

Steg för att skapa projektet:

- **Skapa projektet:**

Skapa en folder på datorn där du vill att projektet ska skapas

Öppna foldern i VS Code

Öppna terminalen i VS Code och kör:

```
npm create vite@latest my-app --template react-ts  
cd my-app  
npm install
```

- **Installera Bootstrap:**

Kör:

```
npm install bootstrap
```

- **Starta applikationen:**

I terminalen, kör:

```
npm run dev
```

Editera filen `index.html` i roten av projektet

- Vi vill ha en full-width-layout i det här projektet och lägger till lite css för detta
- Lägg till följande längst ner i `<head>`-taggen i filen `index.html`

```
<style>
  html,
  body,
  #root {
    width: 100%;
    height: 100%;
    margin: 0;
    padding: 0;
  }
</style>
```

Egen CSS-modul

Skapa en CSS-modul under src-katalogen: `App.module.css` med innehållet:

```
.fullSize {  
  width: 100%;  
  height: 100%;  
  margin: 0;  
  padding: 0;  
}
```

Importera Bootstrap & Använda CSS Modules

- Rensa innehållet i App-komponenten, filen `src/App.tsx`
- Importera Bootstrap och din CSS-modul

```
import "bootstrap/dist/css/bootstrap.min.css";
import styles from "./App.module.css";
//ta bort ev annan kod i App.tsx
const App = () => {
  return (
    <div className={styles.fullSize}>
      {/* Här kommer våra komponenter att ligga */}
    </div>
  );
};

export default App;
```

Övning 1: Greeting Component

Uppgift:

- Skapa mappen `components` under `src`
- Skapa en komponent med namnet `Greeting` i `src/components/` och som visar en centrerad hälsningsfras med Bootstrap-klasser.
- Filen kan heta `Greeting.tsx`
- Innehållet i `jsx` kan vara en `<div>` med en rubrik `<h3>` inuti, tex:

```
<div className="text-center">  
  <h3>Välkommen till Min App!</h3>  
</div>
```

Lägg till Greeting i App-komponenten

- Så här lägger du till din komponent i App

```
const App = () => {  
  return (  
    <div className={styles.fullSize}>  
      <Greeting />  
    </div>  
  );  
};
```

Kolla att din Greeting dyker upp i webbläsaren - uppdateringar ska ske automatiskt när du ändrar din kod

Dynamisk Greeting med Prop

Gör din Greeting-komponent dynamisk genom att låta den ta emot en hälsningstext som prop.

1. Öppna filen `src/components/Greeting.tsx`.
2. Modifiera komponenten så att den tar emot en prop med namnet `message` (av typen `string`) istället för att använda en hådkodad text. Använd en `type GreetingProps` som typ för props.
3. Ändra innehållet i `<h1>` så att det visar värdet av `message`.
4. Öppna filen `src/App.tsx` och uppdatera anropet till Greeting-komponenten så att du skickar in en hälsningsfras via prop `message`.

Se resultatet i webbläsaren

Implementera en Header-komponent med children

- Skapa en Header-komponent i filen `src/components/Header.tsx` med TypeScript.
- Skapa en typ `HeaderProps` där du specificerar attributet `children` av typen `ReactNode`. Detta gör att Header kan ta emot valfritt JSX-innehåll via props.
- Header-komponenten kan rendera ett `<header>` -element med ett `<div>` -element i.
- Inkludera `{children}` inuti `<div>` -elementet, typ så här:

```
//ex: några bootstrap-klasser som styling
<header className="bg-dark text-white py-2">
  <div className="text-center">{children}</div>
</header>
```


- Uppdatera därefter din App-komponent (i `src/App.tsx`). Importera både Header och Greeting. I App kan Header omsluta Greeting, så här:

```
<Header>  
  <Greeting message="Välkommen" />  
</Header>
```

Skapa en footer till applikationen

1. CompanyInfo-komponent:

- Skapa filen `src/components/CompanyInfo.tsx`.
- Skicka in props för `companyName` och `contactMail`, båda av typen `string`, använd en `type CompanyInfoProps`.
- jsx kan se ut så här ca:

```
<div>
  <span className="me-3">&copy; {companyName}</span>
  <span className="me-3">Kontakt: {contactMail}</span>
</div>
```

2. Footer-komponenten:

- Skapa en filen `src/components/Footer.tsx` .
- Använd ett `<footer>` -element för rendering
- Inom `<footer>` , lägg ett `<div>` -element för att organisera innehållet.
- I denna `<div>` , rendera prop `children` , det ska vara din CompanyInfo-komponent.
- Förslag på css-klasser:

```
<footer className="fixed-bottom bg-dark text-white py-2">  
  <div className="text-center">{children}</div>  
</footer>
```

3. App-komponenten:

- Öppna `src/App.tsx`.
- Importera både Footer och CompanyInfo.
- Rendera Footer med CompanyInfo som children, tex så här:

```
<Footer>  
  <CompanyInfo  
    companyName="Ditt Företag AB"  
    contactMail="kontakt@foretaget.se"  
  />  
</Footer>
```