



KLIENTPROGRAMMERING FÖR WEBBSYSTEM

PROJEKTUPPGIFT

Klientprogrammering för webbsystem

PROJEKTUPPGIFT

Läs noggrant igenom hela dokumentet innan ni börjar. Säkerställ att ni har förståelse för vilka krav som ställs. Se även *bedömningsmallen* som kommer att tillämpas vid examinationen i slutet på dokumentet.

ALLMÄNT

Utgångspunkten för projektuppgiften är att ni skall få tillämpa allt kursinnehåll praktiskt samtidigt som ni utvecklar något som är användbart. Ni skall skapa en fiktiv företagshemsida. Denna hemsida kan ni t ex ha som referensarbete när ni i framtiden söker arbete.

Ni ska arbeta i grupp om 4 studenter men projektet innehåller både gemensamma och individuella delar. Kraven på projektuppgiften är indelade så att det finns två uttalade nivåer, en för godkänt (G) betyg och en för väl godkänt (VG) betyg. De gemensamma delarna är endast på G-nivå medan den individuella delen finns på både G- och VG-nivå. Det innebär dock inte att de gemensamma delarna är mindre viktiga, för att gå godkänt på kursen måste dessa få betyget G. Tanken är att lära sig att arbeta gemensamt på ett projekt och hur man kopplar samman olika delar. Utöver detta är det kollektiva lärandet även en viktig del av kursens pedagogik. Det kollektiva lärandet är otroligt kraftfullt, genom att förklara något för en annan person fördjupas ens egna kunskaper.

Får ni då samarbeta med de individuella delarna? Svaret är ja, samarbete är tillåtet, de individuella delarna ska snarare ses som en möjlighet till individuellt arbete än ett påtvingat sådant. Vissa föredrar att arbeta helt i grupp och då är det tillåtet. Däremot så är det viktigt att **alla** lär sig enligt kursens kunskapsmål och har förståelse enligt dem. Därmed kommer var och en av er i gruppen att få förklara och redovisa någon del av projektet vid presentationen. Vilken del slumpas av oss lärare så det kommer inte att vara möjligt att "plugga" in detta. Det gäller alltså att förstå de lösningar som ni har implementerat, såväl individuella som gemensamma. Det är alltså väldigt viktigt att alla i gruppen är delaktiga i de gemensamma delarna.

Min rekommendation är att ni börjar med de gemensamma delarna och lär av varandra och sedan går vidare och börjar på de individuella när ni har en grundstruktur för projektet. De gemensamma delarna syftar nämligen till att ge er detta. Det är även viktigt att ni gör arbetet stegvis, att ni först säkerställer att ni har en fungerande grund, dvs. ett HTML-dokument som är validerat och utan fel. Att ni har en layout för projektet att sedan bygga i. Allt bygger på det och har ni inte en fungerande grund kommer ni få stora problem i ett senare skede. Följ kursens stegvisa upplägg: HTML, CSS, JavaScript, Vue.js. Börja inte på någon del innan ni har förstått den tidigare delen. Planera innan ni börjar! Från tidigare år har jag sett exempel på projekt som byggt på en icke-fungerande HTML-grund och då blir det fortsatta arbetet problematiskt. Jag kan därför inte nog understryka vikten av att ni **validerar er HTML innan ni fortsätter**.

I största möjliga mån kommer det som ni behöver för projektuppgiften att gås igenom under kurstillfällena och utlämnade övningsuppgifter, speciellt för de krav som ställs för betyget G. VG-kraven kräver lite mer egen insats. Under projektet kommer ni säkerligen stöta på utmaningar, Internet är en bra resurs för att hitta lösningar på problem. Utöver det finns även hjälp via handledningstillfället och Blackboards diskussionsforum. Om ni tar hjälp, se till att ni fortfarande förstår koden och att den "är er egen" (annars kommer ni inte att klara examinationen).

PLANERING

Hur ni planerar ert projekt är upp till er i gruppen att komma överens om. Min rekommendation är dock:

1. Diskutera innehåll och struktur, läs projektuppgiften noga och planera för genomförandet, vilka delar som skall finnas, vilken funktionalitet, layout etc. och gör skisser så att ni vet vad ni ska bygga innan ni börjar bygga det.
2. Börja med HTML – vilket innehåll ska finnas på vilka sidor? Börja enkelt! Utgå sedan från innehållet och skissa upp en layout. Layouten ska göra innehållet så tillgängligt som möjligt för era besökare. Börja implementationen med layouten först. Gör de gridsystem ni behöver och se att de verkligen fungerar, är responsiva, innan ni går vidare. Att ha en layout att lägga in era element i gör att ni kan sätta mer egenskaper globalt och färre för specifika element. Det gör projektet väldigt mycket enklare.
3. När ni har en fungerande struktur kan ni börja fundera på utseende lite mer utförligt. Utseende fick ni ju genom er layout, men typsnitt, färger, etc. Ska alla rubriker ha samma typsnitt? Ska stycketext ha ett annat? Börja fundera på det globala innan ni ger er in i det specifika.
4. JavaScript och Vue.js bygger ni vidare på när ni har en fungerande grund att stå på.

GEMENSAM DEL AV PROJEKTET

Det som skall presenteras på den gemensamma delen av projektet är följande:

- **Presentation av företaget** – här presenterar ni företaget, vad ni heter, vad ni gör, vad ni kan erbjuda. Har ni svårt att komma på lämplig text, fråga något lämpligt generativt AI-verktyg om hjälp. Ni ska inte lägga mycket tid på att komma på bra texter även om det gör mycket för intrycket av projektet att det faktiskt finns ett någorlunda vettigt innehåll (dock ej det som bedöms!). Innehåll är valfritt men minst en rubrik samt en paragraf.
- **Anställda/konsulter** – här presenterar ni företagets "anställda", dvs. er själva (eller påhittade personer). "Ni" är ert företags personal i den form och med den roll ni själva väljer. På denna sida skall samtliga anställda presenteras med bild och kort text. Bilden behöver givetvis inte föreställa er själva och texten kan vara helt påhittad. I princip är detta alltså en presentationssida över er gruppmedlemmar fast i egenskap av anställda på ert fiktiva företag. Från denna presentation skall man sedan kunna klicka sig till varje anställds personliga sida (individuell del av projektet).
- **Kontaktsida** – det ska finnas en kontaktsida innehållande ett formulär. I formuläret skall besökaren kunna skriva in namn, telefonnummer, e-postadress samt ett meddelande. Formuläret ska *realtidsvalideras* när användaren matar in information, det ska ske en kontroll att det som matas in är i enlighet med det förväntade och användaren ska få feedback INNAN den klickar på skicka-knappen.

Ni väljer själva om ni vill ha allt på samma sida eller dela upp det på olika sidor. Det finns en stor poäng så här i början med att dela upp logik i olika dokument eftersom allt i samma fil kan bli rörigt och svårare att överblicka. Det kan också, om man inte har helt koll, bli konstigt när logik från olika sidor påverkar varandra. Men detta är ert val inom gruppen. Diskutera igenom och välj det ni känner er mest bekväma med.

KRAV PÅ HTML

HTML-dokumentet ska följa en *standard* och *validera* enligt den. De ska vara *välstrukturerade*, koden ska vara *indenterad* så att det är lätt att se relationerna. Det får inte finnas CSS i HTML-dokumentet (bara

inlänkningen av den fristående filen). Att lägga CSS i HTML-dokumenten görs "i verkligheten" och det kommer ni att se, men i den här kursen lär vi oss att separera för att få sidor som är lätta att underhålla, en effektiv kod som inte upprepas i onödan, möjligheterna att styra vårt utseende helt och hållet beroende på förutsättningarna etc. Det är även för er skull för att ni inte ska få för komplexa dokument inledningsvis.

<https://validator.w3.org/>

KRAV PÅ CSS

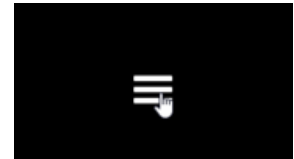
Att inte upprepa kod i onödan, att det finns en logik bakom koden, att egenskaper som är globala sätts på en global nivå för att sedan ärvas till andra element och att egenskaper som är specifika anges på de element de gäller. Koden ska validera.

- **Responsiv layout**

Layouten ska vara responsiv med *minst* kravet att sidan skalar om så att allt innehåll blir tillgängligt för besökaren. Tillgängligt är aningen oprecist så tänk logiskt, att t ex behöva scrolla sidledes för att läsa en text är inte tillgängligt och användarvänligt. Sätter ni storlekar i fasta värden, säkerställ att dessa värden fungerar både för breda och smala viewports. Fasta värden kan användas för att definiera t ex en kolumns storlek och sedan kan flexbox användas för att avgöra hur många kolumner som får plats. Därmed är det inte på något sätt förbjudet att använda fasta värden, men det ska ske genomtänkt och den övergripande layouten på sidan ska vara responsiv.

- **Två olika menyutseenden**

Det ska finnas två olika meny-utseenden. För mobil skall menyn visas som en navicon (se bild). När man klickar på naviconen ska en meny med innehållet i rader visas. Detta skall ske utan att sidan behöver laddas om. För varianten med högre upplösning (t ex min-width: 768px) skall menyn och länkarna presenteras utan att användaren behöver klicka för att visa.



<https://jigsaw.w3.org/css-validator/>

KRAV PÅ JAVASCRIPT

Realtidsvalidering av formulär är grundläggande JavaScript. Denna grundläggande uppgift möjliggör för er att gemensamt komma fram till en lösning på hur man använder JavaScript för att påverka feedback till användaren baserat på vad användaren gör, att ge användaren feedback i realtid när den behövs. Genom att lösa denna uppgift i grupp får ni den kunskap ni behöver för att kunna göra lite mer avancerade funktioner i den individuella uppgiften.

Det som ska valideras är att fältet Namn innehåller tillräckligt många tecken, består av text, att telefonnummer är korrekt angivet (kan bestå av både siffror och +- men inte av bokstäver), att e-postadressen är skriven i rätt format (@ och . måste ingå), att obligatoriska fält är ifyllda. Formuläret behöver inte kunna skickas. För att validera i realtid ska ni använda JavaScript och reagera på vad användaren skriver och ge feedback utifrån det. Det räcker således inte att validera formuläret enbart genom HTML-attribut för formulär (t ex required). Ge användaren visuell feedback, t ex röd/grön text beroende på om inmatningen är korrekt utförd eller ej.

KRAV PÅ VUE.JS

Det är inga krav på Vue.js i den gemensamma delen, endast i den individuella. Därmed är det inte förbjudet, vill ni använda Vue.js till något så är det tillåtet. Anledningen till att Vue.js endast är i den individuella är pga den stegvisa pedagogik som tillämpas i kursen, att stegvist öka komplexiteten.

INDIVIDUELL DEL AV PROJEKTET

Den individuella delen av projektet är den personliga sida som nås genom att användaren klickar på någon person på sidan anställda/konsulter. Den personliga sidan ska, precis som resterande delar av webbplatsen, ha två olika menyutseenden. Denna del av den gemensamma utvecklingen ska således återanvändas till de individuella sidorna. Det är däremot fritt att bestämma över layout och design på den individuella sidan så länge som layouten är responsiv. Sidan skall innehålla (G/VG):

- Bild – kan vara samma som på den gemensamma sidan (G/VG).
- Presentation – lite utförligare text än på den gemensamma sidan, kan vara påhittad (G/VG)
- Projekt som just du ansvarat för – projektets titel, kund, kort beskrivning + frivillig mer information. Minst 5 projekt skall infogas (ej direkt i HTML-dokumentet!) och presenteras på ett överskådligt sätt (G/VG).
- Bildspel (G/VG)
- Dina kompetenser – i form av skillbars (VG)

KRAV PÅ HTML

Den individuella sidan ska uppfylla samma krav som för den gemensamma delen, dvs.

HTML-dokumentet ska följa en *standard* och *validera* enligt den. De ska vara *välstrukturerade*, koden ska vara *indenterad* så att det är lätt att se relationerna. Det får inte finnas CSS i HTML-dokumentet (bara inlänkningen av den fristående filen). Att lägga CSS i HTML-dokumentet görs "i verkligheten" och det kommer ni att se, men i den här kursen lär vi oss att separera för att få sidor som är lätta att underhålla, en effektiv kod som inte upprepas i onödan, möjligheterna att styra vårt utseende helt och hållet beroende på förutsättningarna etc. Det är även för er skull för att ni inte ska få för komplexa dokument inledningsvis.

<https://validator.w3.org/>

Detta gäller såväl för G som för VG.

KRAV PÅ CSS

Den individuella sidan ska uppfylla följande krav för att uppnå godkänt (G) betyg:

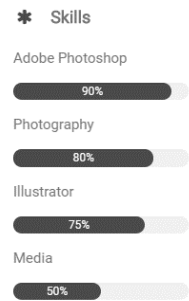
- **Responsiv design (G)**

Precis som för de gemensamma delarna ska den individuella sidan anpassa sig efter förutsättningarna och fungera för olika enheter. Innehållet skall vara lika tillgängligt för en besökare som använder en mobil som för en besökare som använder en dator. Hur du implementerar din responsiva design är upp till dig, du kan t ex använda flexbox eller grid layout men innehållet skall inte bara skalas om utan *anpassas* efter hur mycket plats som finns tillgängligt. Layouten ska på något sätt förändras när bredden på viewport väsentligt förändras (**minst två olika layouter**). Media queries kan användas för att göra denna anpassning (t ex vid användning av grid layout). Använder ni uteslutande flexbox (flytande layout) för att få automatisk anpassning skall ni ändå använda media queries någonstans för att visa att ni hanterar denna teknik. T ex att göra texten lite större vid bredare viewports och lite mindre i smalare.

För att få väl godkänt betyg (VG) skall kraven för G uppfyllas samt nedanstående krav:

- **Kompetenser i form av animerade skillbars (VG)**

När ni presenterar era kompetenser skall detta göras illustrativt med staplar som animeras, se bild till höger för inspiration. När sidan laddas skall staplarna animeras så de fylls i, detta kan göras med CSS (förändring av egenskaper och övergång (transition) när detta sker t ex). Funktionaliteten som efterfrågas benämns vanligen *animated skill bars* ifall ni behöver söka efter information om möjliga lösningar. OBS! Ifall ni söker och hittar en lösning på nätet, se till att göra koden till er egen och att ni förstår den. Ni ska *förstå och kunna förklara koden*. All personlig information kan vara påhittad!



KRAV PÅ JAVASCRIPT

Webbsidan skall uppfylla följande krav för att uppnå betygen G såväl som VG.

- **Bildspel (G)**

På den individuella sidan ska det finnas ett bildspel i vilket man kan bläddra mellan olika bilder. Bildspelet ska inte snurra automatiskt utan användaren måste klicka för att se nästa bild. Det ska vara möjligt att stega både bakåt och framåt i listan av bilder. Det måste således finnas två knappar i bildspelet. Hur dessas knappar ska se ut bestämmer ni själva. Ni får inspireras av färdiga lösningar på nätet men ta inte ett färdigt skript rakt av utan gör koden till er egen och försök själva först. De delar som behövs för att lösa uppgiften har gått igenom på kursträffarna så det handlar om att förstå och tillämpa.

- **Transitions/animeringar styrda av händelse (VG)**

Med JavaScript kan man fånga händelser, det har ni redan gjort tillsammans (realidsvalideringen av formuläret). Man kan använda detta inte bara för att ge feedback utan för att ge lite mer upplevelse. T ex när användaren scollar sidan och visst innehåll hamnar i fokus kan man reagera på detta och förändra innehållet på något sätt för att ge det lite mer fokus. Ni har redan gjort det med CSS så CSS-funktionaliteten kan ni. Ni har även fångat händelser i JavaScript. Nu gäller det att knyta ihop dessa tekniker. Ni kan låta något innehåll komma in från sidan, åka ut åt sidan, hoppa uppåt, dvs. ändra position, alternativt ändra storlek eller liknande. Det finns ett enkelt exempel från förra året då jag inte gick igenom detta i CSS under första kursveckan. Väljer ni att titta på detta exempel får ni se det som inspiration men ni får, precis som med annan inspiration, **inte** ta det rakt av. Ni ska göra något eget. Det är upp till er kreativa frihet att välja hur. Animeringar och transitions används vanligtvis för att ge webbsidor ett mer modernt och sofistikerat intryck. Boken kapitel 18 går igenom detta och på <https://tympamus.net/codrops/> finns massor av inspiration, t ex:

Scroll-based Layout-animations: <https://tympamus.net/Development/ScrollBasedLayoutAnimations/>

On-Scroll Filter: <https://tympamus.net/Development/OnScrollFilter/>

Connected Grid: <https://tympamus.net/Development/ConnectedGrid/>

Inspiration finns i mängder på nätet. Många moderna hemsidor använder denna teknik, t ex:

<http://manifesto.clapat.com/>

Ni behöver inte (bör inte) göra det så avancerat som ovan, se det som inspiration (det kan också bli för mycket av det goda). Boken vägleder er till grunderna i transitions och animering. Kravet i denna uppgift är att ni kan visa upp att ni behärskar transitions och animeringar (CSS/HTML) och att ni kan koppla det mot någon händelse fångad i JavaScript. Observera att detta är en VG-uppgift, den är således lite mer utmanande än övriga. Börja därför med G-kraven och ta denna på slutet om tid och motivation finns (så ni inte fastnar på denna).

KRAV PÅ VUE.JS

En sak ni kommer att märka när ni börjar med Vue.js är att koden inte längre validerar. Tipset för att ändå kunna validera koden är att inte validera den kod ni skriver utan den kod som genereras. Webbsidan skall uppfylla följande krav för att uppnå betygen G såväl som VG.

- **Projekt läses in från textfil (G)**

De projekt som just du ansvarar för ska INTE läggas in som innehåll i HTML-dokumentet. Istället skall innehållet läsas in från en textfil (innehållande JSON-data). Du väljer själv om du vill använda fetch eller axios.

För att få väl godkänt betyg (VG) skall kraven för G uppfyllas samt nedanstående krav:

- **Projekten ska kunna filtreras och sorteras (VG)**

Den lista över projekt som lästes in via fetch eller axios ska kunna filtreras och sorteras. Användaren ska kunna ange t ex att bara se projekt som börjar på eller innehåller bokstaven a (exempel). Projekten ska kunna sorteras i exempelvis bokstavsordning. Listan ska uppdateras i realtid när användaren skriver kriterierna.

EXAMINATION

För att erhålla betyget godkänt (G) skall samtliga allmänna krav vara uppfyllda, både för den gemensamma och den individuella delen. För att erhålla betyget väl godkänt (VG) skall samtliga allmänna och utökade krav vara uppfyllda vid tidpunkten för examinationen. Det går ej att i efterhand komplettera till ett högre betyg. Anledningen är att alla ska ha samma förutsättningar i form av tid.

När det gäller själva bedömningen av projektet så styrs den av att kraven är uppfyllda på den nivå som eftersträvas. All funktionalitet skall finnas och fungera. Skulle det däremot förekomma något försumbart fel i koden (en felskrivning eller ett missat semikolon t ex) så kommer vi ändå att bedöma projektet som godkänt under förutsättning att ni kan förklara och visar förståelse för er kod.

Att ni har utvecklat den kunskap som kursen syftar till och att ni uppfyller kunskapsmålen är det viktigaste. Förklaringen av koden är därmed central för bedömningen (eftersom kod kan genereras). Projektet examinerar att ni kan tillämpa kunskapen (genomförandet av uppgiften) och att ni kan förklara (redovisningen vid examinationen).

Vid examinationen kommer ni att få redovisa som grupp samt enskilt inför en av kursens lärare. Examinationsschema kommer att publiceras under sista kursveckan på Blackboard. Varje projekt examineras under cirka 25 minuter:

- 5 minuter demonstration av projektet
- 5 min per gruppmedlem att förklara utvald, slumpad del (totalt 20 min).

Momentet att förklara är således begränsat i tid men eftersom ni bara förklarar en vald del räcker denna tid. Det som avses är inte att varje tecken ska förklaras utan att lösningen och dess uppbyggnad ska beskrivas på ett sådant sätt att vi kan bedöma att ni förstår er egen kod. Märker vi att ni har svårt att förklara kommer vi att ställa frågor för att hjälpa till. Det som bedöms är förståelse för de implementerade lösningarna.

För G på kursen:

- G på gemensamma delar
- G på individuell del av projekt
- G på muntlig redovisning

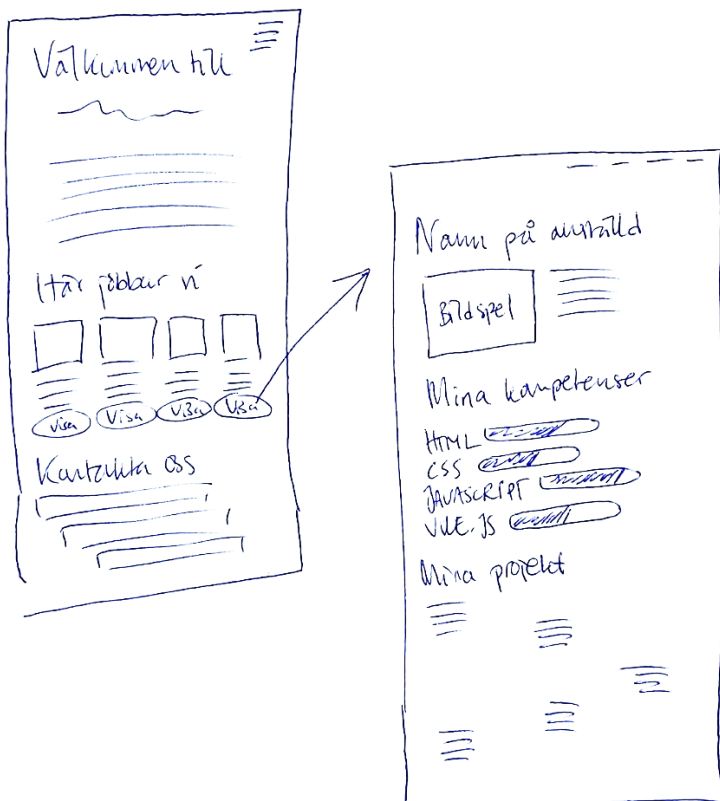
För VG på kursen:

- G på gemensamma delar
- VG på individuell del av projekt
- G på muntlig redovisning

INLÄMNING

Ni lämnar in ert projekt i angivet diskussionsforum i Blackboard. Se Kronox för deadline. Ert projekt ska packas ihop till en fil. Glöm inte att städa koden innan inlämning. Kod som inte längre används skall tas bort.

Lycka till och ha kul med uppgiften!



BEDÖMNINGSMALL

Grupp _____

#	Gruppmedlemmar
1	
2	
3	
4	

GEMENSAM DEL

Det finns...	Ja	Nej
... en presentation av företaget (rubrik, paragraf)?		
... en presentation av anställda/konsulter (bild, text samt länk vidare)?		
... kontaktsida som <i>realtidsvalideras</i> med hjälp av JavaScript?		
HTML-dokumentet följer standard och validerar*?		
CSS-dokumentet/dokumenten följer standard och validerar*?		
HTML- och CSS-dokumenterna är välstrukturerade och fristående		
Layouten är responsiv (minimumkravet är att den skalar om, allt innehåll ska vara tillgängligt såväl i bred som smal viewport).		
Det finns två olika menyutseenden (en navicon och en lista)		

*Examinerande lärare kommer att be er validera någon av projektets sidor som utvalt stickprov.

INDIVIDUELL DEL (G/VG)

Det finns...	Gruppmedlems personliga sida			
	#1	#2	#3	#4
... en bild och en utförligare presentation?				
... minst fem projekt listas (inlästa från JSON-fil)?				
... ett bildspel som man kan bläddra i (stega framåt och bakåt bland bilderna)?				
HTML/CSS-dokumentet följer standard, är strukturerade och validerar.				
Det finns minst två olika layouter (inte bara skalning)*. Media queries ska användas för att reagera på väsentlig förändring i storlek på viewport).				
Animerade skill bars presenteras				
Animering/transition triggad av händelse finns				
De inlästa projekten kan filtreras och sorteras i realtid				

*Det spelar ingen roll om flexbox eller gridlayout har använts så länge som det finns olika layouter (dvs. både flytande och fast är ok).

MUNTLLIG REDOVISNING

Den utvalda delen som ni kommer att behöva förklara kan vara en del av det gemensamma arbetet eller från **er** individuella del (ej från någon annan gruppmedlems individuella sida).

Gruppmedlemmens förklaring av den utvalda delen är:

	#1	#2	#3	#4
Begriplig och visar tydlig förståelse för koden och vad den gör (ja/nej).				

Examinerande lärare: _____