

INDEX

S. No.	Date	Title
1	21/01/25	Azure DevOps Environment SetUp
2	21/01/25	Azure DevOps Project SetUp and User Story Management
3	28/01/25	Setting Up Epics, Feature and User Stories for Project Planning
4	11/02/25	Sprint Planning
5	18/02/25	Poker Estimation
6	25/02/25	Design the Class and Sequence Diagram
7	04/03/25	Design the Activity and Use Case Diagram
8	25/03/25	Testing: Test Plans and Test Cases
9	15/04/25	Pipelines Creation
10	22/04/25	GitHub Project Structure

EXP NO: 1

AZURE DEVOPS ENVIRONMENT SETUP

Aim:

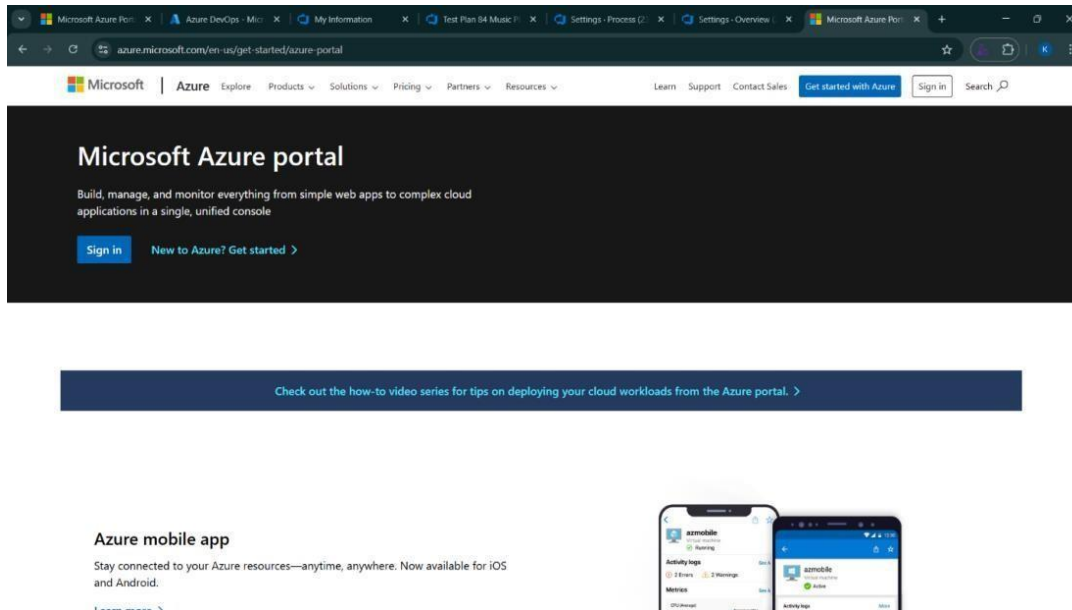
To set up and access the Azure DevOps environment by creating an organization through the Azure portal.

INSTALLATION

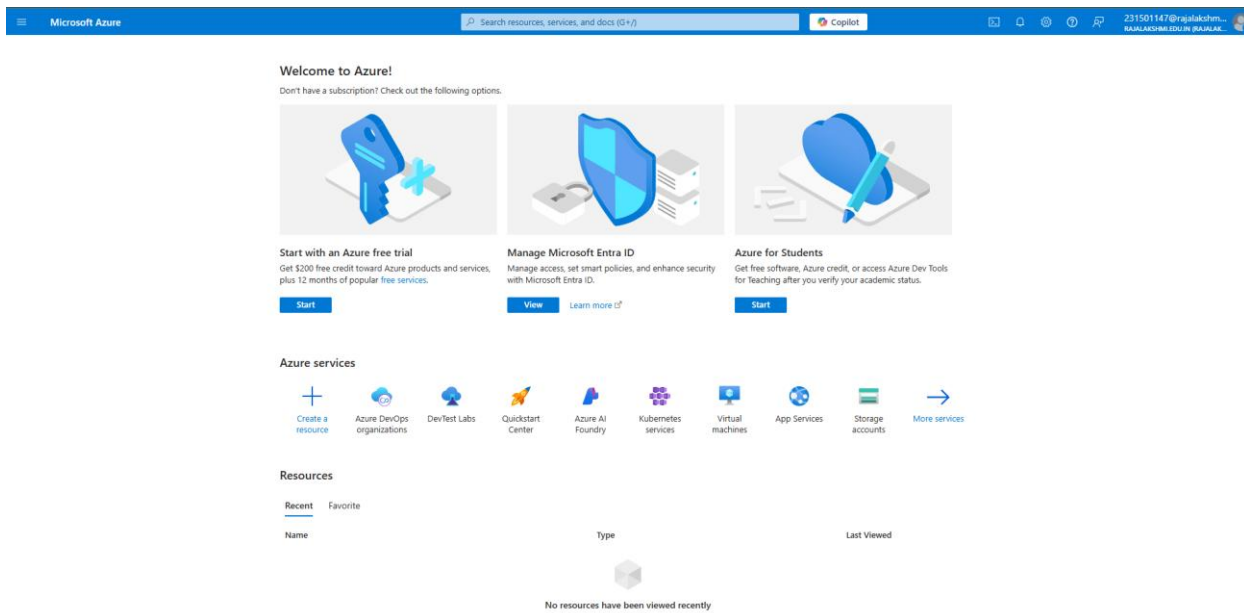
1. Open your web browser and go to the Azure website: <https://azure.microsoft.com/en-us/get-started/azure-portal>.

Sign in using your Microsoft account credentials.

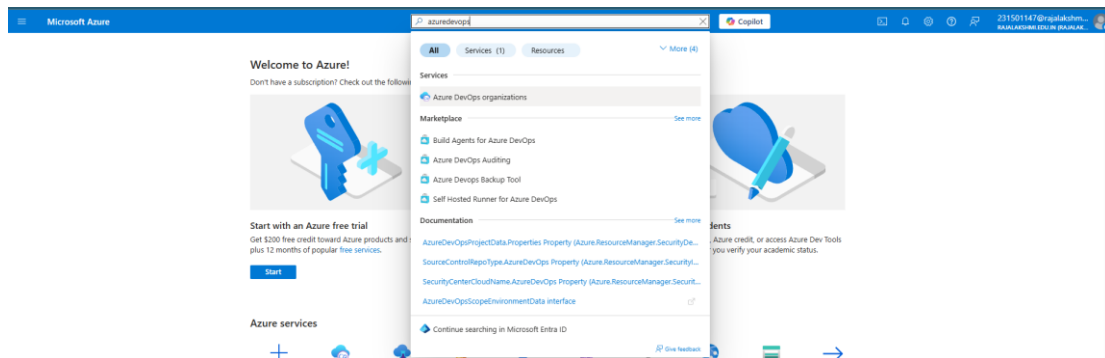
If you don't have a Microsoft account, you can create one here: <https://signup.live.com/?lic=1>



2. Azure home page



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



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

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](#)



Result:

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

2116231501147

CS23432

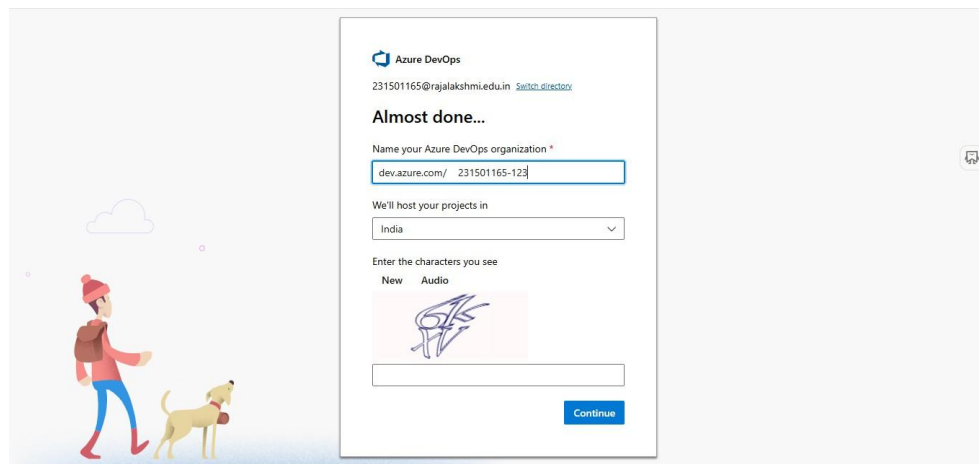
EXP NO: 2

AZURE DEVOPS PROJECT SETUP AND USER STORY MANAGEMENT

Aim:

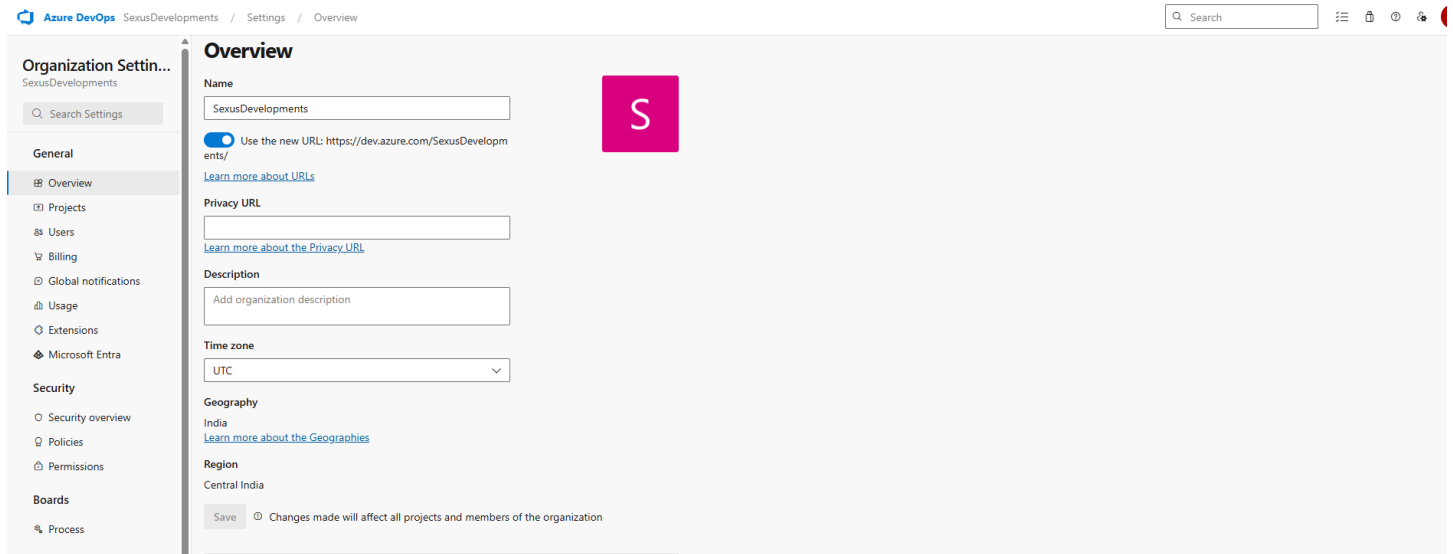
To set up an Azure DevOps project for efficient collaboration and agile work management.

1. Create an Azure Account



2. Create the First Project in Your Organization

- a. After the organization is set up, you'll need to create your first **project**. This is where you'll begin to manage code, pipelines, work items, and more.
- b. On the organization's **Home page**, click on the **New Project** button.
- c. Enter the project name, description, and visibility options:
 - Name:** Choose a name for the project (e.g., **LMS**).
 - Description:** Optionally, add a description to provide more context about the project.
 - Visibility:** Choose whether you want the project to be **Private** (accessible only to those invited) or **Public** (accessible to anyone).
- d. Once you've filled out the details, click **Create** to set up your first project.



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

4. Project dashboard

The screenshot shows the Trackify project dashboard in Azure DevOps. The left-hand navigation menu includes: Trackify, Overview, Summary, Dashboards, Wiki, Boards, Repos, Pipelines, Test Plans, and Artifacts. The main content area is titled 'Trackify' and contains an 'About this project' section with a description of the application. To the right, there are 'Project stats' (Boards: 14 Work items created, 0 Work items completed; Pipelines: 0% Builds succeeded) and 'Members' (1 member).

5. To manage user stories:

a. From the **left-hand navigation menu**, click on **Boards**. This will take you to the main **Boards** page, where you can manage work items, backlogs, and sprints.

b. On the **work items** page, you'll see the option to **Add a work item** at the top. Alternatively, you can find a + button or **Add New Work Item** depending on the view you're in. From the **Add a work item** dropdown, select **User Story**. This will open a form to enter details for the new User Story.

c.

d.

The screenshot shows the Trackify work items page in Azure DevOps. The left-hand navigation menu includes: Trackify, Overview, Boards, Work items, Backlogs, Sprints, Queries, Delivery Plans, Analytics views, Repos, Pipelines, Test Plans, and Artifacts. The main content area is titled 'Work items' and contains a table of work items. The table has columns: ID, Title, Assigned To, State, Area Path, Tags, Comments, and Activity Date. The work items are listed in a table with 14 rows.

ID	Title	Assigned To	State	Area Path	Tags	Comments	Activity Date
1	Expense Management	Unassigned	New	Trackify			5/20/2025 6:31:04 AM
2	Expense Insights & Analytics	Unassigned	New	Trackify			5/20/2025 6:31:28 AM
3	Expense Entry & Editing	Unassigned	New	Trackify			5/20/2025 6:31:04 AM
4	Category Management	Unassigned	New	Trackify			5/20/2025 6:14:29 AM
5	Monthly Expense Reports	Unassigned	New	Trackify			5/20/2025 6:15:30 AM
6	Budget Alerts	Unassigned	New	Trackify			5/20/2025 6:15:46 AM
7	As a user, I want to add an expense so I can track my spending.	Unassigned	New	Trackify			5/20/2025 6:30:37 AM
8	As a user, I want to delete an expense if I entered it incorrectly.	Unassigned	New	Trackify			5/20/2025 6:30:52 AM
9	As a user, I want to create custom categories to organize my expenses	Unassigned	New	Trackify			5/20/2025 6:30:11 AM
10	As a user, I want to view expenses grouped by category.	Unassigned	New	Trackify			5/20/2025 6:16:46 AM
11	As a user, I want to see a monthly summary chart of my expenses.	Unassigned	New	Trackify			5/20/2025 6:17:36 AM
12	As a user, I want to filter expenses by date and category.	Unassigned	New	Trackify			5/20/2025 6:17:50 AM
13	As a user, I want to set a monthly budget to control my spending	Unassigned	New	Trackify			5/20/2025 6:18:01 AM
14	As a user, I want to receive alerts when I exceed my budget.	Unassigned	New	Trackify			5/20/2025 6:18:10 AM

Results:

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

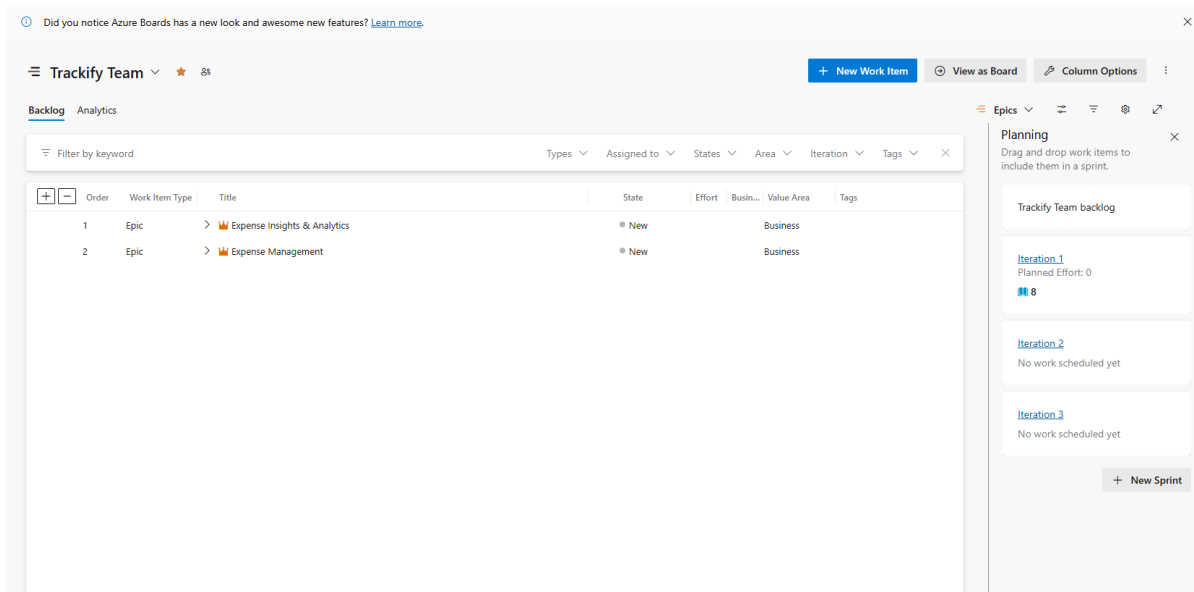
EXP NO: 3

SETTING UP EPICS, FEATURES, AND USER STORIES FOR PROJECT PLANNING

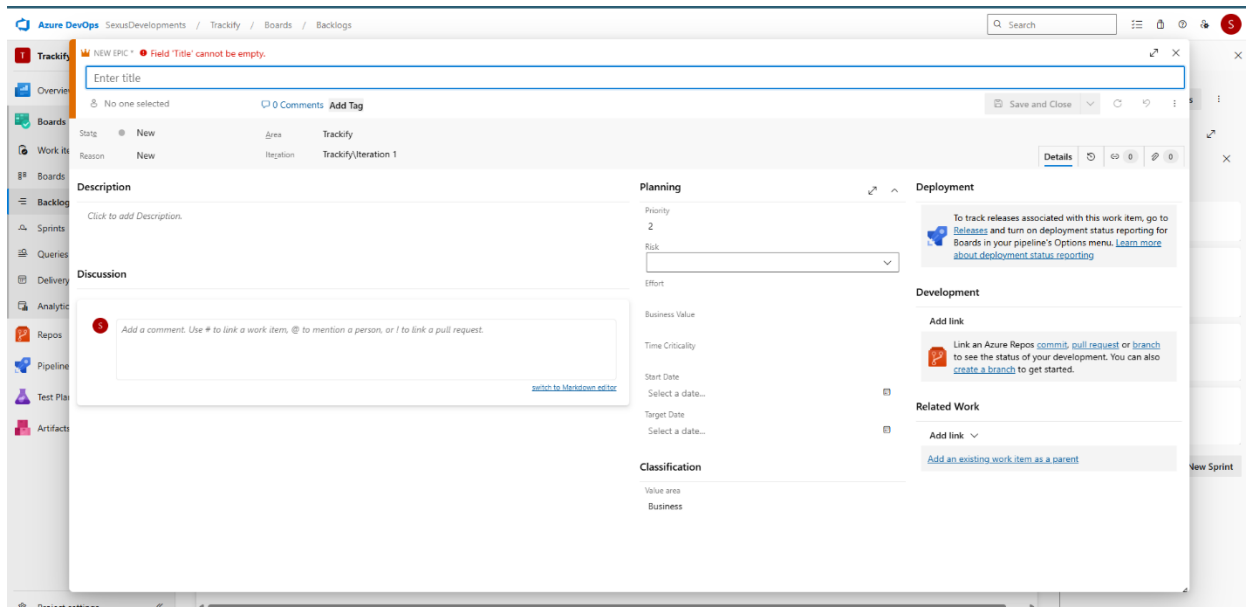
Aim:

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

Create Epic, Features, User Stories, Task



1.Fill in Epics



2116231501147

CS23432

2.Fill in Features

The screenshot shows the 'New Feature' form in Azure DevOps. The left sidebar contains navigation links: Overview, Boards, Work Items, Backlogs, Sprints, Queries, Delivery Plans, Pipelines, and Artifacts. The main area is titled 'Work Items' and includes a 'Back to Work Items' link. A message at the top states: 'NEW FEATURE * Field "Title" cannot be empty.' Below this is a text input field for the title. A table below the title field shows the following data:

State	Reason	Area	Iteration
New	New	To Do List App with Reminders	To Do List App with Reminders

Below the table are sections for 'Description', 'Discussion', 'Planning', and 'Deployment'. The 'Planning' section includes fields for Priority (2), Risk, Effort, Business Value, and Time Criticality. The 'Deployment' section includes a note: '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.'

3.Fill in User Story Details

The screenshot shows the 'New User Story' form in Azure DevOps. The left sidebar contains navigation links: Overview, Boards, Work Items, Backlogs, Sprints, Queries, Delivery Plans, Pipelines, and Artifacts. The main area is titled 'Work Items' and includes a 'Back to Work Items' link. A message at the top states: 'NEW USER STORY * Field "Title" cannot be empty.' Below this is a text input field for the title. A table below the title field shows the following data:

State	Reason	Area	Iteration
New	New	To Do List App with Reminders	To Do List App with Reminders

Below the table are sections for 'Description', 'Acceptance Criteria', 'Discussion', 'Planning', and 'Classification'. The 'Planning' section includes fields for Story Points, Priority (2), Risk, and Classification. The 'Classification' section includes a field for Value area. The 'Deployment' section includes a note: '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.'

Result:

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

EXP NO: 4

SPRINT PLANNING

Aim:

To assign user story to specific sprint for the TO DO List Project.

Sprint Planning:

Sprint 1:

Did you notice Azure Boards has a new look and awesome new features? [Learn more.](#)

Trackify Team

Taskboard Backlog Capacity Analytics

Iteration 1

Order	Title	State	Assigned To	Rem...
1	As a user, I want to add an expense so I can track my spend... @ New			
2	As a user, I want to delete an expense if I entered it incorrec... @ New			
3	As a user, I want to create custom categories to organize m... @ New			
4	As a user, I want to view expenses grouped by category. @ New			
5	As a user, I want to see a monthly summary chart of my exp... @ New			
6	As a user, I want to filter expenses by date and category. @ New			
7	As a user, I want to set a monthly budget to control my spe... @ New			
8	As a user, I want to receive alerts when I exceed my budget. @ New			

Sprint 2

Did you notice Azure Boards has a new look and awesome new features? [Learn more.](#)

Trackify Team

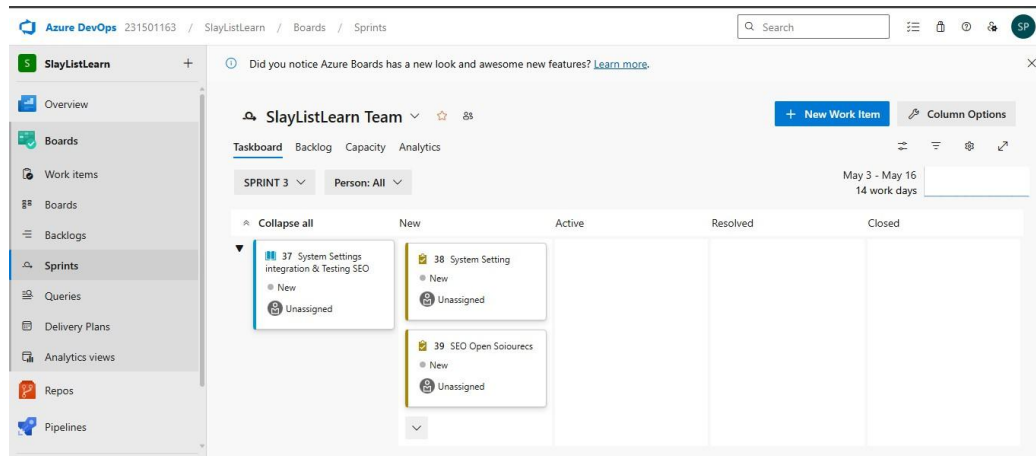
Taskboard Backlog Capacity Analytics

Iteration 1 Person: All

Collapse all

New	Active	Resolved	Closed
<div>7 As a user, I want to add an expense so I can track my spending. @ New Unassigned</div>			
<div>8 As a user, I want to delete an expense if I entered it incorrectly. @ New Unassigned</div>			
<div>9 As a user, I want to create custom categories to organize my expenses. @ New Unassigned</div>			
<div>10 As a user, I want to view expenses grouped by category. @ New Unassigned</div>			

Sprint 3



Result:

The Sprints are created for the To Do List Project.

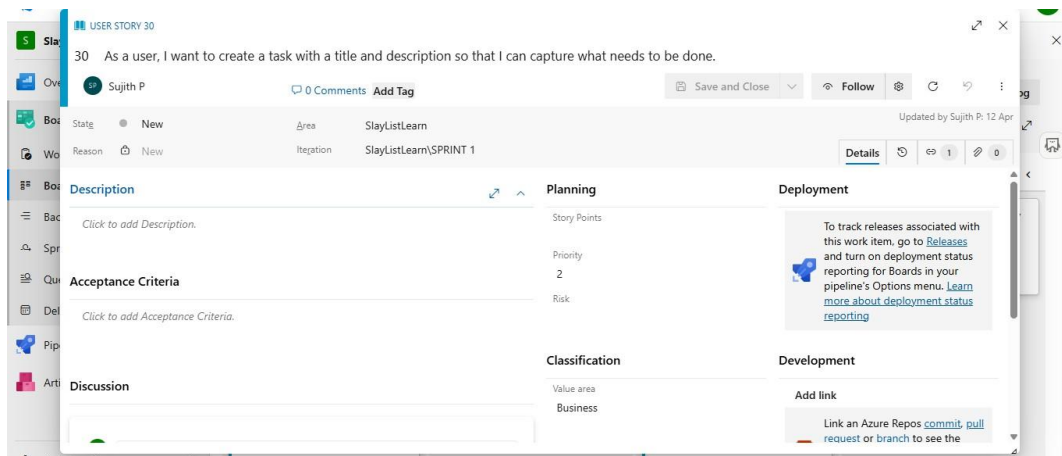
EXP NO: 5

POKER ESTIMATION

Aim:

Create Poker Estimation for the user stories – TO DO List Project.

Poker Estimation



Result:

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

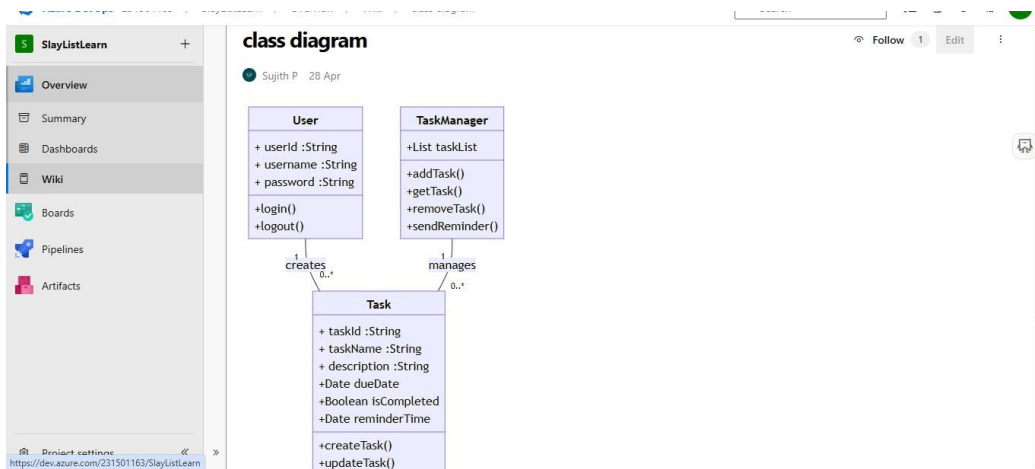
EXP NO: 6

DESIGNING CLASS AND SEQUENCE DIAGRAMS FOR PROJECT ARCHITECTURE

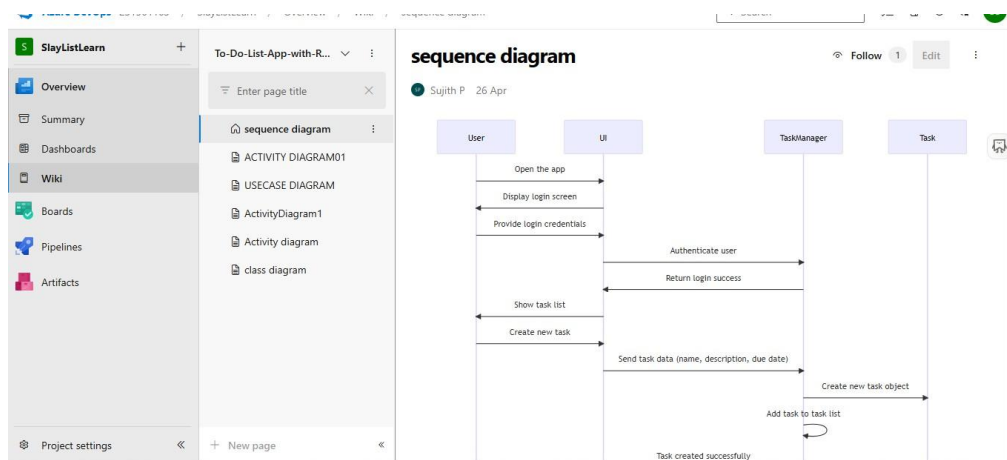
Aim:

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

6A. Class Diagram



6B. Sequence Diagram



Result:

The Class Diagram and Sequence Diagram is designed Successfully for the TO DO List.

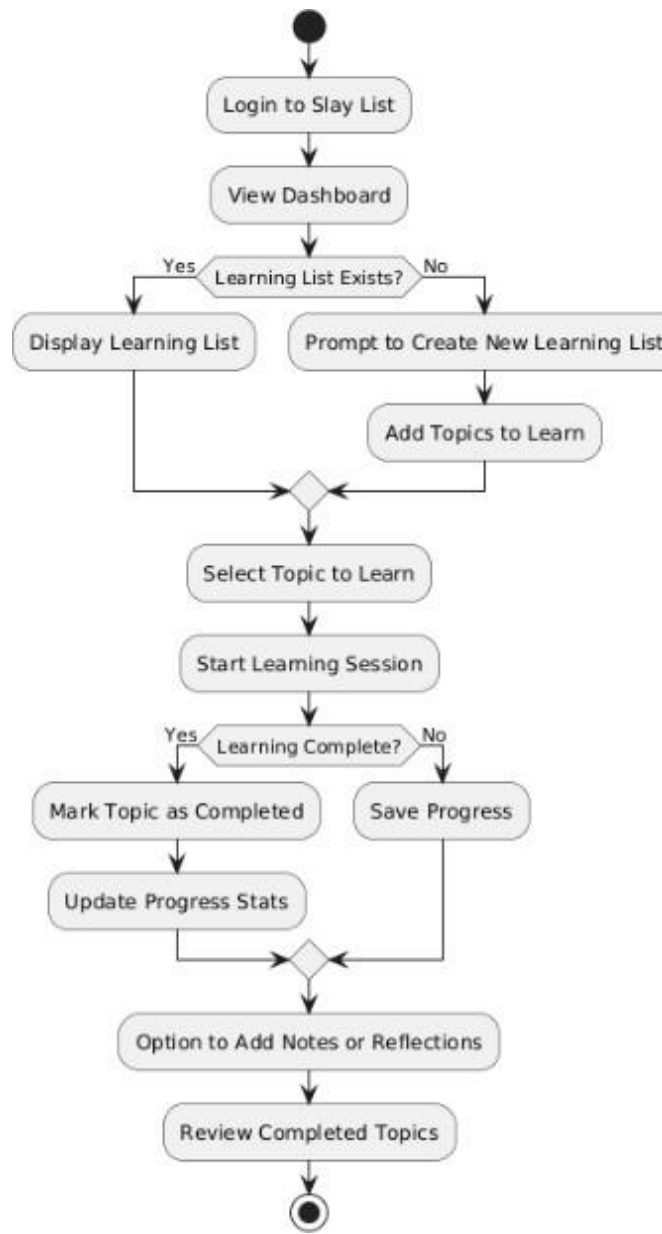
EXP NO: 7

DESIGNING ACTIVITY AND USE CASE DIAGRAMS FOR PROJECT ARCHITECTURE

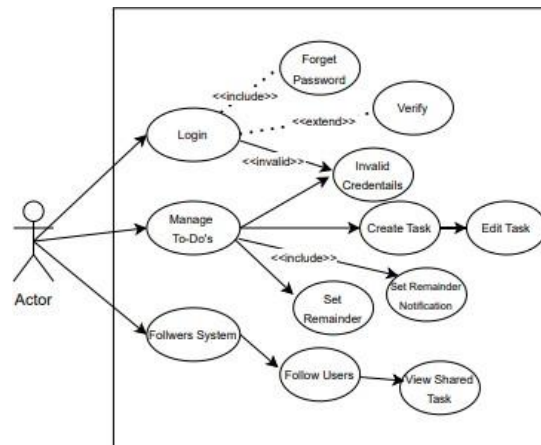
Aim:

To Design an Activity and Use Case Diagram for the given Project.

7A. Activity Diagram



7B. Use Case Diagram



Result:

The Activity and the Use Case Diagram for the given Project is successfully executed.

EXP NO: 8

TESTING – TEST PLANS AND TEST CASES

Aim:

Test Plans and Test Case and write two test cases for at least four 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 behavior (e.g., logging in, renaming a playlist, adding a song).

3. Design Happy Path Test Cases

- Focused on validating that all features function as expected under normal conditions.
- Example: User logs in successfully, adds item to playlist, or creates a category-based playlist.

4. Design Error Path Test Cases

- Simulate negative or unexpected scenarios to test robustness and error handling.
- Example: Login fails with invalid credentials, save fails when offline, no recommendations found.

5. Break Down Steps and Expected Results

- Each test case contains step-by-step actions and a corresponding expected outcome.
- Ensures clarity for both testers and automation scripts.

6. Use Clear Naming and IDs

- Test cases are named clearly (e.g., TC01 – Successful Login, TC10 – Save Playlist Fails).
- Helps in quick identification and linking to user stories or features.

7. Separate Test Suites

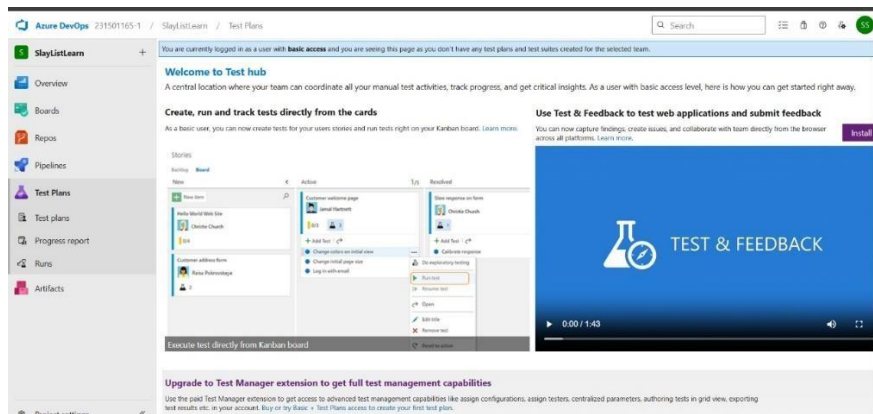
- Grouped test cases based on functionality (e.g., Login, Playlist Editing, Recommendation System).

- Improves organization and test execution flow in Azure DevOps.

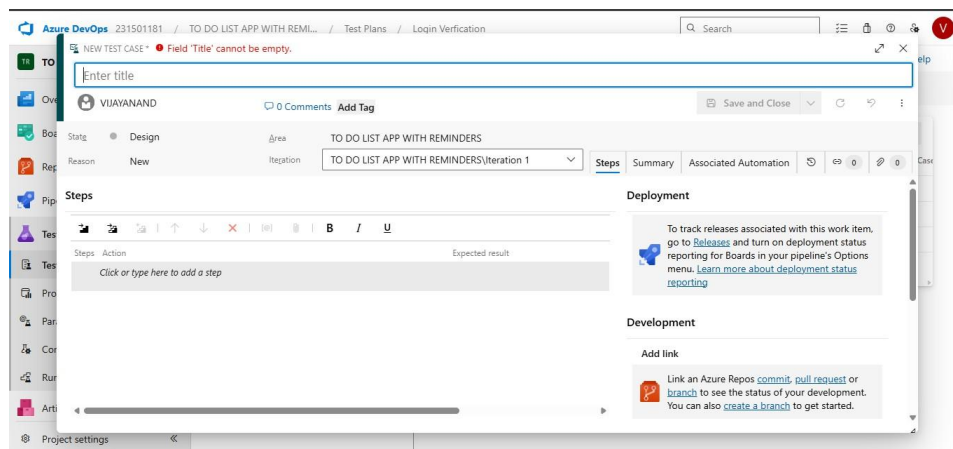
8. Prioritize and Review

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

1.New test plan



2.Test suite



3. Test case

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

Music Playlist Batch Creator – Test Plans

USER STORIES

- As a user, I want to sign up and log in securely so that I can access my playlists (ID: 79).
- As a user, I need to see my playlist in one place (ID: 76).
- As a user, I should be able to create an audio playlist as needed (ID: 73).
- As a user, I should be able to rename, record, and change the playlist (ID: 68).
- As a user, I need to have real-time metadata (ID: 65).

Test Suites

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

1. TC01 – Successful Sign Up

- **Action:**
 - Go to the Sign-Up page.
 - Enter valid name, email, and password.
 - Click "Sign Up".
- **Expected Results:**
 - Sign-Up form is displayed.
 - Fields accept values without error.
 - Account is created, and the user is redirected to the dashboard.
- **Type:** Happy Path

2. TC02 – Secure Login

- **Action:**
 - Go to the Login page.
 - Enter valid email and password.
 - Click on "Login".
- **Expected Results:**
 - Login form is displayed.
 - Fields accept data without error.
 - User is logged in and redirected to the dashboard.
- **Type:** Happy Path

3. TC03 – Sign Up with Existing Email

- **Action:**
 - Go to the Sign-Up page.
 - Enter a name and an already registered email.
 - Click on "Sign Up".
- **Expected Results:**

- Fields accept data.
- Error message "Email already registered" is displayed.
- **Type:** Error Path

4. TC04 – Login with Wrong Password

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

Test Suit: TS02 – Add List (ID: 87)

1. TC05 – View TO DO List Page

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

2. TC06 –List Failure

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

Test Suit: TS03 – Real Time TO DO Socket (ID: 88)

1. TC07 – Real-Time Metadata Display

- **Action:**
 - Display the list panel.
 - Observe the metadata panel.
- **Expected Results:**
 - Metadata (title, user, list, duration) is displayed and updates in real time.
- **Type:** Happy Path

2. TC08 – Metadata Not Updating

- **Action:**

- Add different attribute.
 - Observe the metadata panel.
- **Expected Results:**
 - Metadata remains static or shows default/fallback message.
- **Type:** Error Path

Test Suit: TS04 - Editing (ID: 89)

1. TC09 – Rename Edit Successfully

- **Action:**
 - Navigate to “Rename the List ”.
 - Click "Rename" next to a playlist.
 - Enter a new name and click "Save".
- **Expected Results:**
 - List details / 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 "List name cannot be empty" is shown.
- **Type:** Error Path

3. TC11 – Change Priority Order

- **Action:**
 - Open a List.
 - Drag and drop list to reorder.
 - Click "Save".
- **Expected Results:**
 - Playlist order is updated and saved.
- **Type:** Happy Path

4. TC12 – Change Priority Order Fails

- **Action:**
 - Login and go to “My Playlists”.
 - Select a List.
 - 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 Cases

The screenshot displays the Azure DevOps Test Plans interface. On the left, a sidebar lists navigation options: Overview, Boards, Repos, Pipelines, Test Plans (selected), Test plans, Progress report, Parameters, Configurations, Runs, and Artifacts. The main area shows the 'Login Verification (ID: 26)' test suite. A summary bar indicates 'May 15 - May 22', 'Current', and '100% run, 100% passed. View report'. Below this, a 'Test Suites' section shows 'Login Verification (4)'. The right pane, titled 'Login Verification (ID: 26)', has tabs for 'Define', 'Execute', and 'Chart'. The 'Execute' tab is active, showing 'Test Points (4 items)' with a 'Run for web application' dropdown. The test points are listed in a table:

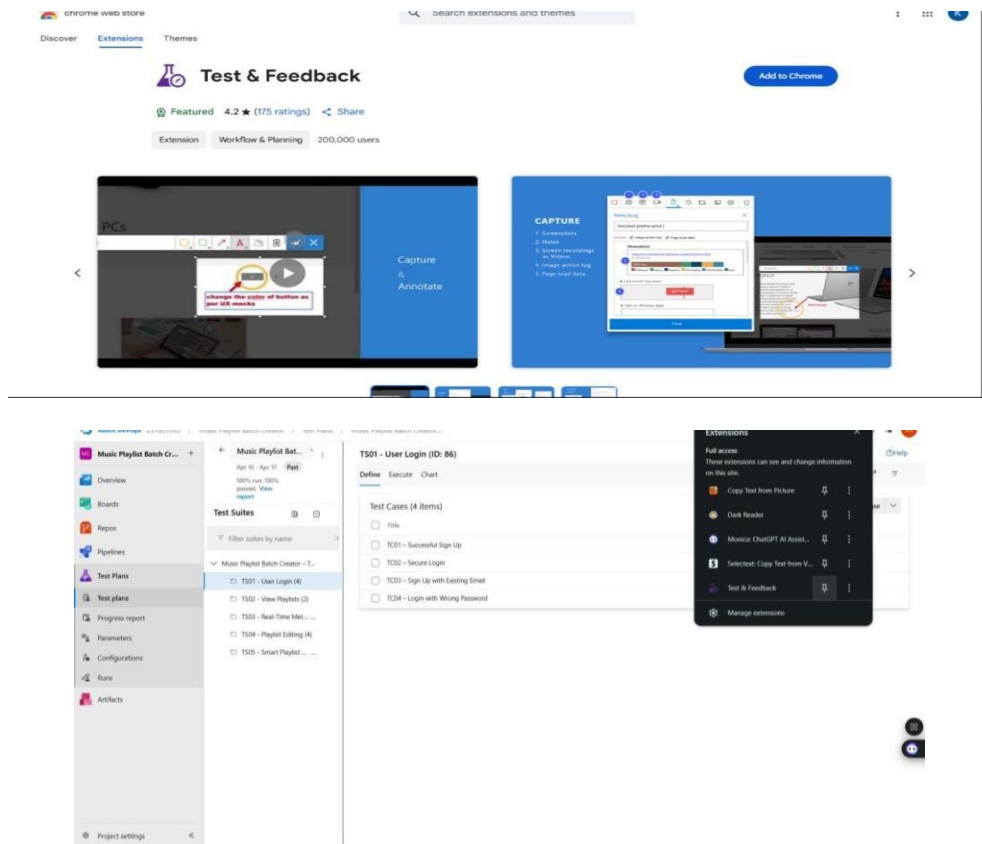
<input type="checkbox"/> Title	Outcome	
<input type="checkbox"/> Login Verification of correct User	Passed	1
<input type="checkbox"/> LoginVerification for Wrong info	Passed	2
<input type="checkbox"/> Add the TO DO Item to the list	Passed	3
<input type="checkbox"/> Application Setting Completion	Passed	4

The screenshot shows a detailed view of a test case execution. The title is '27* Login Verification of correct User'. The execution steps are listed with their outcomes:

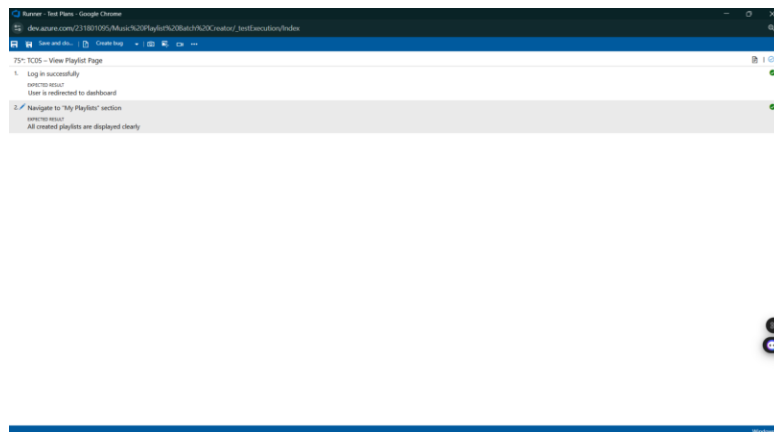
1. Open the browser
EXPECTED RESULT: Redirect to the Main SlayListLearn page
Outcome: Passed
2. Click the log in Button
Outcome: Passed
3. Enter the correct details
Outcome: Passed

The interface includes a search bar, a list of test cases, and a detailed view of the selected test case. The bottom of the window shows 'Windows 10'.

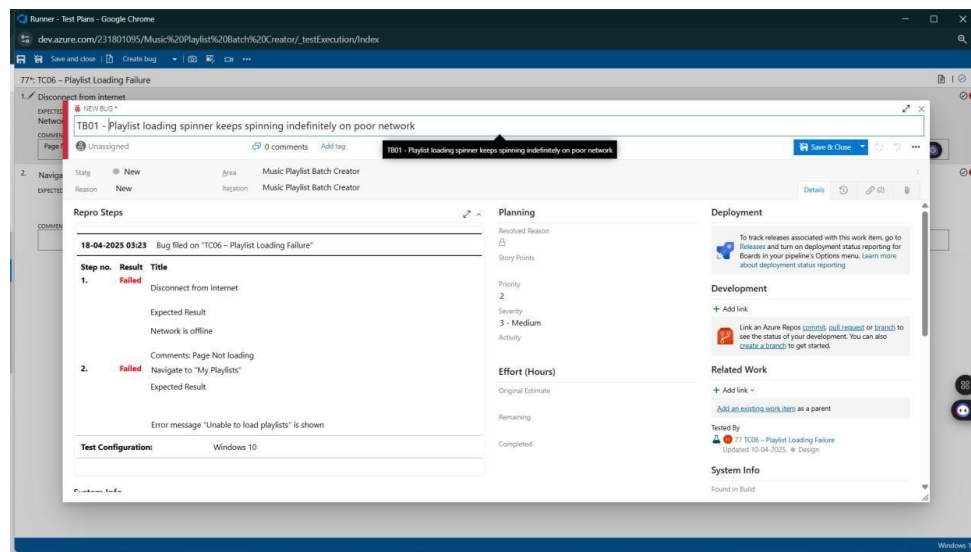
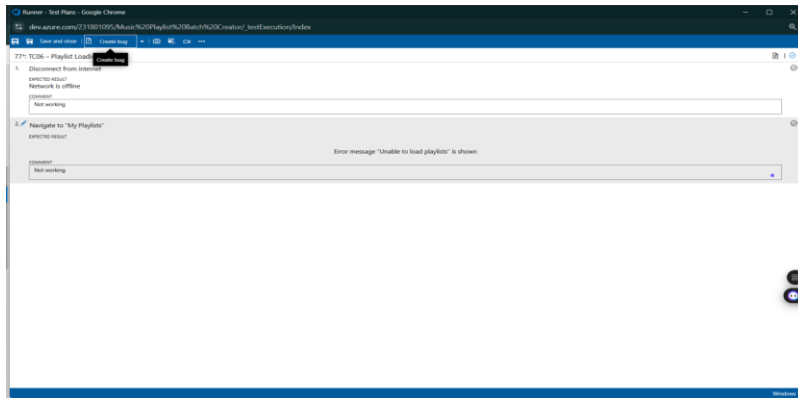
4. Installation of test



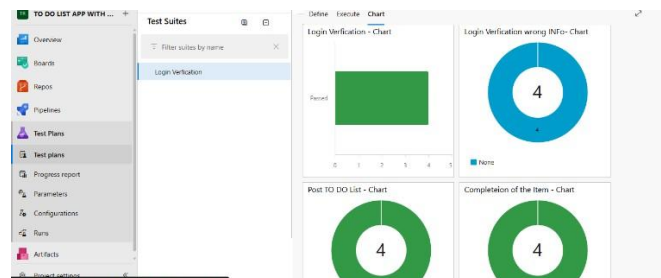
5. Running the test cases

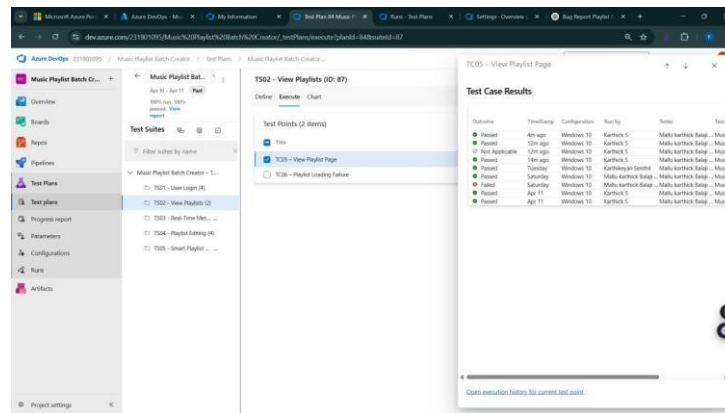


6. Creating the bug



Test Case Results:





Result:

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

Aim:

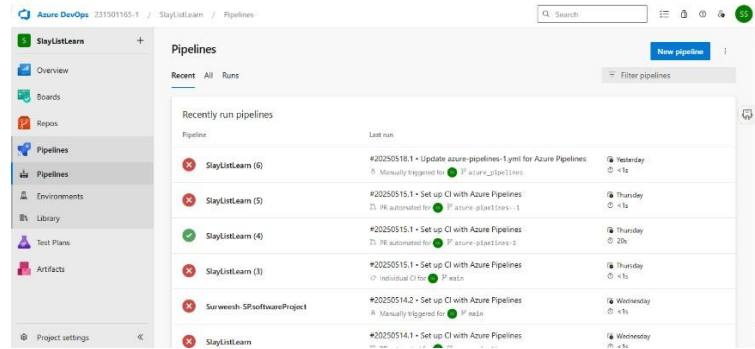
To create an Azure Pipeline resource and run the performance and integrate the targeted endpoint.

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:

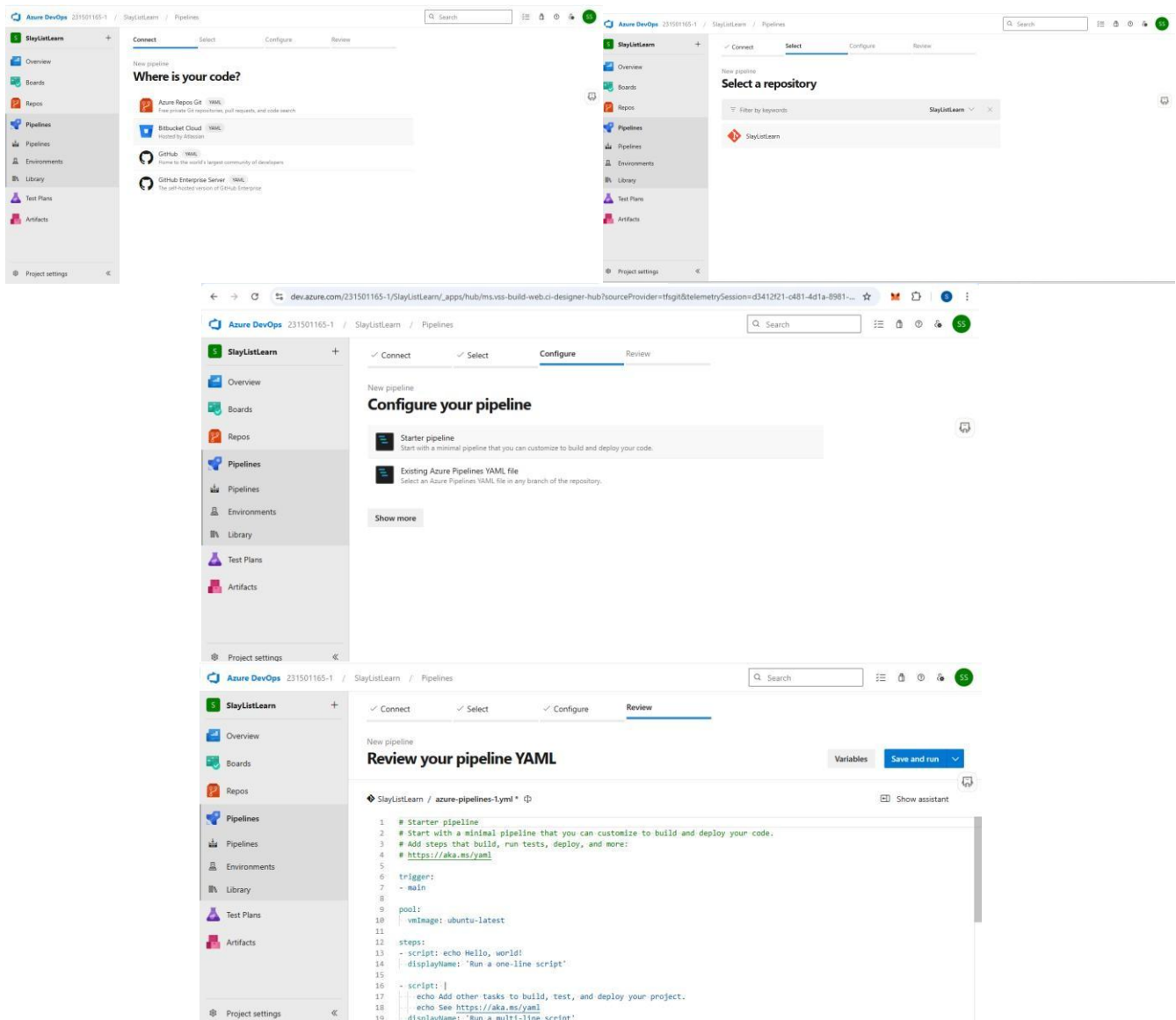
1. **Sign in to Azure DevOps Portal**
Go to <https://dev.azure.com> and log in with your Microsoft account.
2. **Navigate to Your Project**
3. Select your organization and project
4. If you haven't created a project yet, click "**New Project**", fill in the details, and create one.
5. **Create a New Pipeline**
6. On the left sidebar, go to **Pipelines**.
7. Click "**New Pipeline**" at the top right.
8. **Select Your Repository**
9. Choose where your code is stored (e.g., Azure Repos Git, GitHub, Bitbucket, etc.).
10. Authenticate and select the specific repository you want to build.
11. **Configure the Pipeline**
12. Choose a pipeline configuration method (YAML or classic editor).
 - a. **YAML:** Recommended for modern DevOps workflows.
 - b. **Classic editor:** Useful if you prefer a GUI-based configuration.
13. **Set Up Your Build Pipeline**
14. If using YAML:
 - a. Either let Azure generate a template based on your code, or write your own `azure-pipelines.yml` file.
15. If using classic:
 - a. Add tasks such as build, test, and deploy using the step-by-step editor.
16. **Save and Run the Pipeline**
17. Click "**Save and run**" to execute your pipeline for the first time.
18. Review the configuration and click "**Run**".
19. **Monitor the Pipeline Execution**
20. You'll be redirected to the run summary where you can see logs and results for each job and step.
21. *(Optional)* **Add Triggers and Variables**
22. Set up CI/CD triggers, environment variables, and other configurations to fine-tune your pipeline behavior.

Creating Pipelines:



This screenshot shows the 'Recently run pipelines' table in the Azure DevOps interface. The table lists several pipeline runs for 'SlayListLearn' and 'Surveyor-SPOffshoreProject'. Each row includes a pipeline name, a status icon (green for success, red for failure), a description of the trigger and steps, and the last run time.

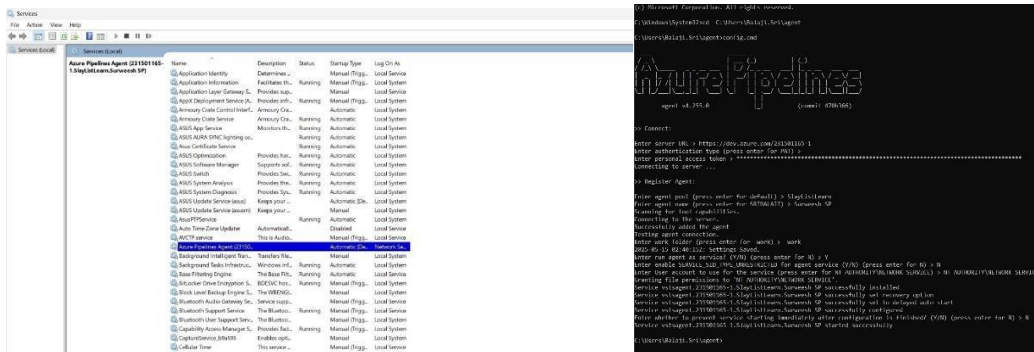
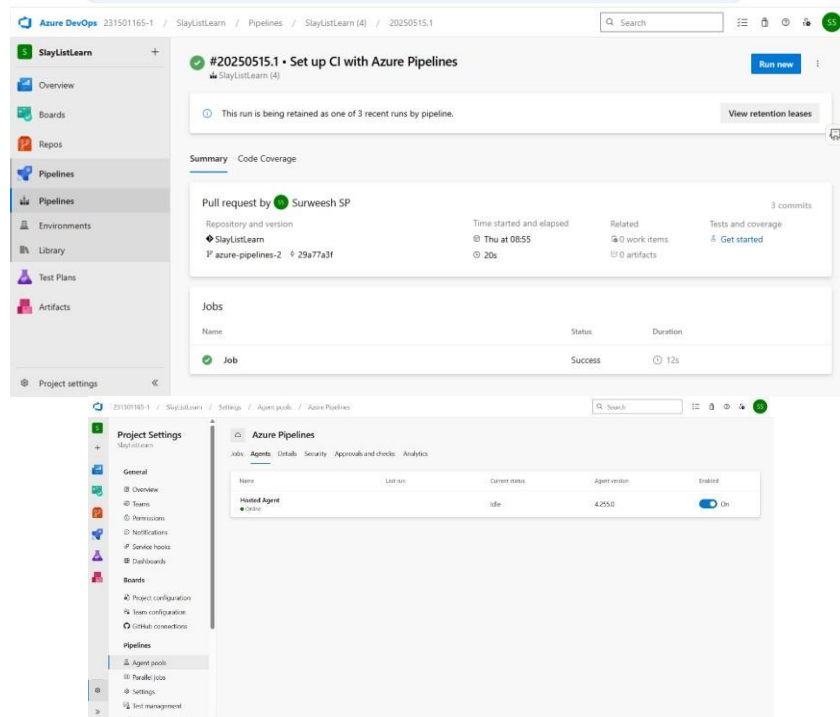
Pipeline	Last run
SlayListLearn (6)	#20250518.1 • Update azure-pipelines-1.yml for Azure Pipelines A: Manually triggered for azure-pipelines-1
SlayListLearn (5)	#20250515.1 • Set up CI with Azure Pipelines T1: PR automated for azure-pipelines-1
SlayListLearn (4)	#20250515.1 • Set up CI with Azure Pipelines T1: PR automated for azure-pipelines-2
SlayListLearn (3)	#20250515.1 • Set up CI with Azure Pipelines Q: Individual CI for azsa
Surveyor-SPOffshoreProject	A: Manually triggered for azsa
SlayListLearn	#20250514.1 • Set up CI with Azure Pipelines



The following screenshots illustrate the process of creating a new pipeline in Azure DevOps:

- Where is your code?**: Shows the initial selection screen with options for Azure Repos Git, Bitbucket Cloud, GitHub, and GitHub Enterprise Server.
- Select a repository**: Shows the selection of the 'SlayListLearn' repository from the list.
- Configure your pipeline**: Shows the 'Configure your pipeline' screen with options for 'Starter pipeline' and 'Existing Azure Pipelines YAML file'.
- Review your pipeline YAML**: Shows the 'Review your pipeline YAML' screen with the generated YAML code for the 'Starter pipeline'.

```
1 # Starter pipeline
2 # Start with a minimal pipeline that you can customize to build and deploy your code.
3 # Add steps that build, run tests, deploy, and more:
4 # https://aka.ms/yaml
5
6 trigger:
7 - main
8
9 pool:
10 - vmImage: ubuntu-latest
11
12 steps:
13 - script: echo Hello, world!
14   displayName: 'Run a one-line script'
15
16 - script: |
17   echo Add other tasks to build, test, and deploy your project.
18   echo See https://aka.ms/yaml
19   displayName: 'Run a multi-line script'
```



Result:

Successfully created the Azure Pipelines resource and executed a performance of the specified endpoint.

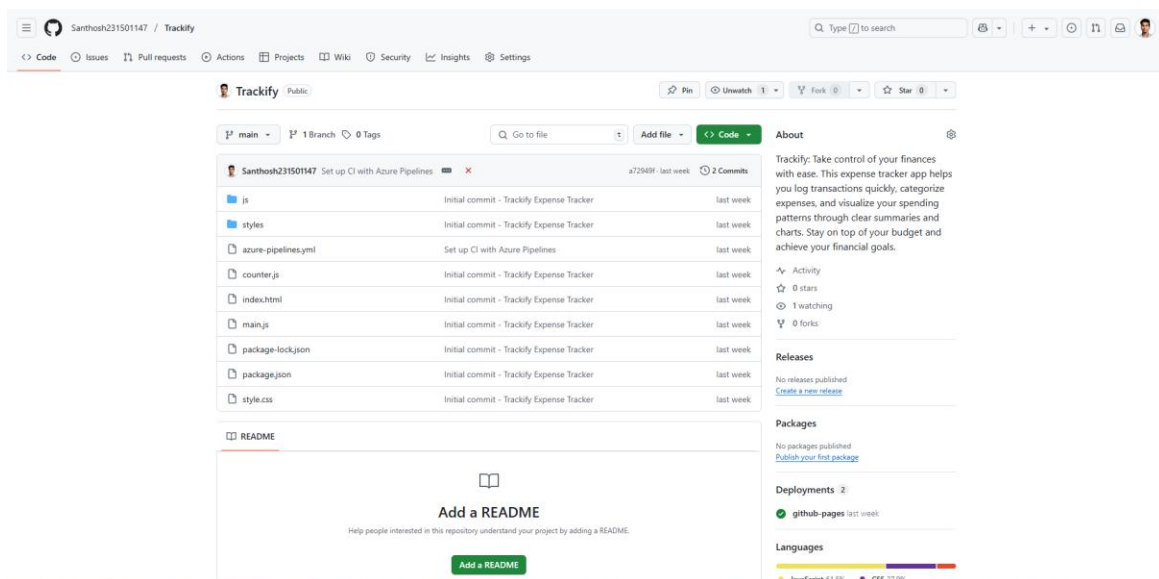
EXP NO: 10

GITHUB: PROJECT STRUCTURE & NAMING CONVENTIONS

Aim:

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

GitHub Project Structure



Result:

The GitHub repository clearly displays the organized project structure and consistent naming conventions, making it easy for users and contributors to understand and navigate the codebase.