

Sensorveiledning

Emneinfo	
Emnekode: DAT152	Emnenavn: Advanced Web Applications
Vurderingsform: Skriftlig 4 timer	Semester: 2021 høst
Eksamensdato/leveringsfrist: 6/12-2021	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 1-3: 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</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>Oppgave 4 (Universell Utforming): For et A-svar skal kandidaten ha en svært god kunnskap og innsikt (forståelse) innen Universell Utforming (UU) generelt – og spesielt innen design, implementering og testing av websider. Dette</p>

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, men minimumet er OWASP sin Top-10 liste. 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 vektingen 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-/prosentsskår 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 gis mye trekk.

5a:

Når det forventes mer enn ett riktig svar i spørsmålet, beregnes poenget som: antall riktige svar/antall utvalgte riktige svar.

5b:

Fullt poeng kan fortsatt gis uten 3.2

5c:

Se løsningsforslaget

5d(1):

Poeng kan trekkes dersom noen av de obligatoriske feltene ikke er nevnt. 50 % tildeles dersom signaturkravet ikke er nevnt, og alle obligatoriske felt er nevnt.

5d(2):

Samme som 5d(1)

5e:

Se løsningsforslaget

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:

Det er tilstrekkelig at studenten har forstått at Web Workers gir separate tråder, og at konsekvensen er at hovedtråden ikke blir blokkert når koden kjøres.

6aii:

I hovedprogrammet gis det ikke trekk om studenten ikke sjekker at nettleser har støtte for Web Workers.

I løsningsforslaget for tråd som utfører utregningen brukes **Array** sin *reduce* metode, men studenten kan bruke enhver løkkestruktur som gir riktig svar.

Det kan gis ca. 20% trekk om studenten ikke sjekker om element i listen er et tall.

6aiii:

For å få poeng på denne oppgaven må det i svaret komme fram at Web Workers ikke har tilgang til DOM-treet.

6bi:

For å få uttelling på denne oppgaven må kandidaten si noe om lasting av websider fra webtjener, og at ingen ny side lastes fra webtjener når bruker navigerer i SPA-applikasjonen.

6bii:

Det er tilstrekkelig at svaret lister viktige karakteristikker. Svaret trenger ikke gå i detalj om de forskjellige karakteristikkene. Hver av karakteristikkene fra løsningsforslaget teller likt, og alle (utenom *routing* som allerede er nevnt i oppgaven) må være nevnt for å få full poengsum.

6biii:

Første del av spørsmålet (*what is meant by routing*) teller 60%, mens siste del (*why do we need routing*) teller 40%.

For full poengsum på siste del må studenten diskutere at vi kan bokmerke ulike SPA tilstander (50%), og at nettleser sin historie-funksjonalitet (f.eks. fram og tilbake-knapper) vil virke også mellom ulike SPA tilstander (50%).

6biv:

Jeg forventer ikke her detaljerte kodeeksempler.

Det kan gis 70% rett hvis studenten vet at nettlesere har støtte for ruting ved bruk av URL-er med hash tegn, og at hendelse *hashchange* fyrer ved endring av hash-del av URL.

For full poengsum må studenten vite at ruting også er mulig ved vanlige URL-er, ved å opprette en ny tilstand for siden i history-objektet, og tilordne URL-en til denne tilstanden.

6ci:

Det gis 50% hvis studenten kun nevner at resultatet er at koden blir behandlet som en modul. For full poengsum må også konsekvensene av at denne impliserer *defer* utdypes.

- Koden kjøres etter at web-dokumentet er lastet gir 25%.
- Koden kan lastet parallelt med at web-dokumentet bygges gir 25%.

6cii:

Se løsningsforslaget.

6ciii:

Se løsningsforslaget.

Det gis ikke trekk om studenten bruker kun offentlige felt og metoder, da støtte for private felt og metoder (navn starter med hash tegn, #) inntil nylig har hatt dårlig støtte i nettleserne.