RAJALAKSHMI ENGINEERING COLLEGE

RAJALAKSHMI NAGAR, THANDALAM – 602 105

RAJALAKSHMI
ENGINEERING COLLEGE

CS23432

SOFTWARE CONSTRUCTION

Laboratory Record Note Book

Name : VARSHINI.D

Year / Branch / Section : 2ND/INFORMATION TECHNOLOGY/AE

Register No. : 231001237

Semester : IV

Academic Year : 2024-2025

RAJALAKSHMI ENGINEERING COLLEGE (AUTONOMOUS)

RAJALAKSHMI NAGAR, THANDALAM – 602 105

BONAFIDE CERTIFICATE

NAME V A R S H I N I . D REGISTER NO. 231001237

ACADEMIC YEAR 2024-25 **SEMESTER**- IV **BRANCH**: B. Tech Information Technology [AD/AE]. This Certification is the Bonafide record of work done by the above student in the **CS23432**- **Software Construction** Laboratory during the year 2024-2025.

Signature of Faculty -in – Charge

Submitted for the Practical Examination held on _____

Internal Examiner External Examiner

LAB PLAN

CS23432-SOFTWARE CONSTRUCTION LAB

Ex No	Date	Topic	Page No	Sign
1	21/01/2025	Study of Azure DevOps		
2	28/01/2025	Problem Statement		
3	04/02/2025	Agile Planning		
4	18/02/2025	Create User stories with Acceptance Criteria		
5	25/02/2025	Designing Sequence Diagrams using Azure DevOps-WIKI		
6	04/03/2025	Designing Class Diagram using Azure DevOps-WIKI		
7	11/03/2025	Designing Use case Diagram using Azure DevOps-WIKI		
8	18/03/2025	Designing Activity Diagrams using Azure DevOps-WIKI		
9	25/03/2025	Designing Architecture Diagram Using Star UML		
10	01/04/2025	Design User Interface		
11	08/04/2025	Implementation – Design a Web Page based on Scrum Methodology		
12	15/04/2025	Testing-Test Plan, Test Case and Load Testing		

EXP NO : 1

DATE : 21.01.2025

STUDY OF AZURE DEVOPS

AIM :

To study how to create an agile project in Azure DevOps environment.

STUDY :

Azure DevOps is a cloud-based platform by Microsoft that provides tools for DevOps practices, including CI/CD pipelines, version control, agile planning, testing, and monitoring. It supports teams in automating software development and deployment.

1. Understanding Azure DevOps

Azure DevOps consists of five key services:

1.1 Azure Repos (Version Control)

Supports Git repositories and Team Foundation Version Control (TFVC). Provides features like branching, pull requests, and code reviews.

1.2 Azure Pipelines (CI/CD)

Automates build, test, and deployment processes.

Supports multi-platform builds (Windows, Linux, macOS).

Works with Docker, Kubernetes, Terraform, and cloud providers (Azure, AWS, GCP).

1.3 Azure Boards (Agile Project Management)

Manages work using Kanban boards, Scrum boards, and dashboards. Tracks user stories, tasks, bugs, sprints, and releases.

1.4 Azure Test Plans (Testing)

Provides manual, exploratory, and automated testing. Supports test case management and tracking.

1.5 Azure Artifacts (Package Management)

Stores and manages NuGet, npm, Maven, and Python packages. Enables versioning and secure access to dependencies.

Getting Started with Azure DevOps :

Step 1: Create an Azure DevOps Account Visit Azure DevOps.

Sign in with a Microsoft Account.

Create an Organization and a Project. Step 2: Set Up a Repository (Azure Repos)

Navigate to Repos.

Choose Git or TFVC for version control. Clone the repository and push your code.

Step 3: Configure a CI/CD Pipeline (Azure Pipelines)

Go to Pipelines → New Pipeline.

Select a source code repository (Azure Repos, GitHub, etc.).

Define the pipeline using YAML or the Classic Editor. Run the pipeline to build and deploy the application.

Step 4: Manage Work with Azure Boards Navigate to Boards.

Create work items, user stories, and tasks. Organize sprints and track progress.

Step 5: Implement Testing (Azure Test Plans) Go to Test Plans.

Create and run test cases

View test results and track bugs.

RESULT:

The study was successfully completed.

EXP NO : 2

DATE : 28.01.2025

PROBLEM STATEMENT

AIM :

To prepare problem statement for your given project.

PROBLEM STATEMENT :

Evaluation of Academic Performance

Students, teachers, and parents often struggle to track academic performance efficiently due to scattered data, lack of real-time insights, and delayed feedback mechanisms. Academic records are frequently fragmented across different platforms, leading to confusion, inefficiencies, and a lack of comprehensive academic tracking. Manual methods of recording and evaluating performance are not only time-consuming but also prone to human error, causing inaccuracies in assessments. Parents usually become aware of performance issues only after final results, reducing opportunities for early intervention and support. Additionally, students often lack a clear understanding of their strengths and weaknesses, making it difficult to take corrective actions in time. Teachers also face challenges in identifying learning gaps across classes without analytical tools. The absence of a centralized, intelligent system prevents early detection of academic trends, personalized recommendations, and targeted interventions. Therefore, there is a strong need for a smart, integrated platform that visualizes grades, tracks progress over time, analyzes academic trends, offers timely feedback, and fosters better communication among students, teachers, and parents, ultimately aiming to enhance learning outcomes and overall academic success.

RESULT:

The problem statement was written successfully.

EXP NO : 3

DATE : 04.02.2025

AGILE PLANNING

AIM :

To prepare an Agile Plan.

THEORY :

Agile planning is a part of the Agile methodology, which is a project management style with an incremental, iterative approach. Instead of using an in-depth plan from the start of the project—which is typically product-related—Agile leaves room for requirement changes throughout and relies on constant feedback from end users.

With Agile planning, a project is broken down into smaller, more manageable tasks with the ultimate goal of having a defined image of a project's vision. Agile planning involves looking at different aspects of a project's tasks and how they'll be achieved, for example:

· Roadmaps to guide a product's release ad schedule

· Sprints to work on one specific group of tasks at a time

· A feedback plan to allow teams to stay flexible and easily adapt to change

User stories, or the tasks in a project, capture user requirements from the end user's perspective Essentially, with Agile planning, a team would decide on a set of user stories to action at any given time, using them as a guide to implement new features or functionalities in a tool. Looking at tasks as user stories is a helpful way to imagine how a customer may use a feature and helps teams prioritize work and focus on delivering value first.

· Steps in Agile planning process

1. Define vision
2. Set clear expectations on goals
3. Define and break down the product roadmap
4. Create tasks based on user stories
5. Populate product backlog
6. Plan iterations and estimate effort

7. Conduct daily stand-ups

8. Monitor and adapt

RESULT:

Thus the Agile plan was completed successfully.

EXP NO : 4

DATE : 18.02.2025

CREATE USER STORIES WITH ACCEPTANCE CRITERIA

AIM :

To create User Stories

THEORY :

A user story is an informal, general explanation of a software feature written from the perspective of the end user. Its purpose is to articulate how a software feature will provide value to the customer.

User story template

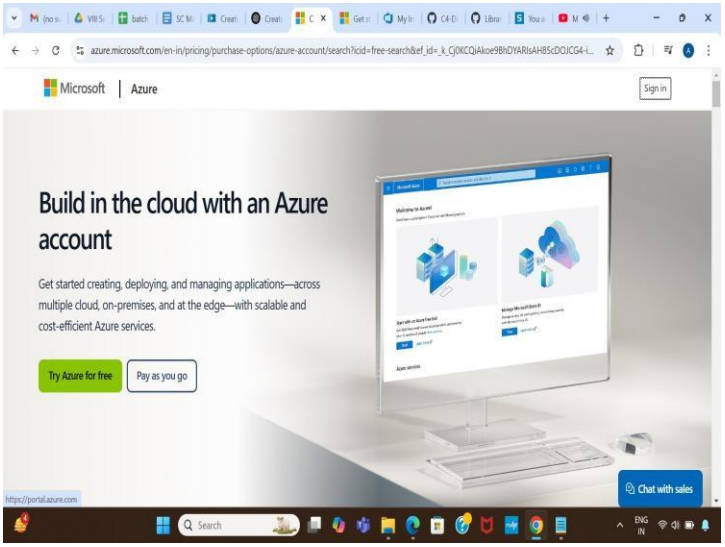
"As a [role], I [want to], [so that]."

PROCEDURE:

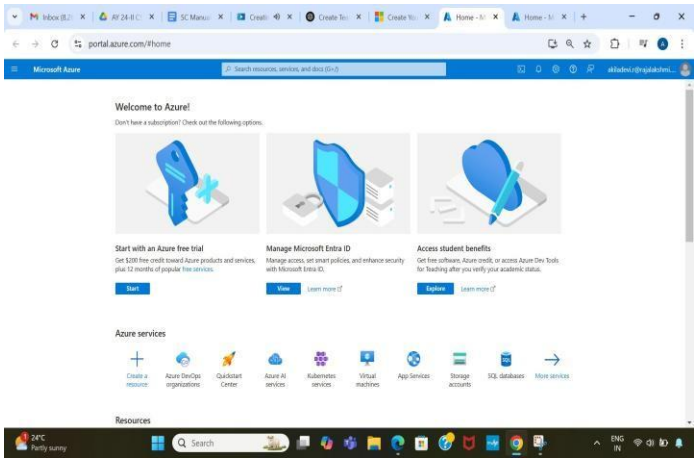
1. Open your web browser and go to the Azure website: https://azure.microsoft.com/en-in Sign in using your Microsoft account credentials. If you don't have an account, you'll need to create one.

2. If you don't have a Microsoft account, you can sign up for

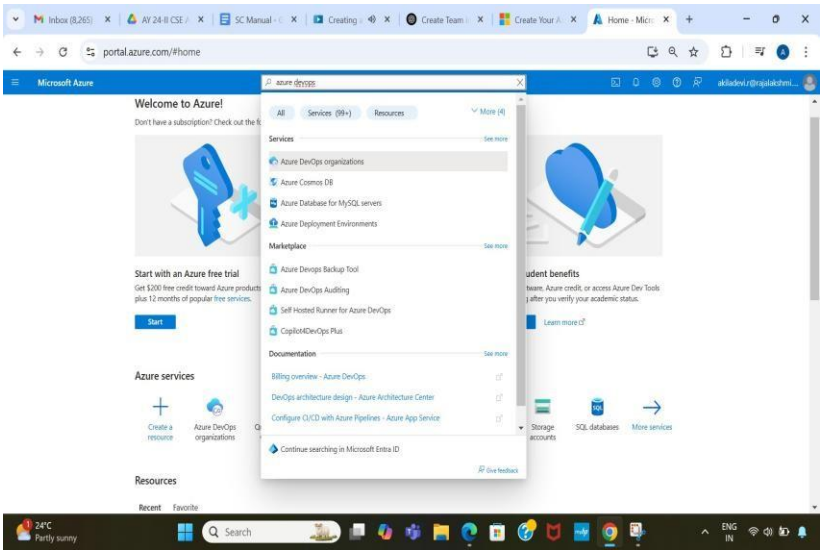
https://signup.live.com/?lic=1



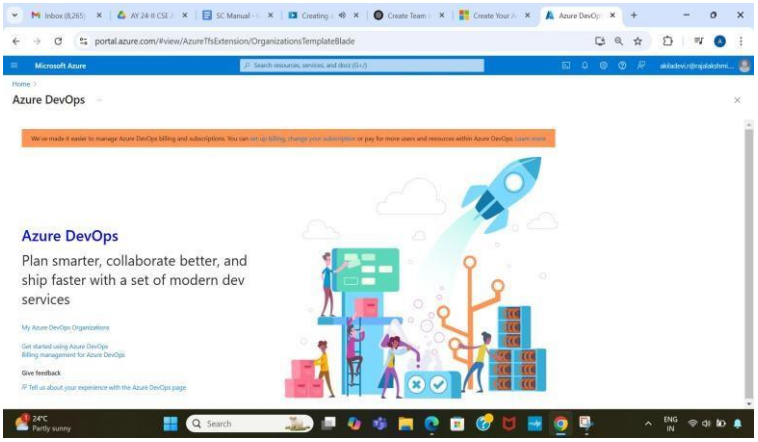
3. Azure home page

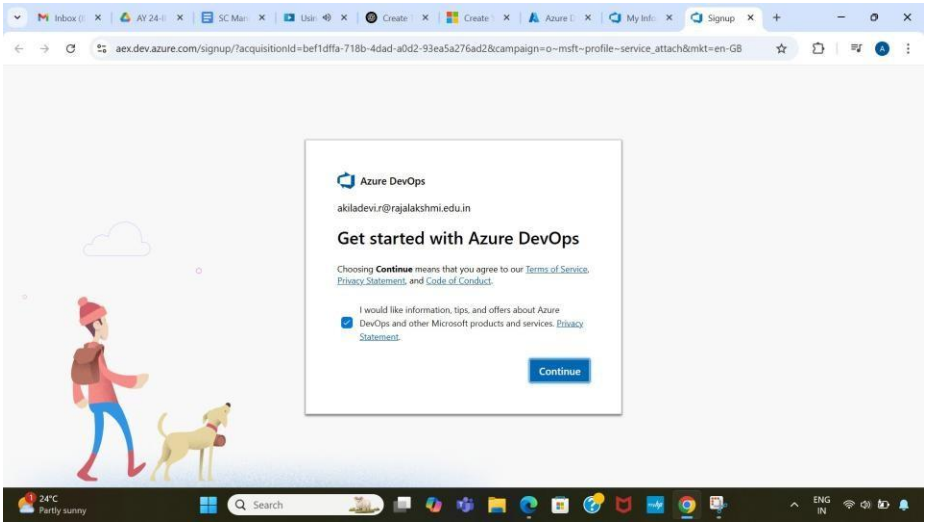


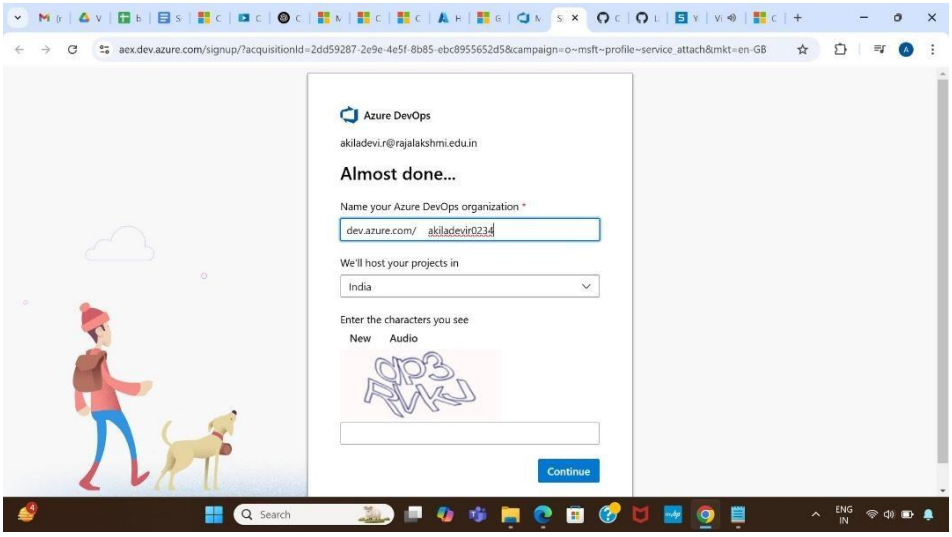
4. Open DevOps environment in the Azure platform by typing Azure DevOps Organizations in the search bar.



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







6. 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.

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

ii. Enter the project name, description, and visibility options:

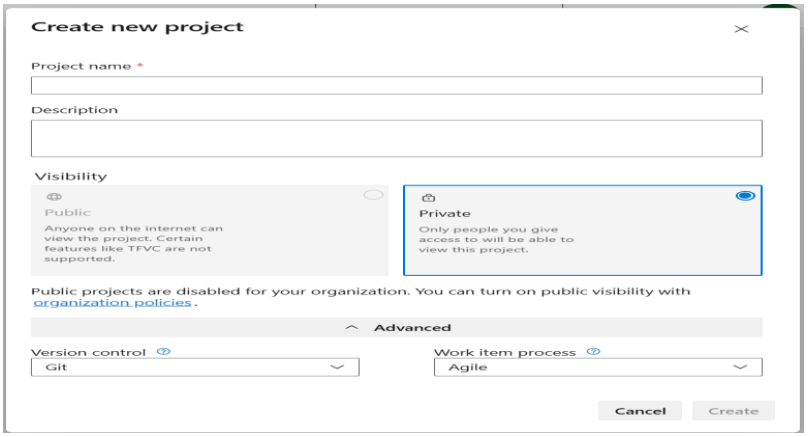
○ Name: Choose a name for the project (e.g., LMS).

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

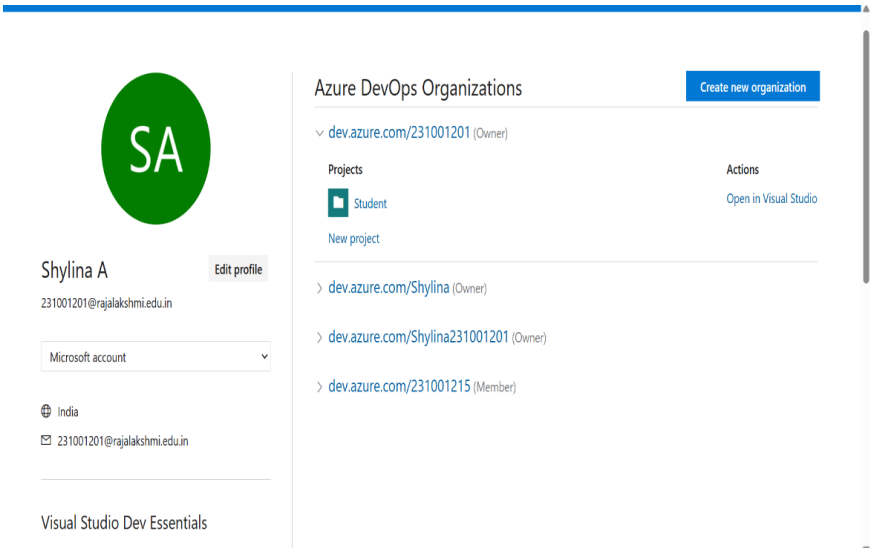
○ Visibility: Choose whether you want the project to be Private

(accessible only to those invited) or Public (accessible to anyone).

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



7. 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.

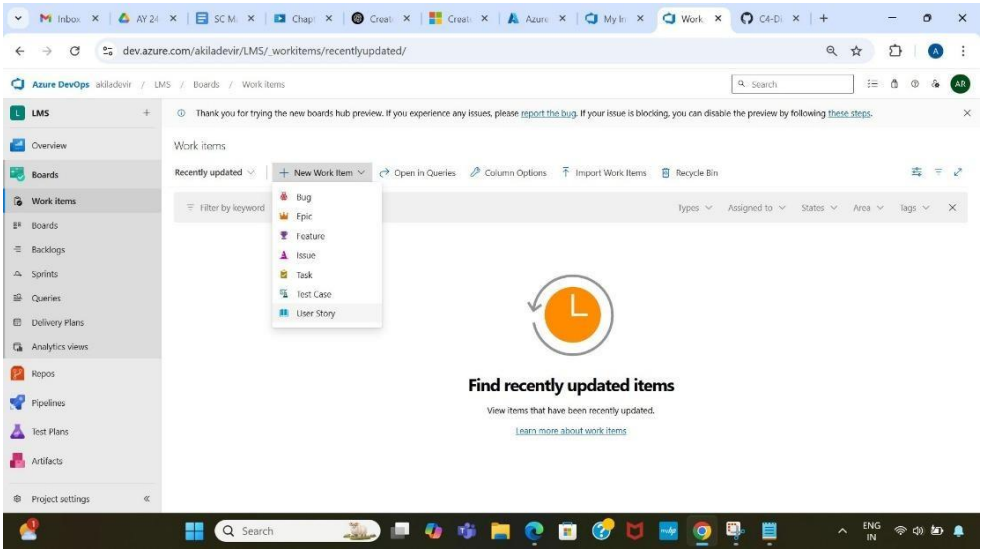


8. Project dashboard

9. To manage user stories

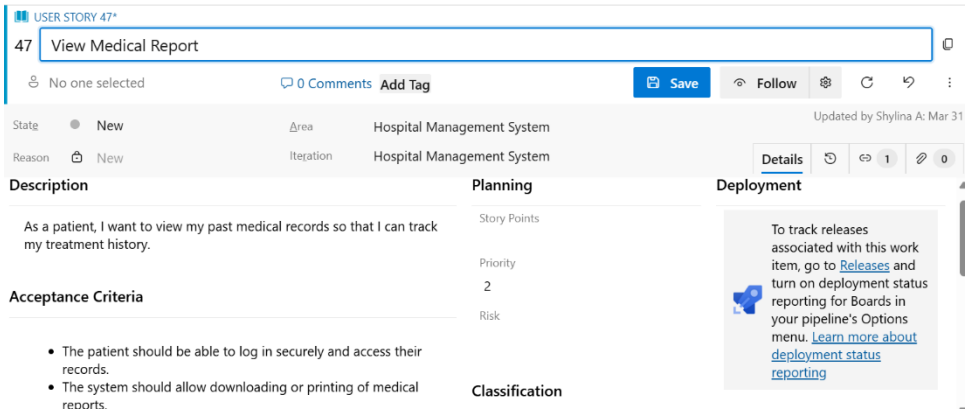
a. From the left-hand navigation menu, click on Boards. This will take you to the main Boards page, where you can manage work items, backlogs, and sprints.

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.



11. Fill in User Story Details

1.



RESULT:

The user story was written successfully.

EXP NO : 5

DATE : 25.02.2025

DESIGNING SEQUENCE DIAGRAM USING AZURE DEVOPS - WIKI

AIM :

To design a Sequence Diagram by using Mermaid.js

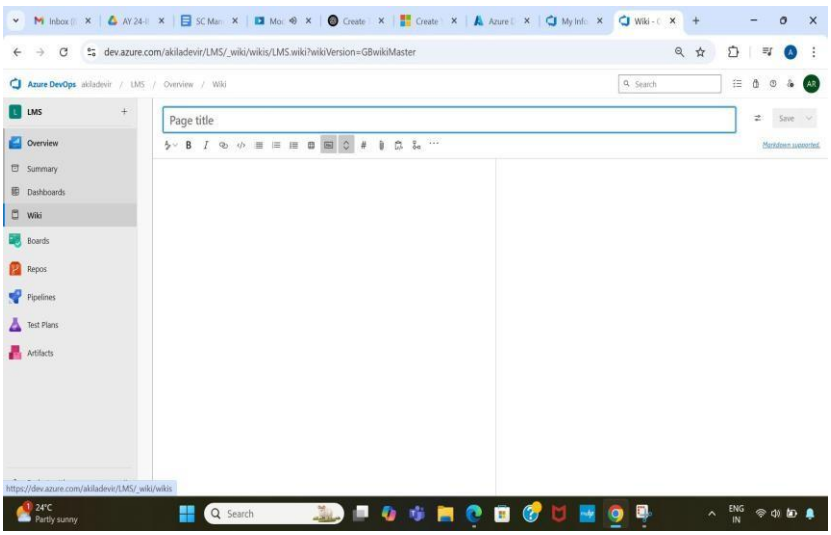
THEORY:

A Sequence Diagram is a key component of Unified Modelling Language (UML) used to visualize the interaction between objects in a sequential order. It focuses on how objects communicate with each other over time, making it an essential tool for modelling dynamic behaviour in a system.

PROCEDURE:

1. Open a project in Azure DevOps Organisations.

2. To design select wiki from menu



3. Write code for drawing sequence diagram and save the code.

:::mermaid

sequenceDiagram

participant Patient

participant Receptionist

participant Doctor

participant Nurse

participant Database

participant Pharmacy

```
    participant Billing

    Patient->>Receptionist: Request Appointment
    Receptionist->>Database: Check Availability
    Database-->>Receptionist: Availability Confirmed
    Receptionist-->>Patient: Appointment Scheduled

    Patient->>Doctor: Attend Appointment
    Doctor->>Database: Retrieve Medical Records
    Database-->>Doctor: Send Records
    Doctor->>Nurse: Request Tests
    Nurse->>Database: Record Test Results
    Database-->>Doctor: Test Results Available
    Doctor->>Patient: Diagnose and Prescribe Medicine

    Doctor->>Pharmacy: Send Prescription
    Pharmacy-->>Patient: Provide Medicine

    Patient->>Billing: Make Payment
    Billing-->>Patient: Generate Invoice
:::
```

Explanation:

participant defines the entities involved.

->> represents a direct message.

-->> represents a response message.

+ after ->> activates a participant.

- after -->> deactivates a participant. alt / else for conditional flows.

loop can be used for repeated actions.

-> Solid line without arrow

--> Dotted line without arrow

->> Solid line with arrowhead

-->> Dotted line with arrowhead

<<->> Solid line with bidirectional arrowheads (v11.0.0+)

<<-->> Dotted line with bidirectional arrowheads (v11.0.0+)

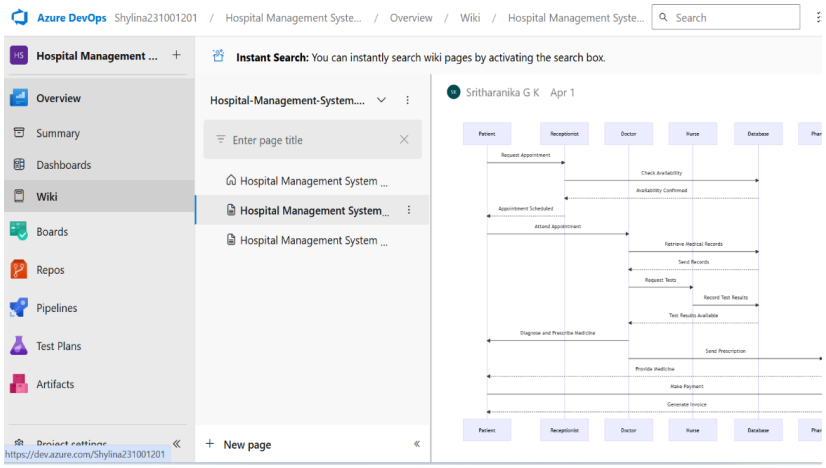
-x Solid line with a cross at the end

--x Dotted line with a cross at the end

-) Solid line with an open arrow at the end (async)

--) Dotted line with a open arrow at the end (async)

4. Click wiki menu and select the page



RESULT:

The sequence diagram was drawn successfully.

EXP NO : 6

DATE : 04.03.2025

DESIGNING CLASS DIAGRAM USING AZURE DEVOPS - WIKI

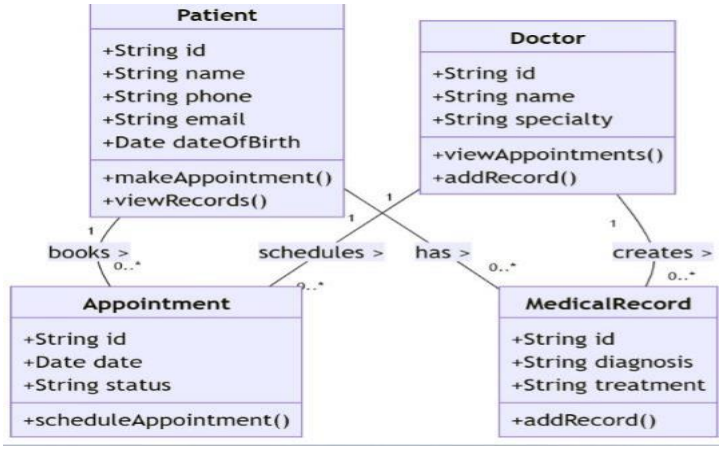
AIM :

To draw a simple class diagram for your project or system.

THEORY :

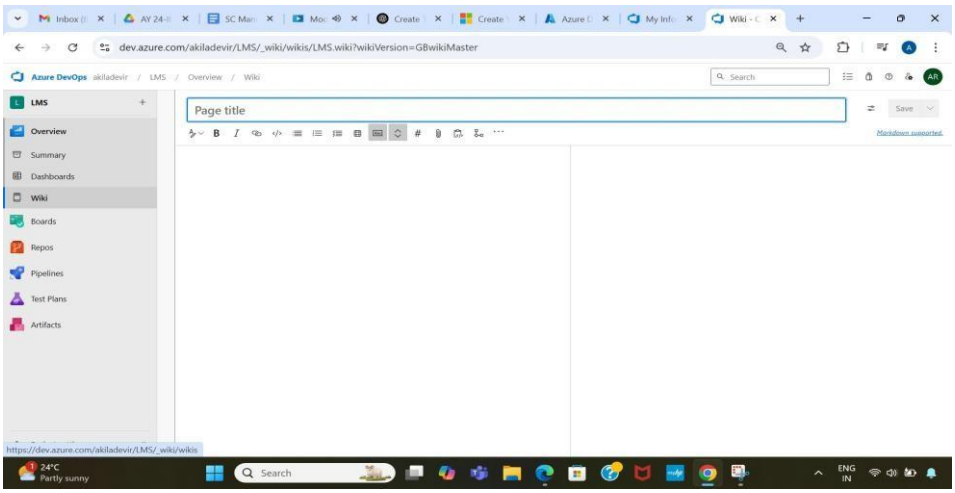
A UML class diagram is a visual tool that represents the structure of a system by showing its classes, attributes, methods, and the relationships between them.

Notations in class diagram



PROCEDURE:

1. Open a project in Azure DevOps Organisations.
2. To design select wiki from menu



3. Write code for drawing class diagram and save the code

```
:::mermaid classDiagram
    direction TB

    class Patient {
        +String id
        +String name
        +String phone
        +String email
        +Date dateOfBirth
        +makeAppointment()
        +viewRecords()
    }

    class Doctor {
        +String id
        +String name
        +String specialty
        +viewAppointments()
        +addRecord()
    }

    class Appointment {
        +String id
        +Date date
        +String status
        +scheduleAppointment()
    }

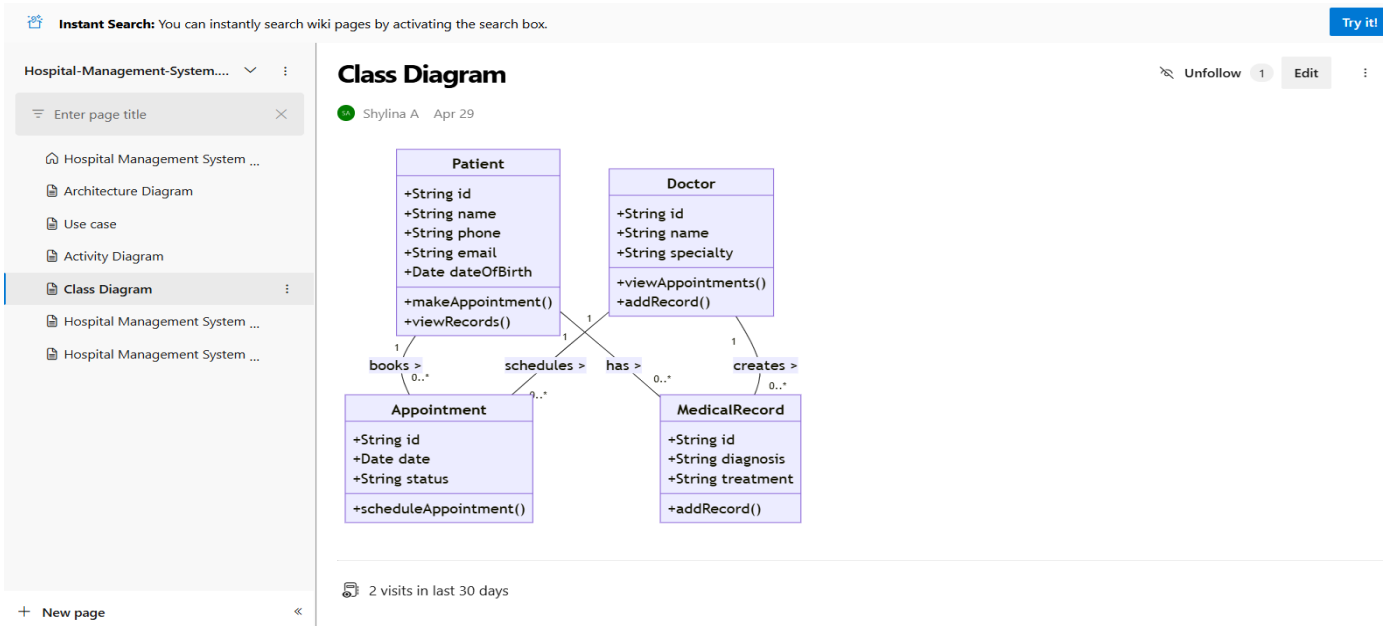
    class MedicalRecord {
        +String id
        +String diagnosis
        +String treatment
        +addRecord()
    }

    Patient "1" -- "0..*" Appointment
: books >
    Doctor "1" -- "0..*" Appointment
: schedules >
```

Patient "1" -- "0..*" MedicalRecord
: has >
Doctor "1" -- "0..*"
MedicalRecord : creates >

Relationship Types

Type	Description
<|	Inheritance
*	Composition
O	Aggregation
➢	Association
<	Association
|>	Realization



RESULT:

The use case diagram was designed successfully.

EXP NO : 7

DATE : 11.03.2025

DESIGNING USECASE DIAGRAM USING AZURE DEVOPS - WIKI

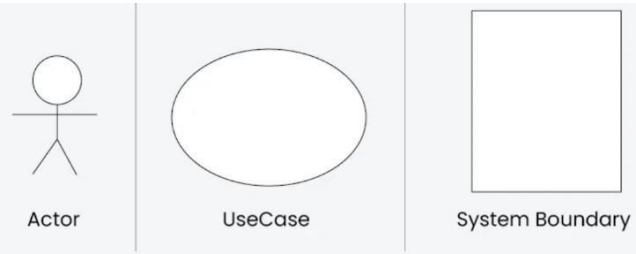
AIM :

Steps to draw the Use Case Diagram using draw.io

THEORY:

UCD shows the relationships among actors and use cases within a system which Provide an overview of all or part of the usage requirements for a system or organization in the form of an essential model or a business model and communicate the scope of a development project

- Use Cases
- Actors
- Relationships
- System Boundary Boxes



PROCEDURE:

Step 1: Create the Use Case Diagram in Draw.io

- Open Draw.io (diagrams.net).
- Click "Create New Diagram" and select "Blank" or "UML Use Case" template.
- Add Actors (Users, Admins, External Systems) from the UML section.
- Add Use Cases (Functionalities) using ellipses.
- Connect Actors to Use Cases with lines (solid for direct interaction, dashed for <<include>> and <<extend>>).
- Save the diagram as .drawio or export as PNG/JPG/SVG.

Step 2: Upload the Diagram to Azure DevOps

Option 1: Add to Azure DevOps Wiki

- Open Azure DevOps and go to your project.
- Navigate to Wiki (Project > Wiki).
- Click "Edit Page" or create a new page.
- Drag & Drop the exported PNG/JPG image.
- Use Markdown to embed the diagram:
- ![Use Case Diagram](attachments/use_case_diagram.png)

Option 2: Attach to Work Items in Azure Boards

- Open Azure DevOps → Navigate to Boards (Project > Boards).
- Select a User Story, Task, or Feature.
- Click "Attachments" → Upload your Use Case Diagram.
- Add comments or descriptions to explain the use case.

RESULT:

The use case diagram was designed successfully.

EXP NO : 8

DATE : 18.03.2025

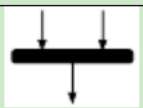
DESIGNING ACTIVITY DIAGRAM USING AZURE DEVOPS - WIKI

AIM :

To draw a sample activity diagram for your project or system.

THEORY :

Activity diagrams are an essential part of the Unified Modelling Language (UML) that help visualize workflows, processes, or activities within a system. They depict how different actions are connected and how a system moves from one state to another.

Notations	Symbol	Meaning
Start		Shows the beginning of a process
Connector		Shows the directional flow, or control flow, of the activity
Joint symbol		Combines two concurrent activities and re-introduces them to a flow where one activity occurs at a time
Decision		Represents a decision
Note		Allows the diagram creators o communicate additional messages
Send signal		Show that a signal is being sent to a receiving activity
Receive signal		Demonstrates the acceptance of an event
Flow final symbol		Represents the end of a specific process flow
Option loop		Allows the creator to model a repetitive sequence within the option loop symbol
Shallow history pseudostate		Represents a transition that invokes the last active state.
End		Marks the end state of an activity and represents the completion of all flows of a process

PROCEDURE :

1. Draw diagram in draw.io

2. Upload the diagram in Azure DevOps wiki

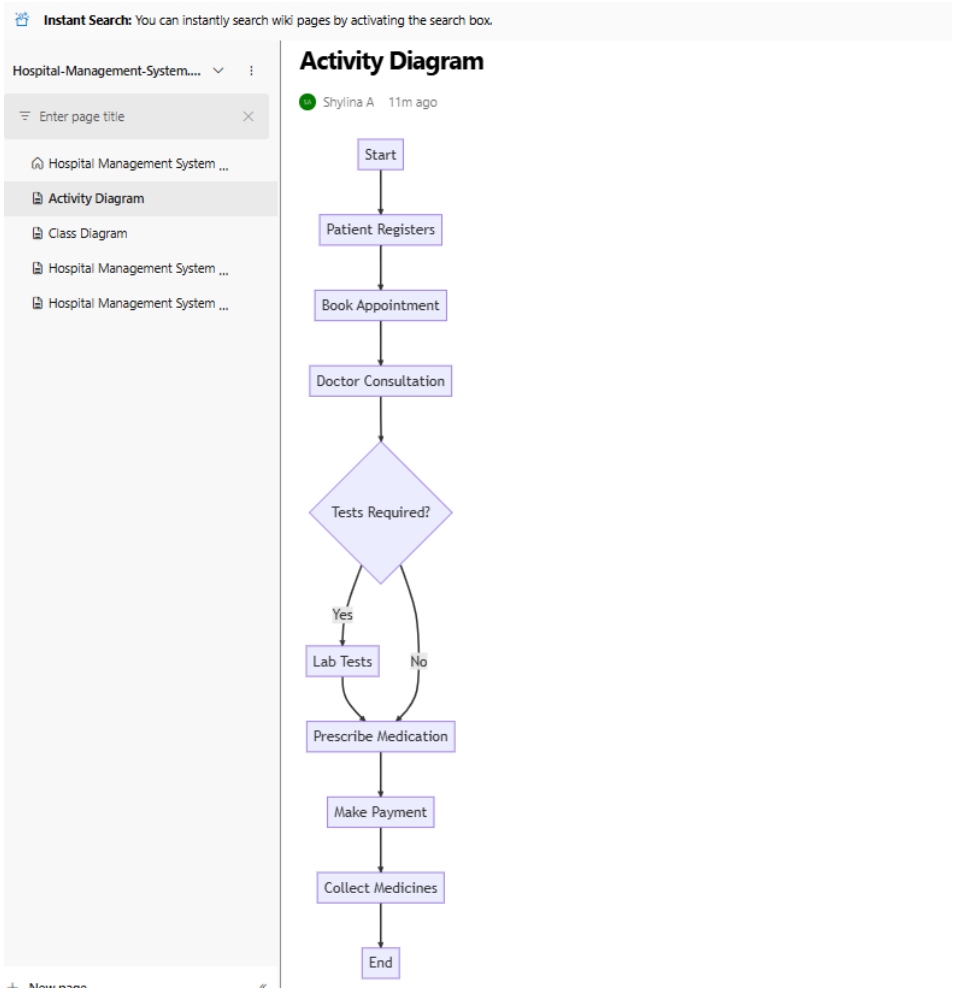
Activity Diagram

Close Save

Markdown supported.

```
::: mermaid
 ```mermaid
flowchart TD
    A[Start] --> B[Patient Registers]
    B --> C[Book Appointment]
    C --> D[Doctor Consultation]
    D --> E{Tests Required?}
    E -- Yes --> F[Lab Tests]
    F --> G[Prescribe Medication]
    E -- No --> G
    G --> H[Make Payment]
    H --> I[Collect Medicines]
    I --> J[End]
```
:::
```

Load diagram



RESULT:

The activity diagram was designed successfully.

EXP NO : 9

DATE : 25.03.2025

DESIGNING ARCHITECTURE DIAGRAM USING STAR UML

AIM :

Steps to draw the Architecture Diagram using draw.io.

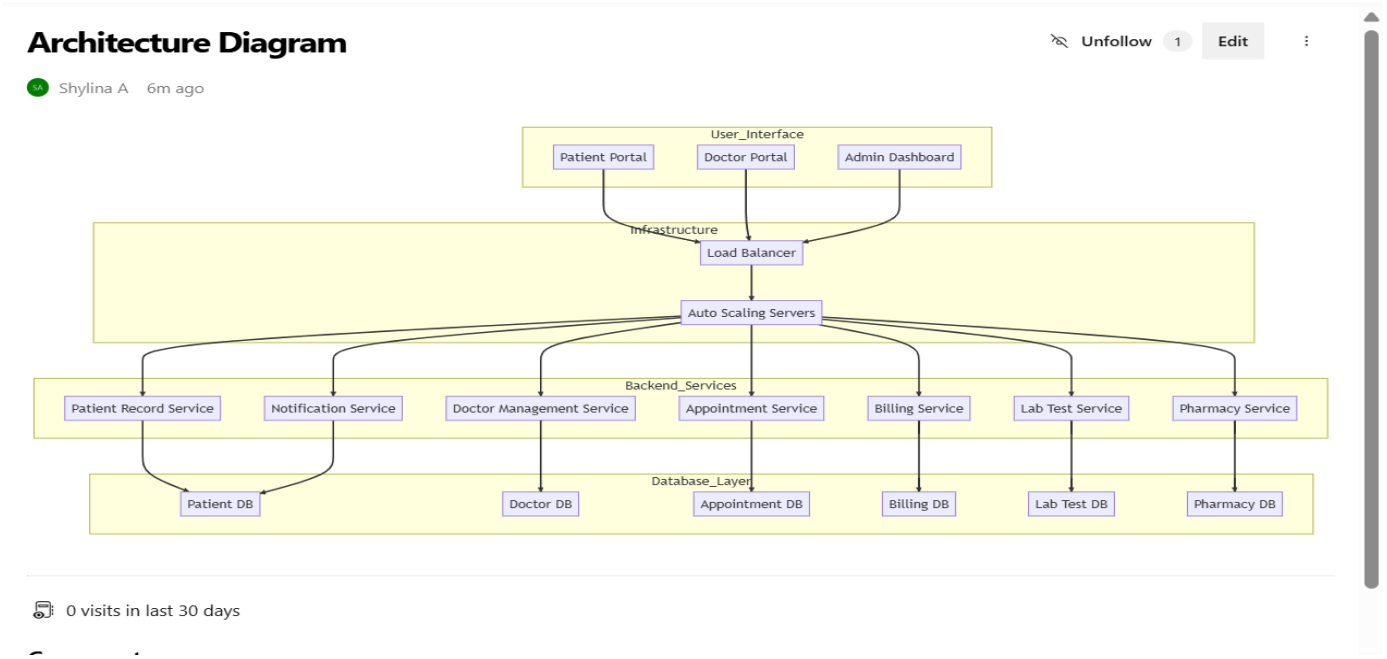
THEORY:

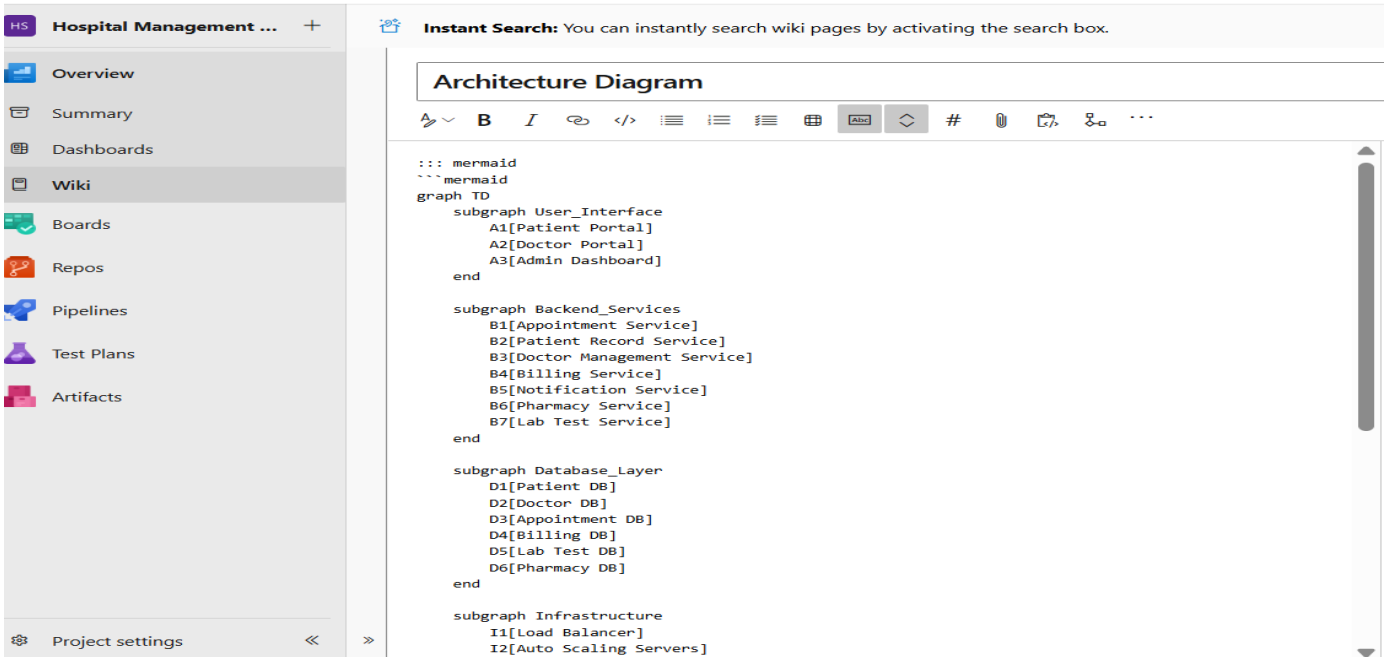
An architectural diagram is a visual representation that maps out the physical implementation for components of a software system. It shows the general structure of the software system and the associations, limitations, and boundaries between each element.

PROCEDURE :

1. Draw diagram in draw.io

2. Upload the diagram in Azure DevOps wiki





RESULT:

The architecture diagram was designed successfully.

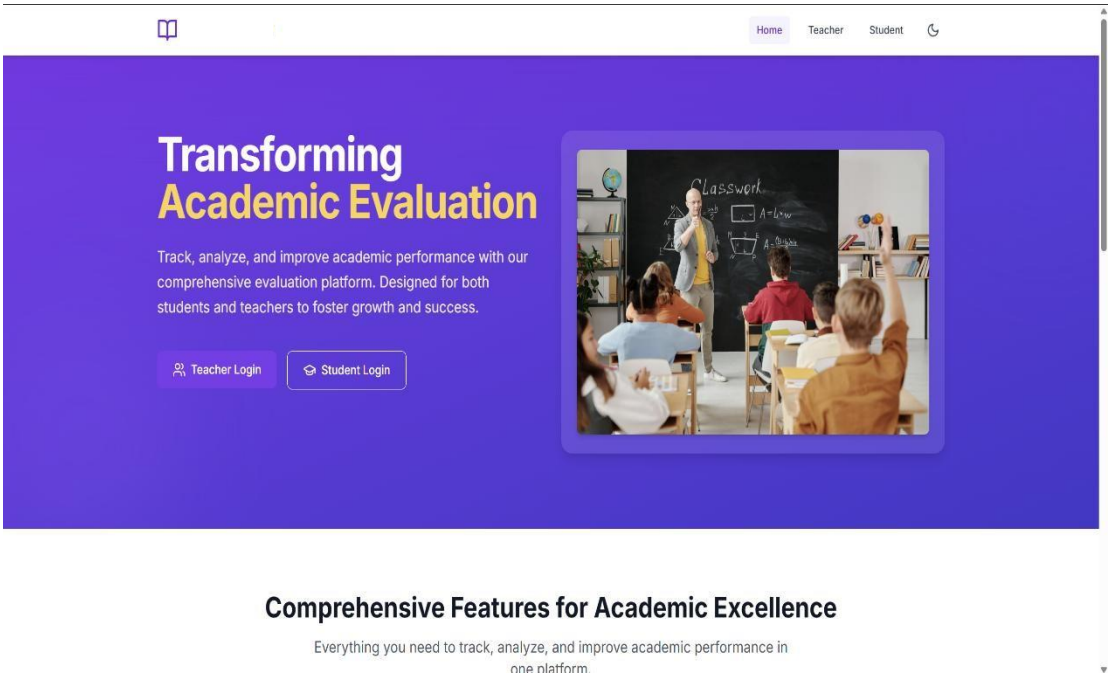
EXP NO : 10

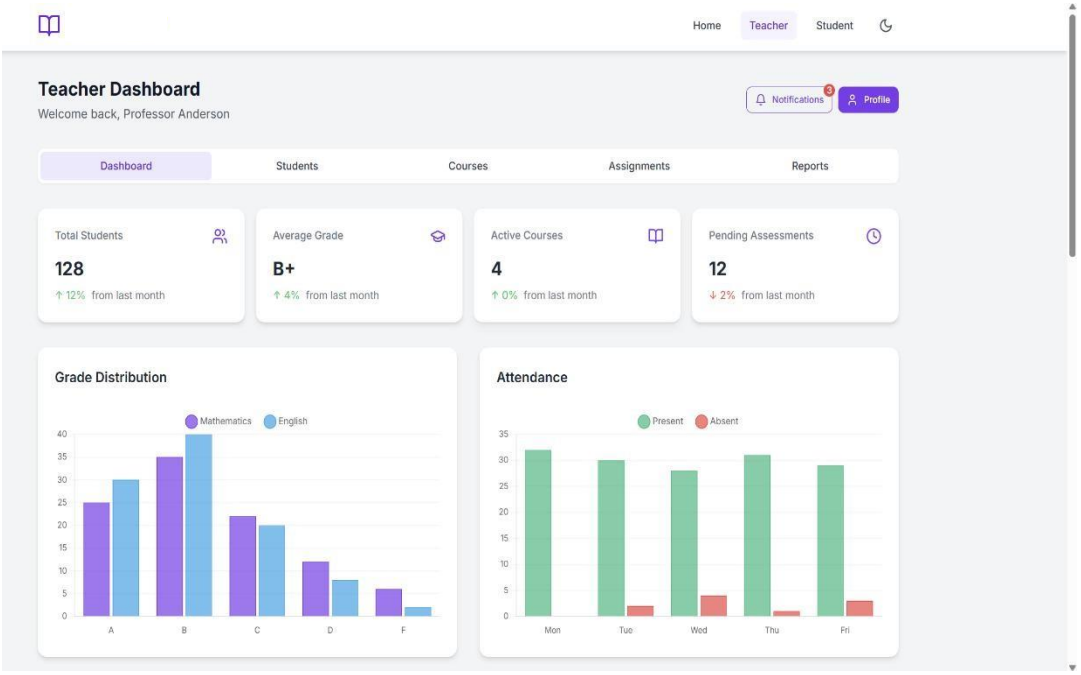
DATE : 01.04.2025

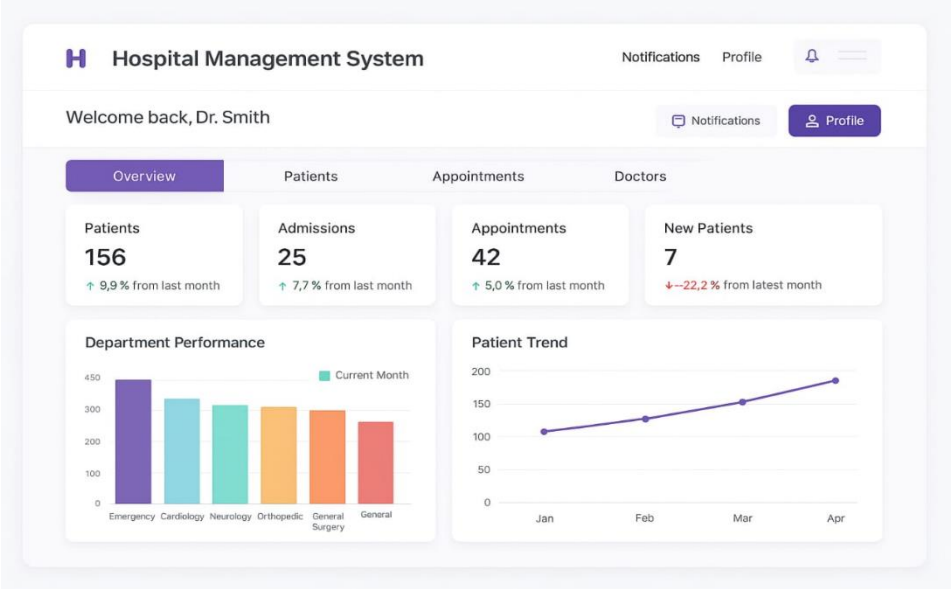
DESIGN USER INTERFACE

AIM :

Design User Interface for the given project.







RESULT:

The UI was designed successfully.

EXP NO : 11

DATE : 08.04.2025

IMPLEMENTATION – DESIGN A WEBPAGE BASED ON SCRUM METHODOLOGY

AIM :

To implement the given project based on Agile Methodology.

PROCEDURE:

Step 1: Set Up an Azure DevOps Project

- Log in to Azure DevOps.
- Click "New Project" → Enter project name → Click "Create".
- Inside the project, navigate to "Repos" to store the code.

Step 2: Add Your Web Application Code

- Navigate to Repos → Click "Clone" to get the Git URL.
- Open Visual Studio Code / Terminal and run:

  git clone <repo_url>

  cd <repo_folder>

- Add web application code (HTML, CSS, JavaScript, React, Angular, or backend like Node.js, .NET, Python, etc.).
- Commit & push:

  git add .

  git commit -m "Initial commit" git push origin main

Step 3: Set Up Build Pipeline (CI/CD - Continuous Integration)

- Navigate to Pipelines → Click "New Pipeline".
- Select Git Repository (Azure Repos, GitHub, or Bitbucket).
- Choose Starter Pipeline or a pre-configured template for your framework.
- Modify the azure-pipelines.yml file (Example for a Node.js app):

```
trigger:
- main

pool:
vmImage: 'ubuntu-latest'

steps:
- task: UseNode@1 inputs:
version: '16.x'

- script: npm install
displayName: 'Install dependencies'

- script: npm run build displayName: 'Build application'

- task: PublishBuildArtifacts@1 inputs:
pathToPublish: 'dist' artifactName: 'drop'
```

Click "Save and Run" → The pipeline will start building app.

Step 4: Set Up Release Pipeline (CD - Continuous Deployment)

- Go to Releases → Click "New Release Pipeline".
- Select Azure App Service or Virtual Machines (VMs) for deployment.
- Add an artifact (from the build pipeline).
- Configure deployment stages (Dev, QA, Production).
- Click "Deploy" to push your web app to Azure.

RESULT :

Thus the application was successfully implemented.

EXP NO : 12

DATE : 15.04.2025

TESTING – TEST PLAN, TEST CASE, LOAD TESTING

AIM:

To design and manage structured test plans and test cases in Azure DevOps for validating core user stories through both happy path and error scenarios and evaluate the performance of the application's endpoint by creating and executing load tests using Azure Load Testing.

PROCEDURE:

TEST CASE DESIGN PROCEDURE

1. Understand Core Features of the Application

- Review requirement documents and user stories.
- Identify all main functionalities of the application.
- Ensure complete coverage of modules before test case creation.

2. Define User Interactions

- Determine common user behaviors based on application flow.
- Translate user actions into testable scenarios.
- Ensure each test case mimics a real user operation.

3. Design Happy Path Test Cases

- Create test cases for expected and correct user actions.
- Ensure each functionality works under normal conditions.
- Add these cases under the relevant Test Suite in Azure DevOps.

4. Design Error Path Test Cases

- Identify edge cases, invalid inputs, and system failures.
- Test how the system handles incorrect or unexpected behavior.
- Add these test cases to the same or a separate Test Suite in Azure DevOps.

5. Break Down Steps and Expected Results

- Write step-by-step instructions in the "Steps" section of the test case.
- Provide expected results for each action.

- Ensure clarity for both manual execution and automation mapping.

6. Use Clear Naming and IDs

- Name test cases clearly using a defined naming convention (e.g., TC01, TC02, etc.).
- Ensure titles reflect the purpose of the test case.
- Azure DevOps auto-generates test case IDs for tracking.

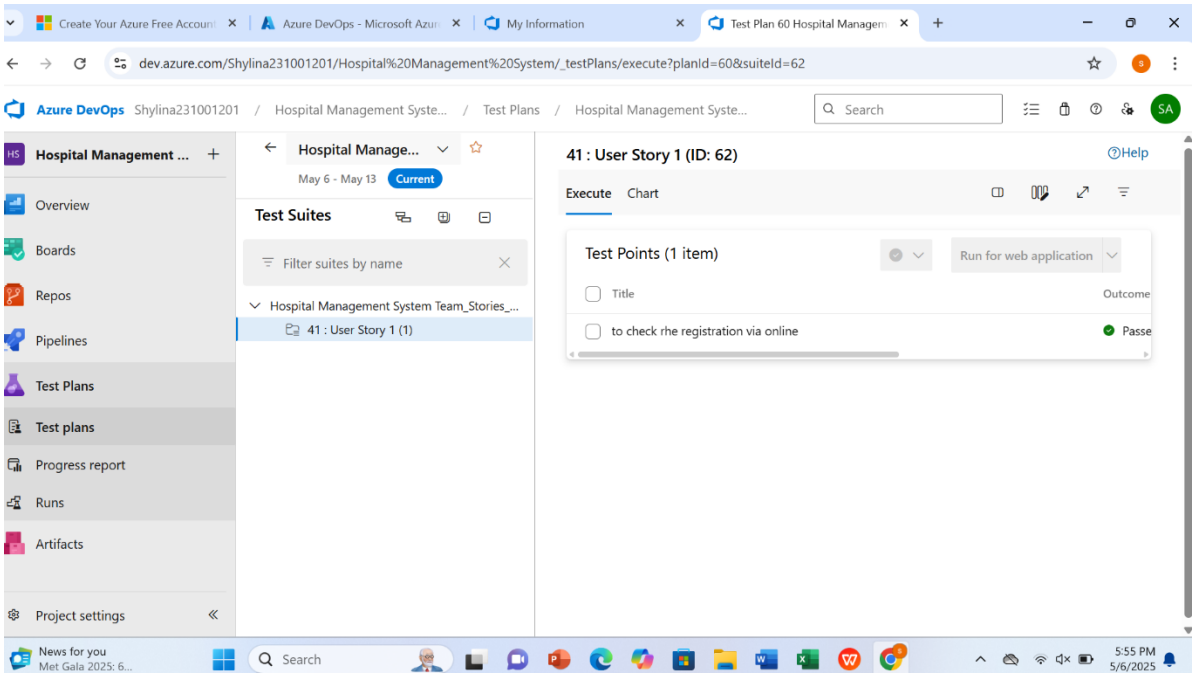
7. Separate Test Suites

- Group test cases based on functionality (e.g., Login, Playlist, Recommendations).
- Use Static, Requirement-based, or Query-based suites in Azure DevOps.
- Improves traceability and execution flow.

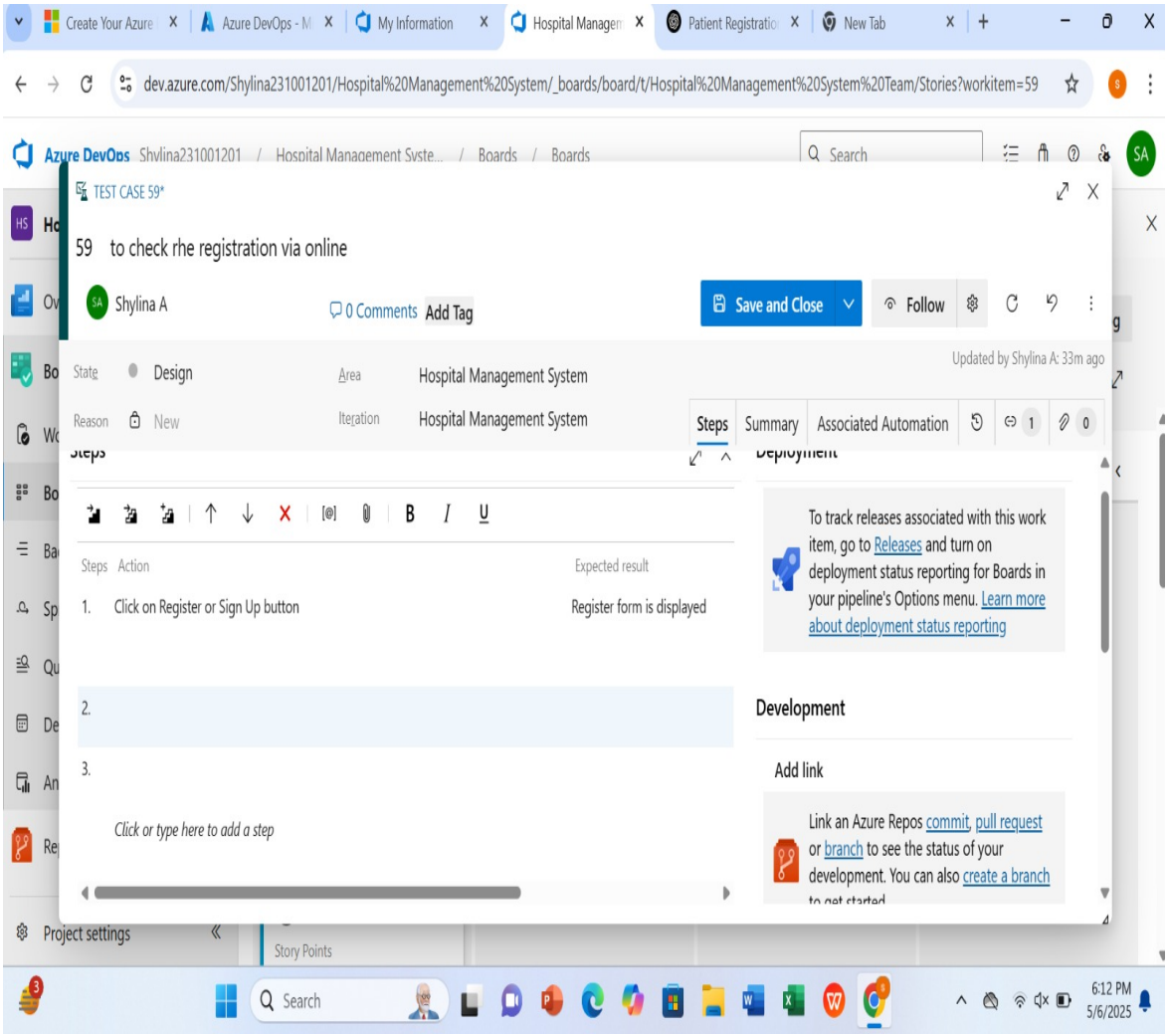
8. Prioritize and Review

- Mark test cases with priority (High, Medium, Low).
- Review test cases for completeness and correctness.
- Ensure alignment with associated user stories or features.

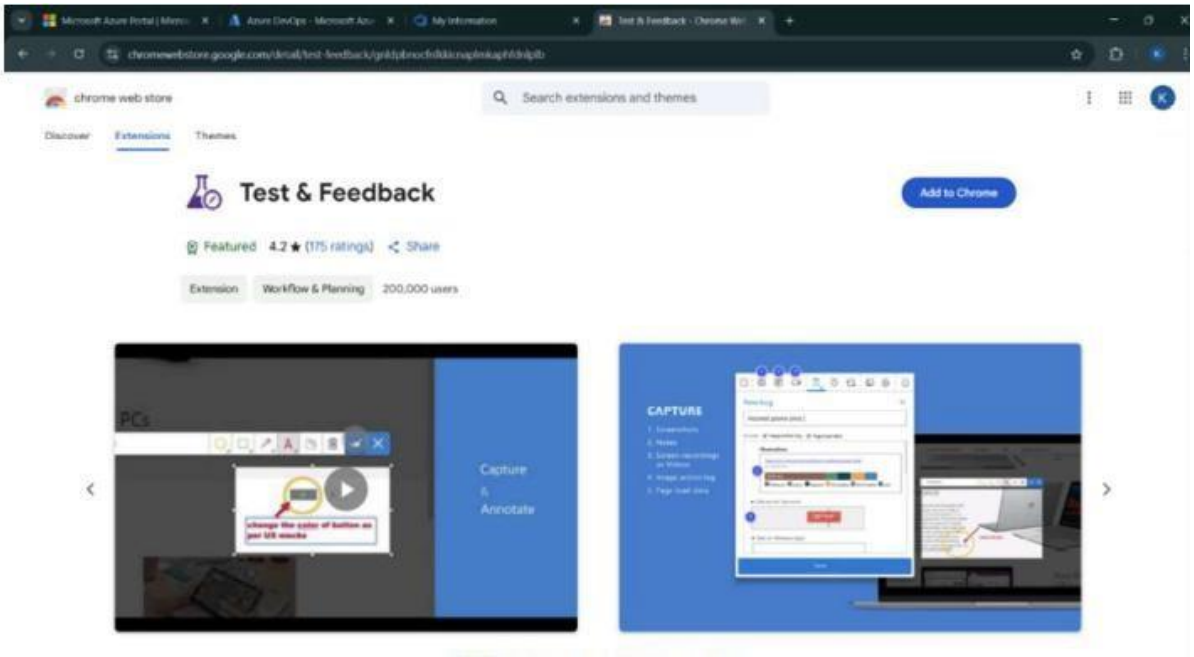
1. **New test plan**



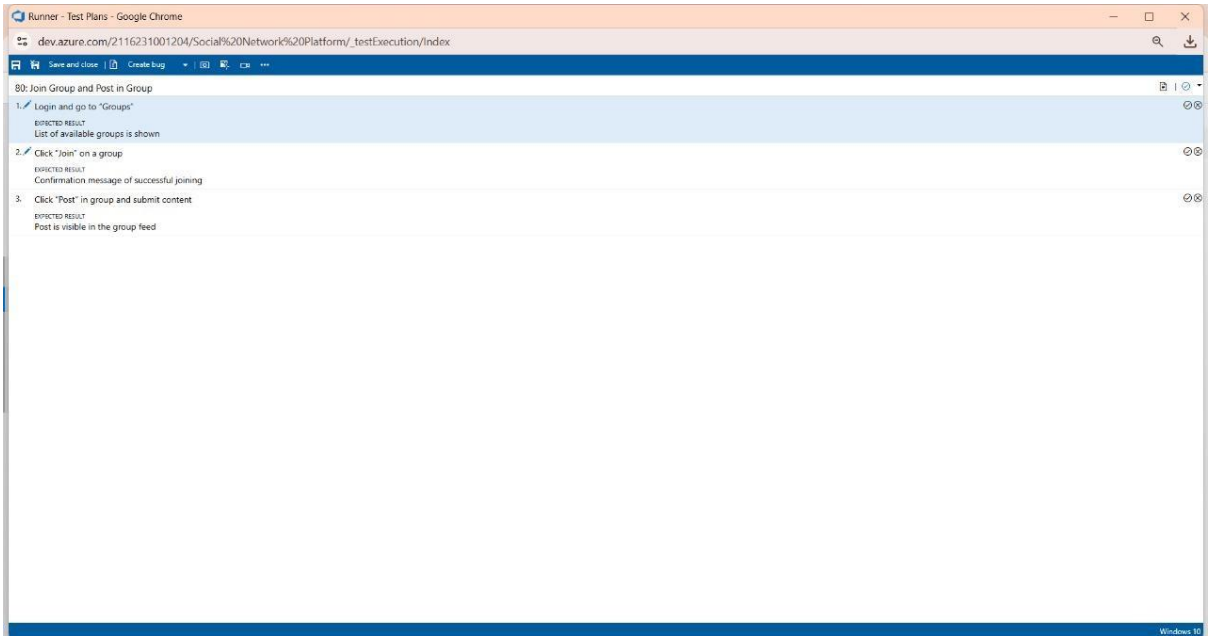
2. Test case



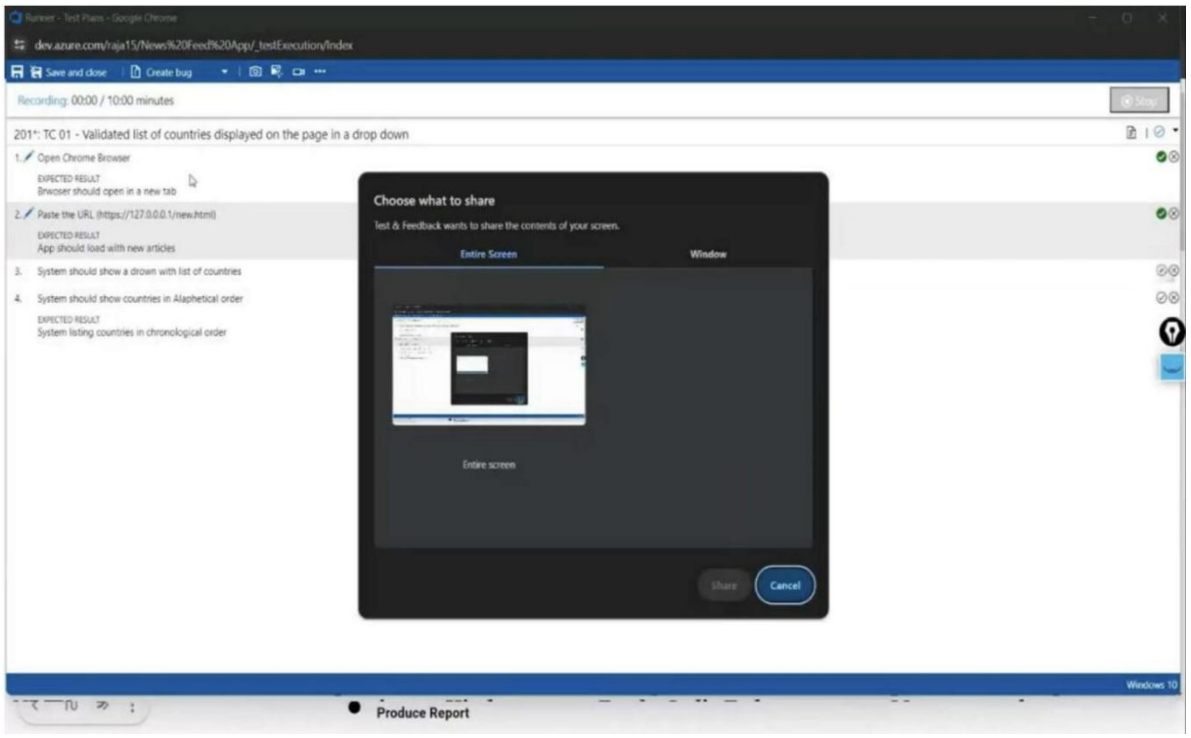
3. Installation of Test


chrome web store
Search extensions and themes
Discover
Extensions
Themes
Test & Feedback
Add to Chrome
Featured
4.2 ★ (175 ratings)
Share
Extension
Workflow & Planning
200,000 users
PCs
Capture
&
Annotate
CAPTURE

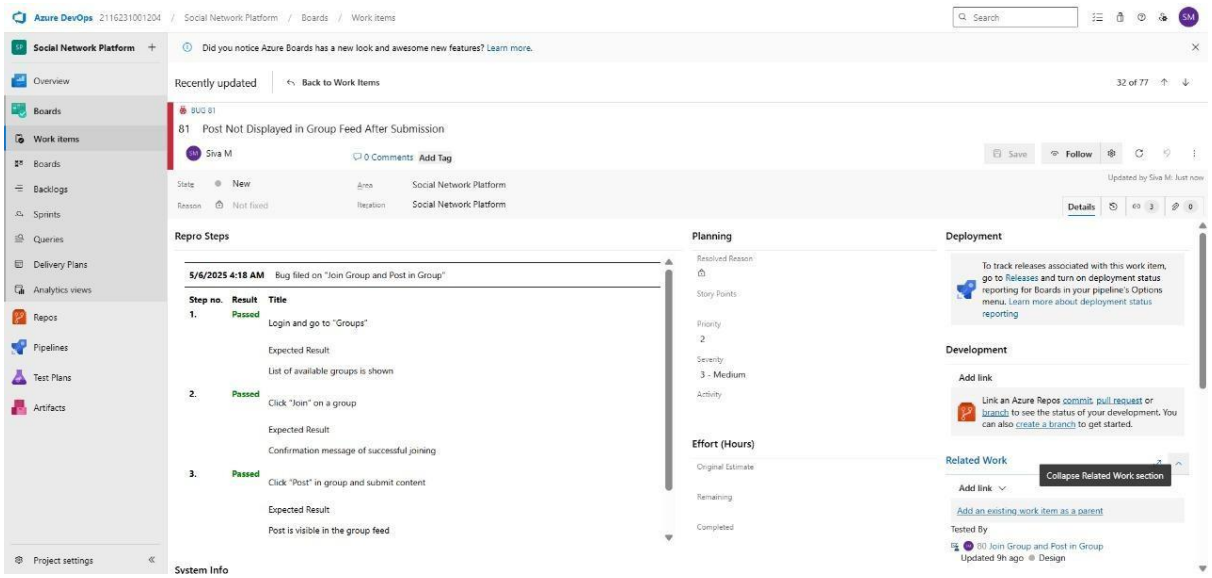
4. Running the Test Cases



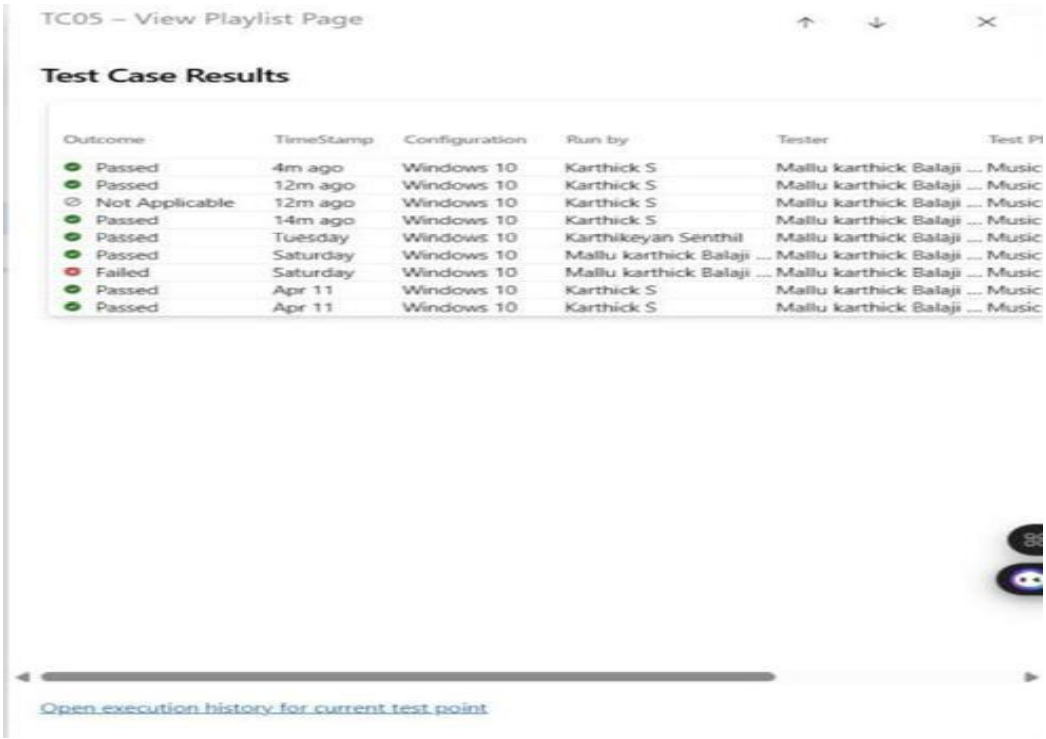
5. Recording the Test Cases



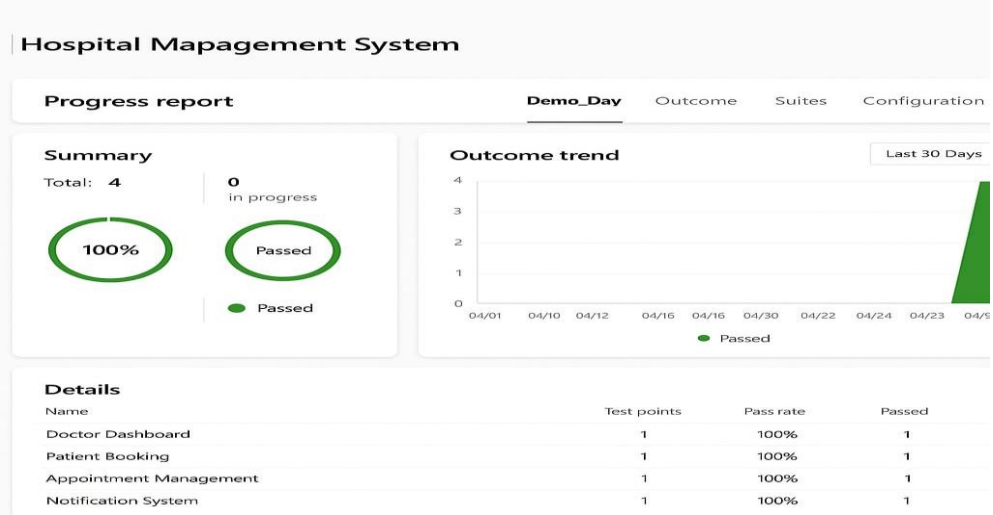
6. Creating Bugs



7. Test Case Results



8. Progress Report



LOAD TESTING PROCEDURE :

Steps to Create an Azure Load Testing Resource:

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

1. Sign in to Azure Portal

   Go to https://portal.azure.com and log in.

2. Create the Resource

- Go to Create a resource — Search for “Azure Load Testing”.
- Select Azure Load Testing and click Create.

3. Fill in the Configuration Details

- Subscription: Choose your Azure subscription.
- Resource Group: Create new or select an existing one.
- Name: Provide a unique name (no special characters).
- Location: Choose the region for hosting the resource.

4. (Optional) Configure tags for categorization and billing.

5. Click Review + Create, then Create.

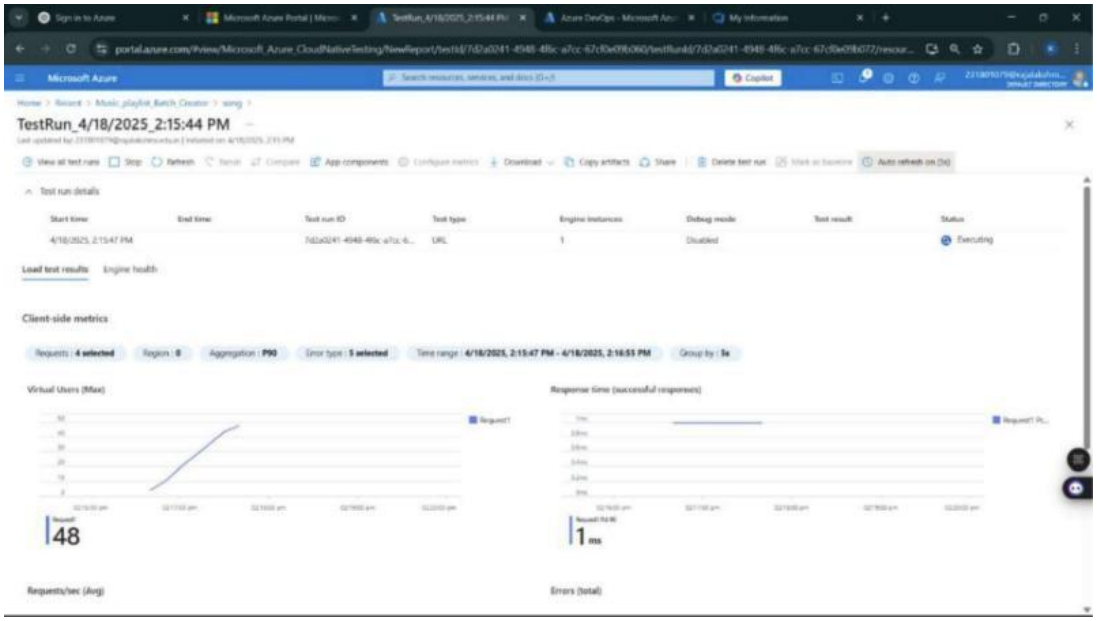
6. Once deployment is complete, click Go to resource.

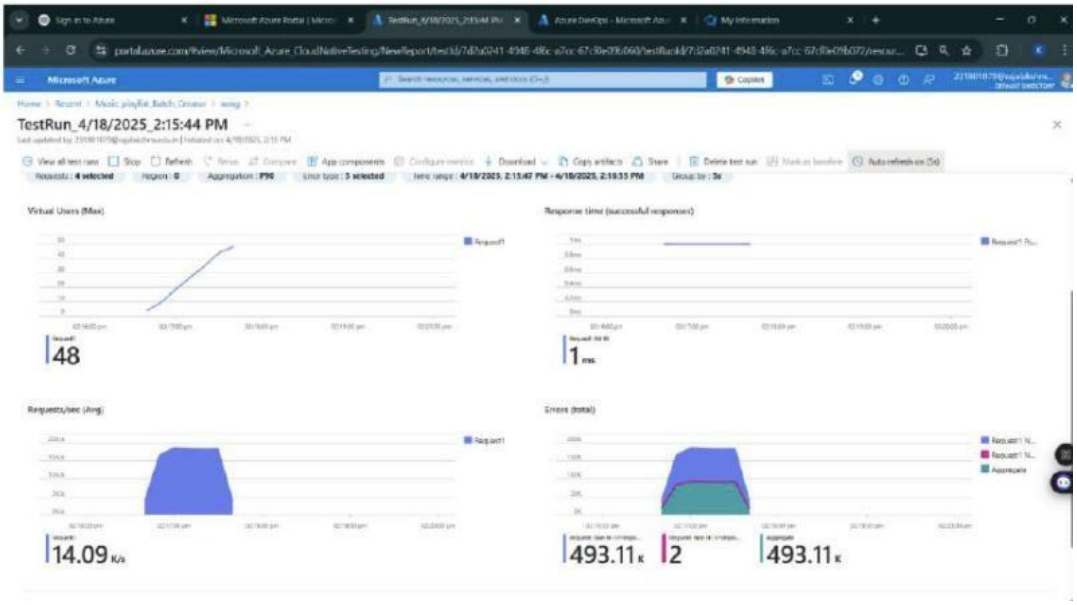
Steps to Create and Run a Load Test:

Once your resource is ready:

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

Load Testing





RESULT:

Test plans and test cases for selected user stories were created in Azure DevOps, covering both happy and error paths and an Azure Load Testing resource was also set up, and a load test was successfully run to evaluate the performance of the target endpoint.