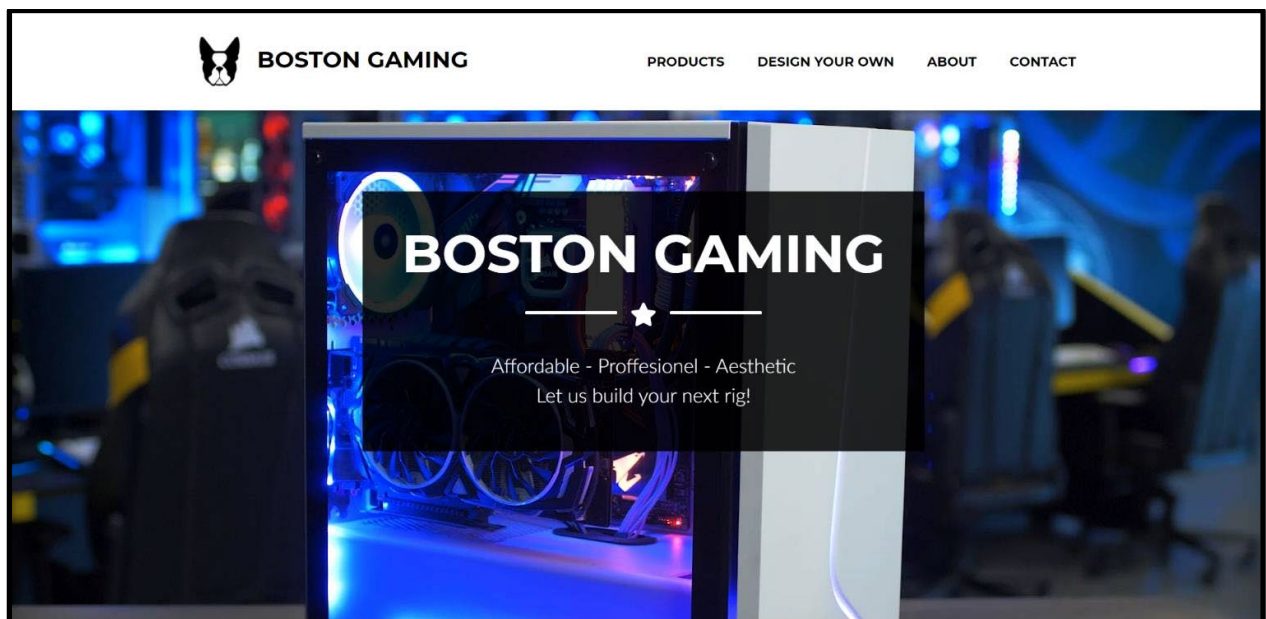


Boston Gaming

Webudvikler Eksamen
Viden Djurs - VID Medie



Klasse: webith219A

Uge 45, november 2020

Indholdsfortegnelse

Indholdsfortegnelse.....	2
Vigtige generelle oplysninger om eksamen	2
Situationsbeskrivelse	3
Opgavebeskrivelse	3
Tekniske krav.....	3
Web API	4
Billeder	4
Projekt-rapport	4
Bilag	5
Forsiden	5
Generelt om rapporten	5
Kravspecifikation.....	5
One Page	6
Administration af content/database-indhold.....	7
Tilvalgsopgaver.....	7
Prioritetsliste	8
Aflevering.....	8

Vigtige generelle oplysninger om eksamen

- Afleveres opgaverne ikke til tiden, er prøven ikke bestået. Ved systemnedbrud eller andre tekniske problemer, på skolens område, kan tiden forlænges, men kun i særlige tilfælde og ifølge aftale.
- Du skal selv sørge for, at din computer er opdateret og klar til eksamen, og du skal løbende lave backup mv. af dine koder, så et evt. nedbrud af udstyr ikke påvirker dit resultat nævneværdigt.
- Alle hjælpemidler, der normalt anerkendes og benyttes i undervisningen, er tilladt.
- Det er tilladt, med tydelig kildeangivelse, at benytte funktioner/objekter (f.eks. JavaScripts) udarbejdet af 3. part. Indholdet af disse vil dog ikke tælle med i bedømmelsen af dit projekt.

Situationsbeskrivelse

Du er ansat i et webbureau, som udvikler dynamiske webløsninger. Du indgår i et udviklingsteam, der består af en designafdeling og en kodeafdeling.

Bureauet har fået en ordre på et website fra en virksomhed. Layoutet og sitets generelle struktur er allerede fastlagt og godkendt af kunden.

Din opgave bliver at kode sitets funktionalitet – herunder frontend-delen samt dele af administrations-delen til indholdet.

Opgavebeskrivelse

Boston Gaming har et ønske om at få en mere synlig profil på Internettet. Derfor ønsker firmaet at få bygget et website, der kan indgå i deres samlede markedsføringsstrategi.

Formålet med sitet er at præsentere firmaets mærkesager på en overskuelig måde, samt give brugere mulighed for let og overskueligt at finde informationer. De ønsker derudover også en ny side til at præsentere deres produkter på en professionel måde samt en måde at kategorisere produkterne på.

Tekniske krav

Designfil udleveres sammen med opgaven. Som udgangspunkt skal du følge designet i opsætningen, men kunden er modtagelig for ændringer i designet, hvis disse kan argumenteres til kundens fordel.

Sitet skal som minimum opbygges med **HTML, CSS og Javascript, og det skal kunne afvikles** på en lokal og evt. webbaseret serverløsning efter eget valg.

Væsentlige dele af sitets indhold (tekst, billeder mv.) skal hentes/administreres fra det udleverede API – læs mere under "Kravspecifikation".

Sitet skal opfylde de gængse standarder inden for **semantisk, korrekt kode** og i tilstræbt **søgemaskineoptimering** (SEO) og **accessibility**.

Sitet skal være **responsivt**.

Sitet bør indeholde fejlhåndtering på alle relevante fejl, så brugeren får en **fejlmeddelelse** der tager hensyn til sitets indhold.

Der skal være relevant **validering** på alle formularer/inpotelementer.

Web API

Dit site skal hente alle væsentlige data (tekst, billeder mv.) fra et API/en backend som udleveres sammen med opgaven.

API'et kører på MongoDB, Express JS og Node JS, og der er vedlagt en beskrivelse i backenden (mappen _INSTALLATION), hvordan API'et/backenden installeres og startes på egen/lokal computer. I samme mappe ligger der en ReadMe.txt – her er generel information om brug af backend/indhold fra backend, login mv.

Når databasen er oprettet og API'et er startet kan du benytte de indbyggede GET, POST, PUT og DELETE metoder, som det fremgår af **rest**-filerne i backenden samt i den udleverede **Postman**-collection.

Du må hente dataene og køre det fra en anden selvvalgt løsning, hvis du foretrækker det. Du må også gerne udvide/ændre i det eksisterende API. **Ændrer du i API'et/backenden skal du notere dette i din dokumentation/projektrapport: HVAD du har ændret/tilføjet – og hvorfor. OG du skal aflevere/linke til den ændrede backend.**

Billeder

Billeder til produkter, slider mv. ligger i backendens "public"-mappe:

Produktbilleder fx: <http://localhost:5039/images/product/xxxx.jpg>

Sliderbilleder fx: <http://localhost:5039/images/slider/xxxx.jpg>

Projekt-rapport

Du skal udarbejde en projektrapport løbende med opgaveløsningen.

Projektrapporten skal som minimum indeholde følgende:

- Vurdering af din **egen indsats** og gennemførelse af opgaveforløbet
- Argumentation for de **valg du har truffet** under løsningen af opgaven
 - herunder hvad du eventuelt har ændret/tilføjet i backend/API
- Beskriv hvilke af **tilvalgsopgaverne**, du har løst
- Liste over **kodeelementer udarbejdet af 3. part** i opgaveløsningen – fx Bootstrap, Sass/Less, færdige scripts/plugins (fx lightbox, gallerier, scroll-funktionalitet mv. som du ikke selv har udviklet)
- Angivelse af de **url-adresser, brugernavne og passwords**, der er nødvendige for at lærer og censor kan se/køre din løsning
- Evt. **link til dit færdige projekt** på GitHub (indsættes på forsiden)

- En beskrivelse af eventuelle **særlige punkter til bedømmelse** – fx hvad du har udvidet med af ekstra funktioner, designelementer mv.
- Har du foretaget **ændringer i API/backend**, skal du nævne i din rapport, hvad du har ændret og hvorfor (og husk at aflevere en kopi af/link til dit API).

Bilag

Bilag til rapporten:

- Din **tidsplan** (estimeret og faktisk tid)
- Evt. uddrag af **særlige kode-eksempler**, hvis det er relevant

Forsiden

På forsiden af rapporten skal fremgå:

- Opgavens navn (Boston Gaming), dit navn, holdnummer, skolens navn og afleveringsdato og tidspunkt

Generelt om rapporten

- Omfanget af rapporten skal være på maksimum 5 A4-sider eksklusive bilag
- Projektrapporten **afleveres som en pdf-fil**
- Projektrapporten skal **placeres i projektets rodmappe** ved aflevering
- Rapportens navn SKAL være **rapport.pdf**

Kravsifikation

Herunder er et overblik over krav til sitets indhold:

Med opgaven er der udleveret en jpg med layout, som understøtter disse krav.

Hvis du finder elementer i layoutet, som ikke er nævnt i teksten, skal du antage, at det skal laves. Det er op til dig, hvordan du vil løse disse elementer.

Websitets indhold (tekst, billeder mv.) skal i videst muligt omfang hentes fra API'et/backend.

Generelle krav

- Sitet bygges i **standardmål** efter layoutet
- Sitet skal være **responsivt** – du bestemmer selv, hvordan løsningen reagerer i forskellige device-størrelser/breakpoints, men du skal kunne argumentere for din valg

Faste elementer på alle sider

- **Logo** (udleveret sammen med layout-filen)
- **Navigation**
- **Footer** med kontakt informationer
- **Søgefelt** (placeres efter eget valg - selve søge-funktionen er en tilvalgsopgave)

One Page

- **Header**
 - Med slider fra API
- **Produkter**
 - Produkter og kategorier hentes fra API'et
 - Hvert produkt tilføjet et link ("læs mere") til produkt detaljer (se herunder)
 - Produkterne inddelt i **kategorier** (fremgår ikke af layoutet – du har frie hænder her)
- **Produktdetaljer**
 - Produktdata (tekster, kategori, billede mv.) hentes fra API'et og vises i en **modal**, selvstændig **page/component** eller lignende efter eget valg
- **Design din egen computer (Design your own rig)**
 - Se layout: Lav en **prototype**, hvor brugeren kan vælge/"sammensætte" en computer (CPU, motherboard, memory osv.). **Summary** viser brugerens valg og den samlede pris
 - Du kan hente **tekstbeskrivelserne** (Gear) fra Postman og kopiere teksterne ind i din kode
 - Du kan evt. vælge at trække data (tekster og kategorierne) fra API'et – dette er en tilvalgsopgave (læs mere under afsnittet om tilvalgsopgaver)
- **Om os**
 - Data hentes fra API
- **Kontakt os**
 - Den udfyldte formular skal POST'es til API/databasen
 - Husk validering af formularens input
 - Efter afsendelse skal brugeren informeres om, at beskeden er sendt og hvilke data brugeren har sendt
- **Footer**
 - Data hentes fra API

Administration af content/database-indhold

Du bestemmer selv, hvordan administrations-delen skal designes – men der skal være en **designmæssig sammenhæng mellem frontend og administration**.

Der skal minimum være administration på følgende dele:

- **Product: Opret, ret og slet**
 - gerne med brug af RTE/editor på produktets tekst/[content](#)
- **Slider: Opret og slet**
- **About: Ret**
 - gerne med brug af RTE/editor på [about](#)-teksterne
- **Footer: Ret**
 - gerne med brug af RTE/editor på [about](#)-tekst og [location](#)
- **Contact** (beskeder sendt fra brugere): **Slet**

Tilvalgsopgaver

Du skal vælge minimum 2 af følgende tilvalgsopgaver. Du må dog gerne vælge flere, såfremt øvrige krav er opfyldt:

1. Login

Lav login-beskyttelse af administrations-delen, så kun brugere/users med e-mail og password (administrator) kan få adgang.

Følgende bruger er oprettet i systemet: Email: jn@testmail.dk, password: [admin123](#).

Du kan selv oprette flere brugere med Postman eller login.rest (i backenden).

2. Tilmeld nyhedsbrev

Brugere skal kunne tilmelde sig BostonGamings nyhedsbrev - du har frie hænder til at designe hvor og hvordan.

I administrationsdelen skal administrator kunne se en **oversigt** over alle tilmeldinger – med mulighed for at **slette** tilmeldinger.

3. Søgning

Søgefeltet skal tilføjes funktionalitet, så brugere kan søge i produkter (titel og indhold). Du har frie hænder til at designe søgeresultat-siden – som popup, modal, udvidelse af eksisterende one-page, en ny page el.lign.

4. Administration af “contacts” (beskeder fra kontaktformularen)

Der skal i administrations-delens oversigt over kontaktformular-beskeder være mulighed for at markere/gemme beskeder/”contacts” som læst/”read” eller ulæst/”unread” (API’et er forberedt til det med et felt “read” som kan være true eller false).

5. Administration af “users”

Fx en bruger, der er logget ind, skal have adgang til at kunne rette egne oplysninger (skifte password, email) og slette sin ”profil”. Og brugeren skal kunne oprette nye brugere. Udvid evt. selv med flere muligheder.

6. Prototypen ”Design your own rig” baseret på data fra API

Prototypen baseres på data hentet fra API’et (der er oprettet metoder til at hente/GET ”gears” og ”gearcategories”).

Ekstra

Har du løst minimum 2 af tilvalgsopgaverne og opfyldt opgavens øvrige krav, har du frie hænder til at udvide med yderligere funktioner – men de skal være relevante, og du skal kunne argumentere for dine valg.

HUSK at notere i rapporten hvilke tilvalgsopgaver, du har løst.

Prioritetsliste

Herunder finder du en liste over, hvordan du bør prioritere i opgaven:

1. **Generel opsætning af design** - sitet skal fremstå færdigt i html og css og matche det udleverede layout
2. **Integration af tekst og billeder** – udtræk data fra API som beskrevet i afsnittet ”One Page”
3. **Produktets detaljeside** (el. modal) – vigtig for kunden, da det er en stor del af deres trafik er afhængig heraf (husk kategori)
4. **Slider**
5. **Kontaktformular** med POST-funktionalitet, validering mv.
6. **Administration af data** som beskrevet i afsnittet ”Administration”
7. **Tilvalgsopgaver** – minimum 2 af opgaverne som beskrevet i afsnittet ”Tilvalgsopgaver”
8. **”Design your own rig”-prototype**

Aflevering

Sidste frist for aflevering er: **fredag den 6. november 2020 kl. 14:45.**

VIGTIGT: Din løsning skal præsenteres til den mundtlige eksamen på en lokal eller webbaseret serverløsning efter eget valg. Det er **vigtigt, at du ikke redigerer i denne løsning efter afleveringsfristen er udløbet.**

Desuden skal du ifølge GDPR lovgivningen **give accept til, at eksaminator og censor må beskue opgaveløsningen på din lokale computer.**

Du skal aflevere følgende på **fællesdrevet** i en mappe med dit navn:

- Mappe med dit **webprojekt** - dog UDEN mappen node_modules!
- **Rapport** som pdf (rapport.pdf placeret i roden af webprojektet)
- **Backend/API** hvis du har foretaget ændringer i det udleverede

Sti på fællesdrevet:

- \\viddata\uv-faelles\Elever VID Medie\Holdmapper**webith219a**\EKSAMEN-aflevering

God arbejdslyst!