

**DEPARTMENT OF COMPUTER SCIENCE AND  
ENGINEERING LAB MANUAL**

**CS23432 – Software Construction**

**(REGULATION 2023)**

**RAJALAKSHMI ENGINEERING COLLEGE  
Thandalam, Chennai-602015**

Name: PRIYANGA M

Register No: 231901037

Year / Branch / Section: 3<sup>rd</sup> / CSE(CS) / A

Semester: V

Academic Year: 2025 - 2026

# 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 Architectural and ER Diagrams for Project Structure.     |
| 8.    | 25/3/25 | Testing – Test Plans and Test Cases.                               |
| 9.    | 15/4/25 | Load Testing and Pipelines.  |
| 10.   | 22/4/25 | GitHub: Project Structure & Naming Conventions.                    |

EXP NO: 1

## AZURE DEVOPS ENVIRONMENT SETUP

### Aim:

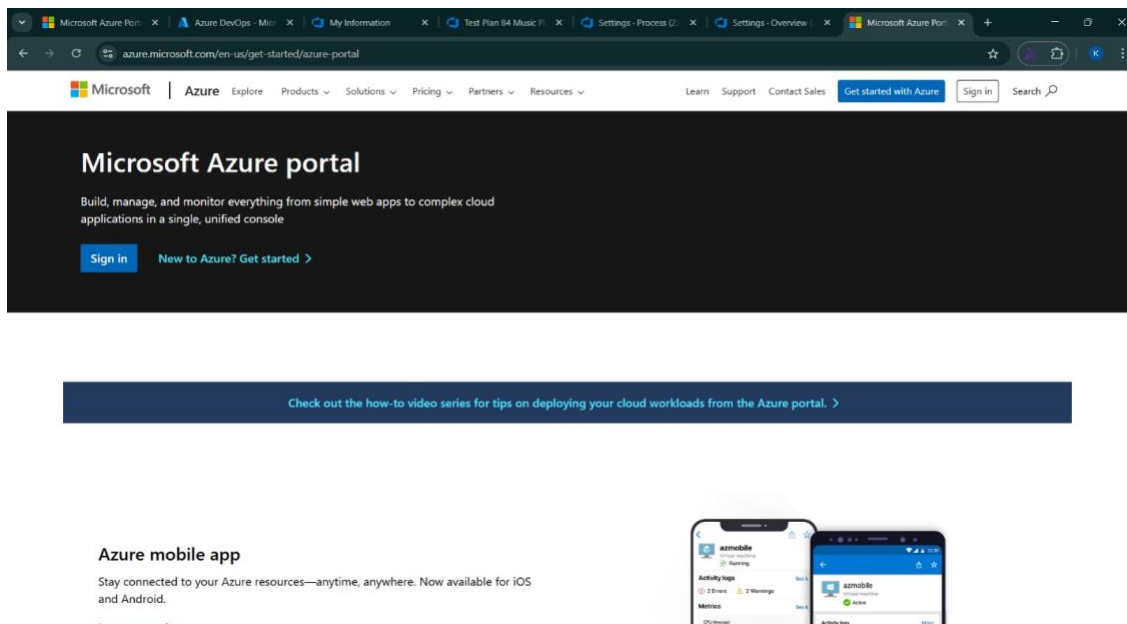
To set up and access the Azure DevOps environment by creating an organization through the Azure portal.

### INSTALLATION

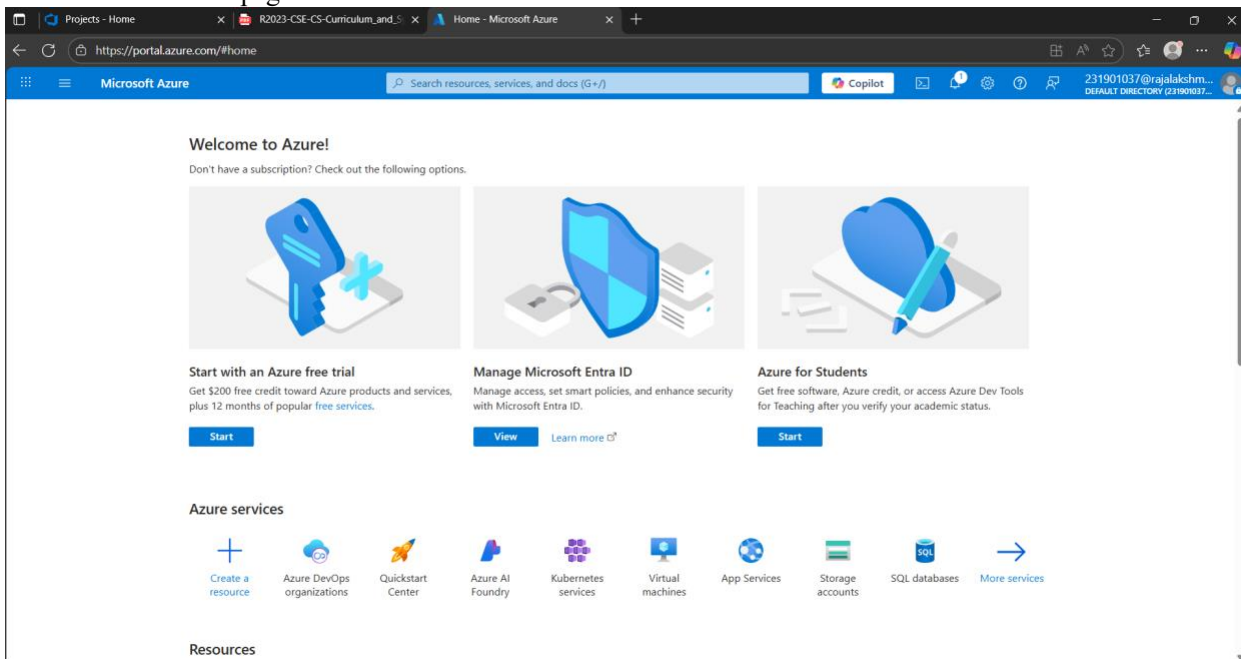
1. Open your web browser and go to the Azure website: <https://azure.microsoft.com/en-us/get-started/azure-portal>.

Sign in using your Microsoft account credentials.

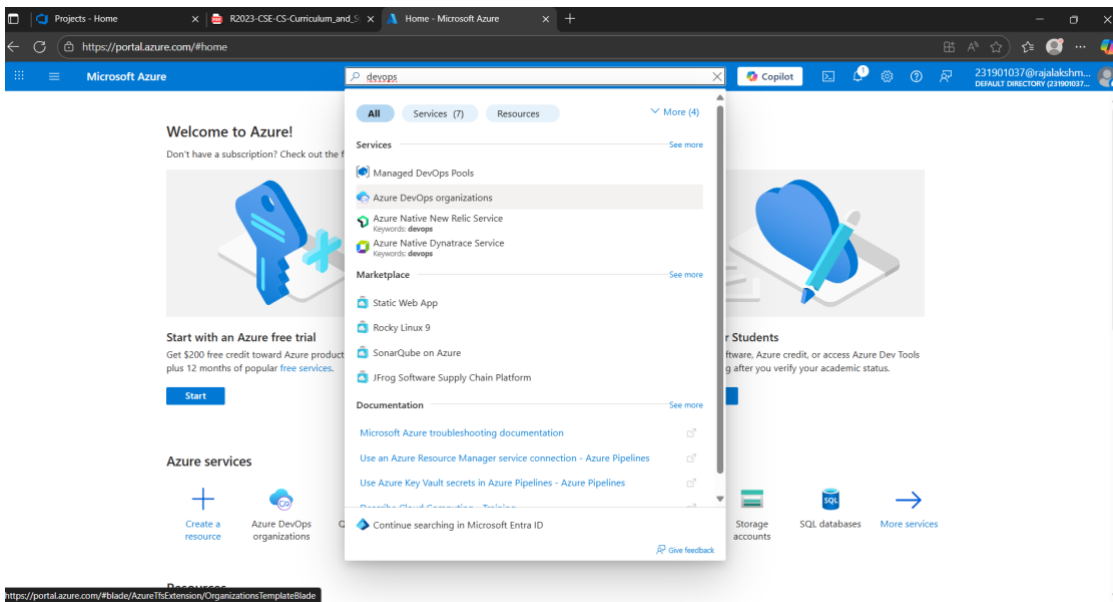
If you don't have a Microsoft account, you can create one here: <https://signup.live.com/?lic=1>



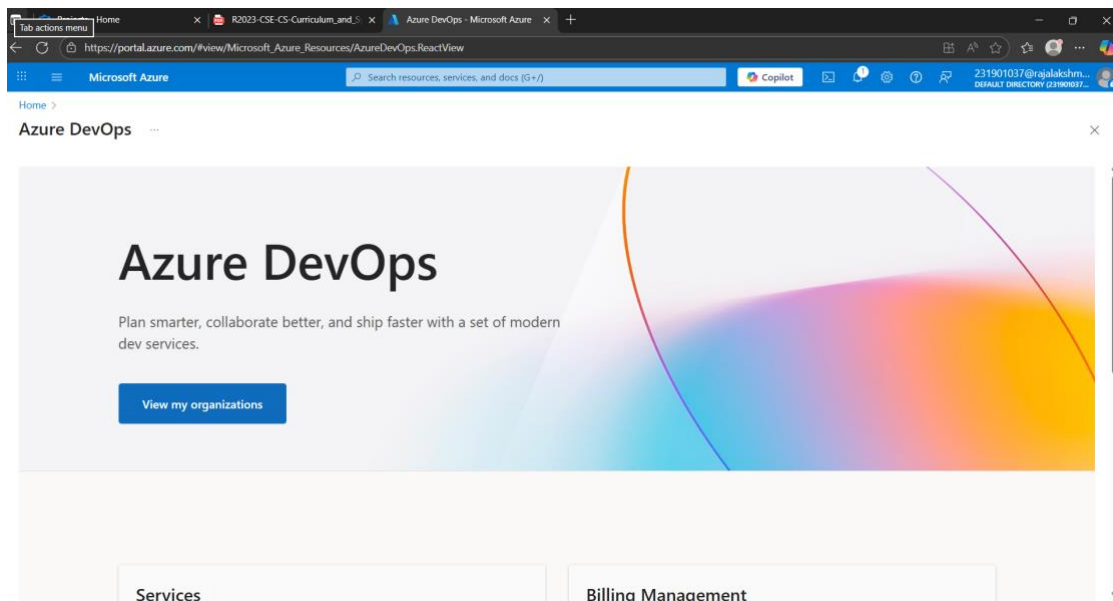
## 2. Azure home page



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.

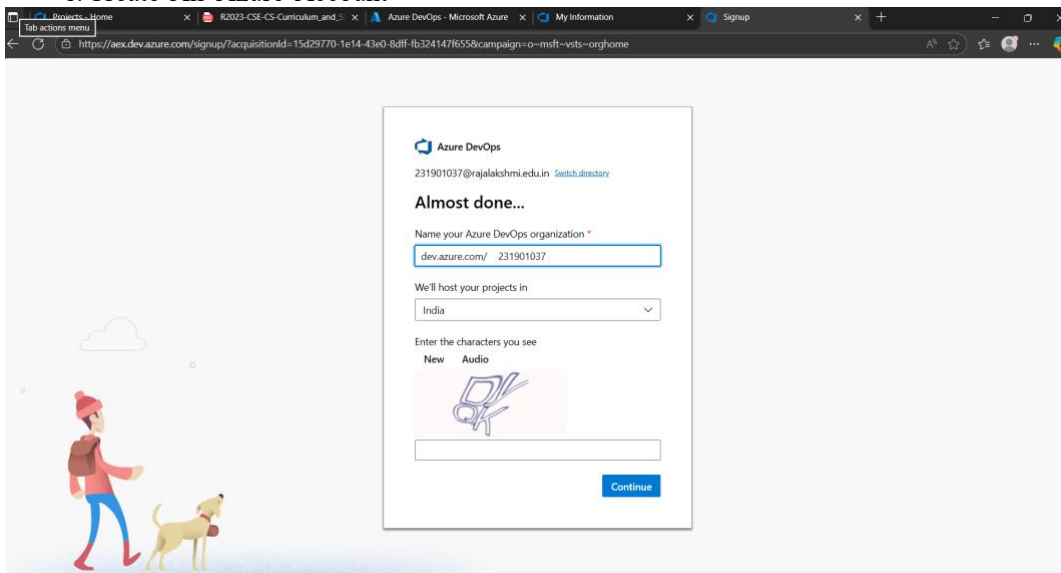
EXP NO: 2

## 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

- 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.
- On the organization's **Home page**, click on the **New Project** button.
- 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).
- Once you've filled out the details, click **Create** to set up your first project.

## Create new project



Project name \*

SoftwareConstruction Projed


Description

Visibility

 **Private**


Only people you give access to will be able to view this. Want to create a public project? [Try GitHub](#)

^ Advanced

Version control 

Git



Work item process 

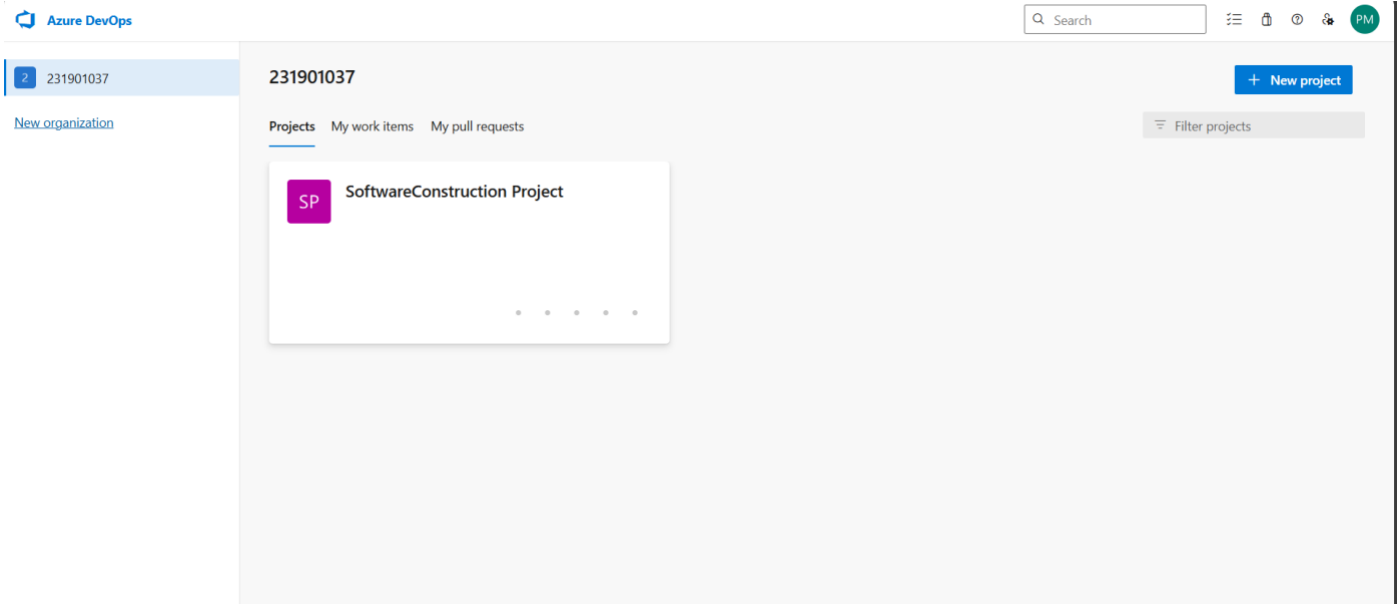
Agile



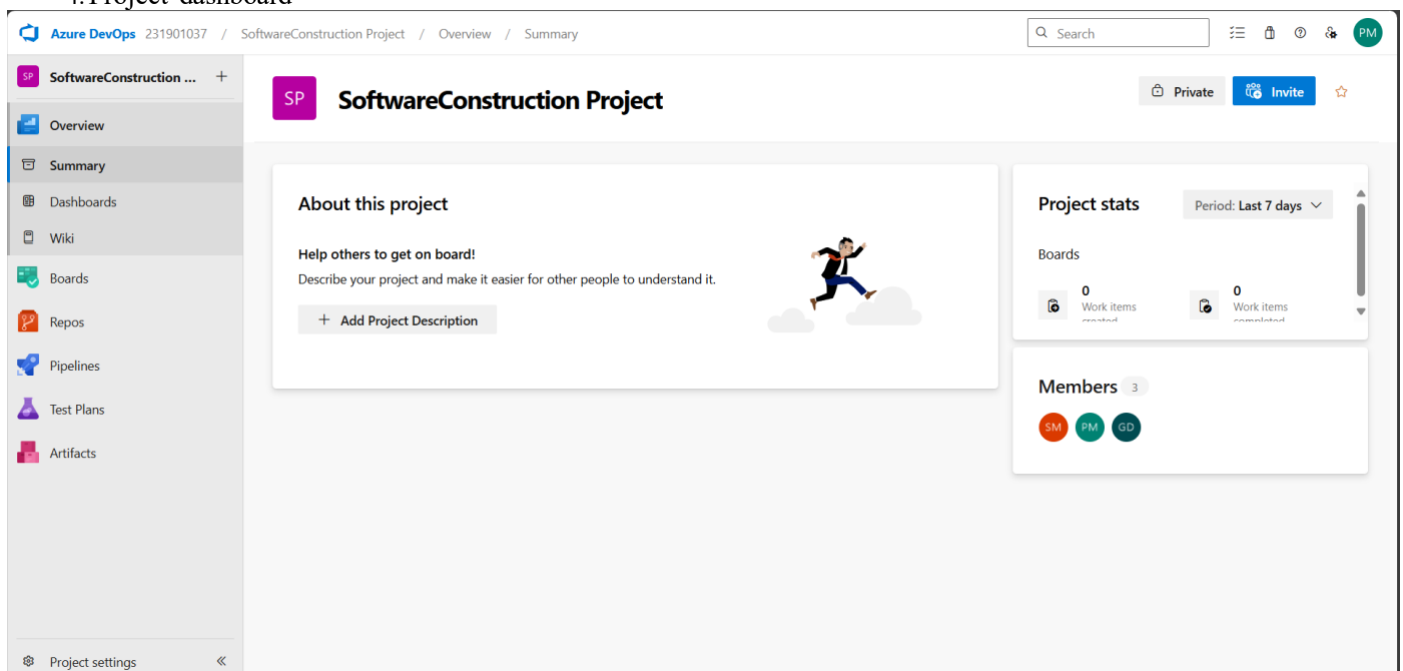
Cancel

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.


c.


## Invite members to SoftwareConstruction Project ×


Search and add users, service principals, or managed identities to your SoftwareConstruction Project project

Users

 Use semicolons to separate multiple members.

 **Guru sai charan D**  
231901501@rajalakshmi.edu.in

 **Saravanan MD**  
231901046@rajalakshmi.edu.in

 **Priyanga M**  
231901037@rajalakshmi.edu.in

**Result:**

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

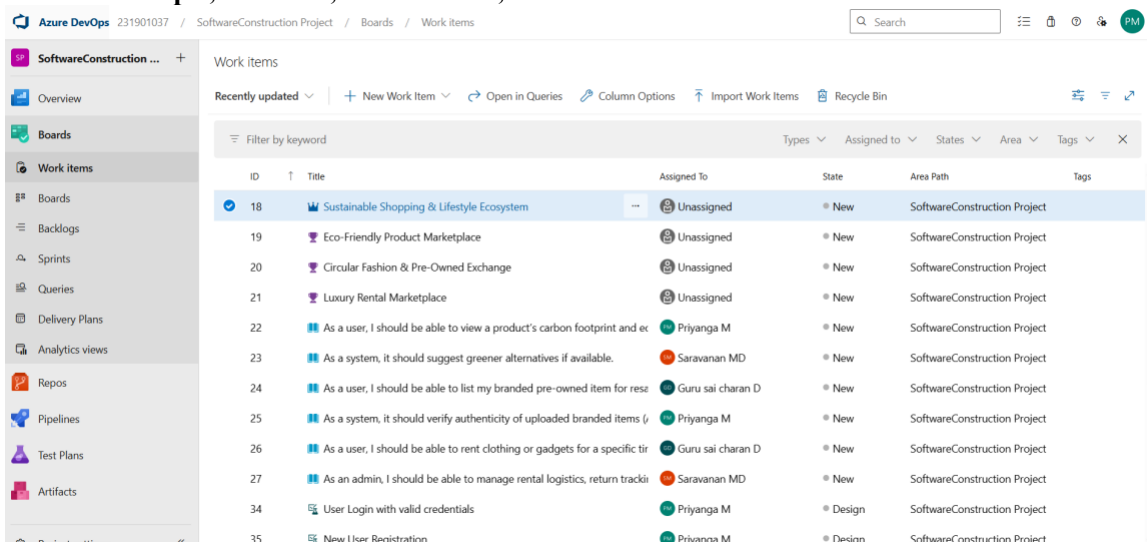
EXP NO: 3

## 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.

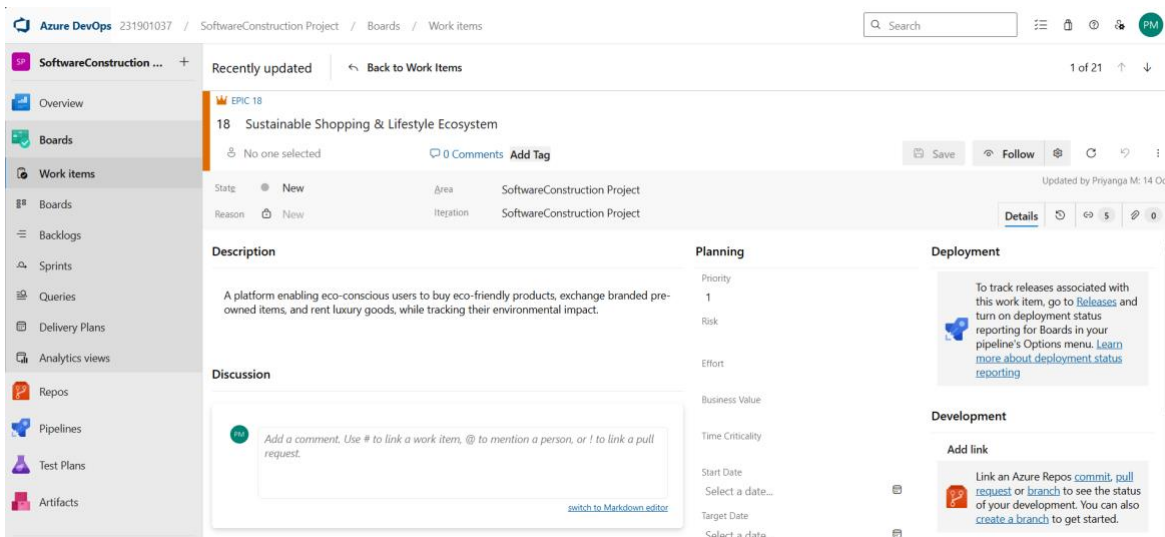
### Create Epic, Features, User Stories, Task



The screenshot shows the Azure DevOps interface for the 'SoftwareConstruction Project'. The 'Work items' section is active, displaying a list of items. The table below represents the data shown in the screenshot:

| ID | Title  | Assigned To       | State  | Area Path                    | Tags |
|----|--|-------------------|--------|------------------------------|------|
| 18 | Sustainable Shopping & Lifestyle Ecosystem                                 | Unassigned        | New    | SoftwareConstruction Project |      |
| 19 | Eco-Friendly Product Marketplace   | Unassigned        | New    | SoftwareConstruction Project |      |
| 20 | Circular Fashion & Pre-Owned Exchange                                      | Unassigned        | New    | SoftwareConstruction Project |      |
| 21 | Luxury Rental Marketplace  | Unassigned        | New    | SoftwareConstruction Project |      |
| 22 | As a user, I should be able to view a product's carbon footprint and ex    | Priyanga M        | New    | SoftwareConstruction Project |      |
| 23 | As a system, it should suggest greener alternatives if available.          | Saravanan MD      | New    | SoftwareConstruction Project |      |
| 24 | As a user, I should be able to list my branded pre-owned item for rese     | Guru sai charan D | New    | SoftwareConstruction Project |      |
| 25 | As a system, it should verify authenticity of uploaded branded items (/    | Priyanga M        | New    | SoftwareConstruction Project |      |
| 26 | As a user, I should be able to rent clothing or gadgets for a specific tir | Guru sai charan D | New    | SoftwareConstruction Project |      |
| 27 | As an admin, I should be able to manage rental logistics, return tracki    | Saravanan MD      | New    | SoftwareConstruction Project |      |
| 34 | User Login with valid credentials  | Priyanga M        | Design | SoftwareConstruction Project |      |
| 35 | New User Registration  | Priyanga M        | Design | SoftwareConstruction Project |      |

### 1. Fill in Epics



The screenshot shows the 'Details' view for Epic 18 in Azure DevOps. The epic is titled 'Sustainable Shopping & Lifestyle Ecosystem'. The description is: 'A platform enabling eco-conscious users to buy eco-friendly products, exchange branded pre-owned items, and rent luxury goods, while tracking their environmental impact.' The 'Planning' section shows a priority of 1. The 'Deployment' section provides instructions on how to track releases and link to development status reporting.

## 2.Fill in Features

The screenshot shows the Azure DevOps interface for a project named 'SoftwareConstruction Project'. The left sidebar contains navigation options: Overview, Boards, Work items, Backlogs, Sprints, Queries, Delivery Plans, Analytics views, Repos, Pipelines, Test Plans, and Artifacts. The main area displays a 'Feature' item titled 'Circular Fashion & Pre-Owned Exchange' (ID 20). The item is in 'New' state, with 'Reason' set to 'New', 'Area' as 'SoftwareConstruction Project', and 'Iteration' as 'SoftwareConstruction Project'. The 'Description' field is empty with a placeholder 'Click to add Description.'. The 'Discussion' section shows a comment box with a placeholder '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. The 'Planning' section includes fields for Priority (2), Risk, Effort, Business Value, Time Criticality, Start Date, and Target Date. The 'Deployment' section has a note about tracking releases and a link to 'Releases'. The 'Development' section has a note about linking Azure Repos and a link to 'Add link'.

## 3.Fill in User Story Details

The screenshot shows the Azure DevOps interface for a project named 'SoftwareConstruction Project'. The left sidebar contains navigation options: Overview, Boards, Work items, Backlogs, Sprints, Queries, Delivery Plans, Analytics views, Repos, Pipelines, Test Plans, and Artifacts. The main area displays a 'User Story' item titled 'As a user, I should be able to list my branded pre-owned item for resale/exchange.' (ID 24). The item is in 'New' state, with 'Reason' set to 'New', 'Area' as 'SoftwareConstruction Project', and 'Iteration' as 'SoftwareConstruction Project/Iteration 1'. The 'Description' field is empty with a placeholder 'Click to add Description.'. The 'Acceptance Criteria' section shows a text area with a placeholder 'I' and a 'switch to Markdown editor' link. The 'Planning' section includes fields for Story Points, Priority (3), and Risk. The 'Deployment' section has a note about tracking releases and a link to 'Releases'. The 'Development' section has a note about linking Azure Repos and a link to 'Add link'.

### Result:

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

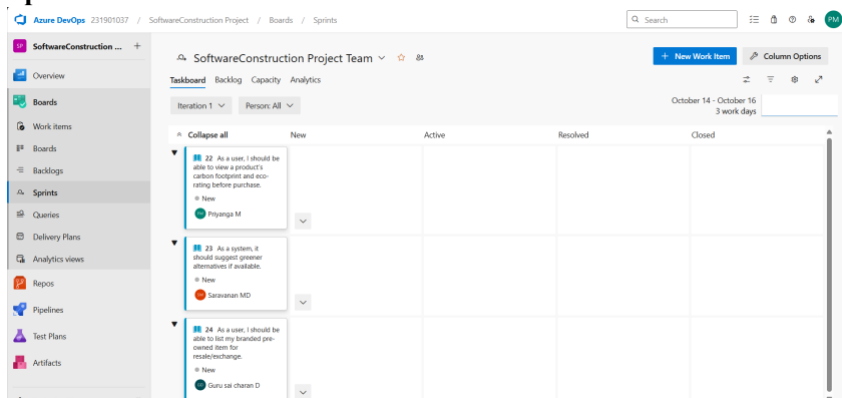
**EXP NO: 4**

# SPRINT PLANNING

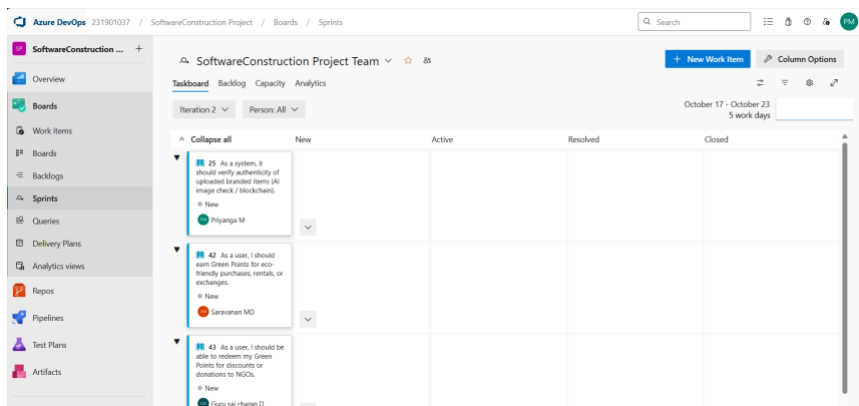
## Aim:

To assign user story to specific sprint for the Sustainable Shopping & Lifestyle Ecosystem project

## Sprint Planning Sprint 1



## Sprint 2



## Result:

The Sprints are created for the Sustainable Shopping & Lifestyle Ecosystem project.

EXP NO: 5

# POKER ESTIMATION

## Aim:

Create Poker Estimation for the user stories – Sustainable Shopping & Lifestyle Ecosystem Project.

## Poker Estimation

The screenshot displays the Azure DevOps interface for a project named 'SoftwareConstruction Project'. The left sidebar shows navigation options: Overview, Boards, Work items, Repos, Pipelines, Test Plans, and Artifacts. The main content area shows a 'Recently updated' list of work items. The selected work item is 'USER STORY 25' with the title 'As a system, it should verify authenticity of uploaded branded items (AI image check / blockchain). (5 points)'. The work item is assigned to 'Priyanga M' and is in the 'New' state. The interface includes tabs for Description, Planning, Deployment, Acceptance Criteria, Classification, and Development. The 'Description' tab is active, showing a placeholder for the description. The 'Planning' tab shows 'Story Points' as 3. The 'Deployment' tab shows a link to 'Learn more about deployment status reporting'. The 'Classification' tab shows 'Value area' as 'Business'. The 'Development' tab shows a link to 'Add link'.

## Result:

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

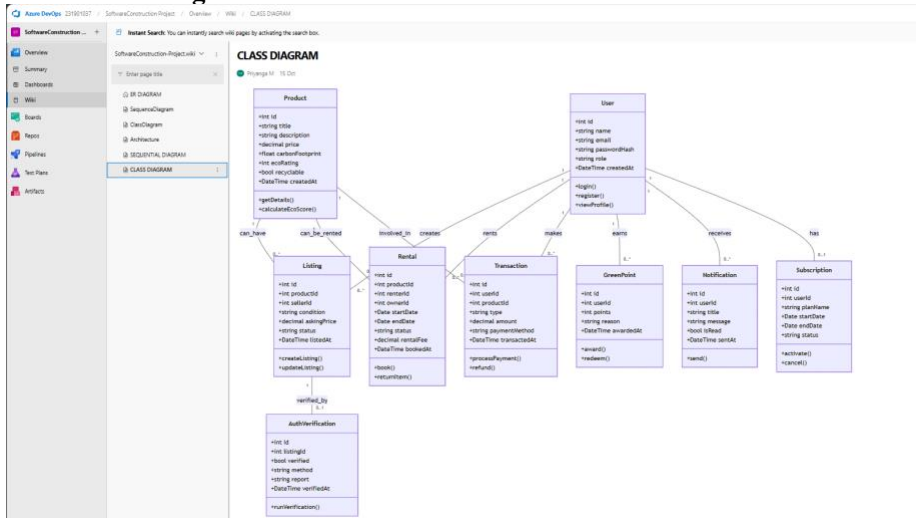
EXP NO: 6

## DESIGNING CLASS AND SEQUENCE DIAGRAMS FOR PROJECT ARCHITECTURE

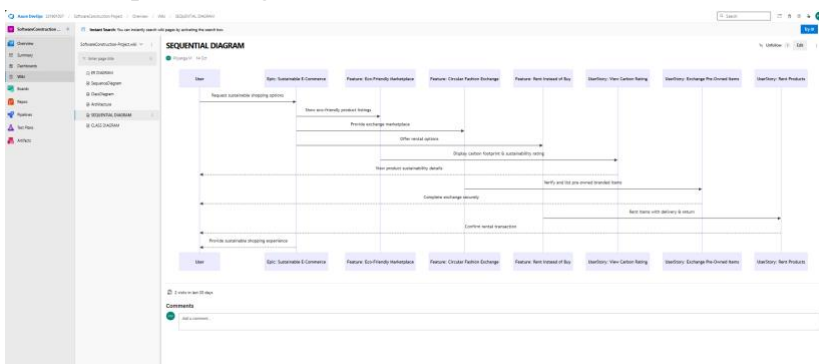
**Aim:**

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 the Sustainable Shopping & Lifestyle Ecosystem

## DESIGNING ARCHITECTURAL AND ER DIAGRAMS FOR PROJECT STRUCTURE

To Design an Architectural Diagram and ER Diagram for the given Project.

[illegible]

The screenshot shows the ER Diagram tool interface with a database schema for a rental system. The schema includes the following tables and their attributes:

- PRODUCT** (PK: ID, FK: CATEGORY\_ID)
  - ID (PK)
  - NAME
  - DESCRIPTION
  - STATUS
  - TYPE
  - PRICE
  - RENTAL\_PRICE
  - RENTAL\_DURATION
  - RENTAL\_CONDITION
  - RENTAL\_RETURN\_DATE
  - RENTAL\_RETURN\_TIME
  - RENTAL\_RETURN\_LOCATION
  - RENTAL\_RETURN\_STATUS
  - RENTAL\_RETURN\_COMMENT
  - RENTAL\_RETURN\_DATE\_TIME
  - RENTAL\_RETURN\_LOCATION\_TIME
  - RENTAL\_RETURN\_STATUS\_TIME
  - RENTAL\_RETURN\_COMMENT\_TIME
  - RENTAL\_RETURN\_DATE\_TIME\_LOCATION
  - RENTAL\_RETURN\_STATUS\_TIME\_LOCATION
  - RENTAL\_RETURN\_COMMENT\_TIME\_LOCATION
  - RENTAL\_RETURN\_DATE\_TIME\_LOCATION\_STATUS
  - RENTAL\_RETURN\_STATUS\_TIME\_LOCATION\_STATUS
  - RENTAL\_RETURN\_COMMENT\_TIME\_LOCATION\_STATUS
- USER** (PK: ID, FK: CATEGORY\_ID)
  - ID (PK)
  - NAME
  - DESCRIPTION
  - STATUS
  - TYPE
  - PRICE
  - RENTAL\_PRICE
  - RENTAL\_DURATION
  - RENTAL\_CONDITION
  - RENTAL\_RETURN\_DATE
  - RENTAL\_RETURN\_TIME
  - RENTAL\_RETURN\_LOCATION
  - RENTAL\_RETURN\_STATUS
  - RENTAL\_RETURN\_COMMENT
  - RENTAL\_RETURN\_DATE\_TIME
  - RENTAL\_RETURN\_LOCATION\_TIME
  - RENTAL\_RETURN\_STATUS\_TIME
  - RENTAL\_RETURN\_COMMENT\_TIME
  - RENTAL\_RETURN\_DATE\_TIME\_LOCATION
  - RENTAL\_RETURN\_STATUS\_TIME\_LOCATION
  - RENTAL\_RETURN\_COMMENT\_TIME\_LOCATION
  - RENTAL\_RETURN\_DATE\_TIME\_LOCATION\_STATUS
  - RENTAL\_RETURN\_STATUS\_TIME\_LOCATION\_STATUS
  - RENTAL\_RETURN\_COMMENT\_TIME\_LOCATION\_STATUS
- RENTAL** (PK: ID, FK: CATEGORY\_ID)
  - ID (PK)
  - NAME
  - DESCRIPTION
  - STATUS
  - TYPE
  - PRICE
  - RENTAL\_PRICE
  - RENTAL\_DURATION
  - RENTAL\_CONDITION
  - RENTAL\_RETURN\_DATE
  - RENTAL\_RETURN\_TIME
  - RENTAL\_RETURN\_LOCATION
  - RENTAL\_RETURN\_STATUS
  - RENTAL\_RETURN\_COMMENT
  - RENTAL\_RETURN\_DATE\_TIME
  - RENTAL\_RETURN\_LOCATION\_TIME
  - RENTAL\_RETURN\_STATUS\_TIME
  - RENTAL\_RETURN\_COMMENT\_TIME
  - RENTAL\_RETURN\_DATE\_TIME\_LOCATION
  - RENTAL\_RETURN\_STATUS\_TIME\_LOCATION
  - RENTAL\_RETURN\_COMMENT\_TIME\_LOCATION
  - RENTAL\_RETURN\_DATE\_TIME\_LOCATION\_STATUS
  - RENTAL\_RETURN\_STATUS\_TIME\_LOCATION\_STATUS
  - RENTAL\_RETURN\_COMMENT\_TIME\_LOCATION\_STATUS
- TRANSACTION** (PK: ID, FK: CATEGORY\_ID)
  - ID (PK)
  - NAME
  - DESCRIPTION
  - STATUS
  - TYPE
  - PRICE
  - RENTAL\_PRICE
  - RENTAL\_DURATION
  - RENTAL\_CONDITION
  - RENTAL\_RETURN\_DATE
  - RENTAL\_RETURN\_TIME
  - RENTAL\_RETURN\_LOCATION
  - RENTAL\_RETURN\_STATUS
  - RENTAL\_RETURN\_COMMENT
  - RENTAL\_RETURN\_DATE\_TIME
  - RENTAL\_RETURN\_LOCATION\_TIME
  - RENTAL\_RETURN\_STATUS\_TIME
  - RENTAL\_RETURN\_COMMENT\_TIME
  - RENTAL\_RETURN\_DATE\_TIME\_LOCATION
  - RENTAL\_RETURN\_STATUS\_TIME\_LOCATION
  - RENTAL\_RETURN\_COMMENT\_TIME\_LOCATION
  - RENTAL\_RETURN\_DATE\_TIME\_LOCATION\_STATUS
  - RENTAL\_RETURN\_STATUS\_TIME\_LOCATION\_STATUS
  - RENTAL\_RETURN\_COMMENT\_TIME\_LOCATION\_STATUS
- BOOKING\_POINT** (PK: ID, FK: CATEGORY\_ID)
  - ID (PK)
  - NAME
  - DESCRIPTION
  - STATUS
  - TYPE
  - PRICE
  - RENTAL\_PRICE
  - RENTAL\_DURATION
  - RENTAL\_CONDITION
  - RENTAL\_RETURN\_DATE
  - RENTAL\_RETURN\_TIME
  - RENTAL\_RETURN\_LOCATION
  - RENTAL\_RETURN\_STATUS
  - RENTAL\_RETURN\_COMMENT
  - RENTAL\_RETURN\_DATE\_TIME
  - RENTAL\_RETURN\_LOCATION\_TIME
  - RENTAL\_RETURN\_STATUS\_TIME
  - RENTAL\_RETURN\_COMMENT\_TIME
  - RENTAL\_RETURN\_DATE\_TIME\_LOCATION
  - RENTAL\_RETURN\_STATUS\_TIME\_LOCATION
  - RENTAL\_RETURN\_COMMENT\_TIME\_LOCATION
  - RENTAL\_RETURN\_DATE\_TIME\_LOCATION\_STATUS
  - RENTAL\_RETURN\_STATUS\_TIME\_LOCATION\_STATUS
  - RENTAL\_RETURN\_COMMENT\_TIME\_LOCATION\_STATUS
- NOTIFICATION** (PK: ID, FK: CATEGORY\_ID)
  - ID (PK)
  - NAME
  - DESCRIPTION
  - STATUS
  - TYPE
  - PRICE
  - RENTAL\_PRICE
  - RENTAL\_DURATION
  - RENTAL\_CONDITION
  - RENTAL\_RETURN\_DATE
  - RENTAL\_RETURN\_TIME
  - RENTAL\_RETURN\_LOCATION
  - RENTAL\_RETURN\_STATUS
  - RENTAL\_RETURN\_COMMENT
  - RENTAL\_RETURN\_DATE\_TIME
  - RENTAL\_RETURN\_LOCATION\_TIME
  - RENTAL\_RETURN\_STATUS\_TIME
  - RENTAL\_RETURN\_COMMENT\_TIME
  - RENTAL\_RETURN\_DATE\_TIME\_LOCATION
  - RENTAL\_RETURN\_STATUS\_TIME\_LOCATION
  - RENTAL\_RETURN\_COMMENT\_TIME\_LOCATION
  - RENTAL\_RETURN\_DATE\_TIME\_LOCATION\_STATUS
  - RENTAL\_RETURN\_STATUS\_TIME\_LOCATION\_STATUS
  - RENTAL\_RETURN\_COMMENT\_TIME\_LOCATION\_STATUS
- SUBSCRIPTION** (PK: ID, FK: CATEGORY\_ID)
  - ID (PK)
  - NAME
  - DESCRIPTION
  - STATUS
  - TYPE
  - PRICE
  - RENTAL\_PRICE
  - RENTAL\_DURATION
  - RENTAL\_CONDITION
  - RENTAL\_RETURN\_DATE
  - RENTAL\_RETURN\_TIME
  - RENTAL\_RETURN\_LOCATION
  - RENTAL\_RETURN\_STATUS
  - RENTAL\_RETURN\_COMMENT
  - RENTAL\_RETURN\_DATE\_TIME
  - RENTAL\_RETURN\_LOCATION\_TIME
  - RENTAL\_RETURN\_STATUS\_TIME
  - RENTAL\_RETURN\_COMMENT\_TIME
  - RENTAL\_RETURN\_DATE\_TIME\_LOCATION
  - RENTAL\_RETURN\_STATUS\_TIME\_LOCATION
  - RENTAL\_RETURN\_COMMENT\_TIME\_LOCATION
  - RENTAL\_RETURN\_DATE\_TIME\_LOCATION\_STATUS
  - RENTAL\_RETURN\_STATUS\_TIME\_LOCATION\_STATUS
  - RENTAL\_RETURN\_COMMENT\_TIME\_LOCATION\_STATUS
- AUTHORIZATION** (PK: ID, FK: CATEGORY\_ID)
  - ID (PK)
  - NAME
  - DESCRIPTION
  - STATUS
  - TYPE
  - PRICE
  - RENTAL\_PRICE
  - RENTAL\_DURATION
  - RENTAL\_CONDITION
  - RENTAL\_RETURN\_DATE
  - RENTAL\_RETURN\_TIME
  - RENTAL\_RETURN\_LOCATION
  - RENTAL\_RETURN\_STATUS
  - RENTAL\_RETURN\_COMMENT
  - RENTAL\_RETURN\_DATE\_TIME
  - RENTAL\_RETURN\_LOCATION\_TIME
  - RENTAL\_RETURN\_STATUS\_TIME
  - RENTAL\_RETURN\_COMMENT\_TIME
  - RENTAL\_RETURN\_DATE\_TIME\_LOCATION
  - RENTAL\_RETURN\_STATUS\_TIME\_LOCATION
  - RENTAL\_RETURN\_COMMENT\_TIME\_LOCATION
  - RENTAL\_RETURN\_DATE\_TIME\_LOCATION\_STATUS
  - RENTAL\_RETURN\_STATUS\_TIME\_LOCATION\_STATUS
  - RENTAL\_RETURN\_COMMENT\_TIME\_LOCATION\_STATUS

The diagram shows relationships between these tables, including one-to-many and many-to-many relationships, with various cardinalities and constraints.

CS23432

**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**

**Project: Sustainable Commerce Platform (GreenLoop)**

**1. Understand Core Features of the Application**

The Sustainable Commerce Platform enables eco-conscious consumers to shop, rent, exchange, and track their carbon footprint. The major features under test include:

- User Authentication and Profile Management
- Product Search, Filter, and View
- Add to Cart and Checkout Process
- Rent or Exchange Product Workflow
- Carbon Footprint Tracking
- Reward Points and Sustainability Badges

**2. Define User Interactions**

Each test case is designed to simulate real-world user actions across multiple modules:

- Logging in or registering with valid and invalid data
- Searching and filtering sustainable products
- Adding or removing items from the cart
- Completing checkout with valid and invalid payment details
- Renting or exchanging products
- Viewing the sustainability dashboard and carbon footprint statistics
- Redeeming eco-reward points

**3. Design Happy Path Test Cases**

Verify that all modules function correctly under normal conditions.

| <b>Test ID</b> | <b>Test Description</b>                         |
|----------------|---|
| TC01           | Register with valid user details                |
| TC02           | Login with valid credentials                    |
| TC03           | Search product by name and category             |
| TC04           | Filter products by sustainability rating        |
| TC05           | Add product to cart successfully                |
| TC06           | Checkout with valid payment details             |
| TC07           | Request a product for rent successfully         |
| TC08           | Exchange product with correct conditions        |
| TC09           | Track carbon footprint correctly after purchase |
| TC10           | Redeem reward points for eco-friendly actions   |

#### 4. Design Error Path Test Cases

Simulate invalid data or user mistakes to verify system resilience and validation.

| Test ID | Test Description                                      |
|---------|---|
| TC11    | Register with missing required fields                 |
| TC12    | Login with invalid password                           |
| TC13    | Search for unavailable product                        |
| TC14    | Add out-of-stock product to cart                      |
| TC15    | Checkout with expired card details                    |
| TC16    | Rent product with invalid duration input              |
| TC17    | Exchange request without uploading proof of ownership |
| TC18    | Access dashboard without login                        |
| TC19    | Attempt to redeem more points than available          |
| TC20    | Upload invalid file format for exchange product       |

#### 5. Break Down Steps and Expected Results

Each test case includes:

- **Preconditions:** For example, user is logged in or product available in catalog.
- **Test Steps:** Sequential actions such as "Click Login", "Enter Details", "Click Submit".
- **Expected Result:**
  - Success message or correct data displayed.
  - Error message displayed for invalid inputs.

##### **Example:**

##### **TC06 – Checkout with Valid Payment Details**

- Steps:
  1. Add a product to the cart.
  2. Go to checkout page.
  3. Enter valid card details.
  4. Click “Confirm Payment”.
- Expected Result:
  - Payment succeeds.
  - Confirmation message displayed.
  - Order ID generated.

#### 6. Use Clear Naming and IDs

All test cases follow consistent naming for easy mapping in Azure Test Plans.

**Format:** TCXX – Module – Action

Example: TC04 – Product Filter – By Sustainability Rating

#### 7. Separate Test Suites

Organized based on core functionalities for modular testing and easier tracking in Azure DevOps Test Plans.

| Test Suite                  | Modules Covered      |
|-----------------------------|----------------------|
| Login and Registration      | TC01–TC02, TC11–TC12 |
| Product Search and Filter   | TC03–TC04, TC13      |
| Cart and Checkout           | TC05–TC06, TC14–TC15 |
| Rent and Exchange           | TC07–TC08, TC16–TC17 |
| Carbon Tracking and Rewards | TC09–TC10, TC18–TC20 |

#### 8. Prioritize and Review

- **High Priority:** Login, Checkout, Rent/Exchange, and Reward Redemption
- **Medium Priority:** Product Search, Filter, and Carbon Tracking
- **Low Priority:** Profile Updates and UI Visuals

All test cases were reviewed for:

- Complete feature coverage
- Traceability to user stories
- Accurate validation steps
- Reusability for automation

## 1. New test plan

NEW TEST CASE \*

Register with valid user details

Priyanga M 0 Comments Add Tag Save and Close

State Design Area SoftwareConstruction Project Reason New Iteration SoftwareConstruction Project

Steps Summary Associated Automation

Steps

| Steps                            | Action | Expected result |
|----------------------------------|--------|-----------------|
| Click or type here to add a step |        |                 |

Recent test results

Run manual tests with web runner. [Learn how to get started](#)

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

## 2. Test case

Test Suite TS01 – User Authentication

User Story ID: US01

Title: As a user, I want to securely sign up and log in so that I can access the platform and manage eco-commerce activities.

Test Case TC01 – Successful Sign Up

Type: Happy Path

Steps:

1. Navigate to the Sign-Up page.
2. Enter valid name, email, and password.
3. Click on Sign Up.  
Expected Result: Account is created and user is redirected to the homepage/dashboard.

Test Case TC02 – Sign Up with Existing Email

Type: Error Path

Steps:

1. Navigate to the Sign-Up page.
2. Enter an already registered email.

3. Click on Sign Up.  
Expected Result: Error message displayed — “Email is already registered.”

#### Test Suite TS02 – Product Browsing and Filtering

User Story ID: US02

Title: As a user, I want to search and filter sustainable products to find items based on eco-ratings and categories.

Test Case TC03 – Successful Product Search

Type: Happy Path

Steps:

1. Log in to the platform.
2. Enter a valid product name in the search bar.
3. Click on Search.  
Expected Result: Matching products are displayed with name, image, and eco-rating.

Test Case TC04 – Filter Products by Sustainability Rating

Type: Happy Path

Steps:

1. Log in to the platform.
2. Open product listing page.
3. Select filter “Sustainability Rating  $\geq 4$ ”.  
Expected Result: Products with sustainability rating 4 and above are displayed.

Test Case TC05 – Search Product Not Found

Type: Error Path

Steps:

1. Enter an invalid or unavailable product name.
2. Click on Search.  
Expected Result: Message displayed — “No products found.”

#### Test Suite TS03 – Cart and Checkout Process

User Story ID: US03

Title: As a user, I want to add items to the cart and checkout securely.

Test Case TC06 – Add Product to Cart

Type: Happy Path

Steps:

1. Log in.
2. Search and open a product.
3. Click on Add to Cart.

Expected Result: Product added to cart and confirmation message shown.

Test Case TC07 – Checkout with Valid Payment Details

Type: Happy Path

Steps:

1. Go to the Cart page.
2. Click on Checkout.
3. Enter valid payment details.
4. Click on Pay Now.

Expected Result: Payment succeeds and order confirmation is displayed.

Test Case TC08 – Checkout with Invalid Card

Type: Error Path

Steps:

1. Go to Checkout.
2. Enter expired card details.
3. Click on Pay Now.

Expected Result: Payment fails and an error message is shown — “Invalid card details.”

Test Suite TS04 – Rent and Exchange Workflow

User Story ID: US04

Title: As a user, I want to rent or exchange items with other users for sustainable consumption.

Test Case TC09 – Rent Product Successfully

Type: Happy Path

Steps:

1. Log in.
2. Select a rentable product.
3. Choose rental duration.
4. Click on Rent Now.

Expected Result: Rental request is created and confirmation displayed.

Test Case TC10 – Exchange Product Successfully

Type: Happy Path

Steps:

1. Log in.
2. Go to Exchange Section.
3. Select item for exchange and upload ownership proof.
4. Click on Submit Request.

Expected Result: Exchange request submitted successfully.

Test Case TC11 – Exchange Without Proof Document

Type: Error Path

Steps:

1. Log in.
2. Try to submit exchange request without uploading proof.  
Expected Result: Error message shown — “Ownership proof required.”

Test Suite TS05 – Carbon Footprint and Rewards

User Story ID: US05

Title: As a user, I want to track my carbon footprint and earn reward points for sustainable purchases.

Test Case TC12 – Carbon Footprint Tracking After Purchase

Type: Happy Path

Steps:

1. Log in.
2. Complete a product purchase.
3. Go to Carbon Dashboard.  
Expected Result: Dashboard updates showing CO<sub>2</sub> saved and footprint reduced.

Test Case TC13 – Redeem Reward Points

Type: Happy Path

Steps:

1. Log in.
2. Go to Rewards Section.
3. Redeem available points for a voucher.  
Expected Result: Points redeemed and voucher generated.

Test Case TC14 – Redeem Insufficient Points

Type: Error Path

Steps:

1. Log in.
2. Try to redeem more points than available.  
Expected Result: Message displayed — “Not enough points to redeem.”

Test Suite TS06 – System Access and Alert

User Story ID: US06

Title: As a user, I want to receive notifications for rent expiry, delivery status, and account alerts.

Test Case TC15 – Receive Rent Expiry Alert

Type: Happy Path

Steps:

1. Log in.
2. Check notifications for rented items nearing end date.  
Expected Result: Alert displayed — “Your rental period expires in 2 days.”

Test Case TC16 – No Alert for Completed Rentals

Type: Error Path

Steps:

1. Log in after rental period has expired.  
Expected Result: No alert shown; status updated as “Completed.”

## Test Cases

The screenshot shows the Azure DevOps Test Plans interface. The left sidebar contains navigation options: Overview, Boards, Repos, Pipelines, Test Plans (selected), Test plans, Progress report, Parameters, Configurations, Runs, and Artifacts. The main area displays the 'Sustainable Comm...' project with a 'Past' status and a 'View report' link. Below this, the 'Test Suites' section shows a filter by name and a list of suites: 'Sustainable Commerce Platform - Fu...', 'Product Search & Filter (2)', and 'Login & Registration (3)'. The 'Login & Registration (ID: 33)' suite is selected, showing a table of test cases:

| Test Cases (3 items)                                       | Order | Test Case Id |
|--|-------|--------------|
| <input type="checkbox"/> Title                             |       |              |
| <input type="checkbox"/> User Login with valid credentials | 1     | 34           |
| <input type="checkbox"/> New User Registration             | 2     | 35           |
| <input type="checkbox"/> User Login with invalid password  | 3     | 39           |

## 1. Installation of test


The screenshot shows the Microsoft Edge Add-ons page for the 'Test & Feedback' extension by Microsoft Corporation. The extension is featured and has a rating of 4.5 stars from 29 reviews, with over 1,300,000 users. It is categorized as a 'Developer tool'. The page includes a video thumbnail titled 'Azure Test Plans: Introducing Test & Feedback' and a description: 'Now everyone on the team can own quality. Capture findings, create issues, and collaborate with the team, directly from the browser.' The right sidebar shows the version (1.0.2561), update date (6 May 2025), and availability in 1 language. Links for 'Terms', 'Privacy policy', and 'Developer' information are also present.

Click to go back, hold to see history

Microsoft Edge Add-ons

DiscoverExtensionsThemes

Search extensions, themes, and more



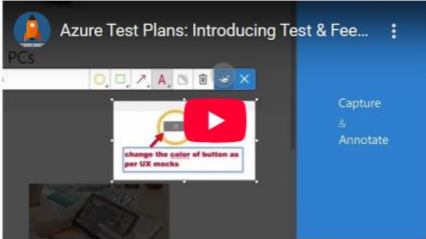
Test & Feedback

Extension | Microsoft Corporation

★★★★☆ (29) | 1,30,000+ Users | Developer tools

Remove

Add-on already installed on your browser



Version 1.0.256.1

Updated 6 May 2025

Available in 1 language

Terms & Privacy policy

Developer: More add-ons from Microsoft Corporation (958)

Report abuse

Description

Now everyone on the team can own quality. Capture findings, create issues, and collaborate with the team, directly from the browser.

Test & Feedback - Now everyone on the team can own quality. Capture findings, create issues, and collaborate with the team, directly from the browser.

## Test and feedback

### Showing it as an extension

Projects - HomeR2023-CSE-CS-Curric...Azure DevOps - Micro...My InformationTest Plan 31 Sustainable...test and feedback ext...Test & Feedback - Mi...

https://dev.azure.com/231901037/SoftwareConstruction%20Project/\_testPlans/define?planId=31&suiteId=33

SoftwareConstruction ...

OverviewBoardsReposPipelinesTest PlansTest plansProgress reportParametersConfigurationsRunsArtifacts

Sustainable Comm...

Oct 15 - Oct 22Past100% run, 100% passedView report

Test Suites

Filter suites by name

Sustainable Commerce Platform - Fu...

- Product Search & Filter (2)
- Login & Registration (3)

Login & Registration (ID: 33)

Sustainable Commerce Platform - Functional & Performance Testing

DefineExecuteChart

Test Cases (3 items)

|  | Order | Test Case Id |
|--|-------|--------------|
| <input type="checkbox"/> Title                             |       |              |
| <input type="checkbox"/> User Login with valid credentials | 1     | 34           |
| <input type="checkbox"/> New User Registration             | 2     | 35           |
| <input type="checkbox"/> User Login with invalid password  | 3     | 39           |

Extensions

Test & Feedback

Manage extensions

Get extensions for Microsoft Edge

## 1. Running the test cases

Login & Registration (ID: 33)

Define Execute Chart

Help

| Test Points (3 items)   |         |       |              |        |               |                |  |
|---|---------|-------|--------------|--------|---------------|----------------|--|
| Title   | Outcome | Order | Test Case Id | State  | Configuration | Current Tester |  |
| <input checked="" type="checkbox"/> User Login with valid credentials | Passed  | 1     | 34           | Design | Windows 10    | Priyanga M     |  |
| <input type="checkbox"/> New User Registration                        | Passed  | 2     | 35           | Design | Windows 10    | Priyanga M     |  |
| <input type="checkbox"/> User Login with invalid password             | Passed  | 3     | 39           | Design | Windows 10    | Priyanga M     |  |

View execution history

Mark Outcome

Run

Reset test to active

Edit test case

Assign tester

View test result

Runner - Test Plans - Profile 3 - Microsoft Edge

https://dev.azure.com/231901037/SoftwareConstruction%20Project/\_testExecution/Index

Save and clo... Create bug

34: User Login with valid credentials

- Go to login page
- Enter valid email & password
- Click Login

EXPECTED RESULT  
User should be redirected to dashboard/home page

## 1. Recording the test case

Runner - Test Plans - Profile 3 - Microsoft Edge

https://dev.azure.com/231901037/SoftwareConstruction%20Project/\_testExecution/Index

Save and clo... Create bug

Recording: 00:00 / 10:00 minutes

Stop

34: User Login with valid credentials

- Go to login page
- Enter valid email & password
- Click Login

EXPECTED RESULT  
User should be redirected to dashboard/home page

Choose what to share with Test & Feedback

The site will be able to see the contents of your screen

Microsoft Edge Tab Window Entire Screen

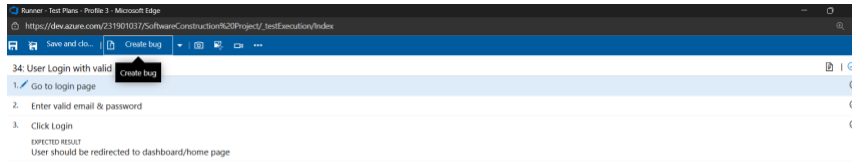
Entire screen

Share Cancel

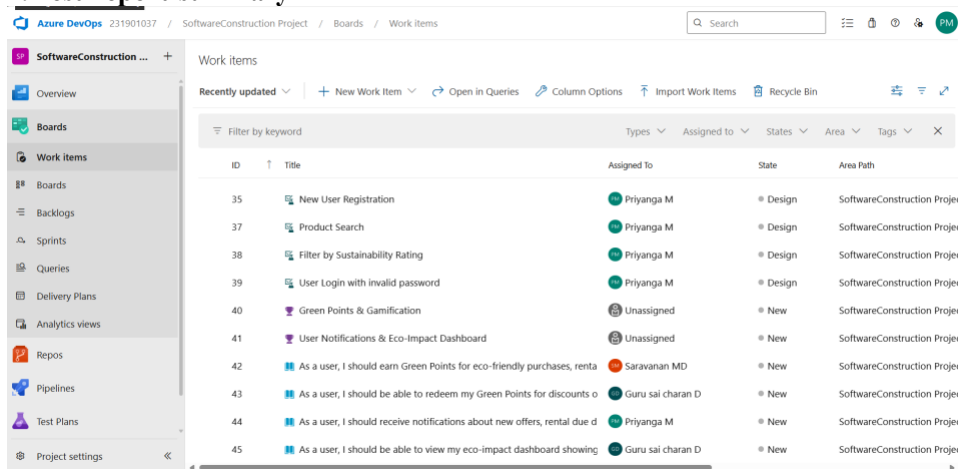
## 2. Creating the bug

2116231901037

CS23432

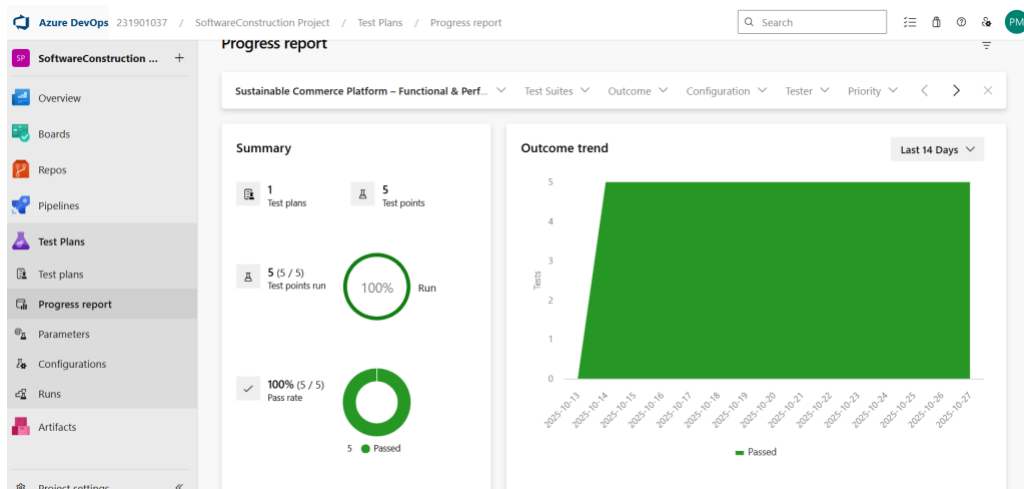


## 1. Test report summary



- Assigning bug to the developer and changing state

## 2. Progress report



## 1. Changing the test template

2116231901037

CS23432



**EXP NO: 9**

## **LOAD TESTING AND PIPELINES**

### **Aim:**

To create an Azure Load Testing resource and run a load test to evaluate the performance of a target endpoint and to create and demonstrate an Azure DevOps pipeline for automating application builds, tests, and deployment.

### **Load Testing**

Azure Load Testing:

Azure Load Testing allows you to simulate high traffic and stress tests for your web applications and APIs to understand how they perform under load. It helps identify performance bottlenecks, scalability issues, and optimize resource usage before deployment.

### **Steps to Create an Azure Load Testing Resource:**

Before you run your first test, you need to create the Azure Load Testing resource:

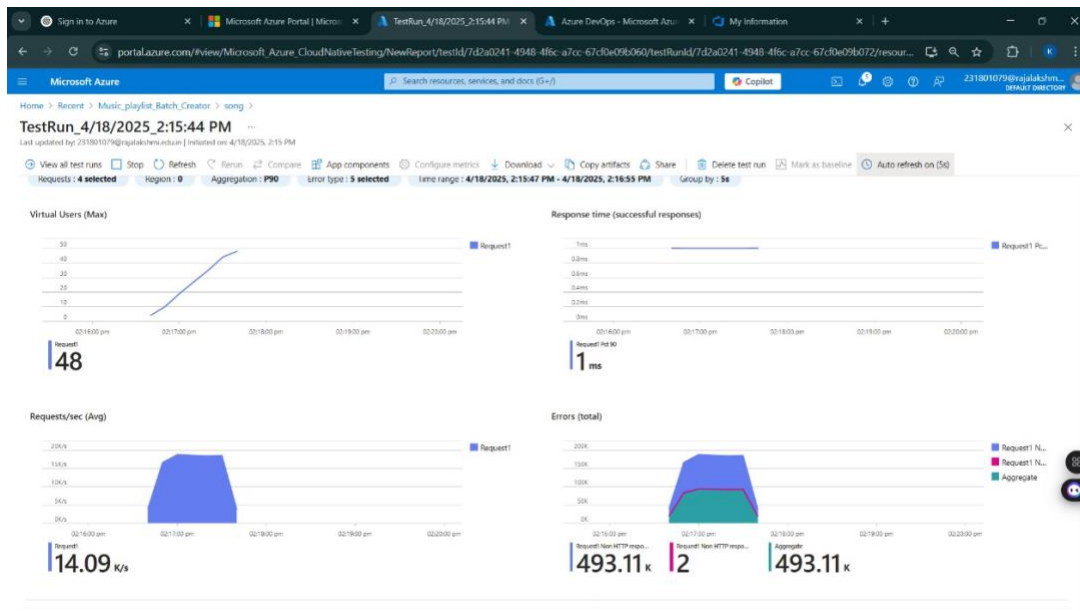
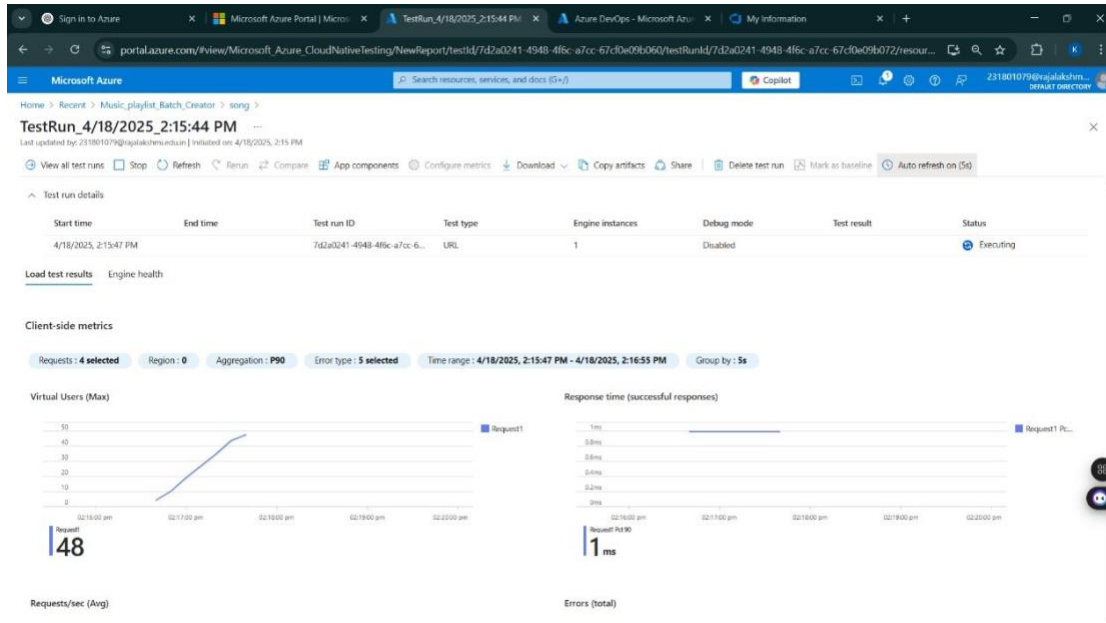
1. Sign in to Azure Portal  
Go to <https://portal.azure.com> and log in.
2. Create the Resource
  - Go to *Create a resource* → Search for “Azure Load Testing”.
  - Select Azure Load Testing and click Create.
3. Fill in the Configuration Details
  - *Subscription*: Choose your Azure subscription.
  - *Resource Group*: Create new or select an existing one.
  - *Name*: Provide a unique name (no special characters).
  - *Location*: Choose the region for hosting the resource.
4. (Optional) Configure tags for categorization and billing.
5. Click Review + Create, then Create.
6. Once deployment is complete, click Go to resource.

### **Steps to Create and Run a Load Test:**

Once your resource is ready:

1. Go to your Azure Load Testing resource and click Add HTTP requests > Create.
2. Basics Tab
  - *Test Name*: Provide a unique name.
  - *Description*: (Optional) Add test purpose.
  - *Run After Creation*: Keep checked.
3. Load Settings
  - *Test URL*: Enter the target endpoint (e.g., <https://yourapi.com/products>).
4. Click Review + Create → Create to start the test.

## Load Testing



## Pipelines

### Description:

This experiment demonstrates how to connect a GitHub-hosted Flask-based music recommendation project with Azure DevOps. The pipeline will automatically install dependencies, run basic tests, and publish artifacts. This ensures that every commit triggers checks for reliability and smooth deployment.

### Steps:

1. Connect GitHub to Azure DevOps:
  - In Azure DevOps, create a new project.
  - Create a pipeline and select GitHub as the source.
  - Authorize access to your GitHub repository, ensuring that Azure DevOps can pull the repository for your pipeline.
2. Create azure-pipelines.yml in Your Repo Root:
  - In your GitHub repository, create a new file called azure-pipelines.yml in the root directory.
  - Add the following basic pipeline configuration for Python and Flask:

### yml Code

trigger:


- main

pool:

vmImage: 'ubuntu-latest'

steps:

- checkout: self

- script: echo " Azure Pipeline for Contract Management System is running!"  
displayName: 'Run sample script'

3. Pipeline Tasks Include:
  - Setting up the Python environment using the UsePythonVersion task.
  - Installing project dependencies from project/requirements.txt. Make sure the path to requirements.txt is correct (it is located under the project folder).
  - Running a simple Python script to verify that Python is set up correctly and the pipeline works.
4. Run and Monitor Pipeline:
  - Commit changes to the main branch of your repository to trigger the pipeline in Azure DevOps.
  - Monitor the logs in the Azure DevOps portal to view logs, errors, or success messages and ensure everything runs smoothly.

## Pipeline

ict Mjanagement Spstern > Pipelines > Balajinamesh120 > 20260504.1

Search



88

86

Summary Code Coverage



### Individuual CI

View change

Repository and version

Balajinamesh120

b3e89175610ffdr<125342755QFCS4F7A4C > C

main

Time started and set at

Today at 11:14 AM

Duration

2m 8's



0 errors



No hosted parallelism has been purchased or granted. To request a free parallelism grant, please fill out the following form: <https://aka.ms/azpipelines-parallelism-parallelism-request:20260504.1>

Disable this check by following trackbuild runs

### Jobs

| Job | Status    | Status    |
|-----|-----------|-----------|
| Job | Succeeded | Succeeded |

### Result:

Successfully created the Azure Load Testing resource and executed a load test to assess the performance of the specified endpoint and also demonstrated pipelines in azure devops.

2116231901037

CS23432

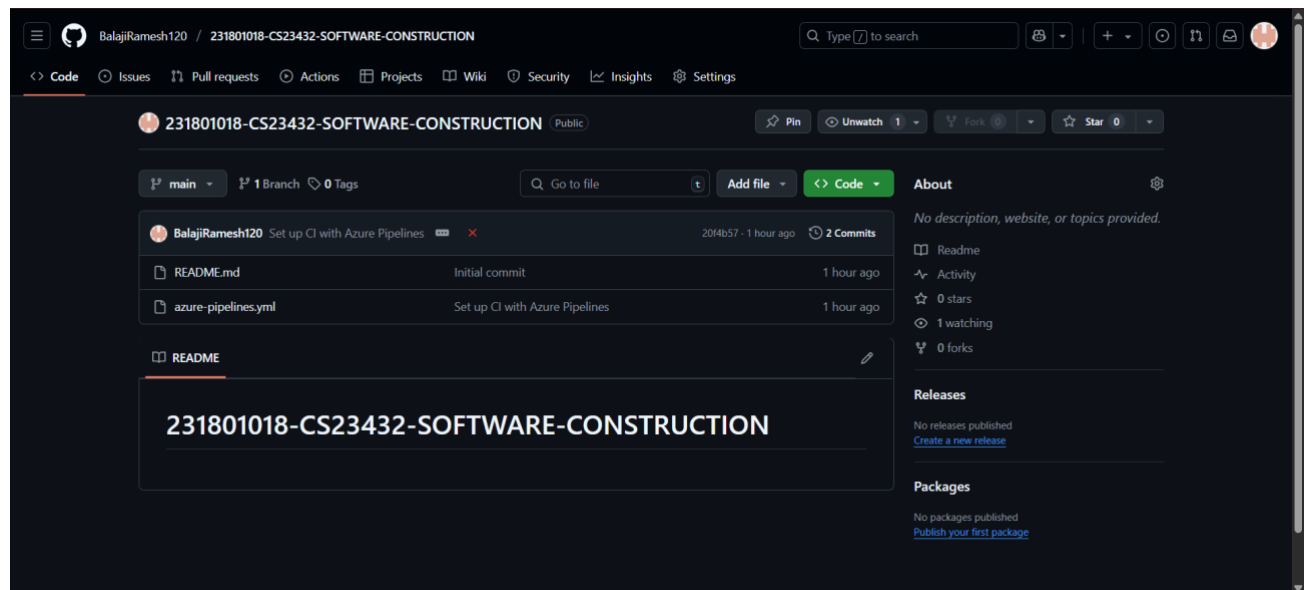
EXP NO: 10

## 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 Contract 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.















































