



Eksamen

Vigtige oplysninger om eksamen

- Afleveres projektet og rapporten ikke til tiden, er prøven ikke bestået.
- Ved systemnedbrud eller andre tekniske problemer på skolens område kan afleveringsfristen udsættes, 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, hvis der opstår 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 arbejde nævneværdigt. Du får ikke forlænget tid pga. manglende backup.
- Alle hjælpemidler, der normalt anerkendes og benyttes i undervisningen, er tilladt. Bemærk dog at det **IKKE er tilladt at bruge AI til eksamen.**
- 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.

Brug af AI til eksamen

- I henhold til gældende regler for erhvervsuddannelserne er **brug af kunstig intelligens (AI), såsom ChatGPT og lignende AI-baserede tekstgeneratorer, ikke tilladt** under eksamen eller en eventuel forberedelse. Dette skyldes, at prøvebesvarelsen skal være eksaminandens egen og selvstændige besvarelse. Anvendelse af ulovlige hjælpemidler vil blive betragtet som eksamenssnyd og resultere i bortvisning fra prøven.
- Derfor **er følgende AI-værktøjer ikke tilladt**: ChatGPT, Gemini, Claude og andre AI-tekstgeneratorer.
- Derimod er **GitHub Copilot og visse AI-assisterede funktioner i Adobe-pakken tilladt**, da de fungerer som kodeassistance og ikke som en automatisk kodegenerator. **Enhver kode, der er genereret med Copilot, skal dog tydeligt kildehenvises** i dokumentationen på samme måde som øvrige eksterne ressourcer. Manglende kildehenvisning kan blive betragtet som plagiat.
- **Det er dit ansvar at sikre, at du ikke bruger ulovlige hjælpemidler. Ved tvivl – spørg underviseren.**
- Se mere på: [Børne- og Undervisningsministeriet: Brug af AI-baserede værktøjer til prøverne og under forberedelse](#)

Kravspecifikation og generelle krav

- Som udgangspunkt skal du **følge designet i opsætningen**.
- Kunden accepterer ændringer i designet, hvis du kan argumentere for, at det er til kundens/brugerens fordel.
- 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 skal følge stilen.
- Der kan omvendt også være elementer på designoplægget, som IKKE er beskrevet i opgaven – disse skal som udgangspunkt indgå i opgaveløsningen. Spørg, hvis du er i tvivl.
- Sitet skal bygges med **HTML, CSS og JavaScript med brug af et frontend-framework fx React, Next** eller tilsvarende - og gerne med brug af Sass/Less el. lign.
- Sitet skal opfylde de gængse standarder inden for **semantisk, korrekt kode** gerne med tanke for **accessibility**/tilgængelighed og **søgemaskineoptimering/SEO**.
- Sitet skal tage udgangspunkt i mobile-first princippet. Sørg for, at siden er optimeret til skærmstørrelser fra **320px op til tablet**. (Responsivitet med breakpoint > 1024 er tilvalgsopgave)
- Sitet bør **indeholde fejlhåndtering på alle relevante fejl**, så brugeren får en fejlmeddelelse, der tager hensyn til sitets indhold – fx forkert URL ('siden findes ikke'), manglende/forsinkede data fra API'et, brugerens fejlinput mv.
- Der skal være **relevant validering på alle formularer/inputelementer**.
- Sitet skal have fokus på brugeroplevelse (UX). Overvej derfor følgende:
 - **Brugervenlighed**: Er knapper, formularer og interaktioner intuitive?
 - **Feedback til brugeren**: Modtager brugeren tydelige fejl- og succesmeddelelser ved interaktioner (f.eks. formularindsendelse)?
 - **Ydeevne og responsivitet**: Er websitet hurtigt og tilpasser det sig problemfrit til forskellige enheder?
- Væsentlige dele af sitets data/indhold (tekst, billeder mv.) skal **hentes/administreres fra det udleverede API**

Dokumentation (rapport)

Tidsplan, estimat og todo-liste

- **Tidsplanen/projektplanlægningen bør være noget af det første**, du bruger tid på, hvor du nedbryder projektet i mindre opgaver/"tasks"/issues og laver et tids-estimat på hver opgave.
- Husk at lave en **daglig opfølgning** på dine fremskridt og anvendt tid/reel forbrugt tid, så du kan sammenligne med dit estimat. Det viser, at du kan planlægge OG det giver dig en fornemmelse af, om din tidsplan holder.

Generelt om projektrapporten/dokumentationen



Dokumentationen skal være på **højst 5 A4-sider** eksklusive bilag og billeder/grafik.



Dokumentationen skal afleveres som en **md- eller pdf-fil** placeret i **projektets rodmappe**.



Dokumentationens filnavn: **rapport_dit-navn-her.md** eller **.pdf**

Projektrapportens indhold



1. Vurdering af egen indsats - Besvar bl.a. følgende spørgsmål:

- **Hvad gik godt?** Hvilke dele af projektet var du mest tilfreds med?
- **Hvad var udfordrende?** Hvilke tekniske problemer stødte du på, og hvordan løste du dem?
- **Hvad ville du gøre anderledes?** Hvis du havde mere tid eller kunne starte forfra, hvad ville du optimere eller ændre?
- **Hvordan har du udviklet dig?** Hvilke nye færdigheder eller metoder har du lært undervejs?

2. Din tidsplan med kommentarer

- med visning af planlagte/udførte opgaver/tasks (estimeret og faktisk tid). Fx suppleret med foto af todo-tavle, link til Trello/Github, screendump eller Gant Chart osv. Evt. vedhæftet som bilag.

3. Din tech-stack

- fx om du har brugt React, Vue, Svelte mv. Om du har benyttet dig af Bootstrap, TailwindCSS, styled components, Sass/scss mv.

4. Beskrivelse af og argumentation for valg du evt. har truffet under løsningen af opgaven.

- Dette gælder også for evt. ændringer i API'et/backend'en (husk at det skal godkendes): HVAD du har ændret/tilføjet – og HVORFOR.
- Hvis det ændrede API er tilgængeligt online, skal du aflevere et link til dette i din rapport.

5. Noter hvilke tilvalgsopgaver, du har løst.

6. Liste over kodeelementer udarbejdet af 3. part i opgaveløsningen

- fx Bootstrap/TailwindCSS, Sass/Less, færdige scripts/plugins (fx lightbox, gallerier, scroll- funktionalitet mv. du ikke selv har udviklet), herunder pakker, du har installeret ud over de nødvendige.

7. Url-adresser, brugernavne og passwords nødvendige for at kunne teste din løsning.

8. Evt. link til dit projekt på GitHub.

9. Vigtigt: Hold kontoen privat indtil efter aflevering er afsluttet.

10. Beskriv eventuelle særlige punkter til bedømmelse.

Aflevering

- **Rapport** i pdf- eller md-format: "*Rapport_dit-navn-her.pdf*" - placeres i roden af webprojektet
- Dit **webprojekt** som enten github-repository eller ZIP-fil - og UDEN mappen node_modules!
- **Backend/API skal IKKE afleveres**, medmindre du (efter aftale) har ændret i det udleverede API. Husk at argumentere for ændringerne i din rapport.
- Sidste frist for aflevering er **fredag den 4. april kl. 15** (eller efter aftale).
- VIGTIGT: Det er **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:**
 - Sted for aflevering
 - Løsningen skal afleveres via Teams hvor opgaven også er udleveret.
 - VIGTIGT: Din aflevering skal godkendes!
 - Når du har afleveret, skal du informere din underviser, som herefter tjekker, at alt er afleveret og at det kan downloades, unzippes osv. Afvent godkendelsen inden du "holder fri".

Bedømmelseskriterier

Projektet

- Den tekniske løsning
- Dokumentation og refleksion
- *Skal beståes med minimum 02*

Den mundtlige præsentation

- Præsentation af projektet
- Refleksion og faglig samtale
- *Skal beståes med minimum 02*

Karakterskala

Karakterer og vurderingskriterier

Tjekliste

Tjekliste til den afsluttende eksamen

Før eksamen

- På selve eksamensdagen ankommer du til skolen i god tid, før eksaminator og censor har skuetid på din projektløsning. Projektet afvikles fra din egen maskine, der bliver forbundet til en større skærm via HDMI (Tjek hjemmefra at forbindelsen virker).
- Hvis ikke du medbringer strømforsyning, skal du sørge for at maskinen er fuldt opladet og at den ikke låser med kode efter få minutter.
- Fjern alt fra skrivebordet som er uvæsentlig for eksamen, og klargør programmer og browser med projektet tændt.
- Du bliver inviteret ind i eksamenslokalet kortvarigt for at opsætte din maskine, og skal derefter forlade lokalet under skuetiden.
- Ved skuetids ophør (ca. 30 minutter) bliver du hentet ind i lokalet igen, og skal begynde din præsentation af projektet.

Under eksamen

- Du har ca. 25 minutters eksamenstid, hvorefter du skal forlade lokalet, mens eksaminator og censor voterer.
- De første 5 minutter skal du holde et oplæg om et valgfrit emne, der relaterer til din opgave.
- Under eksaminationen skal du præsentere din opgaveløsning. Derudover skal du kunne redegøre for enkeltelementer af løsningen - Dette kan være spørgsmål omkring koder, valg af metodikker, navigationselementer, databaselayout eller lignende.
- Udover projektopgaven kan øvrige emner, der relaterer sig til uddannelsens fagområde indgå i eksaminationen.

Råd til en god eksamen

Før du går i gang

- Hils pænt på **både underviser og censor**.
- Smil, og træk vejret roligt – du har styr på det, du selv har lavet!

Når du starter præsentationen

- Indled med en kort præsentation af, hvordan du vil gribe det an:

“Jeg vil starte med et 5 minutters oplæg om mine custom hooks i React. Jeg foretrækker, at I venter med spørgsmål til efter mit oplæg. Derefter vil jeg fremhæve rating-funktionen, som jeg har beskrevet i rapportens afsnit om bedømmelseskriterier. Til sidst vil jeg vise, hvordan jeg har bygget min backoffice.”

- **Vis ikke hele websiden fra ende til anden** – censor og underviser har set den under skuetiden.
- Fokuser i stedet på **koden, funktionerne og de faglige valg**, du har truffet.

Når du bliver spurggt om noget

- Svar roligt og fagligt.
- Hvis du ikke kan huske svaret med det samme, så sig fx:

“Det vil jeg gerne lige vende tilbage til. Først vil jeg lige forklare...”

- Du må gerne **tænke højt** – vis at du forstår, hvordan tingene hænger sammen.
- Brug fagtermer som: *komponenter, props, hooks, state, rendering, datakald, routing* osv.

Brug gerne dine egne noter

- Du må gerne have **kortfattede noter eller stikord** med – bare ikke læse op.
- Lav evt. en oversigt: *“Det vil jeg tale om: 1) Hooks 2) Rating 3) Backoffice”*

Gode pointer at få med

- Hvorfor du valgte React
- Hvad du selv synes er vellykket
- Hvad der var svært og hvordan du løste det
- Hvordan du har testet dit projekt
- Hvad du ville forbedre, hvis du havde mere tid

Vær dig selv – og vis, hvad du har lært!

- Du skal **ikke kunne det hele udenad** – men du skal kunne **forklare det, du har lavet**
- Det handler ikke om at være perfekt – men om at **vise forståelse, struktur og refleksion**

Efter eksamen

- Du vil du blive kaldt ind i lokalet, og mundtligt få udleveret din karakter samt en kort forklaring af karakteren.
- I tilfælde af utilfredshed med karakteren, skal denne ikke diskuteres ved overdragelsen af karakteren, men kan klargøres i en skriftlig klage over karakteren, formen ved eksamen eller karakterens afgivelse.