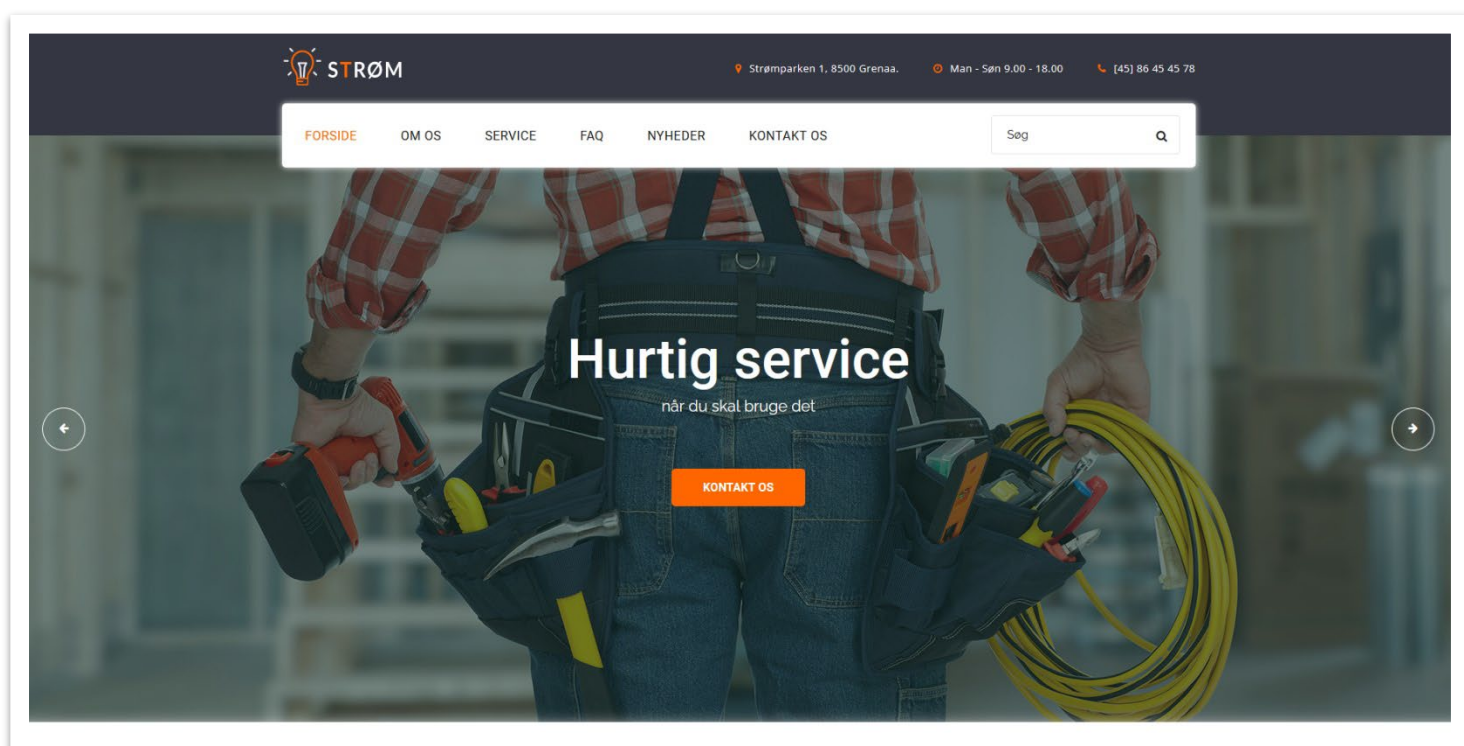


# STRØM

## WEBUDVIKLER EKSAMEN VIDEN DJURS - VID MEDIE



Hold: Webith221

Praktiks eksamen den 7. til 11. november 2022

## Vigtige generelle oplysninger om eksamen

- Afleveres projektet og rapporten 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 efter aftale med den ansvarlige underviser.  
**Har du valgt at arbejde hjemmefra, har du ikke mulighed for forlænget tid pga. systemnedbrud på eget netværk/tekniske problemer.**
- 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. Du får ikke forlænget tid pga. manglende backup.
- Alle hjælpemidler, der normalt anerkendes og benyttes i undervisningen, er tilladt.
- Det er tilladt, med tydelig kildeangivelse (i rapporten), at benytte pakker/funktioner/objekter (f.eks. JavaScript) udarbejdet af 3. part. Indholdet af disse vil dog ikke tælle med i bedømmelsen af dit projekt.
- Se i øvrigt udleverede mere detaljeret information og krav til eksamen.

# Indhold

Vigtige generelle oplysninger om eksamen .....	2
Situationsbeskrivelse .....	4
Opgavebeskrivelse .....	4
Tekniske krav .....	4
Web API .....	4
API endpoints .....	5
Billeder .....	5
Dokumentation (projekt-rapport) .....	5
Bilag .....	6
Forsiden/omslag .....	6
Generelt om rapporten .....	6
Kravspecifikation .....	7
Generelle krav .....	7
Faste elementer .....	7
Forsiden .....	8
"Om os"-siden .....	9
"Service"-siden .....	9
"FAQ"-siden .....	10
"Nyheder"-siden .....	10
"Udvalgt nyhed"-/detalje-side ("der er klikket på en nyhed") .....	11
"Kontakt os"-siden .....	11
Administration af indhold .....	11
Tilvalgsopgaver .....	12
Prioritetsliste .....	15
Aflevering .....	15
Du skal aflevere følgende .....	15
Sted for aflevering .....	16
VIGTIGT: Din aflevering skal godkendes! .....	16

## 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/oplægget og sitets generelle struktur er allerede fastlagt og godkendt af kunden.

**Din opgave bliver at kode sitets funktionalitet** – herunder sitets front-end samt et udvalgt eksempel på administrations-delen (kundens mulighed for at oprette/rette/slette indhold).

## Opgavebeskrivelse

**STRØM A/S** 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 relevante informationer. STRØM A/S ønsker, at den nye side præsenterer deres produkter, services og budskaber på en professionel måde.

## Tekniske krav

Udkast til designet udleveres sammen med opgaven. **Som udgangspunkt skal du følge designet i layoutet/oplægget** (se også den udleverede demo-video), men kunden er modtagelig for ændringer i designet, hvis disse kan argumenteres til kundens fordel – især i den responsive version, men også gerne i brugervenlighed/tilgængelighed og/eller funktionaliteter.

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.

De væsentlige dele af sitets data/indhold (tekst, billeder mv.) **skal hentes/administreres fra det udleverede API** – som beskrevet herunder.

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

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

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

## Web API

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

API'et kører på MongoDB, ExpressJS og NodeJS, 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 **read-me.txt** – her er generel information om brug af backend/indhold fra backend, login, brugere der kan logge ind mv.

Når databasen er oprettet og API'et er startet, kan du benytte de indbyggede GET, POST, PUT og DELETE mv. metoder, som det fremgår i den udleverede **Postman-/Insomnia**-collection.

Du må – hvis det er relevant for opgaven, kravene og/eller kundens behov - udvide/ændre i det eksisterende API. **Ændrer du i API'et/backenden skal du notere dette i din dokumentation (rapport): HVAD du har ændret/tilføjet – og hvorfor. OG hvis det ændrede API er tilgængeligt online, skal du aflevere et link til denne i din rapport.**

## API endpoints

Når backenden er installeret og startet op, kan du se en oversigt over samtlige endpoints samt sti til billeder på adressen: <http://localhost:5333/>

Samtlige endpoints – samt tilhørende request-metoder (**GET, POST, PUT/PATCH, DELETE**) kan tilgås efter opsætning af den udleverede collection i Postman og/eller Insomnia.

Bemærk, at API'et kører på **port 5333**.

## Billeder

Billeder til produkter, slider mv. hentes fra/uploads til backendens "public"-mappe – du kan finde stierne til billederne her: <http://localhost:5333/> - de er også beskrevet i API'ets read-me.txt

## Dokumentation (projekt-rapport)

Du skal udarbejde dokumentation/projektrapport sideløbende med opgaveløsningen. Gem ikke denne del til sidst – og især tidsplanen/din projektplanlægning bør være noget af det første, du bruger tid på og følger op på løbende i projektperioden.

**Dokumentationen skal som minimum indeholde følgende:**

1. Vurdering af din **egen indsats** og gennemførelse af opgaveforløbet
2. Din **tidsplan**, hvor det fremgår, hvordan du har planlagt og udført arbejdet (estimeret og faktisk tid) - det kan fx være foto af tavle, link til Trello/Milanote, screendump eller vedhæftet Gant Chart osv. Evt. vedhæftet som bilag. Brug af det, du har lært om projektkompleksitet/Scrum.
3. Beskrivelse af og argumentation for de **valg du evt. har truffet** under løsningen af opgaven - herunder hvad du eventuelt har ændret/tilføjet i backend/API.
4. Beskriv hvilke af **tilvalgsopgaverne**, du har løst.

5. 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) – herunder de pakker, du har installeret ud over de nødvendige.
6. Angivelse af de **url-adresser, brugernavne og passwords**, der er nødvendige for at lærer og censor kan se/køre din løsning.
7. Evt. **link til dit færdige** projekt på GitHub (indsættes på forsiden).  
Vigtigt: Hold kontoen privat indtil alle har afleveret (mandag kl. 9).
8. En beskrivelse af eventuelle **særlige punkter til bedømmelse** – fx hvad du har udvidet med af ekstra funktioner, designelementer mv. som er udover opgavens krav.
9. 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

- Evt. uddrag af **særlige kode-eksempler**, hvis det er relevant.
- Din **tidsplan** – hvis den ikke er indsat i hoveddelen af rapporten.

## Forsiden/omslag

På forsiden skal fremgå:

- **Opgavens navn (STRØM), dit navn, holdnummer, skolens navn og afleveringsdato og tidspunkt**
- Samt følgende tekst:  
*Jeg bekræfter hermed, at jeg selvstændigt og uden hjælp fra andre har udviklet det afleverede eksamensprojekt.*
- Din underskrift og dato

## Generelt om rapporten

- Omfanget af rapporten skal være på **maksimum 5 A4-sider** eksklusive bilag – hvis der indgår mange billeder, må rapporten fylde tilsvarende mere.
- Projektrapporten **afleveres som en pdf-fil**.
- **Projektrapporten skal placeres i projektets rodmappe** ved aflevering.
- Rapportens navn SKAL være **Rapport\_dit-fulde-navn-her.pdf**.

# Kravspecifikation

Med opgaven er der udleveret et oplæg og et antal billed-filer (suppleret med et videoklip) som viser/demonstrerer layoutet.

Der kan være dele i opgavebeskrivelsen, som IKKE indgår i design-oplægget, hvor du som udvikler har frie hænder til opsætning af et forslag, som dog stadig skal følge den generelle stil som design-oplægget lægger op til. Der kan omvendt også indgå elementer på designoplægget, som IKKE er beskrevet i opgaven – disse skal som udgangspunkt indgå i opgaveløsningen, men spørg, hvis du er i tvivl.

## Generelle krav

- Sitet bygges i mål og opsætning der **modsvare layoutet i det udleverede oplæg**.
- Sitet skal være **responsivt** og optimeret til alle relevante devices (mobile, tablet, desktop mv.). Du bestemmer selv, hvordan løsningen reagerer i forskellige device-størrelser/breakpoints, men du skal kunne argumentere for din valg.
- Sitet skal have et **favicon** svarende til det udleverede logo – tilpasset/-skåret så det passer til favicon-størrelse.
- Hover-animationer og andre animationer og effekter, som vist på den udleverede demo-video, skal tilføjes til sidens elementer.

## Faste elementer

Følgende skal indgå på alle sider:

### Header

---

- Med logo, adresse, åbningstider, telefonnummer.
- Indhold – undtaget logo - hentes fra API (endpoint "[contactinformation](#)").

### Navigation/navbar

---

- Med links til undersider mv.
- Med søgefelt (selve søgefunktionen er en tilvalgsopgave).

### Footer

---

- Med logo, garantiordning-tekst, "Link" (navigation), copyright osv. som vist i oplægget. Dette er statisk indhold – *ikke* fra API.
- "**Kontakt os**" – adresse mv. hentes fra API (endpoint "[contactinformation](#)").

- **Nyhedsbrev-tilmelding** med tilmelding/POST til API (endpoint "[newssubscription](#)") – husk tilbagemelding til bruger efter tilmelding. Håndter gerne, at den samme email-adresse ikke kan tilmelde sig flere gange (API'et svarer med en fejl).
- **Social media ("some") ikoner** hentes fra API (endpoint "[contactinformation](#)").

## Desuden

---

- Fixed button "**scroll back to top**" – altså en fast positioneret knap, som linker til toppen af siden – se det udleverede oplæg (smooth scroll – se video).
- Brødkrummesti/**breadcrumbs** på alle undersider (fx [Forside](#) > [Services](#)).

OBS! Elementerne herover er faste på selve websitet, men kan/skal tilpasses eller udelades på ADMIN-siderne.

## Forsiden

Husk at sammenholde elementerne herunder med det udleverede oplæg og video-klip.

Bemærk, at der er flere hover-, transition- og animationseffekter, som kun fremgår af videoen.

## Slider

---

- Med **navigation** (frem/tilbage) og knap-link til Kontakt-siden.
- Slider-billeder hentes fra API (endpoint "[slider](#)").

## "Lidt om STRØM"

---

- Med knap-link til "Om os"-siden.
- Tekst og titel hentes fra API (endpoint "[about](#)").

## "Skal du bruge hjælp fra STRØM?"

---

- Med knap-link til Kontakt-siden.

## "Vores services"

---

- Afsnittets titel "Vores services", foto og lille tekst under er statisk – *ikke* fra API.
- Services (titel, teasertekst, ikon mv.) hentes fra API (endpoint "[service](#)").  
Ikonerne er "[flaticons](#)" - en font/skrifttype og er blandt det udleverede materiale.

## "Book service nu"

---

- Med POST til API (endpoint "[booking](#)") – husk tilbagemelding til bruger efter booking.



## "Vores kunder siger"

---

- Afsnittets titel "Vores kunder siger" og lille tekst under er statisk – *ikke* fra API.
- Slider/carousel med 4 **tilfældige** kundeudtalelser/testimonials.
- Testimonials (tekster, navne, profilfoto mv.) hentes fra API (endpoint "[testimonial](#)").

## "Vores team"

---

- Afsnittets titel "Vores team" og den lille tekst under er statisk – *ikke* fra API.
- Team (navn, stilling, image mv.) hentes fra API (endpoint "[team](#)").
- Socialmedia-icons er statiske med "døde links" - hentes ikke fra API.

## "Sidste nyt" med 3 nyeste nyheder

---

- Afsnittets titel "Sidste nyt" og lille tekst under er statisk – *ikke* fra API.
- Under de 3 **nyeste** nyheder tilføjes et knap-link til "Nyheder"-siden.
- Hver nyhed vises med:
  - Dato, foto, titel og et **udsnit af teksten** (fx ca. 100 tegn – herefter "..."), som vist på oplæg/layout.
  - **Link til tilhørende detaljeside** (se herunder). Under nyhederne et knap-link til "Nyheder"-siden
- De 3 nyeste nyheder hentes fra API (endpoint "[news](#)").

## "Om os"-siden

- Hovedafsnittets (øverst) titel, tekster og foto hentes fra API (endpoint "[about](#)").
- Knap-link til "Kontakt os"-siden.
- "Vores kunder siger" – som på forsiden (læs herover).
- "Vores team" – som på forsiden (læs herover).

## "Service"-siden

### Venstre kolonne

---

- Services titler (fx Air Condition, Sikkerhed, Strøm osv.), hvor hver service er vist med en pil og fungerer som link/er **klikbar**:

**Ved klik på en titel (fx Strøm) udskiftes indhold i højre kolonne**, så indholdet viser den service (fx Strøm), der er klikket på.

- Titler hentes fra API (endpoint "[service](#)").

### Højre kolonne

---

- Visning af den service, som er **aktiv**/valgt i venstre kolonne – med alt indhold (titel, tekster, foto mv.).
- Indhold hentes fra API (endpoint "[service](#)").

## "FAQ"-siden

- Spørgsmål-svar vises som accordion (folde-ind-ud-tekster – se video).
- Indhold hentes fra API (endpoint "[faq](#)").

## "Nyheder"-siden

### Venstre kolonne

---

De 4 **nyeste** nyheder – hver nyhed vist med:

- Dato, foto, titel og et **udsnit af teksten** (fx ca. 100 tegn – herefter "...") og antal kommentarer som vist på layout/oplæg.
- Hver nyhed **med link til tilhørende detaljeside** (se herunder).
- OBS! Pagination under nyhederne er en tilvalgsopgave.
- Nyhederne hentes fra API (endpoint "[news](#)").

### Højre kolonne – "Arkiv"

---

De 4 "**næst-nyeste**" nyheder (efter nyheder vist i venstre kolonne) - hver nyhed vist med:

- Titel, dato og foto som vist på layout/oplæg.
- Hver nyhed **med link til tilhørende detaljeside** (se herunder). Under nyhederne et knap-link til "Nyheder"-siden.
- Nyhedernes indhold hentes fra API (endpoint "[news](#)").

## "Udvalgt nyhed"-/detalje-side ("der er klikket på en nyhed")

### Venstre kolonne

---

Nyheden vises som på det udleverede oplæg med:

- Titel, tekst, foto osv. med antal kommentarer som vises med ikon + antal over titel.
- Nyhedens kommentarer vises under nyheds-teksten.
- Formular til kommentar vises under kommentarerne.  
OBS! Oprettelse/POST af en kommentar er en tilvalgsopgave.
- Nyheden hentes fra API (endpoint "[news](#)").

### Højre kolonne – "Arkiv" (tilfældige nyheder)

---

4 **tilfældige** nyheder - hver nyhed vist på samme måde som i "Arkiv"/højre kolonne på "Nyheder"-siden.

- Nyhederne hentes fra API (endpoint "[news](#)")

## "Kontakt os"-siden

- Adresse, telefon, email som vist på det udleverede oplæg.
- Kontaktinfo (adresse mv.) hente fra API (endpoint "[contactinfo](#)").
- Kontaktformular med POST til API (endpoint "[contact](#)") – husk tilbagemelding til bruger efter "kontakt".
- Kortudsnit – brug et Leaflet map, Google map eller noget tilsvarende efter eget valg – men ikke et statisk screendump/image.

## Administration af indhold

Du bestemmer selv, hvordan administrations-delen skal designes – men der skal være sammenhæng til det udleverede design.

Der skal minimum være administration på følgende dele (øvrige administration er tilvalgsopgaver):

### Nyheder

---

- **Opret, ret og slet** (endpoint "[news](#)")
- Med brug af RTE/editor på produktets tekst/content

## Om os/About

---

- **Ret** (endpoint "about")
- Med brug af RTE/editor på about-teksterne

## Booking (booking-forespørgsler sendt fra brugerne/kunderne)

---

- **Slet** en booking (endpoint "booking")
- **Marker** en booking som **accepteret/godkendt**
- **Tilføj en note** (notat/antegning/kort besked) til en booking

# Tilvalgsopgaver

**Du skal vælge minimum 2 af følgende tilvalgsopgaver.** Du må dog gerne vælge flere, såfremt du har styr på prioriteringen, og **hvis øvrige krav i kravspecifikationen er opfyldt:**

### 1. Søgning

---

Brugeren kan foretage en søgning i søgefeltet i navbaren. Du har frie hænder til at designe søgeresultat-siden – som popup, modal, udvidelse af en eksisterende side, en ny side/page el.lign. Data/søgning håndteres ved brug af API'et (endpoint "search").

Bemærk at søgeresultatet returnerer indhold fra både "news" og "services" (test det i Postman).

### 2. Pagination af nyheder ("Nyheder"-siden)

---

Nyhederne vises med pagination (se det udleverede oplæg), så der kun vises fx 4 nyheder på "Nyheder"-siden men med mulighed for at "bladre"/linke til side 2, 3, 4 osv.

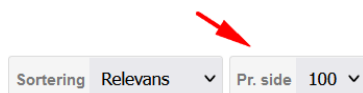
Nyeste nyheder på side 1 – ældste nyheder på sidste side.

### 3. Pagination af nyheder – ekstra (udvidelse til pagination-opgaven herover)

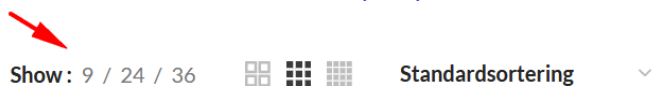
---

Tilføj en **dropdown el.lign.** så brugeren kan vælge hvor mange nyheder, der skal vises på hver side – stadig med mulighed for at "bladre" til side 2, 3, 4 osv.

Tjek fx **dropdown** "Pr. side" her: <https://www.skiftselv.dk/inspiration-gaveartikler/#page=1>



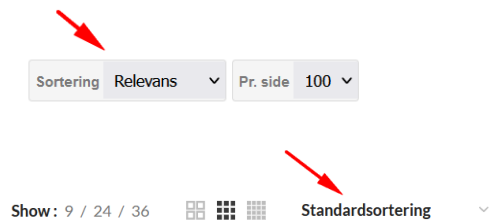
Eller versionen "Show" her: <https://perchs.dk/vare-kategori/sort-te>



## 4. Sortering af nyheder

---

Brugeren kan sortere nyhederne efter fx dato (fx "nyeste" og "ældste"), titel (alfabetisk), antal kommentarer el.lign. Se de to links som i opgaven herover:



## 5. Frameld nyhedsbrev

---

Brugeren kan - ved at sende sin email-adresse - framelde sig nyhedsbrevet. Du har frie hænder til at designe hvor og hvordan, men argumenter gerne for placering mv.

API'et har en DELETE-metode til framelding af nyhedsbrev (endpoint "[newssubscription](#)").

Overvej, hvordan frameldingen skal håndteres, hvis email-adressen ikke findes – og brugeren dermed ikke kan frameldes (fordi der ikke er noget at framelde).

## 6. Brugere kan kommentere på nyhederne

---

Kommentar-formularen på "Udvalgt nyhed"/detalje-siden til nyheder laver et **POST af brugerens kommentar** til en udvalgt nyhed til API'et (endpoint "[news](#)").

## 7. Administration af brugeres kommentarer

---

Administratoren har adgang til en **oversigt** over alle bruger-kommentarer til en nyhed og kan **markere en kommentar "publish"-status** til *true/false*. Kun hvis en kommentar er "publish=*true*", skal den publiceres/vises på nyhedens side.

Data håndteres ved brug af API'et (endpoint "[news](#)" og "[news/comments](#)").

## 8. Administration af beskeder fra kontaktformularen

---

Administratoren har adgang til en **oversigt** over alle brugernes kontaktbeskeder (fra kontaktformularen) med mulighed for at markere/gemme beskeder som **læst** eller **ulæst** ("read" er *true* eller *false*).

Data håndteres ved brug af API'et (endpoint "[contact](#)")

## 10. Administration af "services" og "contactinformation"

---

Administratoren kan administrere "**services**" (endpoint "[services](#)") og "**contactinformation**" (endpoint "[contactinformation](#)"), dvs. rette i data (tekster, billeder mv.) via administrations-siden.

## 11. Login-beskyttelse af kommentarer

---

Kun brugere, der er logget ind kan oprette kommentarer til nyheder. Dette kan demonstreres/simuleres UDEN brug af API.

Det er muligt koble API'ets endpoint `"user"` på. Bemærk at API'ets endpoint `"user"` rummer to typer `"brugere"/roller`: `admin` (`admin: true`) og `"ikke-admin"` (`admin: false`) – du kan evt. bruge denne oplysning, til håndtering men det er ikke nødvendigt.

## 12. Login

---

Tilslut API'ets login-beskyttelse af administrations-delen, så kun brugere/users med godkendt e-mail og password kan få adgang (der medsendes en session-cookie). Brug API'ets endpoints: `"user"` og `"login"`.

Følgende brugere er oprettet i systemet:

- **el@stroem.dk** med password: **admin123** (rolle: `admin: true`)
- **ab@stroem.dk** med password: **user123** (rolle: `admin: false`)
- **cd@stroem.dk** med password: **user456** (rolle: `admin: false`)
- **ef@stroem.dk** med password: **user789** (rolle: `admin: false`)

Du kan evt. selv oprette flere brugere med brug af Postman/Insomnia.

Husk at aktivere login-delen i `server.js` (dvs. fjerne udkommenteringen), som kontrollerer login-sessions og sikrer, at der kun er adgang ved korrekt login.

## 13. Administration af `"users"/brugere`

---

Giv adgang til, at en administrator (`admin: true`) der er logget ind, skal kunne oprette nye brugere samt rette andre brugeres oplysninger, slette brugerkonti mv.

Alle brugere/administratorer (uanset om `admin: true` eller `false`) skal (hvis logget ind) kunne administrere egen profil – dvs. skifte password, email - samt slette egen `"konto"` (endpoint `'user'`), når brugeren er logget ind.

Udvid evt. selv med flere muligheder...

## Ekstra

---

**Har du løst minimum 2 af tilvalgsopgaverne og opfyldt opgavens øvrige krav, kan du udvide med yderligere funktioner efter eget valg og påfund – men de skal være relevante, og du skal kunne argumentere for dine tilføjelser/ændringer.**

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

## Prioritetsliste

Opgaverne skal prioriteres efter denne liste (**især punkt 1-3 er vigtige for at opgaven er bestået**). Detaljer om opgaverne findes i afsnit om Tekniske krav, Kravspecifikation mv.:

- Løsningen skal være **responsiv** som beskrevet i kravspecifikation mv.
- **Generel opsætning af design** - sitet skal fremstå færdigt i html og css og **matche det udleverede oplæg** – og alle ekstra elementer skal tilsvarende ligne oplægget.
- **Tekst, billeder mv.** er udtræk fra API'et som beskrevet i kravspecifikationen – dvs. data skal, som beskrevet, håndteres med **request-metoder** (GET POST PUT/PATCH DELETE).
- **Forsiden** og slideren er vigtig for kunden, da en stor del af deres trafik starter fra forsiden.
- **Kontaktformular, nyhedsbrevstilmelding og mulighed for booking** med POST-funktionalitet, validering mv.
- **Visning af nyhederne** på såvel forside som nyhedsside
- **Øvrige sider med indhold**
- **Administration (opret, ret, slet) af data** som beskrevet i afsnittet "Administration af indhold".
- **Tilvalgsopgaver** – minimum 2 af opgaverne, som beskrevet i afsnittet "Tilvalgsopgaver"
- **Ekstraopgaver, øvrige opgaver, udvidelse af API osv.**

**HUSK også projektdokumentationen/rapporten med visning af, hvordan du har struktureret og planlagt projektet.**

## Aflevering

Sidste frist for aflevering er: **Fredag den 11. november kl. 15.45** (eller efter aftale) – der tillægges tid til installering, opstart og evt. tilretning af API tillægges afleveringstidspunktet, så aflevering sandsynligvis er en time eller mere senere end angivet.

VIGTIGT: Din løsning skal præsenteres til den mundtlige eksamen på en lokal eller webbaseret serverløsning efter eget valg. Det er derfor **vigtigt, at du ikke redigerer i din løsning efter afleveringsfristen er udløbet**. Ønsker du at arbejde videre med projektet EFTER aflevering, så lav en kopi til formålet.

## Du skal aflevere følgende

- **Backend/API skal kun afleveres**, hvis du har foretaget ændringer i det udleverede
- **Rapport** i pdf-format: "*Rapport\_dit-navn-her.pdf*" - placeres i roden af webprojektet
- Dit **webprojekt** som ZIP-fil - og UDEN mappen node\_modules!

## Sted for aflevering

Din løsning skal afleveres i VIDOnline – i opgaven:

Webith221 Eksamen (praktisk og mundtlig) / Eksamensprojekt - og AFLEVERING den 11. november 2022

– som findes her: <https://online.videndjurs.dk/mod/assign/view.php?id=392228>

## VIGTIGT: Din aflevering skal godkendes!

Når du har afleveret, skal du kontakte underviseren, som tjekker, at alt er afleveret – og at det kan downloades, unzippes osv.

**Du har IKKE afleveret før det er godkendt af underviseren**, så afvent godkendelsen, før du evt. forlader skolen, lukker Teams ned osv.

God arbejdslyst!