

**DEPARTMENT OF INFORMATION TECHNOLOGY LAB MANUAL**

**CS23432–Software Construction**

**(REGULATION 2023)**

**RAJALAKSHMI ENGINEERING COLLEGE**

**Thandalam,Chennai-602015**

Name: Dinesh S

Register No: 231001040 Year/Branch/Section: 2nd / IT / FA Semester: IV

AcademicYear: 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 TestCases. |
| 9. | 25/04/25 | Load Testing and Pipelines. |
| 10. | 02/05/25 | GitHub: Project Structure & Naming Conventions. |

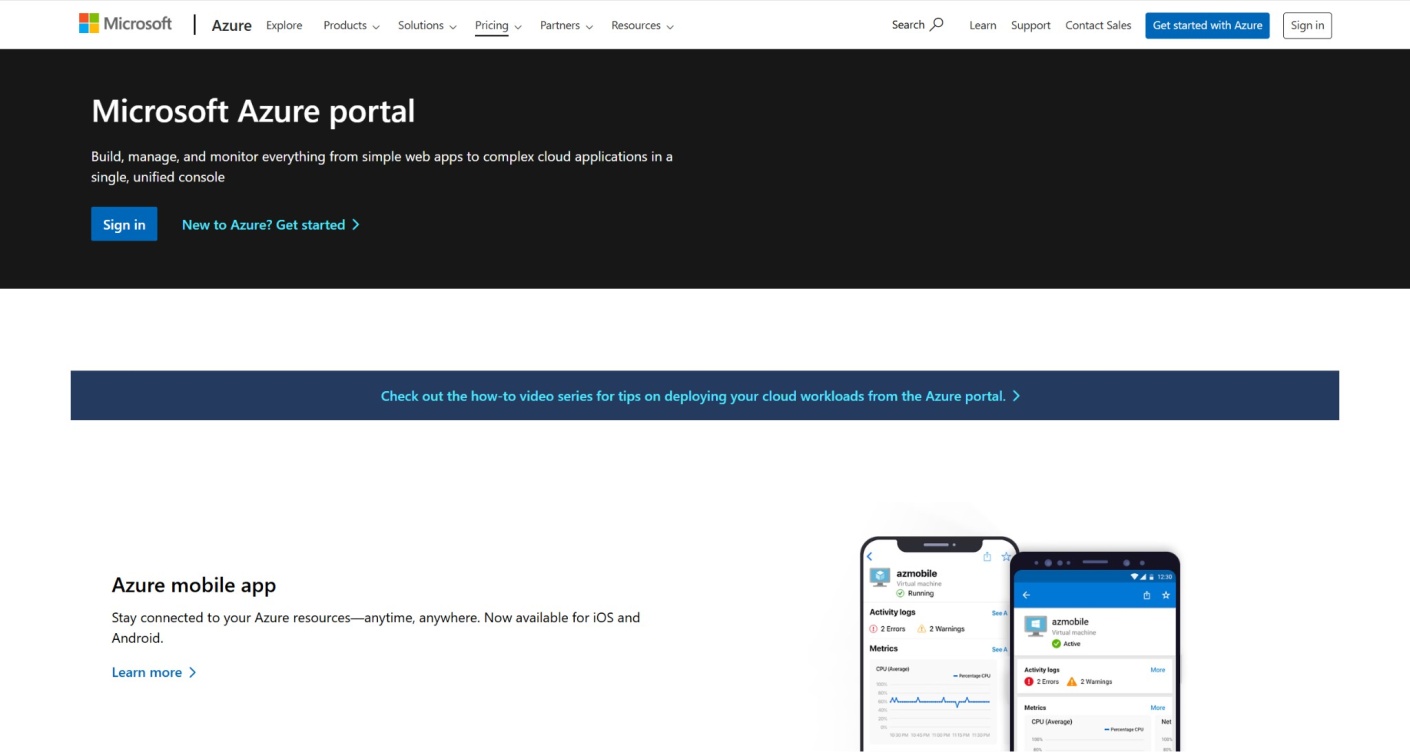
|  |  |
| --- | --- |
| **Exp.No:1** | **AZURE DEVOPS ENVIRONMENT SETUP** |
| **Date:22/01/2025** |

**AIM:**To setup and access the AzureDevOps environment by creating an organization through the Azure Portal.

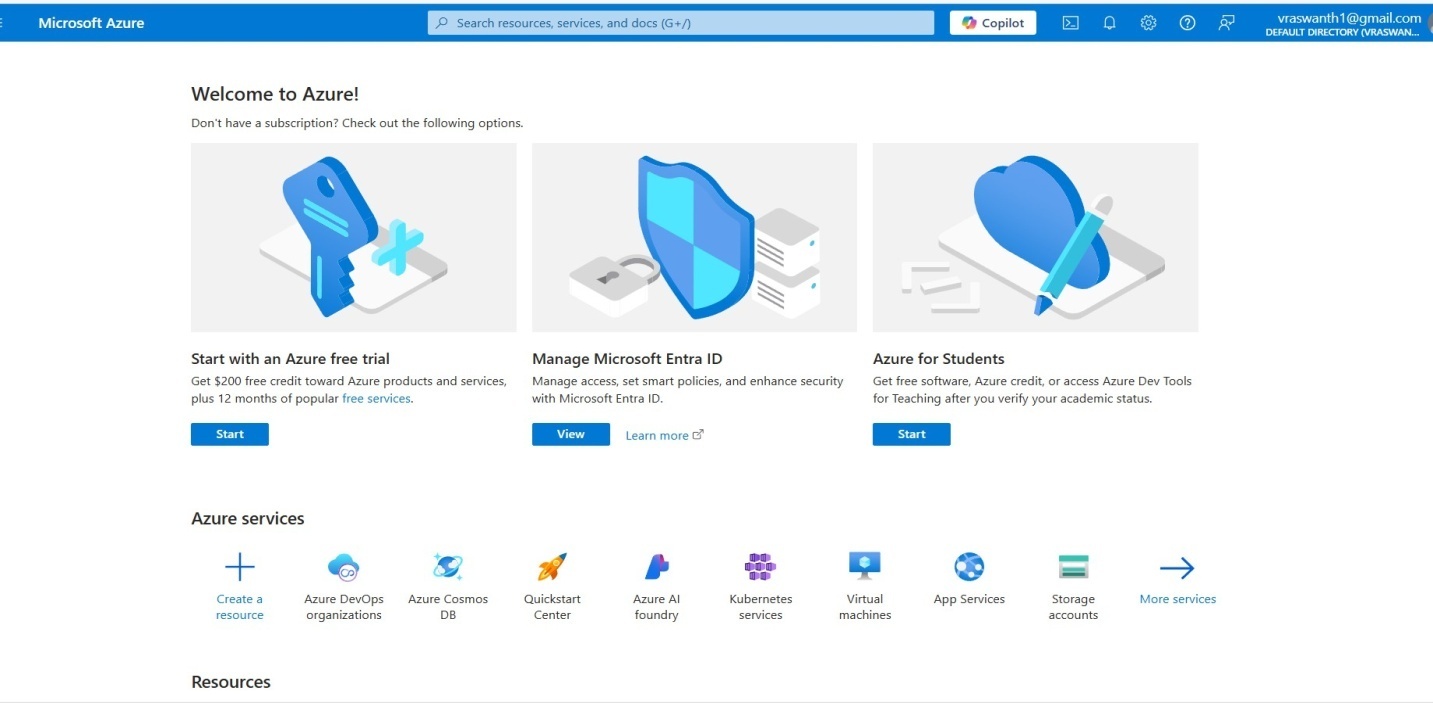
## INSTALLATION

1. Open your web browser and go to the Azure website: [https://azure.microsoft.com/en-us/getstarted/azure-portal.](https://azure.microsoft.com/en-us/get-started/azure-portal)Sign in using your Microsoft account credentials.

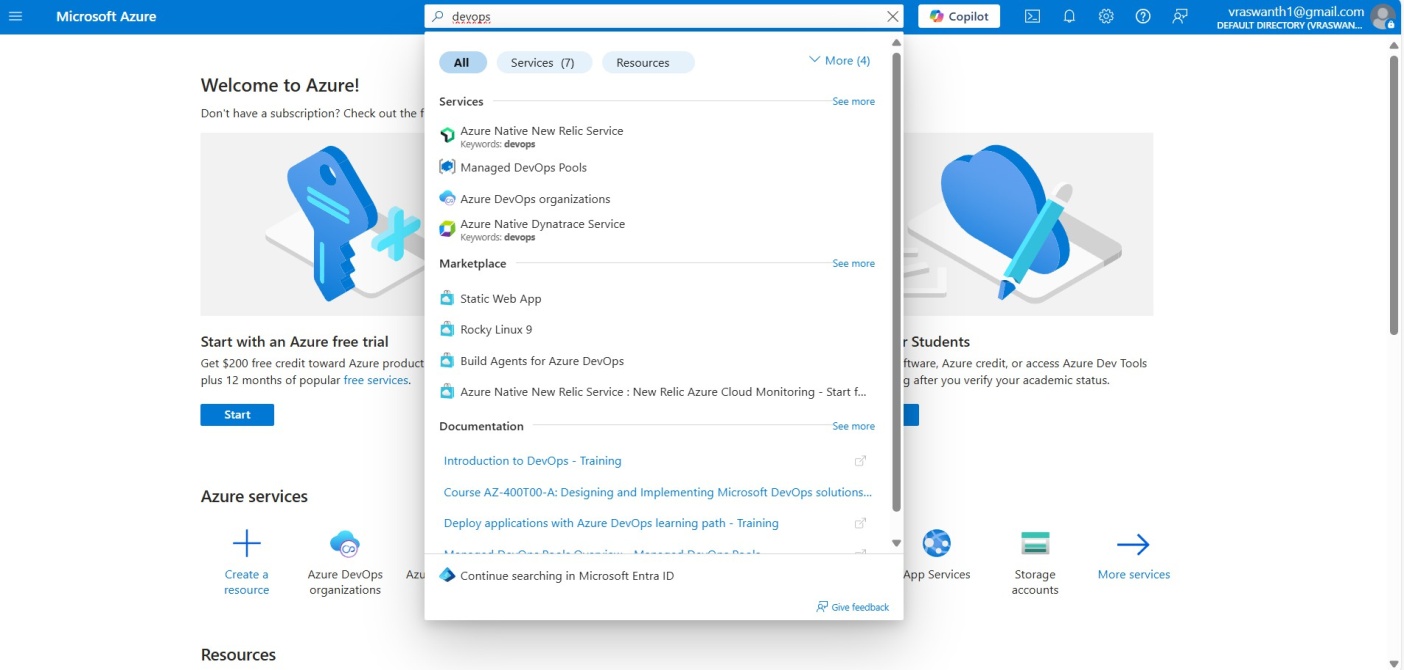
If you don't have a Microsoft account,you can create one here:<https://signup.live.com/?lic=1>



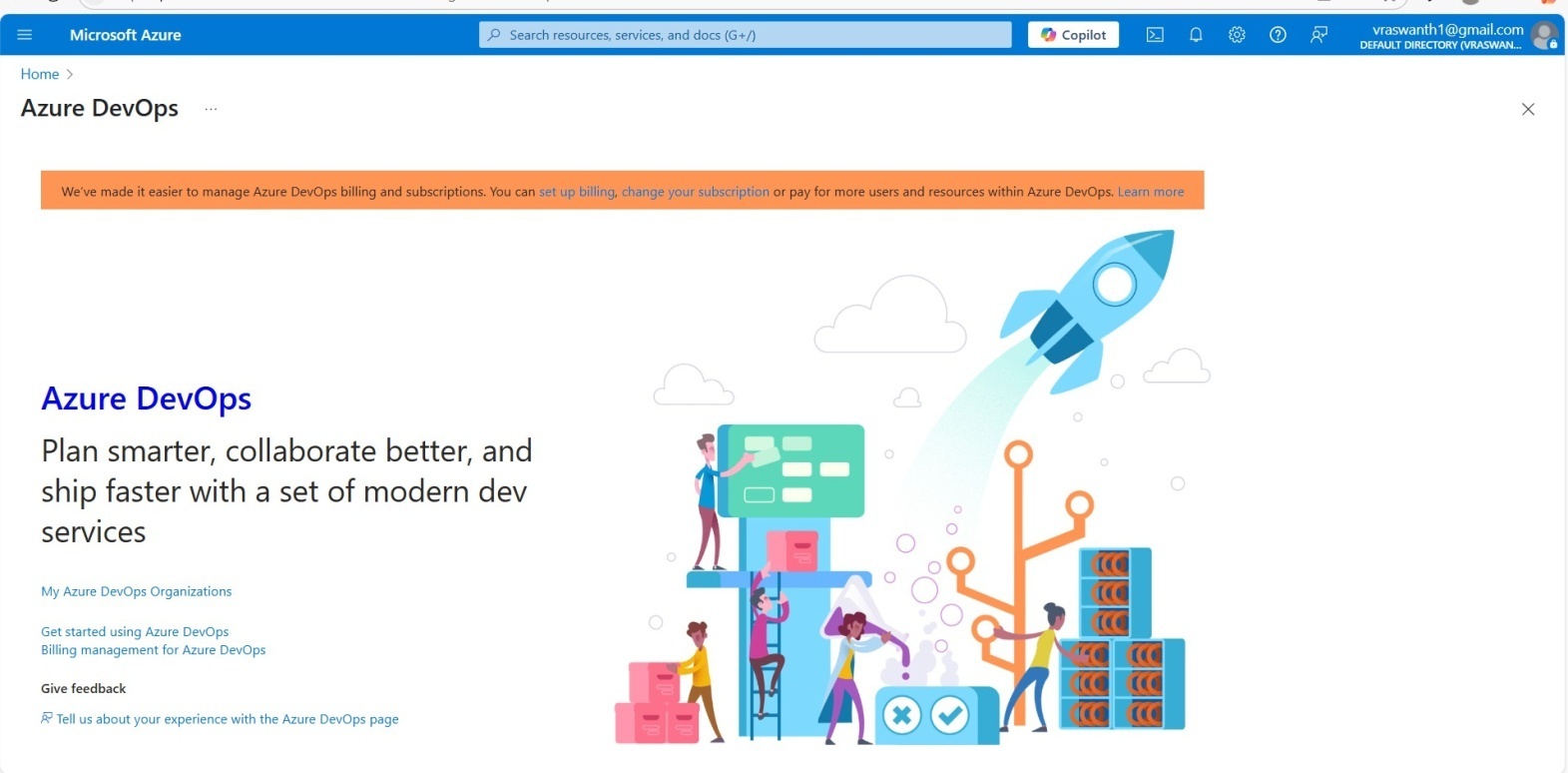
1. Azure home page



1. Open DevOps environment in theAzure platform by typing ***Azure DevOp sOrganizations*** in the search bar.



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



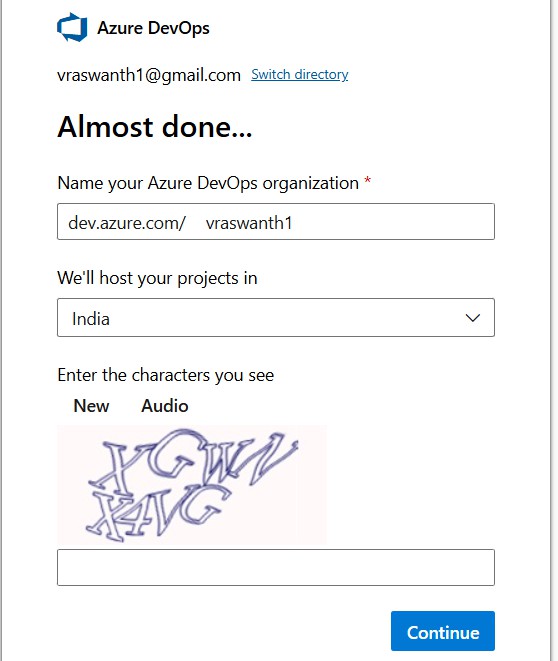
**RESULT:**Successfully accessed the Azure DevOps environment and created an ew organization through the Azure portal.

|  |  |
| --- | --- |
| **Exp.No:2** | **AZURE DEVOPS PROJECT SETUP AND USER STORY MANAGEMENT** |
| **Date:07/02/2025** |

**AIM:**To setup an Azure DevOps project for efficient collaboration and agile work management.

## STEPS:

1. Create An Azure Account

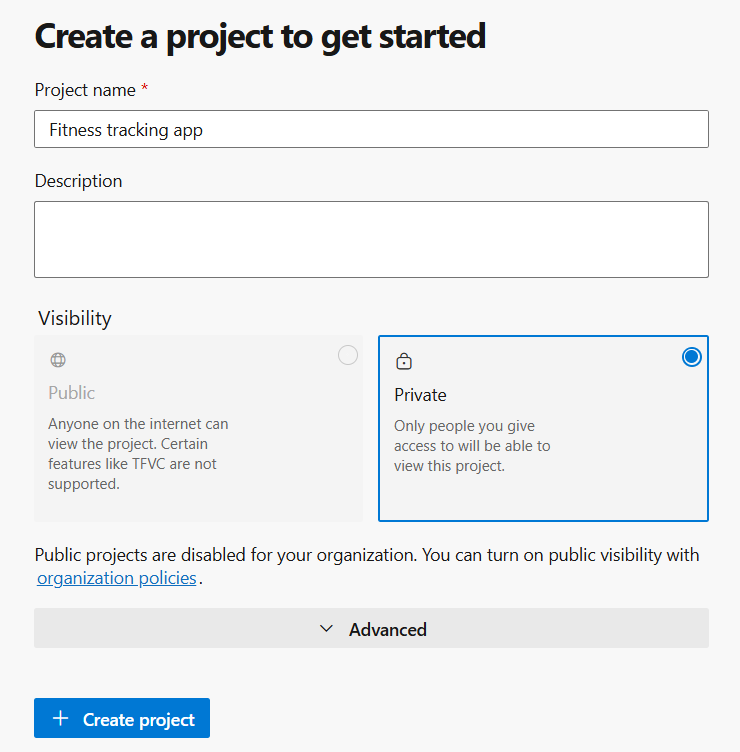


1. Create the First Project inYour Organization
   1. After the organization is setup,you’ll need to create your first **project**.This is where you'll begin to manage code, pipelines, work items, and more.
   2. On the organization’s **Homepage**,click on the **NewProject**button.
   3. Enter the project name,description,and visibility options:

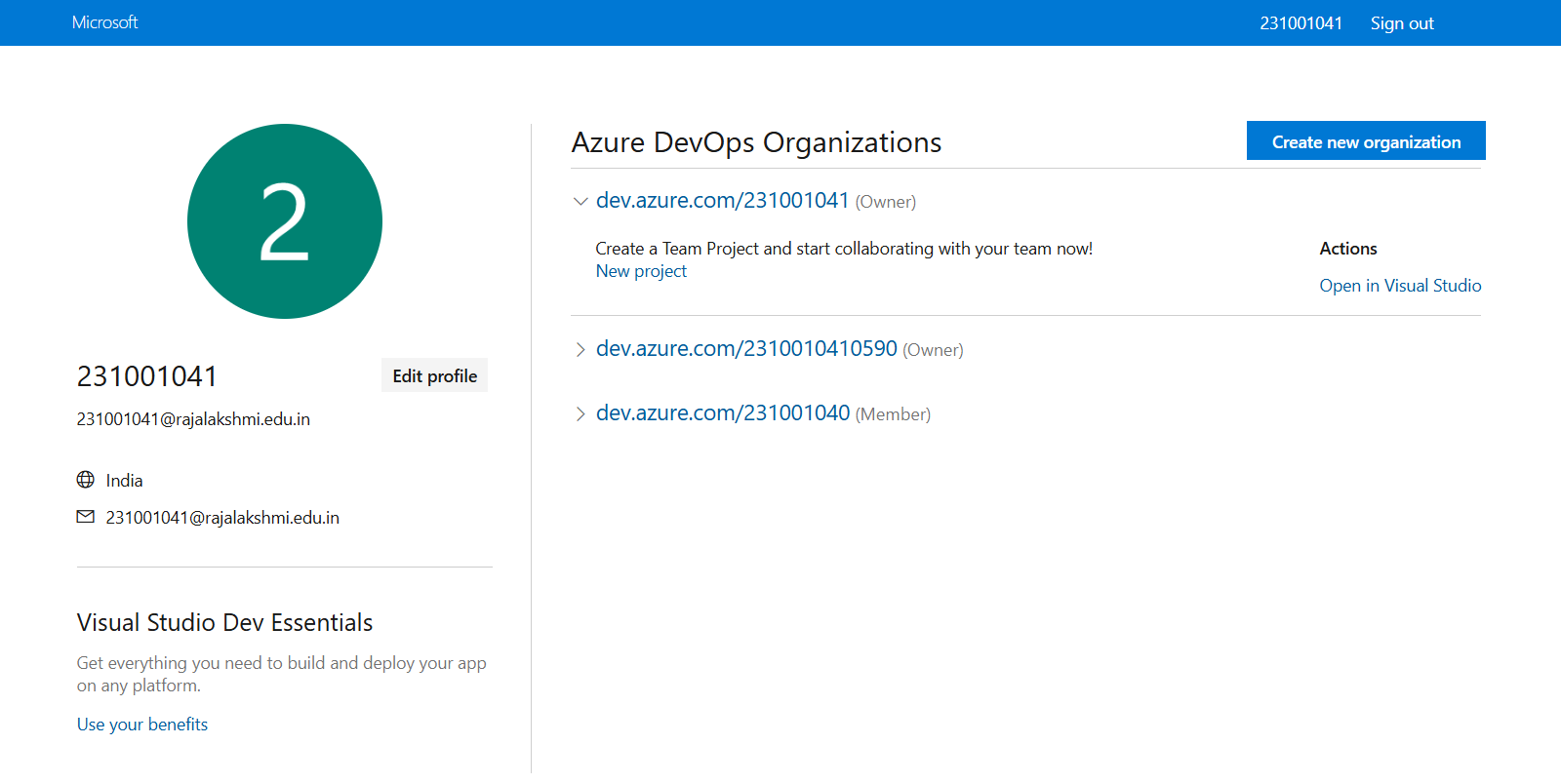
***Name****:*Choose a name for the project(e.g.,LMS).

***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).

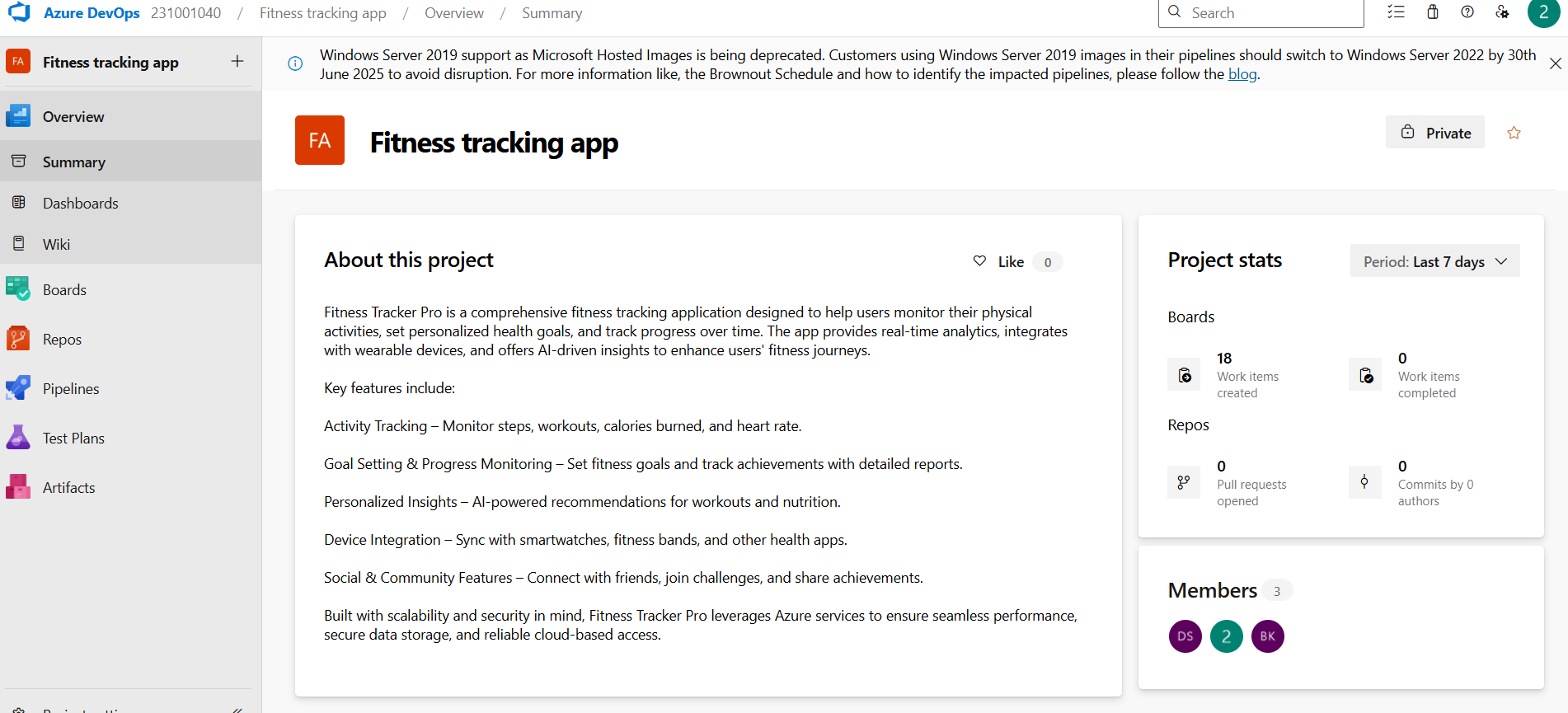
* 1. Once you’ve filled out the details, click **Create** to setup your first project.



1. 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.



1. Project dashboard

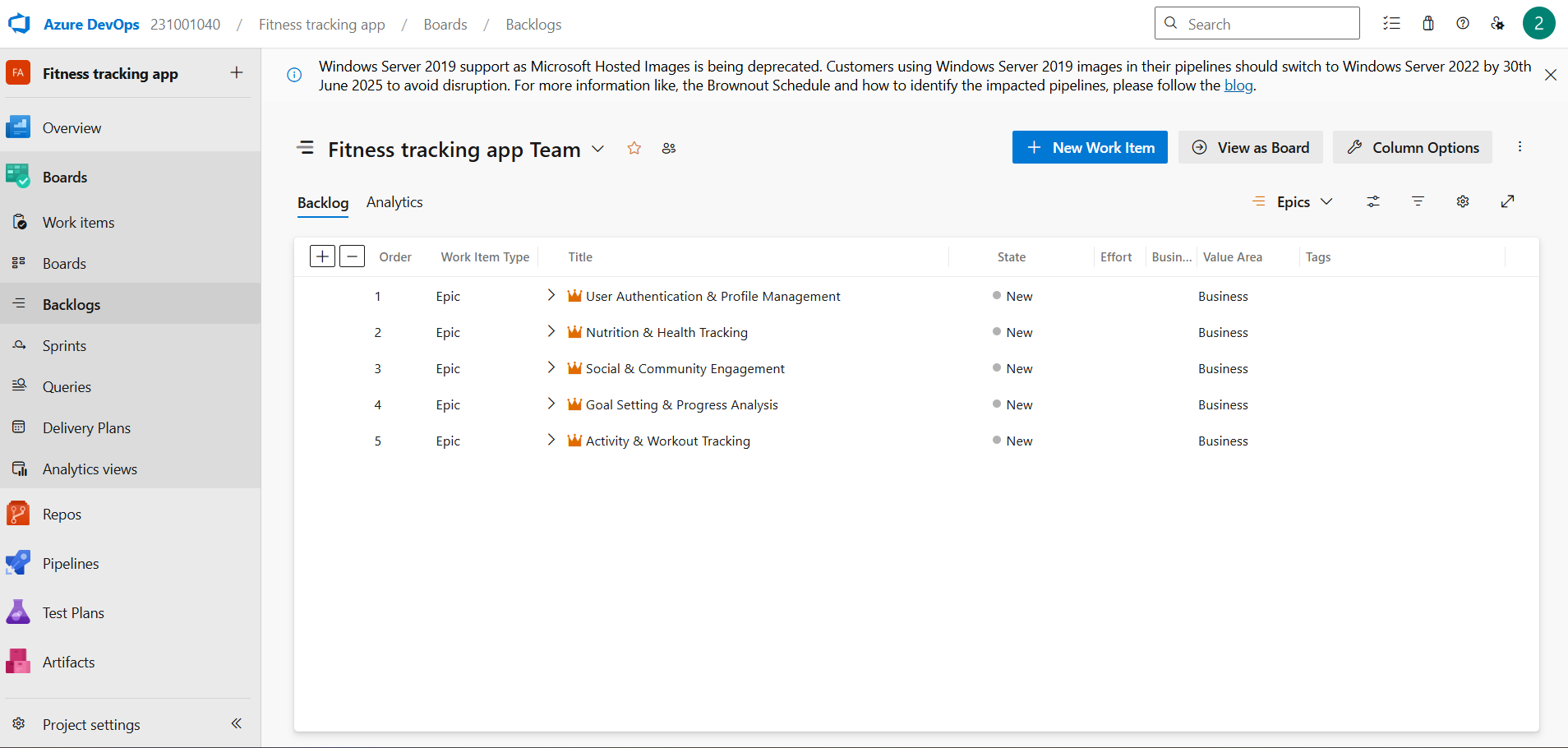


1. To manage user stories:
   1. From the**left-hand navigation menu**,click on **Boards**.This will take you to the main **Boards**

page,where you can manage work items,backlogs,andsprints.

* 1. On the **work items** page,you'll see the option to **Add a work item** a 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** drop down,select **User Story**.This will open a form to enter details for this user story.

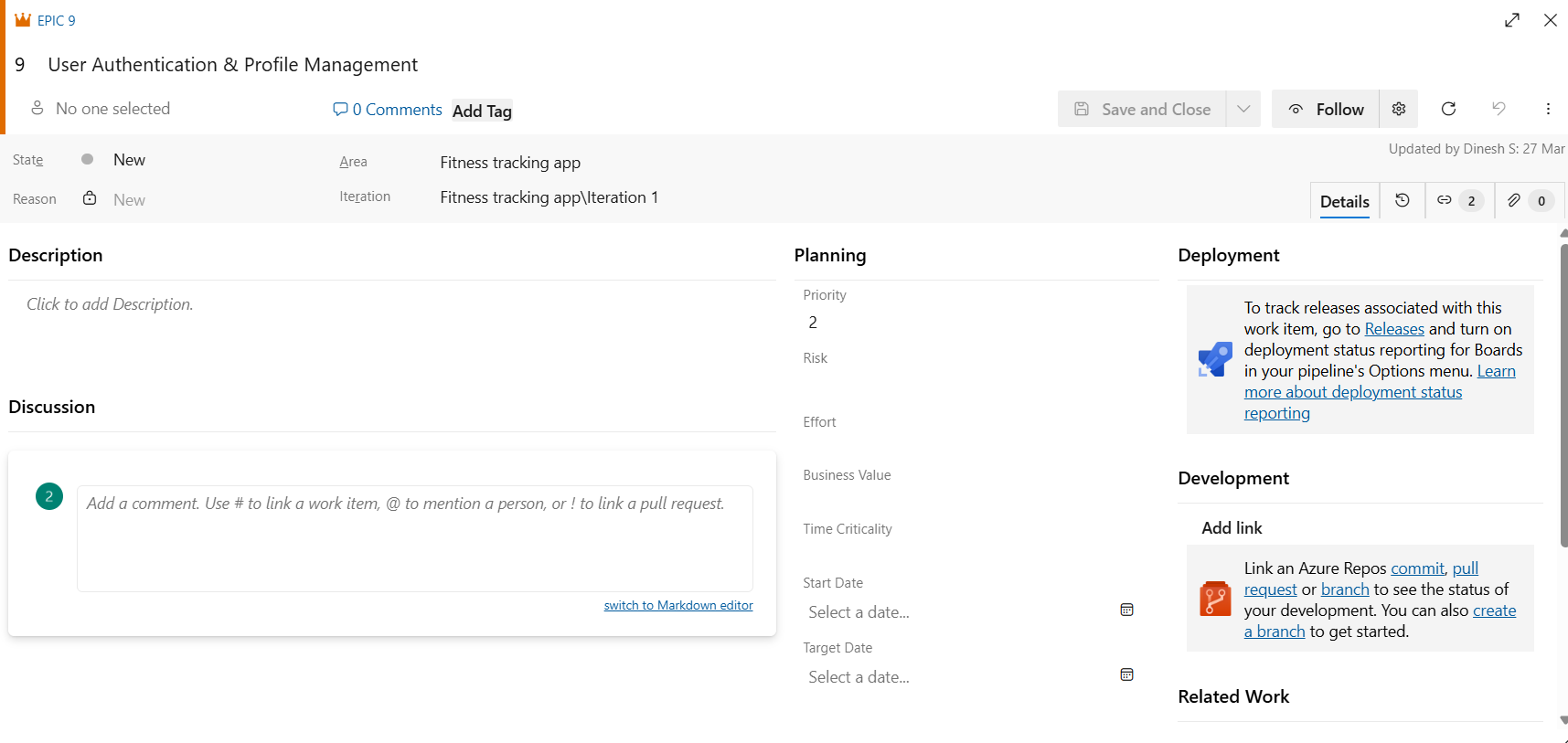


**RESULT:**Successfully created an Azure DevOps project with user story management and agile workflow setup.

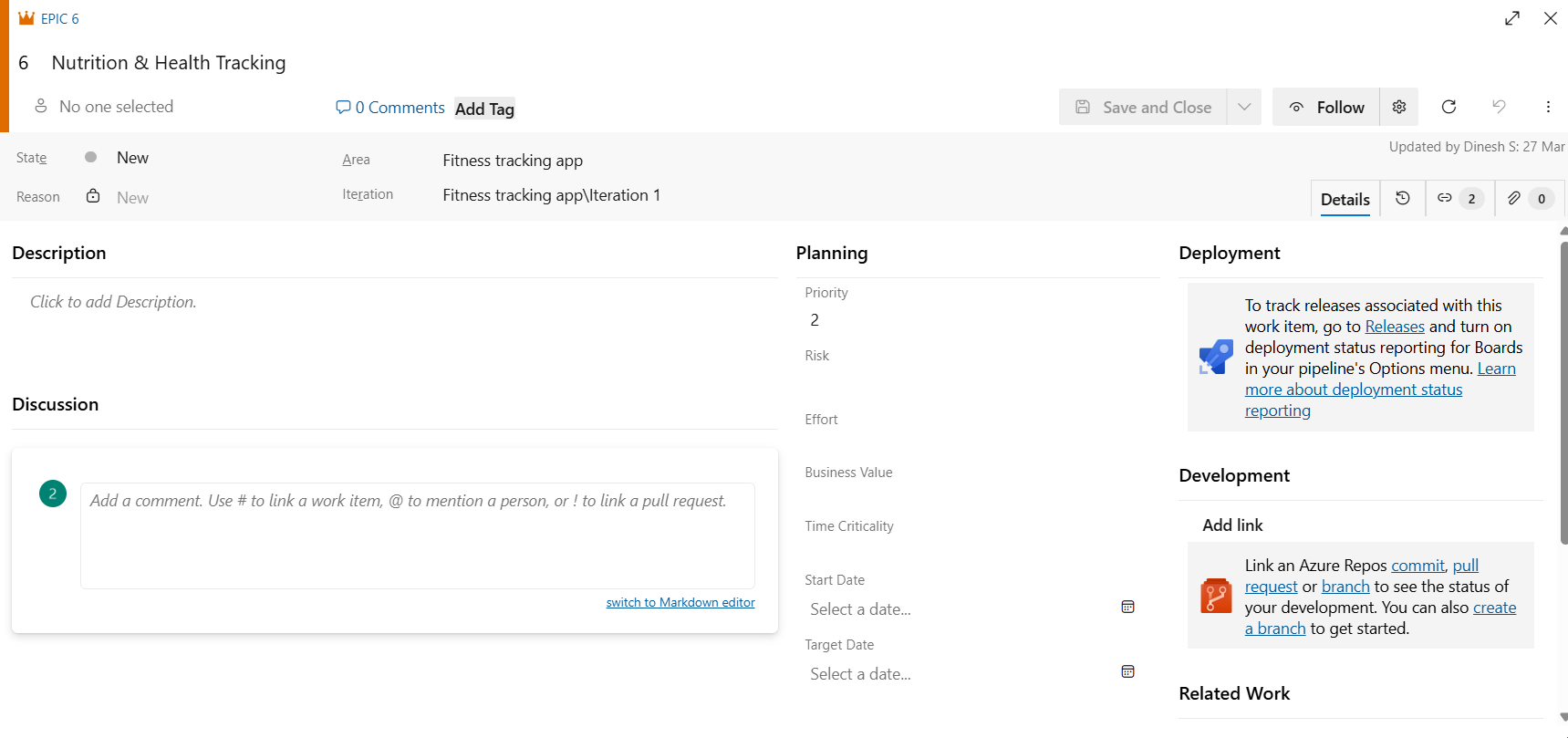
|  |  |
| --- | --- |
| **Exp.No:3** | **SETTING UP EPICS FEATURES AND USER STORIES FOR PROJECT PLANNING** |
| **Date:14/02/2025** |

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

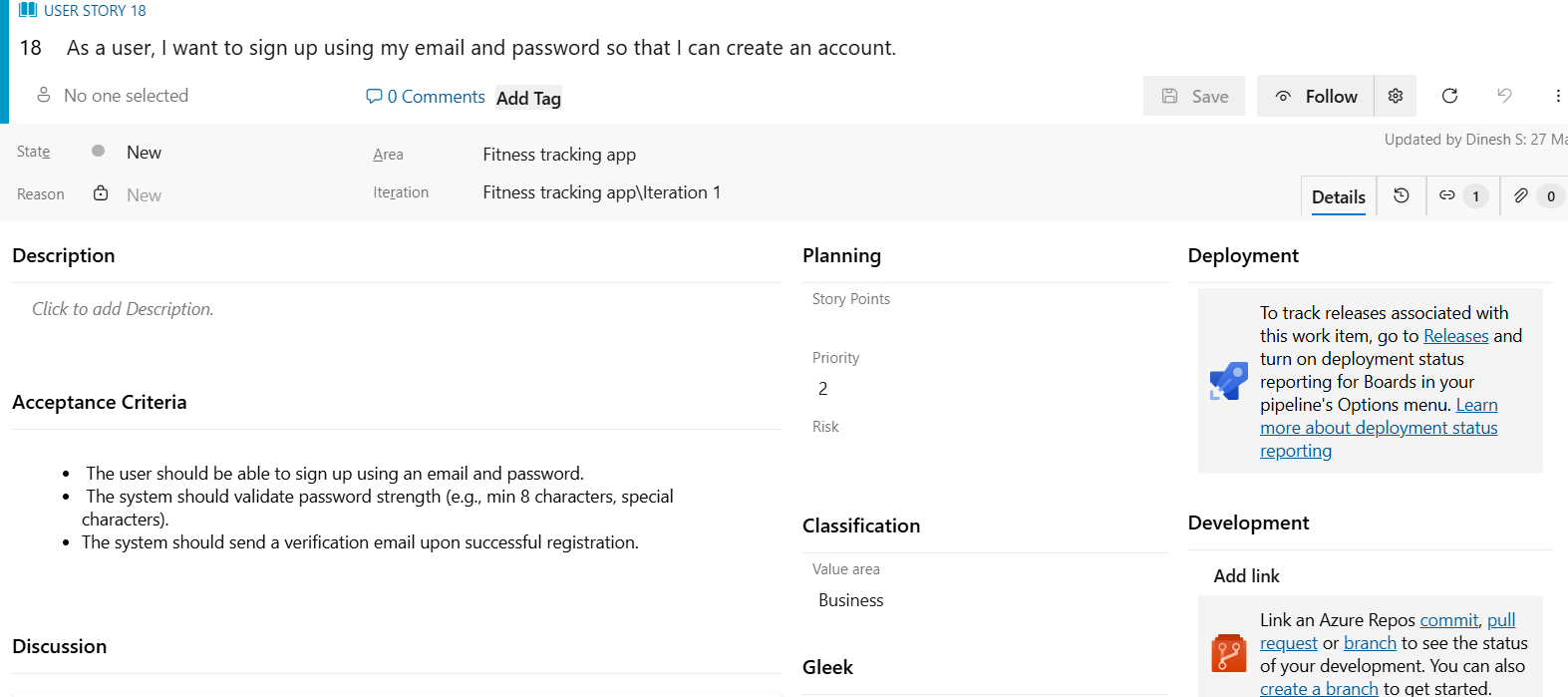
1. **Fill in Epics**

****

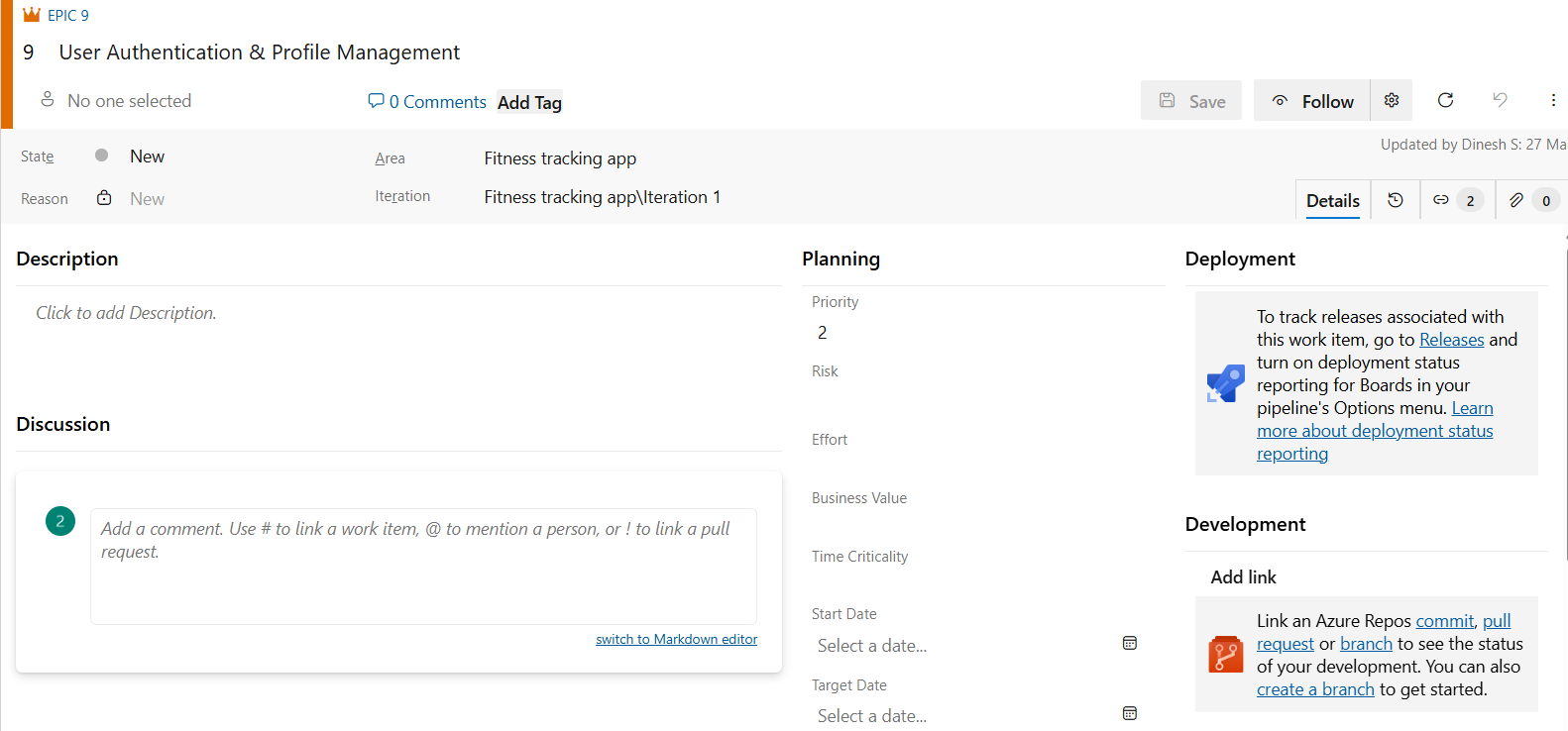
1. **Fill in Features**

****

1. **Fill in User Story Details**

****

1. **Fill in Task Details**

****

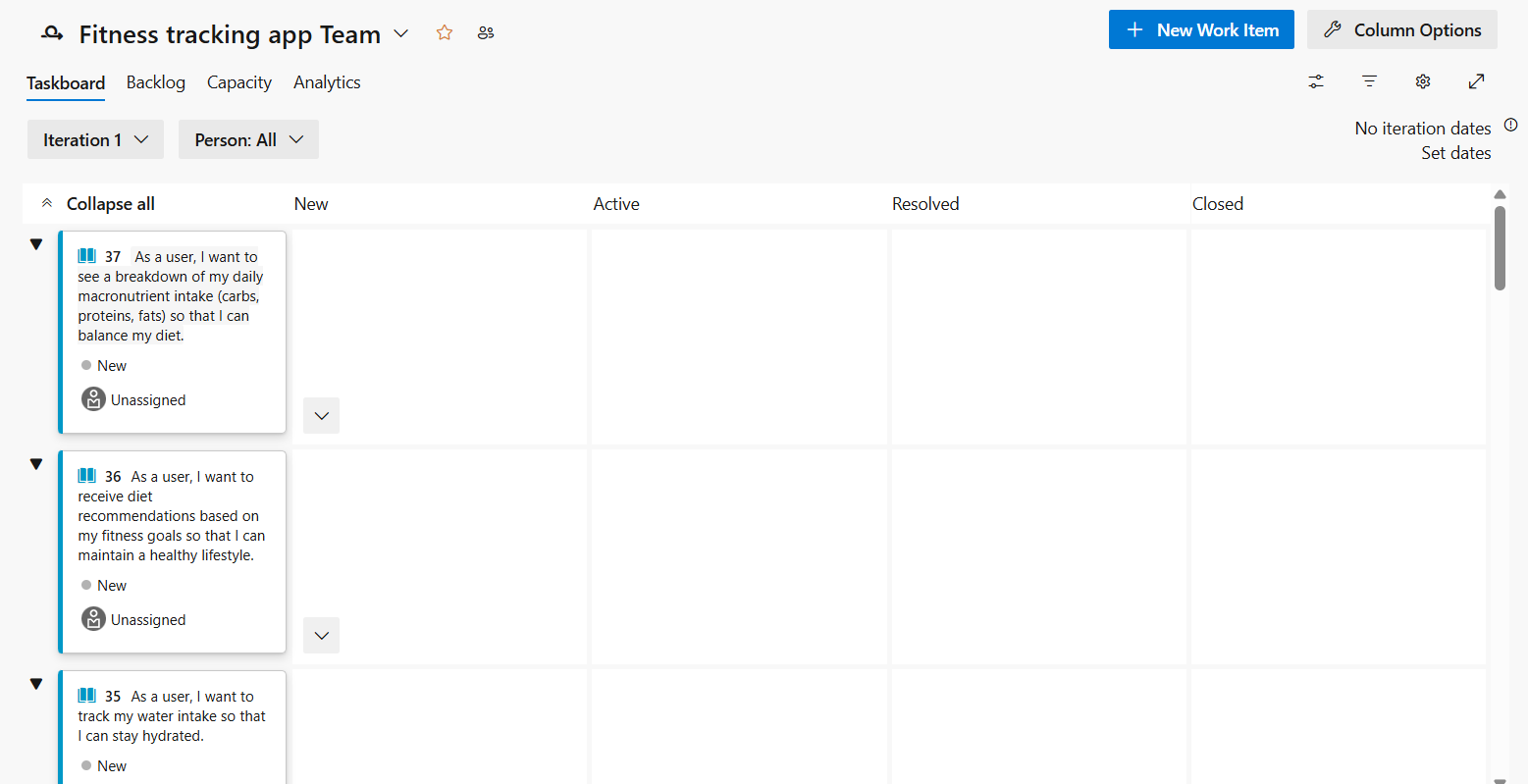
# **RESULT:**Thus, the creation of epics ,features ,user story and task has been created successfully.

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

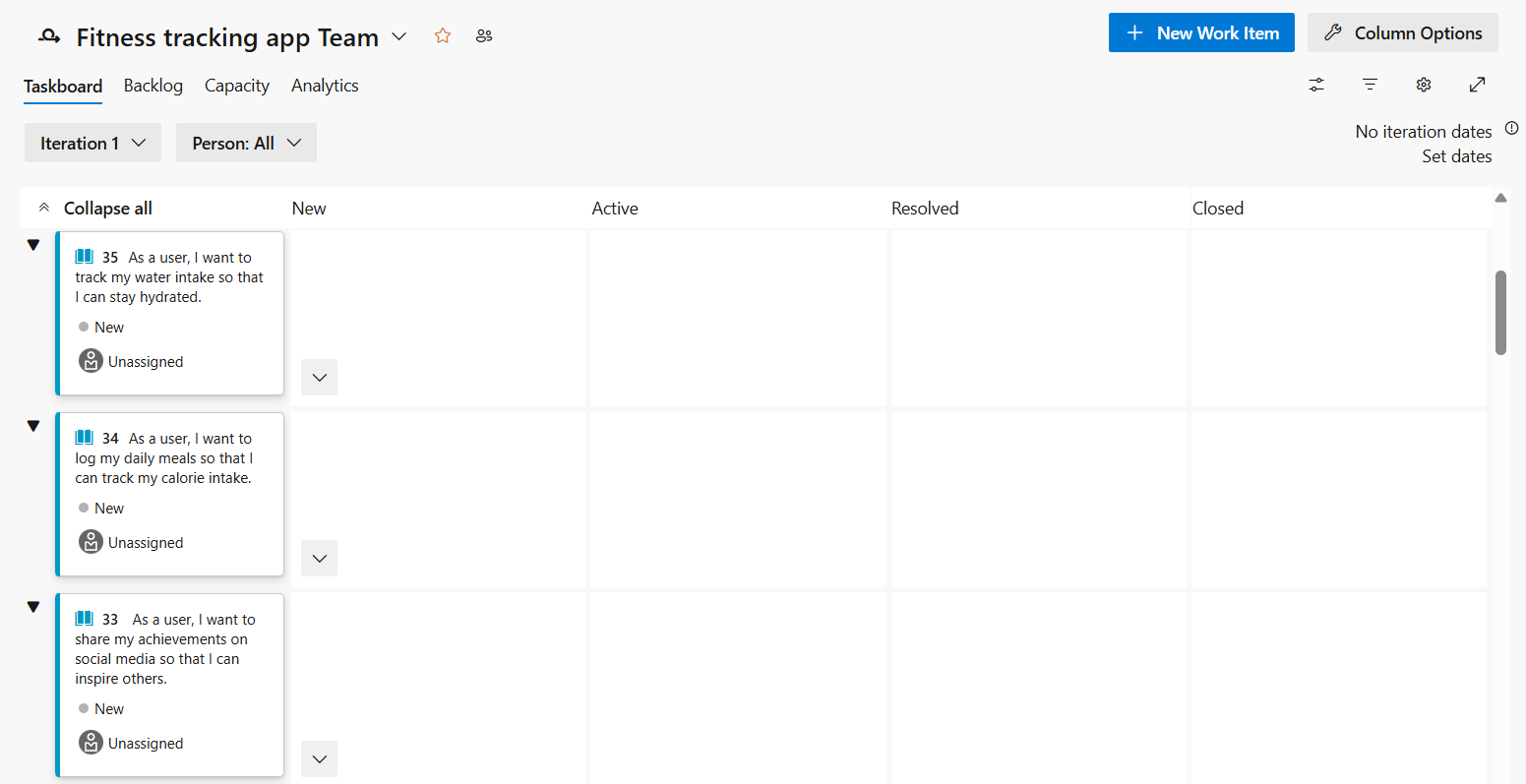
**AIM:**To assign user story to specific sprint for the Fitness tracking app.

## SPRINT PLANNING

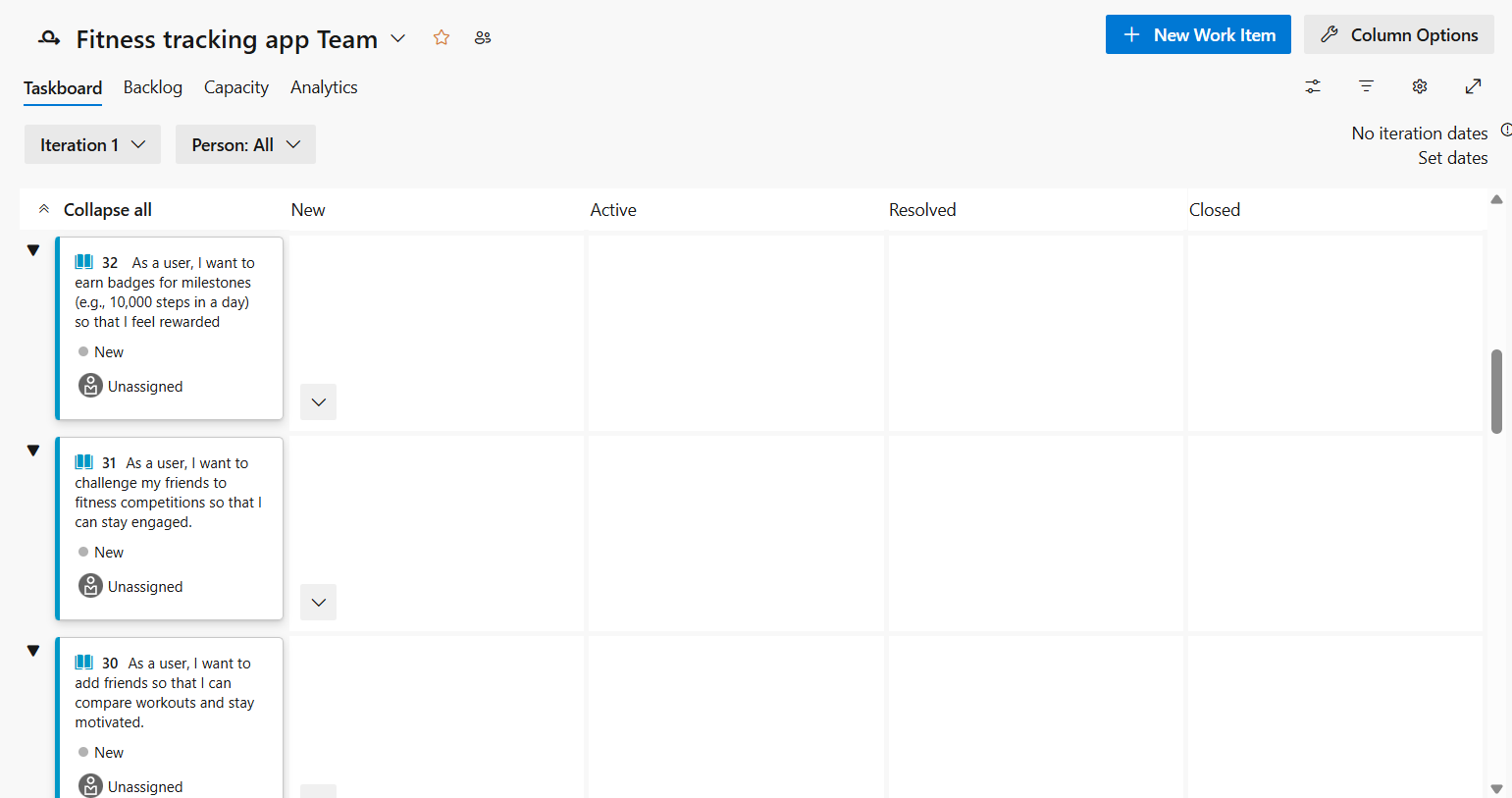
**Sprint 1**

****

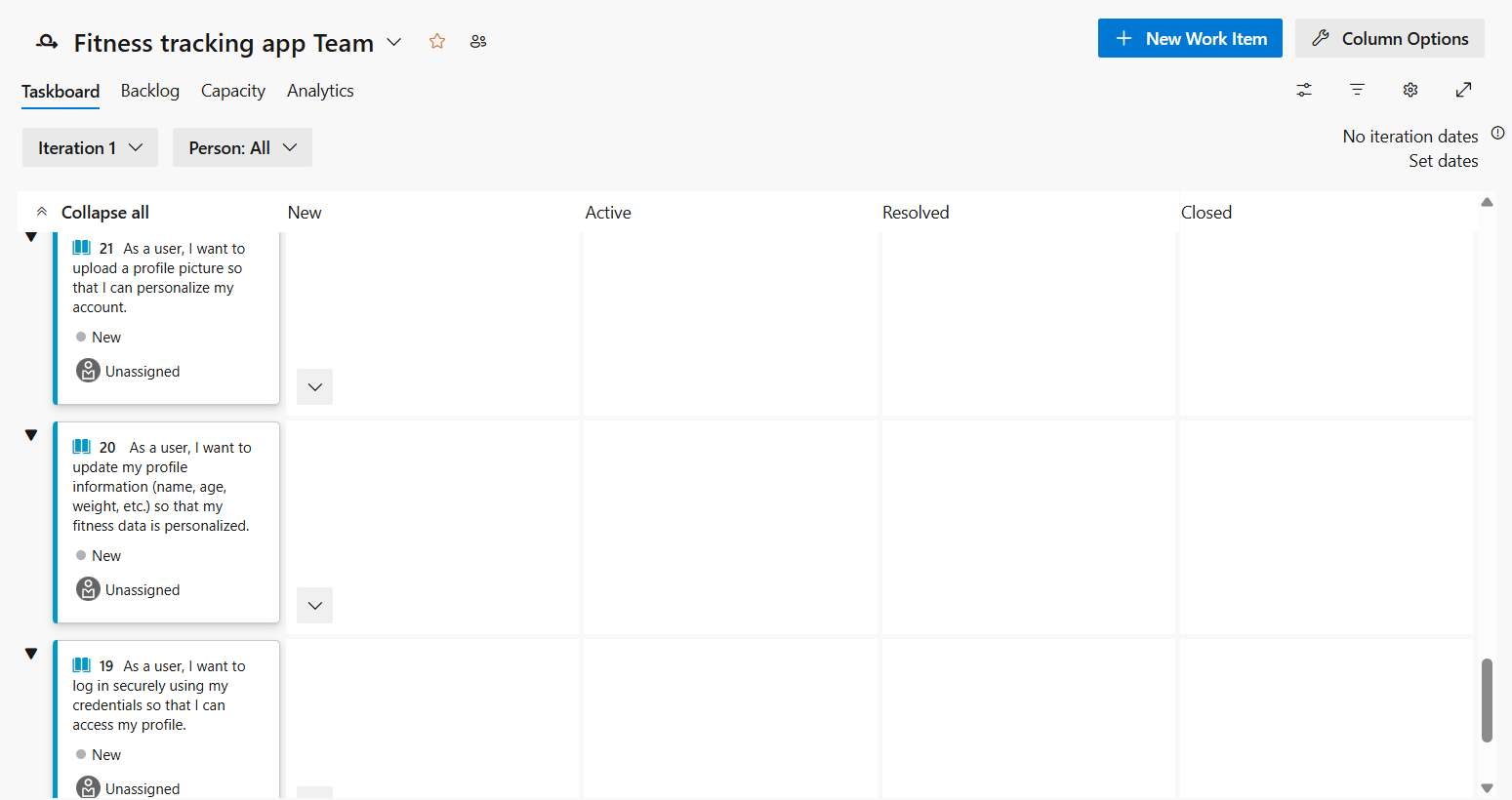
**Sprint 2**

****

**Sprint 3**

****

**Sprint 4**

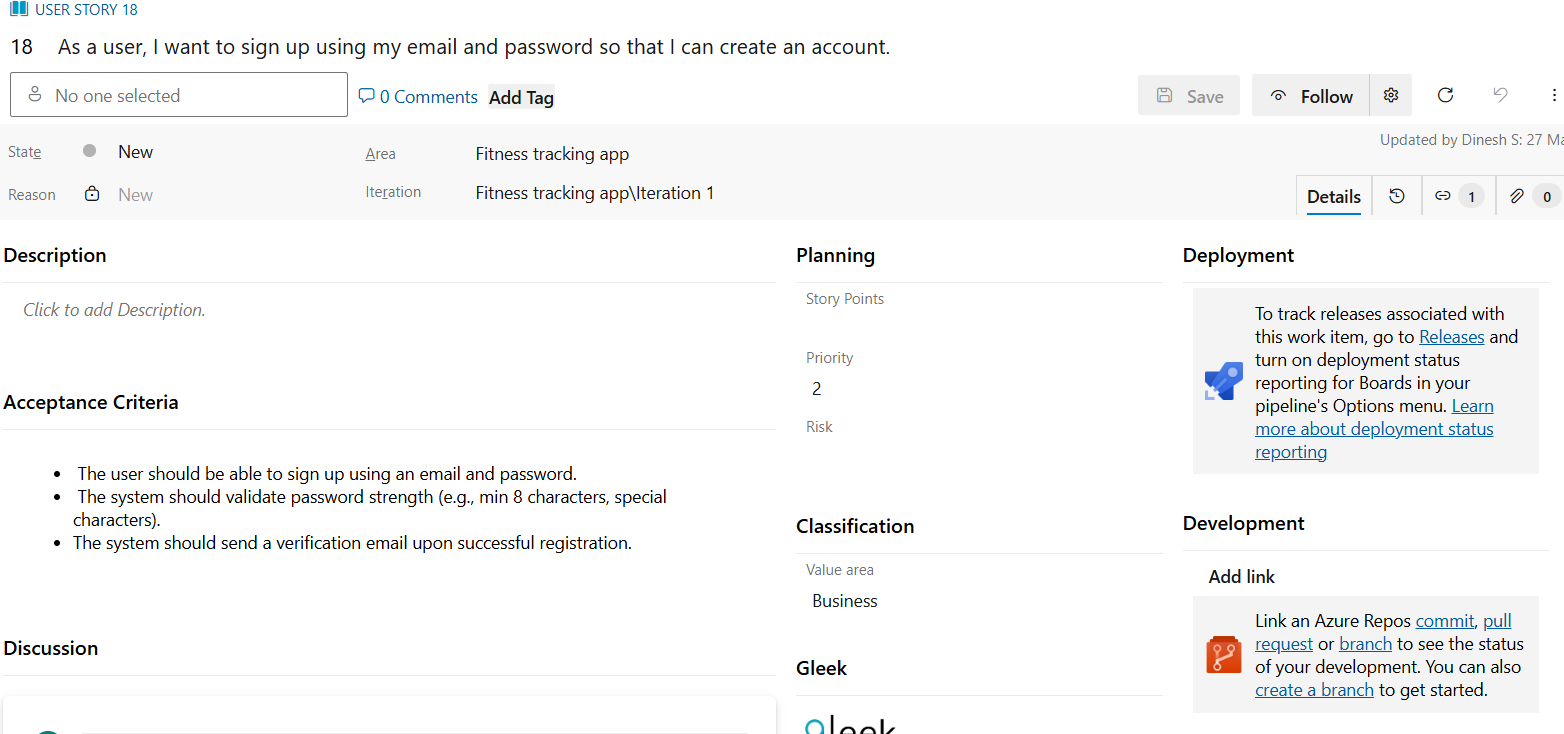
****

**RESULT:**The Sprints are created for the Fitness tracking app.

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

**AIM:**Create Poker Estimation for the user stories-Fitness tracking app Project.

## POKER ESTIMATION

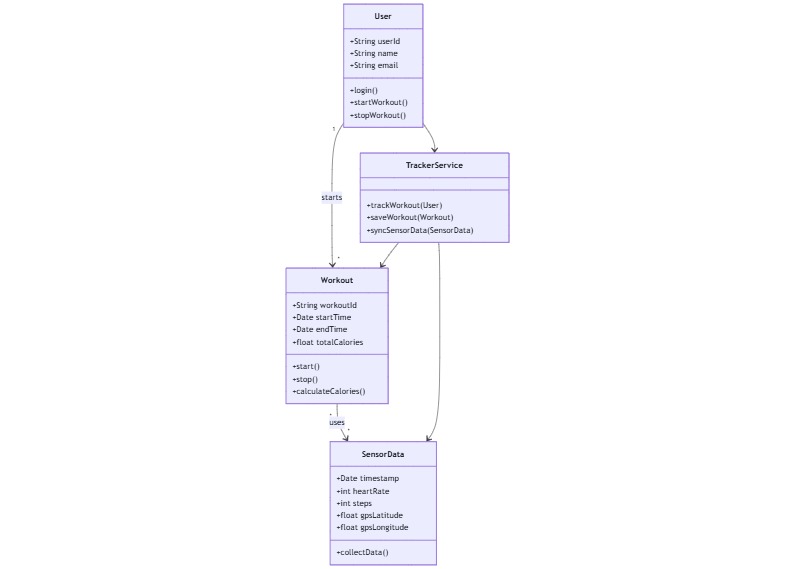
****

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

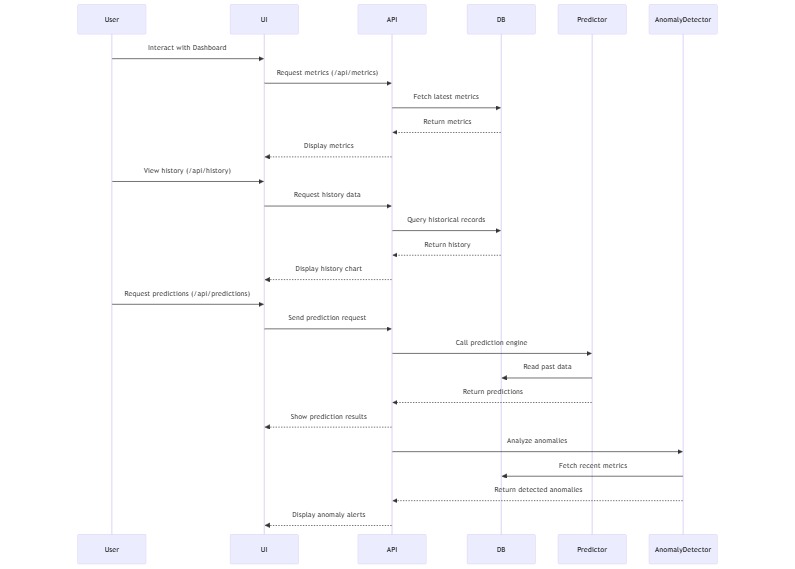
|  |  |
| --- | --- |
| **Exp.No:6** | **DESIGNING CLASS AND SEQUENCE DIAGRAMS FOR PROJECT ARCHITECTURE** |
| **Date: 04/04/2025** |

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

## 6A.CLASS DIAGRAM

****

**6B. SEQUENCE DIAGRAM**

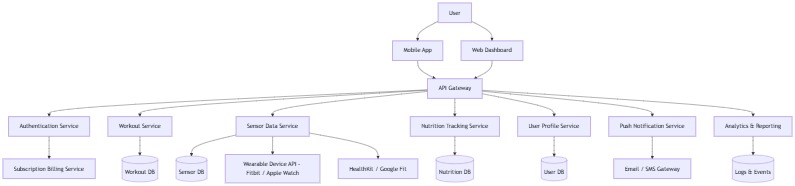
****

**RESULT:** The Class Diagram and Sequence Diagram is designed Successfully for the Fitness tracking app.

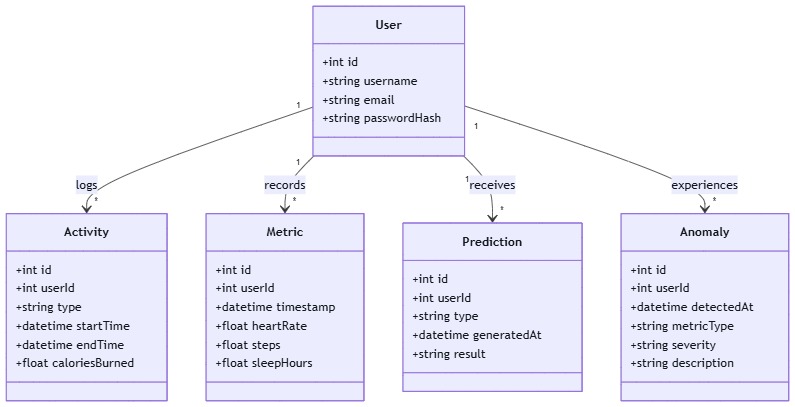
|  |  |
| --- | --- |
| **Exp.No:7** | **DESIGNING ARCHITECTURAL AND ER DIAGRAMS FOR PROJECT STRUCTURE** |
| **Date:16/04/2025** |

**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 Fitness tracking app.

|  |  |
| --- | --- |
| **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 atleast 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 Feature soft he Application
   * User Authentication(Login,Registration,ForgotPassword)
   * Profile Management(View/EditProfile,UploadPicture)
   * Post Feed(Create,View,Edit,DeletePosts)
   * Chat Feature(Real-timeMessaging)
   * Notifications(PostLikes,Comments,Messages)
2. Define User Interactions

Each test case reflects realistic user actions:

* + Logging in and out
  + Editing profile information
  + Creating and managing posts
  + Chatting with another user
  + Receiving notifications for activities

1. Design Happy Path Test Cases
   * Focused on successful flows where inputs are valid and systems behave as expected.
   * Examples:
     + User signs up successfully
     + Create a new post
     + Sends message to another user
     + Views not if I cations fo rnew comments
2. Design Error Path Test Cases
   * Focused on handling in valid ,missing,or unexpected inputs.
   * Examples:
     + Registration fails due to existing email
     + Post creation denied due toempty content
     + Message not sent when user is blocked
     + Unauthorized access to another user’s profile
3. Break Down Steps and Expected Results Each test case includes:
   * Step-by-step actions
   * Input values
   * Expected results

This ensures reproducibility and easy automation later.

1. Use Clear Naming and IDs
   * Example:
     + TC01–RegisterwithEmail&Password
     + TC02–RegisterwithSocialMedia
     + TC03–VerifyEmailafterRegistration
   * Helps trace test cases to users tories and features in AzureDevOps
2. Separate Test Suites

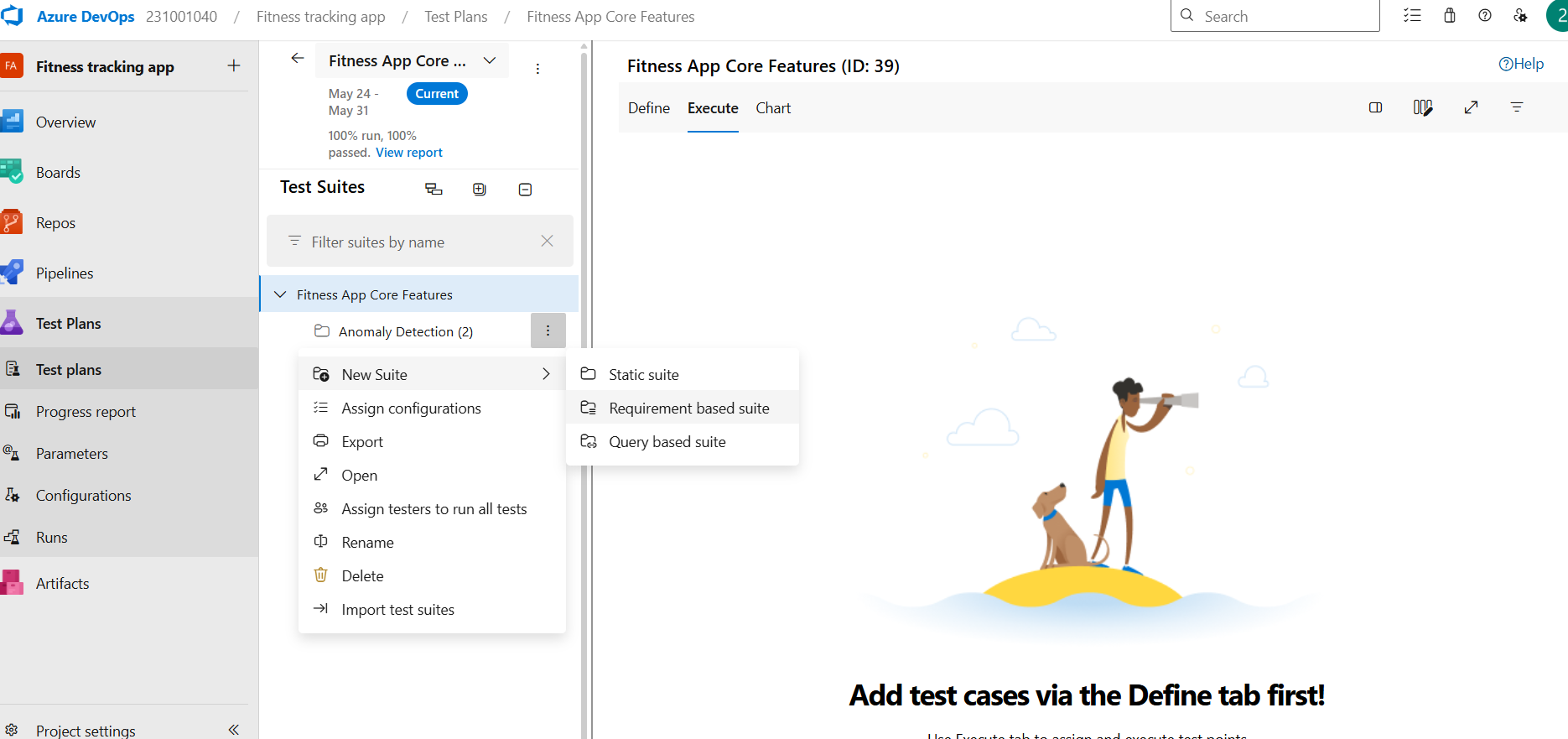
Organize test cases based on functional modules:

* + TS01–Register with Email&Password
  + TS02–Register with Social Media
  + TS03–Verify Email after Registration
  + TS04–Login with Email & Password
  + TS05–Login with Social Media

1. **Newtestplan**

****

1. **Testsuite**

****

1. **Testcase**

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

Social Network Platform–Test Plans

**User Stories– Fitness tracking app**

## USER STORIES – Fitness Tracking Application

1. **US01:** As a user, I want to sign up using my email and password, so I can create an account securely.
2. **US02:** As a user, I want to log my daily steps, so I can track my physical activity.
3. **US03:** As a user, I want to set daily fitness goals, so I can stay motivated.
4. **US04:** As a user, I want to view my weekly activity summary, so I can track my progress.
5. **US05:** As a user, I want to receive reminders to log activity, so I don’t forget.

.

## TESTSUITE:TS01–USERLOGIN(ID: 113)

### TestCase:TC01–RegisterwithValidEmail&Password

**UserStory**:Asauser,Iwanttosignupusingmyemailandpassword,sothatIcancreateanaccount securely.

**TestType:**HappyPath

### StepsandExpectedResults:

* + **Action:**Navigatetotheregistrationpage

**ExpectedResult:**Registrationpagewithemailandpasswordfieldsisdisplayed

* + **Action:**Enteravalidemailandstrongpassword

**ExpectedResult:**Inputisacceptedwithnovalidationerrors

* + **Action:**Clickonthe"Register"button

**ExpectedResult:**Useristakentotheemailverificationscreen

### Notes:

Ensuretheemailisnotalreadyregistered.Simulatessuccessfulemail/passwordregistration.

### TestCase:TC02–RegisterwithAlreadyRegisteredEmail

**UserStory:**Asauser,Iwanttosignupusingmyemailandpassword,sothatIcancreateanaccount securely.

**TestType:**ErrorPath

### StepsandExpectedResults:

* + **Action:**Navigatetotheregistrationpage

**ExpectedResult:**Registrationformisdisplayed

* + **Action:**Enteranemailthatisalreadyregistered

**ExpectedResult:**Emailfieldishighlightedorflagged

* + **Action:**Clickonthe"Register"button

**ExpectedResult:**Errormessage“Emailalreadyregistered”isshown

### Notes:

Ensuresthesystempreventsduplicateaccounts.

### TEST SUITE: Register with Social Media (ID: 112) TestCase:TC03–RegisterwithValidGoogleAccount

**UserStory:**Asauser,IwanttosignupusingmyGoogleorFacebookaccount,sothatIcancreatean account quickly.

**TestType:**HappyPath

### StepsandExpectedResults:

* + **Action:**Navigatetotheregistrationpage

**ExpectedResult:**RegistrationoptionsforGoogleandFacebookaredisplayed

* + **Action:**Clickon“SignupwithGoogle”

**ExpectedResult:**GoogleOAuthloginscreenappears

* + **Action:**ChooseavalidGoogleaccountandgrantpermission

**ExpectedResult:**Userisredirectedbackandaccountiscreated

* + **Action:**Confirmsuccessfulregistration

**ExpectedResult:**Useristakentothewelcomescreenoremailverificationpage(ifrequired)

### Notes:

EnsureOAuthcredentialsareproperlyconfigured.Testaccountmustnotalreadyexistinthesystem.

### TestCase:TC04–RegisterwithDeniedOAuthPermission(Google/Facebook)

**UserStory:**Asauser,IwanttosignupusingmyGoogleorFacebookaccount,sothatIcancreatean account quickly.

**TestType:**ErrorPath

### StepsandExpectedResults:

* + **Action:**Navigatetotheregistrationpage

**ExpectedResult:**Socialmediasign-upoptionsareshown

* + **Action:**Clickon“SignupwithGoogle”or“SignupwithFacebook”

**ExpectedResult:**OAuthpermissionscreenisdisplayed

* + **Action:**Click“Cancel”ordenypermission

**ExpectedResult:**Registrationprocessishalted

### Notes:

SystemshouldhandleOAuthdenialgracefullyandallowusertoretryorchooseanothermethod.

### TESTSUITE:TS03-VerifyEmailafterRegistration(ID:118) Test Case: TC05 – Verify Email After Successful Registration

**UserStory:**Asauser,Iwanttoverifymyemailafterregistration,sothatIcanactivatemyaccount.

**TestType:**HappyPath

### StepsandExpectedResults:

* + **Action:**Completetheregistrationprocessusingemailandpassword

**ExpectedResult:**Registrationissuccessful,andaverificationemailissent

* + **Action:**Opentheemailinboxandclicktheverificationlink

**ExpectedResult:**Userisredirectedtotheverificationconfirmationpage

* + **Action:**Attempttologinafterverification

**ExpectedResult:**Loginissuccessful,anduserisredirectedtothedashboard

### Notes:

Ensuretheverificationlinkisuniqueandonlyusableonce.Simulatesatypicalsuccessfulverification flow.

### TestCase:TC06–EmailVerificationwithExpiredorInvalidLink

**UserStory:**Asauser,Iwanttoverifymyemailafterregistration,sothatIcanactivatemyaccount.

**TestType:**ErrorPath

### StepsandExpectedResults:

* + **Action:**Completetheregistrationandreceivetheverificationemail

**ExpectedResult:**Emailwithverificationlinkisreceived

* + **Action:**Waitforthelinktoexpireormodifythelinkmanually

**ExpectedResult:**Theverificationlinkbecomesinvalid

* + **Action:**Clicktheexpiredorinvalidverificationlink

**ExpectedResult:**Errormessage“Verificationlinkisinvalidorexpired”isshown

### Notes:

Systemshouldprompttheusertorequestanewverificationlinkandnotactivatetheaccountuntil successful verification.

### TESTSUITE:TS04-LoginwithEmail&Password(ID:121) Test Case: TC07 – Login with Valid Email and Password

**UserStory:**Asauser,Iwanttologinusingmyregisteredemailandpassword,sothatIcanaccessmy account.

**TestType**:HappyPath

### StepsandExpectedResults:

* + **Action**:Navigatetotheloginpage

**ExpectedResult:**Loginpageisdisplayedwithemailandpasswordfields

* + **Action:**Enteravalidregisteredemailandcorrectpassword

**ExpectedResult:**Inputisacceptedwithnovalidationerrors

* + **Action**:Clickthe"Login"button

**ExpectedResult:**Userisredirectedtotheaccountdashboard

### Notes:

Ensurethetestaccountisverifiedandactivebeforetesting.Thissimulatesasuccessfullogin.

### TestCase:TC08–LoginwithInvalidPassword

**UserStory:**Asauser,Iwanttologinusingmyregisteredemailandpassword,sothatIcanaccessmy account.

**TestType:**ErrorPath

### StepsandExpectedResults:

* + **Action:**Navigatetotheloginpage

**ExpectedResult:**Loginpageisdisplayed

* + **Action:**Enteravalidregisteredemailandincorrectpassword

**ExpectedResult:**Passwordisflaggedasincorrect

* + **Action:**Clickthe"Login"button

**ExpectedResult:**Errormessage“Incorrectpassword”isdisplayed,anduserremainsonlogin page

### Notes:

SystemshouldlimitrepeatedfailedattemptsandmaypromptCAPTCHAorlockoutmechanismsafter multiple failures.

### TESTSUITE:TS05-LoginwithSocialMedia(ID:124) Test Case: TC09 – Login with Valid Google Account

**UserStory:**Asauser,IwanttologinusingmyGoogleorFacebookaccount,sothatIdon’tneedto remember my password.

**TestType:**HappyPath

### StepsandExpectedResults:

* + **Action:**Navigatetotheloginpage

**ExpectedResult:**LoginpagedisplaysoptionstologinwithGoogleandFacebook

* + **Action:**Clickthe“LoginwithGoogle”button

**ExpectedResult:**GoogleOAuthscreenappears

* + **Action:**SelectavalidGoogleaccountandauthorizeaccess

**ExpectedResult:**Userisredirectedbackandloggedintotheiraccountdashboard

### Notes:

EnsuretheGoogleaccountisalreadylinkedormatchesanexistinguserprofile.OAuthintegrationmust be correctly configured.

### TestCase:TC10–LoginwithDeniedOAuthPermission(Google/Facebook)

**UserStory:**Asauser,IwanttologinusingmyGoogleorFacebookaccount,sothatIdon’tneedto remember my password.

**TestType:**ErrorPath

### StepsandExpectedResults:

* + **Action:**Navigatetotheloginpage

**ExpectedResult:**Socialloginoptionsarevisible

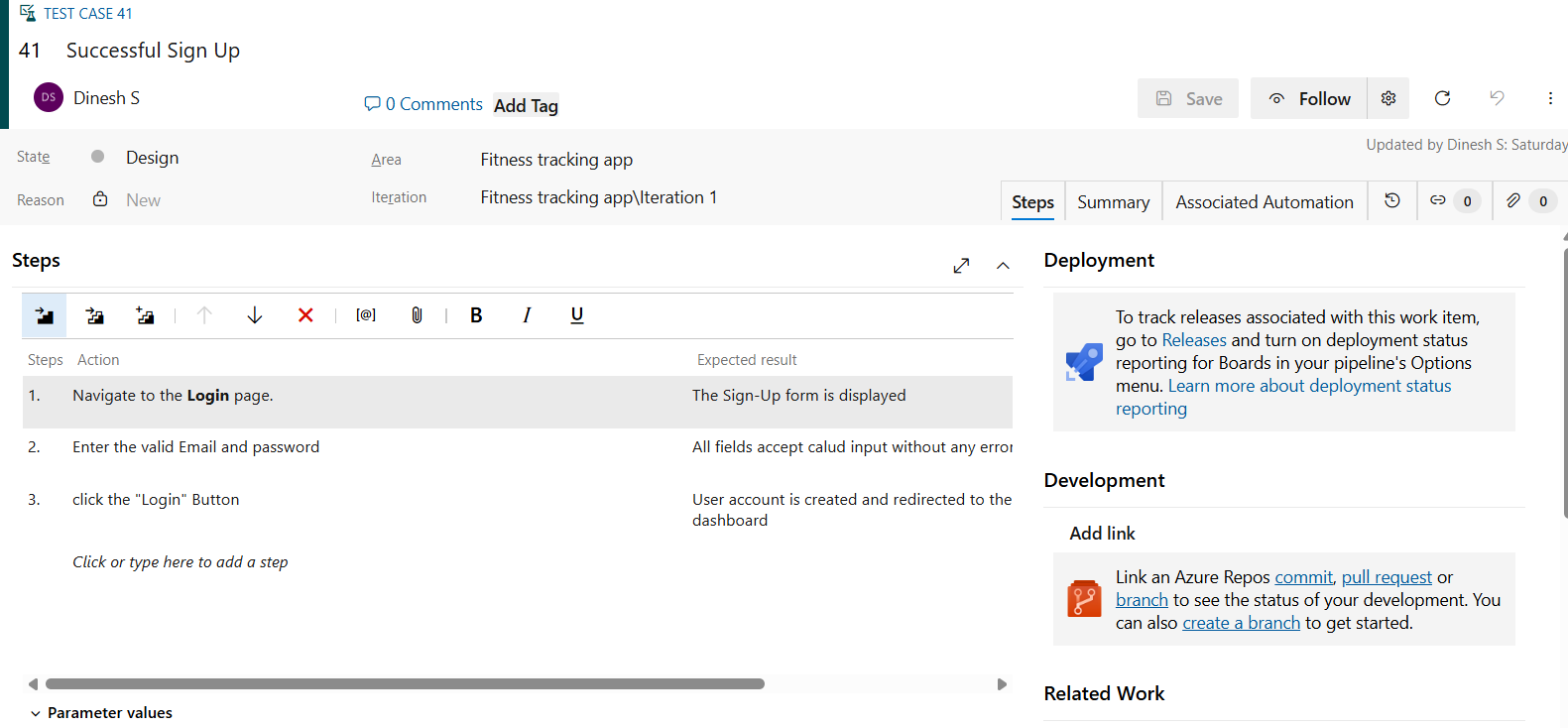
* + **Action:**Click“LoginwithGoogle”or“LoginwithFacebook”

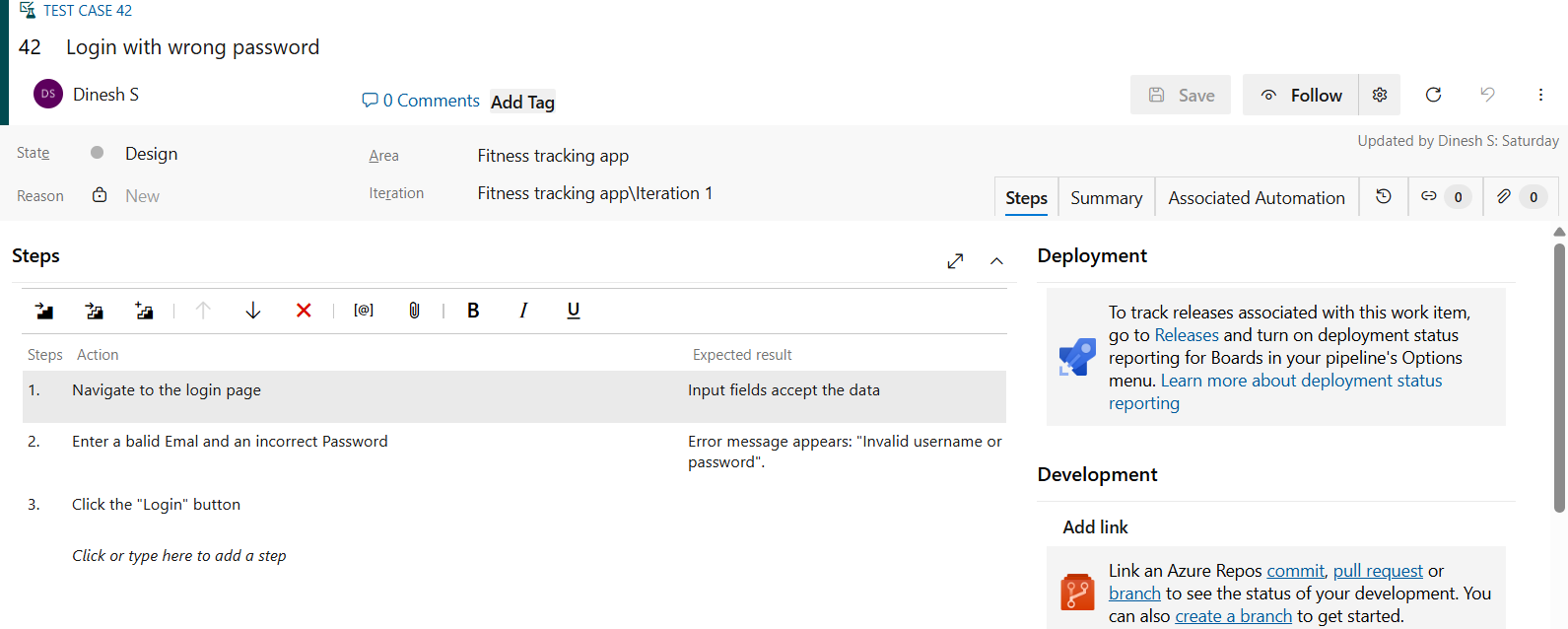
**ExpectedResult:**OAuthconsentscreenappears

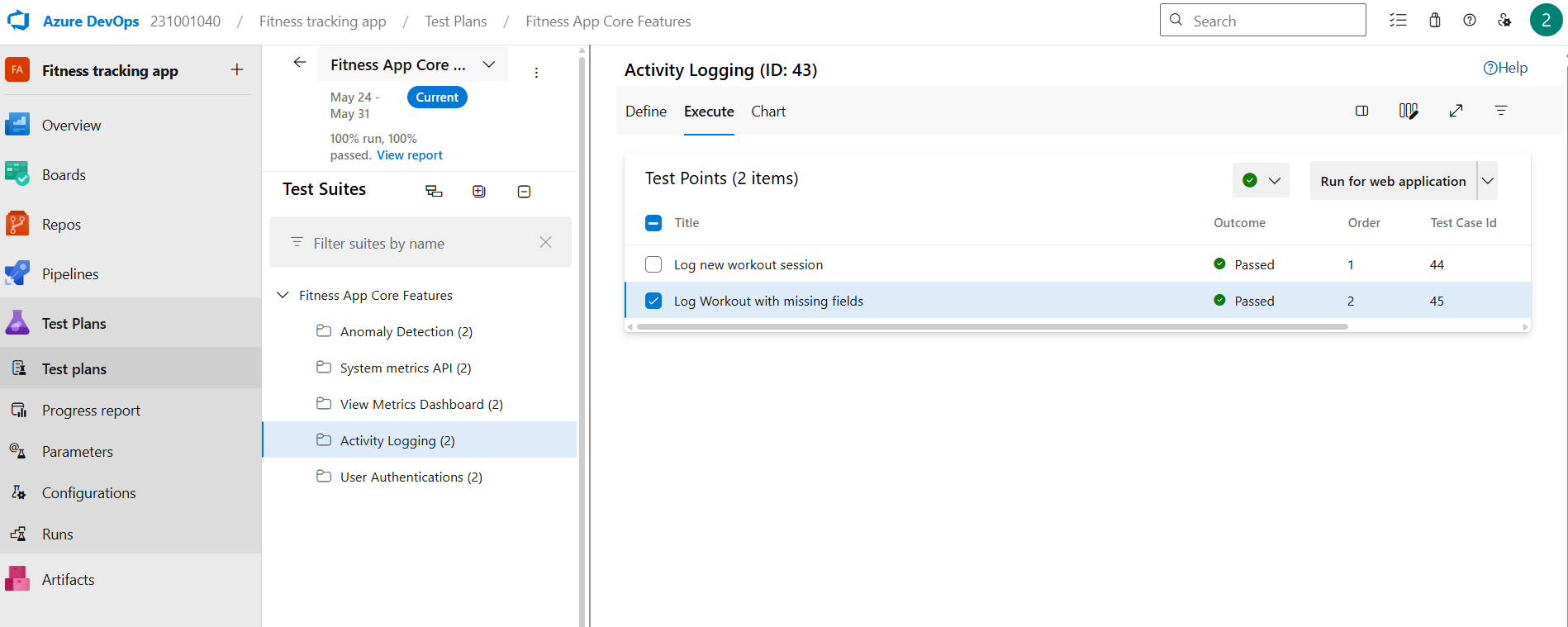
* + **Action:**Click“Cancel”ordenyauthorization

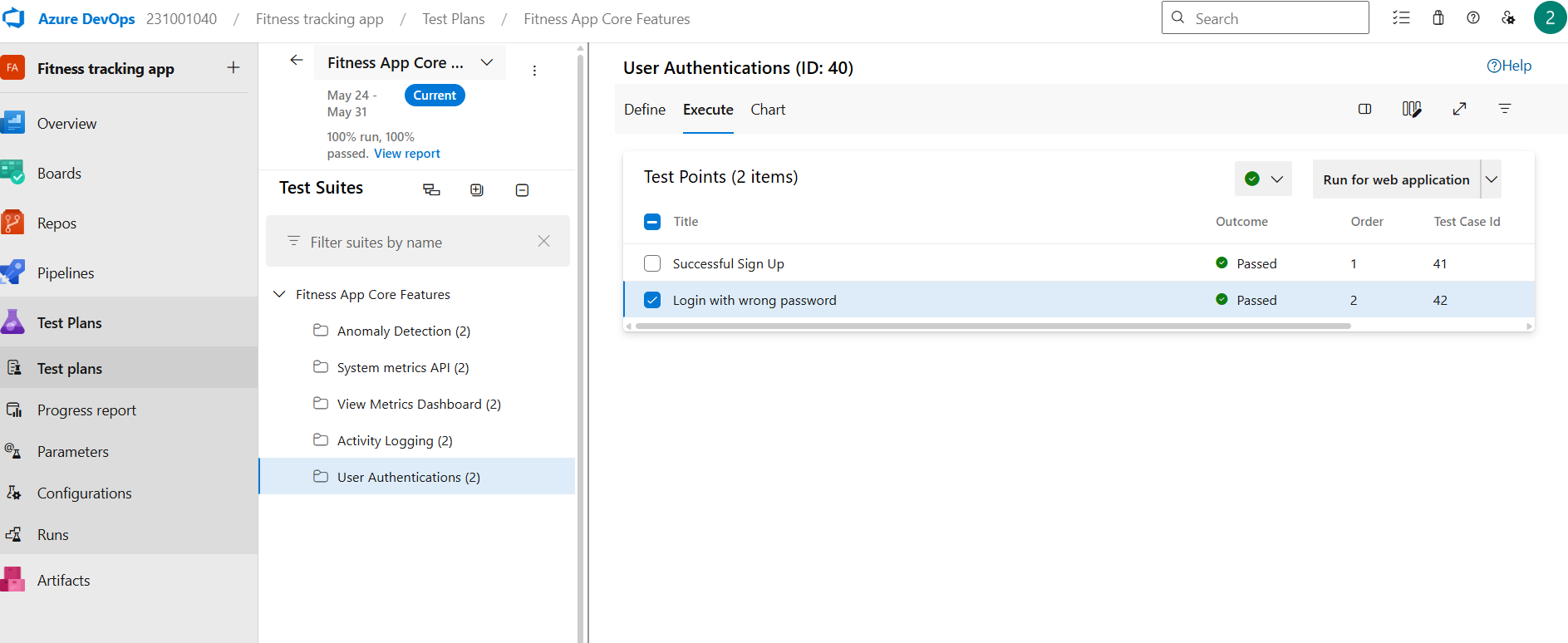
**ExpectedResult:**Userisredirectedbackwitherrormessage“Logincancelledorpermission denied”

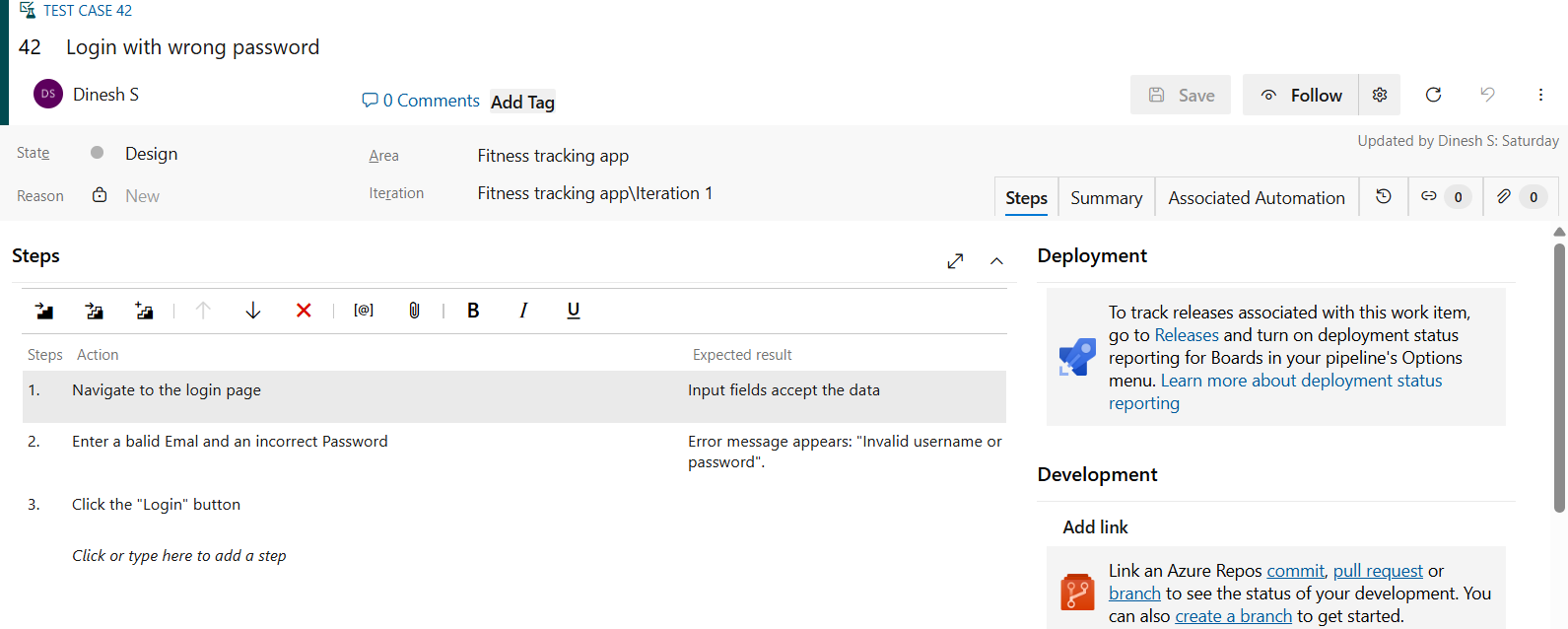
### Test Cases

****

****

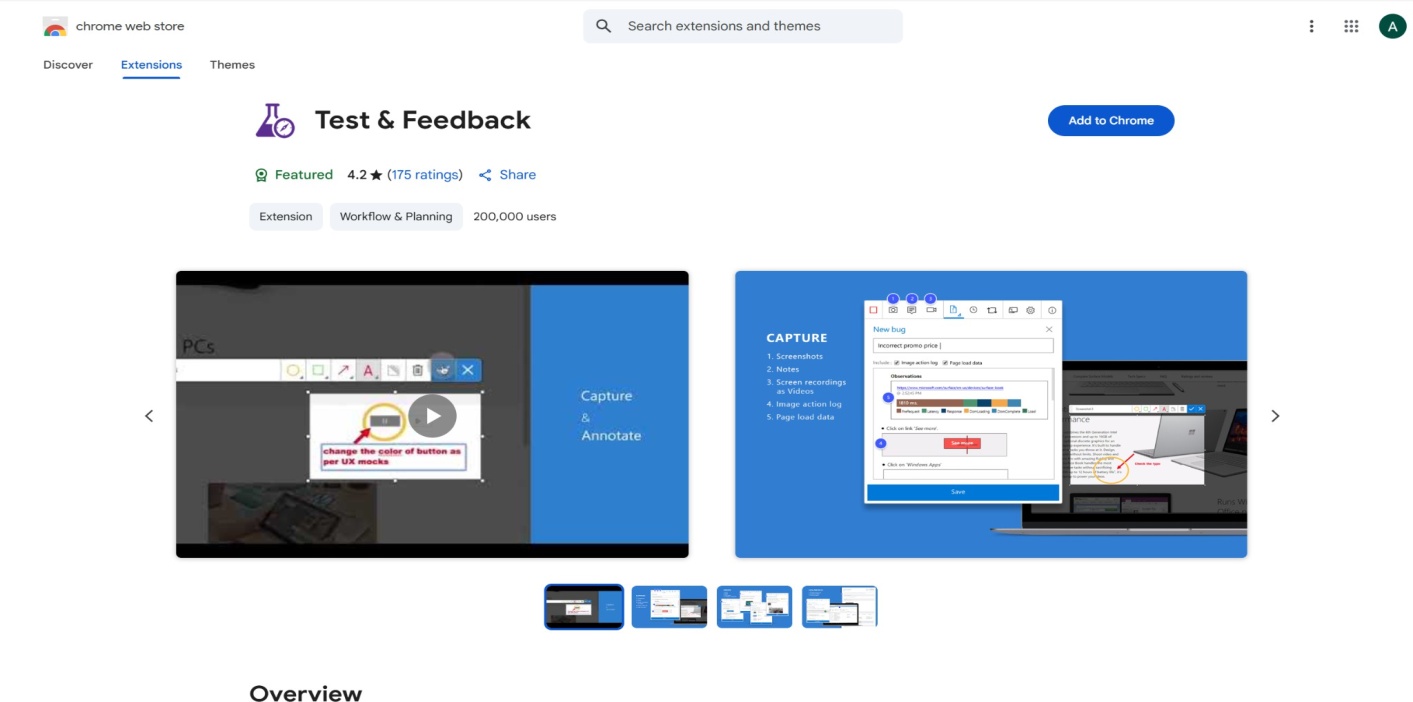


****

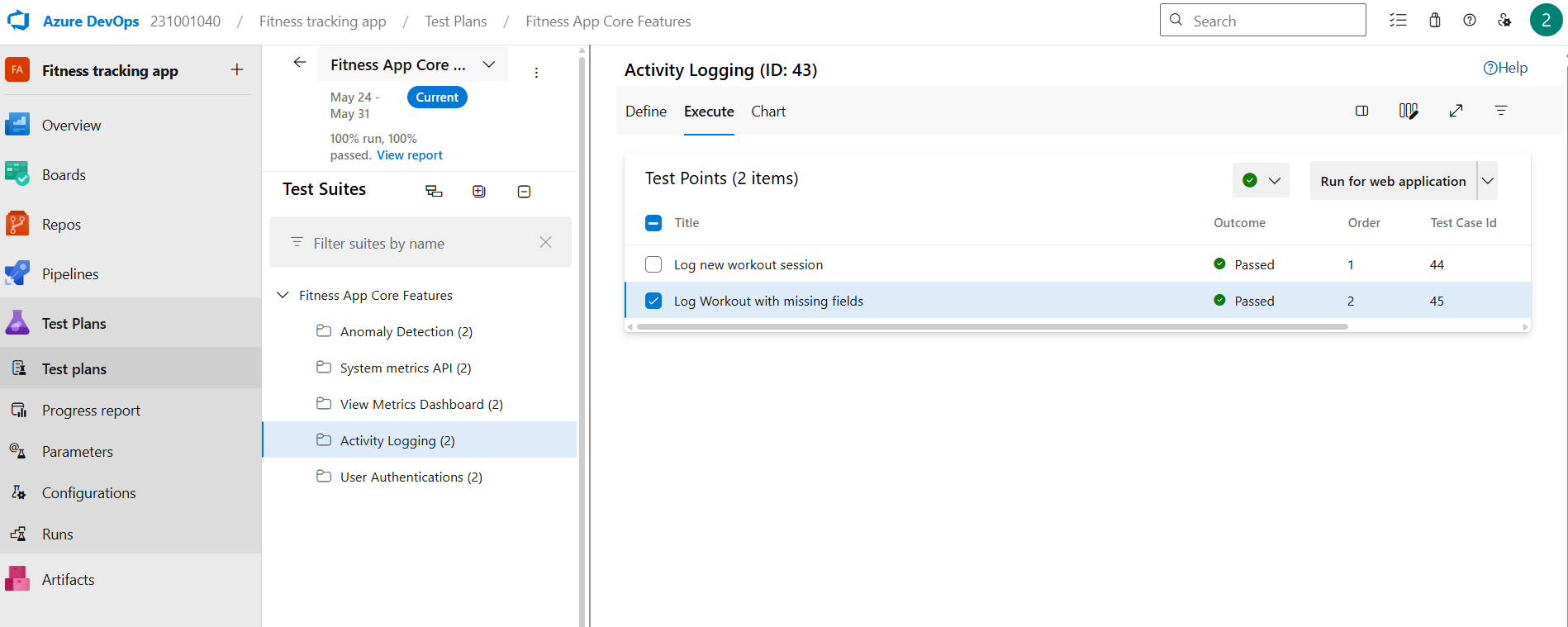


1. **Installation of test**

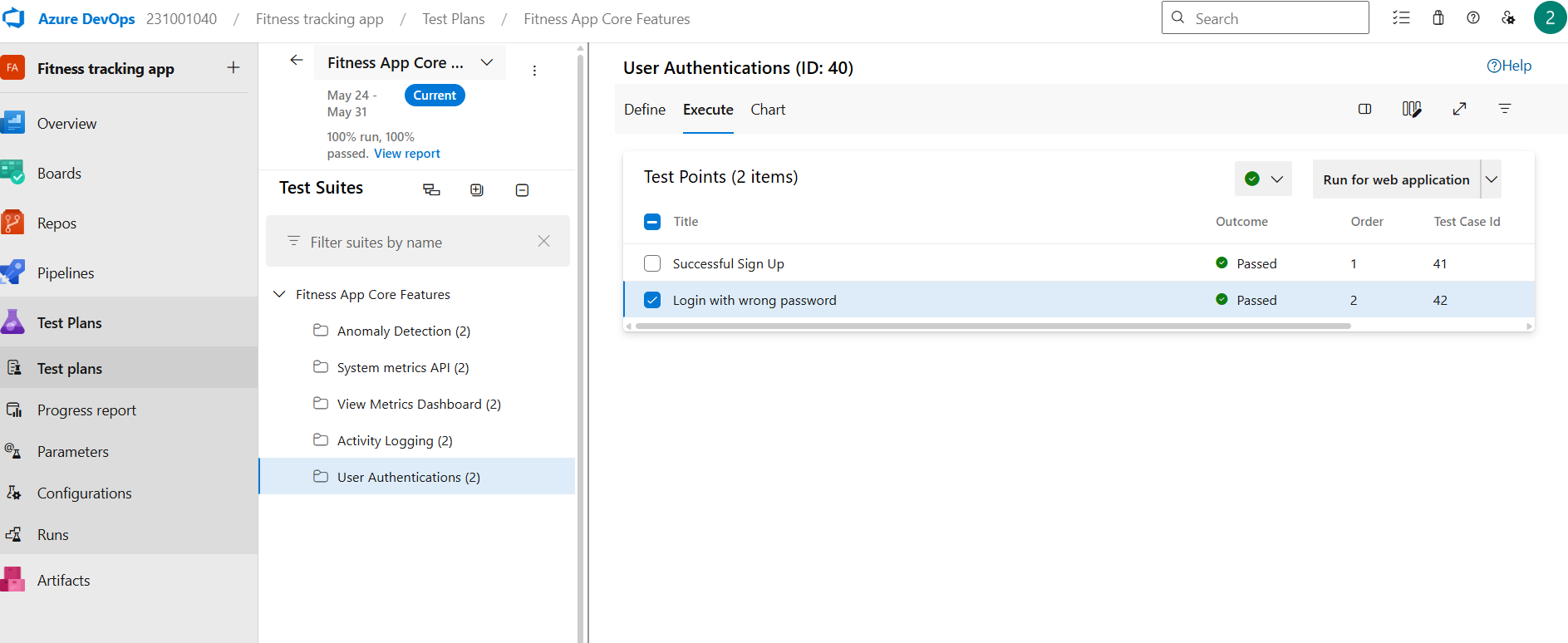
Test and feedback

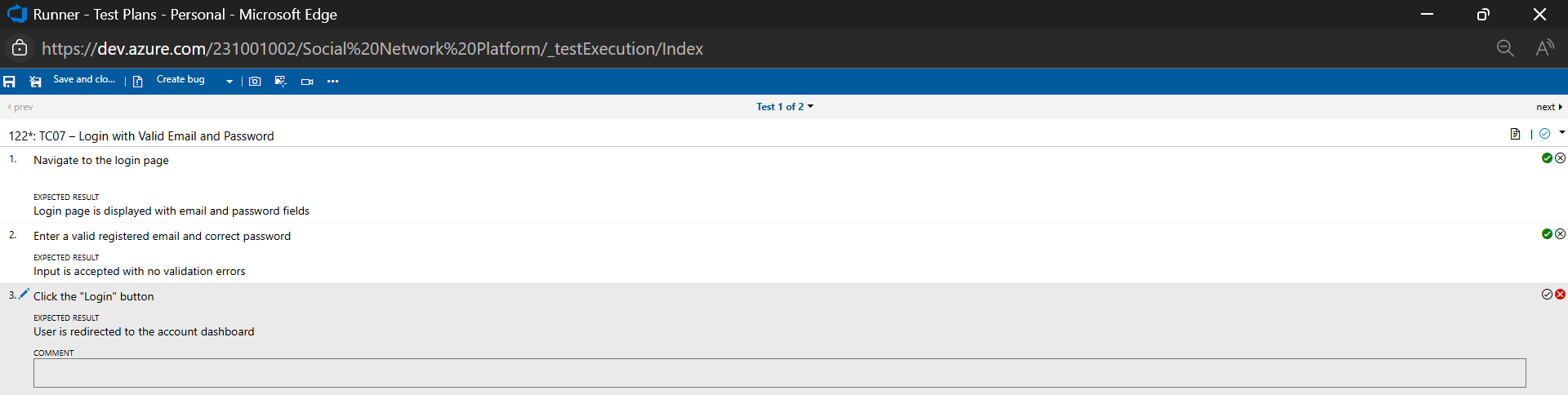


Showing it as an extension

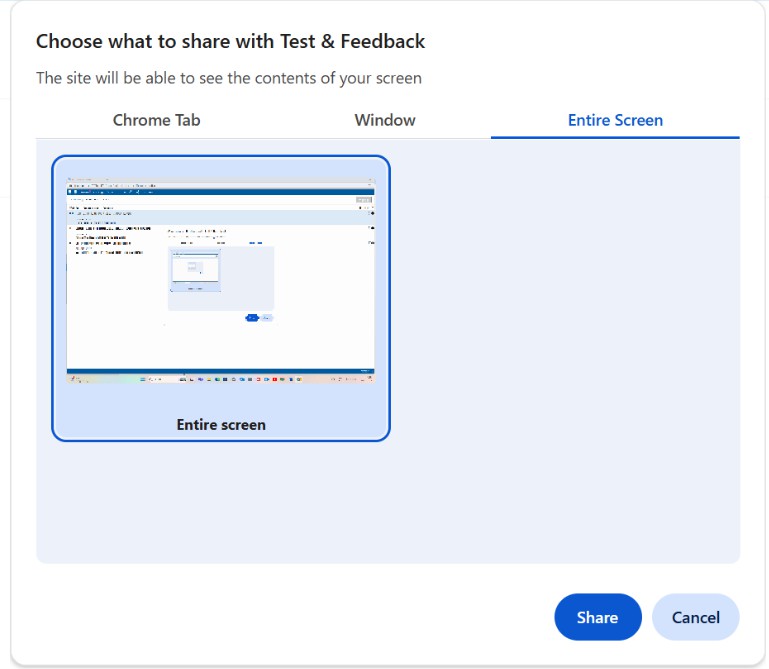
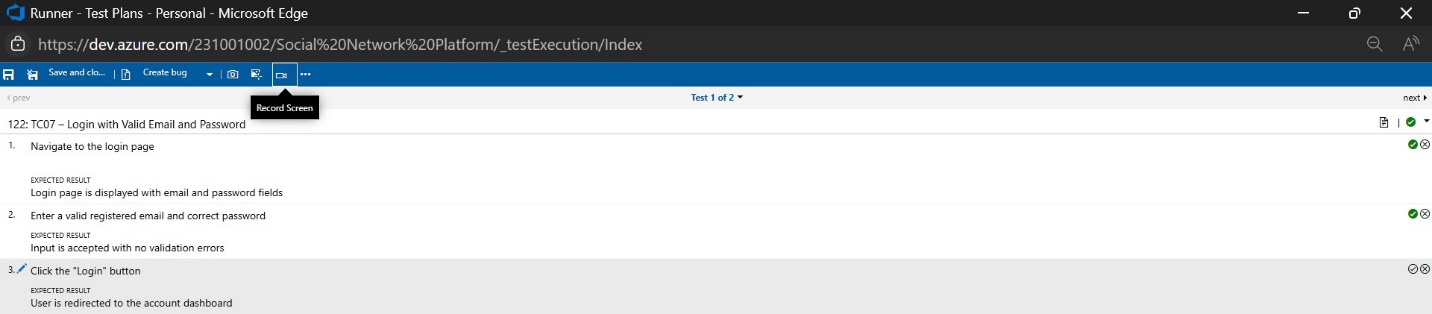


1. **Running the test cases**

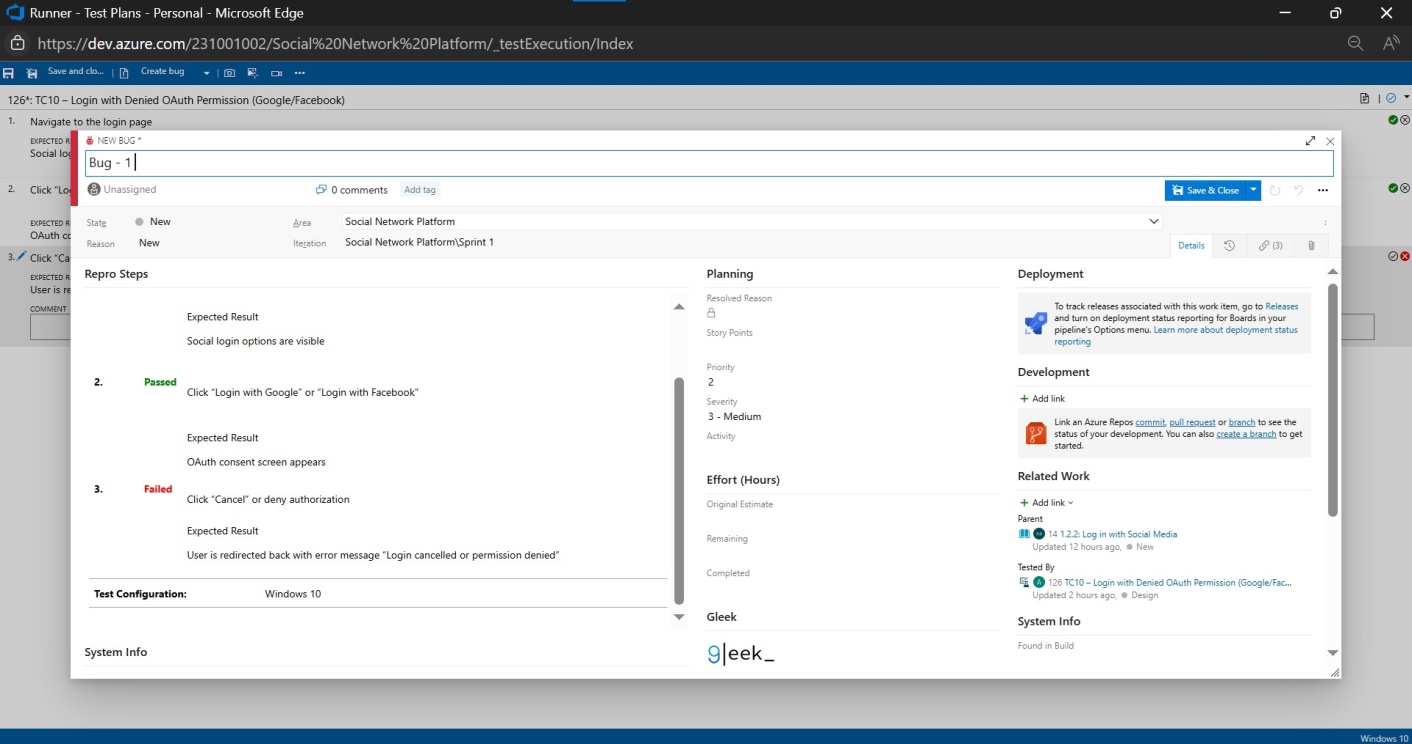
****

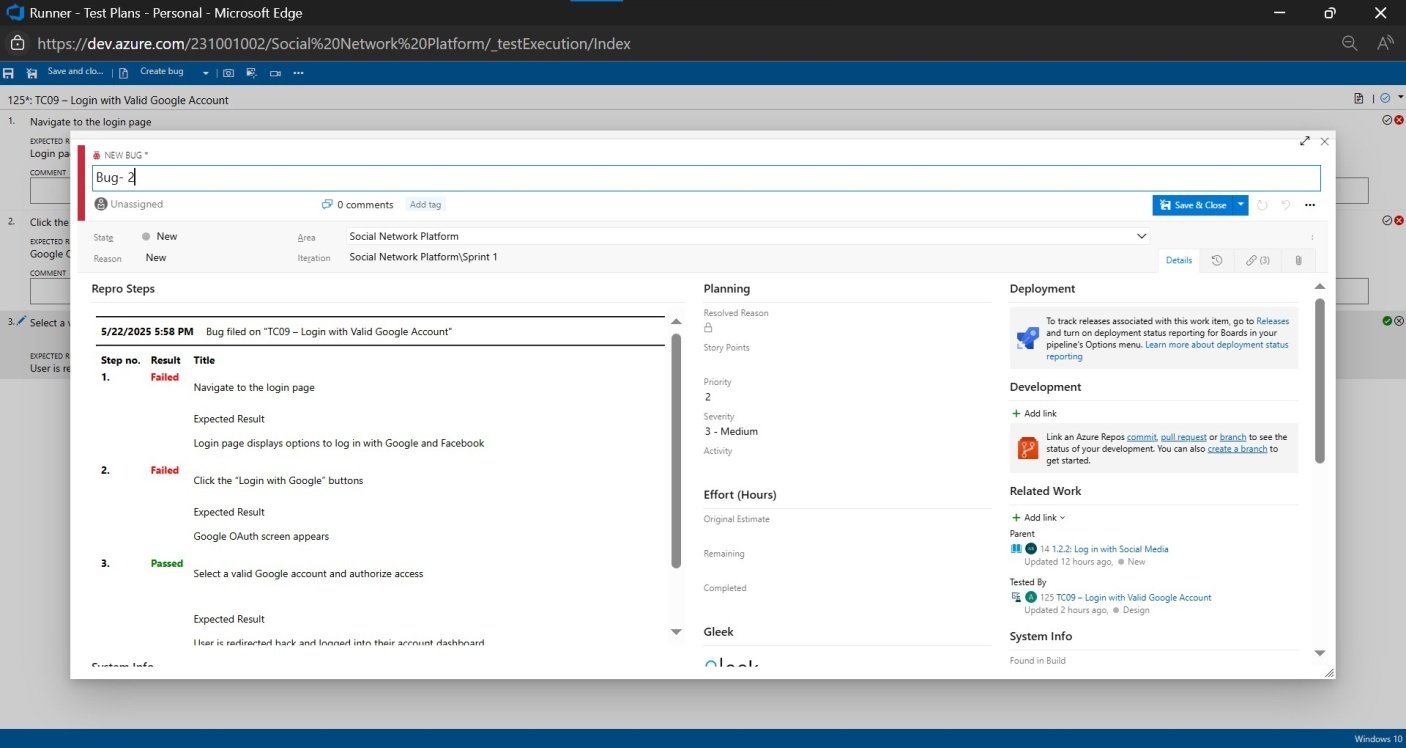


1. **Recording the test case**

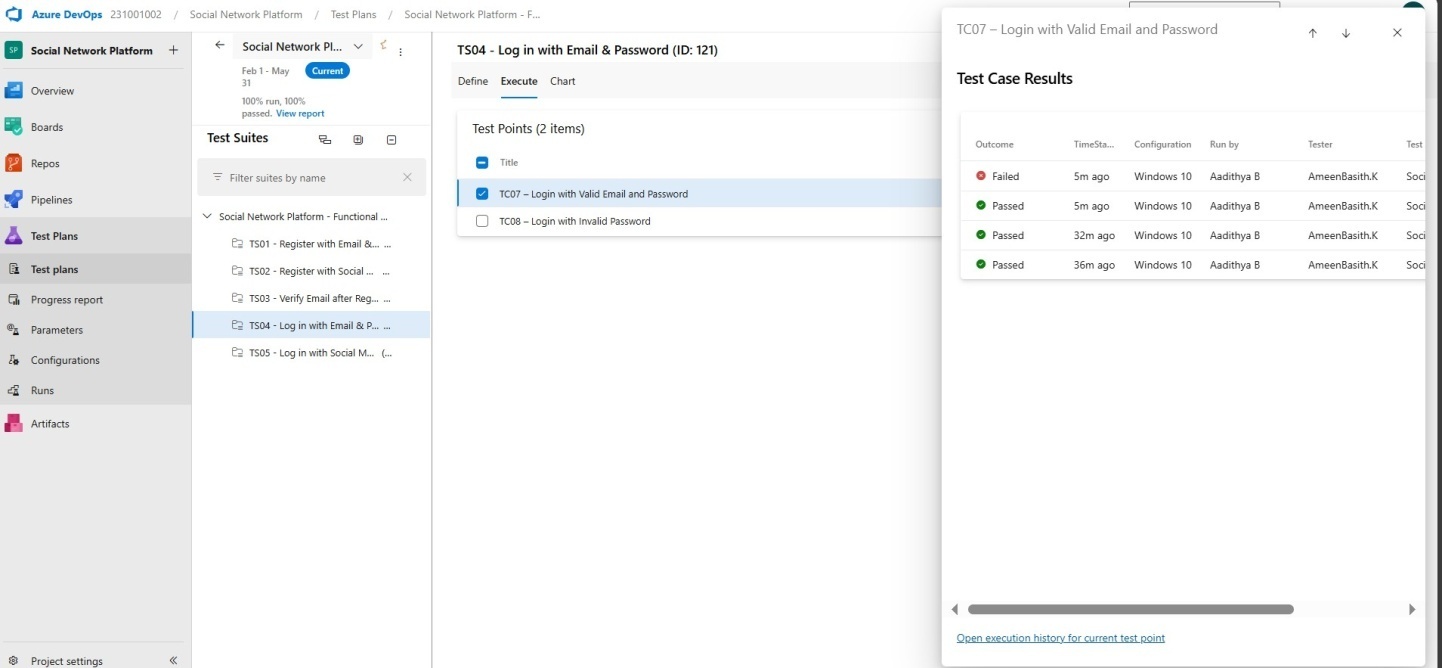


1. **Creating the bug**

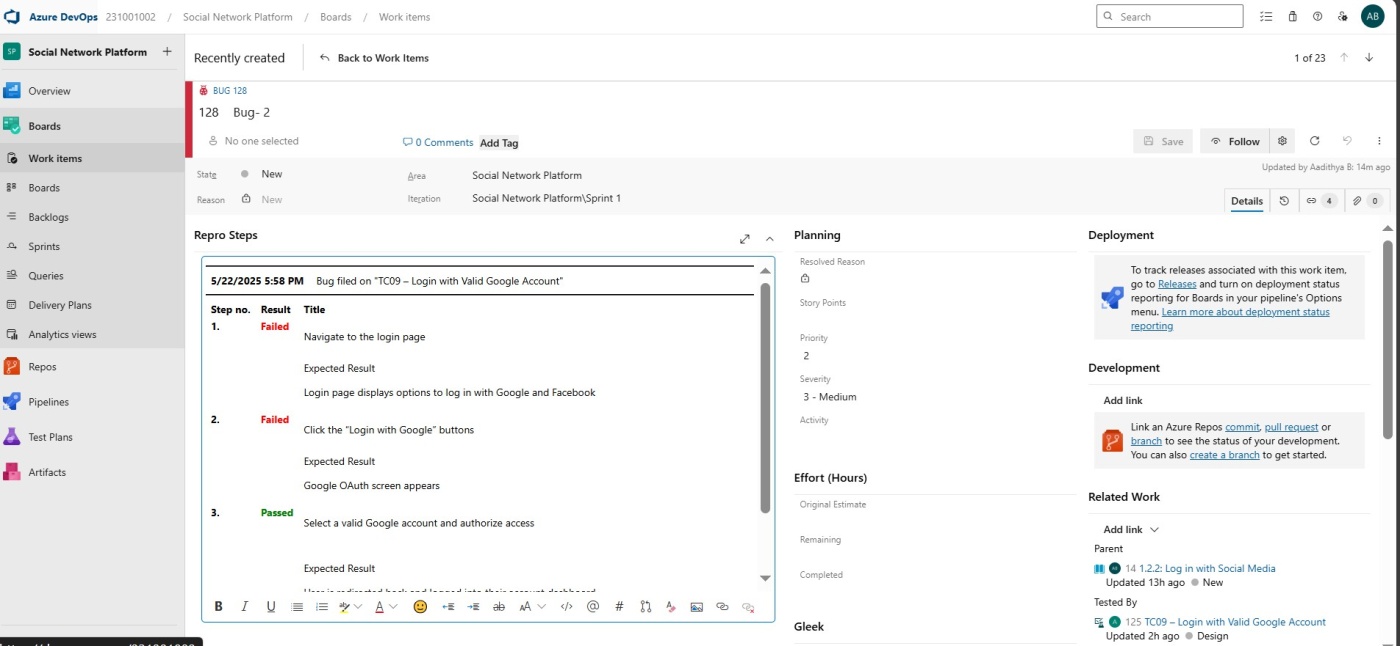
****

****

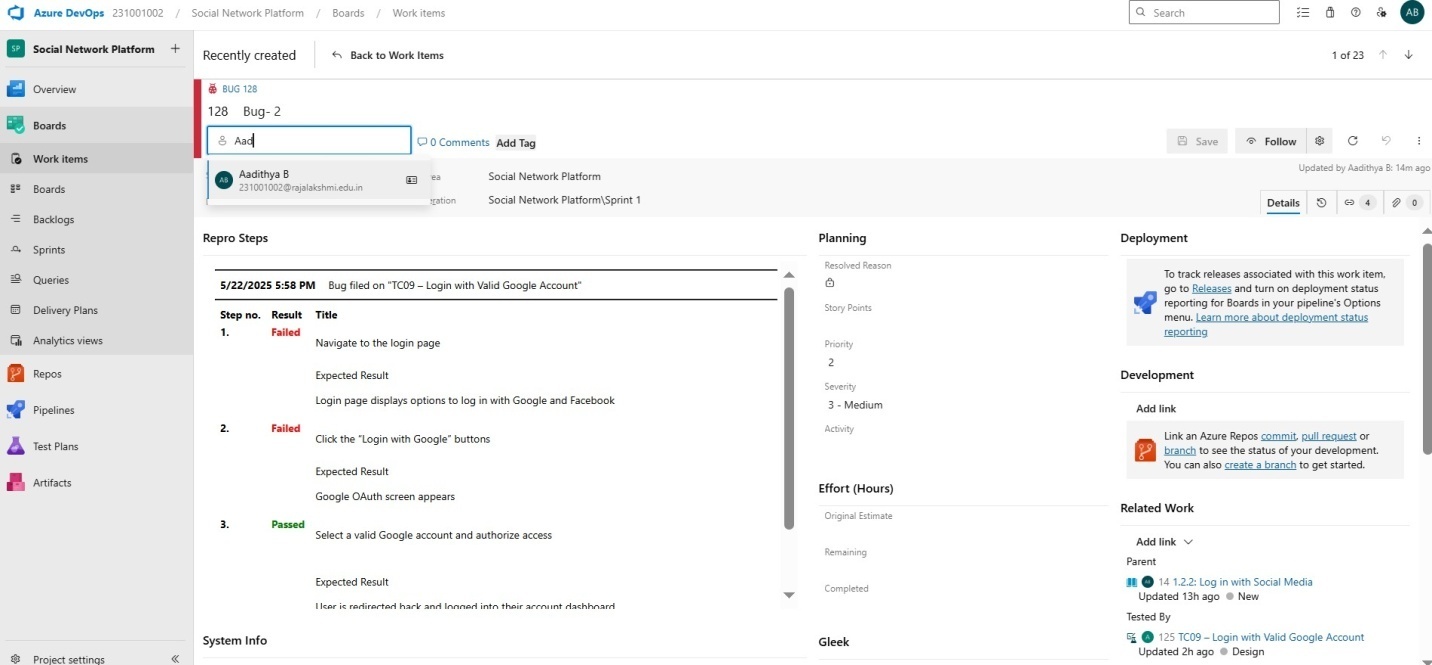
1. **Test case results**

****

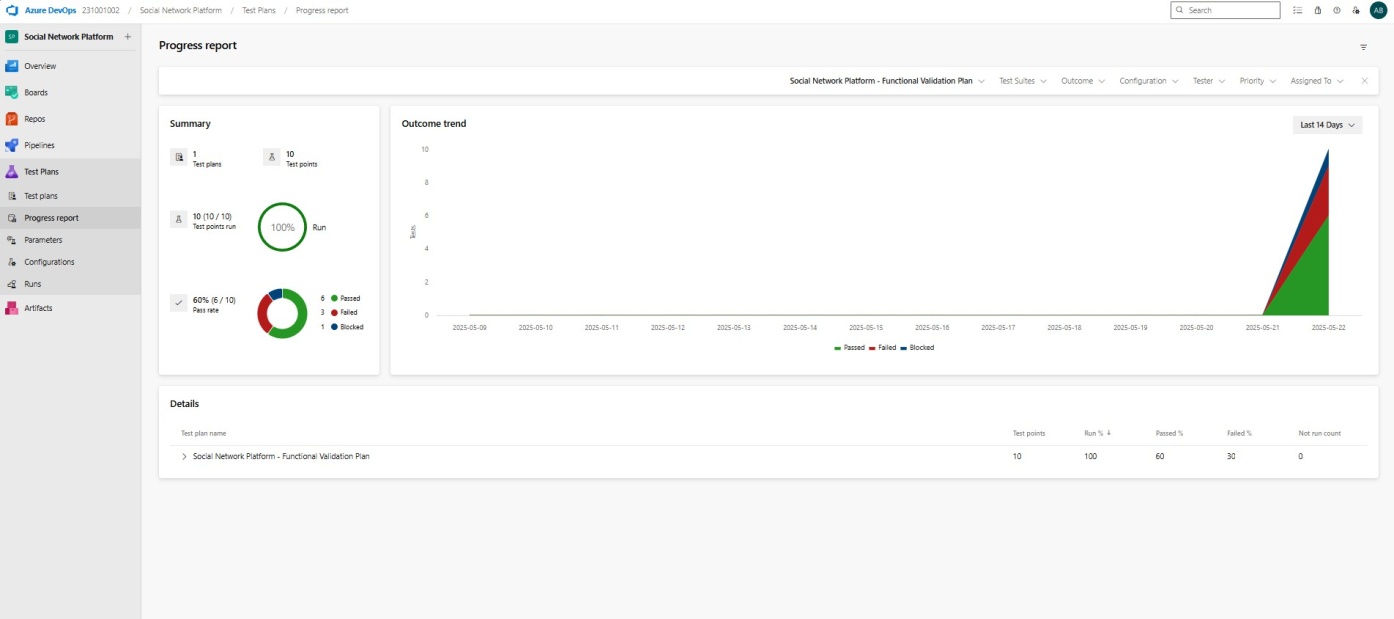
1. **Test report summary**

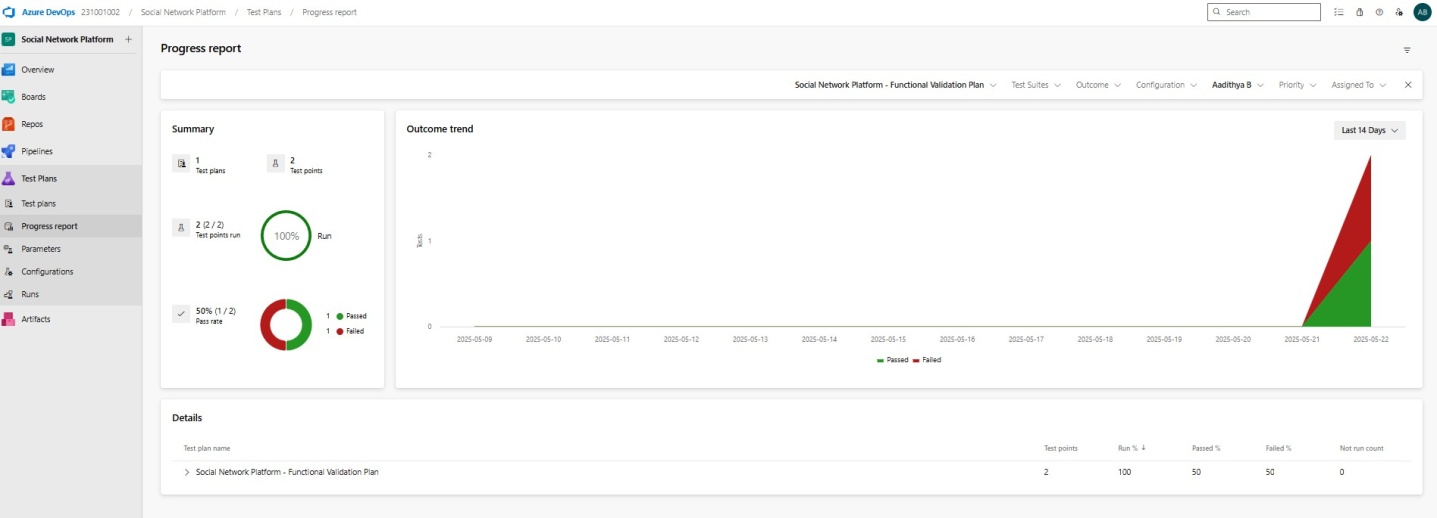
****

* Assigning bug to the developer and changing state

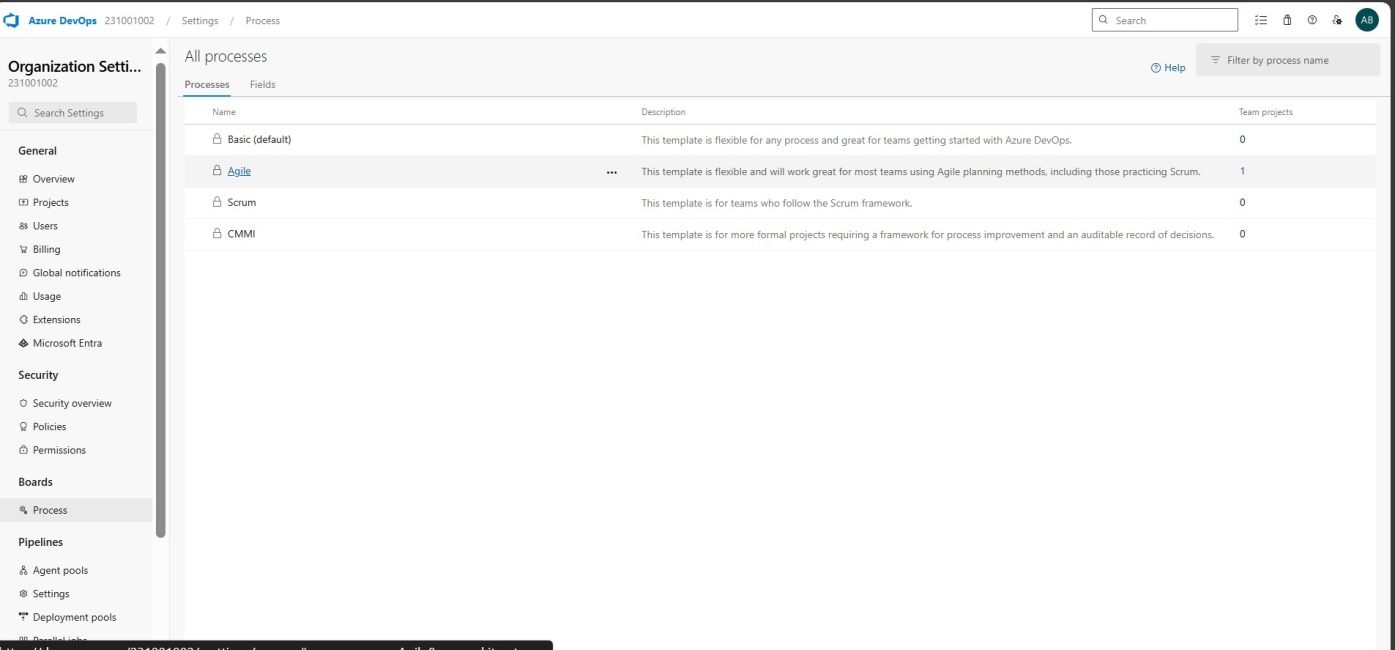


1. **Progress report**

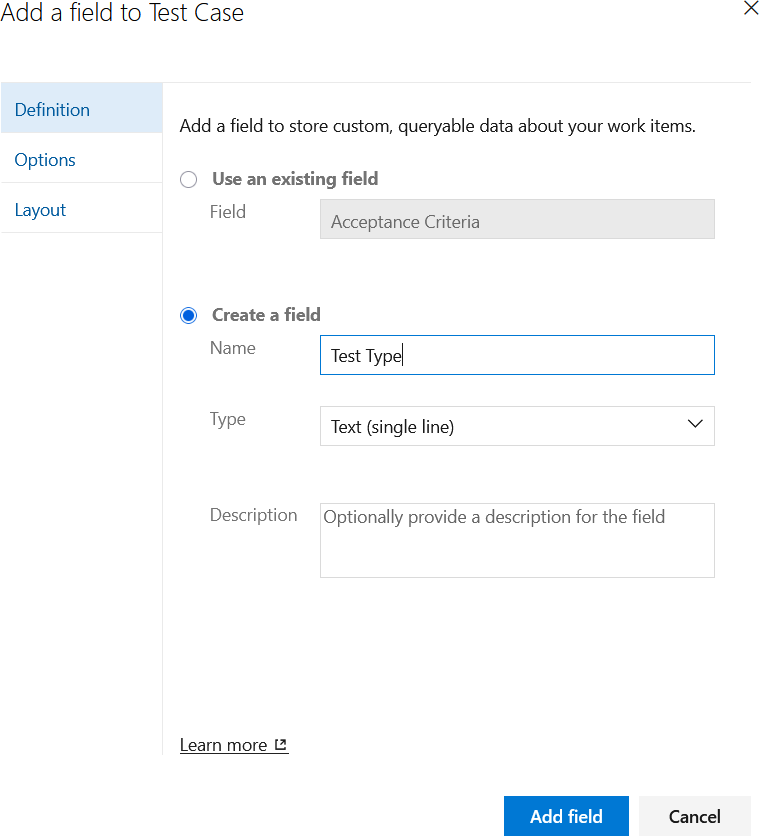
****

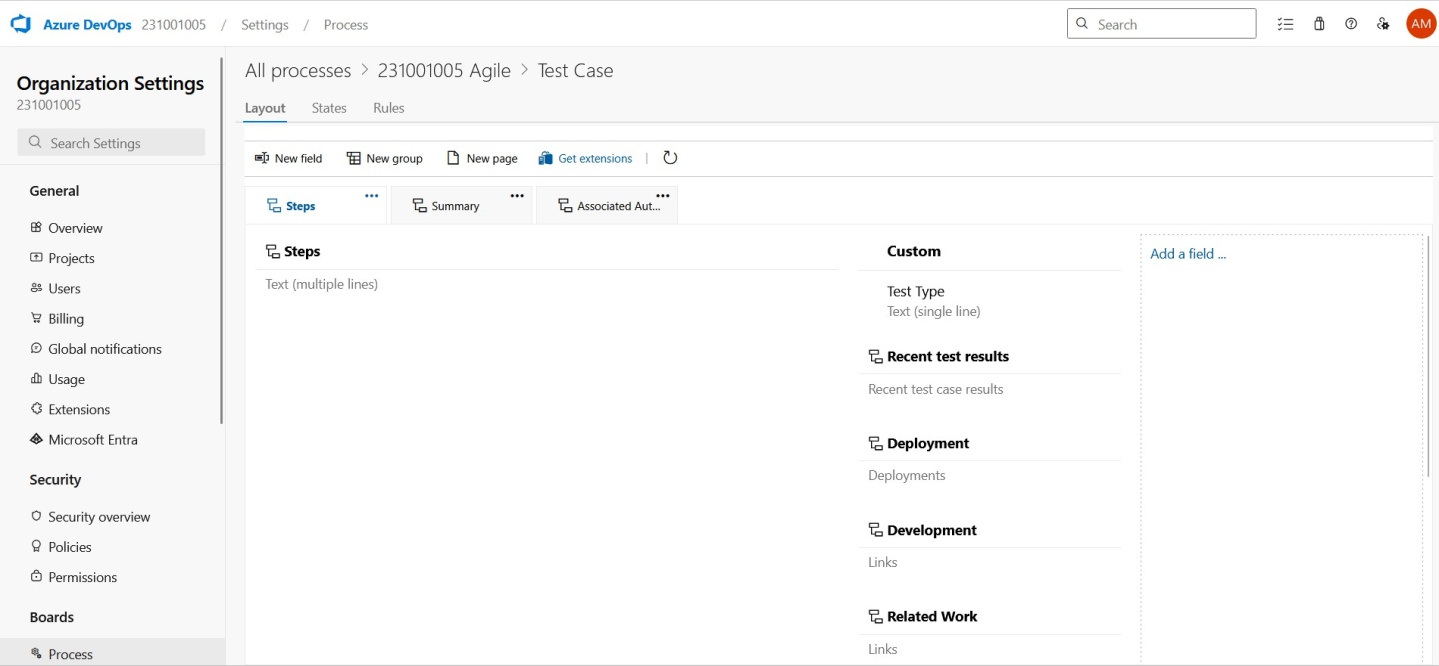


1. **Changing the test template**

****

1. **View the new test case template**

****

****

**RESULT:**The test plans and testcases for the userstories is created in AzureDevOps 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 aload test to evaluate the performance of a target end point.

## LOAD TESTING

### Steps to Create an Azure Load Testing Resource:

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

1. Sign into Azure Portal

Go to [https://portal.azure.com](https://portal.azure.com/) and login.

1. Create the Resource o Go to *Create a resource*→Search for“Azure Load Testing”.

o Select Azure Load Testing and click Create.

1. Fill in the Configuration Details
   * Subscriptio*n:* Choose your Azure subscription
   * *Resource Group:* Create new or select an existing one.
   * *Name :*Provide a unique name(nospecialcharacters).
   * *Location:* Choose the region for hosting there source.

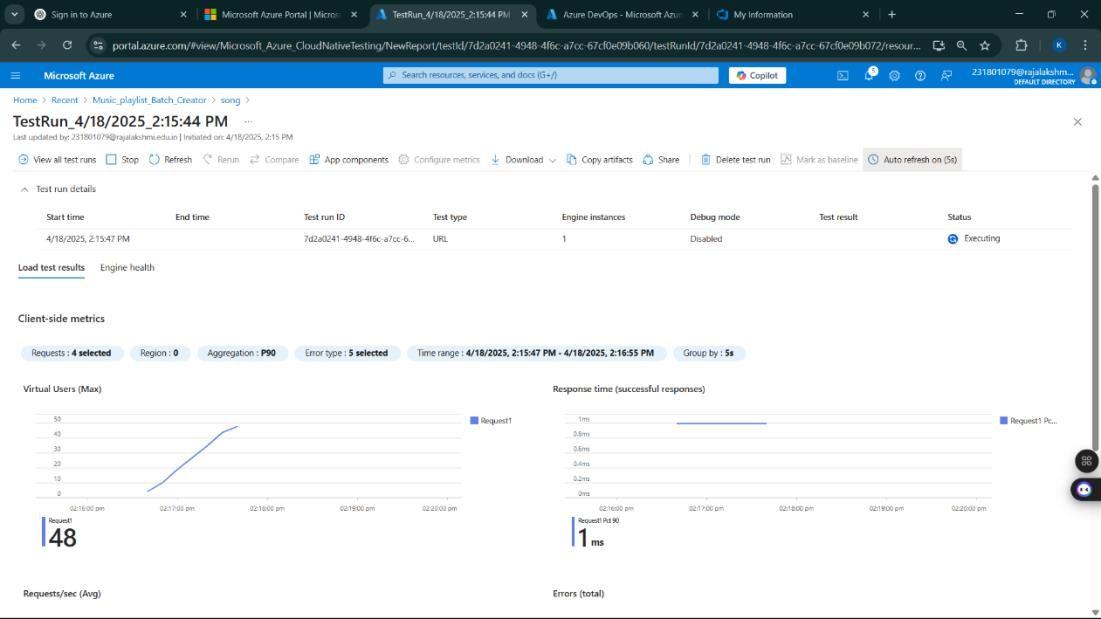
1. (Optional)Configure tags for categorization and billing.
2. Click Review +Create,then Create.
3. Once deployment is complete,click Gotoresource.

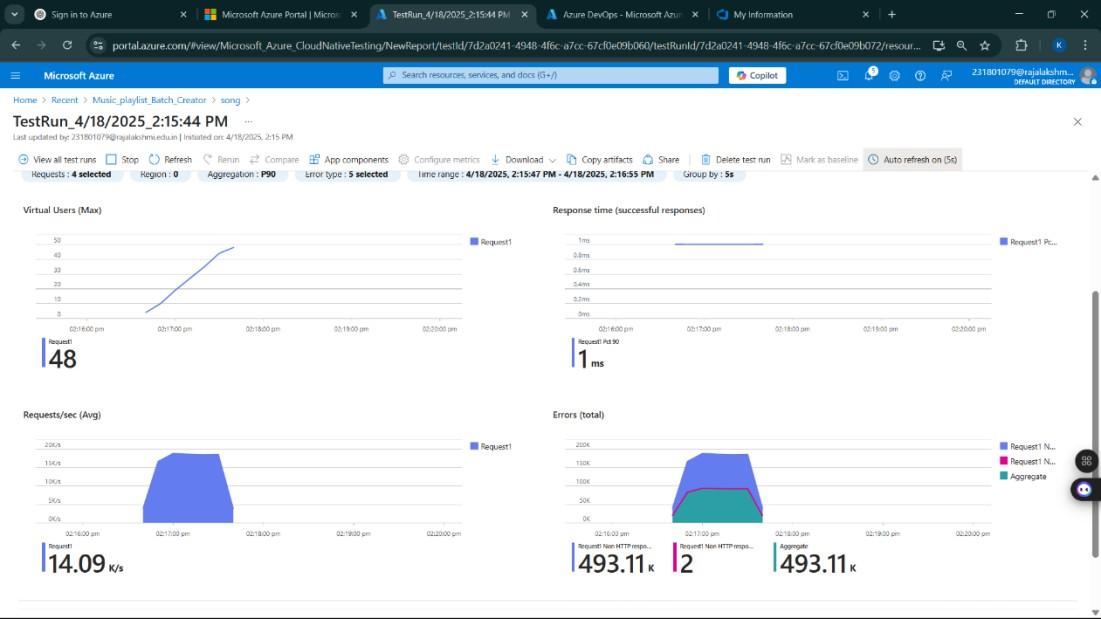
### Steps to Create and Run aLoad Test:

Once your resource is ready:

1. Go to your Azure Load Testing resource and click Add HTTP requests > Create.
2. Basics Tabo *Test Name:*Provide a unique name.
   * *Description:*(Optional) Add test purpose.
   * *Run After Creation:*Keep checked.
3. Load Setting so*Test* [*URL :*Enter](file:///C:\Users\DINESH%20S\AppData\Local\Microsoft\Windows\INetCache\IE\EC8CHZAY\Enter) the target end point(e.g.,https://yourapi.com/products).
4. Click Review +Create →Create to start the test.

### LoadTesting





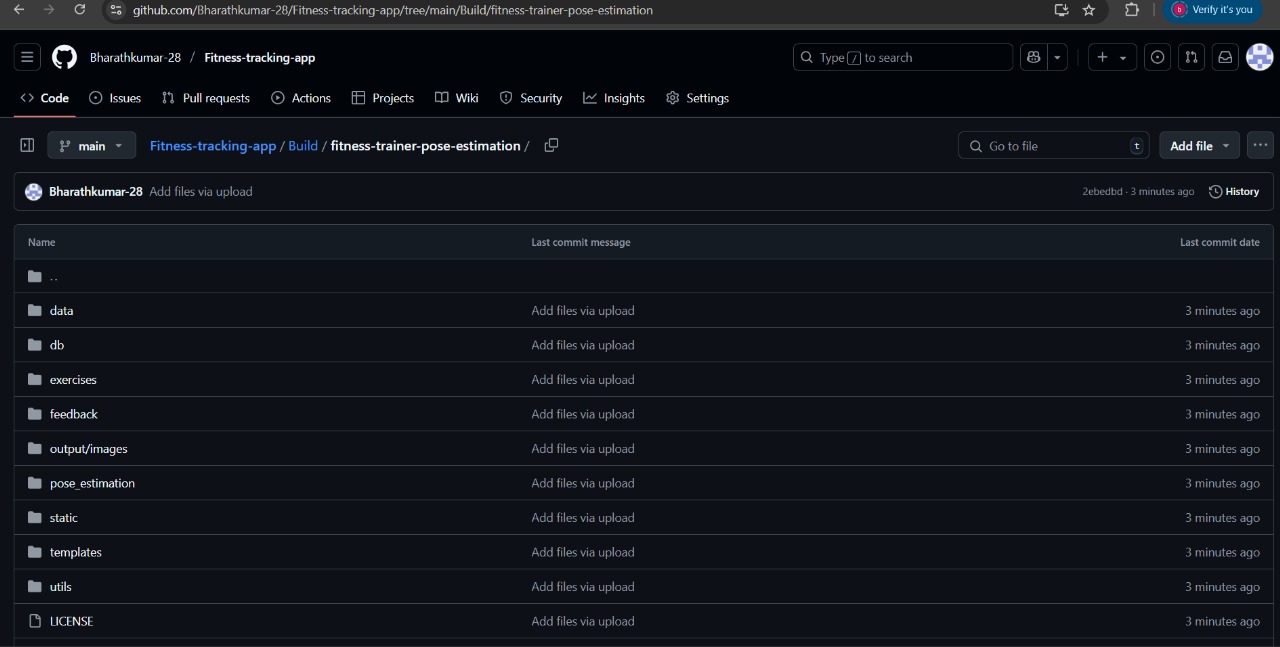


**RESULT:** Successfully created Azure Load Testing resource and executed a load test to assess the performance of the specified end point.

|  |  |
| --- | --- |
| **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 Social Network Platform 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.