RAJALAKSHMI ENGINEERING COLLEGE

RAJALAKSHMI NAGAR, THANDALAM – 602 105

HARDWORK DISCIPLINE

REC

RAJALAKSHMI
ENGINEERING COLLEGE

CS23432
SOFTWARE CONSTRUCTION

Laboratory Record Note Book

Name : . . . .

Year / Branch / Section : . . . .

Register No. : . . . .

Semester : . . . .

Academic Year : . . . .

RAJALAKSHMI ENGINEERING COLLEGE (AUTONOMOUS)

RAJALAKSHMI NAGAR, THANDALAM – 602 105

BONAFIDE CERTIFICATE

NAME ______GOKULA SRINITHI M____ REGISTER NO. ____2116231001048

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

Signature of Faculty -in – Charge

Submitted for the Practical Examination held on __________________

Internal Examiner External Examiner

LAB PLAN

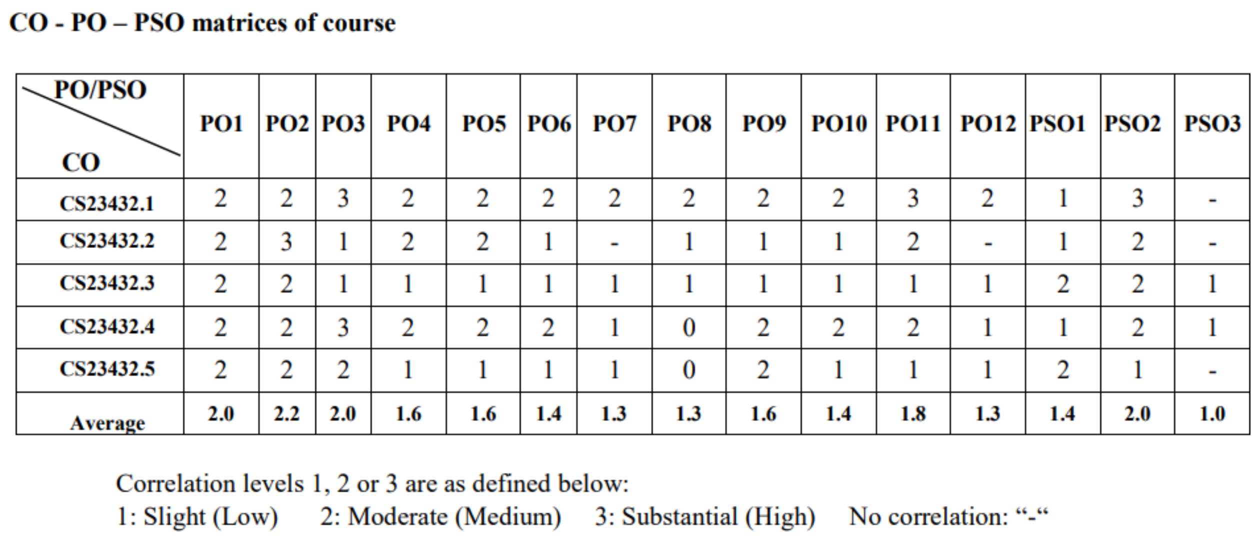
CS19442-SOFTWARE ENGINEERING LAB

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		
5	25/02/2025	Sequence Diagram		
6	04/03/2025	Class Diagram		
7	11/03/2025	Use Case Diagram		
8	18/03/2025	Activity Diagram		
9	25/03/2025	Architecture Diagram		
10	01/04/2025	User Interface		
11	08/04/2025	Implementation		
12	15/04/2025	Testing – Test Plan And Test Cases		

Course Outcomes (COs)

Course Name: Software Engineering
Course Code: CS23432

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



CO - PO – PSO matrices of course

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

Correlation levels 1, 2 or 3 are as defined below:
1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High) No correlation: "-"

EX 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 given project.

Problem Statement:

Effective communication is a critical need in both personal and professional environments. In today's fast-paced world, users require a simple, reliable, and efficient method of sending and receiving messages. Despite the wide range of messaging platforms available, many users still face challenges when it comes to managing conversations, particularly when communicating with multiple contacts at once.

Traditional SMS systems often lack advanced functionalities such as group messaging, editable contact groups, message history management, and visibility controls. Existing solutions can be fragmented, unintuitive, or fail to meet the diverse needs of users who need to coordinate information quickly across several individuals or groups. This gap leads to inefficiencies, miscommunication, missed opportunities, and user frustration.

Our project aims to design and build a robust **SMS/Text Messaging System** that focuses on providing a seamless user experience with core features tailored to modern communication needs. Key functionalities will include:

- **Group Messaging:** Users will have the ability to create groups with multiple contacts, enabling them to send and receive messages within a group context.
- **Editable Groups:** Users can edit group names, add or remove members, and manage group settings dynamically.
- **Visibility and Transparency:** All group members will have access to the full message history and updates, ensuring transparency and improved collaboration.
- **Reliable Delivery:** Messages should be delivered promptly with clear status updates to avoid uncertainty in communication.

The system will be designed to prioritize **usability**, **security**, and **scalability**. It will provide a clean interface suitable for both individual users and organizations, supporting use cases like internal team communication, customer support via SMS, event coordination, and more.

By delivering a comprehensive, user-friendly SMS/Text Messaging System, this project seeks to bridge the communication gaps present in current solutions and create a platform that supports efficient, real-time interaction with flexibility and ease.

Result:

The problem statement was written successfully**.**

EX NO: 3 **DATE: 04/02/2025**

AGILE PLANNING

Aim:

To prepare an AgilePlan.

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 helpfulway 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

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]."

Epic 1: to create a user account , login process, and the user data management

Feature:

forgot password options

User Story 1: forgot password option

as a user , I want forgot password option so if i forgot my password ,i can reset the password and access the service .

Acceptance Criteria:

- user verification through email id or phone number by providing a code or opt.
- the code or opt should match the users input
- allow user to change password and save the password .
- if the code or opt is wrong ,give option to repeat the process.

User Story 2: To create an user account

as a user , i want to create account so i can access the service .

Acceptance Criteria:

- allow user to enter the details required
- all mandatory fields should be filled.
- user details should be stored.
- allow user to login and logout.

Epic 2: Messaging Core System

Feature:

Group Messaging

Acceptance Criteria:

- ❖ **User can create a group with multiple contacts.**
- ❖ **Group messages are visible to all members.**
- ❖ **Group name and members are editable.**
- ❖

Feature:

One-to-One Messaging

User Story1 : One-to-One Messaging

As a user, I want to send and receive messages so that I can chat with my contacts.

Acceptance Criteria:

- ▪ **User can type and send text messages.**
- ▪ **Time stamp is displayed for every message**
- ▪ **Messages sent are visible in the chat.**

Feature : handling trash messages

User Story 1: Create product listing

As a user, I want to create groups to chat with multiple at once.

User Story 1: Messages in Trash

As a user, I want to push the unwanted messages to the trash.

Acceptance Criteria:

1.able to see the deleted messages or messages in the trash

2.able to restore the messages

3.able to permenantly delete the message

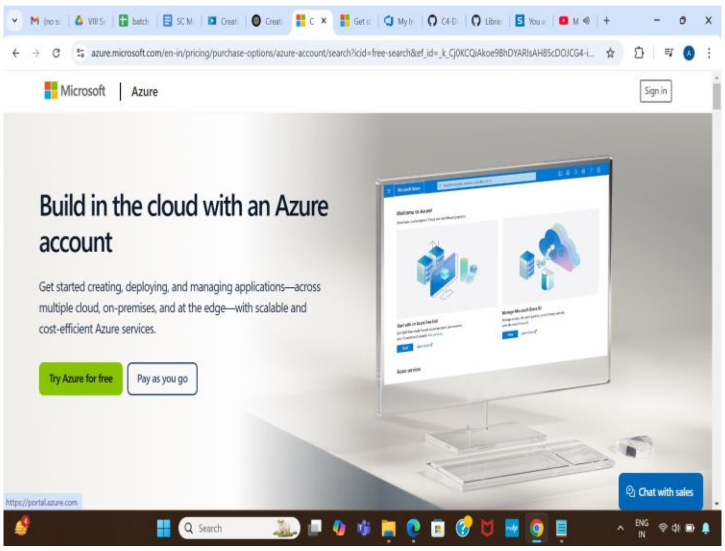
4.able to see the count of messages in trash on the top of trash icon

5.messages in trash will be deleted automatically after a year

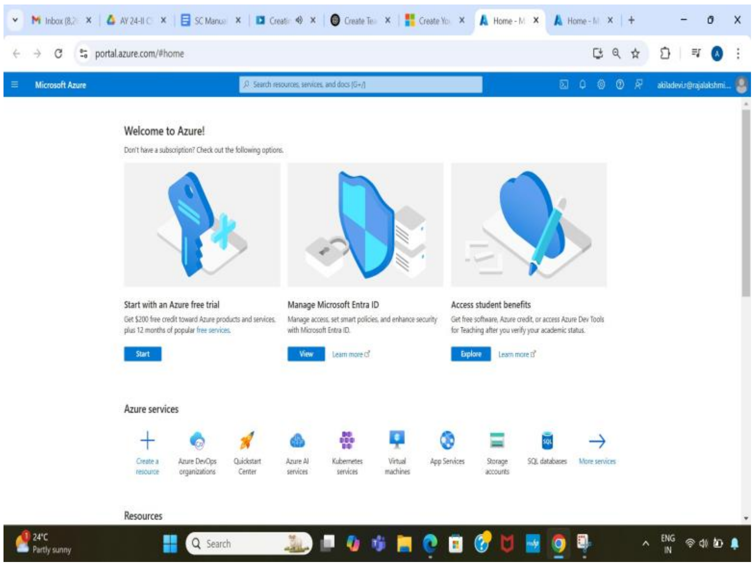
6.able to change the automatic delete option(6months/year)

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*


Microsoft | Azure
Sign in
Build in the cloud with an Azure account
Get started creating, deploying, and managing applications—across multiple cloud, on-premises, and at the edge—with scalable and cost-efficient Azure services.
Try Azure for free
Pay as you go
Chat with sales

3. Azure home page



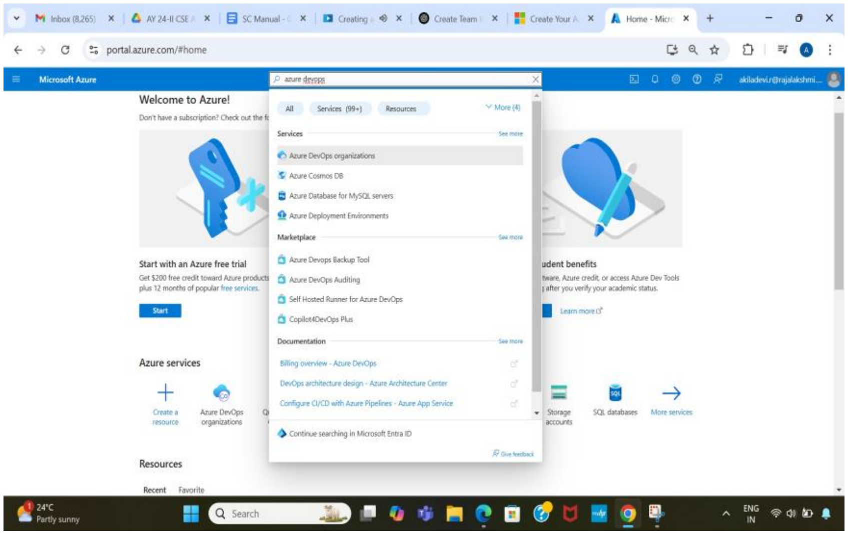
Don't have a subscription? Check out the following options.

Start with an Azure free trial
Get $200 free credit toward Azure products and services, plus 12 months of popular free services.

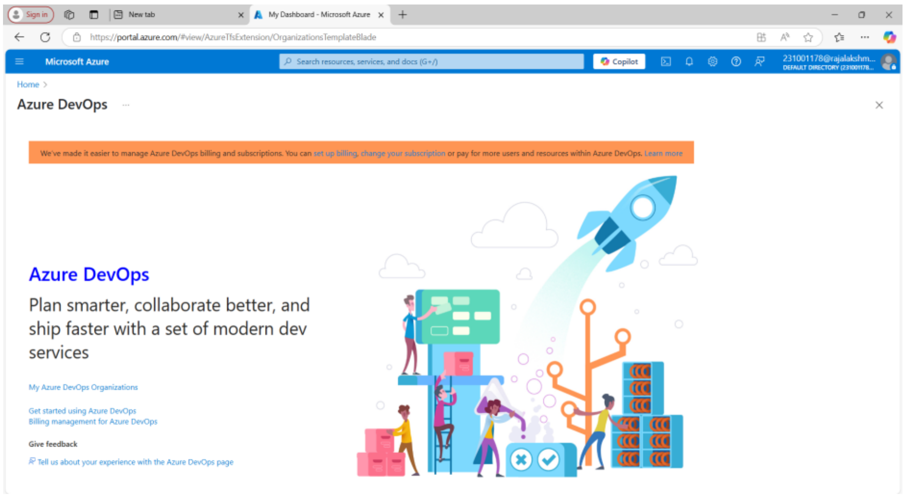
Manage Microsoft Entra ID
Manage access, set smart policies, and enhance security with Microsoft Entra ID.

Access student benefits
Get free software, Azure credit, or access Azure Dev Tools for Teaching after you verify your academic status.

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



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



https://portal.azure.com/#view/AzureTfsExtension/OrganizationsTemplateBlade

Home >

Azure DevOps

We've made it easier to manage Azure DevOps billing and subscriptions. You can set up billing, change your subscription or pay for more users and resources within Azure DevOps. Learn more

Azure DevOps

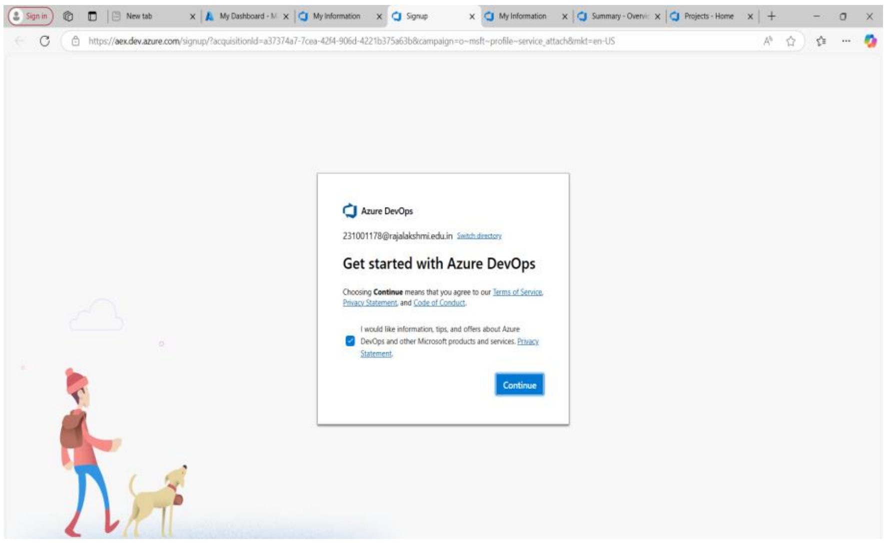
Plan smarter, collaborate better, and ship faster with a set of modern dev services

My Azure DevOps Organizations

Get started using Azure DevOps
Billing management for Azure DevOps

Give feedback

Tell us about your experience with the Azure DevOps page



https://aex.dev.azure.com/signup/?acquisitionId=a37374a7-7cea-42f4-906d-4221b375a63b&campaign=o~msft~profile~service_attach&mkt=en-US

Azure DevOps

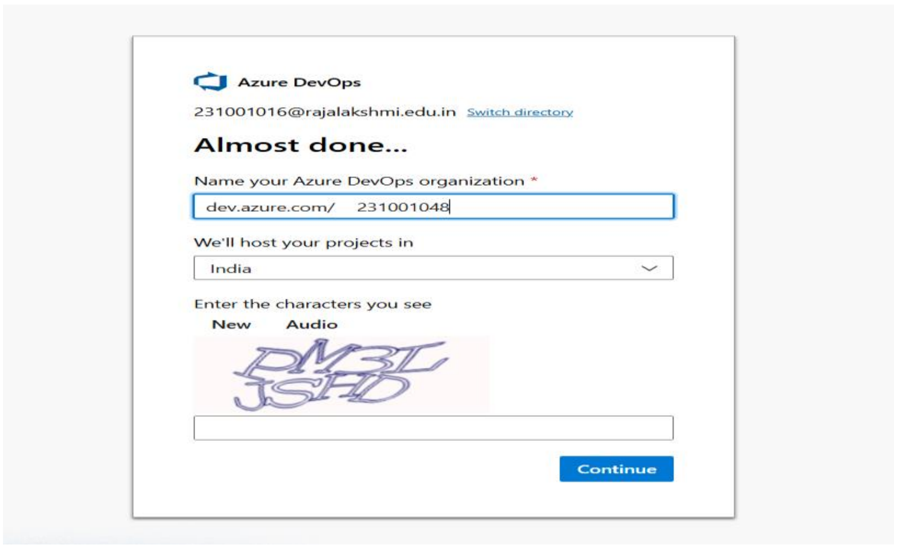
231001178@rajalakshmi.edu.in Switch directory

Get started with Azure DevOps

Choosing **Continue** means that you agree to our Terms of Service, Privacy Statement, and Code of Conduct.

[✓] I would like information, tips, and offers about Azure DevOps and other Microsoft products and services. Privacy Statement.

Continue



Azure DevOps

231001016@rajalakshmi.edu.in Switch directory

Almost done...

Name your Azure DevOps organization *

dev.azure.com/ 231001048

We'll host your projects in

India

Enter the characters you see

New Audio

Continue

1. Create the First Project in Your Organization

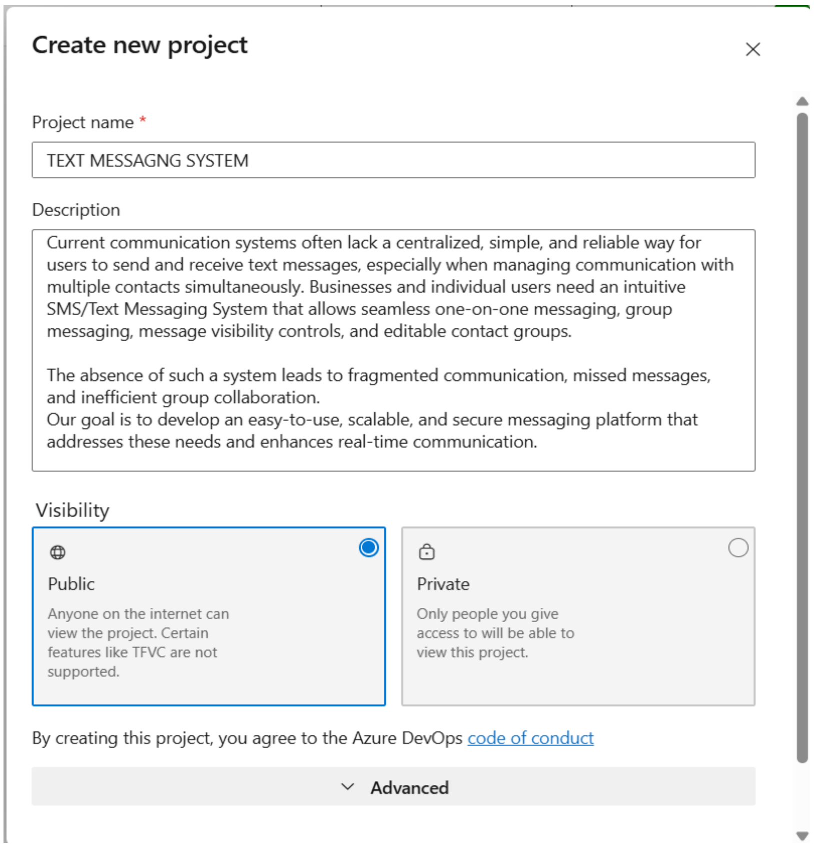
After the organization is set up, you'll need to create your first **project**. This is where you'll begin to manage code, pipelines, work items, and more.

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.


Create new project

Project name *
TEXT MESSAGNG SYSTEM

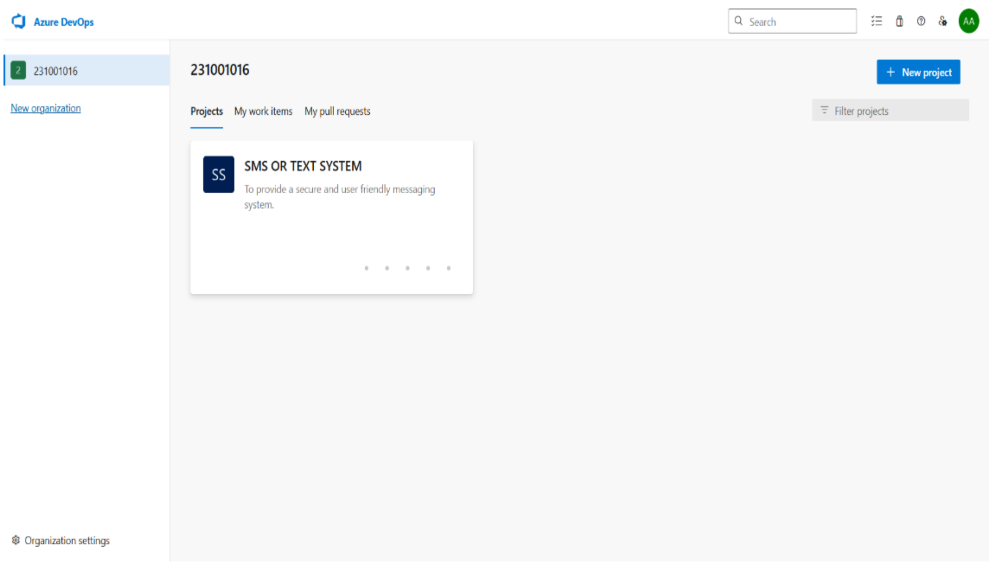
Description
Current communication systems often lack a centralized, simple, and reliable way for users to send and receive text messages, especially when managing communication with multiple contacts simultaneously. Businesses and individual users need an intuitive SMS/Text Messaging System that allows seamless one-on-one messaging, group messaging, message visibility controls, and editable contact groups.

The absence of such a system leads to fragmented communication, missed messages, and inefficient group collaboration.
Our goal is to develop an easy-to-use, scalable, and secure messaging platform that addresses these needs and enhances real-time communication.

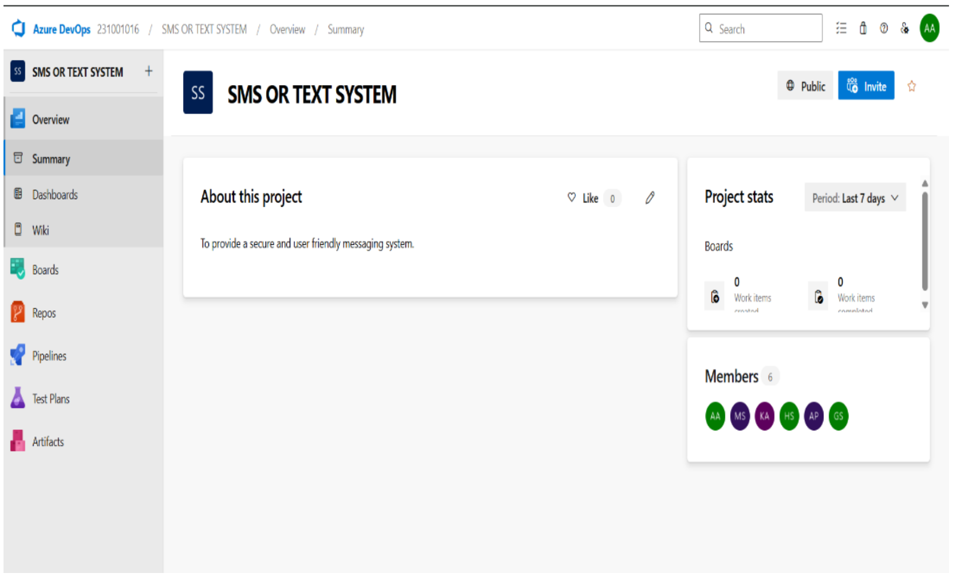
Visibility
[✓ Public] Anyone on the internet can view the project. Certain features like TFVC are not supported.
Private: Only people you give access to will be able to view this project.

By creating this project, you agree to the Azure DevOps code of conduct

2. 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.

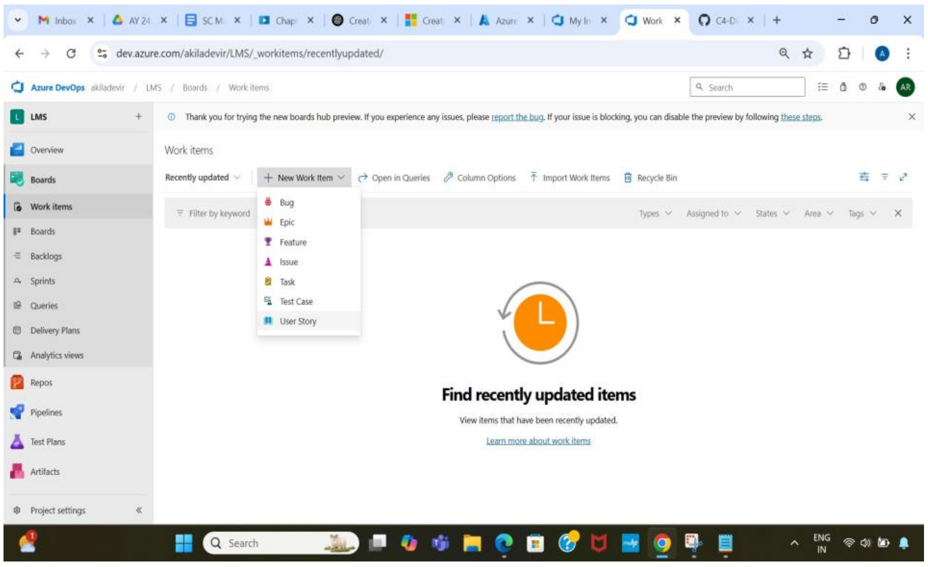


3. Project dashboard

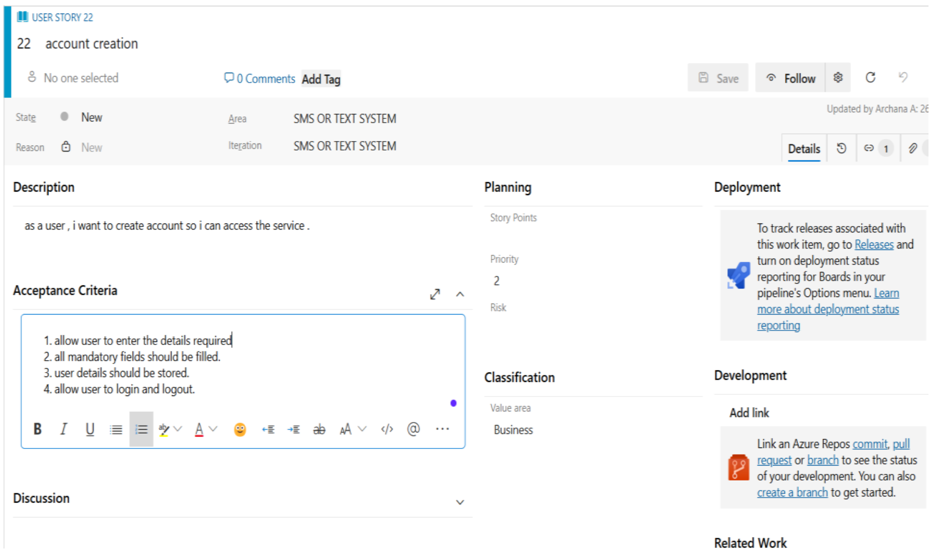


4.To manage user stories

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.


dev.azure.com/akiladevir/LMS/_workitems/recentlyupdated/
Azure DevOps akiladevir / LMS / Boards / Work items
LMS
Overview
Boards
Work items
Boards
Backlogs
Sprints
Queries
Delivery Plans
Analytics views
Repos
Pipelines
Test Plans
Artifacts
Project settings
Thank you for trying the new boards hub preview. If you experience any issues, please report the bug. If your issue is blocking, you can disable the preview by following these steps.
Work items
Recently updated
New Work Item
Open in Queries
Column Options
Import Work Items
Recycle Bin
Filter by keyword
Types
Assigned to
States
Area
Tags
Bug
Epic
Feature
Issue
Task
Test Case
User Story
Find recently updated items
View items that have been recently updated.
Learn more about work items

5.Fill in User Story Details


USER STORY 22

22 account creation

No one selected 0 Comments Add Tag Save Follow

Updated by Archana A: 26

State: New | Area: SMS OR TEXT SYSTEM

Reason: New | Iteration: SMS OR TEXT SYSTEM

Description

as a user , i want to create account so i can access the service .

Acceptance Criteria

1. allow user to enter the details required
2. all mandatory fields should be filled.
3. user details should be stored.
4. allow user to login and logout.

Discussion

Planning

Story Points:

Priority: 2

Risk:

Classification

Value area: Business

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

Result:

The user story was written successfully.

EX NO: 5 **DATE:25/02/2025**

<u>SEQUENCE DIAGRAM</u>

Aim:

To design a Sequence Diagram by using Mermaid.js

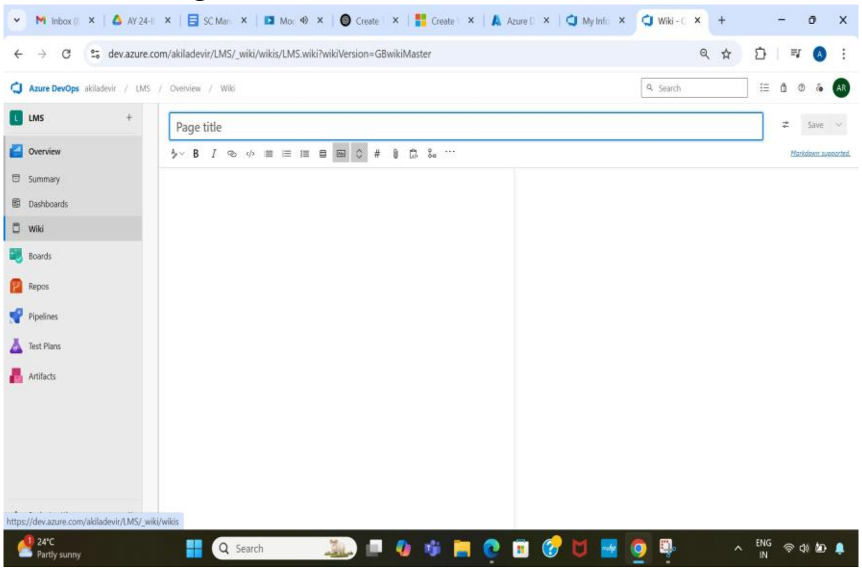
THEORY:

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


dev.azure.com/akiladevir/LMS/_wiki/wikis/LMS.wiki?wikiVersion=GBwikiMaster
Azure DevOps
akiladevir / LMS / Overview / Wiki
Search
LMS
Overview
Summary
Dashboards
Wiki
Boards
Repos
Pipelines
Test Plans
Artifacts
Page title
Save
Markdown supported.
https://dev.azure.com/akiladevir/LMS/_wiki/wikis
24°C
Partly sunny
Search
ENG
IN

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

```
sequenceDiagram
%% User Account Creation
participant User
participant AccountService
participant Database

User ->> AccountService: createAccount(name, phone, email, password)
AccountService ->> Database: storeUserData()
```

```
Database -->> AccountService: success
AccountService -->> User: accountCreated()

%% User Login
User ->> AccountService: login(phone, password)
AccountService ->> Database: validateCredentials()
Database -->> AccountService: validationResult()
alt valid credentials
AccountService -->> User: loginSuccess()
else invalid credentials
AccountService -->> User: loginFailed()
end

%% Forgot Password
User ->> AccountService: requestPasswordReset()
AccountService ->> Database: findUser()
Database -->> AccountService: userExists()
AccountService -->> User: sendResetLink()

%% One-to-One Messaging
User ->> MessagingService: sendMessage(receiver, content)
MessagingService ->> Database: storeMessage()
Database -->> MessagingService: messageStored()
MessagingService ->> SMSGateway: sendToReceiver()
SMSGateway -->> User: messageSent()

%% Group Messaging
User ->> GroupChatService: sendGroupMessage(groupId, content)
GroupChatService ->> Database: storeGroupMessage()
Database -->> GroupChatService: messageStored()
GroupChatService ->> GroupMembers: deliverMessage()

%% Message History & Sync
User ->> SyncManager: syncContacts()
SyncManager ->> Database: fetchContacts()
```

```
Database -->> SyncManager: contactList()
SyncManager -->> User: contactsSynced()

User ->> SyncManager: syncMessageHistory()
SyncManager ->> Database: fetchMessageHistory()
Database -->> SyncManager: messageHistory()
SyncManager -->> User: historySynced()

%% Trash & Deletion
User ->> TrashService: moveMessageToTrash(messageId)
TrashService ->> Database: updateMessageStatus()
Database -->> TrashService: updated
TrashService -->> User: messageMovedToTrash()

User ->> TrashService: restoreMessage(messageId)
TrashService ->> Database: updateMessageStatus()
Database -->> TrashService: restored
TrashService -->> User: messageRestored()

%% Notifications
MessagingService ->> NotificationService: triggerNotification(user)
NotificationService -->> User: displayNewMessageAlert()
```

Explanation:

participant defines the entities involved.

->> represents a direct message.

-->> represents a response message.

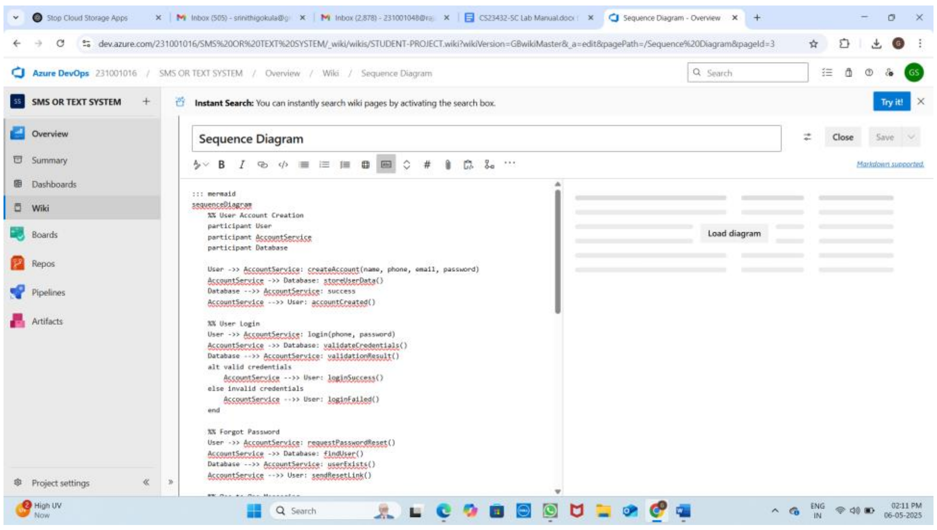
+ after ->> activates a participant.

- 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 a open arrow at the end (async)

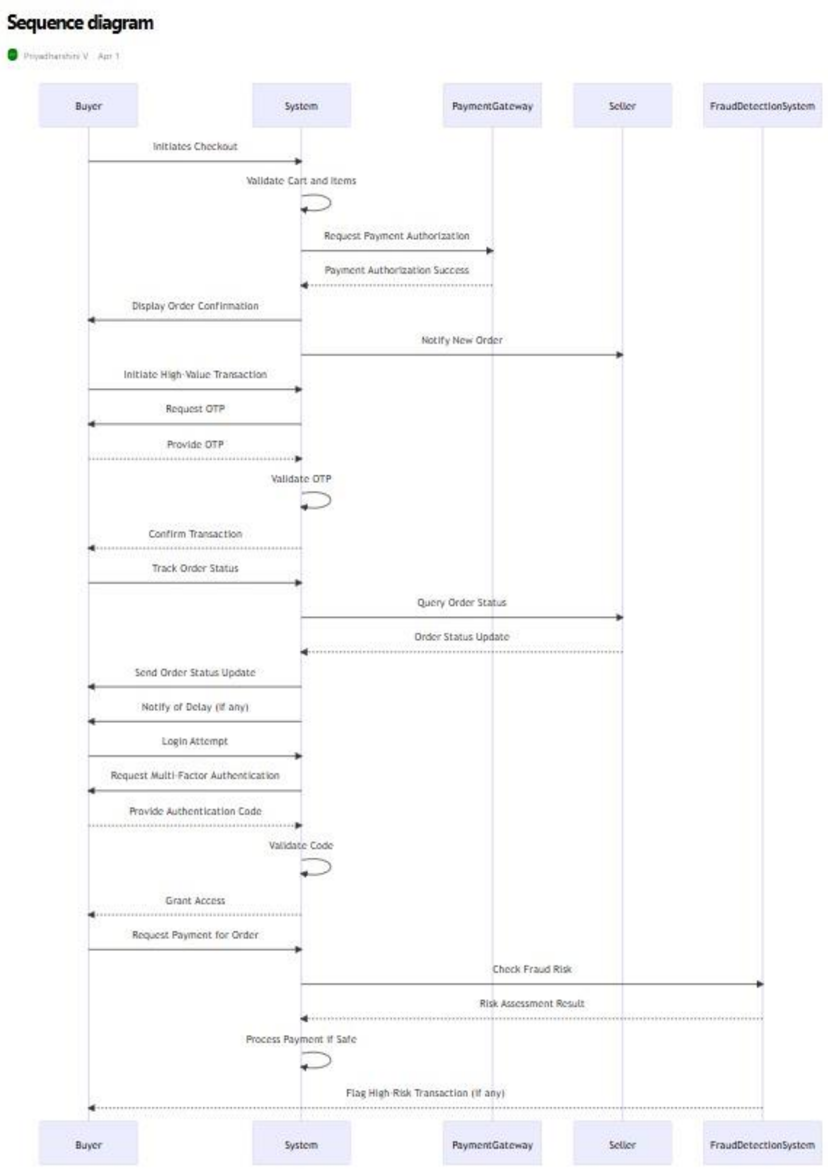


Instant Search: You can instantly search wiki pages by activating the search box.

Sequence Diagram

```
::: mermaid
sequenceDiagram
    %% User Account Creation
    participant User
    participant AccountService
    participant Database

    User ->> AccountService: createAccount(name, phone, email, password)
    AccountService ->> Database: storeUserData()
    Database -->> AccountService: success
    AccountService -->> User: accountCreated()

    %% User Login
    User ->> AccountService: login(phone, password)
    AccountService ->> Database: validateCredentials()
    Database -->> AccountService: validationResult()
    alt valid credentials
        AccountService -->> User: loginSuccess()
    else invalid credentials
        AccountService -->> User: loginFailed()
    end

    %% Forgot Password
    User ->> AccountService: requestPasswordReset()
    AccountService ->> Database: findUser()
    Database -->> AccountService: userExists()
    AccountService -->> User: sendResetLink()
```

4. click wiki menu and select the page



Result:

The sequence diagram was drawn successfully.

EX NO. 6 **DATE: 04/03/2025**

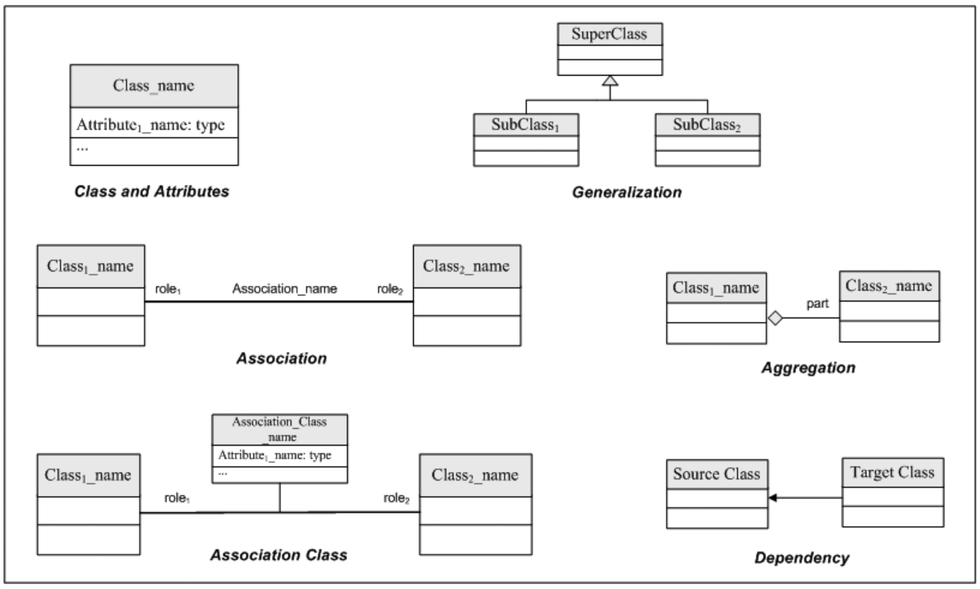
CLASS DIAGRAM

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.

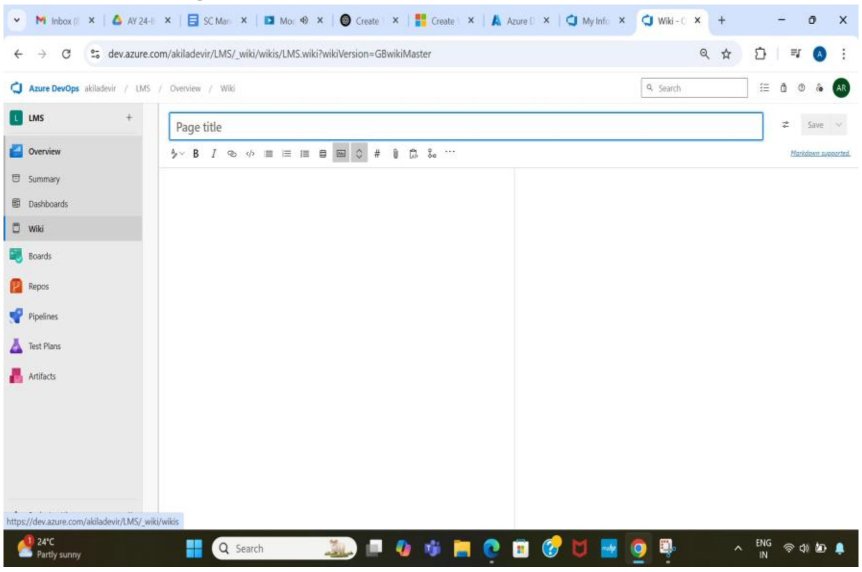


Notations in class diagram

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

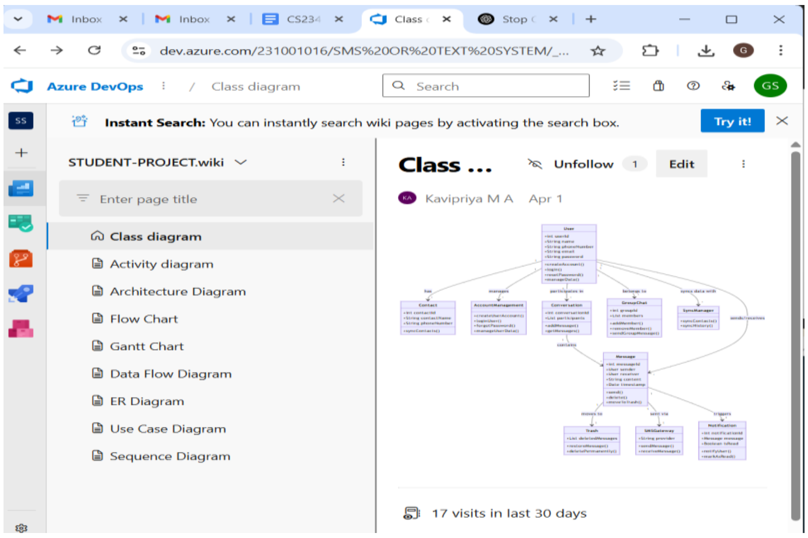
```
classDiagram

    %% User Management

    class User {

        +int userId

        +String name

        +String phoneNumber

        +String email

        +String password

        +createAccount()

        +login()

        +resetPassword()

        +manageData()

    }


    class Contact {

        +int contactId

        +String contactName
```

```
            +String phoneNumber
            +syncContacts()
        }

class AccountManagement {
    +createUserAccount()
    +loginUser()
    +forgotPassword()
    +manageUserData()
}

%% Messaging System
class Message {
    +int messageId
    +User sender
    +User receiver
    +String content
    +Date timestamp
    +send()
    +delete()
    +moveToTrash()
}

class Conversation {
    +int conversationId
    +List<User> participants
    +addMessage()
    +getMessages()
```

```
}

class GroupChat {
    +int groupId
    +List<User> members
    +addMember()
    +removeMember()
    +sendGroupMessage()
}

class Trash {
    +List<Message> deletedMessages
    +restoreMessage()
    +deletePermanently()
}

%% System Functions
class SMSGateway {
    +String provider
    +sendMessage()
    +receiveMessage()
}

class Notification {
    +int notificationId
    +Message message
    +Boolean isRead
    +notifyUser()
    +markAsRead()

}


class SyncManager {

    +syncContacts()

    +syncHistory()

}


%% Relationships

Buyer --> Cart : Manages

Cart --> Order : Creates

Order --> Payment : Makes

Order --> Shipment : Ships

Buyer --> User : Authenticates

Seller --> Product : Owns

Product --> Inventory : Manages

Seller --> Inventory : Updates

User --> FraudDetection : Uses

AutoScaling --> DatabaseOptimization : WorksWith
```

Relationship Types

Type	Description
<|	Inheritance
*	Composition
o	Aggregation
>	Association
<	Association
|>	Realization



Result:

The use case diagram was designed successfully.

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

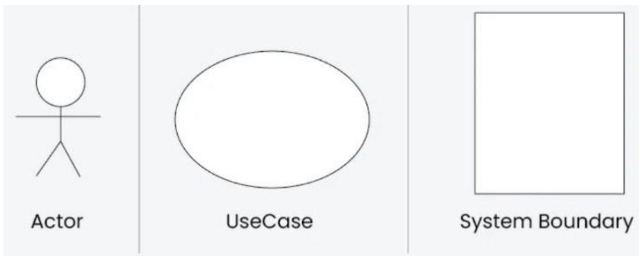
USECASE DIAGRAM

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

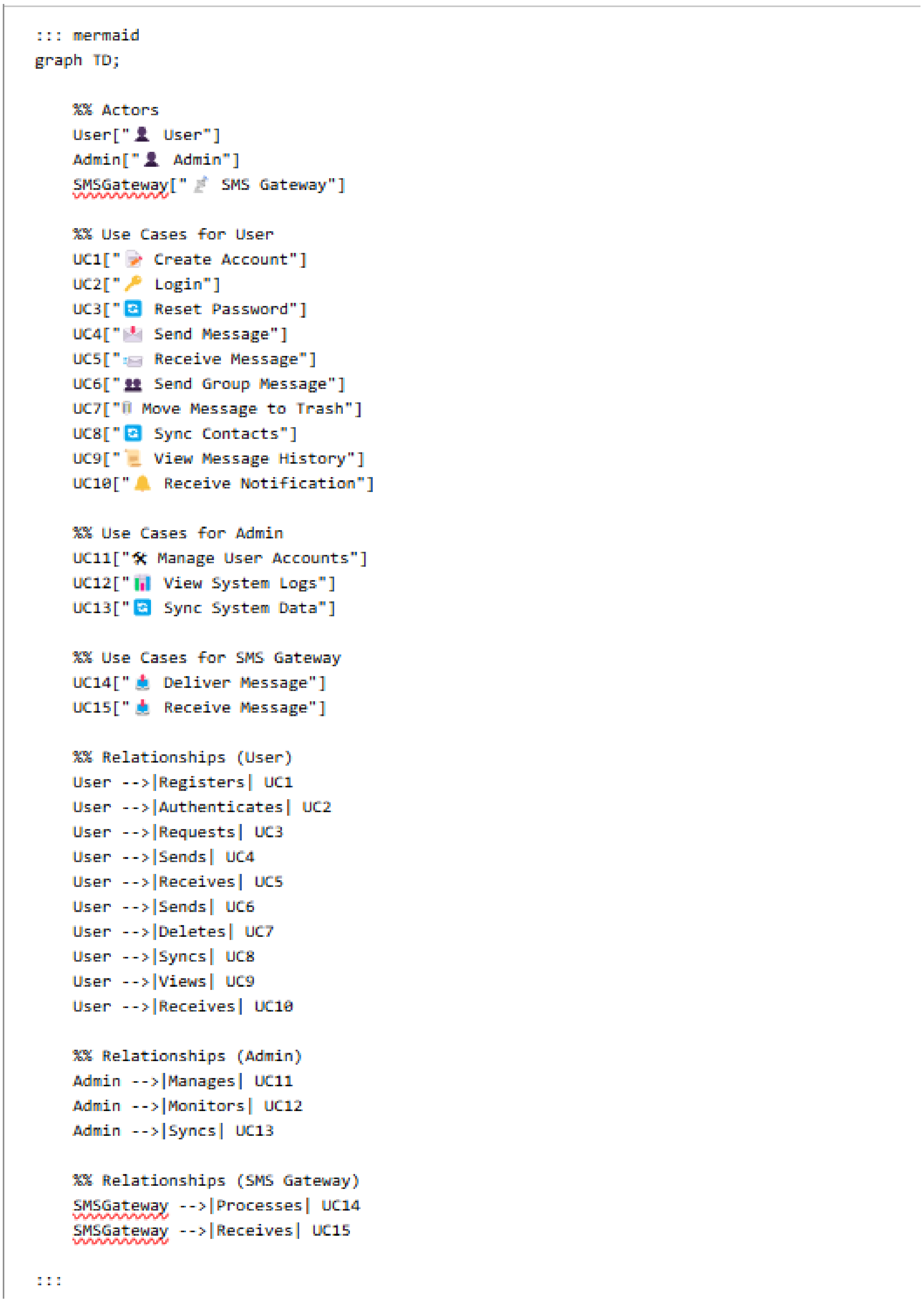
Option 1: Add to Azure DevOps Wiki

- 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.

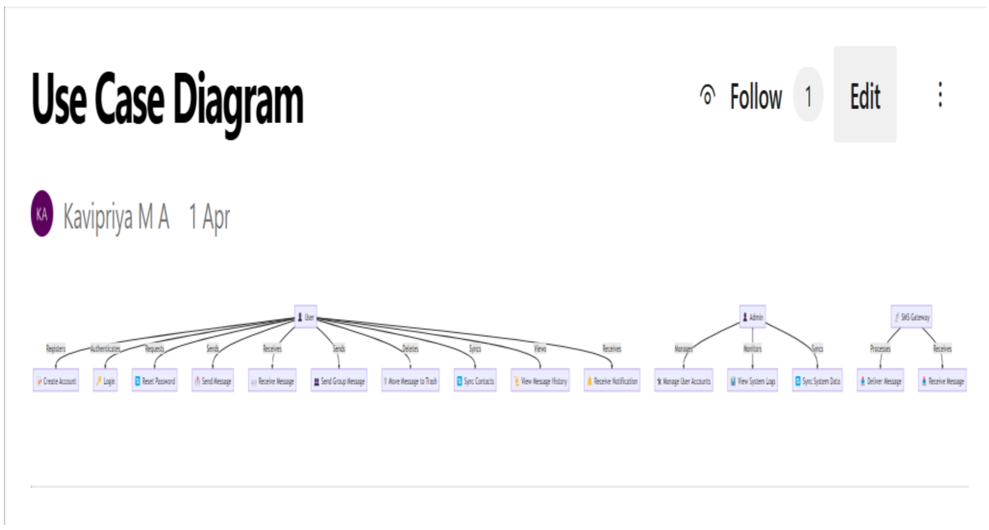
- Use Markdown to embed the diagram:
- ![Use Case Diagram](attachments/use_case_diagram.png)

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.



```
::: mermaid
graph TD;

    %% Actors
    User["👤 User"]
    Admin["👤 Admin"]
    SMSGateway["📡 SMS Gateway"]

    %% Use Cases for User
    UC1["📝 Create Account"]
    UC2["🔑 Login"]
    UC3["🔄 Reset Password"]
    UC4["📤 Send Message"]
    UC5["📨 Receive Message"]
    UC6["👥 Send Group Message"]
    UC7["🗑 Move Message to Trash"]
    UC8["🔄 Sync Contacts"]
    UC9["📜 View Message History"]
    UC10["🔔 Receive Notification"]

    %% Use Cases for Admin
    UC11["🛠 Manage User Accounts"]
    UC12["📊 View System Logs"]
    UC13["🔄 Sync System Data"]

    %% Use Cases for SMS Gateway
    UC14["📥 Deliver Message"]
    UC15["📥 Receive Message"]

    %% Relationships (User)
    User -->|Registers| UC1
    User -->|Authenticates| UC2
    User -->|Requests| UC3
    User -->|Sends| UC4
    User -->|Receives| UC5
    User -->|Sends| UC6
    User -->|Deletes| UC7
    User -->|Syncs| UC8
    User -->|Views| UC9
    User -->|Receives| UC10

    %% Relationships (Admin)
    Admin -->|Manages| UC11
    Admin -->|Monitors| UC12
    Admin -->|Syncs| UC13

    %% Relationships (SMS Gateway)
    SMSGateway -->|Processes| UC14
    SMSGateway -->|Receives| UC15

:::
```



Result:

The use case diagram was designed successfully

EX NO. 8 **DATE:18/03/2025**

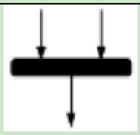
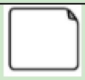
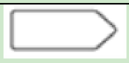
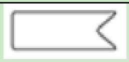
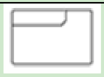
ACTIVITY DIAGRAM

AIM :-

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

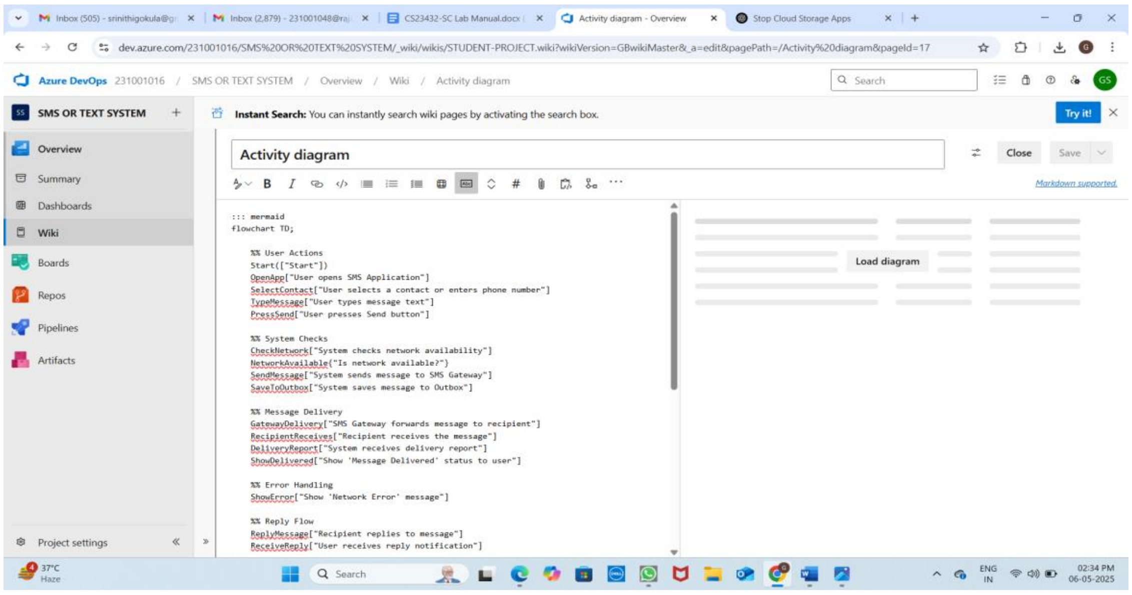
THEORY

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.

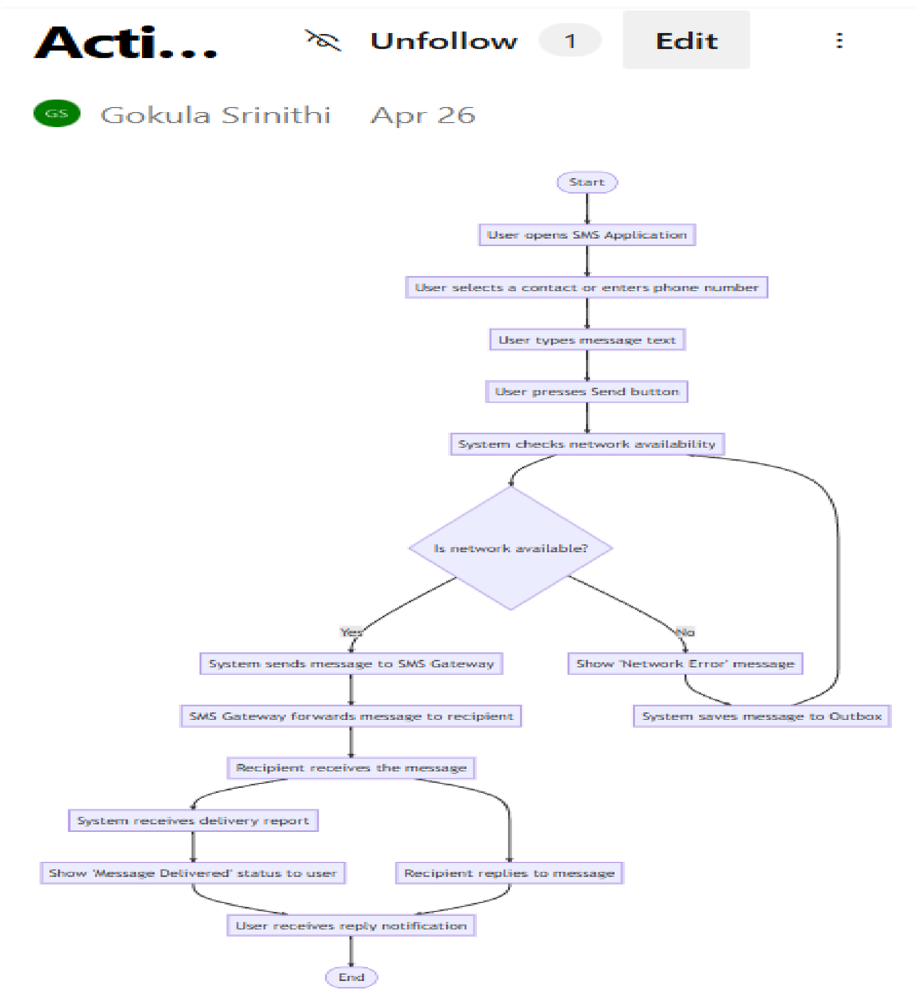
Notations	**Symbol**	**Meaning**
Start	⬬	Shows the beginning of a process
Connector	→	Shows the directional flow, or control flow, of the activity
Joint symbol		Combines two concurrent activities and re-introduces them to a flow where one activity occurs at a time
Decision	◇	Represents a decision
Note		Allows the diagram creators o 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

Procedure

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



```
::: mermaid
flowchart TD;

    %% User Actions
    Start(["Start"])
    OpenApp["User opens SMS Application"]
    SelectContact["User selects a contact or enters phone number"]
    TypeMessage["User types message text"]
    PressSend["User presses Send button"]

    %% System Checks
    CheckNetwork["System checks network availability"]
    NetworkAvailable{"Is network available?"}
    SendMessage["System sends message to SMS Gateway"]
    SaveToOutbox["System saves message to Outbox"]

    %% Message Delivery
    GatewayDelivery["SMS Gateway forwards message to recipient"]
    RecipientReceives["Recipient receives the message"]
    DeliveryReport["System receives delivery report"]
    ShowDelivered["Show 'Message Delivered' status to user"]

    %% Error Handling
    ShowError["Show 'Network Error' message"]

    %% Reply Flow
    ReplyMessage["Recipient replies to message"]
    ReceiveReply["User receives reply notification"]
```


Acti...
Unfollow
1
Edit
GS
Gokula Srinithi
Apr 26
Start
User opens SMS Application
User selects a contact or enters phone number
User types message text
User presses Send button
System checks network availability
Is network available?
Yes
No
System sends message to SMS Gateway
Show 'Network Error' message
SMS Gateway forwards message to recipient
System saves message to Outbox
Recipient receives the message
System receives delivery report
Show 'Message Delivered' status to user
Recipient replies to message
User receives reply notification
End

Result:

The activity diagram was designed successfully

EX NO. 9 **DATE: 25/03/2025**

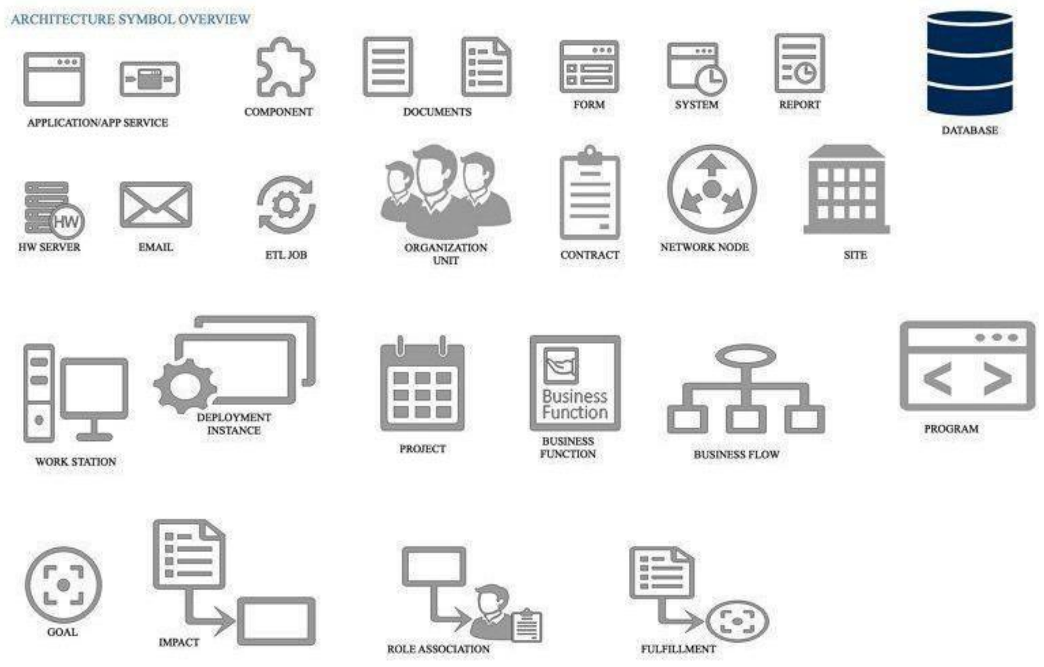
ARCHITECTURE DIAGRAM

Aim:

Steps to draw the Architecture Diagram using draw.io.

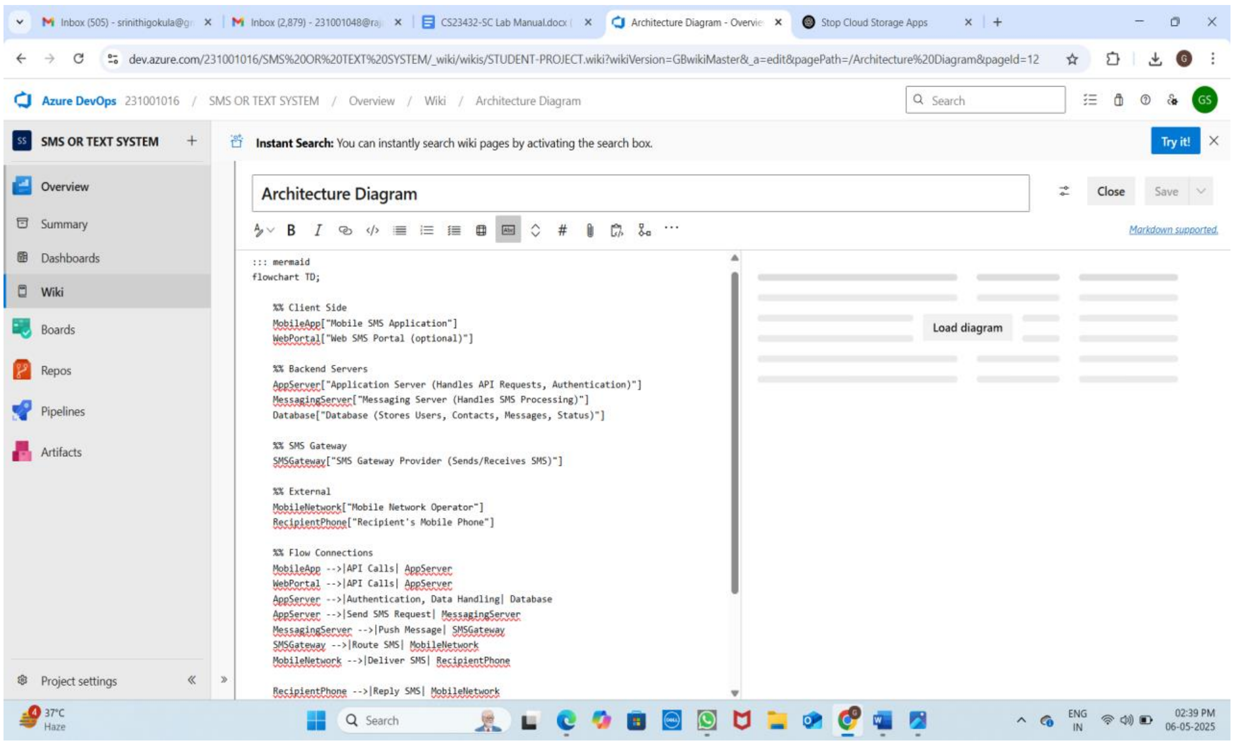
Theory:

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.



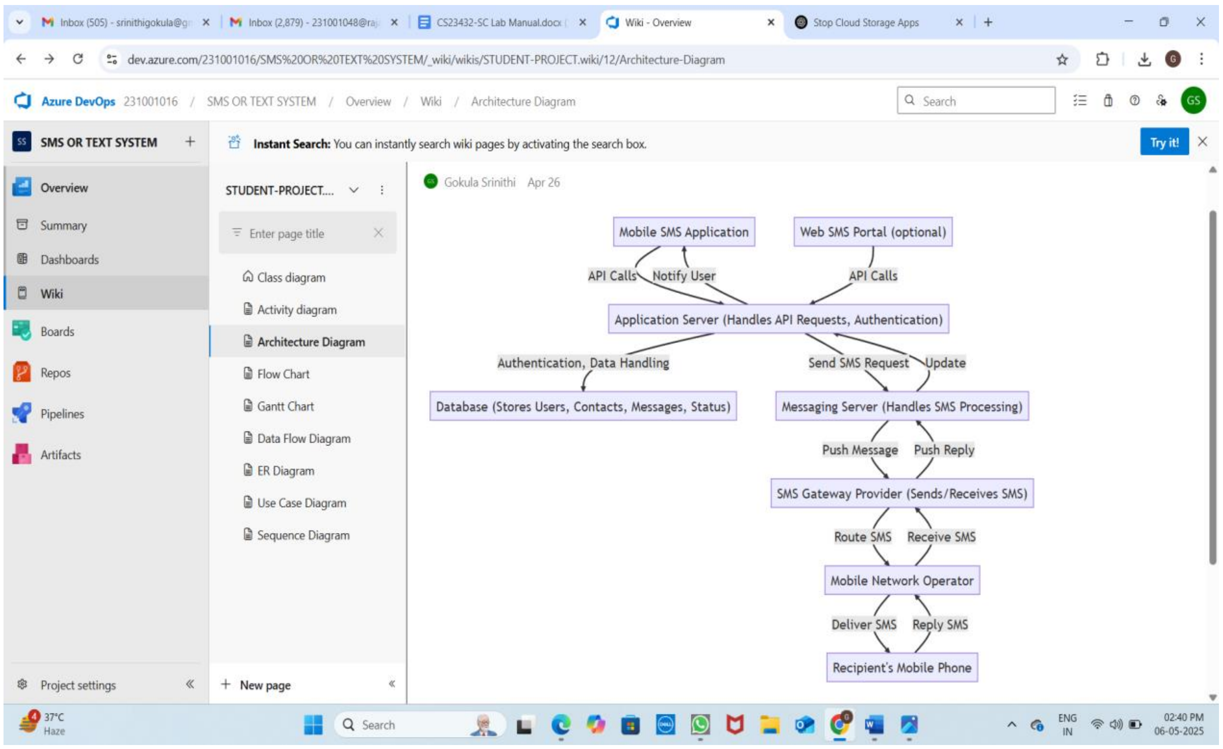
Procedure

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



Architecture Diagram

```
::: mermaid
flowchart TD;

    %% Client Side
    MobileApp["Mobile SMS Application"]
    WebPortal["Web SMS Portal (optional)"]

    %% Backend Servers
    AppServer["Application Server (Handles API Requests, Authentication)"]
    MessagingServer["Messaging Server (Handles SMS Processing)"]
    Database["Database (Stores Users, Contacts, Messages, Status)"]

    %% SMS Gateway
    SMSGateway["SMS Gateway Provider (Sends/Receives SMS)"]

    %% External
    MobileNetwork["Mobile Network Operator"]
    RecipientPhone["Recipient's Mobile Phone"]

    %% Flow Connections
    MobileApp -->|API Calls| AppServer
    WebPortal -->|API Calls| AppServer
    AppServer -->|Authentication, Data Handling| Database
    AppServer -->|Send SMS Request| MessagingServer
    MessagingServer -->|Push Message| SMSGateway
    SMSGateway -->|Route SMS| MobileNetwork
    MobileNetwork -->|Deliver SMS| RecipientPhone

    RecipientPhone -->|Reply SMS| MobileNetwork
```


Gokula Srinithi Apr 26
Mobile SMS Application
Web SMS Portal (optional)
API Calls
Notify User
API Calls
Application Server (Handles API Requests, Authentication)
Authentication, Data Handling
Send SMS Request
Update
Database (Stores Users, Contacts, Messages, Status)
Messaging Server (Handles SMS Processing)
Push Message
Push Reply
SMS Gateway Provider (Sends/Receives SMS)
Route SMS
Receive SMS
Mobile Network Operator
Deliver SMS
Reply SMS
Recipient's Mobile Phone

Result:

The architecture diagram was designed successfully

EX NO. 10 DATE: 01/04/2025

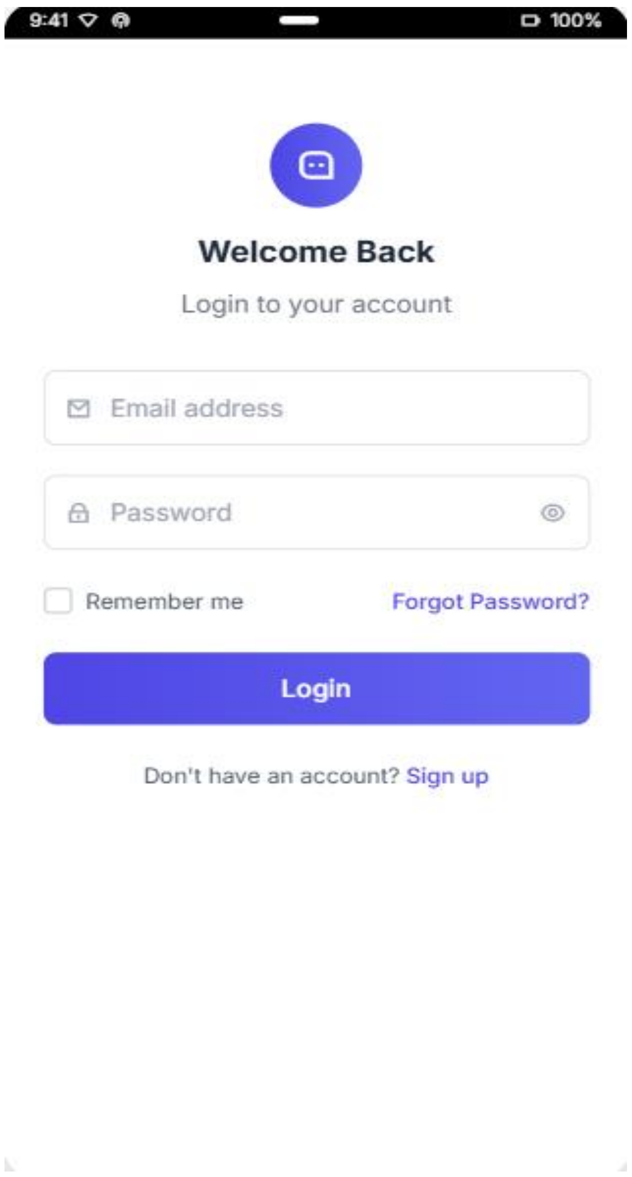
USER INTERFACE

Aim:

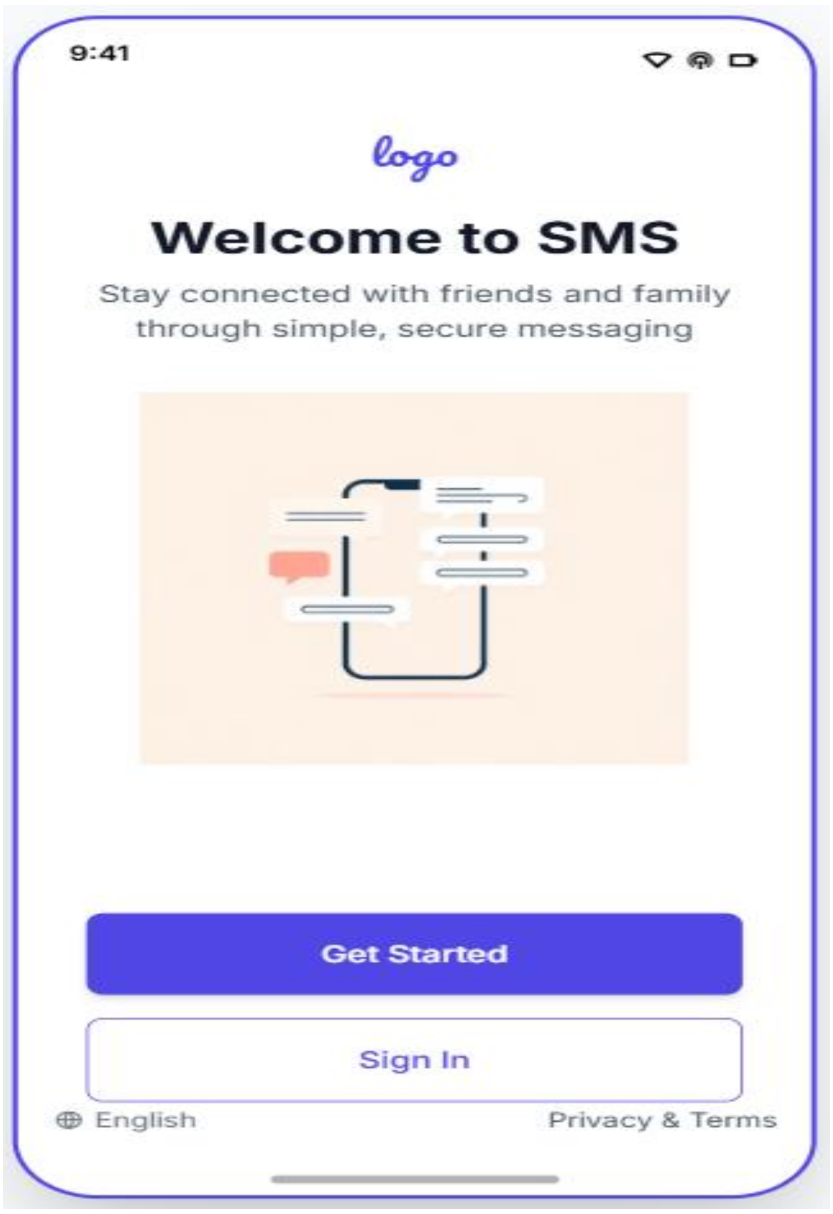
Design User Interface for the given project

PROCEDURE:

1. **Login page :**



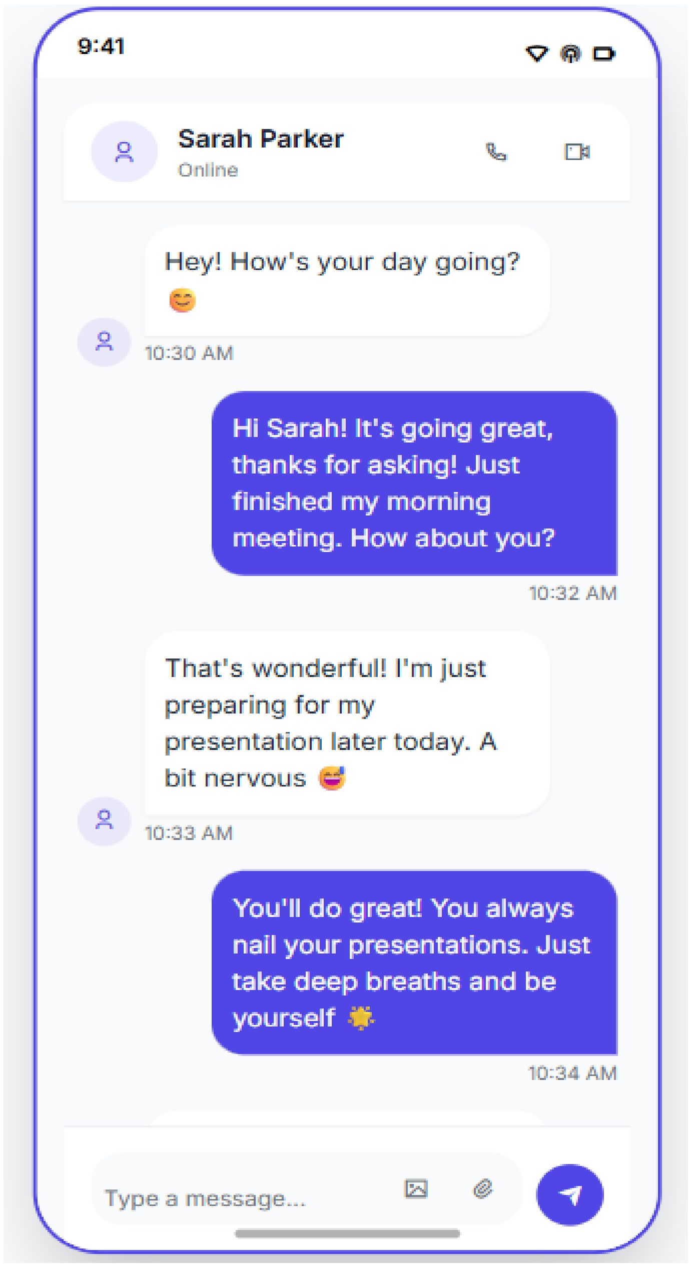
2 . LANDING PAGE:



Welcome to SMS

Stay connected with friends and family through simple, secure messaging

3.CHAT PAGE:



Hey! How's your day going? 😊

10:30 AM

Hi Sarah! It's going great, thanks for asking! Just finished my morning meeting. How about you?

10:32 AM

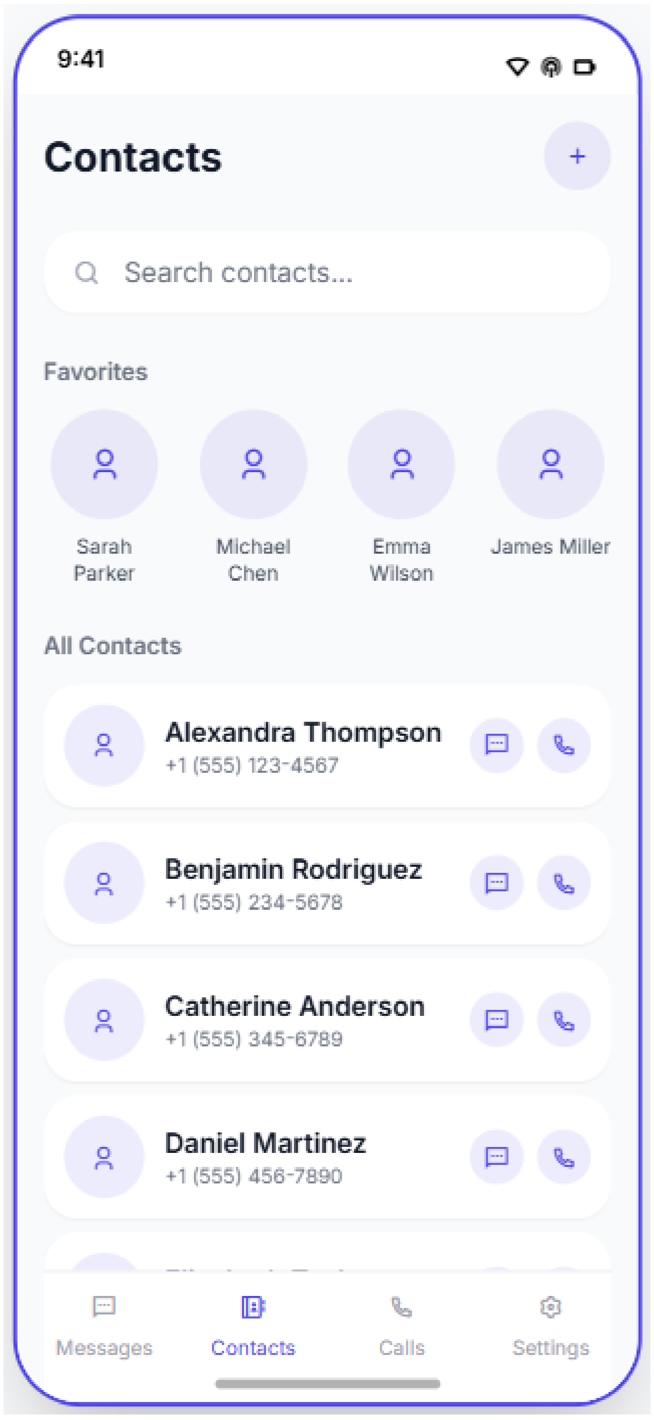
That's wonderful! I'm just preparing for my presentation later today. A bit nervous 😅

10:33 AM

You'll do great! You always nail your presentations. Just take deep breaths and be yourself 🌟

10:34 AM

4.SYNCING CONTACTS :



9:41

Contacts

Search contacts...

Favorites

Sarah Parker

Michael Chen

Emma Wilson

James Miller

All Contacts

Alexandra Thompson
+1 (555) 123-4567

Benjamin Rodriguez
+1 (555) 234-5678

Catherine Anderson
+1 (555) 345-6789

Daniel Martinez
+1 (555) 456-7890

Result:

The UI was designed successfully.

EX NO. 11 **DATE: 08/04/2025**

IMPLEMENTATION

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.

EXP NO:12 **DATE: 15/04/2025**

TESTING – TEST PLANS AND TEST CASES

Aim:

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

Test Planning and Test Case

Test Case Design Procedure

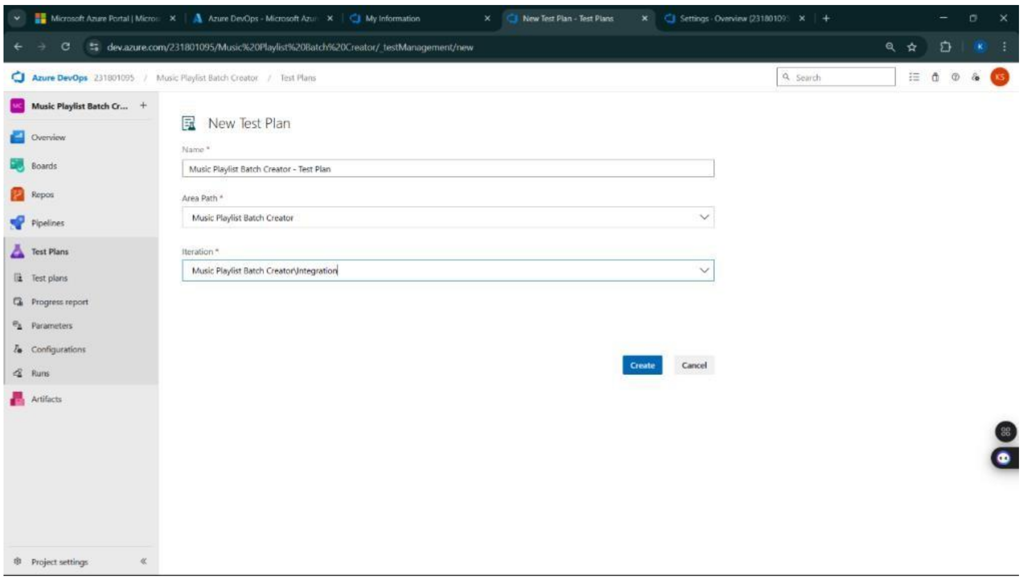
1. **Understand Core Features of the Application**
    - User Signup & Login
    - Viewing and Managing Playlists
    - Fetching Real-time Metadata
    - Editing playlists (rename, reorder, record)
    - Creating smart audio playlists based on categories (mood, genre, artist, etc.)
2. **Define User Interactions**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

- Grouped test cases based on functionality (e.g., Login, Playlist Editing, Recommendation System).
- Improves organization and test execution flow in Azure DevOps.

8. Prioritize and Review

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

1.New test plan



dev.azure.com/231801095/Music%20Playlist%20Batch%20Creator/_testManagement/new

Azure DevOps 231801095 / Music Playlist Batch Creator / Test Plans

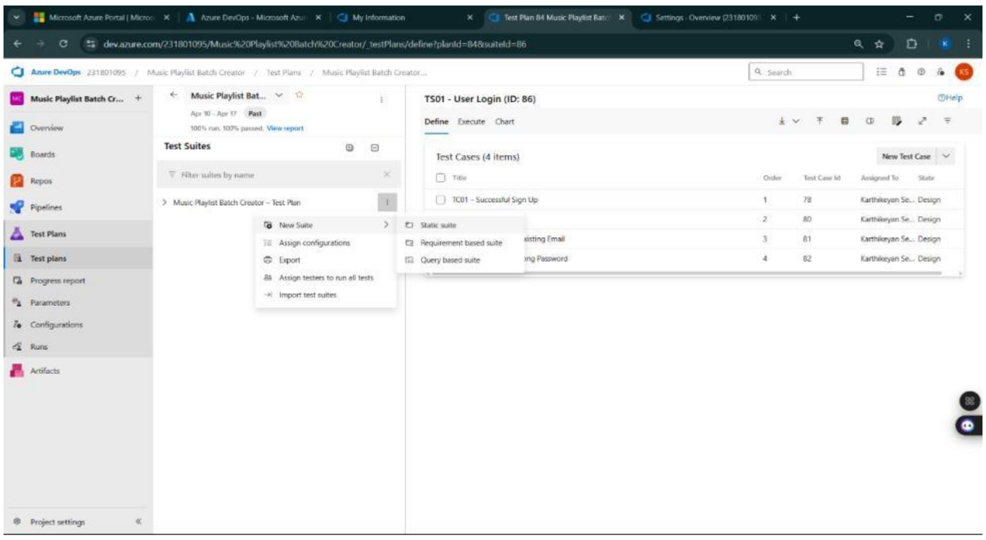
New Test Plan

Name *
Music Playlist Batch Creator - Test Plan

Area Path *
Music Playlist Batch Creator

Iteration *
Music Playlist Batch Creator\Integration

2.Test suite



dev.azure.com/231801095/Music%20Playlist%20Batch%20Creator/_testPlans/define?planId=84&suiteId=86

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

Music Playlist Bat...
Apr 10 - Apr 17 Past
100% run, 100% passed. View report

Test Suites
Music Playlist Batch Creator – Test Plan

TS01 - User Login (ID: 86)

Test Cases (4 items)

Title	Order	Test Case Id	Assigned To	State
TC01 – Successful Sign Up	1	78	Karthikeyan Se...	Design
	2	80	Karthikeyan Se...	Design
xisting Email	3	81	Karthikeyan Se...	Design
ng Password	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 ○

   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)

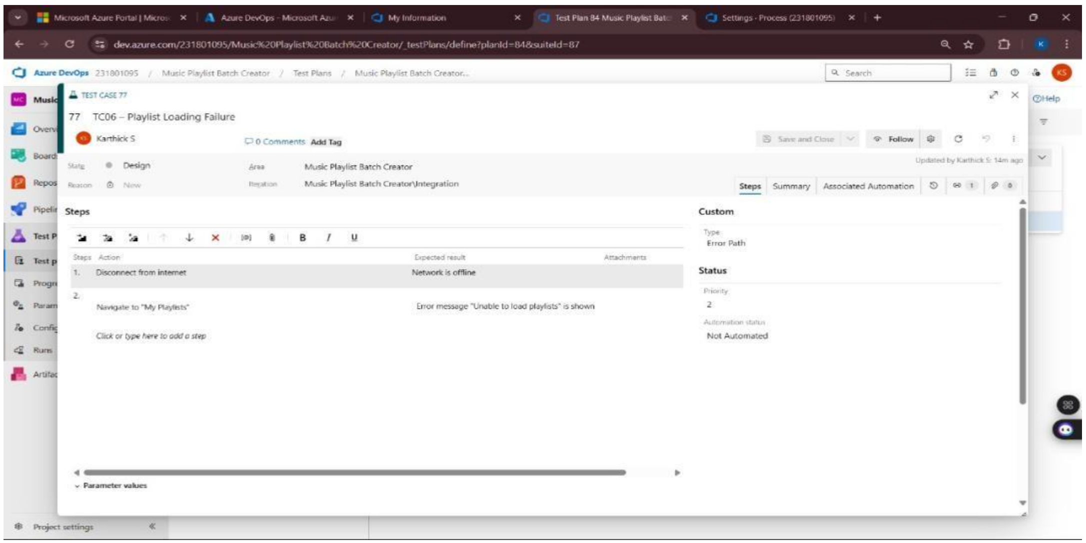
1. **TC13 – Generate Playlist Based on Various Categories** ○

   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


dev.azure.com/231801095/Music%20Playlist%20Batch%20Creator/_testPlans/define?planId=84&suiteId=87

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

TEST CASE 77

77 TC06 – Playlist Loading Failure

Karthick S

State: Design
Reason: New
Area: Music Playlist Batch Creator
Iteration: Music Playlist Batch Creator\Integration

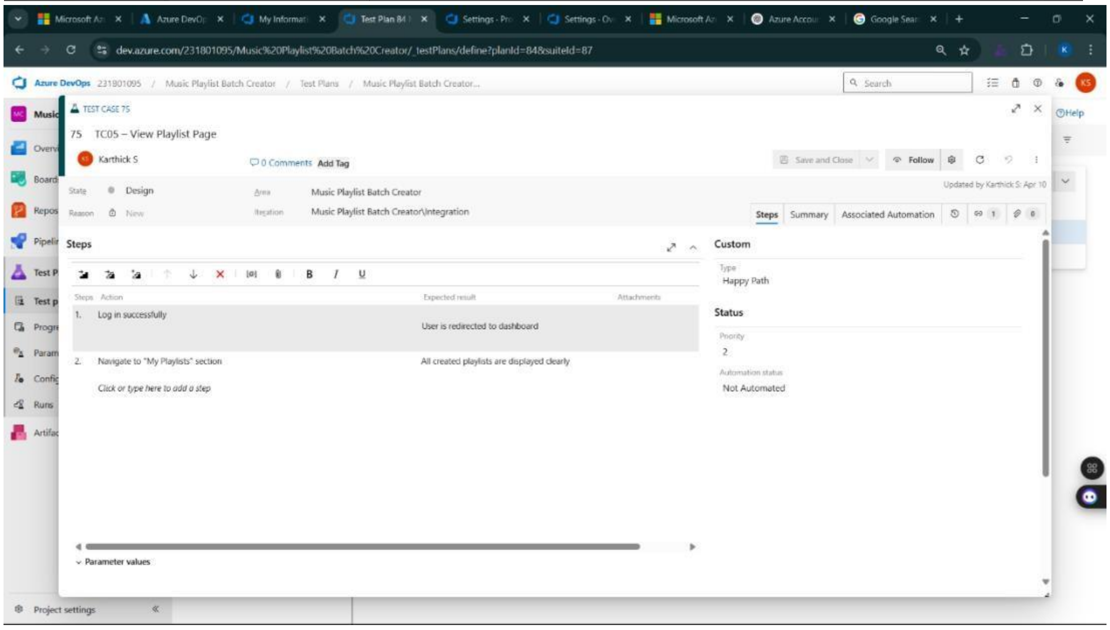
Updated by Karthick S: 14m ago

Steps

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

Custom
Type: Error Path

Status
Priority: 2
Automation status: Not Automated


dev.azure.com/231801095/Music%20Playlist%20Batch%20Creator/_testPlans/define?planId=84&suiteId=87

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

TEST CASE 75

75 TC05 – View Playlist Page

Karthick S

State: Design
Reason: New
Area: Music Playlist Batch Creator
Iteration: Music Playlist Batch Creator\Integration

Updated by Karthick S: Apr 10

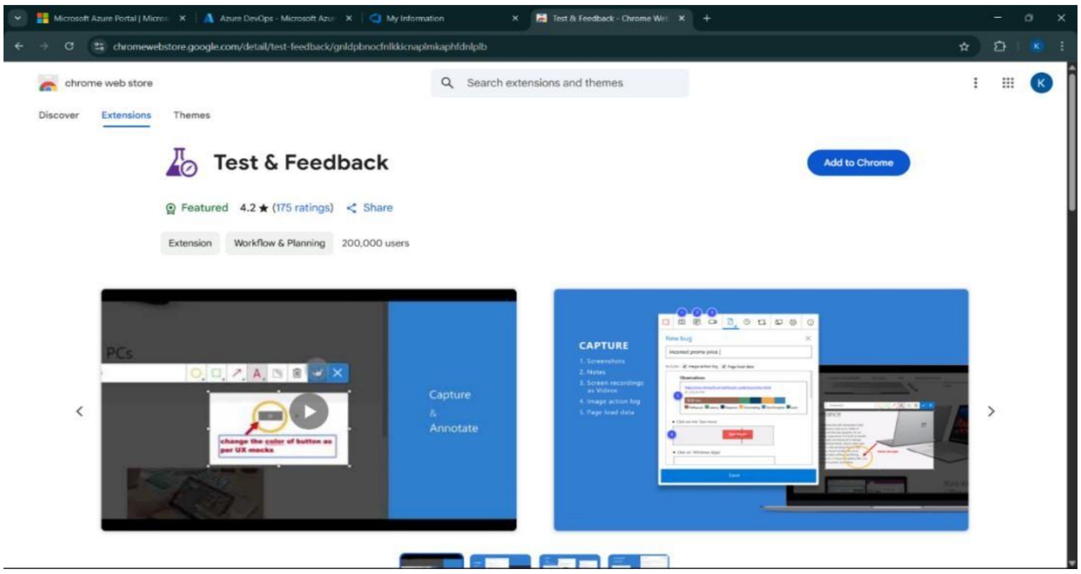
Steps

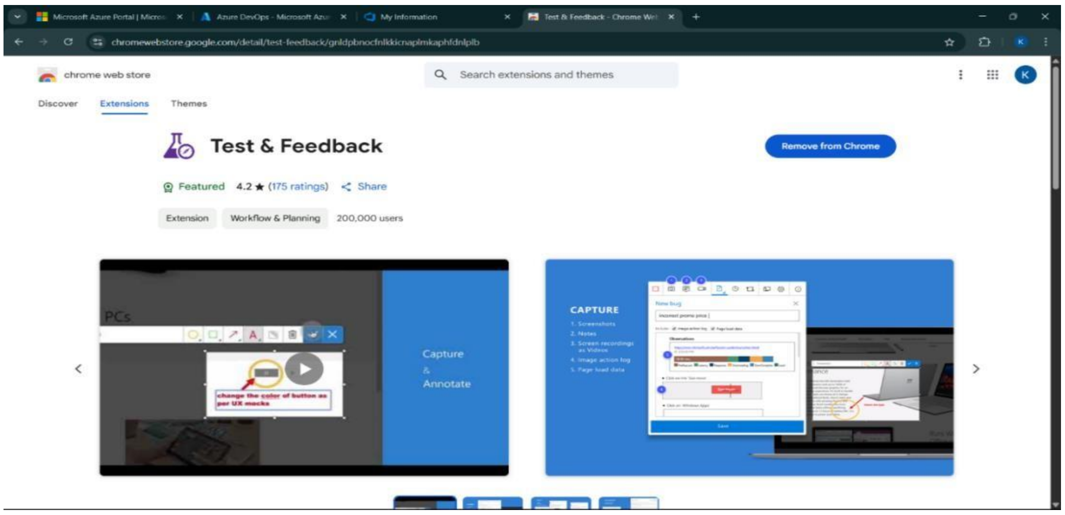
Steps	Action	Expected result	Attachments
1.	Log in successfully	User is redirected to dashboard	
2.	Navigate to "My Playlists" section	All created playlists are displayed clearly	

Custom
Type: Happy Path

Status
Priority: 2
Automation status: Not Automated

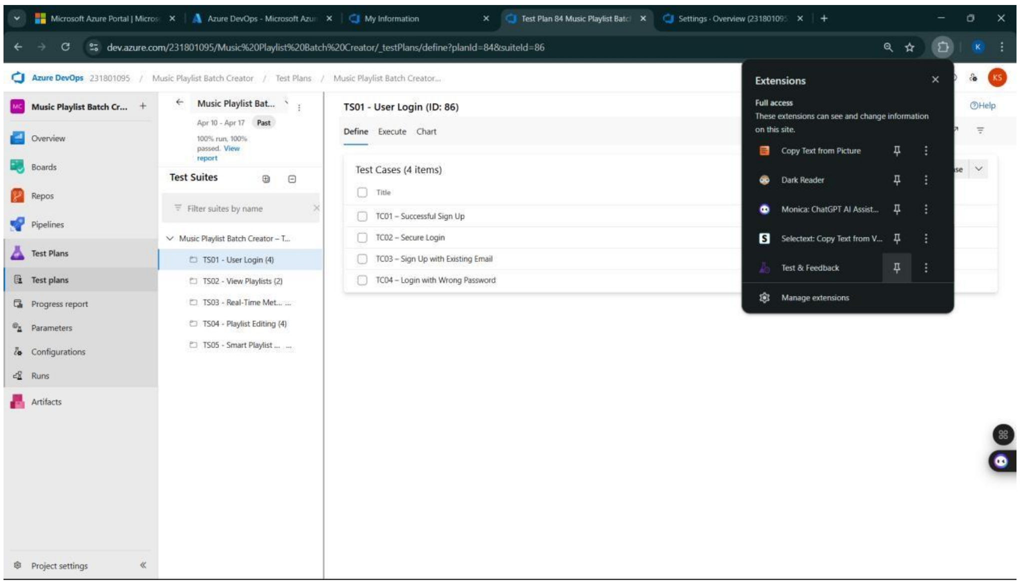
4.Installation of test


chromewebstore.google.com/detail/test-feedback/gnldpbnocfnlkkicnaplmkaphfdnlplb
chrome web store
Search extensions and themes
Discover
Extensions
Themes
Test & Feedback
Add to Chrome
Featured 4.2 ★ (175 ratings) Share
Extension
Workflow & Planning
200,000 users
Capture & Annotate
CAPTURE

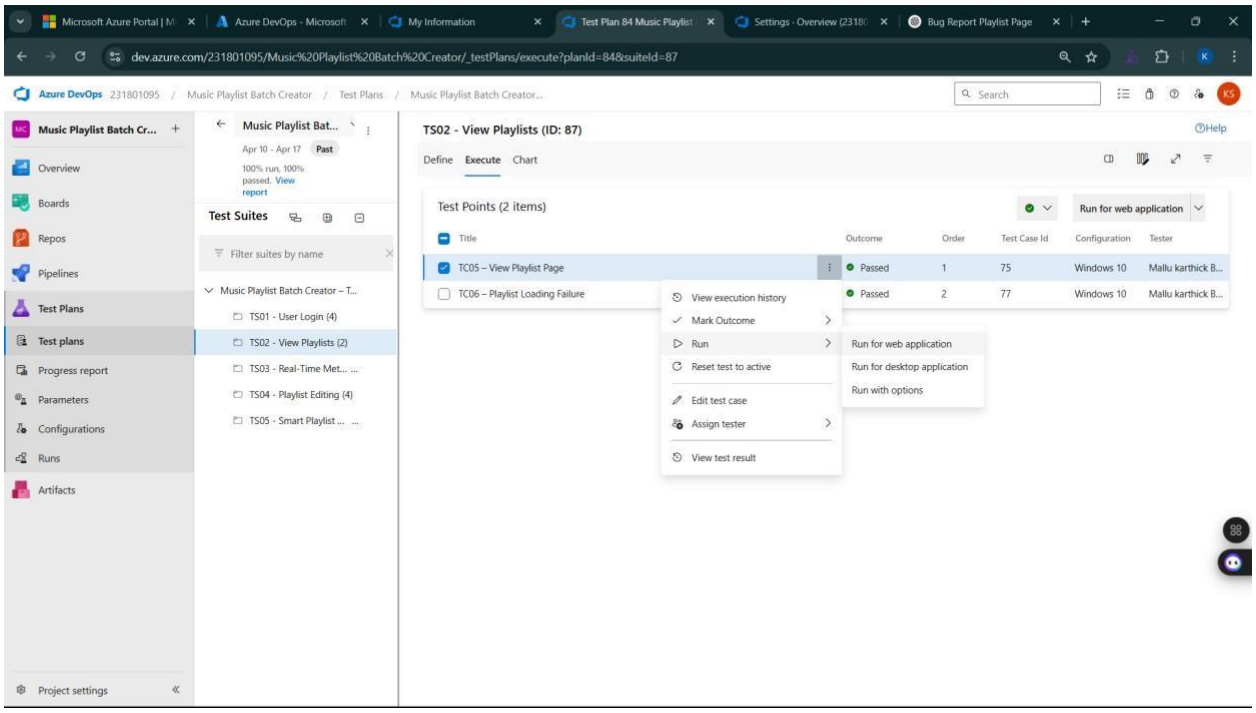

chromewebstore.google.com/detail/test-feedback/gnldpbnocfnlkkicnaplmkaphfdnlplb
chrome web store
Search extensions and themes
Discover
Extensions
Themes
Test & Feedback
Remove from Chrome
Featured 4.2 ★ (175 ratings) Share
Extension
Workflow & Planning
200,000 users
Capture & Annotate
CAPTURE

Test and feedback

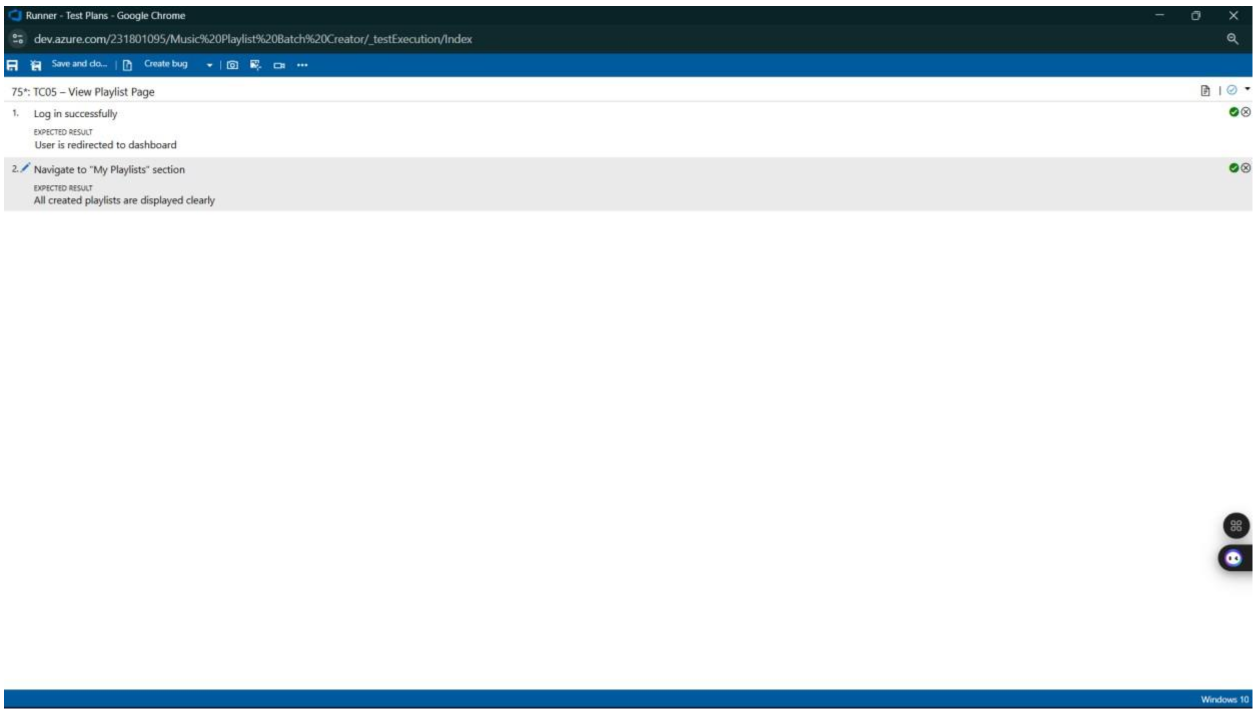
Showing it as an extension


dev.azure.com/231801095/Music%20Playlist%20Batch%20Creator/_testPlans/define?planId=84&suiteId=86
Azure DevOps 231801095 / Music Playlist Batch Creator / Test Plans / Music Playlist Batch Creator...
Music Playlist Batch Cr...
Overview
Boards
Repos
Pipelines
Test Plans
Test plans
Progress report
Parameters
Configurations
Runs
Artifacts
Project settings
Test Suites
Filter suites by name
Music Playlist Batch Creator – T...
TS01 - User Login (4)
TS02 - View Playlists (2)
TS03 - Real-Time Met...
TS04 - Playlist Editing (4)
TS05 - Smart Playlist ...
TS01 - User Login (ID: 86)
Define
Execute
Chart
Test Cases (4 items)
Title
TC01 – Successful Sign Up
TC02 – Secure Login
TC03 – Sign Up with Existing Email
TC04 – Login with Wrong Password
Extensions
Full access
These extensions can see and change information on this site.
Copy Text from Picture
Dark Reader
Monica: ChatGPT AI Assist...
Selectext: Copy Text from V...
Test & Feedback
Manage extensions

5.Running the test cases



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



Runner - Test Plans - Google Chrome

dev.azure.com/231801095/Music%20Playlist%20Batch%20Creator/_testExecution/Index

75*: TC05 – View Playlist Page

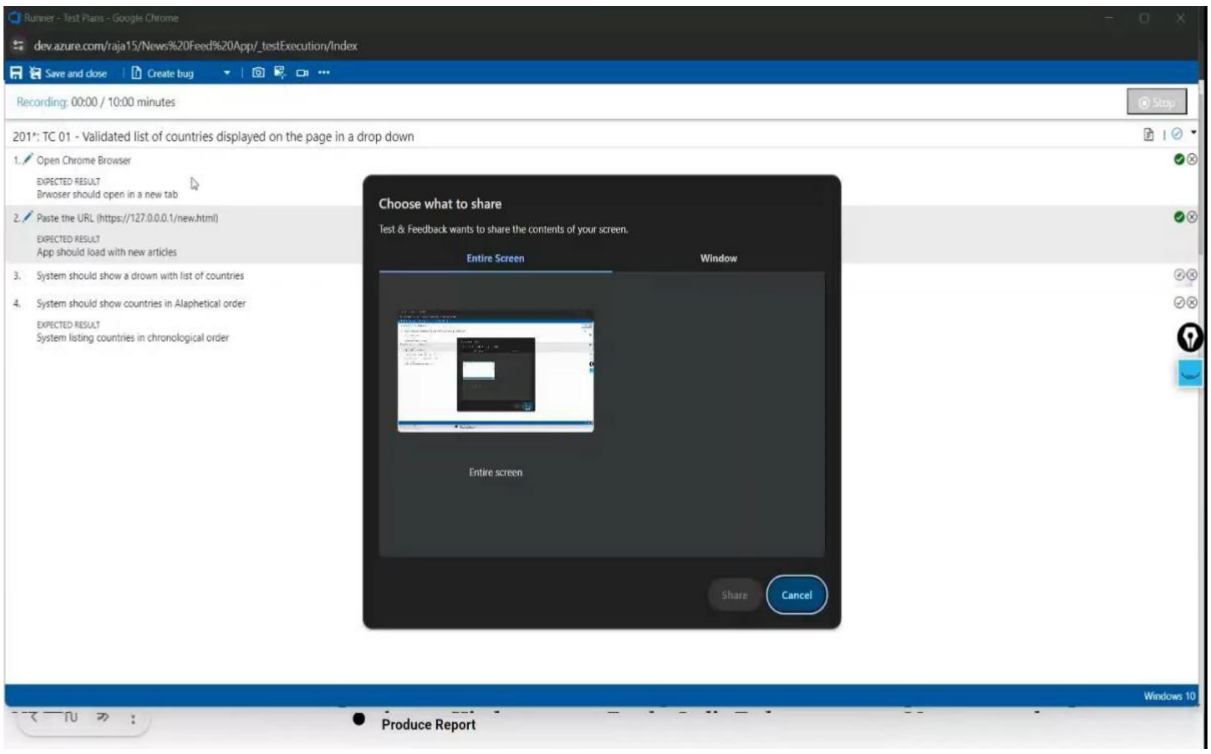
1. Log in successfully

EXPECTED RESULT
User is redirected to dashboard

2. Navigate to "My Playlists" section

EXPECTED RESULT
All created playlists are displayed clearly

6.Recording the test case



Runner - Test Plans - Google Chrome

dev.azure.com/raja15/News%20Feed%20App/_testExecution/Index

Recording: 00:00 / 10:00 minutes

201*: TC 01 - Validated list of countries displayed on the page in a drop down

1. Open Chrome Browser
EXPECTED RESULT
Brwoser should open in a new tab

2. Paste the URL (https://127.0.0.0.1/new.html)
EXPECTED RESULT
App should load with new articles

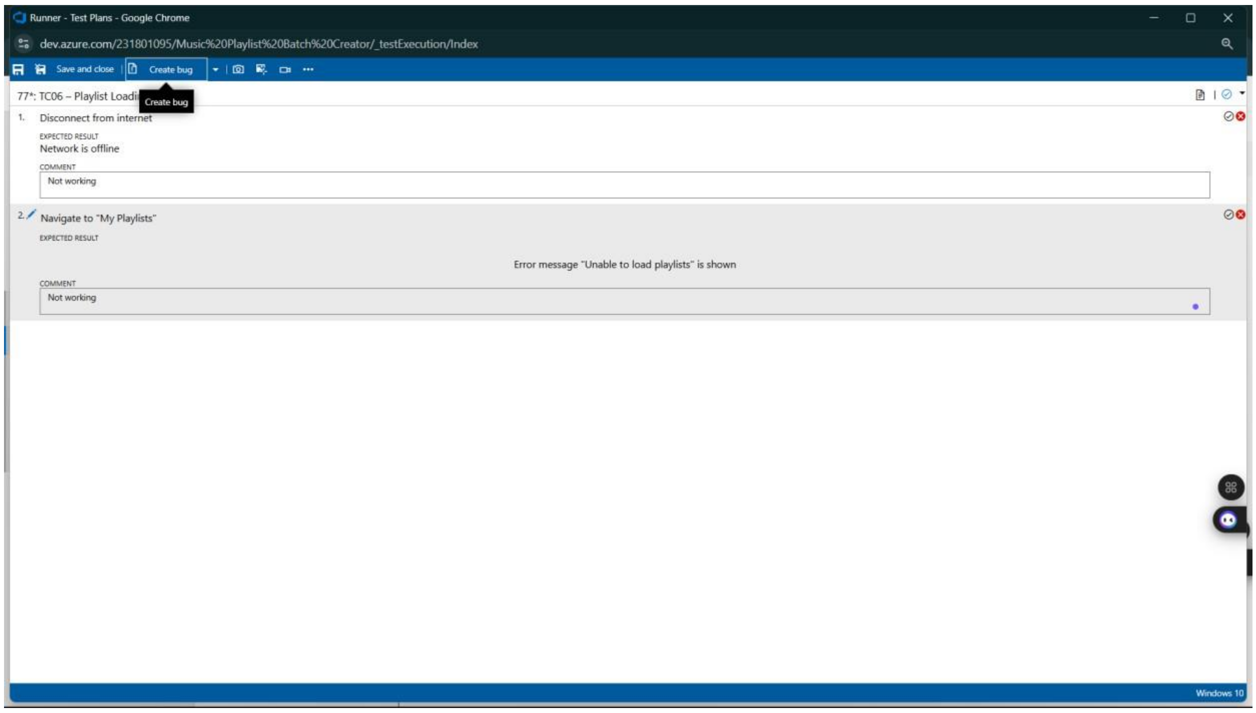
3. System should show a drown with list of countries

4. System should show countries in Alaphetical order
EXPECTED RESULT
System listing countries in chronological order

Choose what to share

Test & Feedback wants to share the contents of your screen.

7.Creating the bug



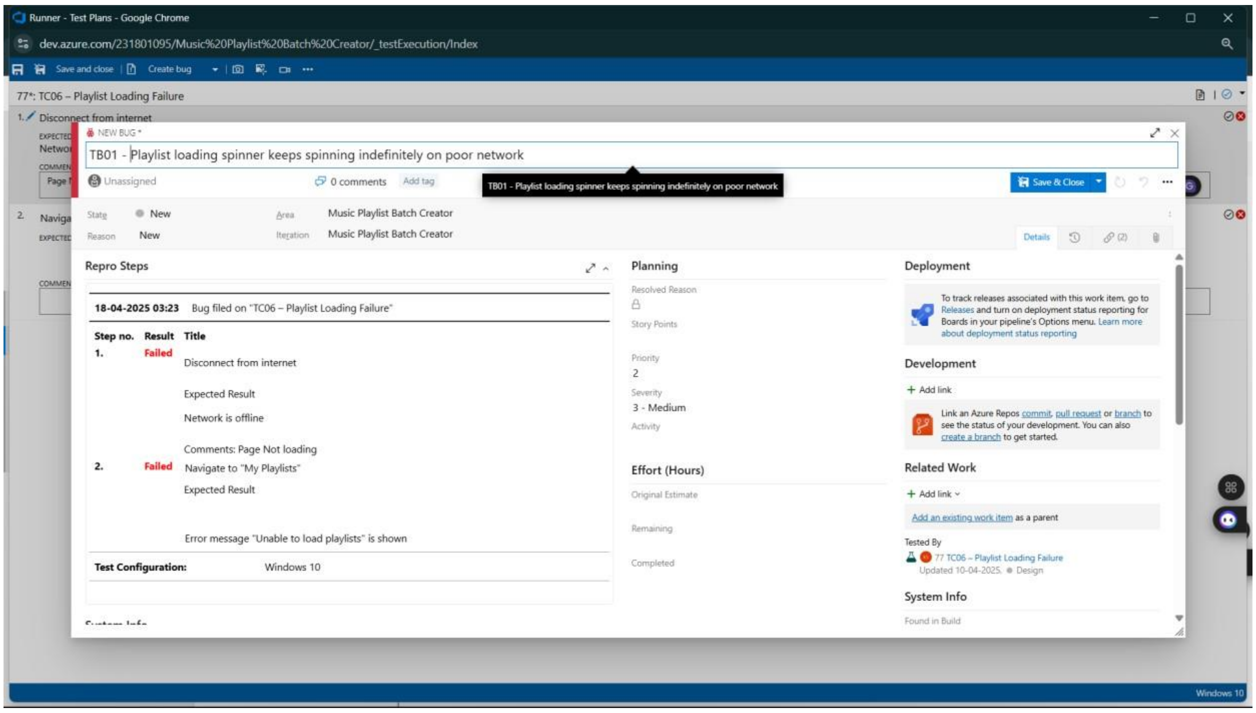
Runner - Test Plans - Google Chrome

dev.azure.com/231801095/Music%20Playlist%20Batch%20Creator/_testExecution/Index

77*: TC06 – Playlist Loadi

1. Disconnect from internet
EXPECTED RESULT
Network is offline
COMMENT
Not working

2. Navigate to "My Playlists"
EXPECTED RESULT
Error message "Unable to load playlists" is shown
COMMENT
Not working



Runner - Test Plans - Google Chrome

dev.azure.com/231801095/Music%20Playlist%20Batch%20Creator/_testExecution/Index

77*: TC06 – Playlist Loading Failure

NEW BUG *

TB01 - Playlist loading spinner keeps spinning indefinitely on poor network

Unassigned · 0 comments · Add tag · Save & Close

State	New	Area	Music Playlist Batch Creator
Reason	New	Iteration	Music Playlist Batch Creator

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 Comments: Page Not loading
2.	Failed	Navigate to "My Playlists" Expected Result Error message "Unable to load playlists" is shown

Test Configuration: Windows 10

Planning

Resolved Reason

Story Points

Priority: 2

Severity: 3 - Medium

Activity

Effort (Hours)

Original Estimate

Remaining

Completed

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

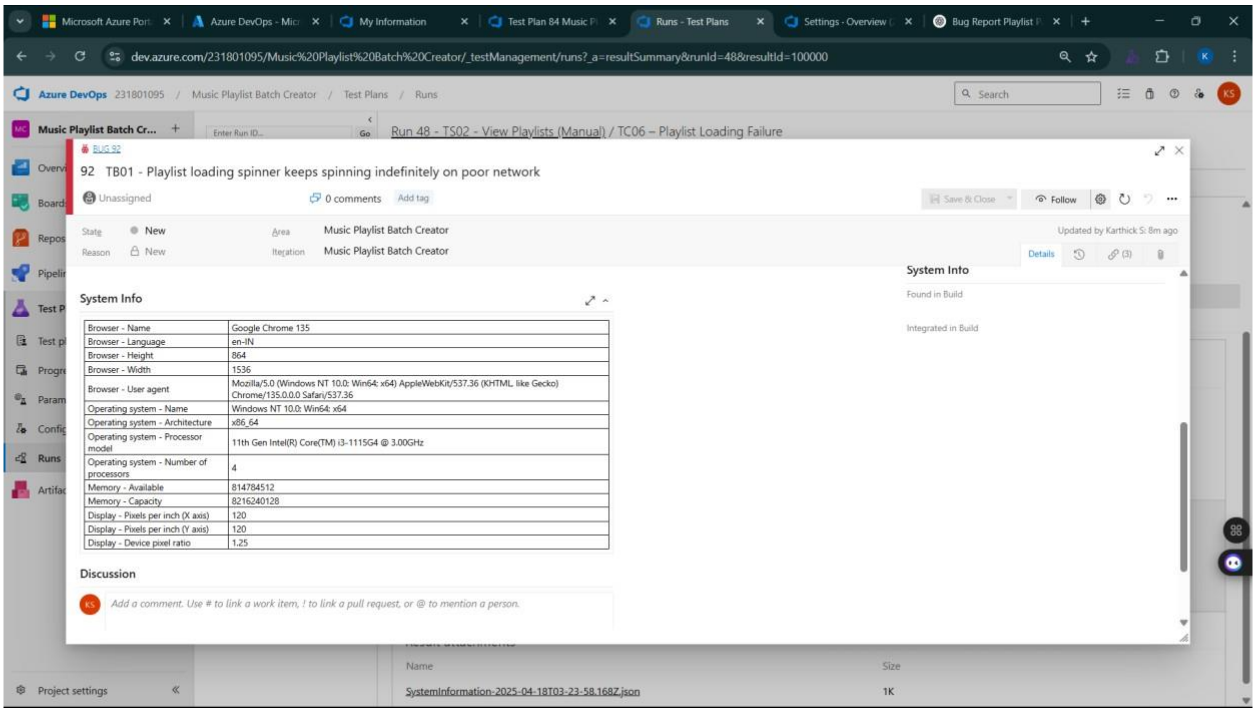
Add an existing work item as a parent

Tested By

77 TC06 – Playlist Loading Failure
Updated 10-04-2025, Design

System Info

Found in Build



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

BUG 92

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

Unassigned · 0 comments · Add tag · Follow

State	New	Area	Music Playlist Batch Creator
Reason	New	Iteration	Music Playlist Batch Creator

Updated by Karthick S: 8m ago

System Info

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	x86_64
Operating system - Processor model	11th Gen Intel(R) Core(TM) i3-1115G4 @ 3.00GHz
Operating system - Number of processors	4
Memory - Available	814784512
Memory - Capacity	8216240128
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.

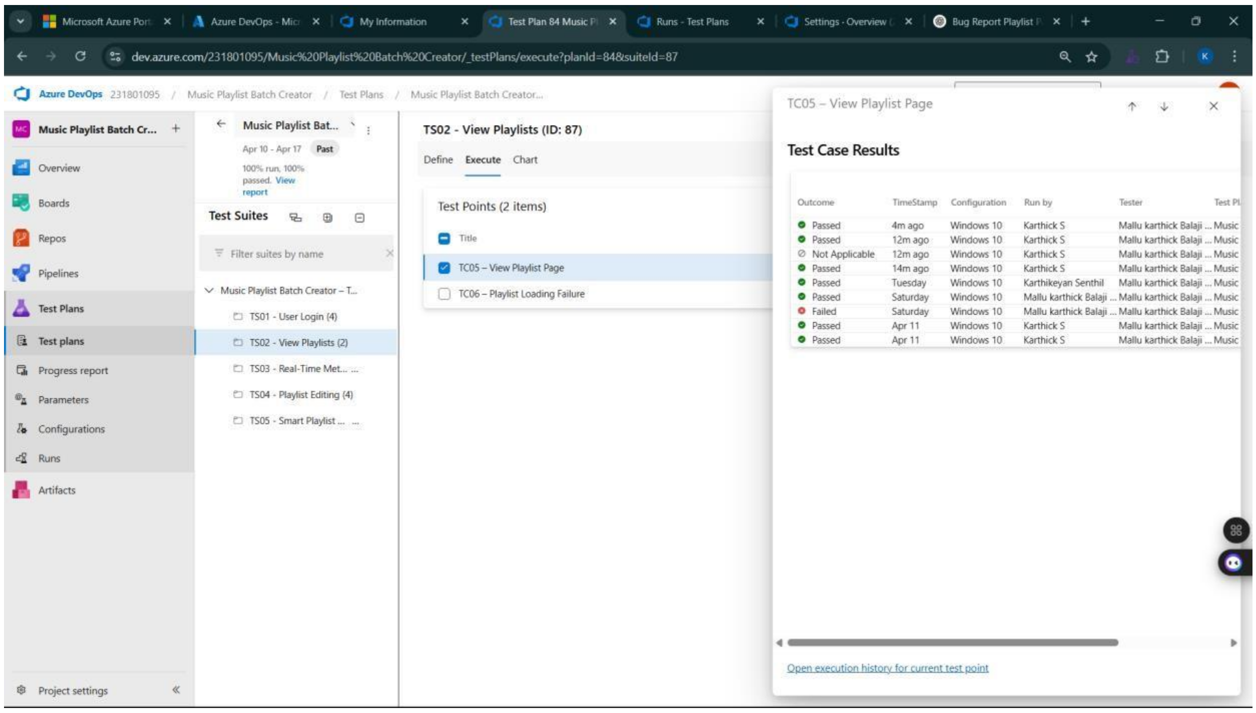
System Info

Found in Build

Integrated in Build

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

8.Test case results



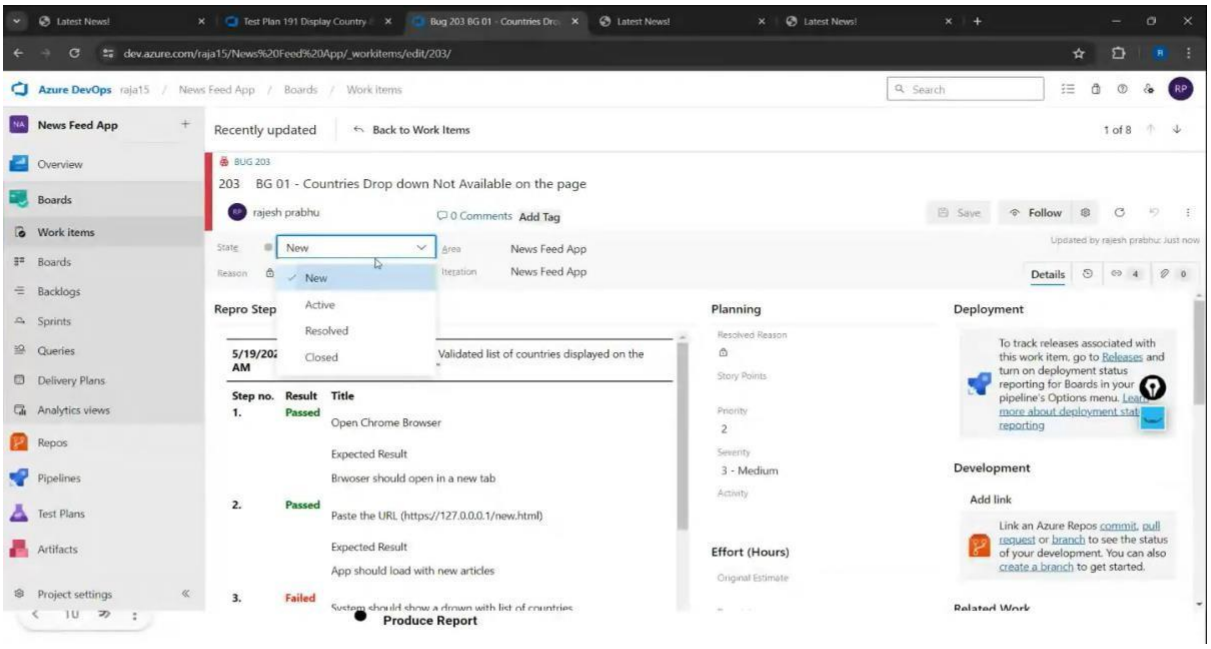
TC05 – View Playlist Page

Test Case Results

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

Open execution history for current test point

9.Test report summary



BUG 203

203 BG 01 - Countries Drop down Not Available on the page

rajesh prabhu

State: New

Area: News Feed App

Iteration: News Feed App

Repro Step

Step no.	Result	Title
1.	Passed	Open Chrome Browser Expected Result Brwoser should open in a new tab
2.	Passed	Paste the URL (https://127.0.0.1/new.html) Expected Result App should load with new articles
3.	Failed	System should show a drown with list of countries

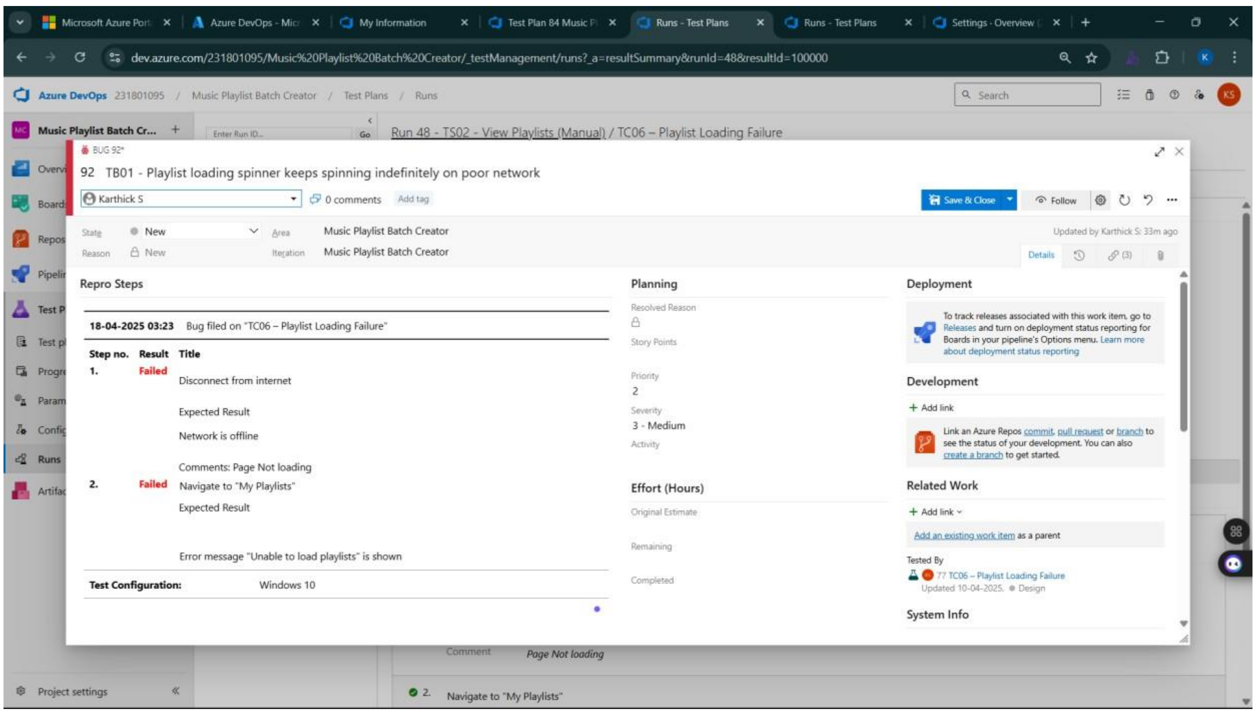
Planning

Priority: 2

Severity: 3 - Medium

- Produce Report

- Assigning bug to the developer and changing state


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

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

BUG 92*

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

Karthick S · 0 comments · Add tag

State: New · Area: Music Playlist Batch Creator
Reason: New · Iteration: Music Playlist Batch Creator

Updated by Karthick S: 33m ago

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 Comments: Page Not loading
2.	Failed	Navigate to "My Playlists" Expected Result Error message "Unable to load playlists" is shown

Test Configuration: Windows 10

Planning

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

Effort (Hours)

Original Estimate
Remaining
Completed

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

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

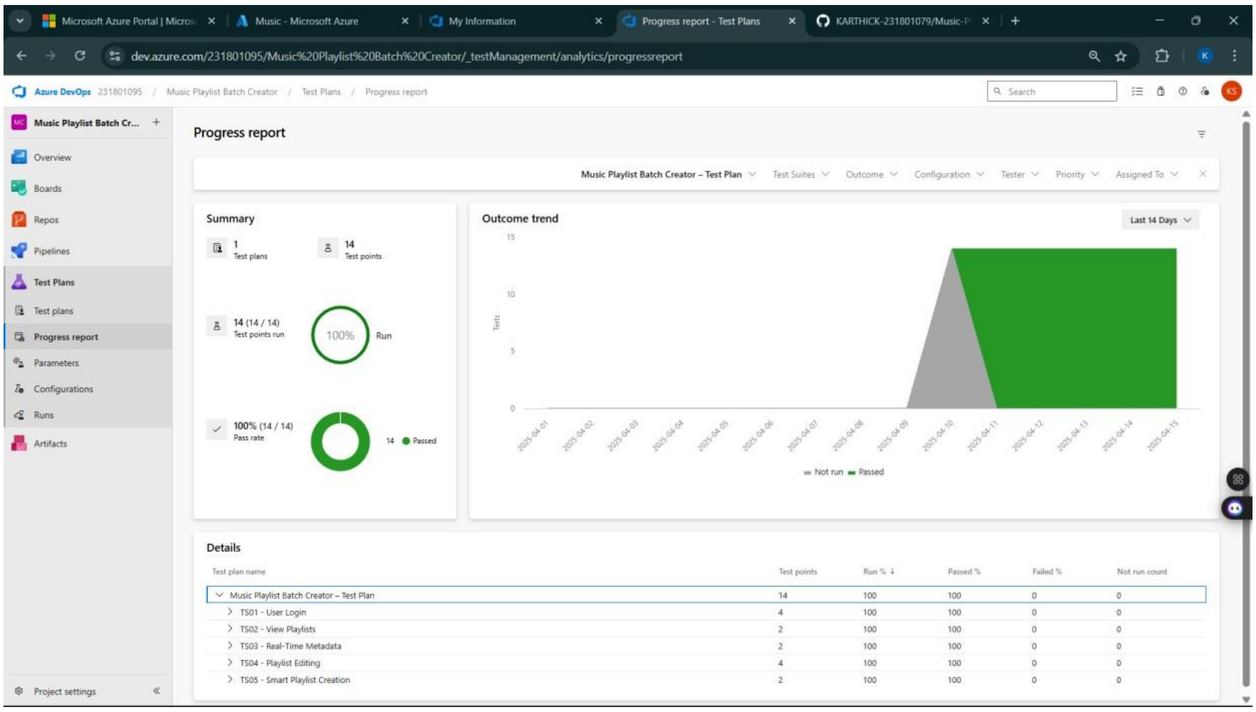
Add an existing work item as a parent

Tested By
77 TC06 – Playlist Loading Failure
Updated 10-04-2025. Design

System Info

Comment: Page Not loading
2. Navigate to "My Playlists"

10.Progress report


dev.azure.com/231801095/Music%20Playlist%20Batch%20Creator/_testManagement/analytics/progressreport

Progress report

Music Playlist Batch Creator – Test Plan

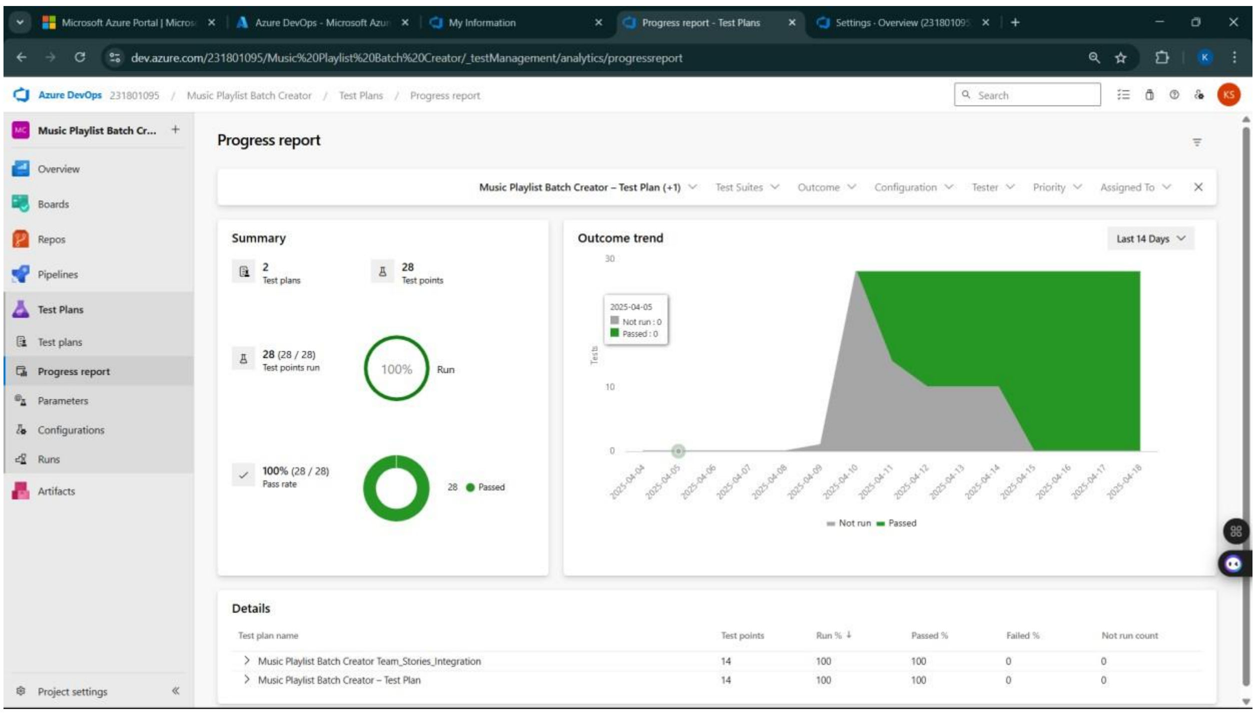
Summary

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

Outcome trend (Last 14 Days)

Details

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



Progress report

Summary

2 Test plans

28 Test points

28 (28 / 28) Test points run — 100% Run

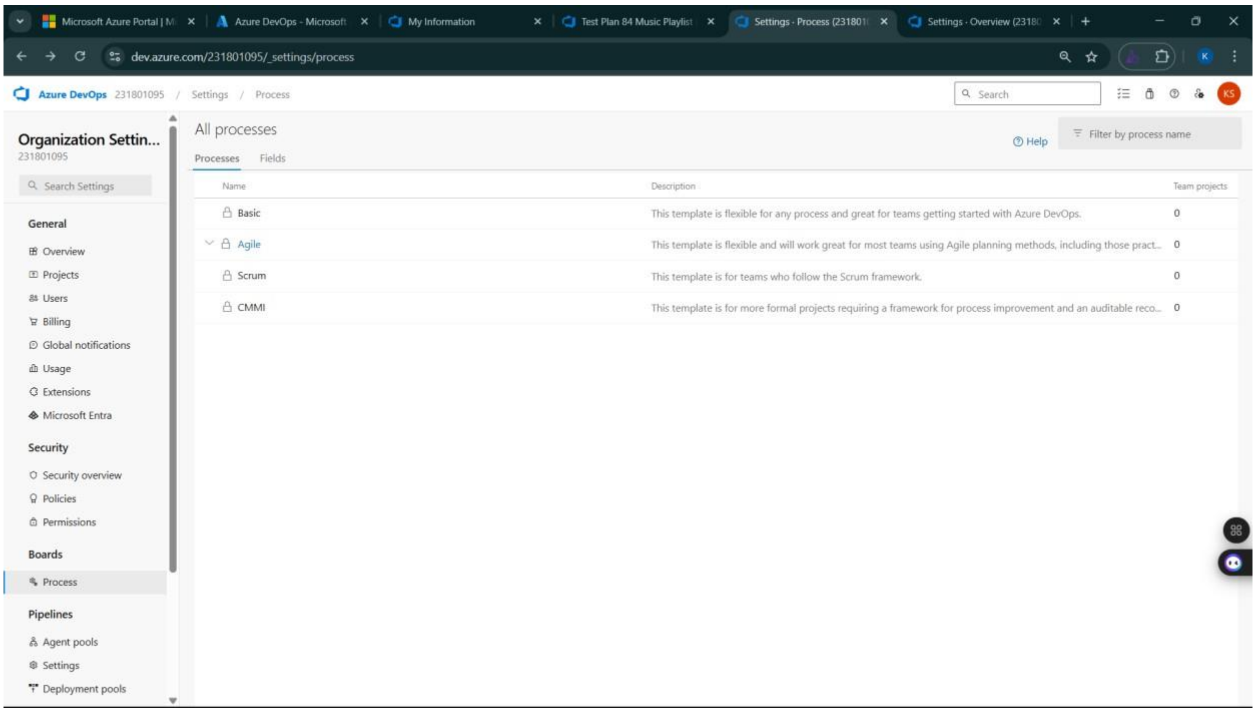
100% (28 / 28) Pass rate — 28 Passed

Outcome trend

Details

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

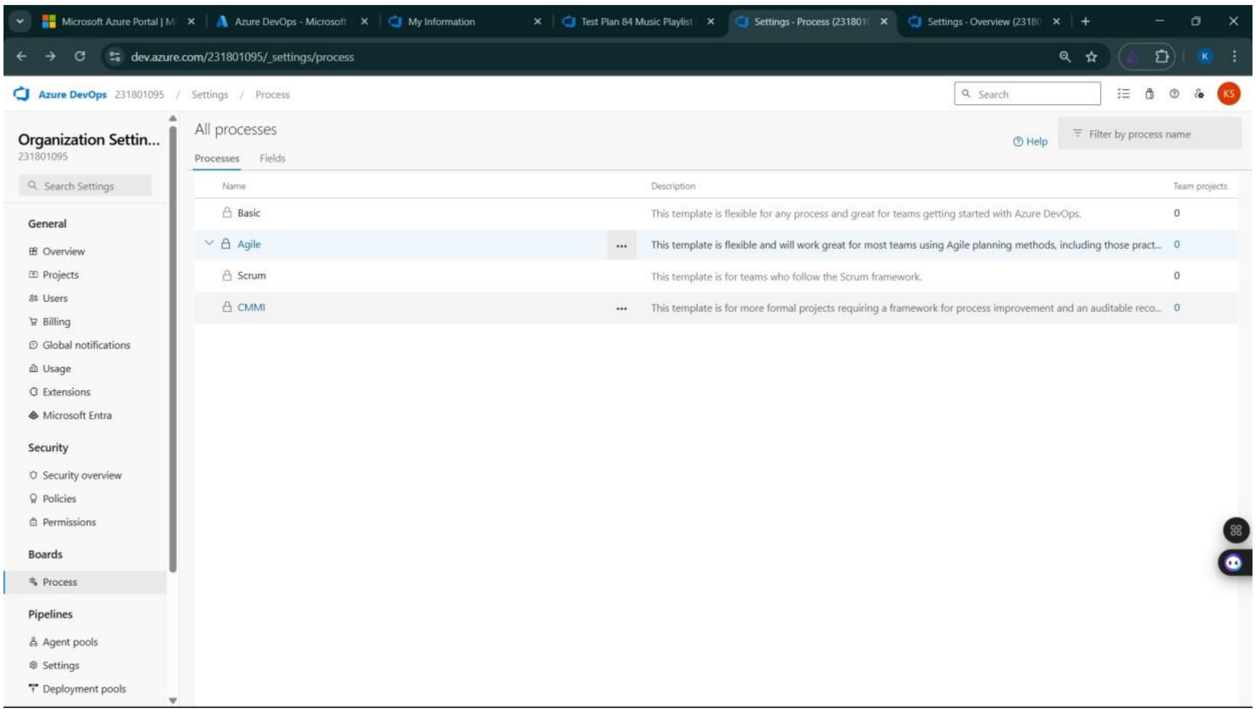
11.Changing the test template



All processes

Processes Fields

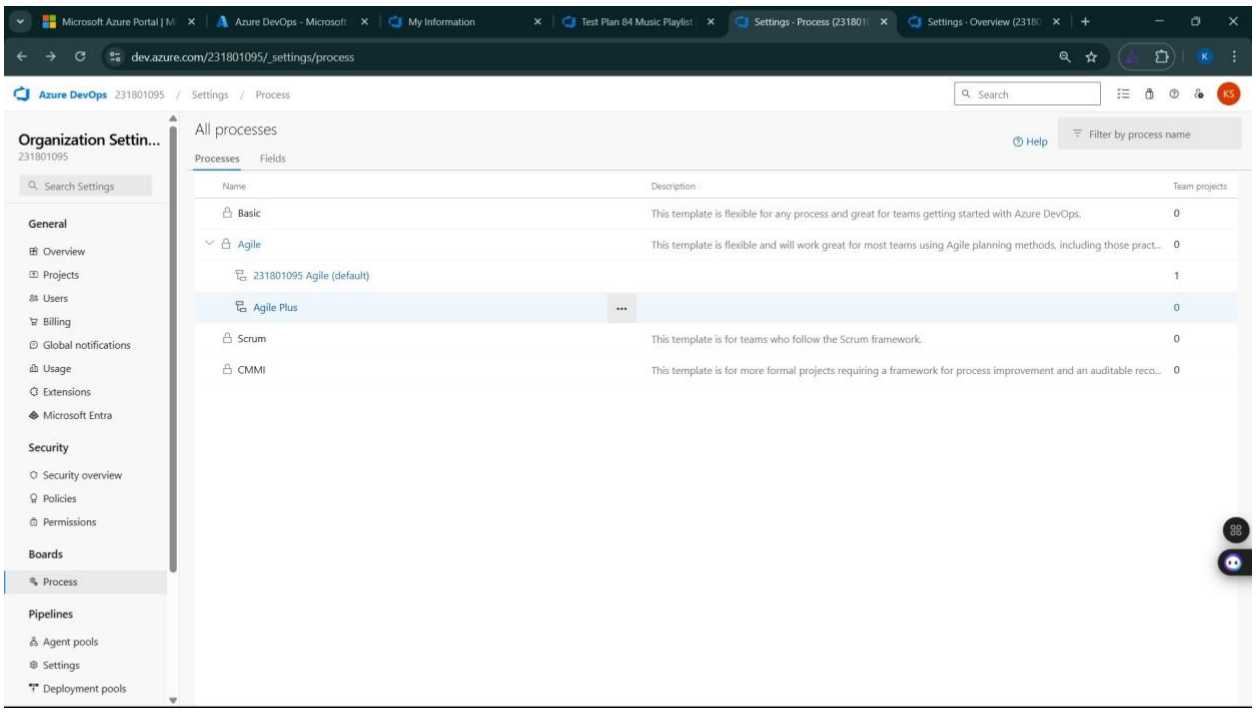
Name	Description	Team projects
Basic	This template is flexible for any process and great for teams getting started with Azure DevOps.	0
Agile	This template is flexible and will work great for most teams using Agile planning methods, including those pract...	0
Scrum	This template is for teams who follow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



dev.azure.com/231801095/_settings/process

All processes

Name	Description	Team projects
Basic	This template is flexible for any process and great for teams getting started with Azure DevOps.	0
Agile	This template is flexible and will work great for most teams using Agile planning methods, including those 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

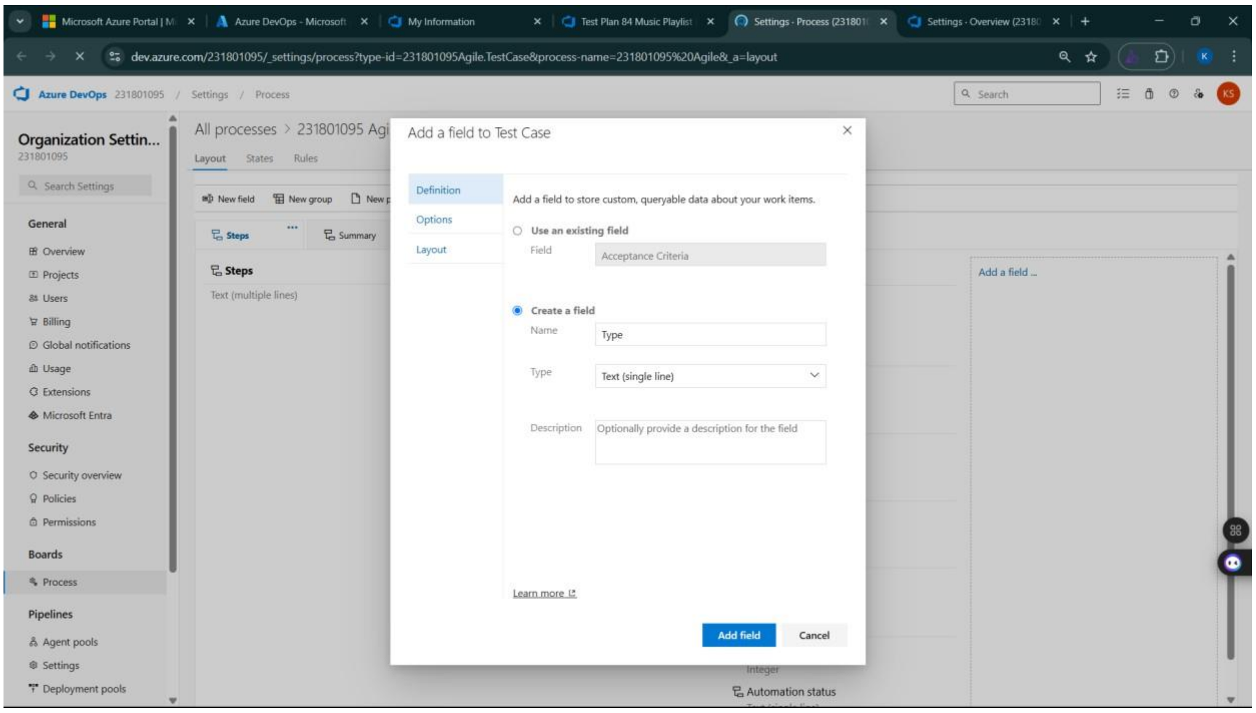


dev.azure.com/231801095/_settings/process

All processes

Name	Description	Team projects
Basic	This template is flexible for any process and great for teams getting started with Azure DevOps.	0
Agile	This template is flexible and will work great for most teams using Agile planning methods, including those 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

12.View the new test case template



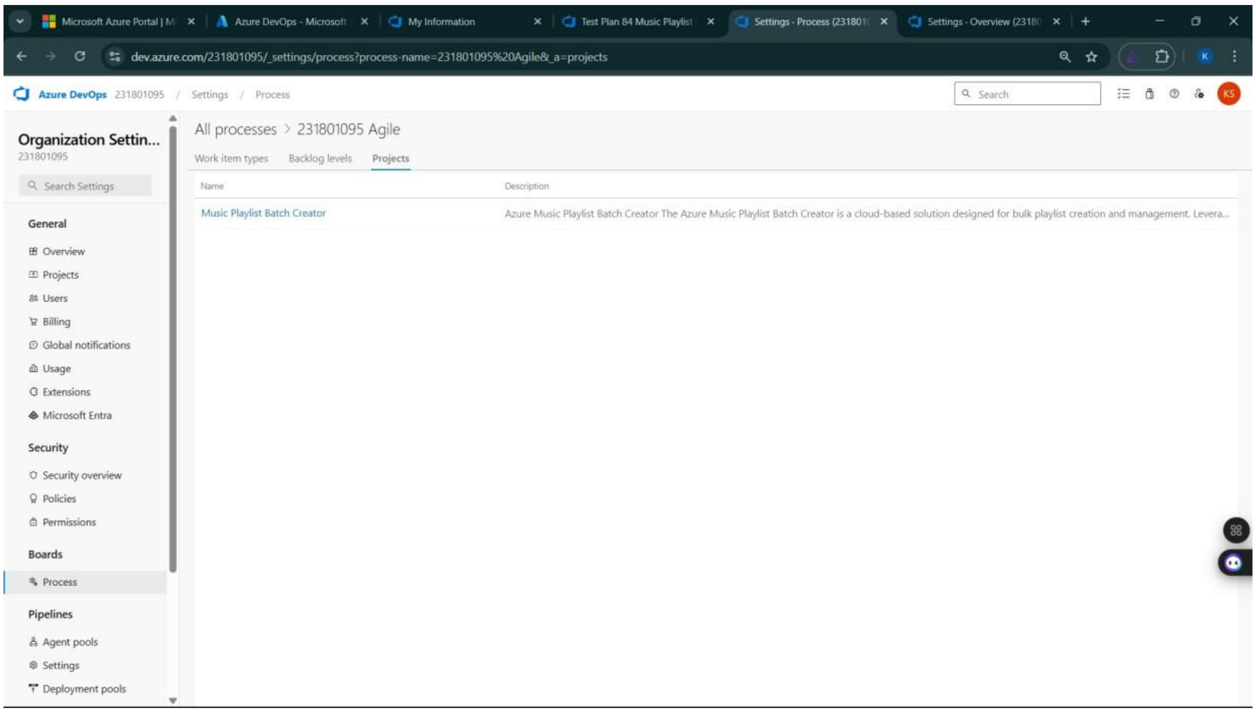
dev.azure.com/231801095/_settings/process?type-id=231801095Agile.TestCase&process-name=231801095%20Agile&_a=layout

Add a field to Test Case

Add a field to store custom, queryable data about your work items.

Use an existing field
Field: Acceptance Criteria

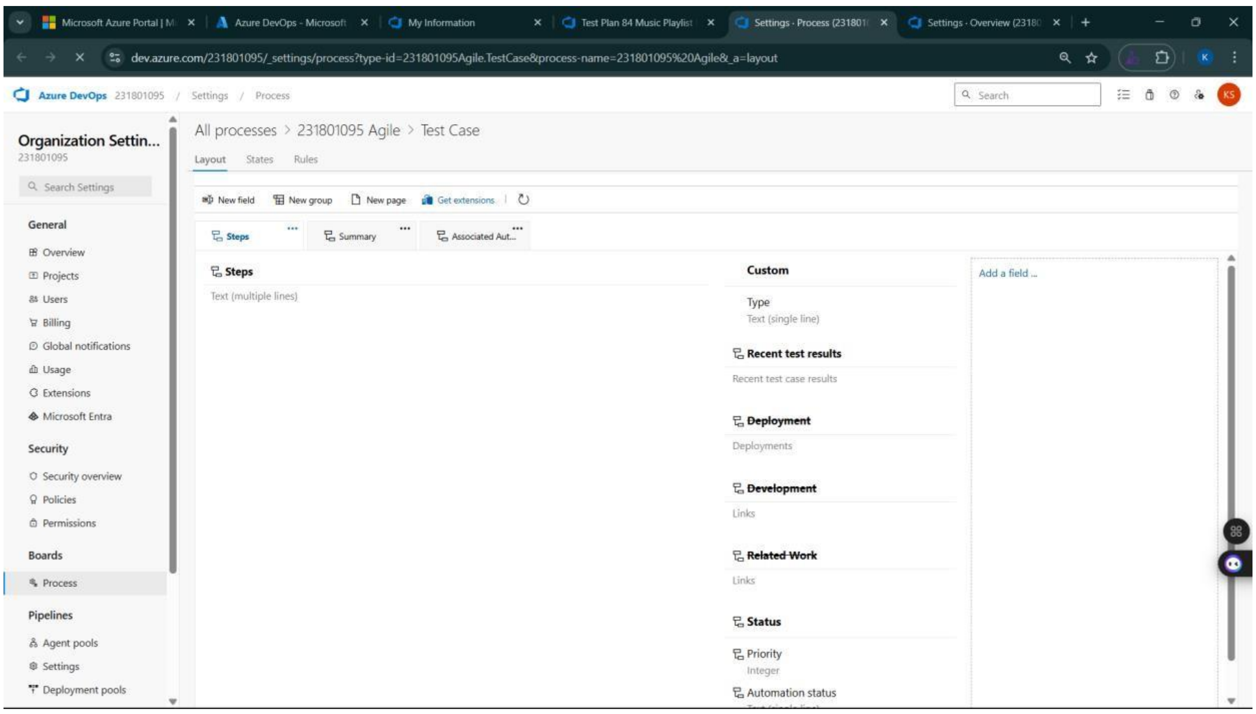
[✓ Create a field]
Name: Type
Type: Text (single line)
Description: Optionally provide a description for the field



dev.azure.com/231801095/_settings/process?process-name=231801095%20Agile&_a=projects

All processes > 231801095 Agile

Name	Description
Music Playlist Batch Creator	Azure Music Playlist Batch Creator The Azure Music Playlist Batch Creator is a cloud-based solution designed for bulk playlist creation and management. Levera...



Result:

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