

EXP NO: 01

AZURE DEVOPS ENVIRONMENT SETUP

Aim:

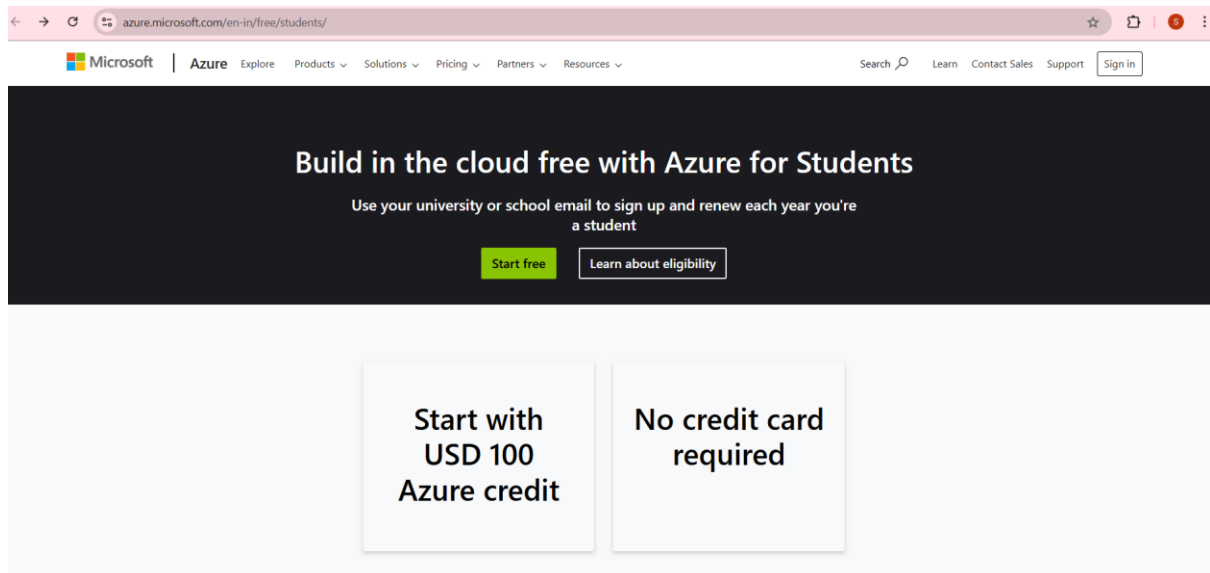
To set up and access the Azure DevOps environment by creating an organization through the Azure portal.

INSTALLATION:

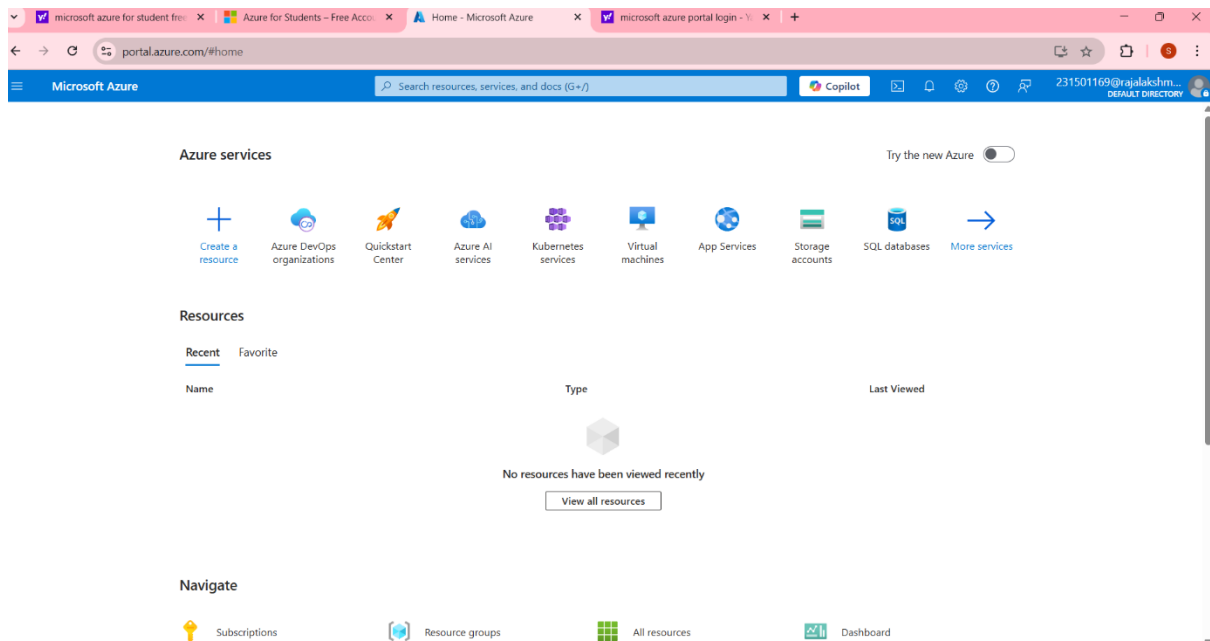
1. Open your web browser and go to the Azure website:
<https://azure.microsoft.com/en-us/getstarted/azure-portal>.

Sign in using your Microsoft account credentials.

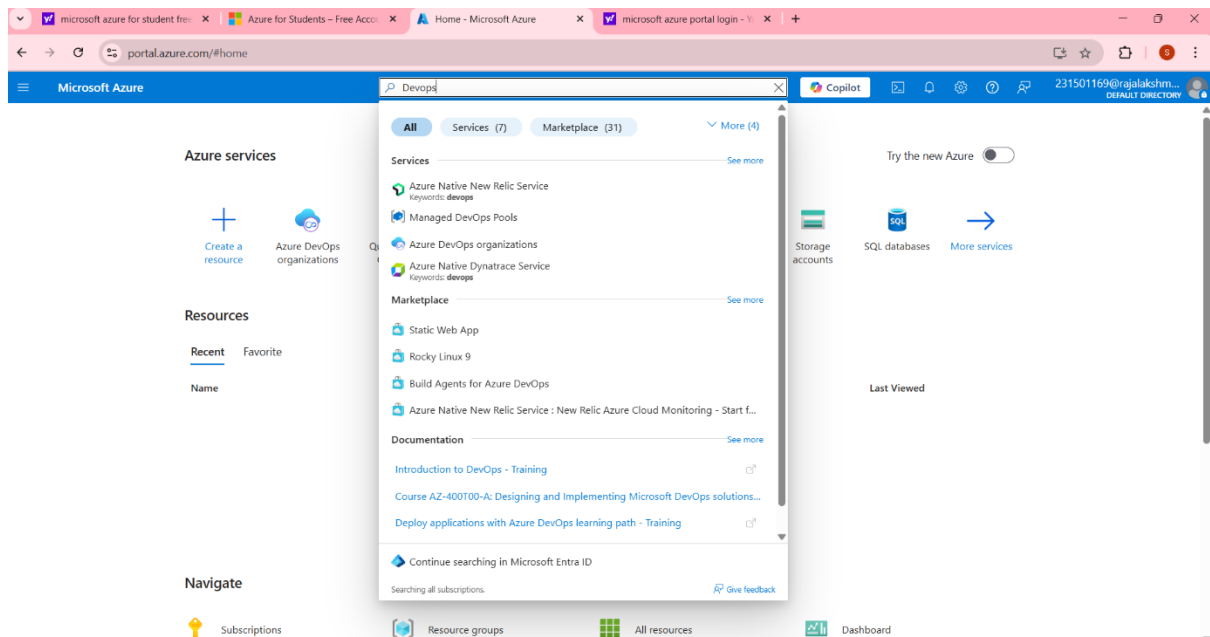
If you don't have a Microsoft account, you can create one here:
<https://signup.live.com/?lic=1>



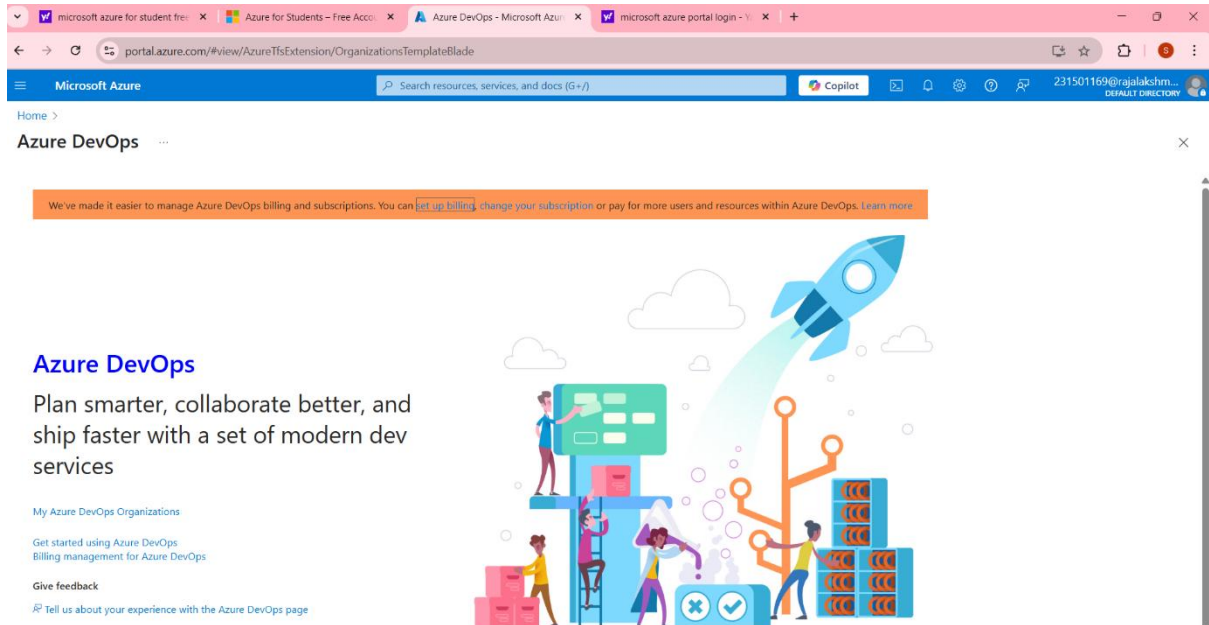
2. Azure home page



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



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



Result:

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

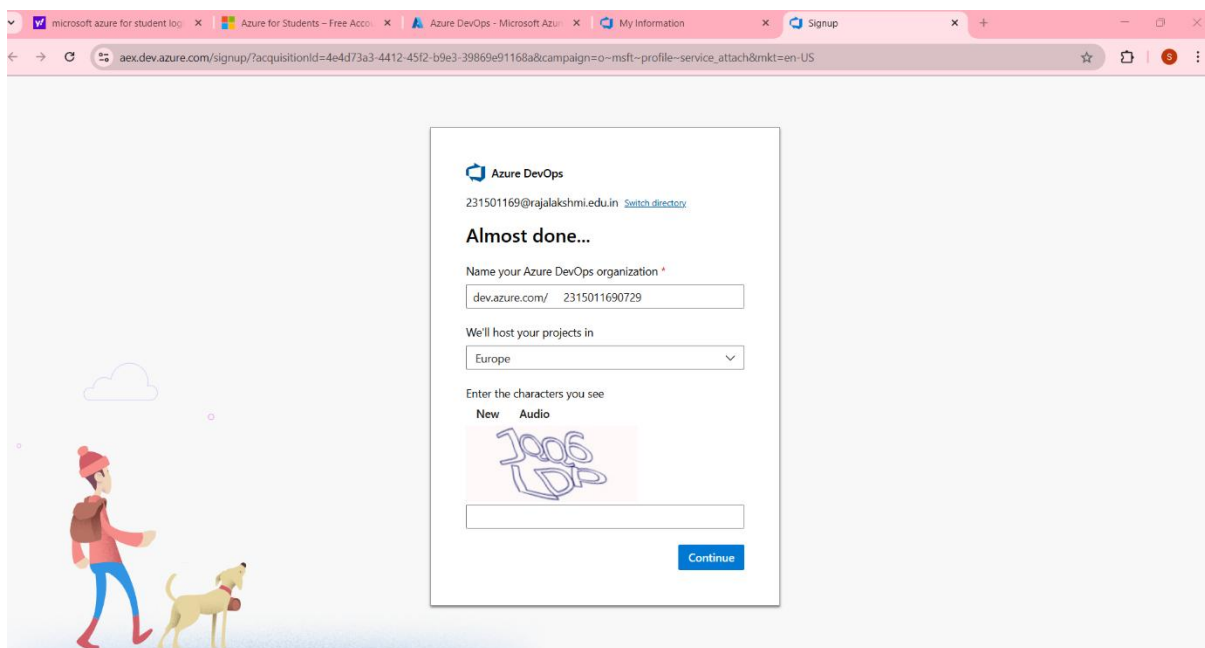
EXP NO: 02

AZURE DEVOPS PROJECT SETUP AND USER STORY MANAGEMENT

Aim:

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

1.Create An Azure Account



2.Create the First Project in Your Organization

- After the organization is set up, you'll need to create your first **project**. This is where you'll begin to manage code, pipelines, work items, and more.
- On the organization's **Home page**, click on the **New Project** button.
- Enter the project name, description, and visibility options:

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

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

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

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

Create new project ×

Project name ^{*}

HospitalManagement

Description

Visibility

☐ Public
Anyone on the internet can view the project. Certain features like TFVC are not supported.

☒ Private
Only people you give access to will be able to view this project.

Public projects are disabled for your organization. You can turn on public visibility with [organization policies](#).

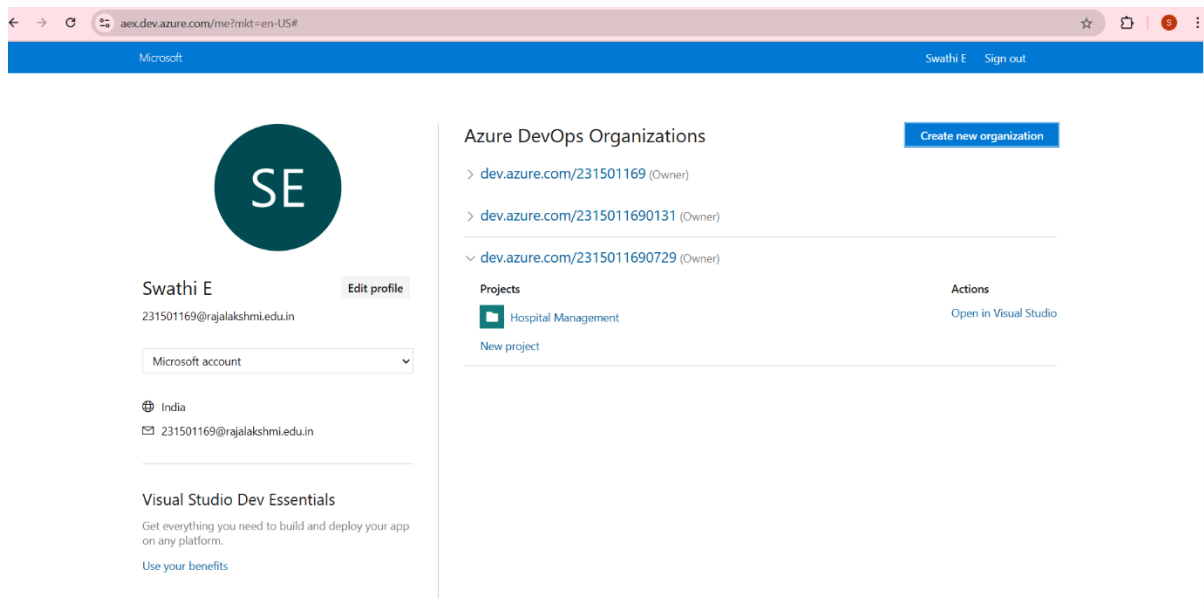
^ Advanced

Version control [?]
Git

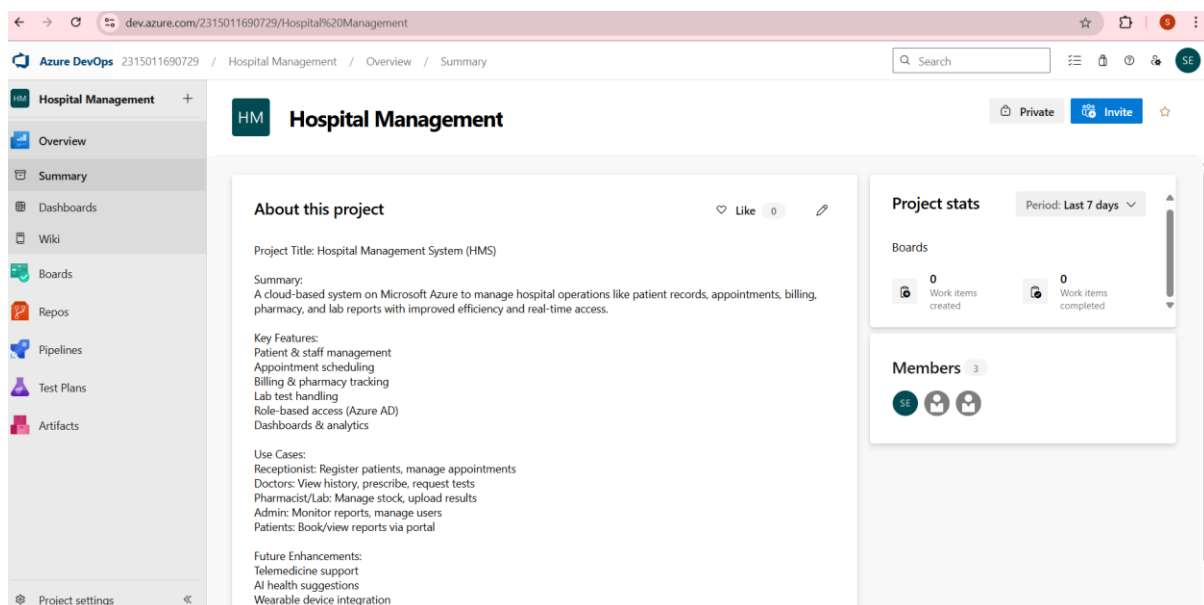
Work item process [?]
Agile

Cancel Create

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

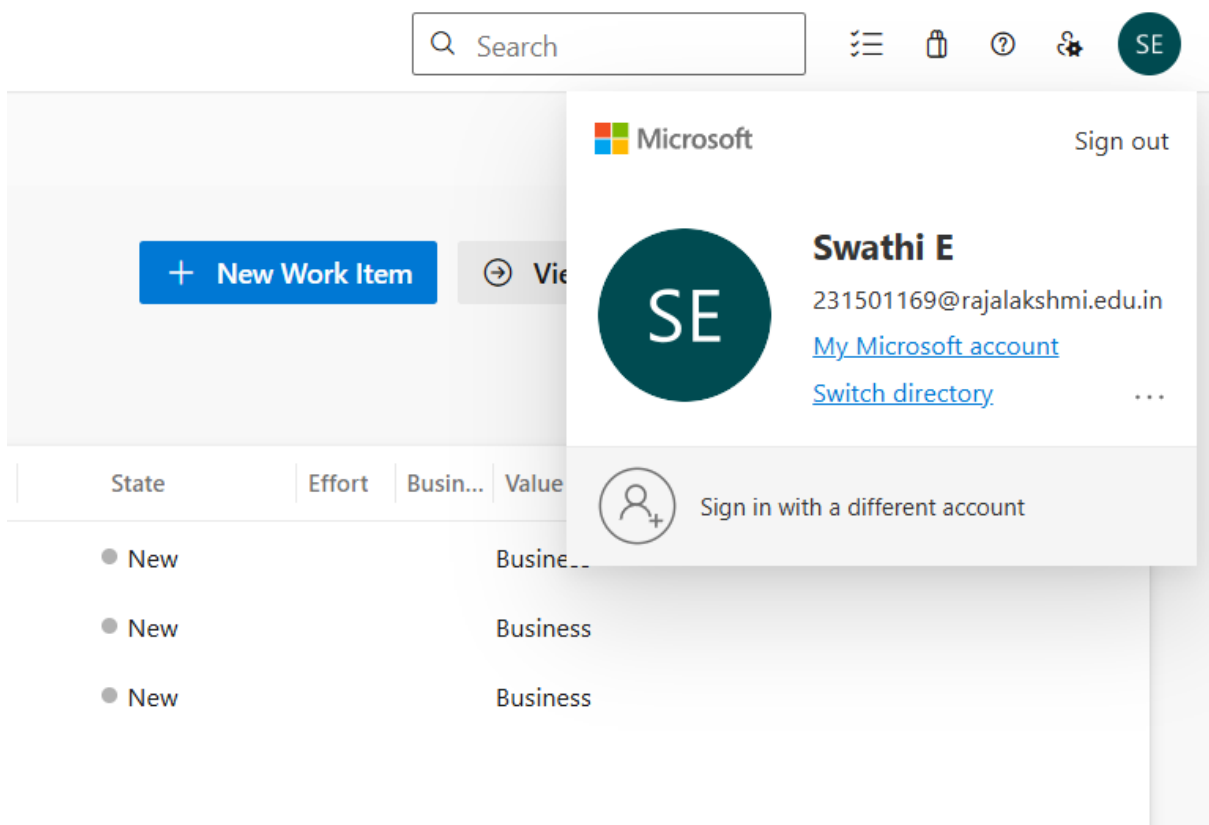
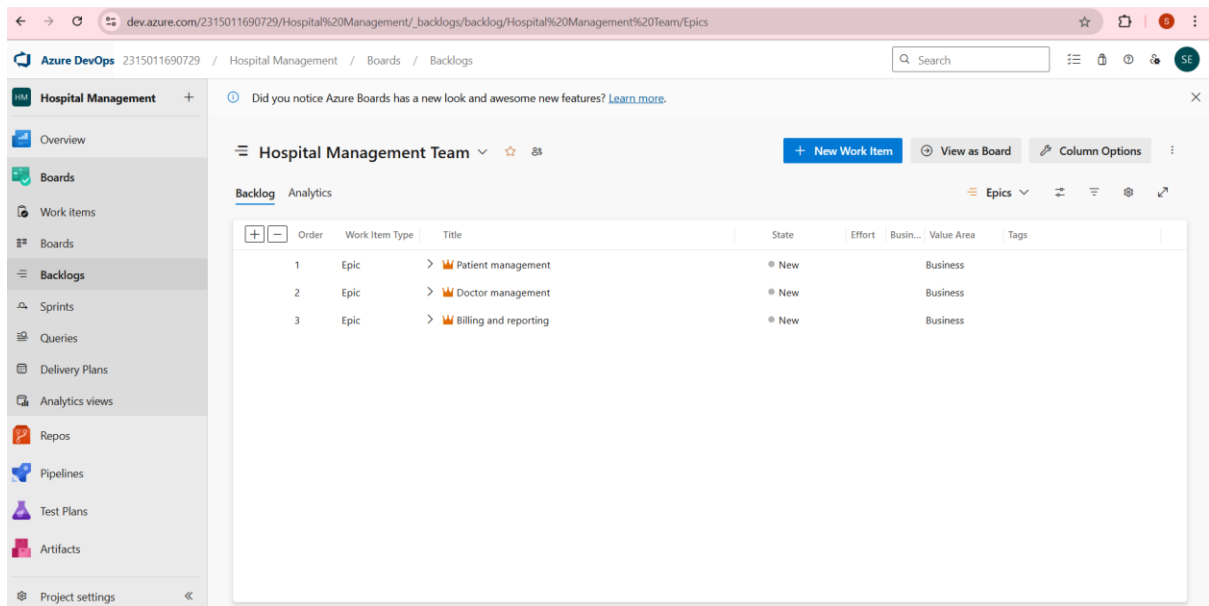


4. Project dashboard



5. To manage user stories:

- 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.
- On the work items page, you'll see the option to Add a work item at the top. Alternatively, you can find a + button or Add New Work Item depending on the view you're in. From the Add a work item dropdown, select User Story. This will open a form to enter details for the new User Story.



Result:

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

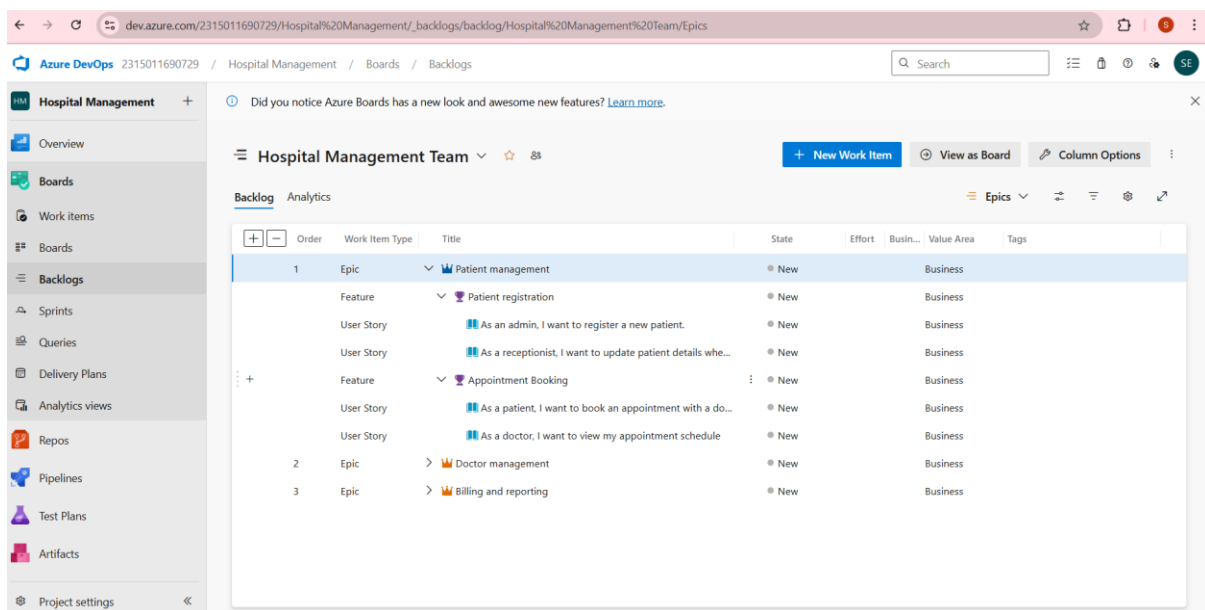
EXP NO:03

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

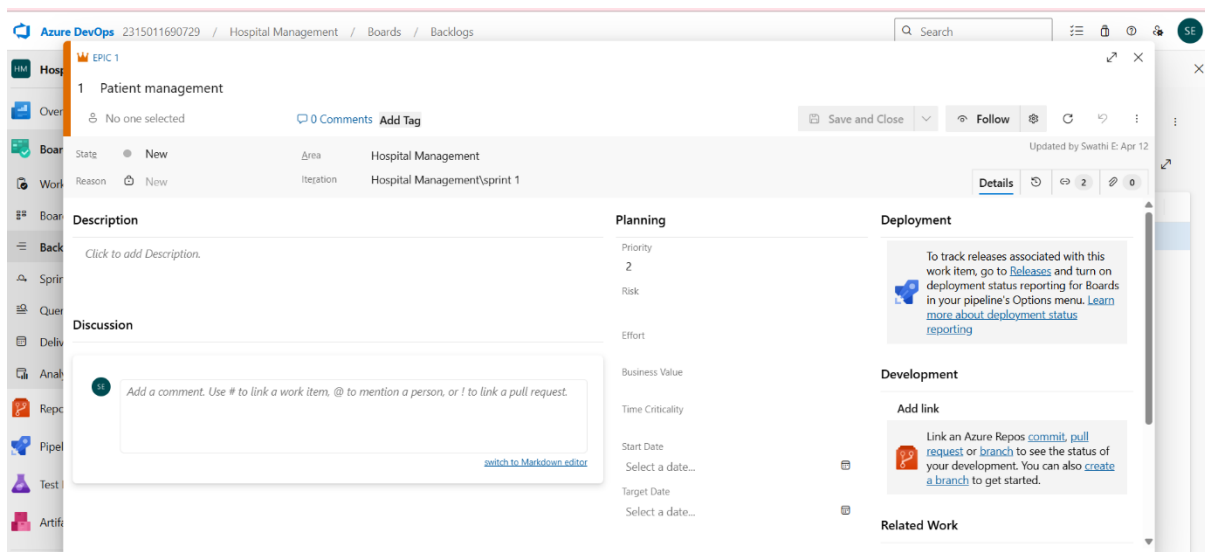
Aim:

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

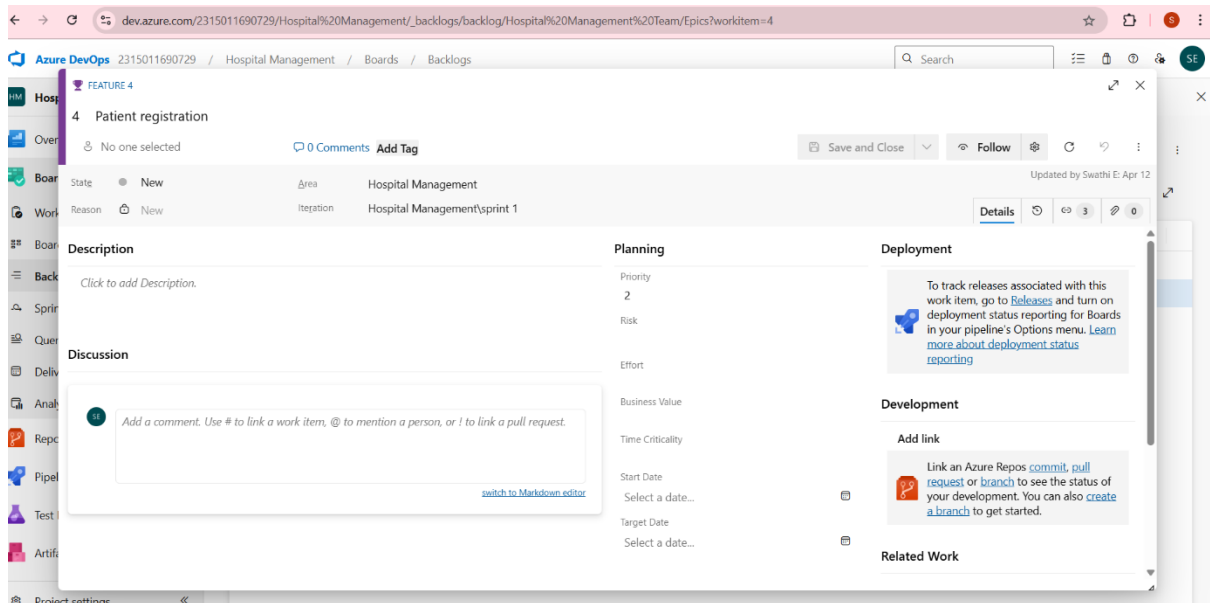
Create Epic, Features, User Stories, Task



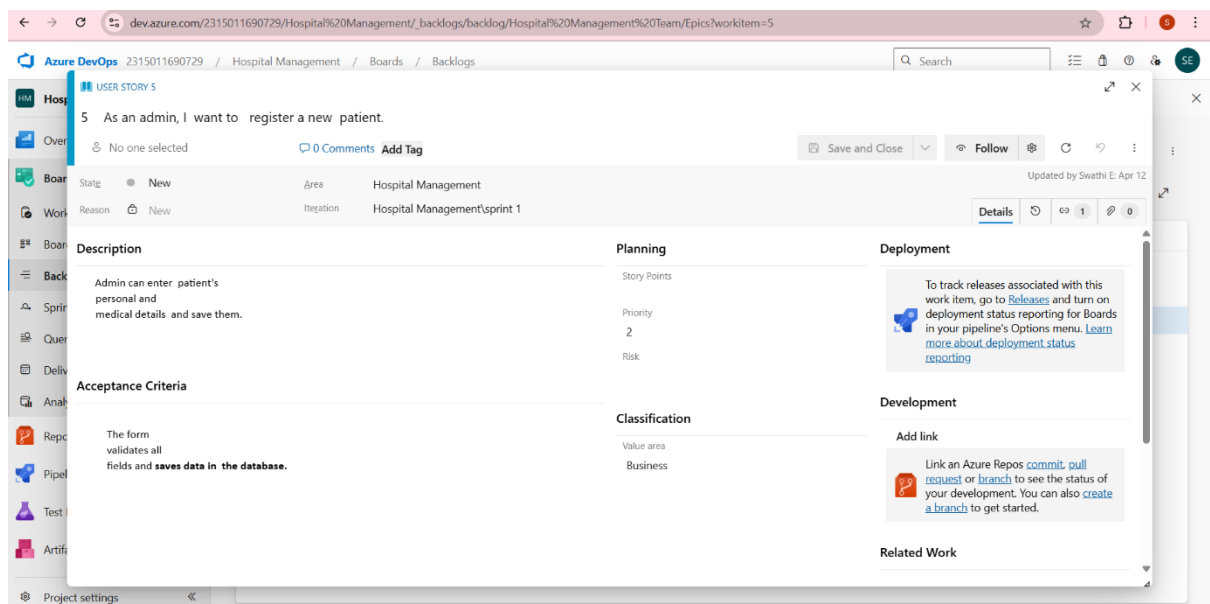
1.Fill in Epics



2.Fill in Features



3.Fill in User Story Details



Result:

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

EXP NO: 04

SPRINT PLANNING

Aim:

To assign user story to specific sprint for the Hospital Management Project.

Sprint Planning

Sprint 1

The screenshot shows the Azure DevOps Sprints board for the 'Hospital Management Team'. The board is titled 'sprint 1' and shows a timeline from April 5 to April 19, 15 work days. The board is divided into four columns: New, Active, Resolved, and Closed. There are three user stories in the 'New' column, all assigned to 'Unassigned'.

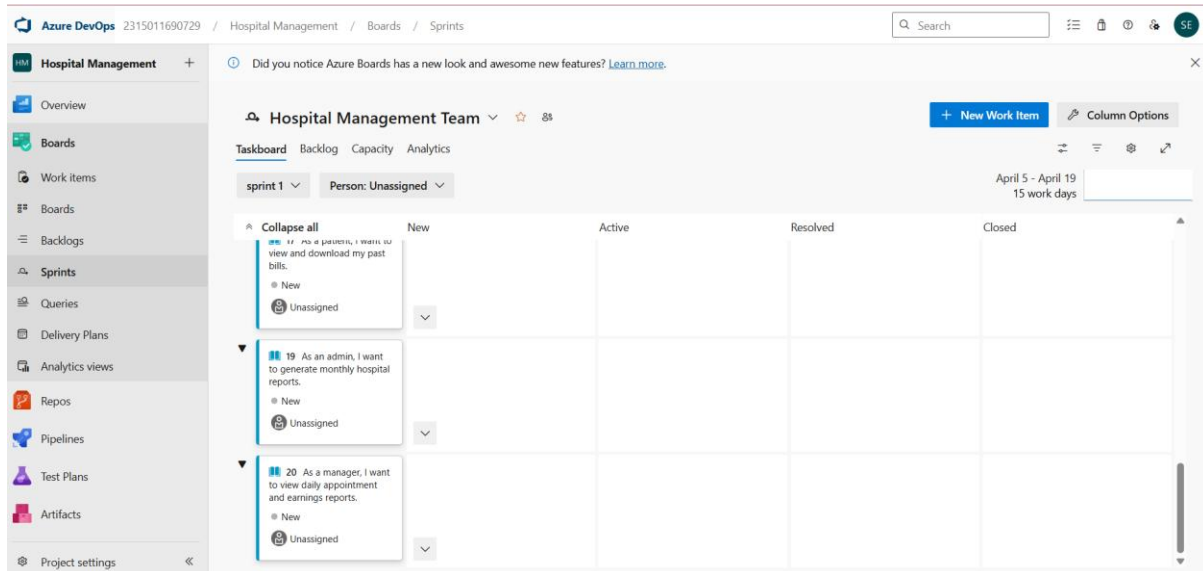
Story ID	Story Description	Assignee	Status
5	As an admin, I want to register a new patient.	Unassigned	New
6	As a receptionist, I want to update patient details when needed.	Unassigned	New
8	As a patient, I want to book an appointment with a doctor.	Unassigned	New

Sprint 2

The screenshot shows the Azure DevOps Sprints board for the 'Hospital Management Team'. The board is titled 'sprint 1' and shows a timeline from April 5 to April 19, 15 work days. The board is divided into four columns: New, Active, Resolved, and Closed. There are three user stories in the 'New' column, all assigned to 'Unassigned'.

Story ID	Story Description	Assignee	Status
9	As a doctor, I want to view my appointment schedule	Unassigned	New
11	As an admin, I want to add new doctors to the system	Unassigned	New
12	As an admin, I want to update or delete doctor profiles.	Unassigned	New

Sprint 3



Result:

The Sprints are created for the Music Playlist Batch Creator Project.

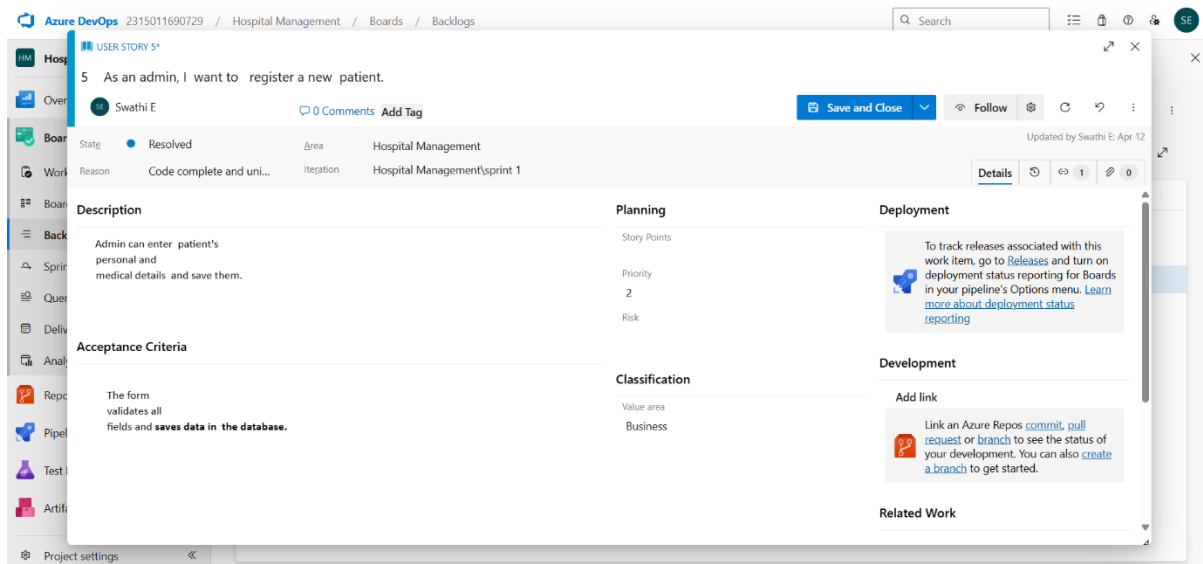
EXP NO:05

POKER ESTIMATION

Aim:

Create Poker Estimation for the user stories - Hospital Management System Project

Poker Estimation



Result:

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

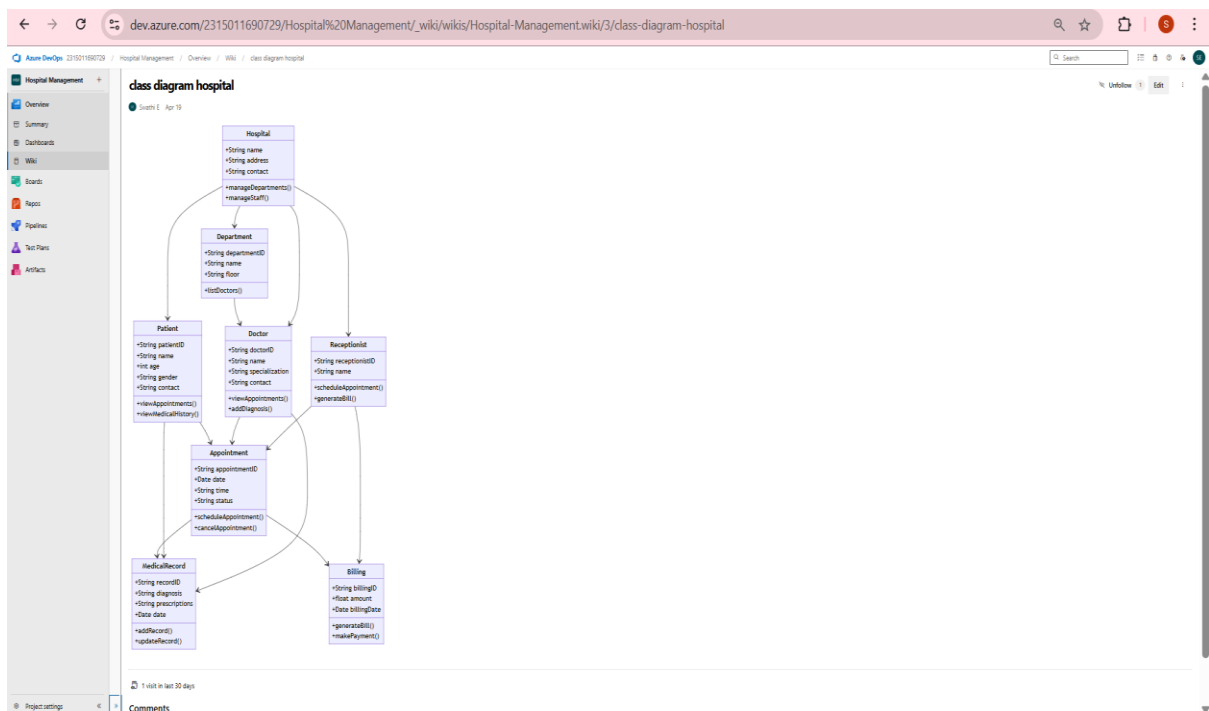
EXP NO:06

DESIGNING CLASS AND SEQUENCE DIAGRAMS FOR PROJECT ARCHITECTURE

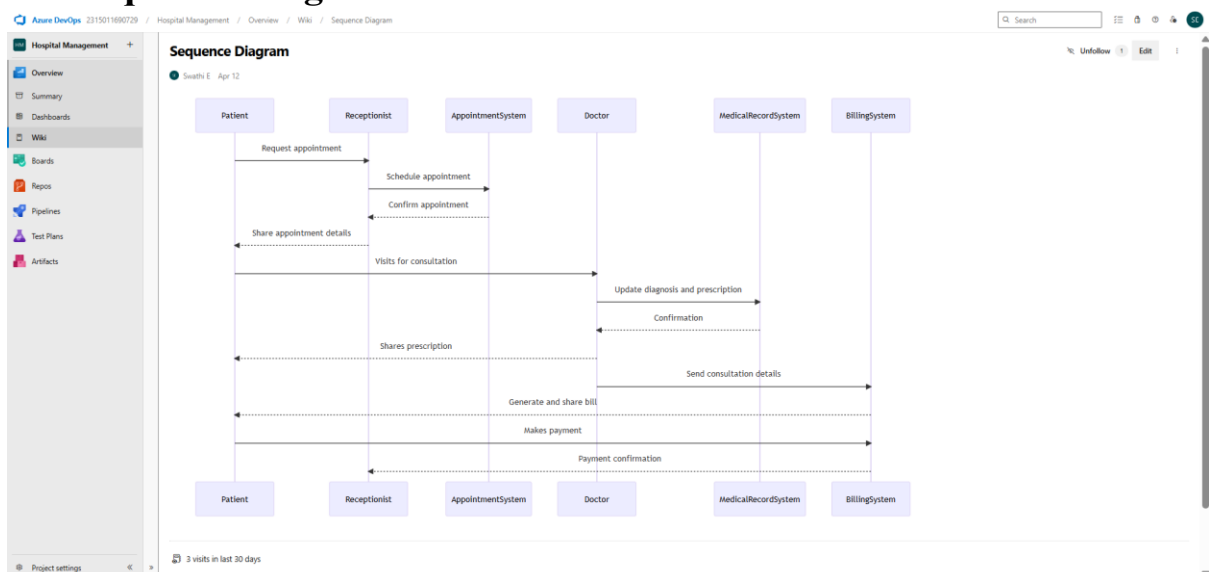
Aim:

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 Project.

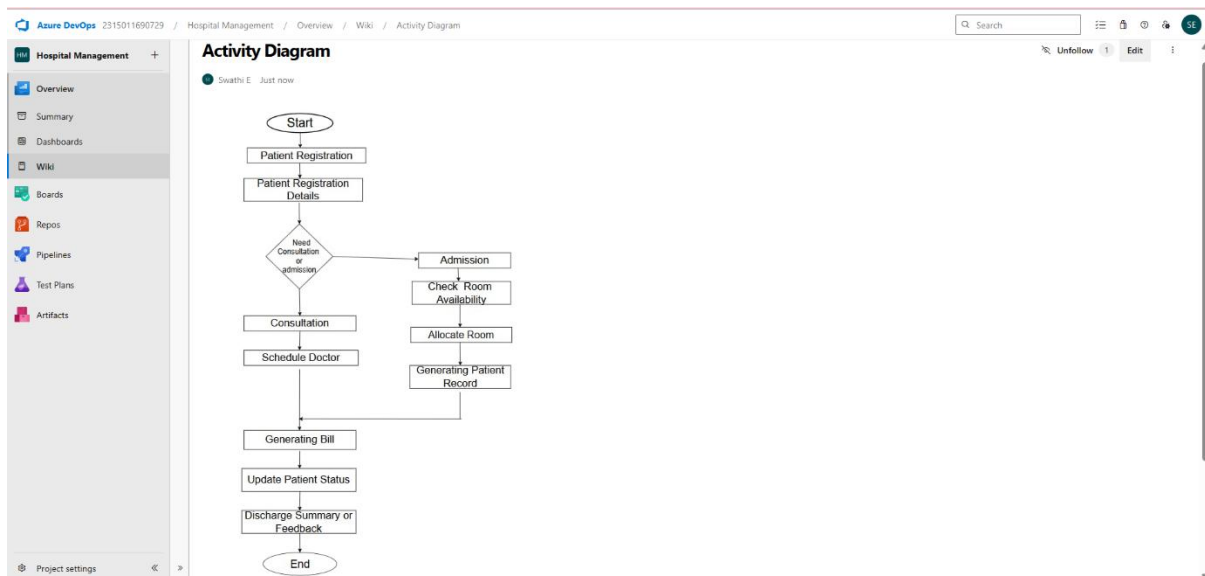
EXP NO: 7

DESIGNING ARCHITECTURAL AND ER DIAGRAMS FOR PROJECT STRUCTURE

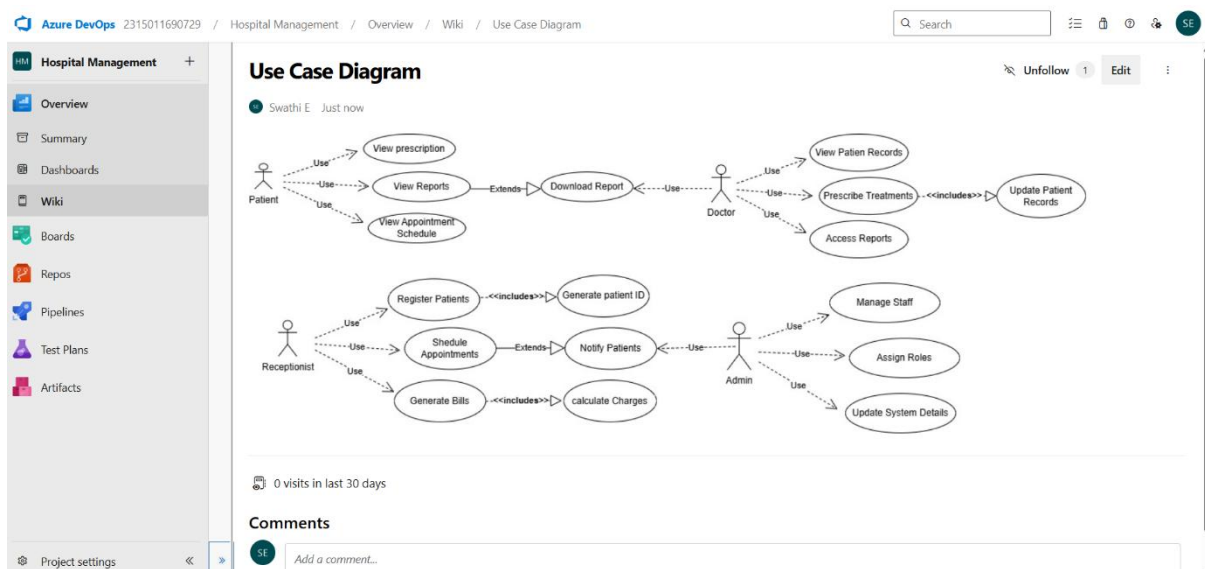
Aim:

To Design an use case diagrams and activity diagram for the given Project.

7A. Activity Diagram



7B. Use case Diagram



Result:

The use case diagrams and activity diagram is designed Successfully for the Hospital Management Project.

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.

DevOps platform**Test Planning and Test Case****Test Case Design Procedure****1.Understand Core Features of the Application Patient Registration**

- Appointment Booking
- Billing and Report Generation
- Doctor Availability Management
- Prescription Management

2.Define User Interactions

- Simulate real scenarios (e.g., register patient, book appointment, generate bill, update availability).

3.Design Happy Path Test Cases

- Validate expected flows (e.g., successful patient registration, successful booking).

4.Design Error Path Test Cases

- Simulate invalid inputs or system limitations (e.g., missing fields, double booking, unavailable doctor).

5.Break Down Steps and Expected Results

- Each test case includes step-by-step actions and clearly defined expected outcomes.

6.Use Clear Naming and IDs

- Example: TC_PM_001 – Register New Patient Successfully

- Naming is consistent with Azure DevOps standards.

7. Separate Test Suites

- Suites are modular: Registration, Appointment, Billing, Availability, Prescription.

8. Prioritize and Review

- Happy path scenarios marked High Priority
- Test cases mapped to linked user stories in Azure DevOps.

1. New test plan

dev.azure.com/2315011690729/Hospital%20Management/_testManagement/new

2315011690729 / Hospital Management / Test Plans

New Test Plan

Name *
Patient registration module

Area Path *
Hospital Management

Iteration *
Hospital Management\sprint 1 4/5/2025 - 4/19/2025

Create Cancel

2. Test suite

dev.azure.com/2315011690729/Hospital%20Management/_testPlans/execute?planId=21&suiteId=22

2315011690729 / Hospital Management / Test Plans / Patient registration module

Patient registration module (ID: 22)

Define Execute Chart

Test Suites (2 items)

Title	Outcome	Order	Test Case Id	Configuration	Tester
Register New patient successfully	Passed	2	24	Windows 10	Swathi E
Show Validation Error for Missing Required Fields	Failed	3	25	Windows 10	Swathi E

The screenshot shows the Azure DevOps Test Plans page for a project named 'Hospital Management'. The table lists three test plans, all in an 'Active' state, assigned to 'Swathi E'. The test plans are for 'Billing and Report Generation', 'Appointment Booking Module', and 'Patient registration module'.

Title	Test Plan ID	State	Area Path	Iteration	Assigned To
Billing and Report Generation	31	Active	Hospital Management	Hospital Management\sprint 1	Swathi E
Appointment Booking Module	27	Active	Hospital Management	Hospital Management\sprint 1	Swathi E
Patient registration module	21	Active	Hospital Management	Hospital Management\sprint 1	Swathi E

3. Test case

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

Hospital Management System – Test Plans

USER STORIES

- As an admin, I want to register a new patient.
- As a patient, I want to book an appointment with a doctor.
- As a cashier, I want to generate bills after consultation.

Test Suites

Test Suite: TS01 - Patient Registration Module

1. TC01 – Register New Patient Successfully ○

Action:

- Login as Admin to the Azure-hosted HMS portal.
- Navigate to the “Register New Patient” form.
- Fill in valid details: Name, Age, Gender, Contact, Medical History. □ Click "Save".

○ Expected Results:

□ A success message is shown: "Patient Registered Successfully".

○ Type: Happy Path

2. TC02 – Show Validation Error for Missing Required Fields.

- **Action:**
 - ▢ Open the registration form.
 - ▢ Leave required fields (e.g., Name, Contact) blank
 - ▢ Click "Save".
- **Expected Results:**
 - ▢ Form displays message: "Name and Contact are required."
- **Type:** Error Path

Test Suite: TS02 - Appointment Booking Module

1. TC03 – Book Appointment Successfully

- **Action:**
 - ▢ Login to the Azure-hosted patient portal.
 - ▢ Navigate to "Book Appointment".
 - ▢ Select Doctor, Date, and Available Time Slot.
 - ▢ Click "Confirm".
- **Expected Results:**
 - ▢ Confirmation message displayed: "Appointment Confirmed".
- **Type:** Happy Path

2. TC04– Prevent

Double Booking ○

Action:

- ▢ Patient A books Dr. Smith at 10:00 AM.
- ▢ Patient B selects the same doctor and time slot.
- ▢ Clicks "Book".
- **Expected Results:**
 - ▢ System returns error: "Time slot already booked."
- **Type:** Error Path

Test Suite: TS03 - Billing and Report Generation

1. TC05 – Generate Consultation Bill ○

Action:

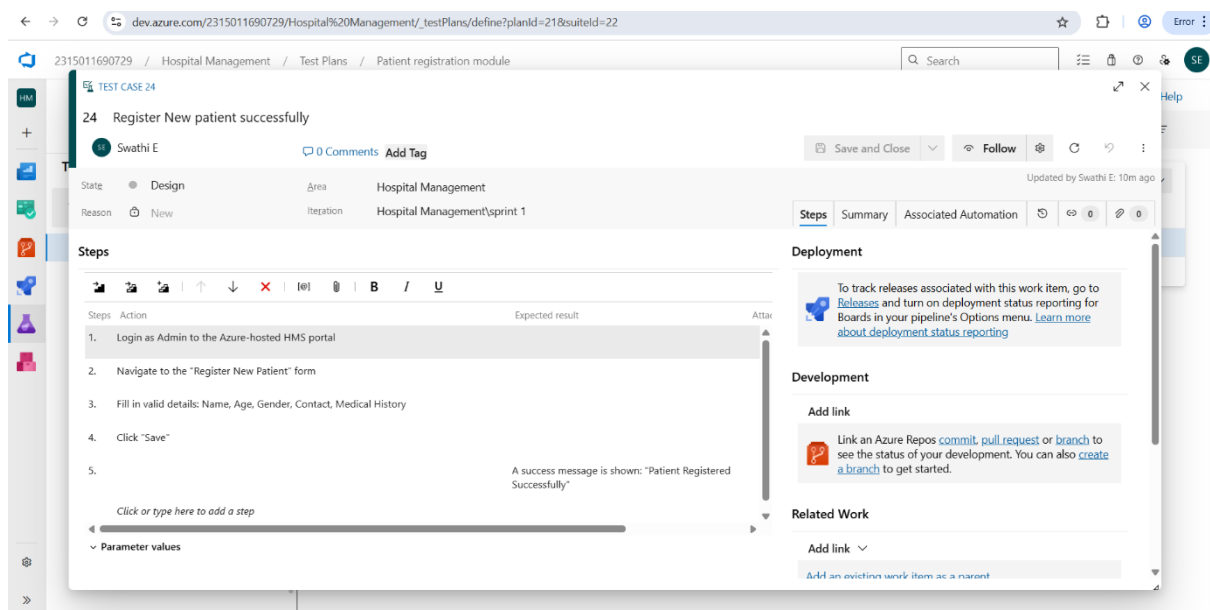
- Login as Cashier.
- Select a completed consultation
- Enter service fees, medicine charges, and lab test fees.

□ Click "Generate Bill" ○ **Expected Results:**

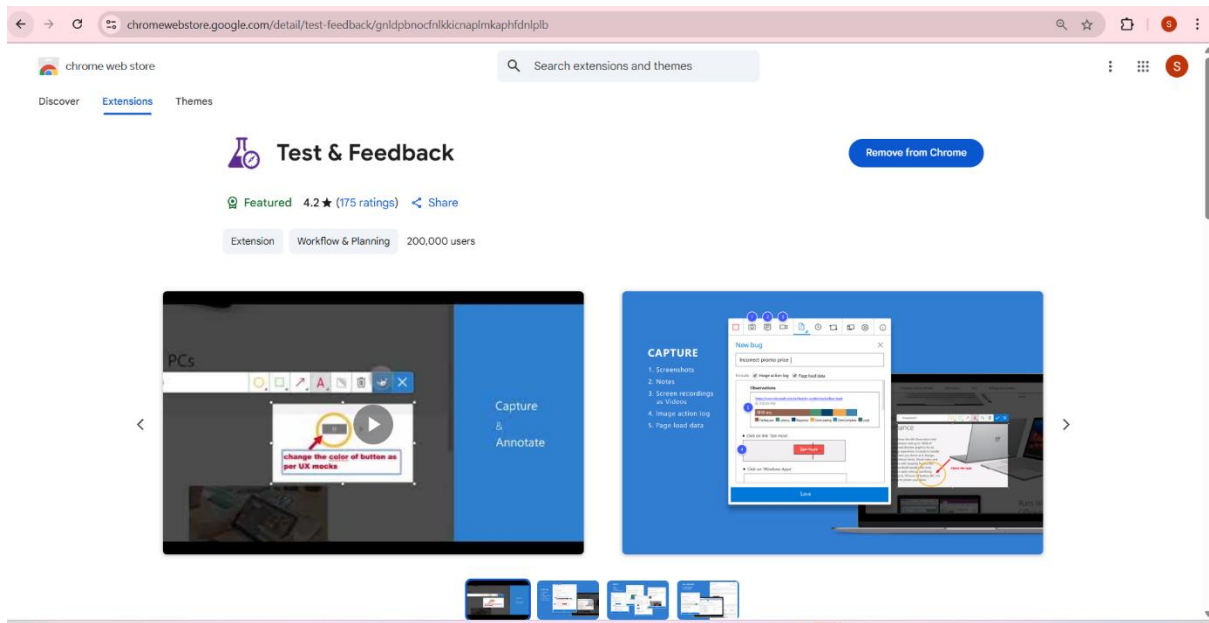
- Confirmation message and downloadable PDF bill are displayed.

○ **Type:** Happy Path

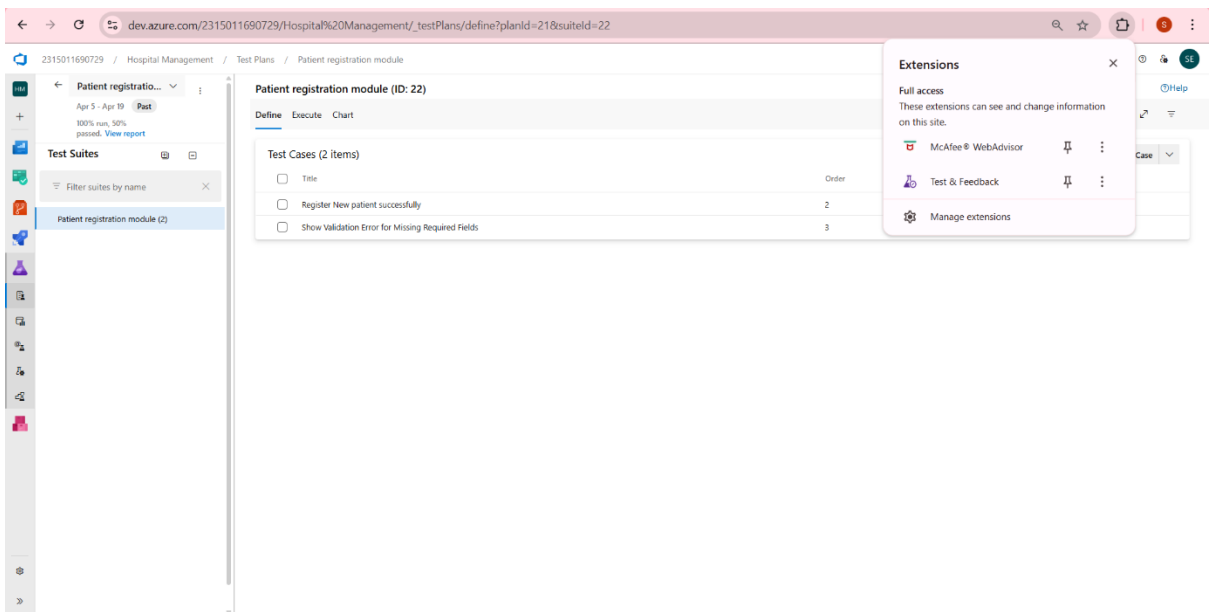
Test cases



4. Installation of test



Test and feedback
Showing it as an extension



5. Running the test case

dev.azure.com/2315011690729/Hospital%20Management/_testPlans/execute?planId=27&suiteId=28

2315011690729 / Hospital Management / Test Plans / Appointment Booking Mod...

Appointment Booking Module (ID: 28)

Define Execute Chart

Test Suites

Filter suites by name

Appointment Booking Module (2)

Test Points (2 items)

Title	Outcome	Order	Test Case Id	Configuration	Tester
Book Appointment Successfully	Passed	1	29	Windows 10	Swathi E
Prevent Double Booking	Failed	2	30	Windows 10	Swathi E

Run for web application

dev.azure.com/2315011690729/Hospital%20Management/_testExecution/index

Save and close Create bug

Test 1 of 2

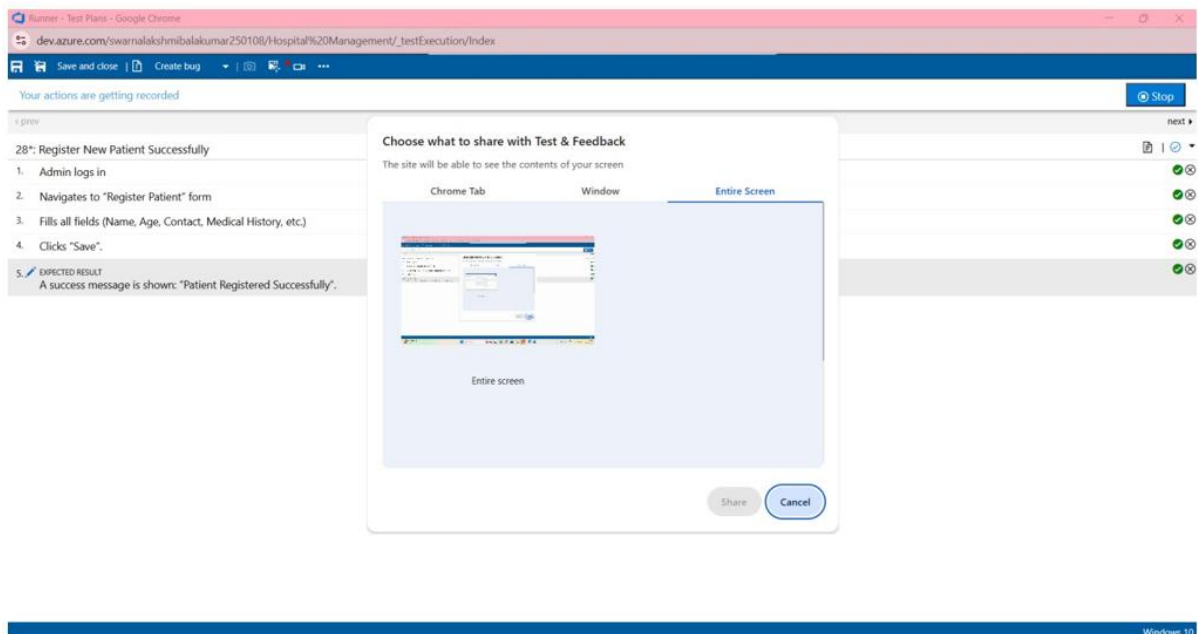
24*: Register New patient successfully

1. Login as Admin to the Azure-hosted HMS portal
2. Navigate to the "Register New Patient" form
3. Fill in valid details: Name, Age, Gender, Contact, Medical History
4. Click "Save"

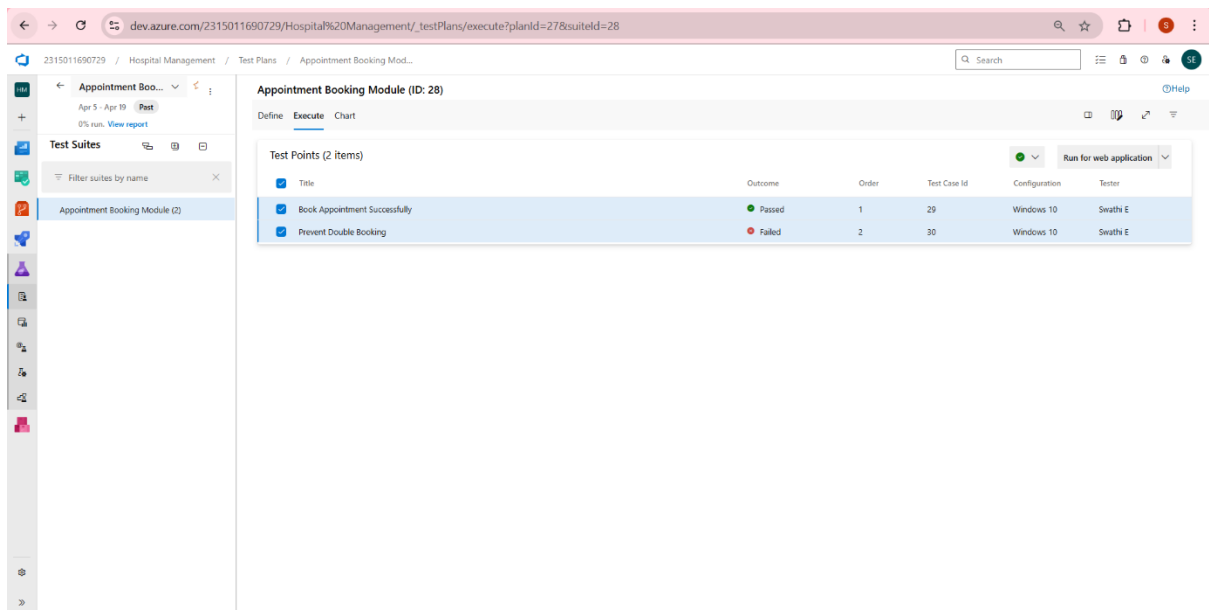
SUCCESSFUL RESULT
A success message is shown: "Patient Registered Successfully"

Windows 10

6. Recording the test case



7. Test case results



8. Test report summary

The screenshot shows the Azure DevOps Test Results page for a test plan titled "29*: Book Appointment Successfully". The page displays a list of test steps, all of which passed. The test configuration is Windows 10. The page also shows a "Repro Steps" section with a detailed description of the test steps and their results.

Test Steps Summary:

Step no.	Result	Title
1.	Passed	Login to the Azure-hosted patient portal
2.	Passed	Navigate to "Book Appointment"
3.	Passed	Select Doctor, Date, and Available Time Slot
4.	Passed	Click "Confirm"
5.	Passed	

Expected Result: Confirmation message displayed: "Appointment Confirmed"

Test Configuration: Windows 10

System Info: Found in Build

9. Progress report

The screenshot shows the Azure DevOps Progress Report page for the "Appointment Booking Module". The page displays a summary of test results, including the number of test plans, test points, and the pass rate. It also includes an "Outcome trend" chart showing the progress of test points over time.

Summary:

- 1 Test plans
- 2 Test points
- 2 (2 / 2) Test points run
- 100% Run
- 50% (1 / 2) Pass rate
- 1 Passed
- 1 Failed

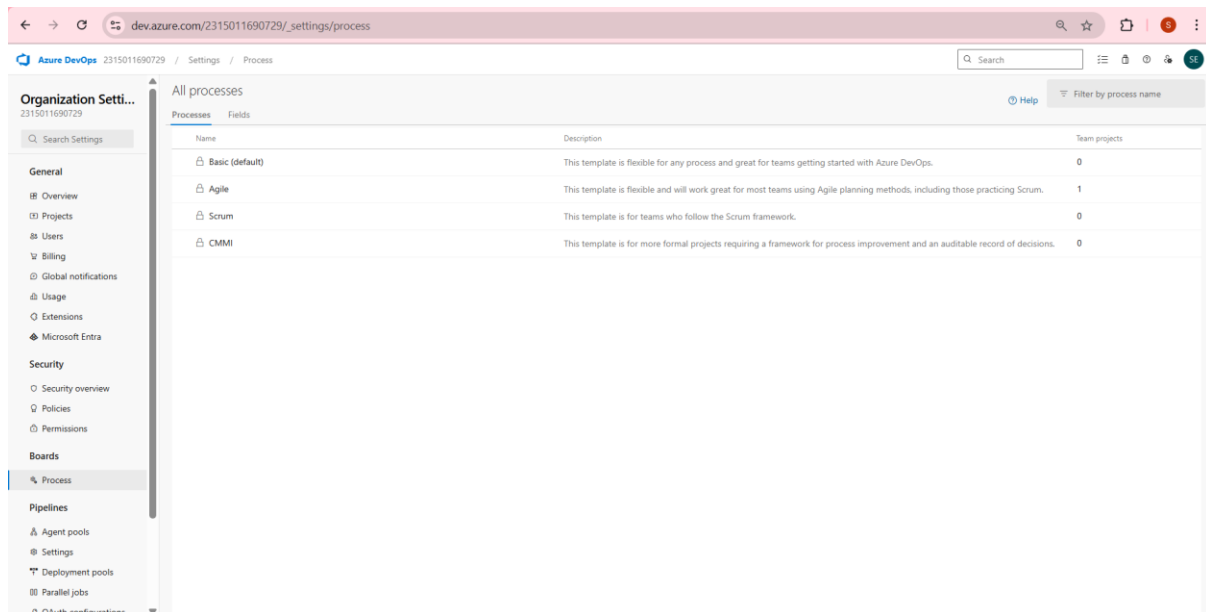
Outcome trend (Last 14 Days):

The chart shows the progress of test points over time. The x-axis represents dates from 2023-05-06 to 2023-05-20. The y-axis represents the number of test points. The legend indicates: Not run (grey), Passed (green), and Failed (red).

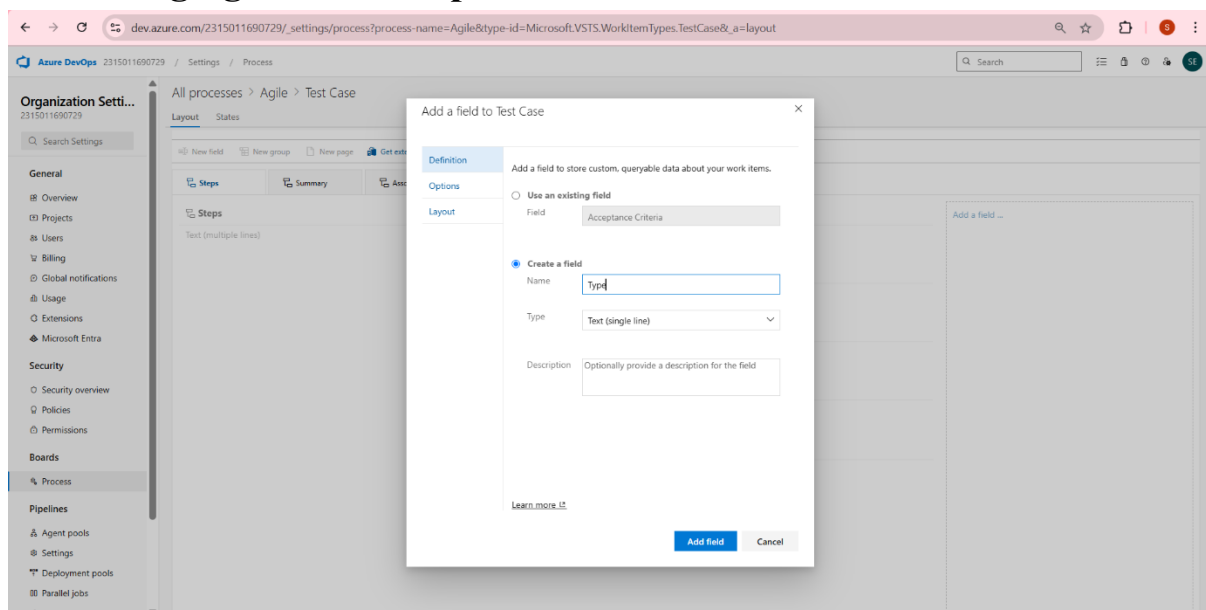
Details:

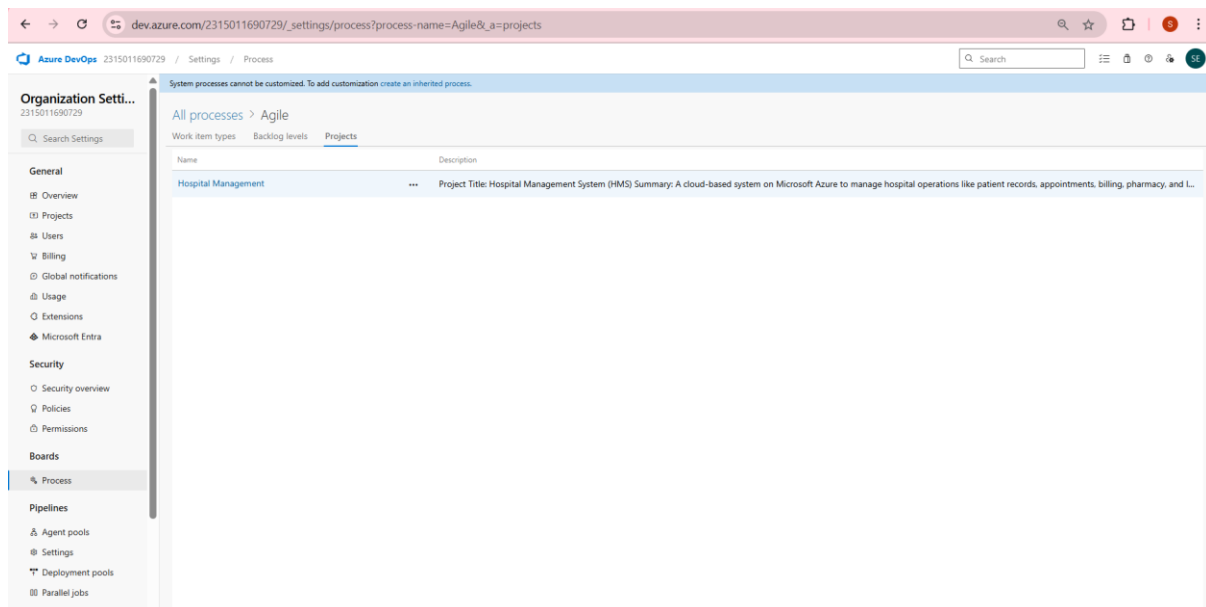
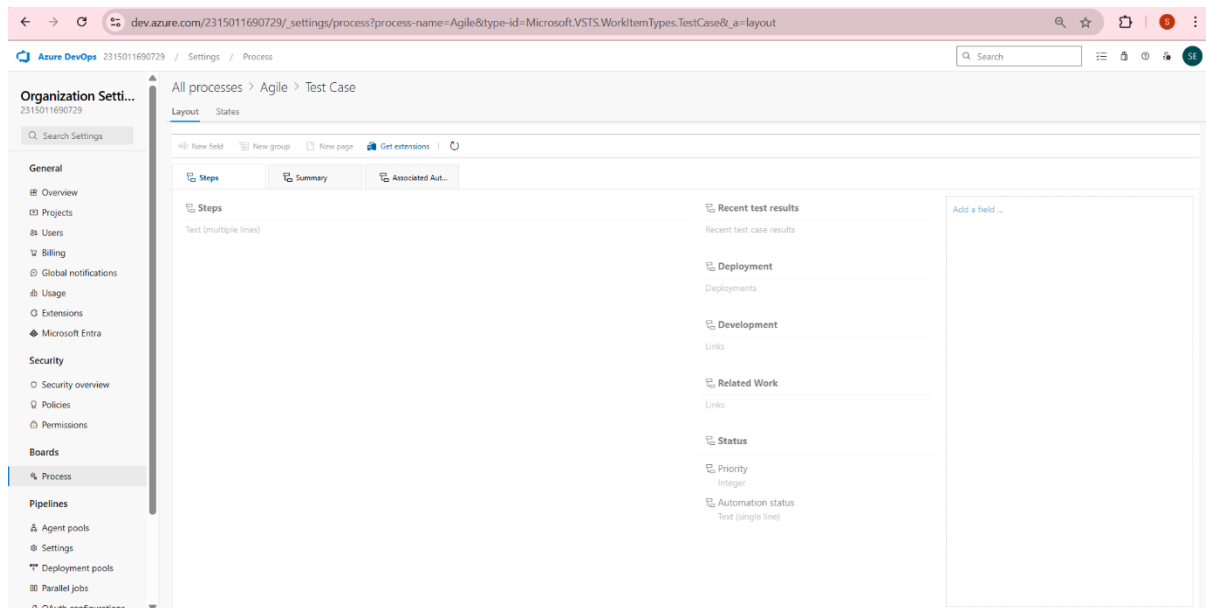
Test plan name	Test points	Run %	Passed %	Failed %	Not run count
Appointment Booking Module	2	100	50	50	0

10. Changing the test template



11. Changing the test template





Result:

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

Aim:

To implement a Continuous Integration and Continuous Deployment (CI/CD) pipeline in Azure DevOps for automating the build, testing, and deployment process of the Student Management System, ensuring faster delivery and improved software quality.

PROCEDURE:**Steps to Create and implement pipelines in Azure:****1. Sign in to Azure DevOps and Navigate to Your Project**

Log in to dev.azure.com, select your organization, and open the project where your Student Management System code resides.

2. Connect a Code Repository (Azure Repos or GitHub)

Ensure your application code is stored in a Git-based repository such as Azure Repos or GitHub. This will be the source for triggering builds and deployments in your pipeline.

3. Create a New Pipeline

Go to the Pipelines section on the left panel and click “Create Pipeline”. Choose your source (e.g., Azure Repos Git or GitHub), and then select the repository containing your project code.

4. Choose the Pipeline Configuration

You can select either the YAML-based pipeline (recommended for version control and automation) or the Classic Editor for a GUI-based setup.

If using YAML, Azure DevOps will suggest a template or allow you to define your own.

5. Define Build Stage (CI - Continuous Integration) from YAML file

6. Install dependencies (e.g., npm install, dotnet restore)

7. Build the application (dotnet build, npm run build)

8. Run unit tests (dotnet test, npm test)

9. Publish build artifacts to be used in the release stage

10. Save and Run the Pipeline for the First Time

Save the YAML or build definition and click “Run”. Azure will fetch the latest code and execute the defined build and test stages.

11. Configure Continuous Deployment (CD)

Navigate to the Releases tab under Pipelines and click “New Release Pipeline”. Add an Artifact (from the build stage) and create a new Stage (e.g., Development, Production).

12. Configure the CD stage with deployment tasks such as deploying to Azure App Service, running database migrations or scripts, and restarting services using the Azure App Service Deploy task linked to your subscription and app details.

13. Set Triggers and Approvals

Enable continuous deployment trigger so the release pipeline runs automatically after a successful build. For production environments, configure pre-deployment approvals to ensure manual verification before release.

14. Monitor Pipelines and Manage Logs

View all pipeline runs under the Runs section.

Check logs for build/test/deploy stages to debug any errors.

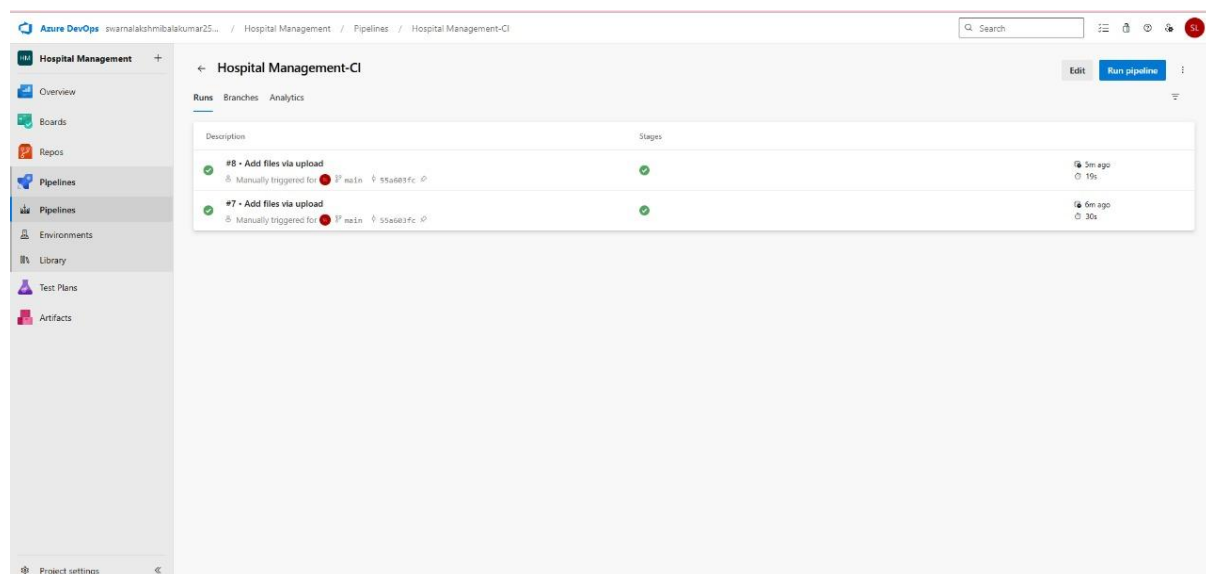
You can also integrate email alerts or Microsoft Teams notifications for build failures.

15. Review and Maintain Pipelines

Regularly update your pipeline tasks or YAML configurations as your application grows. Ensure pipeline runs are clean and artifacts are stored securely.

Integrate quality gates and code coverage policies to maintain code quality.

CI/CD PIPELINES



Azure DevOps swarnalakshmbalakumar25... / Hospital Management / Pipelines / Hospital Management-CI

Search

Hospital Management +

- Overview
- Boards
- Repos
- Pipelines
- Pipelines
- Environments
- Library
- Test Plans
- Artifacts

Project settings

Hospital Management-CI

Runs Branches Analytics

Description	Stages	
#8 - Add files via upload Manually triggered for main 55a603fc	✓	2h ago 19s
#7 - Add files via upload Manually triggered for main 55a603fc	✓	2h ago 30s

Azure DevOps swarnalakshmbalakumar25... / Hospital Management / Pipelines / Hospital Management-CI / 8

Search

Hospital Management +

- Overview
- Boards
- Repos
- Pipelines
- Pipelines
- Environments
- Library
- Test Plans
- Artifacts

Project settings

#8 • Add files via upload

Hospital Management-CI

This run is being retained as one of 3 recent runs by main (Branch). [View retention leases](#)

Summary Code Coverage

Manually run by swarna lakshmi

Repository and version
Hospital Management
main 55a603fc

Time started and elapsed
Today at 2:55 PM
19s

Related
0 work items
1 published; 1 consumed

Tests and coverage
[Get started](#)

Jobs

Name	Status	Duration
Agent job 1	Success	14s

Azure DevOps swarnalakshmbalakumar25... / Hospital Management / Pipelines / Hospital Management-CI / 7

Search

Hospital Management +

- Overview
- Boards
- Repos
- Pipelines
- Pipelines
- Environments
- Library
- Test Plans
- Artifacts

Project settings

#7 • Add files via upload

Hospital Management-CI

This run is being retained as one of 3 recent runs by main (Branch). [View retention leases](#)

Summary Code Coverage

Manually run by swarna lakshmi

Repository and version
Hospital Management
main 55a603fc

Time started and elapsed
Today at 2:54 PM
30s

Related
0 work items
1 published; 1 consumed

Tests and coverage
[Get started](#)

[View 3 changes](#)

Jobs

Name	Status	Duration
Agent job 1	Success	13s

RESULT:

Thus the pipelines for the given project Hospital Management System has been executed successfully.

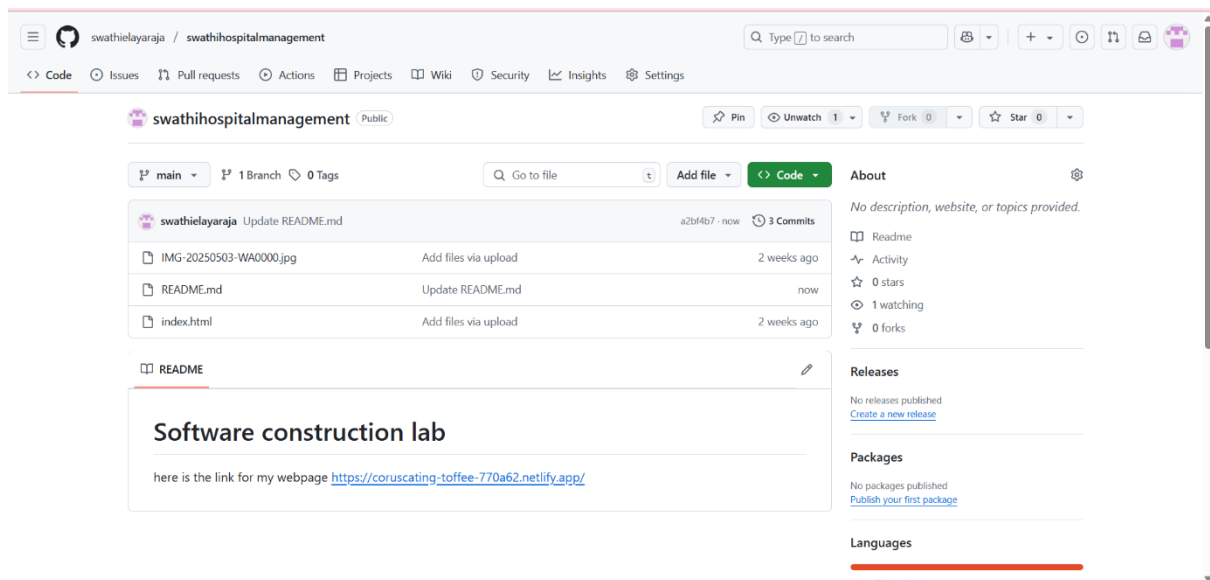
EXP NO: 10

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 project.

GitHub Project Structure:



Result:

The GitHub repository clearly displays the organized project structure and consistent naming conventions, making it easy for users and contributors to understand and navigate the codebase.