



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: Gaurav Ramasubramaniam

Register No: 231801038

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

Semester: IV

Academic Year: 2024 - 2025



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

BONAFIDE CERTIFICATE

NAME GAURAV RAMA LUBRAMANIAM

ACADEMIC YEAR 2024 - 2025 .. SEMESTER IV BRANCH.... AI.....41.DS.....

UNIVERSITY REGISTER No.

2116231801038

Certified that this is the bonafide record of work done by the above student in the
CS23432 - Software
Construction Laboratory during the year **2024 - 2025**

Signature of Faculty - in - Charge

Submitted for the Practical Examination held on.....

External Examiner

Internal Examiner



CamScanner

INDEX

Name:	GAVRAV RAMASURRAHANIAM	Branch:	AI & DS	Sec:	FA	Roll No:	231601038
S.No.	Date	Title		Page No.	Teacher's Sign/Remarks		
1.	22/1/25	Azure DevOps Environment Setup		3	<i>Done</i>		
2.	22/1/25	Azure DevOps Project Setup and User Story Management		7	<i>Done</i>		
3.	29/1/25	Setting Up Epics, Features, and User Stories for Project Planning		11	<i>Done</i>		
4.	12/2/25	Sprint Planning		14	<i>Done</i>		
5.	19/2/25	Poker Estimation		17	<i>Done</i>		
6.	26/2/25	Designing class and Sequence Diagrams For Project Architecture		18	<i>Done</i>		
7.	05/3/25	Designing Architectural and ER Diagram For Project		20	<i>Done</i>		
8.	26/3/25	Testing - Test Plans and Test Cases		22	<i>Done</i>		
9.	16/4/25	Load Testing and pipelines		27	<i>Done</i>		
10.	23/4/25	Github: project structure & Naming conventions		42	<i>Done</i>		
					<i>Completed</i>		
					<i>Completed</i>		

INDEX

S.No.	Date	Title
1.	22/1/25	Azure Devops Environment Setup.
2.	22/1/25	Azure Devops Project Setup and User Story Management.
3.	29/1/25	Setting Up Epics, Features, And User Stories for Project Planning.
4.	12/2/25	Sprint Planning.
5.	19/2/25	Poker Estimation.
6.	26/2/25	Designing Class and Sequence Diagrams for Project Architecture.
7.	05/3/25	Designing Architectural and ER Diagrams for Project Structure.
8.	26/3/25	Testing – Test Plans and Test Cases.
9.	16/4/25	Load Testing and Pipelines.
10.	23/4/25	GitHub: Project Structure & Naming Conventions.

EXP NO: 1

AZURE DEVOPS ENVIRONMENT SETUP

Aim:

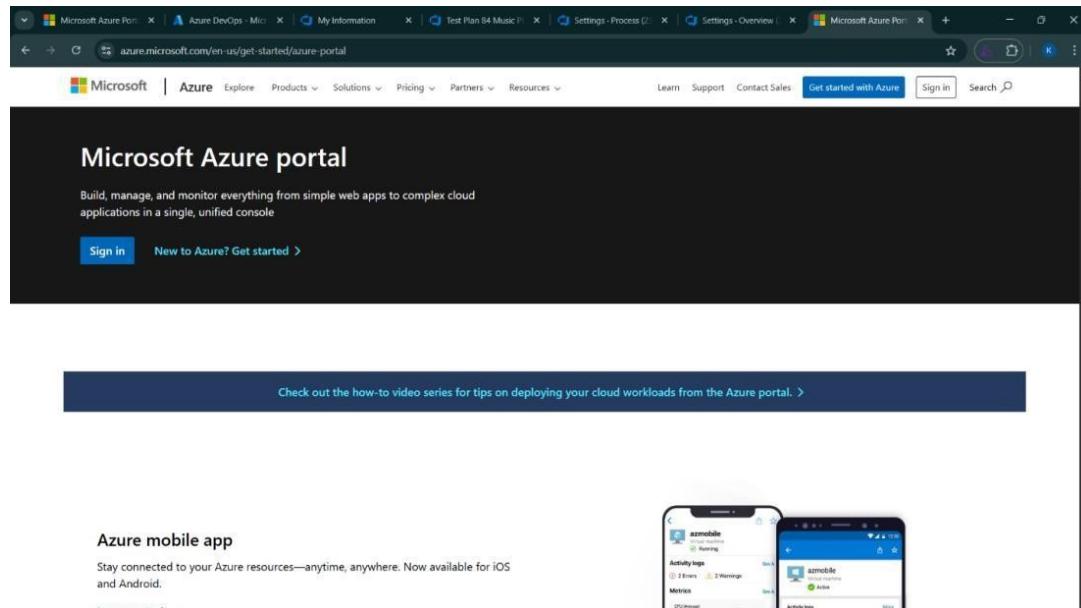
To set up and access the Azure DevOps environment by creating an organization through the Azure portal.

INSTALLATION

1. Open your web browser and go to the Azure website: <https://azure.microsoft.com/en-us/get-started/azure-portal>.

Sign in using your Microsoft account credentials.

If you don't have a Microsoft account, you can create one here: <https://signup.live.com/?lic=1>



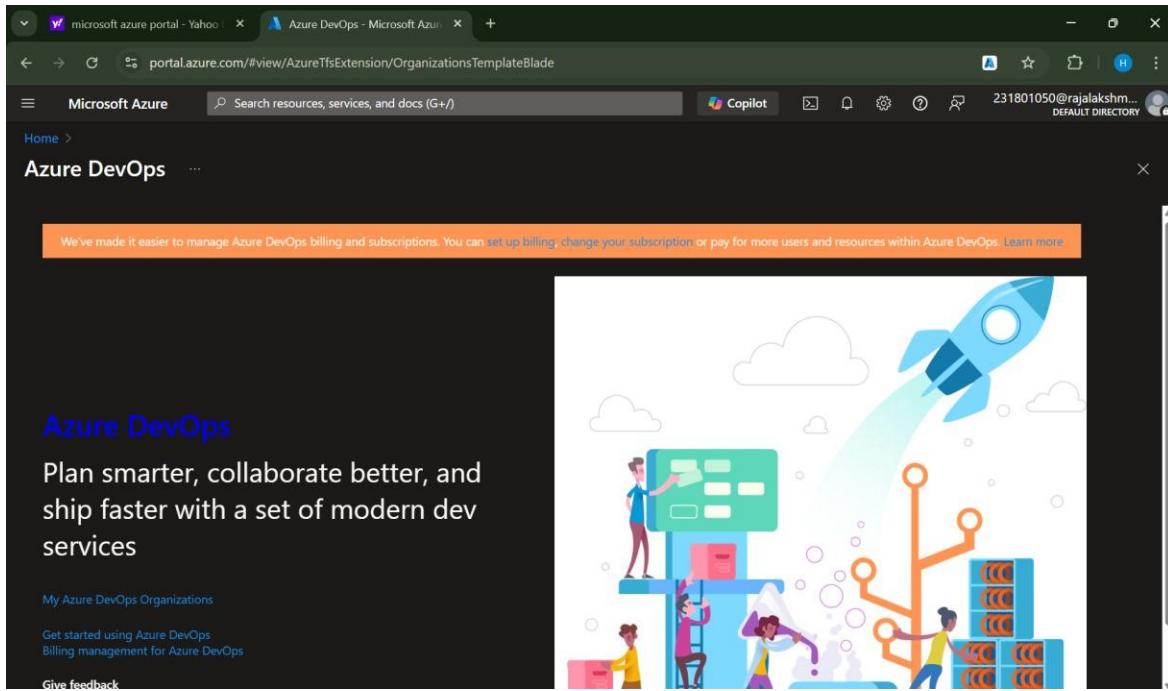
2. Azure home page

The screenshot shows the Microsoft Azure home page. At the top, there's a navigation bar with tabs for 'microsoft azure portal - Yahoo' and 'My Dashboard - Microsoft Azure'. Below the bar is a search bar with placeholder text 'Search resources, services, and docs (G+/-)'. To the right of the search bar are icons for Copilot, notifications, settings, and user profile. The main content area has a dark background. On the left, there's a sidebar with 'Azure services' and 'Resources' sections. Under 'Azure services', there are icons for 'Create a resource', 'Azure DevOps organizations', 'Quickstart Center', 'Azure AI services', 'Kubernetes services', 'Virtual machines', 'App Services', 'Storage accounts', 'SQL databases', and 'More services'. Under 'Resources', there are tabs for 'Recent' (which is selected) and 'Favorite'. A table below shows columns for 'Name', 'Type', and 'Last Viewed'. A message 'No resources have been viewed recently' is displayed, along with a 'View all resources' button. At the bottom, there's a 'Navigate' section with links for 'Subscriptions', 'Resource groups', 'All resources', and 'Dashboard'.

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, showing the Microsoft Azure home page. However, the search bar now contains the text 'Devops'. The rest of the interface, including the sidebar, resource table, and navigation links at the bottom, remains the same.

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



Result:

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

2116231801038



CamScanner
CS23432

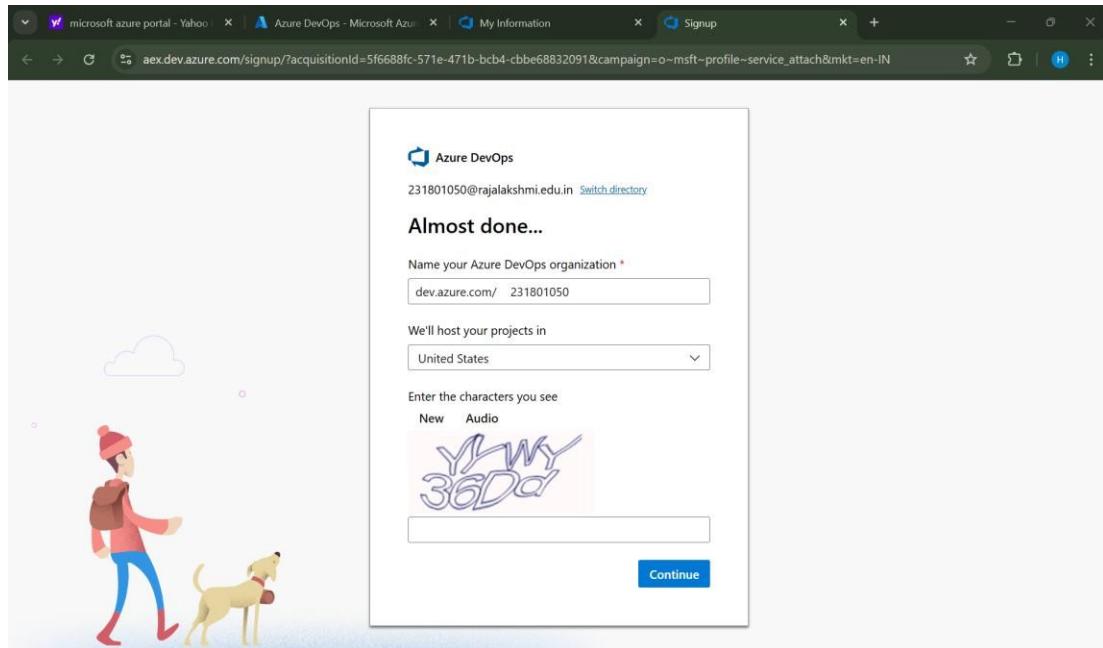
EXP NO: 2

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

Cancel Create

- 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's a profile section for 'Harish Tutu YT' with a purple circular icon containing 'HY'. Below it, there's contact information: '231801050@rajalakshmi.edu.in', 'Microsoft account', 'India', and an email address '231801050@rajalakshmi.edu.in'. A 'Visual Studio Dev Essentials' card is also present. The main right-hand panel displays the 'Azure DevOps Organizations' header with a 'Create new organization' button. Under 'Projects', the 'TO-DO LIST APP WITH REMINDERS' project is listed with an 'Actions' link to 'Open in Visual Studio'. There's also a 'New project' button.

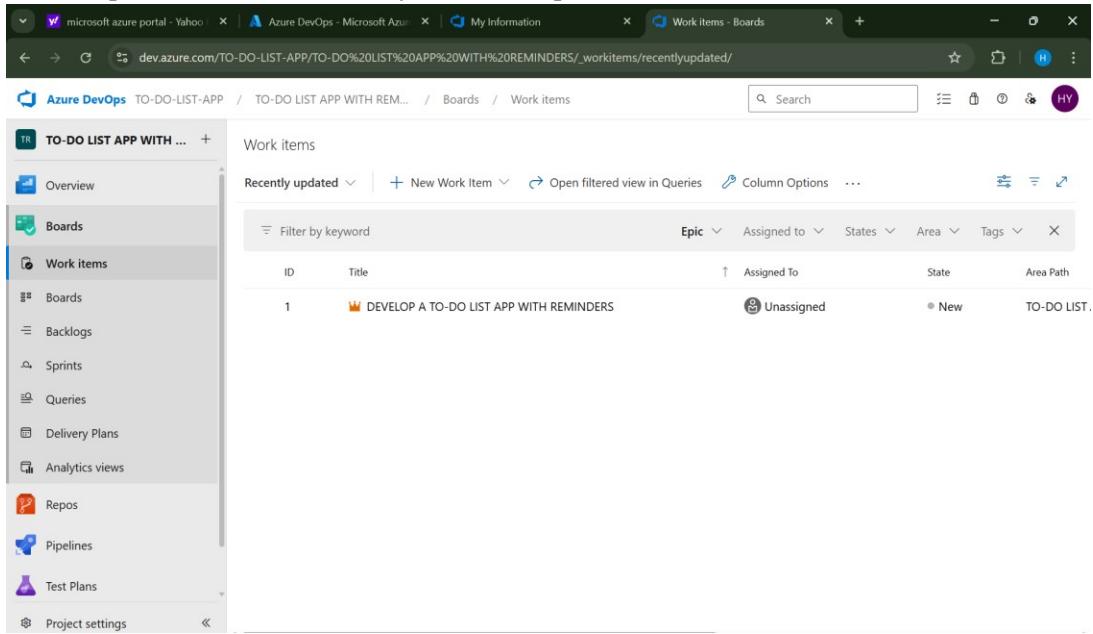
4. Project dashboard

The screenshot shows the 'Summary - Overview' page for the 'TO-DO LIST APP WITH REMINDERS' project. The left sidebar includes links for Overview, Summary, Dashboards, Wiki, Boards, Repos, Pipelines, Test Plans, Artifacts, and Project settings. The main content area features a title 'TO-DO LIST APP WITH REMINDERS' with 'Public' and 'Invite' buttons. Below this is the 'About this project' section, which describes the app as a cloud-based To-Do List App with reminders using Azure services like App Service, Functions, SQL Database, and Notification Hubs. The 'Project stats' section shows '12 Work items created' and '0 Work items completed' over the last 7 days. The 'Members' section lists five team members with their initials: DR, HY, GR, DS, and DK, each represented by a colored circular icon.

5. To manage user stories:

a. From the **left-hand navigation menu**, click on **Boards**. This will take you to the main **Boards** page, where you can manage work items, backlogs, and sprints.

b. On the **work items** page, you'll see the option to **Add a work item** at the top. Alternatively, you can find a + button or **Add New Work Item** depending on the view you're in. From the **Add a work item** dropdown, select **User Story**. This will open a form to enter details for the new User Story.



The screenshot shows the Azure DevOps interface for a project titled "TO-DO LIST APP WITH REMINDERS". The left sidebar is the navigation menu, with "Work items" currently selected. The main area is titled "Work items" and shows a table of one item. The table has columns for ID, Title, Assigned To, State, and Area Path. The single item listed is ID 1, titled "DEVELOP A TO-DO LIST APP WITH REMINDERS", assigned to "Unassigned", in "New" state, and under the "TO-DO LIST." area path. There are buttons for "New Work Item", "Open filtered view in Queries", and "Column Options".

ID	Title	Assigned To	State	Area Path
1	DEVELOP A TO-DO LIST APP WITH REMINDERS	Unassigned	New	TO-DO LIST.

Result:

Successfully created an Azure DevOps project with user story management and agile workflow setup.

2116231801038

EXP NO: 3

SETTING UP EPICS, FEATURES, AND USER STORIES FOR PROJECT PLANNING

Aim:

To learn about how to create epics, user story, features, backlogs for your assigned project.

Create Epic, Features, User Stories, Task

The screenshot shows the Azure DevOps interface for a project named 'TO-DO LIST APP WITH REMINDERS'. The left sidebar is the navigation menu, and the main area is the 'Work items' section. The table lists five work items:

ID	Title	Assigned To	State	Area Path
1	DEVELOP A TO-DO LIST APP WITH REMINDERS	Unassigned	New	TO-DO LIST
4	Reminder Notifications	Unassigned	New	TO-DO LIST
5	Task Categorization & Priority Levels	Unassigned	New	TO-DO LIST
3	Task Creation, Editing, and Deletion	Unassigned	New	TO-DO LIST
2	User Authentication & Profile Management	Unassigned	New	TO-DO LIST

1. Fill in Epics

The screenshot shows the details of an epic named 'EPIC 1' for the project 'TO-DO LIST APP WITH REMINDERS'. The epic has one child item: '1 DEVELOP A TO-DO LIST APP WITH REMINDERS'. The epic details include:

- Description:** This epic covers the development of a to-do list application with task management, reminders, and notifications.
- Planning:** Priority: 2, Risk: 1, Effort: 10, Business Value: 10, Time Criticality: 1.
- 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.

2. Fill in Features

The screenshot shows the Azure DevOps interface for a work item titled "FEATURE 4". The left sidebar is the navigation menu. The main area displays the work item details:

Description:
Description: Users should receive timely reminders for their pending tasks.
Key Functionalities:

- Set reminders for tasks.
- Receive push notifications before the task deadline.
- Snooze or dismiss reminders.

Planning:
Priority: 2
Risk: 1
Effort: 1
Business Value: 1
Time Criticality: 1

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

3. Fill in User Story Details

The screenshot shows the Azure DevOps interface for a work item titled "USER STORY 9". The left sidebar is the navigation menu. The main area displays the work item details:

Description:
As a user, I want to create a new task so that I can keep track of my work.
No one selected
0 Comments Add Tag

Planning:
Story Points: 5
Priority: 2
Risk: 1

Acceptance Criteria:

- Users can enter a task title and description.
- Users can set a due date.
- Task should be saved and displayed in the task list.

Classification:
Value area: Business

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

Result:

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

EXP NO: 4

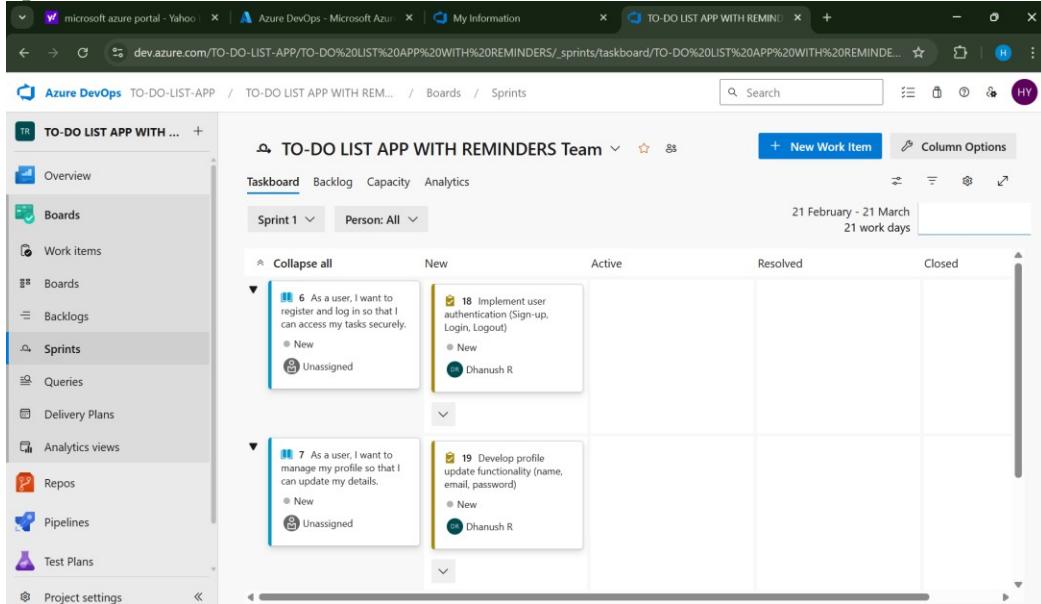
SPRINT PLANNING

Aim:

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

Sprint Planning

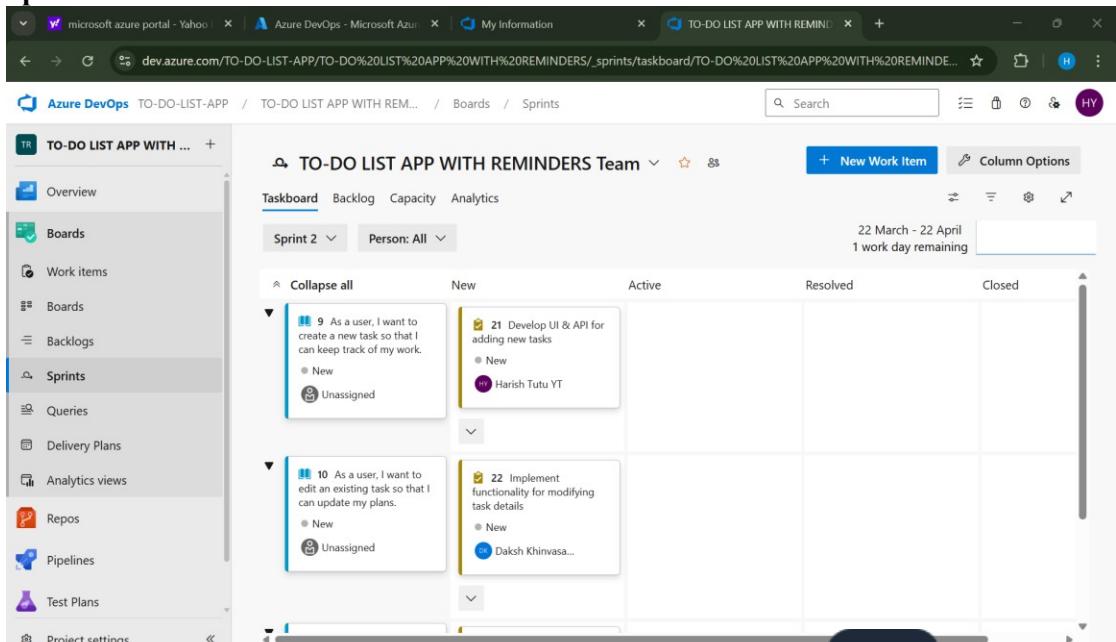
Sprint 1



The screenshot shows the Azure DevOps Taskboard for the 'TO-DO LIST APP WITH REMINDERS' project. The left sidebar is expanded, showing 'Boards' selected under 'Sprints'. The main area displays the 'Taskboard' for 'Sprint 1' from February 21 to March 21. There are four cards visible:

- User Story 6:** As a user, I want to register and log in so that I can access my tasks securely.
 - Status: New
 - Assignee: Unassigned
- User Story 18:** Implement user authentication (Sign-up, Login, Logout).
 - Status: New
 - Assignee: Dhanush R
- User Story 7:** As a user, I want to manage my profile so that I can update my details.
 - Status: New
 - Assignee: Unassigned
- User Story 19:** Develop profile update functionality (name, email, password).
 - Status: New
 - Assignee: Dhanush R

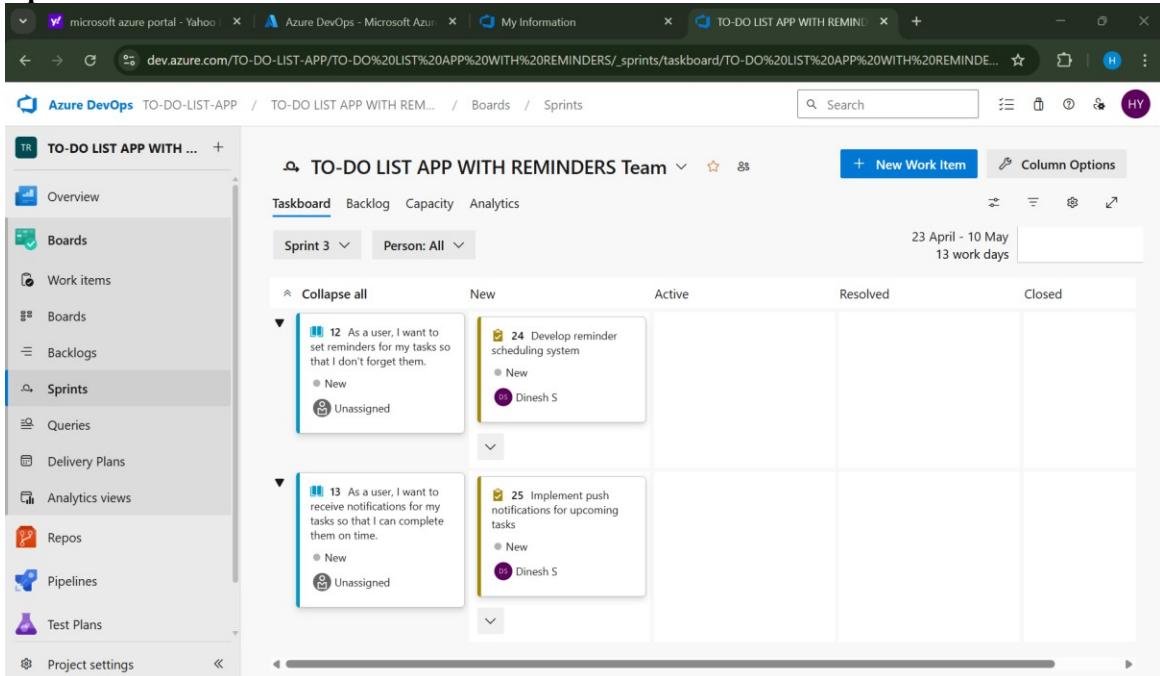
Sprint 2



The screenshot shows the Azure DevOps Taskboard for the 'TO-DO LIST APP WITH REMINDERS' project. The left sidebar is expanded, showing 'Boards' selected under 'Sprints'. The main area displays the 'Taskboard' for 'Sprint 2' from March 22 to April 22. There are four cards visible:

- User Story 9:** As a user, I want to create a new task so that I can keep track of my work.
 - Status: New
 - Assignee: Unassigned
- User Story 21:** Develop UI & API for adding new tasks.
 - Status: New
 - Assignee: Harish Tutu YT
- User Story 10:** As a user, I want to edit an existing task so that I can update my plans.
 - Status: New
 - Assignee: Unassigned
- User Story 22:** Implement functionality for modifying task details.
 - Status: New
 - Assignee: Daksh Khinvasa...

Sprint 3

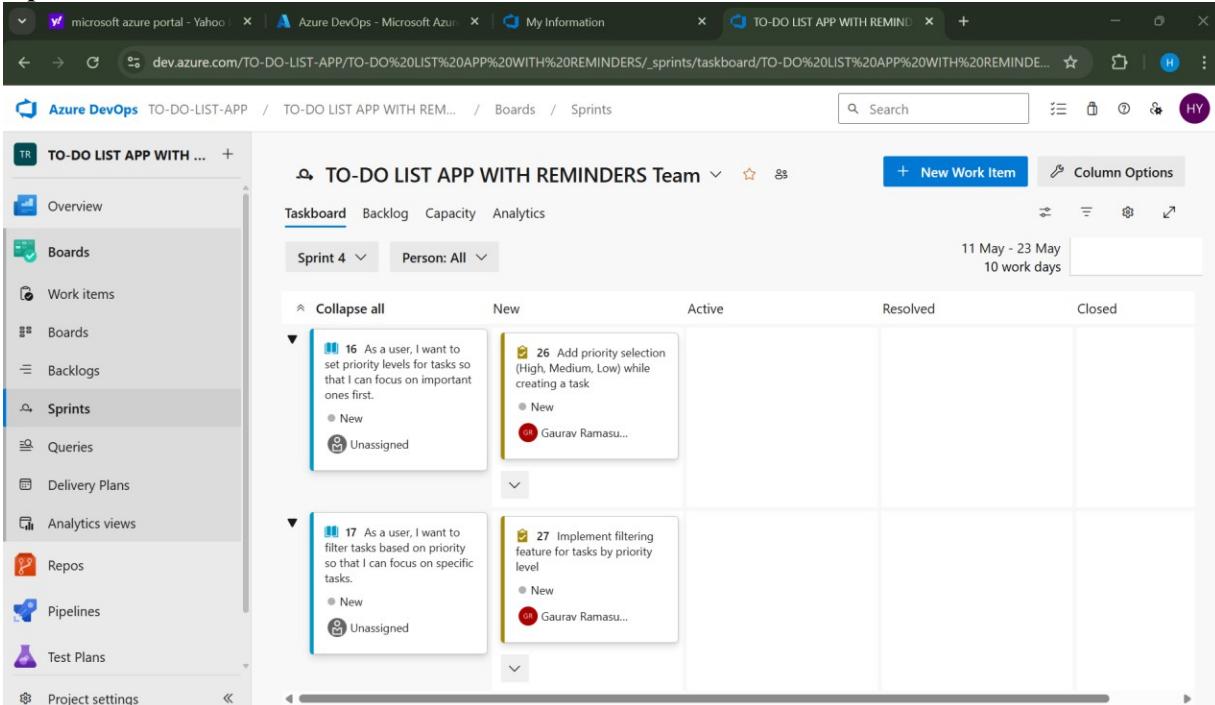


The screenshot shows the Azure DevOps Taskboard for the 'TO-DO LIST APP WITH REMINDERS Team' in 'Sprint 3'. The taskboard displays four items:

- 12** As a user, I want to set reminders for my tasks so that I don't forget them.
 - New
 - Unassigned
- 24** Develop reminder scheduling system
 - New
 - Dinesh S
- 13** As a user, I want to receive notifications for my tasks so that I can complete them on time.
 - New
 - Unassigned
- 25** Implement push notifications for upcoming tasks
 - New
 - Dinesh S

The taskboard includes filters for 'Sprint 3' and 'Person: All'. The timeline shows '23 April - 10 May' with '13 work days'.

Sprint 4



The screenshot shows the Azure DevOps Taskboard for the 'TO-DO LIST APP WITH REMINDERS Team' in 'Sprint 4'. The taskboard displays four items:

- 16** As a user, I want to set priority levels for tasks so that I can focus on important ones first.
 - New
 - Unassigned
- 26** Add priority selection (High, Medium, Low) while creating a task
 - New
 - Gaurav Ramas... (red)
- 17** As a user, I want to filter tasks based on priority so that I can focus on specific tasks.
 - New
 - Unassigned
- 27** Implement filtering feature for tasks by priority level
 - New
 - Gaurav Ramas... (red)

The taskboard includes filters for 'Sprint 4' and 'Person: All'. The timeline shows '11 May - 23 May' with '10 work days'.

Result:

The Sprints are created for the To-do list app with reminders project.

2116231801038

EXP NO: 5

POKER ESTIMATION

Aim:

Create Poker Estimation for the user stories - To-do list app with reminders project.

Poker Estimation

USER STORY 6

6 As a user, I want to register and log in so that I can access my tasks securely.

No one selected

0 Comments Add Tag

Save and Close Follow

Updated by Harish Tutu YT: Feb 21

Description

Planning

Deployment

Acceptance Criteria

- Users can sign up with an email and password.
- Users can log in using valid credentials.
- Users are redirected to the task dashboard upon successful login.

Classification

Development

Discussion

Add link

Project settings

Result:

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

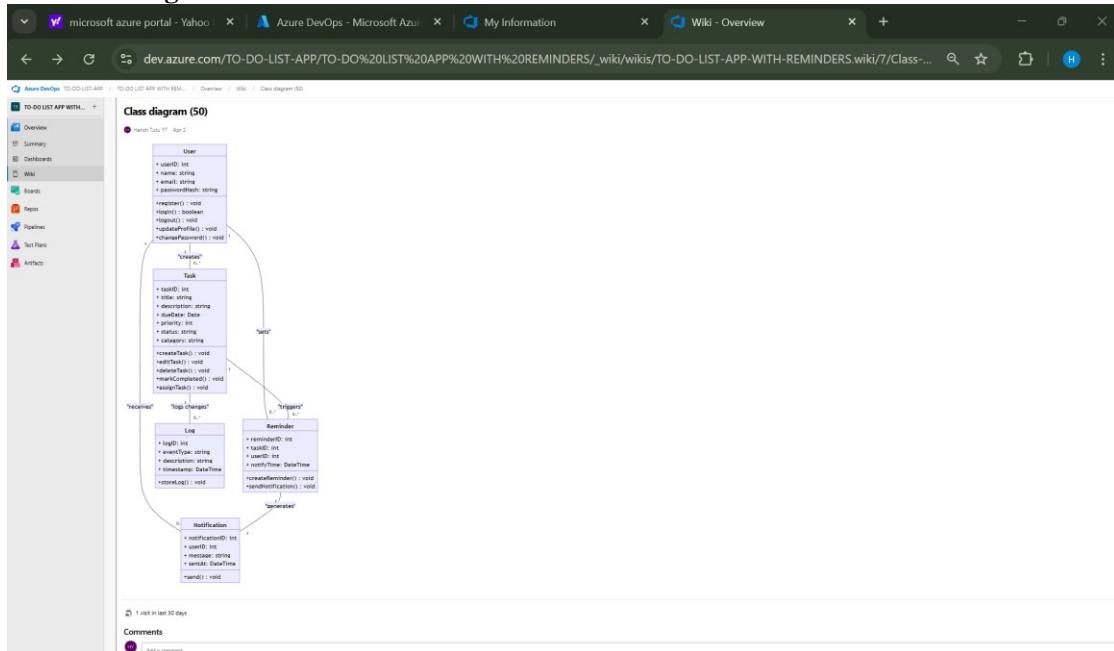
EXP NO: 6

DESIGNING CLASS AND SEQUENCE DIAGRAMS FOR PROJECT ARCHITECTURE

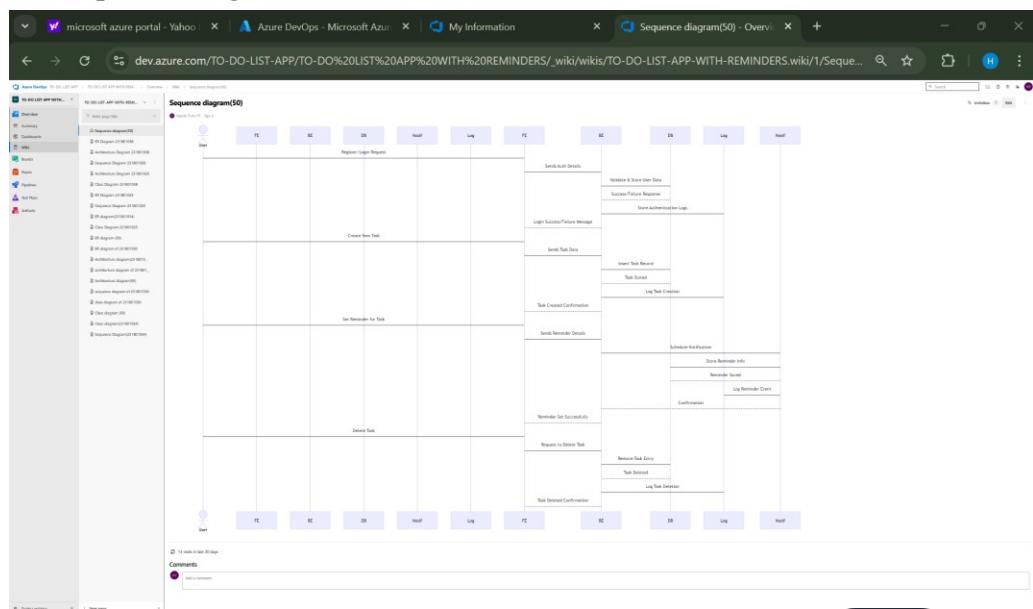
Aim:

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 To-do list app with reminders project.

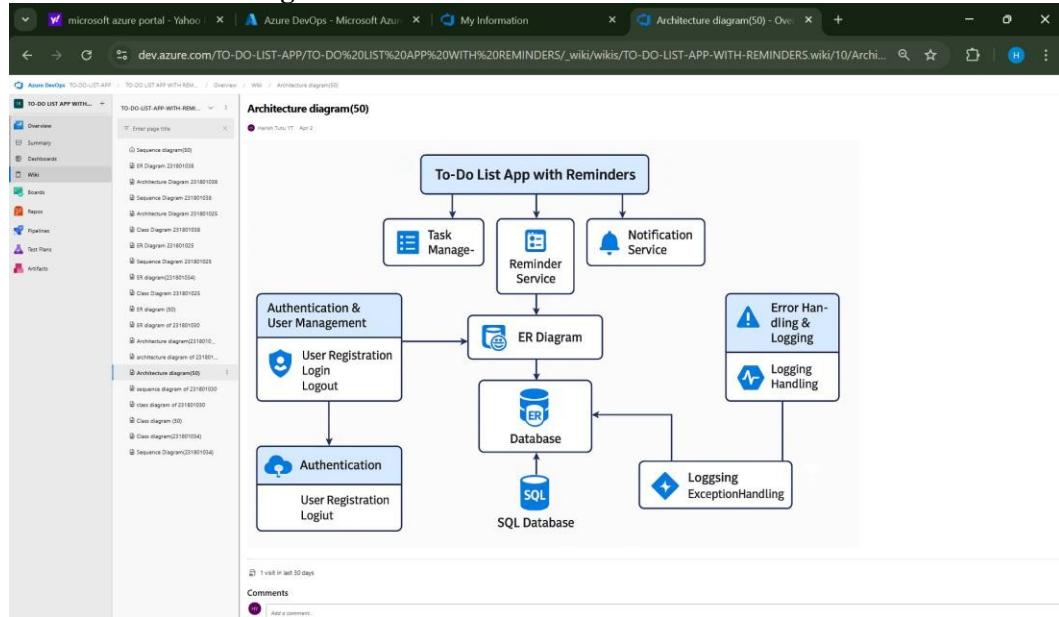
EXP NO: 7

DESIGNING ARCHITECTURAL AND ER DIAGRAMS FOR PROJECT STRUCTURE

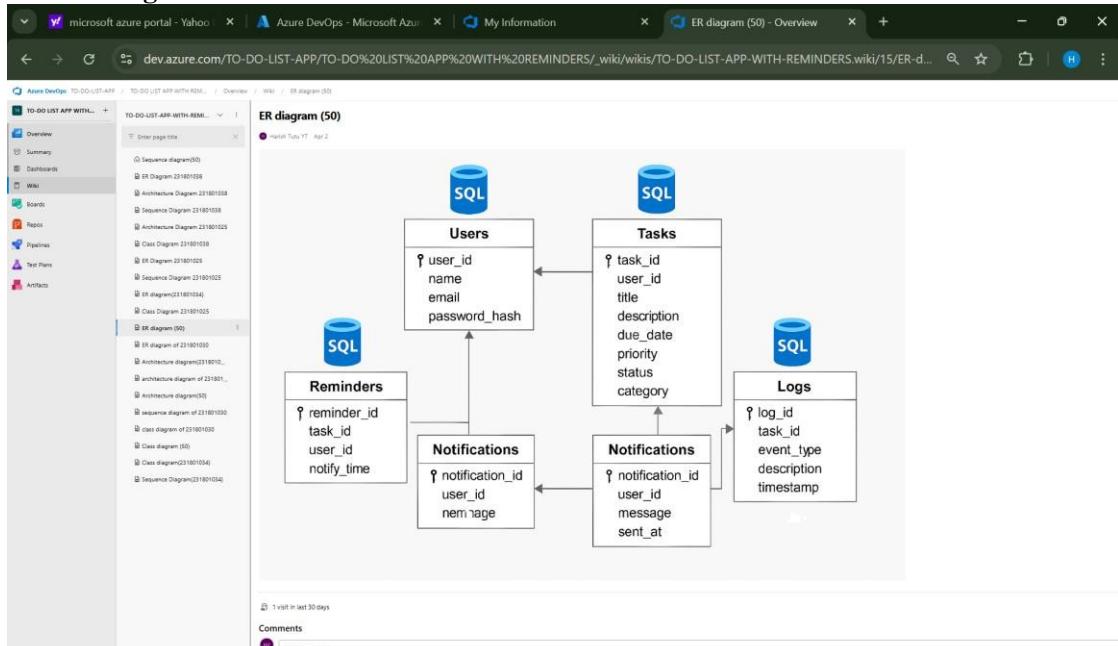
Aim:

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 To-do list app with reminders project.

2116231801038

 CamScanner
CS23432

EXP NO: 8

TESTING – TEST PLANS AND TEST CASES

Aim:

To give test cases for the To-Do List App showcasing both the happy path (expected scenarios) and error path (unexpected scenarios).

Test Planning and Test Case

Test Case Design Procedure

1. Understand Core Features of the Application

- User signup, login, logout, and profile management
- Creating, editing, and deleting tasks
- Setting task reminders and receiving notifications
- Setting and filtering task priorities

2. Define User Interactions

- Each test case is based on real user actions like registering, creating tasks, setting reminders, etc.

3. Design Happy Path Test Cases

- These validate that all core functionalities work as expected under normal conditions..

4. Design Error Path Test Cases

- Simulate negative or unexpected behavior like login failures or invalid inputs.

5. Break Down Steps and Expected Results

- Each test case includes step-by-step user actions and the expected outcome

6. Use Clear Naming and IDs

- Example: TC01-Successful-Login, TC05-Task-Reminder-Failure

7. Separate Test Suites

- Group test cases based on modules: Authentication, Task Management, Reminders, Priority Handling, Security.

8. Prioritize and Review

- High-priority test cases are assigned to core features like login, task creation, and reminder notifications.

1. New test plan

The screenshot shows the 'Test Plan' page for the 'TO-DO LIST APP'. The main header displays the project name and the specific test plan ID (28). The left sidebar includes links for Overview, Boards, Repos, Pipelines, Test Plans, and Artifacts. The right panel contains sections for 'Description' (with a placeholder 'Click to add Description.') and 'Timelines' (showing Start Date: 15-04-2025 15:09 and Finish Date: 22-04-2025 15:09). A 'Discussion' section is present with a comment input field. At the bottom, there are buttons for 'Save and Close', 'Follow', and 'Details'.

2. Test suite

The screenshot shows the 'Test Suites' management interface for the 'TO-DO LIST APP'. The left sidebar lists 'Overview', 'Boards', 'Repos', 'Pipelines', 'Test Plans', 'Test suites', 'Progress report', 'Parameters', 'Configurations', 'Runs', and 'Artifacts'. The 'Test suites' section is currently selected. The main area displays a 'Task management (ID: 31)' card with tabs for 'Define', 'Execute', and 'Chart'. Under the 'Test Cases' section, a table lists four items: 'Create Task' (Order 1, Test 34), 'Edit Task' (Order 2, Test 36), and 'Delete Task' (Order 3, Test 39). A context menu is open over the 'New Suite' option, showing options like 'Static suite', 'Requirement based suite', and 'Query based suite'.

3. Test case

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

To-Do List App – Test Plans

Test Suites

Test Suit: TS01 - User Authentication

1. TC01 – User Registration & Login

Action 1: Open the app and click “Register”.

Expected Result: Registration form appears.

Action 2: Enter valid name, email, and password, then click “Submit”.

Expected Result: Account created, redirected to login page.

Action 3: Enter credentials and click “Login”.

Expected Result: Dashboard loads, user successfully logged in.

2. TC02 – Manage Profile

Action 1: Click on “Profile” in the navigation bar.

Expected Result: Profile details are displayed.

Action 2: Click “Edit”, change user info (e.g., name).

Expected Result: Editable fields appear.

Action 3: Click “Save Changes”.

Expected Result: Profile updated confirmation appears.

3. TC03 – Logout

Action 1: Click on “Logout” button from dashboard.

Expected Result: Logout confirmation popup appears.

2116231801038

Action 2: Confirm logout.

Expected Result: User is logged out.

Action 3: Try accessing dashboard URL directly.

Expected Result: Redirected to login page.

Test Suit: TS02 – Task management

1. TC04 – Create Task

Action 1: Click “Add New Task”.

Expected Result: Task creation form opens.

Action 2: Fill in task title, due date, priority, and click “Save”.

Expected Result: Task added to list.

Action 3: View the dashboard.

Expected Result: Newly created task appears in “Upcoming Tasks”

2. TC05 – Edit Task

Action 1: Click the “Edit” icon next to a task.

Expected Result: Task fields become editable.

Action 2: Change title or date.

Expected Result: Fields accept new values.

Action 3: Click “Update”.

Expected Result: Task details are updated in task list.

3. TC06 – Delete Task

Action 1: Click on the “Delete” button of a task.

Expected Result: Confirmation dialog appears.

Action 2: Click “Yes, Delete”.

Expected Result: Task is removed from the list.

Action 3: Refresh the page.

Expected Result: Task no longer appears.

Test Suit: TS03 – Reminder & notification

1. TC07 – Set Reminders

Action 1: Click “Set Reminder” on a task.

Expected Result: Date and time input appears.

Action 2: Set future time and save.

Expected Result: Reminder is saved.

Action 3: Wait till reminder time.

Expected Result: Notification or alert is triggered.

2. TC08 – Notifications

Action 1: Ensure a task has a reminder set.

Expected Result: Reminder time shows in UI.

Action 2: Wait until the task time arrives.

Expected Result: App shows a popup or notification.

Action 3: Click the notification.

Expected Result: User navigates to the task.

3. TC09 – Set Task Priority

Action 1: Click “New Task”, set priority as “High”.

Expected Result: Priority dropdown is available and saves input.

Action 2: Save the task.

Expected Result: Priority level shows in task card.

Action 3: Hover or click task for details.

Expected Result: Priority tag (e.g., red for high) is visible.

4. TC10 – Filter Tasks by Priority

Action 1: Click “Filter” and choose “High Priority”.

Expected Result: Filter activates.

Action 2: View task list.

Expected Result: Only high-priority tasks are visible.

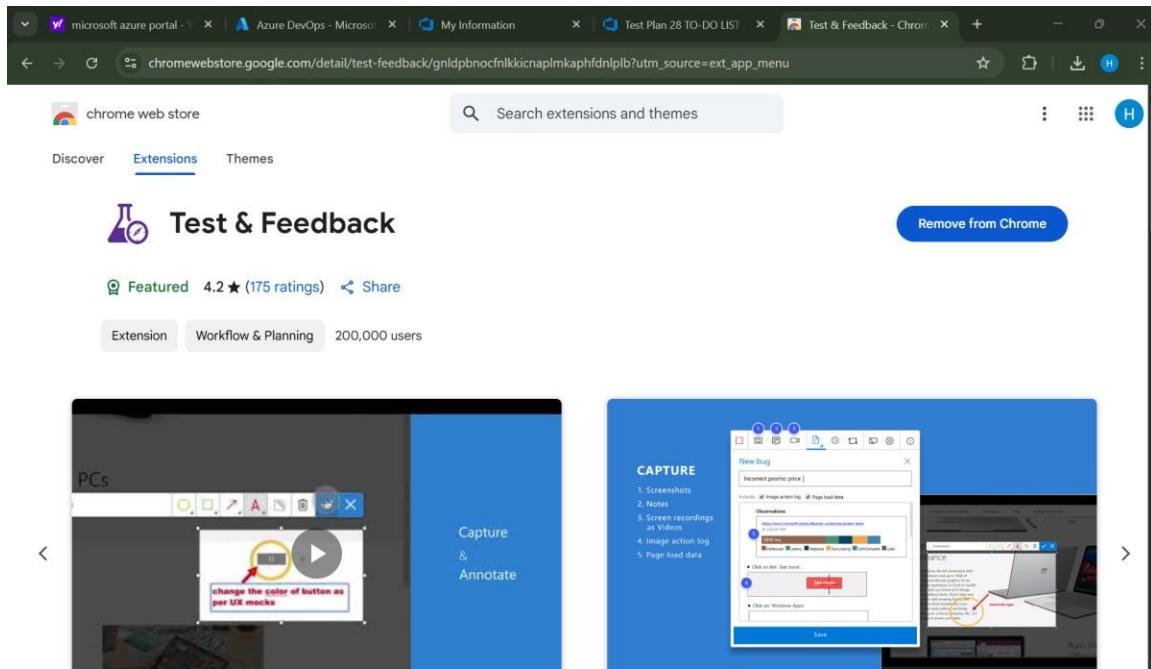
Action 3: Remove filter.

Expected Result: All tasks become visible again.

Test Cases

The screenshot shows the Azure DevOps Test Plan interface. The URL in the browser is https://dev.azure.com/TO-DO-LIST-APP/TO-DO%20LIST%20APP%20WITH%20REMINDERS/_testPlans/define?planId=28&suiteId=30. The page displays a test case titled "33 User Registration & Login" created by Dinesh S. The test case is in the "Design" state and is associated with the "TO-DO LIST APP WITH REMINDERS" area and iteration. The "Steps" tab is selected, showing three steps: 1. Open the app and click "Register". Expected result: Registration form appears. 2. Enter valid name, email, and password, then click "Submit". Expected result: Account created, redirected to dashboard. 3. Enter credentials and click "Login". Expected result: Dashboard loads, user success. The "Deployment" and "Development" sections are also visible.

4. Installation of test



Test and feedback

Showing it as an extension

The screenshot shows the Azure DevOps Test Plan interface. On the left, there's a sidebar with various icons for Test Cases, Test Suites, Pipelines, and Artifacts. The main area displays a test case titled "33 User Registration & Login". The "Steps" section contains three steps: 1. Open the app and click "Register". Expected result: Registration form appears. 2. Enter valid name, email, and password, then click "Submit". Expected result: Account created, redirected to dashboard. 3. Enter credentials and click "Login". Expected result: Dashboard loads, user successfully logged in. Below the steps is a note: "Click or type here to add a step". At the bottom, there's a "Parameter values" section. On the right, a modal window titled "Extensions" is open, showing a list of extensions with "Full access": "Allow Copy - Select & En...", "Blend & Run: Enable copy...", "Test & Feedback" (which is selected and highlighted in purple), and "Manage extensions". Below this, there's a note about tracking releases and deployment status reporting. Under "Development", there's a section for linking to Azure Repos.

5. Running the test cases

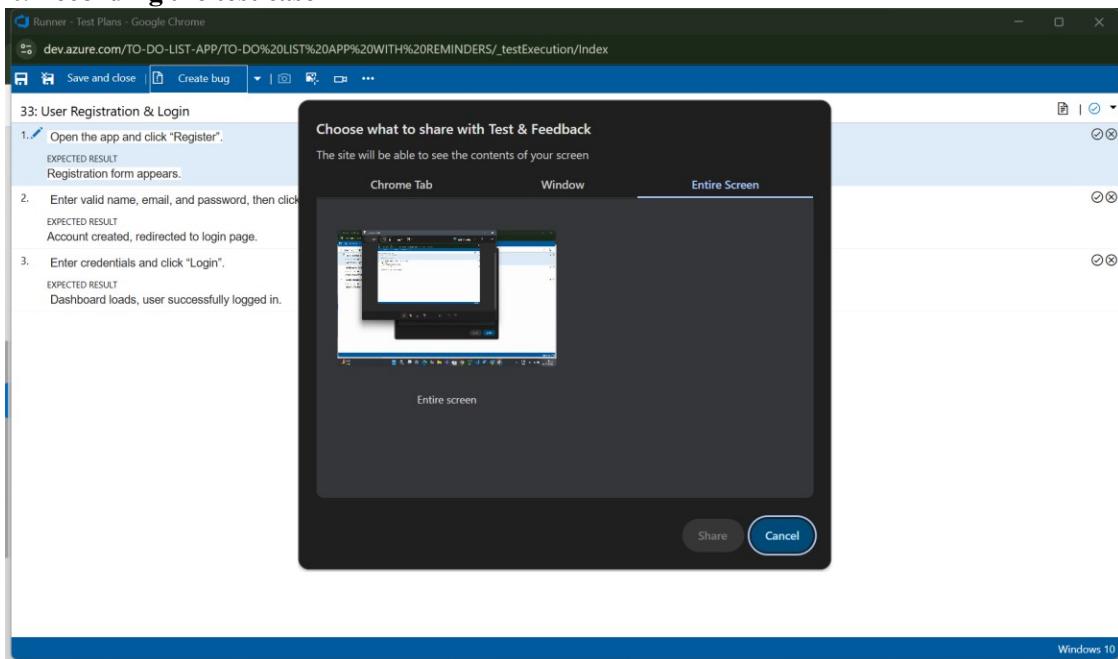
The screenshot shows the Azure DevOps Test Plan interface for a project named "TO-DO LIST APP". The left sidebar shows navigation options like Overview, Boards, Repos, Pipelines, Test Plans, Progress report, Parameters, Configurations, Runs, and Artifacts. Under "Test Plans", "Test plans" is selected. The main area displays a "Test Suites" section for "TO-DO LIST APP" with "Authentication (3)" expanded. A specific test point, "User Registration & Login", is selected and shown in detail. The "Execute" tab is active, showing the outcome as "Passed". Other options available for this test point include "Run for web application", "Run for desktop application", and "Run with options".

6.

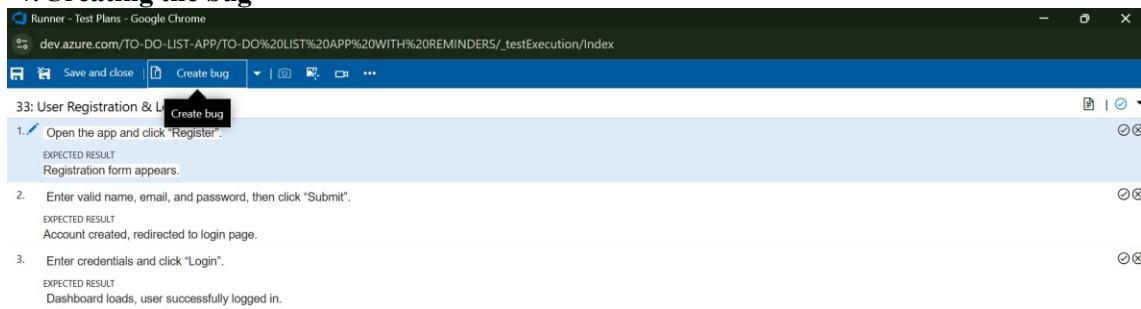
The screenshot shows the Azure DevOps Test Runner interface for a test case titled "33: User Registration & Login". The test steps are listed as follows:

1. Open the app and click "Register".
EXPECTED RESULT
Registration form appears.
2. Enter valid name, email, and password, then click "Submit".
EXPECTED RESULT
Account created, redirected to login page.
3. Enter credentials and click "Login".
EXPECTED RESULT
Dashboard loads, user successfully logged in.

6. Recording the test case



7. Creating the bug



Runner - Test Plans - Google Chrome
dev.azure.com/TO-DO-LIST-APP/TO-DO%20LIST%20APP%20WITH%20REMINDERS/_testExecution/Index

Save and close | Create bug | ...

33: User Registration & Login

1. Op... NEW BUG *

EXPE...
Reg...

2. Ent... Unassigned 0 comments Add tag

Save & Close | ...

3. Ent... State: New Area: TO-DO LIST APP WITH REMINDERS
Reason: New Iteration: TO-DO LIST APP WITH REMINDERS

Repro Steps

22-04-2025 14:41 Bug filed on "User Registration & Login"

Step no. Result Title

1. None Open the app and click "Register".
Expected Result
Registration form appears.

2. None Enter valid name, email, and password, then click "Submit".
Expected Result

Planning

Resolved Reason: Story Points: 1

Priority: 2 Severity: 3 - Medium Activity:

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.

Effort (Hours)

Original Estimate

Windows 10

Runner - Test Plans - Google Chrome
dev.azure.com/TO-DO-LIST-APP/TO-DO%20LIST%20APP%20WITH%20REMINDERS/_testExecution/Index

Save and close | Create bug | ...

33: User Registration & Login

1. Op... NEW BUG *

EXPE...
Reg...

2. Ent... Unassigned 0 comments Add tag

Save & Close | ...

3. Ent... State: New Area: TO-DO LIST APP WITH REMINDERS
Reason: New Iteration: TO-DO LIST APP WITH REMINDERS

Browser - Name: Google Chrome 135
Browser - Language: en-US
Browser - Height: 768
Browser - Width: 1296
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: x64
Operating system - Processor model: 13th Gen Intel(R) Core(TM) i7-1360P
Operating system - Number of processors: 16
Memory - Available: 413633312
Memory - Capacity: 16849256448
Display - Pixels per inch (X axis): 144
Display - Pixels per inch (Y axis): 144

Details | ... | (2)

Found in Build
Integrated in Build

Windows 10

8. Test case results

The screenshot shows the Azure DevOps interface for a test plan named 'TO-DO LIST APP'. The left sidebar is open, showing 'Test Plans' selected. The main area displays 'Test Case Results' for the 'TO-DO LIST APP' suite. The results table includes columns for Outcome (Passed, Failed, In Progress), TimeStamp, Configuration, Run by, Tester, and Test Point ID. There are 10 entries listed.

Outcome	TimeStamp	Configuration	Run by	Tester	Test P.
Passed	2h ago	Windows 10	Harish Tutu YT	Harish Tutu YT	TO-DC
Failed	2h ago	Windows 10	Harish Tutu YT	Harish Tutu YT	TO-DC
Passed	Wednesday	Windows 10	Gaurav Ramasubram...	Harish Tutu YT	TO-DC
Passed	Wednesday	Windows 10	Gaurav Ramasubram...	Harish Tutu YT	TO-DC
In Progress	Wednesday	Windows 10	Harish Tutu YT	Harish Tutu YT	TO-DC
Passed	Wednesday	Windows 10	Harish Tutu YT	Harish Tutu YT	TO-DC
Passed	Wednesday	Windows 10	Harish Tutu YT	Harish Tutu YT	TO-DC
Failed	Wednesday	Windows 10	Harish Tutu YT	Harish Tutu YT	TO-DC
Failed	Wednesday	Windows 10	Harish Tutu YT	Harish Tutu YT	TO-DC

9. Test report summary

The screenshot shows a detailed view of a test case step titled 'Create Task'. The step is currently in 'Design' state. The 'Steps' tab is active, showing three steps: 1. Click "Add New Task"..., 2. Fill in task title, due date, priority, and click "Save"., and 3. View the dashboard. The 'Deployment' section provides instructions on tracking releases associated with the work item. The 'Development' section includes a 'Add link' button and a note about linking to Azure Repos.

10. Progress report

Microsoft Azure portal - Yahoo | Azure DevOps - Microsoft Azure | Progress report - Test Plans | My Information

dev.azure.com/TO-DO-LIST-APP/TO-DO%20LIST%20APP%20WITH%20REMINDERS/_testManagement/analytics/progressreport

Azure DevOps TO-DO-LIST-APP / TO-DO LIST APP WITH REM... / Test Plans / Progress report

Search

TO-DO LIST APP Overview Boards Repos Pipelines Test Plans Test plans Progress report Parameters Configurations Runs Artifacts Project settings

Progress report

TO-DO LIST APP Test Suites Outcome Configuration Tester Priority Assigned To

Summary

- 1 Test plans
- 9 Test points
- 8 (8 / 9) Test points run: 88% Run
- ✓ 100% (8 / 8) Pass rate: 8 Passed

Outcome trend

Last 14 Days

Tests

2025-04-08 2025-04-09 2025-04-10 2025-04-11 2025-04-12 2025-04-13 2025-04-14 2025-04-15 2025-04-16 2025-04-17 2025-04-18 2025-04-19 2025-04-20 2025-04-21 2025-04-22

Not run Passed

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:

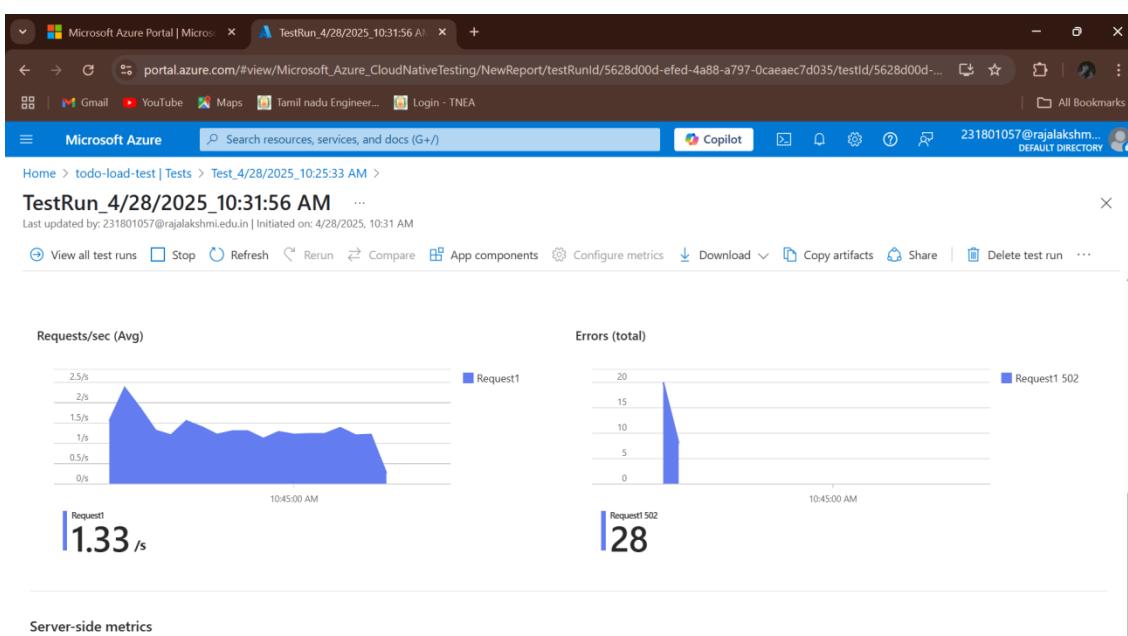
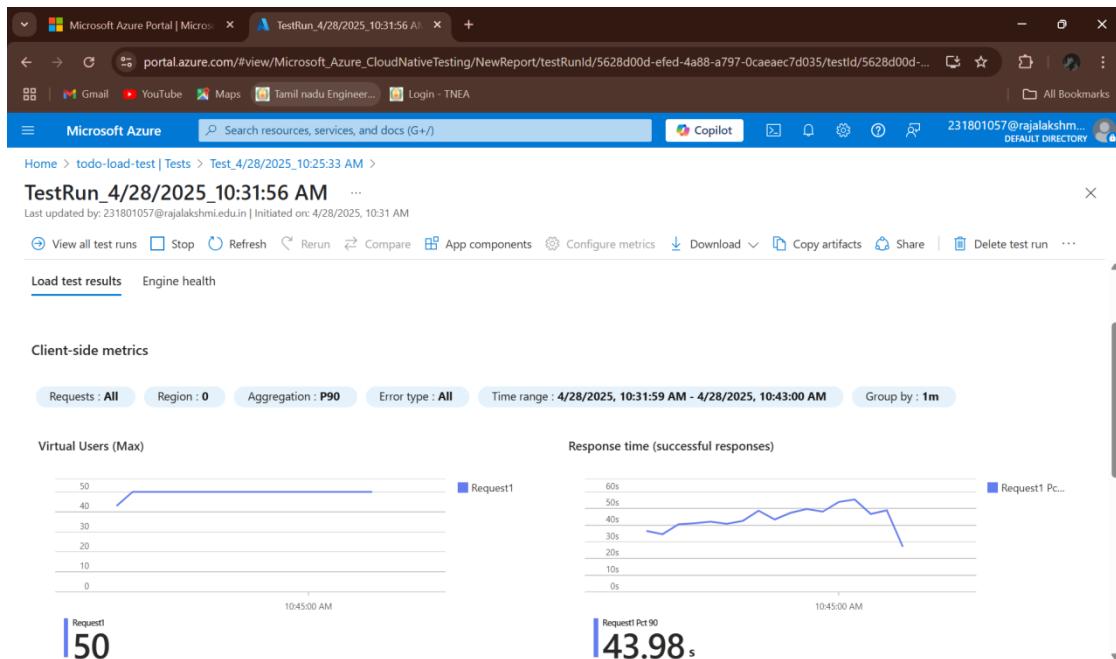
1. Sign in to Azure Portal
Go to <https://portal.azure.com> and log in.
2. Create the Resource
 - o Go to *Create a resource* → Search for “Azure Load Testing”.
 - o Select Azure Load Testing and click Create.
3. Fill in the Configuration Details
 - o *Subscription*: Choose your Azure subscription.
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

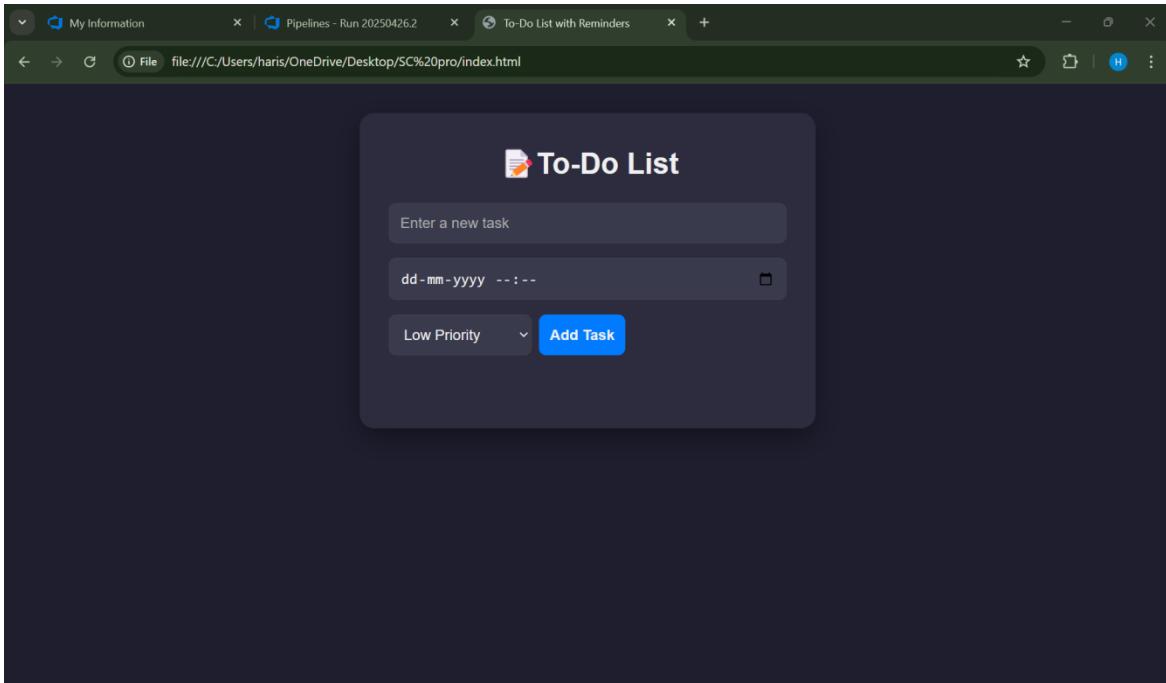
Steps to Create and Run a Load Test:

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 connecting 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 TO-DO LIST APP WITH REMINDERS!')"
    displayName: "Run a Python script"
```

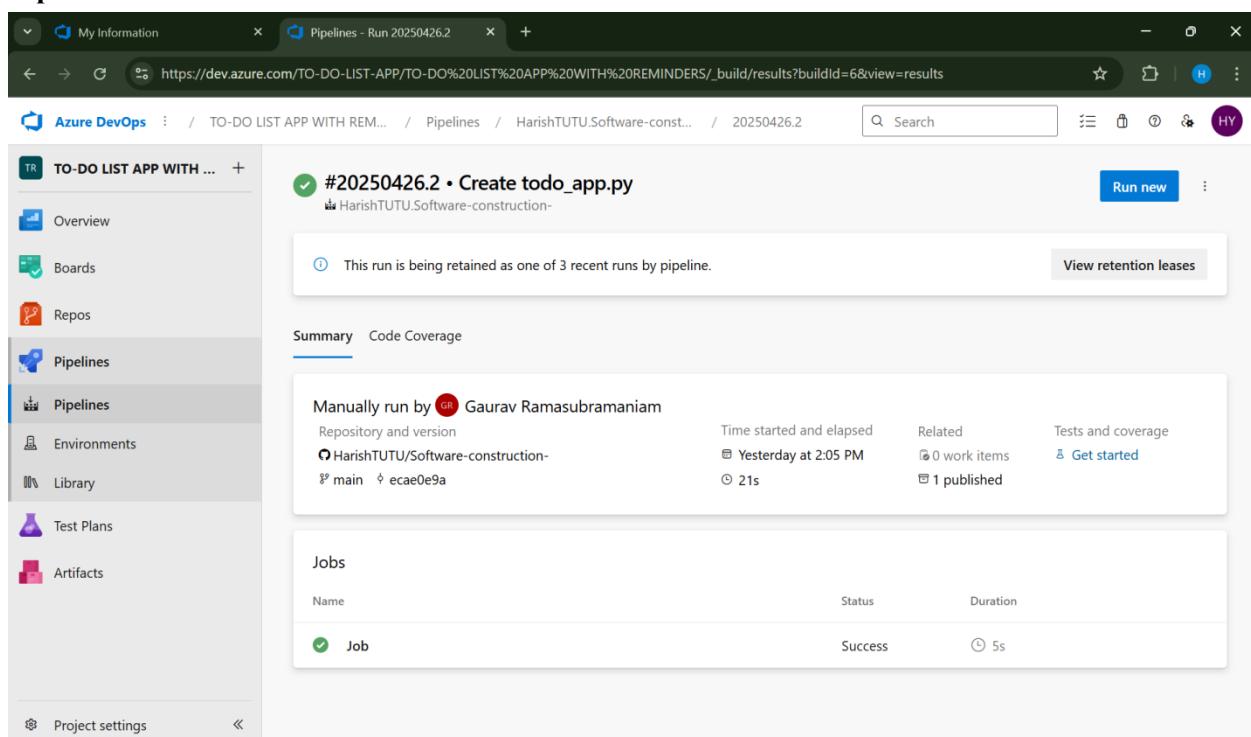
3. Pipeline Tasks Include:

- Setting up the Python environment using the UsePythonVersion task.
- Installing project dependencies from project/requirements.txt. Ensure the path to requirements.txt is correct (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 results page for a run titled '#20250426.2 • Create todo_app.py'. The pipeline was manually run by Gaurav Ramasubramaniam. The repository and version information is listed as HarishTUTU/Software-construction- main ecae0e9a. The run started yesterday at 2:05 PM and took 21s. There are 0 work items related and 1 published. A summary table shows one job named 'Job' which was successful and took 5s.

Name	Status	Duration
Job	Success	5s

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.

2116231801038

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 TO-DO LIST APP WITH REMINDERS project.

GitHub Project Structure

The screenshot shows a GitHub repository page with the following details:

- Repository Name:** TO-DO-LIST-APP-WITH-REMINDERS (Public)
- Branch:** main (selected), 1 Branch
- Tags:** 0 Tags
- Activity:** 10 Commits (by HarishTUTU), 1 minute ago
- Files:** Architecture Diagram, Backlog, Pipelines, Poker Estimation, Progress Report, Project, Sequence Diagram, Sprints, Test Plans And Test Cases, azure-pipelines.yml
- Contributors:** 10 minutes ago
- Statistics:** 0 stars, 1 watching, 0 forks
- Releases:** No releases published. Create a new release
- Packages:** No packages published. Publish your first package
- Languages:** HTML 70.1%, Python 29.9%

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.