|  |
| --- |
|  |
| Eksamen |
| Visualisering 3.1 – Web Development |

|  |  |
| --- | --- |
| Formalia: |  |
| Afdeling | AspIT BL |
| Underviser | Klement Johansen |
| Censor | Preben Bisgaard |
| Forløb | Modulperiode 3 forår 2018 |
| Opgaven udleveres | Mandag den 25. juni 2018 kl. 09.00 |
| Opgaven afleveres | Onsdag den 21. juni 2018 kl. 12.00 |
| Der arbejdes følgende dage i tidsrummet | Mandag kl. 9:00 – 15.00  Tirsdag kl. 8.45 – 15.00  Onsdag kl. 8.45 – 15.00 |

Indholdsfortegnelse

[1 DUMB STARBUCKS COFFEE 3](#_Toc517645691)

[1.1 Indledning 3](#_Toc517645692)

[1.2 Kravspecifikation 3](#_Toc517645693)

[1.2.1 Overordnede krav 3](#_Toc517645694)

[1.2.2 Krav til siden 5](#_Toc517645695)

[1.2.3 Krav til koden 5](#_Toc517645696)

[2 Ekstra opgaver 7](#_Toc517645697)

[3 Regler i eksamenslokalet 8](#_Toc517645698)

[4 Bedømmelseskriterier og evaluering 9](#_Toc517645699)

[4.1 Faglige læringsmål (70%) 9](#_Toc517645700)

[4.2 Personlige læringsmål (10%) 9](#_Toc517645701)

[4.3 Arbejdsevnemæssige læringsmål (20%) 10](#_Toc517645702)

[4.4 Eksamen 10](#_Toc517645703)

[4.5 Karakterskala 11](#_Toc517645704)

# DUMB STARBUCKS COFFEE

## Indledning

Ejeren af den nystartede parodi kaffebutik DUMB STARBUCKS COFFEE har bedt dig om, at lave en simpel shop til dem hvor deres kunder kan forudbestille kaffe for senere at hente den. På et tidspunkt skal siden hente data om de forskellige kaffer fra en anden kilde, men da denne service stadig er under udvikling kan du bruge data fra en inkluderede produkter.json fil.

## Kravspecifikation

I det følgende kan du finde de forskellige krav, der er til websiden, og til opgaven. Kravene er opdelt efter overordnede krav, krav til dokumentation og krav til webshop.

### Overordnede krav

For websiden gælder der følgende overordnede krav, som SKAL overholdes.

* Under udvikling skal du løbende anvende versionsstyringsværktøjet Git. Der skal oprettes et repository på GitHub som du løbende holder opdateret. Når du er færdig med opgaven skal du sende et link til dit repository til din underviser.
* Der er udleveret et screenshot der viser hvordan siden skal se ud. Din side behøver ikke ligne præcist det udleverede billede.
* Du skal til opgaven bruge et front-end framework som bootstrap, materializecss eller lignende.
* Skal overholde HTML5 standarder.
* Løsningen skal være responsiv så den tager sig godt ud på mobiler, tablets og PC.
* Du skal til eksamen kunne redegøre for:
  + Hvordan du praktisk har anvendt et front-end framework til at opstille sides layout og forskellige komponenter.
  + Hvordan og hvorfor du har lavet forskellige JavaScript klasser og metoder.
  + Eventuel opdeling af din kode i genanvendelige dele fx ved at opdele i Model-View-Controller. Det er ikke et krav med MVC men du skal vise, at du har taget aktive valg ift. hvordan din kode er opdelt.
  + Hvordan og hvorfor du har opstillet data i forskellige datastrukturer.
  + Hvordan du ændre i browserens DOM ved anvendelse af Web APIer.
  + Hvordan og hvorfor du anvender [Web Storage](https://developer.mozilla.org/en-US/docs/Web/API/Web_Storage_API) til at gemme data lokalt på brugerens computer. – Du vælger selv om du vil bruge localStorage eller sessionStorage.
  + Hvordan du bruger en eventListener til først at loade produkter ind i browseren når DOM’en er klar og loaded.
  + Hvordan du tilføjer data til bestillingslisten når der trykkes ’Bestil’ på et produkt.

### Krav til siden

I den udleverede mappe ligger der en produkter.json fil med en liste af produkter og tilhørende data. Der ligger også en mappe med billeder til produkter.

* Alle produkter skal kunne vises i en sektion på siden. En kaffe kan vises som et card med tilhørende information som, navn, beskrivelse og pris.
* Hvis der er trykkes ’Bestil’ på en kaffe skal navn og pris sættes ind i bestillingslisten.
* Til sidst i bestillingslisten skal der være en samlet pris for alle de tilføjede produkters pris.
* Når bestillingslisten er tom skal knappen ’Gå til betaling’ skal være inaktiv ELLER bestillingslisten skal ikke vises.

### Krav til koden

I den udleverede mappe ligger der et dokument med en liste af varer til shoppen.

* En klasse der kan oprettes klasse-instanser til hvert produkt kaldet Product. Denne klasse skal have følgende metoder:
  + Product.instances property, hvor alle produkter ligger i et object.
  + createTestData() – Skal sætte data ind i en Product.instances og kalde funktionen saveAll()
  + saveAll() – Skal gemme Product.instances til localStorage eller sessionStorage.
  + loadlAll() – Hvis der findes data i localStorage eller sessionStorage skal denne data loades ind i Product.instances, ellers skal test data oprettes for så at loades ind i Product.instances.
* En klasse til ProductCard der kan tage en eller flere produkter fra Product.instances og oprette et HTML element. Denne klasse kan indeholde følgende metoder:
  + createProductCard() – Opretter et HTML element med informationer fra et produkt.
  + clickHandler() – Bruges når der klikkes på ’Bestil.
    - Tilføjer produktet til bestillingslisten. (Skal controlleren stå for dette?)
    - Lægger prisen til den samlede pris i listen og opdaterer DOM.
    - Efterfølgende skal denne ændring gemmes i localStorage eller sessionStorage så brugeren som minimum ikke mister data ved browser refresh. (Skal controlleren stå for dette?)
  + setupUserInterface() – kalder loadAll() og sætter hvert produkt ind på siden.

# Ekstra opgaver

Du kan lave disse opgaver, hvis der er noget fra kravspecifikationen, som du ikke kunne løse, eller hvis du er færdig og vil vise overskud.

* Tilføj selv flere produkter.
* Tilføj nogle simple [CSS Transitions](https://www.w3schools.com/css/css3_transitions.asp). Fx kan et produkt fade ind i bestillingslisten når det tilføjes, eller hvert productCard kan fade ind når siden loader.
* Opret funktionalitet til at vise hvor mange produkter der er bestilt. På det udleverede screenshot er det tallet over bestillingslisten.
* Opret en ny .html side hvor kunden kan færdiggøre sin ordre, en checkout side. Det kan fx se [sådan her ud i Bootstrap.](https://getbootstrap.com/docs/4.1/examples/checkout/) Vis denne side når kunden trykker på ’Gå til betaling’.
* Re-design bestillingslisten så den viser mere information. Fx noget af beskrivelsen og et lille billede.
* Opret en funktion der henter data fra produkter.json i stedet for at de manuelt oprettes.

# Regler i eksamenslokalet

* I må gerne ringe til jeres klassekammerater og spørge om hjælp.
* Hvis i er i tvivl om opgavebeskrivelsen så ringer i til læreren.
* Opgave og opgaveløsninger må ikke tages med hjem.

# Bedømmelseskriterier og evaluering

## Faglige læringsmål (70%)

* Du er fortrolig med din foretrukne kodeeditor og mindst én browsers udviklingsværktøj.
* Du kan anvende Git og GitHub til at håndtere versionering og vidensdeling af kode.
* Du forståelse for hvordan du kan sætte data op i forskellige strukturer og hvornår de forskellige datastrukturer er relevante.
* Du kan interface med browserens DOM ved brug af Web API.
* Du kan udvikle med HTML, CSS og JavaScript.
* Du har kendskab til og kan anvende et front-end framework som Bootstrap, Materializecc eller lignende.
* Du kan implementere designs der fungerer på flere platforme.

## Personlige læringsmål (10%)

* Du har afprøvet og udviklet dine evner til at indgå i et teamarbejde.
* Du kan give og modtage konstruktiv feedback.
* Du kan reflektere over egne kvalifikationer og interesser inden for webudvikling.
* Du kan udsøge informationer og præsentere disse i relevante sammenhænge.
* Du har udviklet dine evner til at omstille dig til andre arbejdsmetoder og teknologier.
* Du er opsøgende og nysgerrig i forhold til at lære nyt.
* Du har afklaret dit engagement og motivation for faget.

## Arbejdsevnemæssige læringsmål (20%)

* Du er i stand til at levere løsninger inden for en given deadline.
* Du kan dokumentere og videregive din viden.

## Eksamen

Det fremgår af modulbeskrivelsen at:

* Ved eksamen skal du, uden forudgående forberedelse, præsentere din løsning. Desuden skal du besvare spørgsmål fra eksaminator og censor.
* Til eksamen skal du løse én mindre uforberedt opgave som bliver stillet under eksaminationen.
* Eksamen gennemføres over 45 minutter, inkl. tid til bedømmelse, og den uforberedte opgave.
* Din besvarelse bedømmes efter den almindelige karakterskala og karakteren meddeles dig umiddelbart efter eksaminators og censors votering.

## Karakterskala

|  |  |  |
| --- | --- | --- |
| Karakter | Betegnelse | Beskrivelse |
| 12 | Den fremragende præstation | Karakteren 12 gives for den fremragende præstation, der demonstrerer udtømmende opfyldelse af fagets mål, med ingen eller få uvæsentlige mangler. |
| 10 | Den fortrinlige præstation | Karakteren 10 gives for den fortrinlige præstation, der demonstrerer omfattende opfyldelse af fagets mål, med nogle mindre væsentlige mangler. |
| 7 | Den gode præstation | Karakteren 7 gives for den gode præstation, der demonstrerer opfyldelse af fagets mål, med en del mangler. |
| 4 | Den jævne præstation | Karakteren 4 gives for den jævne præstation, der demonstrerer en mindre grad af opfyldelse af fagets mål, med adskillige væsentlige mangler. |
| 02 | Den tilstrækkelige præstation | Karakteren 02 gives for den tilstrækkelige præstation, der demonstrerer den minimalt acceptable grad af opfyldelse af fagets mål. |
| 00 | Den utilstrækkelige præstation | Karakteren 00 gives for den utilstrækkelige præstation, der ikke demonstrerer en acceptabel grad af opfyldelse af fagets mål. |
| -3 | Den ringe præstation | Karakteren -3 gives for den helt uacceptable præstation. |

Tabel 1: Karakterskala for Visualisering Rutineret