EXP NO: 1	AZURE DEVOPS ENVIRONMENT SETUP

Aim:

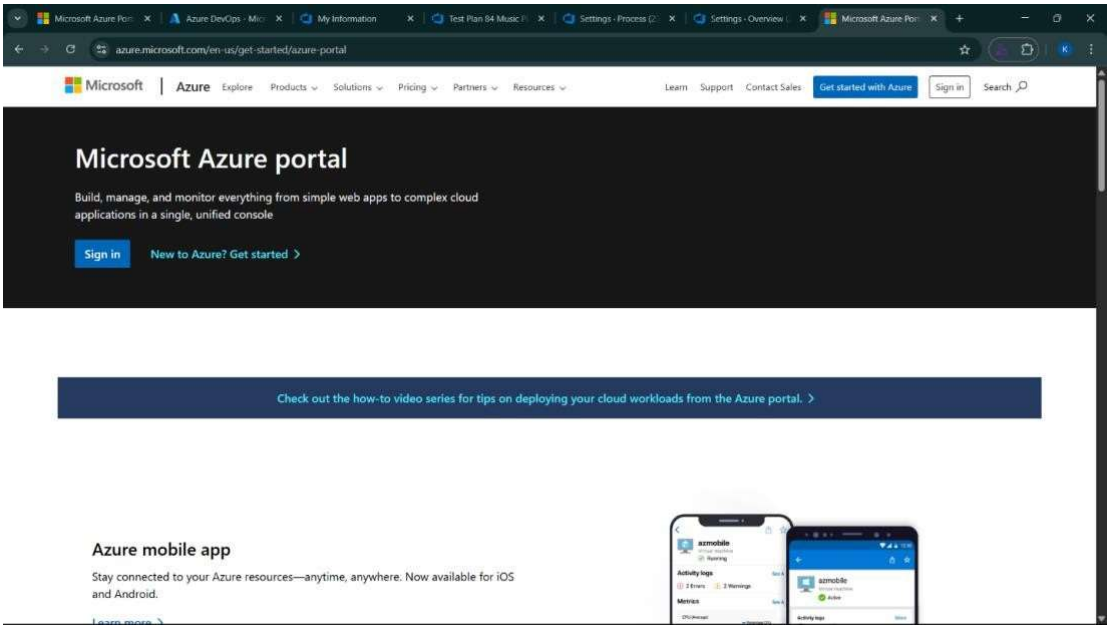
To set up and access the Azure DevOps environment by creating an organization through the Azure portal.

INSTALLATION

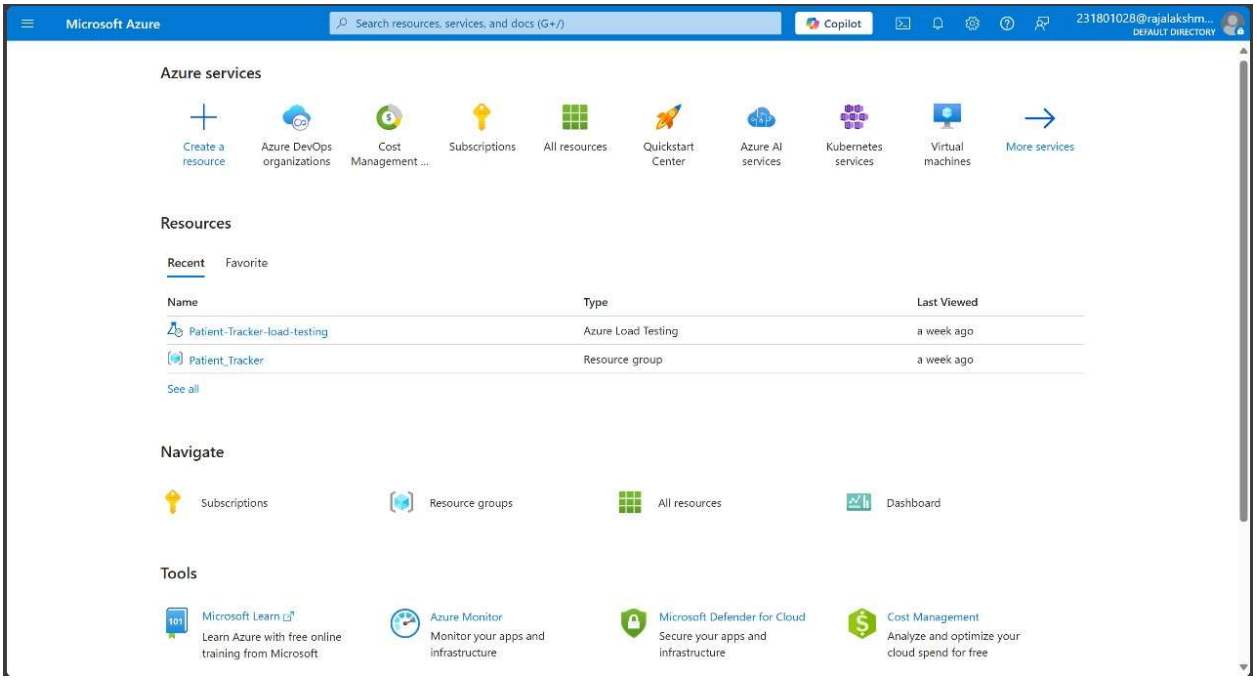
1. Open your web browser and go to the Azure website: https://azure.microsoft.com/en-us/get-started/azure-portal.

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

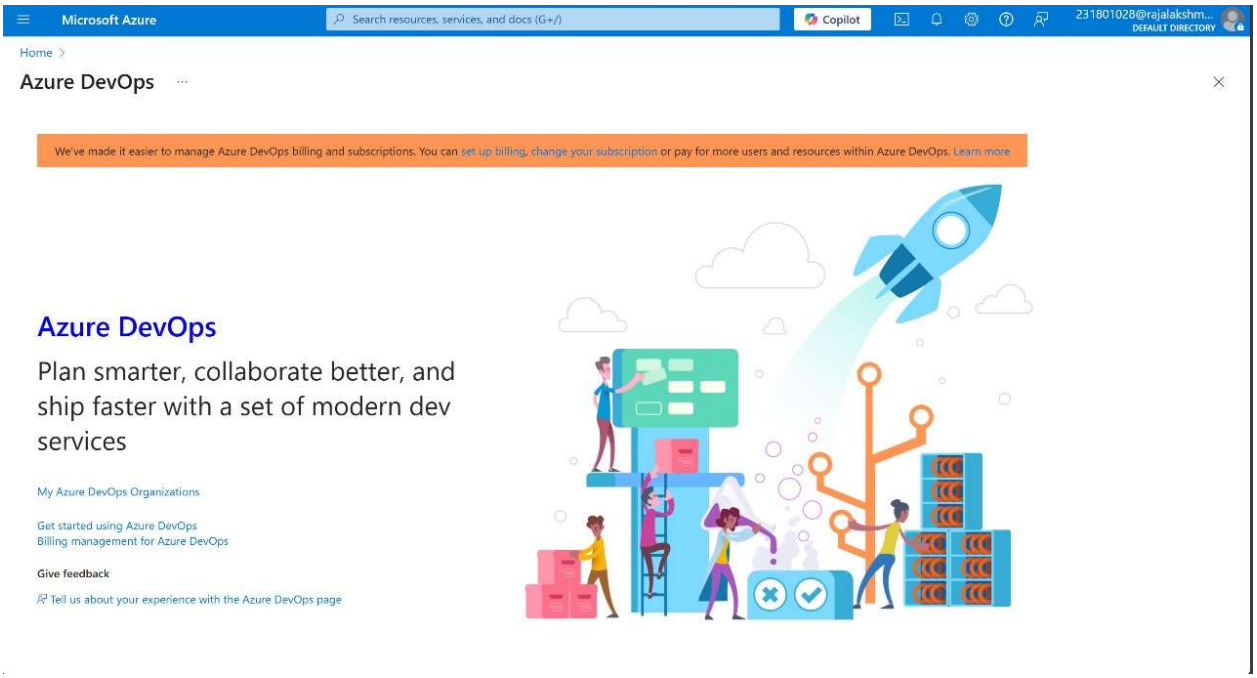


2. Azure home page



3. Open DevOps environment 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.



Result:

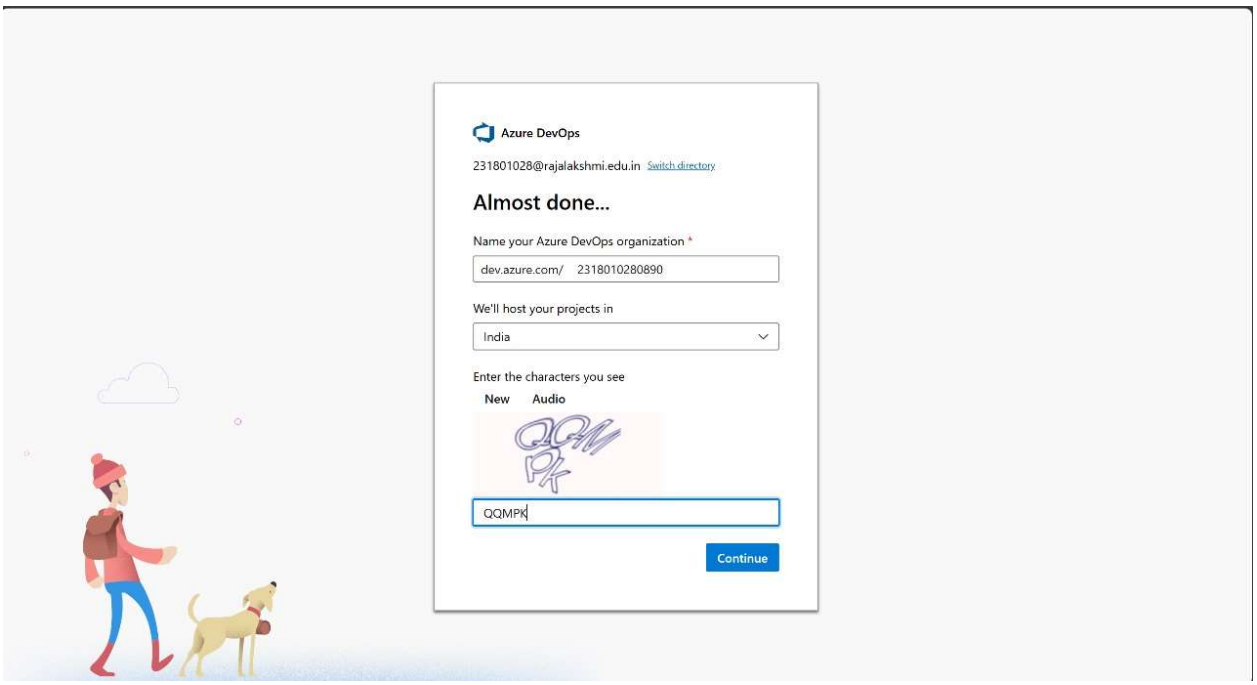
Successfully accessed the Azure DevOps environment and created a new organization through the Azure portal.

EXP NO: 2	AZURE DEVOPS PROJECT SETUP AND USER STORY MANAGEMENT

Aim:

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

1.Create An Azure Account



2. Create the First Project in Your Organization

a. After the organization is set up, you'll need to create your first **project**. This is where you'll begin to manage code, pipelines, work items, and more.

b. On the organization's **Home page**, click on the **New Project** button.

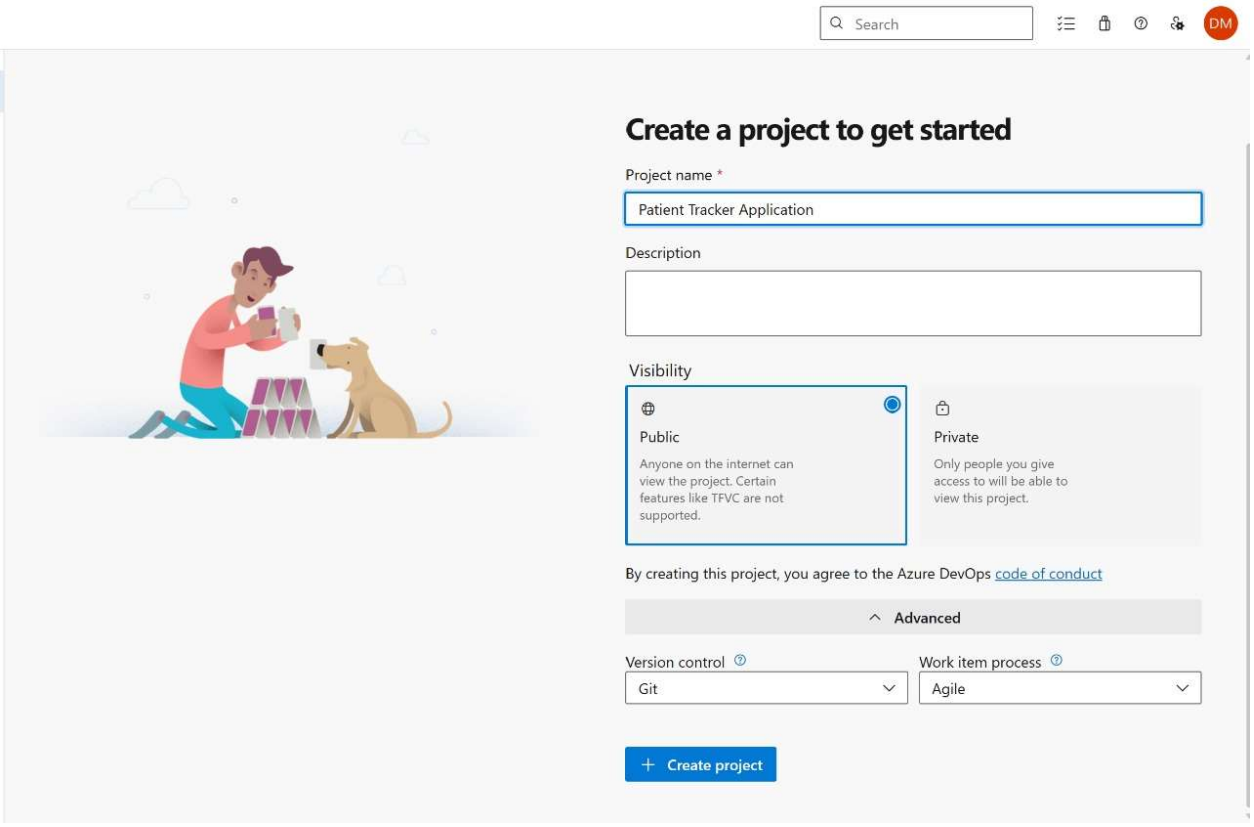
c. nter the project name, description, and visibility options:

***Name**:* Choose a name for the project (e.g., LMS).

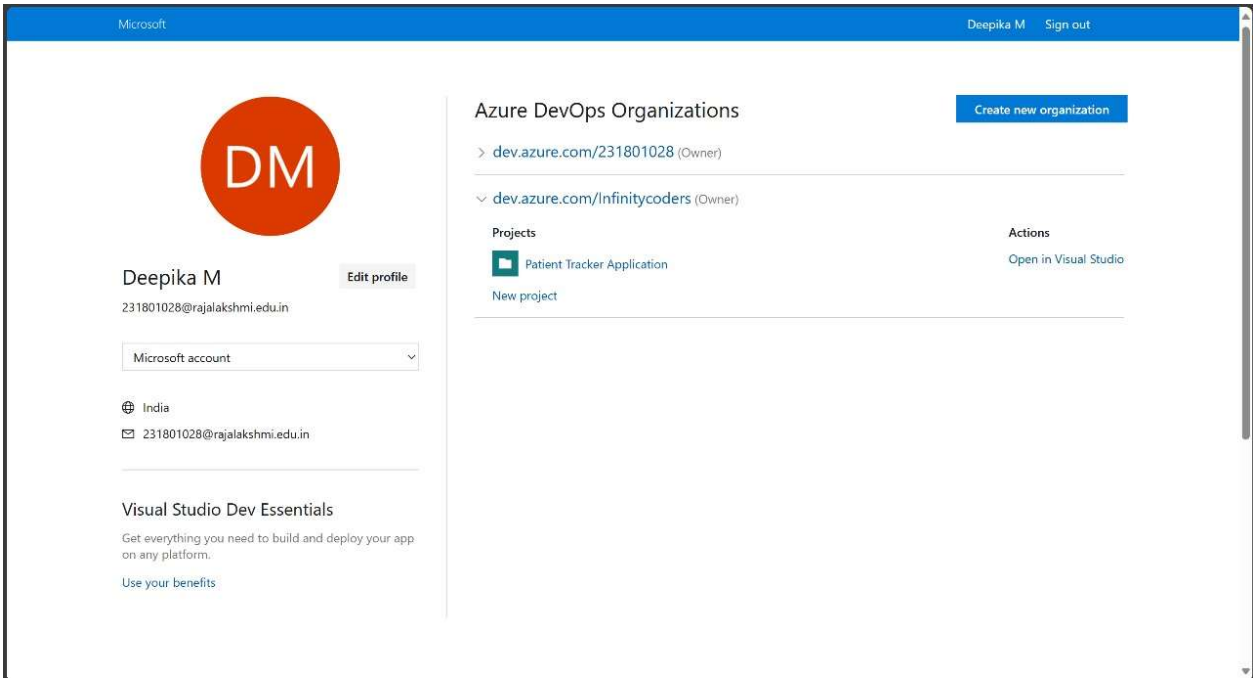
***Description**:* Optionally, add a description to provide more context about the project.

***Visibility**:* Choose whether you want the project to be **Private** (accessible only to those invited) or **Public** (accessible to anyone).

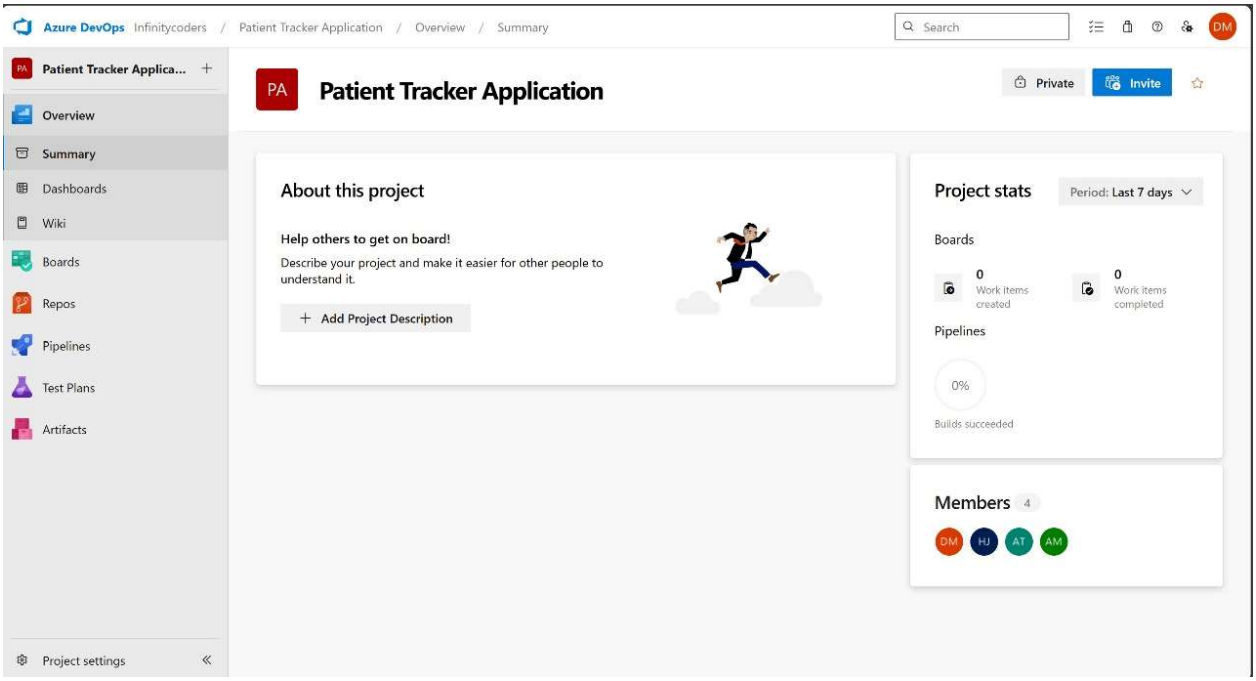
d. Once you've filled out the details, click **Create** to set up your first project.



3. Once logged in, ensure you are in the correct organization. If you're part of multiple organizations, you can switch between them from the top left corner (next to your user profile). Click on the Organization name, and you should be taken to the Azure DevOps Organization Home page.



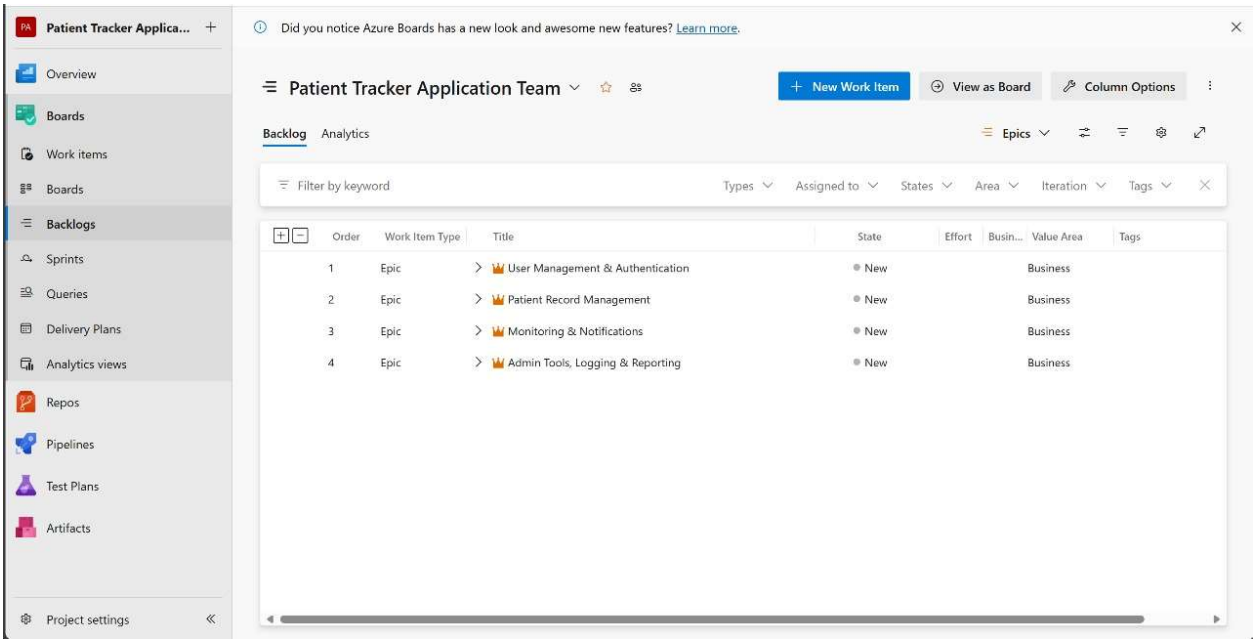
4. Project dashboard



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.



Result:

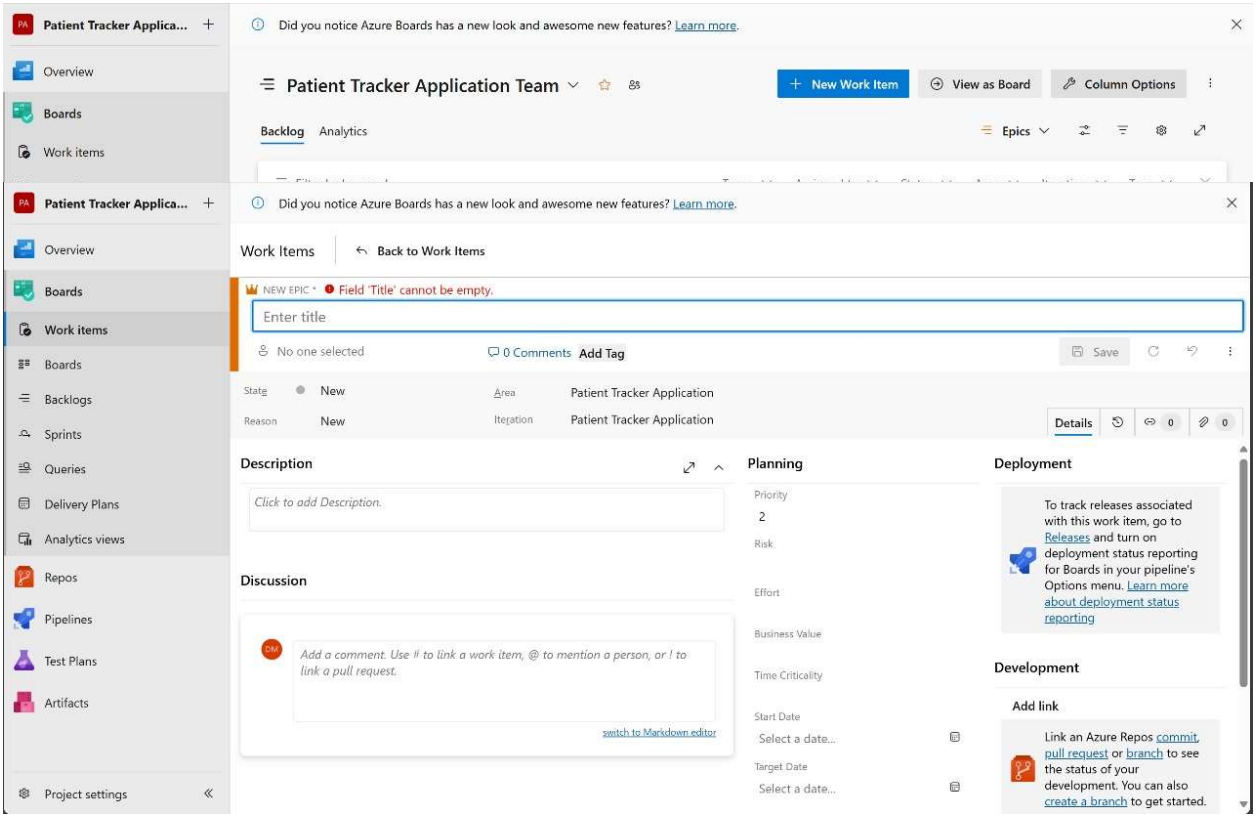
Successfully created an Azure DevOps project with user story management and agile workflow setup.

EXP NO: 3	SETTING UP EPICS, FEATURES, AND USER STORIES FOR PROJECT PLANNING

Aim:

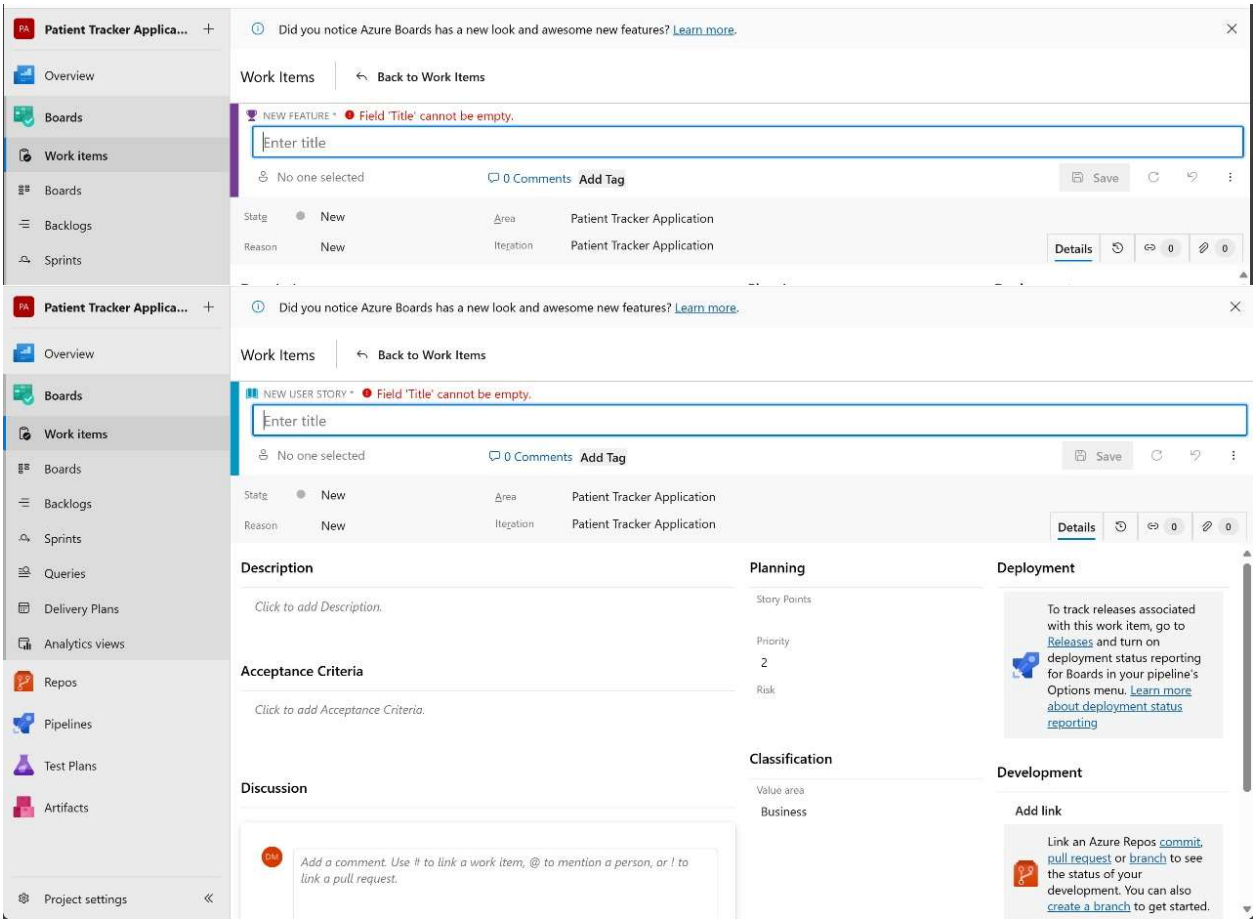
To learn about how to create epics, user story, features, backlogs for your assigned project.

Create Epic, Features, User Stories, Task



1. Fill in Epics

2. Fill in Features



3. Fill in User Story Details

Result:

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

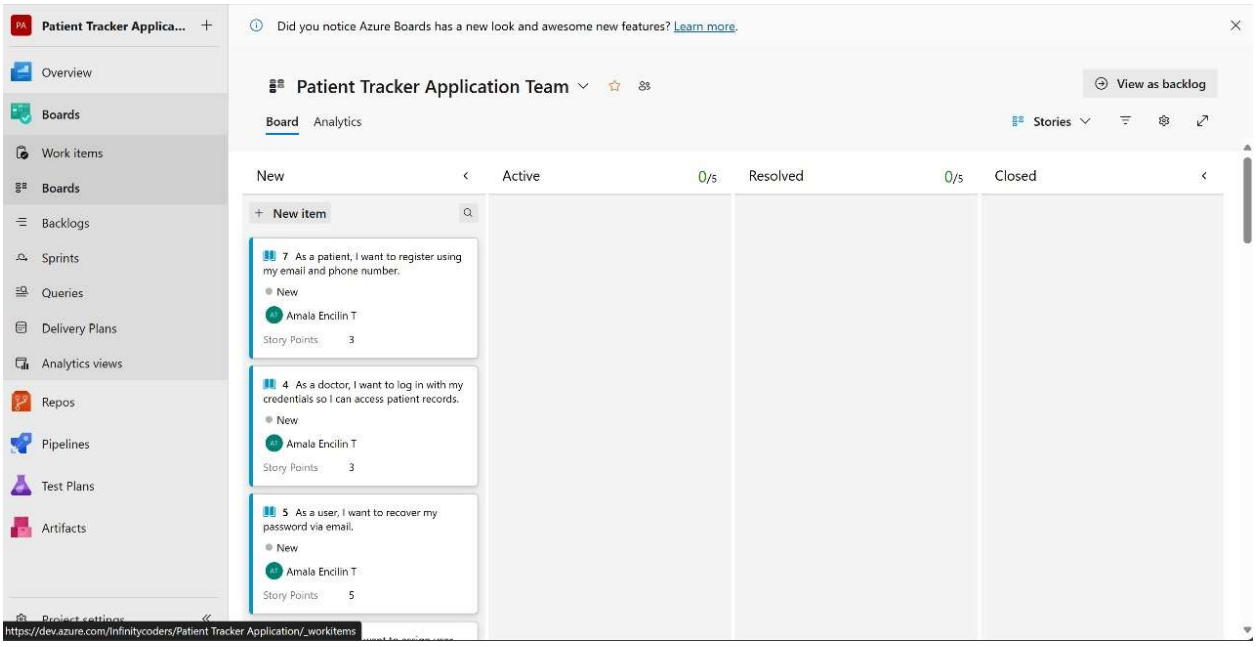
EXP NO: 4	SPRINT PLANNING

Aim:

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

Sprint Planning

Sprint



Result:

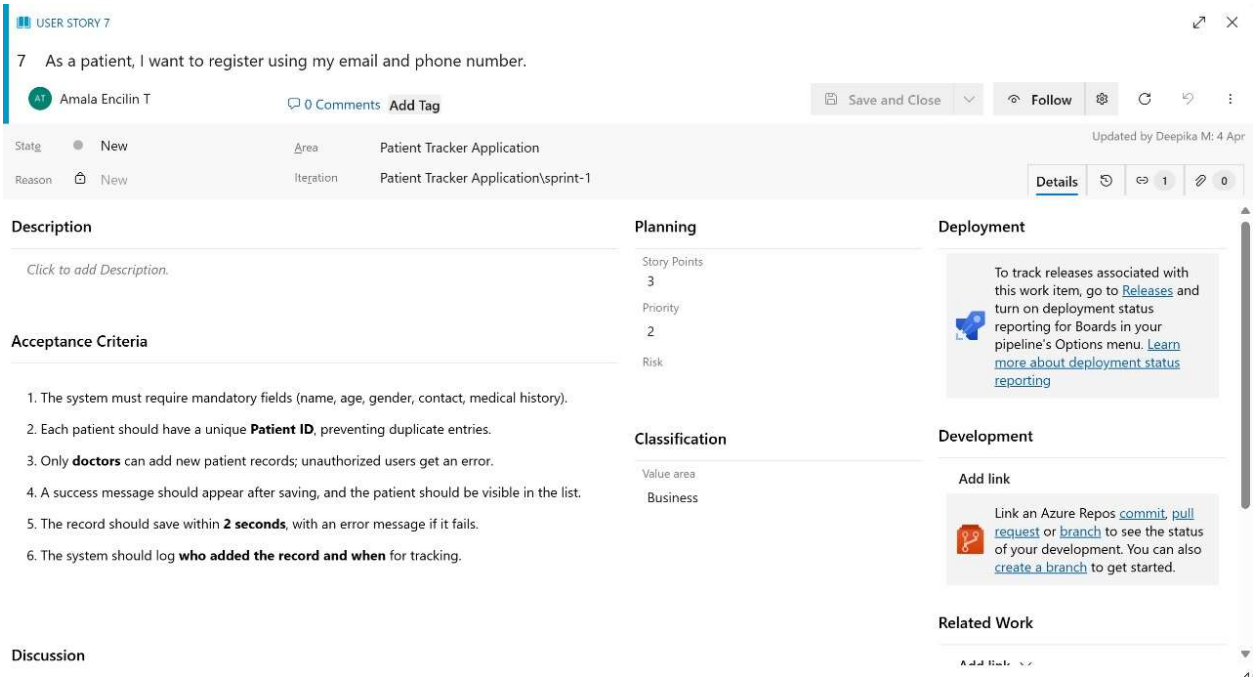
The Sprints are created for the patient tracker application.

EXP NO: 5	POKER ESTIMATION

Aim:

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

Poker Estimation



Result:

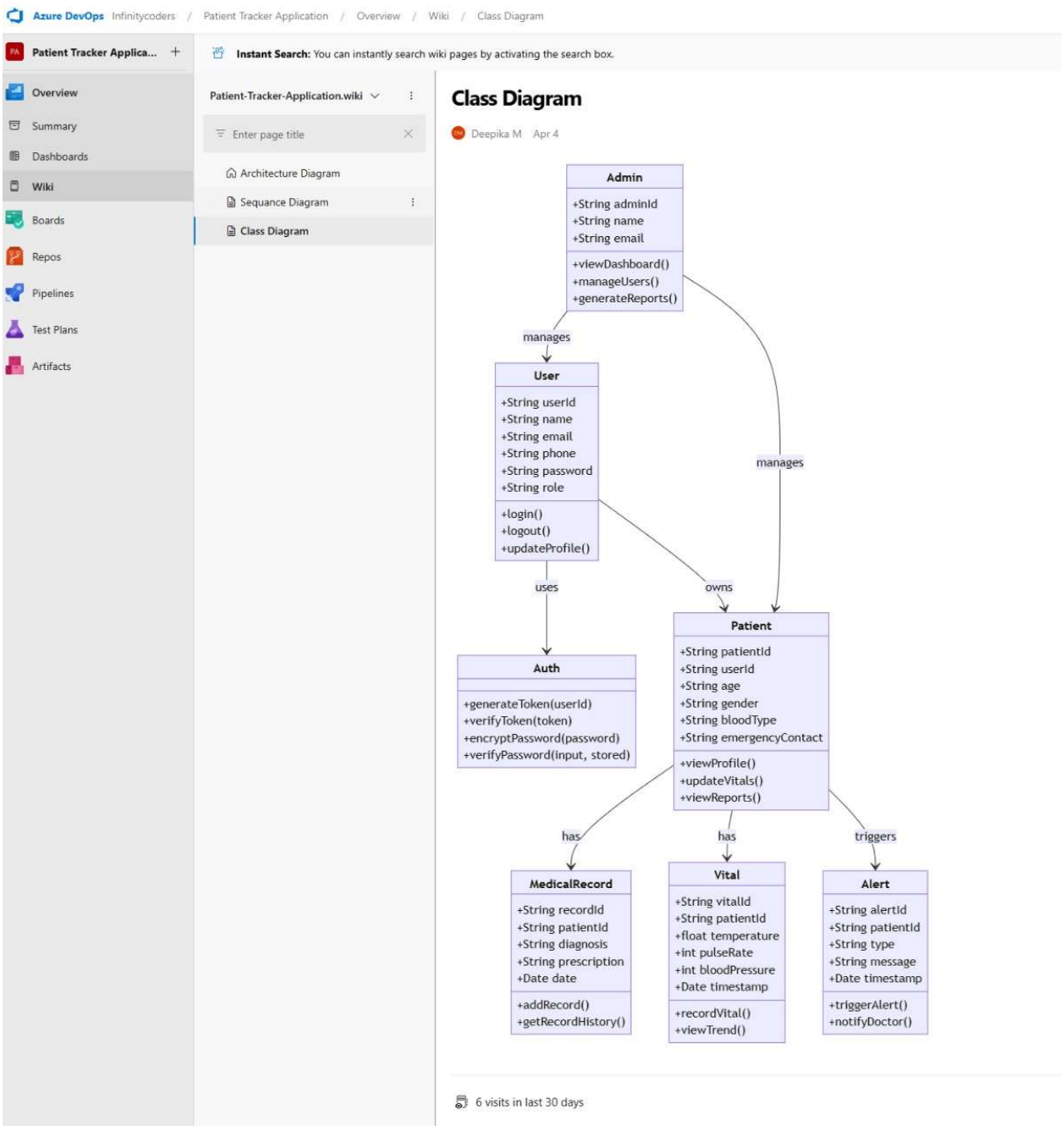
The Estimation/ Story Points is created for the project using Poker Estimation.

EXP NO: 6	DESIGNING CLASS AND SEQUENCE DIAGRAMS FOR PROJECT ARCHITECTURE

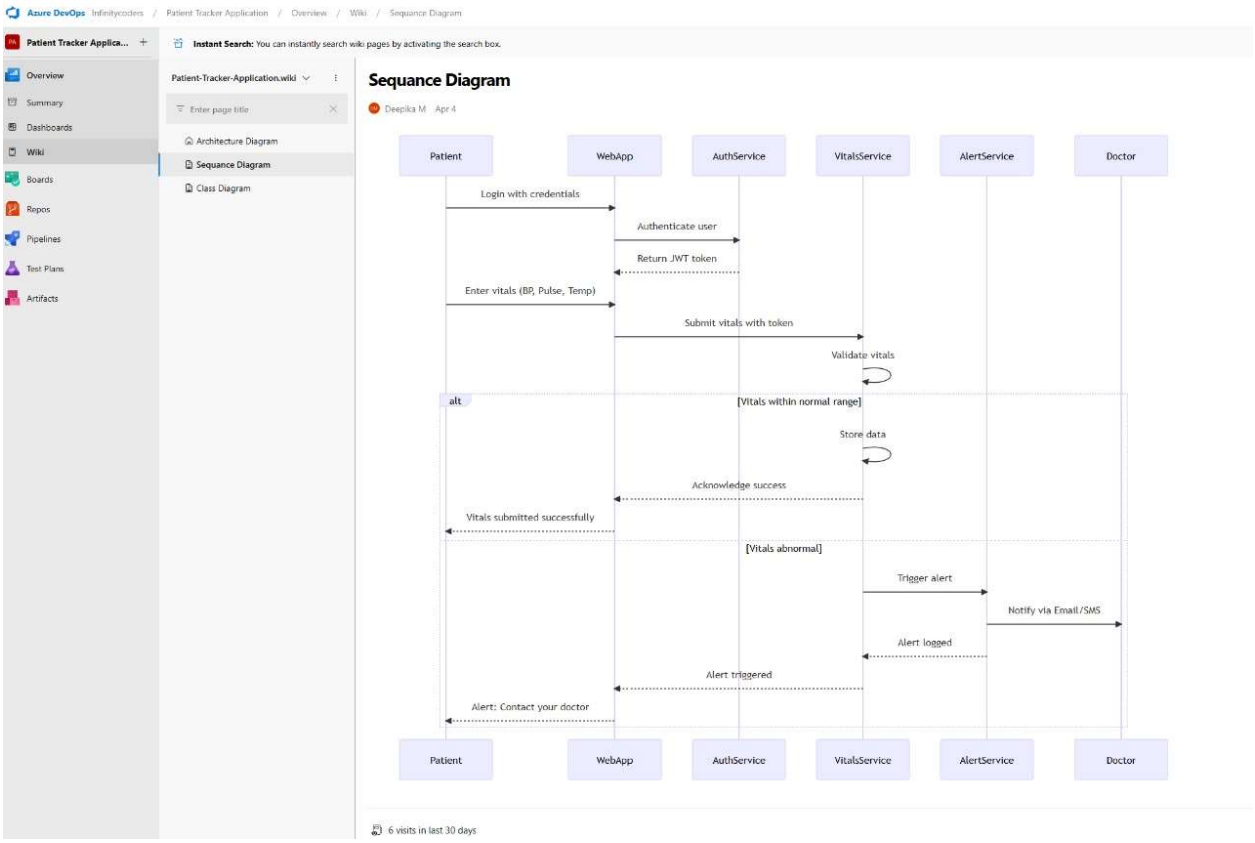
Aim:

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

6A. Class Diagram



6B. Sequence Diagram


Azure DevOps
Patient Tracker Application
Overview
Wiki
Sequance Diagram
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
Architecture Diagram
Sequance Diagram
Class Diagram
Sequance Diagram
Deepika M Apr 4
Patient
WebApp
AuthService
VitalsService
AlertService
Doctor
Login with credentials
Authenticate user
Return JWT token
Enter vitals (BP, Pulse, Temp)
Submit vitals with token
Validate vitals
alt
[Vitals within normal range]
Store data
Acknowledge success
Vitals submitted successfully
[Vitals abnormal]
Trigger alert
Notify via Email/SMS
Alert logged
Alert triggered
Alert: Contact your doctor
Patient
WebApp
AuthService
VitalsService
AlertService
Doctor
6 visits in last 30 days

Result:

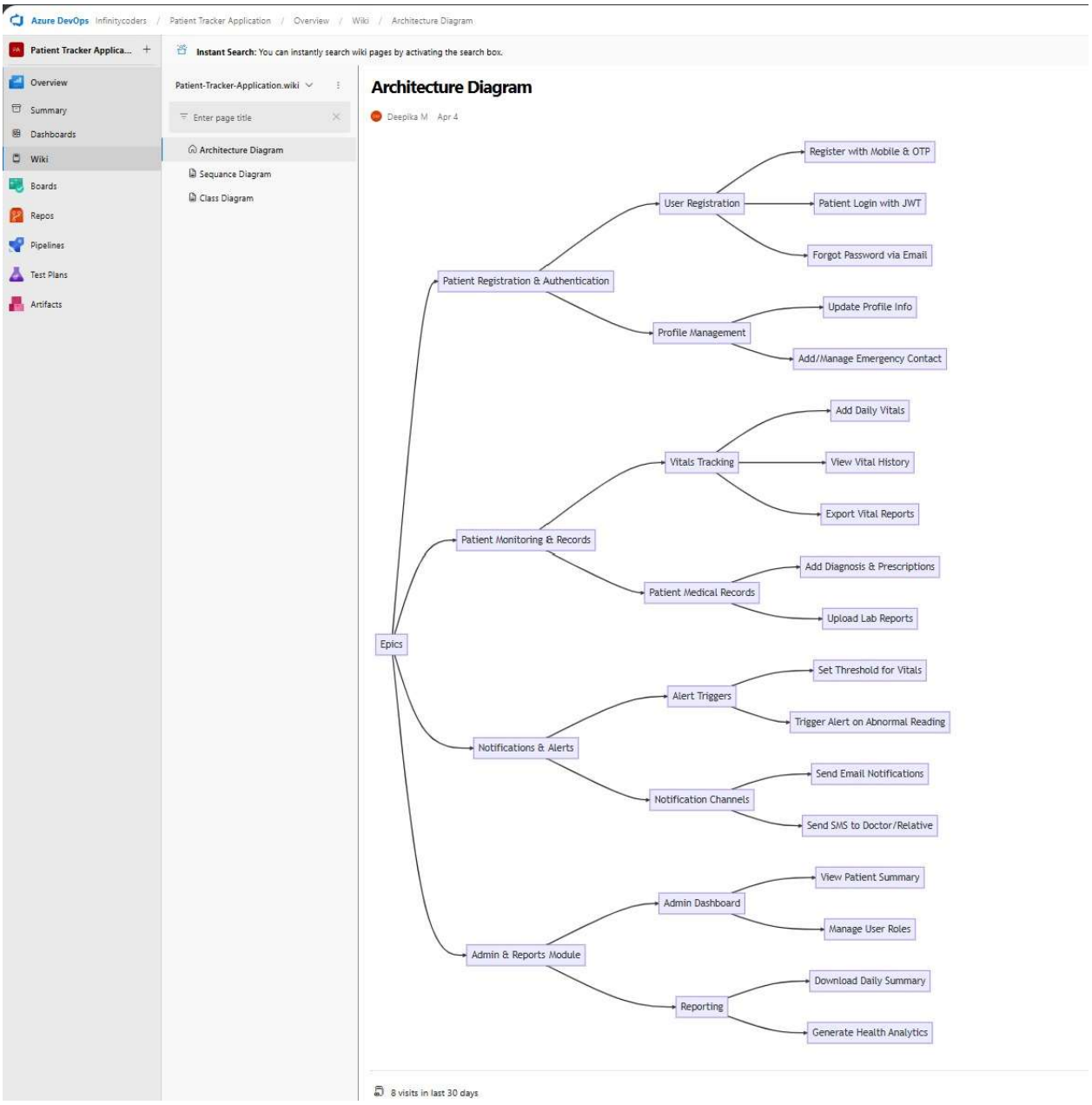
The Class Diagram and Sequence Diagram is designed Successfully for the patient tracker application.

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

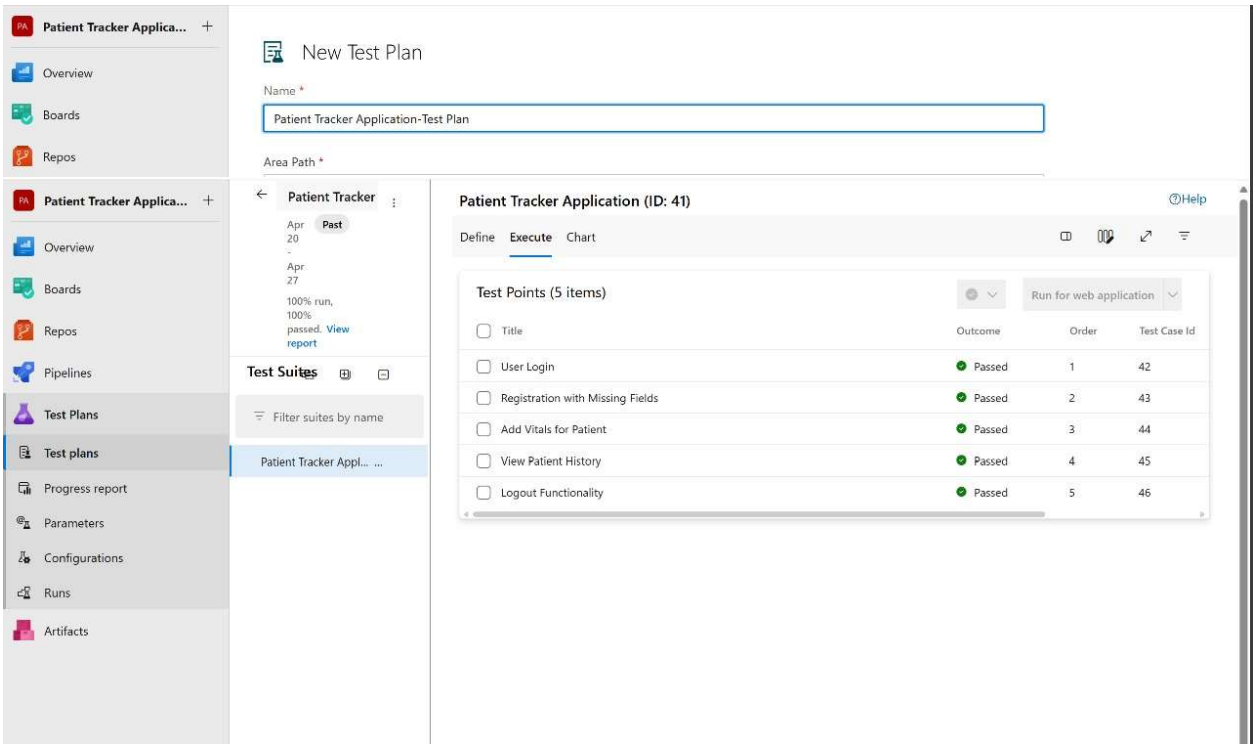
1. **Understand Core Features of the Application**
    - User Signup & Login
    - Viewing and Managing Playlists
    - Fetching Real-time Metadata
    - Editing playlists (rename, reorder, record)
    - Creating smart audio playlists based on categories (mood, genre, artist, etc.)
2. **Define User Interactions**
    - 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



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

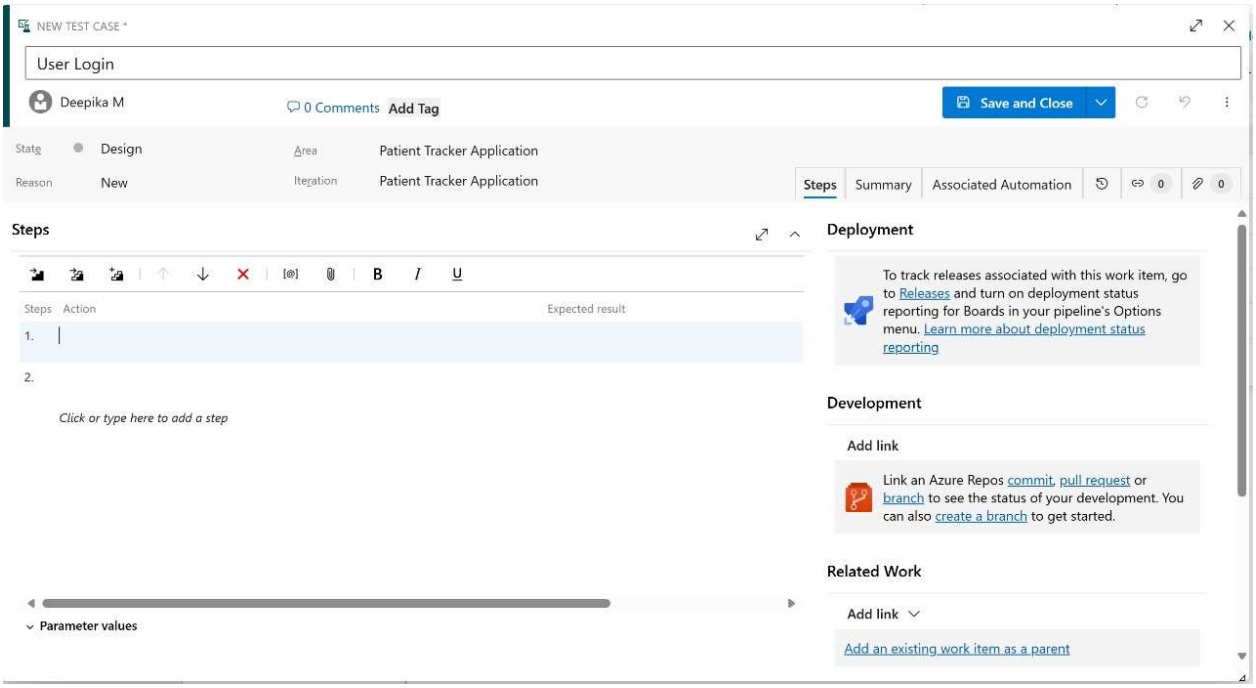
Test Suit: TS04 - Playlist Editing (ID: 89)

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

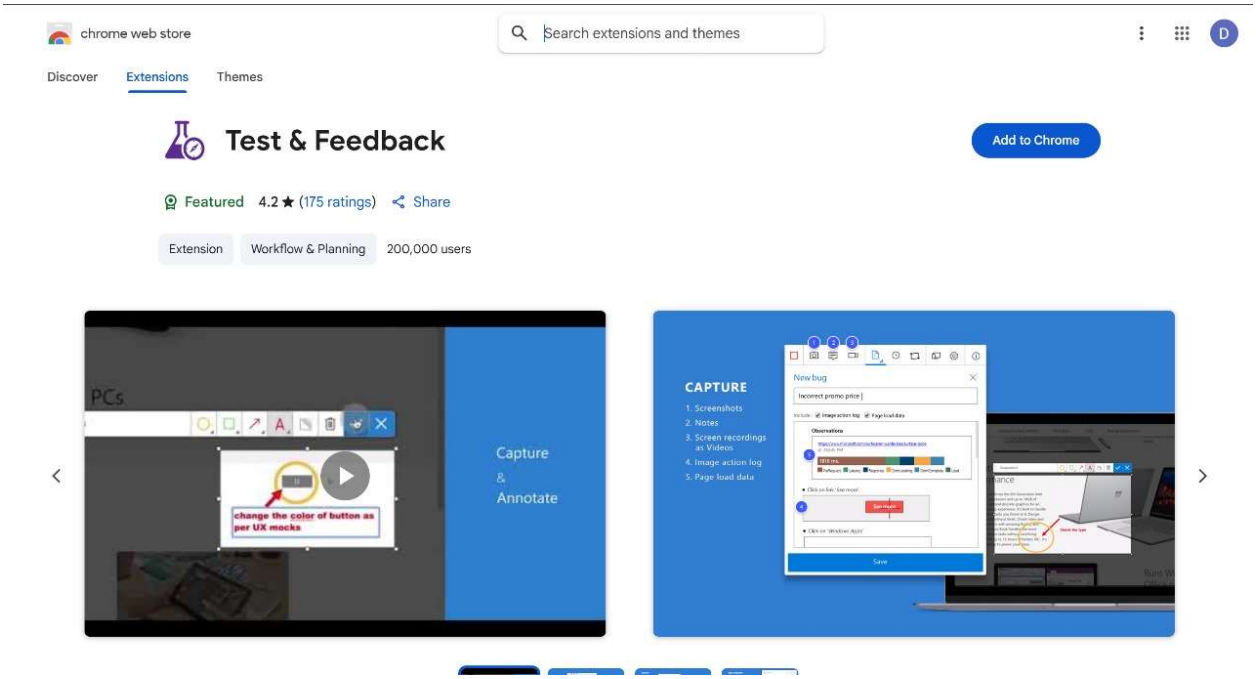
Test Suit: TS05 - Smart Playlist Creation (ID: 90)

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

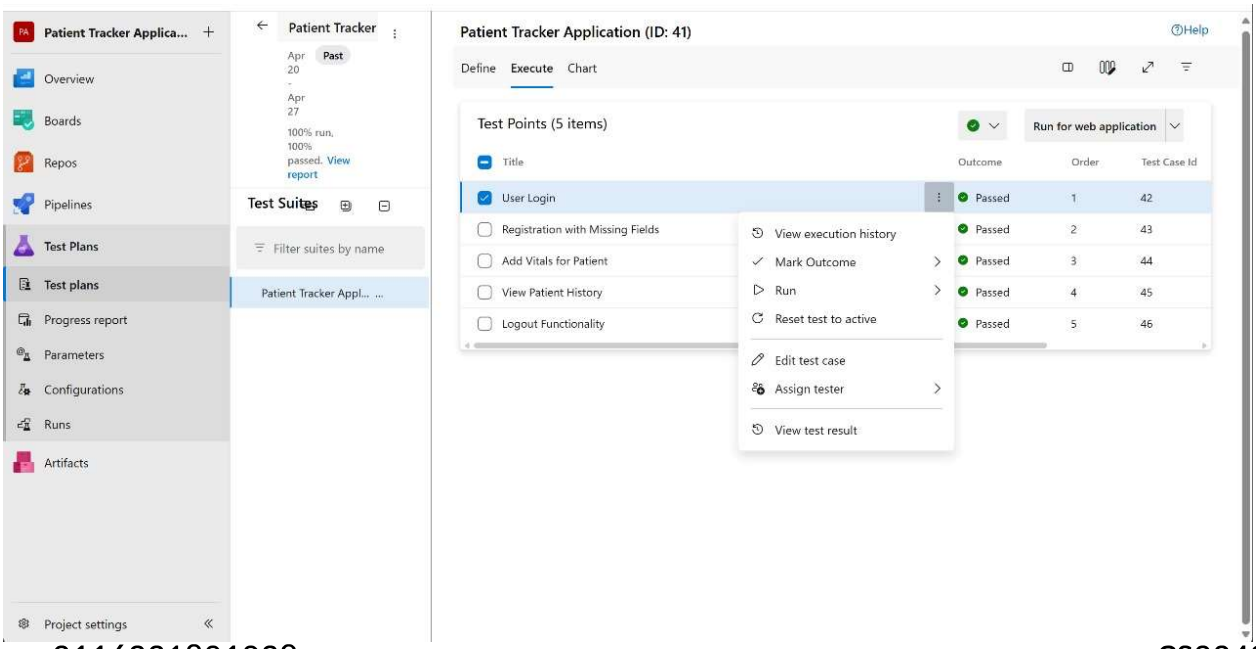


4. Installation of test

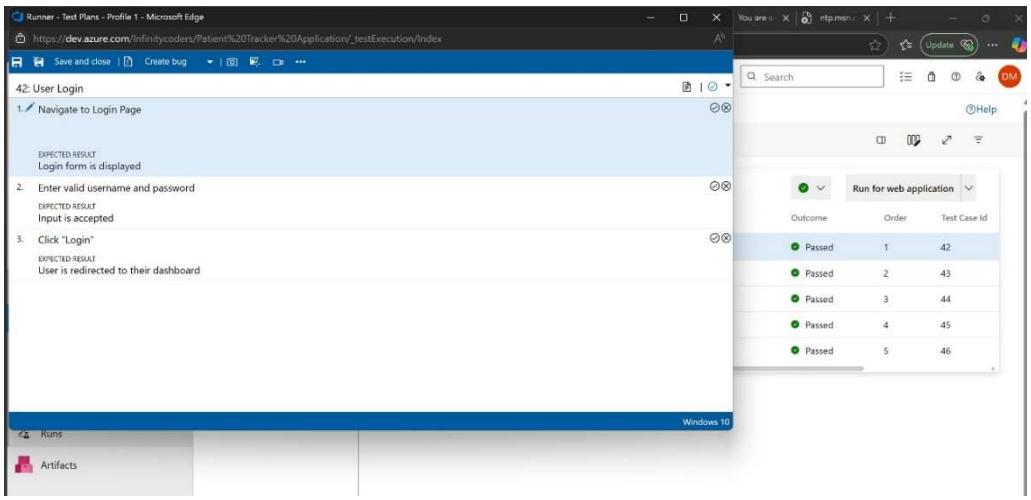


Test and feedback
Showing it as an extension

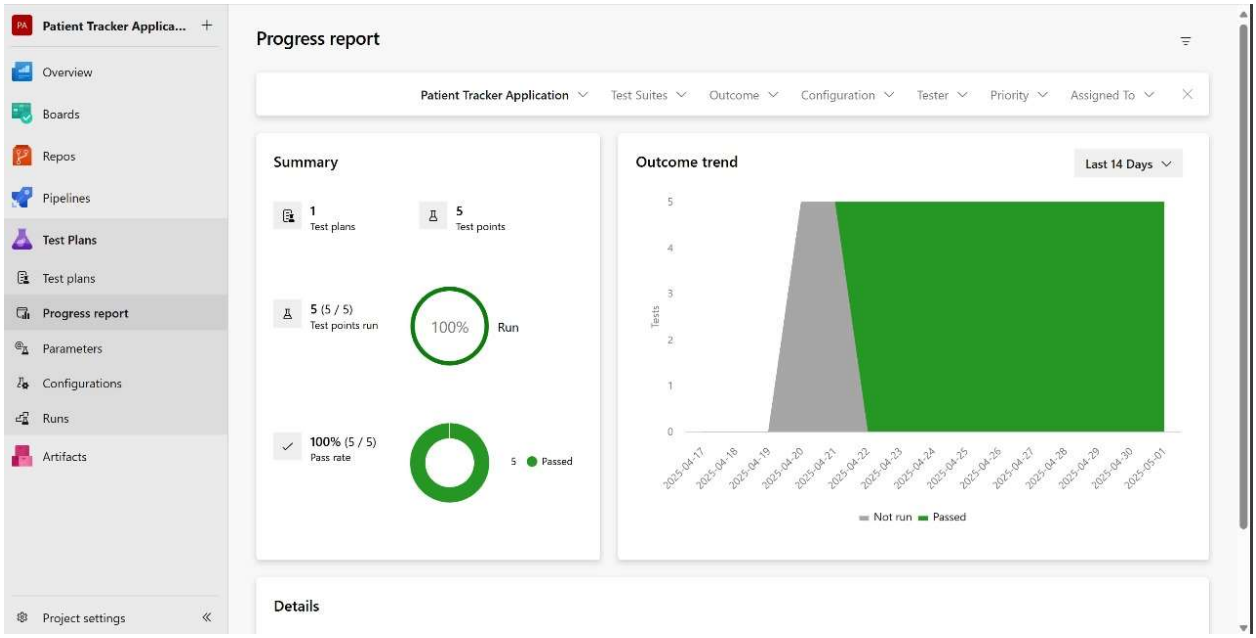
5. Running the test cases



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:

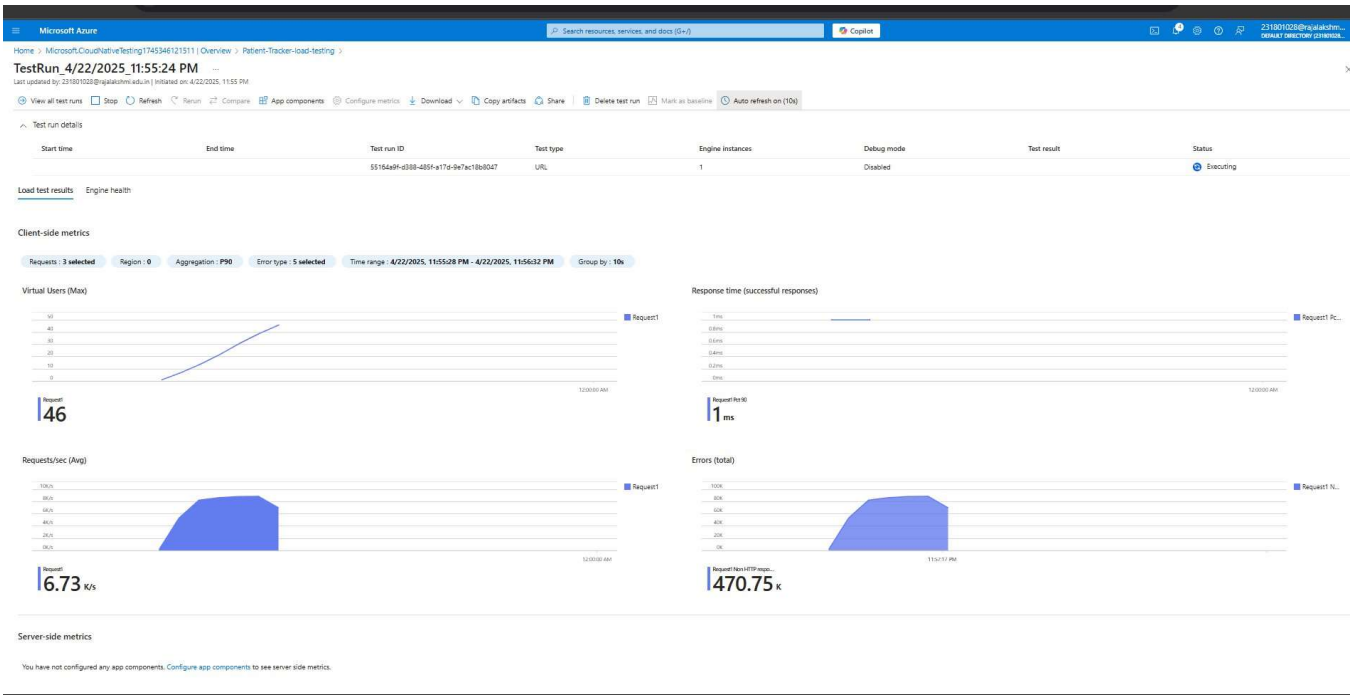
1. Sign in to Azure Portal
   Go to https://portal.azure.com and log in.
2. Create the Resource
   - Go to *Create a resource* → Search for "Azure Load Testing".
   - Select Azure Load Testing and click Create.
3. Fill in the Configuration Details
   - Subscription*:* Choose your Azure subscription.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

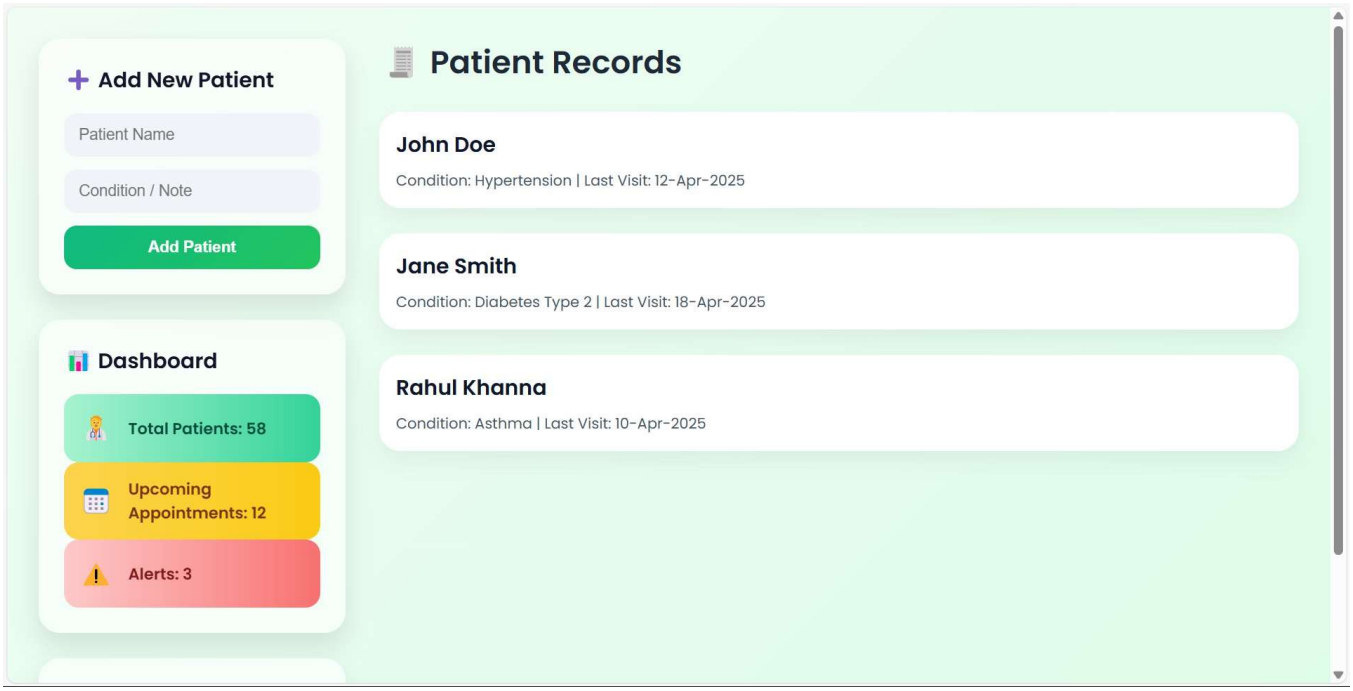
Steps to Create and Run a Load Test:

Once your resource is ready:

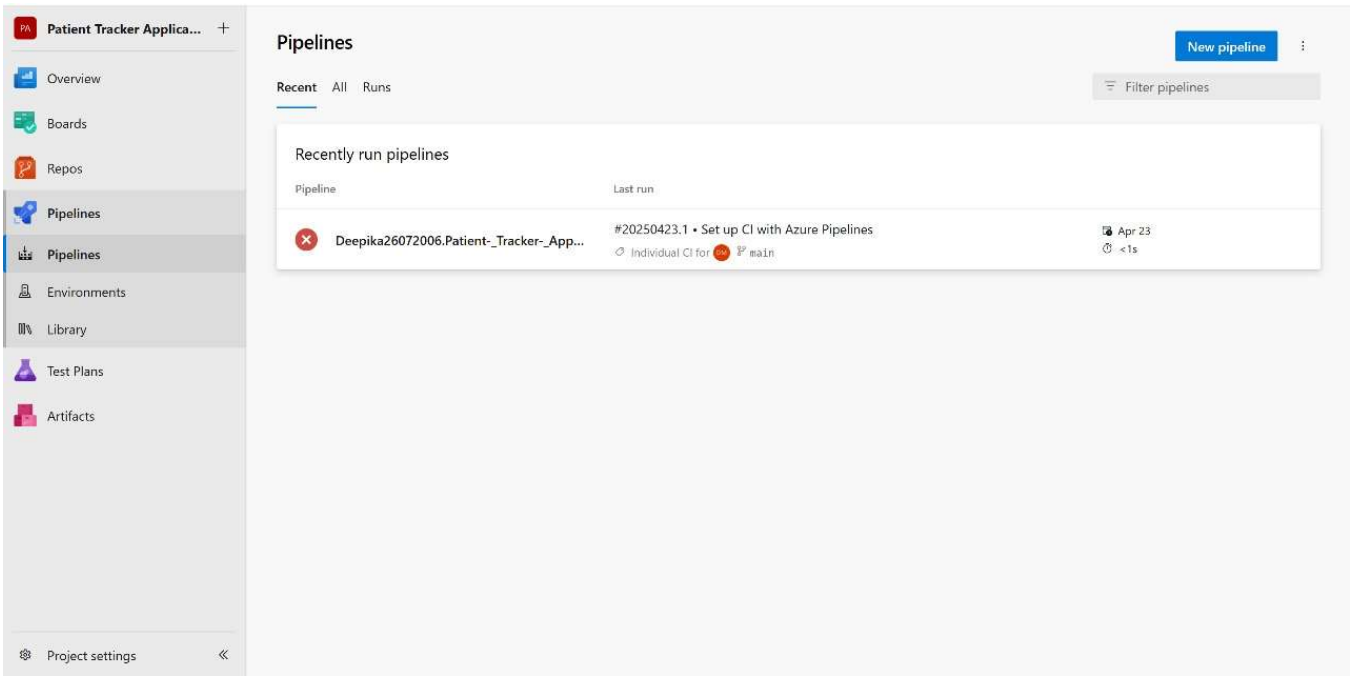
1. Go to your Azure Load Testing resource and click Add HTTP requests > Create.
2. Basics Tab
   - *Test Name:* Provide a unique name.
   - *Description:* (Optional) Add test purpose.
   - *Run After Creation:* Keep checked.
3. Load Settings
   - *Test URL:* Enter the target endpoint (e.g., https://yourapi.com/products).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


Patient Tracker Applica...
Overview
Boards
Repos
Pipelines
Pipelines
Environments
Library
Test Plans
Artifacts
Project settings
Pipelines
New pipeline
Recent
All
Runs
Filter pipelines
Recently run pipelines
Pipeline
Last run
Deepika26072006.Patient-_Tracker-_App...
#20250423.1 • Set up CI with Azure Pipelines
Individual CI for
main
Apr 23
<1s

Result:

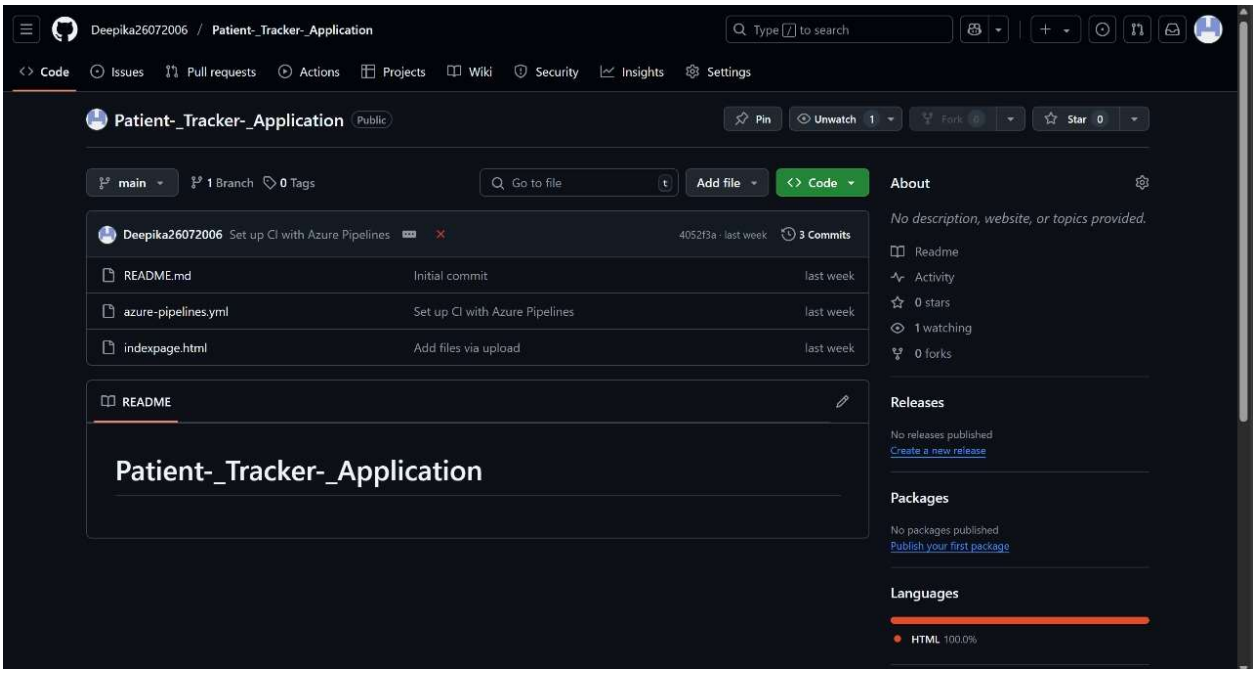
Successfully created the Azure Load Testing resource and executed a load test to assess the performance of the specified endpoint.

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.