

RAJALAKSHMI ENGINEERING COLLEGE
RAJALAKSHMI NAGAR, THANDALAM – 602 105



**RAJALAKSHMI
ENGINEERING COLLEGE**

**CS23432
SOFTWARE CONSTRUCTION**

Laboratory Record Note Book

NAME : SWETHA A

YEAR/BRANCH/SECTION : 2ND IT-AE

REGISTER NO : 231001226

SEMESTER : 4TH

ACADEMIC NUMBER : 2024-2025



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

BONAFIDE CERTIFICATE

NAME SWETHA A REGISTER NO. 231001226
ACADEMIC YEAR 2024-25 SEMESTER- IV BRANCH: B. Tech Information

Technology [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		

EXP NO:1

DATE :21-01-2025

Study of Azure DevOps

AIM:

To study how to create an agile project in Azure DevOps environment.

STUDY:

Azure DevOps is a cloud-based platform by Microsoft that provides tools for DevOps practices, including CI/CD pipelines, version control, agile planning, testing, and monitoring. It supports teams in automating software development and deployment.

1.Understanding Azure DevOps

Azure DevOps consists of five key services:

1.1 Azure Repos (Version Control)

Supports Git repositories and Team Foundation Version Control (TFVC).

Provides features like branching, pull requests, and code reviews.

1.2 Azure Pipelines (CI/CD)

Automates build, test, and deployment processes.

Supports multi-platform builds (Windows, Linux, macOS).

Works with Docker, Kubernetes, Terraform, and cloud providers (Azure, AWS, GCP).

1.3 Azure Boards (Agile Project Management)

Manages work using Kanban boards, Scrum boards, and dashboards.

Tracks user stories, tasks, bugs, sprints, and releases.

1.4 Azure Test Plans (Testing)

Provides manual, exploratory, and automated testing.

Supports test case management and tracking.

1.5 Azure Artifacts (Package Management)

Stores and manages NuGet, npm, Maven, and Python packages.

Enables versioning and secure access to dependencies.

Getting Started with Azure DevOps

Step 1: Create an Azure DevOps Account Visit

Azure DevOps.

Sign in with a Microsoft Account.

Create an Organization and a Project.

Step 2: Set Up a Repository (Azure Repos) Navigate

to Repos.

Choose Git or TFVC for version control.

Clone the repository and push your code.

Step 3: Configure a CI/CD Pipeline (Azure Pipelines) Go to

Pipelines → New Pipeline.

Select a source code repository (Azure Repos, GitHub, etc.). Define the pipeline using YAML or the Classic Editor.

Run the pipeline to build and deploy the application.

Step 4: Manage Work with Azure Boards Navigate to

Boards.

Create work items, user stories, and tasks.

Organize sprints and track progress.

Step 5: Implement Testing (Azure Test Plans) Go

to Test Plans.

Create and run test cases

View test results and track bugs.

Result:

The study was successfully completed.

EX NO:2**DATE:28-01-2025****PROBLEM STATEMENT****AIM:**

To prepare PROBLEM STATEMENT for your given project.

Problem Statement:**TO DO LIST APP WITH REMAINDERS**

In today's fast-paced digital world, individuals often struggle to manage their time efficiently and keep track of daily tasks, deadlines, and responsibilities. Despite the availability of various productivity tools, many users find existing solutions either too complex or lacking critical features such as reminders, recurring tasks, or cross-platform accessibility. This project aims to develop a user-friendly To-Do List Application with Reminder Functionality that helps users organize tasks, set priorities, and receive timely notifications. The application should allow users to create, update, delete, and categorize tasks. Additionally, users must be able to set reminders for upcoming tasks and deadlines, ensuring nothing important is missed. The app should provide an intuitive interface, support for push or email notifications, and data persistence using a backend (e.g., database or cloud storage). Optional features may include user authentication, dark mode, recurring tasks, and synchronization across devices.

Result:

The problem statement was written successfully.

EX NO: 3

DATE:04-02-2025

AGILE PLANNING

AIM:

To prepare an Agile Plan.

THEORY

Agile planning is a part of the Agile methodology, which is a project management style with an incremental, iterative approach. Instead of using an in-depth plan from the start of the project—which is typically product-related—Agile leaves room for requirement changes throughout and relies on constant feedback from end users. With Agile planning, a project is broken down into smaller, more manageable tasks with the ultimate goal of having a defined image of a project's vision. Agile planning involves looking at different aspects of a project's tasks and how they'll be achieved, for example:

- Roadmaps to guide a product's release ad schedule
- Sprints to work on one specific group of tasks at a time
- A feedback plan to allow teams to stay flexible and easily adapt to change

User stories, or the tasks in a project, capture user requirements from the end user's perspective. Essentially, with Agile planning, a team would decide on a set of user stories

to action at any given time, using them as a guide to implement new features or functionalities in a tool. Looking at tasks as user stories is a helpful way to imagine how a customer may use a feature and helps teams prioritize work and focus on delivering value first.

- Steps in Agile planning process

1. Define vision
2. Set clear expectations on goals
3. Define and break down the product roadmap
4. Create tasks based on user stories
5. Populate product backlog
6. Plan iterations and estimate effort
7. Conduct daily stand-ups
8. Monitor and adapt

Result:

Thus the Agile plan was completed successfully.

EX NO: 4

DATE:18-02-2025

CREATE USER STORIES WITH ACCEPTANCE CRITERIA

AIM:

To create User Stories

THEORY

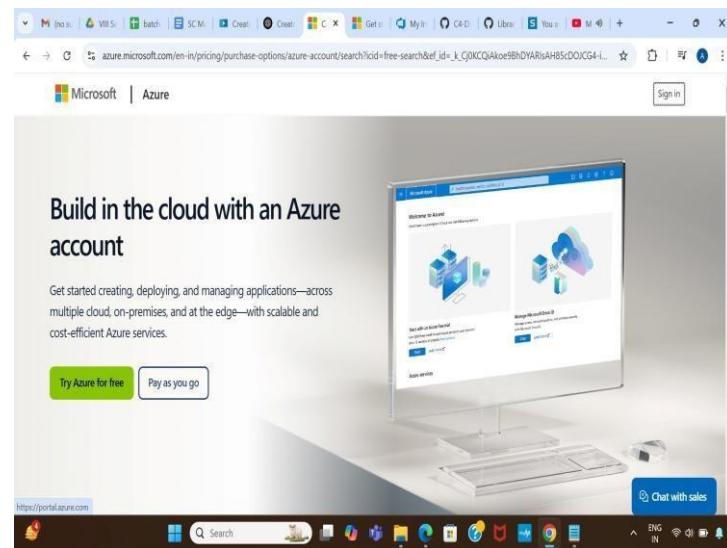
A user story is an informal, general explanation of a software feature written from the perspective of the end user. Its purpose is to articulate how a software feature will provide value to the customer.

User story template

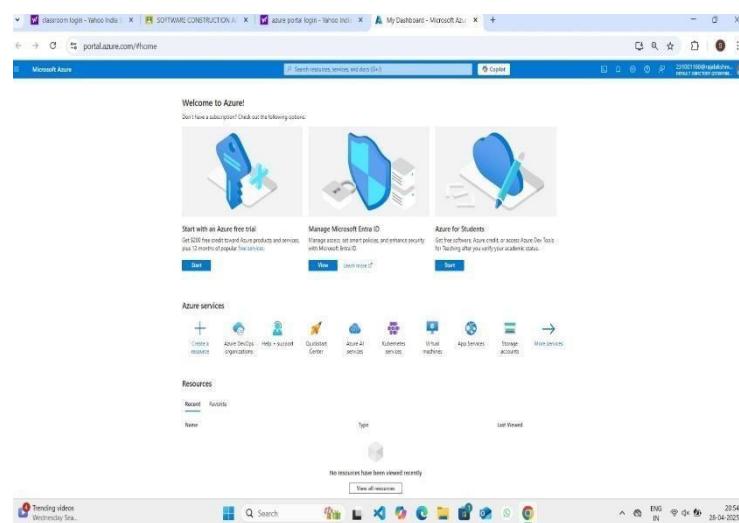
"As a [role], I [want to], [so that]."

PROCEDURE:

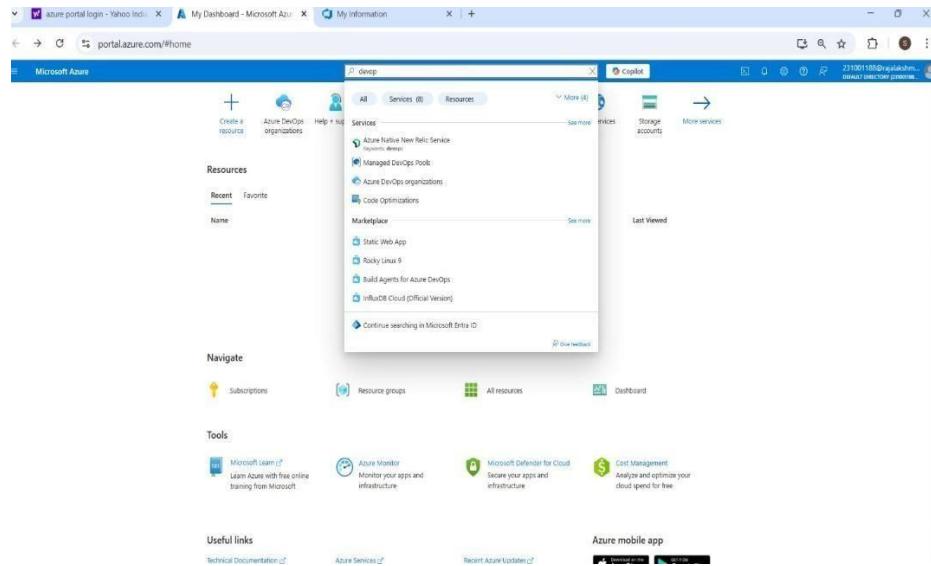
1. Open your web browser and go to the Azure website: <https://azure.microsoft.com/en-in> Sign in using your Microsoft account credentials. If you don't have an account, you'll need to create one.
2. If you don't have a Microsoft account, you can sign up for <https://signup.live.com/?lic=1>



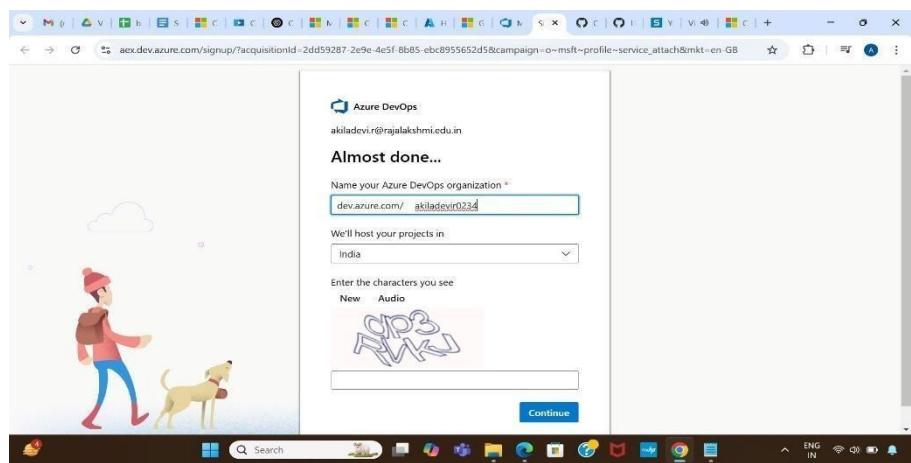
Azure home page



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



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



3. Create the First Project in Your Organization

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.

- i. On the organization's **Home page**, click on the **New Project** button.
- ii. Enter the project name, description, and visibility options:
 - o **Name:** Choose a name for the project (e.g., **LMS**).
 - o **Description:** Optionally, add a description to provide more context about the project.
 - o **Visibility:** Choose whether you want the project to be **Private** (accessible only to those invited) or **Public** (accessible to anyone).
- iii.

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

The screenshot shows the 'Create new project' dialog box. It includes fields for 'Project name *' and 'Description'. Under 'Visibility', there are two options: 'Public' (described as anyone on the internet can view the project) and 'Private' (described as only people you give access to will be able to view this project). The 'Private' option is selected and highlighted with a blue border. Below this, a note states 'Public projects are disabled for your organization. You can turn on public visibility with [organization policies](#)'. There are dropdowns for 'Version control' (set to 'Git') and 'Work item process' (set to 'Agile'). At the bottom are 'Cancel' and 'Create' buttons.

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 for the organization 'dev.azure.com/231001188 (Owner)'. On the left, there's a user profile for 'Shanthini C' with a large circular icon containing 'SC'. Below it are sections for 'Microsoft account' (India, 231001188@rajalakshmi.edu.in) and 'Visual Studio Dev Essentials' (Get everything you need to build and deploy your app on any platform, Use your benefits). On the right, the 'Azure DevOps Organizations' header has a 'Create new organization' button. Below it, under 'Actions', is an 'Open in Visual Studio' link. The main area lists three projects: 'Voiceflow', 'To Do List App With Reminders', and 'To Do List App With Reminders in Scrum'. A 'New project' link is also present.

4. Project dashboard

The screenshot shows the Azure DevOps interface for a project titled 'To Do List App With Reminders'. The left sidebar contains navigation links: Overview, Summary, Dashboards, Boards, Repos, Pipelines, Test Plans, and Artifacts. The main content area has a purple header bar with the project name. Below it, there's a section titled 'About this project' with a placeholder for a project description. To the right, there's a 'Project stats' card showing 'Boards' and 'Members'. The 'Boards' section displays two boards: 'Backend' and 'Frontend', both with zero work items. The 'Members' section shows five team members with their initials: SC, OB, SA, RS, and NK.

5. To manage user stories

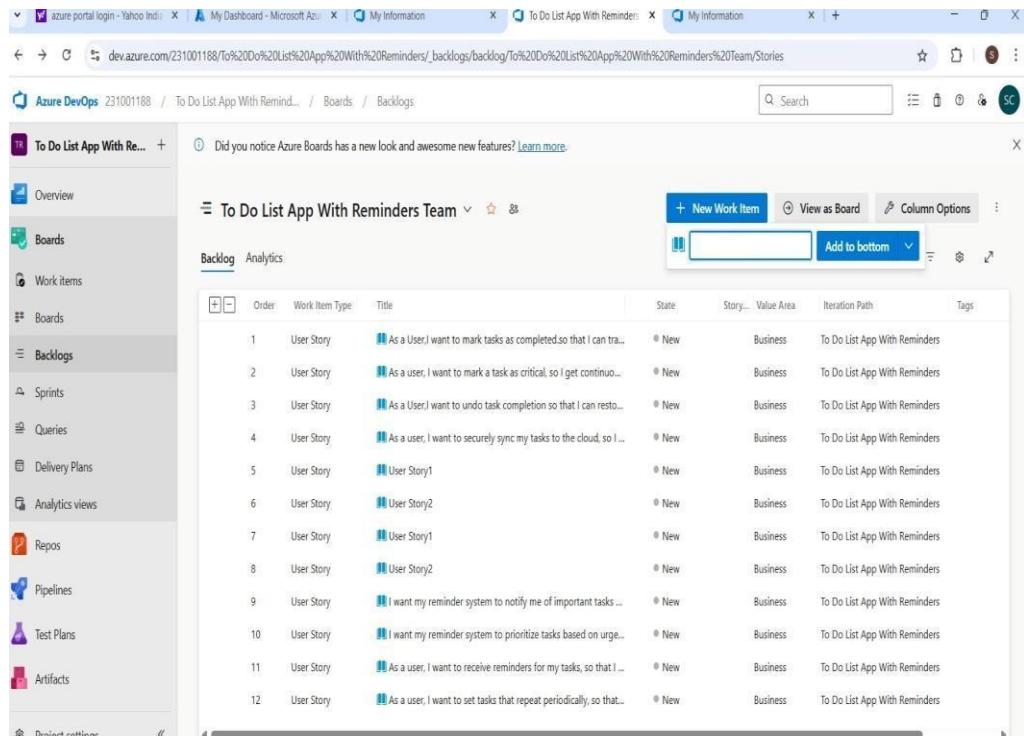
- a. From the **left-hand navigation menu**, click on **Boards**. This will take you to the main **Boards** page, where you can manage work items, backlogs, and sprints.
 - b. On the **work items** page, you'll see the option to **Add a work item** at the top. Alternatively, you can find a + button or **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.

The screenshot shows the Microsoft Azure Boards interface with the 'Work Items - Boards' tab selected. The left sidebar includes sections for Overview, Boards, Work Items, Boards, Sprints, Queries, Delivery Plans, Analytics views, Repos, Pipelines, Test Plans, and Artifacts. The 'Project settings' option is also visible at the bottom.

The main area displays a 'Work items' board titled 'Recently updated'. It lists the following items:

ID	Title	Type	Assigned To	Status	Area Path	Tags	Comments	Activity Date
28	Task [issue]	Bug	Unassigned	New	To Do List App With Reminders			25-03-2025 13:35:02
49	Test Case	User Story	Nithya K	New	To Do List App With Reminders			01-04-2025 13:04:24
31	c my tasks to the cloud, so that I can access them from anywhere	User Story	Gretha A	New	To Do List App With Reminders			25-03-2025 13:50:32
50	As a user, I want to set tasks that repeat periodically, so that I don't forget them	User Story	Nithya K	New	To Do List App With Reminders			01-04-2025 13:09:02
25	As a User I want to mark tasks as completed so that I can track my progress	User Story	Sharathini C	New	To Do List App With Reminders			25-03-2025 13:30:00
30	As a User I want to undo task completion so that I can restore tasks in progress	User Story	Sharathini C	New	To Do List App With Reminders			25-03-2025 13:36:30
18	Emergency Mode for Critical Tasks	User Story	Unassigned	New	To Do List App With Reminders			25-03-2025 13:26:44
32	Escalating alerts (pop-ups, sounds, vibrations) for ignored tasks.	User Story	Unassigned	New	To Do List App With Reminders			25-03-2025 14:33:35
44	I want my reminder system to notify me of important tasks when I am free	User Story	RAVILLA KAVYA SREE	New	To Do List App With Reminders			01-04-2025 12:29:14
45	I want my reminder system to prioritize tasks based on urgency and importance	User Story	RAVILLA KAVYA SREE	New	To Do List App With Reminders			01-04-2025 12:31:25
41	Notification Delivery	Notification	Oviya R	New	To Do List App With Reminders			31-03-2025 15:26:23
48	Reminders functionality	Notification	Nithya K	New	To Do List App With Reminders			01-04-2025 12:52:03
34	Reminder and Notification System	Notification	Oviya R	New	To Do List App With Reminders			31-03-2025 14:09:13
16	Secure Cloud sync & accessibility	Notification	Unassigned	New	To Do List App With Reminders			25-03-2025 13:28:01
46	Smart and Contextual Reminders	Notification	RAVILLA KAVYA SREE	New	To Do List App With Reminders			01-04-2025 12:33:22
37	Smart and Contextual Reminders	Notification	RAVILLA KAVYA SREE	New	To Do List App With Reminders			25-03-2025 13:30:44

6.Fill in User Story Details



The screenshot shows the Azure DevOps interface for a project titled "To Do List App With Reminders Team". The left sidebar has a "Backlogs" section selected. The main area displays a backlog of 12 User Stories. Each story includes a title, state (New), value area (Business), and iteration path (To Do List App With Reminders). The stories are numbered 1 through 12.

Order	Work Item Type	Title	State	Story...	Value Area	Iteration Path	Tags
1	User Story	[User Story 1]	New	Business	To Do List App With Reminders		
2	User Story	[User Story 2]	New	Business	To Do List App With Reminders		
3	User Story	[User Story 3]	New	Business	To Do List App With Reminders		
4	User Story	[User Story 4]	New	Business	To Do List App With Reminders		
5	User Story	[User Story 5]	New	Business	To Do List App With Reminders		
6	User Story	[User Story 6]	New	Business	To Do List App With Reminders		
7	User Story	[User Story 7]	New	Business	To Do List App With Reminders		
8	User Story	[User Story 8]	New	Business	To Do List App With Reminders		
9	User Story	[User Story 9]	New	Business	To Do List App With Reminders		
10	User Story	[User Story 10]	New	Business	To Do List App With Reminders		
11	User Story	[User Story 11]	New	Business	To Do List App With Reminders		
12	User Story	[User Story 12]	New	Business	To Do List App With Reminders		

Result:

The user story was written successfully.

EX NO: 5

DATE:25-02-2025

DESIGNING SEQUENCE DIAGRAMS USING AZURE DEVOPS-WIKI

AIM:

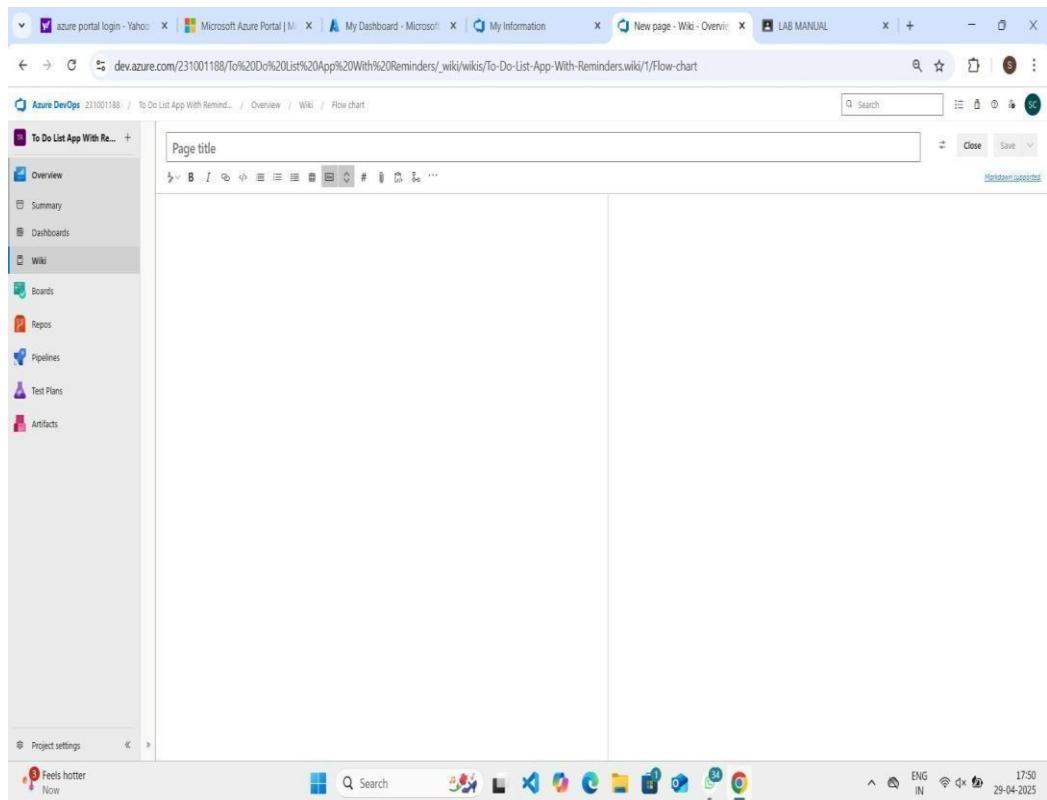
To design a Sequence Diagram by using Mermaid.js

THEORY:

A Sequence Diagram is a key component of Unified Modelling Language (UML) used to visualize the interaction between objects in a sequential order. It focuses on how objects communicate with each other over time, making it an essential tool for modelling dynamic behaviour in a system.

PROCEDURE:

1. Open a project in Azure DevOps Organisations.
2. To design select wiki from menu



3. Write code for drawing sequence diagram and save the code.

```
:::mermaid sequenceDiagram
    participant User
    participant App
    participant Server
    participant Database
```

User->>App: Add Task (title, description, reminder time)

App->>Server: Send Task Data (title, description, reminder time)

Server->>Database: Store Task in DB

Database-->>Server: Confirmation (task saved)

Server-->>App: Task Added Successfully
 App-->>User: Show Task Added

User->>App: Set Reminder

App->>Server: Send Reminder Time

Server->>Database: Store Reminder Time for Task

Database-->>Server: Reminder Stored
Server-->>App: Reminder Set App--
>>User: Show Reminder Set

Note right of Server: At Reminder Time:
\nServer checks for pending reminders
Server->>App: Notify Task Reminder App-
>>User: Push Notification
:::

Explanation:

participant defines the entities involved.

->> represents a direct message. -->>

represents a response message. +

after ->> activates a participant. -

after -->> deactivates a participant.

alt / else for conditional flows. loop

can be used for repeated actions.

-> Solid line without arrow

--> Dotted line without arrow

->> Solid line with arrowhead

-->> Dotted line with arrowhead

<<->> Solid line with bidirectional arrowheads (v11.0.0+)

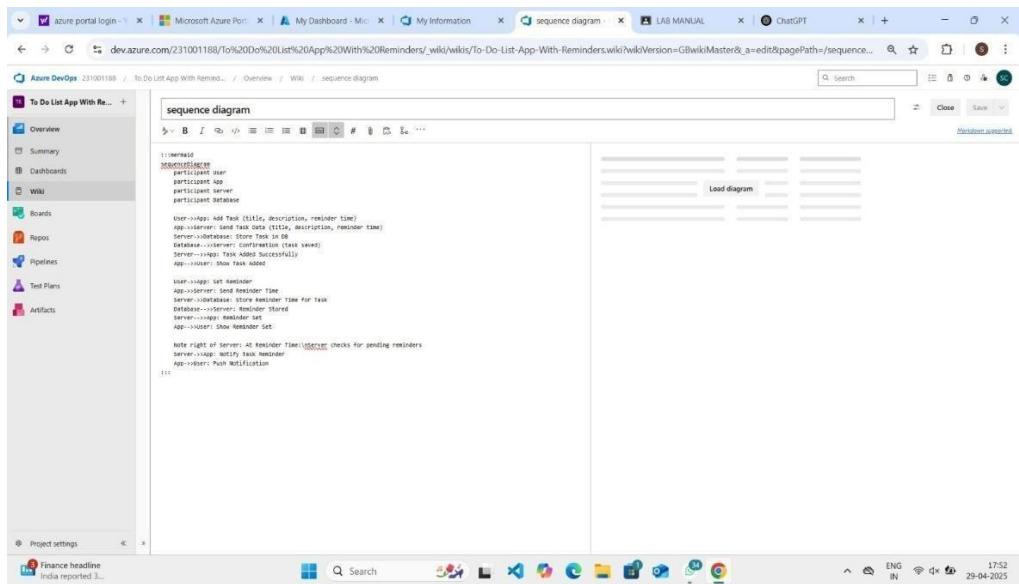
<<->> Dotted line with bidirectional arrowheads (v11.0.0+)

-x Solid line with a cross at the end

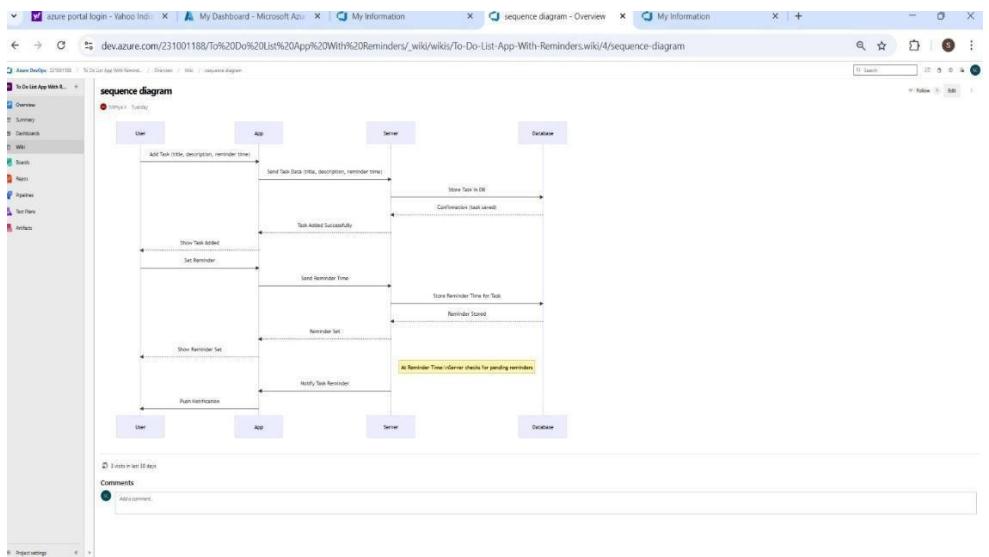
--x Dotted line with a cross at the end

-) Solid line with an open arrow at the end (async)

--) Dotted line with an open arrow at the end (async)



4. click wiki menu and select the page



Result:

The sequence diagram was drawn successfully.

EX NO. 6

DATE:04-03-2025

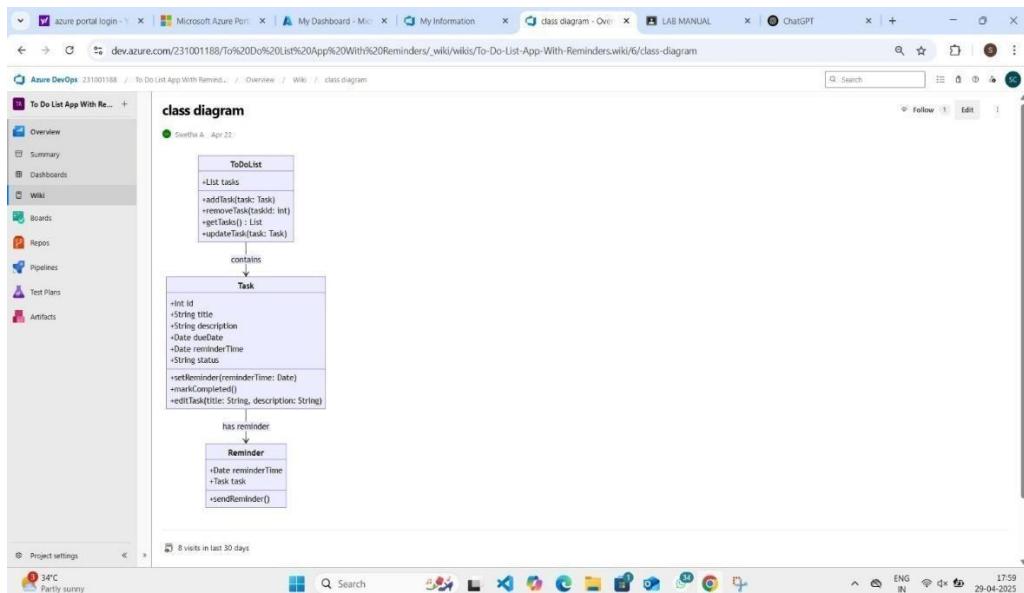
DESIGNING CLASS DIAGRAM USING AZURE DEVOPS-WIKI

AIM :-

To draw a sample class diagram for your project or system.

THEORY

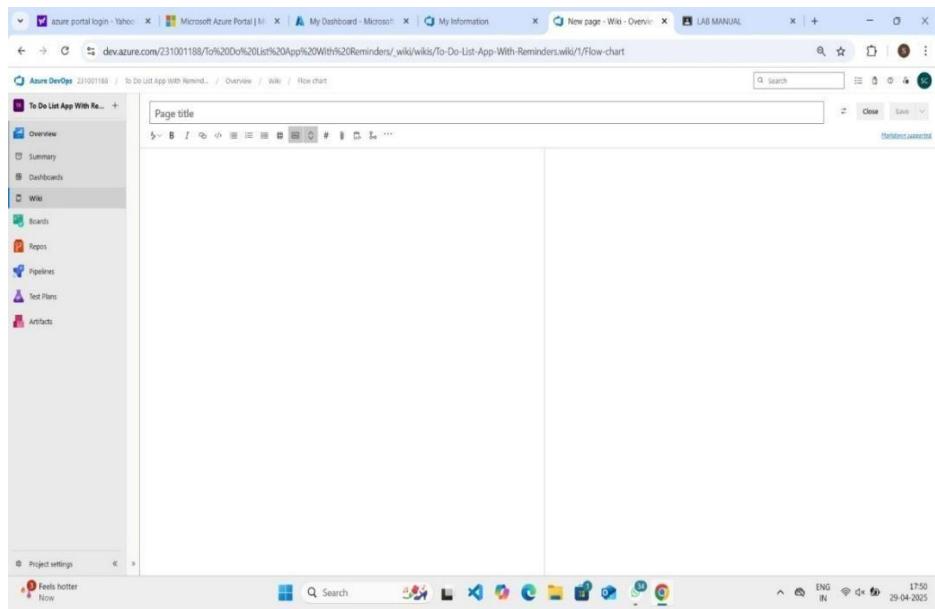
A UML class diagram is a visual tool that represents the structure of a system by showing its classes, attributes, methods, and the relationships between them.



Notations in class diagram

PROCEDURE:

1. Open a project in Azure DevOps Organisations.
2. To design select wiki from menu



3. Write code for drawing class diagram and save the code

```
:::mermaid classDiagram
class Task {
    +int id
    +String title
    +String description
    +Date dueDate
    +Date reminderTime
    +String status
    +setReminder(reminderTime: Date)
    +markCompleted()
    +editTask(title: String, description: String)
}

class Reminder {
    +Date reminderTime
    +Task task
    +sendReminder()
}

class ToDoList {
    +List<Task> tasks
    +addTask(task: Task)
    +removeTask(taskId: int)
    +getTasks(): List<Task>
```

```

+updateTask(task: Task)
}

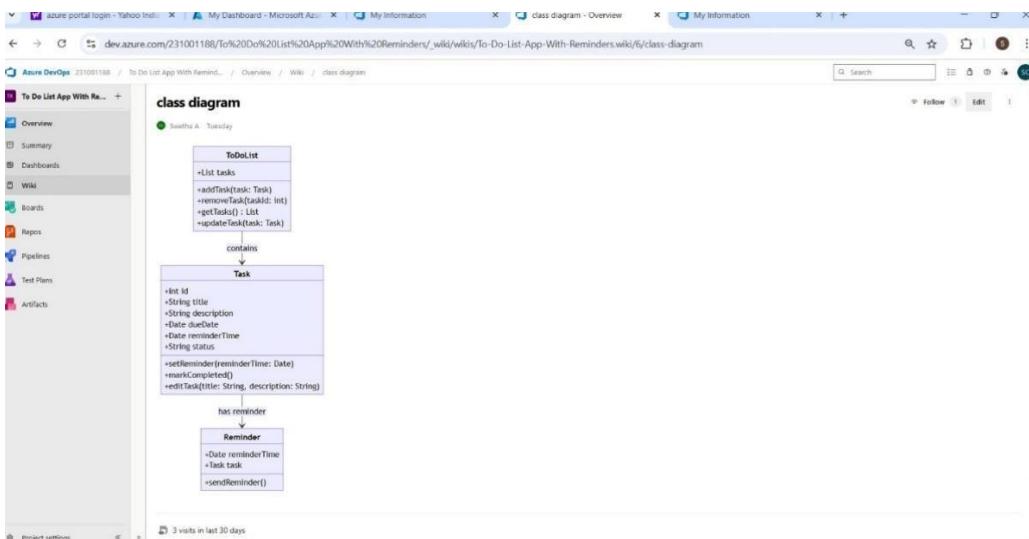
Task --> Reminder : has reminder
ToDoList --> Task : contains
:::

```

Relationship Types

Type Description

<	Inheritance
*	Composition
o	Aggregation
>	Association
<	Association
>	Realization



Visit : <https://mermaid.js.org/syntax/classDiagram.html>

Result:

The use case diagram was designed successfully

EX NO: 7**DATE:11-03-2025**

DESIGNING USECASE DIAGRAM USING AZURE DEVOPS-WIKI

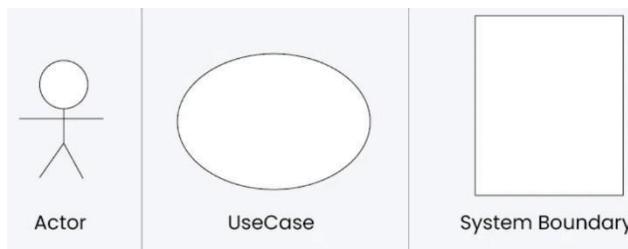
AIM:

Steps to draw the Use Case Diagram using draw.io

THEORY:

- UCD shows the relationships among actors and use cases within a system which Provide an overview of all or part of the usage requirements for a system or organization in the form of an essential model or a business model and communicate the scope of a development project

- **Use Cases**
- **Actors**
- **Relationships**
- **System Boundary Boxes**



PROCEDURE :

Step 1: Create the Use Case Diagram in Draw.io

- Open Draw.io ([diagrams.net](#)).
- Click "Create New Diagram" and select "Blank" or "UML Use Case" template.
- Add Actors (Users, Admins, External Systems) from the UML section.
- Add Use Cases (Functionalities) using ellipses.
- Connect Actors to Use Cases with lines (solid for direct interaction, dashed for <>include<> and <>extend<>).
- Save the diagram as .drawio or export as PNG/JPG/SVG.

Step 2: Upload the Diagram to Azure DevOps

Option 1: Add to Azure DevOps Wiki • Open Azure DevOps and go to your project.

- Navigate to Wiki (Project > Wiki).
- Click "Edit Page" or create a new page.
- Drag & Drop the exported PNG/JPG image.
- Use Markdown to embed the diagram:
- ! [Use Case Diagram](attachments/use_case_diagram.png)

Option 2: Attach to Work Items in Azure Boards

- Open Azure DevOps → Navigate to Boards (Project > Boards).
- Select a User Story, Task, or Feature.
- Click "Attachments" → Upload your Use Case Diagram.
- Add comments or descriptions to explain the use case.

Result:

The use case diagram was designed successfully

EX NO. 8

DATE:18-03-2025

DESIGNING ACTIVITY DIAGRAM USING AZURE DEVOPS-WIKI

AIM :-

To draw a sample activity diagram for your project or system.

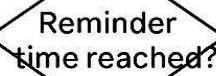
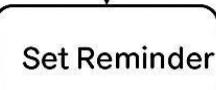
THEORY

Notations	Symbol	Meaning
Start		Shows the beginning of a process
Connector		Shows the directional flow, or control flow, of the activity
Joint symbol		Combines two concurrent activities and reintroduces them to a flow where one activity occurs at a time
Decision		Represents a decision
Note		Allows the diagram creators to communicate additional messages
Send signal		Show that a signal is being sent to a receiving activity
Receive signal		Demonstrates the acceptance of an event
Flow final symbol		Represents the end of a specific process flow
Option loop		Allows the creator to model a repetitive sequence within the option loop symbol
Shallow history pseudostate		Represents a transition that invokes the last active state.
End		Marks the end state of an activity and represents the completion of all flows of a process

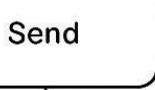
Activity diagrams are an essential part of the Unified Modelling Language (UML) that help visualize workflows, processes, or activities within a system. They depict how different actions are connected and how a system moves from one state to another.

PROCEDURE:

1. Draw diagram in draw.io
2. Upload the diagram in Azure DevOps wiki



Yes



Result:

The activity diagram was designed successfully

EX NO. 9

DATE:25-03-2025

DESIGNING ARCHITECTURE DIAGRAM USING STAR UML

AIM:

Steps to draw the Architecture Diagram using draw.io.

THEORY:

An architectural diagram is a visual representation that maps out the physical implementation for components of a software system. It shows the general structure of the software system and the associations, limitations, and boundaries between each element.



PROCEDURE:

1. Draw diagram in draw.io
2. Upload the diagram in Azure DevOps wiki

EX NO. 10

Result:

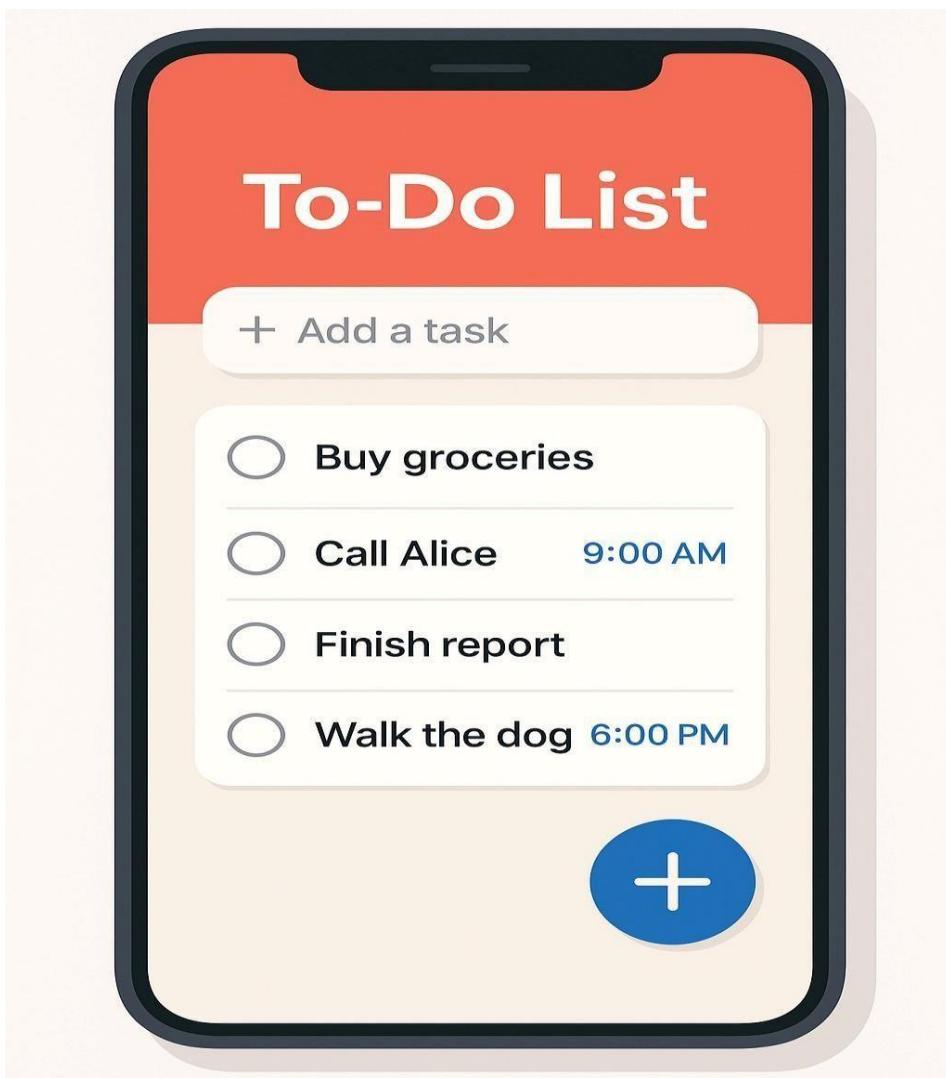
The architecture diagram was designed successfully

DATE:01-04-2025

DESIGN USER INTERFACE

AIM:

Design User Interface for the given project



EX NO. 11

Result:

The UI was designed successfully.

DATE:08-04-2025

IMPLEMENTATION-DESIGN A WEB PAGE BASED ON SCRUM METHODOLOGY

AIM:

To implement the given project based on Agile Methodology.

PROCEDURE:

Step 1: Set Up an Azure DevOps Project • Log in to Azure DevOps.

- Click "New Project" → Enter project name → Click "Create".
- Inside the project, navigate to "Repos" to store the code.

Step 2: Add Your Web Application Code

- Navigate to Repos → Click "Clone" to get the Git URL. • Open Visual Studio Code / Terminal and run: `git clone <repo_url> cd <repo_folder>`
- Add web application code (HTML, CSS, JavaScript, React, Angular, or backend like Node.js, .NET, Python, etc.). • Commit & push: `git add . git commit -m "Initial commit" git push origin main`

Step 3: Set Up Build Pipeline (CI/CD - Continuous Integration) • Navigate to Pipelines → Click "New Pipeline".

- Select Git Repository (Azure Repos, GitHub, or Bitbucket).
- Choose Starter Pipeline or a pre-configured template for your framework.
- Modify the `azure-pipelines.yml` file (Example for a Node.js app):
`trigger: - main`

```
pool:  
  vmImage: 'ubuntu-latest'
```

EX NO. 12

steps:

- task: UseNode@1 inputs:
version: '16.x'
- script: npm install displayName: 'Install dependencies'
- script: npm run build displayName:
'Build application'
- task: PublishBuildArtifacts@1 inputs:
pathToPublish:
'dist' artifactName: 'drop'

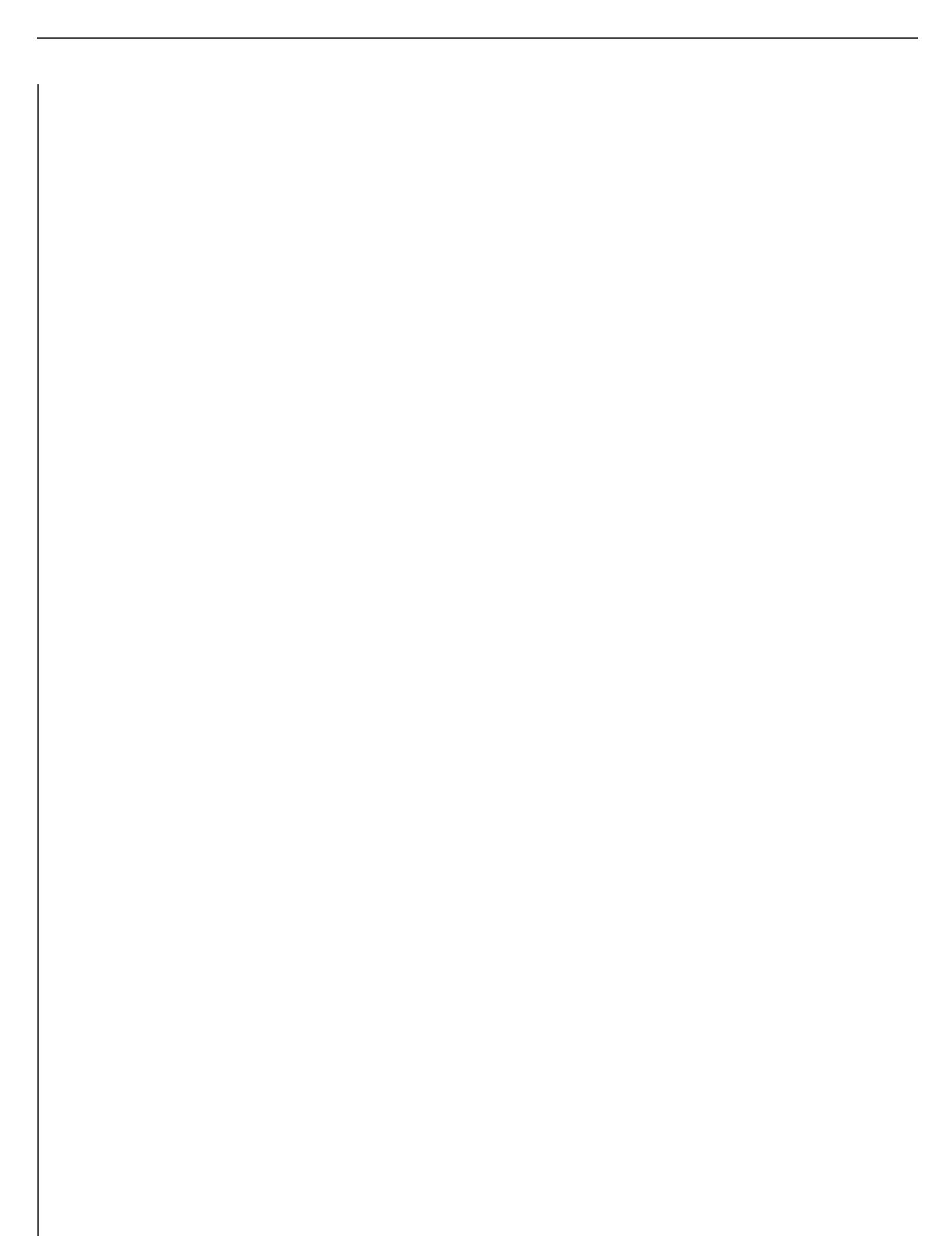
Click "Save and Run" → The pipeline will start building app.

Step 4: Set Up Release Pipeline (CD - Continuous Deployment) • Go to Releases → Click "New Release Pipeline".

- Select Azure App Service or Virtual Machines (VMs) for deployment.
- Add an artifact (from the build pipeline).
- Configure deployment stages (Dev, QA, Production).
- Click "Deploy" to push your web app to Azure.

Result :

Thus the application was successfully implemented.



EX NO:12

DATE:15-04-2025

Testing-Test plan,Test case

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

Azure DevOps Test Plan creation interface:

- Name: Music Playlist Batch Creator - Test Plan
- Area Path: Music Playlist Batch Creator
- Iteration: Music Playlist Batch Creator\Integration

2. Test suite

Title	Order	Test Case ID	Assigned To	State
101 - Successful Sign Up	1	70	Karthikeyan Se... Design	
Static suite	2	80	Karthikeyan Se... Design	
Requirement based suite	3	81	Karthikeyan Se... Design	
Query based suite	4	82	Karthikeyan Se... Design	

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

- 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 a Microsoft Azure DevOps Test Plan interface. The test case is titled "TC06 - Playlist Loading Failure". It has two steps defined:

- Step 1: "Disconnect from internet" with the expected result "Network is offline".
- Step 2: "Navigate to 'My Playlists'" with the expected result "Error message 'Unable to load playlists' is shown".

The status of the test case is set to "Custom" with a priority of 2 and is marked as "Not Automated".

The screenshot shows a Microsoft Azure DevOps Test Plan interface. A specific test case, 'TC05 – View Playlist Page', is selected. The test case details include:

- Author:** Karthick S
- Comments:** 0
- Tags:** Add Tag
- Status:** Design
- Reason:** None
- Description:** Music Playlist Batch Creator/Integration

The **Steps** section contains two steps:

1. Log in successfully. Expected result: User is redirected to dashboard.
2. Navigate to "My Playlists" section. Click or type here to add a step.

The **Custom** tab is selected, showing the following details:

- Type:** Happy Path
- Status:** Priority: 2, Automation status: Not Automated

At the bottom, there are buttons for **Save and Close**, **Follow**, and other navigation options.

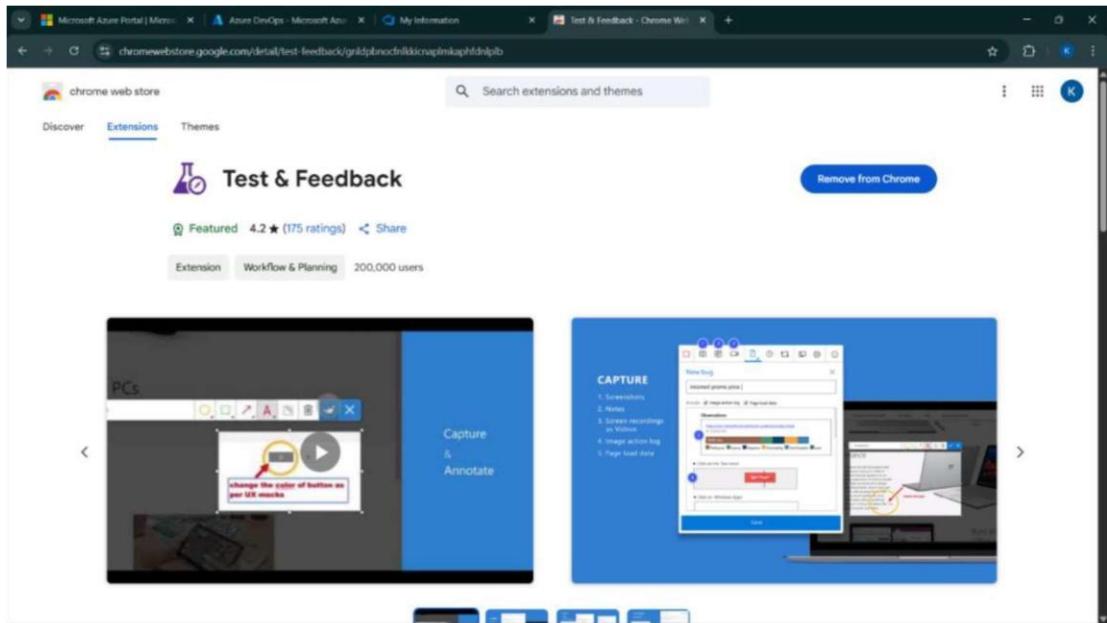
4. Installation of test

The screenshot shows the Chrome Web Store page for the 'Test & Feedback' extension. Key details include:

- Rating:** 4.2 ★ (175 ratings)
- Users:** 200,000 users
- Category:** Extension, Workflow & Planning
- Developer:** chrome web store
- Install button:** Add to Chrome

The extension's interface is demonstrated in two screenshots:

- PCs:** Shows a screenshot of a browser window with a red annotation pointing to a button, labeled 'Change the color of button as per UX needs'.
- Capture & Annotate:** Shows a screenshot of a recording interface with various tools and a preview window.



Test and feedback

Showing it as an extension

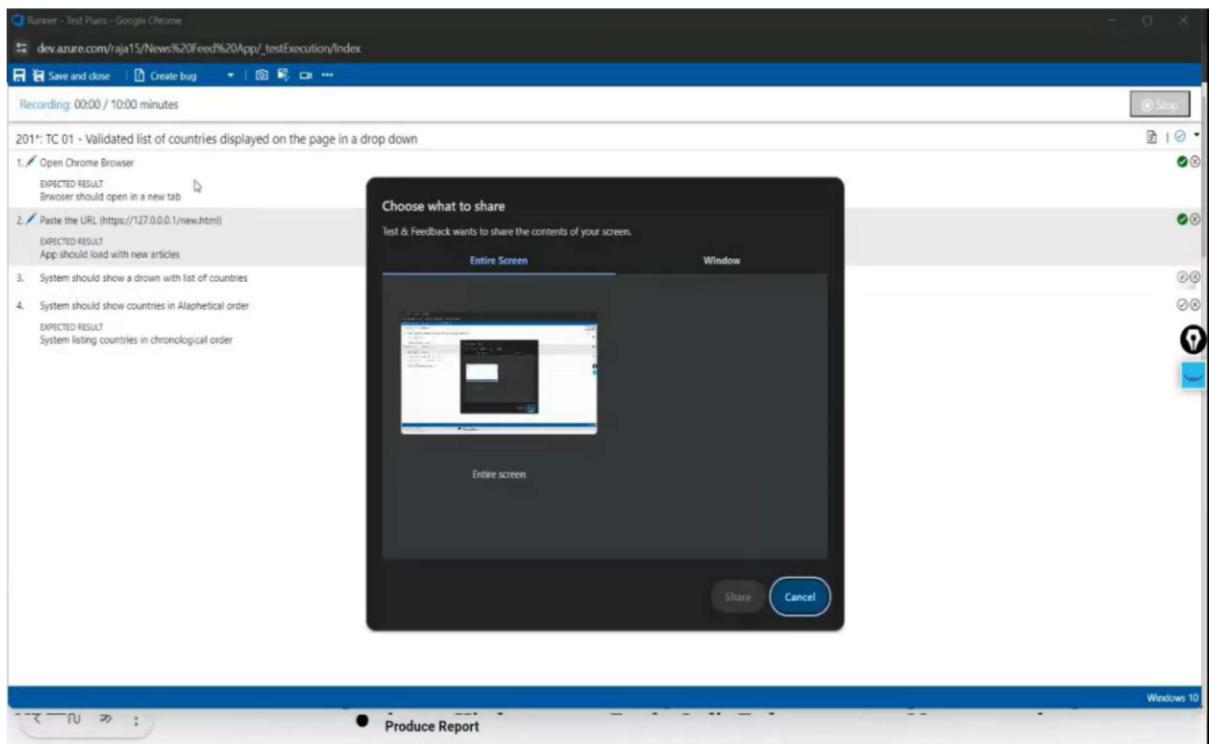
A screenshot of the Azure DevOps interface, specifically the Test Plan section. On the left, the navigation menu includes "Overview", "Boards", "Repos", "Pipelines", "Test Plans", "Progress report", "Parameters", "Configurations", "Runs", and "Artifacts". The "Test Plans" item is currently selected. In the center, a test plan named "TS01 - User Login (ID: 86)" is shown with four test cases: "Title", "TC01 - Successful Sign Up", "TC02 - Secure Login", "TC03 - Sign Up with Existing Email", and "TC04 - Login with Wrong Password". On the right, a floating "Extensions" pane is open, listing installed extensions: "Copy Text from Picture", "Dark Reader", "Monica: ChatGPT AI Assist...", and "Test & Feedback". The "Test & Feedback" extension is highlighted with a blue border. A tooltip for this extension states: "Selected: Copy Text from V...". At the bottom right of the pane, there is a "Manage extensions" button.

5. Running the test cases

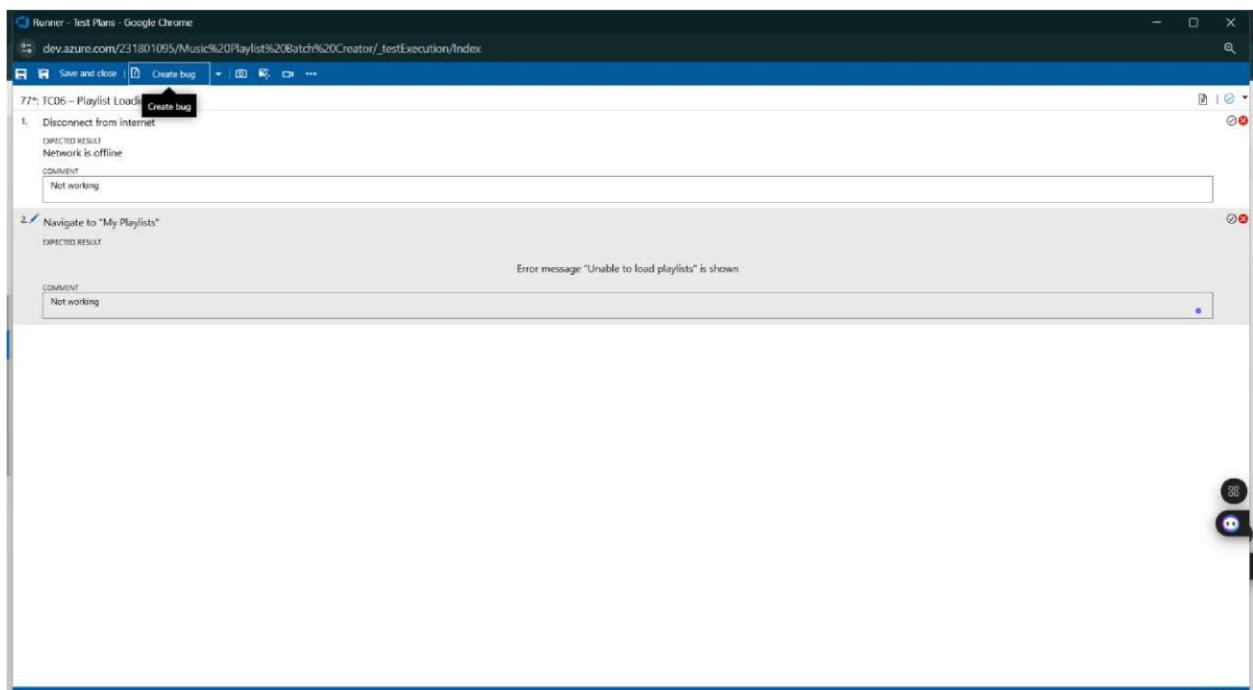
The screenshot shows the Azure DevOps Test Plan interface. On the left, the navigation sidebar is visible with options like Overview, Boards, Repos, Pipelines, Test Plans, and Artifacts. Under Test Plans, 'Test plans' is selected, and 'Music Playlist Batch Creator - T...' is expanded, showing 'TS01 - User Login (4)' and 'TS02 - View Playlists (2)'. 'TS02 - View Playlists (2)' is currently selected. The main area displays 'TS02 - View Playlists (ID: 87)'. It has tabs for Define, Execute, and Chart. The Execute tab is active, showing 'Test Points (2 items)'. Two test points are listed: 'TC05 - View Playlist Page' (Passed, Order 1, Test Case Id 75, Windows 10, Tester Mallu karthick B...) and 'TC06 - Playlist Loading Failure' (Not run). A context menu is open over 'TC05 - View Playlist Page', with options including 'Run for web application', 'Run for desktop application', and 'Run with options'. The status bar at the bottom indicates 'Runner - Test Plans - Google Chrome'.

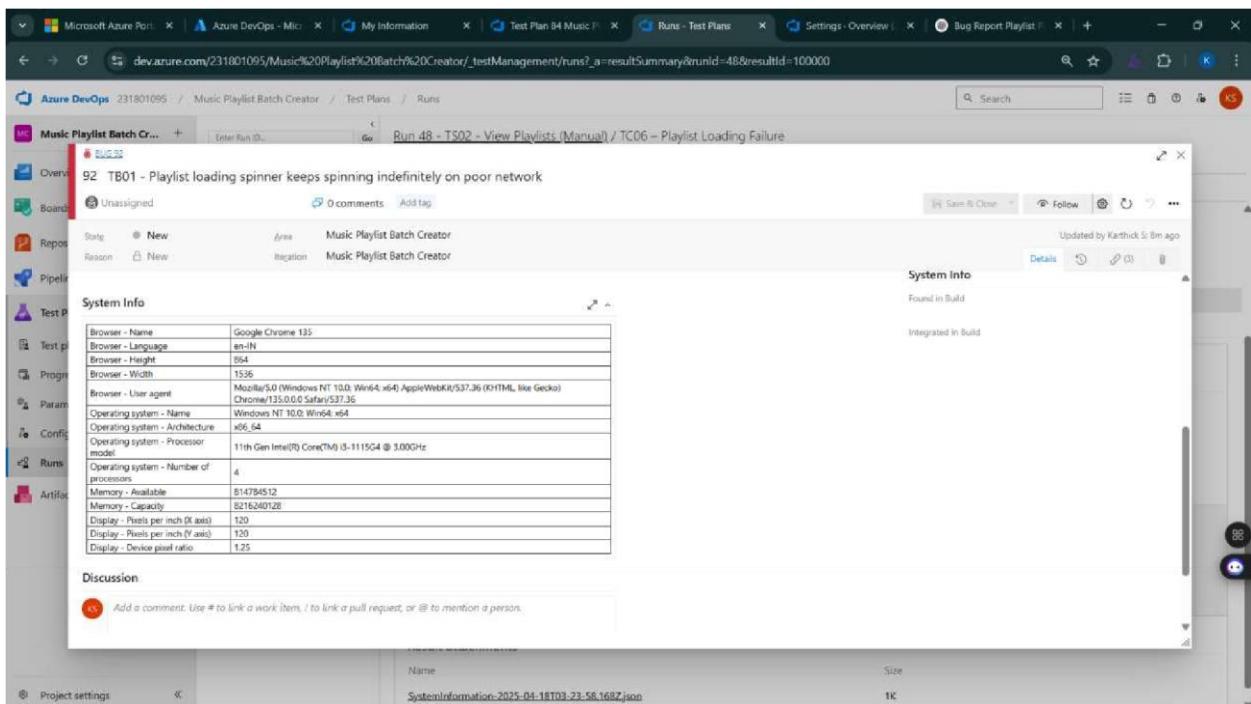
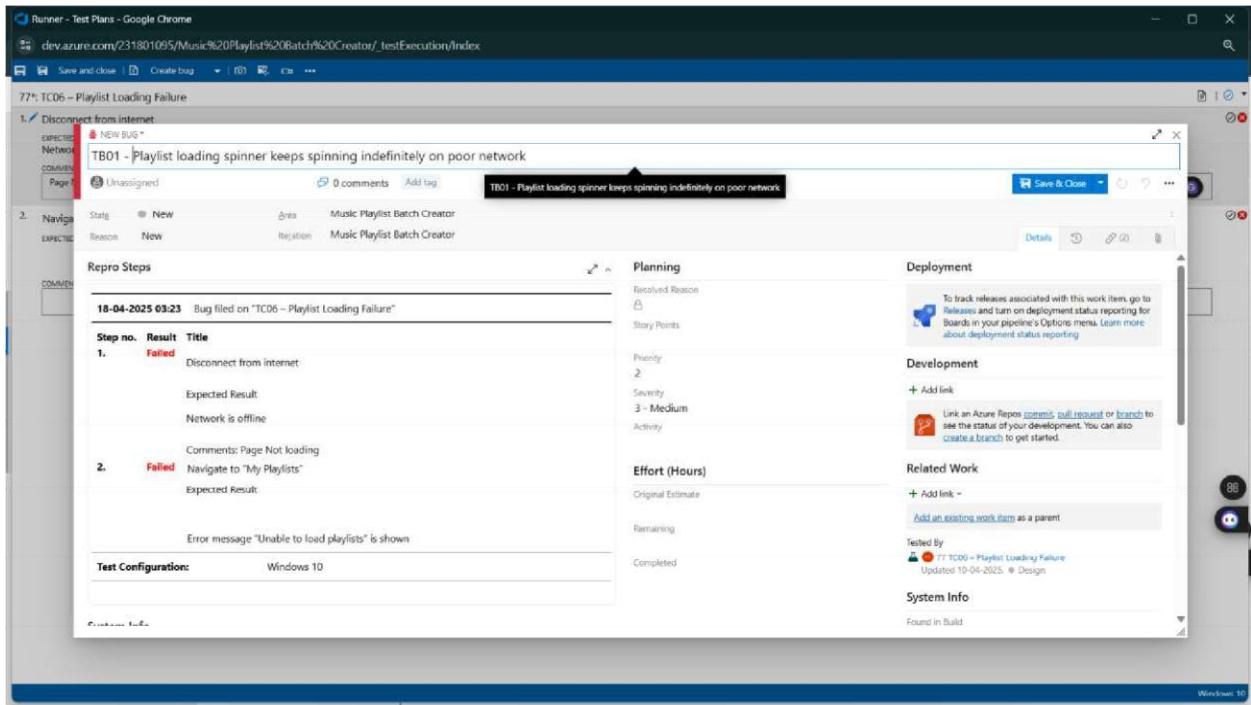
The screenshot shows the browser window titled 'Runner - Test Plans - Google Chrome' with the URL 'dev.azure.com/231801095/Music%20Playlist%20Batch%20Creator/_testExecution/Index'. The page displays the results for 'TC05 - View Playlist Page'. Step 1, 'Log in successfully', is marked as 'EXPECTED RESULT' with the note 'User is redirected to dashboard'. Step 2, 'Navigate to "My Playlists" section', is marked as 'EXPECTED RESULT' with the note 'All created playlists are displayed clearly'. Both steps are marked as passed with green checkmarks.

6.Recording the test case



7.Creating the bug





8. Test case results

The screenshot shows the Azure DevOps interface for a project named "Music Playlist Batch Creator". The left sidebar navigation includes "Overview", "Boards", "Repos", "Pipelines", "Test Plans", "Test plans", "Progress report", "Parameters", "Configurations", "Runs", and "Artifacts". Under "Test plans", "TS02 - View Playlists (2)" is selected. The main content area displays the "TS02 - View Playlists (ID: 87)" test plan. It shows two test points: "TC05 - View Playlist Page" (Passed) and "TC06 - Playlist Loading Failure" (Not Applicable). A detailed view of "TC05 - View Playlist Page" is shown on the right, titled "TC05 – View Playlist Page". The "Test Case Results" section lists 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	Karthick S	Malu karthick Balaji ...	Music
Passed	Saturday	Windows 10	Malu karthick Balaji ...	Malu karthick Balaji ...	Music
Failed	Saturday	Windows 10	Malu karthick Balaji ...	Malu karthick Balaji ...	Music
Passed	April 11	Windows 10	Karthick S	Malu karthick Balaji ...	Music
Passed	April 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 navigation includes "Overview", "Boards", "Work items", "Backlogs", "Sprints", "Queries", "Delivery Plans", "Analytics views", "Repos", "Pipelines", "Test Plans", and "Artifacts". The "Work items" section is active, displaying a work item titled "BUG 203: BG 01 - Countries Drop down Not Available on the page". The work item details include:

- State:** New
- Reason:** New
- Repro Step:** Active
- Resolved:** Resolved
- Step no.**: 1. **Result:** Passed. **Title:** Open Chrome Browser. **Expected Result:** Browser should open in a new tab.
- Step no.**: 2. **Result:** Passed. **Title:** Paste the URL, (<https://127.0.0.1/news.html>). **Expected Result:** App should load with new articles.
- Step no.**: 3. **Result:** Failed. **Title:** System should show a dropdown with list of countries. **Comments:** System should show a dropdown with list of countries.

The "Planning" section shows the work item is assigned to "rajesh prabhu" and has a priority of 2. The "Deployment" section provides information on releases and deployment status. The "Development" section links to Azure Repos for committing code. The "Effort (Hours)" section shows an original estimate of 0 hours.

- Assigning bug to the developer and changing state

Run 48 - TS02 - View Playlists (Manual) / TC06 - Playlist Loading Failure

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

Repro Steps:

- 1. Failed Disconnect from internet
- 2. Failed 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: None
- Story Points: 2
- Priority: 3 - Medium
- Severity: 3 - Medium
- Activity: None

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:

- + Add link: Add an existing work item as a parent.

System Info:

- Tested By: KARTHICK-231801079/Music
- Last updated: 10-04-2025

10. Progress report

Progress report

Summary:

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

Outcome trend:

Music Playlist Batch Creator - Test Plan

Last 14 Days

Legend: Not run (Grey), Passed (Green)

Test point	Run %	Passed %	Failed %	Not run count
TS01 - User Login	100	100	0	0
TS02 - View Playlist	100	100	0	0
TS03 - Real-Time Metadata	100	100	0	0
TS04 - Playlist Editing	100	100	0	0
TS05 - Smart Playlist Creation	100	100	0	0

11. Changing the test template

The screenshot shows the Azure DevOps Settings - Process page. The left sidebar is collapsed, and the main area displays a list of process templates:

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

At the bottom, there is a summary card showing a 100% pass rate for 28 tests, all of which are passed. Below this is a timeline chart from April 4 to April 18, 2021, showing the status of various test runs. A table at the bottom provides detailed test plan statistics.

Test plan name	Test points	Run %	Passed %	Failed %	Not run count
Music Playlist Batch Creator Team_Stories_Integration	14	100	100	0	0
Music Playlist Batch Creator – Test Plan	14	100	100	0	0

The screenshot shows the 'All processes' page in the Azure DevOps Settings. The left sidebar is collapsed, and the main area displays a list of process templates. The 'Agile' template is selected, indicated by a blue border around its name. The 'Basic' template is also visible. Other templates like 'Scrum' and 'CMMI' are listed below.

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

The screenshot shows the same 'All processes' page as the first one, but with a specific change. The 'Agile' template has been renamed to '231801095 Agile (default)', as indicated by the bolded text in the 'Name' column. The other templates ('Basic', 'Scrum', 'CMMI') remain unchanged.

Name	Description	Team projects
Basic	This template is flexible for any process and great for teams getting started with Azure DevOps.	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	This template is for more formal projects requiring a framework for process improvement and an auditable reco...	0

12. View the new test case template

The screenshot shows the 'Add a field to Test Case' dialog box overlaid on the Azure DevOps settings interface. The dialog has a tabbed interface with 'Definition' selected, showing options for adding a new field or using an existing one. The 'Create a field' option is selected, and a new field named 'Acceptance Criteria' is being defined as a 'Text (single line)' type. A description is provided: 'Optionally provide a description for the field'. At the bottom are 'Add Field' and 'Cancel' buttons.

The screenshot shows the 'Process' settings page in Azure DevOps. The 'Projects' tab is selected under the 'Work item types' section. A new work item type named 'Music Playlist Batch Creator' is listed, with a detailed description: 'The Azure Music Playlist Batch Creator is a cloud-based solution designed for bulk playlist creation and management. Leverage...'. The left sidebar shows the 'Process' section is currently selected.

The screenshot shows the Azure DevOps Settings - Process page for a specific organization setting (231801095). The page title is "All processes > 231801095 Agile > Test Case". The left sidebar includes sections for General, Security, Boards, Pipelines, and Process (which is currently selected). The main content area displays the "Test Case" configuration, which includes fields for "Steps" (Text [multiple lines]), "Custom" (Type [Text (single line)]), "Recent test results" (Recent test case results), "Deployment" (Deployments), "Development" (Links), "Related Work" (Links), and "Status" (Priority [Integer] and Automation status).

Result:

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

Testing-Load Testing

Aim:

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

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

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

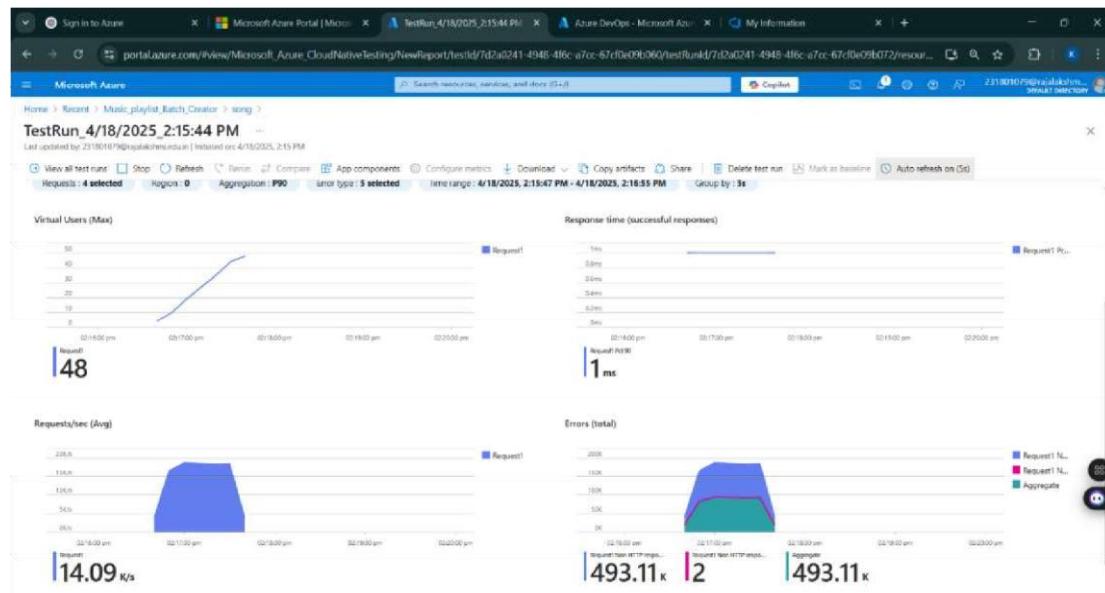
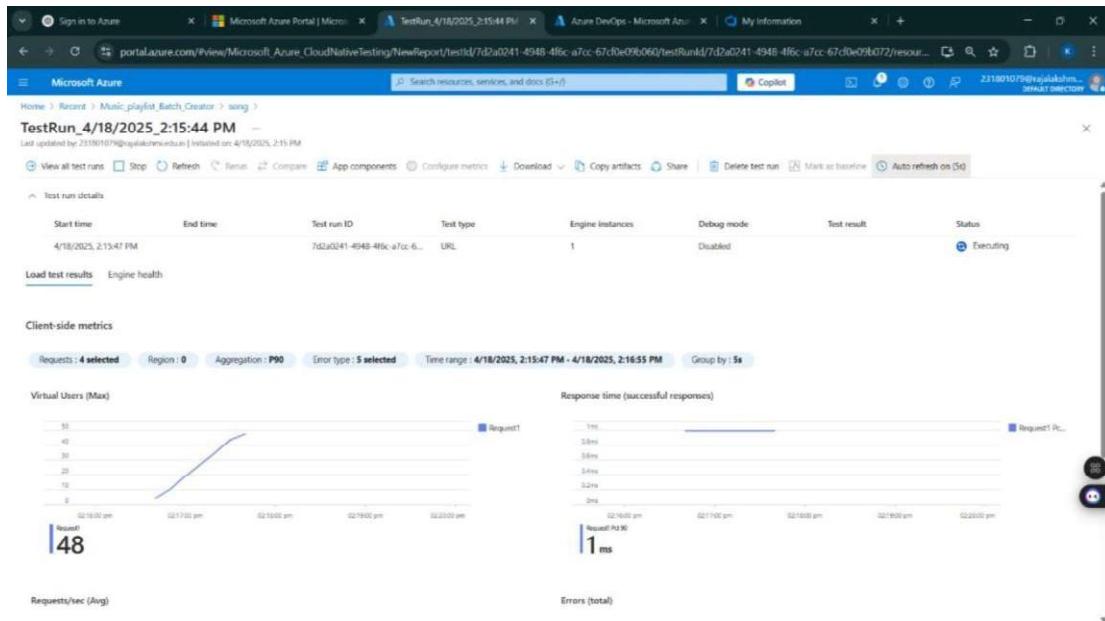
1. Sign in to Azure Portal
Go to <https://portal.azure.com> and log in.
2. Create the Resource
 - o Go to *Create a resource* → Search for “Azure Load Testing”.
 - o Select Azure Load Testing and click Create.
3. Fill in the Configuration Details
 - o *Subscription*: Choose your Azure subscription.
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

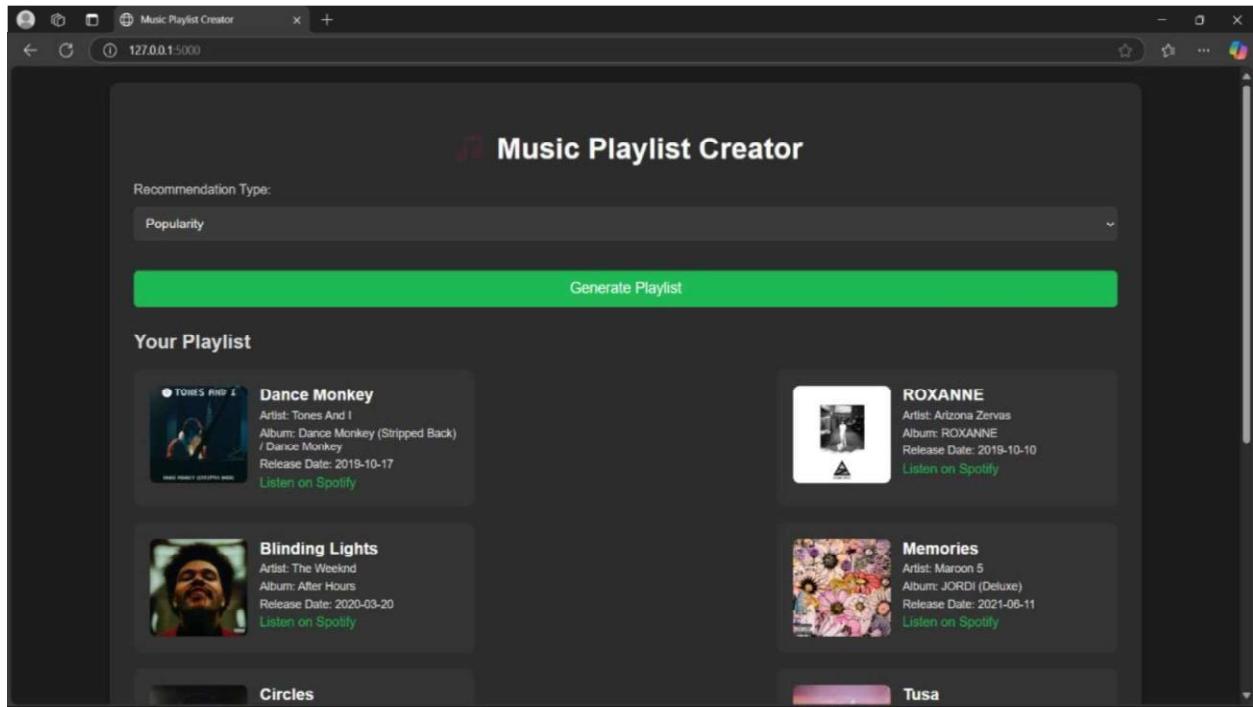
Steps to Create and Run a Load Test:

Once your resource is ready:

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

Load Testing





Result:

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