



Kursnamn: Testverktyg  
Klass: Teknisk testare  
Termin: VT 2025

# INLÄMNINGSUPPGIFT 1

## INLEDNING

Bakgrundsbeskrivning, frågeställning, avgränsning och mål	<p><u>Bakgrund:</u> I denna uppgift ska ni bygga en enkel webbapplikation med HTML, CSS och JavaScript. Ni ska också bygga en Server som kan ta emot requests och skicka tillbaka responses till er webbapplikation med data från en MySQL Databas. Fokuset ligger på testning av Servern</p> <ul style="list-style-type: none"><li>• Enhetstester</li><li>• Integrationstester</li><li>• End-to-end tester</li></ul> <p><u>Mål:</u> En hemsida med formulär för att skicka data och läsa data från en server med databas</p> <p><u>Avgränsning:</u> Det är tillåtet att använda tredjeparts bibliotek/ramverk för denna uppgift. Men koden ska vara skriven i JavaScript, HTML och CSS endast.</p>
Varför ska ni utföra detta arbete?	<p><u>Syftet:</u> Studenten ska lära sig och få färdigheter i</p> <ul style="list-style-type: none"><li>• Databashantering med SQL</li><li>• Verktyg för testfallshantering</li><li>• Verktyg för felrapportering</li><li>• Testscripting</li><li>• GUI</li><li>• Studenten ska få färdighet i att kombinera front-end med back-end</li></ul>
Vad ska ni leverera?	<p>Följande ska levereras:</p> <ol style="list-style-type: none"><li>1. Zip-fil med samtliga projektfiler (ta bort node_modules innan du zippar) samt rapporter för testfallshantering (antingen länkar eller pdf)</li><li>2. En README-fil med instruktioner för hur man startar applikationen och testar</li><li>1. Redovisning av uppgift i video (~10 min) eller i person</li></ol>

## ER PROJEKTUPPGIFT



Vad ska ni göra?

## Struktur

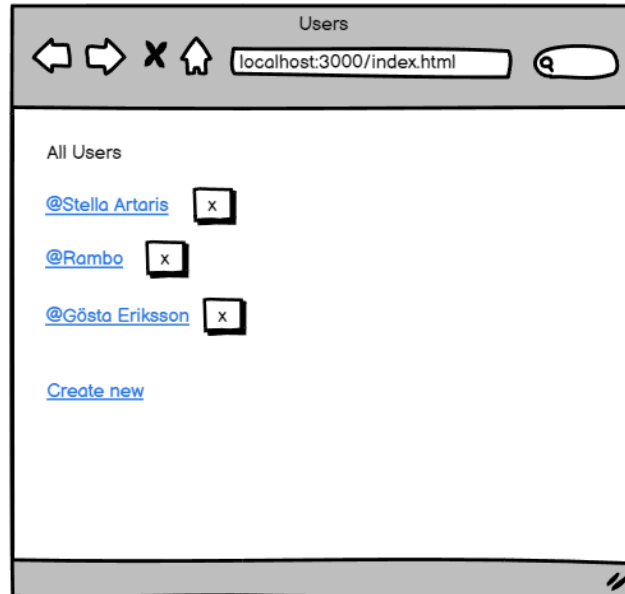
Ni ska ha ett npm projekt med följande struktur

```
__tests__/  
  unit/  
    unit.test.js  
  integration/  
    integration.test.js  
  e2e/  
    end2end.test.js  
public/  
  index.ejs  
  user.ejs  
  edit.ejs  
  create.ejs  
  style.css  
server.js  
logic.js  
database.js  
mockdatabase.js  
package.json  
package-lock.json
```

## Designspecifikation

Följande design är bara ett förslag på hur det kan se ut, ni har fria händer att styla och inkludera mer information. Designen som presenteras kan betraktas som en miniminivå för vad som krävs.

## Startsida



Detta är landningssidan. På denna sida hämtar ni med hjälp av ett fetch anrop i javascript alla användare i från servern. Ni visar upp deras namn som länkar och en knapp med x brevid. Om man klickar på länken kommer man till deras profilsida. Länken ska bestå av url:en `user.html?id={id}` där id är användarens databas id. Detta är viktigt att ha så att profilsidan kan hämta rätt användare från servern. Om man trycker på x-knappen ska användaren tas bort från databasen. Under alla användare finns en länk med "Create user". Om man trycker på den kommer man till en sida för att skapa en ny användare.

## Profilsida



I det här fallet har användaren kommit till en användares profilsida. Här ser användaren:

- Edit knapp



- Namn
- Smeknamn
- Ålder
- Bio: Kort beskrivning
- Länk tillbaka till startsidan (Go to users)

På denna sida hämtar ni med hjälp av ett fetch anrop i javascript en användare i från servern med hjälp av id i url:en. Ni visar upp deras information på sidan. Om man trycker på edit-knappen ska man komma till användarens redigeringssida. Länken ska bestå av url:en `edit.html?id={id}` där id är användarens databas id, samma som på profilsidan. Detta är viktigt att ha så att edit-sidan kan hämta rätt användare från servern. Om man trycker på "Go to users" kommer man tillbaka till startsidan.

## Editsida

Users

localhost:3000/edit.html?id=1

Stella Artaris

Go to users

Nickname: Stellis

Age: 54

Bio: Jag gillar gå på catwalks och dricka champagne

Save

I det här fallet har användaren valt att klicka på "Edit"-länken på profilsidan för att redigera uppgifter om användaren. Här kan vi redigera olika uppgifter om användaren som dess namn och ålder. När vi trycker på spara ska formulärdatan (det vi har skrivit in) skickas till servern så att denna data sparas i databasen, vi ska se den nya datan i formulären nu. Om vi trycker på "Go to users" kommer vi tillbaka till startsidan som vanligt.

## Skapasida



I det här fallet har användaren klickat på "create user" på startsidan. Här fyller användaren in uppgifter för en ny användare. När användaruppgifter har blivit ifyllda och vi trycker på "save" så ska vi komma tillbaka till startsidan. Samma sak sker om vi trycker på "Go to users".

### Spara data

För att spara data använder ni ett MySQL. Vi kopplar databasen till servern med express och med hjälp av npm paket (bibliotek)

### Template engine

Vi kommer att använda en template engine för att rendera vår html (Jag föreslår EJS eftersom det är det vi kommer att gå igenom på kursen, men ni får naturligtvis använda någon annan om ni vill experimentera).

Hur ska ni lösa uppgiften?

1. Skapa en mapp på din dator med namnet "inlämningsuppgift 1".
2. Initiera ett npm projekt i mappen med "npm init".
3. Installera express
4. Skapa server.js och gör routes till de olika sidorna med express
5. Skapa följande filer: style.css, index.ejs, profile.ejs, edit.ejs och create.ejs i public mappen
6. Lägg in hårdkodad html som representerar vyerna enligt spec. I det här skedet koncentrerar vi oss bara på att få rätt utseende, vi gör inget med funktionalitet
7. Börja med att lägga länkar till de olika sidorna så att ni kan se ett flöde mellan olika sidor.
8. Skapa databas i MySQL Workbench med en tabell (users) med kolumner, id, name, nickname, age, och bio. Använd rätt typer för de olika kolumnerna.
9. installera mysql2 i npm projektet för att kunna använda databasen från servern
10. Skapa funktioner för att hämta/uppdatera/ta bort data i databas med hjälp av mysql2. Gör detta i en separat modul (database.js).
11. Skapa en modul (logic.js) som använder database.js för att hämta data. Använd denna istället för att direkt använda database.js i server.js, på det viset kan vi senare använda mockdatabase.js i logic.js och skriva unit-tester för logic.js som inte beror på MySQL.
12. Installera mocha, supertest och mochaawesome i npm projektet för att skriva era tester
13. Skriv tester, unit tester av enskilda funktioner du har skrivit (minst 5), och integrationstester för logic.js (minst 5), kör testerna med mocha och generera testrapport med mochaawesome
14. Installera Playwright i npm projektet



	<p>15. Skriv end-to-end tester med Playwright för hela applikationen. Skriv minst 1 för varje use-case:</p> <ul style="list-style-type: none"><li>• Skapa användare</li><li>• Se alla användare</li><li>• Editera användare</li><li>• Ta bort användare</li></ul>
--	---

## INLÄMNING OCH REDOVISNING

Inlämning	<p>Inlämning sker via LearnPoint <b>datum 4/6 senast kl. 23:59</b></p> <ul style="list-style-type: none"><li>• Inlämning av zip med projektfiler (utan node_modules)</li><li>• README-fil med instruktioner för att köra och testa applikation</li><li>• Video på genomgång av applikation (max 10 min)</li></ul>
-----------	---

## BEDÖMNING OCH ÅTERKOPPLING

Bedömning sker mot följande betygskriterier:	<p>Betygskriterierna för Godkänd respektive Väl godkänd är:</p> <p><b>Godkänd</b></p> <ul style="list-style-type: none"><li>• Den studerande kan med handledning använda vanligt förekommande testverktyg för; testfallshantering, testscriptning och testautomatisering,</li><li>• Den studerande har uppvisat färdighet och kompetens att med handledning driva utveckling/programmering/scriptning/automation på ett marknadsmässigt sätt,</li><li>• Den studerande har uppvisat kunskaper om databaskommandon och färdigheter i databashantering som är relevanta och nödvändiga för teknisk testning,</li><li>• Den studerande har uppvisat kunskap om Testfallsrapportering</li></ul> <p><b>Väl godkänd</b></p> <ul style="list-style-type: none"><li>• Studenten har uppvisat kunskap om hur end-to-end tester ska skrivas och har uppvisat kunskap om testrapportering, och har under redovisning kunnat svara på nyanserade frågor om applikationen funktionalitet och testning av den.</li></ul>
Återkoppling	<p>I och med redovisningstillfället ges muntlig återkoppling till studenten.</p> <p>Studenten får skriftlig återkoppling via lärplattformen LearnPoint <b>senast den 18/6</b></p>