**INDEX**

| **S.NO** | **DATE** | **TITLE** |
| --- | --- | --- |
| **1.** | **21/1/25** | **Azure DevOps Environment Setup** |
| **2.** | **21/1/25** | **Azure DevOps Project Setup and User Story Management** |
| **3.** | **28/1/25** | **Setting Up Epics, Features and User Stories for Project Planning** |
| **4.** | **11/2/25** | **Sprint Planning** |
| **5.** | **18/2/25** | **Poker Estimation** |
| **6.** | **25/2/25** | **Designing Class and Sequence Diagrams for Project Architecture** |
| **7.** | **04/3/25** | **Designing Use-Case and Activity Diagrams for Project Architecture** |
| **8.** | **25/3/25** | **Testing – Test Plans and Test Cases** |
| **9.** | **15/4/25** | **CI/CD Pipelines in Azure** |
| **10.** | **22/4/25** | **GitHub: Project Structure & Naming Convention** |

| **EXP NO:** **1** | **AZURE DEVOPS ENVIRONMENT SETUP** |
| --- | --- |
|  |

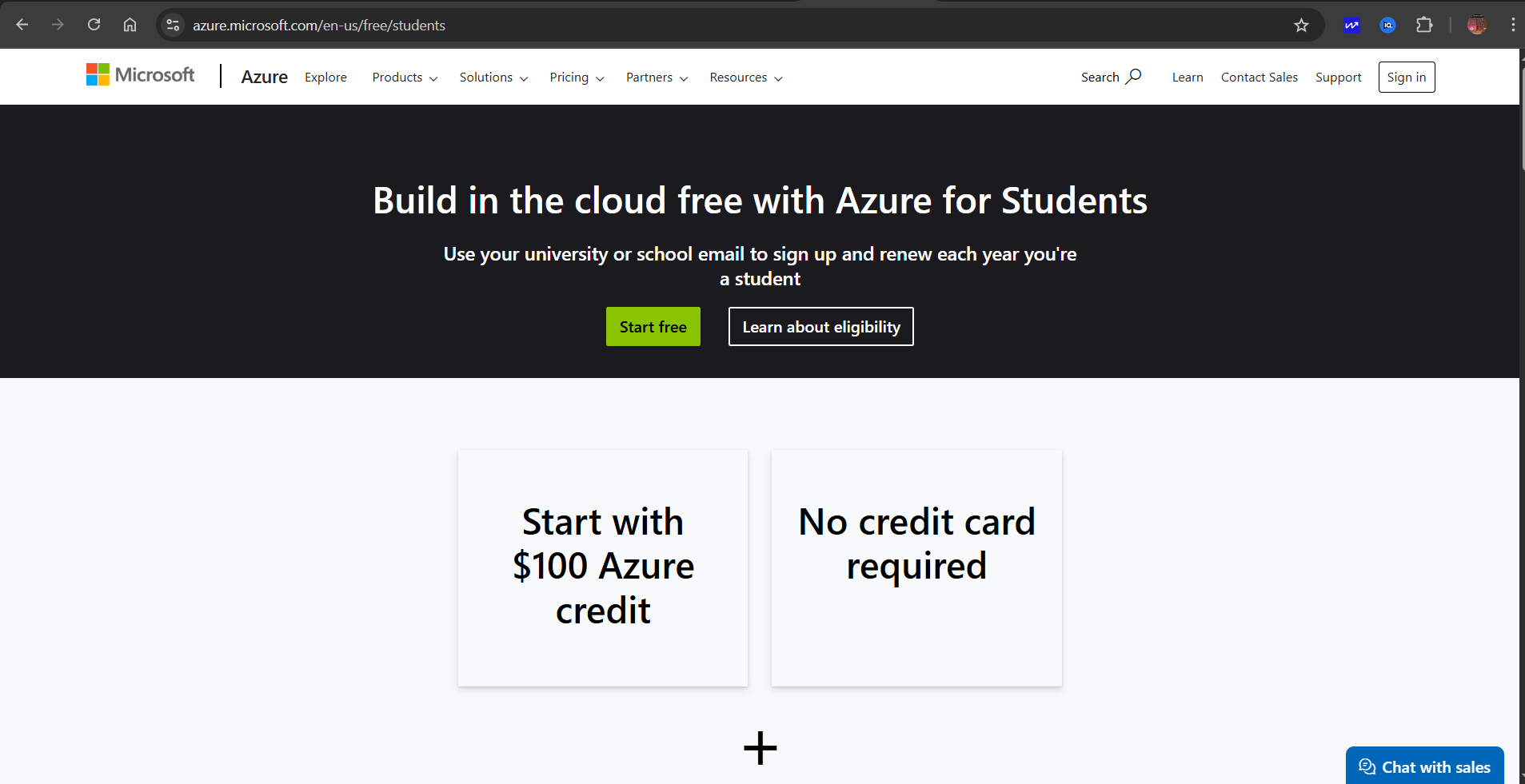
# AIM

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

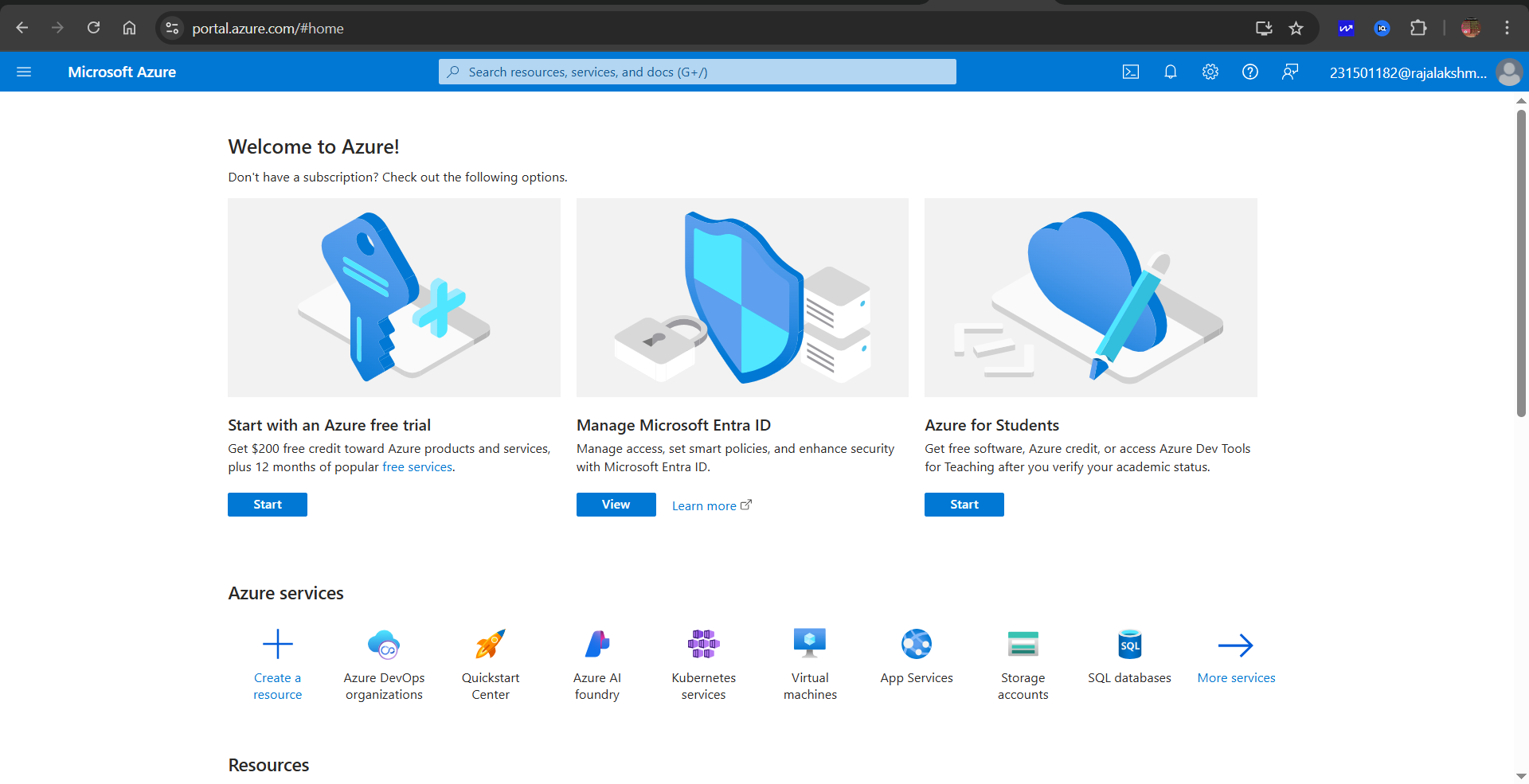
# INSTALLATION

1.Open your web browser and go to the Azure website: [https://azure.microsoft.com/enus/getstarted/azureportal.](https://azure.microsoft.com/en-us/get-started/azure-portal)

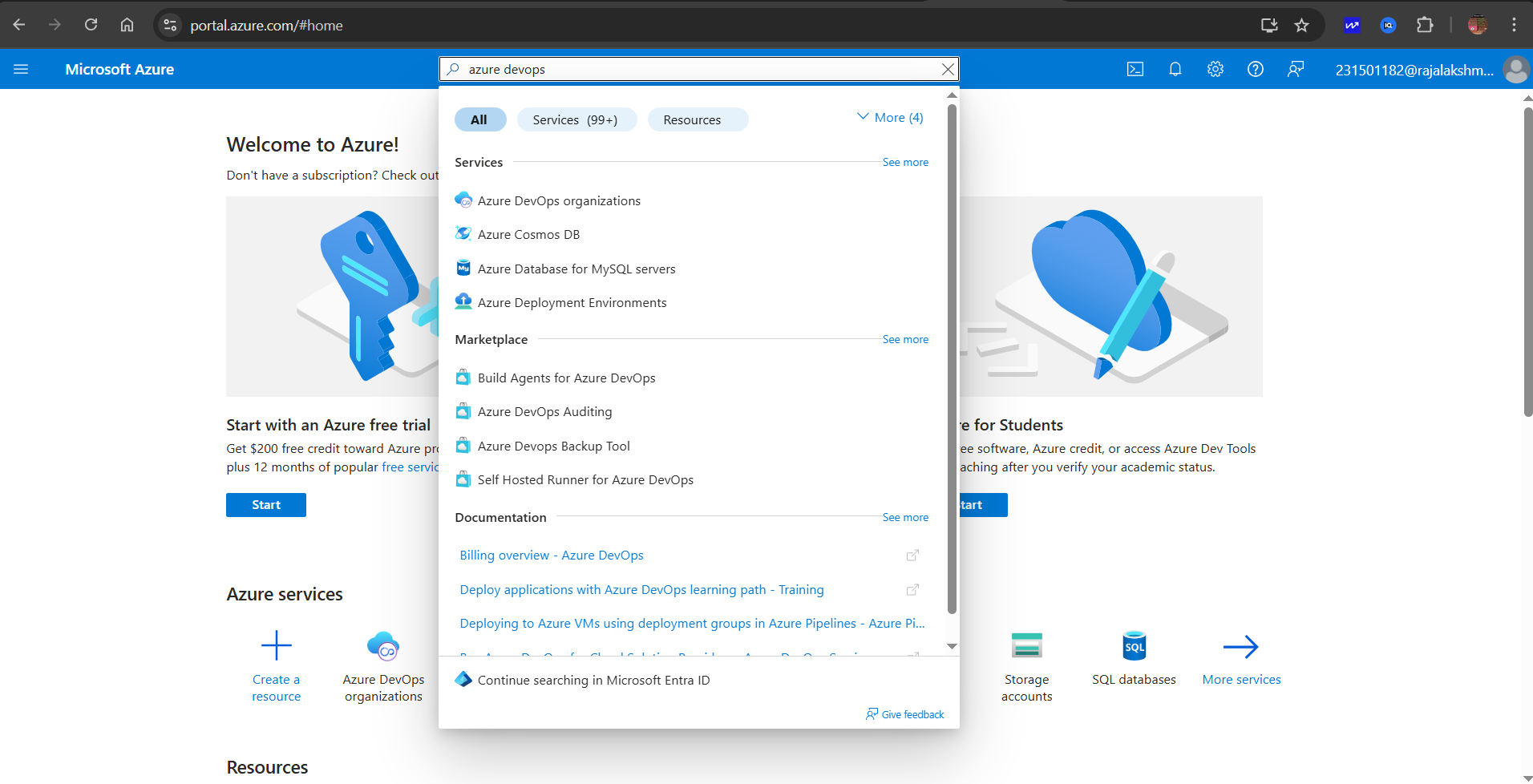
Sign in using your Microsoft account credentials.

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

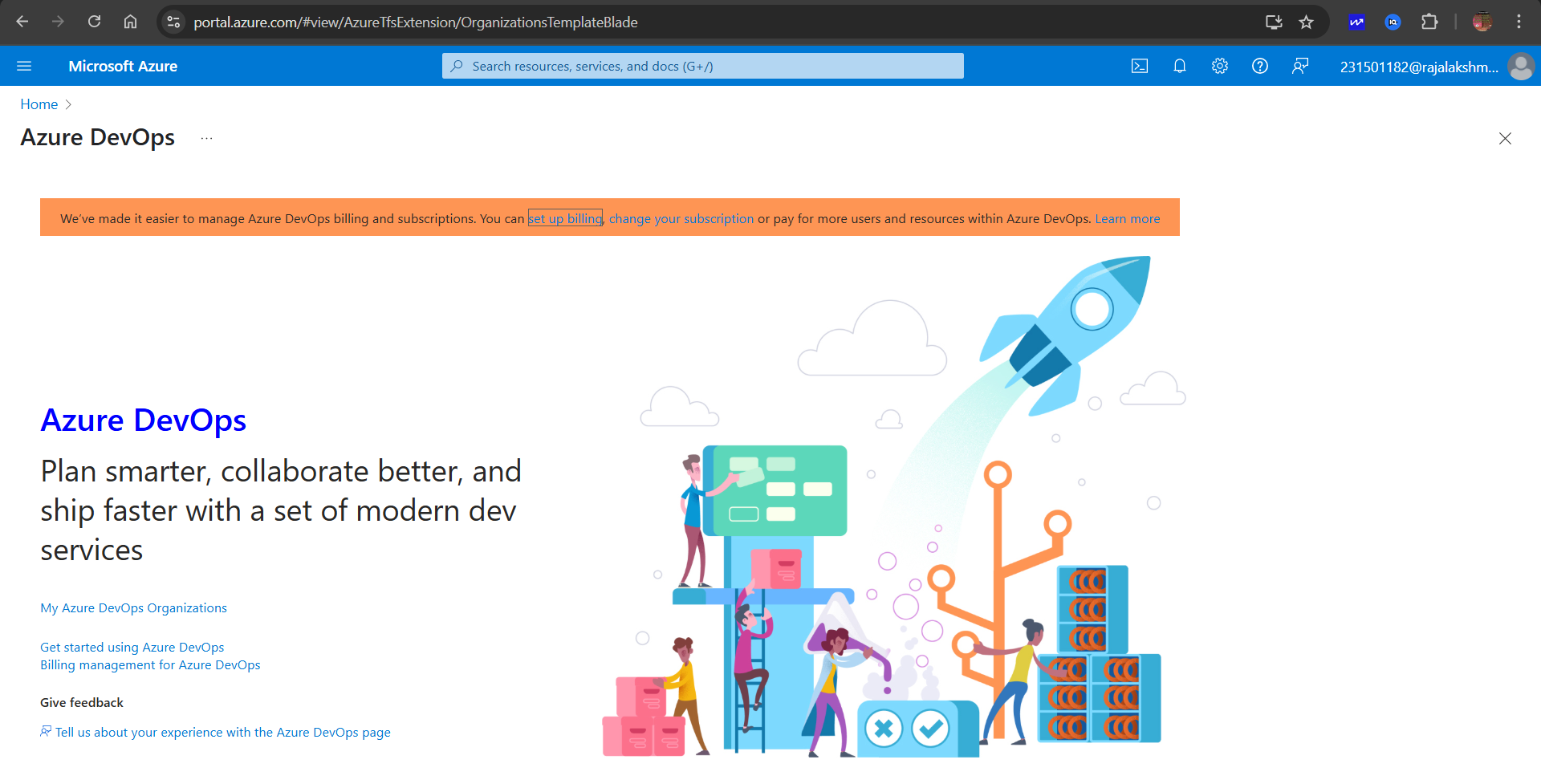
2.Azure home page



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.

****

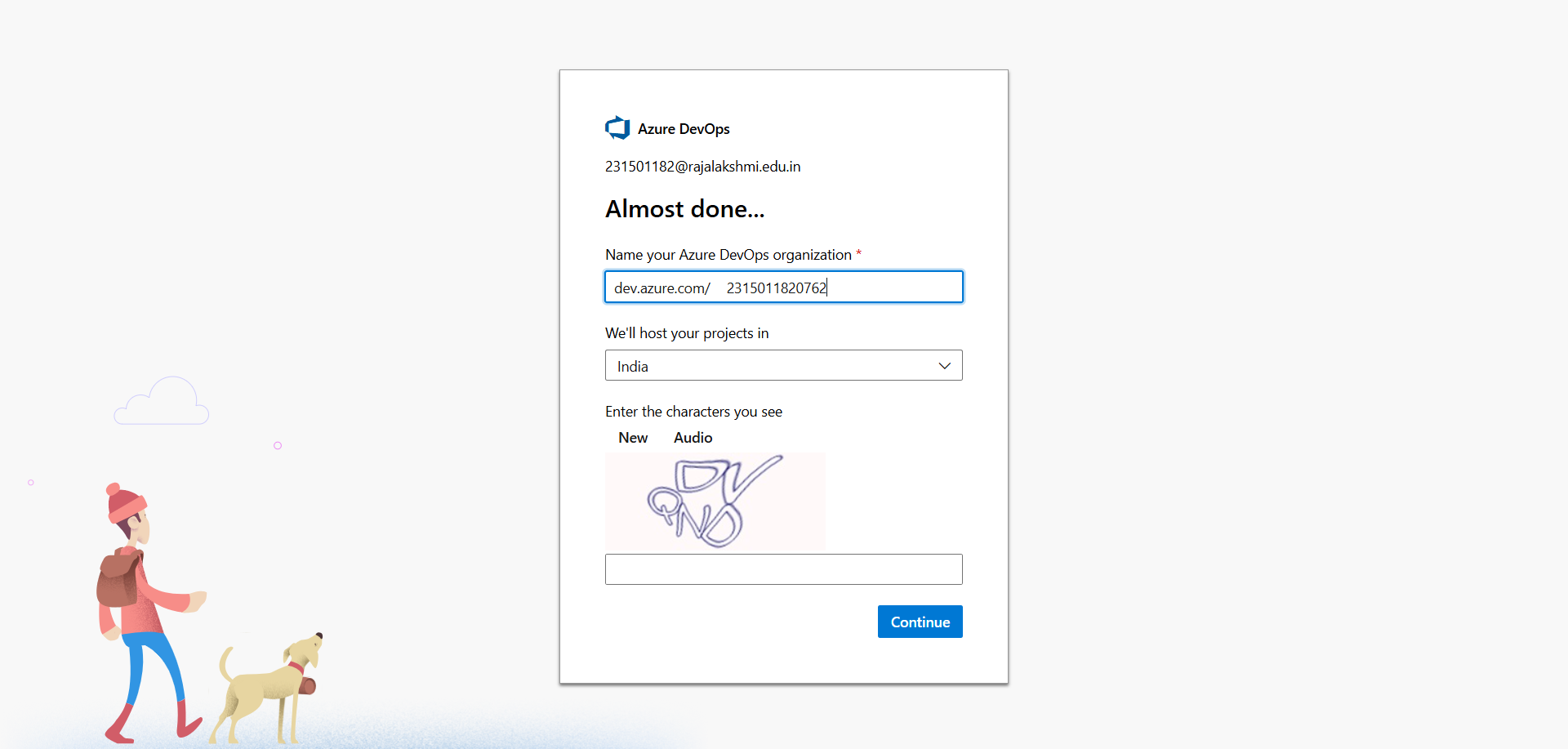
# RESULT

| **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

1. After the organization is set up, you’ll need to create your first **project**. This is where you'll

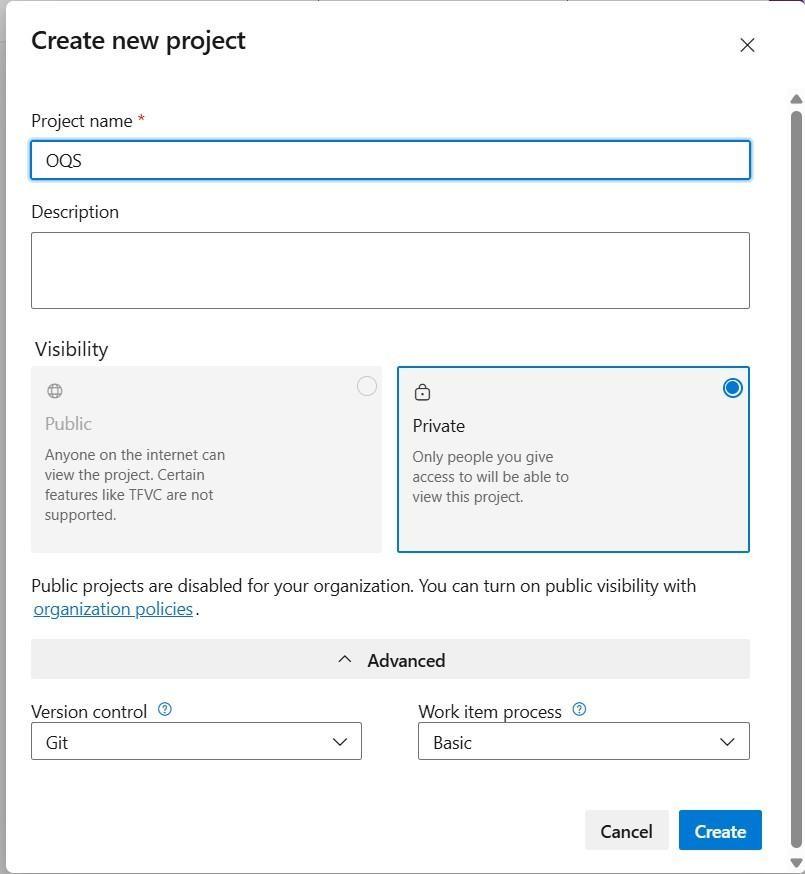
begin to manage code, pipelines, work items, and more.

1. On the organization’s **Home page**, click on the **New Project** button.
2. Enter the project name, description, and visibility options:

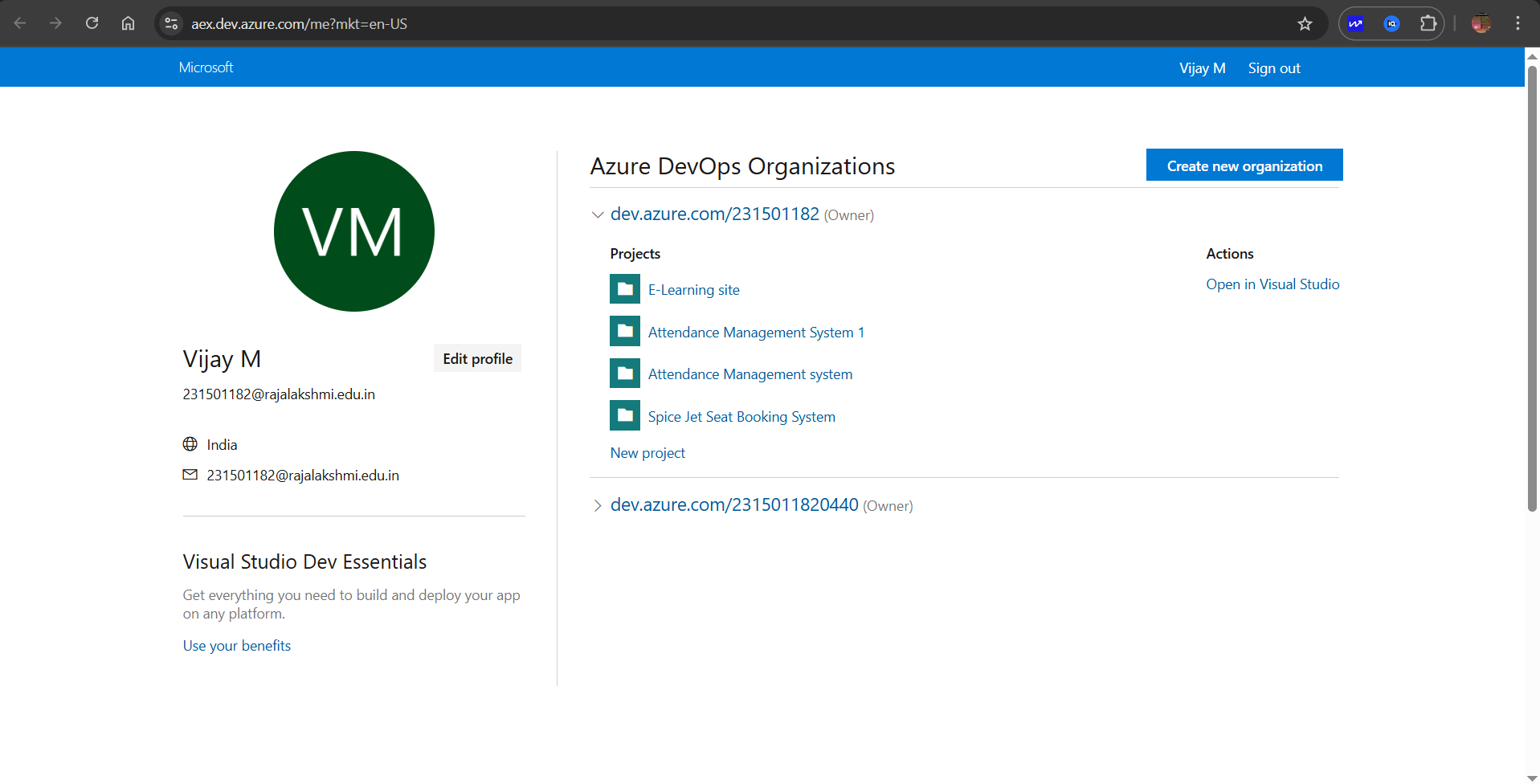
***Name****:* Choose a name for the project (e.g., LMS).

***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).

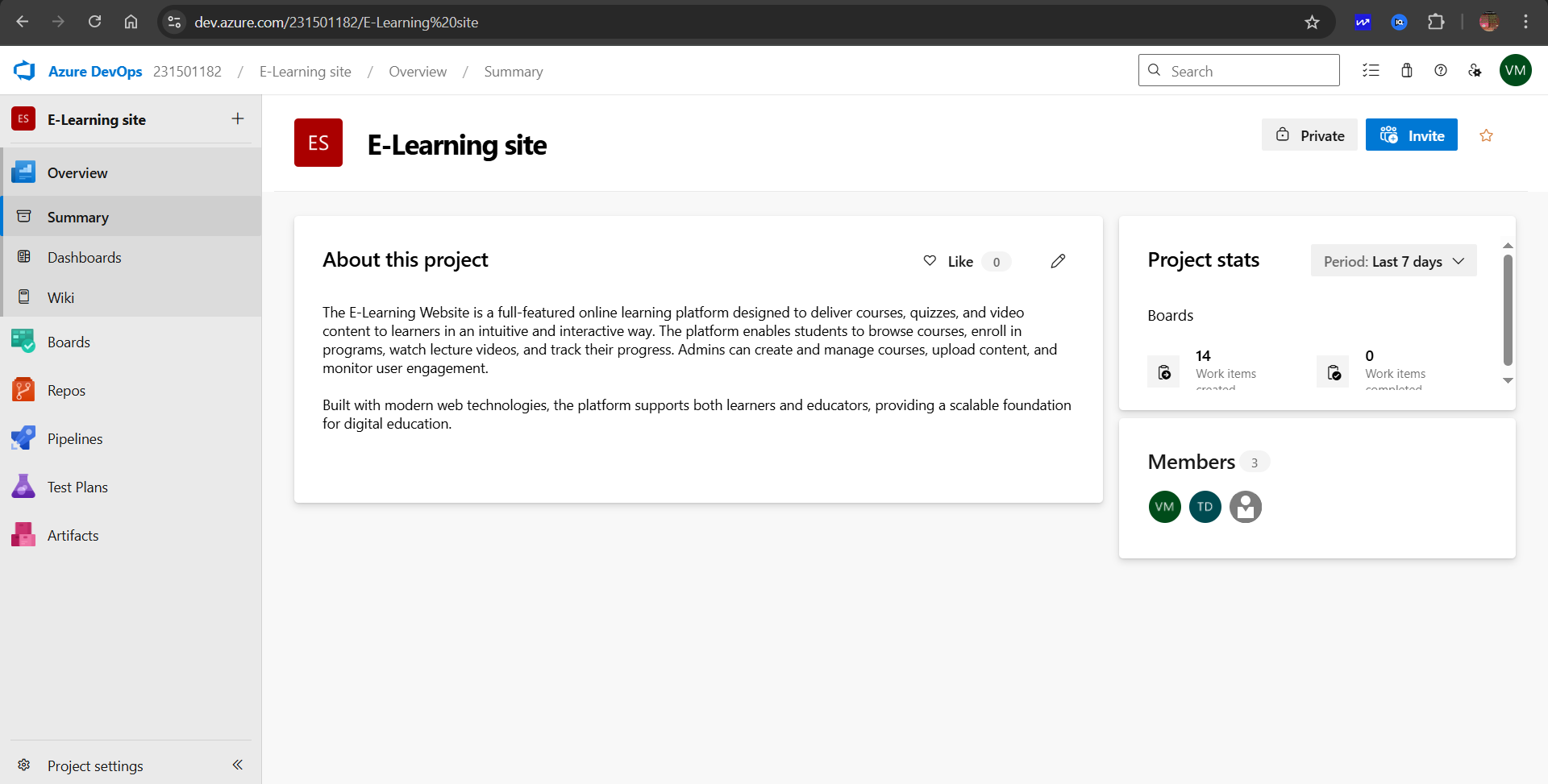
1. Once you’ve filled out the details, click **Create** to set up your first project.



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:

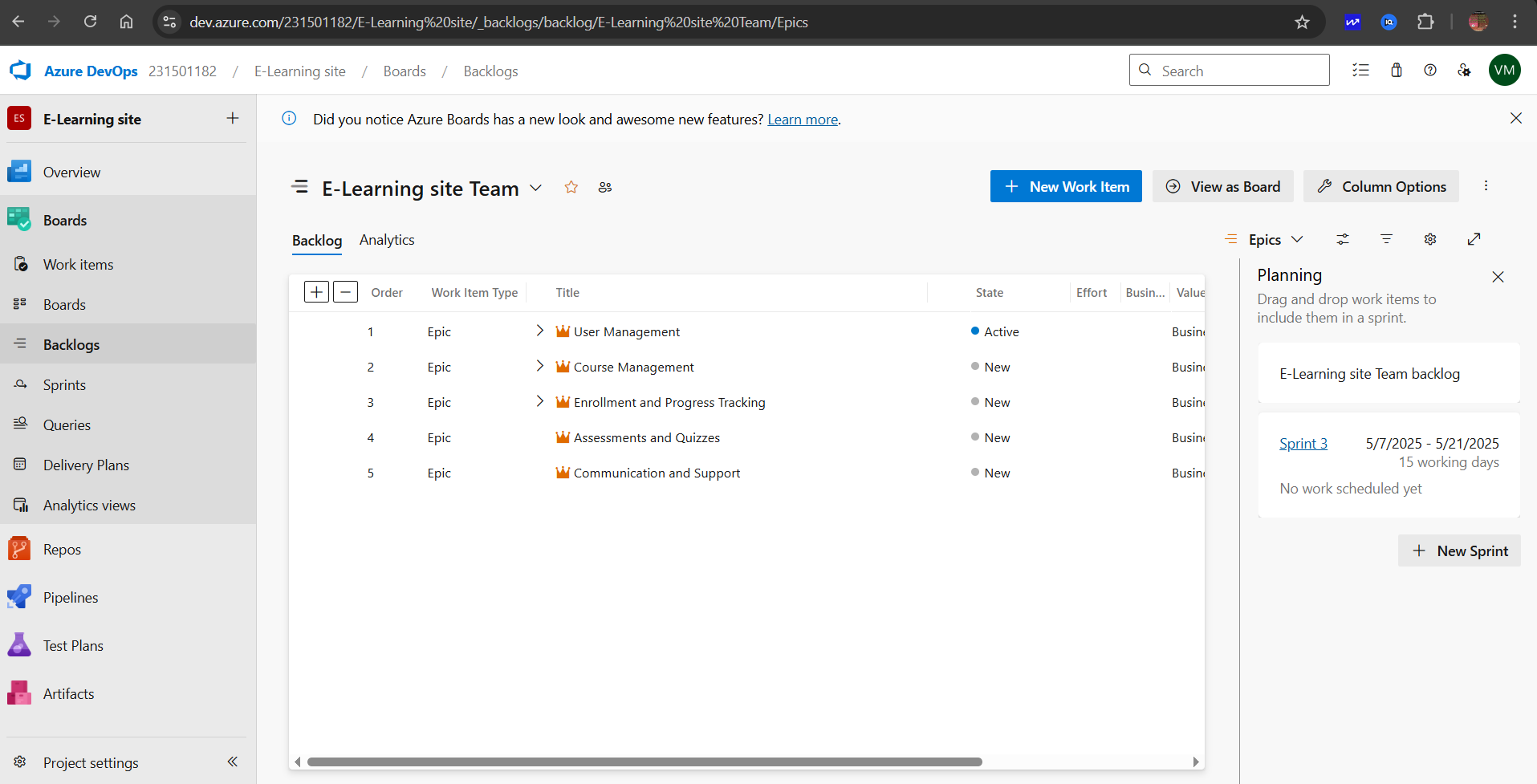
1. From the **left-hand navigation menu**, click on **Boards**. This will take you to the main **Boards**

page, where you can manage work items, backlogs, and sprints.

1. 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

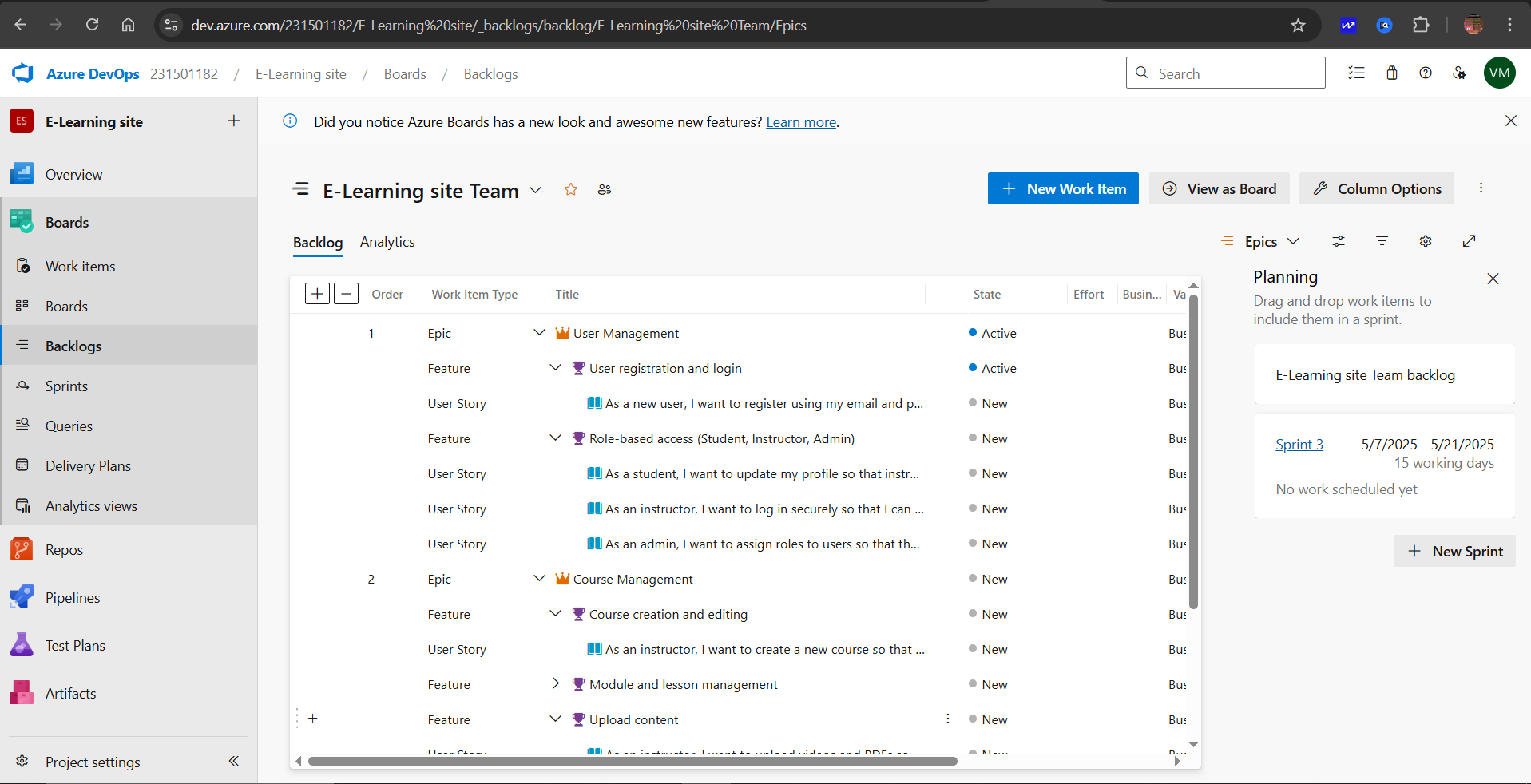
Successfully created an Azure DevOps project with user story management and agile workflow setup.

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

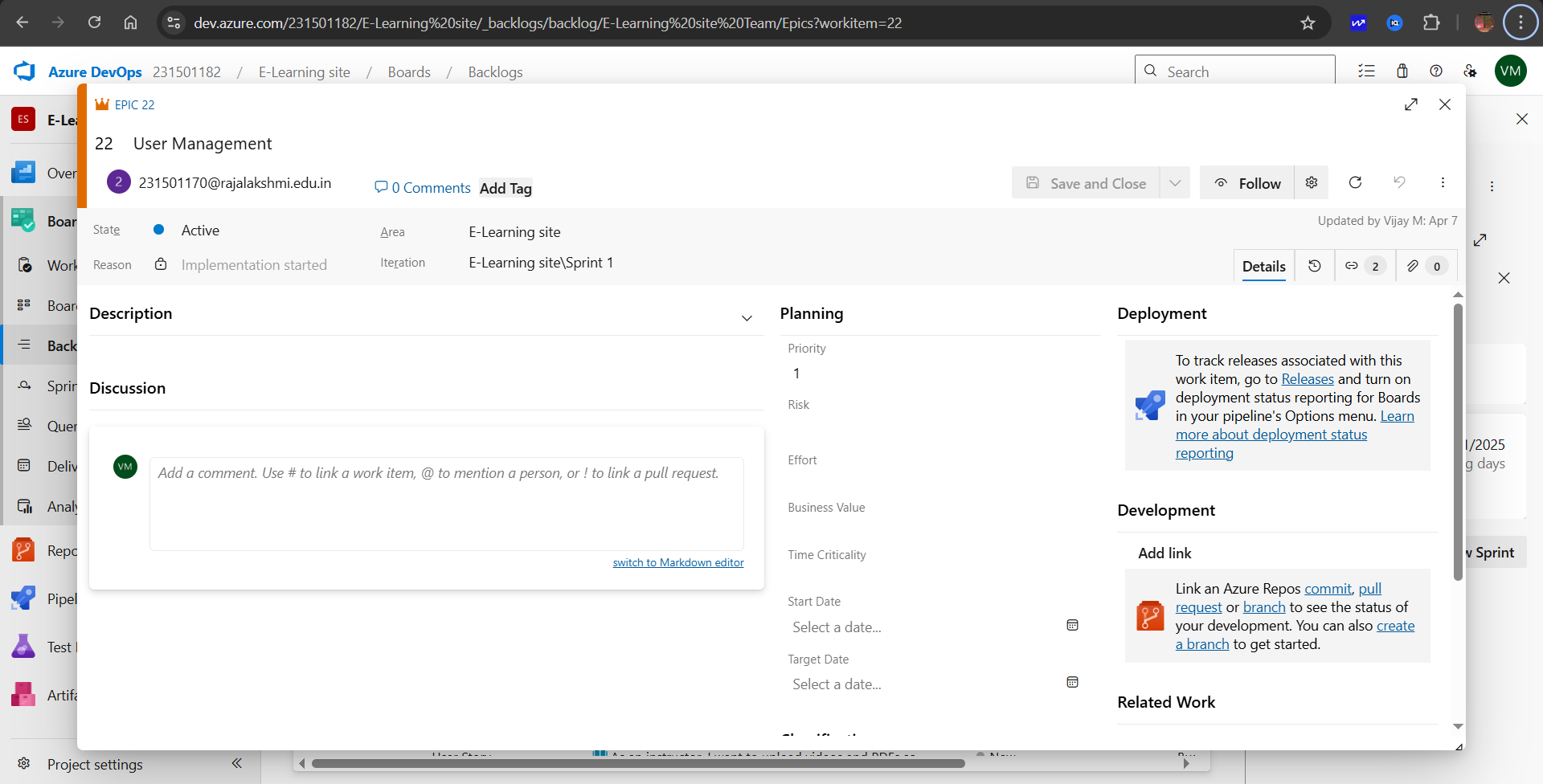
**AIM**

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

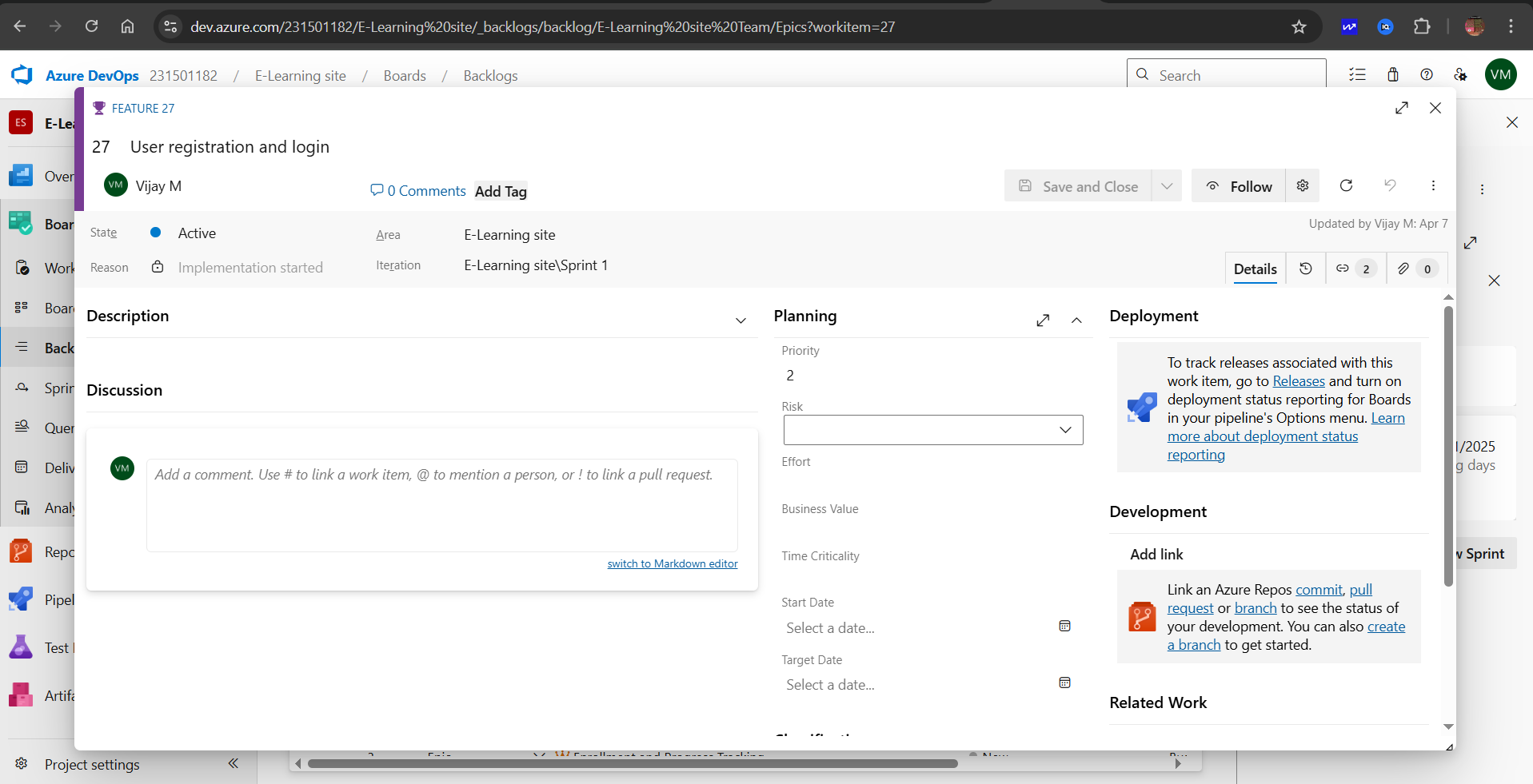
# Create Epic, Features, User Stories, Task



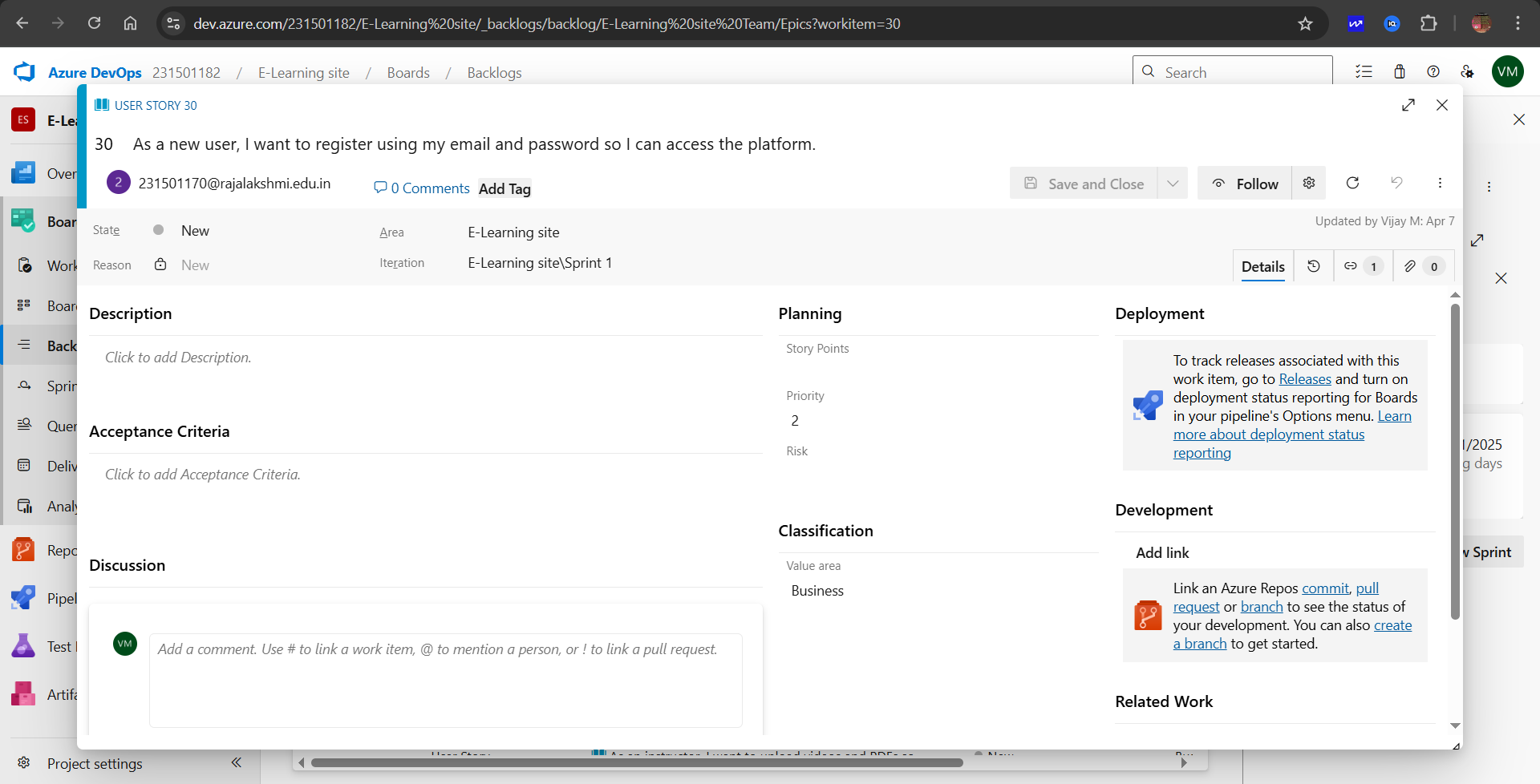
# 1.Fill in Epics



## 2.Fill in Features



# 3.Fill in User Story Details



**RESULT**

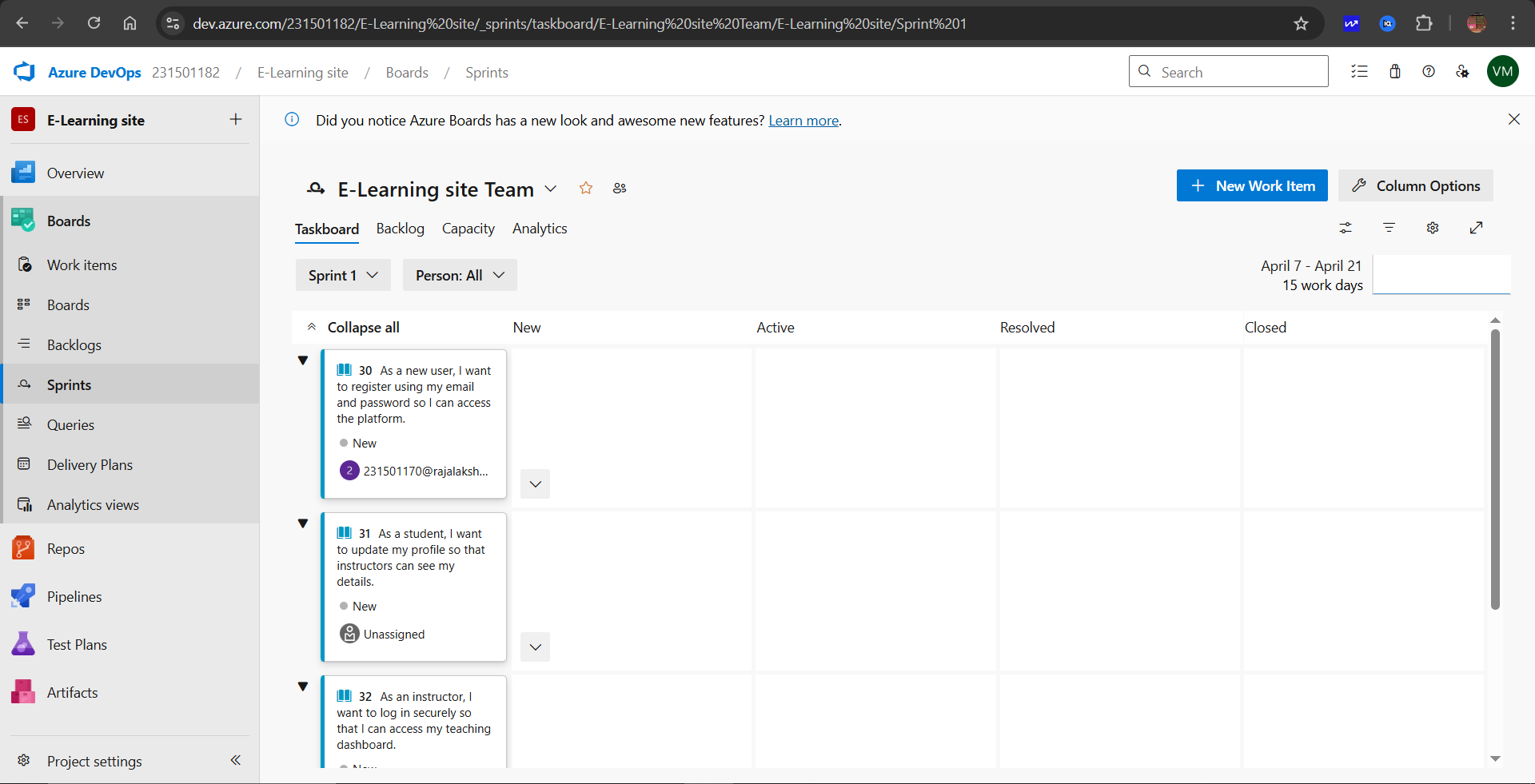
Thus, the creation of epics, features, user story and task has been created successfully.

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

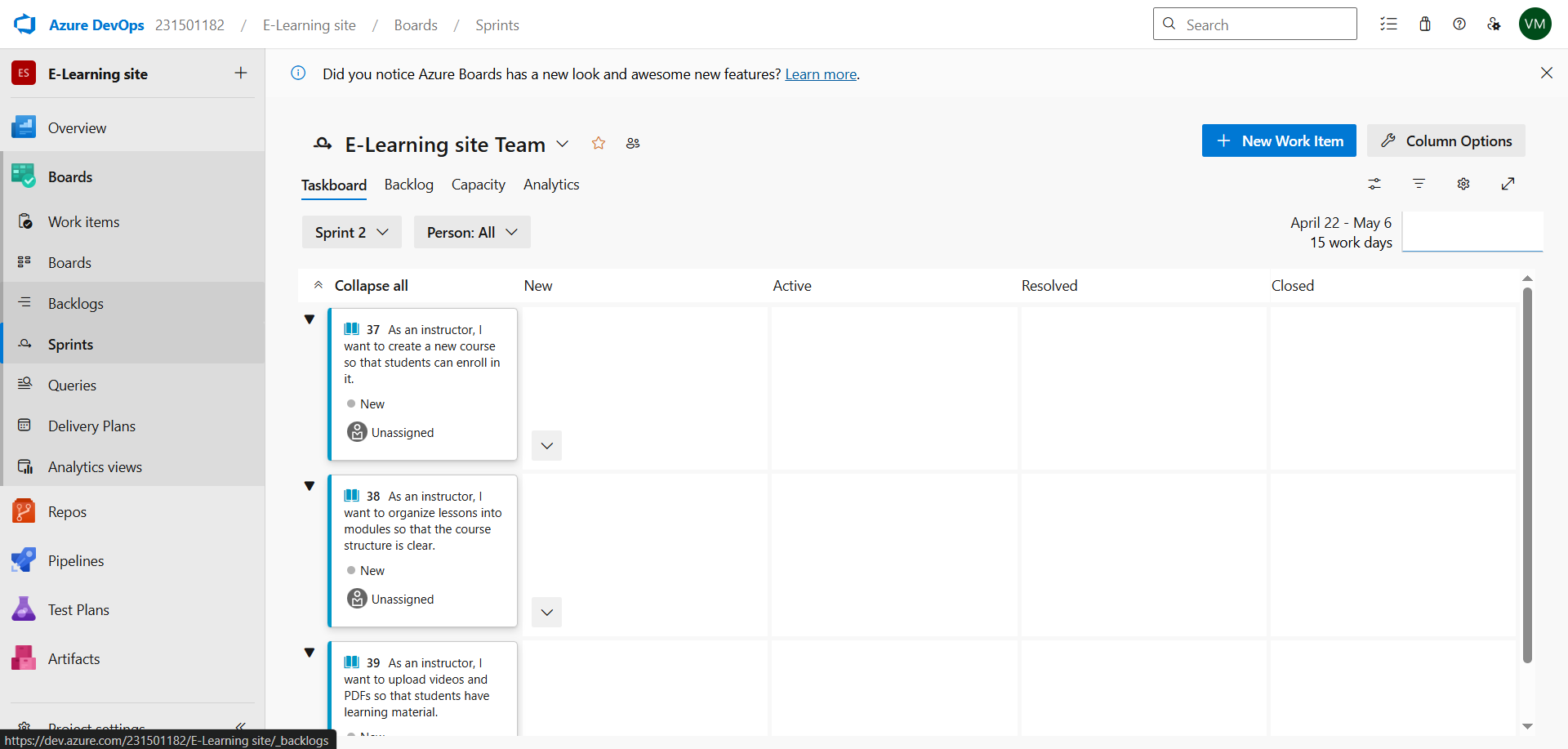
**AIM**

To assign user story to specific sprint for the E-Learning site.

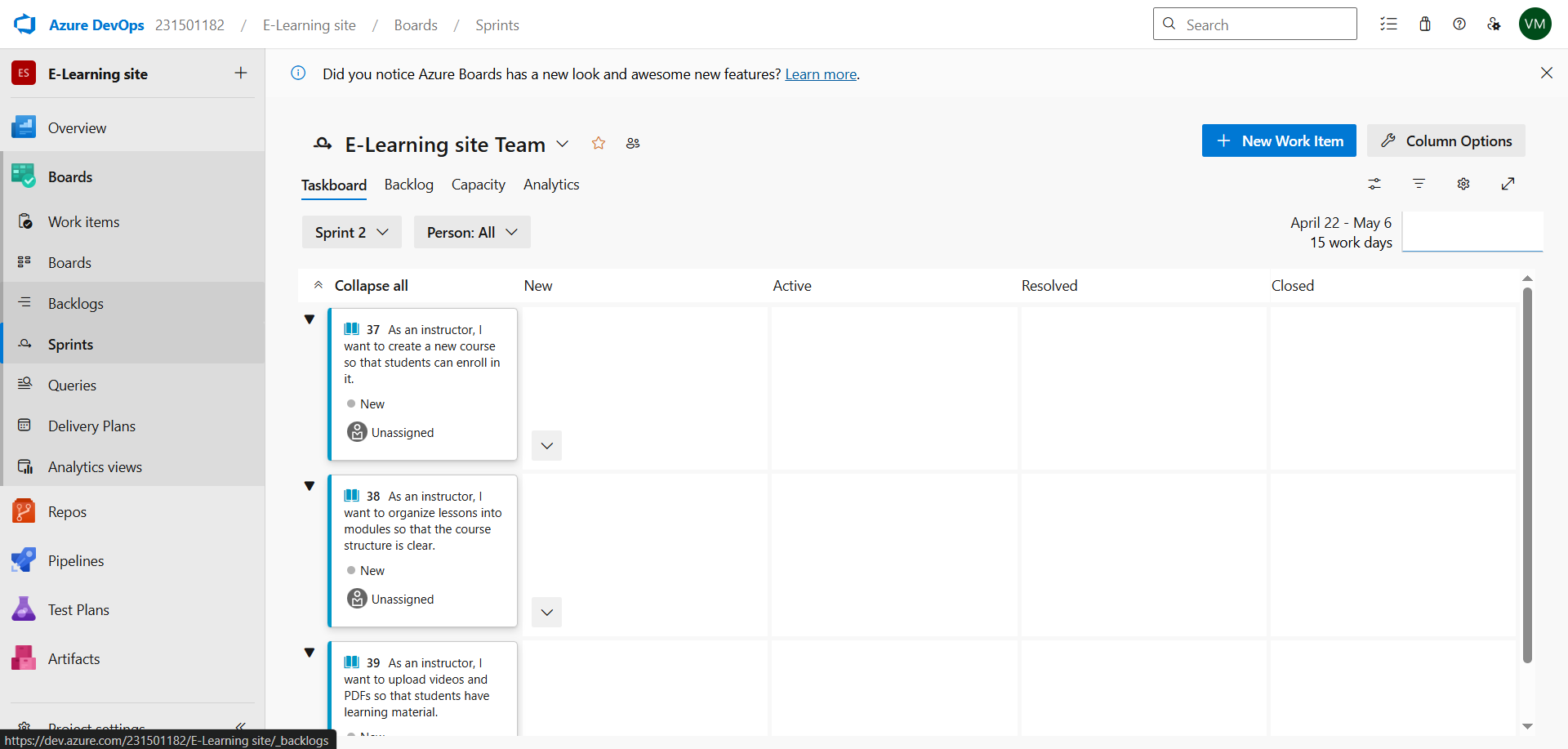
## Sprint Planning Sprint 1



## Sprint 2



## Sprint 3



**RESULT**

The Sprints have been created for E-Learning site.

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

**AIM**

Create Poker Estimation for the user stories –E-Learning site.

## Poker Estimation



**RESULT**

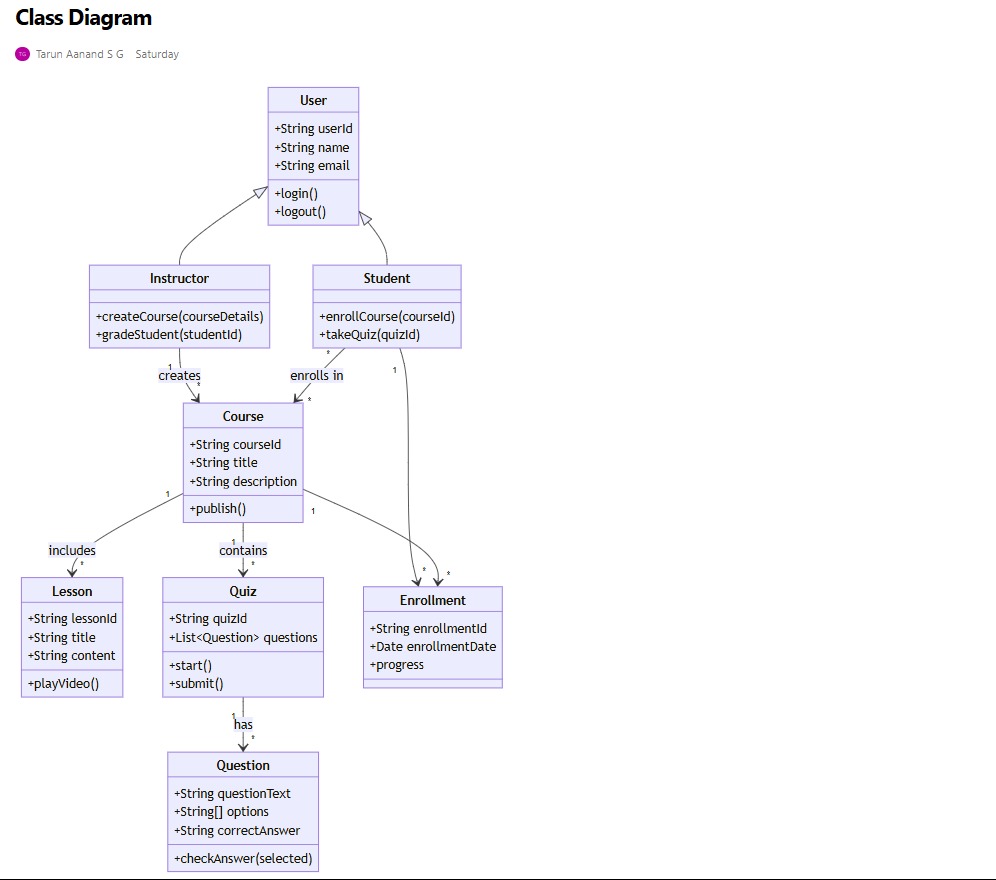
The Estimation/Story Points is created for the project using Poker Estimation.

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

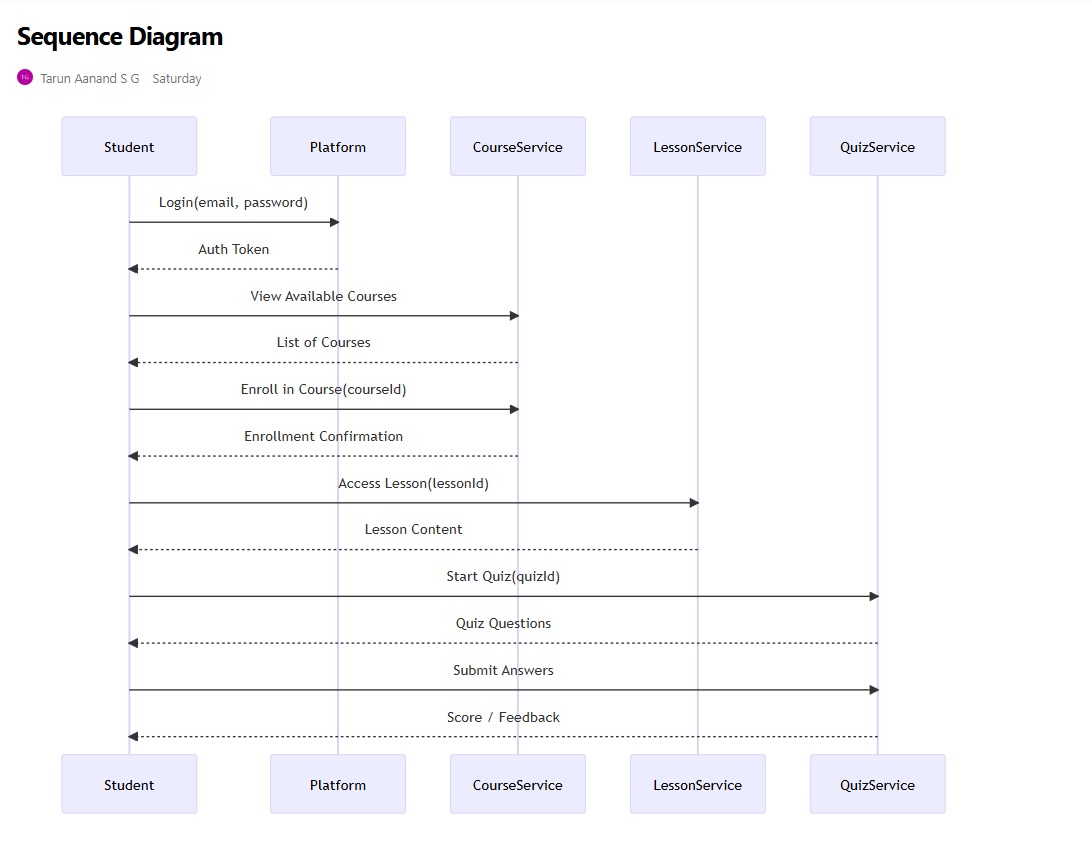
**AIM**

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

## 6A. Class Diagram



## 6B. Sequence Diagram



## RESULT

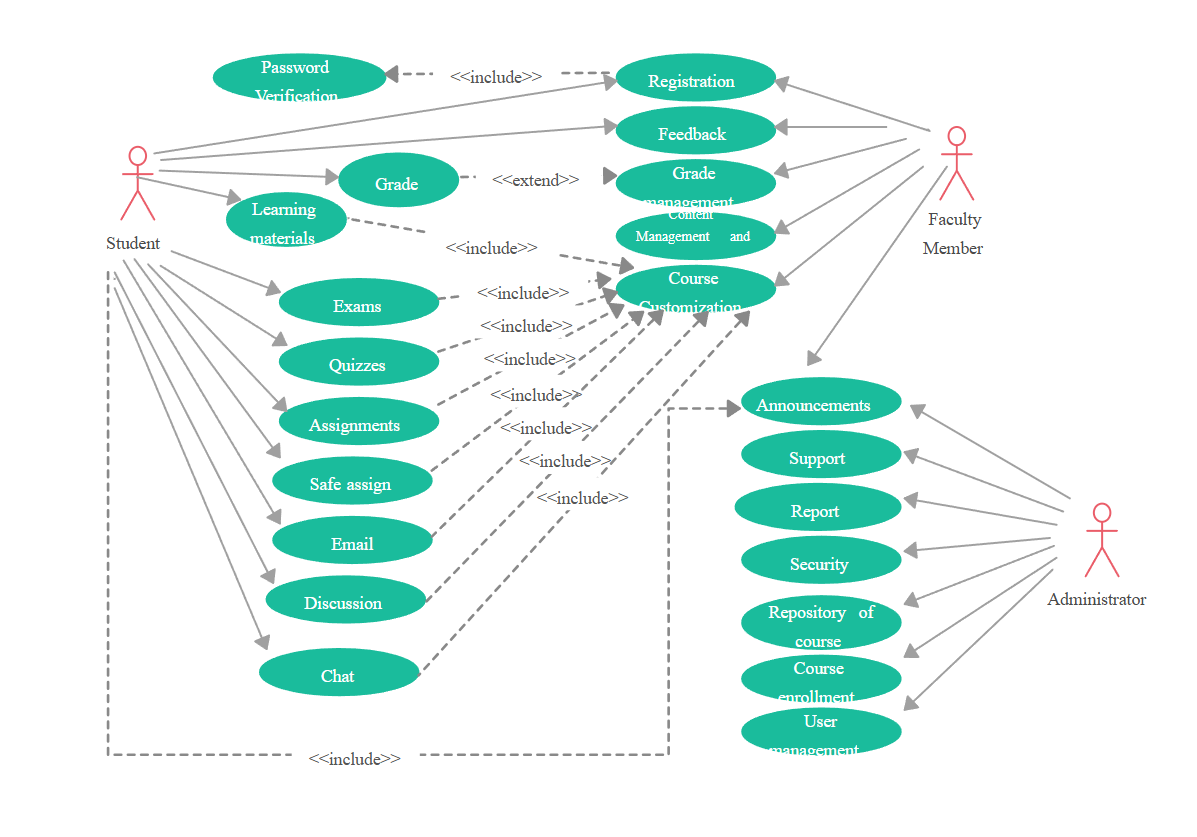
The Class Diagram and Sequence Diagram is designed Successfully for the E-Learning site.

| **EXP NO:** **7** | **DESIGNING USE-CASE AND ACTIVITY DIAGRAMS**    **FOR PROJECT STRUCTURE** |
| --- | --- |

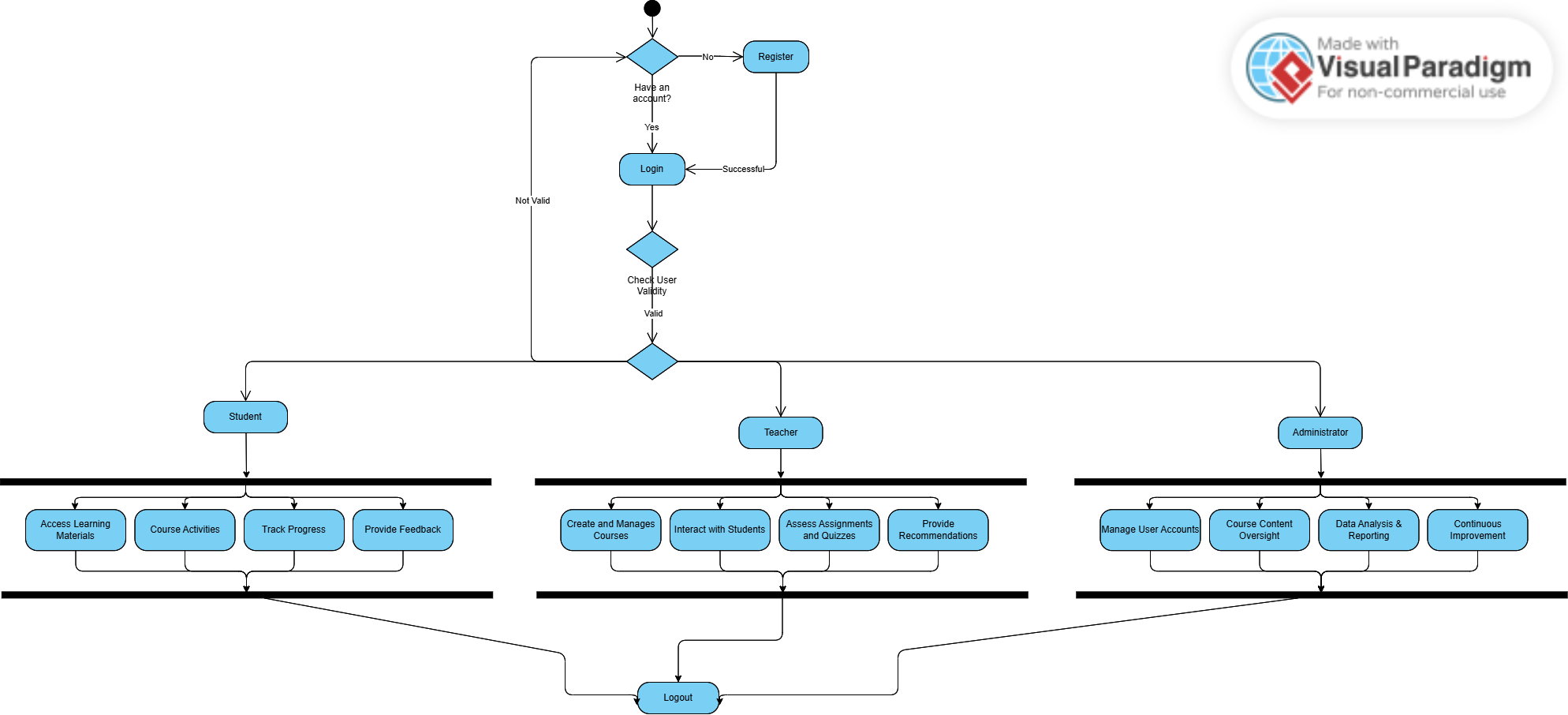
## AIM

To Design an Use-Case Diagram and Activity Diagram for the given Project.

**7A. Use-Case Diagram**



## 7B. Activity Diagram



## RESULT

The Use-Case Diagram and Activity Diagram is designed Successfully for the E-Learning site.

| **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 Login

•

## 2. Define User Interactions

• Each test case simulates a real user behaviour (e.g., logging in, submitting quizzes, viewing results)

## 3. Design Happy Path Test Cases

* Focused on validating that all core functionalities work correctly under normal conditions
* Example: Player registers and logs in, submits quizzes and views results

## 4. Design Error Path Test Cases

* Simulate invalid inputs, system issues or failed actions to ensure proper error handling.
* Example: Login with invalid credentials, submission without attachments, unauthorized access to admin panel.

## 5. Break Down Steps and Expected Results

• Each test case includes a clear sequence of actions and expected results. • Ensures both manual testers and automation tools can follow the process easily.

## 6. Use Clear Naming and IDs

• Test cases are uniquely identifies (e.g., TC01 – Valid Login, TC03 –

Invalid Password). • Facilities easy mapping to features and tracking in Azure DevOps.

## 7. Separate Test

• Grouped by functionality such as: o Login and Registration o Quizzes Submission o Viewing

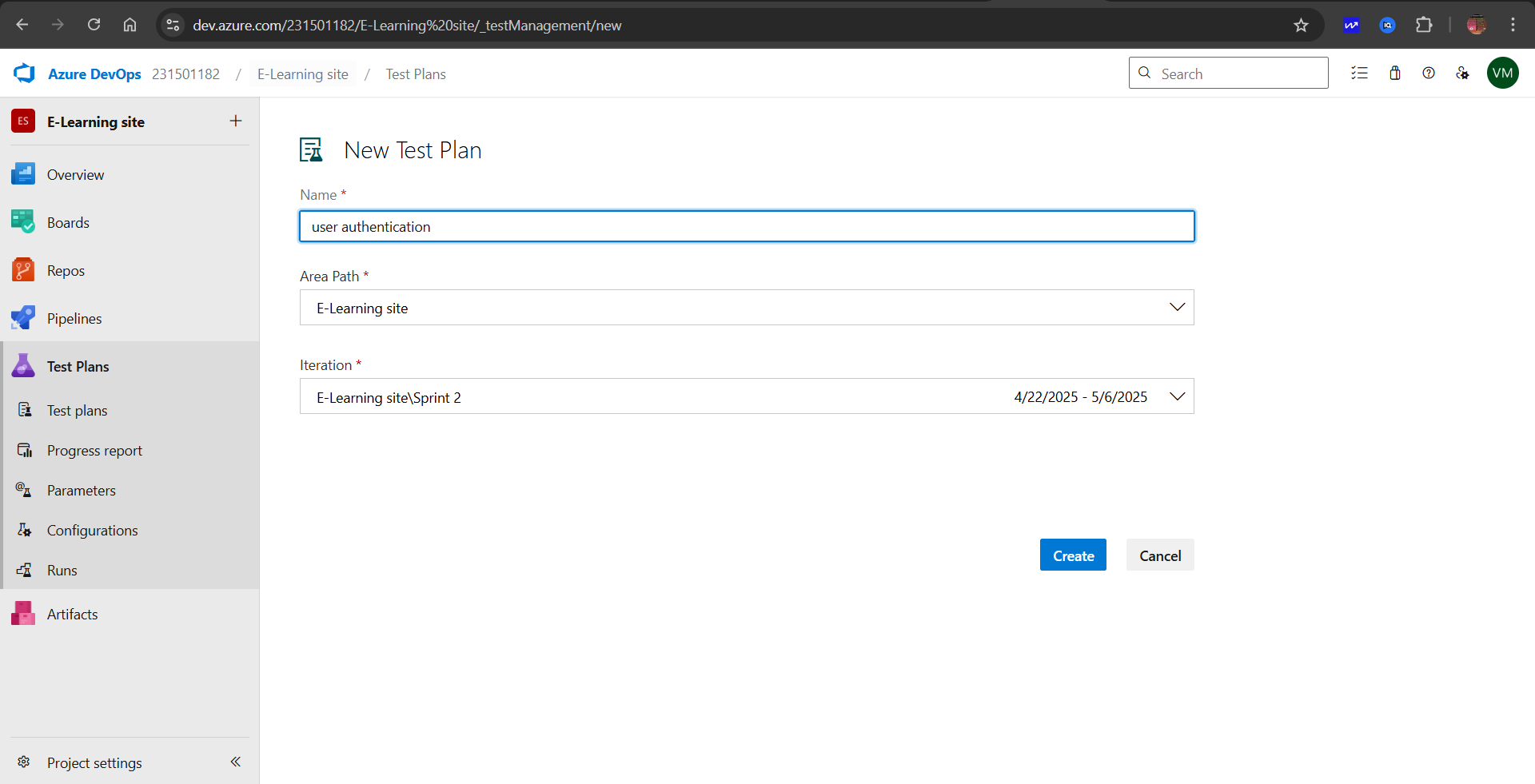
Results o Admin Functions • Improves organization and enables focused execution in Azure DevOps.

## 8. Prioritize and Review

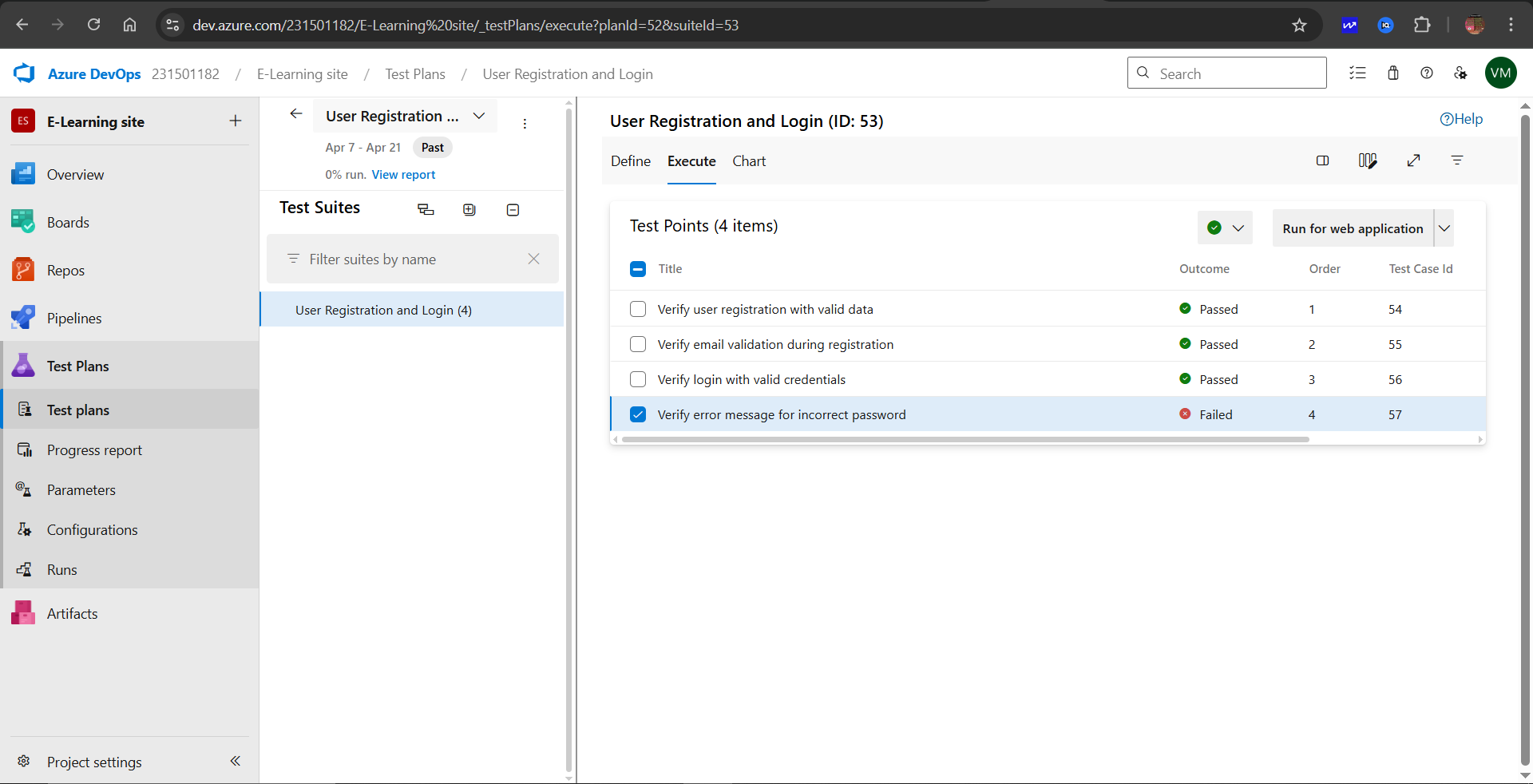
* High-priority assigned to critical workflows like login, quizzes and results.
* Reviewed for completeness, accuracy and alignment with user stories and features definition.

.

## 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.

E-Learning site– Test Plans

## USER STORIES

* **As an admin**, I want to log in to manage courses, videos, and users (ID: 1).
* **As a learner**, I want to register and log in so I can access learning content securely (ID: 2).
* **As an admin**, I want to assign roles (admin/learner) to control platform access (ID: 3).
* **As a learner**, I want to enroll in courses and track my learning progress (ID: 4).
* **As an admin**, I want to upload and manage course materials and videos (ID: 5).
* **As a learner**, I want to take quizzes and view instant feedback to test my knowledge (ID: 6).
* **As an admin**, I want to view reports of learner performance across all courses (ID: 7).
* **As a learner**, I want to download my certificate after completing a course (ID: 8).

## ✅ TEST SUITES

### 🔹 Test Suite: TS01 – Login & Registration (ID: 101)

**1. TC01 – Valid Registration** **Action:** Navigate to the registration page. Fill in valid username, email, and password. Click "Register".  
 **Expected Result:** User is registered successfully and redirected to login page.  
 **Type:** Happy Path

**2. TC02 – Invalid Email Format** **Action:** Enter name and password but use an invalid email format.  
 **Expected Result:** System displays validation error "Enter a valid email address."  
 **Type:** Error Path

**3. TC03 – Login with Correct Credentials** **Action:** Enter registered email and password. Click "Login".  
 **Expected Result:** User is authenticated and redirected to the dashboard.  
 **Type:** Happy Path

**4. TC04 – Login with Incorrect Password** **Action:** Enter valid email and wrong password.  
 **Expected Result:** Error message “Invalid username or password” is shown.  
 **Type:** Error Path

### 🔹 Test Suite: TS02 – Course Enrollment (ID: 105)

**1. TC05 – Enroll in a Course** **Action:** Log in as learner. Select a course and click “Enroll”.  
 **Expected Result:** User is enrolled and course appears on their dashboard.  
 **Type:** Happy Path

**2. TC06 – Enroll Without Login** **Action:** Try enrolling in a course without logging in.  
 **Expected Result:** Redirected to login page or shown "Login required" message.  
 **Type:** Negative Path

### 🔹 Test Suite: TS03 – Video Playback (ID: 110)

**1. TC07 – Start Course Video** **Action:** Open an enrolled course and click on a lesson video.  
 **Expected Result:** Video loads and plays correctly.  
 **Type:** Happy Path

**2. TC08 – Video Controls** **Action:** Play, pause, and seek video.  
 **Expected Result:** Controls function as expected with no lag or error.  
 **Type:** Happy Path

### 🔹 Test Suite: TS04 – Quizzes & Assessments (ID: 115)

**1. TC09 – Complete Quiz with All Questions Answered** **Action:** Start a quiz, answer all questions, click "Submit".  
 **Expected Result:** Answers are saved, and score is calculated.  
 **Type:** Happy Path

**2. TC10 – Submit Without Answering** **Action:** Start quiz, skip questions, and try to submit.  
 **Expected Result:** Prompt “You have unanswered questions” is shown.  
 **Type:** Error Path

**3. TC11 – View Quiz Feedback** **Action:** After submitting quiz, view results.  
 **Expected Result:** Correct/incorrect answers and score are displayed.  
 **Type:** Happy Path

### 🔹 Test Suite: TS05 – Progress Tracking (ID: 120)

**1. TC12 – View Course Progress** **Action:** Navigate to "My Courses". Check progress bar for enrolled course.  
 **Expected Result:** Progress bar reflects modules completed.  
 **Type:** Happy Path

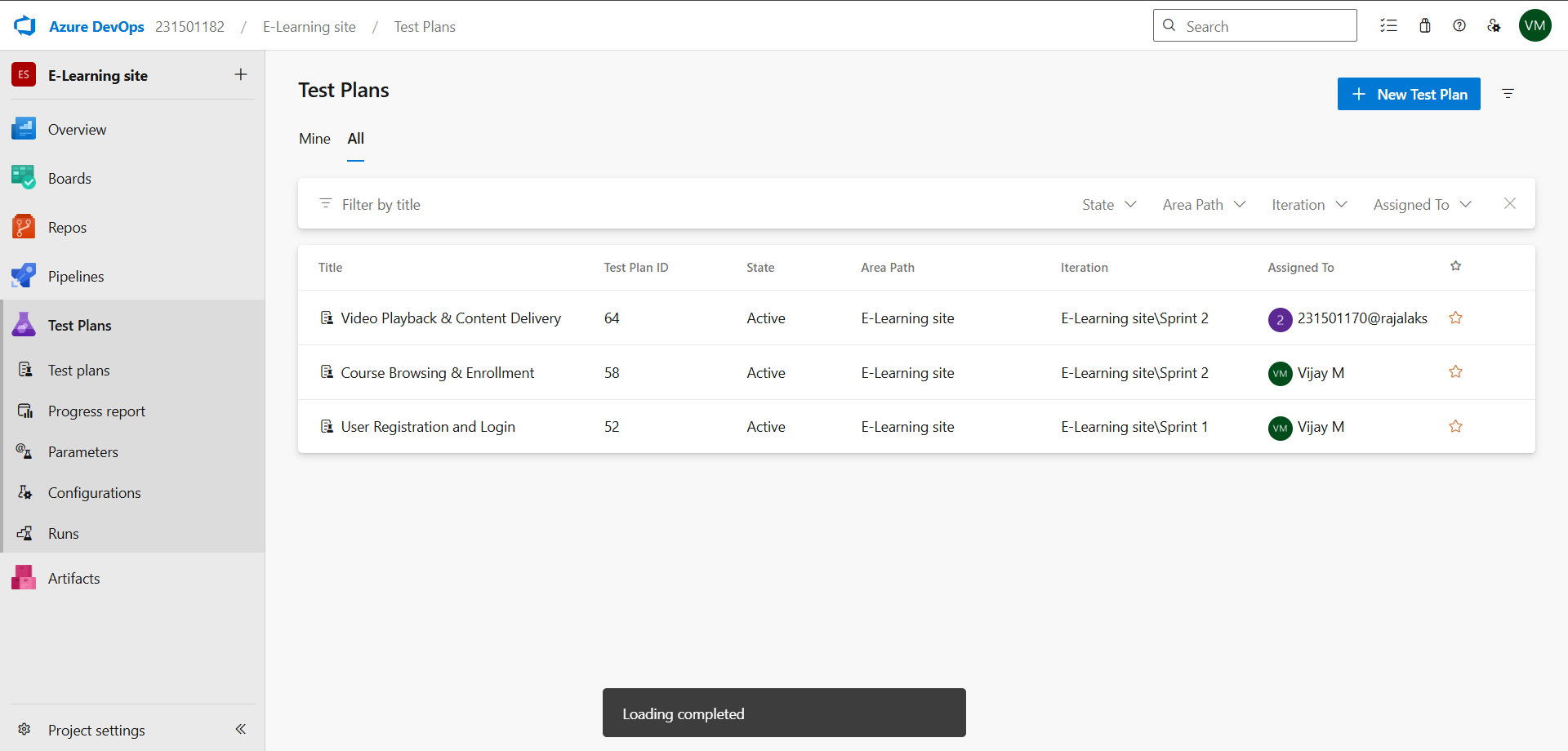
**2. TC13 – Progress Not Updated After Watching Video** **Action:** Watch a lesson video and return to dashboard.  
 **Expected Result:** Progress should update; otherwise, log error.  
 **Type:** Error Path

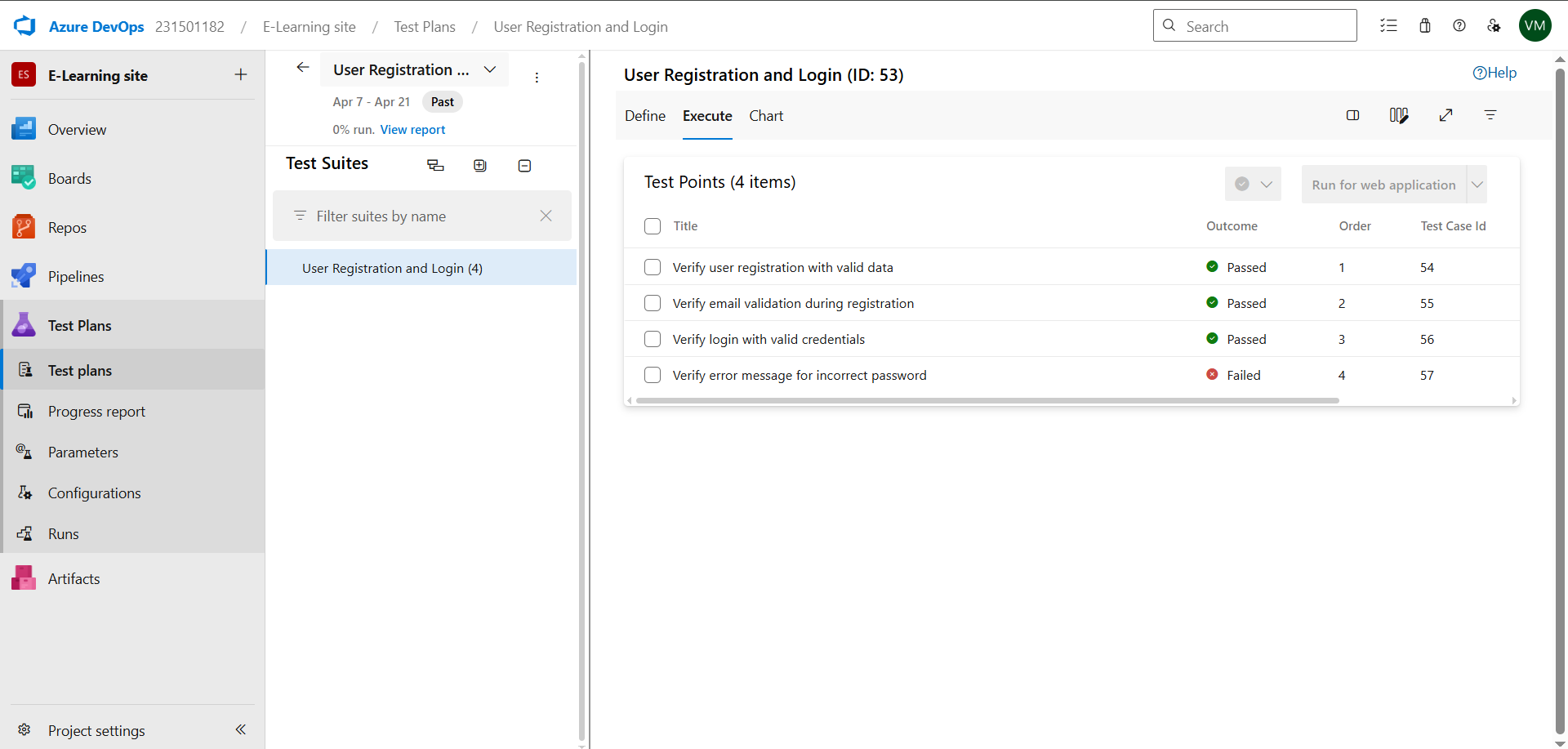
### 🔹 Test Suite: TS06 – Certificate Generation (ID: 125)

**1. TC14 – Generate Certificate After Course Completion** **Action:** Complete all modules and quiz. Click "Generate Certificate".  
 **Expected Result:** Certificate is created and available for download.  
 **Type:** Happy Path

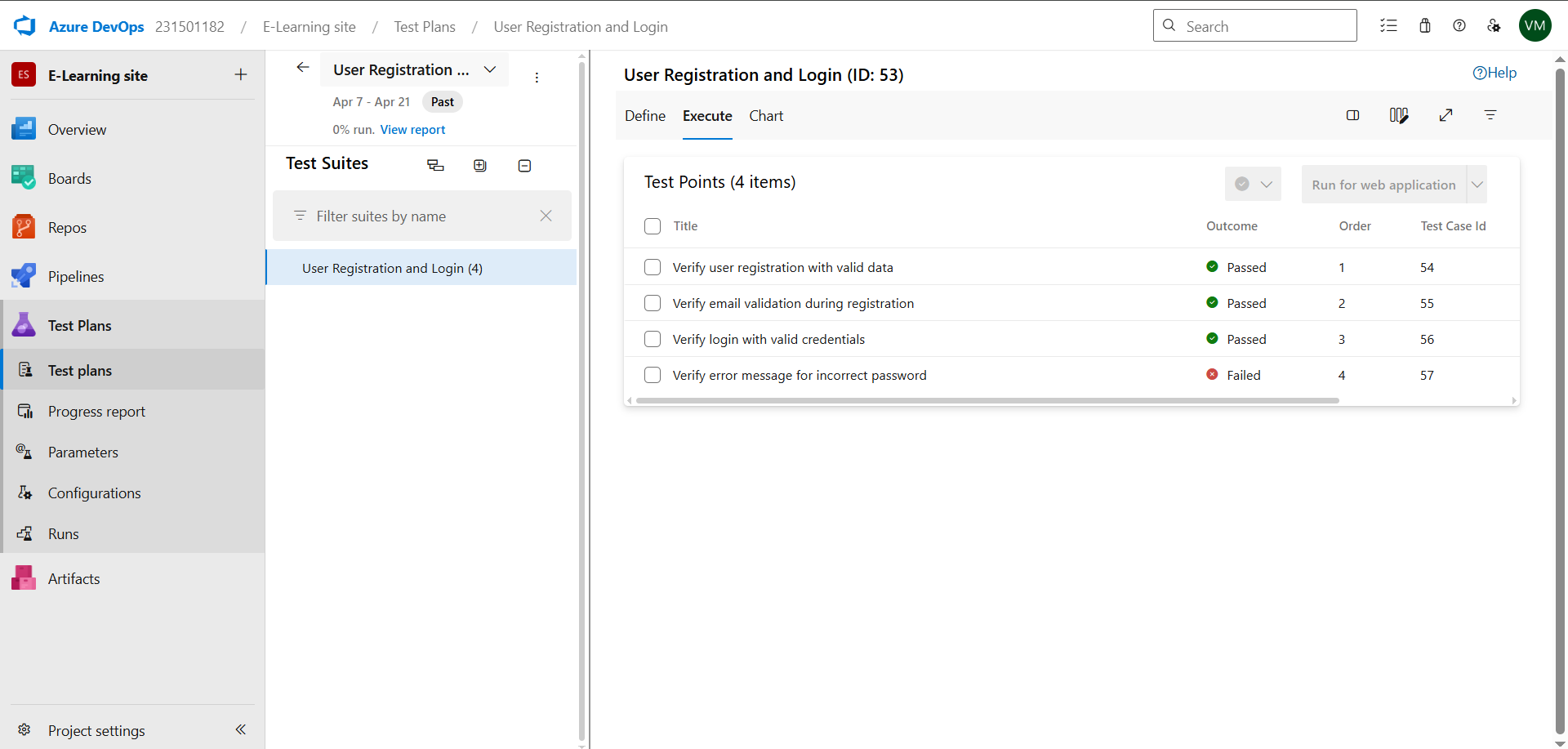
**2. TC15 – Certificate Access Before Completion** **Action:** Try to access certificate before completing all requirements.  
 **Expected Result:** System prevents access and shows message “Complete the course to download certificate.”  
 **Type:** Error Path

## Test Cases







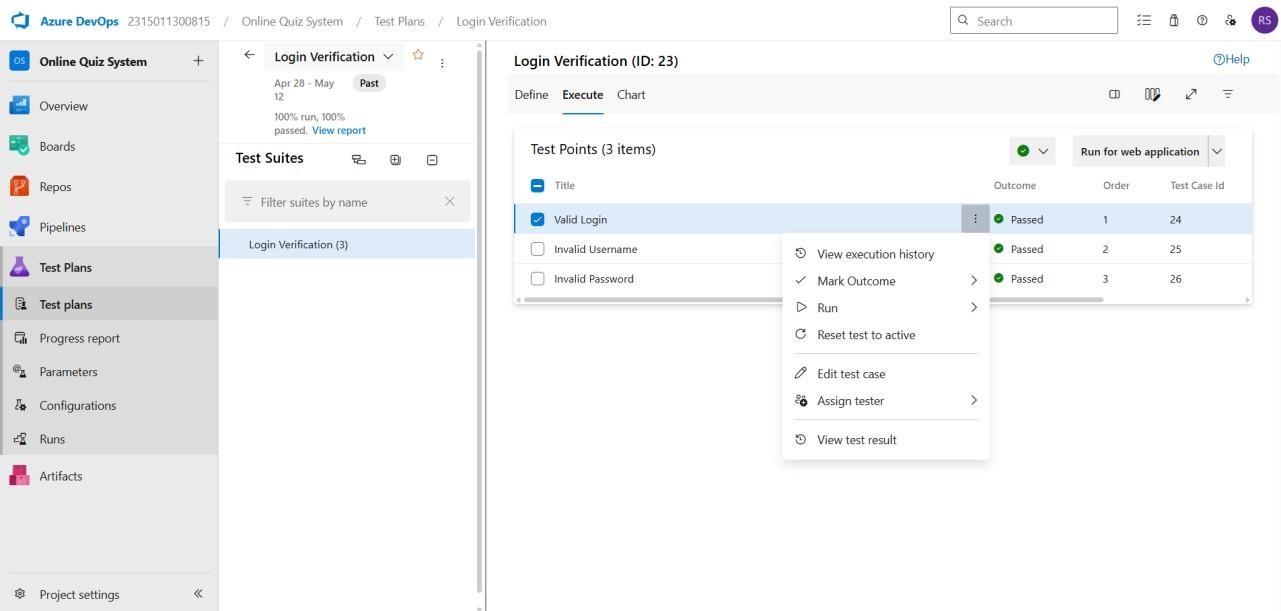


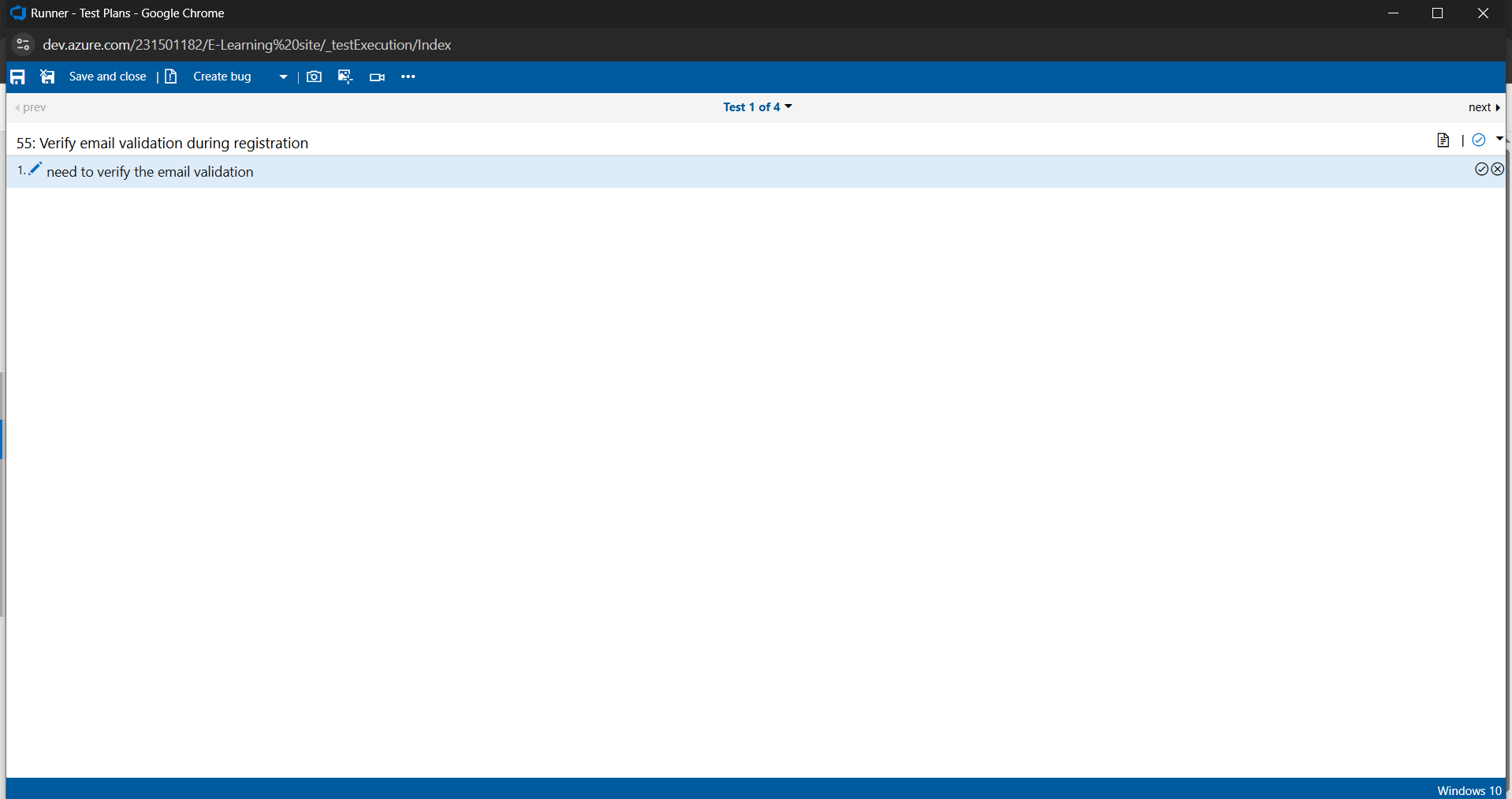


## 4.Installation of test

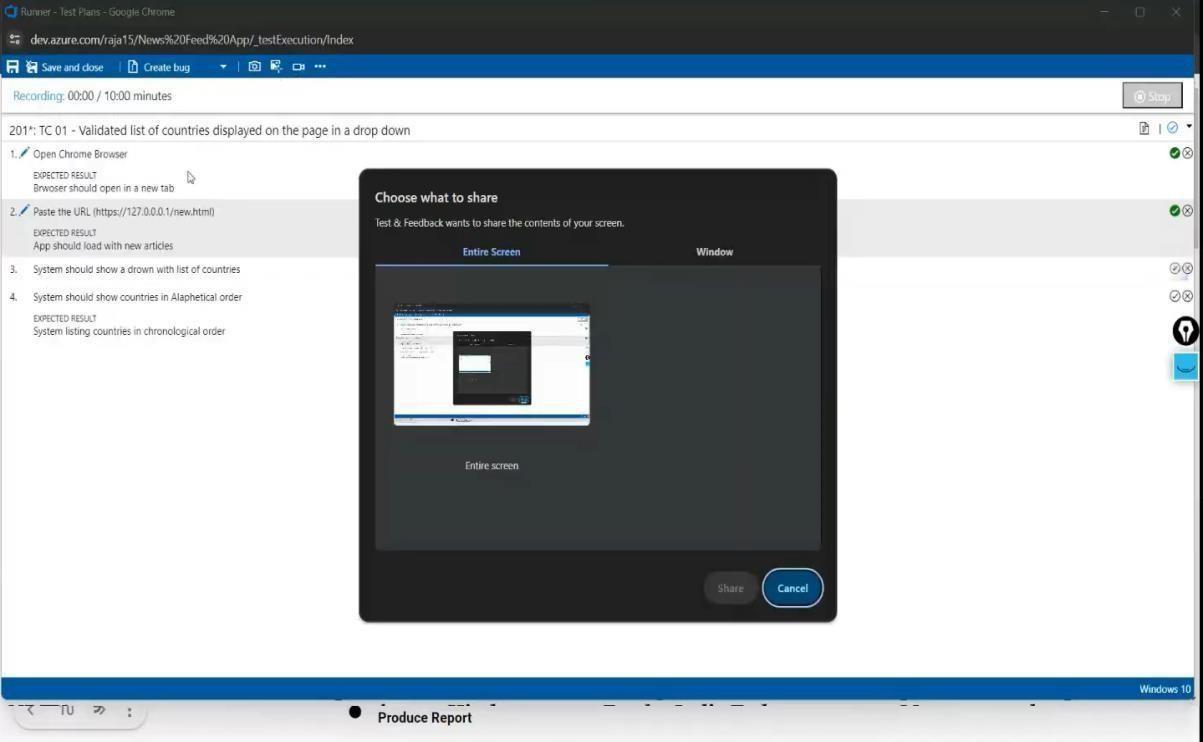


## 5.Running the test cases



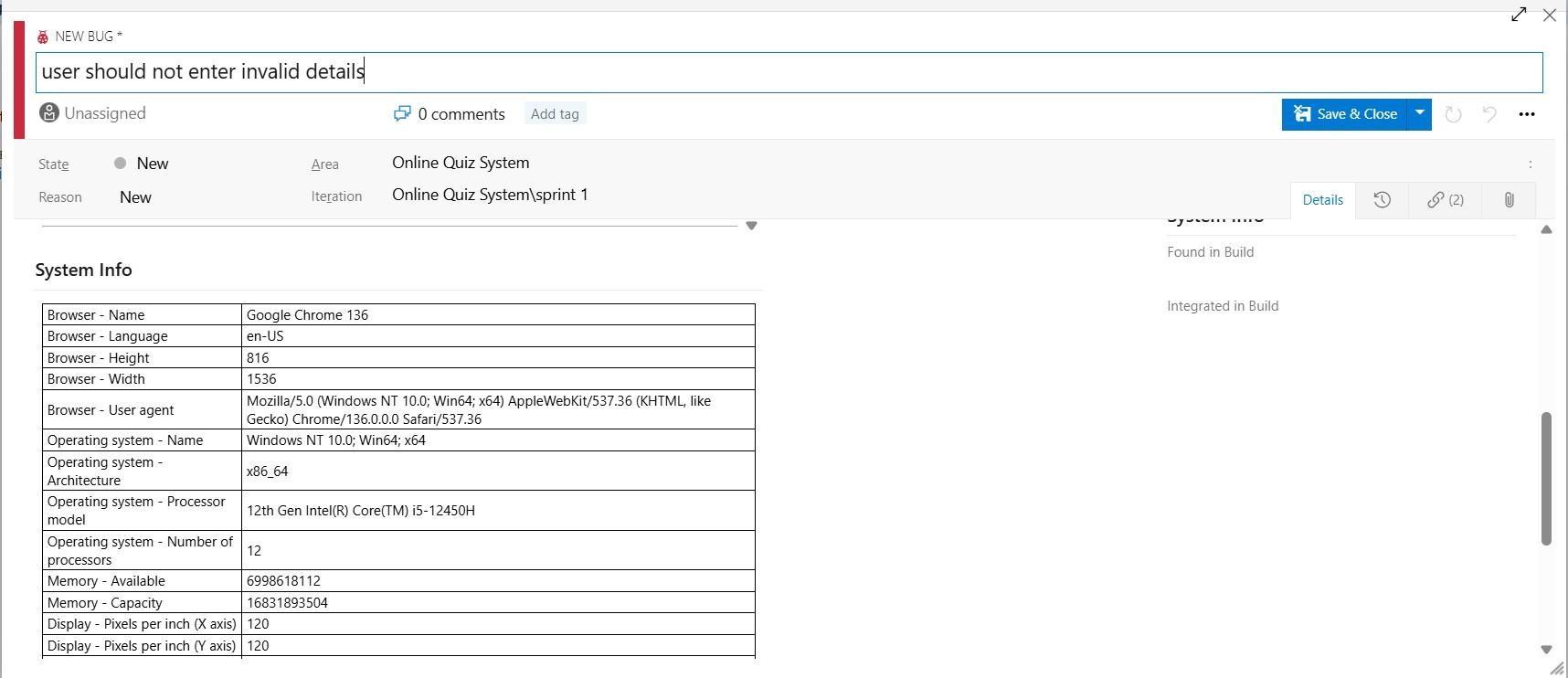


## 6.Recording the test case

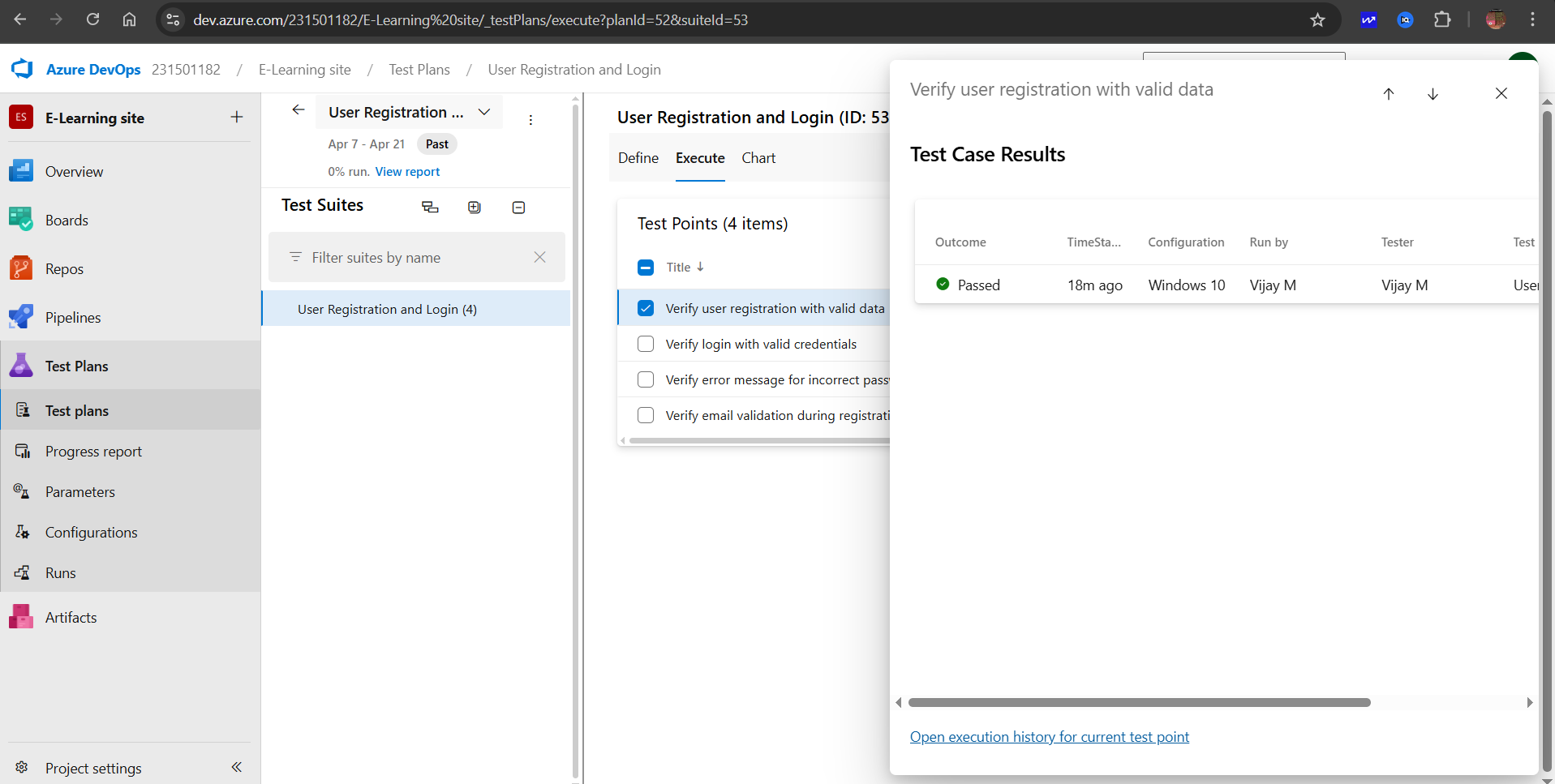


## 7.Creating the bug

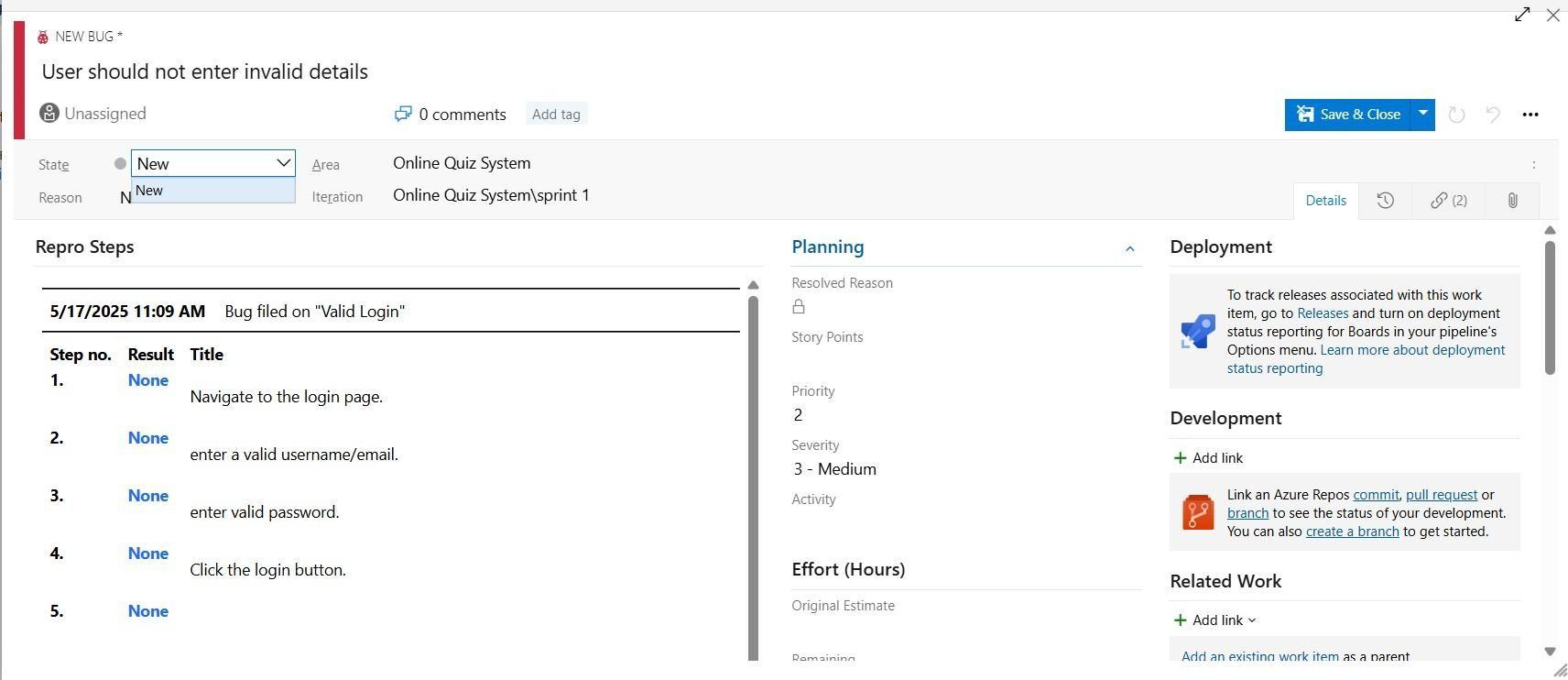




## 8.Test case results

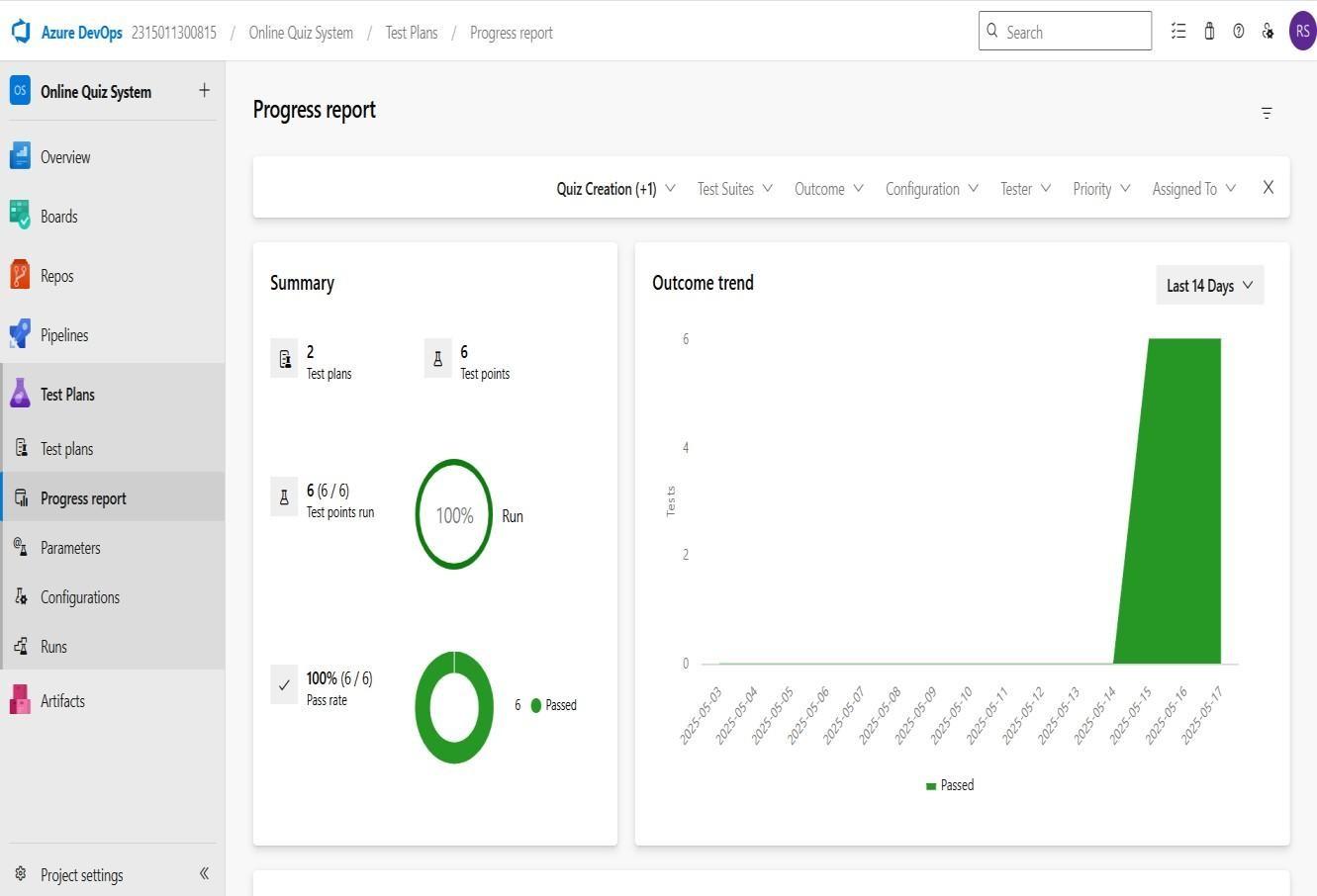


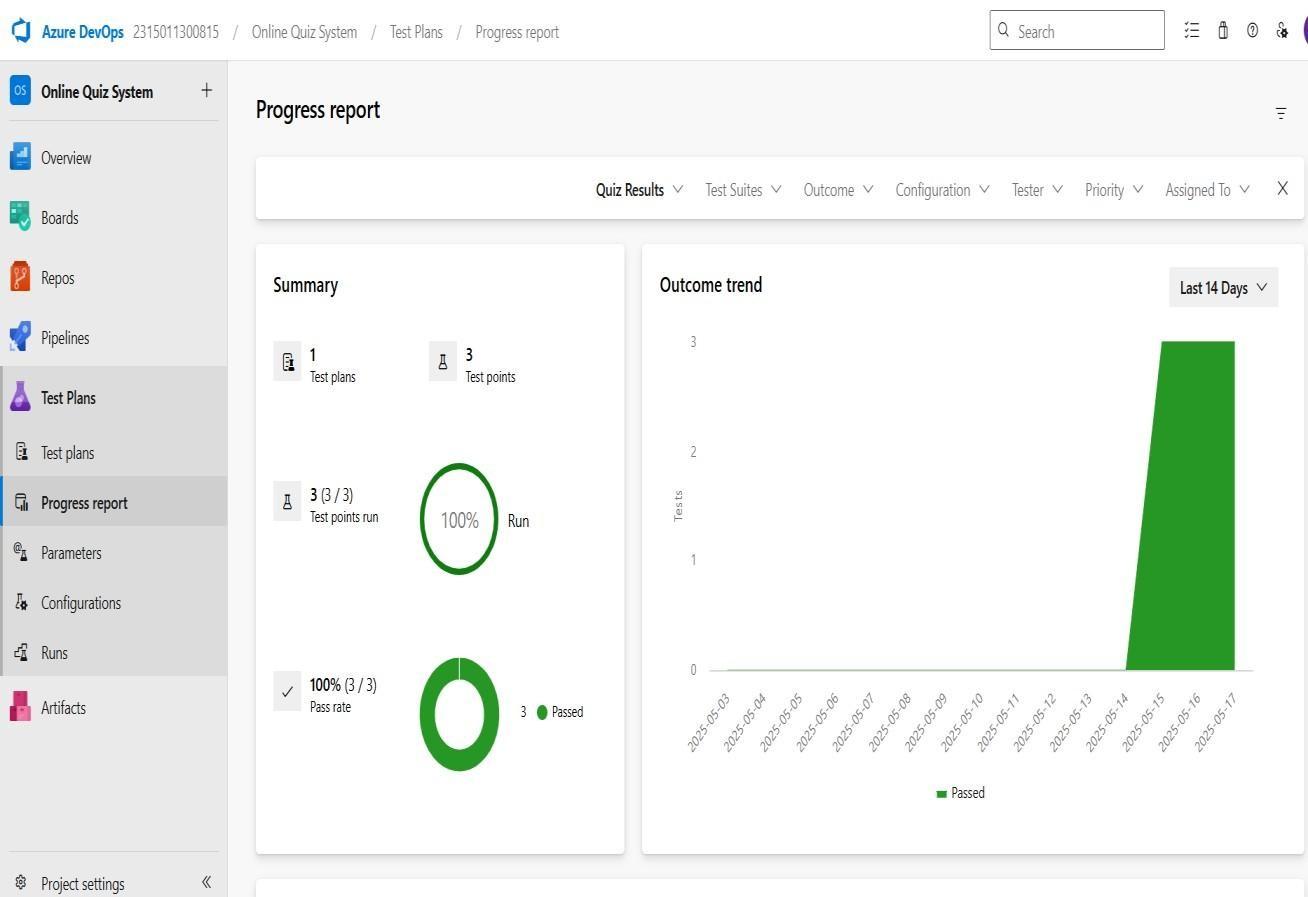
## 9.Test report summary



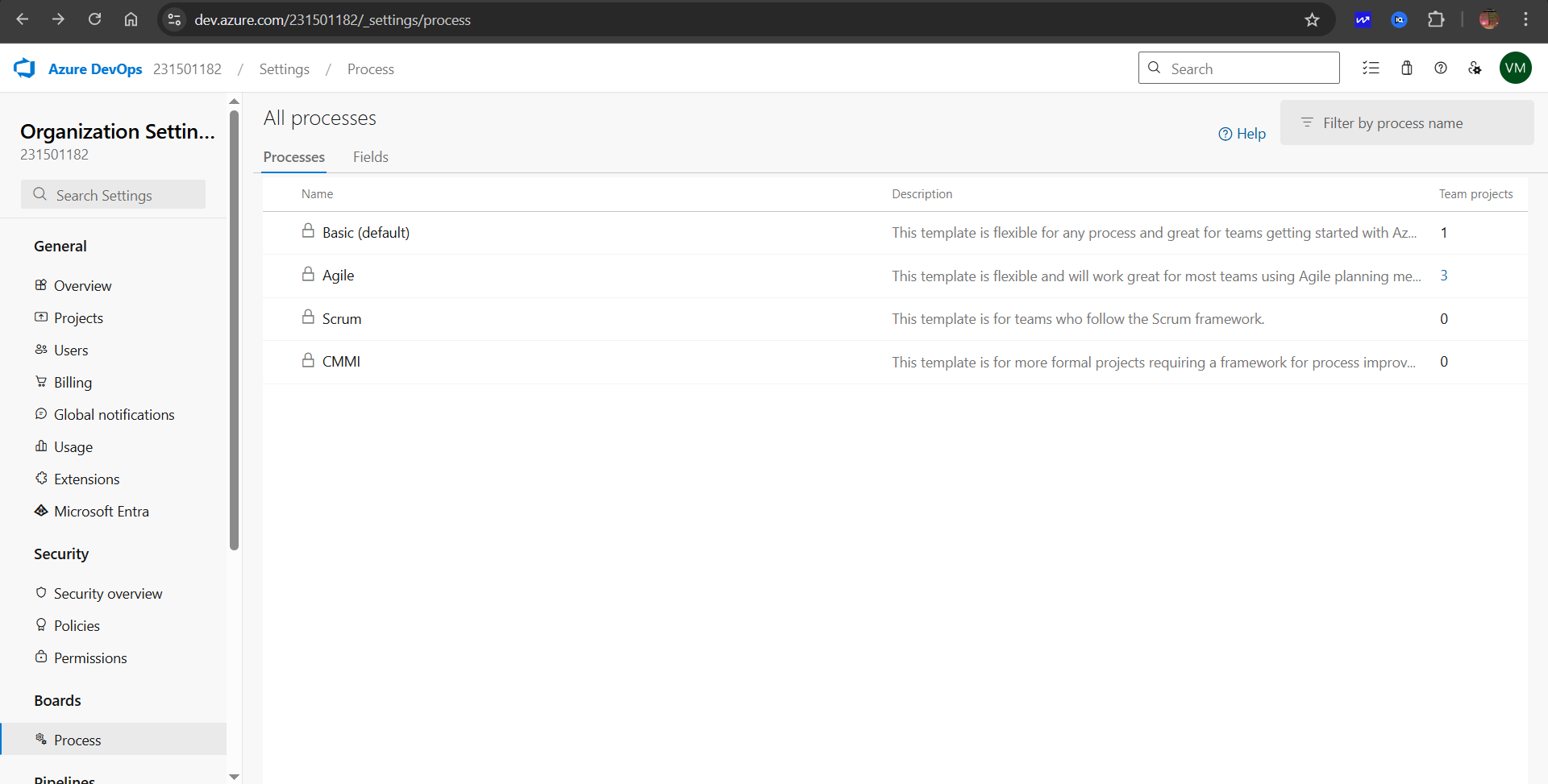
• Assigning bug to the developer and changing state

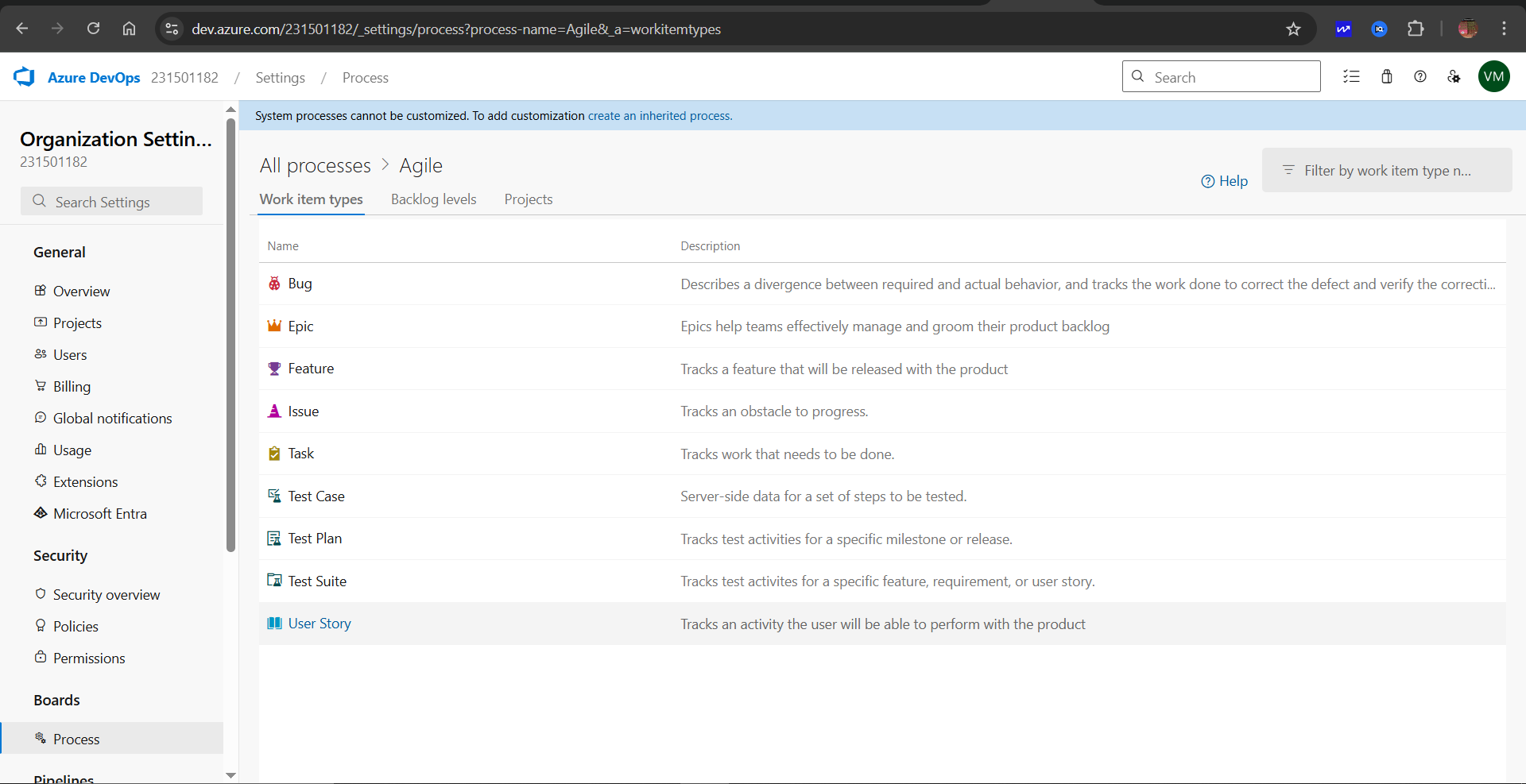
## 10.Progress report





## 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** | **CI/CD PIPELINES IN AZURE** |
| --- | --- |

## AIM

To implement a Continuous Integration and Continuous Deployment (CI/CD) pipeline in Azure DevOps for automating the build, testing, and deployment process of the E-Learning site, ensuring faster delivery and improved software quality**.**

## PROCEDURE

**Steps to Create and implement pipelines in Azure:**

1. Sign in to Azure DevOps and Navigate to Your Project

Log in to [dev.azure.com,](https://dev.azure.com/) select your organization, and open the project where your E-Learning site code resides.

1. Connect a Code Repository (Azure Repos or GitHub)

Ensure your application code is stored in a Git-based repository such as Azure Repos or GitHub. This will be the source for triggering builds and deployments in your pipeline.

1. Create a New Pipeline

Go to the Pipelines section on the left panel and click “Create Pipeline”.

Choose your source (e.g., Azure Repos Git or GitHub), and then select the repository containing your project code.

1. Choose the Pipeline Configuration

You can select either the YAML-based pipeline (recommended for version control and automation) or the Classic Editor for a GUI-based setup. If using YAML, Azure DevOps will suggest a template or allow you to define your own.

1. Define Build Stage (CI - Continuous Integration) from YAML file.

1. Install dependencies (e.g., npm install, dotnet restore).

1. Build the application (dotnet build, npm run build).

1. Run unit tests (dotnet test, npm test).

1. Publish build artifacts to be used in the release stage.

1. Save and Run the Pipeline for the First Time

Save the YAML or build definition and click “Run”.

Azure will fetch the latest code and execute the defined build and test stages.

1. Configure Continuous Deployment (CD)

Navigate to the Releases tab under Pipelines and click “New Release Pipeline”. Add an Artifact (from the build stage) and create a new Stage (e.g., Development, Production).

1. Configure the CD stage with deployment tasks such as deploying to Azure App Service, running database migrations or scripts, and restarting services using the Azure App Service Deploy task linked to your subscription and app details.

1. Set Triggers and Approvals

Enable continuous deployment trigger so the release pipeline runs automatically after a successful build. For production environments, configure pre-deployment approvals to ensure manual verification before release.

1. Monitor Pipelines and Manage Logs

View all pipeline runs under the Runs section.

Check logs for build/test/deploy stages to debug any errors.

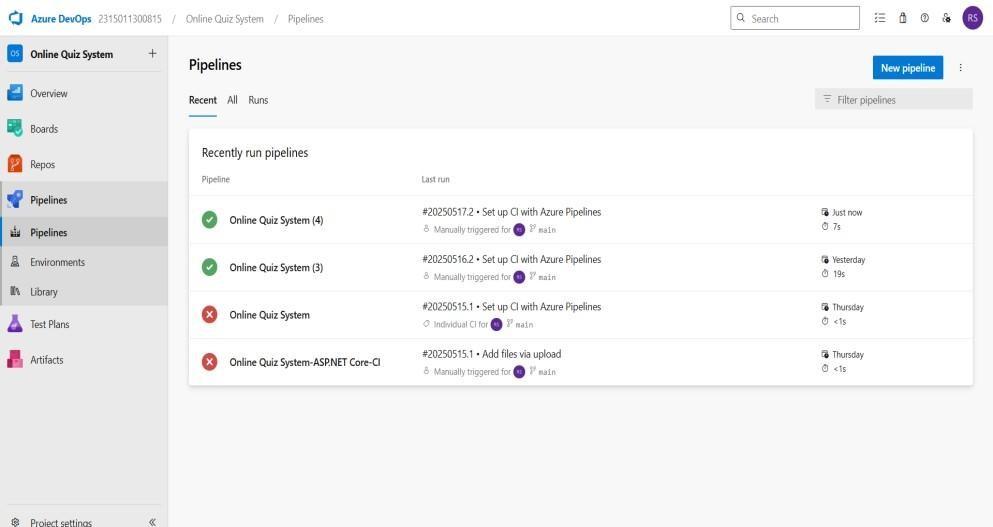
You can also integrate email alerts or Microsoft Teams notifications for build failures.

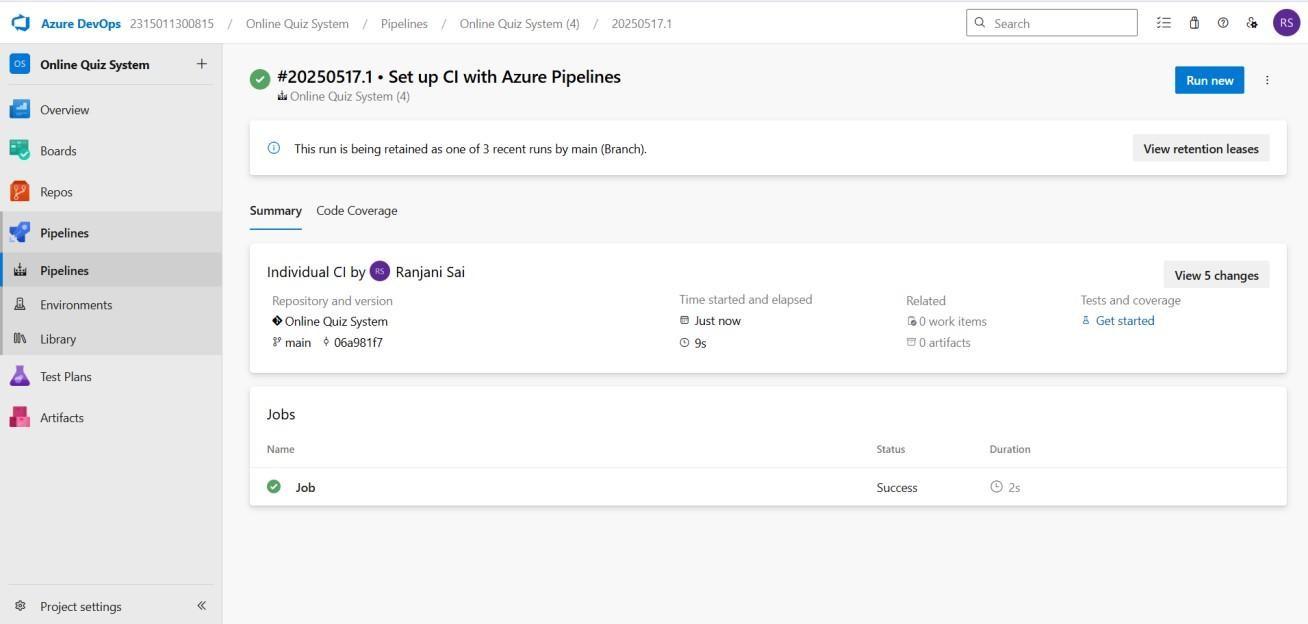
1. Review and Maintain Pipelines

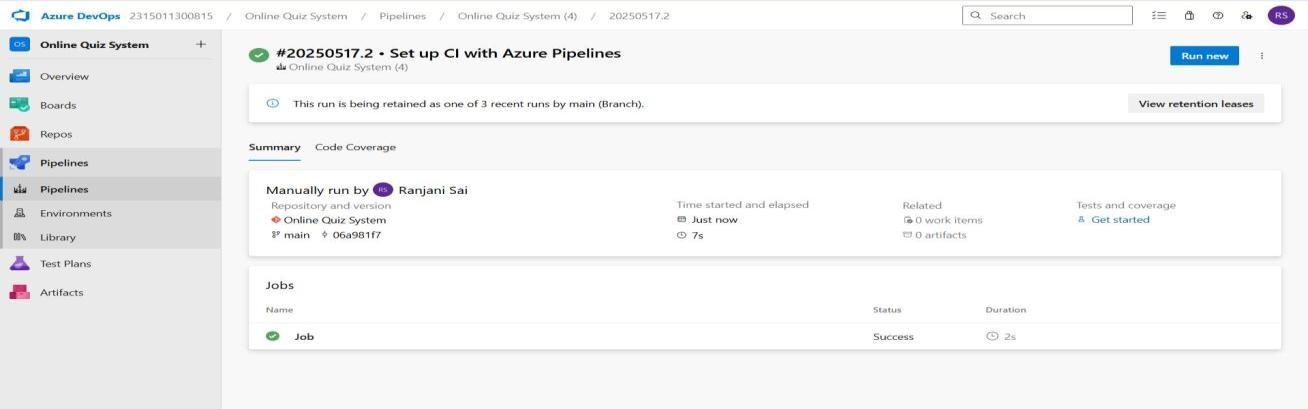
Regularly update your pipeline tasks or YAML configurations as your application grows.

Ensure pipeline runs are clean and artifacts are stored securely.

Integrate quality gates and code coverage policies to maintain code quality.







## RESULT

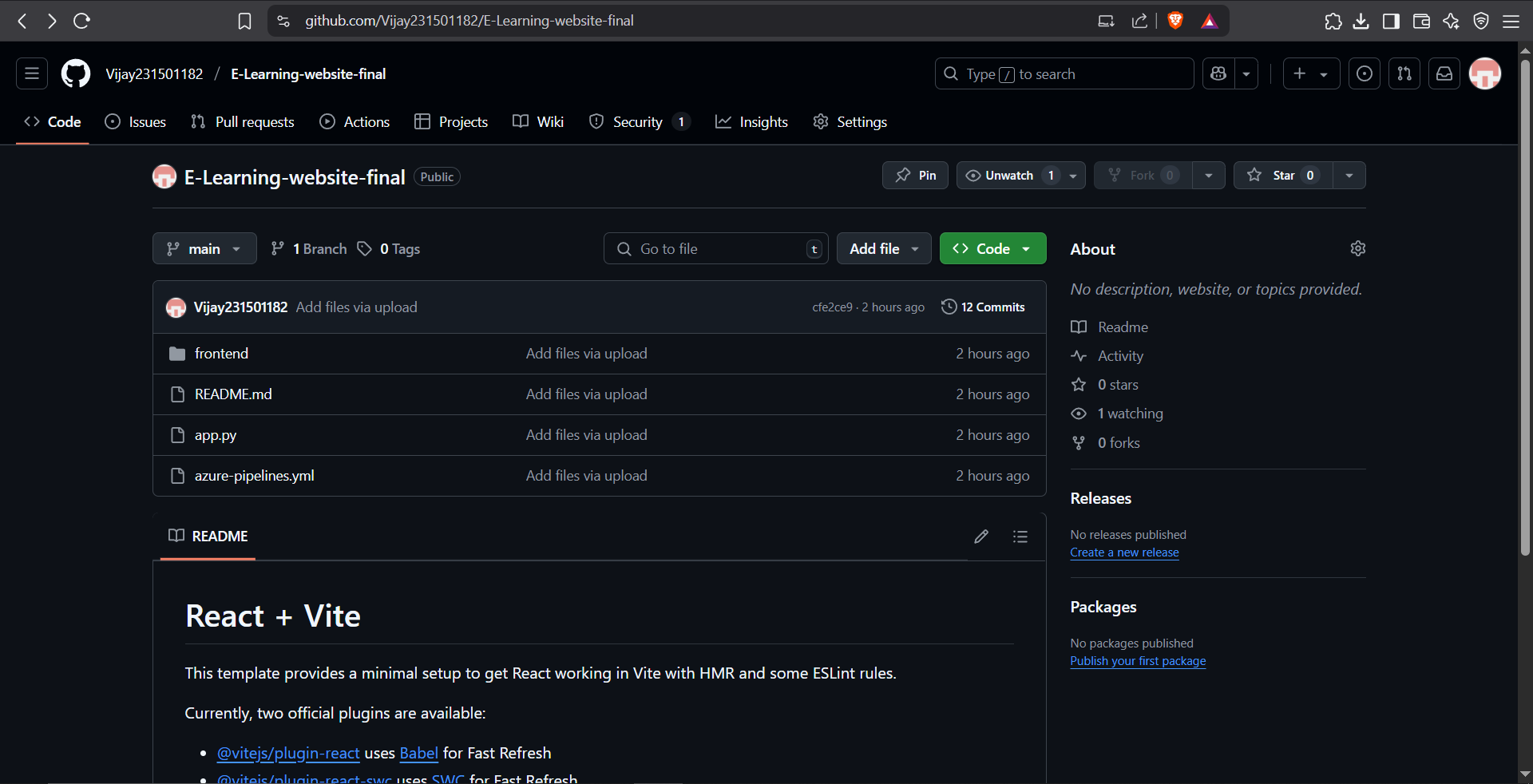
Thus, the pipelines for the given project **“E-Learning site”** has been executed successfully.

| **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 E-Learning site.

## 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.