EXP NO: 1	AZURE DEVOPS ENVIRONMENT SETUP

Aim:

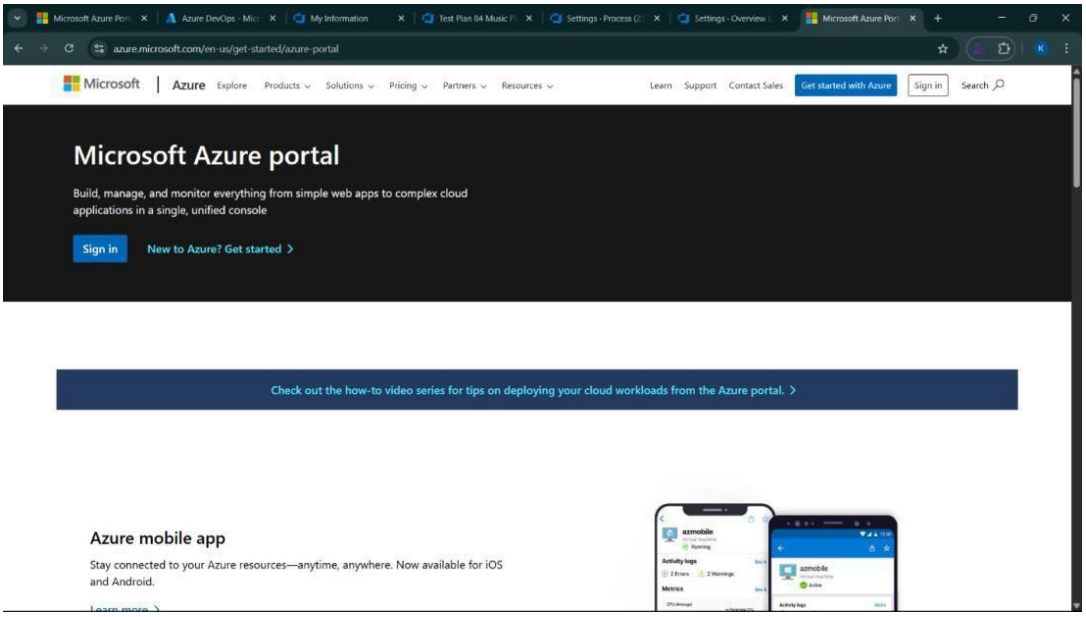
To set up and access the Azure DevOps environment by creating an organization through the Azure portal.

INSTALLATION

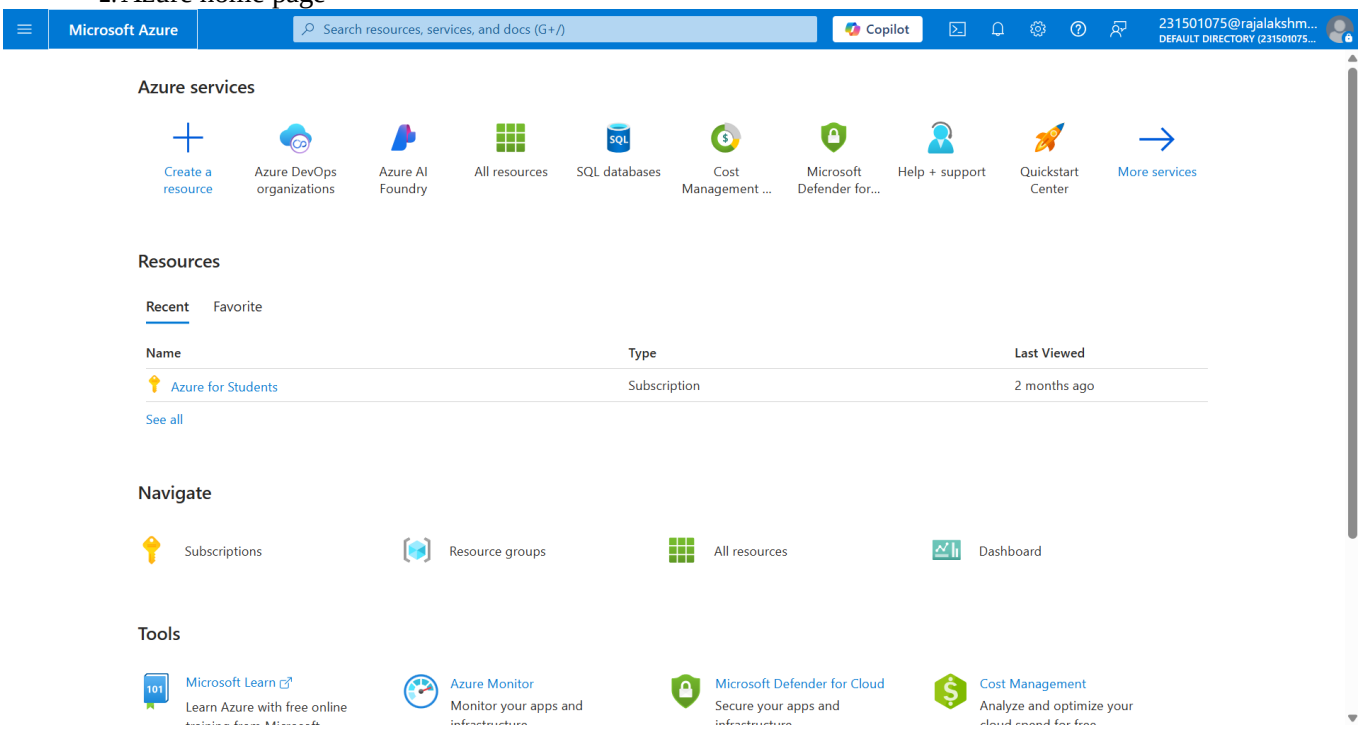
1. Open your web browser and go to the Azure website: https://azure.microsoft.com/en-us/get-started/azure-portal.

Sign in using your Microsoft account credentials.

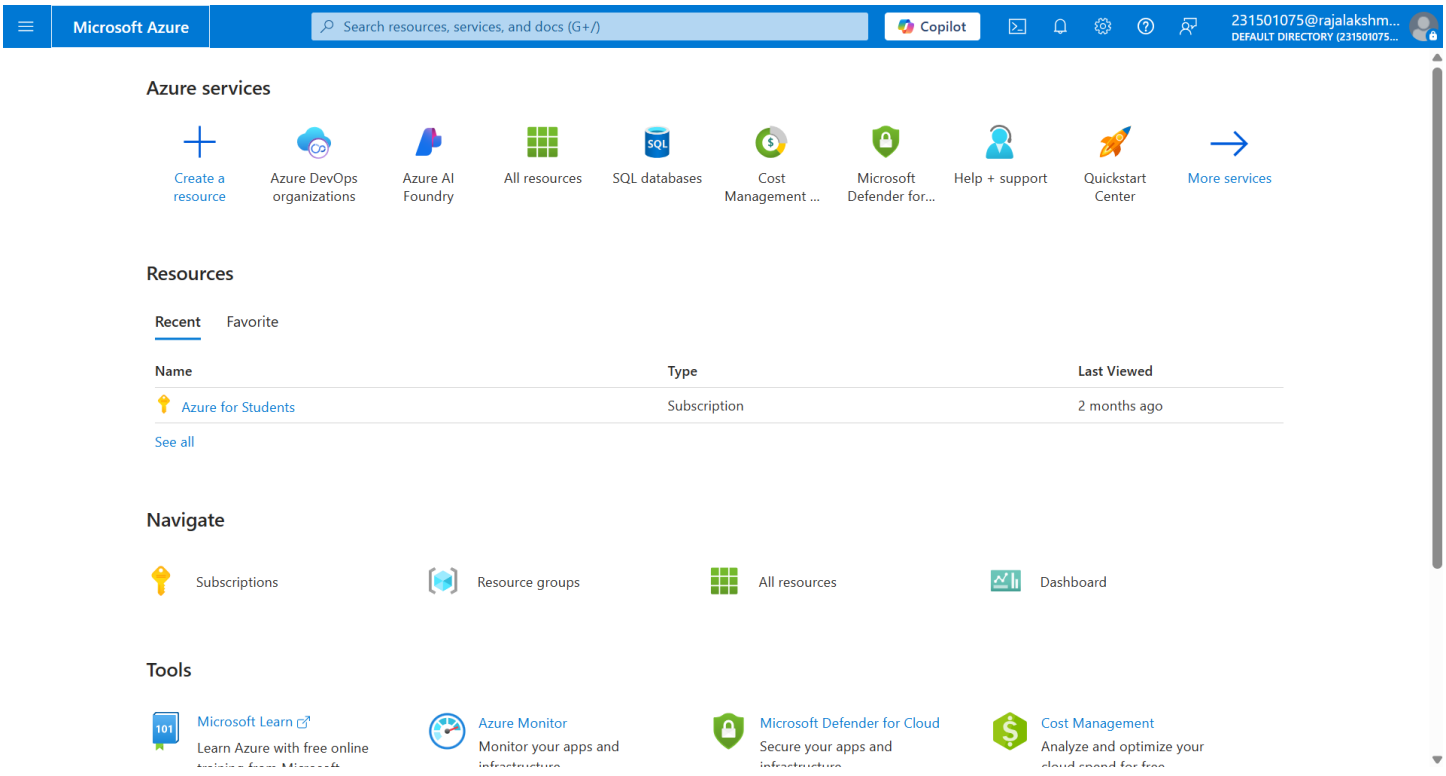
If you don't have a Microsoft account, you can create one here: https://signup.live.com/?lic=1



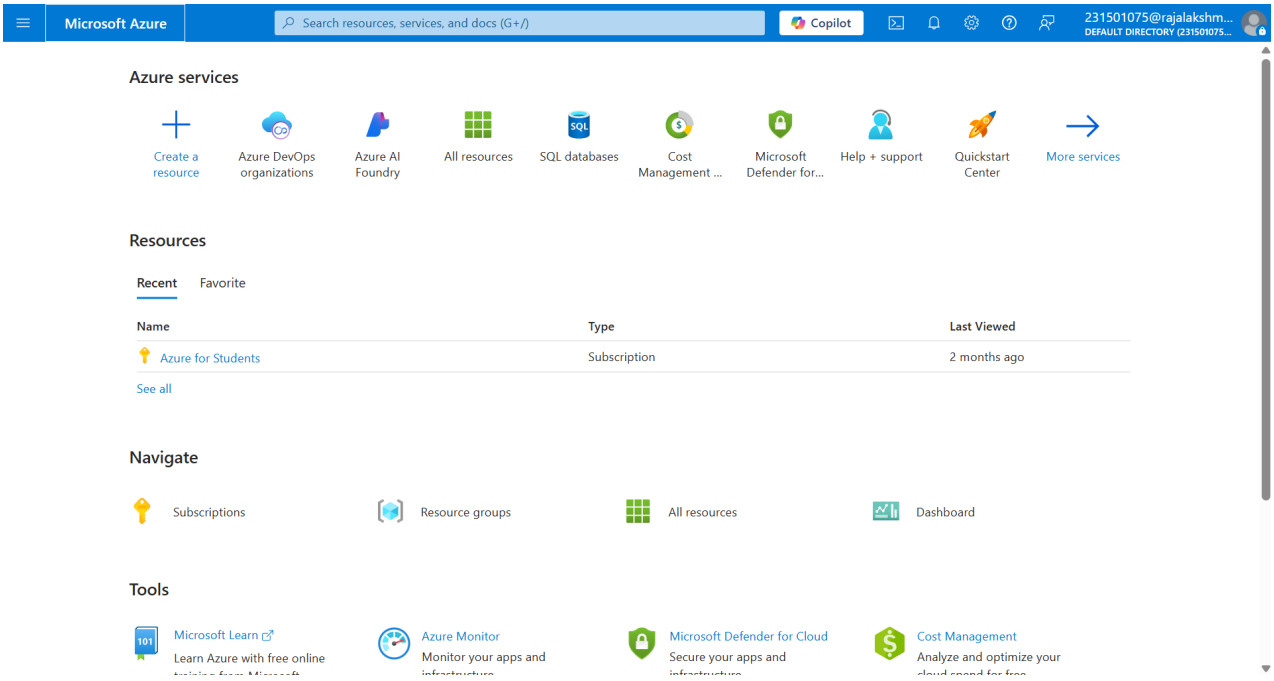
2. Azure home page



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:

2116231501075 CS23432

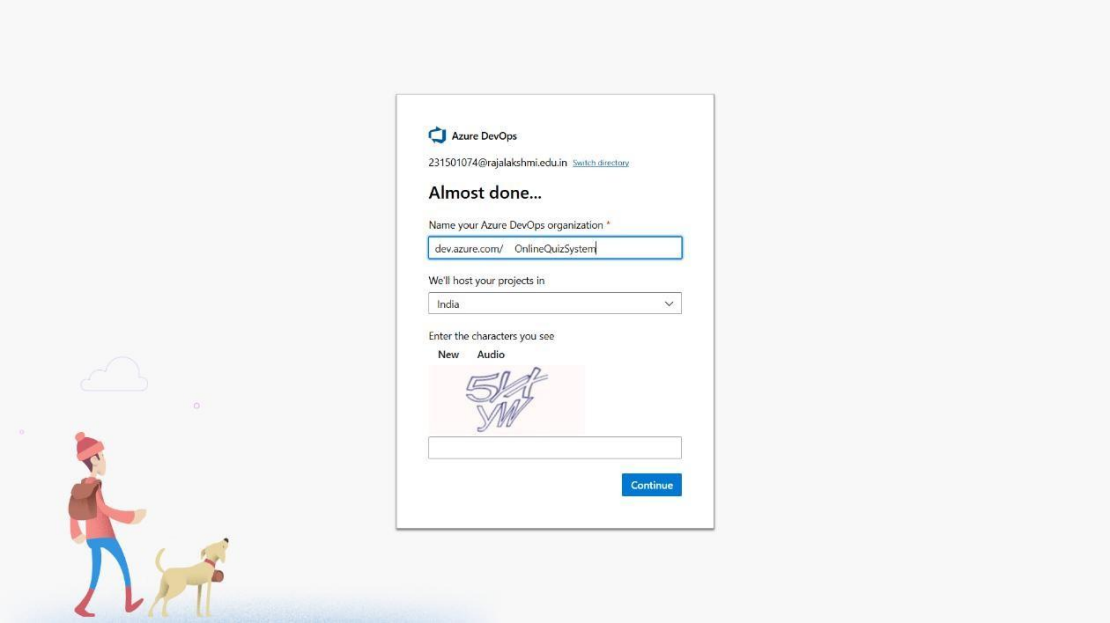
Successfully accessed the Azure DevOps environment and created a new organization through the Azure portal.

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.

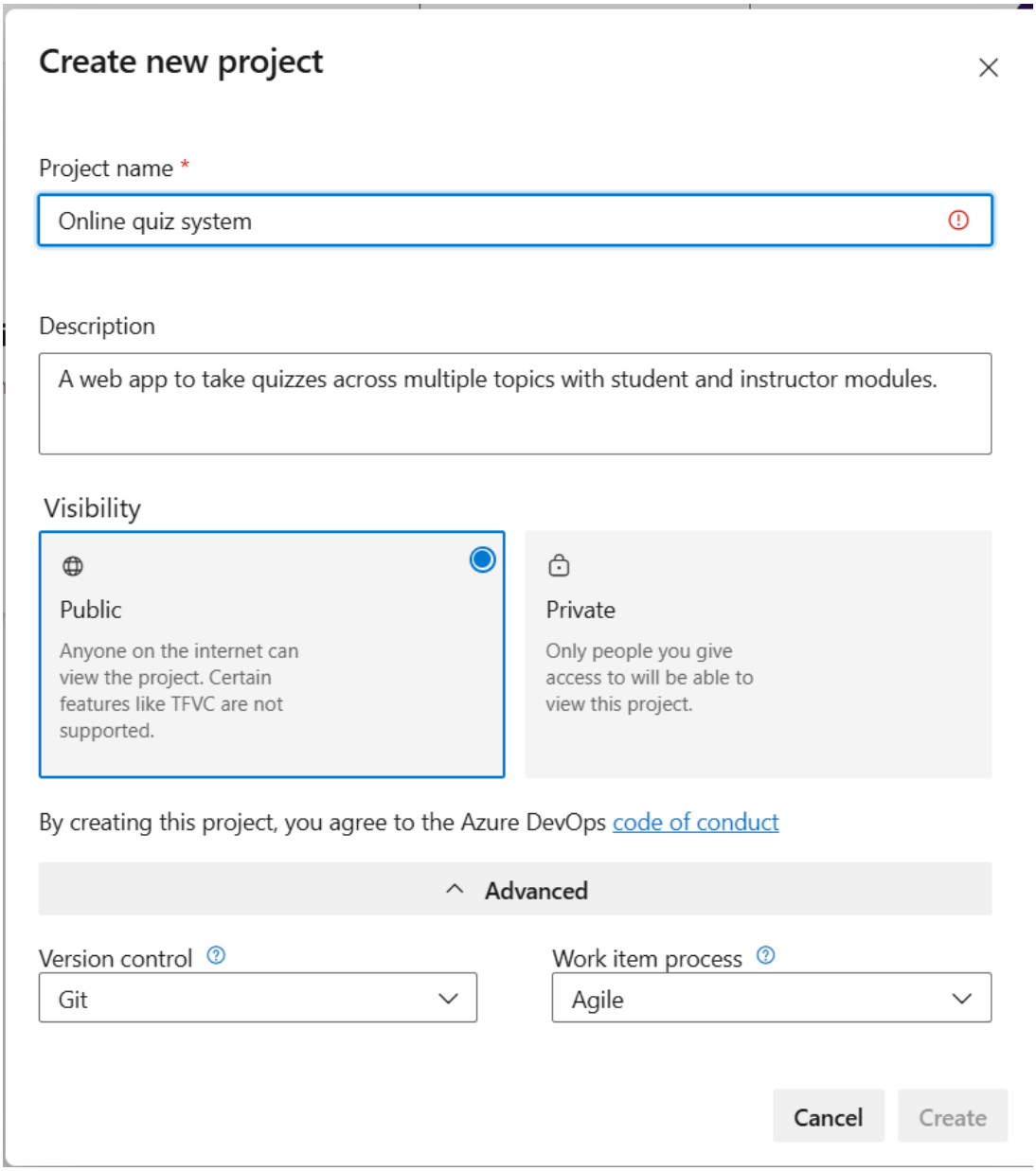
c. Enter the project name, description, and visibility options:

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

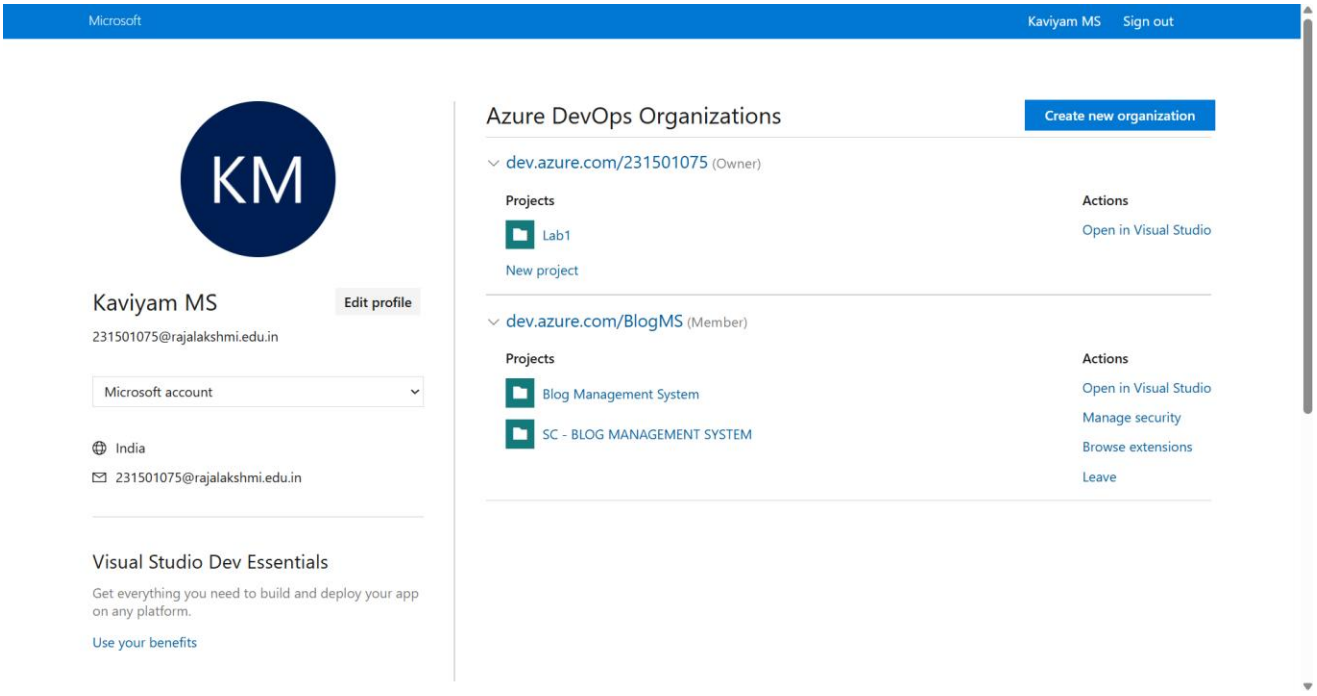
***Description**:* Optionally, add a description to provide more context about the project.

***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.

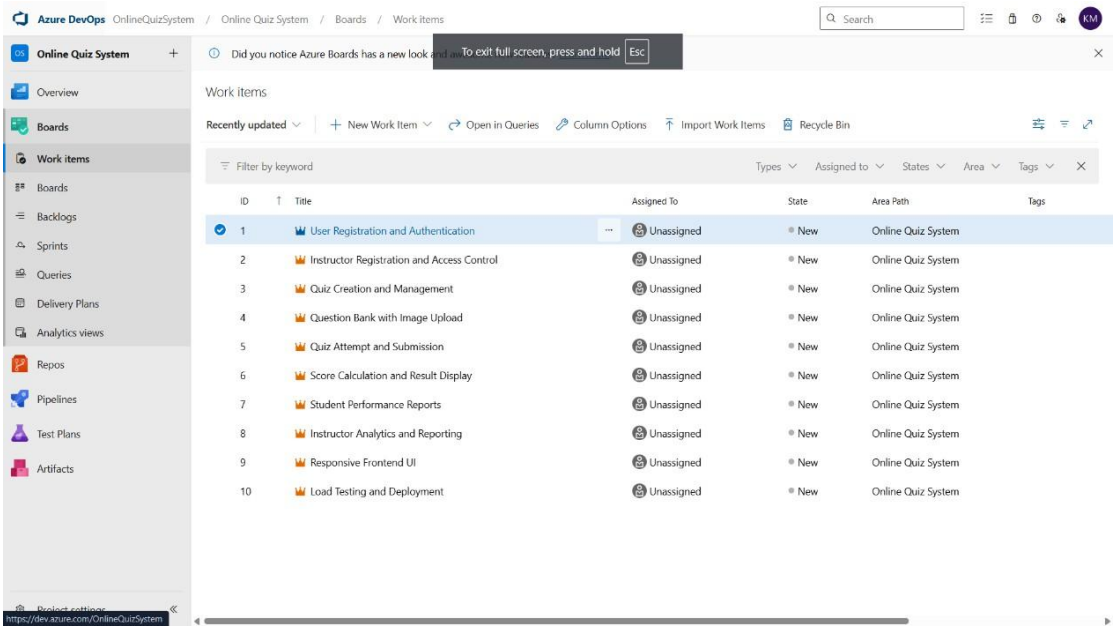


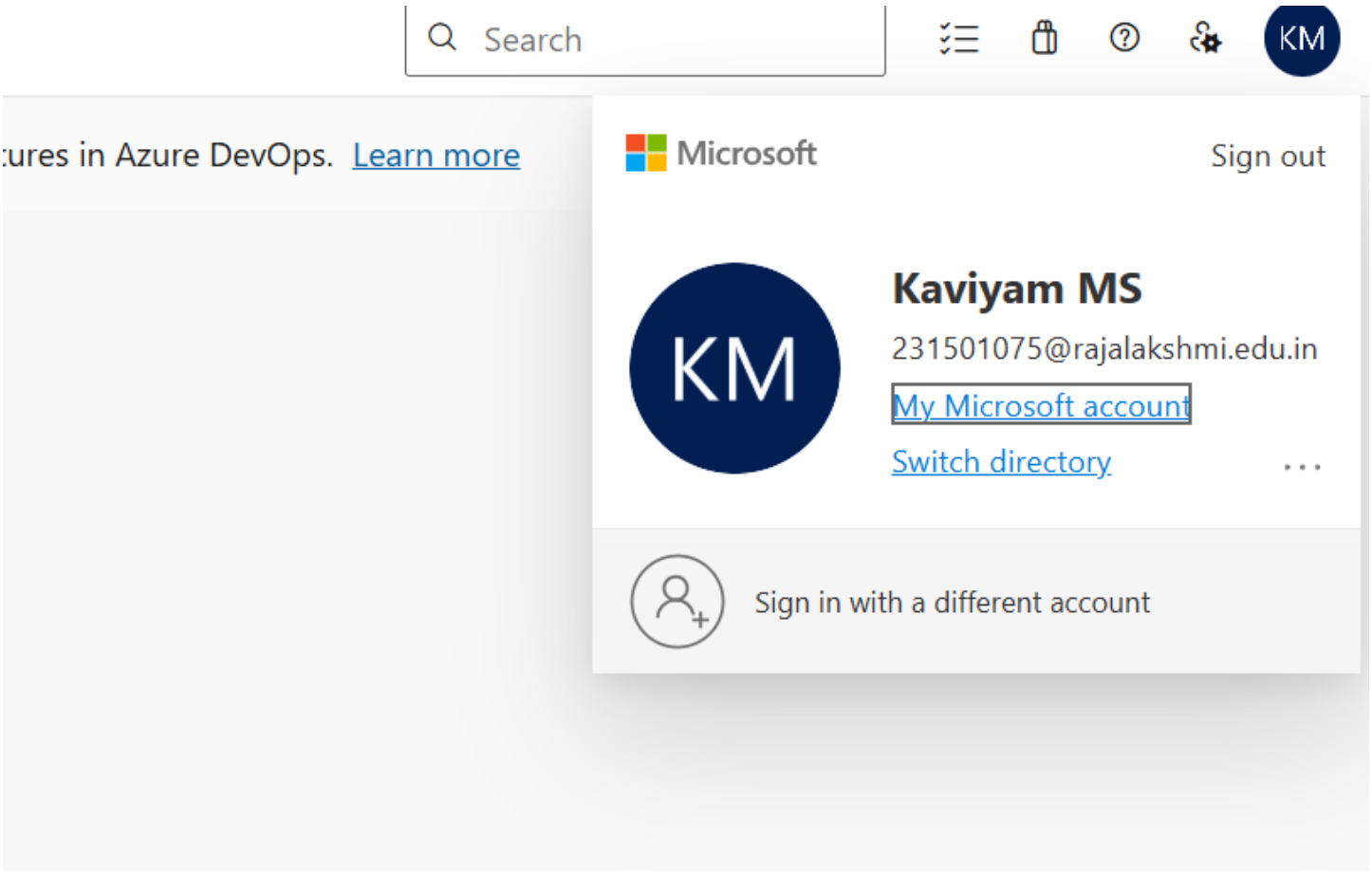
4. Project dashboard

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.




Search
KM
ures in Azure DevOps. Learn more
Microsoft
Sign out
KM
Kaviyam MS
231501075@rajalakshmi.edu.in
My Microsoft account
Switch directory
Sign in with a different account

Result:

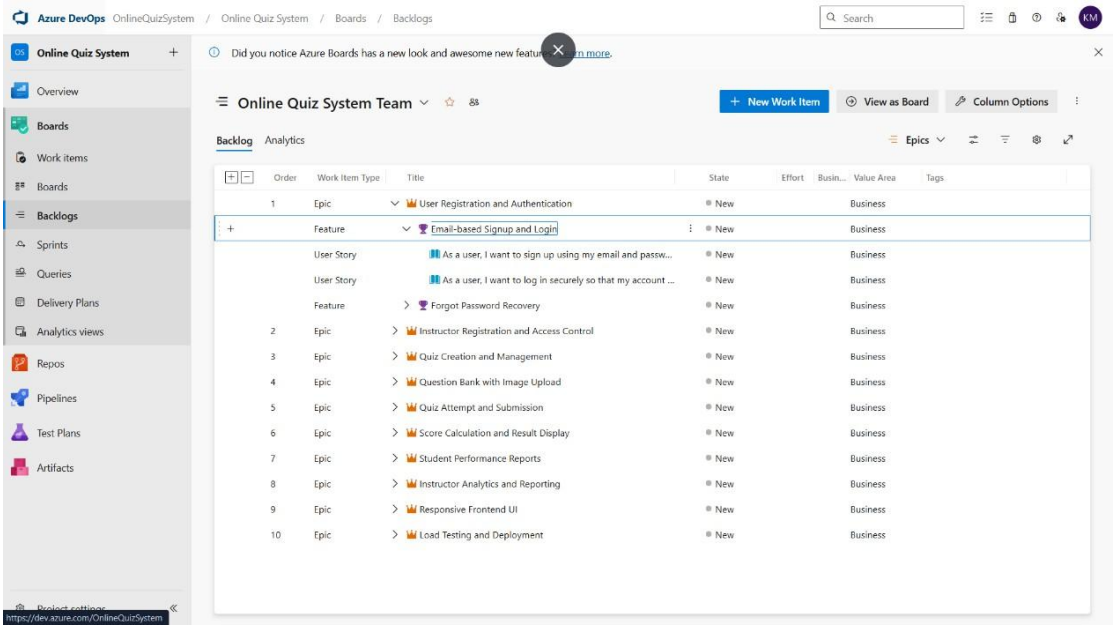
Successfully created an Azure DevOps project with user story management and agile workflow setup.

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

Aim:

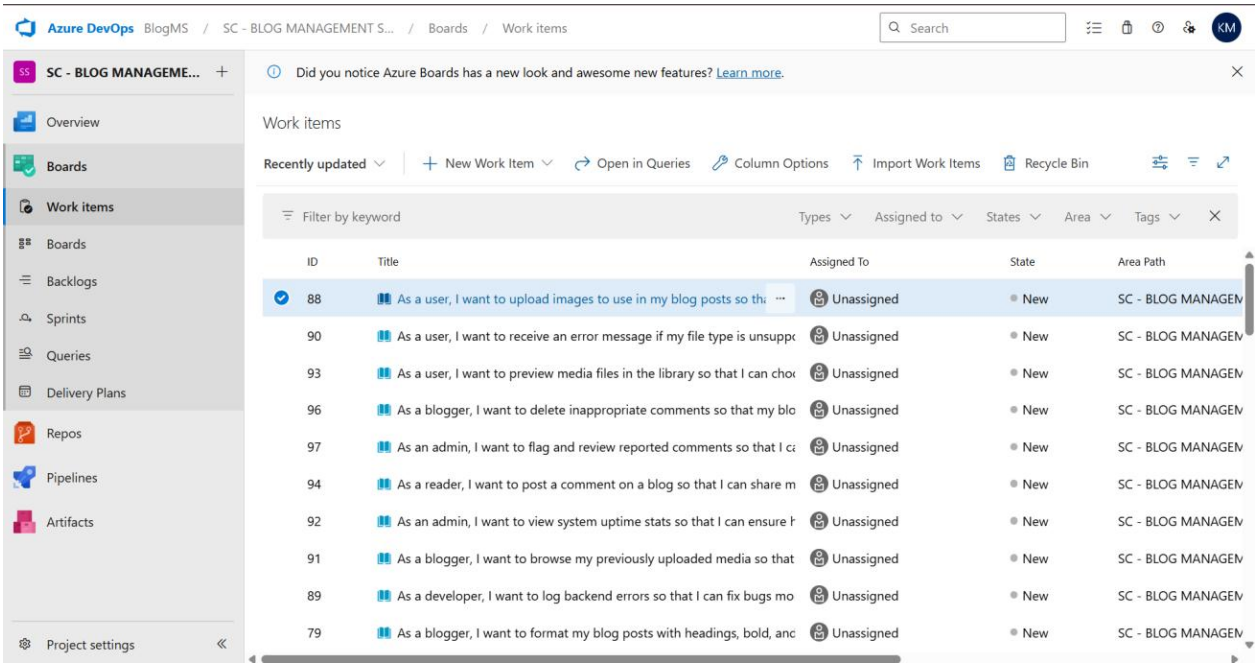
To learn about how to create epics, user story, features, backlogs for your assigned project.

Create Epic, Features, User Stories, Task

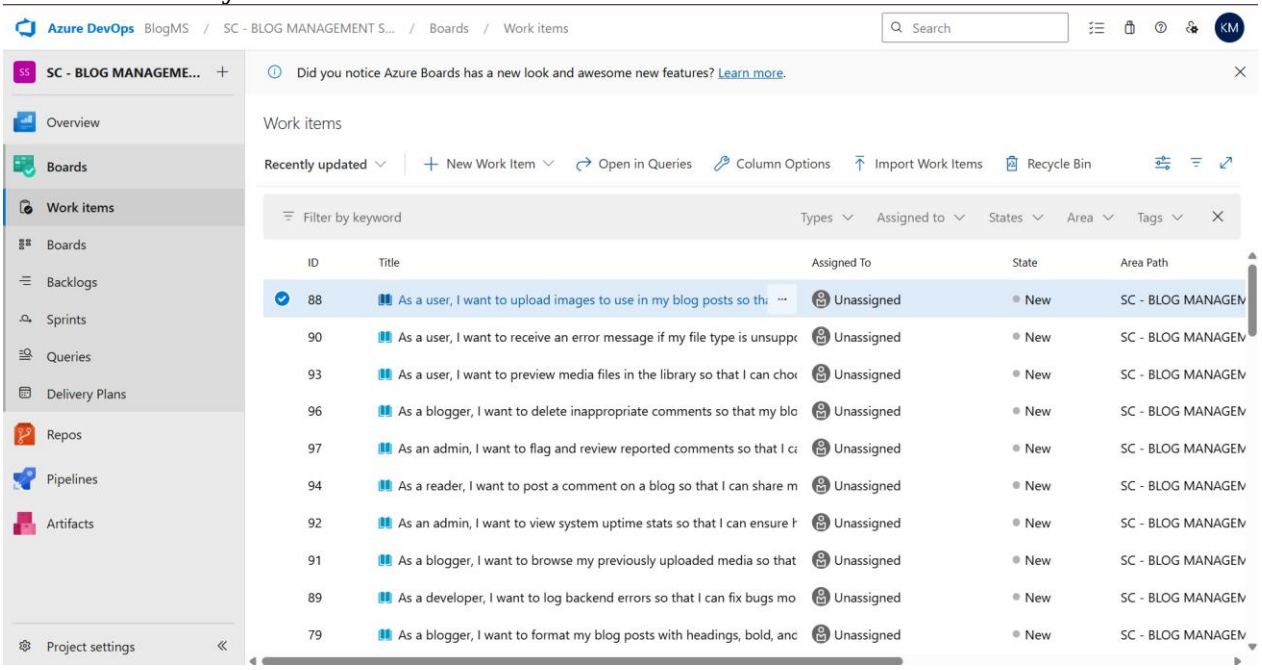


1. Fill in Epics

2. Fill in Features



3. Fill in User Story Details



Result:

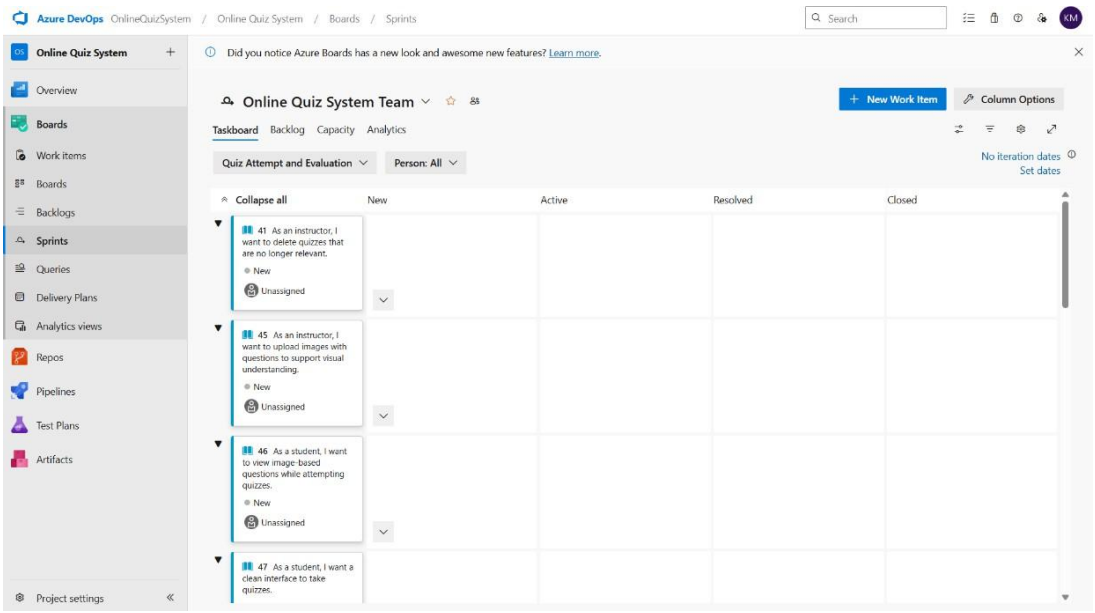
Thus, the creation of epics, features, user story and task has been created successfully.

EXP NO: 4	SPRINT PLANNING

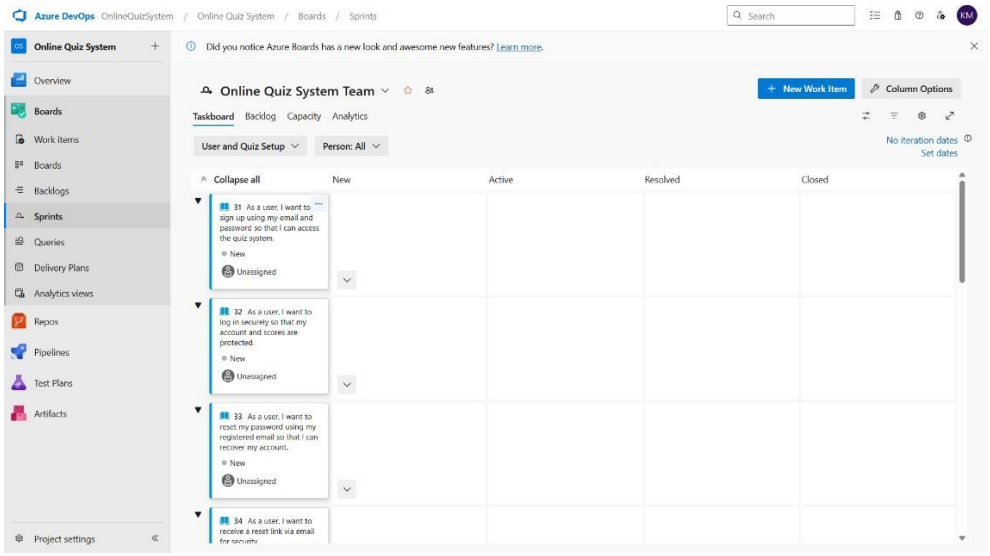
Aim:

To assign user story to specific sprint for the Project Online Quiz System.

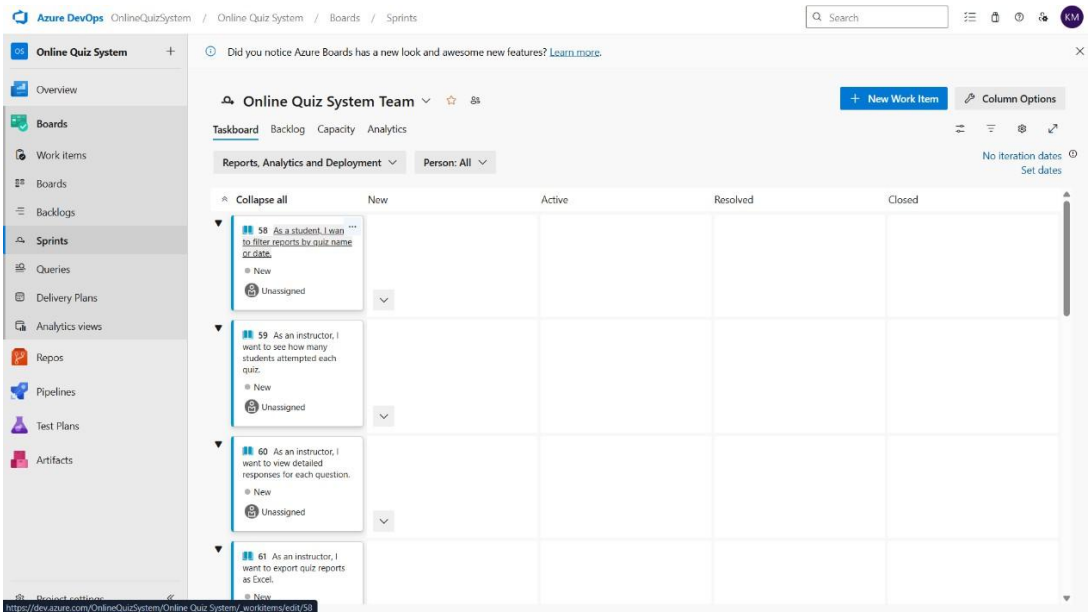
Sprint Planning
Sprint 1



Sprint 2



Sprint 3


Azure DevOps
OnlineQuizSystem / Online Quiz System / Boards / Sprints
Search
Online Quiz System
Did you notice Azure Boards has a new look and awesome new features? Learn more.
Overview
Boards
Work items
Boards
Backlogs
Sprints
Queries
Delivery Plans
Analytics views
Repos
Pipelines
Test Plans
Artifacts
Online Quiz System Team
New Work Item
Column Options
Taskboard Backlog Capacity Analytics
Reports, Analytics and Deployment
Person: All
No iteration dates
Set dates
Collapse all
New
Active
Resolved
Closed
58 As a student, I wan to filter reports by quiz name or date.
New
Unassigned
59 As an instructor, I want to see how many students attempted each quiz.
New
Unassigned
60 As an instructor, I want to view detailed responses for each question.
New
Unassigned
61 As an instructor, I want to export quiz reports as Excel.
New
https://dev.azure.com/OnlineQuizSystem/Online Quiz System/_workitems/edit/58

Result:

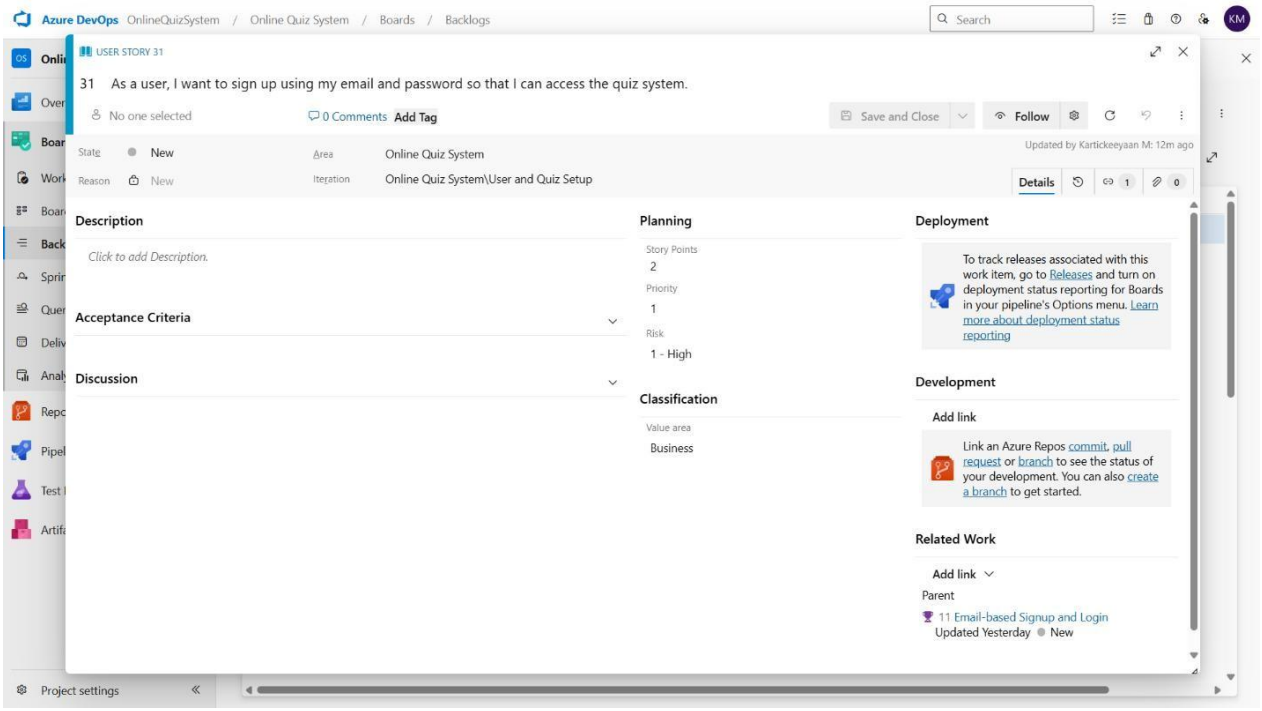
The Sprints are created for the Project Online Quiz System.

EXP NO: 5	POKER ESTIMATION

Aim:

Create Poker Estimation for the user stories – Online Quiz System Project.

Poker Estimation



Result:

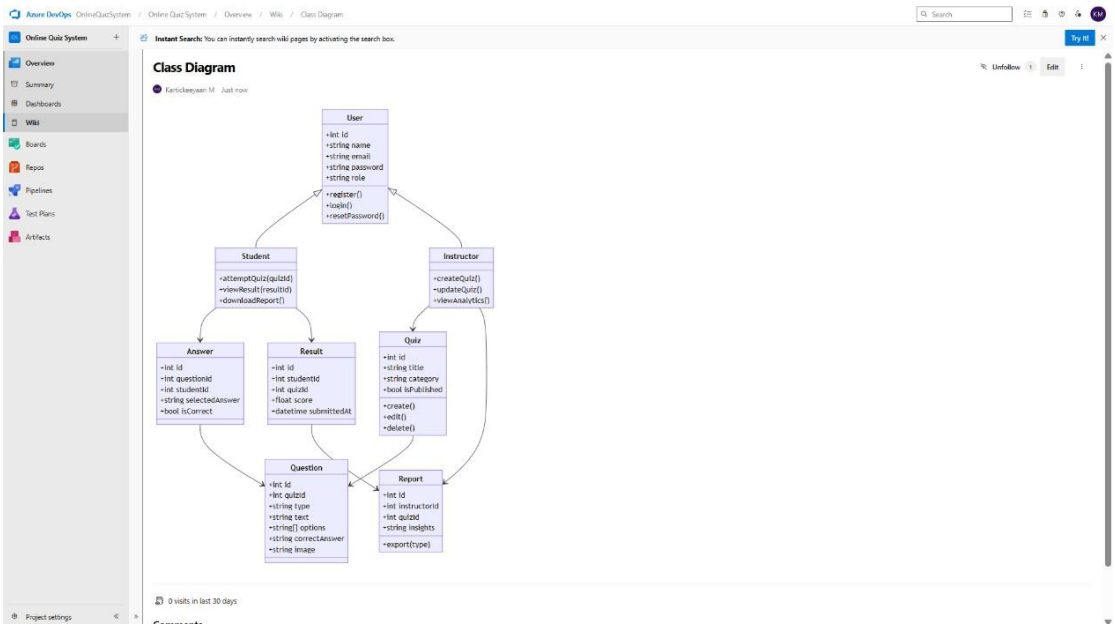
The Estimation/Story Points is created for the project using Poker Estimation.

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

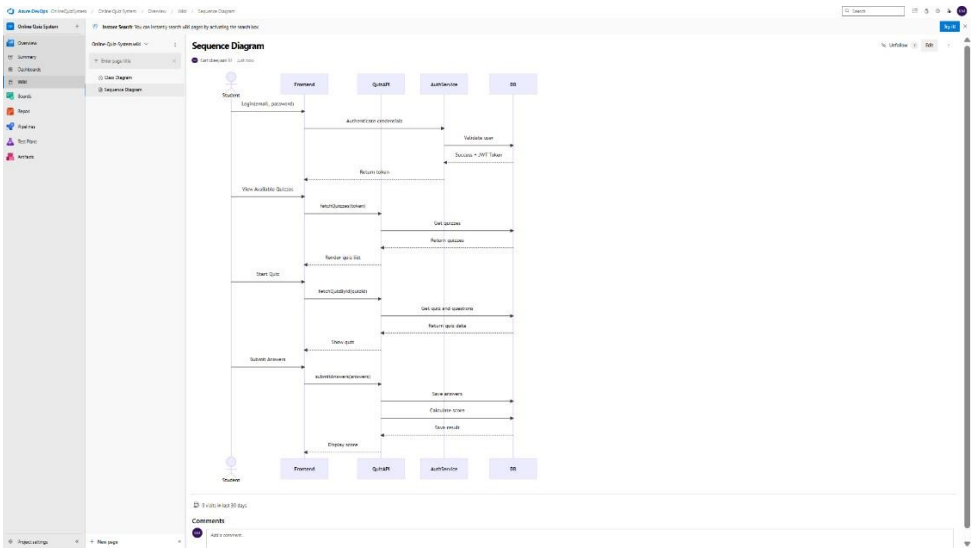
Aim:

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

6A. Class Diagram



6B. Sequence Diagram



Result:

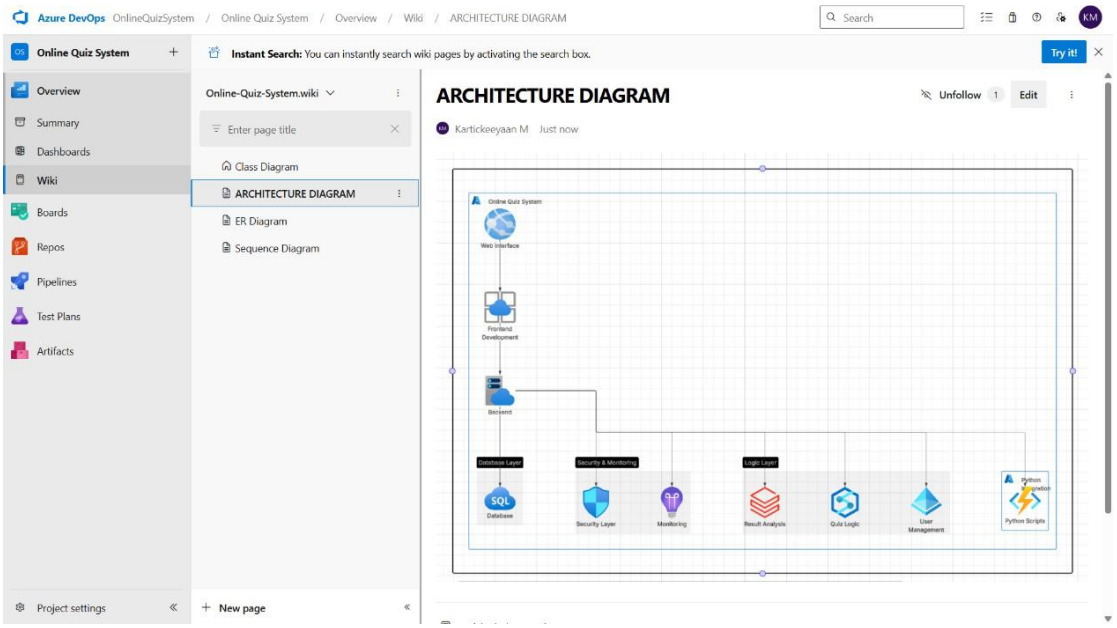
The Class Diagram and Sequence Diagram is designed Successfully for the Project Online Quiz System.

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

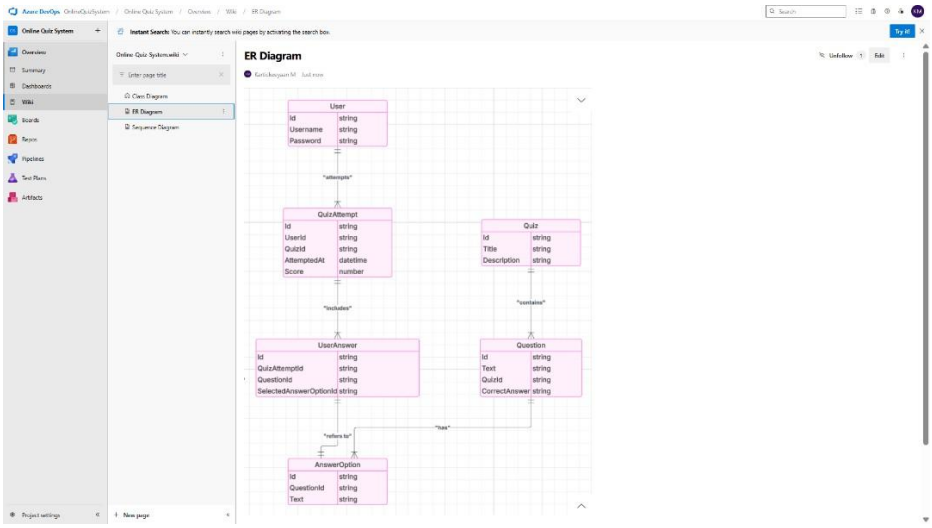
Aim:

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 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

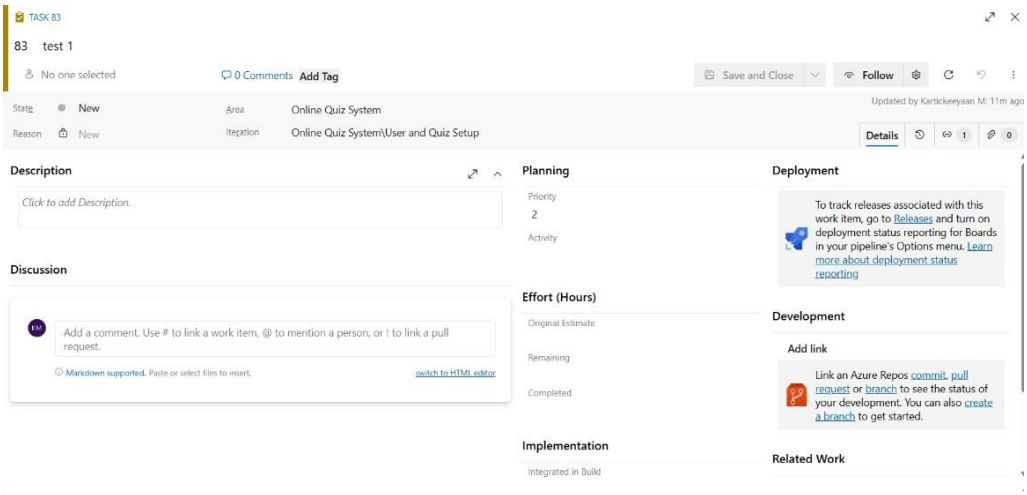
1. **Understand Core Features of the Application**
    - User Signup & Login
    - Viewing and Managing Playlists
    - Fetching Real-time Metadata
    - Editing playlists (rename, reorder, record)
    - Creating smart audio playlists based on categories (mood, genre, artist, etc.)
2. **Define User Interactions**
    - Each test case simulates a real user behaviour (e.g., logging in, renaming a playlist, adding a song).
3. **Design Happy Path Test Cases**
    - Focused on validating that all features function as expected under normal conditions.
    - Example: User logs in successfully, adds item to playlist, or creates a category-based playlist.
4. **Design Error Path Test Cases**
    - Simulate negative or unexpected scenarios to test robustness and error handling.
    - Example: Login fails with invalid credentials, save fails when offline, no recommendations found.
5. **Break Down Steps and Expected Results**
    - Each test case contains step-by-step actions and a corresponding expected outcome.
    - Ensures clarity for both testers and automation scripts.
6. **Use Clear Naming and IDs**
    - Test cases are named clearly (e.g., TC01 – Successful Login, TC10 – Save Playlist Fails).
    - Helps in quick identification and linking to user stories or features.
7. **Separate Test Suites**
    - Grouped test cases based on functionality (e.g., Login, Playlist Editing, Recommendation System).

- Improves organization and test execution flow in Azure DevOps.

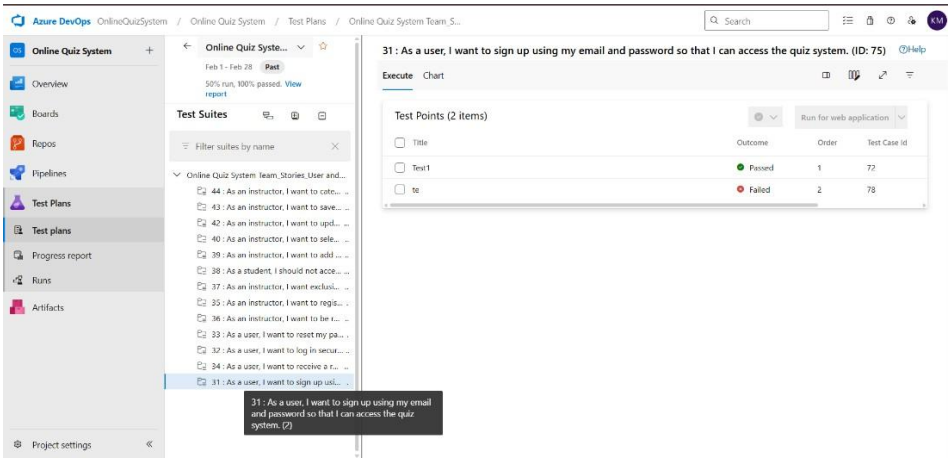
8. Prioritize and Review

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

1. New test plan



2. Test suite



User Story 1: Secure Sign-Up and Login (ID: 79)
Test Plan: Test 1 – User Login

- **TC01 – Successful Sign Up**
  Type: Happy Path
  Action:
    - **Navigate to the Sign-Up page**
    - **Enter valid name, email, and password**
    - **Click "Sign Up"**
      Expected Result:
    - **Fields accept values**
    - **Account is created**
    - **User is redirected to the dashboard**
- **TC02 – Secure Login**
  Type: Happy Path
  Action:
    - **Go to Login page**
    - **Enter valid credentials**
    - **Click "Login"**
      Expected Result:
    - **Login is successful**
    - **User lands on the dashboard**
- **TC03 – Sign Up with Existing Email**
  Type: Error Path
  Action:
    - **Use an already registered email**
    - **Click "Sign Up"**
      Expected Result:
    - **Error: "Email already registered"**
- **TC04 – Login with Wrong Password**
  Type: Error Path
  Action:
    - **Enter valid email and wrong password**
    - **Click "Login"**
      Expected Result:
    - **Error: "Invalid username or password"**

User Story 2: View Quiz List (ID: 76)
Test Plan: Test 2 – View Quizzes

- **TC05 – View Quiz Page**
  Type: Happy Path
  Action:
    - **Login**
    - **Navigate to "Available Quizzes"**
      Expected Result:
    - **All available quizzes are displayed clearly**
- **TC06 – Quiz Loading Failure**
  Type: Error Path
  Action:
    - **Disconnect from the internet**
    - **Try accessing "Available Quizzes"**
      Expected Result:
    - **Error: "Unable to load quizzes"**

User Story 3: Real-Time Score Display (ID: 65)
Test Plan: Test 2 – Real-Time Results

- **TC07 – Score Display After Submission**
  Type: Happy Path
  Action:
    - **Complete a quiz**
    - **Submit answers**
    - **Observe the result panel**
      Expected Result:
    - **Score is calculated and displayed instantly**
- **TC08 – Score Not Displaying**
  Type: Error Path
  Action:
    - **Submit quiz with a backend issue**
      Expected Result:
    - **Error: "Unable to fetch results, please try again"**

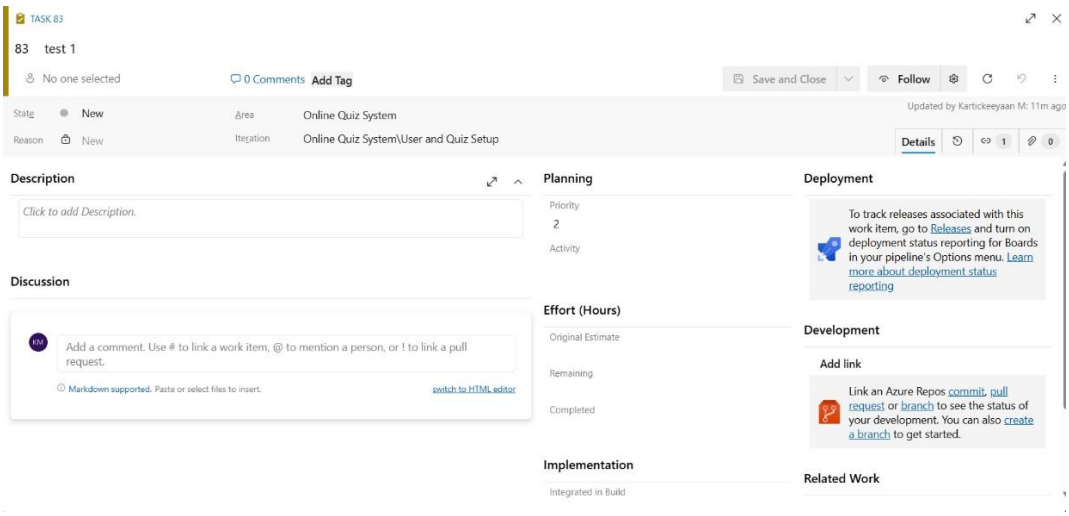
User Story 4: Edit Quiz Details (ID: 68)
Test Plan: Test 3 – Quiz Editing

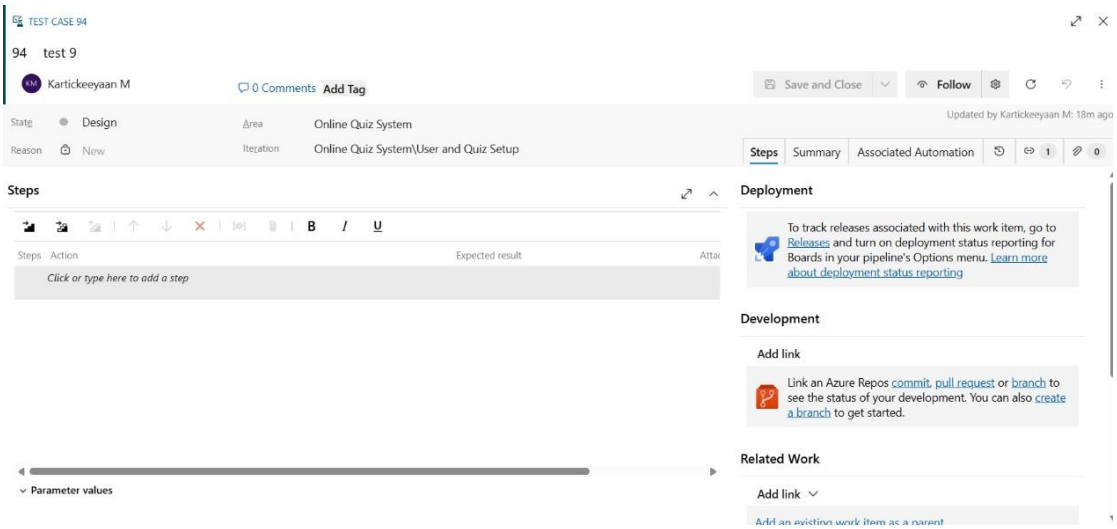
- **TC09 – Rename Quiz Successfully**
  Type: Happy Path
  Action:
    - **Admin navigates to quiz list**
    - **Clicks "Edit" on a quiz**
    - **Changes quiz name and saves**
      Expected Result:
    - **Quiz name updates successfully**
- **TC10 – Rename with Blank Name**
  Type: Error Path
  Action:
    - **Click "Edit"**
    - **Leave the quiz name blank**
    - **Click "Save"**
      Expected Result:
    - **Error: "Quiz name cannot be empty"**
- **TC11 – Change Quiz Question Order**
  Type: Happy Path
  Action:
    - **Open quiz editor**
    - **Drag and drop questions to reorder**
    - **Click "Save"**
      Expected Result:
    - **Question order is saved**
- **TC12 – Reordering Fails Due to Error**
  Type: Error Path
  Action:
    - **Try reordering during server error**
    - **Click "Save"**
      Expected Result:
    - **Error: "Failed to update. Try again later"**

User Story 5: Generate Quiz Based on Categories (ID: 73)
Test Plan: Test 3 – Smart Quiz Generation

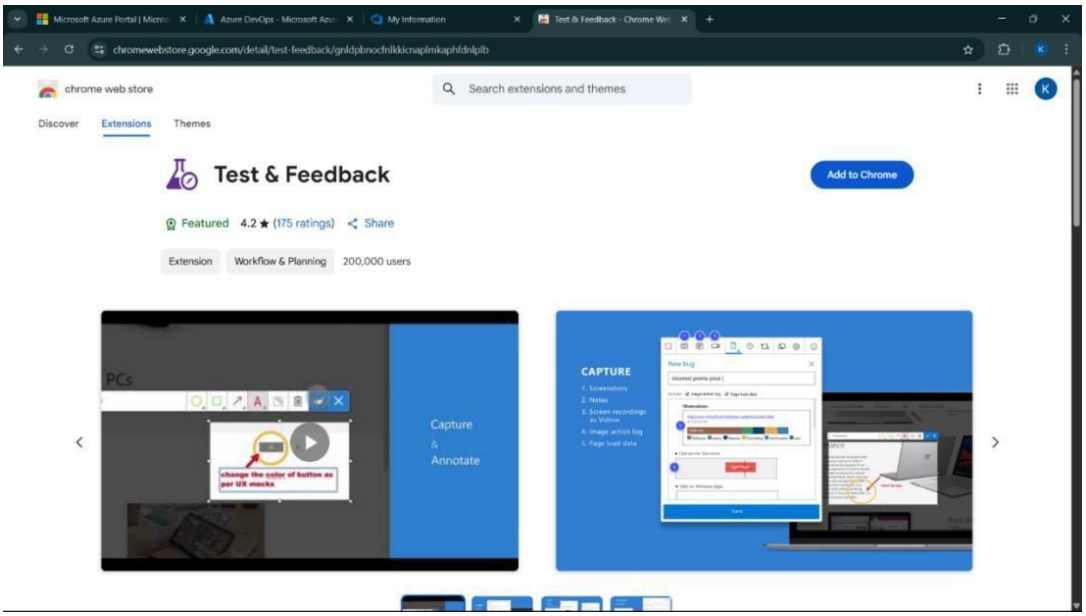
- **TC13 – Generate Quiz Based on Subject**
  Type: Happy Path
  Action:
    - **Select subject and difficulty**
    - **Click "Generate Quiz"**
      Expected Result:
    - **New quiz is created based on selected filters**
- **TC14 – Fail to Generate Quiz (Missing/Invalid Input)**
  Type: Error Path
  Action:
    - **Click "Generate" without selecting subject**
      Expected Result:
    - **Error: “Please select a valid subject/category”**

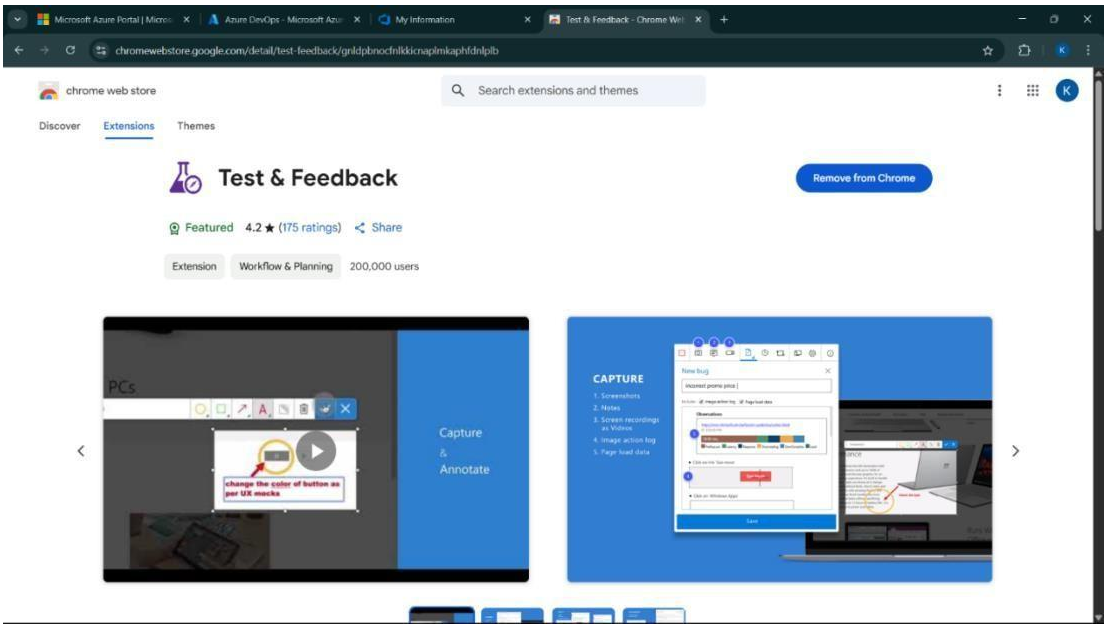
Test Cases



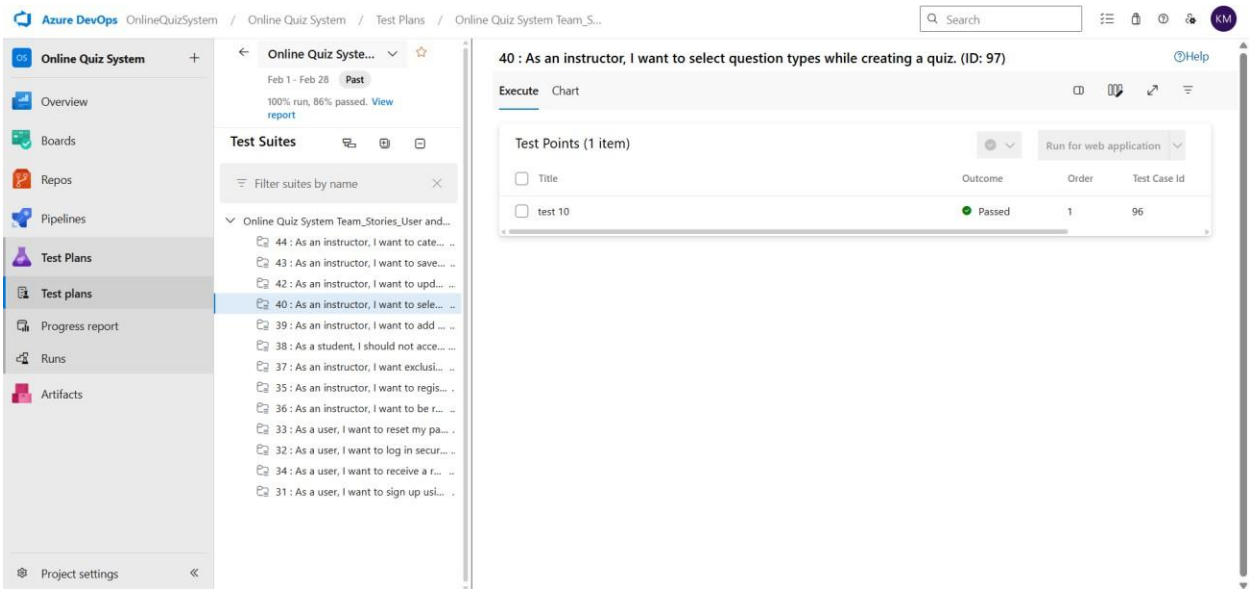


3. Installation of test

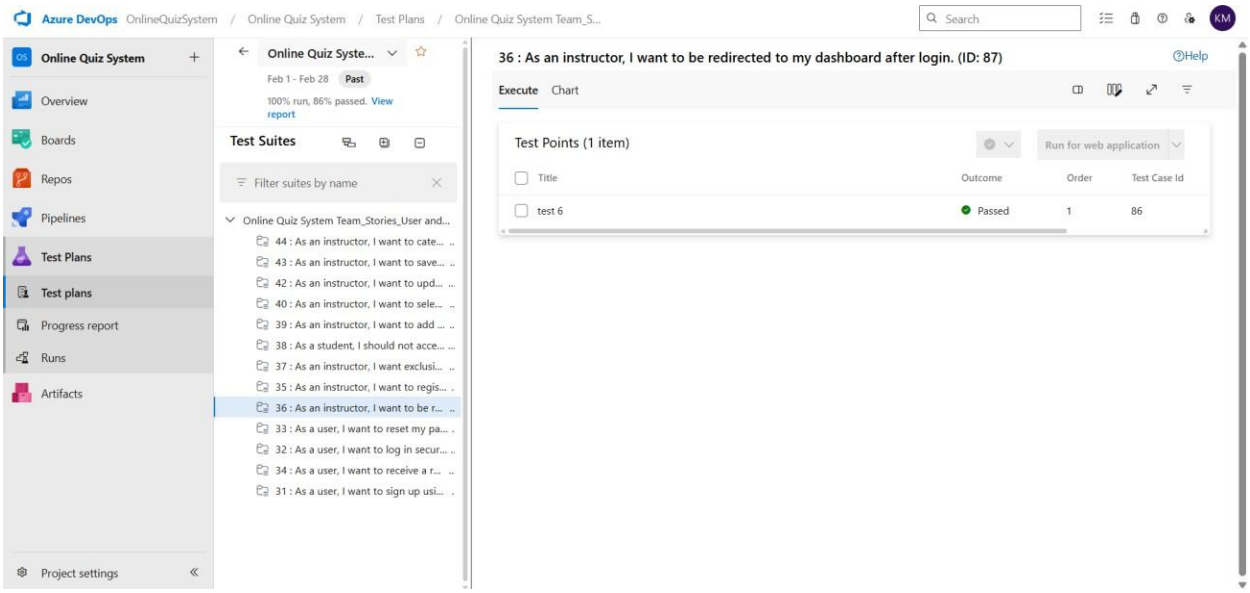


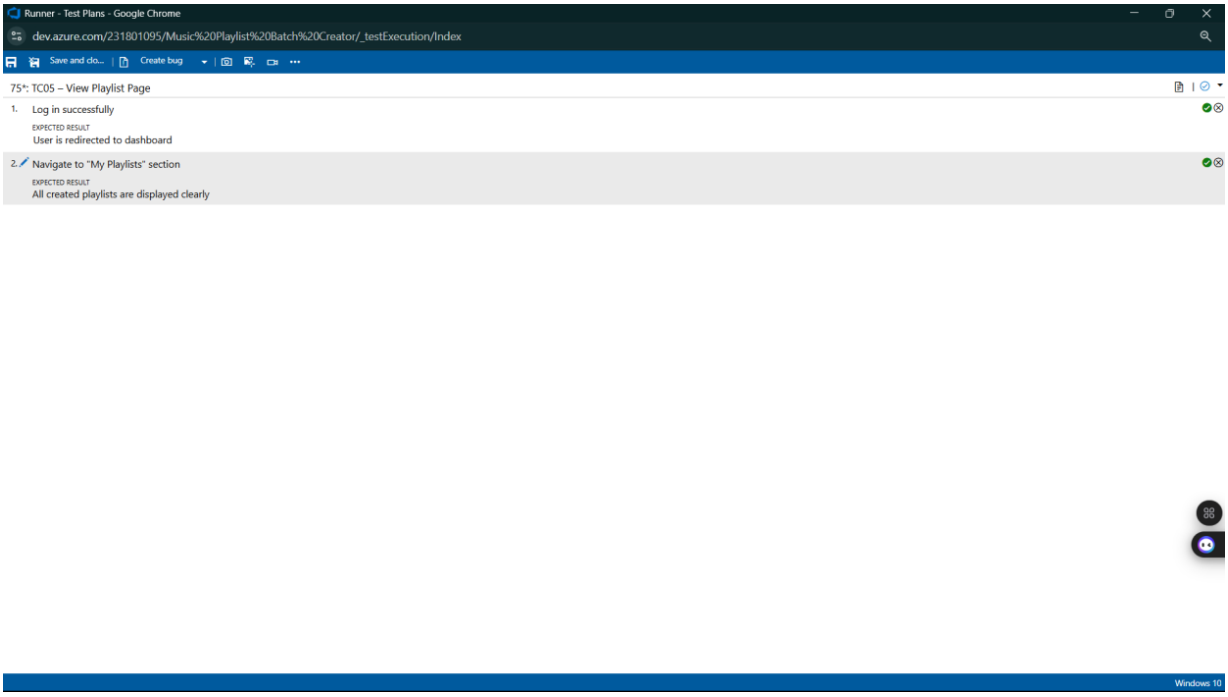


Test and feedback
Showing it as an extension

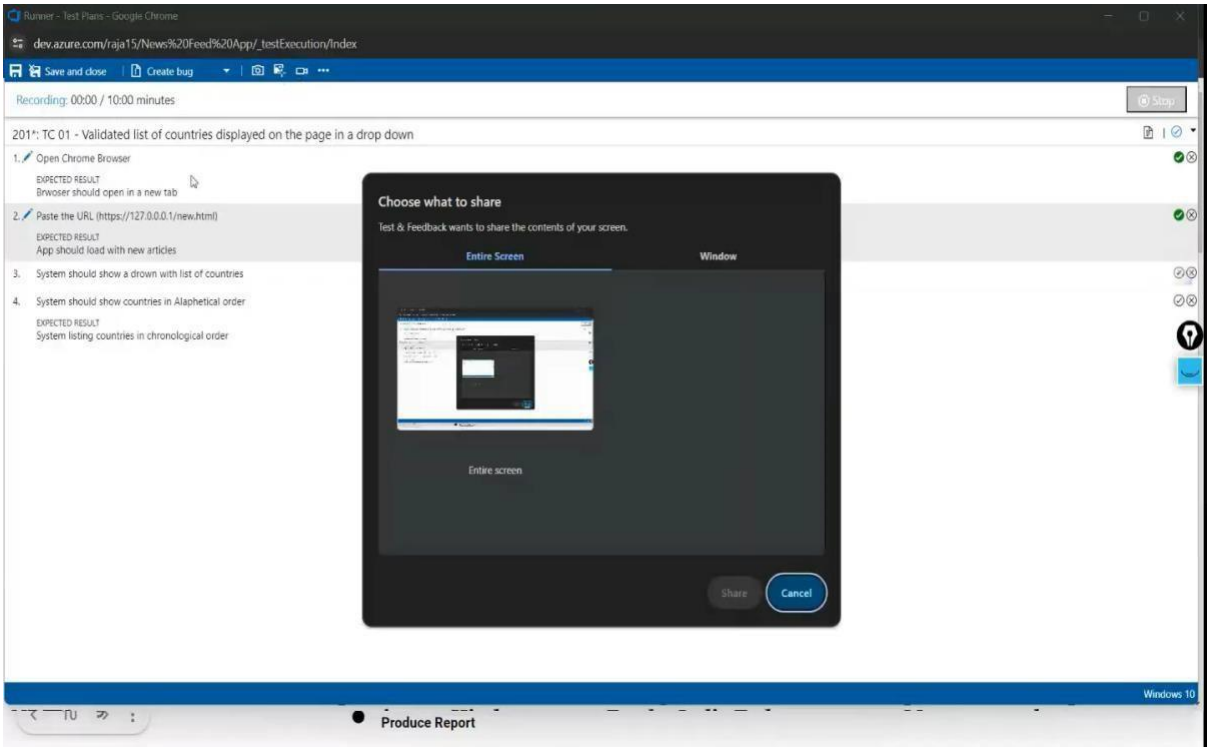


4. Running the test cases

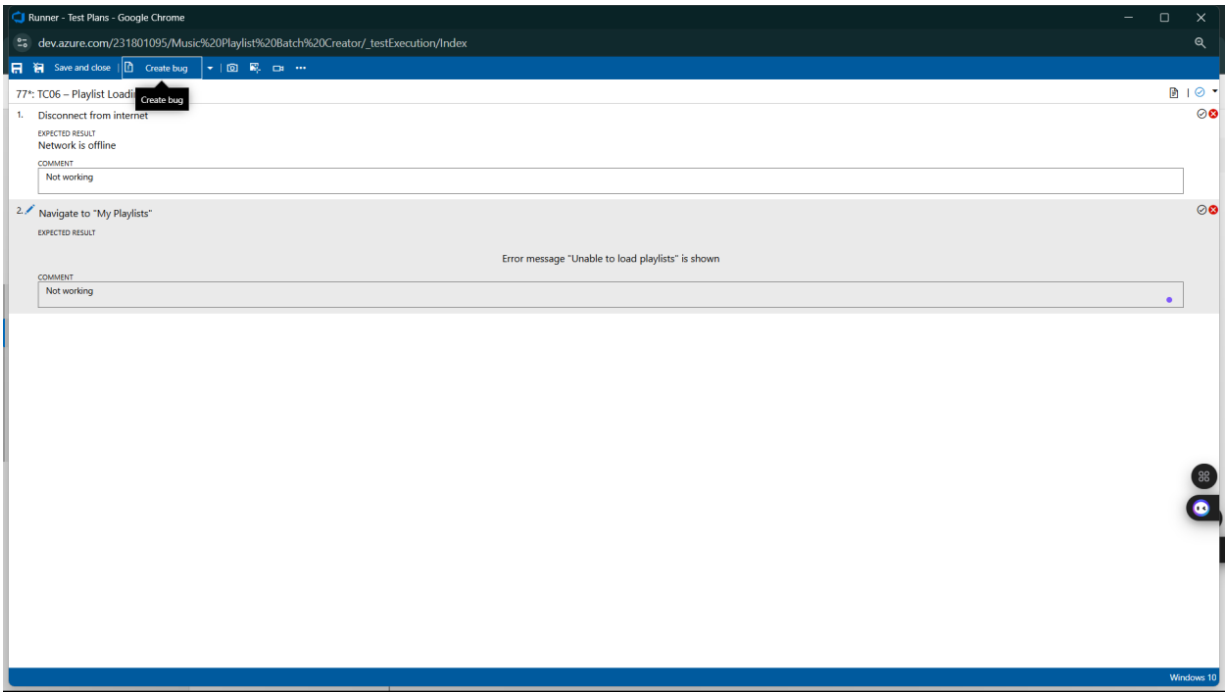




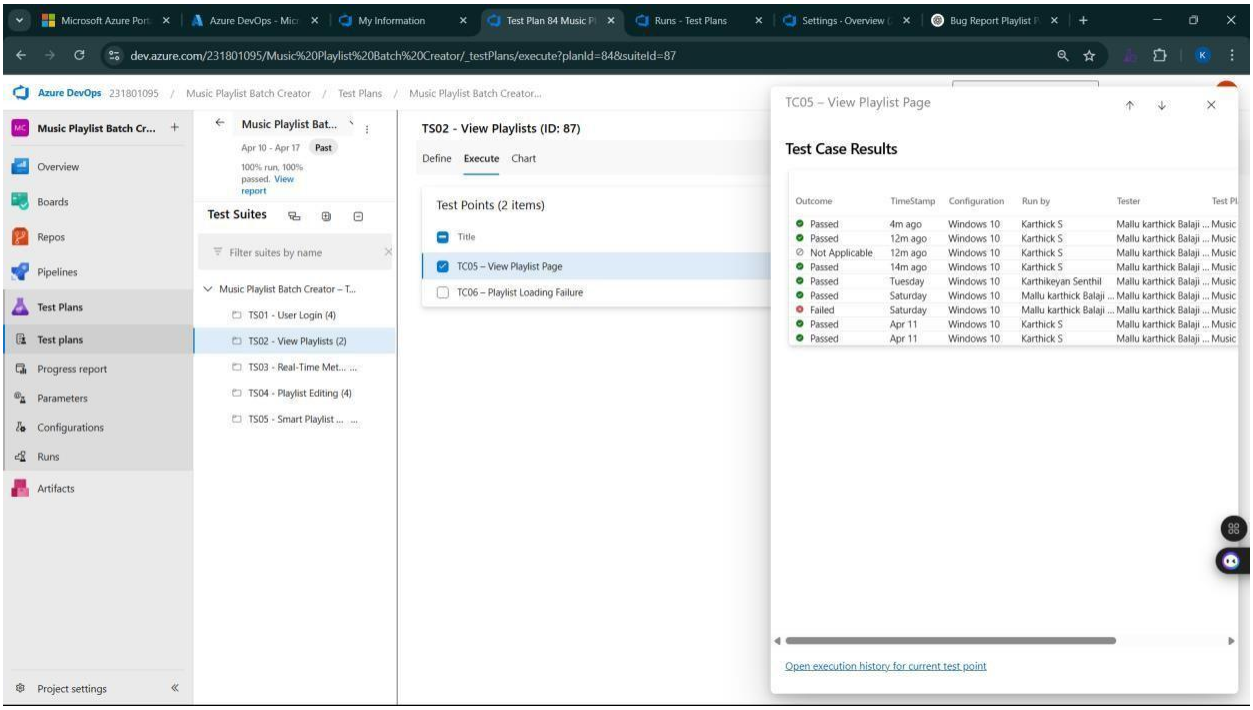
5. Recording the test case



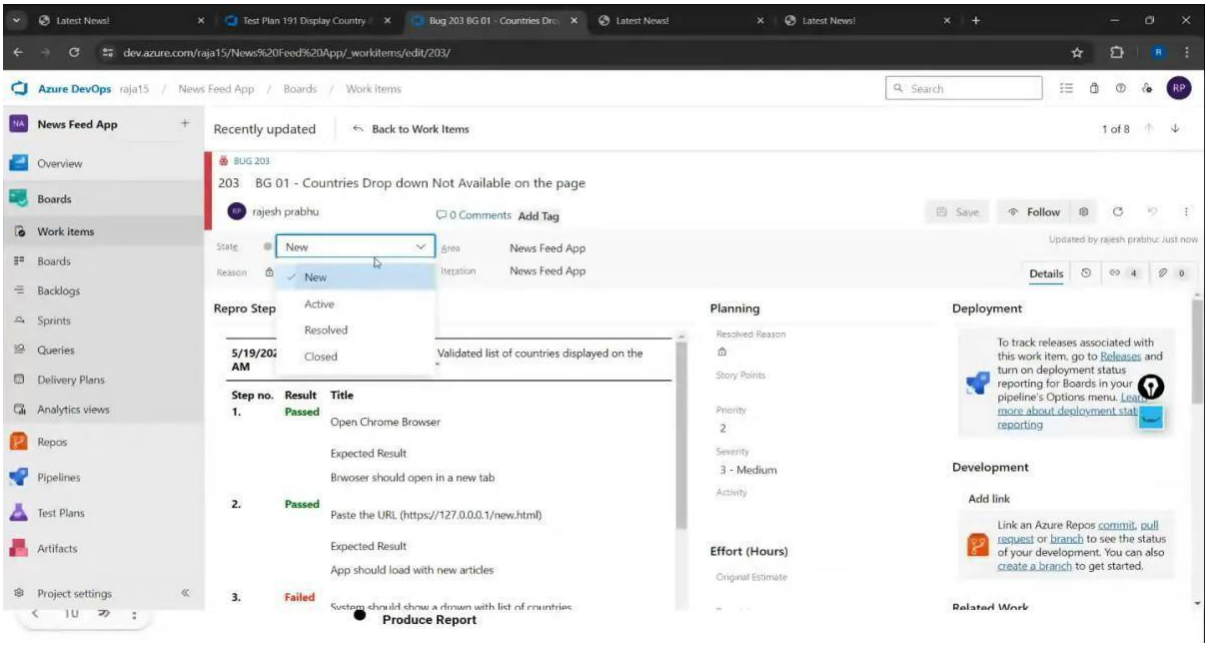
6. Creating the bug



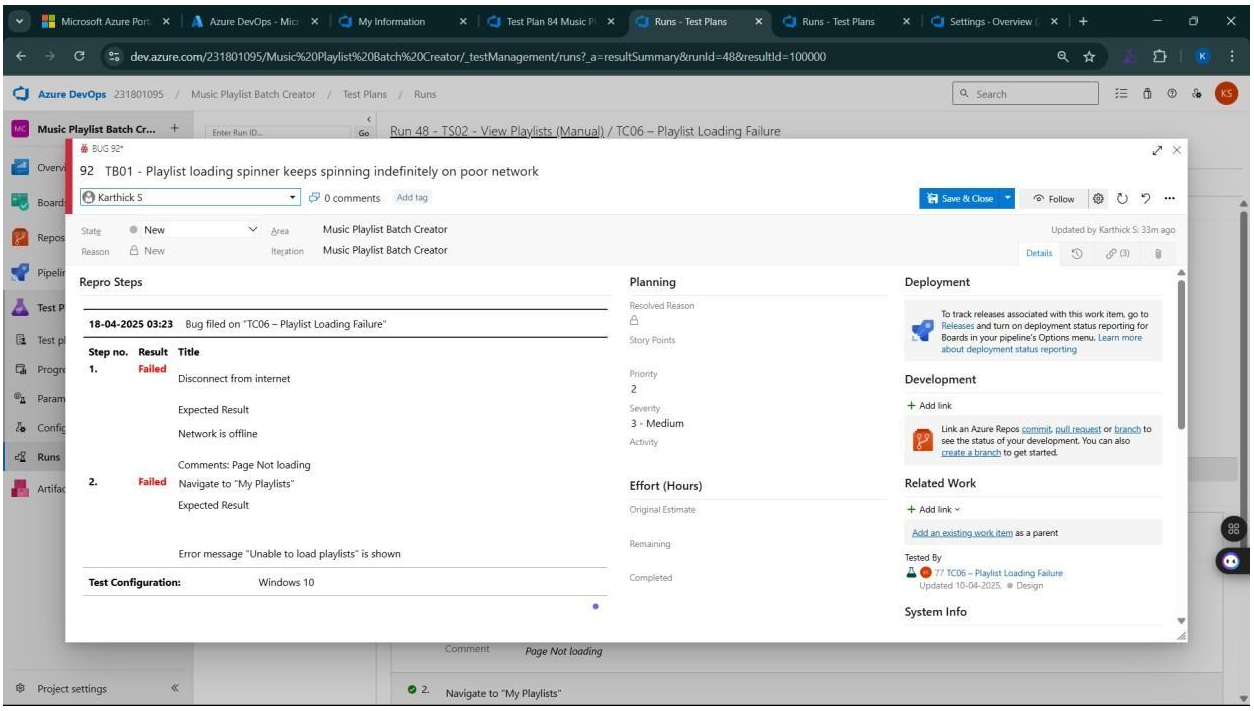
7. Test case results



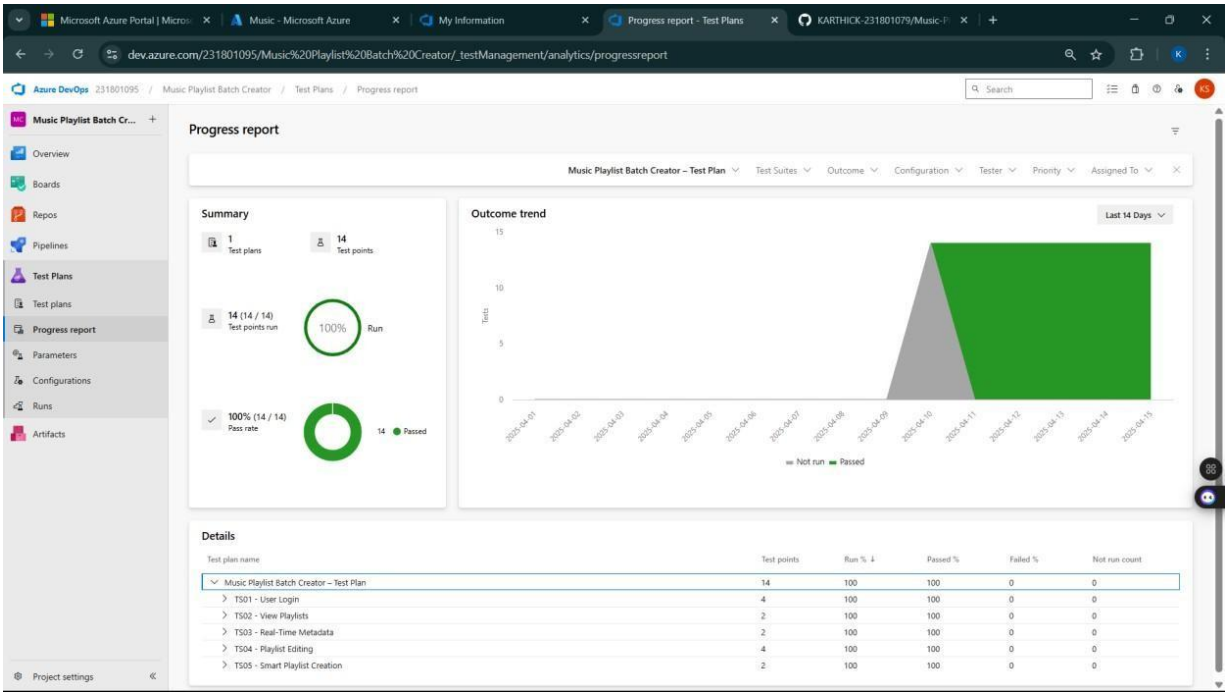
8. Test report summary

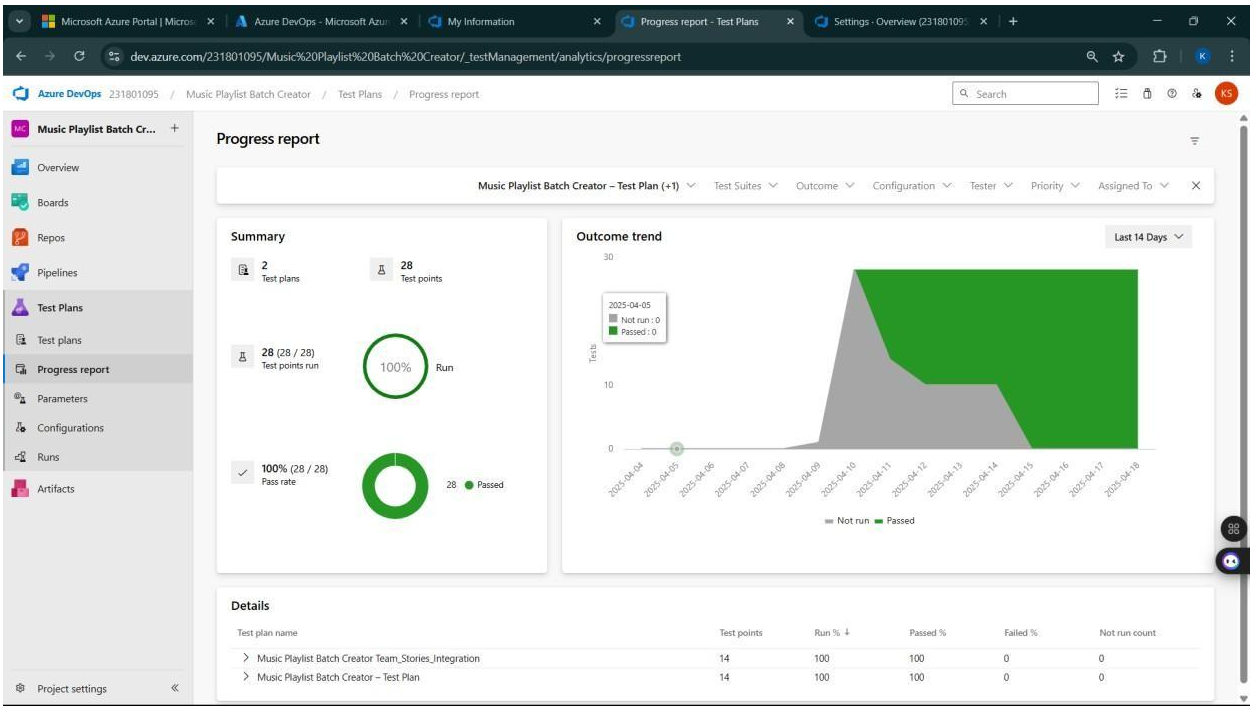


- Assigning bug to the developer and changing state

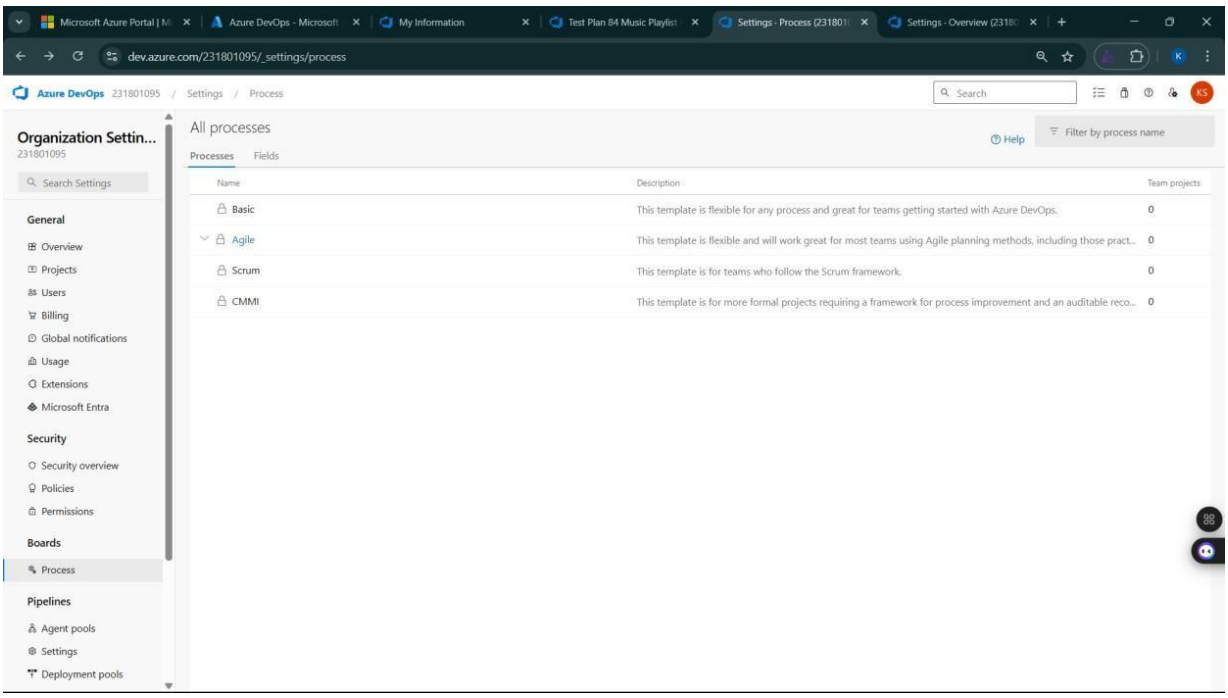


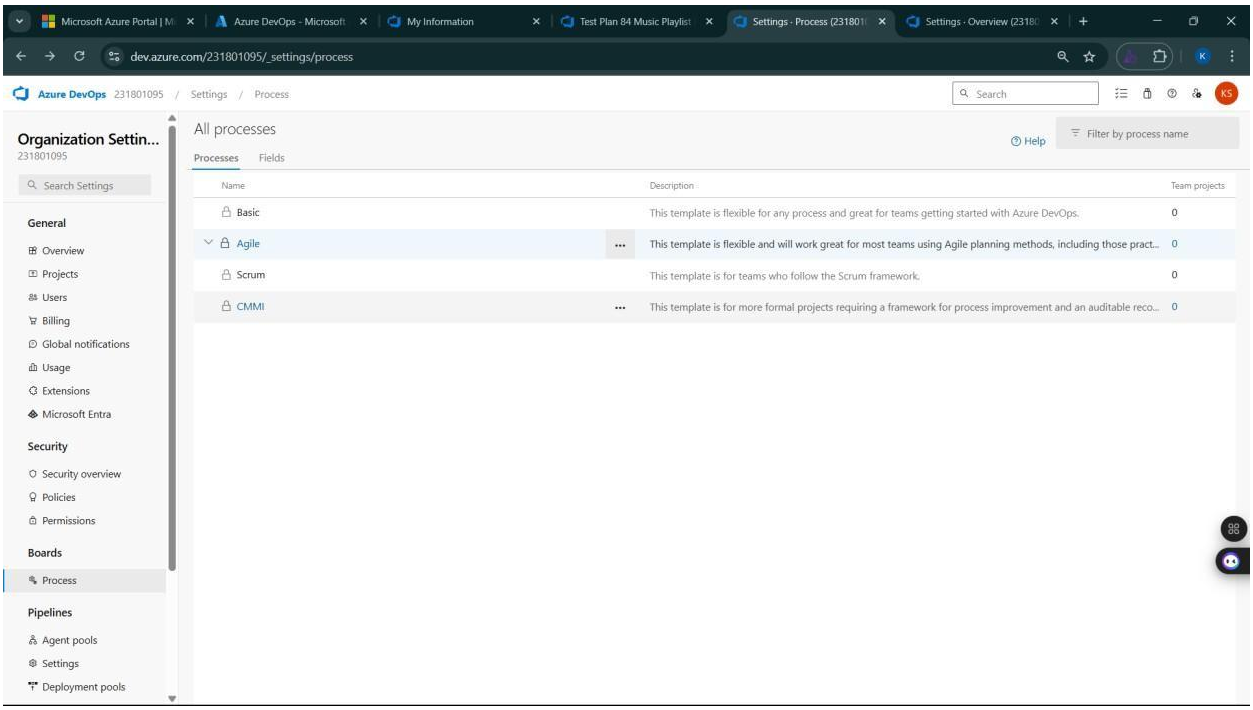
9. Progress report

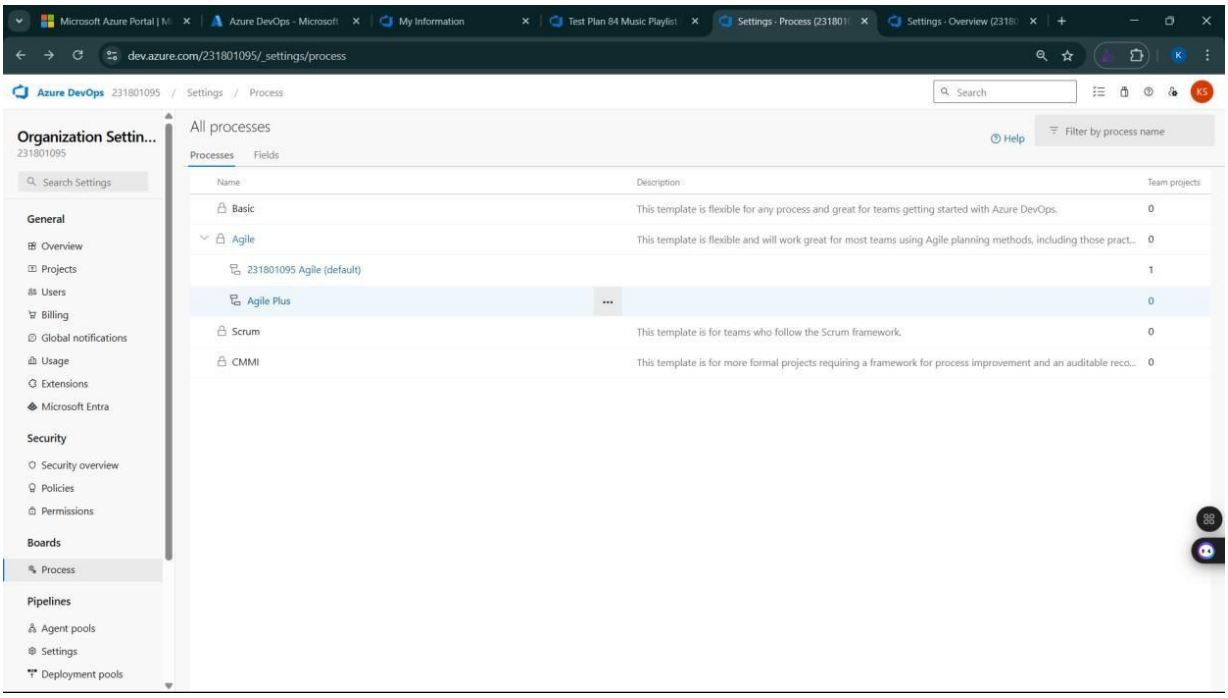




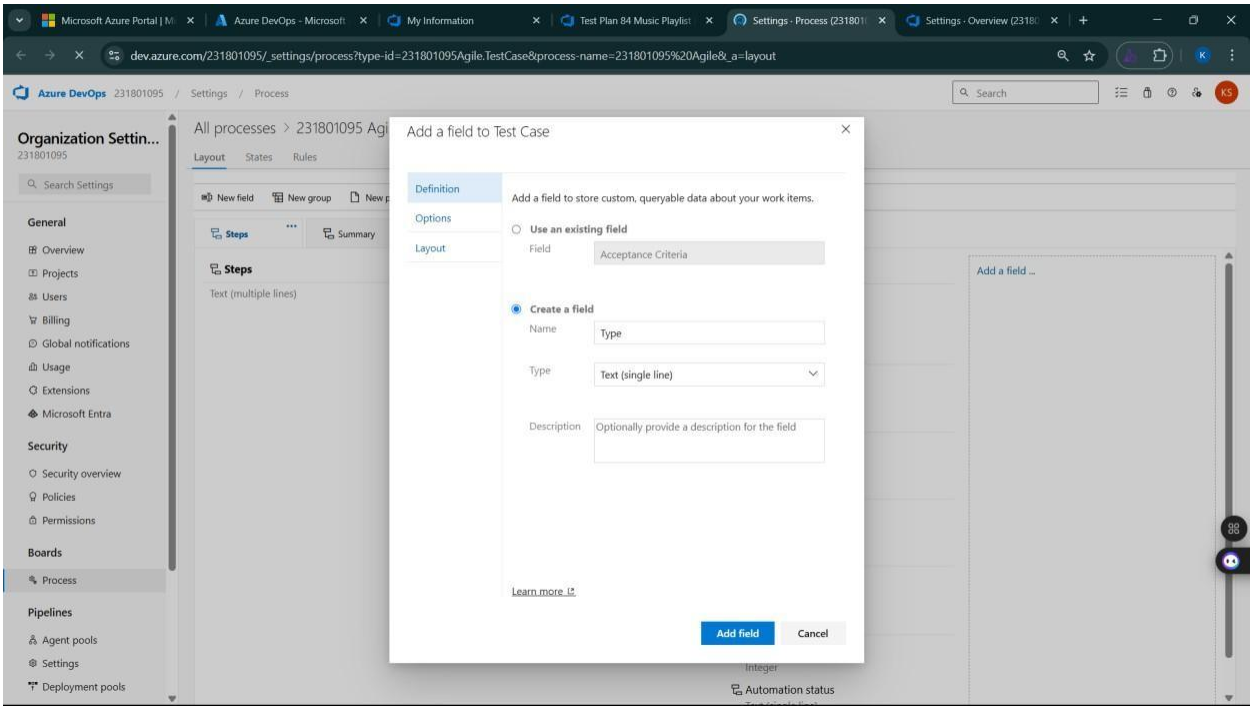
10. Changing the test template

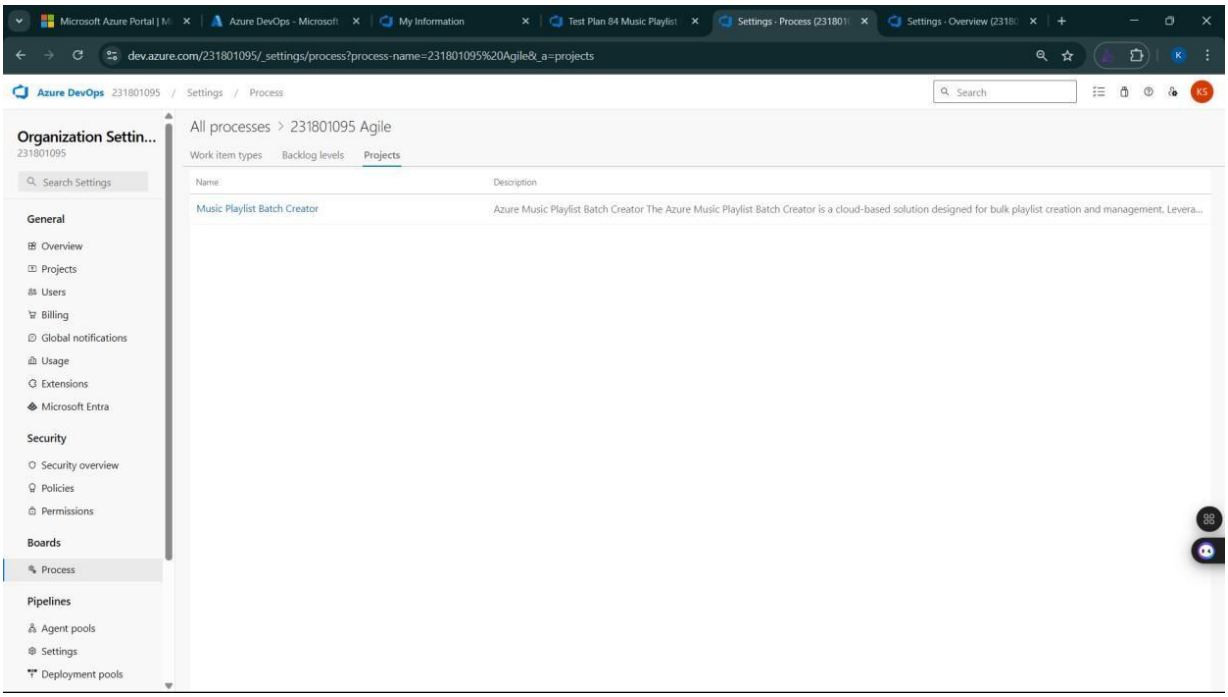


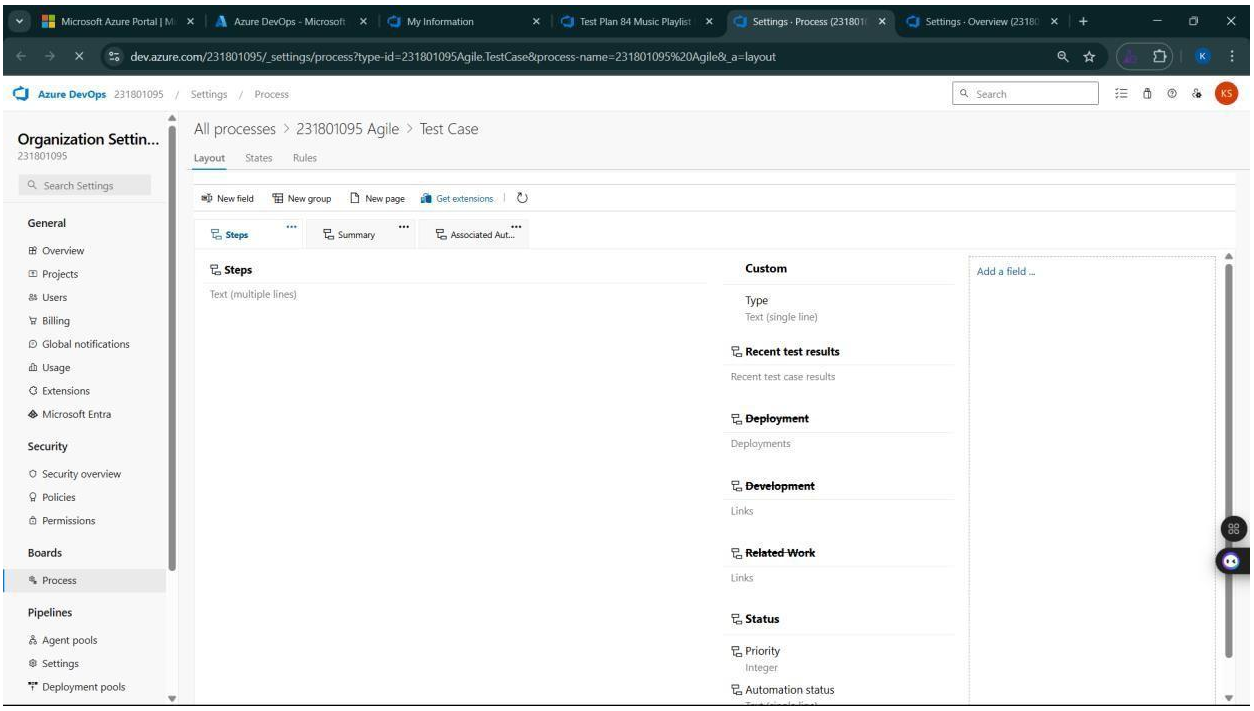

Microsoft Azure Portal | M
Azure DevOps - Microsoft
My Information
Test Plan 84 Music Playlist
Settings · Process (231801
Settings · Overview (2318
dev.azure.com/231801095/_settings/process
Azure DevOps 231801095 / Settings / Process
Search
Organization Settin...
231801095
Search Settings
General
Overview
Projects
Users
Billing
Global notifications
Usage
Extensions
Microsoft Entra
Security
Security overview
Policies
Permissions
Boards
Process
Pipelines
Agent pools
Settings
Deployment pools
All processes
Help
Filter by process name
Processes
Fields
Name
Description
Team projects
Basic
This template is flexible for any process and great for teams getting started with Azure DevOps.
0
Agile
This template is flexible and will work great for most teams using Agile planning methods, including those pract...
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 improvement and an auditable reco...
0


Microsoft Azure Portal | M
Azure DevOps - Microsoft
My Information
Test Plan 84 Music Playlist
Settings · Process (231801
Settings · Overview (2318
dev.azure.com/231801095/_settings/process
Azure DevOps 231801095 / Settings / Process
Search
Organization Settin...
231801095
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
All processes
Help
Filter by process name
Processes
Fields
Name
Description
Team projects
Basic
This template is flexible for any process and great for teams getting started with Azure DevOps.
0
Agile
This template is flexible and will work great for most teams using Agile planning methods, including those pract...
0
231801095 Agile (default)
1
Agile Plus
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 improvement and an auditable reco...
0

11. 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:

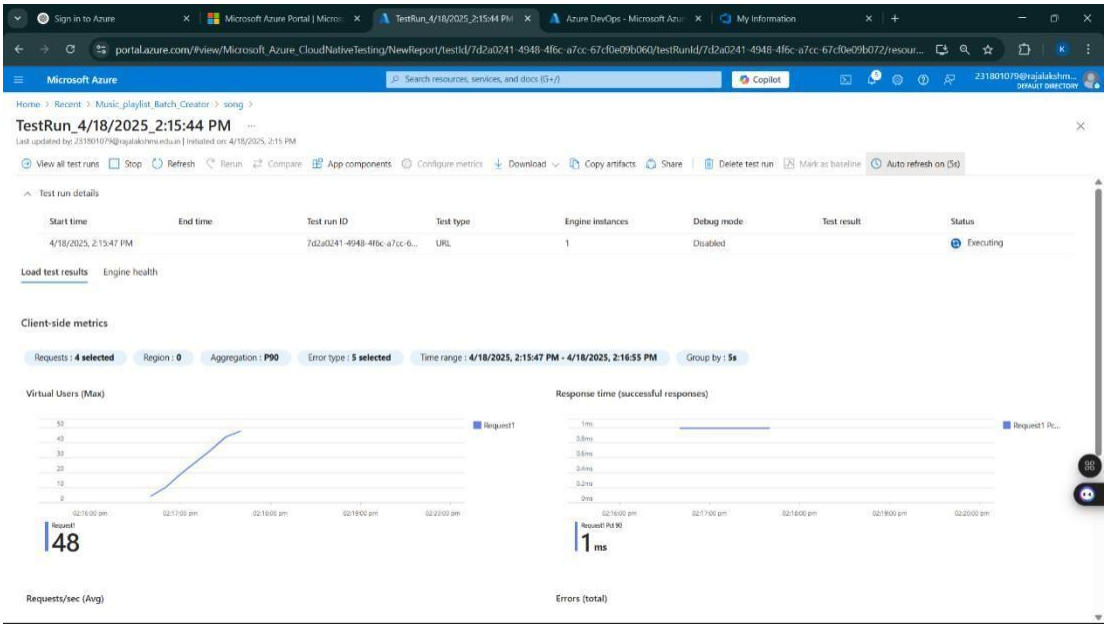
1. Sign in to Azure Portal
   Go to https://portal.azure.com and log in.
2. Create the Resource
   - Go to *Create a resource* → Search for "Azure Load Testing".
   - Select Azure Load Testing and click Create.
3. Fill in the Configuration Details
   - Subscription*:* Choose your Azure subscription.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

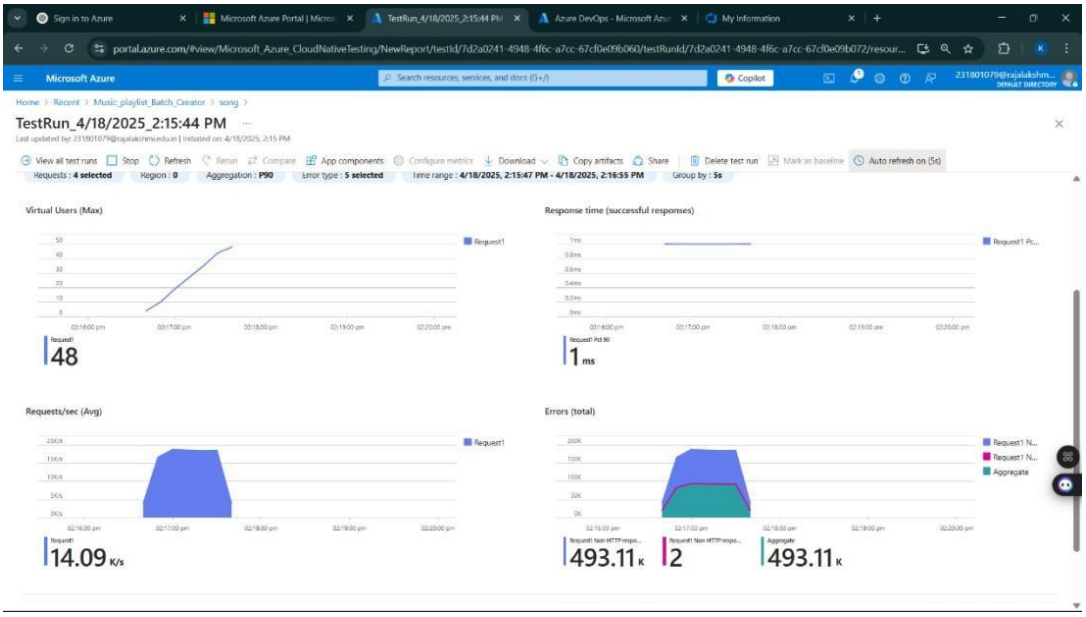
Steps to Create and Run a Load Test:

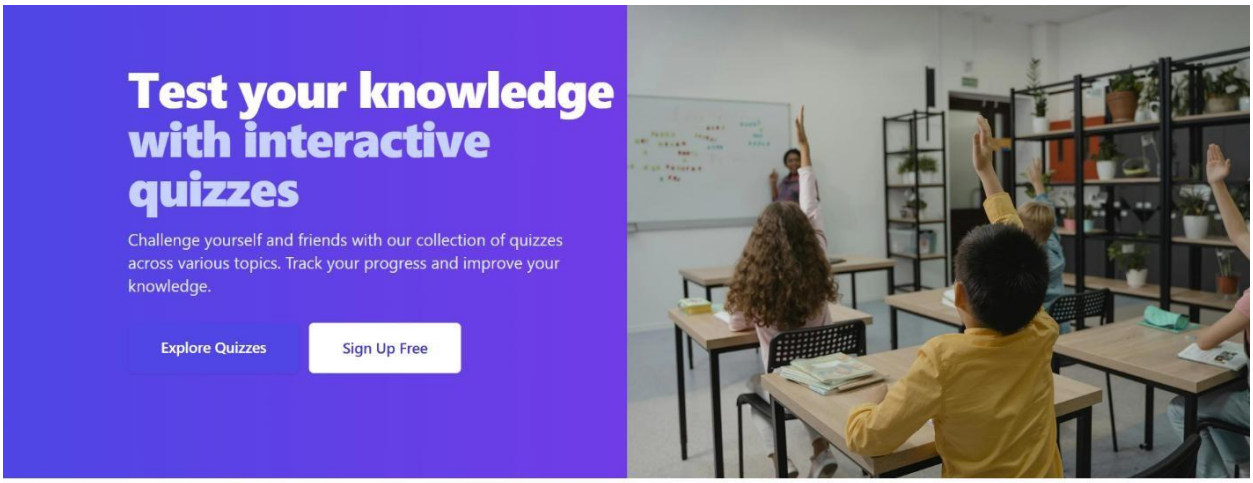
Once your resource is ready:

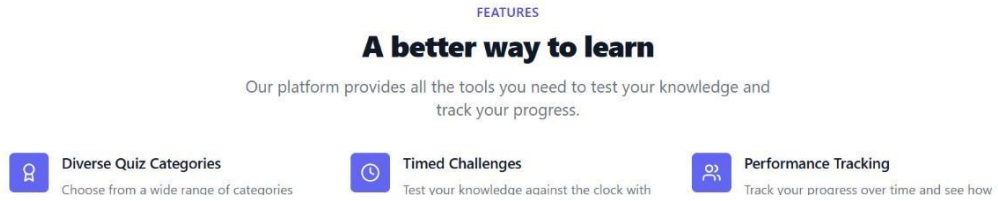
1. Go to your Azure Load Testing resource and click Add HTTP requests > Create.
2. Basics Tab
   - *Test Name:* Provide a unique name.
   - *Description:* (Optional) Add test purpose.
   - *Run After Creation:* Keep checked.
3. Load Settings
   - *Test URL:* Enter the target endpoint (e.g., https://yourapi.com/products).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

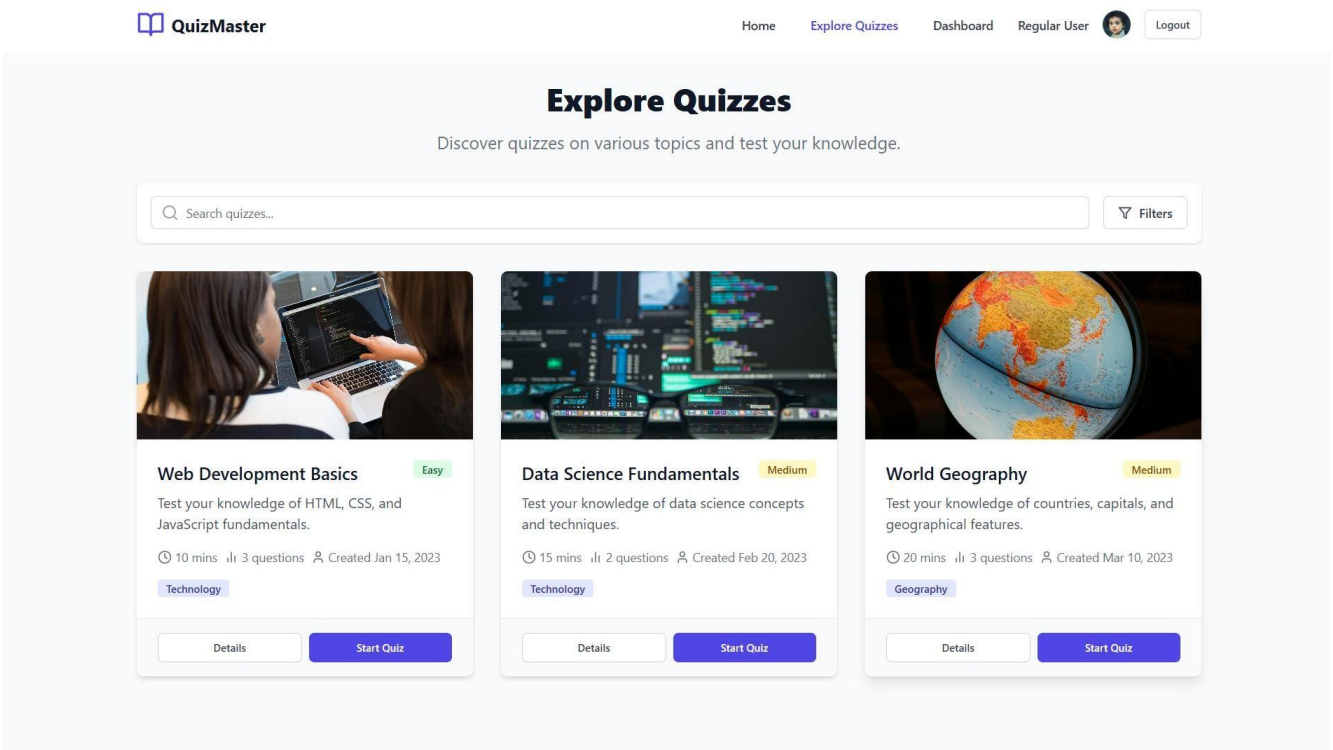
Load Testing

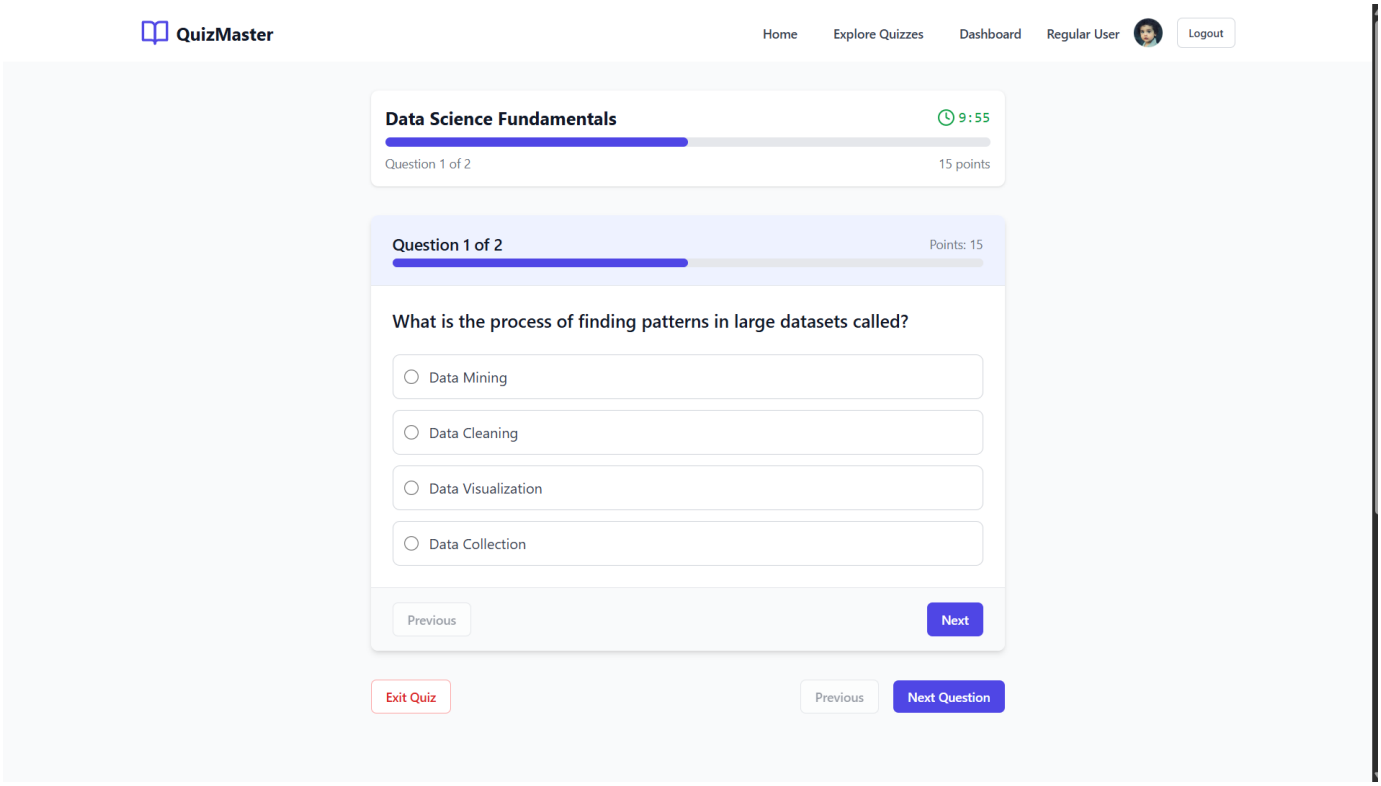





Test your knowledge with interactive quizzes
Challenge yourself and friends with our collection of quizzes across various topics. Track your progress and improve your knowledge.
Explore Quizzes
Sign Up Free


FEATURES
A better way to learn
Our platform provides all the tools you need to test your knowledge and track your progress.
Diverse Quiz Categories
Choose from a wide range of categories
Timed Challenges
Test your knowledge against the clock with
Performance Tracking
Track your progress over time and see how


QuizMaster
Home
Explore Quizzes
Dashboard
Regular User
Logout
Explore Quizzes
Discover quizzes on various topics and test your knowledge.
Search quizzes...
Filters
Web Development Basics
Easy
Test your knowledge of HTML, CSS, and JavaScript fundamentals.
10 mins
3 questions
Created Jan 15, 2023
Technology
Details
Start Quiz
Data Science Fundamentals
Medium
Test your knowledge of data science concepts and techniques.
15 mins
2 questions
Created Feb 20, 2023
Technology
Details
Start Quiz
World Geography
Medium
Test your knowledge of countries, capitals, and geographical features.
20 mins
3 questions
Created Mar 10, 2023
Geography
Details
Start Quiz



Result:

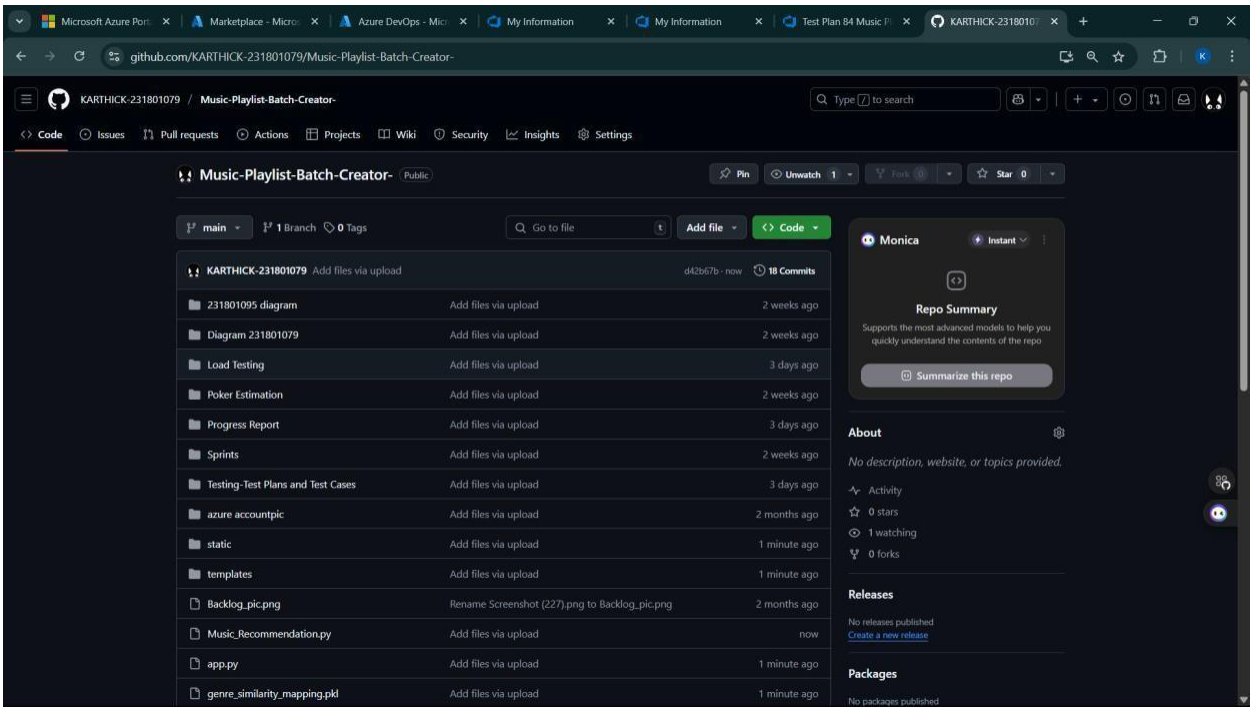
Successfully created the Azure Load Testing resource and executed a load test to assess the performance of the specified endpoint.

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.