

Part 1 : Viden omkring Angular

Den første del af kodetesten handler om at fikse nogle angular-specifikke bugs i projektet, "Bad Angular Implementation". Bad Angular Implementation er et spil inspireret af Top Trumps spillene, hvor man sammenligner to kort med forskellige stats, hvor det kort med højeste stats, i en kategori, er vinderen.

Uheldigvis er spillet blevet implementeret af nogle personer, som har brugt næsten alle de anti design patterns, de kunne finde på.

Note: Du skal ikke ændre på Dummy Databasen, den er der kun for at efterligne en API med en tilkoblet database.

Du har fået til opgave at løse tre større problemer med projektet.

1. Hero Creator Crash:

Når du går ind på Hero Creator siden, er der noget forfærdeligt galt med formularen, man designer heroes med. Der er funktioner, som bliver kaldt et helt utalligt antal gange.

1. Sørg for at funktioner kun bliver kaldt efter behov.
2. Lav en beskrivelse omkring, hvad problemet var, og forklar hvordan du løste dette problem.

2. Battle Bug:

Hvis man går ind på Battle siden, skal det være muligt at få de forskellige helte til at kæmpe. Først bliver to tilfældige helte valgt, derefter skal man kunne trykke på "Start Fight" hvorefter en tilfældig characteristic bliver udvalgt. Helten med den højeste score vinder. Derefter skal knappen sige: "New Challenger", som skal udskifte helten som tabte, (eller begge helte, hvis det blev uafgjort). Hvor knappen så skal skifte tilbage igen til "Start Fight". Som det er nu, bliver knappen ved med at sige: "New Challenger", og når man trykker på knappen, udskifter den både heltene og udvælger en winner.

- a. Sørg for at funktionaliteten på siden fungerer, som er det hensigten.
- b. Lav en beskrivelse omkring, hvad problemet var, og forklar hvordan du løste det.

3. Organiser Chaos-Folder:

Som projektet er organiseret lige nu, er der ikke rigtig nogen god filstruktur. Alle filer ligger inde i en stor folder med titlen, Chaos-folder. Projektet her er et mindre projekt, så det er til at overskue. Det er dog meningen i fremtiden, at projektet skal udvides og dermed vokse sig meget større.

- a. Omorganisér filstrukturen, så den kan håndtere en videreudvikling, uden at blive uoverskuelig. Du kan enten løse denne opgave ved at omstrukturere filerne i selve projektet eller ved at lave din filstruktur på Draw.io. Der medfølger en draw.io fil til opgaven med titlen, FolderStructure. Her finder du den nuværende struktur.
- b. Beskriv fordelene af netop din filstruktur.
- c. Beskriv potentielle ulemper ved din filstruktur.
- d. Beskriv hvilke egenskaber, som du finder vigtigst i en fil struktur, og hvorfor.

Part 2: The Case Study

Første iteration

En blomsterbutik ønsker at få lavet en hjemmeside, hvor kunder kan bestille deres produkter, og hvor det er muligt at hente produkterne senere. Du har fået en uorganiseret liste af disse produkter, som virksomheden sælger. Blomsterbutikkens budget er ikke ret stort, så det er begrænset, hvor mange arbejdstimer, hjemmesiden må tage at udvikle. Listen af produkter findes i excelarket med titlen Blomsterbutik.

Opgaver

1. Beskriv hvilken tech stack du ville bruge
 1. Skriv hvilke fordele, der er, ved denne tech stack i forhold til netop dette projekt.
 2. Lav en kort sammenligning af dine valgte teknologier og en anden relevant teknologi. Hvad er forskellene? I hvilken situation ville du have valgt at bruge den anden teknologi, og hvorfor?
2. Design et UML databasediagram til måden, du har valgt at strukturere din database på.

Anden Iteration

Blomsterforretningen er nu vokset pga. din gode hjemmeside, og det samme har deres budget til udvikling. Derfor vil de nu gerne have, at hjemmesiden automatisk skal kunne give buket tilbud til kunder, som besøger siden. Hver buket er lavet ud fra de blomster, som de har på lager, og som er ved at blive for gamle.

Du får nu også noget ekstra information omkring deres forretning og drift. Du ved nu:

- I starten af hver måned køber butikken det antal blomster ind, som de ikke har på lager. De har adgang til den præcise mængde blomster, som de har købt, og de ved, præcis hvornår blomsterne blev leveret til butikken.
- Ca. hvor lang tid hver type blomst kan holde sig i butikken, før den/de begynder at se for døde ud til at kunne sælges.
- Hex kode farverne på de to primære farver af hver blomst.
- Data'en for hvert salg, så du har styr på hvor mange blomster, de har på lager
- De sælger altid de nyeste blomster først.
- Buketter ser bedst ud, hvis deres farver enten er Analogous eller Complementary.

Opgaver

1. Vil du foretage nogle ændringer i din tech stack, og hvorfor/hvorfor ikke?
2. Beskriv hvordan du vil designe implementationen af denne funktionalitet. Hvilke metoder skal du bruge, hvad skal køre på klienten, serveren og databasen?
3. Gennemgå hvilke designovervejelser du havde.
4. Implementer en del af dit system. Vælg gerne en del, du mener, viser noget interessant kode/design. Du må gerne lave mock funktioner for resten af systemet efter behov, så længe du har en kommentar omkring, hvordan funktionen burde fungere.