

Eksamensopgave

Kursus: SW4FED-02 Front-end Development
Varighed: 24 timer
Eksamenstermin: Eksamen juni 2025
Praktiske informationer: Eksamensbesvarelsen skal afleveres i ZIP-format Husk at aflevere før deadline som er angivet i Wiseflow, hvor opgaven skal afleveres.
Hjælpemidler: Alle hjælpemidler må benyttes, herunder internettet som opslagsværktøj, men opgaven skal besvares individuelt. Og husk referencer hvis du genbruger kode fra andre opgaver eller projekter – skal indsættes som en kommentar. Både brug af ChatGPT og AI integreret i en kodeeditor så som Intellisense og GitHub CoPilot er tilladt.
Ansvarlig underviser: Poul Ejnar Røvsing

Indledning

Ved besvarelsen af opgaverne skal du huske, at du ved fremlæggelsen skal kunne demonstrere opfyldelse af kursets læringsmål ud fra opgaverne. Din opgavebesvarelse skal derfor dække så mange læringsmål som muligt.

Opgave 1

Du skal udvikle en applikation eksaminator kan bruge ved mundtlige eksamener, hvor der skal trækkes et spørgsmål.

Programmet skal kodes i C# til .Net-plattformen med anvendelse af MAUI og køre på en platform (device) efter eget valg. Du skal selv fastlægge programmets brugergrænseflade og softwarearkitektur.

Overordnet beskrivelse

Før eksamen starter kan brugeren (eksaminatoren) oprette en eksamen og indtaste studienumre og navne på de studerende i den rækkefølge, de skal eksamineres.

Under eksaminationen kan eksaminator indtaste kommentarer og efter hver eksaminand indtastes karakteren.

Efter eksamen kan eksaminator se de studerendes karakterer samt et gennemsnit for holdet.

Alle data skal persisteres. Du bestemmer selv hvordan. Du kan f.eks. benytte en SQLite database, men du kan også vælge at benytte filer eller en json-server (som i opgave 2), som du kan tilgå med HttpClient.

Krav til funktionalitet

1 Opret eksamen

Brugeren af appen skal kunne vælge opret eksamen hvorved følgende informationer kan indtastes og lagres i databasen:

- Eksamenstermin (f.eks. "sommer 25")
- Kursusnavn
- Dato
- Antal spørgsmål
- Eksaminationstid i hele minutter
- Starttidspunkt (klokkeslæt)

2 Tilføj studerende

Brugeren af appen skal kunne vælge en eksamen og derefter vælge tilføj studerende, hvorefter der kan indtastes studienummer og navn på studerende, i den rækkefølge de skal eksamineres i.

3 Start eksamen

Brugeren kan starte eksamen for en oprettet eksamen, hvorefter studienummer og navn for den første studerende vises.

4 For hver studerende (eksaminand) er forløbet således:

4.1 Brugeren trykker på "Træk spørgsmål" hvor efter appen kommer med et tilfældigt tal mellem 1 og antallet af spørgsmål for denne eksamen.

4.2 Brugeren trykker på "Start eksamination" hvorefter programmet starter en timer som tæller ned fra eksaminationstiden, og som markerer visuelt og evt. med lyd når eksaminanden har brugt tiden.

4.3 Mens eksaminationen forgår og indtil der er indtastet (valgt) en karakter, kan brugeren indtaste noter til eksaminandens eksamen.

4.4 Brugeren trykker på "Slut eksamination" hvorefter programmet stopper timeren og den faktiske eksaminationstid registreres.

4.5 Brugeren indtaster (eller vælger fra en liste) den studerendes karakter, hvorefter de registrerede oplysninger (spørgsmåls nummer, faktisk eksaminationstid, noter og karakter) persisteres.

4.6 Brugeren trykker "Næste studerende" hvorefter der forsættes fra 4.1, hvis der er flere studerende. Hvis der ikke er flere studerende på listen, så får brugeren en meddelelse om at eksamen er færdig.

5 Se historik

Hvis brugeren vælger "Se historik", så kan der vælges en tidligere afholdt eksamen, og brugen får så vist en liste med de studerende og deres registrerede oplysninger samt et beregnet gennemsnit.

Opgave 2 er på næste side

Opgave 2

Du skal udvikle en front-end til en Web applikation som eksaminator kan bruge ved mundtlige eksamener, hvor der skal trækkes et spørgsmål.

Front-enden skal udvikles som en React app. Du skal selv fastlægge brugergrænsefladen, men funktionaliteten skal være som anført nedenfor.

Der er ikke krav om login, og som server bruges en lokal json-server som vist i lektion 19 "React Fetching data" samt i lab 22 (<https://github.com/typicode/json-server>).

Overordnet beskrivelse

Før eksamen starter kan brugeren (eksaminatoren) oprette en eksamen og indtaste studienumre og navne på de studerende i den rækkefølge, de skal eksamineres.

Under eksaminationen kan eksaminator indtaste kommentarer og efter hver eksaminand indtastes karakteren.

Efter eksamen kan eksaminator se de studerendes karakterer samt et gennemsnit for holdet.

Alle data skal persisteres via et REST api som tilbydes af json-server.

Krav til funktionalitet

1 Opret eksamen

Brugeren af appen skal kunne vælge opret eksamen hvorved følgende informationer kan indtastes og lagres i databasen:

- Eksamenstermin (f.eks. "sommer 25")
- Kursusnavn
- Dato
- Antal spørgsmål
- Eksaminationstid i hele minutter
- Starttidspunkt (klokkeslæt)

2 Tilføj studerende

Brugeren af appen skal kunne vælge en eksamen og derefter vælge tilføj studerende, hvorefter der kan indtastes studienummer og navn på studerende i den rækkefølge de skal eksamineres i.

3 Start eksamen

Brugeren kan starte eksamen for en oprettet eksamen, hvorefter studienummer og navn for den første studerende vises.

4 For hver studerende (eksaminand) er forløbet således:

4.1 Brugeren trykker på "Træk spørgsmål" hvor efter appen kommer med et tilfældigt tal mellem 1 og antallet af spørgsmål for denne eksamen.

4.2 Brugeren trykker på "Start eksamination" hvorefter programmet starter en timer som tæller ned fra eksaminationstiden, og som markerer visuelt og evt. med lyd når eksaminanden har brugt tiden.

4.3 Mens eksaminationen forgår og indtil der er indtastet (valgt) en karakter, kan brugeren indtaste noter til eksaminandens eksamen.

4.4 Brugeren trykker på "Slut eksamination" hvorefter programmet stopper timeren og den faktiske eksaminationstid registreres.

4.5 Brugeren indtaster (eller vælger fra en liste) den studerendes karakter, hvorefter de registrerede oplysninger (spørgsmål, faktisk eksaminationstid, noter og karakter) persisteres.

4.6 Brugeren trykker "Næste studerende" hvorefter der forsættes fra 4.1, hvis der er flere studerende. Hvis der ikke er flere studerende på listen, så får brugeren en meddelelse om at eksamen er færdig.

5 Se historik

Hvis brugeren vælger "Se historik", så kan der vælges en tidligere afholdt eksamen, og brugen får så vist en liste med de studerende og deres registrerede oplysninger samt et beregnet gennemsnit.

Forslag til strukturering af datafil, som definerer API'et (db.json) er vist på næste side. Bemærk, at dette kun er vejledende.

```
{
  "exams": [
    { "id": "1",
      "examtermin": "sommer 25",
      "courseName": "Introduktion til Programmering",
      "date": "2025-06-25",
      "numberOfQuestions": 10,
      "examDurationMinutes": 15,
      "startTime": "09:00" },
    { "id": "2",
      "examtermin": "sommer 25",
      "courseName": "Avanceret Programmering",
      "date": "2025-06-26",
      "numberOfQuestions": 15,
      "examDurationMinutes": 20,
      "startTime": "09:00" }
  ],
  "students": [
    { "id": "1",
      "exam": "1",
      "studenNo": "123456",
      "firstName": "Alice",
      "lastName": "Andersen",
      "questionNo": 1,
      "examDurationMinutes": 15,
      "notes": "Svarer rigtig godt på spørgsmålet, men har dog en fejl i koden",
      "grade": "10" },
    { "id": "2",
      "exam": "1",
      "studenNo": "654321",
      "firstName": "Bob",
      "lastName": "Bakker",
      "questionNo": 7,
      "examDurationMinutes": 16,
      "notes": "God redegørelse med kun ubetydelige mangler",
      "grade": "12" }
  ]
}
```