

EXP NO: 1

AZURE DEVOPS ENVIRONMENT SETUP

Aim:

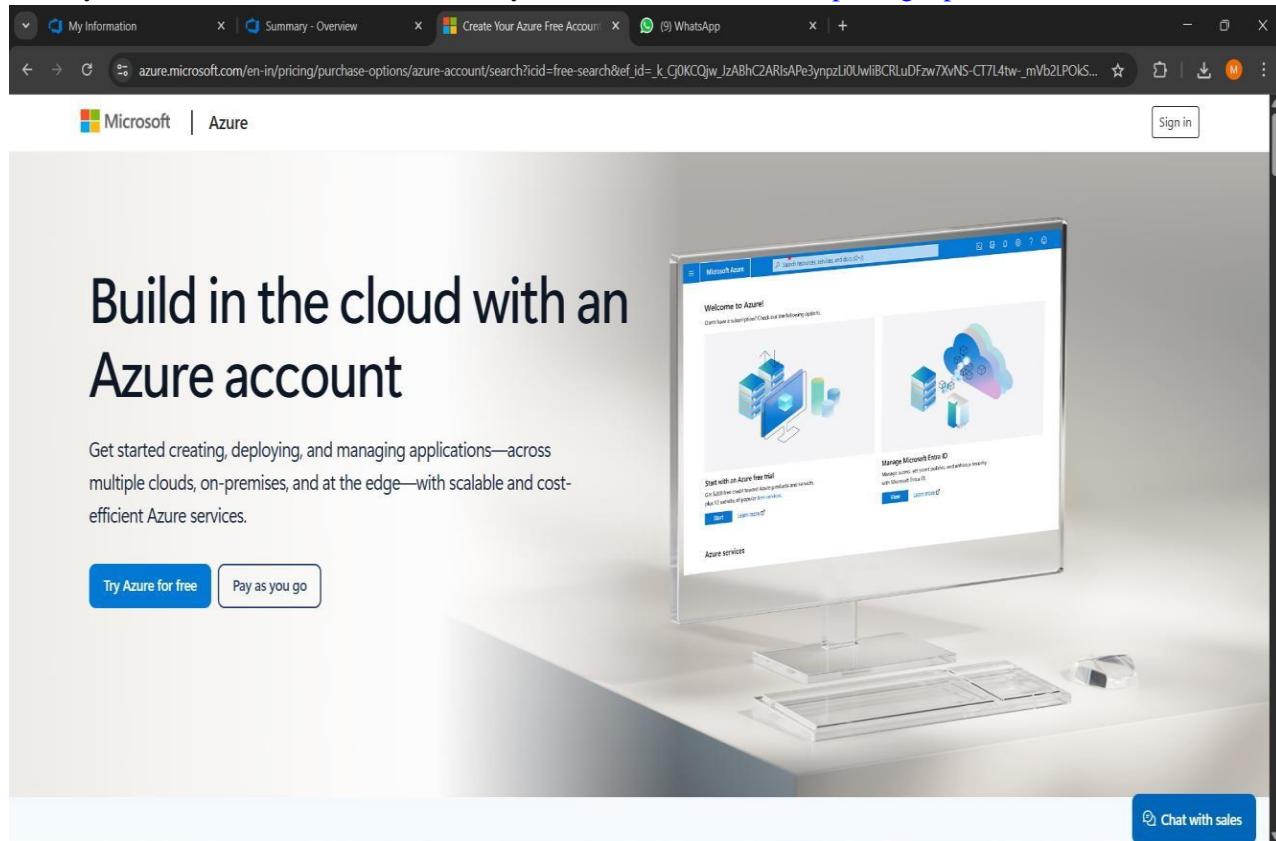
To set up and access the Azure DevOps environment by creating an organization through the Azure portal.

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, there's a section for 'Azure services' with icons for creating a resource, Azure Load Testing, Free services, Storage accounts, Storage browser, Azure DevOps organizations, Dashboard hub, Subscriptions, Virtual machines, and More services. Under 'Resources', there's a table showing a single item: 'Azure for Students' (Subscription type) last viewed 3 months ago. The 'Navigate' section includes links for Subscriptions, Resource groups, All resources, and Dashboard. The 'Tools' section features Microsoft Learn, Azure Monitor, Microsoft Defender for Cloud, and Cost Management.

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

This screenshot is similar to the previous one but with a search bar at the top containing the text 'azure dev'. The search results are displayed on the right side under 'Services' and 'Marketplace'. The 'Services' section shows 'Azure DevOps organizations' as the top result. The 'Marketplace' section shows 'Project', 'Dev center', 'Build Agents for Azure DevOps', and 'Azure DevOps Auditing'. The rest of the page layout is identical to the first screenshot.

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

Microsoft Azure Search resources, services, and docs (G+) Copilot Home > Azure DevOps ... 231801105@rajalakshmi... DEFAULT DIRECTORY

We've made it easier to manage Azure DevOps billing and subscriptions. You can [set up billing](#), change your subscription or pay for more users and resources within Azure DevOps. [Learn more](#)

Azure DevOps

Plan smarter, collaborate better, and ship faster with a set of modern dev services

[My Azure DevOps Organizations](#)

[Get started using Azure DevOps](#)
Billing management for Azure DevOps

[Give feedback](#)

[Tell us about your experience with the Azure DevOps page](#)



Result:

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

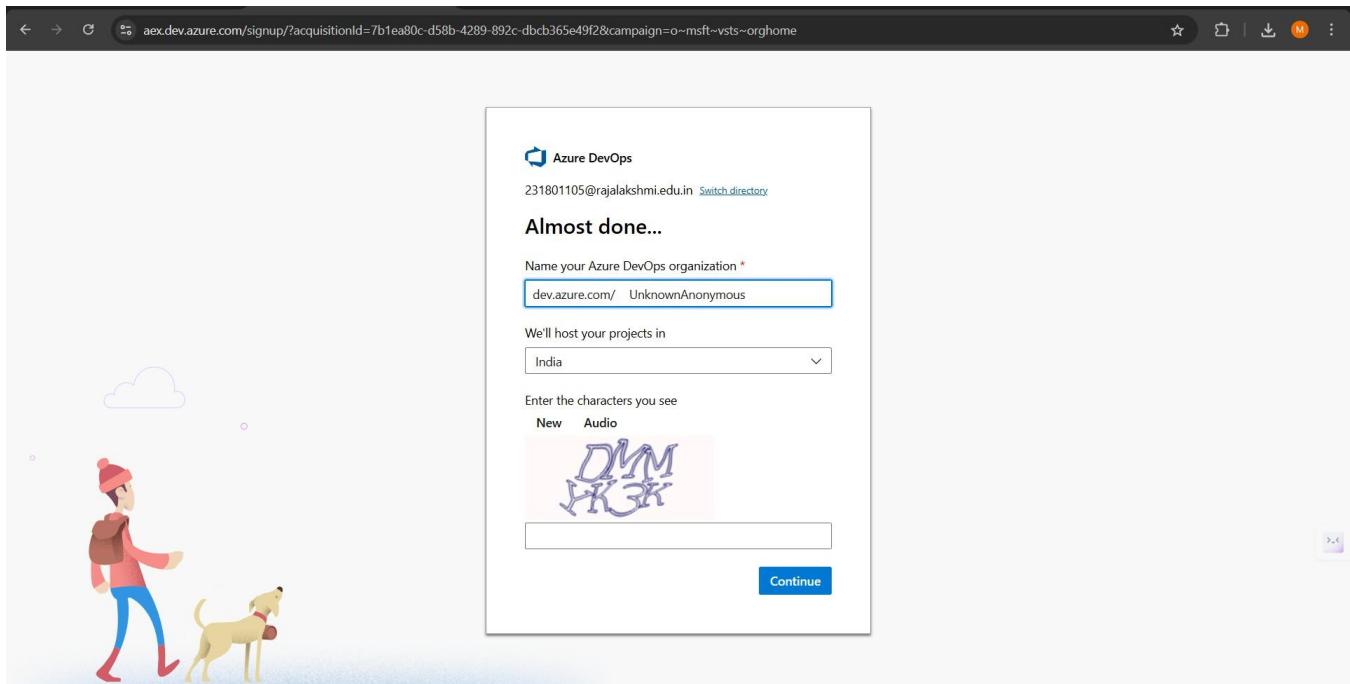
EXP NO: 2

AZURE DEVOPS PROJECT SETUP AND USER STORY MANAGEMENT

Aim:

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

1. Create An Azure Account



2. Create the First Project in Your Organization

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

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

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

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

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

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

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

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 [?](#)

[Cancel](#) [Create](#)

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

The screenshot shows the Azure DevOps interface for the 'UnknownAnonymous' organization. On the left, there's a sidebar with user profiles: 'UnknownAnonymous' (selected), 'FaisalDevOps1', and 'Anonymous1'. Below that are links for 'New organization' and 'Organization settings'. The main area displays the 'UnknownAnonymous' project, with tabs for 'Projects', 'My work items', and 'My pull requests'. A search bar and a 'Filter projects' button are at the top right. The 'E-Commerce' project is highlighted with a large card. At the bottom right of the main area, there's a small icon with a person and a gear.

4. Project dashboard

This screenshot provides a detailed view of the 'E-Commerce' project dashboard. The left sidebar includes links for Overview, Summary (selected), Dashboards, Wiki, Boards, Repos, Pipelines, Test Plans, and Artifacts. The main content area has a title 'E-Commerce' with a 'Private' and 'Invite' button. It features sections for 'About this project' (with a 'Help others to get on board!' callout) and 'Project stats' (over the last 7 days). The 'Project stats' section shows 1 work item created, 2 work items completed, 0 pull requests opened, and 6 commits by 3 authors. Below this is a 'Pipelines' section with a 0% completion circle and a 'Builds succeeded' link. The final section is 'Members' with 5 users listed, each represented by a colored profile icon.

https://dev.azure.com/UnknownAnonymous/E-Commerce/_workitems

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.

Order	Work Item Type	Title	State	Story...	Value Area	Iteration Path	Tags
1	User Story	> need a sign up page	● Active		Business	E-Commerce\Sprint 1	
2	User Story	> User Dashboard UI	● New		Business	E-Commerce\Sprint 1	
3	User Story	> Deploy the Application	● New	3	Business	E-Commerce\Sprint 1	
4	User Story	> Product Listing UI	● New		Business	E-Commerce\Sprint 1	
5	User Story	> Upload Product Information	● New		Business	E-Commerce\Sprint 1	
6	User Story	> User Login System	● New		Business	E-Commerce\Sprint 1	
7	User Story	> Write Unit Tests for Core Features	● New		Business	E-Commerce\Sprint 1	
8	User Story	> Edit Product Information	● New		Business	E-Commerce\Sprint 1	
9	User Story	> Delete Product Information	● New		Business	E-Commerce\Sprint 1	
10	User Story	> Upload Product Information	● New		Business	E-Commerce\Sprint 1	
11	User Story	> User Registration System	● New		Business	E-Commerce\Sprint 1	
12	User Story	> User Registration	● New		Business	E-Commerce\Sprint 1	
13	User Story	> User Login	● New		Business	E-Commerce\Sprint 1	
14	User Story	> File Upload Handling	● New		Business	E-Commerce\Sprint 1	

Order	Work Item Type	Title	State	Story...	Value Area	Iteration Path
1	User Story	> need a sign up page	● Active		Business	E-Commerce\Sprint 1
2	User Story	> User Dashboard UI	● New		Business	E-Commerce\Sprint 1
3	User Story	> Deploy the Application	● New	3	Business	E-Commerce\Sprint 1
4	User Story	> Product Listing UI	● New		Business	E-Commerce\Sprint 1
5	User Story	> Upload Product Information	● New		Business	E-Commerce\Sprint 1
6	User Story	> User Login System	● New		Business	E-Commerce\Sprint 1
7	User Story	> Write Unit Tests for Core Features	● New		Business	E-Commerce\Sprint 1
8	User Story	> Edit Product Information	● New		Business	E-Commerce\Sprint 1
9	User Story	> Delete Product Information	● New		Business	E-Commerce\Sprint 1
10	User Story	> Upload Product Information	● New		Business	E-Commerce\Sprint 1
11	User Story	> User Registration System	● New		Business	E-Commerce\Sprint 1
12	User Story	> User Registration	● New		Business	E-Commerce\Sprint 1
13	User Story	> User Login	● New		Business	E-Commerce\Sprint 1
14	User Story	> File Upload Handling	● New		Business	E-Commerce\Sprint 1

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 interface for the 'E-Commerce' project. On the left, the navigation menu includes 'Overview', 'Boards', 'Work items', 'Backlogs', 'Sprints', 'Queries', 'Delivery Plans', 'Analytics views', 'Repos', 'Pipelines', 'Test Plans', and 'Artifacts'. The 'Backlogs' section is selected. The main area displays a backlog board titled 'E-Commerce Team'. It shows a hierarchy of work items:

- Feature: Product Management System
 - User Story: Edit Product Information
 - Task: Implement an edit feature for products.
 - Task: Create an edit product form.
 - Task: Test the edit functionality.
 - User Story: Delete Product Information
 - Task: Implement a delete feature for products.
 - Task: Add a confirmation prompt before deletion.
 - Task: Test the delete functionality.
 - User Story: Upload Product Information
 - Task: Create a Product model in Django with necessary fields.
 - Task: Implement the product upload form.
 - Task: Test the product upload feature.- Feature: User Authentication System
 - User Story: User Registration

Columns include Order, Work Item Type, Title, State, Effort, Business Area, and Tags. A search bar and various board management buttons are at the top right.

1. Fill in Epics

The screenshot shows the 'Create New Epic' dialog in Azure Boards. The title field is highlighted with a red border and contains the error message 'NEW EPIC * Field 'Title' cannot be empty.' Below the title is a large input field labeled 'Enter title'. To the right, there are sections for 'Planning' and 'Deployment'. The 'Planning' section includes fields for Priority (set to 2), Risk, Effort, Business Value, Time Criticality, Start Date, and Target Date. The 'Deployment' section includes a note about tracking releases and a 'Development' section with a 'Add link' button and a note about linking to Azure Repos. Other tabs like 'Classification' and 'Related Work' are visible at the bottom.

2. Fill in Features

The screenshot shows the 'Create Feature' page in Azure DevOps. The URL is dev.azure.com/UnknownAnonymous/E-Commerce/_backlogs/backlog/E-Commerce%20Team/Features. The page has a header with 'Azure DevOps' and a search bar. On the left, there's a sidebar with project navigation. The main area shows a 'Description' section with a placeholder 'Click to add Description.' and a 'Discussion' section with a comment input field. To the right, there are 'Planning' and 'Deployment' sections. The 'Planning' section includes fields for Priority (set to 2), Risk, Effort, Business Value, Time Criticality, Start Date, and Target Date. The 'Deployment' section includes a note about tracking releases via Releases and Options menu. Below these are 'Development' and 'Related Work' sections.

3. Fill in User Story Details

The screenshot shows the 'Create User Story' page in Azure DevOps. The URL is dev.azure.com/UnknownAnonymous/E-Commerce/_backlogs/backlog/E-Commerce%20Team/Stories. The layout is similar to the Features page, with a sidebar, a central form, and various sections like Description, Acceptance Criteria, and Discussion. The 'Acceptance Criteria' section has a placeholder 'Click to add Acceptance Criteria.'. The 'Classification' section includes a 'Value area' dropdown set to 'Business'. The right side features 'Planning' (Story Points, Priority 2, Risk), 'Deployment' (同樣的部署說明), 'Development' (Add link), and 'Related Work' sections.

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 uploaded Project.

Sprint Planning

Sprint 1

The screenshot shows the Azure DevOps Boards - Sprints page for the 'Music Playlist Batch Creator' project. The left sidebar includes options like Overview, Boards, Backlogs, Sprints (selected), Queries, Delivery Plans, Repos, Pipelines, and Artifacts. The main area displays a taskboard with columns for New, Active, Resolved, and Closed work items. A backlog item titled '19 As a user, I want to sign up and log in securely so that I can access my playlists' is moved from the backlog to the 'New' column. Another backlog item, '21 Implement NT-based authentication', is also visible in the backlog. The sprint duration is listed as '21 March - 4 April' with '4 work days remaining'.

Sprint 2

The screenshot shows the Azure DevOps Boards - Sprints page for the 'Music Playlist Batch Creator' project. The left sidebar is identical to the previous screenshot. The main area displays a taskboard for Sprint 2, which runs from '5 April - 15 April' over '7 work days'. A backlog item titled '46 As a user I should be able to add many songs at once into my playlist' has been moved to the 'New' column and is marked as 'Unassigned'. Another backlog item, '47 As a user I should be able to create audio playlist as I need', is in the backlog and is assigned to 'Karthick S'. A third backlog item, '53 Data needed to be collected', is also in the backlog and is assigned to 'Jagadeeswaran D'.

Sprint 3

The screenshot shows the Azure DevOps Boards backlog for the 'Music Playlist Batch Creator Team'. The backlog is organized into four columns: New, Active, Resolved, and Closed. The 'New' column contains four items:

- 48 As a user, I can Connect with my music apps (New, assigned to Mallu karthick Balaji R.)
- 20 As a user, I want to link my Spotify account or any more platforms so that I can import my music preferences (New, assigned to Mallu karthick Balaji R.)
- 49 As a user i need to have real time meta data (New, assigned to Mallu karthick Balaji R.)
- 58 As a user see my playlist in one place (New, assigned to Mallu karthick Balaji R.)

Sprint 4

The screenshot shows the Azure DevOps Boards backlog for the 'Music Playlist Batch Creator Team'. The backlog is organized into four columns: New, Active, Resolved, and Closed. The 'New' column contains three items:

- 23 As a user i can able to customize my playlist (New, assigned to Karthick S.)
- 50 As a user i should be able to rename record and change the playlist (New, assigned to Unsigned)
- 51 As a user i should be able to access a friendly and Modern UI (New, assigned to Koerthna S.)

Result:

The Sprints are created for the E-Commerce product uploaded Project.

EXP NO: 5

POKER ESTIMATION

Aim:

Create Poker Estimation for the user stories - E-Commerce product uploaded Creator Project.

Poker Estimation

The screenshot shows the Azure DevOps interface for creating a new user story. The story title is "As a seller, I want to upload new products with details and images so that customers can view and buy them." The "Planning" tab is active, showing Story Points set to 5, Priority 1, and Risk 0. The "Classification" tab shows Value area as Business. The "Development" tab includes a note about linking to Azure Repos. The "Related Work" tab shows an option to add a link.

Result:

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

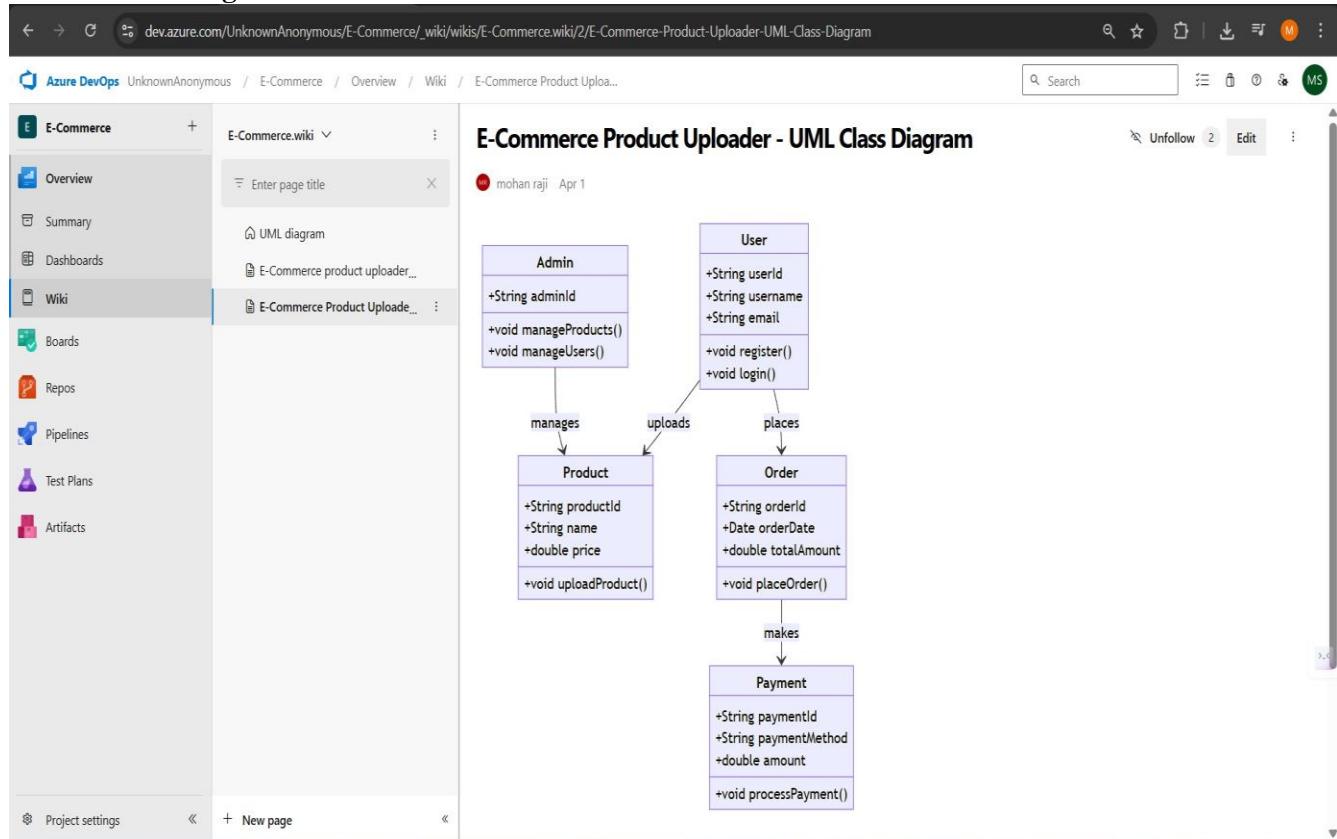
EXP NO: 6

DESIGNING CLASS AND SEQUENCE DIAGRAMS FOR PROJECT ARCHITECTURE

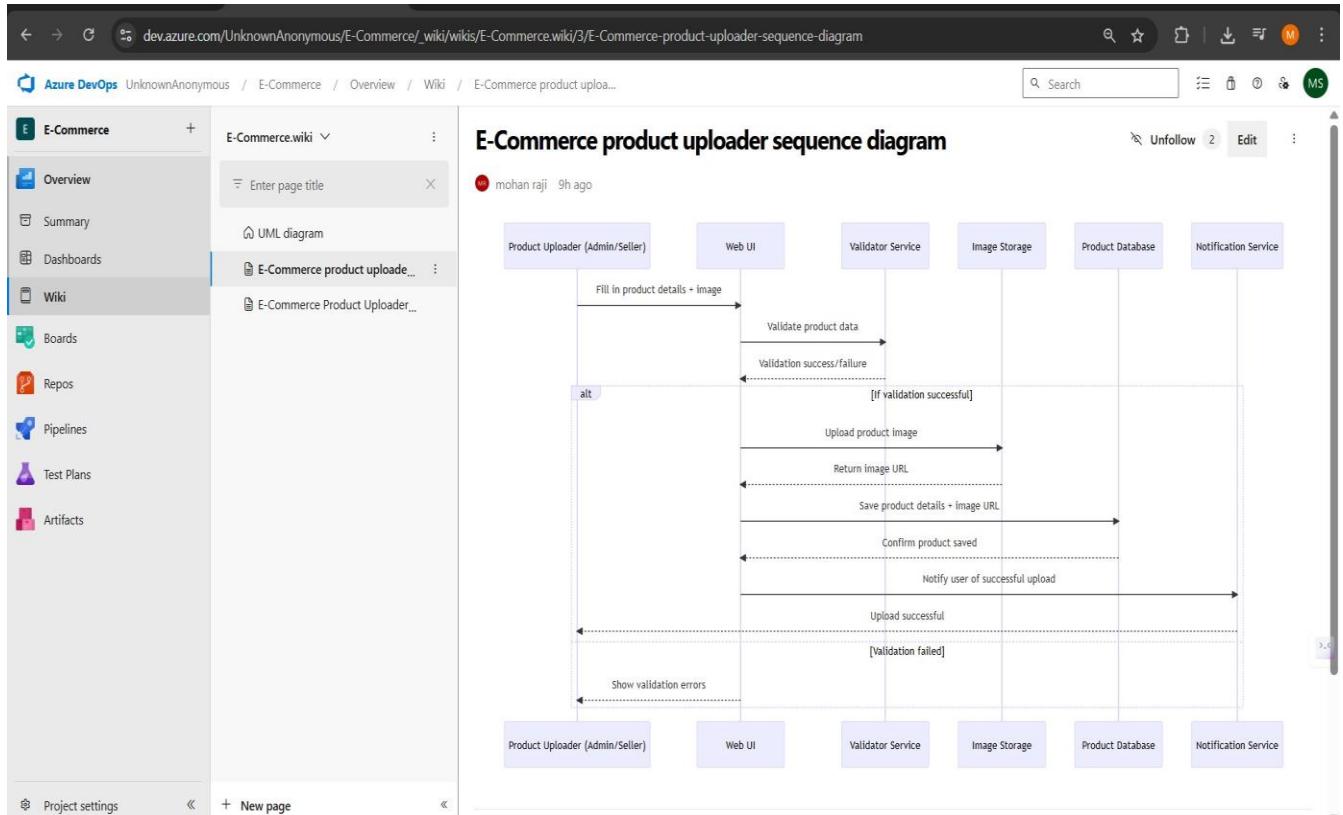
Aim:

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 uploaded.

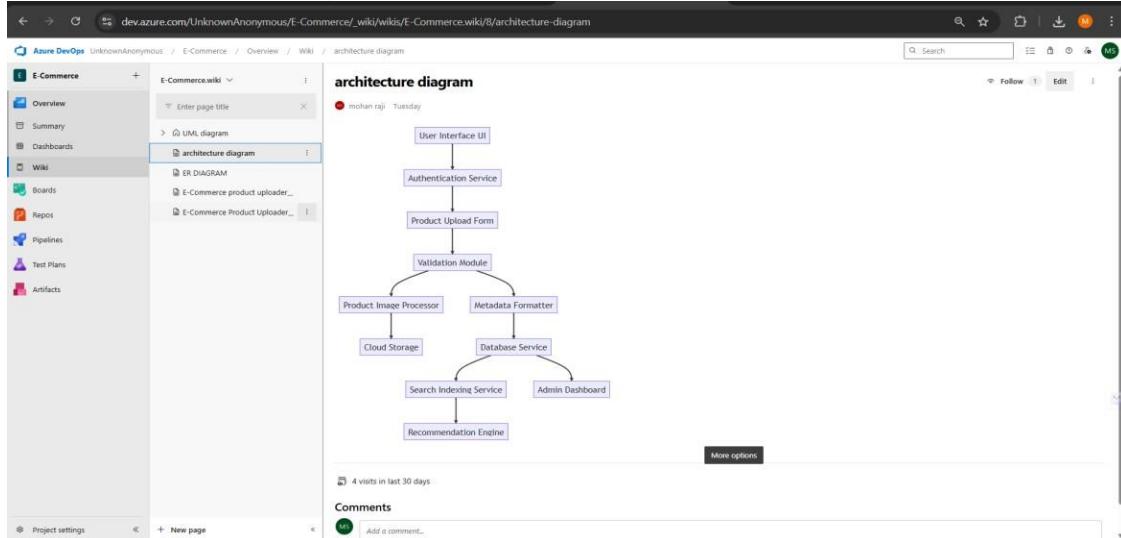
EXP NO: 7

DESIGNING ARCHITECTURAL AND ER DIAGRAMS FOR PROJECT STRUCTURE

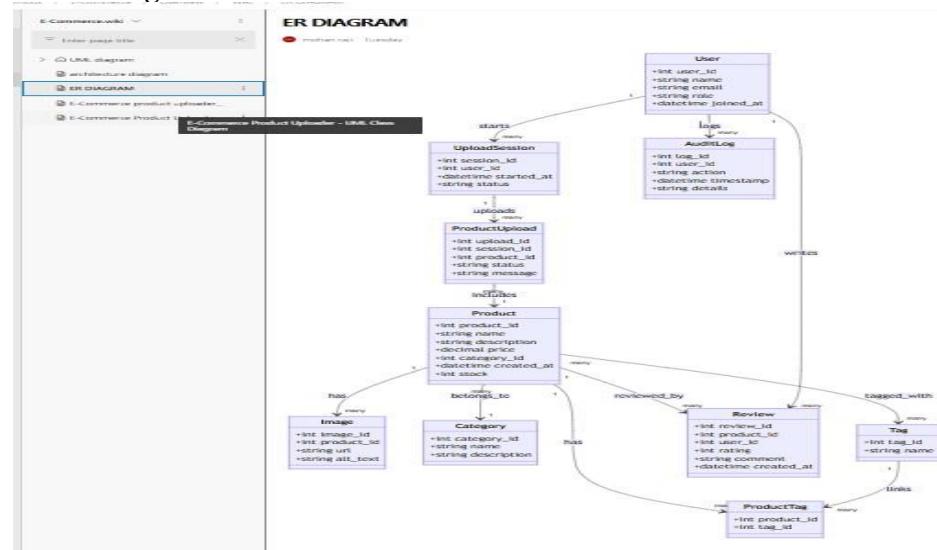
Aim:

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

7A. Architectural Diagram



7B. ER Diagram



m

Result:

The Architecture Diagram and ER Diagram is designed Successfully for the E-Commerce product uploaded.

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

The screenshot shows the 'New Test Plan' dialog in the Azure DevOps interface. The 'Name' field contains 'Music Playlist Batch Creator - Test Plan'. The 'Area Path' field is set to 'Music Playlist Batch Creator'. The 'Iteration' field is set to 'Music Playlist Batch Creator\Integration'. At the bottom right of the dialog are 'Create' and 'Cancel' buttons.

2. Test suite

The screenshot shows the 'Test Suites' section of the Azure DevOps interface. A context menu is open over the 'Test Suites' header, showing options: 'New Suite', 'Assign configurations', 'Export', 'Assign testers to run all tests', and 'Import test suites'. Below this, the 'TS01 - User Login (ID: 86)' suite is listed with its four test cases:

Title	Order	Test Case Id	Assigned To	Status
TC01 - Successful Sign Up	1	78	Karthikeyan Se... Design	
Static suite	2	80	Karthikeyan Se... Design	
Requirement based suite	3	81	Karthikeyan Se... Design	
Query based suite	4	82	Karthikeyan Se... 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.

Music Playlist Batch Creator – Test Plans

USER STORIES

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

Test Suites

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

1. TC01 – Successful Sign Up

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

2. TC02 – Secure Login

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

3. TC03 – Sign Up with Existing Email

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

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

4. TC04 – Login with Wrong Password

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

Test Suit: TS02 - 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

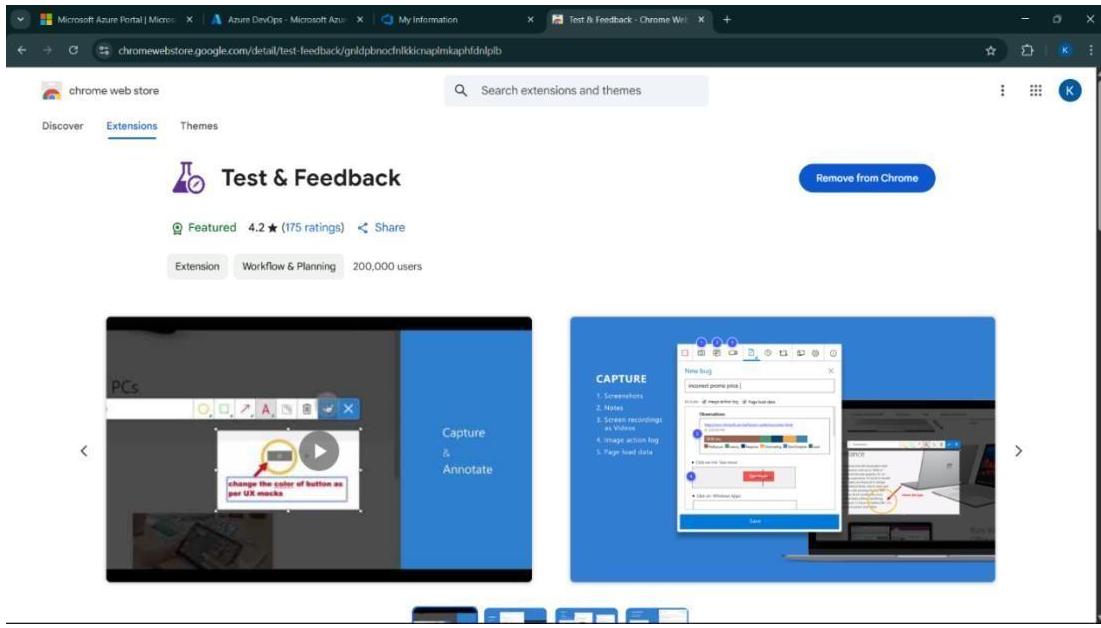
The screenshot shows the Microsoft Azure DevOps Test Plan interface. A test case titled "77 TC06 – Playlist Loading Failure" is displayed. The test case details are as follows:

- Owner:** Karthick S.
- Area:** Music Playlist Batch Creator
- Iteration:** Music Playlist Batch Creator\Integration
- Steps:**
 - 1. Disconnect from internet → Expected result: Network is offline
 - 2. Navigate to "My Playlists" → Expected result: Error message "Unable to load playlists" is shown
- Custom:** Type: Error Path
- Status:** Priority: 2, Automation status: Not Automated

The screenshot shows the Azure DevOps Test Plan interface. A test case named 'TC05 – View Playlist Page' is selected. The 'Steps' section contains two steps: 'Log in successfully' and 'Navigate to "My Playlists" section'. The 'Expected result' for the first step is 'User is redirected to dashboard'. The 'Priority' is set to 2. The 'Automation status' is 'Not Automated'. The 'Custom' tab is selected, showing 'Type: Happy Path'. The sidebar on the left includes links for 'Music', 'Overview', 'Boards', 'Repos', 'Pipelines', 'Test Plans', 'Programs', 'Parameters', 'Configurations', 'Runs', and 'Artifacts'.

4. Installation of test

The screenshot shows the Chrome Web Store page for the 'Test & Feedback' extension. The extension has a rating of 4.2 stars from 175 ratings and 200,000 users. It is categorized under 'Workflow & Planning'. The page features a large screenshot of the extension's interface, which includes a toolbar with icons for screenshots, notes, audio recording, and image action log, and a main panel for capturing and annotating screens. A prominent 'Add to Chrome' button is visible at the top right.



Test and feedback

Showing it as an extension

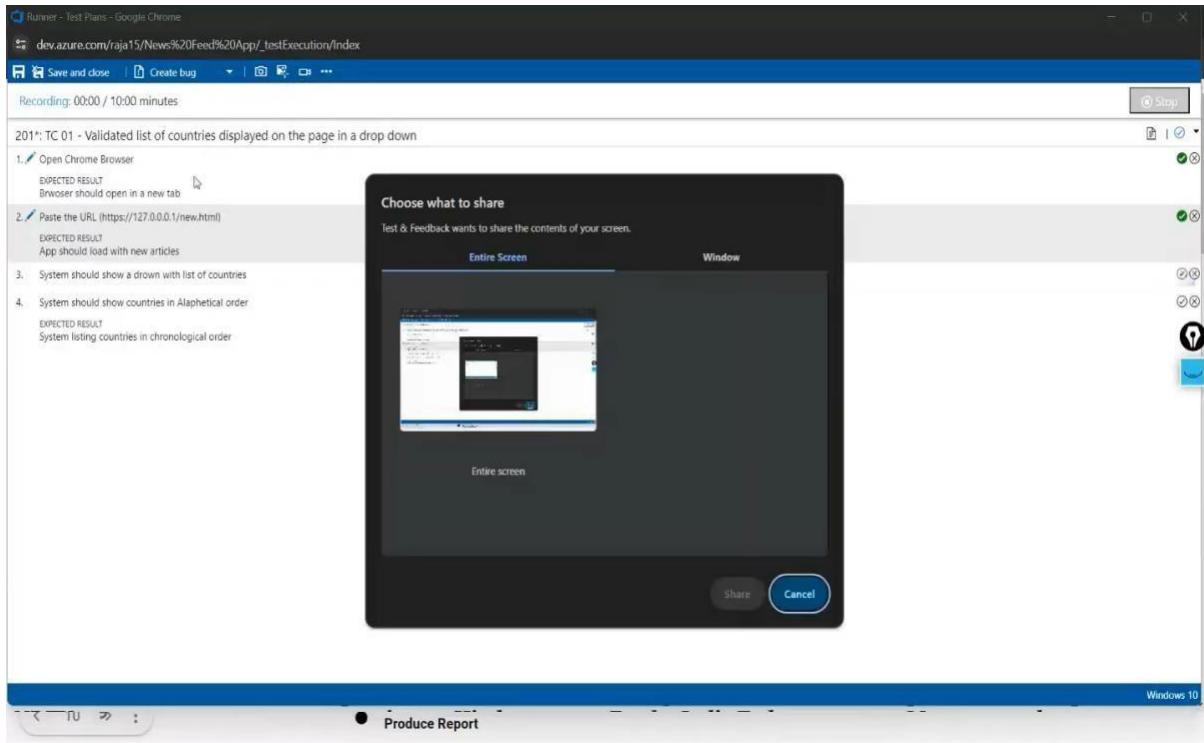
A screenshot of the Azure DevOps Test Plan interface. On the left, there's a sidebar with options like Overview, Boards, Repos, Pipelines, Test Plans, Test plans, Progress report, Parameters, Configurations, Runs, and Artifacts. The "Test plans" section is currently selected. In the main content area, a test plan titled "TS01 - User Login (ID: 86)" is displayed, showing four test cases: TC01 – Successful Sign Up, TC02 – Secure Login, TC03 – Sign Up with Existing Email, and TC04 – Login with Wrong Password. A modal window titled "Extensions" is open on the right, listing extensions with "Full access": Copy Text from Picture, Dark Reader, Monica: ChatGPT AI Assistant, Selectext: Copy Text from V..., and Test & Feedback. There are also "Manage extensions" and "Help" buttons in the modal.

5. Running the test cases

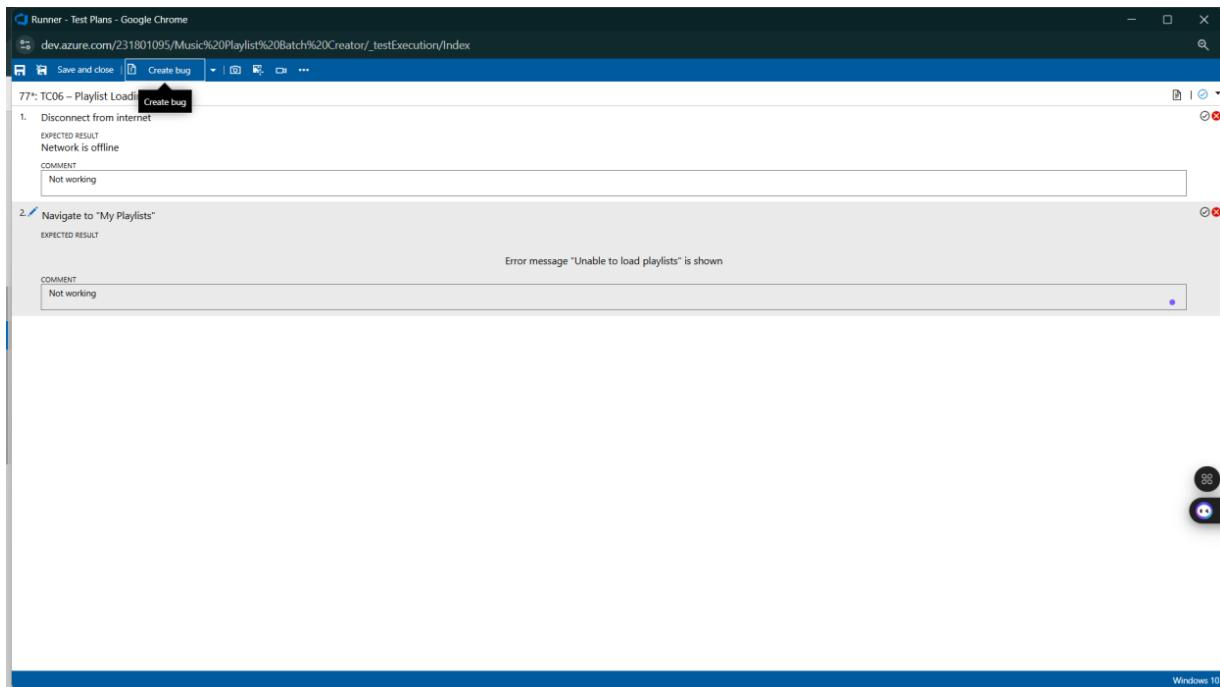
The screenshot shows the Azure DevOps Test Plan interface. On the left, the navigation bar includes 'Overview', 'Boards', 'Repos', 'Pipelines', 'Test Plans', 'Test plans' (which is selected), 'Progress report', 'Parameters', 'Configurations', 'Runs', and 'Artifacts'. Below this is 'Project settings'. The main area displays a 'Test Suites' view for 'Music Playlist Batch Creator - T...'. A specific suite, 'TS02 - View Playlists (ID: 87)', is selected. The 'Execute' tab is active, showing 'Test Points (2 items)'. Two test cases are listed: 'TC05 – View Playlist Page' (Passed) and 'TC06 – Playlist Loading Failure' (Not run). A context menu is open over 'TC05 – View Playlist Page', listing options: 'View execution history', 'Mark Outcome' (with 'Passed' checked), 'Run', 'Reset test to active', 'Edit test case', 'Assign tester', and 'View test result'. The status bar at the bottom right indicates 'Windows 10'.

The screenshot shows a browser window titled 'Runner - Test Plans - Google Chrome' with the URL 'dev.azure.com/231801095/Music%20Playlist%20Batch%20Creator/_testExecution/Index'. The page displays a test log for '75: TC05 – View Playlist Page'. Step 1: 'Log in successfully' (EXPECTED RESULT: User is redirected to dashboard). Step 2: 'Navigate to "My Playlists" section' (EXPECTED RESULT: All created playlists are displayed clearly). The status bar at the bottom right indicates 'Windows 10'.

6. Recording the test case



7. Creating the bug



The screenshot shows a bug entry in the Azure DevOps Test Plan interface. The title of the bug is "TB01 - Playlist loading spinner keeps spinning indefinitely on poor network". The bug is categorized under "Music Playlist Batch Creator" and is currently "Unassigned". The "Repro Steps" section details two steps: disconnecting from the internet and navigating to "My Playlists", both of which failed. The "Comments" section notes an error message: "Unable to load playlists" is shown. The "Test Configuration" is listed as "Windows 10". The "Planning" and "Deployment" sections are visible on the right, along with "Related Work" and "System Info" details.

This screenshot shows the same bug entry as above, but with the "System Info" section expanded. The "System Info" table contains the following data:

Browser - Name	Google Chrome 135
Browser - Language	en-IN
Browser - Height	864
Browser - Width	1536
Browser - User agent	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/135.0.0.0 Safari/537.36
Operating system - Name	Windows NT 10.0; Win64; x64
Operating system - Architecture	x86_64
Operating system - Processor model	11th Gen Intel(R) Core(TM) i3-1115G4 @ 3.00GHz
Operating system - Number of processors	4
Memory - Available	814784512
Memory - Capacity	8216240128
Display - Pixels per inch (X axis)	120
Display - Pixels per inch (Y axis)	120
Display - Device pixel ratio	1.25

8. Test case results

The screenshot shows the Azure DevOps interface for a test plan. On the left, the navigation bar includes 'Test Plans' and 'Test plans'. The main area displays a test suite named 'TS02 - View Playlists (ID: 87)' under 'Execute'. It lists two test points: 'TC05 – View Playlist Page' (Passed) and 'TC06 – Playlist Loading Failure' (Not Applicable). A detailed table of results is shown on the right, listing outcomes like Passed, Failed, and Not Applicable across various configurations and run times.

Outcome	TimeStamp	Configuration	Run by	Tester	Test Point
Passed	4m ago	Windows 10	Karthick S	Malu karthick Balaji ...	Music
Passed	12m ago	Windows 10	Karthick S	Malu karthick Balaji ...	Music
Not Applicable	12m ago	Windows 10	Karthick S	Malu karthick Balaji ...	Music
Passed	14m ago	Windows 10	Karthick S	Malu karthick Balaji ...	Music
Passed	Tuesday	Windows 10	Karthikayen Senthil	Malu karthick Balaji ...	Music
Passed	Saturday	Windows 10	Malu karthick Balaji ...	Malu karthick Balaji ...	Music
Failed	Saturday	Windows 10	Malu karthick Balaji ...	Malu karthick Balaji ...	Music
Passed	Apr 11	Windows 10	Karthick S	Malu karthick Balaji ...	Music
Passed	Apr 11	Windows 10	Karthick S	Malu karthick Balaji ...	Music

9. Test report summary

The screenshot shows the Azure DevOps interface for a work item titled 'BUG 203: BG 01 - Countries Drop down Not Available on the page'. The work item details include the state (New), reason (New), repro step (Active, Resolved, Closed), and steps with results (Passed, Failed). The planning section shows resolved reason, story points (2), priority (2), severity (3 - Medium), and activity. The deployment section provides instructions for tracking releases. The development section includes an add link button and related work links.

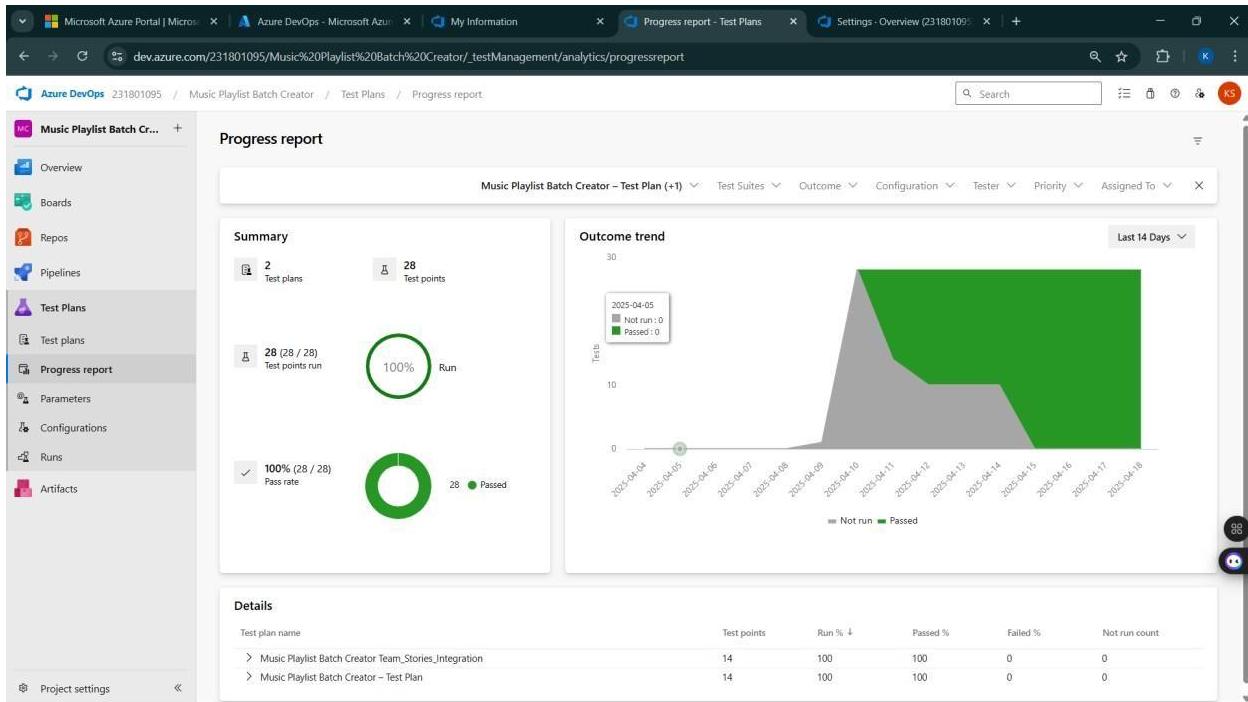
- Assigning bug to the developer and changing state

The screenshot shows a detailed view of a test run in Azure DevOps. The test case is titled "92 TB01 - Playlist loading spinner keeps spinning indefinitely on poor network". A specific step, "18-04-2025 03:23 Bug filed on 'TC06 – Playlist Loading Failure'", has failed. The failure reason is "Disconnect from internet". The expected result was "Network is offline", but the actual result was "Comments: Page Not loading". The error message "Unable to load playlists" is also noted. The test configuration is set to Windows 10. The planning section shows a priority of 2 and a severity of 3 (Medium). The deployment section indicates the work item is tracked by a release. The development section shows a link to an Azure Repos branch. The related work section lists a task for "TC06 – Playlist Loading Failure" that was updated on 10-04-2025. The system info section shows the work item was last updated by Karthick S 33m ago.

10. Progress report

The screenshot displays the progress report for the "Music Playlist Batch Creator" test plan. The summary section shows 14 test points, 14 test runs, and a 100% pass rate. The outcome trend chart tracks the progress over the last 14 days, showing a significant increase in green (Passed) status from April 10th to April 11th. The details section provides a breakdown of the test points across various test cases, all of which have a 100% pass rate and 0 failed points.

Test plan name	Test points	Run %	Passed %	Failed %	Not run count
Music Playlist Batch Creator - Test Plan	14	100	100	0	0
TS01 - User Login	4	100	100	0	0
TS02 - View Playlists	2	100	100	0	0
TS03 - Real-Time Metadata	2	100	100	0	0
TS04 - Playlist Editing	4	100	100	0	0
TS05 - Smart Playlist Creation	2	100	100	0	0



11. Changing the test template

All processes

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 page. The 'Processes' tab is selected. The list includes:

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 page. The 'Processes' tab is selected. A new entry has been added under the Agile category:

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)	This template is flexible and will work great for most teams using Agile planning methods, including those pract...	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. Under the 'Definition' tab, the 'Create a field' option is selected. The 'Name' field is set to 'Type', and the 'Type' dropdown is set to 'Text (single line)'. A description is provided: 'Optionally provide a description for the field'. At the bottom right of the dialog are 'Add field' and 'Cancel' buttons.

The screenshot shows the 'All processes > 231801095 Agile' page. The 'Projects' tab is selected. A new work item type named 'Music Playlist Batch Creator' is listed. The 'Name' column shows 'Music Playlist Batch Creator' and the 'Description' column shows a truncated text: 'Azure Music Playlist Batch Creator The Azure Music Playlist Batch Creator is a cloud-based solution designed for bulk playlist creation and management. Levera...'. The 'Type' field has been added to this work item type.

The screenshot shows the Azure DevOps Settings - Process page for a specific organization setting. The left sidebar lists various settings categories like General, Security, Boards, and Pipelines. The main area displays a 'Test Case' configuration for the 'Steps' field. The 'Steps' field is defined as 'Text (multiple lines)'. To the right of the field definition, there are several sections: 'Recent test results' (Recent test case results), 'Deployment' (Deployments), 'Development' (Links), 'Related Work' (Links), and 'Status' (Priority: Integer, Automation status). A large 'Add a field ...' button is located at the bottom right of the configuration area.

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 PERFORMANCE TESTING
-----------	---

Aim:

To create an Azure Load Testing resource and run a load test to evaluate the performance of a target endpoint.

Load Testing**Steps to Create an Azure Load Testing Resource:**

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

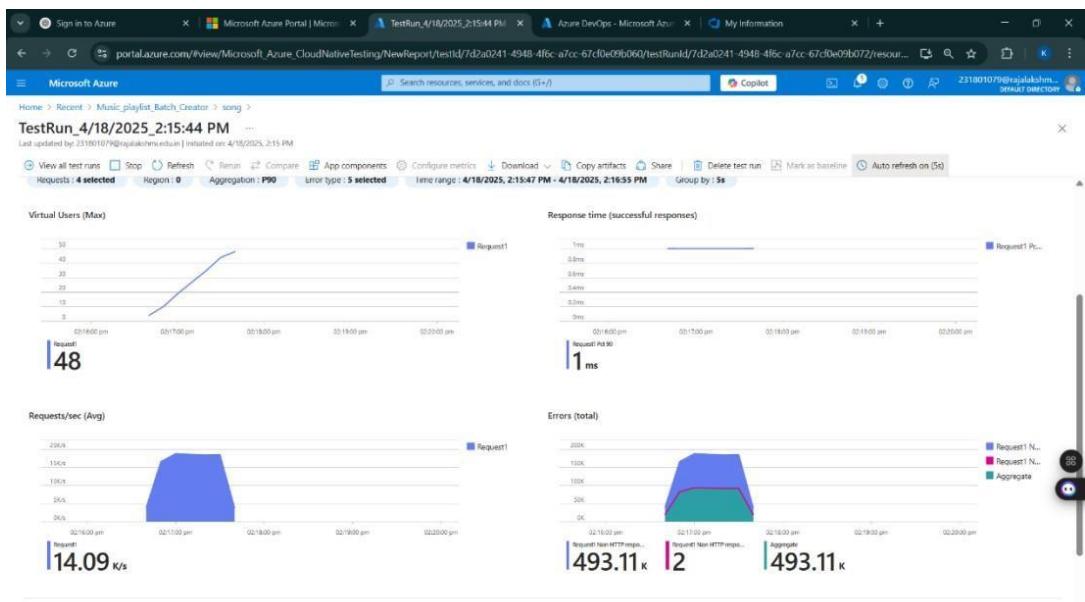
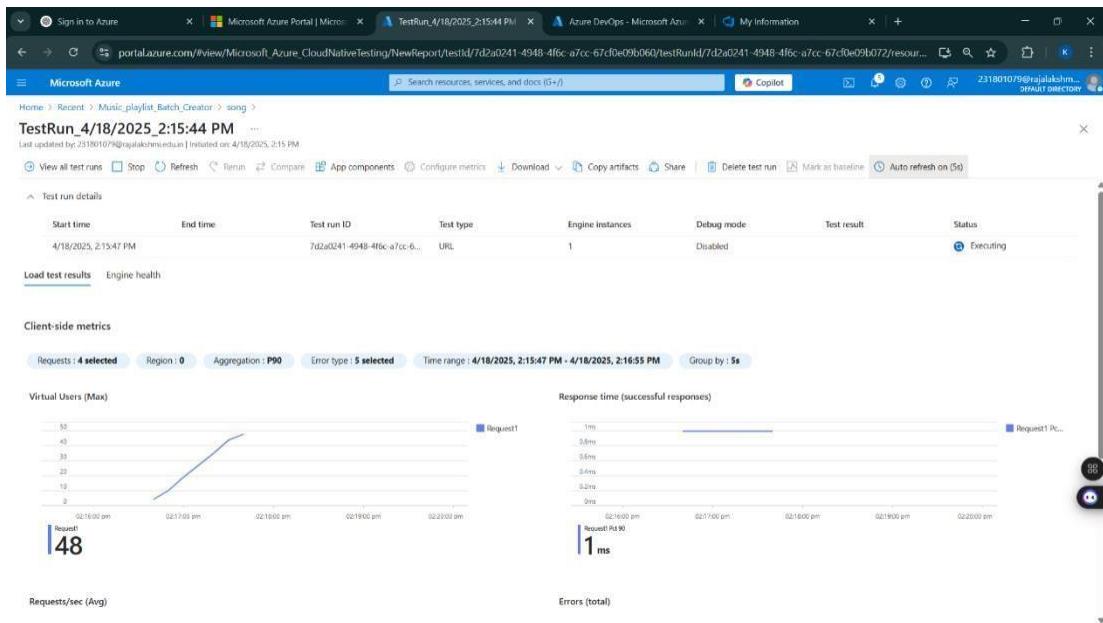
1. Sign in to Azure Portal
Go to <https://portal.azure.com> and log in.
2. Create the Resource
 - o Go to *Create a resource* → Search for “Azure Load Testing”.
 - o Select Azure Load Testing and click Create.
3. Fill in the Configuration Details
 - o *Subscription*: Choose your Azure subscription.
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



Result:

Successfully created the Azure Load Testing resource and executed a load test to assess the performance of the specified endpoint.

EXP NO: 10

GITHUB: PROJECT STRUCTURE & NAMING CONVENTIONS

Aim:

To provide a clear and organized view of the project's folder structure and file naming conventions, helping contributors and users easily understand, navigate, and extend the Music Playlist Batch Creator project.

GitHub Project Structure

The screenshot shows a GitHub repository page for 'E-Commerce_product_uploader'. The repository is public and has 1 branch and 0 tags. The main branch has 57 commits. The commit history includes updates to README.md, deletions of azureBoards/i, azureDashBoard/i, ecom, poker Estimation/i, screen_shots_diagrams/i, and sprint/i, and additions of files via upload. The repository description states: 'This is a web-based Product Upload and Management System built with Django and PostgreSQL, allowing admins to upload products with availability, images, and categorized details. Users can browse through the products and purchase them based on availability.' The repository has 0 stars, 1 watching, and 0 forks. The README file contains the text 'E-commerce Product Uploader'.

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.