

EXP NO: 1

AZURE DEVOPS ENVIRONMENT SETUP

AIM

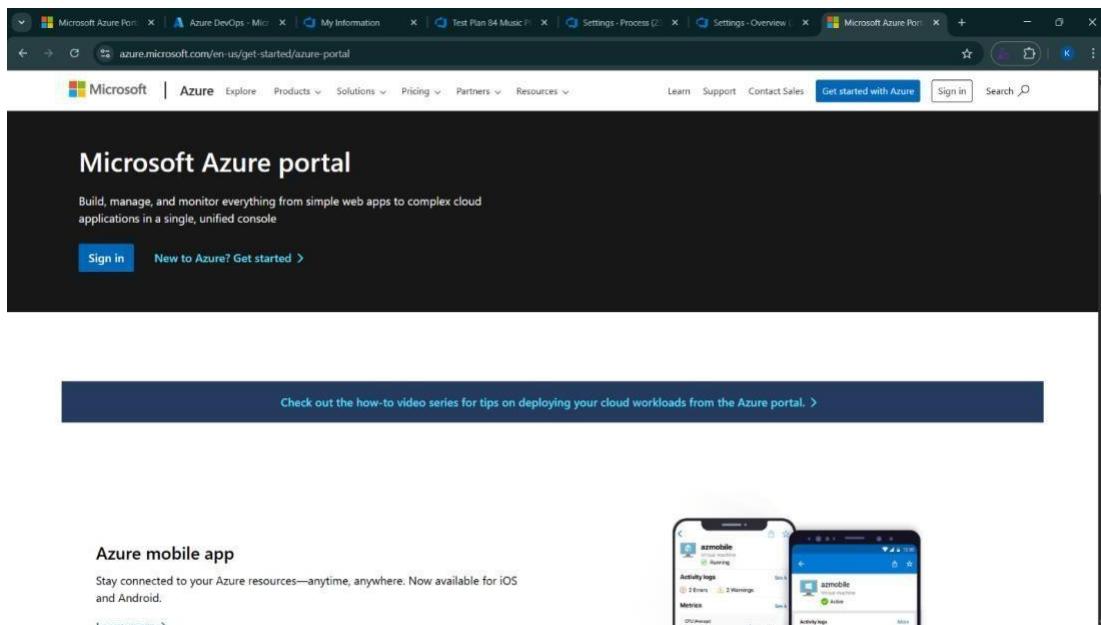
To set up and access the Azure DevOps environment by creating an organization through the Azure portal.

INSTALLATION

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

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>



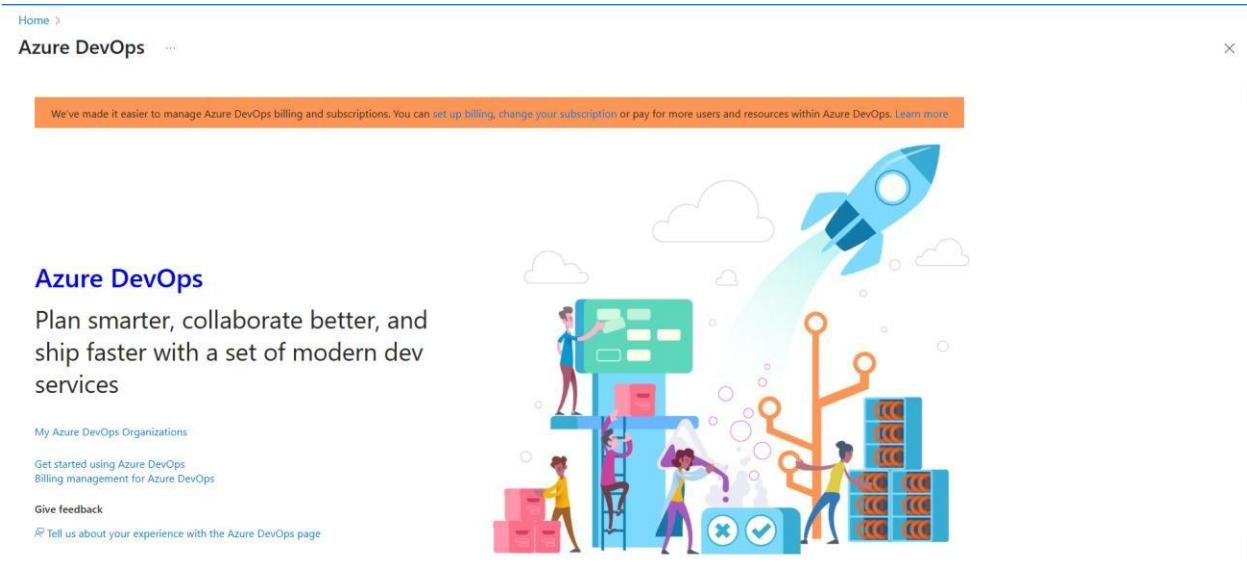
2. Azure home page

The screenshot shows the Microsoft Azure home page. At the top, there's a navigation bar with the Microsoft Azure logo, a search bar, and various icons for Copilot, notifications, and help. Below the navigation bar is the 'Azure services' section, which includes links for 'Create a resource', 'Azure DevOps organizations', 'Quickstart Center', 'Azure AI foundry', 'Kubernetes services', 'Virtual machines', 'App Services', 'Storage accounts', 'SQL databases', and 'More services'. Under the 'Resources' section, there are tabs for 'Recent' and 'Favorite', and a table with columns for 'Name', 'Type', and 'Last Viewed'. A message says 'No resources have been viewed recently' with a 'View all resources' button. At the bottom, there's a 'Navigate' section with links for 'Subscriptions', 'Resource groups', 'All resources', and 'Dashboard'.

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

The screenshot shows the Microsoft Azure home page again, but with the search bar at the top containing the text 'AZURE DEVOPS'. The search results are displayed below, showing 'Services' (with 'Azure DevOps organizations' highlighted), 'Marketplace' (with items like 'Azure Cosmos DB', 'Azure Database for MySQL servers', 'Azure Deployment Environments'), and 'Documentation' (with items like 'Billing overview - Azure DevOps', 'Deploying to Azure VMs using deployment groups in Azure Pipelines - Azure Pl...', 'Buy Azure DevOps for Cloud Solution Providers - Azure DevOps Services'). The rest of the interface is similar to the first screenshot, including the 'Azure services' section and 'Resources' table.

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.

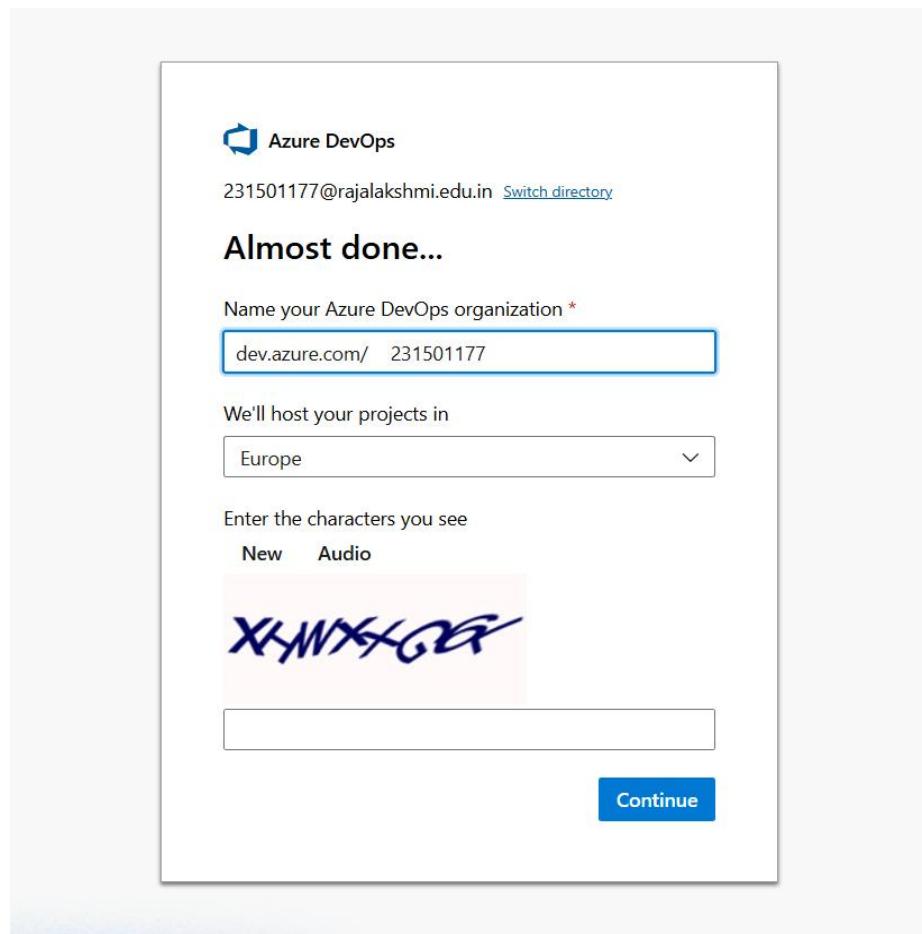
EXP NO: 2

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

- 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.
- 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.

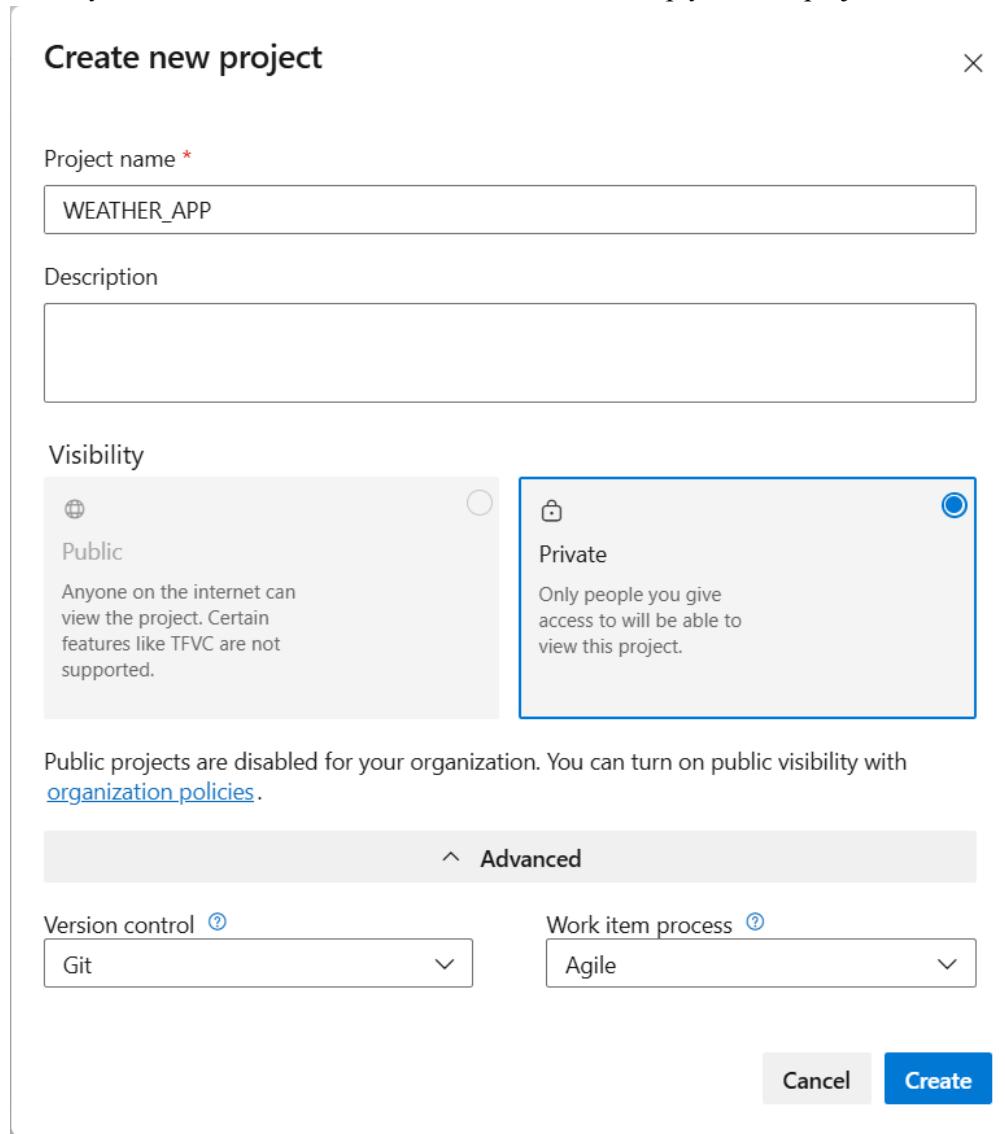
Private
Only people you give access to will be able to view this project.

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 interface. At the top, there's a blue header bar with the Microsoft logo on the left and the user's name 'UMESH SARATHY S K' and a 'Sign out' button on the right. Below the header, there's a large purple circular profile picture with the letters 'UK' in white. To the right of the profile picture, the user's name 'UMESH SARATHY S K' is displayed, along with a 'Edit profile' link and their email address '231501177@rajalakshmi.edu.in'. A dropdown menu labeled 'Microsoft account' is open, showing 'India' and the same email address. Below this, there's a section titled 'Visual Studio Dev Essentials' with a brief description and a 'Use your benefits' link.

Azure DevOps Organizations

[Create new organization](#)

> [dev.azure.com/231501177](#) (Owner)

▽ [dev.azure.com/UmeshSarathy](#) (Owner)

Projects



Weather App

[New project](#)

Actions

[Open in Visual Studio](#)

4. Project dashboard

The screenshot shows the Azure DevOps Project Summary page for the 'Weather App' project. The top navigation bar includes 'Azure DevOps', the organization name 'UmeshSarathy', the project name 'Weather App', and 'Overview' and 'Summary' links. On the far right, there are 'Search', 'Private', 'Invite', and a user icon. The main content area has a title 'Weather App' with a 'WA' icon. On the left, a sidebar lists project navigation options: Weather App (selected), Overview, Summary (highlighted in grey), Dashboards, Wiki, Boards, Repos, Pipelines, Test Plans, and Artifacts. At the bottom of the sidebar are 'Project settings' and a back arrow. The main content area features a 'About this project' section with a description of the WeatherNow application and its functionality. To the right, there are several summary cards: 'Project stats' (Period: Last 7 days) showing 11 work items created and 0 completed; 'Repos' showing 0 pull requests opened and 4 commits by 2 authors; 'Pipelines' showing a green circle at 50% completion with the text 'Builds succeeded'; and a 'Members' section showing 1 member. The overall interface is clean and modern, using a light grey color scheme with blue and purple highlights for important elements.

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.
- 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 Boards Backlog page. The left sidebar is titled "Weather App" and includes options like Overview, Boards, Work items, Backlogs, Sprints, Queries, Delivery Plans, Analytics views, Repos, Pipelines, Test Plans, Artifacts, and Project settings. The main area is titled "Weather App Team" and shows a backlog of work items. The backlog table has columns for Order, Work Item Type, Title, State, Effort, Business Area, and Tags. The backlog contains the following items:

Order	Work Item Type	Title	State	Effort	Business Area	Tags
1	Epic	Weather prediction	Active		Business	
	Feature	> Weather prediction for next one week	New		Business	
	Feature	> hourly weather prediction	New		Business	
2	Epic	Reminder	New		Business	
	Feature	> Hourly Notification with weather details	New		Business	
3	Epic	Real-Time Weather Forecasting	New		Business	
	Feature	> Displays Current weather details accurately	New		Business	

The screenshot shows the Microsoft account sign-in page. At the top right, there is a purple circular icon with the letters "UK". The page displays the user's name, "UMESH SARATHY S K", and email address, "231501177@rajalakshmi.edu.in". Below this, there are links for "My Microsoft account" and "Switch directory". On the left, there is a "Work Item" button and a "View" button. At the bottom, there is a "Sign in with a different account" link and a "Sign in" button with a user icon.

RESULT

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

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 interface for the 'Weather App' project. The left sidebar has 'Backlogs' selected. The main area displays a backlog board for the 'Weather App Team'. The backlog is organized into several items:

- 1 Epic:** Weather prediction (Active, Business)
- 1 Feature:** Weather prediction for next one week (New, Business)
- 1 User Story:** As a user, I want to access to weather details for next ... (New, Business)
 - 1 Task: Get the API key (New)
 - 1 Task: Connect the api key to the project (New)
 - 1 Task: Fetch the weather details using the api key (New)
- 1 Feature:** hourly weather prediction (New, Business)
- 1 User Story:** as a user, I want to access to weather details for next 5... (New, Business)
 - 1 Task: Get the api key (New)
 - 1 Task: Fetch the weather details using the api key (New)
- 2 Epic:** Remainder (New, Business)
 - 1 Feature:** Hourly Notification with weather details (New, Business)
 - 1 User Story:** As a user, I get the hourly notification about the weat... (New, Business)
 - 1 Task: Notify the User (New)

1. Fill in Epics

The screenshot shows the details of an epic work item titled 'EPIC 14 Weather prediction' created by 'UMESH SARATHY S K'. The work item has the following details:

- Description:** Click to add Description.
- Planning:**
 - Priority: 2
 - Risk
 - Effort
 - Business Value
 - Time Criticality
- 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.
- Related Work:** (empty)

EXP NO:

2.Fill in Features

FEATURE 15

15 Weather prediction for next one week

No one selected 0 Comments Add Tag

State: New Area: Weather App Iteration: Weather App\Sprint1

Updated by UMESH SARATHY S K: Apr 7

Description: Click to add Description.

Planning: Priority 2, Risk

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.

Related Work: [Collapsed]

3.Fill in User Story Details

USER STORY 16

16 As a user, I want to access to weather details for next one week, so that I can check the weather details

No one selected 0 Comments Add Tag

State: New Area: Weather App Iteration: Weather App\Sprint1

Updated by UMESH SARATHY S K: Apr 7

Description: Click to add Description.

Planning: Story Points, Priority 2, Risk

Classification: Value area: Business

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.

Related Work: [Collapsed]

RESULT

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

4

SPRINT PLANNING

AIM

To assign user story to specific sprint for the Weather app.

Sprint Planning Sprint 1

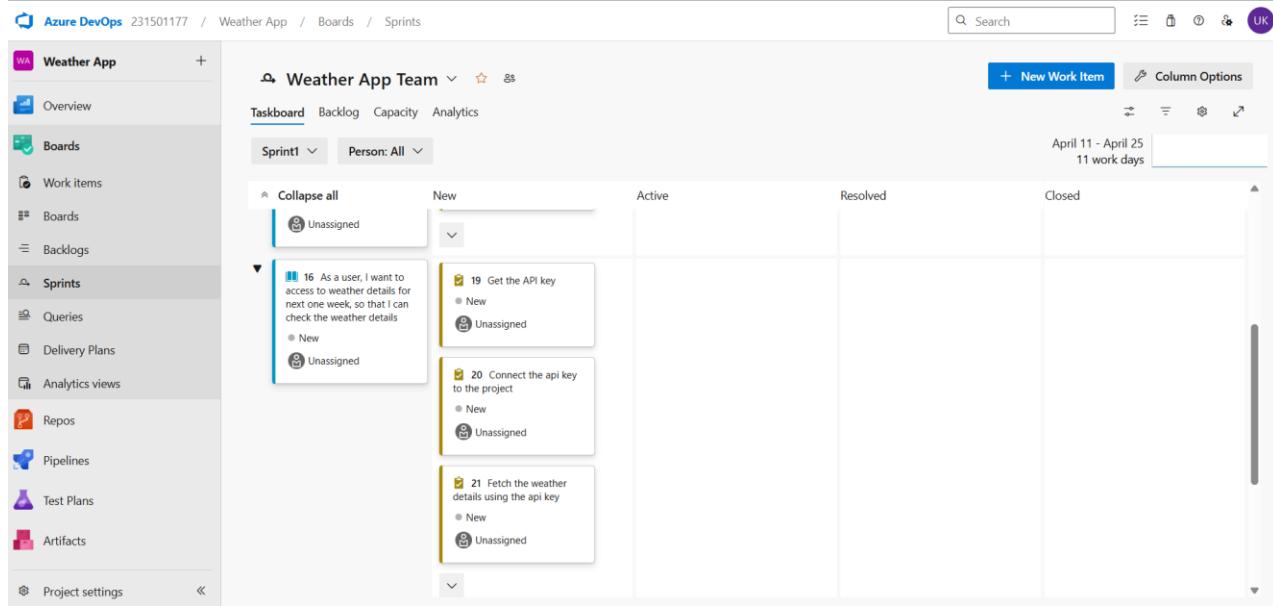
The screenshot shows the Azure DevOps Taskboard for the 'Weather App' project under the 'Sprint 1' column. The board is organized into columns: New, Active, Resolved, and Closed. There are four user stories listed:

- User Story 10: As a user, they visit the webpage, so that they can access the current weather details. Status: New, Unassigned.
- User Story 25: Get the API key. Status: New, Unassigned.
- User Story 26: Fetch the weather details using the API key. Status: New, Unassigned.
- User Story 13: As a user, I get the hourly notification about the weather details, so that I get reminded. Status: New, Unassigned.
- User Story 24: Notify the User. Status: New, Unassigned.

The top navigation bar shows the project name 'Weather App' and the sprint duration 'April 11 - April 25 11 work days'. The left sidebar includes links for Overview, Boards, Work items, Backlogs, Sprints (selected), Queries, Delivery Plans, Analytics views, Repos, Pipelines, Test Plans, Artifacts, and Project settings.

EXP NO:

Sprint 2

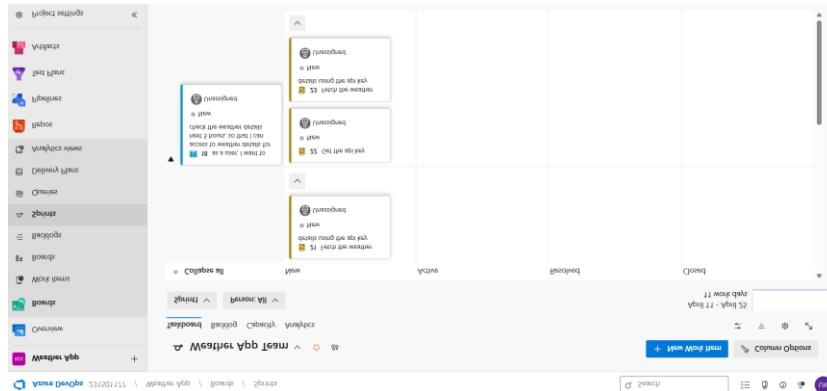


The screenshot shows the Azure DevOps Boards interface for the 'Weather App' project. The left sidebar is collapsed, showing options like Overview, Boards, Work items, Boards, Backlogs, Sprints, Queries, Delivery Plans, Analytics views, Repos, Pipelines, Test Plans, Artifacts, and Project settings. The main area is titled 'Weather App Team' and shows the 'Taskboard' view for 'Sprint1'. The backlog is organized into columns: New, Active, Resolved, and Closed. The 'New' column contains four items, each with a checkbox status and a brief description:

- 16 As a user, I want to access to weather details for next one week, so that I can check the weather details
- 19 Get the API key
- 20 Connect the api key to the project
- 21 Fetch the weather details using the api key

Each item has a 'New' radio button and an 'Unassigned' icon. The 'Active' column is currently empty. The 'Resolved' and 'Closed' columns also have no items.

Sprint 3



The screenshot shows the Azure DevOps Boards interface for the 'Weather App' project, similar to the previous one but with more items in the backlog. The left sidebar is collapsed, showing the same set of options. The main area is titled 'Weather App Team' and shows the 'Taskboard' view for 'Sprint1'. The backlog is organized into columns: New, Active, Resolved, and Closed. The 'New' column contains several items, each with a checkbox status and a brief description. Some items have additional details or attachments shown in a tooltip when hovering over them. The 'Active' column is currently empty. The 'Resolved' and 'Closed' columns also have no items.

RESULT

The Sprints have been created for Weather app.

5

POKER ESTIMATION

AIM

Create Poker Estimation for the user stories – Online Quiz System.

Poker Estimation

USER STORY 13

13 As a user, I get the hourly notification about the weather details, so that I get reminded

No one selected 0 Comments Add Tag

Save and Close Follow Updated by UMESH SARATHY S K Apr 7

State: New Area: Weather App
Reason: New Iteration: Weather App\Sprint1

Description: Click to add Description.

Planning: Story Points: 2, Priority: 2, Risk: Low

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](#)

Classification: Value area: Business

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.

Related Work

RESULT

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

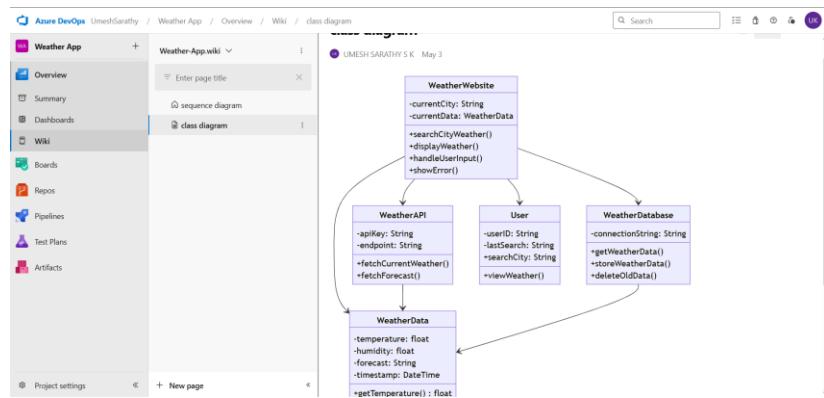
EXP NO: 6

DESIGNING CLASS AND SEQUENCE DIAGRAMS FOR PROJECT ARCHITECTURE

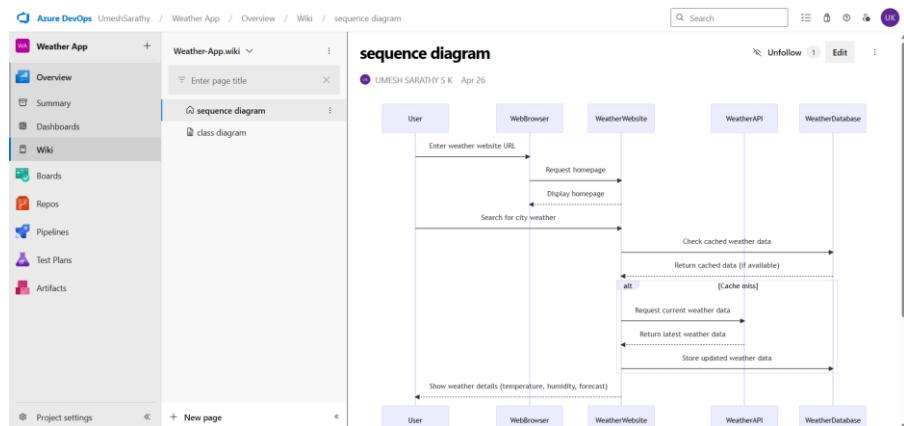
AIM

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 Weather App.

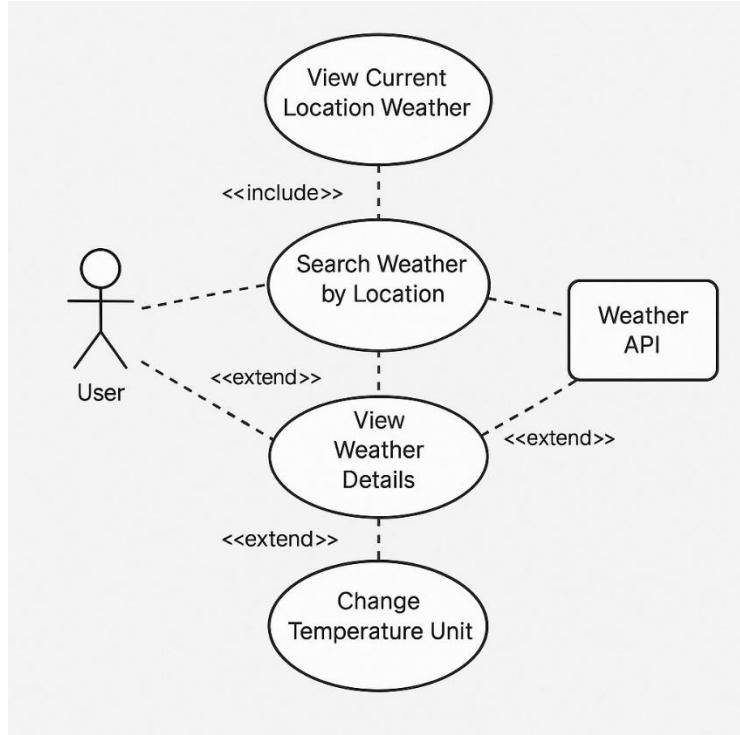
EXP NO: 7

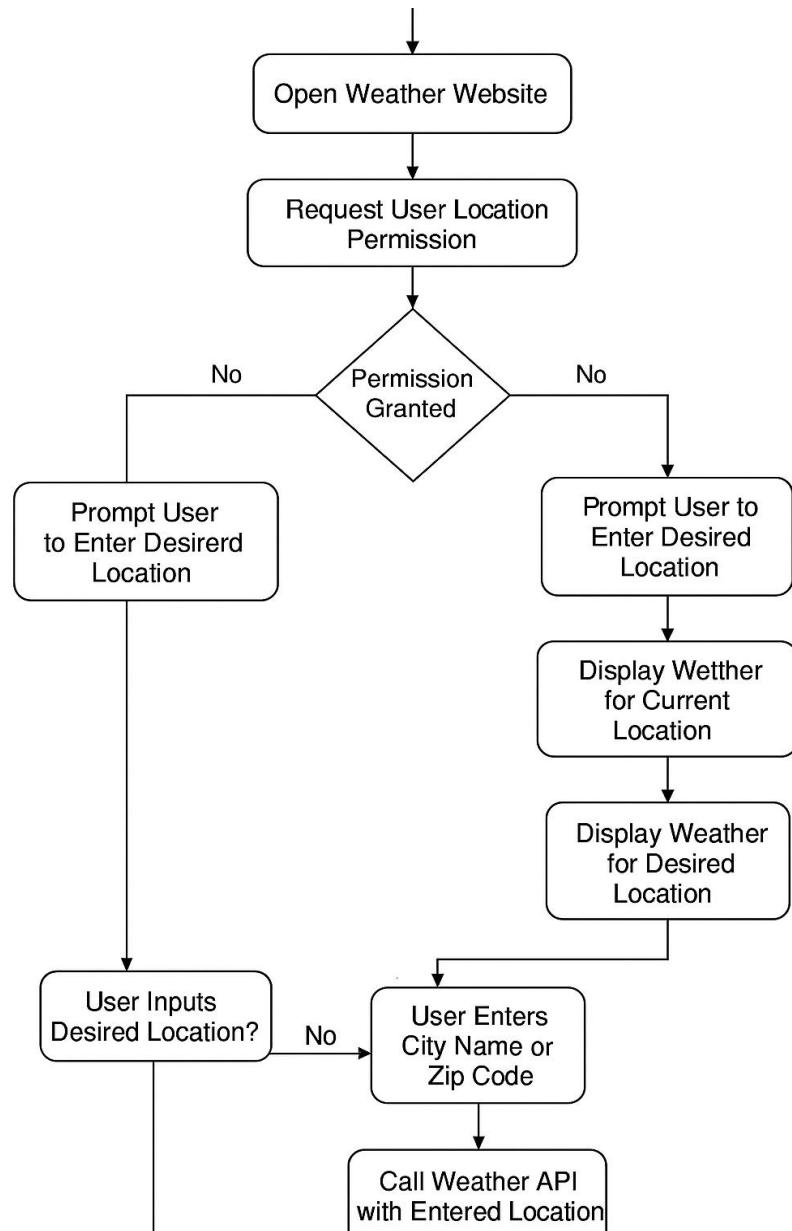
DESIGNING USE-CASE AND ACTIVITY DIAGRAMS FOR PROJECT STRUCTURE

AIM

To Design an Use-Case Diagram and Activity Diagram for the given Project.

7A. Use-Case diagram





RESULT

The Use-Case Diagram and Activity Diagram is designed Successfully for the Weather App



EXP NO:8

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 Login
-

2. Define User Interactions

- Each test case simulates a real user behaviour (e.g., logging in, submitting quizzes, viewing results)

3. Design Happy Path Test Cases

- Focused on validating that all core functionalities work correctly under normal conditions
- Example: Player registers and logs in, submits quizzes and views results

4. Design Error Path Test Cases

- Simulate invalid inputs, system issues or failed actions to ensure proper error handling.
- Example: Login with invalid credentials, submission without attachments, unauthorized access to admin panel.

5. Break Down Steps and Expected Results

- Each test case includes a clear sequence of actions and expected results. • Ensures both manual testers and automation tools can follow the process easily.

6. Use Clear Naming and IDs

- Test cases are uniquely identified (e.g., TC01 – Valid Login, TC03 – Invalid Password).
- Facilities easy mapping to features and tracking in Azure DevOps.

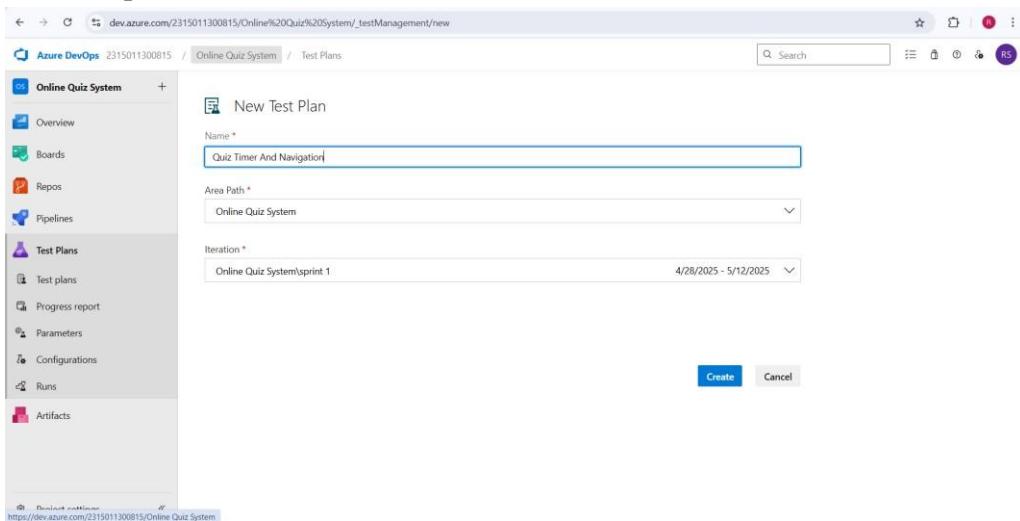
7. Separate Test

- Grouped by functionality such as:
 - Login and Registration
 - Quizzes Submission
 - Viewing Results
 - Admin Functions
- Improves organization and enables focused execution in Azure DevOps.

8. Prioritize and Review

- High-priority assigned to critical workflows like login, quizzes and results.
- Reviewed for completeness, accuracy and alignment with user stories and features definition.

1. New test plan



2. Test suite

The screenshot shows the Azure DevOps Test Plans interface. On the left, the navigation pane is visible with options like Overview, Boards, Repos, Pipelines, Test Plans, Test plans, Progress report, Parameters, Configurations, Runs, and Artifacts. The 'Test Plans' section is selected. In the center, a test suite named 'Weather App1' is displayed under 'Test Suites'. One of the test cases, 'Weather data for desired location (ID: 9)', is currently being viewed. This test case has two test points: 'Entered valid city name' (Outcome: Passed, Order: 1, Test Case Id: 10, Configuration: Windows) and 'Entered Invalid cityname' (Outcome: Failed, Order: 2, Test Case Id: 11, Configuration: Windows). A 'Run for web application' button is also present.

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

ID	User Type	User Story
US01	User	As a user, I want to see the weather for my current location so I can plan my day.
US02	User	As a user, I want to search for weather in any city or region so I can check conditions elsewhere.
US03	User	As a user, I want clear weather details like temperature, condition, humidity, and wind speed.
US04	User	As a user, I want to be alerted if location access is denied so I can manually search weather.
US05	User	As a user, I want the website to load quickly and work on mobile devices.

Test Suites and Test Cases

Test Suite: TS01 – Location Access & Weather Fetch (ID: WS01)

Test Case ID	Name	Type	Actions	Expected Results
TC01	Grant Location Access	Happy Path	Open site → Allow location access	Website displays weather for current location
TC02	Deny Location Access	Negative Path	Open site → Deny location access	Website prompts user to search manually
TC03	Fetch Weather by Location	Happy Path	Allow location access → Site fetches data	Weather details are displayed correctly
TC04	API Failure	Error Path	Simulate API failure	Show "Unable to fetch weather data" message

Test Suite: TS02 – Manual Location Search (ID: WS02)

Test Case ID	Name	Type	Actions	Expected Results
TC05	Search Valid City	Happy Path	Enter a valid city name → Click "Search"	Weather data for the city is displayed
TC06	Search Invalid City	Error Path	Enter random/invalid string → Click "Search"	Show "City not found" or error message
TC07	Empty Input	Negative Path	Leave search field empty → Click "Search"	Display warning: "Please enter a location"

Test Suite: TS03 – UI & Responsiveness (ID: WS03)

Test Case ID	Name	Type	Actions	Expected Results
TC08	Responsive Layout	Happy Path	Open website on phone/tablet	Layout adjusts correctly for screen size

Test Case ID	Name	Type	Actions	Expected Results
TC09	Clear Weather Display	Happy Path	Load weather data	Display includes temperature, humidity, wind, and condition
TC10	Loading Feedback	Happy Path	While weather is loading	Show a loading spinner or message

Test Suite: TS04 – Edge Cases & Errors (ID: WS04)

Test Case ID	Name	Type	Actions	Expected Results
TC11	Slow Network	Error Path	Simulate slow network	Show timeout or loading feedback
TC12	Special Characters in Search	Error Path	Input: !@#%^&*() → Click "Search"	Show invalid input or no result
TC13	Reload Page	Happy Path	Refresh after loading data	Weather remains visible or is re-fetched

Test Cases

Azure DevOps UmeshSarathy / Weather App / Test Plans / Weather App1

Weather App1

May
15 -
May
22
100% run, 66% passed. [View report](#)

Test Suites

Test Cases (2 items)

	Order	Test Case Id	Assigned To	State
<input type="checkbox"/> Title	1	10	UMESH SARATHY	Design
<input type="checkbox"/> Entered valid city name	2	11	UMESH SARATHY	Design
<input type="checkbox"/> Entered Invalid cityname				

Project settings https://dev.azure.com/UmeshSarathy/Weather App/_testPlans/define?planId=1&suiteId=9

TEST CASE 10

10 Entered valid city name

UMESH SARATHY S K 0 Comments Add Tag

State Design Area Weather App
Reason New Iteration Weather App

Updated by UMESH SARATHY S K: Thursday

Steps Summary Associated Automation

Steps

Steps	Action	Expected result	Attachment
1.	enter city name in search box		
2.	press enter		
3.	weather data displayed properly		

Click or type here to add a step

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

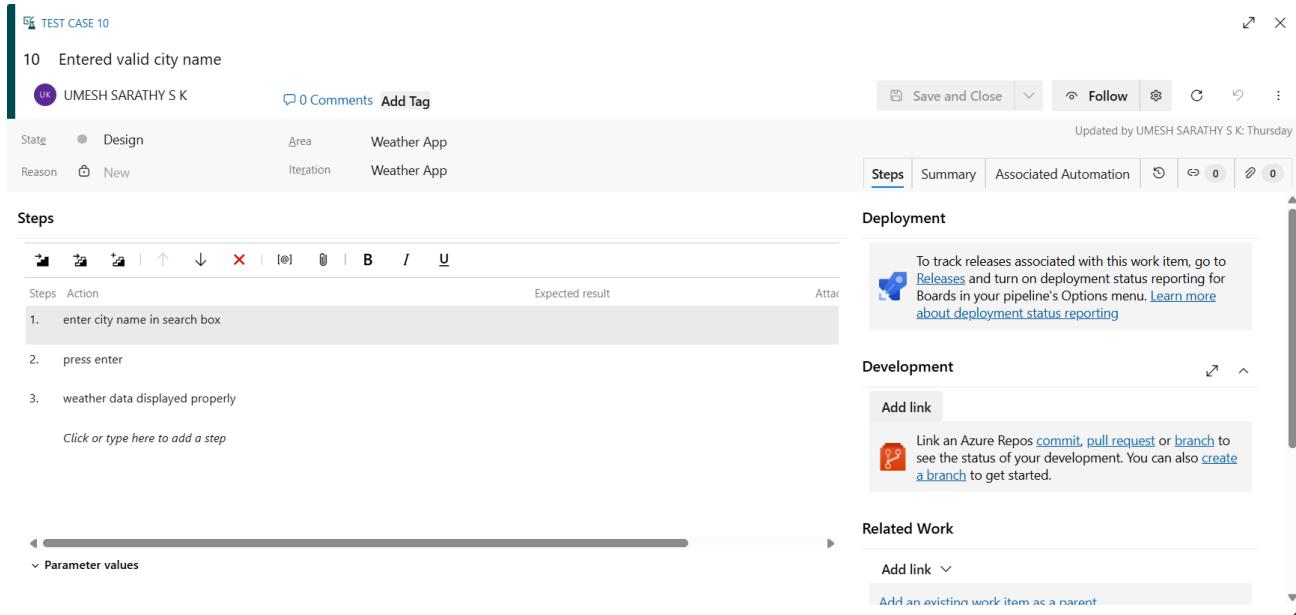
Add link

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.

Related Work

Add link

Add an existing work item as a parent



Azure DevOps - Microsoft | My Information | Test Plan 1 Weather App | Projects - Home | Signup | dev.azure.com/UmeshSarathy/Weather%20App/_testPlans/define?planId=18&suitId=9

Azure DevOps UmeshSarathy / Weather App / Test Plans / Weather App1

Search

Weather App

- Overview
- Boards
- Repos
- Pipelines
- Test Plans**
- Test plans
- Progress report
- Parameters
- Configurations
- Runs
- Artifacts

Test plans

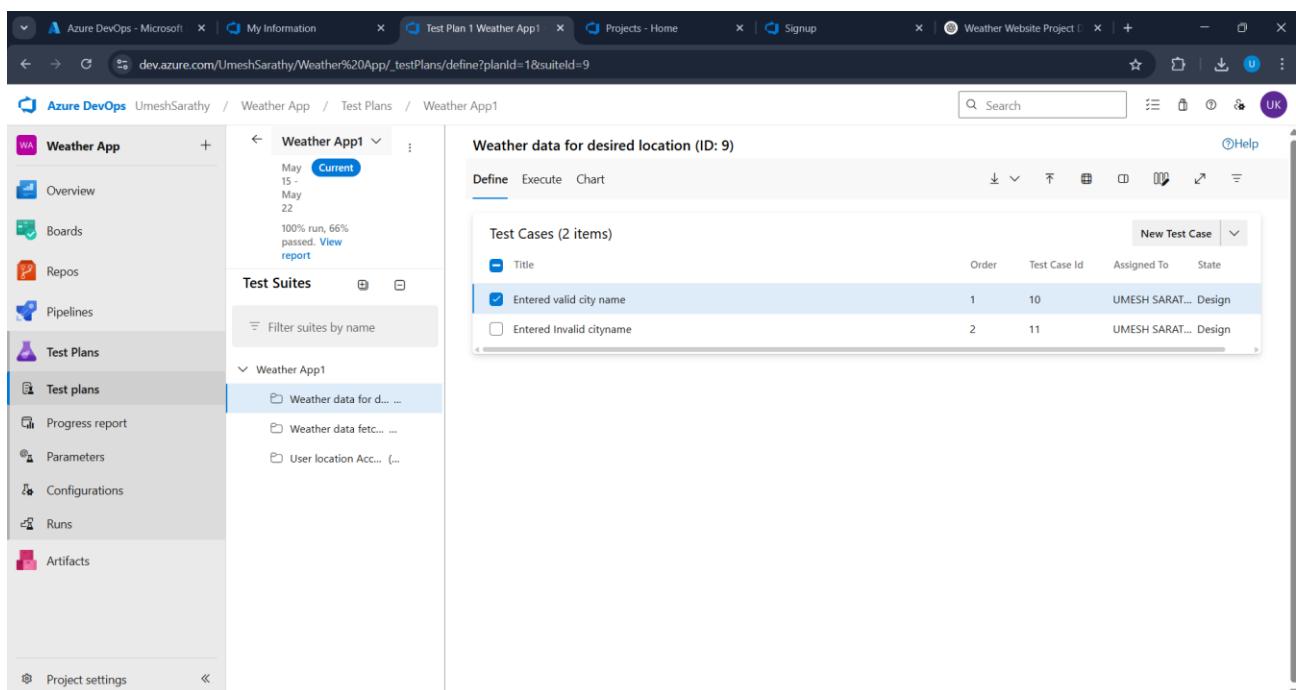
Test Suites

Filter suites by name

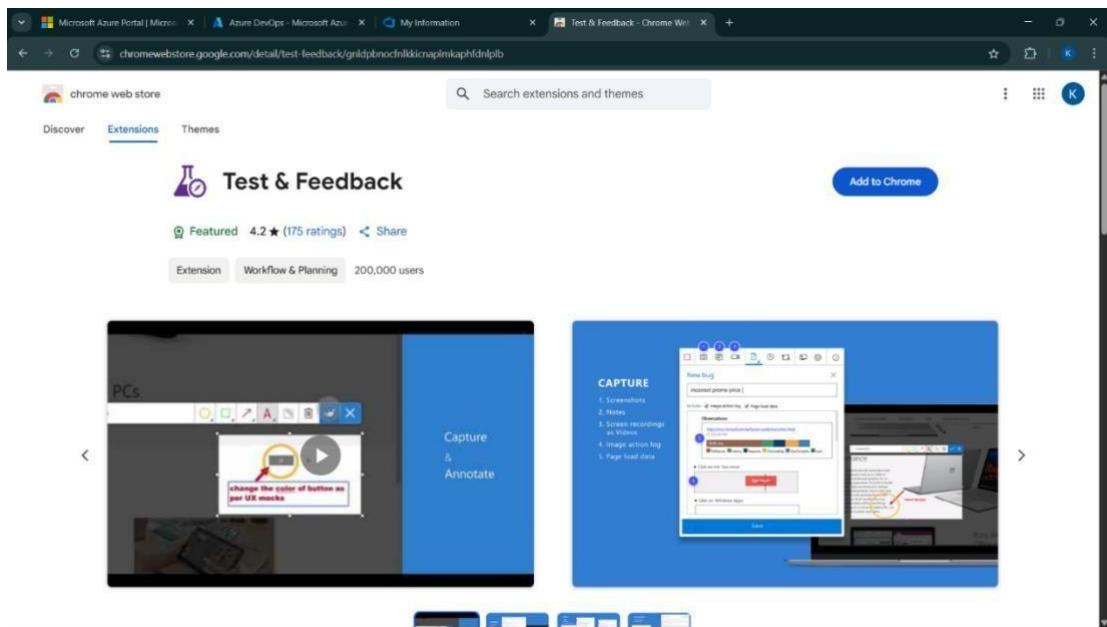
Weather App1

- Weather data for desired location (ID: 9)
- Test Cases (2 items)

Title	Order	Test Case Id	Assigned To	State
Entered valid city name	1	10	UMESH SARATHY S K	Design
Entered Invalid cityname	2	11	UMESH SARATHY S K	Design

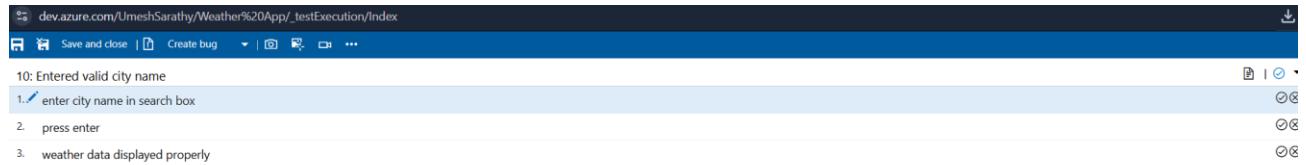


4. Installation of test

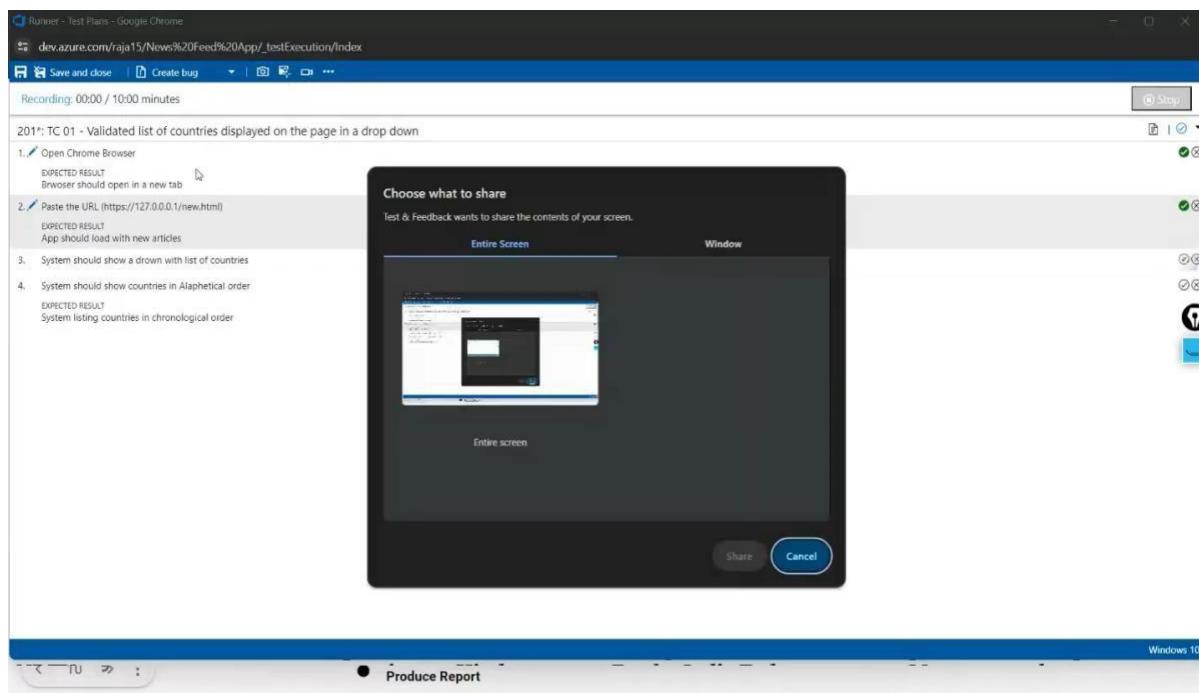


5. Running the test cases

A screenshot of the Azure DevOps Test Plans interface. On the left, a navigation sidebar shows 'Weather App' under 'Test plans'. The main area displays a 'Test Suites' section with 'Weather App1' selected. Under 'Test Cases (2 items)', there are two entries: 'Entered valid city name' (marked with a checkmark) and 'Entered Invalid cityname'. To the right of the table is a context menu with options like 'New Test Case', 'Open test case', 'Remove', 'Edit test case(s) in grid', 'Edit test case(s)', 'Assign configuration', 'Copy test case(s)', 'Export test case(s) to CSV', and 'Export test case(s) to XLSX'. The URL in the browser bar is 'dev.azure.com/UmeshSarathy/Weather%20App/_testPlans/define?planId=1&suiteId=9'.



6.Recording the test case



7.Creating the bug

The screenshot shows the 'New Bug' creation interface. At the top, there's a title bar with a red 'NEW BUG *' button, a search bar containing 'user should not login with invalid details', and a 'Save & Close' button. Below the title bar, the bug details are listed:

- Unassigned**
- 0 comments**
- Add tag**
- State: New**
- Reason: New**
- Area: Online Quiz System**
- Iteration: Online Quiz System\sprint 1**

The main area is divided into sections:

- Repro Steps:** A table showing steps 1 through 5, each with a result of 'None'. Step 1: 'Navigate to the login page.' Step 2: 'enter a valid username/email.' Step 3: 'enter valid password.' Step 4: 'Click the login button.' Step 5: 'None'.
- Planning:** Fields for Resolved Reason (locked), Story Points, Priority (2), Severity (3 - Medium), and Activity.
- Deployment:** A note about tracking releases associated with this work item, linking to Releases and Boards.
- Development:** A note about linking to Azure Repos commit or branch for development status.
- Effort (Hours):** Fields for Original Estimate and Remaining.
- Related Work:** A section for adding links to existing work items.

The screenshot shows the 'New Bug' creation interface with the same basic structure as the first one. The bug details are identical. The main area includes a 'System Info' section with a table of system specifications:

Browser - Name	Google Chrome 136
Browser - Language	en-US
Browser - Height	816
Browser - Width	1536
Browser - User agent	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/136.0.0.0 Safari/537.36
Operating system - Name	Windows NT 10.0; Win64; x64
Operating system - Architecture	x86_64
Operating system - Processor model	12th Gen Intel(R) Core(TM) i5-12450H
Operating system - Number of processors	12
Memory - Available	6998618112
Memory - Capacity	16831893504
Display - Pixels per inch (X axis)	120
Display - Pixels per inch (Y axis)	120

On the right side, there are notes about 'Found in Build' and 'Integrated in Build'.

8. Test case results

The screenshot shows the Azure DevOps Test Plans interface. On the left, the navigation pane is open with 'Weather App' selected. Under 'Test plans', 'Test suites' is expanded, showing 'Weather data for desired location (ID: 9)'. This suite has two test points: 'Entered valid city name' (Passed) and 'Entered Invalid cityname' (Not run). To the right, a modal window titled 'Test Case Results' displays the outcome of the first test point. The table shows:

Outcome	TimeSta...	Configuration	Run by	Tester	Test
Passed	Thursd...	Windows 10	UMESH SARAT...	UMESH SARAT...	Weather data for d...

Below the table, a link 'Open execution history for current test point' is visible.

9. Test report summary

The screenshot shows a bug report titled 'User should not enter invalid details'. The report details are as follows:

- State:** New
- Area:** Online Quiz System
- Reason:** New
- Iteration:** Online Quiz System\sprint 1

Repro Steps:

5/17/2025 11:09 AM Bug filed on "Valid Login"

Step no.	Result	Title
1.	None	Navigate to the login page.
2.	None	enter a valid username/email.
3.	None	enter valid password.
4.	None	Click the login button.
5.	None	

Planning:

- Resolved Reason
- Story Points

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

Effort (Hours):

- Original Estimate
- Remaining

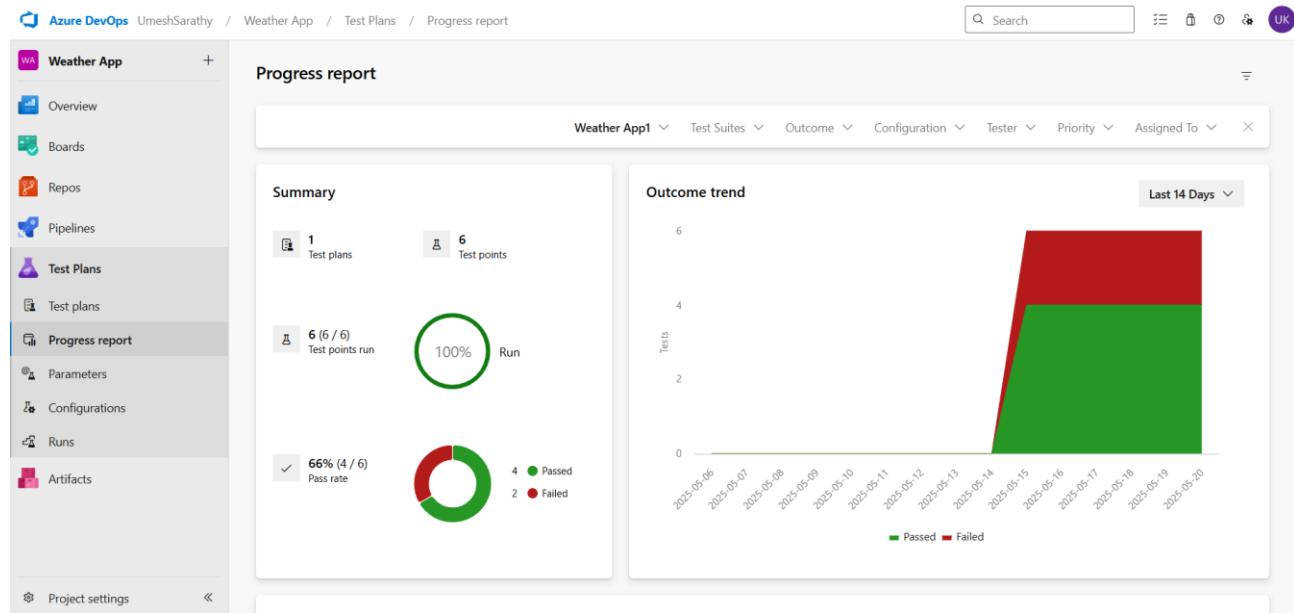
Related Work:

+ Add link

Add an existing work item as a parent

- Assigning bug to the developer and changing state

10. Progress report



11. Changing the test template

The screenshot shows the Azure DevOps 'Process' settings page. On the left, a sidebar lists 'General', 'Security', and 'Boards' sections. The 'Process' section is currently selected. The main area displays a table titled 'All processes' with four rows: 'Basic (default)', 'Agile', 'Scrum', and 'CMMI'. The 'Agile' row is highlighted, indicating it is selected. The table includes columns for 'Name', 'Description', and 'Team projects'.

Name	Description	Team projects
Basic (default)	This template is flexible for any process and great for teams getting started with Az...	1
Agile	This template is flexible and will work great for most teams using Agile planning me...	1
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 improv...	0

The screenshot shows the 'Work item types' section for the 'Agile' template. The sidebar on the left is identical to the previous screenshot. The main area shows a table of work item types: Bug, Epic, Feature, Issue, Task, Test Case, Test Plan, Test Suite, and User Story. Each item has a small icon and a brief description. A message at the top states: 'System processes cannot be customized. To add customization create an inherited process.'

Name	Description
Bug	Describes a divergence between required and actual behavior, and tracks the work done to correct the defect and verify the correct...
Epic	Epics help teams effectively manage and groom their product backlog
Feature	Tracks a feature that will be released with the product
Issue	Tracks an obstacle to progress.
Task	Tracks work that needs to be done.
Test Case	Server-side data for a set of steps to be tested.
Test Plan	Tracks test activities for a specific milestone or release.
Test Suite	Tracks test activities for a specific feature, requirement, or user story.
User Story	Tracks an activity the user will be able to perform with the product

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

AIM

To implement a Continuous Integration and Continuous Deployment (CI/CD) pipeline in Azure DevOps for automating the build, testing, and deployment process of the Student Management System, ensuring faster delivery and improved software quality.

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.

Azure DevOps Pipelines page for the Weather App project. The sidebar shows options like Overview, Boards, Repos, Pipelines (selected), Environments, Library, Test Plans, and Artifacts. The main area displays the Pipelines section with a 'Recently run pipelines' table. The table has columns for Pipeline, Last run, and Duration. One entry is shown: 'Weather App' with a green checkmark icon, 'Last run' at '#20250516.1 • Set up CI with Azure Pipelines', and 'Duration' of 'Friday 9s'. A 'New pipeline' button is located in the top right corner.

Azure DevOps Pipeline Runs page for the Weather App project. The sidebar is identical to the previous screenshot. The main area shows the 'Runs' tab selected under the Weather App pipeline. It displays a single run entry with a green checkmark icon, the description '#20250516.1 • Set up CI with Azure Pipelines', and a stage status column showing a green checkmark. The duration is listed as 'Friday 9s'. There are 'Edit' and 'Run pipeline' buttons in the top right.

RESULT

Thus the pipelines for the given project “**Weather app**” has been executed successfully.

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 Weather App

GitHub Project Structure

The screenshot shows a GitHub repository page for 'Weather-app'. The repository is public and has 5 commits. The files listed are 'images', 'README.md', 'app.js', 'index.html', and 'styles.css'. The 'README' tab is selected, displaying the content 'WEATHER APP'. The repository has 1 branch and 0 tags. The 'About' section describes it as 'It's Website that displays user's current location weather details and desired location's weather details'. It shows 0 forks and 1 watching. There are sections for 'Releases' (no releases published) and 'Packages' (no packages published). The 'Languages' section is also visible.

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.