

Prosjektoppgave Webutvikling H22

Prosjektoppgaven skal gjennomføres gruppevis og teller 40% av karakteren i faget.

Prosjektstart: onsdag 12.10.2022 (uke 41)

Innleveringsfrist: mandag 21.11.2022 kl 12:00 (uke 47)

I prosjektet kan dere enten forholde dere til den forhåndsdefinerte oppgavebeskrivelsen (webapplikasjon for matoppskrifter, se under), eller utvikle en løsning basert på egne ideer. En løsning basert på egne ideer må som et minimum ha en kompleksitet og et omfang som likner på den forhåndsdefinerte oppgaven. Kravene for å få god uttelling er uansett de samme, se "Følgende må være oppfylt for å få god uttelling på prosjektet" nederst i dette dokumentet.

Oppgavebeskrivelse

I prosjektet skal dere utvikle en webapplikasjon for matoppskrifter. Følgende krav og betingelser gjelder:

- Det skal være mulig å søke etter, legge til, endre og slette oppskrifter.
- Antall porsjoner må kunne justeres, og da må også mengden av hver ingrediens i en oppskrift oppdateres.
- Det skal være mulig å "like" en oppskrift etter at den er registrert.
- Det skal være enkelt å oppdage oppskrifter. Dette kan f.eks. gjøres gjennom søk, ved å liste opp nye og populære registreringer, ved å tilby filtrering basert på kategori, land, ingredienser osv.
- Det skal være mulig å legge oppskriftens ingredienser i en handleliste.
- Ingredienser må kunne fjernes fra handlelisten i etterkant.
- Handlelisten må kunne nullstilles.
- Systemet skal tilby en funksjon som foreslår oppskrifter basert på en eller flere oppgitte ingredienser.

Mulig tilleggsfunksjonalitet:

- Henting av oppskrifter fra eksterne kilder, f.eks. TheMealDB (<https://www.themealdb.com/api.php>)
- Deling av oppskrifter og handlelister, f.eks. via sosiale medier, e-post eller som weblenker.
- Anbefalinger som viser lignende oppskrifter.
- Brukerhåndtering gjennom pålogging med autentisering:
 - Teori:
 - Les om hashing teori her: <https://crackstation.net/hashing-security.htm>
 - Siden hashing skal være ressurskrevende: etter autentisering med hashing, brukes access/bearer tokens til etterfølgende forespørsler
 - I stedet for å implementere dette selv: bør bruke et autentiseringsbibliotek som for eksempel <http://www.passportjs.org/>
 - Autentisering ved hjelp av en 3. part også mulig, for eksempel Facebook eller Google, med OAuth2 protokollen (også støttet av passportjs).
- Responsiv layout som også fungerer godt på mobil.
- Ekstra fokus på utforming (fargevalg, typografi, animasjoner og bruk av multimedia)

Følgende må være oppfylt for å få god uttelling på prosjektet:

- Kravene i oppgavebeskrivelsen er implementert. Eventuell tilleggsfunksjonalitet kan bidra positivt i en helhetlig vurdering. Hvis oppgaven er egendefinert er det gruppen selv som definerer disse kravene, men løsningen må som et minimum ha en kompleksitet og et omfang som ligner på den forhåndsdefinerte oppgaven.
- Godt strukturert API som er RESTful.
- Tester av god kvalitet og solid testdekning på både klient- og serversiden.
- Statisk typesjekking på både klient- og serversiden.
- Bruk av Continuous Integration.
- Klient:
 - Oppdeling i komponenter.
 - Bruk av gjenbrukbare komponenter.
 - Bruk av enten *service-* eller *store*-objekter.
- Server:
 - Data lagret i og hentet fra en database.
 - Bruk av *service-* og *route*-objekter.
 - Datamodellen er fornuftig.
- En installasjonsveiledning (README.md) som beskriver hvordan man installerer og setter opp løsningen.

Presentasjon

Hver gruppe skal gi en kort demonstrasjon av den ferdige løsningen for faglærerne. Fokus skal være på produktet og ikke på kode eller prosess.

DCST2002: Det settes av 20 minutter per gruppe, 10 min til presentasjon og 10 min til spørsmål. Info om tid og sted blir kunngjort på BB.

INFT2002: Gruppene lager en videopresentasjon (skjermopptak med muntlig forklaring underveis), maks 10 minutter. Video-fil eller URL til video skal legges ved den endelige innleveringen.

Innlevering

Innlevering skjer i Inspera. Her laster hver gruppe opp en zip-fil som inneholder kildekode mm.