

Expr no 1

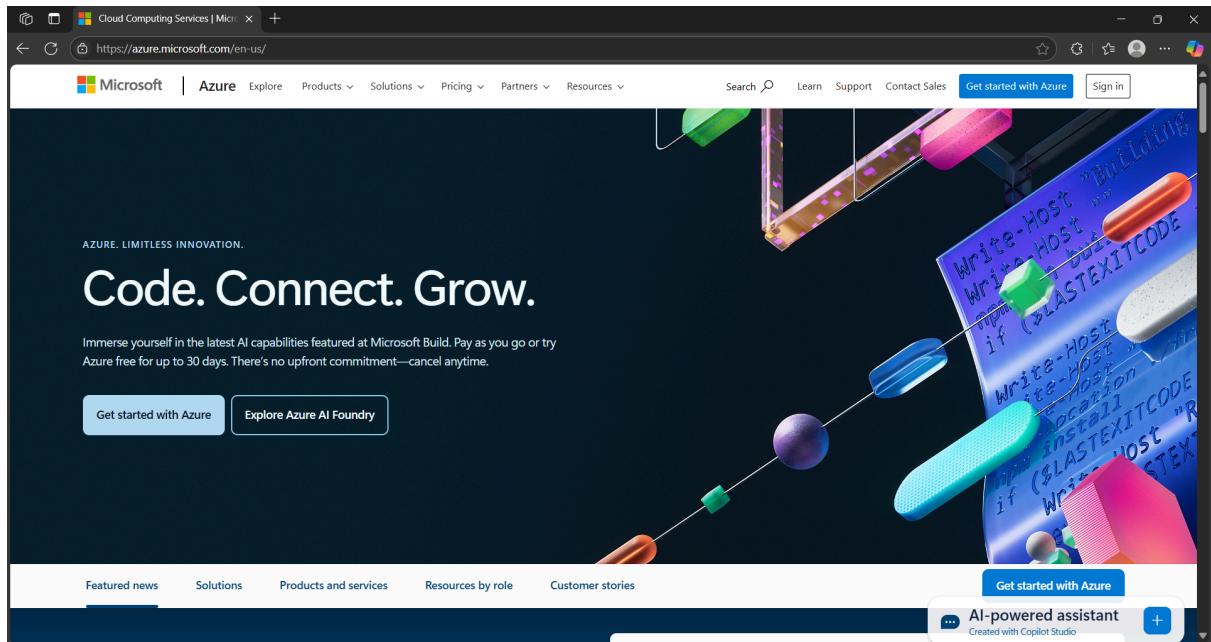
Azure DevOps Environment Setup

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>



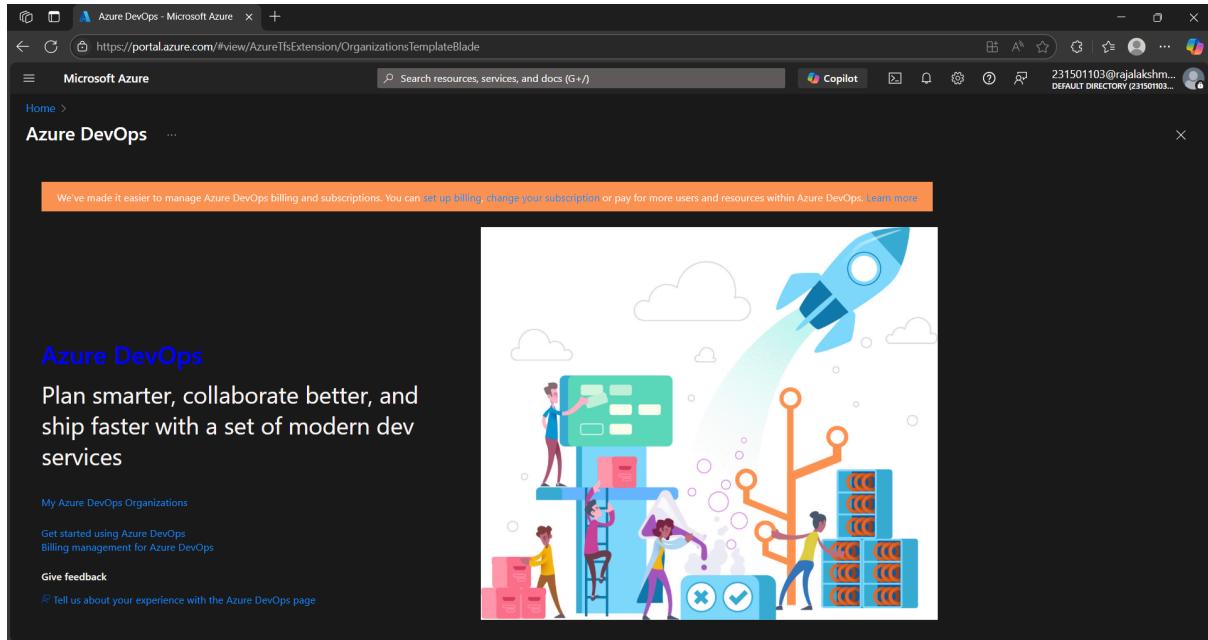
2. Azure Home Page

The screenshot shows the Microsoft Azure Home Page. At the top, there's a search bar with the placeholder "Search resources, services, and docs (G+)" and a Copilot icon. Below the search bar is a navigation bar with links for "Create a resource", "Azure DevOps organizations", "Azure Load Testing", "All resources", "Subscriptions", "Monitor", "Virtual networks", "Storage accounts", "Load balancers", and "More services". The main area is titled "Azure services" and contains sections for "Resources" (Recent and Favorite), "Navigate" (Subscriptions, Resource groups, All resources, Dashboard), and "Tools". A sidebar on the left lists recent resources: harisht143 (Azure Load Testing), Monish_69 (Resource group), and Azure for Students (Subscription). A "Last Viewed" section on the right shows items from two weeks ago and four months ago.

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

This screenshot is similar to the previous one, showing the Microsoft Azure Home Page. However, a search overlay has been triggered. The search bar at the top now displays "azure dev". The search results are shown in a modal window, categorized into "Services", "Marketplace", and "Documentation". Under "Services", results include Azure Device Registry, Azure DevOps organizations, and Azure Database for MySQL servers. Under "Marketplace", results include Build Agents for Azure DevOps, Azure DevOps Auditing, Azure DevOps Backup Tool, and Self Hosted Runner for Azure DevOps. Under "Documentation", results include "Install the Azure Developer CLI", "What is Azure Dev/Test offer?", "Set up Azure Deployment Environments - Azure Deployment Environments", and "Continue searching in Microsoft Entra ID". The rest of the page layout remains consistent with the first screenshot.

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



Result:

2116231501095 3
CS23432

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

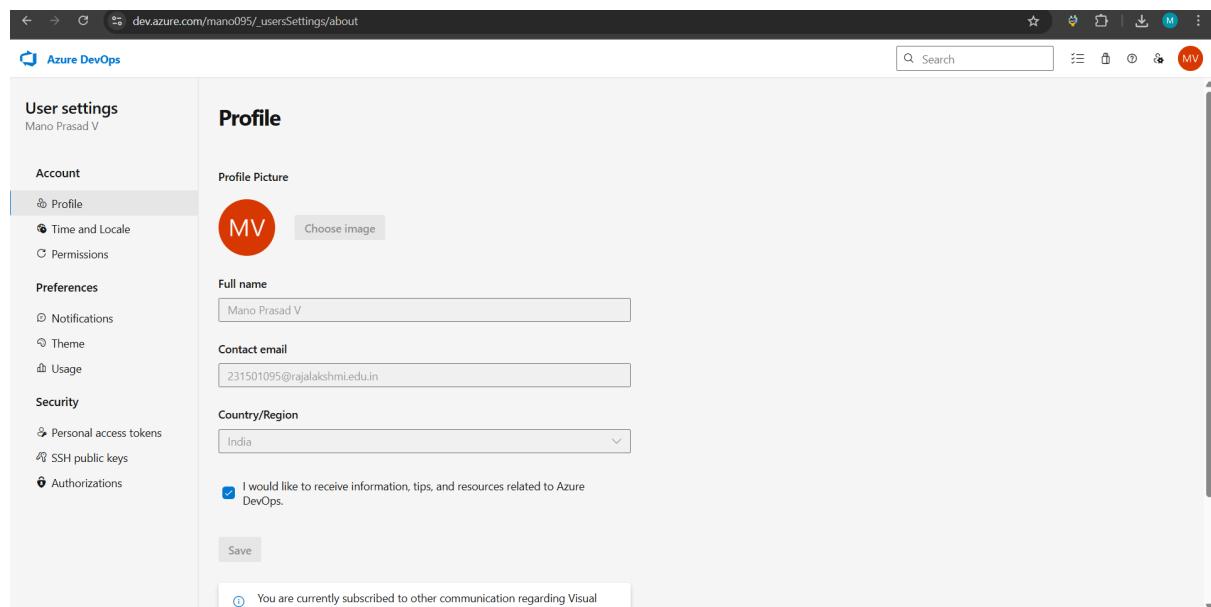
Expr no 2

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

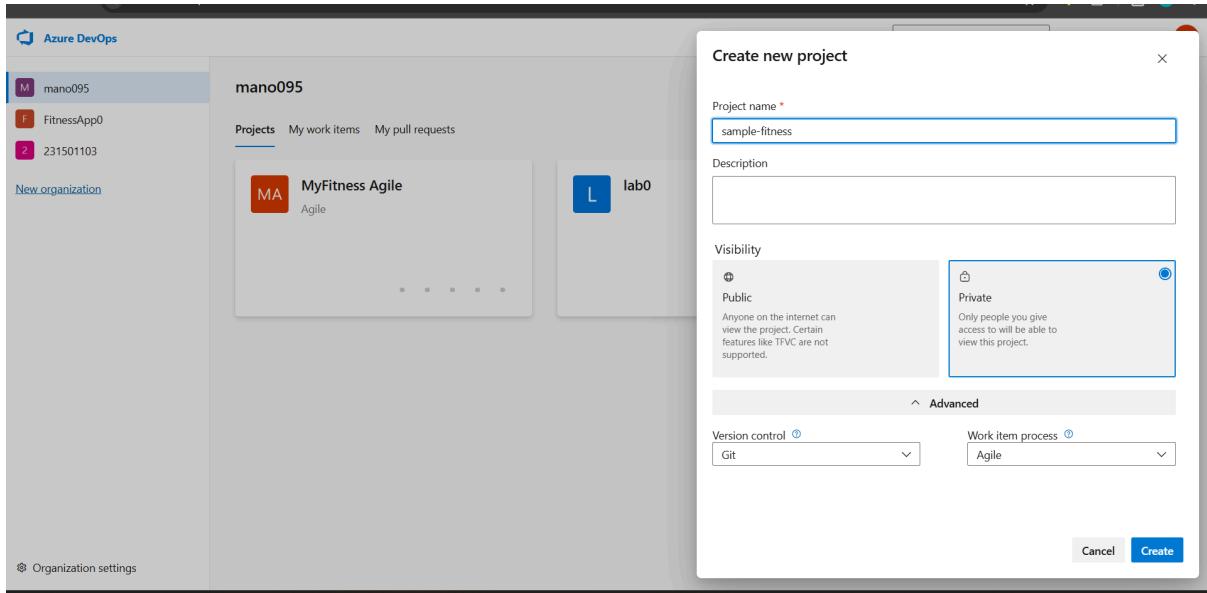
- After the organization is set up, you'll need to create your first **project**. This is where you'll begin to manage code, pipelines, work items, and more.
- On the organization's **Home page**, click on the **New Project** button.
- Enter the project name, description, and visibility options:

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

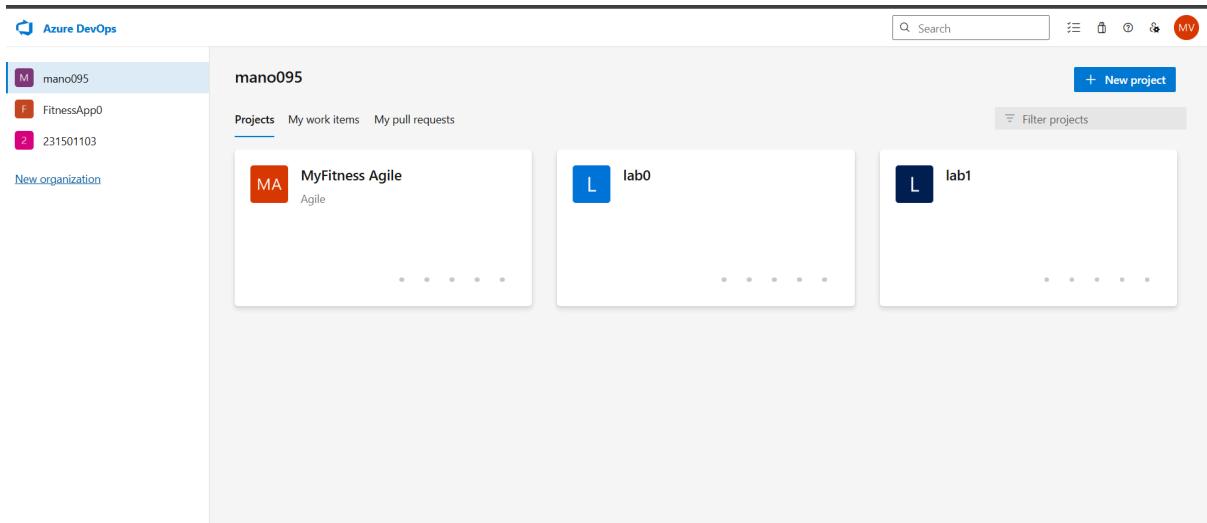
Description: Optionally, add a description to provide more context about the project.

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

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

The screenshot shows the Azure DevOps project dashboard for a project named 'fitnessApp'. The left sidebar contains navigation links for Overview, Summary, Dashboards, Wiki, Boards, Repos, Pipelines, Test Plans, Artifacts, and Project settings. The main content area is titled 'fitnessApp' and includes sections for 'About this project', 'Project stats', and 'Members'. The 'About this project' section provides a brief description of the app's purpose, core features, technologies used, and goals. The 'Project stats' section displays various performance metrics over the last 7 days, such as work items created, completed, pull requests opened, and commits by authors. The 'Members' section shows one member associated with the project.

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.

Work Item Type	Title	State	Value Area
Epic	> 🏋️ Workout Routine	● New	Business
Epic	> 🍗 Nutrition and Meal Plans	● New	Business
Epic	> 📈 Progress Tracking	● New	Business
Epic	> 🌐 Social and Community	● New	Business
Epic	> 📢 Notification and Reminders	● New	Business
Epic	> 🛒 Smart Device Integration	● New	Business
Epic	> 🤖 AI Coach and Recommendations	● New	Business
Epic	> 🎯 Subscription and Monetization	● New	Business
Epic	> 🚶 Admin Dashboard	● New	Business
Epic	> 🌐 User Authentication & Profile	● New	Business

Result:

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

Expr 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 'fitnessApp' project under the 'Backlogs' tab. The backlog is organized by priority (High, Medium, Low) and includes columns for Work Item Type, Title, State, Effort, Business Area, and Tags. Key items listed include 'Workout Routine', 'Browse Workouts', and several user stories related to workout filtering and saving.

Work Item Type	Title	State	Effort	Business Area	Tags
Epic	Workout Routine	New		Business	
Feature	Browse Workouts	New		Business	
User Story	As a user, I want to browse different workout categor...	New		Business	
User Story	As a user, I want to filter workouts by duration and int...	New		Business	
Feature	Save Favourites	New		Business	
User Story	As a user, I want to save my favorite workouts so I can...	New		Business	
User Story	As a user, I want to remove a workout from favorites if...	New		Business	
Feature	Custom Workout plan	New		Business	
User Story	As a user, I want to create a custom workout plan by s...	New		Business	
User Story	As a user, I want to edit or delete my custom plans to ...	New		Business	
Epic	Nutrition and Meal Plans	New		Business	
Epic	Progress Tracking	New		Business	
Epic	Social and Community	New		Business	
Epic	Notification and Reminders	New		Business	

1. Fill in Epics

The screenshot shows the Azure DevOps interface for a backlog item. The URL is https://dev.azure.com/231501103/fitnessApp/_backlogs/backlog/fitnessApp%20Team/Epics?showParents=true&workItem=48. The work item ID is 48, titled "48 Workout Routine". The work item details include:

- Area:** fitnessApp
- Iteration:** fitnessApp\Iteration 1
- Description:** Click to add Description.
- Planning:** Priority: 2, Risk: 1, Business Value: 1, Time Criticality: 1
- 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:** (empty)

The left sidebar shows navigation links for Boards, Backlogs, Work items, and other project management features like Analytics and Pipelines.

2. Fill in Features

A screenshot of the Azure DevOps interface showing a work item details page for a feature. The title is "FEATURE 49" with ID "49". The description is "Browse Workouts". The planning section shows a priority of 2. The deployment section includes a note about tracking releases and turning on deployment status reporting. The development section has a link to an Azure Repos commit or pull request. The related work section is currently empty.

3. Fill in User Story Details

A screenshot of the Azure DevOps interface showing a work item details page for a user story. The title is "USER STORY 52" with ID "52". The description is "As a user, I want to browse different workout categories (e.g., cardio, strength) so I can choose one based on my preference.". The planning section shows a priority of 2. The deployment section includes a note about tracking releases and turning on deployment status reporting. The development section has a link to an Azure Repos commit or pull request. The related work section is currently empty.

Result:

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

Expr no 4

Sprint Planning

Aim:

To assign user story to specific sprint for the Music Playlist Batch Creator Project.

Sprint Planning

Sprint 1

The screenshot shows the Azure DevOps Boards - Sprints page for the fitnessApp project. The left sidebar is visible with options like Overview, Boards, Work items, Boards, Backlogs, Sprints, Queries, Delivery Plans, Analytics views, Repos, Pipelines, Test Plans, Artifacts, and Project settings. The main area is titled "fitnessApp Team" and shows "Iteration 1". A dropdown menu "Person: @Me" is open. The board has columns: New, Active, Resolved, and Closed. There are 15 user stories listed under "New".

User Story	Status
As a user, I want to filter workouts by duration and intensity to find suitable routines.	New
As a user, I want to remove a workout from favorites if I no longer use it.	New
As a user, I want to view ingredients and steps for each meal to prepare them easily.	New
As a user, I want to input my meals manually to track my calorie intake.	New
As a user, I want to see a summary of my daily calorie intake versus my goal.	New
As a user, I want to scan food item barcodes to log them quickly.	New
As a user, I want to view nutrition info of scanned items to decide if I should eat them.	New
As a user, I want to log my weight over time to track my progress.	New
As a user, I want to see a graph of my weight changes to stay motivated.	New
As a user, I want to view a list of past completed workouts to monitor consistency.	New
As a user, I want to see the calories burned per session for performance tracking.	New
As a user, I want to earn badges for completing goals so I feel motivated.	New
As a user, I want to view my earned badges in a profile section.	New

Sprint 2

The screenshot shows the Azure DevOps Boards - Sprints page for the fitnessApp project. The left sidebar is visible with options like Overview, Boards, Work items, Boards, Backlogs, Sprints, Queries, Delivery Plans, Analytics views, Repos, Pipelines, Test Plans, Artifacts, and Project settings. The main area is titled "fitnessApp Team" and shows "Iteration 2". A dropdown menu "Person: @Me" is open. The board has columns: New, Active, Resolved, and Closed. There are 2 user stories listed under "New".

User Story	Status
As a user, I want to save my favorite workouts so I can access them quickly.	New
As a user, I want to create a custom workout plan by selecting exercises so I can tailor it to my needs.	New

Sprint 3

Azure DevOps - Microsoft Azure | My Information | fitnessApp Team Iteration 3 Task | Projects - Home | +

https://dev.azure.com/231501103/fitnessApp/_sprints/taskboard/fitnessApp%20Team/fitnessApp/Iteration%203

Azure DevOps 231501103 / fitnessApp / Boards / Sprints

Search

fitnessApp

Overview

Boards

Work items

Boards

Backlogs

Sprints

Queries

Delivery Plans

Analytics views

Repos

Pipelines

Test Plans

Artifacts

Project settings

fitnessApp Team

Taskboard Backlog Capacity Analytics

Iteration 3 Person: @Me

New Active Resolved Closed

As a user, I want to browse different workout categories (e.g., cardio, strength) so I can choose one based on my preference.

As a user, I want to edit or delete my custom plans to keep them updated.

As a user, I want to receive a daily meal plan based on my fitness goals so I can eat accordingly.

No iteration dates Set dates

Result:

The Sprints are created for the Music Playlist Batch Creator Project.

2116231501095

11
CS23432

Expr no 5

Poker Estimation

Aim:

Create Poker Estimation for the user stories - Music Playlist Batch Creator Project.

Poker Estimation

The screenshot shows the Azure DevOps interface for a project named 'fitnessApp'. On the left, there's a sidebar with options like Overview, Boards, Work items, Repos, Pipelines, Test Plans, Artifacts, and Project settings. The main area is titled 'User Story 56 As a user, I want to create a custom workout plan by selecting exercises so I can tailor it to my needs.' It includes sections for Description, Planning (with Story Points set to 2), Deployment (with a note about tracking releases), Acceptance Criteria (with a placeholder 'Click to add Acceptance Criteria.'), Classification (Value area: Business), Development (with a note about linking to Azure Repos), and Discussion. At the bottom right, there's a 'Related Work' section.

Result:

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

2116231501095

12
CS23432

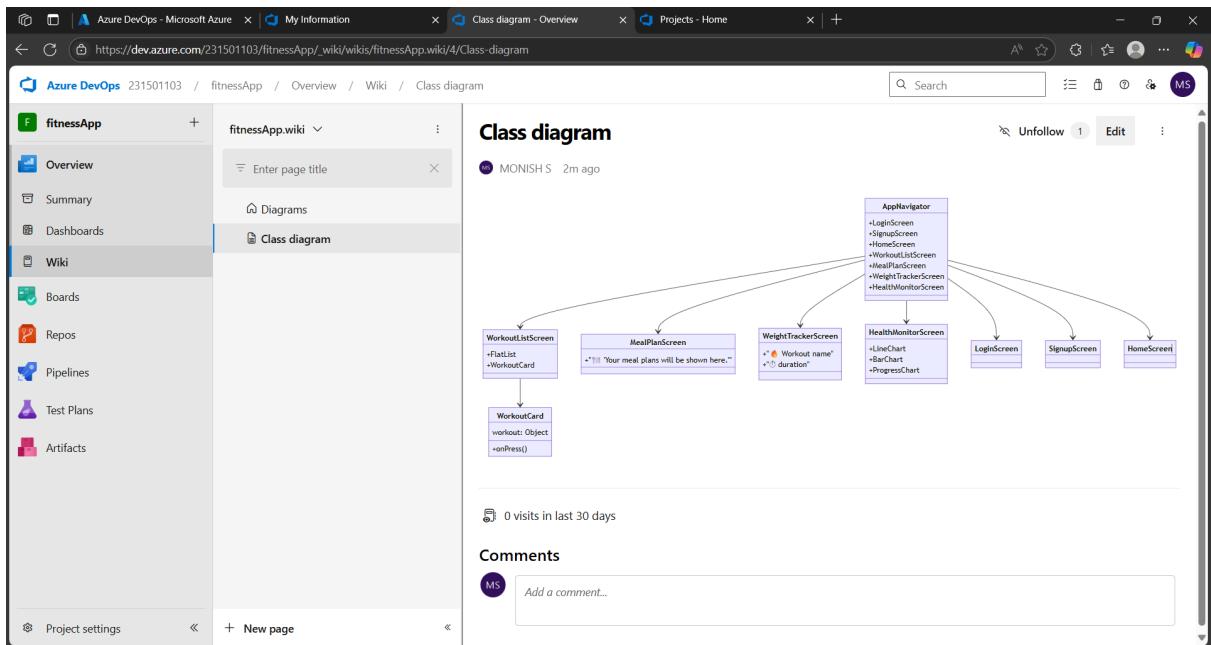
Expr no 6

DESIGNING CLASS AND SEQUENCE DIAGRAMS FOR PROJECT ARCHITECTURE

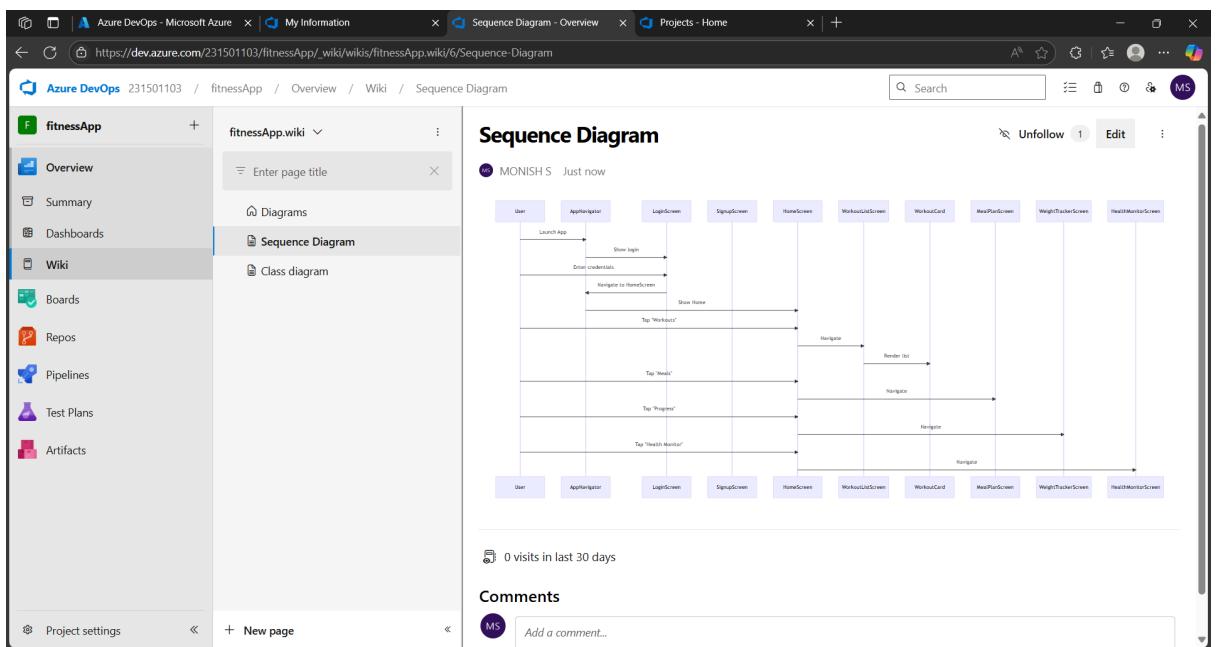
Aim

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

A. Class Diagram



B. Sequence Diagram



Result:

The Class Diagram and Sequence Diagram is designed Successfully for the Music Playlist Batch Creator.

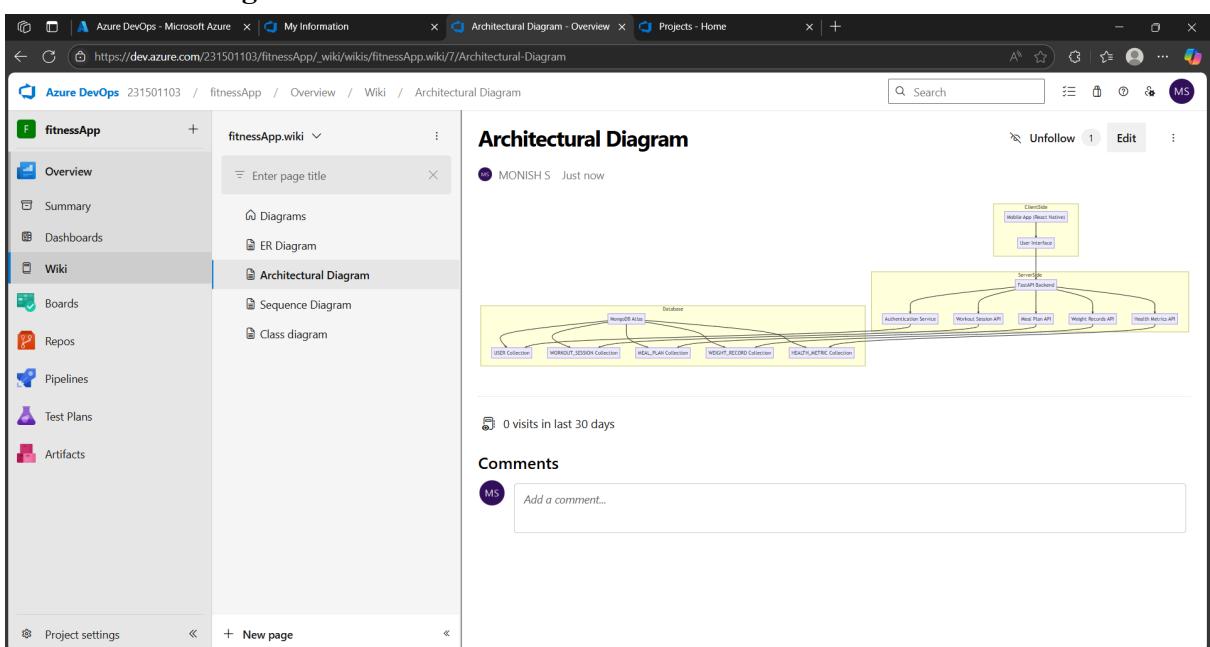
Expr no 7

DESIGNING ARCHITECTURAL AND ER DIAGRAMS FOR PROJECT STRUCTURE

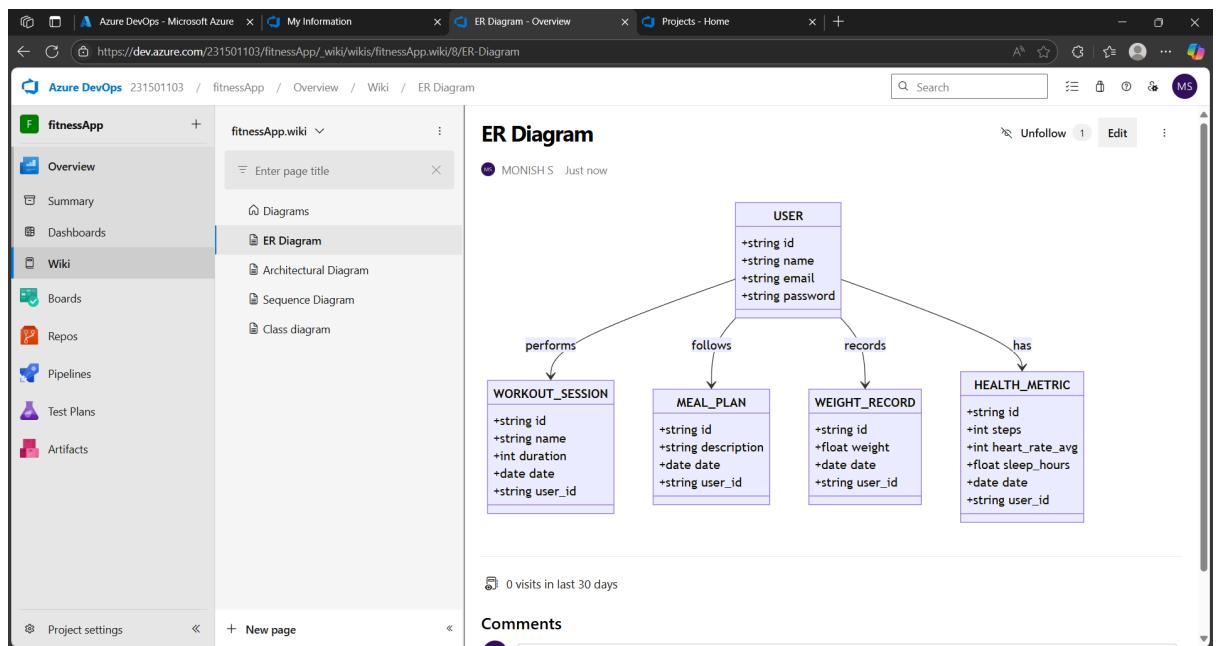
Aim:

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

A. Architectural Diagram



B. ER Diagram



Result:

The Architecture Diagram and ER Diagram is designed Successfully for the Music Playlist Batch Creator

Expr 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**Test Case Design Procedure****1. Understand Core Features of the Application**

- o User Signup & Login
- o Viewing and Managing Playlists
- o Fetching Real-time Metadata
- o Editing playlists (rename, reorder, record)
- o Creating smart audio playlists based on categories (mood, genre, artist, etc.)

2. Define User Interactions

- o Each test case simulates a real user behaviour (e.g., logging in, renaming a playlist, adding a song).

3. Design Happy Path Test Cases

- o Focused on validating that all features function as expected under normal conditions.

- o Example: User logs in successfully, adds item to playlist, or creates a category-based playlist.

4. Design Error Path Test Cases

- o Simulate negative or unexpected scenarios to test robustness and error handling.
- o Example: Login fails with invalid credentials, save fails when offline, no recommendations found.

5. Break Down Steps and Expected Results

- o Each test case contains step-by-step actions and a corresponding expected outcome.
- o Ensures clarity for both testers and automation scripts.

6. Use Clear Naming and IDs

- o Test cases are named clearly (e.g., TC01 – Successful Login, TC10 – Save Playlist Fails).
- o Helps in quick identification and linking to user stories or features.

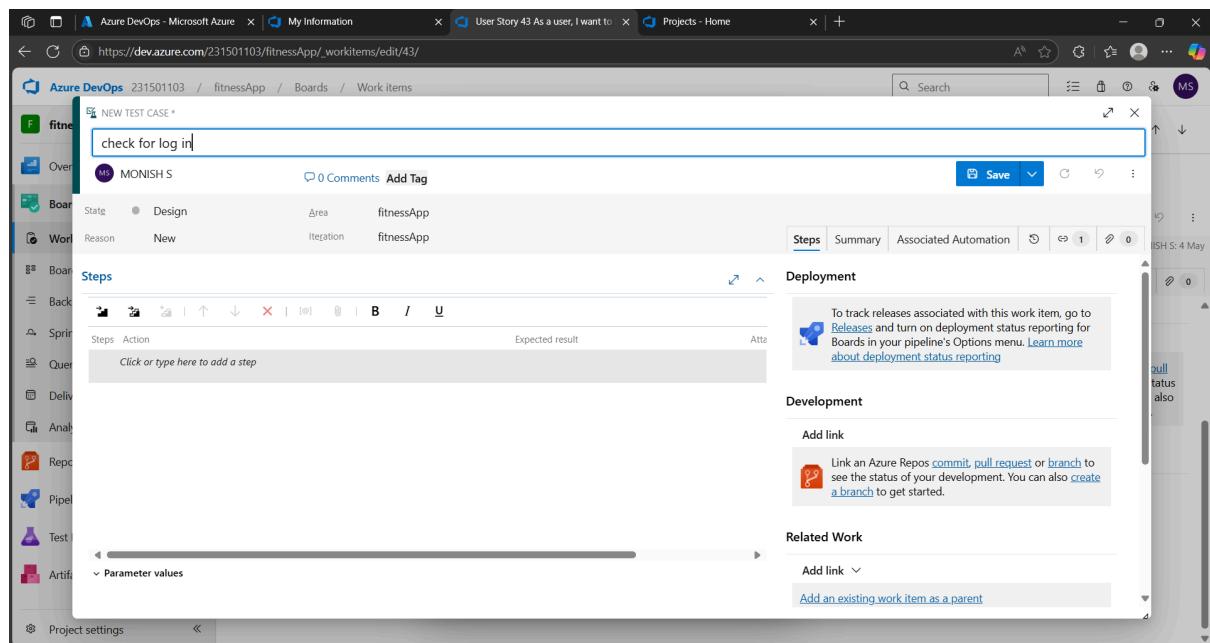
7. Separate Test Suites

- o Grouped test cases based on functionality (e.g., Login, Playlist Editing, Recommendation System).
- o Improves organization and test execution flow in Azure DevOps.

8. Prioritize and Review

- o Critical user actions are marked high-priority.
- o Reviewed for completeness and traceability against feature requirements.

1. New test plan



2. Test suite

The screenshot shows the Azure DevOps interface for a project named "fitnessApp". The left sidebar includes links for Overview, Boards, Repos, Pipelines, Test Plans, Test plans, Progress report, Runs, and Artifacts. The "Test Plans" link is currently selected. The main area displays a "Test Suites" section with a dropdown menu open, showing items like "fitnessApp Team_Stories_fitnessApp" and several numbered test cases (45, 44, 43, 42, 41). To the right, a "Test Points (1 item)" table is shown with one row: "update the details as soon as entering" (Outcome: Passed, Order: 1, Test Case Id: 149). A "Run for web application" button is also present.

3. Test Cases

The screenshot shows the Azure DevOps interface for creating a new work item. The left sidebar lists various work item types: Boards, Work items, Queries, Deliveries, Analytics, Repos, Pipelines, Test, and Artifacts. The "Work items" link is selected. The main area is titled "NEW TEST CASE" and contains fields for "Title" (set to "check for log in"), "Assignee" (MONISH S), "Comments" (0), and "Add Tag". Below these are sections for "State" (Design, Area: fitnessApp), "Reason" (New, Iteration: fitnessApp), and "Steps". The "Steps" section has a table with columns "Action" and "Expected result", and a note "Click or type here to add a step". To the right, there are sections for "Deployment" (with a note about tracking releases via Boards), "Development" (with a note about linking to Azure Repos), and "Related Work" (with a note about adding links to existing work items). A "Save" button is located at the top right.

4. Installation of test

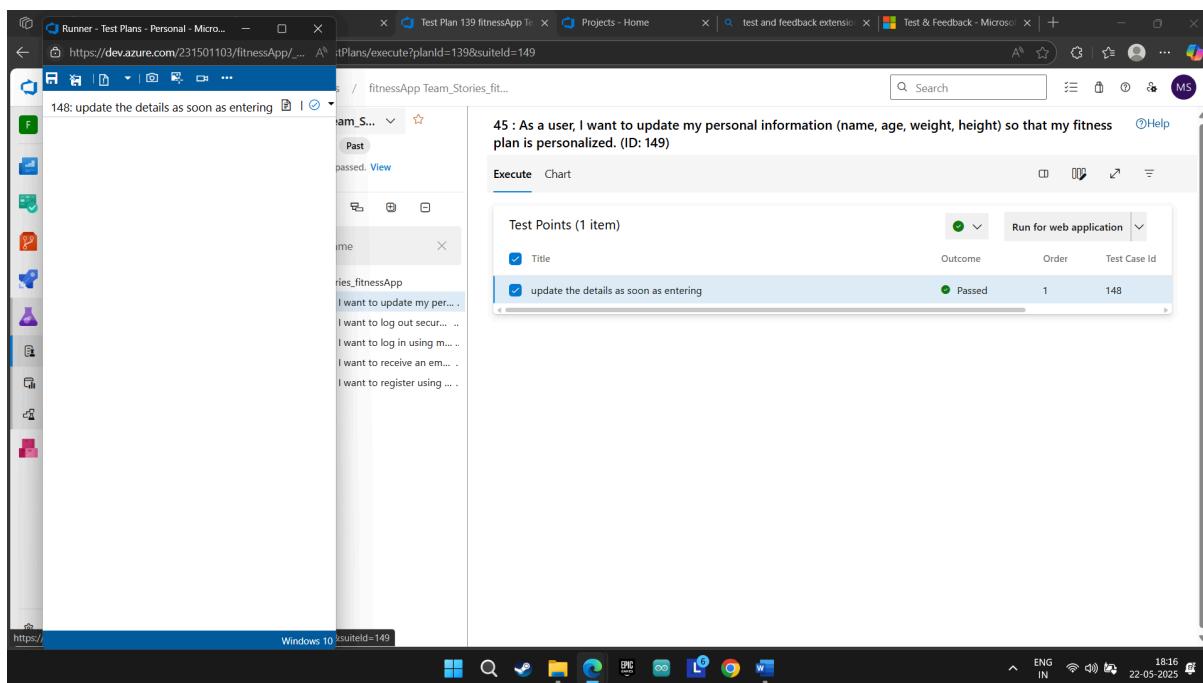
2116231501095

19
CS23432

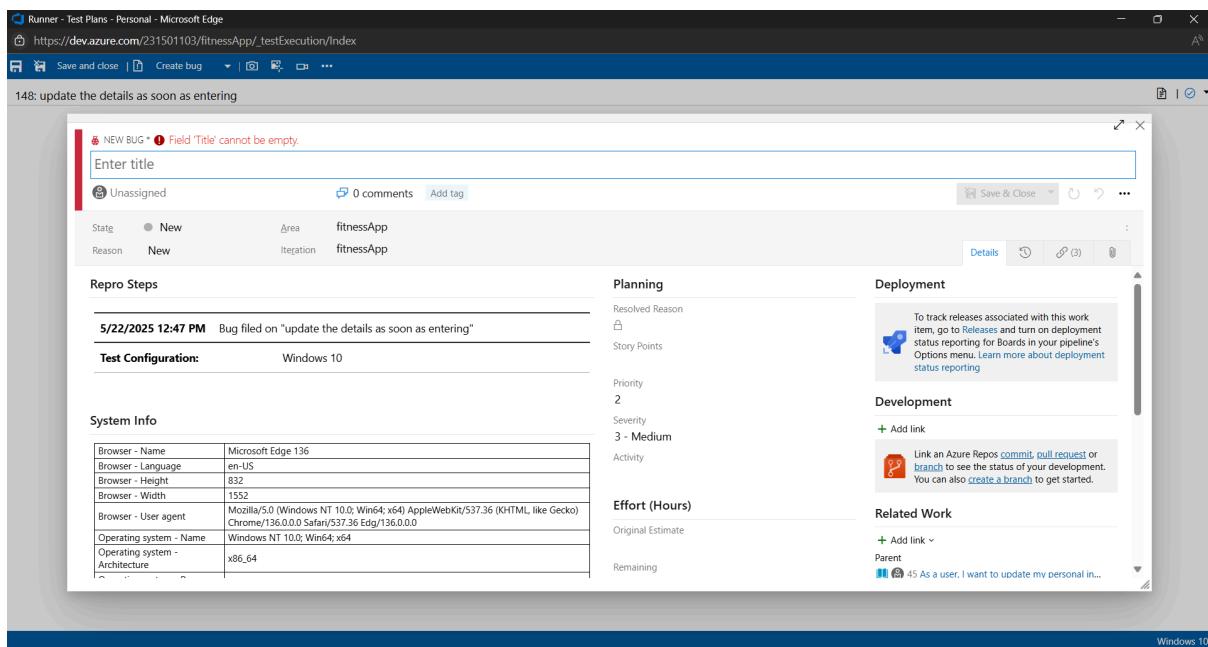
The screenshot shows the Microsoft Edge browser interface with the address bar pointing to <https://microsoftedge.microsoft.com/addons/detail/test-feedback/leakgkdanldebeohldonigkalooaej>. The main content area displays the 'Test & Feedback' extension page. At the top, there's a 'Remove' button and a note stating 'Add-on already installed on your browser'. Below this, there's a preview image of the extension's interface, which includes a screenshot of a web browser with a red box highlighting a button and the text 'change the color of button as per UX mock'. To the right of the preview, it says 'Version 1.0.256.1', 'Updated May 6, 2025', and 'Available in 1 language'. Below the preview, there are links for 'Terms', 'Privacy policy', and 'Developer' (linking to Microsoft Corporation). A 'Report abuse' button is also present. The bottom section contains a 'Description' box with text about team ownership of quality and a screenshot of the extension in action.

5. Running the test cases

The screenshot shows the Azure DevOps interface for 'fitnessApp / fitnessApp / Test Plans / fitnessApp_Team_Stories_fitnessApp'. The left sidebar has 'Test Plans' selected. The main area shows a 'Test Suites' list with one item: 'fitnessApp Team Stories_fitnessApp' (May 4 - May 11, Past, 100% run, 100% passed). Below this, the 'Execute' tab is selected, showing a 'Test Points (1 item)' table. The table has one row: 'Title' (Passed, Order 1, Test Case Id 148). There are buttons for 'Run for web application' and a 'Run' button.

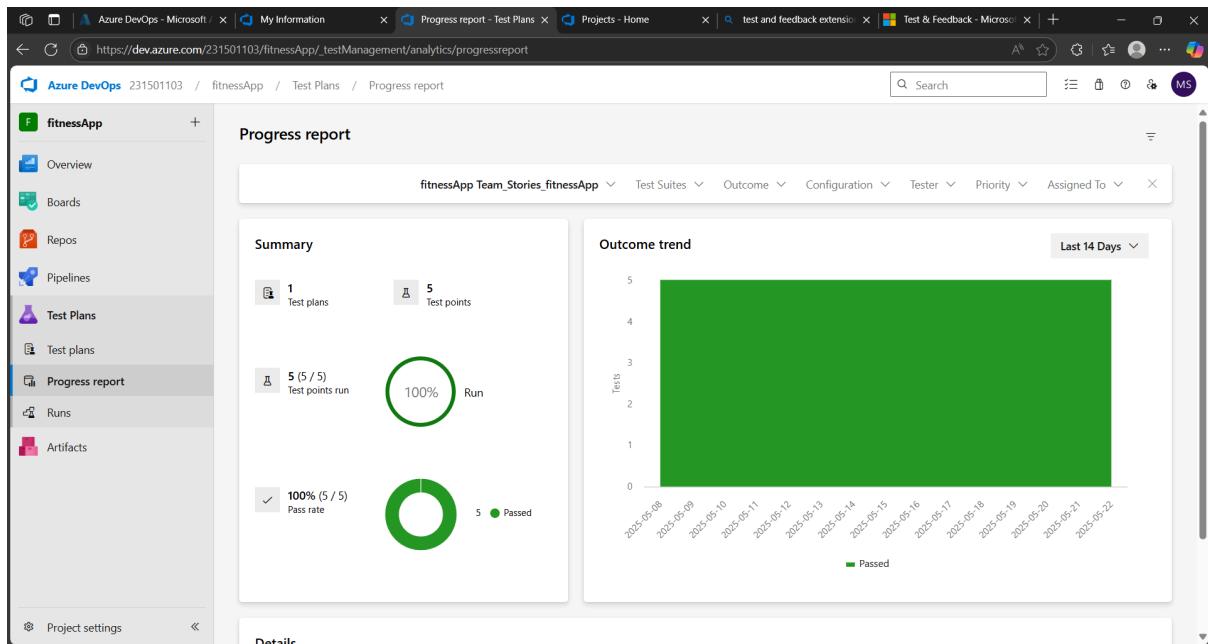


6. Creating a bug



7. Test Case Result

8. Progress Report



Result:

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

Expr no 9

CI/CD Pipelines in Azure

Aim

2116231501095

23
CS23432

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.

Pipeline	Last run	
Social 1 (13)	#20250518.1 • Set up CI with Azure Pipelines Individual CI for main	Sunday 9s
Social 1	#20250518.1 • Set up CI with Azure Pipelines Individual CI for main	Sunday 35s

The image displays two separate Azure Pipeline run pages. Both pages have a dark theme and feature a sidebar on the left with various icons for managing pipelines.

Social 1 Run:

- Description: #20250518.1 - Set up CI with Azure Pipelines
- Stages: One stage, status: Passed
- Completion Time: Sunday, 35s ago

Social 1 (13) Run:

- Description: #20250518.1 - Set up CI with Azure Pipelines
- Stages: One stage, status: Passed
- Completion Time: Sunday, 9s ago

Result:

Thus, the pipelines for the given project “Online Quiz System” has been executed successfully.

2116231501095 26
CS23432

Expr no10

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 Music Playlist Batch Creator project.

GitHub Project Structure

The screenshot shows a GitHub repository page for 'Software_Construction'. The repository is public and has 2 commits. The main branch contains files: assets, components, navigation, screens, App.js, and azure-pipelines.yml. A README file is present but empty. The repository has 1 watching and 0 forks. It uses JavaScript as the primary language.

File	Type	Description	Last Commit
assets	File	Add files via upload	3 weeks ago
components	File	Add files via upload	3 weeks ago
navigation	File	Add files via upload	3 weeks ago
screens	File	Add files via upload	3 weeks ago
App.js	File	Add files via upload	3 weeks ago
azure-pipelines.yml	File	Set up CI with Azure Pipelines	3 weeks ago

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.