

Innlevering, PG4300 Ruby on Rails (60 %)

Norges Informasjonsteknologiske Høgskole

14. mai 2013

Oppgavesettet består av 7 oppgaver fordelt på 6 sider.

1 Rammer

- Frist for innleveringen er **tirsdag 28. mai, kl. 23:55**.
- Innlevering skjer via **itslearning**, som en **ZIP-fil**.
- Oppgaven skal løses **individuell**.
- Kandidaten skal benytte **Ruby 1.9** og **Rails 3.1 eller 3.2**.
- Benytt **SQLite 3** som database i utviklingsmiljøet.
- Kildekode og kommentarer i koden **skal være på engelsk**. Tekst som presenteres til brukeren kan være enten på engelsk eller på norsk, men **hold deg konsekvent til ett språk**.
- **Fyll databasen med noe testdata** før levering. Data som skal leveres inkluderer testbrukere og blogginnlegg (med kommentarer). La brukernavn og passord for testbrukere være dokumentert i vedlagt README-fil.
- Dersom applikasjonen trenger andre kommandoer enn de følgende tre – eksempelvis kommandoen `rake db:seed` – skal dette dokumenteres i vedlagt README-fil.
 - `bundle install`
 - `rake db:migrate`
 - `rails server`
- Husk å dokumentere eventuelle forutsetninger og kilder.

2 Vurderingskriterier

Følgende punkter vil telle positivt mot en vurdering dersom de er oppfylt. Ha disse punktene i bakhodet mens du løser oppgavene. De vil ikke nødvendigvis bli spesifisert direkte i oppgaven.

- Konsis og lesbar kode med riktig innrykk, formatering og navngivning.
- Idiomatisk kode og smarte løsninger.

- Konvensjoner i Rails er benyttet fremfor ekstra konfigurasjon.
- Innebygde hjelpemetoder i Rails er benyttet fremfor å skrive egne metoder/unødvendig kode. Ikke finn opp hjulet på nytt.
- Fornuftig fordeling av kode mellom model, view og controller. Tenk DRY, og bruk view partials, view helpers og filtre dersom det er fornuftig.
- Migrations brukes for endringer mot databasestruktur.
- Validering av input benyttes for å sørge for at bruker sender inn gyldig input.
- Applikasjonen er sikret mot SQL injection og XSS.
- Applikasjonen fungerer som beskrevet i oppgaven.
- Løsningen viser at du har oppnådd kompetansemålene i faget.

Merk at det vil gjøres en helhetlig vurdering av oppgaven, og at deloppgavene derfor ikke har spesifisert noen vektning.

Du trenger ikke tenke på følgende punkter i denne innleveringen:

- Pent brukergrensesnitt (GUI). Det holder at applikasjonen fungerer som beskrevet.
- HTML som validerer mot standarder fra W3C. Dette gjelder også CSS og JavaScript. Det holder med minimalt med HTML, CSS og JavaScript for at applikasjonen fungerer som tiltenkt.

Dersom du ønsker å lage et pent GUI eller kode som validerer mot standardene, så er det lov, men du vil ikke få noen ekstra poeng i vurderingssammenheng.

3 Tilbakemelding

Karakter vil bli gjort tilgjengelig for hver enkelt kandidat gjennom StudentWeb¹.

4 Oppgavebeskrivelse

Du skal utvikle et bloggsystem. Systemet skal tilby mulighet for brukerregistrering og -innlogging, og hver bruker eier én blogg hver. Blogger kan inneholde ett eller flere blogginnlegg, og hvert av disse innleggene kan igjen inneholde kommentarer, samt motta fist bumps.

Oppgave 1 Brukerregistrering/-autentisering

Det skal være mulig å registrere seg som bruker av systemet gjennom et brukergrensesnitt. En bruker skal (minst) bestå av følgende attributter:

- Brukernavn.
- Passord.

¹<https://www.studweb.no/as/WebObjects/studentweb2?inst=NITH>

- Fornavn.
- Etternavn.

Brukernavn og passord skal benyttes ved innlogging. Det forventes at passord lagres på en forsvarlig måte.

1.1 Krav til oppgaven

- Dersom man allerede er logget inn skal det **på enhver side** man besøker
 - vises en link til utlogging av systemet
 - vises en link til profilsiden for innlogget bruker
- Dersom man ikke er logget inn skal det **på enhver side** man besøker
 - vises minst én link til registrering av ny bruker
 - vises minst én link til innlogging av bruker
- Det skal opprettes et brukergrensesnitt for registrering av nye brukere.
- Det skal opprettes et brukergrensesnitt for innlogging av brukere.
- Det skal opprettes en profilside for brukere. Denne siden skal fylles ut i Oppgave 5 på side 5.
- Det skal **ikke** opprettes en liste over alle brukere (typisk en *index*-action).
- Brukere skal kunne slette sin egen brukerkonto. Ved sletting av bruker skal alle brukerens blogginnlegg også bli slettet.

Oppgave 2 Meldinger til brukeren

Etter hver CRUD-handling (eller andre, dersom det faller seg naturlig) skal brukeren få beskjed om handlingen ble fullført riktig eller ikke. Eksempelvis, dersom brukeren forsøker å opprette et blogginnlegg, men mangler nødvendig input, så skal det dukke opp en beskjed på skjermen som forklarer at innlegget ikke ble opprettet.

Dette innebærer også validering av input fra skjemaer brukeren fyller inn.

2.1 Krav til oppgaven

- Etter hver handling brukeren utfører som kan ha to eller flere utfall, skal en beskjed gis til brukeren. Det skal tydelig forklares *hva* som fungerte/gikk galt i beskjeden.
- Beskjedens natur skal være tydelig for brukeren. For eksempel skal det være lett å se på en beskjed at det er noe feil, eller at alt gikk som det skulle.

Oppgave 3 Blogginnlegg

Brukere ha mulighet til å opprette blogginnlegg til bloggen gjennom et grafisk brukergrensesnitt. Det skal også finnes en oversikt over alle innlegg i bloggen.

3.1 Krav til oppgaven

- Dersom man allerede er innlogget i systemet, skal det **på enhver side** man besøker vises en link til brukerens blogg (*Posts#index*).
- Blogginlegg skal minimum ha følgende attributter:
 - Tittel.
 - Innhold.
 - Publiseringstid og -dato.
- Det skal kunne opprettes, endres og slettes blogginlegg fra et grafisk grensesnitt i systemet.
- Det skal opprettes en overiktsside som viser alle blogginlegg for en blogg. Disse skal være sortert etter publiseringstidspunkt, slik at det nyeste innlegget vises først.
- Dersom det ikke finnes noen *publiserte* innlegg i bloggen, skal dette forklares med en tekst på siden.
 - Dette er med unntak av dersom brukeren ser på en egen blogg, i hvilket tilfelle ikke-publiserte innlegg skal vises som vanlige innlegg.
- Kun eieren av bloggen skal ha lov til
 - å opprette, endre og slette blogginlegg
 - å vise innlegg i sine private blogger
 - å vise innlegg som enda ikke er publiserte
- Når et blogginlegg slettes, skal alle tilhørende kommentarer og fist bumps (se henholdsvis Oppgave 4 og Oppgave 6 på neste side) slettes.

Oppgave 4 Kommentarer

Innloggede brukere skal kunne kommentere på blogginlegg, gjennom et grafisk brukergrensesnitt som vises i innleggets *show*-view. Det skal også vises en oversikt over alle kommentarene som er gitt på et innlegg på samme siden.

4.1 Krav til oppgaven

- Kommentarer skal minst inneholde følgende attributter:
 - Innhold.
- Det skal vises en oversikt over alle kommentarer som tilhører et innlegg på samme siden som innlegget vises (*show*-view).
 - Det skal stå oppført sammen med kommentaren hvem som har kommentert på innlegget.
 - Det skal stå oppført sammen med kommentaren når kommentaren ble opprettet.
- Dersom innlegget ikke har noen kommentarer, skal dette vises med en forklarende tekst.
- Kun innloggede brukere skal ha lov til å kommentere på innlegg.

- Det skal være mulig å slette sine egne kommentarer.
 - Kun personen som skrev kommentaren, samt forfatter av blogginnlegg, skal ha lov til dette.

Oppgave 5 Profilsiden

Hver bruker skal ha en profilside som viser informasjon om brukeren, samt en oversikt over brukerens fem siste publiserte innlegg.

5.1 Krav til oppgaven

- Profilsiden skal vise følgende informasjon om brukeren:
 - Brukernavn.
 - Fullt navn.
- En oversikt over brukerens fem (5) siste publiserte innlegg skal vises.
 - Innlegg som vises her skal sorteres etter når de sist ble publisert, i synkende rekkefølge. Med andre ord skal det nyest publiserte innlegget vises først.
 - Det skal være mulig å besøke hvert enkelt innlegg fra denne oversikten.
- Dersom profilsiden til den innloggede brukeren vises, skal linker for å redigere/slette profil vises.
- Dersom det ikke finnes noen publiserte innlegg i bloggen, skal dette forklares med en tekst på siden.
 - Dette er med unntak av dersom brukeren ser på sin egen profil, i hvilket tilfelle ikke-publiserte innlegg skal vises som vanlige innlegg.
- Innleggene skal listes opp på samme måte som i oversiktssiden over alle innleggene. Se Oppgave 3 på side 3.

Oppgave 6 Fist bumps

Innloggede brukere skal kunne vise respekt til andre brukere ved å gi blogginnlegg *fist bumps* (tilsvarende «likes» på Facebook). Det skal ikke være mulig å gi fist bumps til sine egne innlegg, kun til andres, og man skal kun ha mulighet til å gi én fist bump til hvert enkelt blogginnlegg. Dersom man mister respekten for en bruker eller et innlegg, skal man også kunne trekke tilbake fist bumps.

6.1 Krav til oppgaven

- Innloggede brukere skal kunne gi fist bumps til andre brukeres blogginnlegg.
 - Kun blogginnlegg som er synlige for en bruker skal kunne fist bumpes.
 - Hvert blogginnlegg skal bare kunne ha én fist bump fra samme bruker samtidig. Med andre ord: dersom jeg allerede har gitt fist bumps til innlegget «Gratulerer med π -dagen», kan jeg ikke gjøre det igjen, med mindre jeg først fjerner fist bump-en min.

- Det skal ikke være mulig å gi fist bumps til sine egne innlegg.
- Brukere skal kunne trekke tilbake fist bumps.
- En brukers favoritter skal kun kunne fjernes av brukeren selv.

Oppgave 7 Forside med brukersøk

For å finne andre brukere skal det være mulig å søke etter brukere på brukernavn. *Eksempel:* Det finnes brukere med følgende brukernavn i systemet: *test*, *testuser*, *stian*, *sondre* og *joachim*. Tabell 1 viser diverse søkeord og resultater for disse.

Søkeord	Resultater
test	testuser, test
st	testuser, test, stian
o	sondre, joachim
c	joachim

Tabell 1: Eksempler på søkeord og resultater.

7.1 Krav til oppgaven

- Det skal opprettes en forside til applikasjonen, som skal være tilgjengelig på roten av applikasjonen (<http://localhost:3000/>).
- Det skal **på enhver side** man besøker vises en link til forsiden.
- Forsiden skal inneholde et søkefelt, og ved å bruke dette skal en kunne søke på brukernavn.
- Søket skal returnere alle brukernavn som *inneholder* søkestrengen, ved å bruke SQL-kodeordet LIKE.
- Brukerne som blir funnet i søket skal presenteres på resultatsiden.
 - Det skal være mulig å klikke på brukernavnet til en bruker for å besøke brukerens profil.

5 Til slutt

Husk at det ikke settes noe krav til utseende utover hva som er beskrevet i oppgavene. Det er derfor anbefalt at dere fullfører oppgavene ordentlig først, for dere eventuelt ser på CSS og JavaScript dersom dere skulle ønske dette.

*Husk at utførelser utover det som er beskrevet **ikke** gir ekstrapoeng ved vurdering.*

Anbefaler også å sette av litt tid til levering av oppgaven, i tilfelle det skulle oppstå uventede feil i itslearning rundt leveringstidspunkt.

Husk å dokumentere alle forutsetninger dere gjør utover det som står beskrevet.

Lykke til!