**SCRUM Roles & Events:**

As a student intern on a SCRUM-based team, my role would be to contribute as a junior developer. I would actively participate in all SCRUM ceremonies and contribute by communicating my progress, raising blockers, and collaborating with the team to improve the sprint outcomes.

Daily Stand-ups:

* I would give quick updates on what I worked on, what I’m doing next, and if I’m facing any challenges. This will help the team stay in sync and resolve blockers early.

Sprint Planning:

* I would help break down tasks assigned to me, estimate effort if appropriate, and clarify requirements with the Product Owner or senior developers.

Sprint Reviews:

* I would demonstrate what I’ve worked on and explain any technical or user-facing challenges I faced.

Retrospectives:

* I would reflect on what went well and where I could improve, whether it’s communication, time management, or understanding technical concepts.

As a student, my contributions would focus on learning best practices while delivering value. I’d be proactive in asking questions, taking on manageable tasks, and helping test or document features when needed.

**SDLC Planning for Flagging System Redesign**

To redesign the automatic flagging system, I would follow the Software Development Lifecycle (SDLC) in these phases:

Requirements:

* First, I would meet with stakeholders to understand what qualifies as a red flag (some examples include: exam fails or long practice gaps). I’d review the current system and identify what needs improvement or more clarity.

Design:

* Next, I would design the structure for the flagging rules and data format. I would create flowcharts or pseudo-code to map out how data will be evaluated. I’d also design how the frontend will display flags and handle overrides.

Implementation:

* I would build the backend API using Node.js and TypeScript with a /evaluate endpoint that applies flagging rules. For the frontend, I’d use AngularJS with PrimeNG to create a form for data input and flag display.

Testing:

* I would write unit tests for the backend logic to make sure each rule behaves as expected. I’d also manually test the frontend to ensure correct inputs trigger the right flags and that overrides work.

Deployment:

* For this project, deployment would involve running both backend and frontend locally. I’d document the setup clearly in the README for reproducibility.

Maintenance:

* After initial deployment, I’d collect feedback on usability, review flagged edge cases, and iterate on the rules or UI to improve clarity and accuracy.