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

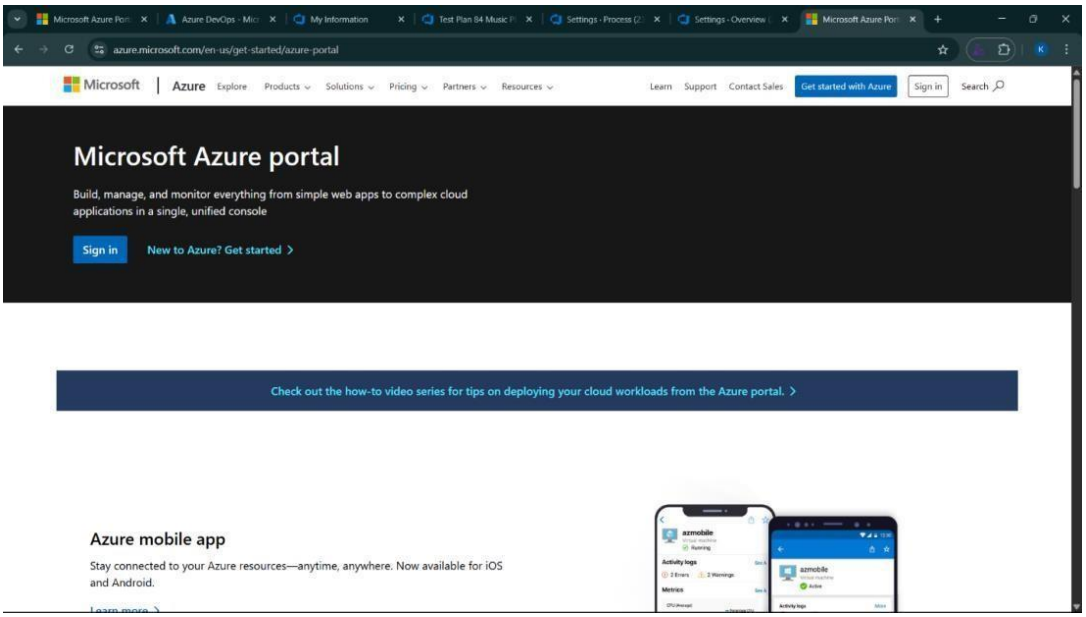
EXP NO: 1 DATE:	AZURE DEVOPS ENVIRONMENT SETUP

AIM:-

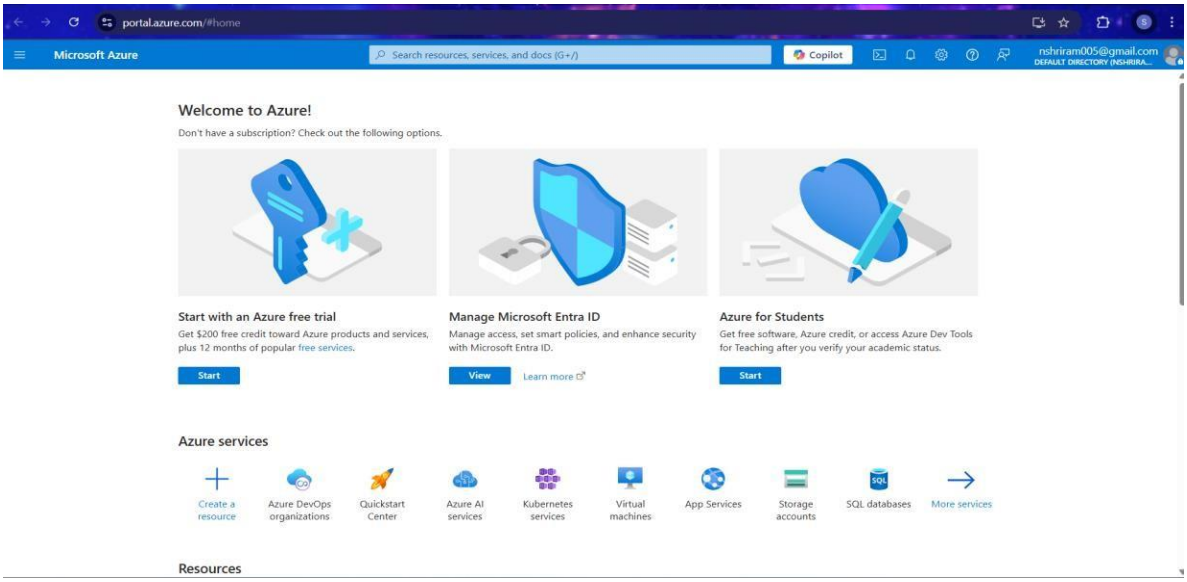
To set up and access the Azure DevOps environment by creating an organization through the Azure portal.

INSTALLATION:-

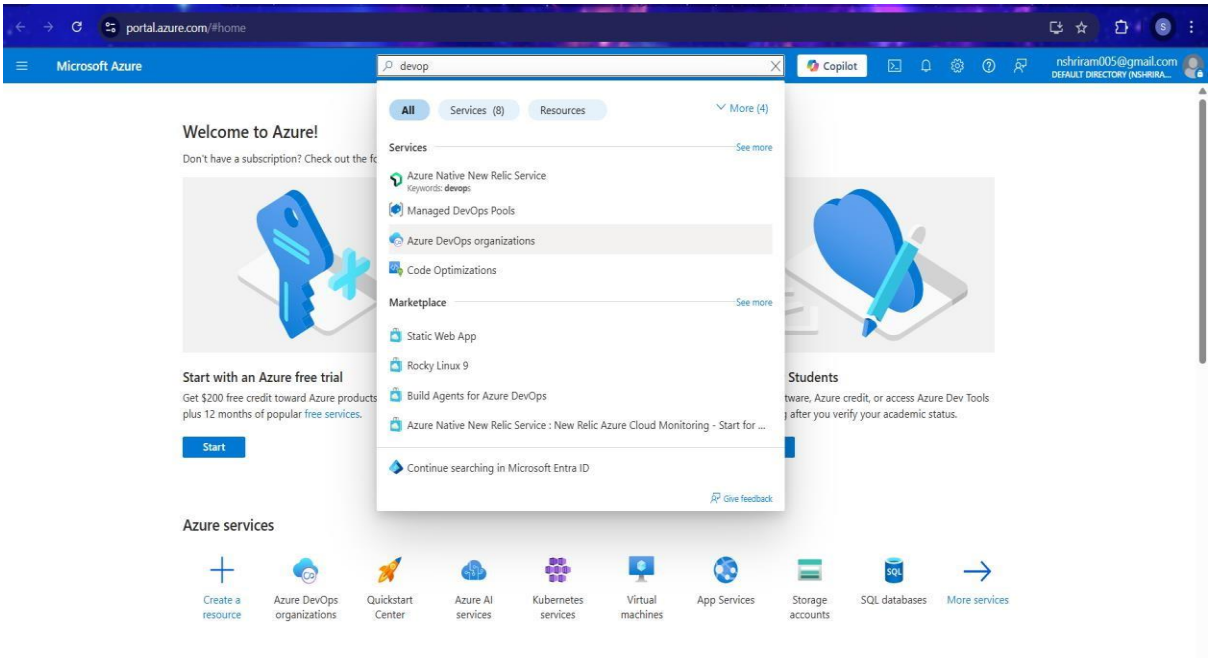
1. Open your web browser and go to the Azure website:
https://azure.microsoft.com/en-us/get- started/azure-portal.
Sign in using your Microsoft account credentials.
If you don't have a Microsoft account, you can create one here: https://signup.live.com/?lic=1



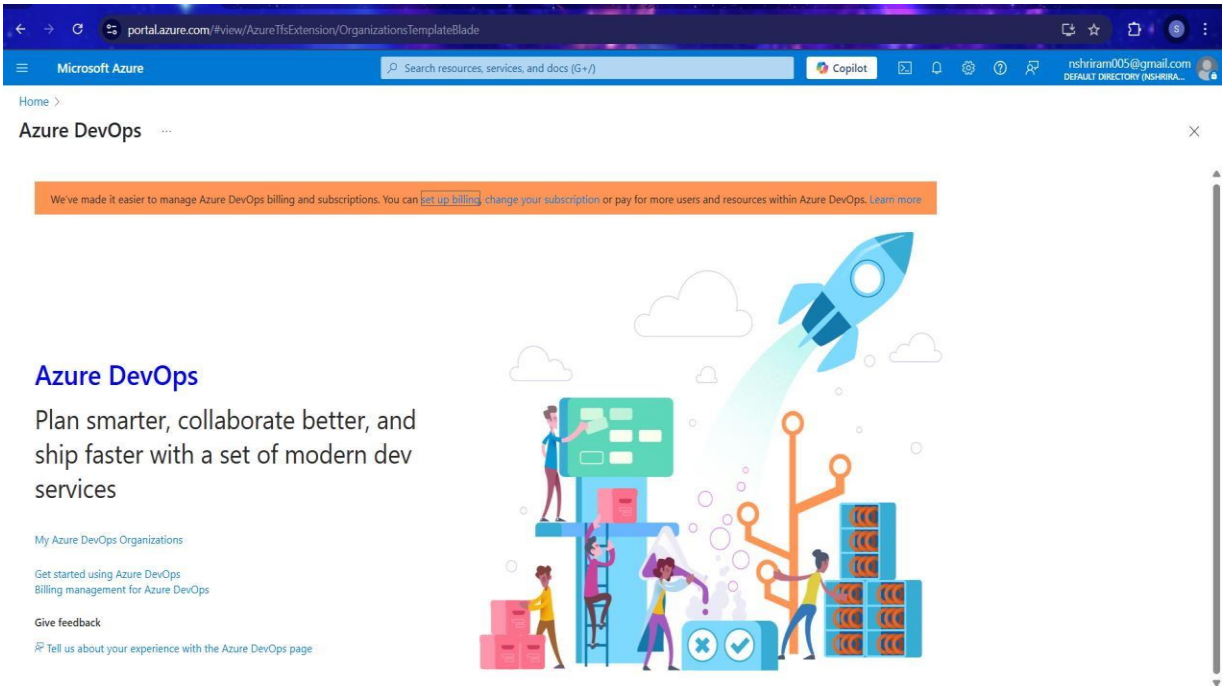
2. Azure home page



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:

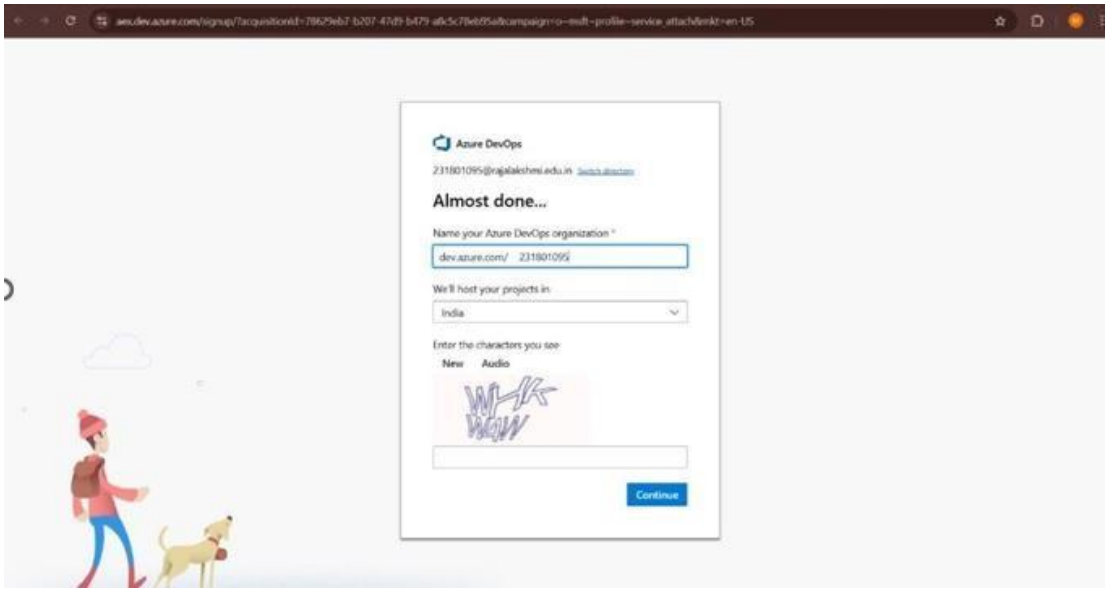
Successfully accessed the Azure DevOps environment and created a new organization through the Azure portal.

EXP NO: 2 DATE:	AZURE DEVOPS PROJECT SETUP AND USER STORY MANAGEMENT

AIM:-

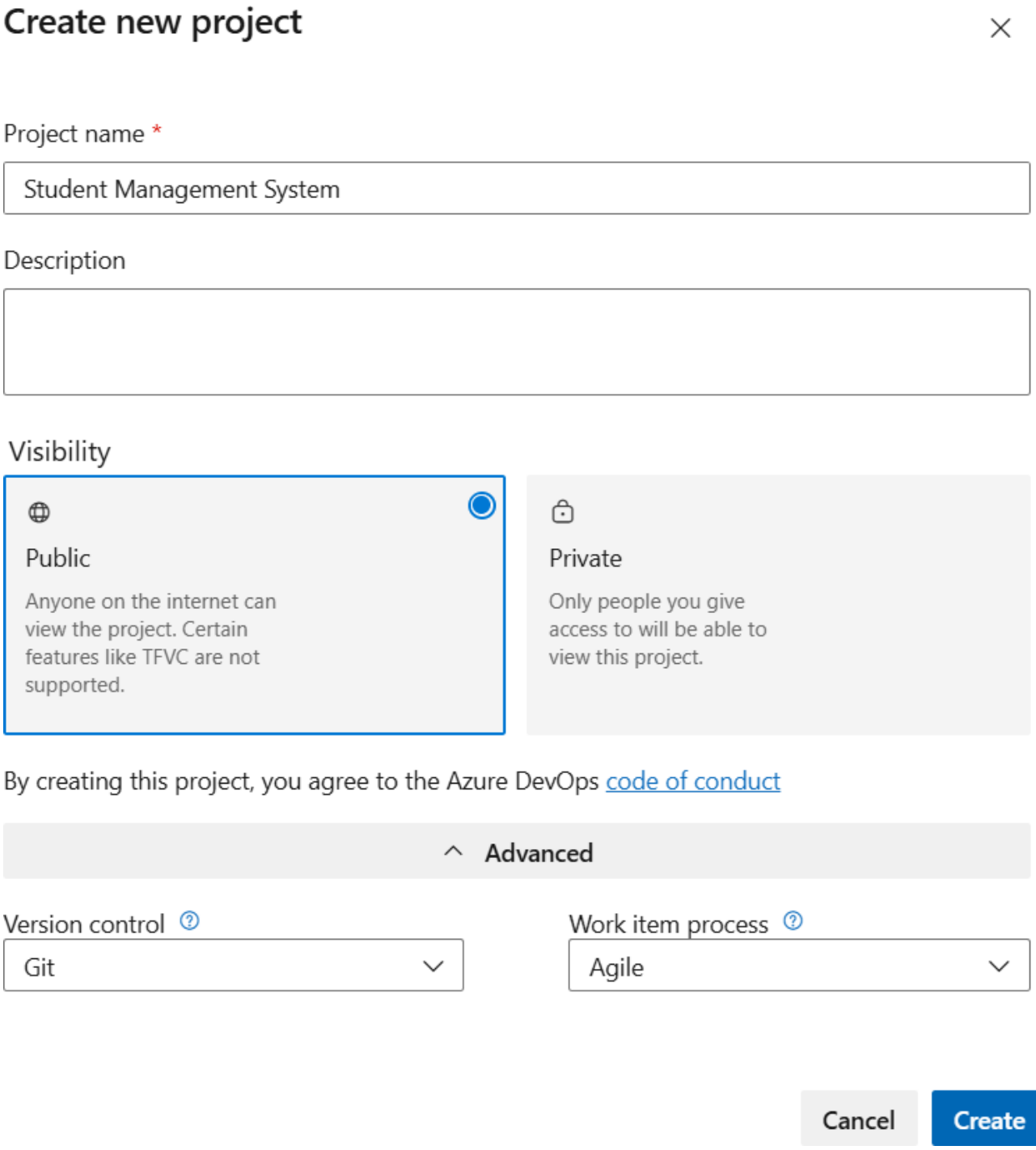
To set up an Azure DevOps project for efficient collaboration and agile work management.

1. Create An Azure Account

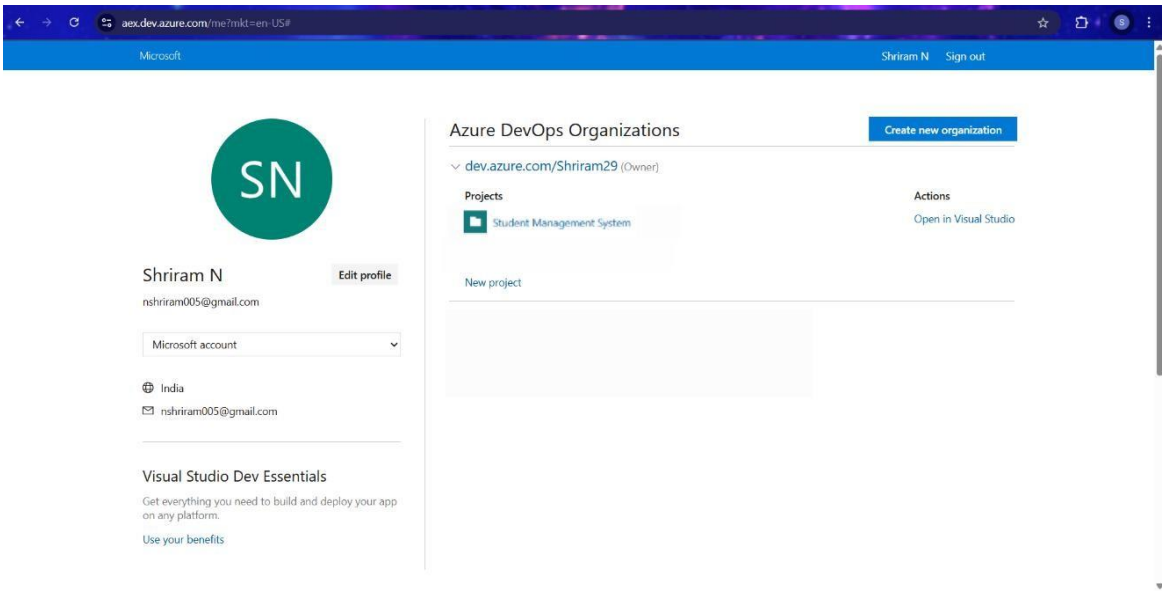


2. Create the First Project in Your Organization

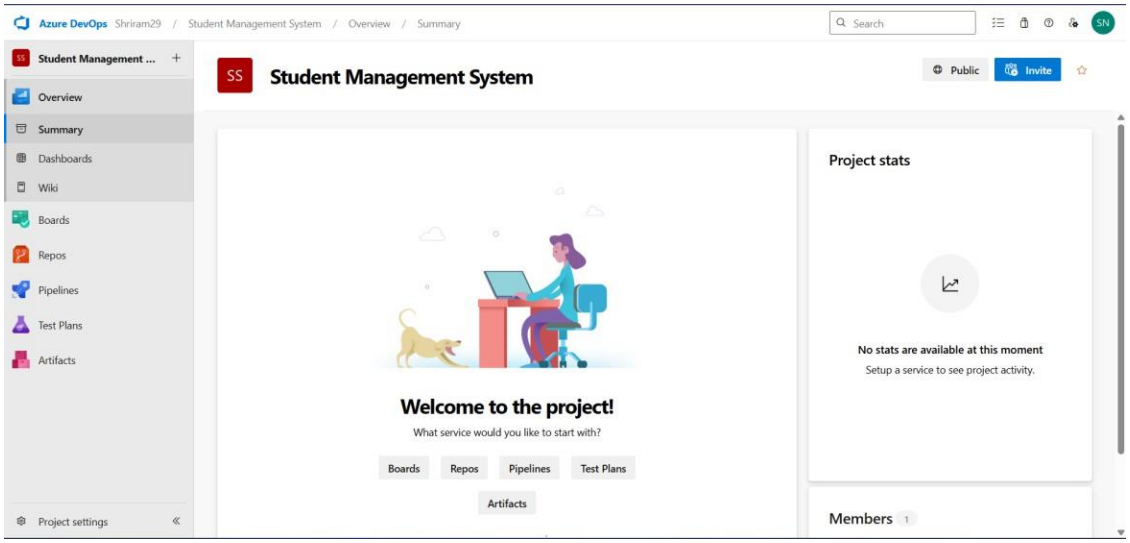
a) After the organization is set up, you'll need to create your first **project**. This is where you'll begin to manage code, pipelines, work items, and more.
b) On the organization's **Home page**, click on the **New Project** button.
c) Enter the project name, description, and visibility options:
Name: 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.



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

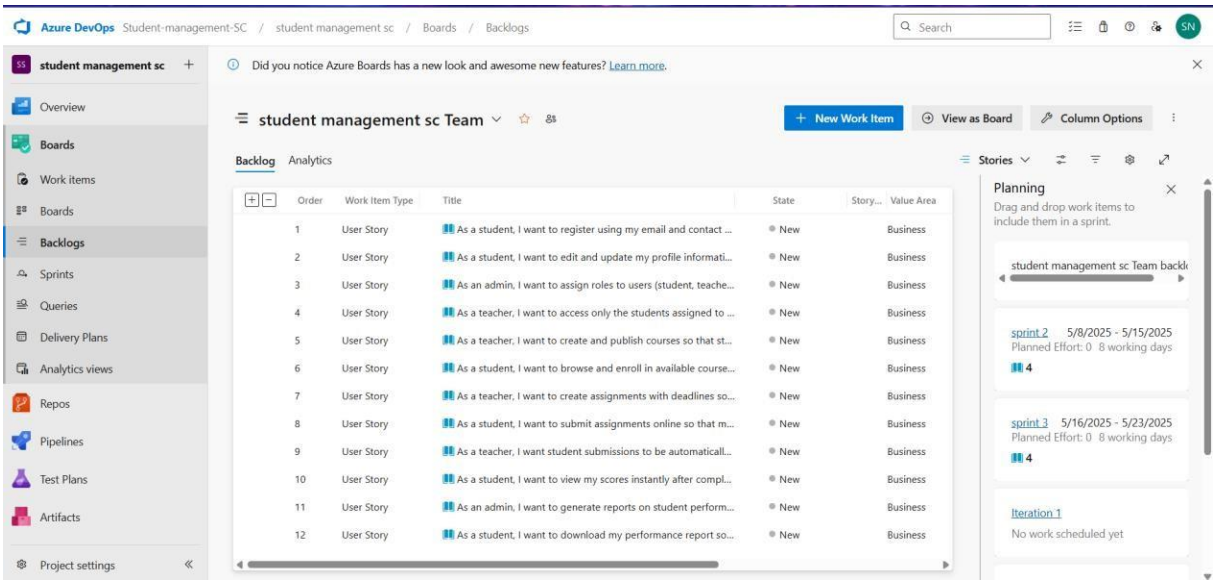


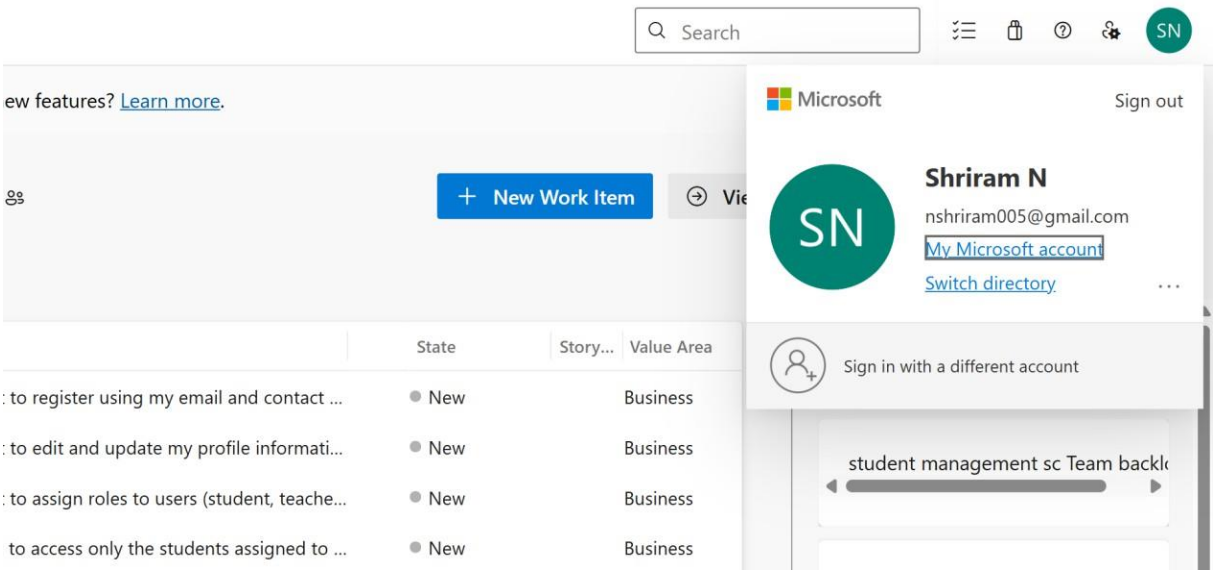
5. To manage user stories:

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:

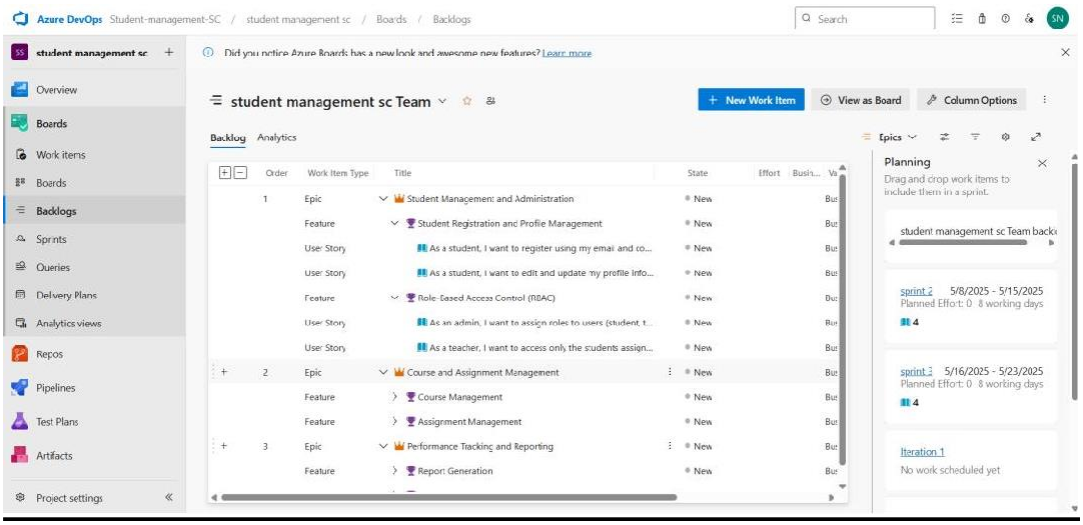
Successfully created an Azure DevOps project with user story management and agile workflow setup.

EXP NO: 3 DATE:	SETTING UP EPICS, FEATURES, AND USER STORIES FOR PROJECT PLANNING

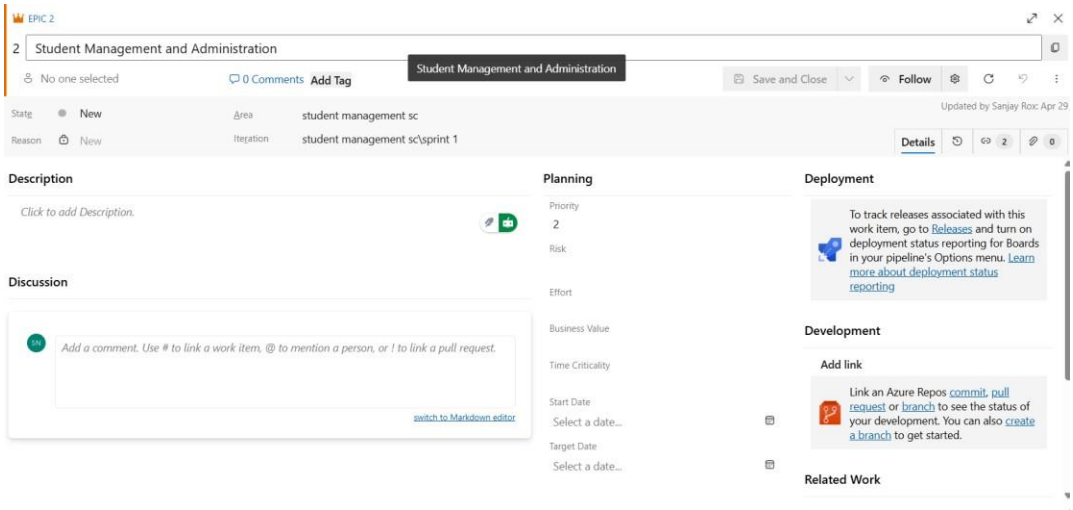
AIM:-

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

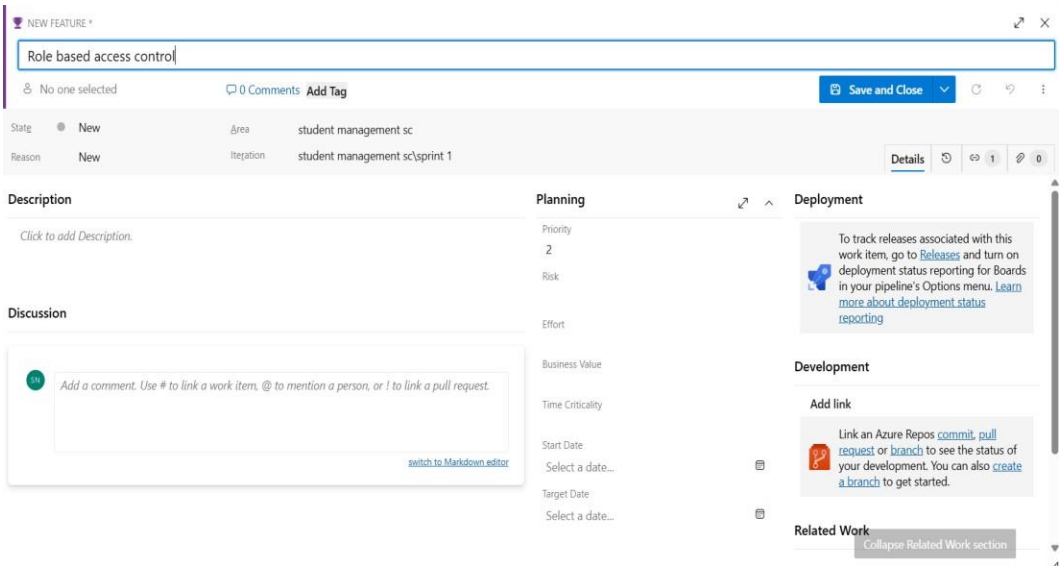
Create Epic, Features, User Stories, Task



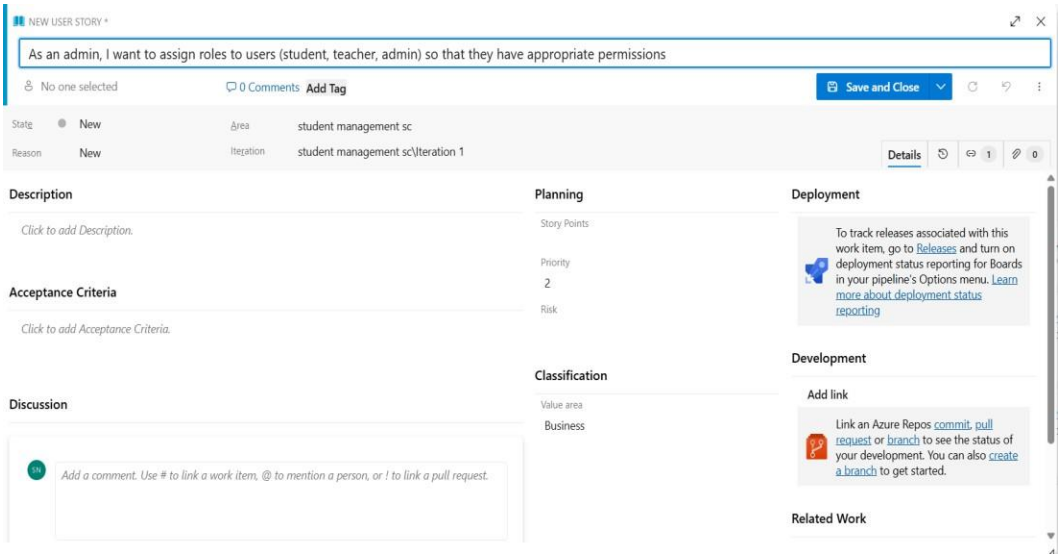
1. Fill in Epics



2. Fill in Features



3. Fill in User Story Details



RESULT:-

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

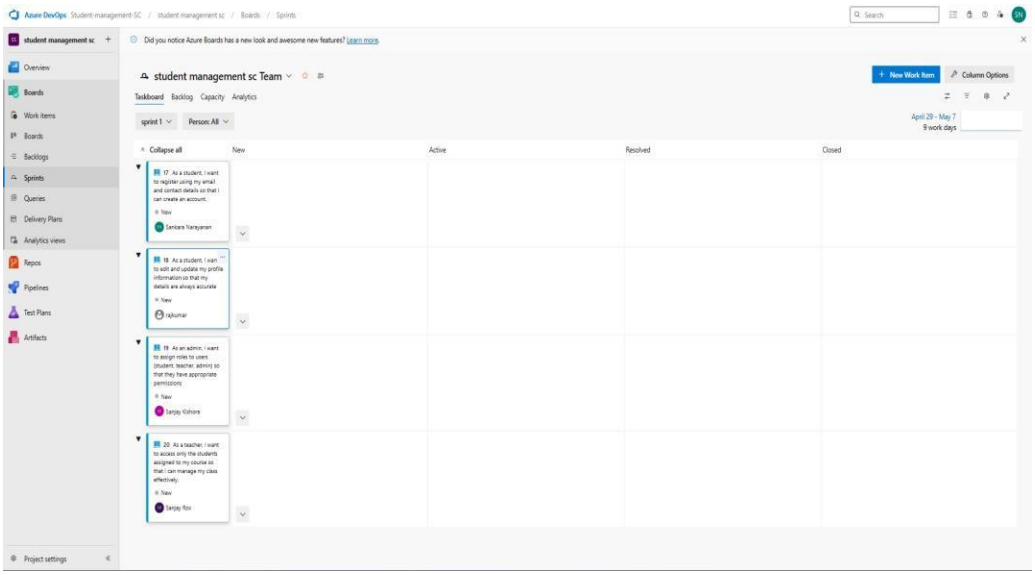
EXP NO: 4 DATE:	SPRINT PLANNING

AIM:-

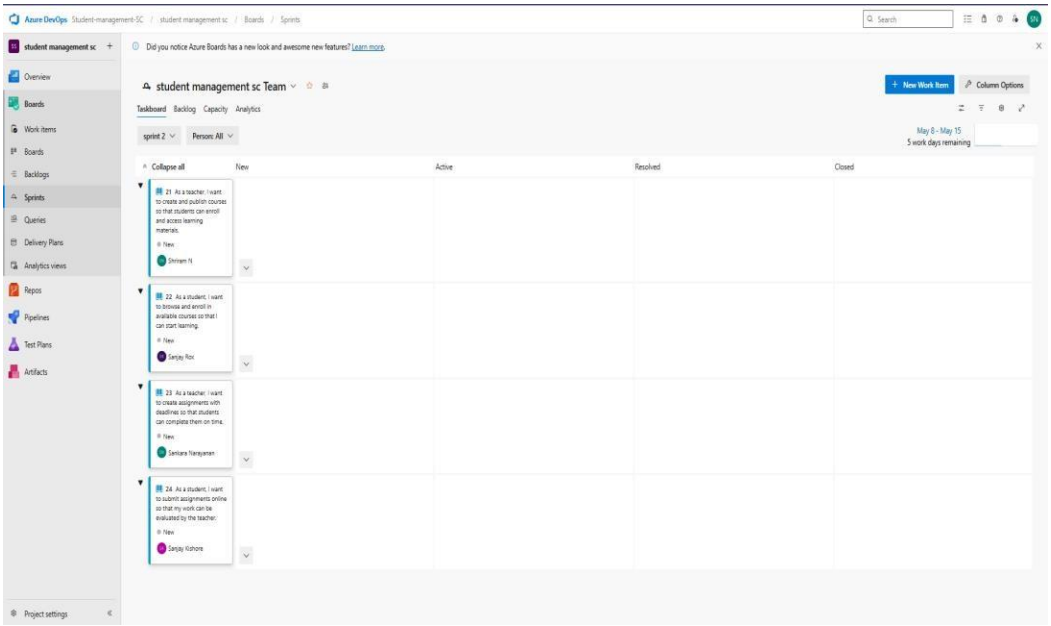
To assign user story to specific sprint for the Student Management System Project.

Sprint Planning:-

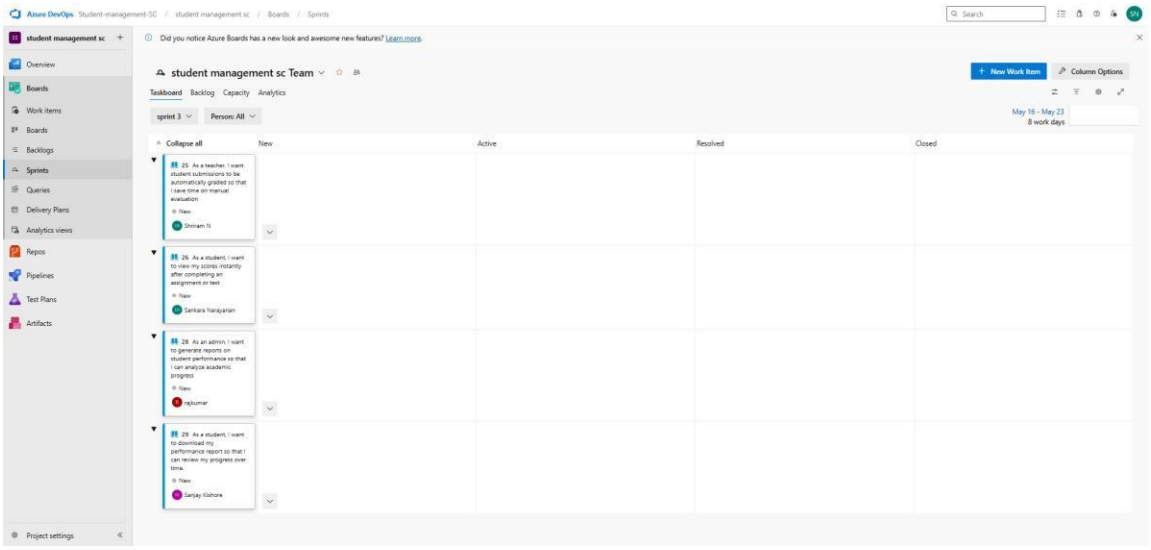
Sprint 1



Sprint 2



Sprint 3



RESULT:-

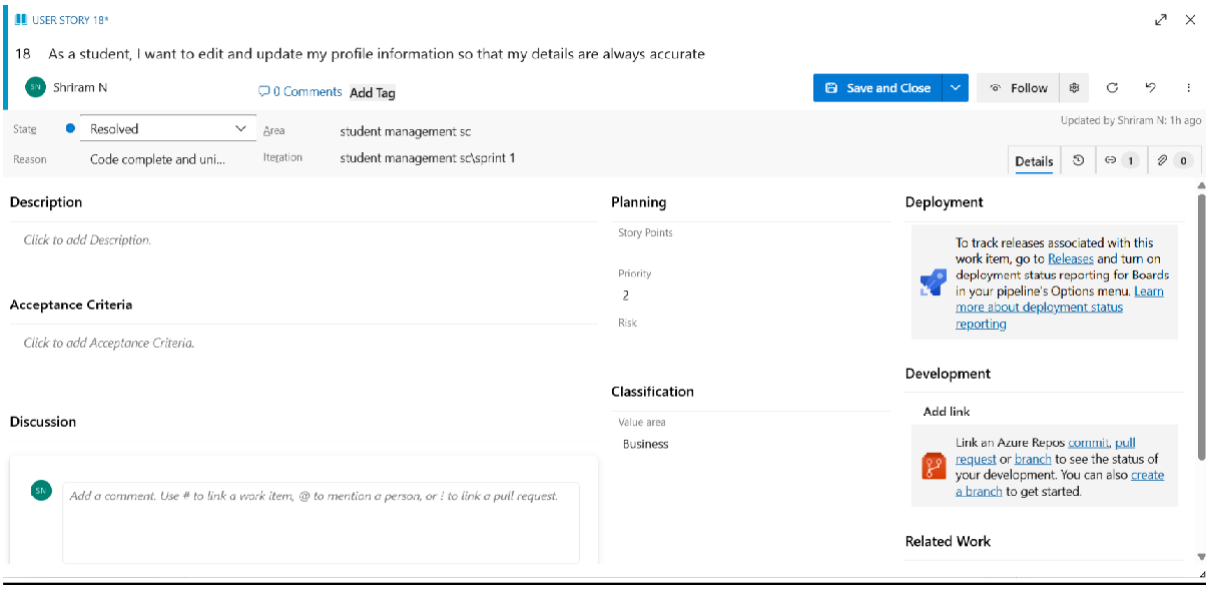
The Sprints are created for the Student Management System.

EXP NO: 5 DATE:	POKER ESTIMATION

AIM:-

Create Poker Estimation for the user stories -Student management System.

Poker Estimation:-



RESULT:-

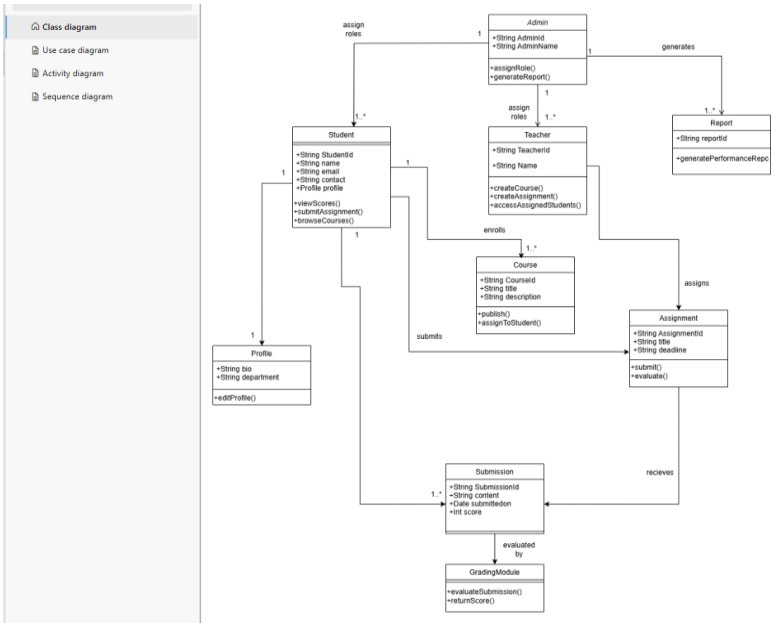
The Estimation/Story Points is created for the project using Poker Estimation.

EXP NO: 6 DATE:	DESIGNING CLASS AND SEQUENCE DIAGRAMS FOR PROJECT ARCHITECTURE

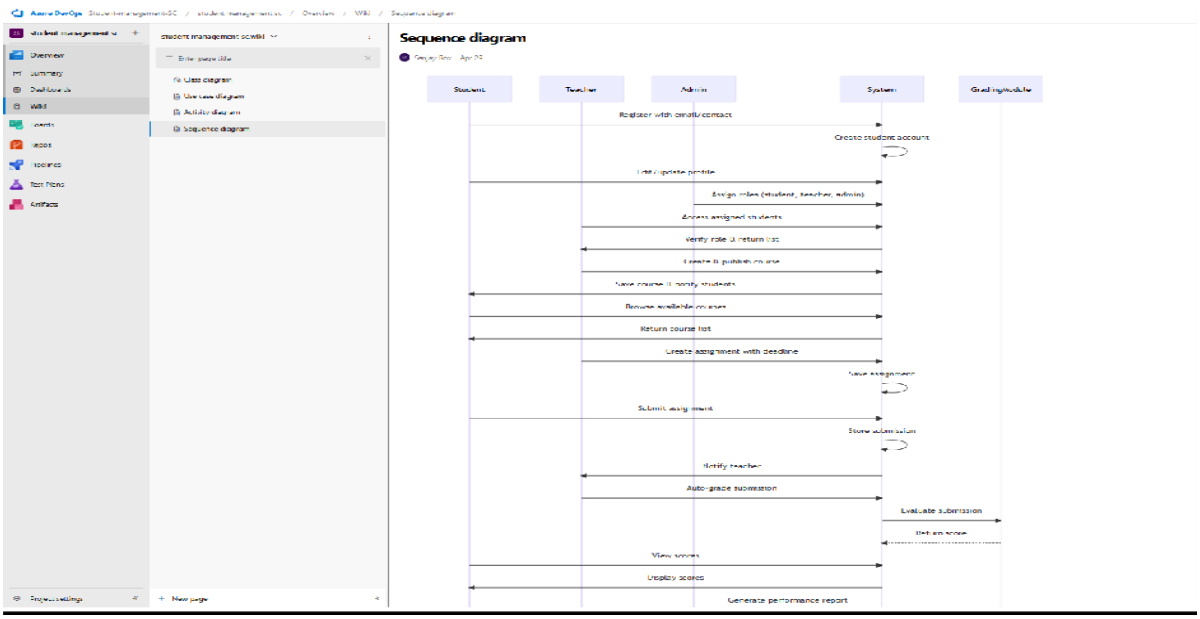
AIM:-

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

6A. CLASS DIAGRAM:-



6B. SEQUENCE DIAGRAM:-



Result:

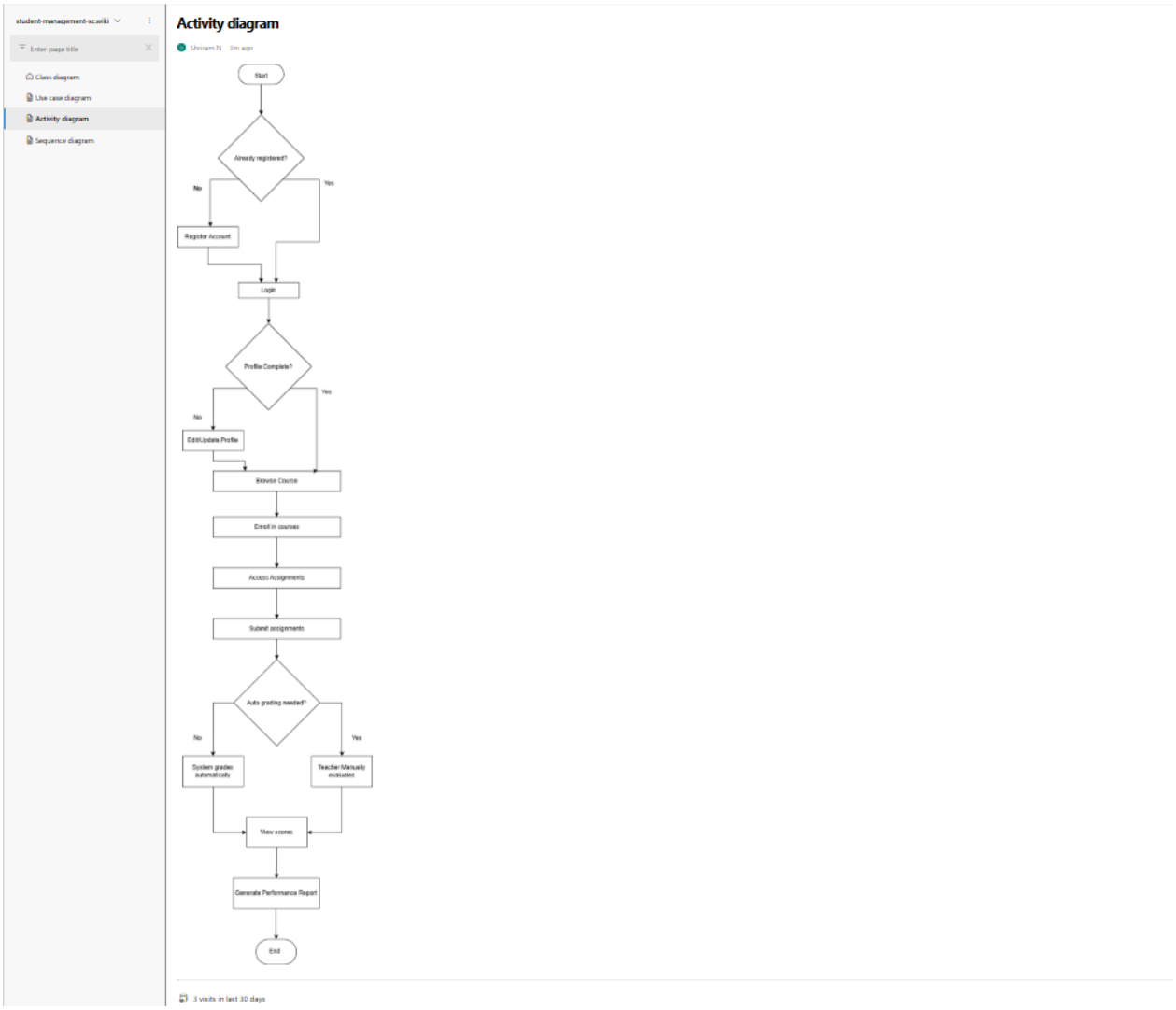
The Class Diagram and Sequence Diagram is designed Successfully for given project.

EXP NO: 7 DATE:	DESIGNING USE CASE AND ACTIVITY DIAGRAM FOR PROJECT ARCHITECTURE

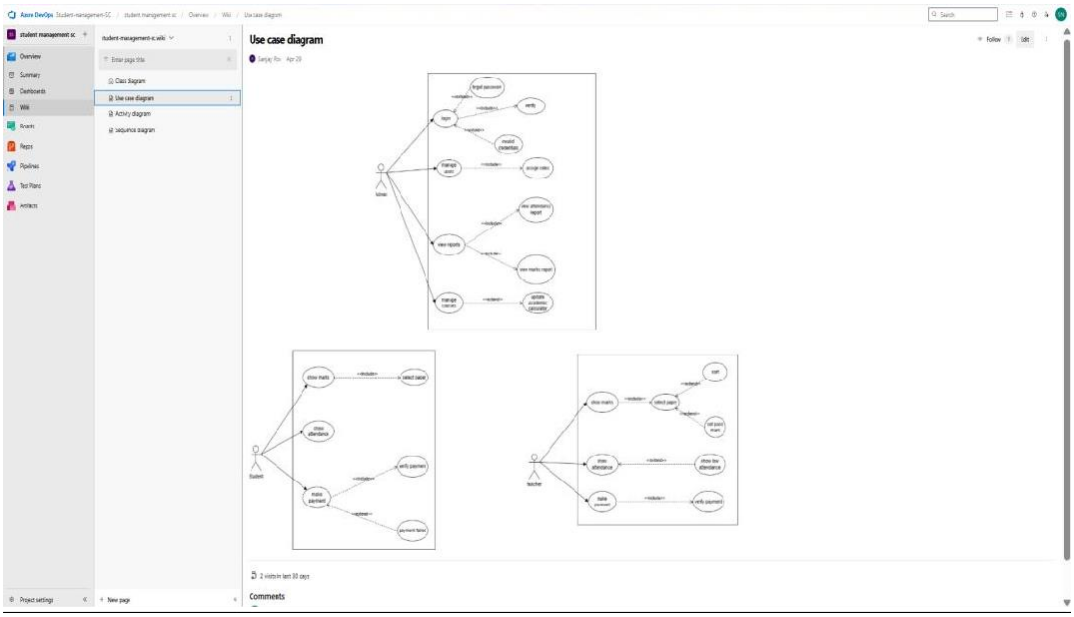
AIM:-

To Design an activity Diagram and use case Diagram for the given Project.

7A. Activity Diagram:-



7B. USE CASE DIAGRAM:-



RESULT:-

Thus activity and use case diagram has been designed successfully for Student Management System

EXP NO: 8 DATE:	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
Test Case Design Procedure

1. Understand Core Features of the Application

- User Signup & Login
- Student Enrollment in Courses
- Assignment Submission & Evaluation
- Viewing Grades and Attendance
- Admin Role Management and Report Generation

2. Define User Interactions

- Each test case simulates a real user behavior (e.g., logging in, submitting an assignment, viewing results).

3. Design Happy Path Test Cases

- Focused on validating that all core functionalities work correctly under normal conditions.
- **Example**: Student registers and logs in, enrolls in a course, submits assignment, and views grades.

4. Design Error Path Test Cases

- Simulate invalid inputs, system issues, or failed actions to ensure proper error handling.
- **Example**: Login with wrong credentials, submission without attachment, 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 identified (e.g., TC01 – Student Login Success, TC12 – Invalid Assignment Submission).
- Facilitates easy mapping to features and tracking in Azure DevOps.

7. Separate Test Suites

- Grouped by functionality such as:
    - Login and Registration
    - Course Enrollment

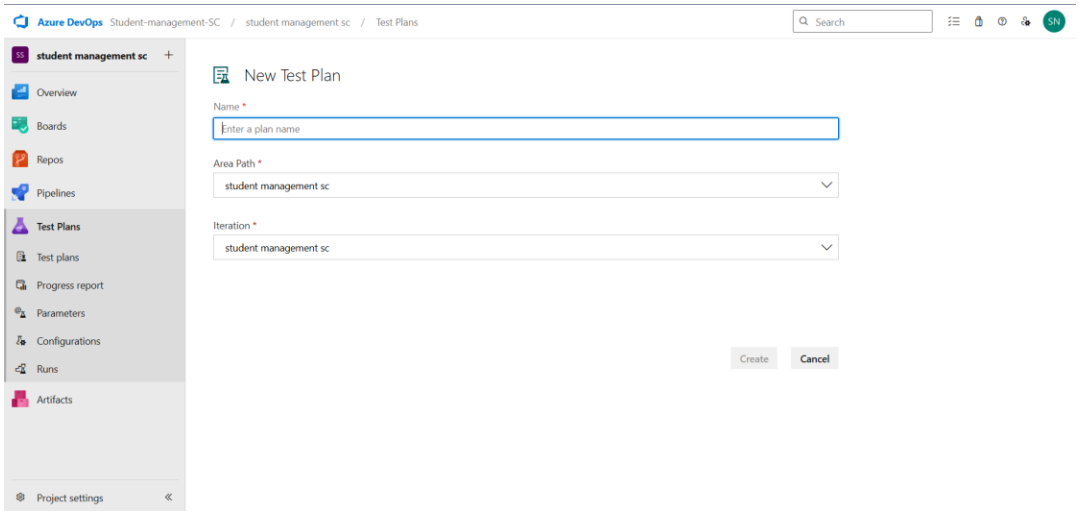
- Assignment Submission
- Report Generation
- Admin Functions

- Improves organization and enables focused execution in Azure DevOps.

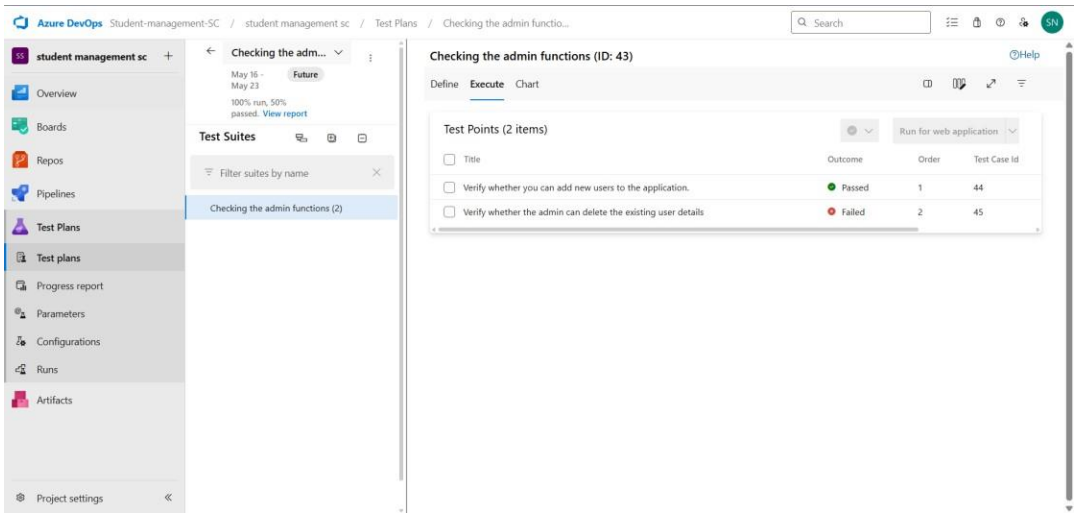
8. Prioritize and Review

- High-priority assigned to critical workflows like login, course access, and grading.
- Reviewed for completeness, accuracy, and alignment with user stories and feature definitions.

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.

Student Management System– Test Plans

USER STORIES:-

- As a student, I want to sign up and log in securely so I can access academic services. *(ID: 101)*
- As a student, I need to enroll in courses to participate in classes. *(ID: 102)*
- As a student, I should be able to submit assignments online for evaluation. *(ID: 103)*
- As a student, I want to view grades and attendance in one place. *(ID: 104)*
- As an admin, I want to manage users and generate performance reports. *(ID: 105)*

Test Suites:-

Test suit : TS01 – User Login & Registration (ID: 201)

1. **TC01 – Successful Student Sign-Up**
    - **Action:**
        - Go to the Sign-Up page
        - Enter valid name, email, and password
        - Click "Sign Up"
    - **Expected Result:**
        - Student account is created and redirected to dashboard
    - **Type:** Happy Path
2. **TC02 – Login with Valid Credentials**
    - **Action:**
        - Go to Login page
        - Enter valid student email & password
        - Click "Login"
    - **Expected Result:**
        - Logged in and taken to student dashboard
    - **Type:** Happy Path
3. **TC03 – Sign-Up with Existing Email**
    - **Action:**
        - Go to Sign-Up
        - Enter already registered email
        - Click "Sign Up"
    - **Expected Result:**
        - Error message: "Email already registered"
    - **Type:** Error Path

4. **TC04 – Login with Incorrect Password**
    - **Action:**
        - Enter wrong password
        - Click "Login"
    - **Expected Result:**
        - Error: "Invalid username or password"
    - **Type:** Error Path

Test Suite: TS02 – Course Enrollment (ID: 202)

1. **TC05 – Enroll in Available Course**
    - **Action:**
        - Login → Navigate to Course List
        - Select a course → Click "Enroll"
    - **Expected Result:**
        - Enrollment confirmation shown
    - **Type:** Happy Path

2. **TC06 – Try Enrolling in a Full Course**
    - **Action:**
        - Select a course with no seats available
        - Click "Enroll"
    - **Expected Result:**
        - Error: "Enrollment failed – course full
    - **Type:** Error Path

Test Suite: TS03 – Assignment Submission (ID: 203)

1. **TC07 – Submit Assignment Successfully**
    - **Action:**
        - Login → Go to Assignments
        - Upload file and click "Submit"
    - **Expected Result:**
        - Submission confirmation message shown
    - **Type:** Happy Path

2. **TC08 – Submit Without Uploading File**
    - **Action:**
        - Open assignment submission
        - Click "Submit" without file
    - **Expected Result:**
        - Error: "Please upload a file before submitting"
    - **Type:** Error Path

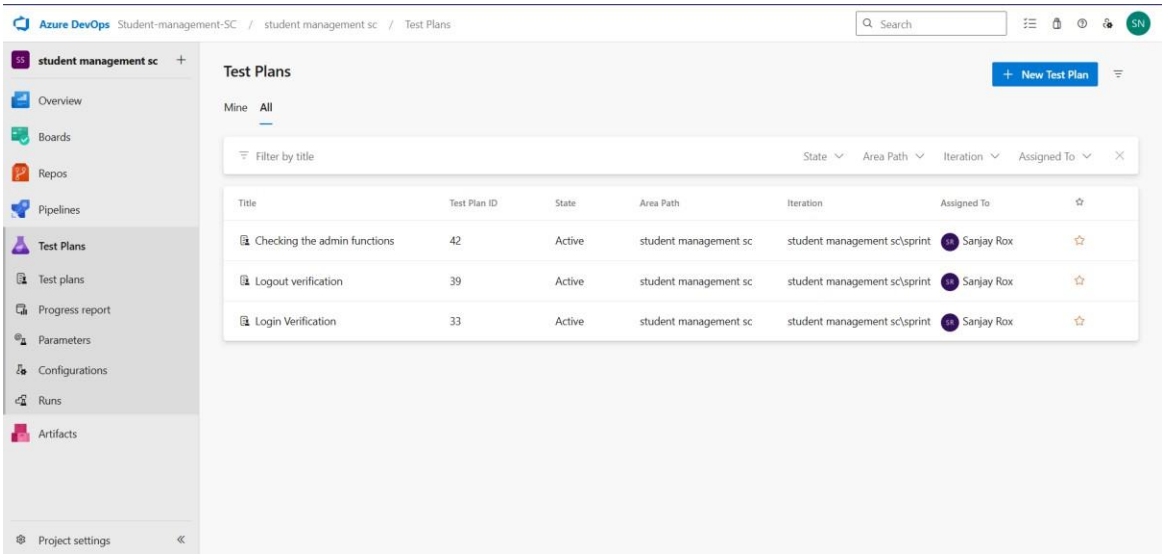
Test Suite: TS04 – View Grades & Attendance (ID: 204)

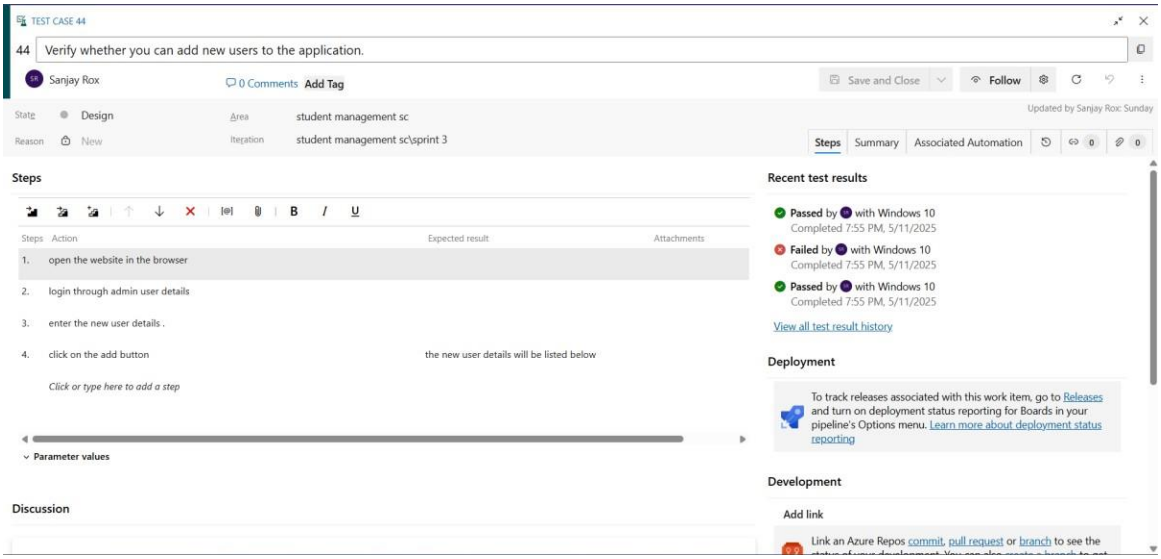
1. **TC09 – View Grades Dashboard**
    - **Action:**
        - Login → Go to "My Grades"
    - **Expected Result:**
        - All grades are displayed by subject
    - **Type:** Happy Path
2. **TC10 – Fail to Fetch Grades (Offline Scenario)**
    - **Action:**
        - Disconnect network → Go to "My Grades"
    - **Expected Result:**
        - Error: "Unable to fetch data – check connection"
    - **Type:** Error Path

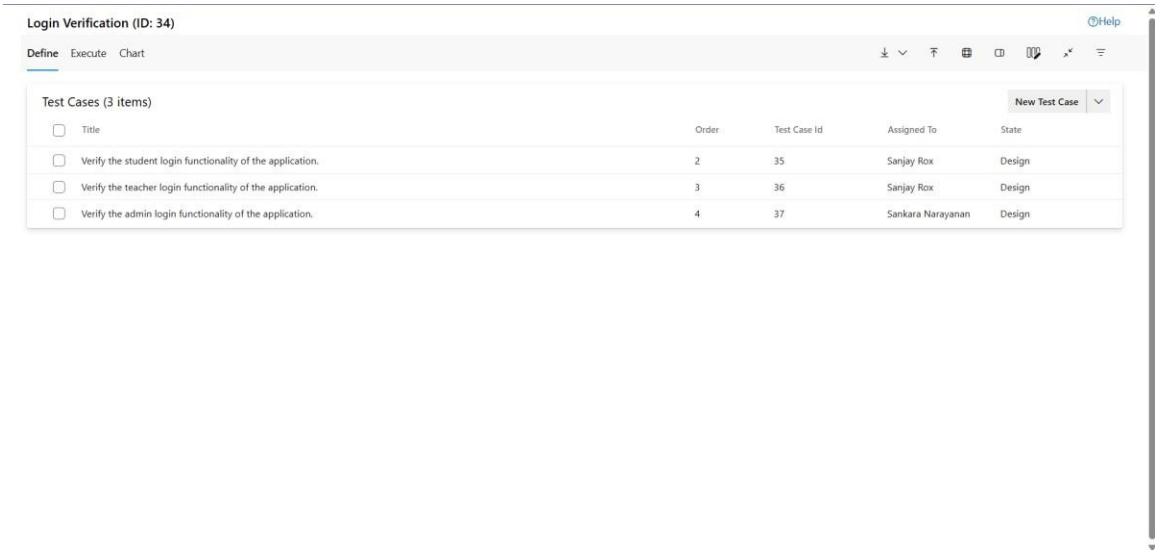
Test Suite: TS05 – Admin Management (ID: 205)

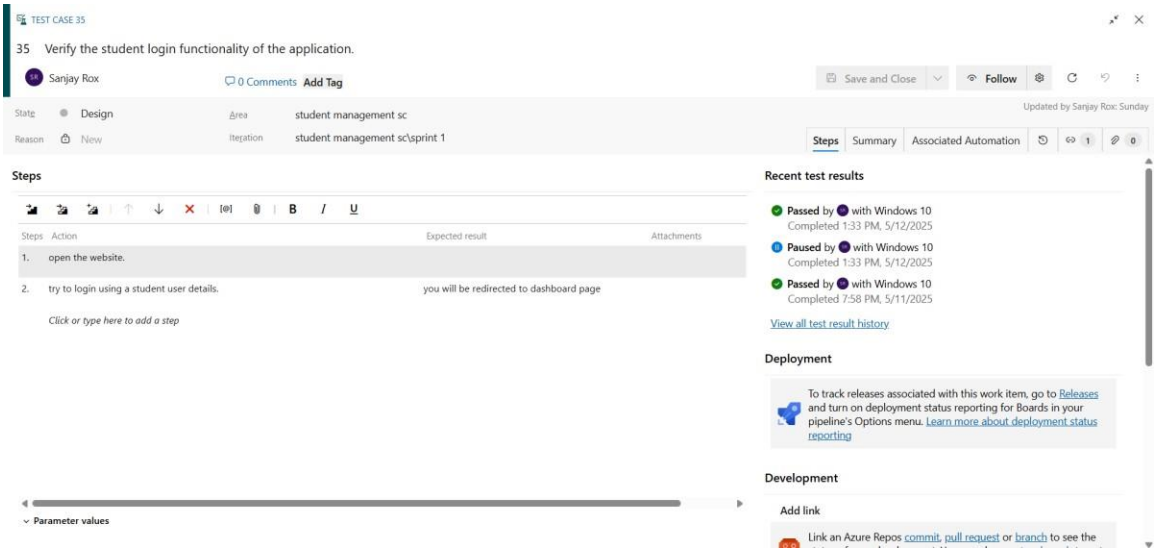
1. **TC11 – Assign Role to New User**
    - **Action:**
        - Login as Admin → Navigate to User Management
        - Select user and assign role
    - **Expected Result:**
        - Confirmation: "Role assigned successfully"
    - **Type:** Happy Path
2. **TC12 – Generate Report with No Data**
    - **Action:**
        - Navigate to Reports → Choose empty semester
        - Click "Generate"
    - **Expected Result:**
        - Message: "No data available to generate report"
    - **Type:** Error Path

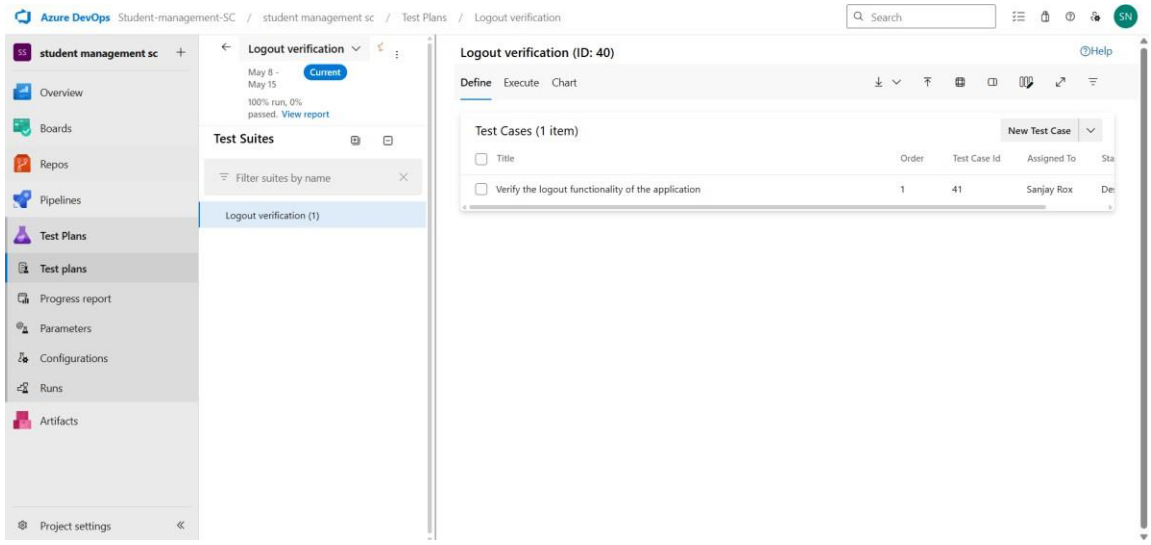
3. Test Cases

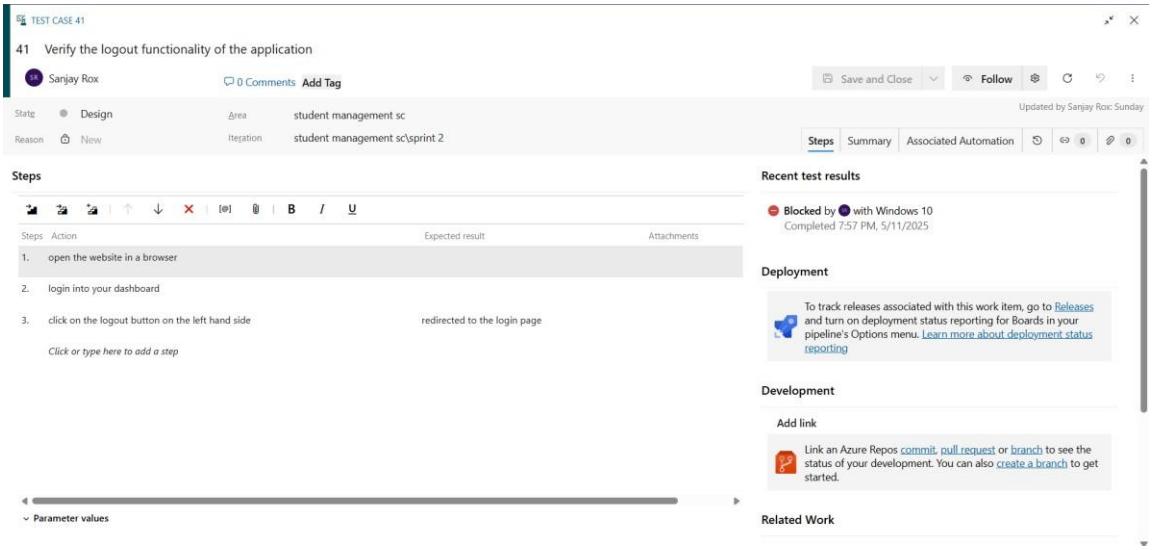

Azure DevOps Student-management-SC / student management sc / Test Plans
Search
student management sc
Overview
Boards
Repos
Pipelines
Test Plans
Test plans
Progress report
Parameters
Configurations
Runs
Artifacts
Project settings
Test Plans
New Test Plan
Mine All
Filter by title
State Area Path Iteration Assigned To
Title Test Plan ID State Area Path Iteration Assigned To
Checking the admin functions 42 Active student management sc student management sc\sprint Sanjay Rox
Logout verification 39 Active student management sc student management sc\sprint Sanjay Rox
Login Verification 33 Active student management sc student management sc\sprint Sanjay Rox


TEST CASE 44
44 Verify whether you can add new users to the application.
Sanjay Rox
0 Comments Add Tag
Save and Close Follow
State Design
Reason New
Area student management sc
Iteration student management sc\sprint 3
Updated by Sanjay Rox: Sunday
Steps Summary Associated Automation
Steps
Steps Action Expected result Attachments
1. open the website in the browser
2. login through admin user details
3. enter the new user details .
4. click on the add button the new user details will be listed below
Click or type here to add a step
Parameter values
Discussion
Recent test results
Passed by with Windows 10
Completed 7:55 PM, 5/11/2025
Failed by with Windows 10
Completed 7:55 PM, 5/11/2025
Passed by with Windows 10
Completed 7:55 PM, 5/11/2025
View all test result history
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

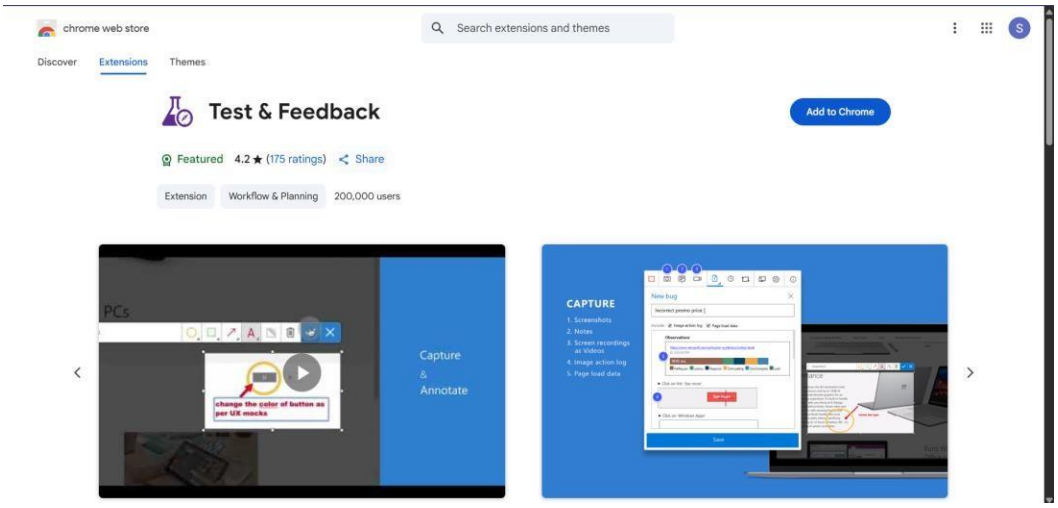

Login Verification (ID: 34)
Help
Define Execute Chart
Test Cases (3 items)
New Test Case
Title Order Test Case Id Assigned To State
Verify the student login functionality of the application. 2 35 Sanjay Rox Design
Verify the teacher login functionality of the application. 3 36 Sanjay Rox Design
Verify the admin login functionality of the application. 4 37 Sankara Narayanan Design

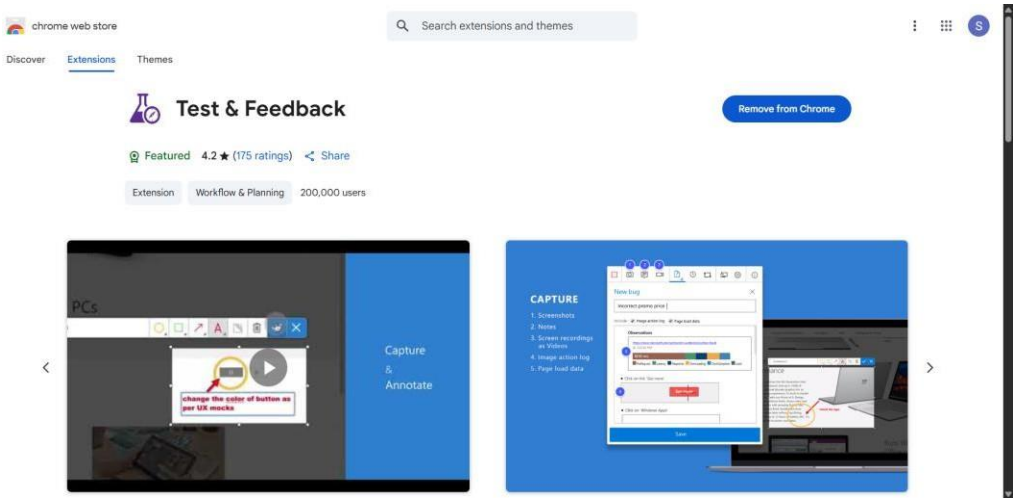

TEST CASE 35
35 Verify the student login functionality of the application.
Sanjay Rox
0 Comments Add Tag
Save and Close
Follow
State Design
Reason New
Area student management sc
Iteration student management sc\sprint 1
Updated by Sanjay Rox: Sunday
Steps Summary Associated Automation
Steps
Steps Action Expected result Attachments
1. open the website.
2. try to login using a student user details. you will be redirected to dashboard page
Click or type here to add a step
Parameter values
Recent test results
Passed by with Windows 10
Completed 1:33 PM, 5/12/2025
Paused by with Windows 10
Completed 1:33 PM, 5/12/2025
Passed by with Windows 10
Completed 7:58 PM, 5/11/2025
View all test result history
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


Azure DevOps Student-management-SC / student management sc / Test Plans / Logout verification
Search
SN
student management sc
Overview
Boards
Repos
Pipelines
Test Plans
Test plans
Progress report
Parameters
Configurations
Runs
Artifacts
Project settings
Logout verification
May 8 - May 15
Current
100% run, 0% passed. View report
Test Suites
Filter suites by name
Logout verification (1)
Logout verification (ID: 40)
Help
Define Execute Chart
Test Cases (1 item)
New Test Case
Title Order Test Case Id Assigned To
Verify the logout functionality of the application 1 41 Sanjay Rox

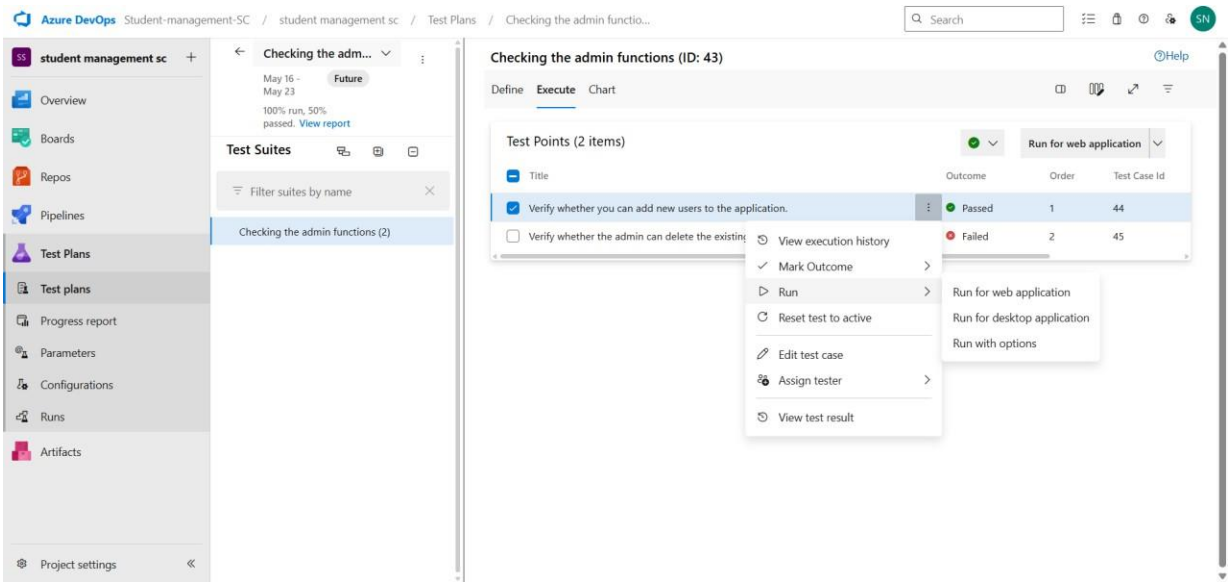

TEST CASE 41
41 Verify the logout functionality of the application
Sanjay Rox
0 Comments Add Tag
Save and Close
Follow
State Design
Reason New
Area student management sc
Iteration student management sc\sprint 2
Updated by Sanjay Rox: Sunday
Steps Summary Associated Automation
Steps
Steps Action Expected result Attachments
1. open the website in a browser
2. login into your dashboard
3. click on the logout button on the left hand side redirected to the login page
Click or type here to add a step
Parameter values
Recent test results
Blocked by with Windows 10
Completed 7:57 PM, 5/11/2025
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

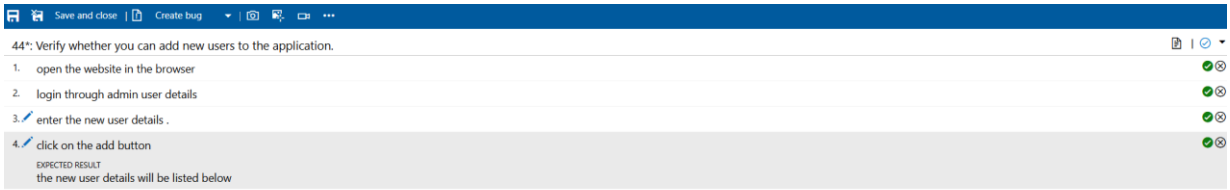
4. Installation of test



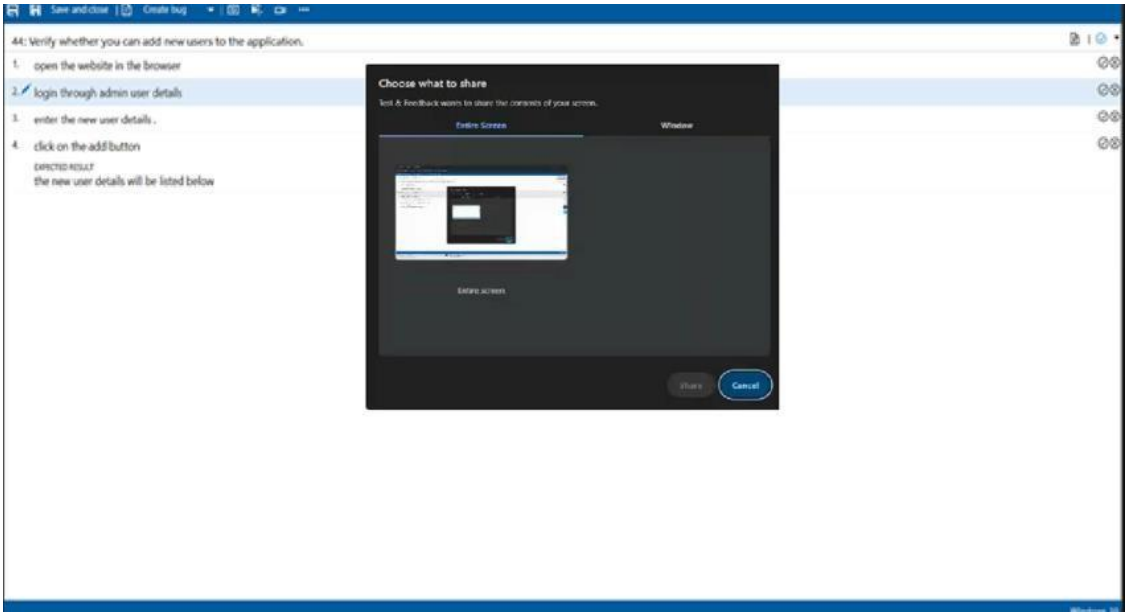


5. Running the test cases

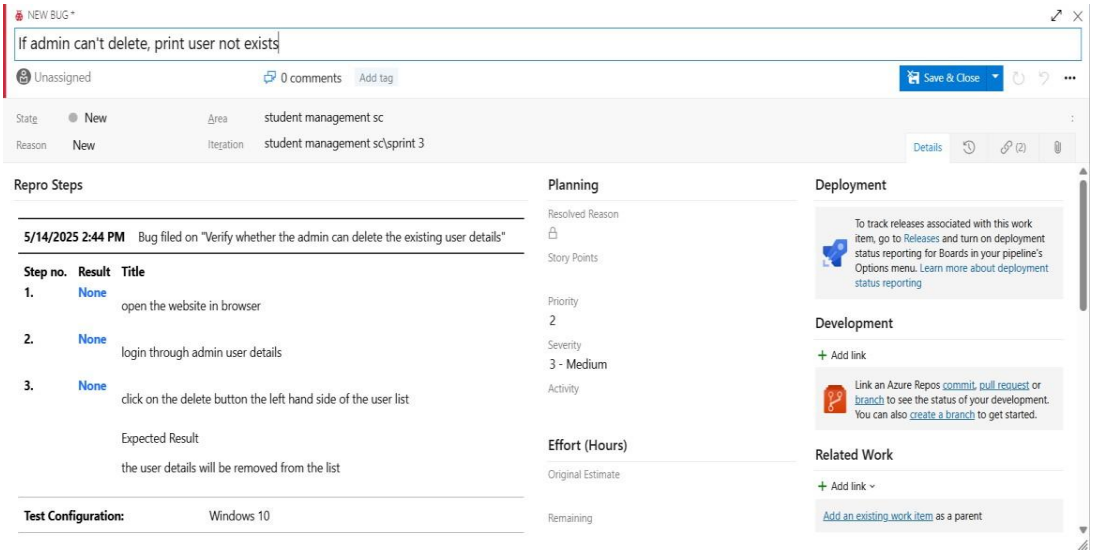


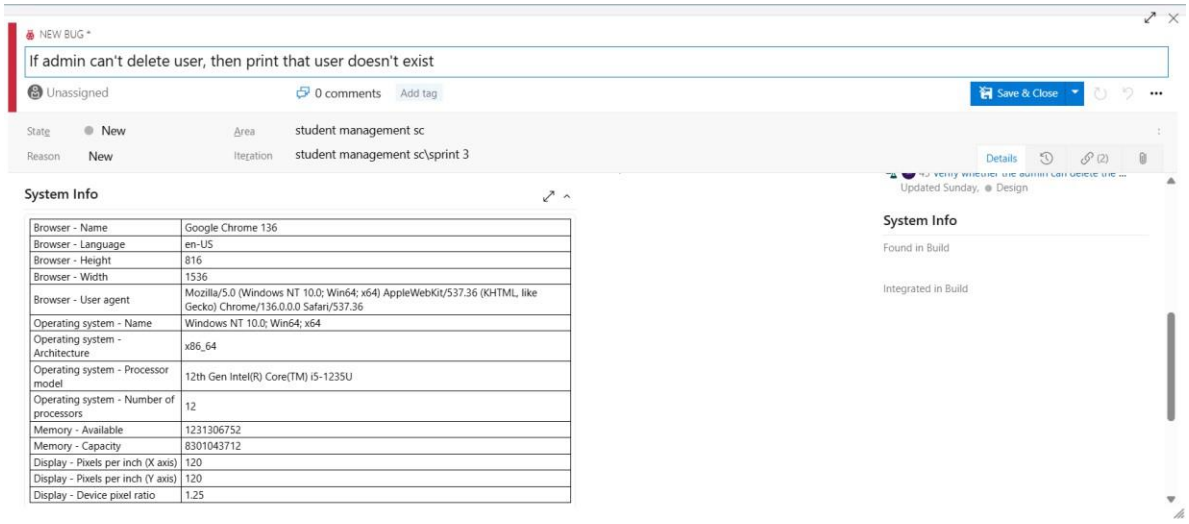


6. Recording the test case



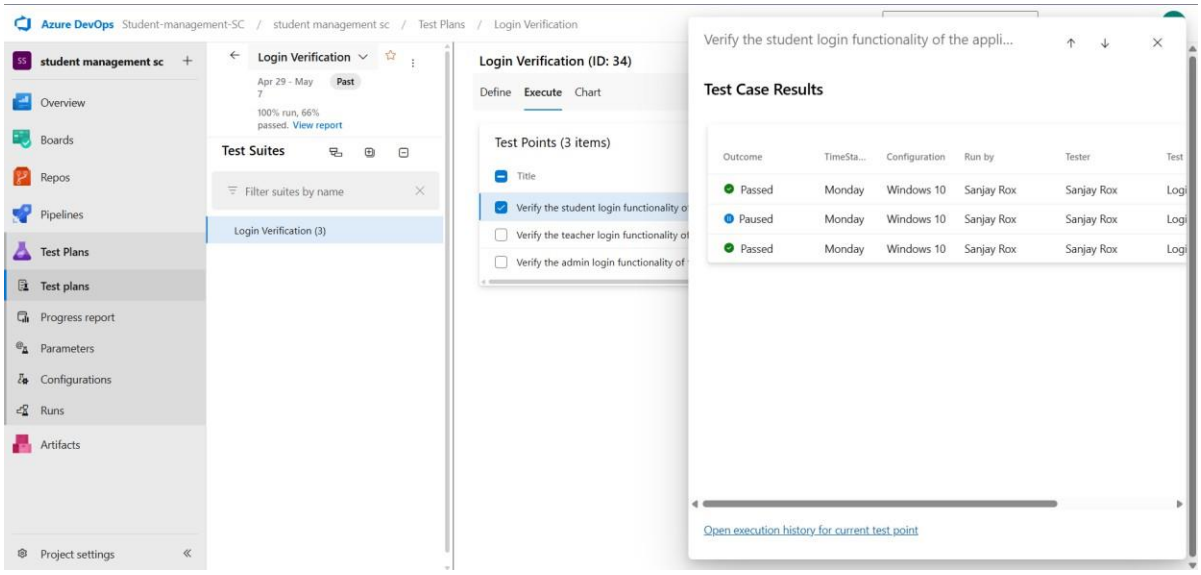
7. Creating the bug

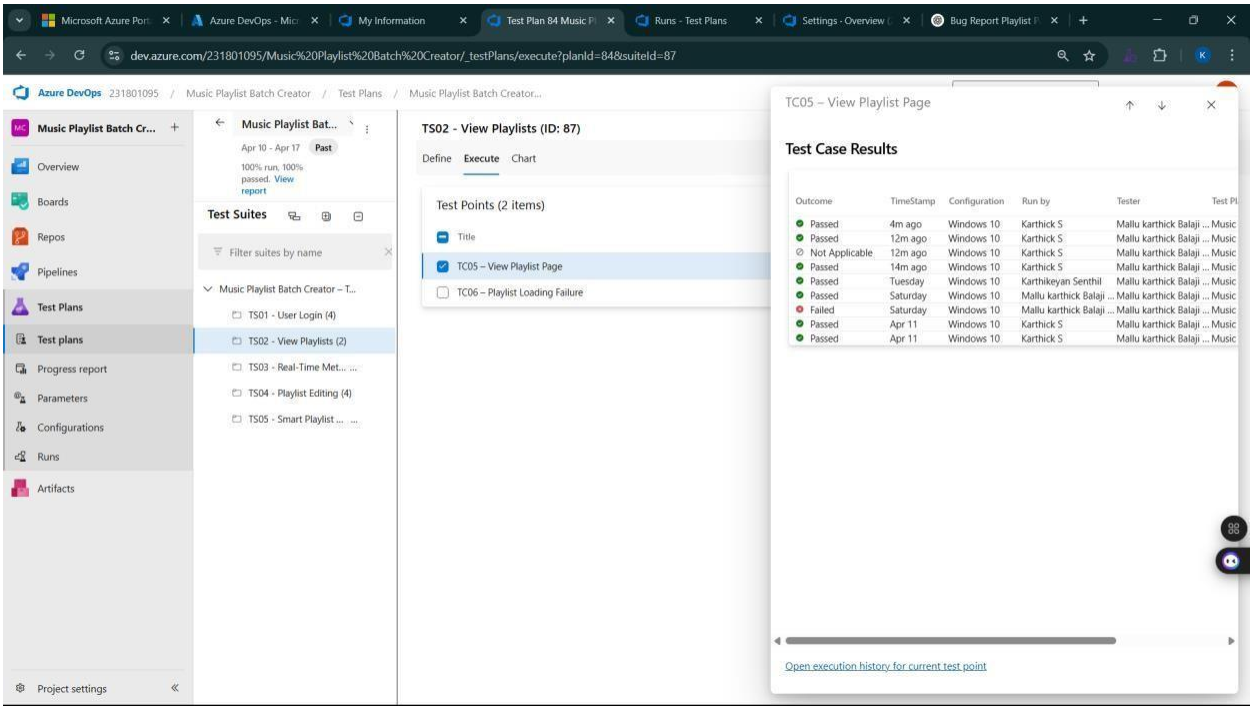




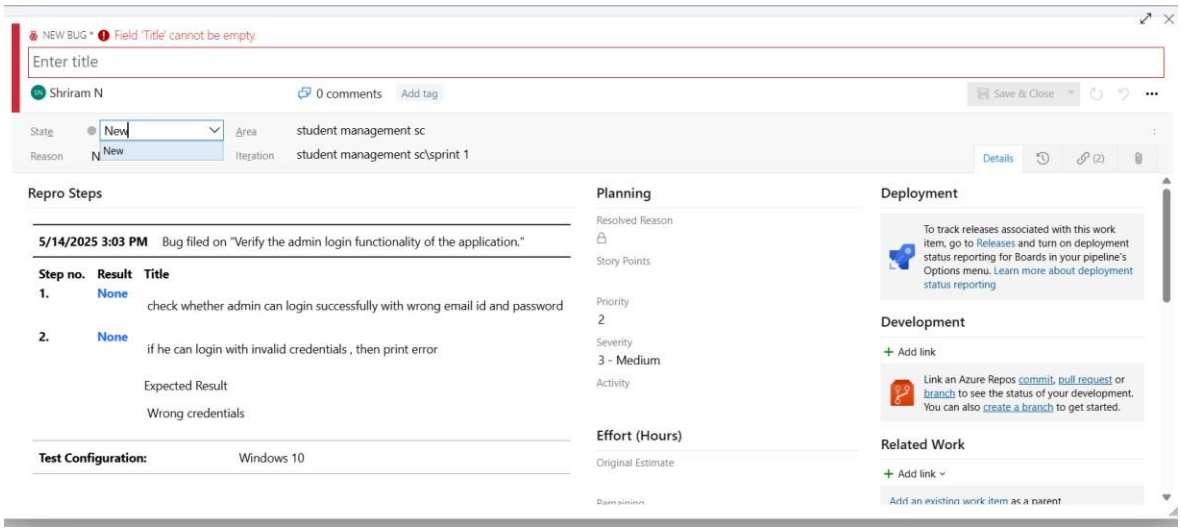
Browser - Name	Google Chrome 136
Browser - Language	en-US
Browser - Height	816
Browser - Width	1536
Browser - User agent	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/136.0.0.0 Safari/537.36
Operating system - Name	Windows NT 10.0; Win64; x64
Operating system - Architecture	x86_64
Operating system - Processor model	12th Gen Intel(R) Core(TM) i5-1235U
Operating system - Number of processors	12
Memory - Available	1231306752
Memory - Capacity	8301043712
Display - Pixels per inch (X axis)	120
Display - Pixels per inch (Y axis)	120
Display - Device pixel ratio	1.25

Test Results



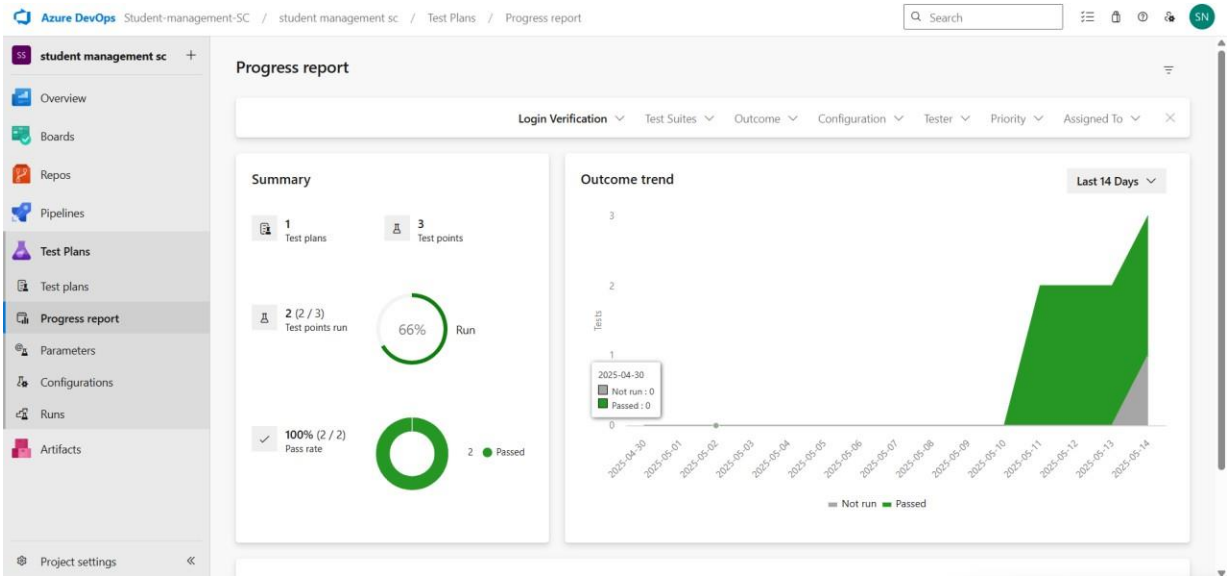


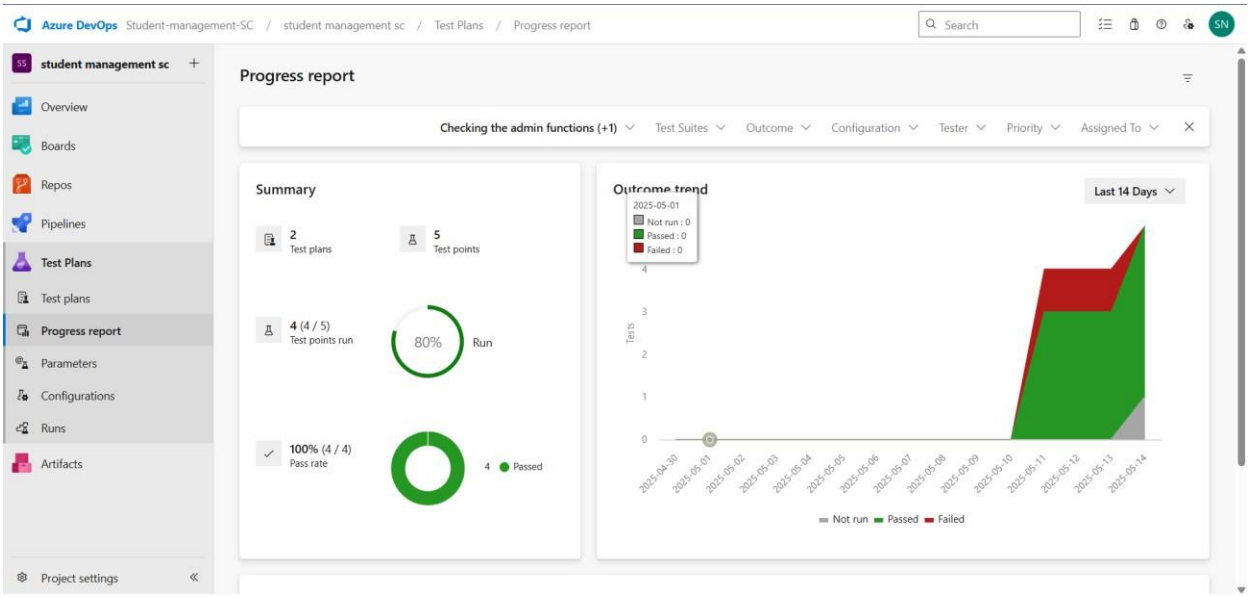
8. Test report summary

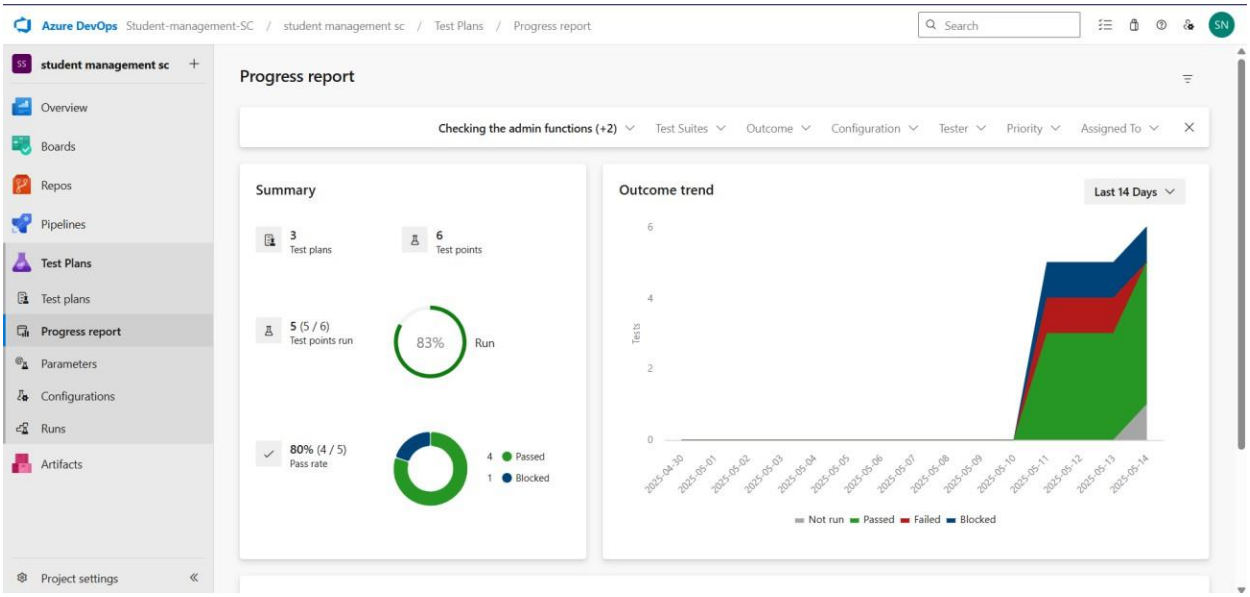


1. Assigning bug to the developer and changing state

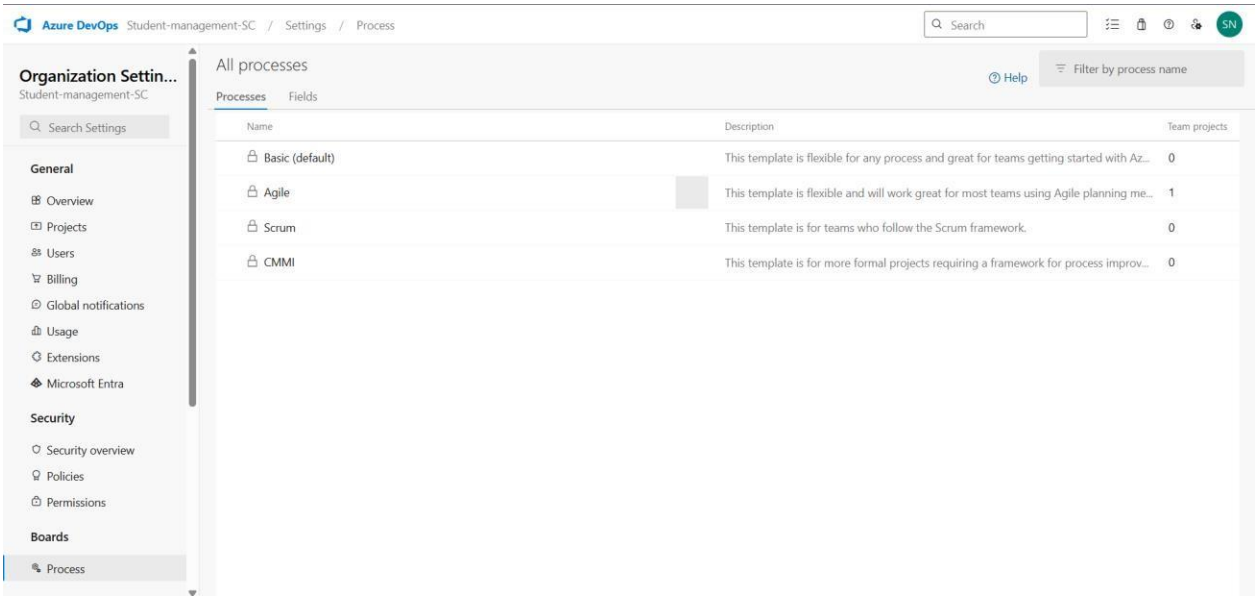
9. Progress report







10. Changing the test template



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 DATE:	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 Student 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.

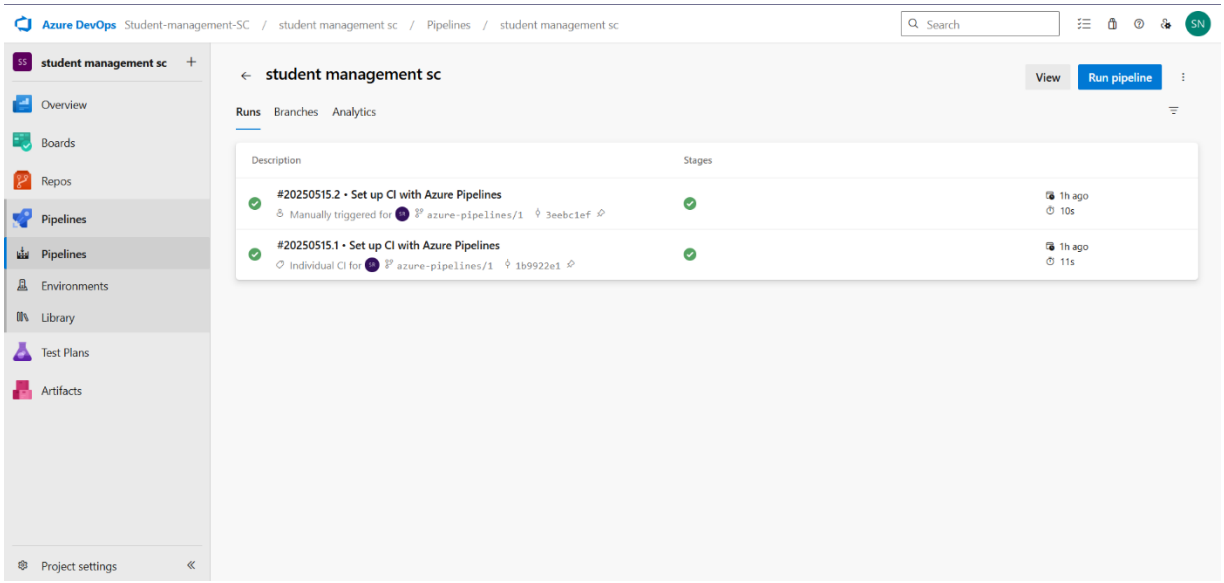
11. Configure Continuous Deployment (CD)
    Navigate to the Releases tab under Pipelines and click “New Release Pipeline”.
    Add an Artifact (from the build stage) and create a new Stage (e.g., Development, Production).

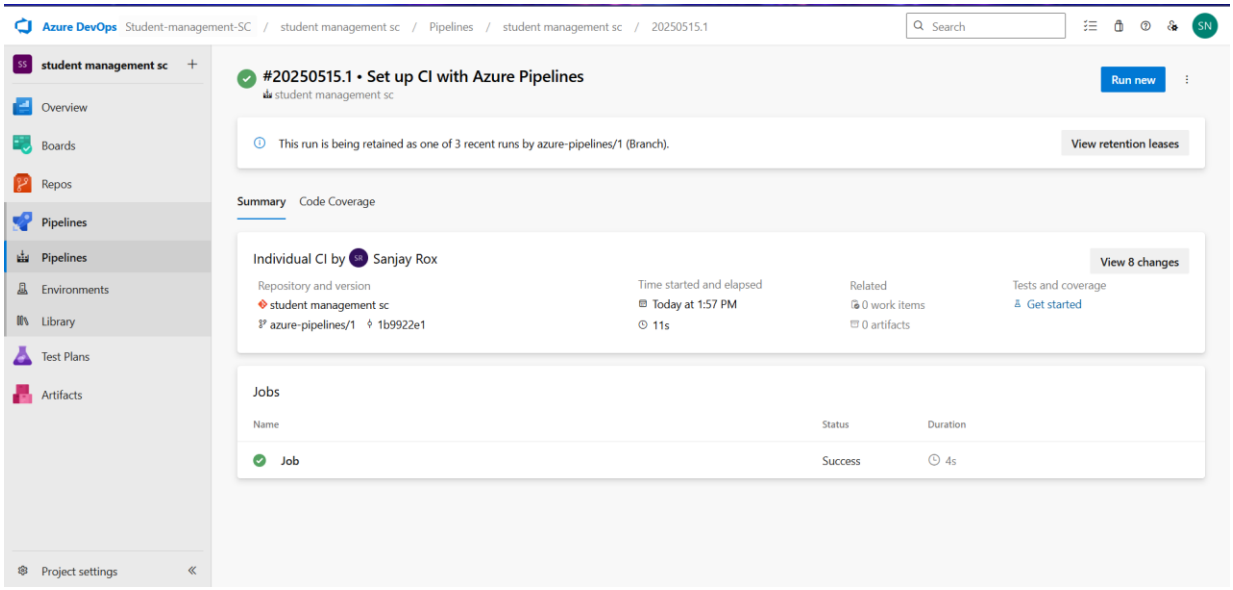
12. Configure the CD stage with deployment tasks such as deploying to Azure App Service, running database migrations or scripts, and restarting services using the Azure App Service Deploy task linked to your subscription and app details.

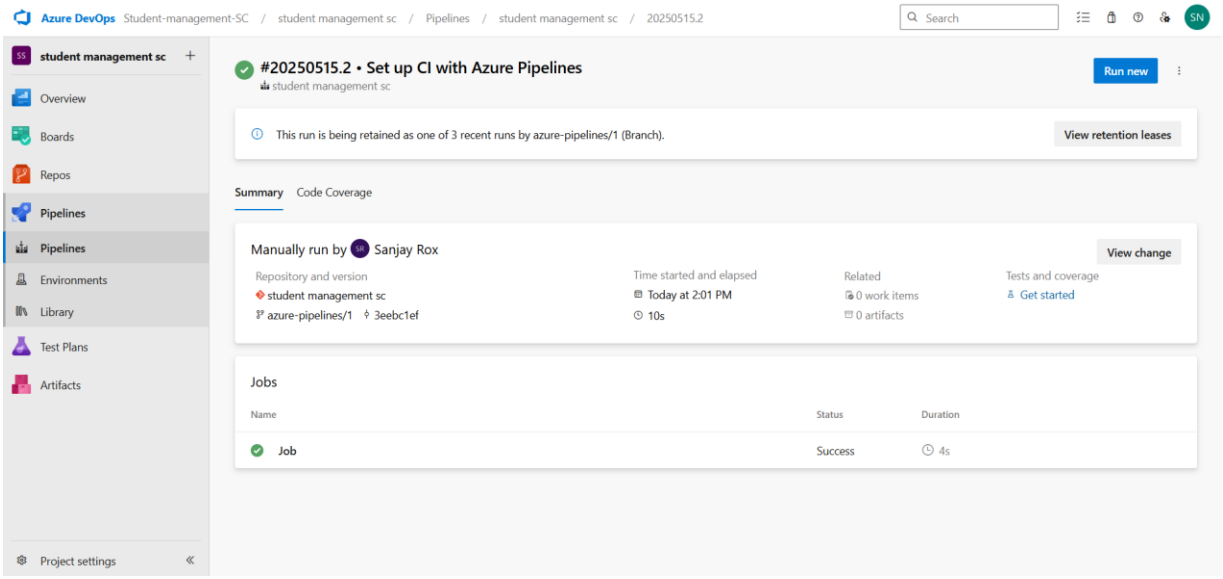
13. Set Triggers and Approvals
    Enable continuous deployment trigger so the release pipeline runs automatically after a successful build.
    For production environments, configure pre-deployment approvals to ensure manual verification before release.

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
Student-management-SC / student management sc / Pipelines / student management sc
Search
SN
student management sc
Overview
Boards
Repos
Pipelines
Pipelines
Environments
Library
Test Plans
Artifacts
Project settings
student management sc
View
Run pipeline
Runs
Branches
Analytics
Description
Stages
#20250515.2 • Set up CI with Azure Pipelines
Manually triggered for
azure-pipelines/1
3eebc1ef
1h ago
10s
#20250515.1 • Set up CI with Azure Pipelines
Individual CI for
azure-pipelines/1
1b9922e1
1h ago
11s


Azure DevOps
Student-management-SC / student management sc / Pipelines / student management sc / 20250515.1
Search
SN
student management sc
Overview
Boards
Repos
Pipelines
Pipelines
Environments
Library
Test Plans
Artifacts
Project settings
#20250515.1 • Set up CI with Azure Pipelines
student management sc
Run new
This run is being retained as one of 3 recent runs by azure-pipelines/1 (Branch).
View retention leases
Summary
Code Coverage
Individual CI by Sanjay Rox
View 8 changes
Repository and version
student management sc
azure-pipelines/1
1b9922e1
Time started and elapsed
Today at 1:57 PM
11s
Related
0 work items
0 artifacts
Tests and coverage
Get started
Jobs
Name
Status
Duration
Job
Success
4s


Azure DevOps Student-management-SC / student management sc / Pipelines / student management sc / 20250515.2
Search
SN
student management sc
Overview
Boards
Repos
Pipelines
Pipelines
Environments
Library
Test Plans
Artifacts
Project settings
#20250515.2 • Set up CI with Azure Pipelines
student management sc
Run new
This run is being retained as one of 3 recent runs by azure-pipelines/1 (Branch).
View retention leases
Summary
Code Coverage
Manually run by Sanjay Rox
View change
Repository and version
student management sc
azure-pipelines/1 3eebc1ef
Time started and elapsed
Today at 2:01 PM
10s
Related
0 work items
0 artifacts
Tests and coverage
Get started
Jobs
Name
Status
Duration
Job
Success
4s

RESULT:

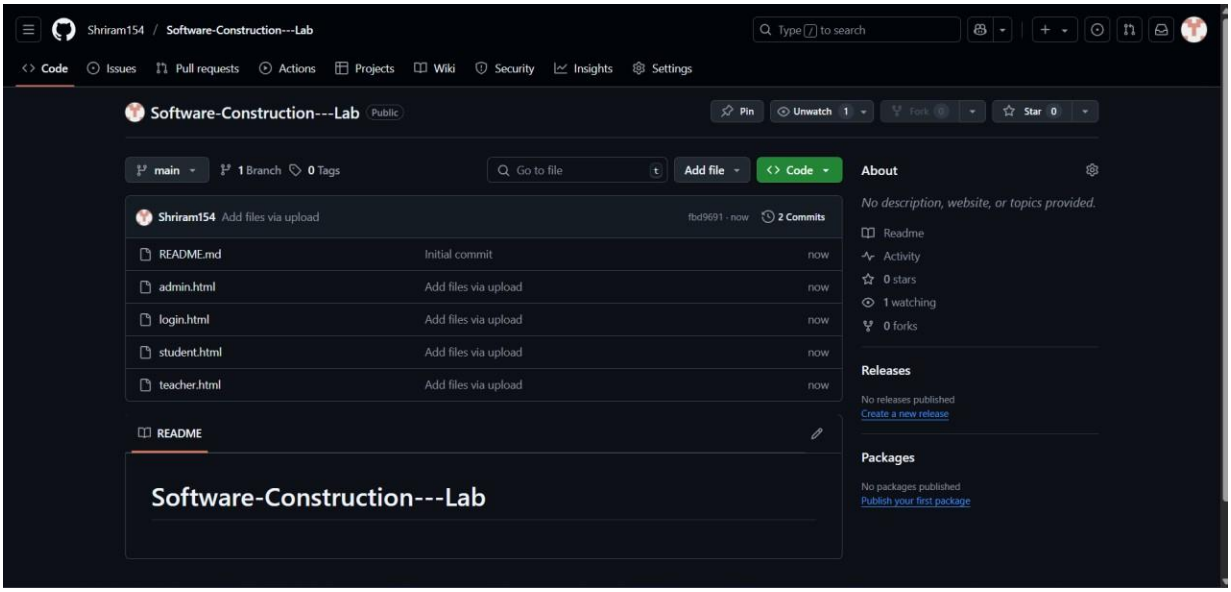
Thus the pipelines for the given project "Student Management System has been executed successfully

EXP NO: 10 DATE:	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 Student Management System project.

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.