Project Proposal

Agile Team Sprint Tracker with GitHub Integration

Submit to:

Dr.Nabeel Awan / Miss Zainab

Team Member:

Saim Ali Khan
Arsalan Nawaz
Khuzaima Asif

1. Project Title

Agile Team Sprint Tracker with GitHub Integration

2. Project Overview

The **Agile Team Sprint Tracker with GitHub Integration** is a web-based application designed to help Agile teams manage their sprints, track tasks, and monitor progress seamlessly. The tool integrates with GitHub to sync issues and pull requests, providing a centralized platform for sprint planning, task management, and progress tracking. The application will support multiple sprints, real-time updates, and visual dashboards for better team collaboration and productivity.

3. Objectives

1. Sprint Management: Enable teams to create, update, and manage multiple sprints with ease.
2. Task Tracking: Allow teams to add, assign, and track tasks within each sprint.
3. GitHub Integration: Sync GitHub issues and pull requests with the sprint tracker for seamless task management**.**
4. Progress Visualization: Provide dashboards and reports (e.g., burndown charts) to visualize sprint progress and team performance**.**
5. User Collaboration: Facilitate team collaboration through task assignments, notifications, and real-time updates.

4. Key Features

1. Sprint Management:
- Create, update, and delete sprints.
- Set sprint duration (start and end dates).
- Assign team members to sprints.

2. Task/Issue Tracking:

- Add tasks/issues to sprints.
- Track task status (To Do, In Progress, Done).
- Assign tasks to team members.

3. GitHub Integration:

- Sync GitHub issues and pull requests with the sprint tracker.
- Automatically update task status based on GitHub activity (e.g., when a pull request is merged).

4. Dashboards and Reports:

- Visualize sprint progress using burndown charts.
- Display team performance metrics (e.g., completed tasks, velocity).

5. User Roles and Permissions:

- **Admin:** Can create sprints, assign tasks, and manage team members.
- **Team Member:** Can view and update assigned tasks.

6. Notifications:

- Notify team members about task assignments and sprint updates.

Functional Requirements:

1. User Management

1. User Registration and Login:

- Users can register using their email or GitHub account (OAuth).
- Users can log in and log out of the system.

2. User Roles:

- **Admin:** Can create sprints, assign tasks, and manage team members.
- **Team Member:** Can view and update assigned tasks.

3. Profile Management:

- Users can update their profile information (e.g., name, email, profile picture).

2. Sprint Management

1. Create Sprint:

- Admins can create a new sprint with a name, start date, and end date.

2. View Sprints:

- Users can view a list of all sprints.
- Users can filter sprints by status (e.g., active, completed).

3. Update Sprint:

- Admins can update sprint details (e.g., name, start date, end date).

4. Delete Sprint:

- Admins can delete a sprint (only if it has no associated tasks).

3. Task/Issue Management

1. Create Task:

- Users can create a new task with a title, description, status, assignee, and due date.
- Tasks can be linked to a specific sprint.

2. View Tasks:

- Users can view a list of tasks in a sprint.
- Users can filter tasks by status (e.g., To Do, In Progress, Done) or assignee.

3. Update Task:

- Users can update task details (e.g., title, description, status, assignee).
- Task status can be updated automatically based on GitHub activity (e.g., when a pull request is merged).

4. Delete Task:
- Users can delete a task (only admins or task creators can delete tasks).

4. GitHub Integration

1. Sync GitHub Issues:
- Users can sync GitHub issues with the sprint tracker.
- Synced issues are displayed as tasks in the sprint.

2. Automatic Status Updates:
- Task status is updated automatically based on GitHub activity (e.g., issue closed, pull request merged).

3. Link Pull Requests:
- Users can link pull requests to tasks for better tracking.

5. Dashboards and Reports

1. Sprint Progress Dashboard:
- Display a burndown chart to visualize sprint progress.
- Show the number of completed tasks vs. total tasks.

2. Team Performance Metrics:
- Display metrics like velocity (tasks completed per sprint) and individual performance.

3. Task Distribution:
- Show a breakdown of tasks by status (e.g., To Do, In Progress, Done).

6. Notifications

1. Task Assignment Notifications:
- Users receive notifications when they are assigned a new task.

2. Sprint Updates: Users receive notifications when a sprint is created, updated, or completed.

3. GitHub Activity Notifications:
- Users receive notifications when a linked GitHub issue or pull request is updated.

7. Search and Filter

1. Search Tasks:

- Users can search for tasks by title, description, or assignee.

2. Filter Tasks:

- Users can filter tasks by status, assignee, or due date.

3. Filter Sprints:

- Users can filter sprints by status (e.g., active, completed).

8. Export and Import

1. Export Data:

- Users can export sprint and task data to CSV or Excel.

2. Import Data:

- Admins can import sprint and task data from CSV or Excel.

9. Real-Time Updates

1. Real-Time Task Updates:

- Task updates are reflected in real-time across all users.

2. Real-Time Sprint Updates:

Sprint updates (e.g., start/end dates) are reflected in real-time.

10. Security

1. Authentication:

- Users must log in to access the system.
- GitHub OAuth is used for secure authentication.

2. Authorization:

- Only admins can create, update, or delete sprints.
- Only task creators or admins can delete tasks.

3. Data Encryption:

- Sensitive data (e.g., passwords) is encrypted.

Non-Functional Requirements

1. Performance:

- The system should handle up to 1,000 concurrent users without performance degradation.

2.Scalability:

- The system should be scalable to support multiple teams and large datasets.

3. Usability:

- The user interface should be intuitive and easy to use.

4. Reliability:

- The system should have 99.9% uptime.

5. Security:

The system should comply with data protection regulations (e.g., GDPR).

Conclusion:

These functional requirements provide a clear roadmap for developing the **Agile Team Sprint Tracker with GitHub Integration**. By implementing these features, the system will help teams manage sprints, track tasks, and monitor progress effectively.