



**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: Dilli Ganesh

Register No: 231801033

Year / Branch / Section: 2<sup>nd</sup> / AI&DS / FA

Semester: IV

Academic Year: 2024 - 2025

# INDEX

S.No.	Date	Title
1.	21/1/25	Azure Devops Environment Setup.
2.	21/1/25	Azure Devops Project Setup and User Story Management.
3.	28/1/25	Setting Up Epics, Features, And User Stories for Project Planning.
4.	11/2/25	Sprint Planning.
5.	18/2/25	Poker Estimation.
6.	25/2/25	Designing Class and Sequence Diagrams for Project Architecture.
7.	04/3/25	Designing Architectural and ER Diagrams for Project Structure.
8.	25/3/25	Testing – Test Plans and Test Cases.
9.	15/4/25	Load Testing and Pipelines.
10.	22/4/25	GitHub: Project Structure & Naming Conventions.

**EXP NO: 1**

## **AZURE DEVOPS ENVIRONMENT SETUP**

### **Aim:**

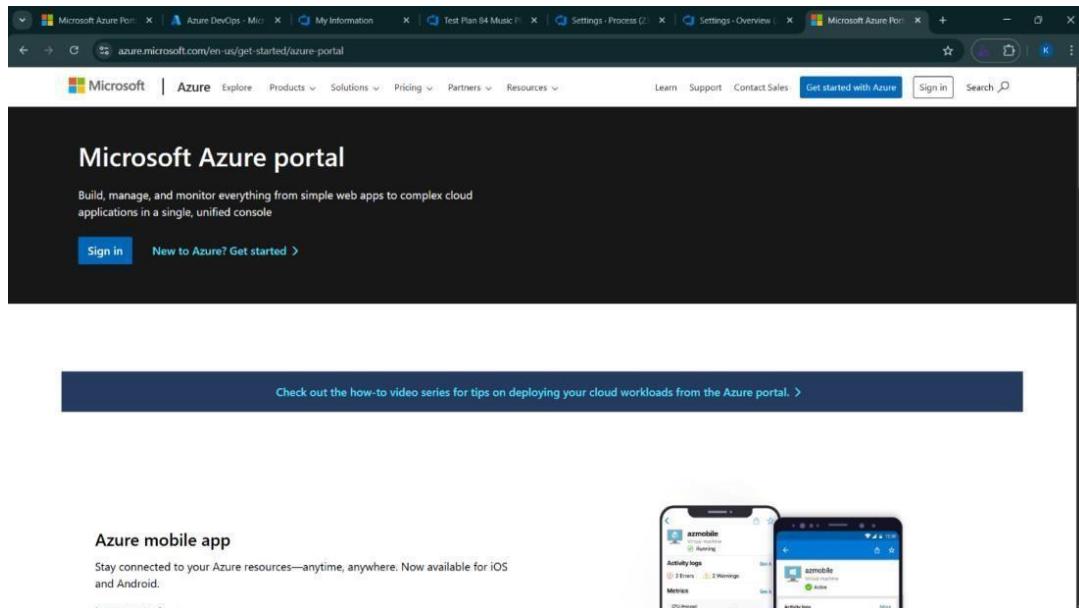
To set up and access the Azure DevOps environment by creating an organization through the Azure portal.

### **INSTALLATION**

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

Sign in using your Microsoft account credentials.

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



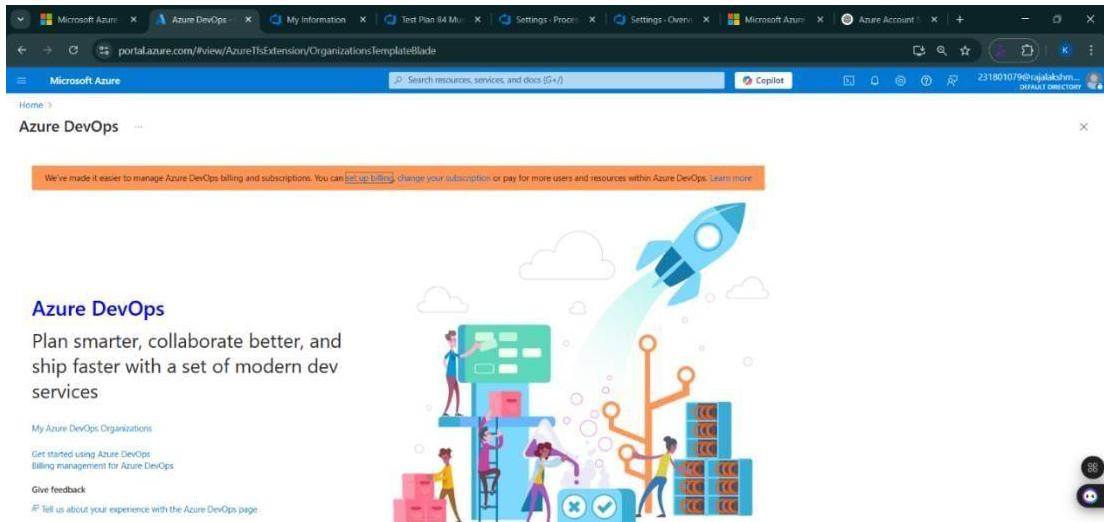
## 2. Azure home page

The screenshot shows the Microsoft Azure home page. At the top, there's a search bar and a Copilot button. Below the header, the "Azure services" section features a "Create a resource" button and icons for Azure DevOps organizations, Subscriptions, Dashboard hub, Resource groups, Azure Load Testing, Quickstart Center, Azure AI services, Kubernetes services, and More services. The "Resources" section displays recent resources: "Music" (Azure Load Testing, last viewed 1 day ago) and "Music\_playlist\_Batch\_Creator" (Resource group, last viewed 3 days ago). The "Navigate" section includes links for Subscriptions, Resource groups, All resources, and Dashboard. The "Tools" section lists Microsoft Learn, Azure Monitor, Microsoft Defender for Cloud, and Cost Management. The "Useful links" section includes Azure mobile app and other Azure services like Static Web App, Build Agents for Azure DevOps, InfluxDB Cloud, and Managed DevOps Pools.

3. Open DevOps environment in the Azure platform by typing **Azure DevOps Organizations** in the search bar.

The screenshot shows the Microsoft Azure home page with the search bar set to "DevOps". The search results are displayed under the "Services" section, showing items like Azure Native New Relic Service, Managed DevOps Pools, Azure DevOps organizations, and Azure Native Dynatrace Service. The "Marketplace" section shows Static Web App, Build Agents for Azure DevOps, and InfluxDB Cloud. The "Documentation" section includes links to DevOps architecture design, Secure your Azure DevOps, Course A2-400T00-A, and Managed DevOps Pools Overview. The "Tools" section shows Microsoft Entra ID, Azure DevOps, and Continue searching in Microsoft Entra ID. The "Useful links" section includes Azure mobile app and other Azure services.

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.

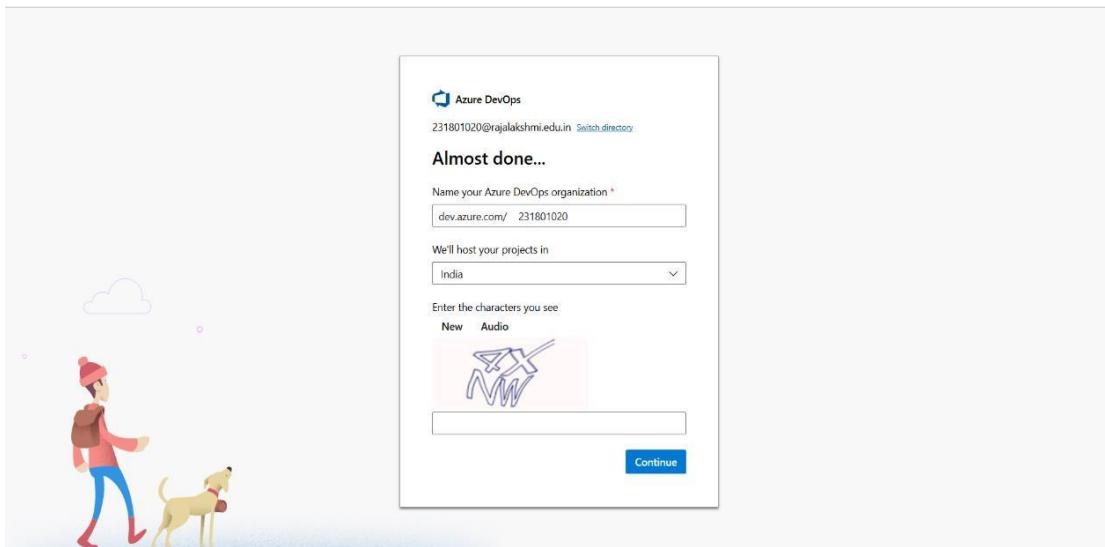
**EXP NO: 2**

## **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**

- 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., **HMS**).
  - 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.

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.

## Create new project

X

Project name \*

Hospital Management System (HMS)

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.

By creating this project, you agree to the Azure DevOps [code of conduct](#)

Advanced

Version control

Git

Work item process

Team-Genesis Agile

Cancel

Create

The screenshot shows the Azure DevOps Organizations dashboard. At the top, there's a profile picture of Barath Kumar S J with the initials 'BJ'. Below it, his name and email are listed: Barath Kumar S J, 231801020@rajalakshmi.edu.in. A Microsoft account dropdown shows 'India' and the same email address. To the right, under 'Azure DevOps Organizations', it says 'dev.azure.com/Team-Genesis (Owner)'. There's a 'Create new organization' button. Under 'Projects', there's one listed: 'Hospital Management System'. An 'Actions' section has a 'New project' link. On the left sidebar, there are sections for 'Visual Studio Dev Essentials' and 'Use your benefits'.

#### 4. Project dashboard

The screenshot shows the Azure DevOps Project Overview for 'Hospital Management System'. The left sidebar has a navigation menu with options like Overview, Summary, Dashboards, Wiki, Boards, Repos, Pipelines, Test Plans, and Artifacts. The main area displays the project title 'Hospital Management System' and an 'About this project' section. It describes the Azure-Powered Hospital Management System (HMS) as a smart, cloud-enabled solution for hospital operations, consultations, and pharmacy workflows. Key features listed include Appointment Scheduling, Electronic Medical Records, Prescription Management, Billing & Payment, User Role Management, Notifications & Reminders, and Analytics Dashboard. The 'Project stats' section shows 16 work items created and 0 work items completed over the last 7 days. The 'Members' section lists several team members represented by icons.

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.

Order	Work Item Type	Title	State	Effort	Business Value Area	Tags
1	Epic	> Admin Management	New		Business	
2	Epic	> Appointment Management	New		Business	
3	Epic	> Patient Consultation	New		Business	
4	Epic	> Pharmacy Management	New		Business	
5	Epic	> Billing & Payments	New		Business	
6	Epic	> Reports & Analytics	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 epics, user story, features, backlogs for your assigned project.

### **Create Epic, Features, User Stories, Task**

The screenshot shows the Azure DevOps Boards Backlog view for the 'Hospital Management System' team. The backlog is organized into several Epics:

- Epic 1: Admin Management
  - Feature: Hospital Data Management
    - User Story: As an admin, I want to manage hospital data so that I can... (with details: Create UI for admins to set data)
    - Feature: User and Role Management
      - User Story: As an admin, I want to manage users and their roles so that I can... (with details: Create UI for admins to set roles)
  - Epic 2: Appointment Management
    - Feature: Doctor Availability Management
      - User Story: As a doctor, I want to update my available time slots so that I can... (with details: Create UI for doctors to set availability, Store available slots in the database, Allow doctors to update or delete available slots)
      - Task: Create UI for doctors to set availability
      - Task: Store available slots in the database.
      - Task: Allow doctors to update or delete available slots.
      - User Story: As a patient, I want to book an available slot with a doctor so that I can... (with details: Create UI for patients to book slots)
  - Epic 3: Patient Consultation
  - Epic 4: Pharmacy Management

The backlog table includes columns for Order, Work Item Type, Title, State, Effort, Business Value Area, and Tags.

### **1. Fill in Epics**

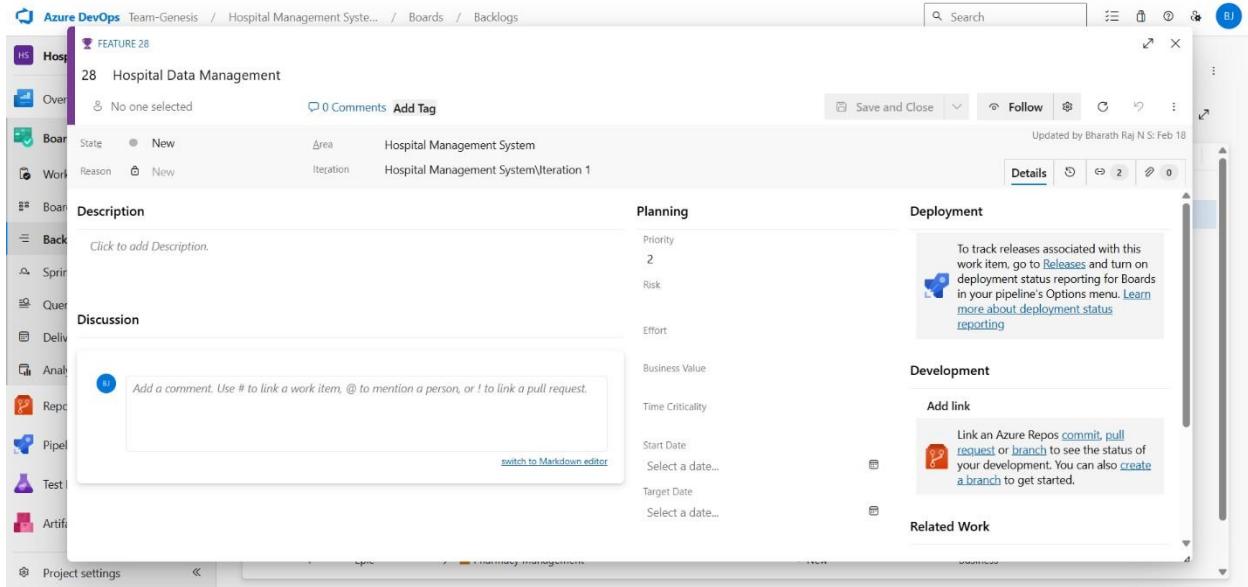
The screenshot shows the details of an epic work item titled 'EPIC.26 Admin Management'. The work item has the following details:

- State:** New
- Area:** Hospital Management System
- Iteration:** Hospital Management System\Iteration 1

The work item is divided into three main sections: Description, Planning, and Deployment.

- Description:** Click to add Description.
- Planning:** Priority: 2, Risk: , Effort: , Business Value: , Time Criticality: .
- 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:** Related work items and tasks.

## 2. Fill in Features



The screenshot shows the Azure DevOps Work Item Editor for a Feature. The work item ID is 28, titled "Hospital Data Management". The "Planning" section shows a Priority of 2. The "Deployment" section includes a note about tracking releases and a link to learn more about deployment status reporting. The "Development" section has a "Add link" button and a note about linking to Azure Repos. The "Related Work" section is currently empty.

FEATURE 28

28 Hospital Data Management

No one selected 0 Comments Add Tag

State: New Area: Hospital Management System

Reason: New Iteration: Hospital Management System\Iteration 1

Updated by Bharath Raj N S: Feb 18

Description

Click to add Description.

Planning

Priority: 2

Risk:

Effort:

Business Value:

Time Criticality:

Start Date: Select a date...

Target Date: Select a date...

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](#)

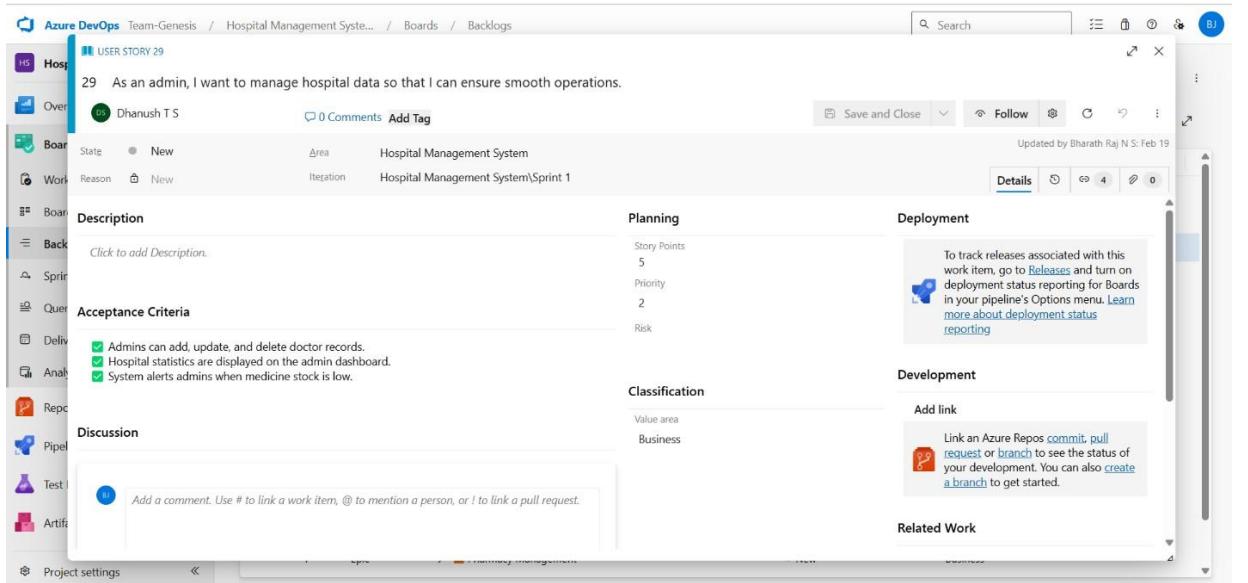
Development

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

## 3. Fill in User Story Details



The screenshot shows the Azure DevOps Work Item Editor for a User Story. The work item ID is 29, titled "As an admin, I want to manage hospital data so that I can ensure smooth operations." The "Planning" section shows Story Points of 5 and a Priority of 2. The "Classification" section indicates the value area is Business. The "Development" section has a "Add link" button and a note about linking to Azure Repos. The "Related Work" section is currently empty.

USER STORY 29

29 As an admin, I want to manage hospital data so that I can ensure smooth operations.

Dhanush T S 0 Comments Add Tag

State: New Area: Hospital Management System

Reason: New Iteration: Hospital Management System\Sprint 1

Updated by Bharath Raj N S: Feb 19

Description

Click to add Description.

Planning

Story Points: 5

Priority: 2

Risk:

Classification

Value area: Business

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:**

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 HMS Project.

## Sprint Planning:

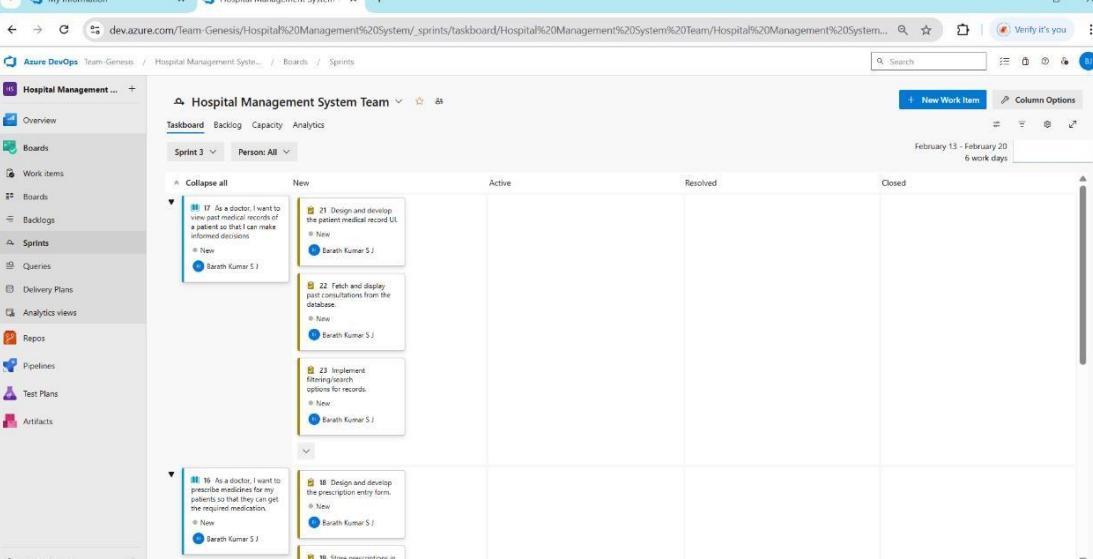
### Sprint 1

The screenshot shows the Azure DevOps Boards interface for the Hospital Management System Team. The left sidebar is expanded to show the 'Sprints' section under 'Backlogs'. The main area displays the 'Sprint 1' backlog. The backlog items are listed in a grid with columns for Type, Assigned to, Status, and Area. The first item is '32 Track inventory and update medicine stock with low stock alerts', assigned to 'Dhanush TS'. The second item is 'As an admin, I want to manage users and their roles so that I can control system access.', assigned to 'alvin'. The third item is '42 Develop a UI to update outgoing medicine stock based on prescriptions', assigned to 'alvin'. The fourth item is '43 Create functionality to alert admins when stock is low', assigned to 'alvin'. The fifth item is '44 Allow pharmacists to', assigned to 'alvin'. There are also three 'not started' items at the bottom of the list.

### Sprint 2

The screenshot shows the Azure DevOps Boards interface for the Hospital Management System Team. The left sidebar is expanded to show the 'Sprints' section under 'Backlogs'. The main area displays the 'Sprint 2' backlog. The backlog items are listed in a grid with columns for Type, Assigned to, Status, and Area. The first item is '7 As a doctor, I want to update my available time slots so that patients can book appointments.', assigned to 'Sharan Raj N S'. The second item is '8 Create UI for doctors to set availability', assigned to 'Sharan Raj N S'. The third item is '9 Store available slots in the database', assigned to 'Sharan Raj N S'. The fourth item is '10 Allow doctors to update or delete available slots.', assigned to 'Sharan Raj N S'. The fifth item is '11 As a patient, I want to book an available slot with a doctor so that I can consult them.', assigned to 'Sharan Raj N S'. The sixth item is '12 Create UI for patients to view available slots.', assigned to 'Sharan Raj N S'. The seventh item is '24 Validate booking conflicts and slot availability.', assigned to 'Sharan Raj N S'.

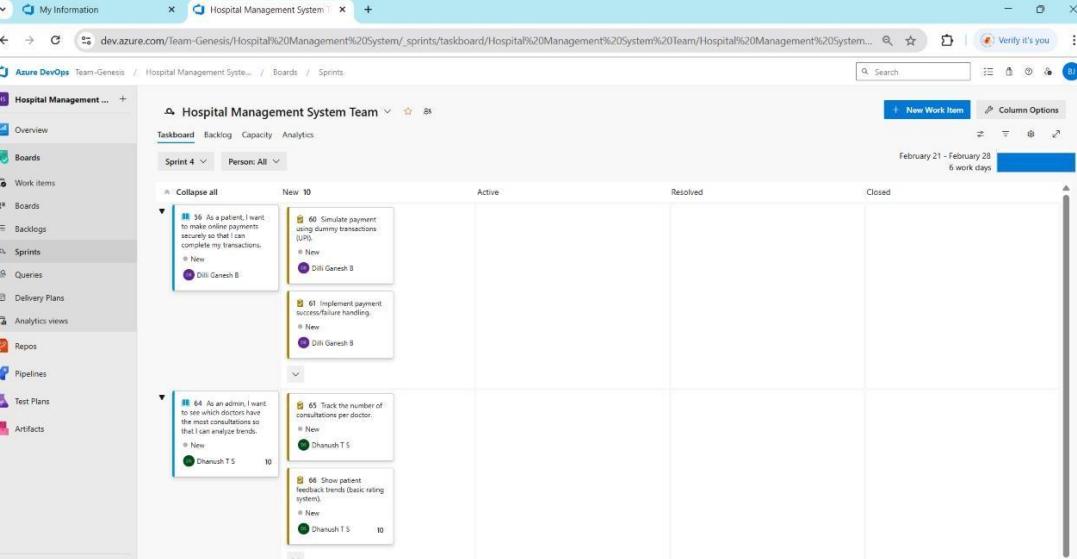
## Sprint 3



The screenshot shows the Azure DevOps Taskboard for Sprint 3. The board has four columns: New, Active, Resolved, and Closed. The 'New' column contains three items, each with a description and assigned to Barath Kumar S J. The 'Active' column contains one item, also assigned to Barath Kumar S J. The 'Resolved' and 'Closed' columns are currently empty.

New	Active	Resolved	Closed
21 Design and develop the patient medical record UI. As a doctor, I want to view all medical records of a patient so that I can make informed decisions New Barath Kumar S J	22 Fetch and display past consultations from the database. As a doctor, I want to fetch and display past consultations from the database. New Barath Kumar S J		
23 Implement filtering/search options for records. As a doctor, I want to prescribe medicines for my patients so that they can get required medication. New Barath Kumar S J			
24 As a patient, I want to make online payments securely so that I can complete my transactions. New Dilli Ganesh B	25 Design and develop the prescription entry form. As a patient, I want to design and develop the prescription entry form. New Barath Kumar S J		
	26 Store prescriptions in the database. As a patient, I want to store prescriptions in the database. New Barath Kumar S J		

## Sprint 4



The screenshot shows the Azure DevOps Taskboard for Sprint 4. The board has four columns: New, Active, Resolved, and Closed. The 'New' column contains two items, both assigned to Dilli Ganesh B. The 'Active' column contains one item, also assigned to Dilli Ganesh B. The 'Resolved' and 'Closed' columns are currently empty.

New	Active	Resolved	Closed
56 As a patient, I want to make online payments securely so that I can complete my transactions. New Dilli Ganesh B	57 Implement payment success/failure handling. As a patient, I want to implement payment success/failure handling. New Dilli Ganesh B		
58 As an admin, I want to see which doctors have the most consultations so that I can analyze trends. New Dhanush T S	59 Track the number of consultations per doctor. As an admin, I want to track the number of consultations per doctor. New Dhanush T S		
	60 Simulate payment using dummy transactions (UP). As a patient, I want to simulate payment using dummy transactions (UP). New Dilli Ganesh B		
	61 Show patient feedback trends (basic rating system). As a patient, I want to show patient feedback trends (basic rating system). New Dhanush T S		

**Result:**

The Sprints are created for the Hospital Management System Project.

**EXP NO: 5**

# **POKER ESTIMATION**

## **Aim:**

Create Poker Estimation for the user stories – Hospital Management System Project.

## **Poker Estimation**

The screenshot shows the Azure DevOps interface for the 'Hospital Management System' project. The left sidebar navigation includes 'Overview', 'Work items', 'Boards', 'Backlogs' (which is selected), 'Sprints', 'Queries', 'Delivery Plans', 'Analytics views', 'Repos', 'Pipelines', 'Test Plans', and 'Artifacts'. The main content area displays the 'Backlog' for the 'Hospital Management System Team'. The backlog table has columns for Order, Work Item Type, Title, Description, Iteration Path, Story Points, Value Area, and State. The backlog contains 11 User Stories, each with a brief description and assigned story points (e.g., 5, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3) across various iterations (Sprint 1, Sprint 2, Sprint 3, Sprint 4).

Order	Work Item Type	Title	Iteration Path	Story Points	Value Area	State
1	User Story	> As a patient, I want to make online payments securely so th...	Hospital Management System\Sprint 4	5	Business	New
2	User Story	> As an admin, I want to manage hospital data so that I can e...	Hospital Management System\Sprint 1	5	Business	New
3	User Story	> As an admin, I want to manage users and their roles so that ...	Hospital Management System\Sprint 1	5	Business	New
4	User Story	> As a pharmacist, I want to manage medicine stock so that p...	Hospital Management System\Sprint 1	3	Business	New
5	User Story	> As a doctor, I want to update my available time slots so that...	Hospital Management System\Sprint 2	3	Business	New
6	User Story	> As a patient, I want to book an available slot with a doctor s...	Hospital Management System\Sprint 2	5	Business	New
7	User Story	> As a patient, I want to view my consultation and medicine b...	Hospital Management System\Sprint 2	3	Business	New
8	User Story	> As a doctor, I want to view past medical records of a patient...	Hospital Management System\Sprint 3	3	Business	New
9	User Story	> As a doctor, I want to prescribe medicines for my patients s...	Hospital Management System\Sprint 3	3	Business	New
10	User Story	> As a pharmacist, I want to validate prescriptions so that me...	Hospital Management System\Sprint 3	3	Business	New
11	User Story	> As an admin, I want to see which doctors have the most con...	Hospital Management System\Sprint 4	3	Business	New

## **Result:**

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

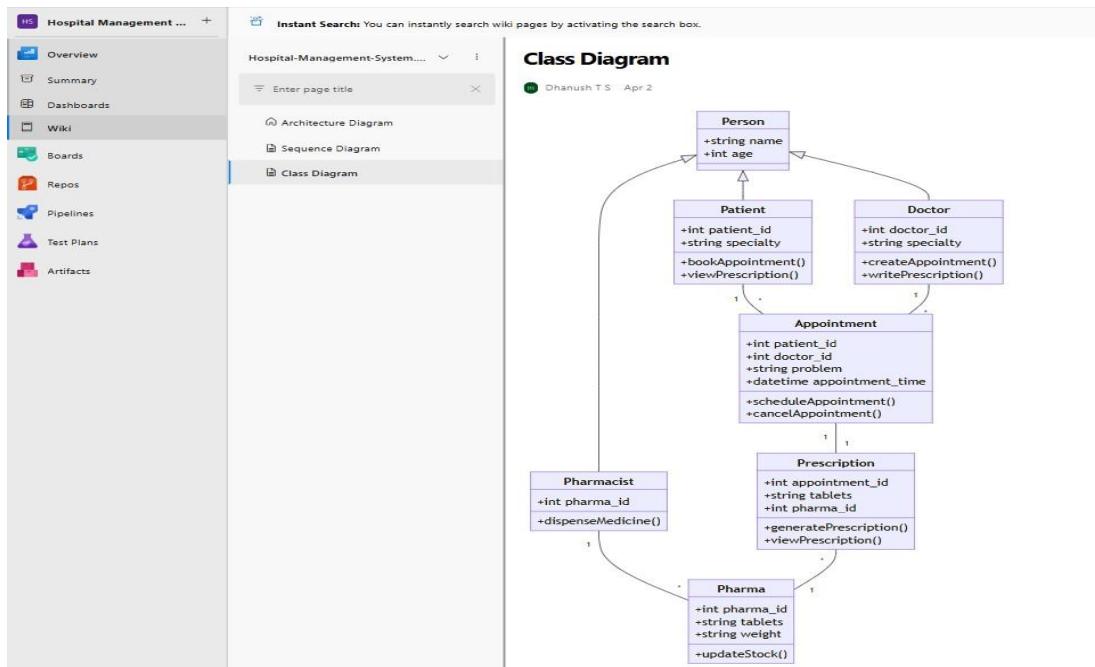
**EXP NO: 6**

## **DESIGNING CLASS AND SEQUENCE DIAGRAMS FOR PROJECT ARCHITECTURE**

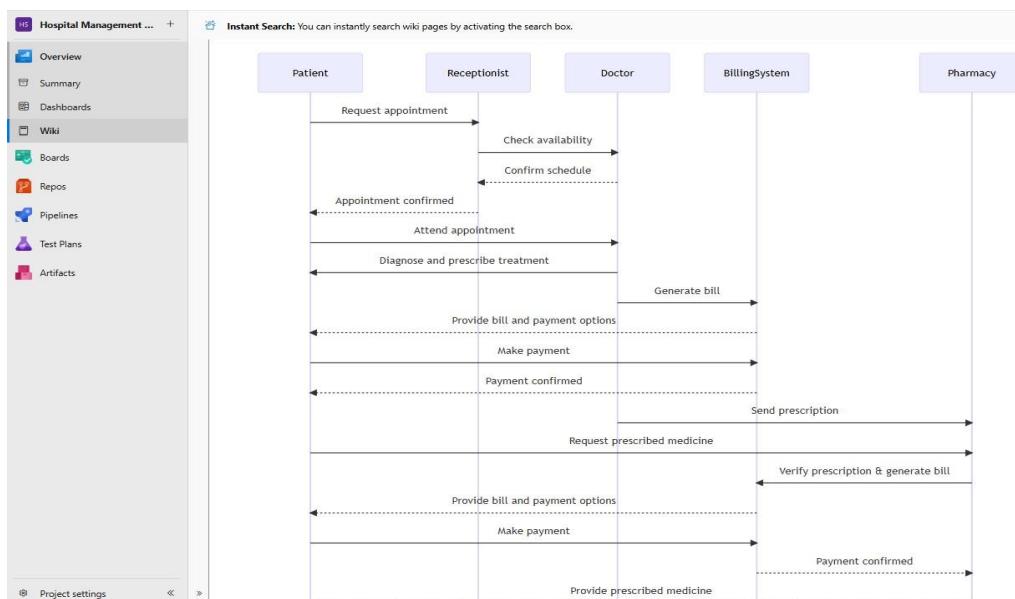
### **Aim:**

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 Hospital Management System.

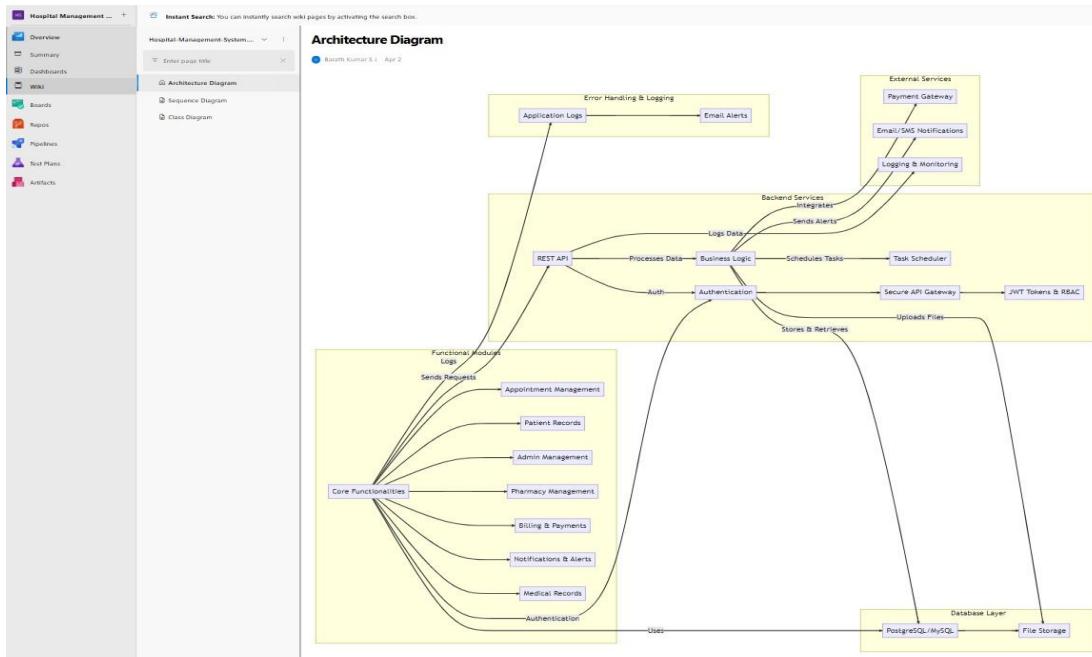
**EXP NO: 7**

## **DESIGNING ARCHITECTURAL AND ER DIAGRAMS FOR PROJECT STRUCTURE**

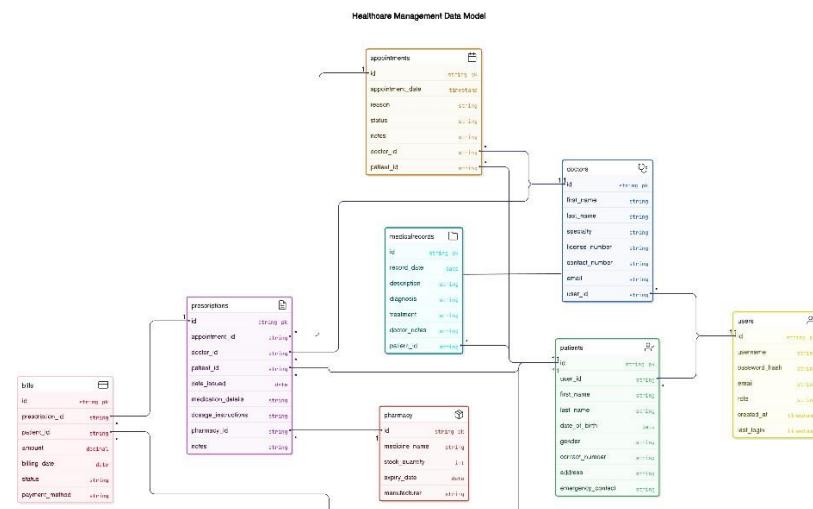
### **Aim:**

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 Hospital Management System.

**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 (Patient, Doctor, Admin)
- Appointment Booking and Availability Management
- Patient Medical Record Viewing and Prescription Entry
- Admin Dashboard for Hospital Data and User Management
- Pharmacy Module for Prescription Fulfillment and Stock Tracking

##### **2. Define User Interactions**

- Each test case simulates a real user behaviour (e.g., Patients booking appointments, Doctors updating availability and prescribing medicines, Admins managing doctor records and stock levels).

##### **3. Design Happy Path Test Cases**

- These test cases validate correct functioning under expected conditions:
- Successful login for each user role
- Patient books an available appointment
- Doctor views medical history and writes a prescription
- Pharmacist fulfills a valid prescription

##### **4. Design Error Path Test Cases**

- These test cases simulate negative or invalid flows:
- Login fails with wrong password
- Patient attempts to book a past/unavailable slot
- Doctor tries to access non-existent patient data
- Admin tries to delete a locked record

##### **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 – Patient Successful Login, TC02 – Invalid Slot

Booking).

- Mapped to relevant user stories and features in Azure DevOps Boards.
- Helps in quick identification and linking to user stories or features.

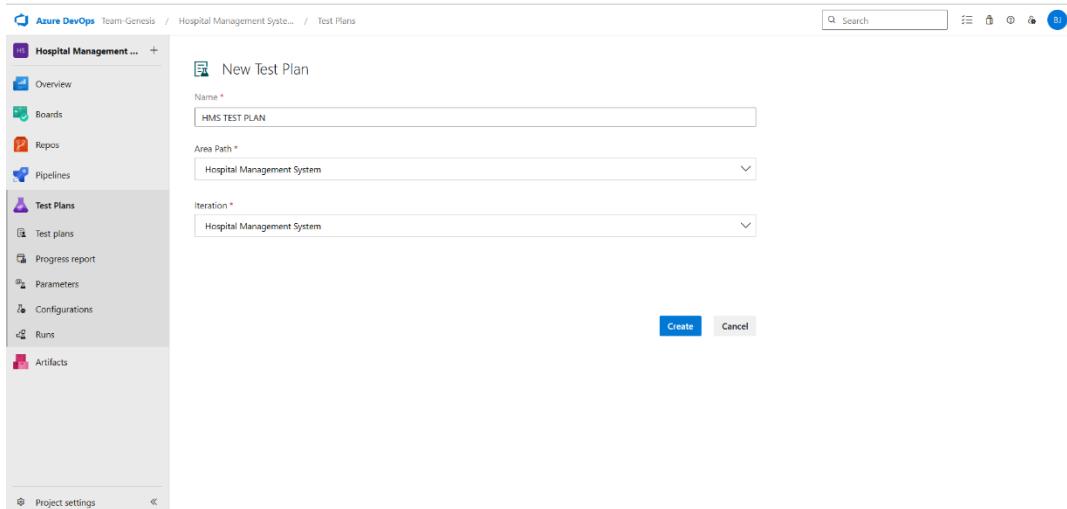
## 7. Separate Test Suites

- Test cases are grouped by functional modules:
- Login & Authentication
- Appointment Management
- Consultation & Prescriptions
- Admin & Role Management
- Pharmacy Management

## 8. Prioritize and Review

- Critical functions like login, appointment booking, and prescription management are prioritized
- Test cases are reviewed for completeness, clarity, and traceability to project requirement

## 1. New test plan



## 2. Test suite

The screenshot shows the 'Patient - Test Cases' page within the 'HMS Test Plan'. The left sidebar shows the 'Test plans' section is selected. The main area displays a list of test cases under the 'Patient' suite. The table shows two items: 'TC01-Book Appointment (HP)' and 'TC02-Book Appointment (ES)'. The table includes columns for 'Title', 'Order', 'Tags', and 'State'.

Title	Order	Tags	State
TC01-Book Appointment (HP)	1	Happy Path	Design
TC02-Book Appointment (ES)	2	Error Scenario	Design

### **3. Test case**

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

HMS – Test Plans

#### **USER STORIES**

- As a **doctor**, I want to update my available time slots so that patients can book appointments.
- As a **patient**, I want to book an available slot with a doctor so that I can schedule a consultation.
- As a **doctor**, I want to prescribe medicines so that patients can get required treatment.
- As an **admin**, I want to manage users and assign roles so that access control is maintained
- As a **pharmacist**, I want to fulfill prescriptions so that patients can receive medicines.

#### **Test Suites**

##### **Test Suit: TS01 - Doctor Availability Management**

###### **1. TC01 – Set Availability Successfully**

- **Action:**
  - Login as Doctor
  - Navigate to “Set Availability”
  - Choose valid date and time slots.
  - Click “Save”.
- **Expected Results:**
  - Availability slots are saved and listed.
  - Fields accept values without error.
- **Type:** Happy Path

###### **2. TC02 – Save Without Selecting Slot**

- **Action:**
  - Login as Doctor
  - Go to “Set Availability”
  - Click “Save” without selecting slot
- **Expected Results:**
  - Error: “Please select a time slot.”
- **Type:** Error 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

### **Test Suit: TS02 - Appointment Booking**

#### **1. TC04 – Patient Books Valid Appointment**

- **Action:**
  - Log in successfully.
  - View available doctors and slot
  - Select slot and click "Book"
- **Expected Results:**
  - Booking confirmation shown.
- **Type:** Happy Path

#### **2. TC05 – Booking Unavailable Slot**

- **Action:**
  - Try booking an already-booked or past slot.
- **Expected Results:**
  - Error: "Slot unavailable or expired."
- **Type:** Error Path

### **Test Suit: TS03 - Prescription Management**

#### **1. TC06 – Doctor Prescribes Medicines**

- **Action:**
  - Login as Doctor
  - Open patient consultation
  - Add medicines and dosage
  - Click "Submit"
- **Expected Results:**
  - Prescription saved and shown in record
- **Type:** Happy Path

#### **2. TC07 – Prescription Without Medicines**

- **Action:**

- Try submitting prescription form without entering medicine details
- **Expected Results:**
  - Error: “Medicine field cannot be empty.”
- **Type:** Error Path

#### Test Suit: TS04 - Admin Role Management

##### 1. TC08 – Assign Role to User

- **Action:**
  - Login as Admin
  - Go to “User Management”
  - Select a user and assign role (doctor/patient/pharmacist)
- **Expected Results:**
  - Role updated successfully.
- **Type:** Happy Path

##### 2. TC09 – Assign Role to Invalid User

- **Action:**
  - Try assigning a role to a non-existing or deleted user
- **Expected Results:**
  - Error: “User not found.”
- **Type:** Error Path

## Test Suit: TS05 - Prescription Fulfillment (Pharmacy)

### 1. TC10 – Fulfill Prescription

- **Action:**
  - Login as Pharmacist
  - View assigned prescriptions
  - Mark as fulfilled and issue medicine
- **Expected Results:**
  - Status updated: "Fulfilled".
- **Type:** Happy Path

### 2. TC11 – Fulfill Without Prescription

- **Action:**
  - Try fulfilling without selecting a prescription
- **Expected Results:**
  - Error message: "No prescription selected.".
- **Type:** Error Path

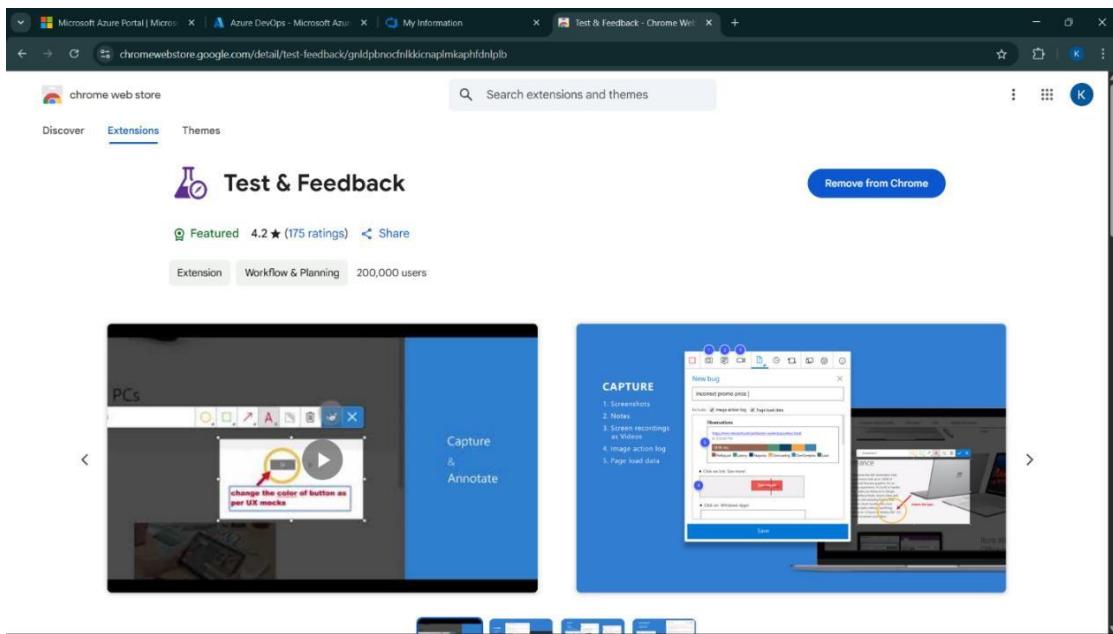
## Test Cases

The screenshot shows a Microsoft Azure DevOps interface for a test plan. The main view displays a test case titled "77 TC06 – Playlist Loading Failure" created by "Karthick S". The test case is categorized under "Music Playlist Batch Creator" and "Music Playlist Batch Creator[Integration]". The "Steps" section contains two steps: 1. "Disconnect from internet" (Expected result: Network is offline) and 2. "Navigate to 'My Playlists'" (Expected result: Error message "Unable to load playlists" is shown). The "Status" section indicates a priority of 2 and an automation status of "Not Automated". The left sidebar shows navigation links for Music, Overview, Boards, Repos, Pipelines, Test Plans, Programs, Parameters, Configurations, Runs, and Artifacts. The bottom navigation bar includes "Project settings" and "Help".

A screenshot of the Azure DevOps Test Plan interface. The page shows a test case titled "TC05 – View Playlist Page" created by "Karthick S". The test case is in the "Design" state and is associated with the "Music Playlist Batch Creator" area and iteration. It has two steps: "Log in successfully" (Expected result: User is redirected to dashboard) and "Navigate to 'My Playlists' section" (Expected result: All created playlists are displayed clearly). The status is set to "Priority 2" and "Not Automated". The "Steps" tab is selected, and the "Associated Automation" tab is visible at the bottom.

#### 4. Installation of test

A screenshot of the Chrome Web Store page for the "Test & Feedback" extension. The extension is featured with a rating of 4.2 stars from 175 ratings and has 200,000 users. It is categorized under "Discover" and "Extensions". The main image shows a screenshot of a browser window with a red box highlighting a button and the text "change the color of button as per UX mockups". Below the main image, there is a "CAPTURE" section with options for screenshots, notes, screen recordings, image action log, and page load data. A "New bug" dialog box is also shown.



## Test and feedback

Showing it as an extension

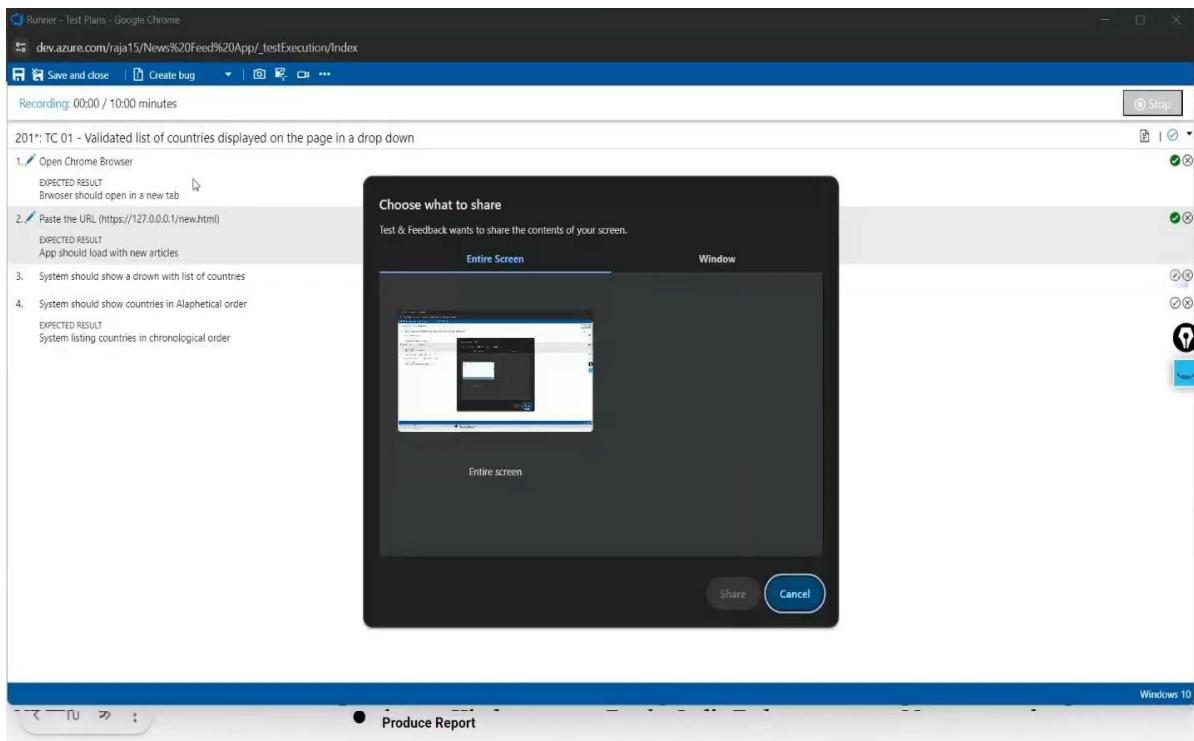
A screenshot of a Microsoft Edge browser window showing the Azure DevOps Test Plan interface. The URL is dev.azure.com/231801095/Music%20Playlist%20Batch%20Creator/\_testPlans/define?planId=84&amp;suiteId=86. On the left, the navigation menu is open, showing "Test Plans" selected. In the center, a test plan named "TS01 - User Login (ID: 86)" is displayed with four test cases: "Title", "TC01 - Successful Sign Up", "TC02 - Secure Login", "TC03 - Sign Up with Existing Email", and "TC04 - Login with Wrong Password". A floating sidebar titled "Extensions" is visible on the right, listing several extensions with "Full access": "Copy Text from Picture", "Dark Reader", "Monica: ChatGPT AI Assist...", "Selectext: Copy text from V...", and "Test &amp; Feedback". The "Test &amp; Feedback" extension is highlighted with a blue border.

## 5. Running the test cases

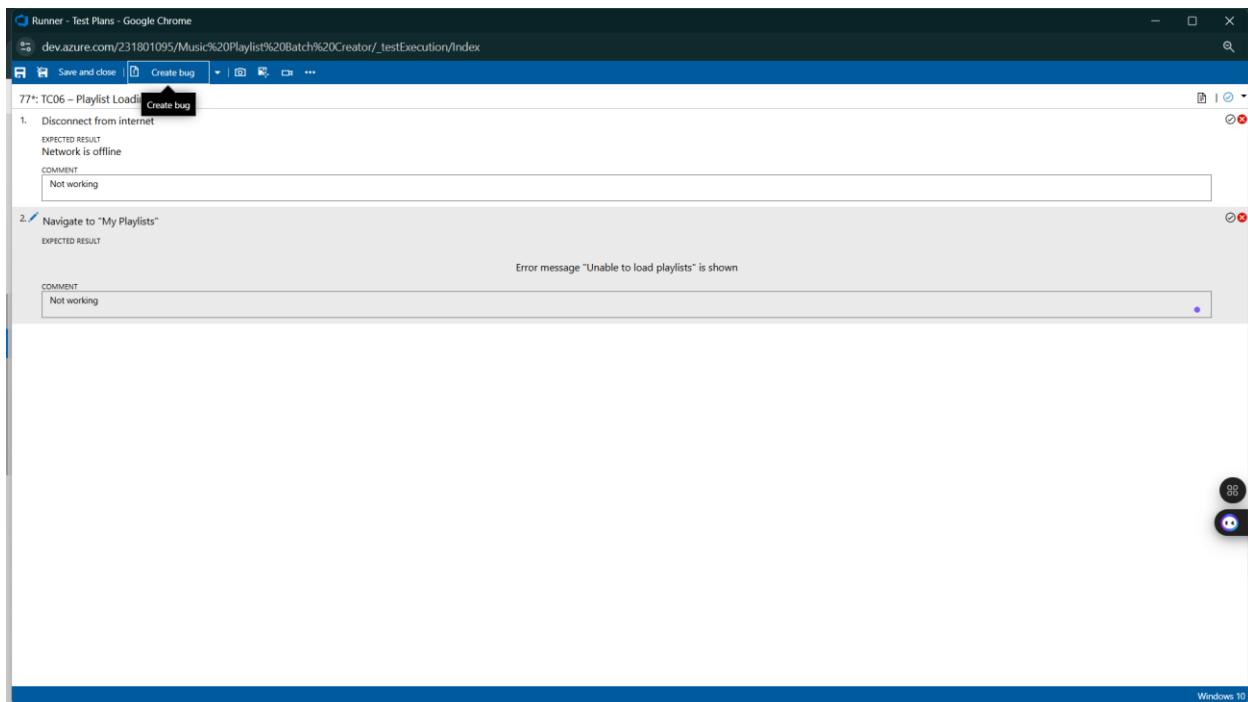
The screenshot shows the Azure DevOps Test Plan interface. On the left, the navigation bar includes 'Overview', 'Boards', 'Repos', 'Pipelines', 'Test Plans', 'Test plans' (selected), 'Progress report', 'Parameters', 'Configurations', 'Runs', and 'Artifacts'. Under 'Test plans', 'Music Playlist Batch Creator - T...' is expanded, showing 'TS01 - User Login (4)', 'TS02 - View Playlists (2)' (selected), 'TS03 - Real-Time Met...', 'TS04 - Playlist Editing (4)', and 'TS05 - Smart Playlist ...'. The main area displays 'TS02 - View Playlists (ID: 87)'. It has tabs for 'Define', 'Execute' (selected), and 'Chart'. Under 'Test Points (2 items)', there are two entries: 'TC05 – View Playlist Page' (Passed) and 'TC06 – Playlist Loading Failure' (Not run). A context menu is open over 'TC05 – View Playlist Page', listing options: 'View execution history', 'Mark Outcome' (checkbox checked), 'Run' (disabled), 'Reset test to active', 'Edit test case', 'Assign tester', and 'View test result'. The status bar at the bottom right shows 'Windows 10'.

The screenshot shows the 'Runner - Test Plans - Google Chrome' window. The URL is 'dev.azure.com/231801095/Music%20Playlist%20Batch%20Creator/\_testExecution/Index'. The page title is '75\*: TC05 – View Playlist Page'. The steps listed are: 1. Log in successfully (EXPECTED RESULT: User is redirected to dashboard). 2. Navigate to "My Playlists" section (EXPECTED RESULT: All created playlists are displayed clearly). The status bar at the bottom right shows 'Windows 10'.

## 6. Recording the test case



## 7. Creating the bug



Runner - Test Plans - Google Chrome  
dev.azure.com/231801095/Music%20Playlist%20Batch%20Creator/\_testExecution/Index

77: TC06 – Playlist Loading Failure

1. Disconnect from internet  
EXPECTED: Network is offline  
ACTUAL: Playlist loading spinner keeps spinning indefinitely on poor network

2. Navigate to "My Playlists"  
EXPECTED: New  
ACTUAL: New

**Repro Steps**

18-04-2025 03:23 Bug filed on "TC06 – Playlist Loading Failure"

Step no. Result Title  
1. Failed Disconnect from internet  
Expected Result  
Network is offline  
Comments: Page Not loading  
2. Failed Navigate to "My Playlists".  
Expected Result  
Error message "Unable to load playlists" is shown

**Test Configuration:** Windows 10

**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.

**Related Work**  
+ Add link  
Add an existing work item as a parent  
Tested By 77 TC06 – Playlist Loading Failure  
Updated 10-04-2025, ● Design

**System Info**  
Found in Build

Microsoft Azure Portal | Azure DevOps - Microsoft | My Information | Test Plan 84 Music | Runs - Test Plans | Settings - Overview | Bug Report Playlist | +

dev.azure.com/231801095/Music%20Playlist%20Batch%20Creator/\_testManagement/runs?a=resultSummary&runId=48&resultId=100000

Azure DevOps 231801095 / Music Playlist Batch Creator / Test Plans / Runs

Run 48 - TS02 - View Playlists (Manual) / TC06 – Playlist Loading Failure

92 TB01 - Playlist loading spinner keeps spinning indefinitely on poor network

Unassigned 0 comments Add tag

State: New Area: Music Playlist Batch Creator  
Reason: New Iteration: Music Playlist Batch Creator

**System Info**

Browser - Name	Google Chrome 135
Browser - Language	en-IN
Browser - Height	864
Browser - Width	1536
Browser - User agent	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/135.0.0.0 Safari/537.36
Operating system - Name	Windows NT 10.0; Win64; x64
Operating system - Architecture	x64_64
Operating system - Processor mode	11th Gen Intel(R) Core(TM) i3-1115G4 @ 3.00GHz
Operating system - Number of processors	4
Memory - Available	814784512
Memory - Capacity	8216240128
Display - Pixels per inch (X axis)	120
Display - Pixels per inch (Y axis)	120
Display - Device pixel ratio	125

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

Name: SystemInformation-2025-04-18T03-23-58.168Z.json  
Size: 1K

Project settings

2116231801033

CS23432

## 8. Test case results

The screenshot shows the Azure DevOps interface for a test plan. On the left, the navigation pane is open with 'Test Plans' selected. In the center, a test suite named 'TS02 - View Playlists (ID: 87)' is displayed under 'Test Suites'. This suite contains two test points: 'TC05 – View Playlist Page' (selected) and 'TC06 – Playlist Loading Failure'. On the right, a detailed view of 'TC05 – View Playlist Page' is shown, titled 'TC05 – View Playlist Page'. It displays a table of 'Test Case Results' with columns: Outcome, TimeStamp, Configuration, Run by, Tester, and Test Plan. The results show various passes and one failure (Failed) for different dates and configurations.

Outcome	TimeStamp	Configuration	Run by	Tester	Test Plan
Passed	4m ago	Windows 10	Karthick S	Malu karthick Balaji ... Music	
Passed	12m ago	Windows 10	Karthick S	Malu karthick Balaji ... Music	
Not Applicable	12m ago	Windows 10	Karthick S	Malu karthick Balaji ... Music	
Passed	14m ago	Windows 10	Karthick S	Malu karthick Balaji ... Music	
Passed	Tuesday	Windows 10	Karthikayen Senthil	Malu karthick Balaji ... Music	
Passed	Saturday	Windows 10	Malu karthick Balaji ...	Malu karthick Balaji ... Music	
Failed	Saturday	Windows 10	Malu karthick Balaji ...	Malu karthick Balaji ... Music	
Passed	Apr 11	Windows 10	Karthick S	Malu karthick Balaji ... Music	
Passed	Apr 11	Windows 10	Karthick S	Malu karthick Balaji ... Music	

## 9. Test report summary

The screenshot shows the Azure DevOps interface for a work item. The left sidebar shows the project structure with 'News Feed App' selected. The main area displays a bug report titled 'BUG 203: BG 01 - Countries Drop down Not Available on the page'. The work item details include:

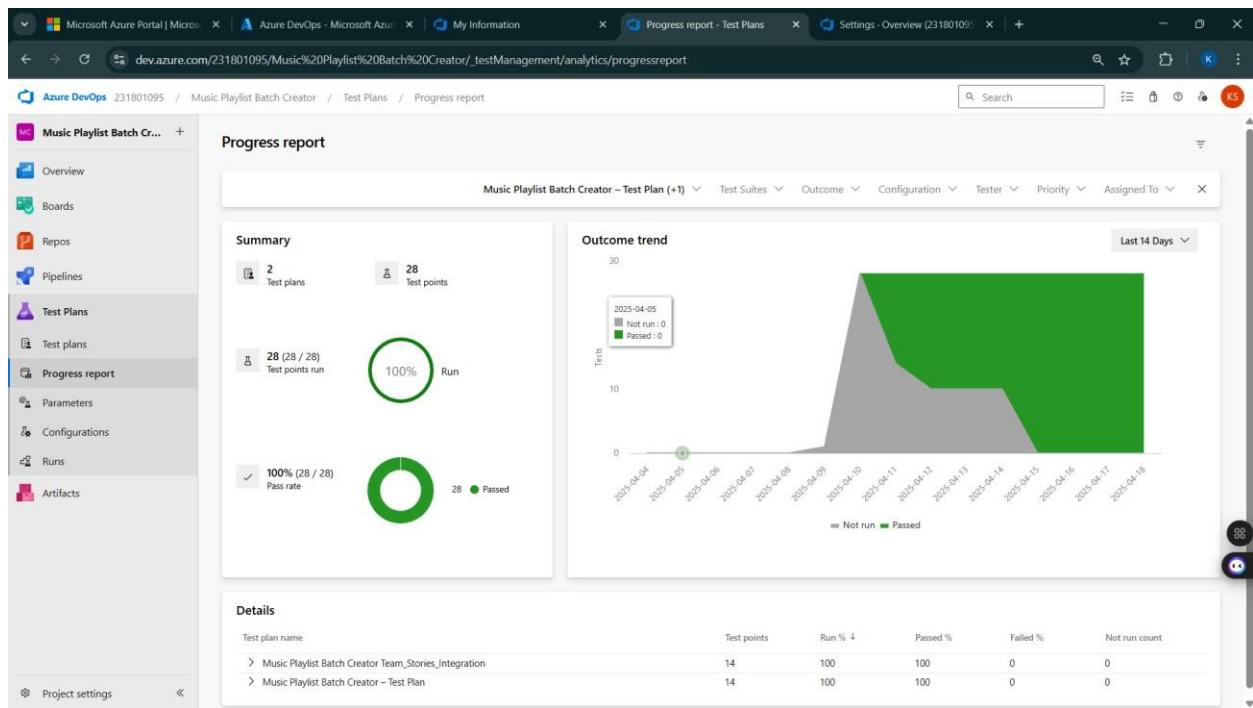
- State:** New
- Reason:** New
- Repro Step:** Active
- Details:** Updated by rajesh prabhu: Just now
- Steps:**
  - Result:** Passed  
**Title:** Open Chrome Browser  
**Expected Result:** Browser should open in a new tab  
**Actual Result:** App should load with new articles
  - Result:** Passed  
**Title:** Paste the URL (<https://127.0.0.1/new.html>)  
**Expected Result:** App should load with new articles  
**Actual Result:** Success! Should show a dropdown with list of countries
  - Result:** Failed  
**Title:** Produce Report  
**Actual Result:** Success! Should show a dropdown with list of countries
- 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
- Related Work:** Related work items

- Assigning bug to the developer and changing state

The screenshot shows a detailed view of a test step in the Azure DevOps Test Plan interface. The step, titled '92 TB01 - Playlist loading spinner keeps spinning indefinitely on poor network', has a status of 'Failed'. It includes two steps: 1. Failed (Disconnect from internet) and 2. Failed (Comments: Page Not loading). The 'Test Configuration' is set to Windows 10. The 'Planning' section shows a priority of 2 and a severity of 3 (Medium). The 'Deployment' section indicates the item is updated by Karthick S 33m ago. The 'Development' section links to an Azure Repos branch named 'TC06 - Playlist Loading Failure'. The 'Related Work' section lists a task '77 TC06 - Playlist Loading Failure' with a status of 'Design' and an update time of 10-04-2025. The 'System Info' section shows the item was tested by Barath Kumar S J.

## 10. Progress report

The screenshot displays the 'Progress report' for the 'HMS Test Plan'. The summary shows 1 test plan, 8 test points, and 8 test runs, all of which have passed (100% pass rate). The 'Outcome trend' chart tracks the number of tests over time, showing a sharp increase starting around April 14th, with all tests passing ('Passed') after April 15th. The 'Details' section provides a breakdown of the test plan's performance, including 8 test points, 100% run completion, 100% passed, 0 failed, and 0 not run.



## 11. Changing the test template

**Organization Settings**

**All processes**

Name	Description	Team projects
Basic	This template is flexible for any process and great for teams getting started with Azure DevOps.	0
Agile	This template is flexible and will work great for most teams using Agile planning methods, including those pract...	0
Scrum	This template is for teams who follow the Scrum framework.	0
CMMI	This template is for more formal projects requiring a framework for process improvement and an auditable reco...	0

The screenshot shows the 'All processes' list in the Azure DevOps Settings - Process page. The 'Agile' template is selected, indicated by a highlighted row. The table columns are Name, Description, and Team projects.

Name	Description	Team projects
Basic	This template is flexible for any process and great for teams getting started with Azure DevOps.	0
Agile	This template is flexible and will work great for most teams using Agile planning methods, including those pract...	0
Scrum	This template is for teams who follow the Scrum framework.	0
CMMI	This template is for more formal projects requiring a framework for process improvement and an auditable reco...	0

The screenshot shows the 'All processes' list in the Azure DevOps Settings - Process page. A new template named '231801095 Agile (default)' has been added under the Agile category, indicated by a new row. The table columns are Name, Description, and Team projects.

Name	Description	Team projects
Basic	This template is flexible for any process and great for teams getting started with Azure DevOps.	0
Agile	This template is flexible and will work great for most teams using Agile planning methods, including those pract...	0
231801095 Agile (default)		1
Agile Plus		0
Scrum	This template is for teams who follow the Scrum framework.	0
CMMI	This template is for more formal projects requiring a framework for process improvement and an auditable reco...	0

## 12. View the new test case template

The screenshot shows the 'Add a field to Test Case' dialog box over a background of the Azure DevOps settings interface. The dialog has tabs for 'Definition', 'Options', and 'Layout'. Under 'Definition', the 'Create a field' option is selected, with 'Name' set to 'Type' and 'Type' set to 'Text (single line)'. A 'Description' field is also present. At the bottom are 'Add field' and 'Cancel' buttons.

The screenshot shows the 'Work-item types' section of the Azure DevOps settings. It lists a single item: 'Music Playlist Batch Creator'. The 'Name' column shows the item name, and the 'Description' column provides a brief overview: 'The Azure Music Playlist Batch Creator is a cloud-based solution designed for bulk playlist creation and management. Leverage...'. The 'General' tab is selected in the sidebar.

The screenshot shows the Azure DevOps Settings - Process page for a specific process. The left sidebar is titled "Organization Settings" and includes sections for General, Security, Boards, Pipelines, and Process. The "Process" section is currently selected. The main content area is titled "All processes > 231801095 Agile > Test Case". It displays a configuration interface with tabs for "Steps", "Summary", and "Associated Art...". A "Custom" section is visible on the right, containing fields for "Type" (Text (single line)), "Recent test results", "Deployment", "Development", "Related Work", and "Status". The "Steps" tab is active, showing a placeholder "Text (multiple lines)".

### 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	<b>LOAD TESTING AND PIPELINES</b>
-----------	-----------------------------------

**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 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
  - 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://team-genesis.netlify.app/>).
4. Click Review + Create → Create to start the test

## Load Testing:

Microsoft Azure

Home > HMS-LOADTEST | Tests >

TestRun\_4/20/2025\_10:35:33 PM

Last updated by: 231801020@rajalakshmi.edu.in | Initiated on: 4/20/2025, 10:35 PM

View all test runs Stop Refresh Rerun Compare App components Configure metrics Download Copy artifacts Share Delete test run Mark as baseline Auto refresh off

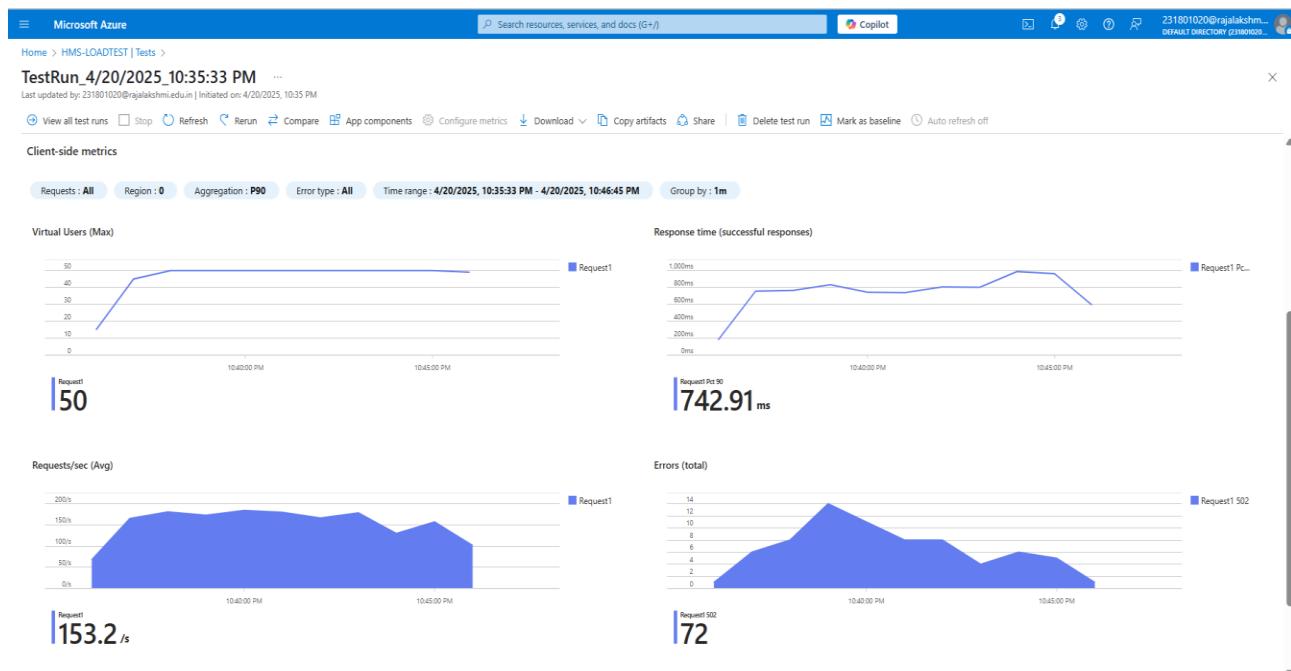
Test run details

Start time	End time	Test run ID	Test type	Engine instances	Debug mode	Test result	Status
		d5cdca84-013d-46c5-b8c4-a632f...	URL	1	Disabled	Not Applicable	Done

Load test results Engine health

Statistics

Load 101113 Total requests	Duration 10 mins, 8 secs	Response time 775.00 ms 90th percentile response time	Error percentage 0.07 % Aggregate requests which failed	Throughput 166.30 /s Request rate
----------------------------------	-----------------------------	---	---	---



 **HMS**

Home About Services Contact [Login](#) [Sign Up](#)

# Modern Healthcare Management Solution

Streamline your hospital operations with our comprehensive management system designed for doctors, patients, and administrators.

[Explore Features →](#) [Contact Us](#)



Team Genesis | HMS [team-genesis.netlify.app/services](#)

 **HMS**

Home About Services Contact [Login](#) [Sign Up](#)

## Core Features

Our hospital management system offers a comprehensive suite of features


**Appointment Scheduling**

Doctors can update their available time slots and patients can book appointments with ease.


**Real-time Availability**

Check doctor availability and schedule appointments in real-time with instant confirmation.


**User Management**

Comprehensive user management for doctors, patients, admins, and pharmacists.


**Prescription Management**

Doctors can prescribe medicines digitally and patients can access their prescriptions online.


**Secure & Reliable**

Multi-factor authentication and 99.9% uptime ensure your data is safe and always accessible.


**Performance Optimized**

Fast loading times with patient records available in under 2 seconds, even under high load.


**Data Analytics**

Gain insights into hospital operations with comprehensive reporting and analytics tools.


**Inventory Management**

Track and manage medicine stock and hospital supplies with automated alerts for low inventory.


**Communication Tools**

Secure messaging system for internal communication between staff members.

Acme Inc [Enterprise](#)

Items

- Home
- Doctors
- Appointments
- Pharma Orders
- Bills
- History

Download

Dashboard

Overview Analytics Reports Notifications

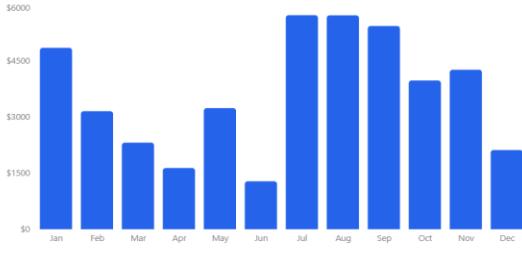
Total Revenue **\$45,231.89**  
+20.1% from last month

Subscriptions **+2350**  
+180.1% from last month

Sales **+12,234**  
+19% from last month

Active Now **+573**  
+201 since last hour

Overview



Recent Sales

You made 265 sales this month.

Initials	Name	Email	Total Sales
OM	Olivia Martin	olivia.martin@email.com	+\$1,999.00
JL	Jackson Lee	jackson.lee@email.com	+\$39.00
IN	Isabella Nguyen	isabella.nguyen@email.com	+\$299.00
WK	William Kim	will@email.com	+\$99.00
SD	Sofia Davis	sofia.davis@email.com	+\$39.00

shadcn m@example.com

## **Pipelines:**

### **Description:**

This experiment demonstrates how to connect 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 Hospital Management System(HMS)!')"
  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. Make sure the path to requirements.txt is correct (it is 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 interface. The left sidebar has 'Hospital Management ...' selected under 'Pipelines'. The main area is titled 'Pipelines' with tabs 'Recent', 'All', and 'Runs'. Under 'Recently run pipelines', there is one entry: 'dhanush-ts.Hospital-Management-Syst...' with a green checkmark icon, indicating it was triggered manually for branch 'main' on 'Yesterday'. A search bar and a 'New pipeline' button are at the top right.

## Result:

Successfully created the Azure Load Testing resource and executed a load test to assess the performance of the specified endpoint and also demonstrated pipelines in Azure Devops.

**EXP NO: 10**

## **GITHUB: PROJECT STRUCTURE & NAMING CONVENTIONS**

### **Aim:**

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

### **GitHub Project Structure:**

The screenshot shows a GitHub repository page for 'Hospital-Management-System-web'. The repository was created by 'dhanush-ts' and has 1 branch and 0 tags. It contains 20 commits from 'dhanush-ts' over the last week. The commit history includes updates to 'layout.js', 'Header.js', and 'appointments' files across 'app', 'components', 'hooks', 'lib', and 'public' directories. Other files visible include '.gitignore', 'README.md', 'azure-pipelines.yml', 'components.json', 'jsconfig.json', 'next.config.mjs', 'package-lock.json', 'package.json', 'postcss.config.mjs', and 'tailwind.config.mjs'. The repository has 1 watcher, 0 forks, and 0 stars. It also has 1 deployment to 'github-pages' 2 weeks 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.