



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: MAHALAKSHMI S

Register No: 231801093

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:

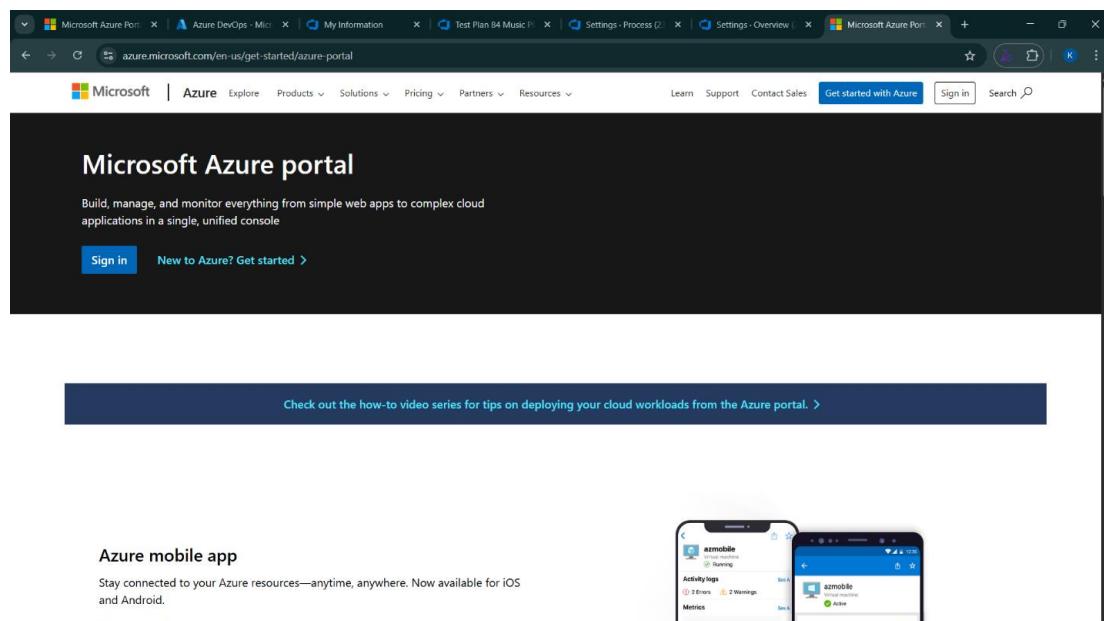
To set up and access the Azure DevOps environment by creating an organization through the Azure portal.

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 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 a 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, 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 two items: "Music" (Azure Load Testing) and "Music_playlist_Batch_Creator" (Resource group). A "See all" link is also present. To the right of the resources is a vertical sidebar with a "Last Viewed" section showing "Music" and "Music_playlist_Batch_Creator" both last viewed 3 days ago. Below this is a "Navigate" section with links for Subscriptions, Resource groups, All resources, and Dashboard. Under "Tools", there are links for Microsoft Learn, Azure Monitor, Microsoft Defender for Cloud, and Cost Management. At the bottom, there's a "Useful links" section and an "Azure mobile app" link.

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 again, but with a search bar at the top containing the text "DevOps". The rest of the interface is identical to the previous screenshot, showing the Azure services dashboard, recent resources, and various tools and documentation links.

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.

The screenshot shows the Microsoft Azure portal interface. The address bar displays 'portalazure.com/#view/AzureTfExtension/OrganizationsTemplateBlade'. The main content area features the 'Azure DevOps' branding with a sub-headline: 'Plan smarter, collaborate better, and ship faster with a set of modern dev services'. Below this, there's a section for 'My Azure DevOps Organizations' and links for 'Get started using Azure DevOps', 'Billing management for Azure DevOps', and 'Give feedback'. A large, colorful illustration depicts a rocket launching from a base of code and data, with several people working on various components like databases and code repositories. At the bottom right of the main content area is a small circular icon with a question mark.

The screenshot shows a Microsoft Edge browser window with multiple tabs open, including 'Inbox', 'AY 24-II', 'SC Man...', 'Usin...', 'Create...', 'Create...', 'Azure...', 'My Info...', and 'Signup'. The active tab is 'Signup' at 'aex.dev.azure.com/signup/?acquisitionId=be1dff...'. The page displays a sign-up form for Azure DevOps. It includes a placeholder email address 'akiladevi.r@rajalakshmi.edu.in', a heading 'Get started with Azure DevOps', and a note about agreeing to the Terms of Service, Privacy Statement, and Code of Conduct. There is a checked checkbox for receiving information from Azure DevOps and other Microsoft products. A prominent blue 'Continue' button is at the bottom right. The background of the page features a cartoon illustration of a person walking a dog under a cloud. The Windows taskbar at the bottom shows various pinned icons and the system tray.

Result:

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

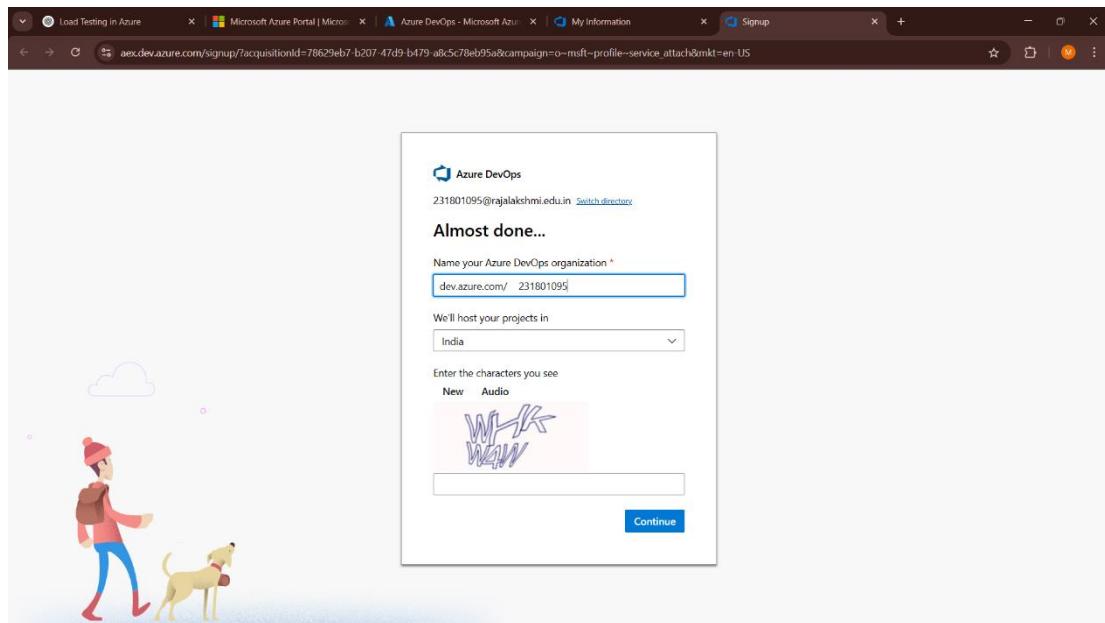
EXP NO: 2

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



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.
- 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).
- 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 [?](#)

Work item process [?](#)

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 dashboard. On the left, there's a user profile for Krithika.M.A. with a purple circular icon containing a white 'K'. Below the profile are details: name, email (krithika7604@gmail.com), location (India), and contact information (krithika7604@gmail.com). To the right, the 'Azure DevOps Organizations' section lists 'dev.azure.com/krithika7604' (Owner) with a project named 'E-commerce product uploader'. There are 'Actions' like 'Open in Visual Studio' and a 'Create new organization' button. Below this, another entry for 'dev.azure.com/231801065' (Member) is shown. On the far left, a sidebar for 'Visual Studio Dev Essentials' offers benefits for building and deploying apps. The bottom of the screen shows a Windows taskbar with various pinned icons.

4. Project dashboard

The screenshot shows the 'E-commerce product uploader' project dashboard. The left sidebar includes options like Overview, Summary (which is selected), Dashboards, Wiki, Boards, Repos, Pipelines, Test Plans, and Artifacts. The main area features a welcome message 'Welcome to the project!' with a sub-question 'What service would you like to start with?'. Below this are tabs for Boards, Repos, Pipelines, Test Plans, and Artifacts (which is currently selected). To the right, there's a 'Project stats' section with a note 'No stats are available at this moment. Setup a service to see project activity.' and a 'Members' section showing one member. The bottom of the screen shows a Windows taskbar.

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 DevOps interface for managing work items. The left sidebar has 'Work items' selected. The main area displays a table of work items with columns for ID, Title, Assigned To, State, and Area Path. A tooltip indicates that the 'New' state means the item is a User Story. The table includes rows for various items like 'Bug', 'Epic', 'Feature', 'Issue', 'Task', 'Test Case', 'Product Upload Management', 'User Authentication and Authorisation', and 'Product Approval and Moderation'. The 'Assigned To' column shows names like 'madhusha', 'Janani Vasu', 'Kritika.MA', 'Janani Rajan', and 'Mahalakshmi231801093'.

The screenshot shows the Microsoft account sign-in page. It features a large purple circular profile picture with a white letter 'K'. The user's name 'Krithika.M.A' and email 'krithika7604@gmail.com' are displayed. There are links for 'My Microsoft account' and 'Switch directory'. Below the profile, there is a 'Sign in with a different account' button.

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 Boards backlog for the 'E-commerce product uploader' team. The backlog is organized into several levels of hierarchy:

- Level 1:** An Epic titled "User Authentication and Authorization".
- Level 2:** A Feature titled "Login page for users".
- Level 3:** A User Story titled "As a user, I want to log in using my email and password".
- Level 4:** Three Tasks under the User Story: "Design the Login Page UI", "Add Validation for Input Fields", and "Handle Incorrect Login Attempts".
- Level 5:** Two more User Stories under the Epic: "As a user, I want to reset my password if I forget it, so ..." and "As a user, I want to authenticate using my social media ...".
- Level 6:** A Feature titled "Login Page for Product Uploaders".

The backlog table includes columns for Order, Work Item Type, Title, State, Effort, Business Area, and Tags. The Business Area column shows "Business" for most items, except for the Feature at Level 6 which is listed as "Businesses".

1.Fill in Epics

The screenshot shows the 'Work items' screen in Azure DevOps, specifically the 'Create Epic' form. The form is titled 'Product Categorization and Tagging' and contains the following details:

- State:** New
- Reason:** New
- Area:** E-commerce product uploader
- Iteration:** E-commerce product uploader\Sprint 1

The 'Description' field is empty, and the 'Planning' section shows a priority of 2. The 'Deployment' section notes that releases associated with this work item can be tracked via the 'Releases' tab. The 'Development' section includes a 'Add link' button and a note about linking to a pull request or branch.

2.Fill in Features

The screenshot shows the Azure DevOps interface for creating a new work item. The URL is https://dev.azure.com/231801065/E-commerce%20product%20uploader/_workitems/create/Feature. The left sidebar shows the project navigation. The main area is titled "Work items" and displays a "NEW FEATURE" card with the title "Login page for users". The card includes fields for State (New), Area (E-commerce product uploader), Reason (New), Iteration (E-commerce product uploader\Sprint 1), and a "Description" section with a placeholder "Click to add Description.". The "Planning" and "Deployment" sections are visible on the right, along with a "Development" section containing a "Add link" button. The status bar at the bottom shows the date as 28-04-2025.

3.Fill in User Story Details

The screenshot shows the Azure DevOps interface for creating a new work item. The URL is https://dev.azure.com/akiladev/LMS/_workitems/create/User%20Story. The left sidebar shows the project navigation. The main area is titled "Work items" and displays a "NEW USER STORY" card with the title "Login". The card includes fields for State (New), Area (LMS), Reason (New), Iteration (LMS), and a "Description" section with a placeholder "As a user I can login using regno and password so I can access my account.". The "Acceptance Criteria" and "Classification" sections are visible on the right, along with a "Development" section containing a "Add link" button. The status bar at the bottom shows the date as 28-04-2025.

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 E Commerce Product UploaderProject.

Sprint Planning

The screenshot shows the Azure DevOps Boards interface. On the left, there's a sidebar with options like Overview, Boards, Work items, Backlogs, Sprints, Queries, Delivery Plans, Pipelines, and Artifacts. The main area is titled 'E-commerce product uploader Team' and shows 'Sprint 1'. The backlog section lists several user stories, and the sprint section shows five tasks assigned to different team members. Task 5 is highlighted.

Order	Title	State	Assigned To	Remaining
1	As a user, I want to authenticate using my social media account so that I can regain access to my account.	New	Janani Rajan	10 work days remaining
2	As a user, I want to reset my password if I forget it, so that I can regain access to my account.	New	Mahalakshmi23...	
3	As a user, I want to log in using my email and password so that I can regain access to my account.	New	Krithika.MA	
4	As a product uploader, I want to log in using my credentials so that I can upload products.	New	231801064@raj...	
5	As a product uploader, I want to authenticate using a role-based authentication system so that I can upload products.	New	231801091@raj...	

Sprint 1

The screenshot shows the Taskboard view for Sprint 1. The backlog is collapsed, and the active tasks are listed in a grid. The tasks are categorized by state: New, Active, Resolved, and Closed. Task 58 is currently active.

Person: All	New	Active	Resolved	Closed
	58 Design the Login UI New Unassigned			
	59 Add Validation for Input Fields. New Unassigned			
	60 Handle incorrect Login attempts. New Unassigned			
	5 As a user, I want to authenticate using my social media account so that I can regain access to my account. New Unassigned			

Sprint 2

The screenshot shows the Azure DevOps Taskboard for the 'E-commerce product uploader' project. The left sidebar navigation includes 'Overview', 'Boards', 'Work items', 'Backlogs', 'Sprints' (selected), 'Queries', 'Delivery Plans', 'Analytics views', 'Repos', 'Pipelines', 'Test Plans', 'Artifacts', and 'Project settings'. The main area displays the 'Taskboard' for 'Sprint 2' from March 3 to March 14, which spans 10 work days. The board has four columns: 'New', 'Active', 'Resolved', and 'Closed'. Three user stories are listed under the 'New' column:

- 12 As a product uploader, I want to enter product details (name, description, price) so that I can list the product on the website.
 - New
 - Unassigned
- 13 As a product uploader, I want to upload product images so that customers can view the product visually.
 - New
 - Unassigned
- 14 As a product uploader, I want to assign categories and tags to the product so that it is listed correctly.
 - New
 - Unassigned

Sprint 3

The screenshot shows the Azure DevOps Taskboard for the 'E-commerce product uploader' project. The left sidebar navigation includes 'Overview', 'Boards', 'Work items', 'Backlogs', 'Sprints' (selected), 'Queries', 'Delivery Plans', 'Analytics views', 'Repos', 'Pipelines', 'Test Plans', 'Artifacts', and 'Project settings'. The main area displays the 'Taskboard' for 'Sprint 3' from March 17 to March 28, which spans 10 work days. The board has four columns: 'New', 'Active', 'Resolved', and 'Closed'. Three user stories are listed under the 'New' column:

- 22 As an admin, I want to review all newly uploaded products before they are listed on the website so that only quality products are listed.
 - New
 - Unassigned
- 23 As an admin, I want to provide feedback on rejected products to the uploader, so they can make necessary changes.
 - New
 - Unassigned
- 25 As a product uploader, I want to receive notifications when my product is approved or rejected.
 - New
 - Unassigned

Result:

The Sprints are created for the E-Commerce Product Uploader Project

EXP NO: 5

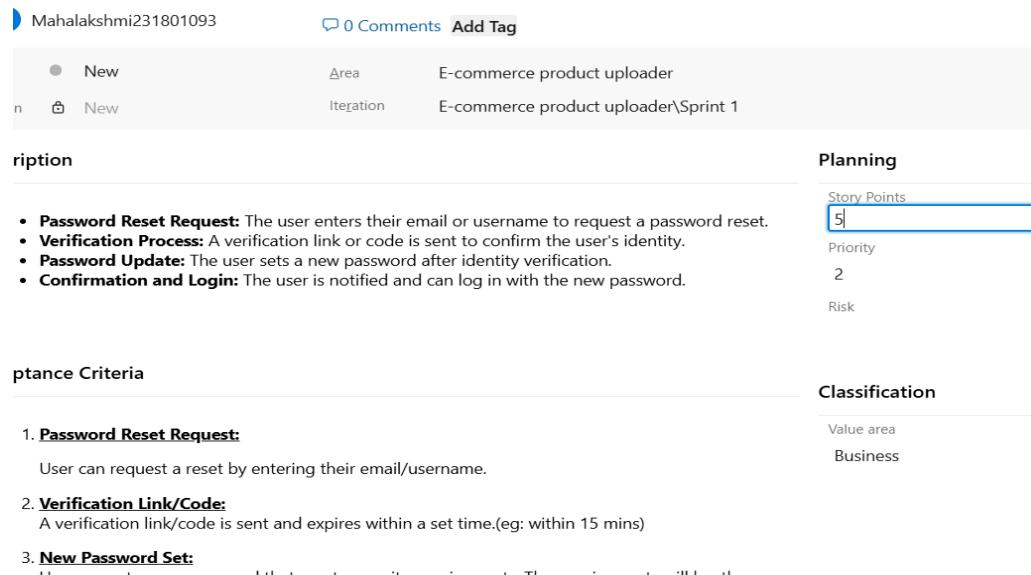
POKER ESTIMATION

Aim:

Create Poker Estimation for the user stories - E Commerce Product Uploader Project.

Poker Estimation

As a user, I want to reset my password if I forget it, so that I can regain access to my account.

A screenshot of a software interface showing a user story card. The card has the following details:

- Owner: Mahalakshmi231801093
- Comments: 0 Comments
- Add Tag
- Status: New
- Area: E-commerce product uploader
- Iteration: E-commerce product uploader\Sprint 1
- Description: (partially visible)
- Planning:
 - Story Points: 5
 - Priority: 2
 - Risk: (partially visible)
- Classification:
 - Value area: Business

Result:

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

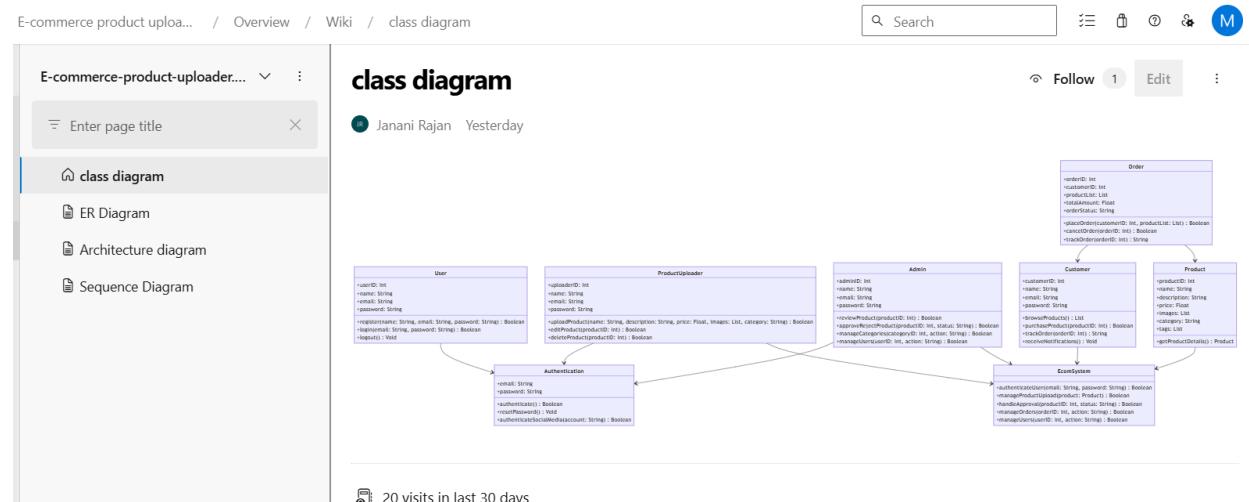
EXP NO: 6

DESIGNING CLASS AND SEQUENCE DIAGRAMS FOR PROJECT ARCHITECTURE

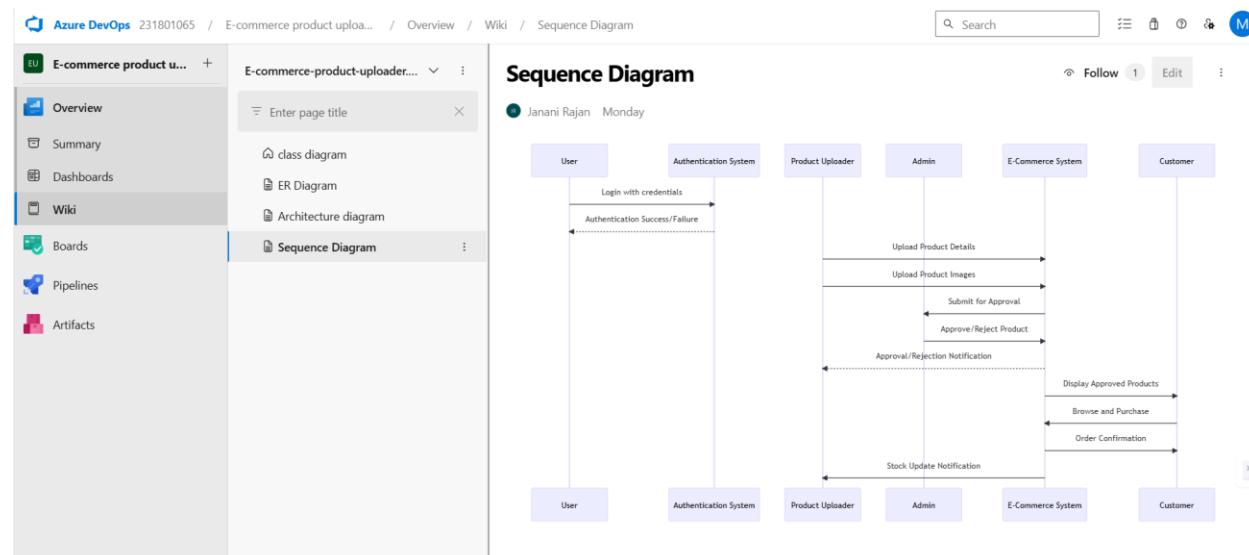
Aim:

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 E-Commerce Product Uploader.

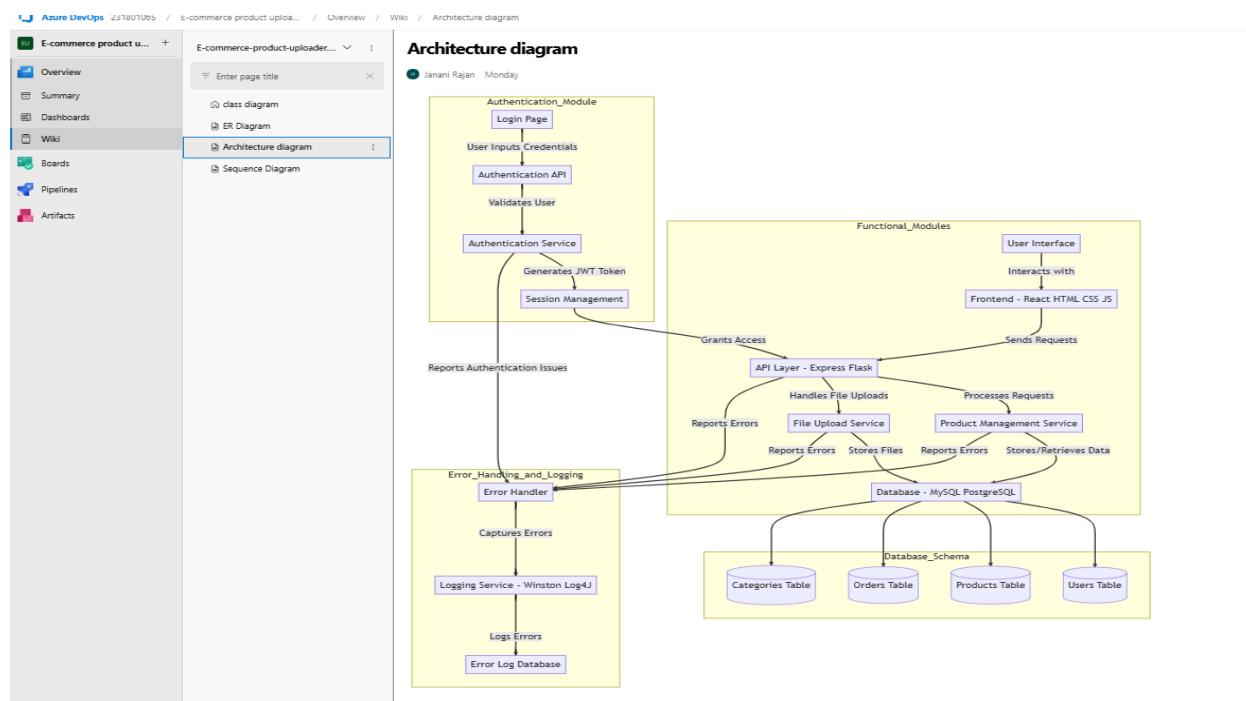
EXP NO: 7

DESIGNING ARCHITECTURAL AND ER DIAGRAMS FOR PROJECT STRUCTURE

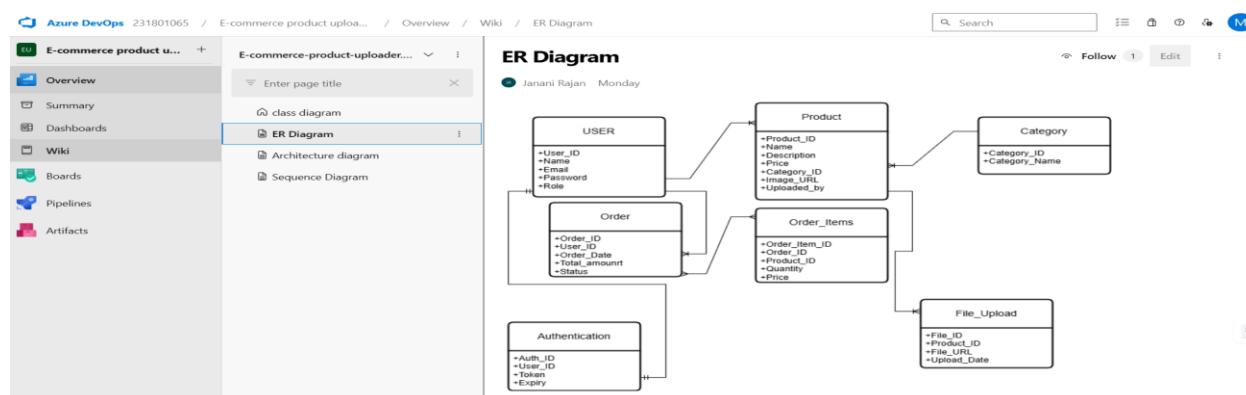
Aim:

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 E-Commerce Product Uploader.

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****1. Understand Core Features of the Application**

- User Signup & Login
- Viewing and Managing Playlists
- Fetching Real-time Metadata
- Editing playlists (rename, reorder, record)
- Creating smart audio playlists based on categories (mood, genre, artist, etc.)

2. Define User Interactions

- Each test case simulates a real user behaviour (e.g., logging in, renaming a playlist, adding a song).

3. Design Happy Path Test Cases

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

4. Design Error Path Test Cases

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

5. Break Down Steps and Expected Results

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

6. Use Clear Naming and IDs

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

7. Separate Test Suites

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

- Improves organization and test execution flow in Azure DevOps.

8. Prioritize and Review

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

1. New test plan

New Test Plan

Name *: E-commerce product uploader - Test plan1

Area Path *: E-commerce product uploader

Iteration *: E-commerce product uploader\Sprint 1

Create Cancel

2. Test suite

Title	Outcome	Order	Test Case Id
Check product categories are displayed in the left	Passed	1	92
Check product categories are displayed in the left	Passed	1	92
Static suite	Passed	2	93
Requirement based suite	Passed	2	93
Query based suite	Failed	3	99
Validate Login with parameters	Active	3	99

3.Test case

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

E Commerce Product Uploader– Test Plans

USER STORIES

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

Test Suites

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

1. TC01 – Successful Sign Up

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

2. TC02 – Secure Login

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

3. TC03 – Sign Up with Existing Email

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

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

4. TC04 – Login with Wrong Password

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

Test Suit: TS02 - View Playlists (ID: 87)

1. TC05 – View Playlist Page

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

2. TC06 – Playlist Loading Failure

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

Test Suit: TS03 - Real-Time Metadata (ID: 88)

1. TC07 – Real-Time Metadata Display

- **Action:**
 - Play a song.
 - Observe the metadata panel.
- **Expected Results:**
 - Metadata (title, artist, album, duration) is displayed and updates in real time.
- **Type:** Happy Path

2. TC08 – Metadata Not Updating

- **Action:**

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

Test Suit: TS04 - Playlist Editing (ID: 89)

1. TC09 – Rename Playlist Successfully

- **Action:**
 - Navigate to "My Playlists".
 - Click "Rename" next to a playlist.
 - Enter a new name and click "Save".
- **Expected Results:**
 - Playlist name updates successfully.
- **Type:** Happy Path

2. TC10 – Rename with Blank Name

- **Action:**
 - Click "Rename" on a playlist.
 - Leave the field blank.
 - Click "Save".
- **Expected Results:**
 - Error message "Playlist name cannot be empty" is shown.
- **Type:** Error Path

3. TC11 – Change Playlist Order

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

4. TC12 – Change Playlist Order Fails

- **Action:**
 - Login and go to "My Playlists".
 - Select a playlist.
 - Go offline or simulate server error.
 - Reorder songs and click "Save Order".
- **Expected Results:**
 - Error message: "Failed to update order. Please check your connection".
- **Type:** Error Path

Test Suit: TS05 - Smart Playlist Creation (ID: 90)

1. TC13 – Generate Playlist Based on Various Categories

- Action:

- Login with valid credentials.
- Click on "Generate Playlist".
- Select categories.
- Click "Generate Playlist".

- Expected Results:

- Playlist is generated based on selected mood and categories.

- Type: Happy Path

2. TC14 – Fail to Generate Playlist Due to Missing Category Selection or Invalid Input

- Action:

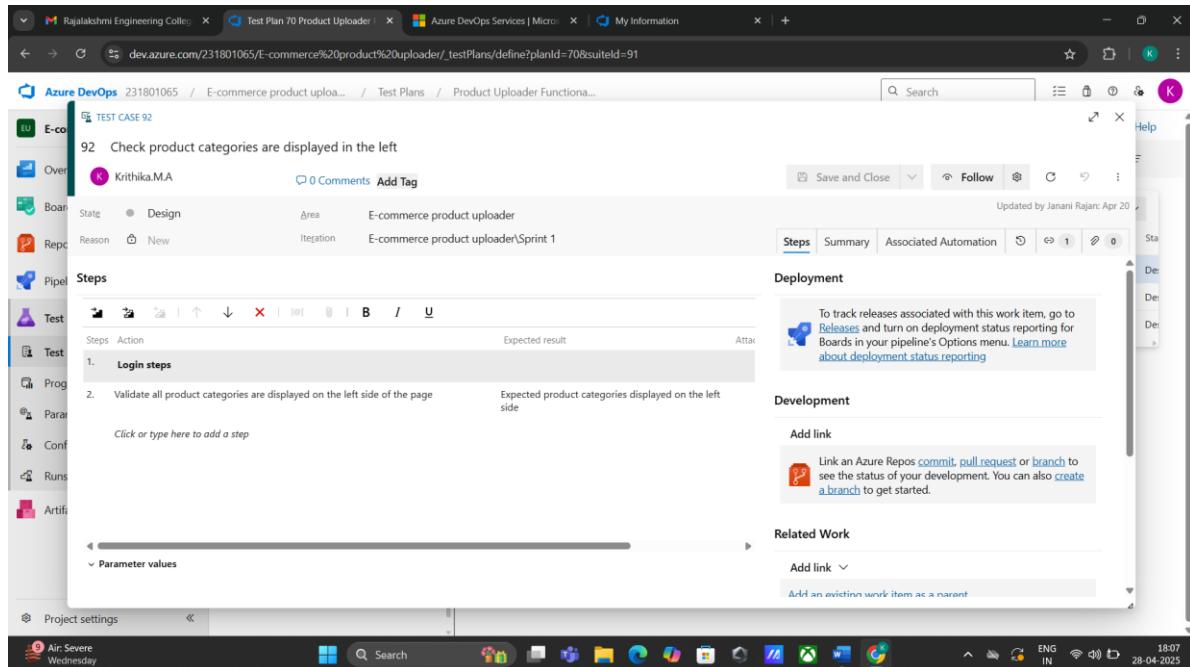
- Login with valid credentials.
- Click on "Generate Playlist".
- Select categories.
- Click "Generate Playlist".

- Expected Results:

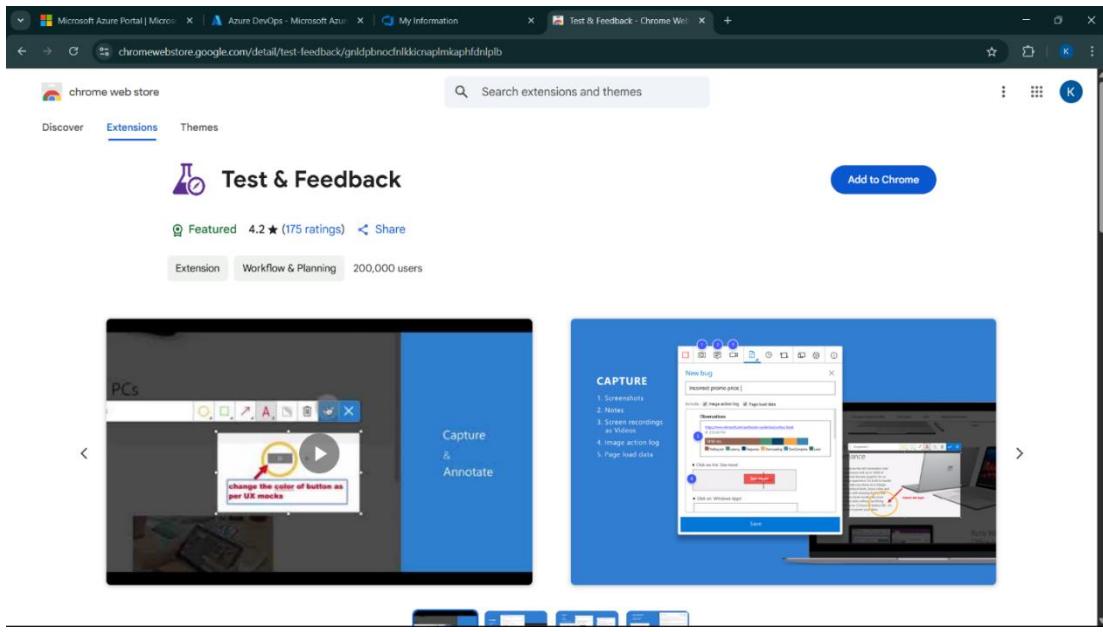
- Error message: "Please select at least one valid category" or "No recommendations found for the selected filters".

- Type: Error Path

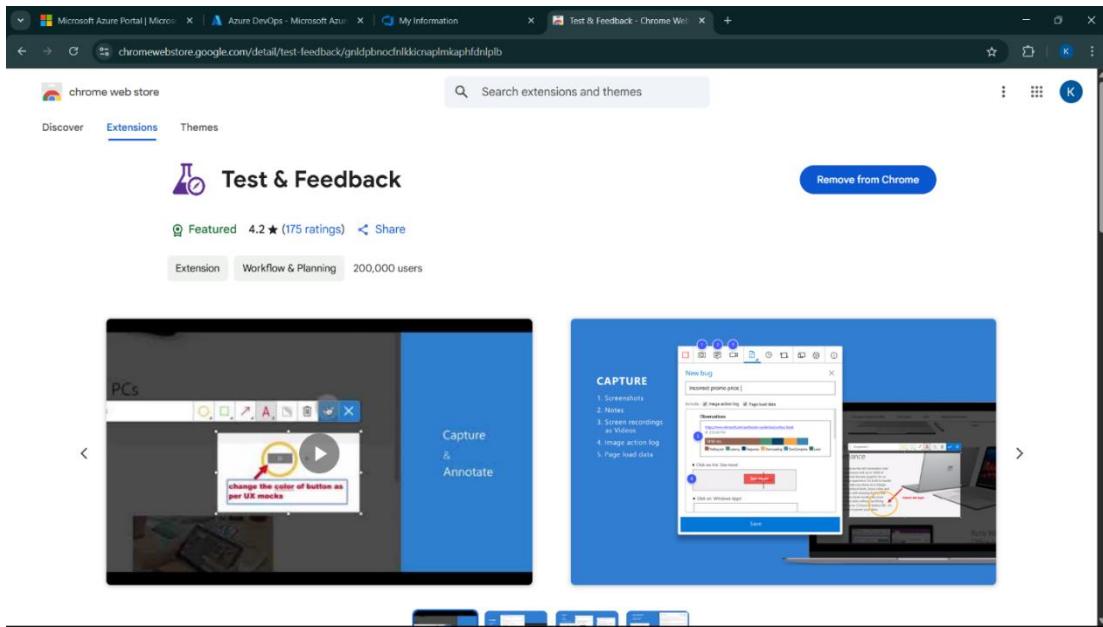
Test Cases



4. Installation of test



Test and feedback



Showing it as an extension

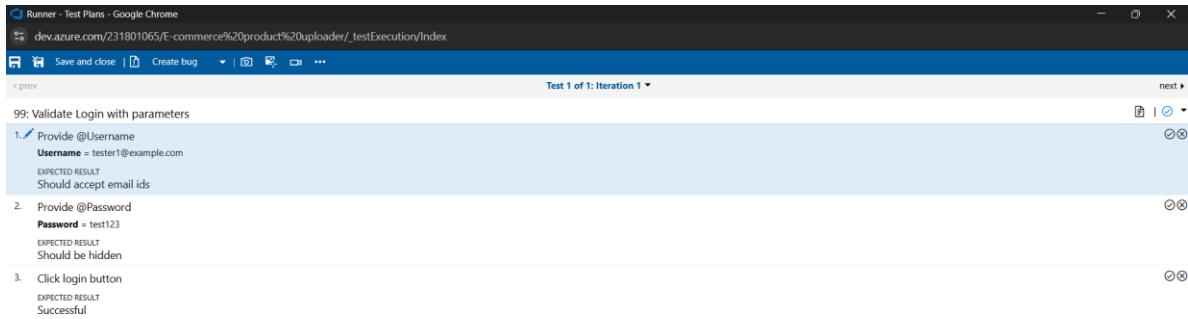
The screenshot shows the Azure DevOps Test Plan interface. On the left, the navigation bar includes 'Overview', 'Boards', 'Repos', 'Pipelines', 'Test Plans' (selected), 'Test plans', 'Progress report', 'Parameters', 'Configurations', 'Runs', and 'Artifacts'. The main area displays 'Product Uploader Functionality and...' under 'Test Suites'. A modal window titled 'ProjectTestSuite (ID: 91)' is open, showing 'Test Cases (3 items)'. The right side features an 'Extensions' panel with 'Full access' granted to 'McAfee® WebAdvisor' and 'Test & Feedback'. Below this is a table of test cases:

Test Case	Order	Author	Delegated To
1. Check product categories are displayed in the left	92	Krithika.M.A	
2. Select Specific product Category	93	Janani Rajan	
3. Validate Login with parameters	99	Janani Rajan	

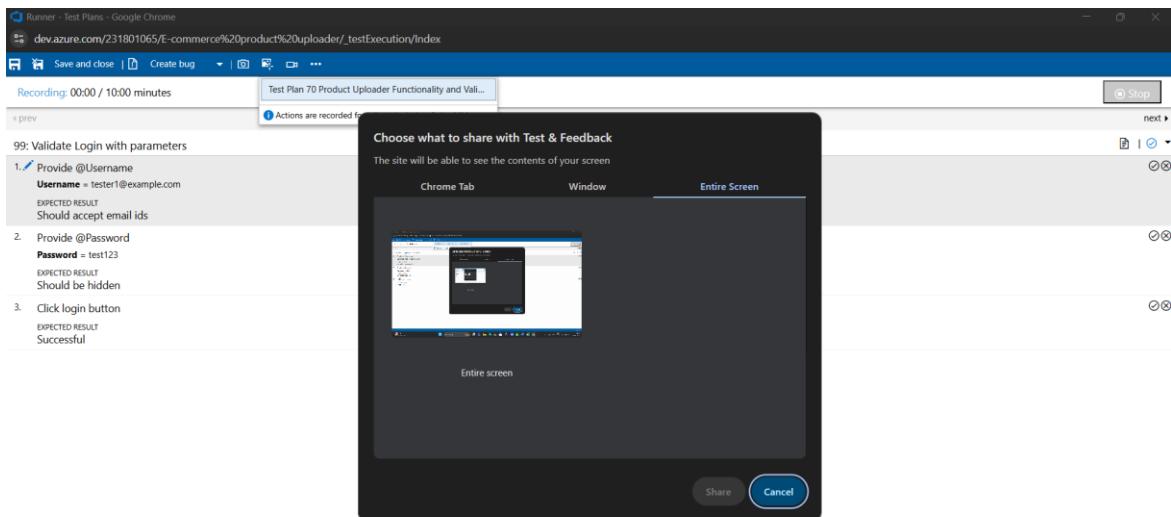
5. Running the test cases

The screenshot shows the Azure DevOps Test Plan interface with 'Test plans' selected in the navigation bar. The main area displays 'Product Uploader Functionality and...' under 'Test Suites'. A modal window titled 'ProjectTestSuite (ID: 91)' is open, showing 'Test Points (6 items)'. The right side features a context menu for the 'Validate Login with parameters' test point, with options like 'Run for web application', 'Run for desktop application', and 'Run with options'.

Test Point	Outcome	Order	Test Case Id
Check product categories are displayed in the left	Passed	1	92
Select Specific product Category	Passed	2	93
Validate Login with parameters	Failed	3	99
Validate Login with parameters	Active	3	99



6. Recording the test case



7.Creating the bug

The screenshot shows a browser window titled "Runner - Test Plans - Google Chrome" with the URL "dev.azure.com/231801065/E-commerce%20product%20uploader/_testExecution/index". The page displays a test plan for "99: Validate Login with parameters". It includes three steps: 1. Provide @Username (Expected Result: Should accept email ids), 2. Provide @Password (Expected Result: Should be hidden), and 3. Click login button (Expected Result: Successful). A "Create bug" button is visible at the top right of the test plan area.



The screenshot shows a browser window titled "Runner - Test Plans - Google Chrome" with the URL "dev.azure.com/231801065/E-commerce%20product%20uploader/_testExecution/index". A "NEW BUG" dialog box is open over the test plan. The dialog has a title bar "Issue" and a status bar "Unassigned". It contains fields for "State" (New), "Area" (E-commerce product uploader), "Reason" (New), and "Iteration" (E-commerce product uploader\Sprint 1). The "Repro Steps" section lists the three steps from the previous screenshot. To the right of the dialog are sections for "Planning" (Resolved Reason, Story Points, Priority, Severity, Activity), "Deployment" (Release notes, Deployment status reporting), "Development" (Add link, Azure Repos integration), and "Related Work" (Add existing work item).



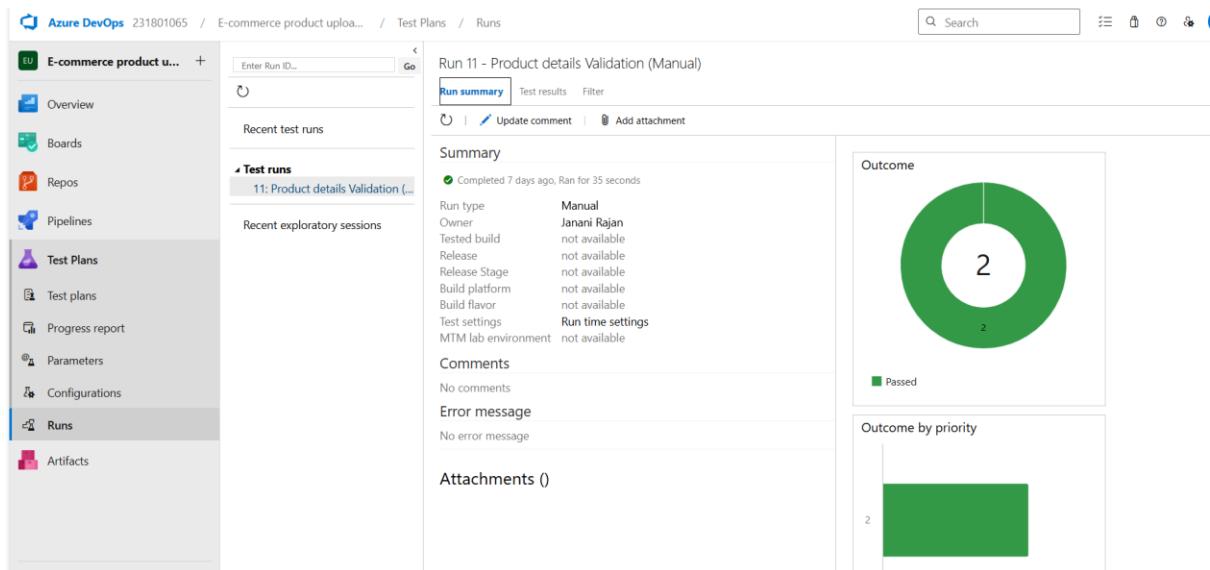
The screenshot shows the Azure DevOps interface for a project titled "E-commerce product uploader". The left sidebar is open, showing options like Overview, Boards, Repos, Pipelines, Test Plans, Test plans, Progress report, Parameters, Configurations, and Runs. The "Runs" option is selected. The main content area displays a test run titled "Run 7 - ProjectTestSuite (Manual) / Check product categories are displayed in the left". The "Summary" section indicates the run is passed, with Janani Rajan as the owner. The "Analysis" section shows failure type and resolution as "None". The "Attachments" section lists a single test step: "1. Login steps" which passed. The "Expected result" was "Login page is displayed". The status bar at the bottom shows it's 6:23 pm on April 28, 2025.

8. Test case results

This screenshot shows a detailed view of test results for a specific suite. The left sidebar is similar to the previous screenshot. The main area shows the "ProjectTestSuite (ID: 91)" details, including the title "Check product categories are displayed in the left". The "Test Case Results" table lists six test points, all of which have passed. The table includes columns for Outcome,TimeStamp, Configuration, Run by, Tester, and Test. The outcomes are all marked as "Passed". The status bar at the bottom shows it's 6:23 pm on April 28, 2025.

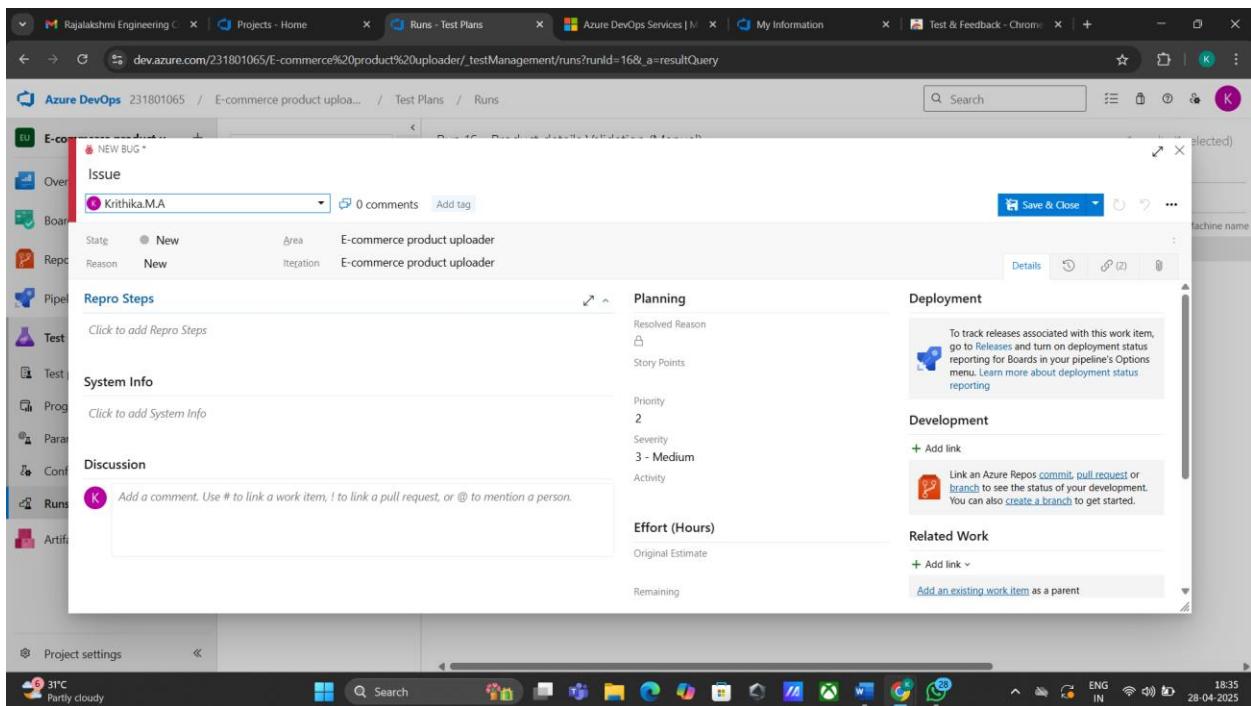
Outcome	TimeStamp	Configuration	Run by	Tester	Test
Passed	Apr 20	Firefox	Janani Rajan	Janani Rajan	Proc
Passed	Apr 20	Chrome	Janani Rajan	Janani Rajan	Proc
Passed	Apr 20	Chrome	Janani Rajan	Janani Rajan	Proc
Passed	Apr 20	Firefox	Janani Rajan	Janani Rajan	Proc
Passed	Apr 20	Chrome	Janani Rajan	Janani Rajan	Proc

9. Test report summary



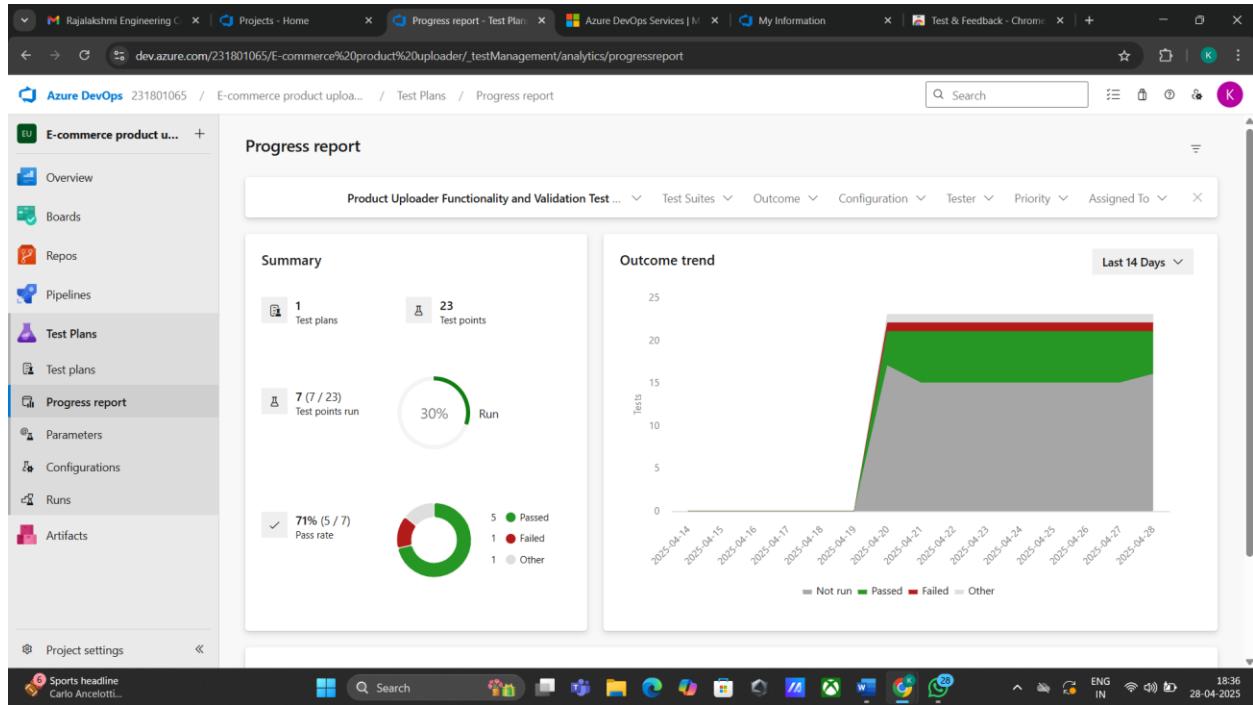
The screenshot shows the Azure DevOps interface for a test plan named "E-commerce product u...". The left sidebar is expanded to show "Test Plans" and "Runs". The main area displays the "Run 11 - Product details Validation (Manual)" summary. It includes sections for "Summary", "Comments", "Error message", and "Attachments". A large circular chart titled "Outcome" shows 2 passed tests. Another chart titled "Outcome by priority" shows 2 high-priority items.

- Assigning bug to the developer and changing state

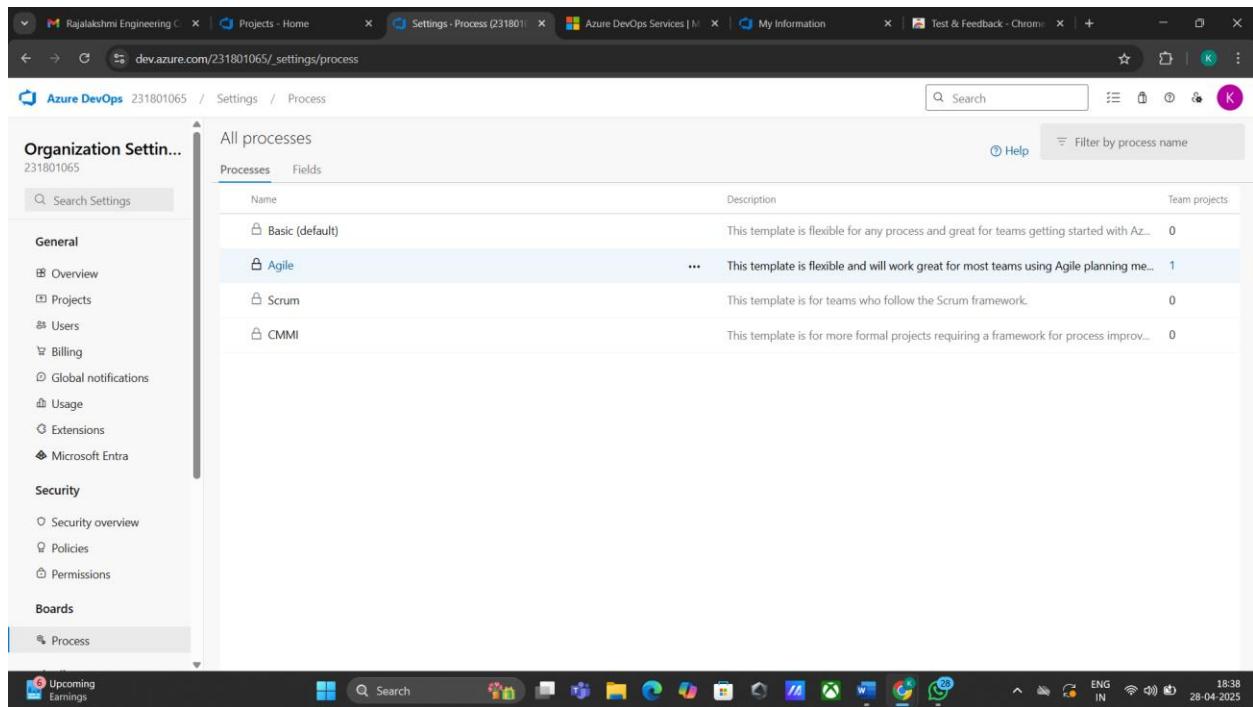


The screenshot shows the Azure DevOps interface for creating a new bug work item. The "Issue" tab is selected, showing fields for "Assignee" (Krithika.MA), "State" (New), "Area" (E-commerce product uploader), "Reason" (New), and "Iteration" (E-commerce product uploader). The "Repro Steps" section is collapsed. The "System Info" section is collapsed. The "Discussion" section contains a placeholder for comments. The "Planning" section shows "Priority" (2) and "Severity" (3 - Medium). The "Deployment" section provides instructions for tracking releases. The "Development" section allows linking to Azure Repos. The "Effort (Hours)" section shows "Original Estimate" and "Remaining" time. The "Related Work" section allows adding links to other work items. The bottom of the screen shows the Windows taskbar with various pinned icons.

10. Progress report



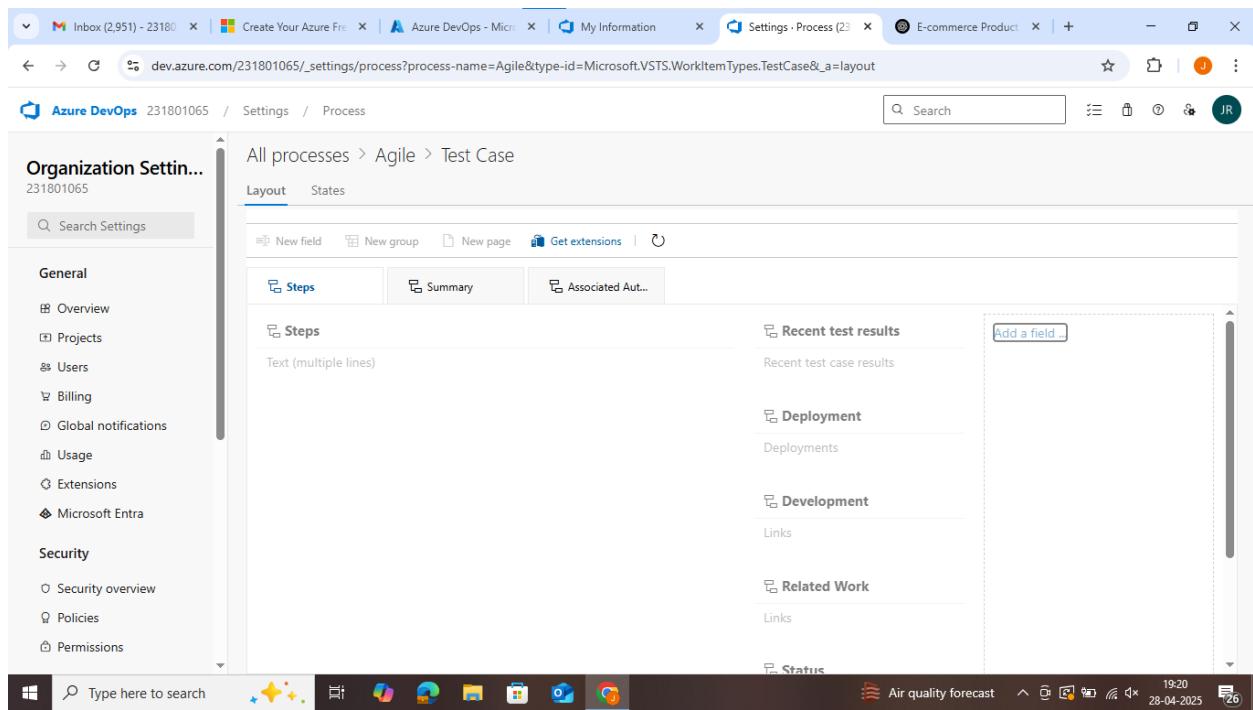
11. Changing the test template



The screenshot shows the 'Settings - Process' page in Azure DevOps. On the left, the navigation menu includes 'General', 'Security', 'Boards', and 'Process'. The 'Process' section is selected. In the center, a modal window titled 'Create inherited process from Agile' is open. It contains fields for 'Process name (required)' and 'Description'. Below the modal are 'Create process' and 'Cancel' buttons. To the right of the modal, there's a sidebar with 'Team projects' and a list of inherited processes: 'Basic (default)', 'Agile', 'Scrum', and 'CMMI'. At the bottom of the page, there's a Windows taskbar with various pinned icons.

12. View the new test case template

The screenshot shows the 'Settings - Process' page in Azure DevOps. The 'Process' section is selected in the navigation menu. A modal window titled 'Add a field to Test Case' is open. It has tabs for 'Definition', 'Options', and 'Layout'. Under 'Definition', the 'Create a field' option is selected, with 'Name' set to 'e-commerce' and 'Type' set to 'Text (single line)'. There's a 'Description' field and a 'Learn more' link. At the bottom are 'Add Field' and 'Cancel' buttons. The background shows the 'All processes' list with items like 'Basic (default)', 'Agile', 'Scrum', and 'CMMI'. The Windows taskbar is visible at the bottom.



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:

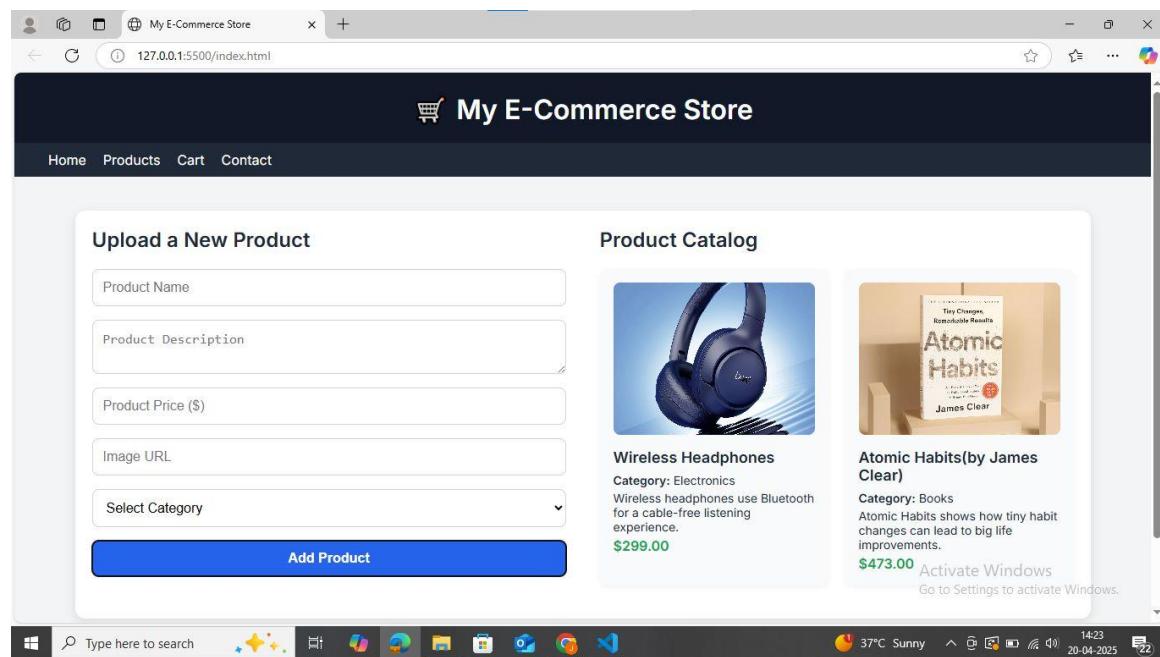
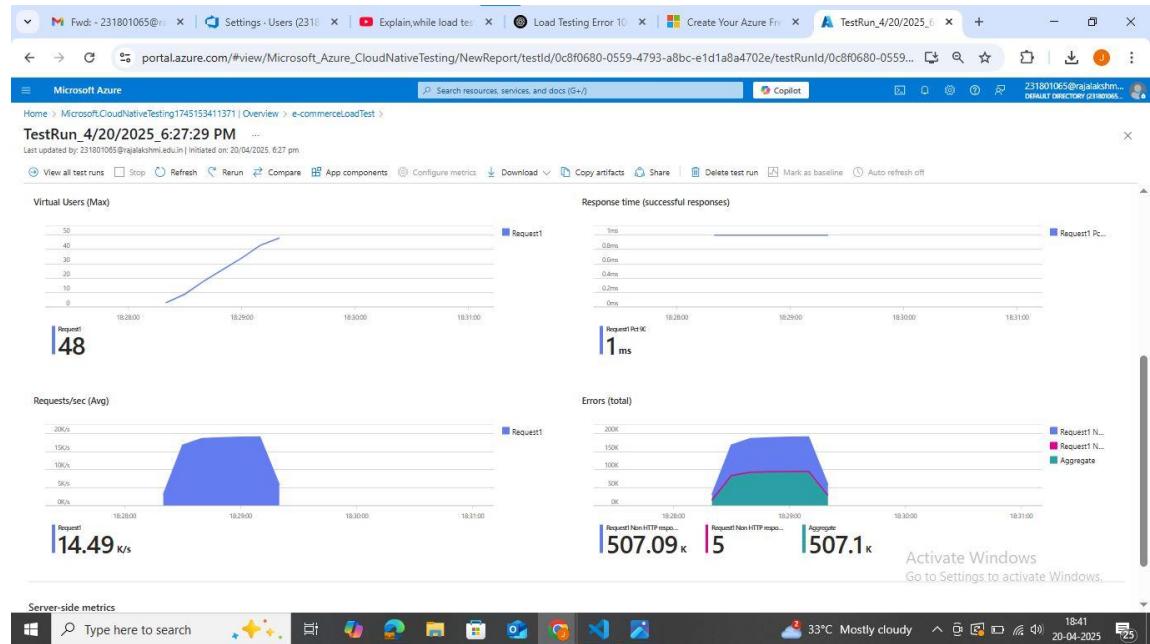
1. Sign in to Azure Portal
Go to <https://portal.azure.com> and log in.
2. Create the Resource
 - o Go to *Create a resource* → Search for “Azure Load Testing”.
 - o Select Azure Load Testing and click Create.
3. Fill in the Configuration Details
 - o *Subscription*: Choose your Azure subscription.
 - 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.
 - o 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:
 - o In your GitHub repository, create a new file called azure-pipelines.yml in the root directory.
 - o 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:

- o 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.

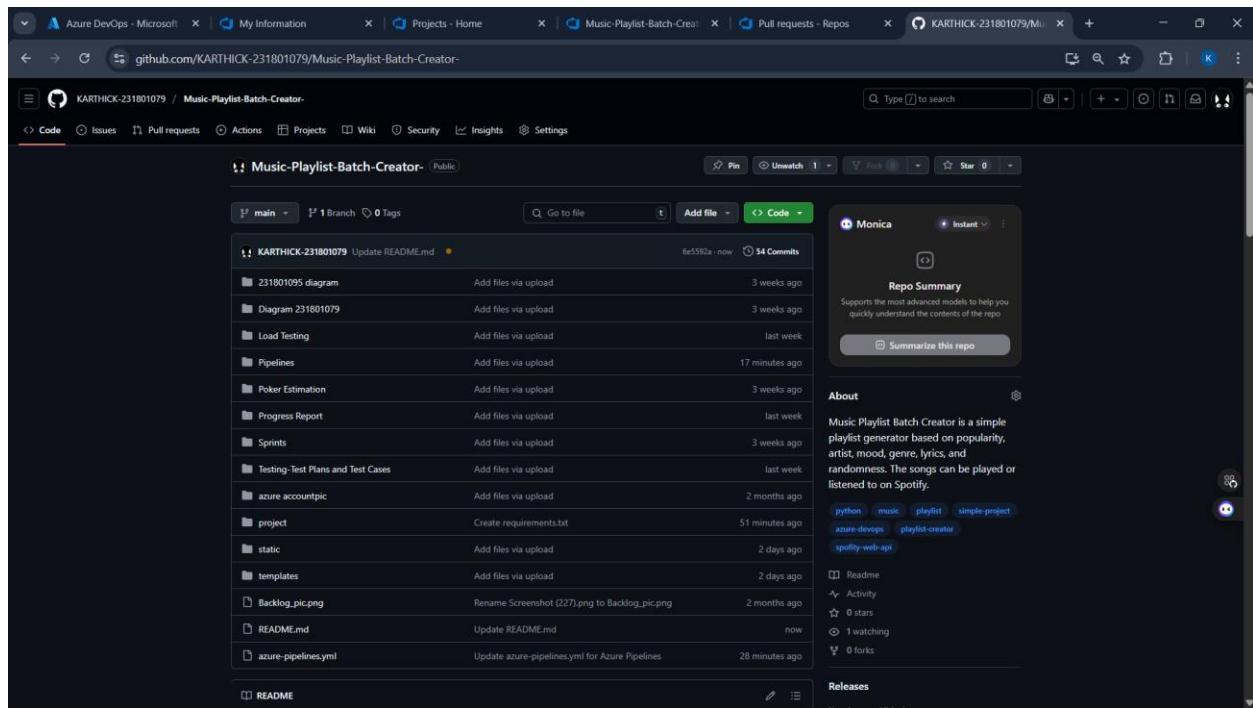
EXP NO: 10

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 E Commerce Product Uploader 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.