

**RAJALAKSHMI ENGINEERING COLLEGE**  
**RAJALAKSHMI NAGAR, THANDALAM – 602 105**



**RAJALAKSHMI  
ENGINEERING COLLEGE**

**CS23432  
SOFTWARE CONSTRUCTION**

**Laboratory Record Note Book**

Name : ...SHANTHINI C.....

Year/Branch/Section : ...II / IT / AE.....

Register No : .....231001188.....

Semester : .....IV.....

Academic Year : .....2024-2025.....



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

**BONAFIDE CERTIFICATE**

NAME SHANTHINI C REGISTER NO. 231001188

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**

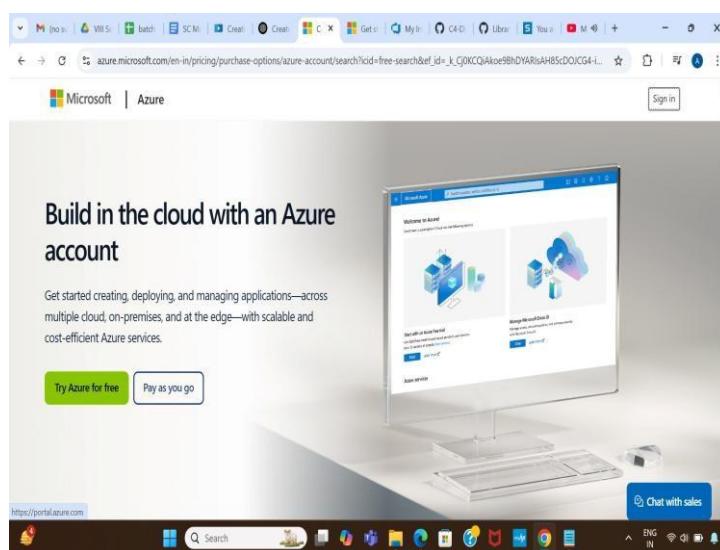
A user story is an informal, general explanation of a software feature written from the perspective of the end user. Its purpose is to articulate how a software feature will provide value to the customer.

User story template

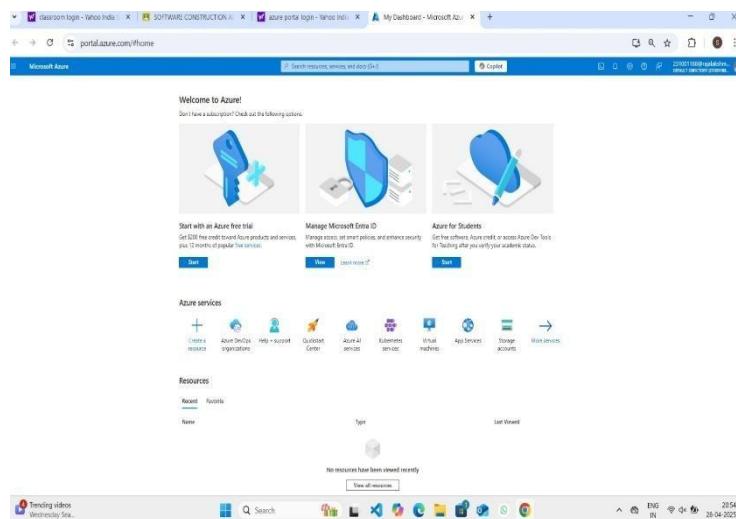
**"As a [role], I [want to], [so that]."**

### **PROCEDURE:**

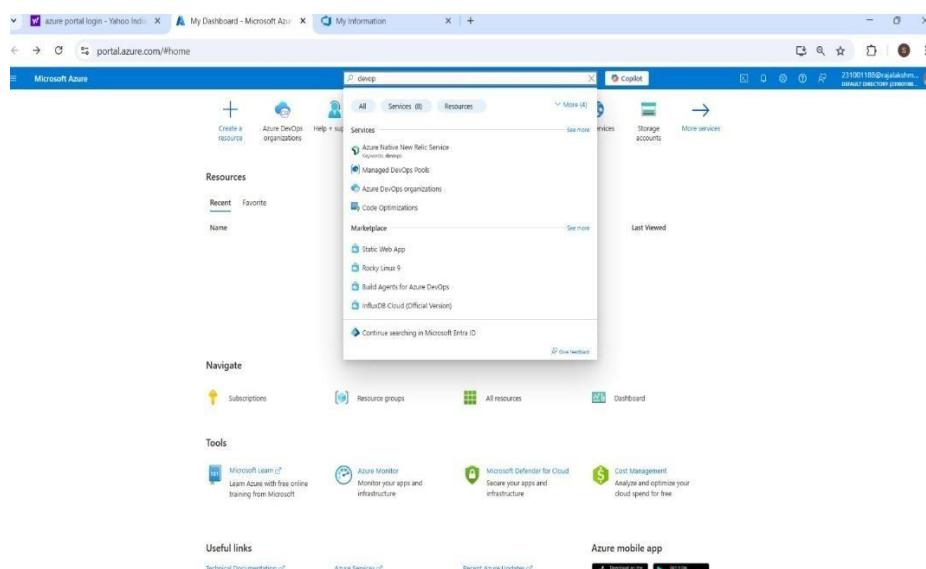
1. Open your web browser and go to the Azure website: <https://azure.microsoft.com/en-in>  
Sign in using your Microsoft account credentials. If you don't have an account, you'll need to create one.
2. If you don't have a Microsoft account, you can sign up for  
<https://signup.live.com/?lic=1>



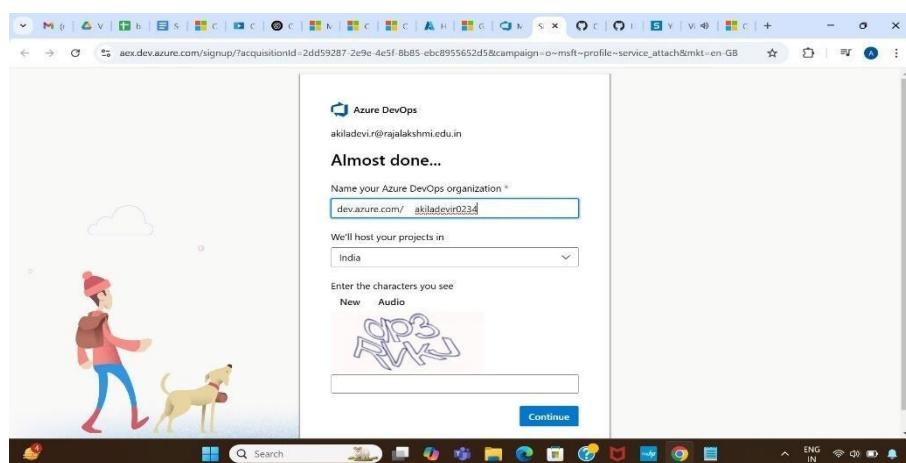
Azure home page



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.

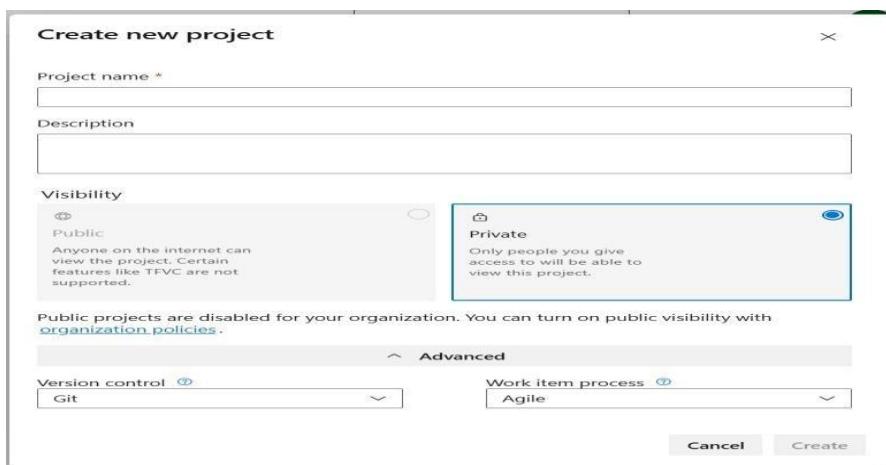


### 3.Create the First Project in Your Organization

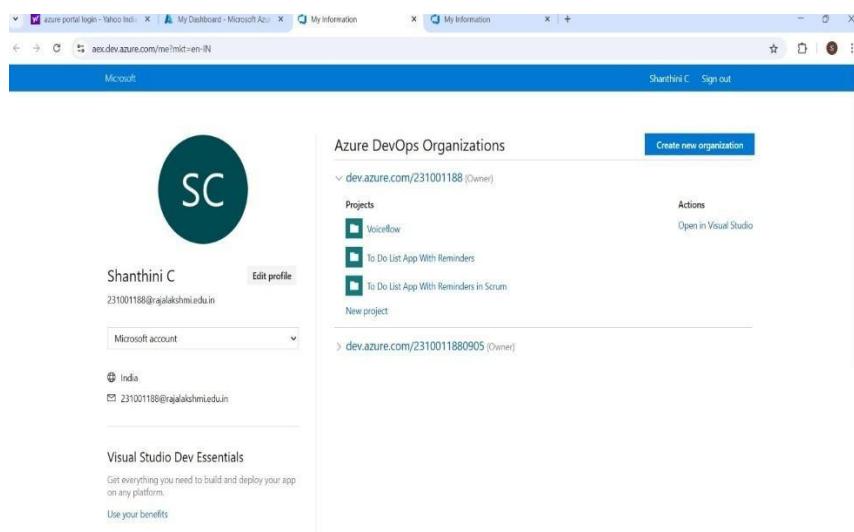
After the organization is set up, you'll need to create your first **project**. This is where you'll begin to manage code, pipelines, work items, and more.

- i. On the organization's **Home page**, click on the **New Project** button.
- ii. Enter the project name, description, and visibility options:
  - **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). iii.

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



3. Once logged in, ensure you are in the correct organization. If you're part of multiple organizations, you can switch between them from the top left corner (next to your user profile). Click on the Organization name, and you should be taken to the Azure DevOps Organization Home page.



## 4. Project dashboard

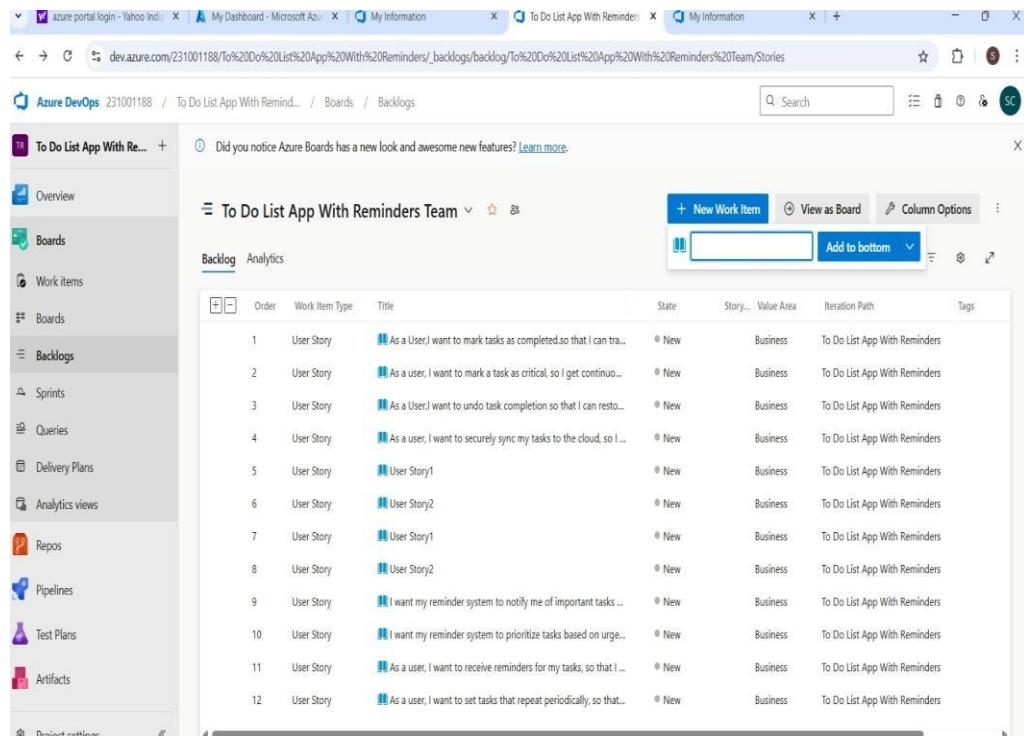
The screenshot shows the Azure DevOps project dashboard for 'To Do List App With Reminders'. The left sidebar includes links for Overview, Summary, Dashboards, Wiki, Boards, Repos, Pipelines, Test Plans, and Artifacts. The main area displays the 'About this project' section with a 'Help others to get on board!' field and a 'Project stats' box showing 0 Work items and 0 Work items assigned. A 'Members' section lists five team members with their status: Online (2), Offline (1), Available (1), and Busy (1).

## 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 'Work items - Boards' page in Azure DevOps. The left sidebar has a 'Work items' section under 'Boards'. The main area displays a table of work items with columns for ID, Title, Type, Assigned To, State, Area Path, Tags, Comments, and Activity Date. The table shows various user stories, tasks, and bugs, such as 'As a user, I want to add reminders for my tasks, so that I can access them from the cloud', 'As a user, I want to set tasks that repeat periodically, so that I don't have to remember them', and 'Emergency Mode for Critical Tasks'.

## 6.Fill in User Story Details



The screenshot shows the Azure DevOps interface for the 'To Do List App With Reminders Team' project. The left sidebar navigation includes 'Overview', 'Boards', 'Work items', 'Backlogs' (selected), 'Sprints', 'Queries', 'Delivery Plans', 'Analytics views', 'Repos', 'Pipelines', 'Test Plans', and 'Artifacts'. The main content area displays the 'Backlog' tab of the 'Analytics' section. At the top right are buttons for '+ New Work Item', 'View as Board', 'Column Options', and a search bar. Below these are buttons for 'Add to bottom' and a refresh icon. The backlog table has columns: Order, Work Item Type, Title, State, Story..., Value Area, Iteration Path, and Tags. There are 12 user stories listed:

Order	Work Item Type	Title	State	Story...	Value Area	Iteration Path	Tags
1	User Story	As a User,I want to mark tasks as completed.so that I can tra...	New	Business	To Do List App With Reminders		
2	User Story	As a user, I want to mark a task as critical, so I get continuo...	New	Business	To Do List App With Reminders		
3	User Story	As a User,I want to undo task completion so that I can resto...	New	Business	To Do List App With Reminders		
4	User Story	As a user, I want to securely sync my tasks to the cloud, so i ...	New	Business	To Do List App With Reminders		
5	User Story	User Story1	New	Business	To Do List App With Reminders		
6	User Story	User Story2	New	Business	To Do List App With Reminders		
7	User Story	User Story1	New	Business	To Do List App With Reminders		
8	User Story	User Story2	New	Business	To Do List App With Reminders		
9	User Story	I want my reminder system to notify me of important tasks ...	New	Business	To Do List App With Reminders		
10	User Story	I want my reminder system to prioritize tasks based on urge...	New	Business	To Do List App With Reminders		
11	User Story	As a user, I want to receive reminders for my tasks, so that I ...	New	Business	To Do List App With Reminders		
12	User Story	As a user, I want to set tasks that repeat periodically, so that...	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:**

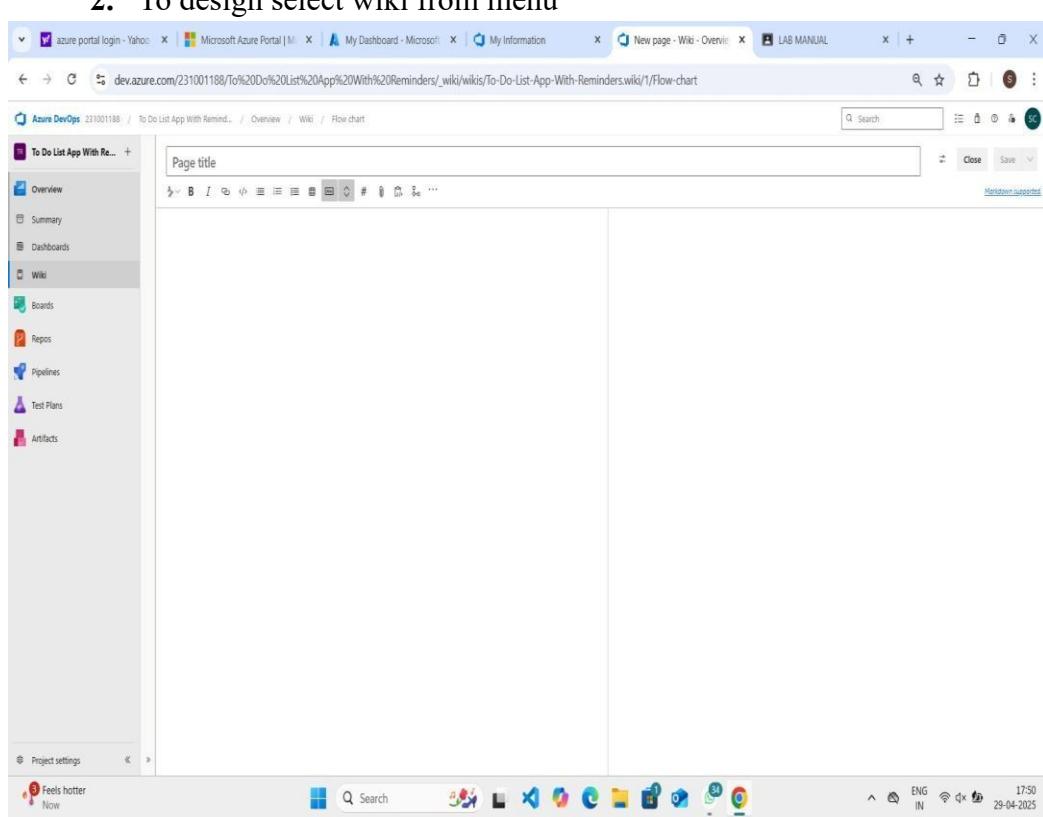
To design a Sequence Diagram by using Mermaid.js

### **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+)

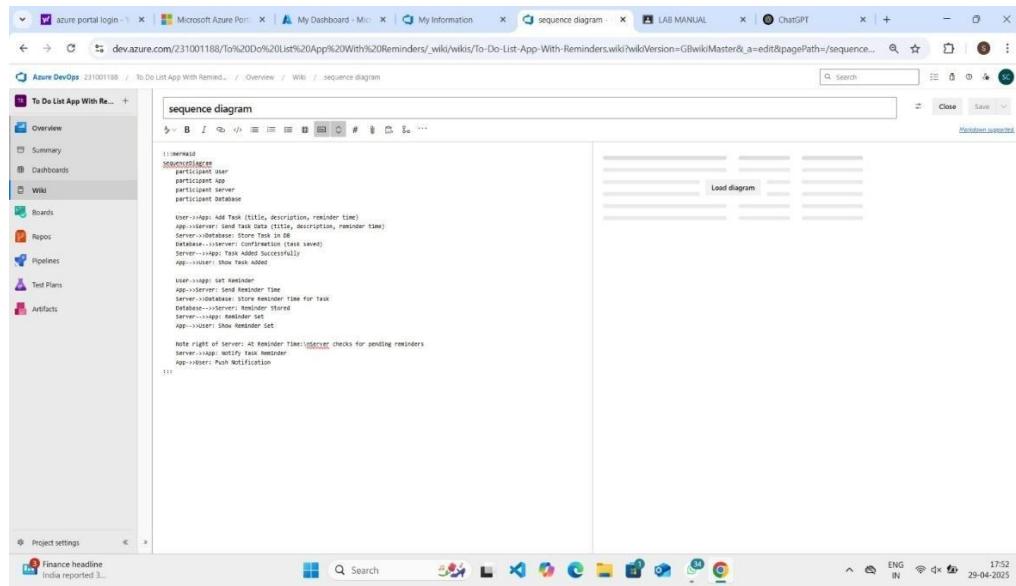
<<->> Dotted line with bidirectional arrowheads (v11.0.0+)

-x Solid line with a cross at the end

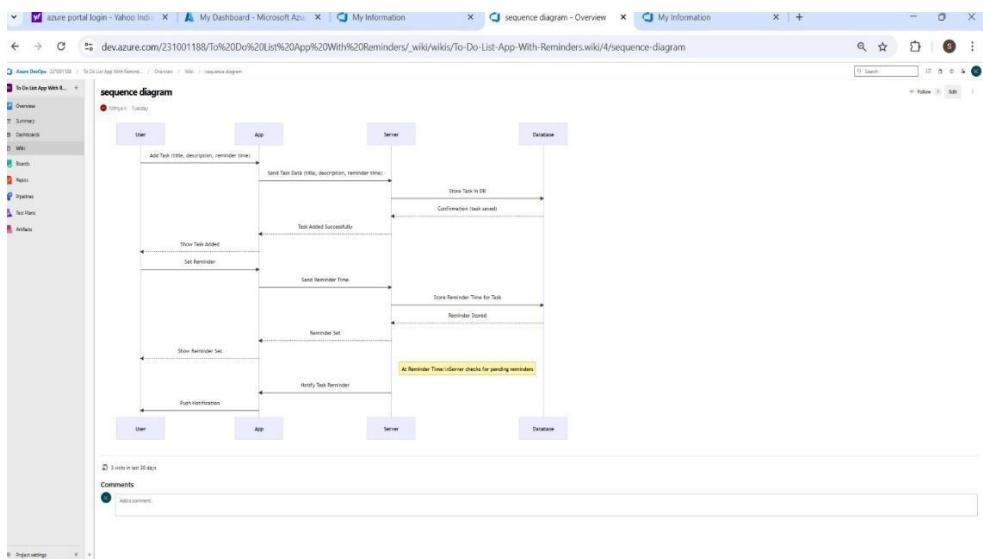
--x Dotted line with a cross at the end

-) 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

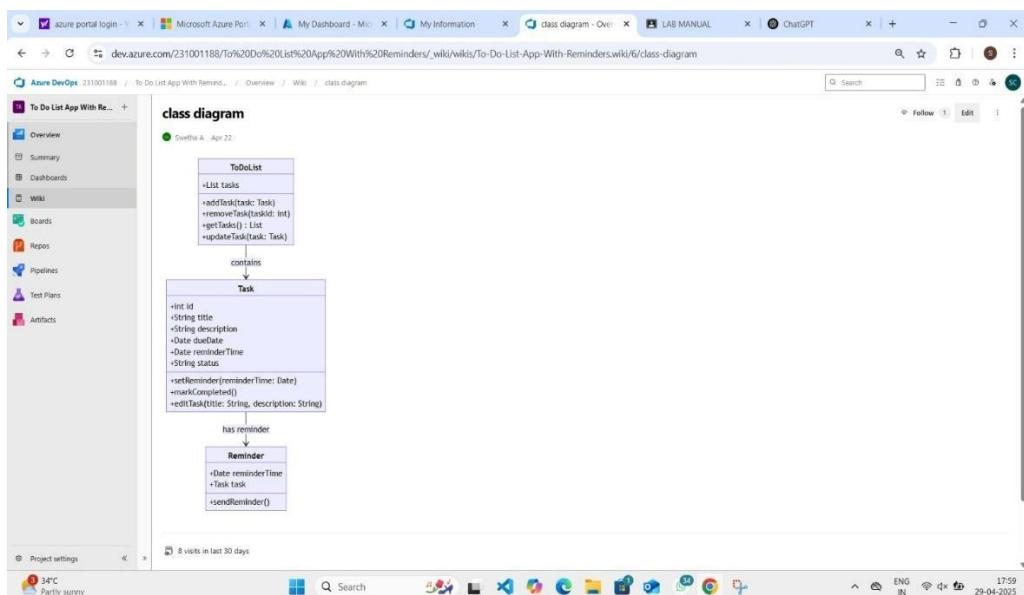
# DESIGNING CLASS DIAGRAM USING AZURE DEVOPS-WIKI

## AIM :-

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

## THEORY

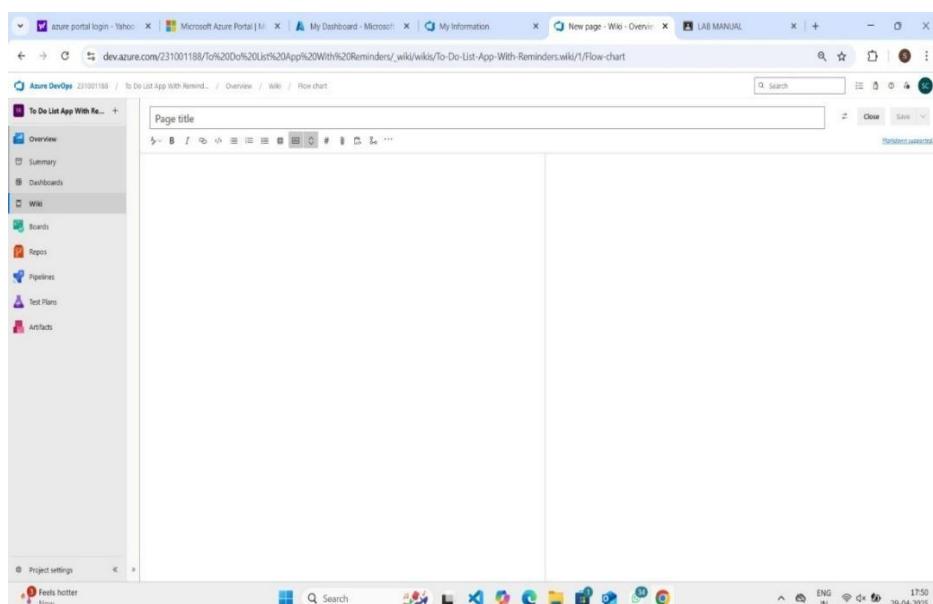
A UML class diagram is a visual tool that represents the structure of a system by showing its classes, attributes, methods, and the relationships between them.



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)
}

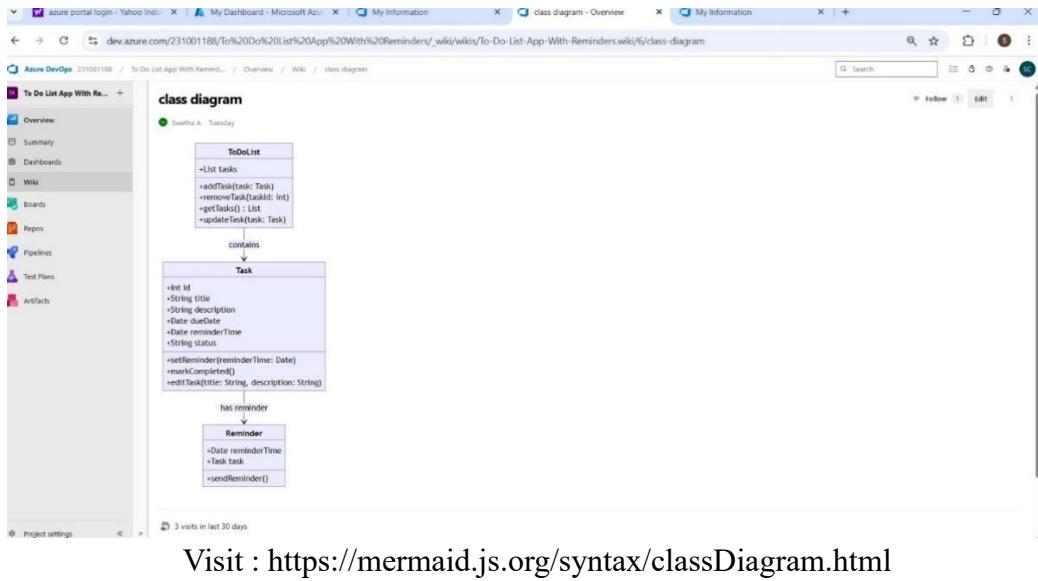
class Reminder {
    +Date reminderTime
    +Task task
    +sendReminder()
}

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**

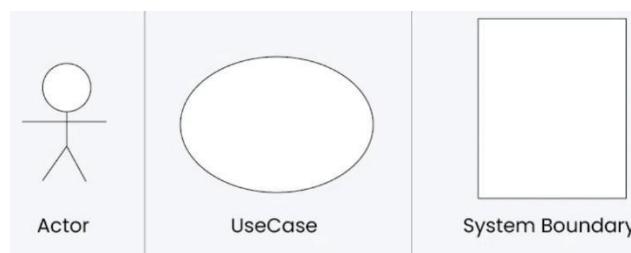
## **DESIGNING USECASE DIAGRAM USING AZURE DEVOPS-WIKI**

### **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 ([draw.io](https://draw.io)).
- 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.

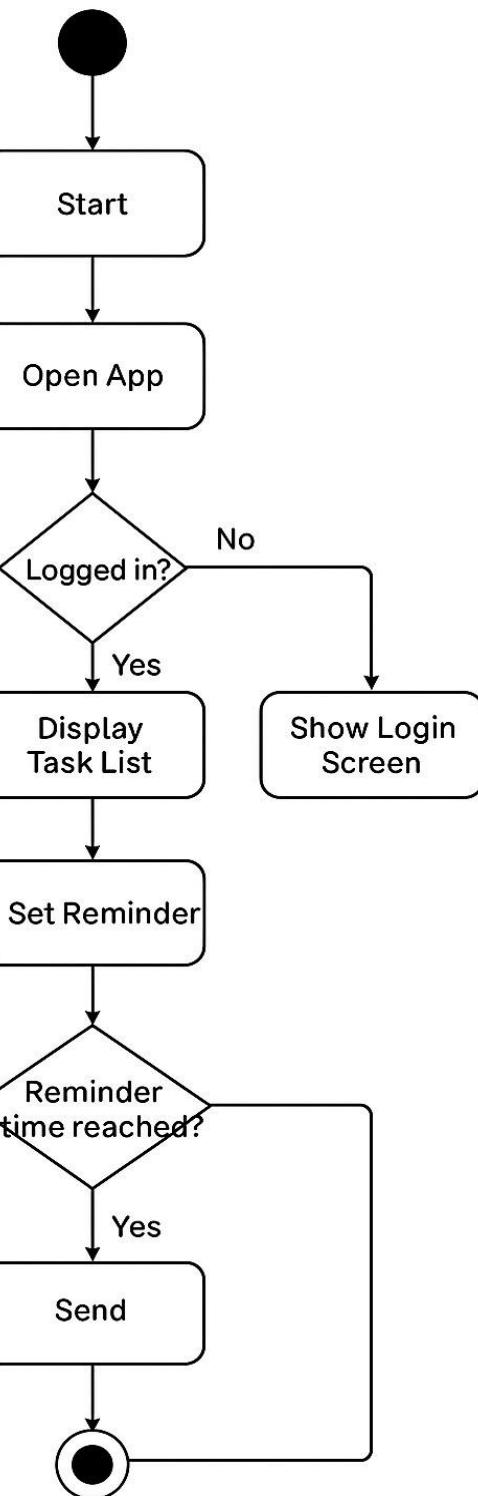
### **THEORY**

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



**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

### **Result:**

The architecture diagram was designed successfully

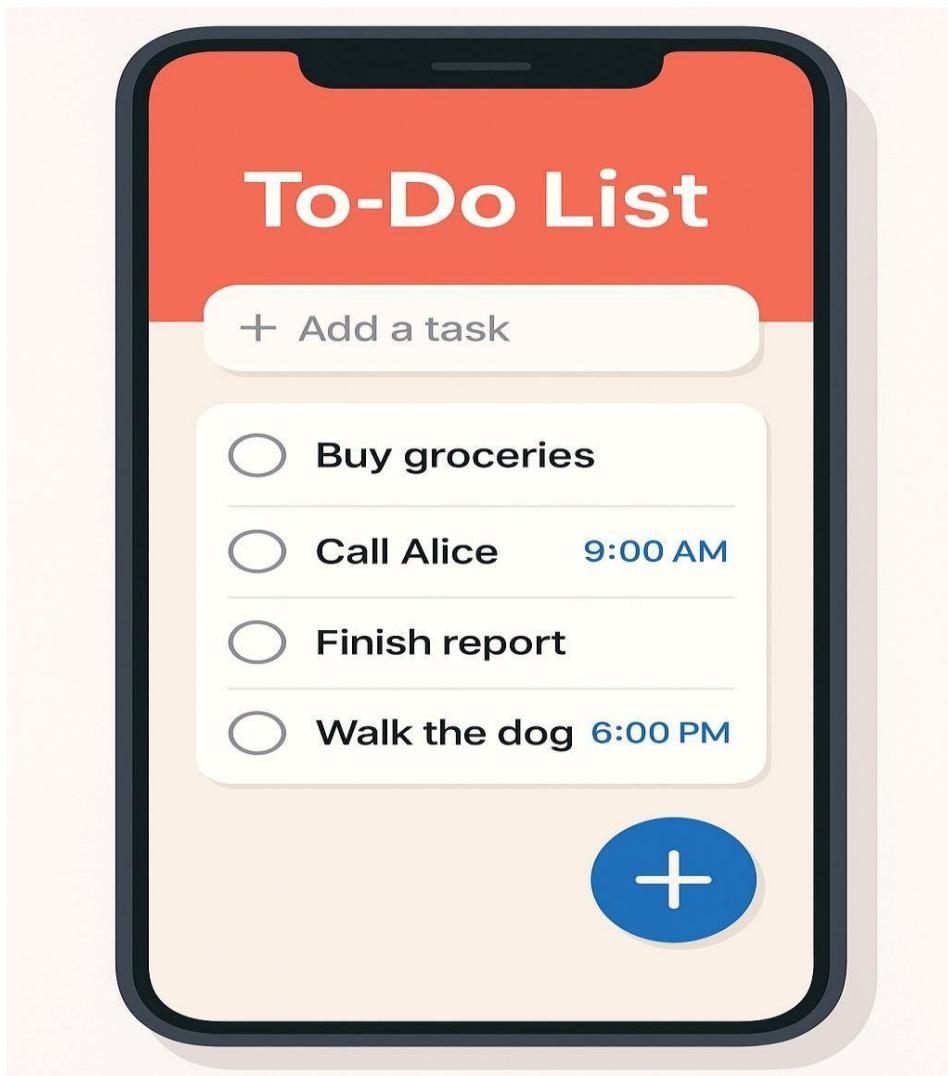
**EX NO. 10**

**DATE:01-04-2025**

## **DESIGN USER INTERFACE**

### **AIM:**

Design User Interface for the given project



### **Result:**

The UI was designed successfully.

**EX NO. 11**

**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'  
  
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

New Test Plan

Name \*  
Music Playlist Batch Creator - Test Plan

Area Path \*  
Music Playlist Batch Creator

Iteration \*  
Music Playlist Batch Creator|Integration

Create Cancel

### 2. Test suite

Test Cases (4 items)	Order	Test Case ID	Assigned To	State
TC01 - Successful Sign Up	1	78	Karthikayen Se... Design	
Static suite	2	80	Karthikayen Se... Design	
Requirement based suite	3	81	Karthikayen Se... Design	
Query based suite	4	82	Karthikayen 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**

- **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 a Microsoft Azure DevOps Test Plan interface. The left sidebar lists various project sections like Music, Overview, Boards, Repos, Pipelines, Test Plans, Programs, Parameters, Configurations, Runs, and Artifacts. The main area displays a test case titled "TC06 - Playlist Loading Failure" under the "Test Cases" section. The test case details are as follows:

- Test Case ID:** 77
- Category:** TEST CASE ??
- Assigned To:** Karthick S
- Comments:** 0
- Attachments:** Add Tag
- Mung:** Design
- Reason:** None
- Iteration:** Music Playlist Batch Creator/Integration

The "Steps" tab shows the following steps:

Step	Action	Expected result	Attachments
1.	Disconnect from internet	Network is offline	
2.	Navigate to "My Playlists"	Error message "Unable to load playlists" is shown	

The "Custom" section includes:

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

At the bottom, there are "Parameter values" and "Project settings" buttons.

The screenshot shows the Azure DevOps Test Plan interface. A test case named 'TC05 – View Playlist Page' is selected. The test case details include:

- State:** Design
- Reason:** None
- Area:** Music Playlist Batch Creator
- Iteration:** 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. Expected result: All created playlists are displayed clearly.

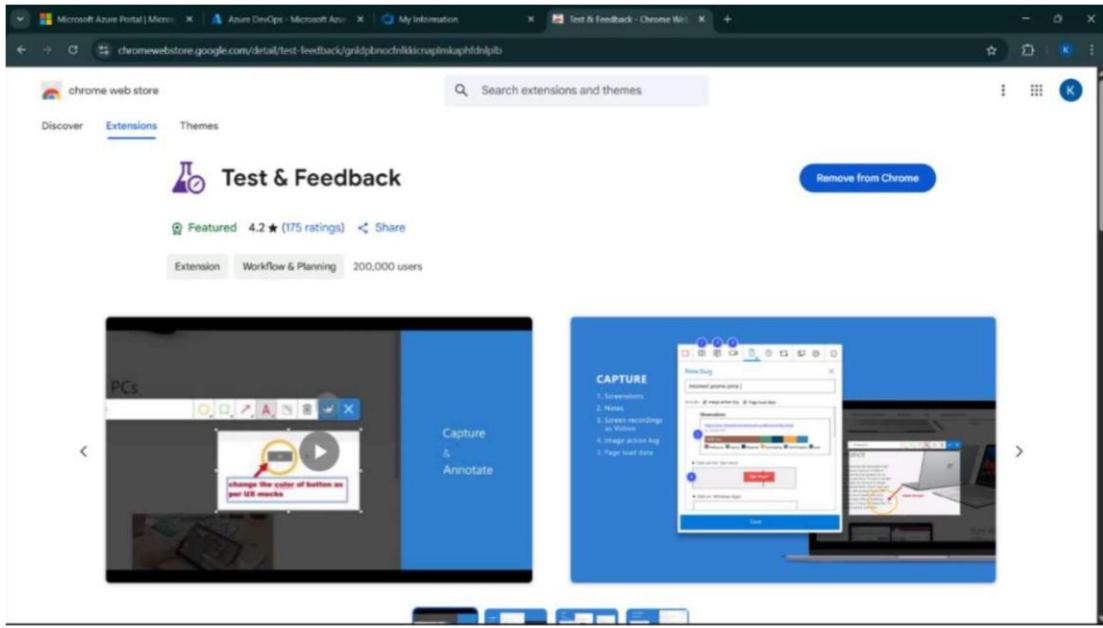
The **Custom** section includes fields for Type (Happy Path) and Status (Priority 2). The Automation status is listed as Not Automated.

#### 4. Installation of test

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

- Rating:** 4.2 stars (175 ratings)
- Category:** Workflow & Planning
- Users:** 200,000 users
- Description:** Capture & Annotate

The extension interface is shown with a screenshot of a capture tool and a screenshot of a browser window with annotations.



## Test and feedback

Showing it as an extension

A screenshot of a web browser window showing the Azure DevOps Test Plan interface. The URL is dev.azure.com/231801095/Music%20playlist%20Batch%20Creator/\_testPlans/define?planId=84&amp;suitId=86. The left sidebar shows 'Test Plans' selected. The main area displays a test plan for 'Music Playlist Batch Creator'. A floating 'Extensions' sidebar is open, listing the 'Test &amp; Feedback' extension as installed. Other extensions listed include 'Copy Text from Picture', 'Dark Reader', 'Monica: ChatGPT AI Assistant', and 'SelectedText: Copy text from V...'. The 'Manage extensions' button is also visible in the sidebar.

## 5. Running the test cases

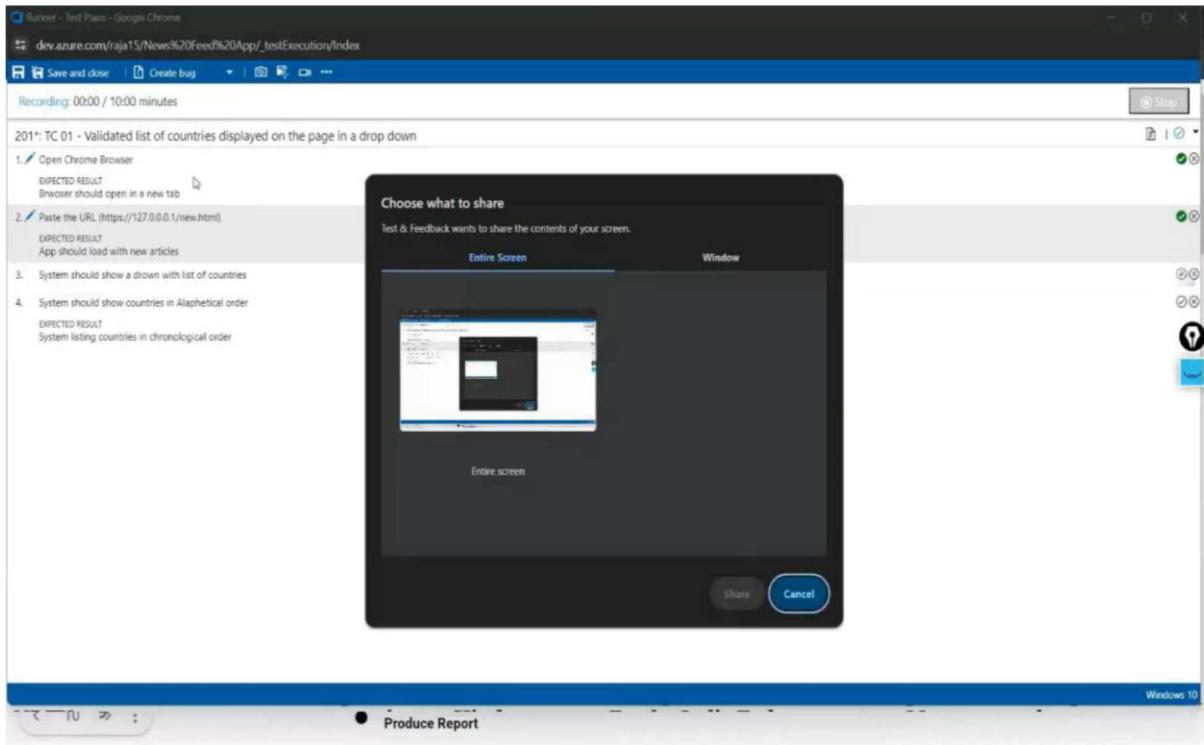
The screenshot shows the Azure DevOps Test Plan interface for a project named "Music Playlist Batch Creator". The left sidebar is collapsed, and the main area displays the "TS02 - View Playlists (ID: 87)" test plan. The "Execute" tab is selected, showing two test points: "TC05 - View Playlist Page" and "TC06 - Playlist Loading Failure". Both test cases have a status of "Passed". A context menu is open over the first test case, listing options such as "Run for web application", "Run for desktop application", and "Run with options".

The screenshot shows the "Runner - Test Plans" interface in Google Chrome, displaying the results of test case "75: TC05 – View Playlist Page". The test steps are:

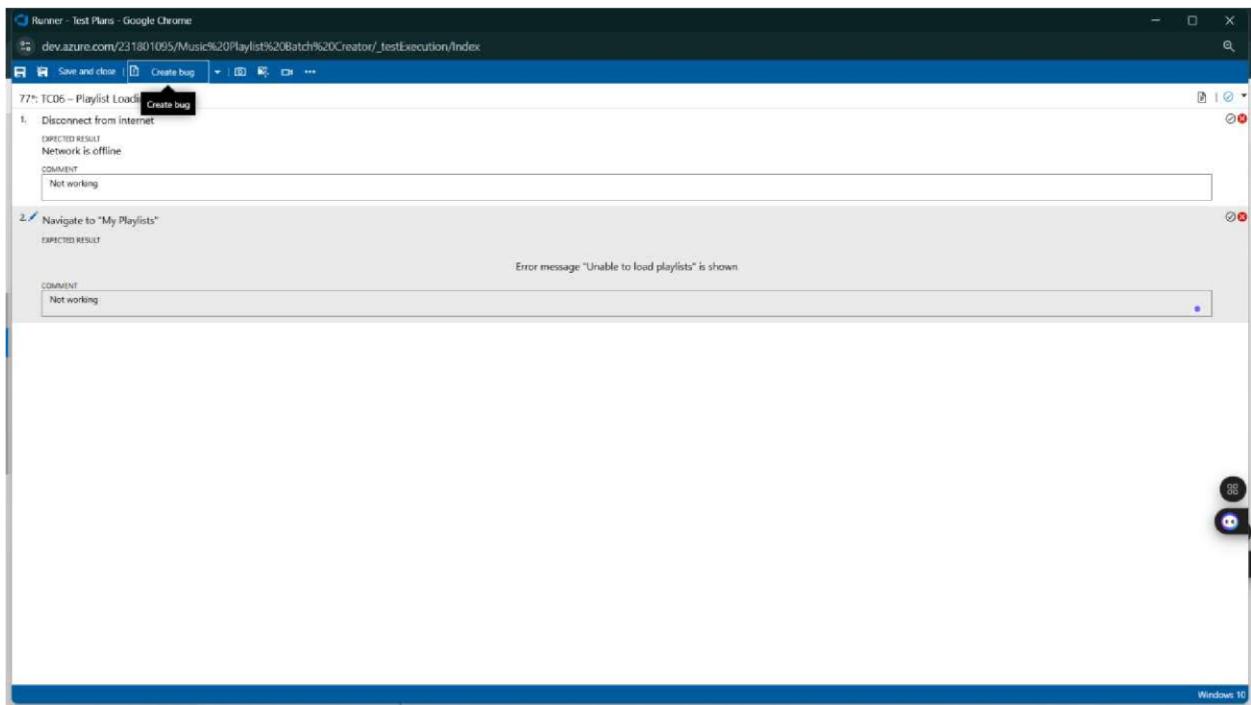
1. Log in successfully  
EXPECTED RESULT:  
User is redirected to dashboard
2. Navigate to "My Playlists" section  
EXPECTED RESULT:  
All created playlists are displayed clearly

The results for both steps are marked as "Passed".

## 6.Recording the test case



## 7.Creating the bug



Runner - Test Plans - Google Chrome  
dev.azure.com/231801095/Music%20Playlist%20Batch%20Creator/\_testExecution/Index

77: TC06 – Playlist Loading Failure

1. Disconnect from internet NEW BUG\*

2. Navigate to "My Playlists"

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

Unassigned | 0 comments | Add tag | TB01 - Playlist loading spinner keeps spinning indefinitely on poor network | Save & Close | ...

**Repro Steps**

18-04-2025 03:23 Bug filed on "TC06 – Playlist Loading Failure"

Step no.	Result	Title
1.	Failed	Disconnect from internet
	Expected Result:	Network is offline
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: NEW BUG\*  
Area: Music Playlist Batch Creator  
Iteration: Music Playlist Batch Creator

Priority: 2  
Severity: 3 - Medium  
Activity

**Deployment**

To track releases associated with this work item, go to 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: 77 TC06 – Playlist Loading Failure  
Updated 10-04-2025. ◉ Design

**System Info**

Found in Build

Windows 10

Microsoft Azure Portal | Azure DevOps - Microsoft | My Information | Test Plan B4 Music | Runs - Test Plans | Settings - Overview | Bug Report Playlist | ...

dev.azure.com/231801095/Music%20Playlist%20Batch%20Creator/\_testManagement/runs?\_a=resultSummary&runId=48&resultId=100000

Azure DevOps 231801095 / Music Playlist Batch Creator / Test Plans / Runs

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

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

Unassigned | 0 comments | Add tag | Save & Close | Follow | ...

Updated by Karthick S 8m ago

**System Info**

Found in Build  
Integrated in build

Browser - Name	Google Chrome 135
Browser - Language	en-IN
Browser - Height	864
Browser - Width	1536
Browser - User agent	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/135.0.0.0 Safari/537.36
Operating system - Name	Windows NT 10.0; Win64; x64
Operating system - Architecture	x64_64
Operating system - Processor model	11th Gen Intel® Core™ i3-1115G4 @ 3.00GHz
Operating system - Number of processors	4
Memory - Available	814784512
Memory - Capacity	8216260128
Display - Pixels per inch (X axis)	120
Display - Pixels per inch (Y axis)	120
Display - Device pixel ratio	1.25

**Discussion**

Add a comment. Use # to link a work item, / to link a pull request, or @@ to mention a person.

Name: SystemInformation-2025-04-18T03-23-58.168Z.json  
Size: 1K

Project settings

## 8. Test case results

The screenshot shows the Azure DevOps interface for a project named "Music Playlist Batch Creator". On the left, the "Test plans" section is selected, showing a list of test suites: "TS01 - User Login (4)", "TS02 - View Playlists (2)", "TS03 - Real-Time Metrics (1)", "TS04 - Playlist Editing (4)", and "TS05 - Smart Playlist (1)". The "TS02 - View Playlists" suite is currently selected. On the right, a detailed view of "TS02 - View Playlists (ID: 87)" is shown, with the "Execute" tab selected. Under "Test Points (2 items)", the point "TC05 - View Playlist Page" is highlighted. A modal window titled "TC05 – View Playlist Page" displays the "Test Case Results" table:

Outcome	TimeStamp	Configuration	Run by	Tester	Test Pl.
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	Karthickey Senthil	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	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 "Work items" section is selected, showing a list of recently updated items. One item, "BUG 203: BUG 01 - Countries Drop down Not Available on the page", is selected and displayed in detail. The work item has the following details:

- Title:** BUG 01 - Countries Drop down Not Available on the page
- Assignee:** rajesh prabhu
- Status:** New
- Reason:** New
- Repro Step:** Active
- Iteration:** News Feed App
- Details:**
  - 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/new.html>). **Expected Result**: App should load with new articles.
  - Step no.**: 3. **Result**: Failed. **Title**: System should show a dropdown with list of countries. **Expected Result**: Produce Report.
- Planning:**
  - Resolved Reason: Validated list of countries displayed on the
  - Story Points: 2
  - Priority: 2
  - Severity: 3 - Medium
  - Activity:
- 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.
- Effort (Hours):** Original Estimate:
- Related Work:**

- Assigning bug to the developer and changing state

**Bug Detail Page:**

- Test Step 1:** Failed. Description: Disconnect from internet. Expected Result: Network is offline.
- Test Step 2:** Failed. Description: Comments: Page Not loading. Navigate to "My Playlists". Expected Result: Error message "Unable to load playlists" is shown.

**Planning:** Resolved Reason: BUG-92, Area: Music Playlist Batch Creator, Reason: New, Reporter: Music Playlist Batch Creator.

**Deployment:** To track releases associated with this work item, go to Boards in your pipeline's Options menu. Learn more about deployment status reporting.

**Related Work:** 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.

**System Info:** Tested By: KARTHICK-231801095, Updated: 10-04-2025, Design.

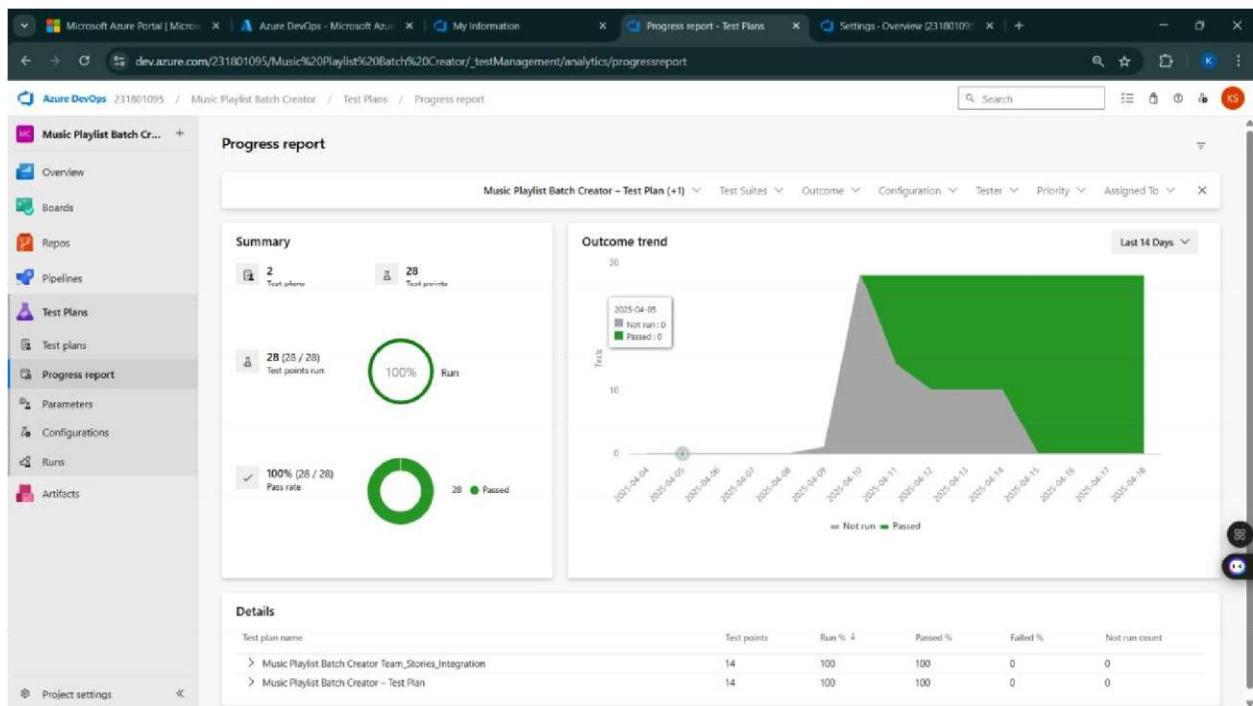
## 10. Progress report

**Progress Report Summary:**

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

**Outcome Trend:** Last 14 Days. The trend shows a significant increase in passed tests starting around April 10th.

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

**Organization Settings**

231801095

**All processes**

Processes Fields

Name	Description	Team projects
Basic	This template is flexible for any process and great for teams getting started with Azure DevOps.	0
Agile	This template is flexible and will work great for most teams using Agile planning methods, including those pract...	0
Scrum	This template is for teams who follow the Scrum framework.	0
CMMI	This template is for more formal projects requiring a framework for process improvement and an auditable reco...	0

Screenshot of the Azure DevOps Settings - Process page (dev.azure.com/231801095/\_settings/process). The page displays a list of available 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

The left sidebar shows the navigation menu under Boards, with Process selected. Other sections include General, Security, Pipelines, and Fields.

Screenshot of the Azure DevOps Settings - Process page (dev.azure.com/231801095/\_settings/process) after creating a new process template. The table now includes the new entry:

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

The left sidebar shows the navigation menu under Boards, with Process selected. Other sections include General, Security, Pipelines, and Fields.

## 12. View the new test case template

The screenshot shows the 'Add a field to Test Case' dialog box over a background of the Azure DevOps settings interface. The dialog has a 'Definition' tab selected, showing options to 'Use an existing field' (with 'Acceptance Criteria' selected) or 'Create a field'. Under 'Create a field', fields for 'Name' (Type), 'Type' (Text single line), and 'Description' (Optional) are defined. Buttons for 'Add field' and 'Cancel' are at the bottom.

The screenshot shows the 'All processes' page in the Azure DevOps settings. A new work item type, 'Music Playlist Batch Creator', has been added under the 'Work item types' section. The 'Description' field contains the text: 'Azure Music Playlist Batch Creator. The Azure Music Playlist Batch Creator is a cloud-based solution designed for bulk playlist creation and management. Leverage...'. The 'General' sidebar on the left is visible.

The screenshot shows the Azure DevOps Settings - Process page. The URL in the browser is [dev.azure.com/231801095/\\_settings/process?type-id=231801095Agile.TestCase&process-name=231801095%20Agile&\\_a=layout](https://dev.azure.com/231801095/_settings/process?type-id=231801095Agile.TestCase&process-name=231801095%20Agile&_a=layout). The page title is "All processes > 231801095 Agile > Test Case". On the left, there is a sidebar with "Organization Settings" for "231801095" and a search bar. The main content area shows a "Steps" section with a "Text (multiple lines)" input field. To the right, there are sections for "Recent test results", "Deployment", "Development", "Related Work", and "Status". A vertical scroll bar is visible on the right side of the main content area.

## 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:

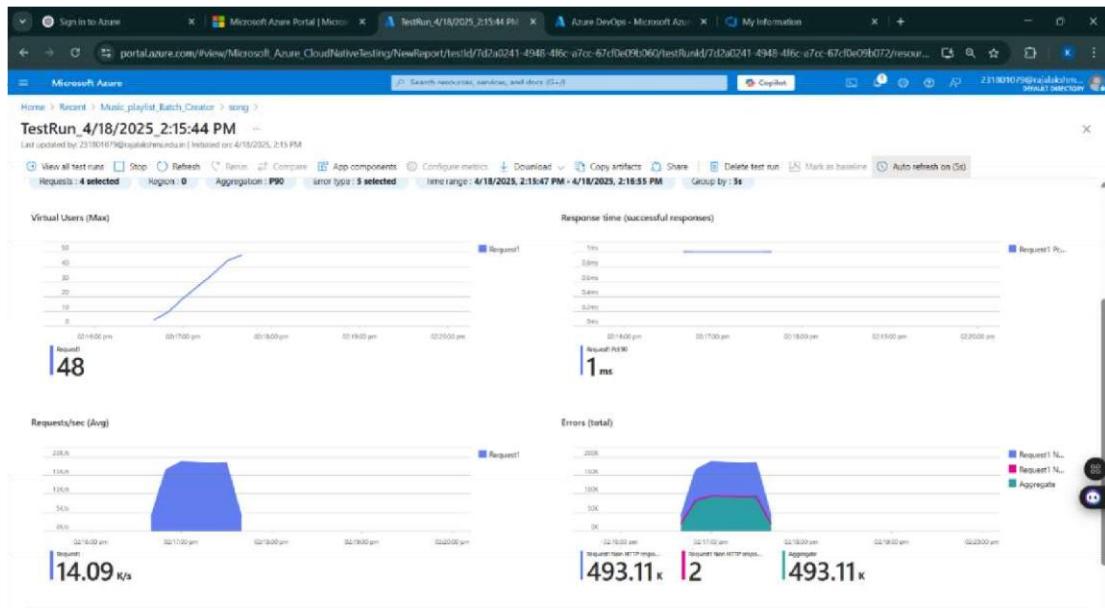
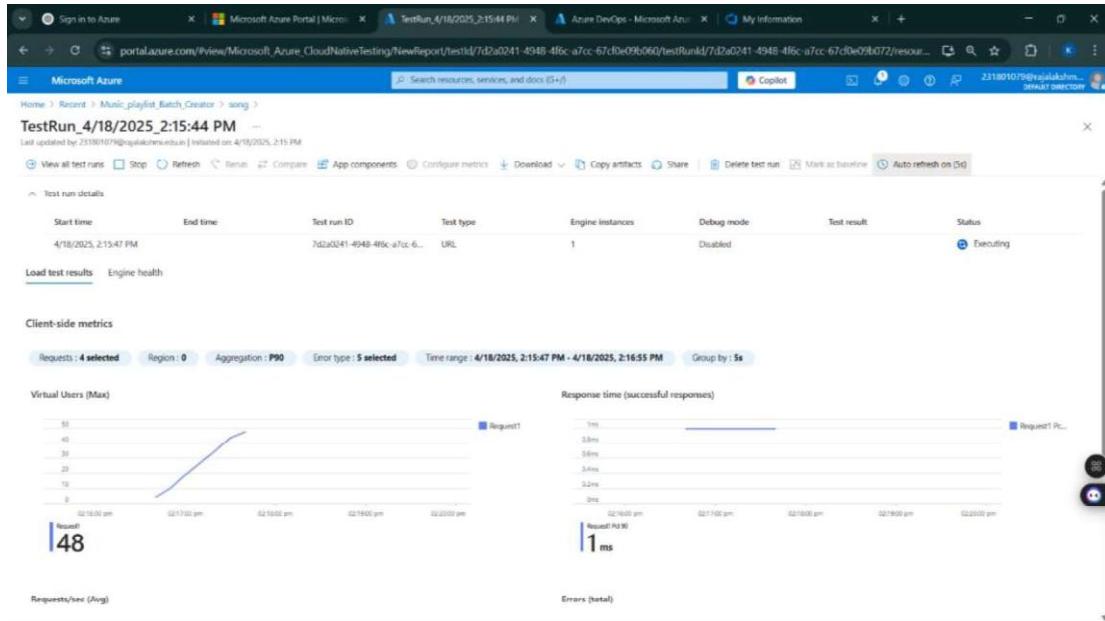
1. Sign in to Azure Portal  
Go to <https://portal.azure.com> and log in.
2. Create the Resource
  - o Go to *Create a resource* → Search for “Azure Load Testing”.
  - o Select Azure Load Testing and click Create.
3. Fill in the Configuration Details
  - o *Subscription*: Choose your Azure subscription.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

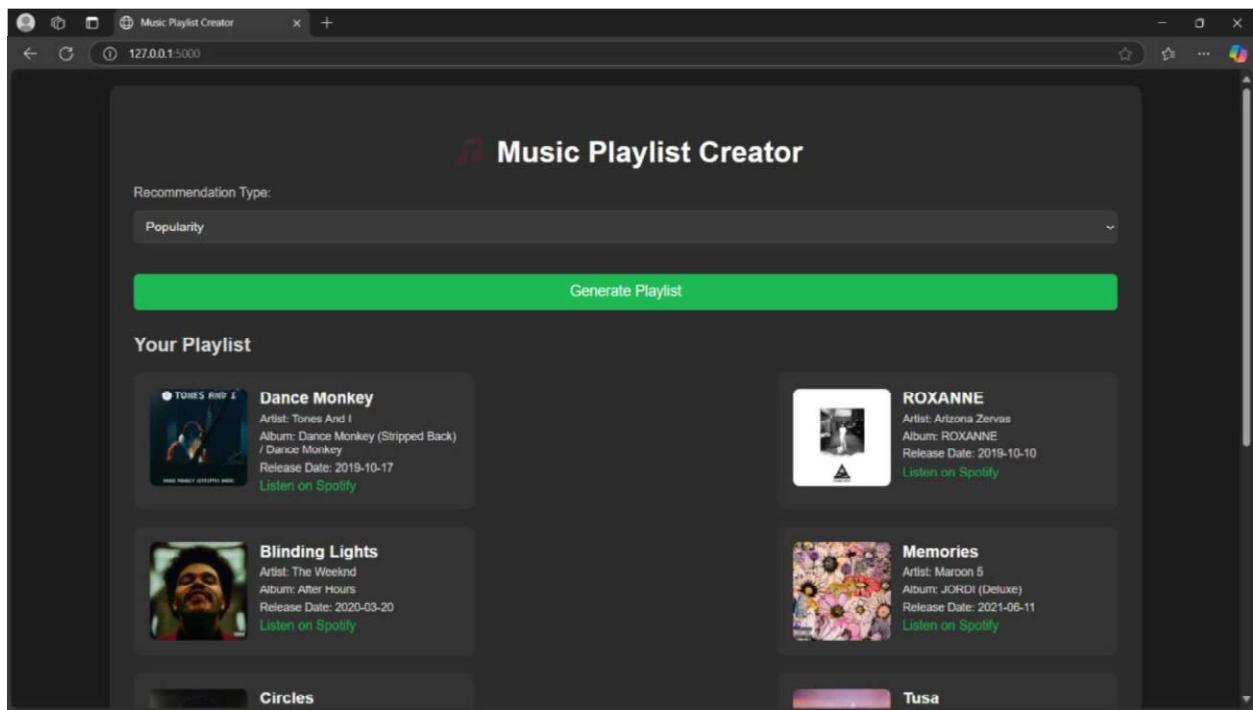
### Steps to Create and Run a Load Test:

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.