INDEX

EXP NO: 1 Date :	AZURE DEVOPS ENVIRONMENT SETUP

Aim:

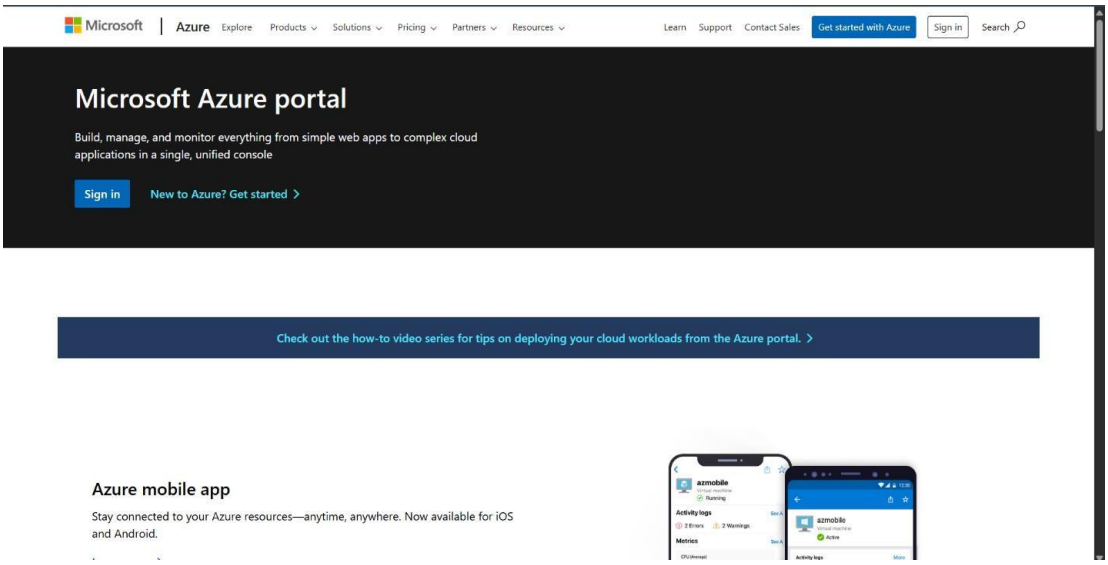
To set up and access the Azure DevOps environment by creating an organization through the Azure portal.

INSTALLATION

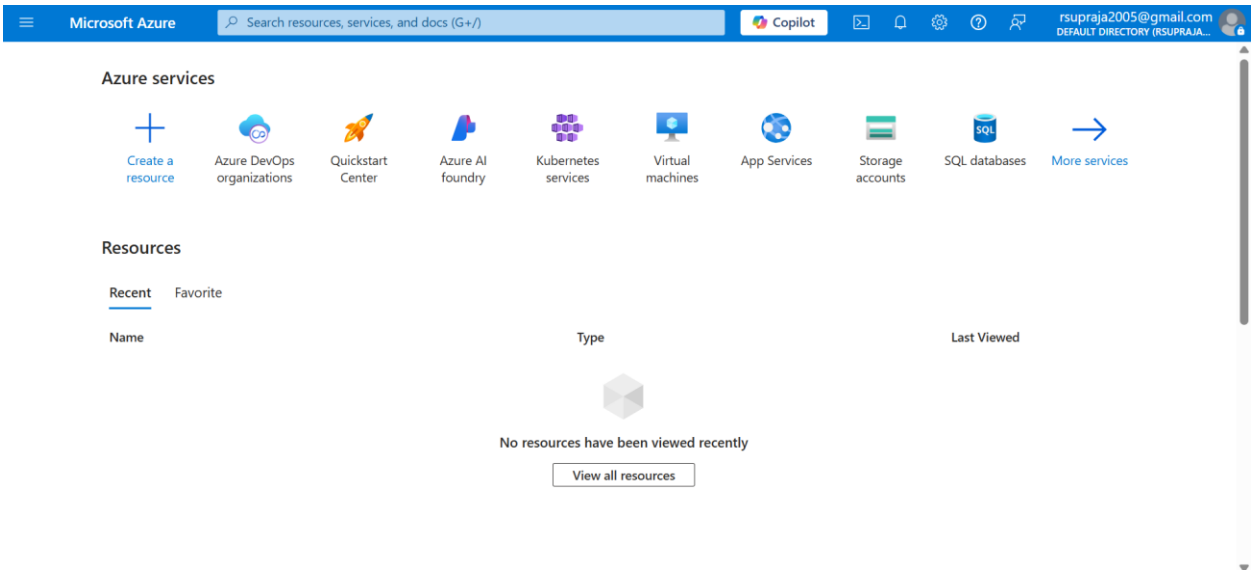
1. Open your web browser and go to the Azure website: https://azure.microsoft.com/en-us/getstarted/azure-portal.

Sign in using your Microsoft account credentials.

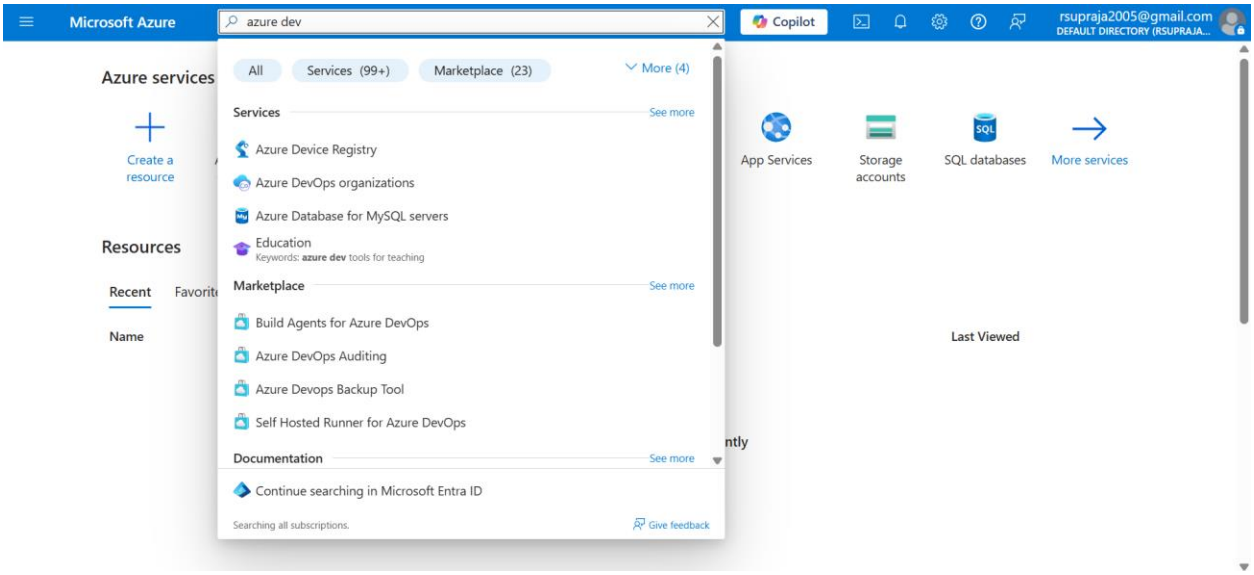
If you don't have a Microsoft account, you can create one here: https://signup.live.com/?lic=1



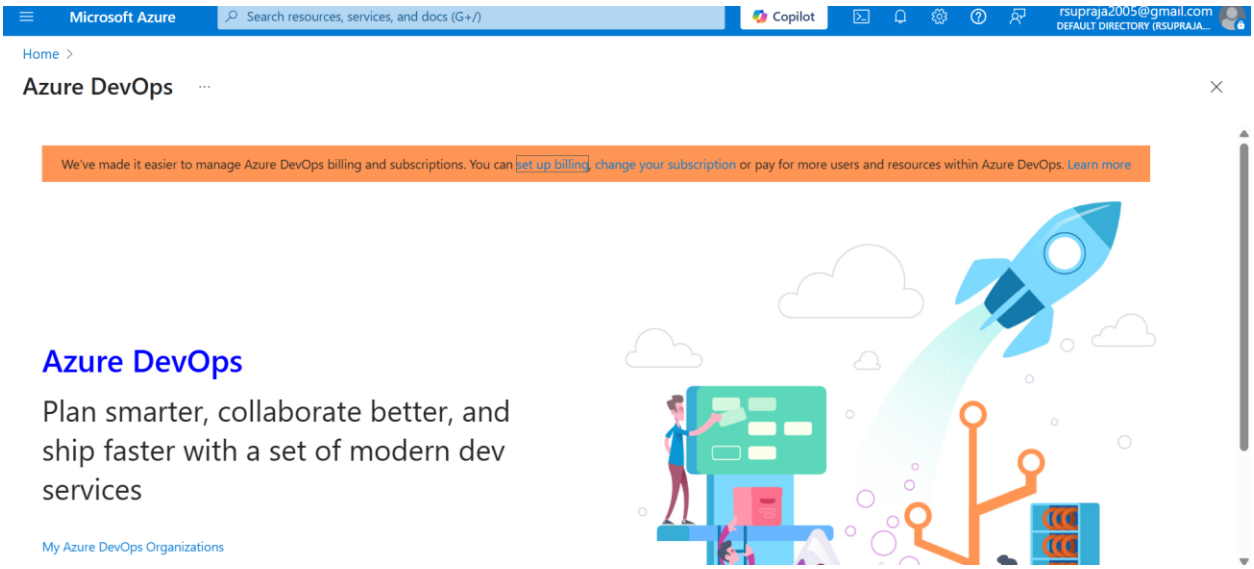
2.**Azure home page**



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



3.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:

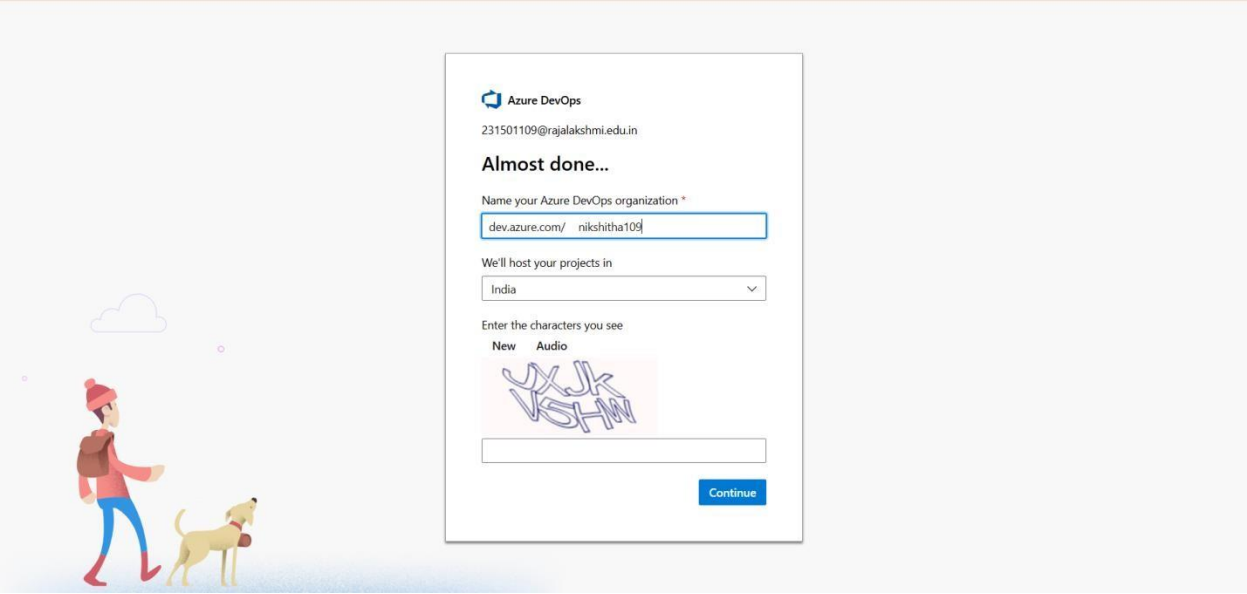
Successfully accessed the Azure DevOps environment and created a new organization through the Azure portal.

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.

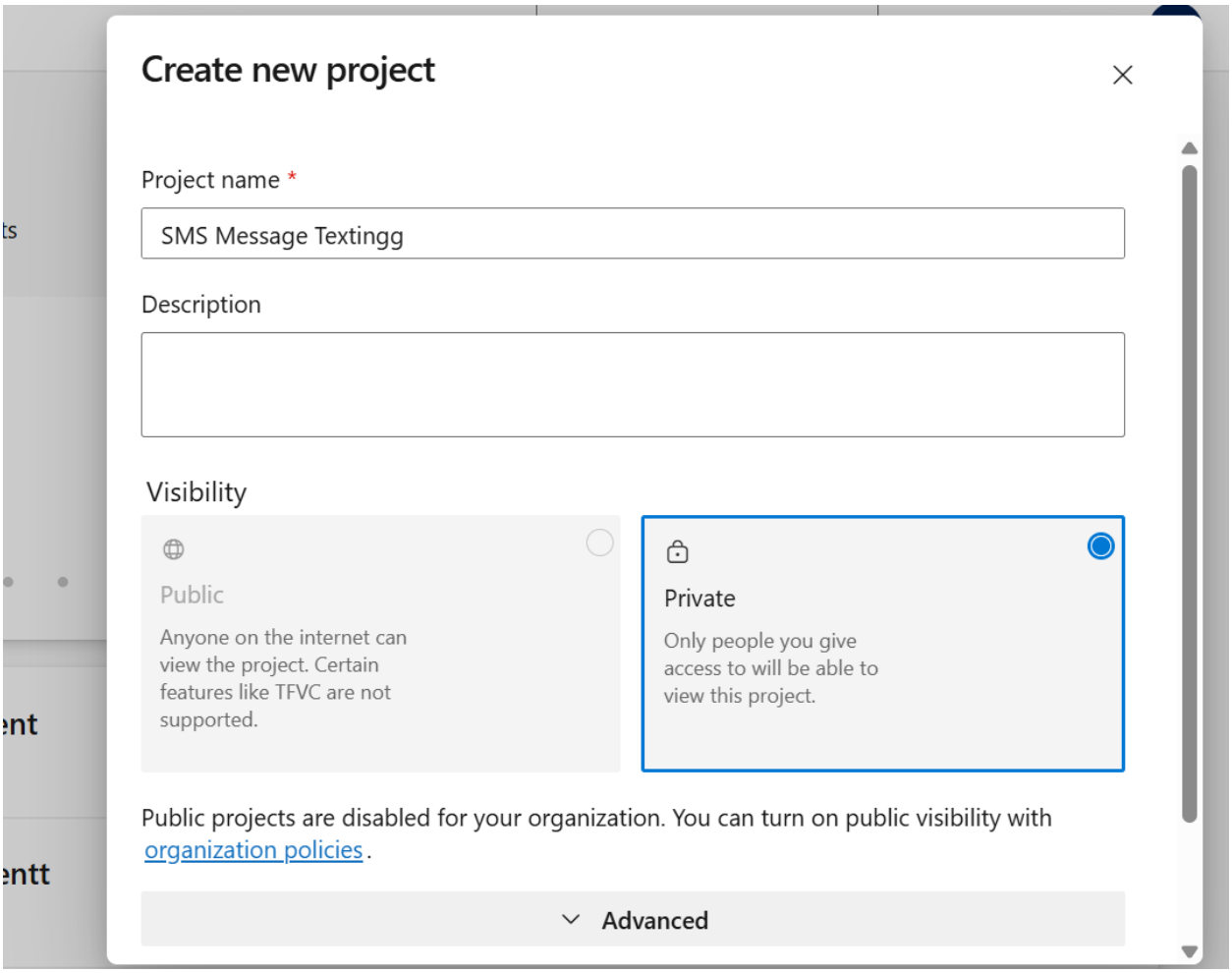
c. Enter the project name, description, and visibility options:

***Name**:* Choose a name for the project (e.g., LMS).

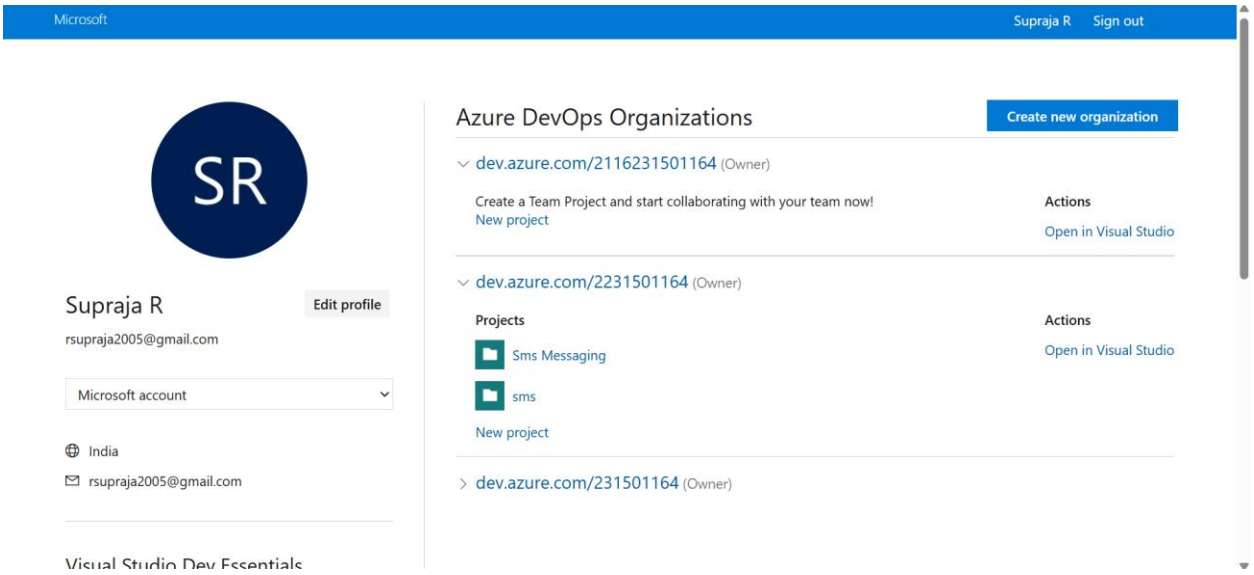
***Description**:* Optionally, add a description to provide more context about the project.

***Visibility**:* Choose whether you want the project to be **Private** (accessible only to those invited) or **Public** (accessible to anyone).

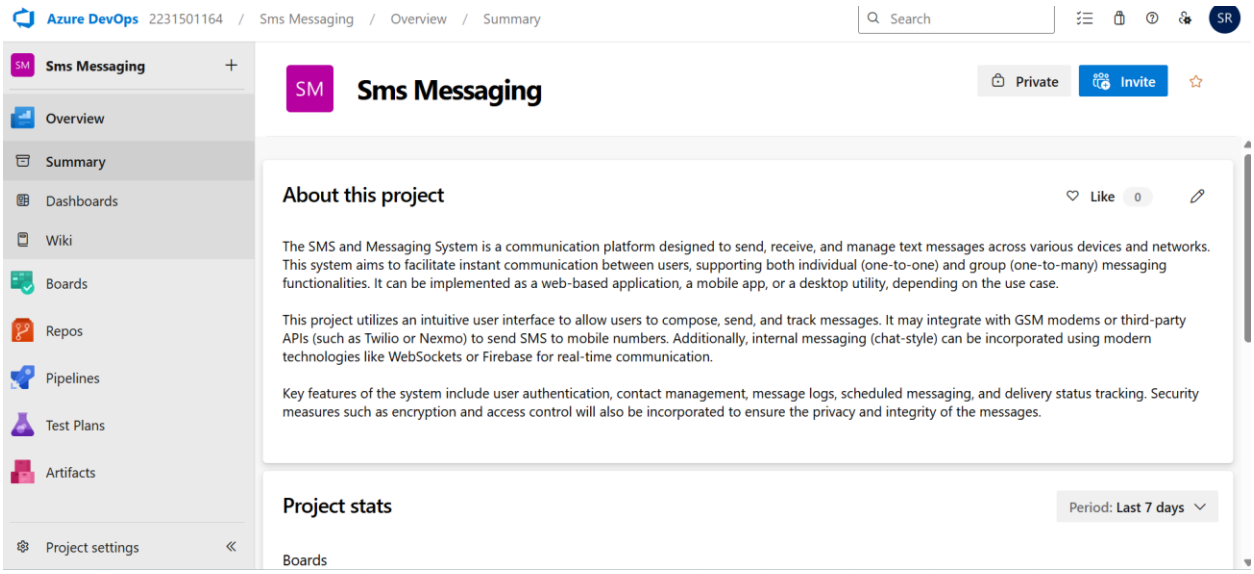
d. Once you've filled out the details, click **Create** to set up your first project.



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.



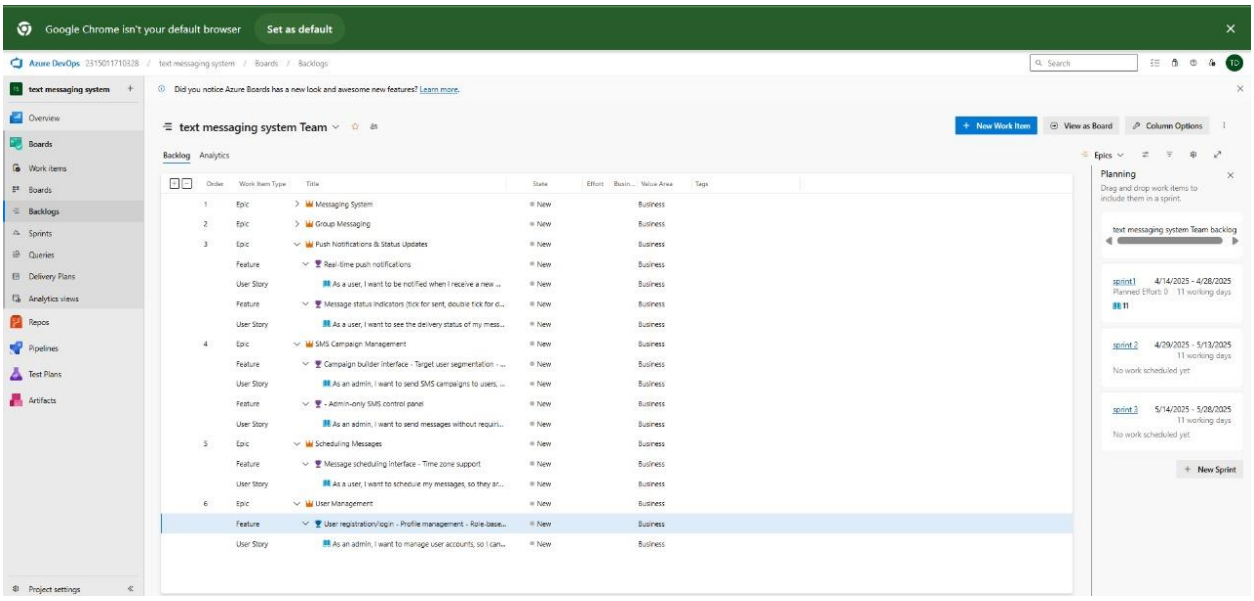
4. Project dashboard

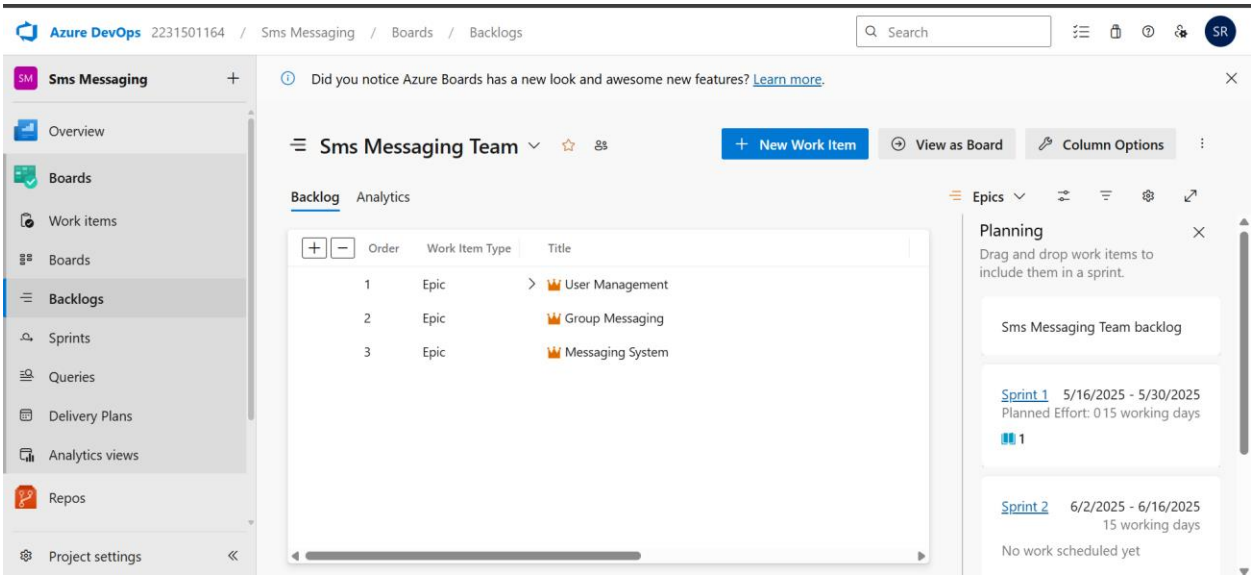


5. To manage user stories:

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.





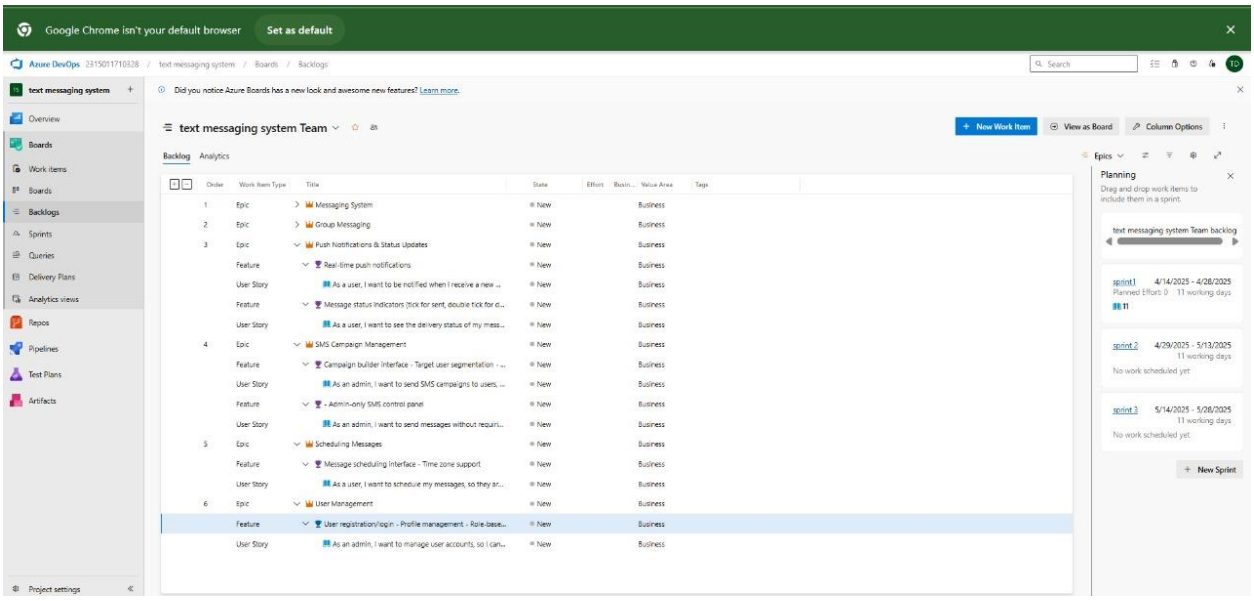
Result: Successfully created an Azure DevOps project with user story management and agile workflow setup.

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

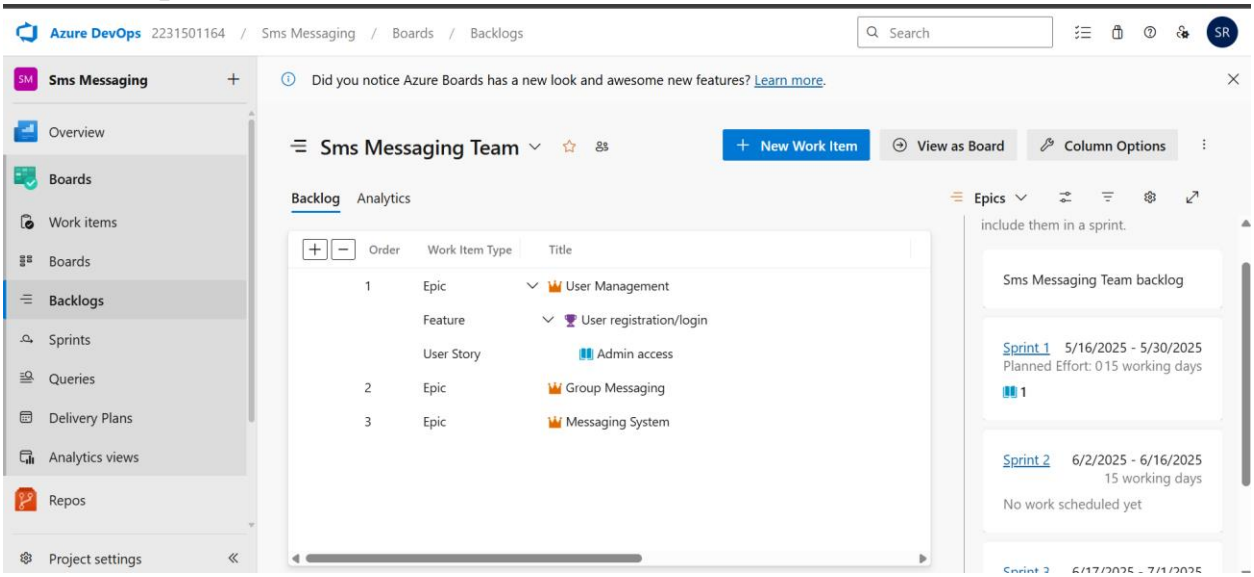
Aim:

To create epics, user stories, features, and tasks for the project, Batch Data Analysis and Visualization.

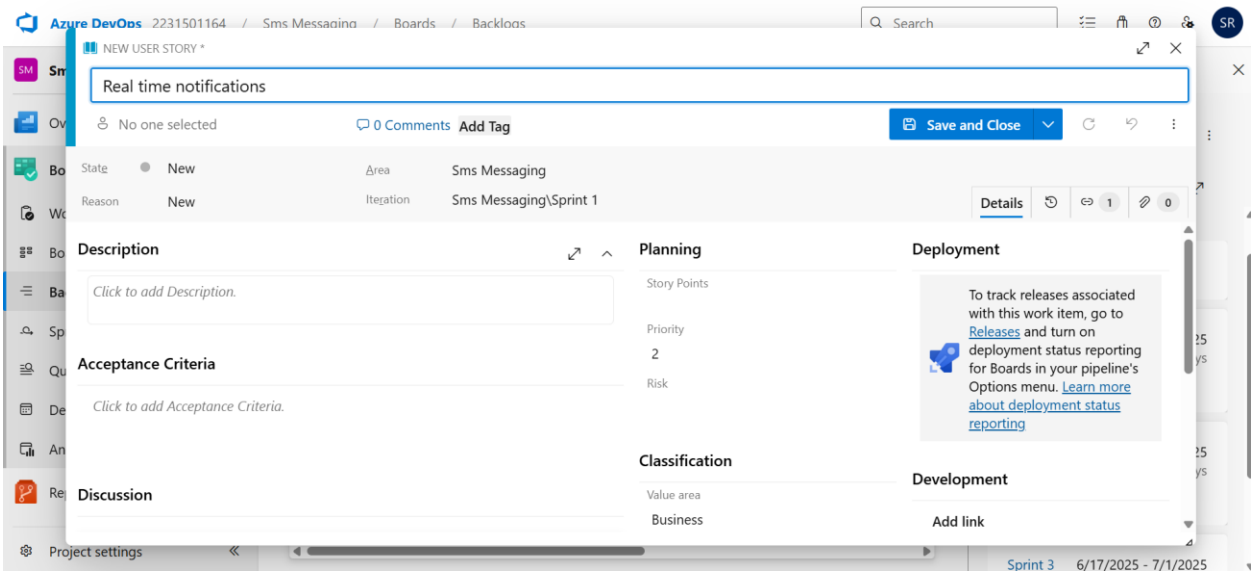
1.Create Epic, Features, User Stories, Task



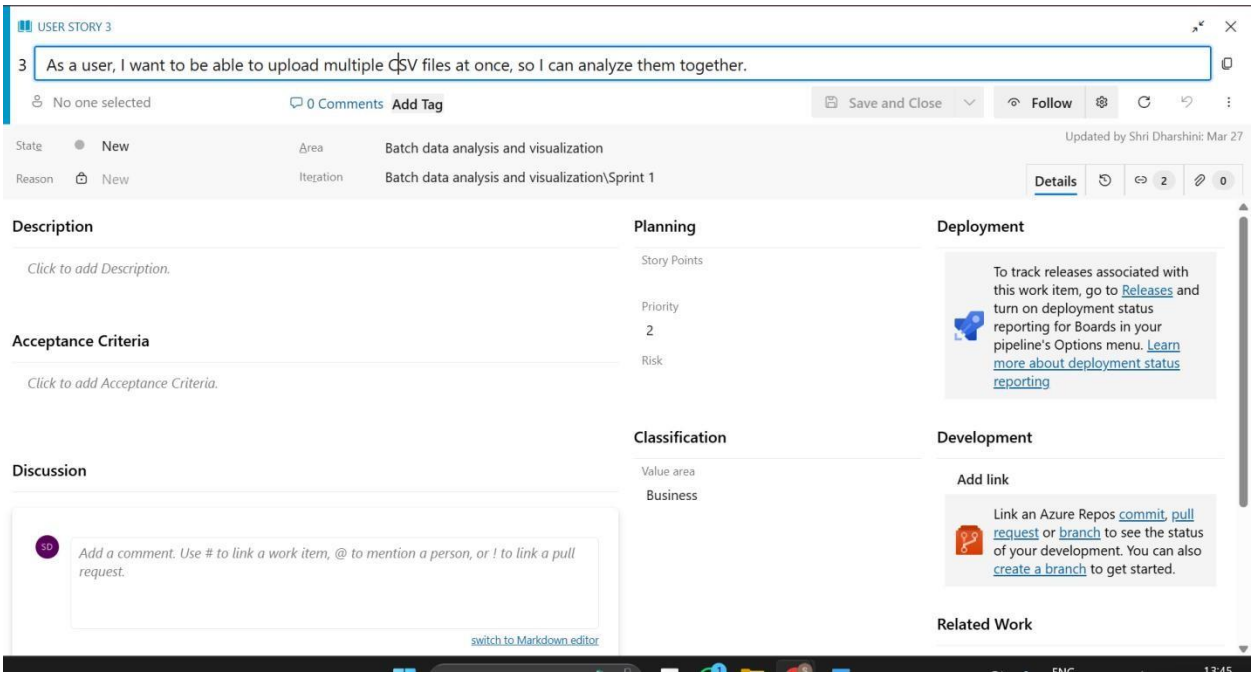
2.Fill in Epics



3. Fill in Features



4.Fill in User Stories



Result: Thus, epics, features, user stories, and tasks have been created successfully.

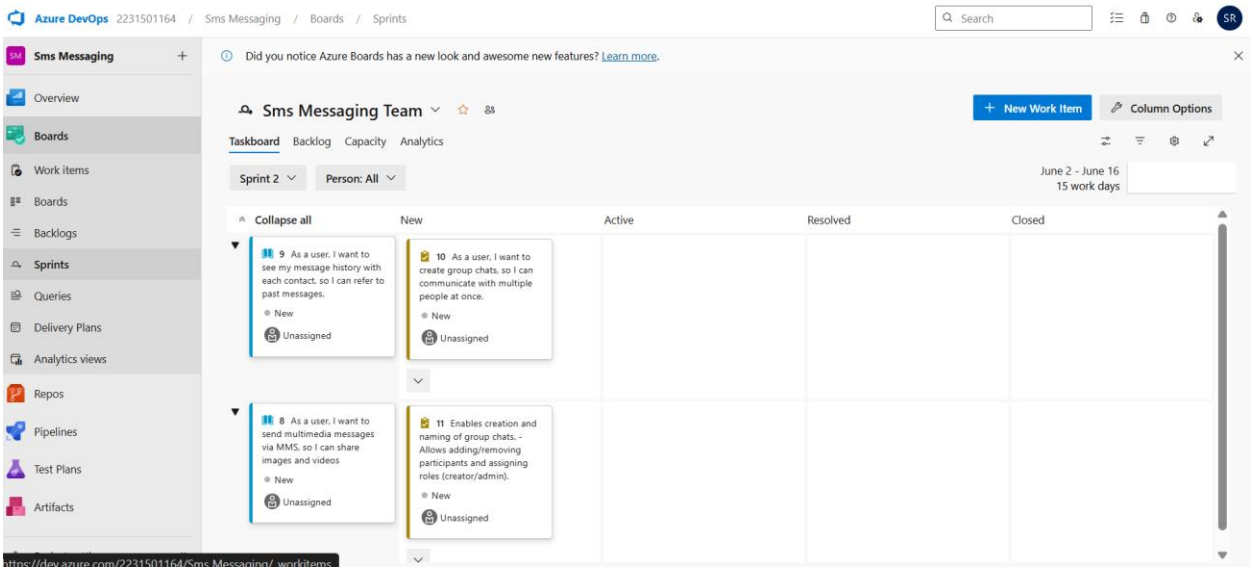
EXP NO: 4 Date :	SPRINTPLANNING

Aim:

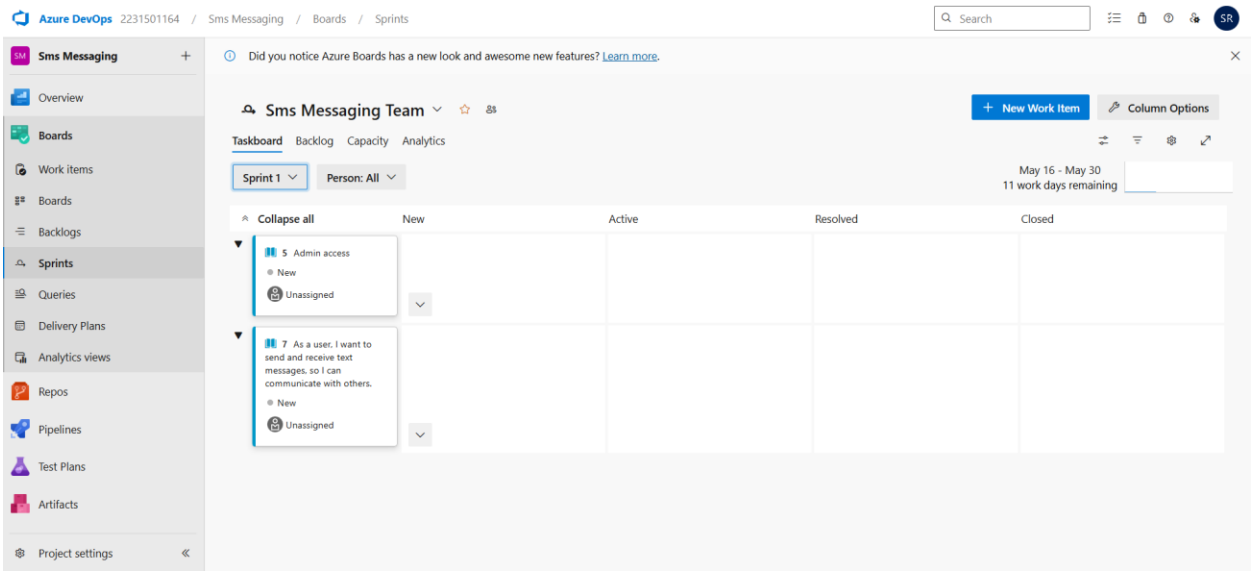
To assign a user story to a specific sprint for the project, Batch Data Analysis and Visualization.

SPRINT PLANNING

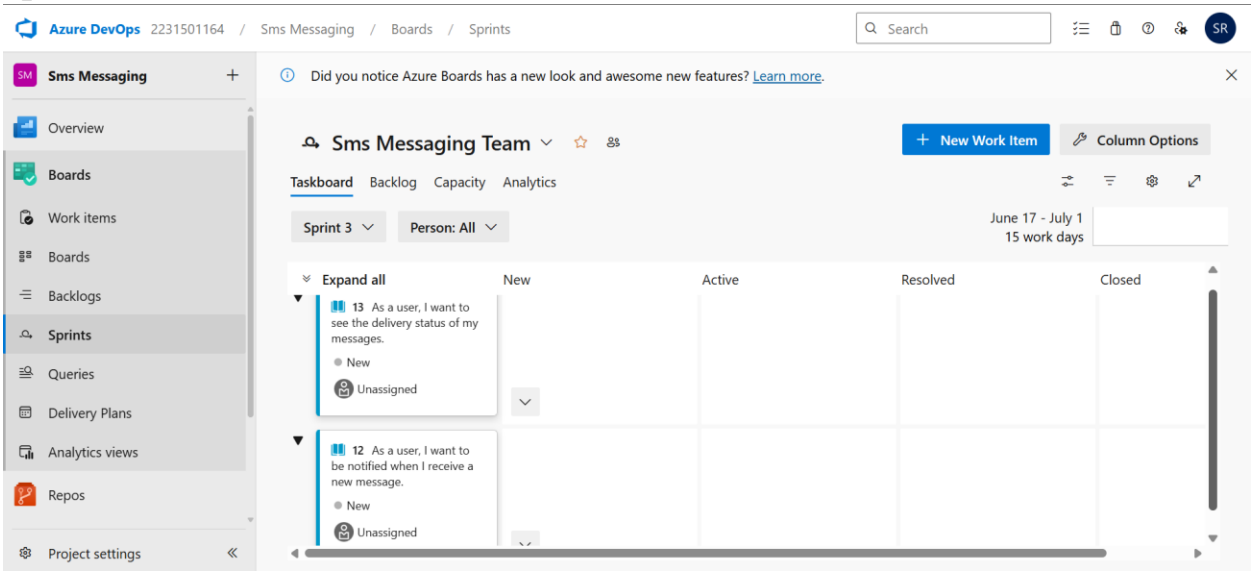
Sprint 1



Sprint 2



Sprint 3



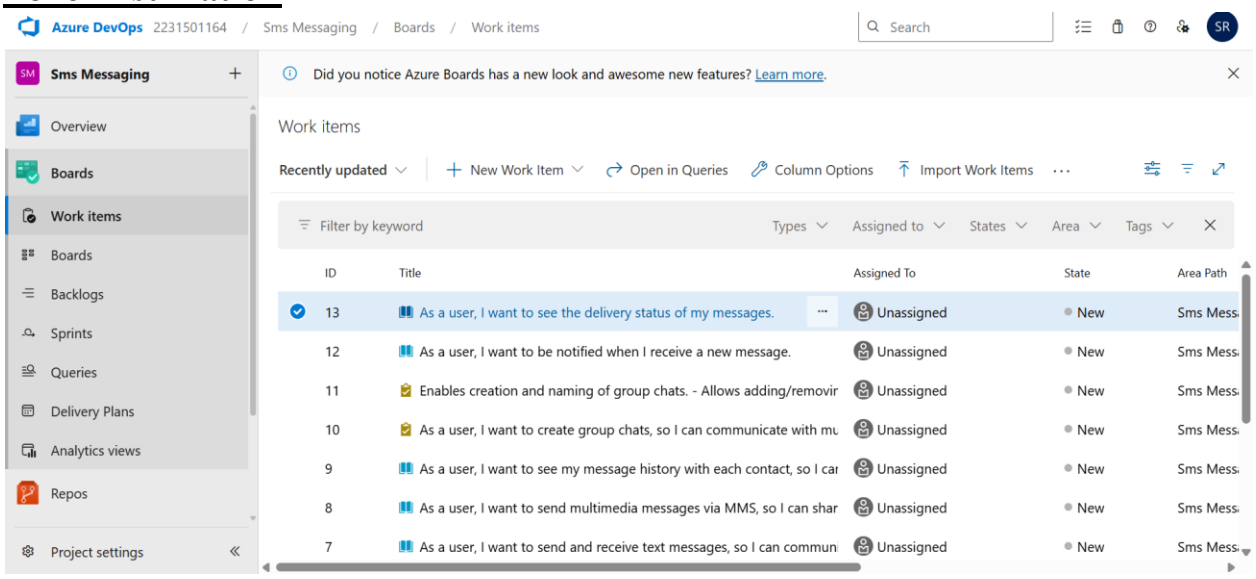
Result: The Sprints are created for the project, Batch Data Analysis and Visualization.

EXP NO: 5 Date :	POKERESTIMATION

Aim:

Create Poker Estimation for the user stories for the project, Batch Data Analysis and Visualization.

Poker Estimation



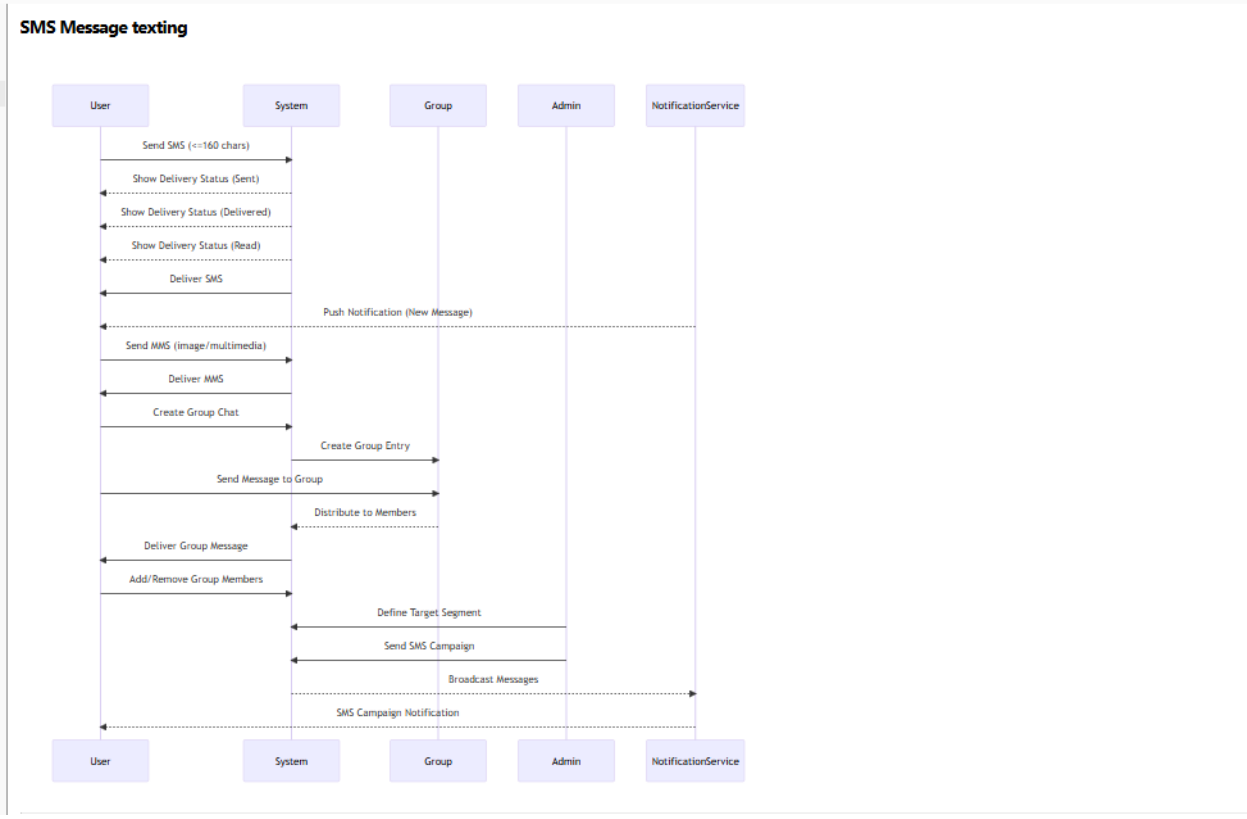
Result: The Estimation/Story Points is created for the project using Poker Estimation.

EXP NO: 6 Date :	DESIGNING CLASS DIAGRAM AND SEQUENCE DIAGRAM

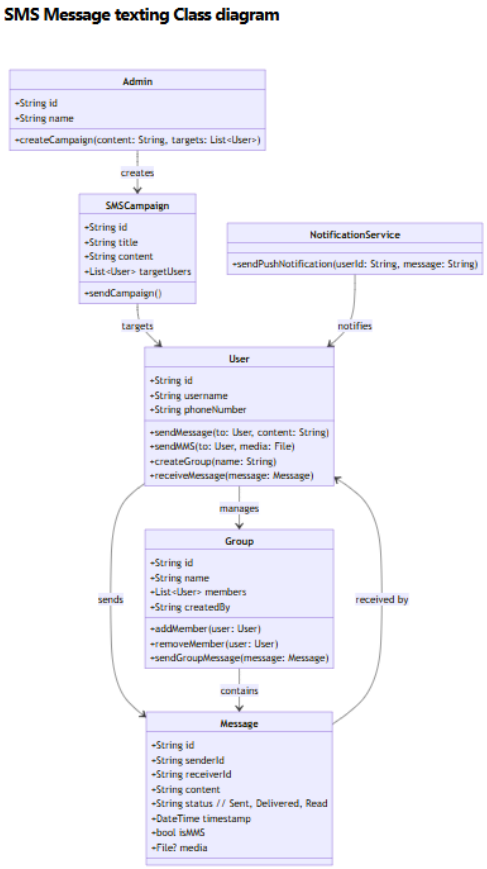
Aim:

To design a Class Diagram and Sequence Diagram for the project, Batch Data Analysis and Visualization.

6A. Class Diagram



6B. Sequence Diagram



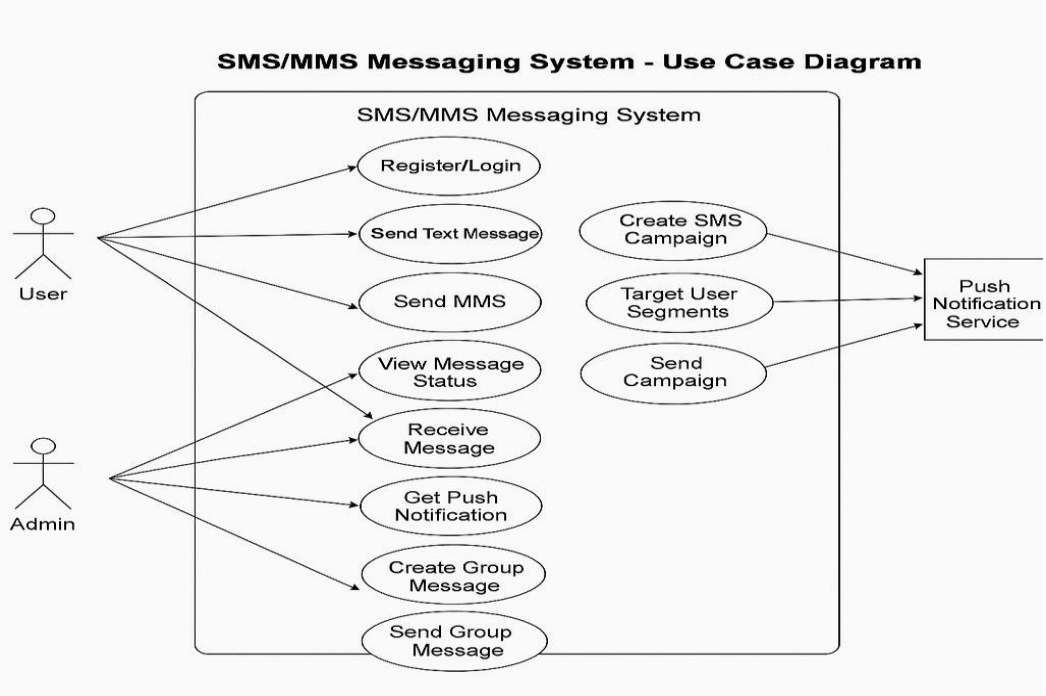
Result: The Class and Sequence Diagrams are designed successfully for the project, SMS Message System.

EXP NO: 7 Date :	DESIGNING USE CASE DIAGRAM AND ACTIVITY DIAGRAM

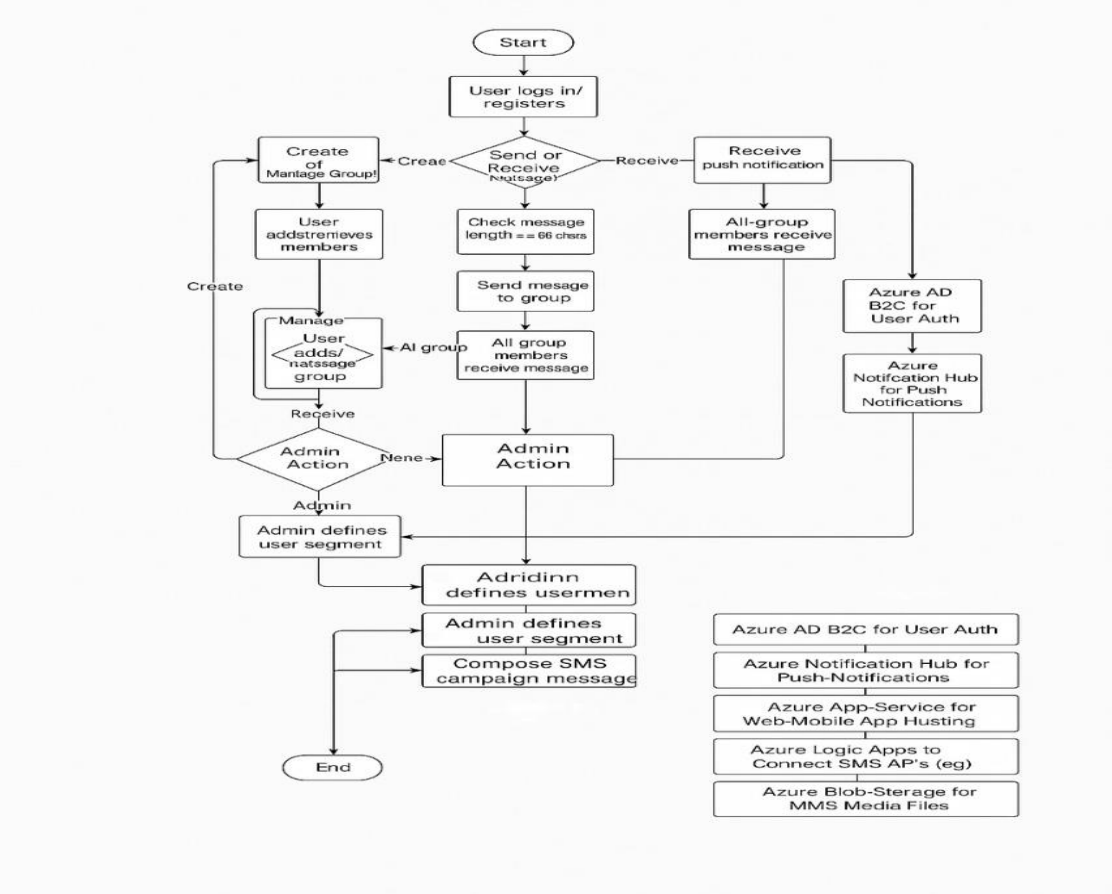
Aim:

To design a Use Case Diagram and an Activity Diagram for the project, SMS Messaging System

7A. Use Case Diagram



7B. Activity Diagram



Result: The Use Case and Activity Diagrams are designed successfully for the project, Batch Data Analysis and Visualization.

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

1. User Authentication
2. Uploading and Managing Batch Data Files
3. Running Batch Analysis Jobs
4. Viewing Interactive Visualizations and Charts
5. Exporting Analysis Results

2. Define User Interactions

- Simulate real scenarios (e.g., upload dataset, trigger job, download result).

3. Design Happy Path Test Cases

- Validate all main functions work properly (e.g., successful login, upload, and visualization).

4. Design Error Path Test Cases

- Simulate unexpected or invalid user behavior (e.g., upload fails, unsupported file, job timeout).

5. Break Down Steps and Expected Results

- Each test case includes step-by-step actions and expected outcomes.

6. Use Clear Naming and IDs

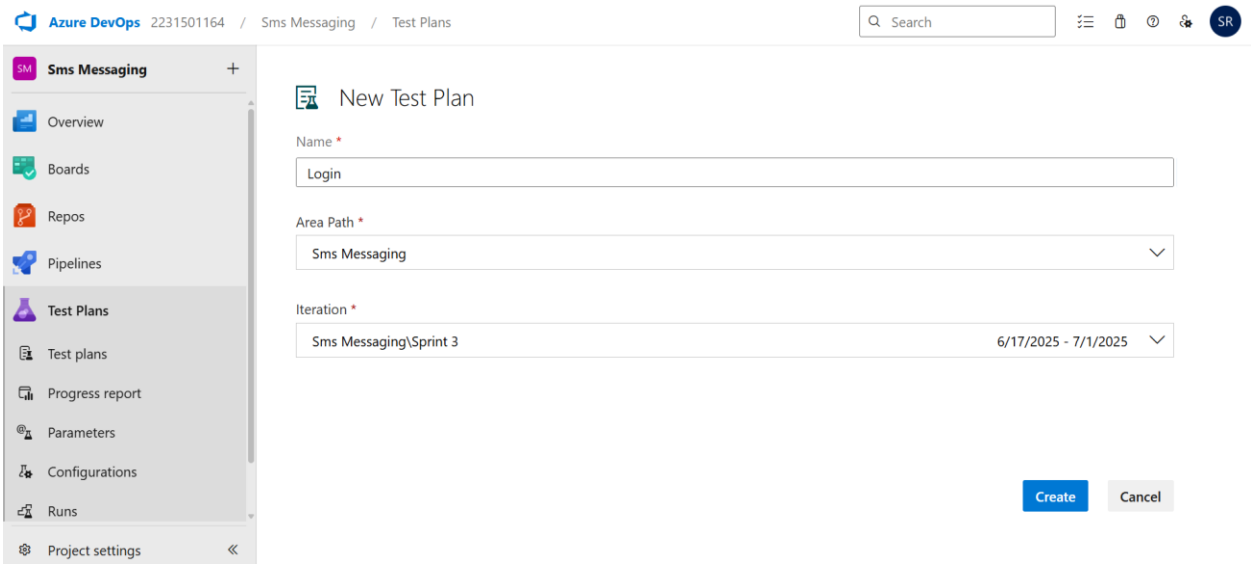
- Example: TC01 – Successful File Upload, TC08 – Visualization Fails.

7. Separate Test Suites

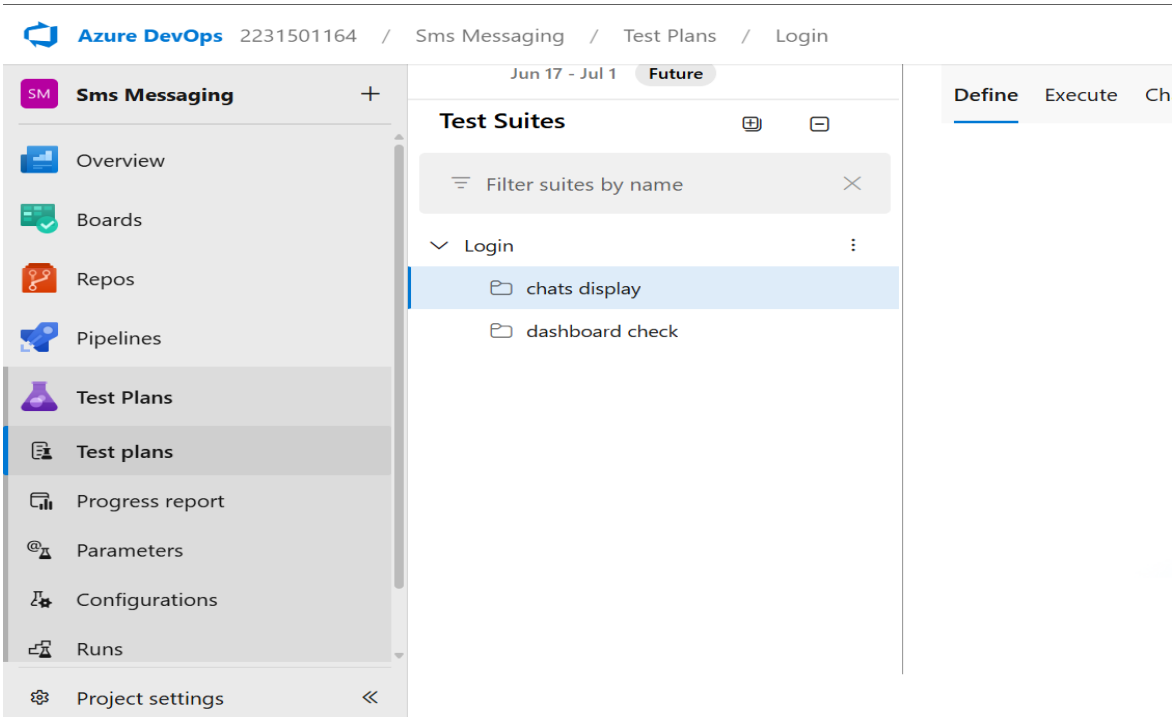
- Suites grouped by modules (Login, File Upload, Job Execution, Visualization, Export).

8. Prioritize and Review

- Critical test cases marked as High Priority.
- Mapped to user stories in Azure DevOps. **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.

Batch Data Analysis and Visualization – Test Plans

USER STORIES

- As a user, I want to log in using my username and password so that I can access my account.
- As a user, I should not be able to submit the login form with empty fields so that I can provide the required data.
- As a user, I want to log out when I click the logout button so that I can end my session securely.
- As a user, I want to be redirected to the login page after logging out so that I know my session has ended and I can log in again if needed.
- As a user, I want to be able to upload multiple CSV files at once, so I can analyze them together.

Test Suites

Test Suite: TS01 - User Authentication (ID: 54)

1. **TC01 – Successful Login** ○
   Action:
   - Navigate to the login page
   - Enter valid credentials
   - Click "Login" ○

   Expected Results:
   - User redirected to dashboard.

   ○ **Type**: Happy Path
2. **TC02 – Prevent Login with**

   Empty Fields ○ **Action:**
   - Navigate to the login page.
   - Leave username and/or password fields empty.
   - Click on "Login".

   ○ **Expected Results:**

- Validation error message is shown prompting user to fill required fields.

- **Type:** Error Path

Test Suite: TS02 - Logout Functionality(ID: 47)

1. **TC03 – Successful Logout and Redirect**

   Action:

   - Log in successfully.
   - Click the "Logout" button.

   - **Expected Results:**
     - User session ends.
     - User is redirected to the login page.
   - **Type:** Happy Path

2. **TC04– Access Protected Page After Logout** **Action:**

   - Logout.
   - Attempt to navigate back to a protected page (e.g., dashboard) via browser back button or URL. **Expected Results:**
     - User is redirected to the login page and denied access.
   - **Type:** Error Path

Test Suite: TS03 - CSV Upload Functionality (ID: 88)

1. **TC05 – Upload Multiple Valid CSV Files**
   Action:

   - Log in successfully
   - Navigate to the CSV upload section
   - Select multiple valid .csv files
   - Click "Upload"

   - **Expected Results:**
     - All files are uploaded successfully.
     - Files are listed and ready for analysis.
   - **Type:** Happy Path

2. **TC06 – Upload Attempt Without Selecting**

Files ○ **Action:**

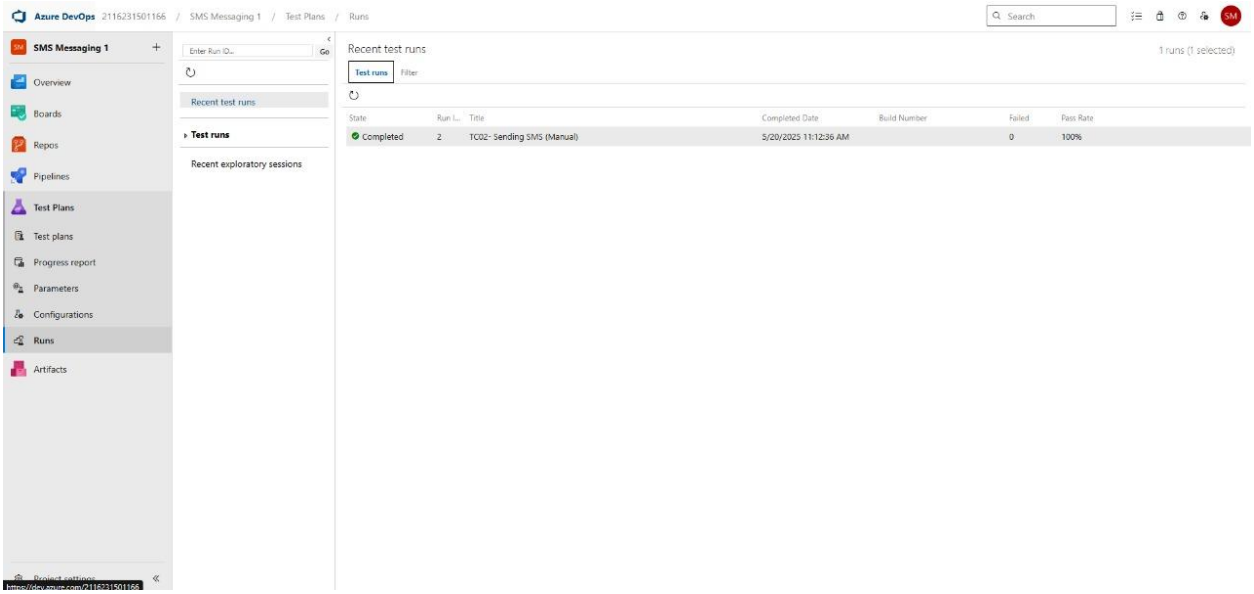
- Navigate to the CSV upload section
- Click "Upload" without selecting any files.

○ **Expected Results:**

- Validation message prompting the user to select at least one file.

Type: Error Path

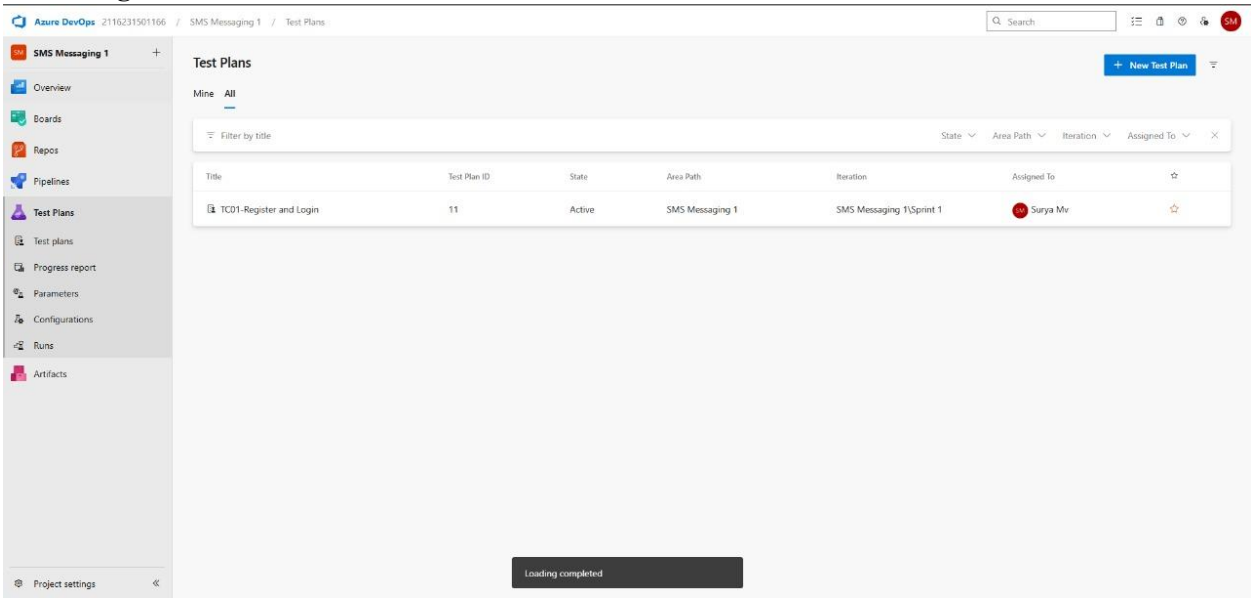
Test Cases

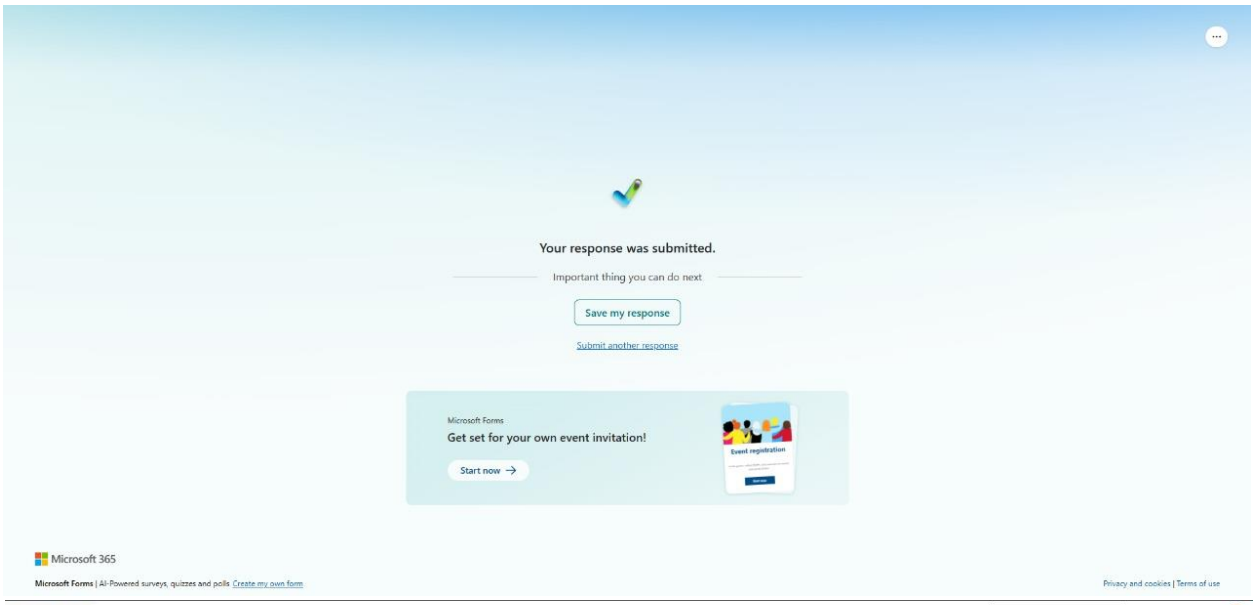


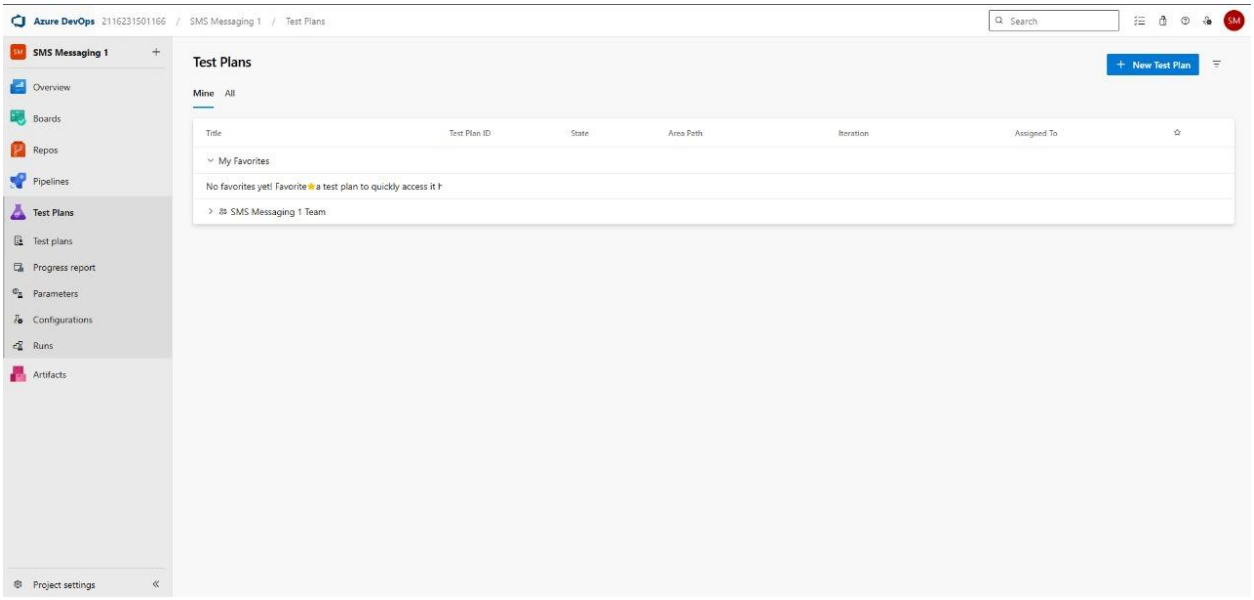
Test and feedback

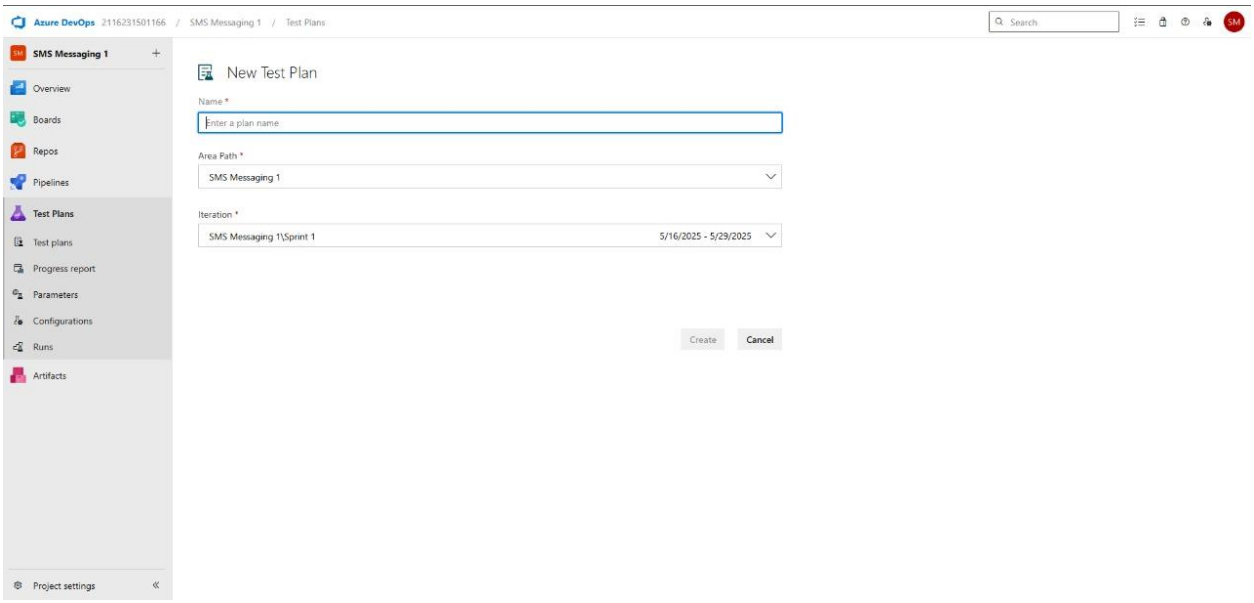
Showing it as an extension

4.Running the test cases

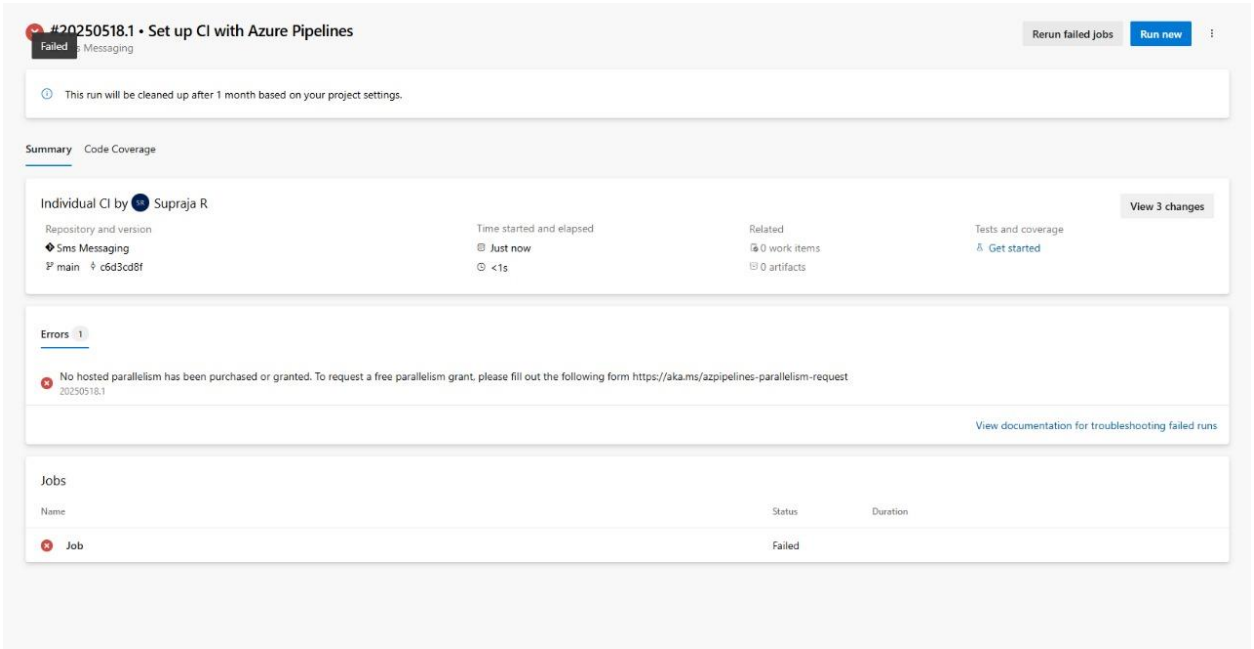



Your response was submitted.
Important thing you can do next
Save my response
Submit another response
Microsoft Forms
Get set for your own event invitation!
Start now
Microsoft 365
Microsoft Forms | AI-Powered surveys, quizzes and polls Create my own form
Privacy and cookies | Terms of use


Azure DevOps 2116231501166 / SMS Messaging 1 / Test Plans
Search
SMS Messaging 1
Overview
Boards
Repos
Pipelines
Test Plans
Test plans
Progress report
Parameters
Configurations
Runs
Artifacts
Project settings
Test Plans
New Test Plan
Mine All
Title
Test Plan ID
State
Area Path
Iteration
Assigned To
My Favorites
No favorites yet! Favorite a test plan to quickly access it h
SMS Messaging 1 Team



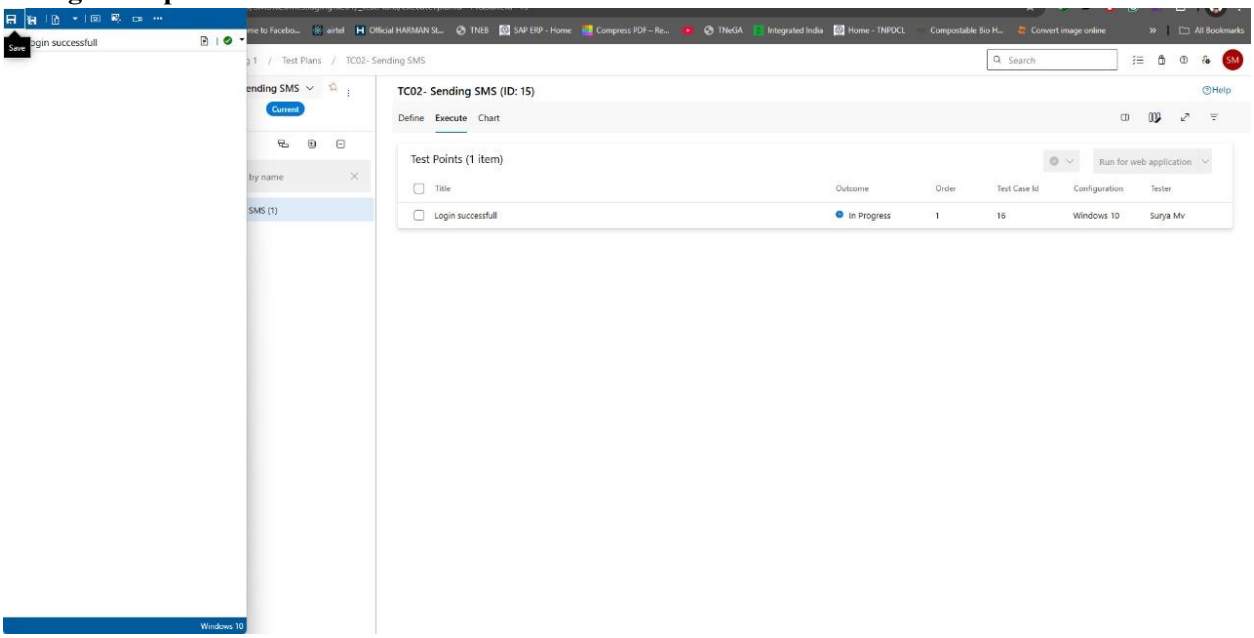
8.Test case results



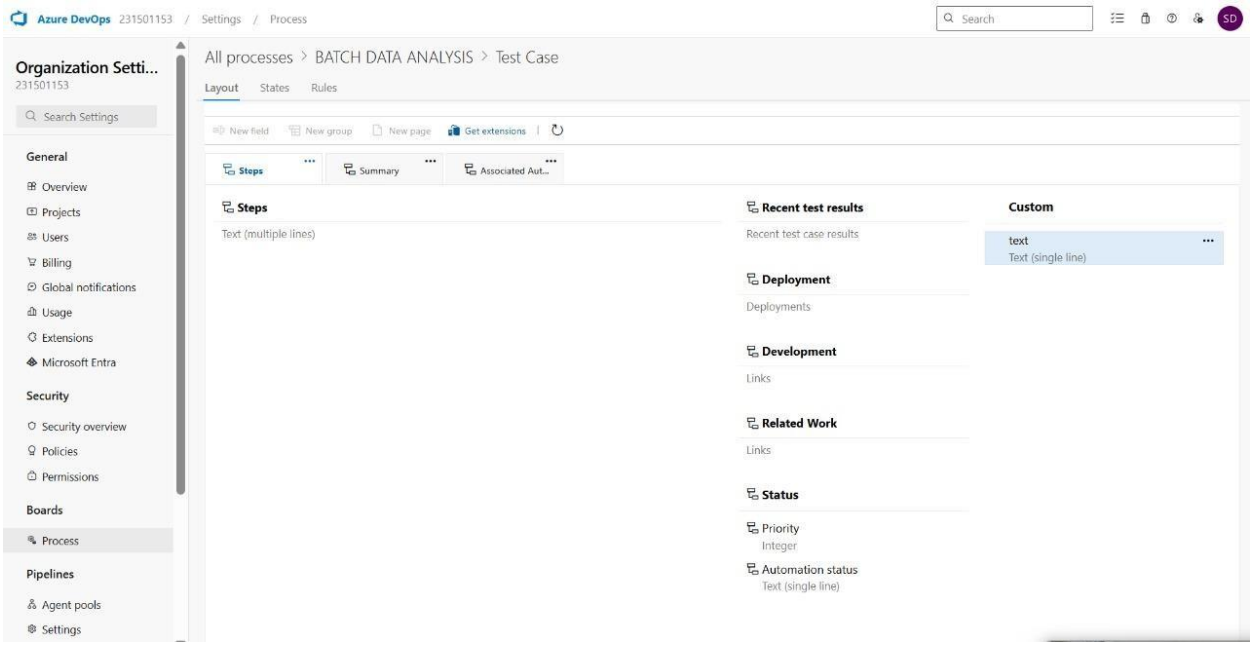
5.Test report summary

- Assigning bug to the developer and changing state

6.Progress report



7.View the new test case 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 create and demonstrate an Azure DevOps pipeline for automating application builds, tests, and deployment.

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.

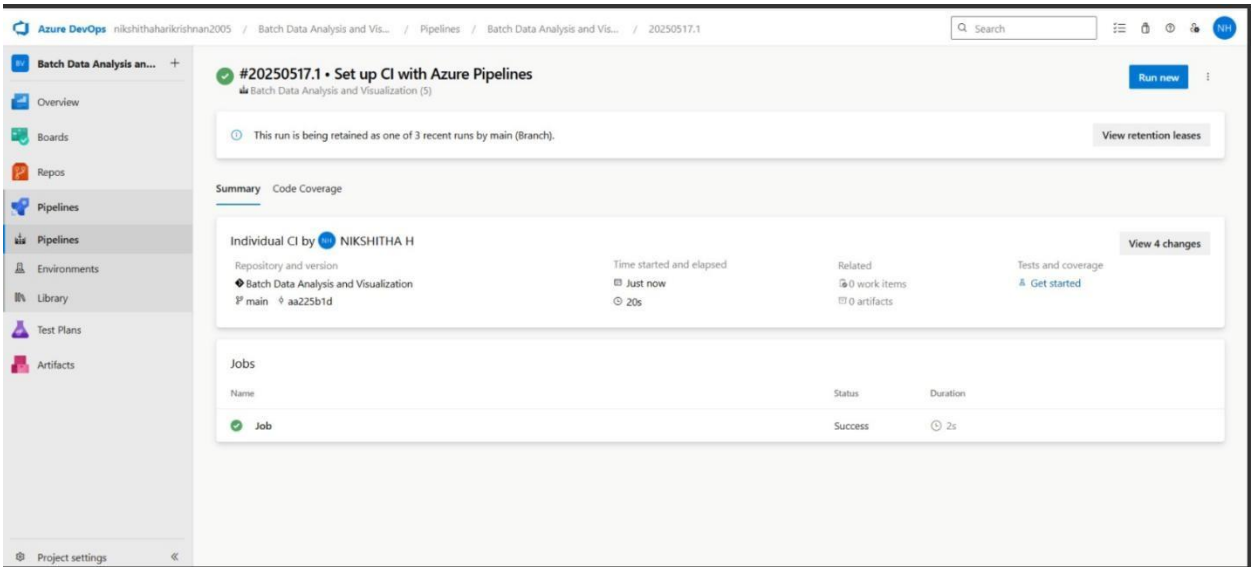
    Check logs for build/test/deploy stages to debug any errors.

    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.

Pipeline



Result:

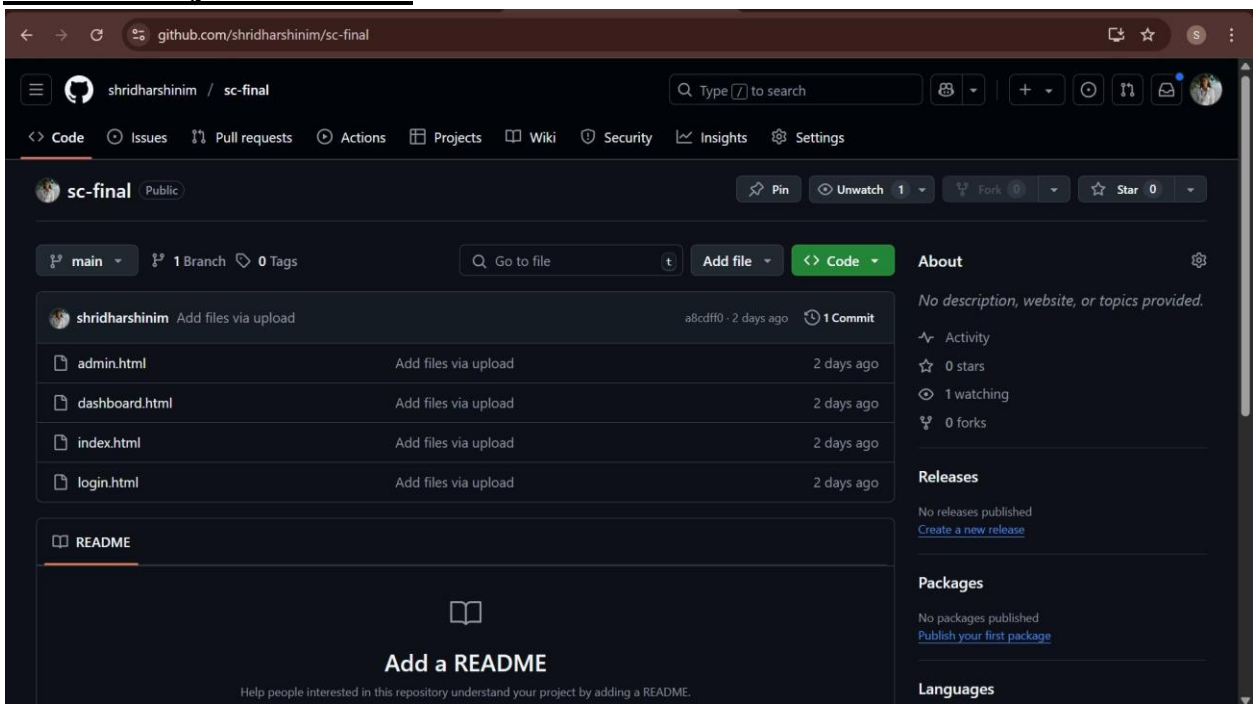
Successfully demonstrated pipelines in azure devops

EXP NO: 10 Create Epic, F **Date :**	**GITHUB: PROJECT STRUCTURE & NAMING** eatures, User Stories, Task **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 Music Playlist Batch Creator 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.