



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: Janani V R

Register No: 231801065

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:

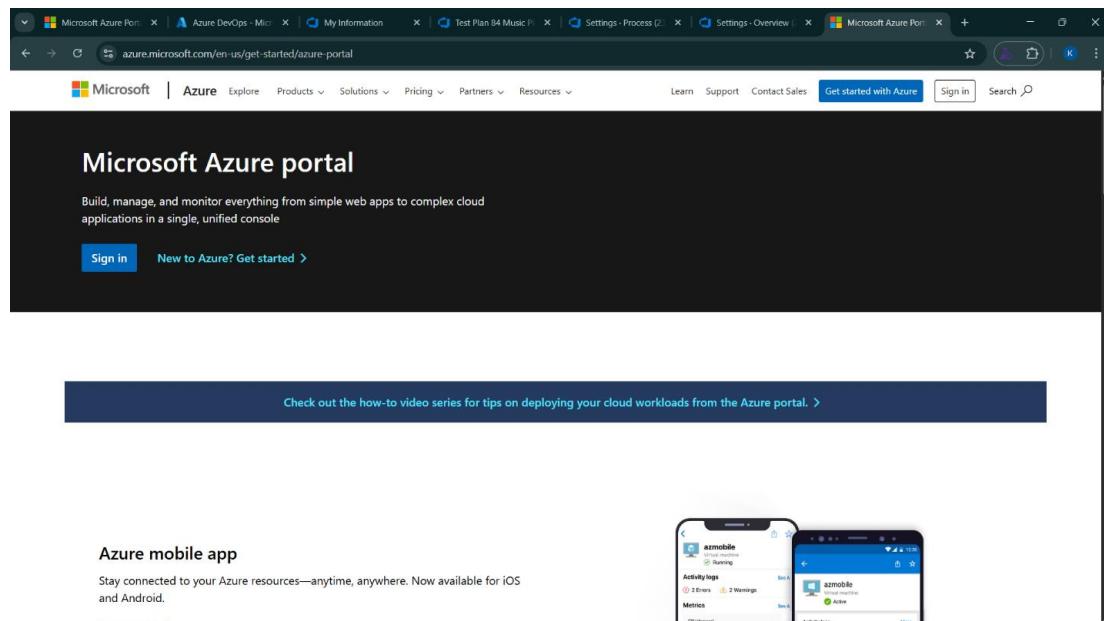
To set up and access the Azure DevOps environment by creating an organization through the Azure portal.

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 header, 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 are displayed under the "Services" tab, showing options like Azure Native New Relic Service, Managed DevOps Pools, Azure DevOps organizations, and Azure Native Dynatrace Service. Below the search results, there are sections for Marketplace, Documentation, and Microsoft Entra ID. The "Resources" section on the left shows the same recent resources as the first screenshot. The "Tools" section and "Useful links" section are also present.

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 with the URL portal.azure.com/#view/AzureTfExtension/OrganizationsTemplateBlade. The page displays the Azure DevOps landing page, which features a central illustration of a rocket launching from a stack of boxes, with several people working on a large green board labeled 'Azure DevOps'. A banner at the top states: "We've made it easier to manage Azure DevOps billing and subscriptions. You can [set up billing](#), change your subscription or pay for more users and resources within Azure DevOps. [Learn more](#)". Below the illustration, there's a section titled "Azure DevOps" with the subtext: "Plan smarter, collaborate better, and ship faster with a set of modern dev services". There are also links for "My Azure DevOps Organizations", "Get started using Azure DevOps", "Billing management for Azure DevOps", "Give feedback", and "Tell us about your experience with the Azure DevOps page".

The screenshot shows a Microsoft Edge browser window with the URL aex.dev.azure.com/signup/?acquisitionId=bef1dfa-718b-4dad-a0d2-93ea5a276ad2&campaign=o~msft~profile~service_attach&mkt=en-GB. The page displays the "Get started with Azure DevOps" sign-up form. It includes fields for email (akiladevi.r@rajalakshmi.edu.in) and a checkbox for accepting terms and conditions. The background features a cartoon illustration of a person walking a dog under a cloud. The Windows taskbar at the bottom shows various open applications like Inbox, SC Manager, Create, and Azure.

Result:

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

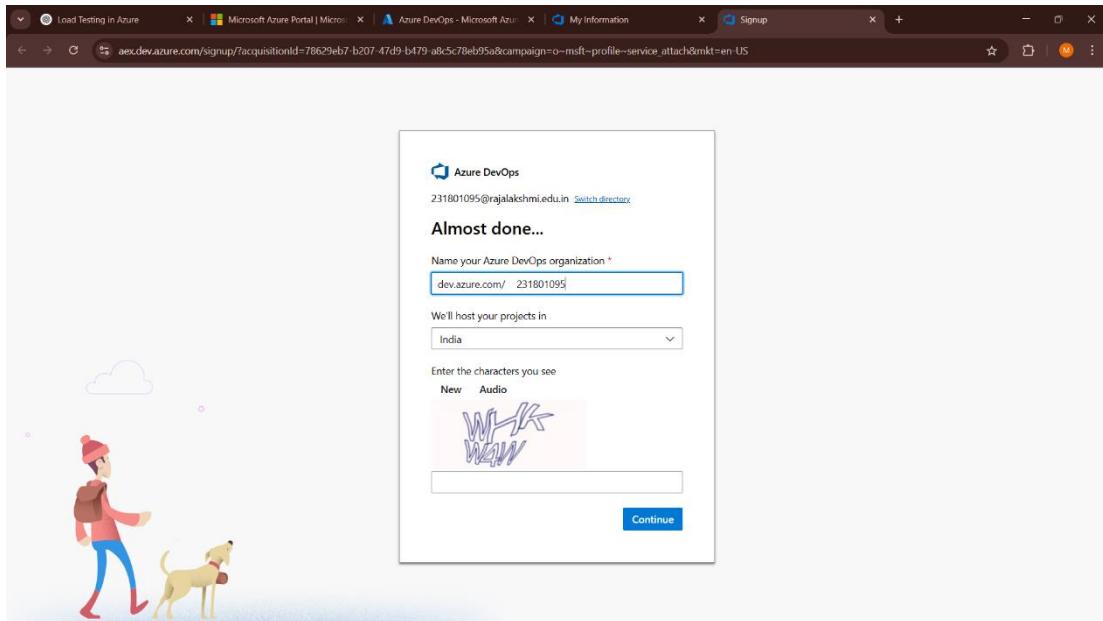
EXP NO: 2

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 interface. On the left, there is a sidebar with Krithika.M.A's profile picture (a purple circle with a white 'K'), name, email (krithika7604@gmail.com), location (India), and a link to edit the profile. Below this is a section for 'Visual Studio Dev Essentials' with a link to 'Use your benefits'. The main area is titled 'Azure DevOps Organizations' and shows 'dev.azure.com/krithika7604' (Owner) with a single project: 'E-commerce product uploader'. There is a 'Create new organization' button at the top right. To the right of the project, there are 'Actions' like 'Open in Visual Studio'. At the bottom of the page, there is a Windows taskbar showing the date (28-04-2025) and time (17:41).

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 content 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 is a 'Project stats' section with a message 'No stats are available at this moment. Setup a service to see project activity.' and a 'Members' section showing one member. The Windows taskbar at the bottom indicates the date (28-04-2025) and time (17:28).

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 'Work items - Boards' page in Azure DevOps. The left sidebar has 'E-commerce product u...' selected under 'Work items'. The main area shows a table of work items with columns: Epic, Assigned to, State, Area Path, and Tags. One row is highlighted for a User Story titled 'User Authentication and Authorisation' assigned to 'madhusha' with state 'New' and area path 'E-commerce product uploader'. Other rows include 'Product Approval and Moderation' assigned to 'Janani Vasu', 'Kritika.M.A.', 'Janani Rajan', and 'Mahalakshmi231801093'.

The screenshot shows the Microsoft sign-in page. The user 'Krithika.M.A.' is logged in, with their email 'krithika7604@gmail.com' and a link to 'My Microsoft account'. There are options to 'Switch directory' and 'Sign out'. Below the sign-in area, there are links for 'Project settings' and 'Boards'.

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 Boards Backlog view for the 'E-commerce product uploader' team. On the left, the navigation menu is visible with 'Backlogs' selected. The main area displays a backlog board with columns for Order, Work Item Type, Title, State, Effort, Business Area, and Tags. The backlog contains the following items:

| Order | Work Item Type | Title | State | Effort | Business Area | Tags |
|-------|----------------|---|-------|--------|---------------|------|
| 1 | Epic | User Authentication and Authorization | New | | Business | |
| | Feature | Login page for users | New | | Business | |
| | User Story | As a user, I want to log in using my email and password. | New | | Business | |
| | Task | Design the Login Page UI | New | | | |
| | Task | Add Validation for Input Fields. | New | | | |
| | Task | Handle Incorrect Login Attempts. | New | | | |
| | User Story | As a user, I want to reset my password if I forget it, so ... | New | | Business | |
| | User Story | As a user, I want to authenticate using my social media... | New | | Business | |
| | Feature | Login Page for Product Uploaders | New | | Business | |

1.Fill in Epics

The screenshot shows the 'New Epic' creation form in Azure DevOps. The form includes fields for Product Categorization and Tagging, Description, Planning, Deployment, and Development. The 'Description' field contains placeholder text: 'Click to add Description.' The 'Planning' section includes fields for Priority (set to 2), Risk, and Effort. The 'Deployment' section includes a note about tracking releases and a link to 'Releases'. The 'Development' section includes a note about linking to Azure Repos and a link to 'Create a branch'. The form also includes sections for 'Discussion' and 'Comments'.

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.

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.

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

| Order | Title | State | Assigned To | Remaining |
|-------|--|-------|------------------|-----------|
| 1 | As a user, I want to authenticate using my social media account. | New | Janani Rajan | |
| 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 access my account. | New | Krithika.M.A | |
| 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 system so that I can manage products. | New | 231801091@raj... | |

Sprint 1

| New | Active | Resolved | Closed |
|--|--------|----------|--------|
| 4. As a user, I want to reset my password if I forget it, so that I can regain access to my account. 58. Design the Login UI. 59. Add Validation for Input Fields. 60. Handle incorrect Login attempts. | | | |
| 5. As a user, I want to authenticate using my social media account. | | | |

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'. The board has columns for 'New', 'Active', 'Resolved', and 'Closed'. There are four tasks visible:

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

The top right of the board shows the sprint duration: 'March 3 - March 14' and '10 work days'. The bottom right of the screen shows the Windows taskbar with various pinned icons and the date/time: '28-04-2025 17:53'.

Sprint 3

The screenshot shows the Azure DevOps Taskboard for the 'E-commerce product uploader' project. The left sidebar navigation is identical to the previous screenshot. The main area displays the 'Taskboard' for 'Sprint 3'. The board has columns for 'New', 'Active', 'Resolved', and 'Closed'. There are three tasks visible:

- Task 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.
 - Status: New
 - Assignee: Unassigned
- Task 23: As an admin, I want to provide feedback on rejected products to the uploader, so they can make necessary changes.
 - Status: New
 - Assignee: Unassigned
- Task 25: As a product uploader, I want to receive notifications when my product is approved or rejected.
 - Status: Unassigned

The top right of the board shows the sprint duration: 'March 17 - March 28' and '10 work days'. The bottom right of the screen shows the Windows taskbar with various pinned icons and the date/time: '28-04-2025 17:54'.

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.

Mahalakshmi231801093 0 Comments Add Tag

| | | |
|--------------------------------------|-----------|--------------------------------------|
| <input checked="" type="radio"/> New | Area | E-commerce product uploader |
| <input type="radio"/> New | Iteration | E-commerce product uploader\Sprint 1 |

Description

• **Password Reset Request:** The user enters their email or username to request a password reset.
• **Verification Process:** A verification link or code is sent to confirm the user's identity.
• **Password Update:** The user sets a new password after identity verification.
• **Confirmation and Login:** The user is notified and can log in with the new password.

Planning

Story Points

Priority
2

Risk

ance Criteria

1. **Password Reset Request:**
User can request a reset by entering their email/username.

2. **Verification Link/Code:**
A verification link/code is sent and expires within a set time.(eg: within 15 mins)

3. **New Password Set:**

Classification

Value area
Business

Result:

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

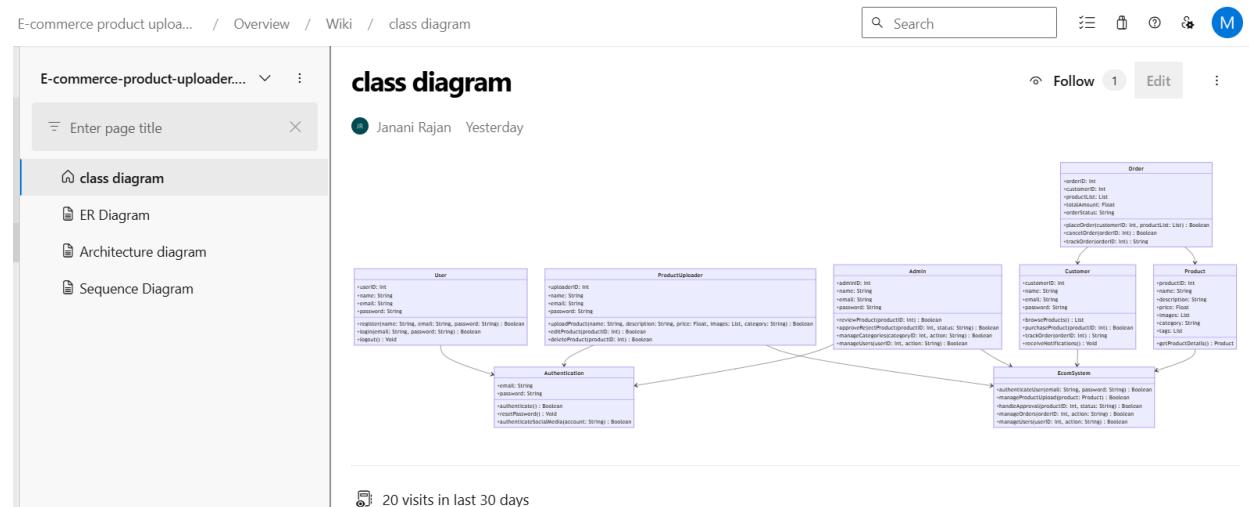
EXP NO: 6

DESIGNING CLASS AND SEQUENCE DIAGRAMS FOR PROJECT ARCHITECTURE

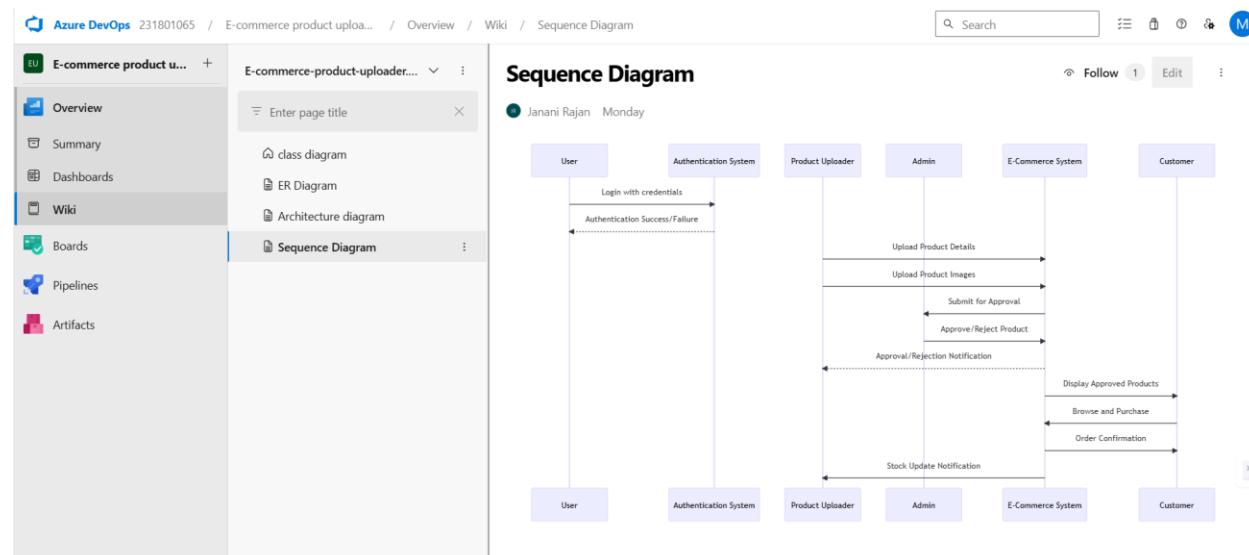
Aim:

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.

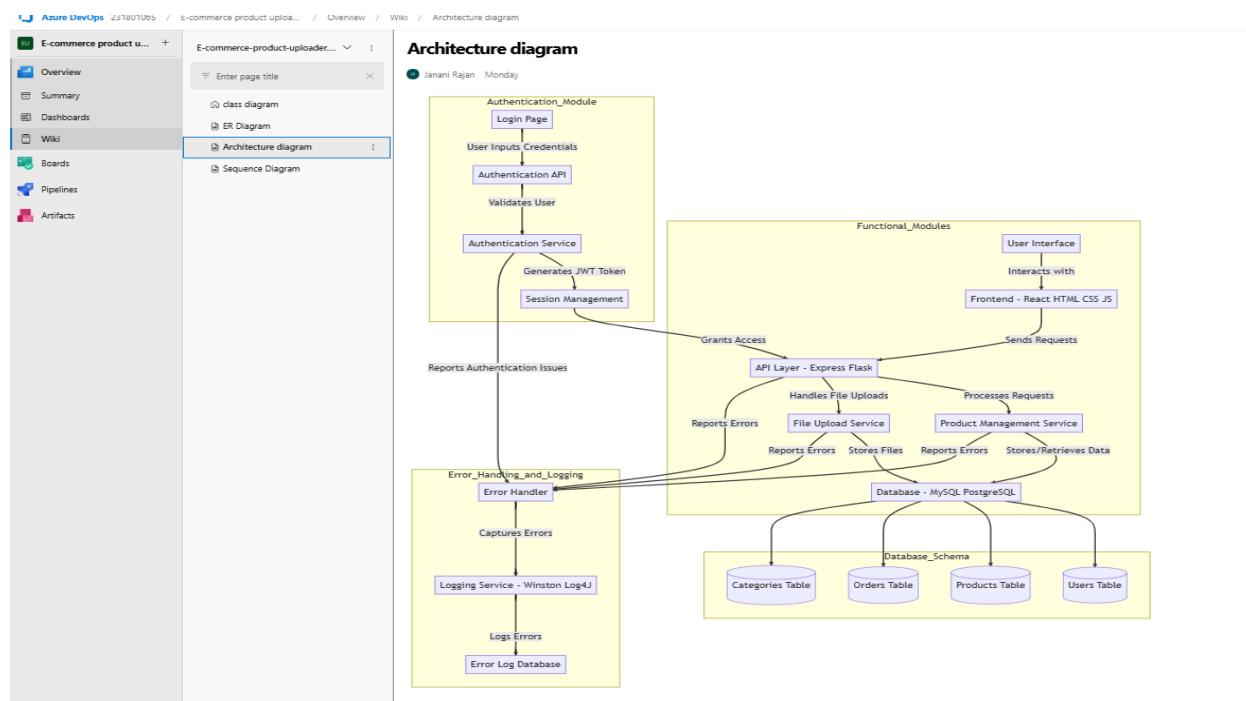
EXP NO: 7

DESIGNING ARCHITECTURAL AND ER DIAGRAMS FOR PROJECT STRUCTURE

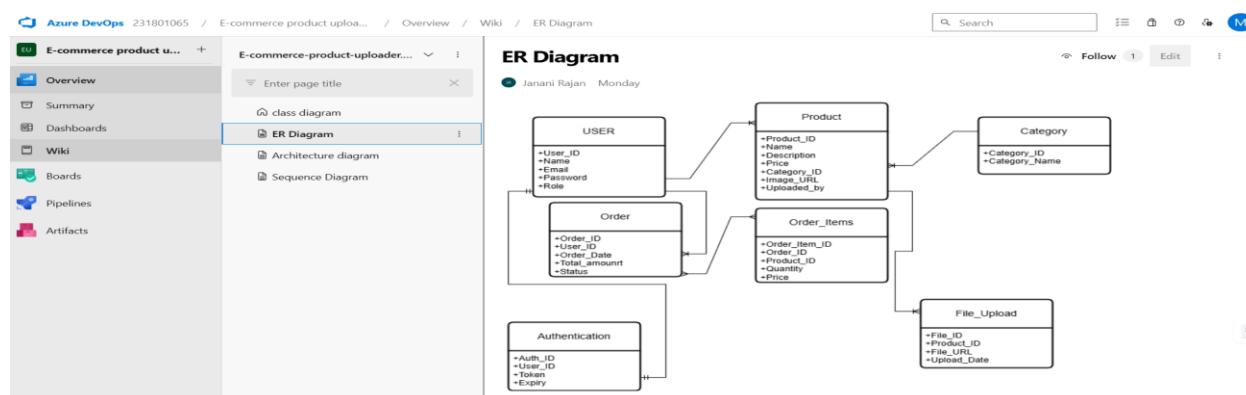
Aim:

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:

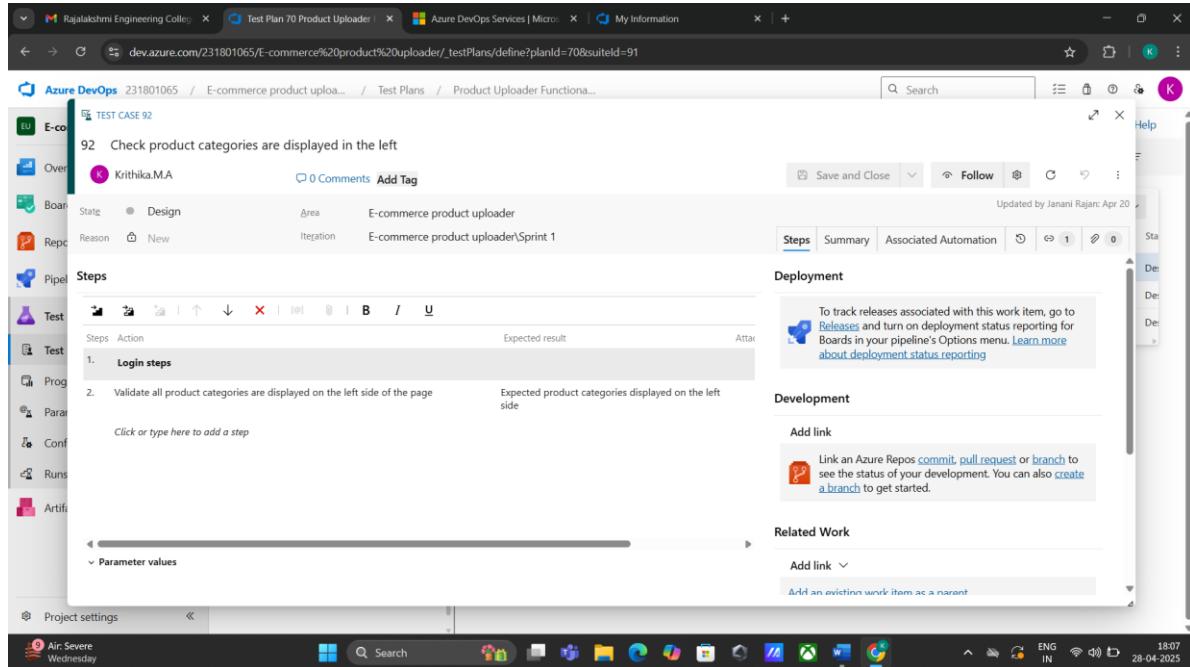
- Login with valid credentials.
- Click on "Generate Playlist".
- Select categories.
- Click "Generate Playlist".

- Expected Results:

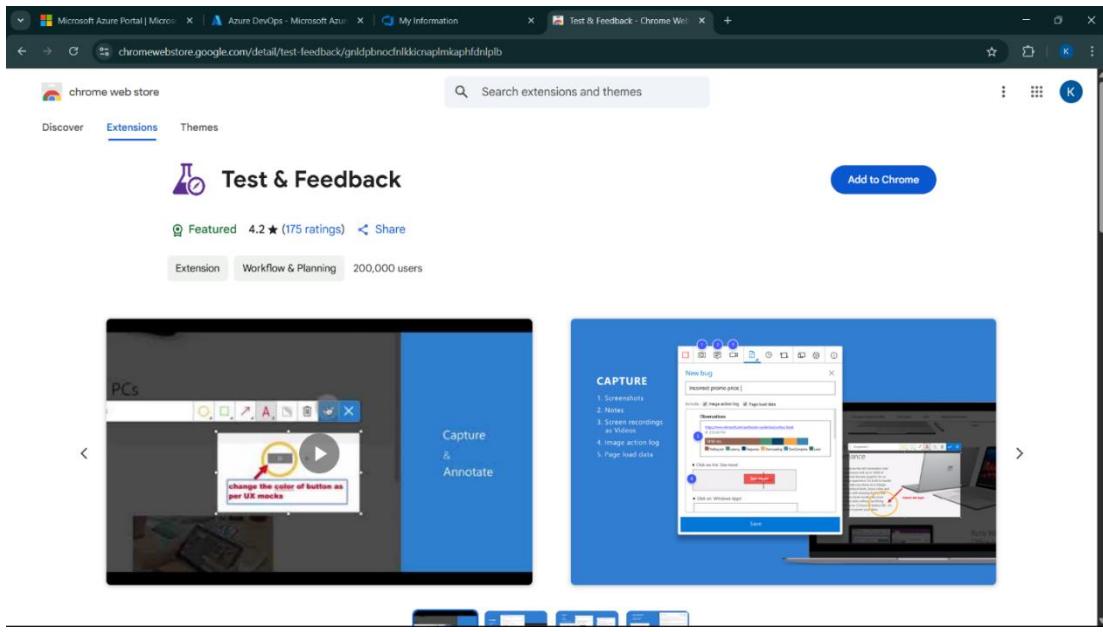
- Error message: "Please select at least one valid category" or "No recommendations found for the selected filters".

- Type: Error Path

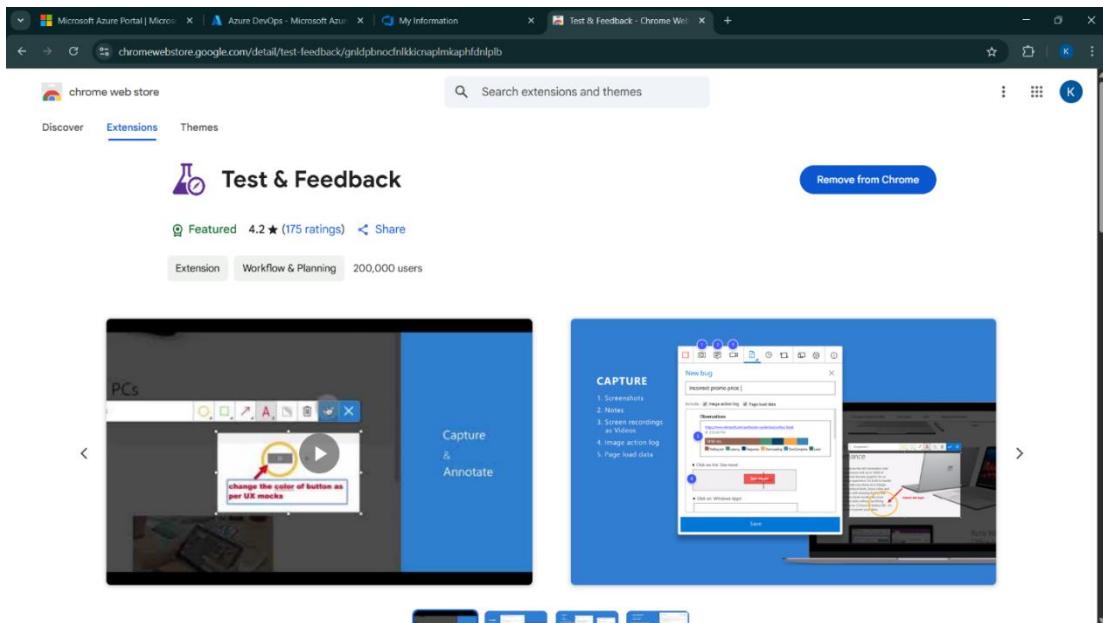
Test Cases



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'. Under 'Test Plans', 'Test Suites' is expanded, showing 'ProjectUploader Functionality and...' which contains 'ProjectTestSuite (3)'. Below this are 'New query-based suite (10)', '7: As a product uploader, I ...', '25: As a product uploader, I ...', 'Product details Validation (3)', and 'Image Validation (3)'. The main area displays 'ProjectTestSuite (ID: 91)' with tabs for 'Define', 'Execute' (selected), and 'Chart'. The 'Test Cases (3 items)' section lists:

| Index | Test Case ID | Author | Delegated To |
|-------|--------------|--------------|--------------|
| 1 | 92 | Krithika.M.A | Delegated |
| 2 | 93 | Janani Rajan | Delegated |
| 3 | 99 | Janani Rajan | Delegated |

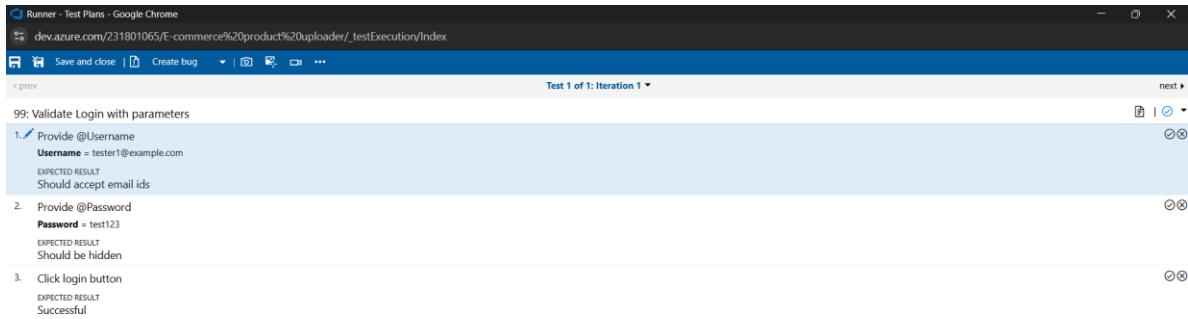
A modal window titled 'Extensions' is open, showing 'Full access' and a list of extensions: 'McAfee® WebAdvisor' and 'Test & Feedback'. A 'Manage extensions' button is also present.

5. Running the test cases

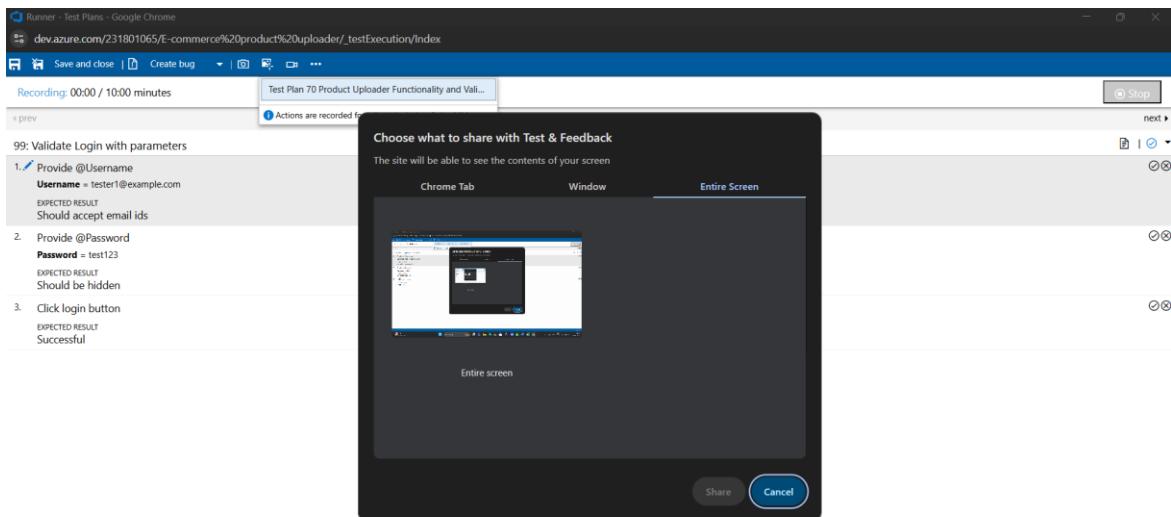
The screenshot shows the same Azure DevOps Test Plan interface as the previous one, but with the 'Execute' tab selected. The 'Test Points (6 items)' section lists:

| 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 |
| Select Specific product Category | Passed | 2 | 93 |
| Select Specific product Category | Passed | 2 | 93 |
| Validate Login with parameters | Failed | 3 | 99 |
| Validate Login with parameters | Active | 3 | 99 |

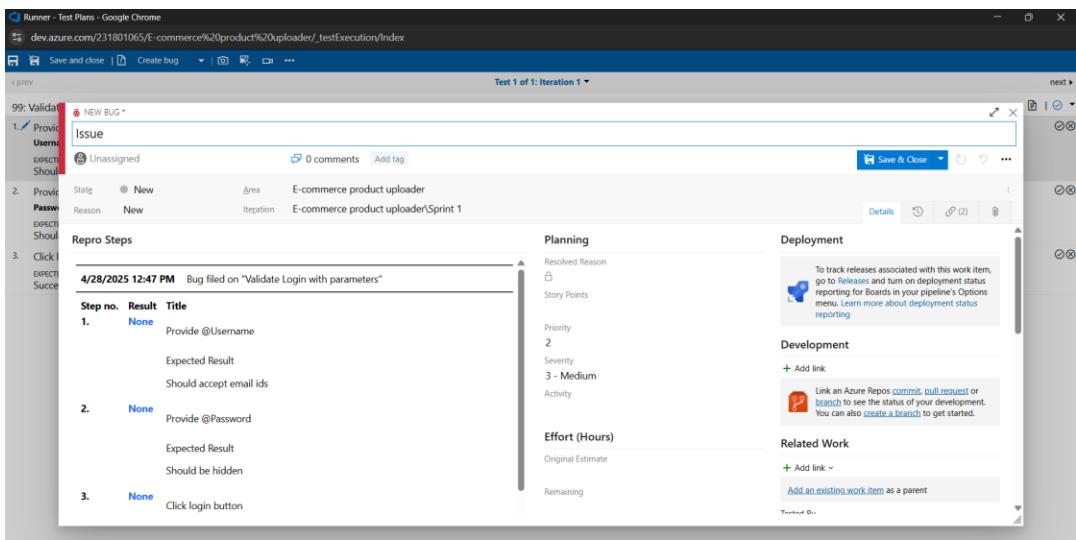
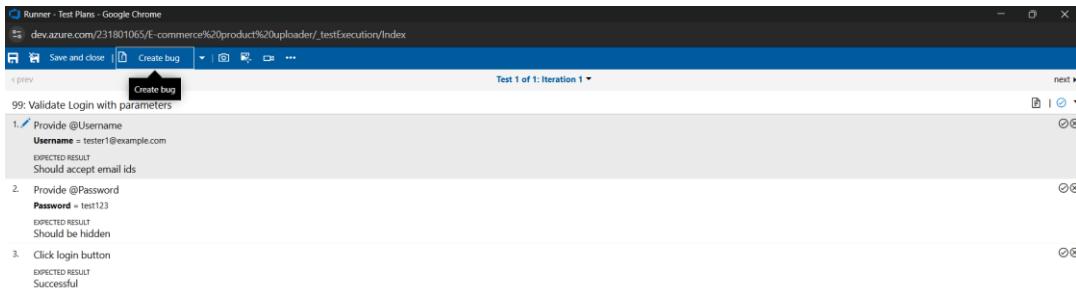
A context menu is open over the 'Validate Login with parameters' row, with options: 'View execution history', 'Mark Outcome', 'Run', 'Reset test to active', 'Edit test case', and 'Assign tester'. Other options like 'Run for web application', 'Run for desktop application', and 'Run with options' are also visible.



6. Recording the test case



7. Creating the bug



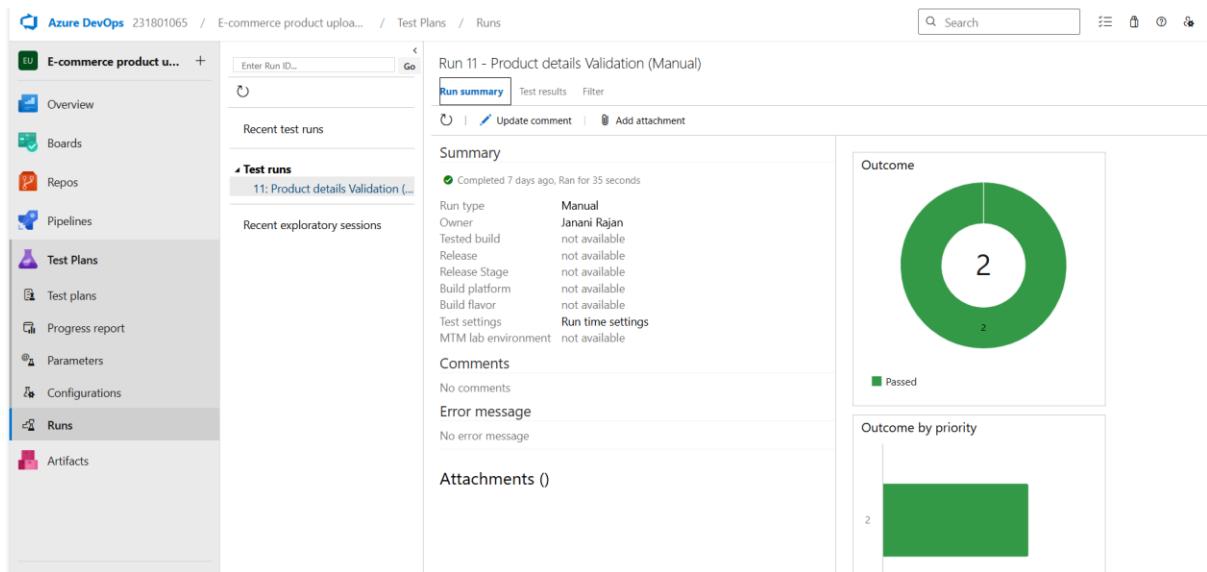
The screenshot shows the Azure DevOps interface for a project named "E-commerce product uploader". The left sidebar is open, showing options like Overview, Boards, Repos, Pipelines, Test Plans, Test plans, Progress report, Parameters, Configurations, and Runs. The "Runs" option is selected. The main content area displays a test run titled "Run 7 - ProjectTestSuite (Manual) / Check product categories are displayed in the left". The "Summary" section includes details such as Run by (Janani Rajan), Tested build (not available), Test Plan (Product Uploader Functionality and Validation Test plan), Priority (2), Test suite (ProjectTestSuite), Test Case (Check product categories are displayed in the left), and Configuration (Chrome). The "Analysis" section shows the owner (Janani Rajan), failure type (None), resolution (None), and comment (not available). Below the summary are sections for "Attachments ()" and "Linked Items (0)". A detailed view of a test step is shown under "Test steps": "1. Login steps" and "1.1 Go to login page". The expected result is "Login page is displayed". The status is "Test passed" with a start time of 4/20/2025 5:48:26 AM and a duration of 0:01:08.835.

8.Test case results

The screenshot shows the Azure DevOps interface for the same project. The left sidebar is open, showing the "Test Plans" section selected. The main content area displays a test case results summary for "ProjectTestSuite (ID: 91)". The title of the test case is "Check product categories are displayed in the left". The "Test Case Results" table shows six items, all of which have passed. The columns include Outcome,TimeStamp, Configuration, Run by, Tester, and Test. The outcomes are: Passed (Apr 20, Firefox, Janani Rajan, Janani Rajan, Product Uploader Functionality and Validation Test plan); Passed (Apr 20, Chrome, Janani Rajan, Janani Rajan, Product Uploader Functionality and Validation Test plan); Passed (Apr 20, Chrome, Janani Rajan, Janani Rajan, Product Uploader Functionality and Validation Test plan); Passed (Apr 20, Firefox, Janani Rajan, Janani Rajan, Product Uploader Functionality and Validation Test plan); and Passed (Apr 20, Chrome, Janani Rajan, Janani Rajan, Product Uploader Functionality and Validation Test plan).

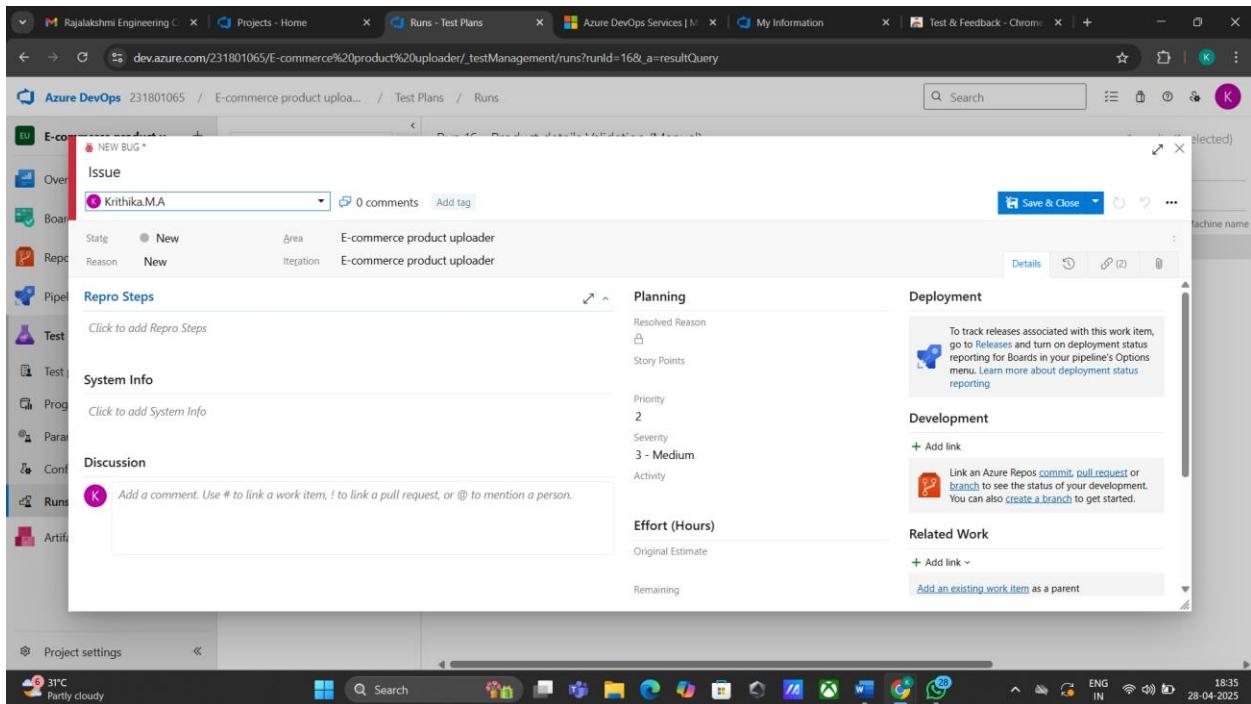
| Outcome | TimeStamp | Configuration | Run by | Tester | Test |
|---------|-----------|---------------|--------------|--------------|---|
| Passed | Apr 20 | Firefox | Janani Rajan | Janani Rajan | Product Uploader Functionality and Validation Test plan |
| Passed | Apr 20 | Chrome | Janani Rajan | Janani Rajan | Product Uploader Functionality and Validation Test plan |
| Passed | Apr 20 | Chrome | Janani Rajan | Janani Rajan | Product Uploader Functionality and Validation Test plan |
| Passed | Apr 20 | Firefox | Janani Rajan | Janani Rajan | Product Uploader Functionality and Validation Test plan |
| Passed | Apr 20 | Chrome | Janani Rajan | Janani Rajan | Product Uploader Functionality and Validation Test plan |

9. Test report summary



The screenshot shows the Azure DevOps interface for a test run. On the left, the navigation bar includes 'Overview', 'Boards', 'Repos', 'Pipelines', 'Test Plans' (selected), 'Test plans', 'Progress report', 'Parameters', 'Configurations', 'Runs' (selected), and 'Artifacts'. The main content area displays 'Run 11 - Product details Validation (Manual)'. It includes tabs for 'Run summary' (selected), 'Test results', and 'Filter'. Below the tabs are sections for 'Recent test runs', 'Summary' (showing completion status and run type), 'Comments' (no comments), 'Error message' (no error message), and 'Attachments ()'. To the right are two donut charts: 'Outcome' (2 Passed) and 'Outcome by priority' (2 Medium priority). A search bar at the top right contains the text 'Search'.

- Assigning bug to the developer and changing state

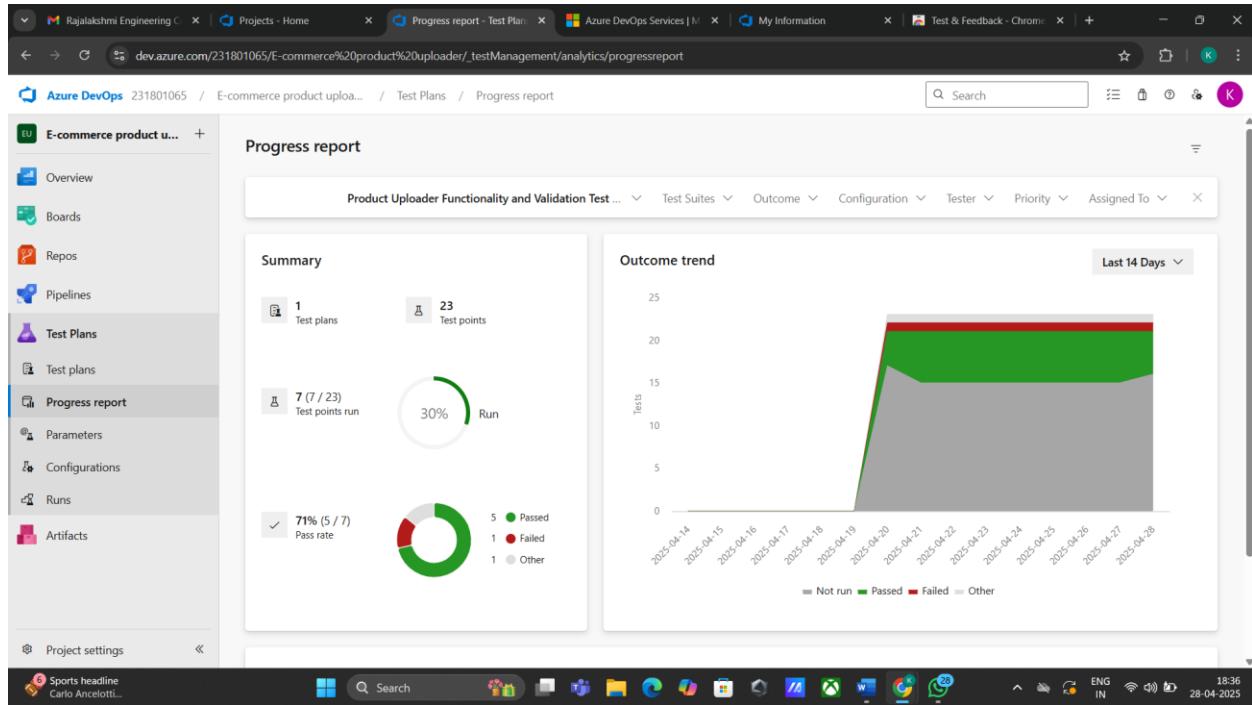


The screenshot shows the Azure DevOps interface for a work item. The left sidebar lists 'Overview', 'Boards', 'Repos', 'Pipelines', 'Test' (selected), 'Test plans', 'Progress report', 'Parameters', 'Configurations', 'Runs', and 'Artifacts'. The main content area shows a 'NEW BUG' work item for 'E-commerce product uploader'. The work item details include:

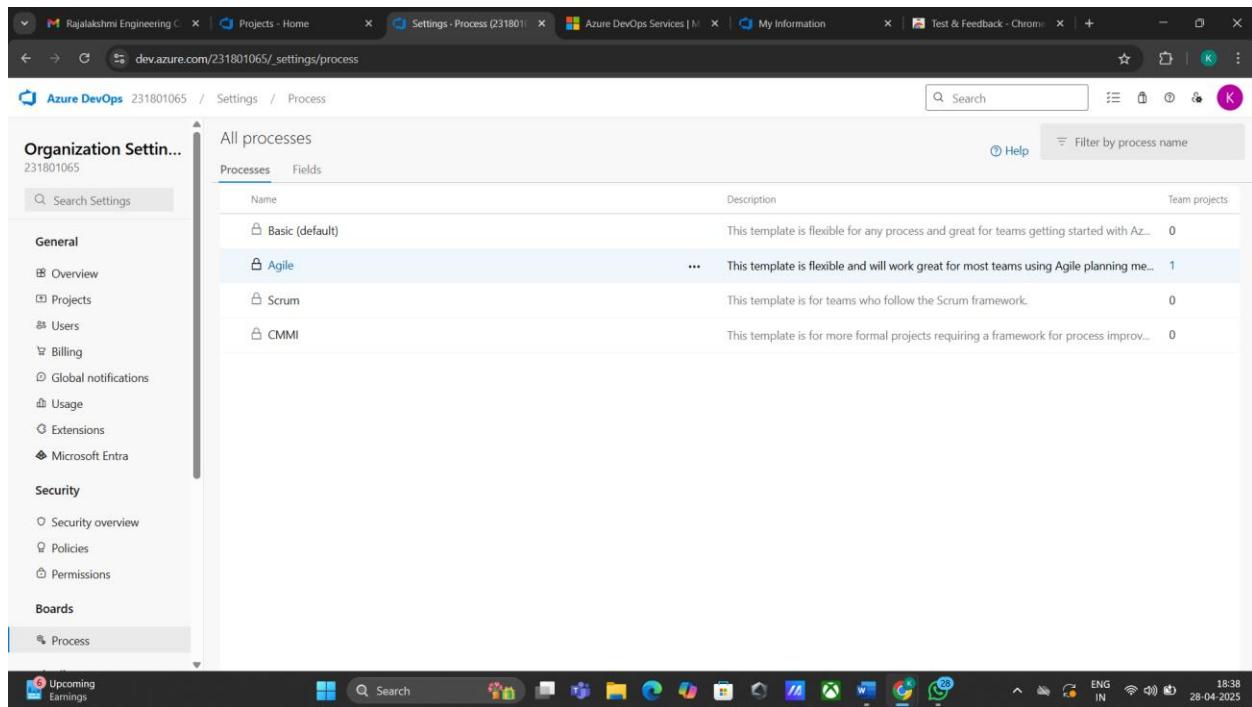
- Issue:** Assigned to 'Krithika.MA', 0 comments, Add tag.
- Planning:** State: New, Area: E-commerce product uploader, Reason: New, Iteration: E-commerce product uploader.
- Repro Steps:** Click to add Repro Steps.
- System Info:** Click to add System Info.
- Discussion:** Add a comment. Use # to link a work item, ! to link a pull request, or @ to mention a person.
- Deployment:** To track releases associated with this work item, go to Releases and turn on deployment status reporting for Boards in your pipeline's Options menu. Learn more about deployment status reporting.
- Development:** Add link, Link an Azure Repos commit, pull request or branch to see the status of your development. You can also create a branch to get started.
- Related Work:** Add link, Add an existing work item as a parent.
- Effort (Hours):** Original Estimate, Remaining.

A status bar at the bottom shows the date '28-04-2025' and time '18:35'.

10. Progress report



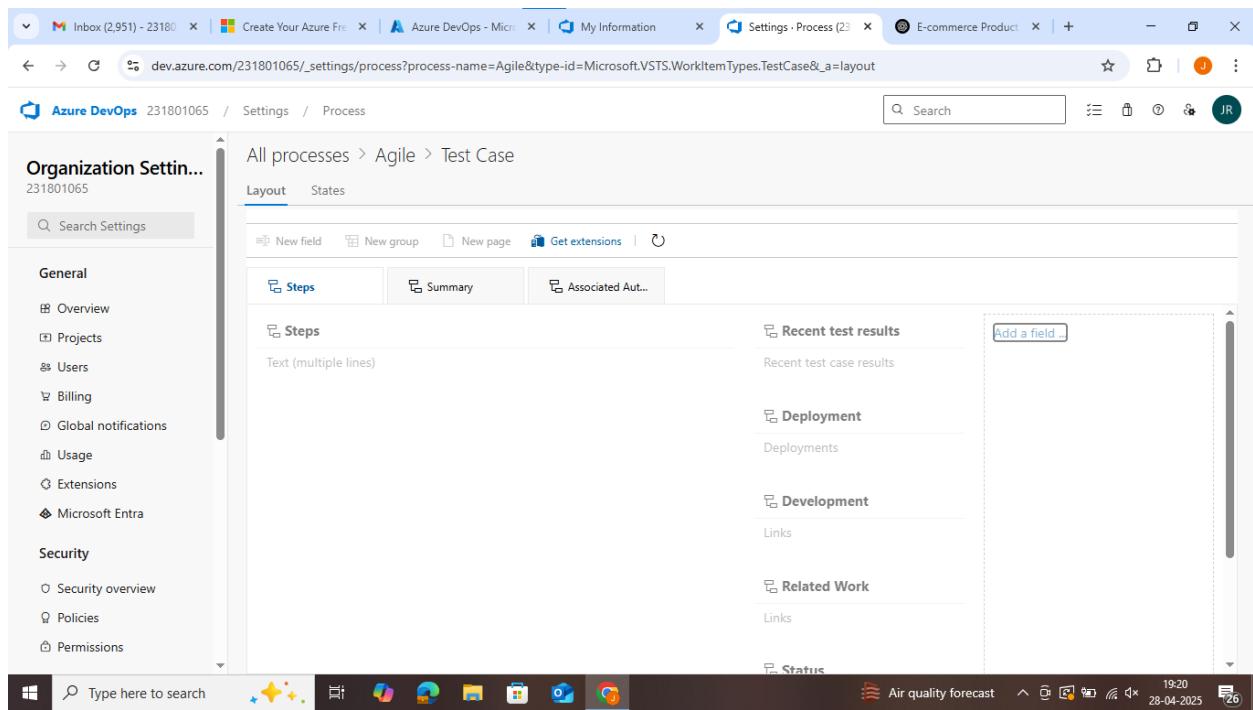
11. Changing the test template



The screenshot shows the 'Settings - Process' page in Azure DevOps. A modal window titled 'Create inherited process from Agile' is open, prompting the user to 'Create a new inherited process to enable customizations.' It includes fields for 'Process name (required)' and 'Description'. The background shows a list of available processes: Basic (default), Agile, Scrum, and CMMI. The 'Agile' process is selected. On the left, the 'Organization Settings' sidebar is visible, and at the bottom, the Windows taskbar.

12. View the new test case template

The screenshot shows the 'Settings - Process' page in Azure DevOps, specifically the 'Add a field to Test Case' dialog. The dialog has three tabs: 'Definition' (selected), 'Options', and 'Layout'. Under 'Definition', the user is creating a new field named 'e-commerce' of type 'Text (single line)'. The 'Layout' tab shows a preview area where the new field can be added. The background shows the 'All processes' list and the 'General' settings sidebar.



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:

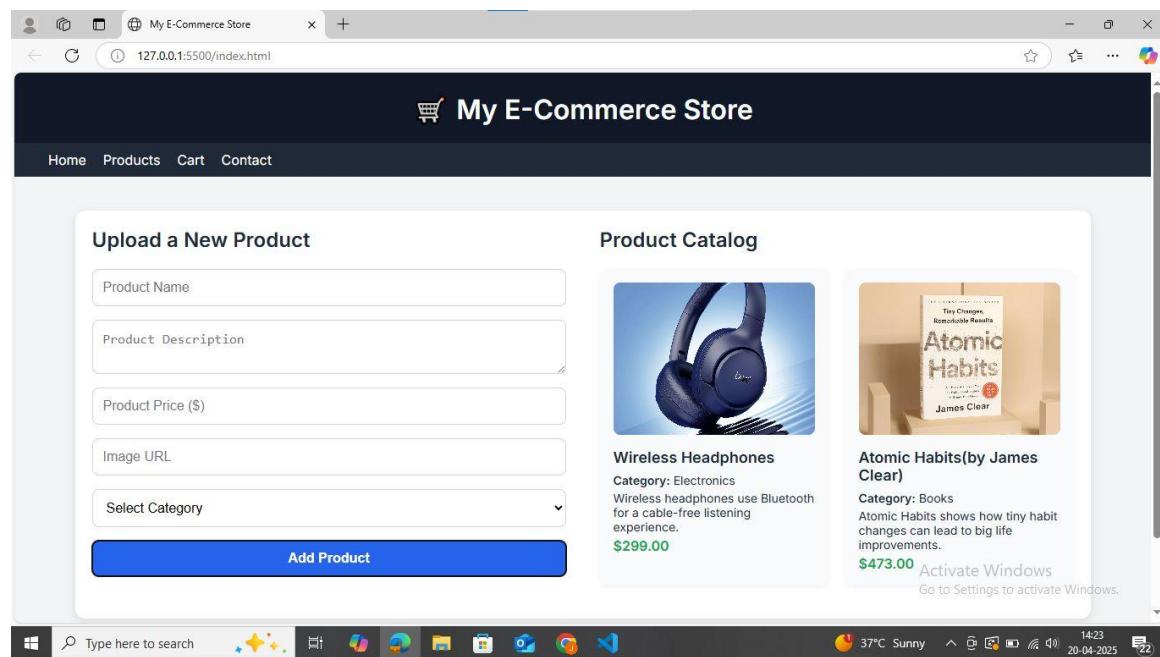
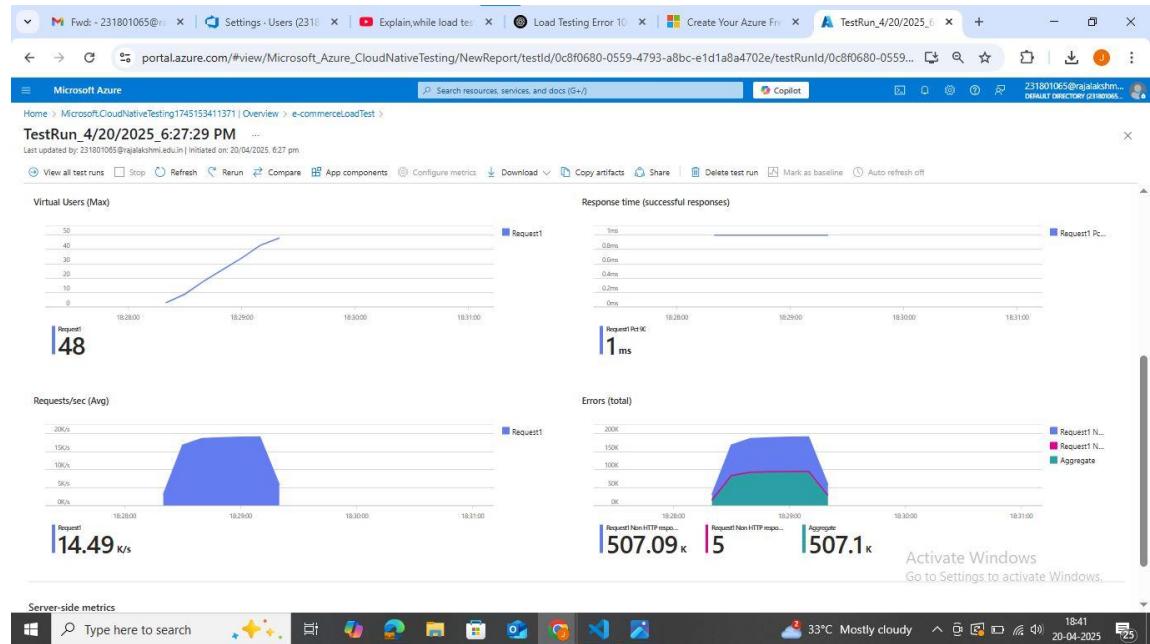
1. Sign in to Azure Portal
Go to <https://portal.azure.com> and log in.
2. Create the Resource
 - o Go to *Create a resource* → Search for “Azure Load Testing”.
 - o Select Azure Load Testing and click Create.
3. Fill in the Configuration Details
 - o *Subscription*: Choose your Azure subscription.
 - 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 is retained as one of three recent runs. It shows the repository and version (Music Playlist Batch Creator@main), the start time (Just now), and duration (24s). The "Jobs" section lists a single job that has completed successfully in 6s. The left sidebar provides navigation links for Overview, Boards, Repos, Pipelines, Environments, Releases, Library, Task groups, Deployment groups, Test Plans, and Artifacts. The top navigation bar includes links for My Information, Settings, Pipelines, Policies, and other project settings.

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.

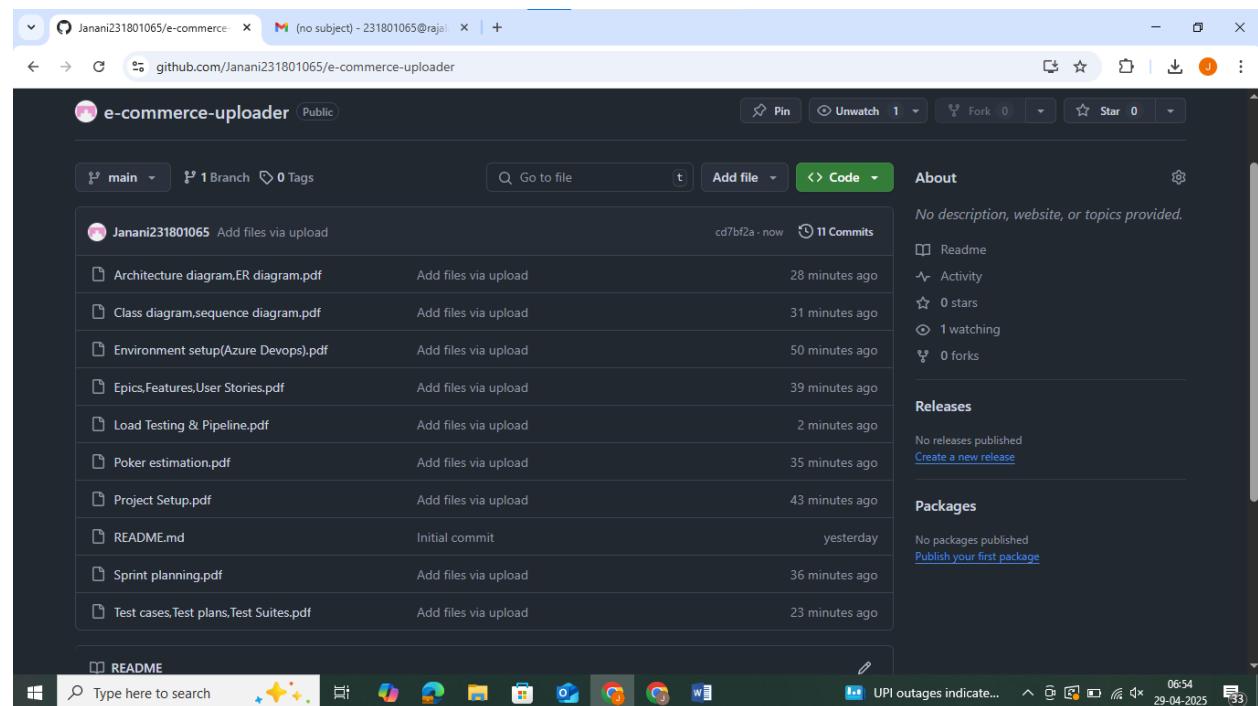
EXP NO: 10

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.