



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: Barath Kumar S J

Register No: 231801020

Year / Branch / Section: 2nd / 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:

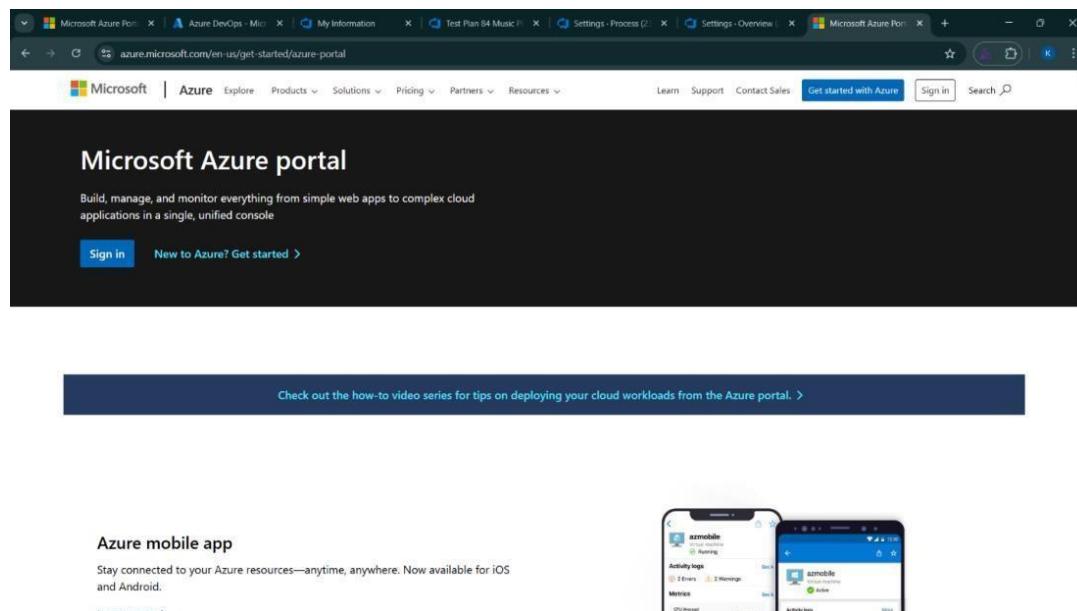
To set up and access the Azure DevOps environment by creating an organization through the Azure portal.

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>



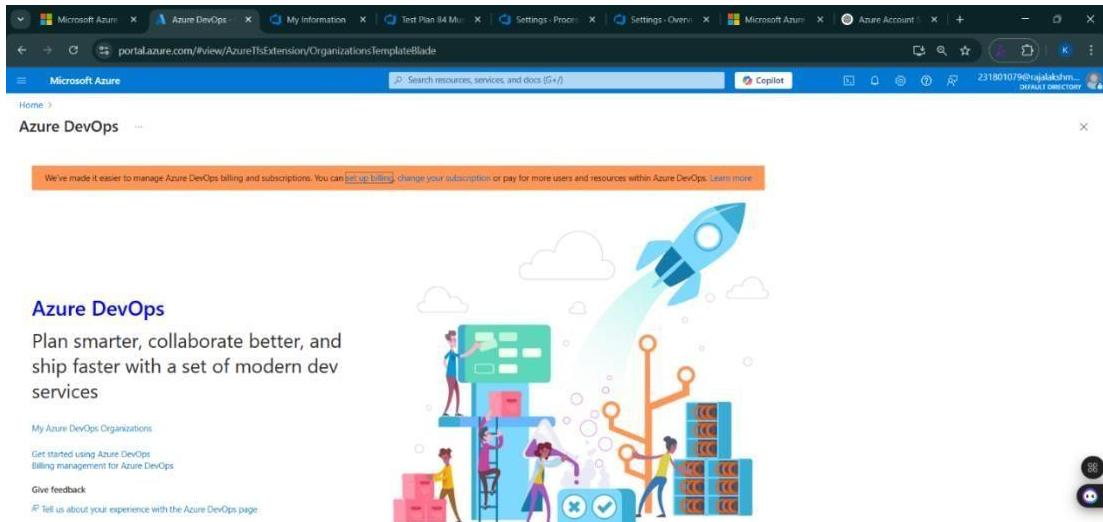
2. Azure home page

The screenshot shows the Microsoft Azure home page. At the top, there's a navigation bar with tabs like 'My Information', 'Test Plan 84 Min...', 'Settings - Process', 'Settings - Overview', 'Microsoft Azure', and 'Azure Account'. Below the navigation bar is the 'Search resources, services, and docs (S+)' search bar. The main content area is titled 'Azure services' and 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'. Below this is a 'Resources' section with a 'Recent' tab showing two items: 'Music' (Azure Load Testing) and 'Music_playlist_Batch_Creator' (Resource group), both last viewed 3 days ago. There's also a 'See all' link. A 'Navigate' section includes links for 'Subscriptions', 'Resource groups', 'All resources', and 'Dashboard'. A 'Tools' section has links for 'Microsoft Learn', 'Azure Monitor', 'Microsoft Defender for Cloud', and 'Cost Management'. At the bottom, there's a 'Useful links' section and an 'Azure mobile app' link.

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

This screenshot is similar to the previous one but with a search bar at the top containing the text 'DevOps'. The rest of the interface is identical, showing the Azure services dashboard, recent resources, and various tools and documentation links.

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.

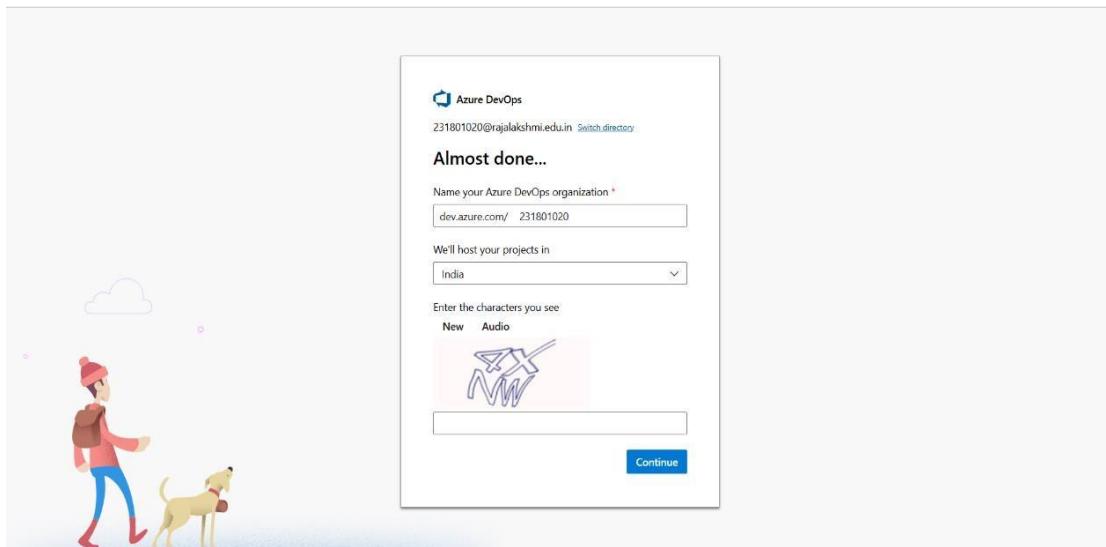
EXP NO: 2

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. 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).
- 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.

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 interface. On the left, there's a profile card for Barath Kumar S J with a blue circular icon containing 'BJ'. Below it, his name and email (231801020@rajalakshmi.edu.in) are listed, along with a Microsoft account dropdown showing 'India' and an email address. A 'Visual Studio Dev Essentials' section is present. On the right, under 'Azure DevOps Organizations', it shows 'dev.azure.com/Team-Genesis (Owner)' with a 'Hospital Management System' project listed. There are 'Actions' like 'Open in Visual Studio' and a 'Create new organization' button.

4. Project dashboard

The screenshot shows the Azure DevOps Project Overview for 'Hospital Management System'. The left sidebar has a navigation menu with 'Overview', 'Summary', 'Dashboards', 'Wiki', 'Boards', 'Repos', 'Pipelines', 'Test Plans', and 'Artifacts'. The main area displays the project title 'Hospital Management System' and sections for 'About this project', 'Project stats', and 'Members'. The 'About this project' section details the system as an Azure-Powered Hospital Management System (HMS) designed to streamline operations. It lists key features like appointment scheduling, electronic medical records, prescription management, billing, and notifications. The 'Project stats' section shows 16 work items created and 0 completed. The 'Members' section shows five 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 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)
 - 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)
 - Task: Create UI for doctors to set availability
 - Task: Store available slots in the database.
 - Task: Allow doctors to update or delete available 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.

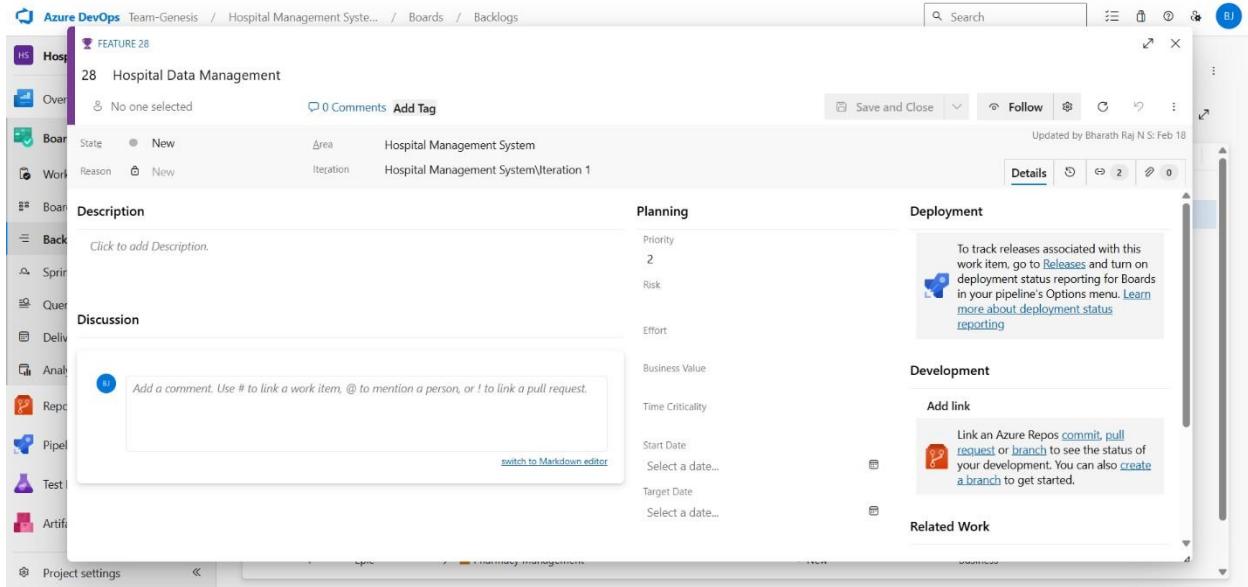
1. Fill in Epics

The screenshot shows the 'Edit Work Item' dialog for an Epic titled '26 - Admin Management'. The dialog is divided into several sections:

- General:** State is set to 'New', Area is 'Hospital Management System', Iteration is 'Hospital Management System\Iteration 1'.
- Description:** A placeholder text 'Click to add Description.'
- Planning:** Priority is '2', Risk is 'Low', Effort is 'Medium', Business Value is 'High', Time Criticality is 'Medium'.
- Deployment:** A note: 'To track releases associated with this work item, go to Releases and turn on deployment status reporting for Boards in your pipeline's Options menu. Learn more about deployment status reporting.'
- Development:** A note: '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:** A section for linking work items.

The left sidebar shows the project navigation and the current board.

2. Fill in Features

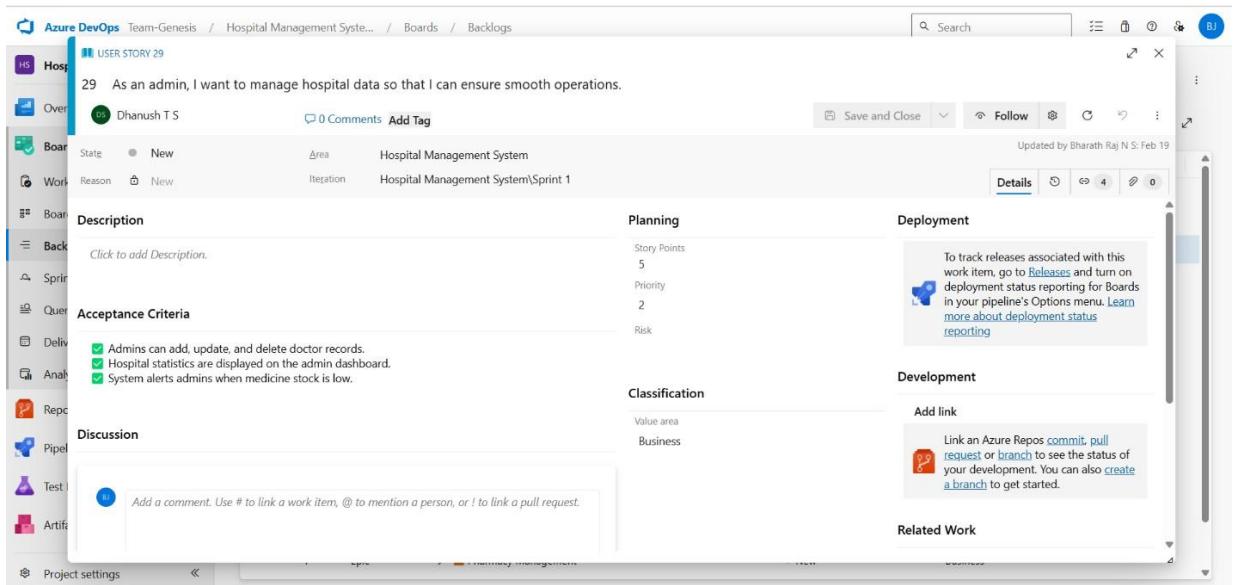


Azure DevOps Backlogs screen showing the creation of a Feature work item. The work item details are as follows:

- Title:** FEATURE 28
- Description:** 28 Hospital Data Management
- Area:** Hospital Management System
- Iteration:** Hospital Management System\Iteration 1
- Priority:** 2
- Risk:** Low
- Business Value:** Not specified
- Time Criticality:** Not specified
- Start Date:** Not specified
- Target Date:** Not specified

The "Deployment" section includes a note about tracking releases and a "Development" section for linking Azure Repos.

3. Fill in User Story Details



Azure DevOps Backlogs screen showing the creation of a User Story work item. The work item details are as follows:

- Title:** USER STORY 29
- Description:** 29 As an admin, I want to manage hospital data so that I can ensure smooth operations.
- Assignee:** Dhanush T S
- Comments:** 0
- Area:** Hospital Management System
- Iteration:** Hospital Management System\Sprint 1
- Story Points:** 5
- Priority:** 2
- Risk:** Low
- Value area:** Business

The "Acceptance Criteria" section lists three items:

- Admins can add, update, and delete doctor records.
- Hospital statistics are displayed on the admin dashboard.
- System alerts admins when medicine stock is low.

The "Classification" section indicates the value area is Business.

The "Development" section includes a note about linking Azure Repos.

Result:

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

EXP NO: 4

SPRINT PLANNING

Aim:

To assign user story to specific sprint for the 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 'Sprints' under 'Backlogs'. The main area displays a backlog for 'Sprint 1' with four user stories listed:

- 32 Track inventory and update medicine stock with low stock alerts. Assigned to: Dhanush TS.
- 42 Develop a UI to update outgoing medicine stock based on prescriptions. Assigned to: alvin.
- 43 Create functionality to alert admins when stock is low. Assigned to: alvin.
- 44 Allow pharmacists to...

The top right corner shows the sprint duration: January 29 - February 4, 5 work days.

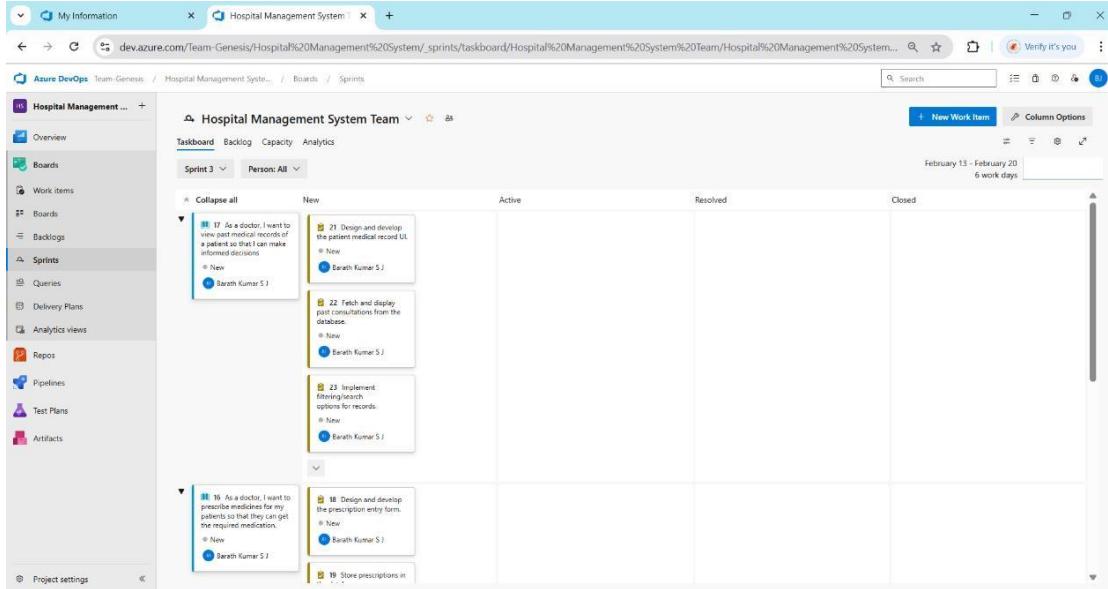
Sprint 2

The screenshot shows the Azure DevOps Boards interface for the Hospital Management System Team. The left sidebar is expanded to show 'Sprints' under 'Backlogs'. The main area displays a backlog for 'Sprint 2' with eight user stories listed:

- 7 As a doctor, I want to update my available time slots so that patients can book appointments. Assigned to: Sharath Raj N S.
- 8 Create UI for doctors to set availability. Assigned to: Sharath Raj N S.
- 9 Store available slots in the database. Assigned to: Sharath Raj N S.
- 10 Allow doctors to update or delete available slots. Assigned to: Sharath Raj N S.
- 11 As a patient, I want to book an available slot with a doctor so that I can consult them. Assigned to: Sharath Raj N S.
- 12 Create UI for patients to view available slots. Assigned to: Sharath Raj N S.
- 13 Validate booking conflicts and slot availability.

The top right corner shows the sprint duration: February 5 - February 12, 6 work days.

Sprint 3



The screenshot shows the Azure DevOps Taskboard for the Hospital Management System Team during Sprint 3. The board is organized into columns: New, Active, Resolved, and Closed. The New column contains several items, many of which are linked to user stories from the backlog. One item is highlighted with a yellow border:

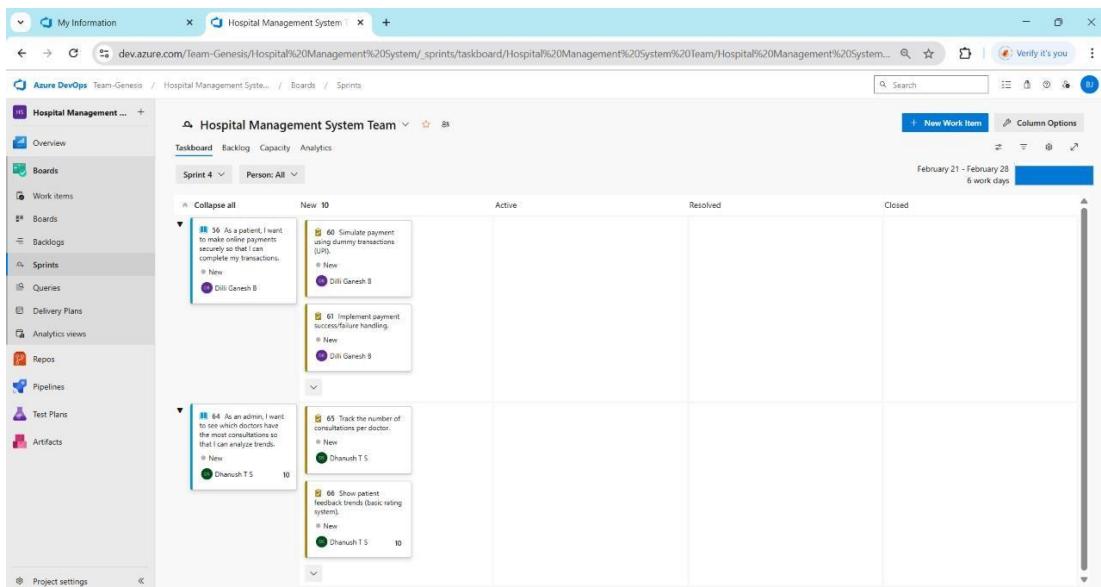
- 21 Design and develop the patient medical record UI.
- 22 Fetch and display past consultations from the database.
- 23 Implement filtering/search options for records.
- 10 As a doctor, I want to prescribe medicines for my patients so that they can get required medication.
- 11 Design and develop the prescription entry form.
- 19 Store prescriptions in

The Active column has one item:

- 21 Design and develop the patient medical record UI.

The Resolved and Closed columns are currently empty.

Sprint 4



The screenshot shows the Azure DevOps Taskboard for the Hospital Management System Team during Sprint 4. The board is organized into columns: New, Active, Resolved, and Closed. The New column contains several items, many of which are linked to user stories from the backlog. One item is highlighted with a yellow border:

- 56 As a patient, I want to make online payments securely so that I can complete my transactions.
- 60 Simulate payment using dummy transactions (UP).
- 61 Implement payment success/failure handling.
- 64 As an admin, I want to see which doctors have the most consultations so that I can analyze trends.
- 65 Track the number of consultations per doctor.
- 66 Show patient feedback trends (basic rating system).

The Active column has one item:

- 60 Simulate payment using dummy transactions (UP).

The Resolved and Closed columns are currently empty.

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, Iteration Path, Story Points, Value Area, and State. The backlog contains 11 User Stories, each with a brief description starting with 'As a...'.

| 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.

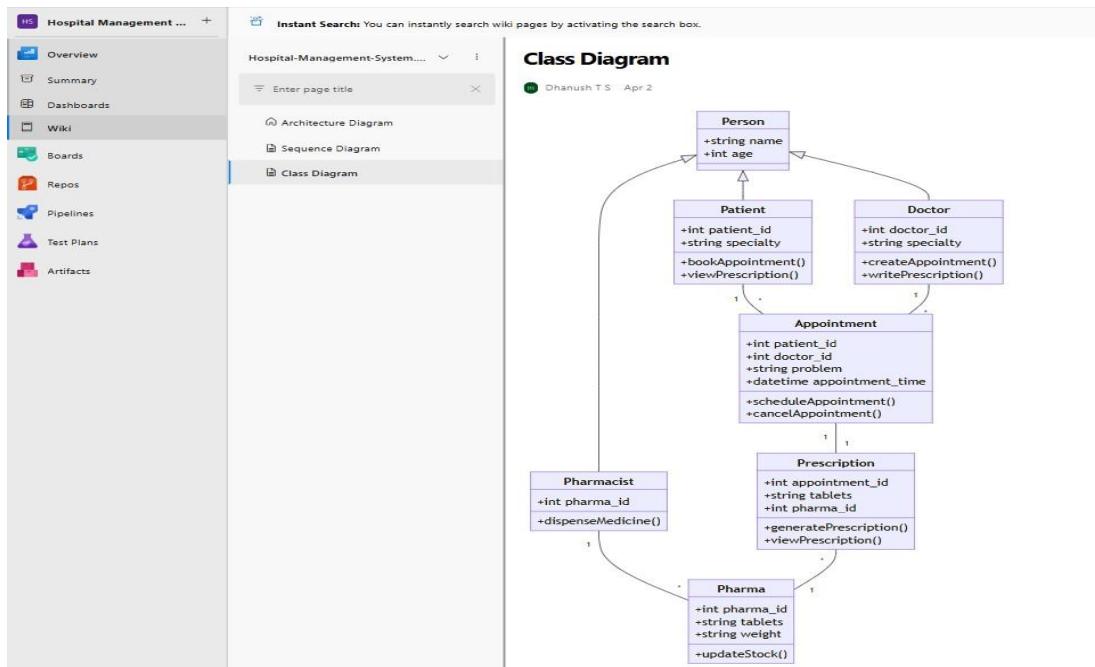
EXP NO: 6

DESIGNING CLASS AND SEQUENCE DIAGRAMS FOR PROJECT ARCHITECTURE

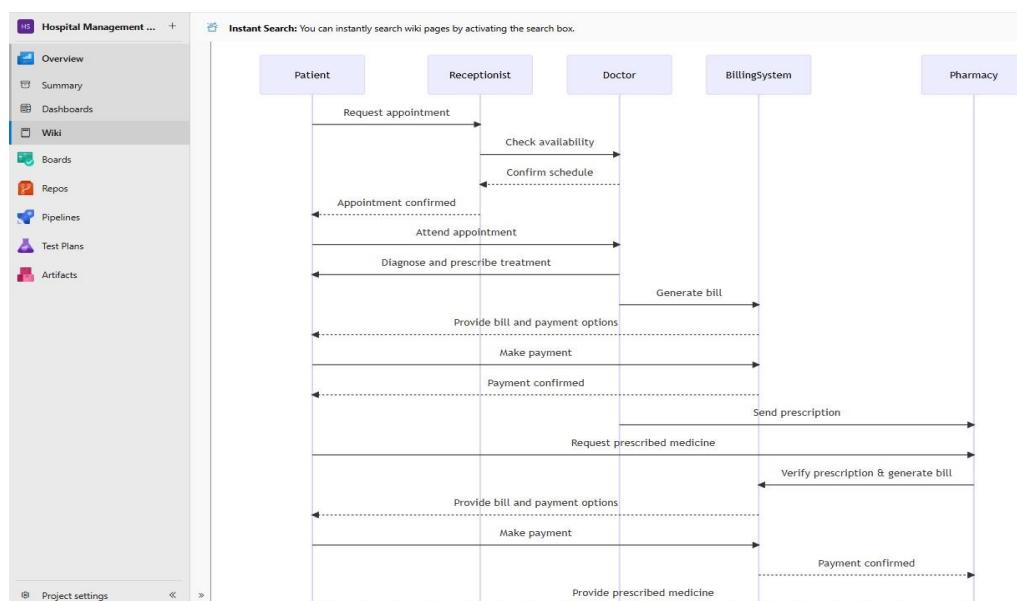
Aim:

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.

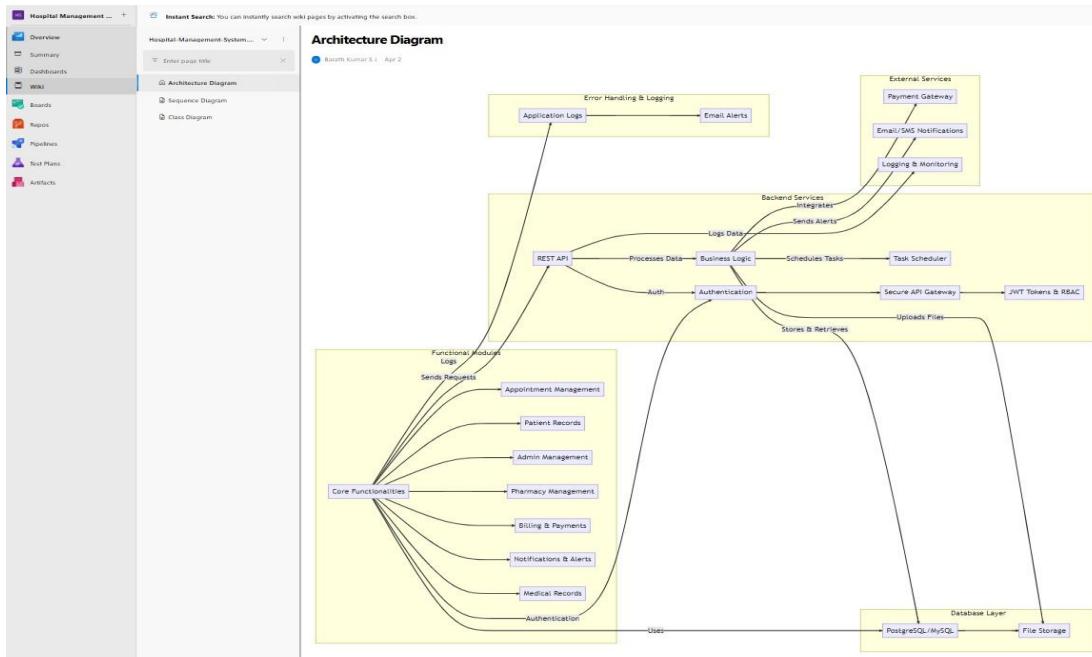
EXP NO: 7

DESIGNING ARCHITECTURAL AND ER DIAGRAMS FOR PROJECT STRUCTURE

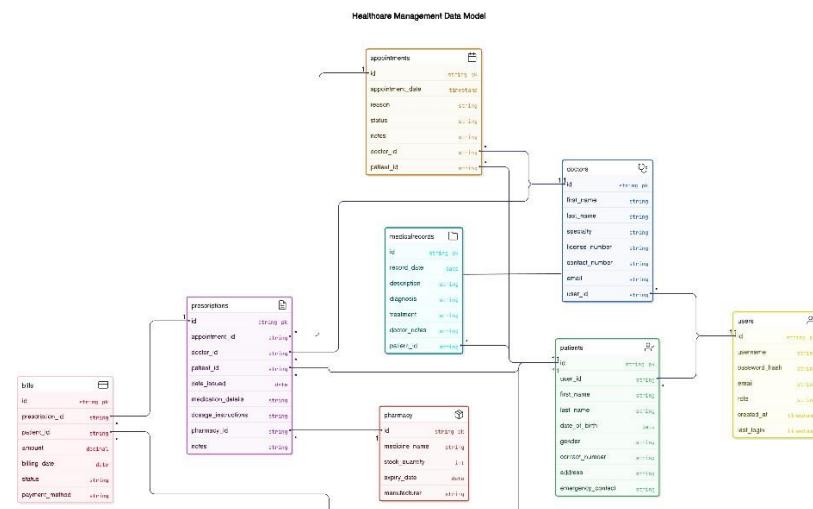
Aim:

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.

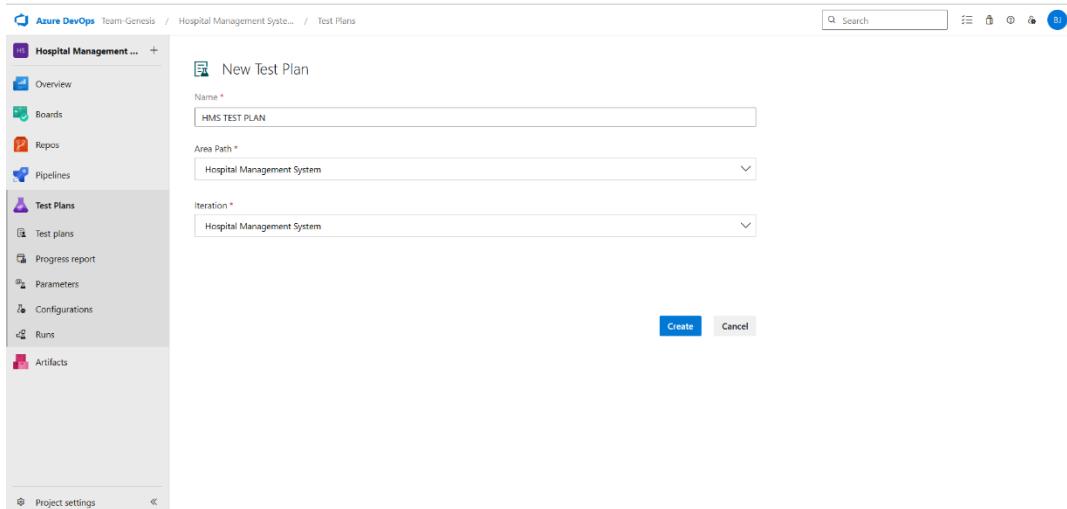
7. Separate Test Suites

- 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 (ID: 81)' list in Azure DevOps. The left sidebar shows the 'Test plans' section is selected. The main area displays a table of test cases under the 'Patient' suite. The table has columns for 'Title', 'Order', 'Tags', and 'State'. Two entries are listed: 'TC01-Book Appointment (HP)' and 'TC02-Book Appointment (ES)'. A 'New Test Case' button is visible at the top right of the table.

| 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 has a priority of 2. It includes a summary of steps and expected results:

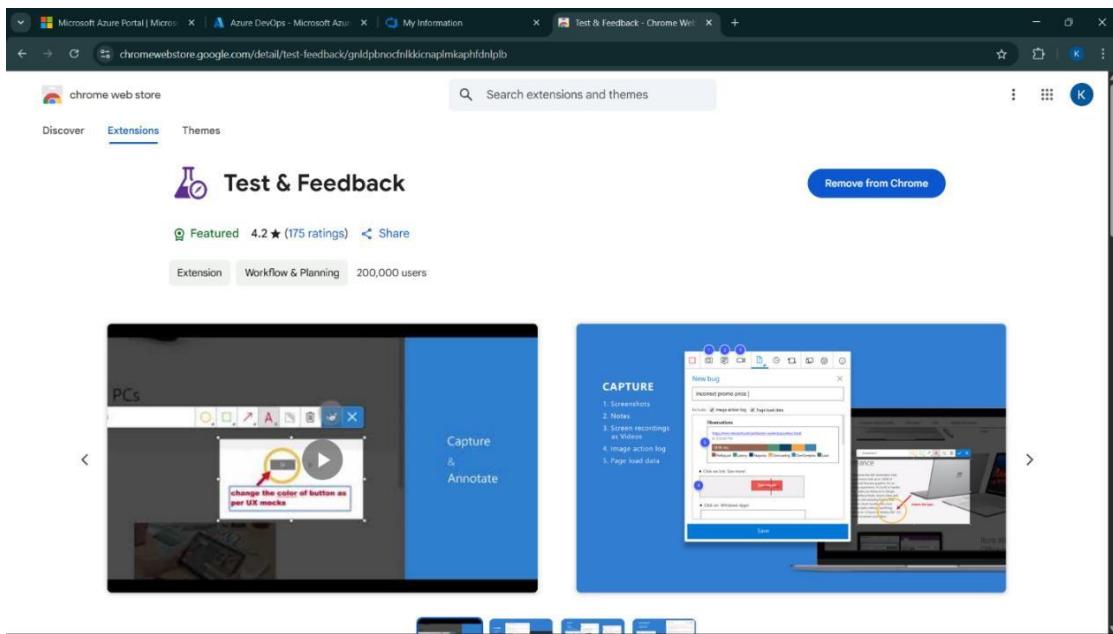
| Steps | Action | Expected result |
|-------|----------------------------|---|
| 1. | Disconnect from internet | Network is offline |
| 2. | Navigate to "My Playlists" | Error message "Unable to load playlists" is shown |

Below the test case details, there is a "Status" section indicating the status as "Not Automated". The bottom of the screen shows project settings and navigation controls.

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 the extension's interface with a toolbar and a capture tool. Another screenshot shows the extension's bug reporting feature. A large "Add to Chrome" button is prominently displayed.



Test and feedback

Showing it as an extension

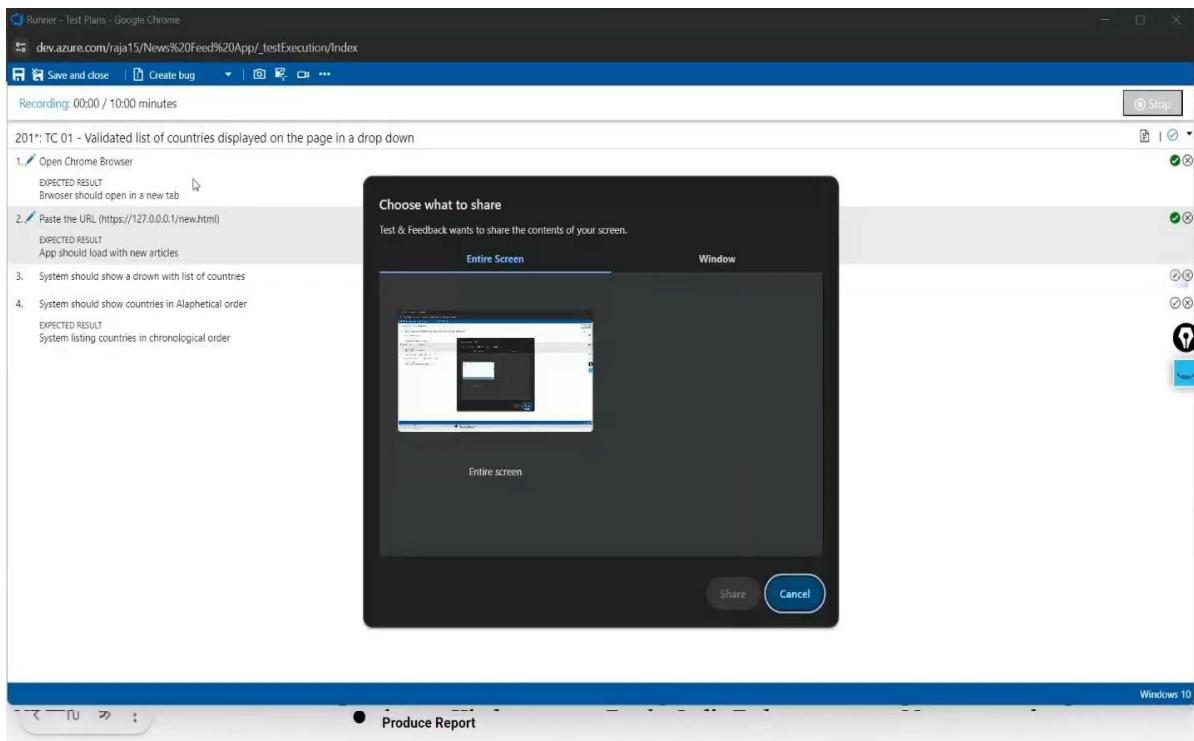
A screenshot of a Microsoft Edge browser window showing the Azure DevOps Test Plan interface. The URL is dev.azure.com/231801095/Music%20Playlist%20Batch%20Creator/_testPlans/define?planId=84&suitId=86. On the left, the navigation menu is open, showing "Test Plans" selected. The main area displays a test plan for "Music Playlist Batch Creator". A modal dialog box is open in the center, titled "Extensions". The "Full access" section lists several extensions: "Copy Text from Picture", "Dark Reader", "Monica: ChatGPT AI Assist...", "Selectext: Copy text from V...", and "Test & Feedback". At the bottom of the modal is a "Manage extensions" button.

5. Running the test cases

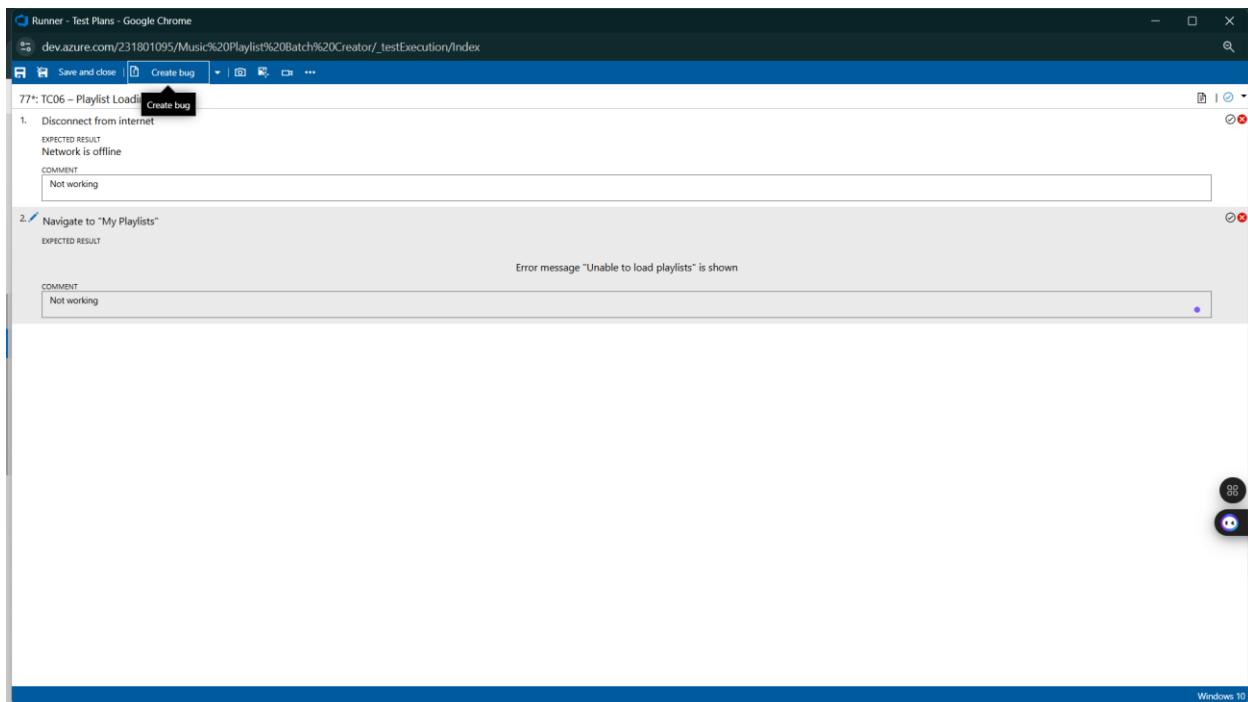
The screenshot shows the Azure DevOps Test Plan interface. On the left, the navigation bar includes 'Overview', 'Boards', 'Repos', 'Pipelines', 'Test Plans', 'Test plans' (selected), 'Progress report', 'Parameters', 'Configurations', 'Runs', and 'Artifacts'. Under 'Test plans', 'Music Playlist Batch 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, Order 1, Test Case Id 75, Windows 10, Tester Mallu karthick B...) and 'TC06 - Playlist Loading Failure' (Not run). A context menu is open over 'TC05 - View Playlist Page' with options: 'View execution history', 'Mark Outcome', 'Run', 'Reset test to active', 'Edit test case', 'Assign tester', and 'View test result'. The status bar at the bottom says 'Apr 10 - Apr 17 Past'.

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 says '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 Result: Playlist loading spinner keeps spinning indefinitely on poor network

2. Navigate to "My Playlists"
EXPECTED: New
Actual Result: 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: **NEW BUG**
Story Points: 1
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

System Info
Tested By: 77 TC06 – Playlist Loading Failure
Updated 10-04-2025, ● Design
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

8. Test case results

The screenshot shows the Azure DevOps interface for a test plan. On the left, the navigation bar includes 'Overview', 'Boards', 'Repos', 'Pipelines', 'Test Plans', 'Test plans', 'Progress report', 'Parameters', 'Configurations', 'Runs', and 'Artifacts'. The 'Test plans' section is currently selected. In the center, a test suite named 'TS02 - View Playlists (ID: 87)' is displayed under 'Music Playlist Batch Creator - T...'. It contains two test points: 'TC05 - View Playlist Page' (selected) and 'TC06 - Playlist Loading Failure'. A modal window titled 'TC05 - View Playlist Page' is open, showing the 'Test Case Results' table:

| Outcome | TimeStamp | Configuration | Run by | Tester | Test Pl. |
|----------------|-----------|---------------|--------------------------|--------------------------|----------|
| 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 | Karthick S | 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 work items. The left sidebar includes 'Overview', 'Boards', 'Work items', 'Backlogs', 'Sprints', 'Queries', 'Delivery Plans', 'Analytics views', 'Repos', 'Pipelines', 'Test Plans', and 'Artifacts'. The 'Work items' section is selected. A specific bug report for 'Bug 203 - Countries Drop down Not Available on the page' is displayed. The report details:

- State:** New
- Reason:** New
- Repro Step:** Active
- Details:** 5/19/2024 AM
- 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
Expected Result: Success! Should show a dropdown with list of countries
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

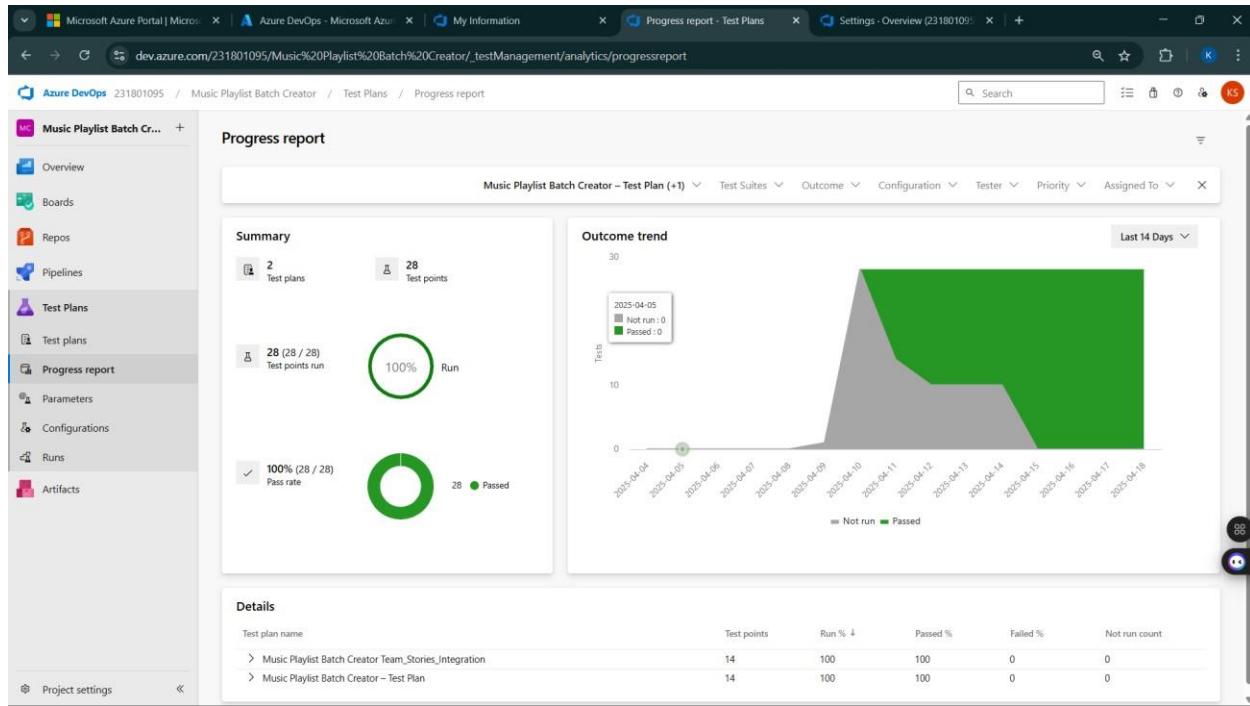
- Assigning bug to the developer and changing state

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 a description of the issue ('Disconnect from internet'), expected results ('Network is offline'), and comments ('Comments: Page Not loading'). The step is linked to a test configuration for 'Windows 10'. To the right, there are sections for 'Planning' (Priority 2, Severity 3 - Medium), 'Deployment' (with a note about tracking releases), 'Development' (linking to an Azure Repos branch), and 'Related Work' (a linked work item). A 'System Info' section at the bottom right provides deployment details.

10. Progress report

The screenshot displays a 'Progress report' for the 'HMS Test Plan'. The left sidebar shows the 'Test Plans' section is selected. The main area features a summary card with metrics: 1 Test plan, 8 Test points, 8 (3 / 8) Test points run (100% Run), and 100% (8 / 8) Pass rate. Below this is a 'Outcome trend' chart showing the number of tests over time, with a sharp increase starting around April 14th. The chart includes a legend for 'Not run' (grey) and 'Passed' (green). At the bottom, a 'Details' table provides a breakdown of the test plan's components.

| Test plan name | Test points | Run % | Passed % | Failed % | Not run count |
|----------------|-------------|-------|----------|----------|---------------|
| HMS Test Plan | 8 | 100 | 100 | 0 | 0 |



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 'Basic' template is also visible.

| 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 process named '231801095 Agile (default)' has been added under the 'Agile' template, indicated by a highlighted row. The 'Basic' template is also visible.

| 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 a tab bar with 'Definition' selected, showing the option to 'Create a field'. A field named 'Acceptance Criteria' is being defined as a 'Text (single line)' type. The background shows the 'Process' section of the settings, with 'Steps' selected under 'Layout'.

The screenshot shows the 'Work-item types' page under the 'Process' section of the Azure DevOps settings. A new work-item type named 'Music Playlist Batch Creator' is listed, with a detailed description provided. The 'Projects' tab is selected at the top of the list.

The screenshot shows the Azure DevOps Settings - Process page for a specific test case. The URL in the browser is dev.azure.com/231801095/_settings/process?type-id=231801095Agile.TestCase&process-name=231801095%20Agile&a=layout. The left sidebar shows 'Organization Settings' for project 231801095, with 'Process' selected. The main content area displays the 'Test Case' configuration, specifically the 'Steps' section. The 'Steps' field is labeled 'Text (multiple lines)' and contains the placeholder 'Steps'. To the right of the steps, there are several sections: 'Recent test results', 'Deployment', 'Development', 'Related Work', and 'Status'. Each section has a brief description and a link to its details. A vertical scroll bar is visible on the right side of the main content area.

Result:

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

| | |
|------------------|-----------------------------------|
| EXP NO: 9 | LOAD TESTING AND PIPELINES |
|------------------|-----------------------------------|

Aim:

To create an Azure Load Testing resource and run a load test to evaluate the performance of a target endpoint and to create and demonstrate an Azure DevOps pipeline for automating application builds, tests, and deployment.

Load Testing**Azure Load Testing:**

Azure Load Testing allows you to simulate high traffic and stress tests for your web applications and APIs to understand how they perform under load. It helps identify performance bottlenecks, scalability issues, and optimize resource usage before deployment.

Steps to Create an Azure Load Testing Resource:

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

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

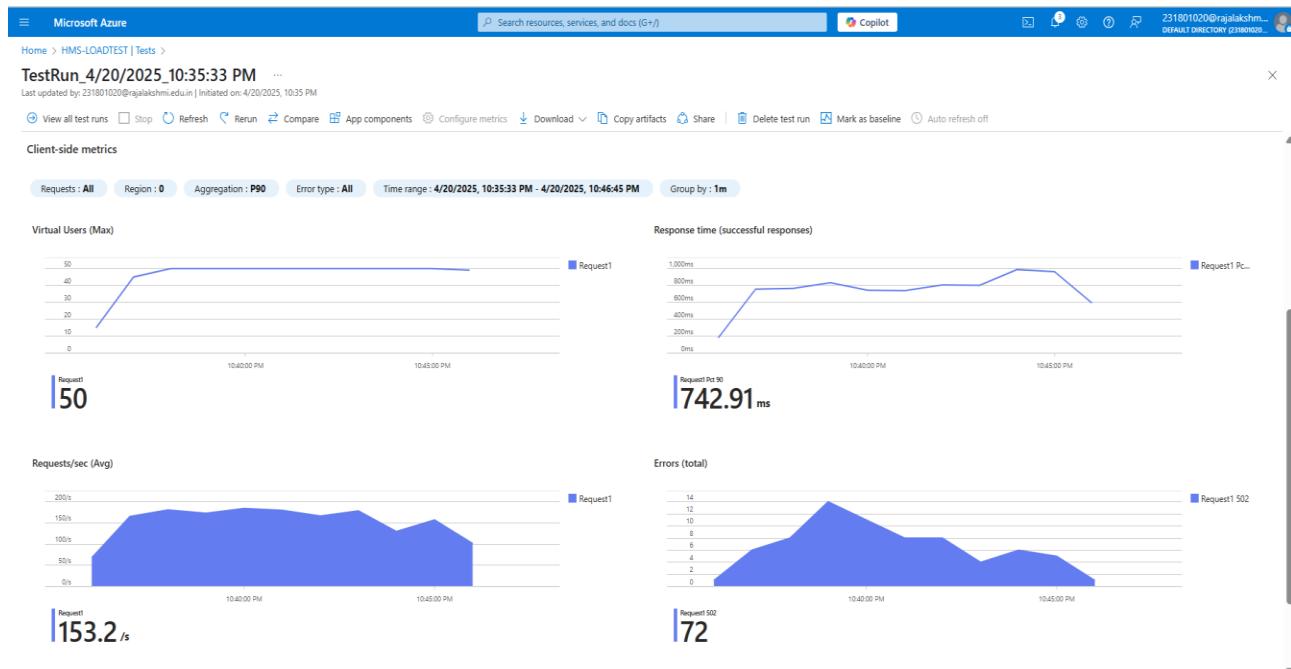
Test run details

| 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

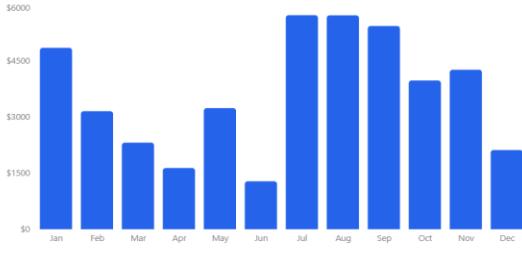
Total Revenue
\$45,231.89
+20.1% from last month

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 | Amount |
|----------|-----------------|---------------------------|-------------|
| 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. On the left, there's a sidebar with project navigation links: Overview, Boards, Repos, Pipelines (which is selected and highlighted in blue), Environments, Library, Test Plans, and Artifacts. Below the sidebar, there are 'Project settings' and a back arrow icon.

The main area is titled 'Pipelines' and shows a list of recently run pipelines. The first item in the list is 'dhanush-ts.Hospital-Management-Syst...' with a green checkmark icon. To its right, it says '#20250428.1 • Update Header.js' and 'Manually triggered for main'. At the bottom right of this card, there are two small icons: one for 'Yesterday' and another for '<1s'.

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' owned by 'dhanush-ts'. The repository has 1 branch and 0 tags. The main branch has 20 commits. The commit history includes updates to 'Header.js', 'app', 'components', 'hooks', 'lib', and 'public' folders, as well as initial commits for '.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'.

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.