

1. SCRUM Roles & Events – what would I actually do as the intern?

I'd jump in as one of the developers, tackling bite-sized tickets like “add a new flag rule” or “clean up the UI.” At the daily stand-up I'd share what I finished yesterday, what I'm doing today, and shout if I'm stuck. During sprint planning I'd help break stories into smaller tasks, throw in my best guess on effort, and claim the ones that fit my skills. In backlog grooming I'd ask questions so the stories make sense and maybe suggest a few that we're missing. At the sprint review I'd demo the new flagging logic so the team and stakeholders can see it live. And in the retro I'd speak up about what worked (say, TypeScript caught a bug) and what didn't (maybe our test data setup is slow). Basically, I'd keep the team looped-in and move tickets forward—no fancy titles, just steady hands-on work.

2. SDLC Planning – how would I rebuild the flagging system, step by step?

1. Requirements – First, sit with the people who know the rules inside out and write down every “if this, then flag that” case.
2. Design – Sketch a simple diagram: browser → small Node API → separate rules file. Decide on data shapes (Candidate, Flag) so everyone codes to the same contract.
3. Implementation – Code in short sprints. Wire up the /evaluate endpoint, drop each rule into TypeScript, and hook the AngularJS form to the API. Push code through pull requests so teammates can review it.
4. Testing – Add unit tests for each rule and a quick end-to-end test that fills the form and checks the flag table. Make GitHub Actions run those tests on every PR.
5. Deployment – Package the API with Docker, serve the static front end with Nginx, and spin it up on a test box. If it looks good, ship the same image to production.
6. Maintenance – Keep an eye on logs and user feedback. If new licensing rules pop up or we spot a bug, jump back to step 1 for that change and loop through the cycle again.