

RAJALAKSHMI ENGINEERING COLLEGE
RAJALAKSHMI NAGAR, THANDALAM - 602 105



RAJALAKSHMI
ENGINEERING COLLEGE

CS23432

SOFTWARE CONSTRUCTION

Laboratory Record Note Book

Name :

Year / Branch / Section :

Register No. :

Semester :

Academic Year :



RAJALAKSHMI ENGINEERING COLLEGE (AUTONOMOUS)

RAJALAKSHMI NAGAR, THANDALAM – 602 105

BONAFIDE CERTIFICATE

NAME _____ REGISTER NO. _____

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		

1.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.

EX NO: 2

PROBLEM STATEMENT

AIM:

To prepare PROBLEM STATEMENT for Student data management system.

Problem Statement:

Student Data Management System

Managing student data manually is time-consuming, error-prone, and inefficient, especially as the number of students increases. Institutions face difficulties in maintaining accurate records of student profiles, enrollments, course details, performance tracking, and document storage. A centralized Student Data Management System is needed to automate these processes, improve data accuracy, ensure secure access, and make information easily available to students, faculty, and administrators.

Result:

The problem statement was written successfully.

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.

CREATE USER STORIES

Aim:

To create User Stories

THEORY

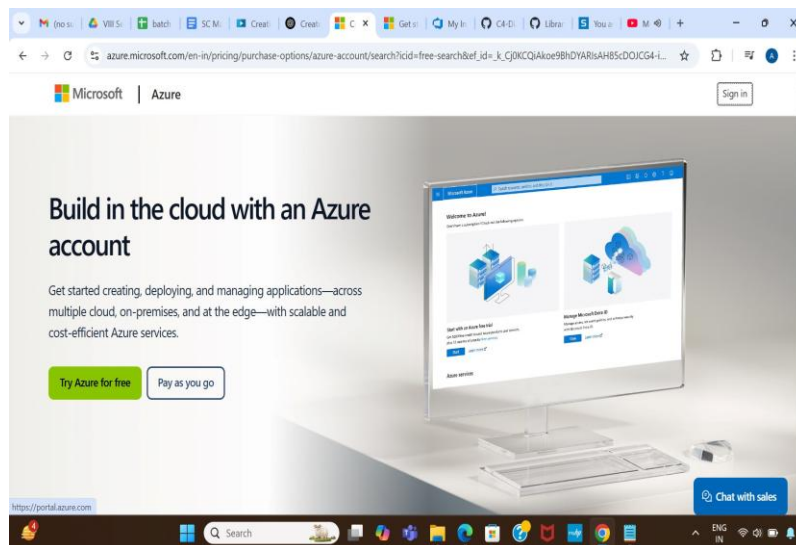
A user story is an informal, general explanation of a software feature written from the perspective of the end user. Its purpose is to articulate how a software feature will provide value to the customer.

User story template

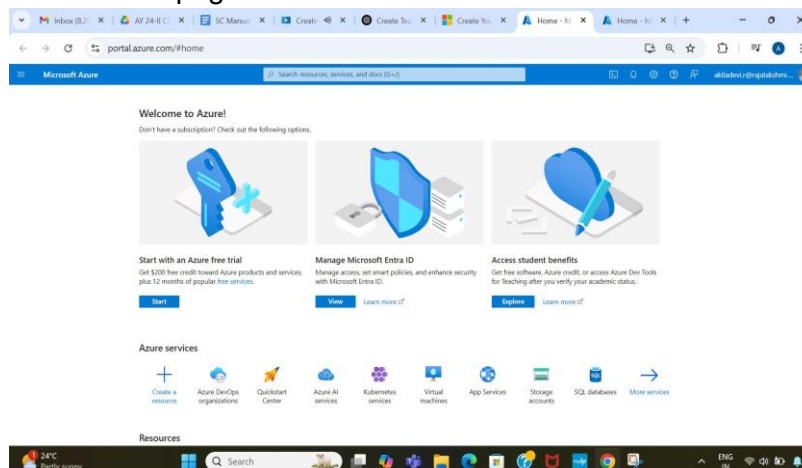
"As a [role], I [want to], [so that]."

Procedure:

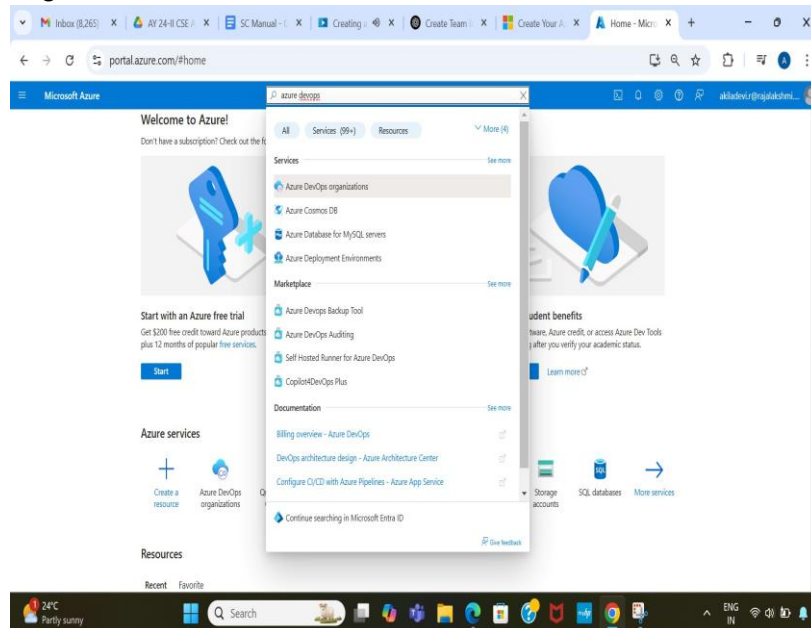
1. Open your web browser and go to the Azure website:
<https://azure.microsoft.com/en-in> Sign in using your Microsoft account credentials. If you don't have an account, you'll need to create one.
2. If you don't have a Microsoft account, you can sign up for
<https://signup.live.com/?lic=1>



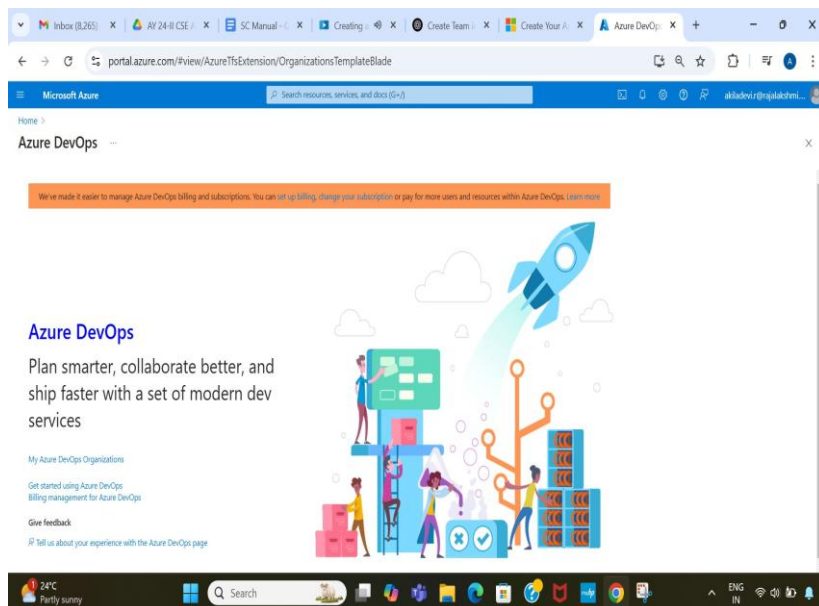
3. Azure home page

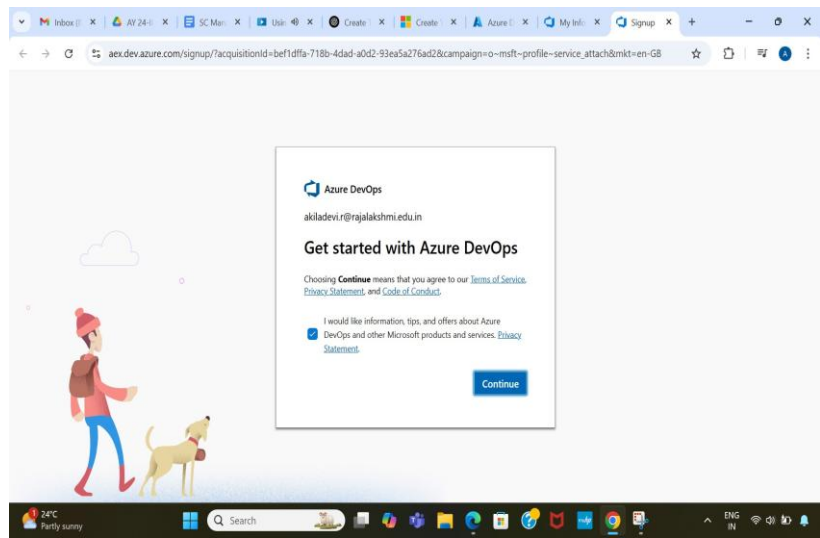


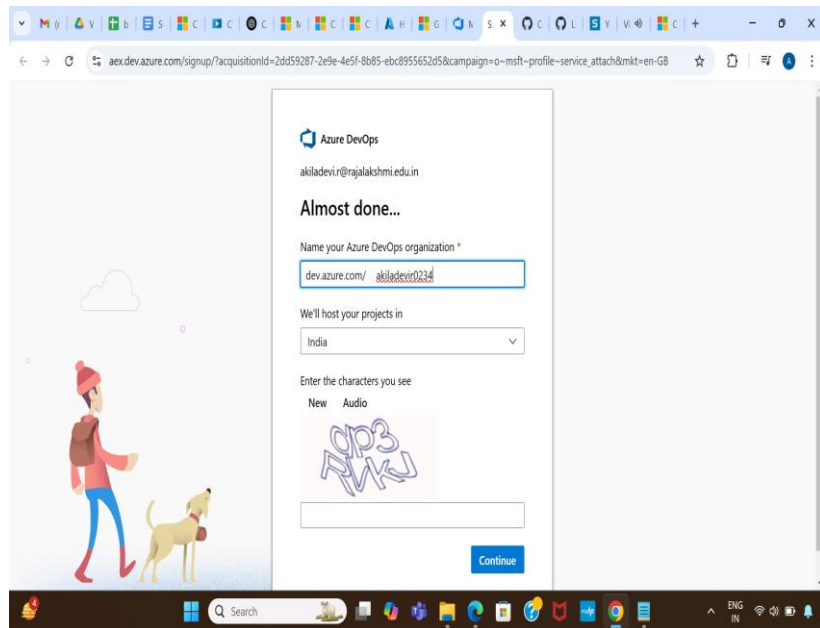
4. Open DevOps environment in the Azure platform by typing Azure DevOps Organizations in the search bar.



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.





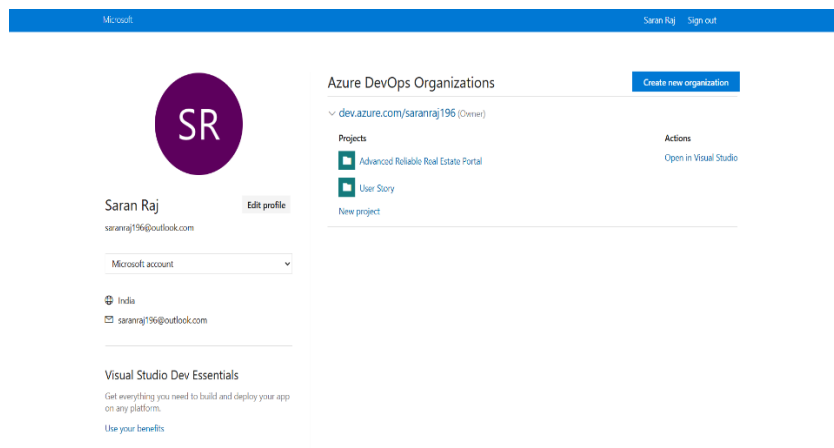


6. Create the First Project in Your Organization

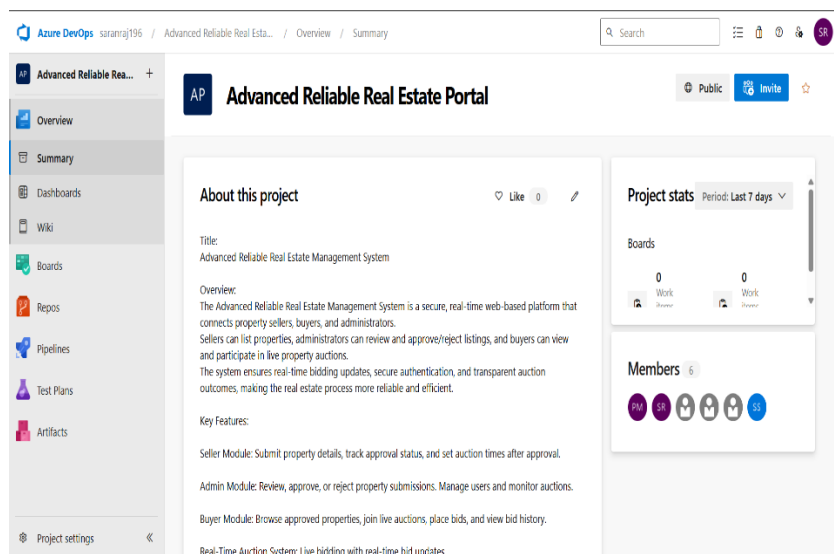
After the organization is set up, you'll need to create your first **project**. This is where you'll begin to manage code, pipelines, work items, and more.

- i. On the organization's **Home page**, click on the **New Project** button.
- ii. Enter the project name, description, and visibility options:
 - o **Name**: Choose a name for the project (e.g., **LMS**).
 - o **Description**: Optionally, add a description to provide more context about the project.
 - o **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.

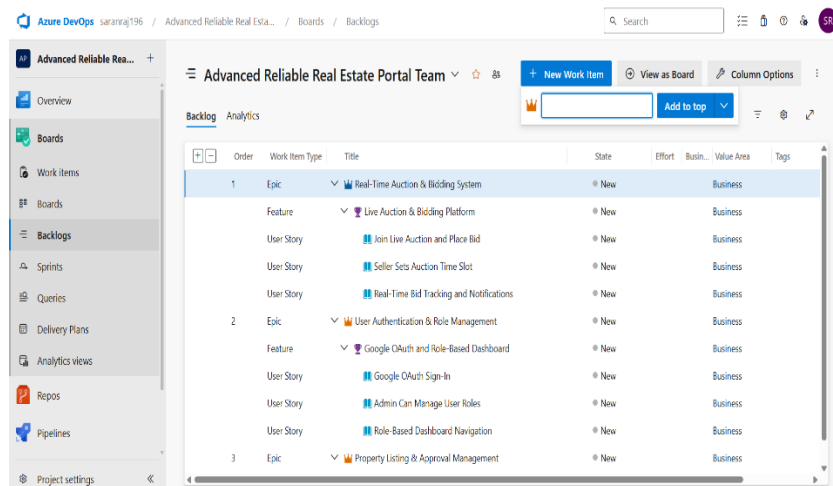


8. Project dashboard



9. To manage user stories

- 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.



10. Fill in User Story Details

USER STORY 24

24 Join Live Auction and Place Bid

samhenry2223@outlook.com

0 Comments

Add Tag

Save and Close

Follow

State: New

Area: Advanced Reliable Real Estate Portal

Updated by Saran Raj 5m ago

Reason: New

Iteration: Advanced Reliable Real Estate Portal/Iteration 1

Details

1

Description

Allow buyers to enter auctions and place bids on active properties

Acceptance Criteria

- Join button visible on auction start
- User can place bid above current
- Real-time updates on new bids

Planning

Story Points

5

Priority

1

Risk

Deployment

Development

Related Work

Add link

Parent

23 Live Auction & Bidding Platform

Updated Tuesday

New

Classification

Value area

Business

Result:

The user story was written successfully.

EX NO: 5

SEQUENCE DIAGRAM

Aim:

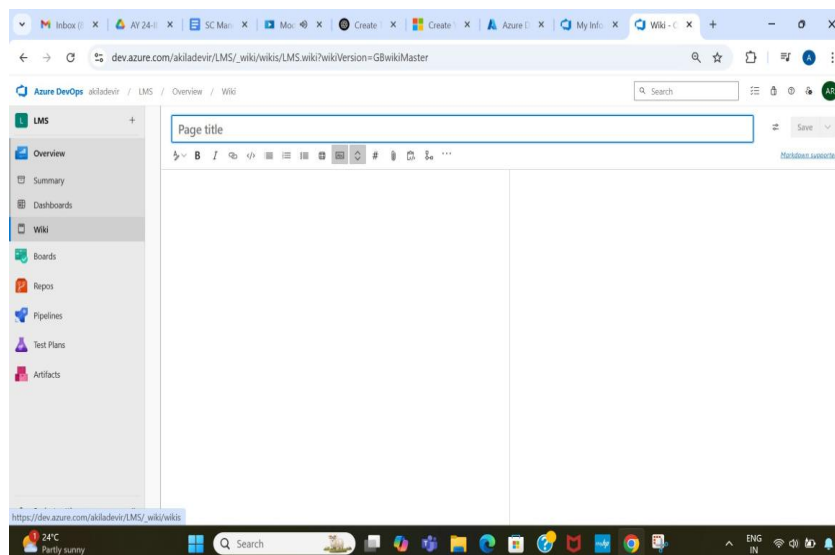
To design a Sequence Diagram by using Mermaid.js

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 Instructor
 participant System
 participant StudentDatabase
 participant CourseDatabase

Instructor->>>System: Login

System->>>Instructor: Authenticate & Show Dashboard

Instructor->>>System: Select Course & Student

System->>>CourseDatabase: Fetch Course Details

System->>>StudentDatabase: Fetch Student Record

Instructor->>System: Enter Grade
 System->>StudentDatabase: Save Grade
 StudentDatabase-->>System: Confirm Grade Saved

 System-->>Instructor: Grade Submission Success::

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+)

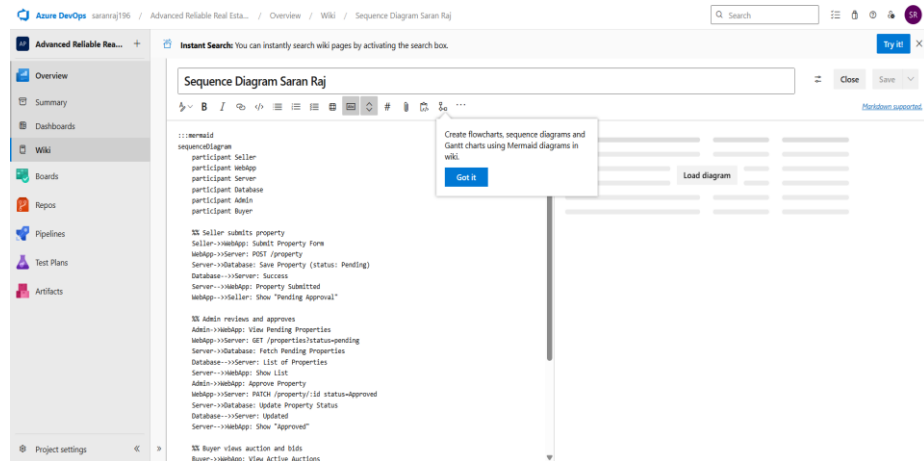
<<->> Dotted line with bidirectional arrowheads (v11.0.0+)

-x Solid line with a cross at the end

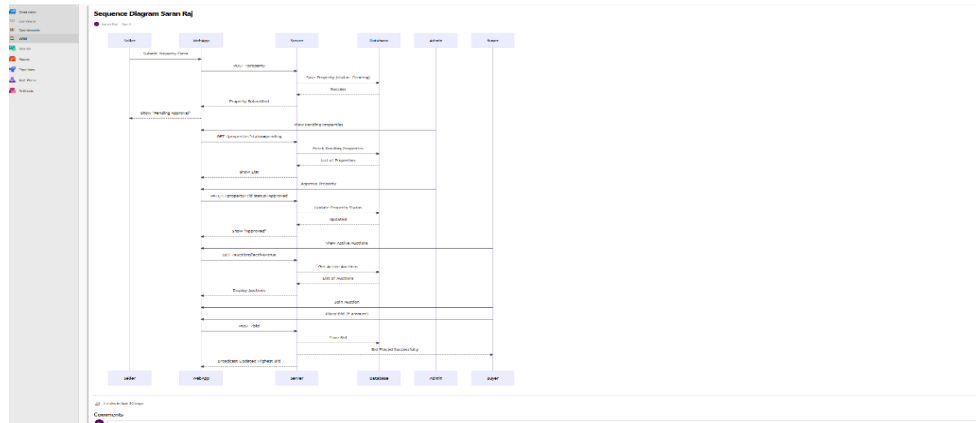
--x Dotted line with a cross at the end

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

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



4. click wiki menu and select the page



Result:

The sequence diagram was drawn successfully.

EX NO. 6

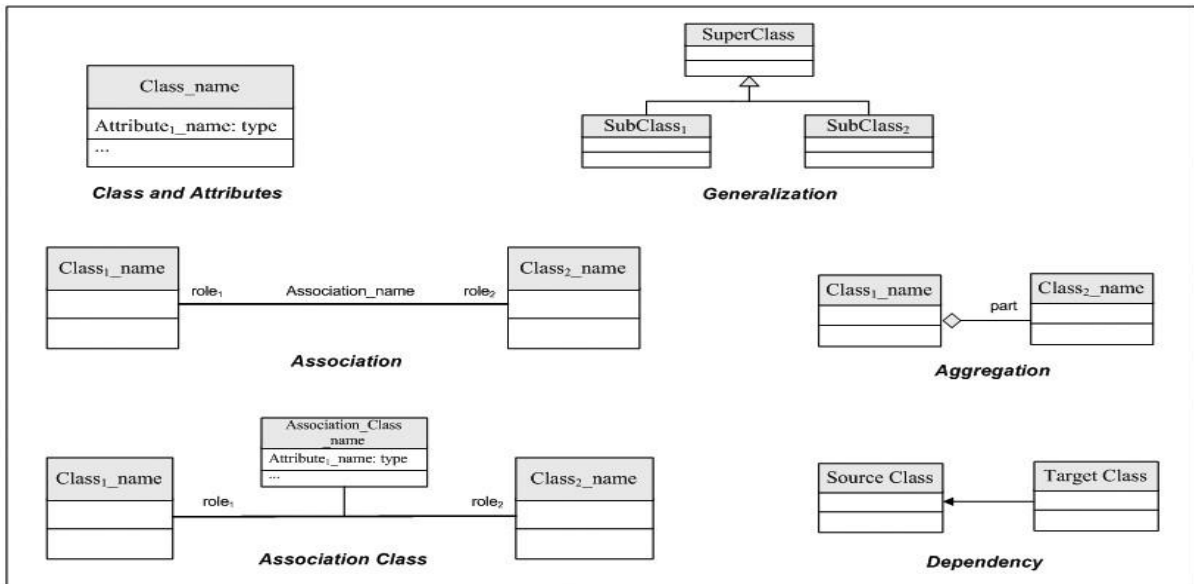
CLASS DIAGRAM

AIM :-

To draw a sample class diagram for Student Management System.

THEORY :

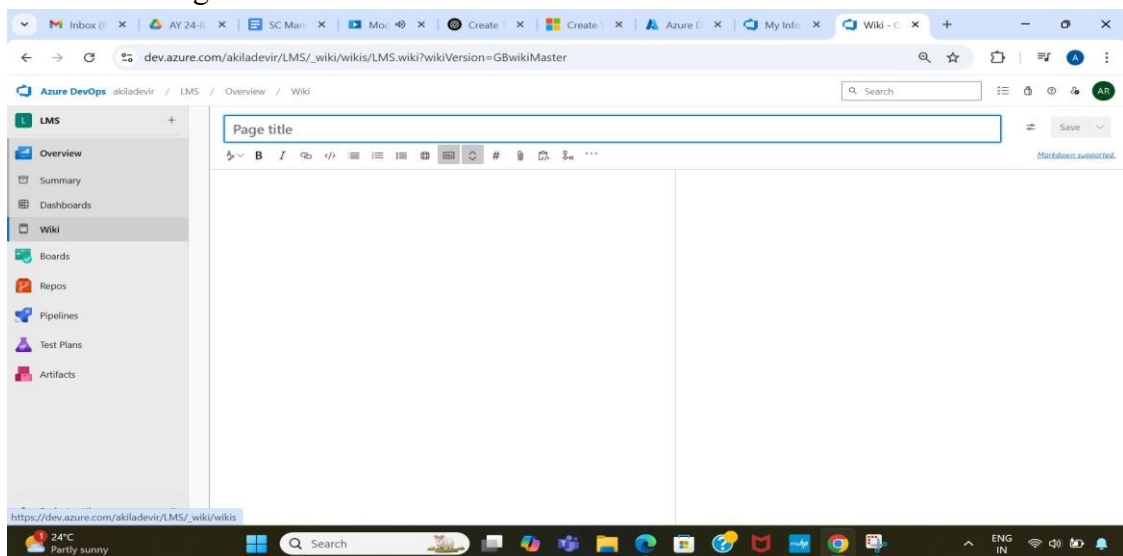
A UML class diagram is a visual tool that represents the structure of a system by showing its classes, attributes, methods, and the relationships between them.



Notations in class diagram

Procedure:

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



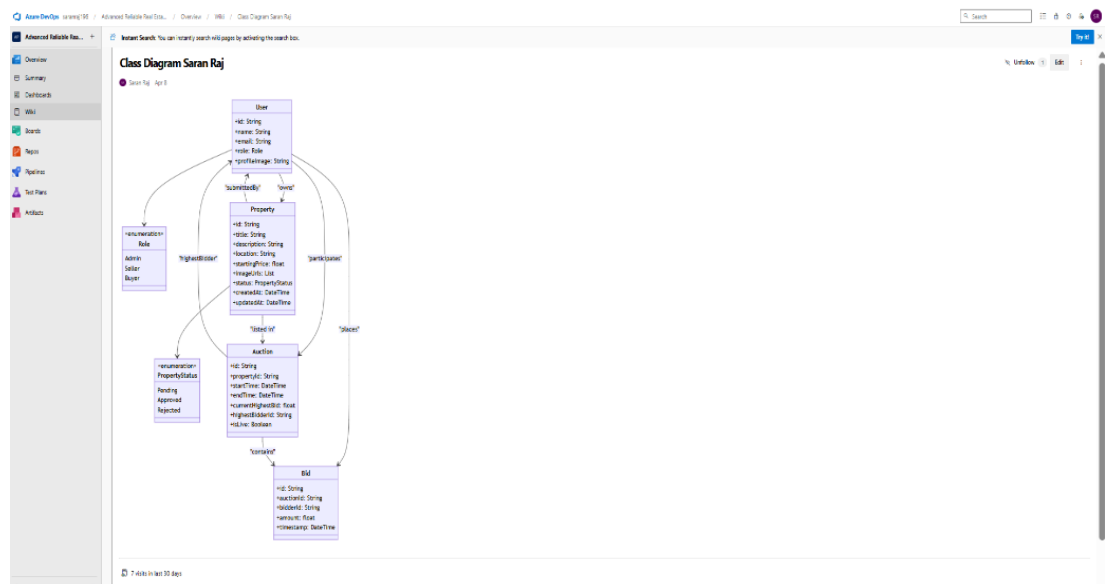
3. Write code for drawing class diagram and save the code
- ```
Class Diagram
class Student {
 +string student Id
```

```

+string name
+Date dob
+string email
}
class Course{
+string course Id
+string title
+int credits
}
class Instructor {
+string instructorId
+string name
+string email
}
class User {
+string userId
+string username
+string password
+string role
}

%% Relationships
Student "1" --> "many" Grade
Course "1" --> "many" Grade
Student "many" --> "many" Course
Course "1" --> "1" Instructor
Student --|> User
Instructor --|> User

```



## Result:

The use case diagram was designed successfully.

**EX NO: 7**

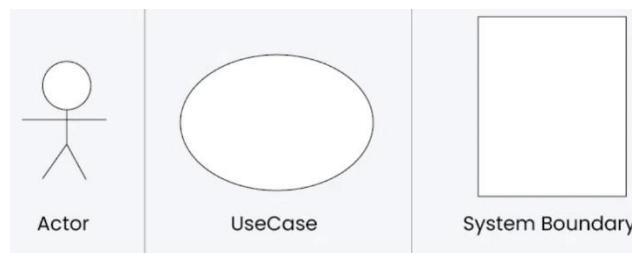
## **USECASE DIAGRAM**

**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.

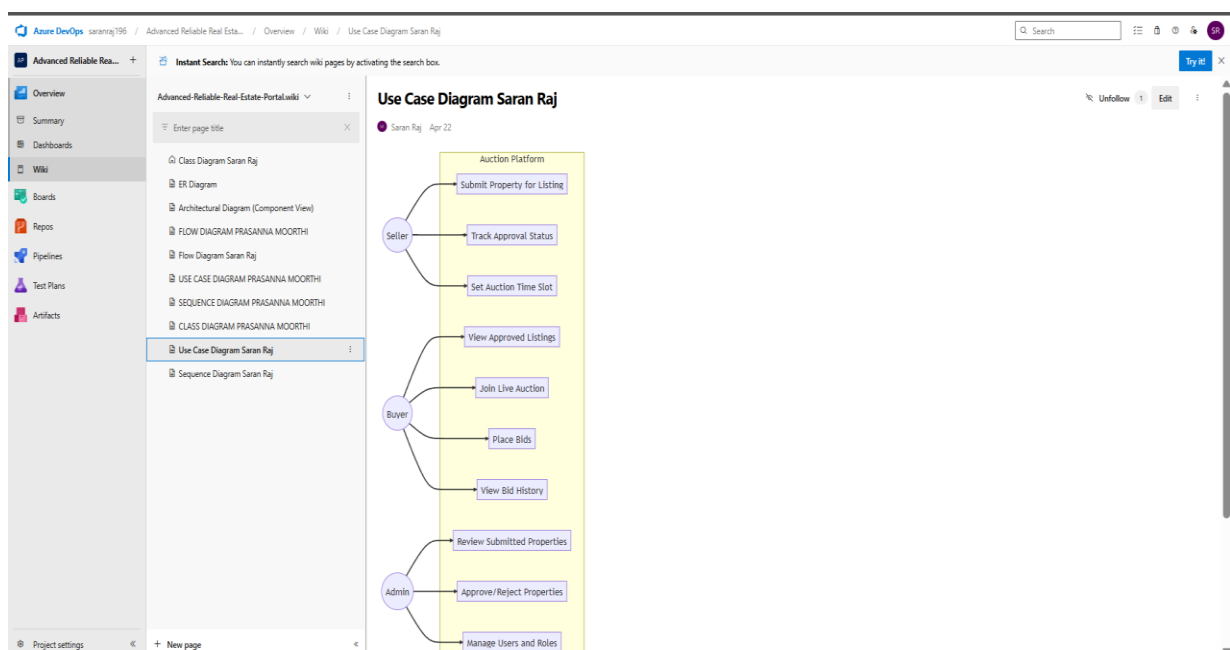
Step 2: Upload the Diagram to Azure DevOps

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

## EX NO. 8



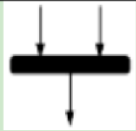


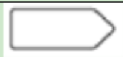





### ACTIVITY DIAGRAM

#### AIM :-

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

#### THEORY

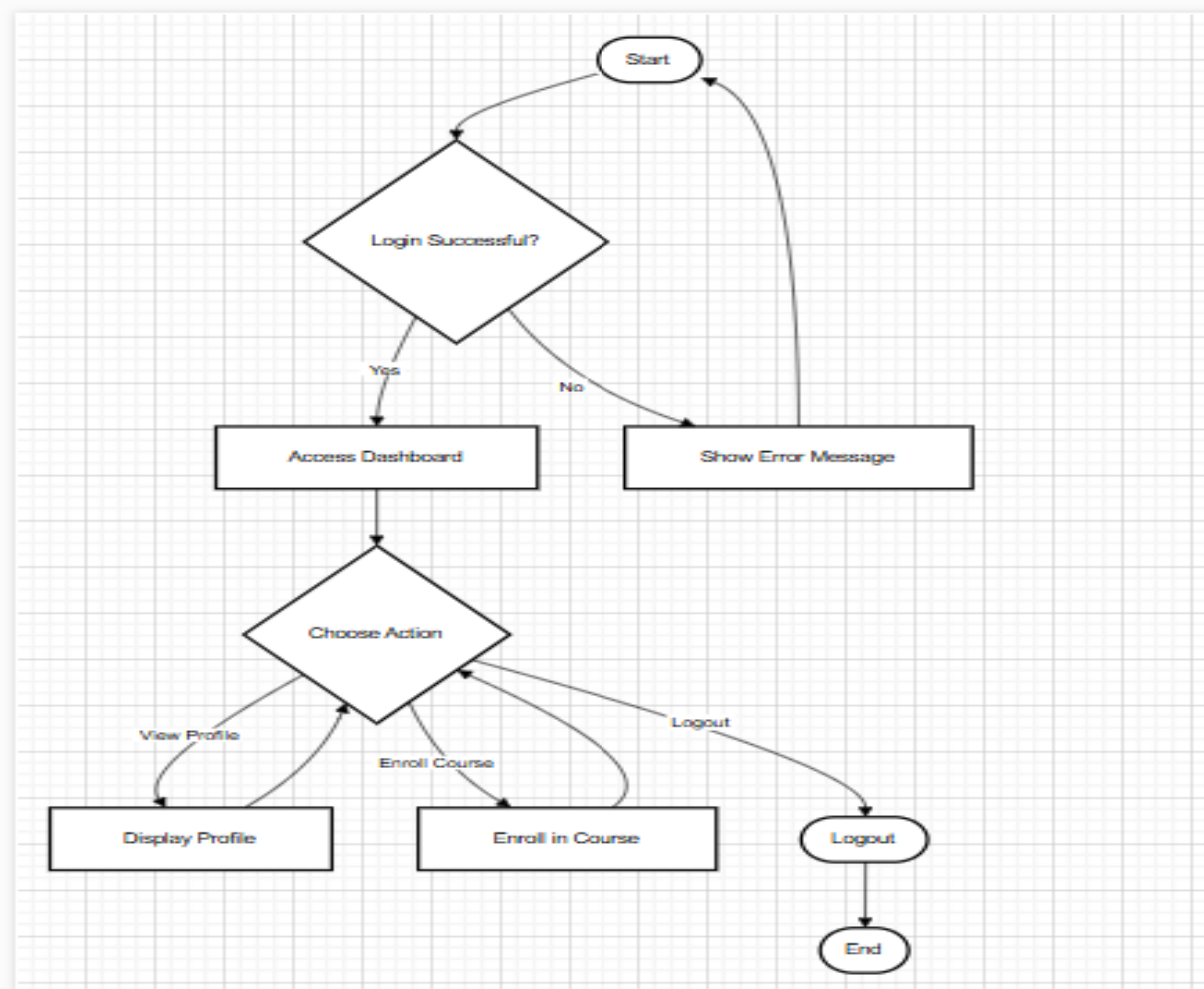
Activity diagrams are an essential part of the Unified Modelling Language (UML) that help visualize workflows, processes, or activities within a system. They depict how different

| 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 to 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              |

actions are connected and how a system moves from one state to another.

#### Procedure

1. Draw diagram in draw.io
2. Upload the diagram in Azure DevOps wiki



**Result:**  
The activity diagram was designed successfully

## EX NO. 9

### ARCHITECTURE DIAGRAM

#### 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.



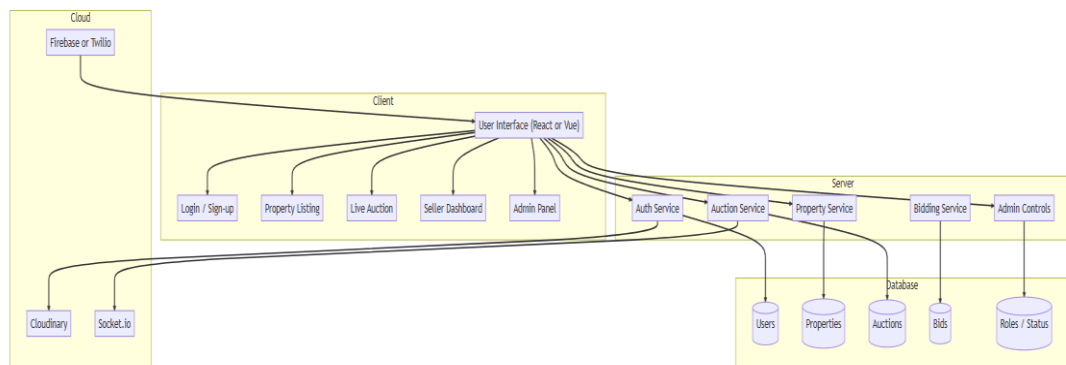
#### Procedure

1. Draw diagram in draw.io
2. Upload the diagram in Azure DevOps wiki

## Architectural Diagram (Component View)

Unfollow 1 Edit

Saran Raj Just now



0 visits in last 30 days

### Comments

Saran Raj Just now

### Result:

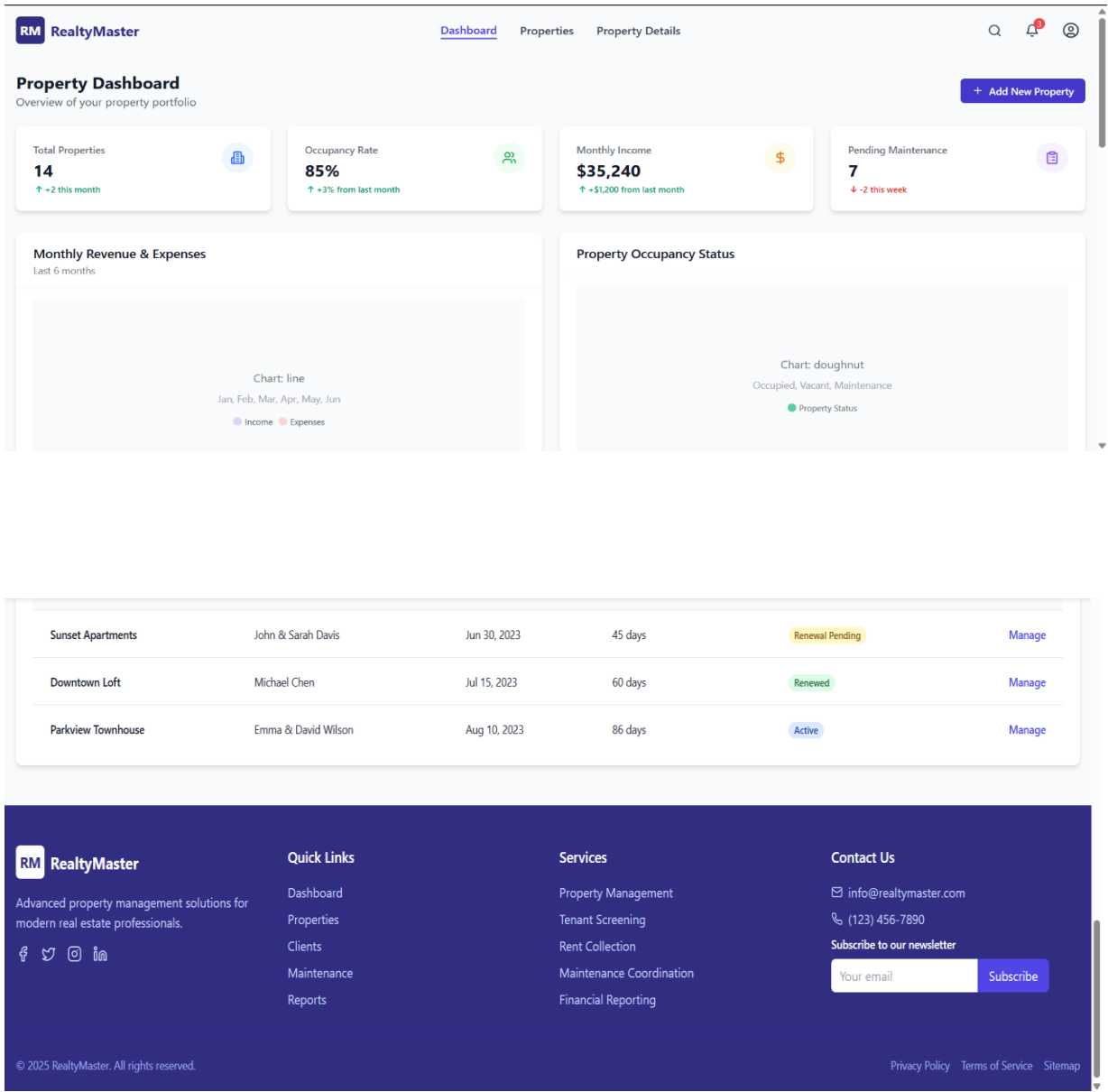
The architecture diagram was designed successfully



EX NO. 10

USER INTERFACE

Aim: Design User Interface for the Student data Management System



RM

RealtyMaster

Dashboard

Properties

Property Details


Properties

Manage your property portfolio

+ Add New Property

Search properties...

Filters



Occupied

Sunset Apartments


123 Sunset Blvd, Los Angeles, CA 90210

2 bd2 ba1,200 sqft

Apartment

Edit

Delete



Vacant

Ocean View Condo


456 Ocean Ave, Miami, FL 33139

3 bd2 ba1,800 sqft

Condo

Edit

Delete



Maintenance

Parkview Townhouse

789 Park Ave, Chicago, IL 60611

3 bd2.5 ba2,000 sqft

Townhouse

Edit

Delete

Overview

Financials

Maintenance

Tenants

Documents

Calendar

Property Details

Property Name

Sunset Apartments

Address

123 Sunset Blvd, Los Angeles, CA 90210

Bedrooms

2

Monthly Rent

\$2,500

Description

Modern apartment complex with amenities including a pool, gym, and covered parking.

Property Type

Apartment

Square Footage

1,200 sqft

Bathrooms

2

Status

Occupied

Financial Summary

Monthly Rent

\$2500

Mortgage

-\$1500

Insurance

-\$150

Property Taxes

-\$200

Maintenance

-\$100

Management

-\$200

Monthly Cash Flow

\$350

View Full Financials

Maintenance Summary

Leaky faucet in bathroom

Due 10/10/22Completed

Features & Amenities

Result:

The UI was designed successfully.

## EX NO. 11

### IMPLEMENTATION

#### **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 : 12.04.2025**

## **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

Azure DevOps saranya196 / Advanced Reliable Real Esta... / Test Plans

Search

Advanced Reliable Rea... +

Overview

Boards

Repos

Pipelines

Test Plans

Test plans

Progress report

Parameters

Configurations

Runs

Artifacts

Project settings

New Test Plan

Name \*

Advanced Reliable Real Estate Portal-Test Plan

Area Path \*

Advanced Reliable Real Estate Portal

Iteration \*

Advanced Reliable Real Estate Portal/Iteration 1

Create Cancel

### 2. Test case

TEST CASE 80

80 Join Group and Post in Group

Siva M

0 Comments Add Tag

Save and Close Follow

Updated by Siva M: 8h ago

Steps Summary Associated Automation

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

Tests

81 Post Not Displayed in Group Feed After Submission

Updated 8h ago Resolved

Status

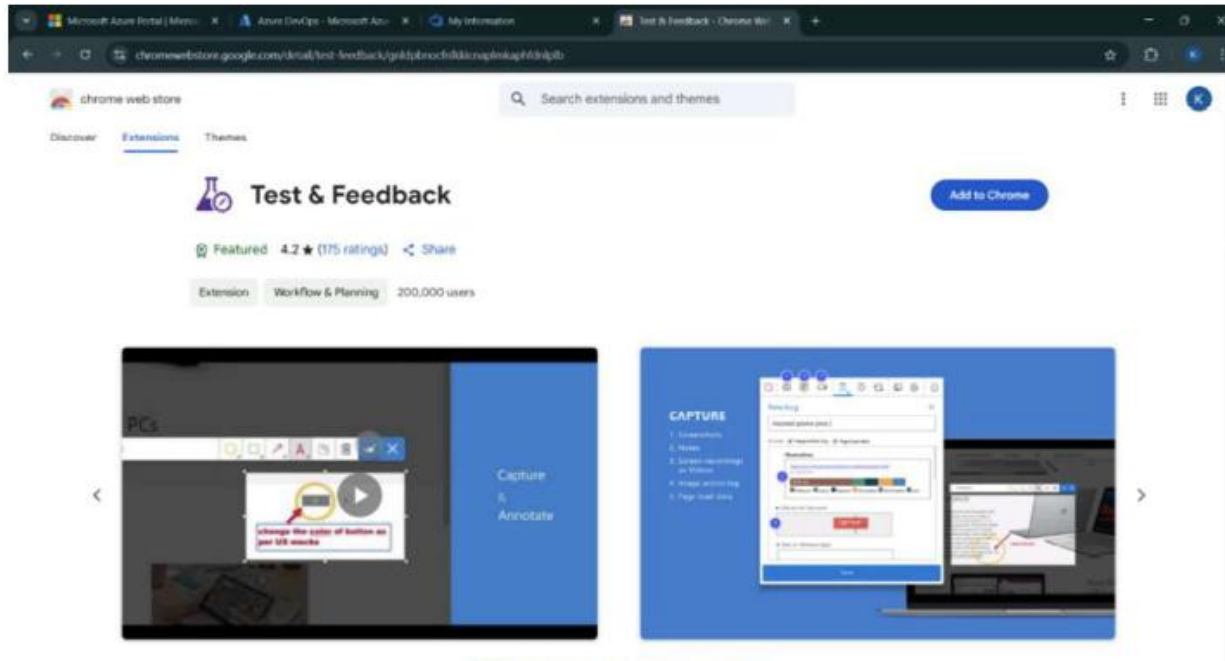
Priority

Parameter values

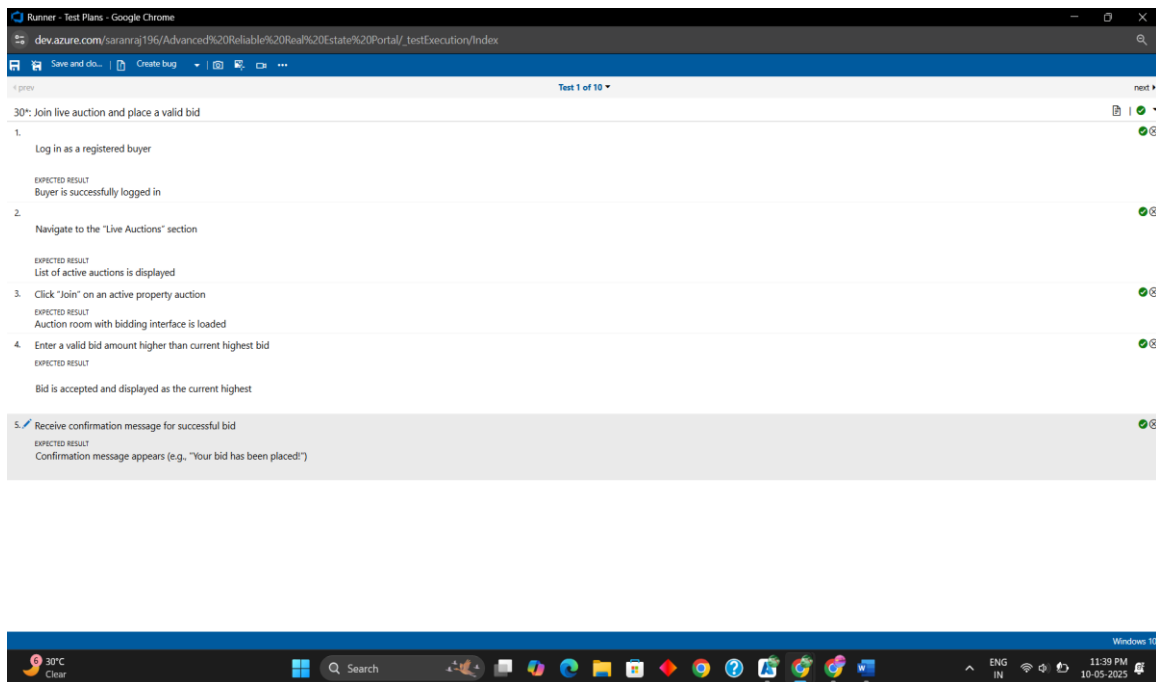
| Steps | Action                                   | Expected result                            | Attachments |
|-------|------------------------------------------|--------------------------------------------|-------------|
| 1.    | Login and go to "Groups"                 | List of available groups is shown          |             |
| 2.    | Click "Join" on a group                  | Confirmation message of successful joining |             |
| 3.    | Click "Post" in group and submit content | Post is visible in the group feed          |             |

Click or type here to add a step

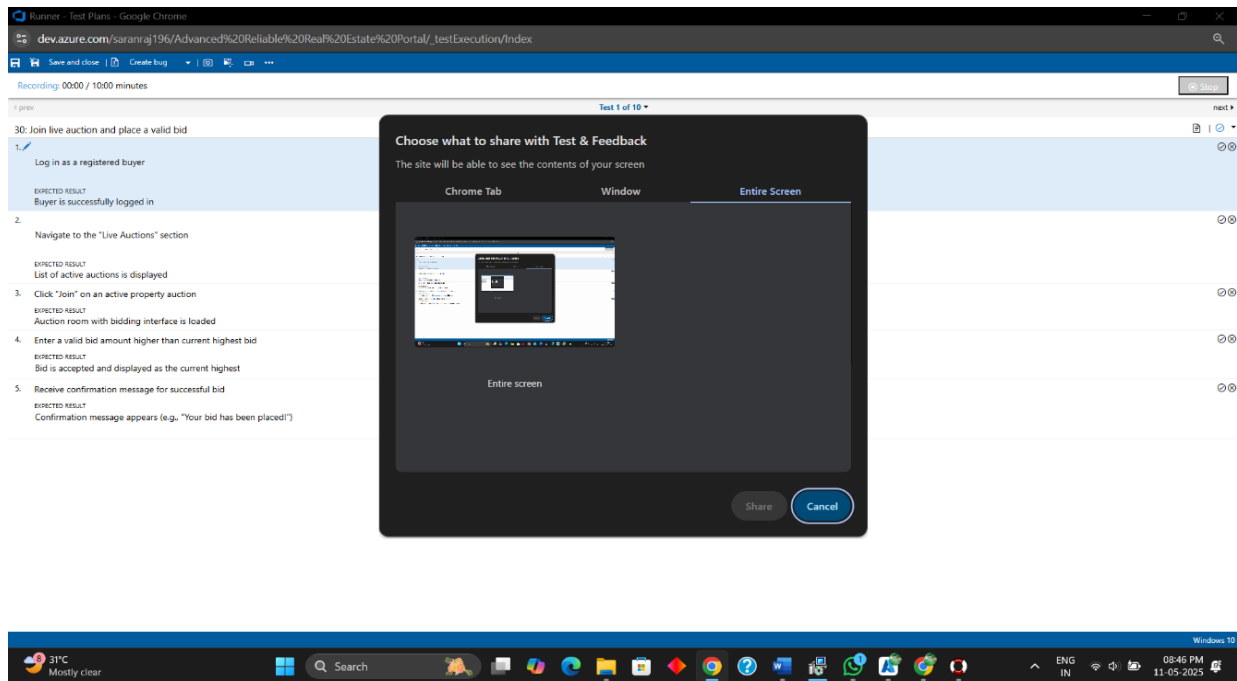
### 3. Installation of Test



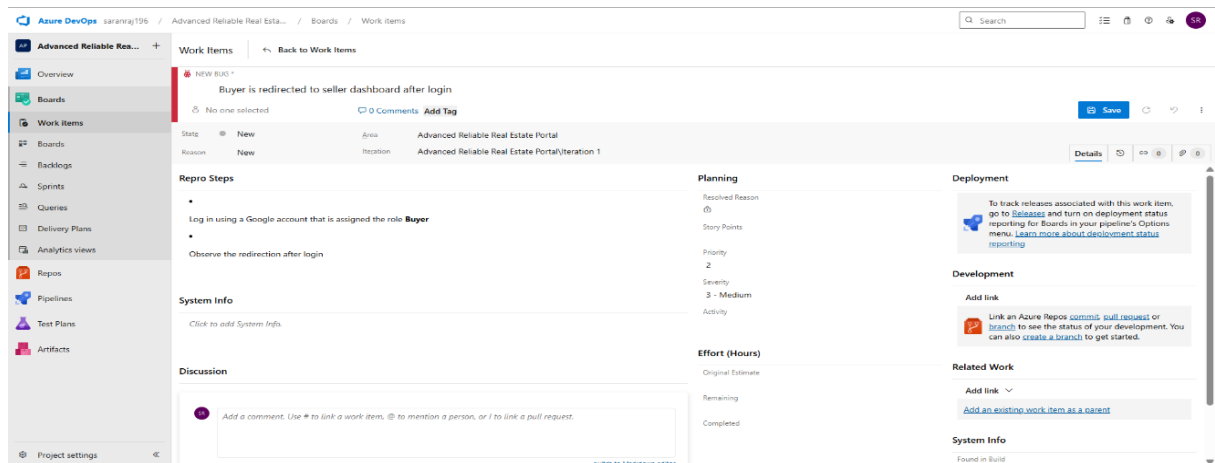
### 4. Running the Test Cases



## 5. Recording the Test Cases



## 6. Creating Bugs





## 7. Test Case Results

Azure DevOps sarani196 / Advanced Reliable Real Esta... / Test Plans / Runs

Search

Advanced Reliable Real Esta... +

Overview  
Boards  
Repos  
Pipelines  
Test Plans  
Test plans  
Progress report  
Parameters  
Configurations  
Runs  
Artifacts

Enter Run ID... Go

Recent test runs

Test runs

3: Advanced Reliable Real Esta...

Recent exploratory sessions

Run 3 - Advanced Reliable Real Estate Portal-Test Plan (Manual)

Run summary **Test results** Filter

Create bug Update analysis

| Outcome | Test Case Title                                                          | Priority | Duration    | Owner     | Configuration | Machine name | Error message |
|---------|--------------------------------------------------------------------------|----------|-------------|-----------|---------------|--------------|---------------|
| Passed  | Join live auction and place a valid bid                                  | 2        | 0:00:06.607 | Saran Raj | Windows 10    |              |               |
| Passed  | Set auction start and end time                                           | 2        | 0:00:36.974 | Saran Raj | Windows 10    |              |               |
| Passed  | Live Bid Update and Notification on Bid Change                           | 2        | 0:01:43.820 | Saran Raj | Windows 10    |              |               |
| Passed  | Verify Google OAuth Sign-In functionality                                | 2        | 0:01:53.797 | Saran Raj | Windows 10    |              |               |
| Passed  | Verify admin can assign or modify user roles                             | 2        | 0:03:08.704 | Saran Raj | Windows 10    |              |               |
| Passed  | Verify users are redirected to the correct dashboard based on their role | 2        | 0:03:12.380 | Saran Raj | Windows 10    |              |               |
| Passed  | Verify property submission by seller and admin approval workflow         | 2        | 0:03:18.630 | Saran Raj | Windows 10    |              |               |
| Passed  | Verify admin can approve or reject submitted properties                  | 2        | 0:03:22.817 | Saran Raj | Windows 10    |              |               |
| Passed  | Verify seller can track listing status of submitted properties           | 2        | 0:03:29.130 | Saran Raj | Windows 10    |              |               |
| Passed  | Verify seller can track listing status of submitted properties           | 2        | 0:03:35.207 | Saran Raj | Windows 10    |              |               |

10 results (1 selected)

## 8. Progress Report

Azure DevOps sarani196 / Advanced Reliable Real Esta... / Test Plans / Progress report

Search

Advanced Reliable Real Esta... +

Overview  
Boards  
Repos  
Pipelines  
Test Plans  
Test plans  
Progress report  
Parameters  
Configurations  
Runs  
Artifacts

Project settings

Progress report

Advanced Reliable Real Estate Portal-Test Plan

Test Suites Outcome Configuration Tester Priority Assigned To

Summary

1 Test plans

10 Test points

10 (10 / 10) Test points run

100% Run

100% (10 / 10) Pass rate

10 Passed

Outcome trend

Last 14 Days

Run

Passed

Details

| Test plan name                                   | Test points | Run % | Passed % | Failed % | Not run count |
|--------------------------------------------------|-------------|-------|----------|----------|---------------|
| > Advanced Reliable Real Estate Portal-Test Plan | 10          | 100   | 100      | 0        | 0             |

## 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:

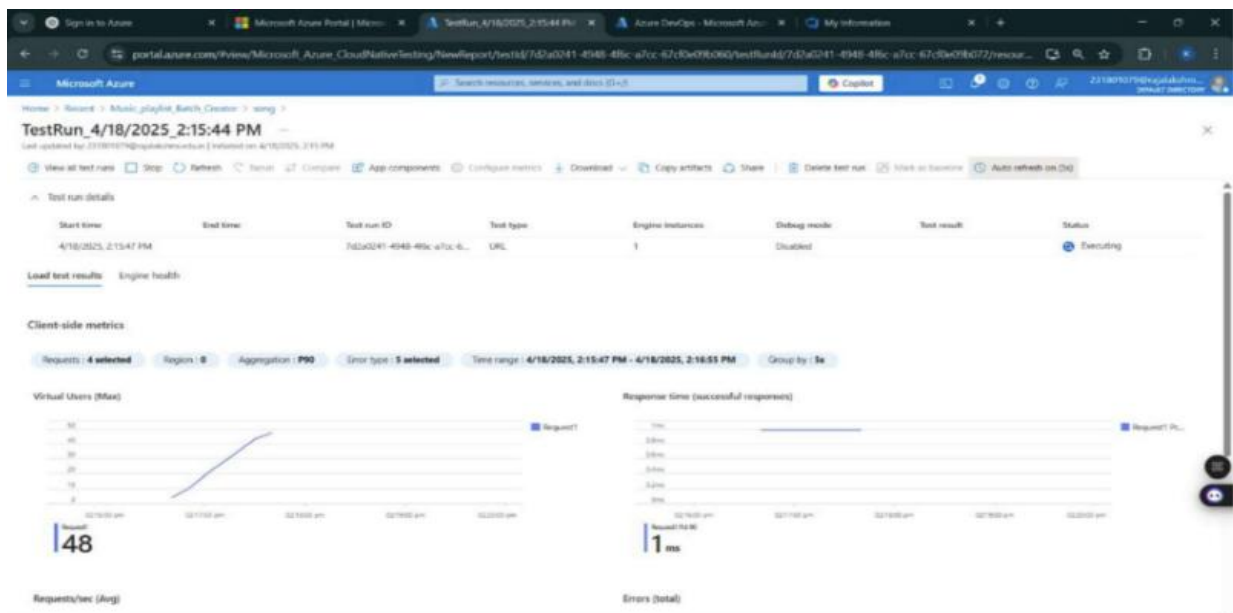
1. Sign in to Azure Portal
- Go to <https://portal.azure.com> and log in.
2. Create the Resource
    - Go to Create a resource — Search for “Azure Load Testing”.
    - Select Azure Load Testing and click Create.
  3. Fill in the Configuration Details
    - Subscription: Choose your Azure subscription.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

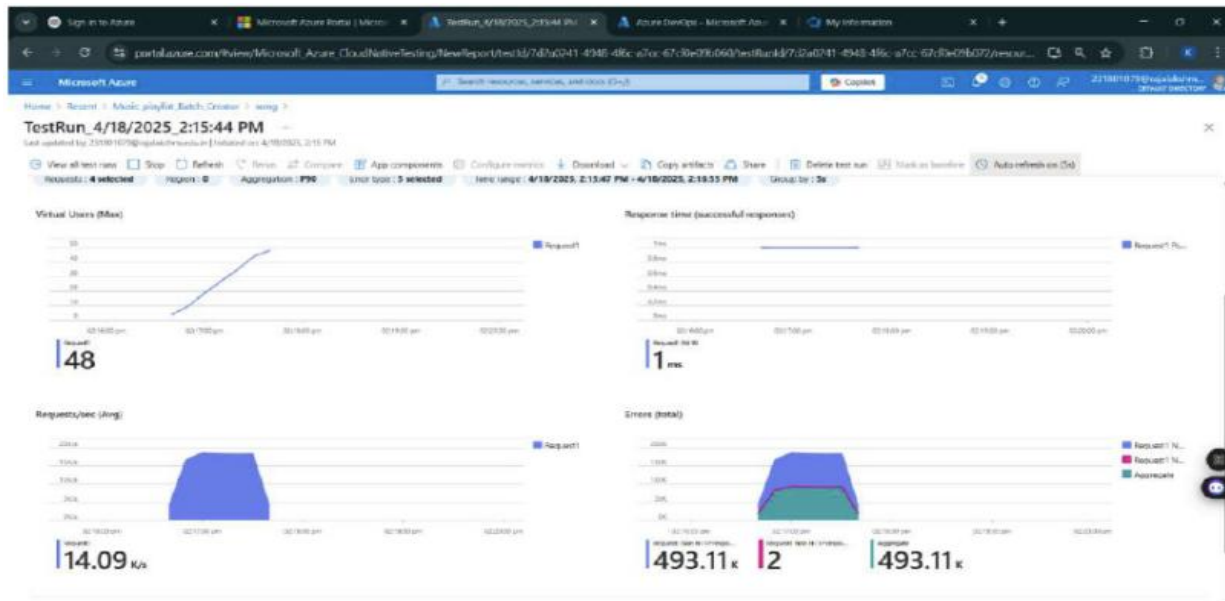
### Steps to Create and Run a Load Test:

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.