

Miniprojekt 2. semester

Formål

Formålet med projektet er at øge jeres færdigheder inden for nogle af de fagelementer, der indgår i eksamensprojektet. Miniprojektet giver jer mulighed for at arbejde projektbaseret og tværfagligt med krav, design og implementering af en webbaseret databaseapplikation.

Leverance

- En webbaseret databaseapplikation.
- Al koden til jeres projekt skal placeres i et Git repository på GitHub.
- Jeres GitHub repository skal indeholde et workflow i GitHub Actions som implementerer elementer af:
 - *Continuous Integration* (automatiseret build af feature-branches eller pull requests til main efterfulgt af relevante tests).
 - *Continuous Delivery* (deploy til Azure cloud (PaaS)).
 - Jeres GitHub repository skal dokumentere projektet, og som minimum indeholde:
 - ER model over databasedesignet Klassesdiagram som sl addedokumentation af programdesignet (*ikke* autogenereret, men på et passende detaljeringsniveau, hvor andre programmører rent faktisk vil kunne bruge det til at få overblik over systemet).
- Risikoanalyse – med ekstra fokus på de mitigerende handlinger
- Interessentanalyse - fyldestgørende, med alle de interessenter I kan komme på, samt hvordan de bør håndteres og hvorfor.
- Med udgangspunkt i jeres Business Model Canvas:
 - Redegør kort for, hvordan I inddrager bæredygtighed og dataetik i løsningen. Jeres bevarelse skal indeholde to dele: 1) På hvilken måde er jeres projekt bæredygtigt i en IT-sammenhæng? Inddrag de 5 P'er fra den Podcast, I har lyttet til i undervisningen, og de digitale FN verdensmål: Global Digital Compact fra FN. (Se side 8 her: <https://www.un.org/sites/un2.un.org/files/our-common-agenda-policy-brief-gobal-digi-compact-en.pdf>.) 2) Beskriv derefter, hvordan I har implementeret bæredygtighed og dataetik, eller hvad implementering ville kræve, hvis det lige nu ikke er en del af jeres løsning. Dette skal fylde 1 normalside.
- Scrum board - i GitHub Projects - med user stories og tasks, der viser projektstatus når I afleverer projektet. I skal fra start spille Planning Poker og estimere de enkelte User Stories med stories point/ eller i timer, hvis det er nemmere.
- SQL-scripts til oprettelse af database og indsættelse af evt. Testdata
 - Jeres SQL scripts placeres i 'resources' folderen i projektet
 - Lav gerne to scripts: et til oprettelse af databasen med tilhørende tabeller og et der indsætter test(start) data i tabellerne.
 - Scriptfilerne skal have extension '.sql'

Problem/udfordring

I skal udarbejde en digital ønskeseddel.

I kan finde inspiration på f.eks. ønskeskyen og wishlist.dk. Den tidligere addwish service (addwish.com) er ikke længere tilgængelig, men så er det jo godt at I kan komme med et bud på et nyt alternativ.

Krav til programmet

De funktionelle krav til applikationen er inddelt således:

1. Minimumskrav, som alle grupper bør implementere:
 - At man som bruger kan lave en digital ønskeseddel
2. Krav, som vil være ønskelige at få med, men som ikke er obligatoriske:
 - At man som bruger kan dele sin ønskeseddel med andre
3. Krav, som vi ikke forventer at mange grupper får tid til at implementere:
 - At andre brugere kan reservere ønsker på ønskesedlen

Produktet skal være et såkaldt [Minimal Viable Produkt](#) (MVP), men I forventes at holde fokus på høj intern kodekvalitet med så lidt [teknisk gæld](#) som muligt.

Gruppestørrelse

2-4 personer (helst 3-4).

Aflevering

Et klikbart link til GitHub repository samt GitHub navn på alle deltagere. I behøver ikke have jeres deployede applikation på Azure tilgængelig for underviserne i forbindelse med aflevering af jeres besvarelse. Men applikationen skal dog kunne tilgås via link på Azure i forbindelse med opfølgning på projektet i undervisningstiden.

Husk at projektet er en obligatorisk opgave, der skal afleveres for at kunne blive indstillet til eksamen.

Vejledning

Der er vejledning i henhold til Fronter planerne.