



Kursnamn: *Webbframverk för JavaScript*
Klass: *Frontendutvecklare webbsäkerhet*
Termin: *VT 2024*

DOGBOOK

INLEDNING

Bakgrundsbeskrivning, frågeställning, avgränsning och mål	<p><u>Bakgrund:</u> I denna uppgift ska ni bygga en webbapplikation med React. Applikationen är tänkt som ett adminverktyg för hunddagispersonal att veta vilka hundar som finns inne på dagiset och vilka av dem som är kompisar. En användare kan lägga upp en profilbild, kort beskrivning, och en vänlista för en hund. Ni kommer använda tidigare kunskaper i backend utveckling som nya kunskaper i front-end utveckling.</p> <p><u>Mål:</u> Ett socialt nätverk för hundar</p> <p><u>Avgränsning:</u> Det är tillåtet att använda tredjeparts bibliotek/ramverk för denna uppgift. så länge React används för själva huvuduppgiften och JavaScript (ES6) används.</p>
Varför ska ni utföra detta arbete?	<p><u>Syftet:</u></p> <ul style="list-style-type: none">• Studenten ska självständigt lära sig tillämpa webbframverk i utvecklingen av applikationer• Studenten ska få utökad kunskap om Arkitekturmönster, exempelvis Model-View-Controller (MVC)• Studenten ska få färdigheter i att Utveckla applikationer genom att använda vanligt förekommande arkitekturmönster, ramverk, bibliotek och designmönster• Studenten ska få färdighet i att kombinera front-end med back-end
Vad ska ni leverera?	<p>Följande ska levereras:</p> <ol style="list-style-type: none">1. Samtliga projektfiler2. En README-fil med instruktioner för hur man startar applikationen och testar3. Video på genomgång av applikation

ER PROJEKTUPPGIFT

Vad ska ni göra?	Node & Webpack / create-react-app / vite
------------------	---

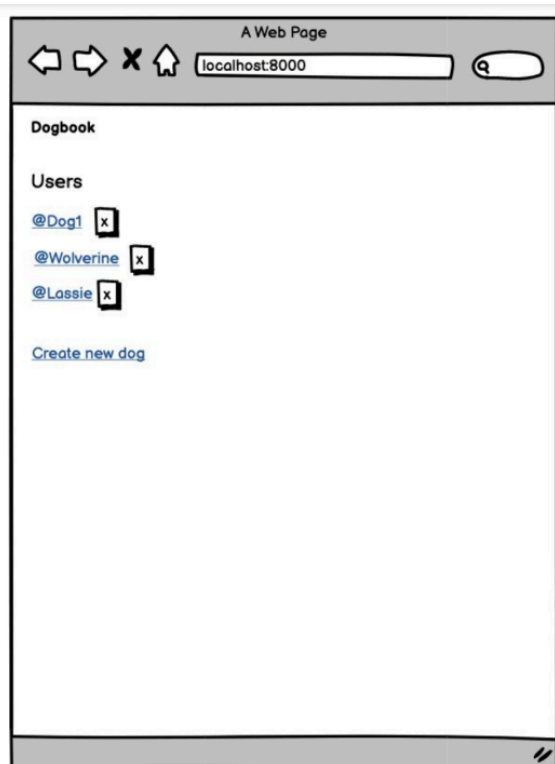
I den här uppgiften, kan ni välja att antingen skapa er egen byggkedja m.h.a Node, NPM och Webpack eller så använder ni create-react-app/vite (som vi har använt under lektionerna) om ni vill lägga mer fokus på React

Struktur

Första delen i uppgiften är att skapa en struktur för applikationen. I den här uppgiften behöver ni inte tänka på responsiv design och vi använder bara 1 HTML dokument (index.html) som kommer vara ingångspunkten för er applikation. Ni ska skapa en Single Page Application (SPA), så all HTML och CSS skapas dynamiskt, på samma sätt som ni har sett under lektionerna.

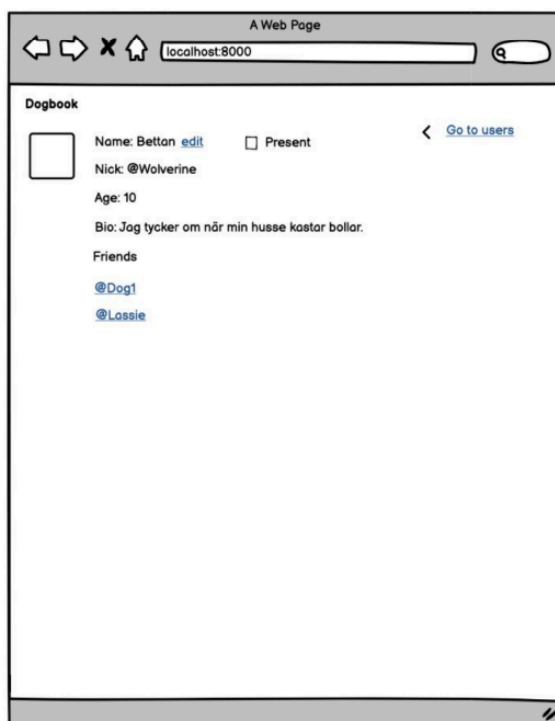
Designspecifikation

Följande design är bara ett förslag på hur det kan se ut, ni har fria händer att styla och inkludera mer information. Designen som presenteras kan betraktas som en miniminivå för vad som krävs.



Startsida

I det här fallet så har användaren kommit till startsidan. Här listas alla hundar som finns inne på gården. I denna listan har alla hundar blåa länkar men i er applikation ska de som är inne på gården markeras med grön text, och de övriga med röd text. Det finns länkar till hundarnas profiler (ex: @Wolverine), en knapp för att ta bort en hund (knappen med 'x') och en länk ("Create new dog") för att skapa en ny hund.

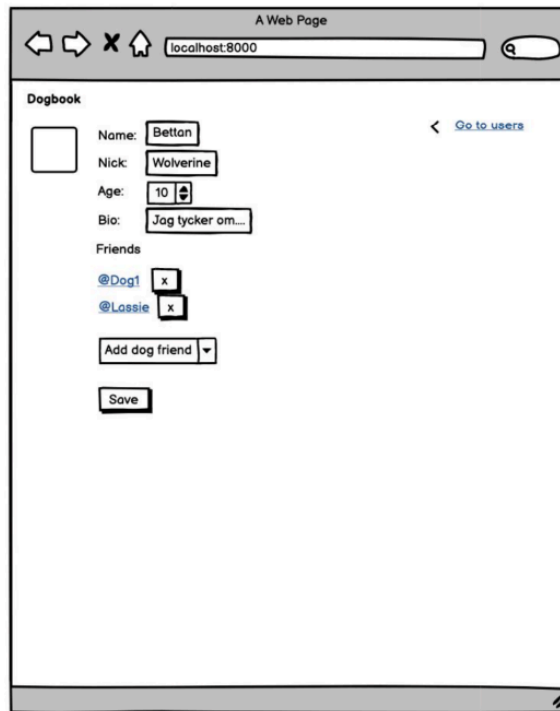


Profilsida

I det här fallet har användaren kommit till en hunds profilsida. Här ser användaren:

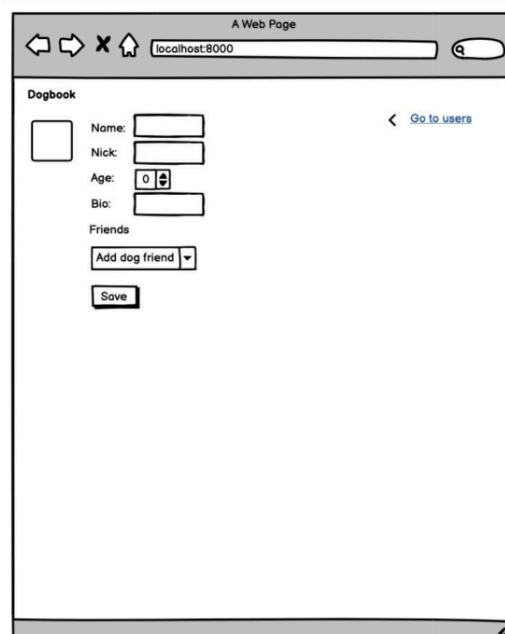
- Bild
- Edit länk
- Namn
- Smeknamn (påsidan)
- Alder
- Kort beskrivning
- Vänlista
- Checkbox för närvaro
- Länk tillbaka till startsidan

Bilden på hunden kommer att tas från ett API (<https://dog.ceo/api/breeds/image/random>) . Detta API kräver ingen nyckel och returnerar en godtycklig bild på en hund som tjänar våra syften att bara ha en statisk bild på en hund eftersom vi inte har några riktiga hundbilder



Editsida

I det här fallet har användaren valt att klicka på "Edit"-länken på profilsidan för att redigera uppgifter om hunden. Här kan användaren lägga till och ta bort hundar från profilens vänlista. Tänk på att när användaren tagit bort en hund från startsidan så ska detta också avspeglas i alla vänlistor som hunden befanns i, dvs de ska tas bort från alla hundars vänlistor.



Skapasida



	<p>I det här fallet har användaren klickat på "create new dog" på startsidan. Här fyller användaren in uppgifter för en ny hund.</p> <p>Spara data</p> <p>För att spara data använder ni ett Node API (förslagsvis med Express som ni kan sedan tidigare) och MongoDB som databas.</p> <p>JavaScript</p> <p>För att lyckas utforma applikationen med god struktur och modularitet så bör ni göra separationer av logik i applikationen med komponenter och funktioner. Ni kan använda antingen Klass komponenter eller Sida 5 av 7 Funktionella komponenter. Det ni bör tänka på är att strukturera applikationen så att det är enkelt för läsaren av koden att följa strukturen. Ni bör med fördel ha följande komponent struktur:</p> <ul style="list-style-type: none">• App - Huvudkomponent som visar den aktuella vyn• Start - Startsidea vyn• Profile - Profilsida vyn• Edit - Editsida vyn• Create - Skapasida vyn <p>Naturligtvis är detta bara ett förslag, men det ska vara enkelt att följa den logiska tråden, och huvudkomponenter som dessa ovan (eller de ni har valt), bör separeras till en egen fil.</p>
Hur ska ni lösa uppgiften?	<ol style="list-style-type: none">1. Skapa en mapp på din dator med namnet "inlämningsuppgift 3".2. Initiera ett projekt i mappen med create-react-app.3. Skapa följande filer: start.js, profile.js, edit.js och create.js.4. Lägg in React komponenter med hårdkodad html som representerar vyerna i spec5. Börja med att lägga in logik osv6. Använd komponenter där du ser att det är lämpligt så att du inte upprepar dig i onödan7. Använd console.log flitigt för att validera ditt arbete hela tiden under processen.

INLÄMNING OCH REDOVISNING

Inlämning	<p>Inlämning sker via LearnPoint datum 7/4 senast kl. 23:59</p> <ul style="list-style-type: none">• Inlämning av zip med projektfiler• README-fil med instruktioner för att köra och testa applikation• Video på genomgång av applikation (max 10 min)
-----------	---

BEDÖMNING OCH ÅTERKOPPLING

Bedömning sker mot följande betygskriterier:	<p>Betygskriterierna för Godkänd respektive Väl godkänd är:</p> <p>Godkänd</p> <ul style="list-style-type: none">• Studenten ska ha uppvisat kunskap om Komponent-Container tänk (arkitekturmönster) och hur komponenter samarbetar i React med props• Studenten ska ha uppvisat kunskap om hur React hanterar state och effekter via det designmönster som React har inbyggt i ramverket.• Studenten ska ha uppvisat kunskap om hur Reacts livscykel fungerar.
--	--



	<ul style="list-style-type: none">• Studenten ska ha uppvisat kunskap om testning och skrivit tester med Jest (bibliotek) för minst 3 av komponenterna• Studenten ska ha använt ett egenutvecklat backend-API med Node som används för att spara data. <p>Väl godkänd</p> <ul style="list-style-type: none">• Studenten ska ha uppvisat kunskap om hur funktionella komponenter och hooks fungerar i React• Studenten ska ha uppvisat kunskap i React Router (bibliotek) och med hjälp av detta bibliotek hantera navigation på tillbörligt sätt• Studenten ska ha uppvisat kunskap i Context API (designmönster) och använt detta på tillbörligt sätt för att hantera state i applikationen.• Studenten ska ha uppvisat kunskap om end-to-end testning och skrivit minst 1 end-to-end test för applikationen.
Återkoppling	I och med redovisningstillfället ges muntlig återkoppling till studenten. Studenten får skriftlig återkoppling via lärplattformen LearnPoint senast den 21/4