



# Examinerande uppgift - Testautomatisering med Continuous Integration

Systemmentor AB

## Del 1 – Rapport (G)

- Beskriv vad **Continuous Integration (CI)** innebär och förklara dess **fördelar och utmaningar**.
- Förklara vad **testautomatisering** är och hur det kopplas samman med **CI**.
- Redogör på ett grundläggande sätt för de **steg som krävs för att tester ska köras automatiskt i GitHub Actions**.

## VG

### – Alla G delar + punkten nedan

- Jämför **Continuous Integration** med **Continuous Delivery** och **Continuous Deployment** – hur skiljer sig dessa steg åt i ett automatiserat flöde?
- Analysera **för- och nackdelar med olika testverktyg** som används i CI-miljöer, till exempel **JUnit** eller **Selenium**.
- Resonera kring **hur val av testteknik och strategi påverkar byggtid**.

## Del 2 – Inloggningsfunktion

På <https://www.saucedemo.com/> finns en enkel inloggningsruta där användaren kan logga in.

Din uppgift är att skapa **automatiserade testfall** för denna funktion med hjälp av **Selenium WebDriver**.

Testerna ska vara tydligt strukturerade och täcka samtliga krav nedan.

### För G – Grundläggande test

- Skapa ett testfall där **inloggningen lyckas** med korrekta användaruppgifter.
- Kontrollera att användaren **loggas in och hamnar på startsidan** efter lyckad inloggning.

### För VG – Utökade tester(Samtliga G delar + utökade tester)

Utöver det grundläggande testet ska du även:

- Skapa ett testfall där **fel användarnamn** anges och verifiera att ett **felmeddelande** visas.
- Skapa ett testfall där **fel lösenord** anges och verifiera att **ett felmeddelande** visas.

**OBS!** Inkludera länk till ditt GitHub-repo i din rapport (del 1)!

**OBS! OBS!** Se till att GitHub-repot är Public och inte Private! Kommer jag inte repot får ni automatiskt underkänt.

**OBS!** Inkludera länk till ditt GitHub-repo i din rapport (del 1)!

## Del 3 - Integrationstester

**OBS! OBS!** Se till att GitHub-repot är Public och inte Private! Kommer jag inte repot får ni automatiskt underkänt.

### För G –

- Skapa **integrationstester** som anropar <https://fakestoreapi.com/>.
- Testa:
  - En GET-förfrågan till /products returnerar statuskod 200.
- Skapa en GitHub Actions-pipeline som automatiskt kör dina tester vid varje push till GitHub.

### För VG –

Utöver kraven för G ska du även:

- Utöka integrationstesterna så att de validerar fler detaljer i API-svaren, till exempel:
  - Antalet produkter som returneras matchar det förväntade antalet.
  - En specifik produkt innehåller korrekta fält som title, price och category.
  - Ett specifikt produkt-ID returnerar rätt data.