

EXP NO: 1

AZURE DEVOPS ENVIRONMENT SETUP

Aim:

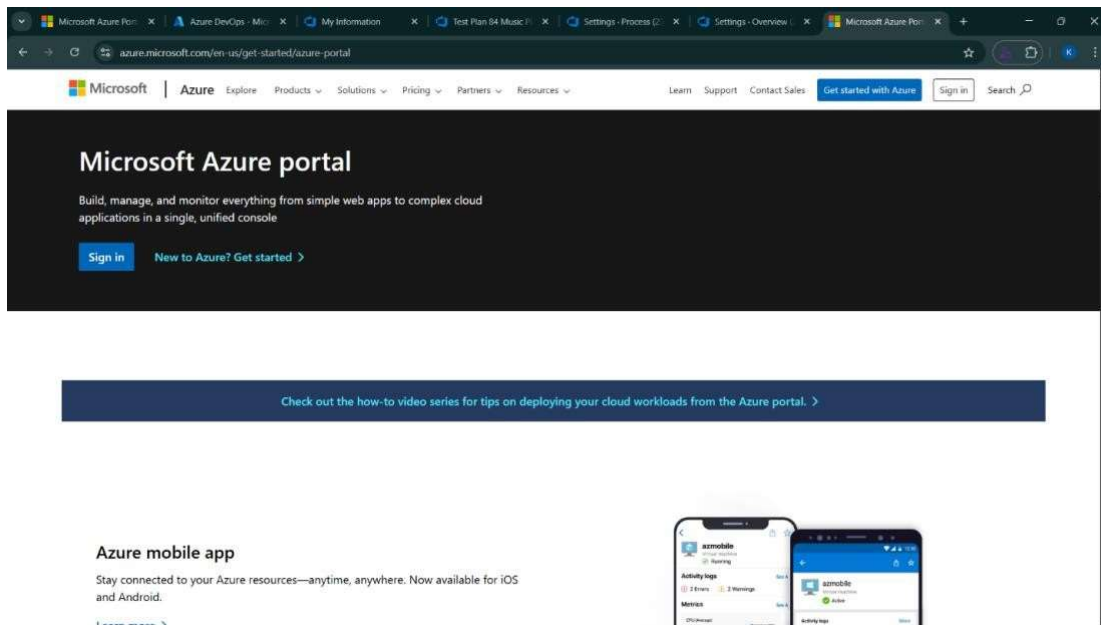
To set up and access the Azure DevOps environment by creating an organization through the Azure portal.

INSTALLATION

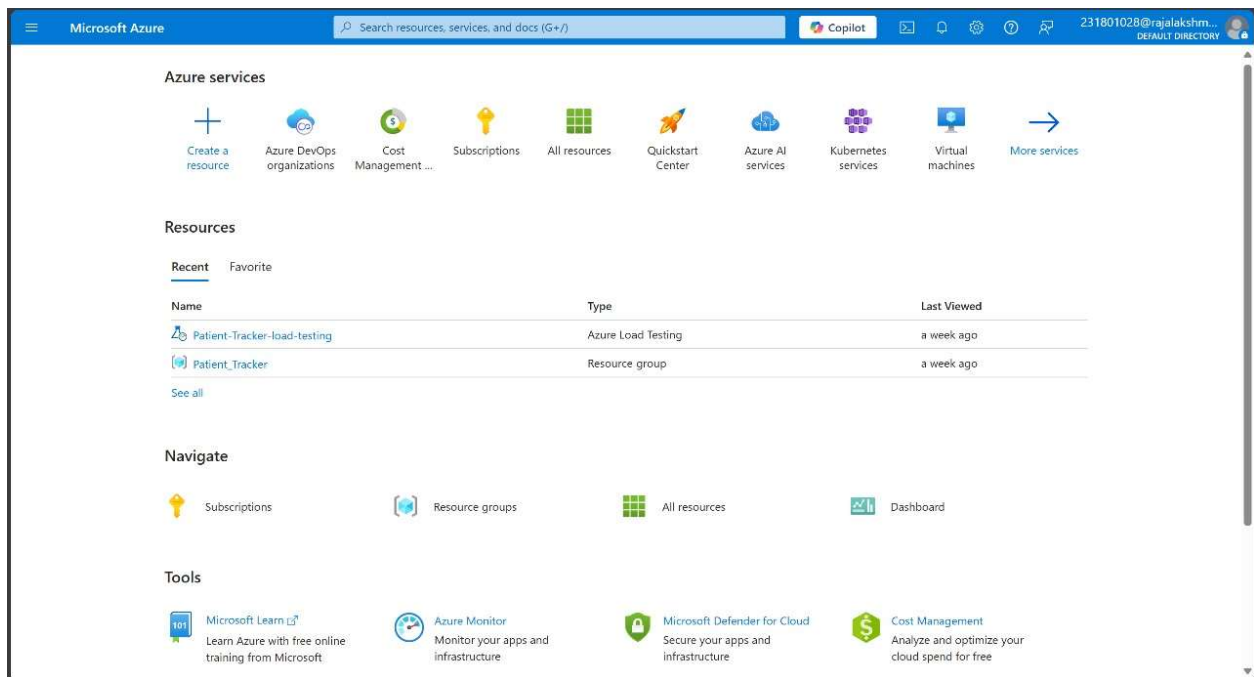
1. Open your web browser and go to the Azure website: <https://azure.microsoft.com/en-us/get-started/azure-portal>.

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



ent in the Azure platform by typing **Azure DevOps Organizations** in the search bar.

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

Microsoft Azure Search resources, services, and docs (G+/f) Copilot 231801028@rajalakshm... DEFAULT DIRECTORY

Home > Azure DevOps ...

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


Azure DevOps

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

[My Azure DevOps Organizations](#)

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

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



Result:

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

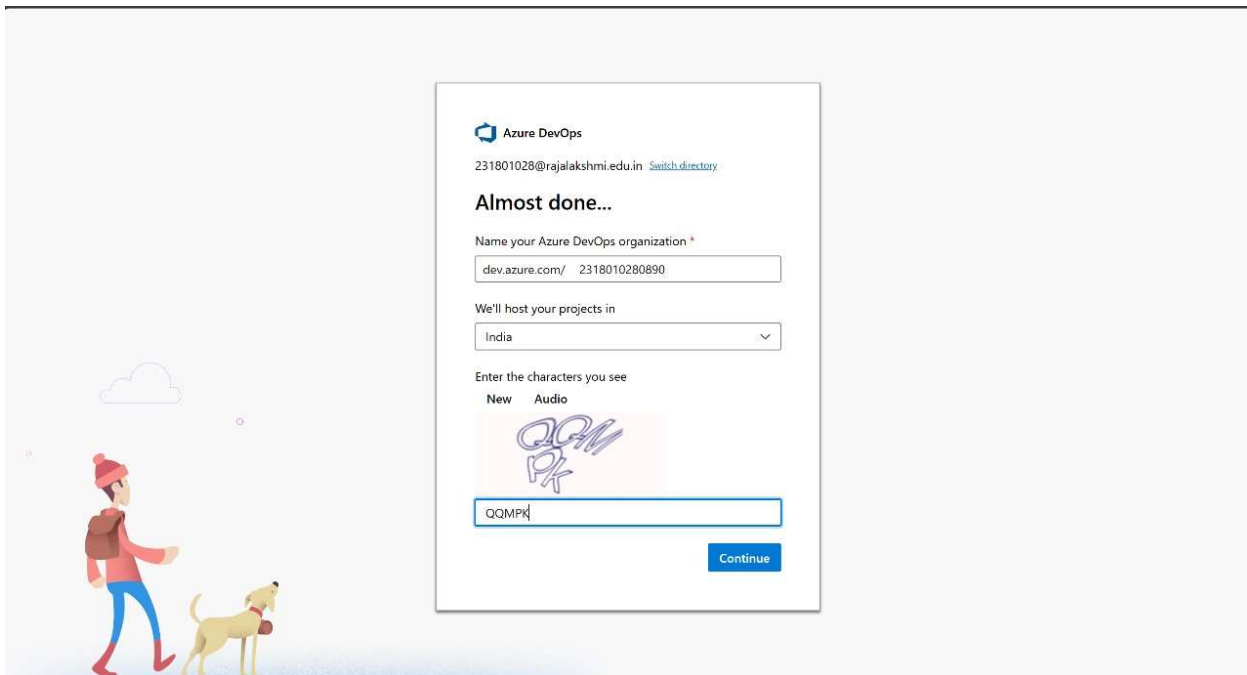
EXP NO: 2

AZURE DEVOPS PROJECT SETUP AND USER STORY MANAGEMENT

Aim:

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

1. Create An Azure Account



2. Create the First Project in Your Organization

- 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.
- On the organization's **Home page**, click on the **New Project** button.
- 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).
- Once you've filled out the details, click **Create** to set up your first project.

4. Project dashboard

The screenshot displays the Azure DevOps interface for a project named 'Patient Tracker Application'. The left sidebar contains navigation links: Overview, Summary, Dashboards, Wiki, Boards, Repos, Pipelines, Test Plans, and Artifacts. The main content area is titled 'Patient Tracker Application' and includes a 'Private' status and an 'Invite' button. The 'About this project' section prompts users to 'Help others to get on board!' and provides a button to 'Add Project Description'. The 'Project stats' section shows '0 Work Items created' and '0 Work Items completed' for the 'Last 7 days' period. The 'Pipelines' section shows '0%' completion and 'Builds succeeded'. The 'Members' section lists four team members: DM, PU, AT, and AM.

Azure DevOps Infinitycoders / Patient Tracker Application / Overview / Summary

Search

Patient Tracker Applica... +

Overview

Summary

Dashboards

Wiki

Boards

Repos

Pipelines

Test Plans

Artifacts

Project settings

PA Patient Tracker Application

Private Invite

About this project

Help others to get on board!

Describe your project and make it easier for other people to understand it.

+ Add Project Description

Project stats Period: Last 7 days

Boards

0 Work Items created

0 Work Items completed

Pipelines

0%

Builds succeeded

Members 4

DM PU AT AM

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 displays the Azure Boards interface for the 'Patient Tracker Application Team'. The left-hand navigation menu is visible, with 'Boards' selected. The main area shows the 'Backlog' view with a table of work items. The table has columns for Order, Work Item Type, Title, State, Effort, Business Value, Area, and Tags. Four work items are listed, all of type 'Epic' and state 'New', with titles related to user management, patient record management, monitoring, and admin tools.

Order	Work Item Type	Title	State	Effort	Busin...	Value Area	Tags
1	Epic	> 🏰 User Management & Authentication	🟢 New			Business	
2	Epic	> 🏰 Patient Record Management	🟢 New			Business	
3	Epic	> 🏰 Monitoring & Notifications	🟢 New			Business	
4	Epic	> 🏰 Admin Tools, Logging & Reporting	🟢 New			Business	

Result:

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

EXP NO: 3

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

Aim:

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

Create Epic, Features, User Stories, Task

The screenshot displays the Azure Boards interface for the 'Patient Tracker Application Team'. The left sidebar shows navigation options: Overview, Boards, Work items, Repos, Pipelines, Test Plans, and Artifacts. The main area is titled 'Patient Tracker Application Team' and includes a 'New Work Item' button. Below this, there's a 'Backlog' tab and a 'Work Items' section. A 'NEW EPIC' form is open, with a red error message: 'Field 'Title' cannot be empty.' The form has a 'Title' input field, a 'No one selected' dropdown, and an 'Add Tag' button. Below the form, there are sections for 'Description', 'Discussion', 'Planning', 'Deployment', and 'Development'. The 'Description' section has a placeholder 'Click to add Description.' The 'Discussion' section has a placeholder 'Add a comment. Use # to link a work item, @ to mention a person, or ! to link a pull request.' The 'Planning' section has fields for Priority (2), Risk, Effort, Business Value, Time Criticality, Start Date, and Target Date. The 'Deployment' section has a placeholder '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.' The 'Development' section has a placeholder '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.'

1. Fill in Epics

2.Fill in Features

Patient Tracker Applica... +

Overview

Boards

Work items

Boards

Backlogs

Sprints

Patient Tracker Applica... +

Overview

Boards

Work items

Boards

Backlogs

Sprints

Queries

Delivery Plans

Analytics views

Repos

Pipelines

Test Plans

Artifacts

Project settings <<

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

Work Items Back to Work Items

NEW FEATURE * Field 'Title' cannot be empty.

Enter title

No one selected 0 Comments Add Tag Save

State New Area Patient Tracker Application Reason New Iteration Patient Tracker Application Details 0

Description

Click to add Description.

Acceptance Criteria

Click to add Acceptance Criteria.

Discussion

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

Planning

Story Points

Priority 2 Risk

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 [commit](#), [pull request](#) or [branch](#) to see the status of your development. You can also [create a branch](#) to get started.

3. Fill in User Story Details

Result:

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

EXP NO: 4

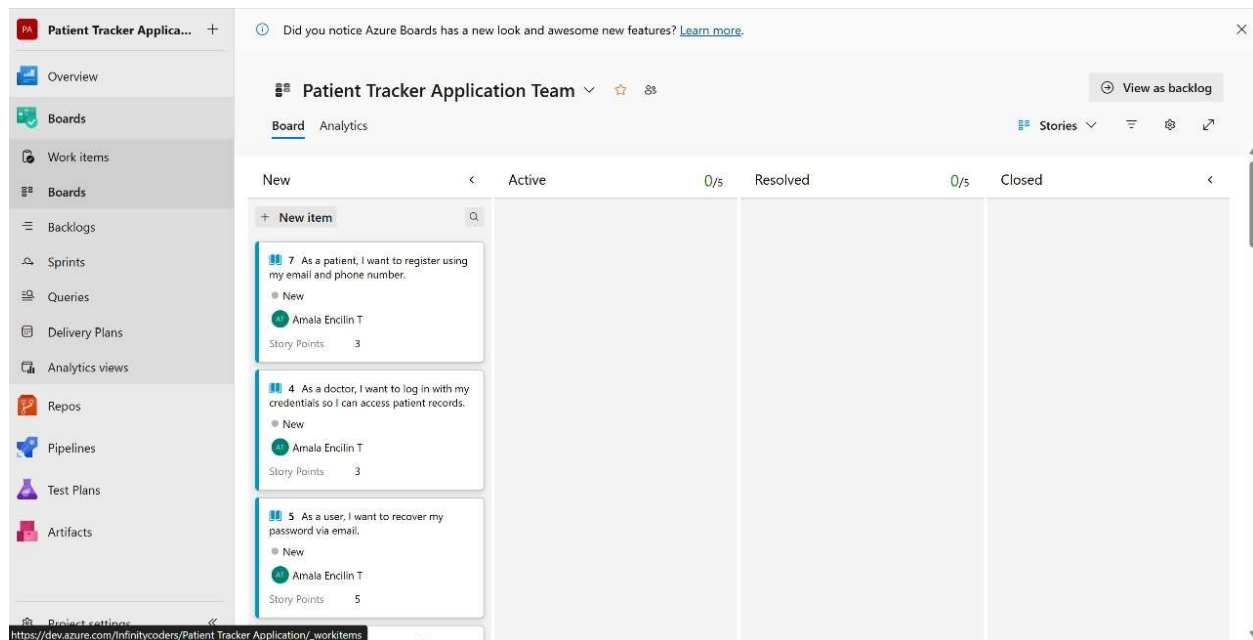
SPRINT PLANNING

Aim:

To assign user story to specific sprint for patient tracker application.

Sprint Planning

Sprint



Result:

The Sprints are created for the patient tracker application.

EXP NO: 5	POKER ESTIMATION
-----------	------------------

Aim:
 Create Poker Estimation for the user stories -Patient tracker application.

Poker Estimation

USER STORY 7

7 As a patient, I want to register using my email and phone number.

Amala Encilin T

0 Comments Add Tag

Save and Close

Follow

Updated by Deepika M: 4 Apr

Statg

New

Area

Patient Tracker Application

Reason

New

Iteration

Patient Tracker Application\sprint-1

Details

1

0

Description

Click to add Description.

Acceptance Criteria

1. The system must require mandatory fields (name, age, gender, contact, medical history).

2. Each patient should have a unique **Patient ID**, preventing duplicate entries.

3. Only **doctors** can add new patient records; unauthorized users get an error.

4. A success message should appear after saving, and the patient should be visible in the list.

5. The record should save within **2 seconds**, with an error message if it fails.

6. The system should log **who added the record and when** for tracking.

Planning

Story Points

3

Priority

2

Risk

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 [commit](#), [pull request](#) or [branch](#) to see the status of your development. You can also [create a branch](#) to get started.

Related Work

Result:
 The Estimation/Story

Points is created for the project using Poker Estimation.

2116231801028

CS23432

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

- Patient Tracker Applica... +
- Overview
- Summary
- Dashboards
- Wiki
- Boards
- Repos
- Pipelines
- Test Plans
- Artifacts

Instant Search: You can instantly search wiki pages by activating the search box.

Patient-Tracker-Application.wiki

Enter page title

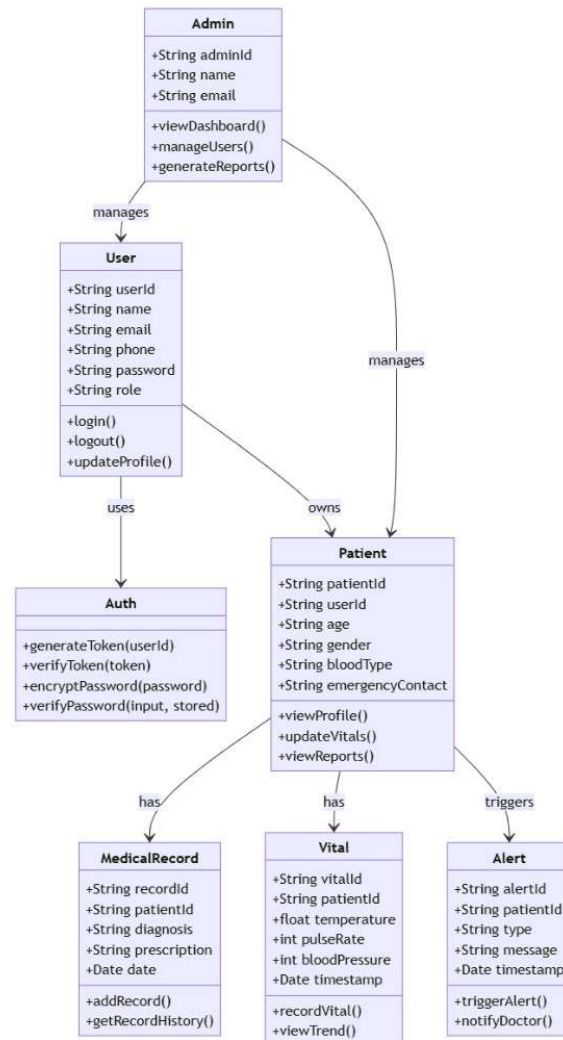
Architecture Diagram

Sequence Diagram

Class Diagram

Class Diagram

Deepika M Apr 4



6 visits in last 30 days

6B.
Seque

nce Diagram

Patient Tracker Applica... + Instant Search: You can instantly search wiki pages by activating the search box.

- Overview
- Summary
- Dashboards
- Wiki
- Boards
- Repos
- Pipelines
- Test Plans
- Artifacts

Patient-Tracker-Application.wiki

Enter page title

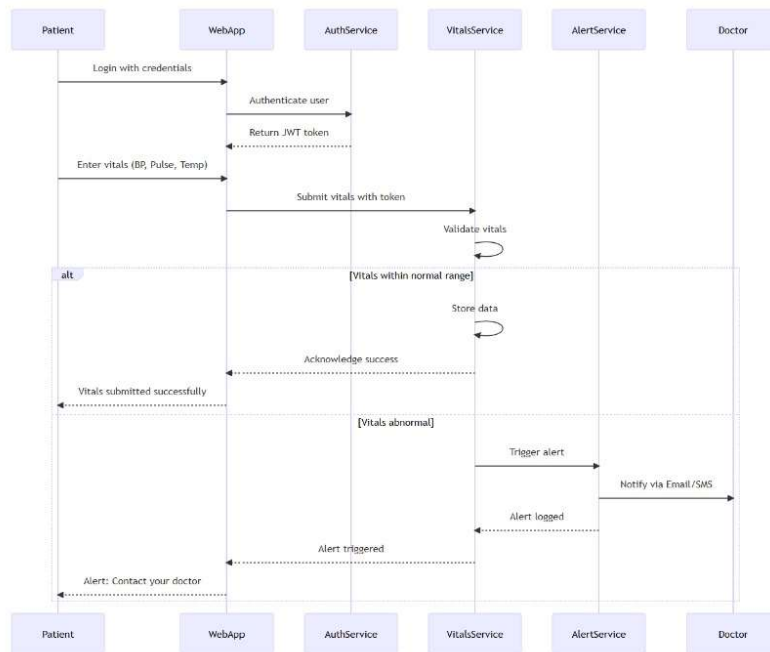
Architecture Diagram

Sequence Diagram

Class Diagram

Sequence Diagram

Deepika M Apr 4



6 visits in last 30 days

Result:

The Class Diagram and Sequence Diagram is designed Successfully for the patient tracker application.

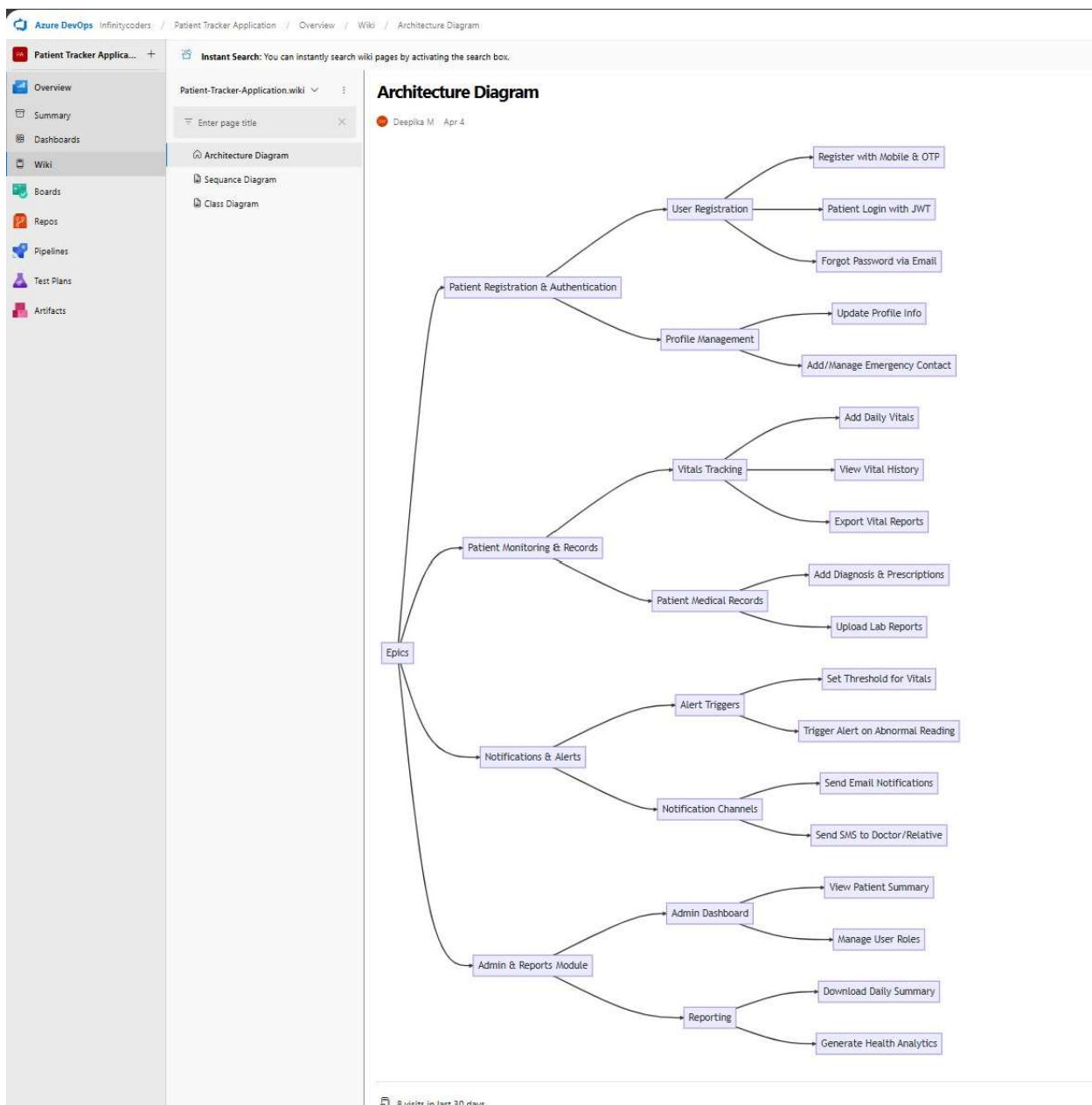
EXP NO: 7

DESIGNING ARCHITECTURAL AND ER DIAGRAMS FOR PROJECT STRUCTURE

Aim:

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

Architectural Diagram



Result:

The Architecture Diagram and ER Diagram is designed Successfully for the patient tracker application.

EXP NO: 8	TESTING – TEST PLANS AND TEST CASES
------------------	--------------------------------------------

Aim:

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

Test Planning and Test Case

Test Case Design Procedure

1. Understand Core Features of the Application

- User Signup & Login
- Viewing and Managing Playlists
- Fetching Real-time Metadata
- Editing playlists (rename, reorder, record)
- Creating smart audio playlists based on categories (mood, genre, artist, etc.)

2. Define User Interactions

- Each test case simulates a real user behaviour (e.g., logging in, renaming a playlist, adding a song).

3. Design Happy Path Test Cases

- Focused on validating that all features function as expected under normal conditions.
- Example: User logs in successfully, adds item to playlist, or creates a category-based playlist.

4. Design Error Path Test Cases

- Simulate negative or unexpected scenarios to test robustness and error handling.
- Example: Login fails with invalid credentials, save fails when offline, no recommendations found.

5. Break Down Steps and Expected Results

- Each test case contains step-by-step actions and a corresponding expected outcome.
- Ensures clarity for both testers and automation scripts.

6. Use Clear Naming and IDs

- Test cases are named clearly (e.g., TC01 – Successful Login, TC10 – Save Playlist Fails).
- Helps in quick identification and linking to user stories or features.

7. Separate Test Suites

- Grouped test cases based on functionality (e.g., Login, Playlist Editing, Recommendation System).

- Improves organization and test execution flow in Azure DevOps.

8. Prioritize and Review

- Critical user actions are marked high-priority.
- Reviewed for completeness and traceability against feature requirements.

1. New test plan

The screenshot displays the Azure DevOps web interface for creating a new test plan. On the left, a sidebar shows navigation options: Overview, Boards, Repos, Pipelines, Test Plans (selected), Progress report, Parameters, Configurations, Runs, and Artifacts. The main area is titled 'New Test Plan' and includes a 'Name' field with the text 'Patient Tracker Application-Test Plan' and an 'Area Path' field. Below this, a 'Patient Tracker' section shows a timeline with dates Apr 20 and Apr 27, and a status '100% run, 100% passed. View report'. A 'Test Suites' section is visible with a filter 'Filter suites by name' and a list containing 'Patient Tracker Appl...'. The rightmost section, 'Patient Tracker Application (ID: 41)', has tabs for 'Define', 'Execute', and 'Chart'. The 'Execute' tab is active, showing a table of 'Test Points (5 items)' with columns for Title, Outcome, Order, and Test Case Id. All test points are marked as 'Passed'.

Title	Outcome	Order	Test Case Id
User Login	Passed	1	42
Registration with Missing Fields	Passed	2	43
Add Vitals for Patient	Passed	3	44
View Patient History	Passed	4	45
Logout Functionality	Passed	5	46

2. Test 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.

Music Playlist Batch Creator – Test Plans

USER STORIES

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

Test Suites

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

1. TC01 – Successful Sign Up

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

2. TC02 – Secure Login

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

3. TC03 – Sign Up with Existing Email

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

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

4. TC04 – Login with Wrong Password

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

Test Suit: TS02 - View Playlists (ID: 87)

1. TC05 – View Playlist Page

- **Action:**
 - Log in successfully.
 - Navigate to "My Playlists" section.
- **Expected Results:**
 - All created playlists are displayed clearly.
- **Type:** Happy Path

2. TC06 – Playlist Loading Failure

- **Action:**
 - Disconnect from the internet.
 - Navigate to "My Playlists".
- **Expected Results:**
 - Network is offline.
 - Error message "Unable to load playlists" is shown.
- **Type:** Error Path

Test Suit: TS03 - Real-Time Metadata (ID: 88)

1. TC07 – Real-Time Metadata Display

- **Action:**
 - Play a song.
 - Observe the metadata panel.
- **Expected Results:**
 - Metadata (title, artist, album, duration) is displayed and updates in real time.
- **Type:** Happy Path

2. TC08 – Metadata Not Updating

- **Action:**

- Play a different song.
 - Observe the metadata panel.
- **Expected Results:**
 - Metadata remains static or shows default/fallback message.
- **Type:** Error Path

Test Suit: TS04 - Playlist Editing (ID: 89)

1. TC09 – Rename Playlist Successfully

- **Action:**
 - Navigate to "My Playlists".
 - Click "Rename" next to a playlist.
 - Enter a new name and click "Save".
- **Expected Results:**
 - Playlist name updates successfully.
- **Type:** Happy Path

2. TC10 – Rename with Blank Name

- **Action:**
 - Click "Rename" on a playlist.
 - Leave the field blank.
 - Click "Save".
- **Expected Results:**
 - Error message "Playlist name cannot be empty" is shown.
- **Type:** Error Path

3. TC11 – Change Playlist Order

- **Action:**
 - Open a playlist.
 - Drag and drop songs to reorder.
 - Click "Save".
- **Expected Results:**
 - Playlist order is updated and saved.
- **Type:** Happy Path

4. TC12 – Change Playlist Order Fails

- **Action:**
 - Login and go to "My Playlists".
 - Select a playlist.
 - Go offline or simulate server error.
 - Reorder songs and click "Save Order".
- **Expected Results:**
 - Error message: "Failed to update order. Please check your connection".
- **Type:** Error Path

Test Suit: TS05 - Smart Playlist Creation (ID: 90)

1. TC13 – Generate Playlist Based on Various Categories

- **Action:**
 - Login with valid credentials.
 - Click on "Generate Playlist".
 - Select categories.
 - Click "Generate Playlist".
- **Expected Results:**
 - Playlist is generated based on selected mood and categories.
- **Type:** Happy Path

2. TC14 – Fail to Generate Playlist Due to Missing Category Selection or Invalid Input

- **Action:**
 - Login with valid credentials.
 - Click on "Generate Playlist".
 - Select categories.
 - Click "Generate Playlist".
- **Expected Results:**
 - Error message: "Please select at least one valid category" or "No recommendations found for the selected filters".
- **Type:** Error Path

Test Cases

The screenshot displays a 'NEW TEST CASE' form in a software testing tool. The title bar shows 'NEW TEST CASE *'. The main form has a title field containing 'User Login'. Below the title, there's a user profile for 'Deepika M' and a '0 Comments' section with an 'Add Tag' button. A 'Save and Close' button is visible. The form is divided into sections: 'Statg' (Design), 'Area' (Patient Tracker Application), 'Reason' (New), and 'Iteration' (Patient Tracker Application). The 'Steps' section is active, showing a table with columns 'Steps' and 'Action'. The first step is numbered '1.' and the second is numbered '2.'. Below the table, there's a prompt 'Click or type here to add a step'. The right sidebar contains sections for 'Deployment', 'Development', and 'Related Work', each with an 'Add link' button and a description of how to link related items.

NEW TEST CASE *

User Login

Deepika M 0 Comments Add Tag Save and Close

Statg Design Area Patient Tracker Application Reason New Iteration Patient Tracker Application

Steps Summary Associated Automation 0 0

Steps Action Expected result

1. |

2. |

Click or type here to add a step

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

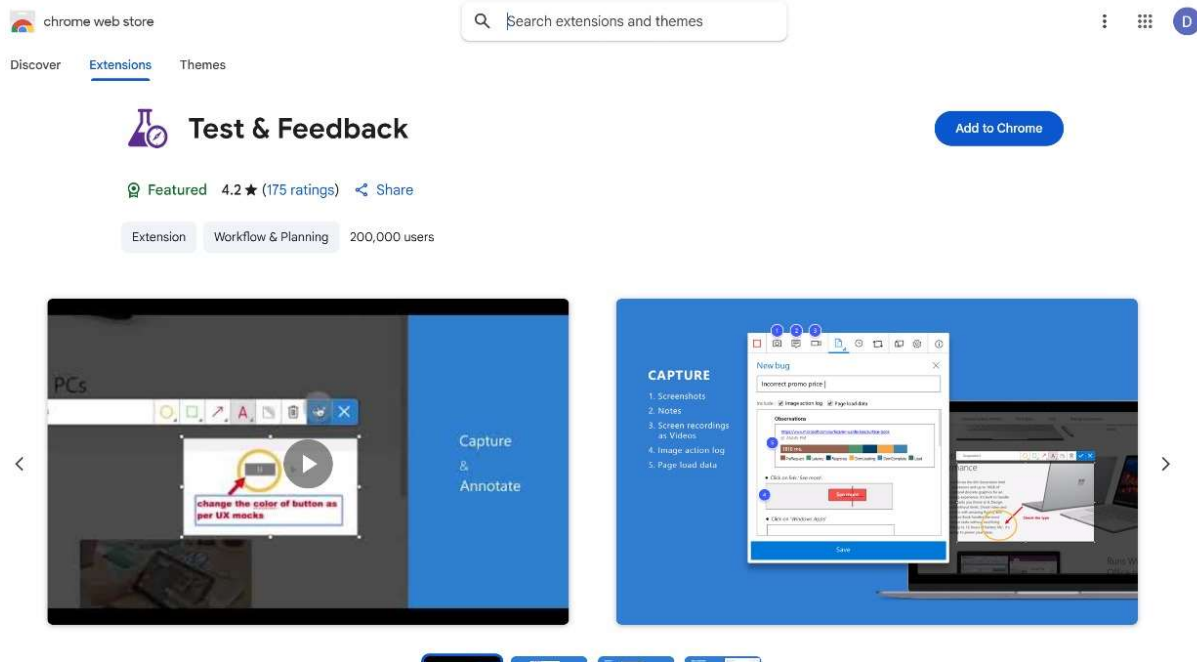
Link an Azure Repos [commit](#), [pull request](#) or [branch](#) to see the status of your development. You can also [create a branch](#) to get started.

Related Work

Add link

[Add an existing work item as a parent](#)

4. Installation of test



Test and feedback
Showing it as an extension

5. Running the test cases

Patient Tracker Application (ID: 41)

Define Execute Chart

Test Points (5 items)

Title	Outcome	Order	Test Case Id
<input checked="" type="checkbox"/> User Login	Passed	1	42
<input type="checkbox"/> Registration with Missing Fields	Passed	2	43
<input type="checkbox"/> Add Vitals for Patient	Passed	3	44
<input type="checkbox"/> View Patient History	Passed	4	45
<input type="checkbox"/> Logout Functionality	Passed	5	46

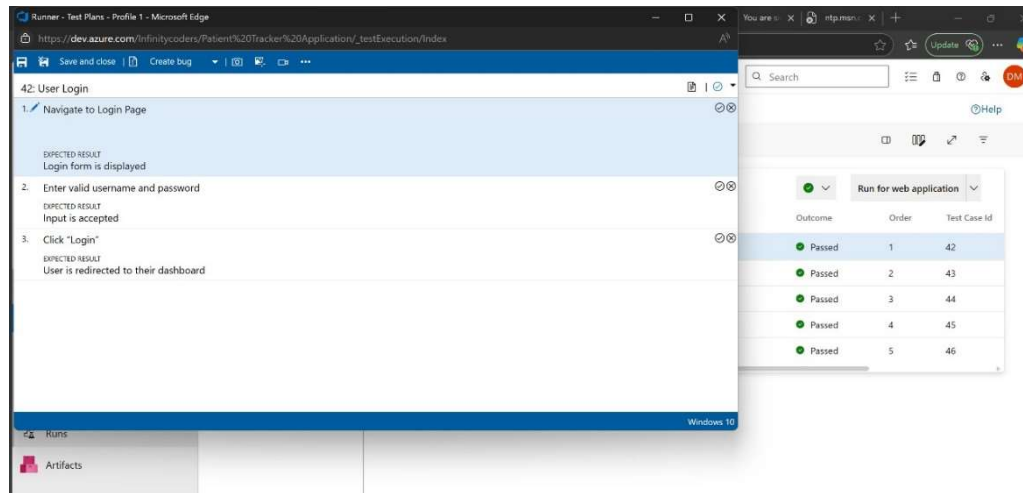
Run for web application

View execution history
Mark Outcome
Run
Reset test to active
Edit test case
Assign tester
View test result

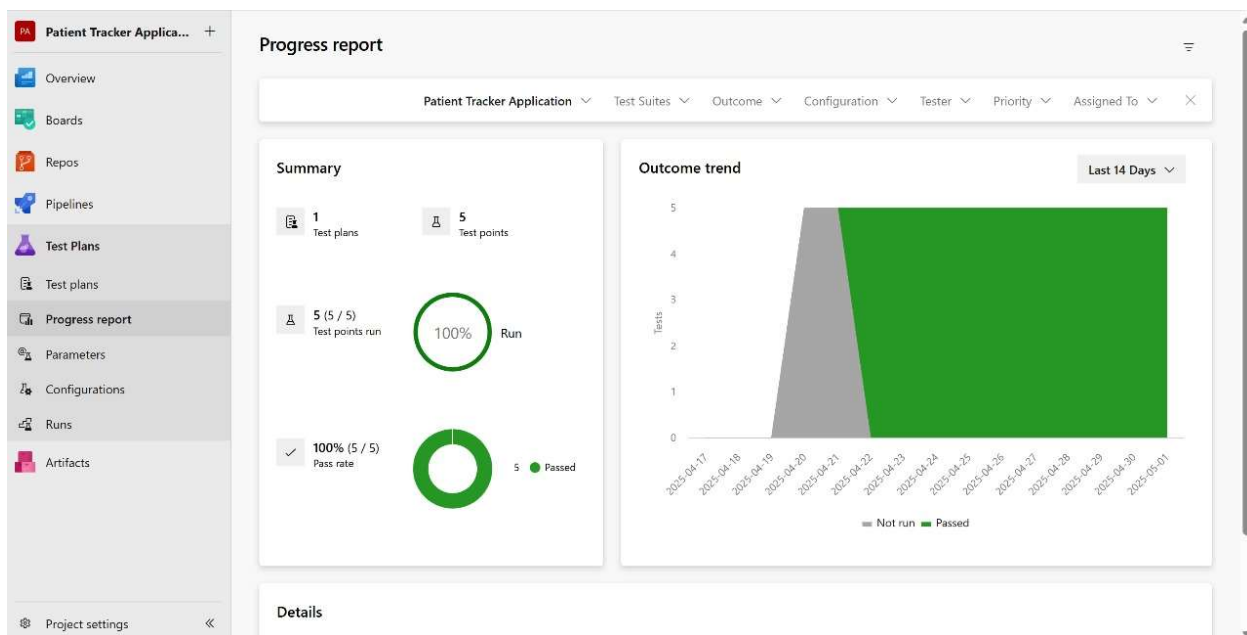
2116231801028

CS23432

Recording test case



Progress report



Result:

The test plans and test cases for the user stories is created in Azure DevOps with Happy Path and Error Path

EXP NO: 9

LOAD TESTING AND PERFORMANCE TESTING

Aim:

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

Load Testing

Steps to Create an Azure Load Testing Resource:

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

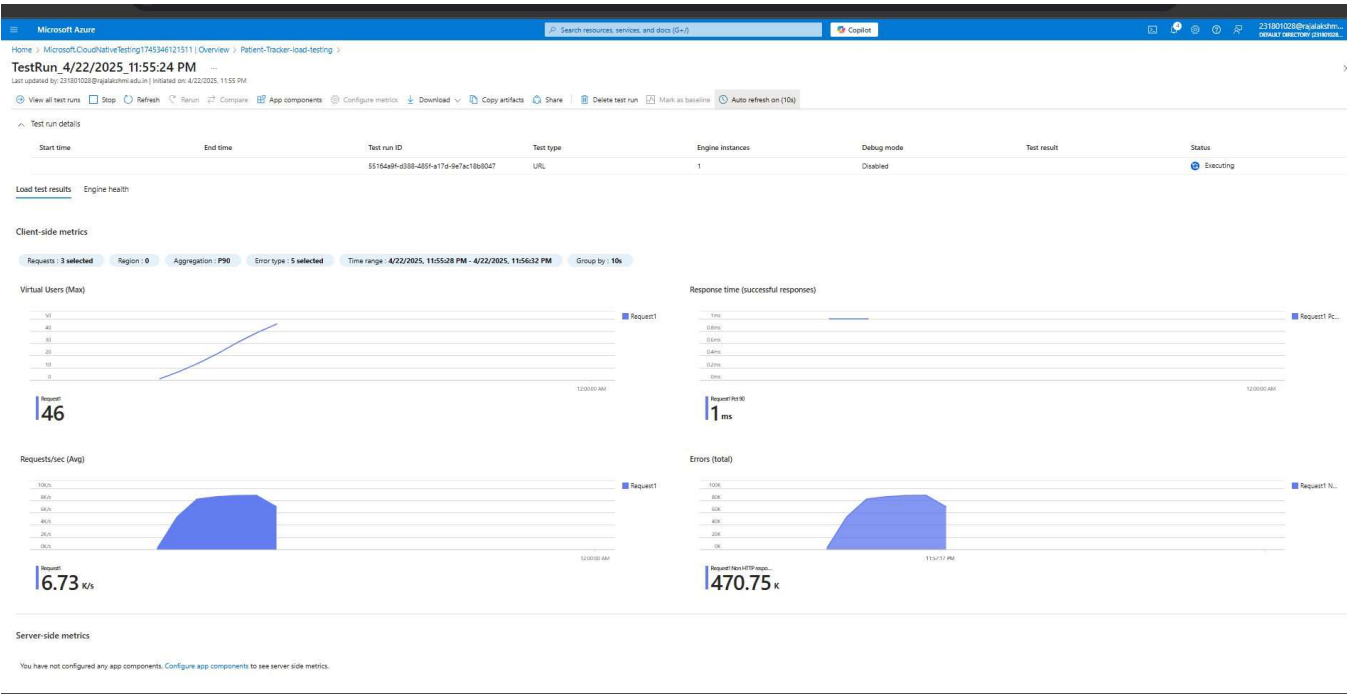
1. Sign in to Azure Portal
Go to <https://portal.azure.com> and log in.
2. Create the Resource
 - Go to *Create a resource* → Search for “Azure Load Testing”.
 - Select Azure Load Testing and click Create.
3. Fill in the Configuration Details
 - *Subscription*: Choose your Azure subscription.
 - *Resource Group*: Create new or select an existing one.
 - *Name*: Provide a unique name (no special characters).
 - *Location*: Choose the region for hosting the resource.
4. (Optional) Configure tags for categorization and billing.
5. Click Review + Create, then Create.
6. Once deployment is complete, click Go to resource.

Steps to Create and Run a Load Test:

Once your resource is ready:

1. Go to your Azure Load Testing resource and click Add HTTP requests > Create.
2. Basics Tab
 - *Test Name*: Provide a unique name.
 - *Description*: (Optional) Add test purpose.
 - *Run After Creation*: Keep checked.
3. Load Settings
 - *Test URL*: Enter the target endpoint (e.g., <https://yourapi.com/products>).
4. Click Review + Create → Create to start the test.

Load Testing



+ Add New Patient

Patient Name

Condition / Note

Add Patient

Dashboard

Total Patients: 58

Upcoming Appointments: 12

Alerts: 3

Patient Records

John Doe

Condition: Hypertension | Last Visit: 12-Apr-2025

Jane Smith

Condition: Diabetes Type 2 | Last Visit: 18-Apr-2025

Rahul Khanna

Condition: Asthma | Last Visit: 10-Apr-2025

Pipelines:

PA Patient Tracker Applica... +

Overview

Boards

Repos

Pipelines

Pipelines

Environments

Library

Test Plans

Artifacts

Project settings <<

Pipelines

Recent All Runs

Filter pipelines

Recently run pipelines

Pipeline	Last run
<div>Deepika26072006.Patient_Tracker_App...</div> <div>#20250423.1 • Set up CI with Azure Pipelines</div> <div>Individual CI for main</div>	<div>Apr 23</div> <div>< 1s</div>

2116231801028

CS23432

Result:

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

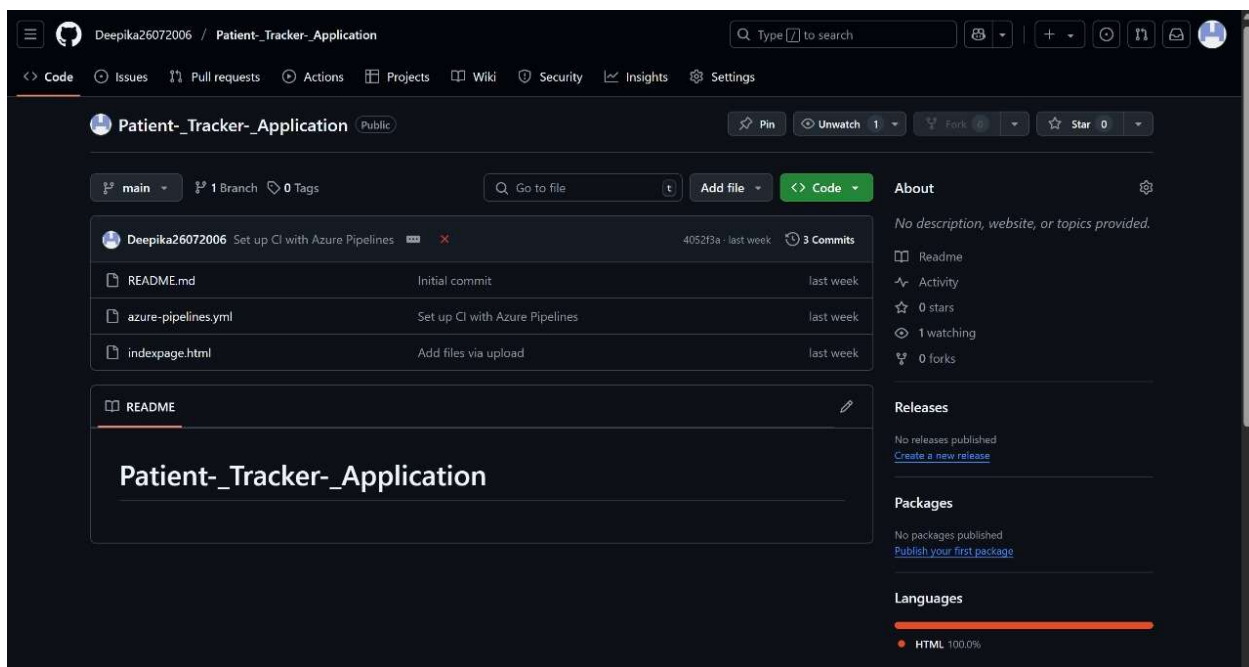
EXP NO: 10

GITHUB: PROJECT STRUCTURE & NAMING CONVENTIONS

Aim:

To provide a clear and organized view of the project's folder structure and file naming conventions, helping contributors and users easily understand, navigate, and extend the Patient tracker application.

GitHub Project Structure



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.

