Suit: TS03 - Real-Time Metadata (ID: 88)

#### 1. TC07 – Real-Time Metadata Display

- **Action:**
  - Play a song.
  - Observe the metadata panel.
- **Expected Results:**
  - Metadata (title, artist, album, duration) is displayed and updates in real time.
- **Type:** Happy Path

#### 2. TC08 – Metadata Not Updating

- **Action:**

- Play a different song.
  - Observe the metadata panel.
- **Expected Results:**
  - Metadata remains static or shows default/fallback message.
- **Type:** Error Path

#### **Test Suit: TS04 - Playlist Editing (ID: 89)**

##### **1. TC09 – Rename Playlist Successfully**

- **Action:**
  - Navigate to "My Playlists".
  - Click "Rename" next to a playlist.
  - Enter a new name and click "Save".
- **Expected Results:**
  - Playlist name updates successfully.
- **Type:** Happy Path

##### **2. TC10 – Rename with Blank Name**

- **Action:**
  - Click "Rename" on a playlist.
  - Leave the field blank.
  - Click "Save".
- **Expected Results:**
  - Error message "Playlist name cannot be empty" is shown.
- **Type:** Error Path

##### **3. TC11 – Change Playlist Order**

- **Action:**
  - Open a playlist.
  - Drag and drop songs to reorder.
  - Click "Save".
- **Expected Results:**
  - Playlist order is updated and saved.
- **Type:** Happy Path

##### **4. TC12 – Change Playlist Order Fails**

- **Action:**
  - Login and go to "My Playlists".
  - Select a playlist.
  - Go offline or simulate server error.
  - Reorder songs and click "Save Order".
- **Expected Results:**
  - Error message: "Failed to update order. Please check your connection".
- **Type:** Error Path

## Test Suit: TS05 - Smart Playlist Creation (ID: 90)

### 1. TC13 – Generate Playlist Based on Various Categories

- Action:

- Login with valid credentials.
- Click on "Generate Playlist".
- Select categories.
- Click "Generate Playlist".

- Expected Results:

- Playlist is generated based on selected mood and categories.

- Type: Happy Path

### 2. TC14 – Fail to Generate Playlist Due to Missing Category Selection or Invalid Input

- Action:

- Login with valid credentials.
- Click on "Generate Playlist".
- Select categories.
- Click "Generate Playlist".

- Expected Results:

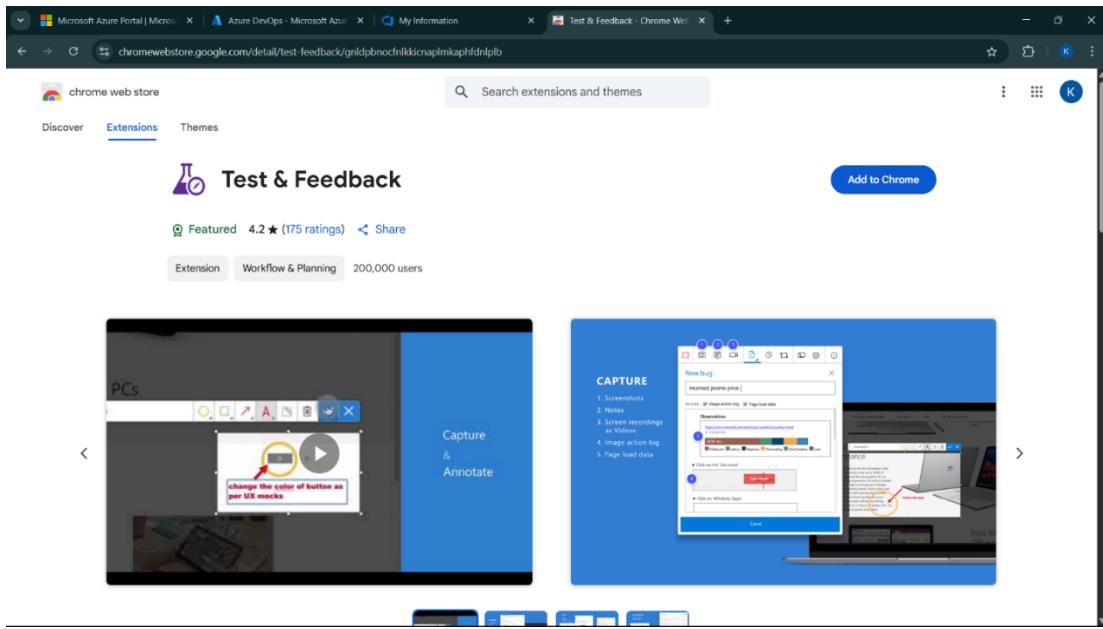
- Error message: "Please select at least one valid category" or "No recommendations found for the selected filters".

- Type: Error Path

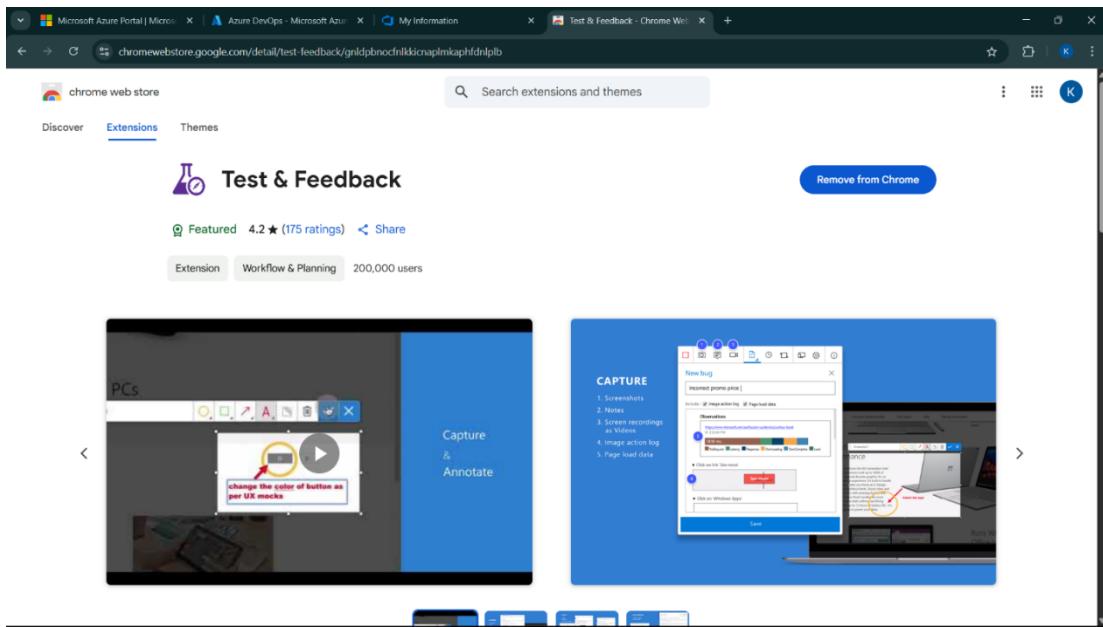
## Test Cases

The screenshot shows a Microsoft Edge browser window displaying the Azure DevOps Test Plan interface. The URL is dev.azure.com/231801065/E-commerce%20product%20uploader/\_testPlans/define?planId=70&suiteId=91. The main content area shows a test case titled 'TEST CASE 92' with the description 'Check product categories are displayed in the left'. The test case is assigned to 'Krithika.M.A' and has a status of 'Design'. It is part of the 'E-commerce product uploader' project and 'Sprint 1'. The 'Steps' section contains two steps: 'Login steps' and 'Validate all product categories are displayed on the left side of the page'. The 'Expected result' for the second step is 'Expected product categories displayed on the left side'. The 'Deployment' section provides instructions to track releases associated with this work item. The 'Development' section includes a 'Add link' button and a note about linking to Azure Repos. The 'Related Work' section allows adding links to other work items. The bottom of the screen shows the Windows taskbar with various pinned icons and the system tray indicating the date as 28-04-2025 and time as 18:07.

#### 4. Installation of test



#### Test and feedback



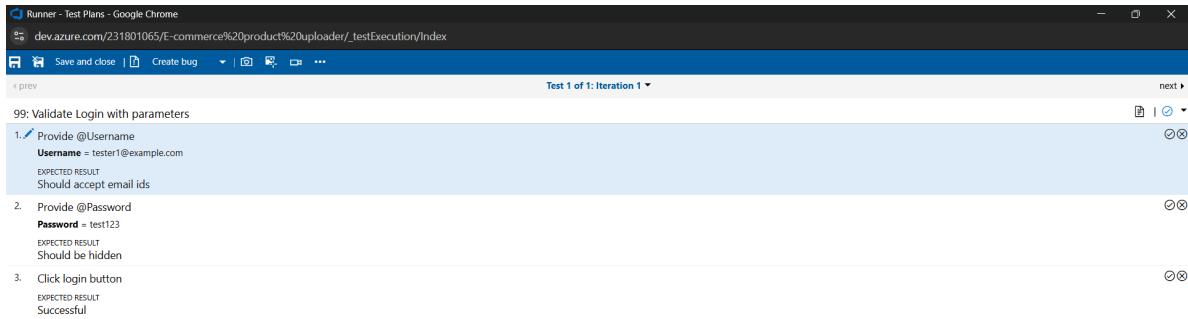
## Showing it as an extension

The screenshot shows the Azure DevOps interface with the 'Test Plan 70 Product Uploader' selected. The left sidebar is open, showing 'Test Plans' as the active section. In the center, a 'ProjectTestSuite (ID: 91)' is displayed under 'Test Suites'. On the right, a modal window titled 'Extensions' is open, listing extensions with 'Full access' on the site. One extension, 'Test & Feedback', is highlighted. Below the extensions, a table lists 'Test Cases (3 items)'.

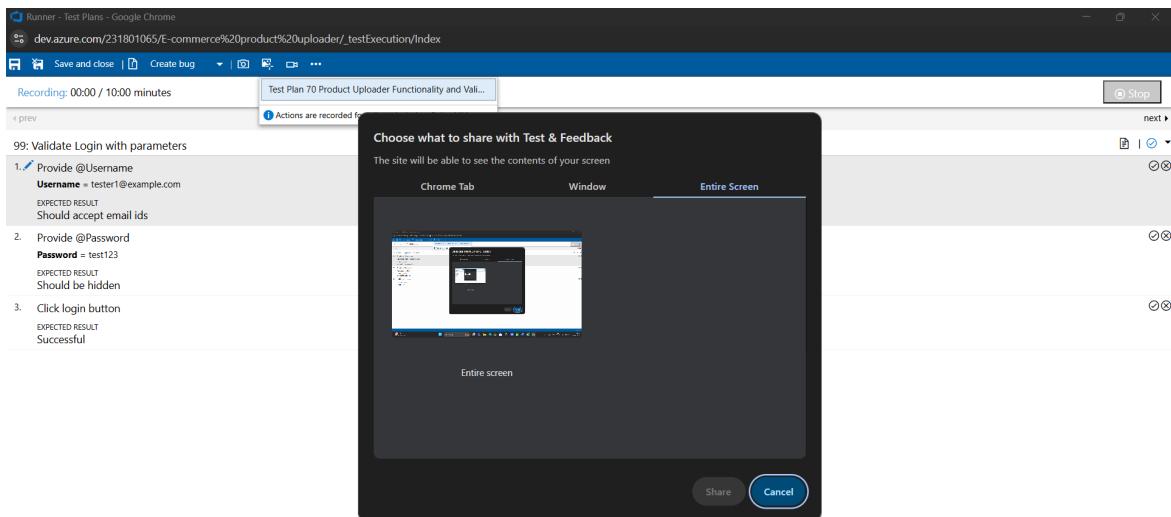
Test Case	Order	Tester
1	92	Krithika.M.A
2	93	Janani Rajan
3	99	Janani Rajan

## 5. Running the test cases

The screenshot shows the same Azure DevOps interface as before, but the 'Execute' tab is now selected in the 'ProjectTestSuite (ID: 91)' header. A context menu is open over the third test case, 'Validate Login with parameters'. The menu options include 'Run for web application', 'Run for desktop application', and 'Run with options'.



## 6. Recording the test case



## 7.Creating the bug

The screenshot shows a browser window titled "Runner - Test Plans - Google Chrome" with the URL "dev.azure.com/231801065/E-commerce%20product%20uploader/\_testExecution/index". The page displays a test plan for "99: Validate Login with parameters". The steps listed are:

1. Provide @Username  
Username = tester1@example.com  
EXPECTED RESULT  
Should accept email ids
2. Provide @Password  
Password = test123  
EXPECTED RESULT  
Should be hidden
3. Click login button  
EXPECTED RESULT  
Successful

A "Create bug" button is visible at the top right of the test plan interface.



The screenshot shows a browser window titled "Runner - Test Plans - Google Chrome" with the URL "dev.azure.com/231801065/E-commerce%20product%20uploader/\_testExecution/index". A "NEW BUG" dialog box is open over the test plan. The bug details are:

**Title:** Issue  
**State:** Unassigned  
**Area:** E-commerce product uploader  
**Iteration:** E-commerce product uploader\Sprint 1

**Repro Steps:**

4/28/2025 12:47 PM Bug filed on "Validate Login with parameters"

Step no.	Result	Title
1.	None	Provide @Username
		Expected Result Should accept email ids
2.	None	Provide @Password
		Expected Result Should be hidden
3.	None	Click login button

**Planning:**

- Resolved Reason
- Story Points

**Deployment:**

To track releases associated with this work item, go to Releases and turn on deployment status reporting. 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

**Related Work:**

+ Add link

Add an existing work item as a parent



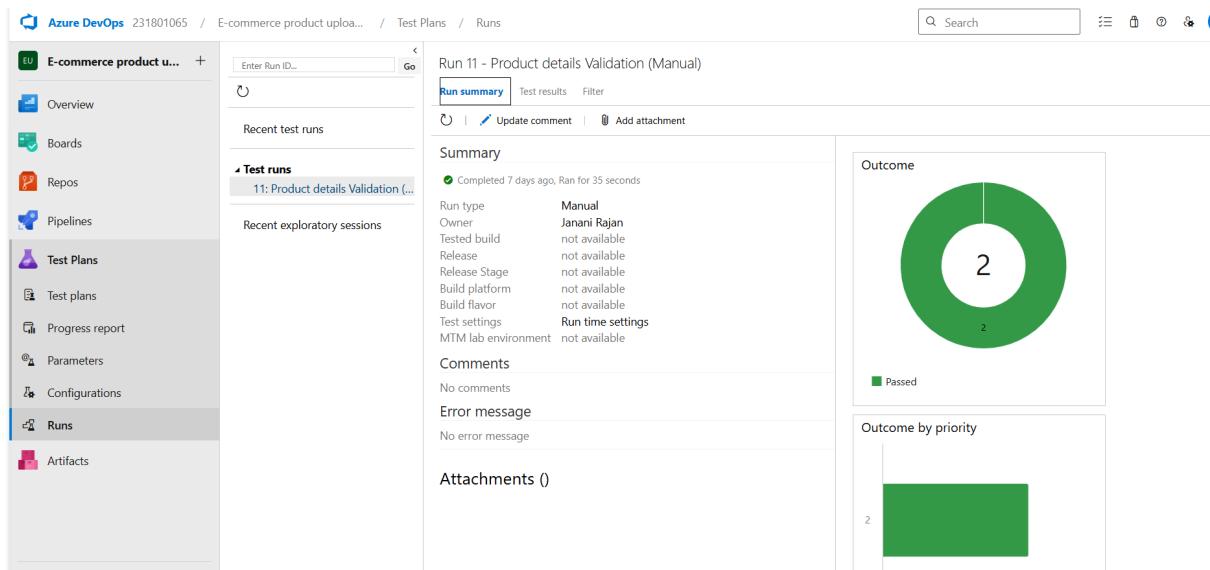
The screenshot shows the Azure DevOps interface for a project titled "E-commerce product uploader". The left sidebar is open, showing options like Overview, Boards, Repos, Pipelines, Test Plans, Test plans, Progress report, Parameters, Configurations, and Runs. The "Runs" option is selected. The main content area displays a test run titled "Run 7 - ProjectTestSuite (Manual) / Check product categories are displayed in the left". The "Summary" section shows the run was passed by Janani Rajan, tested on build 28, and used the "Product Uploader Functionality and Validation Test plan". The "Analysis" section shows the owner is Janani Rajan, failure type is None, resolution is None, and comment is not available. Below the summary, there's a section for "Attachments ()" and "Linked Items (0)". The bottom right corner shows the Windows taskbar with various pinned icons.

## 8. Test case results

This screenshot shows the same Azure DevOps interface as the previous one, but the "Execute" tab is selected in the "ProjectTestSuite (ID: 91)" dialog. The "Test Case Results" table lists six test points, all of which have passed. The table includes columns for Outcome,TimeStamp, Configuration, Run by, Tester, and Test. The outcomes are: Passed (Apr 20, Firefox, Janani Rajan, Janani Rajan, Product Uploader Functionality and Validation Test plan), Passed (Apr 20, Chrome, Janani Rajan, Janani Rajan, Product Uploader Functionality and Validation Test plan), Passed (Apr 20, Chrome, Janani Rajan, Janani Rajan, Product Uploader Functionality and Validation Test plan), Passed (Apr 20, Firefox, Janani Rajan, Janani Rajan, Product Uploader Functionality and Validation Test plan), and Passed (Apr 20, Chrome, Janani Rajan, Janani Rajan, Product Uploader Functionality and Validation Test plan). The bottom right corner shows the Windows taskbar.

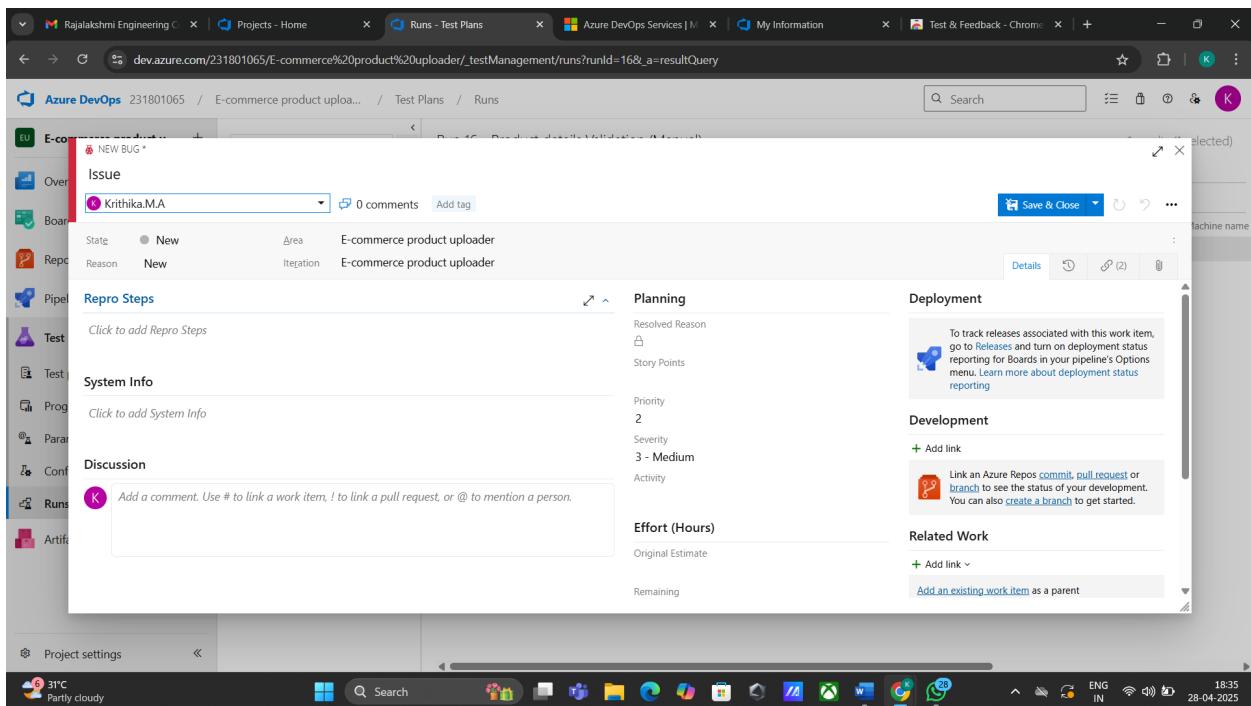
Outcome	TimeStamp	Configuration	Run by	Tester	Test
Passed	Apr 20	Firefox	Janani Rajan	Janani Rajan	Product Uploader Functionality and Validation Test plan
Passed	Apr 20	Chrome	Janani Rajan	Janani Rajan	Product Uploader Functionality and Validation Test plan
Passed	Apr 20	Chrome	Janani Rajan	Janani Rajan	Product Uploader Functionality and Validation Test plan
Passed	Apr 20	Firefox	Janani Rajan	Janani Rajan	Product Uploader Functionality and Validation Test plan
Passed	Apr 20	Chrome	Janani Rajan	Janani Rajan	Product Uploader Functionality and Validation Test plan

## 9. Test report summary



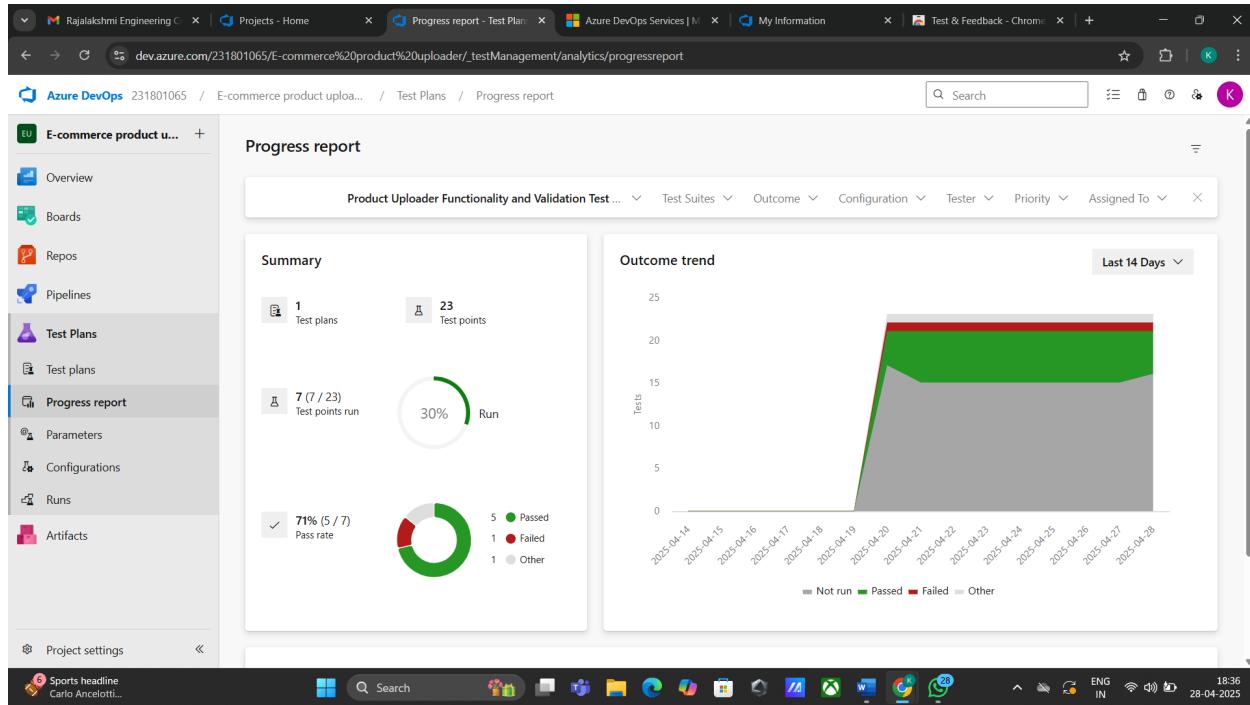
The screenshot shows the Azure DevOps interface for a test run. The left sidebar is titled 'E-commerce product u...' and includes options like Overview, Boards, Repos, Pipelines, Test Plans, Test plans, Progress report, Parameters, Configurations, Runs, and Artifacts. The 'Runs' option is selected. The main area is titled 'Run 11 - Product details Validation (Manual)'. It has tabs for Run summary, Test results, and Filter. Under 'Run summary', there's a 'Recent test runs' section with a link to '11: Product details Validation ...'. Below it is a 'Summary' section with a green checkmark indicating the run completed 7 days ago, ran for 35 seconds, and was manual. It lists various build and release details. To the right is a donut chart titled 'Outcome' showing 2 passed tests. Another chart below it shows 'Outcome by priority' with 2 green bars.

- Assigning bug to the developer and changing state

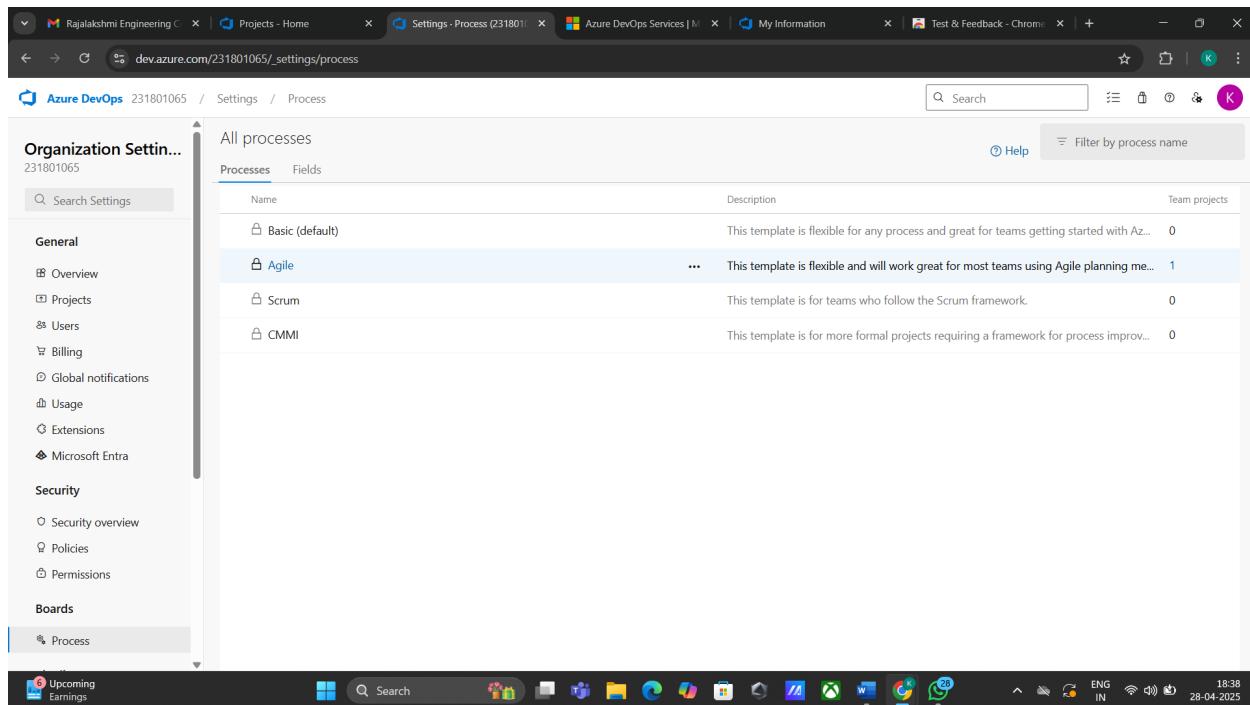


The screenshot shows the 'NEW BUG' work item creation dialog in Azure DevOps. The left sidebar is identical to the previous screenshot. The main dialog has sections for Issue (assignee: Krithika.MA, state: New, reason: New), Planning (Priority: 2, Severity: 3 - Medium), Deployment (Development section with a note about releases), and Effort (Hours). A 'Discussion' section at the bottom allows comments. The status bar at the bottom shows weather (31°C, Partly cloudy), system icons, and the date/time (28-04-2025).

## 10. Progress report



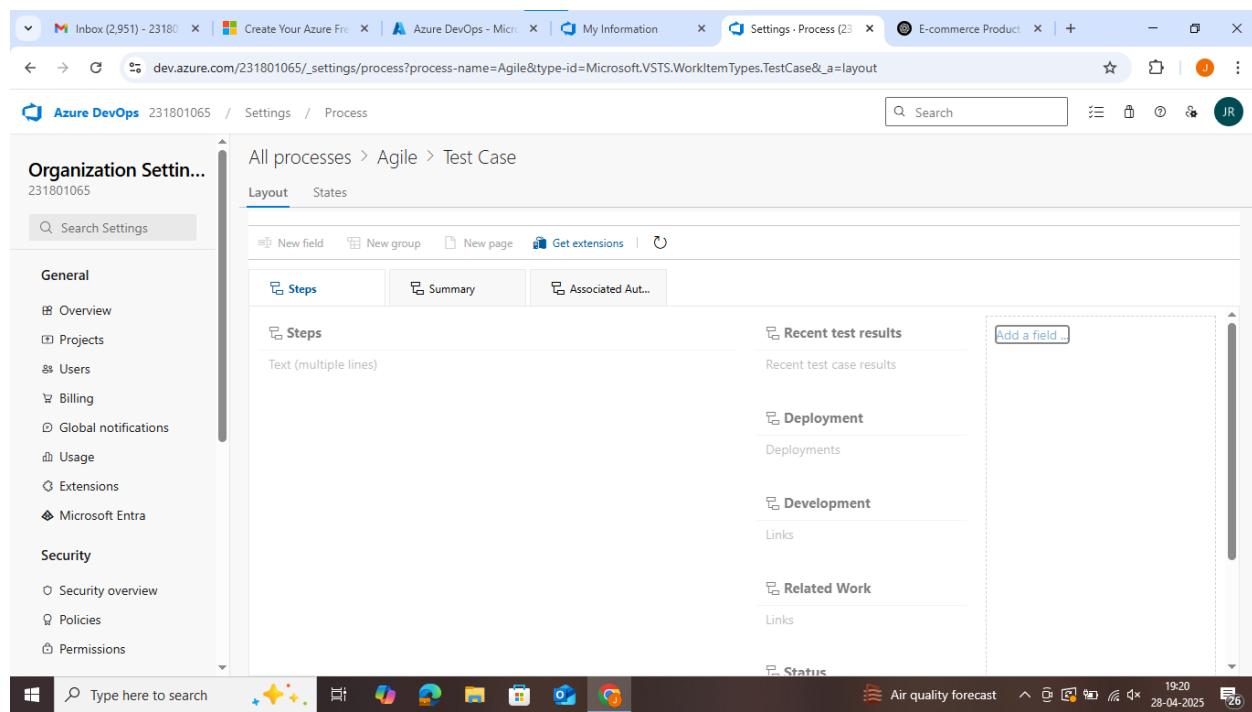
## 11. Changing the test template



The screenshot shows the 'Settings - Process' page in Azure DevOps. On the left, the navigation menu includes 'General', 'Security', and 'Process'. The 'Process' section is selected. In the center, a modal dialog titled 'Create inherited process from Agile' is open. It contains fields for 'Name' (set to 'Agile [system process]') and 'Description'. Below the modal, a list of processes is visible, including 'Basic (default)', 'Agile', 'Scrum', and 'CMMI'. A search bar at the top right allows filtering by process name.

## 12. View the new test case template

The screenshot shows the 'Settings - Process' page in Azure DevOps. The 'Process' section is selected in the navigation menu. A modal dialog titled 'Add a field to Test Case' is open, showing the 'Definition' tab. Under 'Create a field', a new field named 'e-commerce' is being created with a type of 'Text (single line)'. The 'Add field' button is at the bottom right of the modal.



### 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**

## **LOAD TESTING AND PIPELINES**

### **Aim:**

To create an Azure Load Testing resource and run a load test to evaluate the performance of a target endpoint and to create and demonstrate an Azure DevOps pipeline for automating application builds, tests, and deployment.

### **Load Testing**

#### **Azure Load Testing:**

Azure Load Testing allows you to simulate high traffic and stress tests for your web applications and APIs to understand how they perform under load. It helps identify performance bottlenecks, scalability issues, and optimize resource usage before deployment.

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

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

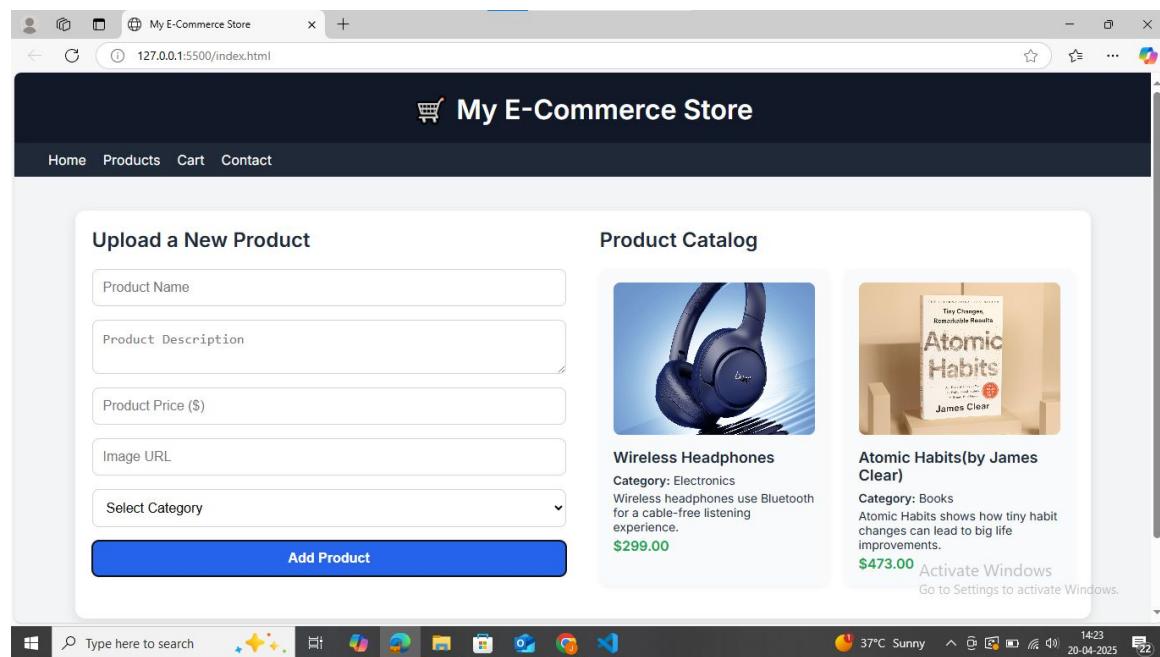
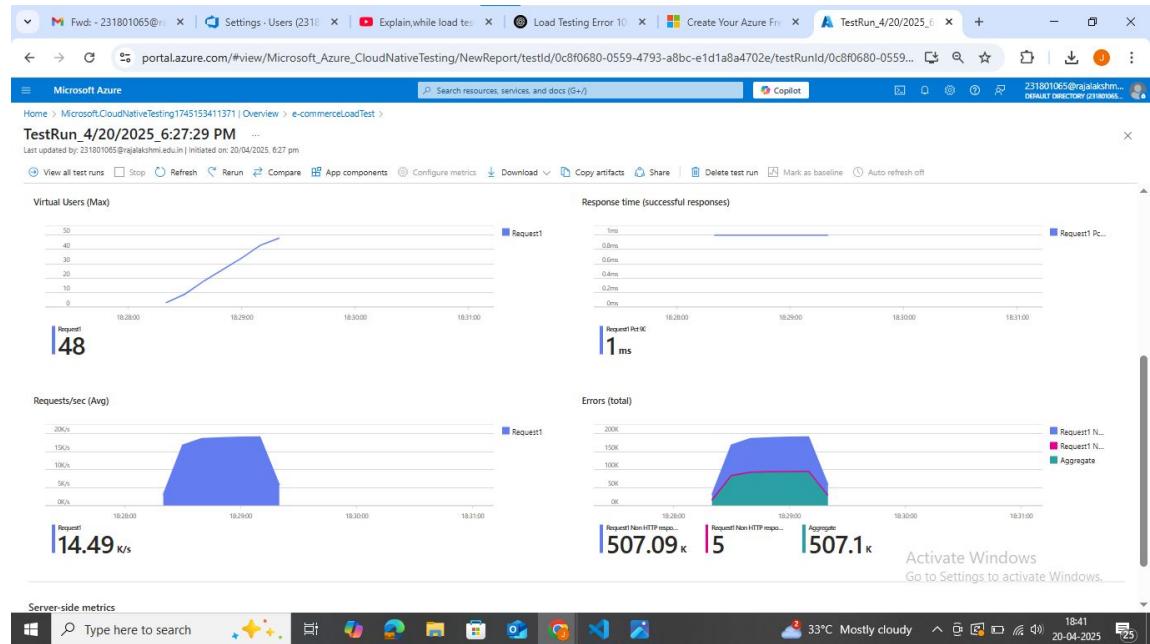
1. Sign in to Azure Portal  
Go to <https://portal.azure.com> and log in.
2. Create the Resource
  - o Go to *Create a resource* → Search for “Azure Load Testing”.
  - o Select Azure Load Testing and click Create.
3. Fill in the Configuration Details
  - o *Subscription*: Choose your Azure subscription.
  - o *Resource Group*: Create new or select an existing one.
  - o *Name*: Provide a unique name (no special characters).
  - o *Location*: Choose the region for hosting the resource.
4. (Optional) Configure tags for categorization and billing.
5. Click Review + Create, then Create.
6. 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
  - o *Test Name*: Provide a unique name.
  - o *Description*: (Optional) Add test purpose.
  - o *Run After Creation*: Keep checked.
3. Load Settings
  - o *Test URL*: Enter the target endpoint (e.g., <https://yourapi.com/products>).
4. Click Review + Create → Create to start the test.

## Load Testing



## Pipelines

### Description:

This experiment demonstrates how to connect a GitHub-hosted Flask-based music recommendation project with Azure DevOps. The pipeline will automatically install dependencies, run basic tests, and publish artifacts. This ensures that every commit triggers checks for reliability and smooth deployment.

Steps:

1. Connect GitHub to Azure DevOps:
  - o In Azure DevOps, create a new project.
  - o Create a pipeline and select GitHub as the source.
  - o Authorize access to your GitHub repository, ensuring that Azure DevOps can pull the repository for your pipeline.
2. Create azure-pipelines.yml in Your Repo Root:
  - o In your GitHub repository, create a new file called azure-pipelines.yml in the root directory.
  - o Add the following basic pipeline configuration for Python and Flask:

### **yml Code**

trigger:

```
- main # Trigger pipeline when changes are pushed to the main branch
```

pool:

```
vmImage: ubuntu-latest # Use a hosted Ubuntu agent
```

steps:

```
# Step 1: Checkout the code from GitHub
```

```
- checkout: self
```

```
# Step 2: Set up Python environment
```

```
- task: UsePythonVersion@0
```

inputs:

```
versionSpec: '3.x' # Use the latest Python 3.x version
```

```
displayName: "Set up Python"
```

```
# Step 3: Install dependencies from the correct path
```

```
- script: |
```

```
    python -m pip install --upgrade pip
```

```
    pip install -r project/requirements.txt # Adjusted path to requirements.txt
```

```
displayName: "Install dependencies"
```

```
# Step 4: Run a simple Python script to check the environment
```

```
- script: |
```

```
    python -c "print('♪ Hello from Music Playlist Batch Creator!')"
```

```
displayName: "Run a Python script"
```

3. Pipeline Tasks Include:

- o Setting up the Python environment using the UsePythonVersion task.

- Installing project dependencies from project/requirements.txt. Make sure the path to requirements.txt is correct (it is located under the project folder).
  - Running a simple Python script to verify that Python is set up correctly and the pipeline works.
4. Run and Monitor Pipeline:
- Commit changes to the main branch of your repository to trigger the pipeline in Azure DevOps.
  - Monitor the logs in the Azure DevOps portal to view logs, errors, or success messages and ensure everything runs smoothly.

## Pipeline

The screenshot shows the Azure DevOps Pipelines interface for a project named "Music Playlist Batch Creator". A recent pipeline run (#20250424.3) is displayed. The summary card indicates the run was manually triggered by Karthick S and retained as one of three recent runs. It shows the repository version (main@a87bd670), the start time (Just now), and duration (24s). The "Jobs" section lists a single job that completed successfully in 6s. The left sidebar navigation includes Overview, Boards, Repos, Pipelines (selected), Environments, Releases, Library, Task groups, Deployment groups, Test Plans, and Artifacts. The bottom left corner shows "Project settings".

## Result:

Successfully created the Azure Load Testing resource and executed a load test to assess the performance of the specified endpoint and also demonstrated pipelines in azure devops.

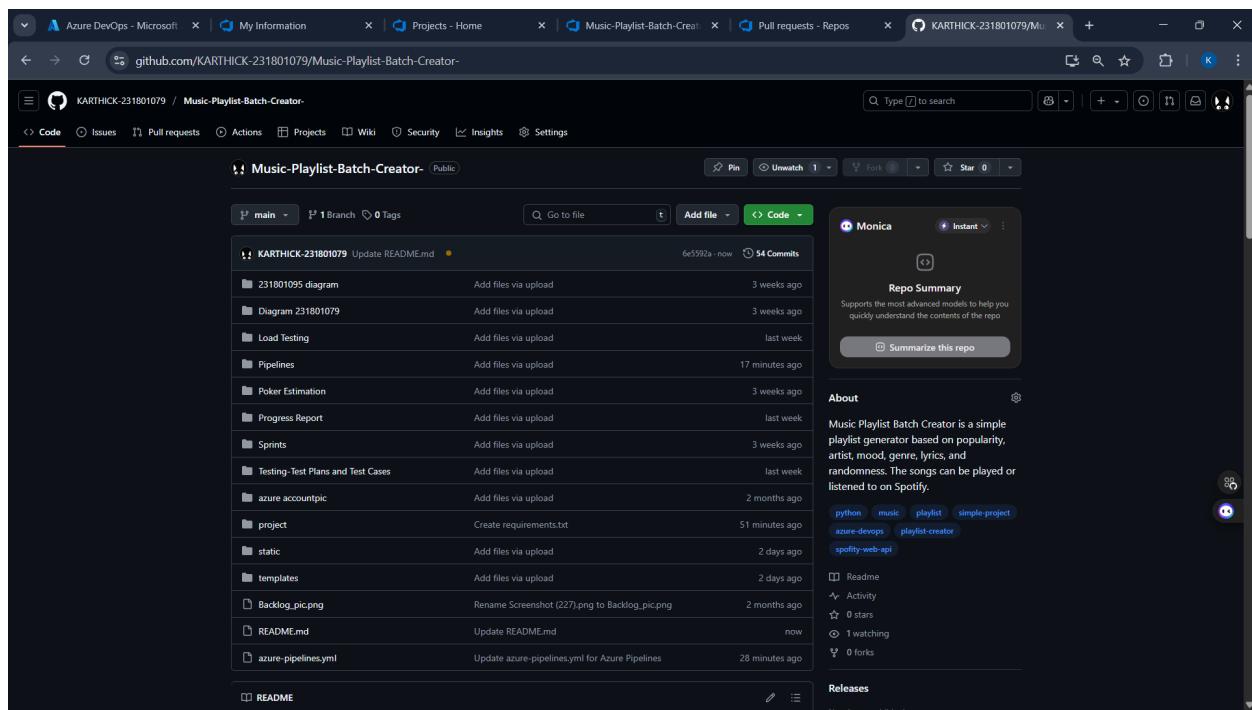
**EXP NO: 10**

## **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 E Commerce 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.