Project Retrospective: Sprints 1

Project: CharityPulse (PESU_EC_CSE_J_P49)

Team: 8

Team: Ishaan (Backend), Sujoy (Frontend), Ishika (Frontend), Tanishq (Testing)

Sprint 1: Core Functionality & CI/CD Setup

Goal: Build the Minimum Viable Product (Login, Create Event, Pledge, See Progress) and establish the 5-stage CI/CD pipeline.

What Went Well?

- **Successful MVP:** We successfully built the entire user flow for the core product. An organizer can register, log in, create an event, and a donor can submit a pledge and see the progress bar update in real-time.
- **Git Workflow:** Our branching strategy (feature/ branches, PRs) was a success. We prevented direct commits to main and used Pull Requests for all code reviews.
- **Test-Driven Development (TDD) Success:** Our CI/CD test job *successfully* caught a major logic bug. The events.test.js file failed because our DELETE /api/events route was not checking for existing pledges (as per SRS CFT-F-002). This proved our testing strategy was working from Day 1.
- **Good Task Breakdown:** We successfully translated all SRS requirements into actionable user stories in Jira, complete with Epics and Story Points.

What Didn't Go Well? (Problems)

- **Initial Setup Friction (NPM Issues):** We wasted time debugging local server crashes. The backend server failed with Error: Cannot find module 'cors' and later Error: Cannot find module 'winston'. We learned that because we have two package.json files (frontend/backend), we must run npm install in *both* src/backend and src/frontend directories.
- **CI/CD Test Failures (Import Stack):** Our first integration tests failed the whole pipeline. auth.test.js (which was correct) failed because it imported server.js, which imported events.js, which imported a non-existent logger.js. This was a key learning experience.
- **Workload Imbalance:** As the sprint progressed, it became clear the workload was not evenly distributed. The backend role (Ishaan) was a bottleneck. Almost every frontend story required a new API endpoint, giving Ishaan twice as many PRs and tasks as the frontend developers.

What Did We Learn?

- A "full-stack" story (like "Create Event") is really two separate tasks (frontend UI + backend API) and must be planned as such.
- A failing test in the pipeline is a *success*, not a problem. It's the system working as designed.

- The CI/CD test environment is "dumber" than our local one. It crashes on simple import errors that our local machines might ignore, forcing us to write cleaner code.

Action Items (for Sprint 2)

- **Redistribute Workload:** Re-plan Sprint 2 to account for the backend bottleneck. Frontend developers (Sujoy, Ishika) will be the "Lead" on a feature, and Ishaan will be assigned the specific backend sub-tasks.
- **Backend-Heavy Stories:** Ishaan will focus on complex, backend-only stories (MFA, Account Lockout), while Tanishq will take the lead on all CI/CD and Quality Gate stories (CEFTS-36, CEFTS-37).
- **Fix CI Crashes Immediately:** If the test suite fails to *run* (like the logger.js crash), that PR becomes the number one priority to fix and merge, as it blocks all other tests.