Betygsgrundande inlämningsuppgift: DevOps Pipeline med testning av fullstack-applikation

Deadline för inlämning: 28 december (dvs söndag sista kursveckan för första delkursen *01 DevSecOps kontinuerlig utveckling och automatisering*).

Översikt

Du ska bygga en fullstack-webbapplikation med ett REST API och sätta upp en komplett CI/CD-pipeline i GitHub Actions. Applikationens tema och omfattning väljer du själv, men den måste uppfylla de tekniska kraven nedan.

Applikationen ska lösa ett verkligt eller påhittat problem. En todo-app är godkänd, men en app med bara en resurs som heter “Item” med fältet “name” är inte tillräckligt. Din huvudresurs ska ha minst 3 fält/attribut utöver ID.

Har du en **färdig app** sedan tidigare - helt ok att återanvända, men skriv test- och devop-flöden för den nu!

Tekniska krav

Backend (valfritt språk: Node.js, Python, C#, Java, Go, etc.)

- REST API med **minst 4 endpoints** som implementerar:
  - `GET` – hämta en eller flera resurser
  - `POST` – skapa ny resurs
  - `PUT` eller `PATCH` – uppdatera resurs
  - `DELETE` – ta bort resurs
- Någon form av **datalagring** (databas, JSON-fil, eller in-memory med seed-data)
- Korrekt användning av **HTTP-statuskoder** (200, 201, 400, 404, etc.)

Tips! Om du inte vill använda en “riktig databas”

JSON-server är en npm-modul (**npm install json-server**) som tillåter GET/POST/PUT/DELETE requests mot en JSON-fil och automatiskt ger dig ett REST-api. *Dokumentation här*. Vi går igenom detta bibliotek vecka 6!

Frontend (vanilla JS/TS eller valfritt ramverk: React, Vue, Svelte, Angular)

- Ska **konsumera API:et** och visa data för användaren
- Användaren ska kunna utföra **minst 3 av 4 CRUD-operationer** via gränssnittet
- Grundläggande **felhantering** (visa meddelande om något går fel)

Tester

- **Vitest/Jest** – enhetstester för backend-logik
- **Playwright/Cypress** – end-to-end-tester för frontend
- **Postman/Newman** – API-tester

Minimikrav: 5 meningsfulla tester som faktiskt testar applikationen inom varje testtyp.

GitHub Actions Pipeline

- Workflow som triggas på `push` och/eller `pull_request`
- Ska köra alla tester
- Ska **passera** (grön bock) vid inlämning

Betygskriterier

Godkänt (G)

Kriterium	Krav
API	4 endpoints (GET, POST, PUT/PATCH, DELETE) som fungerar
Frontend	Visar data från API, minst 3 CRUD-operationer fungerar
Tester	Minst 5 tester per testtyp, som passerar och testar reell funktionalitet
Pipeline	GitHub Actions workflow som kör tester och passerar
Branch protection	Main-branch är skyddad – kräver att workflow passerar före merge
Dokumentation	README med: hur man startar projektet, kort beskrivning av applikationen
Kod	Går att klona och köra lokalt med dokumenterade instruktioner

Väl Godkänt (VG)

Uppfyller alla krav för G, plus minst 4 av följande:

Kriterium	Beskrivning
Testbredd	Minst 10 tester per testtyp
Kodkvalitet	Automatisk kodstilskontroll (t.ex. ESLint, Pylint) som körs i pipeline

Kriterium	Beskrivning
Säkerhet	Pipeline kontrollerar att inga dependencies har kända säkerhetsvarningar (t.ex. `npm audit`, `pip-audit`)
Produktionsbygge	Om ramverket har produktionsläge: testerna körs mot produktionsversionen, inte utvecklingsversionen
API-design	Validering av input (t.ex. "namn får inte vara tomt"), tydliga felmeddelanden till användaren
Frontend UX	Minst 2 av: fungerar på dator och mobil, bekräftelse innan radering, felmeddelanden visas för användaren
Branch protection	Kräver godkänd code review från annan person utöver att tester passerar
Dokumentation	Instruktioner för hur API:et används (vilka endpoints finns, vad skickar man in, vad får man tillbaka)

Inlämning

Checklista

- Repo är publikt eller lärare är inbjuden som collaborator
- README innehåller startinstruktioner
- GitHub Actions workflow syns och passerar
- Alla endpoints kan testas (bifoga Postman-collection eller curl-exempel)

Redovisning (om gränsfall, otydlig funktionalitet, annars inte)

Var beredd att:

- Förklara dina arkitekturbeslut
- Visa att du förstår din egen kod
- Svara på frågor om pipeline-konfigurationen

Exempelapplikationer (inspiration)

För att ge dig en känsla för lämplig omfattning:

- Bokhantering för ett bibliotek
- Receptsamling med ingredienser
- Enkelt bokningssystem
- Produktkatalog med kategorier
- Quiz-applikation med frågor och resultat

Lycka till!