EXP NO: 1	AZURE DEVOPS ENVIRONMENT SETUP

Aim:

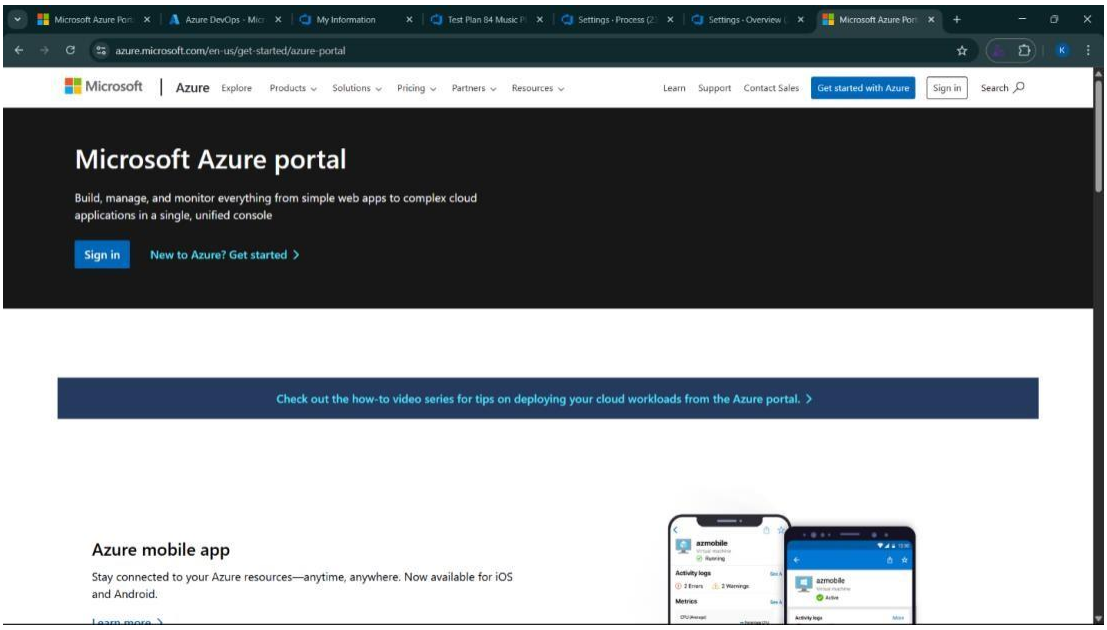
To set up and access the Azure DevOps environment by creating an organization through the Azure portal.

INSTALLATION

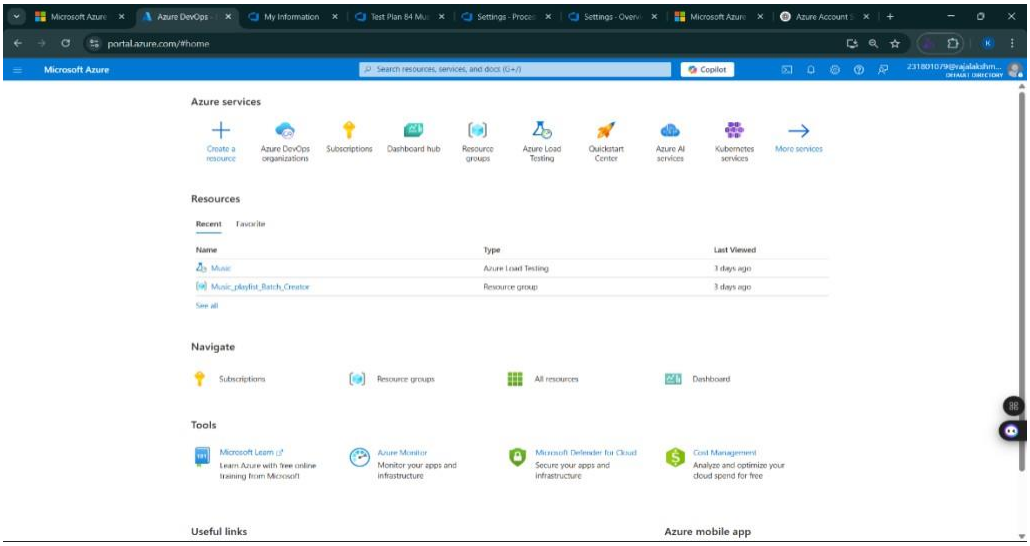
1. Open your web browser and go to the Azure website: https://azure.microsoft.com/en-us/get-started/azure-portal.

Sign in using your Microsoft account credentials.

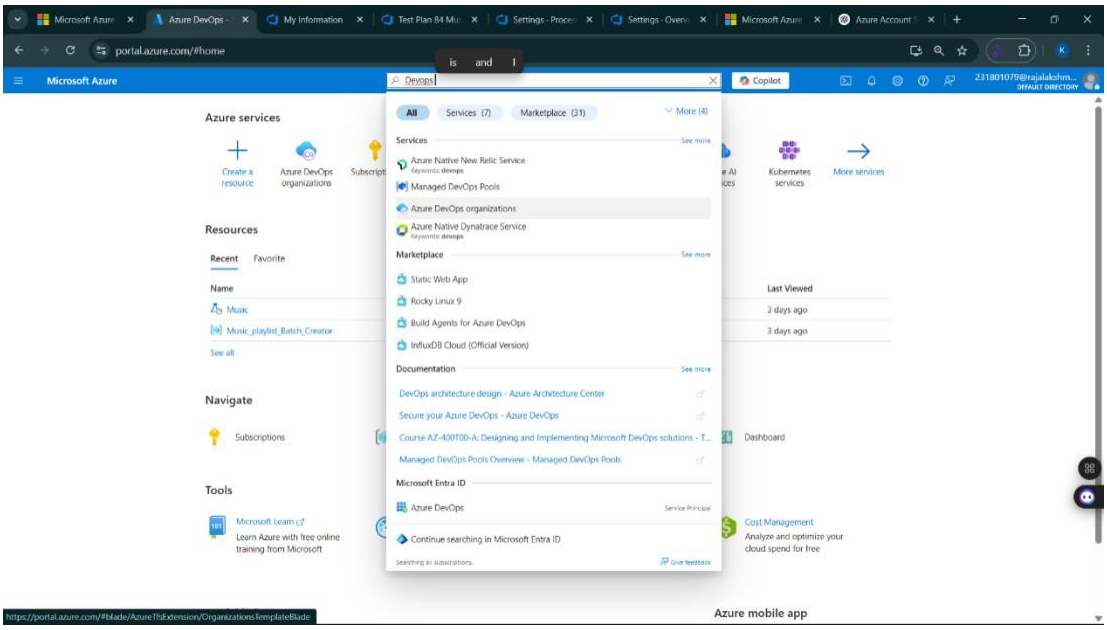
If you don't have a Microsoft account, you can create one here: https://signup.live.com/?lic=1



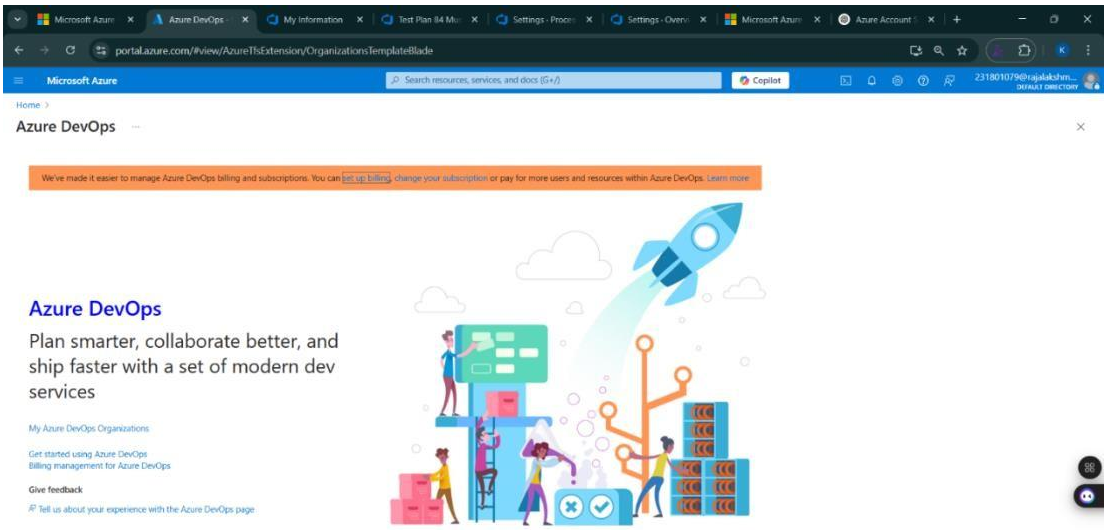
2. Azure home page



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:

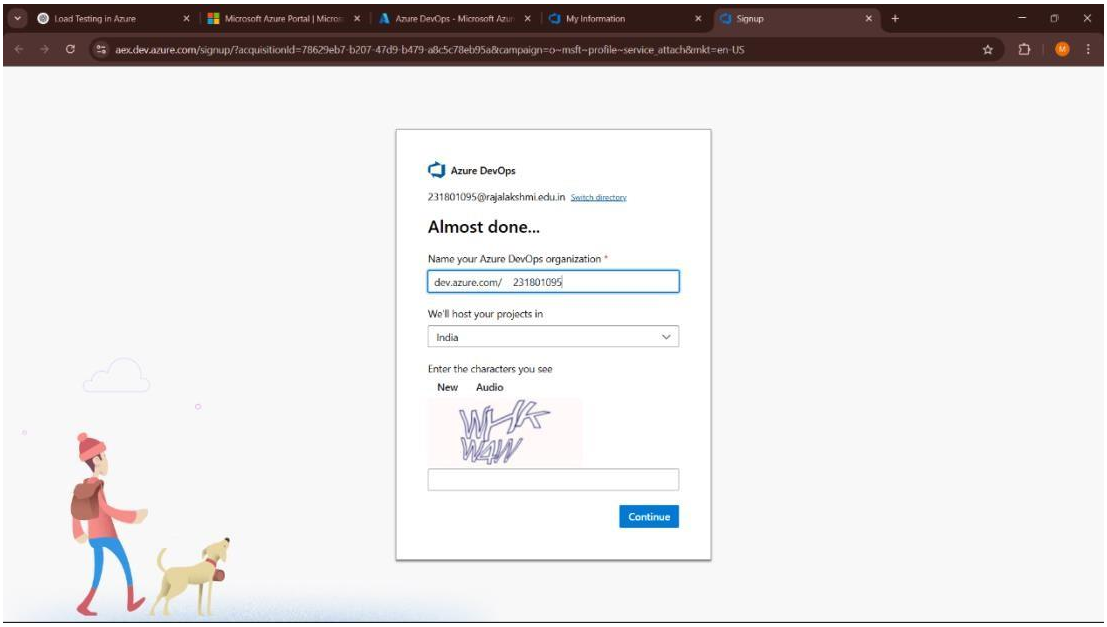
Successfully accessed the Azure DevOps environment and created a new organization through the Azure portal.

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.

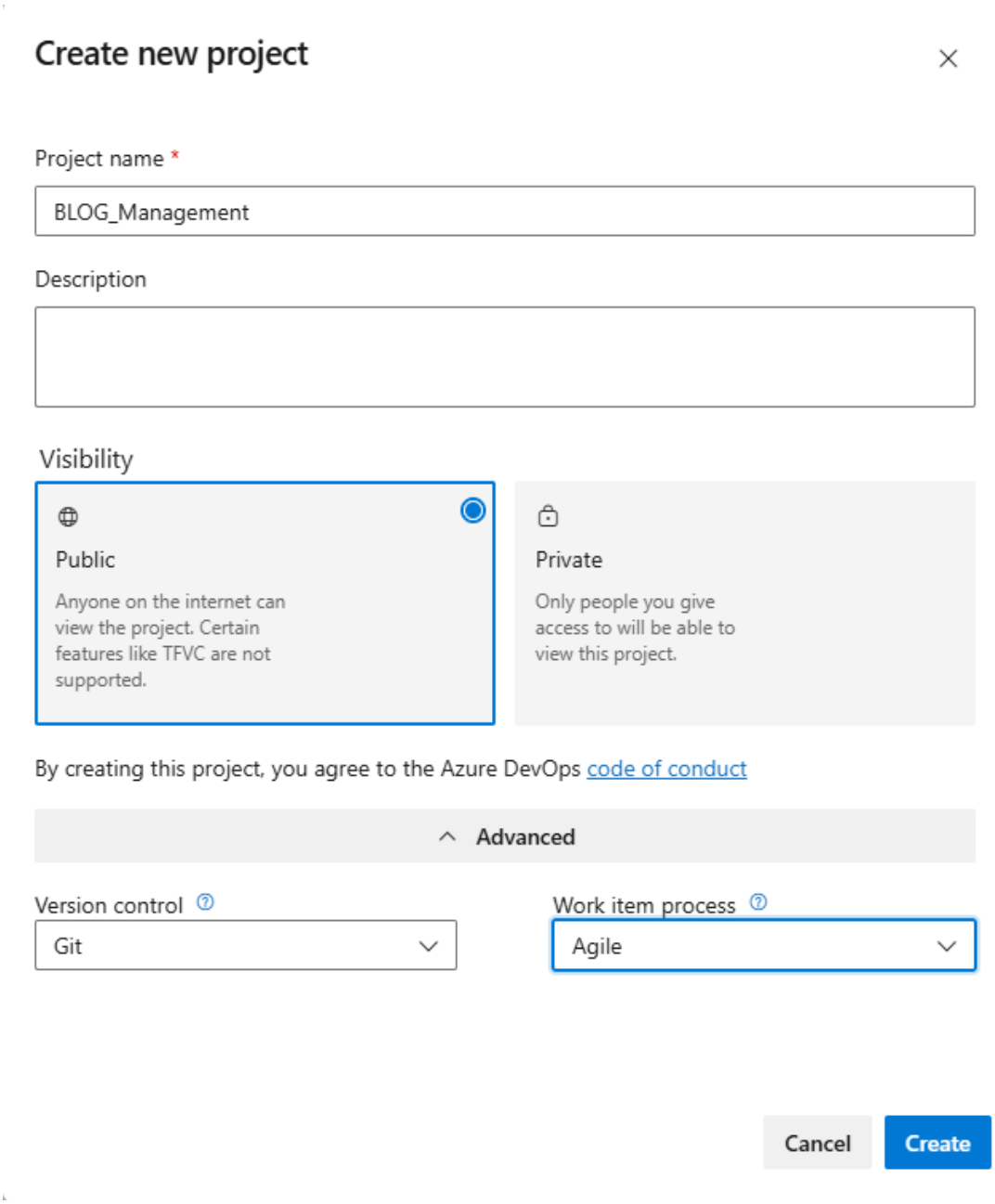
c. Enter the project name, description, and visibility options:

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

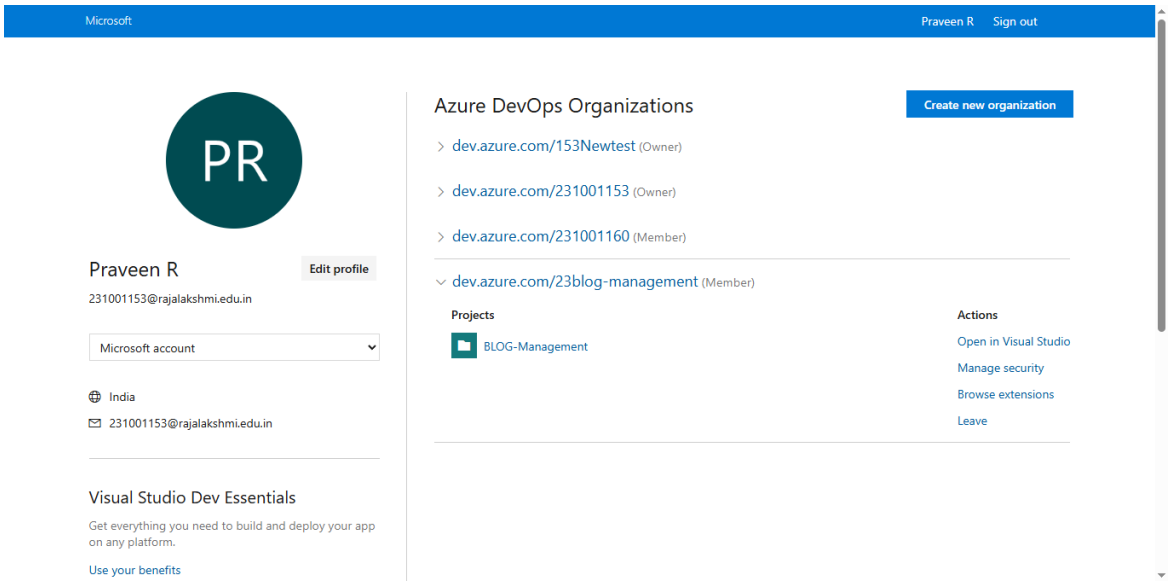
***Description**:* Optionally, add a description to provide more context about the project.

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

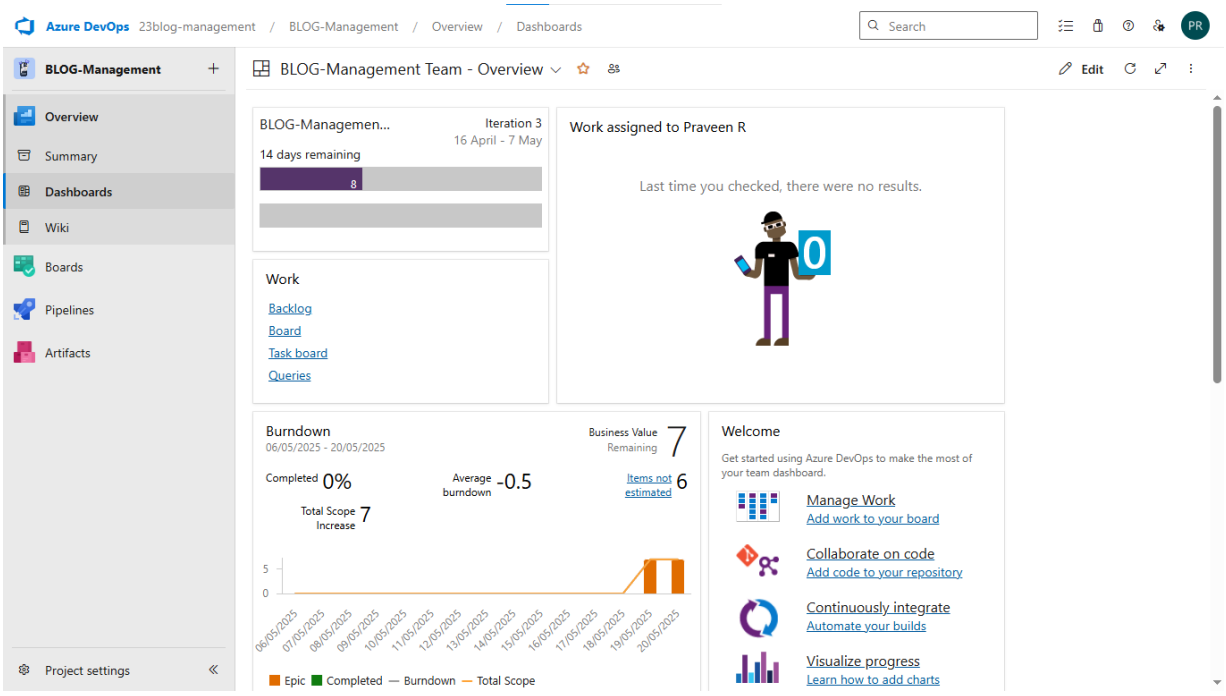
d. Once you've filled out the details, click **Create** to set up your first project.



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.



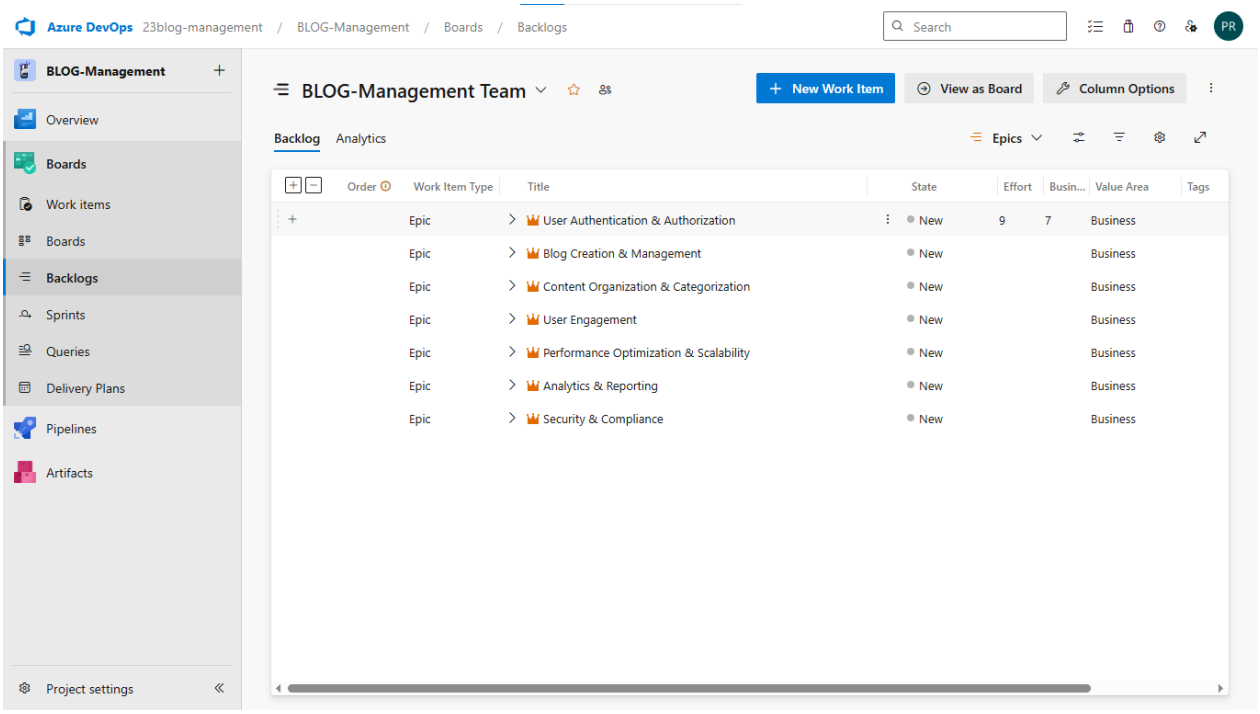
4. Project dashboard

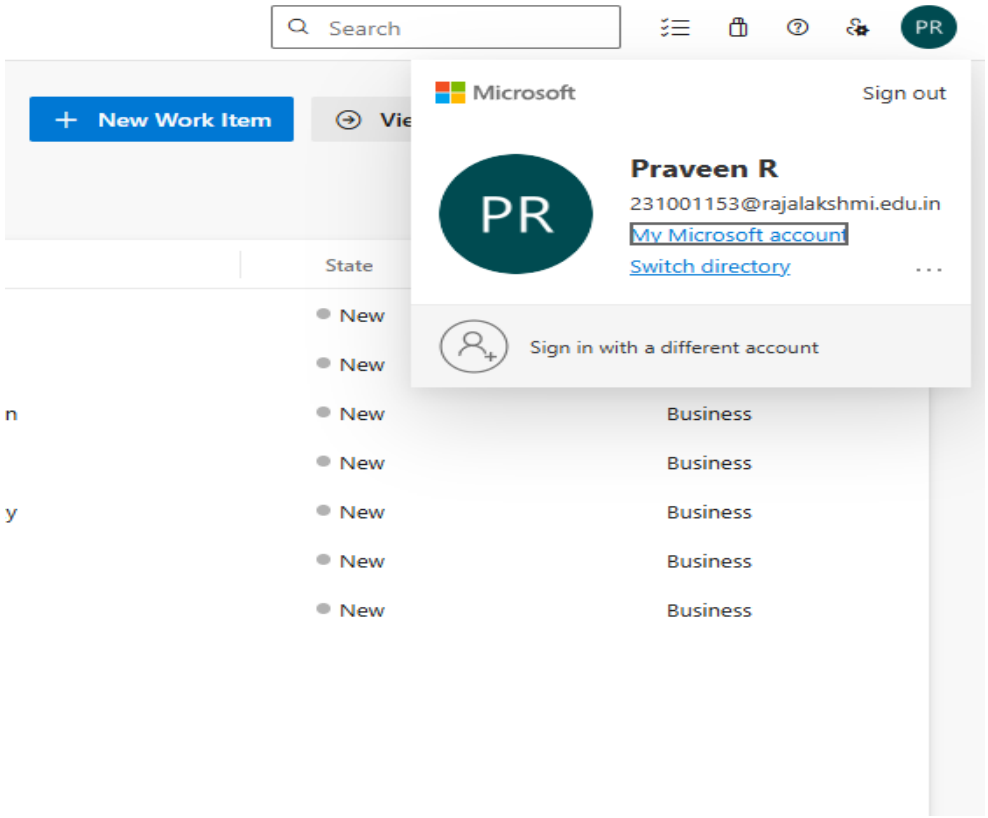


5. To manage user stories:

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:

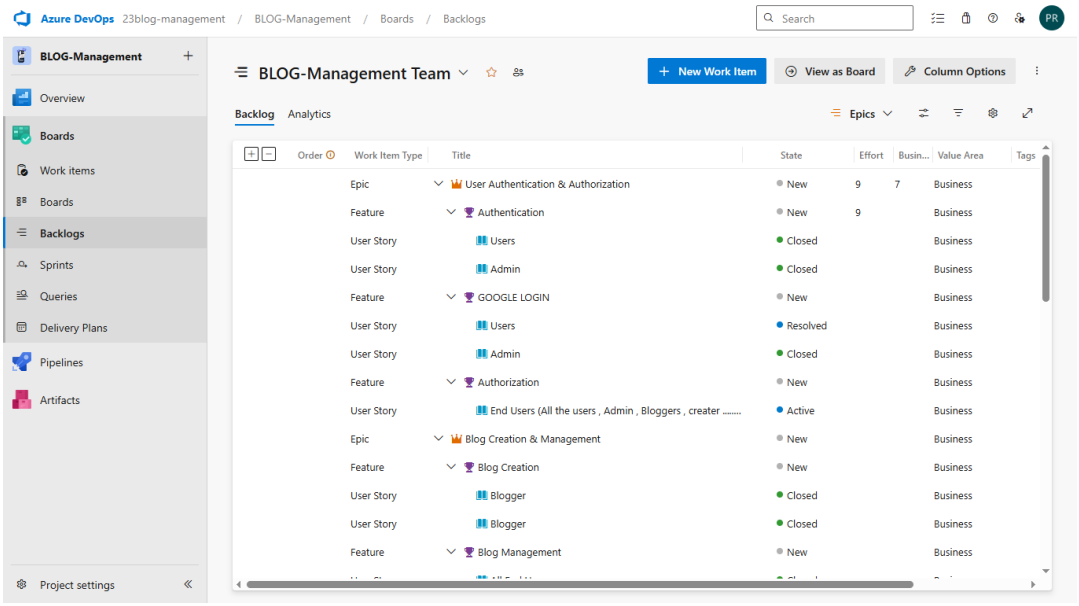
Successfully created an Azure DevOps project with user story management and agile workflow setup.

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

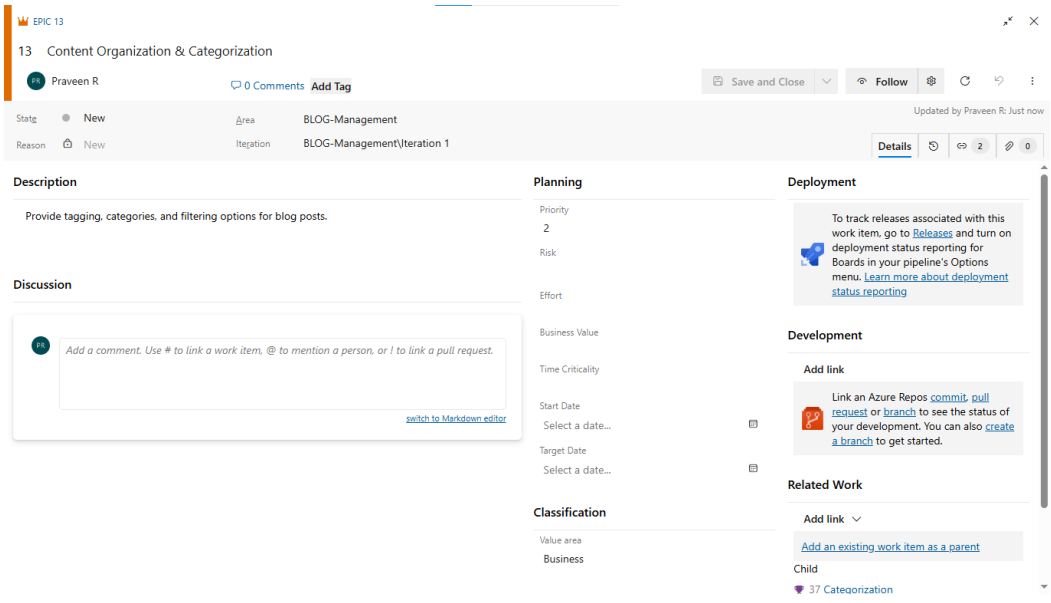
Aim:

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

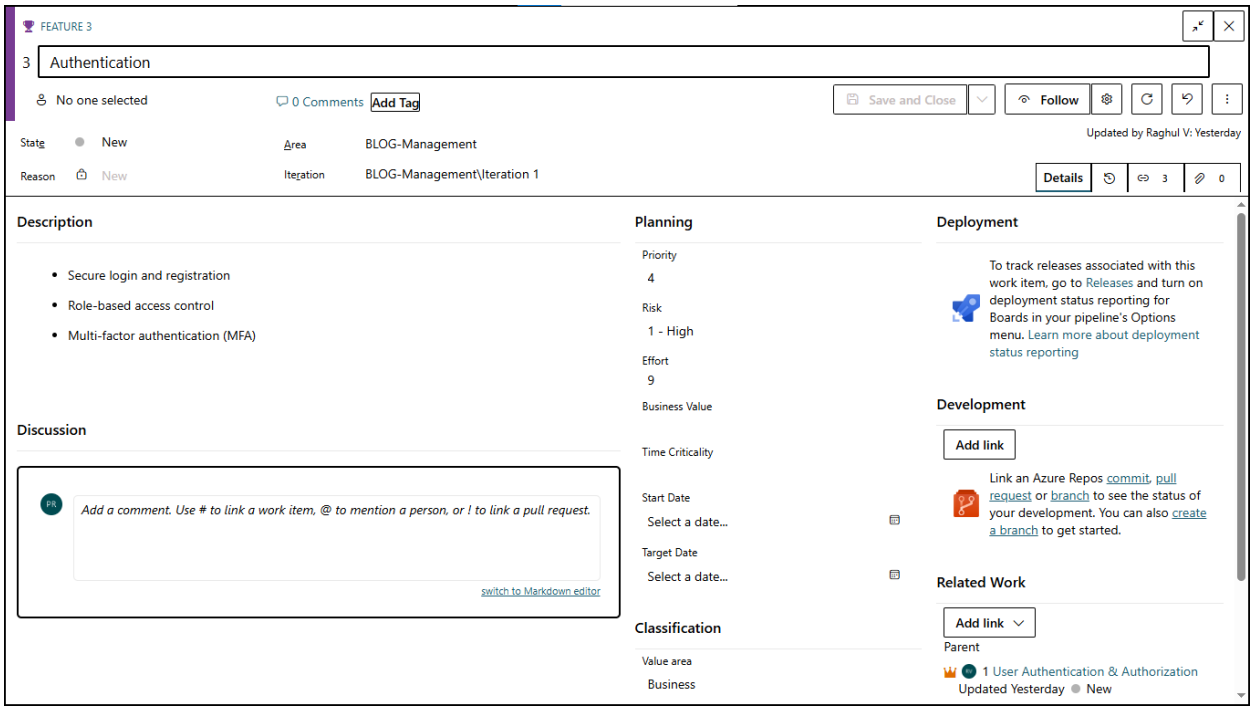
Create Epic, Features, User Stories, Task



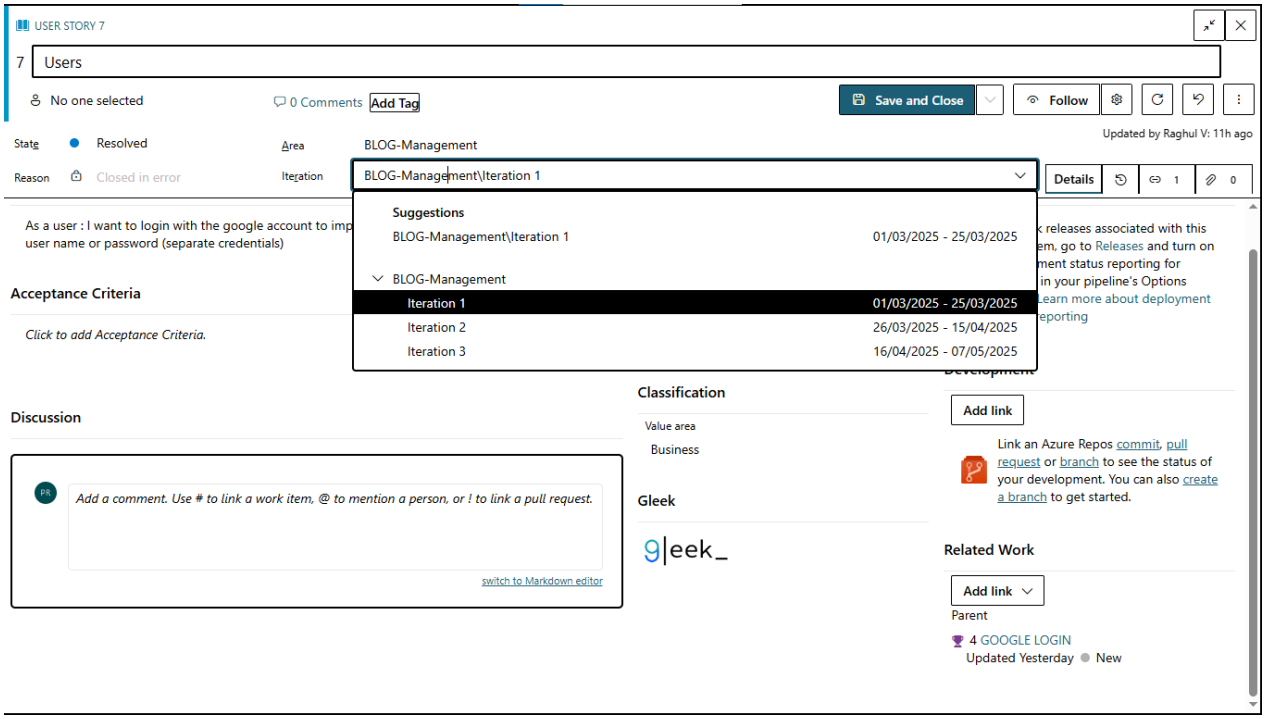
1. Fill in Epics



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.

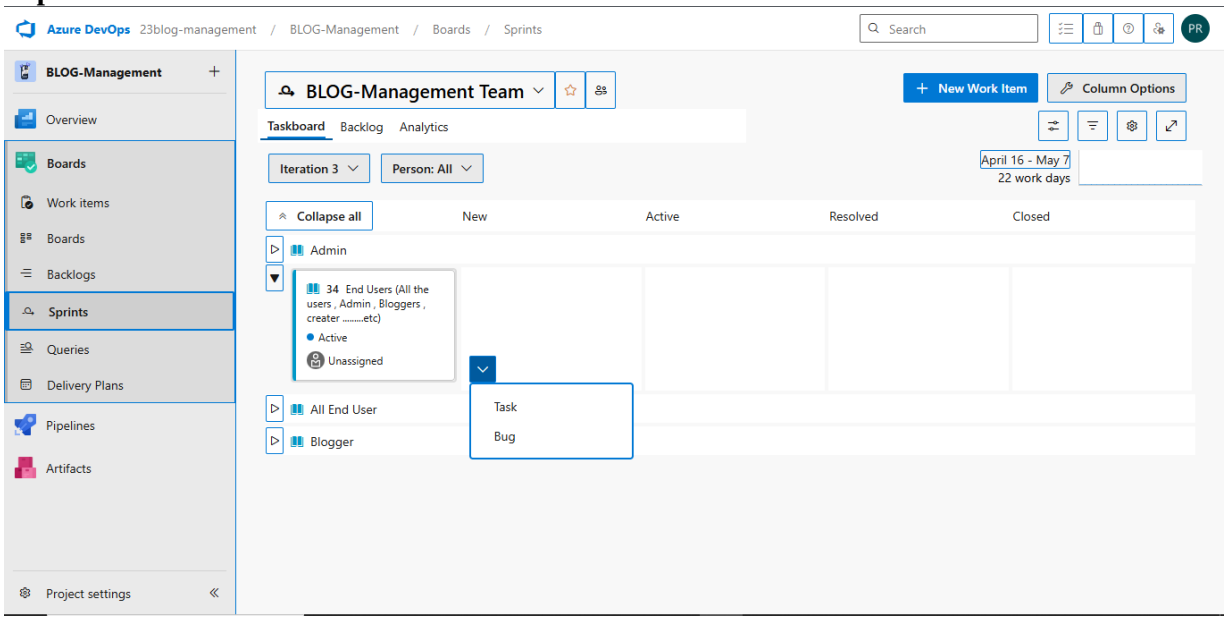
EXP NO: 4	SPRINT PLANNING

Aim:

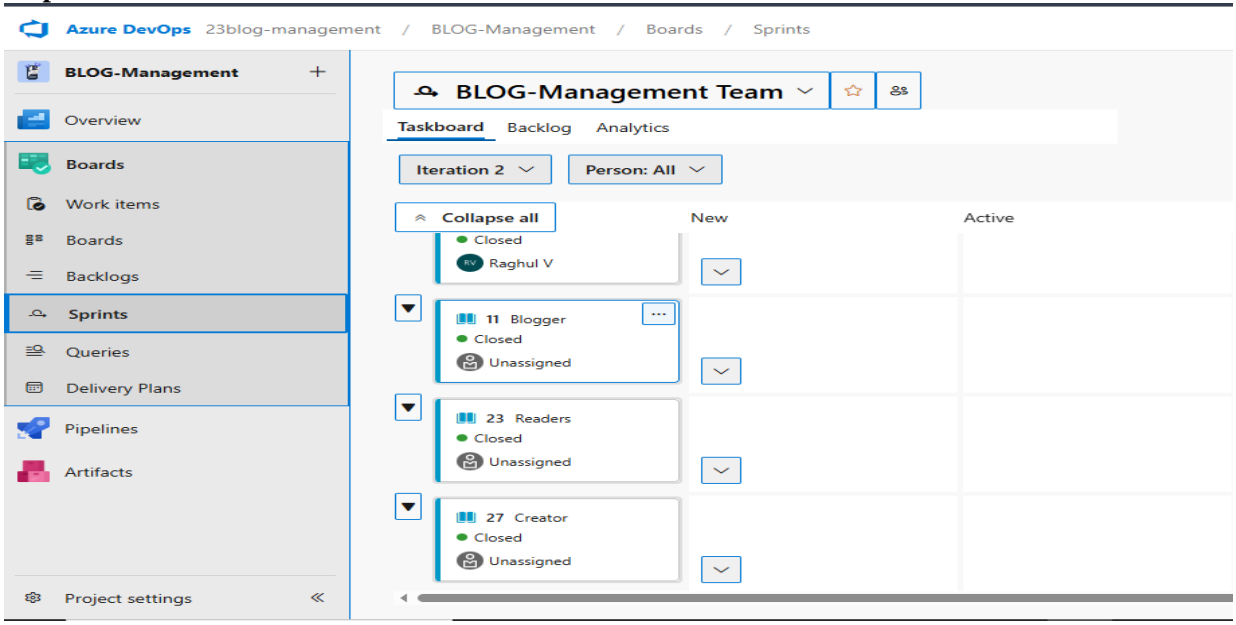
To assign user story to specific sprint for the BLOG-management team Creator Project.

Sprint Planning

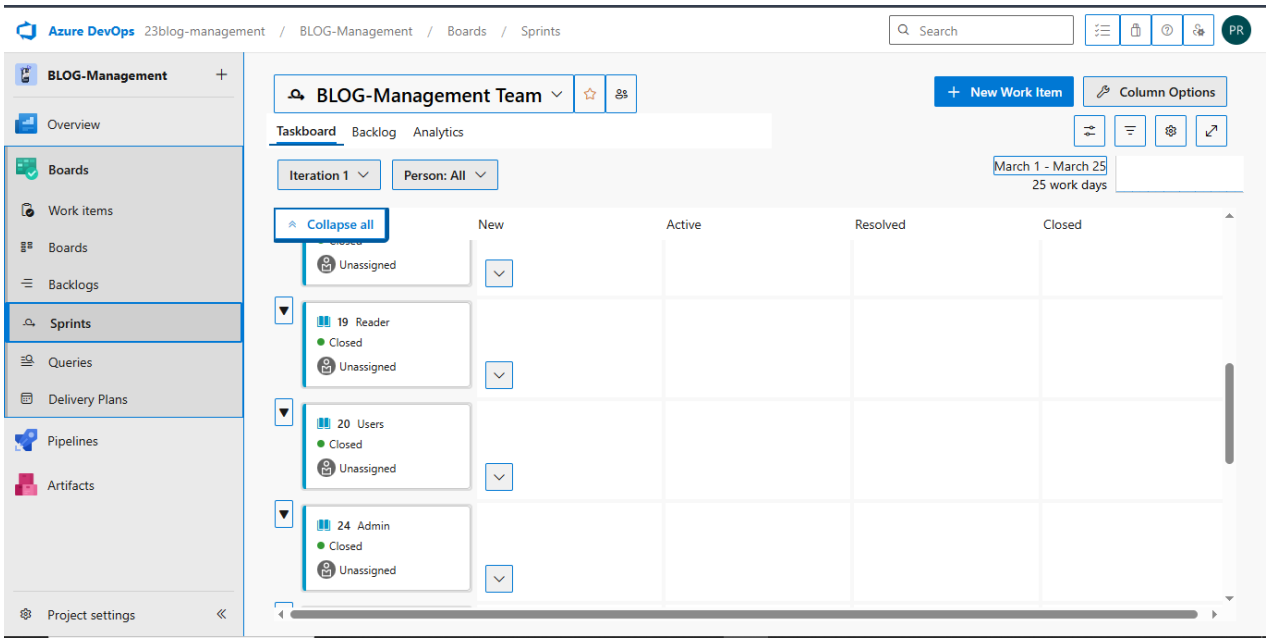
Sprint 1



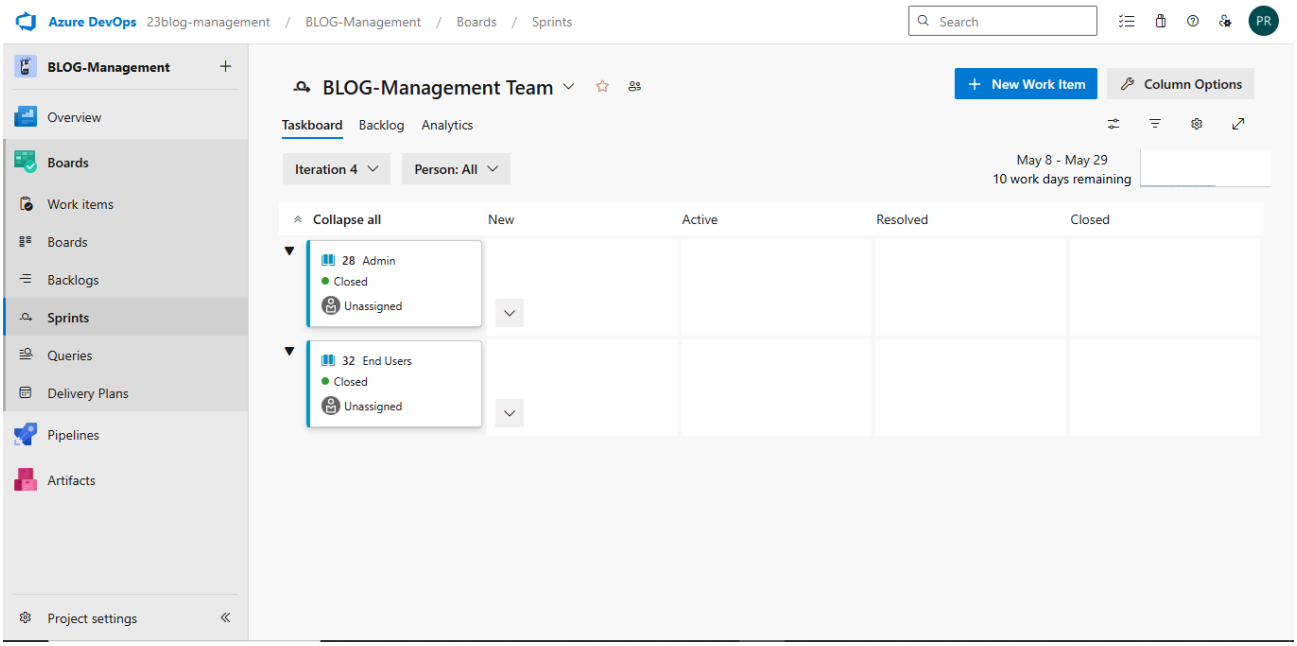
Sprint 2



Sprint 3



Sprint 4



Result:

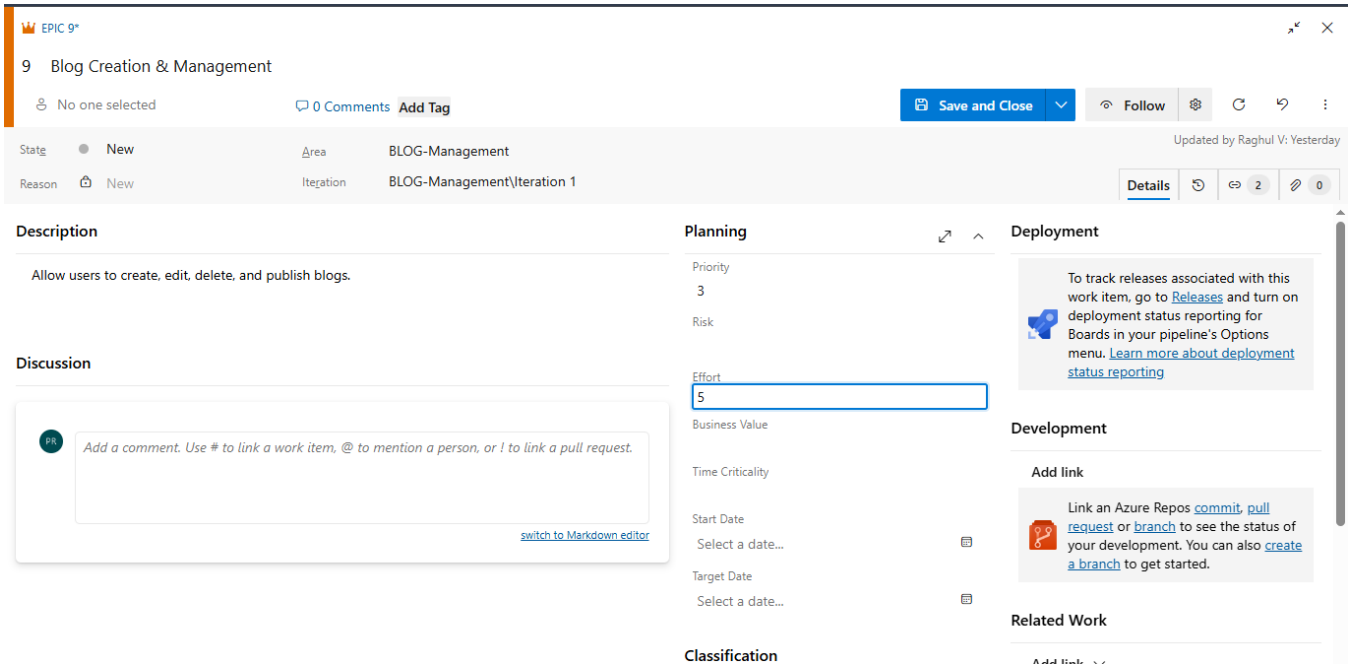
The Sprints are created for the BLOG-management Team creator Project.

EXP NO: 5	POKER ESTIMATION

Aim:

Create Poker Estimation for the user stories – BLOG-management System Project.

Poker Estimation



Result:

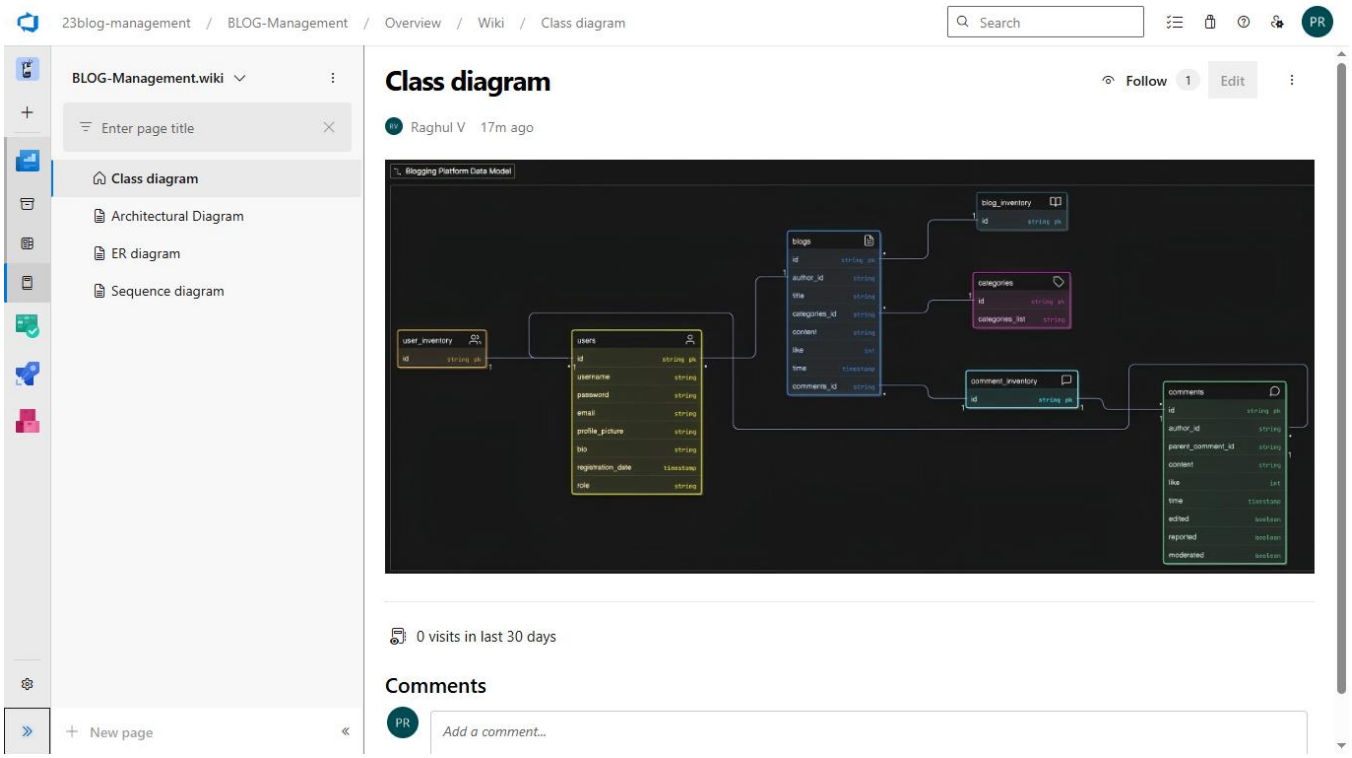
The Estimation/Story Points is created for the project using Poker Estimation.

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

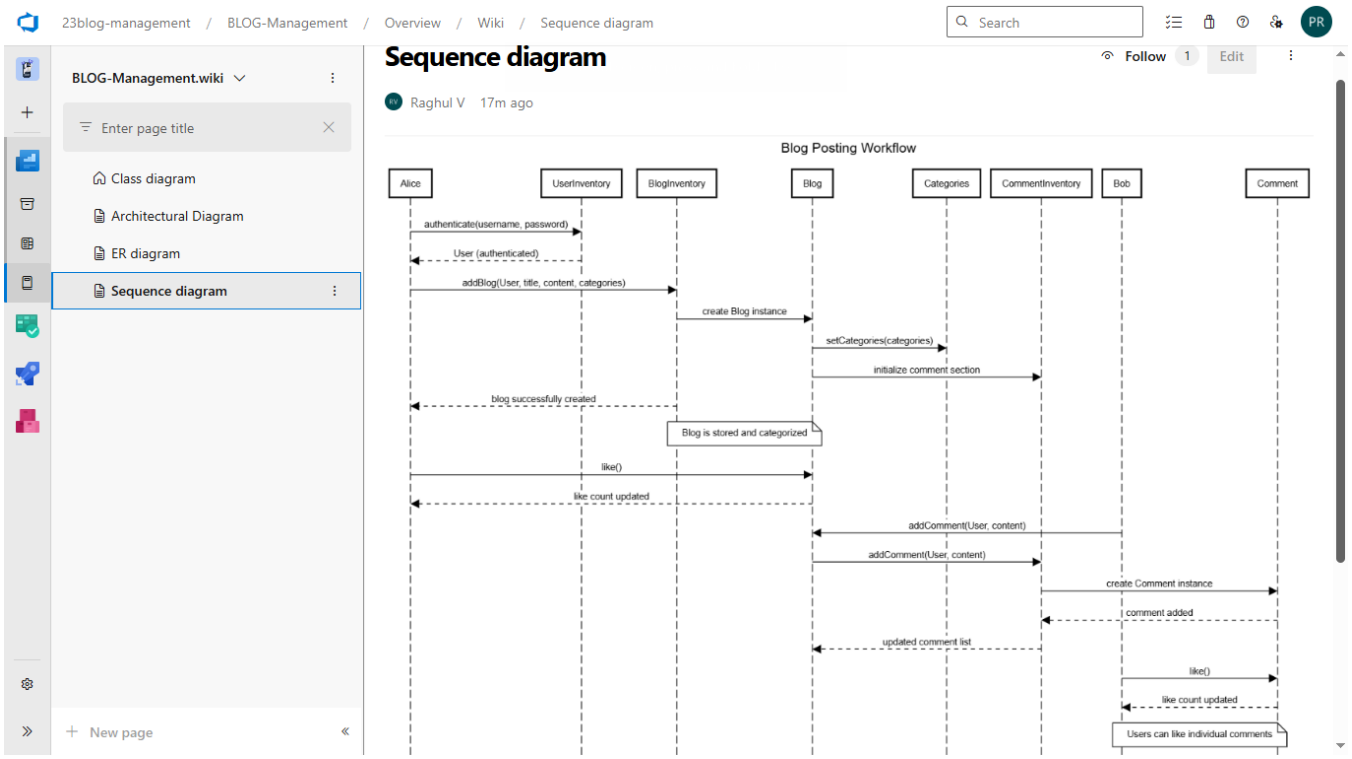
Aim:

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

6A. Class Diagram



6B. Sequence Diagram



Result:

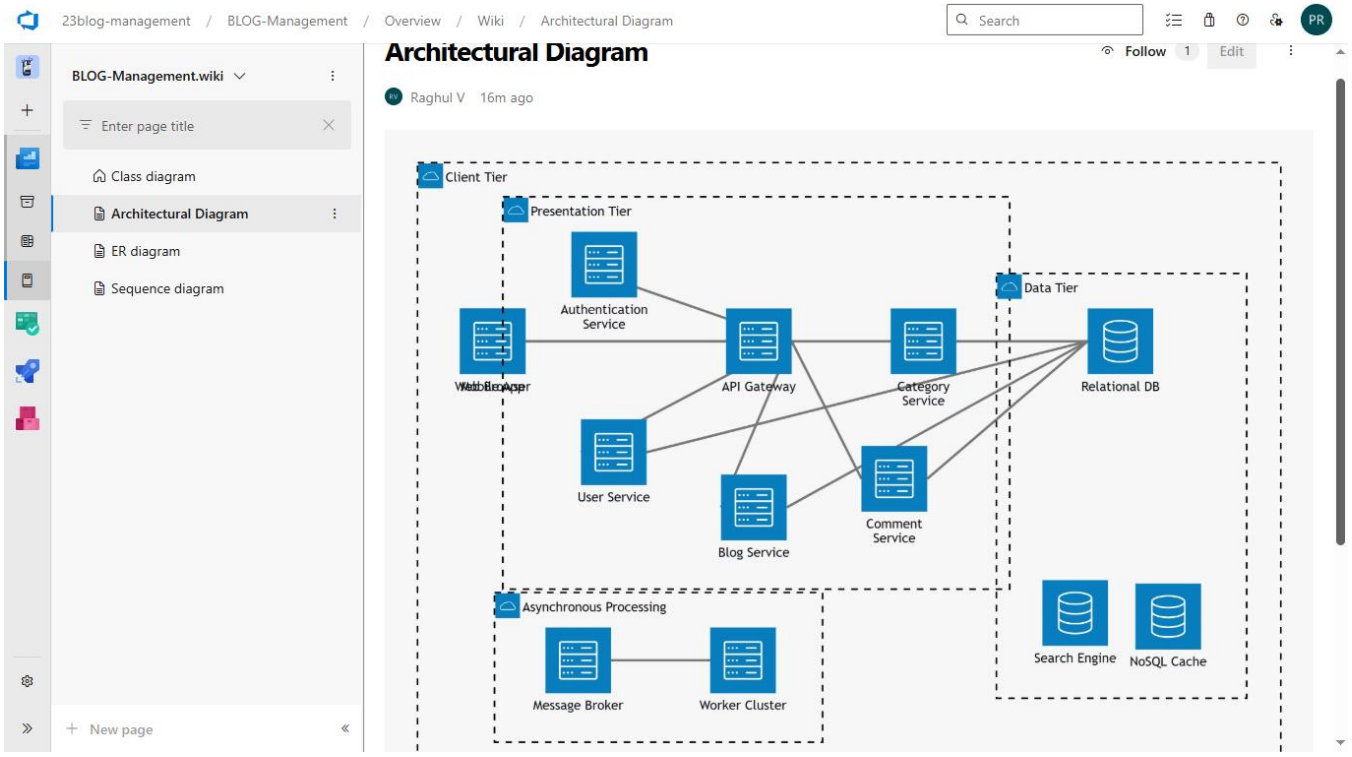
The Class Diagram and Sequence Diagram is designed Successfully for the BLOG management System.

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

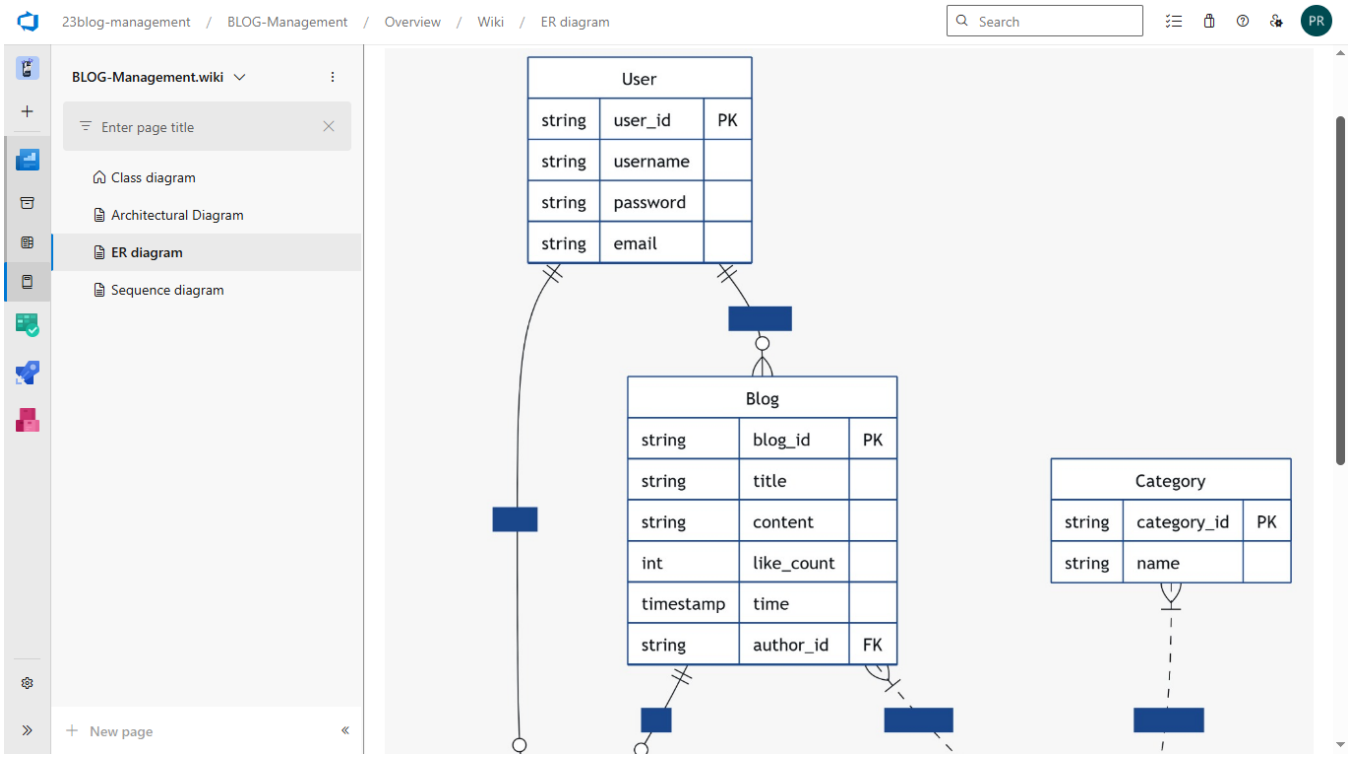
Aim:

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 BLOG -management System.

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

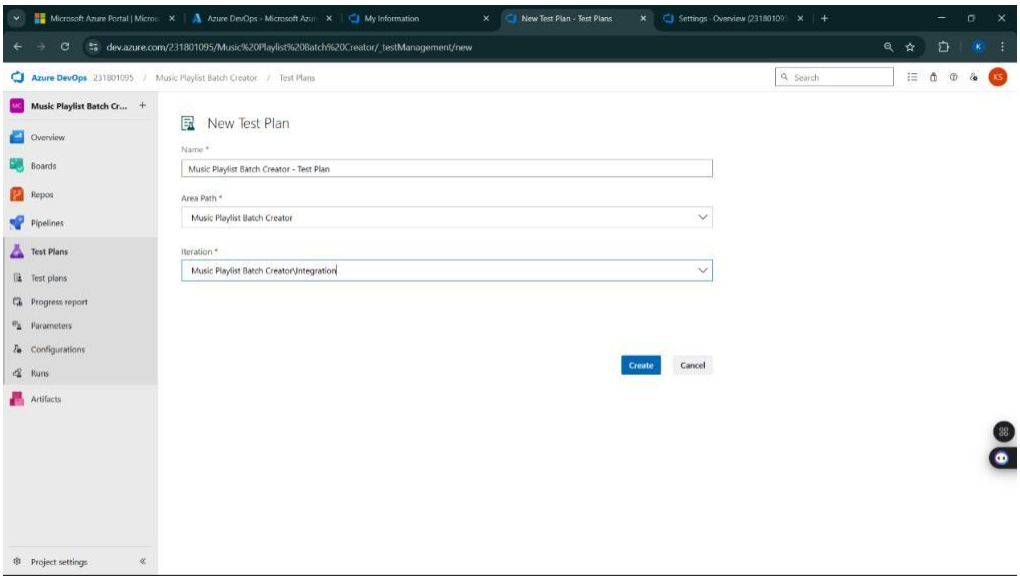
1. **Understand Core Features of the Application**
    - User Signup & Login
    - Viewing and Managing Playlists
    - Fetching Real-time Metadata
    - Editing the Blog (Title, Content, Tags)
    - Creating Smart Blog Filters Based on Categories
2. **Define User Interactions**
    - Each test case simulates a real user behavior.
3. **Design Happy Path Test Cases**
    - Focused on validating that all features function as expected under normal conditions
    - Example: user logs in, creates a blog post, or filters blogs by category.
4. **Design Error Path Test Cases**
    - Simulate negative or unexpected scenarios to test robustness and error handling.
    - Example: login fails with invalid credentials, blog creation fails with empty fields, no blogs found for selected category.
5. **Break Down Steps and Expected Results**
    - Each test case includes clear steps and expected results to ensure accuracy for testers and automation in blog management.
6. **Use Clear Naming and IDs**
    - Test cases are named clearly.
    - TC01 – Successful Login, TC05 – Blog Save Fails) to allow quick identification and linking to blog-related user stories or features.
7. **Separate Test Suites**
    - Test cases are grouped by functionality (e.g., Login, Blog Editing, Category Filtering) for better organization in blog management.

- Improves organization and test execution flow in Azure DevOps.

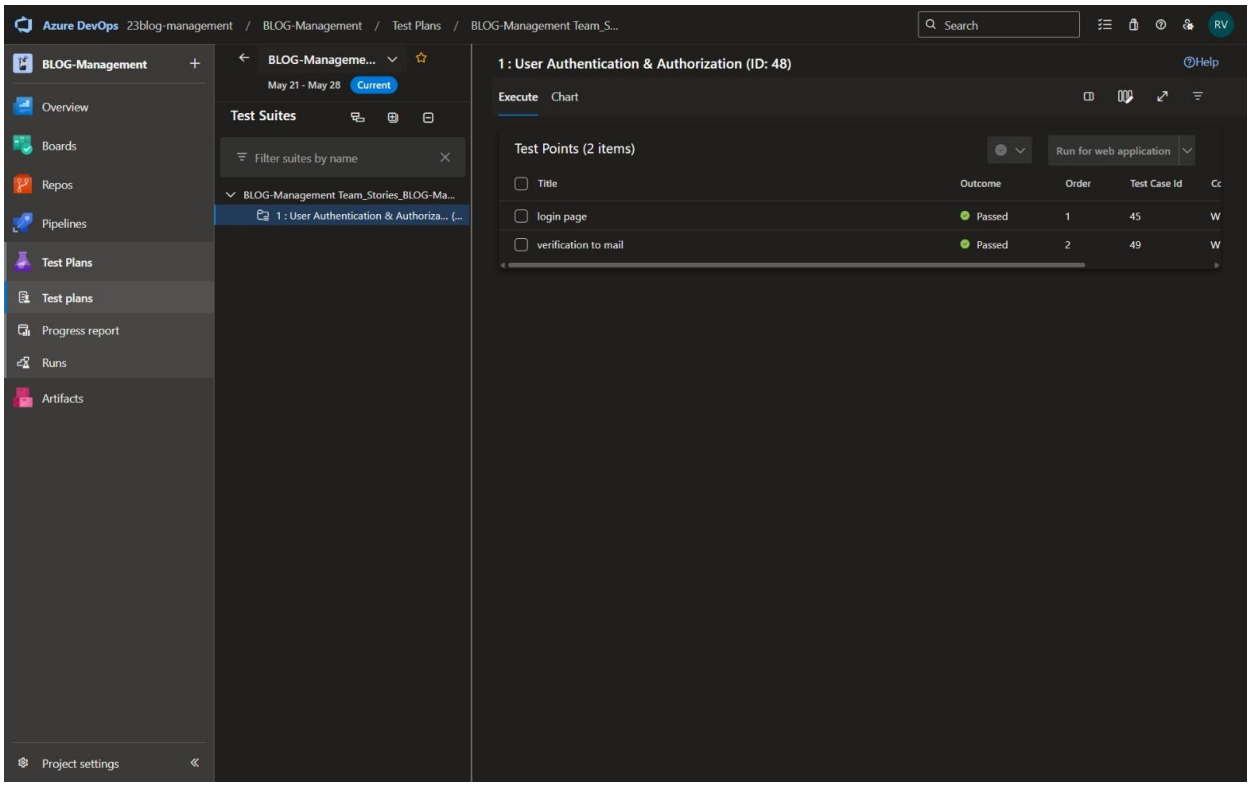
8. Prioritize and Review

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

1.New test plan



2.Test suite



3. Test case

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

USER STORIES

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

Test Suites

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

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

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

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

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

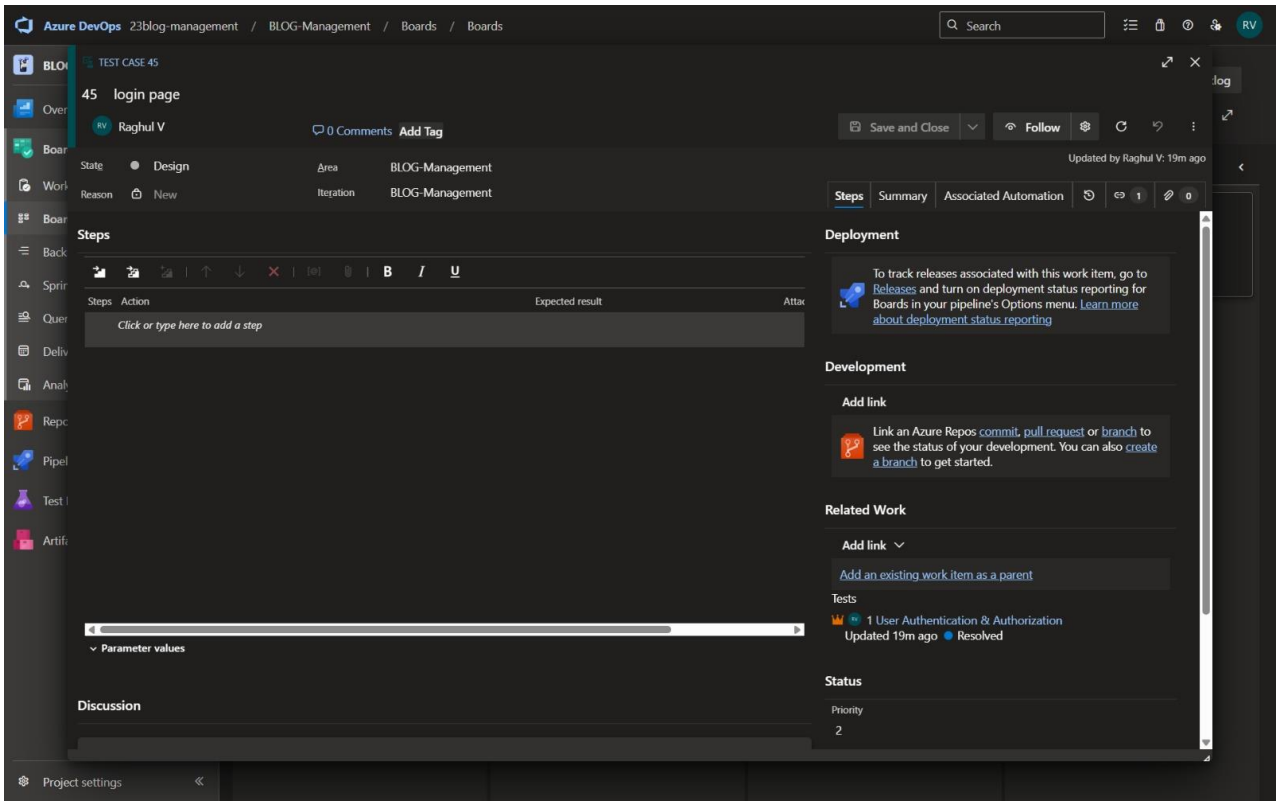
1. **TC05 – View Playlist Page**
    - **Action:**
        - Log in successfully.
        - Navigate to "Home page of Blog" section.
    - **Expected Results:**
        - All created playlists are displayed clearly.
    - **Type:** Happy Path
2. TC06 – Home page Loading Failure
    - **Action:**
        - Disconnect from the internet.
        - Navigate to “Home page”.
    - **Expected Results:**
        - Network is offline.
        - Error message "Unable to load playlists" is shown.
    - **Type:** Error Path

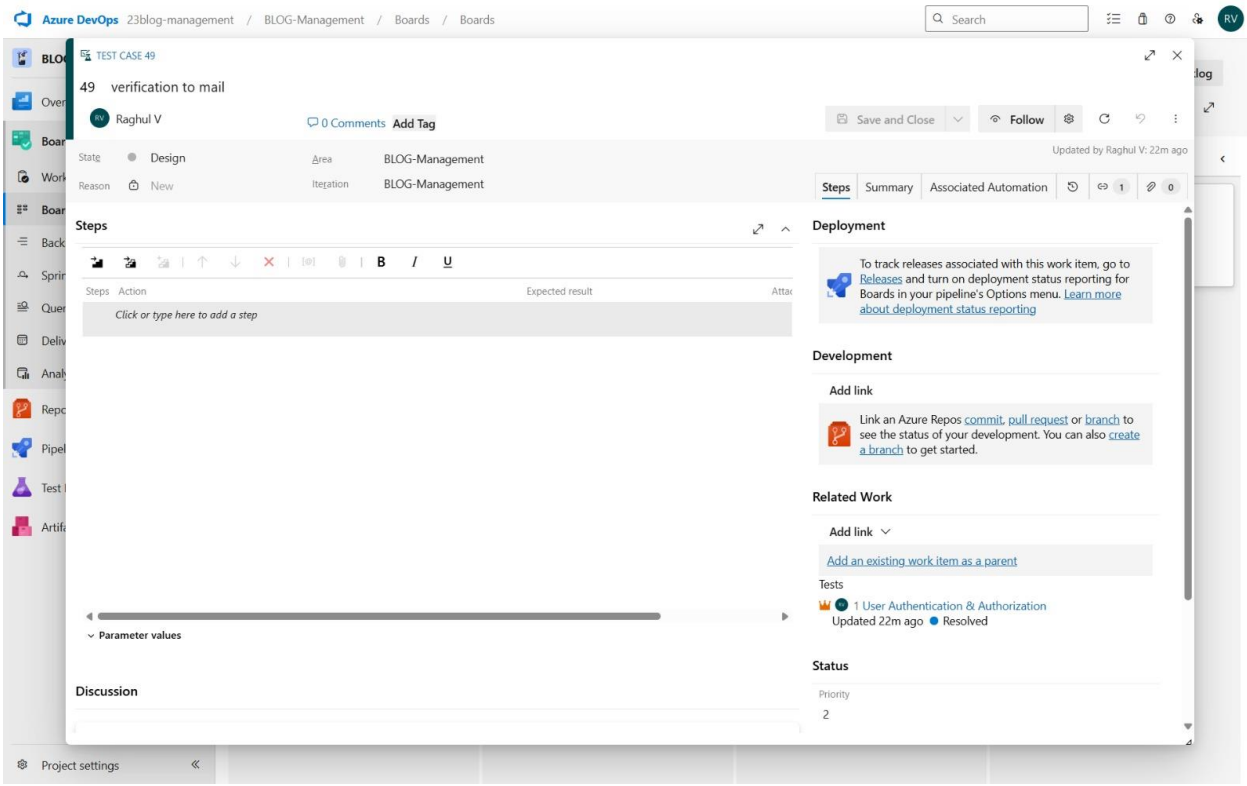
Test Suit: TS03 - Real-Time Metadata (ID: 88)

1. **TC07 – Real-Time Metadata Display**
    - **Action:**
        - Create blog.
        - Observe the metadata panel.
    - **Expected Results:**
        - Metadata is displayed and updates in real time.
    - **Type:** Happy Path

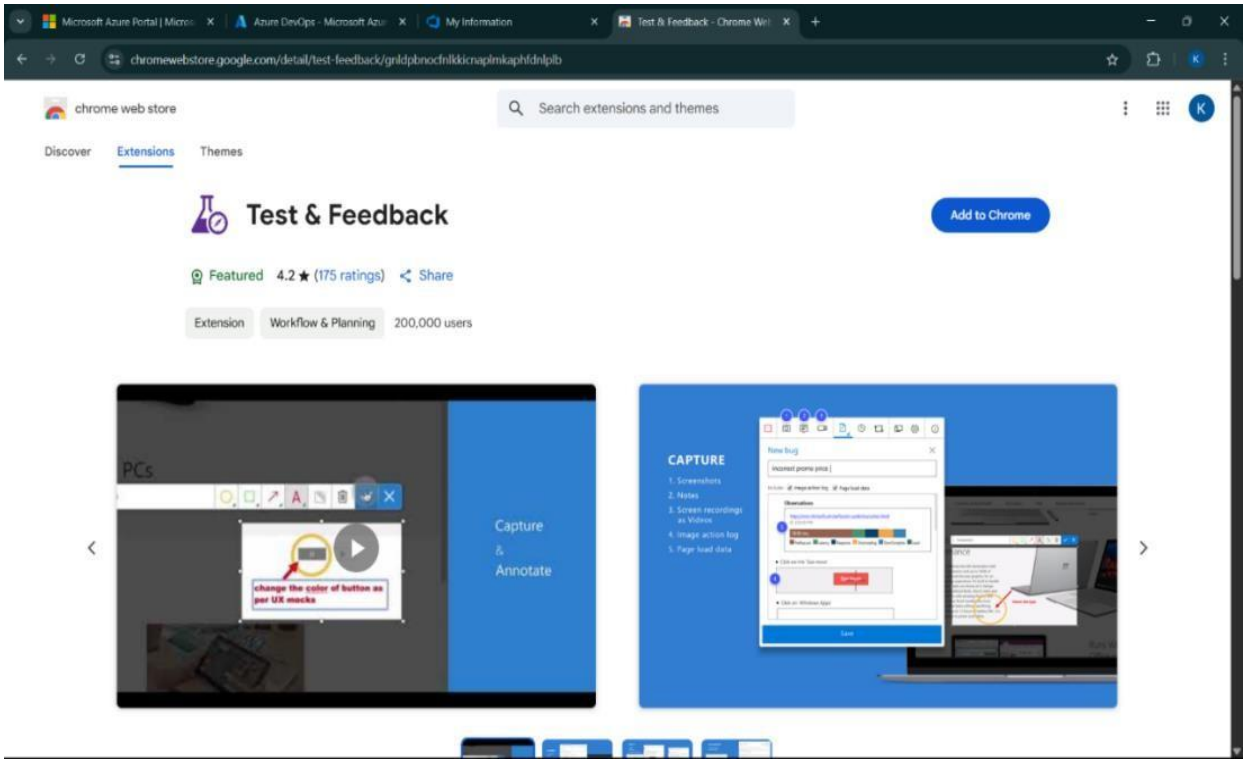
2. TC08 – Metadata Not Updating
    - **Action:**
        - Create a blog.
        - Observe the metadata panel.
    - **Expected Results:**
        - Metadata remains static or shows default/fallback message.
    - **Type:** Error Path

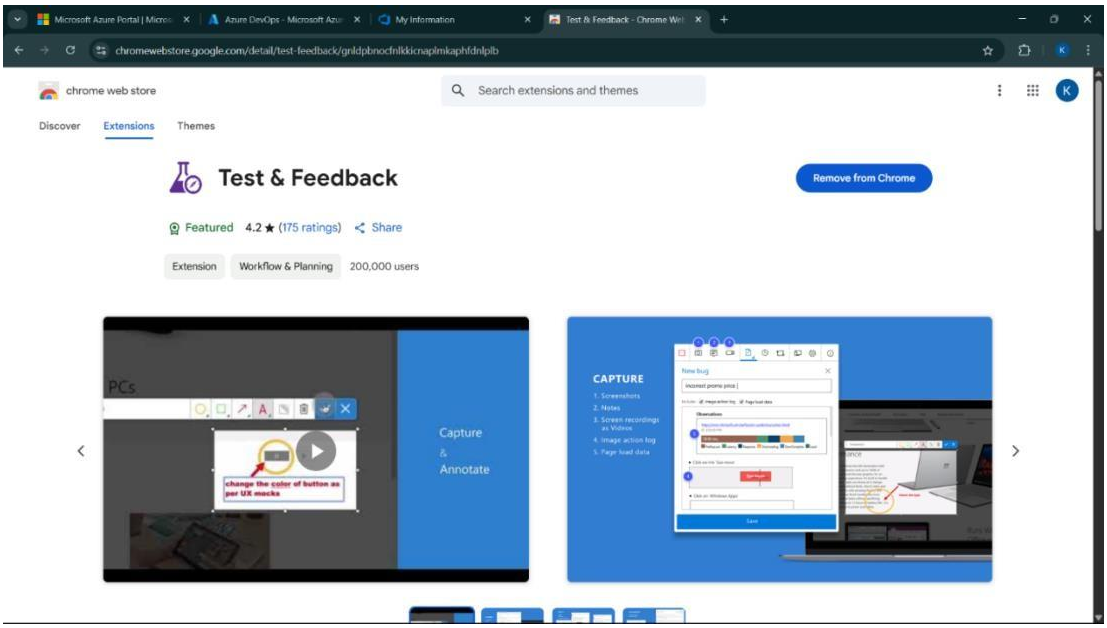
Test Cases



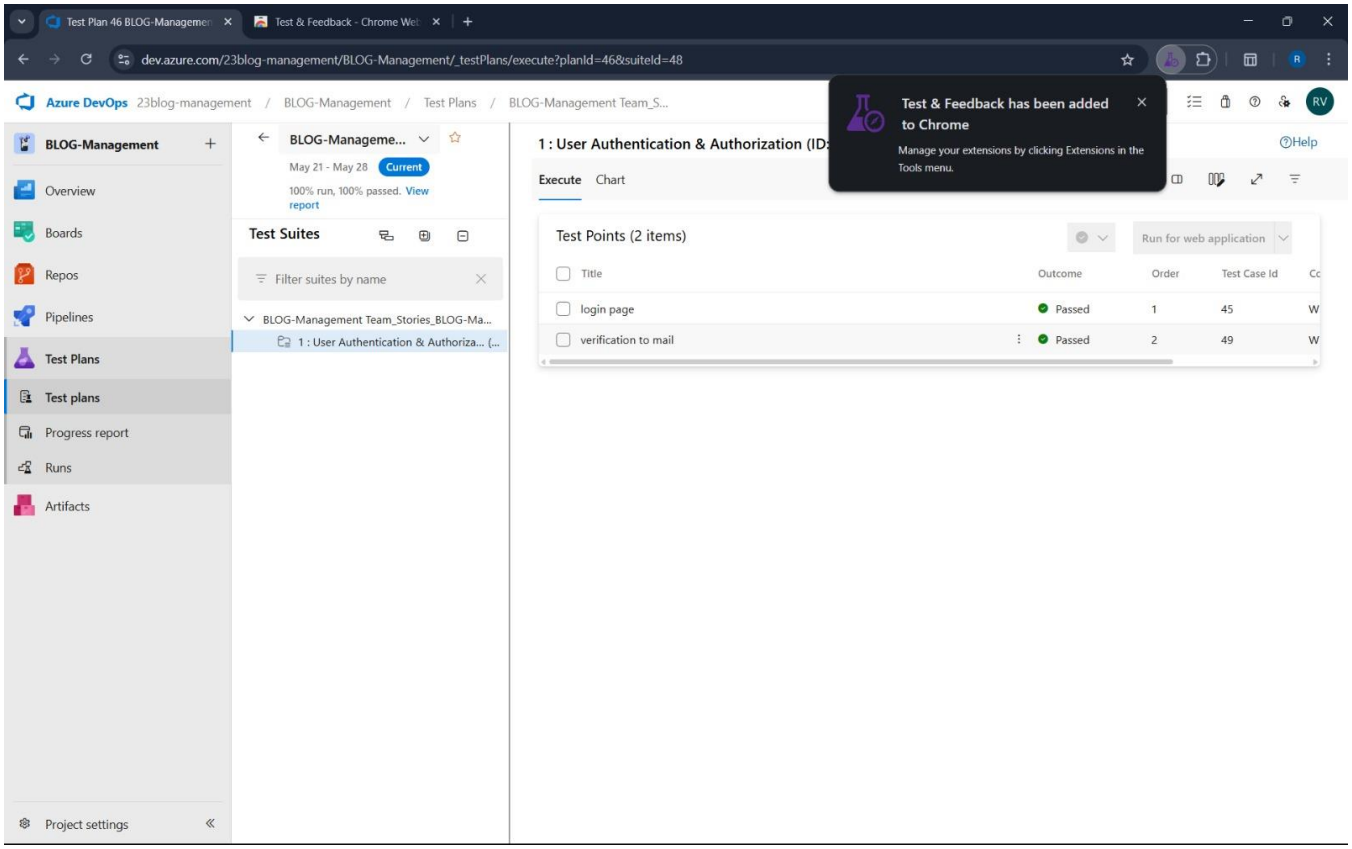


4. Installation of test

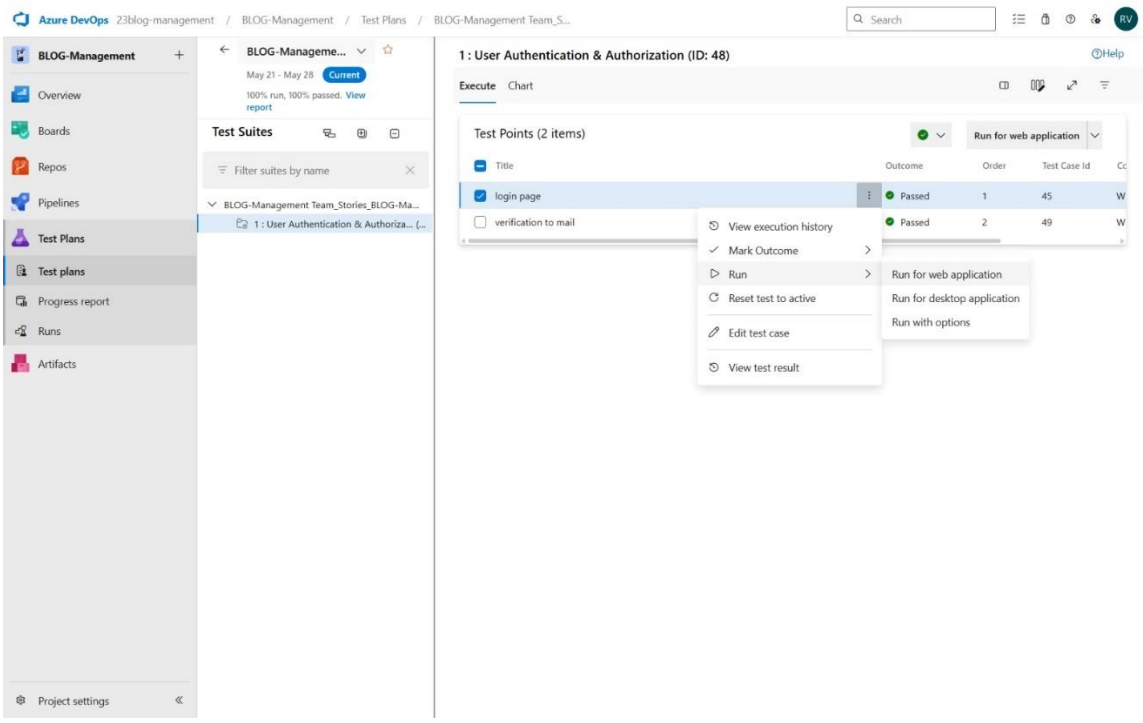


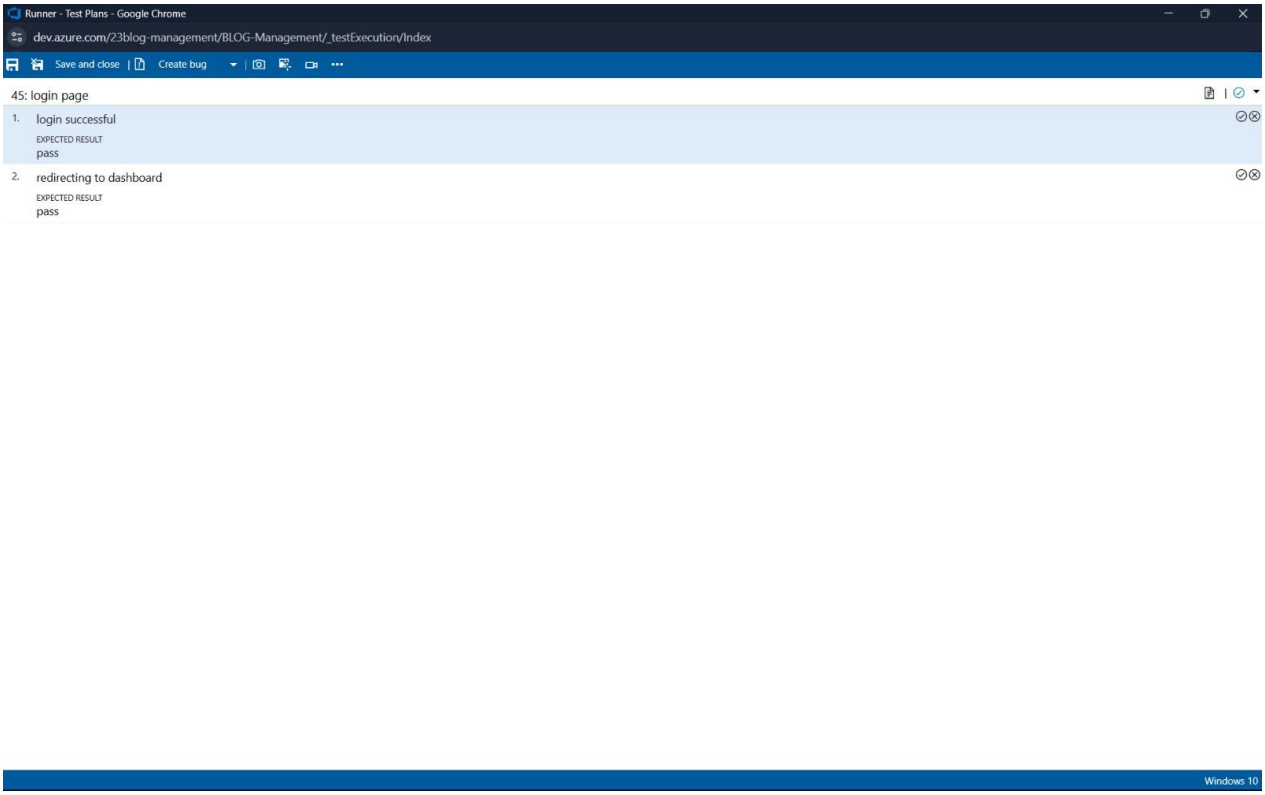


Test and feedback
Showing it as an extension

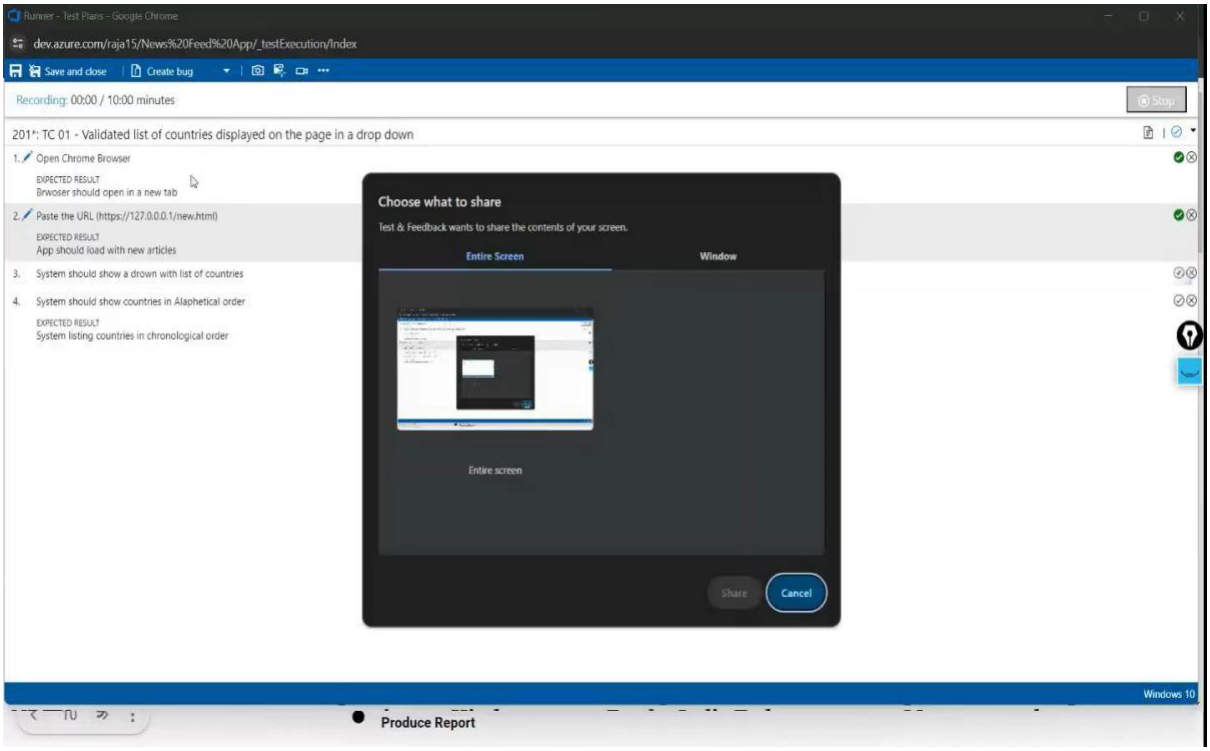


5. Running the test cases

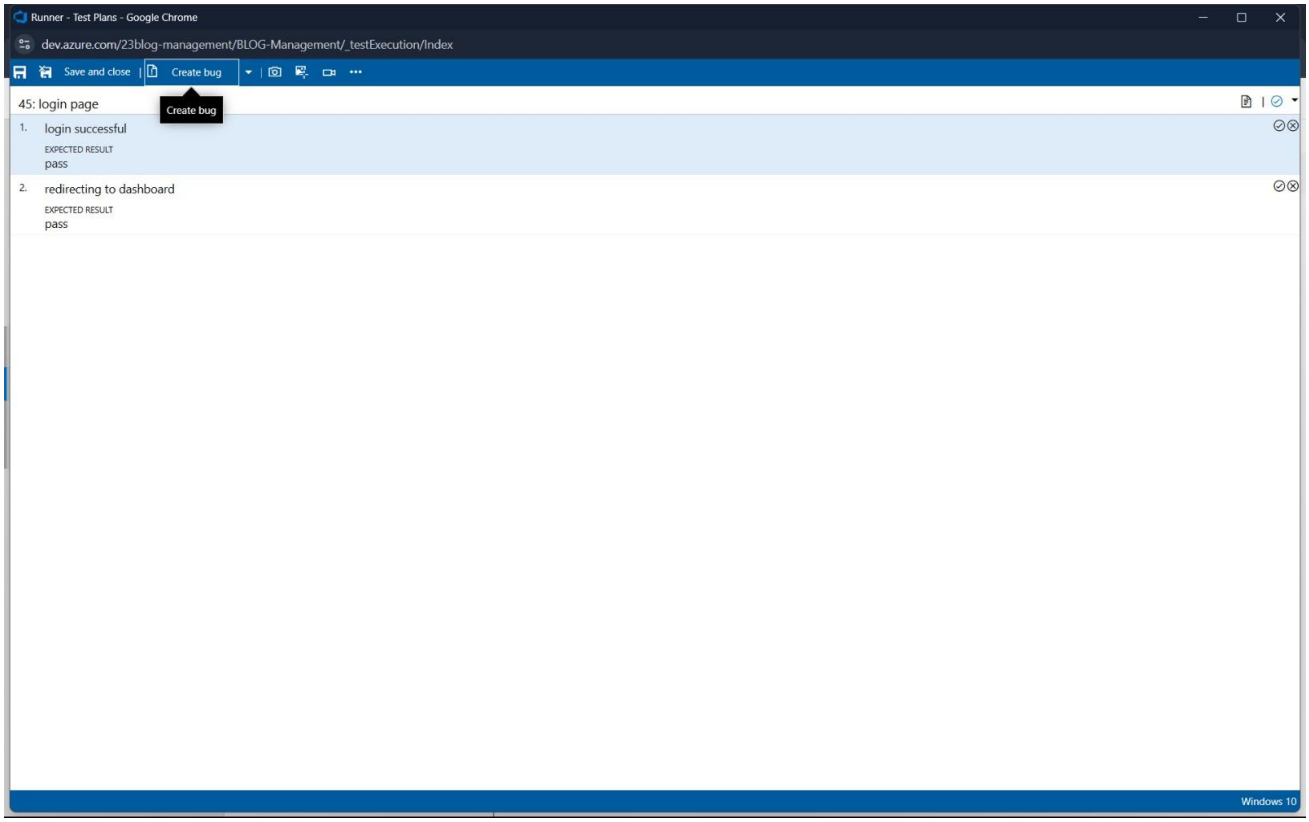


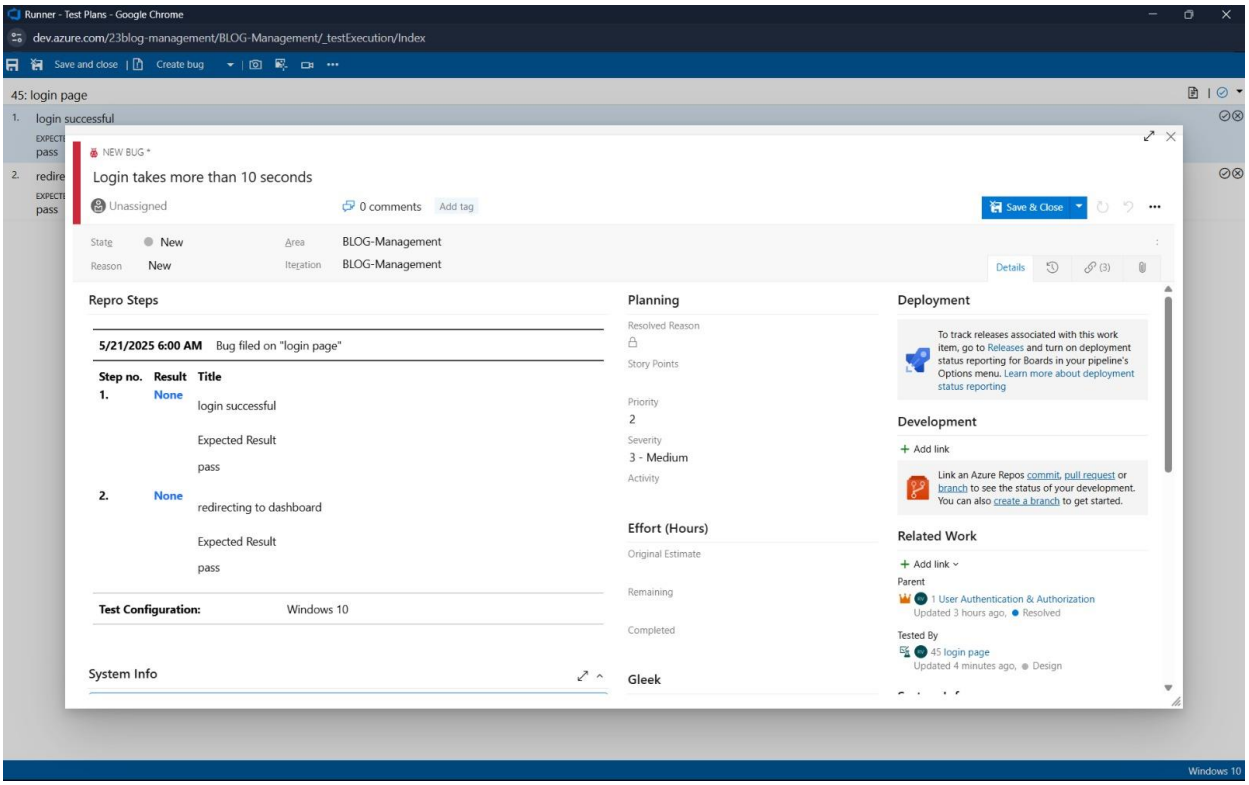


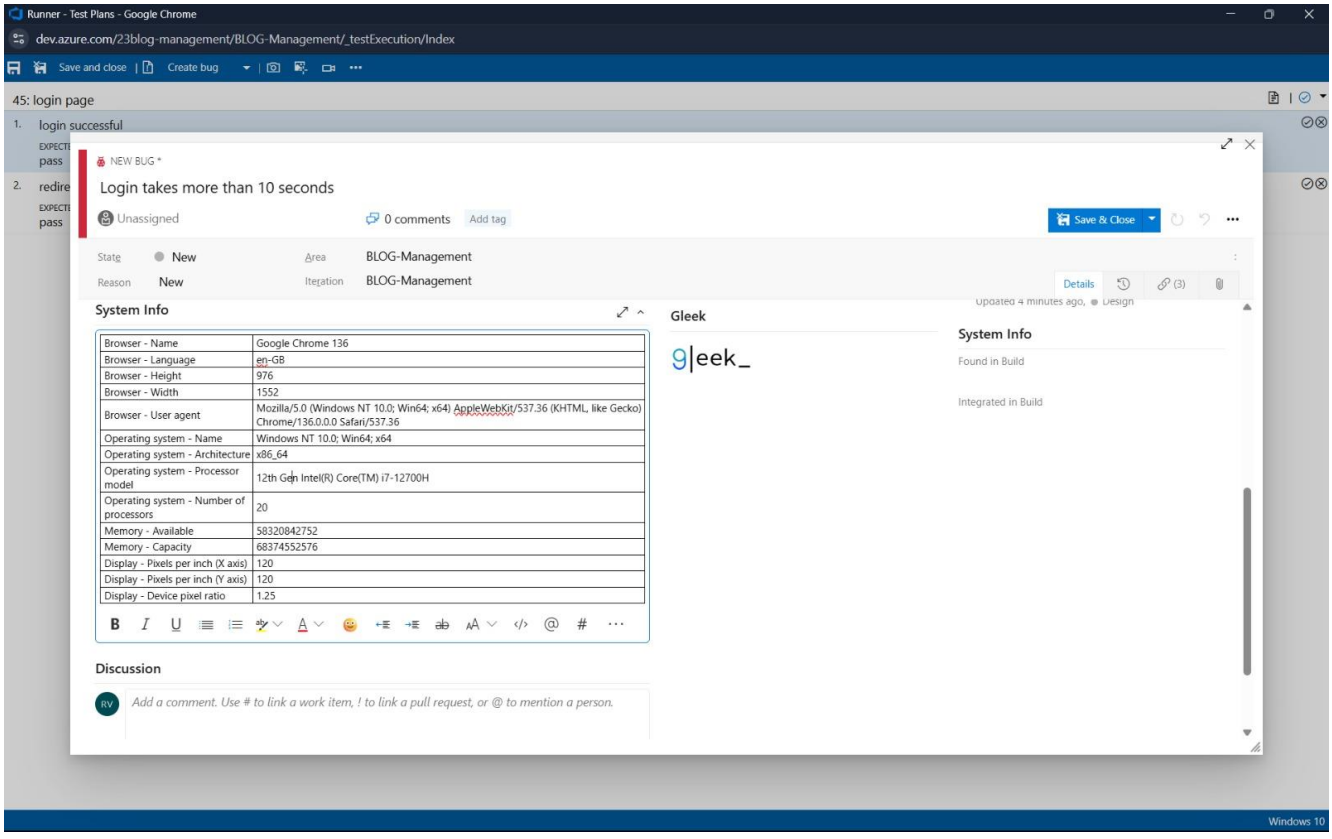
6. Recording the test case



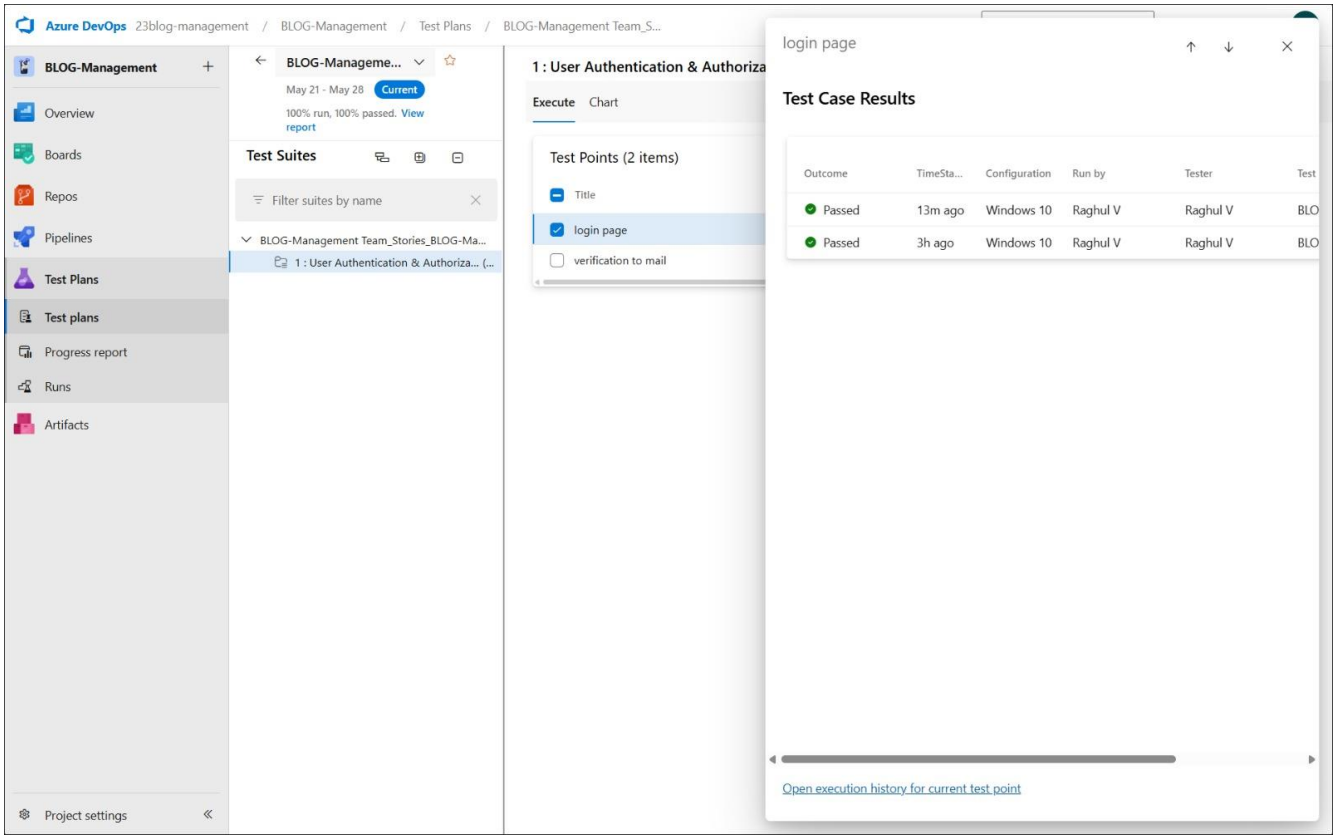
7. Creating the bug



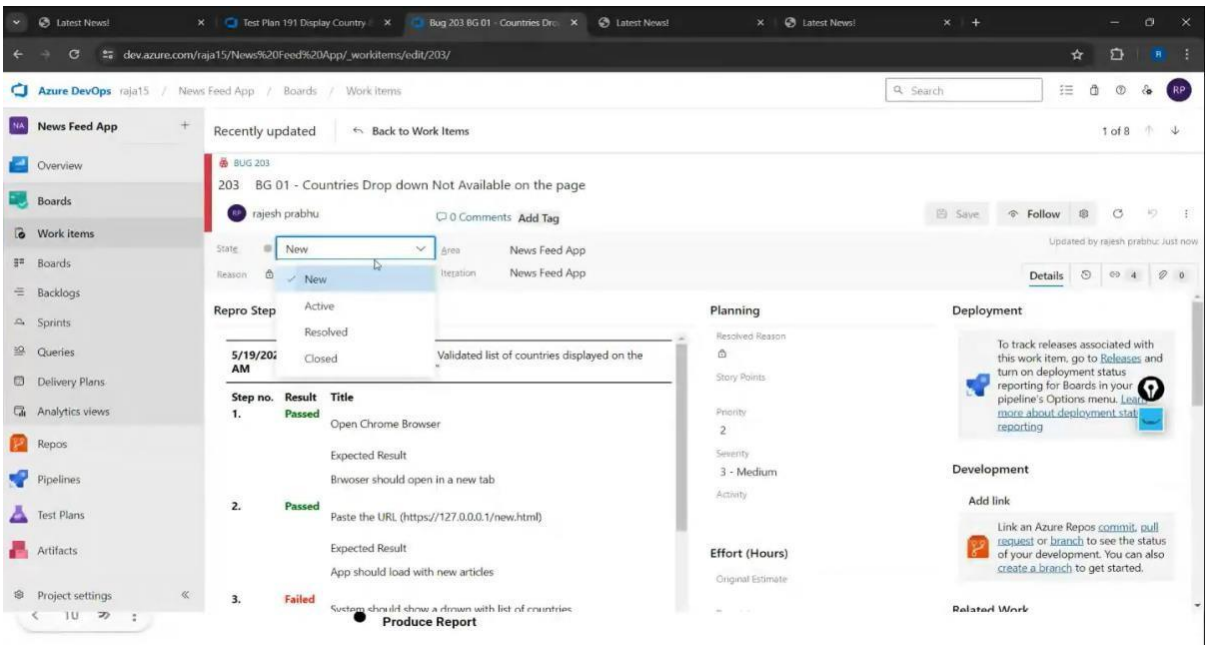

Runner - Test Plans - Google Chrome
dev.azure.com/23blog-management/BLOG-Management/_testExecution/Index
Save and close | Create bug
45: login page
1. login successful
NEW BUG *
Login takes more than 10 seconds
Unassigned
0 comments
Add tag
Save & Close
State New
Reason New
Area BLOG-Management
Iteration BLOG-Management
Details
Repro Steps
5/21/2025 6:00 AM Bug filed on "login page"
Step no. Result Title
1. None login successful
Expected Result
pass
2. None redirecting to dashboard
Expected Result
pass
Test Configuration: Windows 10
System Info
Planning
Resolved Reason
Story Points
Priority
2
Severity
3 - Medium
Activity
Effort (Hours)
Original Estimate
Remaining
Completed
Gleek
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
Parent
1 User Authentication & Authorization
Updated 3 hours ago, Resolved
Tested By
45 login page
Updated 4 minutes ago, Design
Windows 10


Runner - Test Plans - Google Chrome
dev.azure.com/23blog-management/BLOG-Management/_testExecution/Index
Save and close | Create bug
45: login page
1. login successful
NEW BUG *
Login takes more than 10 seconds
Unassigned
0 comments
Add tag
Save & Close
State New
Reason New
Area BLOG-Management
Iteration BLOG-Management
Details
System Info
Browser - Name Google Chrome 136
Browser - Language en-GB
Browser - Height 976
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 x86_64
Operating system - Processor model 12th Gen Intel(R) Core(TM) i7-12700H
Operating system - Number of processors 20
Memory - Available 58320842752
Memory - Capacity 68374552576
Display - Pixels per inch (X axis) 120
Display - Pixels per inch (Y axis) 120
Display - Device pixel ratio 1.25
Discussion
Add a comment. Use # to link a work item, ! to link a pull request, or @ to mention a person.
Gleek
9|eek_
System Info
Found in Build
Integrated in Build
Windows 10

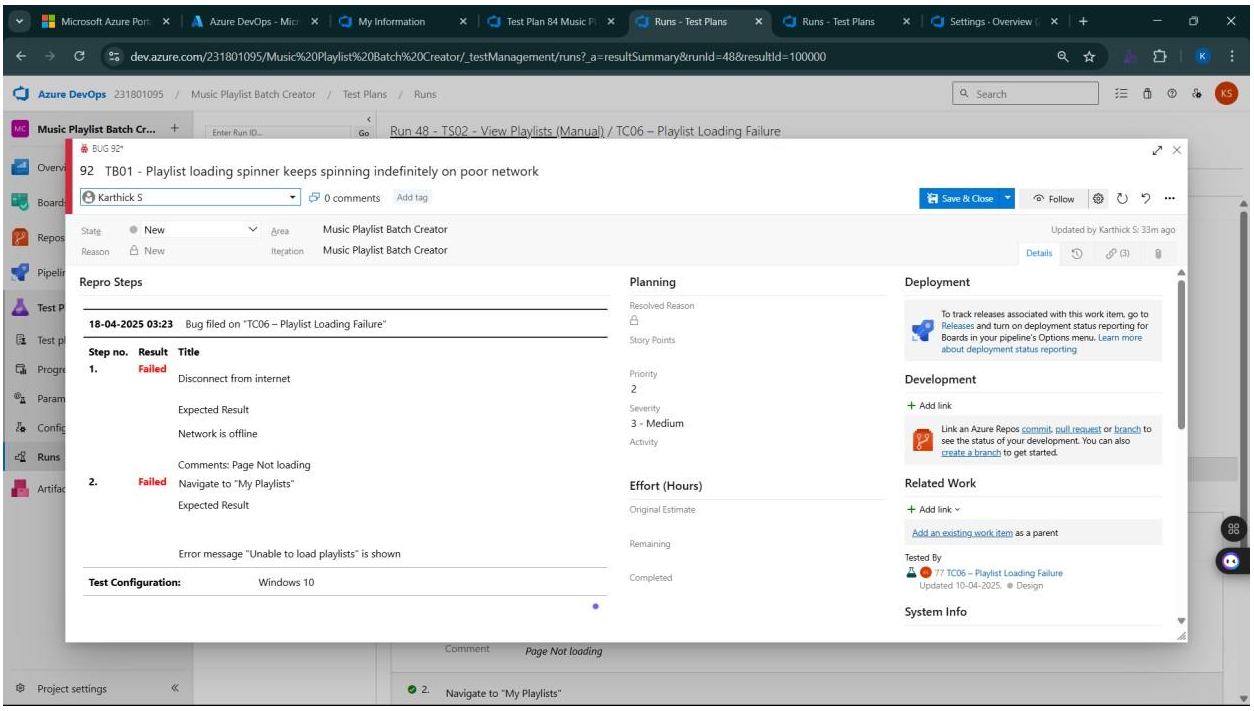
8. Test case results



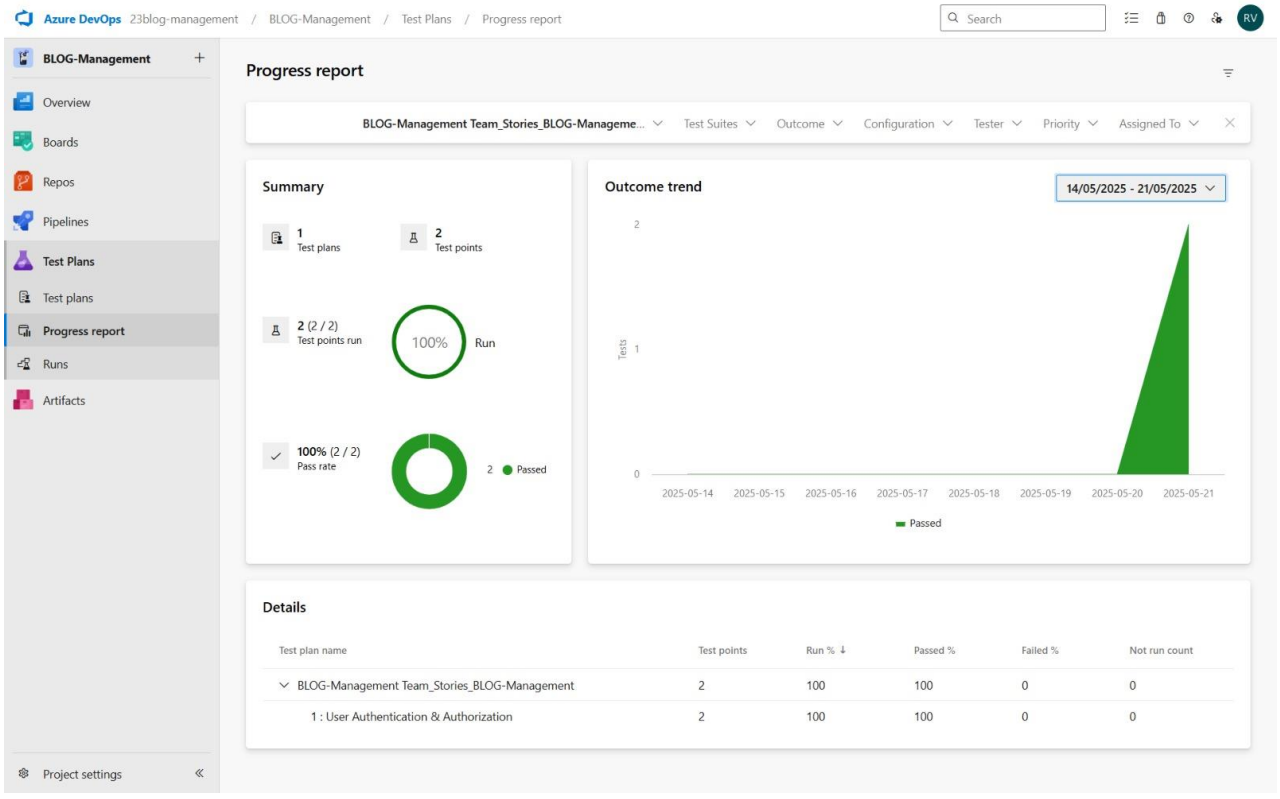
9. Test report summary



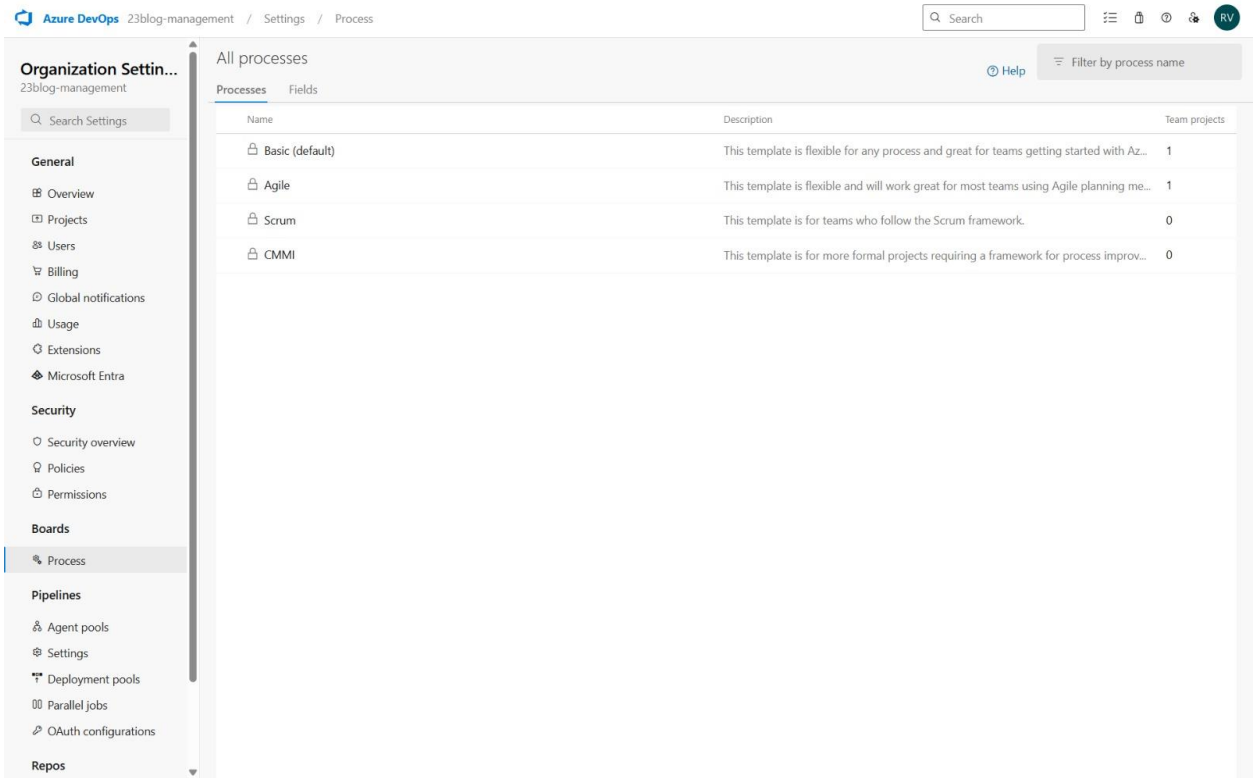
- Assigning bug to the developer and changing state

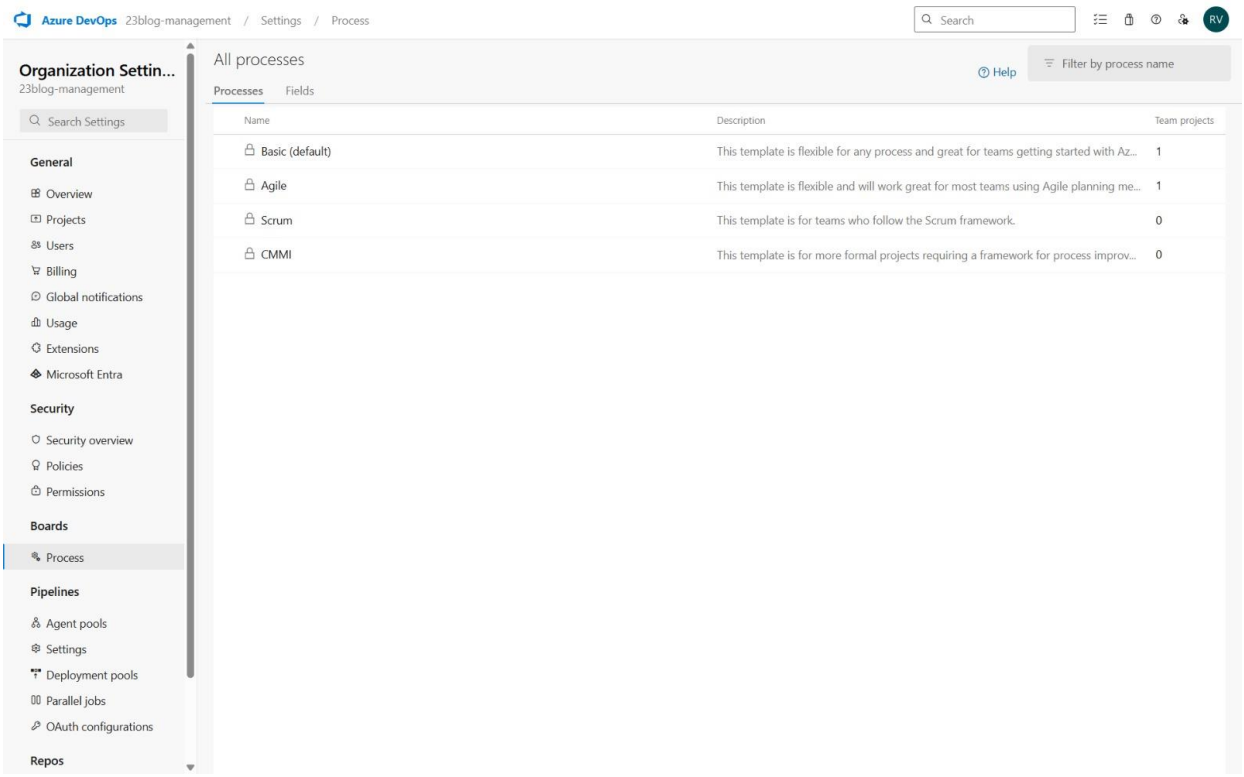


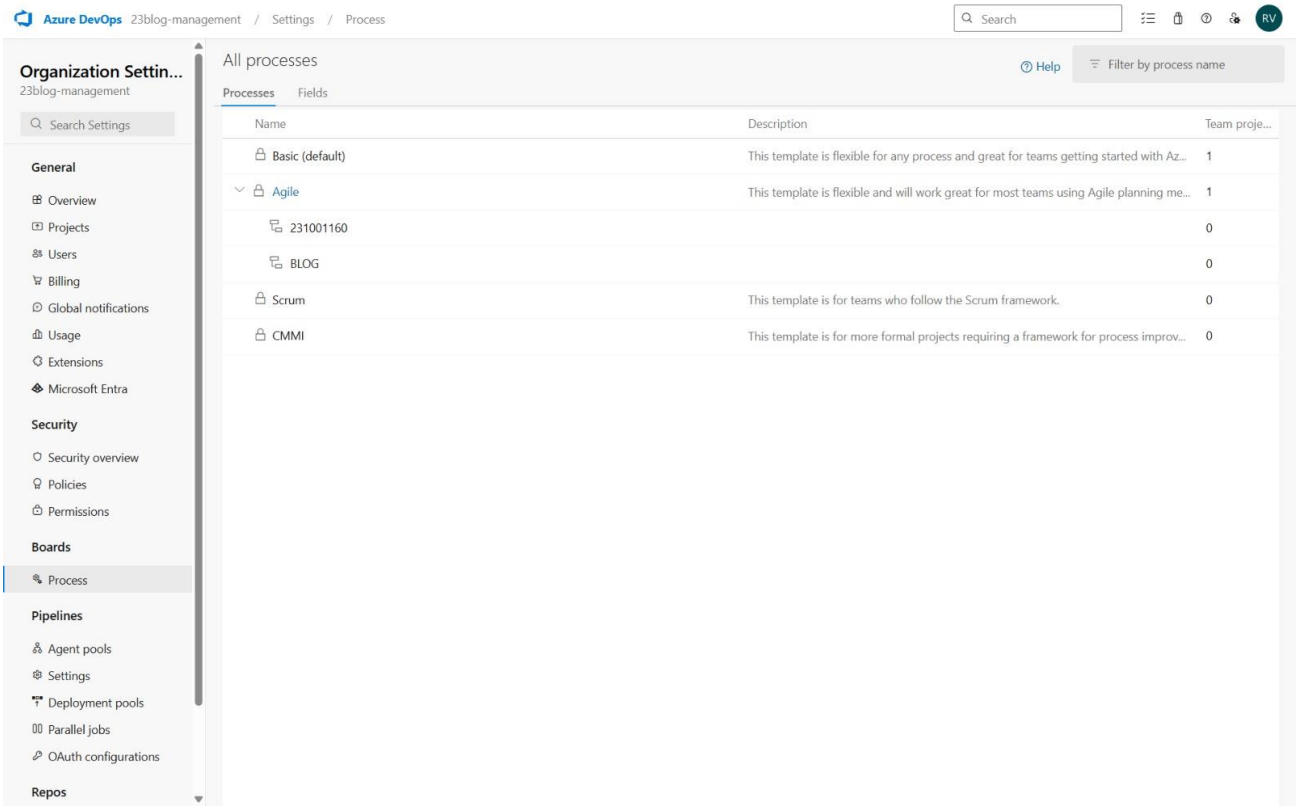
10. Progress report



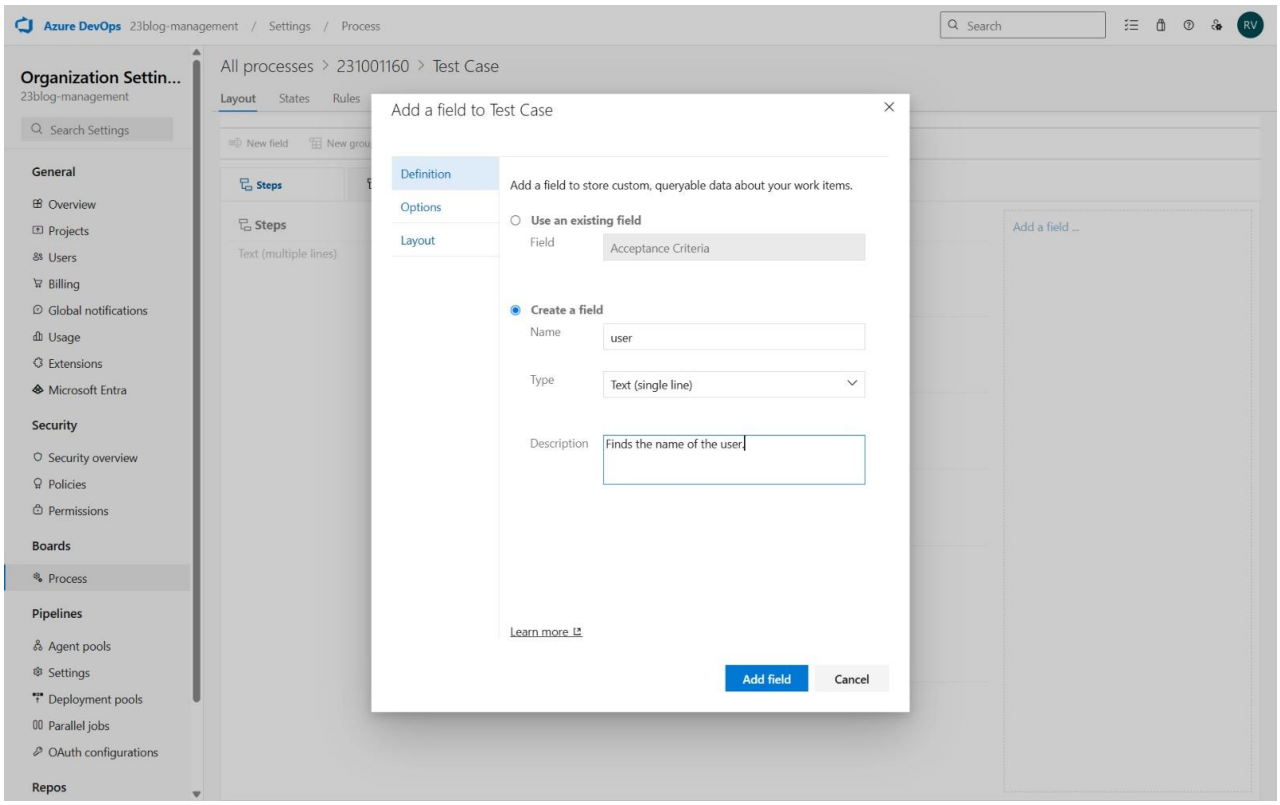
11. Changing the test template

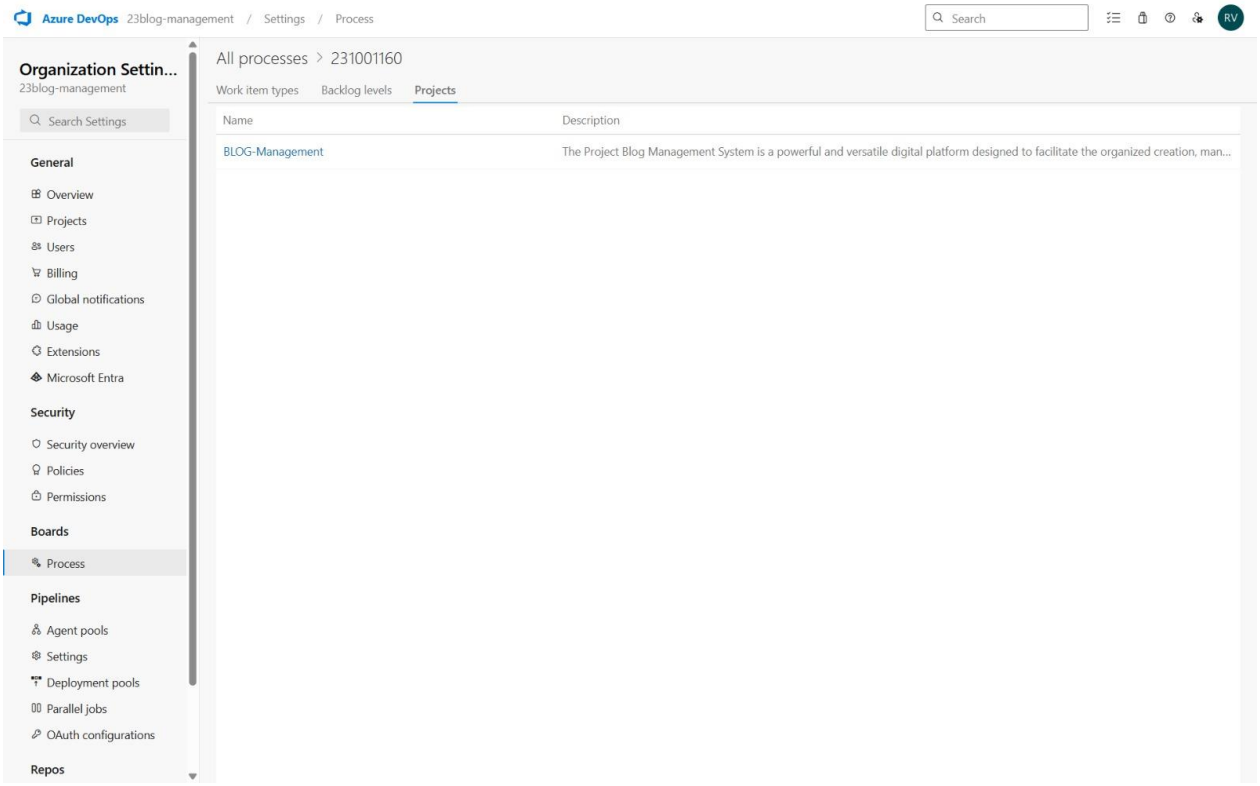


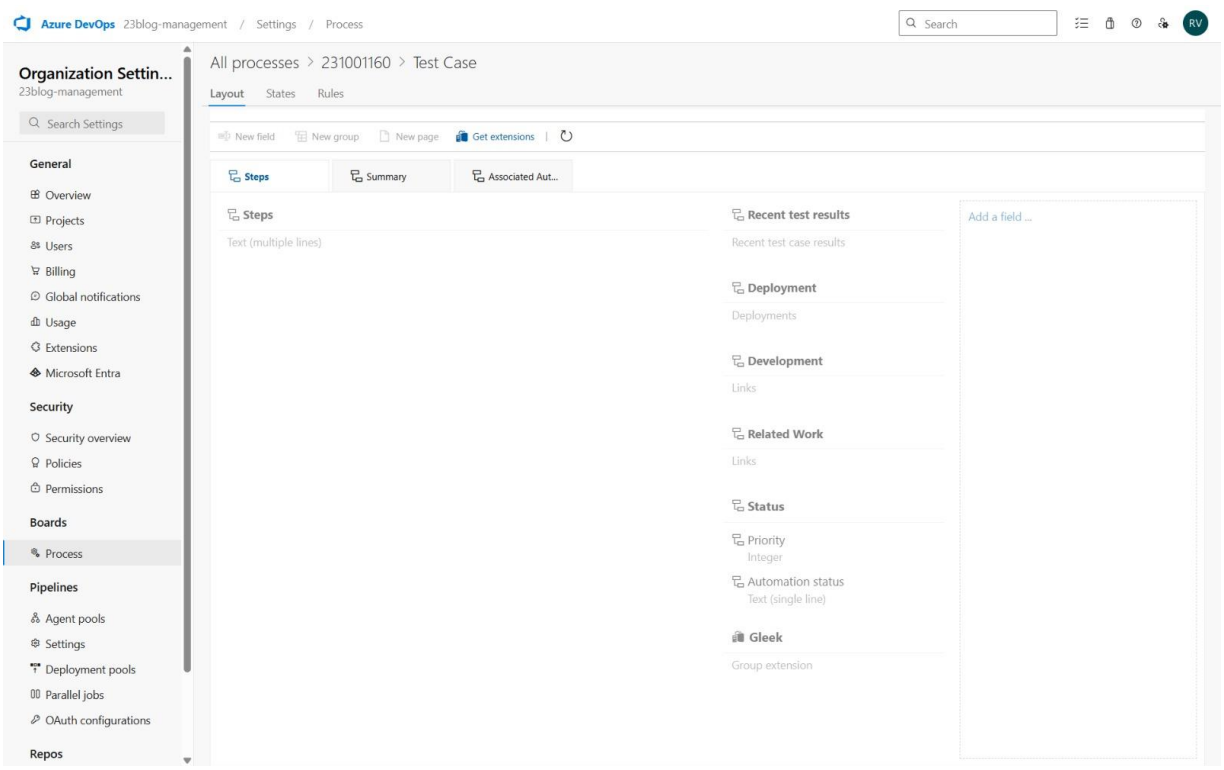

Azure DevOps 23blog-management / Settings / Process
Search
RV
Organization Settin...
23blog-management
Search Settings
General
Overview
Projects
Users
Billing
Global notifications
Usage
Extensions
Microsoft Entra
Security
Security overview
Policies
Permissions
Boards
Process
Pipelines
Agent pools
Settings
Deployment pools
Parallel jobs
OAuth configurations
Repos
All processes
Help
Filter by process name
Processes
Fields
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


Azure DevOps 23blog-management / Settings / Process
Search
RV
Organization Settin...
23blog-management
Search Settings
General
Overview
Projects
Users
Billing
Global notifications
Usage
Extensions
Microsoft Entra
Security
Security overview
Policies
Permissions
Boards
Process
Pipelines
Agent pools
Settings
Deployment pools
Parallel jobs
OAuth configurations
Repos
All processes
Help
Filter by process name
Processes
Fields
Name
Description
Team proje...
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
231001160
0
BLOG
0
Scrum
This template is for teams who follow the Scrum framework.
0
CMMI
This template is for more formal projects requiring a framework for process improv...
0

12. View the new test case template







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:

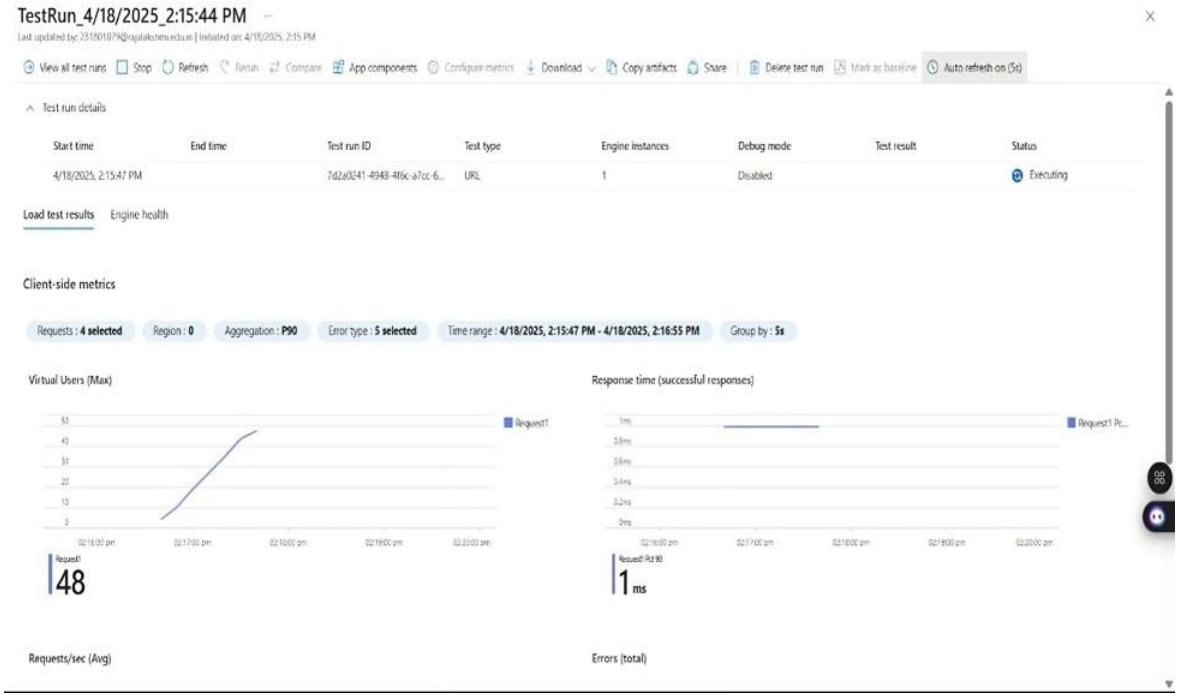
1. Sign in to Azure Portal
   Go to https://portal.azure.com and log in.
2. Create the Resource
   - Go to *Create a resource* → Search for "Azure Load Testing".
   - Select Azure Load Testing and click Create.
3. Fill in the Configuration Details
   - Subscription*:* Choose your Azure subscription.
   - *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.

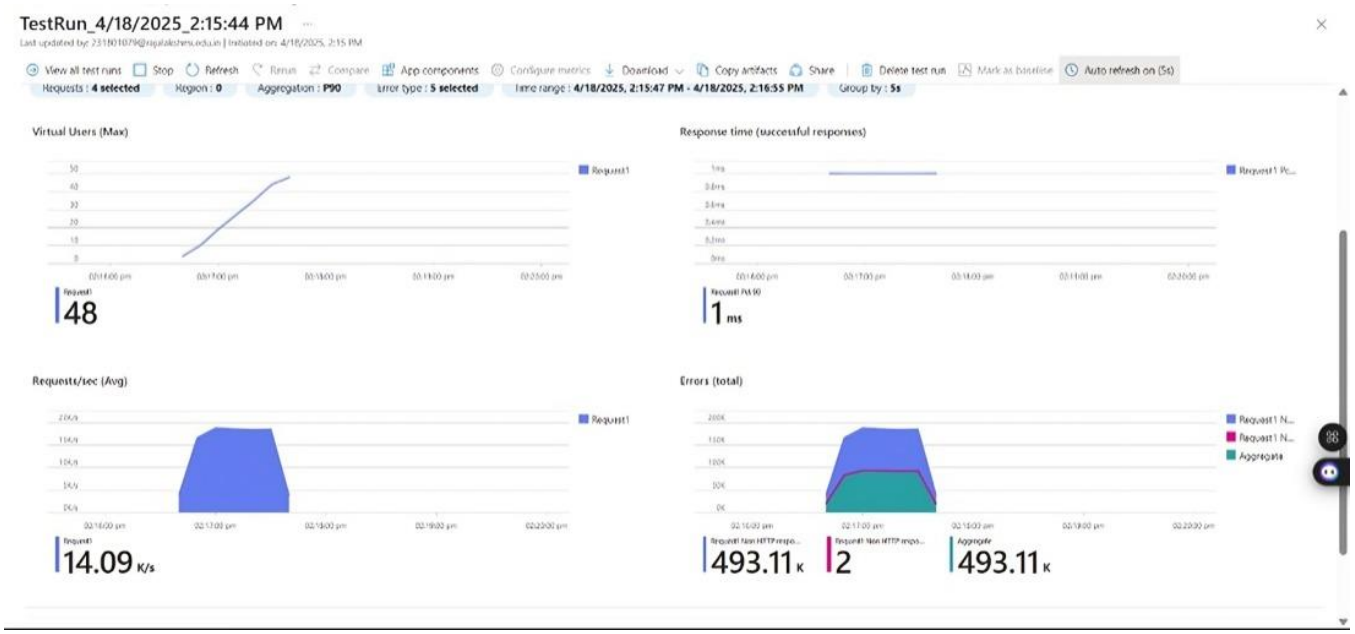
Steps to Create and Run a Load Test:

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:

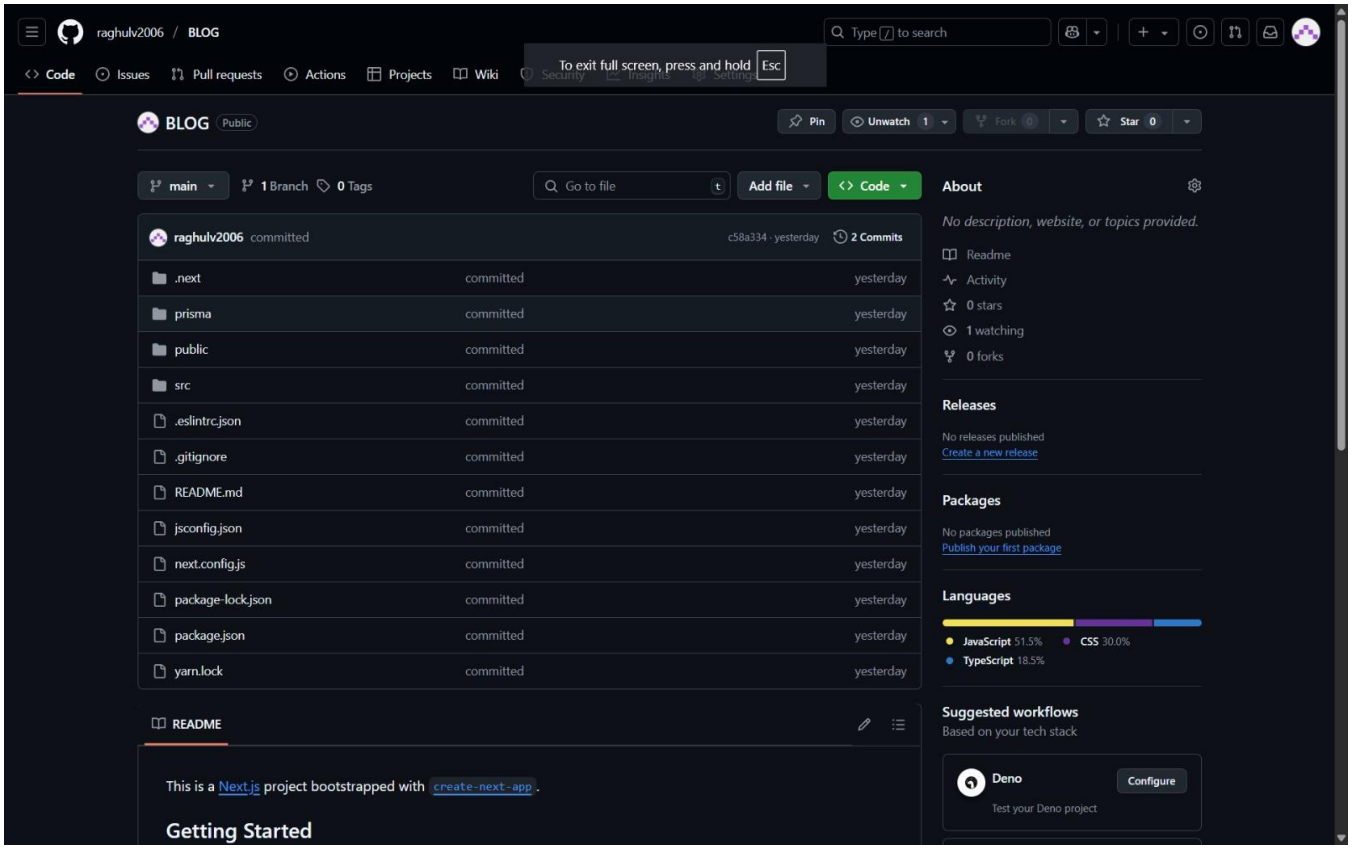
Successfully created the Azure Load Testing resource and executed a load test to assess the performance of the specified endpoint.

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 Music BLOG-management 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.