



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 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:

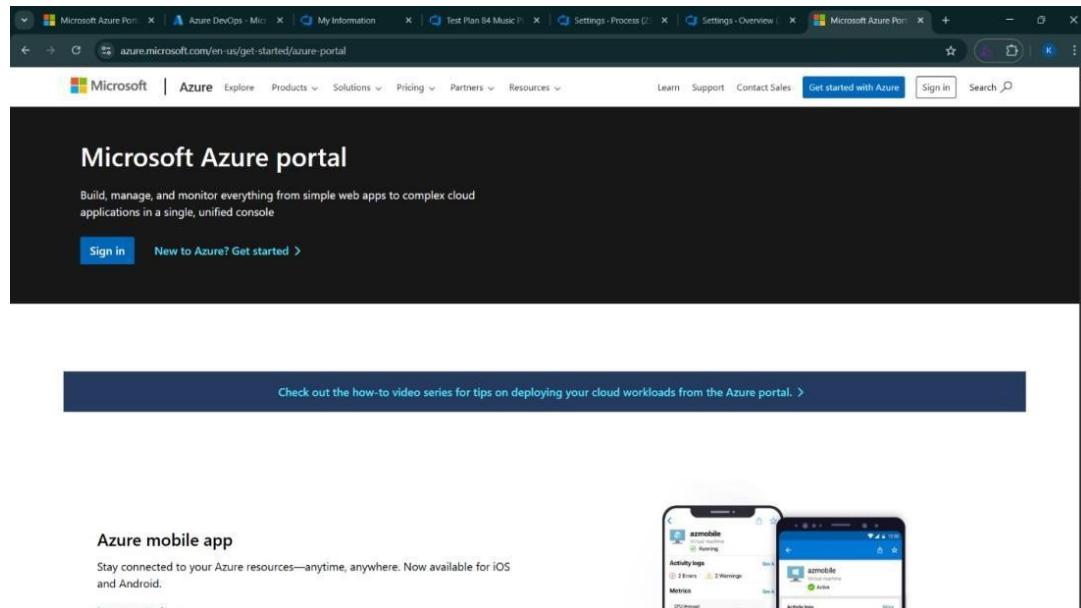
To set up and access the Azure DevOps environment by creating an organization through the Azure portal.

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>



Azure mobile app

Stay connected to your Azure resources—anytime, anywhere. Now available for iOS and Android.

[Learn more](#)



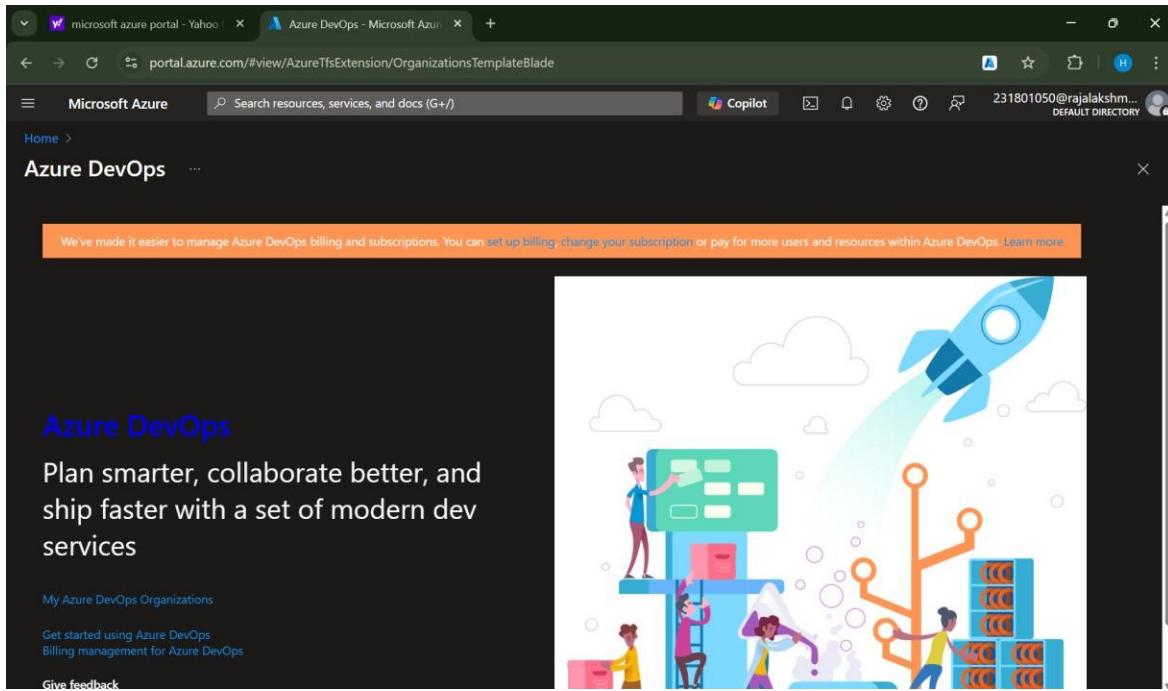
2. Azure home page

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, the 'Microsoft Azure' logo is displayed, followed by 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 is titled 'Azure services' and features a grid of service icons: 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 displays a message: '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', with 'Devops' also appearing in the 'Search history' dropdown. The rest of the interface is identical to the first screenshot, showing the Azure services menu and the 'Resources' section with its 'Recent' tab selected.

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

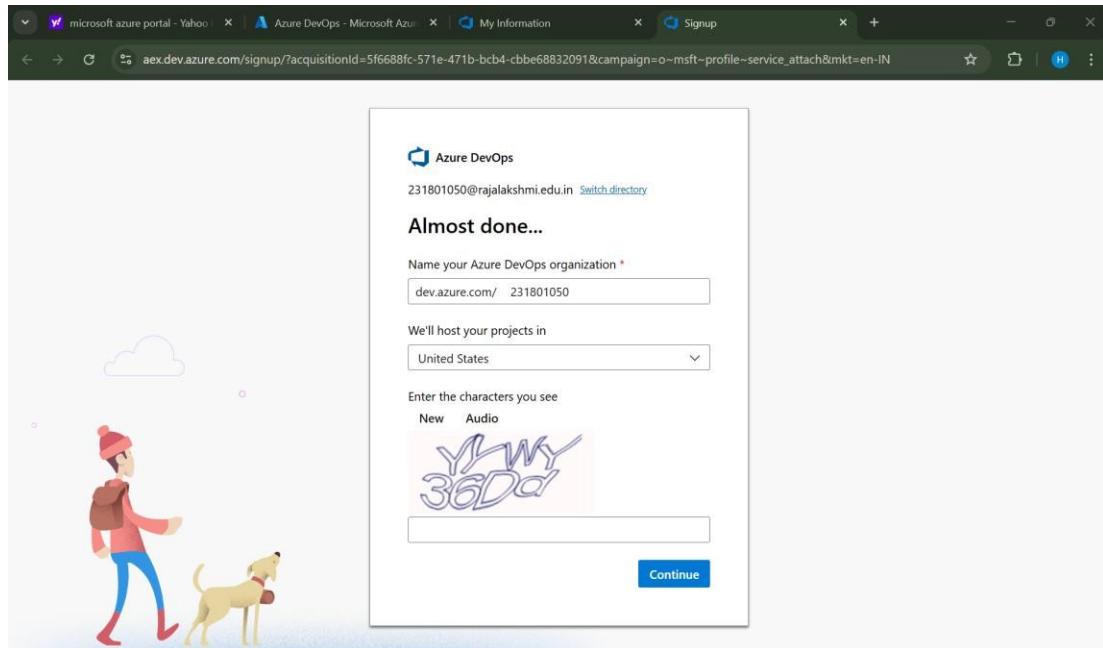
EXP NO: 2

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' and a dropdown menu showing 'Microsoft account'. Further down are location ('India') and email details ('231801050@rajalakshmi.edu.in'). A 'Visual Studio Dev Essentials' section is also present. On the right, the main dashboard for the 'TO-DO LIST APP WITH REMINDERS' project is displayed. It includes a 'Create new organization' button, a 'Projects' section with one item ('TO-DO LIST APP WITH REMINDERS'), and an 'Actions' section with a 'Open in Visual Studio' link. There's also a 'New project' button.

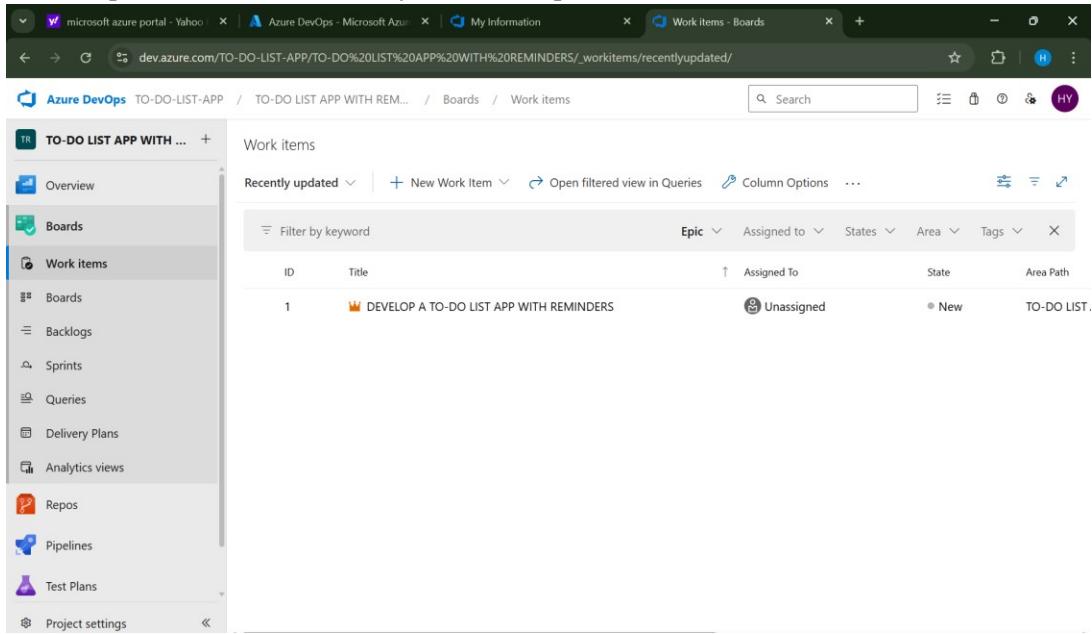
4. Project dashboard

The screenshot shows the 'Summary - Overview' page for the 'TO-DO LIST APP WITH REMINDERS' project. The top navigation bar includes links for 'microsoft portal - Yahoo', 'Azure DevOps - Microsoft Azure', 'My Information', and the current 'Summary - Overview' tab. The main content area features a title 'TO-DO LIST APP WITH REMINDERS' with a 'Public' and 'Invite' button. Below this are sections for 'About this project', 'Project stats', and 'Members'. The 'About this project' section contains a brief description of the app. The 'Project stats' section displays metrics like '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. A sidebar on the left provides links to 'Overview', 'Summary', 'Dashboards', 'Wiki', 'Boards', 'Repos', 'Pipelines', 'Test Plans', 'Artifacts', and 'Project settings'.

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 'Work items' section of the Azure DevOps interface. The left sidebar has 'Work items' selected. The main area displays a table of work items with columns for ID, Title, Assigned To, State, and Area Path. The table contains five entries:

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 'Work items' section of the Azure DevOps interface, focusing on an epic. The left sidebar has 'Work items' selected. The main area shows the details of an epic named 'EPIC 1'. The epic title is '1 DEVELOP A TO-DO LIST APP WITH REMINDERS'. The epic is in 'New' state and is associated with the 'TO-DO LIST APP WITH REMINDERS' area. 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 attachments. The epic is updated by Dinesh S on Feb 15.

2. Fill in Features

The screenshot shows the Azure DevOps interface for a work item titled "FEATURE 4". The left sidebar is visible with options like Overview, Boards, Work items, Backlogs, Sprints, Queries, Delivery Plans, Analytics views, Repos, Pipelines, Test Plans, and Project settings. The main area displays the following 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
- 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 visible with the same set of options. The main area displays the following details:

- Description:** 9 As a user, I want to create a new task so that I can keep track of my work.
- 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
- 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 interface for the 'TO-DO LIST APP WITH REMINDERS' project. The left sidebar is open with 'Boards' selected. The main area displays the 'Taskboard' for 'Sprint 1'. The board has columns for 'New', 'Active', 'Resolved', and 'Closed'. There are four user stories listed:

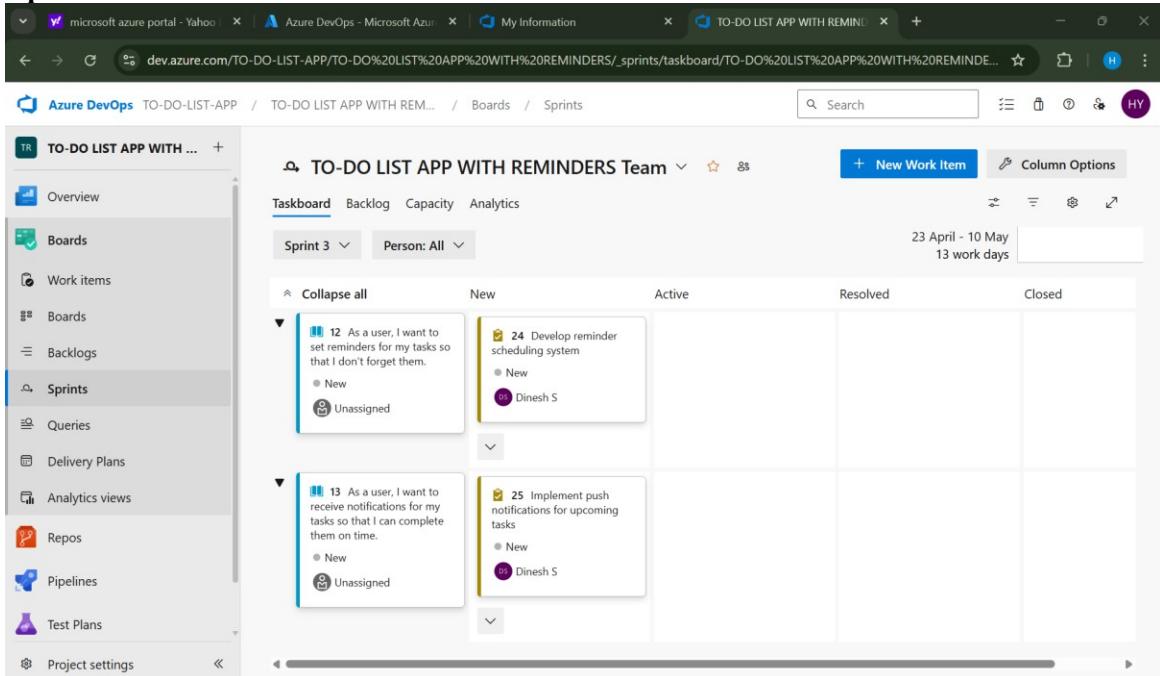
- User Story 6: As a user, I want to register and log in so that I can access my tasks securely. Status: New, Assigned to Dhanush R.
- User Story 18: Implement user authentication (Sign-up, Login, Logout). Status: New, Assigned to Dhanush R.
- User Story 7: As a user, I want to manage my profile so that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

Sprint 2

The screenshot shows the Azure DevOps interface for the 'TO-DO LIST APP WITH REMINDERS' project. The left sidebar is open with 'Boards' selected. The main area displays the 'Taskboard' for 'Sprint 2'. The board has columns for 'New', 'Active', 'Resolved', and 'Closed'. There are four user stories listed:

- User Story 9: As a user, I want to create a new task so that I can keep track of my work. Status: New, Unassigned.
- User Story 21: Develop UI & API for adding new tasks. Status: New, Assigned to 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, Unassigned.
- User Story 22: Implement functionality for modifying task details. Status: New, Assigned to Daksh Khinvasa...

Sprint 3

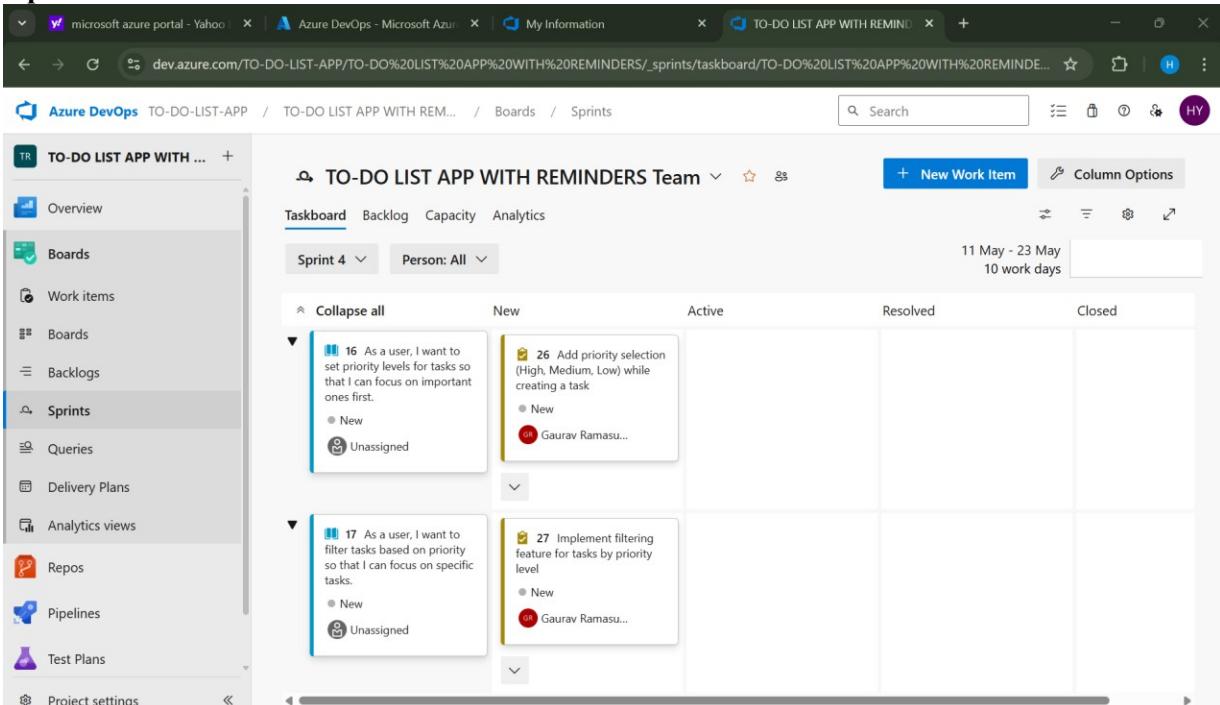


The screenshot shows the Azure DevOps Taskboard for the 'TO-DO LIST APP WITH REMINDERS Team' in 'Sprint 3'. The taskboard displays four items:

- 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 has columns for New, Active, Resolved, and Closed. The timeline at the top indicates the sprint runs from 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 timeline at the top indicates the sprint runs from 11 May - 23 May, covering 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

microsoft portal - Yahoo | Azure DevOps - Microsoft Azur | My Information | TO-DO LIST APP WITH REMINDERS | +

dev.azure.com/TO-DO-LIST-APP/TO-DO%20LIST%20APP%20WITH%20REMINDERS/_backlogs/backlog/TO-DO%20LIST%20APP%20WITH%20REMINDERS

Azure DevOps TO-DO-LIST-APP / TO-DO LIST APP WITH REMINDERS / Boards / Backlogs Search

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 Iteration: Updated by Harish Tutu YT: Feb 21

State: New Area: TO-DO LIST APP WITH REMINDERS Reason: New Iteration: TO-DO LIST APP WITH REMINDERS\Sprint 1

Details 0 2 0

Description	Planning	Deployment
Users should be able to sign up and log in using email and password authentication.	Story Points: 3 Priority: 1 Risk:	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

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

Value area: Business

Development

Add link

Link an Azure Repos [commit](#), [pull request](#) or [branch](#) to see the status of your

Discussion

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

Project settings

Result:

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

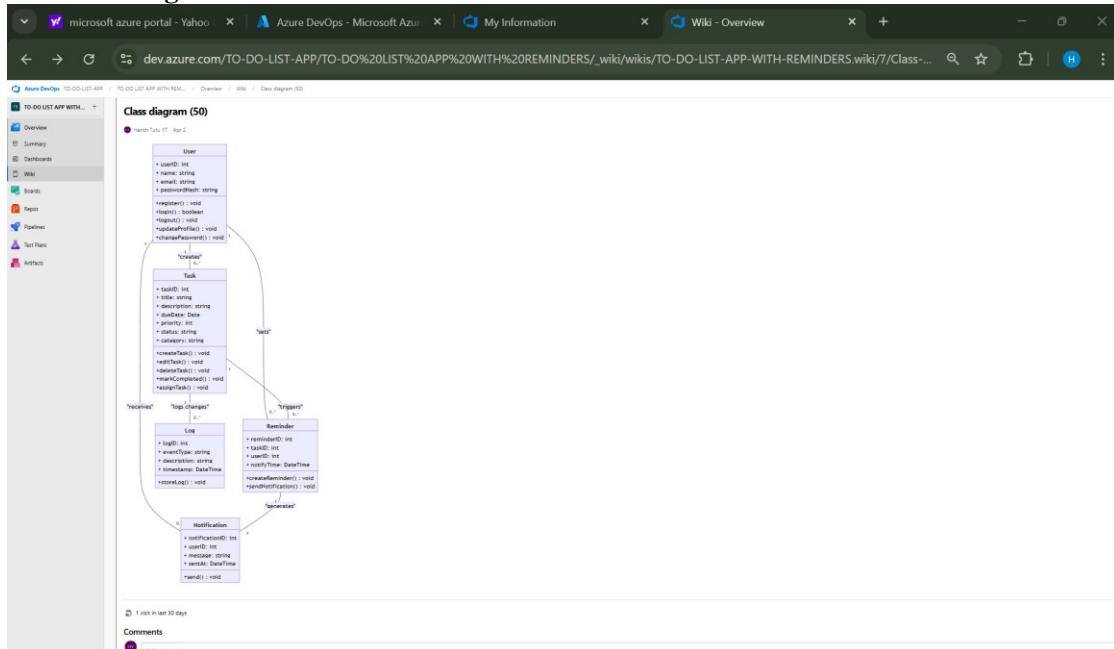
EXP NO: 6

DESIGNING CLASS AND SEQUENCE DIAGRAMS FOR PROJECT ARCHITECTURE

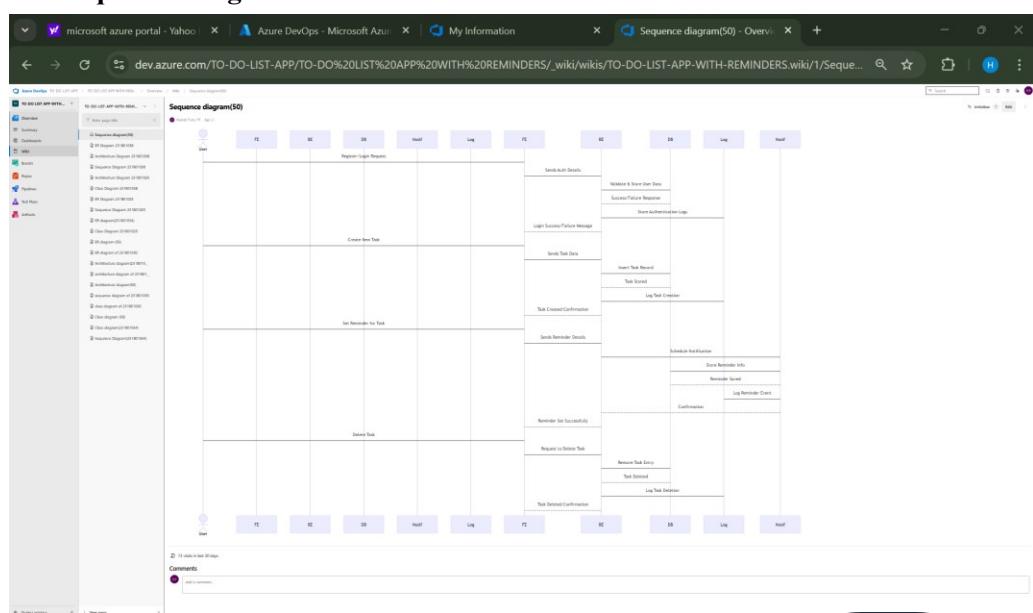
Aim:

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.

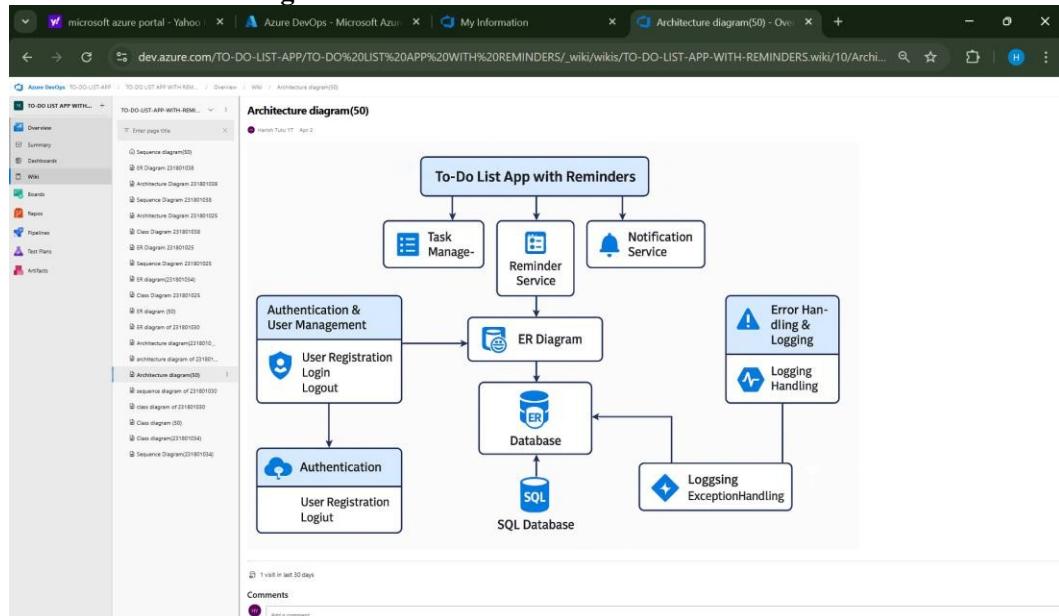
EXP NO: 7

DESIGNING ARCHITECTURAL AND ER DIAGRAMS FOR PROJECT STRUCTURE

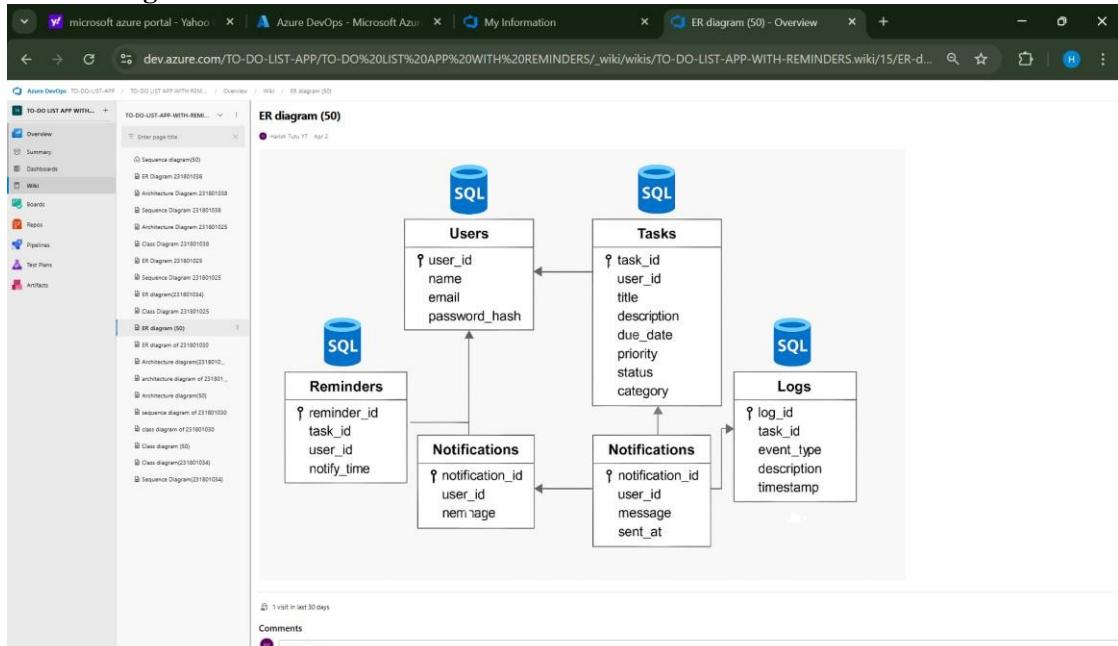
Aim:

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.

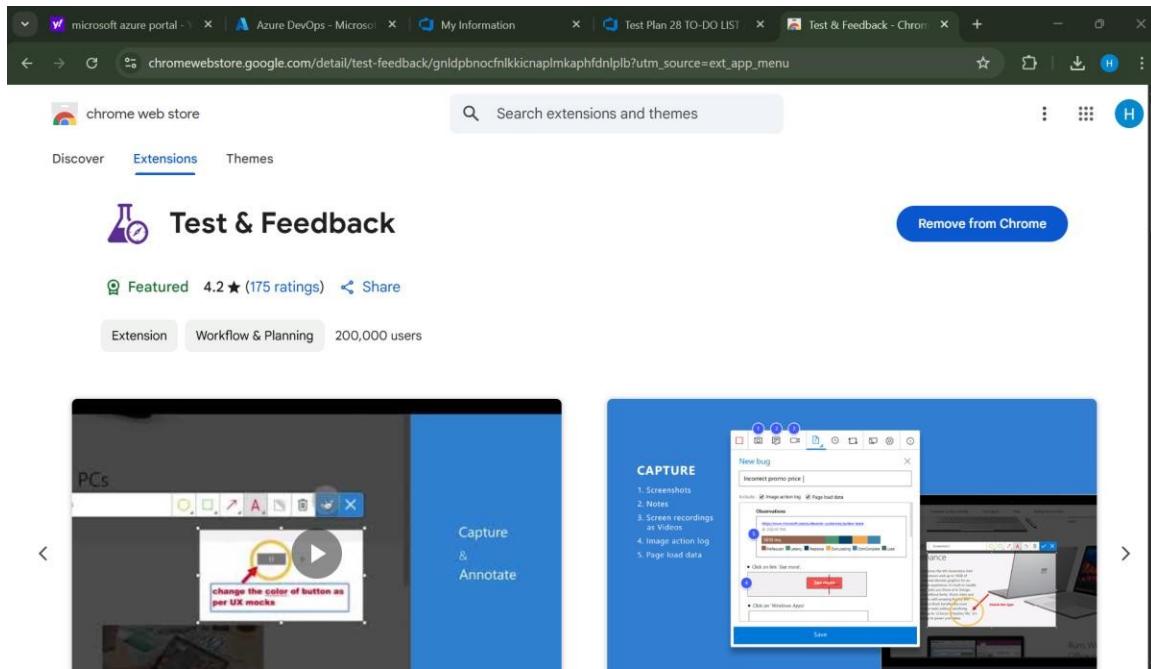
Action 3: Remove filter.

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 with their corresponding actions and expected results:

Steps	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 succes

Below the steps, there's 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, listing extensions with "Full access":

- Allow Copy - Select & Ena...
- Blend & Run: Enable copy...
- Test & Feedback** (highlighted)
- Manage extensions

Below the extensions, there's a note: "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](#)".

At the bottom of the modal, under "Development", there's a "Add link" section with instructions: "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."

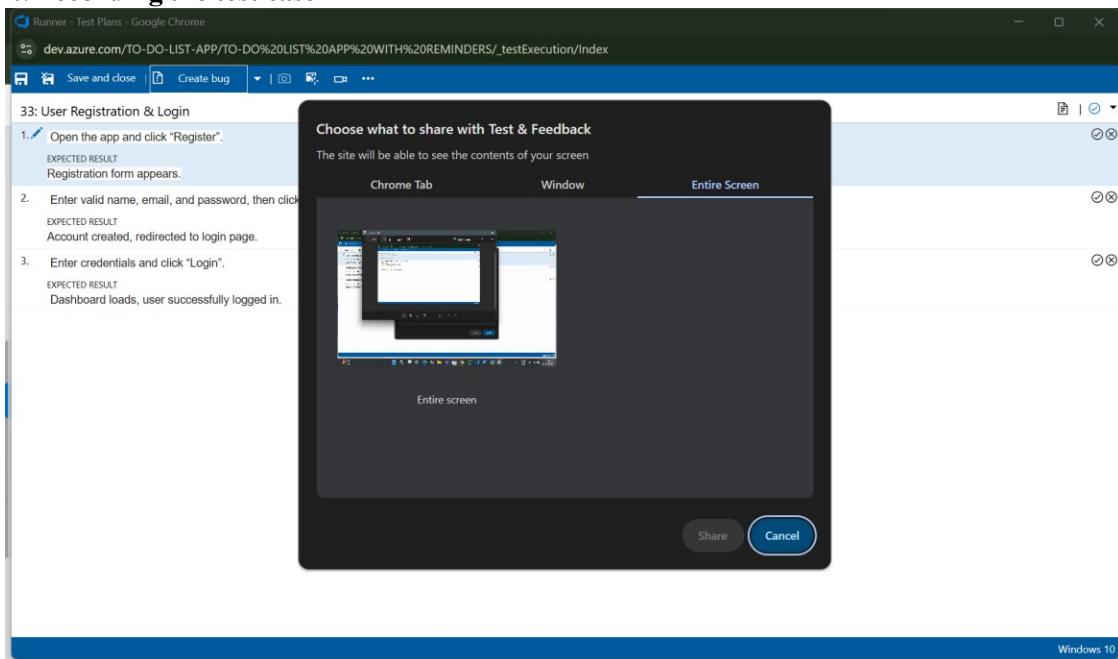
5. Running the test cases

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".

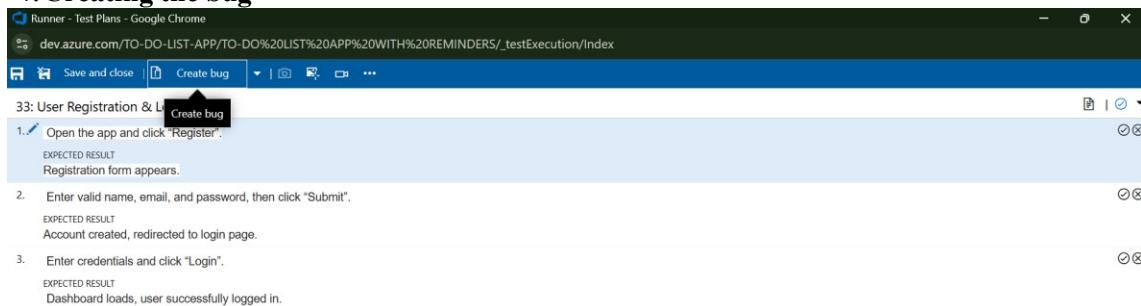
6.

The screenshot shows the "Runner - Test Plans" interface in Google Chrome. It displays a test case titled "33: User Registration & Login". Step 1, "Open the app and click "Register".", has an "EXPECTED RESULT" of "Registration form appears." Step 2, "Enter valid name, email, and password, then click "Submit".", has an "EXPECTED RESULT" of "Account created, redirected to login page." Step 3, "Enter credentials and click "Login".", has an "EXPECTED RESULT" of "Dashboard loads, user successfully logged in." The browser toolbar at the top includes "Save and close", "Create bug", and other standard browser controls.

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...

loading due to poor network

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

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...

loading due to poor network

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: 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

Details | ... | (2) |

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 bar includes 'Overview', 'Boards', 'Repos', 'Pipelines', 'Test Plans', 'Test plans' (which is selected), 'Progress report', 'Parameters', 'Configurations', 'Runs', and 'Artifacts'. Under 'Test plans', there are sections for 'Authentication (3)', 'Task management (3)', and 'Reminder & Notif...'. The main area displays 'Test Case Results' for the 'TO-DO LIST APP' test plan, dated April 15 - April 22. The results table has columns for Outcome (Passed, Failed, In Progress, Pending), TimeStamp, Configuration, Run by, Tester, and Test Plan ID. The data shows several successful runs (Passed) and one failed run (Failed). A link at the bottom says 'Open execution history for current test point'.

9. Test report summary

The screenshot shows a detailed view of a test case step titled 'Create Task'. The step is labeled '34 Create Task' and is assigned to 'Harish Tutu YT'. It has a status of 'Design' and a reason of 'Ready'. 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 'Expected result' column describes the outcome of each step. To the right, there are sections for 'Deployment' (with a note about tracking releases) and 'Development' (with a note about linking to Azure Repos). A sidebar on the left lists 'TEST CASE 34*' and a validation error: 'Field 'State' cannot be empty.'

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

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

Project settings

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:

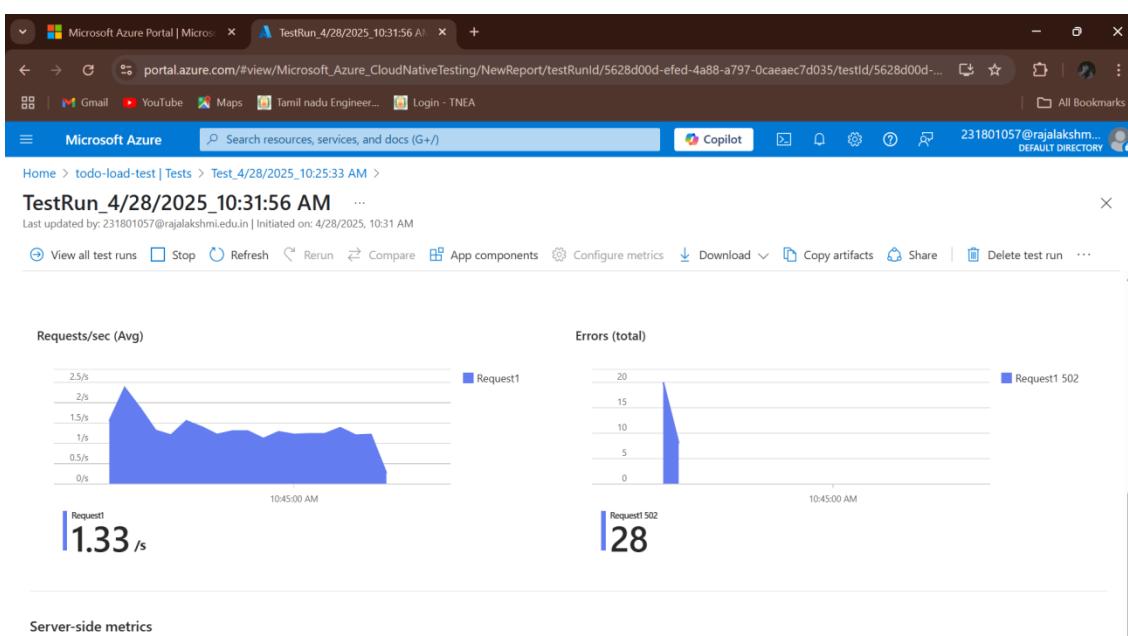
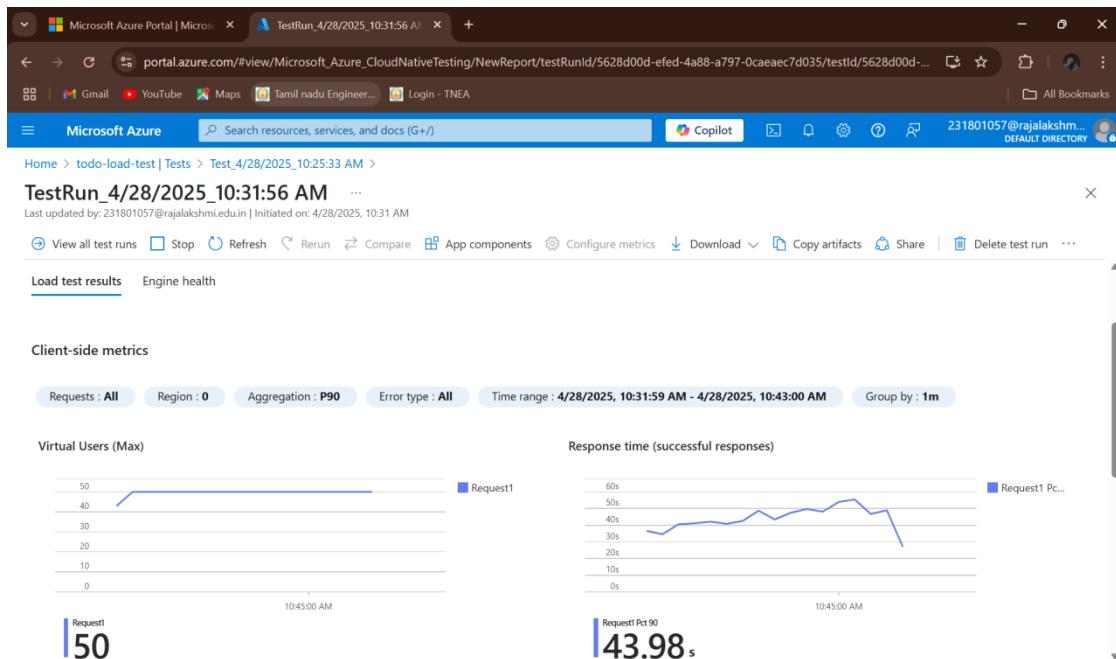
1. Sign in to Azure Portal
Go to <https://portal.azure.com> and log in.
2. Create the Resource
 - o Go to *Create a resource* → Search for “Azure Load Testing”.
 - o Select Azure Load Testing and click Create.
3. Fill in the Configuration Details
 - o *Subscription*: Choose your Azure subscription.
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

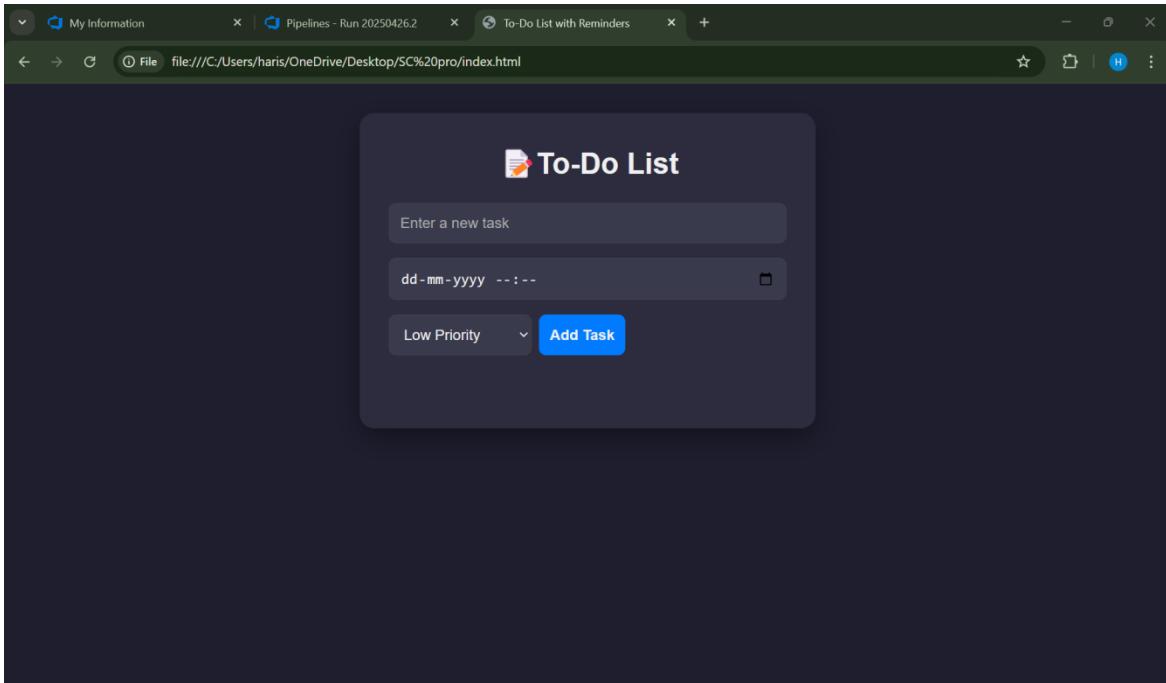
Load Testing



2116231801038



CamScanner
CS23432



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"
```

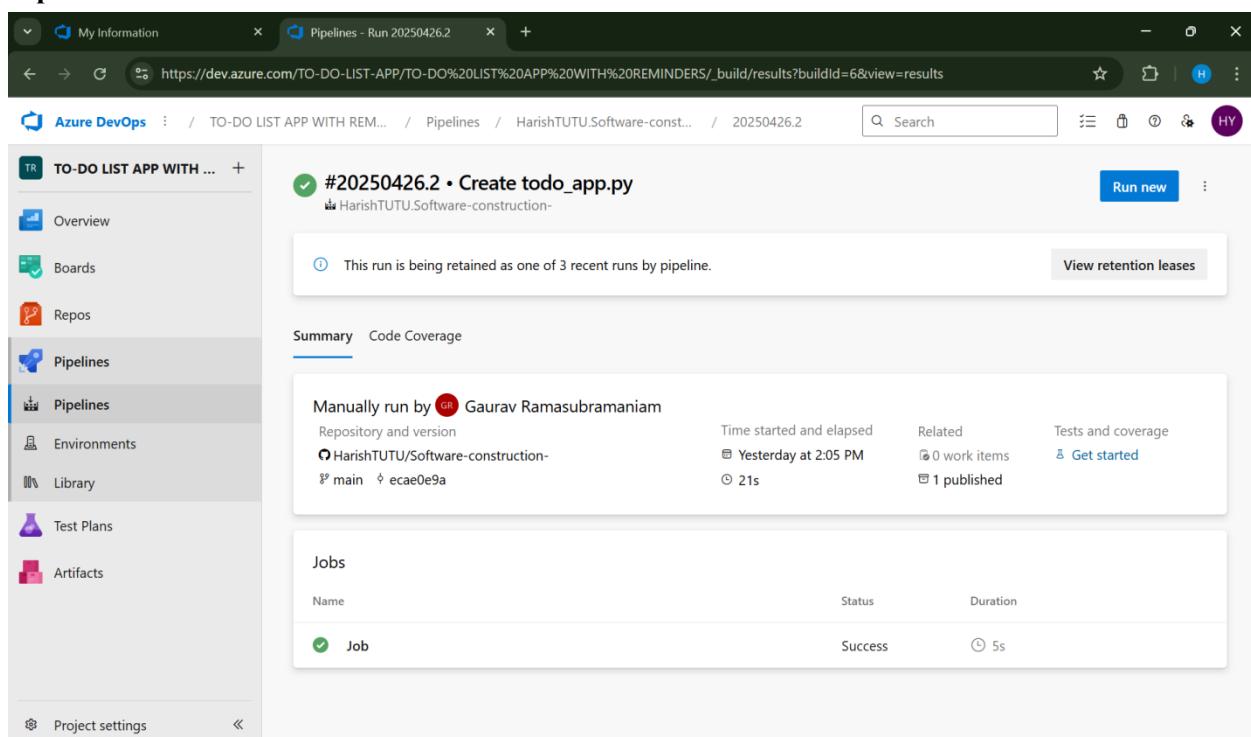
3. Pipeline Tasks Include:

- 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 test. The 'Jobs' section shows one job named 'Job' which was successful and completed 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.

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 the GitHub repository page for 'TO-DO-LIST-APP-WITH-REMINDERS'. The repository is public and has 1 branch and 0 tags. The main branch was last updated 1 minute ago. There are 10 commits by HarishTUTU. The repository has 0 stars, 1 watcher, and 0 forks. The 'About' section indicates no description, website, or topics provided. The 'Activity' section shows recent uploads of files like 'Architecture Diagram', 'Backlog', 'Pipelines', etc. The 'Releases' section shows no releases published. The 'Packages' section shows no packages published. The 'Languages' section shows HTML at 70.1% and Python at 29.9%.

File/Folder	Action	Last Updated
Architecture Diagram	Add files via upload	10 minutes ago
Backlog	Add files via upload	10 minutes ago
Pipelines	Add files via upload	10 minutes ago
Poker Estimation	Add files via upload	1 minute ago
Progress Report	Add files via upload	10 minutes ago
Project	Add files via upload	1 minute ago
Sequence Diagram	Add files via upload	10 minutes ago
Sprints	Add files via upload	1 minute ago
Test Plans And Test Cases	Add files via upload	10 minutes ago
azure-pipelines.yml	Rename azure-pipelines-1.yml to azure-pipelines.yml	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.