



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

DEPARTMENT OF INFORMATION TECHNOLOGY
LAB MANUAL

CS23432 – Software Construction

(REGULATION 2023)

RAJALAKSHMI ENGINEERING COLLEGE
Thandalam, Chennai-602015

Name: Sameeha A

Register No: 231001176

Year / Branch / Section: 2nd/IT/ FB

Semester: IV

Academic Year: 2024 - 2025

Ex No	List of Experiments
1	Study of Azure DevOps
2	Designing Project using AGILE-SCRUM Methodology.
3	Agile Planning
4	User stories – Creation
5	Architecture Diagram Using AZURE
6	Designing Use case and Class Diagram
7	Designing Interaction Diagrams
8	Design Interface
9	Implementation – Design a Web Page based on Scrum Methodology
10	Testing using Azure.
11	Deployment

INDEX

Ex No	Date	Topic	Page No	Sign
1		Study of Azure DevOps		
2		Problem Statement		
3		Agile Planning		
4		Create User Stories		
5		Sequence Diagram		
6		Class Diagram		
7		Use Case Diagram		
8		Activity Diagram		
9		Architecture Diagram		
10		User Interface		
11		Implementation		

Course Outcomes (COs)

Course Name: Software Engineering

Course Code: 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:

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.

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 your given project.

Problem Statement:

Online Quiz System

Develop a web application that allows users to take quizzes on various topics.

Implement features like user registration, question banks, and score tracking.

Implement Quiz Management that would allow setting up of various quiz types like MCQ, True/False, Fill in Blanks etc

Upload and CRUD functionality for Image uploading

Generate reports on student performance for individual quizzes and overall topic areas.

Instructors can view detailed reports on student responses to individual questions

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.

EX NO: 4

CREATE USER STORIES

Aim:

To create User Stories

THEORY

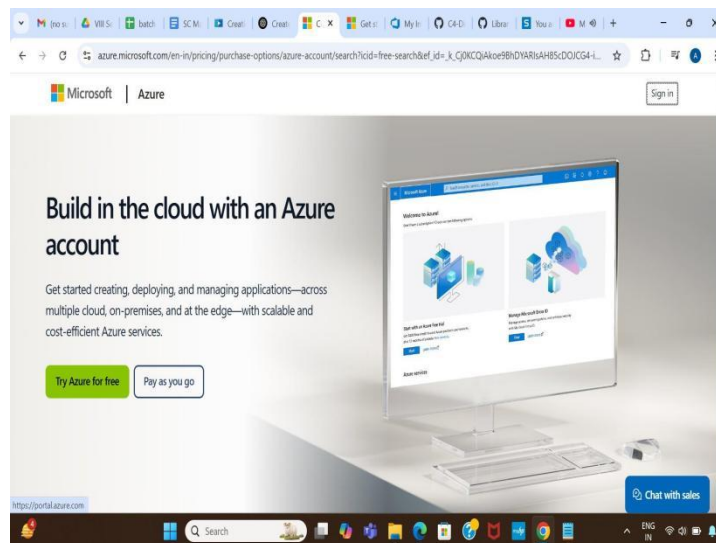
A user story is an informal, general explanation of a software feature written from the perspective of the end user. Its purpose is to articulate how a software feature will provide value to the customer.

User story template

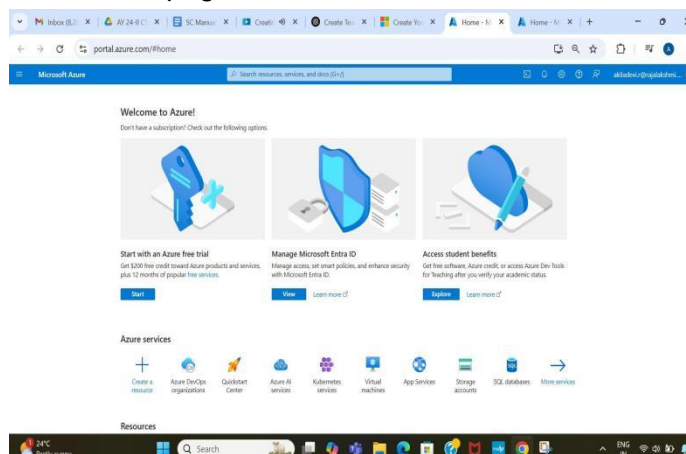
"As a [role], I [want to], [so that]."

Procedure:

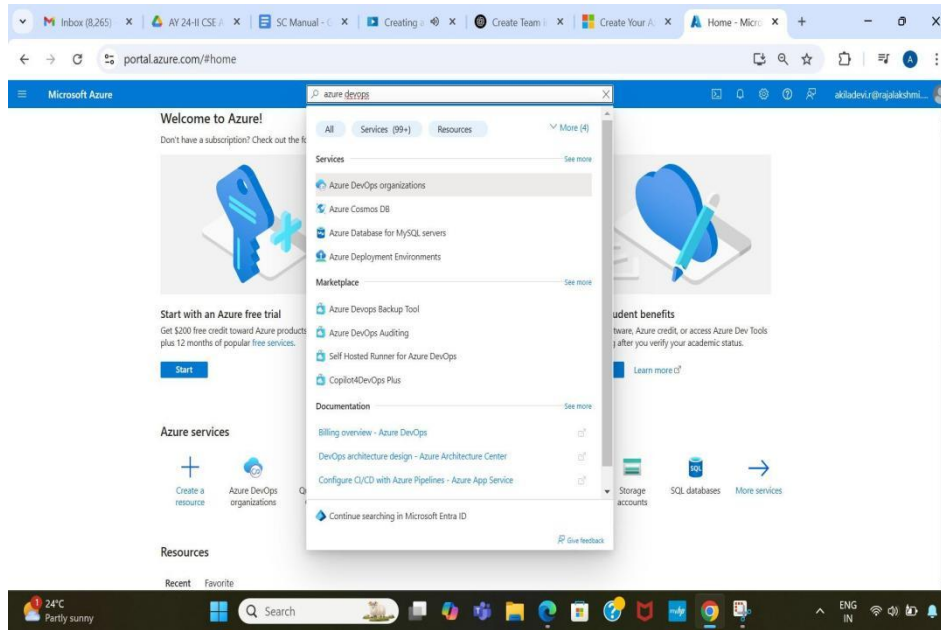
1. Open your web browser and go to the Azure website:
<https://azure.microsoft.com/en-in> Sign in using your Microsoft account credentials. If you don't have an account, you'll need to create one.
2. _If you don't have a Microsoft account, you can sign up for
<https://signup.live.com/?lic=1>



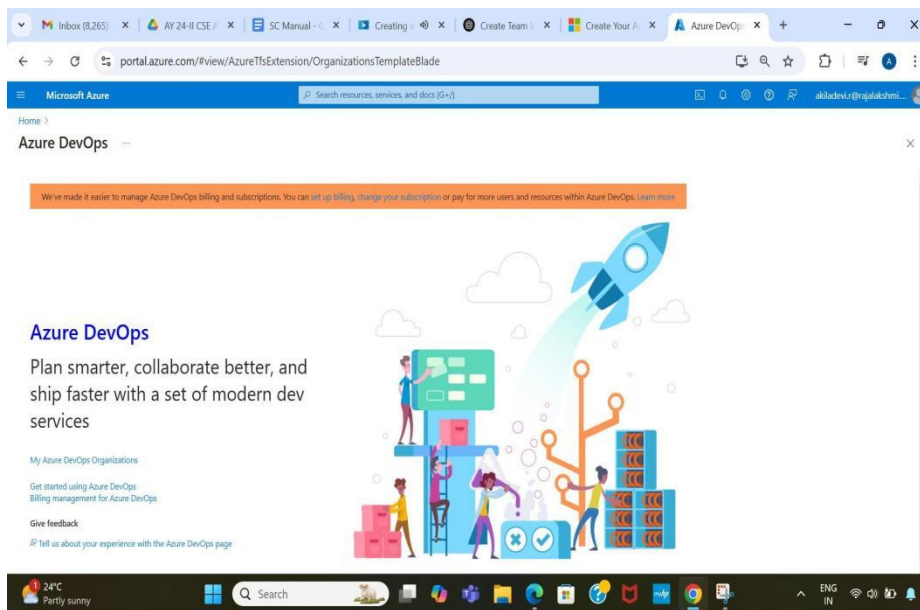
3. Azure home page



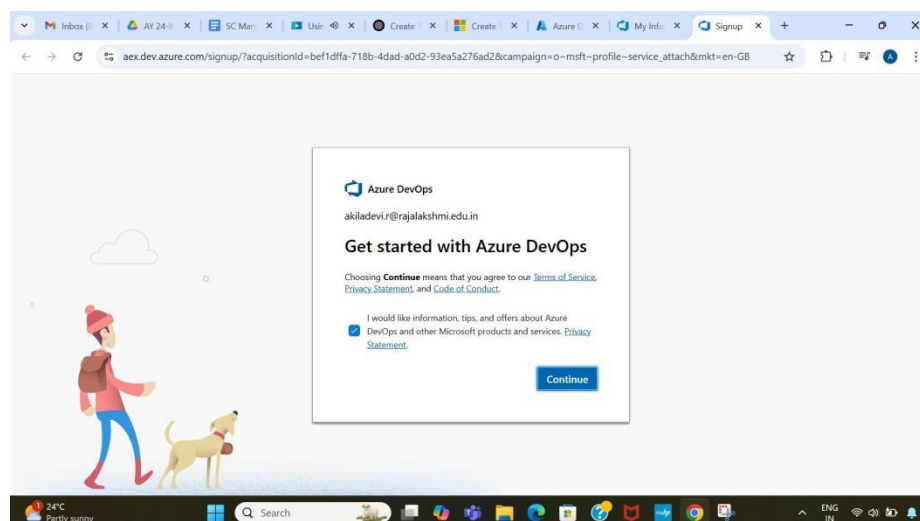
4. Open DevOps environment in the Azure platform by typing Azure DevOps Organizations in the search bar.

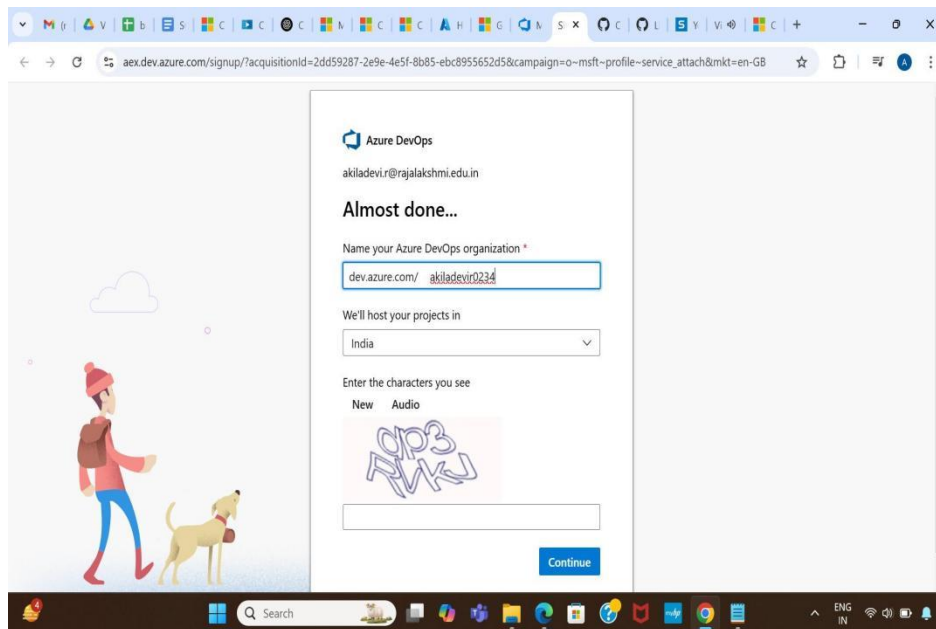


5. Click on the My Azure DevOps Organization link and create an organization and you should be taken to the Azure DevOps Organization



Home page.



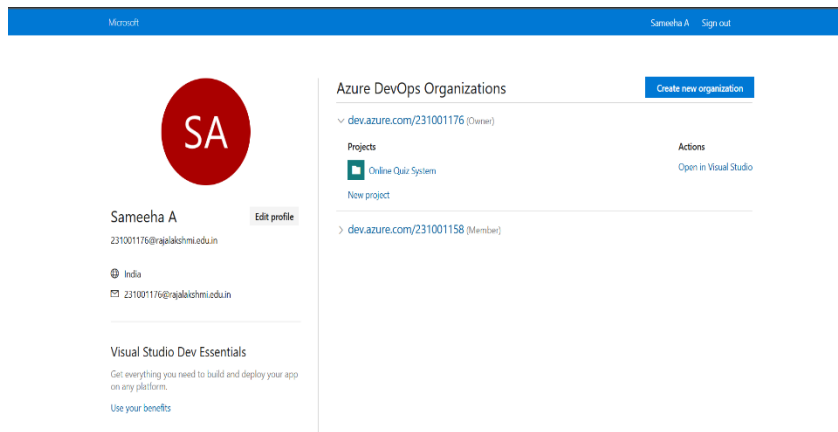


6. Create the First Project in Your Organization

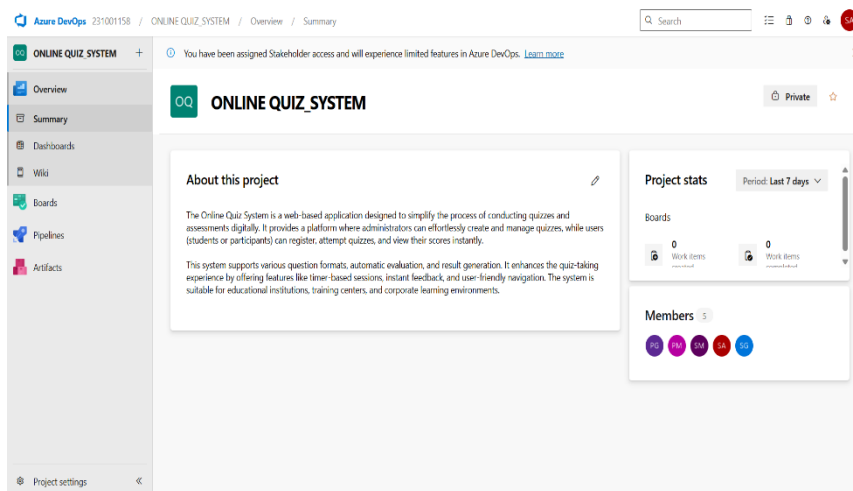
After the organization is set up, you'll need to create your first **project**. This is where you'll begin to manage code, pipelines, work items, and more.

- 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.

<div> <div>Azure DevOps</div> <div>231001158 / ONLINE QUIZ_SYSTEM / Boards / Work Items</div> <div> <div>Q Search</div> <div> <div></div> <div></div> <div></div> <div></div> </div> <div>SA</div> </div> </div>						
<div> <div>ONLINE QUIZ_SYSTEM</div> <div> <div>Overview</div> <div>Boards</div> <div>Work Items</div> <div>Boards</div> <div>Backlogs</div> <div>Sprints</div> <div>Queries</div> <div>Delivery Plans</div> <div>Pipelines</div> <div>Artifacts</div> </div> <div>Project settings</div> </div>						
<div> <div>Work Items</div> <div> <div>Recently updated</div> <div>+ New Work Item</div> <div>Open in Queries</div> <div>Column Options</div> <div>Import Work Items</div> <div>Recycle Bin</div> </div> <div> <div>Filter by keyword</div> <div>Types</div> <div>Assigned to</div> <div>States</div> <div>Area</div> <div>Tags</div> <div>X</div> </div> </div>						
ID	Title	Assigned To	State	Area Path	Tags	
7	1. User Registration and Profile Management	Paranashwari Muniyandi	New	ONLINE QUIZ_SYSTEM		
8	3. Question Bank Management	Sahana J G	Active	ONLINE QUIZ_SYSTEM		
9	4. Quiz Taking and User Experience	Sameeha A	Active	ONLINE QUIZ_SYSTEM		
10	3.1 Question Categorization	Sahana J G	Active	ONLINE QUIZ_SYSTEM		
11	4.1: Interactive Quiz Interface	Sameeha A	Active	ONLINE QUIZ_SYSTEM		
12	1.1 User Registration	Paranashwari Muniyandi	New	ONLINE QUIZ_SYSTEM		
13	6. As an instructor, I want to categorize quiz questions by topics, so th	Sahana J G	Active	ONLINE QUIZ_SYSTEM		
14	6 Security and Data Privacy	PUSHPA G	Active	ONLINE QUIZ_SYSTEM	Access Control	
15	6.1: Secure Data Storage	PUSHPA G	Active	ONLINE QUIZ_SYSTEM	Azure Compli	
16	3.2 Randomized Question Pool	Sahana J G	Active	ONLINE QUIZ_SYSTEM		
17	1.2 User Login & Profile Management	Paranashwari Muniyandi	New	ONLINE QUIZ_SYSTEM		
18	7. As an instructor, I want to create a pool of questions for each toic.	Sahana J G	Active	ONLINE QUIZ_SYSTEM		

10. Fill in User Story Details

Azure DevOps

231001158 / ONLINE QUIZ_SYSTEM / Boards / Work Items

Q Search

SA

ONLINE QUIZ_SYSTEM

Overview

Boards

Work Items

Boards

Backlogs

Sprints

Queries

Delivery Plans

Pipelines

Artifacts

Project settings

Recently updated

Back to Work Items

3 of 49

EPIC 8

4. Quiz Taking and User Experience

Sameeha A

4 Comments

Add Tag

Save

Follow

State: Active

Area: ONLINE QUIZ_SYSTEM

Reason: Implementation started

Iteration: ONLINE QUIZ_SYSTEM/Iteration 1

Updated by Sameeha A: Mar 27

Details

Description

The objective of this epic is to create an engaging and user-friendly quiz-taking experience for students. It includes the design of an interactive interface for various question formats, ensuring the quiz experience is smooth, responsive, and real-time for users. Students should be able to test their knowledge through multiple types of questions such as MCQs, True/False, Fill in the Blanks, and others, while tracking their time and progress seamlessly.

Planning

Priority: 2

Risk

Effort

Business Value

Time Criticality

Start Date: Select a date...

Target Date: Select a date...

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.

Discussion

Add a comment. Use # to link a work item, @ to mention a person, or ! to link a pull request.

switch to Markdown editor

Azure DevOps

231001158 / ONLINE QUIZ_SYSTEM / Boards / Work Items

Q Search

SA

ONLINE QUIZ_SYSTEM

Overview

Boards

Work Items

Boards

Backlogs

Sprints

Queries

Delivery Plans

Pipelines

Artifacts

Project settings

Recently updated

Back to Work Items

21 of 49

USER STORY 27

27 9. As a student, I want to take a quiz and answer questions in different formats, so I can test my knowledge.

Sameeha A

1 Comment

Add Tag

Save

Follow

State: Active

Area: ONLINE QUIZ_SYSTEM

Reason: Implementation started

Iteration: ONLINE QUIZ_SYSTEM/Iteration 1

Updated by Sameeha A: Mar 27

Details

Description

As a student, I want to take a quiz that includes questions in different formats, such as multiple-choice, fill-in-the-blank, true/false, and short answer, so I can effectively test my knowledge in various ways. This diversity in question formats helps improve my understanding, assess my learning more accurately, and make the quiz experience more engaging.

Planning

Story Points

Priority: 2

Risk

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](#)

Acceptance Criteria

Acceptance Criteria:

- The quiz interface should present a mix of different question types (MCQs, True/False, Fill-in-the-Blanks, etc.).
- Students should be able to answer each question easily and intuitively, depending on the question format.
- The system should guide the student on how to answer different question formats (radio buttons, checkboxes, text fields, etc.).
- After completing the quiz, the student should see a summary of their answers and scores.

Classification

Value area: Business

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.

Azure DevOps 231001158 / ONLINE QUIZ_SYSTEM / Boards / Work items

ONLINE QUIZ_SYSTEM +

Overview Boards Work items Boards Backlogs Sprints Queries Delivery Plans Pipelines Artifacts Project settings

Recently updated Back to Work Items 22 of 49

USER STORY 28

28 3. As an instructor, I want to create multiple quiz types (MCQ, True/False, Fill-in-the-blanks), so I can assess students' knowledge in different formats.

Paraneshwari Muniyandi 0 Comments Add Tag Save Follow

Updated by Paraneshwari Muniyandi: Mar 27

Details

State: New Area: ONLINE QUIZ_SYSTEM Reason: New Iteration: ONLINE QUIZ_SYSTEM

Description

As an instructor, I want to create multiple quiz types such as Multiple Choice Questions (MCQ), True/False, and Fill-in-the-blanks, so I can assess students' knowledge in different formats. This will allow me to design quizzes that best evaluate various aspects of a student's understanding, providing a more comprehensive assessment experience. The system should enable me to easily create, modify, and manage quizzes with these different question formats through a user-friendly interface.

Acceptance Criteria

The instructor should be able to access a user interface for creating new quizzes. Within this interface, the instructor will have the ability to select from different question types (MCQ, True/False, Fill-in-the-blanks) while adding questions to the quiz. For Multiple Choice Questions, the system should allow the instructor to specify the question and provide several answer choices, marking the correct one. For True/False questions, the instructor will define a statement and select either "True" or "False" as the valid options. In the case of Fill-in-the-blanks, the system will enable the instructor to create a question with one or more blanks where students will need to provide the correct answers.

Planning

Story Points
Priority: 2
Risk

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](#)

Classification

Value area: Business

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.

Azure DevOps 231001158 / ONLINE QUIZ_SYSTEM / Boards / Work items

ONLINE QUIZ_SYSTEM +

Overview Boards Work items Boards Backlogs Sprints Queries Delivery Plans Pipelines Artifacts Project settings

Recently updated Back to Work Items 31 of 49

USER STORY 40

40 14 As an instructor, I want to view detailed reports on student responses to individual questions, so I can understand where students are struggling.

Saiswetha M 3 Comments Add Tag Save Follow

Updated by Saiswetha M: Mar 27

Details

State: Active Area: ONLINE QUIZ_SYSTEM Reason: Implementation started Iteration: ONLINE QUIZ_SYSTEM

Description

As an instructor, I need to analyze student responses to individual quiz questions to identify trends and common mistakes. This will help me understand which concepts students struggle with and adjust my teaching strategies accordingly. The system should allow me to view detailed response analytics, including correctness, response time, and patterns of incorrect answers. Additionally, I should be able to filter and sort responses based on different criteria, such as question difficulty, student performance, and time taken.

Acceptance Criteria

- 1. The system must display student responses for each quiz question, including correct/incorrect status.
- 2. Instructors must be able to filter responses by question, student, and performance.
- 3. Sorting options should be available for response time and correctness.

Planning

Story Points
Priority: 3
Risk

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](#)

Classification

Value area: Business

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:

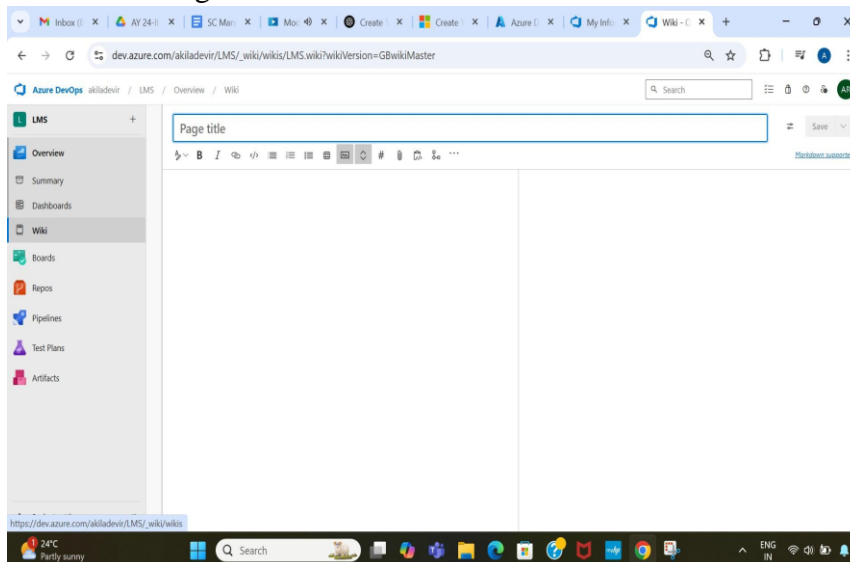
To design a Sequence Diagram by using Mermaid.js

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

participant User

participant Instructor

participant StudentAdmin as Student College
Administrator

participant RegSystem as Registration System

participant QuizSystem as Quiz Management System

participant AzureDB as Azure SQL Database

participant AzureAD as Azure AD BC

participant Dashboard as Analytics Dashboard

User->>RegSystem: Enter Registration Details

RegSystem->>AzureAD: Validate User Credentials

AzureAD-->>RegSystem: Validation response

RegSystem->>AzureDB: Store User Data

RegSystem-->>User: Registration Successful

Instructor->>QuizSystem: Create Quiz With Questions

QuizSystem->>AzureDB: Store Quiz Data

QuizSystem-->>Instructor: Quiz Created Successfully

User->>QuizSystem: Start Quiz Attempt

QuizSystem->>AzureDB: Fetch Questions

AzureDB-->>QuizSystem: Return Questions

QuizSystem-->>User: Display Quiz Questions

User->>QuizSystem: Submit Answers

QuizSystem->>AzureDB: Store Answers

QuizSystem-->>User: Show score and feedback

StudentAdmin->>Dashboard: Request student performance report

Dashboard->>AzureDB: Fetch Quiz Performance Data

AzureDB-->>Dashboard: Return performance data

Dashboard-->>StudentAdmin: Display performance report

:::

Explanation:

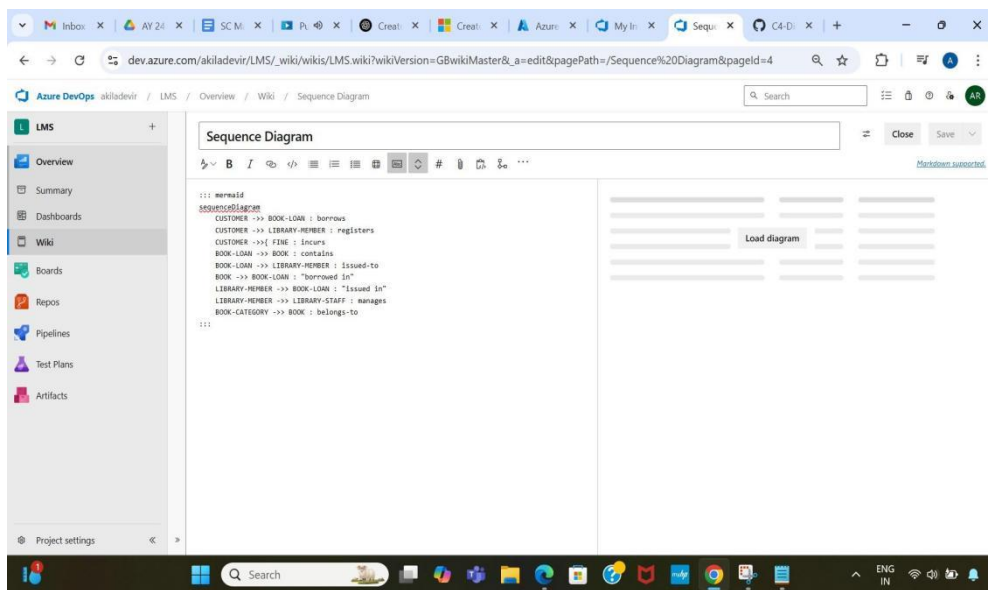
participant defines the entities involved.

->> represents a direct message.

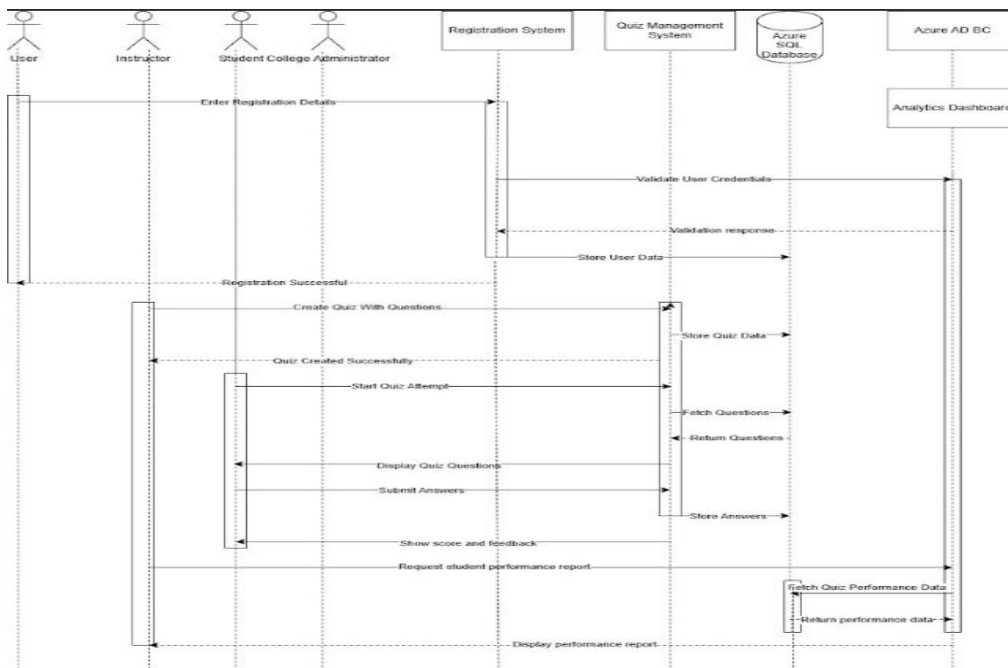
-->> represents a response message.

+ 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.

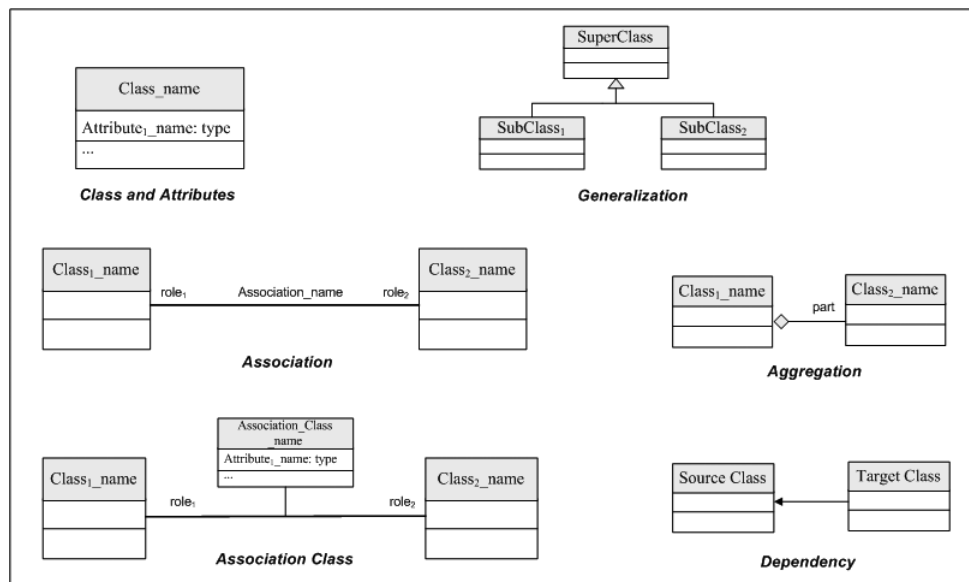
CLASS DIAGRAM

AIM :-

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

THEORY

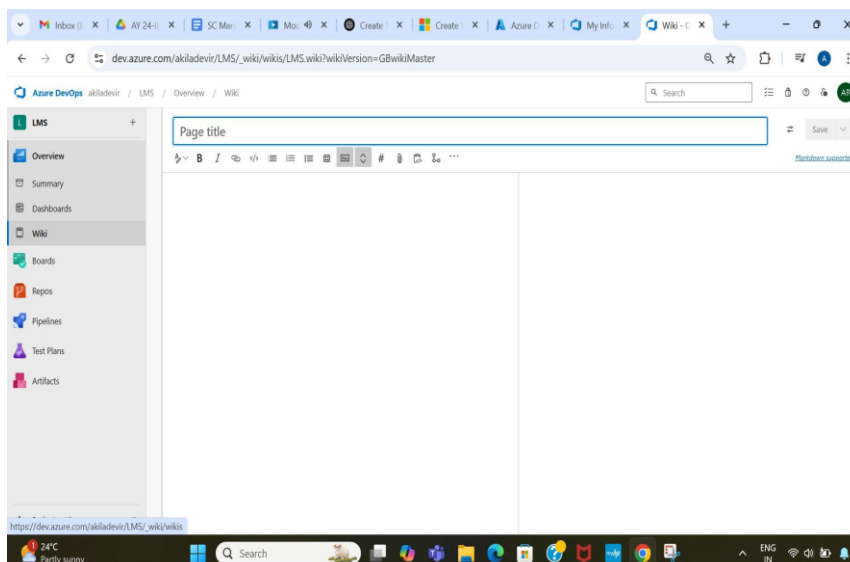
A UML class diagram is a visual tool that represents the structure of a system by showing its classes, attributes, methods, and the relationships between them.



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

classDiagram

```
class User {  
    +String name  
    +String email  
    +register()  
    +startQuiz()  
    +submitAnswers()  
}
```

```
class Instructor {  
    +createQuiz()  
    +reviewPerformance()  
}
```

```
class StudentCollegeAdministrator {  
    +requestReport()  
}
```

```
class RegistrationSystem {  
    +validateCredentials()  
    +storeUserData()  
}
```

```
class QuizManagementSystem {  
    +createQuiz()  
    +fetchQuestions()  
    +storeAnswers()  
    +evaluateQuiz()  
}
```

```
class AzureSQLDatabase {  
    +storeUser()  
    +storeQuiz()  
    +storeAnswers()  
    +fetchQuestions()  
    +fetchPerformanceData()  
}
```

```
class AzureADBC {  
    +validateUser()  
}
```

```
class AnalyticsDashboard {  
    +displayPerformanceReport()  
    +fetchPerformanceData()  
}
```

User --> RegistrationSystem : registers
RegistrationSystem --> AzureADBC : validates
RegistrationSystem --> AzureSQLDatabase : stores user

Instructor --> QuizManagementSystem : creates quiz
QuizManagementSystem --> AzureSQLDatabase : stores quiz

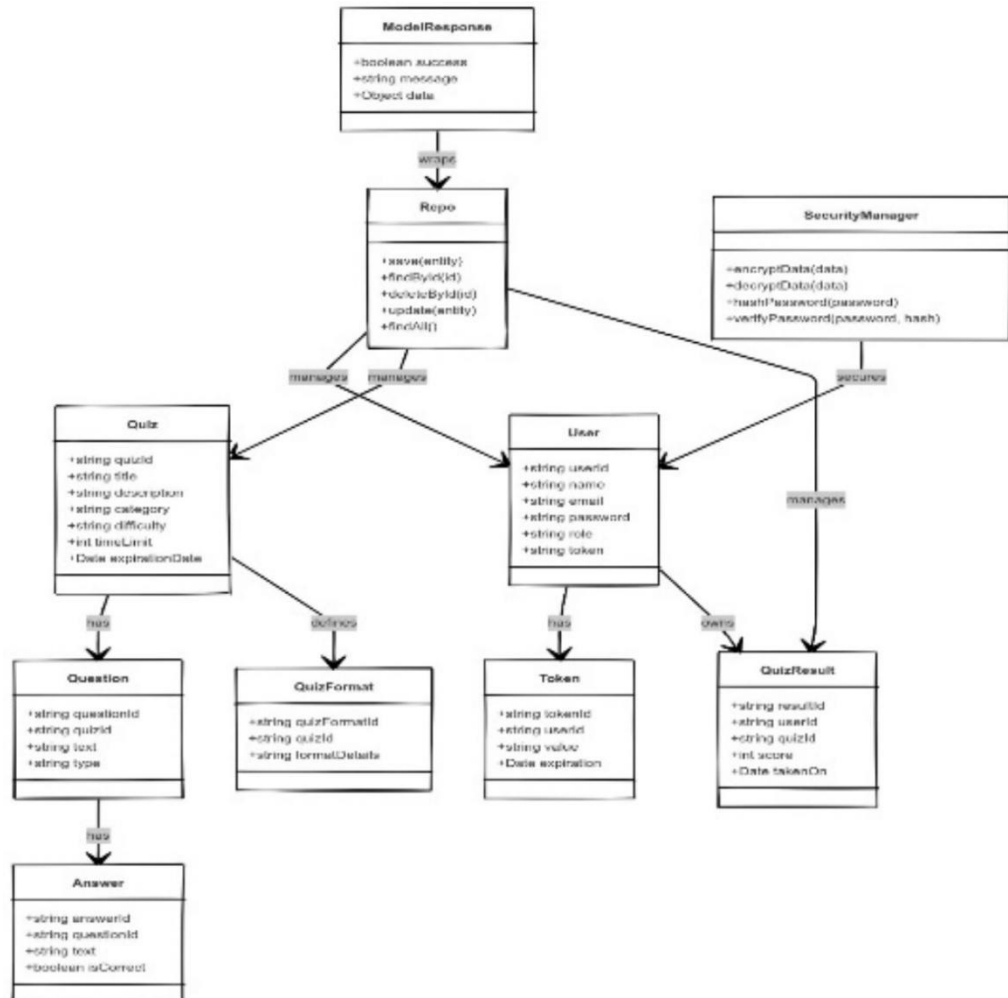
User --> QuizManagementSystem : attempts quiz
QuizManagementSystem --> AzureSQLDatabase : fetches questions
User --> QuizManagementSystem : submits answers
QuizManagementSystem --> AzureSQLDatabase : stores answers

StudentCollegeAdministrator --> AnalyticsDashboard : requests report
AnalyticsDashboard --> AzureSQLDatabase : gets performance data

Relationship Types

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

Class Diagram



Result:

The class diagram was designed successfully.

EX NO: 7

USE CASE DIAGRAM

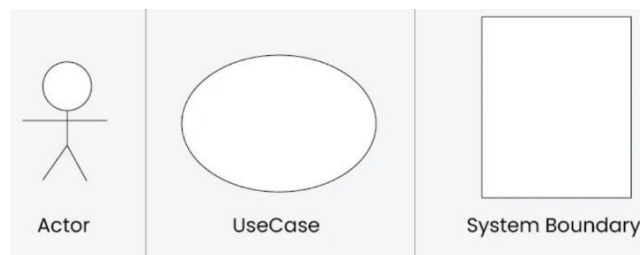
Aim:

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

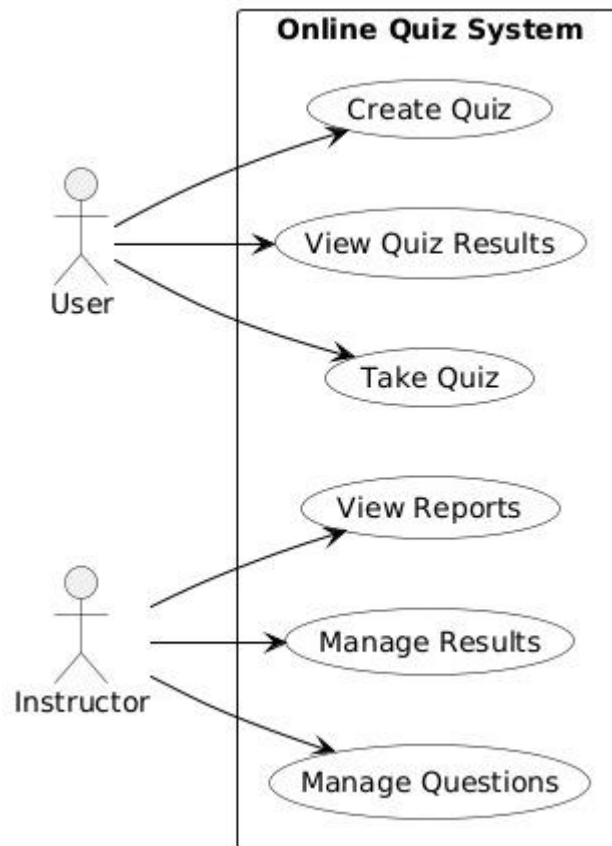
Option 1: Add to Azure DevOps Wiki

- 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.*

Use Case Diagram



Result:



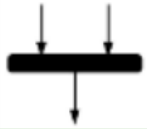








The use case diagram was designed successfully

EX NO. 8**ACTIVITY DIAGRAM****AIM :-**

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

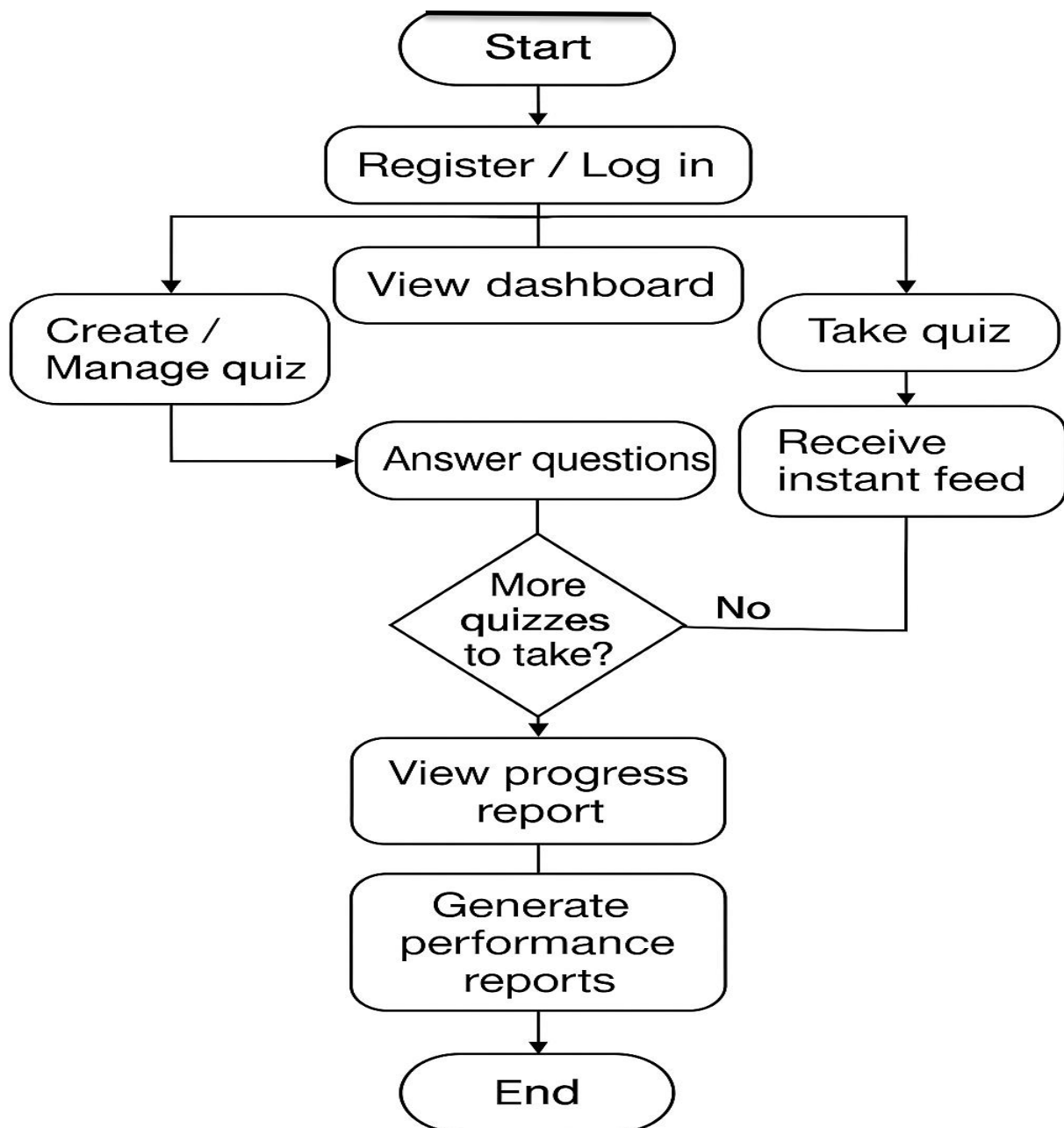
THEORY

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 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



Result:

The activity diagram was designed successfully

EX NO. 9

ARCHITECTURE DIAGRAM

Aim:

Steps to draw the Architecture Diagram using draw.io.

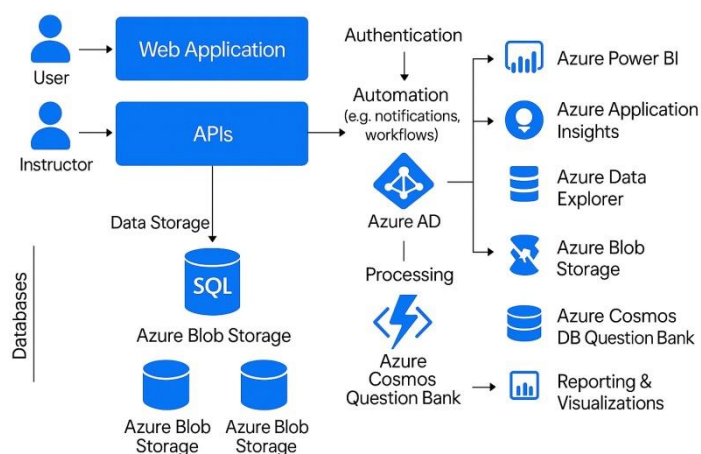
Theory:

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



Result:

The architecture diagram was designed successfully

EX NO. 10

USER INTERFACE

Aim:

Design User Interface for the given project

QuizCraft

Log inSign up

Create an account

Enter your details to get started with QuizCraft

Name

John Doe

Email

you@example.com

Password

I am a...

Student

Create account

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

QuizCraft

Log inSign up

Welcome back

Enter your credentials to access your account

Email

you@example.com

Password

Sign in

[Forgot password?](#)

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

QuizCraft

DashboardQuizzes

SUG

Dashboard

Overview

Quizzes

Progress

Quizzes Completed

5

Out of 12 total

Average Score

78%

Across all quizzes

Upcoming Deadlines

2

In the next 7 days

Recent Activity

Your recently completed and upcoming quizzes

Introduction to Biology

Due: Apr 20, 2025

85%

Chemical Reactions

Due: Apr 25, 2025

Start

World History: Ancient Civilizations

Due: Apr 30, 2025

Start

[View all quizzes ->](#)

QuizCraft

DashboardQuizzes

SJS

Introduction to Biology

0/5 questions answered

29:55

Question 1

1 point

What is the powerhouse of the cell?

☐ Mitochondria

☐ Nucleus

☐ Ribosome

☐ Golgi Apparatus

< Previous

Next >

QuizCraft

DashboardQuizzes

SJS

Quiz Results

Introduction to Biology

Your Score

100%

0/5 points

Excellent work!

You've completed the quiz. Review your answers below to see what you got right and wrong.

Review Your Answers

Question 1 (1 point)

Correct

What is the powerhouse of the cell?

☒ Mitochondria

Your answer

☐ Nucleus

☐ Ribosome

☐ Golgi Apparatus

Explanation:

Mitochondria are organelles that act as the powerhouse of the cell by generating most of the cell's supply of adenosine triphosphate (ATP), used as a source of chemical energy.

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.

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.

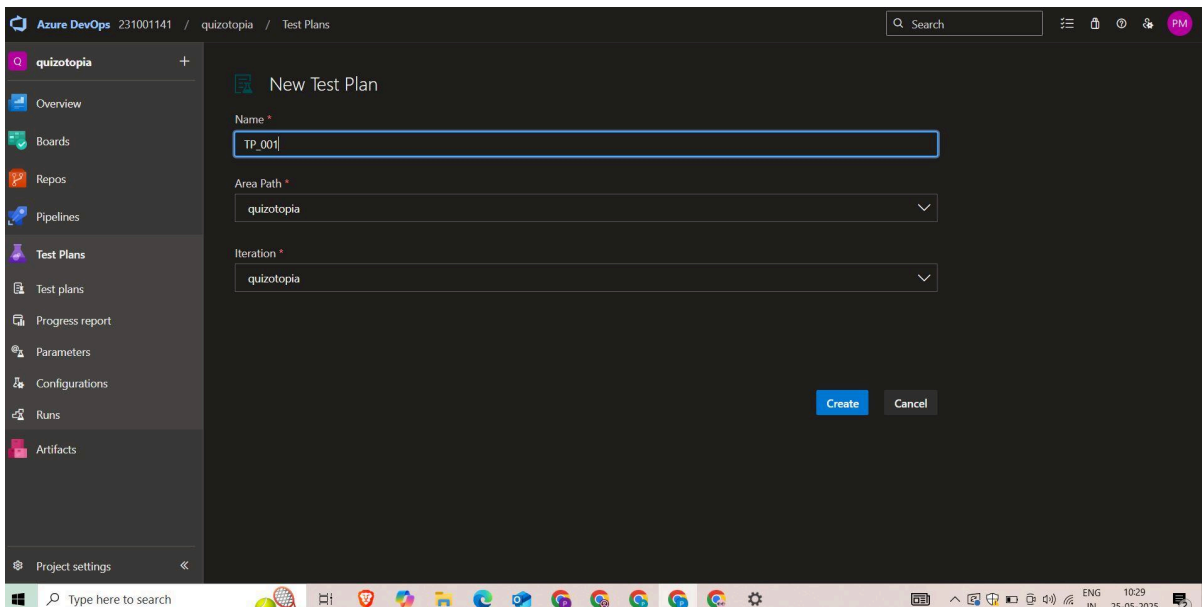
7. Separate Test Suites

- 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.

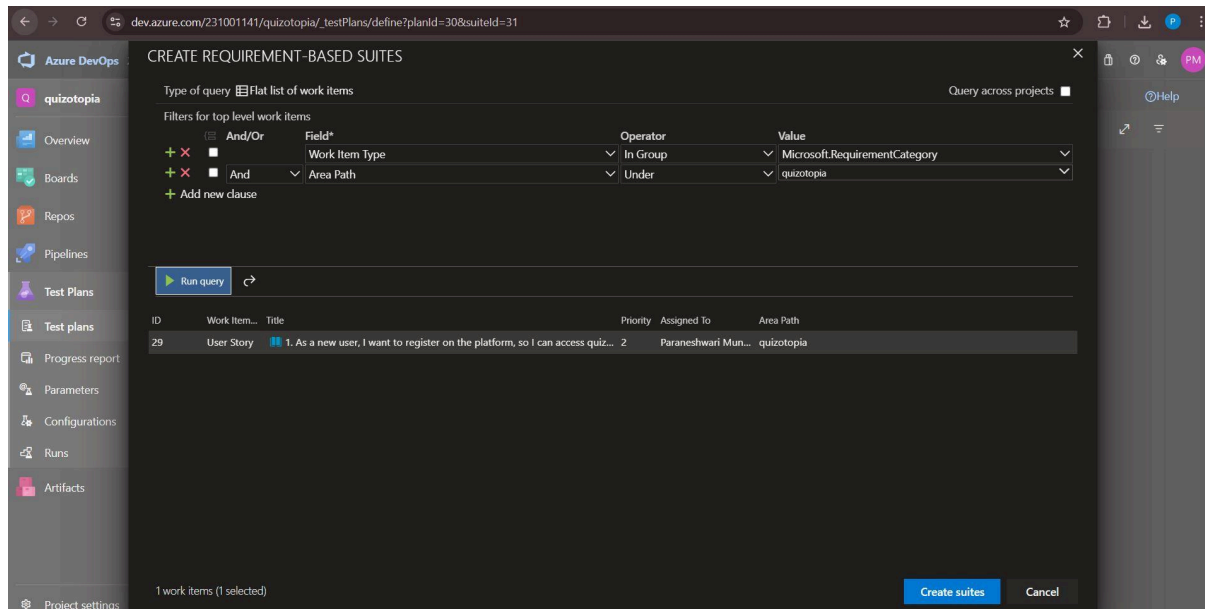
8. Prioritize and Review

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

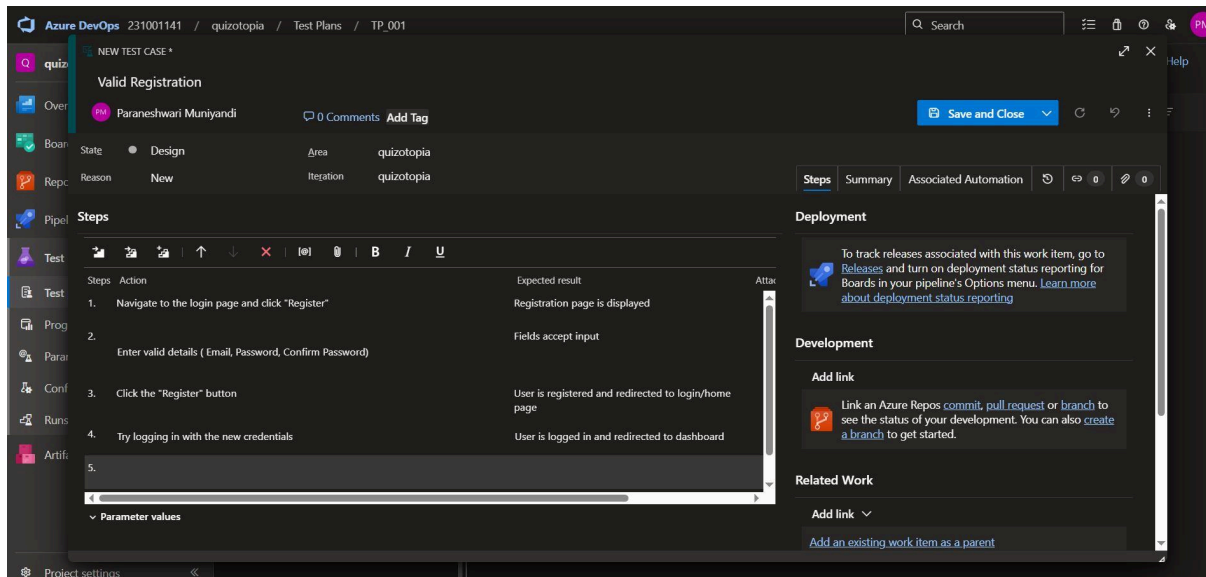
1. New test plan



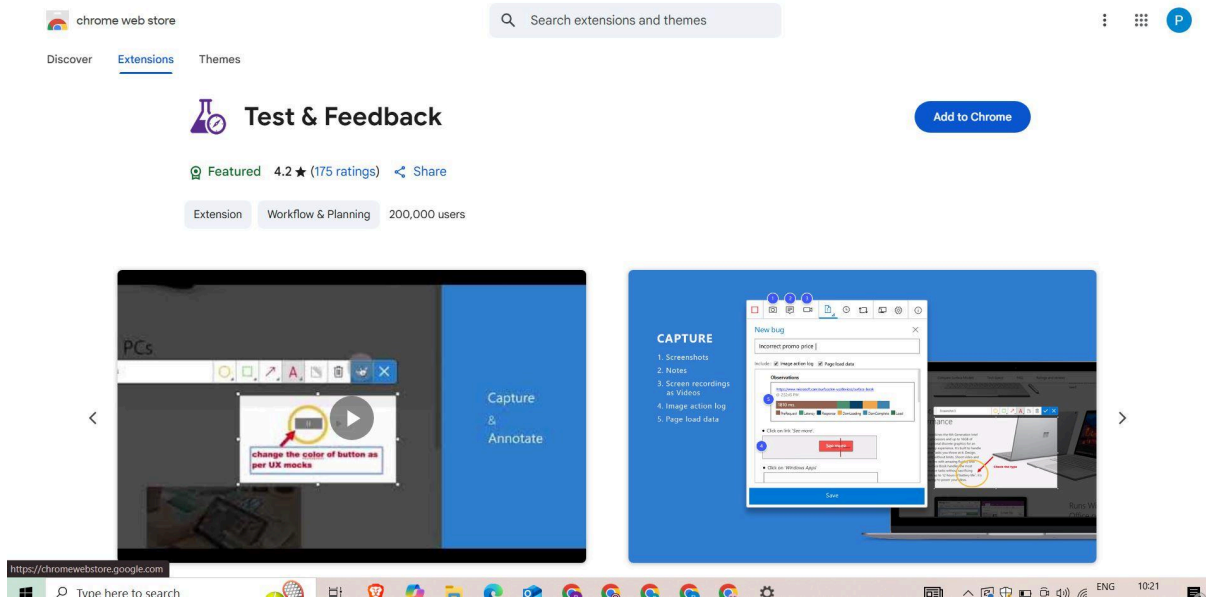
2. Test Suite



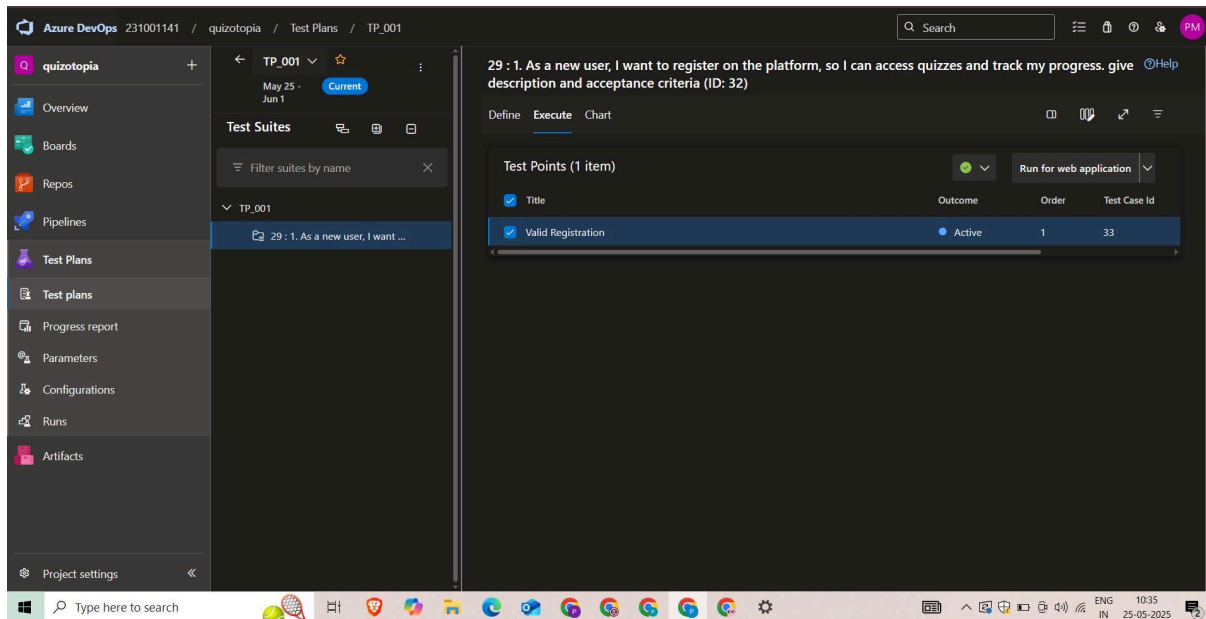
3. Test case



4. Installation of Test & Feedback



5. Running the Test Cases



Runner - Test Plans - Google Chrome

dev.azure.com/231001141/quizotopia/_testExecution/Index

Save and close | Create bug | Search

33: Valid Registration

1.

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

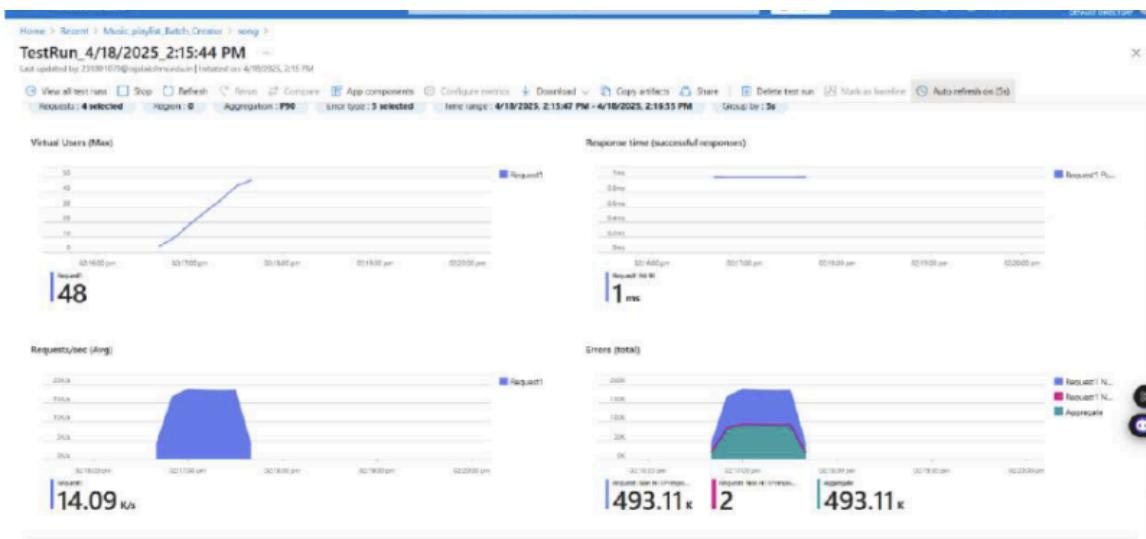
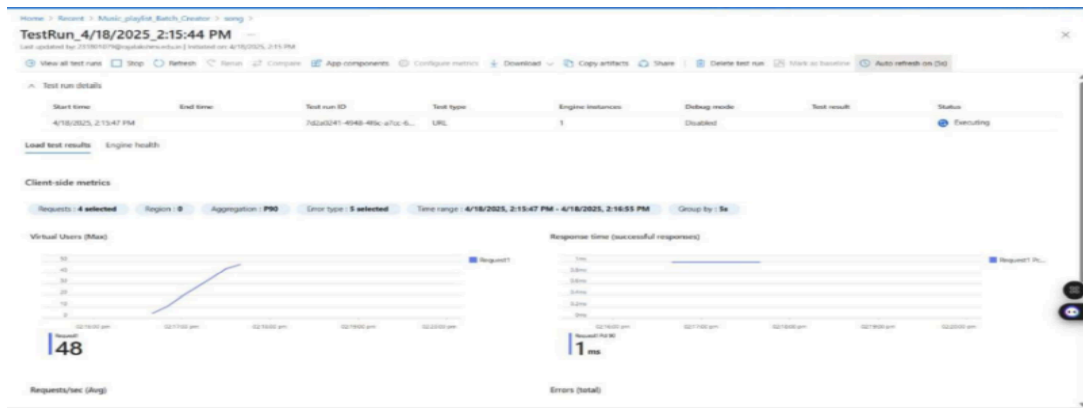
- Test Name: Provide a unique name.
- Description: (Optional) Add test purpose.
- Run After Creation: Keep checked.

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.