



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: Harish tutu YT

Register No: 231801050

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 HARI.S.H T.Y.T.O.Y.Y.T

ACADEMIC YEAR ...2024-2025.... SEMESTERI..... BRANCH.....A.I.O.D.S.-F.A.

UNIVERSITY REGISTER No.

2116231801050

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.....

Internal Examiner



External Examiner
CamScanner

INDEX

Name: Harish Tawar Branch: AI&DS Sec: FA Roll No: 231801050

S.No.	Date	Title	Page No.	Teacher's Sign/Remarks
1.	22/11/25	Azure DevOps Environment Setup	4	✓
2.	22/11/25	Azure devops project setup and uses story management.	8	✓
3.	29/11/25	Setting up Epics, Features and uses stories for project planning.	12	✓
4.	10/12/25	sprint planning.	15	✓
5.	19/12/25	Pokes estimation	14	✓
6.	26/12/25	Designing class and sequence Diagram	19	✓
7.	5/1/25	Designing Architecture and ER Diagram	31	✓
8.	26/1/25	Testing- Test plan and Test cases	33	✓
9.	16/1/25	Load Testing and Pipeline	38	✓
10.	02/4/25	github: Project structure and naming conventions	13	✓

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:

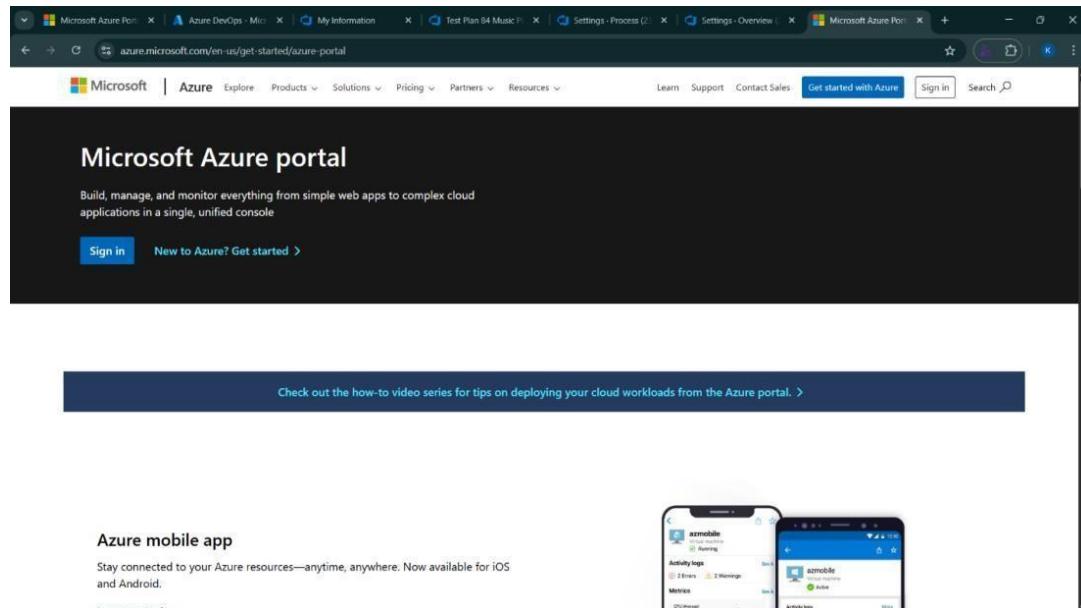
To set up and access the Azure DevOps environment by creating an organization through the Azure portal.

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>



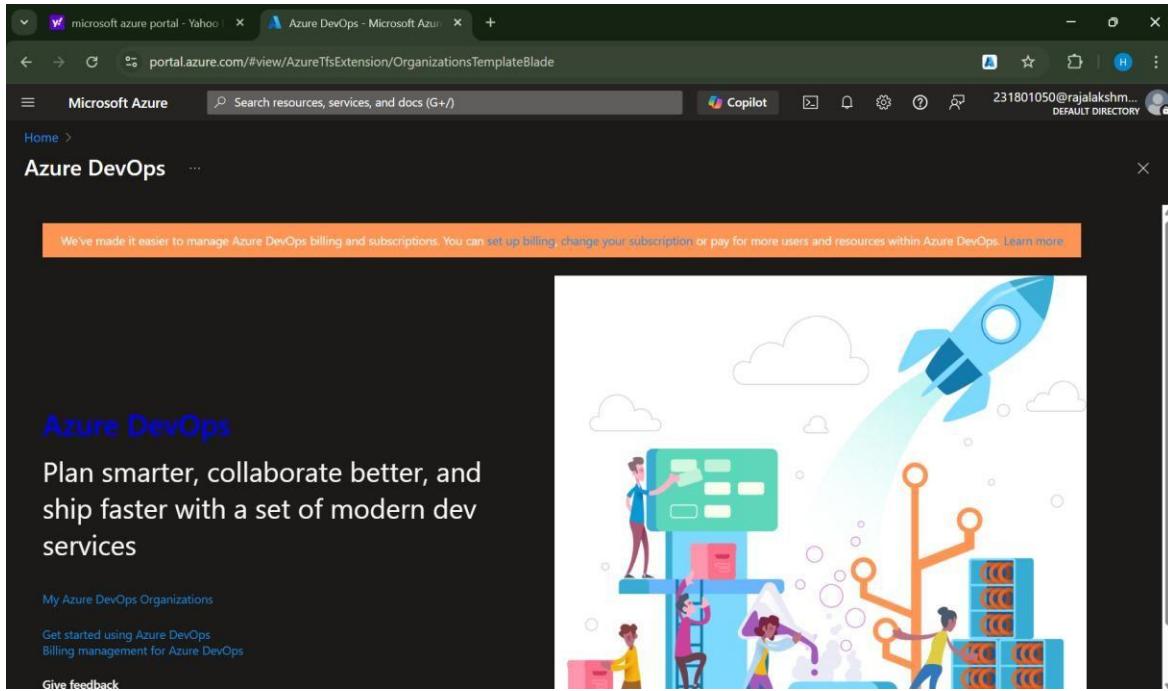
2. Azure home page

The screenshot shows the Microsoft Azure home page. At the top, there's a navigation bar with a back arrow, forward arrow, refresh button, and a search bar containing "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 titled "Azure services" with 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". Below this is a section titled "Resources" with tabs for "Recent" (which is selected) and "Favorite". It lists "Name", "Type", and "Last Viewed" columns. A message says "No resources have been viewed recently" 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 but with a search term in the search bar. The search bar now contains "Devops" and has a "Clear all" link next to it. The rest of the interface is identical to the first screenshot, showing the Azure services menu and the "Resources" section with no recent resources.

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.

2116231801050



CamScanner
CS23432

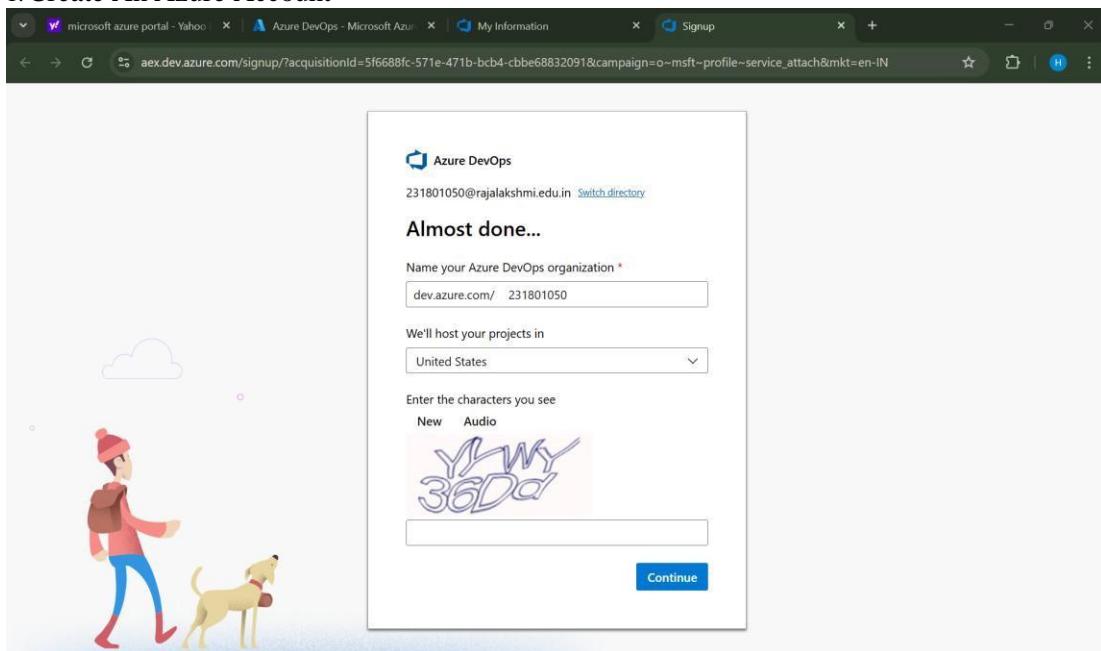
EXP NO: 2

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 sidebar with the user's profile picture (a purple circle with 'HY'), name (Harish Tutu YT), email (231801050@rajalakshmi.edu.in), and account information (Microsoft account, India). Below this is a section for Visual Studio Dev Essentials. On the right, the main area displays the 'Azure DevOps Organizations' header with a 'Create new organization' button. Under 'Projects', it lists 'dev.azure.com/TO-DO-LIST-APP (Owner)' which contains the 'TO-DO LIST APP WITH REMINDERS' project. There are also 'Actions' like 'Open in Visual Studio' and a 'New project' button.

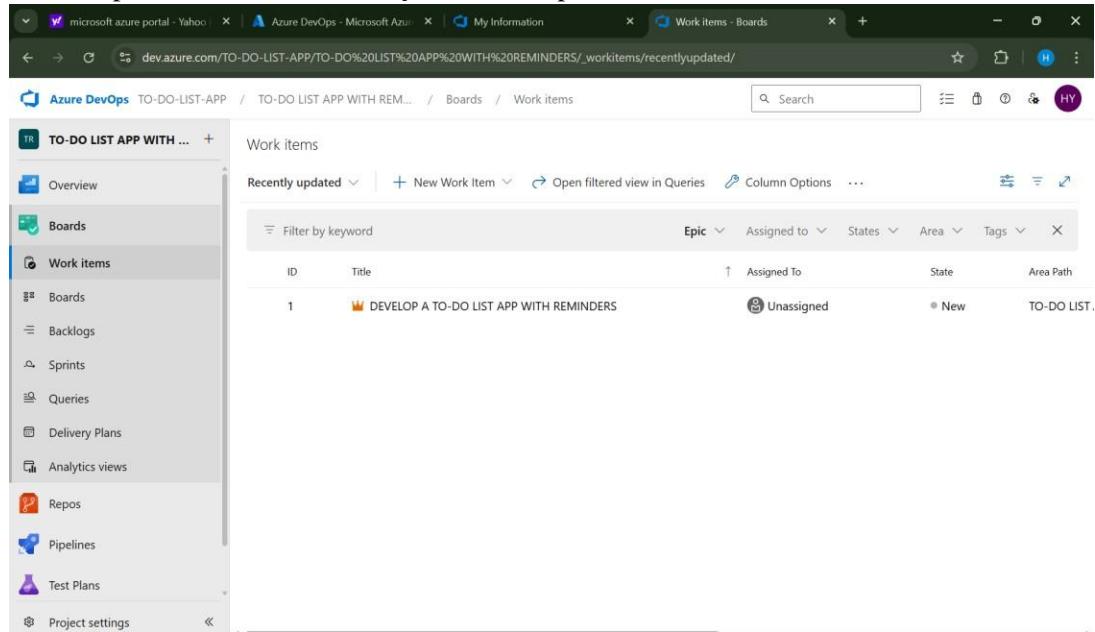
4. Project dashboard

The screenshot shows the project dashboard for 'TO-DO LIST APP WITH REMINDERS'. The left sidebar includes links for Overview, Summary (which is selected), Dashboards, Wiki, Boards, Repos, Pipelines, Test Plans, Artifacts, and Project settings. The main content area features a title bar with the project name 'TO-DO LIST APP WITH REMINDERS' and a 'Public' button. 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. The 'Project stats' section shows data for the last 7 days: 12 work items created and 0 work items completed. 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 managing work items. The left sidebar is titled 'TO-DO LIST APP WITH REMINDERS' and includes options like Overview, Boards, Work items (which is selected), Boards, Backlogs, Sprints, Queries, Delivery Plans, Analytics views, Repos, Pipelines, Test Plans, and Project settings. The main area is titled 'Work items' and shows a table with one row. The table columns are ID, Title, Assigned To, State, and Area Path. The single row contains: ID 1, Title 'DEVELOP A TO-DO LIST APP WITH REMINDERS', Assigned To 'Unassigned', State 'New', and Area Path 'TO-DO LIST.'. There are also buttons for 'Recently updated', 'New Work Item', 'Open filtered view in Queries', 'Column Options', and a search bar.

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.

2116231801050

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 titled "TO-DO LIST APP WITH REMINDERS". The left sidebar is open, showing options like Overview, Boards, Work items, Backlogs, Sprints, Queries, Delivery Plans, Analytics views, Repos, Pipelines, Test Plans, and Project settings. The main area is titled "Work items" and shows a list of recently updated items. The columns include ID, Title, Assigned To, State, and Area Path. The first item listed is "DEVELOP A TO-DO LIST APP WITH REMINDERS".

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 Azure DevOps interface for a project titled "TO-DO LIST APP WITH REMINDERS". The left sidebar is open, showing options like Overview, Boards, Work items, Backlogs, Sprints, Queries, Delivery Plans, Analytics views, Repos, Pipelines, Test Plans, and Project settings. The main area is titled "Work items" and shows a list of recently updated items. The first item listed is "EPIC 1". The epic details are shown in the center, including its title, state (New), area (TO-DO LIST APP WITH REMINDERS), iteration (TO-DO LIST APP WITH REMINDERS), and deployment status (Updated by Dinesh S: Feb 15). The epic description states: "This epic covers the development of a to-do list application with task management, reminders, and notifications." The epic has no comments or add tags.

2. Fill in Features

The screenshot shows the Azure DevOps Work Items page for a project titled "TO-DO LIST APP WITH REMINDERS". A feature named "4 Reminder Notifications" is selected. The "Description" section includes a bulleted list of requirements: "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.", and "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.". The "Planning" section shows a priority of 2 and a risk of 1. The "Deployment" section contains a note about tracking releases. The "Development" section includes a link to "Link an Azure Repos".

3. Fill in User Story Details

The screenshot shows the Azure DevOps Work Items page for a project titled "TO-DO LIST APP WITH REMINDERS". A user story named "9 As a user, I want to create a new task so that I can keep track of my work." is selected. The "Description" section includes a bulleted list of requirements: "Users should be able to add a task with a title, description, and due date.", "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.", and "Classification: Value area Business". The "Planning" section shows a story points of 5 and a priority of 2. The "Deployment" section contains a note about tracking releases. The "Development" section includes a link to "Link an Azure Repos".

Result:

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

EXP NO: 4

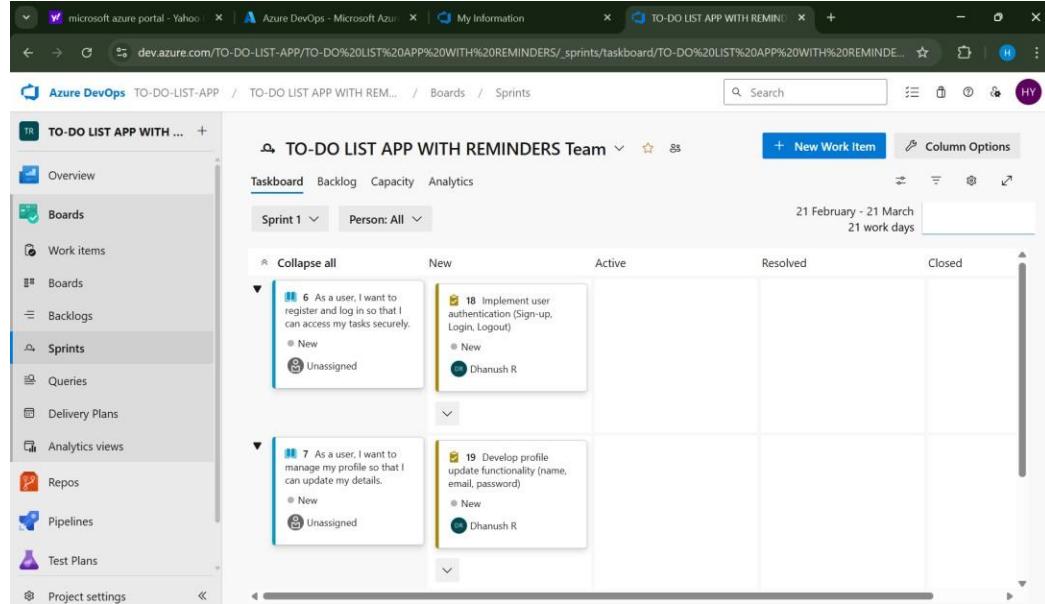
SPRINT PLANNING

Aim:

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

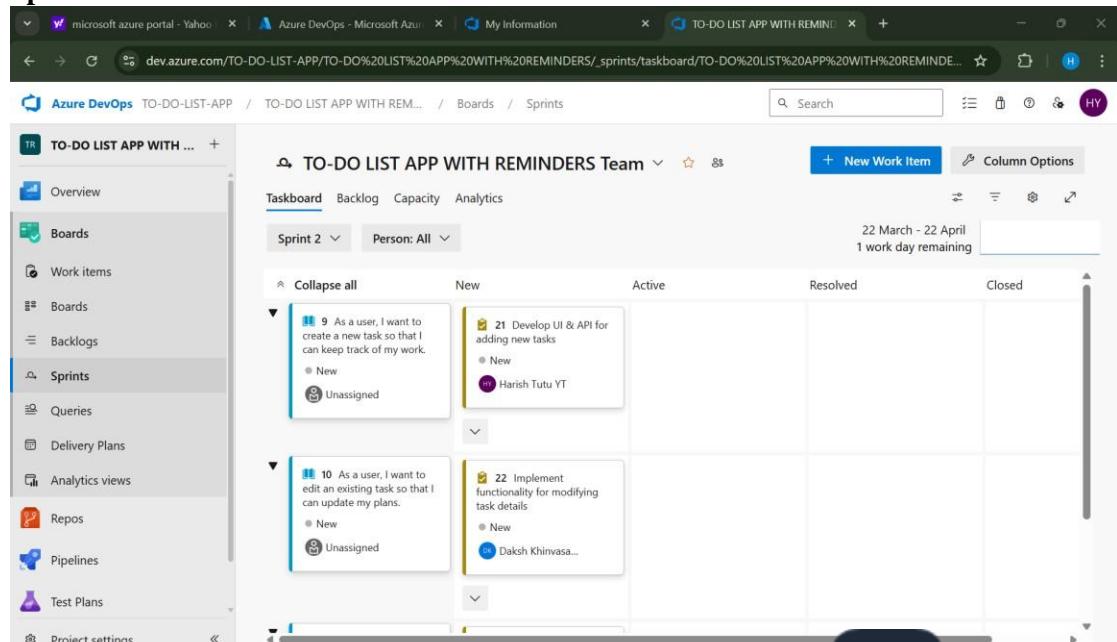
Sprint Planning

Sprint 1



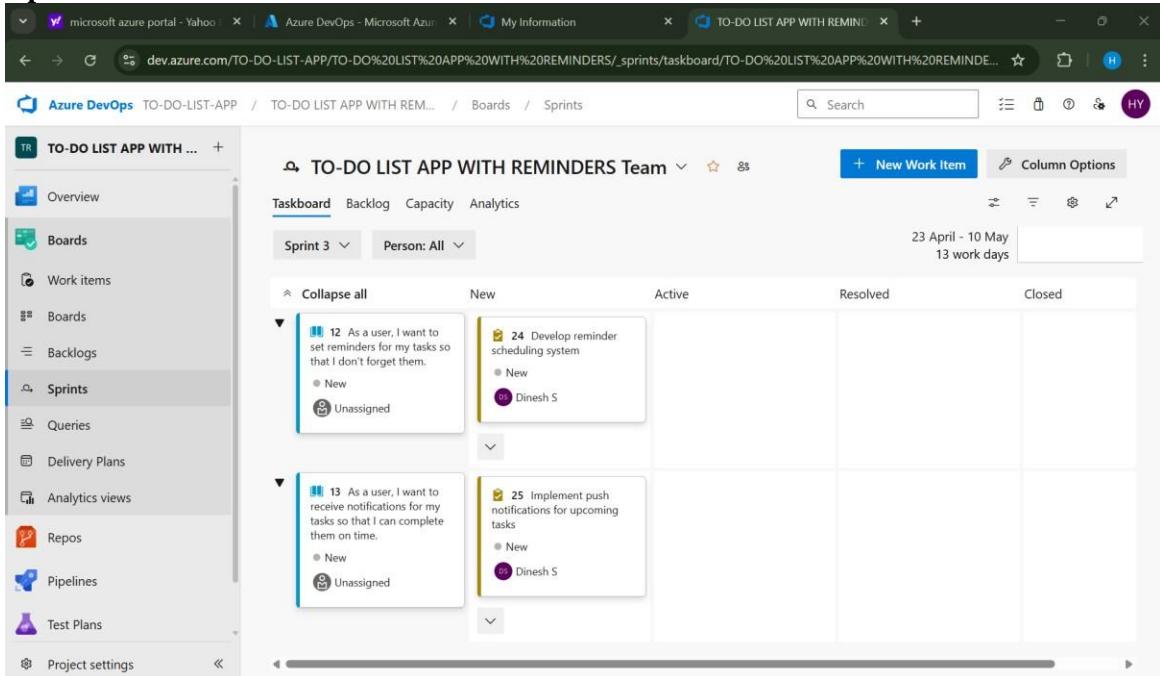
The screenshot shows the Azure DevOps Taskboard for the 'TO-DO LIST APP WITH REMINDERS' project. The left sidebar is expanded to show 'Boards' under 'Sprints'. The main area displays the 'Taskboard' for 'Sprint 1' from February 21 to March 21. The backlog contains several user stories, some of which have been assigned to team members. For example, user story #6 is assigned to Dhanush R, and user stories #18 and #19 are also assigned to him. Other stories are marked as 'New' or 'Unassigned'.

Sprint 2



The screenshot shows the Azure DevOps Taskboard for the 'TO-DO LIST APP WITH REMINDERS' project. The left sidebar is expanded to show 'Boards' under 'Sprints'. The main area displays the 'Taskboard' for 'Sprint 2' from March 22 to April 22. The backlog contains several user stories, some of which have been assigned to team members. For example, user story #9 is assigned to Harish Tutu YT, and user stories #21 and #22 are also assigned to him. Other stories are marked as 'New' or 'Unassigned'.

Sprint 3

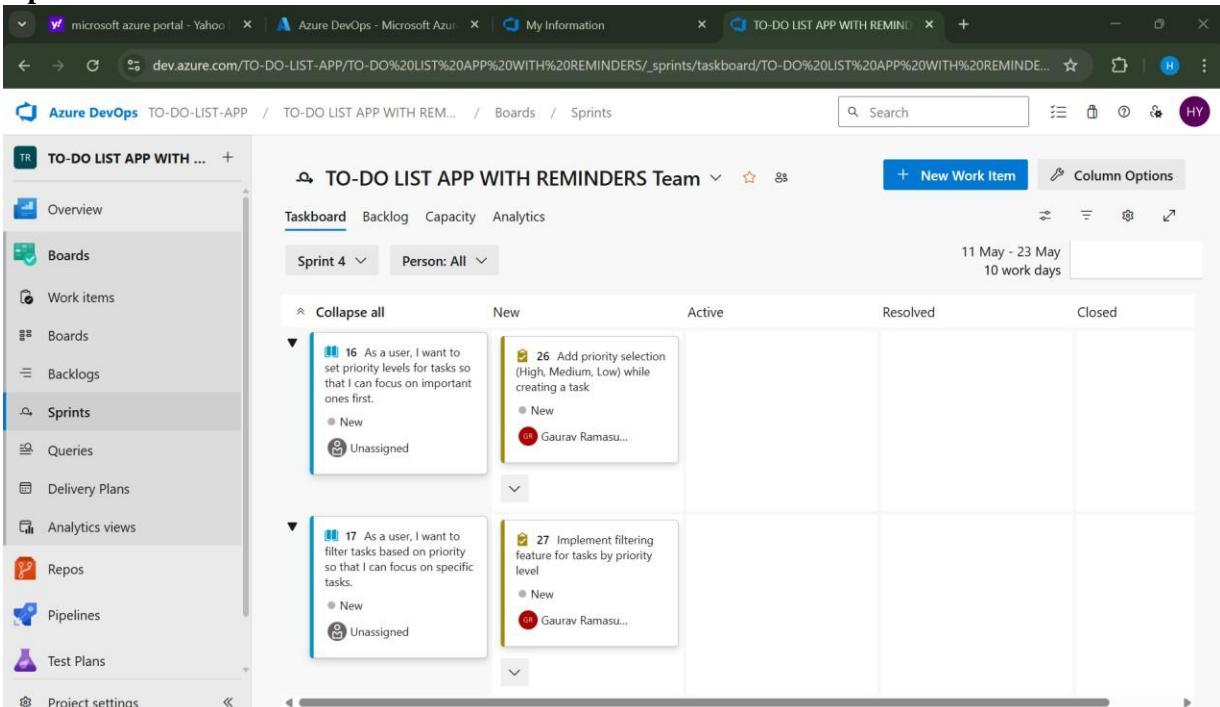


The screenshot shows the Azure DevOps Taskboard for the 'TO-DO LIST APP WITH REMINDERS Team' in 'Sprint 3'. The taskboard displays four items:

- 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 has columns for New, Active, Resolved, and Closed. The date range for the sprint is 23 April - 10 May, covering 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 has columns for New, Active, Resolved, and Closed. The date range for the sprint is 11 May - 23 May, covering 10 work days.

Result:

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

2116231801050

EXP NO: 5

POKER ESTIMATION

Aim:

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

Poker Estimation

The screenshot shows a detailed view of a User Story in Azure DevOps. The story is titled "As a user, I want to register and log in so that I can access my tasks securely." It has a state of "New" and is assigned to the "TO-DO LIST APP WITH REMINDERS" area and "TO-DO LIST APP WITH REMINDERS/Sprint 1" iteration. The "Description" field contains the requirement: "Users should be able to sign up and log in using email and password authentication." The "Acceptance Criteria" section lists: "Users can sign up with an email and password.", "Users can log in using valid credentials.", and "Users are redirected to the task dashboard upon successful login.". In the "Planning" section, Story Points are set to 3 and Priority to 1. The "Deployment" section provides instructions for tracking releases. The "Classification" section indicates the value area is "Business". The "Development" section includes a link to an Azure Repos commit or pull request. The "Discussion" section allows for comments, with a placeholder "Add a comment. Use # to link a work item, @ to mention a person, or ! to...".

Result:

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

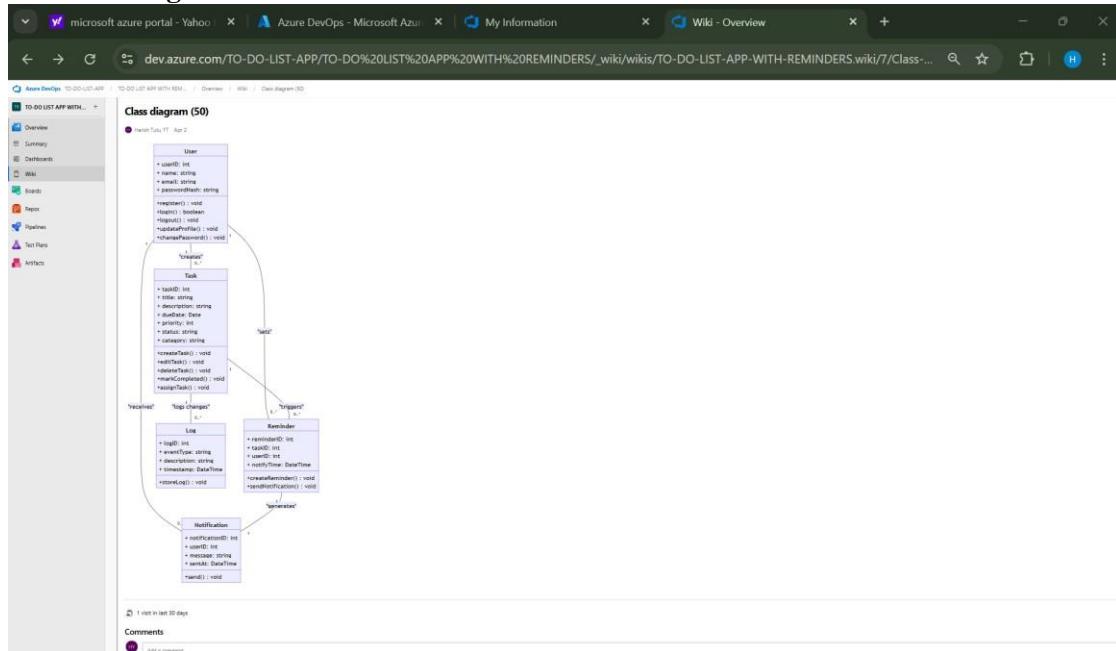
EXP NO: 6

DESIGNING CLASS AND SEQUENCE DIAGRAMS FOR PROJECT ARCHITECTURE

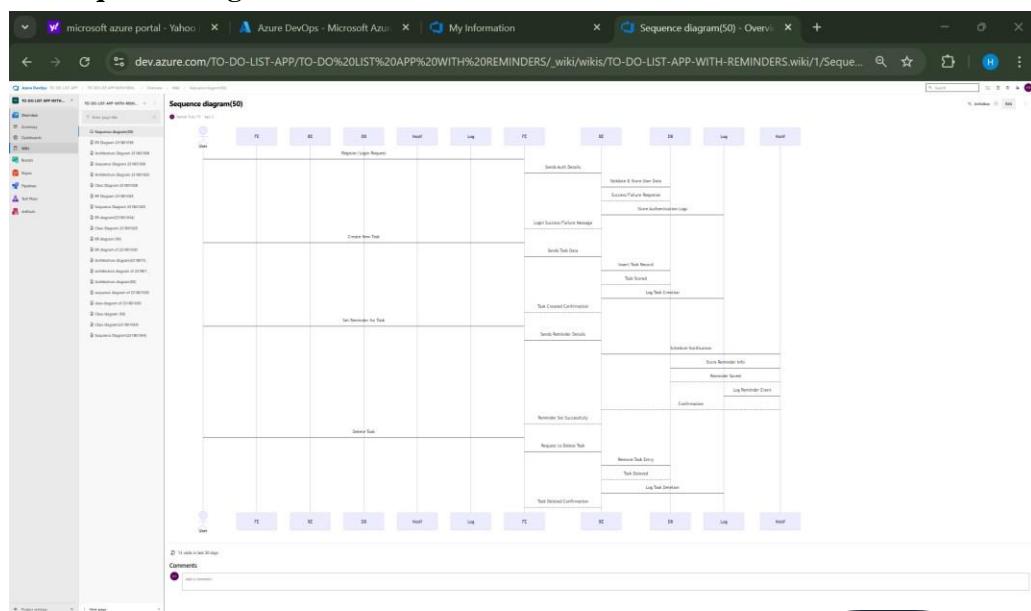
Aim:

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.

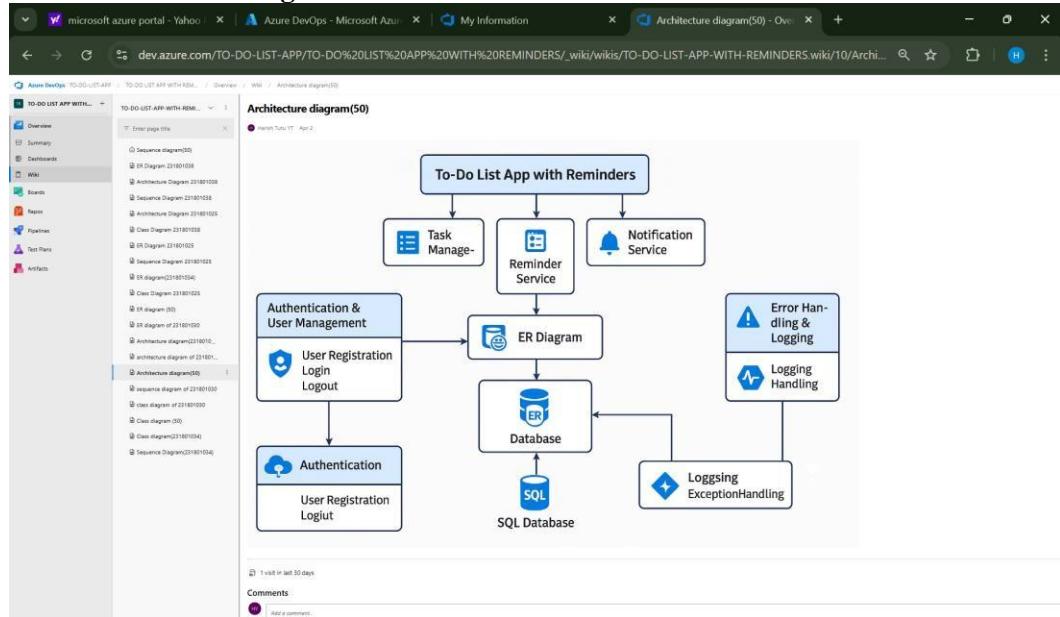
EXP NO: 7

DESIGNING ARCHITECTURAL AND ER DIAGRAMS FOR PROJECT STRUCTURE

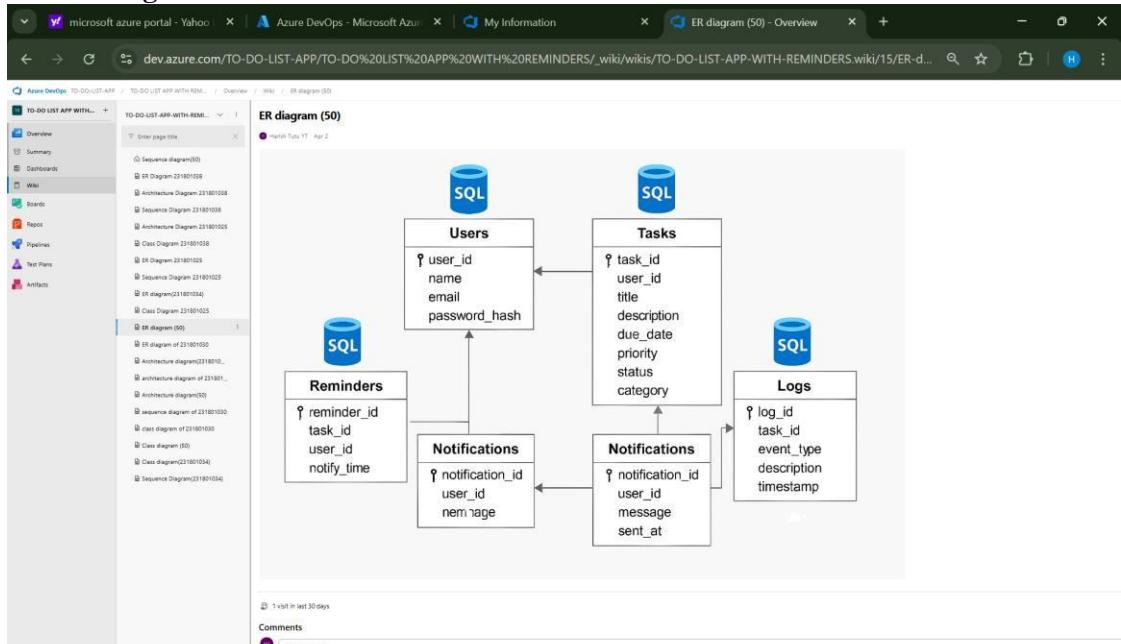
Aim:

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.

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 Plans' section of the Azure DevOps interface. A specific test plan titled '28 TO-DO LIST APP' is selected. The plan details include:

- Owner:** Harish Tutu YT
- Comments:** 0 Comments
- Add Tag:** Add Tag
- Status:** Active
- Reason:** New test plan
- Area:** TO-DO LIST APP WITH REMINDERS
- Iteration:** TO-DO LIST APP WITH REMINDERS

The 'Timelines' section shows the start date as 15-04-2025 15:09 and the finish date as 22-04-2025 15:09. The 'Description' and 'Discussion' sections are currently empty. A note at the bottom of the discussion area suggests using Markdown editor.

2. Test suite

The screenshot shows the 'Test Suites' management screen for the 'TO-DO LIST APP'. The left sidebar lists various project components, and the main area displays the 'Task management (ID: 31)' section. Under 'Test Suites', there is a dropdown menu with the following options:

- New Suite
- Static suite
- Requirement based suite
- Query based suite

The 'Test Cases' section lists three items:

Title	Order	Test
Create Task	1	34
Edit Task	2	36
Delete Task	3	39

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.

2116231801050

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 a Microsoft Edge browser window displaying the Azure DevOps Test Plan interface. The URL in the address bar is dev.azure.com/TO-DO-LIST-APP/TO-DO%20LIST%20APP%20WITH%20REMINDERS/_testPlans/define?planId=28&suiteId=30. The page title is "Test Plan 28 TO-DO LIST APP".

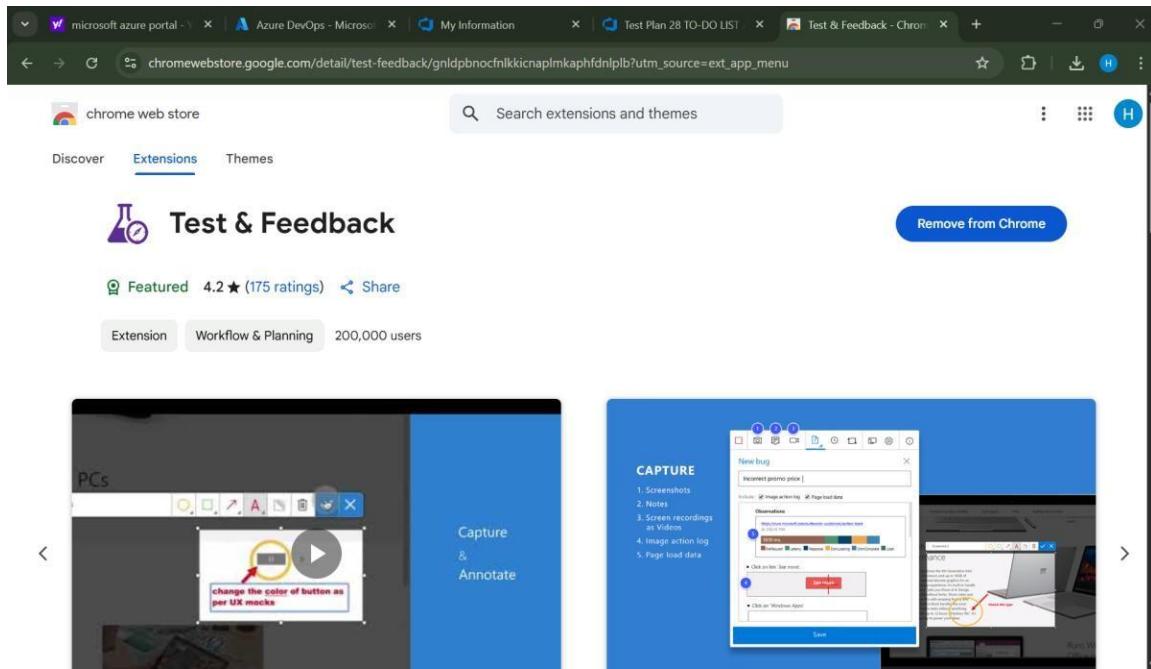
The main content area displays a test case titled "33 User Registration & Login". The test case is assigned to "Dinesh S" and has 0 comments and 0 tags. It is in the "Design" state, part of the "TO-DO LIST APP WITH REMINDERS" area, and is associated with the "TO-DO LIST APP WITH REMINDERS" iteration. The "Steps" tab is selected, showing three steps:

Step	Action	Expected result
1.	Open the app and click "Register".	Registration form appears.
2.	Enter valid name, email, and password, then click "Submit".	Account created, redirected to
3.	Enter credentials and click "Login".	Dashboard loads, user success

Below the steps, there is a note: "Click or type here to add a step". A "Parameter values" section is also present.

On the right side of the screen, there are two sections: "Deployment" and "Development". The "Deployment" section provides instructions on tracking releases. The "Development" section provides instructions on linking Azure Repos commits, pull requests, or branches.

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.

A modal window titled "Extensions" is open on the right. It lists extensions under "Full access": "Allow Copy - Select & En...", "Blend & Run: Enable copy...", "Test & Feedback" (which is 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.

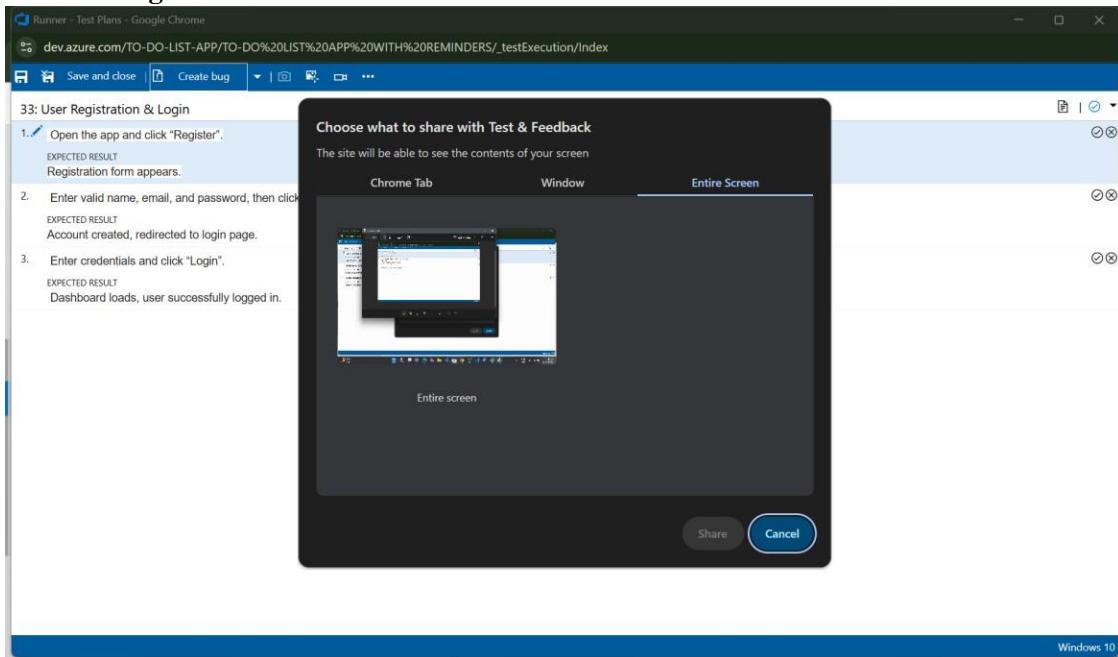
5. Running the test cases

The screenshot shows the Azure DevOps Test Plan interface for the 'TO-DO LIST APP'. The left sidebar shows the project navigation with 'Test Plans' selected. In the main area, the 'Authentication (ID: 30)' test plan is open. The 'Execute' tab is active, showing 'Test Points (3 items)'. The first item, 'User Registration & Login', is selected and marked as 'Passed'. A context menu is open over this item, listing options like 'Run for web application', 'Run for desktop application', and 'Run with options'. Other items in the list include 'Manage Profile' and 'Logout', each with their own outcome status.

6.

The screenshot shows the 'Runner - Test Plans' interface in Google Chrome. The title bar says 'Runner - Test Plans - Google Chrome' and the address bar shows 'dev.azure.com/TO-DO-LIST-APP/TO-DO%20LIST%20APP%20WITH%20REMINDERS/_testExecution/Index'. The page displays a test case titled '33: User Registration & Login'. Step 1, 'Open the app and click "Register".', has an 'EXPECTED RESULT' section stating 'Registration form appears.' Step 2, 'Enter valid name, email, and password, then click "Submit".', has an 'EXPECTED RESULT' section stating 'Account created, redirected to login page.' Step 3, 'Enter credentials and click "Login".', has an 'EXPECTED RESULT' section stating 'Dashboard loads, user successfully logged in.' There are two small circular icons with checkmarks at the bottom right of the test steps.

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. Opt EXP Reg Ent EXP Acc

2. Ent EXP Acc

3. Ent EXP Acc

NEW BUG *

loading due to poor network

Unassigned 0 comments Add tag

Save & Close

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: 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.

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. Opt EXP Reg Ent EXP Acc

2. Ent EXP Acc

3. Ent EXP Acc

NEW BUG *

loading due to poor network

Unassigned 0 comments Add tag

Save & Close

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

Environment

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	x86_64
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

Found in Build
Integrated in Build

Windows 10

8. Test case results

The screenshot shows the Azure DevOps interface for a 'TO-DO LIST APP' project. On the left, the navigation menu is open, showing 'Test Plans' selected under 'Test'. In the center, a 'Test Suites' section is displayed for 'TO-DO LIST APP', with a sub-section for 'Task management (3)'. A modal window titled 'Test Case Results' is open, showing a table of test outcomes:

Outcome	TimeStamp	Configuration	Run by	Tester	Test Point
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	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

At the bottom of the modal, there is a link: 'Open execution history for current test point'.

9. Test report summary

The screenshot shows a detailed view of a test case step for 'Create Task'. The step is titled '34 Create Task' and is assigned to 'Harish Tutu YT'. The 'State' dropdown is set to 'Design'. The 'Reason' dropdown shows 'Ready' as the selected option. The 'Steps' tab is active, displaying the following steps:

- Click "Add New Task"...
- Fill in task title, due date, priority, and click "Save".
- View the dashboard.

The 'Expected result' column indicates that the first step leads to 'Task creation form opens.', the second to 'Task added to list', and the third to 'Newly created task appears in'. To the right of the steps, there are sections for 'Deployment' and 'Development'. 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 commits or branches.

10. Progress report

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

Search

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

Legend: Not run (grey), Passed (green)

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:

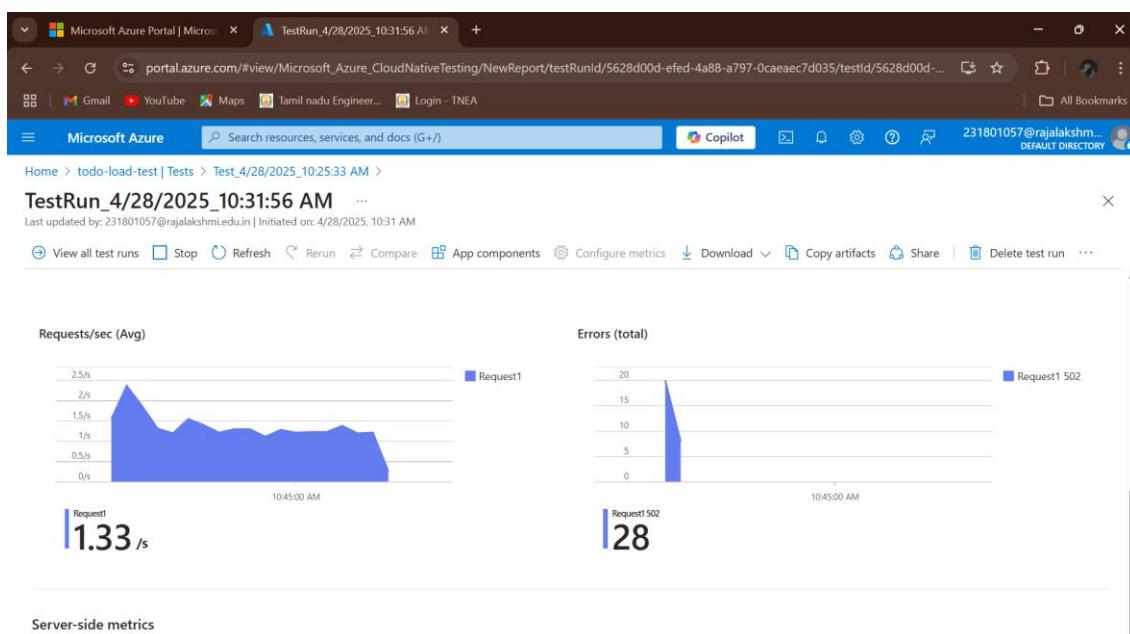
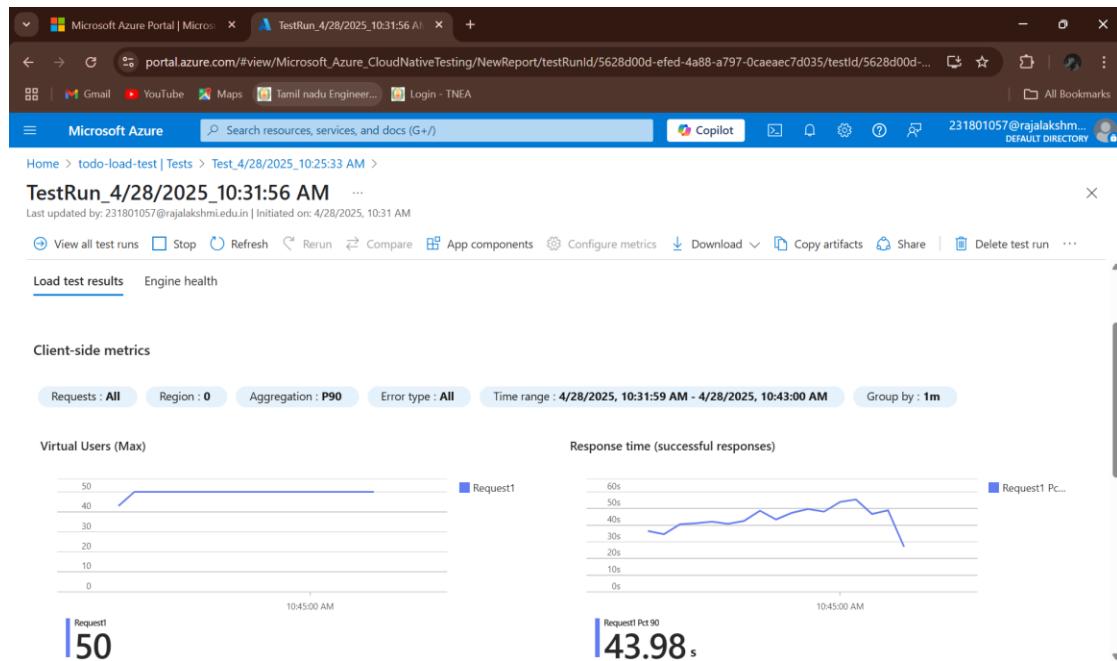
1. Sign in to Azure Portal
Go to <https://portal.azure.com> and log in.
2. Create the Resource
 - o Go to *Create a resource* → Search for “Azure Load Testing”.
 - o Select Azure Load Testing and click Create.
3. Fill in the Configuration Details
 - o *Subscription*: Choose your Azure subscription.
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

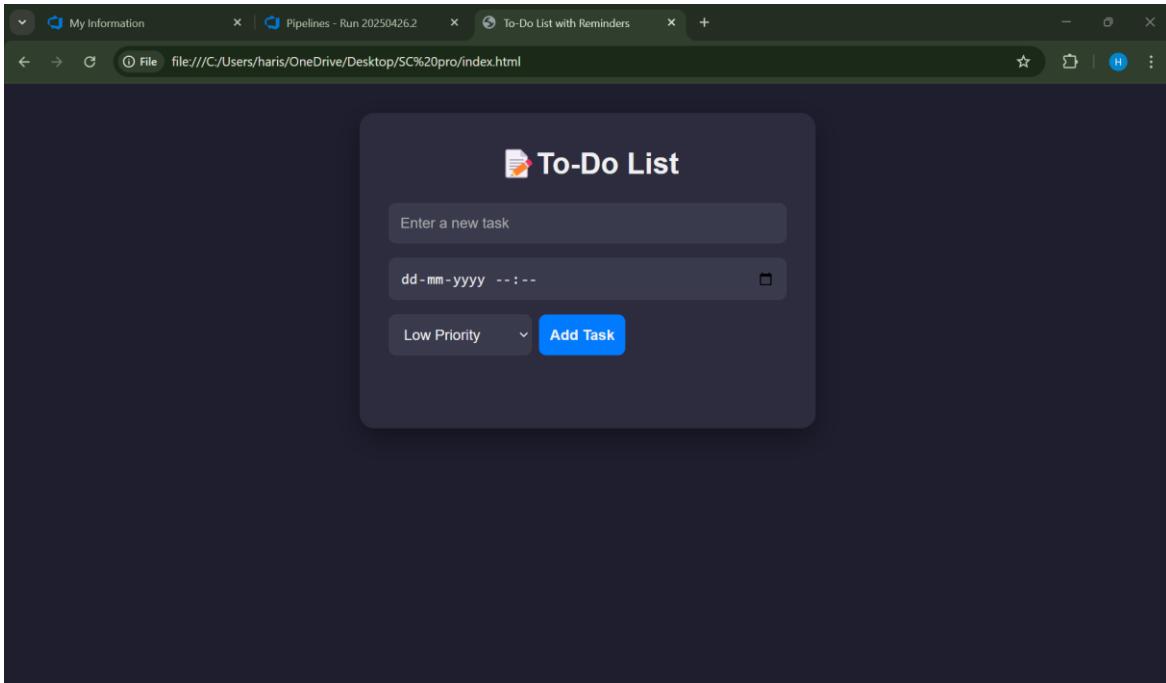
Steps to Create and Run a Load Test:

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"
```

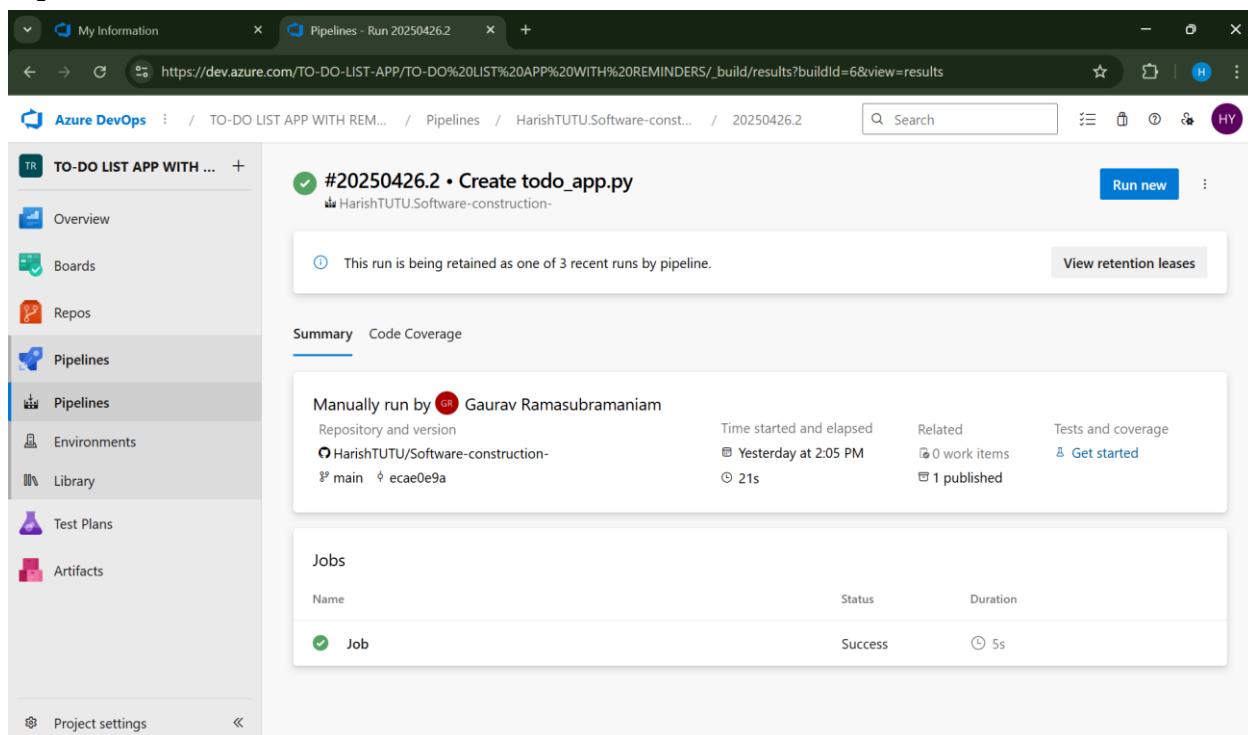
3. Pipeline Tasks Include:

- o Setting up the Python environment using the UsePythonVersion task.
- o Installing project dependencies from project/requirements.txt. Ensure the path to requirements.txt is correct (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 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 elapsed for 21s. There are 0 work items related and 1 published test. The 'Jobs' section shows a single job named 'Job' which completed successfully in 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.

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 for 'TO-DO-LIST-APP-WITH-REMINDERS'. The repository is public and has 1 branch and 0 tags. The main branch has 10 commits. The repository has 0 stars, 1 watcher, and 0 forks. There are no releases or packages published. The languages used are HTML (70.1%) and Python (29.9%). The repository structure includes files like 'azure-pipelines.yml', 'Test Plans And Test Cases', 'Sprints', 'Project', 'Progress Report', 'Poker Estimation', 'Pipelines', 'Backlog', 'Architecture Diagram', and a file added via upload by 'HarishTUTU'. The repository was last updated 44 minutes ago.

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.