

RAJALAKSHMI ENGINEERING COLLEGE
RAJALAKSHMI NAGAR, THANDALAM - 602 105



**RAJALAKSHMI
ENGINEERING COLLEGE**

CS23432

SOFTWARE CONSTRUCTION

Laboratory Record Note Book

Name :

Year / Branch / Section :

Register No. :

Semester :

Academic Year :



RAJALAKSHMI ENGINEERING COLLEGE (AUTONOMOUS)
RAJALAKSHMI NAGAR, THANDALAM – 602 105

BONAFIDE CERTIFICATE

NAME _____ REGISTER NO. _____

ACADEMIC YEAR 2024-25 **SEMESTER- IV** **BRANCH:** B. Tech Information Technology [AD/AE]. This Certification is the Bonafide record of work done by the above student in the **CS23432- Software Construction** Laboratory during the year 2024-2025.

Signature of Faculty -in – Charge

Submitted for the Practical Examination held on _____

Internal Examiner

External Examiner

LAB PLAN
CS23432-SOFTWARE CONSTRUCTION LAB

Ex No	Date	Topic	Page No	Sign
1	21/01/2025	Study of Azure DevOps		
2	28/01/2025	Problem Statement		
3	04/02/2025	Agile Planning		
4	18/02/2025	Create User stories with Acceptance Criteria		
5	25/02/2025	Designing Sequence Diagrams using Azure DevOps-WIKI		
6	04/03/2025	Designing Class Diagram using Azure DevOps-WIKI		
7	11/03/2025	Designing Use case Diagram using Azure DevOps-WIKI		
8	18/03/2025	Designing Activity Diagrams using Azure DevOps-WIKI		
9	25/03/2025	Designing Architecture Diagram Using Star UML		
10	01/04/2025	Design User Interface		
11	08/04/2025	Implementation – Design a Web Page based on Scrum Methodology		
12	15/04/2025	Testing-Test Plan, Test Case and Load Testing		

Course Outcomes (COs)

Course Name: Software Engineering

Course Code: CS23432

CO 1	Understand the software development process models.
CO 2	Determine the requirements to develop software
CO 3	Apply modeling and modeling languages to design software products
CO 4	Apply various testing techniques and to build a robust software products
CO 5	Manage Software Projects and to understand advanced engineering concepts

CO - PO – PSO matrices of course

PO/PSO CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CS23432.1	2	2	3	2	2	2	2	2	2	2	3	2	1	3	-
CS23432.2	2	3	1	2	2	1	-	1	1	1	2	-	1	2	-
CS23432.3	2	2	1	1	1	1	1	1	1	1	1	1	2	2	1
CS23432.4	2	2	3	2	2	2	1	0	2	2	2	1	1	2	1
CS23432.5	2	2	2	1	1	1	1	0	2	1	1	1	2	1	-
Average	2.0	2.2	2.0	1.6	1.6	1.4	1.3	1.3	1.6	1.4	1.8	1.3	1.4	2.0	1.0

Correlation levels 1, 2 or 3 are as defined below:

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High) No correlation: “-“

EXP NO: 1

AZURE DEVOPS ENVIRONMENT SETUP

Aim:

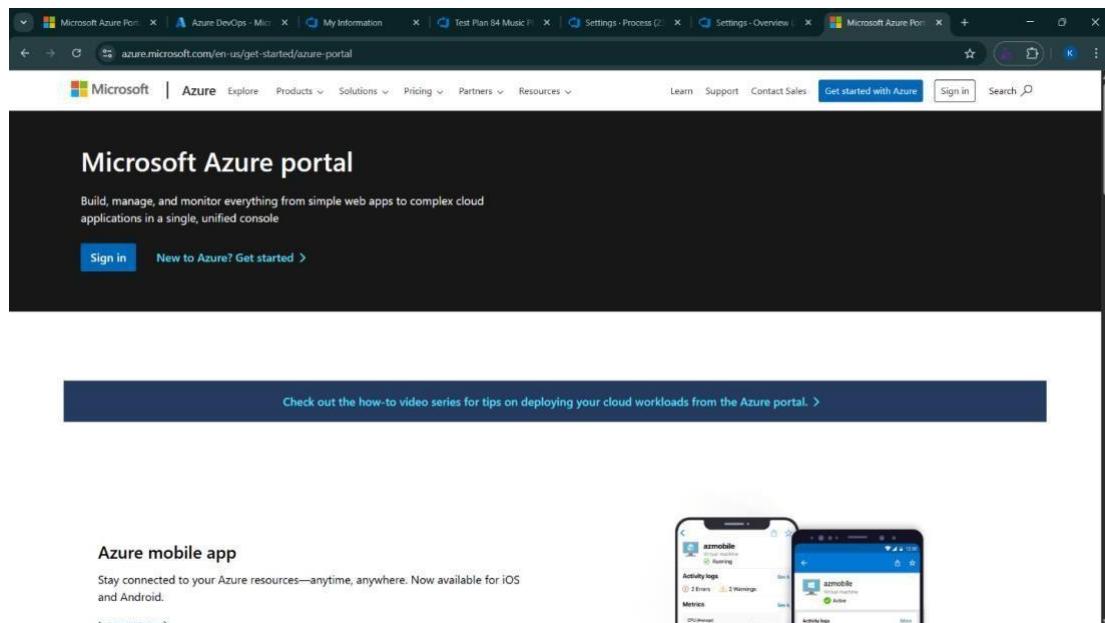
To set up and access the Azure DevOps environment by creating an organization through the Azure portal.

INSTALLATION

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



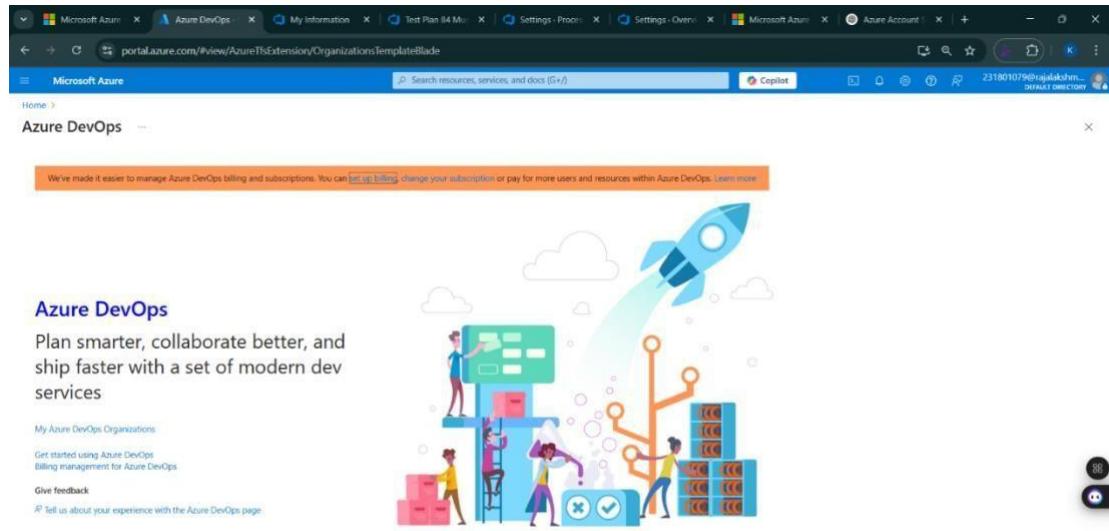
2. Azure home page

The screenshot shows the Microsoft Azure home page. At the top, there's a search bar and a Copilot button. Below the search bar, there are sections for 'Azure services' (with links to Create a resource, Azure DevOps organizations, Subscriptions, Dashboard hub, Resource groups, Azure Load Testing, Quickstart Center, Azure AI services, Kubernetes services, and More services), 'Resources' (listing recent items like 'Music' and 'Music_playlist_Batch_Creator'), 'Navigate' (links to Subscriptions, Resource groups, All resources, and Dashboard), 'Tools' (links to Microsoft Learn, Azure Monitor, Microsoft Defender for Cloud, and Cost Management), and 'Useful links' (links to Azure mobile app, Microsoft Learn, Azure Monitor, Microsoft Defender for Cloud, and Cost Management). A sidebar on the right shows a user profile and a 'Dashboard' section.

3. Open DevOps environment in the Azure platform by typing **Azure DevOps Organizations** in the search bar.

The screenshot shows the Microsoft Azure home page with a search bar containing the text 'Azure DevOps'. A dropdown menu appears below the search bar, listing 'Services' (7) and 'Marketplace' (31). Under 'Services', 'Azure DevOps organizations' is highlighted. Other service options include Azure Native New Relic Service, Managed DevOps Pools, and Azure Native Dynatrace Service. Under 'Marketplace', items like Static Web App, Rocky Linux 9, Build Agents for Azure DevOps, and InfluxDB Cloud (Official Version) are listed. The 'Documentation' section includes links to DevOps architecture design - Azure Architecture Center, Secure your Azure DevOps - Azure DevOps, Course AZ-400100-A: Designing and Implementing Microsoft DevOps solutions - T..., and Managed DevOps Pools Overview - Managed DevOps Pools. The 'Tools' section includes Microsoft Learn, Azure DevOps, and Continue searching in Microsoft Extra ID. The bottom of the screen shows the URL 'http://portal.azure.com/#blade/Azure%20Extensions/OrganizationsTemplateBlade' and an 'Azure mobile app' link.

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.

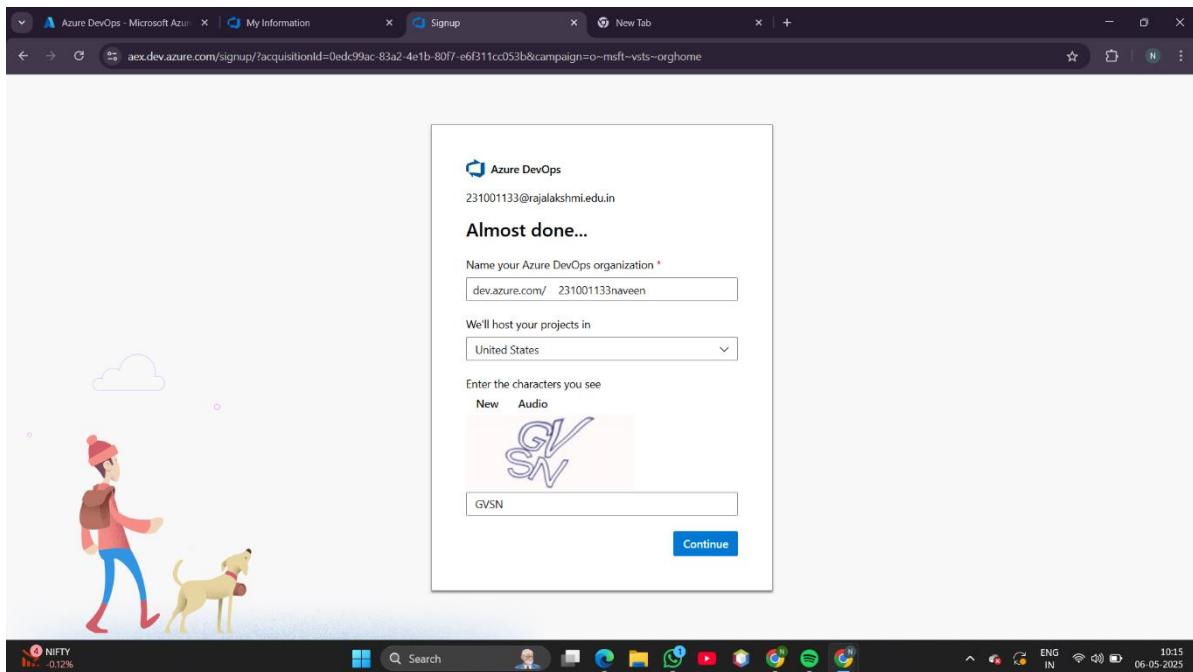
EXP NO: 2

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.

Create new project

X

Project name *

Music Playlist Batch Creator App

Description

Visibility



Public

Anyone on the internet can view the project. Certain features like TFVC are not supported.



Private

Only people you give access to will be able to view this project.

Public projects are disabled for your organization. You can turn on public visibility with [organization policies](#).

▼ Advanced

Cancel

Create

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.

The screenshot shows the Azure DevOps Organization Home page. At the top, there's a Microsoft logo and a sign-out link. On the left, a sidebar displays the user's profile picture (ND), name (Naveen Devadass), email (231001133@rajalakshmi.edu.in), location (India), and a Visual Studio Dev Essentials card. The main content area is titled "Azure DevOps Organizations" and shows the organization "dev.azure.com/231001133 (Owner)". It lists three projects: "Music Playlist Batch Creator", "Boot_Tech(1)", and "Music Playlist Batch Creator". There's also a "New project" button and an "Actions" section with a "Open in Visual Studio" link. A "Create new organization" button is at the top right. The URL in the address bar is dev.azure.com/231001133.

4. Project dashboard

The screenshot shows the Azure DevOps Project Overview page for the "Music Playlist Batch Creator" project. The top navigation bar includes tabs for "My Information", "Summary - Overview", "New Tab", and "Azure DevOps". The main content area has a title "Music Playlist Batch Creator" and sections for "About this project" and "Project stats". The "About this project" section contains a "Project Overview" paragraph and a "Like" button. The "Project stats" section shows "Boards" with 20 work items and "Members" with 3 users (ND, [redacted], [redacted]). The left sidebar has a "Music Playlist Batch Cr..." tab and links for "Overview", "Summary", "Dashboards", "Wiki", "Boards", "Repos", "Pipelines", "Test Plans", and "Artifacts". At the bottom, there's a "Project settings" link and a taskbar with various icons. The URL in the address bar is dev.azure.com/231001133/Music%20Playlist%20Batch%20Creator.

5. To manage user stories:

- 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.
- 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.

The screenshot shows the Azure DevOps interface for the 'Music Playlist Batch Creator Team'. The left sidebar is collapsed, showing options like Overview, Boards, Work items, Boards, Backlogs, Sprints, Queries, Delivery Plans, Analytics views, Repos, Pipelines, Test Plans, and Artifacts. The main area displays a backlog board titled 'Backlog' under 'Analytics'. At the top right are buttons for '+ New Work Item', 'View as Board', and 'Column Options'. Below is a table with columns: Order, Work Item Type, Title, State, Effort, Busin..., Value Area, and Tags. The backlog contains several items, mostly 'Feature' type, with titles such as 'Playlist Creation and Customization', 'Song Management Within Playlists', 'Song Search Functionality', 'Song Recommendations', 'Playlist Sharing and Collaboration', 'Playlist Sharing', and 'Collaborative Playlists'. All items are marked as 'New' in the State column and 'Business' in the Value Area column.

The screenshot shows the Microsoft sign-in page. At the top right, it says 'Sign out'. In the center, there's a large purple circular profile picture with the letters 'ND' in white. To the right of the picture, the name 'Naveen Devadass' is displayed in bold, along with the email address '231001133@rajalakshmi.edu.in'. Below the name is a blue link 'My Microsoft account'. Underneath the account info are links 'Switch directory' and '...'. At the bottom left is a circular icon with a person and a plus sign, next to the text '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, backlog for your assigned project.

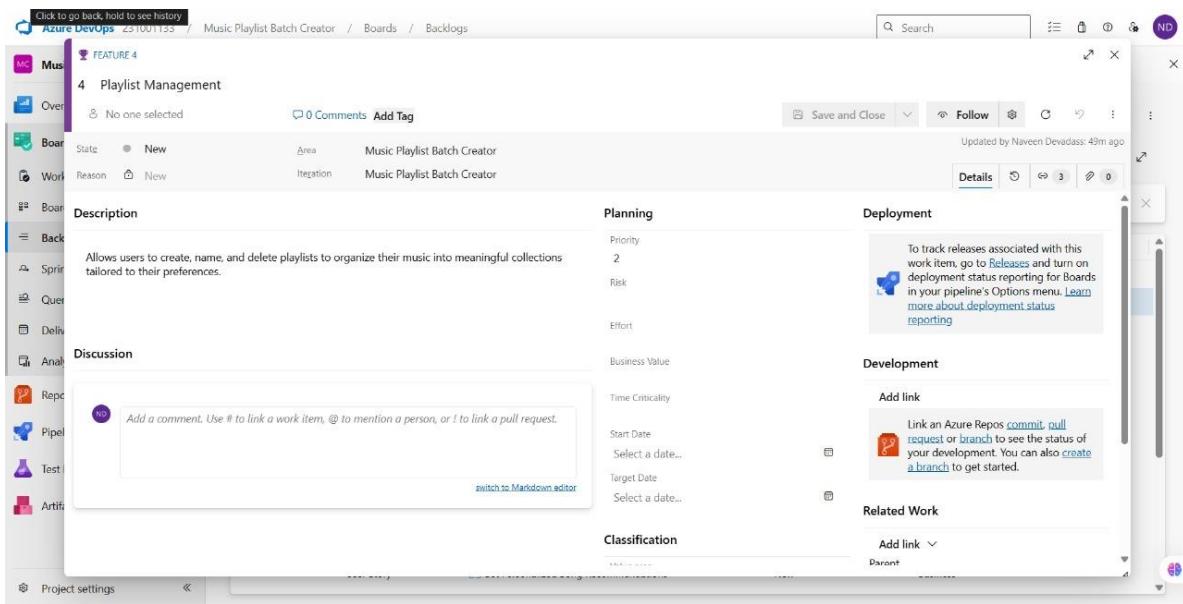
Create Epic, Features, User Stories, Task

The screenshot shows the Azure Boards Backlog view for the 'Music Playlist Batch Creator' project. The backlog is organized by work item type: Epic, Feature, and User Story. Each item has a title, state (New), effort (Business), and tags (Business). The backlog includes items like 'Playlist Creation and Customization', 'Song Management Within Playlists', and 'Create Playlist with Custom Name'. The interface includes a search bar, filter options, and various navigation links on the left side.

1. Fill in Epics

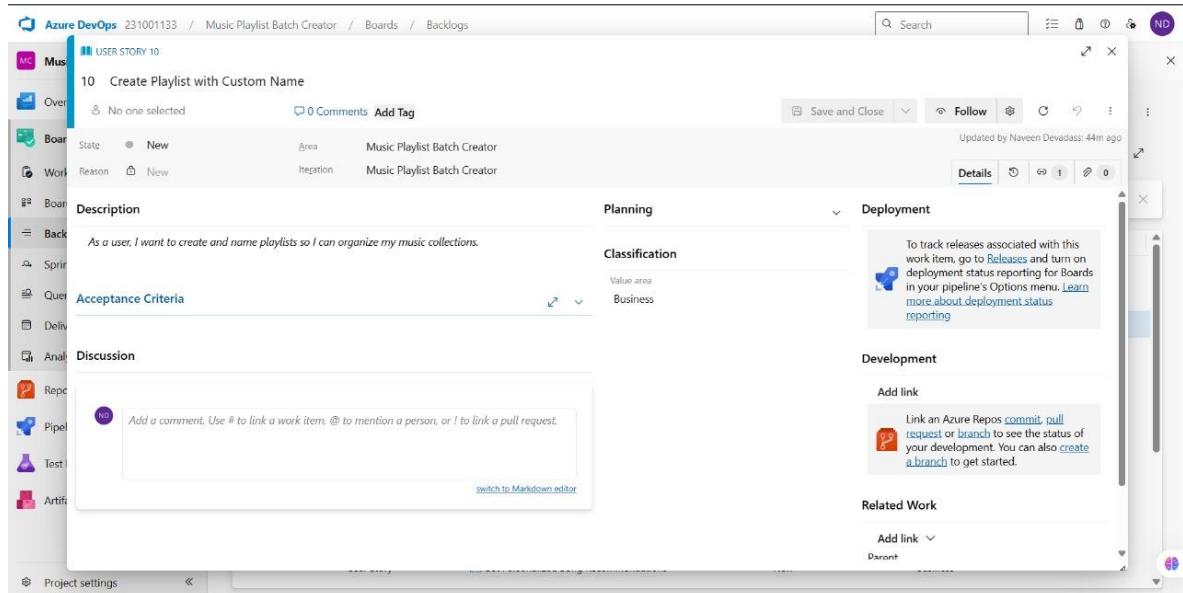
The screenshot shows the 'Create Epic' dialog in Azure DevOps. The epic is titled 'EPIC 1: Playlist Creation and Customization'. It has a state of 'New', is assigned to the 'Music Playlist Batch Creator' area, and is part of the 'Music Playlist Batch Creator' iteration. The dialog includes sections for 'Description', 'Planning' (Priority: 2, Risk), 'Deployment' (instructions for tracking releases), 'Development' (linking to a GitHub commit or pull request), 'Discussion' (comment input field), 'Classification' (tags dropdown), and 'Related Work' (add link button). The left sidebar shows the project structure with 'Backlogs' selected.

2. Fill in Features



The screenshot shows the 'Music Playlist Batch Creator' project in Azure DevOps. A new feature titled '4 Playlist Management' is being created. The 'Description' field contains the text: 'Allows users to create, name, and delete playlists to organize their music into meaningful collections tailored to their preferences.' The 'Planning' section shows a priority of 2. The 'Deployment' section includes a note about tracking releases. The 'Development' section has a placeholder for linking an Azure Repos commit or pull request. The 'Classification' section indicates the value area is 'Business'.

3. Fill in User Story Details



The screenshot shows the 'Music Playlist Batch Creator' project in Azure DevOps. A new user story titled '10 Create Playlist with Custom Name' is being created. The 'Description' field contains the text: 'As a user, I want to create and name playlists so I can organize my music collections.' The 'Classification' section indicates the value area is 'Business'. The 'Development' section has a placeholder for linking an Azure Repos commit or pull request. The 'Related Work' section is currently empty.

Result:

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

EXP NO: 4

SPRINT PLANNING

Aim:

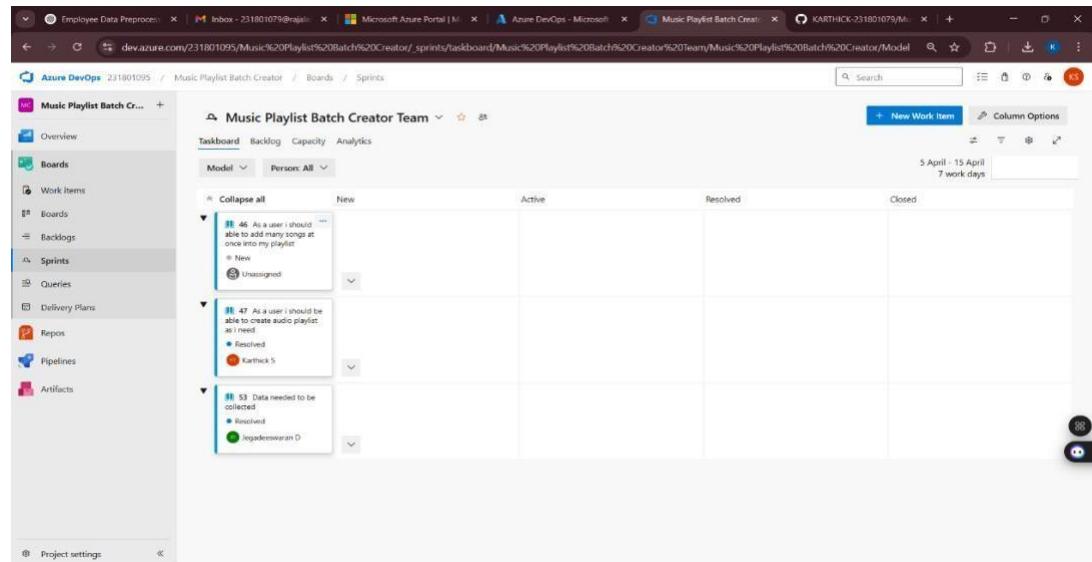
To assign user story to specific sprint for the Music Playlist Batch Creator Project.

Sprint Planning

Sprint 1

The screenshot shows the Azure DevOps Taskboard for the 'Music Playlist Batch Creator' project. The left sidebar is collapsed, showing the 'Boards' section is selected. The main area displays a taskboard with columns: New, Active, Resolved, and Closed. There are two tasks visible: '19 As a user I want to sign up and log in securely so that I can access my playlists' (New) and '21 Implement API-based authentication' (New). Both tasks are assigned to 'Karthikayen Senthil'. The top right of the board shows the sprint duration: '21 March - 4 April' and '4 work days remaining'. A 'New Work Item' button and 'Column Options' are also present at the top right.

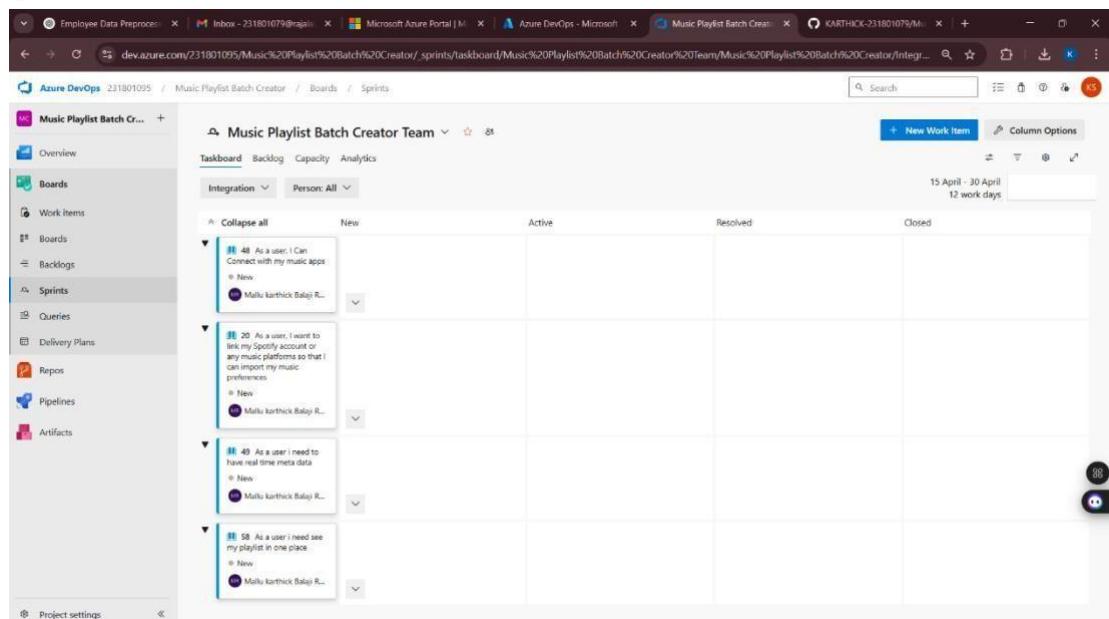
Sprint 2



The screenshot shows the Azure DevOps Taskboard for the 'Music Playlist Batch Creator Team' in Sprint 2. The board has four columns: New, Active, Resolved, and Closed. There are three work items visible:

- 46:** As a user I should be able to add many songs at once into my playlist.
 - New
 - Unassigned
- 47:** As a user I should be able to create audio playlist.
 - Resolved
 - Karthick S
- 53:** Data needed to be collected
 - Resolved
 - regadenwaram D

Sprint 3



The screenshot shows the Azure DevOps Taskboard for the 'Music Playlist Batch Creator Team' in Sprint 3. The board has four columns: New, Active, Resolved, and Closed. There are four work items visible:

- 48:** As a user, I Can Connect with my music apps.
 - New
 - Mallu karthick Balaji R...
- 20:** As a user, I want to link my Spotify account or any other platforms so that I can import my music preferences.
 - New
 - Mallu karthick Balaji R...
- 49:** As a user I need to have real time meta data.
 - New
 - Mallu karthick Balaji R...
- 58:** As a user I need see my playList in one place.
 - New
 - Mallu karthick Balaji R...

Sprint 4

The screenshot shows the Azure DevOps Taskboard for the 'Music Playlist Batch Creator Team' project. The board is divided into four columns: New, Active, Resolved, and Closed. Under the 'New' column, there are three tasks:

- S1 As a user i should be able to access a friendly and Modern UI
- S2 As a user i should be able to rename record and change the playlist
- S3 As a user i can able to customize my playlist

Each task has a status indicator: S1 is 'New', S2 is 'Unassigned', and S3 is 'New'. The 'Person: All' filter is selected. The top right corner shows the sprint duration: 29 April - 30 April, 2 work days.

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

The screenshot shows a Microsoft Azure DevOps work item details page for a User Story. The URL in the browser is https://dev.azure.com/231801095/Music%20Playlist%20Batch%20Creator/_backlogs/backlog/Music%20Playlist%20Batch%20Creator%20Team/Epics?workitem=47. The work item ID is 47, and the title is "USER STORY 47". The story points are set to 3, priority to 2, and risk to 1. The classification is under Value area Business. The acceptance criteria and discussion sections are empty. A related work item titled "Auto-Playlist Creation Based on user preference" is listed, updated on Feb 18.

Result:

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

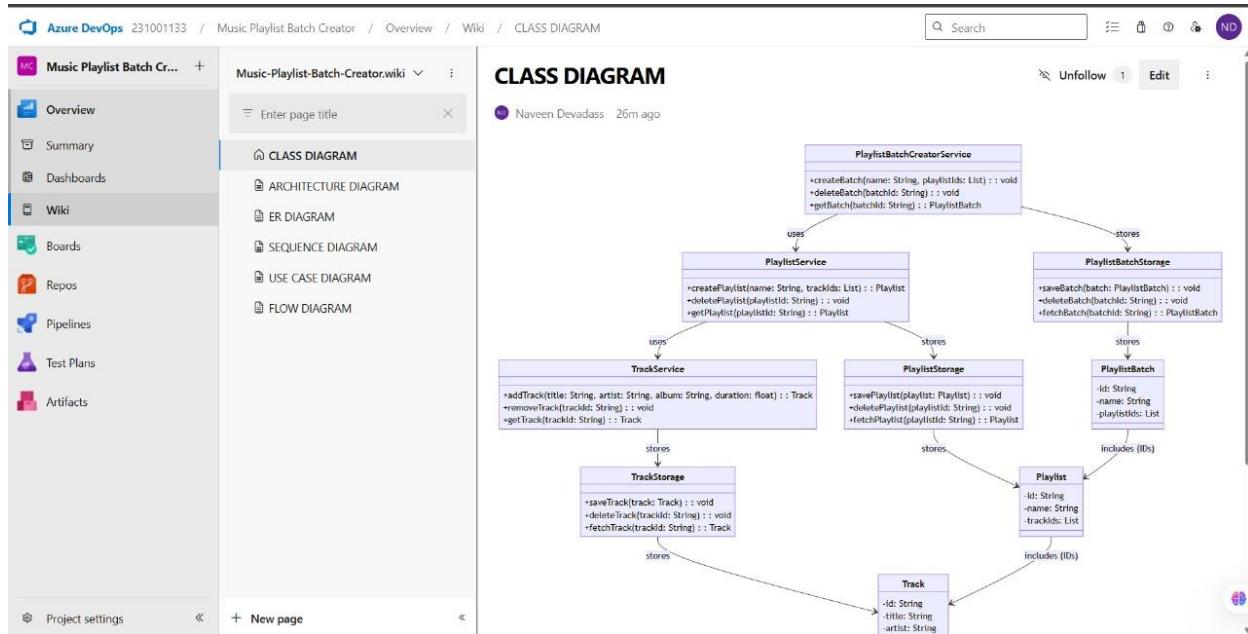
EXP NO: 6

DESIGNING CLASS AND SEQUENCE DIAGRAMS FOR PROJECT ARCHITECTURE

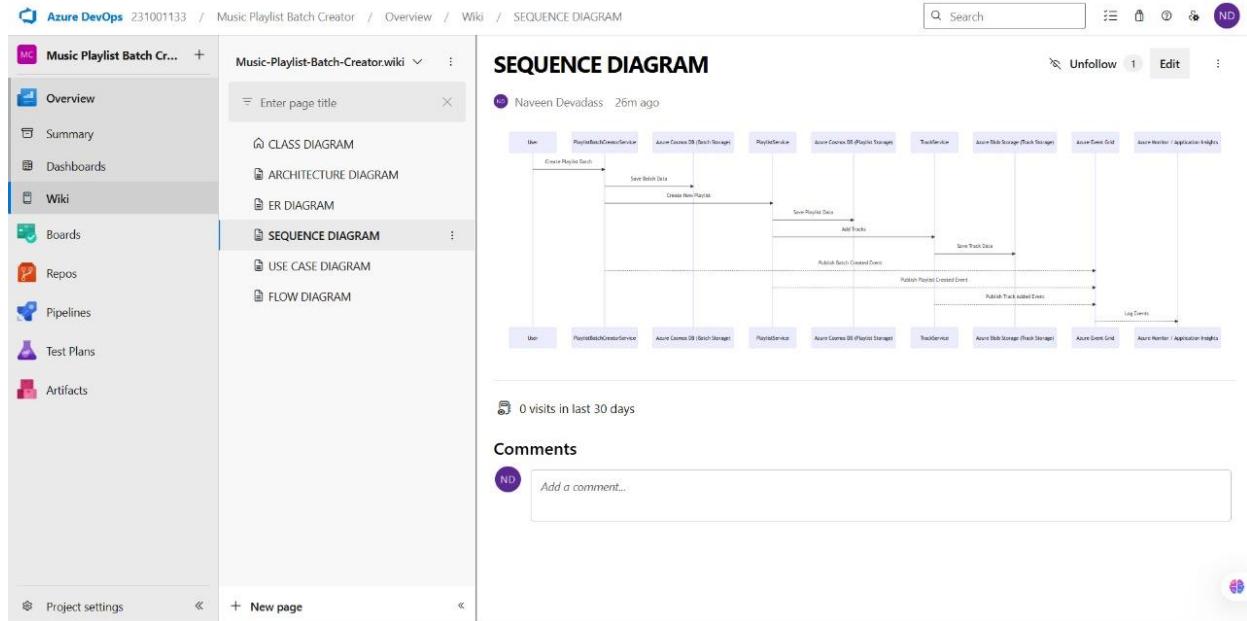
Aim:

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.

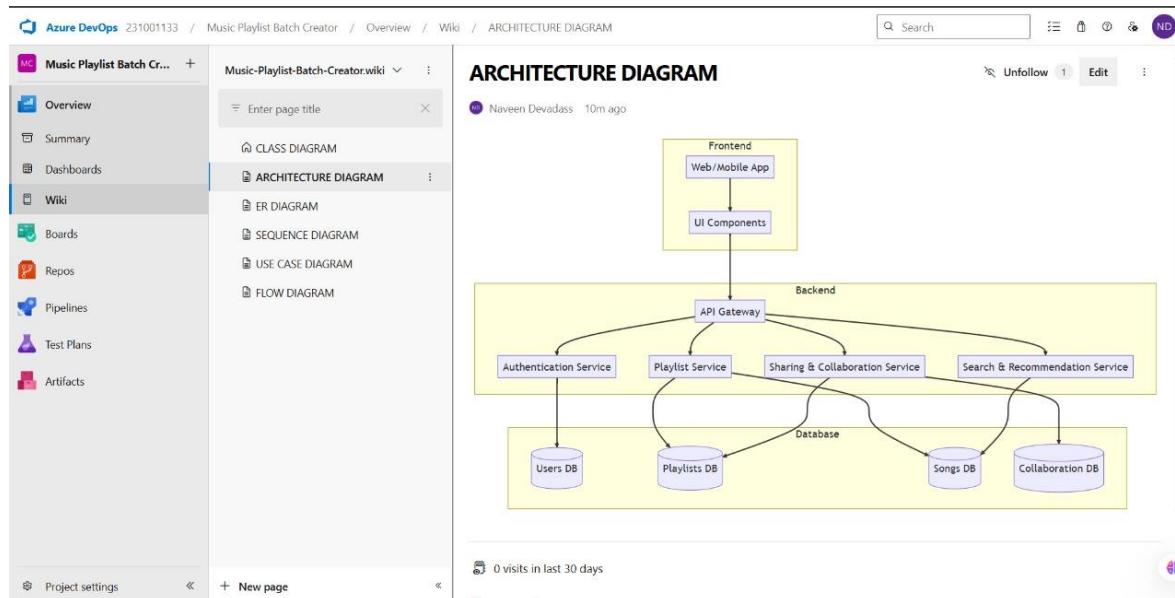
EXP NO: 7

DESIGNING ARCHITECTURAL AND ER DIAGRAMS FOR PROJECT STRUCTURE

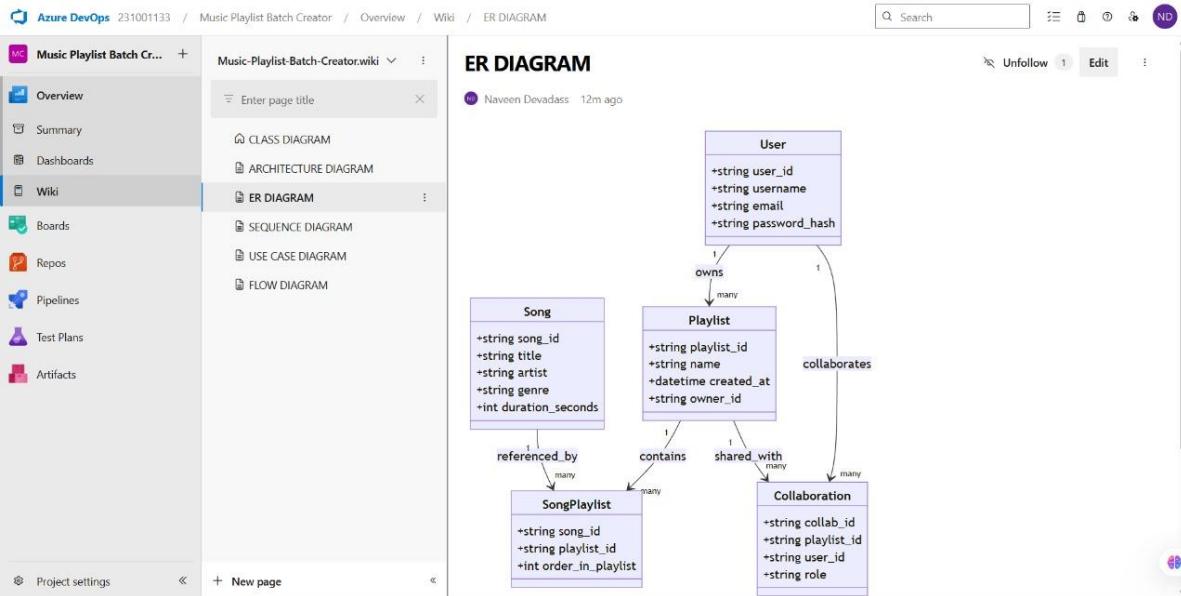
Aim:

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** ○ 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

3. Test case

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

Music Playlist Batch Creator – Test Plans

USER STORIES

- As a user, I want to sign up and log in securely so that I can access my playlists (ID: 79).

- As a user, I need to see my playlist in one place (ID: 76).
- As a user, I should be able to create an audio playlist as needed (ID: 73).
- As a user, I should be able to rename, record, and change the playlist (ID: 68).
- As a user, I need to have real-time metadata (ID: 65).

Test Suites

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

1. TC01 – Successful Sign Up

- Action:
 - Go to the Sign-Up page.
 - Enter valid name, email, and password.
 - Click "Sign Up".

○ Expected Results:

- - Sign-Up form is displayed.
 - Fields accept values without error.
 - Account is created, and the user is redirected to the dashboard.

○ Type: Happy Path

2. TC02 – Secure Login

○ Action:

- - Go to the Login page.
 - Enter valid email and password.
 - Click on "Login".

○ Expected Results:

- - Login form is displayed.
 - Fields accept data without error.
 - User is logged in and redirected to the dashboard.

○ Type: Happy Path

3. TC03 – Sign Up with Existing Email

○ Action:

- - Go to the Sign-Up page.
 - Enter a name and an already registered email.
 - Click on "Sign Up".

○ Expected Results:

- - Fields accept data.
 - Error message "Email already registered" is displayed.

○ Type: Error Path

4. TC04 – Login with Wrong Password

○ Action:

- - Go to the Login page.

- Enter valid email and incorrect password.
- Click on "Login".
- **Expected Results:**
 - Input is accepted.
 - Error message "Invalid username or password" is shown.
- **Type:** Error Path

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

1. TC05 – View Playlist Page

- **Action:**
 - Log in successfully.
 - Navigate to "My Playlists" section.
- **Expected Results:**
 - All created playlists are displayed clearly.
- **Type:** Happy Path

2. TC06 – Playlist Loading Failure

- **Action:**
 - Disconnect from the internet.
 - Navigate to "My Playlists".
- **Expected Results:**
 - Network is offline.
 - Error message "Unable to load playlists" is shown.
- **Type:** Error Path

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

1. TC07 – Real-Time Metadata Display

- **Action:**
 - Play a song.
 - Observe the metadata panel.
- **Expected Results:**
 - Metadata (title, artist, album, duration) is displayed and updates in real time.
- **Type:** Happy Path

2. TC08 – Metadata Not Updating

- **Action:**
 - Play a different song.
 - Observe the metadata panel.
- **Expected Results:**
 - Metadata remains static or shows default/fallback message.

- **Type:** Error Path

Test Suit: TS04 - Playlist Editing (ID: 89)

1. TC09 – Rename Playlist Successfully ○

Action:

- Navigate to "My Playlists".
- Click "Rename" next to a playlist.
- Enter a new name and click "Save".

- **Expected Results:**

- Playlist name updates successfully.

- **Type:** Happy Path

2. TC10 – Rename with Blank Name ○

Action:

- Click "Rename" on a playlist.
- Leave the field blank.
- Click "Save".

- **Expected Results:**

- Error message "Playlist name cannot be empty" is shown.

- **Type:** Error Path

3. TC11 – Change Playlist Order ○

Action:

- Open a playlist.
- Drag and drop songs to reorder.
- Click "Save".

- **Expected Results:**

- Playlist order is updated and saved.

- **Type:** Happy Path

4. TC12 – Change Playlist Order Fails ○

Action:

- Login and go to "My Playlists".
- Select a playlist.
- Go offline or simulate server error.
- Reorder songs and click "Save Order".

- **Expected Results:**

- Error message: "Failed to update order. Please check your connection".

- **Type:** Error Path

Test Suit: TS05 - Smart Playlist Creation (ID: 90)

1. TC13 – Generate Playlist Based on Various Categories ◦

Action:

- Login with valid credentials.
- Click on "Generate Playlist".
- Select categories.
- Click "Generate Playlist".

○ Expected Results:

- Playlist is generated based on selected mood and categories.

○ Type: Happy Path

2. TC14 – Fail to Generate Playlist Due to Missing Category Selection or Invalid Input ◦

Action:

- Login with valid credentials.
- Click on "Generate Playlist".
- Select categories.
- Click "Generate Playlist".

○ Expected Results:

- Error message: "Please select at least one valid category" or "No recommendations found for the selected filters".

○ Type: Error Path

Test Cases

The screenshot shows the Azure DevOps Test Plan interface. A test case titled "77 - TC06 - Playlist Loading Failure" is displayed. The test case details are as follows:

- Maintainer:** Karthick S
- Area:** Music Playlist Batch Creator
- Iteration:** Music Playlist Batch Creator\Integration
- Status:** Not Automated
- Priority:** 2
- Type:** Error Path

The **Steps** section contains two steps:

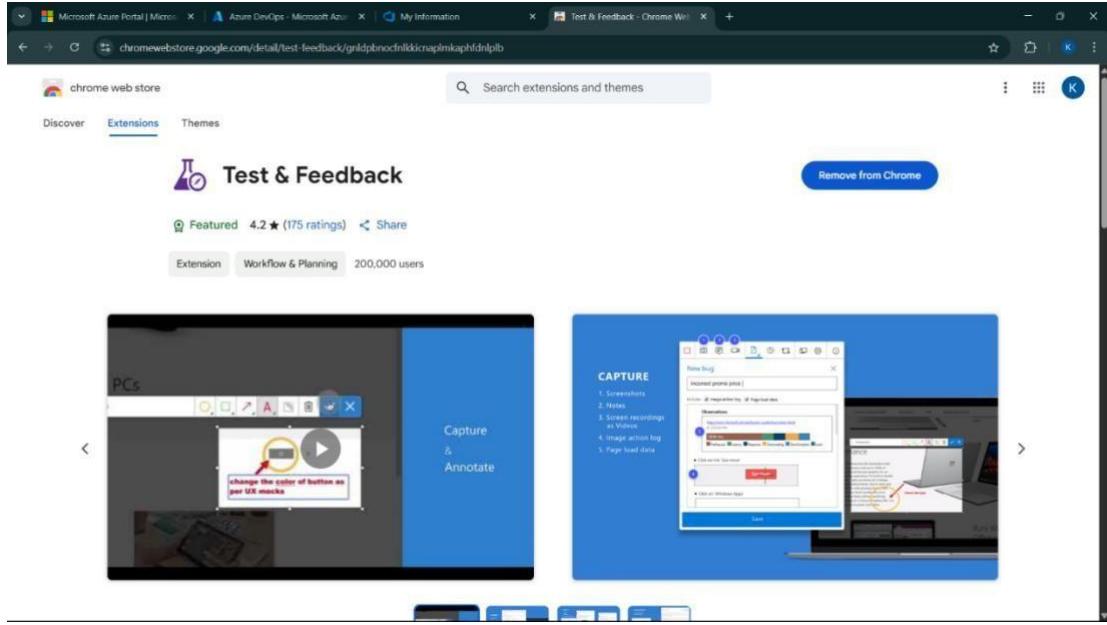
- Action: Disconnect from internet. Expected result: Network is offline.
- Action: Navigate to "My Playlists". Expected result: Error message "Unable to load playlists" is shown.

At the bottom left, there is a link to "Parameter values".

The screenshot shows the Azure DevOps Test Plan interface. A test case titled "75 TC05 – View Playlist Page" is selected. The "Steps" section contains two steps: "Log in successfully" (Expected result: User is redirected to dashboard) and "Navigate to "My Playlists" section" (Expected result: All created playlists are displayed clearly). The "Status" section indicates a priority of 2 and an automation status of "Not Automated".

4. Installation of test

The screenshot shows the Chrome Web Store page for the "Test & Feedback" extension. It has a rating of 4.2 stars from 175 ratings and 200,000 users. The extension is described as "Capture & Annotate". Two screenshots are shown: one demonstrating annotation on a UI element and another showing the extension's interface for capturing and analyzing bugs.



Test and feedback

Showing it as an extension

5. Running the test cases

TS02 - View Playlists (ID: 87)

Test Points (2 items)

Title	Outcome	Order	Test Case Id	Configuration	Tester
TC05 – View Playlist Page	Passed	1	75	Windows 10	Mallu karthick B...
TC06 – Playlist Loading Failure	Passed	2	77	Windows 10	Mallu karthick B...

Test Point Context Menu Options:

- View execution history
- Mark Outcome
- Run
- Reset test to active
- Edit test case
- Assign tester
- View test result

Runner - Test Plans - Google Chrome

75* TC05 – View Playlist Page

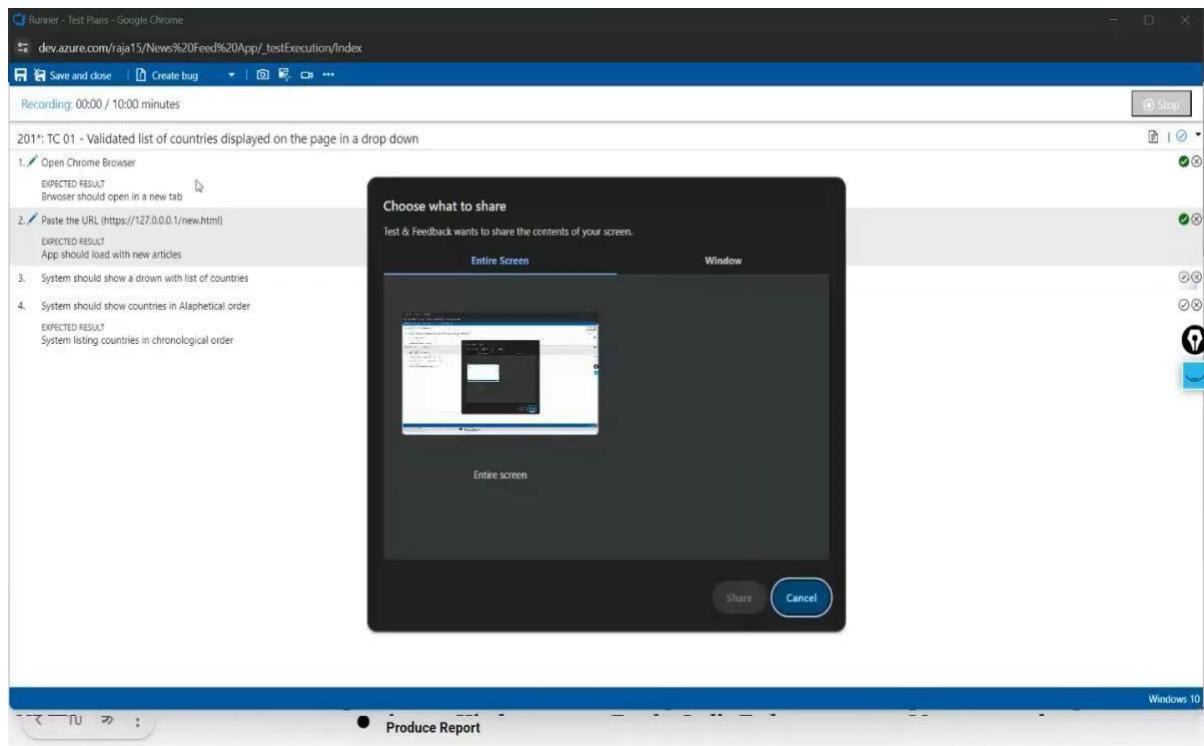
- Log in successfully
- Navigate to "My Playlists" section

EXPECTED RESULT

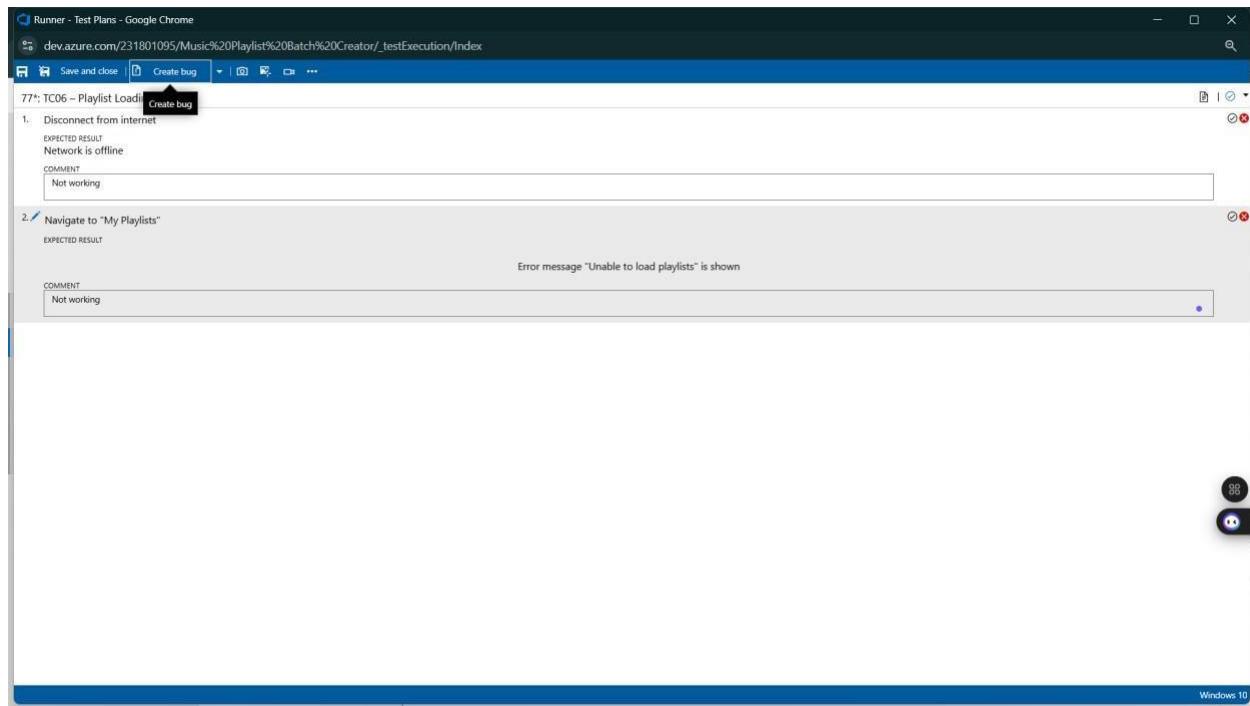
User is redirected to dashboard

All created playlists are displayed clearly

6. Recording the test case



7. Creating the bug



The screenshot shows a bug item in the Azure DevOps Test Plan interface. The title is "TB01 - Playlist loading spinner keeps spinning indefinitely on poor network". The status is "New" and the area is "Music Playlist Batch Creator". The repro steps describe a failure to connect to the internet. The system info table includes details like browser (Google Chrome 135), operating system (Windows NT 10.0: Win64: x64), and processor (11th Gen Intel(R) Core(TM) i3-1115G4 @ 3.00GHz). The deployment section shows it's integrated into a build.

This screenshot shows the same bug item from the previous screen, but with more detailed system information. The "System Info" table is expanded to show memory, display, and device pixel ratio details. The "Discussion" section at the bottom has a note about using #, !, or @ to link work items, pull requests, or mentions.

8. Test case results

The screenshot shows the Azure DevOps interface for a project named "Music Playlist Batch Creator". The left sidebar is open, showing options like Overview, Boards, Repos, Pipelines, Test Plans, Test plans, Progress report, Parameters, Configurations, Runs, and Artifacts. Under "Test Plans", "Test suites" is selected, and a list of suites is shown, including "TS01 - User Login (4)", "TS02 - View Playlists (2)", "TS03 - Real-Time Met...", "TS04 - Playlist Editing (4)", and "TS05 - Smart Playlist ...". The "TS02 - View Playlists (ID: 87)" suite is currently selected. The main area displays "Test Case Results" for "TC05 - View Playlist Page". The results table shows the following data:

Outcome	TimeStamp	Configuration	Run by	Tester	Test Plan
Passed	4m ago	Windows 10	Karthick S	Malu karthick Balaji ... Music	
Passed	12m ago	Windows 10	Karthick S	Malu karthick Balaji ... Music	
Not Applicable	12m ago	Windows 10	Karthick S	Malu karthick Balaji ... Music	
Passed	14m ago	Windows 10	Karthick S	Malu karthick Balaji ... Music	
Passed	Tuesday	Windows 10	Kartikeyan Senthil	Malu karthick Balaji ... Music	
Passed	Saturday	Windows 10	Malli karthick Balaji ...	Malu karthick Balaji ... Music	
Failed	Saturday	Windows 10	Malli karthick Balaji ...	Malu karthick Balaji ... Music	
Passed	Apr 11	Windows 10	Karthick S	Malu karthick Balaji ... Music	
Passed	Apr 11	Windows 10	Karthick S	Malu karthick Balaji ... Music	

9. Test report summary

The screenshot shows the Azure DevOps interface for a project named "News Feed App". The left sidebar is open, showing options like Overview, Boards, Work items, Boards, Backlogs, Sprints, Queries, Delivery Plans, Analytics views, Repos, Pipelines, Test Plans, Artifacts, and Project settings. The "Work items" section is selected, showing a list of work items. One work item is detailed, titled "BUG 203: BG 01 - Countries Drop down Not Available on the page". The work item details are as follows:

- Title:** BUG 203: BG 01 - Countries Drop down Not Available on the page
- Assignee:** rajesh prabhu
- Comments:** 0 Comments
- Tags:** Add Tag
- State:** New
- Reason:** New
- Iteration:** News Feed App
- Repro Step:** Active
- Result:** Passed
- Description:** Open Chrome Browser
Expected Result
Browser should open in a new tab
- Step no.:** 1
- Result:** Passed
- Description:** Paste the URL (<https://127.0.0.1/new.html>)
Expected Result
App should load with new articles
- Step no.:** 2
- Result:** Failed
- Description:** Section should show a dropdown with list of countries
Reason: Produce Report

Planning: Resolved Reason: Validated list of countries displayed on the AM, Story Points: 2, Priority: 2, Severity: 3 - Medium, Activity: Effort (Hours): Original Estimate.

Deployment: To track releases associated with this work item, go to [Releases](#) and turn on deployment status reporting for Boards in your pipeline's Options menu. Learn more about deployment status reporting.

Development: Add link: Link an Azure Repos [commit](#), [pull request](#) or [branch](#) to see the status of your development. You can also [create a branch](#) to get started.

Related Work:

- Assigning bug to the developer and changing state

92 TB01 - Playlist loading spinner keeps spinning indefinitely on poor network

Step no. **Result** **Title**

1. Failed Disconnect from internet
2. Failed Expected Result Network is offline

Comments: Page Not loading
Navigate to "My Playlists"

Expected Result

Error message "Unable to load playlists" is shown

Test Configuration: Windows 10

Planning

- Resolved Reason
- Story Points
- Priority: 2
- Severity: 3 - Medium
- Activity

Deployment

To track releases associated with this work item, go to Boards and turn on deployment status reporting for Boards in your pipeline's Options menu. Learn more about deployment status reporting

Development

+ Add link

Link an Azure Repos commit, pull request or branch to see the status of your development. You can also create a branch to get started.

Related Work

+ Add link

Add an existing work item as a parent

Tested By

7 TC06 – Playlist Loading Failure Updated 10-04-2025. Design

System Info

10. Progress report

Progress report

Summary

- 1 Test plans
- 14 Test points
- 14 (14 / 14) Test points run
- 100% Run
- 100% (14 / 14) Pass rate
- 14 Passed

Outcome trend

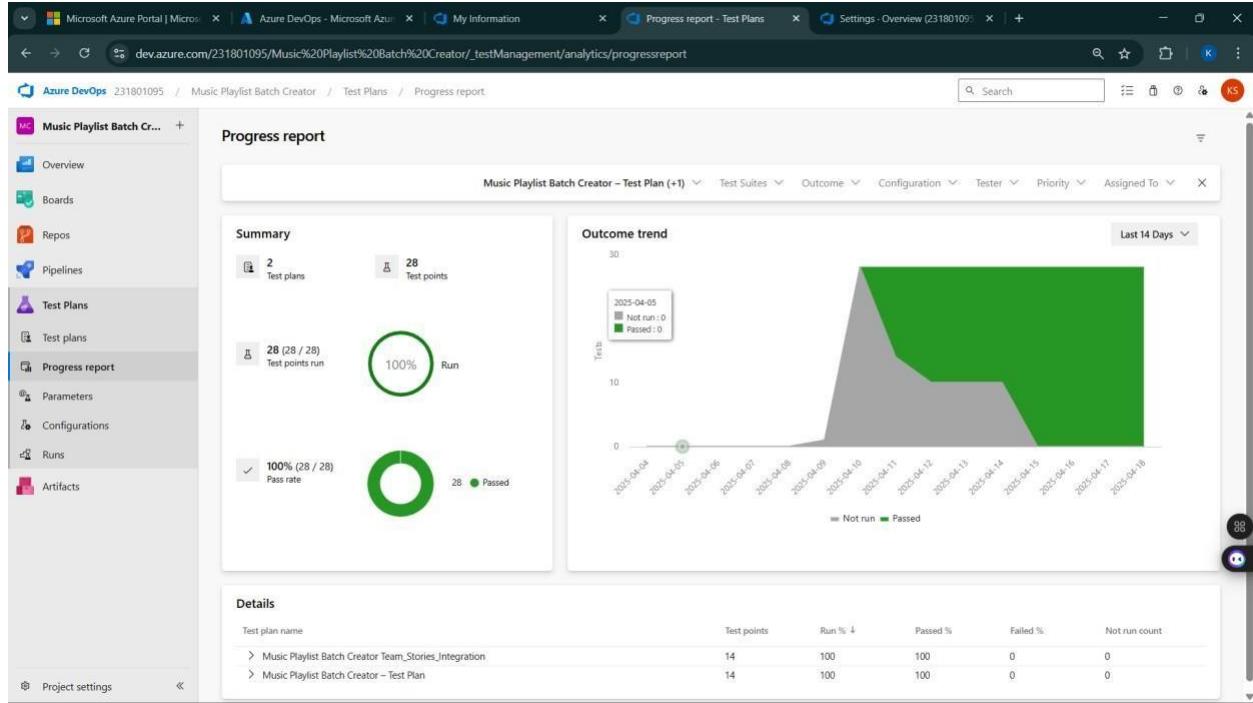
Last 14 Days

Date	Not run	Passed
2025-04-01	14	0
2025-04-02	14	0
2025-04-03	14	0
2025-04-04	14	0
2025-04-05	14	0
2025-04-06	14	0
2025-04-07	14	0
2025-04-08	14	0
2025-04-09	14	0
2025-04-10	0	14
2025-04-11	0	14
2025-04-12	0	14
2025-04-13	0	14
2025-04-14	0	14
2025-04-15	0	14

Details

Test plan name

Test plan name	Test points	Run %	Passed %	Failed %	Not run count
Music Playlist Batch Creator – Test Plan	14	100	100	0	0
> TS01 - User Login	4	100	100	0	0
> TS02 - View Playlists	2	100	100	0	0
> TS03 - Real-Time Metadata	2	100	100	0	0
> TS04 - Playlist Editing	4	100	100	0	0
> TS05 - Smart Playlist Creation	2	100	100	0	0



11. Changing the test template

All processes

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 practic...	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 recor...	0

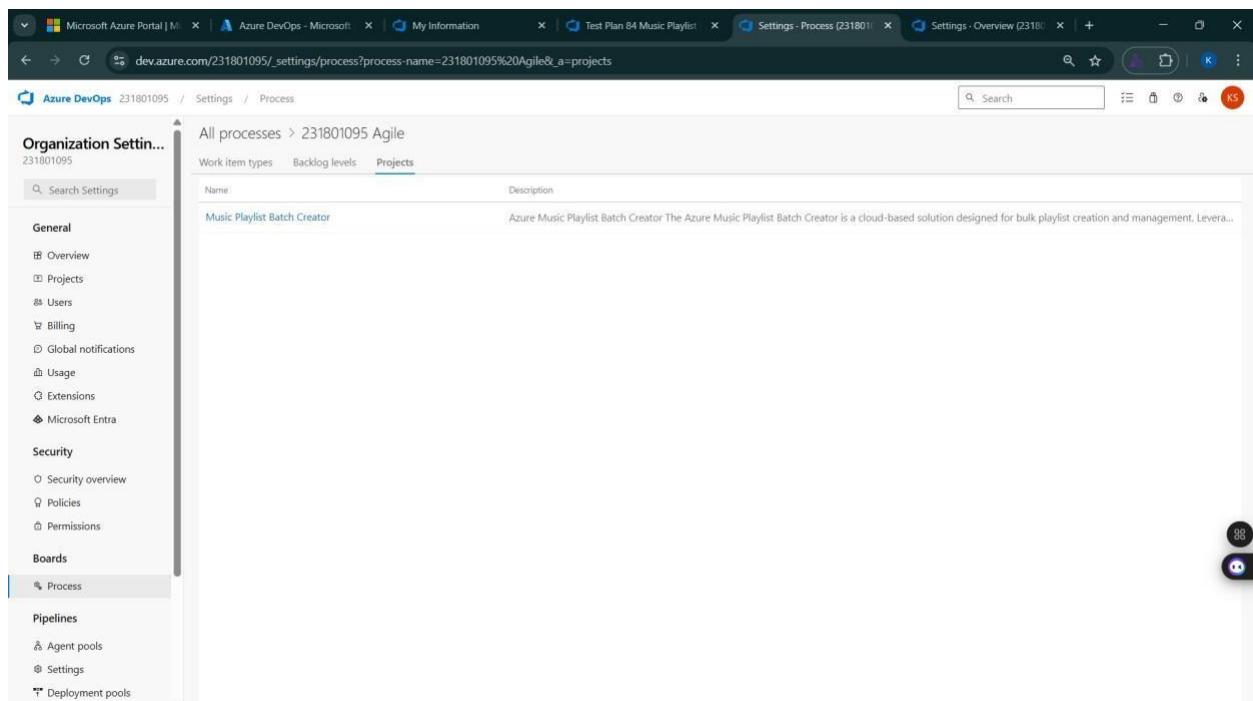
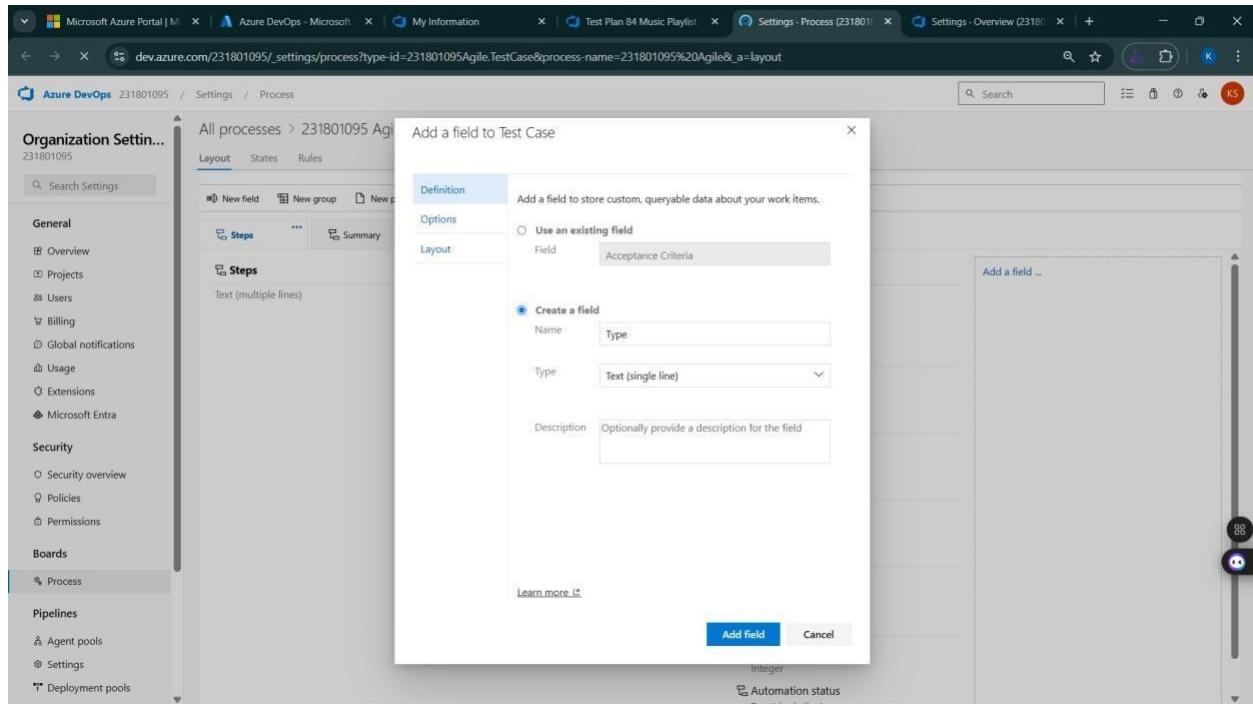
The screenshot shows the Azure DevOps Settings - Process page. On the left, there is a navigation sidebar with sections like General, Security, Boards, Pipelines, and Process. The Process section is currently selected. The main content area displays a list of process templates under 'All processes'. The 'Agile' section is expanded, showing three options: 'Basic', 'Scrum', and 'CMMI'. The 'Basic' template is described as flexible for any process and great for teams getting started with Azure DevOps. The 'Scrum' template is for teams who follow the Scrum framework. The 'CMMI' template is for more formal projects requiring a framework for process improvement and an auditable record.

Name	Description	Team projects
Basic	This template is flexible for any process and great for teams getting started with Azure DevOps.	0
Agile	...	0
Scrum	This template is for teams who follow the Scrum framework.	0
CMMI	...	0

The screenshot shows the same Azure DevOps Settings - Process page, but now the 'Agile' section is expanded to show a new entry: '231801095 Agile (default)'. This indicates that a new Agile template has been created for the organization. The other templates ('Basic', 'Scrum', 'CMMI') remain visible.

Name	Description	Team projects
Basic	This template is flexible for any process and great for teams getting started with Azure DevOps.	0
Agile	...	0
231801095 Agile (default)	This template is flexible and will work great for most teams using Agile planning methods, including those pract...	1
Agile Plus	...	0
Scrum	This template is for teams who follow the Scrum framework.	0
CMMI	...	0

12. View the new test case template



The screenshot shows the Azure DevOps Settings - Process page. The URL in the address bar is dev.azure.com/231801095/_settings/process?type-id=231801095Agile.TestCase&process-name=231801095%20Agile&_a=layout. The page title is "Settings - Process (231801095)". The left sidebar shows "Organization Settings" for project ID 231801095, with "Process" selected under Boards. The main content area shows a table for creating a new field. The first column has "Steps" selected. The second column shows "Custom" type, "Text (single line)" as the subtype, and a "Recent test results" section. The third column contains sections for "Deployment", "Development", "Related Work", and "Status". A search bar at the top right contains "Search" and a red circular icon with the number "15".

Result:

The test plans and test cases for the user stories is created in Azure DevOps with Happy Path and Error Path

EXP NO: 9

LOAD TESTING AND PERFORMANCE TESTING

Aim:

To create an Azure Load Testing resource and run a load test to evaluate the performance of a target endpoint.

Load Testing

Steps to Create an Azure Load Testing Resource:

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

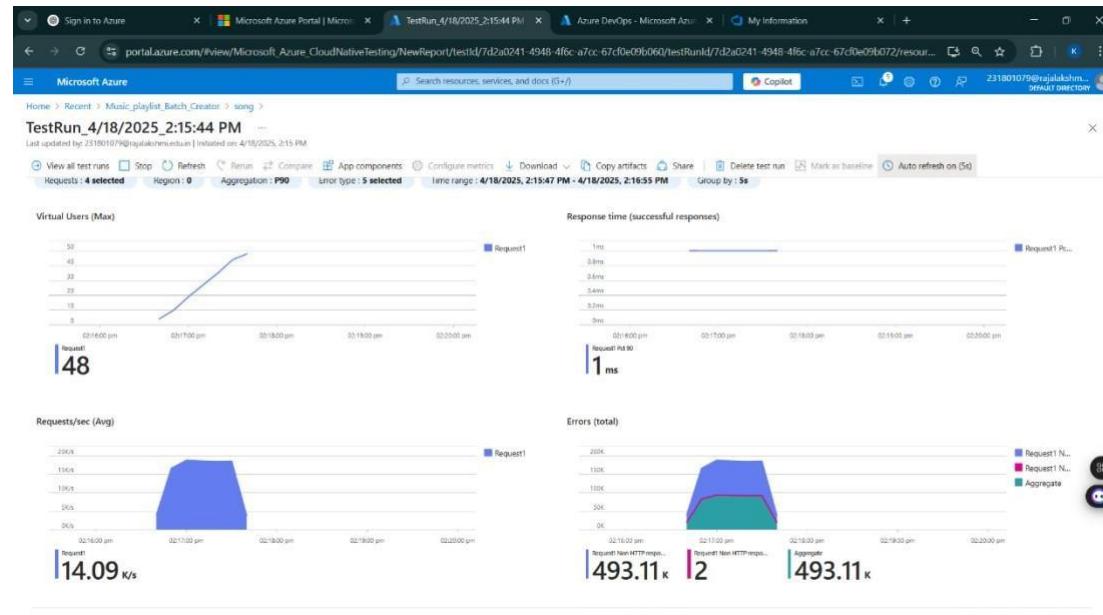
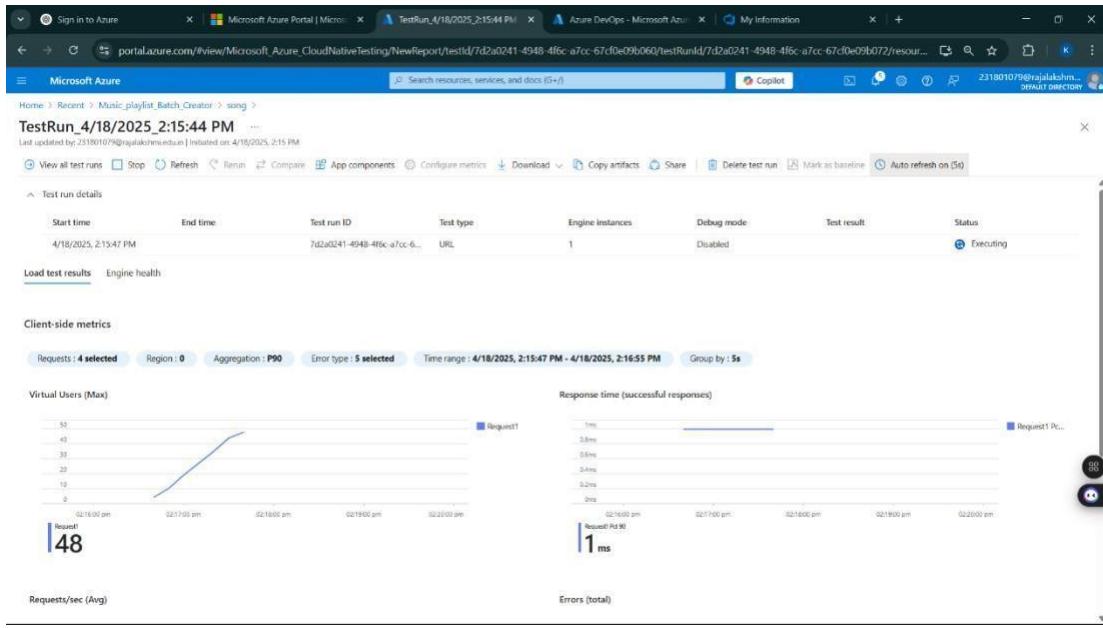
1. Sign in to Azure Portal
Go to <https://portal.azure.com> and log in.
2. Create the Resource Go to *Create a resource* → Search for “Azure Load Testing”.
 Select Azure Load Testing and click Create.
3. Fill in the Configuration Details Subscription: Choose your Azure subscription. Resource Group: Create new or select an existing one. Name: Provide a unique name (no special characters).
 Location: Choose the region for hosting the resource.
4. (Optional) Configure tags for categorization and billing.
5. Click Review + Create, then Create.
6. Once deployment is complete, click Go to resource.

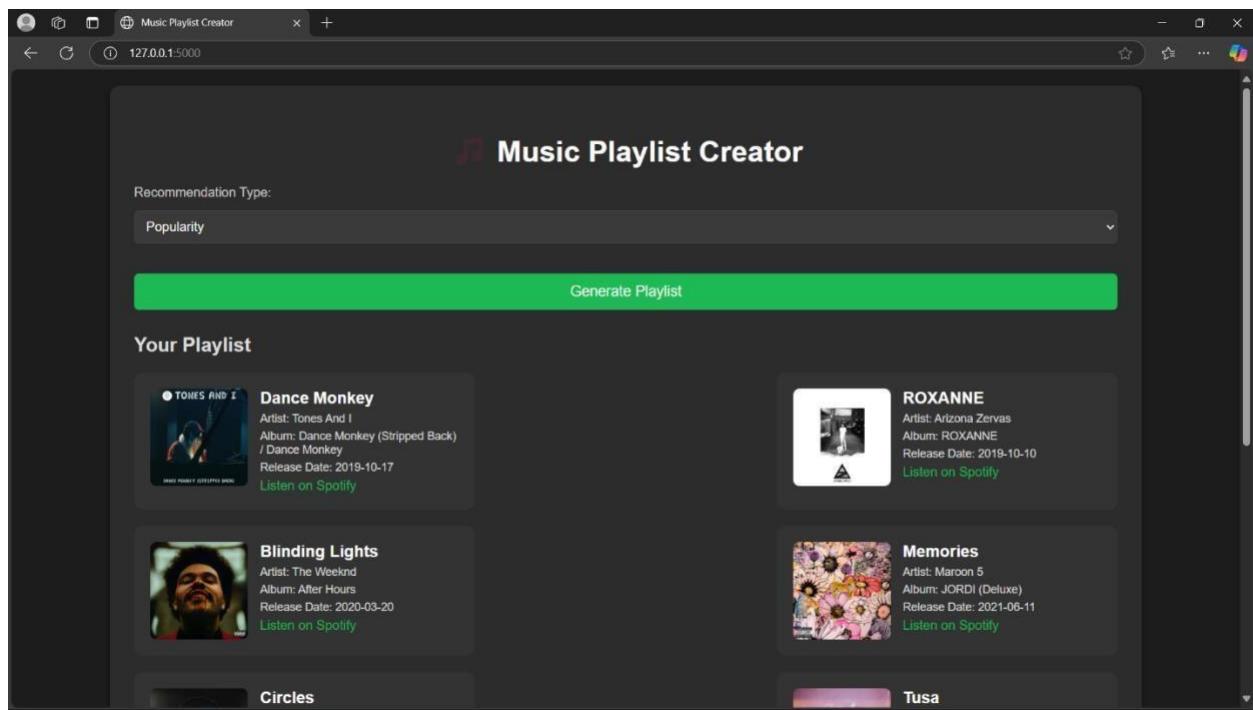
Steps to Create and Run a Load Test:

Once your resource is ready:

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

Load Testing





Result:

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

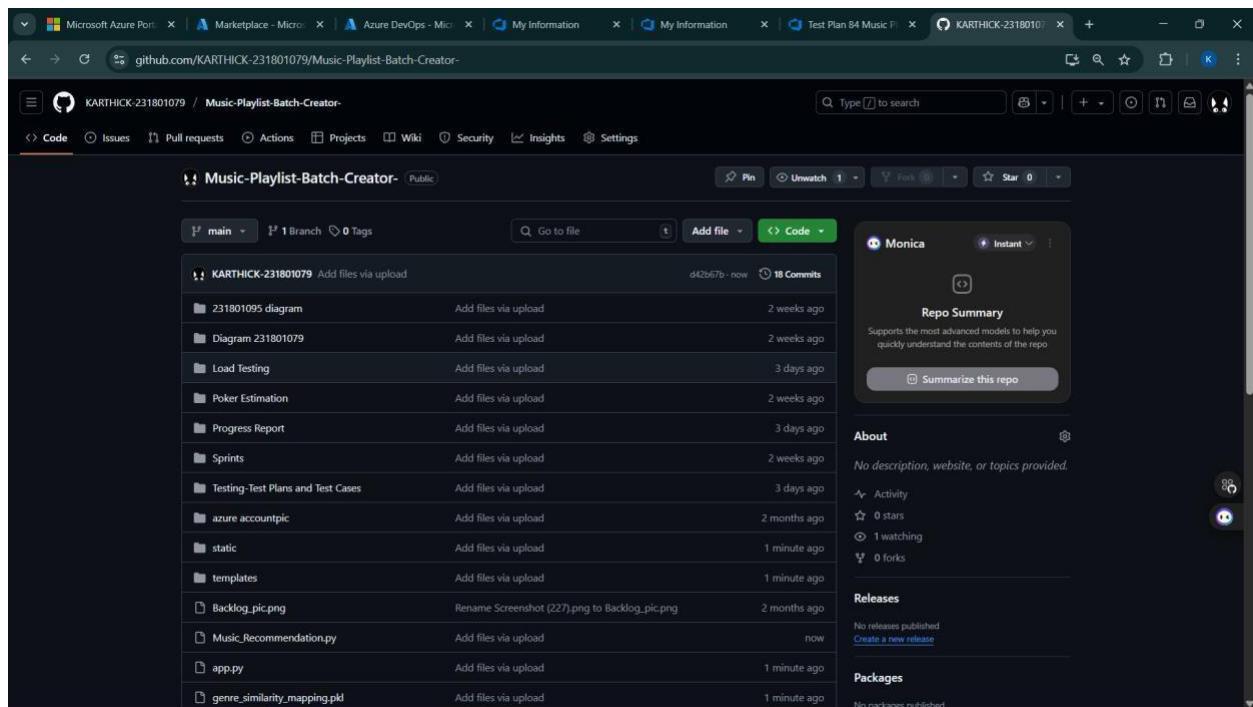
EXP NO: 10

GITHUB: PROJECT STRUCTURE & NAMING CONVENTIONS

Aim:

To provide a clear and organized view of the project's folder structure and file naming conventions, helping contributors and users easily understand, navigate, and extend the Music 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.

EXP NO : 12

DATE : 12.04.2025

TESTING – TEST PLAN, TEST CASE, LOAD TESTING

AIM:

To design and manage structured test plans and test cases in Azure DevOps for validating core user stories through both happy path and error scenarios and evaluate the performance of the application's endpoint by creating and executing load tests using Azure Load Testing.

PROCEDURE:

TEST CASE DESIGN PROCEDURE

1. Understand Core Features of the Application

- Review requirement documents and user stories.
- Identify all main functionalities of the application.
- Ensure complete coverage of modules before test case creation.

2. Define User Interactions

- Determine common user behaviors based on application flow.
- Translate user actions into testable scenarios.
- Ensure each test case mimics a real user operation.

3. Design Happy Path Test Cases

- Create test cases for expected and correct user actions.
- Ensure each functionality works under normal conditions.
- Add these cases under the relevant Test Suite in Azure DevOps.

4. Design Error Path Test Cases

- Identify edge cases, invalid inputs, and system failures.
- Test how the system handles incorrect or unexpected behavior.
- Add these test cases to the same or a separate Test Suite in Azure DevOps.

5. Break Down Steps and Expected Results

- Write step-by-step instructions in the "Steps" section of the test case.
- Provide expected results for each action.

- Ensure clarity for both manual execution and automation mapping.

6. Use Clear Naming and IDs

- Name test cases clearly using a defined naming convention (e.g., TC01, TC02, etc.).
- Ensure titles reflect the purpose of the test case.
- Azure DevOps auto-generates test case IDs for tracking.

7. Separate Test Suites

- Group test cases based on functionality (e.g., Login, Playlist, Recommendations).
- Use Static, Requirement-based, or Query-based suites in Azure DevOps.
- Improves traceability and execution flow.

8. Prioritize and Review

- Mark test cases with priority (High, Medium, Low).
- Review test cases for completeness and correctness.
- Ensure alignment with associated user stories or features.

1. New test plan

The screenshot shows the Azure DevOps interface for managing test plans. On the left, there's a sidebar with project navigation options like Overview, Boards, Repos, Pipelines, and Test Plans. Under Test Plans, 'Test plans' is selected. In the main area, a test plan titled 'Demo.Day' is displayed. The plan has a status of 'Map 5 - May' and '12' items. It includes a progress report showing '100% run, 100% passed'. Below this, a 'Test Suites' section lists 'Demo.Day (1)'. A table titled 'Test Cases (1 item)' shows one entry: 'Join Group and Post in Group' with Order 1, Test Case Id 00, Assigned To Silvia M, and State Design. There are tabs for 'Define', 'Execute', and 'Chart'. The top right corner shows standard browser controls and a search bar.

2. Test case

The screenshot shows a Microsoft Azure DevOps Test Case details page for 'TEST CASE 80'. The test case title is 'Join Group and Post in Group'. The test step table contains three rows:

Step	Action	Expected result
1.	Login and go to "Groups"	List of available groups is shown
2.	Click "Join" on a group	Confirmation message of successful joining
3.	Click "Post" in group and submit content	Post is visible in the group feed

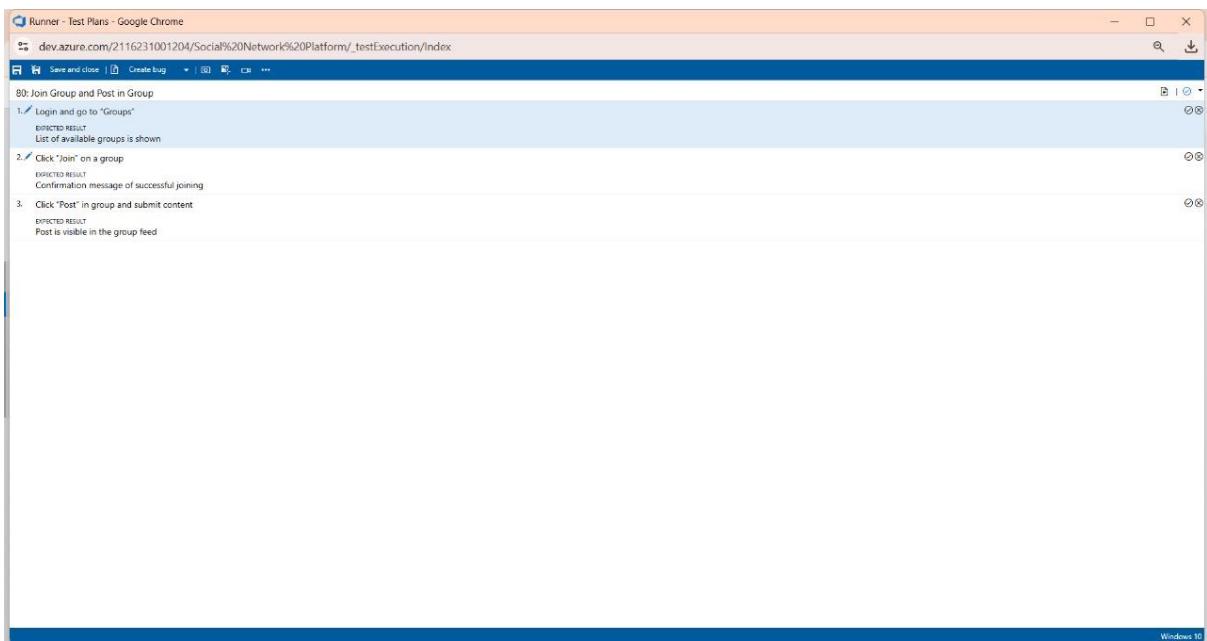
Below the table, there is a note: 'Click or type here to add a step'.

On the right side of the page, there are sections for Deployment, Development, Related Work, and Status. Deployment includes a note about tracking releases. Development includes a link to Azure Repos. Related Work lists a task titled 'Post Not Displayed in Group Feed After Submission' with status 'Resolved'. The Status section shows a priority level of 'Priority'.

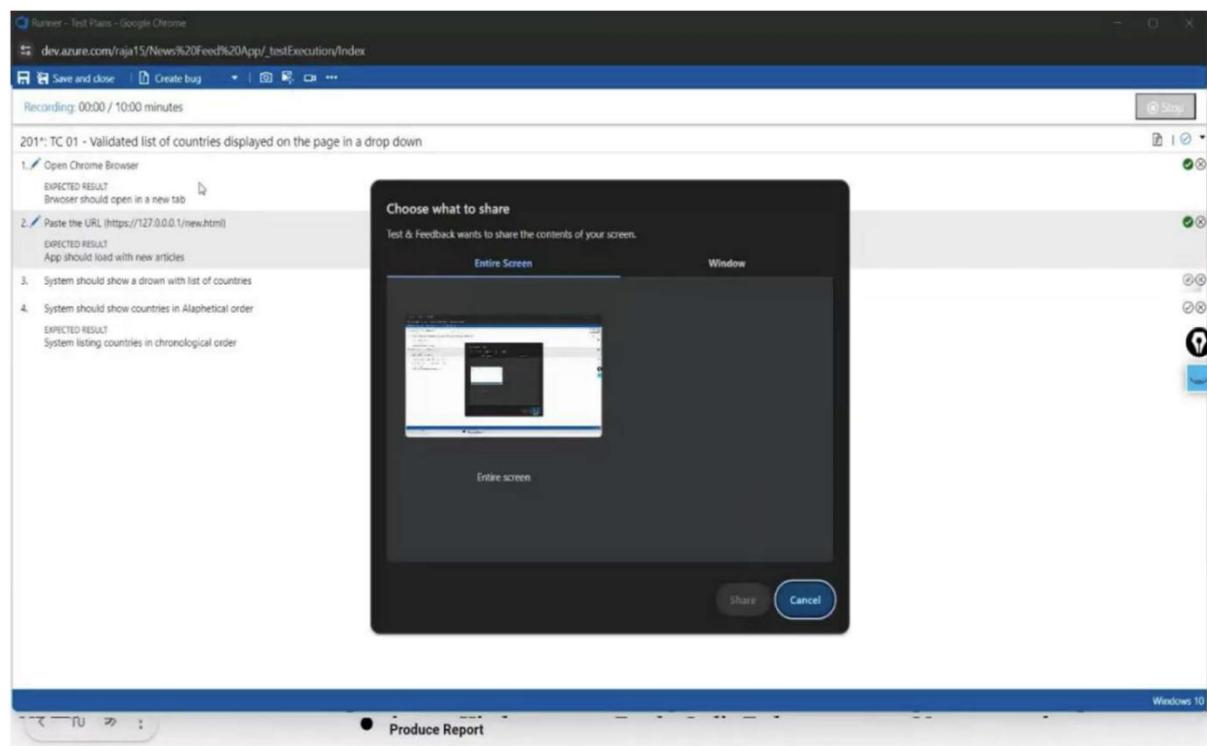
3. Installation of Test

The screenshot shows the Chrome Web Store page for the 'Test & Feedback' extension. The extension has a rating of 4.2 stars from 175 ratings and 200,000 users. It is categorized under 'Workflow & Planning'. The page features two main screenshots: one showing a user interface for 'Capture' and 'Annotate' with a red box highlighting a button, and another showing a recording interface with a timeline and various controls.

4. Running the Test Cases



5. Recording the Test Cases



6. Creating Bugs

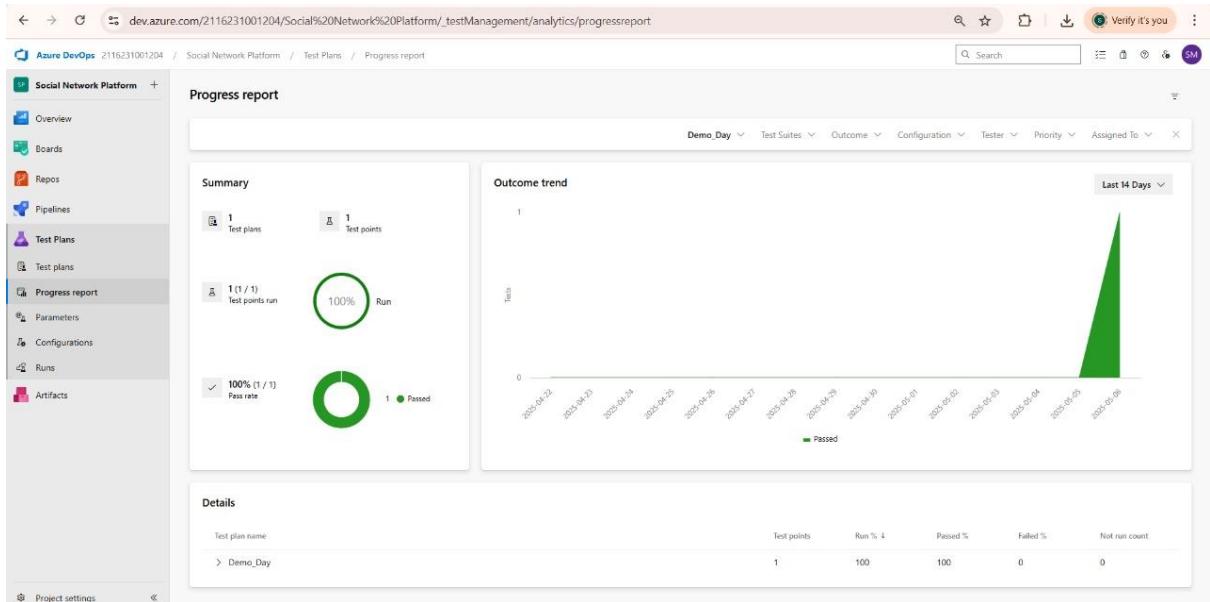
The screenshot shows the Azure DevOps Work Items page for the 'Social Network Platform' project. A new bug has been created with the ID BUG 81, titled '81 Post Not Displayed in Group Feed After Submission'. The bug was filed by Siva M on 5/6/2025 at 4:18 AM. The 'Repro Steps' section details three steps: logging in, joining a group, and posting. The 'Planning' and 'Deployment' sections are visible on the right.

7. Test Case Results

The screenshot displays the 'Test Case Results' section of the TCOS application. It lists 10 test cases with their outcomes, timestamps, configurations, runners, testers, and test IDs. All tests have passed.

Outcome	TimeStamp	Configuration	Run by	Tester	Test ID
Passed	4m ago	Windows 10	Karthick S	Mallu karthick Balaji ...	Music
Passed	12m ago	Windows 10	Karthick S	Mallu karthick Balaji ...	Music
Not Applicable	12m ago	Windows 10	Karthick S	Mallu karthick Balaji ...	Music
Passed	14m ago	Windows 10	Karthick S	Mallu karthick Balaji ...	Music
Passed	Tuesday	Windows 10	Karthikeyan Senthil	Mallu karthick Balaji ...	Music
Passed	Saturday	Windows 10	Mallu karthick Balaji ...	Mallu karthick Balaji ...	Music
Failed	Saturday	Windows 10	Mallu karthick Balaji ...	Mallu karthick Balaji ...	Music
Passed	Apr 11	Windows 10	Karthick S	Mallu karthick Balaji ...	Music
Passed	Apr 11	Windows 10	Karthick S	Mallu karthick Balaji ...	Music

8. Progress Report



LOAD TESTING PROCEDURE :

Steps to Create an Azure Load Testing Resource:

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

1. Sign in to Azure Portal

Go to <https://portal.azure.com> and log in.

2. Create the Resource

- Go to Create a resource — Search for “Azure Load Testing”.
- Select Azure Load Testing and click Create.

3. Fill in the Configuration Details

- Subscription: Choose your Azure subscription.
- Resource Group: Create new or select an existing one.
- Name: Provide a unique name (no special characters).
- Location: Choose the region for hosting the resource.

4. (Optional) Configure tags for categorization and billing.

5. Click Review + Create, then Create.

6. Once deployment is complete, click Go to resource.

Steps to Create and Run a Load Test:

Once your resource is ready:

1. Go to your Azure Load Testing resource and click Add HTTP requests > Create.

2. Basics Tab

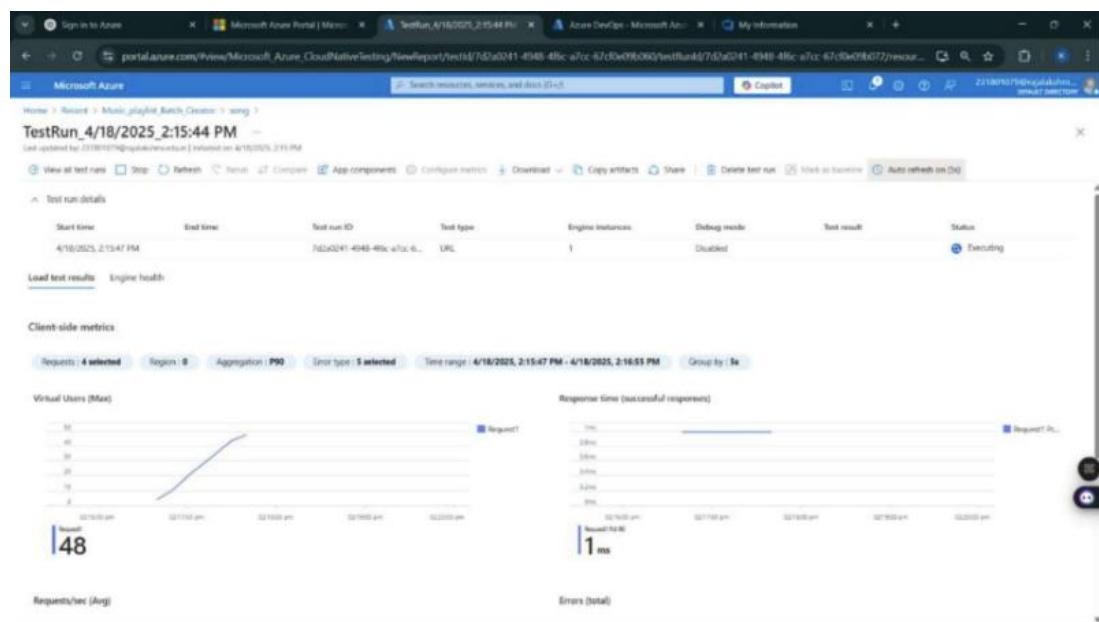
- Test Name: Provide a unique name.
- Description: (Optional) Add test purpose.
- Run After Creation: Keep checked.

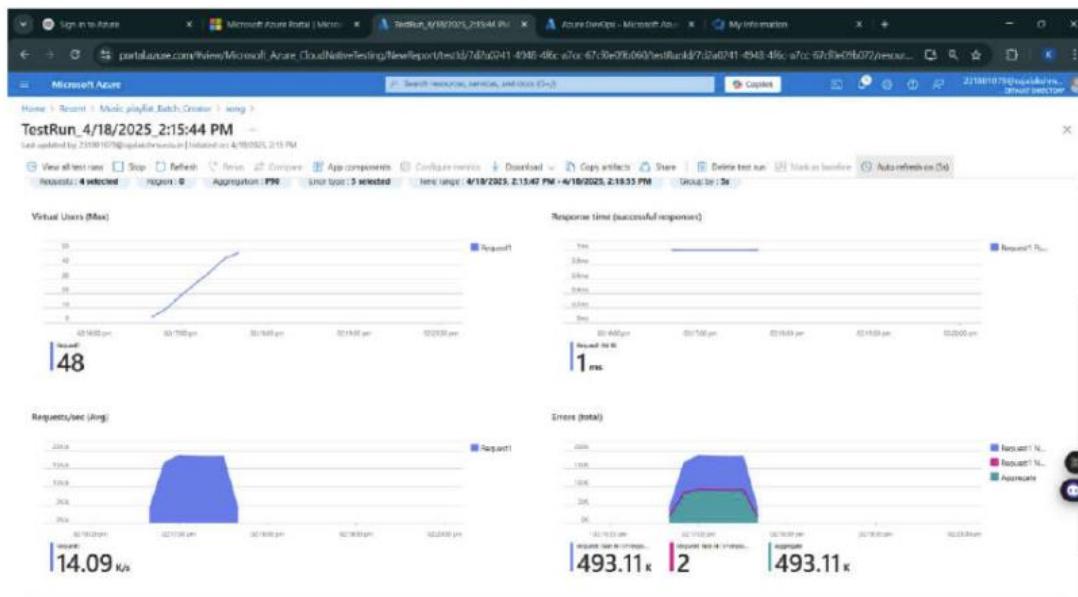
3. Load Settings

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

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

Load Testing





RESULT:

Test plans and test cases for selected user stories were created in Azure DevOps, covering both happy and error paths and an Azure Load Testing resource was also set up, and a load test was successfully run to evaluate the performance of the target endpoint.