



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: Iraiyanbu S T

Register No: 231801061

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:

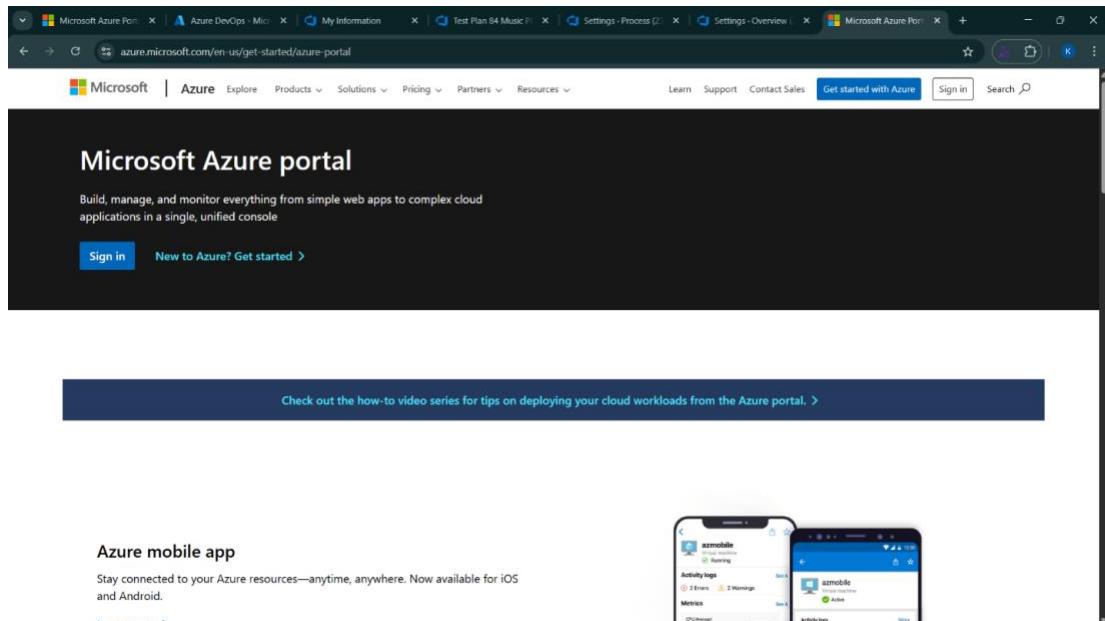
To set up and access the Azure DevOps environment by creating an organization through the Azure portal.

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

Welcome to Azure!

Don't have a subscription? Check out the following options.

Start with an Azure free trial
Get \$200 free credit toward Azure products and services, plus 12 months of popular [free services](#).

Manage Microsoft Entra ID
Manage access, set smart policies, and enhance security with Microsoft Entra ID.

Azure for Students
Get free software, Azure credit, or access Azure Dev Tools for Teaching after you verify your academic status.

Azure services

- Create a resource
- Azure DevOps organizations
- Quickstart Center
- Azure AI services
- Kubernetes services
- Virtual machines
- App Services
- Storage accounts
- SQL databases
- More services

Resources

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

Welcome to Azure!

Don't have a subscription? Check out the following options.

Start with an Azure free trial
Get \$200 free credit toward Azure products and services, plus 12 months of popular [free services](#).

Azure services

- Create a resource
- Azure DevOps organizations
- Quickstart Center
- Azure AI services
- Kubernetes services
- Virtual machines
- App Services
- Storage accounts
- SQL databases
- More services

Resources

Search bar: devops

Search results (Services tab):

- All
- Services (7)
- Resources
- More (4)

Services:

- Azure Native New Relic Service
- Managed DevOps Pools
- Azure DevOps organizations**
- Azure Native Dynatrace Service

Marketplace:

- Static Web App
- Build Agents for Azure DevOps
- Rocky Linux 9
- InfluxDB Cloud (Official Version)

Documentation:

- Course AZ-400T00-A: Designing and Implementing Microsoft DevOps solutions...
- Managed DevOps Pools Overview - Managed DevOps Pools
- DevOps considerations - Cloud Adoption Framework

Feedback: Continue searching in Microsoft Entra ID | Give feedback

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+J)

Copilot

iraiyanbut@outlook.com

DEFAULT DIRECTORY (IRAIYANBL...)

Home > Azure DevOps ...

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.

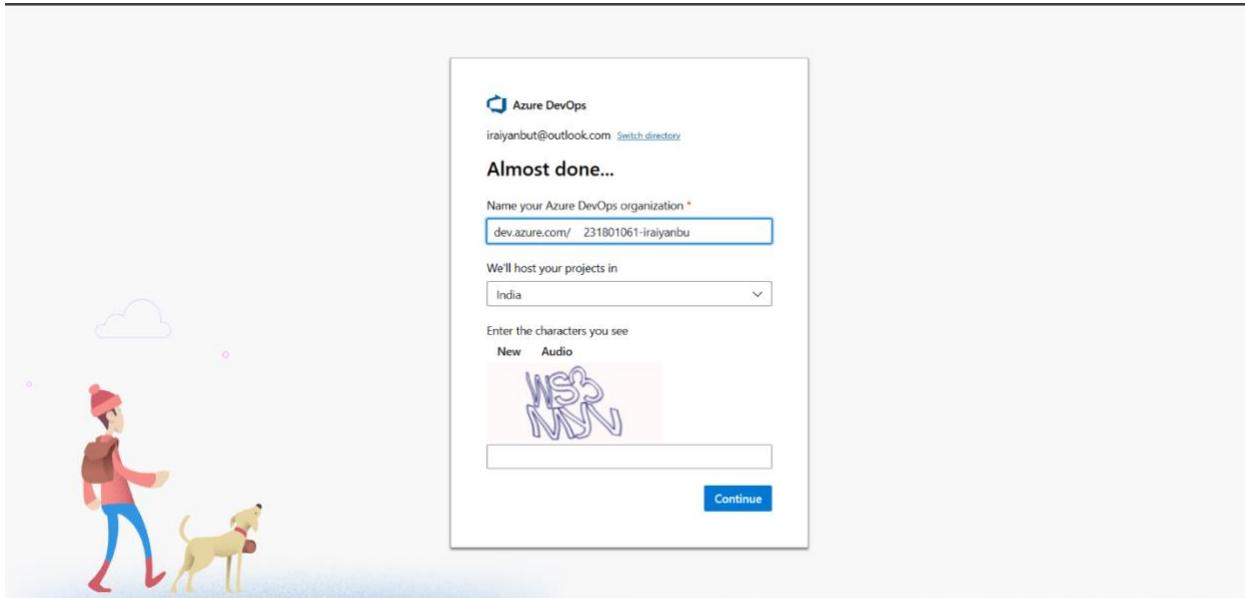
EXP NO: 2

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.

By creating this project, you agree to the Azure DevOps [code of conduct](#)

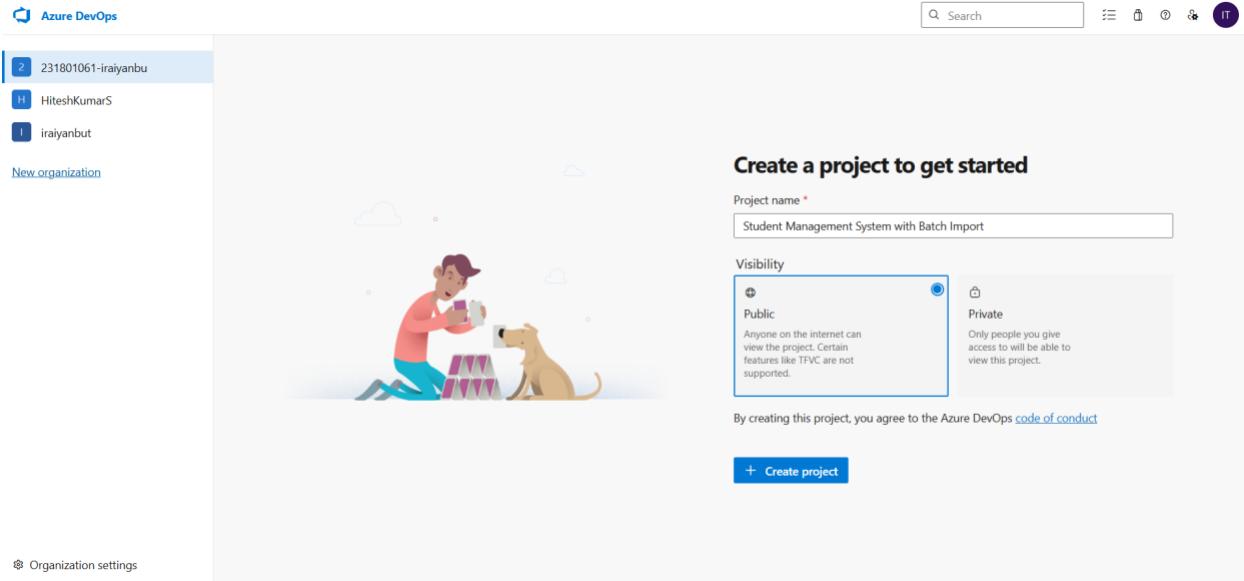
Advanced

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.



4. Project dashboard

The screenshot shows the Azure DevOps project dashboard for 'Student data management system with batch import'. The top navigation bar includes the Azure DevOps logo, a search bar, and various icons. The left sidebar shows the project name 'Student data management ...' and a list of project management tools: Overview, Summary, Dashboards, Wiki, Boards, Repos, Pipelines, Test Plans, and Artifacts. The main content area displays the project title 'Student data management system with batch import'. It includes a 'About this project' section with a placeholder for a project description and a 'Help others to get on board!' message. The 'Project stats' section provides an overview of recent activity, showing 3 work items created and 0 work items completed over the last 7 days. The 'Members' section lists five team members: HU, HS, IT, HS, and HS, each represented by a colored circular icon. The URL 'https://dev.azure.com/hiteshkumar5/' is visible at the bottom left.

5. To manage user stories:

- 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.
- 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.

Azure DevOps HiteshKumarS / Student data management ... / Boards / Backlogs

Did you notice Azure Boards has a new look and awesome new features? [Learn more.](#)

Student data management system with batch import Team

+ New Work Item View as Board Column Options Epics

Backlog Analytics

Order	Work Item Type	Title	State	Effort	Business Value
1	Epic	Student Management System with Batch Import <ul style="list-style-type: none"> Batch Import of Student Data User Authentication & Authorization Student Data Management Admin & System Maintenance Reporting & Analytics 	New		
	Feature	> Batch Import of Student Data	New		
	Feature	> User Authentication & Authorization	New		
	Feature	> Student Data Management	New		
	Feature	> Admin & System Maintenance	New		
	Feature	> Reporting & Analytics	New		

Planning
Drag and drop work items to include them in a sprint.

Sprint 3 4/10/2025 - 4/30/2025
Planned Effort: 16 15 working days
2 2

Sprint 4 5/1/2025 - 5/21/2025
Planned Effort: 21 15 working days
3 3

+ New Sprint

Search

IT

Microsoft Sign out

Iraiyanbu S T
iraiyanbut@outlook.com
[My Microsoft account](#)
[Switch directory](#)

Business Value

Sign in with a different account

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 with the following details:

- Project:** Student data management...
- Team:** Student data management system with batch import
- Backlog View:** Backlog
- Work Items:**
 - An Epic titled "Student Management System with Batch Import" has several child items:
 - Feature: "Import of Student Data"
 - User Story: "As an admin, I want to upload student data in bulk so..."
 - User Story: "As an admin, I want to preview data and see errors before..."
 - Task: "Implement data validation checks and a preview scr..."
 - Feature: "User Authentication & Authorization"
 - Feature: "Student Data Management"
 - Feature: "Admin & System Maintenance"
 - Feature: "Reporting & Analytics"
- Planning Panel:** Shows two sprints:
 - Sprint 3:** 4/10/2025 - 4/30/2025, Planned Effort: 16 - 15 working days, 2 tasks assigned.
 - Sprint 4:** 5/1/2025 - 5/21/2025, Planned Effort: 21 - 15 working days, 3 tasks assigned.

1. Fill in Epics

The screenshot shows the Azure Boards Epic creation dialog with the following details:

- Title:** Student Management System with Batch Import
- Description:** Click to add Description.
- Planning:**
 - Priority: 2
 - Risk:
 - Effort:
 - Business Value:
 - Time Criticality:
 - Start Date: Select a date...
 - Target Date: Select a date...
- 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
- Related Work:**

2.Fill in Features

The screenshot shows the 'FEATURE 11' creation page. At the top, there's a header with a profile picture, the title 'FEATURE 11', and a save button. Below the header, there are sections for 'Description' (with a rich text editor), 'Planning' (Priority: 2, Risk: 1, Story Points: 8, Priority: 1), 'Deployment' (with a note about tracking releases), 'Development' (with a note about linking to Azure Repos), and 'Related Work'. The 'Details' tab is selected at the bottom.

3.Fill in User Story Details

The screenshot shows the 'USER STORY 12' creation page. At the top, there's a header with a profile picture, the title 'USER STORY 12', and a save button. Below the header, there are sections for 'Description' (with a rich text editor), 'Acceptance Criteria' (listing requirements like CSV upload and validation), 'Planning' (Story Points: 8, Priority: 1, Risk: 1), 'Classification' (Value area: Business), 'Deployment' (with a note about tracking releases), 'Development' (with a note about linking to Azure Repos), and 'Related Work'. The 'Details' tab is selected at the bottom.

Result:

Thus, the creation of epics, features, user story and task has been created successfully.

EXP NO: 4	SPRINT PLANNING
------------------	------------------------

Aim:

To assign user story to specific sprint for the Music Playlist Batch Creator Project.

**Sprint Planning
Sprint 1**

Sprint 1

New	Active	Closed
6 As a user, I want to log out securely so that my session is terminated properly. New Hemanth Kumar D	32 Add a logout button to the UI, clear session data, and redirect users to the login page. New Hemanth Kumar D	
24 As an admin, I want to assign specific roles to users so that they have appropriate permissions. New Hemakumar U	29 Develop an admin panel for assigning and managing user roles. New Hemakumar U	

Sprint 2

Sprint 2

New	Active	Closed
8 As an admin, I want to add, edit, and delete student records so that data remains accurate. New Hemaprasath D S	45 Develop REST API and UI for CRUD operations with input validation. New Hemaprasath D S	
9 As a teacher, I want to view student profiles so that I can monitor student progress. Active Hitesh Kumar S	47 Create a searchable student database UI to display student records. Active Hitesh Kumar S	
65 As a teacher, I want to search and filter student ... New	66 Add filters based on grade, name, or enrollment New	

Sprint 3

The screenshot shows the Azure Boards interface for the "Student data management system with batch import" project. The left sidebar navigation includes: Overview, Boards, Work items, Boards, Backlogs, Sprints (selected), Queries, Delivery Plans, Analytics views, Repos, Pipelines, Test Plans, Artifacts, and Project settings. The main area displays the backlog for "Sprint 3". The backlog is organized into columns: New, Active, and Closed. Two work items are visible in the New column:

- 12** As an admin, I want to upload student data in bulk so that I can save time.
Status: New
Assigned to: Hemanth Kumar D...
- 40** Implement file upload functionality for CSV/Excel student data.
Status: New
Assigned to: Hemanth Kumar D...

At the bottom of the backlog, there are two more items:

- 60** As an admin, I want to preview data and see errors before importing it into the system.
Status: New
Assigned to: Hemaprasath D S...
- 61** Implement data validation checks and a preview screen for import errors.
Status: New
Assigned to: Hemaprasath D...

Sprint 4

The screenshot shows the Azure Boards interface for the "Student data management system with batch import" project. The left sidebar navigation is identical to the previous screenshot. The main area displays the backlog for "Sprint 4". The backlog is organized into columns: New, Active, and Closed. Two work items are visible in the New column:

- 16** As a student, I want to view my academic records so that I can stay informed about my performance.
Status: New
Assigned to: Unassigned
- 62** Implement report generation logic and visualization in the UI.
Status: New
Assigned to: Unassigned

At the bottom of the backlog, there are two more items:

- 21** As an admin, I want to monitor system logs so that I can detect and resolve issues quickly.
Status: New
Assigned to: Unassigned
- 38** Set up a logging dashboard for tracking errors and system events.
Status: New
Assigned to: Unassigned

Result:

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

EXP NO: 5

POKER ESTIMATION

Aim:

Create Poker Estimation for the user stories - Music Playlist Batch Creator Project.

Poker Estimation

USER STORY 16

16 As a student, I want to view my academic records so that I can stay informed about my performance.

No one selected 0 Comments Add Tag

State: New Area: Student data management system with batch import Updated by Hemanth Kumar D: Apr 20

Reason: New Iteration: Student data management system with batch import\Sprint 4

Description

Acceptance Criteria

- students can generate student reports from the system.
- Reports should be exportable in **PDF** and **Excel** formats.

Discussion

Add a comment. Use # to link a work item, @ to mention a person, or ! to link a pull request.

Planning

Story Points: 8
Priority: 2
Risk: R

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.

Classification

Value area: Business

Related Work

Result:

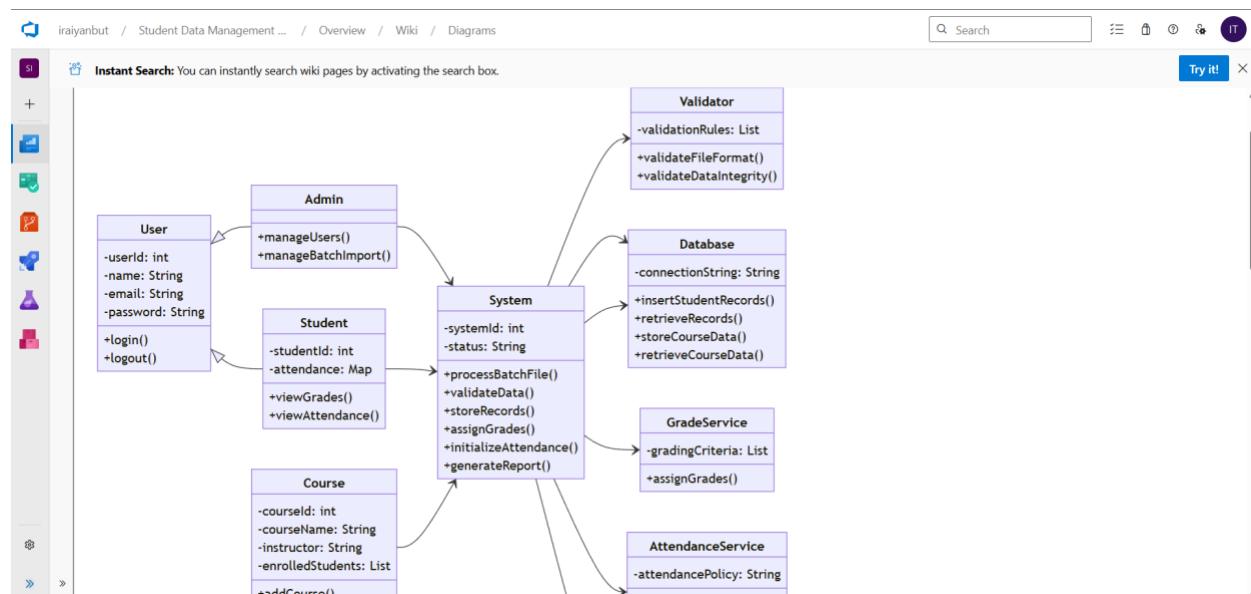
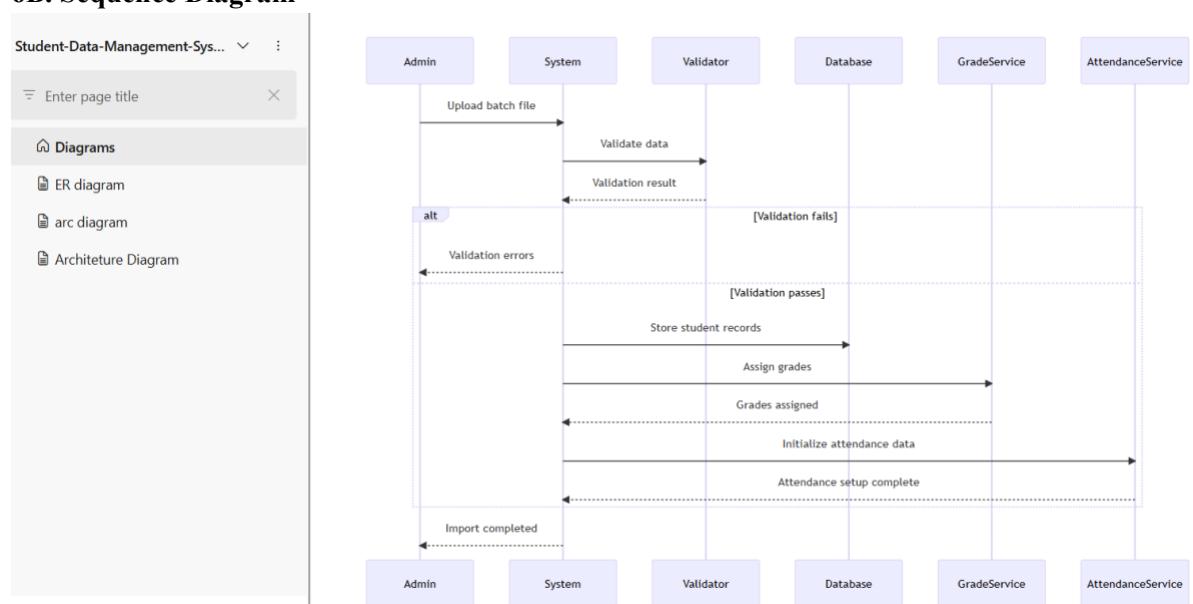
The Estimation/Story Points is created for the project using Poker Estimation.

EXP NO: 6

DESIGNING CLASS AND SEQUENCE DIAGRAMS FOR PROJECT ARCHITECTURE

Aim:

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

6A. Class Diagram**6B. Sequence Diagram**

Result:

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

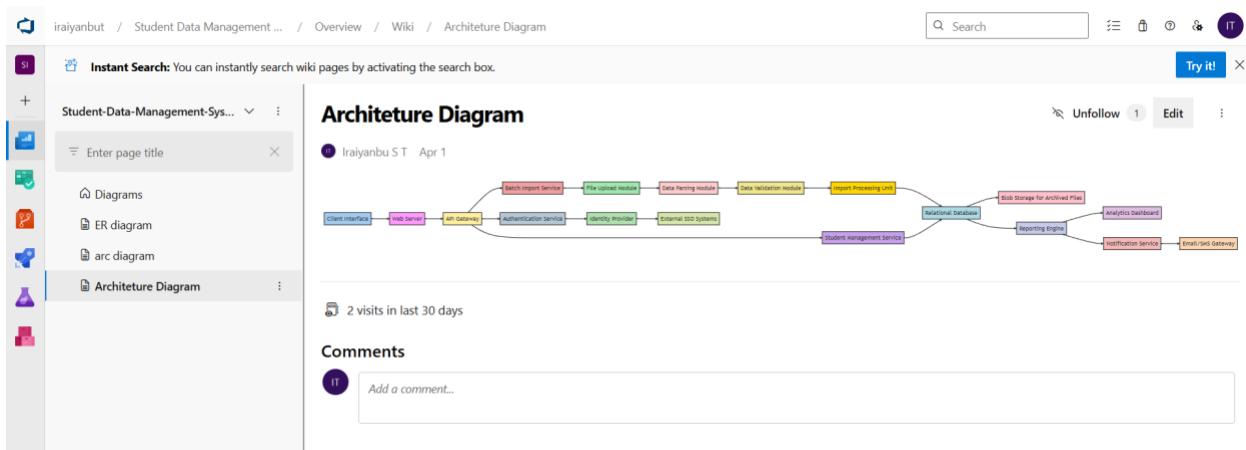
EXP NO: 7

DESIGNING ARCHITECTURAL AND ER DIAGRAMS FOR PROJECT STRUCTURE

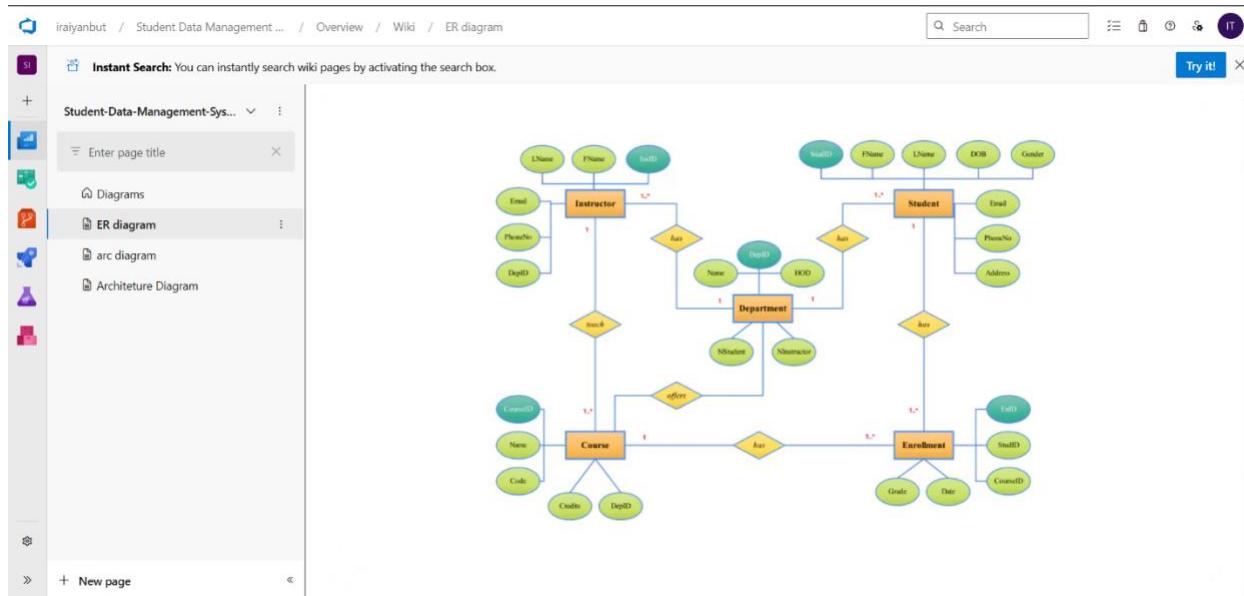
Aim:

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

7A. Architectural Diagram



7B.ER Diagram



Result:

The Architecture Diagram and ER Diagram is designed Successfully for the Music Playlist Batch Creator

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

TEST PLAN 69*

69 Student data management system-Test plans

Iraiyanbu S T | 0 Comments | Add Tag | Save and Close | Follow | Details

Description
Click to add Description.

Timelines
Start Date: 4/15/2025 5:59 AM | Finish Date: 4/22/2025 5:59 AM

Discussion
Add a comment. Use # to link a work item, @ to mention a person, or ! to link a pull request.

2. Test suite

Student data management system-Test plans

Test Suites

TS01-Batch Import Feature (ID: 72)

Title	Outcome	Order	Test Case Id
TC01-Upload valid CSV with complete student data	Passed	2	73
TC03-Verify Error for Invalid CSV Format	Passed	3	99
TC02-Upload File with Missing Headers	Passed	4	74

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

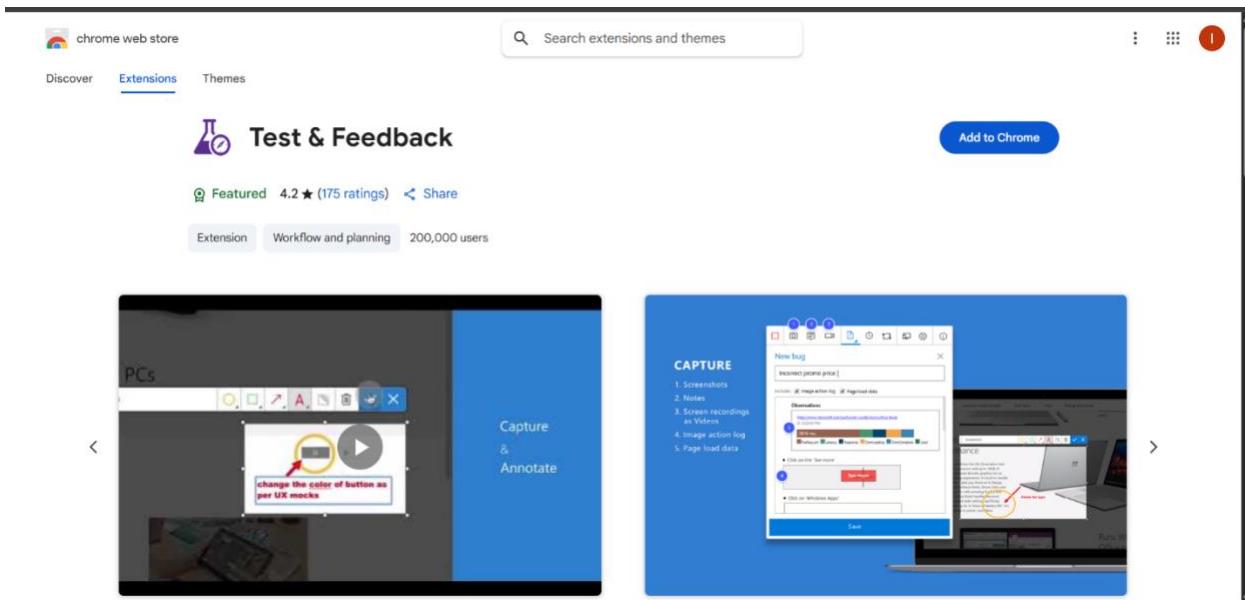
The screenshot shows a Microsoft Azure DevOps Test Case page. The title is 'TEST CASE 73'. The case is for 'TC01-Upload valid CSV with complete student data'. It is assigned to 'Iraiyarbu S T' and has a status of 'Design'. The area is 'Student data management system with batch import' and the reason is 'New'. Iteration is 'Student data management system with batch import'. The last update was by 'Hemanth Kumar D' on April 19. The page includes sections for 'Steps', 'Deployment', 'Development', and 'Related Work'.

Steps	Action	Expected result	Attachment
1.	Navigate to batch import page	The system loads the batch import interface with options to upload a file.	
2.	Upload a valid CSV file	The file is accepted by the system (validated for correct format and headers).	
3.	Click import	System processes the file. All valid student records are added to the database. A success message is shown (e.g., "100 students imported successfully").	

Click or type here to add a step

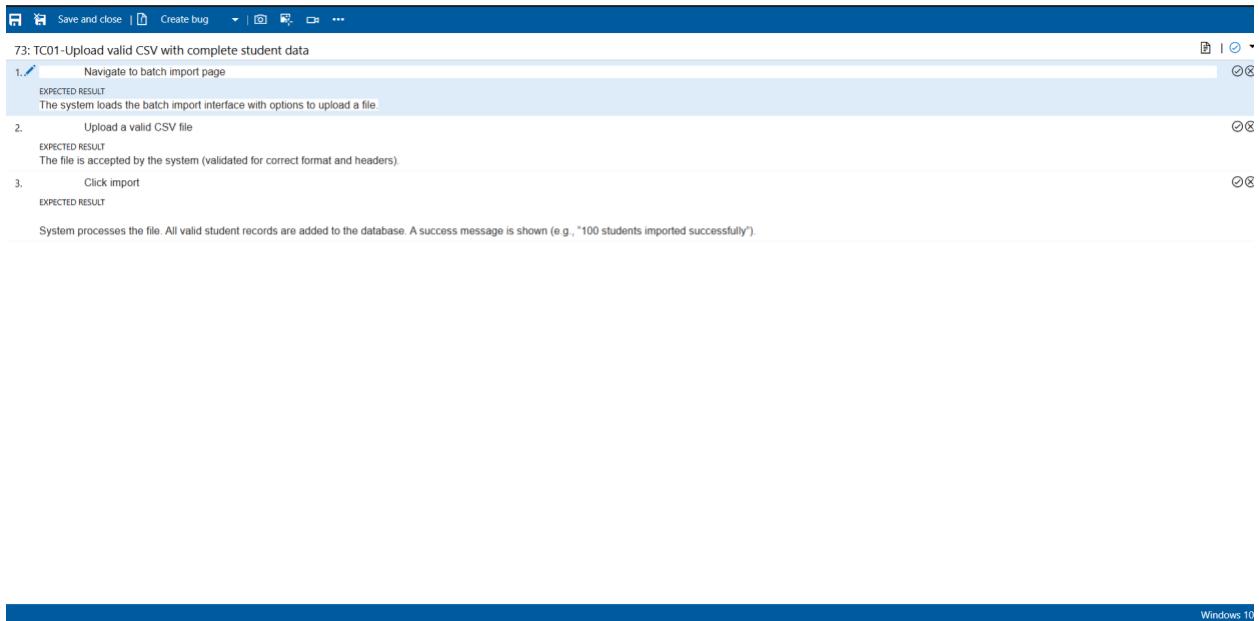
Parameter values

4. Installation of test

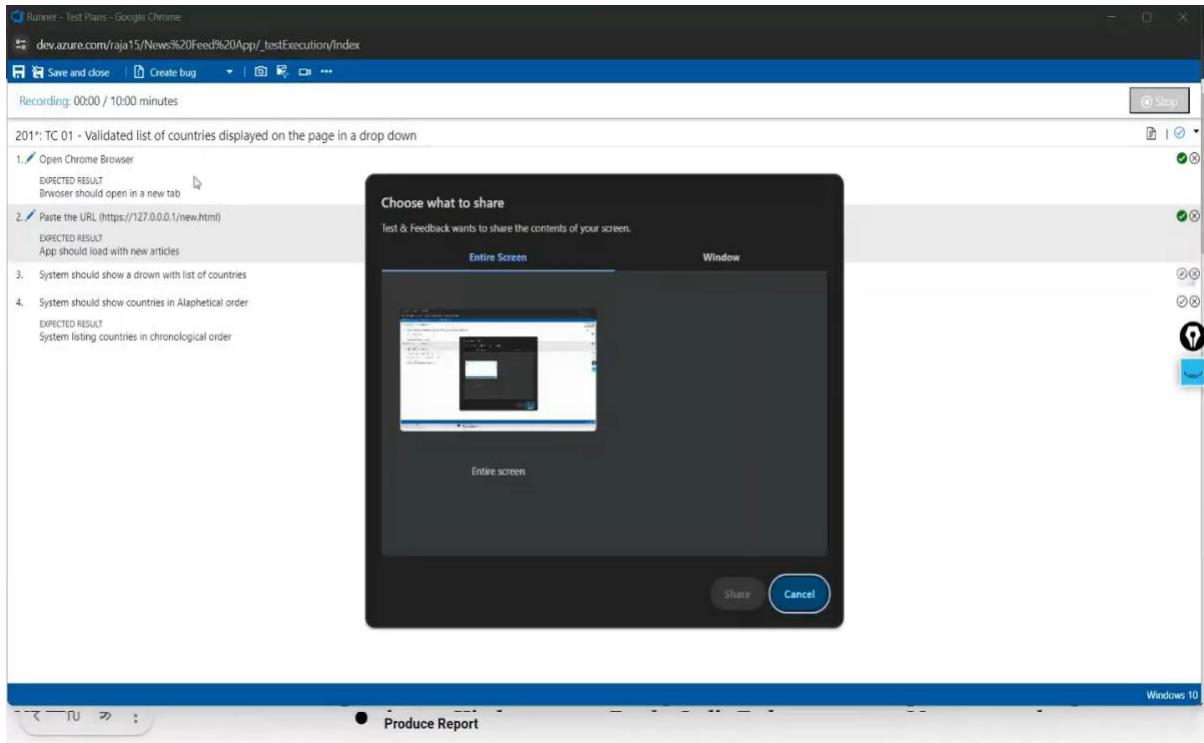


5. Running the test cases

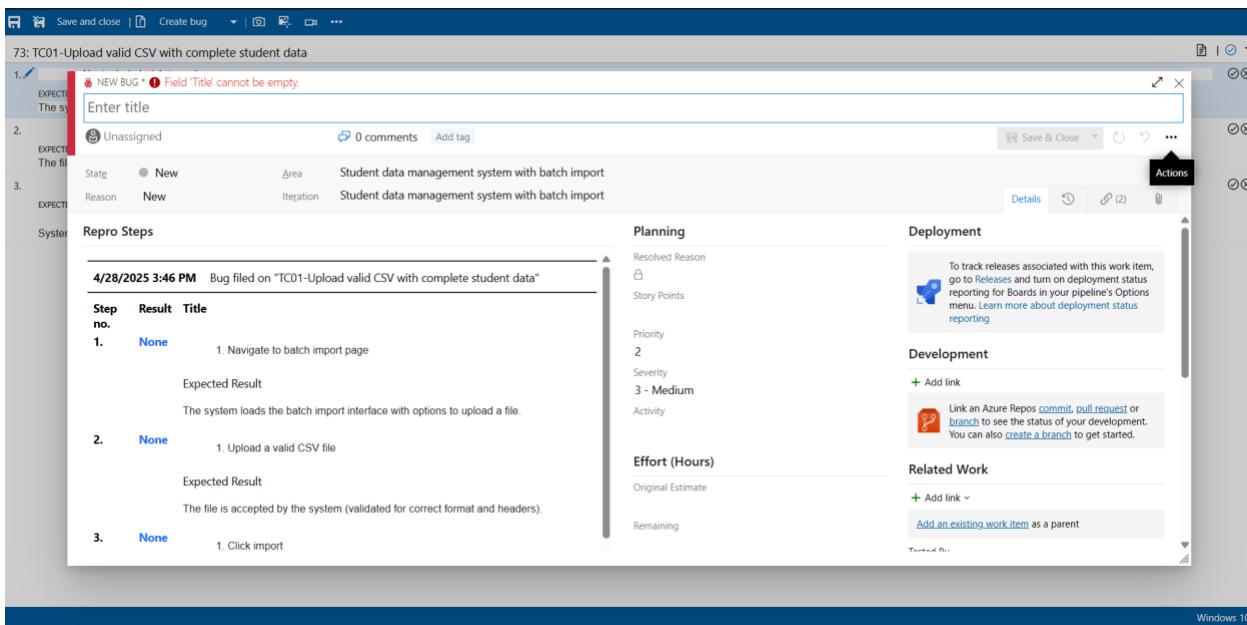
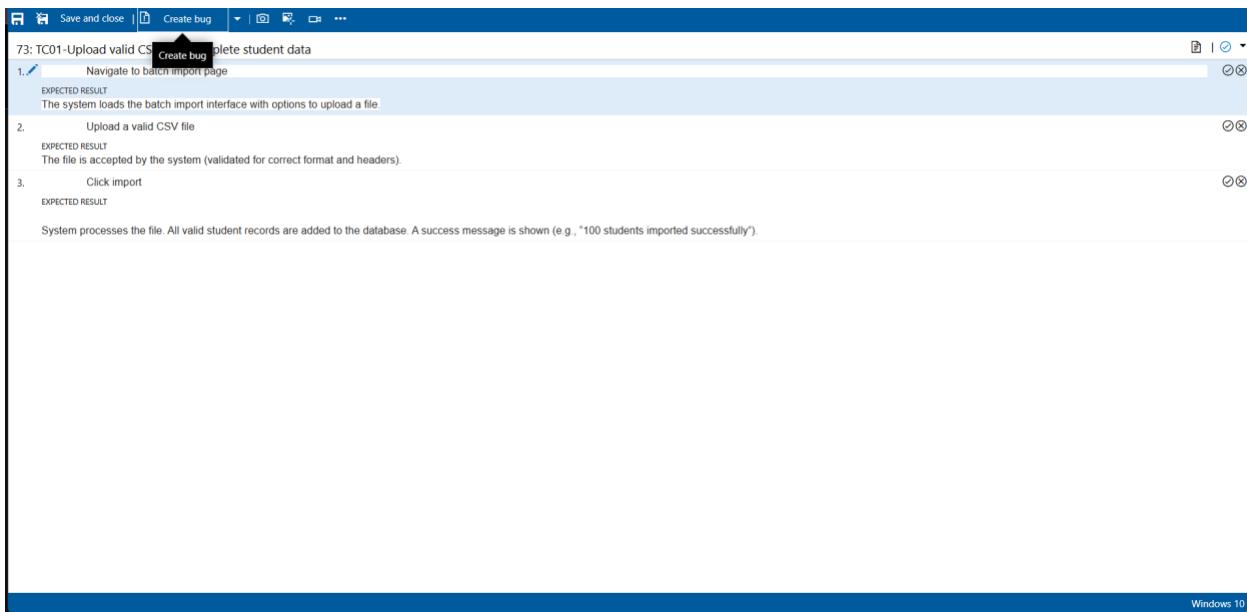
The screenshot shows the Microsoft Test Plan tool interface. On the left, the navigation pane includes 'Student data management system...', 'Overview', 'Boards', 'Repos', 'Pipelines', 'Test Plans', 'Test plans' (selected), 'Progress report', 'Parameters', 'Configurations', 'Runs', and 'Artifacts'. The main area displays the 'TS01-Batch Import Feature (ID: 72)' details. Under the 'Execute' tab, the 'Test Points (3 items)' section lists three test cases: 'TC01-Upload valid CSV with complete student data' (Passed, Order 2, Test Case Id 73), 'TC03-Verify Error for Invalid CSV Format' (Not Started), 'TC02-Upload File with Missing Headers' (Not Started). A context menu is open over the first test point, showing options like 'Run for web application', 'Run for desktop application', and 'Run with options'.



6.Recording the test case



7.Creating the bug



The screenshot shows the Azure DevOps interface for a test plan. The left sidebar is titled "Test Plans" and has several other options like Overview, Boards, Repos, Pipelines, Test Points, Progress report, Parameters, Configurations, Runs, and Artifacts. The main area shows a test run titled "Run 48 - TS02 - View Playlists (Manual) / TC06 – Playlist Loading Failure". It includes sections for "System Info" (with browser, OS, and display details), "Discussion" (with a comment placeholder), and a "Test Point" table.

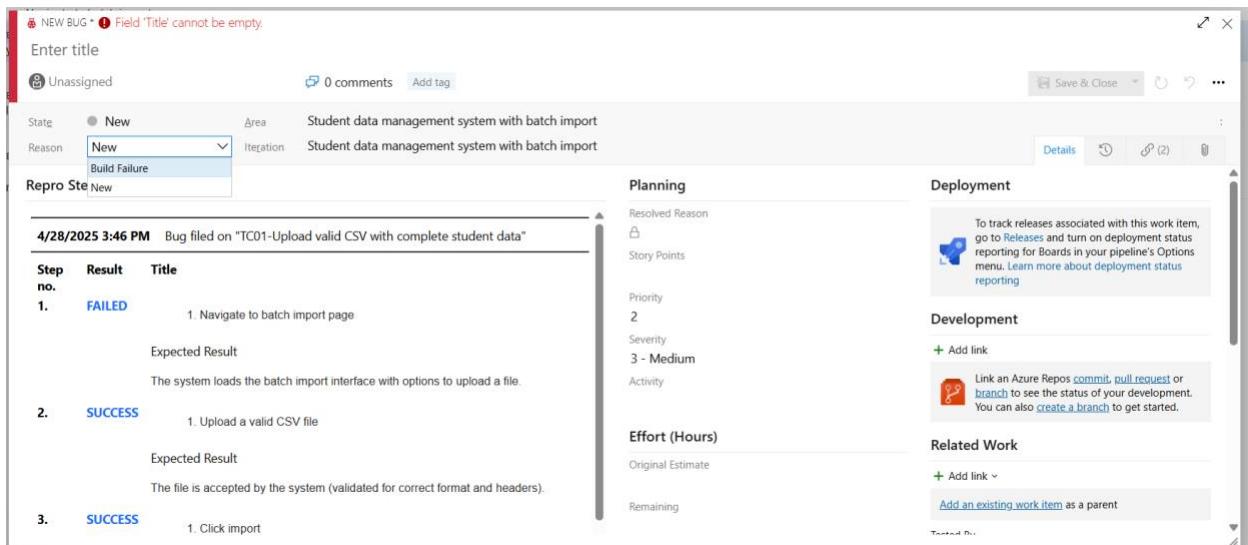
Name	Size
SystemInformation-2025-04-18T03-23-58.168Z.json	1K

8.Test case results

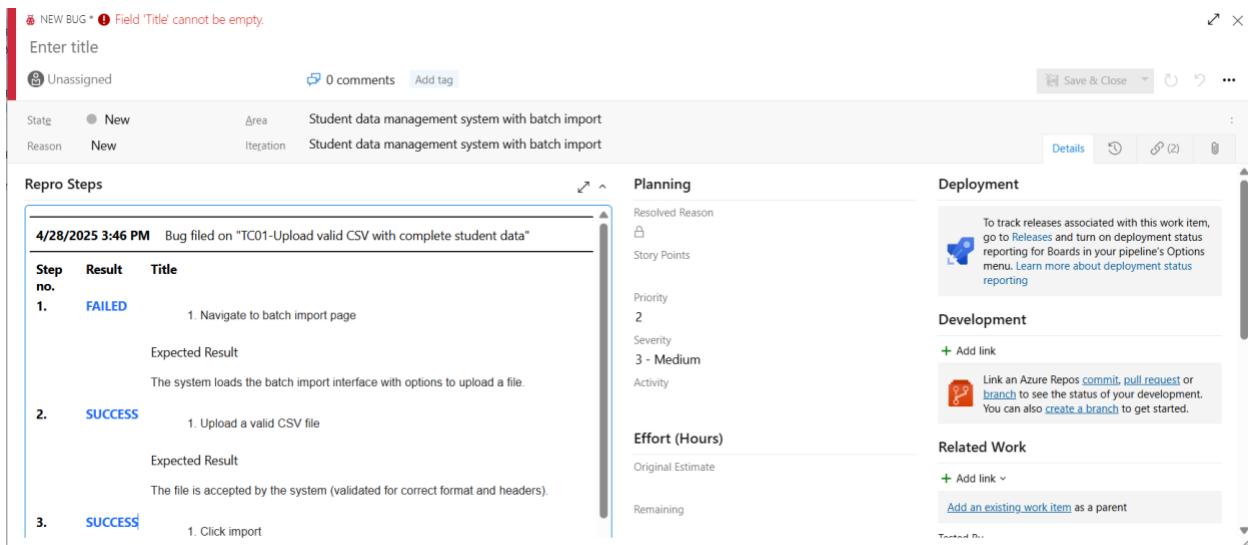
The screenshot shows the Azure DevOps interface for a test plan. The left sidebar is titled "Test Plans" and has several other options like Overview, Boards, Repos, Pipelines, Test Points, Progress report, Parameters, Configurations, Runs, and Artifacts. The main area shows a test suite titled "TS01-Batch Import Feature (ID: 72)". It includes sections for "Test Suites" (with a filter for "Student data management system..."), "Test Points (3 items)" (with a list of test cases like "TC01-Upload valid CSV with complete" and "TC02-Upload File with Missing Headers"), and a "Test Case Results" table.

Outcome	TimeStamp	Configuration	Run by	Tester	Test Pl.
Passed	Apr 21	Windows 10	Hitesh Kumar S	Hitesh Kumar S	Studer
Passed	Apr 19	Windows 10	Hemanth Kumar D	Hitesh Kumar S	Studer
Passed	Apr 15	Windows 10	Hitesh Kumar S	Hitesh Kumar S	Studer
Failed	Apr 15	Windows 10	Hitesh Kumar S	Hitesh Kumar S	Studer
Passed	Apr 15	Windows 10	Hitesh Kumar S	Hitesh Kumar S	Studer
Passed	Apr 15	Windows 10	Hitesh Kumar S	Hitesh Kumar S	Studer
Failed	Apr 15	Windows 10	Hitesh Kumar S	Hitesh Kumar S	Studer
Failed	Apr 15	Windows 10	Hitesh Kumar S	Hitesh Kumar S	Studer
Failed	Apr 15	Windows 10	Hemanth Kumar D	Hitesh Kumar S	Studer
Failed	Apr 15	Windows 10	Hemanth Kumar D	Hitesh Kumar S	Studer
Passed	Apr 15	Windows 10	Hitesh Kumar S	Hitesh Kumar S	Studer
Passed	Apr 15	Windows 10	Hitesh Kumar S	Hitesh Kumar S	Studer
Failed	Apr 15	Windows 10	Iraiyabu S T	Hitesh Kumar S	Studer
Failed	Apr 15	Windows 10	Hitesh Kumar S	Hitesh Kumar S	Studer
Passed	Apr 15	Windows 10	Hemaprasath D S	Hitesh Kumar S	Studer
Failed	Apr 15	Windows 10	Hemaprasath D S	Hitesh Kumar S	Studer

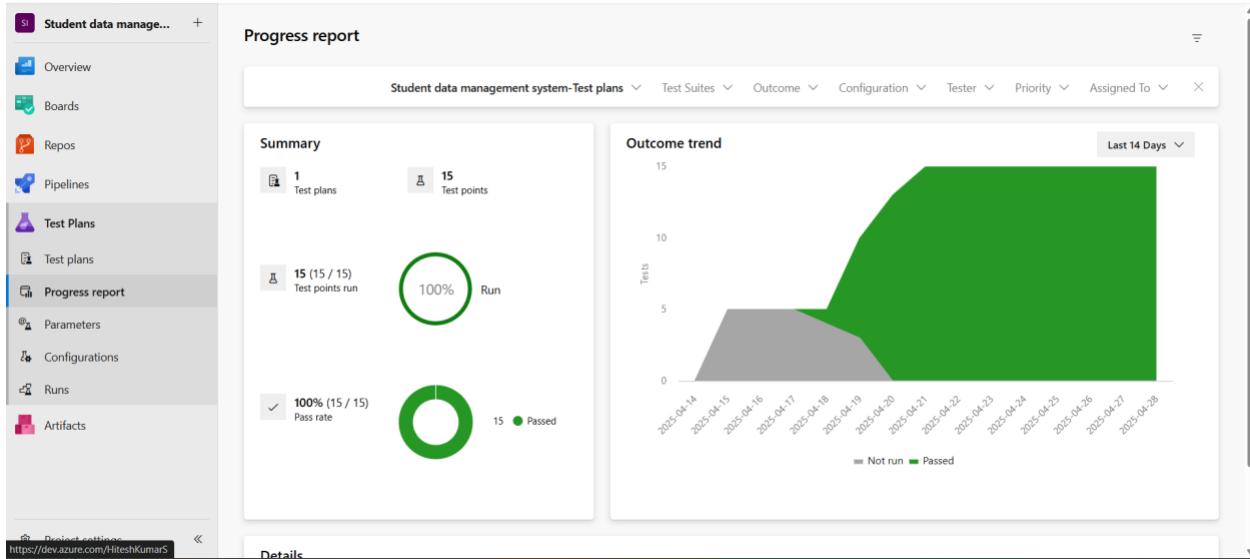
9.Test report summary



- Assigning bug to the developer and changing state



10. Progress report



11. Changing the test template

All processes

Name	Description	Team projects
Basic (default)	This template is flexible for any process and great for teams getting started with Az...	0
Agile	This template is flexible and will work great for most teams using Agile planning me...	1
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 improv...	0

Organization Settings
HiteshKumarS

All processes > Agile

Work item types Backlog levels Projects

Name	Description
Student data management system with batch import	...

General

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

Security

- Security overview
- Policies
- Permissions

Boards

- Process

<https://dev.azure.com/HiteshKumarS>

12. View the new test case template

Add a field to Test Case

New field

Definition Options Layout

Add a field to store custom, queryable data about your work items.

Use an existing field Field Acceptance Criteria

Create a field Name Type

Type Text (single line)

Description Optionally provide a description for the field

Learn more (?)

Add field Cancel

Integer

Automation status

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.

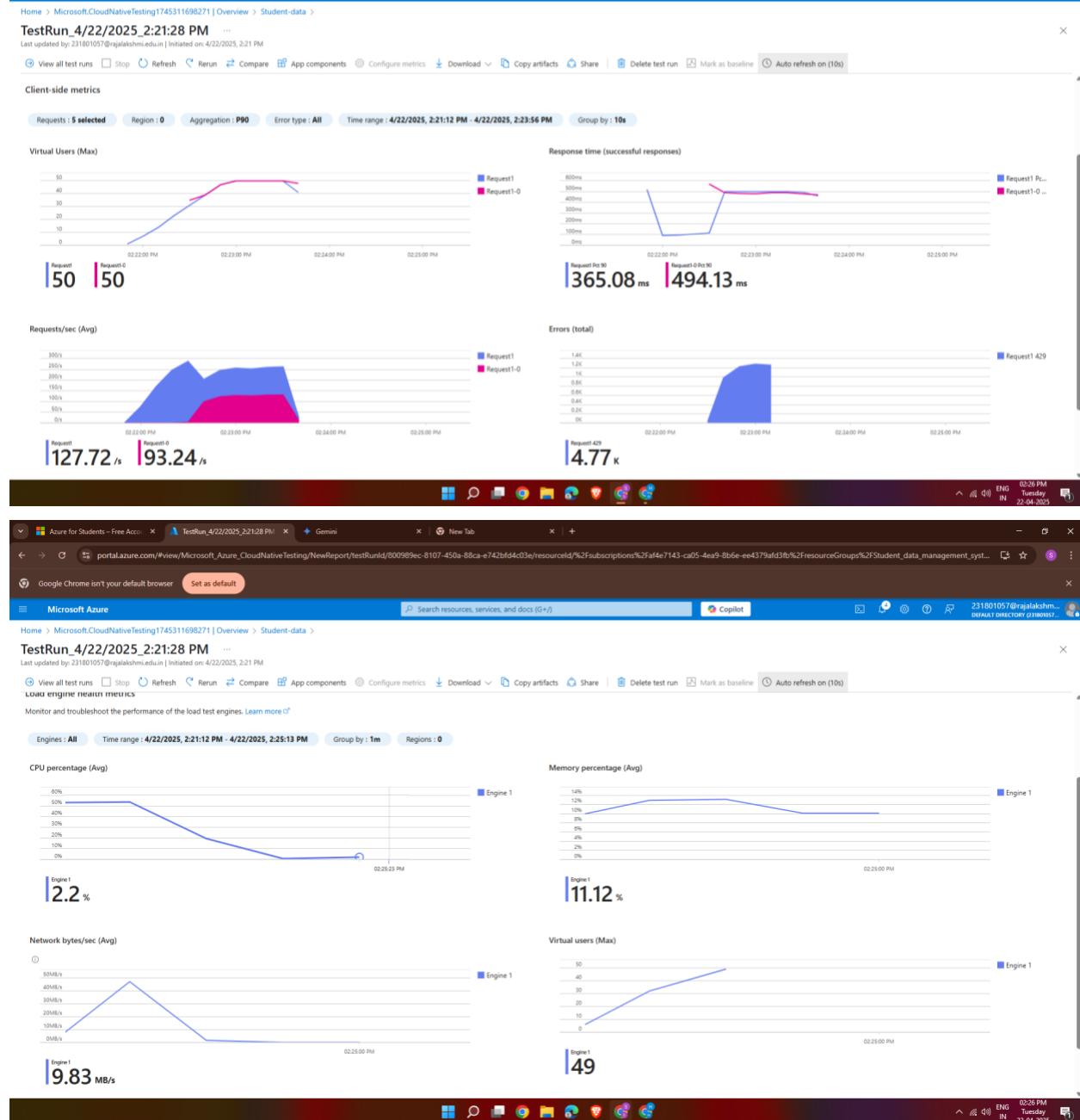
Steps to Create and Run a Load Test:

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.
- o Installing project dependencies from project/requirements.txt. Make sure the path to requirements.txt is correct (it is located under the project folder).
- o Running a simple Python script to verify that Python is set up correctly and the pipeline works.

4. Run and Monitor Pipeline:

- o Commit changes to the main branch of your repository to trigger the pipeline in Azure DevOps.
- o 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 "Iraiyanbu S T-231801061.Student_data_management_system". The left sidebar has "Pipelines" selected. The main area displays a single pipeline run titled "#20250425.1 • Student_data_management_system-Pipelines". The run status is "Succeeded" with a green checkmark. It was triggered by "Individual CI for main" at "6c9498a2" and completed on "Friday" ago. The run duration was "21s".

Result:

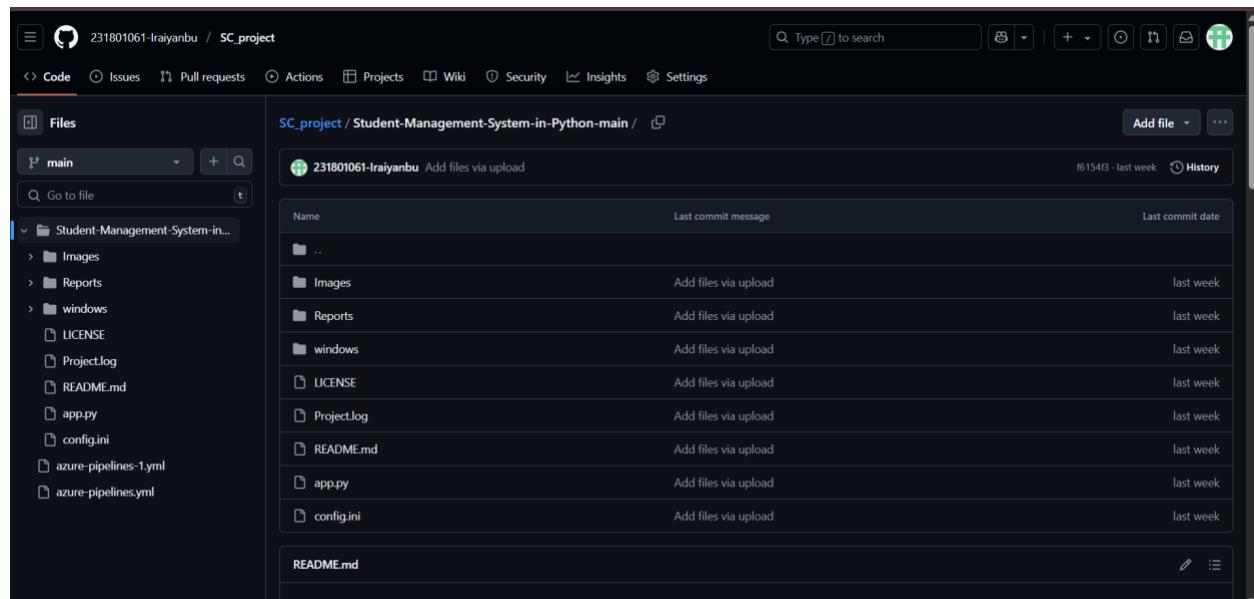
Successfully created the Azure Load Testing resource and executed a load test to assess the performance of the specified endpoint and also demonstrated pipelines in azure devops.

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 Music Playlist Batch Creator project.

GitHub Project Structure


The screenshot shows a GitHub repository named 'SC_project' belonging to user '231801061-Iraiyanbu'. The repository path is 'SC_project / Student-Management-System-in-Python-main /'. The left sidebar displays the project structure:

```

SC_project
  +-- Student-Management-System-in...
    +-- Images
    +-- Reports
    +-- windows
    +-- LICENSE
    +-- Project.log
    +-- README.md
    +-- app.py
    +-- config.ini
    +-- azure-pipelines-1.yml
    +-- azure-pipelines.yml
  
```

The main content area shows a list of files with their last commit details:

Name	Last commit message	Last commit date
..		
Images	Add files via upload	last week
Reports	Add files via upload	last week
windows	Add files via upload	last week
LICENSE	Add files via upload	last week
Project.log	Add files via upload	last week
README.md	Add files via upload	last week
app.py	Add files via upload	last week
config.ini	Add files via upload	last week
README.md		

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.