

Agile Sprint Timeline for Dynamic SPA Project

This document outlines a proposed 6-sprint Agile timeline (~12 weeks total) for developing a dynamic Single Page Application (SPA) with a Node.js backend. Each sprint is designed to be about 2 weeks long with clear deliverables at the end.

Sprint 1: Foundation & Setup (Week 1–2)

- Set up project repo (GitHub, branching strategy).
- Configure dev environment (Node.js + Express, PostgreSQL or MariaDB).
- Deploy a 'Hello World' Node/Express app to Railway or GCP (CI/CD pipeline).
- Serve static landing page (index.html) via Express.
- Document setup in README.
- Deliverable: Working app skeleton, visible online at project URL.

Sprint 2: Authentication & User Accounts (Week 3–4)

- Implement registration & login for students & instructors.
- Secure password storage (bcrypt).
- Basic session handling (JWT or cookies).
- Add form validation (emails, password complexity).
- Add logout.
- Deliverable: Users can sign up, sign in, and log out securely.

Sprint 3: Instructor Core Features (Week 5–6)

- Instructor can create courses.
- Instructor can enroll students (upload list or send code).
- Instructor can create teams inside courses.
- Build UI flow for these actions (SPA components + backend API).
- Deliverable: Instructors manage courses & teams via UI.

Sprint 4: Reviews & Questions (Week 7–8)

- Instructor can create reviews with CRUD questions.
- Question types: Likert, multiple choice, short answer.
- Students can complete reviews assigned to them.
- Reviews have public & private feedback fields.
- Deliverable: Students complete reviews, instructors see responses.

Sprint 5: Reports & Metrics (Week 9–10)

- Student average scores (convert to 100pt scale).
- Class metrics (sortable, filterable).
- Student can view combined feedback & overall scores.
- Instructor can view aggregated reports.

- Deliverable: Functional reporting dashboard for students & instructors.

Sprint 6: Security, UX, and Polish (Week 11–12)

- Add MFA (email or TOTP w/ Passport.js).
- Improve responsive design (Tailwind or Bootstrap).
- Accessibility audit (alt text, aria labels).
- Add PWA manifest ('Add to Home Screen').
- Finalize CI/CD (GitHub Actions tests + auto-deploy).
- HTTPS via Certbot or GCP managed SSL.
- Deliverable: Production-ready dynamic SPA that meets requirements.

Ongoing (across all sprints):

- Write tests (unit & integration) as you build features.
- Maintain backlog in Trello/Jira/Notion.
- Demo at end of each sprint (even just to yourself) — keeps scope realistic.