



**RAJALAKSHMI
ENGINEERING COLLEGE**
An AUTONOMOUS Institution
Affiliated to ANNA UNIVERSITY, Chennai

**DEPARTMENT OF COMPUTER SCIENCE &
ENGINEERING ACADEMIC YEAR 2024-2025**

EVEN SEMESTER



CS23432 SOFTWARE ENGINEERING LAB

LAB MANUAL

SECOND YEAR

FOURTH SEMESTER

2024-2025

EVEN SEMESTER

Ex No	List of Experiments
1	Study of Azure DevOps
2	DesigningProject using AGILE-SCRUM Methodology.
3	Agile Planning
4	User stories – Creation
5	Architecture Diagram Using AZURE
6	Designing Usecase and Class Diagram
7	Designing Interaction Diagrams
8	Design Interface
9	Implementation–DesignaWebPagebasedonScrum Methodology
10	TestingusingAzure.
11	Deployment

Requirements	
Hardware	Intel i3, CPU @ 1.20GHz 1.19 GHz, 4 GB RAM, 32 Bit Operating System
Software	StarUML , Azure

LAB PLAN

CS19442-SOFTWAREENGINEERINGLAB

Ex No	Date	Topic	Page No	Sign
1		Study of Azure DevOps		
2		Writing Problem Statement		
3		Designing Project using AGILE-SCRUM Methodology by using Azure.		
4		Agile Planning		
5		User stories – Creation		
6		Architecture Diagram Using AZURE		
7		Designing Use case Diagram using StarUML		
8		Designing Activity Diagrams using StarUML		
9		Designing Sequence Diagrams using StarUML		
10		Design Class Diagram		
10		Design User Interface		
11		Implementation–Design a Web Page based on Scrum Methodology		
12		Testing		
13		Deployment		

Course Outcomes (COs)

Course Name : Software Engineering

CourseCode:CS23432

CO 1	Understand the software development process models.
CO 2	Determine the requirements to develop software
CO 3	Apply modeling and modeling languages to design software products
CO 4	Apply various testing techniques and to build a robust software products
CO 5	Manage Software Projects and to understand advanced engineering concepts

CO - PO – PSO matrices of course

PO/PSO CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CS23432.1	2	2	3	2	2	2	2	2	2	2	3	2	1	3	-
CS23432.2	2	3	1	2	2	1	-	1	1	1	2	-	1	2	-
CS23432.3	2	2	1	1	1	1	1	1	1	1	1	1	2	2	1
CS23432.4	2	2	3	2	2	2	1	0	2	2	2	1	1	2	1
CS23432.5	2	2	2	1	1	1	1	0	2	1	1	1	2	1	-
Average	2.0	2.2	2.0	1.6	1.6	1.4	1.3	1.3	1.6	1.4	1.8	1.3	1.4	2.0	1.0

Correlation levels 1, 2 or 3 are as defined below:

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High) No correlation: “-”

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:

Azure Repos(Version Control)

Supports Git repositories and Team Foundation Version Control(TFVC).

Provides features like branching, pull requests, and code reviews.

Azure Pipelines (CI/CD)

Automates build, test, and deployment processes.

Supports multi-plat form builds (Windows,Linux, macOS).

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

Azure Boards (Agile Project Management)

Manages work using Kanban boards, Scrum boards ,and dashboards .

Tracks user stories, tasks, bugs, sprints, and releases.

Azure Test Plans(Testing)

Provides manual, exploratory, and automated testing.

Supports test case management and tracking.

Azure Artifacts (Package Management)

Stores and manages NuGet ,npm ,Maven ,and Pythonpackages.

Enables versioning and secure access to dependencies.

Getting Started with Azure DevOps

Step1: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 Gitor TFVC for version control.

Clone the repository and push your code.

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

Go to Pipelines→NewPipeline.

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.

Step4 : Manage Work with Azure Boards

Navigate to Boards.

Create work items, user stories, and tasks.

Organize sprints and track progress.

Step5: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 your given project.

Problem Statement:

Smart Property Listing, Search & Review Platform

In the real estate market, buyers and sellers often struggle with finding reliable listings and verifying the authenticity of dealers. This project aims to build a smart property listing and search platform where users can post properties for sale with key details like price, location, and contact information. Buyers can filter listings based on criteria such as BHK, area, amenities, and locality. A review system enables users to share their post-transaction experiences, building trust and helping others make informed decisions. Verified builders can log in to post new residential and commercial projects to attract potential buyers.

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:

- Road maps to guide a product's released 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 userstories to action at any given time, using them as a guide to implement new features or functionalities in a tool. Looking at tasks as userstories is a helpful way to imagine how a customer may use a feature and help steams 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.

EX NO: 4

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

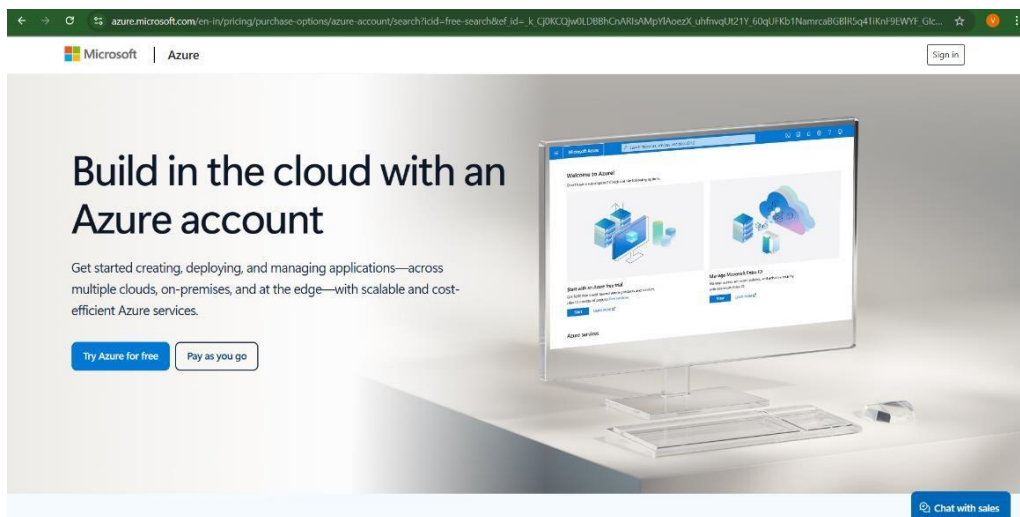
"As a [role], I [want to], [so that]."

Procedure:

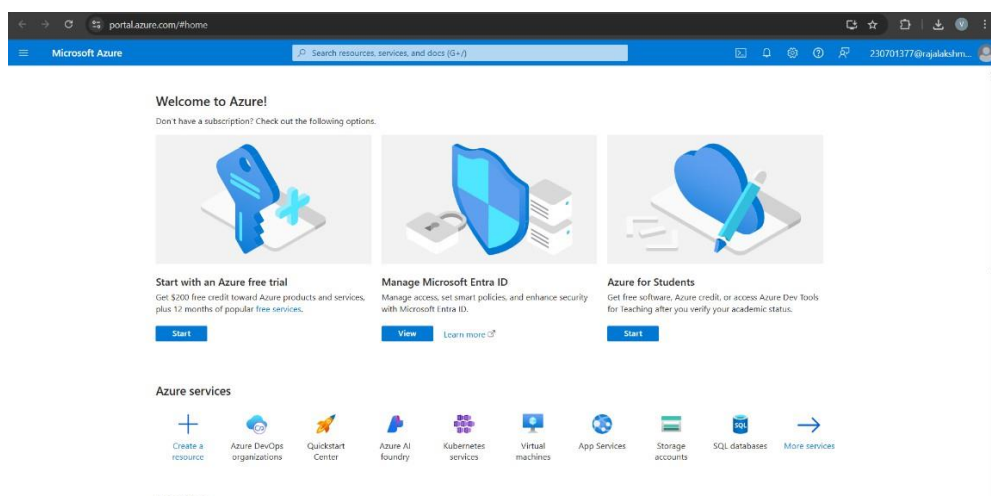
1. Open your web browser and go to the Azure website:
<https://azure.microsoft.com/en-in>. Sign in using your Microsoft account credentials. If you don't have an account, you'll need to create one.

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

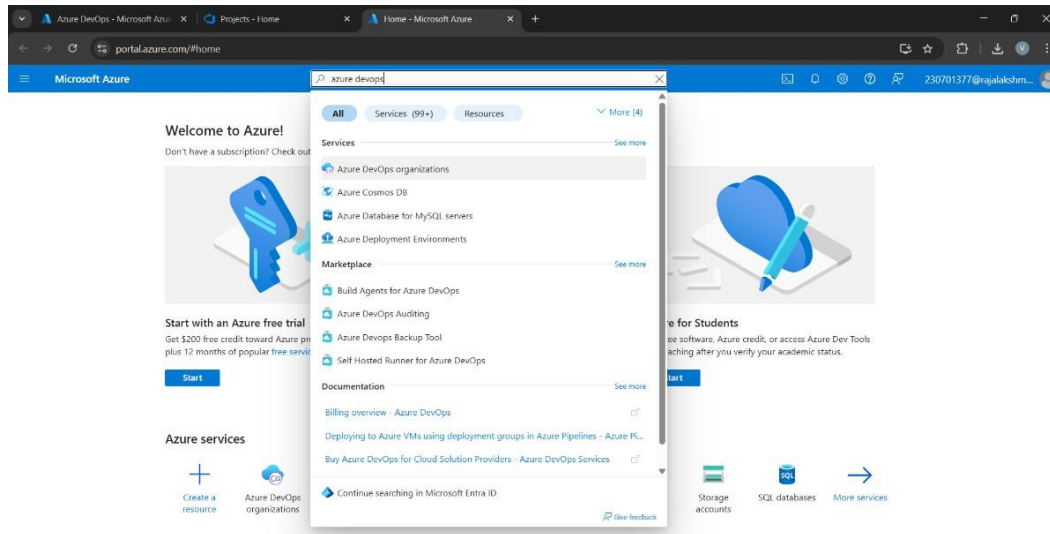
<https://signup.live.com/?lic=1>



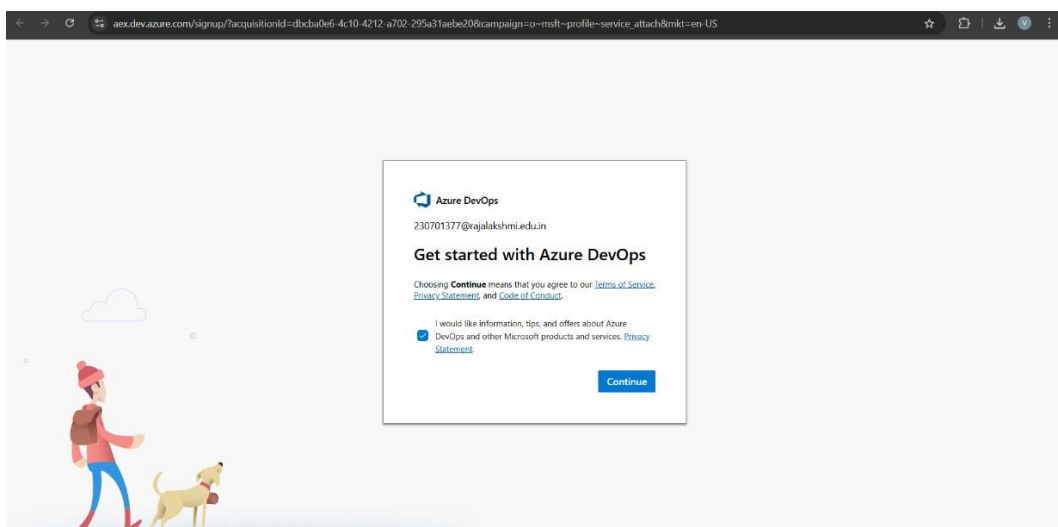
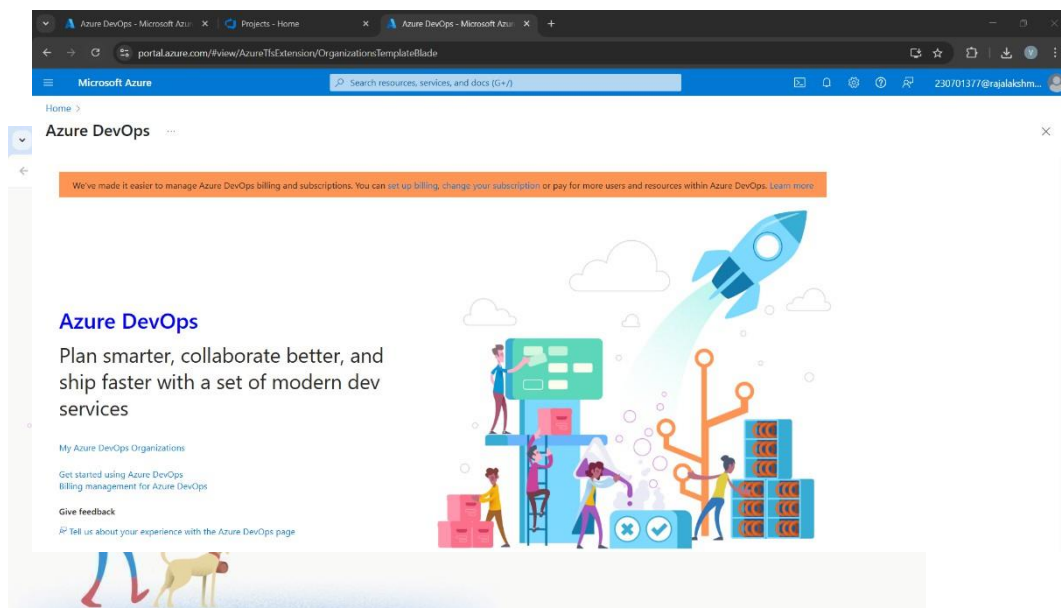
3. Azure home page

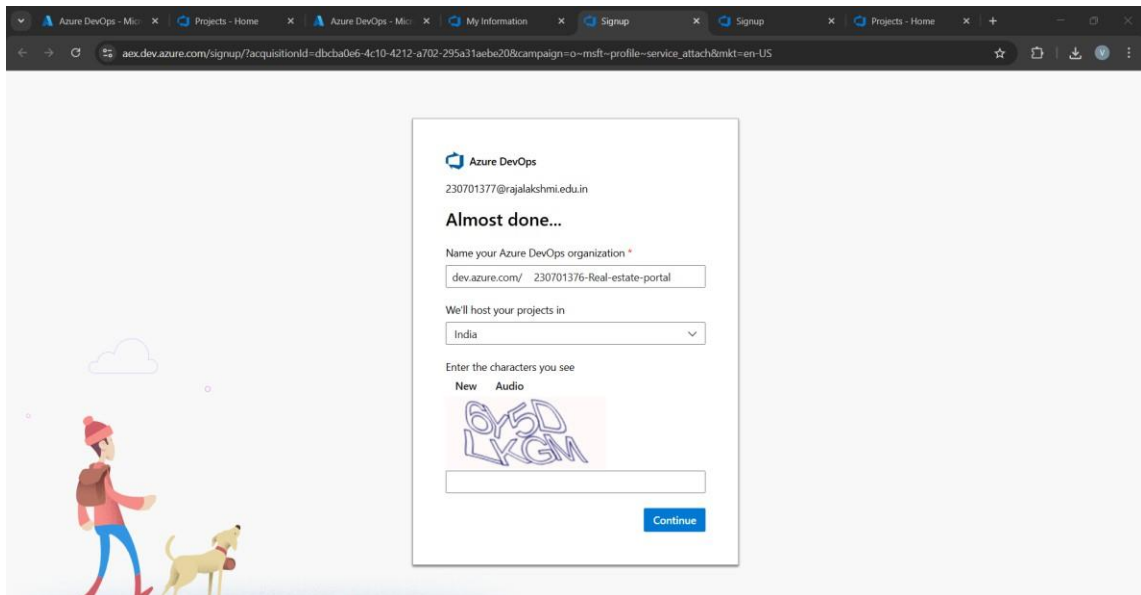


4. Open DevOps environment in the Azure platform by typing Azure DevOps Organizations in the search bar.



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



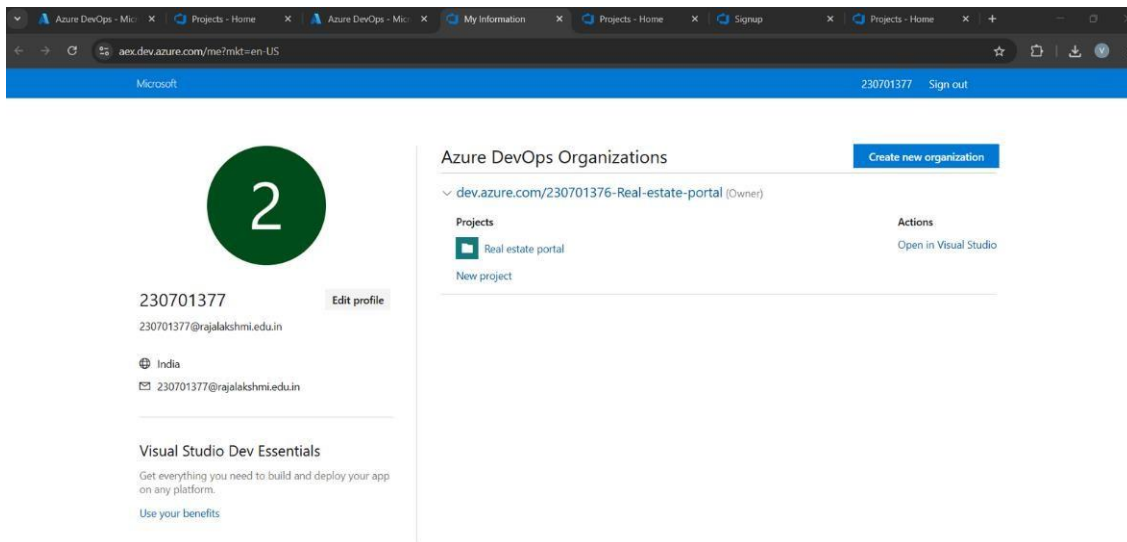


6. Create the First Project in Your Organization

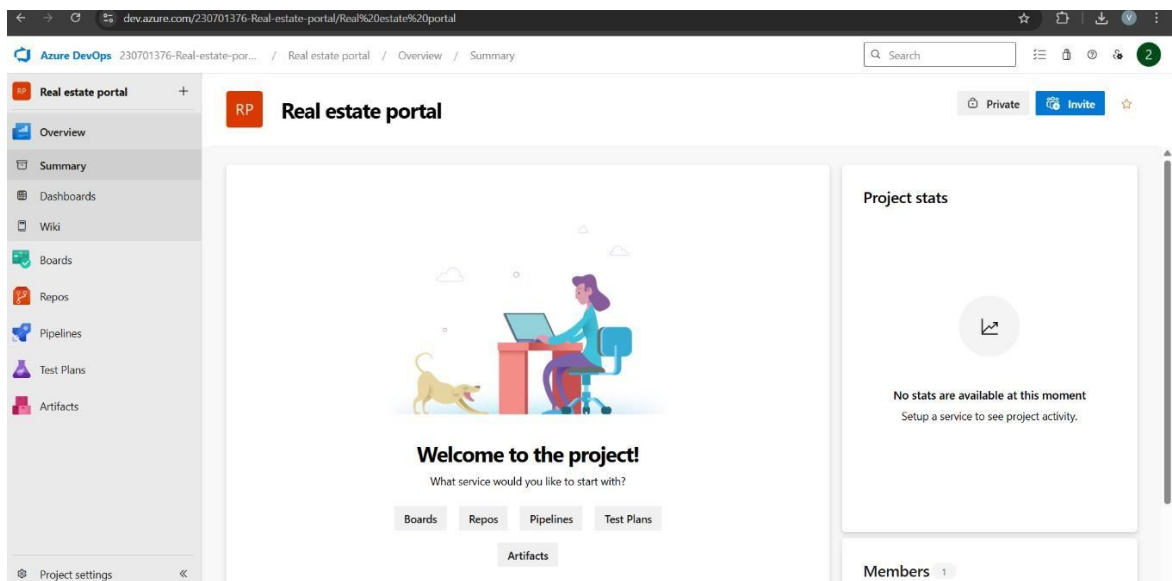
After the organization is setup, you'll need to create your first **project**. This is where you'll begin to manage code, pipelines, work items, and more.

- i. On the organization's **Homepage**, 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 setup your first project.

- 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.



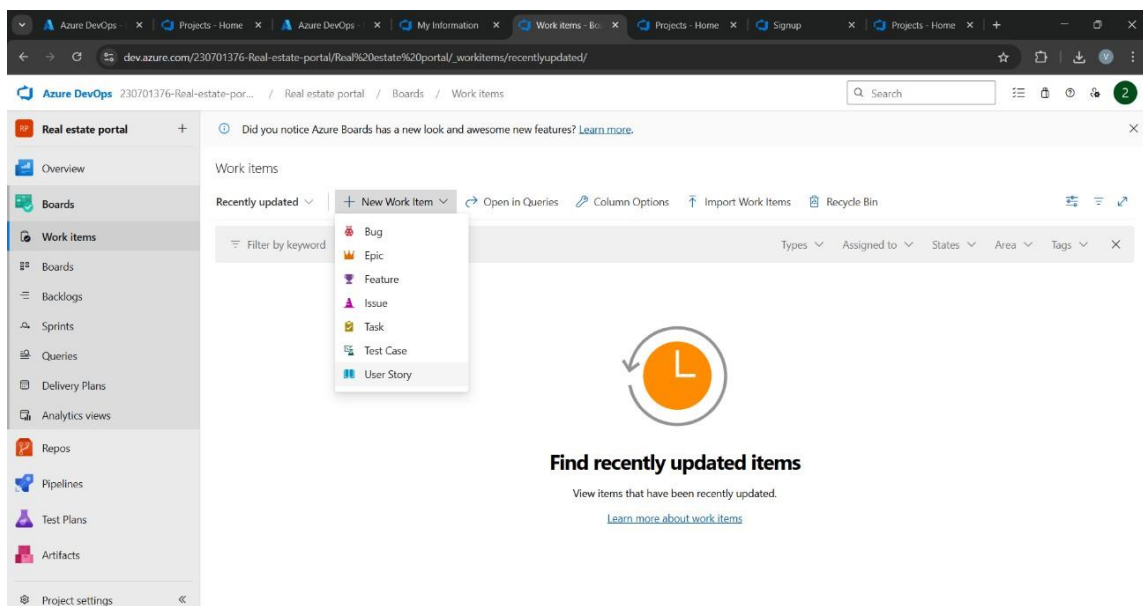
8. Project Dashboard



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

dev.azure.com/230701376/Real%20estate%20portal/_workitems/edit/34/

Azure DevOps 230701376 / Real estate portal / Boards / Work items

Did you notice Azure Boards has a new look and awesome new features? [Learn more.](#)

Recently updated Back to Work Items 4 of 15

USER STORY 34

34 Preprocess and Clean Dataset

230701377 0 Comments Add Tag

Save Follow

Updated by 230701377: Apr 3

Details

Stat: New Area: Real estate portal Reason: New Iteration: Real estate portal

Description

As a data scientist,
I want to clean and preprocess the collected data.
So that the AI model is trained on high-quality and error-free information.

Acceptance Criteria

- Duplicate and incomplete records are identified and removed.
- Missing values are handled using appropriate data imputation techniques.
- Outliers and incorrect values (e.g., extremely low or high prices) are detected and corrected.

Planning

Story Points

Priority

2

Risk

Classification

Value area

Business

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.

Result:

The user story was written successfully.

EX NO: 5

SEQUENCE DIAGRAM

Aim:

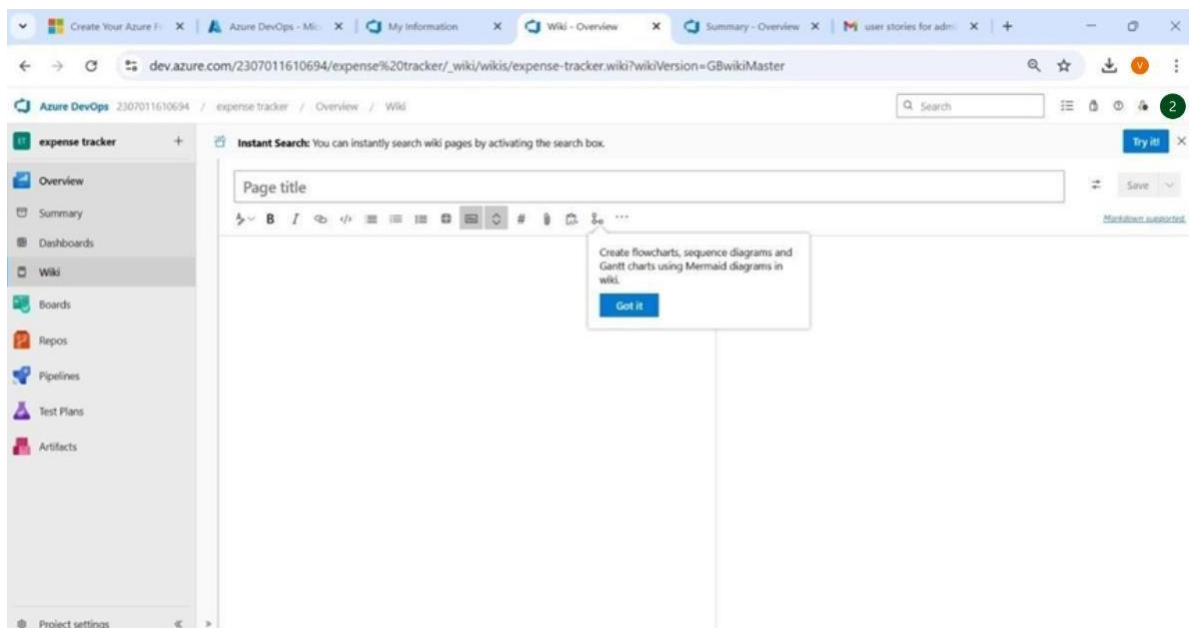
To design a Sequence Diagram by using Mermaid.js

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
```

```
actor User
```

```
participant Builder
```

```
participant Portal
```

```
participant Database
```

```
User->>Builder: post properties details
```

```
Builder->>Portal: Store property details
```

Portal->>Database: Store property details

Database-->>Portal: Confirmation

Portal-->>Builder: Confirmation

Builder-->>User: Property listed

User->>Builder: Search properties

Builder->>Portal: Query properties

Portal->>Database: Query properties

Database-->>Portal: Property list

Portal-->>Builder: Property list

Builder-->>User: Display properties

User->>Builder: Post review

Builder->>Portal: Store review

Portal->>Database: Store review

Database-->>Portal: Confirmation

Portal-->>Builder: Confirmation

Builder-->>User: Review posted

User->>Portal: Login request

alt Access granted

Portal-->>User: Login successful

User->>Builder: post new project

Builder->>Portal: Store project details

Portal->>Database: Store project details

Database-->>Portal: Confirmation

Portal-->>Builder: Confirmation

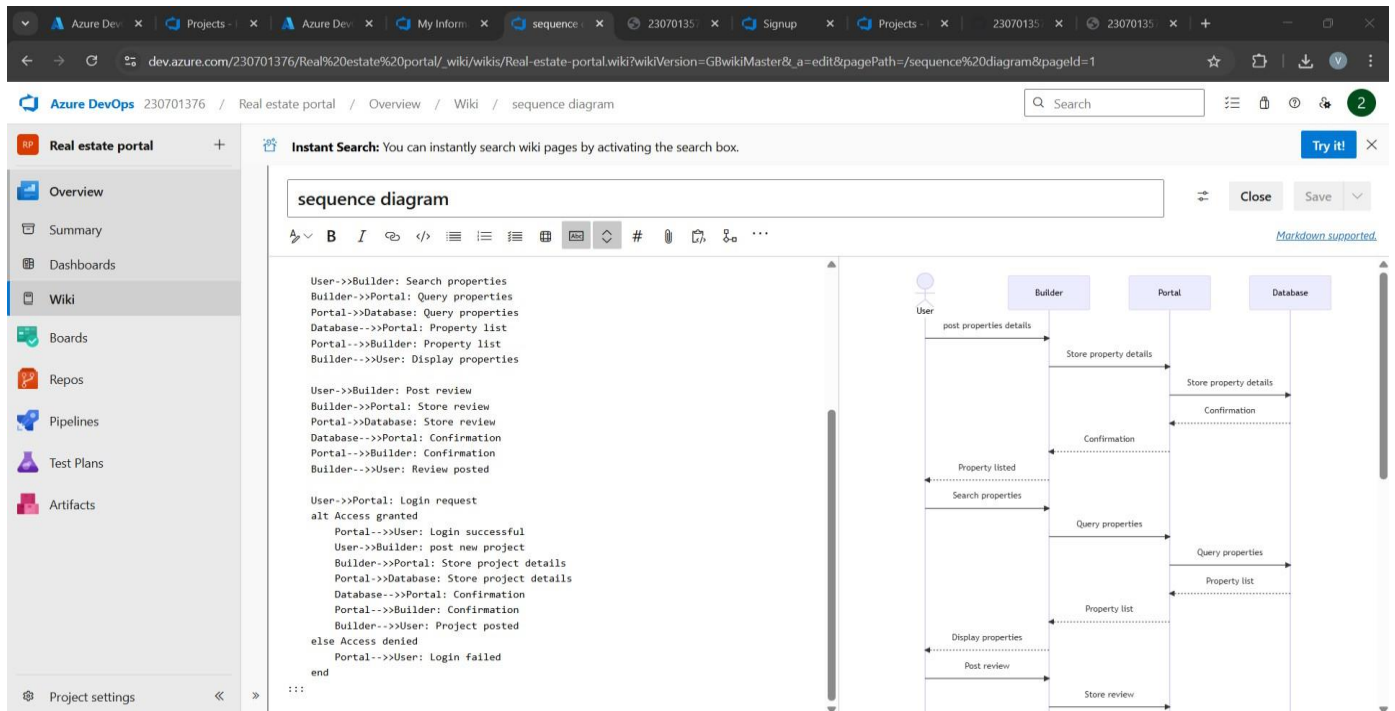
Builder-->>User: Project posted

else Access denied

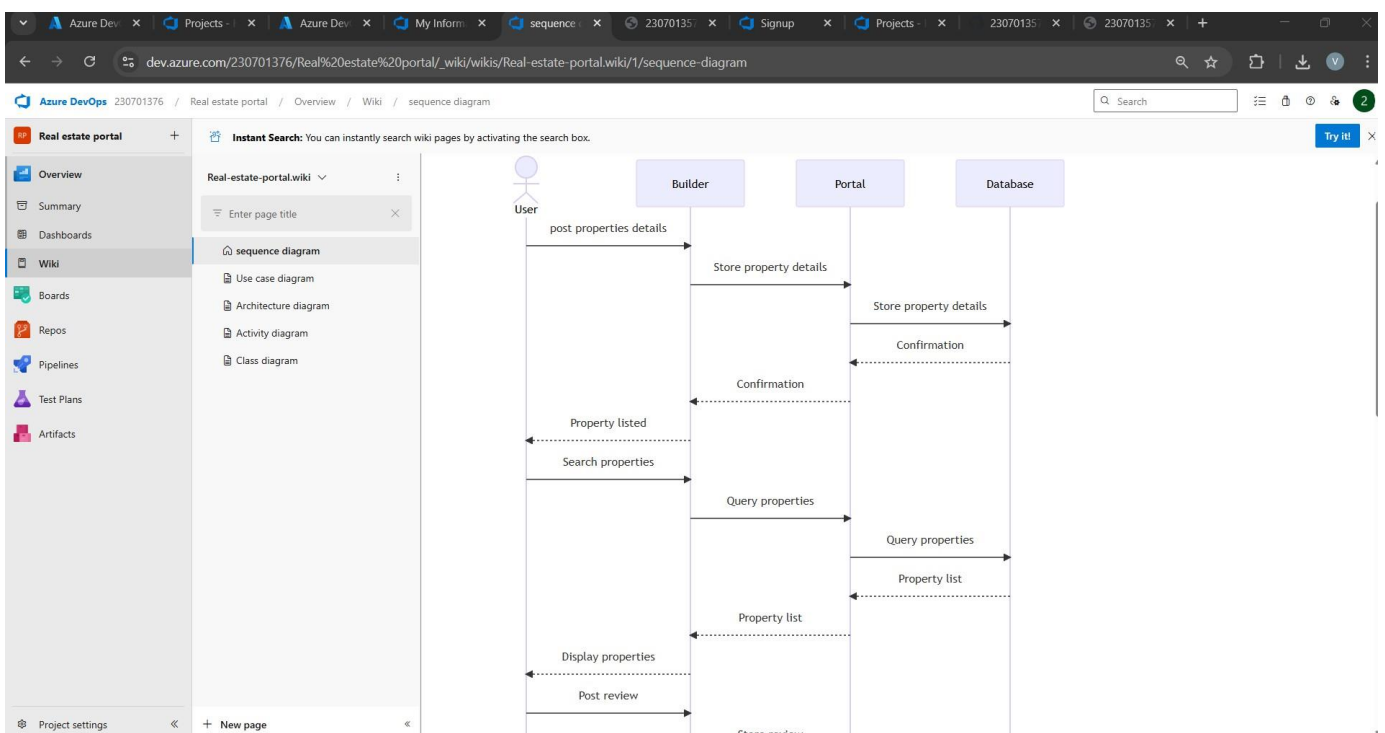
Portal-->>User: Login failed

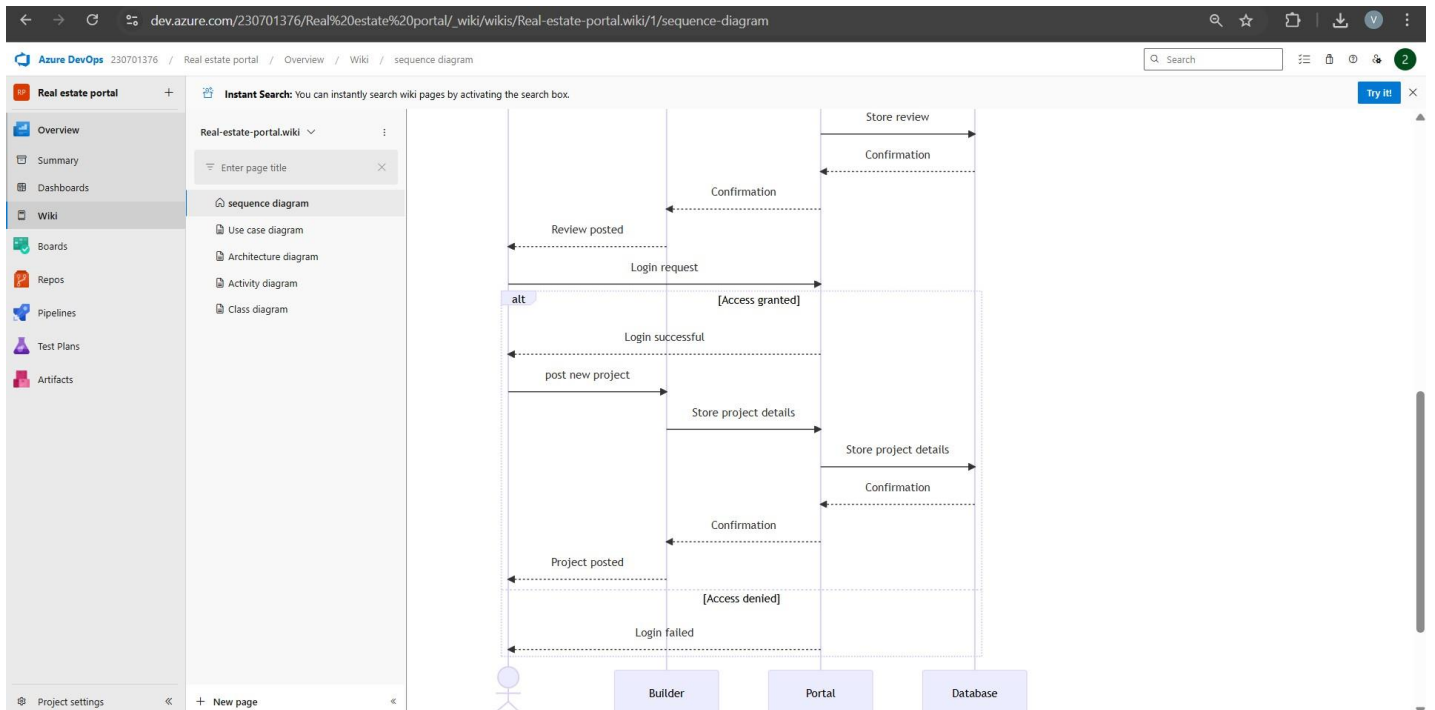
end

...



4. Click wiki menu and select the page





Result:

The sequence diagram was drawn successfully.

EX NO. 6

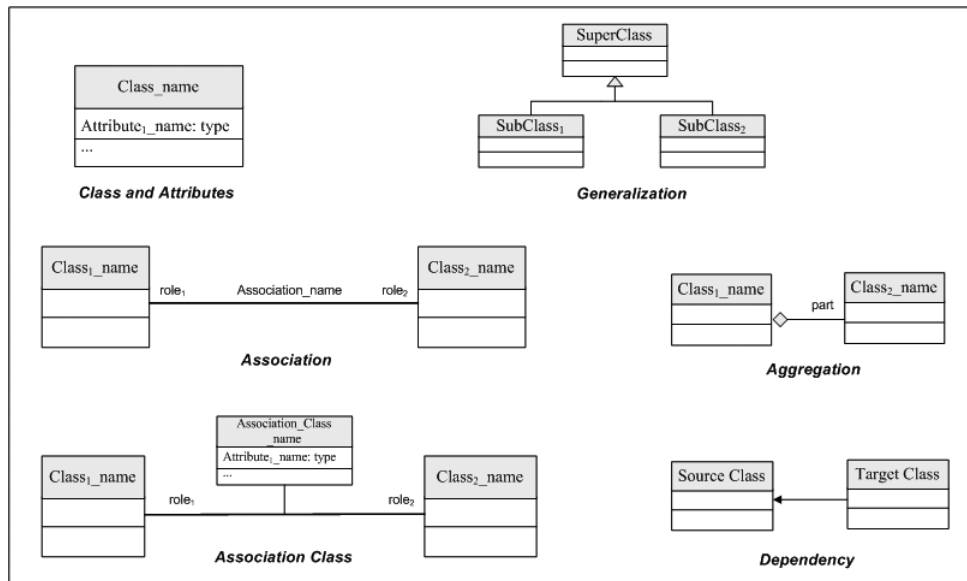
CLASS DIAGRAM

AIM :-

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

THEORY

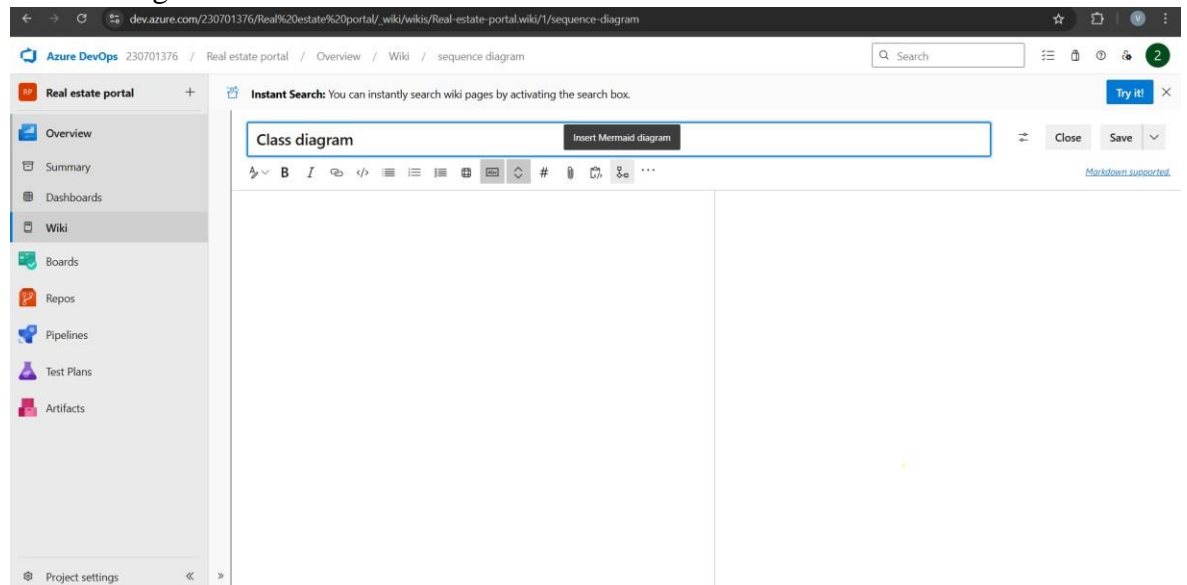
A UML class diagram is a visual tool that represents the structure of a system by showing its classes, attributes, methods, and the relationships between them.



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 LR
```

```
class User {
  -userID: int
  -name: string
  -email: string
  +register()
  +login()
  +postReview()
}
```

```
class Property {
  -propertyID: int
  -title: string
  -price: float
  -location: string
  -BHK: int
  +addProperty()
  +updateDetails()
}
```

```
class Review {
  -reviewID: int
  -rating: int
  -comment: string
  +addReview()
}
```

```
class Project {
  -projectID: int
  -title: string
  -location: string
  +addProject()
}
```

```
class Builder {
  -builderID: int
  -companyName: string
  +postProject()
}
```

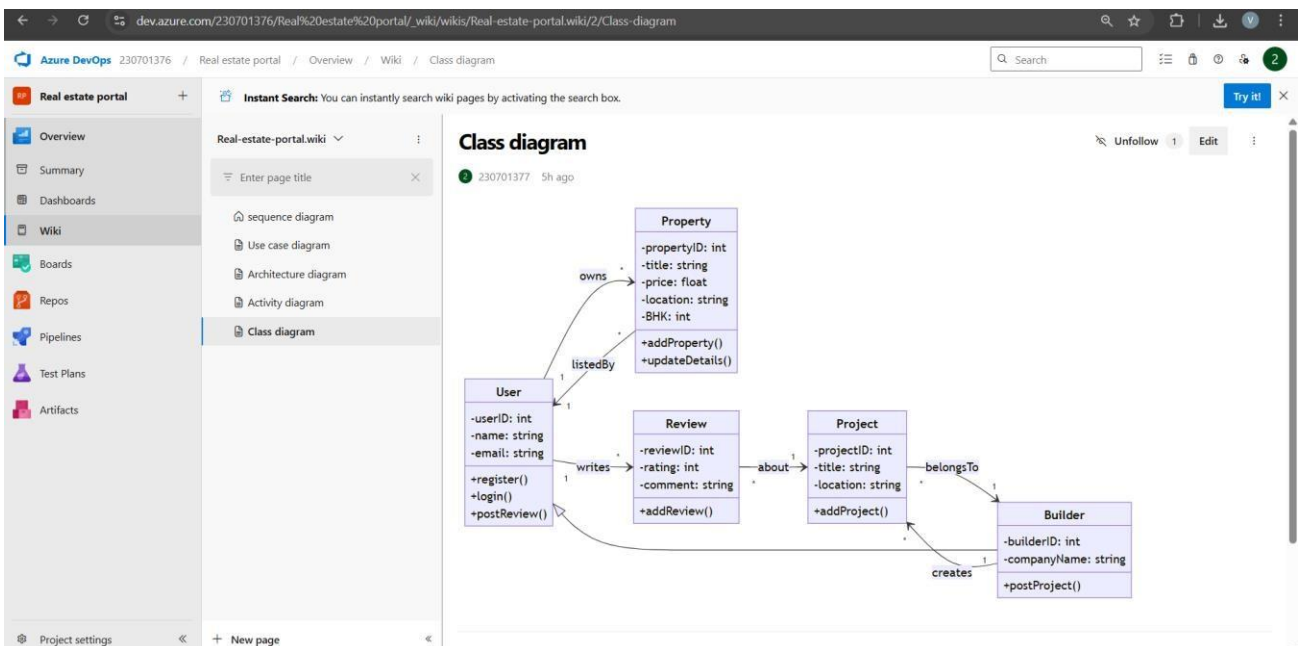
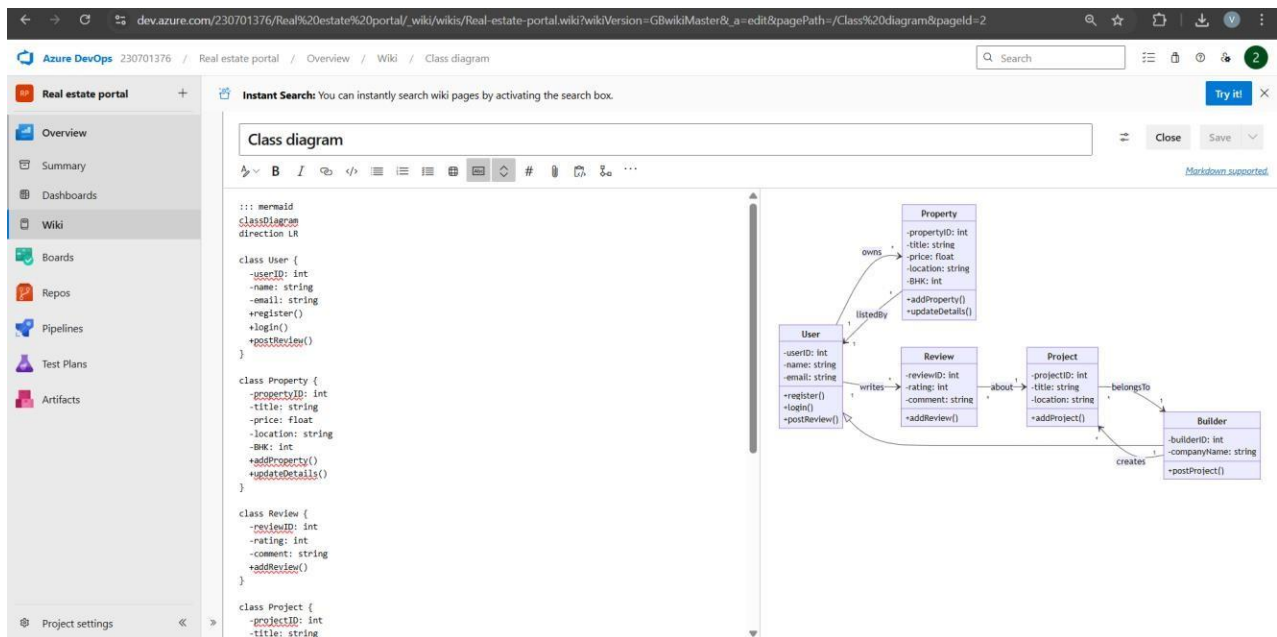
```
User "1" --> "*" Property : owns
Property "*" --> "1" User : listedBy
```

```
User "1" --> "*" Review : writes
Review "*" --> "1" Project : about
```

```
Builder --|> User
Builder "1" --> "*" Project : creates
Project "*" --> "1" Builder : belongsTo
```

```
:::
```

Type	Description
<	Inheritance
*	Composition
o	Aggregation
>	Association
<	Association
>	Realization



Result:

The use case diagram was designed successfully.

USECASE DIAGRAM

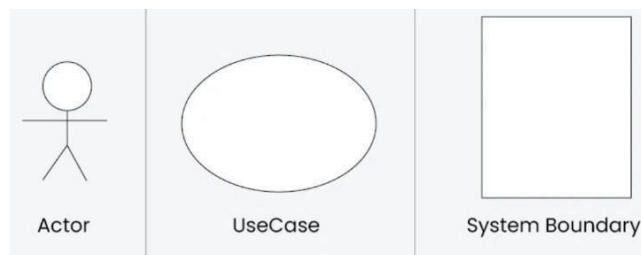
Aim:

Stepsto 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

Step1: 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 draw.io or export as PNG/JPG/SVG.

Step 2: Upload the Diagram to Azure DevOps

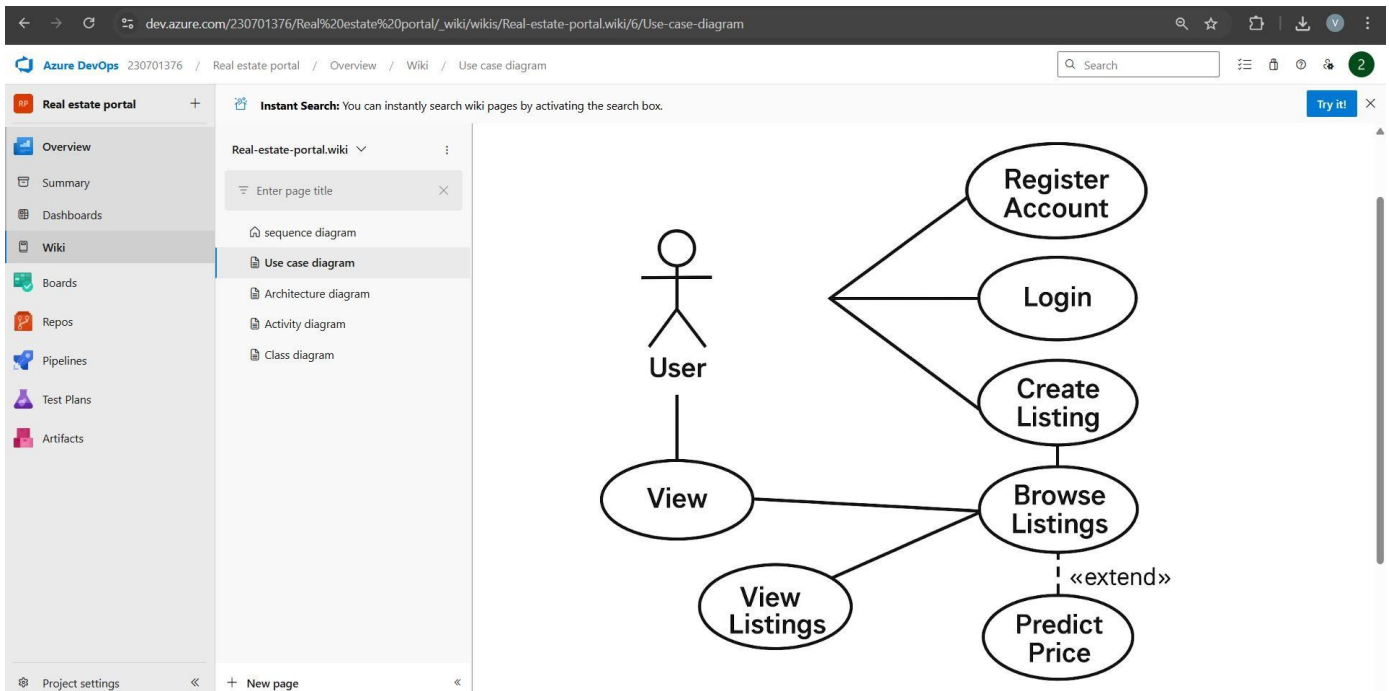
Option 1: Add to Azure DevOps Wiki

- 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.

Output:



Result:



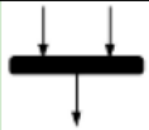


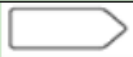





The use case diagram was designed successfully

ACTIVITY DIAGRAM**AIM :-**

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

THEORY

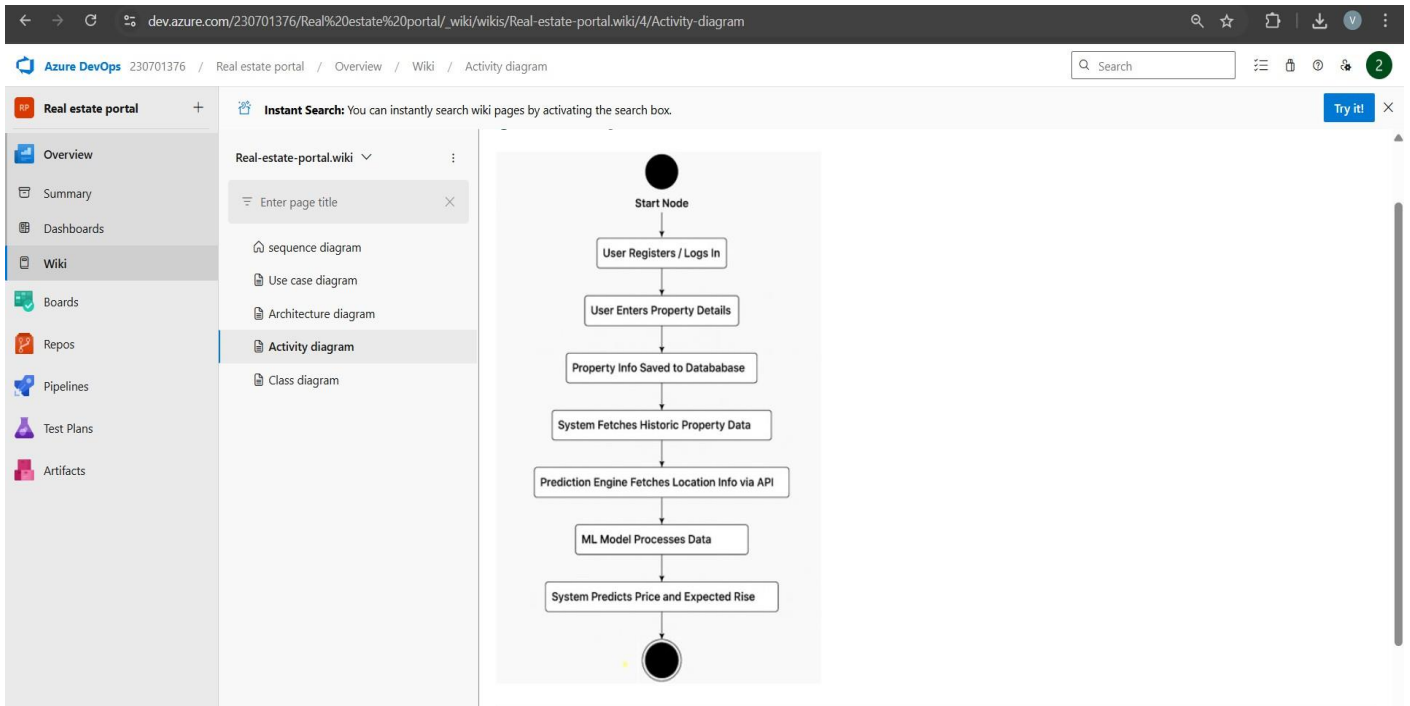
Activity diagrams are an essential part of the Unified Modelling Language (UML) that help visualize workflows, processes, or activities within a system. They depict how different actions are connected and how a system moves from one state to another.

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 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

Procedure

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

Output



Result:

The activity diagram was designed successfully

EX NO. 9

ARCHITECTURE DIAGRAM

Aim:

Steps to draw the Architecture Diagram using draw.io.

Theory:

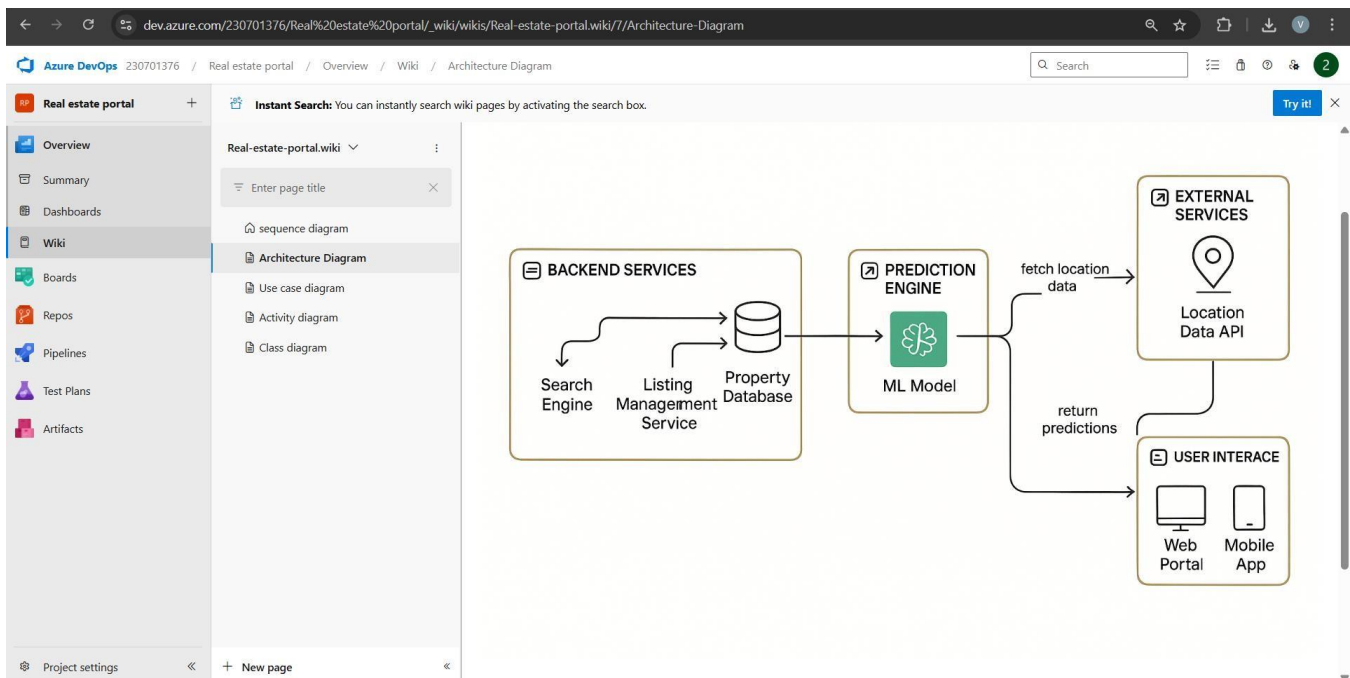
An architectural diagram is a visual representation that maps out the physical implementation for components of a software system. It shows the general structure of the software system and the associations, limitations and boundaries between each element.



Procedure

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

Output



Result:

The architecture diagram was designed successfully

USER INTERFACE

Aim:

Design User Interface for the given project

Output:

Chennai Property Connect[Home](#)[Properties](#)[Login](#)[Signup](#)[Builder Login](#)

User Login

Enter your email and password to access your account

Email

Password

[Forgot password?](#)

[Log in](#)

Demo credentials: user@example.com / password

Don't have an account? [Sign up](#)

Are you a builder? [Builder Login](#)

Chennai Property Connect

Your trusted partner for finding properties in Tamil Nadu. We connect buyers, sellers, and builders to create seamless property transactions.

Quick Links

[Home](#)
[Properties](#)
[Login](#)
[Signup](#)

Popular Locations

Chennai
Coimbatore
Madurai
Salem

Contact Us

123 Anna Salai, Chennai, Tamil Nadu
+91 9876543210
info@chennaipropertyconnect.com

localhost:8080/login

Chennai Property Connect[Home](#)[Properties](#)[Login](#)[Signup](#)[Builder Login](#)

Create an account

Enter your information to create a user account

Full Name

Email

Password

Confirm Password

[Sign up](#)

Already have an account? [Log in](#)

Chennai Property Connect

Your trusted partner for finding properties in Tamil Nadu. We connect buyers, sellers, and builders to create seamless property transactions.

Quick Links

[Home](#)
[Properties](#)
[Login](#)
[Signup](#)

Popular Locations

Chennai
Coimbatore
Madurai
Salem

Contact Us

123 Anna Salai, Chennai, Tamil Nadu
+91 9876543210
info@chennaipropertyconnect.com

Find Your Perfect Property in Tamil Nadu

Connect with trusted property sellers, agents, and builders across Tamil Nadu. Find your dream home today!

[Browse Properties](#)

City

Select City

Area

Select Area

Property Type

Select Type

BHK

Select BHK

Search Properties

Featured Properties

[View All](#)

Browse Properties

City

Select City

Area

Select Area

Property Type

Select Type

BHK

Select BHK

Search Properties

Filters

Price Range

Min

Max

₹

₹

Sort By

Newest First

Amenities

- ☐ Swimming Pool
- ☐ Gym
- ☐ Security
- ☐ Power Backup
- ☐ Lift
- ☐ Car Parking
- ☐ Playground
- ☐ Garden
- ☐ Gated Community
- ☐ Club House

16 properties found

[Grid](#) [List](#)

Elegant 3 BHK Apartment in Anna Nagar

Anna Nagar, Chennai

3 BHK

1450 sq.ft

₹85,00,000

Posted on 10/15/2023



Modern 2 BHK Flat in OMR

OMR, Chennai

2 BHK

1100 sq.ft

₹60,00,000

Posted on 10/19/2023





Builder Login

Access your builder dashboard to manage your projects

Email

Password

[Forgot password?](#)

Log In

Demo credentials: builder@example.com / password

[Not a builder? User Login](#)

Chennai Property Connect

Your trusted partner for finding properties in Tamil Nadu. We connect buyers, sellers, and builders to create seamless property transactions.

Quick Links

[Home](#)
[Properties](#)
[Login](#)

Popular Locations

[Chennai](#)
[Coimbatore](#)
[Madurai](#)

Contact Us

[123 Anna Salai, Chennai, Tamil Nadu](#)
[+91 9876543210](#)

Result:

The UI was designed successfully.

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'

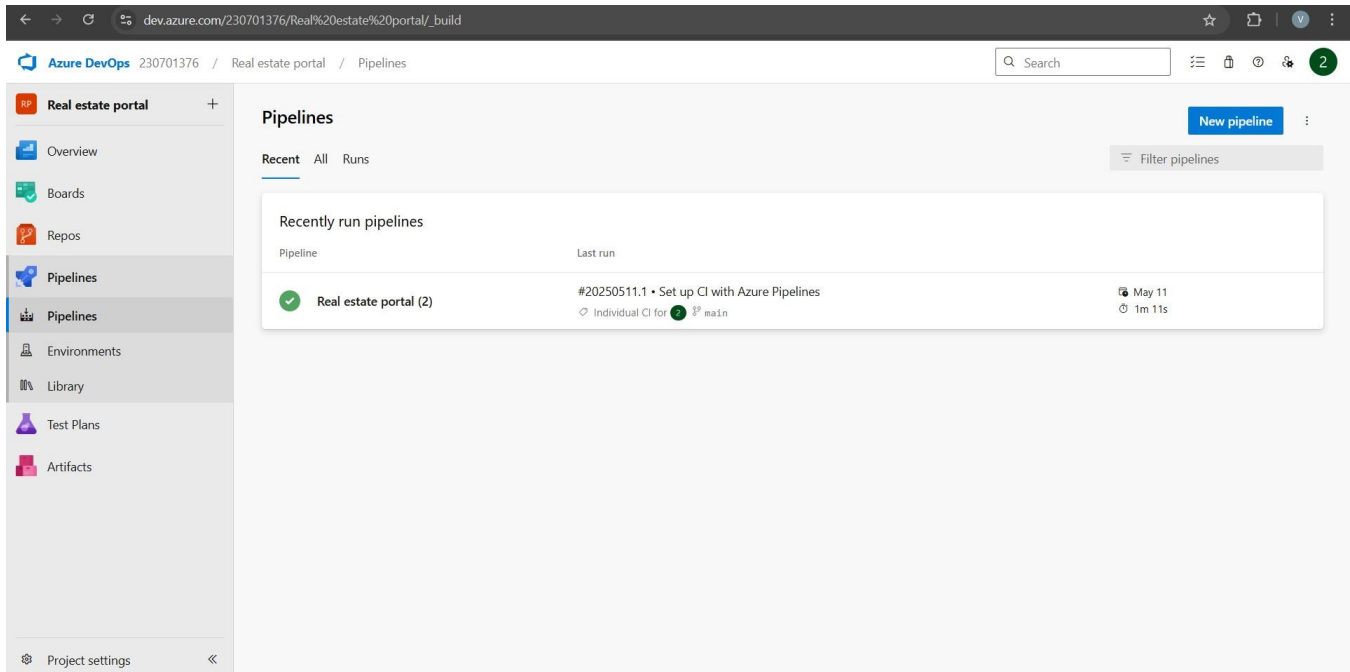
- script: npm run build
  displayName: 'Build application'

- 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.