

Webbprogrammering

**(relevant screenshot från app tillåtet
här)**

**Namn N (Förnamn och EN bokstav från efternamn skriv EJ
personnummer här)**

1 Inlämningsuppgifter Scripting 2,0 hp

1.1 Sidnavigering / Single Page Application

Programkod som hanterar sidnavigering. Programkodssnuttar skall ha en monotyp-font och ha ett enkelt radavstånd. Kom ihåg att indentera programkodssnuttarna så de blir läsliga. Varje kodsnudd skall visa ett avgränsat koncept och hållas så kort som möjligt. Det går bra att använda ellipsis (...) för indikera att ovesäntlig kod klippts bort för att spara textutrymme (figur 2).

```
// Programkodssnuttar skall vara mörk text med vit bakgrund
// som skall gå att markera som text
// Programkodssnuttar får INTE vara en screenshot av
// kodeditor med svart bakgrund
// Programkodssnuttar får gärna med syntax-highlighting
// men det är inte nödvändigt
```

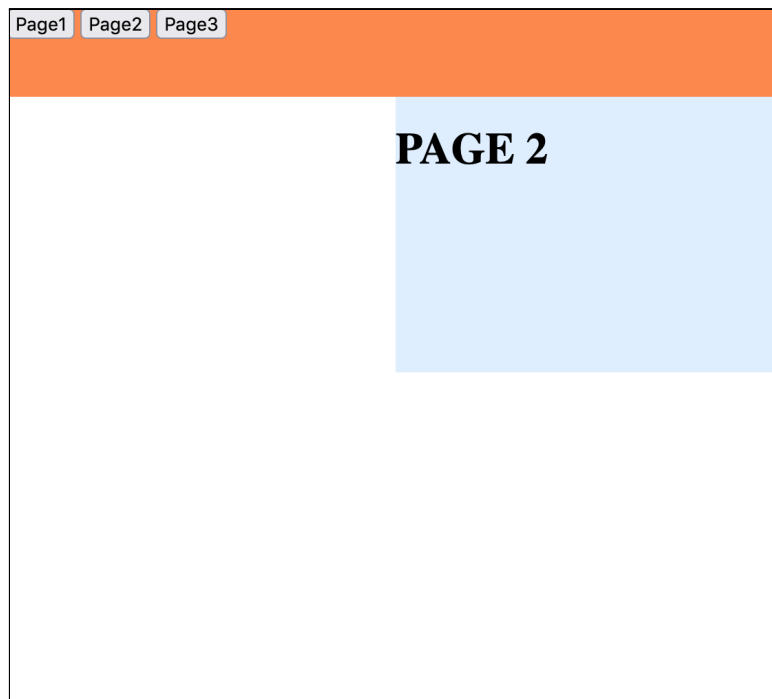
Figur 1 Programkod Sidnavigering

Mera diskussion kring annat stycke programkod som har med sidnavigering att göra.

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8"/>
  <title>React Example</title>
</head>
<body>
  ...
<div id="main-menu"></div>
  ...
</body>
</html>
```

Figur 2 Programkod Sidnavigering

Diskussion figur 3, eftersom figur 3 visar exempel på dropdown som sidnavigering är det lämpligt att diskutera användandet av dropdown här, innan figuren kommer i texten. Figurer skall ha en border så vi tydligt kan se var screenshot börjar och slutar.



Figur 3 Sidnavigering Med Dropdown för Page 2

1.2 Historikhantering

Eftersom vi kommer att navigera till olika sidor (screens) utan att göra sidnavigering till nya html-dokument så kommer vi vara tvungna att manuellt uppdatera historiken i webbläsaren. Om en användare t ex har besökt sök-sidan och sedan navigerar till bokningshistoriken så skall ett tryck på webbläsarens “tillbaka”-knapp förflytta användaren tillbaka till sök-sidan (inte till webbsidan som vi besökte innan vi navigerade till bokningssystemet)

1.3 Validering

På kund-registrerings-screen skall javascript användas för för validering, dvs att innehållet i input-element (input, select, textview, etc) kontrolleras (med handskriven med javascript) exempelvis för att garantera att nummer matas in i nummer-fält eller att email-adresser är korrekt formaterade. Vi fokuserar i kursen på att koppla events till inputs och hur man hanterar validering i Javascript. Det är tillåtet att använda form-tags men utan action (eftersom vi bygger single-page-applikationer) och med preventdefault

1.4 Input-hantering/ Animering med Toolkits / Javascript

Skall exempelvis ett datum matas in så skall det helst göras med någon form av date-picker och inte ett textfält, eller om en resurs skall väljas så är inte ett textfält

det mest optimala HTML-element för detta ändamål. Skall ett element döljas eller information visas upp så skall animering användas på några olika ställen detta kan implementeras med CSS Transitions/CSS Animations/Jquery. Exempelvis kan elementet långsamt bli genomskinligt eller en dropdown långsamt falla ner. Både Javascript och Toolkits skall användas för animering, minst en gång vardera.

1.5 Koppling till boknings-API

Dynamiskt innehåll så som till exempel användar-information eller bokningar skall laddas asynkront med jquerys AJAX-anrop eller HTML5 Fetch API. När du har kopplat dig mot Boknings-API så skall din mockup-html bytas ut mot HTML genererad med hjälp av javascript.

1.5.1 Kundregistrering

Programkod och screenshot på hur kundregistrering fungerar. Hur ser en lyckad registrering ut? Hur ser en misslyckad registrering ut?

1.5.2 Inloggning / Utloggning

Programkod och screenshot på hur inloggning / utloggning fungerar. Hur ser en lyckad inloggning ut? Hur ser en misslyckad inloggning ut?

1.5.3 Sökning på Resurs

Programkod och screenshot på hur sökning av resurs fungerar. Är det skillnad på 0, 1, flera träffar?

1.5.4 Genomföra Bokning

Programkod och screenshot på hur bokning av resurs fungerar. Hur vet jag som användare om min bokning är genomförd? Hur blir jag informerad som användare om min bokning misslyckades?

1.5.5 Visa Bokningshistorik

Programkod och screenshot på hur listning av bokningshistorik fungerar. Är det någon skillnad på inga bokningar, få bokningar, eller många bokningar?

1.6 Local Storage

Programkod och screenshot på hur local storage har använts.

1.7 VG Krav (stryk kapitlet om VG-kraven ej möts)

1.7.1 Generiska funktioner

1.7.2 Sorterad lista

Sorterad lista med eller utan sökning implementerad med Javascript eller JQuery.

1.7.3 Tangentbordsinteraktion

1.7.4 Icke Triviala Regular Expressions

1.7.5 Validering med flera olika felmeddelanden

2 Inbäddat Innehåll

2.1.1 Animerad Pixelgrafik från .png fil i canvas

Med hjälp av `drawImageData()` i canvas skall pixelgrafik importerad från en .png fil ritas ut i canvas. Grafiken skall vara rörlig (animerad) animerad men måste inte använda transformationer för denna animering.

2.1.2 Vektorgrafik från Importerad SVG i canvas

Vektorgrafik importerad från svg fil skall användas i canvas. Vektorgrafiken skall helst ritas ut i samma canvas som pixelgrafiken.

2.1.3 Transformation av vektorgrafik

Vektorgrafiken från SVG-filen skall animeras med hjälp av transformationer. Man skall välja minst två av de tre typerna av transformationer (Translation/Rotation/Skalning). Samma objekt måste inte exempelvis både translateras och skalas utan man kan exempelvis ha vissa vektorgrafik-objekt som translaterar och andra som skalas.

2.1.4 Klippning av grafik

På minst ett ställe skall handskrivna klippning av element eller grafik användas (Javascript och CSS `clip:rect` eller Animerad vektoriserad HTML5-klippning med `clip`-kommandot.). Klippning kan exempelvis implementera en karta som användaren kan navigera över eller som en scrollande display av produkter, eller någon annan mera innovativ form av automatiserad klippning. Klippningen skall vara rörlig eller de objekt som klipps skall vara rörliga, det vill säga, någon visuell förändring av klippningen måste ske över tid.

2.2 VG Krav Canvas (stryk kapitlet om VG-kraven ej möts)

2.2.1 Easing functions för att transformera objekt i canvas

2.2.2 Hierarkiska transformationer

2.2.3 Icke-trivial gradient som skiljer sig från de som givits som exempel

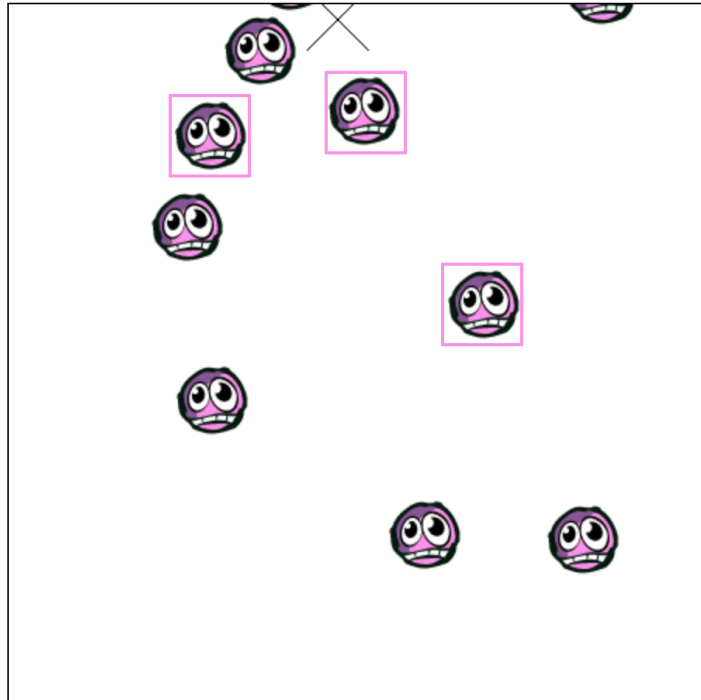
2.2.4 Icke-Trivial Clipping som skiljer sig från de som givits som exempel

2.3 Klickdetektering

Klickdetektering i någon form skall implementeras (clickmaps/if-satser/indelning i rutor).

Klickdetektering skall vara del i den grafisk interaktiviteten i webbapplikationen. Till exempel kan klickdetekttering användas för att markera olika resurser/objekt i bokningsapplikationen eller så kan klickdetekteringen vara del i ett enklare spel. Klickdetekteringen skall helst användas i interaktionen med hur man bokar resurserna i bokningssystemet exv vektorgrafik för biografstol för att boka biobiljett.

Screenshot med markering av olika klickbara objekt exv med röda fyrkanter (finns det väldigt många måste inte samtliga markeras).



Figur 4 Canvas med klickbara ytor markerade med rektangel

2.4 VG Krav Klickdetektering (stryk kapitlet om VG-kraven ej möts)

2.4.1 Multipla rörliga objekt

Klickdetektering som klarar flera olika rörliga objekt i samma canvas.

2.4.2 Klickdetektering som påverkar state

Klickdetektering skall helst användas i interatktionen mot användaren exv vektorgrafik för biografstol för att boka biobiljett, eller ett spel med interaktiv grafik som påverkar animering av när objekt kolliderar med varandra och/eller när man klickar i objekten.