|  |  |
| --- | --- |
| **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.

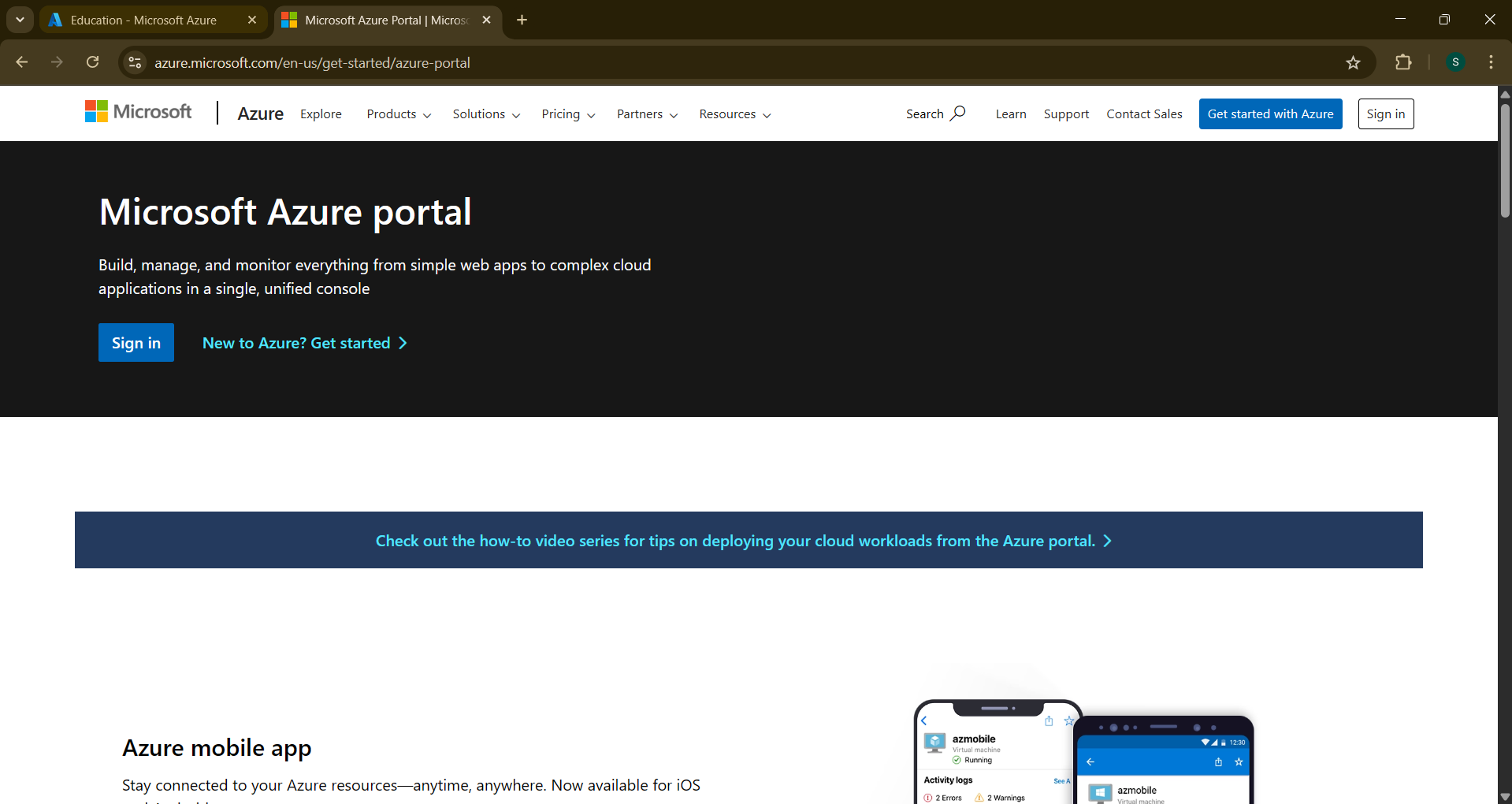
**INSTALLATION**

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

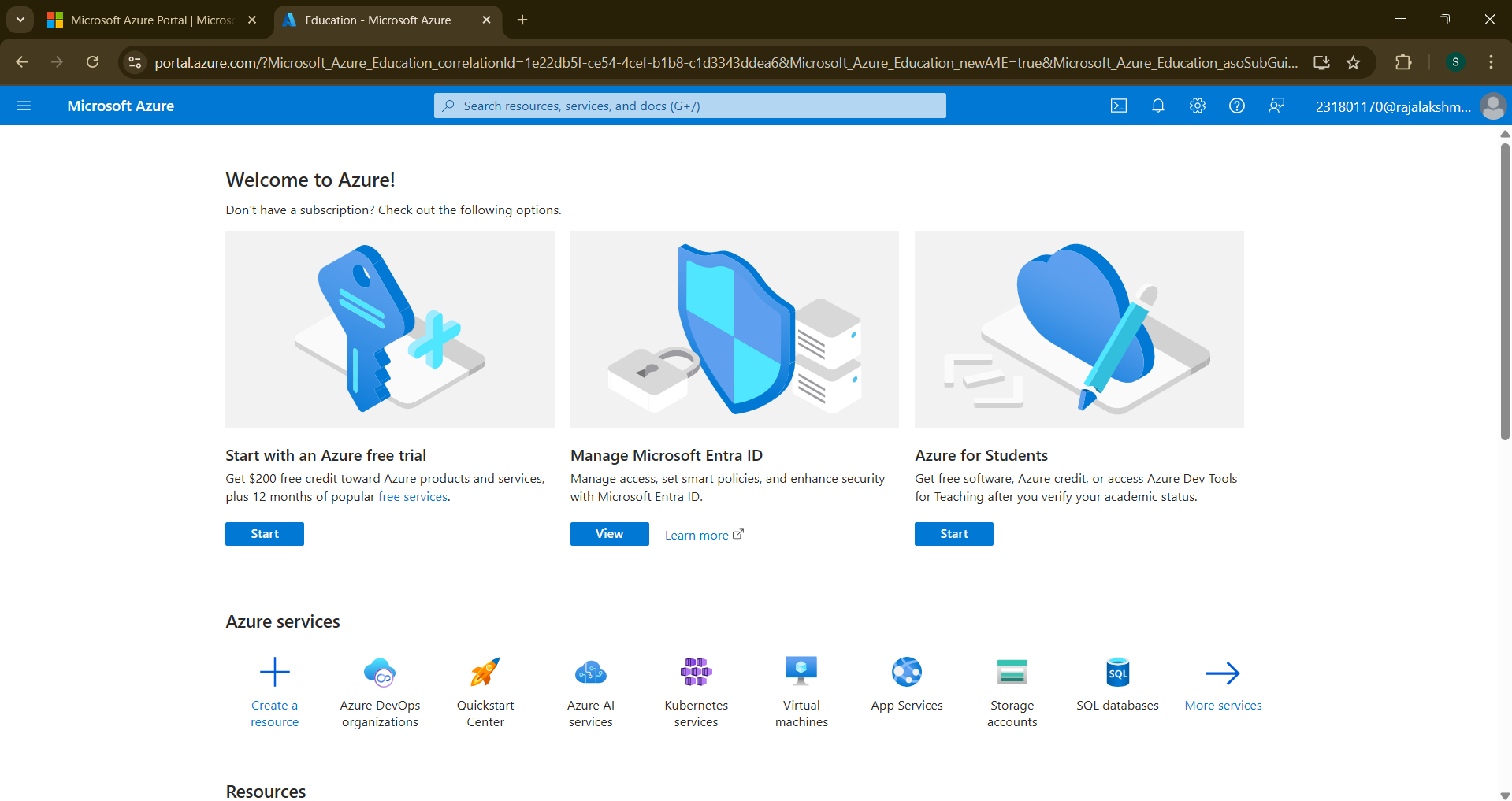
started/azure-portal.

Sign in using your Microsoft account credentials.

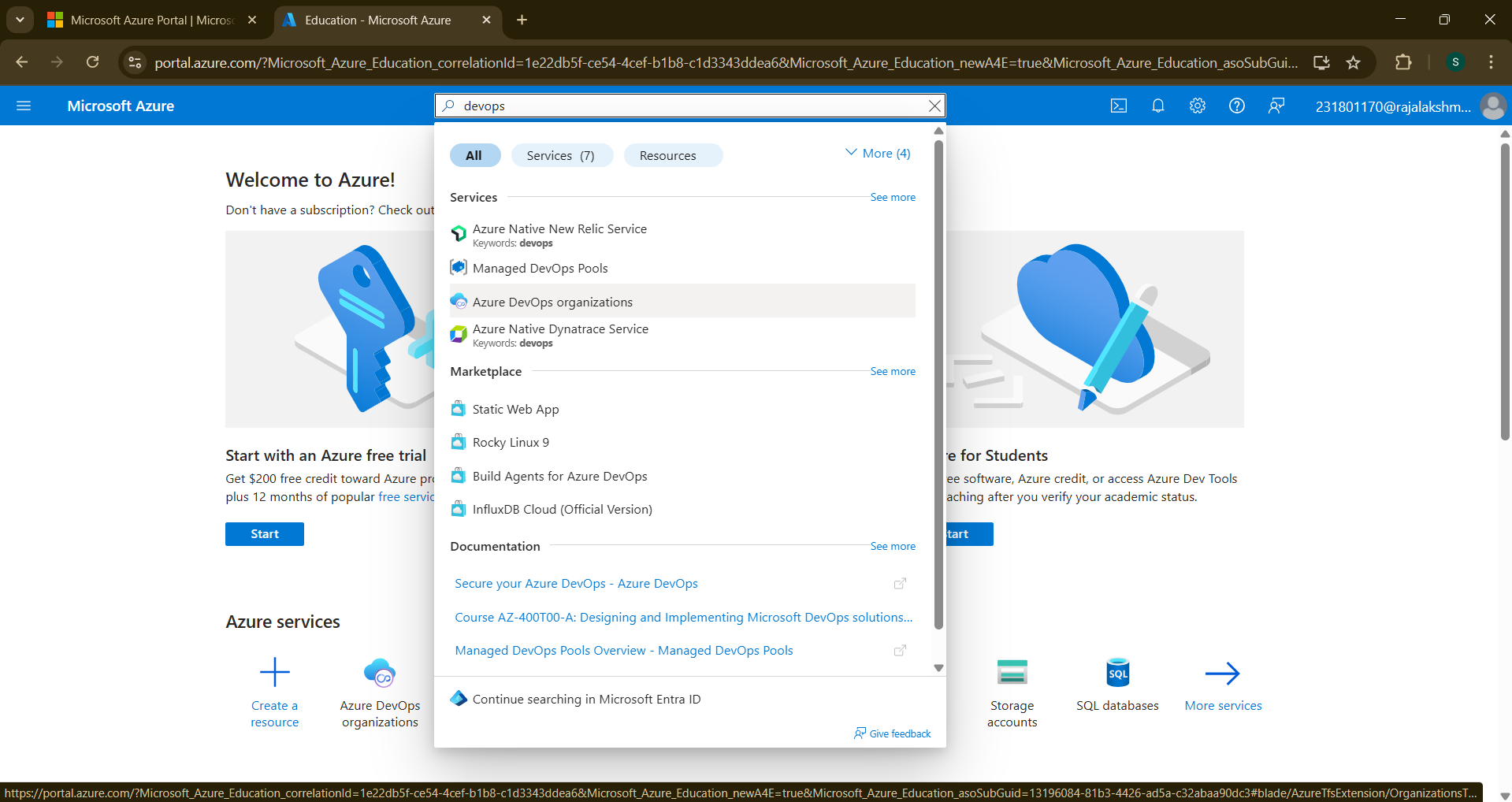
If you don't have a Microsoft account, you can create one here: <https://signup.live.com/?lic=1>



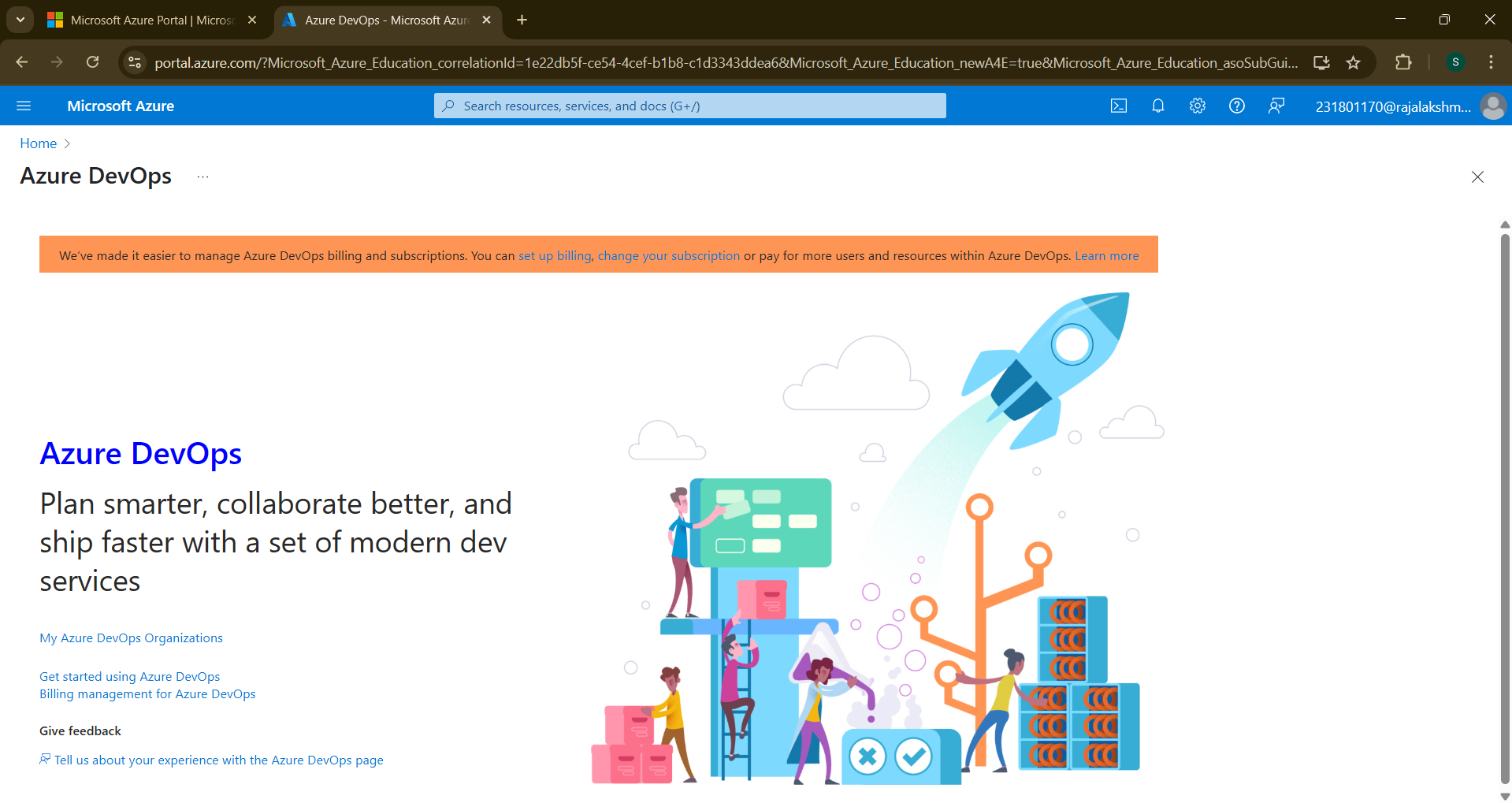
2.Azure home page



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:**

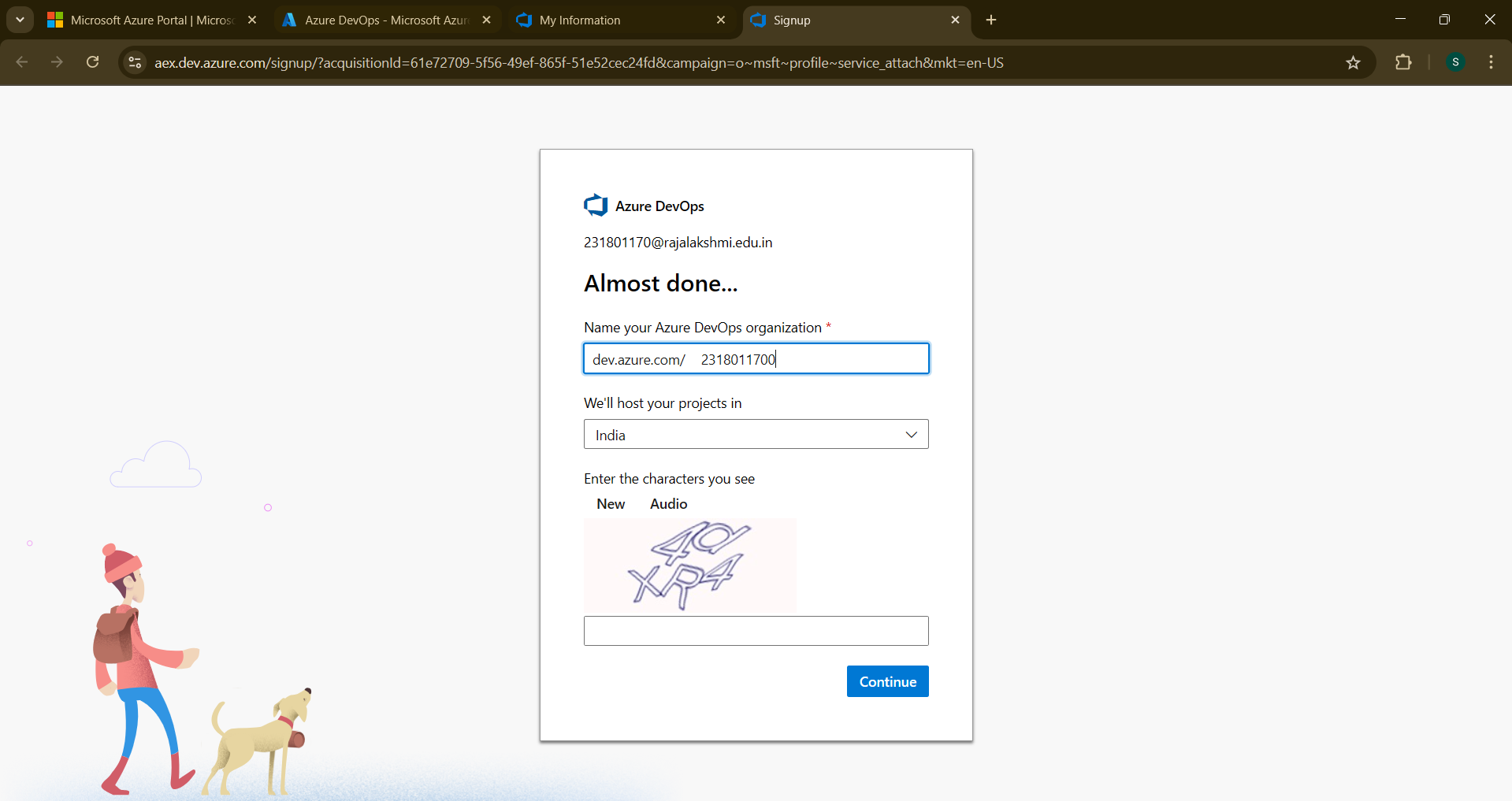
Successfully accessed the Azure DevOps environment and created a new organization through the Azure portal.

|  |  |
| --- | --- |
| **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.

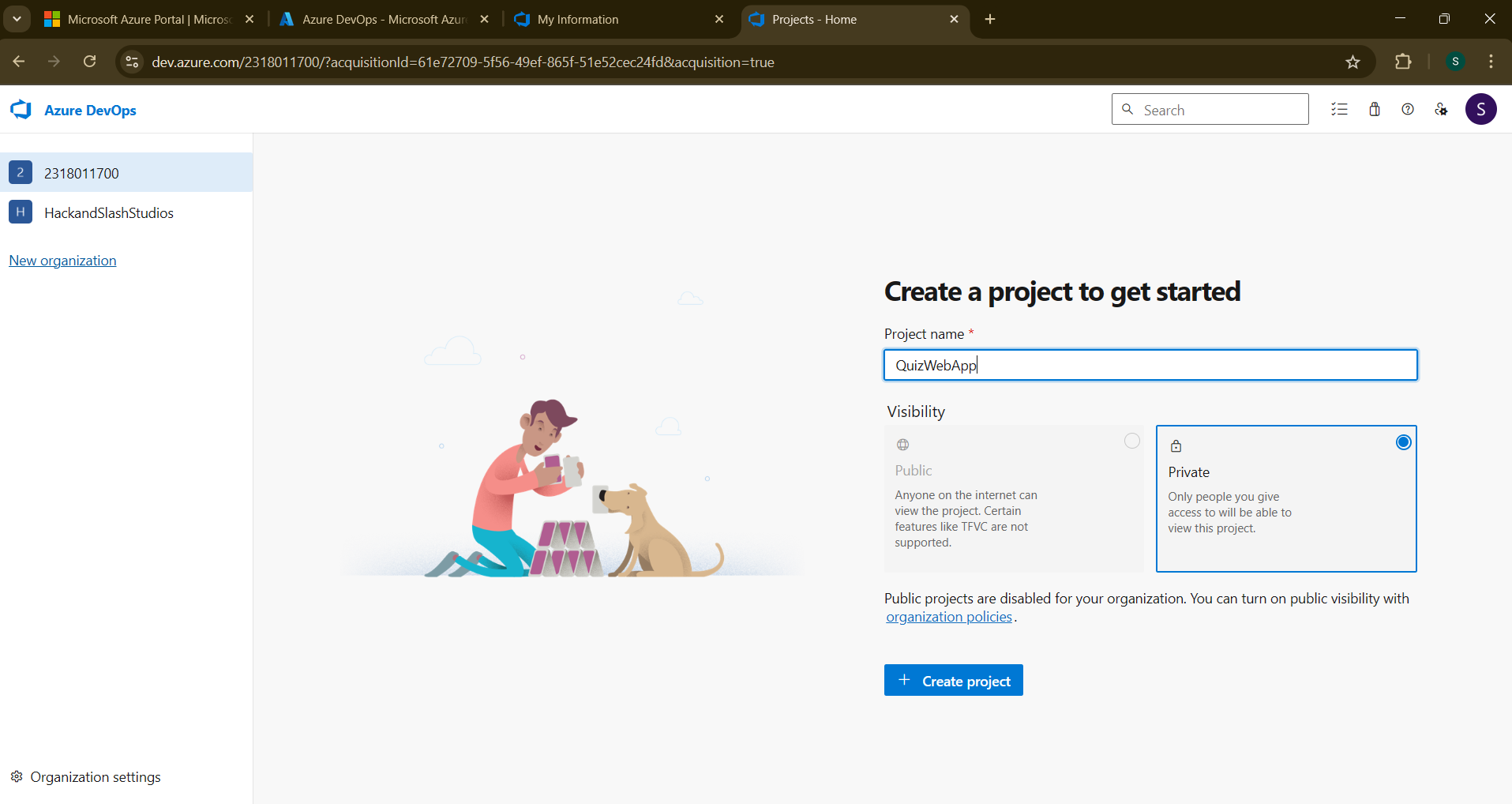
c. Enter the project name, description, and visibility options:

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

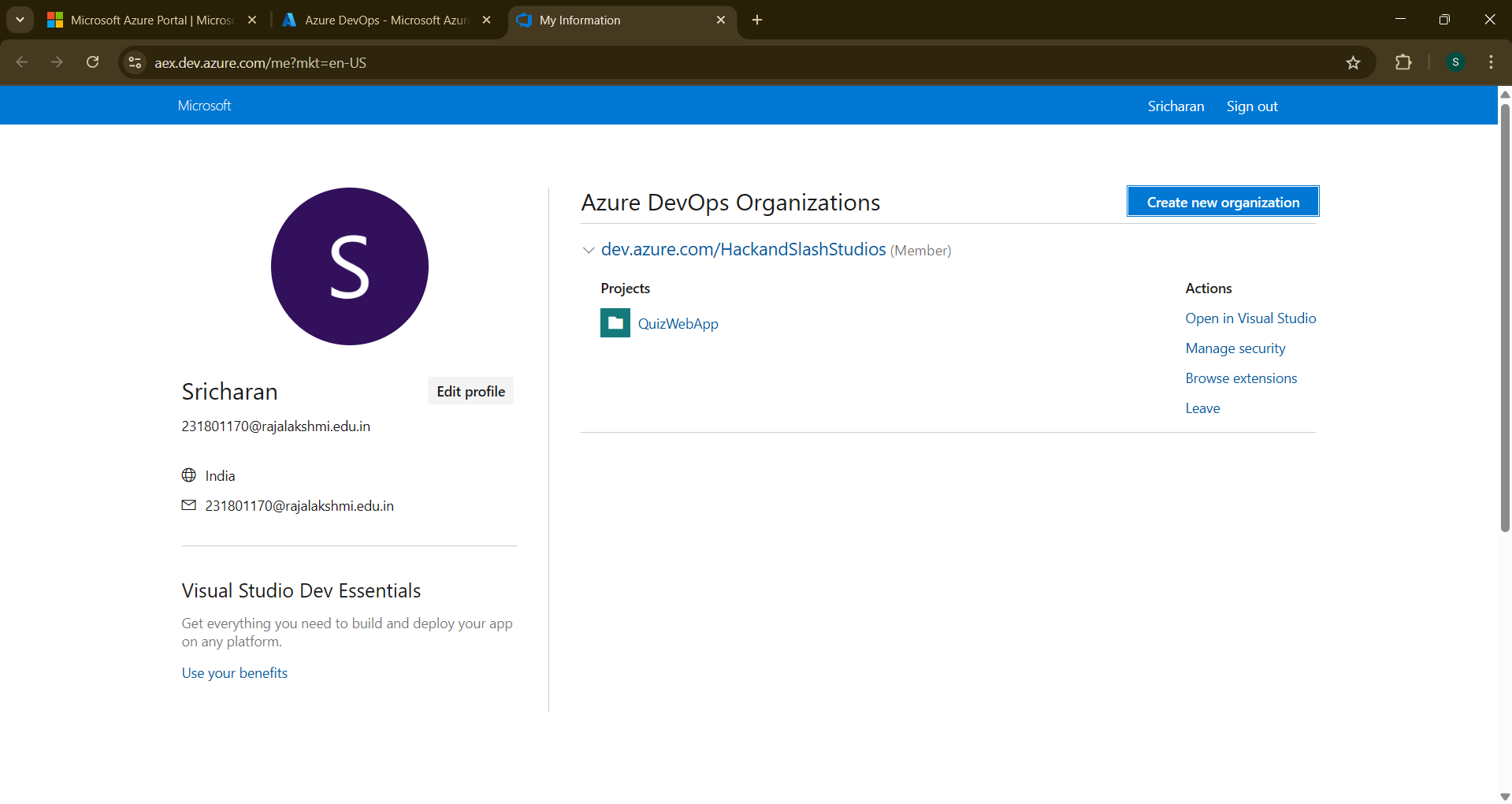
**Description**: Optionally, add a description to provide more context about the project.

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

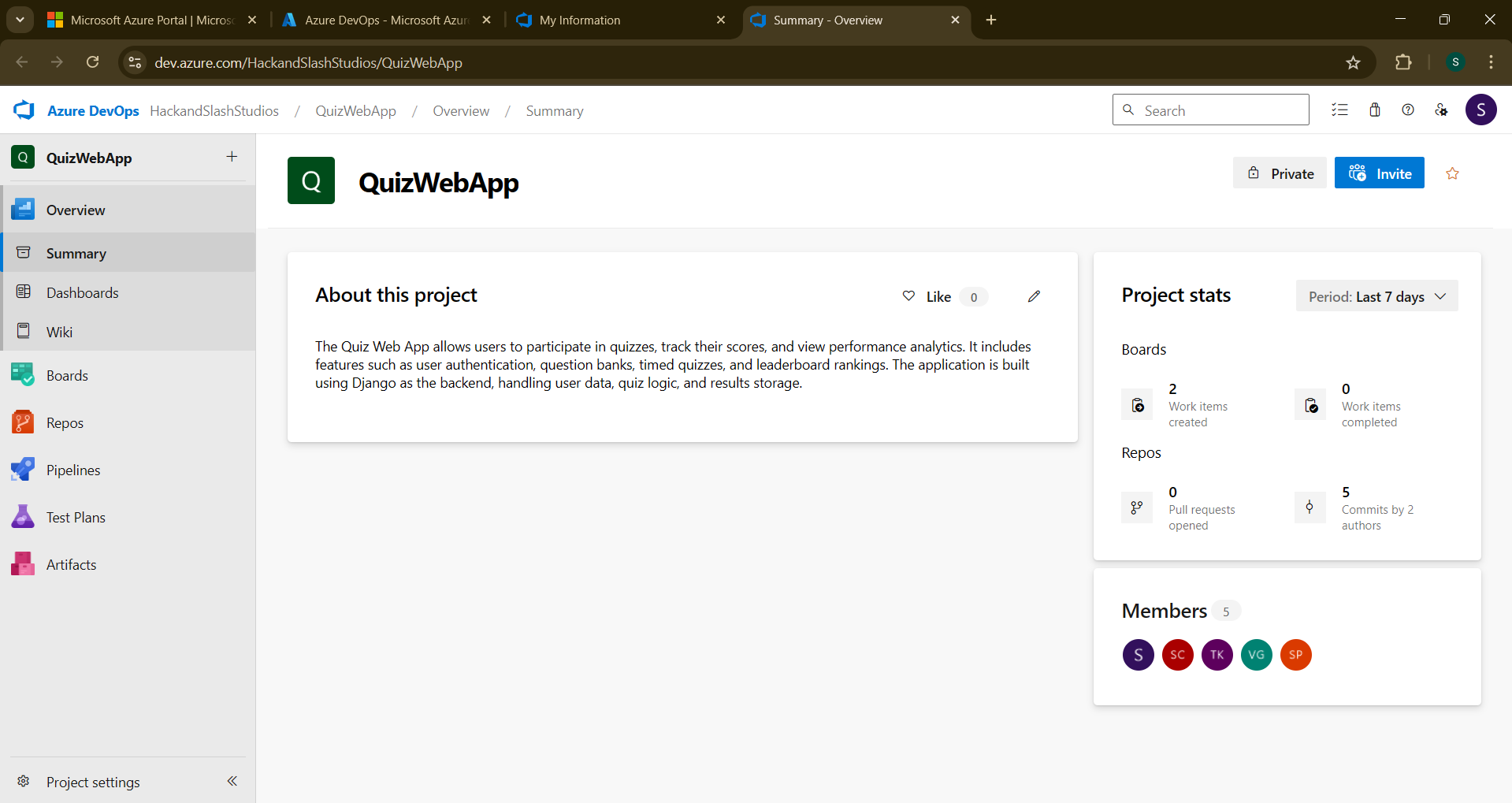
d. Once you’ve filled out the details, click **Create** to set up your first project.



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.



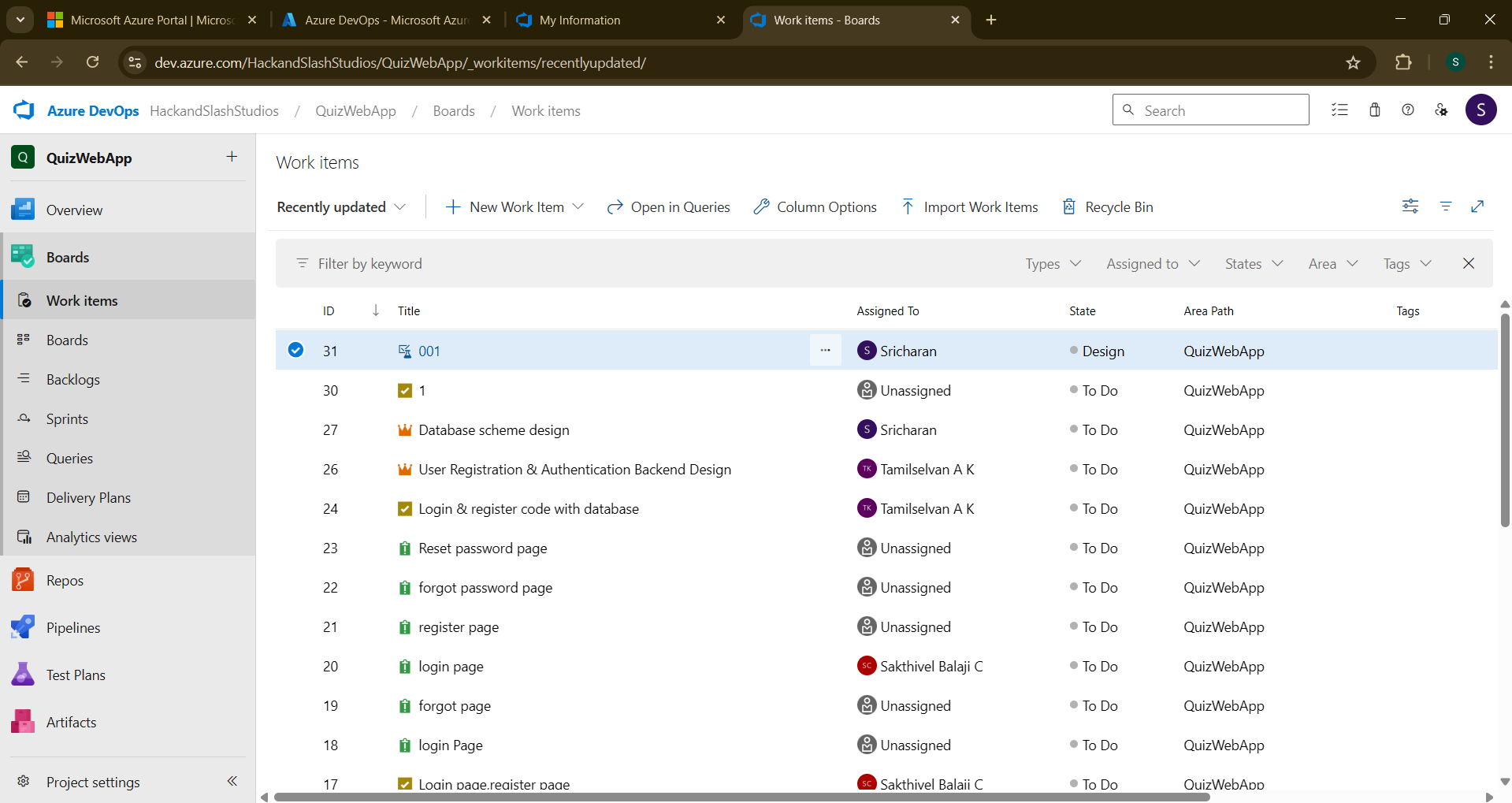
4.Project dashboard

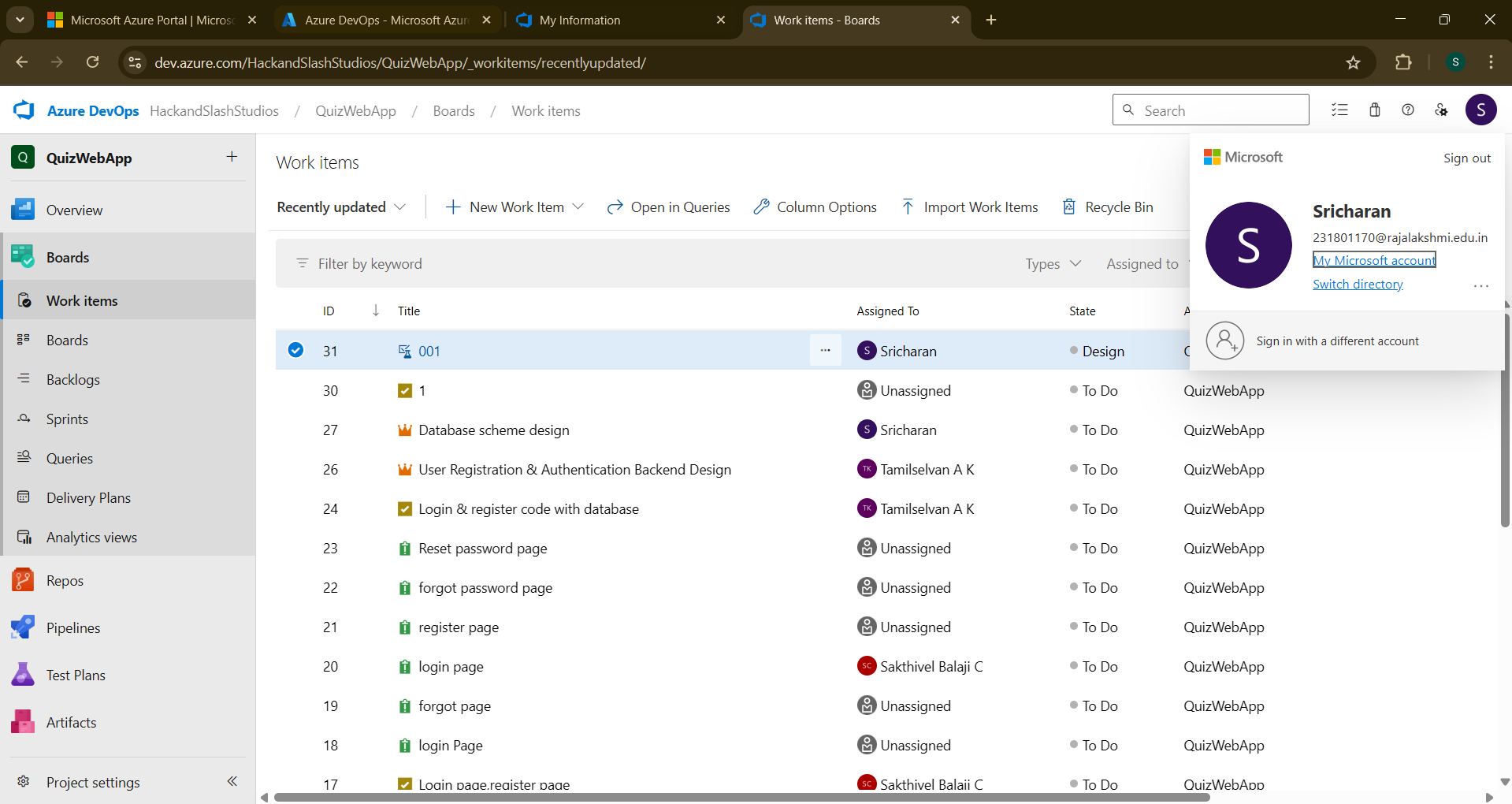


5.To manage user stories:

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:**

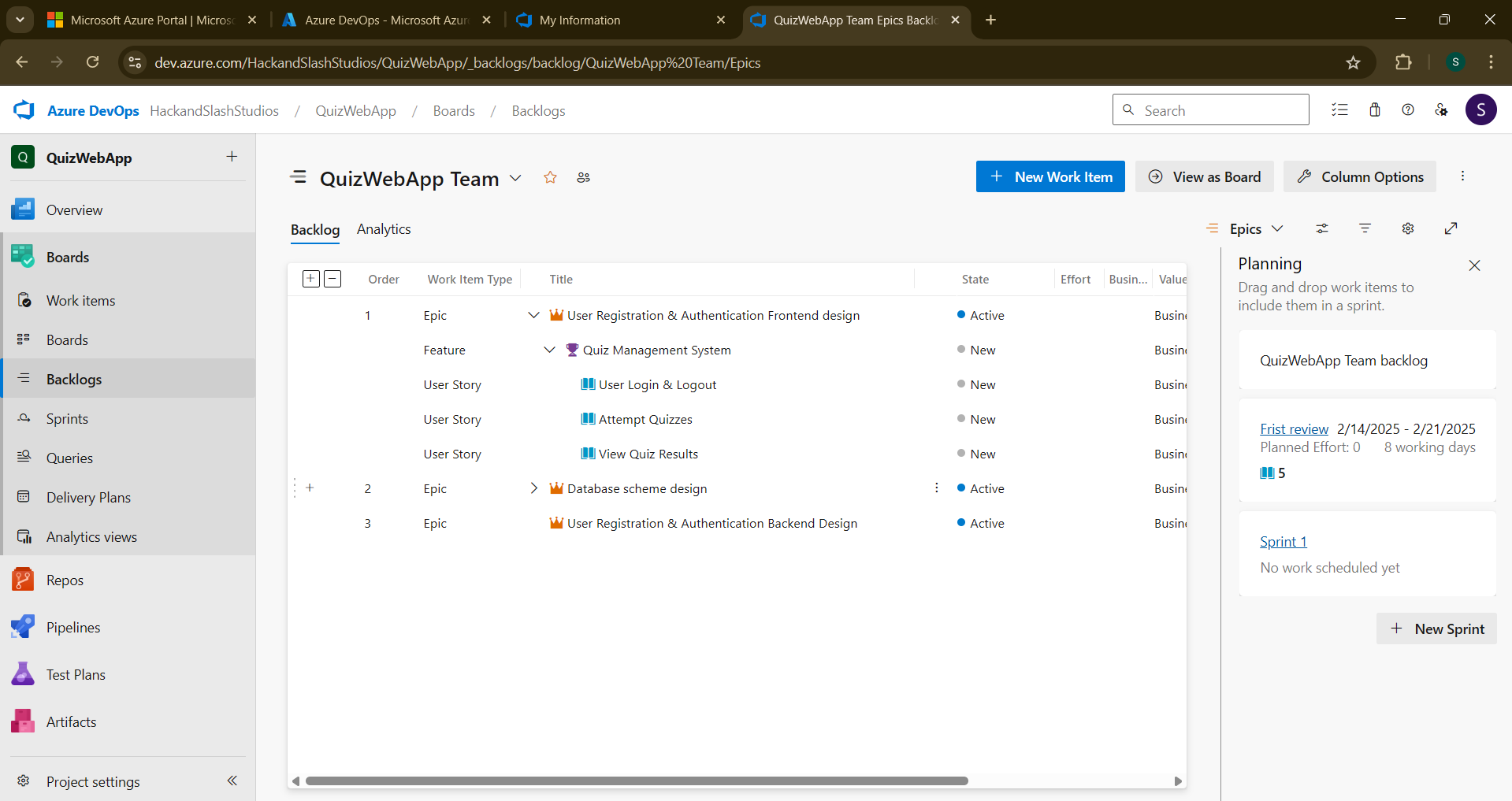
Successfully created an Azure DevOps project with user story management and agile workflow setup.

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

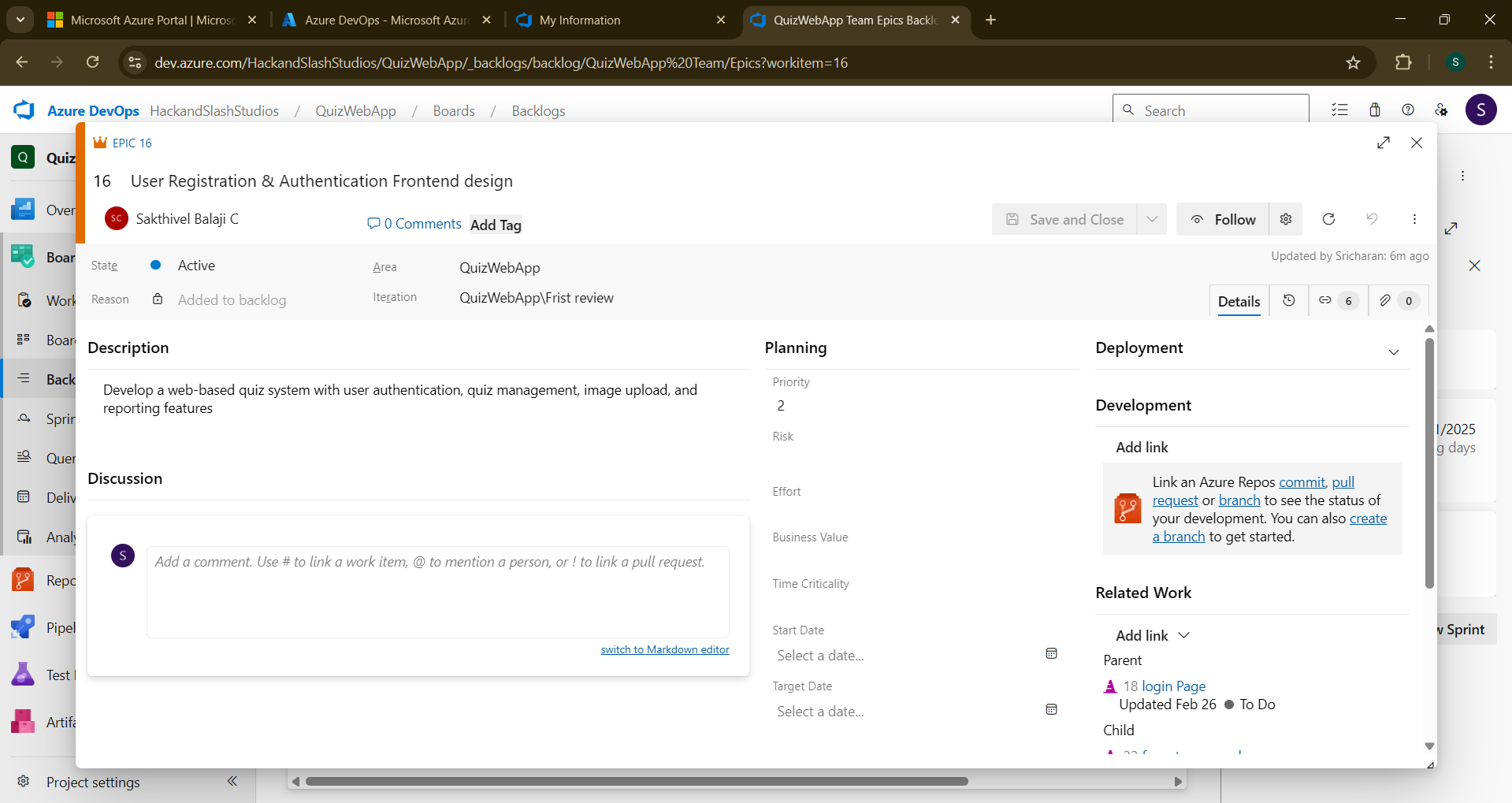
**Aim:**

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

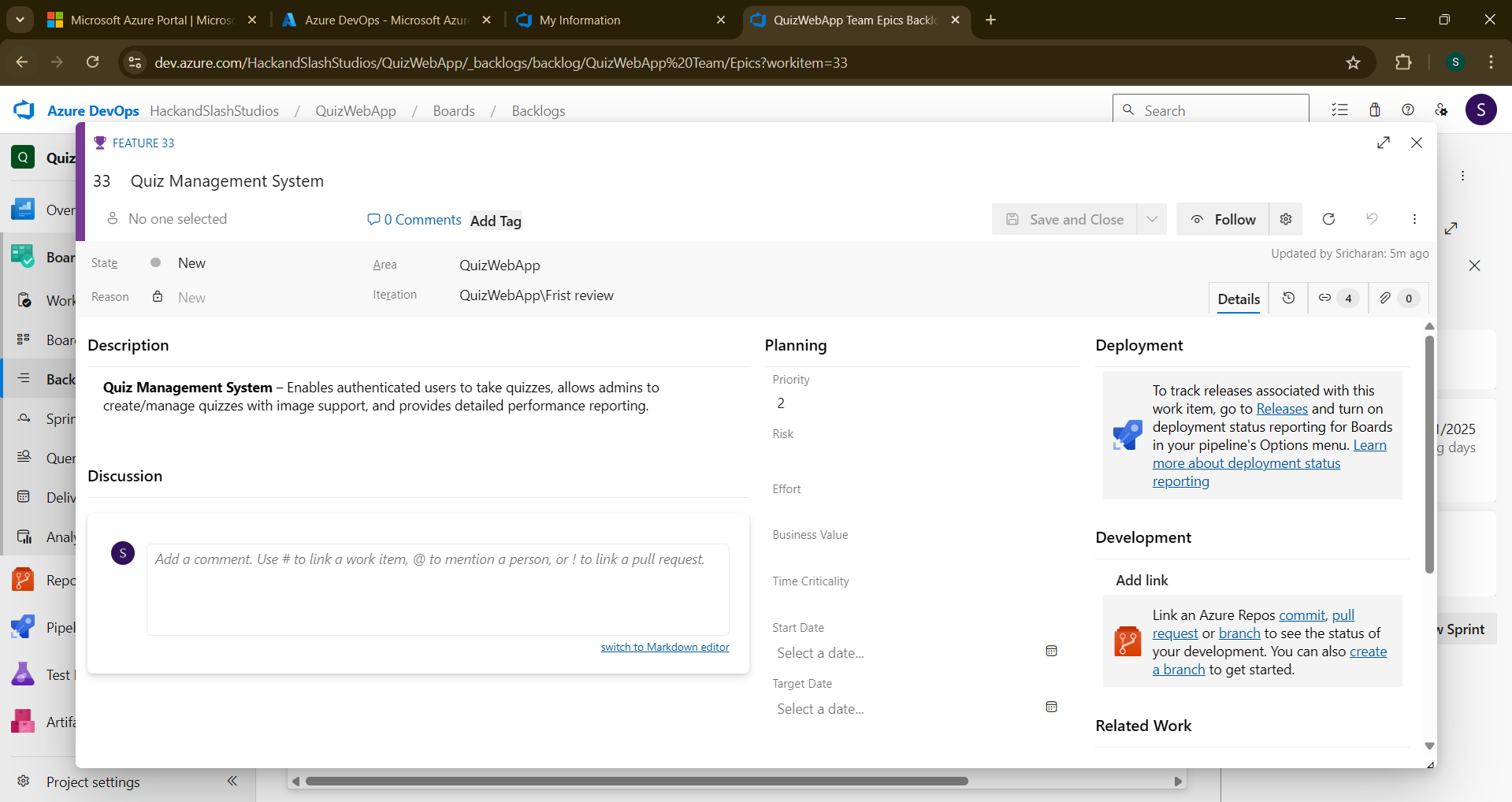
**Create Epic, Features, User Stories, Task**

****

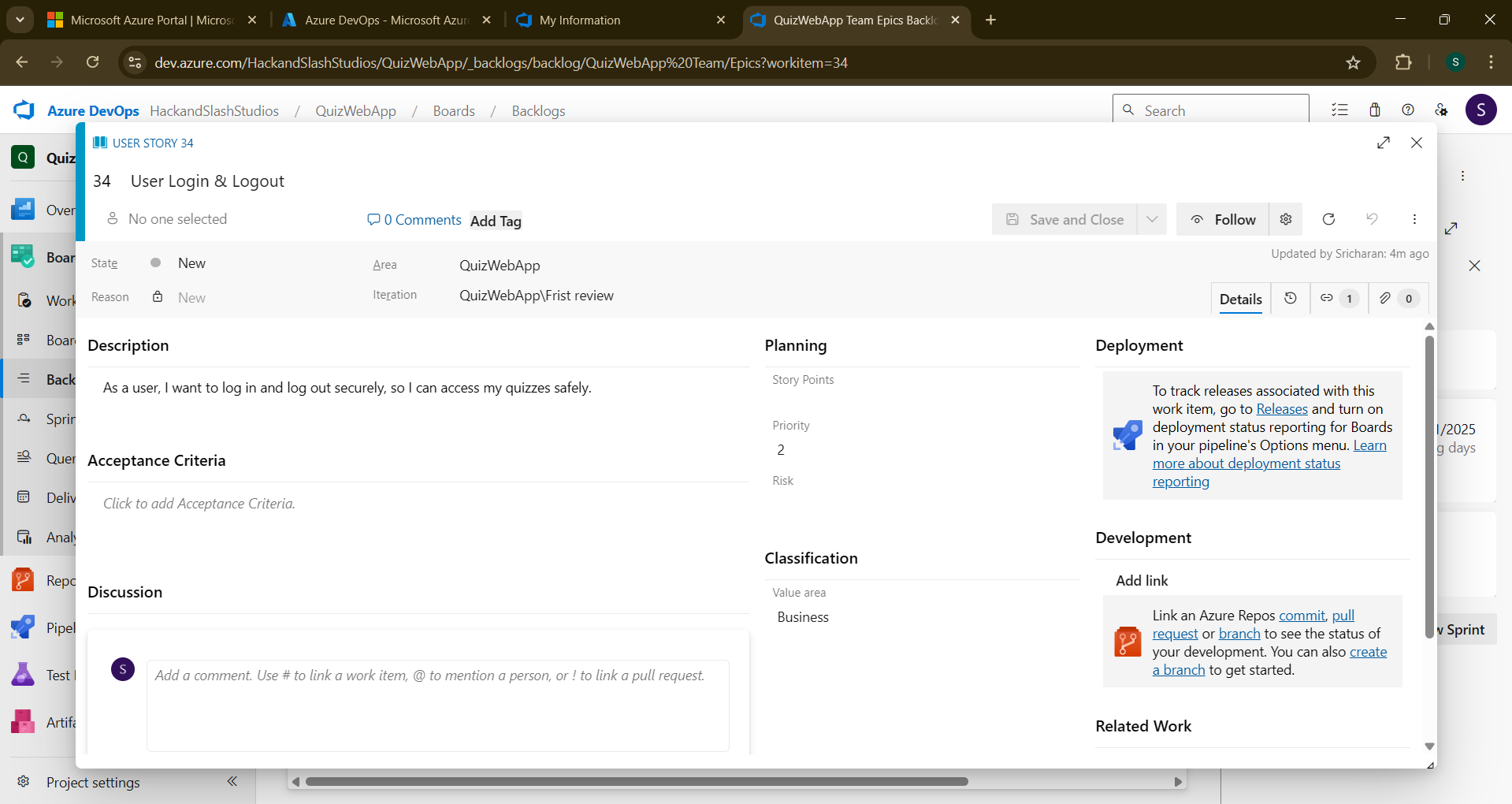
**1.Fill in Epics**

****

**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.**

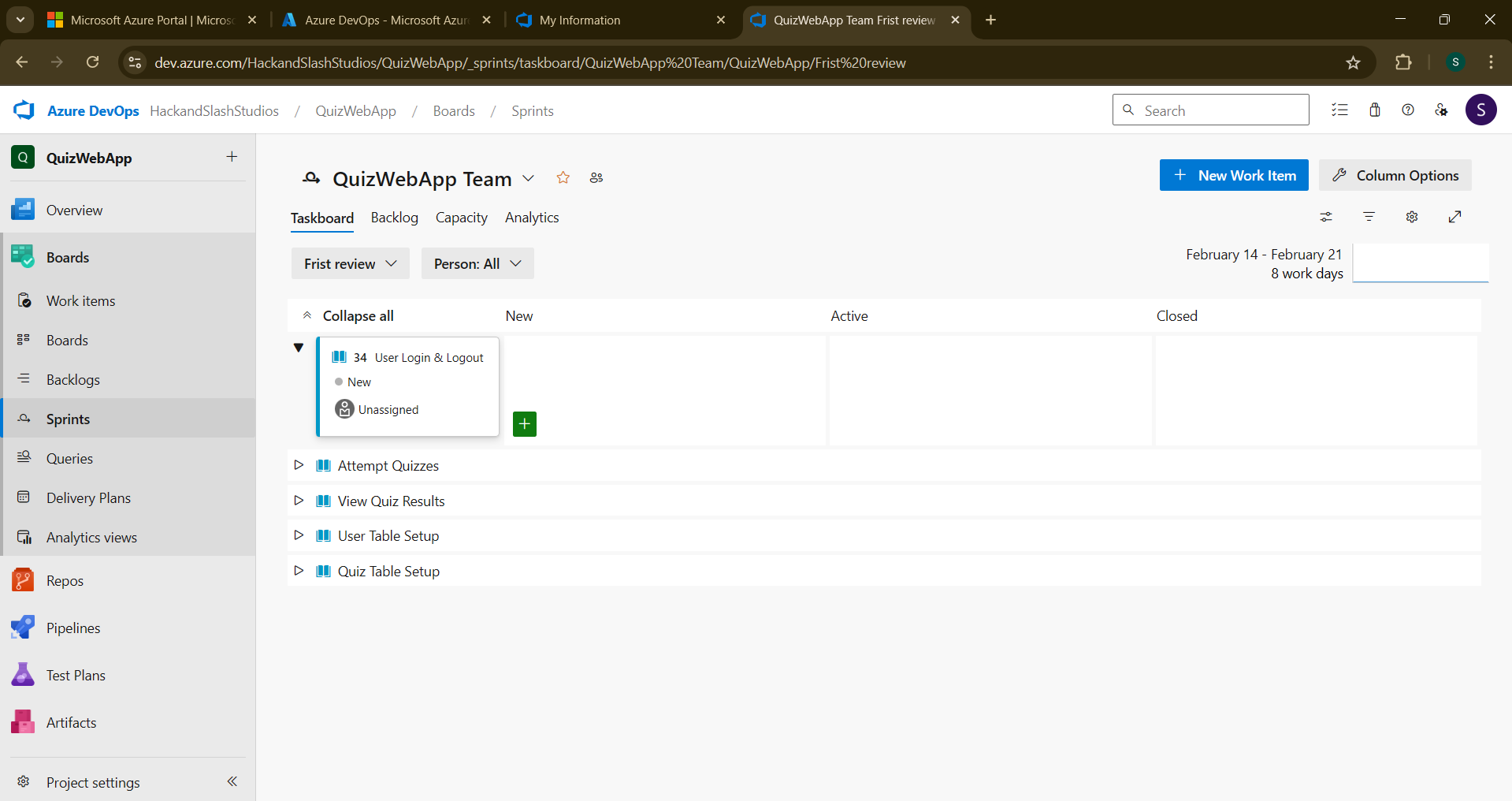
|  |  |
| --- | --- |
| **EXP NO: 4** | **SPRINT PLANNING** |

**Aim:**

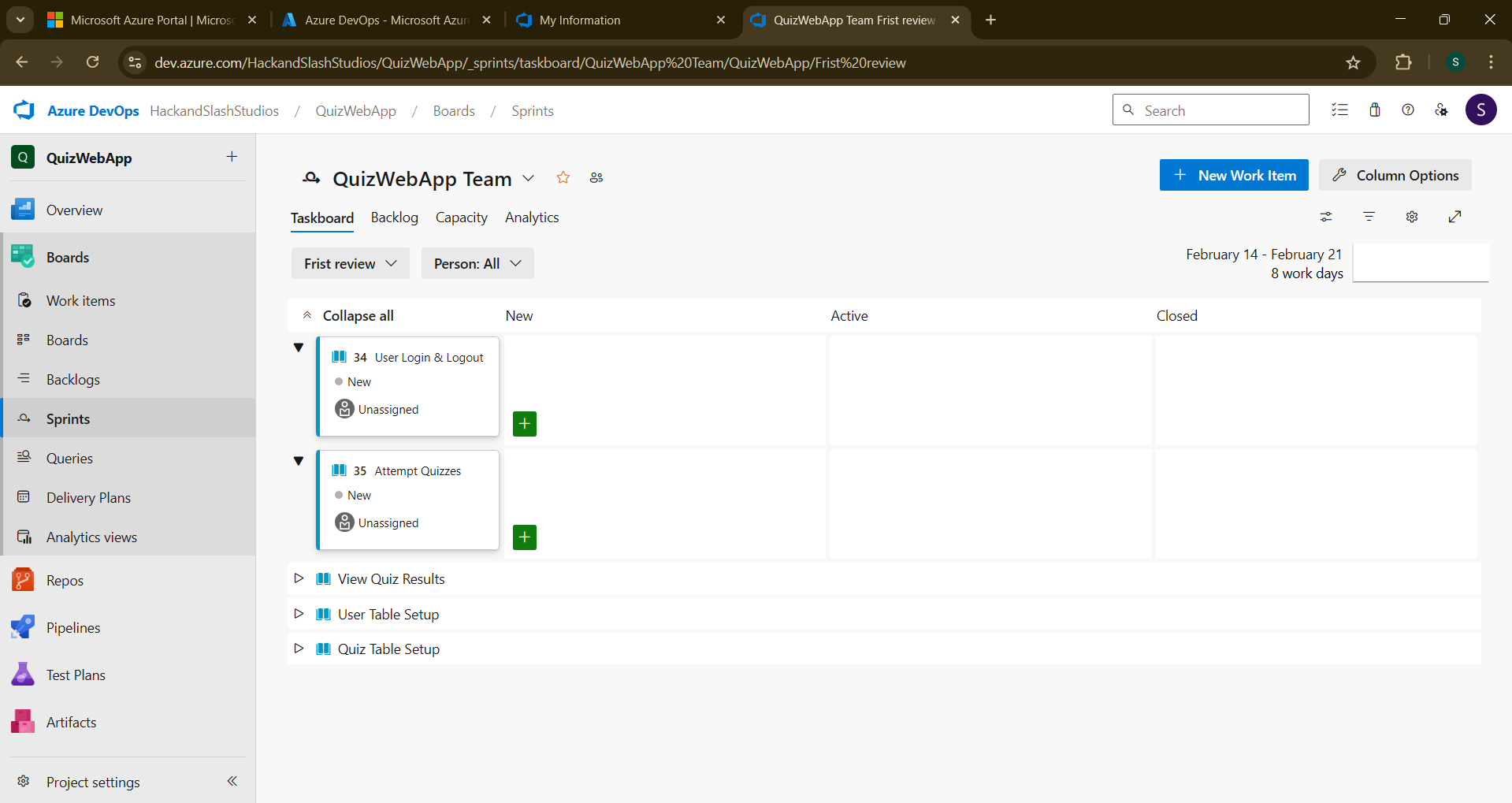
**To assign user story to specific sprint for the Music Playlist Batch Creator Project.**

**Sprint Planning**

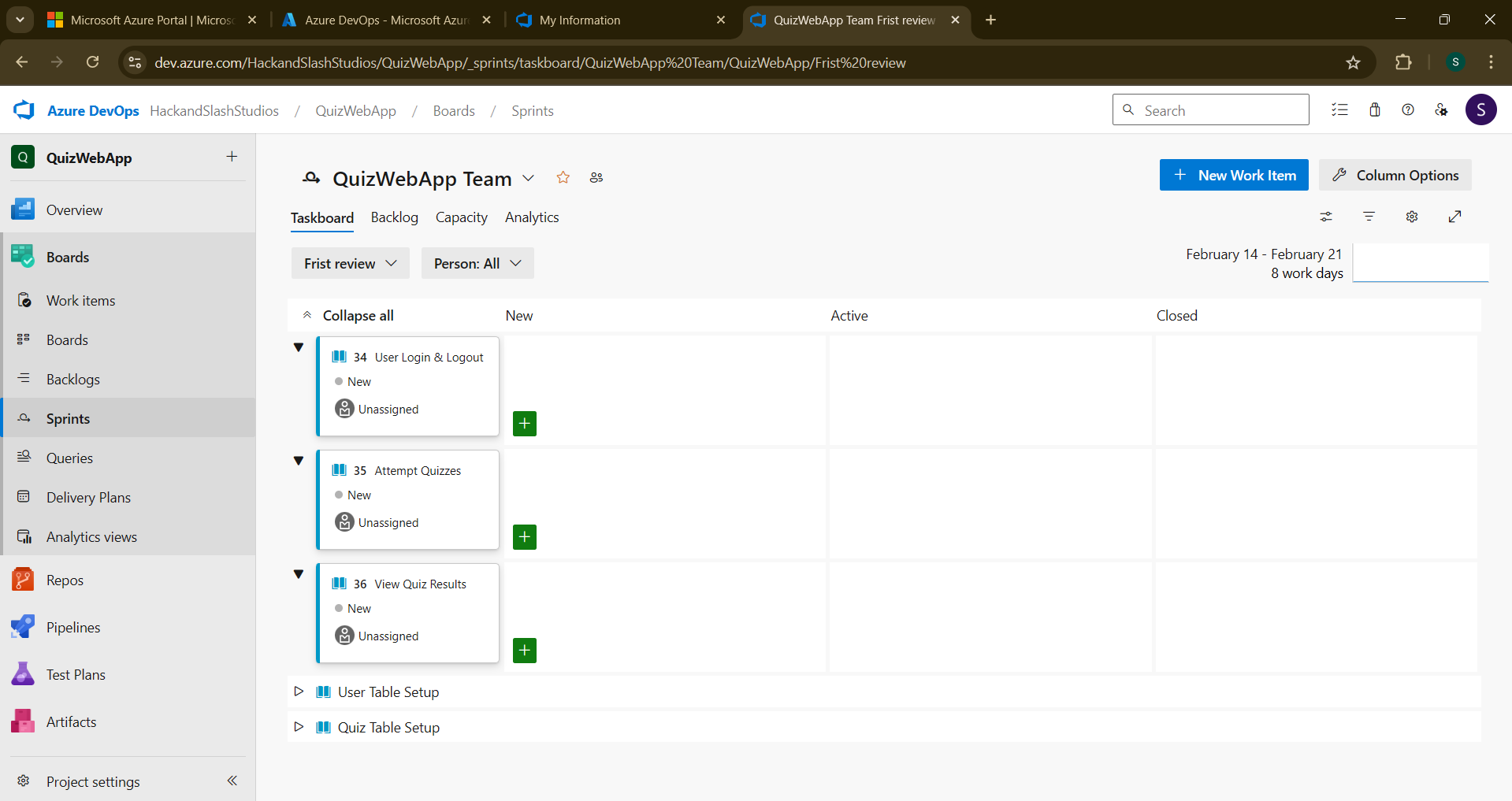
**Sprint 1**

****

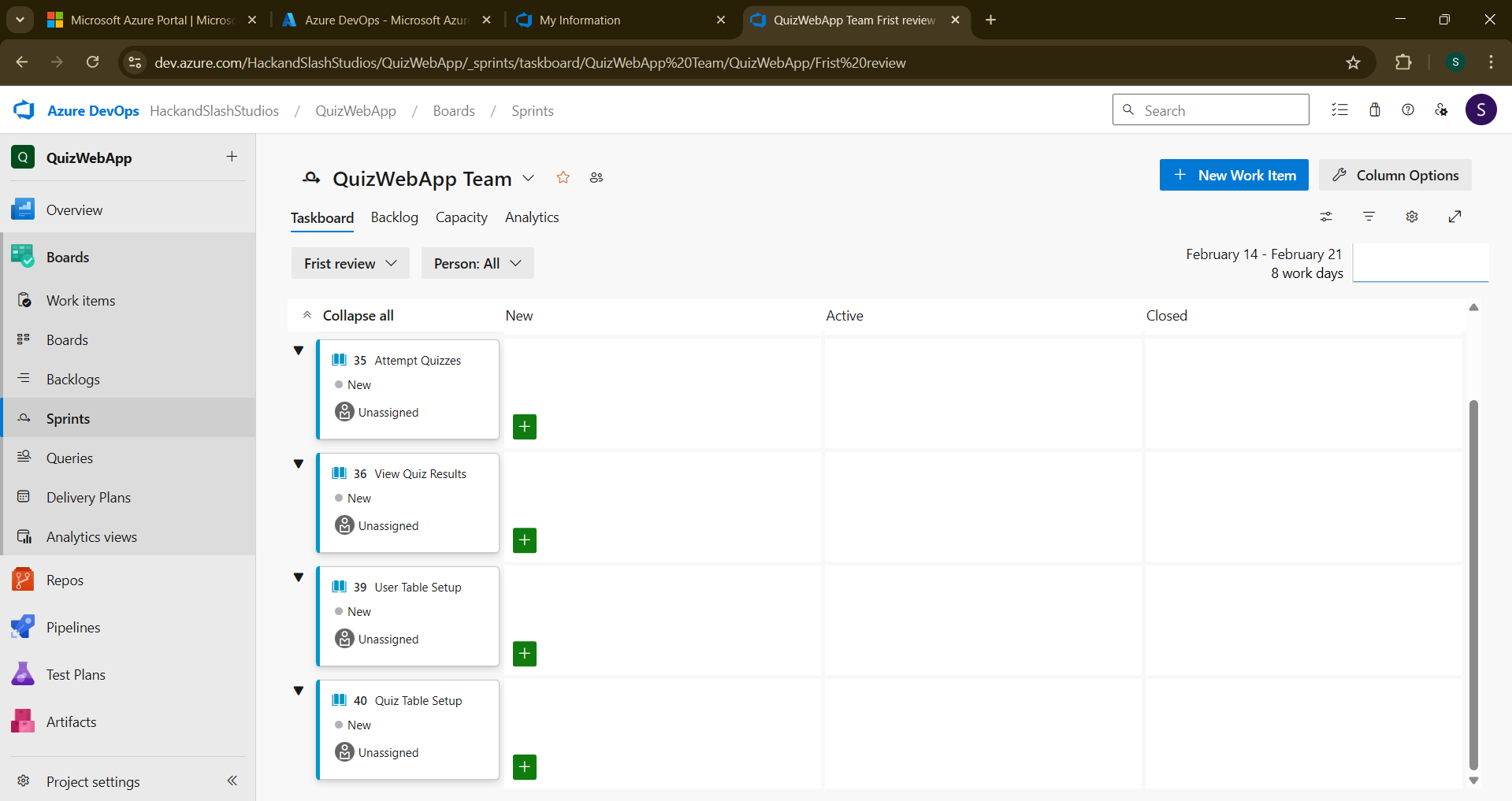
**Sprint 2**

****

**Sprint 3**

****

**Sprint 4**

****

**Result:**

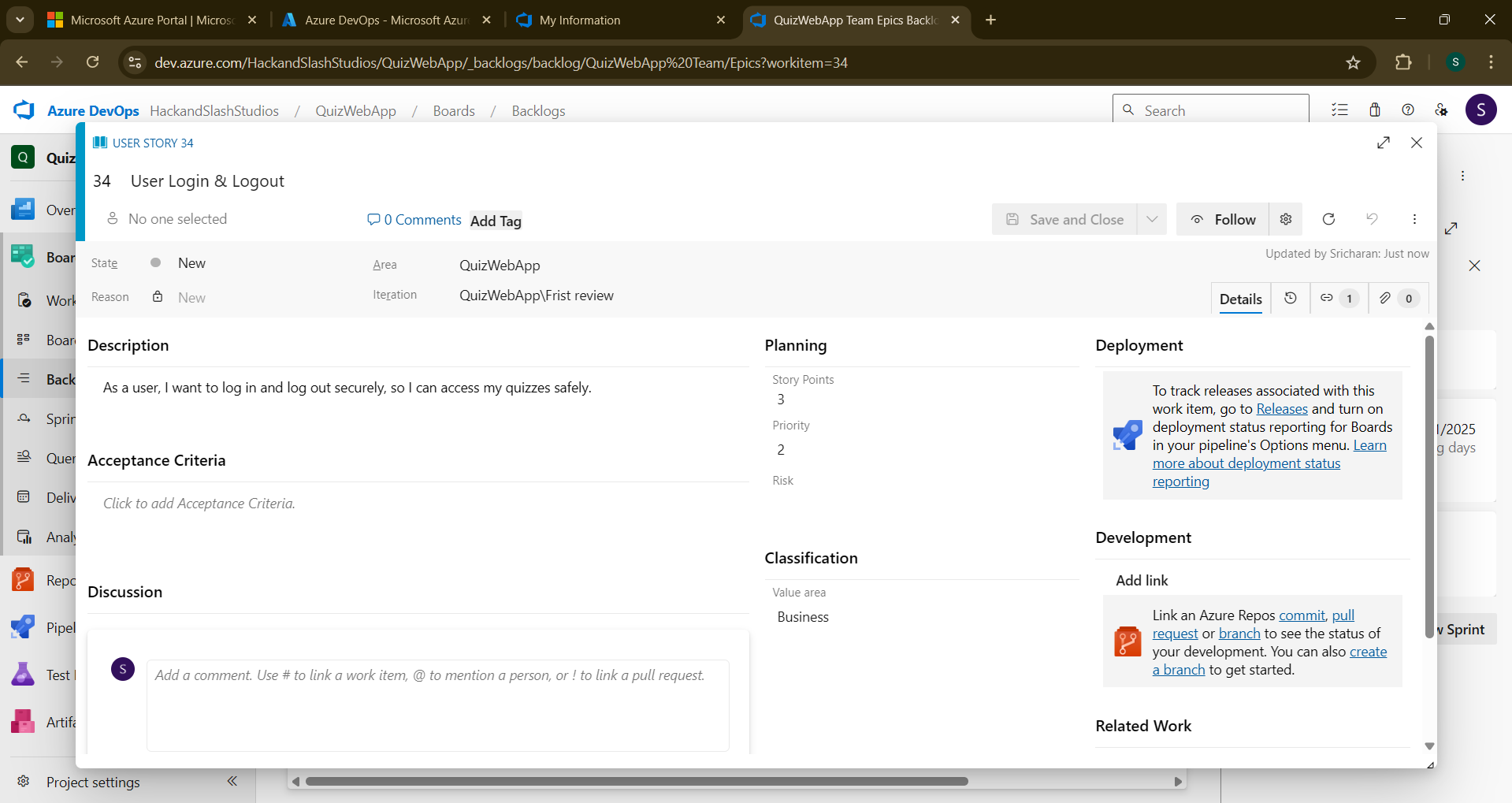
**The Sprints are created for the Music Playlist Batch Creator Project.**

|  |  |
| --- | --- |
| **EXP NO: 5** | **POKER ESTIMATION** |

**Aim:**

Create Poker Estimation for the user stories - Music Playlist Batch Creator Project.

**Poker Estimation**

****

**Result:**

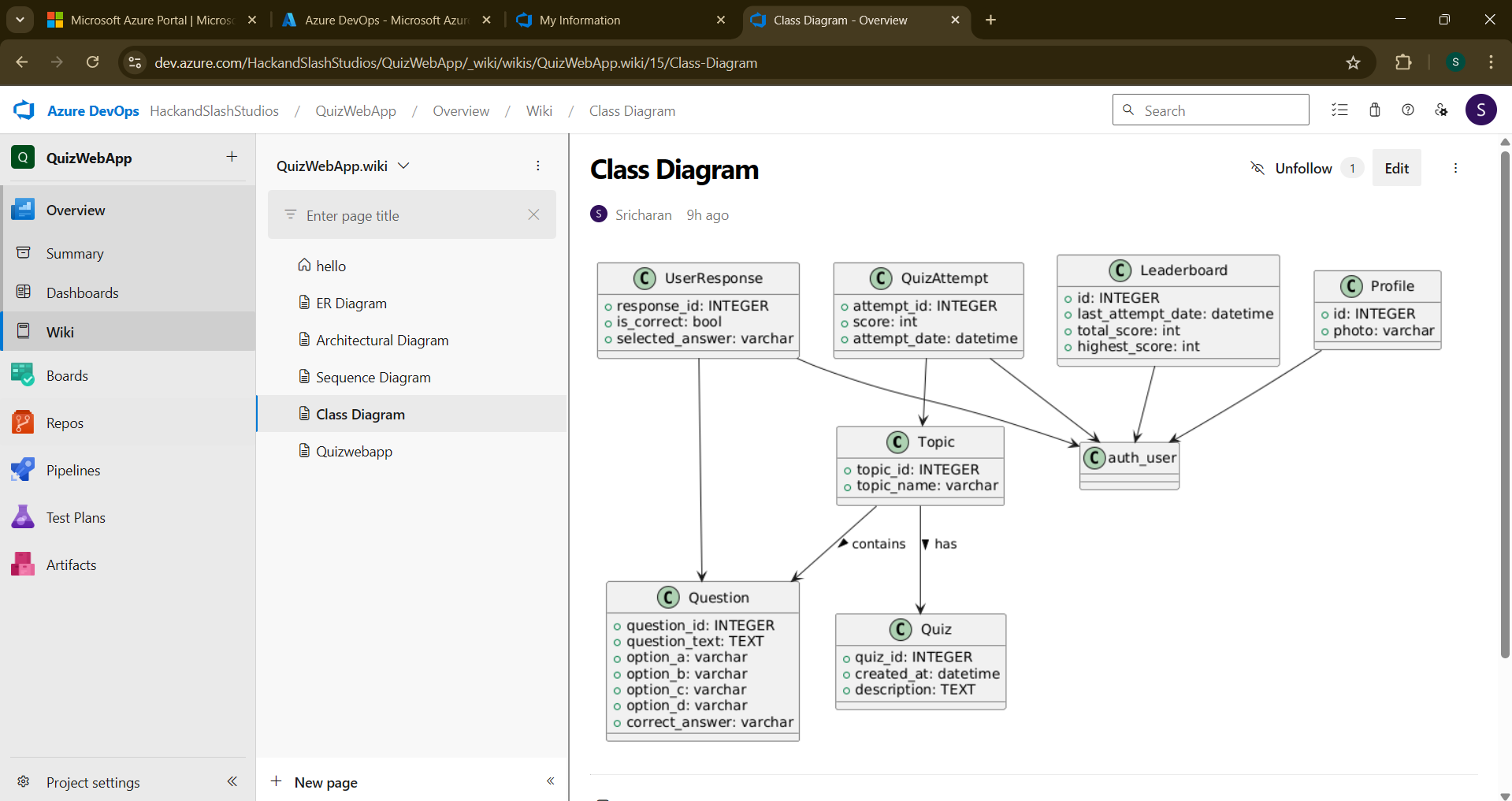
**The Estimation/Story Points is created for the project using Poker Estimation.**

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

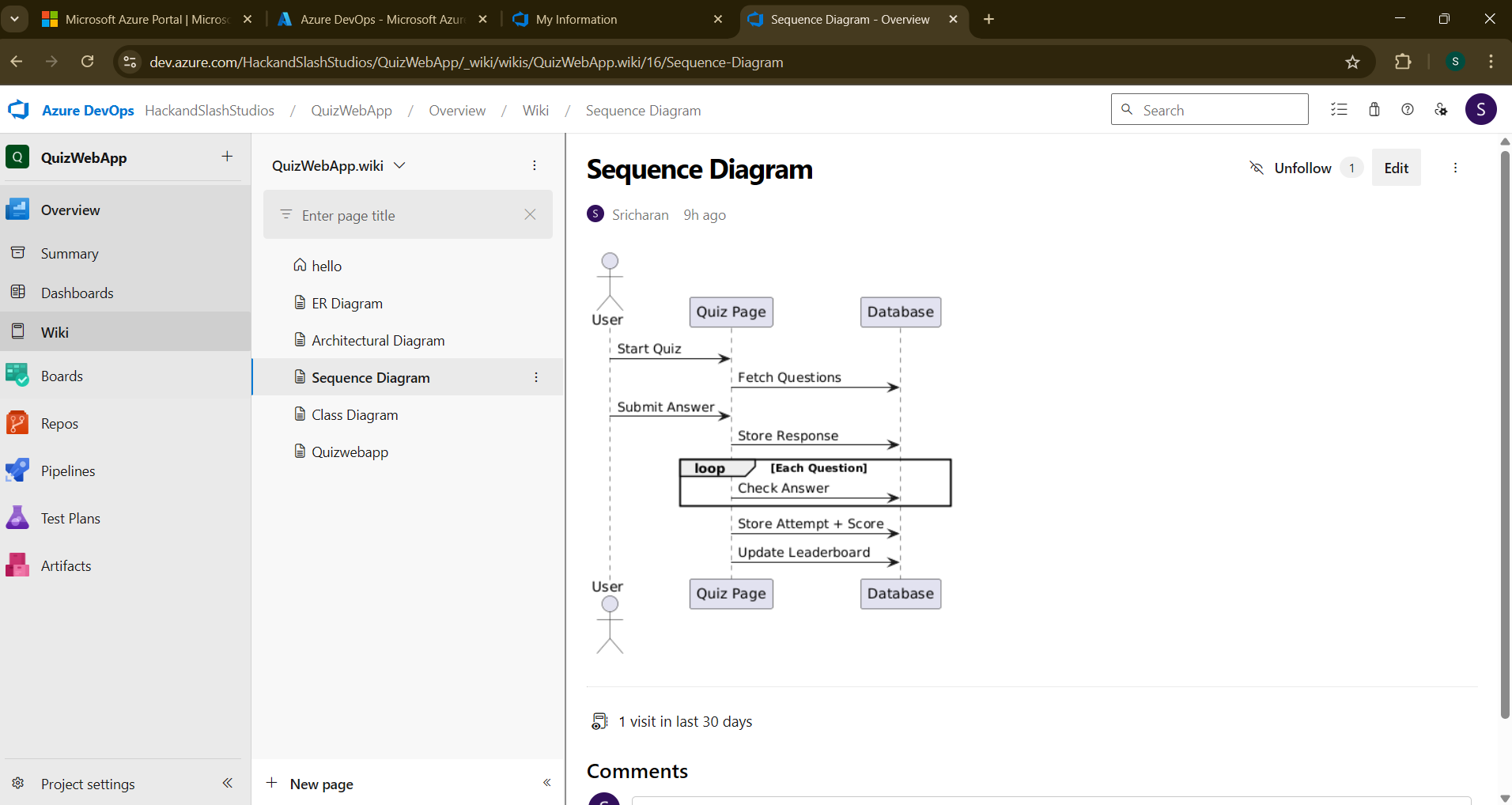
**Aim:**

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

**6A. Class Diagram**

****

**6B. Sequence Diagram**

****

**Result:**

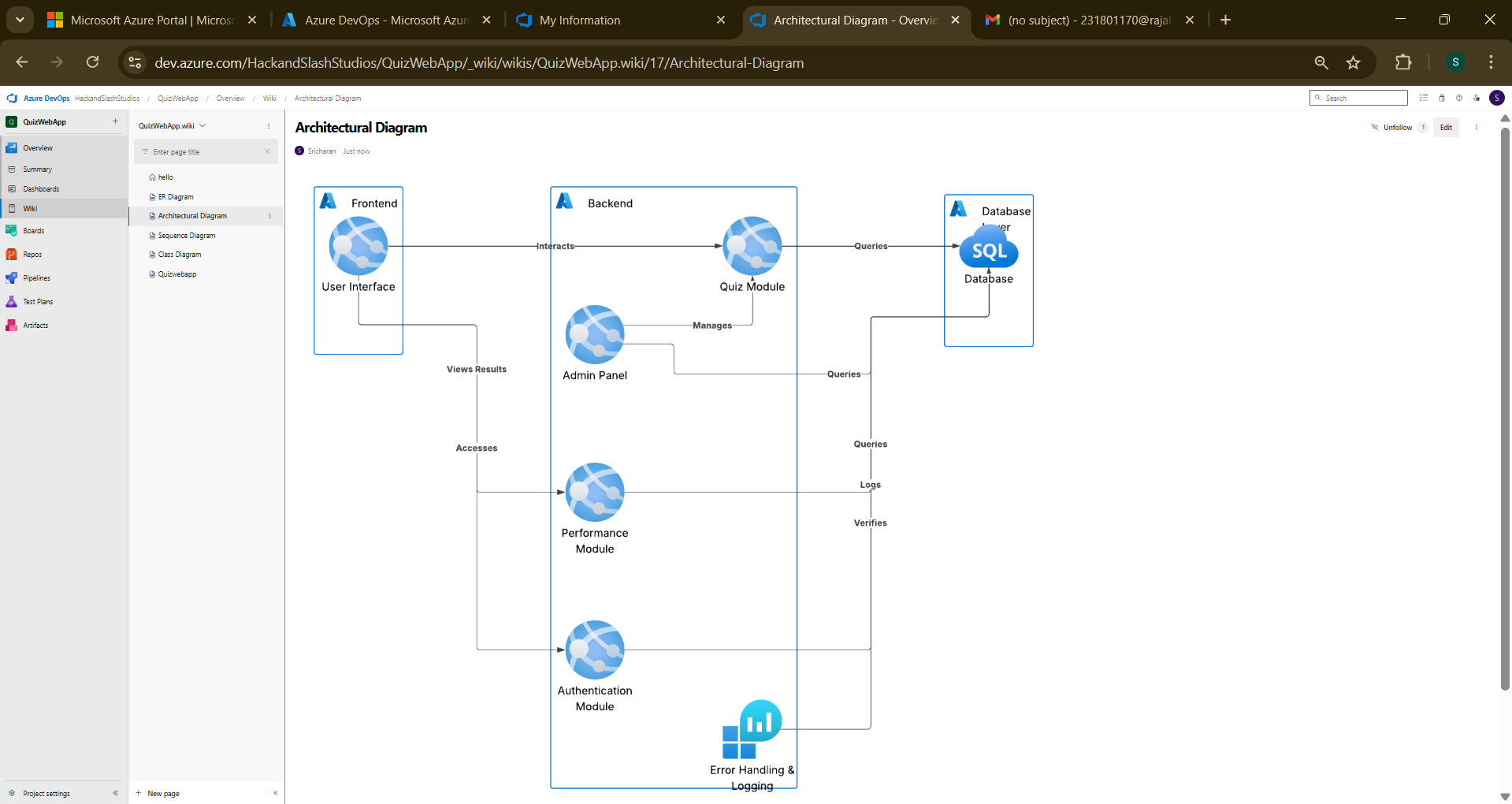
The Class Diagram and Sequence Diagram is designed Successfully for the Music Playlist Batch Creator.

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

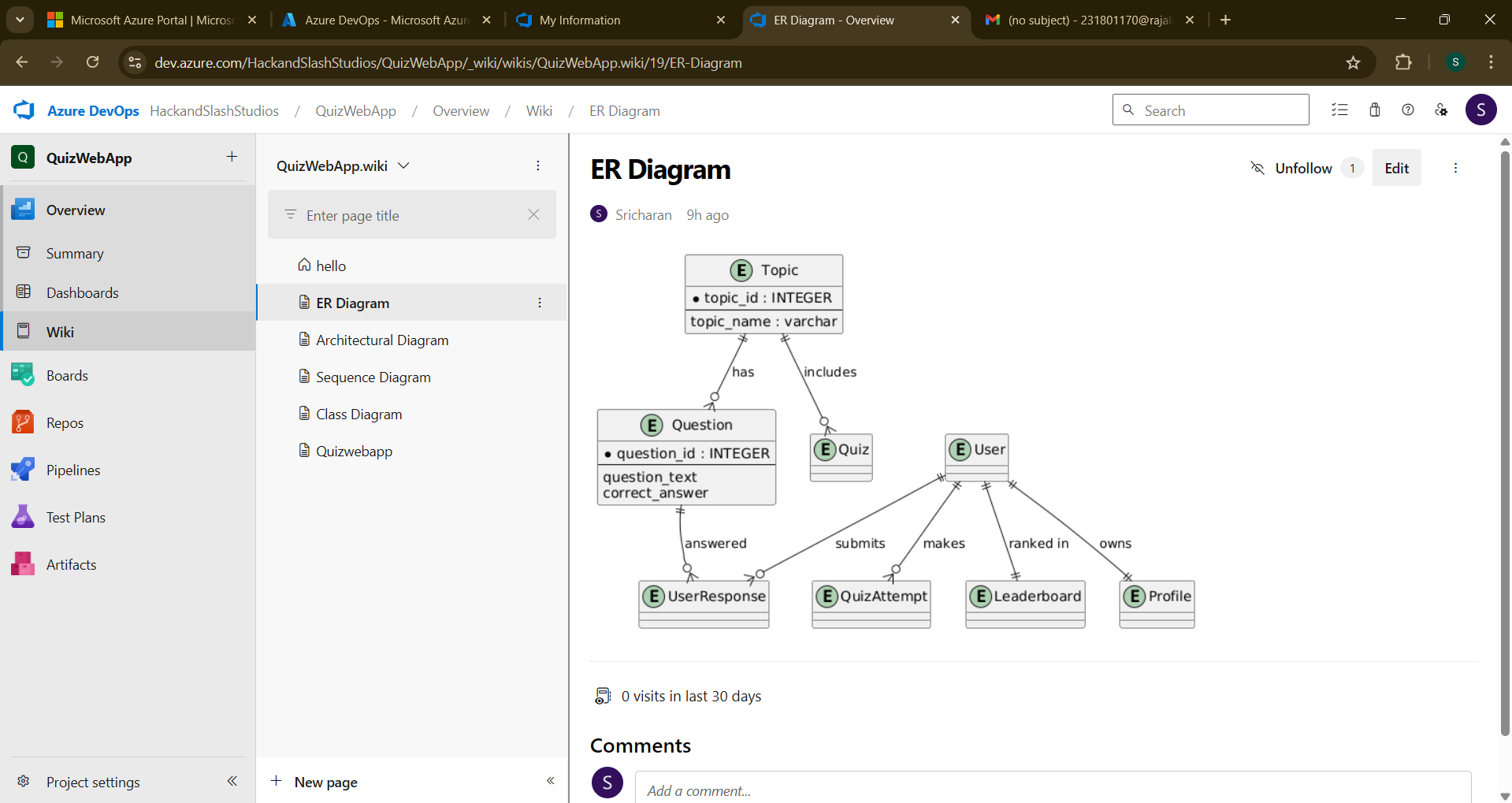
**Aim:**

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 Music Playlist Batch Creator

|  |  |
| --- | --- |
| **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**

o User Signup & Login

o Viewing and Managing Playlists

o Fetching Real-time Metadata

o Editing playlists (rename, reorder, record)

o Creating smart audio playlists based on categories (mood, genre, artist, etc.)

**2. Define User Interactions**

o Each test case simulates a real user behaviour (e.g., logging in, renaming a playlist, adding a song).

**3. Design Happy Path Test Cases**

o Focused on validating that all features function as expected under normal conditions.

o Example: User logs in successfully, adds item to playlist, or creates a category- based playlist.

**4. Design Error Path Test Cases**

o Simulate negative or unexpected scenarios to test robustness and error handling.

o Example: Login fails with invalid credentials, save fails when offline, no recommendations found.

**5. Break Down Steps and Expected Results**

o Each test case contains step-by-step actions and a corresponding expected outcome.

o Ensures clarity for both testers and automation scripts.

**6. Use Clear Naming and IDs**

o Test cases are named clearly (e.g., TC01 – Successful Login, TC10 – Save Playlist Fails).

o Helps in quick identification and linking to user stories or features.

**7. Separate Test Suites**

o Grouped test cases based on functionality (e.g., Login, Playlist Editing, Recommendation System).

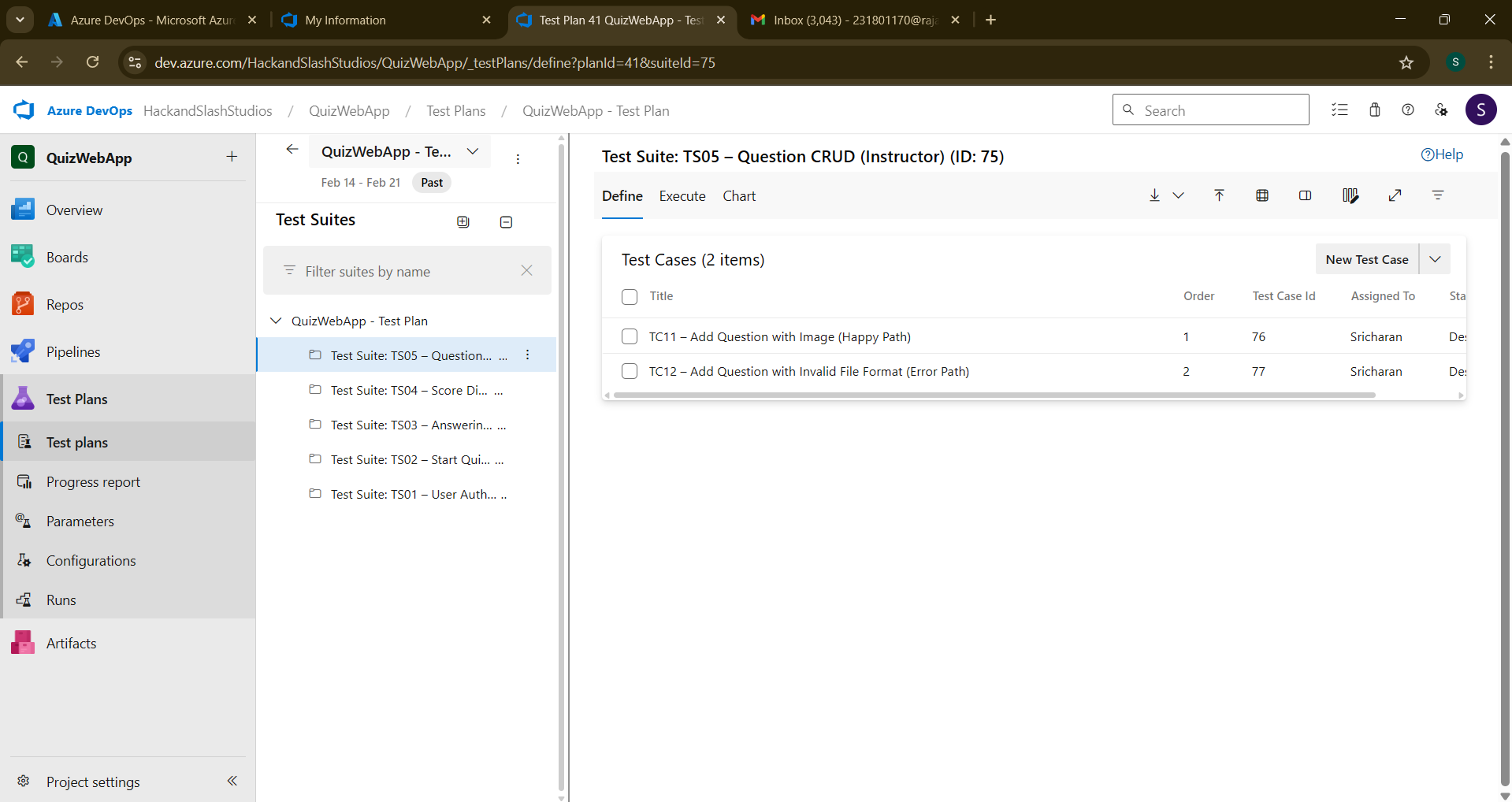
o Improves organization and test execution flow in Azure DevOps.

**8. Prioritize and Review**

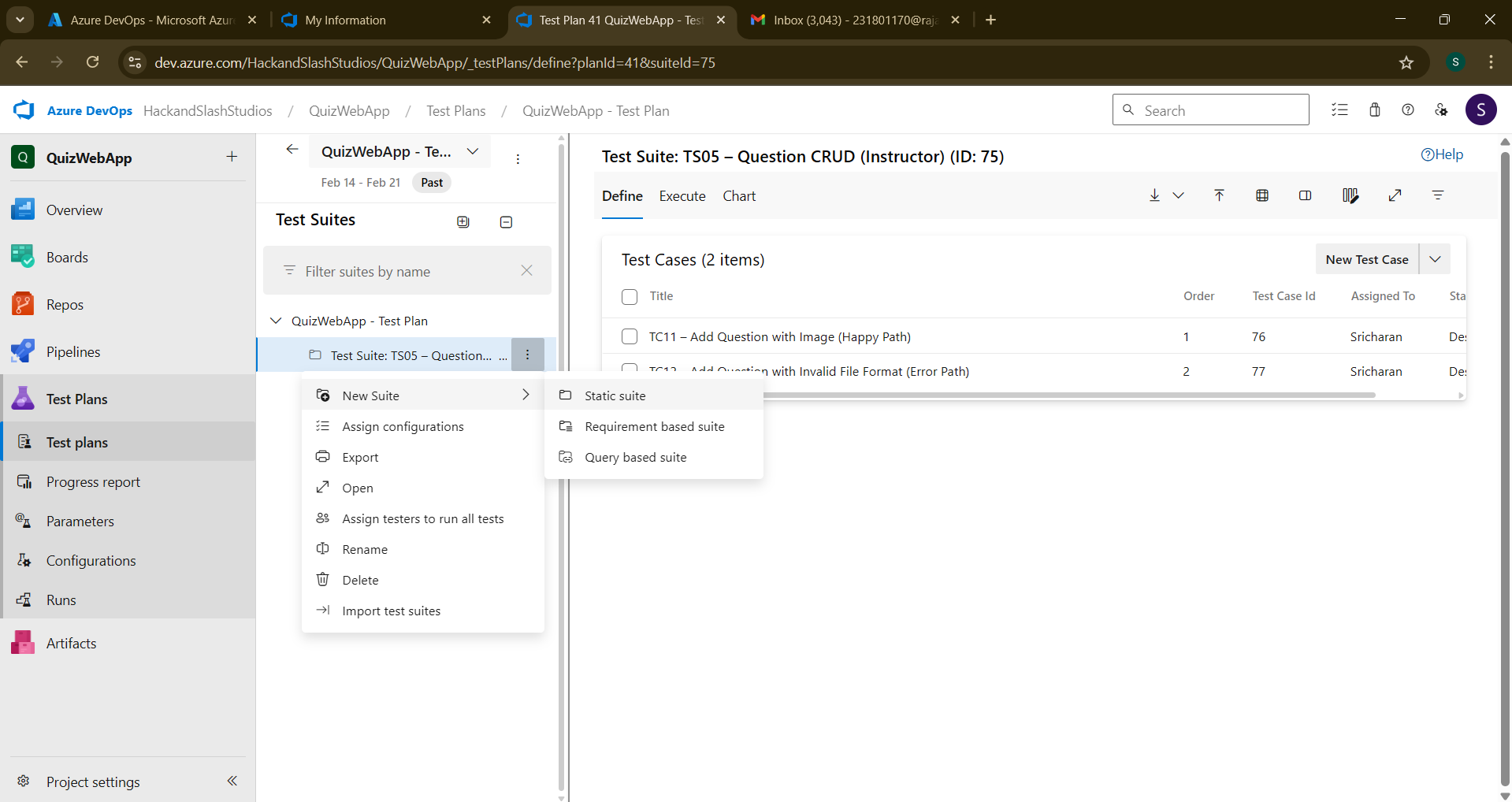
o Critical user actions are marked high-priority.

o 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.

Online Quiz System – Test Plans

USER STORIES  
• US01: As a user, I want to sign up and log in securely so I can access quizzes.  
• US02: As a student, I want to select a quiz topic and start a quiz.  
• US03: As a user, I want to answer different types of questions (MCQ, True/False, Fill in the Blank).  
• US04: As a student, I want to see my quiz score after completion.  
• US05: As an instructor, I want to create, update, and delete questions with image support.

Test Suite: TS01 – User Authentication (Sign Up & Login)

TC01 – Successful Sign Up (Happy Path)  
Action:  
• Navigate to Sign-Up page  
• Enter valid name, email, and password  
• Click "Sign Up"  
Expected Result:  
• Account created, redirected to dashboard

TC02 – Sign Up with Existing Email (Error Path)  
Action:  
• Enter already registered email  
Expected Result:  
• Error message: “Email already registered”

TC03 – Login with Correct Credentials (Happy Path)  
Action:  
• Enter valid email and password  
Expected Result:  
• Redirected to dashboard

TC04 – Login with Wrong Password (Error Path)  
Action:  
• Enter valid email and wrong password  
Expected Result:  
• Error: “Invalid credentials”

Test Suite: TS02 – Start Quiz Flow

TC05 – Topic Selection and Start Quiz (Happy Path)  
Action:  
• Click on a topic card (e.g., Science)  
• Click "Start Quiz"  
Expected Result:  
• Quiz starts, question panel visible

TC06 – Start Quiz Without Selecting Topic (Error Path)  
Action:  
• Click "Start Quiz" without selecting a topic  
Expected Result:  
• Alert: “Please select a topic first”

Test Suite: TS03 – Answering Questions

TC07 – Submit Valid MCQ Answer (Happy Path)  
Action:  
• Select a valid option  
• Click "Next"  
Expected Result:  
• Answer recorded, next question shown

TC08 – Submit Blank for Fill in the Blank (Error Path)  
Action:  
• Leave input field blank and click “Next”  
Expected Result:  
• Error: “Answer cannot be empty”

Test Suite: TS04 – Score Display

TC09 – Score Displayed After Completion (Happy Path)  
Action:  
• Complete all questions  
Expected Result:  
• Score page displays total correct answers

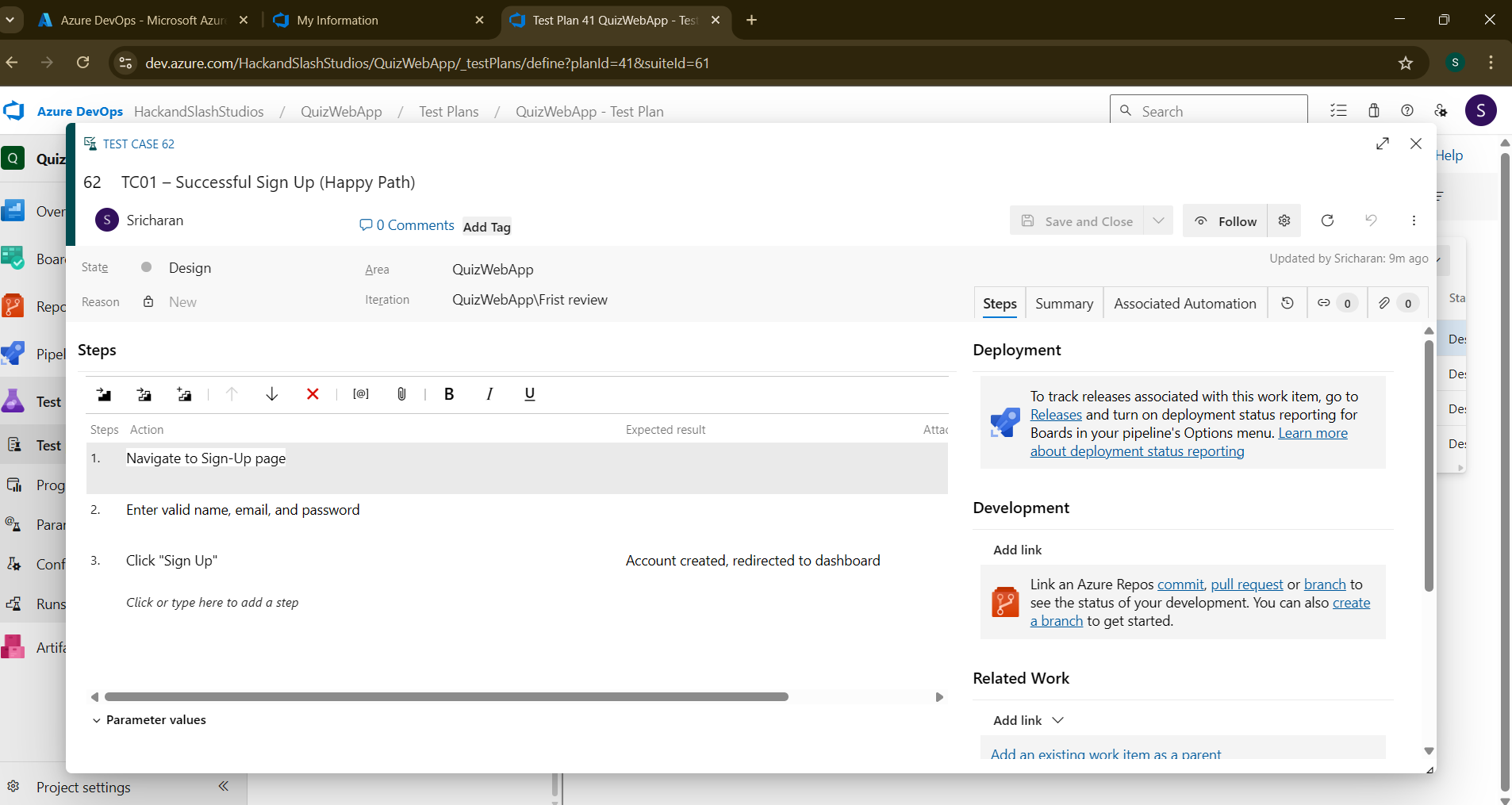
TC10 – Incomplete Quiz Submission (Error Path)  
Action:  
• Try submitting before answering all questions  
Expected Result:  
• Prompt: “Answer all questions before submitting”

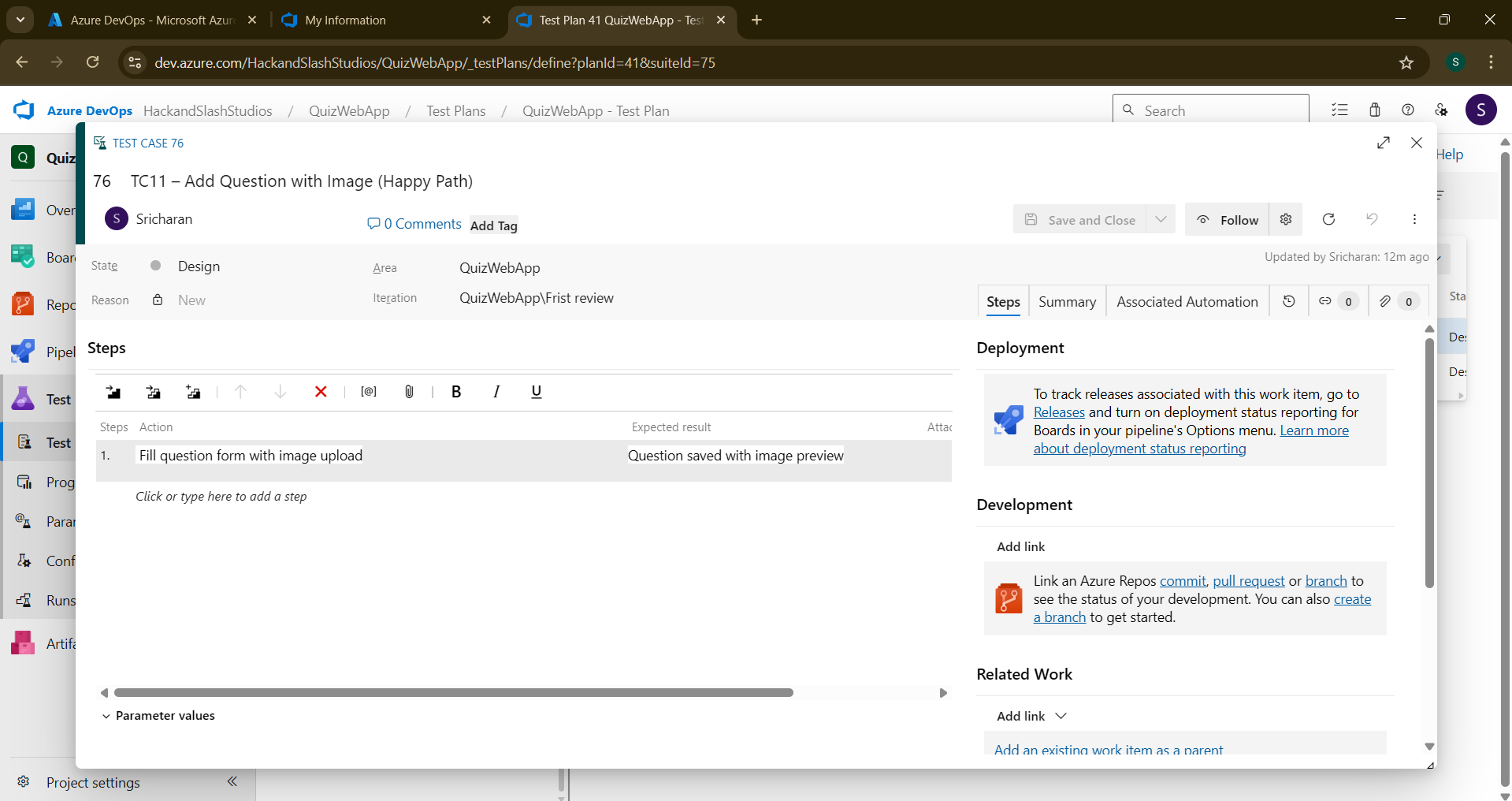
Test Suite: TS05 – Question CRUD (Instructor)

TC11 – Add Question with Image (Happy Path)  
Action:  
• Fill question form with image upload  
Expected Result:  
• Question saved with image preview

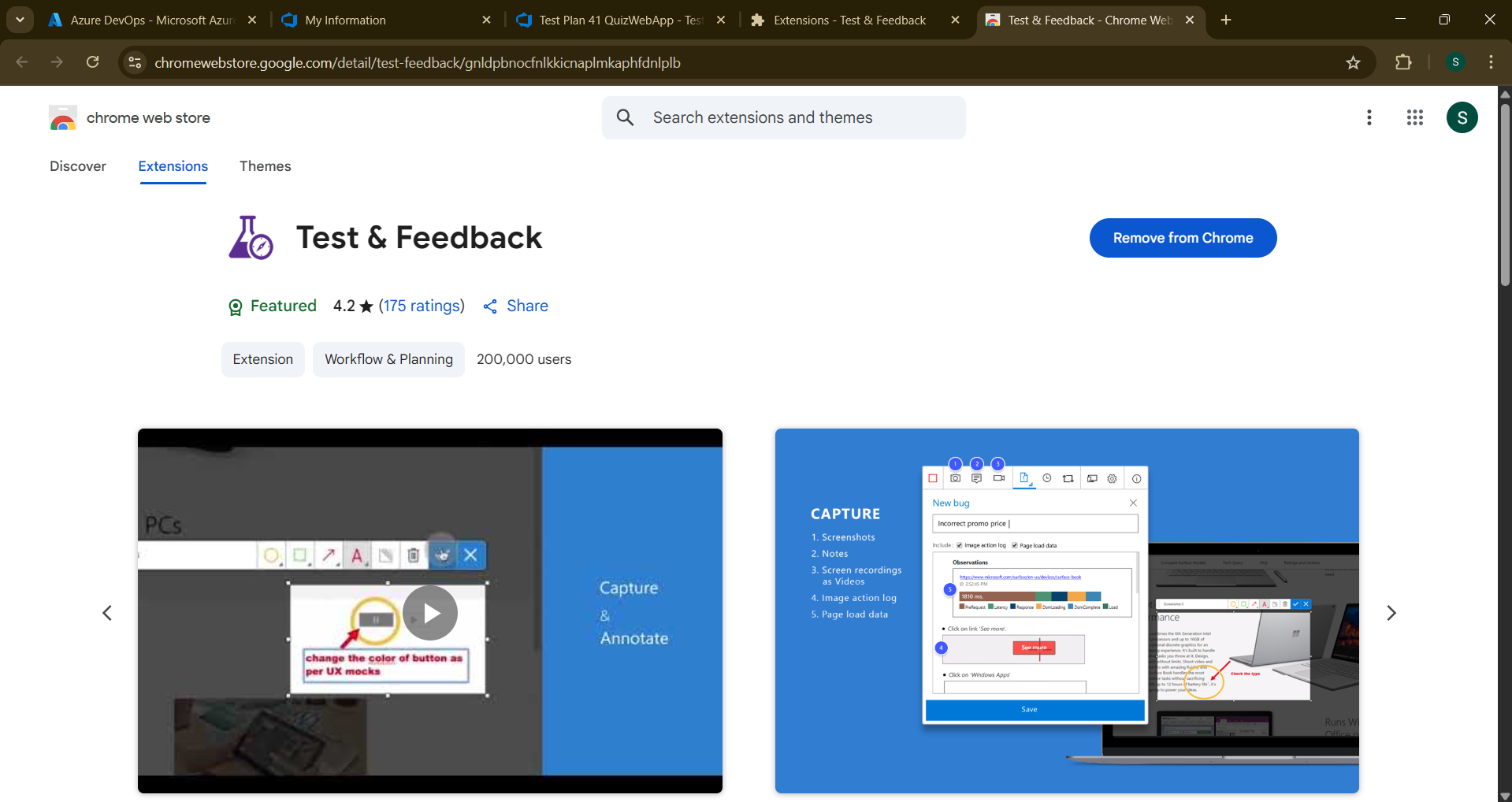
TC12 – Add Question with Invalid File Format (Error Path)  
Action:  
• Upload .exe file as image  
Expected Result:  
• Error: “Unsupported file type”

**Test Cases**

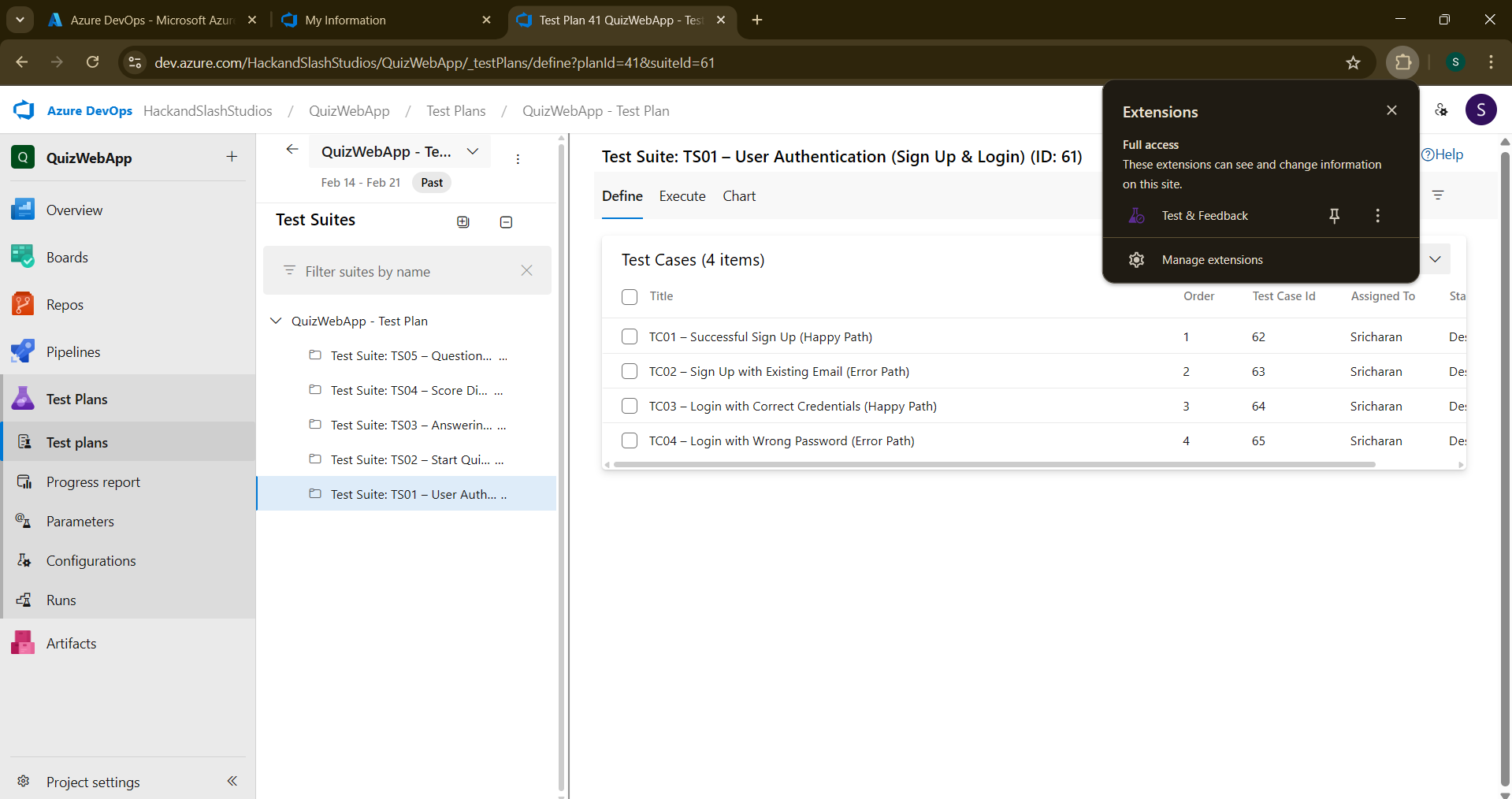
****

****

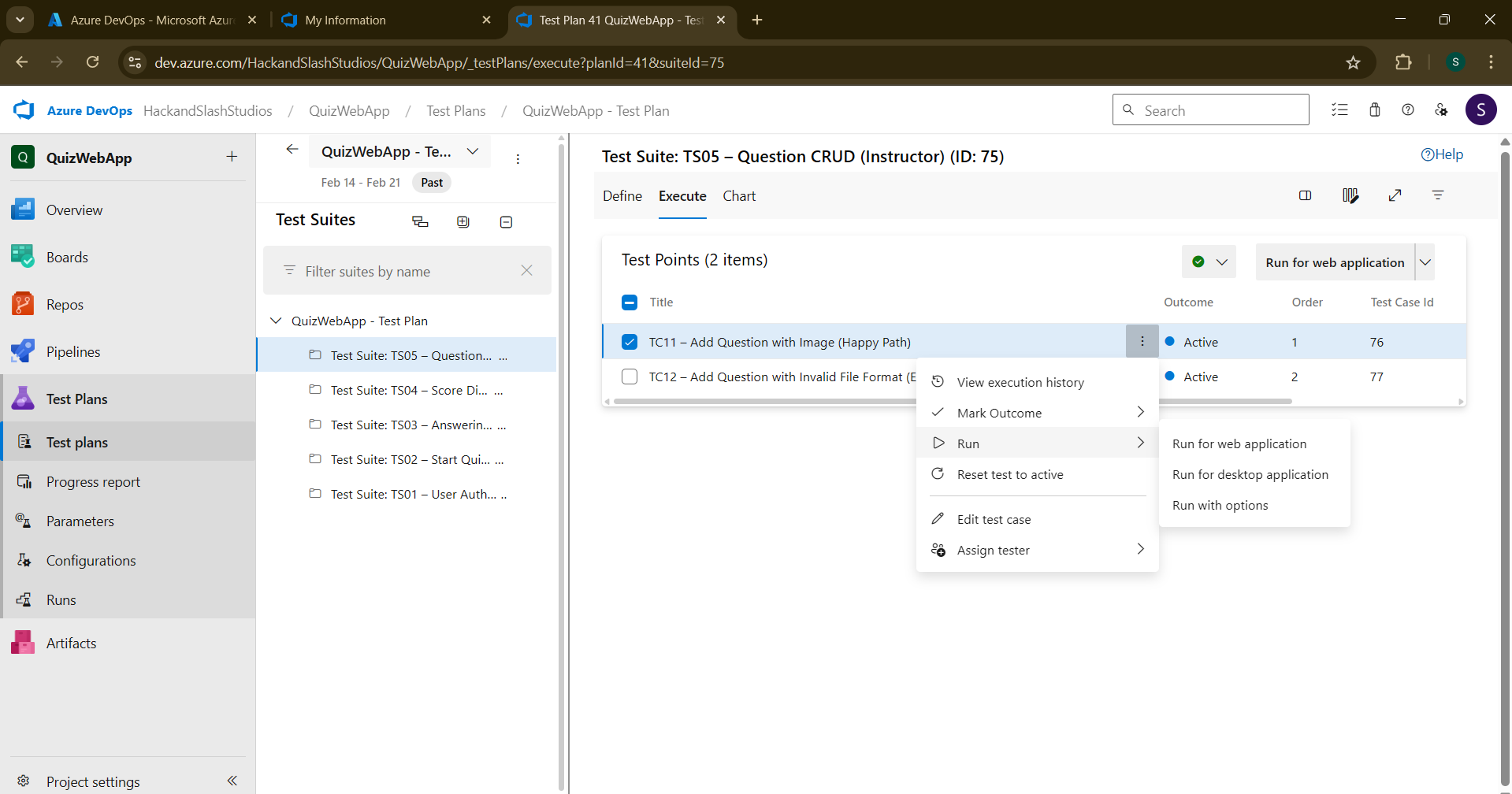
**4.Installation of test**

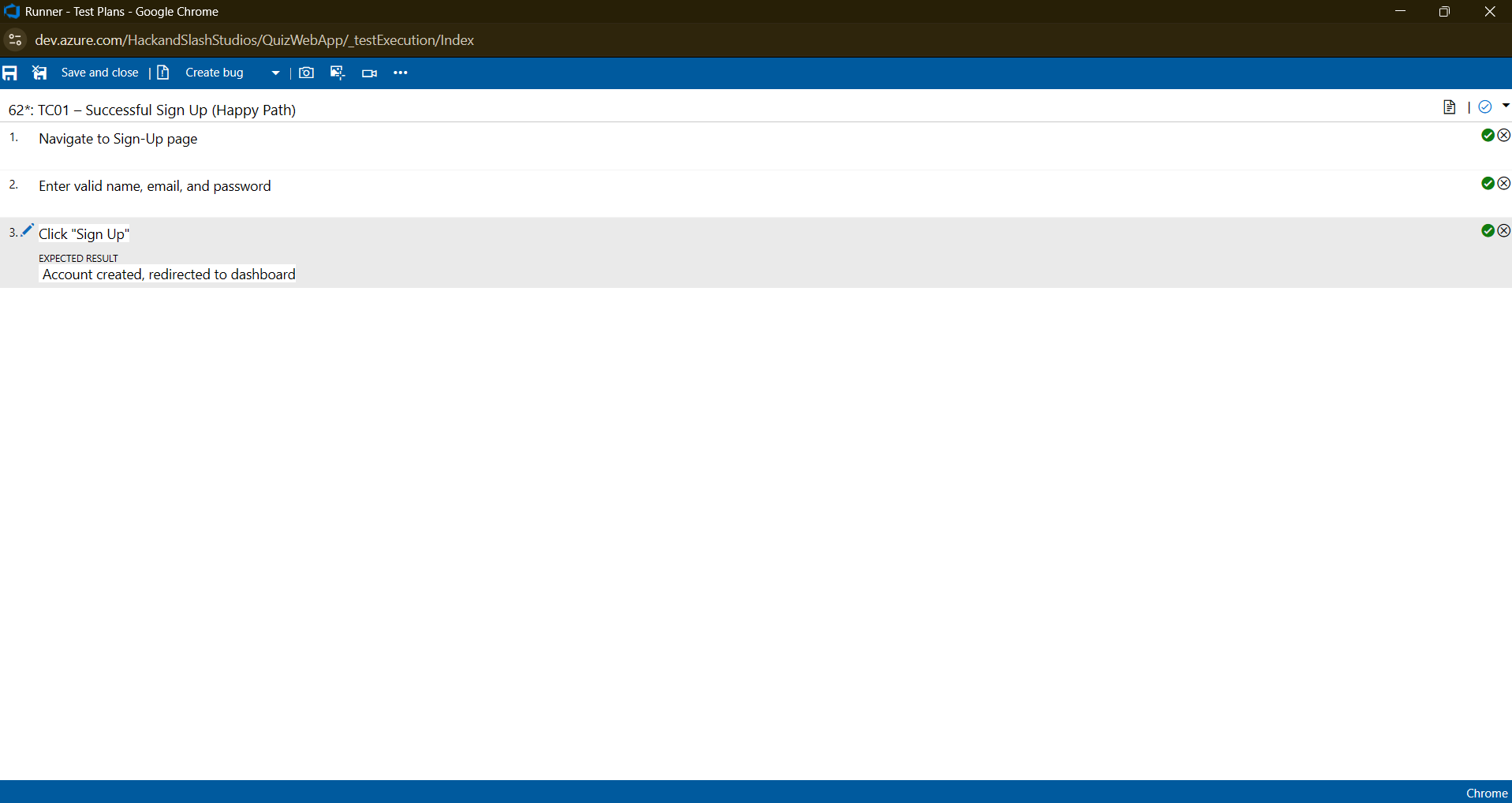
****

**Test and feedback Showing it as an extension**

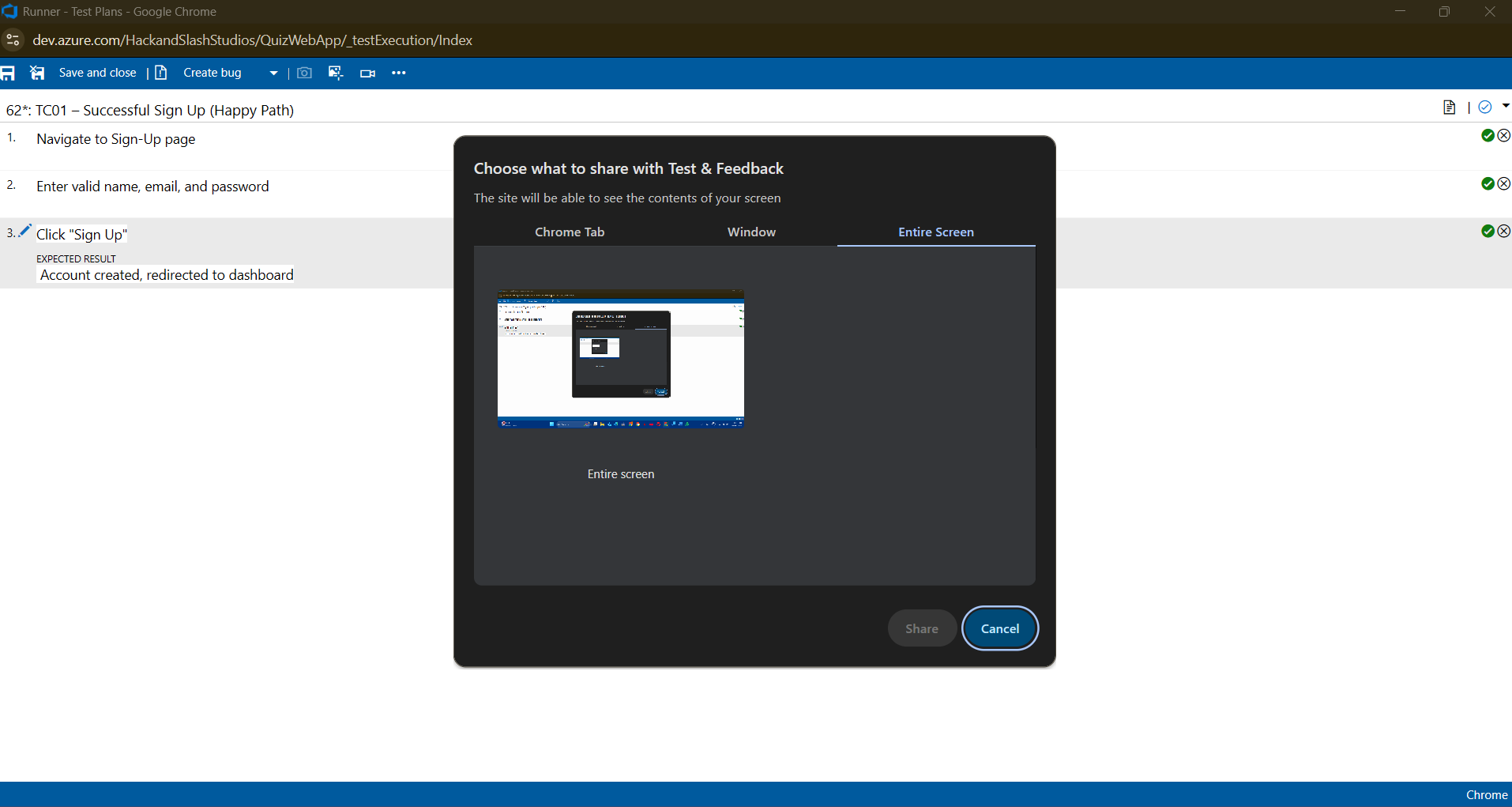
****

**5.Running the test cases**

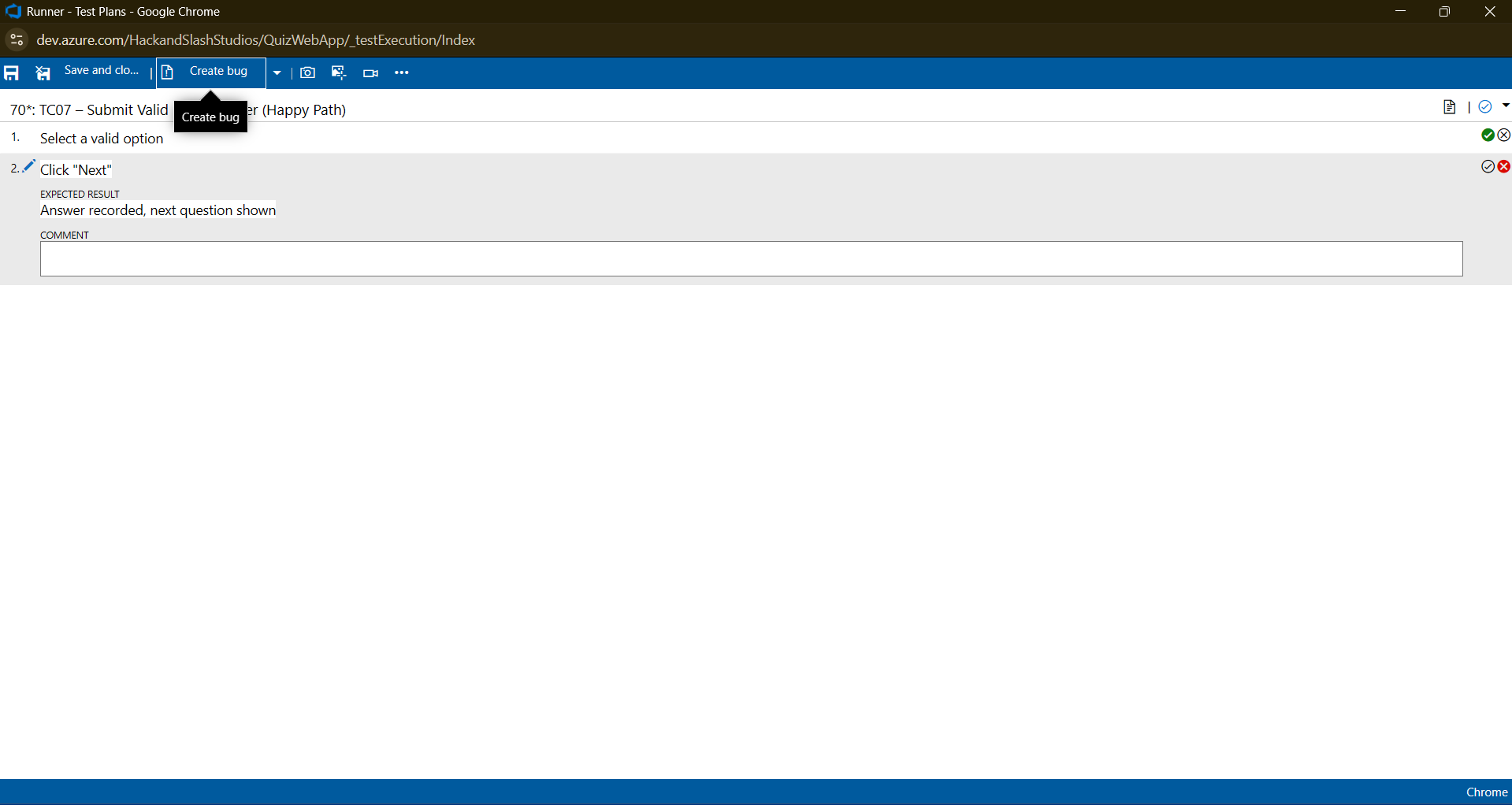
****

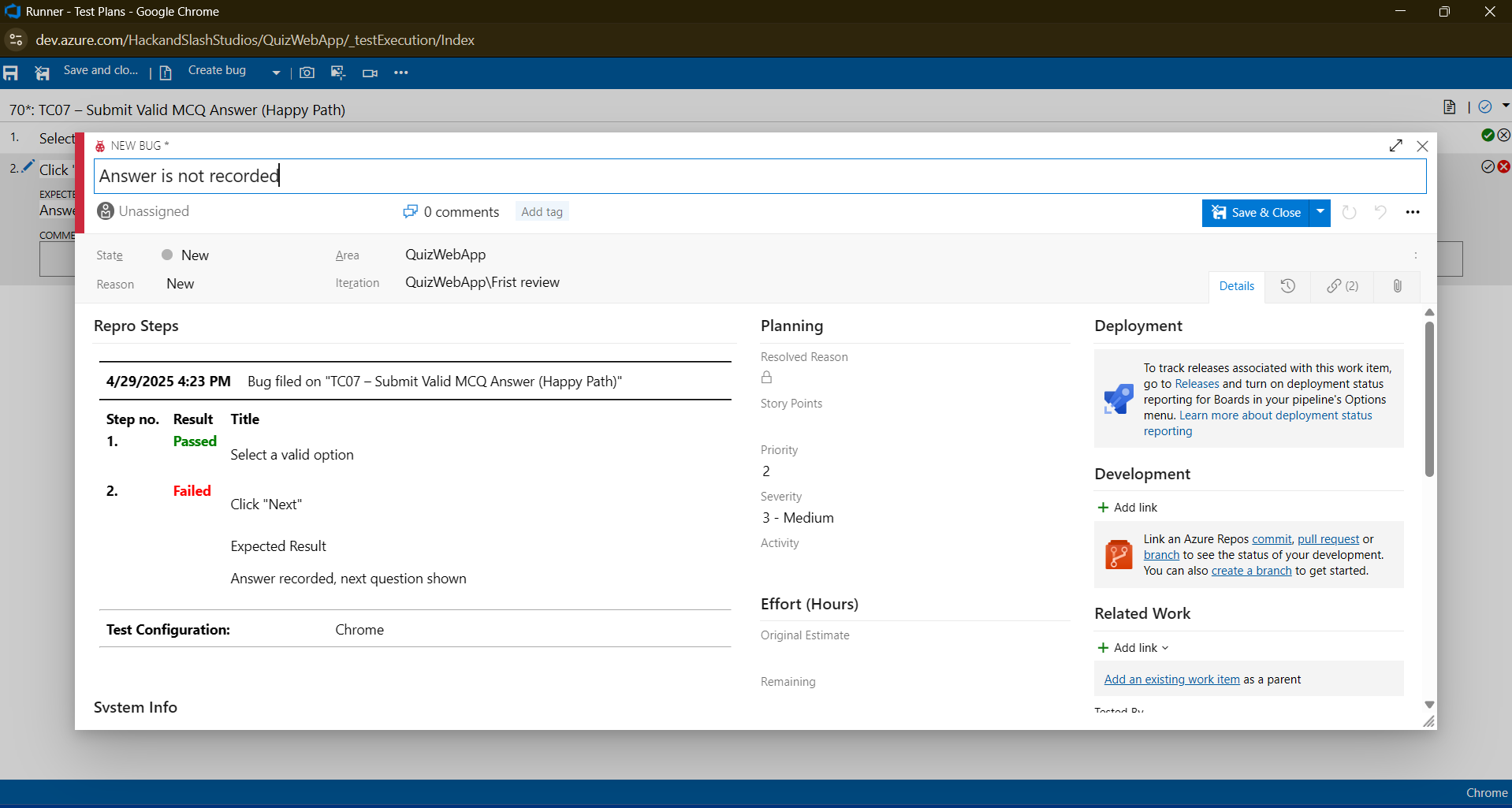
****

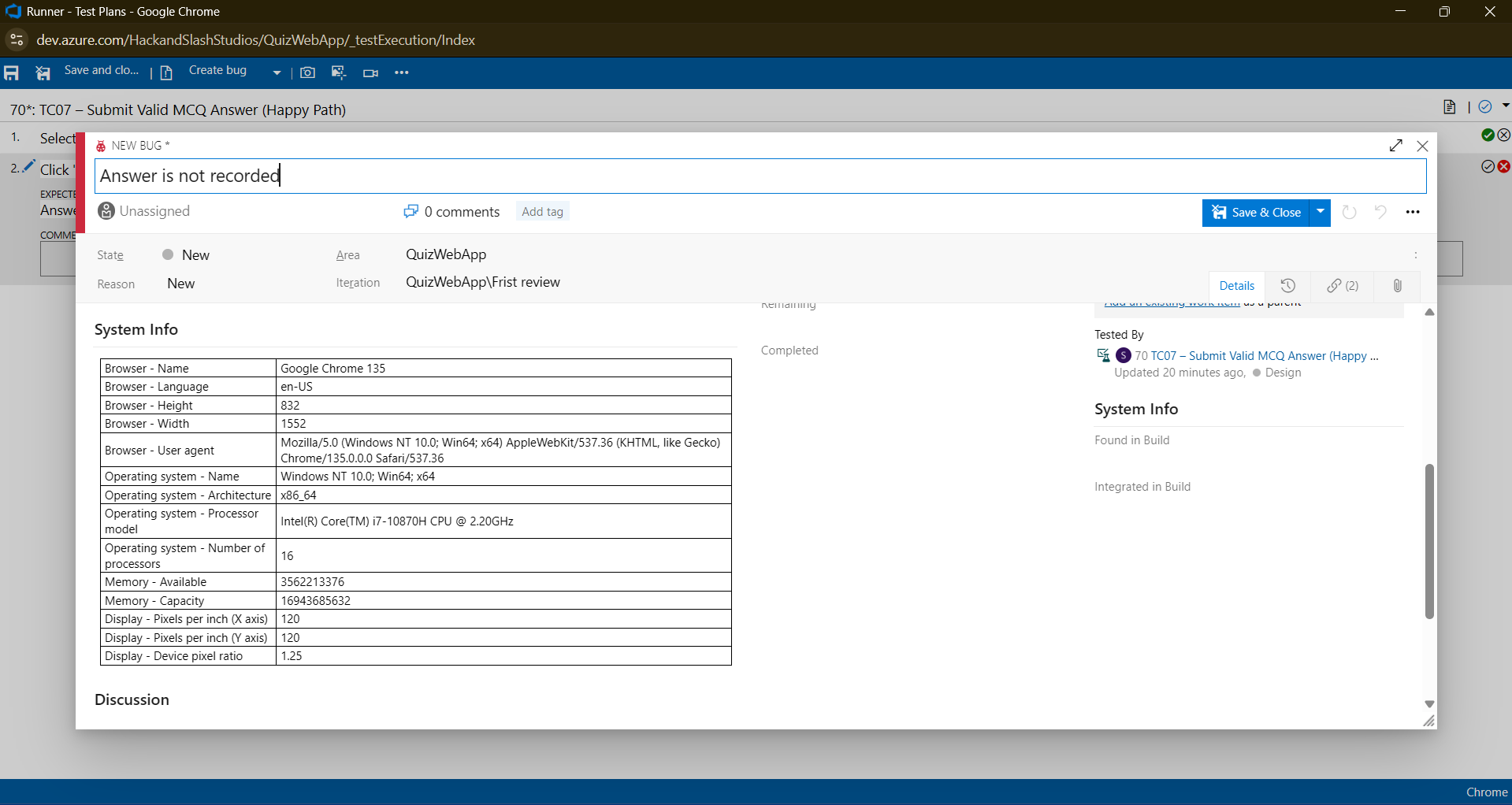
**6.Recording the test case**

****

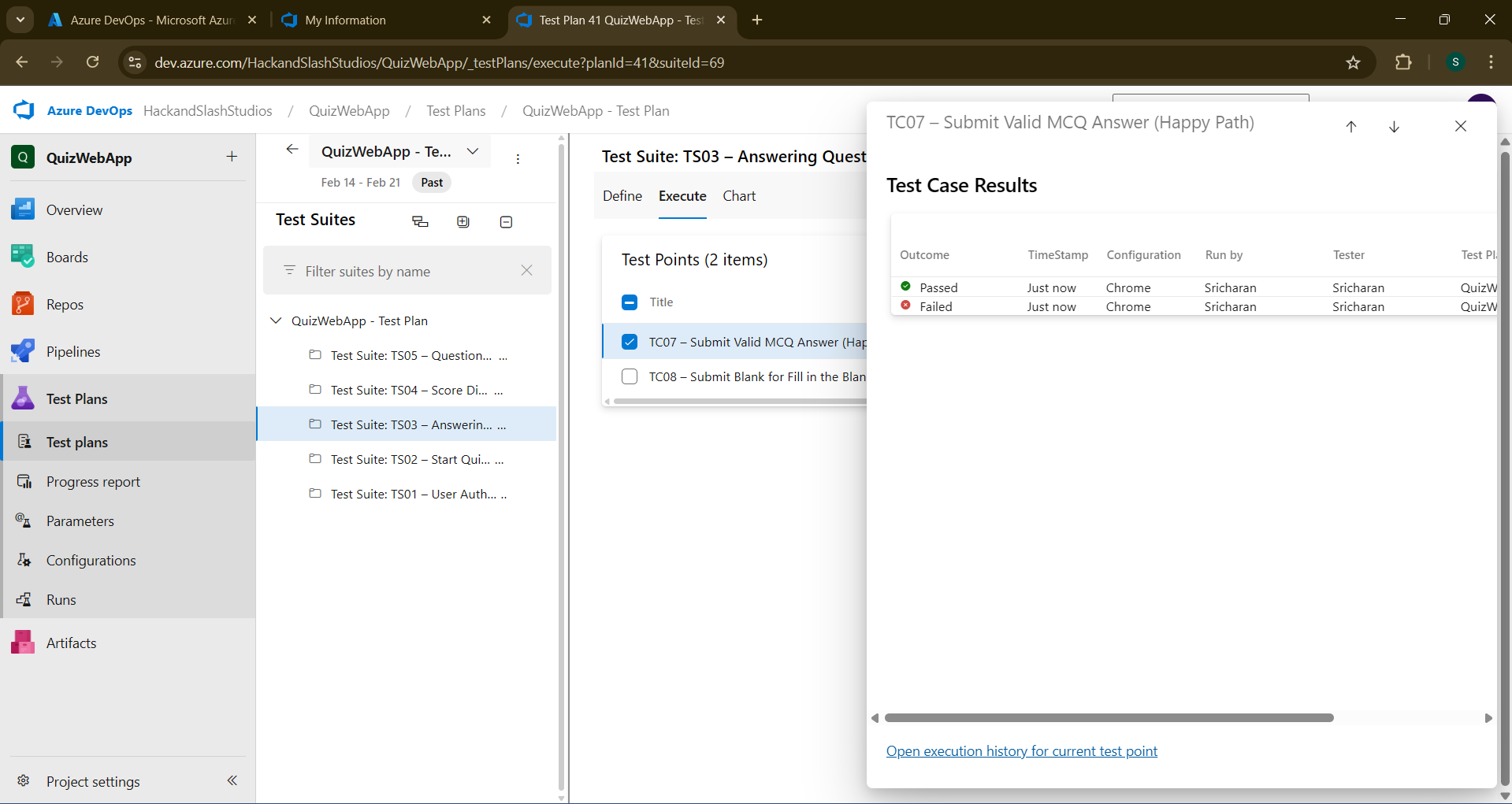
**7.Creating the bug**

****

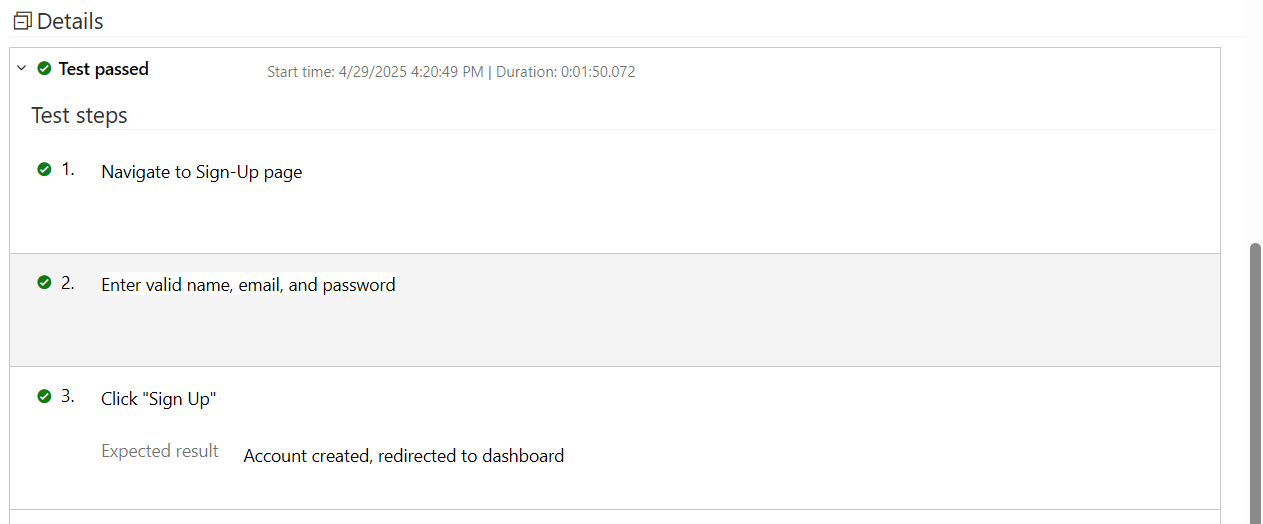
****

****

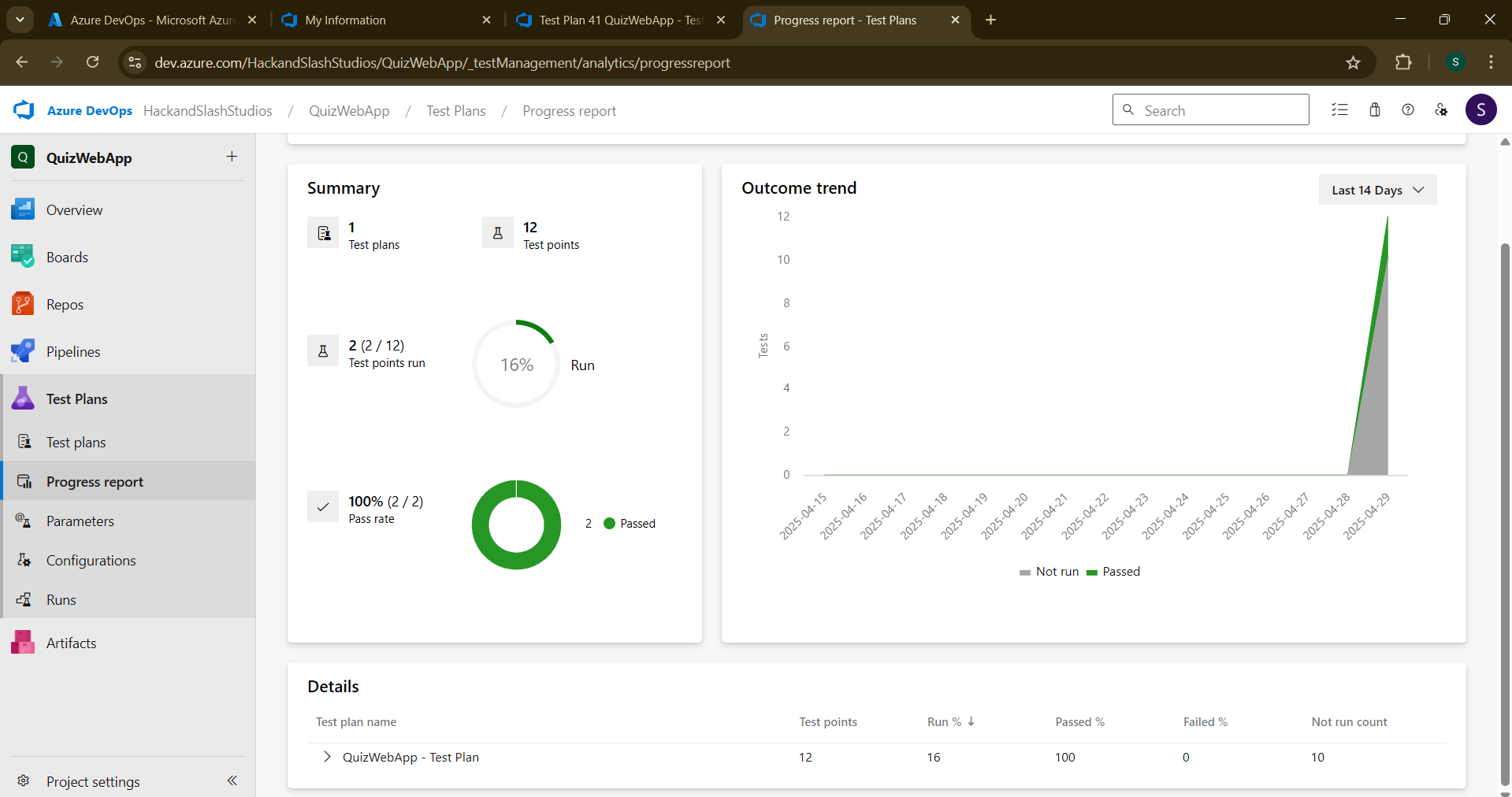
**8.Test case results**

****

**9.Test report summary**

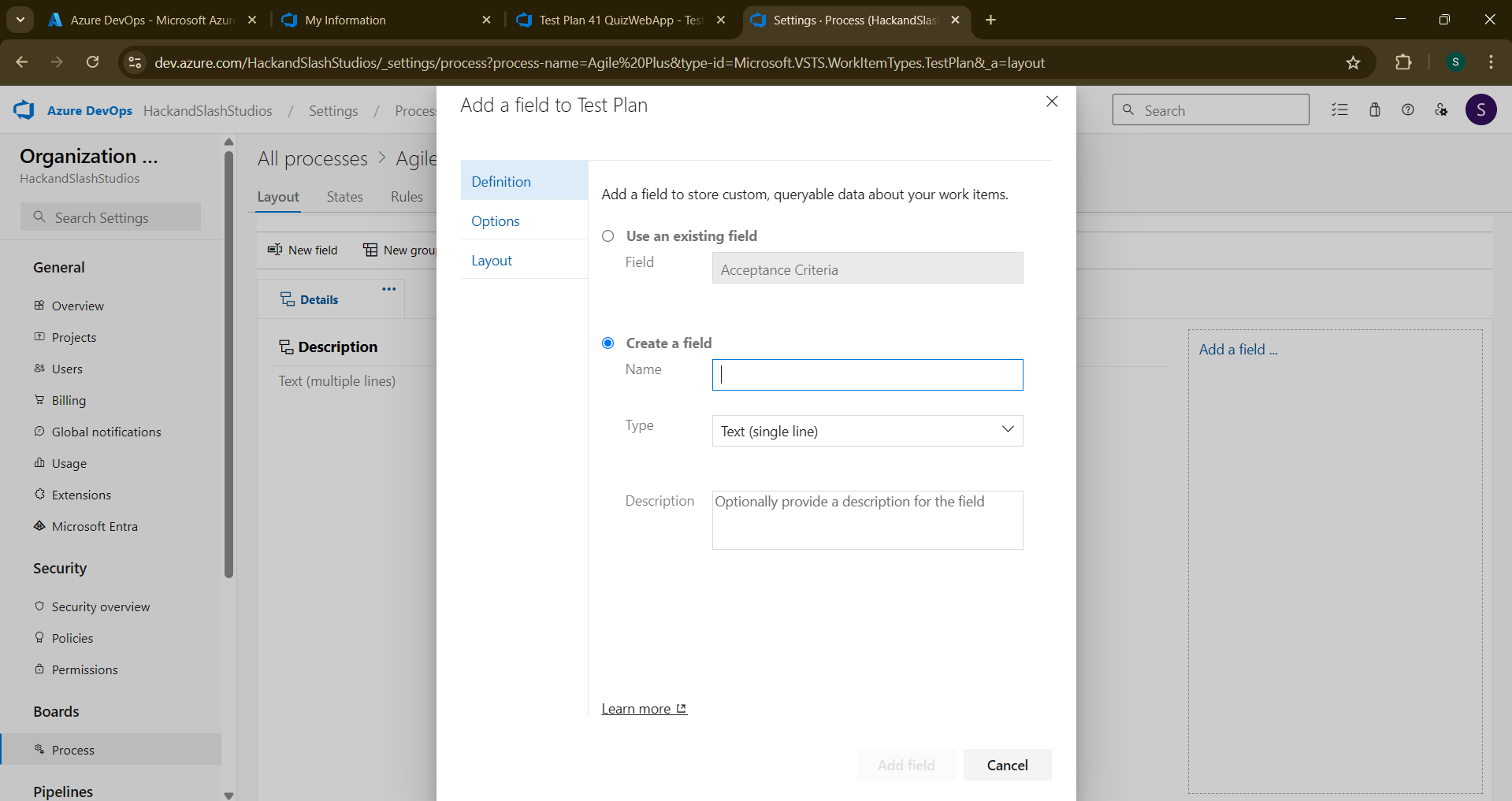
****

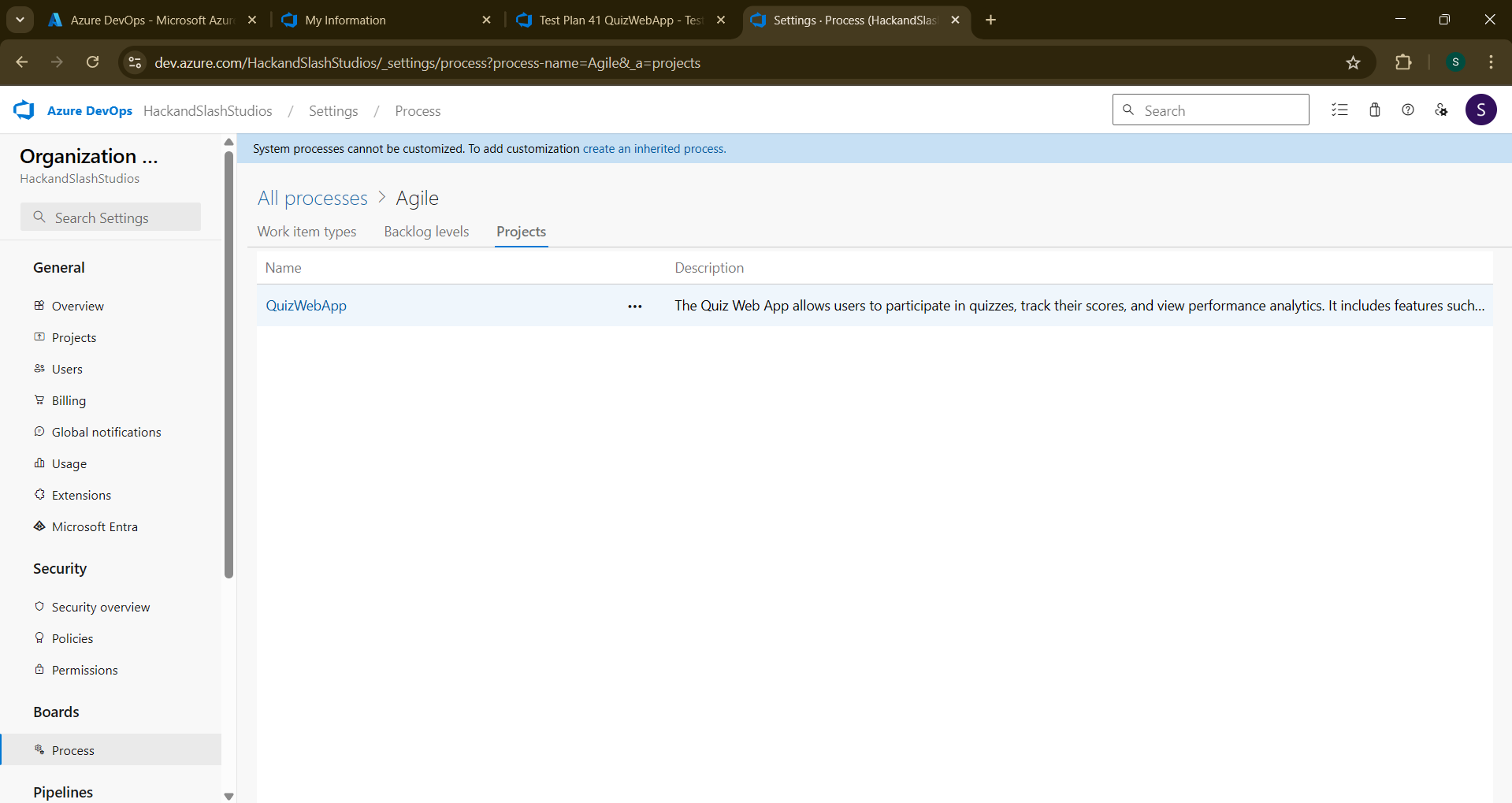
**10.Progress report**

****

**11.Changing the test 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

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.

**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.

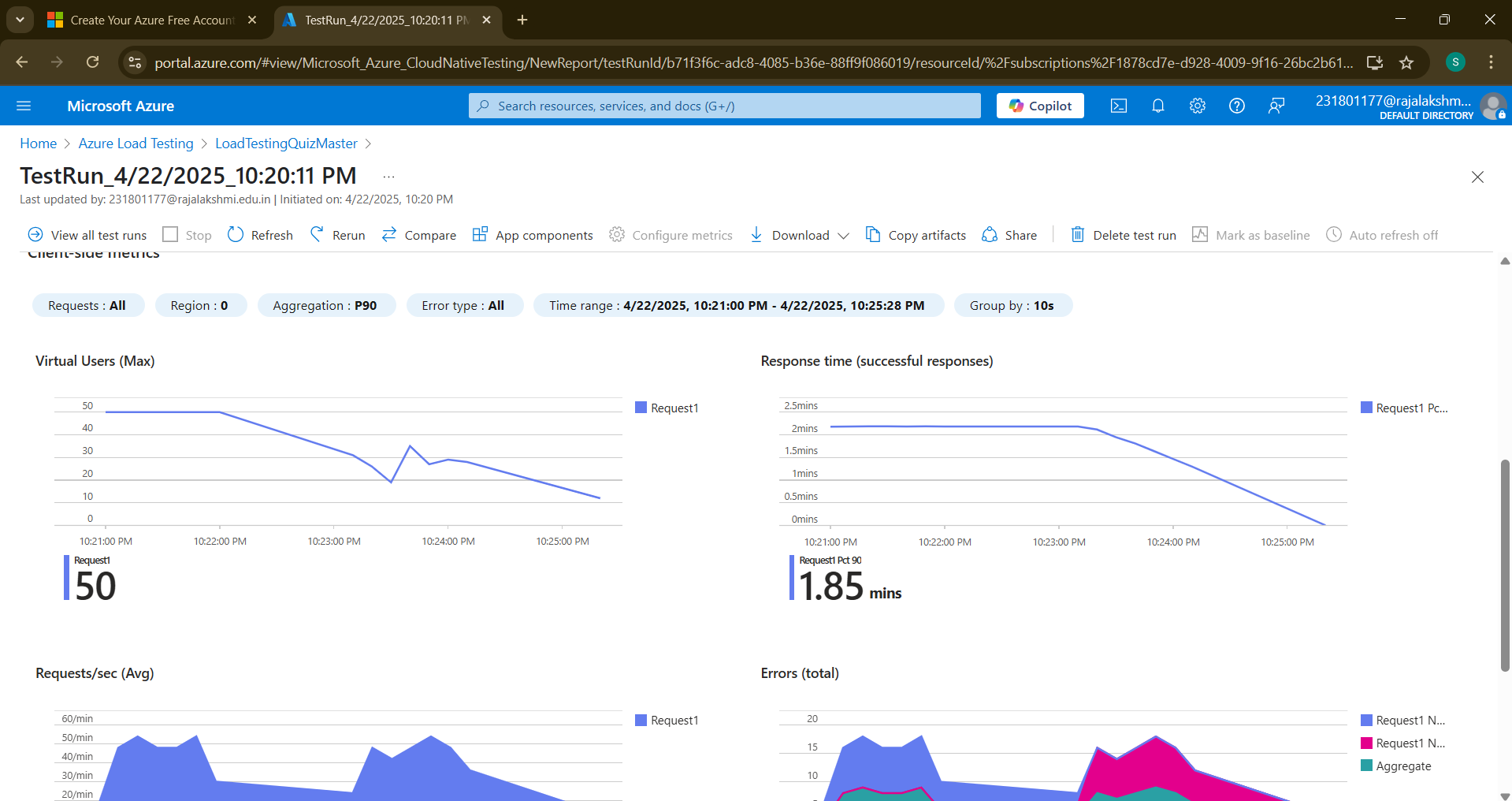
o Run After Creation: Keep checked.

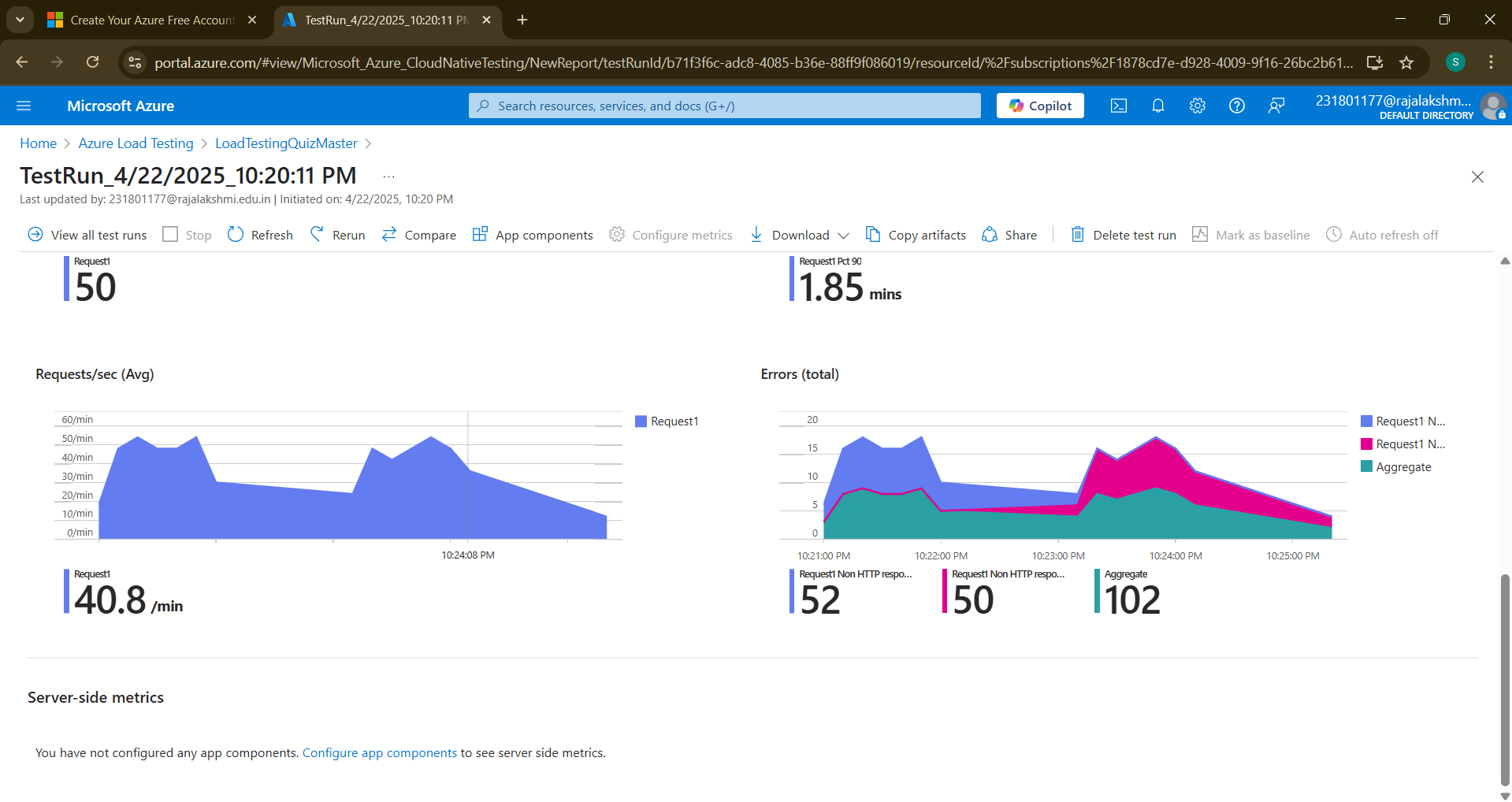
3. Load Settings

o Test URL: Enter the target endpoint (e.g., https://yourapi.com/products).

4. Click Review + Create → Create to start the test.

**Load Testing**

****

****

**Result:**

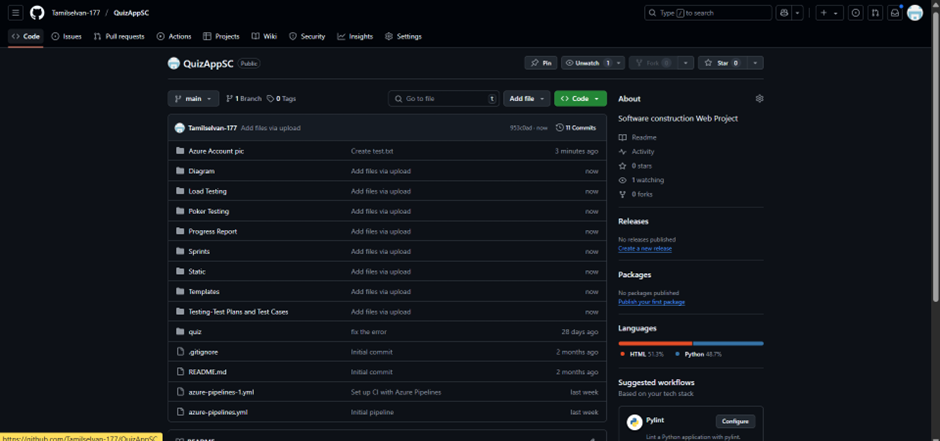
Successfully created the Azure Load Testing resource and executed a load test to assess the performance of the specified endpoint.

|  |  |
| --- | --- |
| **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 Playlist Batch Creator 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.