



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

DEPARTMENT OF INFORMATION TECHNOLOGY
LAB MANUAL

CS23432 – Software Construction

(REGULATION 2023)

RAJALAKSHMI ENGINEERING COLLEGE
Thandalam, Chennai-602015

Name: Hemaprabha S

Register No: 231001063

Year / Branch / Section: 2nd/IT/ FA

Semester: IV

Academic Year: 2024 - 2025

INDEX

S.No.	Date	Title
1.	22/01/25	Azure Devops Environment Setup.
2.	07/02/25	Azure Devops Project Setup and User Story Management.
3.	14/02/25	Setting Up Epics, Features, And User Stories for Project Planning.
4.	20/03/25	Sprint Planning.
5.	28/03/25	Poker Estimation.
6.	04/04/25	Designing Class and Sequence Diagrams for Project Architecture.
7.	16/04/25	Designing Architectural and ER Diagrams for Project Structure.
8.	18/04/25	Testing – Test Plans and Test Cases.
9.	25/04/25	Load Testing and Pipelines.
10.	02/05/25	GitHub: Project Structure & Naming Conventions.

Exp.No: 1	AZURE DEVOPS ENVIRONMENT SETUP
Date:22/01/2025	

AIM: To set up and access the Azure DevOps environment by creating an organization through the Azure Portal.

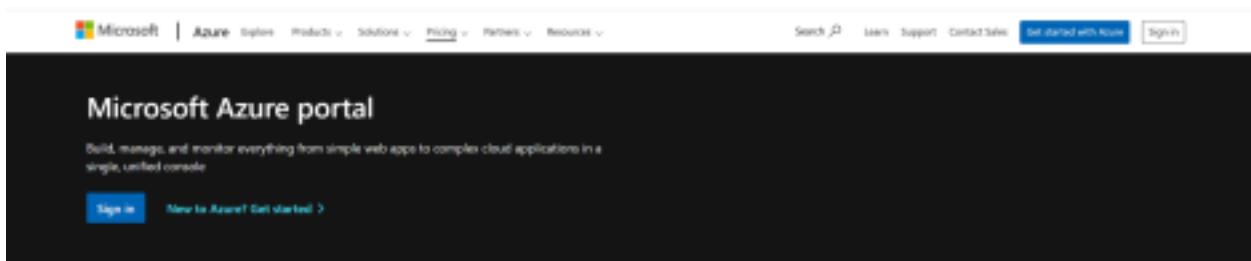
INSTALLATION

1. Open your web browser and go to the Azure website:

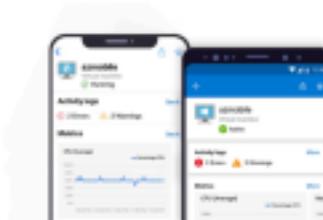
<https://azure.microsoft.com/en-us/getstarted/azure-portal>.

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>



Azure mobile app
Stay connected to your Azure resources—anytime, anywhere. Now available for iOS and Android.
[Learn more >](#)



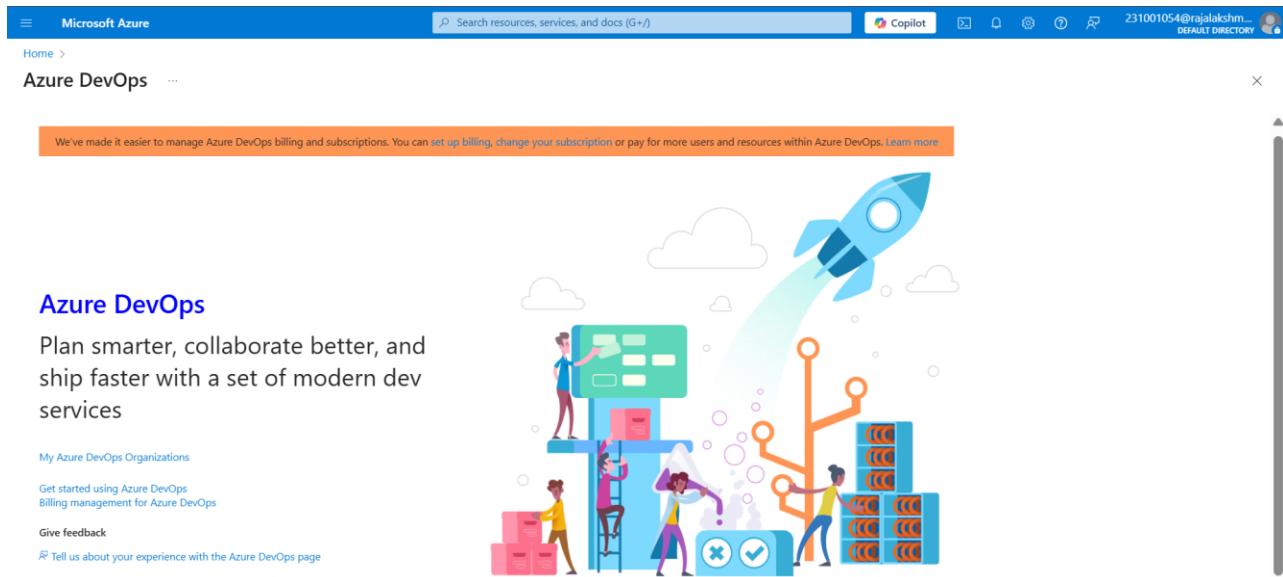
2.Azure home page

The screenshot shows the Microsoft Azure home page. At the top, there's a search bar and a Copilot button. Below the header, the "Azure services" section features a grid of icons for various services: Create a resource, Subscriptions, Azure DevOps organizations, Oracle Database@Azure, Quickstart Center, Azure AI foundry, Kubernetes services, Virtual machines, App Services, and More services. The "Resources" section shows a message: "No resources have been viewed recently" with a "View all resources" button. The "Navigate" section includes links for Subscriptions, Resource groups, All resources, and Dashboard. The "Tools" section lists Microsoft Learn, Azure Monitor, Microsoft Defender for Cloud, and Cost Management. The "Useful links" section includes Technical Documentation, Azure Services, Recent Azure Updates, and links to the App Store and Google Play. The "Azure mobile app" section shows download links for both platforms.

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". A dropdown menu appears, listing "All", "Services (7)", and "Resources". Under "Services", "Azure DevOps organizations" is highlighted. Other listed services include Azure Native New Relic Service, Managed DevOps Pools, and Azure Native Dynatrace Service. The rest of the page content is identical to the first screenshot, including the "Azure services", "Resources", "Navigate", "Tools", "Useful links", and "Azure mobile app" sections.

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 SET UP AND USER STORY MANAGEMENT
Date: 07/02/2025	

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

STEPS:

- 1.Create An Azure Account



Azure DevOps
231001054@rajalakshmi.edu.in [Switch directory](#)

Almost done...

Name your Azure DevOps organization *

We'll host your projects in

Enter the characters you see

New Audio



Continue

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.

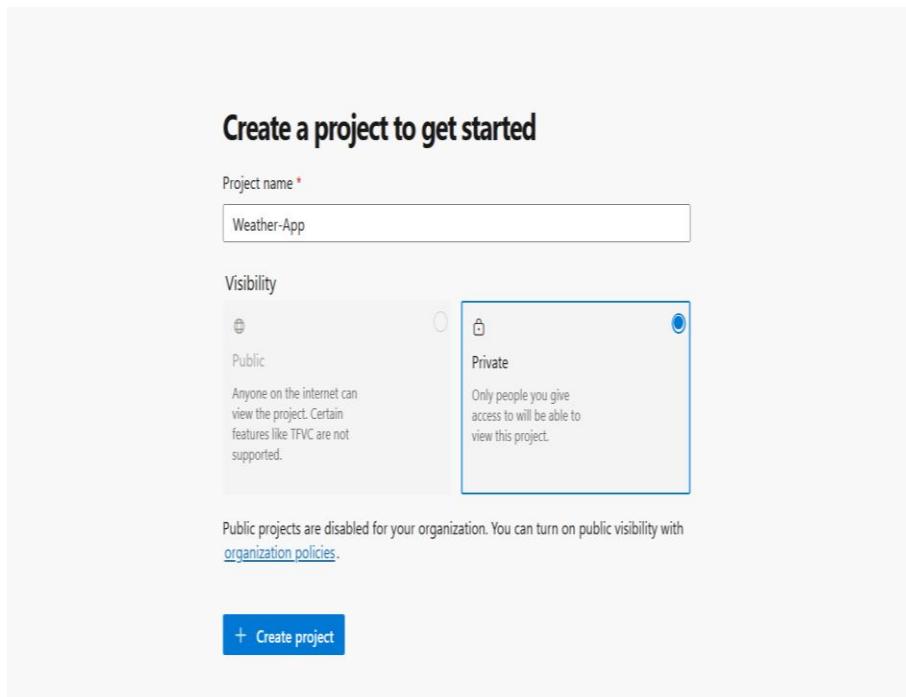
c. Enter the project name, description, and visibility options:

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 Organizations home page. At the top, there's a blue header bar with the Microsoft logo and a sign-out link. Below the header, there's a large green circular icon with the number '2' in white. To its right, the text '231001063' and 'Edit profile' are visible. Further down, there are sections for 'India' and 'Visual Studio Dev Essentials'. On the right side, under 'Azure DevOps Organizations', there are two collapsed organization cards:

- Software Construction** (Owner): Projects: Software Construction; Actions: Open in Visual Studio.
- hema** (Owner): Projects: hema, weather app; Actions: Open in Visual Studio.

A 'Create new organization' button is located at the top right of the organization list.

4. Project dashboard

The screenshot shows the Azure DevOps project dashboard for the 'weather app' project. At the top, there's a navigation bar with the Azure DevOps logo, the project name 'weather app', and links for Overview, Summary, Dashboards, Wiki, Boards, Repos, Pipelines, Test Plans, and Artifacts. The 'Summary' link is currently selected.

The main content area has a title 'weather app' with a red 'WA' icon. Below it is a section titled 'About this project' with a 'Like' button (0 likes) and a 'Comment' button. The 'About this project' section contains the following information:

- Project Description**: This project is a comprehensive Weather Forecasting System developed under Microsoft Azure's cloud ecosystem. It is designed to provide users with real-time weather information, a 5-day forecast, personalized settings, and safety alerts to improve daily decision-making and outdoor planning. The system is structured around four core functional themes:
- Weather Updates & Personalization**: This module offers real-time weather data and a 5-day forecast with temperature, humidity, and wind speed. Users can manually refresh or rely on automatic updates, with icons for sunny, rainy, or cloudy conditions. It supports toggling between Celsius and Fahrenheit for temperature display. Users can also save favorite cities, with preferences stored for future visits.
- Safety & Alerts**: This module provides emergency weather alerts for storms, heatwaves, and heavy rain. It also shows sunrise and sunset times and sends timely push notifications for critical updates.
- Admin Dashboard & User Management**: Admins can manage users, suspend/reactivate accounts, and monitor activity. The dashboard includes real-time server monitoring, error logs, and uptime tracking.
- Performance and Scalability**: Each module is designed for performance, scalability, and user engagement. The system uses Azure's cloud platform to ensure high availability and service continuity.

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 this user story.

The screenshot shows the Azure DevOps interface for the 'weather app' project. The left sidebar is visible with options like Overview, Boards, Work items, Boards, Backlogs, Sprints, Queries, Delivery Plans, Analytics views, Repos, Pipelines, and Project settings. The 'Backlogs' option is selected. The main area is titled 'weather app Team' and shows a 'Backlog' view. A table lists four backlog items, all categorized as 'Epic':

Order	Work Item Type	Title	State	Effort	Business	Value Area	Tags
1	Epic	> 🌡 Epic : Accurate and Real-Time Weather Updates	New		Business		
2	Epic	> 🌡 User Customization Options	New		Business		
3	Epic	> 🌡 Epic : Emergency Weather Alerts & Notifications	New		Business		
4	Epic	> 🌡 Epic : User and System Control	New		Business		

A blue 'New Work Item' button is located in the top right corner of the backlog view.

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

Exp.No: 3	SETTING UP EPICS FEATURES AND USER STORIES FOR PROJECT PLANNING
Date: 14/02/2025	

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

1.Fill in Epics

EPIC 1

1 Epic : Accurate and Real-Time Weather Updates

No one selected 0 Comments Add Tag

Save and Close Follow Details

Updated by Harini K Mar 31

Description	Planning	Deployment
Click to add Description.	Priority 2 Risk	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
Discussion	Business Value	Development
Add a comment. Use # to link a work item, @ to mention a person, or ! to link a pull request.	Time Criticality	Add link
switch to Markdown editor	Start Date Select a date...	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.
	Target Date Select a date...	Related Work
	Classification	Add link
	Value area Business	Add an existing work item as a parent
		Child
		2 Real-Time Weather Updates and 5-Day Forecast

2.Fill in Features

The screenshot shows the 'FEATURE 2' details page in Azure DevOps. The title is '2 Real-Time Weather Updates and 5-Day Forecast'. Key fields include:

- State:** New
- Area:** weather app
- Reason:** New
- Iteration:** weather app\sprint 1

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](#).

Discussion: Add a comment. Use # to link a work item, @ to mention a person, or ! to link a pull request.

Development: Add link

Related Work: 1 Epic : Accurate and Real-Time Weather Updates

Classification:

- Value area: Business
- Business

3.Fill in User Story Details

The screenshot shows the 'USER STORY 4' details page in Azure DevOps. The title is 'As a user, I want to see real-time weather updates so that I always have the latest weather information.'. Key fields include:

- State:** New
- Area:** weather app
- Reason:** New
- Iteration:** weather app\sprint 1

Description: Click to add Description.

Acceptance Criteria:

1. The app should fetch and display current temperature, humidity, and wind speed.
2. Weather data should update automatically at regular intervals.
3. A refresh button should be available for manual updates.

Planning:

- Story Points: 1
- Priority: 1
- 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](#).

Discussion: Add a comment. Use # to link a work item, @ to mention a person, or ! to link a pull request.

Development: Add link

Related Work: 2 Real-Time Weather Updates and 5-Day Forecast

Classification:

- Value area: Business
- Business

4.Fill in Task Details

The screenshot shows the 'Task' details page in Microsoft Azure DevOps. The task is titled '7 The app should fetch and display current temperature, humidity, and wind speed.' It has a status of 'New' and is assigned to 'weather app' in the 'weather app\sprint 1' iteration. The 'Description' section is empty, with a placeholder 'Click to add Description.' The 'Planning' section shows a priority of '2'. The 'Effort (Hours)' section lists 'Original Estimate', 'Remaining', and 'Completed' fields. The 'Implementation' section indicates 'Integrated in Build'. The 'Deployment' section contains a note about tracking releases. The 'Development' section links to Azure Repos for commits and pull requests. The 'Related Work' section shows a linked item '4 As a user, I want to see real-time weather upda...'.

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

Exp.No: 4	SPRINT PLANNING
Date: 20/03/2025	

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

SPRINT PLANNING

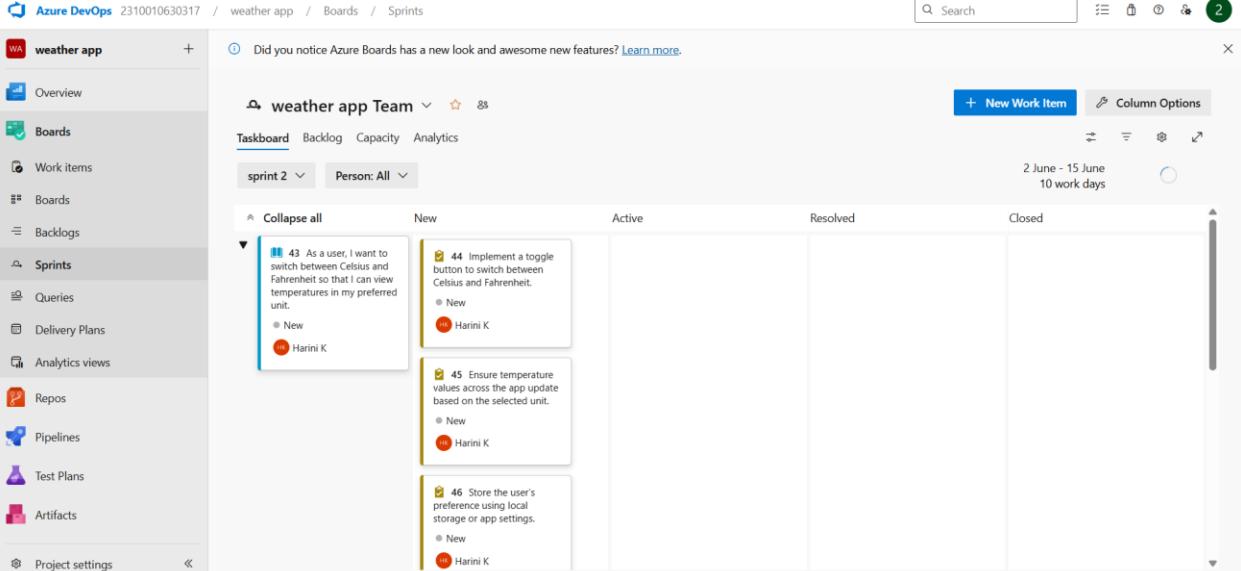
Sprint 1

The screenshot shows the Azure DevOps Boards Taskboard for the 'weather app' project. The board is set up for 'Sprint 1' and has four columns: 'New', 'Active', 'Resolved', and 'Closed'. Three user stories are visible in the 'New' column:

- User Story 4: As a user, I want to see real-time weather updates so that I always have the latest weather information. Status: New, Assigned to: 231001063.
- User Story 7: The app should fetch and display current temperature, humidity, and wind speed. Status: New, Assigned to: 231001063.
- User Story 8: Weather data should update automatically at regular intervals. Status: New, Assigned to: 231001063.

The sidebar on the left shows navigation links for Overview, Boards, Work items, Boards, Backlogs, Sprints (selected), Queries, Delivery Plans, Analytics views, Repos, Pipelines, Test Plans, Artifacts, and Project settings.

Sprint 2

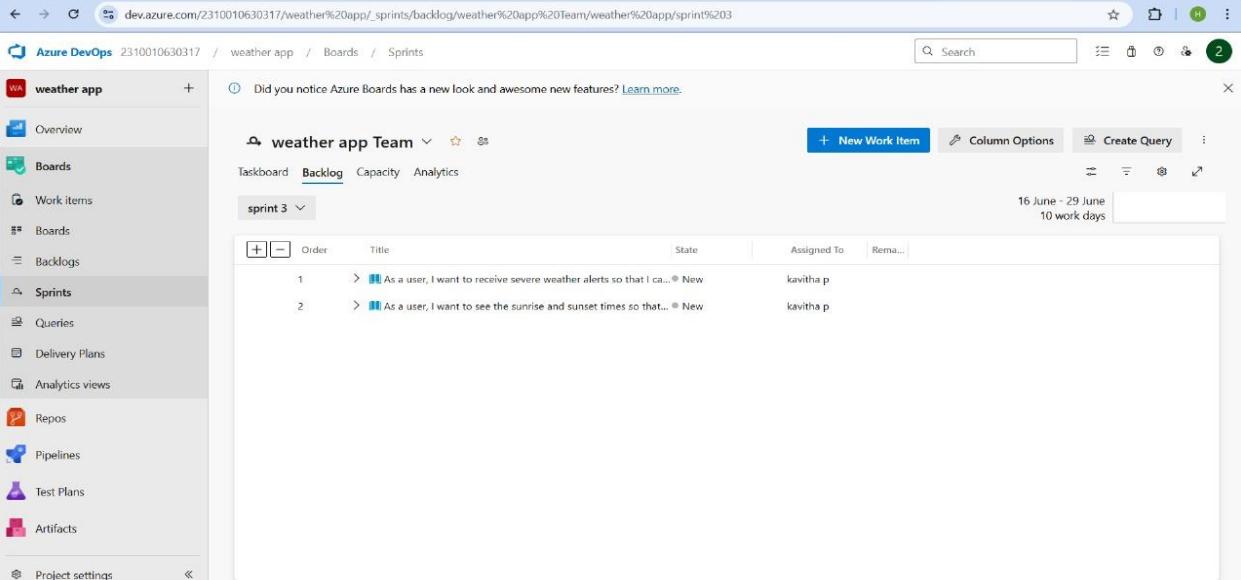


The screenshot shows the Azure Boards Taskboard for the 'weather app' project. The left sidebar navigation includes 'Overview', 'Boards', 'Work items', 'Backlogs', 'Sprints' (selected), 'Queries', 'Delivery Plans', 'Analytics views', 'Repos', 'Pipelines', 'Test Plans', 'Artifacts', and 'Project settings'. The main area displays the 'weather app Team' backlog under 'Taskboard'. The current sprint is 'sprint 2' (selected) and the filter is 'Person: All'. The backlog is organized into columns: New, Active, Resolved, and Closed. The 'New' column contains three work items:

- 43 As a user, I want to switch between Celsius and Fahrenheit so that I can view temperatures in my preferred unit. (New, assigned to Harini K)
- 44 Implement a toggle button to switch between Celsius and Fahrenheit. (New, assigned to Harini K)
- 45 Ensure temperature values across the app update based on the selected unit. (New, assigned to Harini K)
- 46 Store the user's preference using local storage or app settings. (New, assigned to Harini K)

At the top right, there are buttons for '+ New Work Item', 'Column Options', and a search bar.

Sprint 3



The screenshot shows the Azure Boards Backlog for the 'weather app' project. The left sidebar navigation is identical to the previous screenshot. The main area displays the 'weather app Team' backlog under 'Backlog' (selected). The current sprint is 'sprint 3' and the filter is 'Person: All'. The backlog is organized into columns: Order, Title, State, Assigned To, and Remarks. Two work items are listed:

Order	Title	State	Assigned To	Remarks
1	> As a user, I want to receive severe weather alerts so that I can... (New)	New	kavitha p	
2	> As a user, I want to see the sunrise and sunset times so that... (New)	New	kavitha p	

At the top right, there are buttons for '+ New Work Item', 'Column Options', 'Create Query', and a search bar.

Sprint 4

The screenshot shows the Azure DevOps Boards interface for the 'weather app' project. The left sidebar navigation bar is visible, showing options like Overview, Boards, Work items, Backlogs, Sprints, Queries, Delivery Plans, Analytics views, Repos, Pipelines, Test Plans, Artifacts, and Project settings. The main area displays the 'weather app Team' backlog under the 'Backlog' tab. A message at the top indicates a new look and features. The backlog table has columns for Order, Title, State, Assigned To, and Remaining. Two items are listed:

Order	Title	State	Assigned To	Remaining
1	> As an admin, I want to manage registered users so that I can... (New)	New	Harisha D	
2	> As an admin, I want to track system performance and server... (New)	New	Harisha D	

At the bottom right of the backlog view, it says '30 June - 13 July' and '10 work days'.

RESULT: The Sprints are created for the Weather App Project.

Exp.No: 5	POKER ESTIMATION
Date: 28/03/2025	

AIM: Create Poker Estimation for the user stories - Weather App Project. **POKER ESTIMATION**

User Story #4
As a user, I want to see real-time weather updates so that I always have the latest weather information.

Story Points: 23 | Priority: 1 | Risk: 1

Description: Click to add Description.

Acceptance Criteria:

- The app should fetch and display current temperature, humidity, and wind speed.
- Weather data should update automatically at regular intervals.
- A refresh button should be available for manual updates.

Classification: Value area: Business

Planning: Story Points: 23, Priority: 1, Risk: 1

Development: 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](#).

Related Work: Add link ▾ Parent: TC102_2: Real-Time Weather Updates and 5-Day Forecast Updated Mar 31 | New Child:

- TC102_3: Fetch and display current temperature, humidity, and wind speed. Updated Sunday | New
- TC102_4: Weather data should update automatically at regular intervals. Updated Sunday | New

Tested By: TC102_2: Validate displayed weather data Updated Mar 31 | Design
TC102_3: Fetch and display current temperature, humidity, and wind speed. Updated Mar 31 | Design
TC102_4: Weather data should update automatically at regular intervals. Updated Sunday | New

Show more (6 of 7). Not shown: Tested By (1)

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

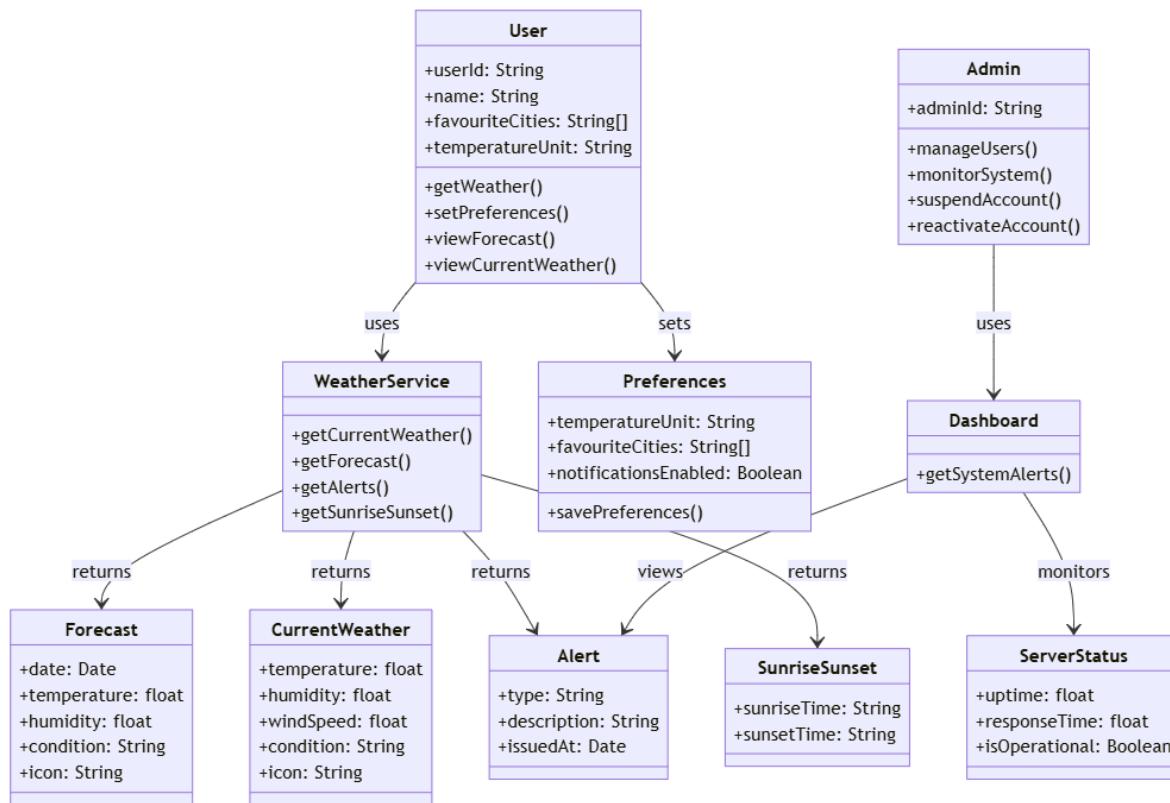
Exp.No: 6

DESIGNING CLASS AND SEQUENCE DIAGRAMS FOR PROJECT ARCHITECTURE

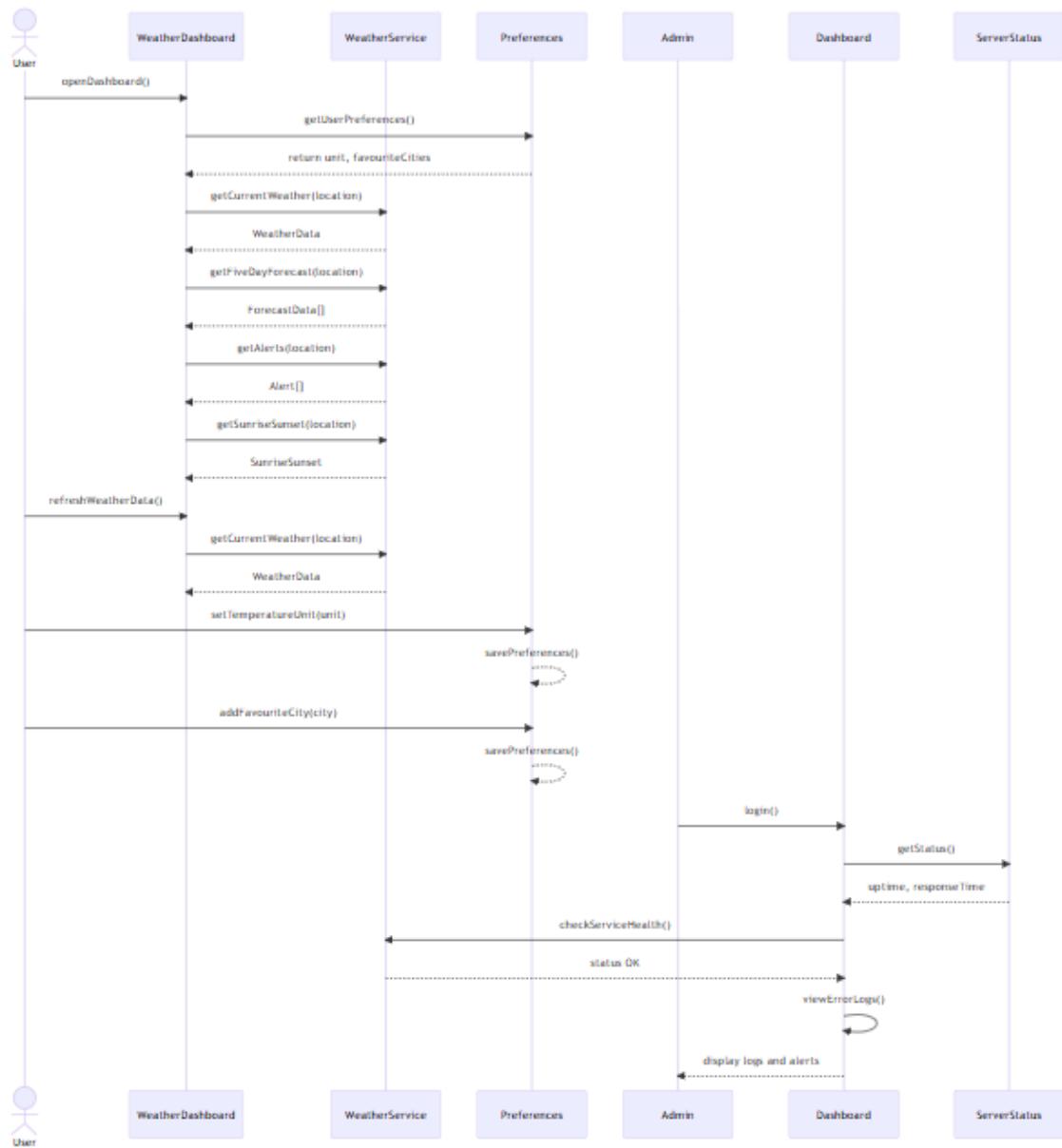
Date: 04/04/2025

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

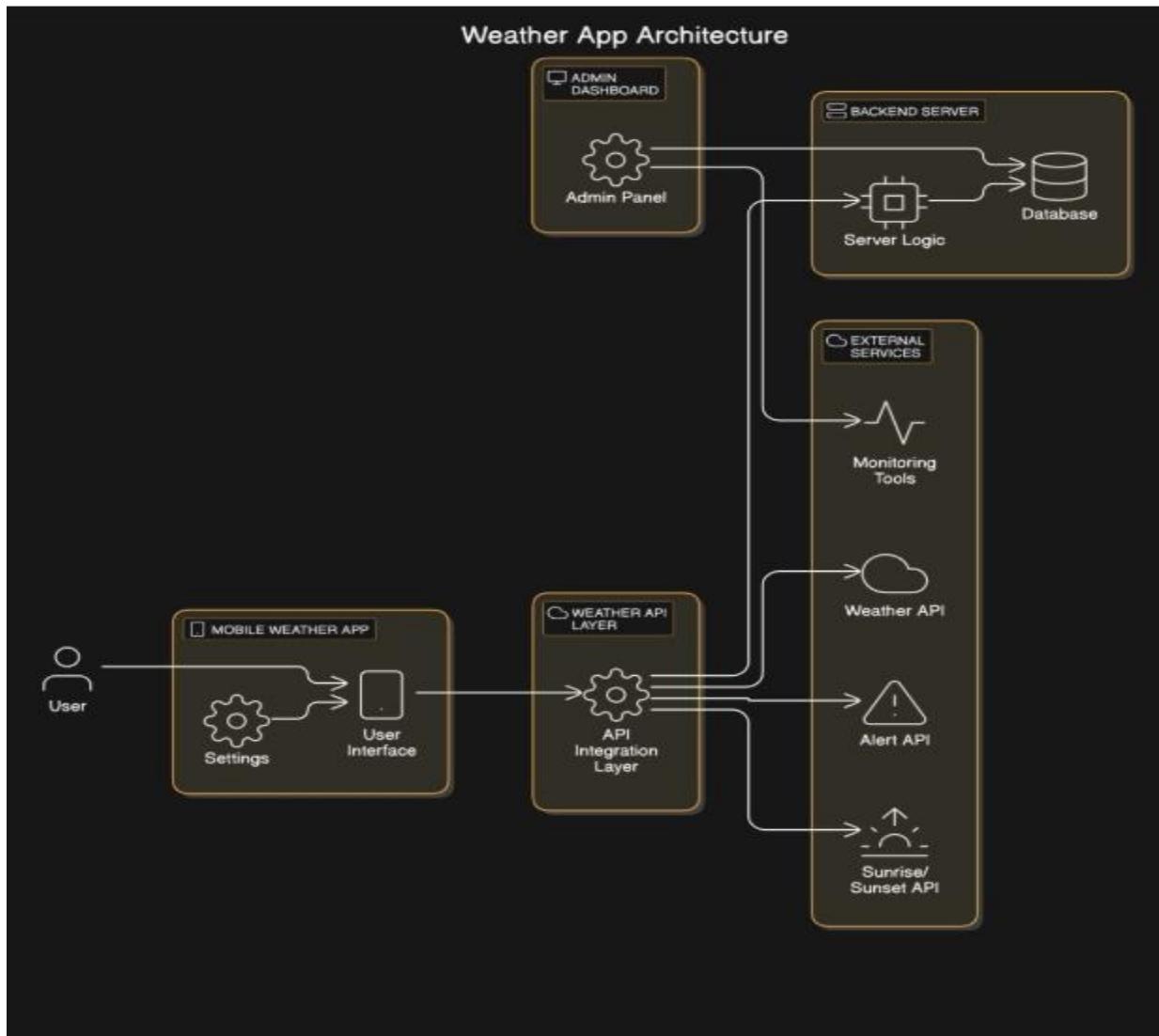
Exp.No: 7

DESIGNING ARCHITECTURAL AND ER DIAGRAMS FOR PROJECT STRUCTURE

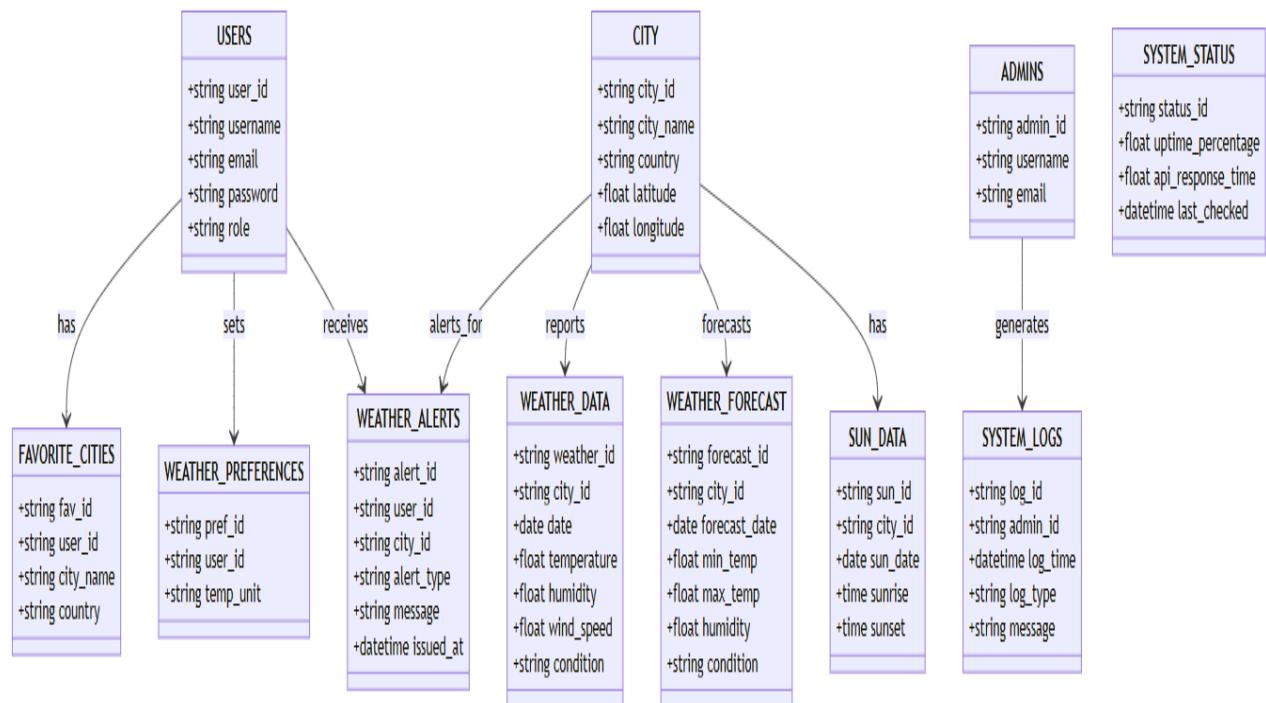
Date: 16/04/2025

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 Online Banking System.

Exp.No: 8	TESTING-TEST PLANS AND TEST CASES
Date: 18/04/2025	

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

- Real-Time Weather Updates (Current temperature, humidity, wind speed)
- 5-Day Weather Forecast (Daily conditions, temperature trends, icons)
- Emergency Weather Alerts (Storms, heatwaves, heavy rain notifications)
- Sunrise & Sunset Information (Based on selected city and date)
- Admin Dashboard (User management, system monitoring, API uptime)

2. Define User Interactions

Each test case reflects realistic user actions:

- Viewing real-time weather for a selected city
- Checking the 5-day weather forecast
- Switching between Celsius and Fahrenheit temperature units
- Receiving emergency weather alerts and notifications
- Viewing sunrise and sunset times for better daily planning
- Admin managing users and monitoring system status

3. Design Happy Path Test Cases

- Focused on successful flows where inputs are valid and systems behave as expected.
- Examples:
 - User views current weather for a selected city
 - User successfully switches temperature units
 - 5-day forecast displays accurate and updated data

4. Design Error Path Test Cases

- Focused on handling invalid, missing, or unexpected inputs.
- Examples:
 - Weather data fails to load due to no internet connection
 - API error returns blank forecast data
 - User tries to add a duplicate city to favourites

5. Break Down Steps and Expected Results

Each test case includes:

- Clear step-by-step user actions (e.g., selecting a city, clicking refresh)
- Specific expected outcomes for each action (e.g., weather data updates, alert is triggered)
This ensures reproducibility and easy automation later.

6. Use Clear Naming and IDs

- Example:
 - TC01 – View Real-Time Weather Successfully
 - TC04 – Toggle Temperature Unit to Fahrenheit
 - TC08 – Alert Not Displayed (No Severe Weather)
- Helps trace test cases to user stories and features in Azure DevOps

7. Separate Test Suites

Organize test cases based on functional modules:

- TS01 – Real-Time Weather & Forecast
- TS02 – Temperature Unit Toggle & Preferences
- TS03 – Favourite Cities Management
- TS04 – Emergency Alerts & Notifications
- TS05 – Sunrise/Sunset Info & UI Display
- TS06 – Admin Dashboard & System Monitoring

1. New test plan

 New Test Plan

Name *

Area Path *

Iteration *

 5/19/2025 - 6/1/2025 

Create Cancel

2.Test suite

The screenshot shows the Azure DevOps Test Plans interface for a project named 'weather app'. The left sidebar is expanded to show 'Test Plans' and 'Test suites'. A context menu is open over a test suite named 'Weather App - Functional Validation'. The menu options include 'New Suite', 'Assign configurations', 'Export', 'Assign testers to run all tests', and 'Import test suites'. The main area displays the 'Weather App - Functional Validation (ID: 87)' page, which includes tabs for 'Define', 'Execute', and 'Chart'. Below the tabs is a section titled 'Add a test case' with a sub-instruction 'Use this tab to collate, add and manage test cases' and a 'New Test Case' button. There is also a cartoon illustration of a painter.

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 – Weather Application

- **US199:** As a user, I want to see real-time weather updates so that I always have the latest weather information.
- **US200:** As a user, I want to view a 5-day weather forecast so that I can plan my activities in advance.
- **US196:** As an admin, I want to manage registered users so that I can monitor app usage and ensure compliance with policies.
- **US198:** As a user, I want to receive severe weather alerts so that I can take precautions.
- **US201:** As a user, I want to switch between Celsius and Fahrenheit so that I can view temperatures in my preferred unit

TEST SUITE: TC301_1- RECEIVE ALERT FOR STORM (ID:57)

Test Case:TC301_1: Receive alert for storm during disasters

User Story: As a user, I want to receive severe weather alerts so that I can take necessary precautions.

Test Type: Happy Path

Steps and Expected Results:

- **Action:** Enable storm alerts in settings
Expected Result: Setting is saved successfully
- **Action:** Trigger storm alert from system
Expected Result: Alert is activated
- **Action:** Send alert to user via preferred method
Expected Result: User receives storm alert notification

Notes:

Ensure user's contact info is valid. This test checks successful alert delivery.

TEST SUITE: TC301_1- GET PUSH NOTIFICATION (ID:57)

Test Case: TC301_2: Get push notification by the user

User Story: As a user, I want to receive severe weather alerts so that I can take necessary precautions.

Test Type: Error Path

Steps and Expected Results:

- **Action:** Enable push notifications for weather alerts
Expected Result: Setting is saved successfully
- **Action:** Disable app notification permissions from device settings
Expected Result: App no longer has permission to send push notifications
- **Action:** Trigger a severe weather alert
Expected Result: User does not receive the push notification

Notes:

Validates graceful handling of blocked notifications with a fallback or log entry.

TEST SUITE: TC202_1: OPEN SYSTEM DASHBOARD (ID:56)

Test Case: TC202_1: Open system dashboard by admin.

User Story: As an admin, I want to track system performance and server status so that I can ensure smooth app functionality.

Test Type: Happy Path

Steps and Expected Results:

- **Action:** Navigate to the login page
Expected Result: Login page is displayed
- **Action:** Enter valid admin username and password
Expected Result: Credentials are accepted with no errors
- **Action:** Click the "Login" button
Expected Result: Admin is redirected to the system dashboard

Notes:

Ensure the test user has admin privileges. This test confirms successful access to the dashboard under normal conditions.

Test Case: TC301_3: No alert when clear weather

User Story: As a user, I want to receive severe weather alerts so that I can take necessary precautions.

Test Type: Error Path

Steps and Expected Results:

- **Action:** Enable severe weather alerts in user settings
Expected Result: Setting is saved successfully
- **Action:** Simulate or confirm clear weather conditions
Expected Result: No alert is triggered by the system
- **Action:** Monitor user's device for notifications
Expected Result: No alert or notification is received

TEST SUITE: TC102_3: Verify last updated time shown (ID:54)

Test Case: TC102_3: Verifies the last updated time shown

User Story: As a user, I want to see real-time weather updates so that I always have the latest weather information.

Test Type: Happy Path

Steps and Expected Results:

- **Action:** Open the weather application/dashboard
Expected Result: Weather data is displayed along with the "Last Updated" timestamp
- **Action:** Wait for a scheduled weather data refresh
Expected Result: "Last Updated" time updates to reflect the most recent fetch
- **Action:** Compare displayed time with system/server time
Expected Result: Timestamp accurately reflects the latest data update

TEST SUITE: TC102_2- Validate displayed weather data (ID:54)

Test Case: TC102_2- Validate displayed weather data

User Story: As a user, I want to see real-time weather updates so that I always have the latest weather information.

Test Type: Error Path

Steps and Expected Results:

- **Action:** Disconnect the device from the internet
Expected Result: App fails to fetch latest weather data
- **Action:** Open the weather application/dashboard
Expected Result: An error message like “Unable to retrieve data” is shown
- **Action:** Check displayed weather information
Expected Result: Outdated data is shown with a warning or timestamp indicating it's not current

TEST SUITE: TC201– DELETE A USER (ID: 55)

Test Case: TC201 – Deleting a user

User Story: As an admin, I want to manage registered users so that I can monitor app usage and ensure compliance with policies.

Test Type: Happy Path

Steps and Expected Results:

- **Action:** Log in with valid admin credentials
Expected Result: Admin is redirected to the admin dashboard
- **Action:** Navigate to the "User Management" section
Expected Result: List of registered users is displayed
- **Action:** Select a user and click the "Delete" button
Expected Result: Confirmation prompt appears
- **Action:** Confirm the deletion
Expected Result: User is removed from the system and a success message is displayed

TEST SUITE: TC201_3 - INVALID USER DELETE ATTEMPT (ID:55)

Test Case:TC201_3 - invalid user delete attempt

User Story: As an admin, I want to manage registered users so that I can monitor app usage and ensure compliance with policies.

Test Type: Error Path

Steps and Expected Results:

- **Action:** Log in with valid admin credentials
Expected Result: Admin is redirected to the admin dashboard
- **Action:** Navigate to the "User Management" section
Expected Result: List of registered users is displayed
- **Action:** Attempt to delete a non-existent or already deleted user
Expected Result: Error message is shown (e.g., “User not found” or “Unable to delete user”)

TEST SUITE: TC102_1: CHECK UPDATE REFRESH INTERVAL (ID:54)

Test Case: Check update refresh interval

User Story: As a user, I want to see real-time weather updates so that I always have the latest weather information.

Test Type: Happy Path

Steps and Expected Results:

- **Action:** Open the weather application/dashboard
Expected Result: Weather data is displayed with a visible "Last Updated" timestamp
- **Action:** Wait for the predefined refresh interval (e.g., 10 minutes)
Expected Result: Weather data is refreshed automatically without manual action
- **Action:** Verify that the "Last Updated" timestamp has changed
Expected Result: Timestamp reflects the new update time, confirming automatic refresh

TEST SUITE: TC102_2: Validate displayed weather data (ID:54)

Test Case: Validate displayed weather data

User Story: As a user, I want to see real-time weather updates so that I always have the latest weather information.

Test Type: Error Path

Steps and Expected Results:

- **Action:** Disconnect the device from the internet
Expected Result: Network connection is lost
- **Action:** Open the weather application/dashboard
Expected Result: Weather data fails to update
- **Action:** Observe the displayed weather information
Expected Result: App shows an error message (e.g., "No internet connection") and displays previously cached data with an outdated timestamp

Test Cases

TEST CASE 62*

62 TC101_2: Verify forecast shows temperature and condition

Harini K 0 Comments Add Tag

State: Design Area: weather app
Reason: New Iteration: weather app\sprint 1

Steps

Steps	Action	Expected result
1.	Verify that the 5-day forecast displays the daily temperature correctly for each day.	The forecast should show accurate temperature values for all five days matching the API data.
2.	Verify that the weather condition (e.g., sunny, rainy, cloudy) is shown for each day in the 5-day forecast.	Each day in the forecast displays a correct weather condition icon corresponding to the API data.
3.	Verify that the forecast updates automatically when new data is available from the weather API.	The forecast data refreshes without user intervention, showing the latest temperature and condition.

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

Tests

As a user, I want to view a 5-day weather forecast so th... Updated Sunday New

NEW TEST CASE*

Admin User Management

Harini K 0 Comments Add Tag

State: Design Area: weather app
Reason: New Iteration: weather app\sprint 1

Steps

Steps	Action	Expected result	Attachments
1.	Verify that the admin can access the list of all registered users from the admin dashboard.	The admin should see a complete and up-to-date list of all registered users displayed clearly.	
2.	Verify that the user list shows relevant user details such as username, email, and account status.	The list displays accurate user information for each registered user without missing data.	

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

Status

Priority: 2

TEST CASE 88*

88 Admin User Management

Harini K 0 Comments Add Tag

Save and Close Follow

State: Design Area: weather app Reason: New Iteration: weather app\sprint 1 Updated by Harini K 2m ago

Steps Summary Associated Automation

Steps Deployment Custom

Test Type: Error Path

Steps Action Expected result

- Verify that the admin can access the list of all registered users from the admin dashboard. The admin should see a complete and up-to-date list of all registered users displayed clearly.
- Verify that the user list shows relevant user details such as username, email, and account status. The list displays accurate user information for each registered user without missing data.

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

Status

Priority: 2

TEST CASE 88*

88 Admin User Management

Harini K 0 Comments Add Tag

Save and Close Follow

State: Design Area: weather app Reason: New Iteration: weather app\sprint 1 Updated by Harini K Just now

Steps Summary Associated Automation

Steps Deployment Custom

Test Type: Happy Path

Steps Action Expected result Attachments

- Verify that the admin can access the list of all registered users from the admin dashboard. The admin should see a complete and up-to-date list of all registered users displayed clearly.
- Verify that the user list shows relevant user details such as username, email, and account status. The list displays accurate user information for each registered user without missing data.

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

Status

Priority: 2

Parameter values

TEST CASE T2*

72 TC202_3: High Usage Alert Handling

Harini K 0 Comments Add Tag

Save Follow Steps Summary Associated Automation 2 0

Updated by 231001063: 11h ago

State: Design Area: weather app Reason: New Iteration: weather app\sprint 1

Steps

Steps	Action	Expected result
1.	Simulate high API request volume exceeding the rate limit.	Show alert message "API rate limit exceeded. Please wait before retrying."
2.	Trigger multiple simultaneous data refresh requests from the app.	Display warning "Too many requests. Updab temporarily paused."

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.

Custom

Test Type: Error Scenario

Related Work

Add link

Add an existing work item as a parent

Tests

18 As an admin, I want to track system performance... Updated Sunday New

4. Installation of test

Test and feedback

chrome web store

Discover Extensions Themes

Search extensions and themes

Test & Feedback

Featured 4.2★ (175 ratings) Share

Add to Chrome

Extension Workflow & Planning 200,000 users

Capture & Annotate

CAPTURE

1. Screenshots
2. Notes
3. Screen recordings
4. Image action log
5. Page load data

Overview

Showing it as an extension

The screenshot shows the Azure DevOps Test Plans interface for a project named "weather app". The left sidebar is visible with options like Overview, Boards, Repos, Pipelines, Test Plans, Test plans, Progress report, Parameters, Configurations, Runs, Artifacts, and Project settings. The "Test Plans" section is currently selected. In the center, a test plan titled "Weather App Test Plan" is displayed, showing a test suite named "33 : As a user, I want to receive severe weather alerts so that I can...". This suite contains three test points: "TC301_1: Receive alert for storm" (Failed, 1, 73), "TC301_2: Get push notification" (Passed, 2, 74), and "TC301_3: No alert when clear weather" (Passed, 3, 75). A modal window titled "Extensions" is open on the right, showing a list of extensions with "Full access" and their descriptions. The "McAfee WebAdvisor" extension is listed with a status of "Failed". Other extensions shown are "Test & Feedback" and "Manage extensions".

Title	Status	Count	Score
TC301_1: Receive alert for storm	Failed	1	73
TC301_2: Get push notification	Passed	2	74
TC301_3: No alert when clear weather	Passed	3	75

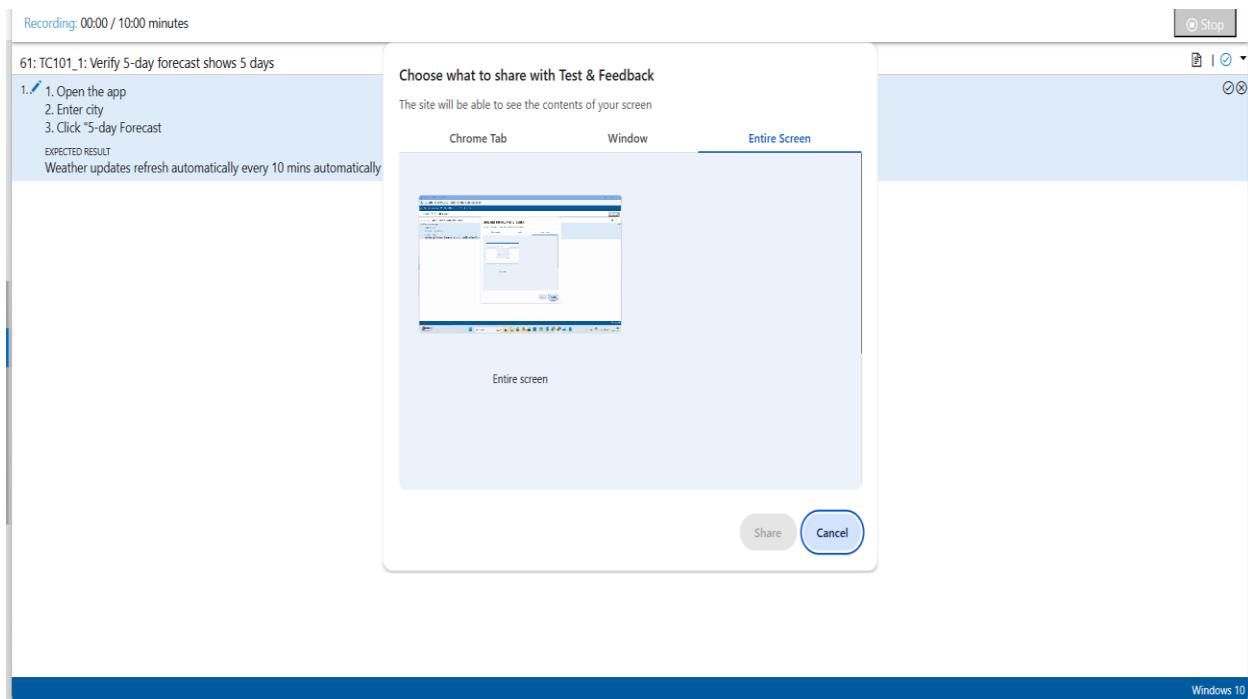
5. Running the test cases

The screenshot shows the Azure DevOps Test Plans interface for a project named "weather app". The left sidebar is visible with options like Overview, Boards, Repos, Pipelines, Test Plans, Test plans, Progress report, Parameters, Configurations, Runs, Artifacts, and Project settings. The "Test Plans" section is currently selected. In the main area, a "Weather App Test Plan" is selected, showing a summary: "May 19 - Jun 1", "100% run, 66% passed", and a "View report" link. Below this, there's a "Test Suites" section with a search bar and a "Test Points (3 items)" table.

Title	Outcome	Order	Test Case Id
TC101_1: Verify 5-day forecast shows 5 days	Passed	1	61
TC101_2: Verify forecast shows temperature and condition	Passed	2	62
TC101_3: Handle forecast API failure	Failed	3	63

The screenshot shows a browser window titled "Runner - Test Plans - Google Chrome" with the URL "dev.azure.com/2310010630317/weather/_testExecution/Index". The page displays a test case titled "61: TC101_1: Verify 5-day forecast shows 5 days". The test steps listed are: "1. Open the app", "2. Enter city", and "3. Click '5-day Forecast'". Below the steps, under "EXPECTED RESULT", it states: "Weather updates refresh automatically every 10 mins automatically every 10 mins". The browser interface includes standard navigation buttons (back, forward, search) and a toolbar with "Save and close" and "Create bug" buttons.

6.Recording the test case



7.Creating the bug

The screenshot shows the Azure DevOps work item creation page for a bug.

Bug 85 BUG_NO_01

Recently updated

85 BUG_NO_01 by Harini K

State: New **Reason:** New

Area: weather app **Iteration:** weather app\sprint 1

Comments: 0 **Add Tag:**

Details

Browser - Name	Google Chrome 136
Browser - Language	en-US
Browser - Height	832
Browser - Width	1552
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	x64_64
Operating system - Processor model	13th Gen Intel(R) Core(TM) i5-1335U
Operating system - Number of processors	12
Memory - Available	5513699328
Memory - Capacity	16849256448
Display - Pixels per inch (X axis)	120
Display - Pixels per inch (Y axis)	120
Display - Device pixel ratio	1.25

Related Work

- Added link: 18 As an admin, I want to track syst... (New)
- Tested By: 72 TC202_3: High usage alert trigger... (Design)

System Info

- Found in Build
- Integrated in Build

21162310010_

CS23432

The screenshot shows the Azure DevOps interface for a project named 'weather app'. The main view is the 'Work items' section, specifically the 'Recently updated' tab. A single work item is selected: 'BUG 85* BUG_NO_01' created by Harini K. The work item details show it is a 'New' bug in the 'weather app' iteration 'weather app/sprint 1'. The 'Repro Steps' section contains a step that failed: '1. Failed 1. Simulate CPU load > 90% 2. Observe alerts'. The 'Expected Result' is 'Admin sees alert: "High CPU usage"'. The 'Planning' and 'Deployment' sections are visible on the right, along with a note about deployment status reporting. The 'Development' section includes an 'Add link' button and a note about linking to Azure Repos. The bottom of the screen shows the Windows taskbar.

8. Test case results

The screenshot shows the Azure DevOps interface for a project named 'weather app'. The left sidebar is the 'Test Plans' section, with 'Test plans' selected. A specific test plan, 'Weather App Test Plan', is open. It contains several test suites, with one named '3 : As a user, I want to view a 5-day forecast' highlighted. This suite has three test points: 'TC101_1: Verify 5-day forecast shows 5 days' (Passed), 'TC101_2: Verify forecast shows temp' (Failed), and 'TC101_3: Handle forecast API failure' (Passed). To the right, a detailed view of the first test point is shown in a modal window. The modal title is 'TC101_1: Verify 5-day forecast shows 5 days'. It shows the 'Test Case Results' table:

Outcome	TimeStamp	Configuration	Run by	Tester	Test
Passed	20m ago	Windows 10	231001063	Harini K	Week 1
Failed	21m ago	Windows 10	231001063	Harini K	Week 1
Passed	21m ago	Windows 10	231001063	Harini K	Week 1

9. Test report summary

The screenshot shows the Azure DevOps interface for a project named 'weather app'. The left sidebar has 'Work items' selected. The main area displays a work item titled 'BUG_90' with the description '90 BUG_NO_1: Weather Data Does Not Refresh After Auto or Manual Refresh'. The work item details include:

- Repro Steps:**

Step No.	Result	Title
1	Passed	Launch the weather application. Verify the home screen is fully rendered, including all UI elements.
2	Passed	Wait for the configured auto-refresh interval. Alternatively, manually refresh the page using the refresh icon Confirm that the page reloads without errors.
3	Failed	Check the weather data after refresh. Verify if the data updates correctly with the latest information. If the data remains unchanged, confirm stale data is displayed despite refresh.
- Planning:**
 - Resolved Reason: None
 - Story Points: 0
 - Priority: 2
 - Severity: 3 - Medium
 - Activity: None
- Deployment:**
 - To track releases associated with this work item, go to [Releases](#) and turn on deployment status reporting for Boards in your pipeline's Options menu. [Learn more about deployment status reporting](#)
- Development:**
 - Add link: Link an Azure Repos [commit](#), [pull request](#) or [branch](#) to see the status of your development. You can also [create a branch](#) to get started.
- Effort (Hours):**
 - Original Estimate: 0
 - Remaining: 0
 - Completed: 0
- Related Work:**
 - Add link: Add an existing work item as a parent
- System Info:**
 - Found in Build
 - Integrated in Build

- Assigning bug to the developer and changing state

The screenshot shows the Azure DevOps interface for a project named 'weather app'. The left sidebar has 'Work items' selected. The main area displays a work item titled 'BUG_89' with the description '89 BUG_NO_2: Weather Data Not Updating in Real Time'. The work item details include:

- Repro Steps:**

Step No.	Result	Title
1	Passed	Launch the weather application. Verify the home screen is fully rendered, including all UI elements
2	Failed	Wait for the configured auto-refresh interval Or manually refresh by clicking the refresh icon or using the browser reload. Observe the application during and after refresh.
- Planning:**
 - Resolved Reason: None
 - Story Points: 0
 - Priority: 2
 - Severity: 3 - Medium
 - Activity: None
- Deployment:**
 - To track releases associated with this work item, go to [Releases](#) and turn on deployment status reporting for Boards in your pipeline's Options menu. [Learn more about deployment status reporting](#)
- Development:**
 - Add link: Link an Azure Repos [commit](#), [pull request](#) or [branch](#) to see the status of your development. You can also [create a branch](#) to get started.
- Effort (Hours):**
 - Original Estimate: 0
- Related Work:**
 - Add link: Add an existing work item as a parent

10. Progress report

Azure DevOps 2310010630317 / weather app / Test Plans / Progress report

Search 2

WA weather app +

Overview

Boards

Repos

Pipelines

Test Plans

Test plans

Progress report

Parameters

Configurations

Runs

Artifacts

Project settings

Progress report

Weather App Test Plan Test Suites Outcome Configuration Tester Priority Assigned To X

Summary

1 Test plans 15 Test points

15 (15 / 15) Test points run 100% Run

✓ 66% (10 / 15) Pass rate 10 Passed 5 Failed

Outcome trend

Last 14 Days

Tests

2025-05-07 2025-05-08 2025-05-09 2025-05-10 2025-05-11 2025-05-12 2025-05-13 2025-05-14 2025-05-15 2025-05-16 2025-05-17 2025-05-18 2025-05-19 2025-05-20 2025-05-21

Legend: Not run (light grey), Passed (green), Failed (red)

Progress report

Weather App Test Plan Test Suites Outcome Configuration Tester Priority Assigned To X

Summary

1 Test plans 15 Test points

15 (15 / 15) Test points run 100% Run

✓ 66% (10 / 15) Pass rate 10 Passed 5 Failed

Outcome trend

Last 14 Days

Tests

2025-05-07 2025-05-08 2025-05-09 2025-05-10 2025-05-11 2025-05-12 2025-05-13 2025-05-14 2025-05-15 2025-05-16 2025-05-17 2025-05-18 2025-05-19 2025-05-20 2025-05-21

Legend: Not run (light grey), Passed (green), Failed (red)

11.Changing the test template

The screenshot shows the 'All processes' section of the Azure DevOps settings. On the left, there's a sidebar with 'Organization Settings' and various sections like General, Security, Boards, and Process. Under Process, 'Pipelines' is expanded, showing Agent pools, Settings, Deployment pools, Parallel jobs, and OAuth configurations. The main area shows a table of process templates:

Name	Description	Team projects
Basic (default)	This template is flexible for any process and great for teams getting started with Azure DevOps.	1
Agile	This template is flexible and will work great for most teams using Agile planning methods, including those practicing Scrum.	0
2310010630317 Agile	(selected)	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 improvement and an auditable record of decisions.	0

12.View the new test case template

The screenshot shows the 'Add a field to Test Case' dialog. The left sidebar has tabs for 'Definition', 'Options', and 'Layout'. The 'Definition' tab is active, showing the instruction: 'Add a field to store custom, queryable data about your work items.' There are two options: 'Use an existing field' (radio button is not selected) and 'Create a field' (radio button is selected). For 'Create a field', the 'Name' field contains 'Test Type', 'Type' is set to 'Text (single line)', and 'Description' is 'Optionally provide a description for the field'. At the bottom, there are 'Learn more' and 'Add field' (blue button) and 'Cancel' buttons.

The screenshot shows the Azure DevOps interface for managing test cases. The left sidebar is titled "Organization Settings" and includes sections for General (Overview, Projects, Users, Billing, Global notifications, Usage, Extensions, Microsoft Entra), Security (Security overview, Policies, Permissions), Boards, and Process. The "Process" section is currently selected. The main content area is titled "All processes > 2310010630317 Agile > Test Case". It features a "Layout" tab with options for "New field", "New group", "New page", "Get extensions", and a refresh button. Below the layout tabs are three buttons: "Steps", "Summary", and "Associated Aut...". The "Steps" section contains a "Text (multiple lines)" input field. To the right of the steps are several sections: "Recent test results" (Recent test case results), "Deployment" (Deployments), "Development" (Links), "Related Work" (Links), and "Status" (Priority: Integer, Automation status). A search bar at the top right contains the text "Search" and a notification badge with the number "2".

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 PERFORMANCE TESTING
Date: 25/04/2025	

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 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.

Microsoft Azure

Home > Azure Load Testing > Create a load testing resource ...

Basics Encryption Tags Review + create

Azure Load Testing is a fully managed load-testing service that makes it easy to generate high-scale load and identify performance bottlenecks. [Learn more](#)

Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription * Azure for Students

Resource group * Weather_app

Create new

Instance details

Name * WEATHERAPP

Region * East Asia

Previous Next Review + create

Microsoft Azure

Home > Create a resource > Marketplace > Azure Load Testing > Create a load testing resource ...

Validation passed.

Basics Encryption Tags Review + create

Basics

Subscription	Azure for Students
Resource group	Weather_app
Name	WEATHERAPP
Region	East Asia

Encryption

Encryption type MMK

Previous Next Create

Microsoft Azure

Home > Microsoft.CloudNativeTesting1747797956985 | Overview ...

Deployment

Search Delete Cancel Redeploy Download Refresh

Overview

Your deployment is complete

Deployment name : Microsoft.CloudNativeTesting1747797956985
 Subscription : Azure for Students
 Resource group : Weather_app

Start time : 5/21/2025, 8:56:25 AM
 Correlation ID : 665ecbb1-ec3d-4a6a-9c93-d0429dc2bcd1

Deployment details

Next steps

Go to resource

Give feedback
 Tell us about your experience with deployment

Cost management
 Get notified to stay within your budget and prevent unexpected charges on your bill.
[Set up cost alerts >](#)

Microsoft Defender for Cloud
 Secure your apps and infrastructure
[Go to Microsoft Defender for Cloud >](#)

Free Microsoft tutorials
[Start learning today >](#)

Work with an expert
 Azure experts are service provider partners who can help manage your assets on Azure and be your first line of support.
[Find an Azure expert >](#)

Add or remove favorites by pressing **Ctrl+Shift+F**

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.

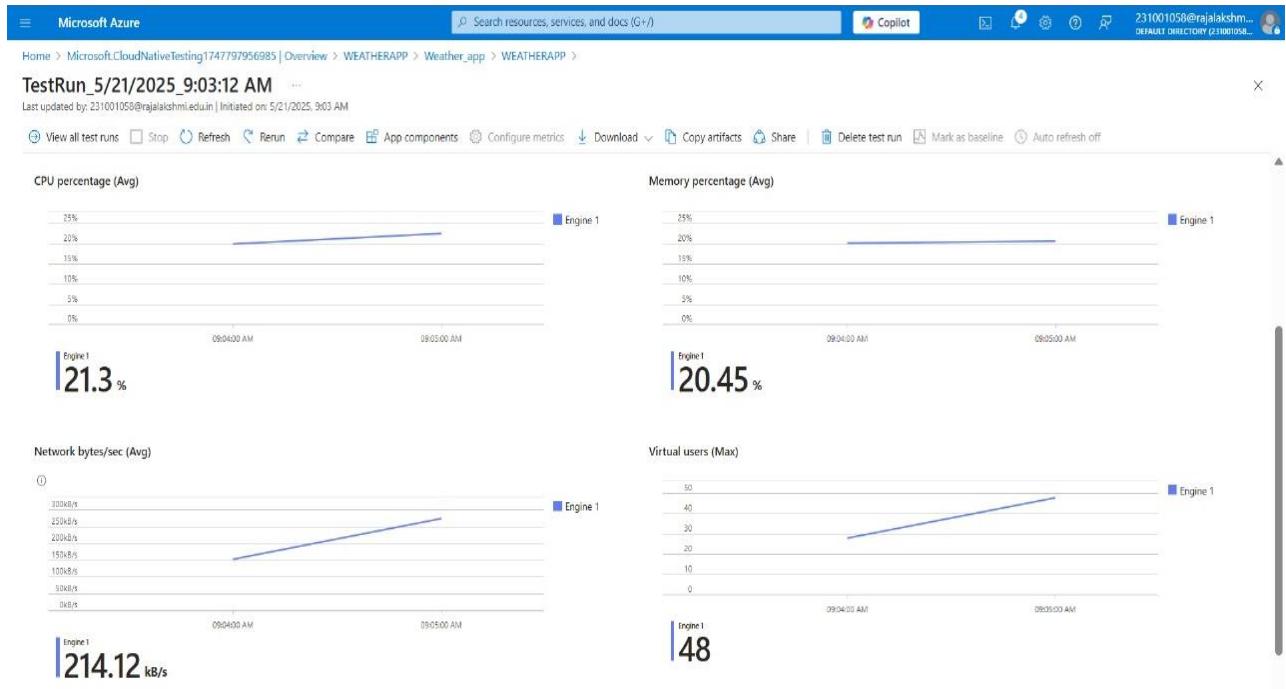
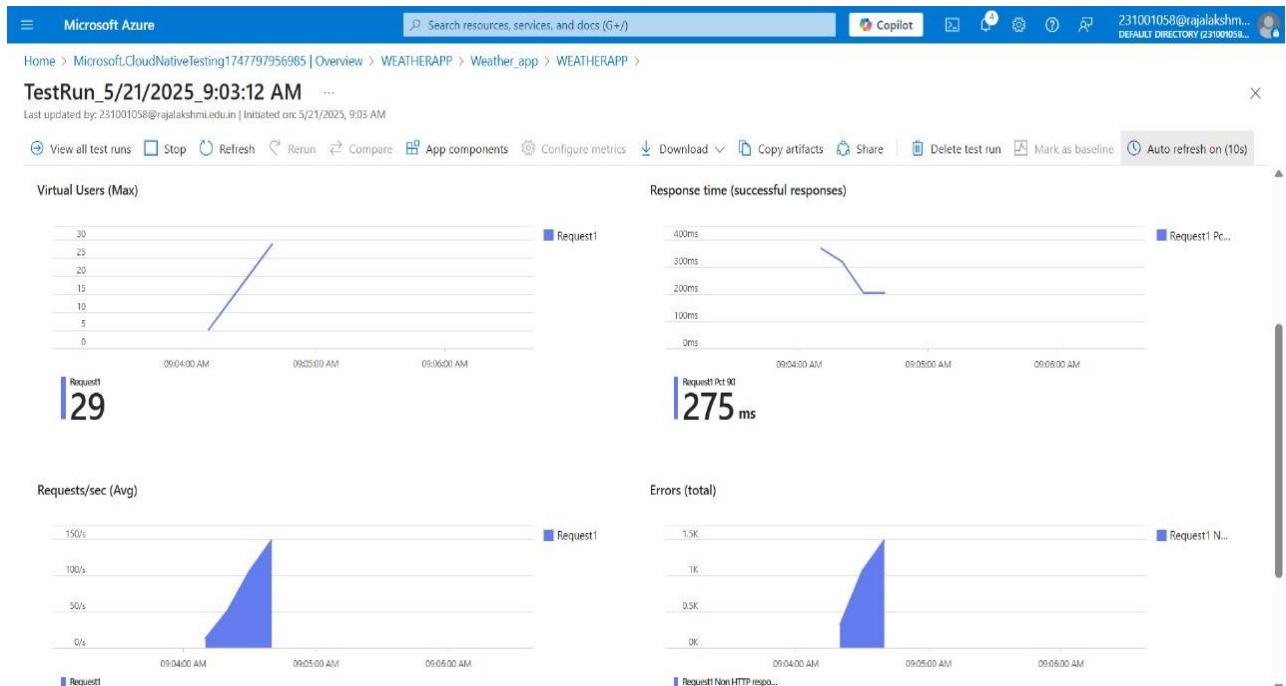
3. Load Settings o *Test URL*: Enter the target endpoint (e.g.,

<https://yourapi.com/products>). 4. Click Review + Create → Create to start the test.

The screenshot shows the 'Create a load testing resource' wizard in the Azure portal. The 'Basics' tab is selected. The 'Subscription' dropdown is set to 'Azure for Students'. The 'Resource group' dropdown shows '(New) BankProject' with 'Create new' as an option. Under 'Instance details', the 'Name' field is 'MYBANKAPP' and the 'Region' is 'East Asia'. At the bottom, there are 'Previous', 'Next', and 'Review + create' buttons, with 'Review + create' being the active button.

The screenshot shows the 'Create a URL-based test' configuration page. The 'Review + create' tab is selected. The 'Test tool' is 'JMeter'. The 'Test name' is 'Test_5/21/2025_8:59:27 AM'. The 'Test description' is 'Disabled'. Under 'Test plan', 'Test method' is 'URL' and 'Requests' is 'Request#1'. Under 'Load', 'Engine instances' is '1', 'Load pattern' is 'Linear', 'Concurrent users per engine' is '50', 'Test duration (minutes)' is '20', and 'Ramp-up time (minutes)' is '1'. At the bottom, there are 'Previous', 'Next', and 'Create' buttons, with 'Create' being the active button.

Load Testing



Weather App - All In One

Home Dashboard Settings

Home

Current Weather

📍 City: New York
🌡️ Temperature: 25°C

5-Day Forecast

Day	Icon	Temp
Mon	*	26°C
Tue	rain	28°C
Wed	cloud	21°C
Thu	sun	24°C
Fri	cloud	20°C

Weather App - All In One

Home Dashboard Settings

Dashboard

User Analytics

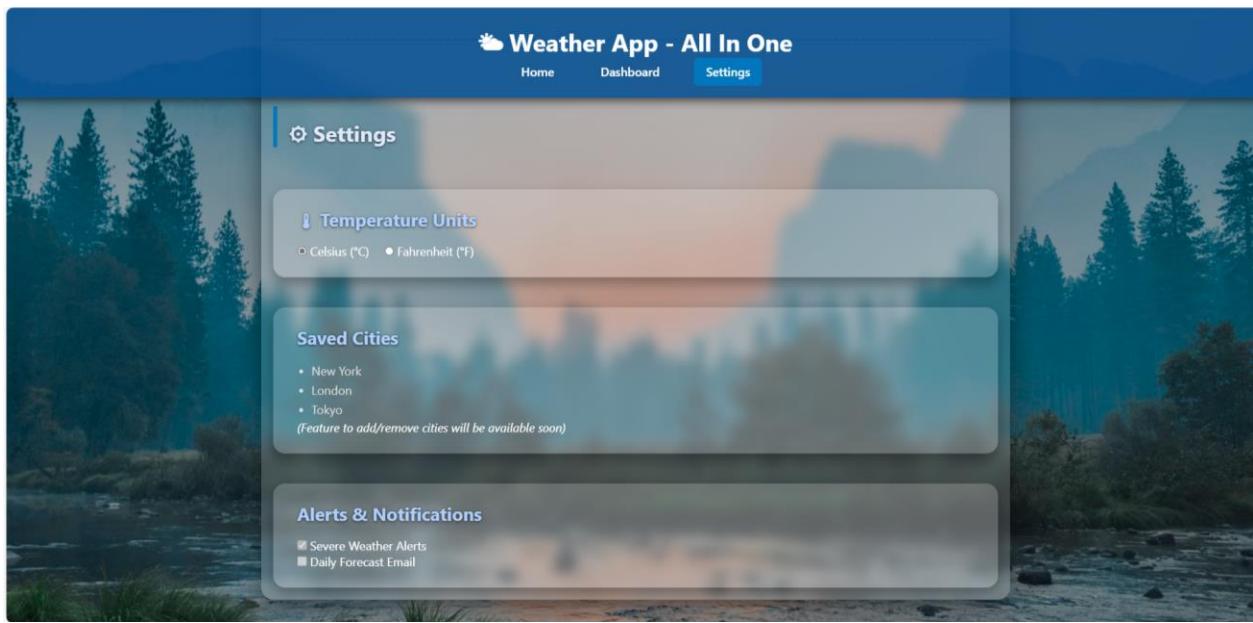
Total Registered Users: 125
Active Today: 38
Suspended Accounts: 3

Server & API Status

Uptime: **99.98%**
Response Time: 184ms
Status: **Green**

Recent Logs

- [08:31] ✓ API Call Success - Weather Data
- [08:33] △ Minor delay in forecast fetch
- [08:35] ✓ Admin login from IP 192.168.1.24

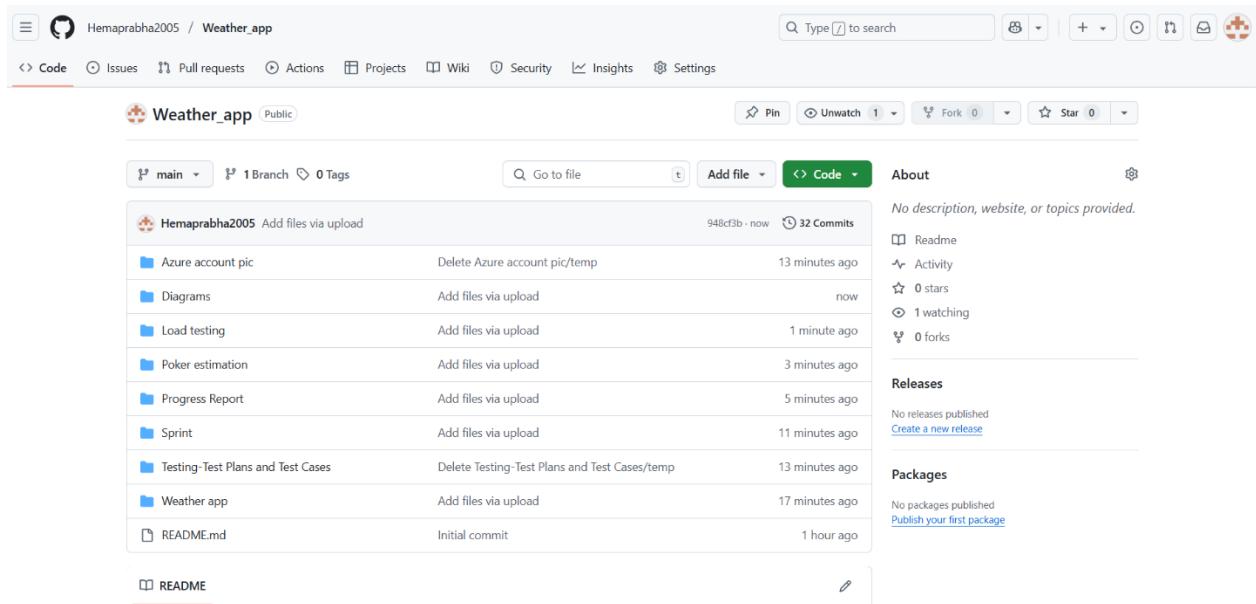


RESULT: Successfully created the Azure Load Testing resource and executed a load test to assess the performance of the specified endpoint.

Exp.No: 10	GITHUB PROJECT STRUCTURE & NAMING CONVENTIONS
Date: 02/05/2025	

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 owned by 'Hemapraba2005'. It has 1 branch and 0 tags. The main branch has 32 commits. The commits listed are:

- 948cf3b · now Add files via upload
- 13 minutes ago Delete Azure account pic/temp
- now Add files via upload
- 1 minute ago Add files via upload
- 3 minutes ago Add files via upload
- 5 minutes ago Add files via upload
- 11 minutes ago Add files via upload
- 13 minutes ago Delete Testing-Test Plans and Test Cases/temp
- 17 minutes ago Add files via upload
- 1 hour ago Initial commit

The repository has no description, website, or topics provided. It has 0 stars, 1 watching, and 0 forks. There are no releases or packages published.

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.