

EXP NO: 1
Create Epic, F

AZURE DEVOPS ENVIRONMENT SETUP

Aim:

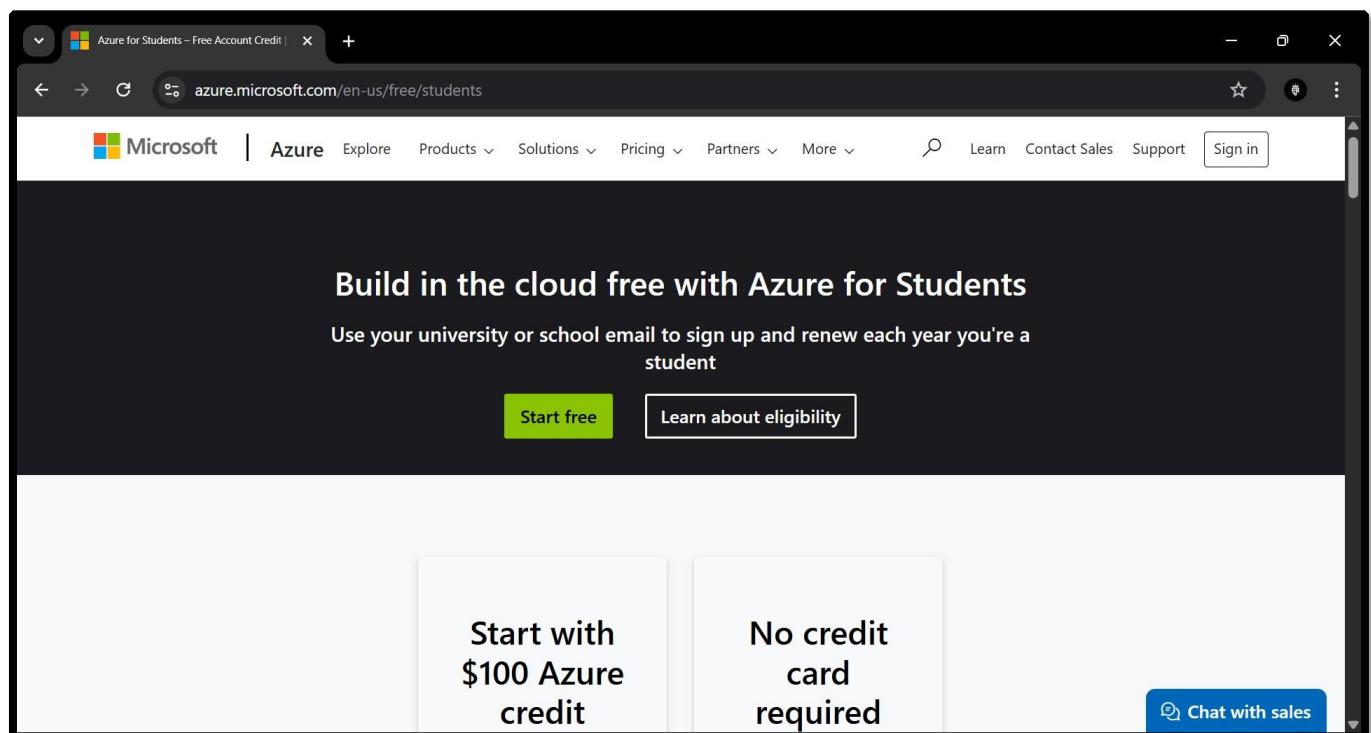
To set up and access the Azure DevOps environment by creating an organization through the Azure portal.

INSTALLATION

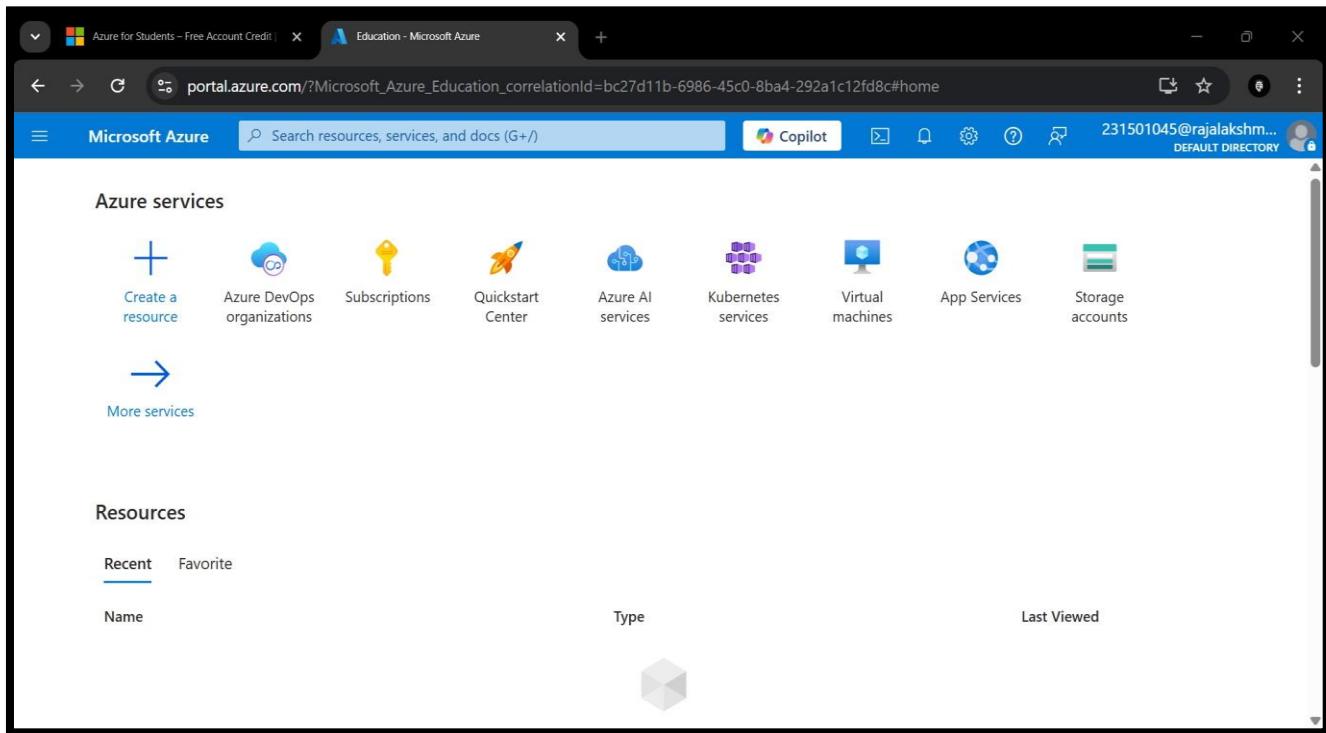
1. Open your web browser and go to the Azure website: <https://azure.microsoft.com/en-us/get-started/azure-portal>.

Sign in using your Microsoft account credentials.

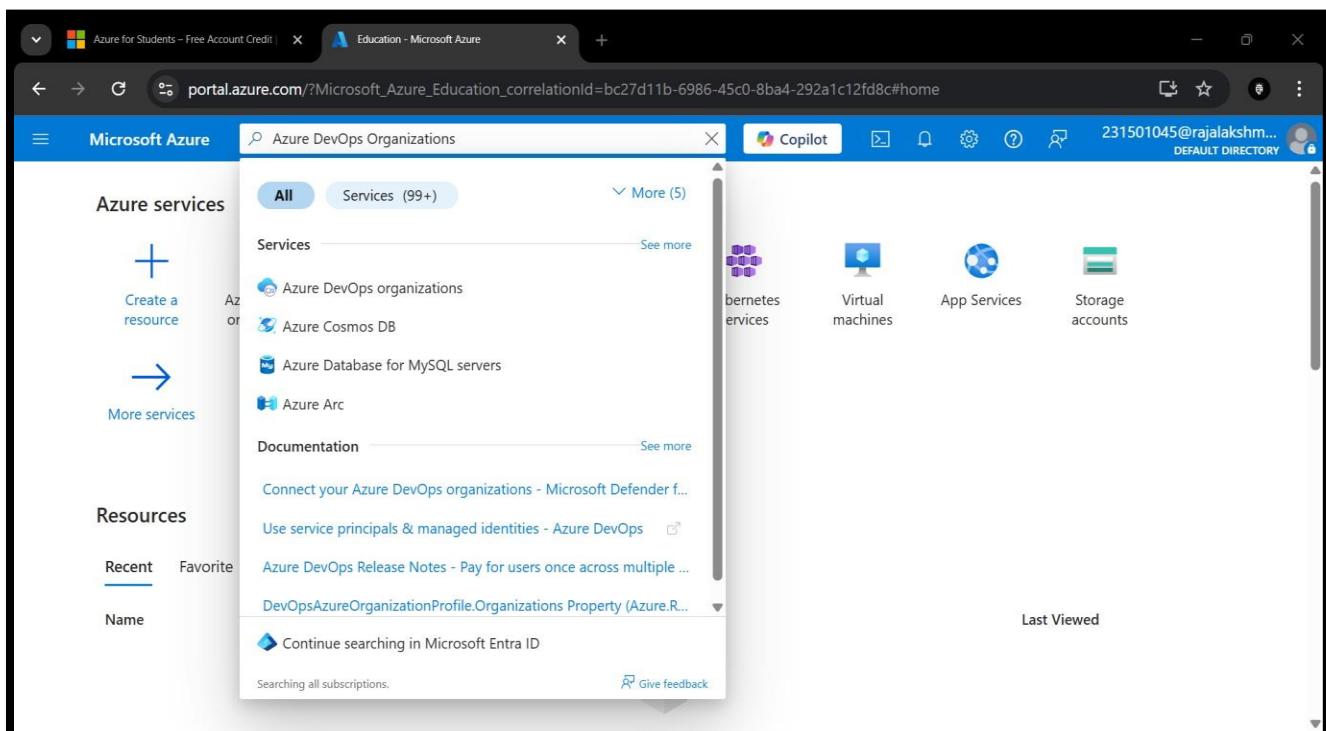
If you don't have a Microsoft account, you can create one here: <https://signup.live.com/?lic=1>



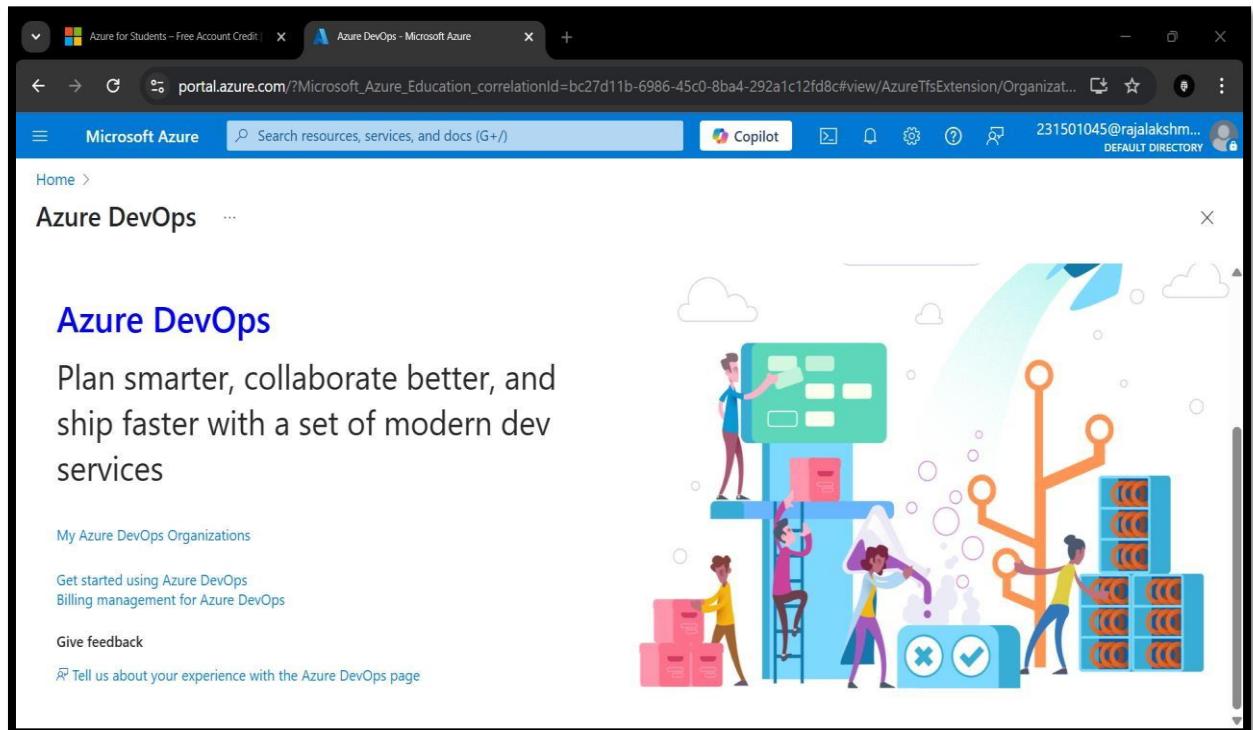
2. Azure home page



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



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:

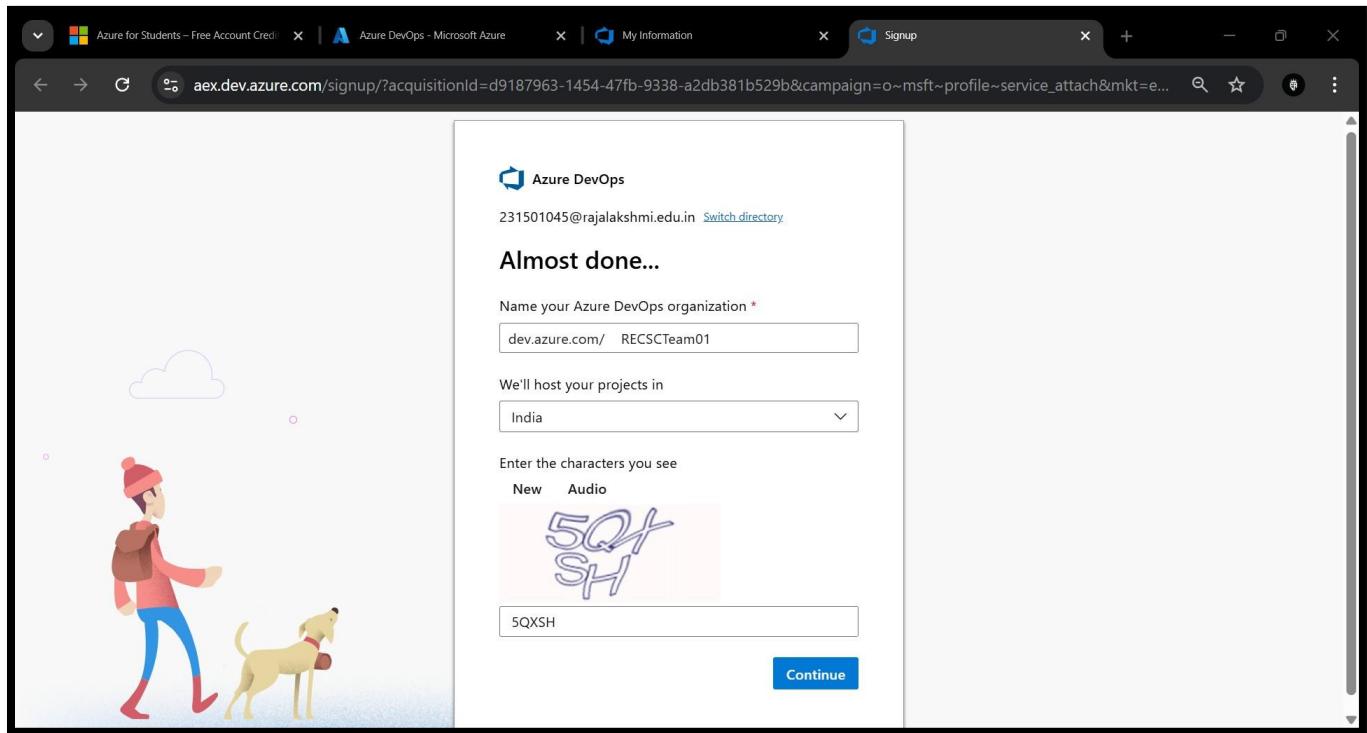
Successfully accessed the Azure DevOps environment and created a new organization through the Azure portal.

EXP NO: 2 Create Epic, Features, User Stories, Task	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.

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 Organization Home page for 'RECSCTeam01'. The left sidebar lists organization members: RECSCTeam01 (R), Ezhil007 (E), and Devesh09 (D). A 'New organization' link is also present. The main area displays the project 'Weather App' (WA) with a brief description: 'This project is a beginner-friendly Weather Application developed using HTML, CSS, and JavaScript. The application enables users to enter a location and receive real-time weather data, includin...'. Navigation links include 'Projects', 'My work items', 'My pull requests', and a 'Filter projects' button. A 'New project' button is located in the top right. The bottom left of the main area has a 'Organization settings' link.

4. Project dashboard

The screenshot shows the Azure DevOps Project Overview page for the 'Weather App' project under 'RECSCTeam01'. The left sidebar includes links for 'Overview', 'Summary', 'Dashboards', 'Wiki', 'Boards', 'Repos', 'Pipelines', 'Test Plans', and 'Artifacts'. The main content area features the title 'Weather App' and a 'About this project' section with a description of the application's purpose and key features. To the right, there are 'Project stats' and 'Members' sections. The 'Project stats' section shows '35 Work items created' and '0 Work items completed' over the last 7 days. The 'Members' section shows one member, 'EP'. A 'Period: Last 7 days' dropdown is also present.

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 Weather App Team. The left sidebar includes options like Overview, Boards, Work items, Backlogs, Sprints, Queries, Delivery Plans, Analytics views, and Project settings. The main area displays the Backlog for the Weather App Team, showing three items:

Order	Work Item Type	Title	State
1	Epic	> EPIC 1: User Interface Design	New
2	Epic	> EPIC 2: Weather Data Integration	New
3	Epic	> EPIC 3: App Reliability and Enhancement	New

This screenshot is similar to the previous one, showing the Azure DevOps Boards Backlog page. A Microsoft sign-in overlay is visible on the right side, displaying the user's profile picture (EP), name (Ezhil Adhithya P), email (231501045@rajalakshmi.edu.in), and links to 'My Microsoft account' and 'Switch directory'. The backlog table remains the same:

Order	Work Item Type	Title	State
1	Epic	> EPIC 1: User Interface Design	New
2	Epic	> EPIC 2: Weather Data Integration	New
3	Epic	> EPIC 3: App Reliability and Enhancement	New

2116231501020

CS23432

Result:

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

EXP NO: 3 Create Epic, Features, User Stories, Task	<h2>SETTING UP EPICS, FEATURES, AND USER STORIES FOR PROJECT PLANNING</h2>
---	--

Aim:

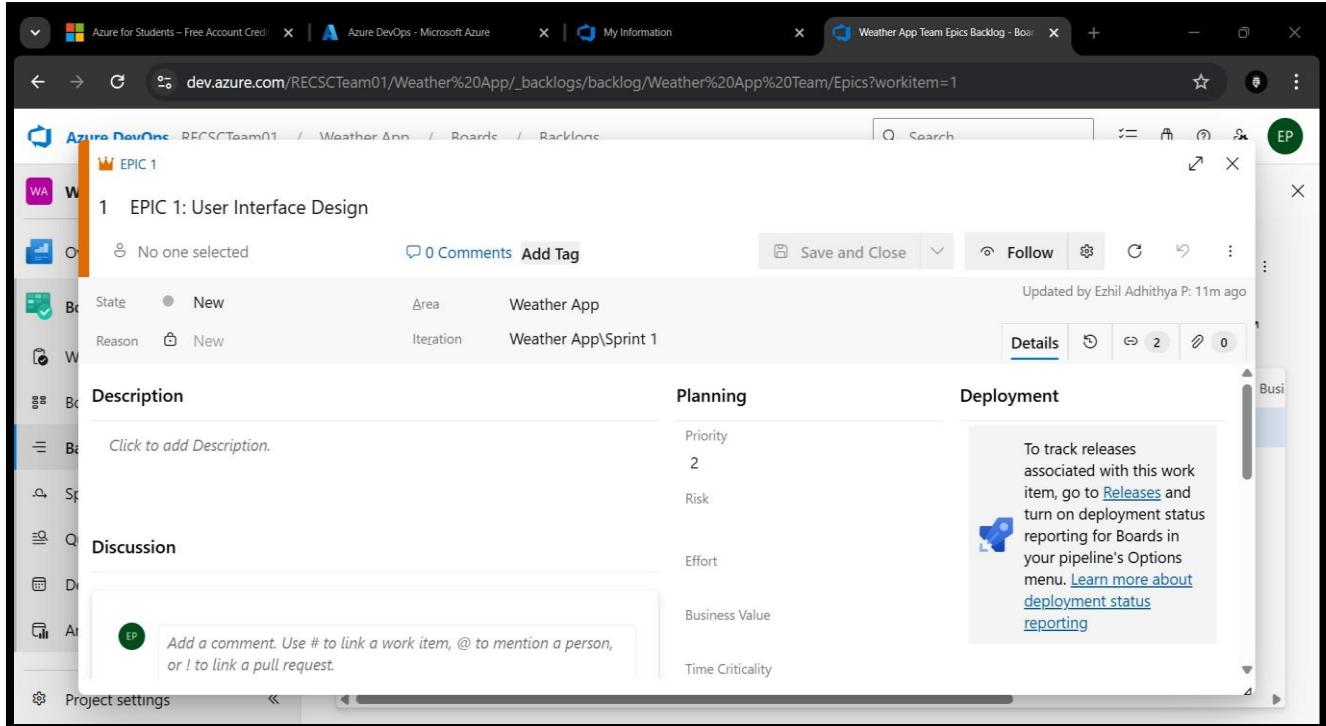
To learn about how to create epics, user story, features, backlogs for your assigned project.

Create Epic, Features, User Stories, Task

The screenshot shows the Azure DevOps Backlog page for the Weather App Team. The left sidebar includes options like Overview, Boards, Work items, Boards, Backlogs, Sprints, Queries, Delivery Plans, Analytics views, and Project settings. The main area displays a backlog with the following structure:

Order	Work Item Type	Title	State	Effort	Business Value
1	Epic	EPIC 1: User Interface Design	New		
	Feature	Feature 1.1: Basic Layout and Styling	New		
	User Story	User Story 1.1.1: As a user, I want to see a simple and ...	New		
	User Story	User Story 1.1.2: As a user, I want the app to be responsiv...	New		
2	Feature	Feature 1.2: Interactive Input and Display	New		
	Epic	EPIC 2: Weather Data Integration	New		
3	Epic	EPIC 3: App Reliability and Enhancement	New		

1. Fill in Epics

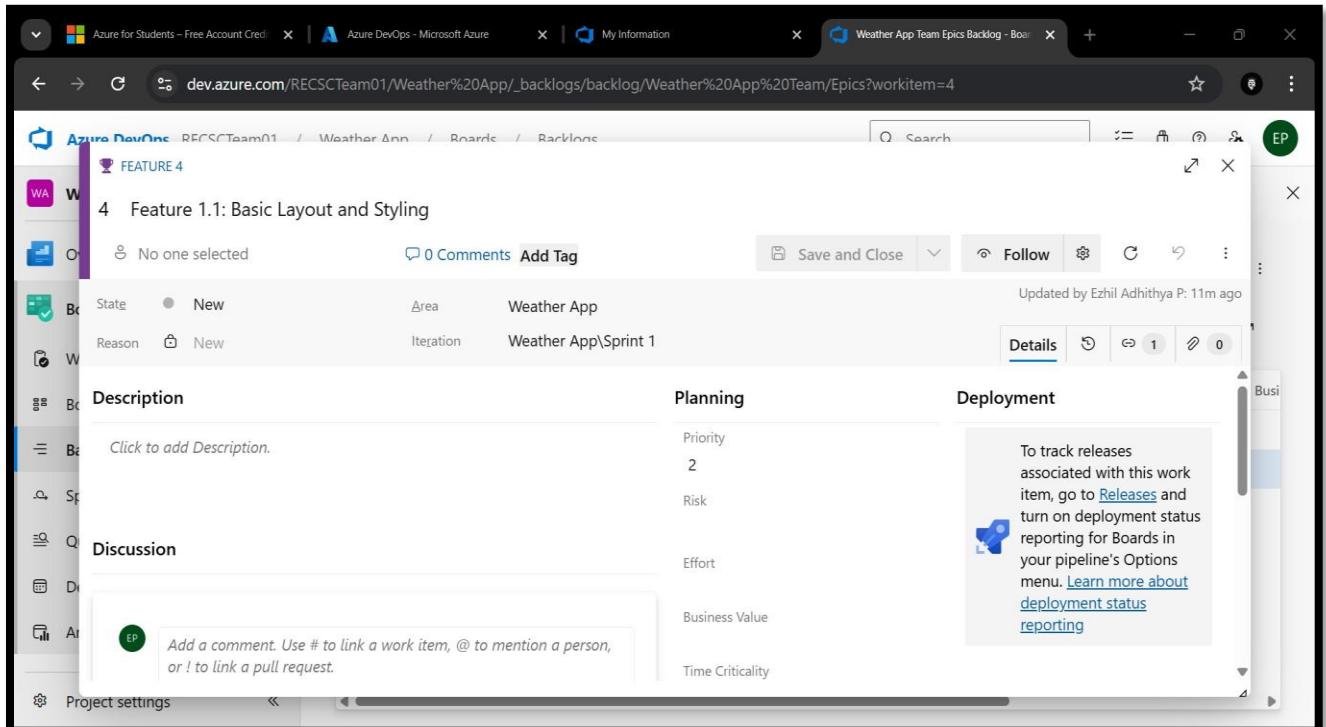


A screenshot of the Azure DevOps interface showing the creation of an Epic. The URL is https://dev.azure.com/RECSCTeam01/Weather%20App/_backlogs/backlog/Weather%20App%20Team/Epics?workitem=1. The page title is "EPIC 1". The epic details are as follows:

Category	Value
State	New
Reason	New
Area	Weather App
Iteration	Weather App\Sprint 1

The "Description" section contains the placeholder text "Click to add Description." The "Planning" section includes fields for Priority (set to 2), Risk, Effort, Business Value, and Time Criticality. A tooltip in the "Deployment" section provides instructions on tracking releases associated with the work item.

2. Fill in Features



A screenshot of the Azure DevOps interface showing the creation of a Feature. The URL is https://dev.azure.com/RECSCTeam01/Weather%20App/_backlogs/backlog/Weather%20App%20Team/Epics?workitem=4. The feature details are as follows:

Category	Value
State	New
Reason	New
Area	Weather App
Iteration	Weather App\Sprint 1

The "Description" section contains the placeholder text "Click to add Description." The "Planning" section includes fields for Priority (set to 2), Risk, Effort, Business Value, and Time Criticality. A tooltip in the "Deployment" section provides instructions on tracking releases associated with the work item.

3. Fill in User Story Details

The screenshot shows the Azure DevOps interface for editing a work item. The URL in the browser is https://dev.azure.com/RECSCTeam01/Weather%20App/_backlogs/backlog/Weather%20App%20Team/Epics?workitem=14. The work item type is 'USER STORY' and the ID is 14. The title of the user story is '14 User Story 1.1.1: As a user, I want to see a simple and clean layout, so I can easily understand the weather information.' The 'Area' is 'Weather App' and the 'Iteration' is 'Weather App\Sprint 1'. The 'State' is 'New'. There are 0 comments and 0 tags. The 'Details' tab is selected. The 'Description' field contains the story text. The 'Planning' section includes 'Story Points' (set to 2), 'Priority' (set to 2), and 'Risk'. The 'Deployment' section has 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 'Classification' section is empty.

Result:

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

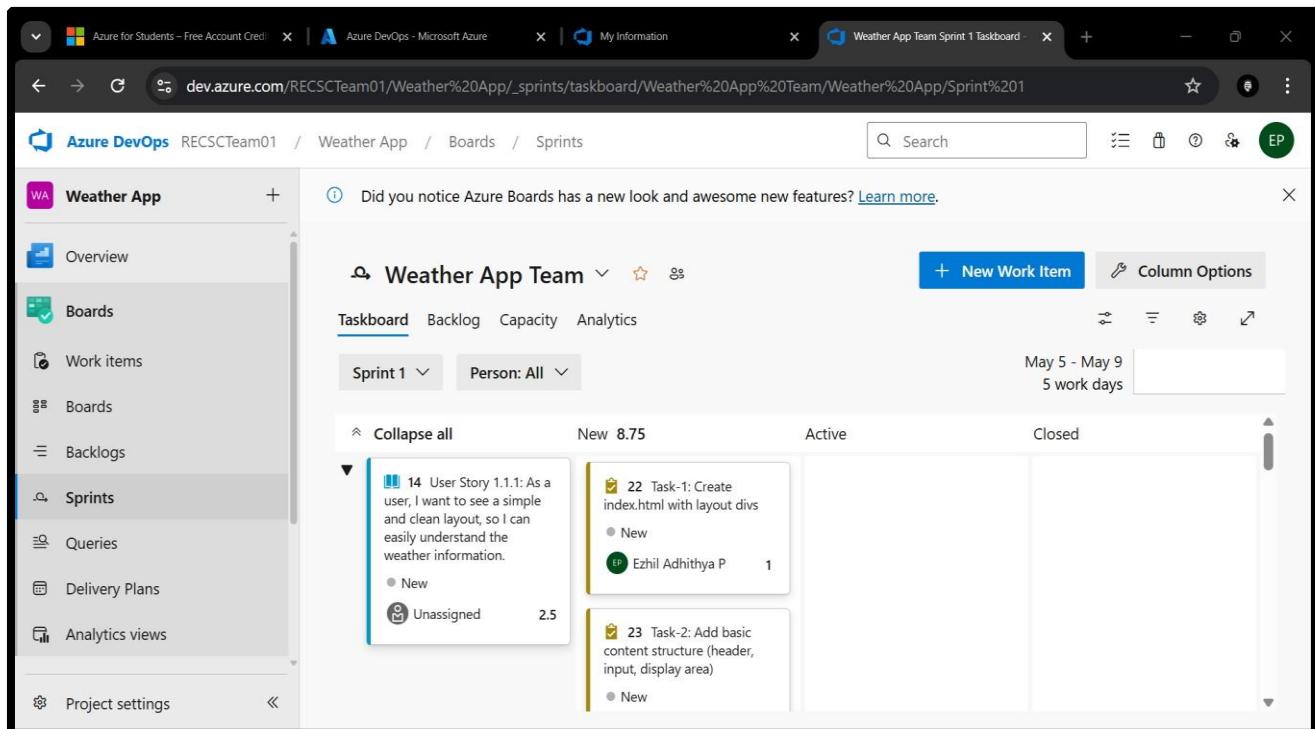
EXP NO: 4
Create Epic, F

SPRINT PLANNING

Aim:

To assign user story to specific sprint for the Weather App Project.

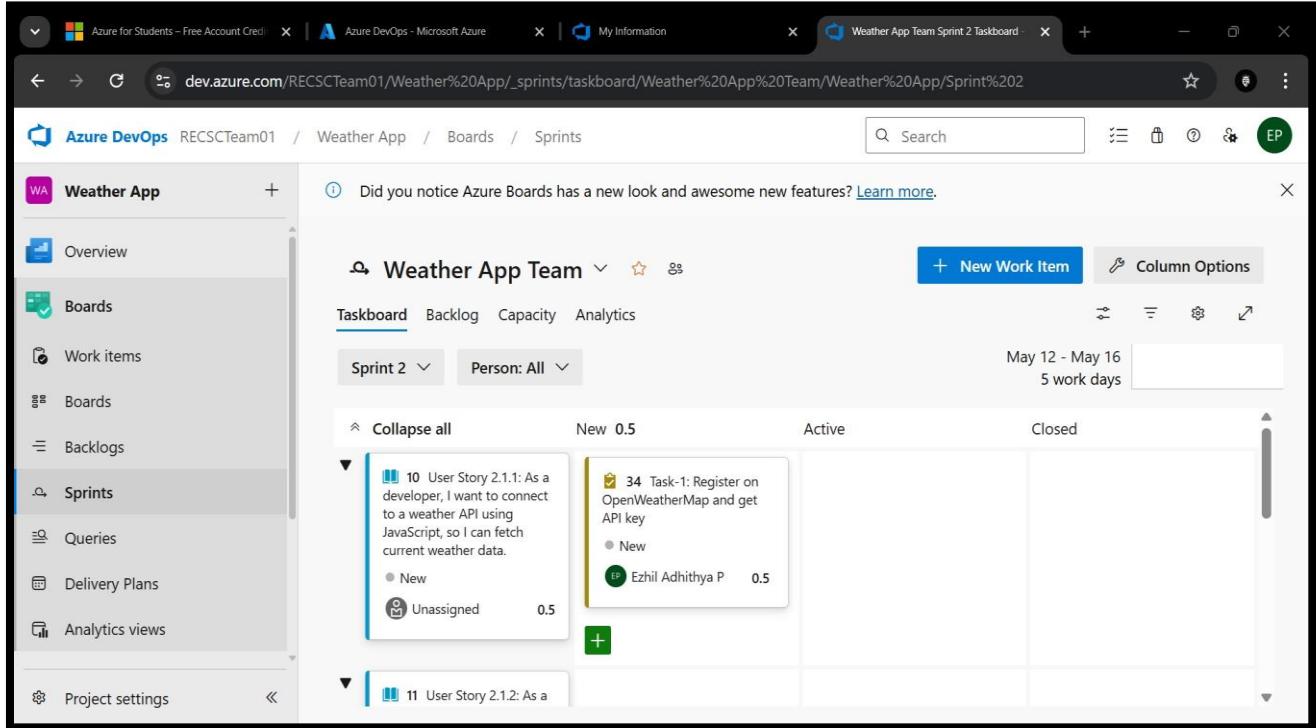
Sprint Planning Sprint 1



The screenshot shows the Azure DevOps Taskboard for the Weather App project. The left sidebar navigation includes Weather App, Overview, Boards, Work items, Boards, Backlogs, Sprints (selected), Queries, Delivery Plans, Analytics views, and Project settings. The main area displays the Taskboard for the Weather App Team, showing the following details:

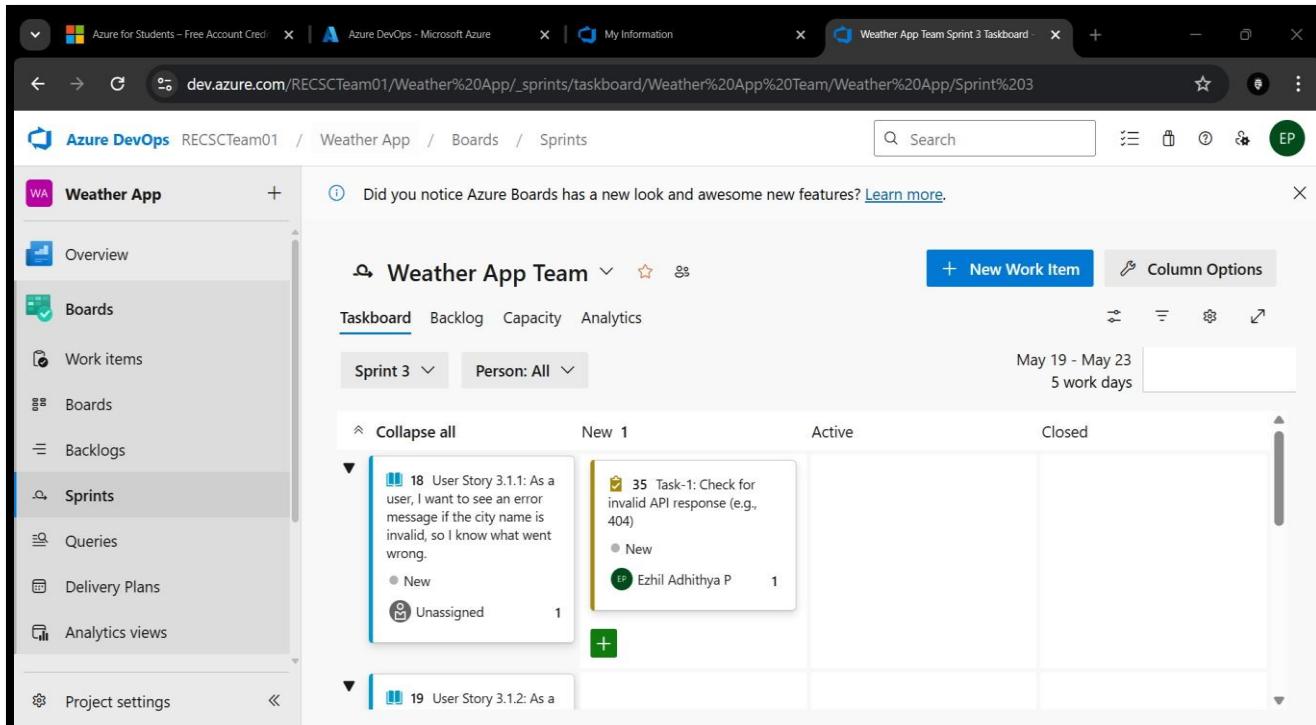
- Taskboard:** Weather App Team
- Sprint:** Sprint 1
- Person:** All
- Timeline:** May 5 - May 9, 5 work days
- Columns:** New, Active, Closed
- Work Items:** A list of tasks and user stories assigned to the sprint:
 - User Story 1.1.1: As a user, I want to see a simple and clean layout, so I can easily understand the weather information. (New, Unassigned, 2.5 points)
 - Task-1: Create index.html with layout divs (New, Ezhil Adhithya P, 1 point)
 - Task-2: Add basic content structure (header, input, display area) (New)

Sprint 2



A screenshot of the Azure DevOps Taskboard for Sprint 2. The left sidebar shows the project navigation: Azure DevOps, RECSCTeam01, Weather App, Boards, Sprints, Work items, Boards, Backlogs, Queries, Delivery Plans, Analytics views, and Project settings. The main area displays the Taskboard for the Weather App Team, with the title "Weather App Team". The board has columns for Taskboard, Backlog, Capacity, and Analytics. The "Sprint 2" column is selected, showing tasks from May 12 to May 16 (5 work days). A tooltip for task 10 provides a detailed description: "User Story 2.1.1: As a developer, I want to connect to a weather API using JavaScript, so I can fetch current weather data." The task is assigned to Ezhil Adhithya P and has a priority of 0.5.

Sprint 3



A screenshot of the Azure DevOps Taskboard for Sprint 3. The left sidebar shows the project navigation: Azure DevOps, RECSCTeam01, Weather App, Boards, Sprints, Work items, Boards, Backlogs, Queries, Delivery Plans, Analytics views, and Project settings. The main area displays the Taskboard for the Weather App Team, with the title "Weather App Team". The board has columns for Taskboard, Backlog, Capacity, and Analytics. The "Sprint 3" column is selected, showing tasks from May 19 to May 23 (5 work days). A tooltip for task 18 provides a detailed description: "User Story 3.1.1: As a user, I want to see an error message if the city name is invalid, so I know what went wrong." The task is assigned to Ezhil Adhithya P and has a priority of 1.

Result:

The Sprints are created for the Weather App Project.

2116231501001

CS23432

2116231501001

CS23432

EXP NO: 5
Create Epic, F

POKER ESTIMATION

Aim:

Create Poker Estimation for the user stories – Weather App Project.

Poker Estimation

The screenshot shows a Microsoft Azure DevOps interface for a 'Weather App' project. A specific user story, 'User Story 3.1.1: As a user, I want to see an error message if the city name is invalid, so I know what went wrong.', is selected. The story has a state of 'New', is assigned to the 'Weather App' area, and is part of the 'Weather App\Sprint 3' iteration. The 'Planning' section indicates 3 Story Points and Priority 1. The 'Acceptance Criteria' section lists two items: '1) Error message shows if API returns 404' and '2) Error disappears when a new search succeeds'. The 'Classification' section is currently empty. A tooltip on the right provides information about tracking releases and deployment status reporting.

Result:

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

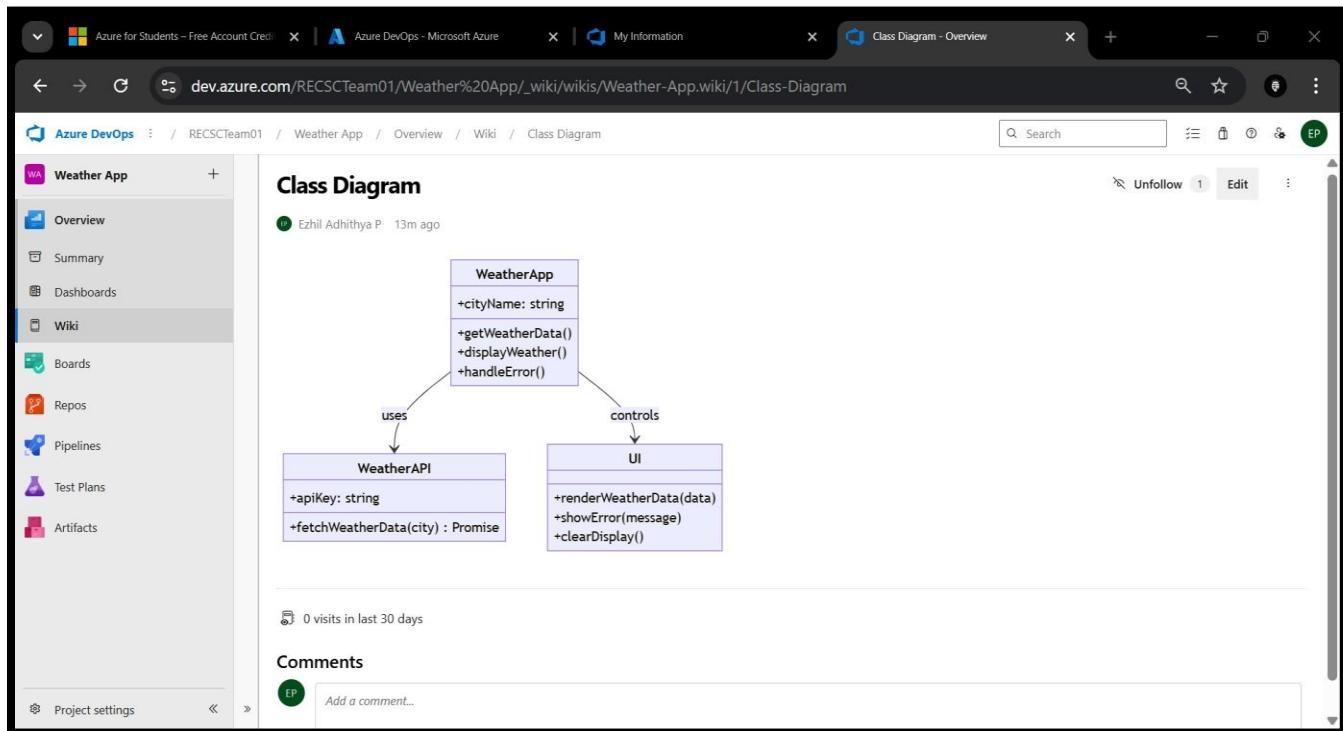
EXP NO: 6
Create Epic, Features, User Stories, Task

DESIGNING CLASS AND SEQUENCE DIAGRAMS FOR PROJECT ARCHITECTURE

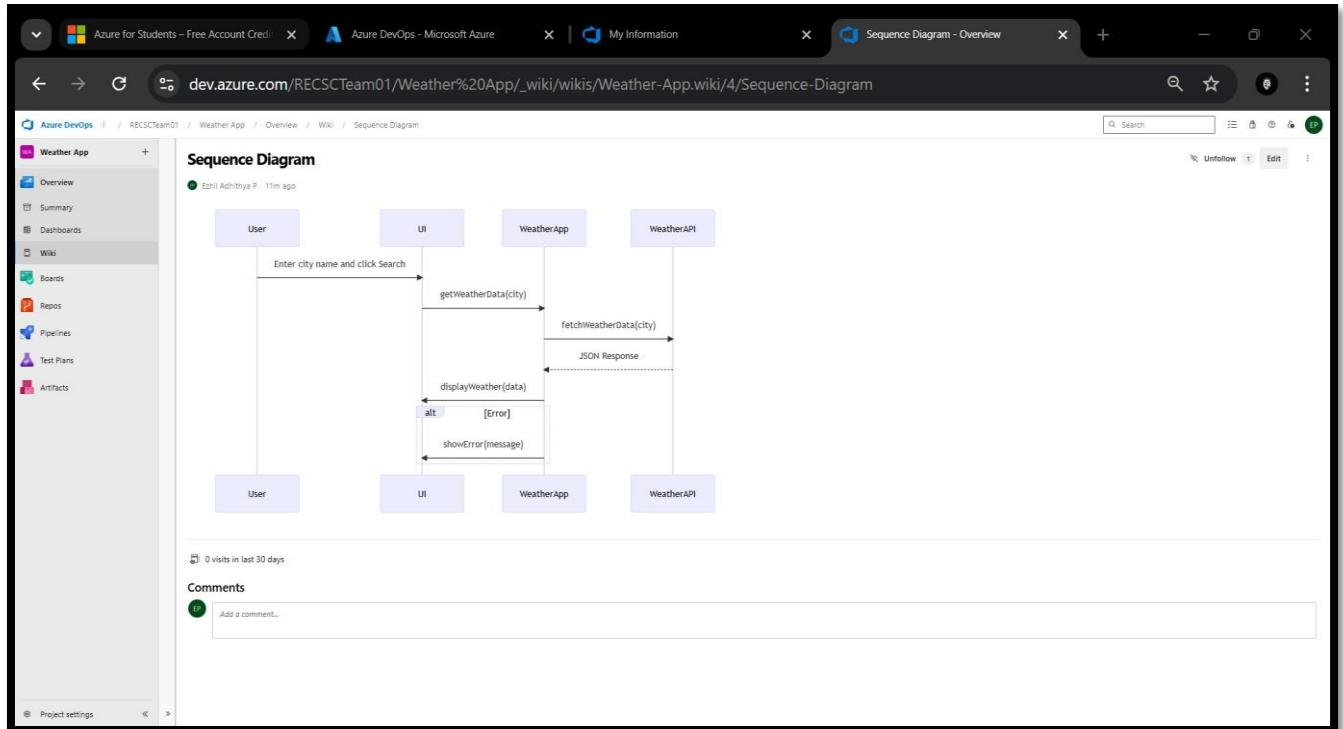
Aim:

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 Weather App Project.

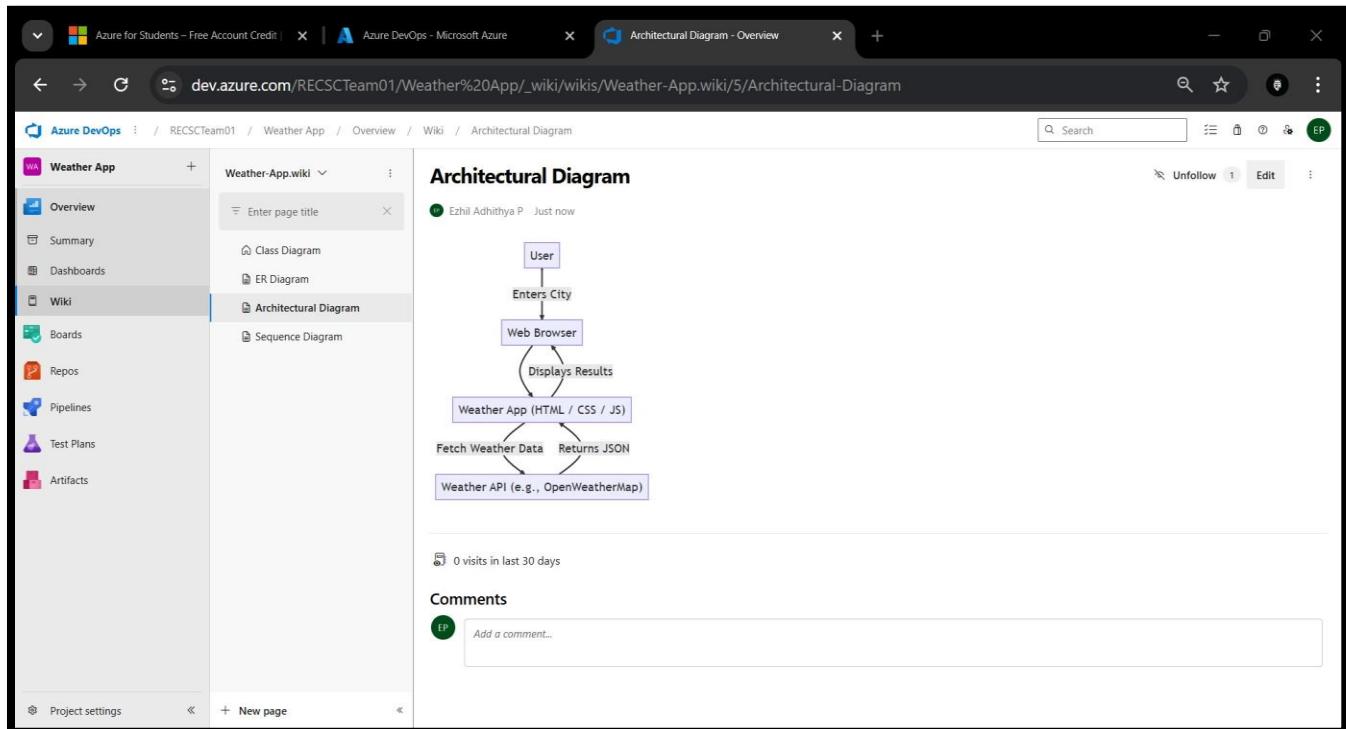
EXP NO: 7
Create Epic, Features, User Stories, Task

DESIGNING ARCHITECTURAL AND ER DIAGRAMS FOR PROJECT STRUCTURE

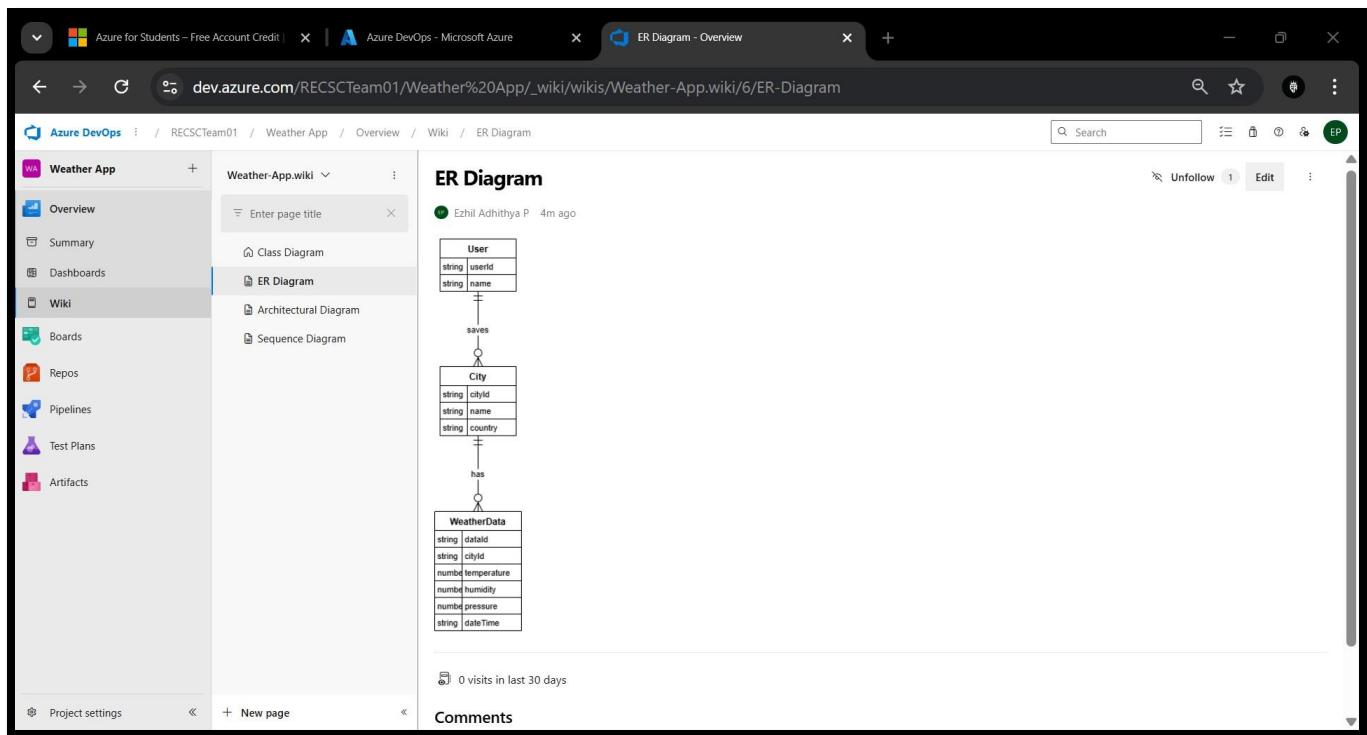
Aim:

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

7A. Architectural Diagram



7B. ER Diagram



CS23432

2116231501001

CS23432

Result:

The Architecture Diagram and ER Diagram is designed Successfully for the Weather App project.

2116231501001

EXP NO: 8 Create Epic, F	TESTING – TEST PLANS AND TEST CASES
--	--

Aim:

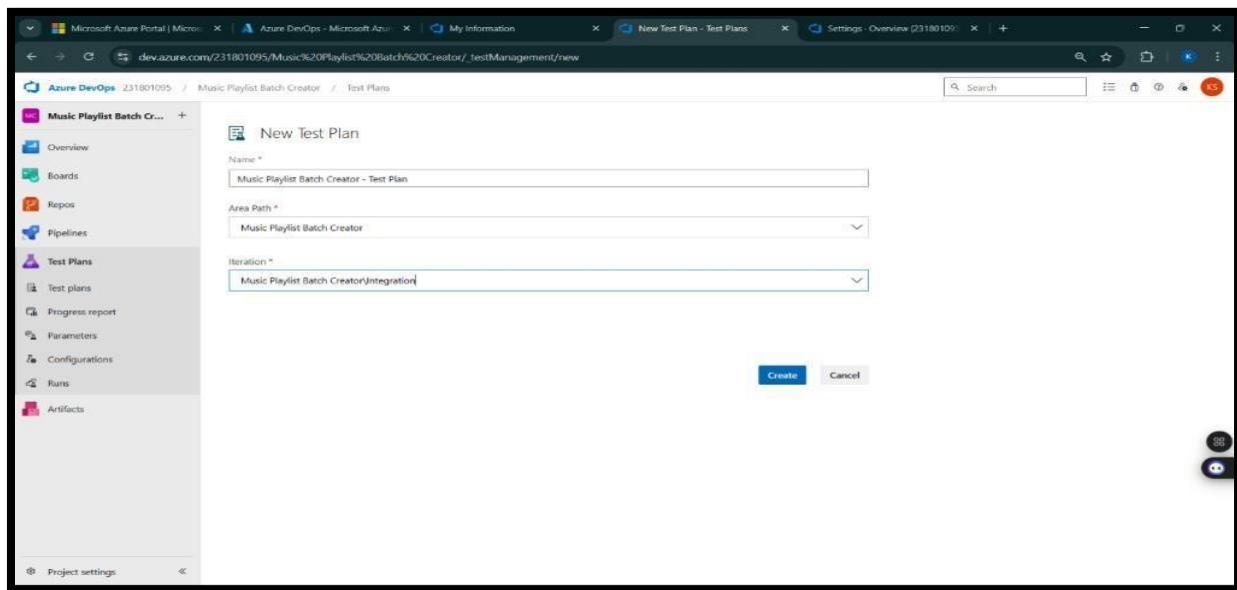
Test Plans and Test Case and write two test cases for at least five user stories showcasing the happy path and error scenarios in azure DevOps platform.

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



2. Test suite

Order	Test Case Id	Assigned To	Status
1	78	Karthikeyan Se...	Design
2	80	Karthikeyan Se...	Design
3	81	Karthikeyan Se...	Design
4	82	Karthikeyan Se...	Design

3. Test case

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

Weather App – 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

2116231501001

CS23432

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".

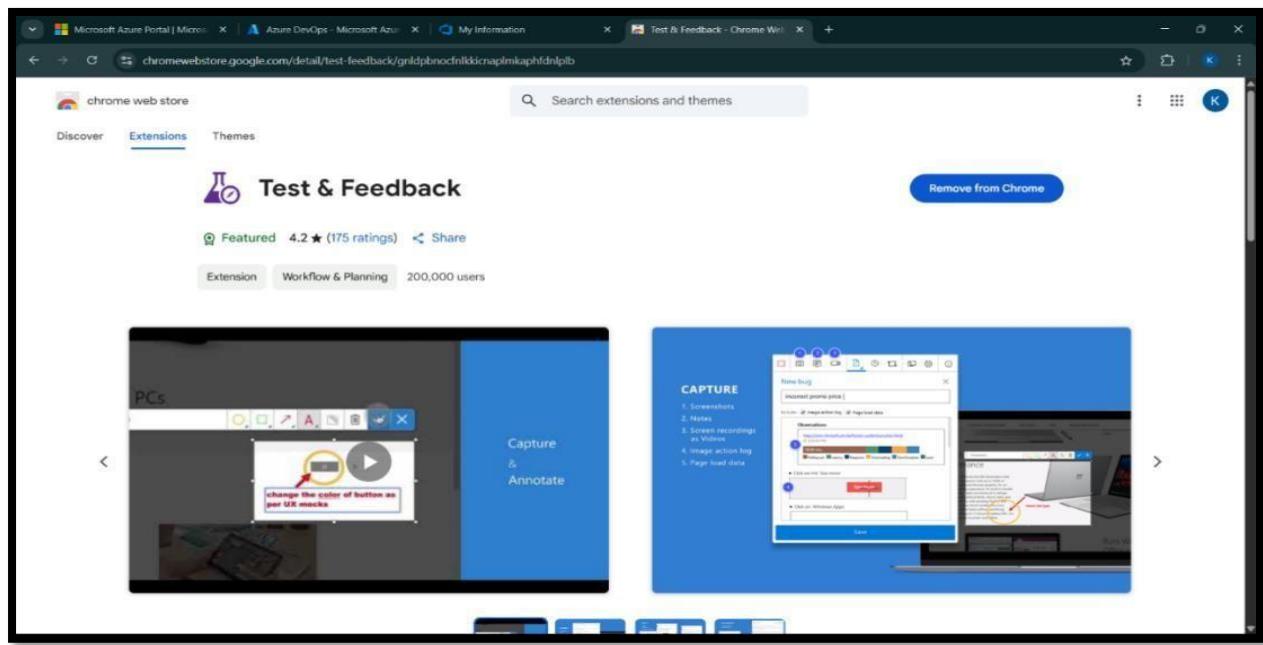
◦ Type: Error Path

Test Cases

The screenshot shows the Azure DevOps interface for a project named 'Weather App'. The left sidebar is open, showing options like Overview, Boards, Work items (which is selected), Backlogs, Sprints, Queries, Delivery Plans, Analytics views, Repos, and Project settings. The main area displays a work item titled 'TEST CASE 40' with the sub-id '40 TC 01: Simple Run'. The work item is assigned to 'Ezhil Adhithya P' and has 0 comments. It is in the 'Design' state under the 'Weather App' area, with a reason of 'New' and iteration 'Weather App\Sprint 1'. The 'Steps' tab is active, showing three steps: 'Click on text box', 'Update input field with value 'Chennai'', and 'Click on button 'Get Weather''. Below the steps, there is a note: 'Click or type here to add a step'. To the right of the work item details, there are sections for 'Deployment' and 'Development'. The deployment section includes a note about tracking releases and turning on deployment status reporting for Boards. The development section is currently empty.

4. Installation of test

The screenshot shows the Chrome Web Store page for the 'Test & Feedback' extension. The extension is listed as 'Featured' with a rating of 4.2 stars from 175 ratings and 200,000 users. The 'Add to Chrome' button is prominently displayed. The page features two main screenshots: one showing a PC interface with a red arrow pointing to a button labeled 'Change the color of button as per UX mockup', and another showing a laptop screen with a 'CAPTURE' interface for capturing screenshots, notes, and screen recordings. The extension is categorized under 'Workflow & Planning'.



Test and feedback

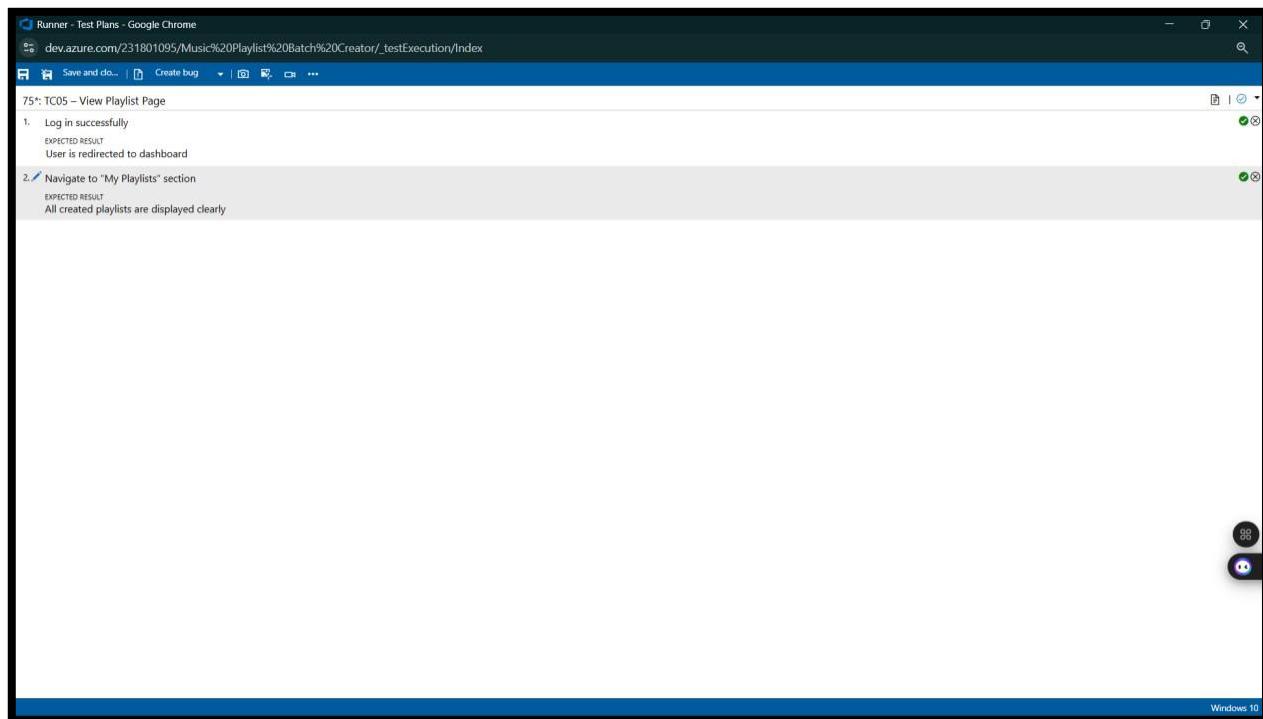
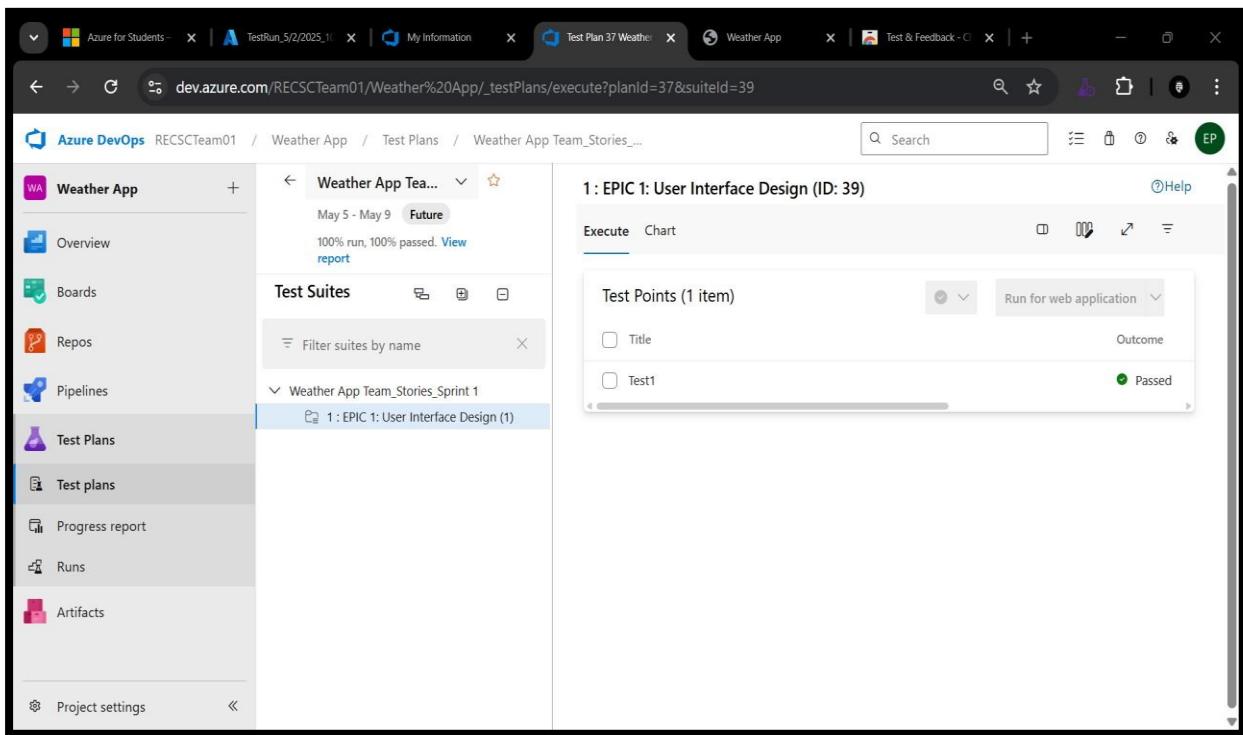
Showing it as an extension

A screenshot of a Microsoft Edge browser window showing the Azure DevOps interface. The user is viewing a test plan for a 'Music Playlist Batch Creator' project. On the right side of the screen, a floating 'Extensions' sidebar is open, listing several extensions including 'Copy Text from Picture', 'Dark Reader', 'Monica: ChatGPT AI Assist...', 'Selected: Copy Text from V...', and 'Test & Feedback'. The 'Test & Feedback' extension is highlighted with a dark overlay. The main test plan view shows a 'TS01 - User Login (ID: 86)' test suite with four test cases: 'TC01 – Successful Sign Up', 'TC02 – Secure Login', 'TC03 – Sign Up with Existing Email', and 'TC04 – Login with Wrong Password'. The status for all cases is 'Passed'.

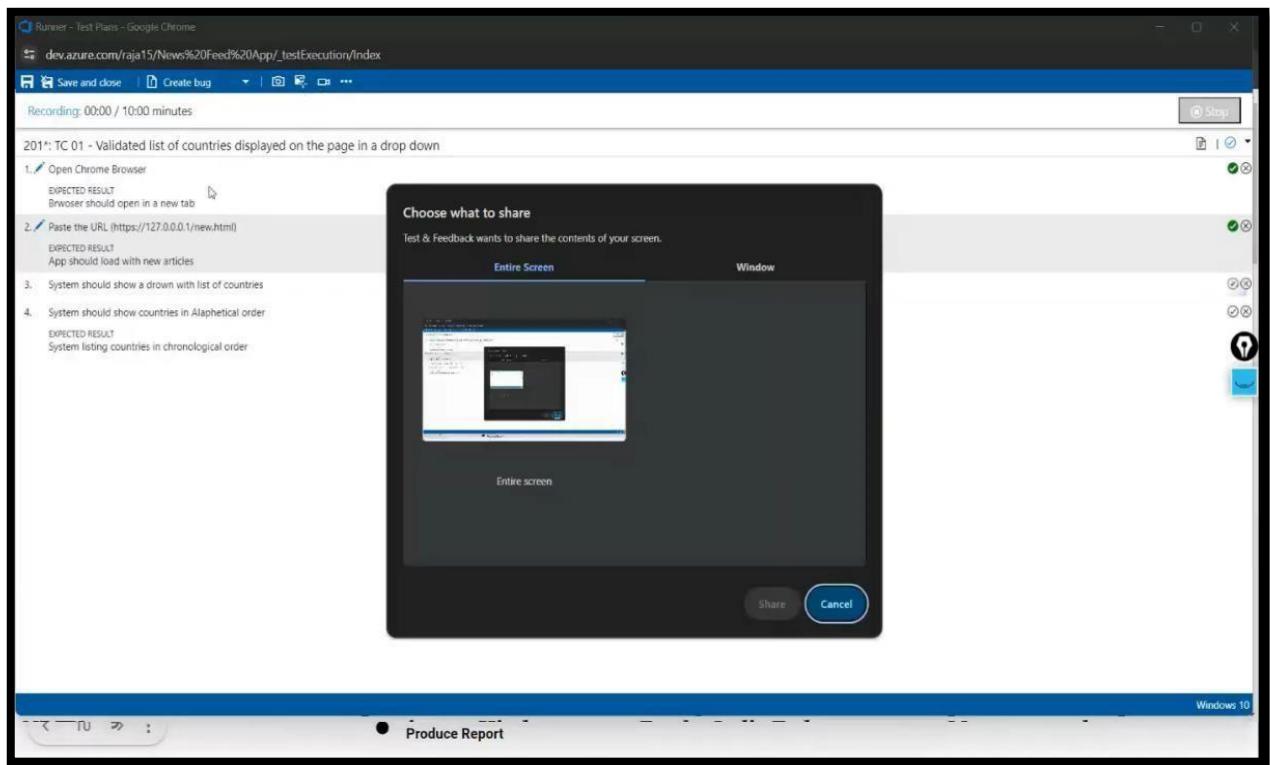
5. Running the test cases

2116231501001

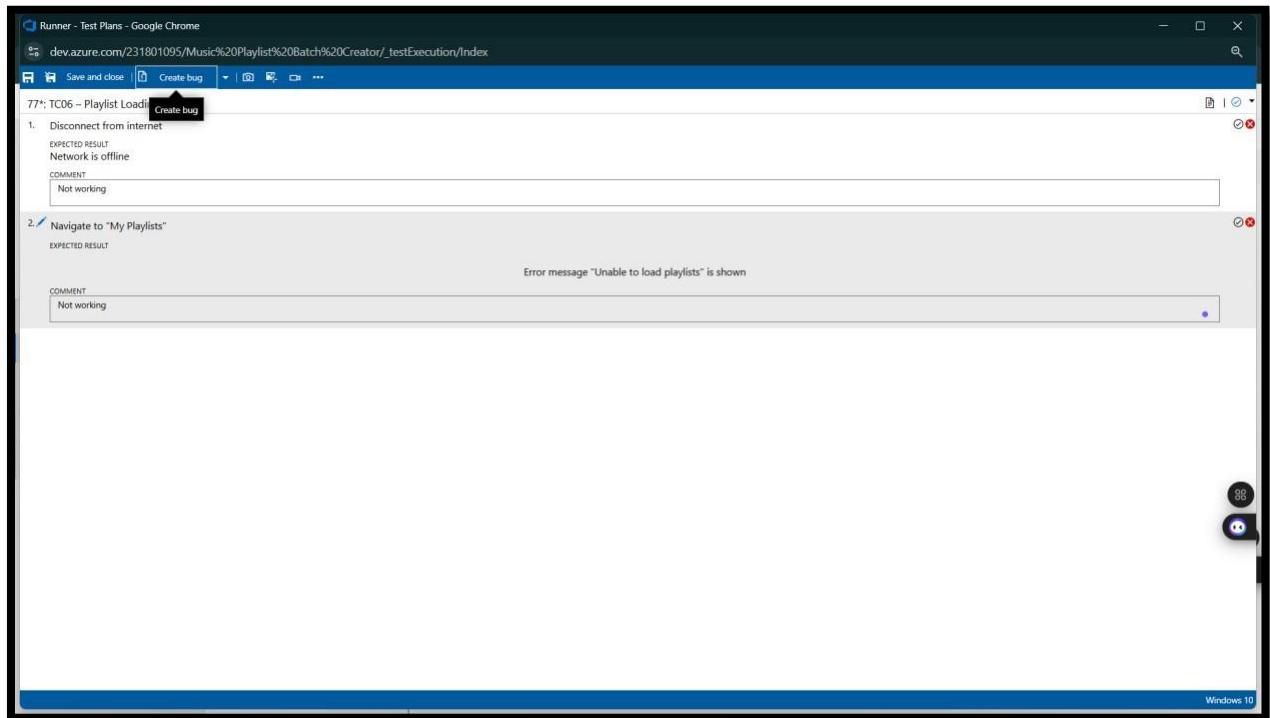
CS23432



6. Recording the test case



7. Creating the bug



The screenshot shows a Microsoft Edge browser window with multiple tabs open. The active tab is 'dev.azure.com/RECSCTeam01/Weather%20App/_workitems/edit/41/'. The page displays a bug work item titled 'BUG 41' with the ID '41 - Bug 01'. The work item details include:

- State:** New
- Area:** Weather App
- Iteration:** Weather App/Sprint 1
- Repro Steps:** (Empty)
- System Info:**

Browser - Name	Google Chrome 135
Browser - Language	en-US
Browser - Height	640
Browser - Width	1194
Browser - UserAgent	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/135.0.0.0 Safari/537.36
OS - Name	Windows NT 10.0; Win64 x64
OS - Architecture	x64_64
OS - ProcessorModel	Intel(R) Core(TM) i5-10210U CPU @ 1.60GHz
OS - NumberOfProcessors	8
Memory - Available	69557568
Memory - Capacity	835267296
Display - PixelsPerInchXAxis	192
Display - PixelsPerInchYAxis	156
Display - DevicePixelRatio	1.625
- Planning:**
 - Resolved Reason: (empty)
 - Story Points: (empty)
 - Priority: 2
 - Severity: 3 - Medium
 - Activity: (empty)
- Deployment:**
 - To track releases associated with this work item, go to [Releases](#) and turn on deployment status reporting. [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: (empty)
 - Remaining: (empty)
- Related Work:**
 - Add link: [Add an existing work item as a parent](#)

8. Test case results

The screenshot shows a Microsoft Edge browser window with multiple tabs open. The active tab is 'dev.azure.com/231801095/Music%20Playlist%20Batch%20Creator/_testPlans/execute?planId=84&suitId=87'. The page displays the results for the 'TS02 - View Playlists' test suite, which has 2 items:

- TC05 - View Playlist Page**: Passed (4m ago, Windows 10, Tester: Karthick S, Tester Notes: Mallu karthick Balaji ... Music)
- TC06 - Playlist Loading Failure**: Failed (Saturday, Windows 10, Tester: Karthick S, Tester Notes: Mallu karthick Balaji ... Music)

The left sidebar shows the Azure DevOps navigation menu with 'Test plans' selected. The 'Runs - Test Plans' tab is also visible at the top.

9. Test report summary

2116231501001

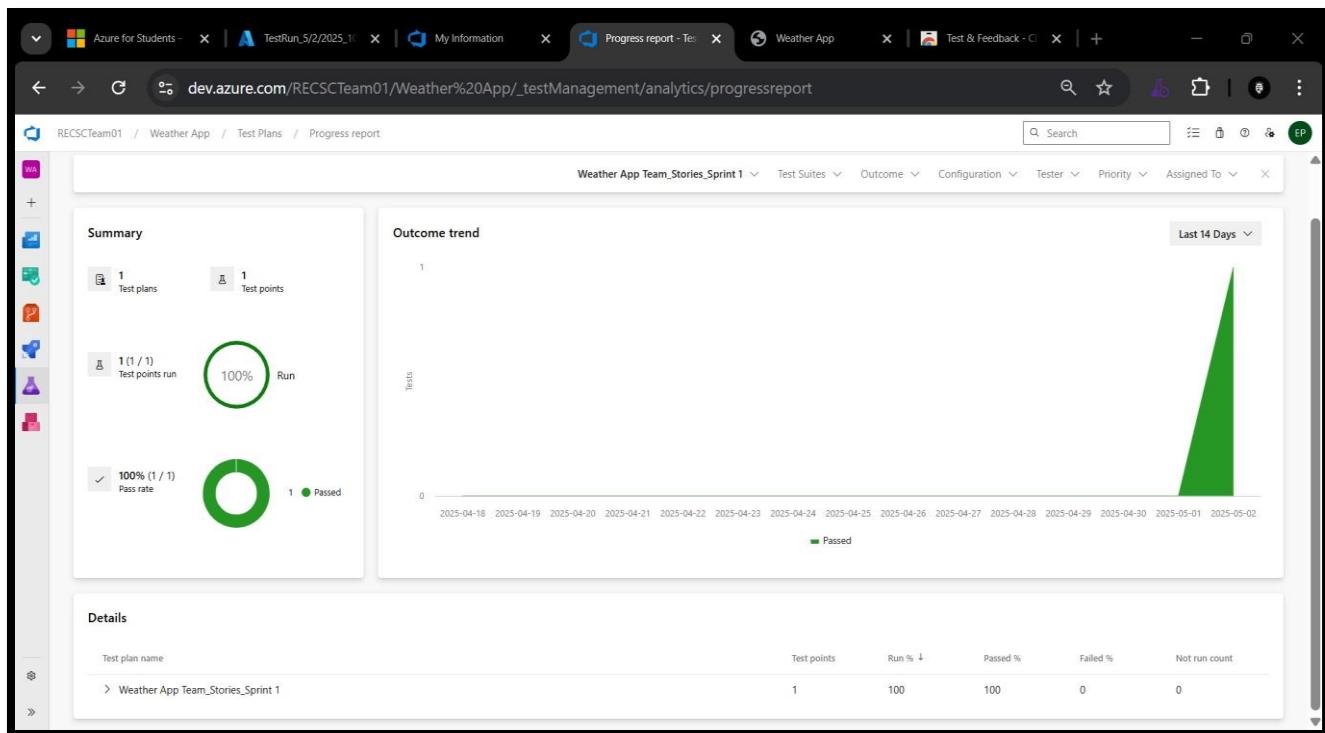
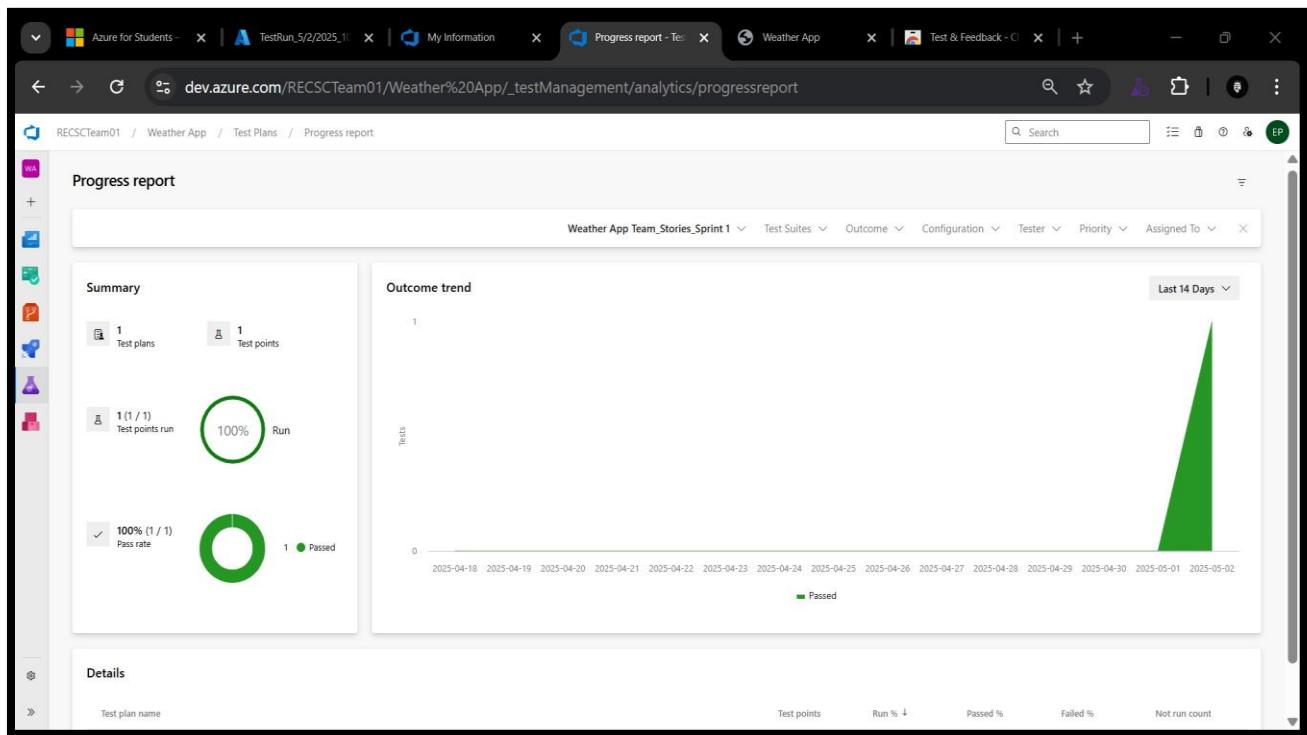
CS23432

The screenshot shows the Azure DevOps interface for a 'News Feed App' project. A bug work item titled 'BUG 203 BG 01 - Countries Drop down Not Available on the page' is selected. The 'State' dropdown is set to 'New'. The 'Repro Step' section contains three steps: 1. Passed (Title: Open Chrome Browser, Result: Passed, Expected Result: Browser should open in a new tab), 2. Passed (Title: Paste the URL (https://127.0.0.1/new.html), Result: Passed, Expected Result: App should load with new articles), and 3. Failed (Title: System should show a dropdown with list of countries, Result: Failed, Expected Result: System should show a dropdown with list of countries). The 'Planning' and 'Deployment' sections are visible on the right.

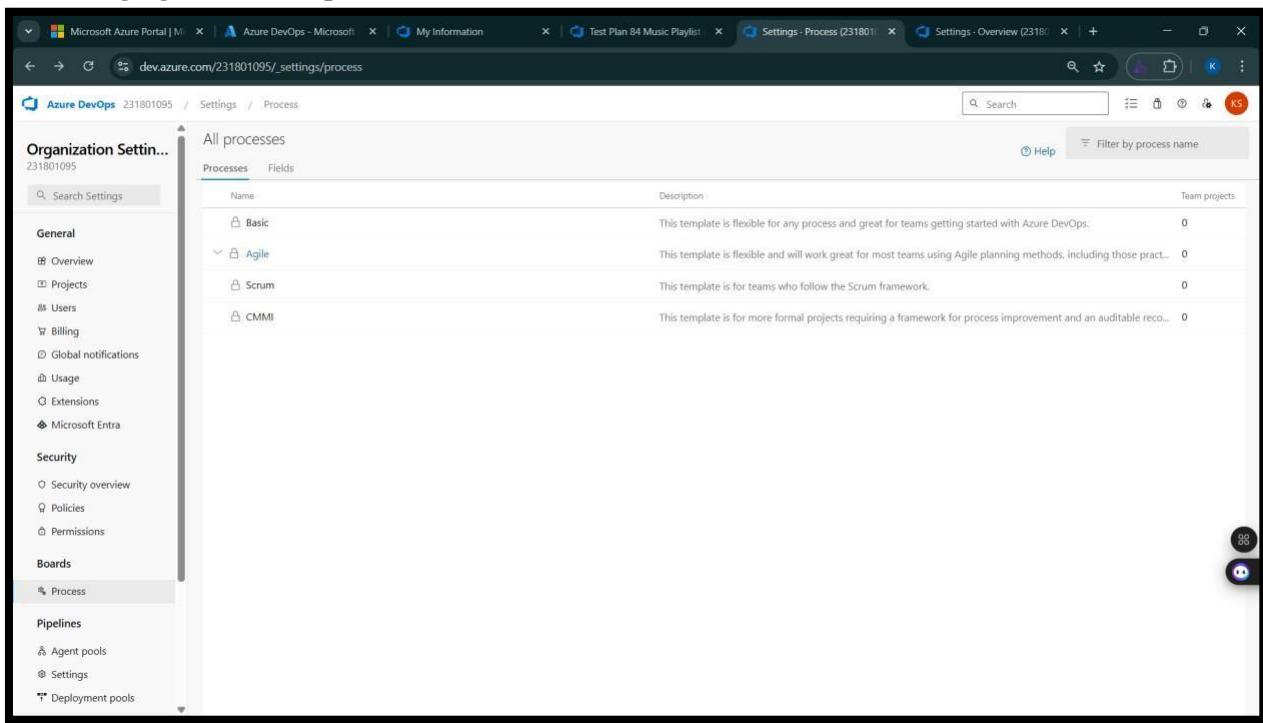
- Assigning bug to the developer and changing state

The screenshot shows the Azure DevOps interface for a 'Music Playlist Batch Creator' project. A test case titled 'Run 48 - TS02 - View Playlists (Manual) / TC06 - Playlist Loading Failure' is selected. The 'State' dropdown is set to 'New'. The 'Repro Steps' section contains two steps: 1. Failed (Title: Bug filed on 'TC06 - Playlist Loading Failure', Result: Failed, Expected Result: Network is offline, Comments: Page Not loading, Error message: "Unable to load playlists" is shown) and 2. Failed (Title: Navigate to "My Playlists", Result: Failed, Expected Result: Error message "Unable to load playlists" is shown). The 'Planning' and 'Deployment' sections are visible on the right.

10. Progress report



11. Changing the test template



The screenshot shows the 'All processes' section in the Azure DevOps Settings - Process page. The 'Processes' tab is selected. There are four listed: 'Basic', 'Agile', 'Scrum', and 'CMMI'. Each entry includes a description and a 'Team projects' count of 0.

Name	Description	Team projects
Basic	This template is flexible for any process and great for teams getting started with Azure DevOps.	0
Agile	This template is flexible and will work great for most teams using Agile planning methods, including those pract...	0
Scrum	This template is for teams who follow the Scrum framework.	0
CMMI	This template is for more formal projects requiring a framework for process improvement and an auditable reco...	0

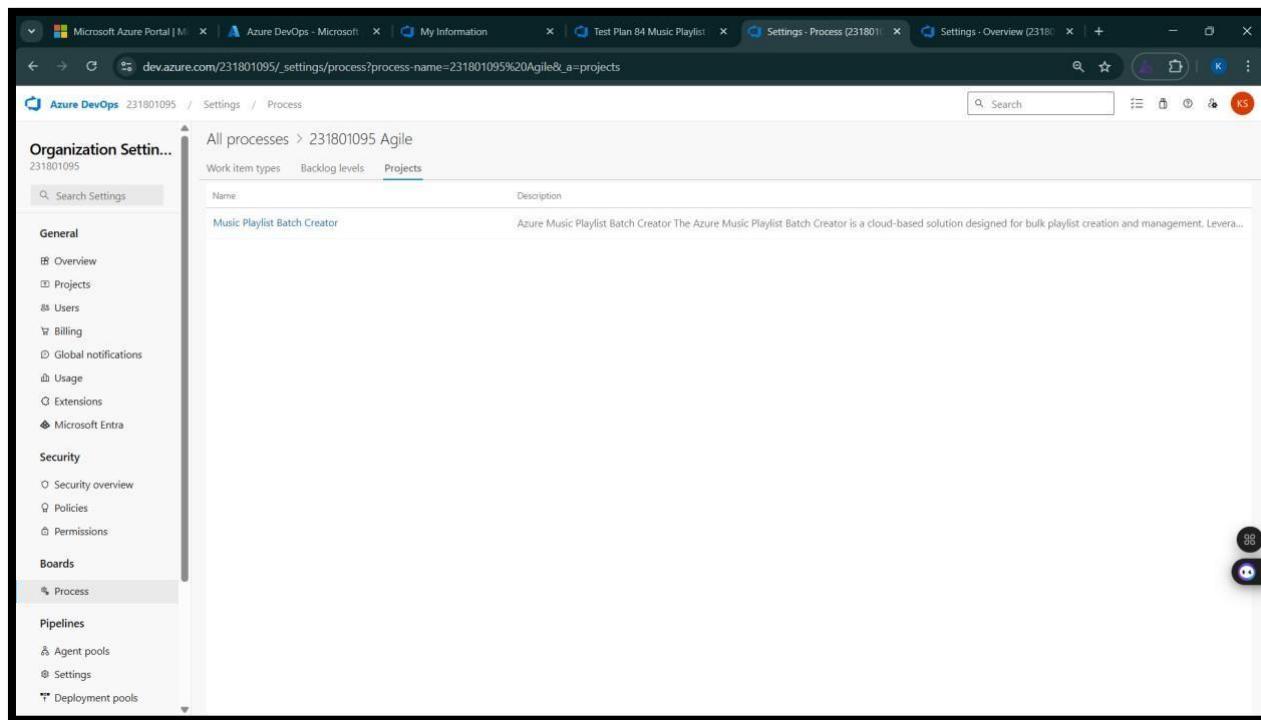
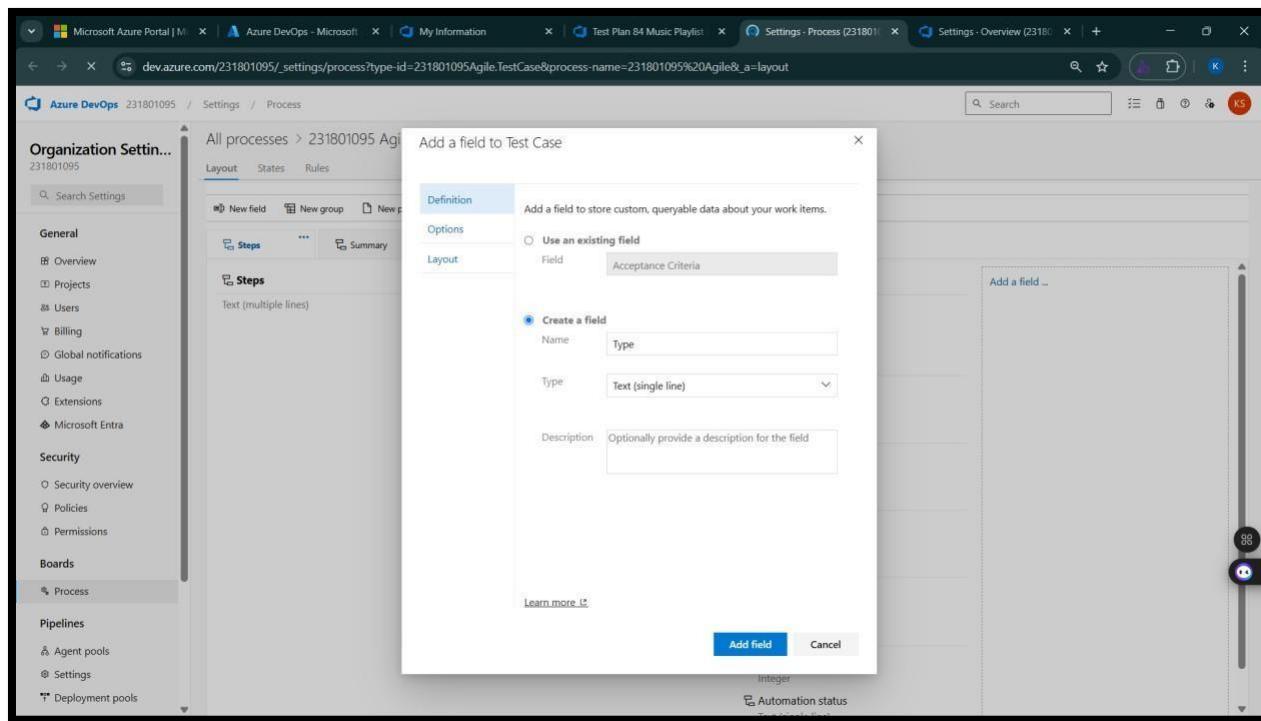
The screenshot shows the 'All processes' section of the Azure DevOps Settings - Process page. The 'Agile' template is selected, indicated by a blue background. The table lists the following process templates:

Name	Description	Team projects
Basic	This template is flexible for any process and great for teams getting started with Azure DevOps.	0
Agile	This template is flexible and will work great for most teams using Agile planning methods, including those pract...	0
Scrum	This template is for teams who follow the Scrum framework.	0
CMMI	This template is for more formal projects requiring a framework for process improvement and an auditable reco...	0

The screenshot shows the 'All processes' section of the Azure DevOps Settings - Process page. A new entry, '231801095 Agile (default)', has been added under the Agile category. The table now includes this entry along with the other templates:

Name	Description	Team projects
Basic	This template is flexible for any process and great for teams getting started with Azure DevOps.	0
Agile	This template is flexible and will work great for most teams using Agile planning methods, including those pract...	0
231801095 Agile (default)		1
Agile Plus		0
Scrum	This template is for teams who follow the Scrum framework.	0
CMMI	This template is for more formal projects requiring a framework for process improvement and an auditable reco...	0

12. View the new test case template



The screenshot shows the Azure DevOps Settings - Process page for a specific organization setting (231801095). The left sidebar lists various settings categories like General, Security, Boards, Pipelines, etc., with 'Process' selected. The main area displays the 'Test Case' section of the '231801095 Agile' process template. It includes tabs for 'Steps', 'Summary', and 'Associated Aut...'. A 'Custom' section is visible on the right, listing fields such as 'Recent test results', 'Deployment', 'Development', 'Related Work', and 'Status'. A search bar at the top right allows for filtering.

Result:

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

2116231501001

CS23432

EXP NO: 9
Create Epic, F

LOAD TESTING AND PERFORMANCE TESTING

Aim:

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

Load Testing

Steps to Create an Azure Load Testing Resource:

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

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

Load Testing

Azure for Students - TestRun_5/2/2025_10 | My Information | Bug 41 Bug 01 | Weather App | Test & Feedback -

portal.azure.com/?Microsoft_Azure_Education_correlationId=bc27d11b-6986-45c0-8ba4-292a1c12fd8c#view/Microsoft_Azu...

Microsoft Azure Search resources, services, and docs (G+/-) Copilot Home > Microsoft.CloudNativeTesting1746204463822 | Overview > Weather-App >

TestRun_5/2/2025_10:21:51 PM

Last updated by: 231501045@rajalakshmi.edu.in | Initiated on: 5/2/2025, 10:21 PM

View all test runs Stop Refresh Rerun Compare App components Configure metrics Download Copy artifacts Share ...

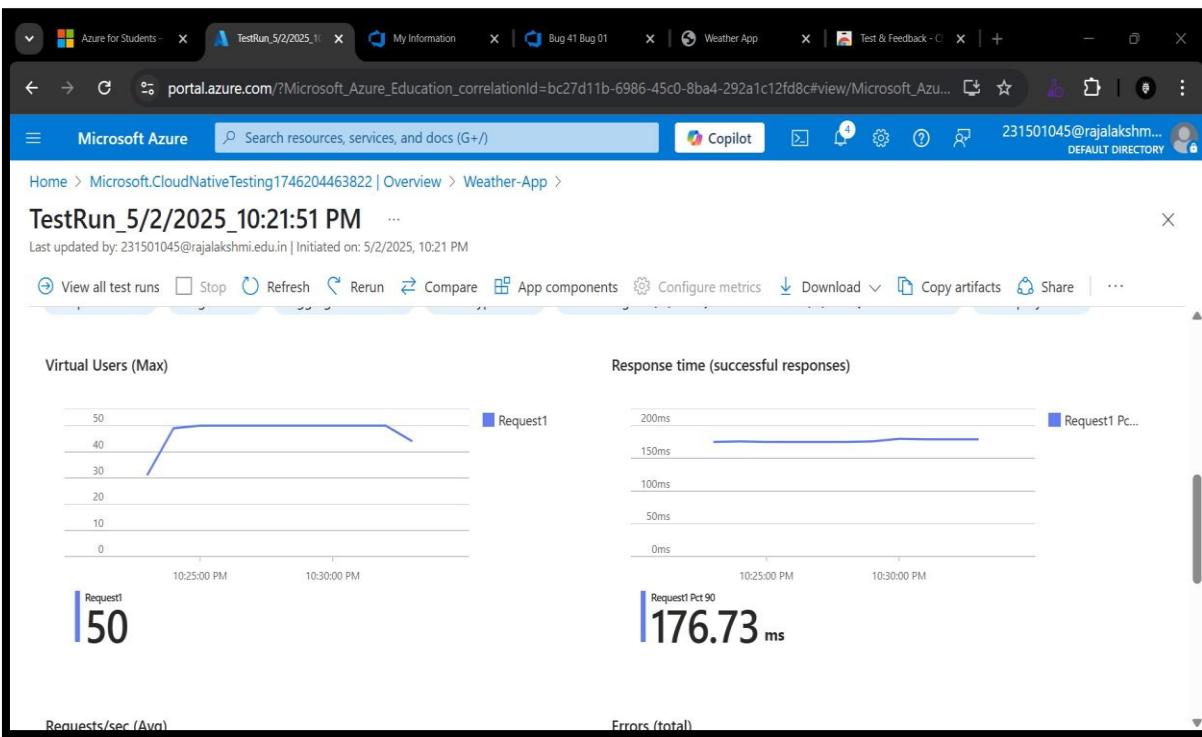
Test run details

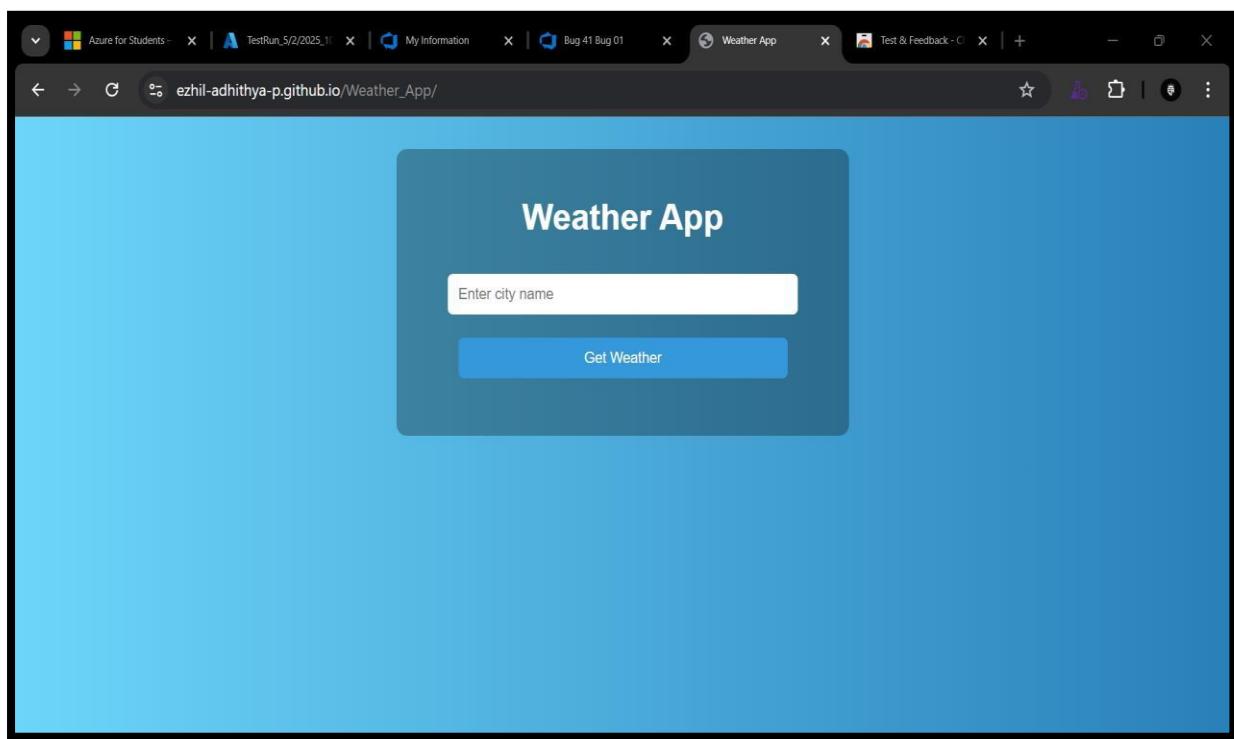
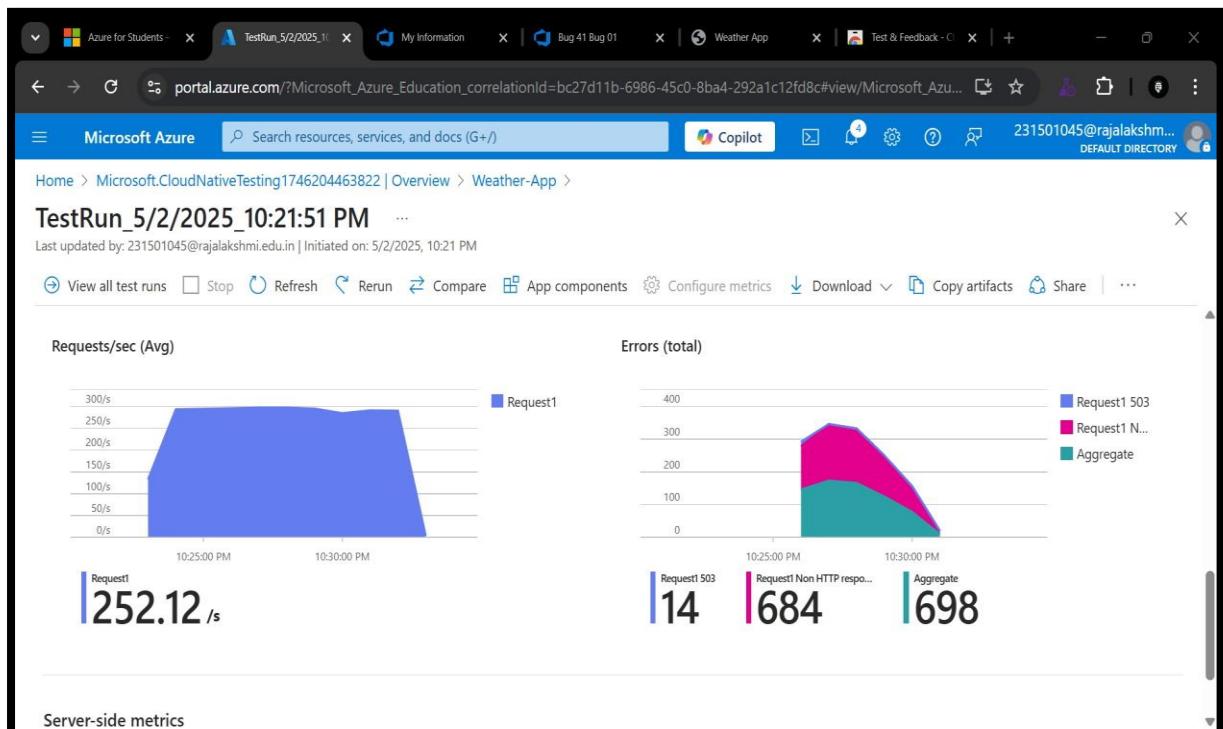
Start time	End time	Test run ID	Test type	Engine instances	Debug mode	Test result	Status
		13b3ac26-f67a-4d7...	URL	1	Enabled	Not Applicable	✓

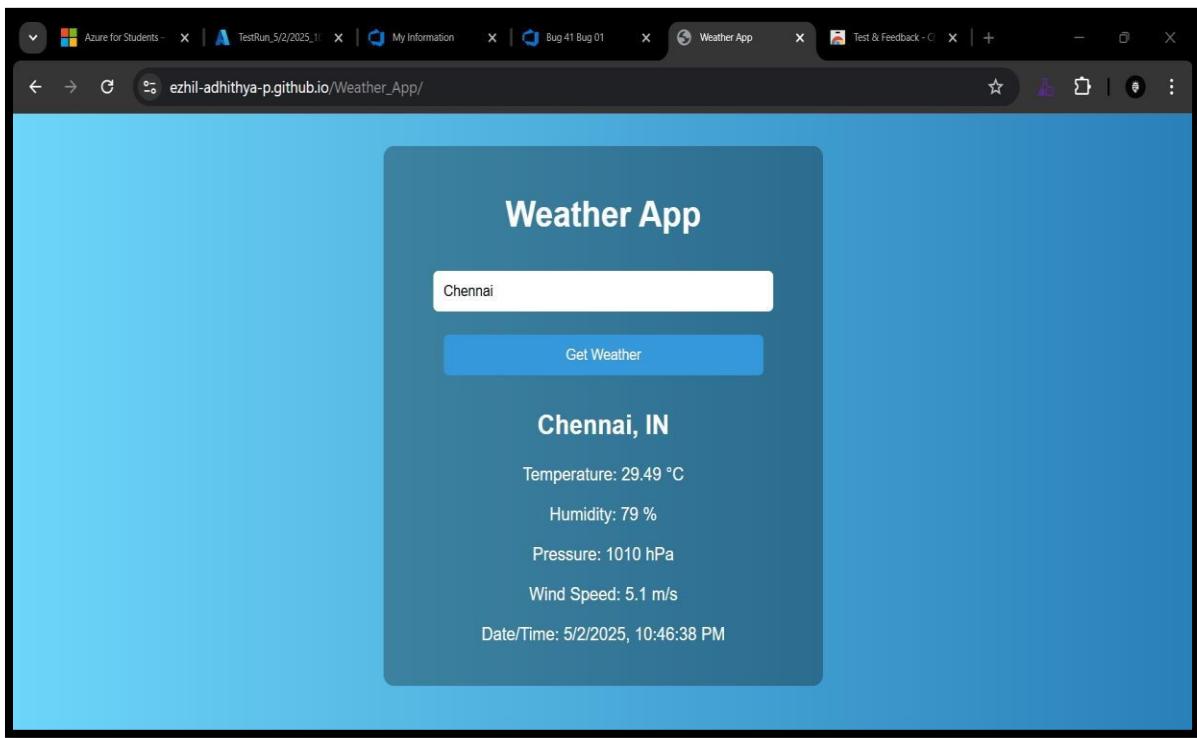
Load test results Engine health

Statistics

Load 166402 Total requests	Duration 10 mins	Response time 176.00 ms 90th percentile response time	Error percentage 0.42 % Aggregate requests which failed	Throughput 277.34 /s Request rate
----------------------------------	---------------------	---	---	---





**Result:**

Successfully created the Azure Load Testing resource and executed a load test to assess the performance of the specified endpoint.

2116231501001

CS23432

EXP NO: 10
Create Epic, F

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 Weather App Project.

GitHub Project Structure:

The screenshot shows a GitHub repository page for 'Weather_App'. The repository is public and has 2 commits. The main branch is 'main'. The commits are:

- Ezhil-Adhithya-P Create README.md (fe11030 · now) - 2 Commits
- README.md Create README.md now
- index.html Add files via upload yesterday
- script.js Add files via upload yesterday
- style.css Add files via upload yesterday

The repository has 0 stars, 1 watching, and 0 forks. There are no releases published.

The screenshot shows a GitHub repository page for 'Weather_App' at github.com/Ezhil-Adhithya-P/Weather_App/tree/main. The repository has one file, 'style.css', and a README file. The README contains project details, key features, and deployment information. The repository has no releases, packages, or deployments. The language usage chart shows JavaScript at 47.1%, CSS at 27.0%, and HTML at 25.9%.

Project Title: Weather App

Description:

This project is a beginner-friendly Weather Application developed using HTML, CSS, and JavaScript. The application enables users to enter a location and receive real-time weather data, including temperature, humidity, air quality index (AQI), pressure, wind speed, and date/time. It utilizes a third-party weather API to fetch accurate data and displays it in a clean, user-friendly interface.

Key features include:

- (i) Location-based weather search
- (ii) Display of essential weather parameters
- (iii) Error handling for invalid inputs or API failures
- (iv) Responsive and interactive UI
- (v) Optional enhancements like daily/hourly updates and weather alerts (if time permits)

The app is designed with beginners in mind and focuses on practical learning and API integration. It also emphasizes clean code structure and UI/UX best practices.

Releases: No releases published [Create a new release](#)

Packages: No packages published [Publish your first package](#)

Deployments: 1 [github-pages](#) yesterday

Languages: JavaScript 47.1%, CSS 27.0%, HTML 25.9%

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.