

INDEX

S.NO	DATE	TITLE
1.	21/1/25	Azure DevOps Environment Setup
2.	21/1/25	Azure DevOps Project Setup and User Story Management
3.	28/1/25	Setting Up Epics, Features and User Stories for Project Planning
4.	11/2/25	Sprint Planning
5.	18/2/25	Poker Estimation
6.	25/2/25	Designing Class and Sequence Diagrams for Project Architecture
7.	04/3/25	Designing Use-Case and Activity Diagrams for Project Architecture
8.	25/3/25	Testing – Test Plans and Test Cases
9.	15/4/25	CI/CD Pipelines in Azure
10.	22/4/25	GitHub: Project Structure & Naming Convention

EXP NO: 1

AZURE DEVOPS ENVIRONMENT SETUP

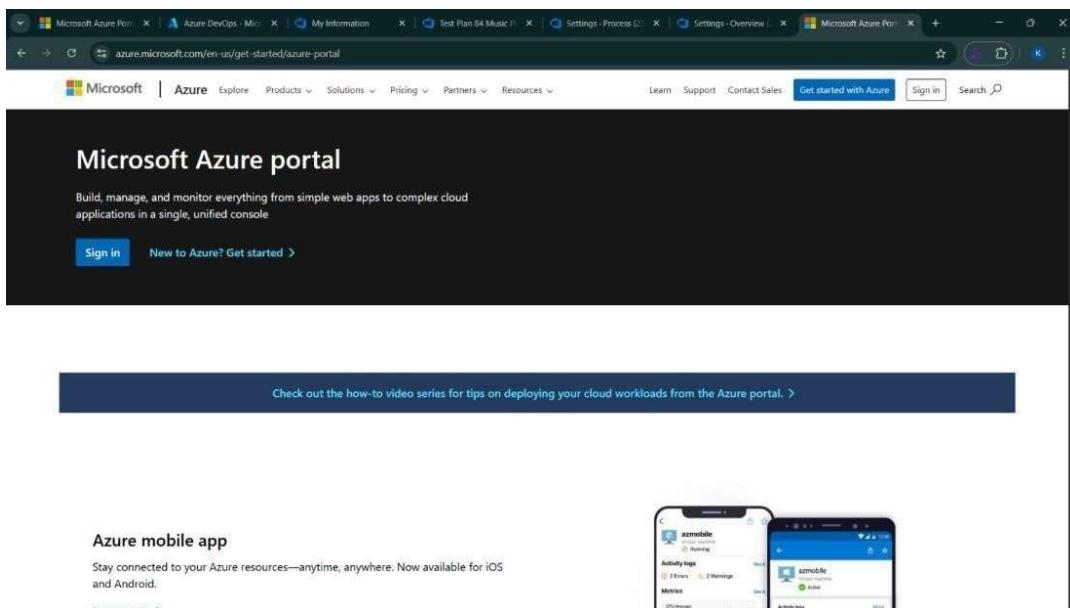
AIM

To set up and access the Azure DevOps environment by creating an organization through the Azure portal.

INSTALLATION

1. Open your web browser and go to the Azure website: <https://azure.microsoft.com/en-us/getstarted/azureportal>. Sign in using your Microsoft account credentials.

If you don't have a Microsoft account, you can create one here: <https://signup.live.com/?lic=1>



2. Azure home page

The screenshot shows the Microsoft Azure home page. At the top, there's a search bar and a Copilot button. Below the header, the "Azure services" section features a "Create a resource" button and icons for various services: Azure DevOps organizations, Quickstart Center, Azure AI foundry, Kubernetes services, Virtual machines, App Services, Storage accounts, and SQL databases. A "More services" link is also present. The "Resources" section displays a table with columns for Name, Type, and Last Viewed, showing a single entry: "No resources have been viewed recently". Below this is a "View all resources" button. The "Navigate" section includes links for Subscriptions, Resource groups, All resources, and Dashboard.

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

The screenshot shows the Microsoft Azure search results for "azure devops". The search bar at the top contains the query. The results are categorized under "All", "Services (99+)", and "Marketplace (21)". The "Services" section lists "Azure DevOps organizations", "Azure Cosmos DB", "Azure Database for MySQL servers", and "Azure Deployment Environments". The "Marketplace" section lists "Build Agents for Azure DevOps", "Azure DevOps Auditing", "Azure DevOps Backup Tool", and "Self Hosted Runner for Azure DevOps". The "Documentation" section lists "Billing overview - Azure DevOps". At the bottom, there's a "Continue searching in Microsoft Entra ID" link and a "Give feedback" button. The interface is identical to the Azure home page shown in the previous screenshot.

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



RESULT

Successfully accessed the Azure DevOps environment and created a new organization through the Azure portal.

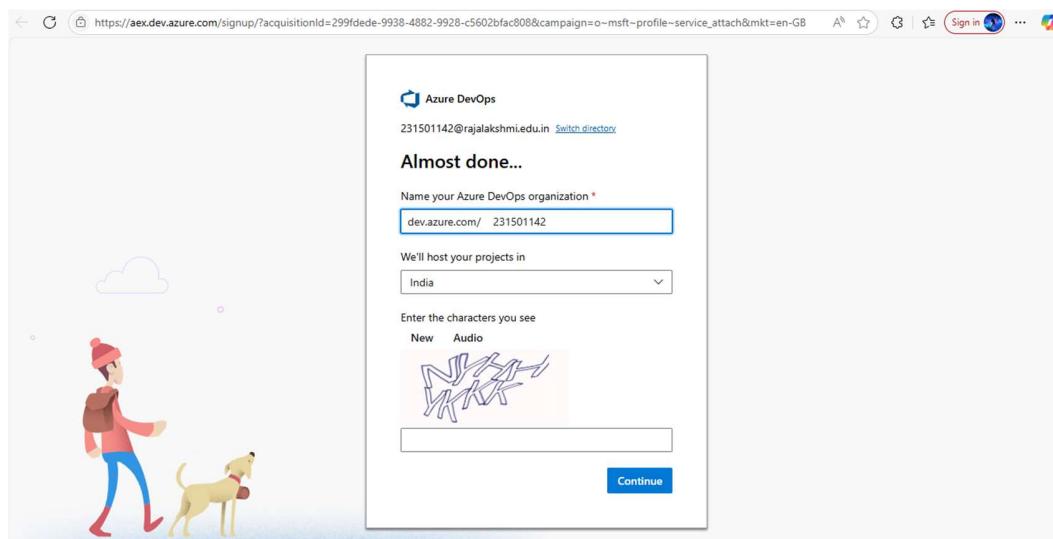
EXP NO: 2

AZURE DEVOPS PROJECT SETUP AND USER STORY MANAGEMENT

AIM

To set up an Azure DevOps project for efficient collaboration and agile work management.

1.Create An Azure Account



2.Create the First Project in Your Organization

- a. 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.
- b. On the organization's **Home page**, click on the **New Project** button.
- c. 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).

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

Create new project

Project name *

Blog management system 2

Description

Visibility



Public

Anyone on the internet can view the project. Certain features like TFVC are not supported.



Private

Only people you give access to will be able to view this project.

Public projects are disabled for your organization. You can turn on public visibility with [organization policies](#).

Advanced

Version control

Git

Work item process

Agile

Cancel

Create

3. 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.

The screenshot shows the Azure DevOps Organizations dashboard. On the left, there's a profile card for 'Sai Sanjay S V' with a large green circular icon containing 'SV'. Below the icon, it says 'Sai Sanjay S V' and '231501142@rajalakshmi.edu.in'. There are dropdown menus for 'Microsoft account' and 'India'. A section for 'Visual Studio Dev Essentials' is present. On the right, under 'Azure DevOps Organizations', there's a list of projects: 'dev.azure.com/231501142' (Owner), which has a sub-item 'Blog Management System'; and 'dev.azure.com/231501142-AIML' (Owner) and 'dev.azure.com/231501142-aiml-fb' (Owner). A 'Create new organization' button is at the top right.

4. Project dashboard

The screenshot shows the Azure DevOps project dashboard for 'Blog Management System'. The left sidebar includes options like Overview, Summary, Dashboards, Wiki, Boards, Repos, Pipelines, Test Plans, Artifacts, and Project settings. The main area displays the project name 'Blog Management System' and sections for 'About this project' (describing it as a software platform for blog management) and 'Project stats' (showing 0 work items created, 0 work items completed, 0 pull requests opened, and 0 commits by 0 authors). A search bar and a 'Private' button are also visible.

5. 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.

The screenshot shows the Azure DevOps interface for a project named "Blog Management System". The left sidebar is open, showing options like Overview, Boards, Work items, Boards, Backlogs, Sprints, Queries, Delivery Plans, Analytics views, Repos, Pipelines, and Project settings. The "Backlogs" option is selected. The main area displays the backlog for the "Blog Management System Team". The backlog is organized into two columns: "Backlog" and "Analytics". The "Backlog" column lists work items categorized by type: Epic, Feature, Feature, Epic, Feature, Epic, Feature, Feature. The "Analytics" column provides details such as State (New), Effort, Business Value, and Priority. A message at the top right encourages users to try the new look and features.

Work Item Type	Title	State	Effort	Business Value
Epic	User Management and Authentication	New		Bus
Feature	Role-Based Access Control	New		Bus
Feature	User Registration and Login	New		Bus
Epic	Blog Post Management	New		Bus
Feature	Create and Edit Blog Posts	New		Bus
Epic	Reader Interaction and Moderation	New		Bus
Feature	Commenting System	New		Bus
Feature	Content Moderation	New		Bus

RESULT

Successfully created an Azure DevOps project with user story management and agile workflow setup.

EXP NO:3

SETTING UP EPICS, FEATURES, AND USER STORIES FOR PROJECT PLANNING

AIM

To learn about how to create epics, user story, features, backlogs for your assigned project.

Create Epic, Features, User Stories, Task

The screenshot shows the Azure DevOps Backlog view for the 'Blog Management System' team. The backlog is organized into columns for Work Item Type, Title, State, Effort, and Business Value. The backlog includes the following items:

Work Item Type	Title	State	Effort	Business Value
Epic	User Management and Authentication	New	B	B
Feature	Role-Based Access Control	New	B	B
User Story	As an admin, I want to assign roles to users so that I can control their access...	New	B	B
Feature	User Registration and Login	New	B	B
User Story	As a new user, I want to sign up and log in securely so that I can access and...	New	B	B
Task	Implement frontend forms and backend API for sign-up and login using ...	New	B	B
Epic	Blog Post Management	New	B	B
Feature	Create and Edit Blog Posts	New	B	B
User Story	As a blogger, I want to create and edit my blog posts so that I can manage ...	New	B	B
Epic	Reader Interaction and Moderation	New	B	B
Feature	Commenting System	New	B	B

1. Fill in Epics

The screenshot shows the 'Edit work item' dialog for an Epic titled 'User Management and Authentication'. The dialog is divided into several sections:

- General:** Status is set to 'New', Area is 'Blog Management System', Reason is 'New', Iteration is 'Blog Management System'. The 'Details' tab is selected.
- Description:** A placeholder text 'Click to add Description.' is present.
- Planning:** Priority is '2', Risk is 'Low', Effort is 'Medium'.
- Deployment:** A note about tracking releases is shown, along with a link to 'Learn more about deployment status reporting'.
- Development:** A 'Add link' section with instructions to link to a commit, pull request, or branch.
- Related Work:** A section for linking to other work items.
- Discussion:** A comment input field with placeholder text 'Add a comment. Use # to link a work item, @ to mention a person, or ! to link a pull request.' and a 'switch to Markdown editor' link.

2.Fill in Features

FEATURE 13

13 Role-Based Access Control

No one selected 0 Comments Add Tag

State: New Area: Blog Management System
Reason: New Iteration: Blog Management System\Sprint 1

Description

Click to add Description.

Planning

Priority: 2
Risk
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.

Related Work

3.Fill in User Story Details

USER STORY 14

14 As an admin, I want to assign roles to users so that I can control their access to different parts of the system

No one selected 0 Comments Add Tag

State: New Area: Blog Management System
Reason: New Iteration: Blog Management System\Sprint 1

Description

Click to add Description.

Acceptance Criteria

Click to add Acceptance Criteria.

Discussion

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

Classification

Value area: Business
Risk

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](#)

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

RESULT

Thus, the creation of epics, features, user story and task has been created successfully.

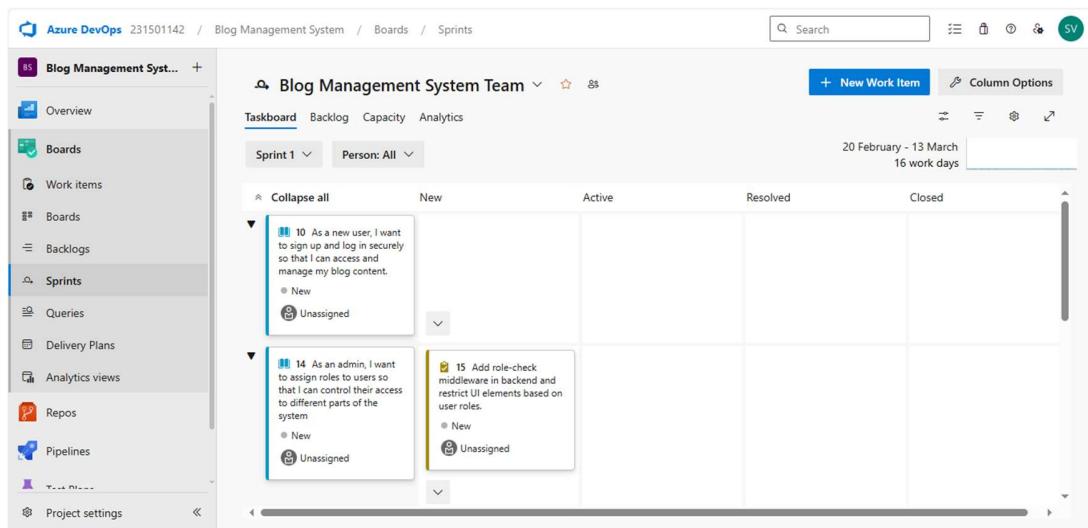
EXP NO:
4

SPRINT PLANNING

AIM

To assign user story to specific sprint for the Blog Management System.

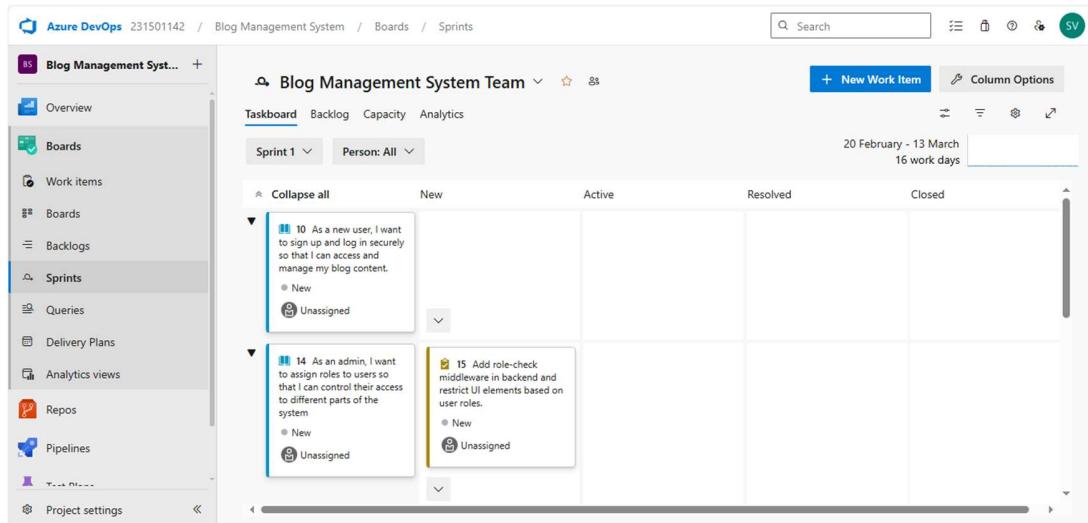
Sprint Planning Sprint 1



The screenshot shows the Azure DevOps Taskboard for the 'Blog Management System Team' under 'Sprint 1'. The board has columns for New, Active, Resolved, and Closed. Three user stories are visible:

- User Story 10: As a new user, I want to sign up and log in securely so that I can access and manage my blog content. Status: New, Unassigned.
- User Story 14: As an admin, I want to assign roles to users so that I can control their access to different parts of the system. Status: New, Unassigned.
- User Story 15: Add role-check middleware in backend and restrict UI elements based on user roles. Status: New, Unassigned.

Sprint 2



The screenshot shows the Azure DevOps Taskboard for the 'Blog Management System Team' under 'Sprint 1'. The board has columns for New, Active, Resolved, and Closed. The same three user stories are present, but their status has changed:

- User Story 10: As a new user, I want to sign up and log in securely so that I can access and manage my blog content. Status: Resolved, Closed.
- User Story 14: As an admin, I want to assign roles to users so that I can control their access to different parts of the system. Status: New, Unassigned.
- User Story 15: Add role-check middleware in backend and restrict UI elements based on user roles. Status: New, Unassigned.

Sprint 3

The screenshot shows the Azure DevOps Taskboard for the 'Blog Management System Team'. The board is set to 'Sprint 1' and 'Person: All'. It displays three backlog items:

- 10**: As a new user, I want to sign up and log in securely so that I can access and manage my blog content.
Status: New
Assignee: Unassigned
- 14**: As an admin, I want to assign roles to users so that I can control their access to different parts of the system.
Status: New
Assignee: Unassigned
- 15**: Add role-check middleware in backend and restrict UI elements based on user roles.
Status: New
Assignee: Unassigned

The board also includes columns for 'New', 'Active', 'Resolved', and 'Closed'.

RESULT

The Sprints have been created for Blog Management System.

EXP NO:5

POKER ESTIMATION

AIM

Create Poker Estimation for the user stories –Blog Management System.

Poker Estimation

The screenshot shows the 'USER STORY 11' creation screen. At the top, there is a summary: '11 As a new user, I want to sign up and log in securely so that I can access and manage my blog content.' Below this, there are several fields:

- State:** New
- Reason:** New
- Area:** Blog Management System
- Iteration:** Blog Management System

On the right side, there are buttons for 'Save and Close', 'Follow', and other options. A note indicates the item was updated by Sai Sanjay S V on 16 Apr.

The main area is divided into sections:

- Description:** Click to add Description.
- Acceptance Criteria:** Click to add Acceptance Criteria.
- Discussion:** A comment input field with placeholder text: 'Add a comment. Use # to link a work item, @ to mention a person, or ! to link a pull request.'
- Planning:** Story Points, Priority (set to 2), and Risk.
- Classification:** Value area (Business).
- Deployment:** A note about tracking releases associated with this work item.
- Development:** Add link, with instructions to link an Azure Repos commit, pull request, or branch.
- Related Work:** A dropdown menu.

RESULT

The Estimation/Story Points is created for the project using Poker Estimation.

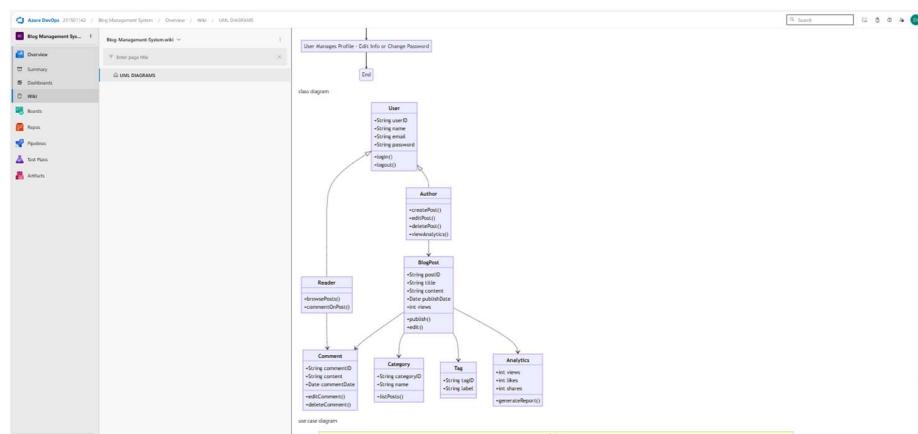
EXP NO: 6

DESIGNING CLASS AND SEQUENCE DIAGRAMS FOR PROJECT ARCHITECTURE

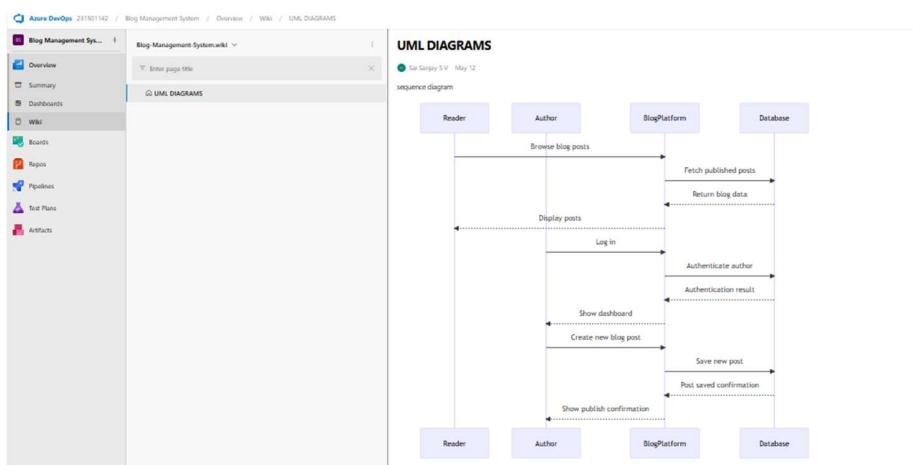
AIM

To Design a Class Diagram and Sequence Diagram for the given Project.

6A. Class Diagram



6B. Sequence Diagram



RESULT

The Class Diagram and Sequence Diagram is designed Successfully for Blog Management System.

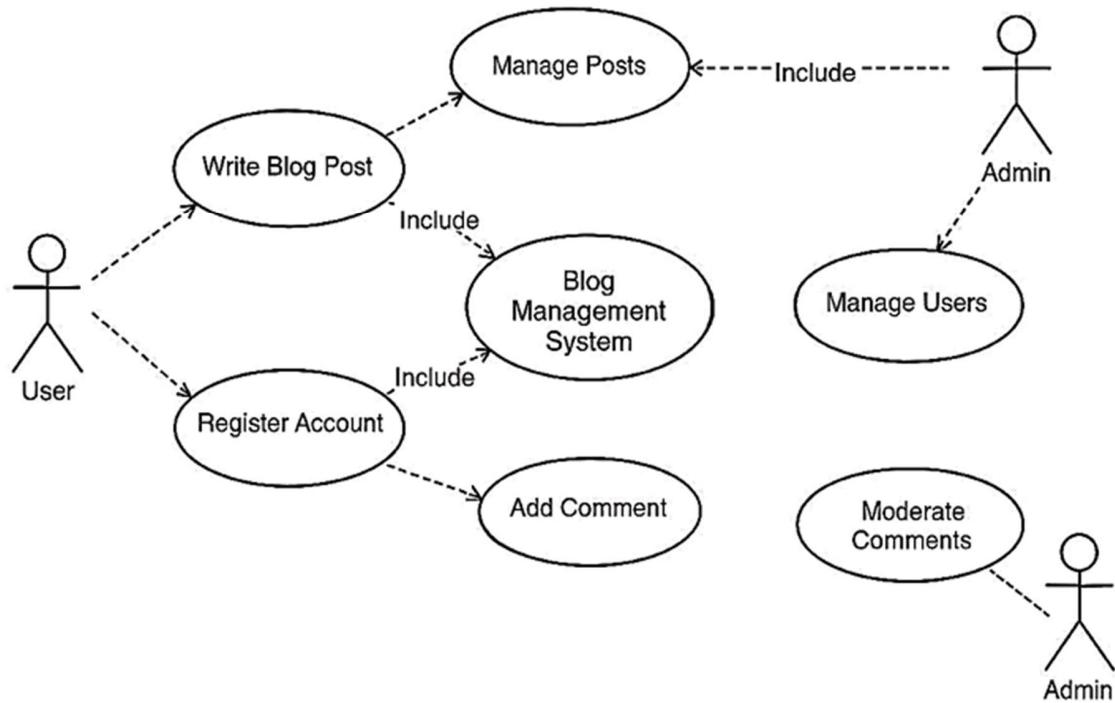
EXP NO: 7

DESIGNING USE-CASE AND ACTIVITY DIAGRAMS FOR PROJECT STRUCTURE

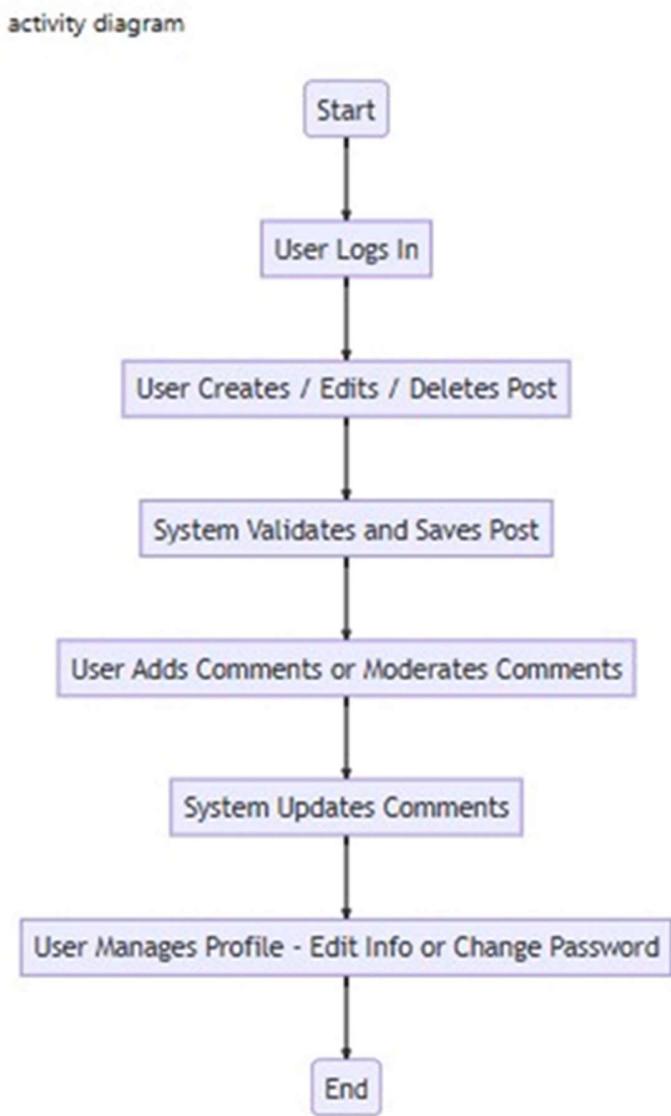
AIM

To Design an Use-Case Diagram and Activity Diagram for the given Project.

7A. Use-Case Diagram



7B. Activity Diagram



RESULT

The Use-Case Diagram and Activity Diagram is designed Successfully for the Blog Management System.

EXP NO: 8

TESTING TEST PLANS AND TEST CASES

AIM

Test Plans and Test Case and write two test cases for at least five user stories showcasing the happy path and error scenarios in azure DevOps platform.

Test Planning and Test Case Design Procedure

1. Understand Core Features of the Application

- User Login

2. Define User Interactions

- Each test case simulates a real user behaviour (e.g., logging in, submitting quizzes, viewing results)

3. Design Happy Path Test Cases

- Focused on validating that all core functionalities work correctly under normal conditions
- Example: Player registers and logs in, submits quizzes and views results

4. Design Error Path Test Cases

- Simulate invalid inputs, system issues or failed actions to ensure proper error handling.
- Example: Login with invalid credentials, submission without attachments, unauthorized access to admin panel.

5. Break Down Steps and Expected Results

- Each test case includes a clear sequence of actions and expected results. • Ensures both manual testers and automation tools can follow the process easily.

6. Use Clear Naming and IDs

- Test cases are uniquely identifies (e.g., TC01 – Valid Login, TC03 – Invalid Password). • Facilities easy mapping to features and tracking in Azure DevOps.

7. Separate Test

- Grouped by functionality such as:
 - Login and Registration
 - Quizzes Submission
 - Viewing

Results

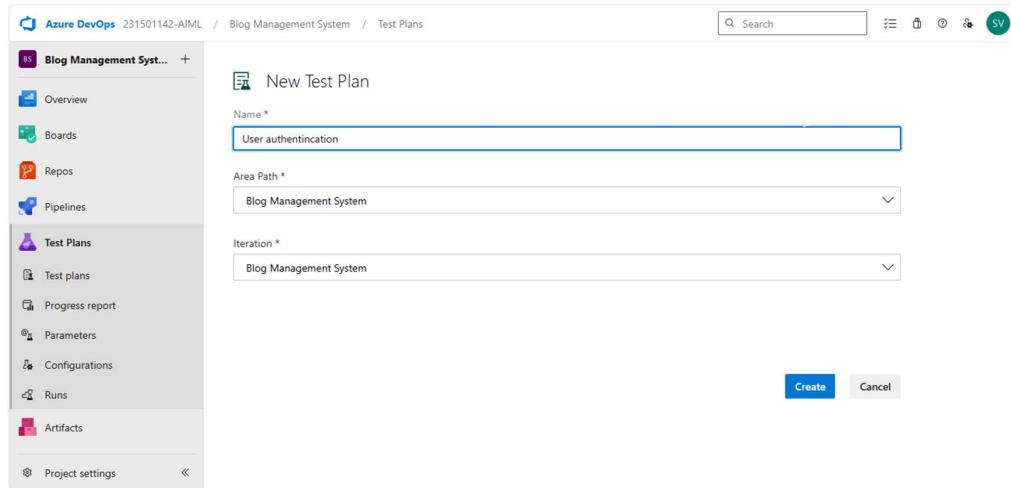
- Admin Functions

 • Improves organization and enables focused execution in Azure DevOps.

8. Prioritize and Review

- High-priority assigned to critical workflows like login, quizzes and results.
- Reviewed for completeness, accuracy and alignment with user stories and features definition.

1. New test plan



New Test Plan

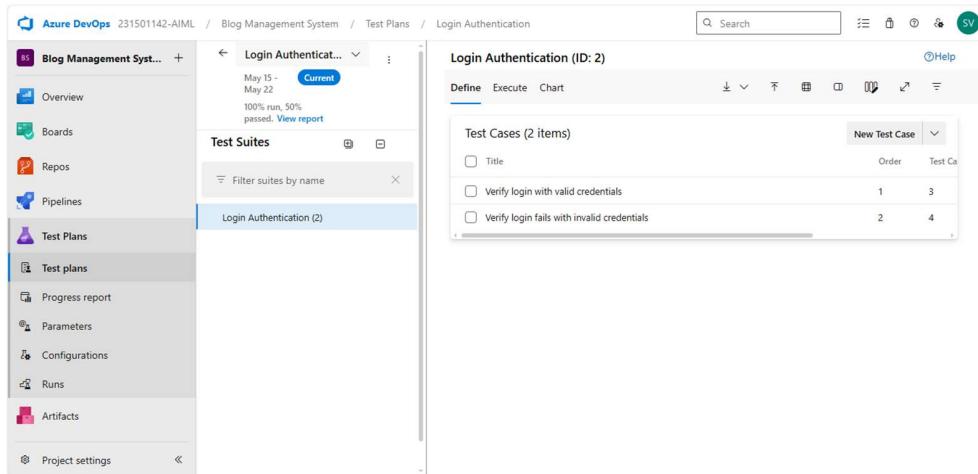
Name *
User authentication

Area Path *
Blog Management System

Iteration *
Blog Management System

Create Cancel

2. Test suite



Login Authentication (ID: 2)

Title	Order	Test Ca
Verify login with valid credentials	1	3
Verify login fails with invalid credentials	2	4

3.Test case

Give two test cases for at least five user stories showcasing the happy path and error scenarios in azure DevOps platform.

Blog Management System– Test Plans

USER STORIES

- As an admin, I want to log in to manage blog posts and users (ID: 3).
- As a writer, I want to register and log in so that I can create and manage my blog posts securely (ID: 4).
- As an admin, I want to assign roles and permissions to users so they can access the appropriate features (ID: 6).
- As a writer, I want to see only my posts and related analytics to manage my content efficiently (ID: 7).
- As an admin, I want to create and schedule blog posts with publication dates to control the content flow (ID: 10).

TEST SUITES

Test Suite: TS01 - Login Verification (ID: 23)

1. TC01 – Valid Login

Action:

- Go to the Sign-Up page.
- Enter valid name, email, and password.
- Click “Login” button.

Expected Results:

- Login form is displayed.
- Fields accept valid credentials without error.
- Account is successfully logged in and redirected to the dashboard.

Type: Happy Path

2. TC02 – Invalid Username

Action:

- Go to the Login page.
- Enter invalid or unregistered username.
- Click on "Login" button.

Expected Results:

- Login form is displayed.
- System displays an error message like “Invalid username”.
- User is not logged in and remains on the login page.

Type: Negative Path

3. TC03 – Invalid Password

Action:

- Go to the Login page.
- Enter a valid username or email.
- Enter an incorrect or invalid password.
- Click on "Login" button.

Expected Results:

- Login form is displayed.
- System shows an error message like "Invalid username or password."
- User is not logged in and stays on the login page.

Type: Error Path

4. TC04 – Login with Wrong Password

Action:

- Go to the Login page.
- Enter valid email and incorrect password.
- Click on "Login".

Expected Results:

- Input is accepted.
- Error message "Invalid username or password" is shown.

Type: Error Path

Test Suite: TS02 – Blog Post Creation (ID: 28)

1. TC05 – Valid Blog Post Creation

Action:

- Navigate to the "Create Post" page.
- Enter valid blog title, content, and metadata.
- Click the "Publish" or "Save Draft" button.

Expected Results:

- Blog post is successfully saved in the database.

Type: Happy Path

2. TC06 – Missing Title

Action:

- Navigate to "Create Post" page.

- Leave the title field empty and fill in the content.
- Click the “Publish” or “Save Draft” button.

Expected Results:

- System displays a validation error message like “Title is required”.
- Post is not submitted or saved in the database.

Type: Error Path

Test Suite: TS03 – Blog Analytics Viewing (ID: 32)

1. TC07 – View Analytics as Writer

Action:

- Log in as a writer.
- Navigate to “My Analytics” section.
- Select a blog post to view analytics.

Expected Results:

- Writer sees views, likes, and comment count for the selected post.

Type: Happy Path

2. TC08 – View All Analytics as Admin

Action:

- Log in as an admin.
- Navigate to the “Analytics” or “All Posts Data” section.
- Select a post to view its performance.

Expected Results:

- Admin sees a list of posts with detailed analytics including author, views, likes, and comments.

Type: Happy Path

Test Suite: TS04 – Blog Publishing and Scheduling (ID: 36)

1. TC09 – Schedule Blog Post with All Fields Completed

Action:

- Log in and navigate to the "Create Post" page.
- Fill in title, content, and metadata.
- Set a future publication date.
- Click "Schedule".

Expected Results:

- Post is saved and scheduled to publish at the selected time.

Type: Happy Path

2. TC10 – Attempt to Schedule Without Content

Action:

- Log in and go to "Create Post" page.
- Enter title but leave content empty.
- Click "Schedule".

Expected Results:

- System displays a warning or prompt (e.g., “Content cannot be empty”).

Type: Error Path

Test Cases

The screenshot shows the Azure DevOps interface for the 'Blog Management System' project. The left sidebar includes 'Overview', 'Boards', 'Repos', 'Pipelines', 'Test Plans' (selected), 'Test plans', 'Progress report', 'Parameters', 'Configurations', 'Runs', 'Artifacts', and 'Project settings'. The main area is titled 'Test Plans' with a 'Mine' filter. It lists six test plans: 'Prevent Too Many Comments', 'Delete Blog by Admin', 'Features To be Tested', 'Blog Comment System', 'Blog Post Management', and 'Login Authentication'. Each entry includes a title, Test Plan ID, State (Active), Area Path (Blog Management Syste...), Iteration (Blog Management System), and Assigned To (Rohith S, SARVEESH B, 231501133, Rohith S, Sai Sanjay S V, Sai Sanjay S V).

The screenshot shows the 'Login Authentication' test plan details in Azure DevOps. The left sidebar is identical to the previous screenshot. The main area shows the 'Login Authentication (ID: 2)' test plan. It has three tabs: 'Define' (selected), 'Execute', and 'Chart'. Under 'Test Cases (2 items)', there are two entries: 'Verify login with valid credentials' (checked) and 'Verify login fails with invalid credentials'. The 'Define' tab also shows a summary: 'May 15 - May 22', '100% run, 50% passed', and a link to 'View report'. The 'Execute' tab shows a progress bar indicating 50% completion.

The screenshot shows the Azure DevOps Test Plans interface. On the left, the navigation pane is visible with options like Overview, Boards, Repos, Pipelines, Test Plans, Test plans, Progress report, Parameters, Configurations, Runs, Artifacts, and Project settings. The 'Test plans' section is currently selected. In the center, a test plan titled 'Blog Post Management (ID: 6)' is displayed. The plan is set for the date range May 15 - May 22 and has a status of 100% run, 100% passed. A link to 'View report' is present. Below this, the 'Test Suites' section shows a single suite named 'Blog Post Management (1)'. Under 'Test Cases (1 item)', there is one entry: 'Verify successful creation of a new blog post'.

This screenshot shows the details of a specific test case, 'TEST CASE 7'. The title is '7 Verify successful creation of a new blog post'. The test case is created by 'Sai Sanjay S V' and has 0 comments and 0 tags. It is in the 'Design' state and assigned to the 'Blog Management System' area and iteration. The 'Steps' tab is active, displaying the following steps:

Step	Action	Expected result
1.	Navigate to dashboard	
2.	Click on "Create Blog Post"	
3.	Enter the blog title, content, category, and tags	
4.	Click the "Publish" button	Success message displayed: "Blog post published successfully."

A note at the bottom of the steps section says 'Click or type here to add a step'. The 'Deployment' section contains a note about tracking releases. The 'Development' section includes an 'Add link' button and a note about linking to Azure Repos. The 'Related Work' section also has an 'Add link' button.

The screenshot shows the Azure DevOps interface for a project named "Blog Management Syst...". The left sidebar is open, showing the "Test plans" section selected. In the center, a "Test Suites" view is displayed for the "Blog Comment System". A single suite named "Blog Comment System (1)" is listed. On the right, the details for this suite are shown, including a summary table with one test case:

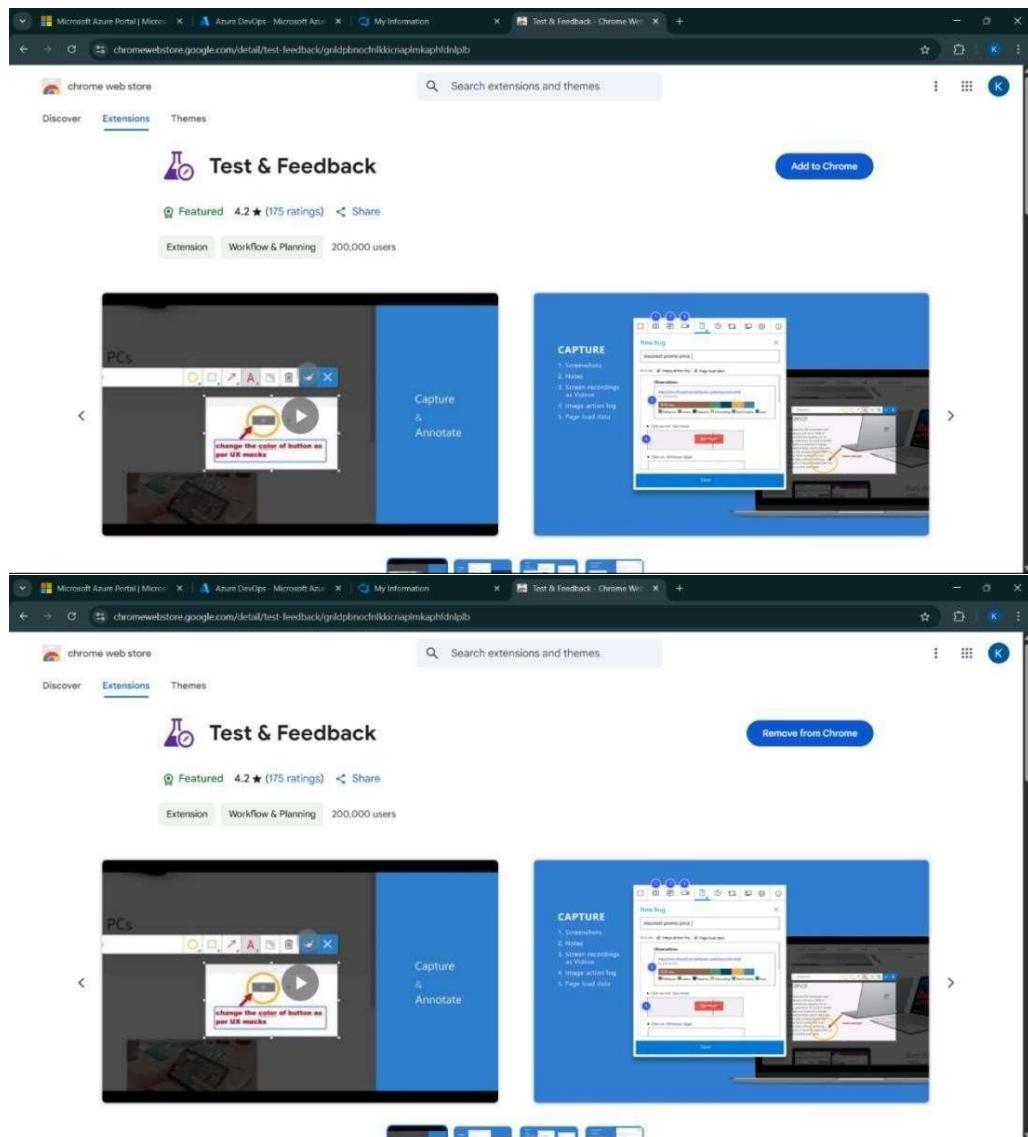
Title	Order	Test Ca
Verify that a logged-in user can add a comment to a blog post	1	10

The screenshot shows a detailed view of a test case titled "TEST CASE 10". The test case description is: "10 Verify that a logged-in user can add a comment to a blog post". The test case is assigned to "SARVEESH B". The "Steps" tab is active, showing the following steps:

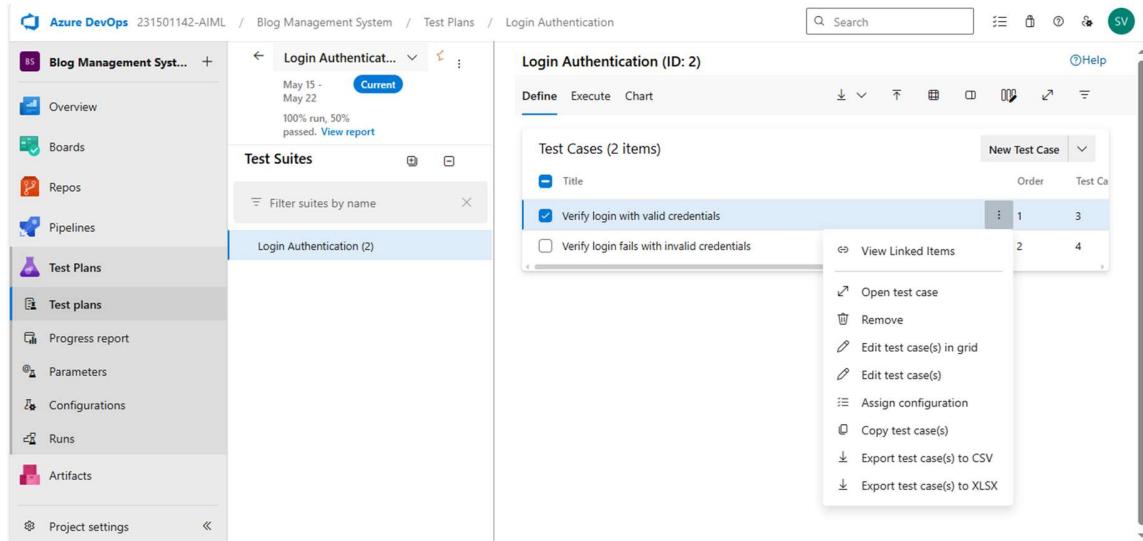
- Log in as a registered user
- Navigate to the specific blog post
- Scroll to the comment section
- Enter the comment text
- Click the post comment button

The "Expected result" for step 5 is: "Comment is successfully posted and displayed below the blog post". The "Deployment" section contains a note about tracking releases. The "Development" section has a link to Azure Repos. The "Related Work" section has a link to add existing work items.

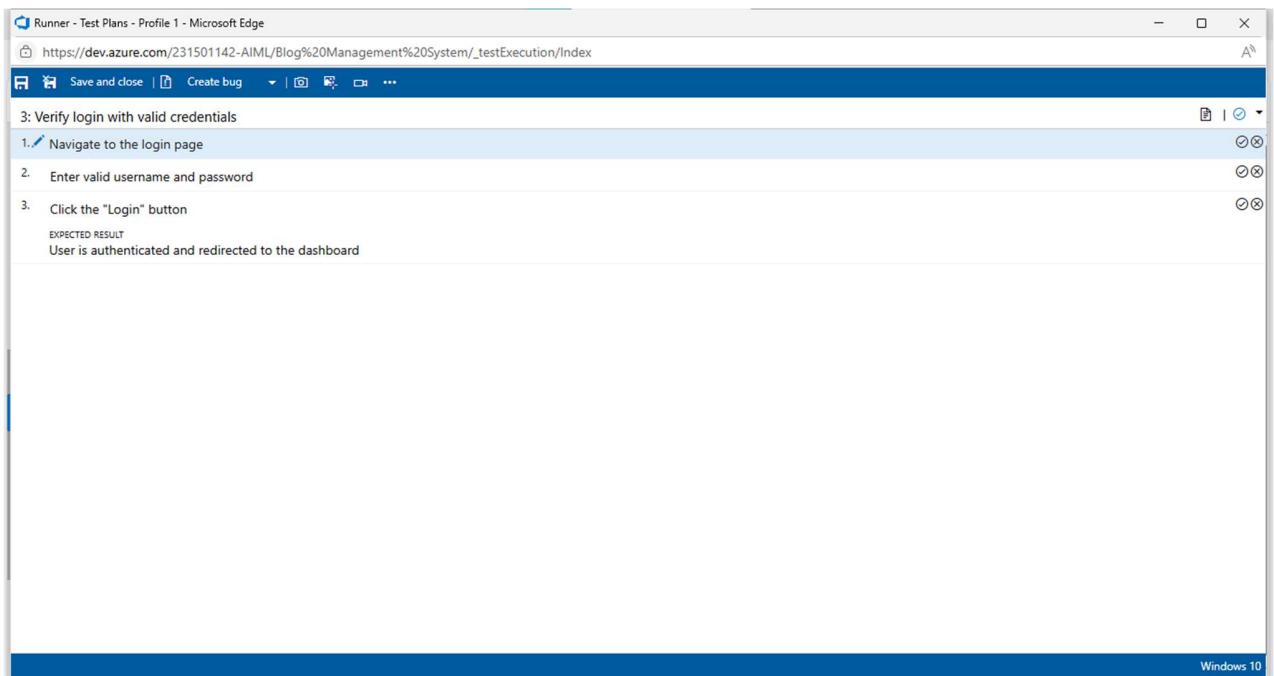
4. Installation of test



5. Running the test cases



The screenshot shows the Azure DevOps Test Plans interface. On the left, the navigation bar includes 'Overview', 'Boards', 'Repos', 'Pipelines', 'Test Plans', 'Test plans', 'Progress report', 'Parameters', 'Configurations', 'Runs', 'Artifacts', and 'Project settings'. The 'Test Plans' section is currently selected. In the center, a 'Test Suites' list shows 'Login Authentication (2)'. The 'Test Cases' section lists 'Verify login with valid credentials' (selected) and 'Verify login fails with invalid credentials'. A context menu is open over the selected test case, containing options: 'New Test Case', 'Order', 'Test Ca', 'View Linked Items', '1 3', '2 4', 'Open test case', 'Remove', 'Edit test case(s) in grid', 'Edit test case(s)', 'Assign configuration', 'Copy test case(s)', 'Export test case(s) to CSV', and 'Export test case(s) to XLSX'. The top right corner has a 'Help' link.



Runner - Test Plans - Profile 1 - Microsoft Edge
https://dev.azure.com/231501142-AIML/Blog%20Management%20System/_testExecution/Index

Save and close | Create bug | ...

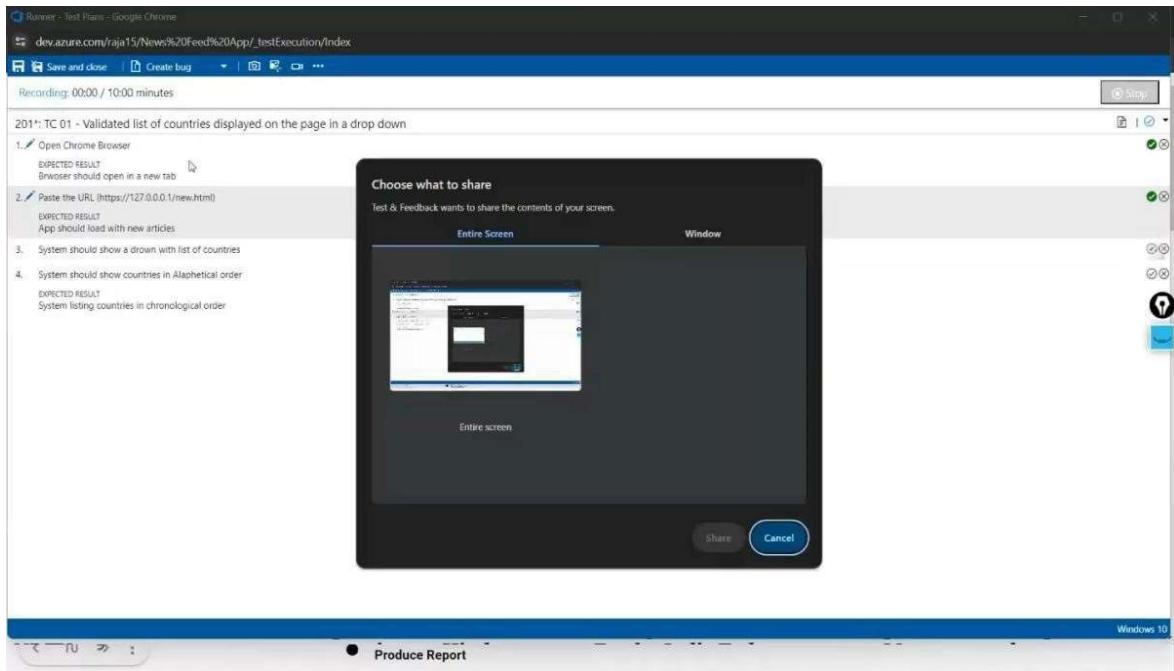
3: Verify login with valid credentials

1. Navigate to the login page
2. Enter valid username and password
3. Click the "Login" button

EXPECTED RESULT
User is authenticated and redirected to the dashboard

Windows 10

6.Recording the test case



7.Creating the bug

A screenshot of an Azure DevOps work item creation page for a "BUG" work item. The title is "user should not login with invalid details". The work item fields include:

- State: Unassigned
- Reason: New
- Area: Online Quiz System
- Iteration: Online Quiz System\sprint 1

The "Repro Steps" section contains the following steps:

- 5/17/2025 11:00 AM Bug filed on "Valid Login"
- Step no. Result Title
1. None Navigate to the login page.
2. None enter a valid username/email.
3. None enter valid password.
4. None Click the login button.
5. None

The "Planning" section includes:

- Resolved Reason: [Link](#)
- Story Points: [Link](#)
- Priority: 2
- Severity: 3 - Medium
- Activity: [Link](#)

The "Deployment" section includes:

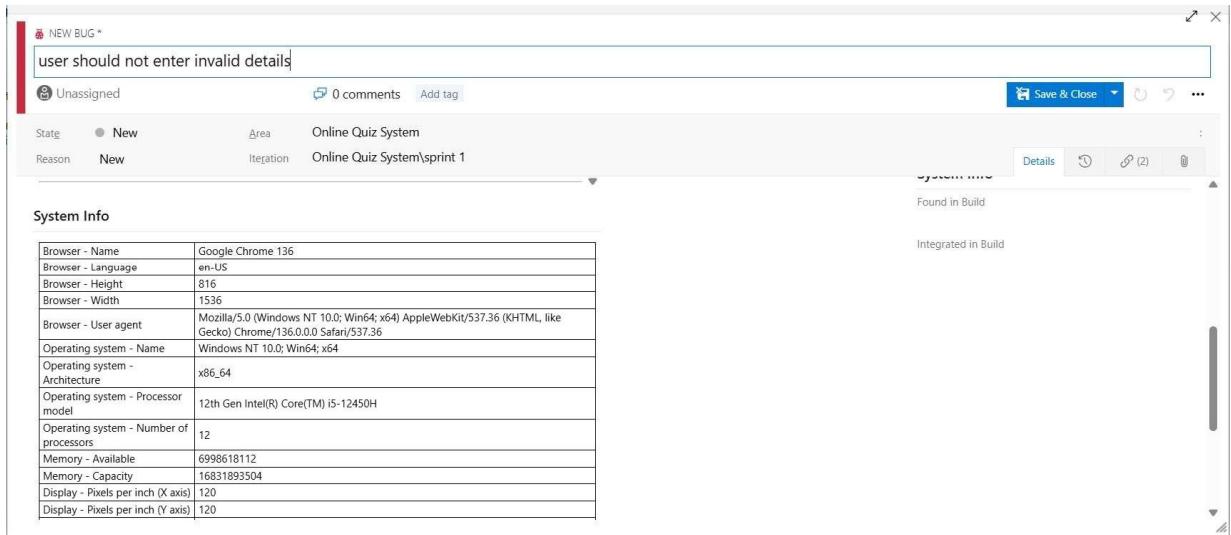
- To track releases associated with this work item, go to Releases and turn on deployment status reporting for Boards in your pipeline's Options menu. Learn more about deployment status reporting.

The "Development" section includes:

- + Add link: [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.

The "Related Work" section includes:

- + Add link: [Link](#)
- Add an existing work item as a parent: [Link](#)



8. Test case results

The screenshot shows the Azure DevOps Test Plans interface. On the left, the sidebar includes 'Overview', 'Boards', 'Repos', 'Pipelines', 'Test Plans' (selected), 'Test plans', 'Progress report', 'Parameters', 'Configurations', 'Runs', 'Artifacts', and 'Project settings'. The main area displays a 'Login Authentication' test plan under 'Blog Management Syst...'. It shows a timeline from May 15 to May 22, with 100% run and 50% passed, and a link to 'View report'. A 'Test Suites' section lists 'Login Authentication (2)'. The 'Test Case Results' section shows a single test point named 'Verify login with valid credentials' with an outcome of 'Passed'. The status bar at the bottom indicates 'Open execution history for current test point'.

9. Test report summary

The screenshot shows a 'NEW BUG' card in the Azure DevOps interface. The card details are as follows:

- Title:** User should not enter invalid details
- Status:** New
- Area:** Online Quiz System
- Reason:** N New
- Iteration:** Online Quiz System\Sprint 1
- Comments:** 0 comments
- Tags:** Add tag
- Repro Steps:** 5/17/2025 11:09 AM Bug filed on "Valid Login"
- Step no.**: 1. None Navigate to the login page.
- Step no.**: 2. None enter a valid username/email.
- Step no.**: 3. None enter valid password.
- Step no.**: 4. None Click the login button.
- Step no.**: 5. None

The card also includes sections for Planning (Resolved Reason, Story Points, Priority, Severity, Activity), Deployment (Release status reporting), Development (Add link), and Related Work (Add link).

- Assigning bug to the developer and changing state

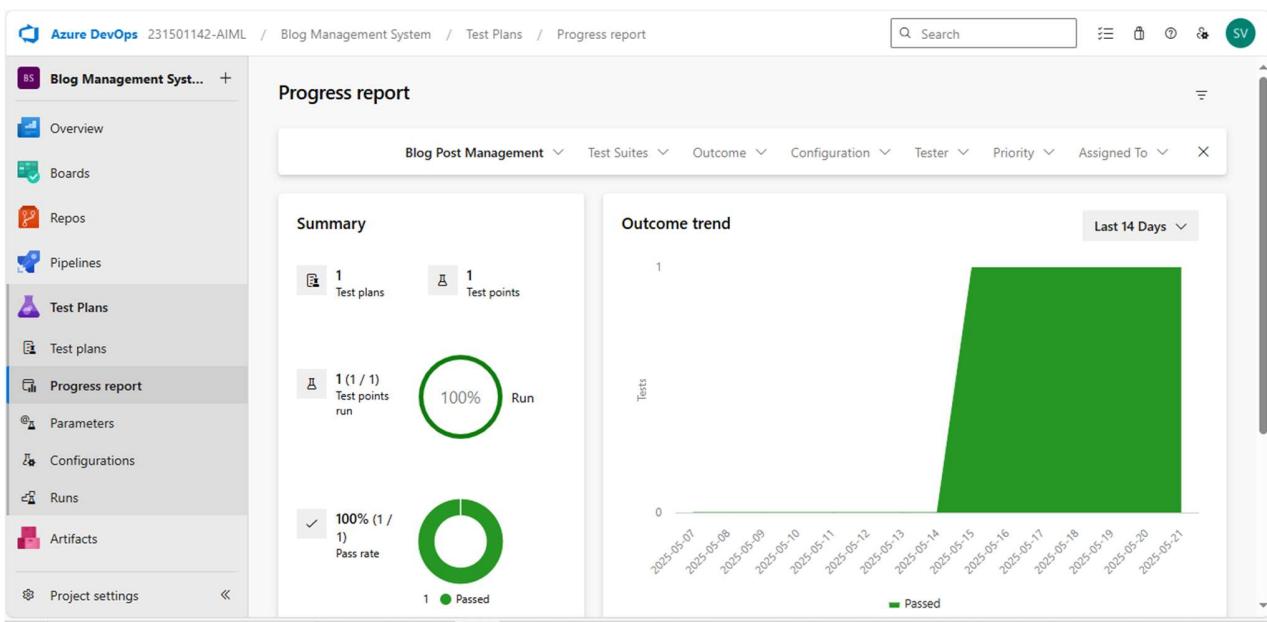
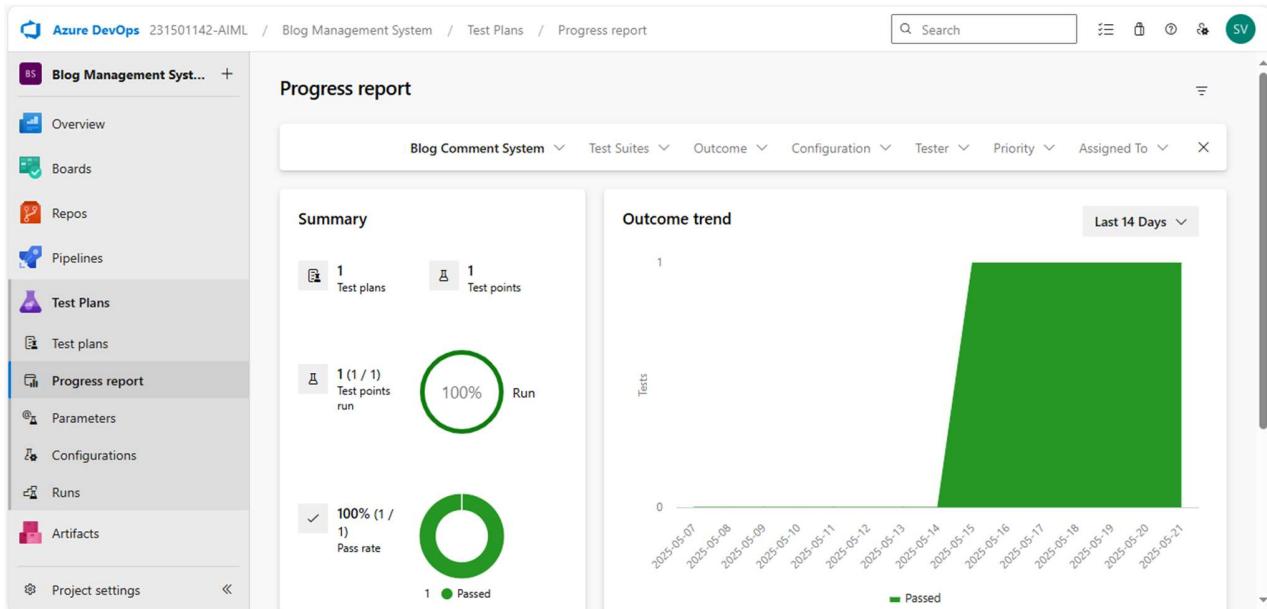
10. Progress report

The screenshot shows the 'Progress report' section for the 'Login Authentication' test plan. The left sidebar navigation includes:

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

The main area displays the following data:

- Summary:** 1 Test plans, 2 Test points, 2 (2 / 2) Test points run, 100% Run, 50% (1 / 2) Pass rate, 1 Passed.
- Outcome trend:** A chart showing the outcome trend from May 7 to May 21. The chart indicates 2 Failed and 1 Passed tests over the period.



11.Changing the test template

The screenshot shows the 'All processes' list in the Azure DevOps 'Process' settings. The list includes:

Name	Description	Team projects
Basic (default)	This template is flexible for any process and great for t...	0
Agile	This template is flexible and will work great for most te...	1
Scrum	This template is for teams who follow the Scrum frame...	0
CMMI	This template is for more formal projects requiring a fr...	0

The screenshot shows the 'Work item types' list under the 'Agile' process. The list includes:

Name	Description
Bug	Describes a divergence between required and actual behavior, and tracks the work done to correct the ...
Epic	Epics help teams effectively manage and groom their product backlog
Feature	Tracks a feature that will be released with the product
Issue	Tracks an obstacle to progress.
Task	Tracks work that needs to be done.
Test Case	Server-side data for a set of steps to be tested.
Test Plan	Tracks test activities for a specific milestone or release.
Test Suite	Tracks test activites for a specific feature, requirement, or user story.
User Story	Tracks an activity the user will be able to perform with the product

RESULT

The test plans and test cases for the user stories is created in Azure DevOps with Happy Path and Error Path.

EXP NO: 9

CI/CD PIPELINES IN AZURE

AIM

To implement a Continuous Integration and Continuous Deployment (CI/CD) pipeline in Azure DevOps for automating the build, testing, and deployment process of the Blog Management System, ensuring faster delivery and improved software quality.

PROCEDURE

Steps to Create and implement pipelines in Azure:

1. Sign in to Azure DevOps and Navigate to Your Project

Log in to dev.azure.com, select your organization, and open the project where your Student Management System code resides.

2. Connect a Code Repository (Azure Repos or GitHub)

Ensure your application code is stored in a Git-based repository such as Azure Repos or GitHub. This will be the source for triggering builds and deployments in your pipeline.

3. Create a New Pipeline

Go to the Pipelines section on the left panel and click “Create Pipeline”.

Choose your source (e.g., Azure Repos Git or GitHub), and then select the repository containing your project code.

4. Choose the Pipeline Configuration

You can select either the YAML-based pipeline (recommended for version control and automation) or the Classic Editor for a GUI-based setup. If using YAML, Azure DevOps will suggest a template or allow you to define your own.

5. Define Build Stage (CI - Continuous Integration) from YAML file.

6. Install dependencies (e.g., npm install, dotnet restore).

7. Build the application (dotnet build, npm run build).
 8. Run unit tests (dotnet test, npm test).
 9. Publish build artifacts to be used in the release stage.
10. Save and Run the Pipeline for the First Time
Save the YAML or build definition and click “Run”.
Azure will fetch the latest code and execute the defined build and test stages.
11. Configure Continuous Deployment (CD)
Navigate to the Releases tab under Pipelines and click “New Release Pipeline”. Add an Artifact (from the build stage) and create a new Stage (e.g., Development, Production).
 12. Configure the CD stage with deployment tasks such as deploying to Azure App Service, running database migrations or scripts, and restarting services using the Azure App Service Deploy task linked to your subscription and app details.
 13. Set Triggers and Approvals
Enable continuous deployment trigger so the release pipeline runs automatically after a successful build. For production environments, configure pre-deployment approvals to ensure manual verification before release.
 14. Monitor Pipelines and Manage Logs
View all pipeline runs under the Runs section.
Check logs for build/test/deploy stages to debug any errors.
You can also integrate email alerts or Microsoft Teams notifications for build failures.
 15. Review and Maintain Pipelines
Regularly update your pipeline tasks or YAML configurations as your application grows.
Ensure pipeline runs are clean and artifacts are stored securely.
Integrate quality gates and code coverage policies to maintain code quality.

Pipelines

Recent All Runs

Filter pipelines

New pipeline

Blog Management System (3)

#20250515.1 • Set up CI with Azure Pipelines

Individual CI for main

Thursday 19s

#20250515.1 • Set up CI with Azure Pipelines

Run new

This run is being retained as one of 3 recent runs by main (Branch).

View retention leases

Summary Code Coverage

Individual CI by Sai Sanjay S V

Repository and version

Blog Management System

main d93823c9

Time started and elapsed

Thu at 10:14 pm

19s

Related

0 work items

Tests and coverage

Get started

View 7 changes

Jobs

Name	Status	Duration
Job	Success	12s

RESULT

Thus, the pipelines for the given project **Blog Management System** has been executed successfully.

EXP NO: 10

GITHUB: PROJECT STRUCTURE & NAMING CONVENTIONS

Aim:

To provide a clear and organized view of the project's folder structure and file naming conventions, helping contributors and users easily understand, navigate, and extend the Blog Management System.

GitHub Project Structure

The screenshot shows a GitHub repository page for 'Blog-Management-System'. The repository was created by 'SaiSanjay-142' and has 1 branch and 0 tags. The main branch contains 5 commits. The files listed are README.md, add-post.html, edit-post.html, index.html, login.html, and view-post.html. The README file is currently selected. The repository has no description, activity, or releases. It also has no packages published.

File	Commit Message	Last Commit
README.md	Update README.md	last week
add-post.html	Add files via upload	last week
edit-post.html	Add files via upload	last week
index.html	Add files via upload	last week
login.html	Add files via upload	last week
view-post.html	Add files via upload	last week

Result:

The GitHub repository clearly displays the organized project structure and consistent naming conventions, making it easy for users and contributors to understand and navigate the codebase.