EXP NO: 1	AZURE DEVOPS ENVIRONMENT SETUP

Aim:

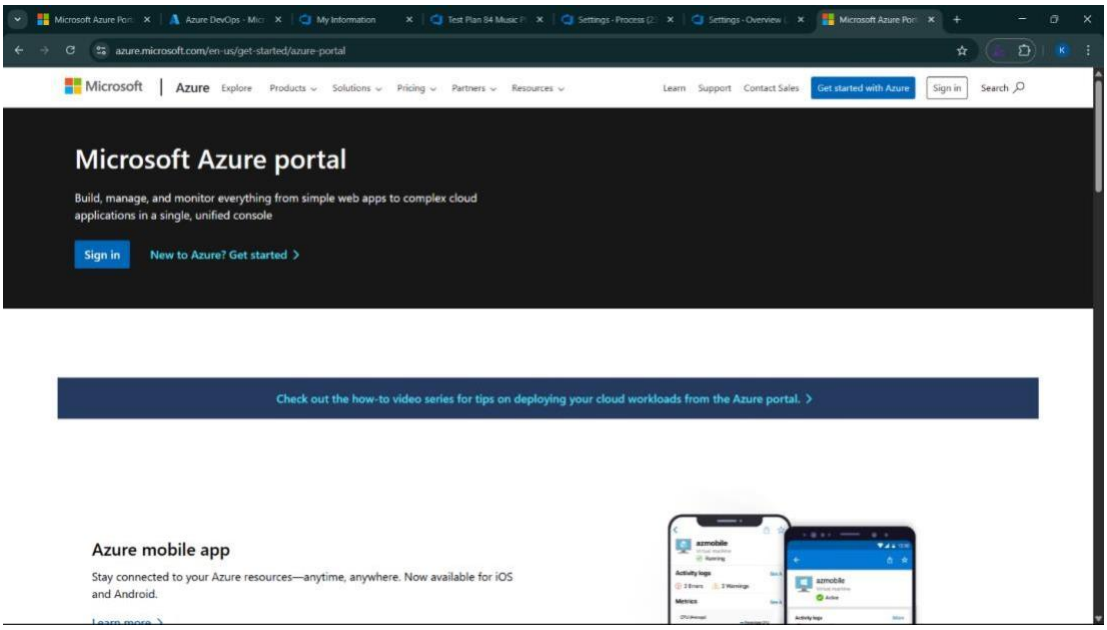
To set up and access the Azure DevOps environment by creating an organization through the Azure portal.

INSTALLATION

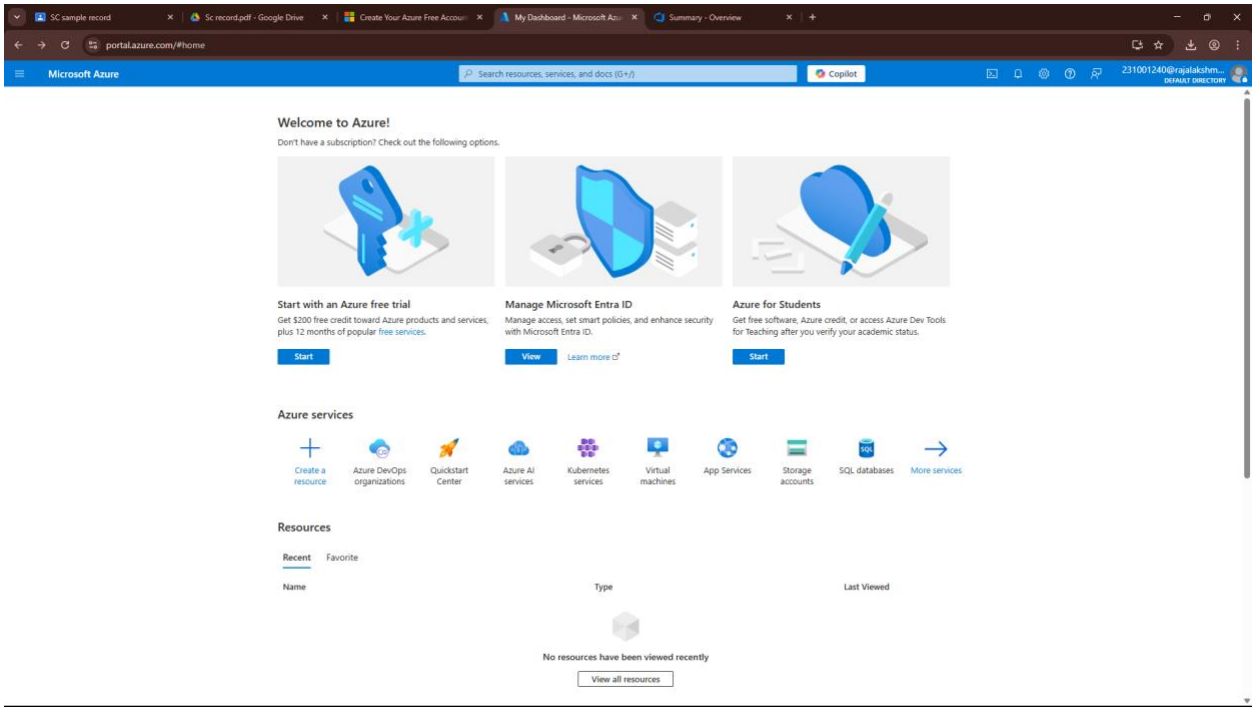
1. Open your web browser and go to the Azure website: https://azure.microsoft.com/en-us/get-started/azure-portal.

Sign in using your Microsoft account credentials.

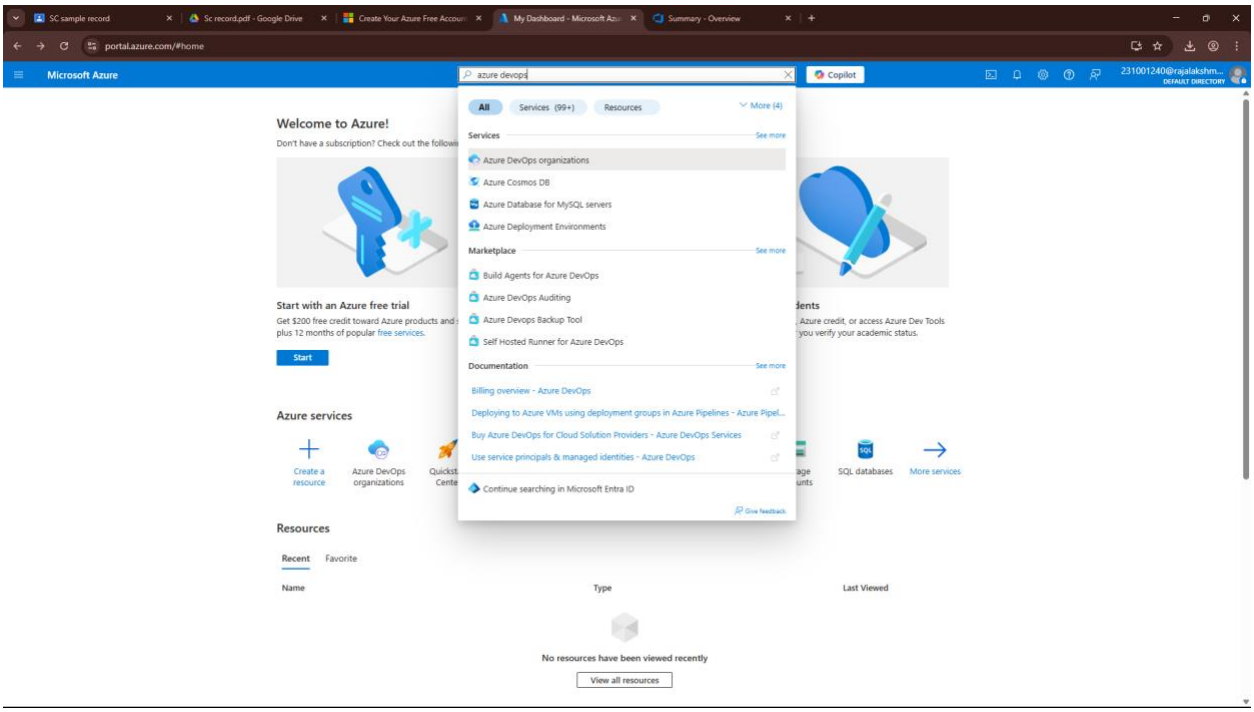
If you don't have a Microsoft account, you can create one here: https://signup.live.com/?lic=1



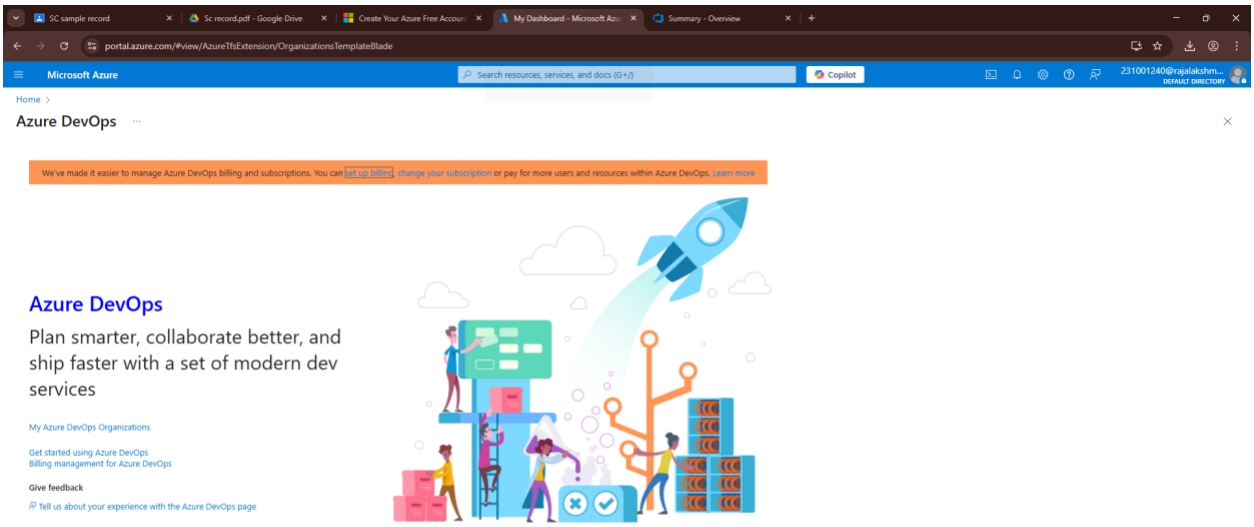
2. Azure home page



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



4. Click on the ***My Azure DevOps Organization*** link and create an organization and you should be taken to the Azure DevOps Organization Home page.



Result:

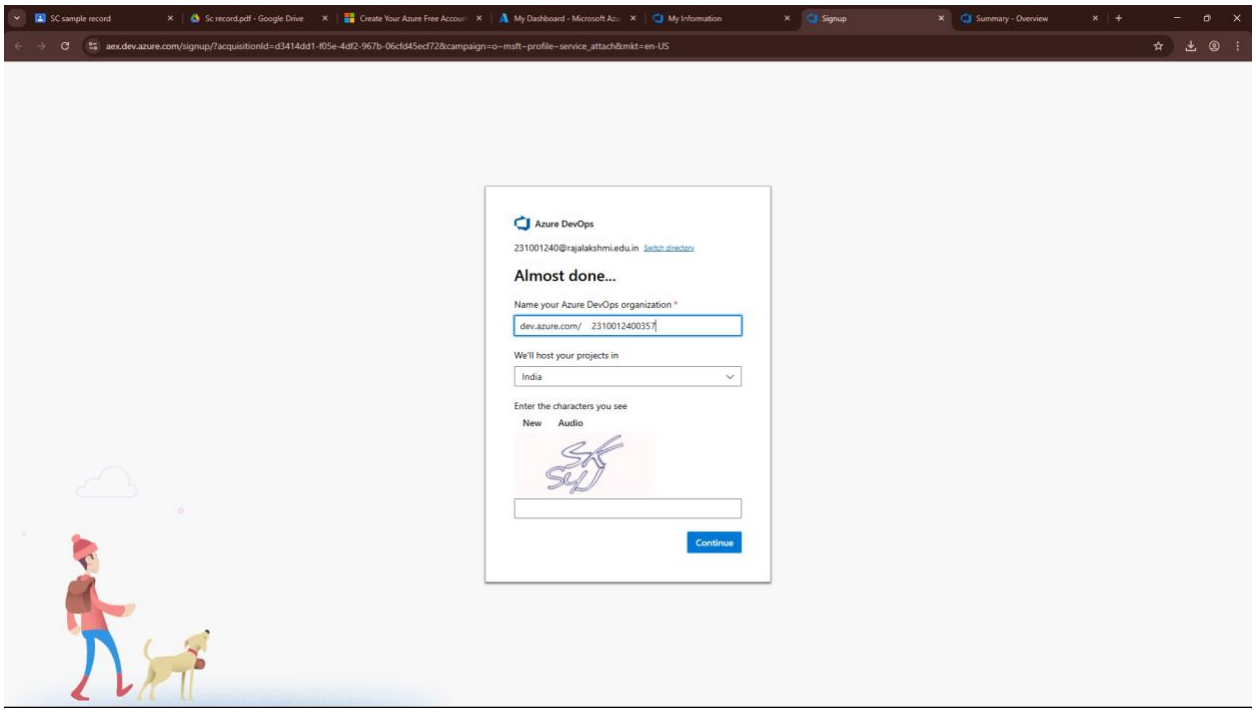
Successfully accessed the Azure DevOps environment and created a new organization through the Azure portal.

EXP NO: 2	AZURE DEVOPS PROJECT SETUP AND USER STORY MANAGEMENT

Aim:

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

1. Create An Azure Account



2. Create the First Project in Your Organization

a. After the organization is set up, you'll need to create your first **project**. This is where you'll begin to manage code, pipelines, work items, and more.

b. On the organization's **Home page**, click on the **New Project** button.

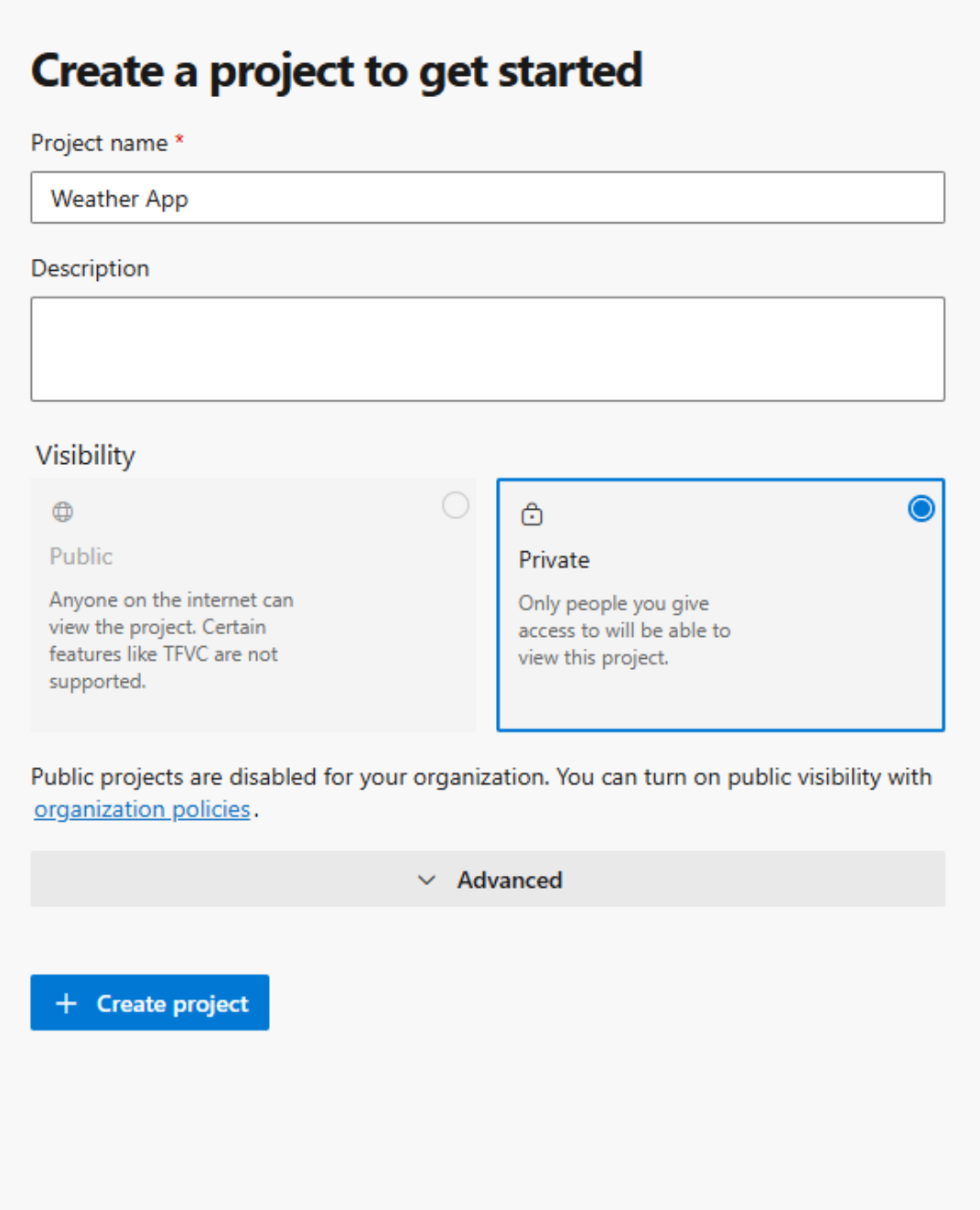
c. Enter the project name, description, and visibility options:

***Name**:* Choose a name for the project (e.g., LMS).

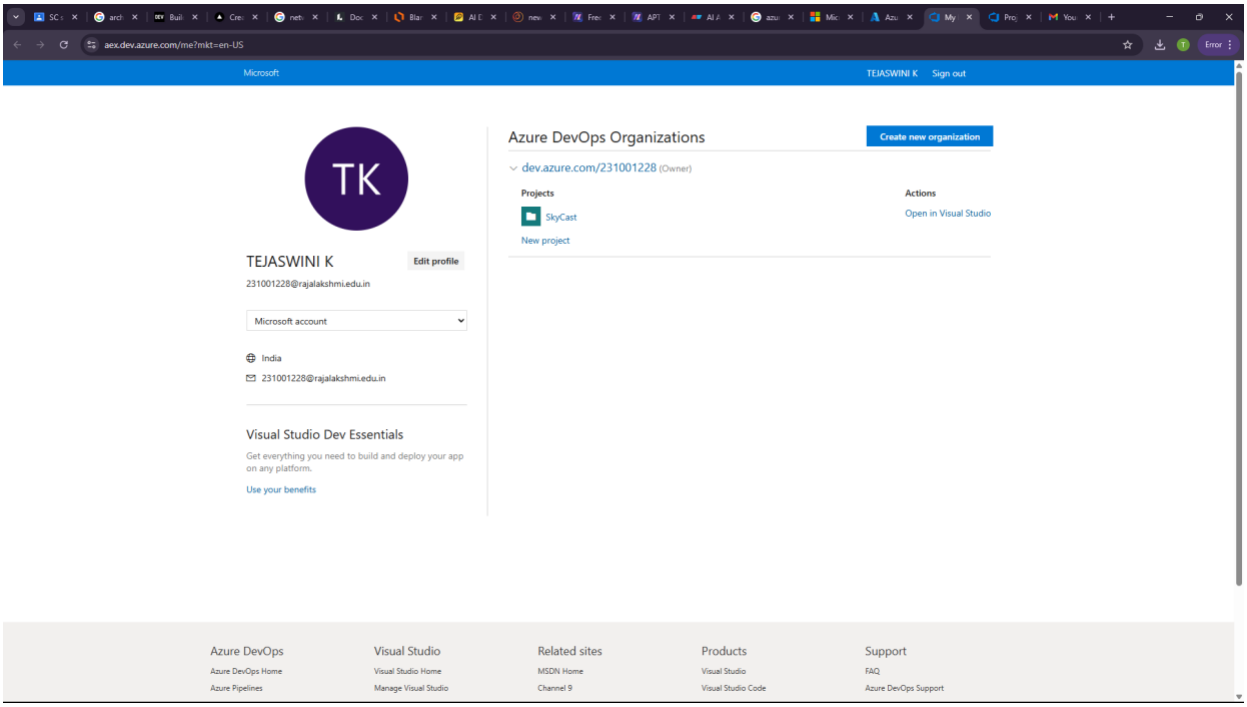
***Description**:* Optionally, add a description to provide more context about the project.

***Visibility**:* Choose whether you want the project to be **Private** (accessible only to those invited) or **Public** (accessible to anyone).

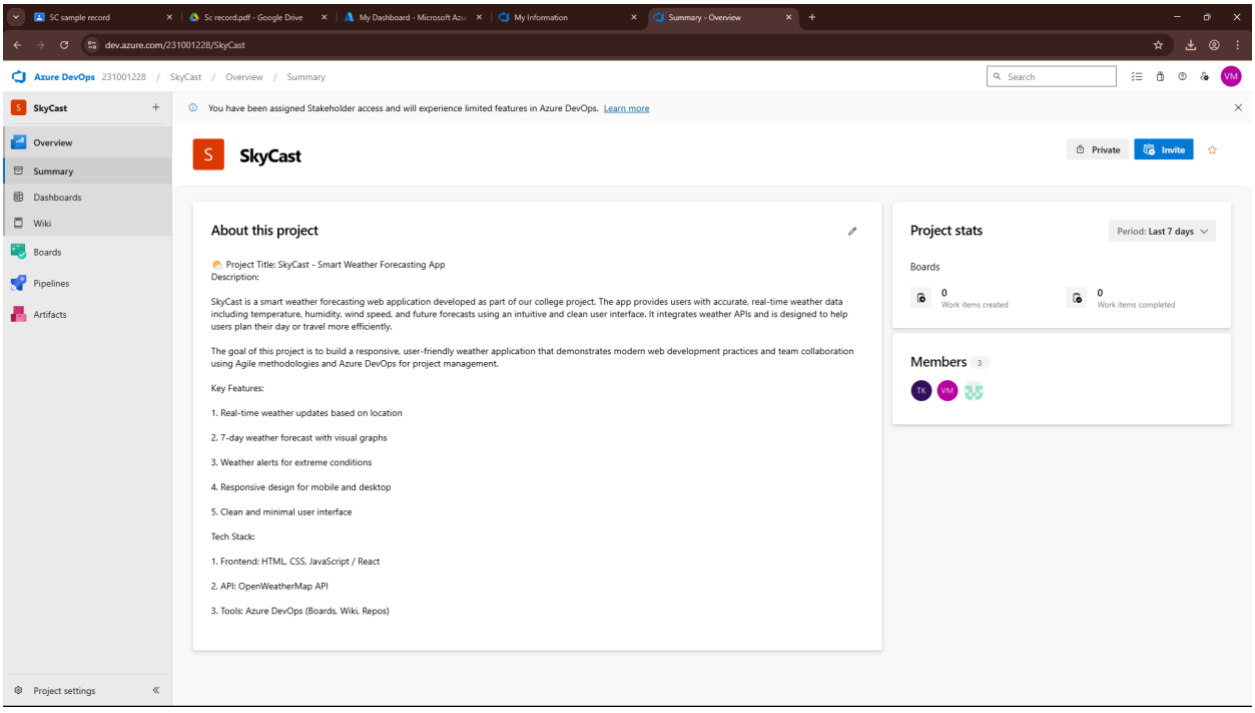
d. Once you've filled out the details, click **Create** to set up your first project.



3. Once logged in, ensure you are in the correct organization. If you're part of multiple organizations, you can switch between them from the top left corner (next to your user profile). Click on the Organization name, and you should be taken to the Azure DevOps Organization Home page.



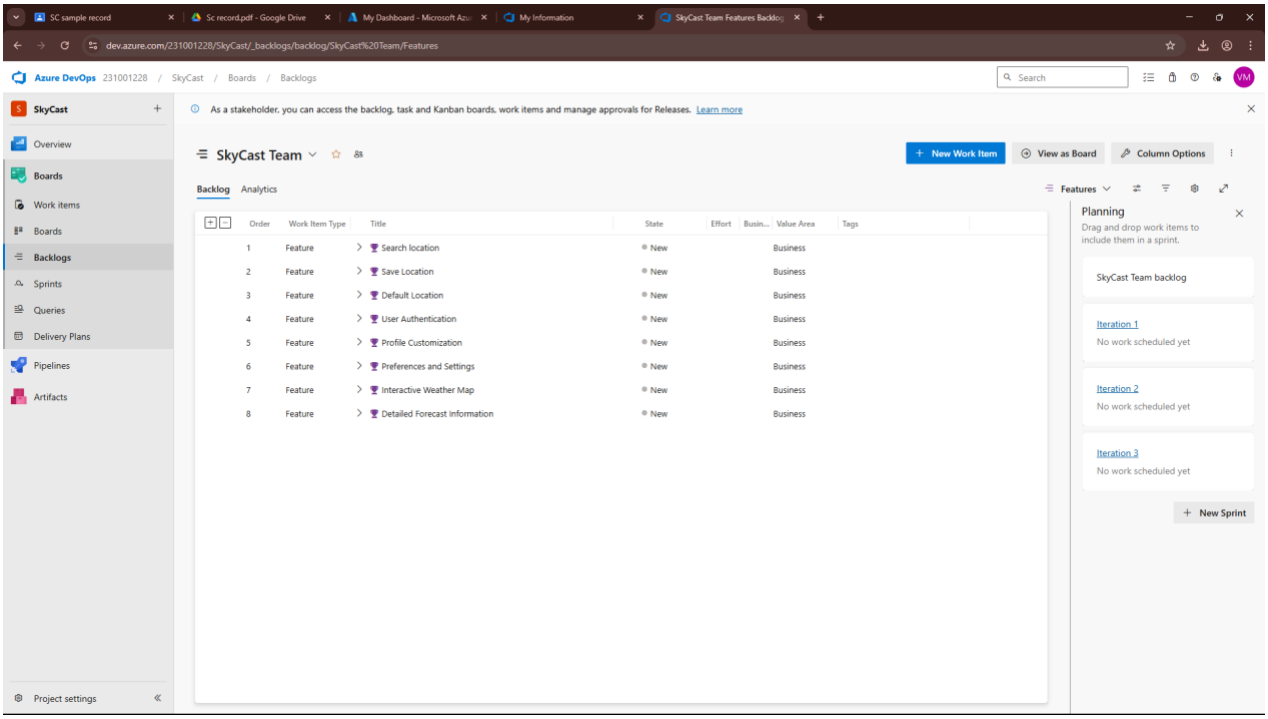
4. Project dashboard

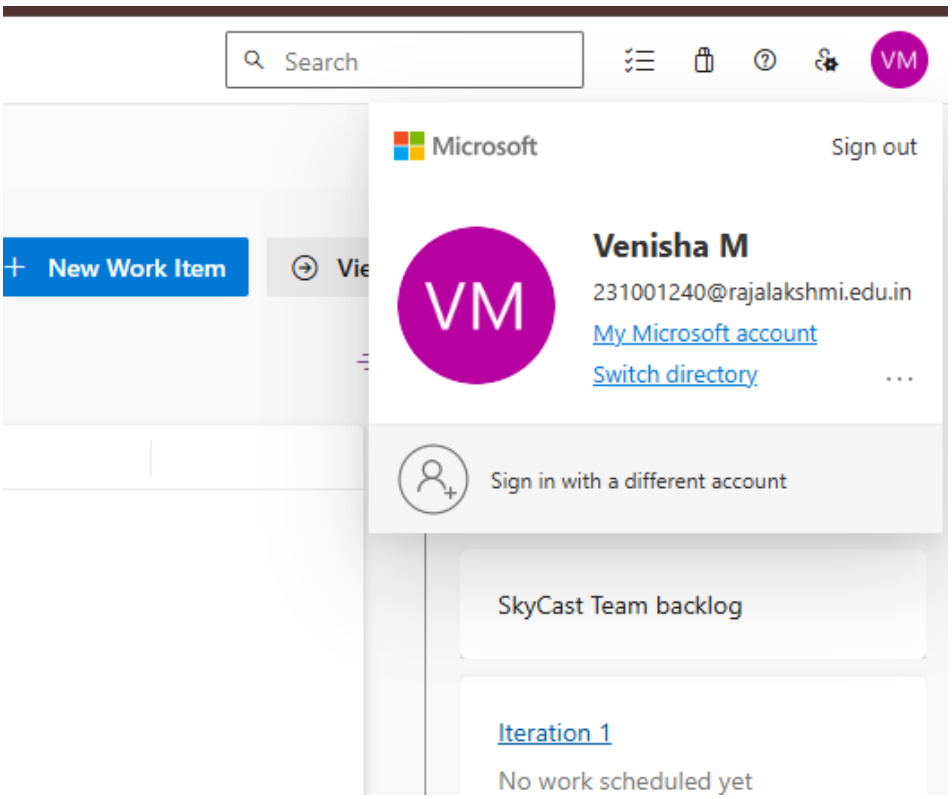


5. To manage user stories:

a. From the **left-hand navigation menu**, click on **Boards**. This will take you to the main **Boards** page, where you can manage work items, backlogs, and sprints.

b. On the **work items** page, you'll see the option to **Add a work item** at the top. Alternatively, you can find a + button or **Add New Work Item** depending on the view you're in. From the **Add a work item** dropdown, select **User Story**. This will open a form to enter details for the new User Story.





Result:

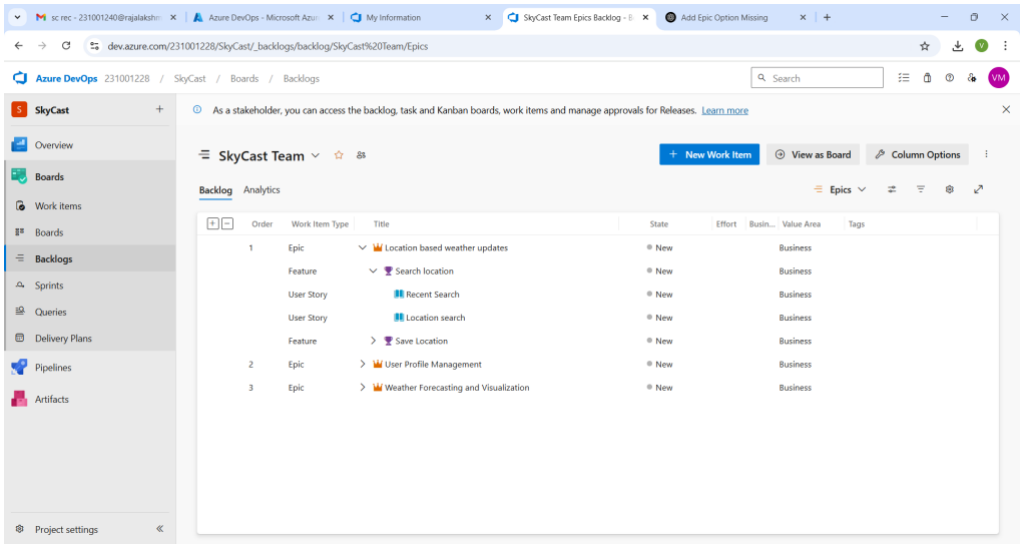
Successfully created an Azure DevOps project with user story management and agile workflow setup.

EXP NO: 3	SETTING UP EPICS, FEATURES, AND USER STORIES FOR PROJECT PLANNING

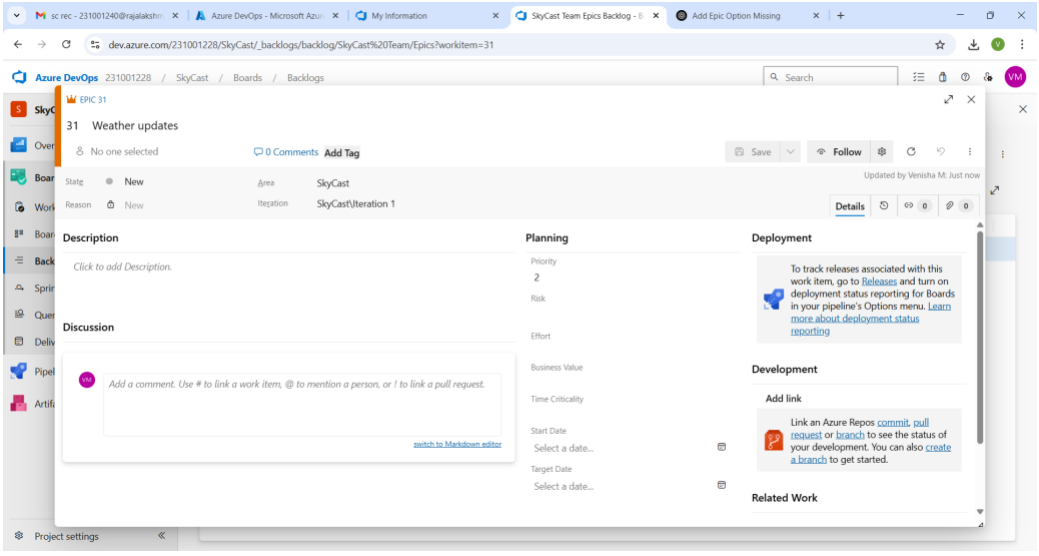
Aim:

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

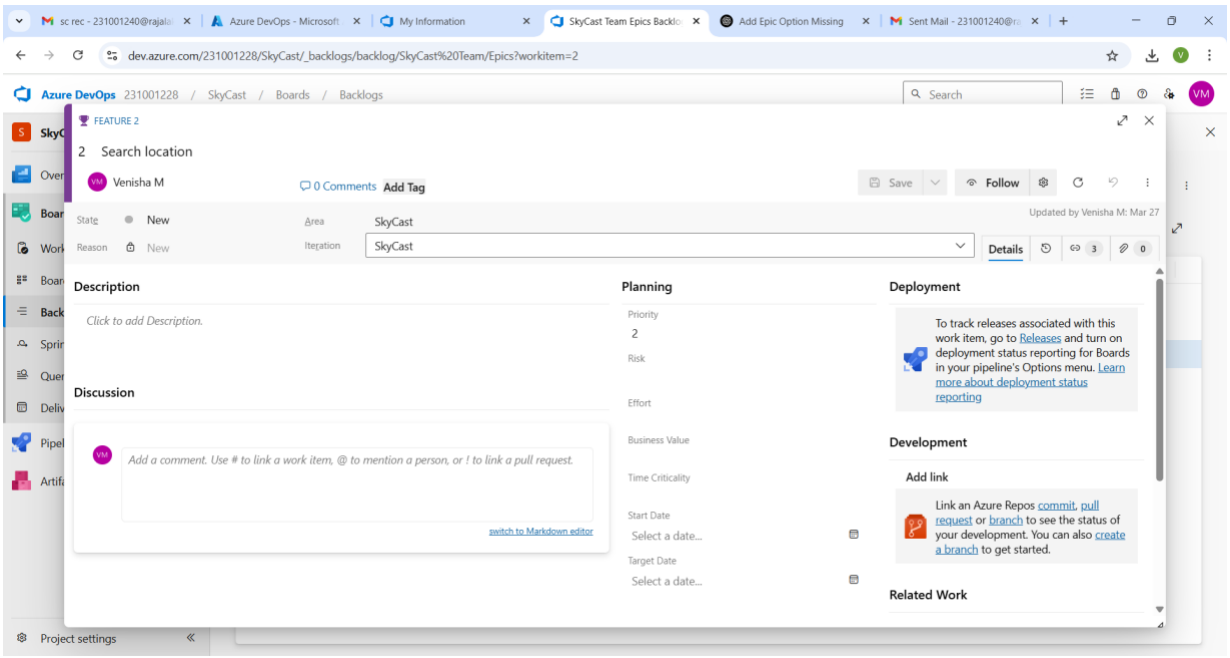
Create Epic, Features, User Stories, Task



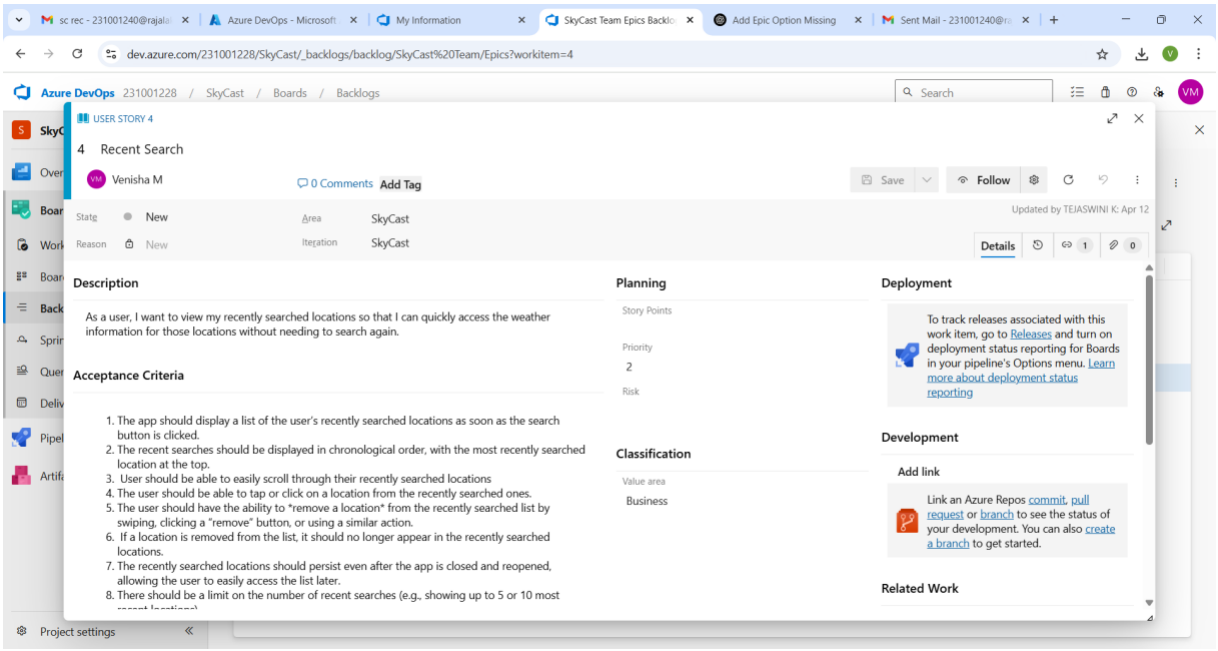
1. Fill in Epics



2. Fill in Features



3. Fill in User Story Details



Result:

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

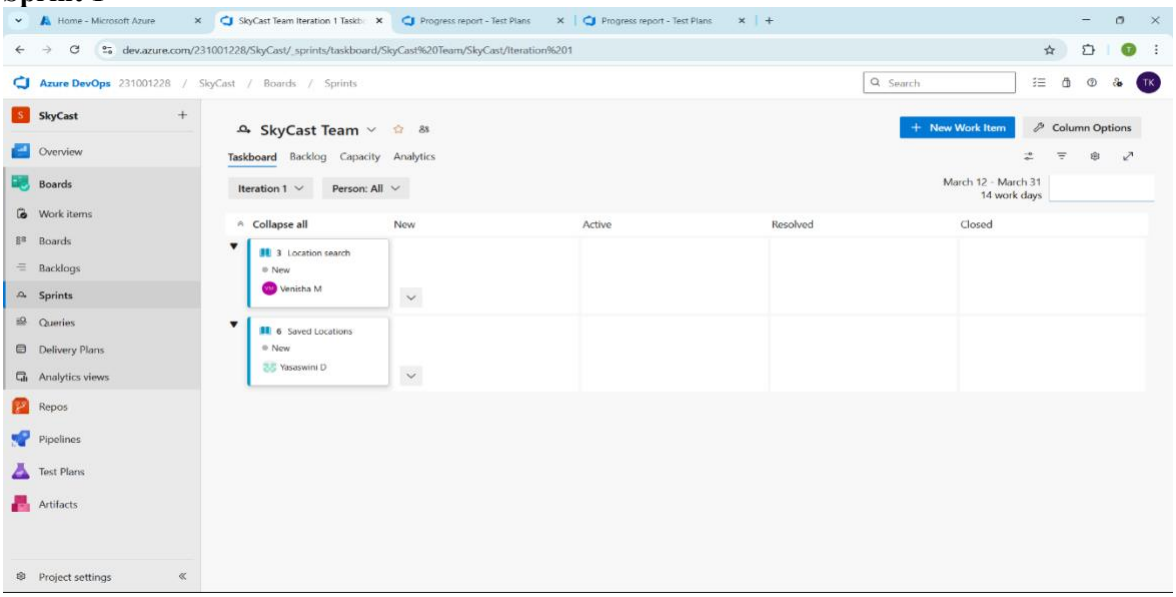
EXP NO: 4	SPRINT PLANNING

Aim:

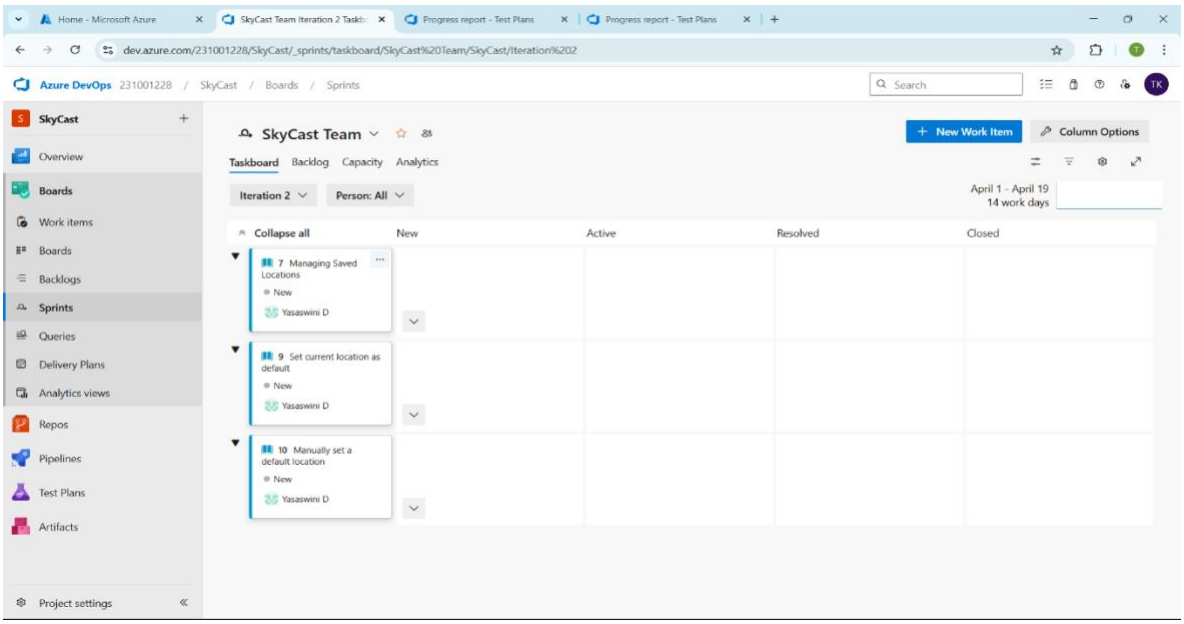
To assign user story to specific sprint for the Weather Application Project.

Sprint Planning

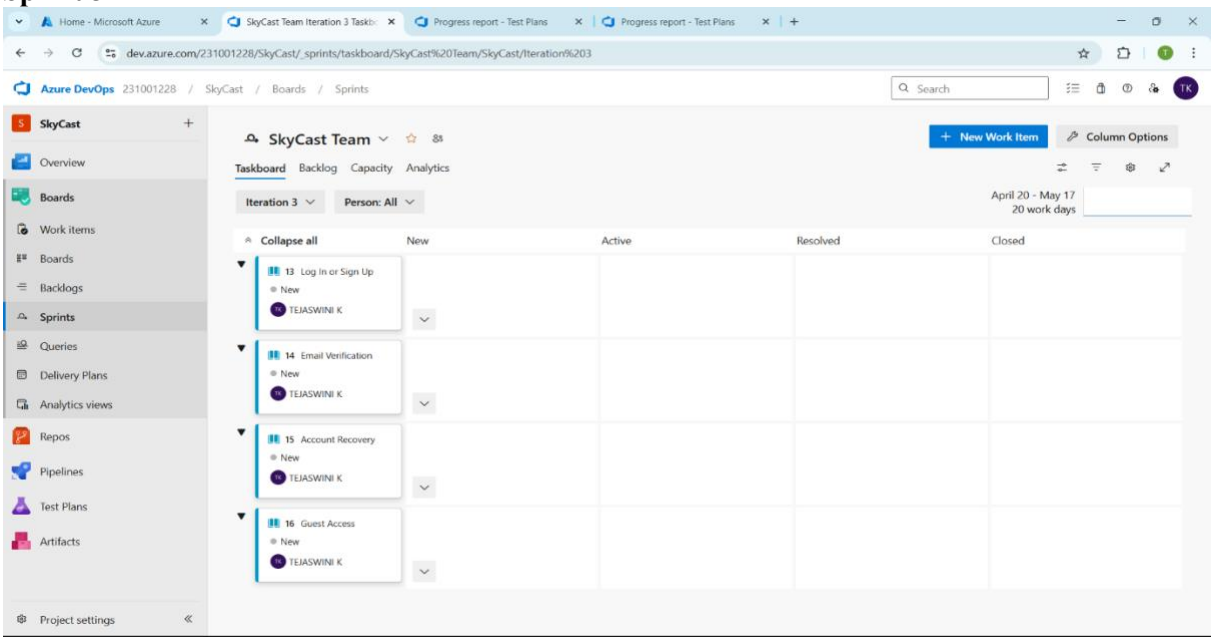
Sprint 1



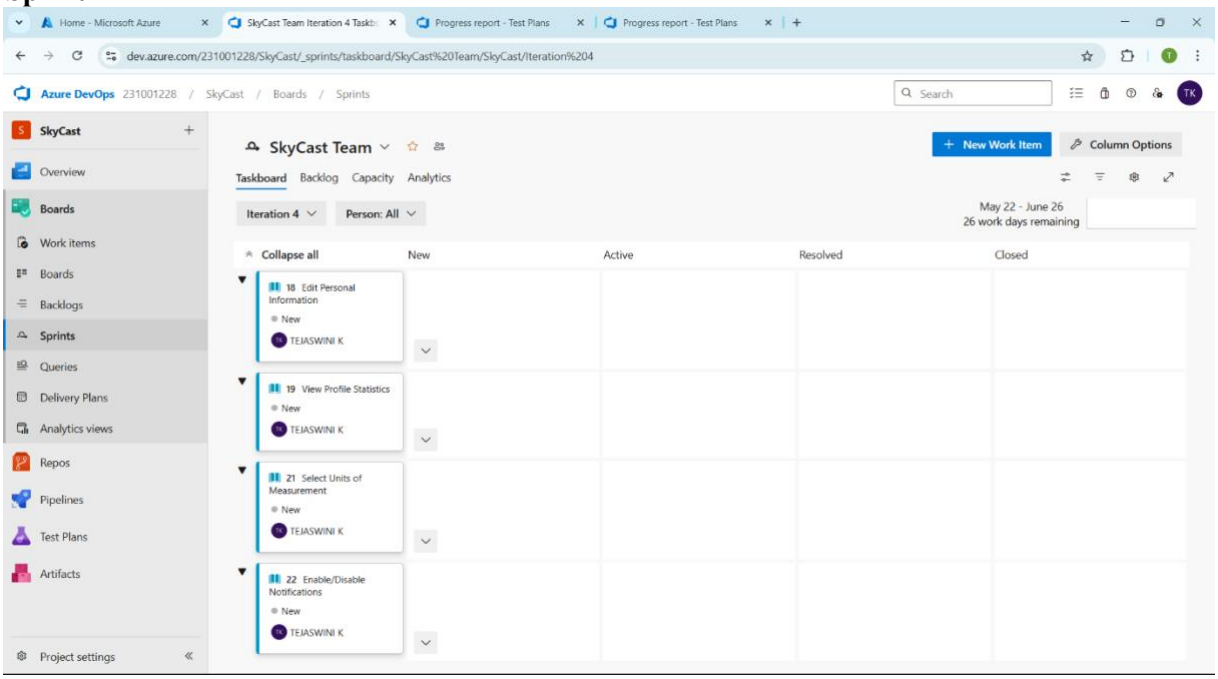
Sprint 2



Sprint 3



Sprint 4



Result:

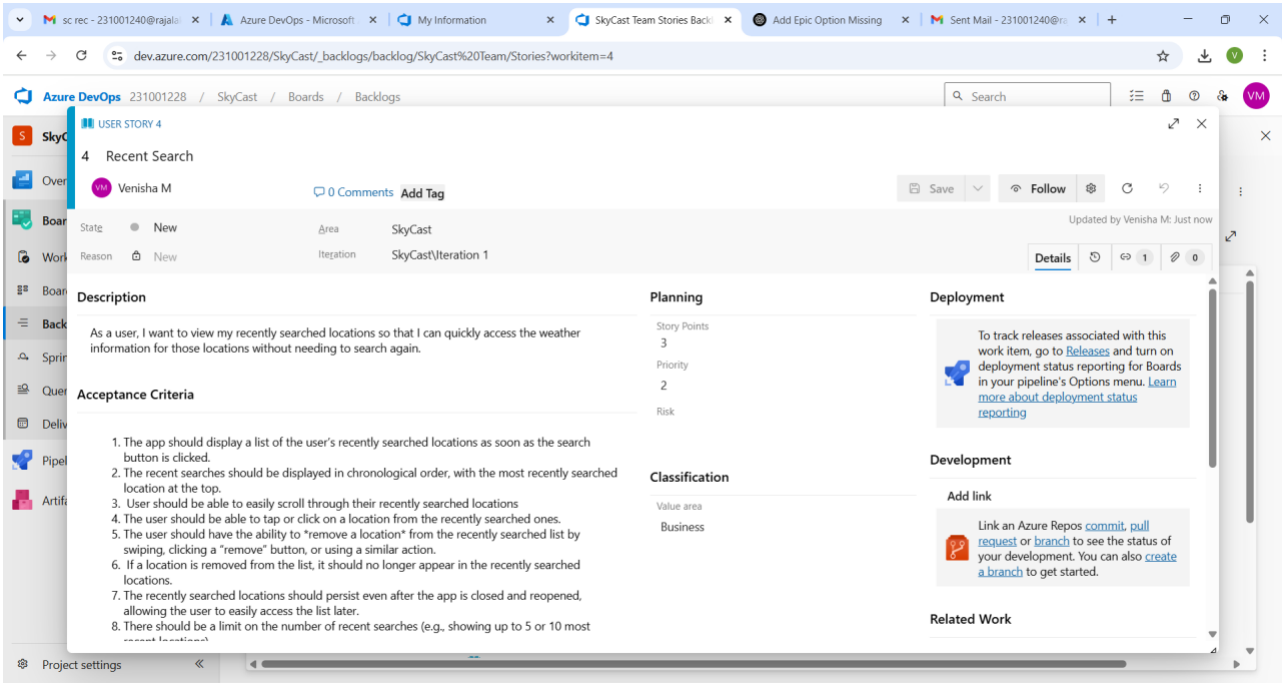
The Sprints are created for the Weather Application.

EXP NO: 5	POKER ESTIMATION

Aim:

Create Poker Estimation for the user stories – Weather Application.

Poker Estimation



Result:

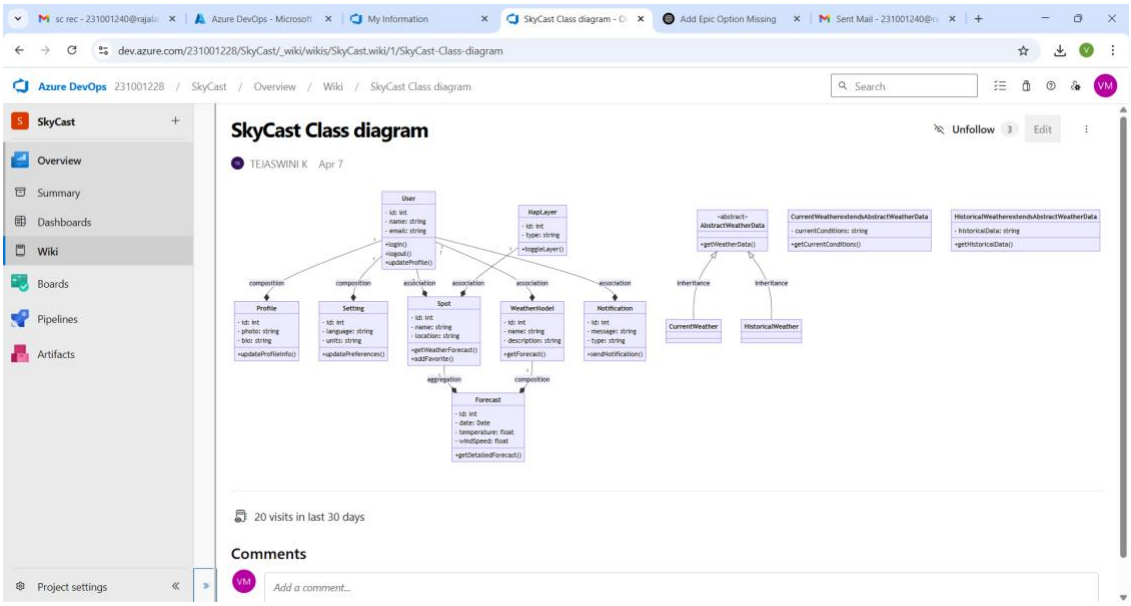
The Estimation/Story Points is created for the project using Poker Estimation.

EXP NO: 6	DESIGNING CLASS AND SEQUENCE DIAGRAMS FOR PROJECT ARCHITECTURE

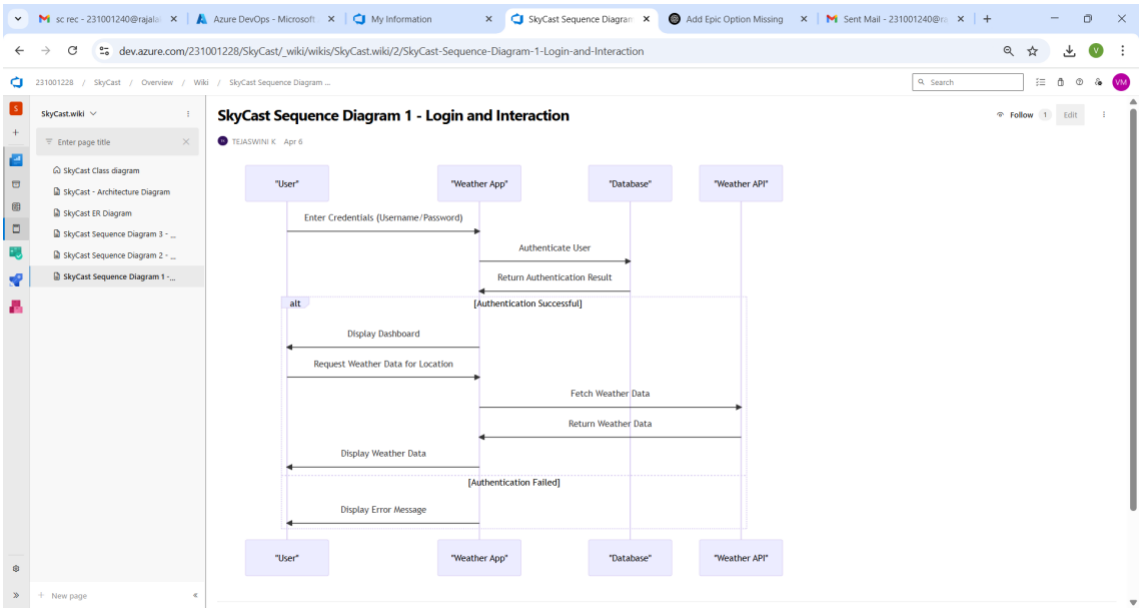
Aim:

To Design a Class Diagram and Sequence Diagram for the given Project.

6A. Class Diagram



6B. Sequence Diagram



Result:

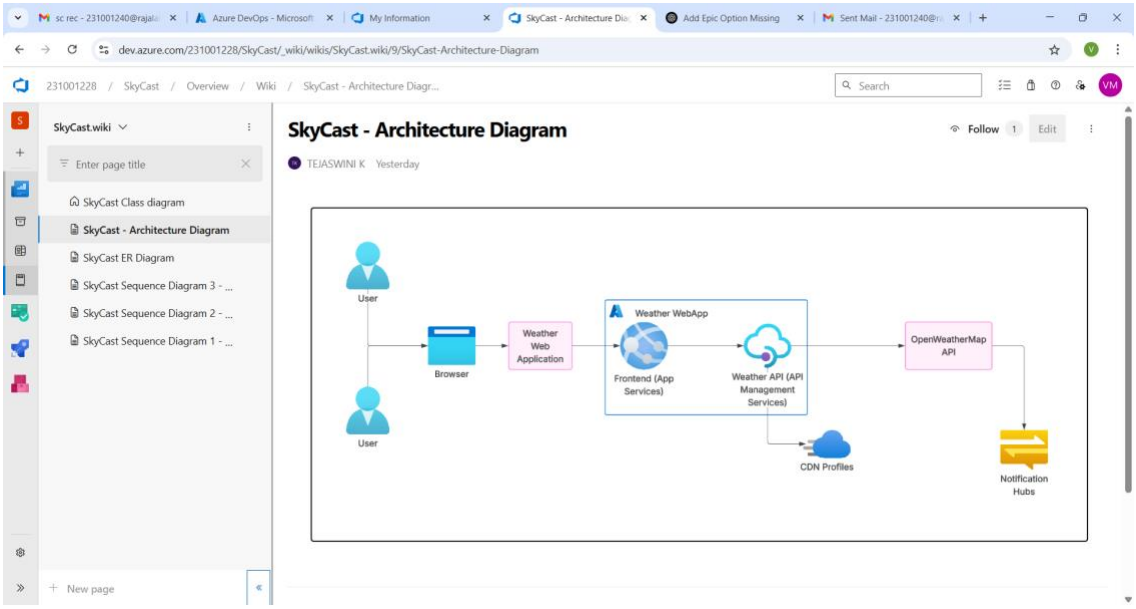
The Class Diagram and Sequence Diagram is designed Successfully for the Weather Application.

EXP NO: 7	DESIGNING ARCHITECTURAL AND ER DIAGRAMS FOR PROJECT STRUCTURE

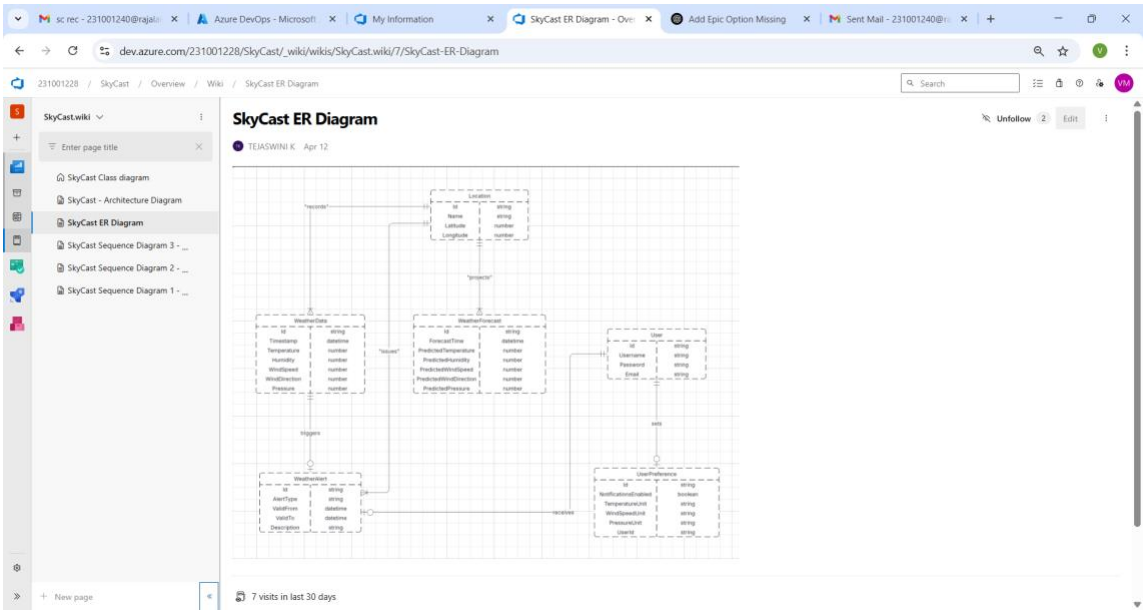
Aim:

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

7A. Architectural Diagram



7B.ER Diagram



Result:

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

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
- Enter city name to fetch weather
- Display current weather (temperature, condition)
- Show 5-day weather forecast
- Save preferred cities
- Handle invalid city names and API errors gracefully

2. Define User Interactions
Each test case simulates a real user behavior.
Examples include:
- Searching for a city
- Viewing the forecast
- Saving or retrieving a preferred city

3. Design Happy Path Test Cases
Focused on validating that all features function as expected under normal conditions.
Examples include:
- User searches for a valid city and sees weather data
- User adds a city to the preferred list successfully

4. Design Error Path Test Cases
Simulate negative or unexpected scenarios to test robustness.
Examples include:
- Weather fetch fails for an invalid city name
- API error occurs and is handled gracefully

5. Break Down Steps and Expected Results
Each test case includes step-by-step actions and the expected outcome.
This ensures clarity for testers and supports automation where needed.

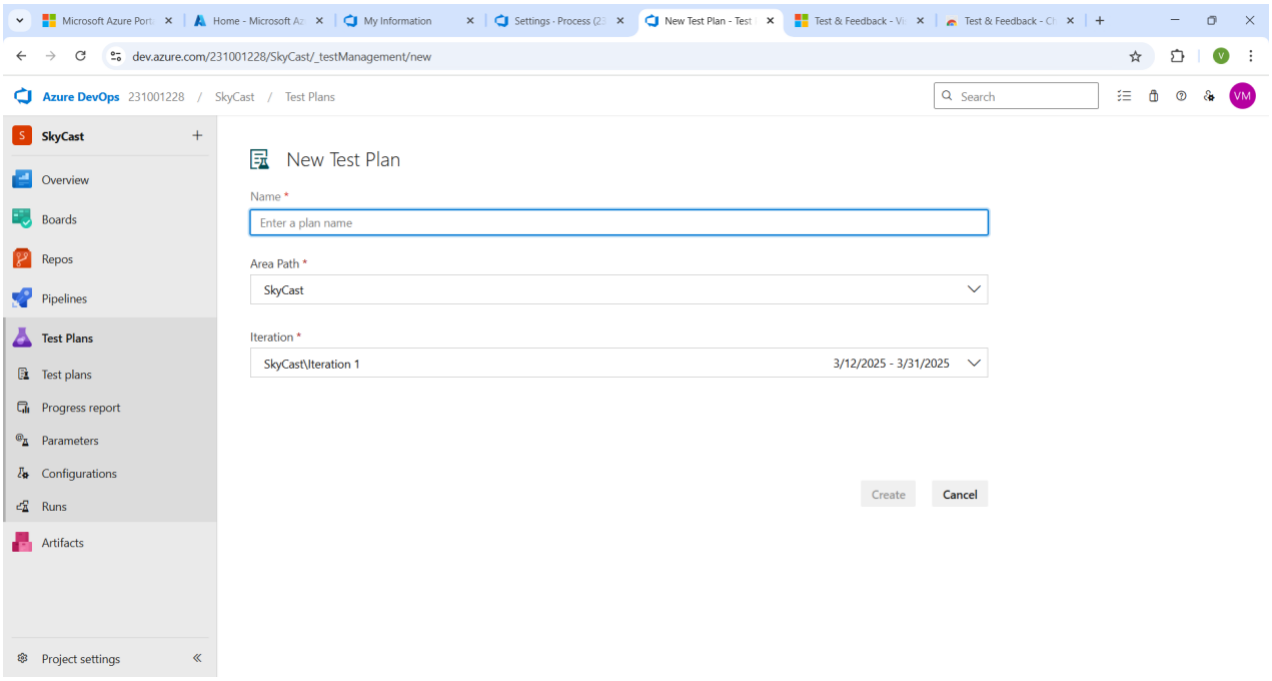
6. Use Clear Naming and IDs
Test cases are named with a clear and consistent format (e.g., TC01 – Valid City Search, TC08 – Invalid City Name Error).
This helps in quick identification and easy traceability.

7. Separate Test Suites

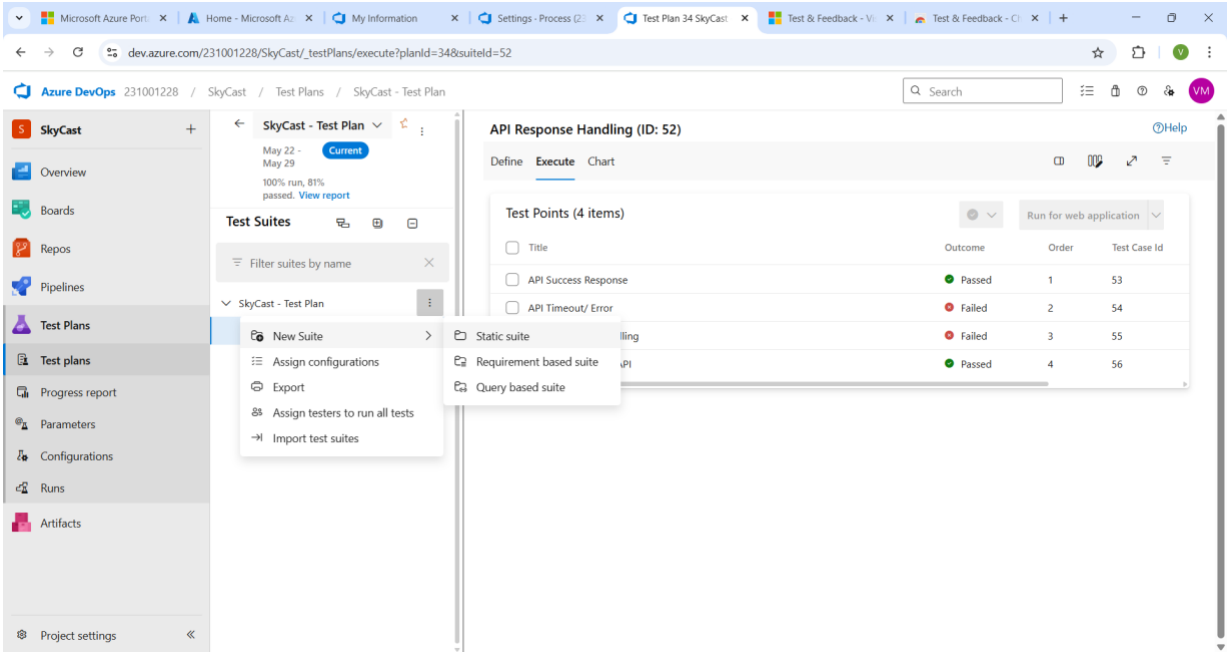
Group test cases based on functionality:

- City Search
- Weather Display
- Forecast
- Preferred City Management
- Error Handling

1. New test plan



2. Test suite



3. Test case

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

USER STORIES

- As a user, I want to sign up and log in securely so that I can access personalized features. (ID: 36).
- As a user, I want to search the weather for a specific city (ID: 37).
- As a user, I should be able to able to view accurate weather updates (ID: 47).

Test Suites

Test Suit: TS01 - User Login (ID: 36)

1. **TC01 – Successful Sign Up**
    - **Action:**
        - Go to the Sign-Up page.
        - Enter valid name, email, and password.
        - Click "Sign Up".
    - **Expected Results:**
        - Sign-Up form is displayed.
        - Fields accept values without error.
        - Account is created, and the user is redirected to the dashboard.
    - **Type**: Happy Path
2. **TC02 – Secure Login**
    - **Action:**
        - Go to the Login page.
        - Enter valid email and password.
        - Click on "Login".
    - **Expected Results:**
        - Login form is displayed.
        - Fields accept data without error.
        - User is logged in and redirected to the dashboard.
    - **Type:** Happy Path
3. **TC03 – Sign Up with Existing Email**
    - **Action:**
        - Go to the Sign-Up page.
        - Enter a name and an already registered email.
        - Click on "Sign Up".
    - **Expected Results:**

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

4. **TC04 – Login with Wrong Password**
    - **Action:**
        - Go to the Login page.
        - Enter valid email and incorrect password.
        - Click on "Login".
    - **Expected Results:**
        - Input is accepted.
        - Error message "Invalid username or password" is shown.
    - **Type:** Error Path

Test Suit: TS02 – Weather Search (ID: 37)

1. **TC05 – Search weather for valid city**
    - **Action:**
        - Log in successfully.
        - Enter “New York” in the search bar.
        - Click “Search”.
    - **Expected Results:**
        - The User is redirected to the dashboard.
        - Input is accepted.
        - Weather data for New York is displayed.
    - **Type:** Happy Path
2. **TC06 – Search weather for invalid city**
    - **Action:**
        - Log in.
        - Enter “Xyzcitynotexist” in search.
        - Click “Search”.
    - **Expected Results:**
        - User lands on dashboard.
        - Input is accepted.
        - Error message: “City not found” is shown.
    - **Type:** Error Path
3. **TC07 – Case Insensitivity in Search**
    - **Action:**
        - Log in
        - Enter “London”, “LONDON”, “London” one by one.
    - **Expected Results:**
        - User lands on dashboard.
        - In all cases, valid weather data for London is shown, confirming case-insensitive search.
    - **Type**: Happy path.
4. **TC08 – Special Characters in input**

- **Action:**
  - Log in.
  - Enter city name with special characters like “@Paris!” or “123London”.
- **Expected Results:**
  - Redirected to dashboard.
  - “City not found” message is shown.
- **Type:** Error path.

Test Suit: TS03 – Error and Edge cases (ID: 46)

1. **TC09 – Blank Search query**
   - **Action:**
     - Log in.
     - Leave Search bar empty.
     - Click “Search”.
   - **Expected Results:**
     - Redirected to Dashboard.
     - Input remains blank.
     - Warning message “Please enter a city name” is shown.
   - **Type:** Error Path
2. **TC10 – Network Loss during Operation**
   - **Action:**
     - Disable internet connection.
     - Try searching for a city.
   - **Expected Results:**
     - Website detects offline state.
     - Error message “Network offline, please check your connection is shown” is shown.
   - **Type:** Error Path

Test Suit: TS04 – Data Accuracy (ID: 47)

1. **TC11 – Verify Current Temperature accuracy**
   - **Action:**
     - User searches for a valid city.
     - System fetches temperature data from the weather API.
     - App displays the temperature on screen.
   - **Expected Results:**
     - Weather data is displayed for Mumbai.
     - API response includes a valid temperature field.

- Displayed temperature matches the API value.

- **Type:** Happy path

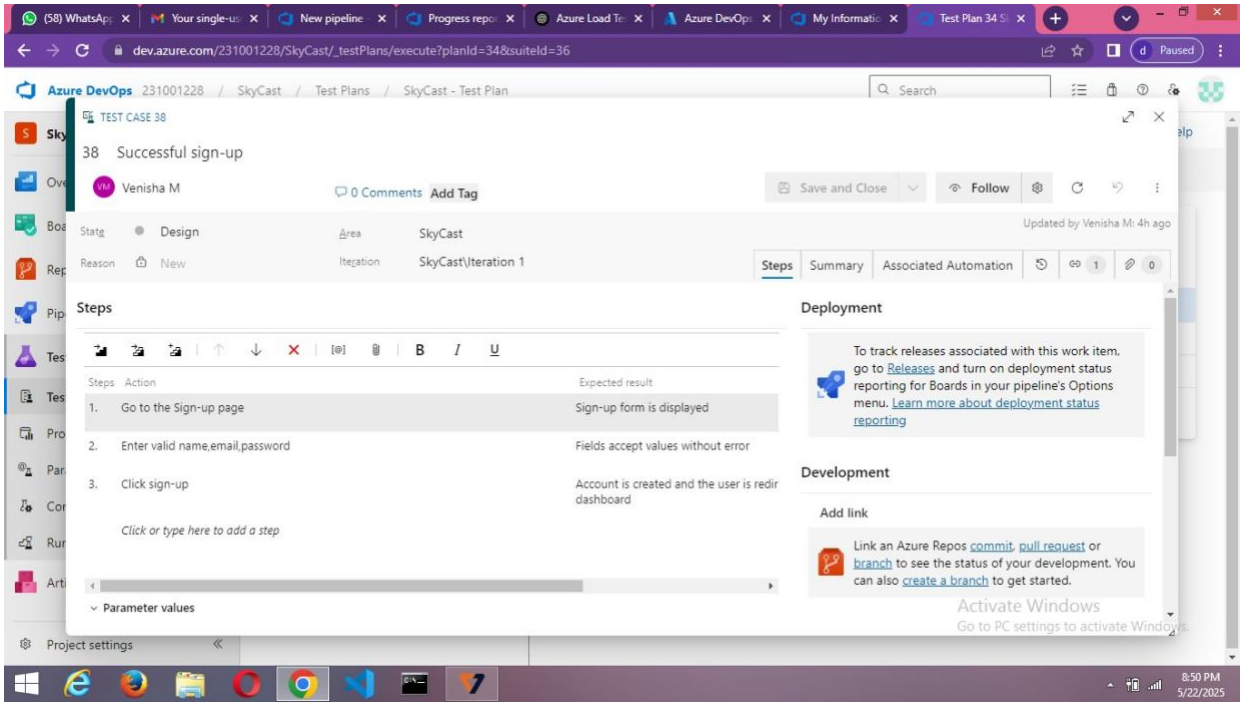
2. **TC12 – Verify humidity and wind speed accuracy**
   - **Action:**
     - User selects a city (e.g., "Dubai") and views details.
     - App retrieves humidity and wind speed from API.
     - App displays these values on screen.
   - **Expected Results:**
     - Humidity and wind speed values are shown.
     - API returns accurate values (e.g., 70% humidity, 12 km/h)
     - Value match API exactly and use correct units.
   - **Type:** Happy Path

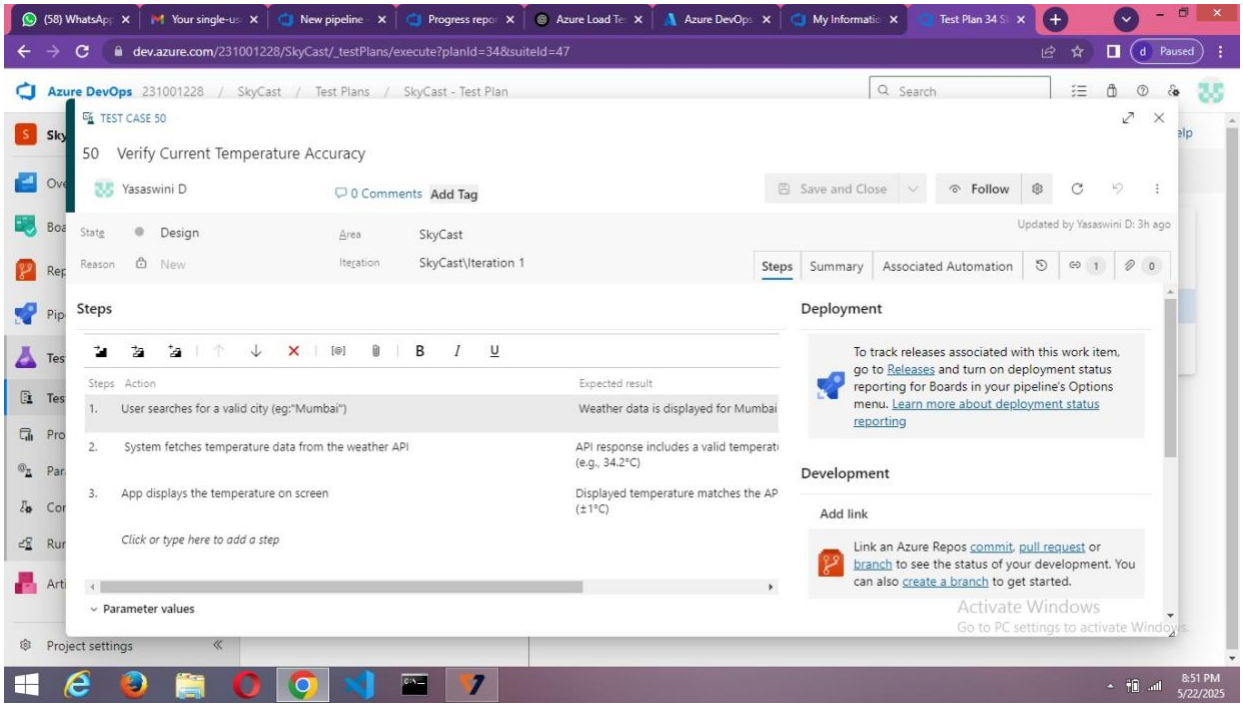
Test Suit: TS045 – API Response handling (ID: 52)

1. **TC13 – API Success Response**
   - **Action:**
     - Log in.
     - Search for a valid city (e.g., "Paris").
     - Weather data loads on the screen.
   - **Expected Results:**
     - Dashboard is displayed.
     - API request is made and a 200 OK response is received.
     - Weather details (temperature, humidity, condition) for Paris are correctly rendered
   - **Type:** Happy Path
2. **TC14– API Timeout/ Error**
   - **Action:**
     - Simulate network latency or backend delay.
     - Try searching for any city.
   - **Expected Results:**
     - Frontend continues to wait for a response.
     - After timeout, an error message "Request timed out. Please try again." is shown.
   - **Type:** Error Path
3. **TC15 – Invalid API key handling**
   - **Action:**
     - Replace the API key with an invalid one or simulate expired key.
     - Search for any city.
   - **Expected Results:**
     - API returns 401 unauthorized.
     - Error message like "Authentication failed. Please contact support." is displayed.
   - **Type:** Error Path

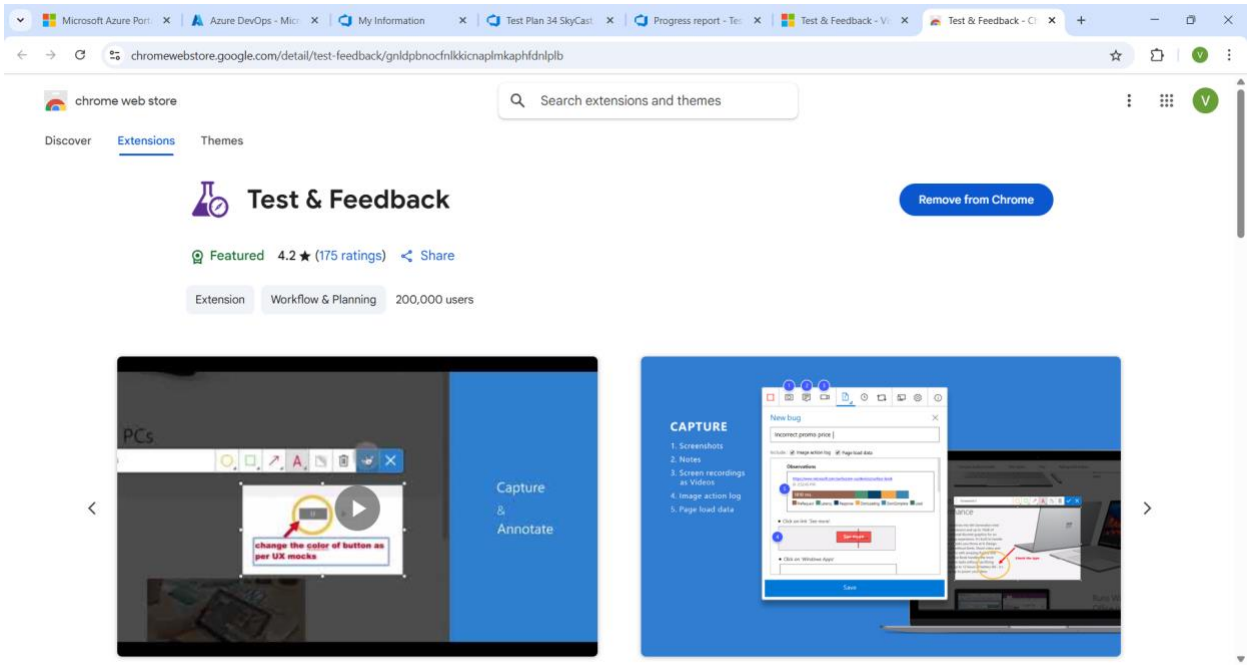
4. **TC16 – No data for city in API**
    - **Action:**
        - Search for a real but obscure location.
        - App processes the response.
    - **Expected Results:**
        - API returns empty or incomplete data.
        - Message “Weather data unavailable for this location is shown.
    - **Type:** Error Path

Test Cases

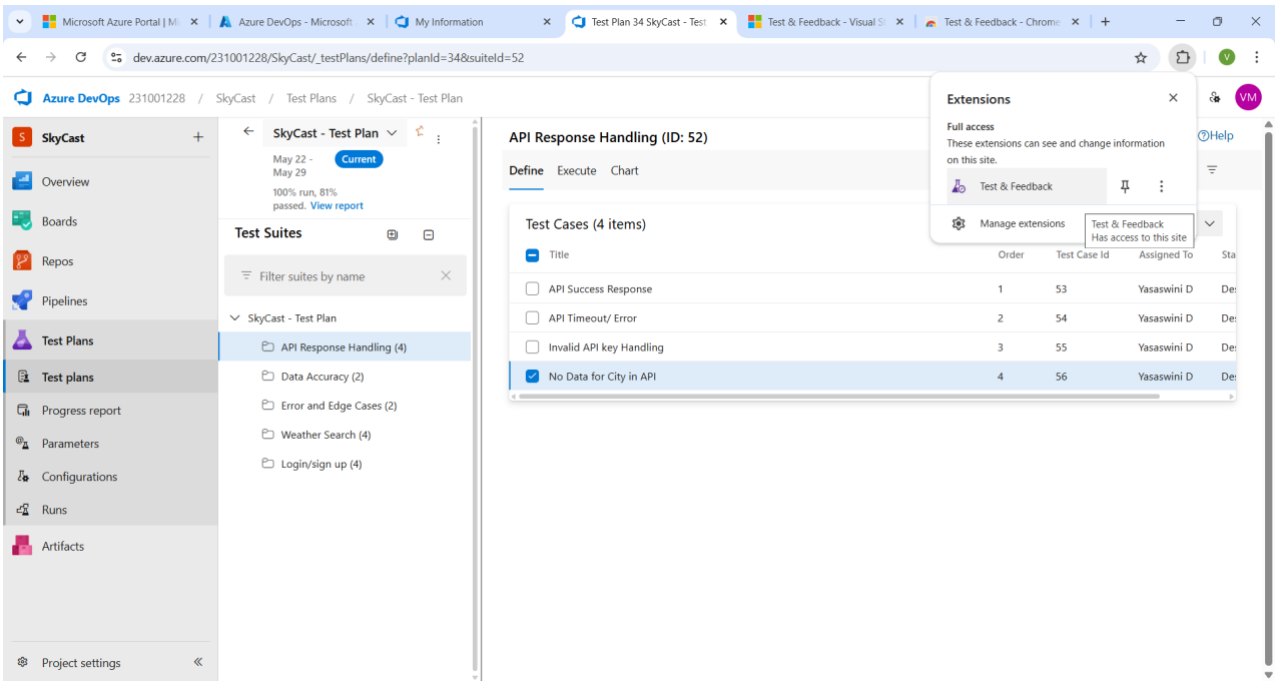

dev.azure.com/231001228/SkyCast/_testPlans/execute?planId=34&suiteId=36
Azure DevOps 231001228 / SkyCast / Test Plans / SkyCast - Test Plan
TEST CASE 38
38 Successful sign-up
Venisha M
0 Comments Add Tag
Save and Close
Follow
State Design
Area SkyCast
Reason New
Iteration SkyCast\Iteration 1
Updated by Venisha M: 4h ago
Steps Summary Associated Automation
Steps
Steps Action Expected result
1. Go to the Sign-up page Sign-up form is displayed
2. Enter valid name,email,password Fields accept values without error
3. Click sign-up Account is created and the user is redir dashboard
Click or type here to add a step
Parameter values
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.
Activate Windows
Go to PC settings to activate Windows.
Project settings
8:50 PM
5/22/2025


dev.azure.com/231001228/SkyCast/_testPlans/execute?planId=34&suiteId=47
Azure DevOps 231001228 / SkyCast / Test Plans / SkyCast - Test Plan
TEST CASE 50
50 Verify Current Temperature Accuracy
Yasaswini D
0 Comments Add Tag
Save and Close
Follow
State Design
Area SkyCast
Reason New
Iteration SkyCast\Iteration 1
Updated by Yasaswini D: 3h ago
Steps Summary Associated Automation
Steps
Steps Action Expected result
1. User searches for a valid city (eg:"Mumbai") Weather data is displayed for Mumbai
2. System fetches temperature data from the weather API API response includes a valid temperat (e.g., 34.2°C)
3. App displays the temperature on screen Displayed temperature matches the AP (±1°C)
Click or type here to add a step
Parameter values
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.
Activate Windows
Go to PC settings to activate Windows.
Project settings
8:51 PM
5/22/2025

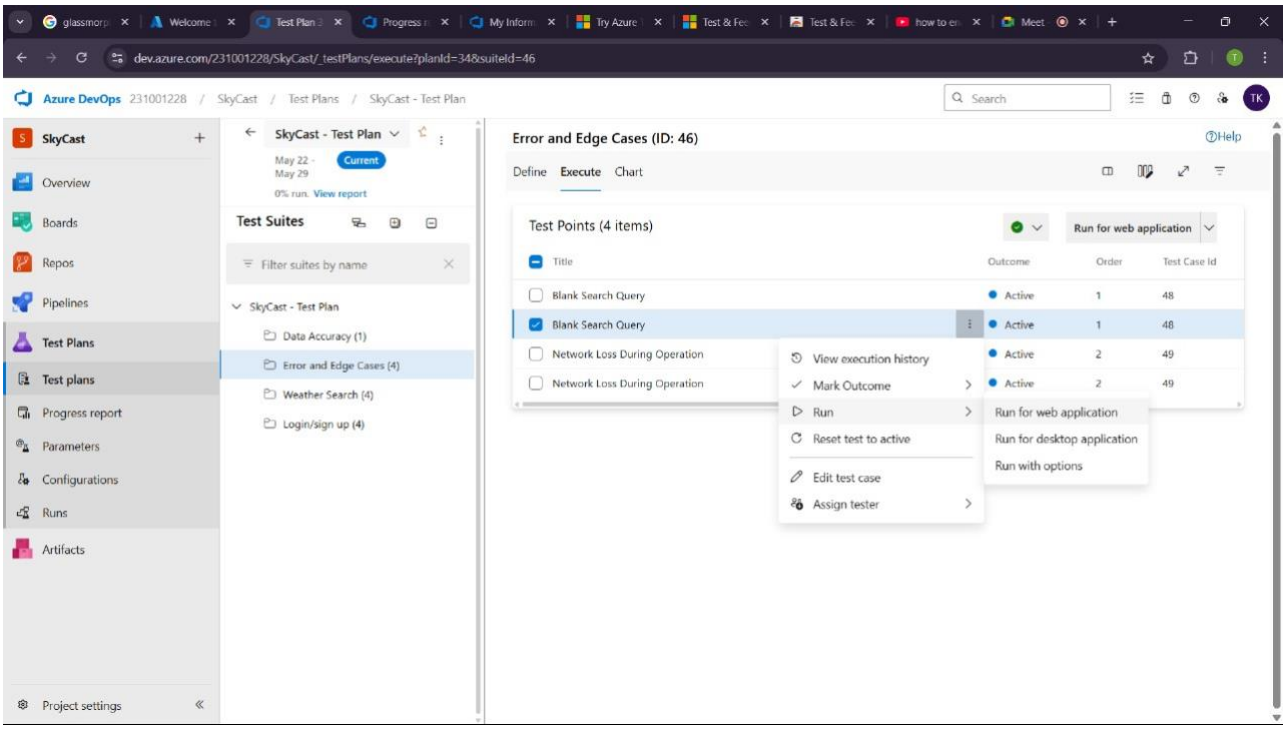
4. Installation of test

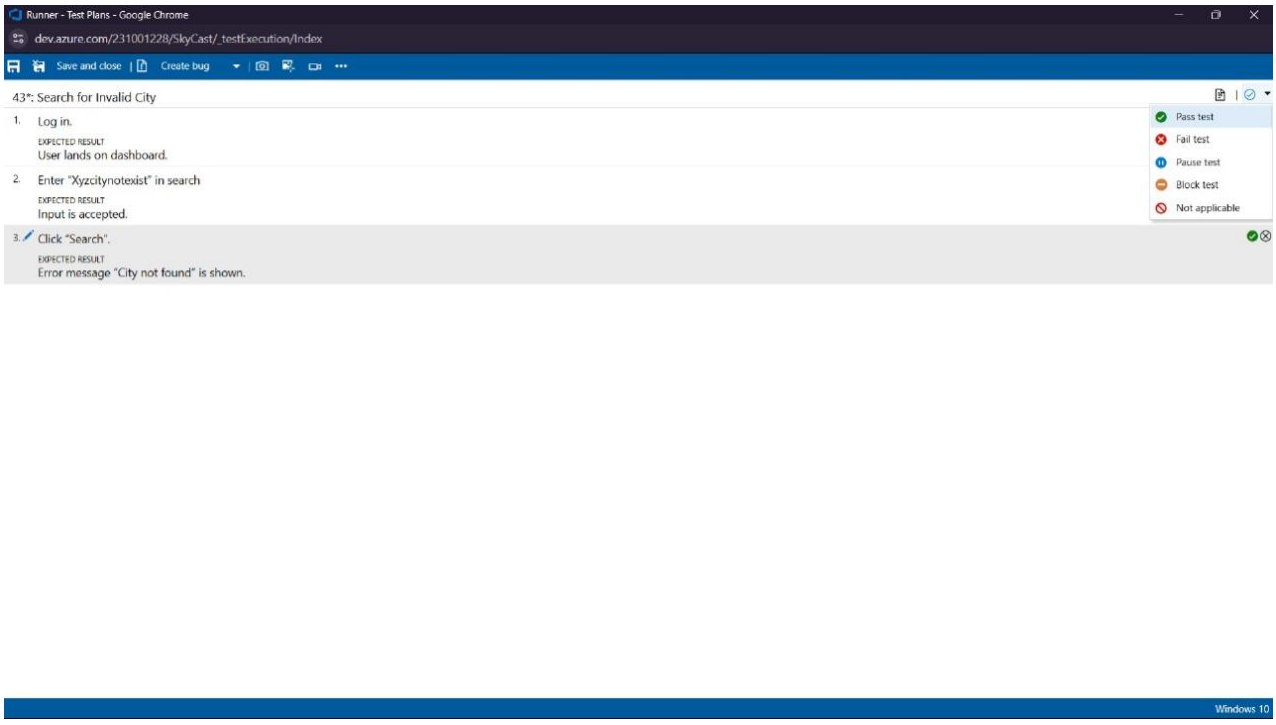


Test and feedback Showing it
as an extension

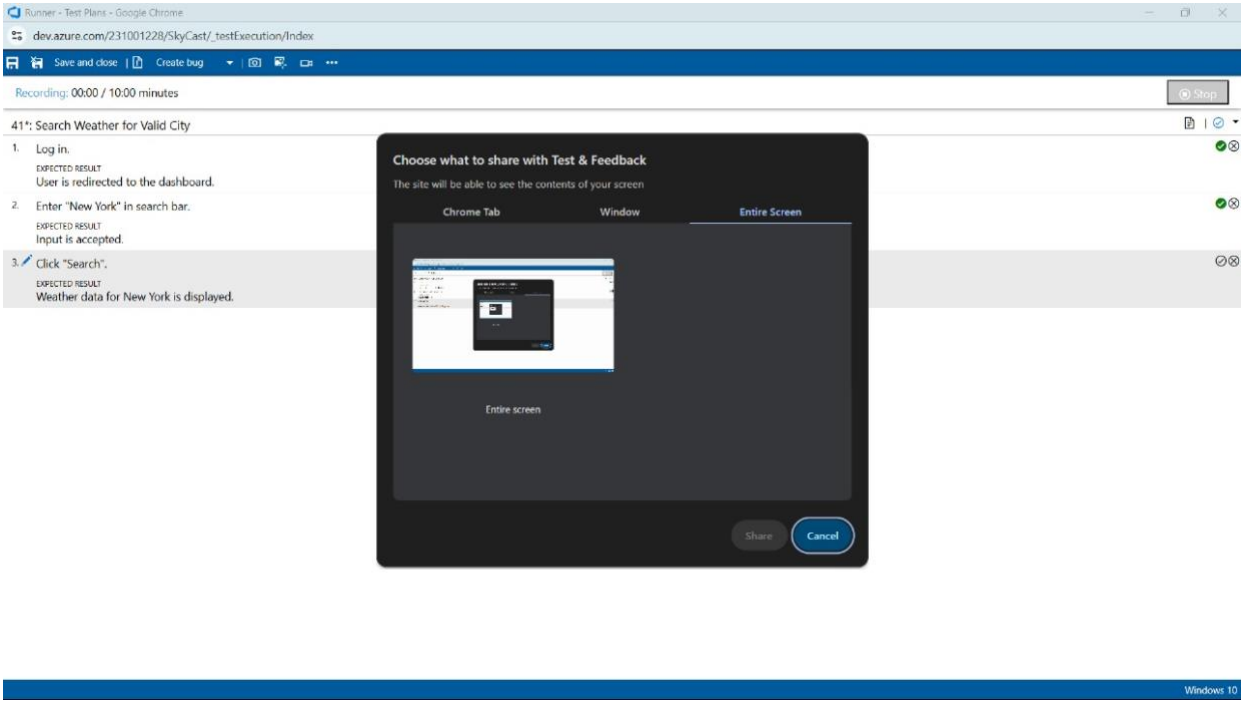


5. Running the test cases

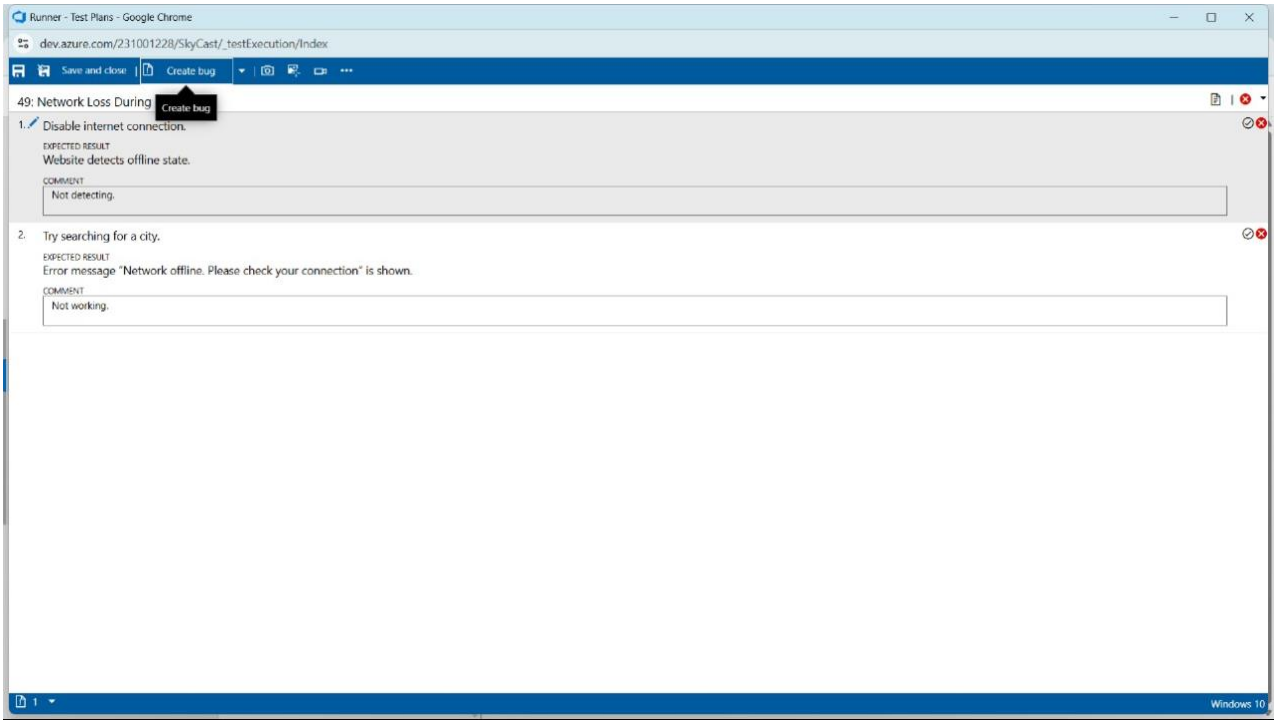


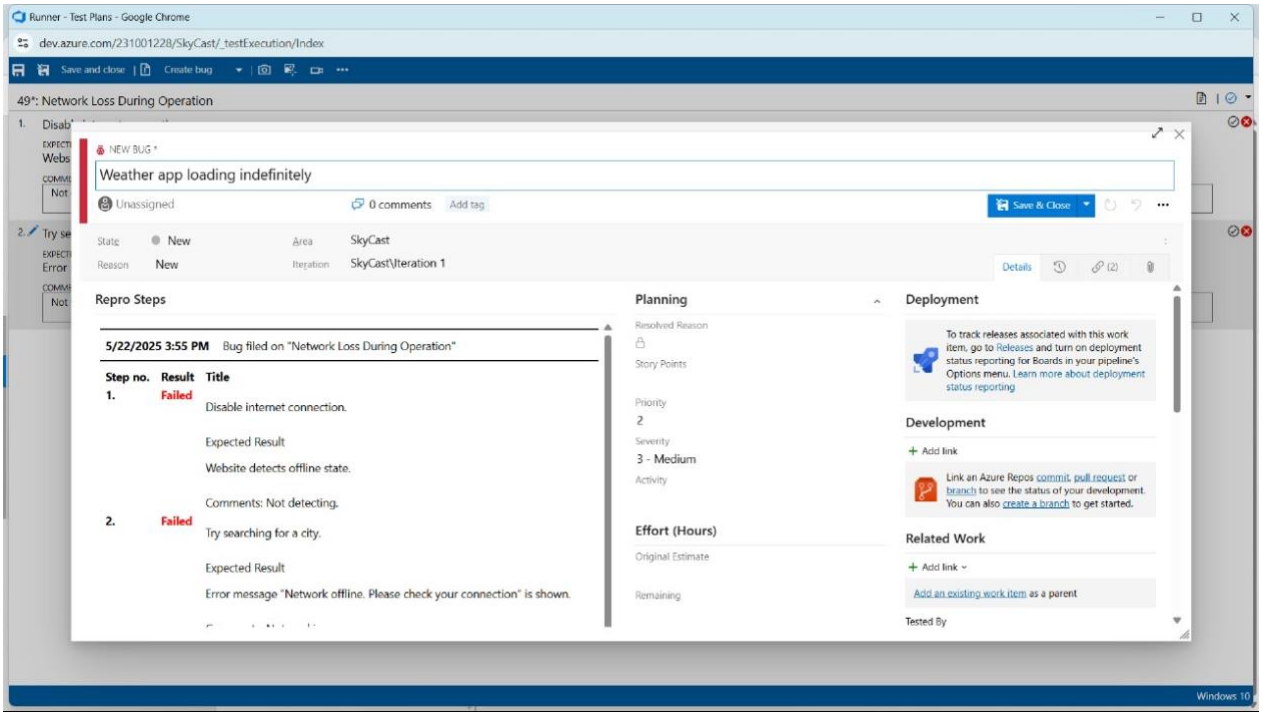


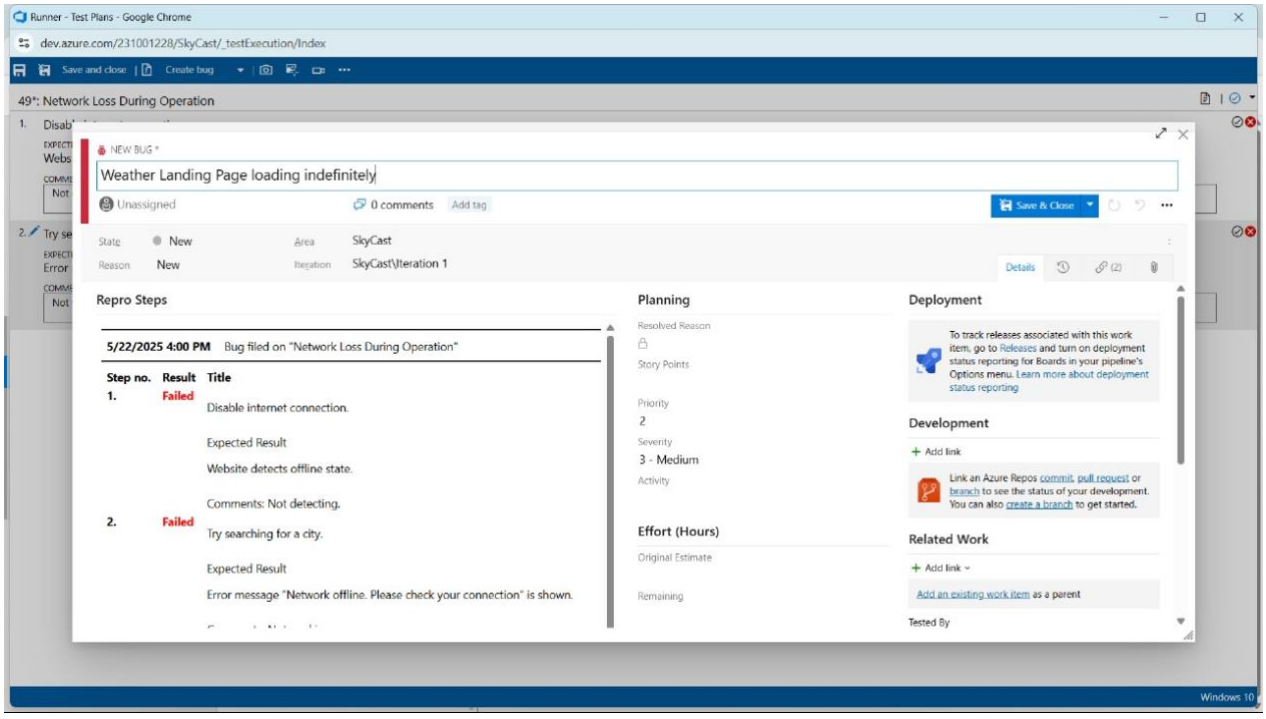
6. Recording the test case



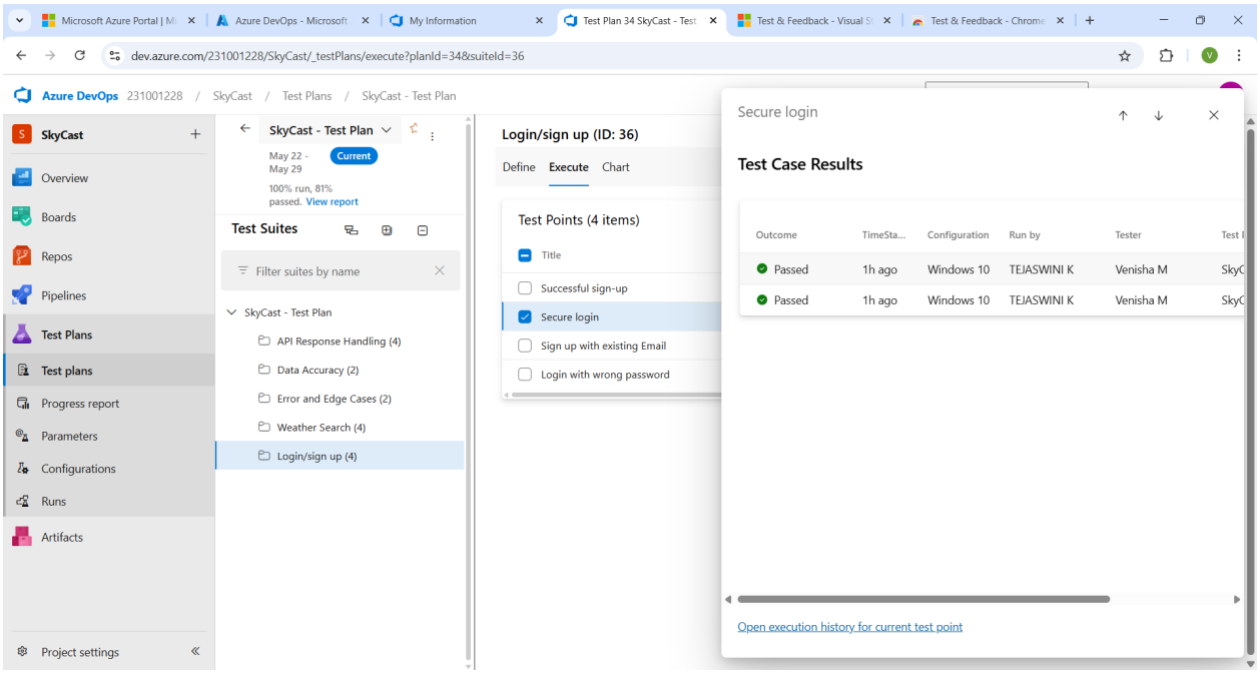
7. Creating the bug



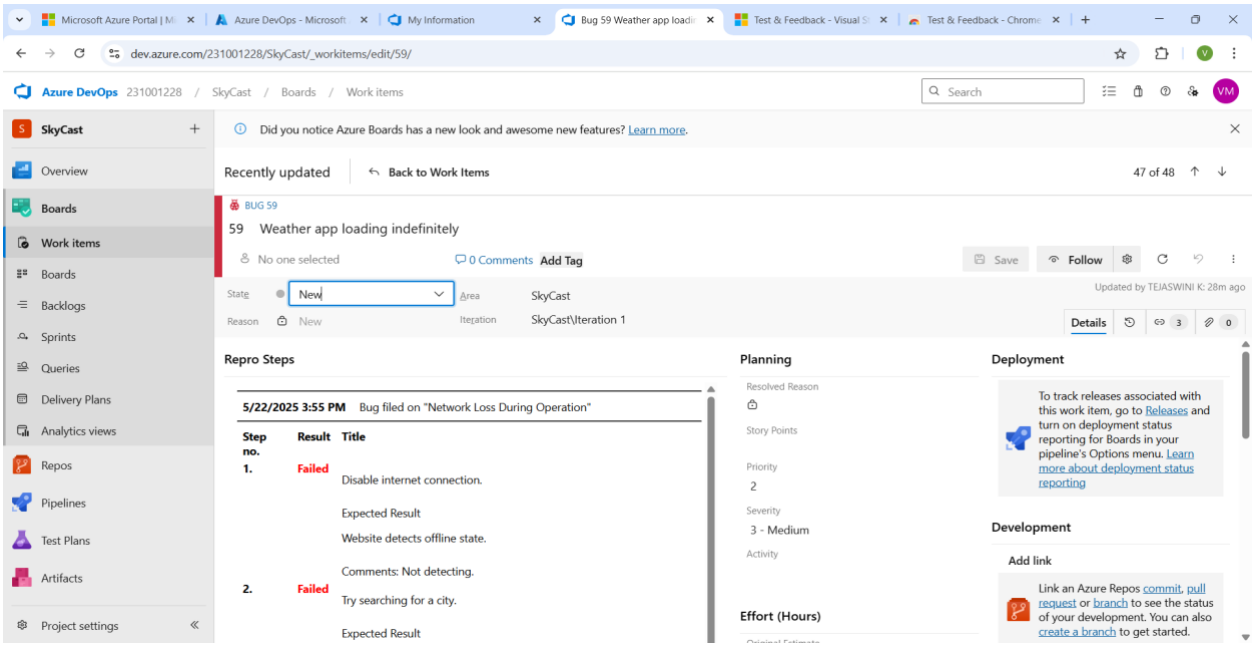

Runner - Test Plans - Google Chrome
dev.azure.com/231001228/SkyCast/_testExecution/Index
Save and close | Create bug
49*: Network Loss During Operation
NEW BUG *
Weather app loading indefinitely
Unassigned
0 comments
Add tag
Save & Close
State New
Reason New
Area SkyCast
Iteration SkyCast\Iteration 1
Details
Repro Steps
5/22/2025 3:55 PM Bug filed on "Network Loss During Operation"
Step no. Result Title
1. Failed
Disable internet connection.
Expected Result
Website detects offline state.
Comments: Not detecting.
2. Failed
Try searching for a city.
Expected Result
Error message "Network offline. Please check your connection" is shown.
Planning
Resolved Reason
Story Points
Priority
2
Severity
3 - Medium
Activity
Effort (Hours)
Original Estimate
Remaining
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
Tested By
Windows 10


Runner - Test Plans - Google Chrome
dev.azure.com/231001228/SkyCast/_testExecution/Index
Save and close | Create bug
49*: Network Loss During Operation
NEW BUG *
Weather Landing Page loading indefinitely
Unassigned
0 comments
Add tag
Save & Close
State New
Reason New
Area SkyCast
Iteration SkyCast\Iteration 1
Details
Repro Steps
5/22/2025 4:00 PM Bug filed on "Network Loss During Operation"
Step no. Result Title
1. Failed
Disable internet connection.
Expected Result
Website detects offline state.
Comments: Not detecting.
2. Failed
Try searching for a city.
Expected Result
Error message "Network offline. Please check your connection" is shown.
Planning
Resolved Reason
Story Points
Priority
2
Severity
3 - Medium
Activity
Effort (Hours)
Original Estimate
Remaining
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
Tested By
Windows 10

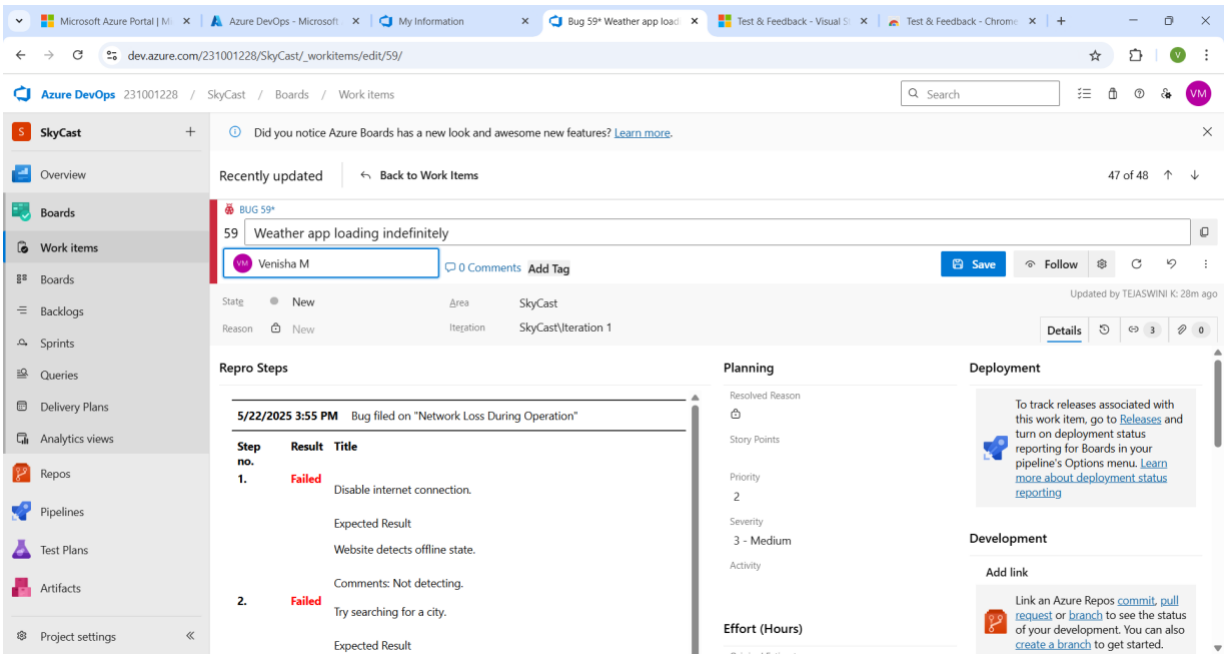
8. Test case results



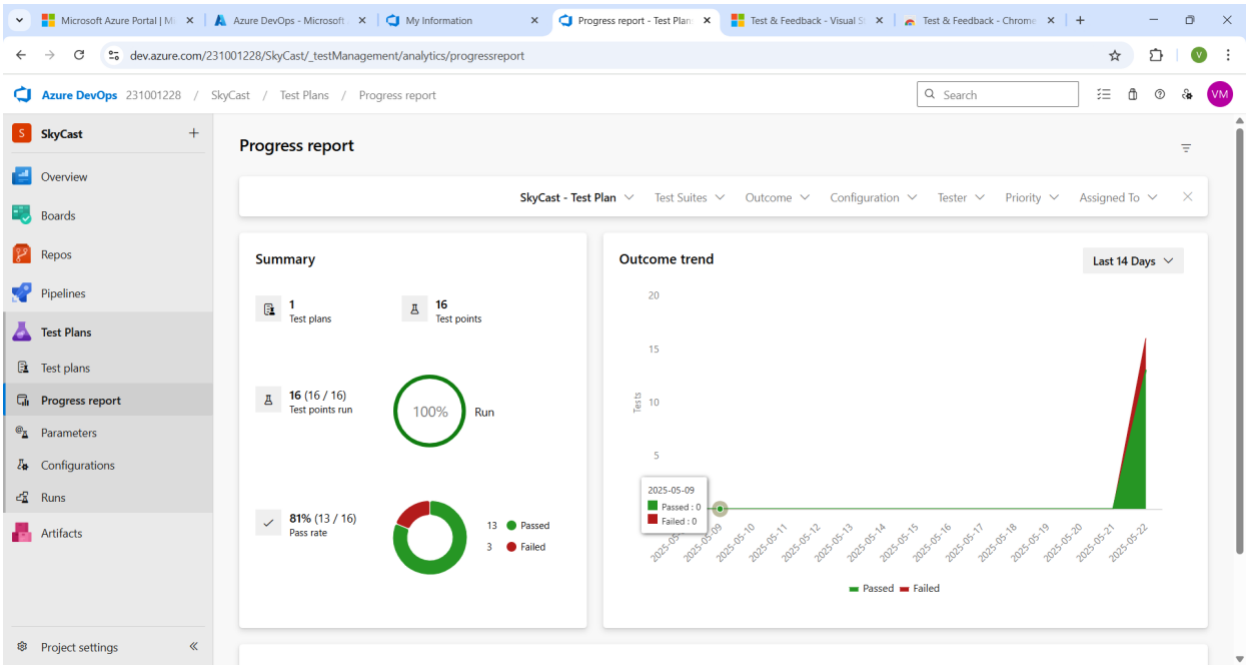
9. Test report summary



- Assigning bug to the developer and changing state



10. Progress report



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

Aim:

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

Load Testing

Steps to Create an Azure Load Testing Resource:

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

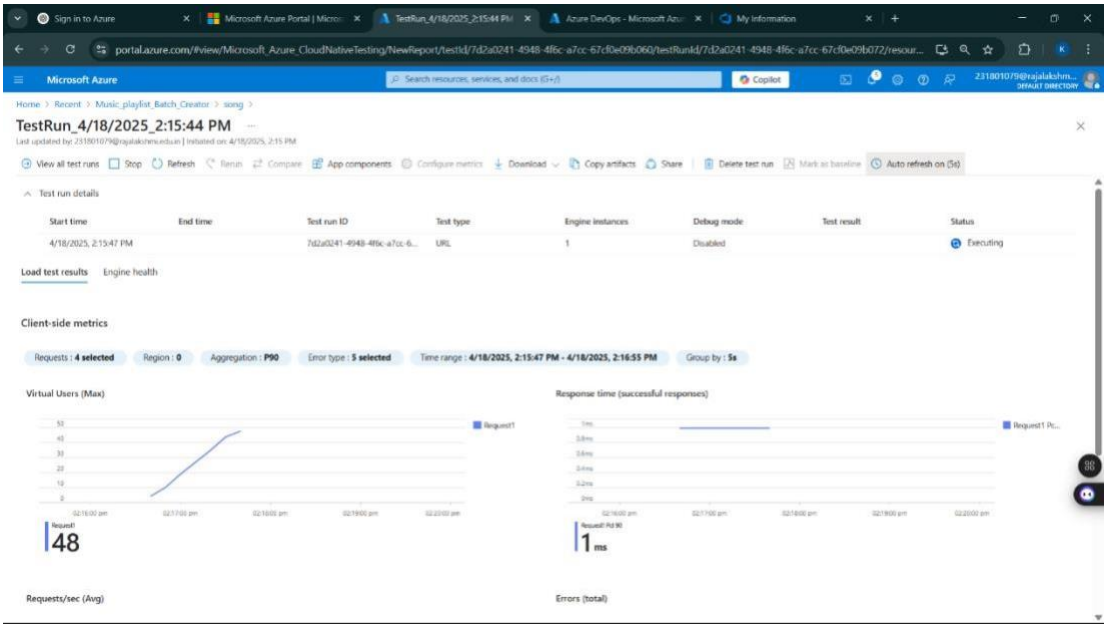
1. Sign in to Azure Portal
   Go to https://portal.azure.com and log in.
2. Create the Resource
   - Go to *Create a resource* → Search for "Azure Load Testing".
   - Select Azure Load Testing and click Create.
3. Fill in the Configuration Details
   - Subscription*:* Choose your Azure subscription.
   - *Resource Group:* Create new or select an existing one.
   - *Name:* Provide a unique name (no special characters).
   - *Location:* Choose the region for hosting the resource.
4. (Optional) Configure tags for categorization and billing.
5. Click Review + Create, then Create.
6. Once deployment is complete, click Go to resource.

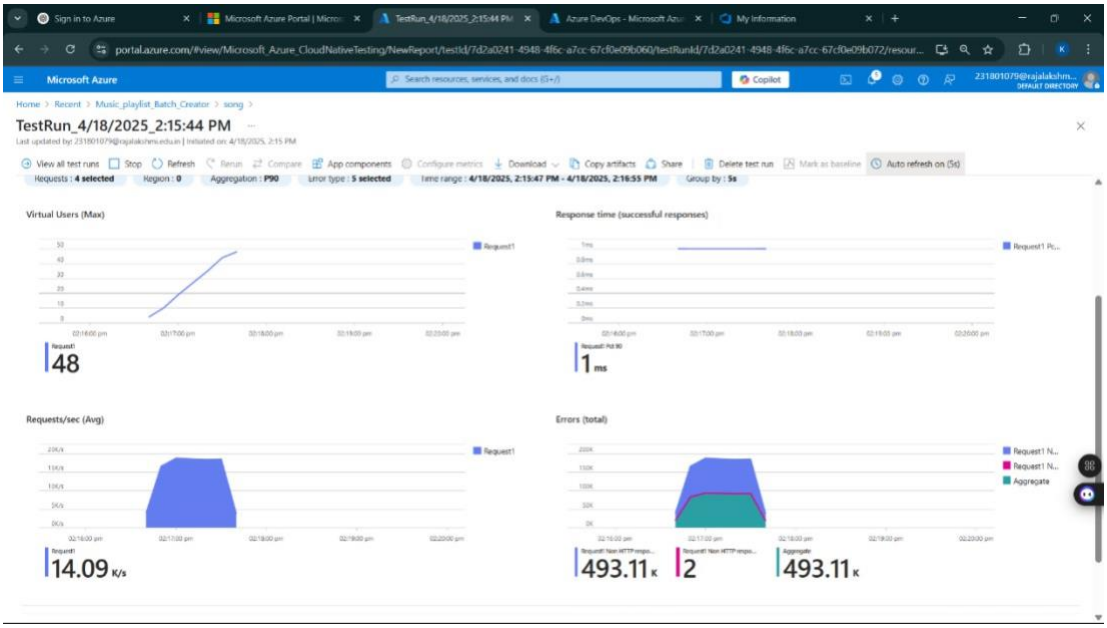
Steps to Create and Run a Load Test:

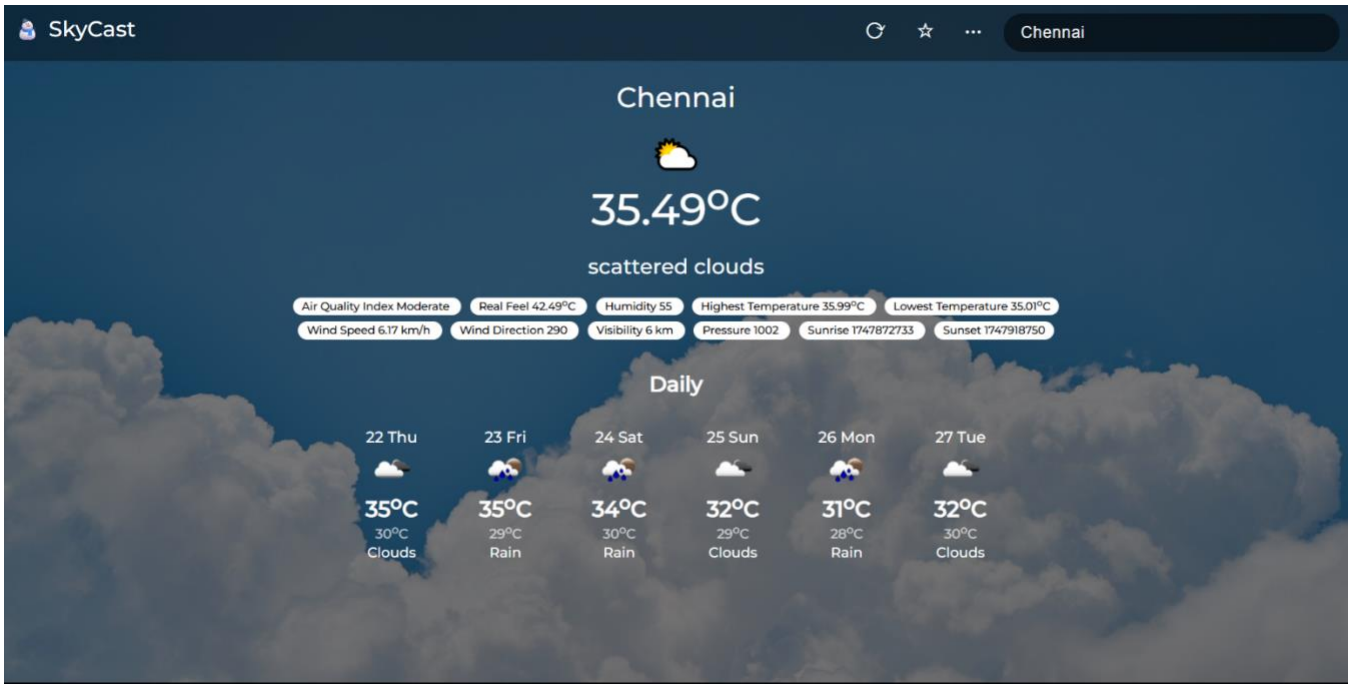
Once your resource is ready:

1. Go to your Azure Load Testing resource and click Add HTTP requests > Create.
2. Basics Tab
   - *Test Name:* Provide a unique name.
   - *Description:* (Optional) Add test purpose.
   - *Run After Creation:* Keep checked.
3. Load Settings
   - *Test URL:* Enter the target endpoint (e.g., https://yourapi.com/products).
4. Click Review + Create → Create to start the test.

Load Testing







Result:

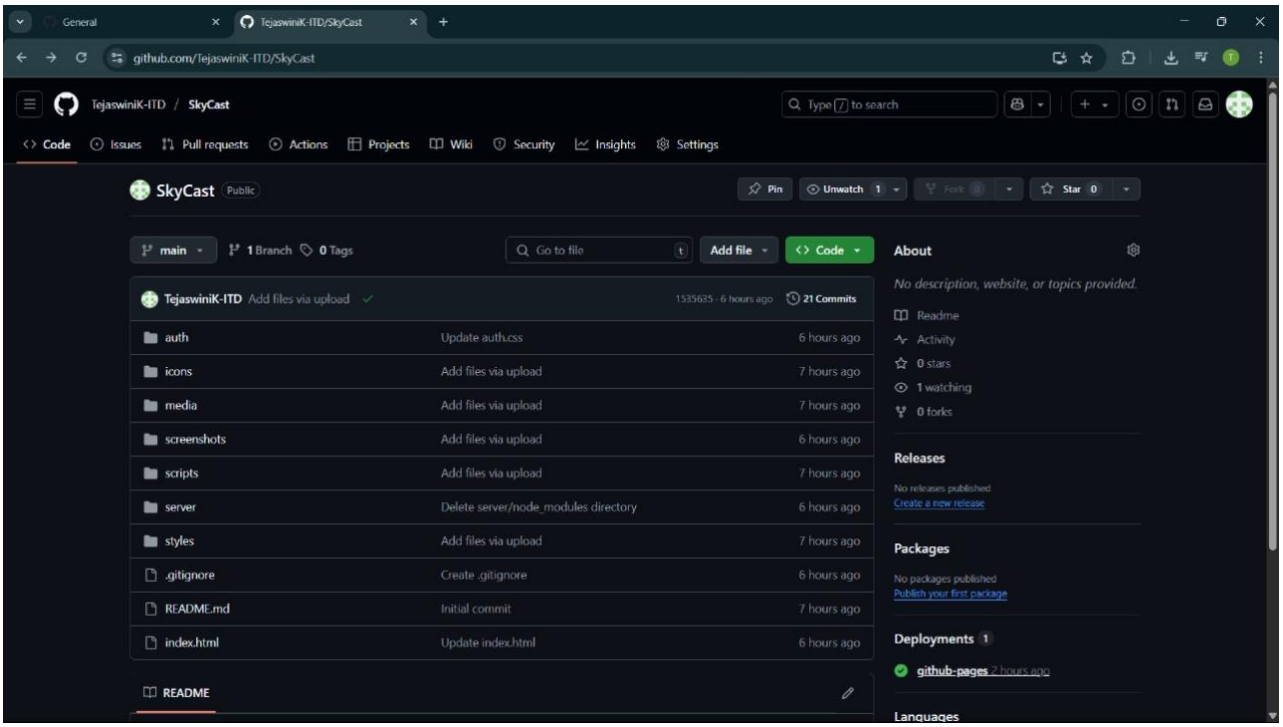
Successfully created the Azure Load Testing resource and executed a load test to assess the performance of the specified endpoint.

EXP NO: 10	GITHUB: PROJECT STRUCTURE & NAMING CONVENTIONS

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



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.