



INTERAKTIVA WEBBSYSTEM

D0037E

Interaktiva webbsystem, Lp4, V24

Av:

Mebaselassie Kidane Kebede

mebkeb-0@student.ltu.se

Kurs ansvarig:

Josef Hallberg

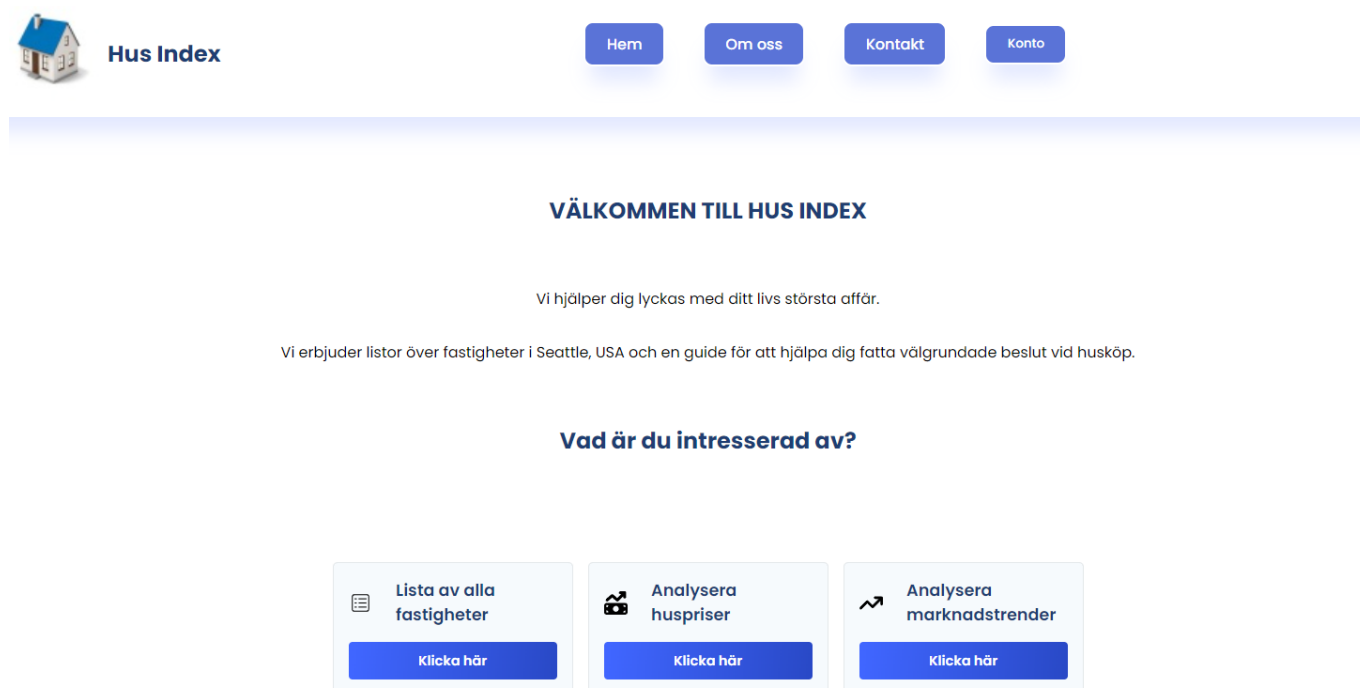
josef.hallberg@ltu.se

Innehåll

1	Design och Användbarhet	2
2	Förstå och använda grundläggande HTML och CSS för webbdesign	4
3	Backend integration och datahantering	5
4	Framtida förbättringar	9
5	Komma igång	9

1 Design och Användbarhet

- Webbplatsen är utvecklad med en tydlig förståelse för användarflödet och designval som är intuitiva för målgruppen nämligen, svensk-amerikaner i Seattle.
- Hierarkiskt organiserat innehåll och ren, minimalistisk design underlättar användarens fokus.
- Kontrasten mellan vit bakgrund och blå element gör det lätt att läsa texten och identifiera interaktiva element, placering av innehållet på lämpliga platser för att skapa en tydlig balanserad och tilltalande design.
- Användargränssnittet är väl strukturerat och följer F-formen, vilket är ett vanligt läsmönster på webben, där de viktigaste informationen presenteras från topp till botten och från vänster till höger för att underlätta för användaren att hitta informationen.



Figur 1: Header sektion

Mest utvalda

Populära fastigheter

<

>



\$250,000.00

Luxury Heights

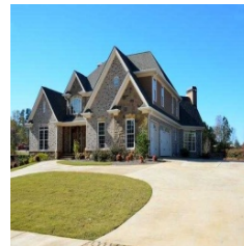
5 sovrum, 4 badrum



\$1,950,000.00

Vista View

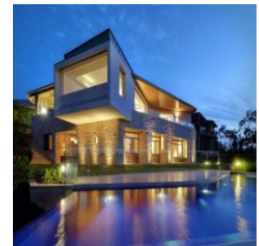
4 sovrum, 3 badrum



\$820,000.00

Sunrise Sanctuary

3 sovrum, 2 badrum



\$1,400,000.00

Golden Grove

4 sovrum, 3.5 badrum

Figur 2: Body sektion

Kontakt oss

Vi är alltid redo att hjälpa dig genom att ge dig den bästa tjänsten.

Telefon
0712 120 120

Ring oss

E-postadress
husindex@gmail.com

Mejla

Online kundtjänst
olinebot

Chatta

Mer information
Erbjudande broschyr

Ladda ned

Komma igång med Hus Index

Prenumerera på vårt veckobrev och hitta superattraktiva prisofferter från oss.
Helt gratis!

Fortsätta

Figur 3: Footer sektion

2 Förstå och använda grundläggande HTML och CSS för webbdesign

- Webbplatsen använder JavaScript-ramverket React Vite för att skapa interaktiva sidor och dynamiska listor som uppdateras automatiskt.
- En JSON-fil används för att lagra fastighetslistor och implementera sök- och filtreringsfunktioner utan att behöva ladda om sidan.
- Det finns ett exempel på en kodsnuitt från *slider.json* som innehåller en lista av fastigheter och hur den används i *Houses.jsx* under Coursel-mappen genom att enkelt kalla t.ex *property.name*, *property.price*, osv i rad 80 - 97.

```
{
  "name": "Luxury Heights",
  "price": "250,000.00",
  "detail": "5 sovrum, 4 badrum",
  "image": "./1.jpg"
},
{
  "name": "Vista View",
  "price": "1,950,000.00",
  "detail": "4 sovrum, 3 badrum",
  "image": "./2.jpg"
}
```

- Det används också mycket grundläggande HTML-element såsom `<a>`, `<button>`, `<input>`, osv. För att exemplifiera i *Header.jsx* rad 75 - 83, används både `<a>` och `<button>` i navigationsfältet för att skapa interaktivitet och navigationsfunktioner.
- Elementen `<a>` och `<button>` i detta fall erbjuder olika syfte och även har olika beteende. Tänken att använda elementet `<a>` var främst för att skapa hyperlänkar, som gör det möjligt för användare att navigera till en annan webbsida eller en specifik plats på samma sida, t.ex *Om oss* eller *Kontakt*.
- Däremot används elementet `<button>` för att skapa klickbara knappar på webbsidor det vill säga knappar som kan utlösa olika åtgärder, som att skicka ett formulär, öppna en modal dialogruta eller köra JavaScript-funktioner. I detta fall har just denna `<button>` integrerat med funktionen `<Register>` rad 88 i *Header.jsx* en formulär inläggningsfunktioner som skickas data senare till databasen.

```
<a className="button" href="">Hem</a>
```

```
<button className="button" onClick={() =>
  setButtonPopup(true)}> Konto
</button>
```

- Webbplatsen använder också fördefinierade CSS-attribut i App.css för att upprätthålla den ursprungliga designen och undvika upprepning av samma attribut i varje .jsx-fil. Detta minskar också antalet överflödiga attribut. Exempel på dessa inkluderar *.flexCenter*, *.primaryText*, *.flexStart*, *.button* och så vidare, vilka sedan har använts i olika sektioner.

```
<div className="flexCenter button">

<div className="innerWidth flexCenter2 paddings hero-container">

<span className = "primaryText">Populära fastigheter</span>

<span className="secondaryText">{property.detail}</span>
```

3 Backend integration och datahantering

- Webbplatsen integrerar backend-systemet, byggt på Node.js - *server.js*, med den befintliga frontend-applikationen.
- Genom att skapa en direkt anslutning till en MySQL-databas möjliggörs lagring och hämtning av data.

```
const db = mysql.createConnection({
  user: "root",
  host: "localhost",
  password: "6%guUk!4muBk^B",
  database: "interactive",
})

app.get('/lista_av_alla_fastigheter', (re, res) => {
  //const q = "SELECT * FROM houses"
  const q = "SELECT * FROM sales"
  db.query(q, (err, data) => {
    if(err) return res.json(err)
    return res.json(data)
  })
})
```

	PropertyID	PropertyName	PropertyType	Bathrooms	Bedrooms	SalePrice	ZipCode	NewConstruction	Image	Img
▶	1	Cozy Cottage	Multiplex	3	6	280000	98002	FALSKT	00FC30M41U9X	./0G0VJO0TKJAH.jpg
	2	Lakeview Manor	Single Family	3.75	4	1000000	98166	SANT	03HWK80EXDW9	NULL
	3	Sunny Villa	Single Family	1.75	4	745000	98166	FALSKT	03VX6PD8CRJ7	NULL
	4	Riverside Retreat	Single Family	3.75	5	425000	98168	FALSKT	082I5VIWXZ0E	NULL
	5	Urban Loft	Single Family	1.75	4	240000	98168	FALSKT	0G0VJO0TKJAH	NULL
	6	Mountain Hideaway	Townhouse	1.5	2	349900	98144	SANT	0N646IFYO6KP	NULL
	7	Beachfront Bungalow	Single Family	1.5	3	327500	98178	FALSKT	0SFS1LP5FSR2	NULL
	8	Cityscape Condo	Single Family	1.75	4	347000	98178	FALSKT	0UC5H3ZIO6W7	NULL
	9	Country Estate	Single Family	1	2	220400	98032	FALSKT	0UNMPHBUXYW7	NULL
	10	Luxury Penthouse	Multiplex	2	4	437500	98055	FALSKT	0V7UYVT5ECPY	NULL
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL


Figur 4: HouseListing data från MySQL

- Nedanför kodavsnittet demonstreras ett effektivt sätt att asynkront hämta data i React med hjälp av useEffect-hook i *HouseList.jsx* från en specificerad API-slutpunkt.
- Funktionen tar två parametrar: en funktion och en valfri uppsättning beroende variabler. I detta fall används en tom array [] för att säkerställa att effekten endast körs en gång efter den första renderingen.
- Därefter konverteras svaret från hämtningsförfrågan till JSON-format och bearbetas. Dessutom fångas eventuella fel upp och loggas till konsolen för felsökning, om några fel skulle uppstå under hämtningsprocessen.

```
useEffect(() => {
  fetch('http://localhost:8081/lista_av_alla_fastigheter')
    .then(response => response.json())
    .then(data => {
      setHouses(data);
      console.log(data);
    })
    .catch(error => console.error('Error fetching houses:', error));
}, []);
```

Lista av alla fastigheter

Återställ ▼

Fastigheter ID	Fastigheter namn	Fastigheter typ	Badrum	Sovrum	Försäljningspris	Postnummer	Nybyggd	Bild
1	Cozy Cottage	Multiplex	3	6	\$ 280000	98002	FALSKT	
2	Lakeview Manor	Single Family	3.75	4	\$ 1000000	98166	SANT	
3	Sunny Villa	Single Family	1.75	4	\$ 745000	98166	FALSKT	
4	Riverside Retreat	Single Family	3.75	5	\$ 425000	98168	FALSKT	
5	Urban Loft	Single Family	1.75	4	\$ 240000	98168	FALSKT	
6	Mountain Hideaway	Townhouse	1.5	2	\$ 349900	98144	SANT	
7	Beachfront Bungalow	Single Family	1.5	3	\$ 327500	98178	FALSKT	
8	Cityscape Condo	Single Family	1.75	4	\$ 347000	98178	FALSKT	
9	Country Estate	Single Family	1	2	\$ 220400	98032	FALSKT	
10	Luxury Penthouse	Multiplex	2	4	\$ 437500	98055	FALSKT	

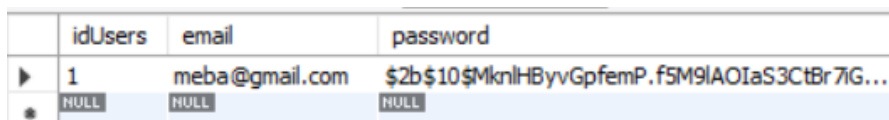
Figur 5: Lista av fastigheterna från databasen

- När datan har hämtats och visas i webbgränssnittet, kan denna funktion enkelt användas för att hämta rätt data från databasen genom att antingen använda fastighetens namn eller typ.
- Det är möjligt att använda både stora och små bokstäver eller postnummer för att söka efter önskad information.

```
const filteredHouses = sortedHouses.filter(house => {  
  return house.PropertyName.toLowerCase().includes(searchQuery.toLowerCase()) ||  
    house.PropertyType.toLowerCase().includes(searchQuery.toLowerCase()) ||  
    house.ZipCode.includes(searchQuery);  
});
```

- Vidare är det möjligt att filtrera datan efter olika kriterier såsom pris, antal badrum och sovrum, postnummer, eller om fastigheten är nybyggd eller inte, utan att behöva ladda om sidan. Detta åstadkoms genom att använda enkel sorteringsmetod i *HouseList.jsx* under Hero-mappen rad 28 - 46.
- Kodavsnittet nedan, hämtat från rad 61 i *server.js*, illustrerar en POST-funktion som tar emot indata från webbklienten när en användare registrerar sig med sin e-postadress och lösenord. Denna information skickas sedan till databasen och lagras där.

```
app.post("/register", (req, res) => {  
  const email = req.body.email;  
  const password = req.body.password;  
  
  bcrypt.hash(password, saltRounds, (err, hash) => {  
    if (err) {  
      console.log(err)  
    }  
    db.query (  
      "INSERT INTO users (email, password) VALUES (?, ?)",  
      [email, hash],  
      (err, result) => {  
        console.log(err); // Log any errors  
      });  
    });  
  });  
});
```



	idUsers	email	password
▶	1	meba@gmail.com	\$2b\$10\$MknlHByvGpfemP.f5M9IAOIaS3CtBr7IG...
•	NULL	NULL	NULL

Figur 6: User tabllen från MySQL

4 Framtida förbättringar

- Förslag inkluderar att förbättra bildkolumnen i MySQL-databasen för att säkerställa korrekt visning av fastigheter bilder i front-end-applikationen
- Behovet av att optimera responsiviteten och layouten för att anpassa till olika skärmlösningar samt införa ett nattläge för ökad användarvänlighet eftersom webbplatsen inte innehåller mycket text eller bilder som kan lindra dess intensitet.
- Slutligen behöver navigeringsfältet förbättras för att tydligare leda användarna till specifika sidor på webbplatsen.

5 Komma igång

För att starta, kör **npm run dev**, i *src/main/frontend* och om allt funkar då kommer det visa *http://localhost:xxxx/*

```
PS D:\DATATEKNIK\D0037E Interaktiva webbsystem\Hus Index\src\main\frontend> npm run dev

> frontend@0.0.0 dev
> vite

VITE v5.2.6 ready in 1873 ms
→ Local:   http://localhost:5173/
→ Network: use --host to expose
→ press h + enter to show help
```

Figur 7: Körning från frontend

För att starta, kör **node server.js** i *src/main/backend* och om allt funkar då kommer visas *Server has started..* meddelande på terminalen.

```
PS D:\DATATEKNIK\D0037E Interaktiva webbsystem\Hus Index\src\main\backend> node server.js
Server has started..
█
```

Figur 8: Körning från backend