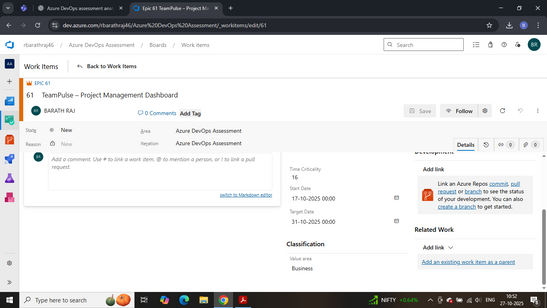
**Azure DevOps Theoretical Assessment**

**Project :** TeamPulse – Project Management Dashboard

**Epic:**

**TeamPulse – Project Management Dashboard**

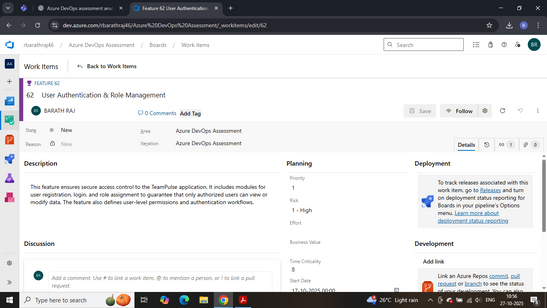
TeamPulse is a web-based dashboard that helps teams manage projects efficiently. It includes modules for **user authentication**, **analytics**, **sprint tracking**, and **team communication**, enabling better collaboration, progress tracking, and transparency throughout the project.



**Feature 1 : User Authentication & Role Management**

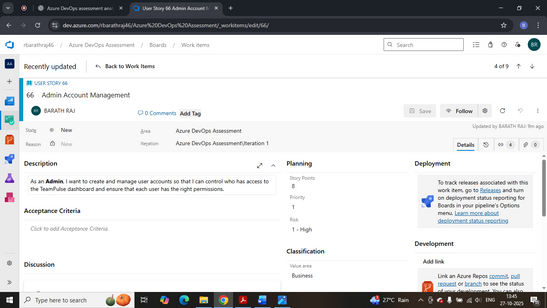
**Feature Description**

This feature manages secure access to TeamPulse through **user registration**, **login**, and **role assignment**. It ensures that only authorized users can access or modify data, with clearly defined **permissions** and **authentication workflows**.



**User Story 1.1 - Admin Account Management**

As an Admin, I want to create and manage user accounts so that I can control who has access to the TeamPulse dashboard and ensure that each user has the right permissions.



**Tasks**

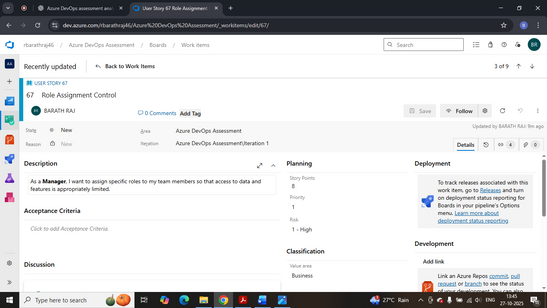
- Define User Roles and Permissions: Identify all user types and access levels.

- Create Database Schema for Users and Roles: Logical schema for user credentials.

- Design UI for Adding and Editing Users: Mock UI for management.

**User Story 1.2 - Role Assignment Control**

As a Manager, I want to assign specific roles to my team members so that access to data and features is appropriately limited.



**Tasks**

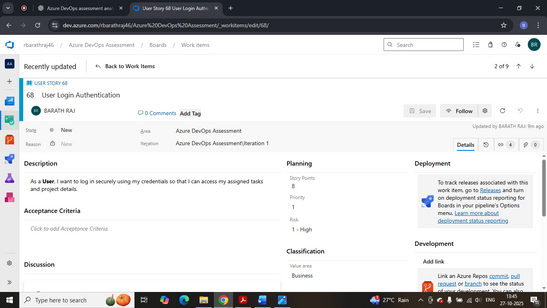
- Map User IDs to Specific Roles.

- Implement Role Selection Dropdown.

- Validate Role Updates.

**User Story 1.3 - User Login Authentication**

As a User, I want to log in securely using my credentials so that I can access my assigned tasks and project details.



**Tasks**

- Design Login and Registration Page.

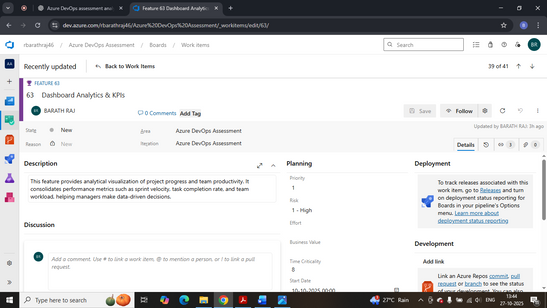
- Integrate Theoretical Authentication Logic.

- Document Password Reset Flow.

**Feature 2 : Dashboard Analytics & KPIs**

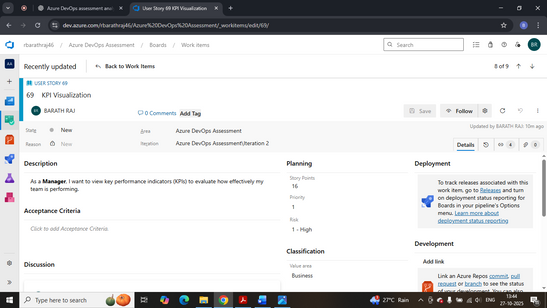
**Feature Description**

Provides analytical visualization of project progress and team productivity. Consolidates metrics such as sprint velocity and task completion rate for better decision-making.



**User Story 2.1 - KPI Visualization**

As a Manager, I want to view KPIs to evaluate team performance.



**Tasks**

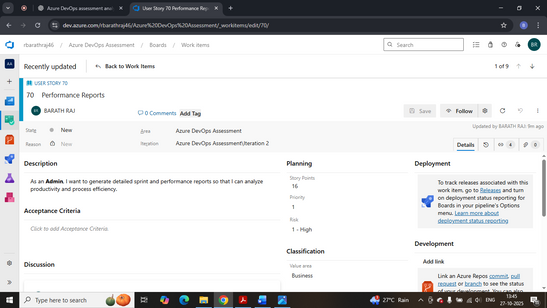
- Identify Relevant KPIs.

- Design KPI Widgets.

- Document Data Refresh Cycle.

**User Story 2.2 - Performance Reports**

As an Admin, I want to generate performance reports for each sprint to analyze team efficiency.



**Tasks**

- Define Report Structure.

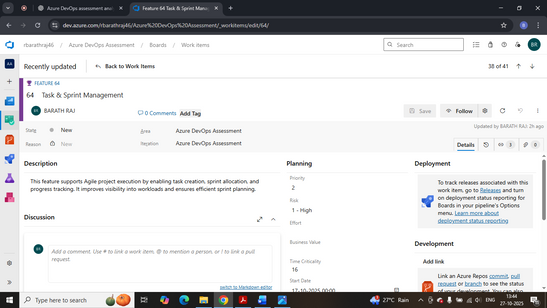
- Design Report Template.

- Document Export Procedure.

**Feature 3 : Task & Sprint Management**

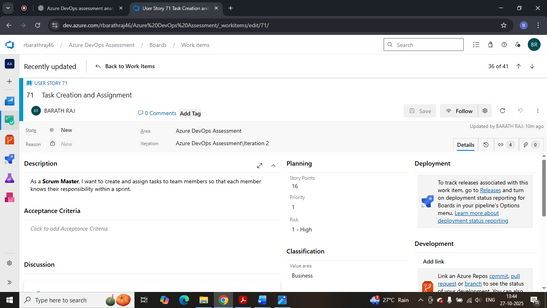
**Feature Description**

Supports Agile execution by enabling task creation, sprint allocation, and progress tracking for better visibility.



**User Story 3.1 - Task Creation and Assignment**

As a Scrum Master, I want to create and assign tasks to team members to define responsibilities clearly.



**Tasks**

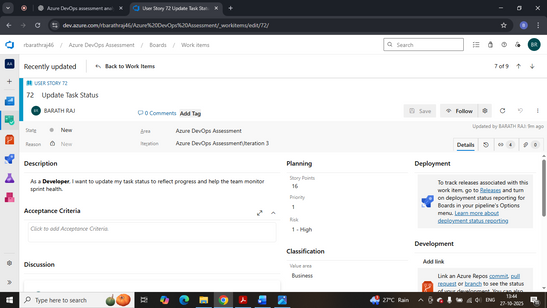
- Design Task Creation Form.

- Define Task Attributes.

- Map Tasks to Sprints.

**User Story 3.2 - Update Task Status**

As a Developer, I want to update my task status to show progress.



**Tasks**

- Create Kanban Board Columns.

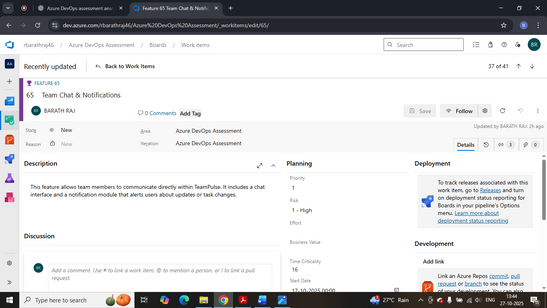
- Implement Status Update Logic.

- Track Status History.

**Feature 4 : Team Chat & Notifications**

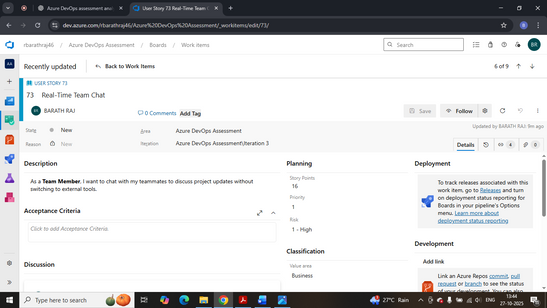
**Feature Description**

Enables internal communication with chat and notification modules for real-time updates.



**User Story 4.1 - Real-Time Team Chat**

As a Team Member, I want to chat with my teammates to discuss project updates.



**Tasks**

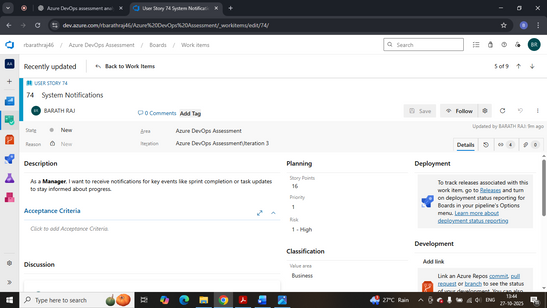
- Design Chat Interface.

- Define Message Schema.

- Document Chat Workflow.

**User Story 4.2 - System Notifications**

As a Manager – Get alerts for sprint completions and task updates.



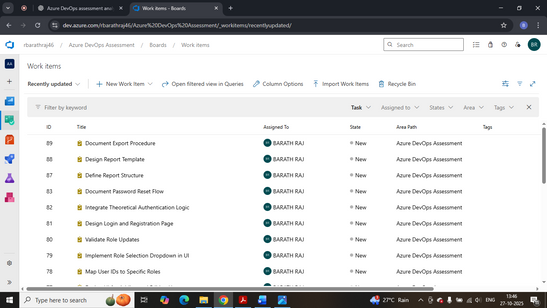
**Tasks**

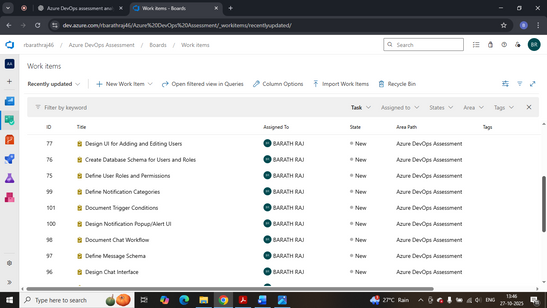
- Define Notification Categories.

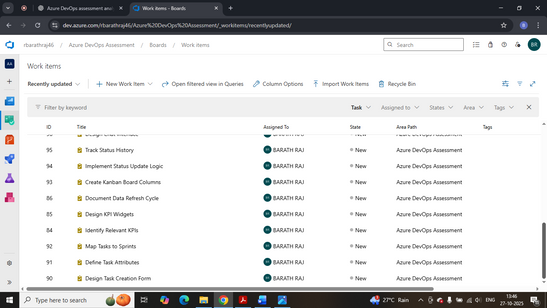
- Design Notification Popup/Alert UI.

- Document Trigger Conditions.

**TASKS OF ALL USER STORIES :**







**Additional Sections:**

**1. Iteration / Sprint Planning**

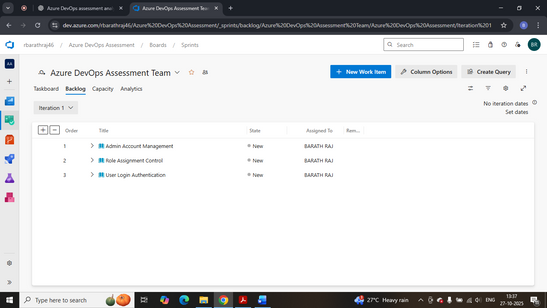
Sprint Planning in **Azure DevOps** organizes work (User Stories and Tasks) into short, focused cycles for consistent progress.

**Sprint Name:** Sprint 1 – Foundation Build   
 **Duration:** 2 Weeks   
 **Goal:** Set up authentication and task management modules

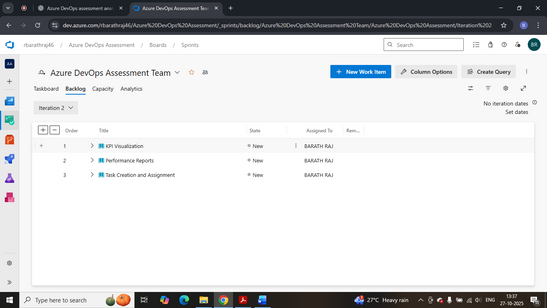
**Steps in Azure DevOps:**

* 1. Go to *Project Settings → Boards → Sprints*
  2. Create *Sprint 1* with start and end dates
  3. Move selected stories and tasks from *Backlog* to *Sprint 1*
  4. Assign tasks to team members
  5. Track progress in *Taskboard* (To Do → In Progress → Done)

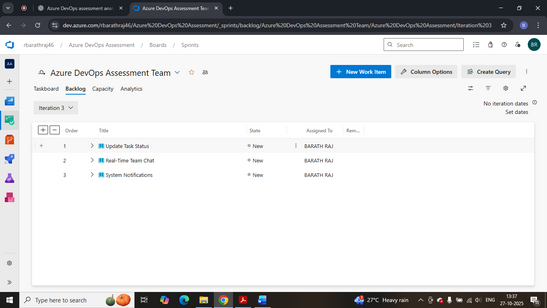
Sprint planning ensures clear goals, balanced workload, and alignment with Agile practices.



Sprint 1 – iteration 1



Sprint 1 – iteration 2



Sprint 1 – iteration 3

**2. Tags and Priorities**

Tags and priorities organize work items by type and urgency, improving team coordination.

**Example for TeamPulse:**

* + **Frontend:** Dashboards, chat (P2)
  + **Backend:** Authentication, APIs (P1)
  + **Database:** Schema, data design (P1)
  + **Analytics:** Reports, charts (P3)

In **Azure DevOps**, add tags to work items and use **priorities (P1–P3)** to show importance. Filtering by these helps manage workload effectively.

**3. Acceptance Criteria**

Acceptance Criteria describe success conditions for each User Story, ensuring clarity among all team members.

**Examples for TeamPulse:**

* + **Login Authentication:** Reject invalid logins, auto-expire sessions, and redirect after success.
  + **KPI Visualization:** Dashboard shows accurate sprint data and auto-refreshes.
  + **Task Assignment:** Tasks link to a sprint and include required fields.
  + **Team Chat:** Only project members can message; show unread alerts.

**4. Definition of Done (DoD)**

Definition of Done sets clear completion rules for all tasks.   
 A story or task is **Done** when:

* + Acceptance criteria are met
  + Documentation is complete and reviewed
  + Work is ready for testing or closure
  + Dependencies are resolved and linked

**5. Burndown Chart**

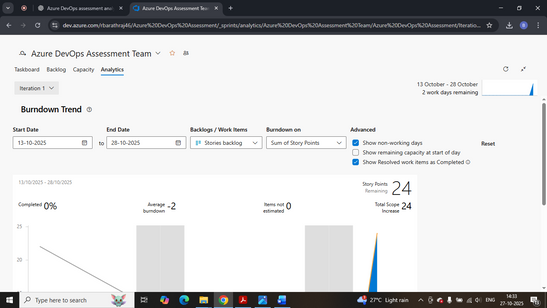
Burndown Chart shows sprint progress by tracking remaining work over time.   
 **In Azure DevOps:**

* + Go to *Boards → Sprints → Analytics tab*
  + Select *Burndown Chart* widget
  + Assign all *Sprint 1* items

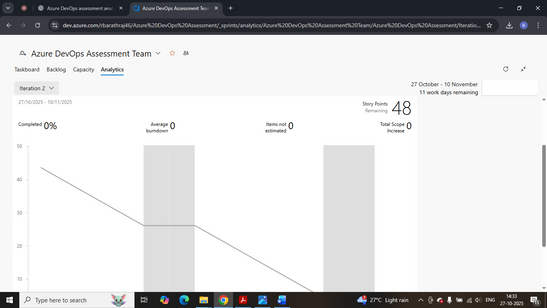
**Chart axes:**

* + X-axis: Time
  + Y-axis: Remaining tasks or story points

It helps teams spot delays early and maintain steady progress.



For iteration 1



For iteration 2

**Summary Table – Overview of extra sections.**

Sprint Planning – Plan and organize tasks for Sprint 1.

Tags & Priorities – Label and rank tasks; include tag list screenshot.

Acceptance Criteria – Set success points; list them per story.

Definition of Done – Set task completion standards.

Burndown Chart – Visually track sprint progress.