

Sensorveiledning

Emneinfo	
Emnekode: DAT152	Emnenavn: Advanced Web Applications
Vurderingsform: Skriftlig 4 timer	Semester: 2024 høst
Eksamensdato/leveringsfrist: 28/11-2024	Emneansvarlig/koordinator: Bjarte Kileng

Om sensorveiledningen

Alle vurderingsformer som teller med i endelig karakter i et emne skal ha en sensorveiledning til bruk ved sensur og eventuell ny sensur etter klage, jf. UH-loven § 3-9 (2). Den skal også være tilgjengelig for kandidatene etter publisering av resultatene, jf. § 5-3 (3).

Sensorveiledningen bør inneholde følgende punkt:

1. Løsningsforslag og/eller vurderingskriterier for oppgavesettet
2. Definisjon av typisk A-, C- og E-svar
3. Vektlegging av oppgaver/deloppgaver i oppgavesettet eller relevans av bruk av pensum
4. Bruk av karakterskalaen (vurderingskriterier)

Løsningsforslag og/eller vurderingskriterier

Ved gitte oppgaver, definer hva kandidaten må ha med i svaret sitt for å svare på hver oppgave, f.eks. i form av fasit ved matematiske eksamener, eller moment fra pensum i disiplinbaserte eksamener/muntlige eksamener. Ved individuelt utformede oppgaver, definer punkt/moment oppgaven bør inneholde. I mer åpne oppgaver er det aktuelt med vurderingskriterier.

Løsningsforslag/vurderingskriterier
Løsningsforslag er vedlagt i egen fil.

Definisjon av typisk A-, C- og E-svar

Hvilke forventninger er det til et A-svar, et godt gjennomsnittlig svar (altså C-svar) og et bestått svar (E-svar)? Dette punktet kan implementeres i løsningsforslaget.

Definisjon av svar
<p>Oppgavene 1:</p> <p>For et A-svar skal kandidaten ha en svært god kunnskapsmengde og innsikt (forståelse) om REST API, presis og riktig controller og service koden til problemstilling og riktig forklaring av OAuth2 resource owner credentials grant. I tillegg skal kandidaten ha en klar fremstilling (struktur) og benytte begrepene helt korrekt.</p> <p>For et C-svar skal kandidaten ha en god kunnskapsmengde og innsikt, men kan være noe upresis.</p> <p>For et E-svar kan kandidaten mangle en del vesentlige kunnskaper og være upresis med en del begreper.</p> <p>Oppgavene 2:</p>

For et A-svar skal kandidaten ha en svært god kunnskapsmengde og innsikt (forståelse) og han skal ha en klar fremstilling (struktur) og benytte begrepene helt korrekt. I tillegg skal kandidaten produsere riktig kode som svar til koderelaterte delen av oppgave.

Oppgavene 3:

For et A-svar skal kandidaten ha en svært god kunnskapsmengde og innsikt (forståelse) og han skal ha riktig kode som svar til koderelaterte delen av oppgave.

For et C-svar skal kandidaten ha en god kunnskapsmengde og innsikt, men kan være noe upresis og noe feil kode.

For et E-svar kan kandidaten mangle en del vesentlige kunnskaper og være upresis med en del begreper og feil kode.

Oppgave 4 (Universell Utforming):

For et A-svar skal kandidaten ha en svært god kunnskap og innsikt (forståelse) innen Universell Utforming (UU) generelt. Dette krever også inngående kjennskap til og kunnskap om den internasjonale tekniske standarden WCAG 2.1 for tilgang til innhold på web'en – hvordan standarden er bygd opp og hvordan denne brukes og kan realiseres, og da med spesielt fokus på de deler som inngår i Norsk regelverk for UU. Kandidaten skal også kjenne til og kunne gjøre god praktisk bruk av verktøy for evalueringer av websider iht. standard.

For et C-svar skal kandidaten ha en god kunnskapsmengde og innsikt i det som er angitt over, men kan være noe upresis og mangelfull i sitt svar.

For et E-svar kan kandidaten mangle en del vesentlige kunnskaper og være upresis med en del begreper.

Oppgave 5 (Web-sikkerhet):

For et A-svar skal kandidaten ha meget gode kunnskaper om begreper og metoder som brukes i web-sikkerhet og om hvordan en utvikler sikre web-applikasjoner. Kandidaten skal også kunne gjenkjenne alle sårbarheter i en gitt kode, og komme med gode løsninger for å takle disse sårbarhetene.

For et C-svar skal kandidaten ha gode kunnskaper om hvordan en utvikler sikre web-applikasjoner, kjenne til metoder og begreper som brukes i fagområdet. Kandidaten skal kunne avdekke de fleste sårbarheter i en gitt kode og komme med mottiltak til disse.

For et E-svar skal kandidaten kjenne til begreper og metoder som brukes i fagområdet. Kandidaten skal kunne skille mellom sikker og usikker kode i en web-applikasjon, og skal kunne avdekke de mest åpenbare sårbarhetene i en gitt kode og foreslå/skissere mottiltak til disse.

Oppgave 6:

For et A-svar skal kandidaten ha en svært god kunnskapsmengde og innsikt (forståelse). Kandidaten må gi presise og tydelige svar på alle spørsmål. All JavaScript kode må være oversiktlig og moderne, og følge nyere JavaScript-standarder.

For et C-svar skal kandidaten ha god kunnskap og innsikt. Svarene må i all hovedsak være riktige, men kan være noe upresise og utydelige. JavaScript kode må være oversiktlig, men kan bruke utdaterte eller ikke standard elementer.

For et E-svar kan kandidaten mangle en del vesentlige kunnskaper og være upresis eller utydelig. JavaScript kode kan ha større mangler i programlogikken. Studenten kan ha brukt utdaterte eller ikke standard elementer i JavaScript.

Vektlegging av oppgaver/deloppgaver

Hvis vektningen av oppgaver ikke går fram av oppgavesettet, må det defineres. Skal alle deloppgaver telle likt på sluttresultatet, eller summeres for hver oppgave? Hvilke deler av pensum har størst relevans for svaret? Dette punktet kan implementeres i løsningsforslaget.

Vektlegging

Vekten til oppgavene 1 til 6 er spesifisert i oppgaveteksten. Innenfor hver oppgave teller hver deloppgave likt. Når det samlede resultatet skal settes vil også spredningen telle med. Er det bare de enkleste spørsmålene som er besvart? Er det deler av pensum kandidaten ikke har forstått noe av?

Bruk av karakterskalaen

Definer hvordan kandidatens svar skal vurderes i forhold til løsningsforslaget eller vurderingskriteriene. Skal f.eks. «følgefeil» i matematiske oppgaver gi gjentatt trekk for hvert svar? Dersom det blir brukt «omregningstabell» fra poeng-/prosentskår til bokstavkarakter, må den legges ved.

Bruk av karakterskala

Oppgavene 1-3:

Det skal legges vekt på grunnleggende forståelse. Feil i syntaks, variabelnavn, metodenavn, manglende parametere, og annet utviklingsverktøyet automatisk ville gitt, skal det trekkes lite for. Upresise eller ufullstendige formuleringer og forklaringer kan ikke gi full uttelling.

Oppgave 4:

Det skal legges vekt på grunnleggende forståelse. Presise og fullstendige svar, formuleringer, forklaringer og diskusjoner i samsvar med (eller svært nært samsvar med) løsningsforslag skal det gis full uttelling for. Trekk gjøres ut fra svars grad av korrekthet/dekning i forhold til løsningsforslag.

Oppgave 5:

Det skal legges vekt på grunnleggende forståelse. Presise og fullstendige formuleringer, forklaringer og bruk av sentrale begreper skal gi full uttelling. Kjennskap om viktige tema som sårbarheter ved injeksjon og i autentisering og korrekt beskrivelse og koden gis full trekk. Trekk gjøres ut fra svars grad av korrekthet/dekning i forhold til løsningsforslag.

Oppgave 6:

Det skal legges vekt på grunnleggende forståelse. Feil i syntaks, variabelnavn, metodenavn, og manglende parametere trekkes det lite for. Upresise, utydelige og ufullstendige formuleringer og forklaringer gis mye trekk. Det gis også endel trekk ved bruk av utdaterte eller ikke standard JavaScript elementer. Det samme gjelder dårlig JavaScript kode.

6ai:

En vanlig feil er å blande sammen bruk av tråder med asynkron oppførsel.

Det gis ingen uttelling dersom det settes likhet mellom asynkron oppførsel og bruk av tråder.

Dersom studenten gir gode eksempler på asynkron oppførsel gis det poeng, også om det blandes inn bruk av tråder.

For maksimal uttelling på denne oppgaven bør studenten si noe om *addEventListener*, og ikke blande asynkron oppførsel med bruk av tråder.

6aii:

Uten bruk av *WebWorkers* er det bare et rett svar her, at nettleser, evt. fanen i nettleser fryser. For å få poeng på denne oppgaven må studenten forklare hva som skjer også uten bruk av *WebWorkers*.

6aiii

Det gis 50% for å for riktig type objekt, og 50% for kode for en *waitForSomething* funksjon.

Det eneste riktige svar for objekt type er en **Promise**.

Et kodeeksempel med riktig bruk av *fetch* gir full uttelling. Men; da må funksjonen returnere resultatet fra *fetch* uten bruk av *await*:

```
function waitForSomething() {  
    // Evt. mere kode  
    return fetch(...);  
}
```

Det gis trekk dersom det returneres en **Promise** som ikke inkluderer asynkron kode. For full uttelling må altså *waitForSomething* benytte *addEventListener*, *fetch* eller noe annet som involverer asynkron kode.

6b:

Se løsningsforslaget.

Det gis ingen uttelling for kun å kopiere kode fra oppgaveteksten.

6c:

Se løsningsforslaget.

Det gis trekk dersom løsningen ikke benytter de vedlagte HTML template fra oppgaveteksten.

Løsningsforslaget bruker en **Map** for å registrere callbacks som skal kjøres når info om eksamen skal oppdateres. Det er tilstrekkelig å kunne registrere kun en enkelt callback for dette.