



RAJALAKSHMI
ENGINEERING COLLEGE
An AUTONOMOUS Institution
Affiliated to ANNA UNIVERSITY, Chennai

**DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND
DATA SCIENCE LAB MANUAL**

CS23432 – Software Construction

(REGULATION 2023)

RAJALAKSHMI ENGINEERING COLLEGE
Thandalam, Chennai-602015

Name: Jerlin Rose V

Register No: 231801071

Year / Branch / Section: 2nd / AI&DS / FA

Semester: IV

Academic Year: 2024 - 2025



RAJALAKSHMI
ENGINEERING COLLEGE
An AUTONOMOUS Institution
Affiliated to ANNA UNIVERSITY, Chennai

BONAFIDE CERTIFICATE

NAME V. Jeelin Rose

ACADEMIC YEAR 2024 - 2025 SEMESTER IV BRANCH A.I.E.E.D.S

UNIVERSITY REGISTER No. 2116231801071

Certified that this 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.....

External Examiner

Internal Examiner

INDEX

Name: V. Jashin Rose Branch: ATSPDS Sec: FA Roll No: 2318 D1041

INDEX

S.No.	Date	Title
1.	21/1/25	Azure Devops Environment Setup.
2.	21/1/25	Azure Devops Project Setup and User Story Management.
3.	28/1/25	Setting Up Epics, Features, And User Stories for Project Planning.
4.	11/2/25	Sprint Planning.
5.	18/2/25	Poker Estimation.
6.	25/2/25	Designing Class and Sequence Diagrams for Project Architecture.
7.	04/3/25	Designing Architectural and ER Diagrams for Project Structure.
8.	25/3/25	Testing – Test Plans and Test Cases.
9.	15/4/25	Load Testing and Pipelines.
10.	22/4/25	GitHub: Project Structure & Naming Conventions.

EXP NO: 1

AZURE DEVOPS ENVIRONMENT SETUP

Aim:

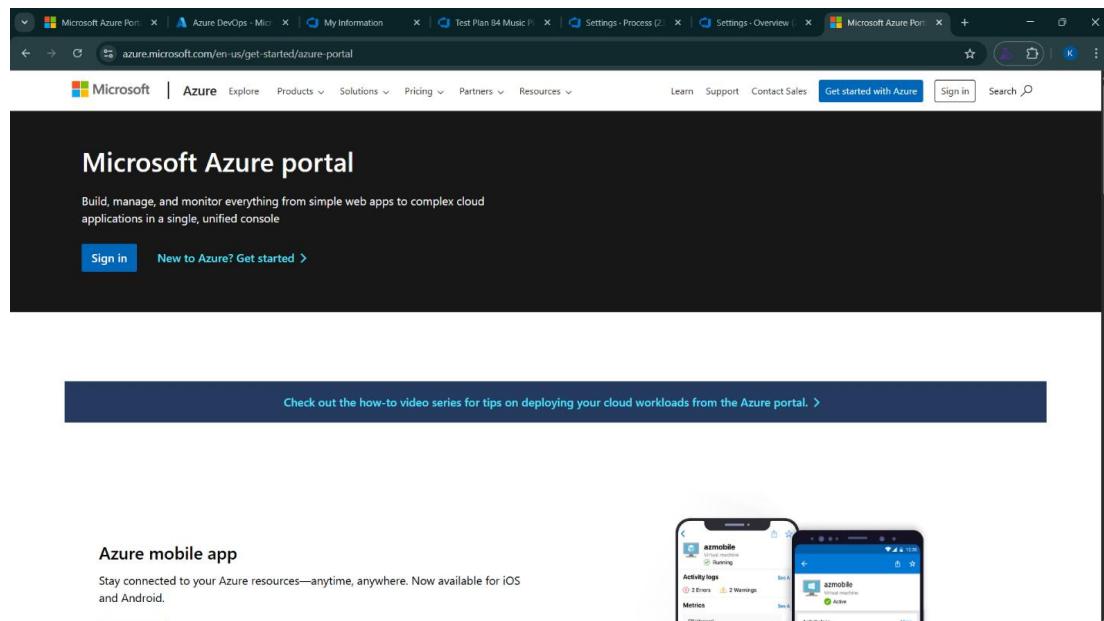
To set up and access the Azure DevOps environment by creating an organization through the Azure portal.

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>



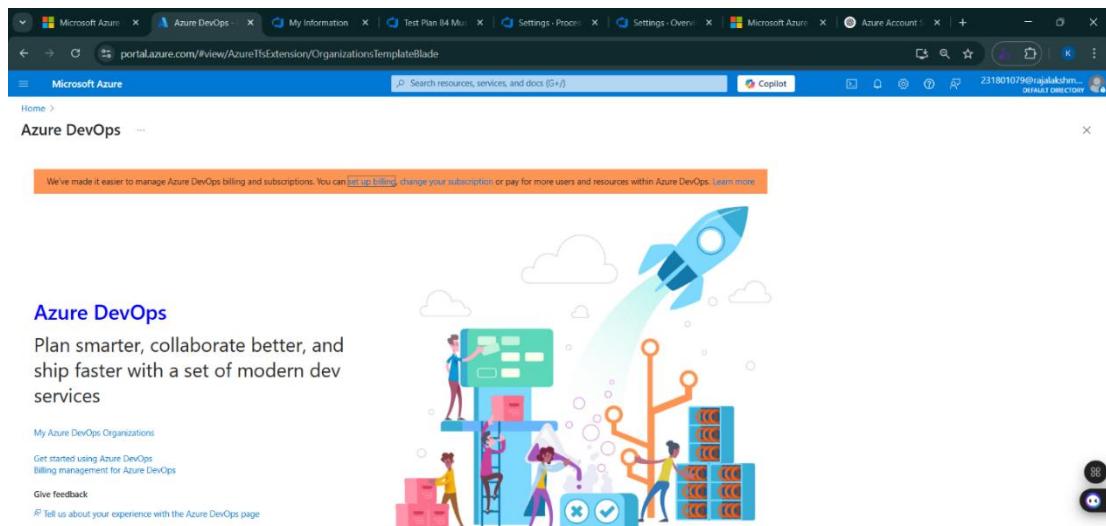
2. Azure home page

The screenshot shows the Microsoft Azure home page. At the top, there's a navigation bar with tabs like Microsoft Azure, Azure DevOps, My Information, Test Plan 84 M... (highlighted), Settings - Process, Settings - Overview, Microsoft Azure, and Azure Account. Below the navigation bar is the search bar with the placeholder "Search resources, services, and docs (S+)" and a Copilot button. The main content area is titled "Azure services" and includes a "Create a resource" button and icons for Azure DevOps organizations, Subscriptions, Dashboard hub, Resource groups, Azure Load Testing, Quickstart Center, Azure AI services, Kubernetes services, and More services. Below this is a "Resources" section with a "Recent" tab showing "Music" (Azure Load Testing) and "Music_playlist_Batch_Creator" (Resource group), both last viewed 3 days ago. There's also a "See all" link. A "Navigate" section has links for Subscriptions, Resource groups, All resources, and Dashboard. A "Tools" section includes Microsoft Learn, Azure Monitor, Microsoft Defender for Cloud, and Cost Management. A "Useful links" section lists Azure mobile app and other Azure services.

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

This screenshot is similar to the previous one but with a search bar containing the text "DevOps". The search results are displayed below the search bar, under sections for "Services", "Marketplace", and "Documentation". The "Services" section shows Azure Native New Relic Service, Managed DevOps Pools, Azure DevOps organizations, and Azure Native Dynatrace Service. The "Marketplace" section shows Static Web App, Rocky Linux 9, Build Agents for Azure DevOps, and InfluxDB Cloud (Official Version). The "Documentation" section includes links to DevOps architecture design, Secure your Azure DevOps, Course AZ-400T00-A: Designing and Implementing Microsoft DevOps solutions, Managed DevOps Pools Overview, Microsoft Entra ID, Azure DevOps, and Continue searching in Microsoft Entra ID. The rest of the page layout is identical to the first screenshot.

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



Result:

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

EXP NO: 2

AZURE DEVOPS PROJECT SETUP AND USER STORY MANAGEMENT

Aim:

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

1.Create An Azure Account

The screenshot shows the final step of creating an Azure DevOps organization. It features the Azure DevOps logo and the email address 231801071@rajalakshmi.edu.in with a 'Switch directory' link. A large bold heading 'Almost done...' is centered. Below it, a field asks 'Name your Azure DevOps organization *' with the value 'dev.azure.com/ 2116231801071'. Another field asks 'We'll host your projects in' with the value 'India'. A CAPTCHA section displays the text 'S5V GX' over a grid. A text input field for the CAPTCHA answer is empty. A blue 'Continue' button is at the bottom right.

2.Create the First Project in Your Organization

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

b. On the organization's **Home page**, click on the **New Project** button.

c. Enter the project name, description, and visibility options:

Name: Choose a name for the project (e.g., **LMS**).

Description: Optionally, add a description to provide more context about the project.

Visibility: Choose whether you want the project to be **Private** (accessible only to those invited) or **Public** (accessible to anyone).

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

Create new project

Project name *

Description

Visibility

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

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

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

Advanced

Version control ②

Work item process ②

Cancel **Create**

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

The screenshot shows the Azure DevOps Organizations home page. On the left, there's a sidebar with the user's profile picture (JV), name (Jerlin Rose V), email (231801071@rajalakshmi.edu.in), and Microsoft account information. Below this is a section for 'Visual Studio Dev Essentials'. The main content area is titled 'Azure DevOps Organizations' and shows two projects: 'Expense Tracker App' (Owner) and 'dev.azure.com/2318010710991 (Owner)'. There are 'Actions' like 'Open in Visual Studio' and 'New project'. A 'Create new organization' button is at the top right. The browser toolbar at the bottom includes icons for weather, search, and various extensions.

4. Project dashboard

The screenshot shows the Azure DevOps Project Summary page for the 'Expense Tracker App'. The left sidebar lists project components like EA, Key Features, Use Cases, Technical Stack, and Future Enhancements. The main content area displays 'Key Features' (Expense Logging, Budget Management, Data Visualization, Recurring Expenses, Notifications, Multi-Device Access, Expense Sharing), 'Work items created' (4), 'Work items annotated' (4), and a 'Members' section with four team members (JV, MN, JB, KR). The browser toolbar at the bottom is visible.

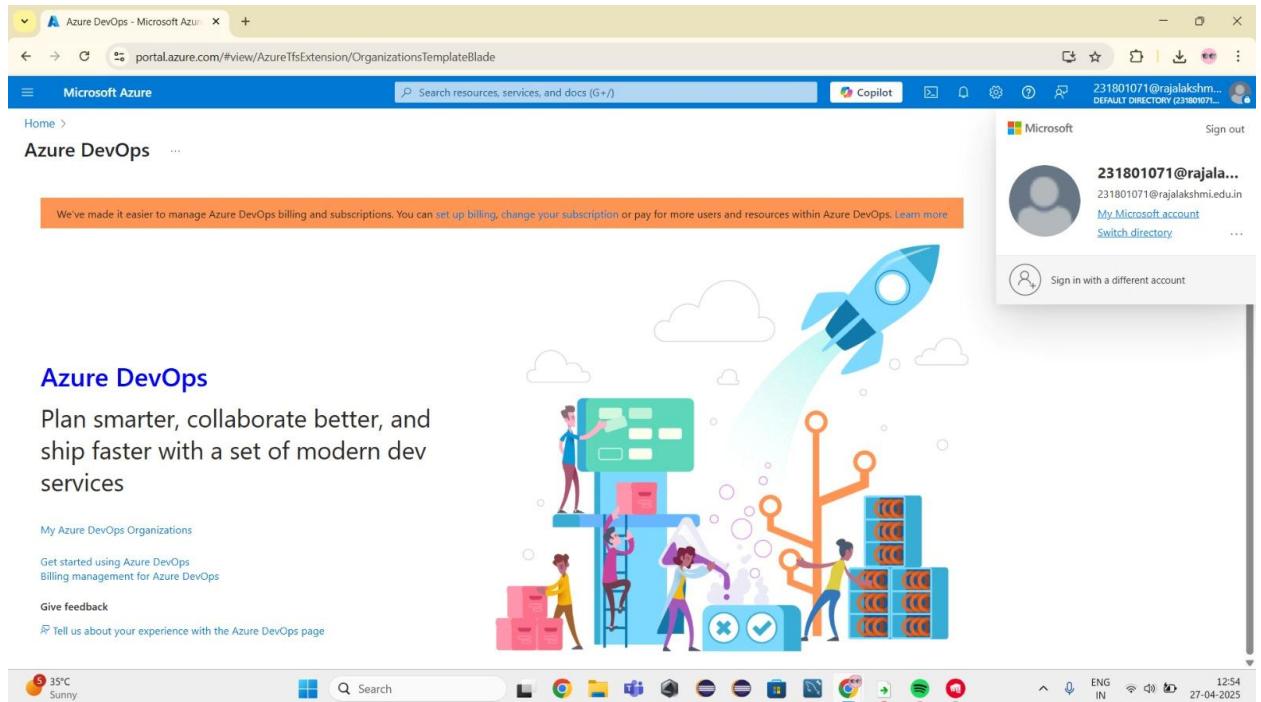
5. To manage user stories:

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

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

The screenshot shows the Azure Boards interface for the 'Expense Tracker App Team' epic. The left sidebar has icons for Epics, Stories, Tasks, and Bugs. The main area is titled 'Backlog' and shows a table of work items. The columns are Order, Work Item Type, Title, State, Effort, Business Area, and Tags. There are six epic items listed:

Order	Work Item Type	Title	State	Effort	Business Area	Tags
1	Epic	> User Management and Authentication	New		Business	
2	Epic	> Expense Management	New		Business	
3	Epic	> Reporting and Analytics	New		Business	
4	Epic	> Budgeting and Alerts	New		Business	
5	Epic	> Data Synchronization and Backup	New		Business	
6	Epic	> User Interface and Experience (UI/UX)	New		Business	



Result:

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

EXP NO: 3

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

Aim:

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

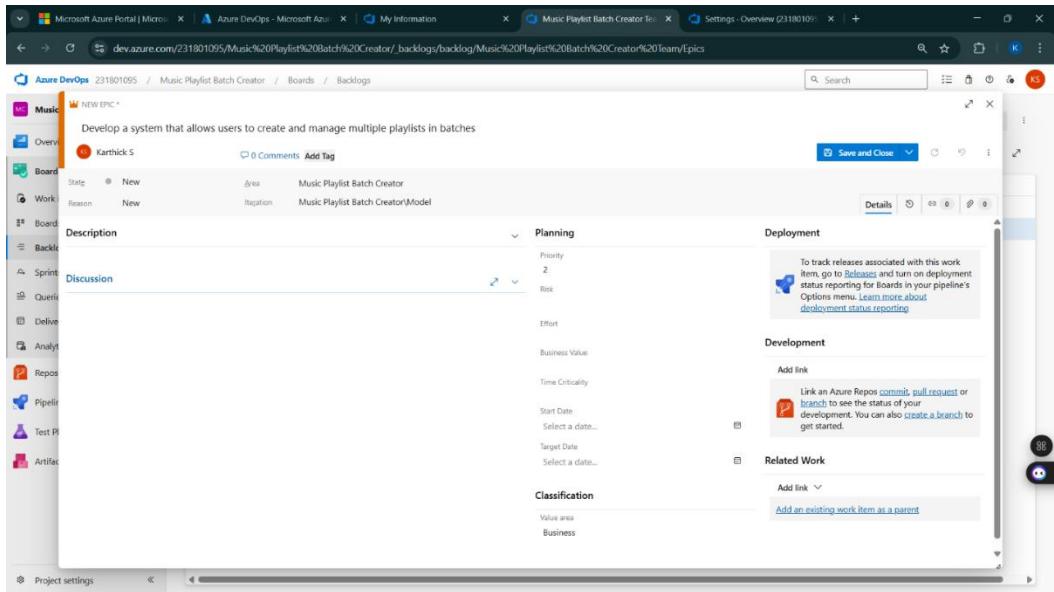
Create Epic, Features, User Stories, Task

The screenshot shows the Azure DevOps interface for the 'Expense Tracker App Team' backlog. The board displays a hierarchy of work items: Epics, Features, and Tasks. The 'Backlog' tab is selected, showing the following structure:

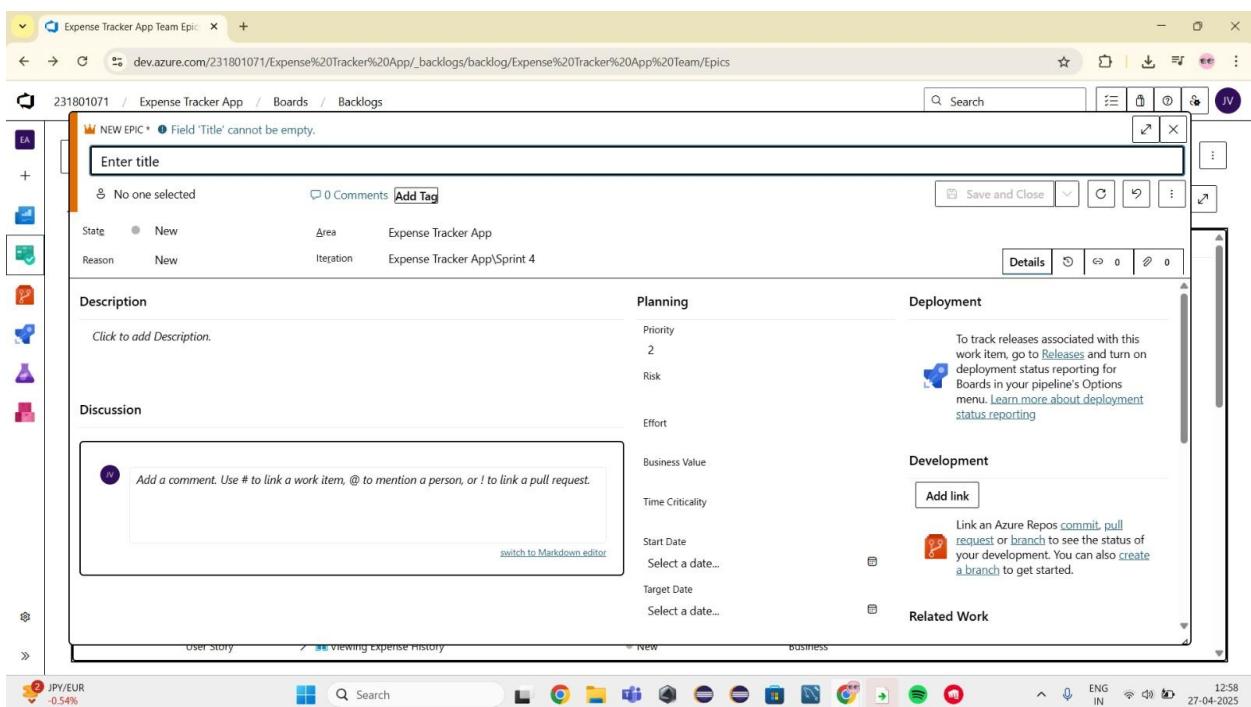
Order	Work Item Type	Title	State	Effort	Business Area	Tags
1	Epic	> 🚀 User Management and Authentication	New		Business	
2	Epic	> 🚀 Expense Management	New		Business	
	Feature	> 🎯 Add New Expense	New		Business	
	Feature	> 🎯 Edit Expense	New		Business	
	Feature	> 🎯 Delete Expense	New		Business	
	Feature	> 🎯 View Expense History	New		Business	
	Feature	> 🎯 Categorize Expenses	New		Business	
3	Epic	> 🚀 Reporting and Analytics	New		Business	
4	Epic	> 🚀 Budgeting and Alerts	New		Business	
5	Epic	> 🚀 Data Synchronization and Backup	New		Business	
6	Epic	> 🚀 User Interface and Experience (UI/UX)	New		Business	

The interface includes various navigation and search tools typical of a project management application.

1.Fill in Epics



2.Fill in Features



3. Fill in User Story Details

The screenshot shows the Azure DevOps interface for the 'Expense Tracker App Team'. A new user story is being created, indicated by the title 'NEW USER STORY *' and the error message 'Field 'Title' cannot be empty.' in the top bar. The 'Backlogs' tab is selected. The user story details are as follows:

- Title:** Enter title
- Comments:** 0 Comments, Add Tag
- State:** New
- Area:** Expense Tracker App
- Reason:** New
- Iteration:** Expense Tracker App\Sprint 4

The main body of the dialog is divided into sections:

- Description:** Click to add Description.
- Acceptance Criteria:** Click to add Acceptance Criteria.
- Discussion:** Add a comment. Use # to link a work item, @ to mention a person, or ! to link a pull request.
- Planning:** Story Points, Priority (2), Risk.
- 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.
- Classification:** Value area (Business).
- 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.
- Planning Poker:**
- Related Work:** Save and re-open the new work item.

At the bottom of the dialog, there are buttons for Save and Close, and a 'Details' section. The footer of the browser window shows the URL dev.azure.com/231801071/Expense%20Tracker%20App/_backlogs/backlog/Expense%20Tracker%20App%20Team/Stories, the date 27-04-2025, and the time 12:59.

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 Expense Tracker AppProject.

Sprint Planning

Sprint 1

The screenshot shows the Azure DevOps Backlog board for the 'Expense Tracker App Team' project. The board is currently displaying the 'Sprint 1' backlog. The columns are Order, Title, State, Assigned To, and Rema... (partially visible). There are four user stories listed:

Order	Title	State	Assigned To	Rema...
1	As a user, I want to authenticate using OAuth (Google, Microsoft)	New	JUSTIN BREETLEE	
2	As a user, I want my data to be encrypted for security.	New	Kabithvajan RV	
3	As a user, I want to register and log in securely	New	Mahisha Param...	
4	As a user, I want to reset my password if I forget it.	New	Jerlin Rose V	

The board interface includes various buttons for managing work items and column options. At the bottom, there is a date range indicator for February 25 - March 10, which spans 14 work days. The overall interface is clean and modern, typical of a cloud-based project management tool.

Sprint 2

The screenshot shows the Azure Boards interface for the 'Expense Tracker App Team' project. The 'Backlog' tab is selected. A card for 'Sprint 2' is open, showing four user stories:

Order	Title	State	Assigned To	Remainder
1	As a user, I want to filter expenses by category and date.	New	JUSTIN BREETLEE	
2	As a user, I want to validate my expense input fields.	New	Kabithvajan RV	
3	As a user, I want to categorize my expenses.	New	Jerlin Rose V	
4	As a user, I want to add, edit, and delete expenses.	New	Mahisha Param...	

The sprint duration is March 11 - March 24, 14 work days. The browser address bar shows the URL: dev.azure.com/231801071/Expense%20Tracker%20App/_sprints/backlog/Expense%20Tracker%20App%20Team/Expense%20Tracker%20App/Sprint%202.

Sprint 3

A screenshot of the Azure DevOps Backlog board for the 'Expense Tracker App Team'. The board shows a list of user stories in 'Sprint 3' from March 25 to April 7, which spans 14 work days. The backlog items are:

Order	Title	State	Assigned To	Remainder
1	As a user, I want to get alerts when I exceed my budget.	New	Jerlin Rose V	
2	As a user, I want to set a monthly budget.	New	JUSTIN BREETLEE	
3	As a user, I want to export my expenses as CSV/PDF.	New	Mahisha Param...	
4	As a user, I want to see a dashboard with my expense sum...	New	Kabithvajan RV	

The interface includes a sidebar with team avatars and a search bar at the top. A Snipping Tool window is visible in the foreground, showing a screenshot of the application's UI.

Sprint 4

A screenshot of the Azure DevOps Backlog board for the 'Expense Tracker App Team'. The board shows a list of developer tasks in 'Sprint 4' from April 8 to April 21, which spans 14 work days. The backlog items are:

Order	Title	State	Assigned To	Remainder
1	As a developer, I want to deploy the application to Azure.	New	Jerlin Rose V	
2	As a developer, I want to set up a CI/CD pipeline in Azure D...	New	Mahisha Param...	
3	As a developer, I want to connect the backend to the databa...	New	JUSTIN BREETLEE	
4	As a developer, I want to store expenses in an Azure databa...	New	Kabithvajan RV	

The interface includes a sidebar with team avatars and a search bar at the top. A Snipping Tool window is visible in the foreground, showing a screenshot of the application's UI.

Result:

The Sprints are created for the Expense Tracker AppProject.

EXP NO: 5

POKER ESTIMATION

Aim:

Create Poker Estimation for the user stories - Expense Tracker AppProject.

Poker Estimation

The screenshot shows a Microsoft Azure DevOps interface for a project titled "Expense Tracker App Team Sprint". A specific user story, "USER STORY 104", is displayed. The story title is "As a developer, I want to deploy the application to Azure." It was created by "Jerlin Rose V" and has "0 Comments" and "Add Tag" options. The story is in the "New" state and is part of the "Expense Tracker App" area and "Expense Tracker App/Sprint 4" iteration. The "Planning" section shows "Story Points" (Priority: 2). The "Deployment" section provides instructions on tracking releases. The "Classification" section indicates the value area is "Business". The "Development" section includes a "Planning Poker" button and a "Related Work" link. The interface also shows a sidebar with various project management icons and a bottom navigation bar with system status indicators like weather and system logs.

Result:

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

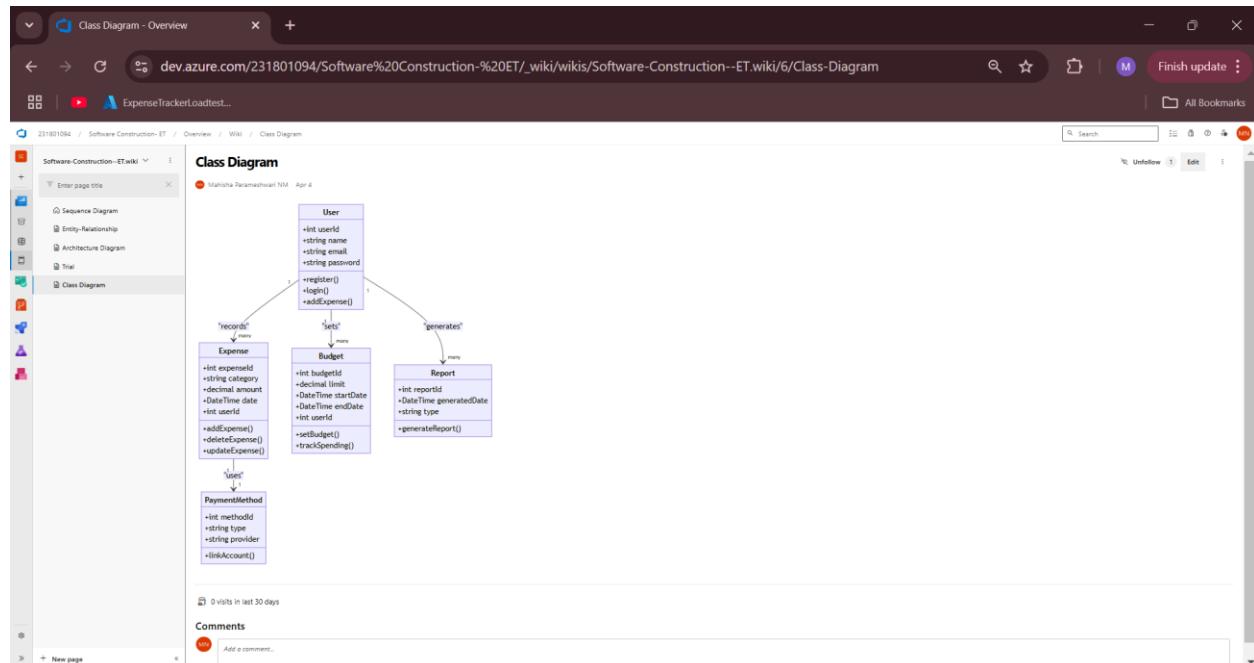
EXP NO: 6

DESIGNING CLASS AND SEQUENCE DIAGRAMS FOR PROJECT ARCHITECTURE

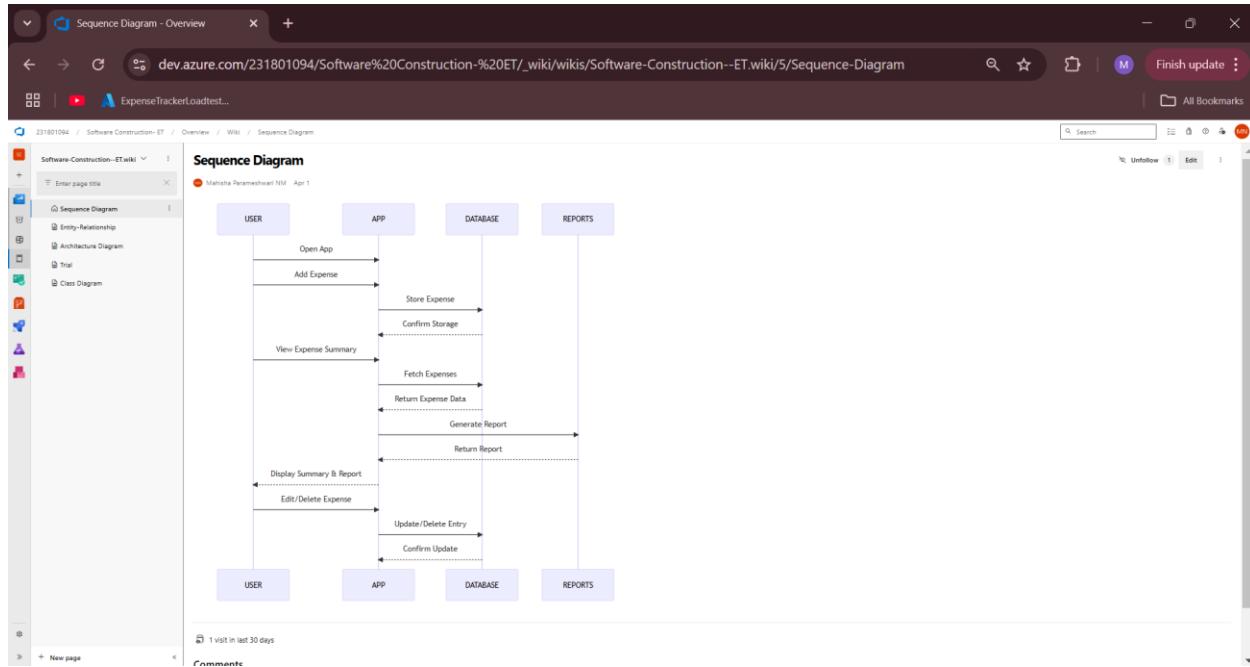
Aim:

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

6A. Class Diagram



6B. Sequence Diagram



Result:

The Class Diagram and Sequence Diagram is designed Successfully for the Music Playlist Batch Creator.

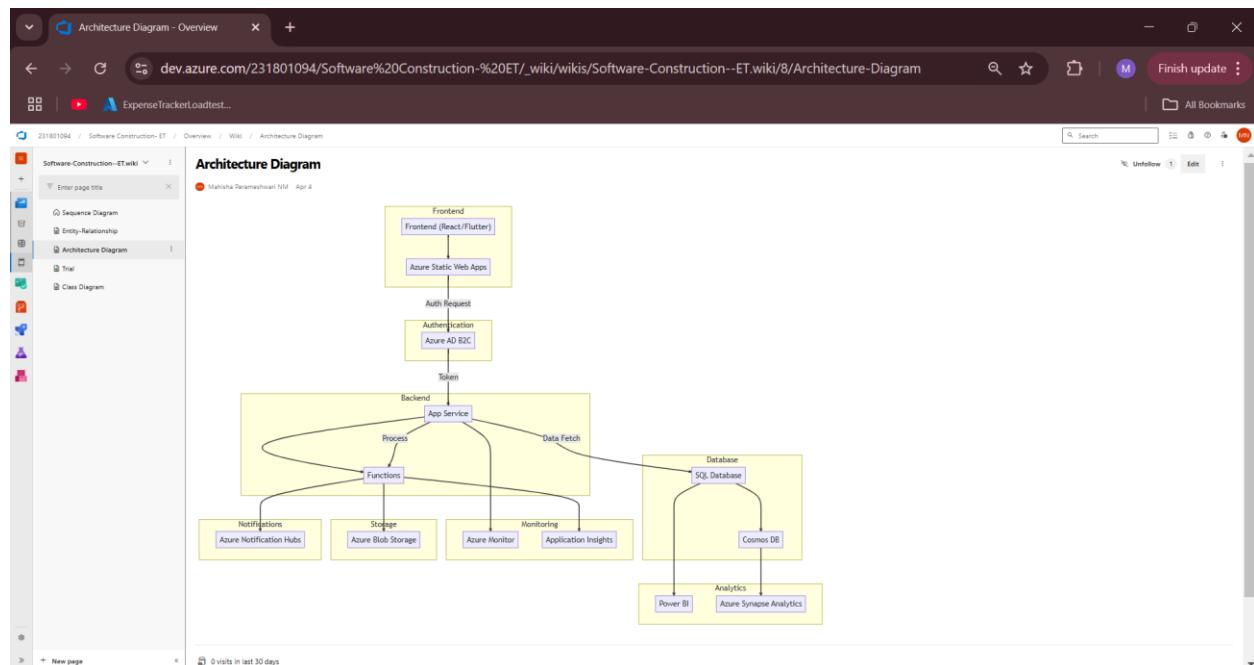
EXP NO: 7

DESIGNING ARCHITECTURAL AND ER DIAGRAMS FOR PROJECT STRUCTURE

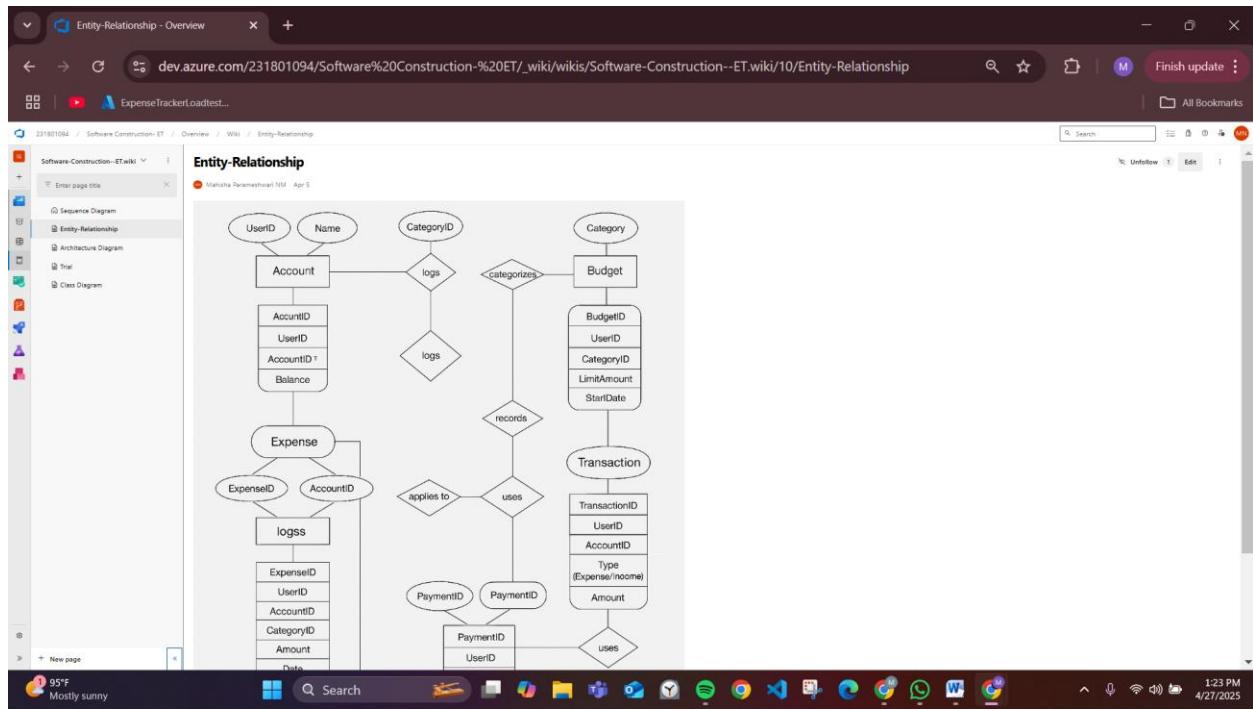
Aim:

To Design an Architectural Diagram and ER Diagram for the given Project.

7A. Architectural Diagram



7B.ER Diagram



Result:

The Architecture Diagram and ER Diagram is designed Successfully for the Expense Tracker

EXP NO: 8

TESTING – TEST PLANS AND TEST CASES

Aim:

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

Test Planning and Test Case

Test Case Design Procedure for Expense Tracker App

1. Understand Core Features of the Application

- User Signup & Login

Secure registration, login, and password recovery.

- Adding and Managing Expenses

Users can add, edit, and delete expenses.

- Budget Setting and Tracking

Set budgets for different categories and monitor spending.

- Data Visualization

Graphical view of expenses (pie charts, bar graphs).

- Expense Sharing and Exporting

Export expenses in PDF/Excel formats and share.

- Notifications and Alerts

Alerts for budget overruns and upcoming bills.

- Recurring Expense Management

Handle repeating expenses like rent or subscriptions.

2. Define User Interactions

- Each test case simulates a real-world user action.

(e.g., user adds a new expense, sets a budget, gets notified after exceeding budget).

3. Design Happy Path Test Cases

- Focused on validating that all core features work correctly under normal conditions.

- Example:
 - User logs in successfully.
 - User adds an expense and it reflects in monthly summary.
 - User gets a budget notification when the limit is exceeded.

4. Design Error Path Test Cases

- Simulate negative or unexpected scenarios to ensure robustness.
- Example:
 - Login fails with wrong credentials.
 - User tries to add an expense without an amount.
 - Budget not updating due to server error.

5. Break Down Steps and Expected Results

- Every test case will clearly outline:
 - Steps (e.g., Navigate to "Add Expense" → Enter details → Save).
 - Expected Outcome (e.g., Expense should appear under selected category).

6. Use Clear Naming and IDs

- Test cases are named and numbered systematically.
- Examples:
 - TC01 – Successful Registration
 - TC07 – Add Expense Without Amount (Error Handling)
 - TC15 – Budget Exceeded Notification Display
- Easy to trace back to user stories and features.

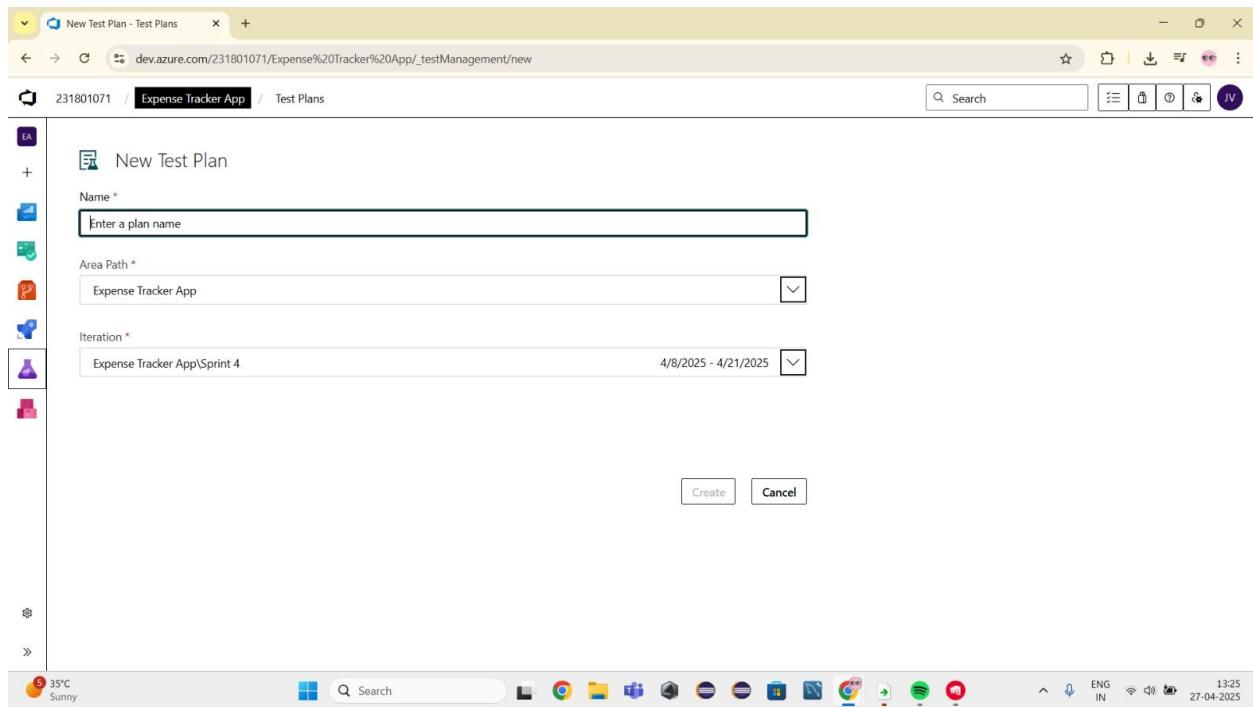
7. Separate Test Suites

- Organize test cases into functional groups:
 - Authentication Suite (Signup, Login, Logout)
 - Expense Management Suite (Add, Edit, Delete)
 - Budget Suite (Set Budget, Notifications)
 - Visualization Suite (Graphs, Reports)
 - Recurring Expenses Suite
- Improves organization and systematic testing in Azure DevOps or any test management tool.

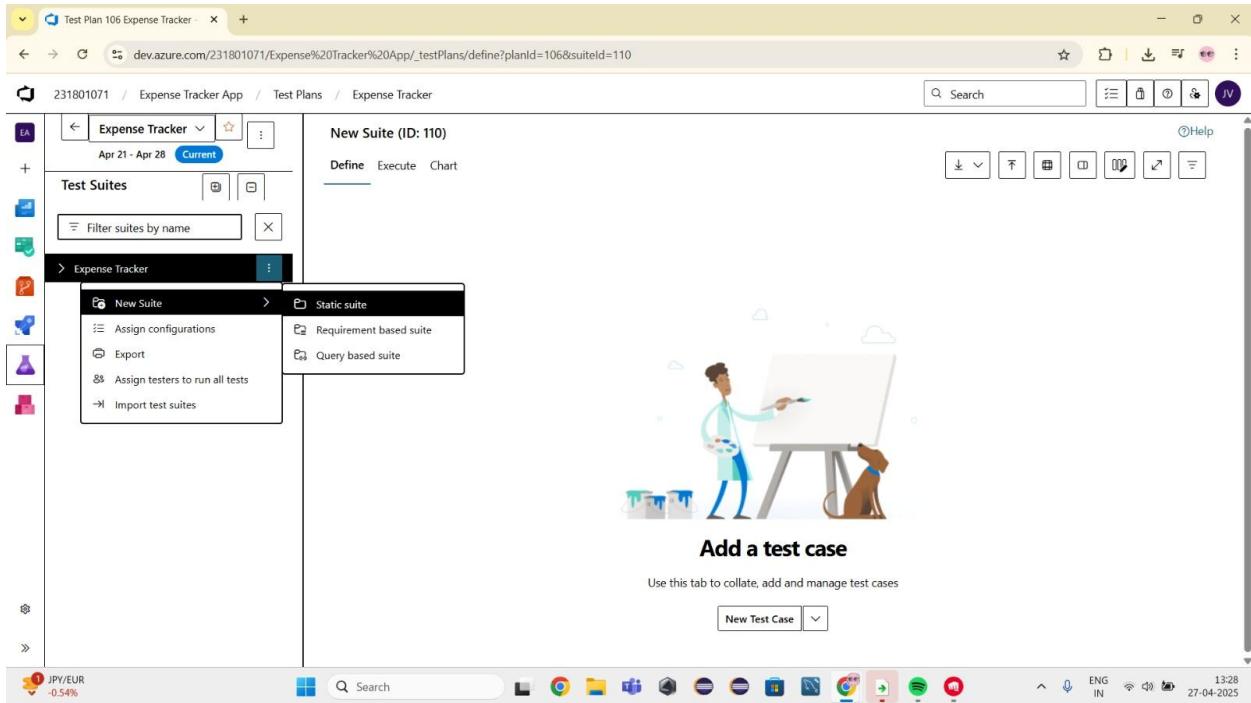
8. Prioritize and Review

- **High Priority:** Core actions like login, add expense, budget alert.
- **Medium Priority:** Visualization, export, sharing.
- **Low Priority:** UI enhancements, optional settings.
- All test cases reviewed for completeness and mapped to 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.

Expense Tracker – Test Plans

4. User Stories

User Story 1: User Sign-Up and Login (ID: 101)

User Story 2: Add and View Expenses (ID: 102)

User Story 3: Categorize Expenses (ID: 103)

User Story 4: Set Monthly Budget (ID: 104)

User Story 5: Generate Reports (ID: 105)

Test Suites

Test Suite 1: User Login & Sign-Up

TC01 – Successful Sign Up

- **Action:**
Go to the Sign-Up page → Enter valid name, email, 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

TC02 – Successful Login

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

TC03 – Sign Up with Existing Email

- **Action:**
Go to the Sign-Up page → Enter a name and an already registered email → Click "Sign Up"
- **Expected Results:**
Fields accept data. Error message "Email already registered" is displayed.
- **Type:** Error Path

TC04 – Login with Wrong Password

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

Test Suite 2: Add & Manage Expenses

TC05 – Add Expense Successfully

- **Action:**
Enter ₹100 for Food, category: Food, date → Click Save

- **Expected Results:**
Expense saved and shown in the list.
- **Type:** Happy Path

TC06 – Add Expense with Missing Amount

- **Action:**
Leave amount empty → Click Save
- **Expected Results:**
Error message "Amount is required" is shown.
- **Type:** Error Path

TC07 – Edit Expense Successfully

- **Action:**
Navigate to an existing expense → Change ₹100 to ₹120 → Save
- **Expected Results:**
Expense updated and shown with new amount.
- **Type:** Happy Path

TC08 – Edit Expense with Invalid Data

- **Action:**
Leave "Category" empty and save the expense
 - **Expected Results:**
Error message "Category cannot be empty" is shown.
 - **Type:** Error Path
-

Test Suite 3: Budget Management

TC09 – Set Budget Successfully

- **Action:**
Go to Budget section → Enter ₹5000 as the budget → Click Save
- **Expected Results:**
Budget saved and displayed on the dashboard.
- **Type:** Happy Path

TC10 – Set Budget with Invalid Input

- **Action:**
Enter text instead of a number → Click Save
- **Expected Results:**
Error message "Please enter a valid number" is shown.
- **Type:** Error Path

Test Suite 4: Expense Reports

TC11 – Generate Monthly Report

- **Action:**
Click “Generate Report” for April
- **Expected Results:**
Report with total and category-wise breakdown.
- **Type:** Happy Path

TC12 – Report with No Data

- **Action:**
Generate report for a month with no expenses
 - **Expected Results:**
Message “No data available for report” is shown.
 - **Type:** Error Path
-

Test Suite 5: Notifications & Alerts

TC13 – Receive Budget Alert

- **Action:**
Set a budget of ₹5000 → Add ₹6000 expense
- **Expected Results:**
Budget exceeded alert is shown.
- **Type:** Happy Path

TC14 – Notification Failure

- **Action:**
Disable push notifications → Try to trigger a budget alert
- **Expected Results:**
Error message “Failed to send notification” is shown.
- **Type:** Error Path

Creating a Test Cases

A screenshot of the Azure Test Plan interface. The title bar shows "Test Plan 106 Expense Tracker - dev.azure.com/231801071/Expense%20Tracker%20App/_testPlans/define?planId=106&suiteId=107". The main content area displays a "NEW TEST CASE" dialog. The "Title" field contains "Field 'Title' cannot be empty." Below it, there's a "Enter title" input field. On the left, a sidebar lists icons for Test Plan, Test Suite, Test Case, Test Result, and Test Environment. The "Test Case" icon is selected. The "Steps" section has a table with columns: Steps, Action, Expected result, and Attachments. A note says "Click or type here to add a step". The "Deployment" section includes a note about tracking releases and a "Development" section with a "Add link" button. The "Related Work" section also has an "Add link" button. The bottom of the screen shows the Windows taskbar with various pinned icons.

Test case

A screenshot of the Azure Test Plan interface, similar to the previous one but for a different test case. The title bar shows "231801094 / Software Construction- ET / Test Plans / Expense Tracker". The main content area displays a "TEST CASE 31" dialog. The title is "TC13 – Receive Budget Alert". The "Steps" section shows a single step: "1. Set a budget of ₹5000 → Add ₹6000 expense" with the expected result "Budget exceeded alert is shown". The "Deployment" and "Development" sections are present, along with the "Related Work" section. The bottom of the screen shows the Windows taskbar.

The screenshot shows the Microsoft Azure DevOps Test Plan interface. A test case titled "TC14 – Notification Failure" is selected. The test case details include:

- State:** Design
- Area:** Software Construction- ET
- Reason:** New
- Iteration:** Software Construction- ET

The **Steps** section contains one step:

Steps	Action	Expected result	Attachments
1.	Disable push notifications → Try to trigger a budget alert	Error message "Failed to send notification" is shown.	

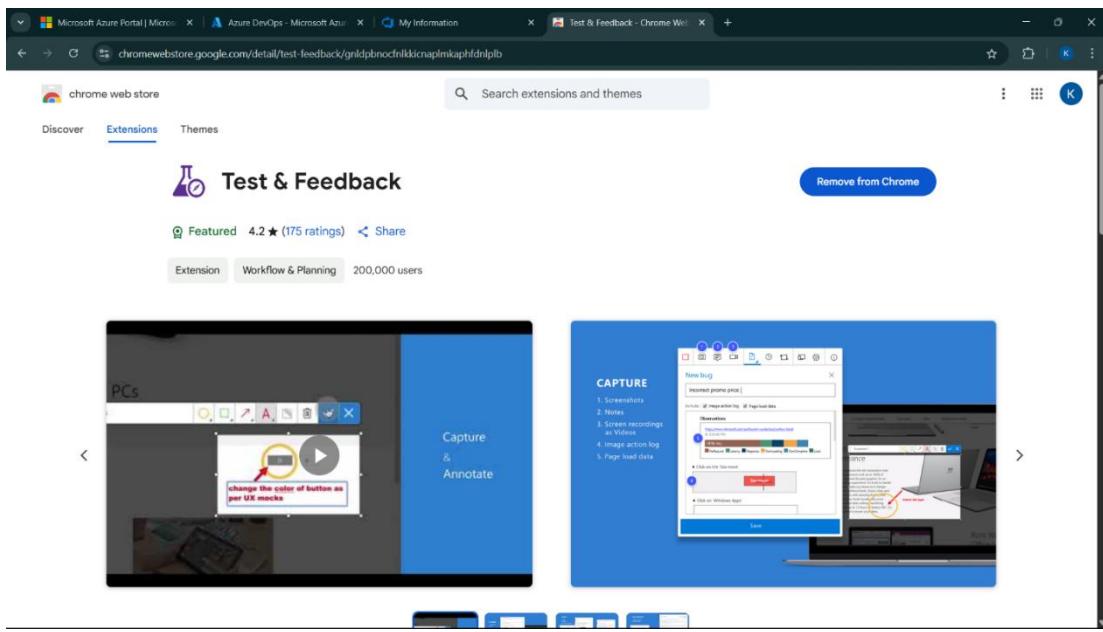
The **Deployment** section includes a note about tracking releases and deployment status reporting. The **Development** section has an "Add link" button. The **Related Work** section also has an "Add link" button and a link to "Add an existing work item as a parent".

4. Installation of test

The screenshot shows the Chrome Web Store page for the "Test & Feedback" extension. The extension is listed as "Featured" with a rating of 4.2 stars from 175 ratings and 200,000 users. It is categorized under "Extension" and "Workflow & Planning". The page features two main screenshots:

- PCs:** Shows a screenshot of a laptop screen with a red annotation box containing the text "change the color of button as per UX needs".
- Capture:** Shows a screenshot of a Windows application window with a "New bug" dialog open, displaying steps like "Screenshot", "Notes", "Screen recordings as Videos", "Image action log", and "Page load data".

An "Add to Chrome" button is visible at the top right of the extension's card.



Test and feedback

Showing it as an extension

A screenshot of a web browser window showing the Azure DevOps Test Plans interface for a project named "Music Playlist Batch Creator". The left sidebar shows navigation options like Overview, Boards, Repos, Pipelines, Test Plans, and Artifacts. The "Test plans" option is selected. In the center, a test plan titled "TS01 - User Login (ID: 86)" is displayed with tabs for Define, Execute, and Chart. Under the "Test Cases (4 items)" section, there are four entries: Title, TC01 - Successful Sign Up, TC02 - Secure Login, TC03 - Sign Up with Existing Email, and TC04 - Login with Wrong Password. A floating sidebar on the right is titled "Extensions" and lists the "Test & Feedback" extension as having "Full access" to the site. Other extensions listed include Copy Text from Picture, Dark Reader, Monica: ChatGPT AI Assist., Selectext: Copy text from V..., and Test & Feedback.

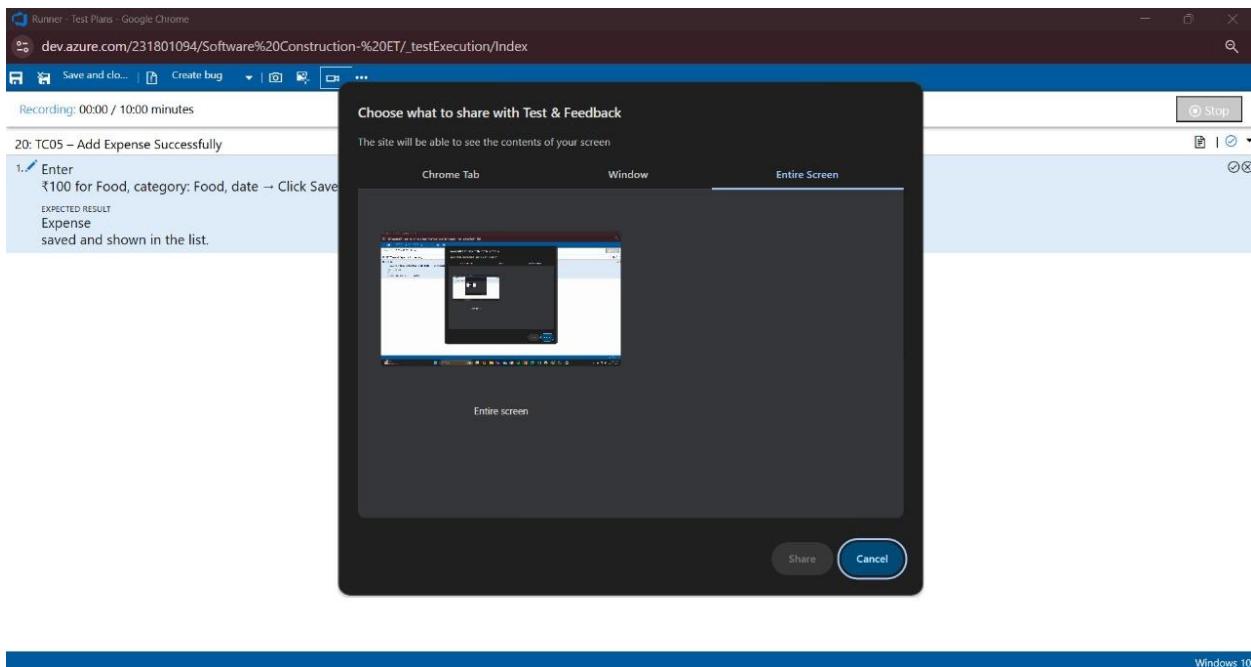
5. Running the test cases

The screenshot shows the Microsoft Test Plan interface. On the left, there's a navigation pane with 'Expense Tracker' selected. In the center, under 'Test Suites', 'Test Suite 5: Notifications & Alerts (ID: 30)' is selected. Below it, 'Test Points (2 items)' are listed: 'TC13 – Receive Budget Alert' and 'TC14 – Notification Failure'. A context menu is open over 'TC14 – Notification Failure', with options like 'View execution history', 'Mark Outcome', 'Run', 'Reset test to active', 'Edit test case', 'Assign tester', and 'View test result'. At the top right, there's a search bar and a help icon. The bottom shows a Windows taskbar with various icons.

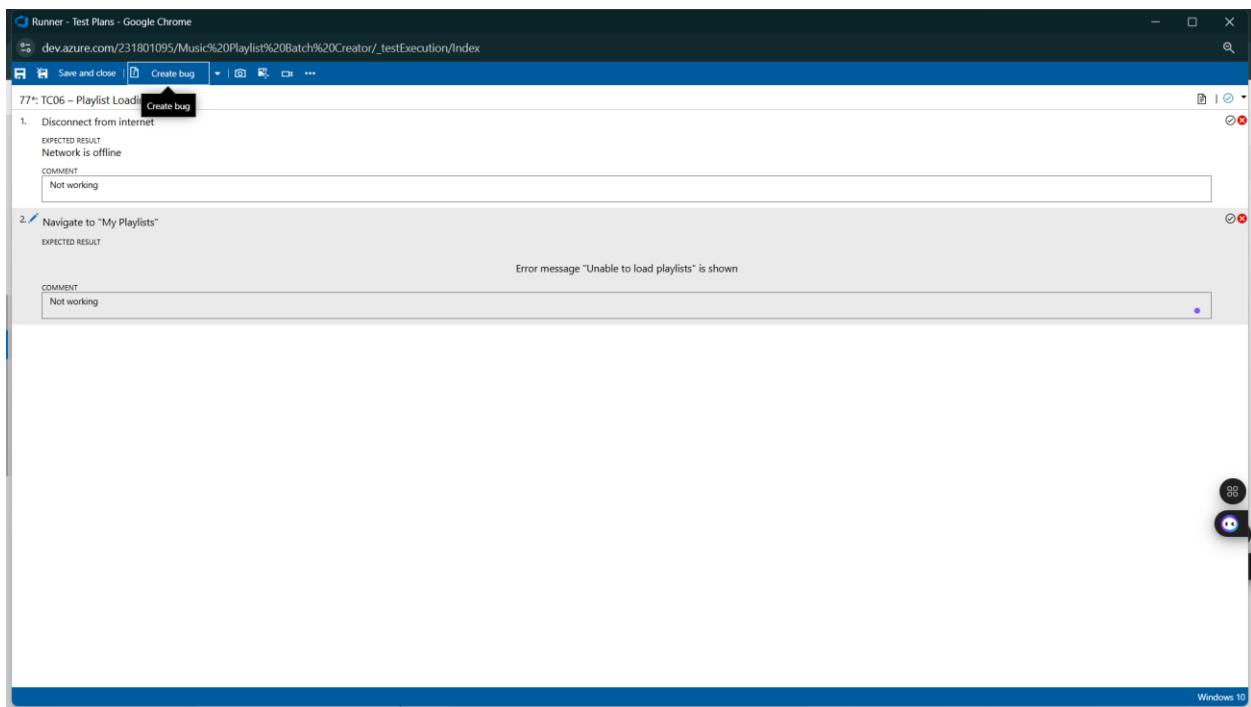
This screenshot shows a browser window titled 'Runner - Test Plans - Google Chrome' with the URL 'dev.azure.com/231801094/Software%20Construction-%20ET/_testExecution/Index'. The page displays the results for '31: TC13 – Receive Budget Alert'. It shows a step: '1. Set a budget of ₹5000 -- Add ₹6000 expense'. Below it, the 'EXPECTED RESULT' section says 'Budget exceeded alert is shown'. The browser's address bar also shows the same URL.



6.Recording the test case



7.Creating the bug



Runner - Test Plans - Google Chrome
dev.azure.com/231801094/Software%20Construction-%20ET/_testExecution/Index

31: TC13 – Receive Budget Alert

Budget exceeded

Bug filed NEW BUG *

Budget exceeded

Unassigned 0 comments Add tag

State New **Area** Software Construction- ET
Reason New **Iteration** Software Construction- ET

Repro Steps

4/27/2025 8:24 AM Bug filed on "TC13 – Receive Budget Alert"

Step no. **Result** **Title**
1. Failed Set a budget of ₹5000 → Add ₹6000 expense
Expected Result
Budget exceeded alert is shown

Test Configuration: Windows 10

System Info

Browser - Name	Google Chrome 131
Browser - Language	en-US
Browser - Height	705
Browser - Width	1315
Browser - User agent	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.0.0 Safari/537.36
Operating system - Name	Windows NT 10.0; Win64; x64

Planning

Resolved Reason
Story Points

Priority 2 **Severity** 3 - Medium **Activity**

Deployment

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

Development

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

Related Work

+ Add link
Add an existing work item as a parent

Tested By
1 TC13 – Receive Budget Alert Updated 4/21/2025, 8 Design

System Info

Windows 10 1:55 PM 4/27/2025

Runner - Test Plans - Google Chrome
dev.azure.com/231801094/Software%20Construction-%20ET/_testExecution/Index

23: TC08 – Edit Expense with Invalid Data

Cat NEW BUG *

Error mes

Unassigned 0 comments Add tag

State New **Area** Software Construction- ET
Reason New **Iteration** Software Construction- ET

System Info

Browser - Name	Google Chrome 131
Browser - Language	en-US
Browser - Height	705
Browser - Width	1315
Browser - User agent	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.0.0 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) i5-1135G7 @ 2.40GHz
Operating system - Number of processors	8
Memory - Available	422739968

Original estimate
Remaining
Completed

Deployment

+ Add link
Add an existing work item as a parent

Tested By
2 TC08 – Edit Expense with Invalid Data Updated 48 minutes ago, 8 Design

System Info

Found in Build
Integrated in Build

Windows 10

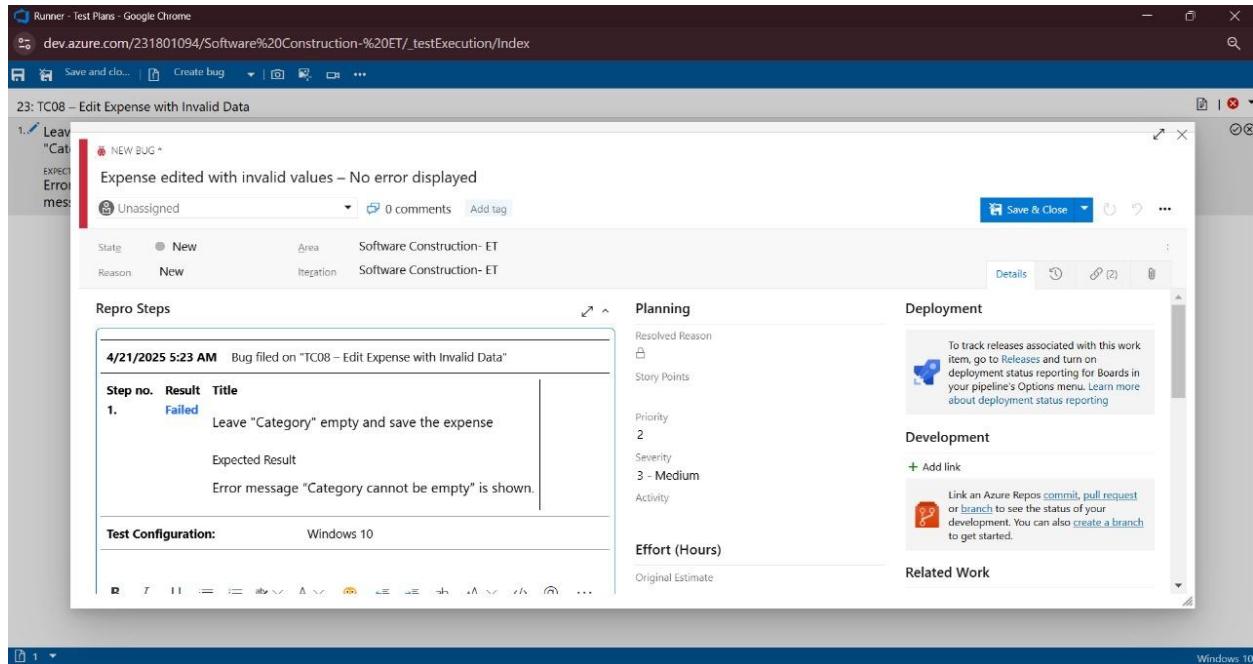
8. Test case results

The screenshot shows the Microsoft Test Plan interface. On the left, there's a navigation pane with icons for Home, Test Suites, Work Items, Pipelines, and Artifacts. The main area shows a 'Test Suites' list under 'Expense Tracker'. A specific test suite, 'Test Suite 1: User Login & Sign-Up (ID: 13)', is selected. Inside this suite, a 'Test Points (4 items)' section is displayed. One item, 'Successful Sign Up', is highlighted and has a blue border. Below it are three other items: 'Successful Login', 'Sign Up with Existing Email', and 'Login with Wrong Password'. To the right of the test points, a 'Test Case Results' table is shown with one row: 'Passed' (Outcome), '2m ago' (TimeStamp), 'Windows 10' (Configuration), 'Mahisha Parameshw...' (Run by), and 'Mahisha Parameshw...' (Tester). At the bottom of the results table, there's a link: 'Open execution history for current test point'.

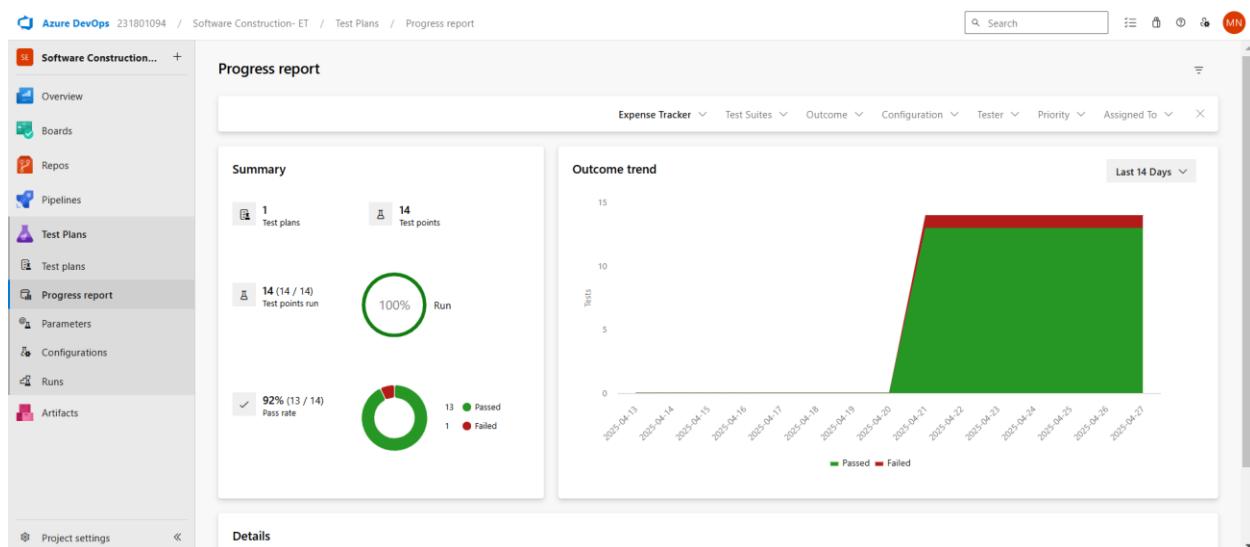
9. Test report summary

The screenshot shows the Azure DevOps Work Items interface. The left sidebar includes links for Overview, Boards, Work items, Backlogs, Sprints, Queries, Delivery Plans, Analytics views, Repos, Pipelines, Test Plans, Artifacts, and Project settings. The main area displays a work item titled 'BUG 203 203 BG 01 - Countries Drop down Not Available on the page'. The work item details show the state is 'New', reason is 'New', repro step is 'Active', and iteration is 'News Feed App'. The steps section lists three steps: 1. Passed (Title: Open Chrome Browser, Result: Passed, Description: Expected Result, Browswer should open in a new tab), 2. Passed (Title: Paste the URL (https://127.0.0.1/new.html), Result: Passed, Description: Expected Result, App should load with new articles), and 3. Failed (Title: System should show a dropdown with list of countries, Result: Failed, Description: Produce Report). To the right of the steps, sections for Planning (Resolved Reason, Story Points, Priority, Severity, Activity), Deployment (Release status), 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), and Related Work are visible. A note at the bottom right says: '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.'

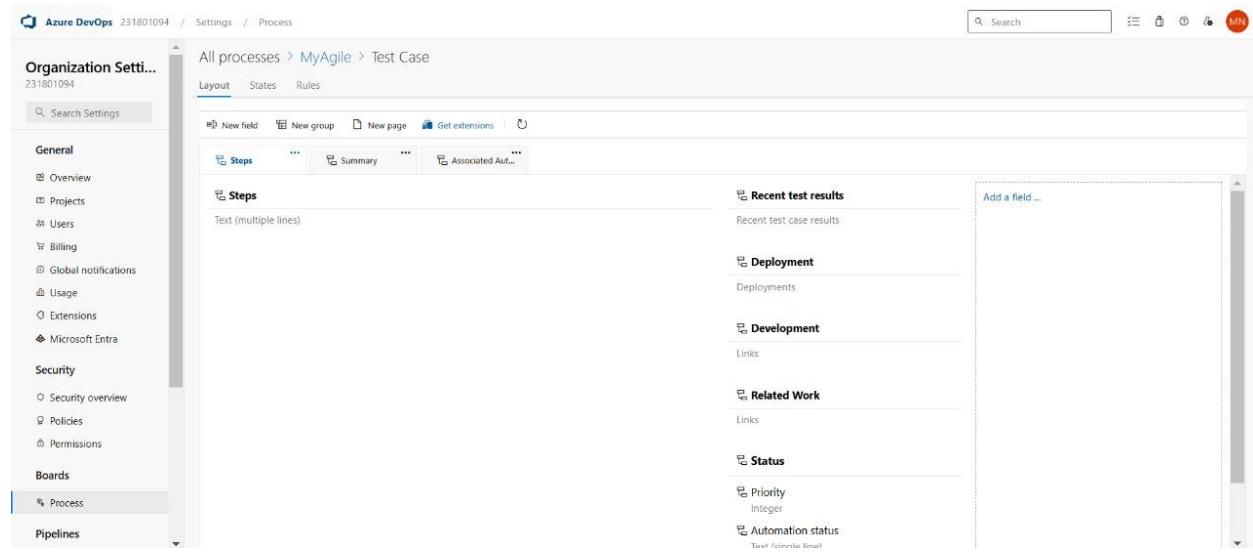
- Assigning bug to the developer and changing state



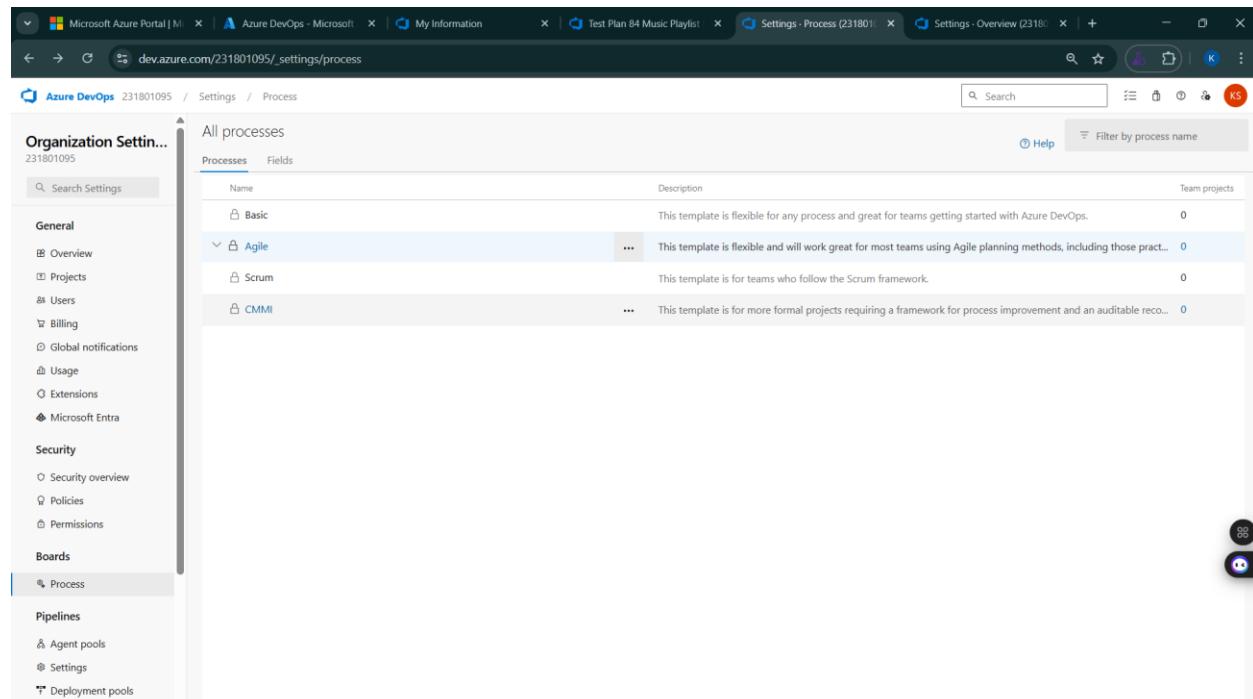
10. Progress report



11. Changing the test template



The screenshot shows the Azure DevOps settings interface for a process named 'MyAgile'. The 'Test Case' template is selected. The 'Layout' tab is active, displaying fields for 'Steps', 'Recent test results', 'Deployment', 'Development', 'Related Work', and 'Status'. A 'Recent test results' section shows 'Recent test case results'. A 'Deployment' section shows 'Deployments'. A 'Development' section shows 'Links'. A 'Related Work' section shows 'Links'. A 'Status' section shows 'Priority' (Integer) and 'Automation status' (Text (multiple lines)). A sidebar on the left lists 'General', 'Security', 'Boards', and 'Process' under 'Organization Settings'. The 'Process' section is expanded, showing 'Pipelines' and 'Agent pools'.



The screenshot shows the 'All processes' screen in Azure DevOps. It lists three process templates: 'Basic', 'Agile', and 'CMMI'. The 'Agile' template is selected. The 'Basic' template is described as flexible for any process. The 'Agile' template is described as flexible for teams using Agile planning methods. The 'CMMI' template is described as formal projects requiring a framework for process improvement. The sidebar on the left is identical to the previous screenshot, showing 'General', 'Security', 'Boards', and 'Process' under 'Organization Settings'.

The screenshot shows the 'All processes' section of the Azure DevOps Settings - Process page. The 'Agile' template is selected. The table lists the following process templates:

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

The screenshot shows the 'Add a field to Test Case' dialog box. The 'Create a field' option is selected, and a new field named 'Acceptance Criteria' is being created with a type of 'Text (single line)'. The dialog also includes a description field and a 'Learn more' link.

Microsoft Azure Portal | My Information | Azure DevOps - Microsoft | My Information | Test Plan 84 Music Playlist | Settings - Process (231801095) | Settings - Overview (231801095) | + | - | ○ | ×

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

Azure DevOps 231801095 / Settings / Process

All processes > 231801095 Agile

Work-item types Backlog levels Projects

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...

Organization Settings... 231801095

General

- Overview
- Projects
- Users
- Billing
- Global notifications
- Usage
- Extensions
- Microsoft Entra

Security

- Security overview
- Policies
- Permissions

Boards

- Process

Pipelines

- Agent pools
- Settings
- Deployment pools

Search Settings

Microsoft Azure Portal | My Information | Azure DevOps - Microsoft | My Information | Test Plan 84 Music Playlist | Settings - Process (231801095) | Settings - Overview (231801095) | + | - | ○ | ×

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

Azure DevOps 231801095 / Settings / Process

All processes > 231801095 Agile > Test Case

Layout States Rules

New field New group New page Get extensions

Steps Summary Associated Aut...

Custom

Type Text (single line)

Recent test results

Recent test case results

Deployment

Development

Related Work

Status

Add a field ...

Organization Settings... 231801095

General

- Overview
- Projects
- Users
- Billing
- Global notifications
- Usage
- Extensions
- Microsoft Entra

Security

- Security overview
- Policies
- Permissions

Boards

- Process

Pipelines

- Agent pools
- Settings
- Deployment pools

Search Settings

2116231801071

CS23432

Result:

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

EXP NO: 9

LOAD TESTING AND PIPELINES

Aim:

To create an Azure Load Testing resource and run a load test to evaluate the performance of a target endpoint and to create and demonstrate an Azure DevOps pipeline for automating application builds, tests, and deployment.

Load Testing

Azure Load Testing:

Azure Load Testing allows you to simulate high traffic and stress tests for your web applications and APIs to understand how they perform under load. It helps identify performance bottlenecks, scalability issues, and optimize resource usage before deployment.

Steps to Create an Azure Load Testing Resource:

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

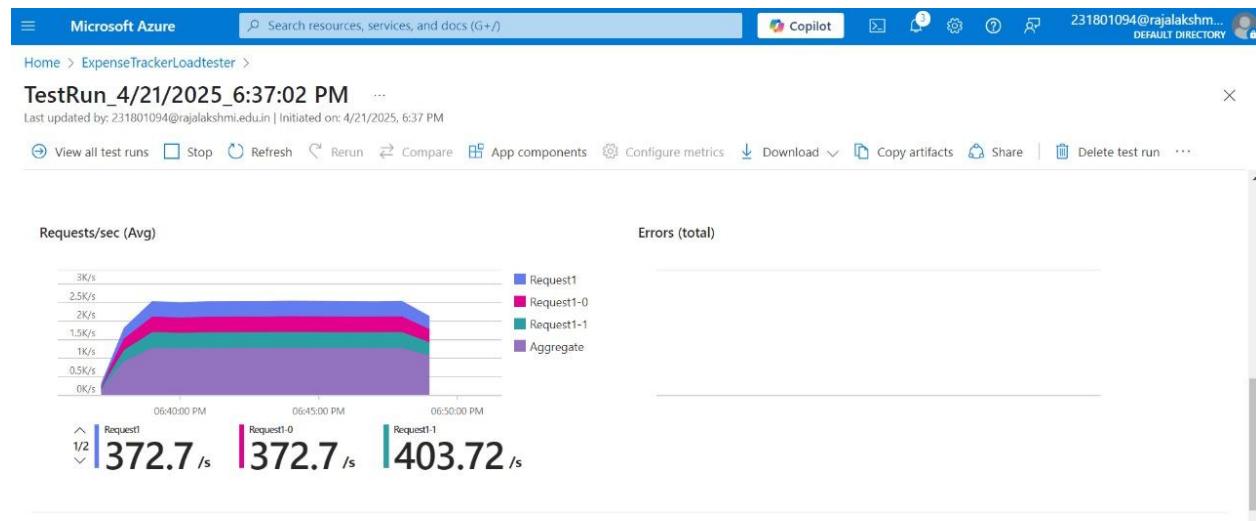
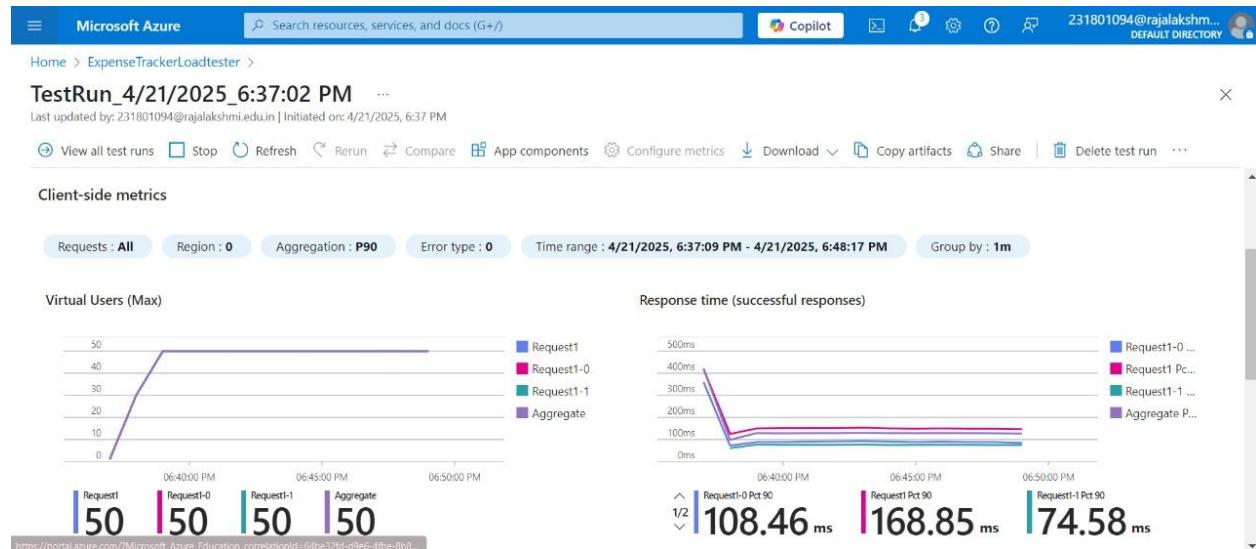
1. Sign in to Azure Portal
Go to <https://portal.azure.com> and log in.
2. Create the Resource
 - o Go to *Create a resource* → Search for “Azure Load Testing”.
 - o Select Azure Load Testing and click Create.
3. Fill in the Configuration Details
 - o *Subscription*: Choose your Azure subscription.
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

Steps to Create and Run a Load Test:

Once your resource is ready:

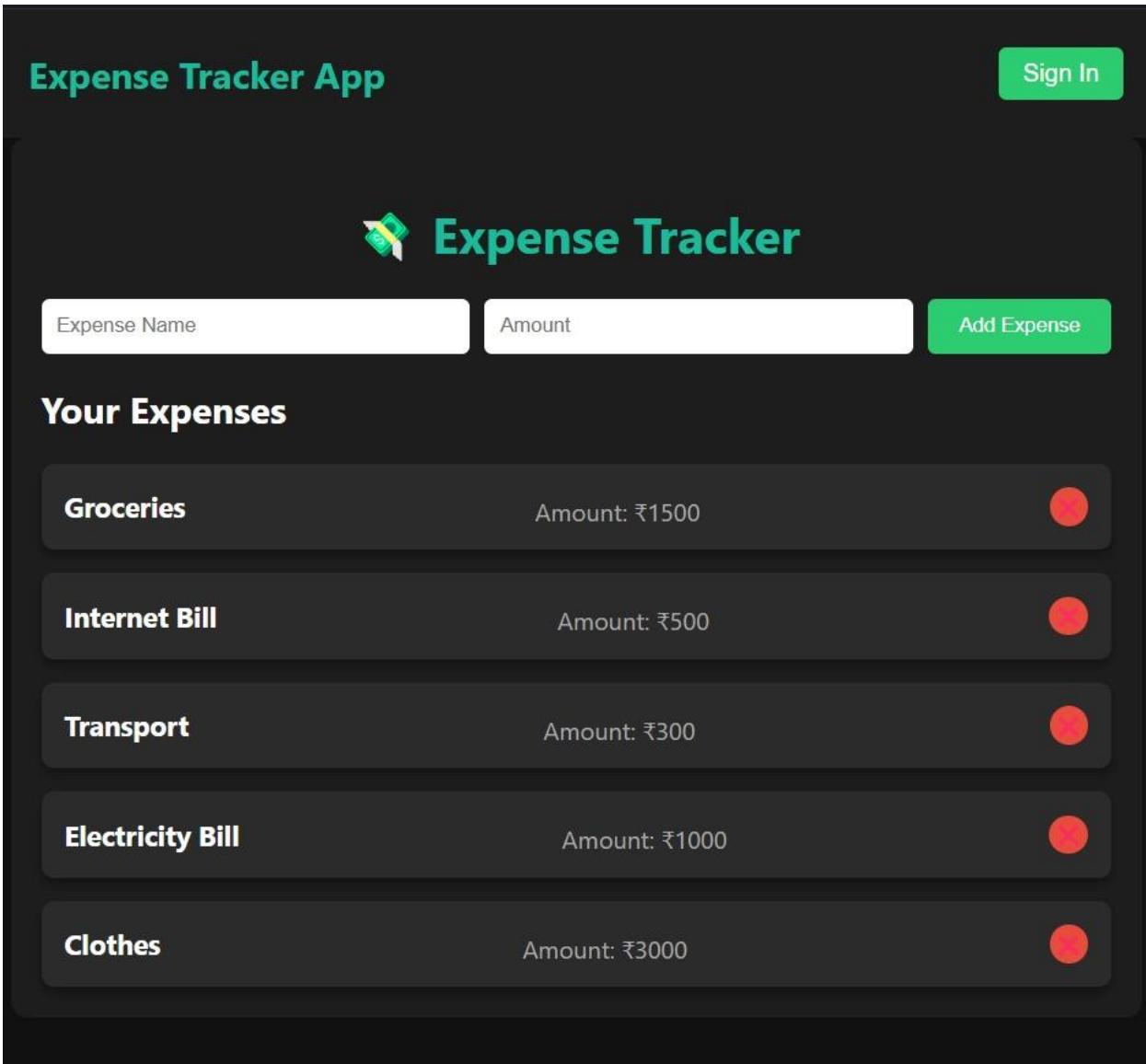
1. Go to your Azure Load Testing resource and click Add HTTP requests > Create.
2. Basics Tab
 - o *Test Name*: Provide a unique name.
 - o *Description*: (Optional) Add test purpose.
 - o *Run After Creation*: Keep checked.
3. Load Settings
 - o *Test URL*: Enter the target endpoint (e.g., <https://yourapi.com/products>).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

Load Testing



2116231801071

CS23432



Pipelines

Description:

This experiment demonstrates how to connect a GitHub-hosted Flask-based music recommendation project with Azure DevOps. The pipeline will automatically install dependencies, run basic tests, and publish artifacts. This ensures that every commit triggers checks for reliability and smooth deployment.

Steps:

1. Connect GitHub to Azure DevOps:
 - o In Azure DevOps, create a new project.
 - o Create a pipeline and select GitHub as the source.

- Authorize access to your GitHub repository, ensuring that Azure DevOps can pull the repository for your pipeline.
2. Create azure-pipelines.yml in Your Repo Root:
 - In your GitHub repository, create a new file called azure-pipelines.yml in the root directory.
 - Add the following basic pipeline configuration for Python and Flask:

yml Code

trigger:

```
- main # Trigger pipeline when changes are pushed to the main branch
```

pool:

```
vmImage: ubuntu-latest # Use a hosted Ubuntu agent
```

steps:

Step 1: Checkout the code from GitHub

```
- checkout: self
```

Step 2: Set up Python environment

```
- task: UsePythonVersion@0
```

inputs:

```
versionSpec: '3.x' # Use the latest Python 3.x version
```

```
displayName: "Set up Python"
```

Step 3: Install dependencies from the correct path

```
- script: |
```

```
  python -m pip install --upgrade pip
```

```
  pip install -r project/requirements.txt # Adjusted path to requirements.txt
```

```
displayName: "Install dependencies"
```

Step 4: Run a simple Python script to check the environment

```
- script: |
```

```
  python -c "print('⚡ Hello from Music Playlist Batch Creator!')"
```

```
displayName: "Run a Python script"
```

3. Pipeline Tasks Include:

- Setting up the Python environment using the UsePythonVersion task.

- Installing project dependencies from project/requirements.txt. Make sure the path to requirements.txt is correct (it is located under the project folder).
 - Running a simple Python script to verify that Python is set up correctly and the pipeline works.
4. Run and Monitor Pipeline:
- Commit changes to the main branch of your repository to trigger the pipeline in Azure DevOps.
 - Monitor the logs in the Azure DevOps portal to view logs, errors, or success messages and ensure everything runs smoothly.

Pipeline

The screenshot shows the Azure DevOps Pipelines interface for a project named "Music Playlist Batch Creator". A specific pipeline run, #20250424.3, is displayed. The summary card indicates the run was manually triggered by Karthick S and completed just now, taking 24 seconds. It shows 0 work items and 0 artifacts. The "Jobs" section lists a single job that succeeded in 6 seconds. The left sidebar shows other pipeline-related options like Overview, Boards, Repos, and Test Plans.

Result:

Successfully created the Azure Load Testing resource and executed a load test to assess the performance of the specified endpoint and also demonstrated pipelines in azure devops.

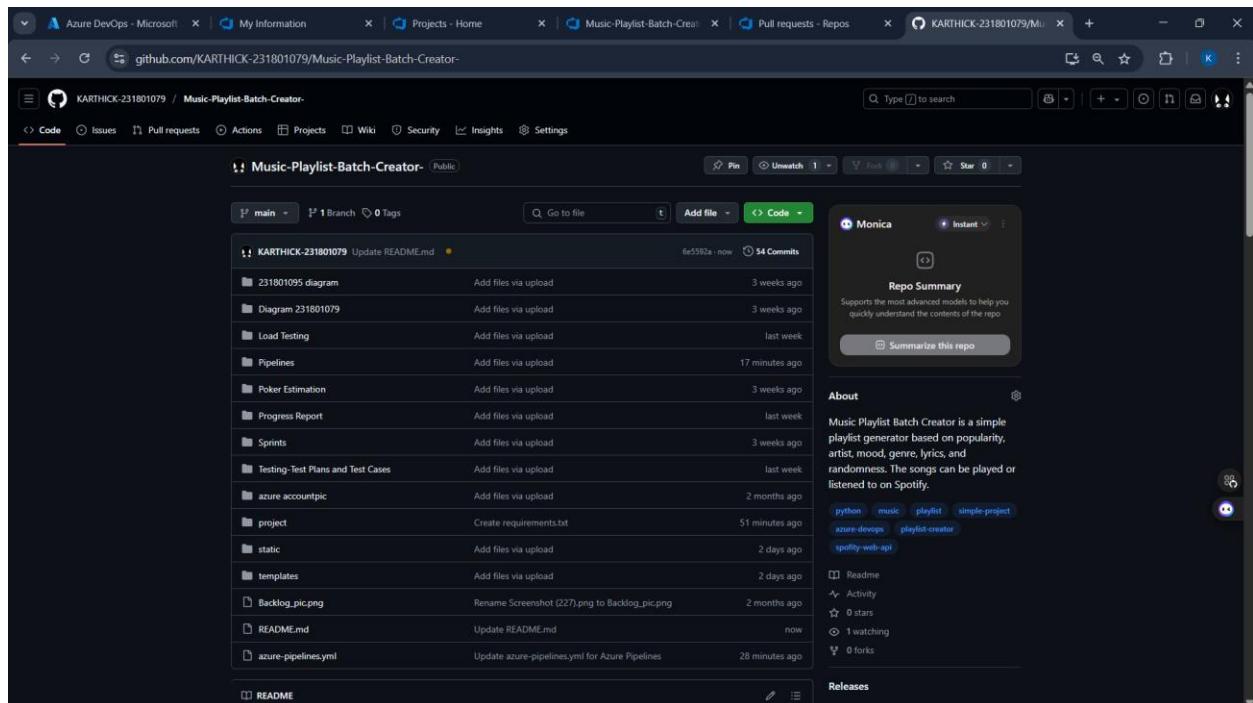
EXP NO: 10

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 Expense Tracker Appproject.

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.