PRO101 Webprosjekt

Eksamensoppgave

June 5, 2019

PROGRAMVARE & PROTOTYPE BESKRIVELSE

Utviklet av

Jonas Say, Fredrik Holanger, Fredrik Jansen, Sindre Fromeide og Benjamin Opsal

Gruppe 31 har i forbindelse med eksamen i faget PRO101 utviklet en web-applikasjon (prototype) i henhold til eksamenskriterier. Prosjektet finnes i sin helhet slik det ble levert på: <https://github.com/FrikoPro/Gruppe31>. Alle endringer som er utført etter 07/06/2019 klokken 15:00 inngår ikke i den leverte eksamensoppgaven.

Prototypen er et lettvekts prosjektstyringsverktøy. Applikasjonen simulerer en klassisk Kanban tavle der arbeidsoppgaver deles opp og plasseres inn i kolonner. Kolonnene illustrerer visuelt hvor i prosessen en arbeidsoppgave til en hver tid befinner seg. Applikasjonen skal fungere som et prosjekteringsverktøy som gir et helhetlig bilde av fremdrift og arbeidsoppgaver.

Bruk:

Applikasjonen er tiltenkt mindre prosjekter der bruker ikke har tid eller ønsker å sette seg inn i et nytt prosjekstyringksverktøy. Brukergrensesnitt, ikoner og plassering av objekter følger generelle normer for god GUI-design. Applikasjonen er generell, ryddig og kan settes opp på under 5 minutter.

Funksjonalitet:

En bruker logger inn i applikasjonen ved å taste inn brukernavn og passord på innloggingssiden. Etter godkjent innlogging føres brukeren til applikasjonens dashbord. I tillegg til å fungere som en oversikt kan nye prosjekter opprettes, slettes eller legges i historie arkivet. Applikasjonens grensesnitt er delt inn i tre hovedgrupper. På venstre side finner vi en meny med ulik funksjonalitet. Høyre side er forbeholdt en aktivitets-logg som viser hendelser i kronologisk rekkefølge. Mellom disse finner vi et hovedområdet som er tiltenkt de ulike prosjektene som opprettes. Brukeren kan navigere inn i de ulike prosjektene via en knapp på selve prosjekt-kortet etter at det er opprettet.

Navigerer vi inn i et prosjekt vil vi se at grensesnittet med menyen til venstre, aktivitet til høyre og hovedområdet i sentrum går igjen. Området i sentrum inneholder nå kolonner som brukeren kan opprette, endre eller slette etter eget ønske. Nye oppgaver kan genereres og flyttes mellom de ulike kolonnene etter hvert som oppgavene blir fullført. I tillegg finnes en fremgangs-bar (progress-bar) som viser hvor langt et prosjekt er på vei fra 0% til 100%. Ulike oppgaver kan delegeres til medlemmer i tillegg til dato og tid. Det er også lagt inn en “dark-mode” funksjon som gjør siden mørk.

Teknologi:

Applikasjonen tar i bruk HTML (html5), CSS og JavaScript. Det er ikke tatt i bruk rammeverk utenfor det som regnes som standard i de respektive teknologiene. Det er fire HTML filer som sammen danner grunnlaget for applikasjonen. Disse har hvert sin CSS fil i tillegg til en felles CSS-fil som inneholder stiler som går igjen.

HTML: Html benytter html5 syntax og head med en felles gjennomgående struktur: html -> head -> body -> main (content) -> footer

CSS: Applikasjonen tar i bruk CSS-Grid og er skalerbar ned mot, men ikke helt ned til mobile enheter. Css-grid er i brukt for å plassere og gi egenskaper til større områder av grensesnittet. Css-flex er brukt på mindre områder. For å enkelt skape et gjennomgående design benytter applikasjonen en felles CSS-fil “common.css” i tillegg til en egen dedikert CSS-fil. Denne “common.css” inneholder felles stiler og gjør det enkelt å endre felles trekk i applikasjonen. Alle klassene i denne filen starter med bokstavene “cm” for common. For eksempel vil “cm-button” gi et html element en felles stil og animasjon.

JavaScript: Applikasjonen benytter JavaScript til å utføre en mengde oppgaver. Mange av HTML elementene genereres dynamisk ved hjelp av JavaScript når brukeren utfører handling i appen. Hver hovedside, login.html, dashboard.html og project.html har sin egen dedikerte JavaScript-fil. I tillegg benytter applikasjonen egne JavaScript-filer som inneholder funksjonalitet som benyttes på tvers av prototypen. Ulike objekter inneholder ulik statisk og dynamisk data som kan hentes ut på tvers av siden.

Applikasjons-tilstand og overføring av data internt: Prototypen benytter “cookies” og “localStorage” for å simulere tilstand. Både “localStorage” og “cookies” benyttes til å lagre brukere. Prototypen benytter pr. dags. dato ikke noen form for database eksternt. Det er lagt til rette for slik funksjonalitet uten større endringer i applikasjonens struktur.

KILDEKODE

Her finnes tekniske detaljer utover det som er kommenter i koden. Se kommentarer i de respektive filene hvis du ikke finner dokumentasjon her. I dette dokumentet omtales kun filer og funksjoner av en uvanlig art.

Filstruktur:

gruppe31\_app (mappe)

- css (mappe)

- img (mappe)

- js (mappe)

- dashboard.html

- landing.html

- project.html

- welcome.html

Strøm: welcome -> landing -> dashboard -> project.

CSS (Mappe):

common.css:

Inneholder en rekke felles klasser som skrives på formen cm-<klassenavn>. Eksempel: “cm-grid” er en klasse som utvider “grid”. Her finnes også prosjektspesifikke klasser som ikke er skrevet på “cm” formatet, men som benyttes flere steder i applikasjonen.

dashboard.css, landing.css, project.css, landing.css og welcome.css:

Disse filene hører til html-filene med samme navn og leses fra topp til bunn. Samtlige filer starter med html,body og ender med en footer (hvis dette finnes på siden). Den respektive html-filen vil gi et inntrykk av hvor i css filen en klasse befinner seg. Ønsker du å finne et element der klasse-navnet starter med “cm” finnes dette i common.css.

JS (Mappe):

cookies.js: Denne filen inneholder funksjonene getCookie(cname) og setCookie(cname, cvalue, exyears). Disse brukes til å hente ut data og sette data inn i cookie strengen.

darkmode.js: Endrer og styler ulike elementer for å lage “dakmode”. Lagrer også dette i cookie.