Azure DevOps – Theoretical Assignment

Project Title: TeamPulse – Project Management Dashboard

Objective:

To structure a web-based project management dashboard using **Azure DevOps**, demonstrating the use of **Epics, Features, User Stories, and Tasks** in an Agile environment.

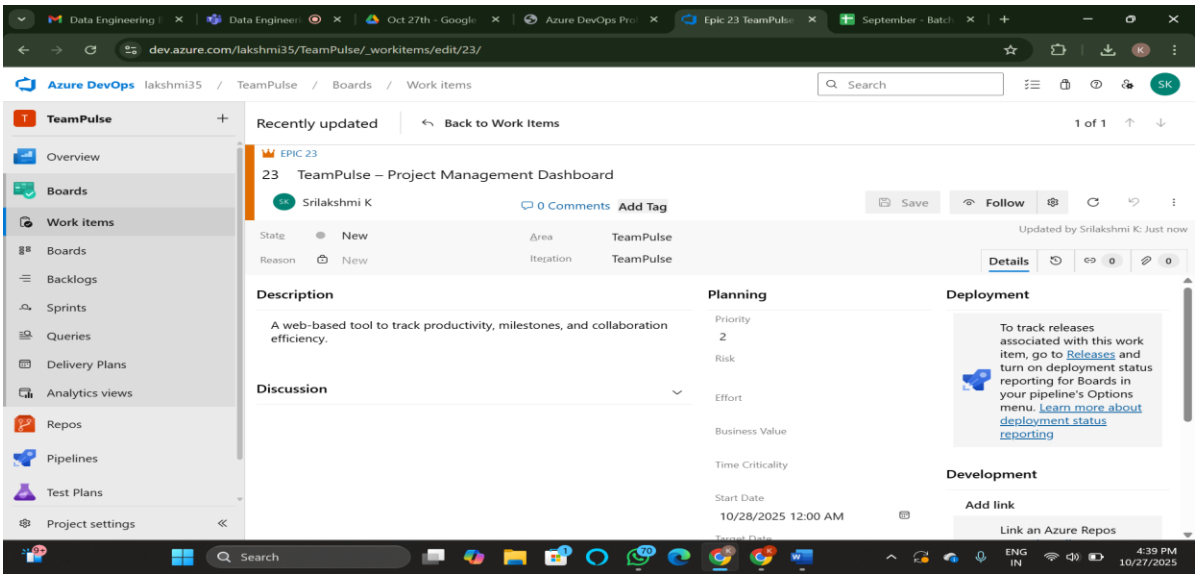
Epic

Epic Title:

TeamPulse – Project Management Dashboard

Description:

This epic focuses on building an internal tool for tracking project milestones, team productivity, and collaboration efficiency within an organization.



Features

1. User Authentication & Role Management

Manages login, registration, and user access based on roles.

2. **Dashboard Analytics & KPIs**

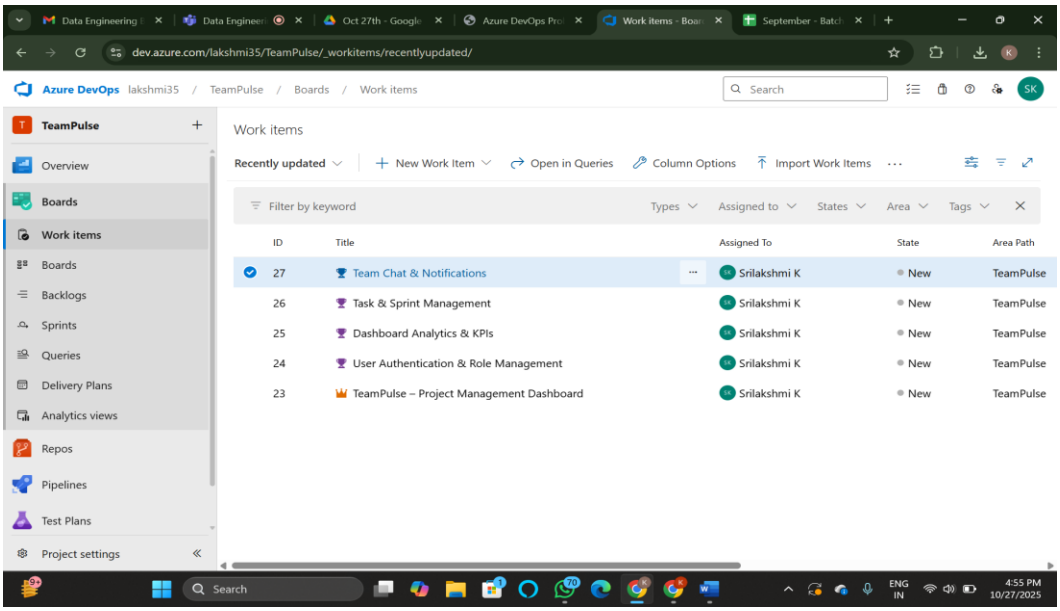
Displays performance metrics, reports, and progress indicators.

3. **Task & Sprint Management**

Handles task creation, updates, and sprint progress tracking.

4. **Team Chat & Notifications**

Provides real-time communication and system alerts for collaboration.



User Stories and Tasks

Feature 1: User Authentication & Role Management

User Story	Task Title
As an Admin, I want to create user accounts so that I can manage access.	Create User Account Form
As a Manager, I want to assign roles to users for access control.	Define Role Assignment Structure
As a User, I want to log in using my credentials to view assigned tasks.	Design Login Page Workflow

Feature 2: Dashboard Analytics & KPIs

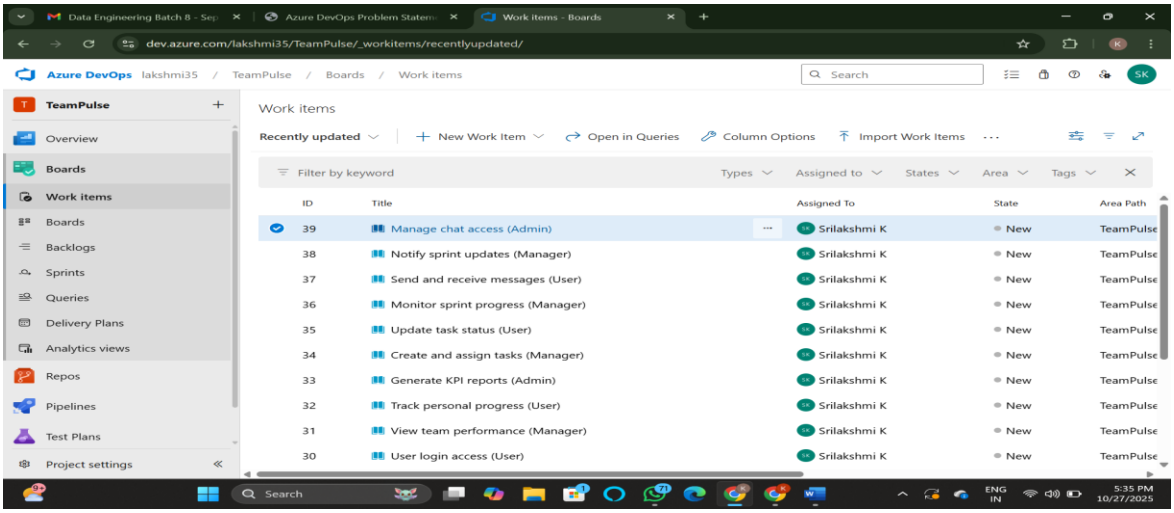
User Story	Task Title
As a Manager, I want to view real-time project performance metrics so that I can monitor team productivity.	Identify Key Performance Indicators
As a User, I want to see my task completion rate so that I can track my performance.	Create Task Completion Progress Bar
As an Admin, I want to generate detailed reports of team KPIs so that I can present them to management.	Design KPI Report Layout

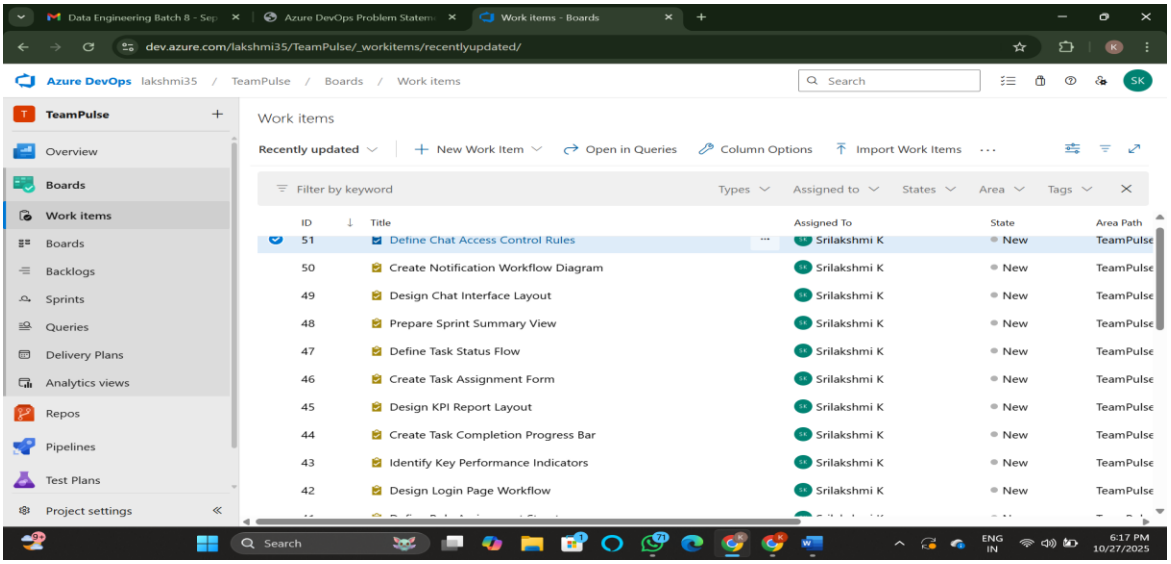
Feature 3: Task & Sprint Management

User Story	Task Title
As a Manager, I want to create and assign tasks so that work is distributed properly among the team.	Create Task Assignment Form
As a User, I want to update the status of my tasks so that my progress is visible to the team.	Define Task Status Flow
As a Manager, I want to view sprint progress so that I can evaluate completion timelines.	Prepare Sprint Summary View

Feature 4: Team Chat & Notifications

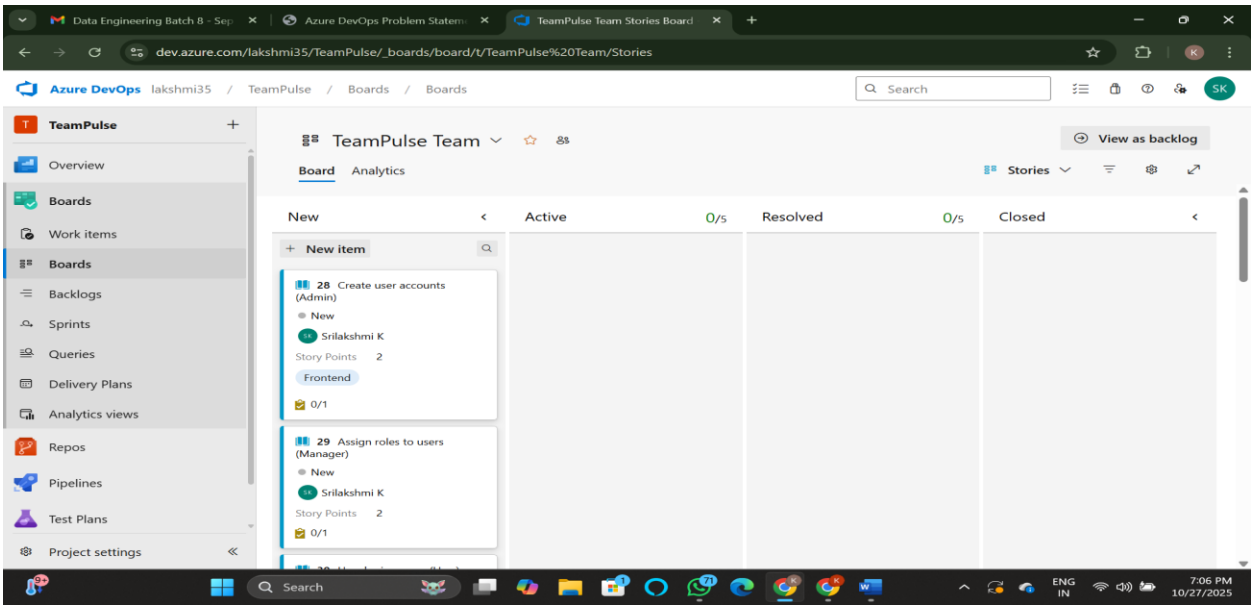
User Story	Task Title
As a User, I want to send and receive messages so that I can collaborate with my team members.	Design Chat Interface Layout
As a Manager, I want to send notifications about sprint updates so that everyone stays informed.	Create Notification Workflow Diagram
As an Admin, I want to manage chat access and monitor communication so that data remains secure.	Define Chat Access Control Rules

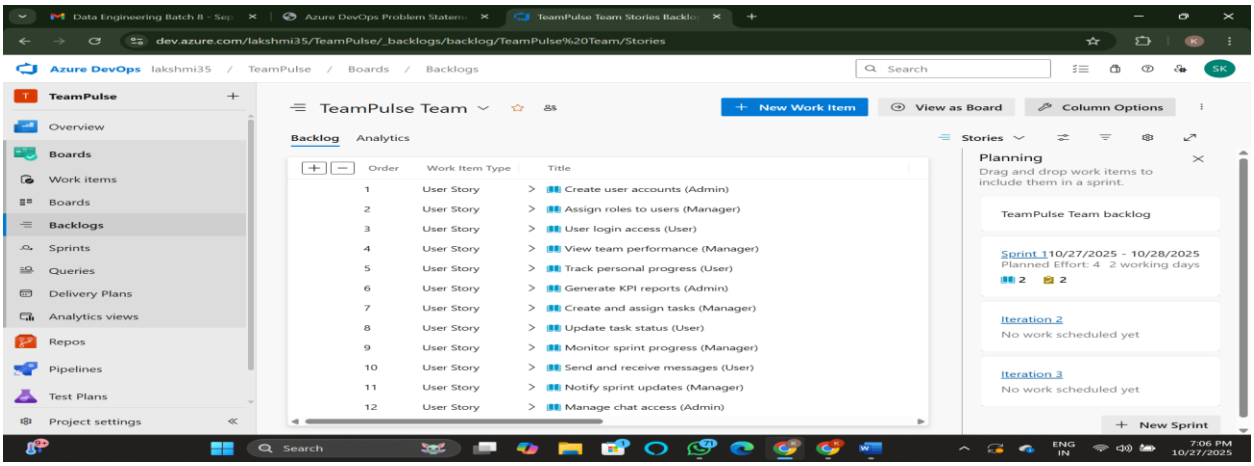




Board / Backlog View

All Epics, Features, User Stories, and Tasks were organized in the **Azure DevOps Board and Backlog View** to show the clear hierarchy.

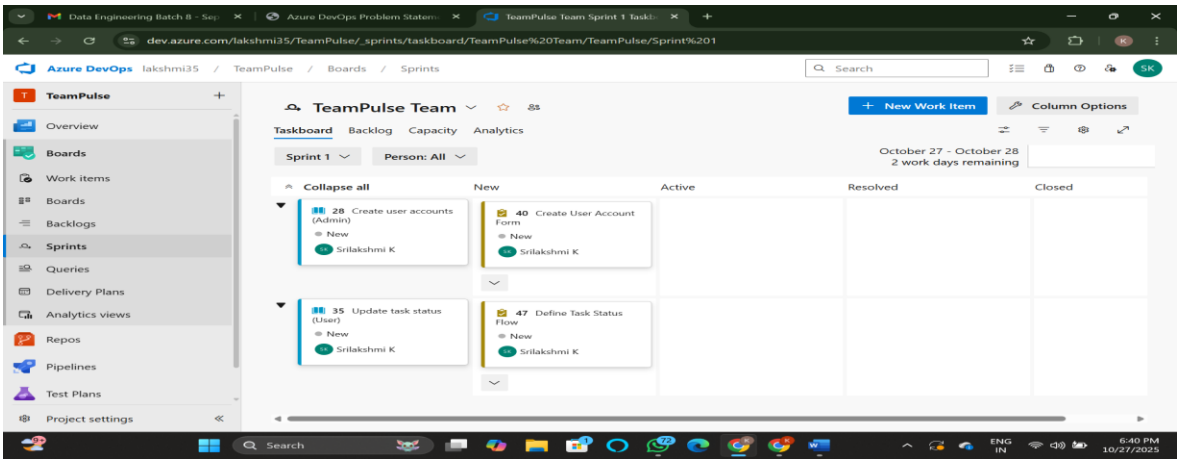




Iteration / Sprint Planning

Sprint 1 Created

- **Sprint Name:** Sprint 1
- **Duration:** 01-Nov-2025 to 14-Nov-2025
- **Assigned Items:** Few user stories and tasks from each feature



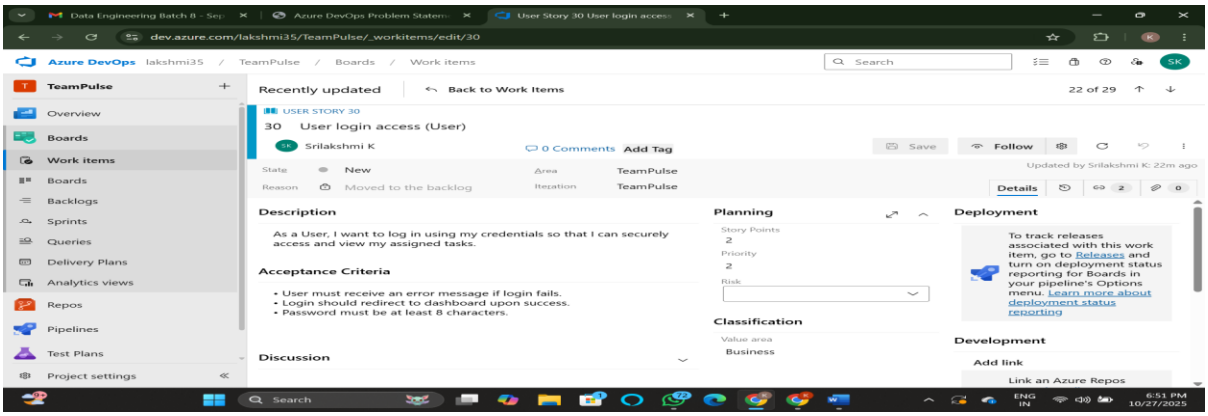
Tags and Priorities

Tag	Usage
Frontend	Used for UI-related user stories (e.g., Login, Dashboard).
Backend	Used for server-side operations (e.g., Role Assignment, Reports).
Database	Used for data storage and management tasks.

Priority Levels:

- **P1 (High):** Login, Dashboard, Chat Features
- **P2 (Medium):** Reports, Task Assignment
- **P3 (Low):** Notifications

Acceptance Criteria

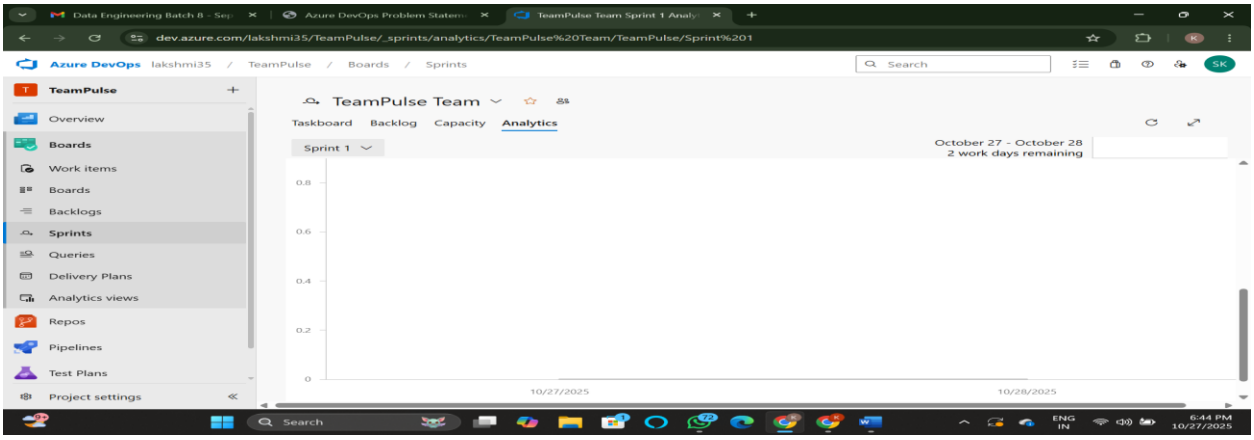


Definition of Done (DoD)

A work item is marked as *Done* when it meets all acceptance criteria, has been reviewed by a team member, tested for functionality, and is ready for integration into the final project build.

Burndown Chart (Optional – Bonus)

To analyze team progress, a **Burndown Chart** was viewed under the **Analytics → Sprint 1** tab



How Azure DevOps Supports Agile Methodology

Azure DevOps supports Agile by providing tools for managing **Epics, Features, User Stories, and Tasks** in a structured way.

It allows teams to **plan sprints**, **track progress visually**, and **collaborate efficiently**. Work can be easily updated, prioritized, and monitored through boards, burndown charts, and dashboards.

Benefits of Breaking Large Projects into Smaller Work Items

- Easier to manage and track progress.
- Improves team collaboration and accountability.
- Helps in identifying issues early.
- Enables better sprint planning and timely delivery.
- Provides clear visibility of work at every level (Epic → Task).

Conclusion

This Azure DevOps setup demonstrates the complete Agile workflow — from Epic creation to sprint planning — showing how teams can effectively plan, track, and deliver software projects with transparency and efficiency.