

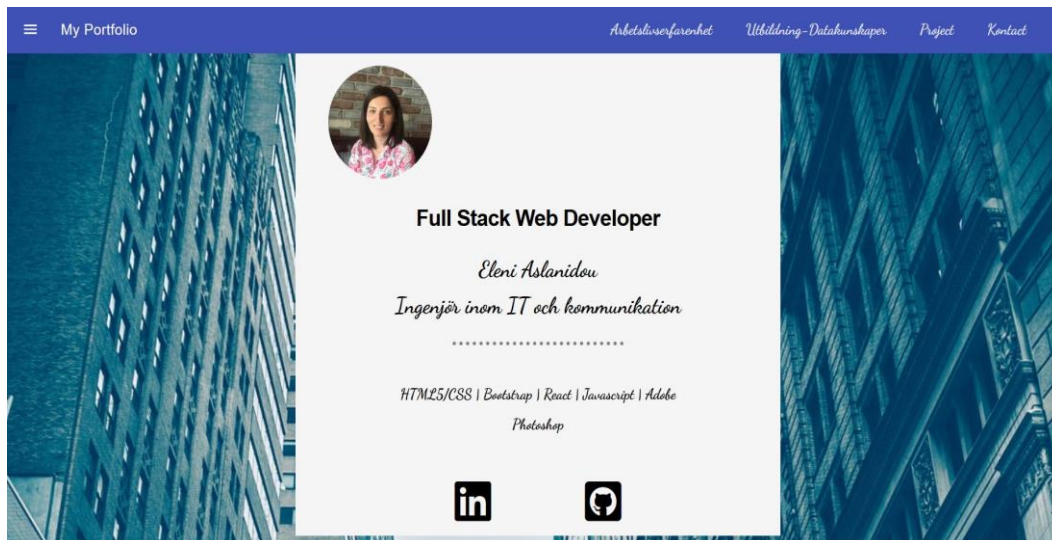


## Teoretisk redogörelse Individuell Projektuppgift 2 FE19 "Portföljsidan"



Länken av sidan → <http://myportfolio-eleni-aslanidou.surge.sh/>

Det nytt Projekt gäller om my portfoliowebbsite. Jag skapade det med hjälp av React. I början var det jättesvårt eftersom det är var nåt helt nytt för mig. Efter några veckor med hjälp av boken (the road to learn react) och många videor på Youtube skapade jag min portfolio sida.



Jag har valt den typ av style-struktur eftersom jag tror att det är intressant för nån som kommer att besöka den och naturligtvis passar bra för min personlighet.

### Index.html

In the body →

```
<noscript> You need to enable JavaScript to run this app. </noscript>
<div id="root"></div>
```

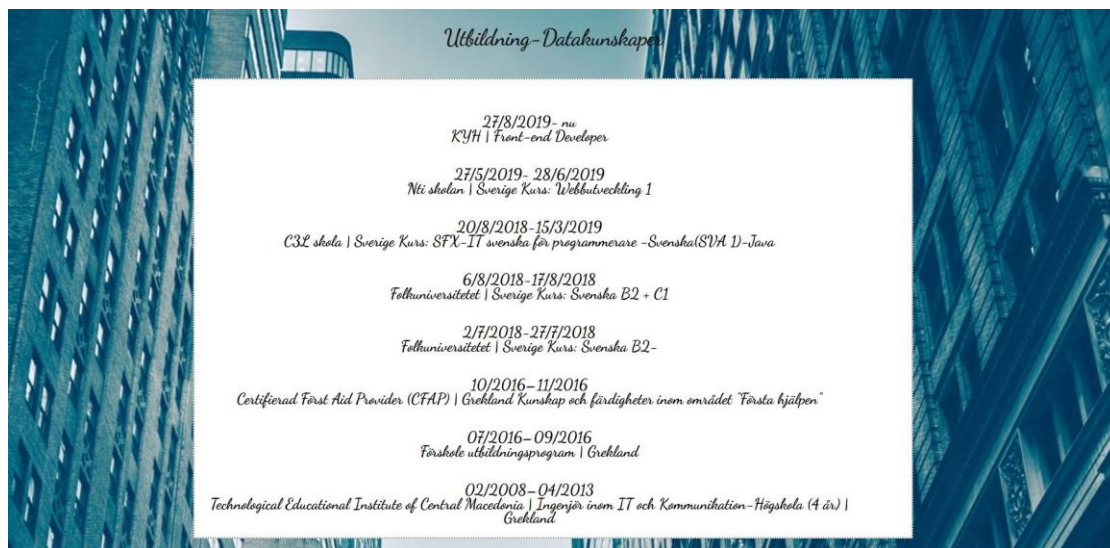
Sidan tala om för användaren att JavaScript krävs om den visas i en browser utan stöd för JavaScript, Sidan ska se ut som man tänkt er i Chrome.

Utan först sida (LandingPage.js), finns det fyra sidor till.

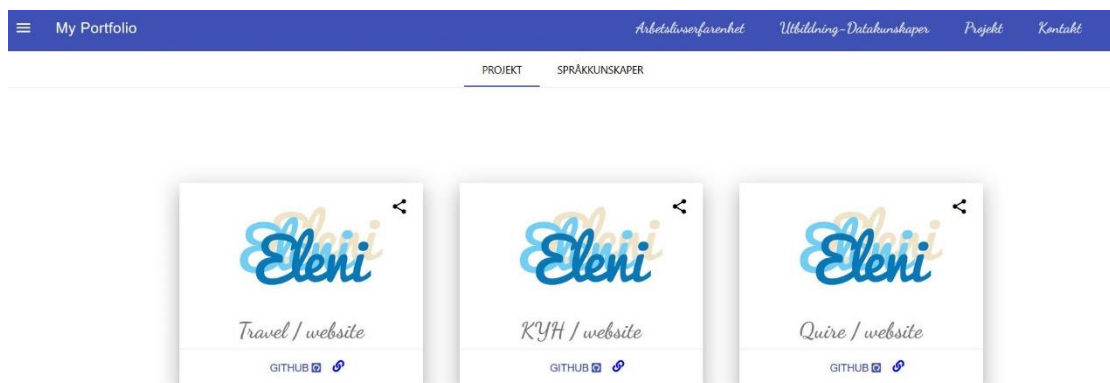
✓ Arbetslivserfarenhet.js



✓ Utbildning-Datakunskaper.js



✓ Projects.js



✓ Contact.js



problem:

- Jag hade problem med react struktur. **Lösning:** Med hjälp av boken allt har blivit lättare.

**Index.html**

In the body →

```
<noscript> You need to enable JavaScript to run this app. </noscript>
<div id="root"></div>
```

## Målen

Javascript, 1.1 Javascript, grundläggande syntax:

**JS1.1.1.**

```
class App extends Component {
  render(){
    return (
      <div className="App">
        <header className="App-header">
          <div className="demo-big-content">
            <Layout>
              <Header className='header-color' title={<Link style={{ textDecoration: 'none', color: 'white' }} to="/">My Portfolio</Link>} scroll>
                <Navigation>
                  <Link to="/Arbetslivserfarenhet" style={{fontSize: '20px', color: 'white', fontFamily: 'Dancing Script' }}>Ar-
betslivserfarenhet</Link>
                  <Link to="/UtbildningDatakunskaper" style={{font-
Size: '20px', color: 'white', fontFamily: 'Dancing Script' }}>Utbildning-
Datakunskaper</Link>
                  <Link to="/projects" style={{font-
Size: '20px', color: 'white', fontFamily: 'Dancing Script' }}>Projekt</Link>
                  <Link to="/contact" style={{font-
Size: '20px', color: 'white', fontFamily: 'Dancing Script' }}>Kontakt</Link>
                </Navigation>
              </Header>

              <Drawer title={<Link style={{textDecoration: 'none', color: 'blue', font-
Size: '25px', fontFamily: 'Dancing Script'}} to="/">Home</Link>} >
                <Navigation>
                  <Link to="/Arbetslivserfaren-
het" style={{color: 'black', fontSize: '25px', fontFamily: 'Danc-
ing Script'}}>Arbetslivserfarenhet</Link>
                  <Link to="/UtbildningDatakun-
skaper" style={{color: 'black', fontSize: '25px', fontFamily: 'Danc-
ing Script'}}>Utbildning-Datakunskaper</Link>
```

```

        <Link to="/projects" style={{color: 'black', font-
Size: '25px', fontFamily: 'Dancing Script'}}>Projekt</Link>
        <Link to="/contact" style={{color: 'black', font-
Size: '25px', fontFamily: 'Dancing Script'}}>Kontakt</Link>
      </Navigation>
    </Drawer>

```

Den koden ligger i sidan App.js. Det beskriver navigation bar och navigation som ligger i vänster av sidan. Den navigation bar innehåller de andra sidor som är: Arbetslivserfarenhet.js , UtbildningDatkunskaper.js, Projekts.js, Contact.js

Den annan navigation som ligger i vänster innehåller samma sidor och Home page. De alla ligger i en class som heter " App ". Med hjälp av "style" kunde jag byta färg, storlek och fontfamily.

Vidare ligger de alla i en specifik struktur:

```

<Layout>
<Header>
<Navigation>
<Link>
</Navigation>
</Header>
</Layout>

```

I Home page (Landingpage.js) :

```

import React, { Component } from 'react';
import { Grid, Cell, Layout } from 'react-mdl';

class Landing extends Component {
  render() {
    return(
      <div className="container" style={{width: '100%', margin: 'au-
to'}}>
        <Layout className="backgroundImage">
          <Grid className="landing-grid">
            <Cell col={15}>
              
            </Cell>
            <div className="banner-text">
              <h1>Full Stack Web Developer</h1>
              <h2>Eleni Aslanidou<br/>Ingenjör inom IT och kom-
munikation</h2>
              <hr/>
            </div>
          </Grid>
        </Layout>
      </div>
    );
  }
}

```

```

        <h3>HTML5/CSS | Bootstrap | React | Javas-
cript | Adobe Photoshop</h3>
        <div className="social-links">
            { /*LinkedIn*/ }
            <a href="https://www.linkedin.com/in/eleni-
aslanidou/" rel="noopener noreferrer" target="_blank">
                <i className="fab fa-linkedin-
square" aria-hidden="true" />
            </a>
            { /*GitHub*/ }
            <a href="https://github.com/eleni-
aslanidou89" rel="noopener noreferrer" target="_blank">
                <i className="fab fa-github-square" aria-
hidden="true" />
            </a>
        </div>
    </div>
</Grid>
</Layout>
</div>
    )
}
}

export default Landing;

```

I alla sidor började jag med **import** för att visa vad vill jag innehålla i min sida.

```

import React, { Component } from 'react';
import { Grid, Cell, Layout } from 'react-mdl';

```

Till exempel vilka Components finns det.

Och alltid slutade jag min kod med:

```

export default Landing;

```

**export default** och specifikt namn av sidan.

I den sidan skapade jag en class som heter "container". Den här classen innehåller några information av mig, en bild av mig och två länkar (Linkedin och Github).

Jag skapade den med samma struktur <Layout> <Grid> och två <Cell>. De alla sidor har specifik style som jag har skrivit i App.css

Javascript, 1.2 Javascriptsversioner och dess kompatibilitet:

**JS1.2.1.**

Tre javascripts versions är:

ECMAScript 2017

ECMAScript 2018

ECMAScript 5.1 (2011)

Javascript, 1.3 Hur Javascript körs i webbläsaren:

**JS1.3.1.**

Det finns flera exempel om Javascript motor. Motor består av två huvudkomponenter:

- Memory Heap : det är här minnesfördelningen sker
- Call Stack : det är här våra stapelramar och kod körs

Javascript, 1.4 Javascript-bibliotek:

**JS1.4.1.**

Exempel på JS-bibliotek: - JQuery, React.js, Backbone.js

I mitt projekt använde jag React.js

Javascript, 1.5 Javascript-ramverk:

**JS1.5.1. Ramverk**

- För:**
- Man har möjlighet att skapa lättare och snabbare projekt via färdiga kod.
  - Mest av kod är öppen källkod och gratis.
  - Användare känner sig säkert när de använder ramverk.
  - Det finns många tutorials som finns online.

Nackdelar:

- Det finns många ramverk som är inte så lätt att använda.
- Några av dem är inte gratis.

HTML&CSS, 1.5 Olika CSS-ramverk:

**HC1.5.1**

CSS frameworks är jätte viktigt för våra projekt. Jag kommer att nämna några exempel:

1. Bootstrap
2. Bulma
3. Bootstrap

## HTML&CSS, 1.6 SASS eller LESS:

### HC1.6.1

Sass och Less är också jätteviktiga för CSS.

De är viktiga eftersom vi kan spara tid, de är lätt att använda och mer organized.

Vidare koden är mer lätt att förstå. Dessutom är det lätt att konvertera bara om man skriver .scss or .less

## Javascript, 1.6 DOM:

### JS1.6.1. vad är DOMen?

DOMen är struktur av vår dokument(root,elements,text,attribute). En typ av representation av text.

HTML DOM tillhandahåller ett gränssnitt (API) för att gå igenom och modifiera noderna. Den innehåller metoder som getElementById eller removeChild. Vi använder vanligtvis JavaScript-språk för att arbeta med DOM.

Till exempel:     <html>

```

    <title>.....</title>
    <body>
        <h1> ... </h1>
        <p> ... </p>
        <a> ... </a>
    </body>
</html>
```

### JS1.6.2. Vad är en Virtual DOM? Ge exempel. För-, nackdelar?

Virtual DOM är en del av DOM. Det finns inte stor skillnad mellan DOM och Virtual DOM. När man har en HTML code och vill byta ut det till static React component, man kan bara byta HTML coden till render och class till className.

### JS1.7.1. Vad är AJAX?

AJAX är metoden för att utbyta data med en server och uppdatera delar av en webbsida - utan att ladda hela sidan.

### JS1.7.2. Vad används det till? För-, och nackdelar?

Fördelar:

- **Callbacks**-Ajax används för att utföra ett återuppringning gör en snabb rundresa till och från servern för att hämta eller spara data utan att posta hela sidan tillbaka till servern.
- **Making Asynchronous Calls**- Med AJAX kan man ringa asynchronous till en webbserver. Detta gör att klientens webbläsare kan undvika att vänta på att all data kommer fram innan användaren kan agera en gång till.
- **User-Friendly**- eftersom en sidbackback elimineras, Ajax-aktiverade applikationer alltid att vara mer lyhörda, snabbare och mer användarvänliga.

- **Increased Speed-** Användaren betygsätter en action för den action sparas i sin databas utan att vänta på att sidan ska uppdateras eller laddas om. Dessa movements sparas i sin databas utan att hela sidan skickas tillbaka till servern.

Javascript, 1.8 HTTP1.1 / 2.0:

**JS1.8.1.** Vad är skillnaderna mellan HTTP 1.1 och 2.0 och varför är 2.0 bättre?

Hyper Text Transfer Protocol (HTTP), det enkla är en typ av protokollet utgör grunden för World Wide Web.

HTTP1.1 var begränsat till att endast behandla en begäran per TCP-anslutning, vilket tvingade webbläsare att använda flera TCP-anslutningar för att behandla flera requests samtidigt.