1. **INLEDNING**
   1. **Allmänt om kursens upplägg och examination**

Denna kurs tar sin utgångspunkt i ett projekt där målet är att utveckla ett webbaserat informationssystem. Projektets genomförande är vägen mot kursplanens lärandemål och således är alla resurser i form av föreläsningar, handledningstillfällen och litteraturstudier också riktade mot projektet. Kursen examineras genom värdering av a) det slutliga systemet och b) en individuellt utförd muntlig och skriftlig värdering av ett system som utvecklats av en annan projektgrupp. Det egna systemet förevisas och granskas under ett särskilt examinationsseminarium (se nedan).

* 1. **Om projektuppgiften**

Projektuppgiften går ut på att bygga en CV-sajt i ASP.NET och MVC. Kraven på projektuppgiften är indelade så att det finns två uttalade nivåer vilka också är relaterade till kursens betygskriterier. Utförlig information om projektets genomförande ges vid det första kurstillfället (se även föreläsningsbilder på Blackboard).

1. **PRAKTISK INFORMATION**
   1. **Tidsåtgång**

Projektet löper under hela kursen och är beräknat att ta all tillgänglig tid i anspråk (beräknat på heltidsstudier exklusive den schemalagda tiden). Således är det av yttersta vikt att projektets planering påbörjas så fort som möjligt.

* 1. **Gruppindelning**

Projektet genomförs i grupper om tre personer och indelningen i grupper genomförs av kursdeltagarna själva. Se länk på Blackboard. **Gruppanmälan görs senast tisdag 14/12 klockan 14.00**. Om man inte har någon att arbeta med så skriver man upp sig i en grupp i sådant fall det finns 0–2 studenter i den gruppen. Om en student inte låter meddela sig vad gäller gruppanmälan kommer man inte att planeras in i examinationsschemat. Ett tips för att hitta samarbetspartners är att använda kursens diskussionsforum där särskild tråd för detta finns. Vi kan komma att flytta om grupperna under tisdag eftermiddag för att det i så stor utsträckning som möjligt skall vara tre i resp. grupp. Enda tillfälle fler än fyra tillåts är om det sker pga. det är fler studenter än jämnt antal 3 grupper dvs. ni kommer få ett nekande svar om ni frågar om ni kan arbeta i en grupp om 4.

* 1. **Läsanvisningar**

I största möjliga mån kommer de bitar som ni behöver för projektuppgiften att gås igenom under föreläsningarna, men ni kommer att behöva läsa på och söka information även på annat håll. Om det är någon annan information ni söker, så finns alltid hjälp via **handledningstillfällen** och **Blackboard.**

* 1. **Frågor**

Har ni frågor som är av allmänintresse? Ställ dem på handledningstillfällen eller via diskussionsforumet på **Blackboard**. Har ni specifika frågor kring just ert projekt? Skicka ett meddelande via Blackboard.

**Se till att använda Blackboard för att ställa frågor, ofta stöter man på liknande problem och kan ta hjälp av varandra!**

1. **CV-SITEN – APPLIKATIONSKRAV**

Sammanfattning: Bygg en webbsida som hanterar din och alla dina kamraters CV och projekt ni har jobbat med. Webbsidan ska kunna hantera att du själv uppdaterar din information via hemsidan dynamiskt och all information sparas i en databas. Med fördel kan du skapa en profilsida som du har riktig information om dig själv och såsom ditt CV är idag, så du på ett bra sätt kan visa för kommande arbetsgivare lite om dig själv \*hint\*.

Applikationen **måste** uppfylla följande krav för att uppnå betygen G såväl som VG.

* CV-siten ska byggas med hjälp av ASP.NET MVC.
* Fokus är inte på design, men CSS skall användas för att få ett konsekvent utseende på sajtens sidor. (Använd gärna bootstrap, som ingår i MVC-templaten i Visual Studio.)
* Data om CV-siten användare och övrig nödvändig information skall lagras i en databas.
* Systemet bör vara uppdelat i flera projekt för att efterleva god arkitektur, till exempel genom ett webbapplikationsprojekt och ett datalager.
* Applikationen skall innefatta minst en ApiController. Närmare beskrivning av funktionalitet finns under ”Allmänna krav”.
* Gränssnittsdelen av projektet skall skapas som en ”Web Application”.
* All inmatning i applikationen skall valideras.
* Applikationen skall använda sig av undantagshantering med informativa felmeddelanden.
* Koden ska vara skriven med en konsekvent syntax och namnsättning. Den ska dessutom vara kommenterad (där det behövs) och beskrivande till den grad att en student med kunskapsnivå som motsvarar studier på Informatik fortsättningsnivå-nivå enkelt kan följa funktionaliteten.
  1. **Allmänna krav (G)**

Applikationen **måste** uppfylla följande krav för att uppnå betygen G såväl som VG.

* Startsida som visar ett urval av CVs som är registrerade och länkar till personens profil vars CV det är. Startsidan ska även visa information om det senast upplagda projektet som personer kan koppla till sina profiler. Någonstans ska även det visas en login och en registrera funktion.
* Som användare skall det gå att logga in.
* Nya användare ska kunna registrera sig.
* En användare ska kunna redigera sina uppgifter. Man ska kunna:
  + Ändra sitt lösenord
  + Ändra profilinformation såsom namn, adress, privat profil etc.
  + Ändra uppgifter i sitt CV
    - Information som ska finnas med är minst; Kompetenser, Utbildningar, Tidigare erfarenhet
  + Ladda upp en bild på sig själv för sitt CV
  + Välja projekt som den ingått i (alla projekt som finns, även sådana man ej skapat själv)
* Varje användare har en CV sida där information om användaren visas upp.
  + Kontaktuppgifter
  + Information från sitt CV. Kompetenser (ex “ASP NET”, “Fullstack” etc.), Utbildningar (ex ORU, systemvetenskap), Tidigare erfarenhet (ex “Jobbade på ICA som kassör”)
  + Profilbild
  + Projekt som valts att ingå i
* En användare skall kunna ange om ens profil är privat eller inte.
  + **Om den är privat ska endast andra inloggade användare kunna se** den vars CV är privat. Den får heller inte listas på exempelvis startsidan om man valt att göra sig privat.
* Man ska kunna se en användares CV och de projekt som personen har valt in att denne ingått i, dock endast i inloggat läge om personens vars CV det är har valt att ha den privat.
* Man ska kunna skapa projekt, som har en titel, en beskrivning mm.
* De projekt användaren har **skapat själv** ska man även kunna editera.
* Projekten ska visas genom en sida som listar listar alla projekt.
  + All information om projekten finns antingen med i listningen eller som en separat sida som man länkas till. I informationen om projektet ska det visas vilka personer som valt att ingå i det. Användare som valt att ha sin profil privat ska **inte** visas i listningen, om personen som tittar **inte är inloggad**.
* När en person är inne på en användares CV sida så ska man kunna skicka ett meddelande till denne.
  + Meddelandet ska skickas till en APIController **via JavaScript/AJAX**. Som inloggad användare så **visas och bifogas** den inloggade användarens namn, annars måste man som anonym användare skriva i sitt namn innan man får skicka ett meddelande.
* Man ska kunna **söka efter användare** via deras namn, som inloggad användare ska då även privata CV komma med i sökresultatet, annars inte.
* Som inloggad användare så ska du kunna se de meddelanden du har fått.
  + Du ska även kunna markera meddelandet som “läst”.
* Som inloggad användare ska du kunna se om du har fått ett meddelande som du inte har läst än (som du ej har satt till “läst”).
  + Denna notifikation ska finnas på alla sidor, alltså **oberoende** vilken sida du befinner dig på (exempelvis i en header som säger: “du har xx olästa meddelanden”).
  1. **Utökande krav (VG)**

För väl godkänt (VG) ska, utöver kraven för godkänt (G), applikationen även ha minst följande funktionalitet:

* Spara information om hur många personer som varit inne på dennes CV sida.
* På **den egna** användarens CV ska denne kunna se hur många personer som varit inne på dennes CV. Denna information skall hämtas från din APIController **via javascript/AJAX**.
* En användare skall kunna avsluta sitt konto. Denna ska bli deaktiverad (obs, ej tas bort)
  + När ett konto avslutas så ska användaren inaktiveras och inte längre synas någonstans på sidan eller i sökningar.
* Användare ska kunna ta bort meddelanden som den har fått skickat till sig.
  + Innan meddelandet tas bort ska användaren få “bekräfta” borttagningen (du får inte använda en “window.confirm()”)
  + Borttagningen ska göras via **AJAX**
  + Man ska **inte** kunna ta bort andra användares meddelanden
* Varje användares CV sida ska ha en funktion för att “hitta liknande person” och systemet ska då matcha den persons vars CV man är inne på med en annan som verkar relevant.
  + Ni bestämmer själva hur matchningen ska genomföras, men den ska på något sätt vara baserad på profildata.
* Utöka söken att även kunna söka på kompetenser, samtidigt som du även kan söka på deras namn, samtidigt. (ex: “Nicklas” “.Net” ska anges samtidigt och matchas)
* Användare ska kunna ange sitt GitHub användarnamn, därefter när användare går in på deras CV sida så ska systemet via **AJAX** från en ApiController läsa in de repositories som github listar att användaren har.
  + Systemet hanterar inläsningen av data från github, som den sedan skickar vidare ut till frontend(webbsidan).
  + APIet för githubs användares repositories nås via: [https://api.github.com/users](https://api.github.com/users/DegJ/repos)/ANVÄNDARNAMNHÄR/repos
  + Exempelanvändare: <https://api.github.com/users/DegJ/repos>
* Applikationen skall innehålla en exportera profil-funktion som tar befintlig profildata (inkl CV och ev. skapade projekt) och skapar en XML fil (skall ske genom serialization), den ska sedan ges att få “laddas ner” i användarens webbläsare.
  + Användarens lösenord får **inte** komma med någonstans i xml filen

Utöver de här kraven är det fritt fram att implementera egna idéer och övrig funktionalitet!

* 1. **Krav på miljö**

Koden ska vara körbar på en installation av Visual Studio.

**Ingenting ska installeras på IIS – allting ska köras i utvecklingsservern i Visual Studio.**

**Tänk på att använda relativ sökväg i connectionstring så att den inte pekar på en sökväg som endast finns lokalt på er dator. Då fungerar applikationen på alla datorer utan konfigurering. Testa på olika datorer innan inlämning!**

1. **INLÄMNING AV PROJEKTUPPGIFT**

Er applikation ska packas ihop i en .rar eller .zip-fil. Så små filer som möjligt uppskattas så kom ihåg att tömma bin- och obj-katalogerna innan ni packar ihop projektet. Alla JavaScript, bilder och CSS-filer skall ligga med i mappstrukturen i projektet. Detta gäller även databasen (MDF+LDF), som beskrevs ovan under rubrik 3.3. Kod som inte används ska städas bort innan inlämning. Det packade projektet postas på för ändamålet särskilt avsett blackboardforum **senast den 10/1 kl. 23.59.**

För att ett projekt skall vara bedömningsbart gäller följande:

* systemet är inlämnat i tid
* systemet är körbart

Ej bedömningsbara system medför kursbetyg U (underkänt) med möjlighet till bedömning vid kursens senare examinationstillfällen (se nedan).

1. **Seminarium och granskning av annan grupps system**

I den muntliga och skriftliga redovisning som uttrycks i kursplanen ingår att individuellt granska och utvärdera ett system som utvecklats av en annan projektgrupp. Granskningen sker skriftligt utifrån en uppsättning kriterier och kommuniceras sedan till systemets utvecklare muntligt under ett seminarium. Utvecklarnas roll under seminariet blir att visa sin applikation samt besvara granskarnas och lärarnas frågor om systemet (vi tillämpar alltså ett oppositionsförfarande). Granskaren (opponenten) skall även lämna in sin värdering skriftligt till systemets utvecklare (respondenterna) samt ansvariga lärare via Blackboard senast den **11/1 kl. 23.59.** Underlag för oppositionen finns i dokumentet ”Oppositionsunderlag” som anslås på Blackboard. Schema för opponering kommer att presenteras på Blackboard när samtliga grupper har anmält sig.

Alltså:

* varje student skall tillsammans med sin gruppkamrat förevisa och kunna svara på frågor om lösningar i sitt eget system.
* varje student skall enskilt (inte tillsammans med tidigare gruppkamrater) värdera en annan grupps system genom att besvara en specifik fråga angående respondentens applikation.
* granskning sker inför ett seminarium och värderingen kommuniceras både muntligt och skriftlig.

1. **EXAMINATION, VÄRDERINGSSKALA, VÄRDERINGSKRITERIER OCH BETYG**

Examination på denna kurs kommer att bestå av två delar vilka bedöms av examinerande lärare. Det som bedöms är:

1. Bedömningsbart system (den version som finns på inlämningsforumet vid deadline)
2. Muntlig och skriftlig granskning av annan grupps system (muntlig opposition under seminarium samt inlämnad skriftlig granskning).

Den omdömesskala som tillämpas är ”minus”, OK respektive ”plus” enligt figur 1 nedan.

**Figur 1. Examinationens värderingsgrunder**

| **System** | |
| --- | --- |
| Omdöme | Värderingskriterium |
| plus (+) | *Systemet uppfyller alla de allmänna kraven enligt avsnitt 3.1, samt alla de utökade*  *kraven enligt avsnitt 3.2 ovan.* |
| OK | *Systemet uppfyller alla de allmänna kraven enligt avsnitt 3.1 ovan.* |
| minus (-) | *Systemet uppfyller inte alla de allmänna kraven enligt avsnitt 3.1.* |
|  |  |
| **Granskning** | |
| OK | *Omdöme OK ges för en granskning som uppvisar alla aspekter enligt följande:*   * *granskning har kommunicerats skriftligt och muntligt* * *granskning omfattar alla uppsatta granskningskriterier*   - *granskningen innehåller såväl utlåtande som motivering vilka uppvisar de kunskaper och förmågor som uttrycks i kursplanen* |
| minus (-) | *Omdöme minus (-) ges för en granskning som uppvisar något av följande:*   * *granskning har inte kommunicerats skriftligt och muntligt* * *granskning omfattar inte alla uppsatta granskningskriterier* * *granskningen innehåller inte utlåtande eller motivering* * *utlåtande eller motivering vilka inte uppvisar de kunskaper och förmågor som uttrycks i kursplanen* |

* 1. **Betyg**

I enlighet med kursen kursplan används Underkänd (U), Godkänd (G) eller Väl Godkänd (VG) som betyg på kursen. För kursbetyg godkänt (G) krävs minst omdömet OK för systemet samt omdömet OK för granskningen. För kursbetyget väl godkänt (VG) krävs omdömet plus för systemet samt omdömet OK för granskningen.

* 1. **Kompletteringar**

Vid bedömningen minus på någon av de båda examinationsdelarna ges (en) möjlighet till komplettering. Utgångspunkten för kompletteringsbehovet/kravet vad gäller systemet är de krav som uttalats i uppgiftsspecifikationen enligt ovan. Kompletteringen sker genom förändringar respektive inlämning av ny systemversion. Examinerande lärare specificerar kompletteringskravet skriftligt.

Kompletteringsbehovet/kravet vad gäller granskningen är de värderingskriterier som uttalats i

 dokumentet ”Riktlinjer för granskning” (se blackboard).  Kompletteringen utförs skriftligt men

använder det egna systemet som utgångspunkt. Kompletteringsuppgiften innebär alltså det egenutvecklade systemet värderas utifrån ett eller flera kriterier. Examinerande lärare specificerar kompletteringskravet skriftligt.

Om inte annat avtalats är tiden för komplettering 7 dagar efter meddelat betyg. Någon möjlighet till komplettering av kompletteringen ges ej.

**Om någon gruppmedlem avbryter kursen -> obligatorisk avstämningsinlämning**

Om någon gruppmedlem avbryter kursen skall detta meddelas omgående från gruppmedlemmarna till kursansvarig. Det arbete som genomförts fram tills det att medlem lämnar gruppen är i normalfallet att betrakta som gemensam prestation (programkod, dokument etc.) och kan komma att användas som underlag då den student som avbryter kursen önskar genomföra/examinera kursen på nytt. För att detta skall vara möjligt krävs dock att den student som avbryter kursen gör en ”avstämningsinlämning” av det material som tagits fram t.o.m. tidpunkten för avbrottet och snarast kontaktar kursansvarig för instruktioner om hur detta skall gå till.

1. **NÄSTA EXAMINATIONSTILLFÄLLE**

Nästa examinationstillfälle för kursen är i slutet av februari 2022 enligt samma upplägg som ovan. Information om exakt inlämningstid för omexaminationen, oppositionsordning etc. anslås senare på kursens Blackboardsida. Det efter omexaminationen nästkommande examinationstillfället är under perioden för uppsamlingsexaminationer i augusti 2022.

Mer exakt information om uppsamlingstillfället ges av Handelshögskolans utbildningsadministration (AnnaCarin Gustafsson). Vid båda dessa examinationstillfällen är det projektuppgiften ovan som gäller som underlag. Praktiskt sett är det alltså fullt möjligt att arbeta vidare på ett tidigare underkänt projekt och examinera en omarbetad version vid ett senare examinationstillfälle.