Programmering og udvikling af små systemer samt databaser

EKSAMENSOPGAVE FORÅR 2024 eksamen

Forord

Denne opgavebeskrivelse er udarbejdet af Regitze Sdun og Stefan R. Jørgensen til brug ved eksamen i faget Programmering og udvikling af små systemer for foråret 2024.

Eksamensprojektet skal udarbejdes i grupper af 2-4 personer.

Eksamensprojektet bygger på pensum for hele kurset, og I vil kunne finde inspiration og løsninger til store dele af jeres eksamensprojekt fra tidligere ugeopgaver og godkendelsesopgaver. Vær opmærksom på at nogle af emnerne i eksamensopgaven først bliver gennemgået efter udlevering af eksamensopgaven herunder testing.

I skal igennem arbejdet med jeres eksamensprojekt vise de kompetencer frem, som I har opnået igennem kurset. Vi tester derved jeres evner til at arbejde med de teknologier som vi har arbejdet med i kurset: HTML, CSS, JavaScript, Node.js og SQL. Opgaven binder flere at de emner, som er introduceret igennem kurset sammen og udmærker sig ved at det skal afleveres som en virkende webapplikation. Vi forventer derfor, at I kan leve op til alle fagets læringsmål.

Eksamensopgaven er en udvidelse af eksamensopgaven fra efteråret, hvor I nu skal inkorporere det, som I har lært om programmering og databaser i jeres løsning af det oprindelige problem.

Det forventes at man designer og udvikler løsning sammen i gruppen. Det betyder også at alle skal kunne stå inde for alt der bliver afleveret, dette vil blive prøvet til den mundtlige eksamen. Husk også, at hvis I benytter kode fra tutorials, eksterne biblioteker eller lignende skal dette angives med en kildehenvisning.

I må gerne diskutere løsninger på opgaven, men kopier ikke kode fra andre, da alt kode vil blive plagiatkontrolleret.

Spørgsmål

Hvis der skulle være spørgsmål omkring formalia til opgaven bedes lagt på Canvas, så alle kan få glæde af svaret.

Hvis der ved kursets sidste forelæsning stadig er elementer i opgaven som er uklare, vil der være en spørgesession, hvor der kan stilles spørgsmål til emner eller omfang.

Opgaven skal til enhver tid overholde CBS' formalia. Såfremt spørgsmålet omhandler personlige forhold, som for eksempel sygdomsperioder, skal disse sendes til studieadministrationen.

Introduktion

I arbejder stadigvæk for Magnus Lindholm, som vil skabe webapplikation "NutriTracker", der skal gøre det nemt for brugerne at spore deres daglige næringsindtag, dykke ned i de komplekse komponenter af deres måltider og gøre sunde valg lettere tilgængelige.

MVP'en fra efteråret har vist sig at være en succes og investorerne er overbevist om, at forretningsideen er værd at investere i.

Jeres projektleder har derfor bedt jer om at opskalere jeres løsning, så at investorerne bag NutriTracker kan få adgang til den dyrebare data bag servicen. Dette nødvendiggør dog at løsningen ikke længere skal gøre brug af Local Storage men i stedet være baseret på en rigtig relationel database som kun skal være tilgængelig gennem et veldefineret API.

Derudover har han bedt jer om at tilføje tracking af motions til app'en, hvor man kan holde øje med, hvor mange brugere, der er på appen, og hvordan de bruger app'en. Således at han løbende kan rapportere tilbage til investorerne på app'ens success.

I skal derfor fortsætte arbejdet med at udvikle app'en i Node.JS med følgende struktur

- Frontend (JS, HTML, CSS)
- Server/API (Node.JS & express)
- Storage (SQL server)

Kravene for disse enkelte dele fremgår under tekniske krav til besvarelsen.

Funktionelle krav til besvarelsen

Alle krav skal løses i med udgangspunkt i det datasæt, der er stillet til rådighed for opgaven og beskrevet i afsnittet Data

1. **Bruger styring**
    a. App'en skal tillade en bruger at oprette en profil
    b. App'en skal tillade en bruger at slette sin egen profil
    c. App'en skal tillade en bruger at opdatere sine personlige detaljer for sin egen profil herunder vægt, alder og køn
    d. App'en skal tillade brugeren at logge ind
    e. App'en skal tillade at hvis en bruger er logget ind, kan de forblive logget ind ved sideskift.

2. **Meal Creator:**
    a. Det skal være muligt for en bruger at oprette et eller flere måltider, som skal bestå af 1 eller flere ingredienser.
    b. Det skal være muligt at søge sig frem til den rigtige ingrediens.
    c. For hvert måltid skal der udregnes samlet ernæringsindhold (Energi, Protein, Fedt og Fiber) baseret på alle de fødevarer, som bliver brugt i opskriften.
    d. Det skal være muligt at se hvilke ingredienser, hvert måltid består af, samt vægten af hver ingrediens i måltidet.
    e. Det skal være muligt at finde information om hver enkelt fødevare i datasættet samt dets samlede ernæringsindhold.

3. **Meal Tracker:**
    a. Det skal være muligt i Appen at registrere ens næringsindtag som vægt af en given opskrift, som er skabt i Meal Creator.

b. Det skal være muligt at registrere indtag af en enkelt ingrediens udenom måltiderne fra Meal Creator.
c. Der skal være en visning af ens indtag som inkluderer hvornår det er indtaget, vægten og ernæringsindholdet.
d. Alle Registreringer skal indeholde dato og tid for indtaget. Disse skal være autoudfyldt med nuværende dato og tid.
e. Alle registreringer skal have en lokation, der kan sættes for indtaget. Den skal hentes fra browserens Geolocation API.
f. Det skal være muligt at redigere og slette ens registreringer.

4. Activity Tracker

a. App'en skal tillade en bruger at registrere de aktiviteter, hvor man forbrænder kalorier https://www.bodylab.dk/shop/saa-mange-kalorier-1116c1.html
Her må du antage at forbrændingen er den samme uanset køn, alder og vægt
b. App'en skal beregne basalstofskifte baseret på vægt, alder og køn for hver bruger ud fra følgende formel: https://nexs.ku.dk/forskning/vidensbanken/basalstofskifte/

5. Daily Nutri:

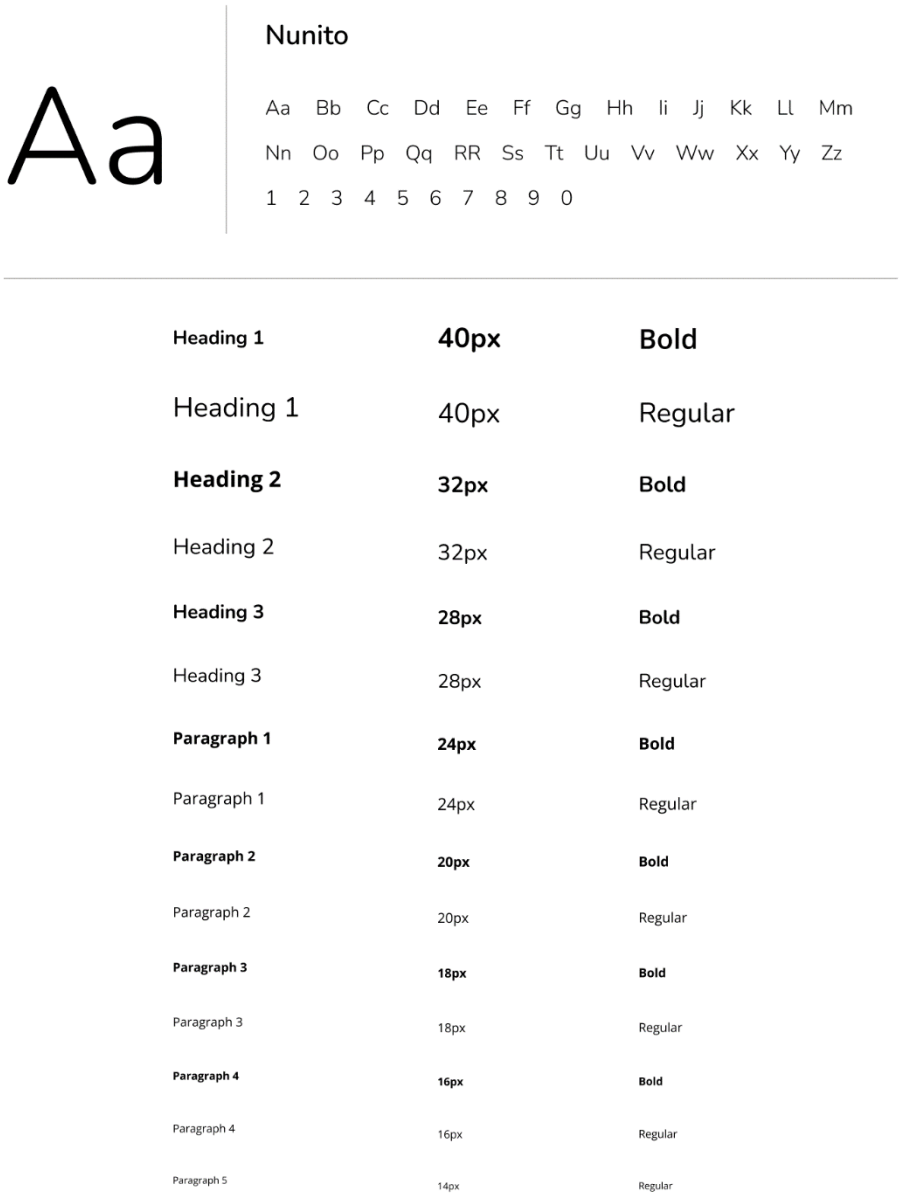
a. I tabelform skal I vise for hver time på døgnet over de sidste 24 timer
    - Indtag af energi
    - Indtag af vand
    - Forbrænding (basalforbrænding delt med 24 + puls aktivitet i det tidsrum)
    - Kalorie overskud/underskud

b. Det skal være muligt at vælge dags view i stedet for 24 timers view, så vil man se det per dag for den sidste måned.

Krav til webapplikationens udseende

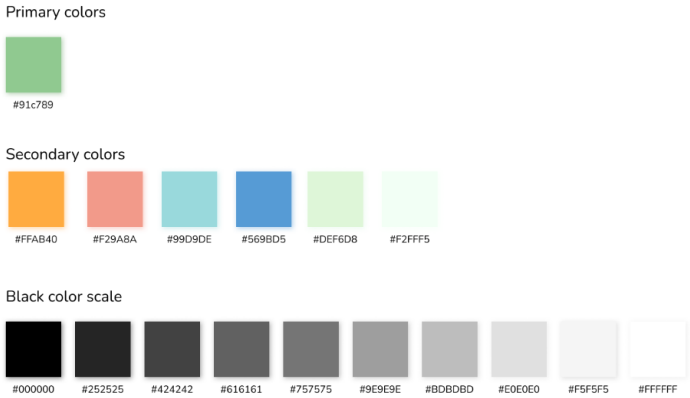
- Der er til opgaven givet et designforslag/mockup, dette er ikke ændret siden den første prototype. Webapplikationen skal bruge temaet som er givet i designforslaget samt tilhørende typografi og farvekort.

  Typografi, som skal anvendes:

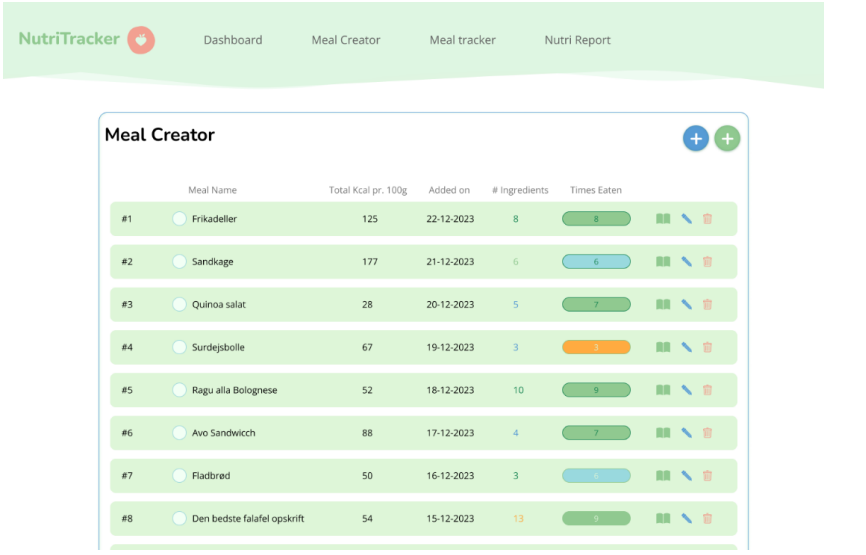


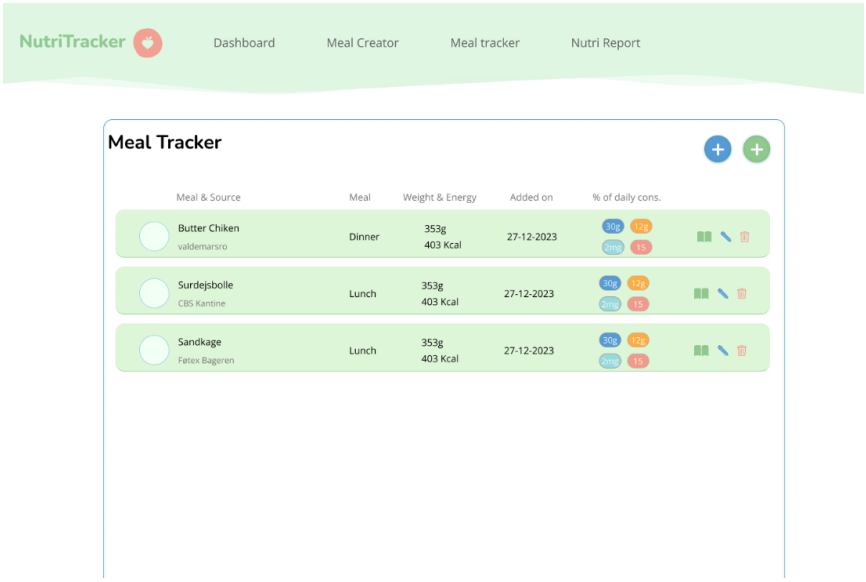
https://fonts.google.com/specimen/Nunito

Farver, som skal anvendes:



Inspiration til design





Det forventes, at det endelige produkt ligner design forslaget i form, stil og udtryk.

Formelle **krav til besvarelsen**

1. Rapporten skal afleveres som PDF via Digital Eksamen.
2. Alle sider i rapporten – inklusive forsiden - skal være fortløbende nummererede.
3. Antallet af sider skal fremgå af forsiden.
4. Rapporten for Programmering og udvikling af små systemer samt databaser må højst fylde 30 normalsider. Alt, hvad der ligger ud over disse sider, vil ikke blive taget i betragtning ved bedømmelsen. Se regler for optælling af sider på my.cbs.dk. Er den kortere er det også fint, I bliver bedømt på indhold, ikke længde. En fyldestgørende rapport kan derfor godt skrives på 15 sider.
5. Antallet af normalsider dokumenteres ved f.eks. udskrift af rapport fra tekstbehandlingsprogrammets statistik-funktion.
6. Figurer i rapporten skal være tydelige og læselige.
7. Rapporten skal være gennemlæst og må ikke indeholde stavefejl.
8. Jeres kildekode skal kommenteres og vedhæftes afleveringen som en zip-fil, som også skal indeholde eventuelle andre relevante bilag. Hvis der er tvivl i forhold til plagiat, er det vigtigt at have kommentarer som forklarer hvad man gør som passer til koden.
9. I skal lave en video med lyd, hvor I gennemgår de forskellige krav, som er stillet i opgaven. Der skal være et link til videoen i rapporten, Som beskrevet i afsnittet Video gennemgang.

Tekniske krav til besvarelsen

Applikationen skal være en webapplikation, men den del, som I skal udvikle, skal bare køre i browseren. Der er altså ingen krav om en distribueret applikation. Applikationen skal udvikles som en client-facing front-end applikation. I denne opgave skal I fokusere på at udarbejde funktionalitet og brugervenlighed så applikationen kan bruges som en prototype.

Jeres system skal bestå af tre dele:

1. Frontend (JS, HTML, CSS)
2. Server/API (Node.JS & express)
3. Storage (SQL server)

1 - Klienten

Jeres klient laves i helt simpelt Javascript, HTML og CSS. Vi forventer at I kommenterer koden, hvis det giver mening og beskriver, hvordan forskellige stykker af koden løser diverse krav.

2 - Serveren

I skal udarbejde serveren i express med Node.js. Arbejdet går på at udarbejde et API, som kan benyttes i forbindelse med jeres klient. API'ets hovedfunktion er at forbinde klienten og databasen og dermed være mellemled (backend) imellem de to.

Applikationen udstiller et HTTP-interface, som kan give svar på HTTP-requests og sende svar tilbage. Dataudvekslingsformatet skal være JSON.

I skal lægge vægt på at opbygge et velgennemtænkt modellag, som danner grundlaget for jeres applikation.

3 - Storage

For at opfylde kravene til projektet er det nødvendigt at gemme noget data. Dette data forventes gemt i en normaliseret T-SQL

database sådan, at det er nemt at udvide med mere data senere - skulle dette være nødvendigt.

De valg, I gør jer for storage skal dokumenteres i jeres rapport. Dette omfatter f.eks. Et Entity-Relations Diagram: Altså hvordan I har valgt at strukturere jeres data.

Eksekvering

I skal kun dokumentere den teori, som vi har gennemgået i pensum (grundbog, opgivne websider, forelæsninger samt øvelser). Teori uden for pensum belønnes ikke og kan i værste fald trække ned, fordi man ikke får dækket de andre vigtige områder. Brug derfor ingen javascript frontend frameworks til løsningen.

Herudover skal I følge normal god kodeskik og dokumentere jeres kode med kommentarer og en fornuftig filstruktur. Koden skal være nem at vedligeholde for andre efter aflevering, og koden skal naturligvis være versioneret ved brug af Git.

Data

Da det ernæringsmæssige ikke er det centrale i opgaven, har vi gjort dette tilgængeligt for jer. Data for de forskellige ingredienser er gjort tilgængelig på:

https://nutrimonapi.azurewebsites.net/index.html

På siden er der en API-definition for de forskellige end points, som I skal bruge for at løse opgaven.

I vælger selv om I vil

- kalde det API'et fra jeres webapplikation og have en database til at understøtte activity tracker og daily nutri
- kopierer data fra API'et ind i din egen database

Alt data til brug i webapplikationens forskellige funktioner skal hentes via et web API, som I selv skal udvikle.

Materialer og gode råd

Som baggrund for at gennemføre opgaven anvendes denne opgavebeskrivelse samt fagets pensum (grundbog, opgivne websider, forelæsninger samt øvelser og opgaver). I kan derudover søge oplysninger i andre offentligt tilgængelige kilder (**HUSK** at angive kilde med APA-modellen).

Der opfordres til at I benytter forløbets forskellige muligheder for at diskutere projektet og de problemstillinger, I møder undervejs. Dertil opfordres I til at sparre med hinanden og tilbyde gennemsyn af hinandens løsninger. Ofte ender man med at se sig blind i det, man selv laver, hvorfor det kan være nyttigt med et sæt friske øjne.

I er velkommen til at poste tekniske problemer på Canvas, hvor vi gerne vil hjælpe. Når I spørger om hjælp, er det krav at I som minimum beskriver, hvad I allerede har forsøgt for at løse problemet. Vi forventer at I også hjælper hinanden og byder ind i forhold til hinandens problemer.

Det er vigtigt at I holder jer for øje, hvad dette fag fokuserer på og dermed undlader elementer, som for dette fag ikke er relevante.

I skal lægge vægt på refleksion og diskussion, hvilket bør vægte højere end beskrivelse. Såfremt I benytter teori, er det derfor vigtigt at I ikke blot beskriver teorien, men reflekterer over brug af teorien og dermed konkretiserer relevansen for netop jeres projekt.

Rapporten

Nedenstående er anbefalinger listet til de enkelte del-elementer i rapporten:

Kravspecifikationen er en del af problemfeltet, i denne opgave får i dette givet og i skal ikke selv bruge ressourcer på at skabe disse. I afsnittet om tekniske krav, skal I lave en tabeloversigt over hvilke krav, der er opfyldt og hvilke der ikke er.

Løsningsovervejelser skal være en tungtvejende del af jeres opgave. Vi forventer en beskrivelse af, hvordan I forholder jer til eventuelle uklarheder i opgaveteksten samt de overvejelser, I har gjort jer om strukturen for design og implantation, samt eventuelle erkendte begrænsninger og bevidste fravalg i funktionalitet og udførsel. En god løsningsovervejelse laves på baggrund af det analyseret materialer fra tidligere afsnit.

Entity Relationship Diagram skal bruges til at give en oversigt over jeres data og hvordan jeres database er designet. Diagrammet giver et hurtigt overblik over, hvordan I har valgt at gemme jeres data. Det forventes derfor, at I vedlægger et ER diagram i rapporten samt argumenterer for, hvorfor det ser ud, som det gør (Hint: det er oftest en god ide, at det er normaliseret til 3nf).

Tests er en vigtig del af softwareudvikling og derfor skal I også teste jeres program. Da tests er meget tidskrævende må I vælge et punkt fra den funktionelle krav specifikation, som I skal unitteste så grundigt som muligt. Dette skal dokumenteres i rapporten.
I skal kun dokumentere hvordan I unittester, beskrivelser af hvordan man debugger og andre typer tests gives der ikke point for.

Konklusion

I dette afsnit skal i konkludere om jeres løsning klarer de krav som er blevet givet tidligere.

Perspektivering
Dette afsnit kan i bruge til at beskrive hvordan i alternativt kunne være gået til opgaven. I kan også beskrive hvilke andre metoder I kunne have brugt eller hvilke andre løsninger som måske have været bedre? Hertil opfordres I til at angive løsningovervejelser for de tekniske krav, som ikke er blevet opfyldt.

Procesevalueringen har til formål, at I forklarer jeres læring i forløbet. I dette afsnit kan I for eksempel beskrive 2-3 hændelser, der er opstået undervejs i jeres projektforløb og hvordan I valgte at løse disse hændelser metodisk. Dette behøver ikke at være ting som var succesfulde, det er typisk lettere at fokusere på de fejl man har lavet.

Rapporten skal først og fremmest beskrive løsningen, dvs. produktet og de dertilhørende designvalg, i mindre grad den proces at nå frem til produktet bortset fra i de 2 sidste afsnit perspektivering og procesevaluerings som netop omhandler netop dette. Der er ikke påkrævet et metodeafsnit i denne rapport.

Video gennemgang
Som bilag til rapporten skal I vedlægge en kort (Max 10 minutter) video med lyd af jeres applikation. I denne video skal I gennemgå kravene og vise hvordan de er implementeret i applikationen. Dette kan gøres ved at klikke igennem jeres applikation og vise funktionaliteten. Brug Loom. Sørg for ikke at uploade videoen, men bare linke til den i rapporten. I skal ikke gå igennem kode i denne video. Bare de funktionelle krav som er givet i den rækkefølge de er givet.

Eksamen i Programmering og udv. af små systemer samt databaser

Eksamen udføres af fagets underviser med intern censur, og bedømmes efter 7-trins skalaen. Karakteren gives efter en samlet bedømmelse af rapporten og jeres kodebase

Bedømmelsen af rapporten vil lægge vægt på følgende:

- Disposition og rapport-systematik
- Jeres kodebase og konstruktionen heraf
- Jeres forståelse for tekniske og forretningsmæssige begrænsninger af jeres implementering, samt refleksion over de valg I har truffet heraf
- Beskrivelse af jeres forudsætninger, valg af perspektiv(er) og redegørelse for jeres valg
- Begreber og definitioner (deres præcision, konsistens og relevans)
- Teori- og metodeanvendelse
- Tabeller, figurer og diagrammer (deres informationsværdi, konsistens og relevans)
- Anvendt litteratur (omfang og dybde)
- Kildekritik (både teori og data)
- Forklaringer: redegørelse for årsag og virkning
- Procesevalueringens kobling til pensum
- Bilag (deres kvalitet og relevans)
- Rapportens omfang (den skal holde de formelle retningslinjer)
- Sprog, kildehenvisninger og struktur
- Match mellem afleveret rapport og produkt.
- Evt. afskrift / direkte reproduktion uden kildehenvisninger

Der gives karakter efter fagets læringsmål, som kan ses i kursuskataloget.