

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		

EX NO: 1

DATE: 21/01/2025

STUDY OF AZURE DEVOPS

AIM:

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

STUDY:

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

1. Understanding Azure DevOps

Azure DevOps consists of five key services:

1.1 Azure Repos (Version Control)

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

1.2 Azure Pipelines (CI/CD)

Automates build, test, and deployment processes.

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

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

1.3 Azure Boards (Agile Project Management)

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

1.4 Azure Test Plans (Testing)

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

1.5 Azure Artifacts (Package Management)

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

Getting Started with Azure DevOps

Step 1: Create an Azure DevOps Account

Visit Azure DevOps.

Sign in with a Microsoft Account.

Create an Organization and a Project.

Step 2: Set Up a Repository (Azure Repos)

 Navigate to Repos.

 Choose Git or TFVC for version control.

 Clone the repository and push your code.

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

 Go to Pipelines → New Pipeline.

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

 Define the pipeline using YAML or the Classic Editor.

 Run the pipeline to build and deploy the application.

Step 4: Manage Work with Azure Boards

 Navigate to Boards.

 Create work items, user stories, and tasks.

 Organize sprints and track progress.

Step 5: Implement Testing (Azure Test Plans)

 Go to Test Plans.

 Create and run test cases

 View test results and track bugs.

RESULT:

The study was successfully completed.

EX NO: 2

DATE: 28/01/2025

PROBLEM STATEMENT

AIM:

To prepare Problem Statement for your given project.

PROBLEM STATEMENT:

Educational institutions and organizations need a flexible, scalable solution to deliver high-quality learning experiences beyond the limitations of traditional classrooms. An E-Learning Platform will provide users with access to a variety of courses, quizzes, and certifications anytime and anywhere, enabling self-paced learning. The platform will allow students to track their progress, identify strengths and weaknesses, and receive personalized course recommendations to improve performance. Instructors will be able to create and manage course content easily, while administrators can monitor user activities and system performance, ensuring a smooth, effective, and data-driven learning environment that supports career and academic growth.

RESULT:

The problem statement was written successfully.

EX NO: 3

DATE: 04/02/2025

AGILE PLANNING

AIM:

To prepare an Agile Plan.

THEORY:

Agile Planning for E-Learning Platform Development

Agile planning is essential to building an E-Learning Platform that is flexible, user-centered, and capable of adapting to evolving learning needs. Following the Agile methodology, we avoid rigid, upfront plans and instead use an incremental, iterative approach. Continuous feedback from students, instructors, and administrators will guide improvements throughout the project.

In Agile planning for the E-Learning Platform, the project will be broken down into smaller, manageable pieces of work (features and improvements), with a clear vision in mind. Agile planning will include:

- Roadmaps to guide the platform's feature releases and development schedule.
- Sprints to focus on building and delivering specific modules (e.g., user authentication, course catalog, quizzes, progress tracking).
- Feedback loops to stay responsive to users' needs and quickly adapt the platform.

Tasks will be framed as user stories to capture real requirements from the perspective of platform users (students, teachers, admins). For example:

"As a student, I want to track my progress so that I can stay motivated."

By thinking in terms of user stories, the team ensures that we are always focused on delivering real value to end-users.

Steps in Agile Planning Process for the E-Learning Platform

1. Define Vision

Create a clear vision for the E-Learning Platform: to offer an engaging, interactive, accessible, and personalized learning experience to a global audience.

2. Set Clear Expectations on Goals

Outline project goals, such as:

- Enable seamless course enrollment and management.
- Support multimedia content (videos, PDFs, quizzes).
- Track and display learner progress.

- Facilitate communication between students and instructors.
3. Define and Break Down the Product Roadmap

Develop a roadmap identifying major features (e.g., user authentication, course creation, learning analytics) and prioritize their release.
 4. Create Tasks Based on User Stories

Identify user stories, for example:

 - "As an instructor, I want to create and manage courses."
 - "As a learner, I want to view and complete assigned lessons."
 - "As an admin, I want to manage user access rights."
 5. Populate Product Backlog

Create a backlog listing all user stories, tasks, bugs, and enhancements, prioritized based on importance and user impact.
 6. Plan Iterations and Estimate Effort

Organize work into 1–2 week sprints, estimating the effort for each task using story points or time estimates.
 7. Conduct Daily Stand-Ups

Hold short daily meetings to update on progress, discuss challenges, and plan the day's work.
 8. Monitor and Adapt

Continuously monitor progress, collect user feedback after each sprint, and adjust the backlog and roadmap to better meet users' needs.

RESULT:

Thus, the Agile plan was completed successfully.

EX NO: 4

DATE: 18/02/2025

CREATE USER STORIES WITH ACCEPTANCE

CRITERIA

AIM:

To create User Stories with Acceptance Criteria.

THEORY:

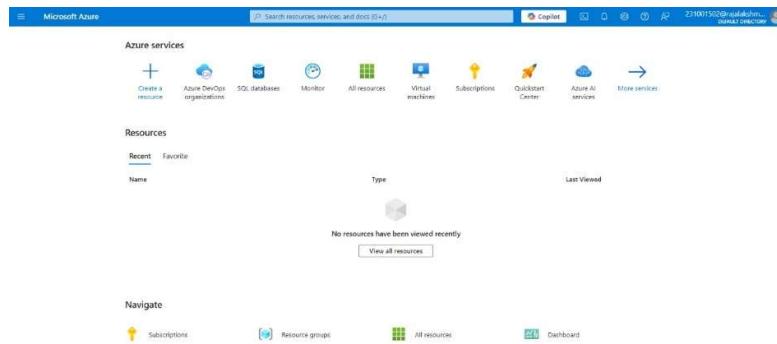
A user story is an informal, general explanation of a software feature written from the perspective of the end user. Its purpose is to articulate how a software feature will provide value to the customer.

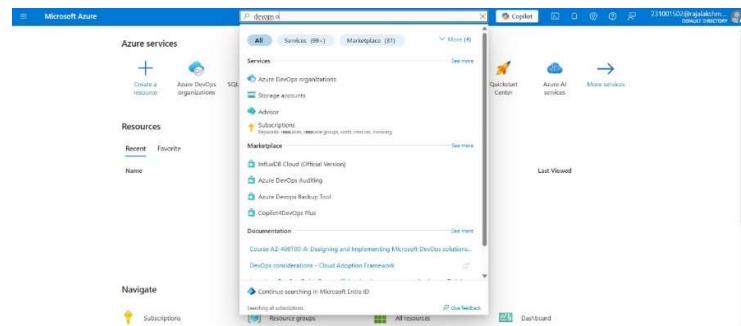
User story template

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

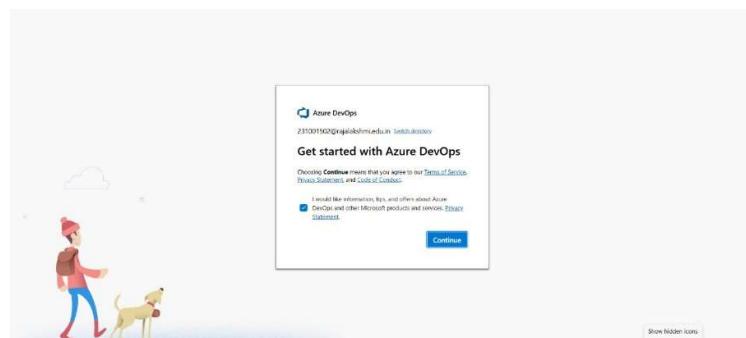
PROCEDURE:

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





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

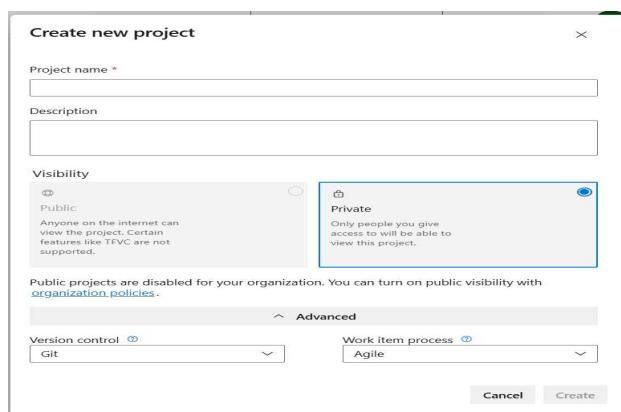


6. Create the First Project in Your Organization

After the organization is set up, you'll need to create your first **project**.

This is where you'll begin to manage code, pipelines, work items, and more.

- i. On the organization's **Home page**, click on the **New Project** button.
- ii. Enter the project name, description, and visibility options:
 - **Name:** Choose a name for the project (e.g., **LMS**).
 - **Description:** Optionally, add a description to provide more context about the project.
 - **Visibility:** Choose whether you want the project to be **Private** (accessible only to those invited) or **Public** (accessible to anyone).
- iii. Once you've filled out the details, click **Create** to set up your first project.



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

8. Project dashboard

The screenshot shows the Azure DevOps project dashboard for 'E-Learning Platform'. The left sidebar includes links for Overview, Summary, Dashboards, Wiki, Boards, Repos, Pipelines, and Artifacts. The main area displays 'About this project' with a 'Help others to get on board!' section and a 'Project stats' panel showing 2 Work items and 0 Work items. A 'Members' section lists six team members with their icons. At the bottom, there's a 'Project settings' link.

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

The screenshot shows the 'User Story' creation form for 'E-Learning Platform'. The top bar includes a search bar and project settings. The main form has a title '3 As a new user, I want to sign up using email or Google authentication, so that I can access courses easily.' and a note 'No one selected'. It has sections for 'Description', 'Acceptance Criteria', 'Planning', 'Deployment', 'Development', and 'Related Work'. The 'Description' section contains a detailed registration requirement. The 'Acceptance Criteria' section lists three items. The 'Planning' section shows 'Story Points' (2), 'Priority' (2), and 'Risk'. The 'Deployment' section has a note about tracking releases. The 'Development' section has a note about linking to Azure Repos. The 'Related Work' section shows a link to 'User Registration & Login'.

USER STORY 4

4 As a returning user, I want to log in securely using my credentials, so that I can resume learning from where I left off.

No one selected Comments Add Tag

Story Points: 1
Priority: 2
Risk: 2

Description

The login system should authenticate users via email/password or Google authentication. It must handle incorrect credentials graciously and support "Remember Me" functionality for convenience.

Acceptance Criteria

1. Users must be able to log in using their registered email and password.
2. Incorrect credentials must display an error message.
3. A "Remember Me" option should be available for faster logins.

Discussion

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

USER STORY 11

11 As an instructor, I want to create, edit, and delete courses, so that I can manage my content easily.

No one selected Comments Add Tag

Story Points: 1
Priority: 2
Risk: 2

Description

Instructors should have a dashboard where they can add new courses by providing titles, descriptions, categories, and media (like images). They should also be able to update or remove courses as needed.

Acceptance Criteria

1. Instructors must be able to add a title, description, category and thumbnail to courses.
2. The system should allow adding, editing, and deleting course modules and lessons.
3. Instructors should be able to preview the course before publishing it.

Discussion

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

USER STORY 14

14 As an admin, I want to approve or reject courses, so that I can maintain content quality.

No one selected Comments Add Tag

Story Points: 1
Priority: 2
Risk: 2

Description

Every newly created course must be sent for admin approval. The admin can review course details and either approve it for public availability or reject it with comments explaining the decision.

Acceptance Criteria

1. Courses must have a pending approval status before being published.
2. Admins must be able to approve or reject courses and provide feedback.
3. Approved courses must become visible to students.

Discussion

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

USER STORY 19

19 As an instructor, I want to schedule live sessions, so that I can teach students in real time.

No one selected Comments Add Tag

Story Points: 1
Priority: 2
Risk: 2

Description

Instructors should be able to schedule live classes by specifying the date, time, duration, and topic. Students should be automatically notified and able to join sessions easily.

Acceptance Criteria

1. Instructors must schedule sessions with date, time, and duration.
2. Notifications should be sent to enrolled students.
3. Reminder notifications must trigger 30 minutes before the session.

Discussion

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

USER STORY 17

17 As a student, I want to receive notifications for upcoming live classes, so that I don't miss important sessions.

No one selected 0 Comments Add Tag

State: New Area: E-Learning Platform
Reason: None Iteration: E-Learning Platform/Iteration 1

Description
Students should receive timely alerts for scheduled live classes through email and in-app notifications. A calendar view must help them track upcoming sessions.

Acceptance Criteria

- Notifications must be sent upon scheduling a new session.
- A calendar view should list all upcoming classes.
- Clicking a class should display the joining instructions.

Discussion

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

[switch to Markdown editor](#)

Planning
Story Points: 2
Priority: 2
Risk: 1

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.

Related Work
Add link
Parent: 11 Live Class Scheduling
Updated Apr 1 0 New

USER STORY 18

18 As a student, I want to take quizzes and submit assignments, so that I can test my knowledge.

No one selected 0 Comments Add Tag

State: New Area: E-Learning Platform
Reason: None Iteration: E-Learning Platform/Iteration 1

Description
Students should access quizzes after completing course modules. Quizzes must include multiple-choice, short answers, and coding questions. Assignments should support file submissions.

Acceptance Criteria

- Quiz must support multiple question types.
- Timer should be available for timed quizzes.
- Instant feedback must be shown after submission.

Discussion

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

[switch to Markdown editor](#)

Planning
Story Points: 2
Priority: 2
Risk: 1

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.

Related Work
Add link
Parent: 10 Quiz & Assignment Module
Updated Apr 1 0 New

USER STORY 20

20 As an instructor, I want to create quizzes with multiple question types, so that I can assess students effectively.

Saravanan G 0 Comments Add Tag

State: New Area: E-Learning Platform
Reason: None Iteration: E-Learning Platform/Iteration 1

Description
Instructors should be able to create quizzes by adding different types of questions (MCQs, short answers, coding). They should also define passing criteria for course completion.

Acceptance Criteria

- Allow adding various question formats.
- Define passing marks/criteria for quizzes.
- Save quiz results to the student's profile.

Discussion

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

[switch to Markdown editor](#)

Planning
Story Points: 2
Priority: 2
Risk: 1

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.

Related Work
Add link
Parent: 10 Quiz & Assignment Module
Updated Apr 1 0 New

USER STORY 23

23 As a student, I want to receive a certificate upon course completion, so that I can showcase my achievement.

No one selected 0 Comments Add Tag

State: New Area: E-Learning Platform
Reason: None Iteration: E-Learning Platform/Iteration 1

Description
After successful course completion, the system must generate a downloadable and shareable certificate containing the student's name, course title, completion date, and a verification link.

Acceptance Criteria

- Certificates should be auto-generated after course completion.
- Certificates must include student name, course title, and date.
- Downloadable and shareable certificate formats must be available.

Discussion

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

[switch to Markdown editor](#)

Planning
Story Points: 2
Priority: 2
Risk: 1

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.

Related Work
Add link
Parent: 12 Certification Generation
Updated Apr 1 0 New

RESULT:

The user story was written successfully.

EX NO: 5

DATE: 25/02/2025

DESIGNING SEQUENCE DIAGRAMS USING AZURE DEVOPS-WIKI

AIM:

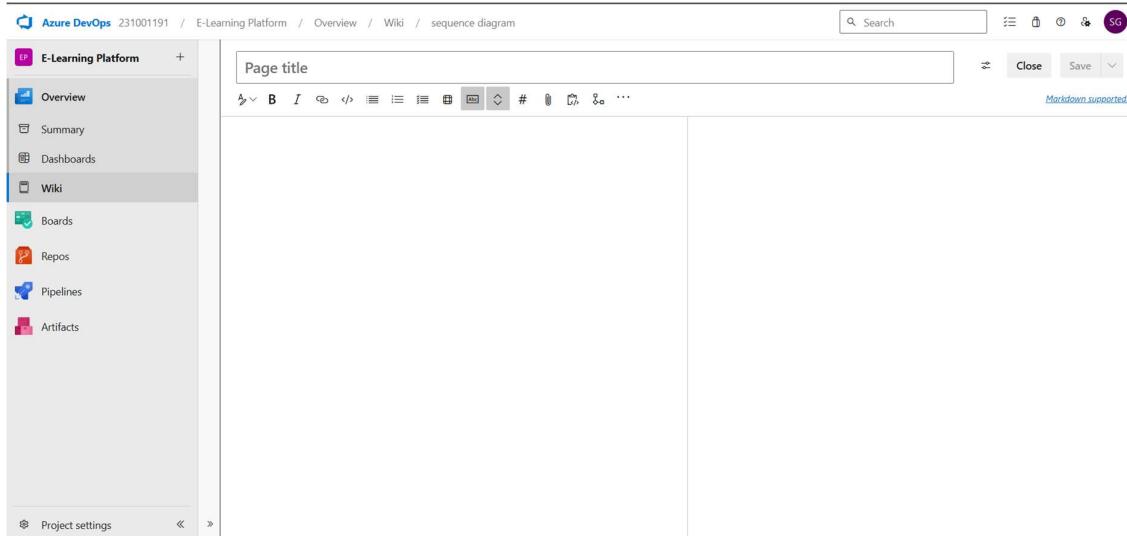
To design a Sequence Diagram by using Azure DevOps-WIKI

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 Frontend
    participant Backend
    participant Database
```

```

User->>Frontend: Click "Enroll" button
Frontend->>Backend: POST /enroll (userId, courseId)
Backend->>Database: Check enrollment status
alt User is not enrolled
    Database-->>Backend: Not enrolled
    Backend-->>Database: Insert enrollment record
    Database-->>Backend: Enrollment confirmed
    Backend-->>Frontend: 200 OK (Success)
    Frontend-->>User: Show success message
else User already enrolled
    Database-->>Backend: Already enrolled
    Backend-->>Frontend: 409 Conflict
    Frontend-->>User: Show "Already enrolled" message
end
:::

```

Explanation:

participant defines the entities involved.

->> represents a direct message.

-->> represents a response message.

+ after ->> activates a participant.

- after -->> deactivates a participant.

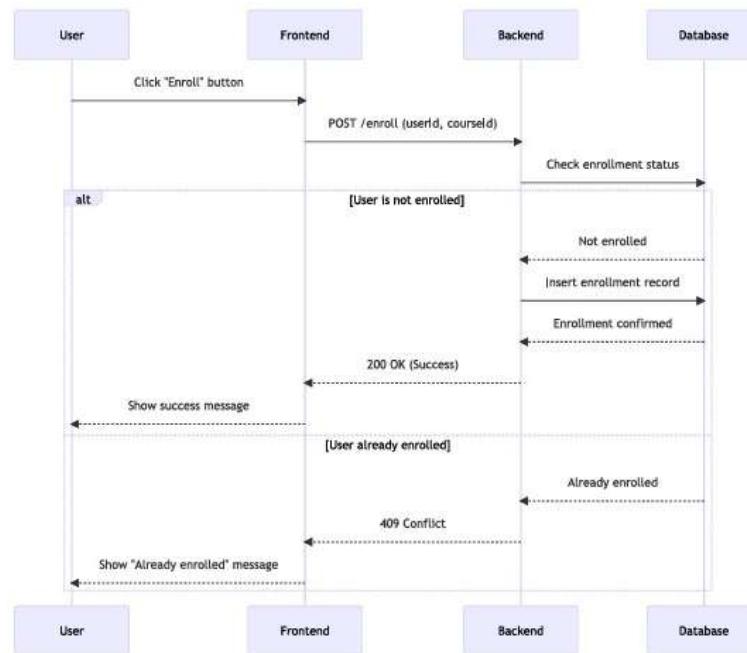
alt / else for conditional flows.

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)

The screenshot shows the Azure DevOps Wiki sequence diagram editor. The left sidebar has a 'Wiki' section selected. The main area displays a sequence diagram titled 'sequence diagram'. The diagram shows interactions between 'User', 'Frontend', 'Backend', and 'Database' participants. The sequence starts with 'User->Frontend: Click "Enroll" button', followed by 'Frontend->Backend: POST /enroll (userId, courseId)'. The 'Backend' participant then sends a 'Check enrollment status' message to the 'Database'. An 'alt' block handles two cases: '[User is not enrolled]' and '[User already enrolled]'. For '[User is not enrolled]', the 'Backend' inserts an enrollment record ('Insert enrollment record') and returns 'Enrollment confirmed' to the 'Frontend', which then returns a '200 OK (Success)' response and a 'Show success message' to the 'User'. For '[User already enrolled]', the 'Backend' returns 'Already enrolled' to the 'Frontend', which then returns a '409 Conflict' response and a 'Show "Already enrolled" message' to the 'User'. The 'Load diagram' button is visible at the bottom right of the editor.

4. click wiki menu and select the page



RESULT:

Thus, the sequence diagram was created successfully.

EX NO. 6

DATE: 04/03/2025

DESIGNING CLASS DIAGRAM USING AZURE

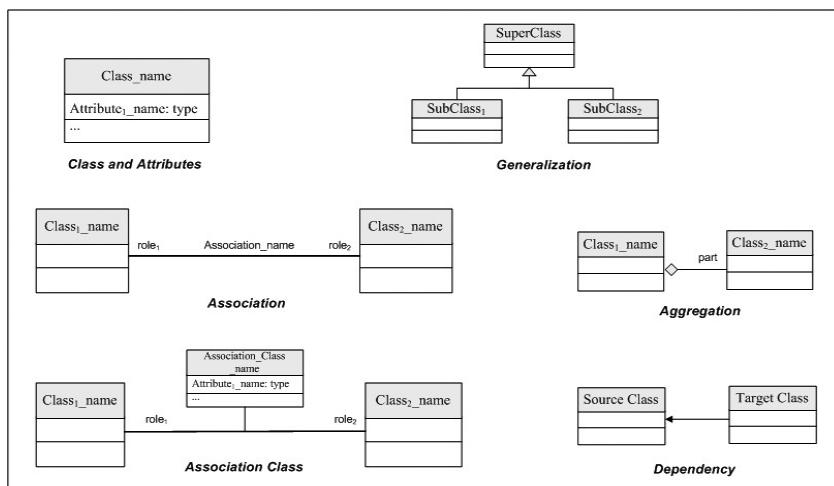
DEVOPS-WIKI

AIM:

To design a Class Diagram by using Azure DevOps-WIKI

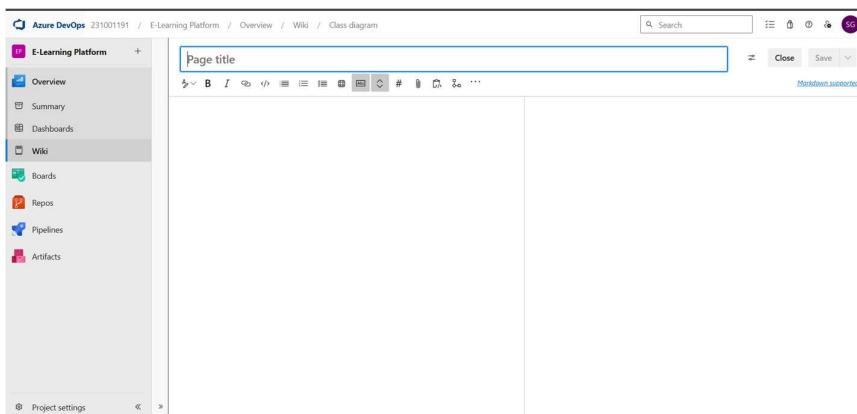
THEORY:

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.



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

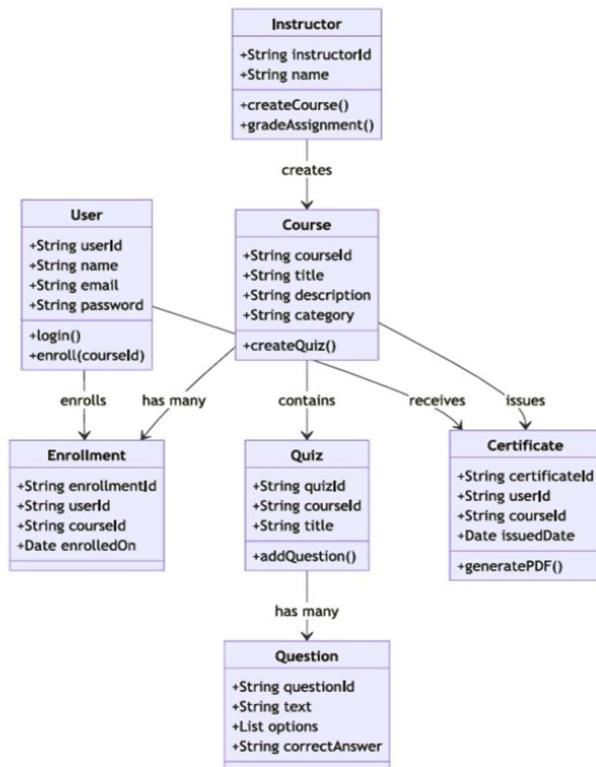
```
::: mermaid
classDiagram
    class User {
        +String userId
        +String name
        +String email
        +String password
        +login()
        +enroll(courseId)
    }
    class Instructor {
        +String instructorId
        +String name
        +createCourse()
        +gradeAssignment()
    }
    class Course {
        +String courseId
        +String title
        +String description
        +String category
        +createQuiz()
    }
    class Enrollment {
        +String enrollmentId
        +String userId
        +String courseId
        +Date enrolledOn
    }
    class Quiz {
        +String quizId
        +String courseId
        +String title
        +addQuestion()
    }
    class Question {
        +String questionId
        +String text
        +List<String> options
        +String correctAnswer
    }
```

```

class Certificate {
    +String certificateId
    +String userId
    +String courseId
    +Date issuedDate
    +generatePDF()
}

User --> Enrollment : enrolls
Course --> Enrollment : has many
Instructor --> Course : creates
Course --> Quiz : contains
Quiz --> Question : has many
User --> Certificate : receives
Course --> Certificate : issues
:::

```



RESULT:

Thus, the class diagram was created successfully.

EX NO: 7

DATE: 11/03/2025

DESIGNING USE CASE DIAGRAM USING AZURE

DEVOPS-WIKI

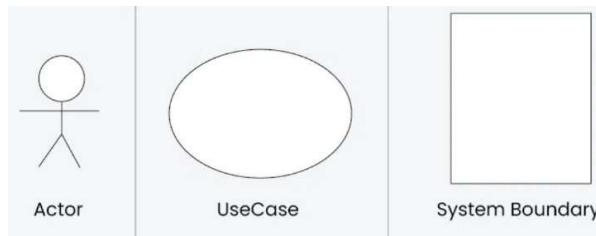
AIM:

Steps to draw the Use Case Diagram using Azure DevOps -WIKI

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: Open Azure DevOps Wiki

- Go to your Azure DevOps project.
- From the left sidebar, click on Wiki.

Step 2: Create or Edit a Wiki Page

- Click New Page to create a new page.
- Or, navigate to an existing page and click the Edit () button.

Step 3: Insert a Mermaid Code Block

Use a fenced code block with the mermaid keyword:

::: mermaid

graph TD

actorUser([User])

actorInstructor([Instructor])

actorAdmin([Admin])

actorUser --> UC1[Register/Login]

actorUser --> UC2[Browse Courses]

actorUser --> UC3[Enroll in Course]

actorUser --> UC4[Attend Live Class]

actorUser --> UC5[Take Quiz]

actorUser --> UC6[Download Certificate]

actorInstructor --> UC1

actorInstructor --> UC7[Create Course]

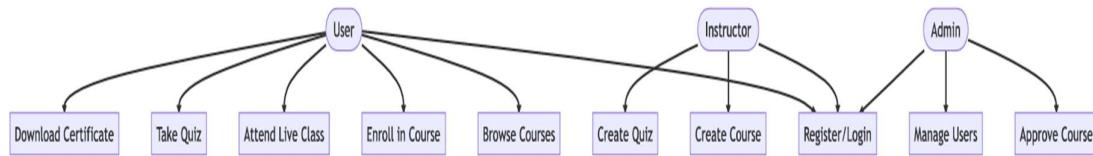
actorInstructor --> UC8[Create Quiz]

actorAdmin --> UC1

actorAdmin --> UC9[Approve Course]

actorAdmin --> UC10[Manage Users]

:::



RESULT:

The Use Case diagram was designed successfully.

EX NO. 8

DATE: 18/03/2025

DESIGNING ACTIVITY DIAGRAMS USING AZURE

DEVOPS-WIKI

AIM:

To design 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	(H)	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:

Step 1: Open Azure DevOps Wiki

- Go to your Azure DevOps project.
- In the left menu, click Wiki.

Step 2: Create or Edit a Wiki Page

- Click New Page or open an existing page.

- Click the Edit () button to begin editing.

Step 3: Insert a Mermaid Code Block

Use triple backticks and mermaid to insert the diagram:

```
::: mermaid
```

```
stateDiagram-v2
```

```
[*] --> Login
```

```
Login --> Dashboard : Success
```

```
Login --> Error : Failure
```

```
Error --> Login
```

```
Dashboard --> BrowseCourses
```

```
BrowseCourses --> SelectCourse
```

```
SelectCourse --> Enroll : Not Enrolled
```

```
SelectCourse --> AccessContent : Already Enrolled
```

```
Enroll --> AccessContent
```

```
AccessContent --> ViewLectures
```

```
ViewLectures --> TakeQuiz
```

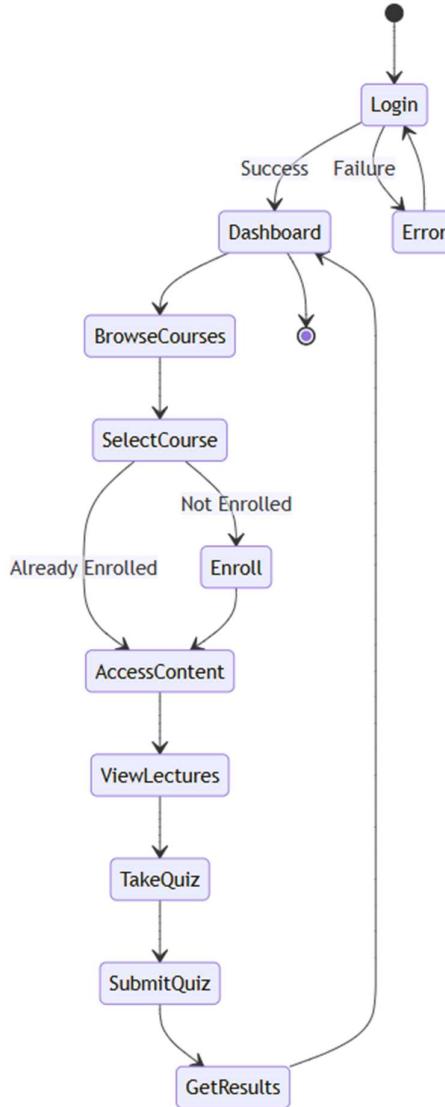
```
TakeQuiz --> SubmitQuiz
```

```
SubmitQuiz --> GetResults
```

```
GetResults --> Dashboard
```

```
Dashboard --> [*]
```

```
:::
```



RESULT:

The activity diagram was designed successfully.

EX NO. 9

DATE: 25/03/2025

DESIGNING ARCHITECTURE DIAGRAM

USING STAR UML

AIM:

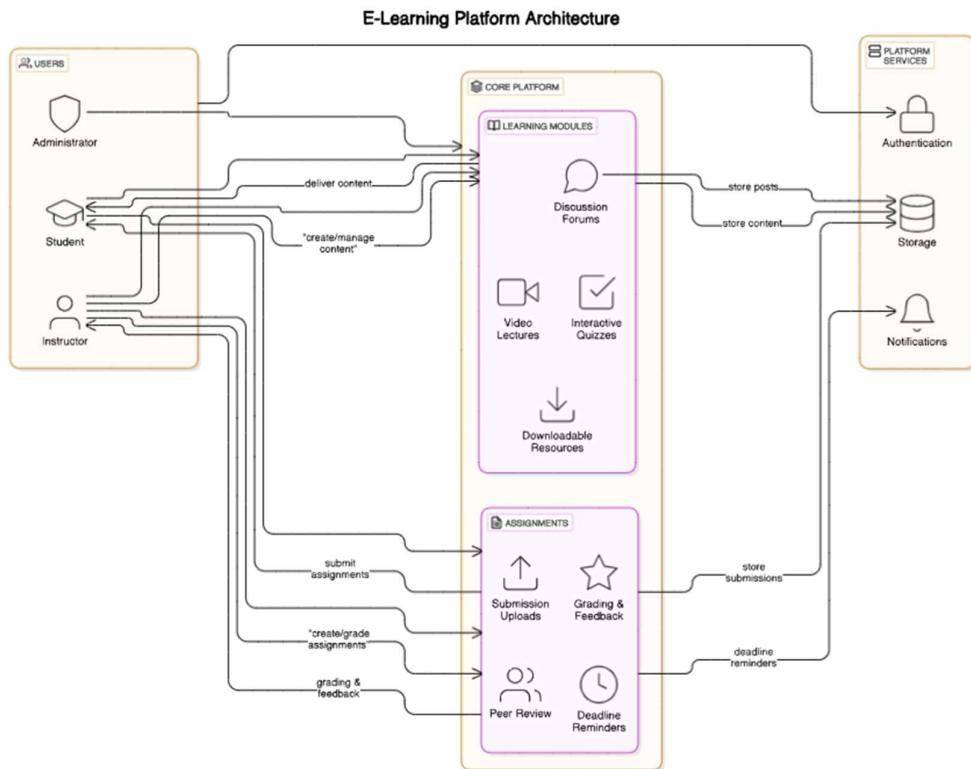
Steps to draw the Architecture Diagram using Star UML

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 Star UML



RESULT:

The architecture diagram was designed successfully.

EX NO. 10

DATE: 01/04/2025

DESIGN USER INTERFACE

AIM:

Design User Interface for the given project.

The image displays three vertically stacked screenshots of a user interface design for an 'E-Learning Platform'.
The top screenshot shows a 'Login' screen with a purple header containing the platform name. It features a central login form with fields for 'Email:' and 'Password:', and a teal 'Login' button. Below the form is a link to 'Sign up'.
The middle screenshot shows the main landing page with a purple header and a large central area titled 'Learn Anything, Anytime' with a subtext 'Join thousands of learners worldwide'. It includes a teal 'Get Started' button. The background has a gradient from purple at the top to teal at the bottom.
The bottom screenshot shows a section titled 'Popular Courses' with three cards: 'Web Development' (Learn HTML, CSS, JavaScript and more), 'Data Science' (Learn Python, Machine Learning), and 'Graphic Design' (Learn Photoshop, Illustrator). Each card has a teal 'Enroll Now' button. Below this is an 'About Us' section with a brief description and a 'Contact' section with an email address. A copyright notice is at the bottom of this page.

RESULT:

The UI was designed successfully.

EX NO. 11

DATE: 08/04/2025

IMPLEMENTATION – DESIGN A WEB PAGE BASED ON SCRUM METHODOLOGY

AIM:

To implement the given project based on Agile Methodology.

PROCEDURE:

Step 1: Set Up an Azure DevOps Project

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

Step 2: Add Your Web Application Code

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

```
git clone <repo_url>
cd <repo_folder>
```

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

```
git add .
git commit -m "Initial commit"
git push origin main
```

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

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

```
trigger:
  - main
pool:
  vmImage: 'ubuntu-latest'
```

```

steps:
  task: UseNode@1
    inputs:
      version: '16.x'
    -script: npm install
      displayName: 'Install dependencies'
    -script: npm run build
      displayName: 'Build application'
  -task: PublishBuildArtifacts@1
    inputs:
      pathToPublish: 'dist'
      artifactName: 'drop'

```

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

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

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

RESULT:

Thus, the application was successfully implemented.

EX.NO: 12

DATE: 15/04/2025

TESTING-TEST PLAN, TEST CASE AND 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 behaviours 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.

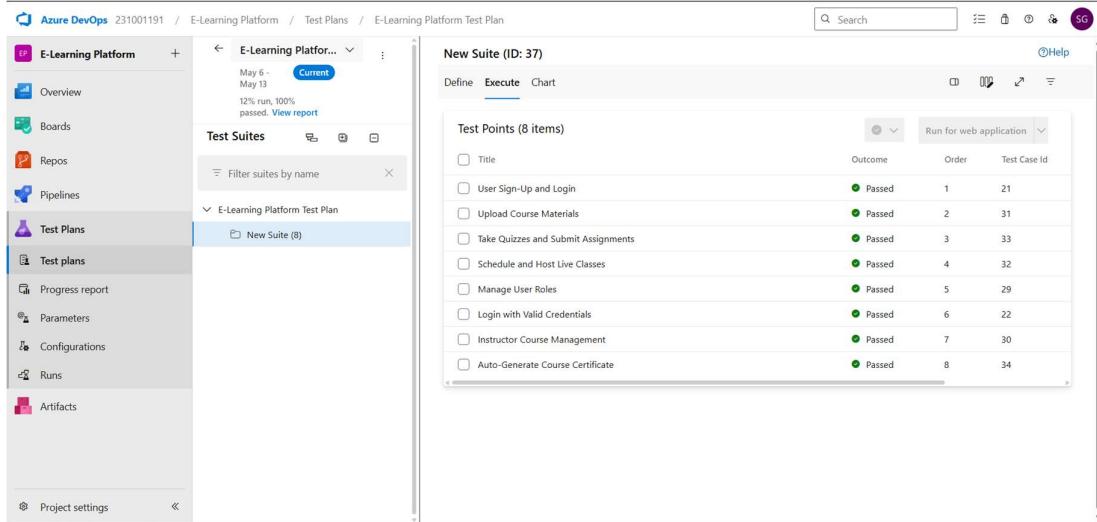
7. Separate Test Suites

- 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



The screenshot shows the Azure DevOps interface for managing test plans. On the left, there's a sidebar with options like Overview, Boards, Repos, Pipelines, Test Plans, Test plans, Progress report, Parameters, Configurations, Runs, and Artifacts. The 'Test Plans' section is currently selected. In the center, under 'Test Plans', it says 'E-Learning Platform' and 'May 6 - May 13'. Below that is a 'Test Suites' section with a 'New Suite (8)' option highlighted. To the right, a detailed view of a 'New Suite (ID: 37)' is shown. This view includes tabs for 'Define', 'Execute' (which is selected), and 'Chart'. Under 'Execute', there's a table titled 'Test Points (8 items)' with columns for 'Title', 'Outcome' (all marked as 'Passed'), 'Order' (ranging from 1 to 8), and 'Test Case Id'. The table lists eight test points: User Sign-Up and Login, Upload Course Materials, Take Quizzes and Submit Assignments, Schedule and Host Live Classes, Manage User Roles, Login with Valid Credentials, Instructor Course Management, and Auto-Generate Course Certificate.

Title	Outcome	Order	Test Case Id
User Sign-Up and Login	Passed	1	21
Upload Course Materials	Passed	2	31
Take Quizzes and Submit Assignments	Passed	3	33
Schedule and Host Live Classes	Passed	4	32
Manage User Roles	Passed	5	29
Login with Valid Credentials	Passed	6	22
Instructor Course Management	Passed	7	30
Auto-Generate Course Certificate	Passed	8	34

2. Test case

The screenshot shows a Microsoft Azure DevOps work item titled "TEST CASE 30". The work item details are as follows:

- State:** Design
- Area:** E-Learning Platform
- Iteration:** E-Learning Platform\Iteration 1
- Reason:** New

The "Steps" section contains the following test cases:

Steps	Action	Expected result	Attachments
1.	Instructor creates a new course with title, description, and thumbnail	Course is saved or submitted for approval	
2.	Edit existing course details and save	Changes are updated successfully	
3.	Delete a course	Course is removed from instructor's dashboard	
4.	Login as student and view course list	Only approved courses are visible	

Below the steps, there is a note: "Click or type here to add a step".

The right side of the screen displays sections for Deployment, Development, and Related Work.

- Deployment:** A note about tracking releases and deployment status reporting.
- Development:** A note about linking to Azure Repos and creating branches.
- Related Work:** A list of related work items, including one titled "As an instructor, I want to create, edit, and delete courses, so that...".

3. Installation of Test

The screenshot shows the "Test & Feedback" extension page on the Chrome Web Store. The page includes the following information:

- Rating:** 4.2 stars (175 ratings)
- Users:** 200,000 users
- Category:** Extension, Workflow & Planning
- Developer:** PC's
- Description:** Capture & Annotate
- Features:** Screenshots, Notes, Screen recording, Webcam, Image action log, Page load data.
- Install button:** Add to Chrome

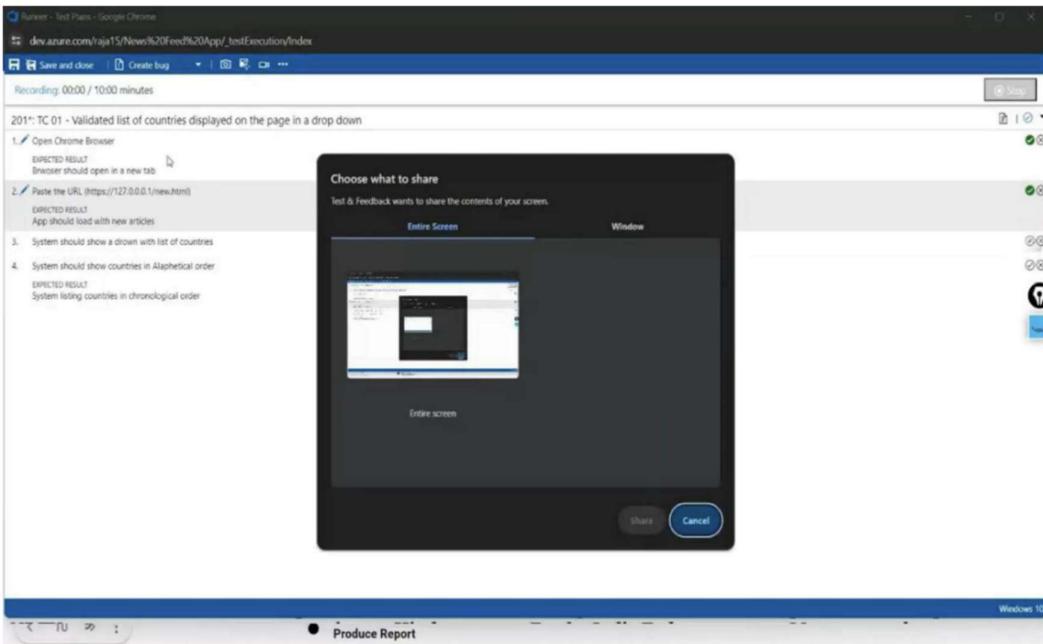
Two screenshots of the extension are shown:

- Capture & Annotate:** Shows a screenshot of a video player interface with a red annotation overlay.
- Capture:** Shows a screenshot of a "New Log" dialog box with various settings and a preview window.

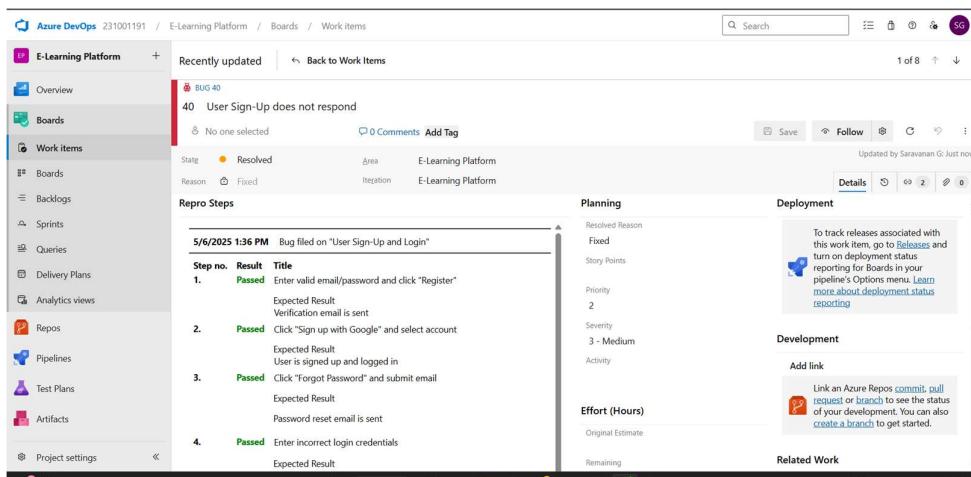
4. Running the Test Cases



5. Recording the Test Cases



6. Creating Bugs



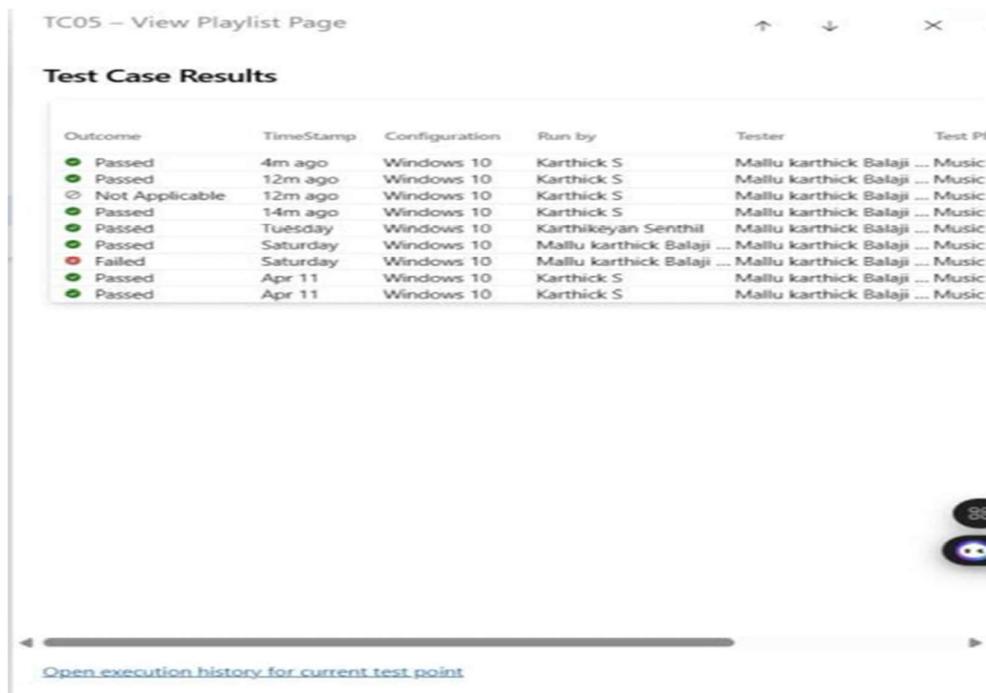
7. Test Case Results

TC05 – View Playlist Page

Test Case Results

Outcome	TimeStamp	Configuration	Run by	Tester	Test PI
Passed	4m ago	Windows 10	Karthick S	Mallu karthick Balaji ... Music	
Passed	12m ago	Windows 10	Karthick S	Mallu karthick Balaji ... Music	
Not Applicable	12m ago	Windows 10	Karthick S	Mallu karthick Balaji ... Music	
Passed	14m ago	Windows 10	Karthick S	Mallu karthick Balaji ... Music	
Passed	Tuesday	Windows 10	Karthikeyan Senthil	Mallu karthick Balaji ... Music	
Passed	Saturday	Windows 10	Mallu karthick Balaji ...	Mallu karthick Balaji ... Music	
Failed	Saturday	Windows 10	Mallu karthick Balaji ...	Mallu karthick Balaji ... Music	
Passed	Apr 11	Windows 10	Karthick S	Mallu karthick Balaji ... Music	
Passed	Apr 11	Windows 10	Karthick S	Mallu karthick Balaji ... Music	

Open execution history for current test point



8. Progress Report

dev.azure.com/2116231001204/Social%20Network/_testManagement/analytics/progressreport

Progress report

Demo_Day Test Suites Outcome Configuration Tester Priority Assigned To

Summary

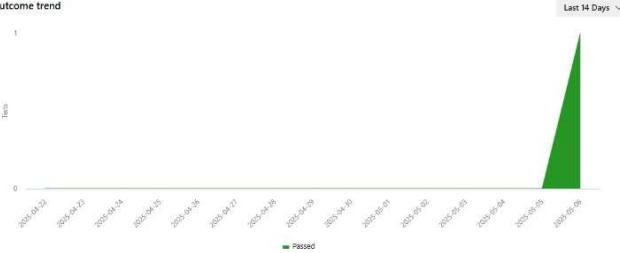
1 Test plans 1 Test points

1 (1 / 1) Test points run 100% Run

100% (1 / 1) Pass rate 1 Passed

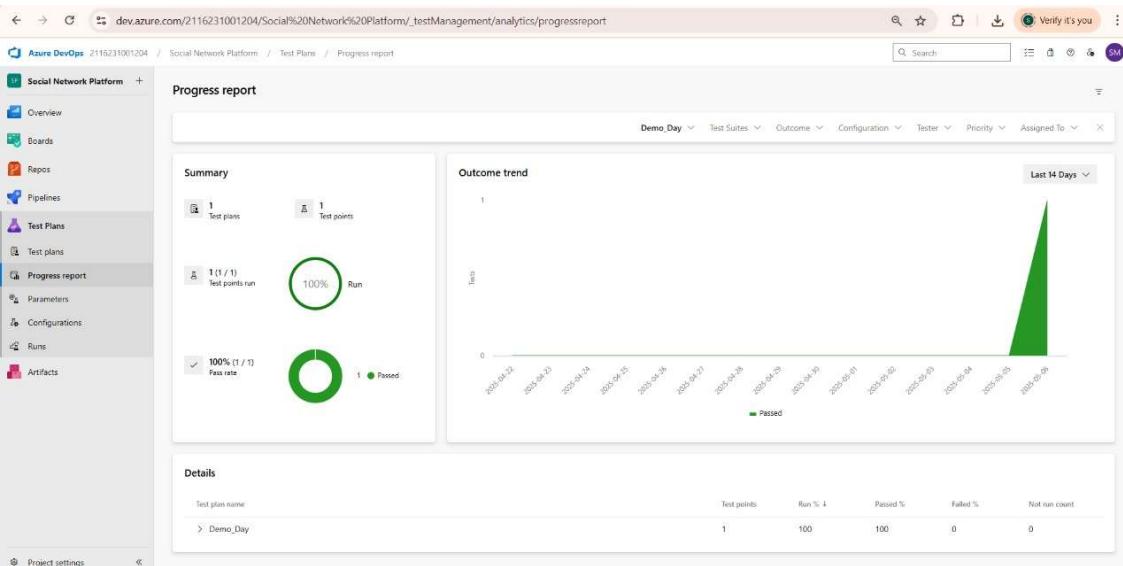
Outcome trend

Last 14 Days



Test plan name	Test points	Run %	Passed %	Failed %	Not run count
> Demo_Day	1	100	100	0	0

Project settings



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

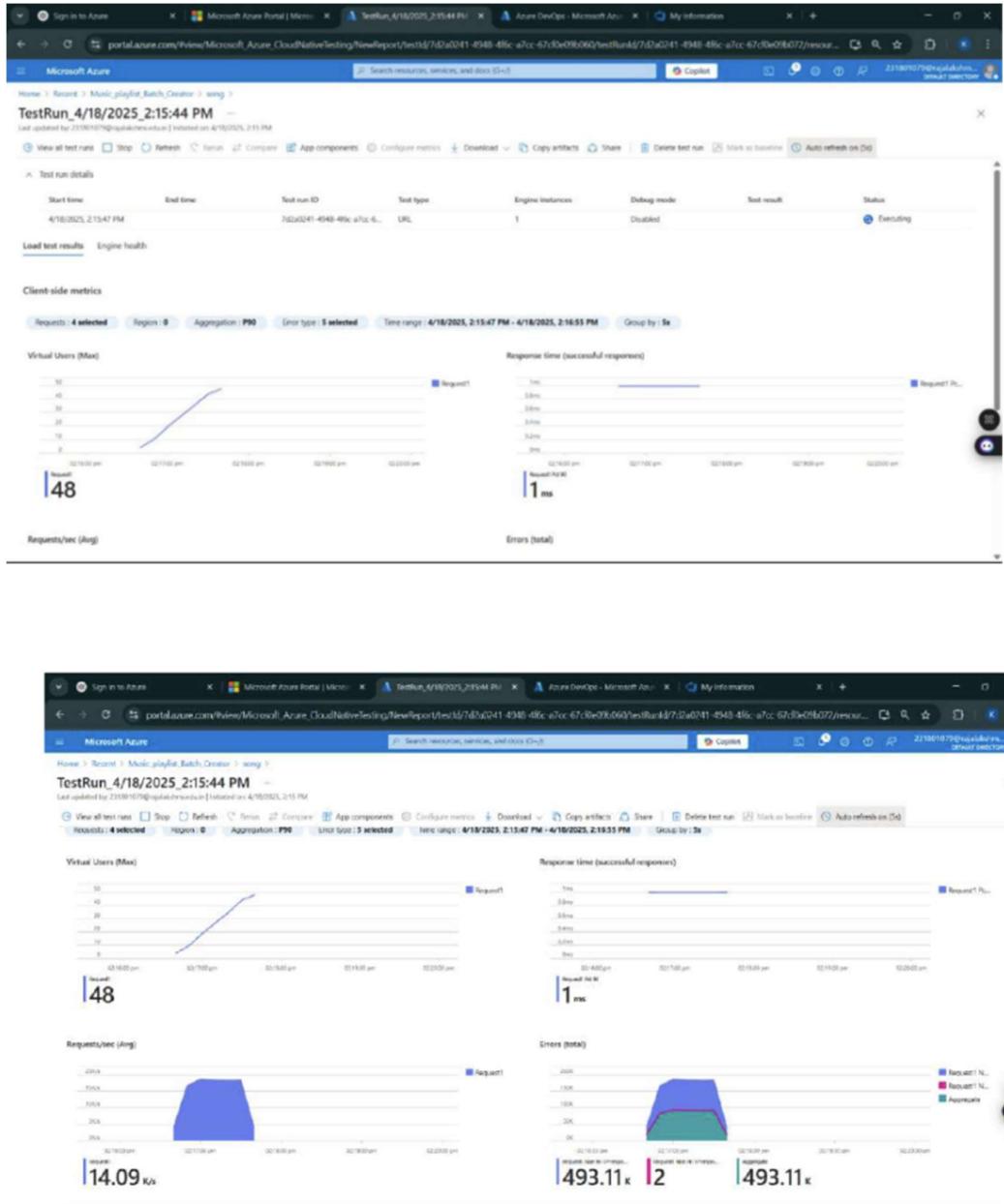
- Test Name: Provide a unique name.
- Description: (Optional) Add test purpose.
- Run After Creation: Keep checked.

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.