

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

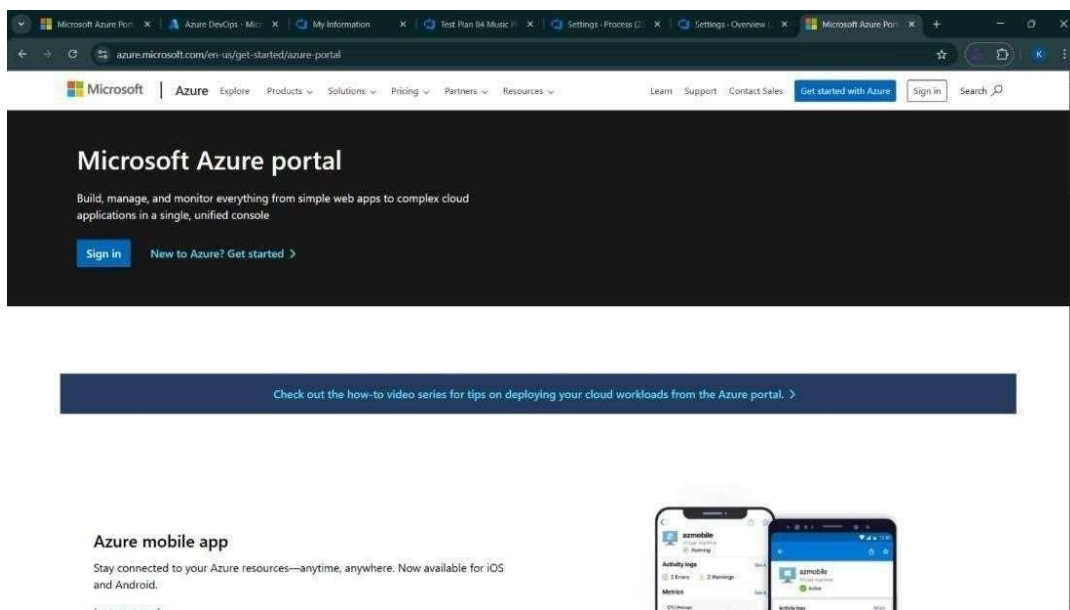
AIM

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

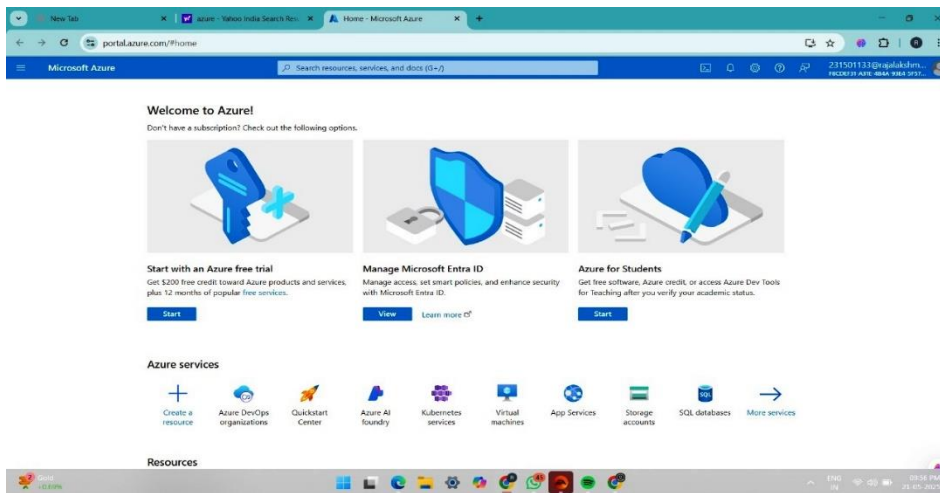
INSTALLATION

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.

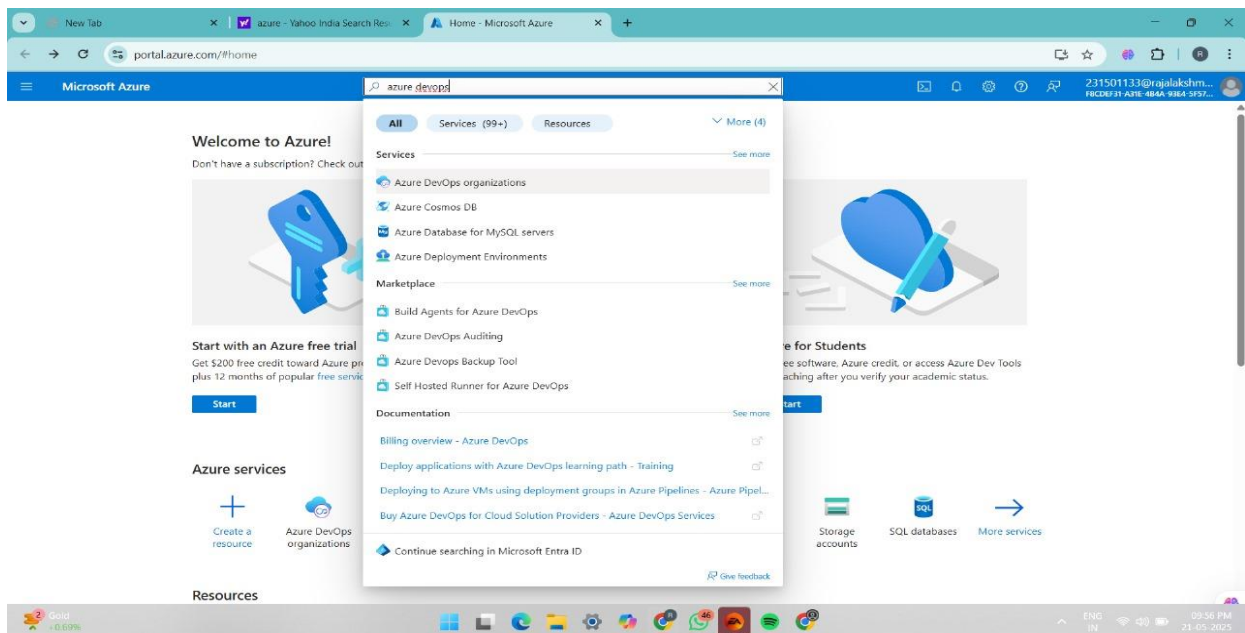
If you don't have a Microsoft account, you can create one here: <https://signup.live.com/?lic=1>



2. Azure home page



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



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.

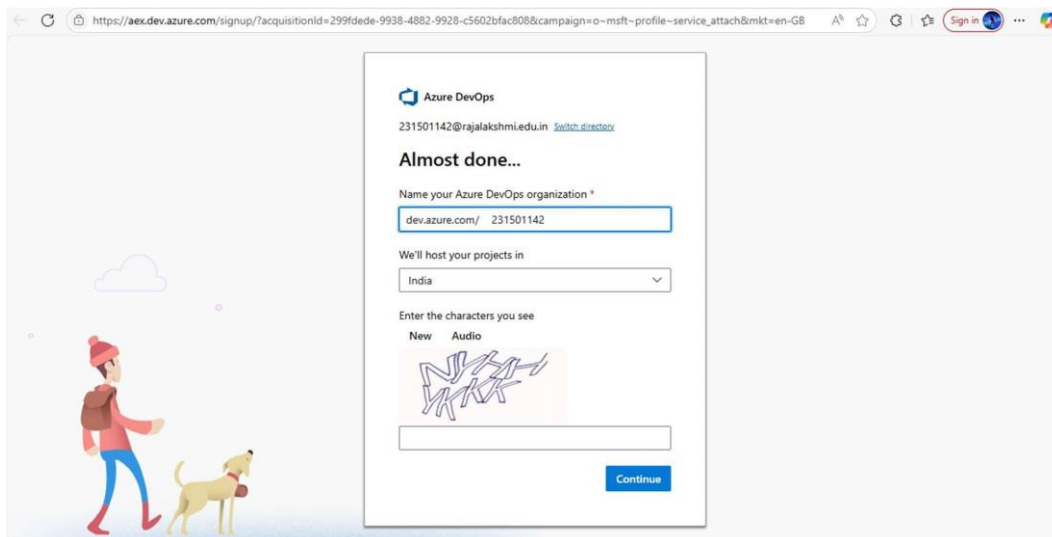
EXP NO: 2

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



The screenshot shows the Azure DevOps sign-up process. The user is at the 'Almost done...' stage. The organization name is 'dev.azure.com/ 231501142' and the region is 'India'. A CAPTCHA challenge is displayed with the text 'Enter the characters you see' and 'New Audio'.

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 [Close]

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](#).

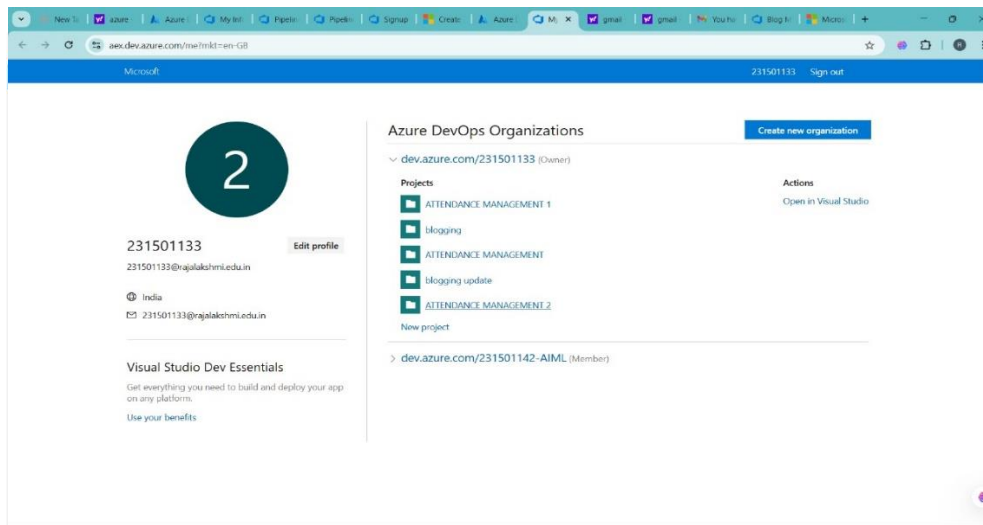
^ Advanced

Version control ⓘ
Git

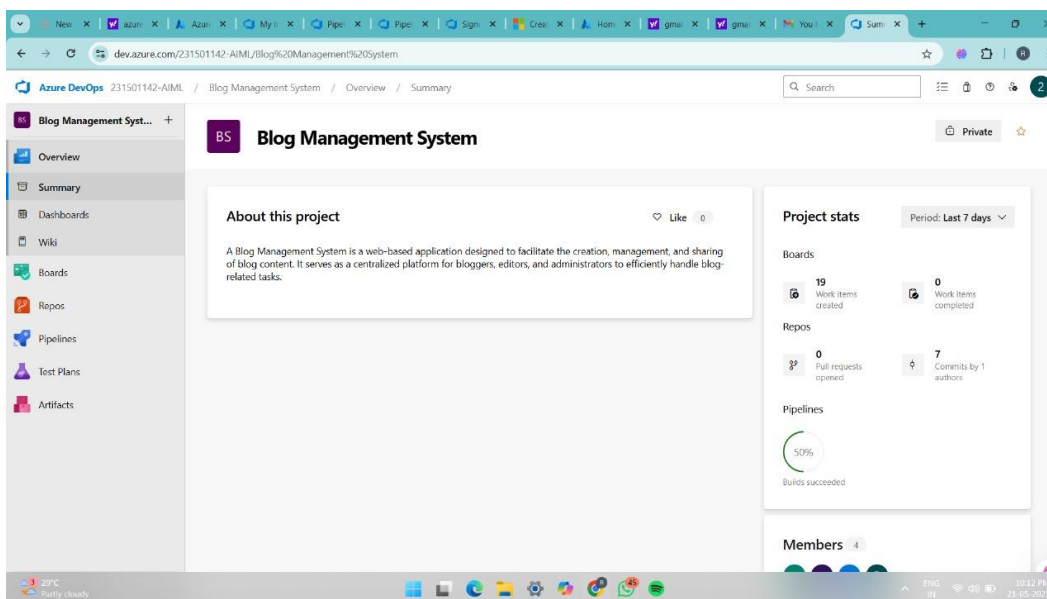
Work item process ⓘ
Agile

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.



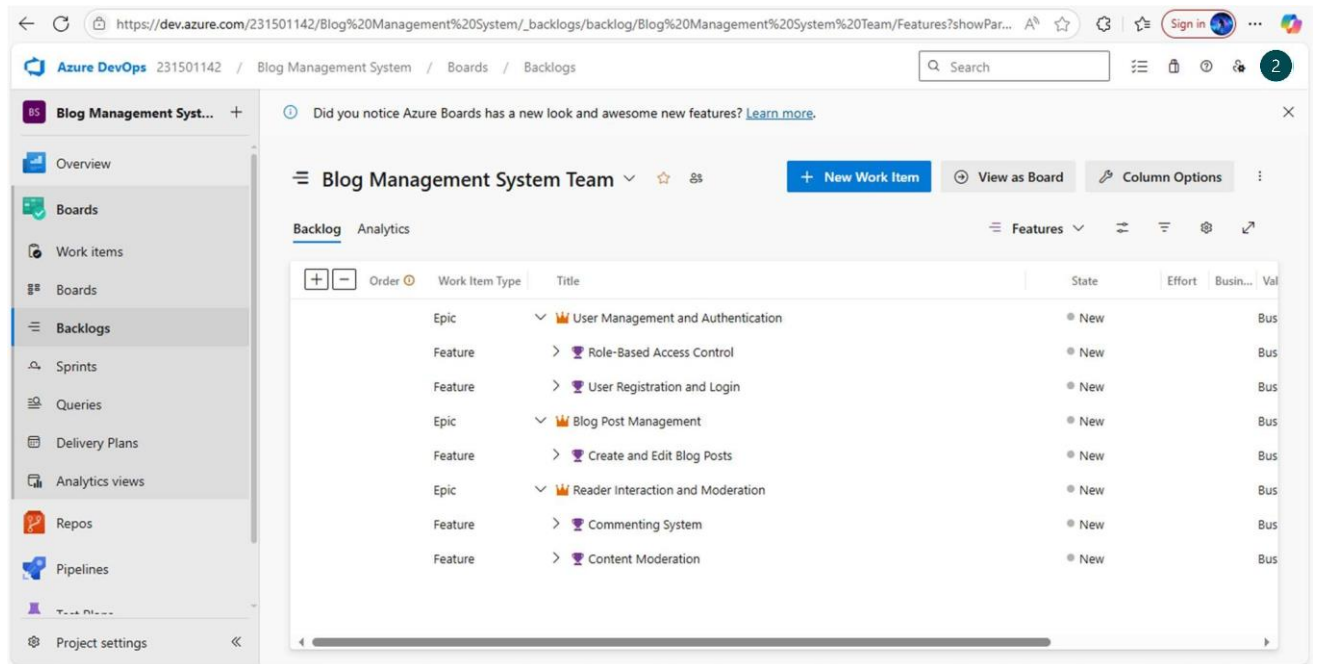
4. Project dashboard



5. To manage user stories:

a. 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.

b. 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.



RESULT

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

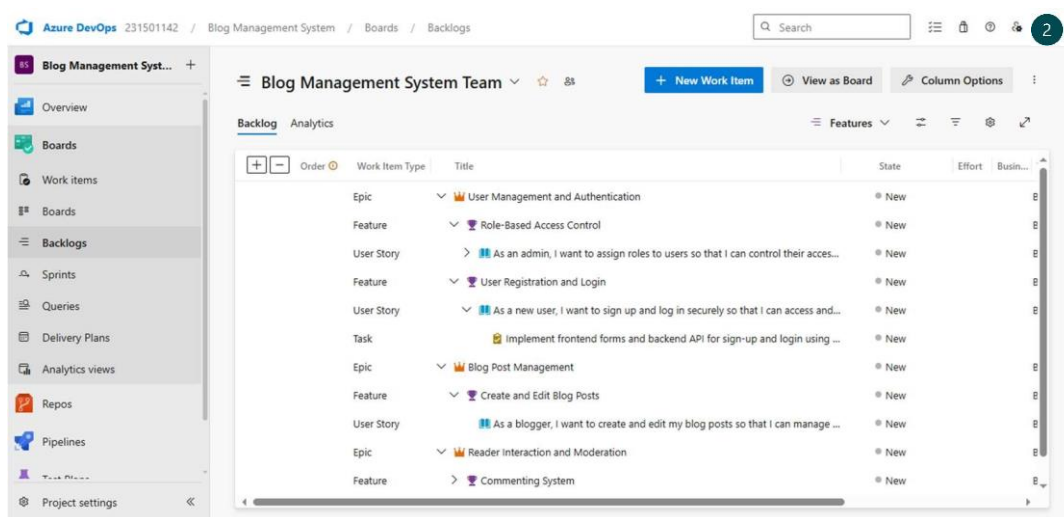
EXP NO:3

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

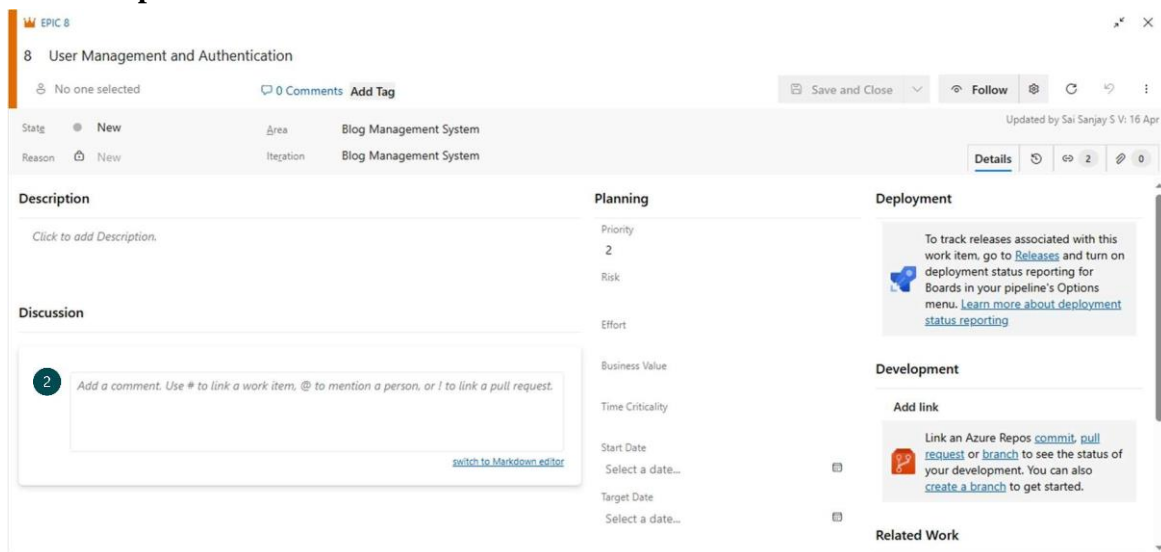
AIM

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

Create Epic, Features, User Stories, Task



1. Fill in Epics



2. Fill in Features

FEATURE 13

13 Role-Based Access Control

No one selected0 CommentsAdd Tag

Save and CloseFollow

Updated by Sai Sanjay S V: 16 Apr

Details

State: NewReason: NewArea: Blog Management SystemIteration: Blog Management System\Sprint 1

Description

Click to add Description.

Discussion

2Add a comment. Use # to link a work item, @ to mention a person, or ! to link a pull request.

Planning

Priority: 2Risk: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.

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 selected0 CommentsAdd Tag

Save and CloseFollow

Updated by Sai Sanjay S V: 16 Apr

Details

State: NewReason: NewArea: Blog Management SystemIteration: Blog Management System\Sprint 1

Description

Click to add Description.

Acceptance Criteria

Click to add Acceptance Criteria.

Discussion

2Add a comment. Use # to link a work item, @ to mention a person, or ! to link a pull request.

Planning

Story Points:Priority: 2Risk:

Classification

Value area:Business:

Deployment

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

Development

Add link

Link an Azure Repos [commit](#), [pull request](#) or [branch](#) to see the status of your development. You can also [create a branch](#) to get started.

Related Work

RESULT

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

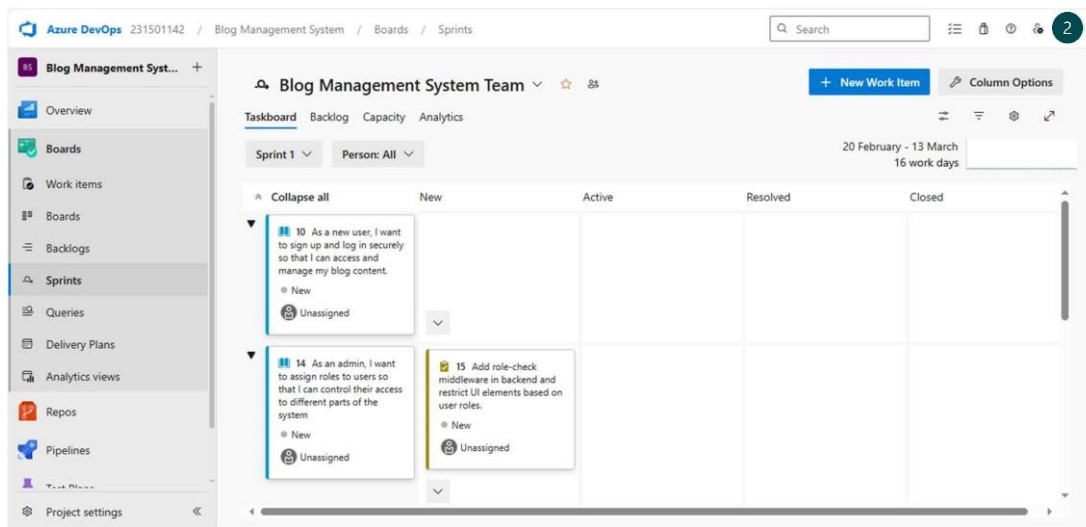
EXP NO:
4

SPRINT PLANNING

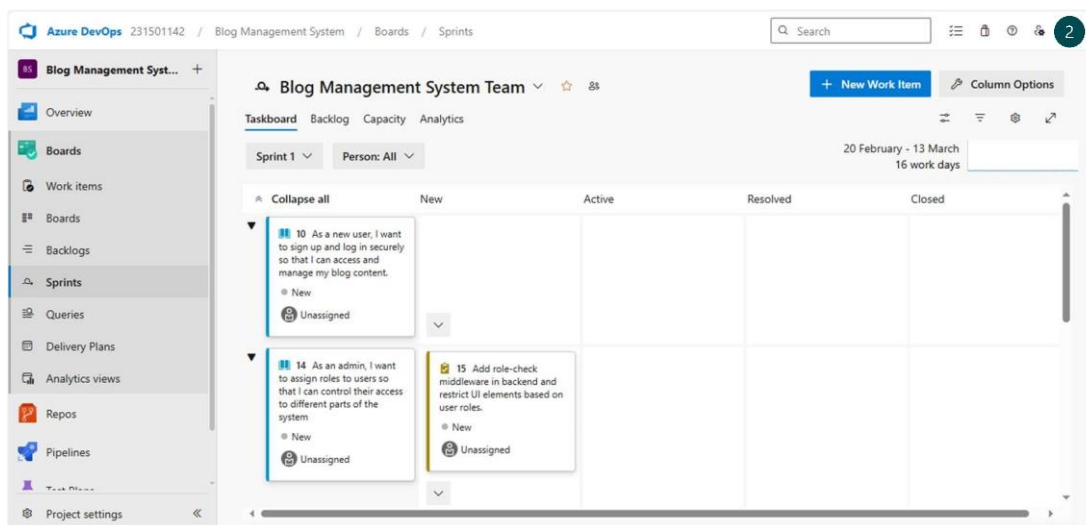
AIM

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

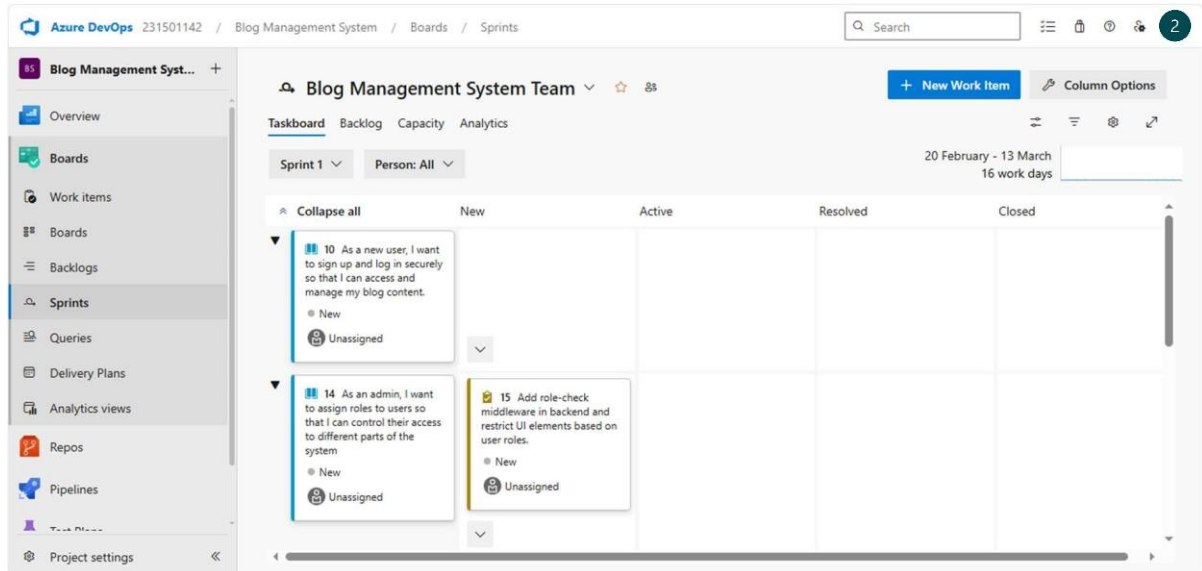
Sprint Planning Sprint 1



Sprint 2



Sprint 3



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

USER STORY 11

11 As a new user, I want to sign up and log in securely so that I can access and manage my blog content.

No one selected0 CommentsAdd Tag

Save and CloseFollow

Updated by Sai Sanjay S V: 16 Apr

Details20

Description

Click to add Description.

Acceptance Criteria

Click to add Acceptance Criteria.

Discussion

2Add a comment. Use # to link a work item, @ to mention a person, or ! to link a pull request.

Planning

Story Points

Priority

2

Risk

Classification

Value area

Business

Deployment

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

Development

Add link

Link an Azure Repos [commit](#), [pull request](#) or [branch](#) to see the status of your development. You can also [create a branch](#) to get started.

Related Work

RESULT

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

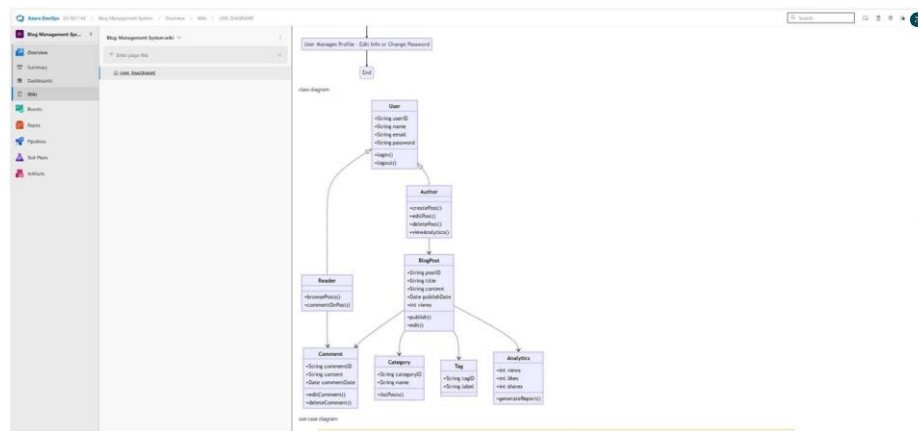
EXP NO: 6

DESIGNING CLASS AND SEQUENCE DIAGRAMS FOR PROJECT ARCHITECTURE

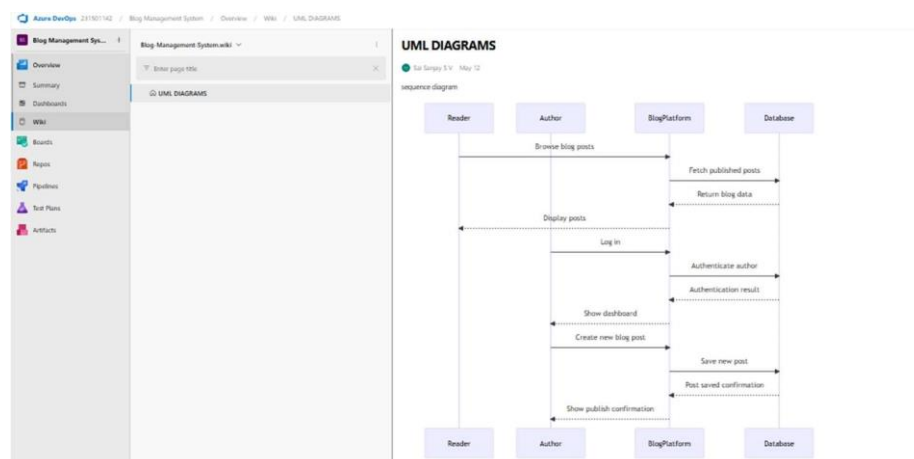
AIM

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.

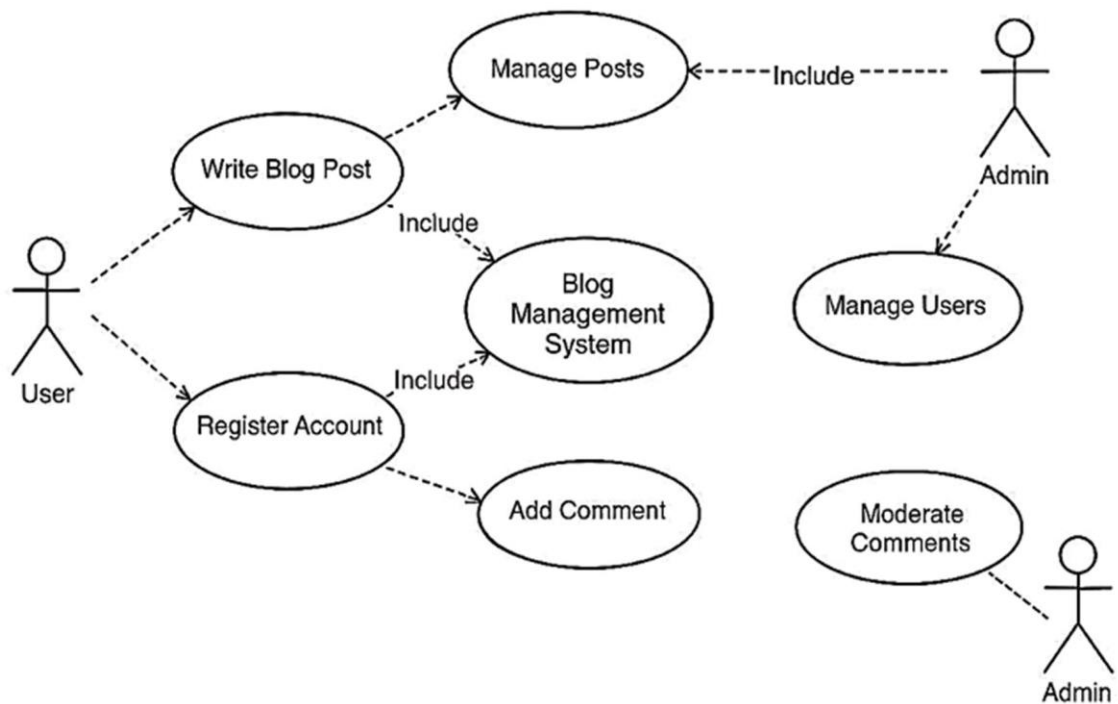
EXP NO: 7

DESIGNING USE-CASE AND ACTIVITY DIAGRAMS FOR PROJECT STRUCTURE

AIM

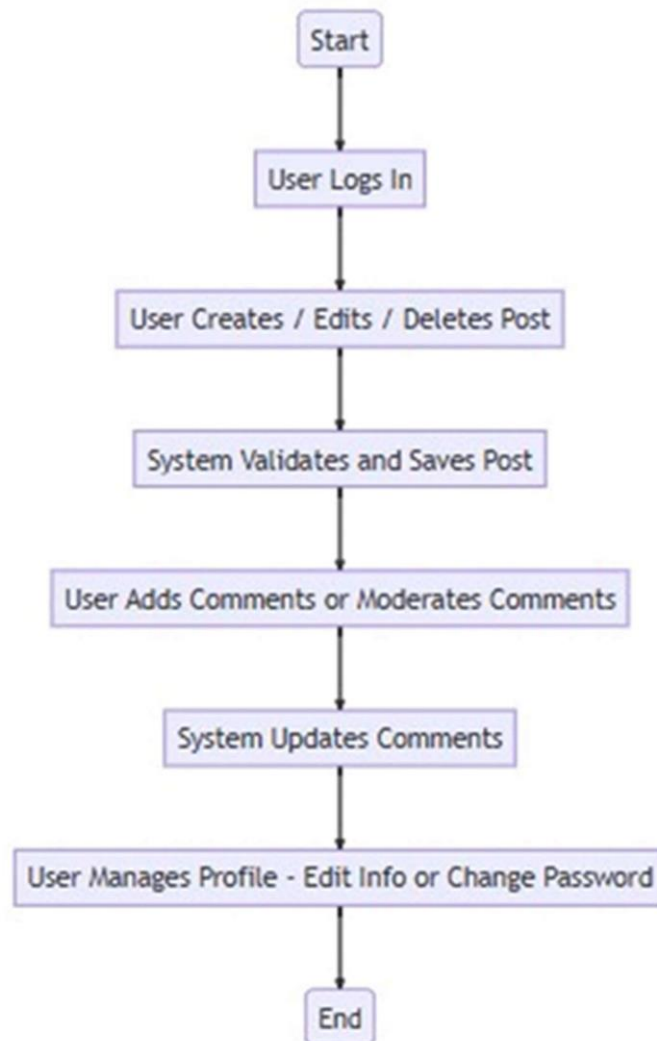
To Design an Use-Case Diagram and Activity Diagram for the given Project.

7A. Use-Case Diagram



7B. Activity Diagram

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

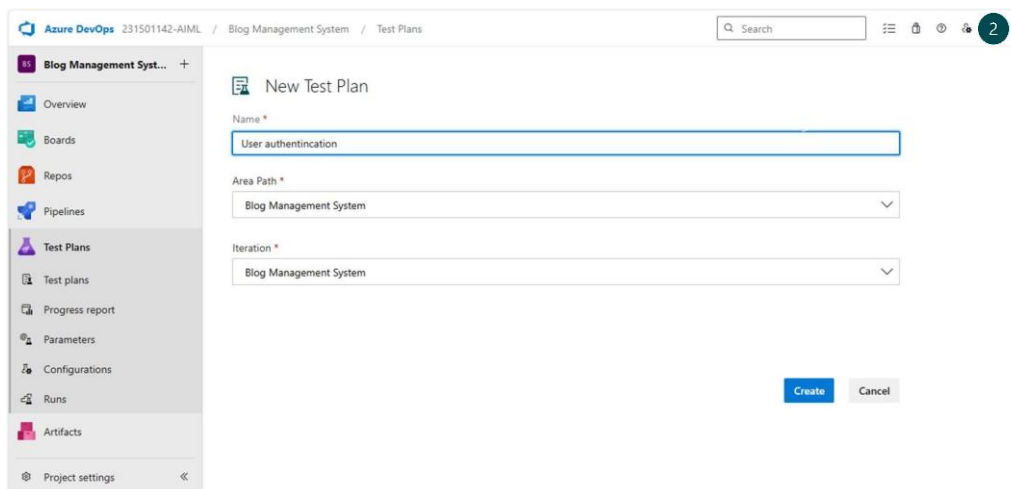
- Grouped by functionality such as: ○ Login and Registration ○ Quizzes Submission ○ Viewing

Results ○ Admin Functions • Improves organization and enables focused execution in Azure DevOps.

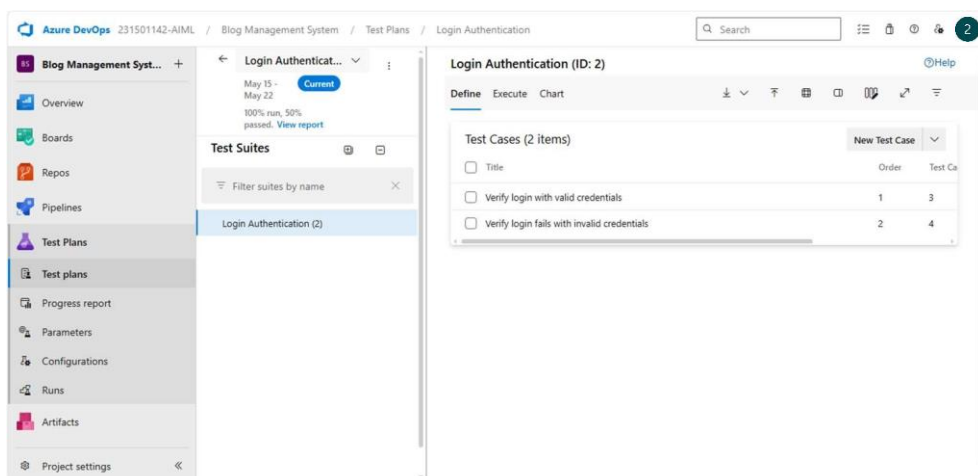
8. Prioritize and Review

- 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



2. Test suite



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:

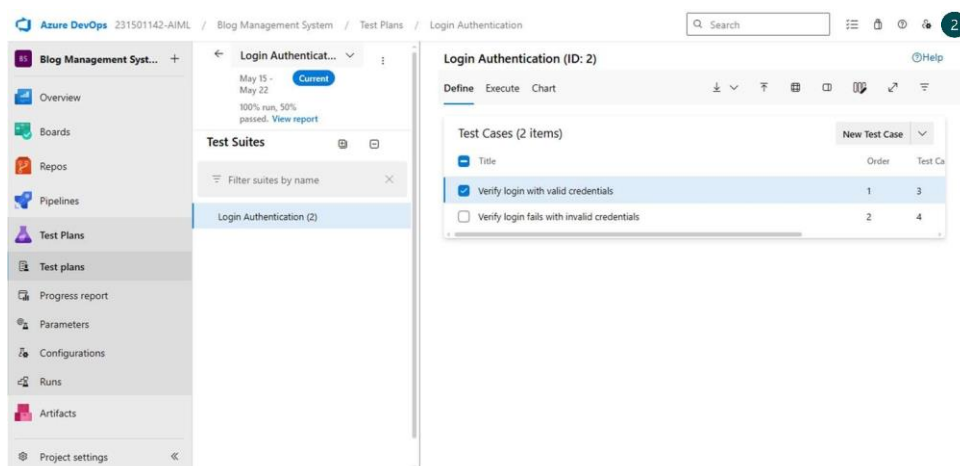
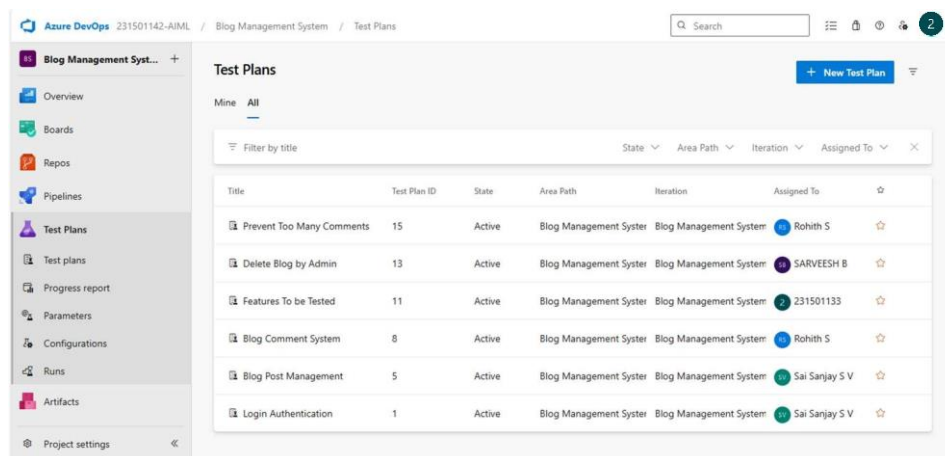
- Log in and go to "Create Post" page.
- Enter title but leave content empty.
- Click "Schedule".

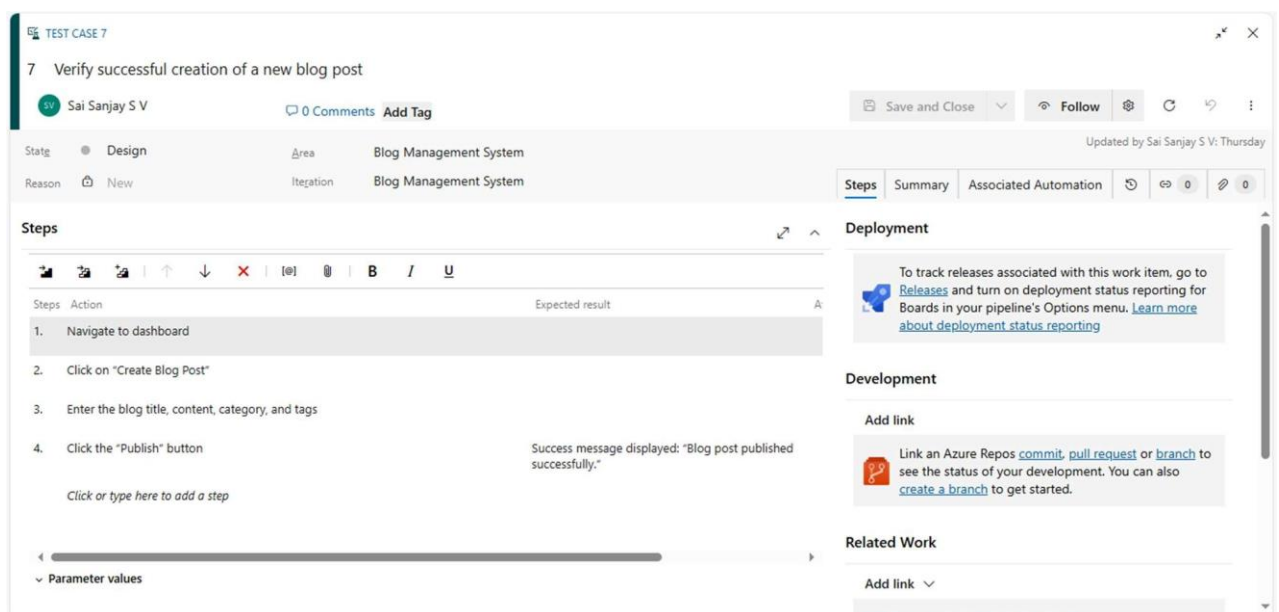
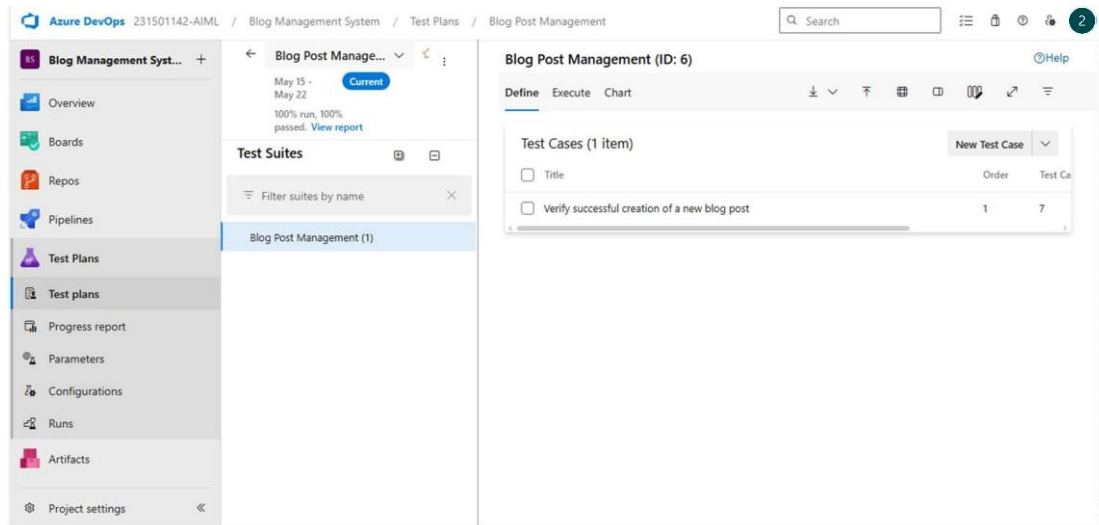
Expected Results:

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

Type: Error Path

Test Cases





Azure DevOps 231501142-AIML / Blog Management System / Test Plans / Blog Comment System

Search

Blog Management Syst... +

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

Blog Comment Sy... Current

May 15 - May 22
100% run, 100% passed. [View report](#)

Test Suites

Filter suites by name

Blog Comment System (1)

Blog Comment System (ID: 9)

Define Execute Chart

Test Cases (1 item)

New Test Case

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

TEST CASE 10

10 Verify that a logged-in user can add a comment to a blog post

SARVEESH B

0 Comments Add Tag

Save and Close Follow

Updated by Sai Sanjay S V: Thursday

State Design Area Blog Management System

Reason New Iteration Blog Management System

Steps

Deployment

Development

Related Work

Steps

Steps	Action	Expected result
1.	Log in as a registered user	
2.	Navigate to the specific blog post	
3.	Scroll to the comment section	
4.	Enter the comment text	
5.	Click the post comment button	Comment is successfully posted and displayed below the blog post

Click or type here to add a step

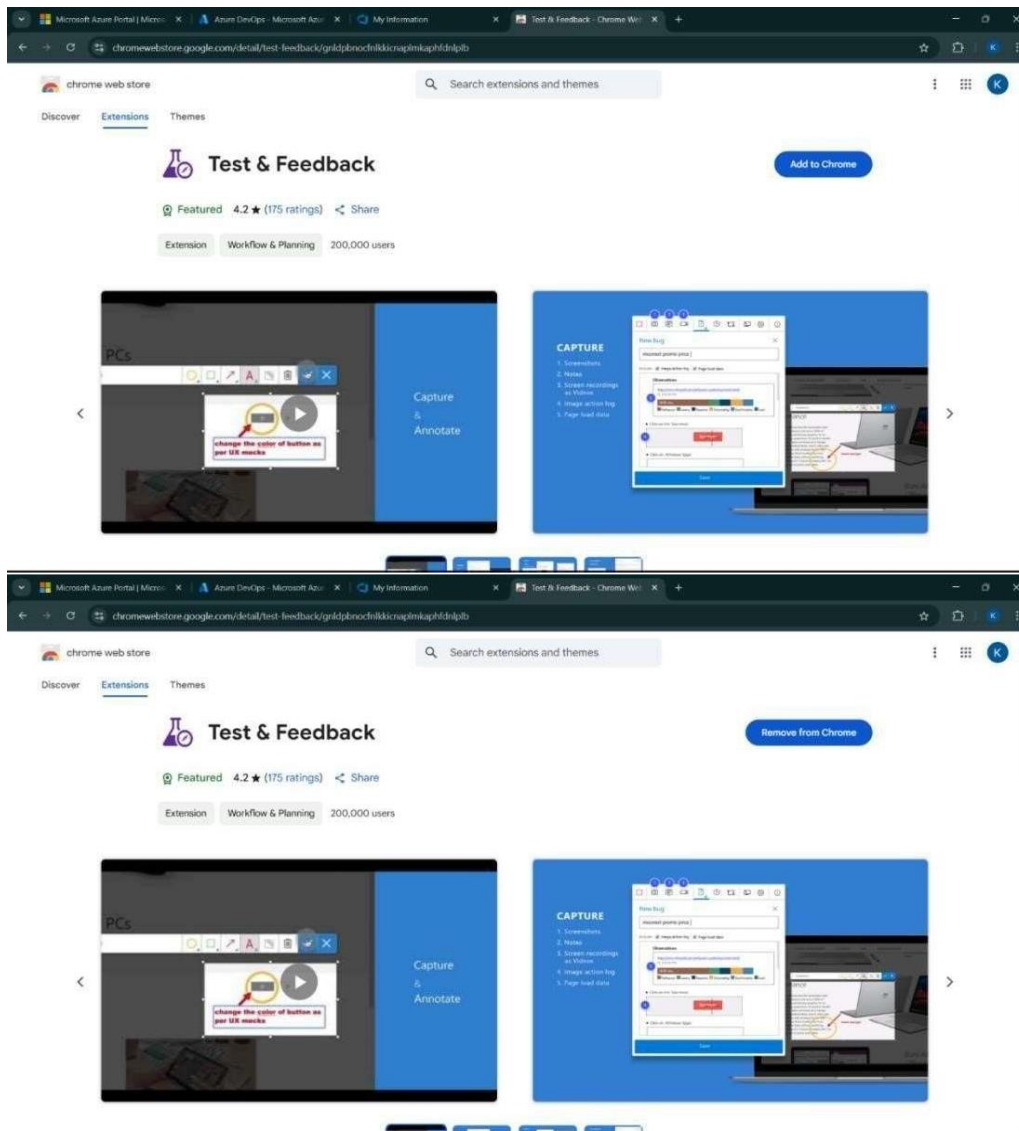
Parameter values

Add link

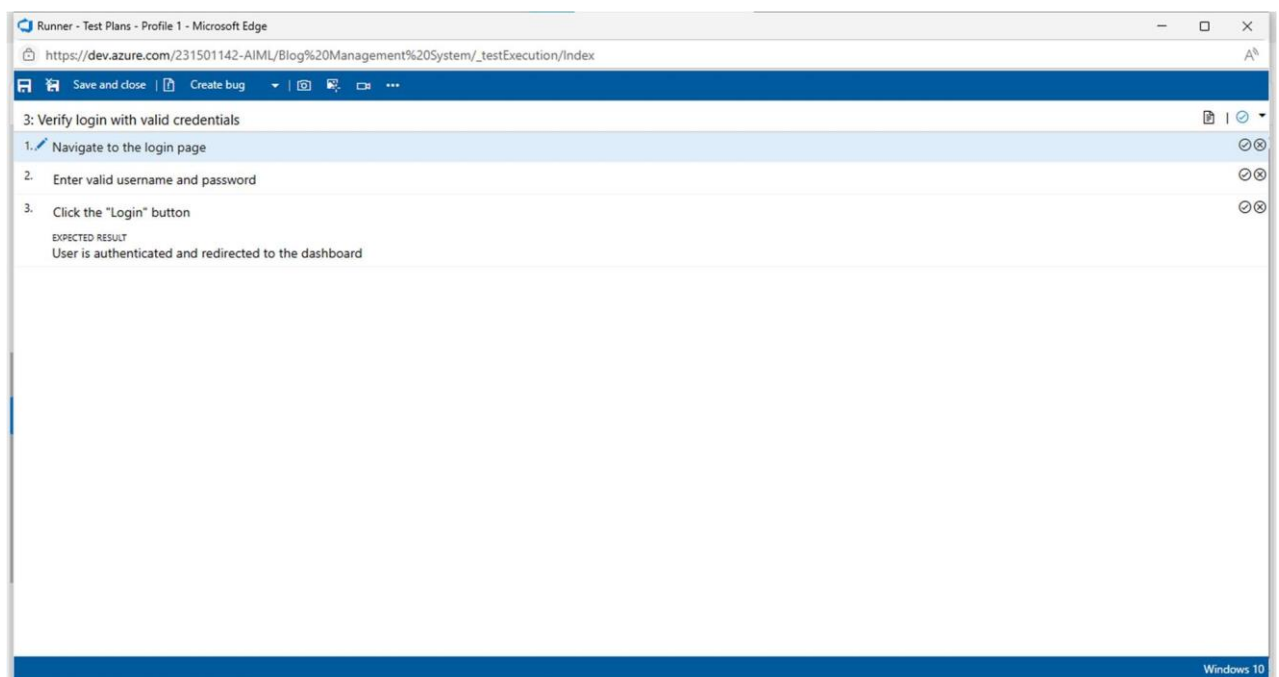
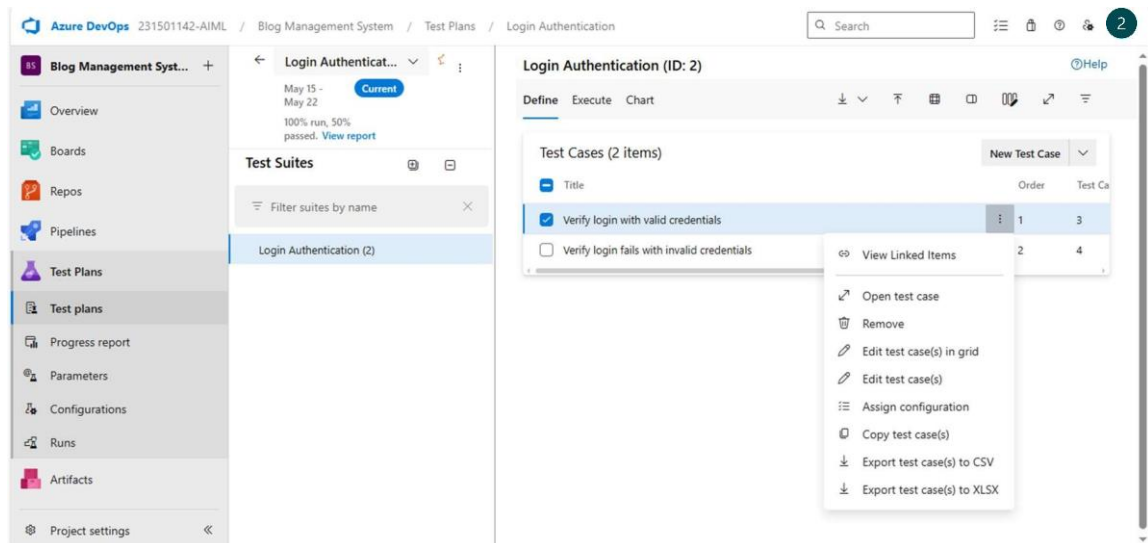
Add link

Add an existing work item as a parent

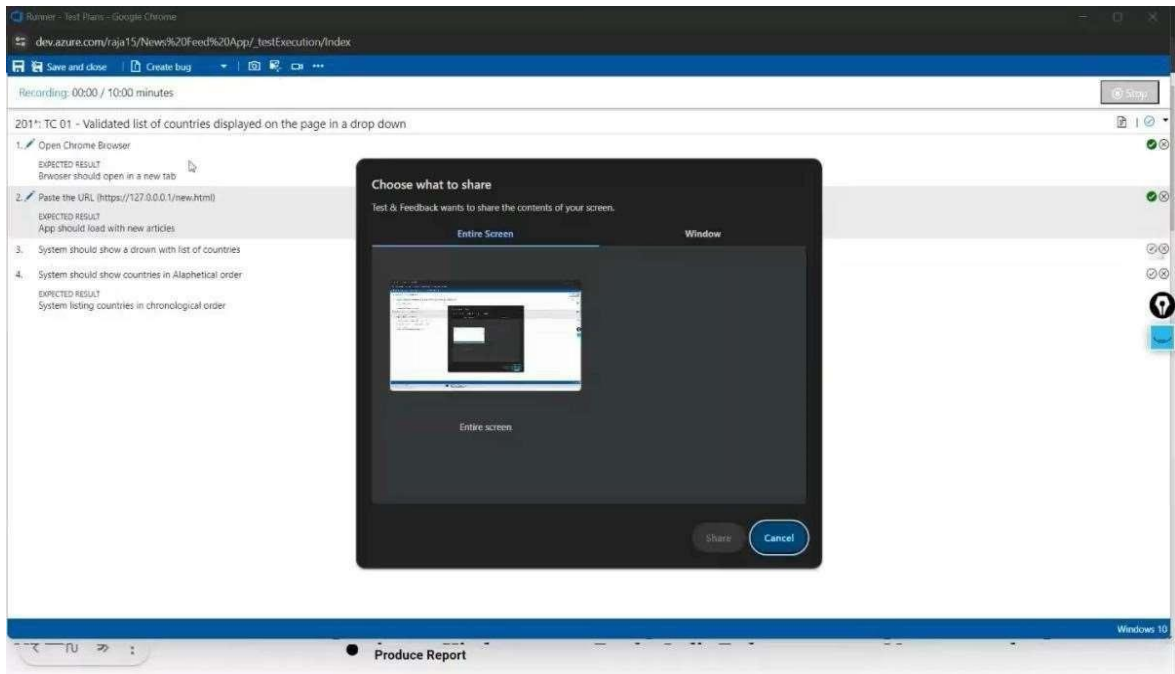
4. Installation of test



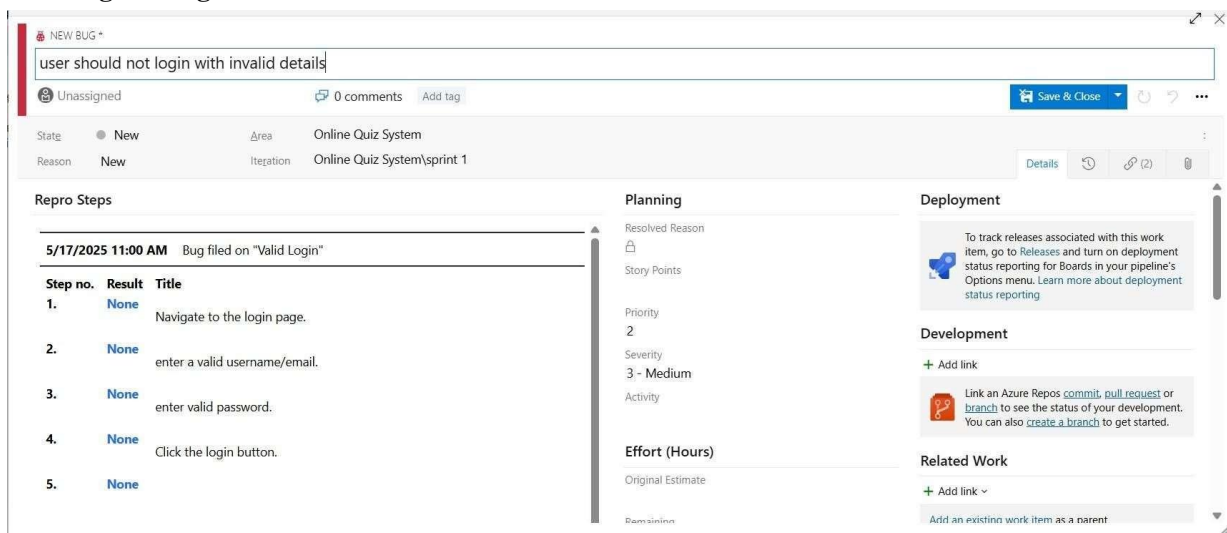
5. Running the test cases



6. Recording the test case



7. Creating the bug



9. Test report summary

NEW BUG

User should not enter invalid details

Unassigned0 commentsAdd tag

Save & Close

State: NewArea: Online Quiz SystemReason: NewIteration: Online Quiz System\sprint 1

Details

Repro Steps

5/17/2025 11:09 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	

Planning

Resolved Reason

Story Points

Priority: 2

Severity: 3 - Medium

Activity

Effort (Hours)

Original Estimate

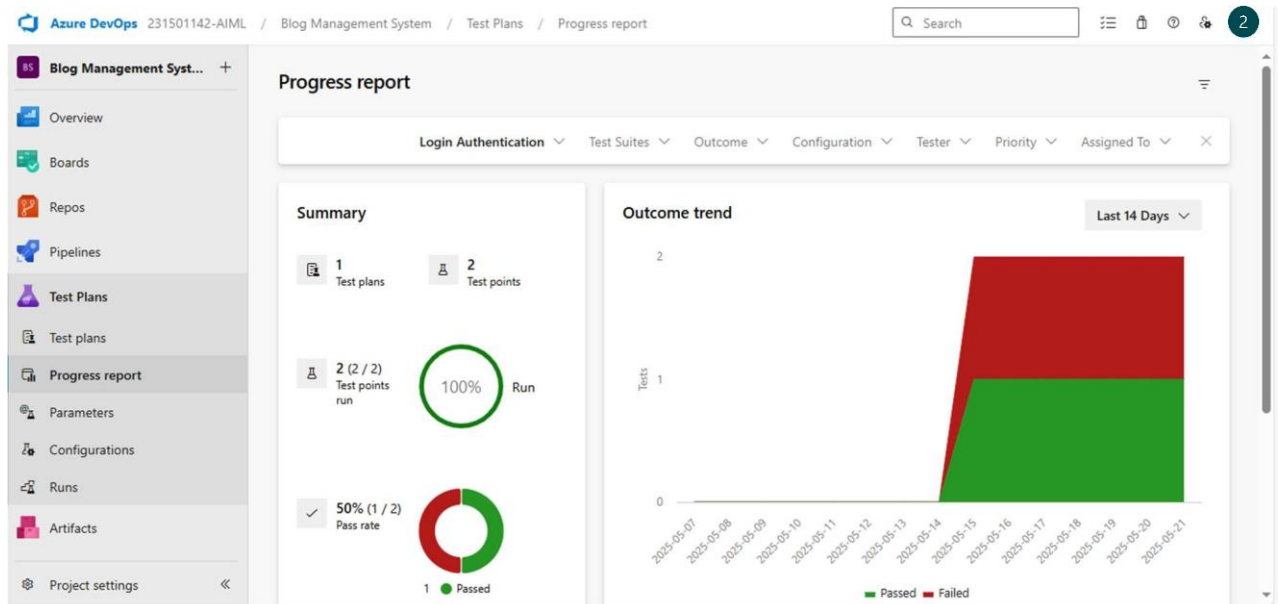
Deployment

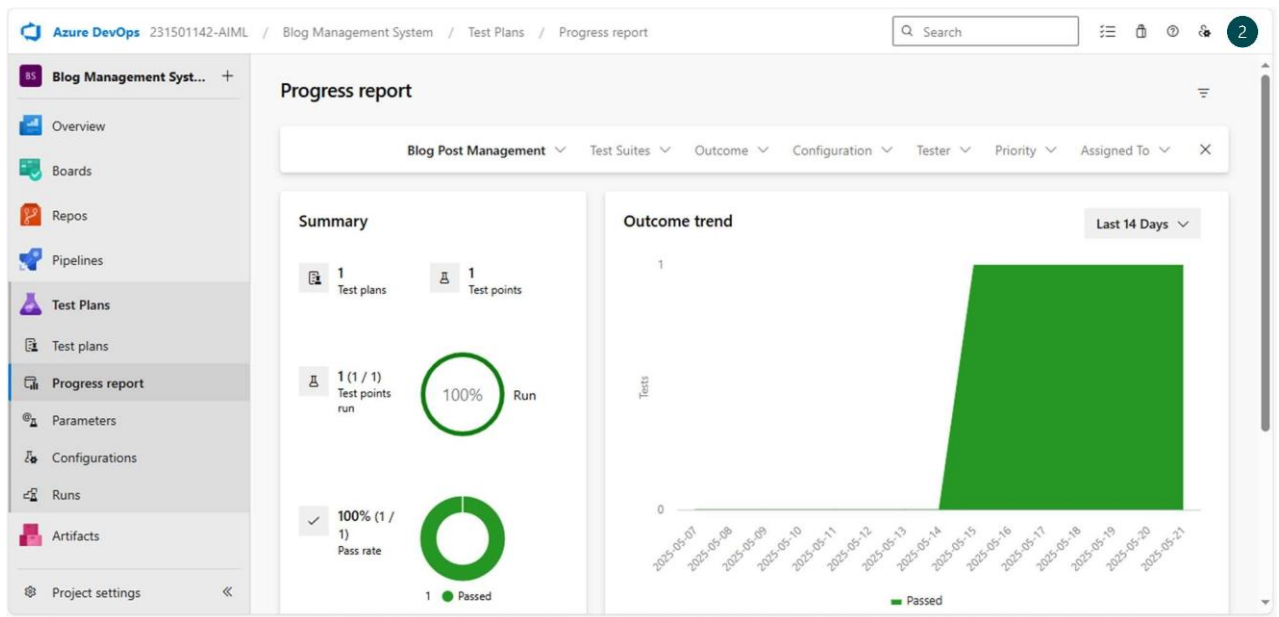
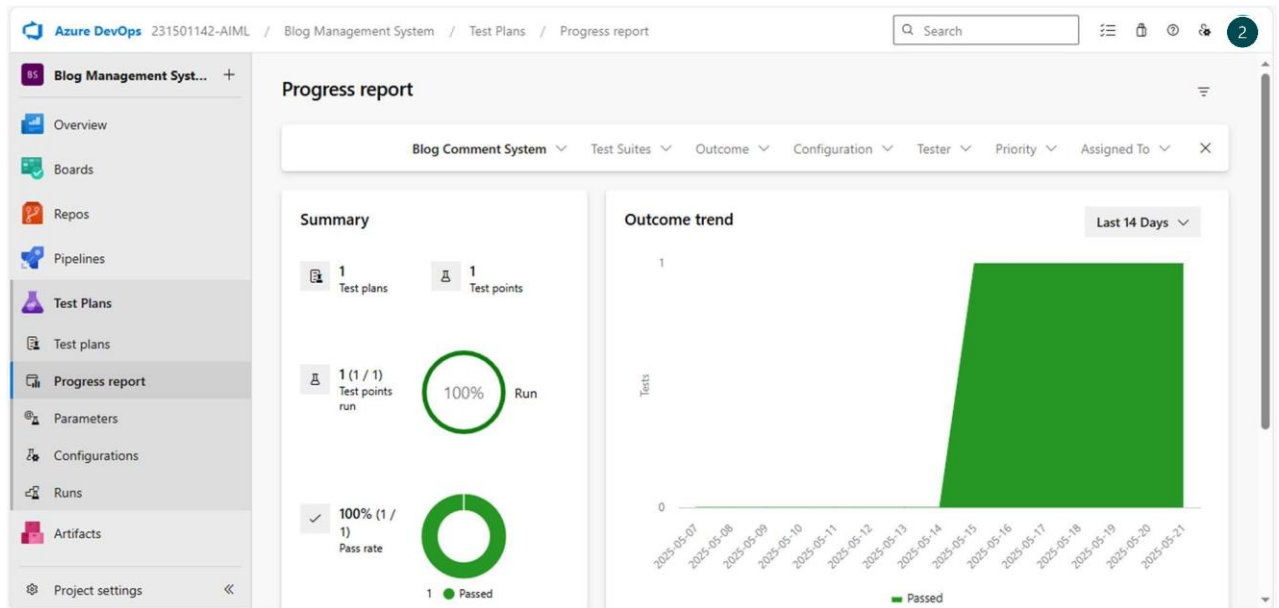
Development

Related Work

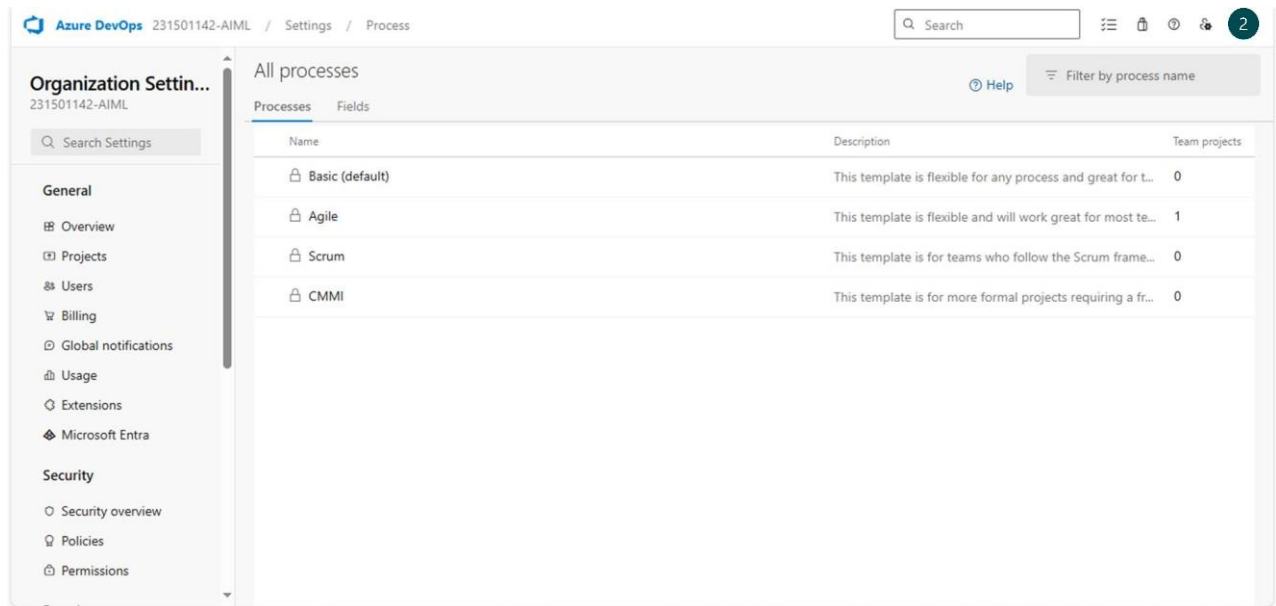
- Assigning bug to the developer and changing state

10. Progress report



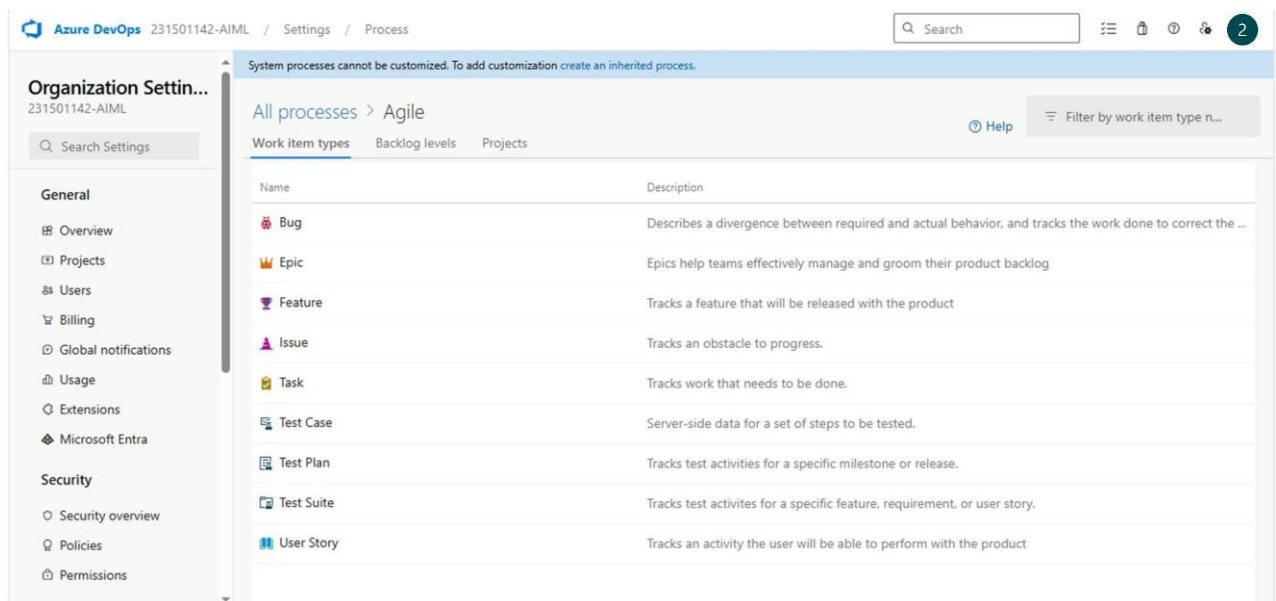


11. Changing the test template



The screenshot shows the 'All processes' page in Azure DevOps. The left sidebar contains the 'Organization Settings' menu with options like General, Overview, Projects, Users, Billing, Global notifications, Usage, Extensions, Microsoft Entra, Security, Security overview, Policies, and Permissions. The main content area displays a table of processes.

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 'All processes > Agile' page in Azure DevOps. The left sidebar is the same as the previous screenshot. The main content area displays a table of work item types.

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 activities 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.

Azure DevOps 231501142-AIML / Blog Management System / Pipelines

Search

New pipeline

Filter pipelines

Pipelines

Recent All Runs

Recently run pipelines

Pipeline	Last run
Blog Management System (3)	#20250515.1 • Set up CI with Azure Pipelines Individual CI for main Thursday 19s

Project settings

Azure DevOps 231501142-AIML / Blog Management System / Pipelines / Blog Management System (3) / 20250515.1

Search

Run new

This run is being retained as one of 3 recent runs by main (Branch). [View retention leases](#)

#20250515.1 • Set up CI with Azure Pipelines

Blog Management System (3)

Summary Code Coverage

Individual CI by Sai Sanjay S V [View 7 changes](#)

Repository and version	Time started and elapsed	Related	Tests and coverage
Blog Management System main d93823c9	Thu at 10:14 pm 19s	0 work items 0 artifacts	Get started

Jobs

Name	Status	Duration
Job	Success	12s

Project settings

RESULT

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

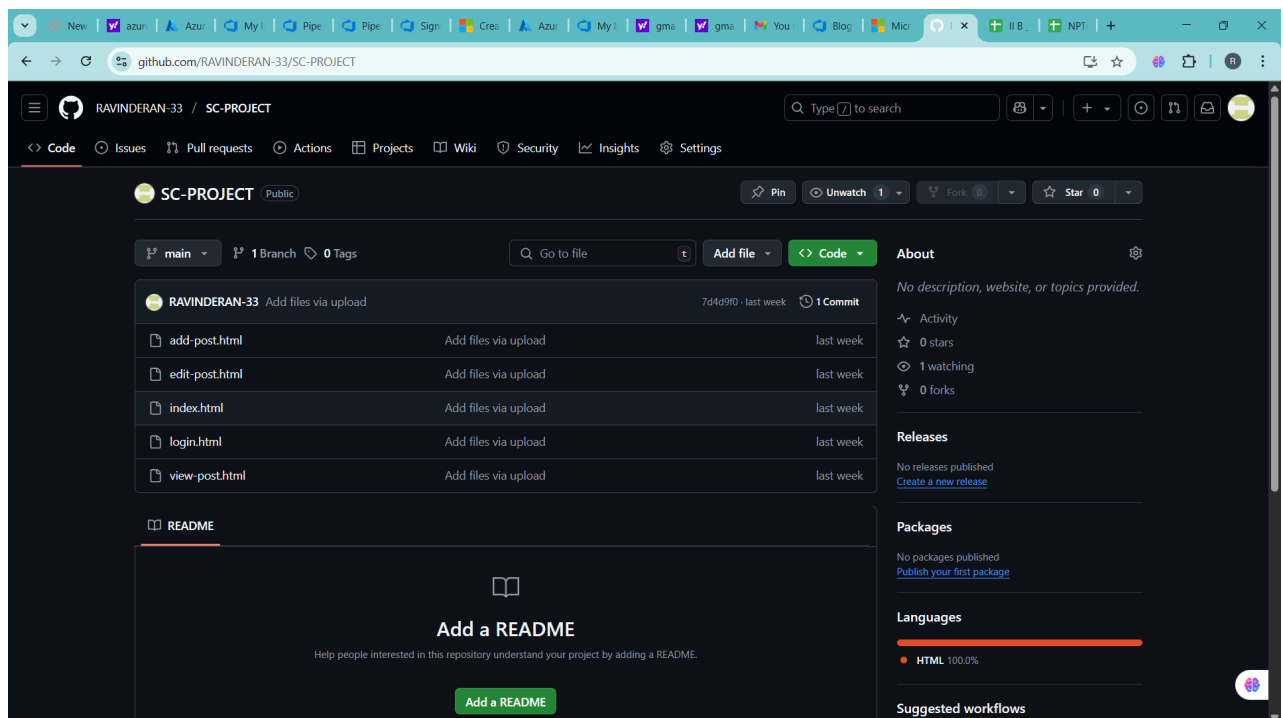
EXP NO: 10

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



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.