



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

BONAFIDE CERTIFICATE

NAME.....Mahisha Parameshwari N.M......

ACADEMIC YEAR.....2024 - 2025..... SEMESTER

IV

BRANCH.....AI & DS.....

UNIVERSITY REGISTER No.

2116231801094

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: Mahisha Parameshwari Branch: AI&DS Sec: FA Roll No: 231801094

S.No.	Date	Title	Page No.	Teacher's Sign/Remarks
1.	21/1/25	Azure DevOps Environment Setup.	3	
2.	21/1/25	Azure DevOps Project Setup and User Story Management.	6	
3.	28/1/25	Setting Up Epics, Features, And User Stories for Project Planning.	11	
4.	11/2/25	Sprint Planning	16	
5.	18/2/25	Poker Estimation	20	
6.	25/2/25	Designing Clays and Sequence diagrams for Project Architecture.	25	
7.	4/3/25	Architectural and ER diagrams	37	
8.	25/3/25	Testing - Test Plans and Test cases	42	
9.	15/4/25	Load Testing and Pipelines	49	
10.	22/4/25	Github: Project Structure and Naming conventions.	53	
		Completed		



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

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING LAB MANUAL

CS23432 – Software Construction

(REGULATION 2023)

RAJALAKSHMI ENGINEERING COLLEGE
Thandalam, Chennai-602015

Name: Mahisha Parameshwari NM

Register No: 231801094

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

Semester: IV

Academic Year: 2024 - 2025

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:

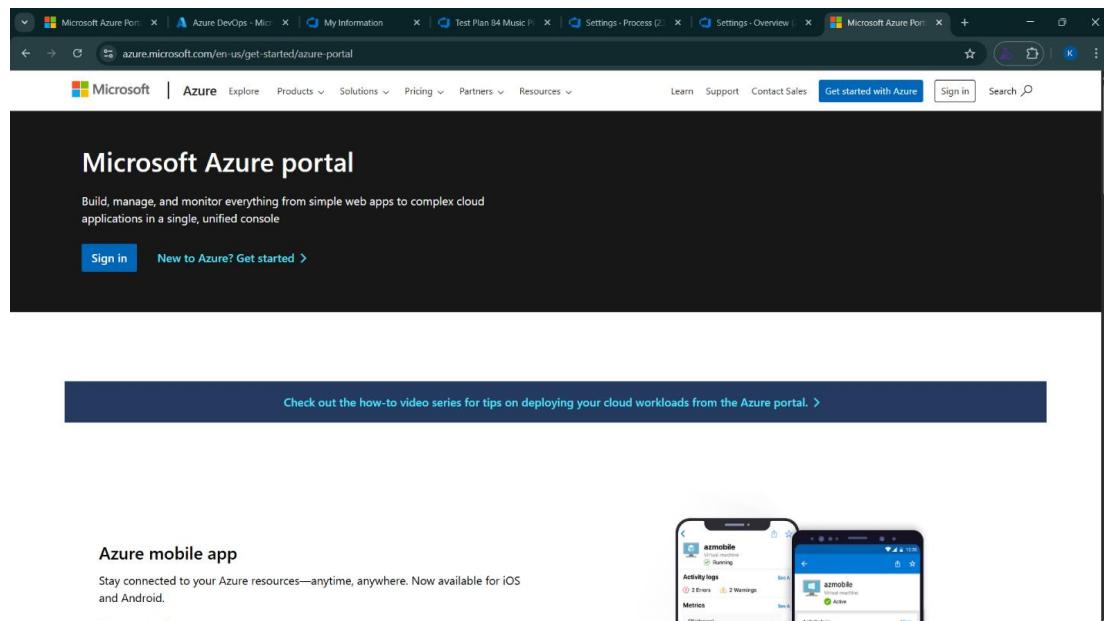
To set up and access the Azure DevOps environment by creating an organization through the Azure portal.

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>



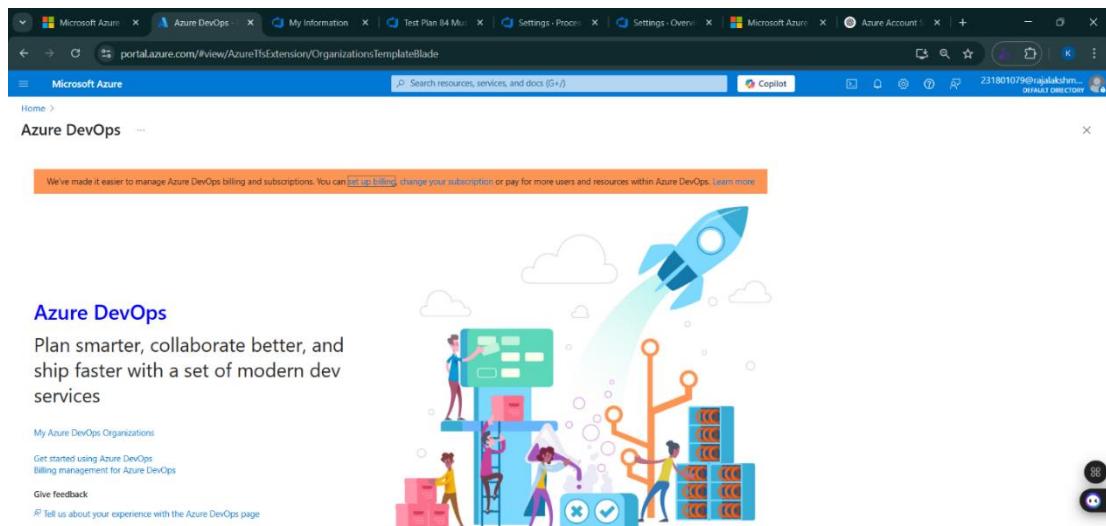
2. Azure home page

The screenshot shows the Microsoft Azure home page. At the top, there's a search bar and a Copilot button. Below the search bar, the 'Azure services' section features 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. The 'Resources' section displays recent resources: 'Music' (Azure Load Testing) and 'Music_playlist_Batch_Creator' (Resource group), both last viewed 3 days ago. The 'Navigate' section includes links for Subscriptions, Resource groups, All resources, and Dashboard. The 'Tools' section lists Microsoft Learn, Azure Monitor, Microsoft Defender for Cloud, and Cost Management. The 'Useful links' section includes a link to the Azure mobile app.

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

The screenshot shows the Microsoft Azure home page with a search bar containing 'DevOps'. The search results for 'DevOps' are displayed under the 'Services' section, listing 'Azure Native New Relic Service', 'Managed DevOps Pools', 'Azure DevOps organizations', and 'Azure Native Dynatrace Service'. Below the search bar, the 'Resources' section shows the same recent resources as the first screenshot. The 'Tools' section includes Microsoft Learn, Azure DevOps, and Cost Management. The 'Useful links' section includes a link to the Azure mobile app.

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.

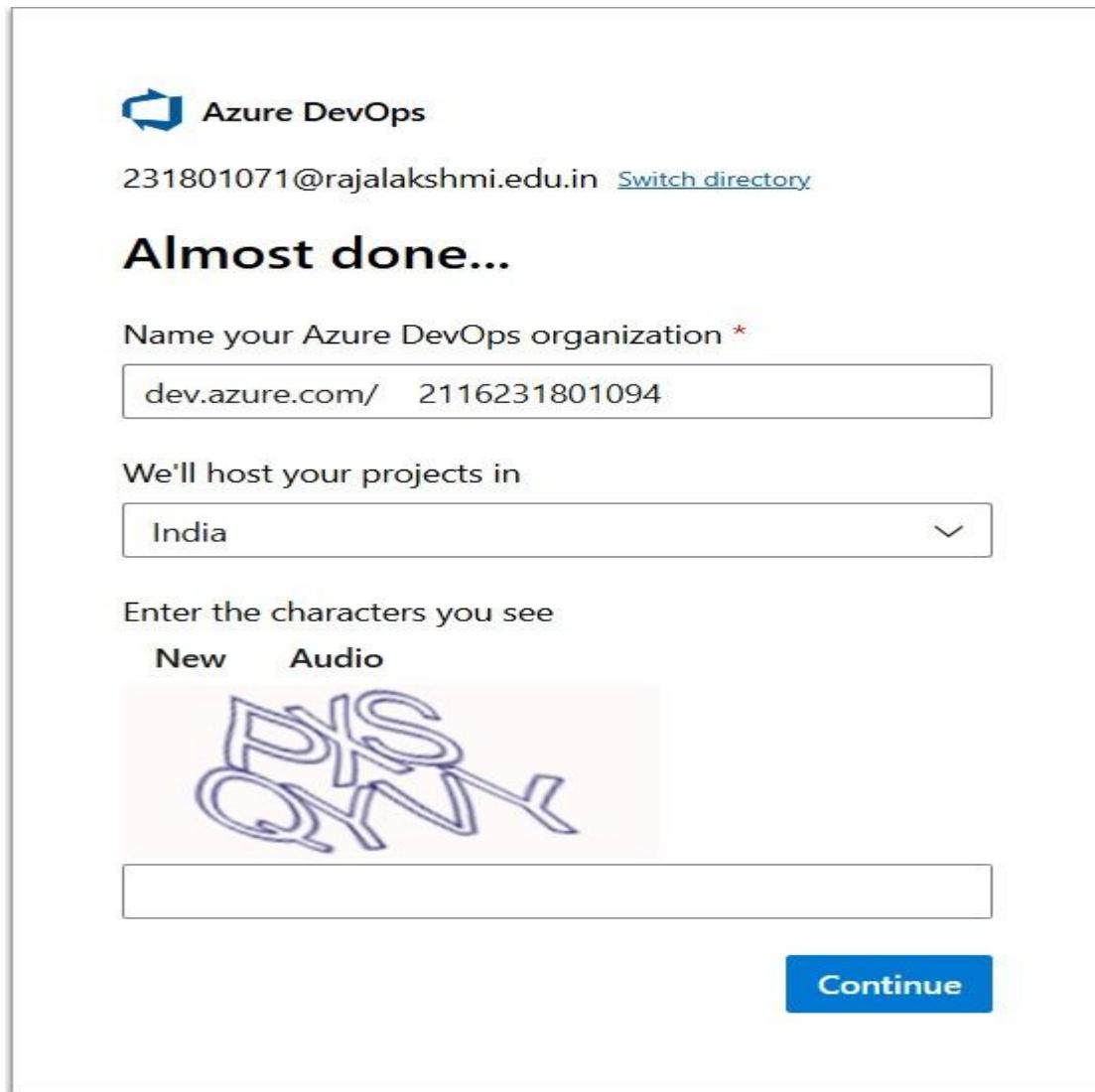
EXP NO: 2

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 second step of the Azure DevOps organization creation process. At the top, it says "Almost done...". It asks for the name of the organization, which is "dev.azure.com/ 2116231801094". Below that, it asks where to host projects, with "India" selected. There's a CAPTCHA challenge with the text "New Audio" and a distorted purple text "ASQW". A blue "Continue" button is at the bottom right.

2.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.
- 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.

4. Project dashboard

About this project

The Azure Expense Tracker App is a cloud-based solution designed for efficient financial management and personal expense tracking. Leveraging Azure services, this tool allows users to monitor, categorize, and analyze their expenses, helping them make smarter financial decisions.

Key Features

- Expense Logging: Add and categorize expenses quickly with details like amount, category, date, and notes.
- Budget Management: Set monthly budgets for different categories and track spending against limits.
- Data Visualization: View expenses with intuitive graphs and charts (pie charts, bar graphs).
- Recurring Expenses: Automate monthly bills like rent, subscriptions, and utilities.
- Notifications: Get alerts for budget overruns, upcoming bill payments, and savings tips.
- Multi-Device Access: Access your financial data securely from web and mobile platforms.
- Expense Sharing: Share expense reports with family members or roommates easily.

Use Cases

- Students: Track monthly pocket money and educational expenses.
- Working Professionals: Manage salaries, investments, and daily expenses smartly.
- Families: Plan and monitor household budgets and shared expenses.
- Small Business Owners: Keep business expenses organized and generate simple reports.

Technical Stack

- Frontend: Web-based UI (HTML, CSS, JavaScript, React.js)

Project stats

Period: Last 7 days

Boards	Repos	Pipelines	Members
23 Work items created	0 Work items completed	0 Pull requests opened	3 Commits by 2 authors
Builds succeeded			

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' backlog. The left sidebar includes icons for EA, +, Boards, Analytics, Epics, Stories, Tasks, and Bugs. The main area displays a table of backlog items:

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	

At the top right, there are buttons for '+ New Work Item', 'View as Board', and 'Epics'. The bottom of the screen shows a Windows taskbar with various pinned icons and system status.

The screenshot shows the Azure DevOps Project Summary page for 'Software Construction- ET'. The left sidebar includes icons for Overview, Summary, Dashboards, Wiki, Boards, Repos, Pipelines, Test Plans, and Artifacts. The main area displays project information:

- About this project:** The Azure Expense Tracker App is a cloud-based solution designed for efficient financial management and personal expense tracking. Leveraging Azure services, this tool allows users to monitor, categorize, and analyze their expenses, helping them make smarter financial decisions.
- Key Features:**
 - Expense Logging: Add and categorize expenses quickly with details like amount, category, date, and notes.
 - Budget Management: Set monthly budgets for different categories and track spending against limits.
 - Data Visualization: View expenses with intuitive graphs and charts (pie charts, bar graphs).
 - Recurring Expenses: Automate monthly bills like rent, subscriptions, and utilities.
 - Notifications: Get alerts for budget overruns, upcoming bill payments, and savings tips.
 - Multi-Device Access: Access your financial data securely from web and mobile platforms.
 - Expense Sharing: Share expense reports with family members or roommates easily.
- Use Cases:**
 - Students: Track monthly pocket money and educational expenses.
 - Working Professionals: Manage salaries, investments, and daily expenses smartly.
 - Families: Plan and monitor household budgets and shared expenses.
 - Small Business Owners: Keep business expenses organized and generate simple reports.
- Technical Stack:**
 - Frontend: Web-based UI (HTML, CSS, JavaScript, React.js)
 - Backend: Azure Functions, Node.js

The right side of the screen shows 'Project stats' with metrics for work items created (23), work items completed (0), pull requests opened (0), commits by 2 authors (3), and builds succeeded (0%). It also shows a 'Members' section with 5 users.

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 Backlog board for the "Expense Tracker App Team". The board displays a hierarchy of work items, starting with Epics and branching down to Features and Tasks. The backlog is organized by Order, Work Item Type, Title, State, Effort, Business Area, and Tags. The "Business" value area is consistently applied across all items.

Order	Work Item Type	Title	State	Effort	Busin...	Value 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	

1.Fill in Epics

The screenshot shows the 'Expense Tracker App Team Epic' page in Azure DevOps. A red error message at the top left states 'NEW EPIC • Field 'Title' cannot be empty.' The main area is titled 'Enter title' with a placeholder 'Click to add Description.' Below it is a 'Discussion' section with a comment input field. On the left sidebar, 'Epics' is selected. The top navigation bar shows the URL 'dev.azure.com/231801071/Expense%20Tracker%20App/_backlogs/backlog/Expense%20Tracker%20App%20Team/Epics'. The bottom status bar indicates 'JPY/EUR -0.54%' and the date '27-04-2025'.

2.Fill in Features

The screenshot shows the 'Software Construction- ET' project's feature creation interface. A red error message at the top left states 'FEATURE 4 • Field 'Title' cannot be empty.' The main area is titled 'FEATURE 4' with a placeholder 'Click to add Description.' Below it is a 'Discussion' section with a comment input field. On the left sidebar, 'Features' is selected. The top navigation bar shows the URL 'dev.azure.com/231801094/Software%20Construction-%20ET/_boards/Boards'. The bottom status bar indicates 'Updated by Mahisha Parameshwari NM: Feb 4' and the date '27-04-2025'.

3. Fill in User Story Details

The screenshot shows the Azure DevOps interface for creating a new user story. The URL is dev.azure.com/231801071/Expense%20Tracker%20App/_backlogs/backlog/Expense%20Tracker%20App%20Team/Stories. The page title is 'Expense Tracker App / Boards / Backlogs'. A modal window titled 'NEW USER STORY *' is open, with a validation message 'Field 'Title' cannot be empty.' The 'Enter title' field is empty. Below it, there are sections for 'Description', 'Acceptance Criteria', 'Discussion', 'Planning', 'Deployment', 'Classification', 'Development', and 'Planning Poker'. The 'Planning' section shows 'State: New', 'Area: Expense Tracker App', 'Reason: New', and 'Iteration: Expense Tracker App\Sprint 4'. The 'Deployment' section has a note about tracking releases. The 'Classification' section shows 'Value area: Business'. The 'Development' section has a note about linking to Azure Repos. The 'Planning Poker' section has a note about saving the work item. At the bottom of the modal, there is a 'Save and Close' button and a 'Save and re-open the new work item' button.

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 Music Playlist Batch Creator Project.

Sprint Planning

Sprint 1

The screenshot shows the Azure DevOps Boards interface for the 'Expense Tracker App Team' project. The 'Backlog' tab is selected. A single sprint, 'Sprint 1', is visible, spanning from 'February 25 - March 10' (14 work days). The backlog table lists four user stories:

Order	Title	State	Assigned To	Rema...
1	As a user, I want to authenticate using OAuth (Google, Micr... (1 story)	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 interface includes various navigation and search tools, as well as a weather widget indicating 35°C and sunny conditions.

Sprint 2

The screenshot shows the Azure DevOps Backlog board for the 'Expense Tracker App Team' in the 'Sprint 2' iteration. The backlog is organized into four columns: Order, Title, State, and Assigned To. There are four items listed:

Order	Title	State	Assigned To
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 board also includes navigation tabs for Taskboard, Backlog, Capacity, and Analytics. A summary at the top right indicates the sprint runs from March 11 to March 24, covering 14 work days. The environment is a Windows desktop with a taskbar showing various application icons.

Sprint 3

The screenshot shows the Azure DevOps Backlog board for the 'Expense Tracker App Team' in the 'Sprint 3' iteration. The backlog is organized into four columns: Order, Title, State, and Assigned To. There are four items listed:

Order	Title	State	Assigned To
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 board also includes navigation tabs for Taskboard, Backlog, Capacity, and Analytics. A summary at the top right indicates the sprint runs from March 25 to April 7, covering 14 work days. A Snipping Tool window is overlaid on the bottom right, showing a screenshot of the application interface and providing options to copy it to the clipboard or mark it up.

Sprint 4

The screenshot shows the Azure DevOps Backlog board for the 'Expense Tracker App Team' project. The board is set to the 'Backlog' tab and displays the 'Sprint 4' backlog. The backlog table has columns for Order, Title, State, Assigned To, and Remaining. There are four items listed:

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

A context menu is open over the fourth item, showing options like 'Move to Sprint', 'Edit', 'Delete', and 'Copy'. A tooltip indicates '14 work days' for the sprint duration. The interface includes a sidebar with team avatars and a header with navigation links like Taskboard, Capacity, and Analytics.

Result:

The Sprints are created for the Music Playlist Batch Creator Project.

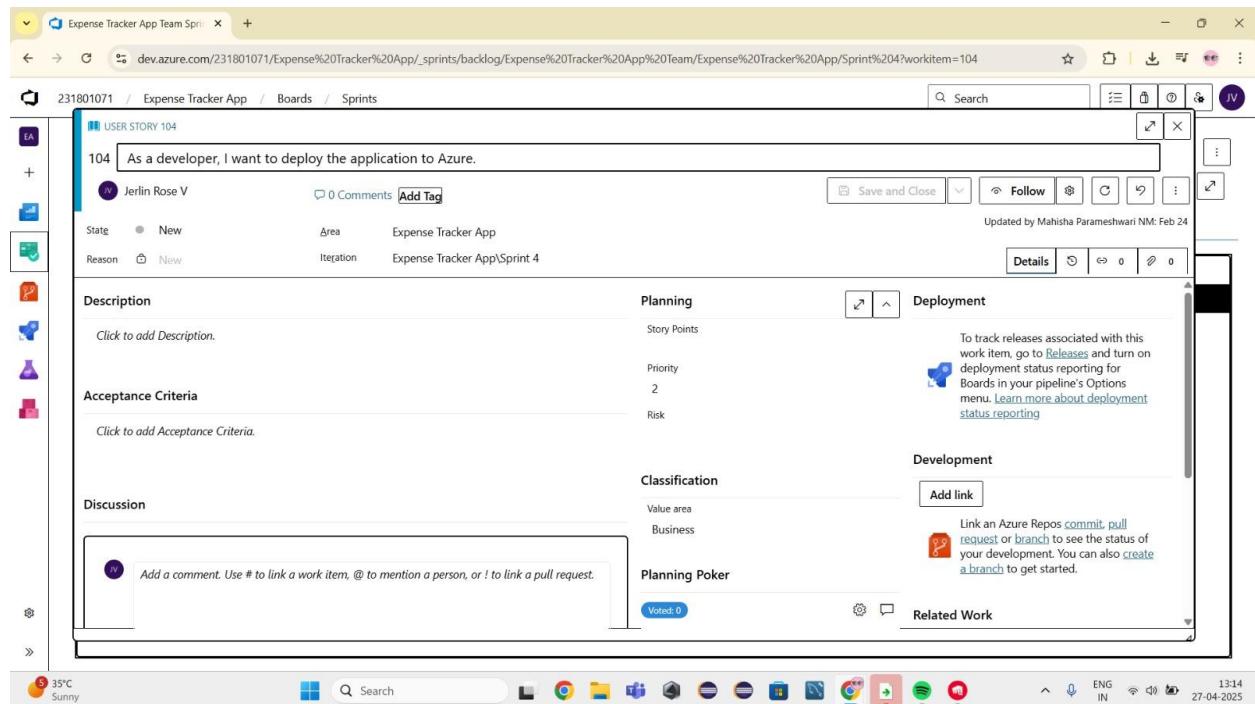
EXP NO: 5

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". A specific user story, "104 As a developer, I want to deploy the application to Azure.", is displayed. The story is assigned to "Jerlin Rose V" and is currently "New". It is categorized under the "Expense Tracker App" area and iteration. The "Planning" section indicates a priority of 2 and 2 story points. The "Deployment" section provides instructions for tracking releases. The "Classification" section identifies the value area as "Business". The "Development" section includes a "Planning Poker" feature with a "Voted: 0" button and a "Related Work" link. The interface also includes a sidebar with various project management icons and a bottom navigation bar with system status indicators.

Result:

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

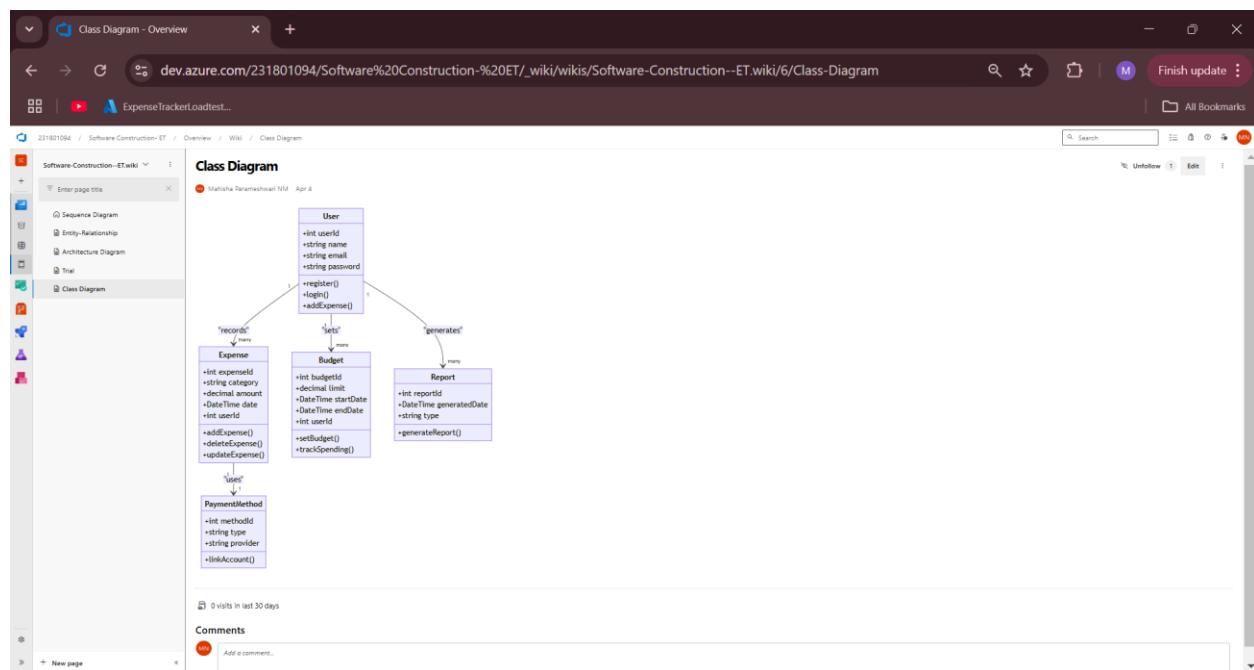
EXP NO: 6

DESIGNING CLASS AND SEQUENCE DIAGRAMS FOR PROJECT ARCHITECTURE

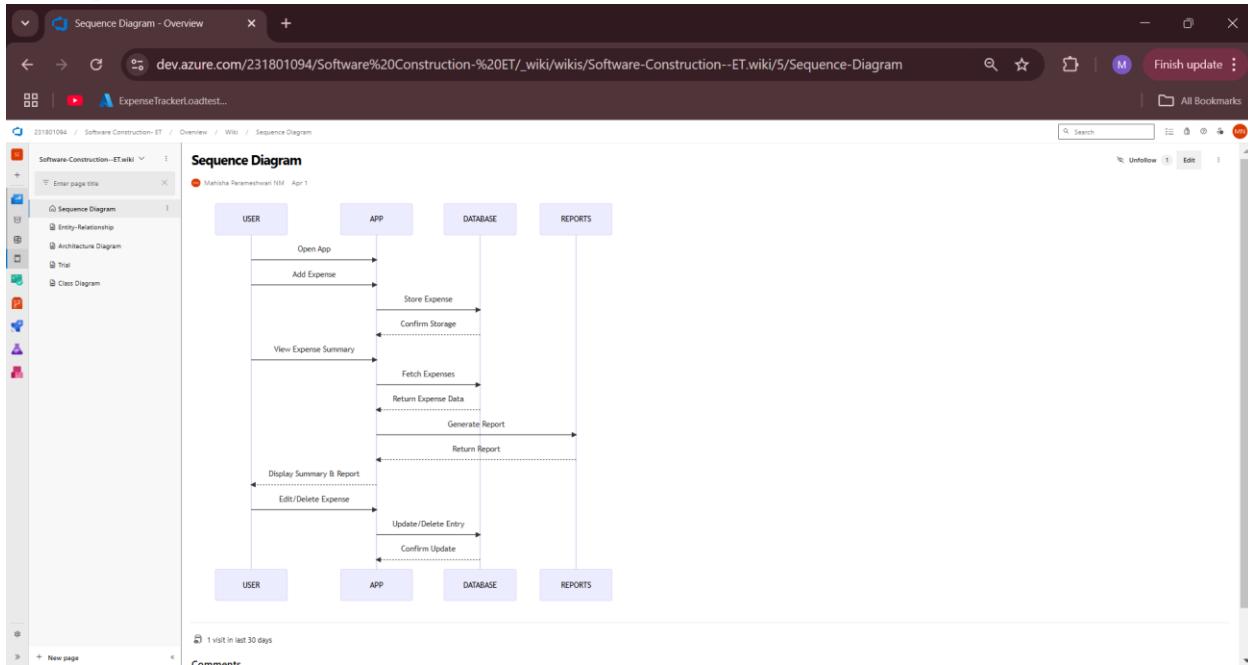
Aim:

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 Expense Tracker

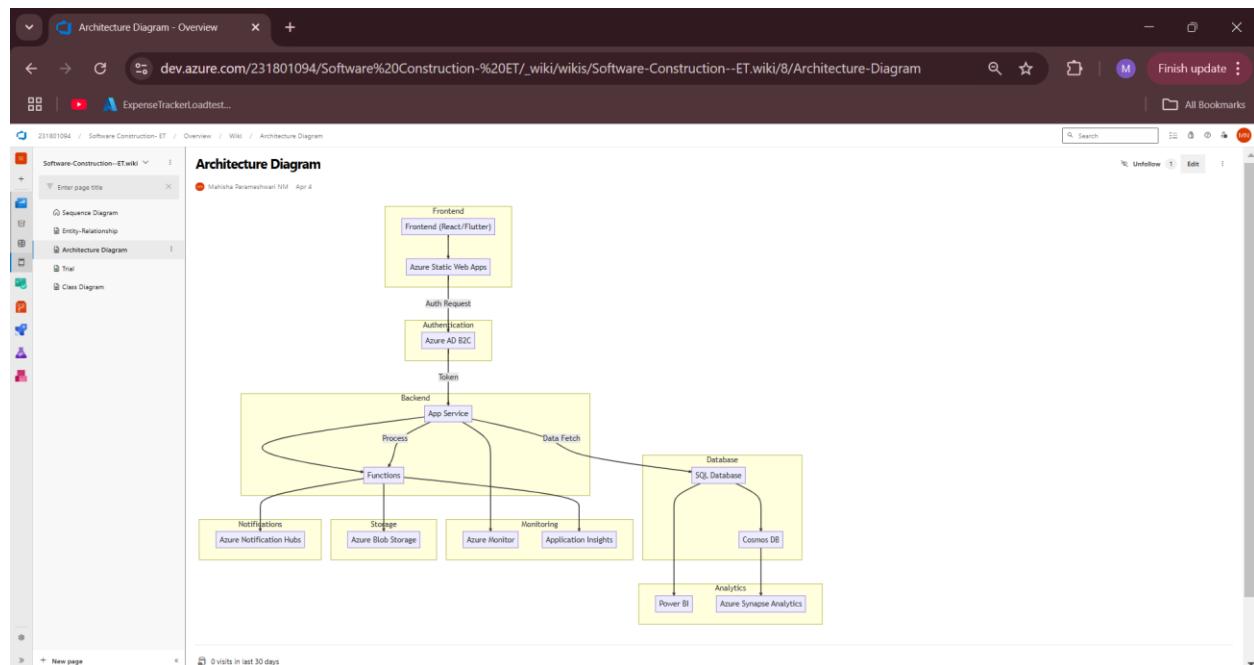
EXP NO: 7

DESIGNING ARCHITECTURAL AND ER DIAGRAMS FOR PROJECT STRUCTURE

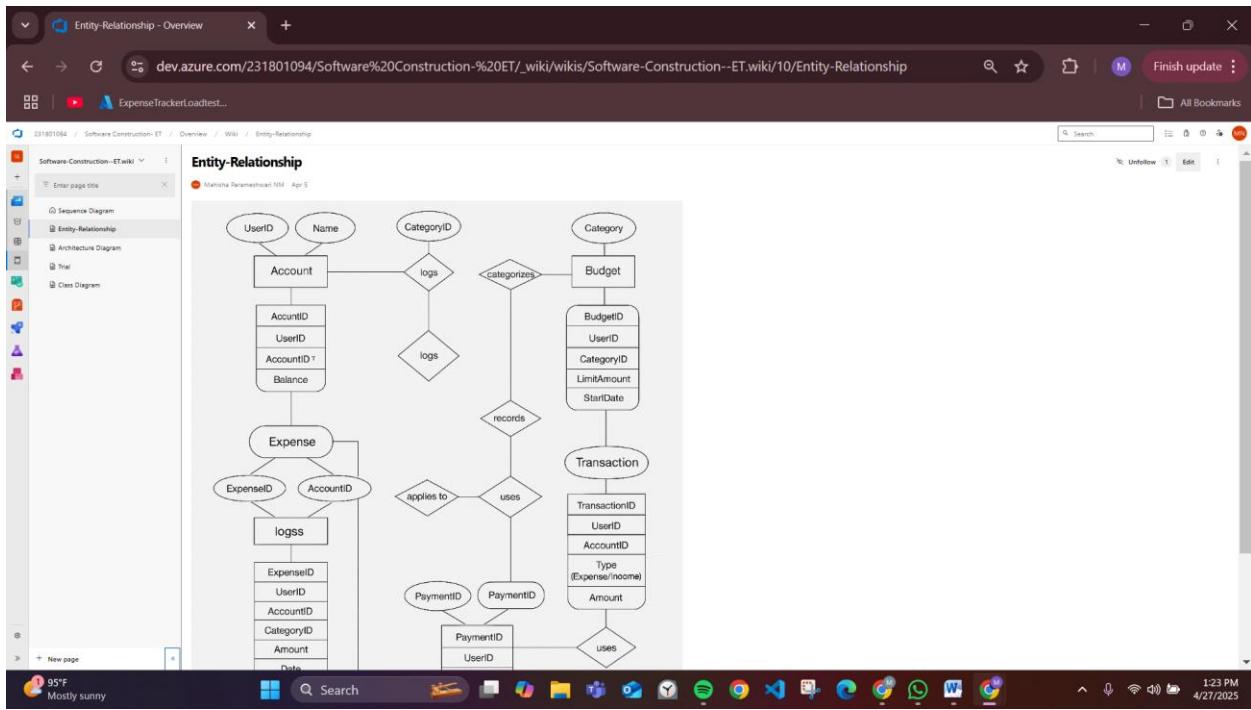
Aim:

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

The screenshot shows the 'New Test Plan' dialog box. On the left is a vertical toolbar with icons for creating new items. The main area has fields for 'Name' (containing 'Log in and sign up'), 'Area Path' (set to 'Software Construction- ET'), and 'Iteration' (also set to 'Software Construction- ET'). At the bottom are 'Create' and 'Cancel' buttons.

2. Test suite

The screenshot shows the 'Expense Tracker' test suite details. The left sidebar lists 'Test Suites' and 'Expense Tracker'. A context menu is open over a 'Test Suite' item, showing options like 'New Suite', 'Assign configurations', 'Export', 'Open', 'Assign testers to run all tests', 'Rename', 'Delete', and 'Import test suites'. The main pane displays 'Test Suite 1: User Login & Sign-Up (ID: 13)' with tabs for 'Define', 'Execute', and 'Chart'. Under 'Test Cases (4 items)', there is a table:

	Order	Test Case Id	Assigned To	State
<input type="checkbox"/> Title	1	14	Mahisha Parameshw...	Design
<input type="checkbox"/> Successful Sign Up	2	15	Mahisha Parameshw...	Design
<input type="checkbox"/> Successful Login	3	16	Mahisha Parameshw...	Design
<input type="checkbox"/> Create New User Using Email	4	17	Mahisha Parameshw...	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.

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

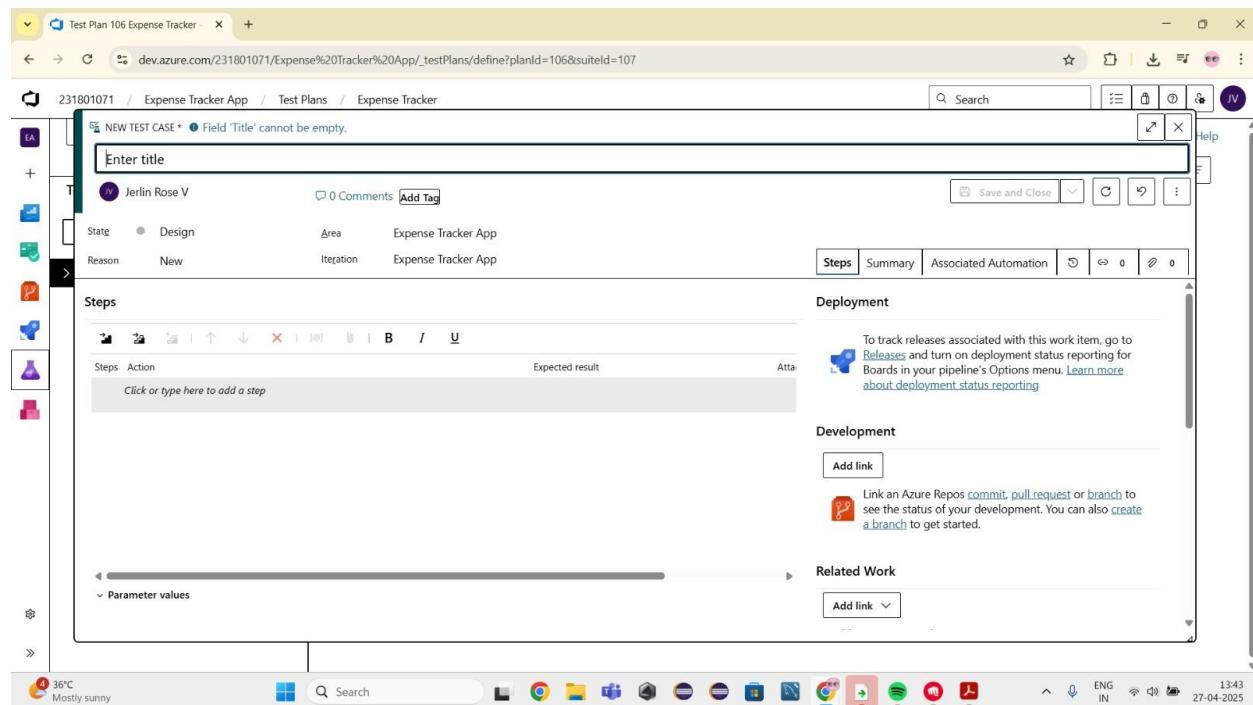
TC13 – Receive Budget Alert

- 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

Create a test case



Test Cases

The image displays two separate screenshots of the Microsoft Test Case management interface, showing the details of two different test cases.

Test Case 31: TC13 – Receive Budget Alert

Test Case 32: TC14 – Notification Failure

Common Test Case Details:

- Owner:** Mahisha Parameshwari NM
- Comments:** 0
- Add Tag:** Available
- State:** Design
- Reason:** New
- Area:** Software Construction- ET
- Iteration:** Software Construction- ET
- Last Updated:** Updated by Mahisha Parameshwari NM: Monday
- Steps Tab:** Active
- Summary Tab:** Available
- Associated Automation Tab:** Available

Test Case 31: TC13 – Receive Budget Alert

Steps:

Steps	Action	Expected result	Attachments
1.	Set a budget of ₹5000 → Add ₹6000 expense	Budget exceeded alert is shown	

Test Case 32: TC14 – Notification Failure

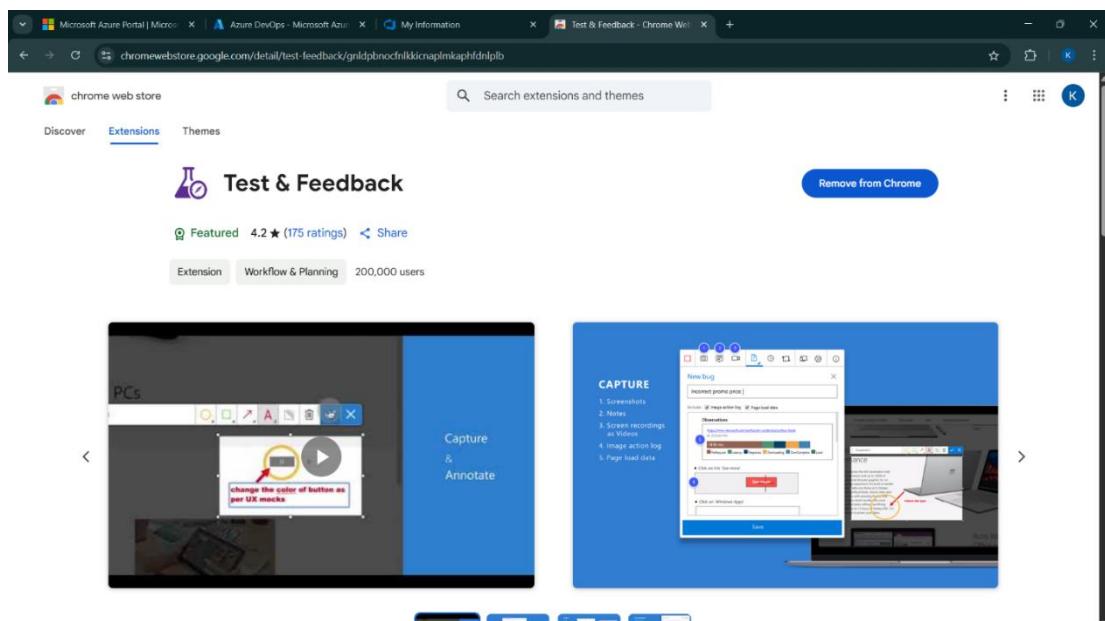
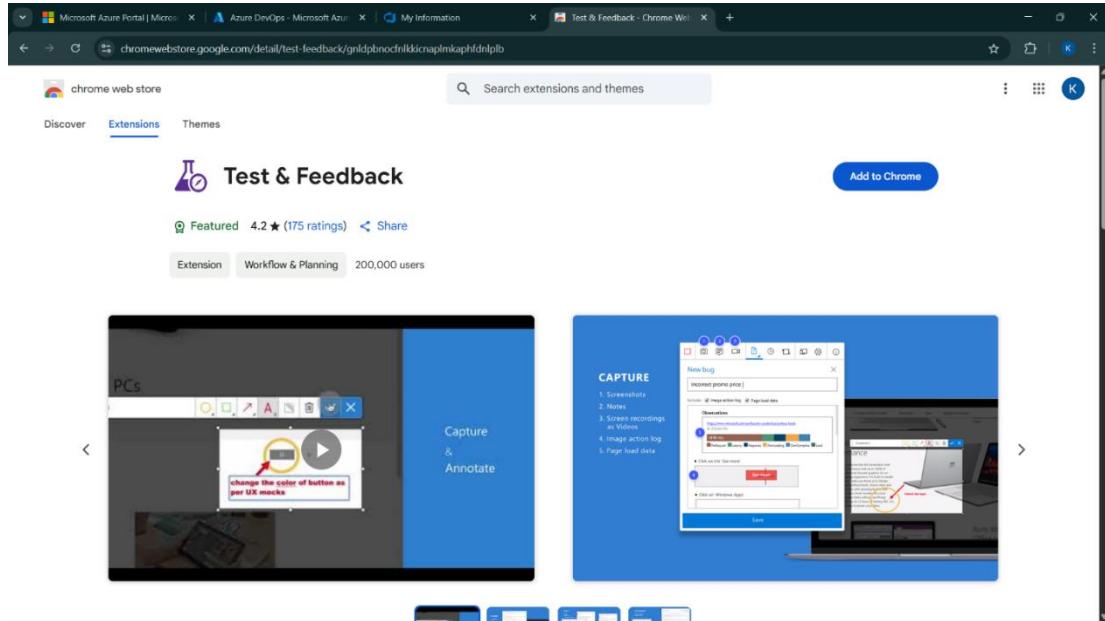
Steps:

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

Common UI Elements (Both Screens):

- Search Bar:** Top right corner.
- Help Button:** Top right corner.
- Deployment Section:** Includes a link to Releases and deployment status reporting.
- Development Section:** Includes a link to Azure Repos commit, pull request or branch.
- Related Work Section:** Includes links to Add link and Add an existing work item as a parent.
- Parameter Values Section:** A dropdown menu at the bottom left of each screen.

4. Installation of test



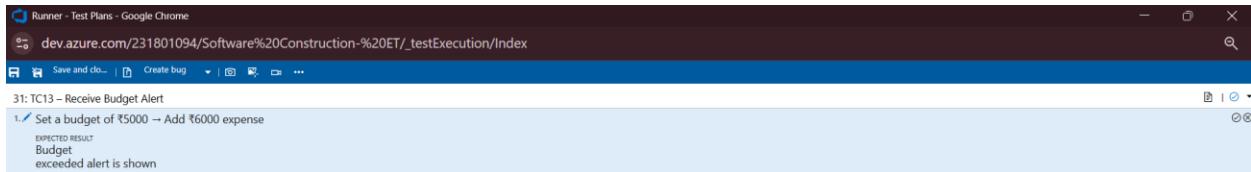
Test and feedback

Showing it as an extension

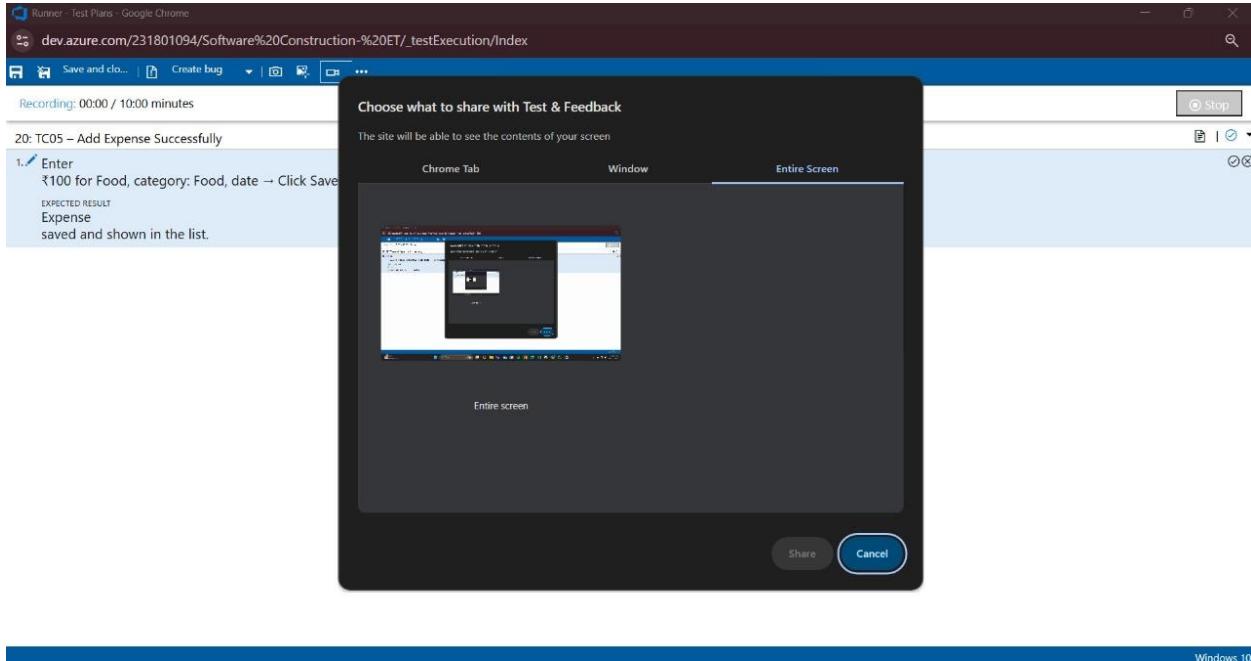
The screenshot shows the Azure DevOps Test Plans interface for a project named "Music Playlist Batch Creator". The left sidebar is open, showing "Test Plans" selected. The main area displays a test suite named "TS01 - User Login (ID: 86)". A modal window titled "Extensions" is open, listing several extensions with "Full access": "Copy Text from Picture", "Dark Reader", "Monica: ChatGPT AI Assist...", "Selectext: Copy text from V...", and "Test & Feedback". The "Manage extensions" button is also visible.

5. Running the test cases

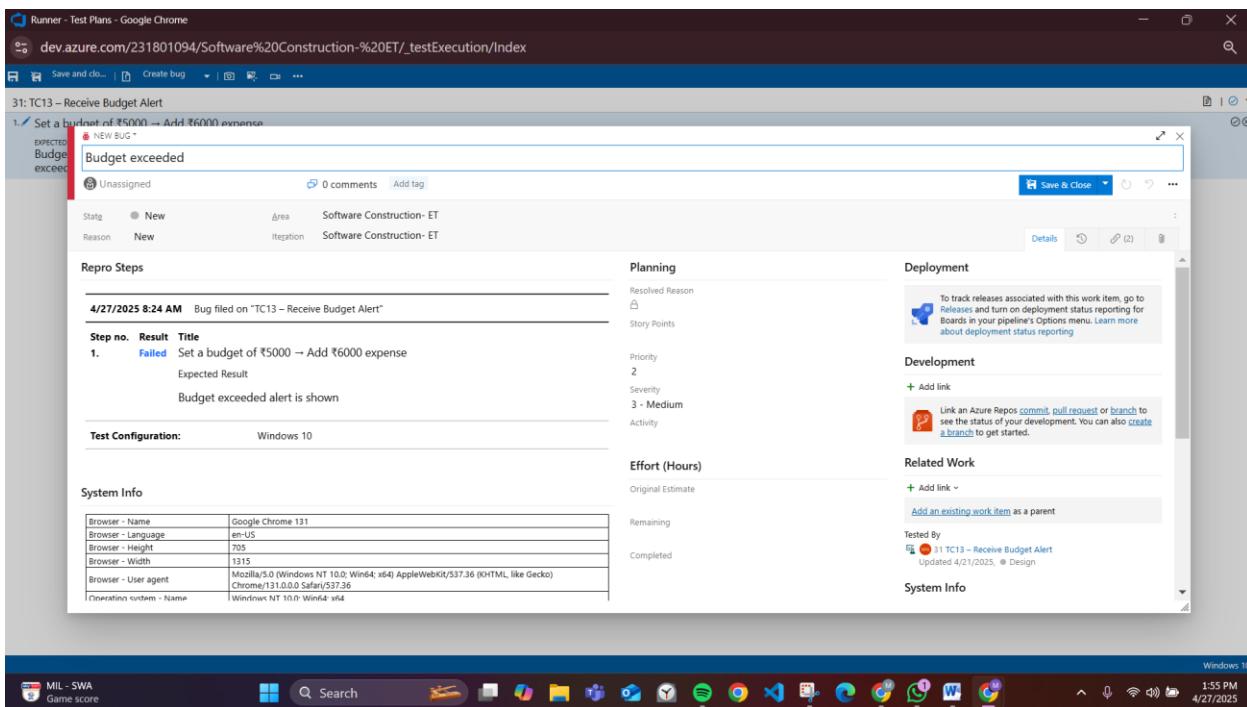
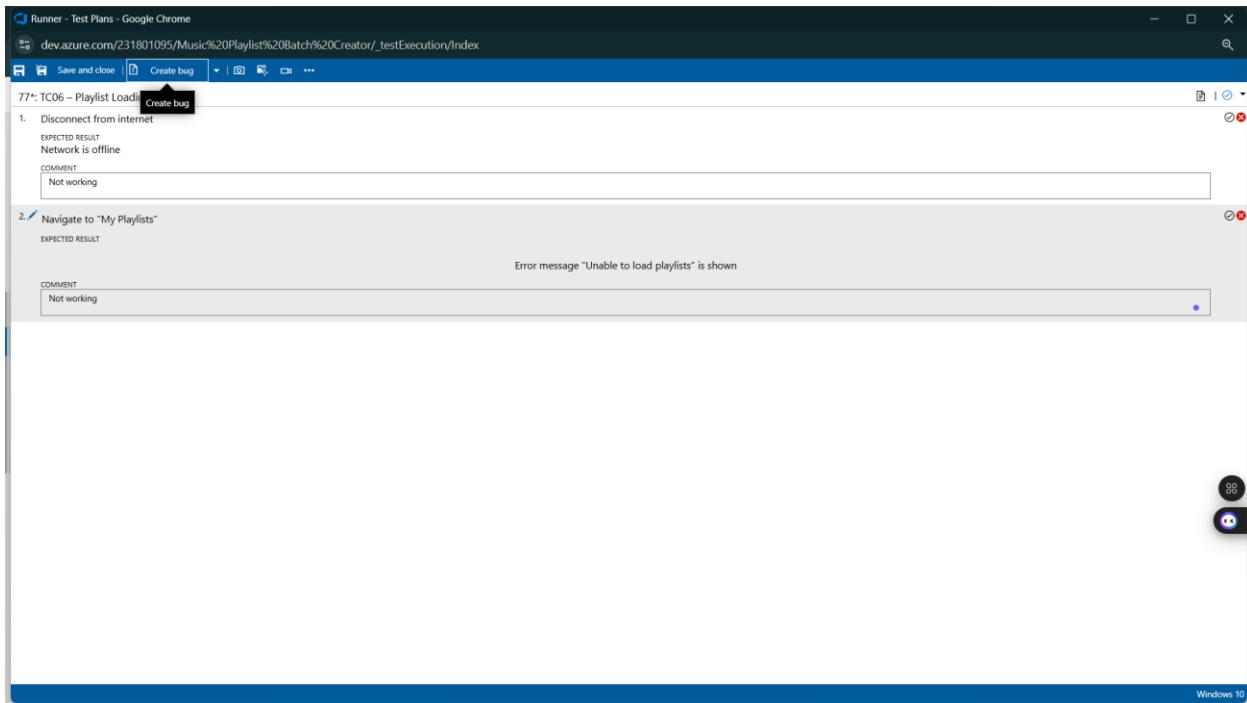
The screenshot shows the Azure DevOps Test Plans interface for a project named "Expense Tracker". The left sidebar is open, showing "Test Suites" selected. The main area displays a test suite named "Test Suite 5: Notifications & Alerts (ID: 30)". A context menu is open over a test point named "TC14 - Notification Failure". The menu options include: "View execution history", "Mark Outcome", "Run" (with sub-options "Run for web application", "Run for desktop application", and "Run with options"), "Reset test to active", "Edit test case", "Assign tester", and "View test result".



6. Recording the test case



7.Creating the bug



The screenshot shows a Microsoft Azure Test Plans interface. A bug titled "Expense edited with invalid values – No error displayed" is selected. The bug details include:

- Category:** "Cat" (highlighted in red)
- Severity:** NEW BUG
- State:** Unassigned
- Area:** Software Construction- ET
- Reason:** New
- Iteration:** Software Construction- ET
- Original estimate:** Remaining
- Completed:** Completed
- System Info:** A table showing system configuration details.
- Tested By:** Mahisha Parameshwari (Updated 48 minutes ago, Design)
- System Info:** Found in Build, Integrated in Build

8. Test case results

The screenshot shows the Microsoft Azure Test Plans interface with the following details:

- Project:** 231801094 / Software Construction- ET / Test Plans / Expense Tracker
- Test Suites:** Expense Tracker (Current, April 21 - April 28, 0% run, View report)
- Test Suite Selected:** Test Suite 1: User Login & Sign-Up (ID: 13)
- Test Points:** 4 items (Successful Sign Up, Successful Login, Sign Up with Existing Email, Login with Wrong Password). "Successful Sign Up" is highlighted.
- Test Case Results:** A modal window titled "Successful Sign Up" showing the results of the test point "Successful Sign Up". The table includes columns: Outcome, TimeStamp, Configuration, Run by, Tester, and Test Plan. One row is shown: Passed, 2m ago, Windows 10, Mahisha Parameshwari, Mahisha Parameshwari, Expense...

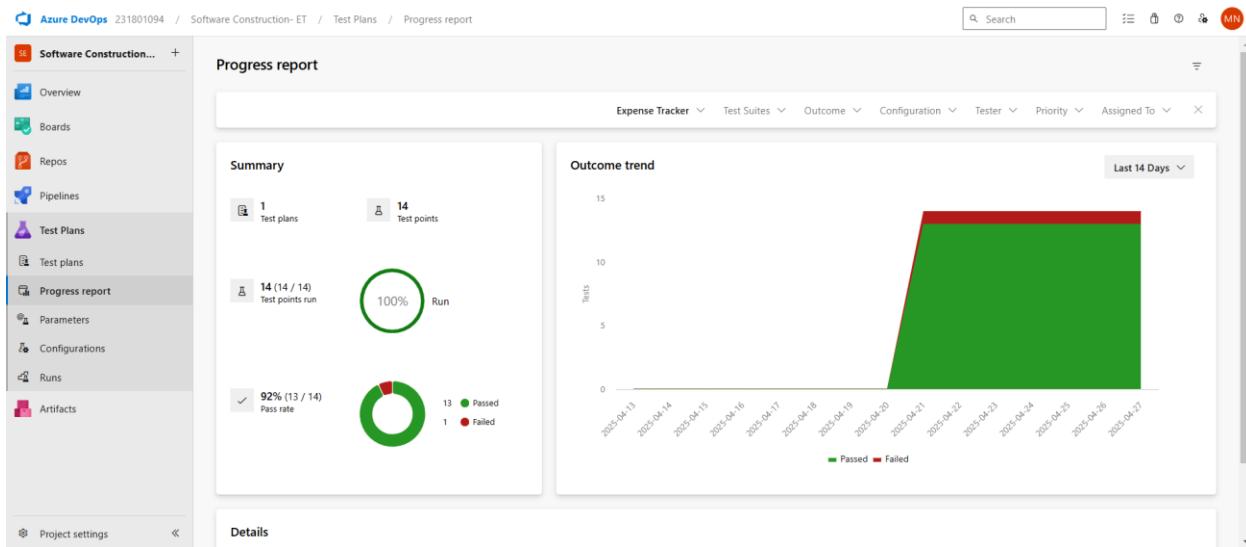
9. Test report summary

The screenshot shows the Azure DevOps interface for a 'News Feed App' project. A specific work item, 'BUG 203', is selected. The title is '203 - BG 01 - Countries Drop down Not Available on the page'. The 'Repro Step' section details three steps: 1. Passed (Open Chrome Browser, Expected Result: Browser 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), and 3. Failed (System should show a dropdown with list of countries, Expected Result: Produce Report). The 'Planning' and 'Deployment' sections are visible on the right.

- Assigning bug to the developer and changing state

The screenshot shows a browser window titled 'Runner - Test Plans - Google Chrome' with the URL 'dev.azure.com/231801094/Software%20Construction-%20ET/_testExecution/Index'. A modal dialog is open for 'TC08 – Edit Expense with Invalid Data'. The dialog shows a 'NEW BUG' section with a red error icon. The 'Repro Steps' section contains one step: '1. Failed Leave "Category" empty and save the expense'. The 'Expected Result' is 'Error message "Category cannot be empty" is shown.' The 'Test Configuration' is listed as 'Windows 10'. The 'Planning' and 'Deployment' sections are visible on the right.

10. Progress report



11. Changing the test template

The screenshot shows the 'Process' settings for the 'Test Case' template under the 'MyAgile' process. The left sidebar lists general organization settings, security, boards, and pipelines. The main area shows the 'Test Case' template configuration, which includes fields for 'Steps' (Text (multiple lines)), 'Recent test results' (Recent test case results), 'Deployment' (Deployments), 'Development' (Links), 'Related Work' (Links), and 'Status' (Priority integer, Automation status Tovef / (multiple lines)). A 'New field' button is visible at the top of the form.

The screenshot shows the 'All processes' list in the Azure DevOps Settings - Process section. The 'Agile' template is selected, indicated by a grey background. The 'Basic' template is also listed.

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

The screenshot shows the 'All processes' list in the Azure DevOps Settings - Process section. The '231801095 Agile (default)' template is selected, indicated by a grey background. Other templates like 'Basic', 'Agile Plus', 'Scrum', and 'CMMI' are also listed.

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 over a background of the Azure DevOps settings interface. The dialog has tabs for 'Definition', 'Options', and 'Layout'. Under 'Definition', the 'Create a field' option is selected, showing fields for Name (Type), Type (Text (single line)), and Description. A preview pane on the right shows the field being added to a test case template. Buttons at the bottom include 'Add field' (highlighted in blue) and 'Cancel'.

The screenshot shows the 'Test Case' settings page under 'All processes > 231801095 Agile'. The 'Steps' section contains a 'Custom' field with the type 'Text (single line)', labeled 'Recent test results'. Other sections like 'Deployment', 'Development', 'Related Work', and 'Status' are also visible. The left sidebar shows the 'Process' tab is selected.

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:

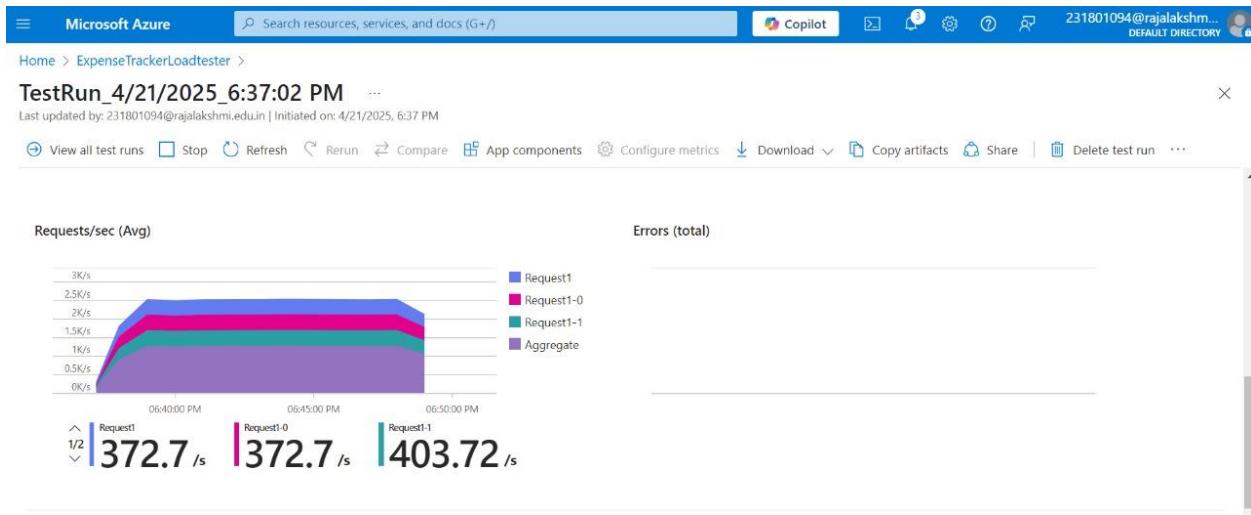
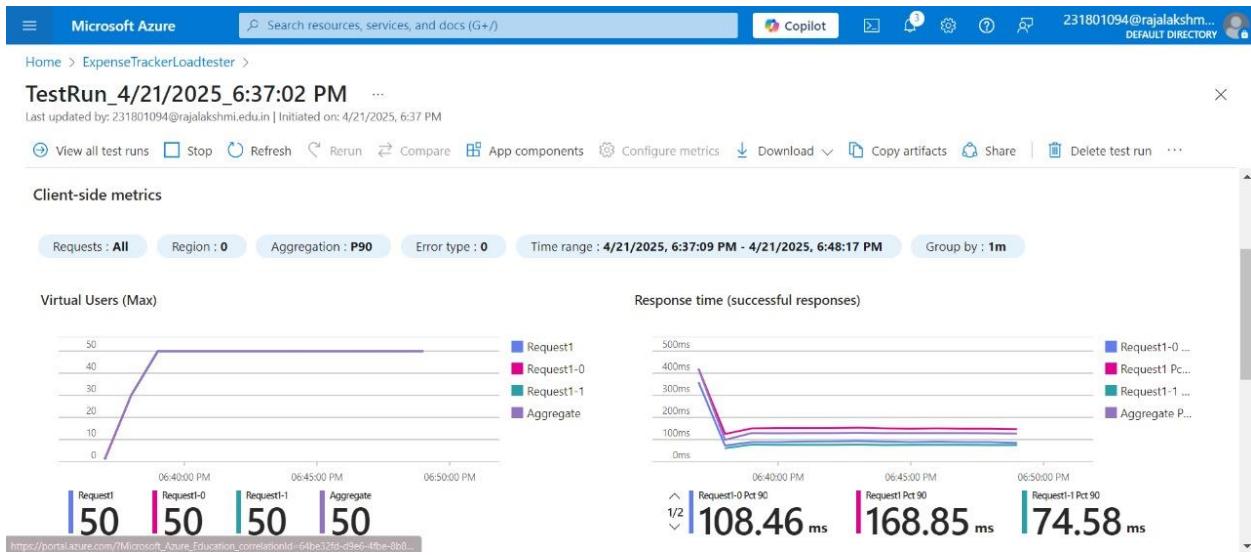
1. Sign in to Azure Portal
Go to <https://portal.azure.com> and log in.
2. Create the Resource
 - o Go to *Create a resource* → Search for “Azure Load Testing”.
 - o Select Azure Load Testing and click Create.
3. Fill in the Configuration Details
 - o *Subscription*: Choose your Azure subscription.
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

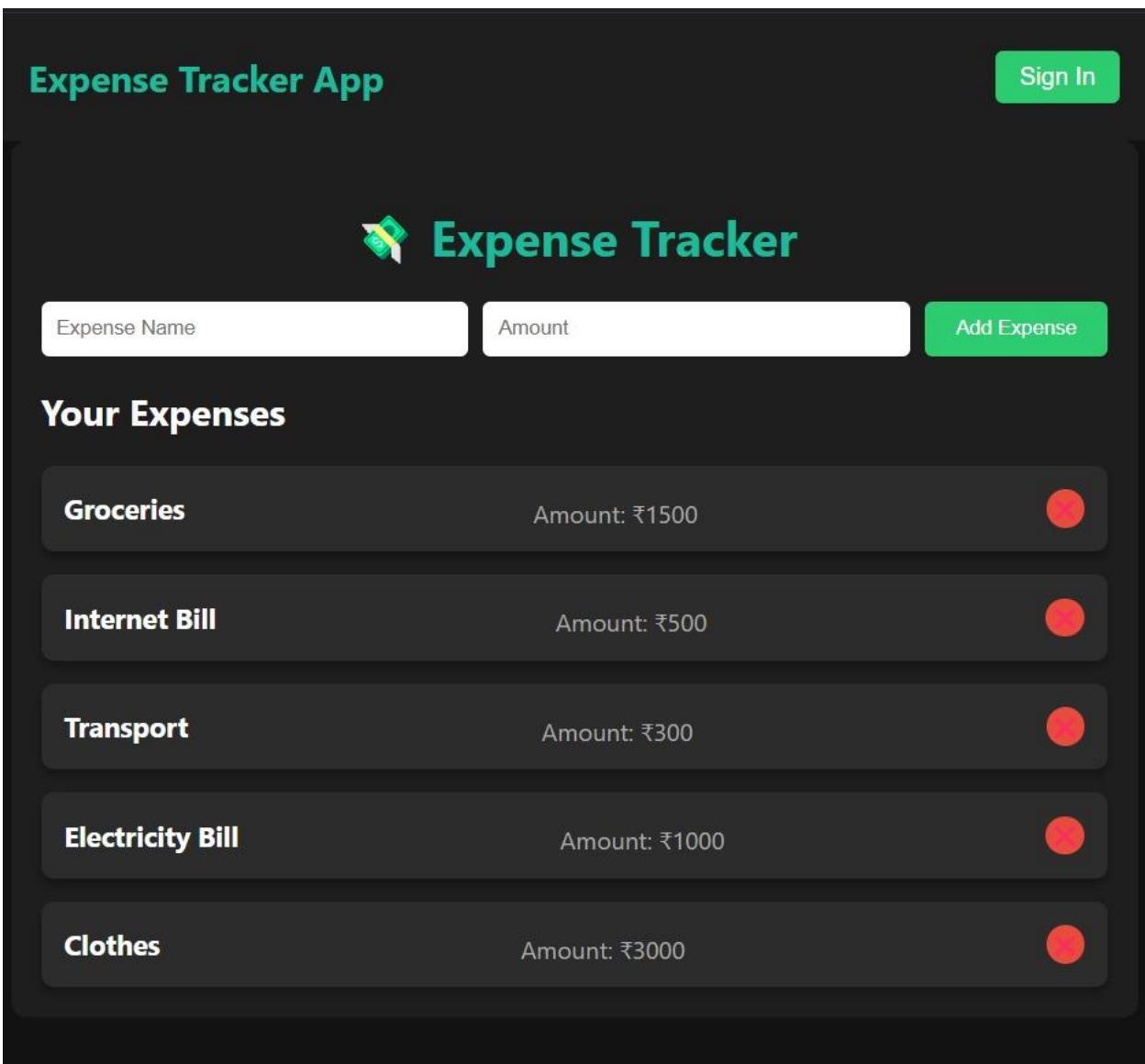
Steps to Create and Run a Load Test:

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





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.

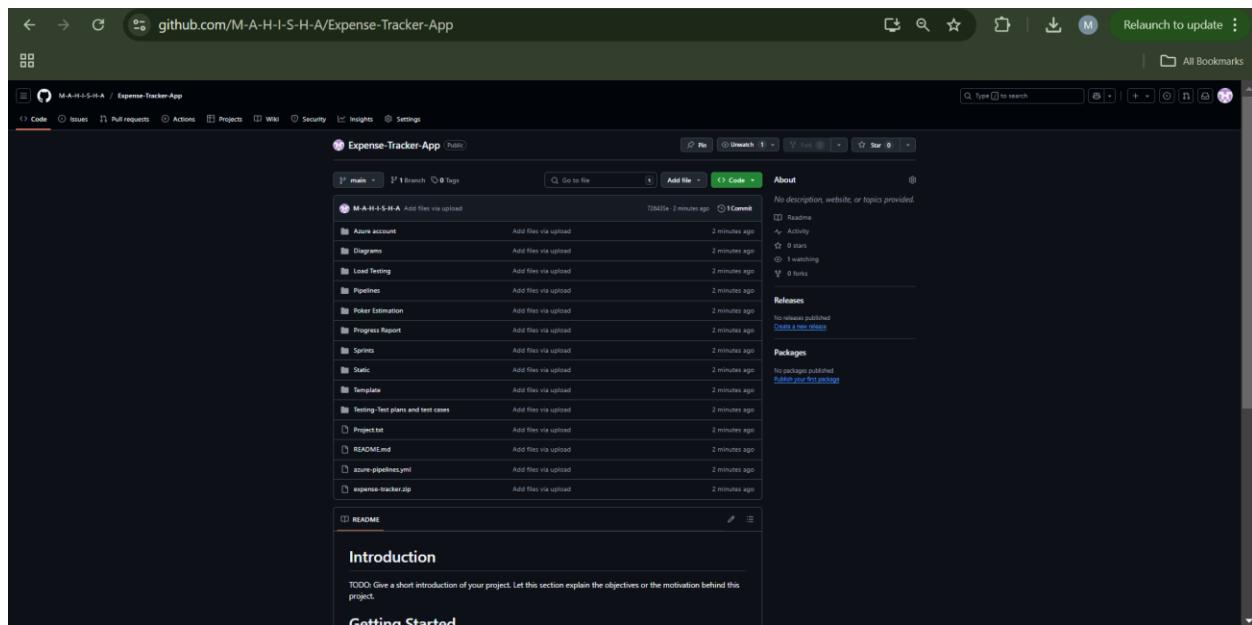
EXP NO: 10

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 App 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.