

Miniprojekt 1 - Create React App

Grupp G



Först svara på frågorna:

Gillar du filmen Wild Wild West med Will Smith?

Ja ● Nej ●

Gillar du att vara i centrum?

Ja ● Nej ●

Har du sett alla Star Wars filmer?

Ja ● Nej ●

Type your name...

Done

Databas

ProfileData

```
components
  viewContainer
    # checkbox.css
    # fonts.css
    FormField.tsx
    Header.tsx
    LinkButton.tsx
    MainView.tsx
  TS ProfileData.ts
    ProfileImg.tsx
    ProfileView.tsx
    QuestionForm.tsx
    Title.tsx
    TodoList.tsx
    TodoListItem.tsx
  TS types.ts
    App.tsx
    Layout.tsx
```

ProfileData.ts

```
src > components > viewContainer > TS ProfileData.ts > ...
12
13  export interface ProfileData {
14    id: string
15    profileImg?: string
16    profileText?: string
17    headerImg?: string
18    alt: string
19    style: CSSProperties
20    titel: string
21    initialTodos: Array<Todo>
22  }
23
24  const profileData: ProfileData[] = [
25    {
26      id: 'mainView',
27
28      profileImg: require('....../img/maincircle.jpg'),
29      profileText: 'R U CHORE?',
30      headerImg: require('....../img/UnknownBG.jpg'),
31      alt: 'Unknown person',
32      style: {
33        fontSize: "1.2rem",
34        fontFamily: 'arial',
35        color: 'white',
36        textShadow: '0 0 3px #000000, 0 0 5px #000000',
37        height: '100vh',
38        backgroundImage: `url(${ mainviewbackground })`,
39        backgroundPosition: 'center',
40        backgroundSize: 'cover',
41        overflow: 'scroll'
42      },
43      titel: 'Svara på frågorna nedan',
44      initialTodos: []
45    },
46    {
47      id: 'steampunk',
48      profileImg: require('....../img/steamFace2.jpg'),
49      headerImg: require('....../img/kugghjul.png'),
50      alt: 'steampunkProfile',
51      style: {
52        height: "100vh",
53        fontSize: "1.2rem",
54        fontFamily: "luminari",
55        color: 'black',
56        textShadow: 'none',
57        backgroundImage: `url(${ steambackground })`,
58        backgroundPosition: 'center',
59        backgroundSize: 'cover',
60        overflow: 'scroll'
61      },
62      titel: ', dina goggles är nyputsade och lädret är brunt.',
63      initialTodos: [
64        { text:'Putsa Pistolerna', complete: false},
65        { text: 'Köpa Höghatt', complete: false},
66        { text: 'Laga fickuret', complete: false},
67      ]
68    }
69  ]
```

Components

Trädstrukturer idag
*= komponenter använder ProfileData



App

The screenshot shows a code editor interface with a sidebar on the left and a main editor area on the right.

File Explorer (Left):

- components
 - viewContainer
 - # checkbox.css
 - # fonts.css
 - FormField.tsx
 - Header.tsx
 - LinkButton.tsx
 - MainView.tsx
 - ProfileData.ts (TS)
 - ProfileImg.tsx
 - ProfileView.tsx
 - QuestionForm.tsx
 - Title.tsx
 - TodoList.tsx
 - TodoListItem.tsx
 - types.ts (TS)
 - App.tsx (highlighted with a red border)
 - Layout.tsx (highlighted with a yellow border)

Editor Area (Right):

File: App.tsx

Path: src > components > App.tsx

```
1 import React from "react";
2 import { BrowserRouter } from 'react-router-dom'
3 import Layout from "./Layout"
4
5 const App = () => {
6   return (
7     <BrowserRouter>
8       <Layout />
9     </BrowserRouter>
10    )
11  };
12
13 export default App;
14
```

Layout

The screenshot shows a code editor interface with a sidebar on the left and a main editor area on the right.

File Structure:

- components
 - viewContainer
 - # checkbox.css
 - # fonts.css
 - FormField.tsx
 - Header.tsx
 - LinkButton.tsx
 - MainView.tsx
 - ProfileData.ts
 - ProfileImg.tsx
 - ProfileView.tsx
 - QuestionForm.tsx
 - Title.tsx
 - TodoList.tsx
 - TodoListItem.tsx
 - types.ts
 - App.tsx
 - Layout.tsx

Editor Area:

The main editor area displays the content of `Layout.tsx`:

```
16
17     return (
18         <div style={ content }>
19             <div style={profileData[0].style}>
20                 <MainView profile={profile} name={name} onProfileSelected={handleProfileSelected} />
21             </div>
22         </div>
23     );
24 }

25
26 export default withRouter(Layout);

27
28 const content: CSSProperties = {
29     display: 'flex',
30     flexDirection: 'column',
31     height: '100vh',
32 }
33
34
35
```

A red box highlights the line of code that renders the `MainView` component. A red arrow points from the `MainView` entry in the sidebar to this highlighted line in the editor.

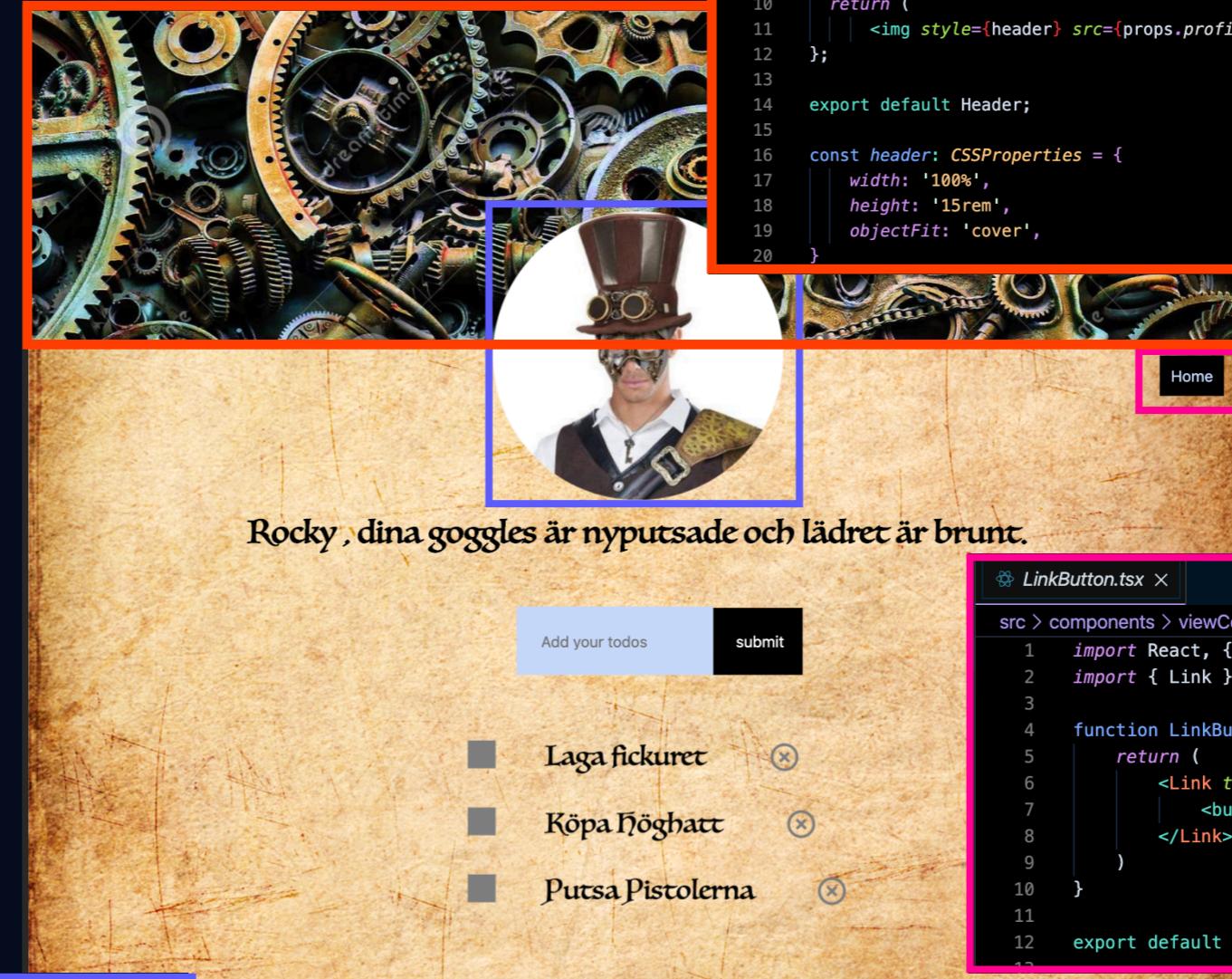
MainView

```
components
  viewContainer
  # checkbox.css
  # fonts.css
  FormField.tsx
  Header.tsx
  LinkButton.tsx
  MainView.tsx
  TS ProfileData.ts
  ProfileImg.tsx
  ProfileView.tsx
  QuestionForm.tsx
  Title.tsx
  TodoList.tsx
  TodoListItem.tsx
  TS types.ts
  App.tsx
  Layout.tsx
```

```
MainView.tsx
src > components > viewContainer > MainView.tsx > MainView
1 import React from "react";
2 import { Switch, Route } from "react-router-dom";
3 import QuestionForm from "./QuestionForm";
4 import ProfileView from "./ProfileView";
5 import profileData, { ProfileData } from "./ProfileData";
6 import ProfileImg from "./ProfileImg";
7 import Header from "./Header";
8 import LinkButton from "./LinkButton";
9 import { CSSProperties } from "@material-ui/core/styles/withStyles";
10
11 interface Props {
12   onProfileSelected: (profile: ProfileData, name: string) => void;
13   profile: ProfileData;
14   name: string;
15 }
16
17 const MainView = (props: Props) => {
18   return [
19     <Switch>
20       <Route exact path="/">
21         <div style={profileData[0].style}>
22           <Header profile={profileData[0]} />
23             <div style={style}>
24               <ProfileImg profile={profileData[0]} />
25             </div>
26             <QuestionForm onSubmit={props.onProfileSelected} />
27           </div>
28         </Route>
29
30         <Route path="/profile-view">
31           <div style={props.profile.style}>
32             <Header profile={props.profile} />
33               <div style={style}>
34                 <ProfileImg profile={props.profile} />
35               </div>
36               <LinkButton />
37               <ProfileView name={props.name} profile={props.profile} />
38             </div>
39           </Route>
40
41         <Route exact path="/">
42           <div style={profileData[0].style}>
43             <Header profile={profileData[0]} />
44               <div style={style}>
45                 <ProfileImg profile={profileData[0]} />
46               </div>
47               <QuestionForm onSubmit={props.onProfileSelected} />
48             </div>
49           </Route>
50
51           /* //ett sista alternativ, om ingen av dessa ska väljas. lägg in vad som ska visas här */
52         </Switch>
53     ];
54   };
55
56   const style: CSSProperties = {
57     position: "relative"
58   }
59
60   export default MainView;
```

components

- viewContainer
- # checkbox.css
- # fonts.css
- FormField.tsx
- Header.tsx
- LinkButton.tsx
- MainView.tsx
- TS ProfileData.ts
- ProfileImg.tsx
- ProfileView.tsx
- QuestionForm.tsx
- Title.tsx
- TodoList.tsx
- TodoListItem.tsx
- TS types.ts



```
ProfileImg.tsx
src > components > (alias) interface ProfileData > ProfileImg
1 import React from 'react';
2 import { ProfileData } from './ProfileData';
3
4 interface Props {
5   profile: ProfileData;
6 }
7
8 function ProfileImg(props: Props) {
9   return 

10     ![{props.profile.alt}]({props.profile.profileImg});
11     

<{props.profile.profileText}>


12

;
13 }
14
15 export default ProfileImg;
```

Header.tsx

```
src > components > viewContainer > Header.tsx > ...
1 import React, { CSSProperties } from 'react';
2 import { ProfileData } from './ProfileData';
3
4 interface Props {
5   profile: ProfileData;
6 }
7
8 //gets props from profile data
9 function Header(props: Props) {
10   return (
11     ![{props.profile.alt}]({props.profile.headerImg});
12   );
13 }
14
15 export default Header;
16
17 const header: CSSProperties = {
18   width: '100%',
19   height: '15rem',
20   objectFit: 'cover',
21 }
```

LinkButton.tsx

```
src > components > viewContainer > LinkButton.tsx > [x] showButton
1 import React, { CSSProperties } from 'react';
2 import { Link } from 'react-router-dom';
3
4 function LinkButton() {
5   return (
6     
7       >Home</button>
8     
9   );
10 }
11
12 export default LinkButton;
```

QuestionForm

components

viewContainer

checkbox.css

fonts.css

FormField.tsx

Header.tsx

LinkButton.tsx

MainView.tsx

TS ProfileData.ts

ProfileImg.tsx

ProfileView.tsx

QuestionForm.tsx

Title.tsx

TodoList.tsx

TodoListItem.tsx

TS types.ts

App.tsx

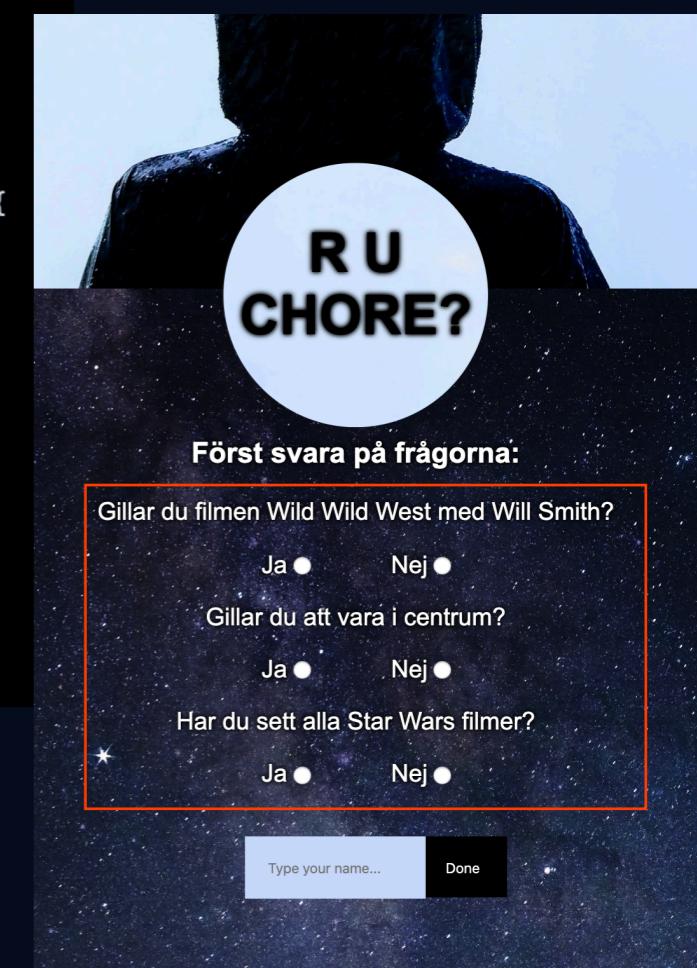
Layout.tsx

```
class QuestionForm extends React.Component<Props, State> {
  constructor(props: Props) {
    super(props);

    this.state = {
      name: "",
      shouldRedirect: false
    };
  }

  handleOnSubmit = (e: FormEvent) => {
    // stoppa form elemtet att uppdatera sidan automatiskt
    e.preventDefault();

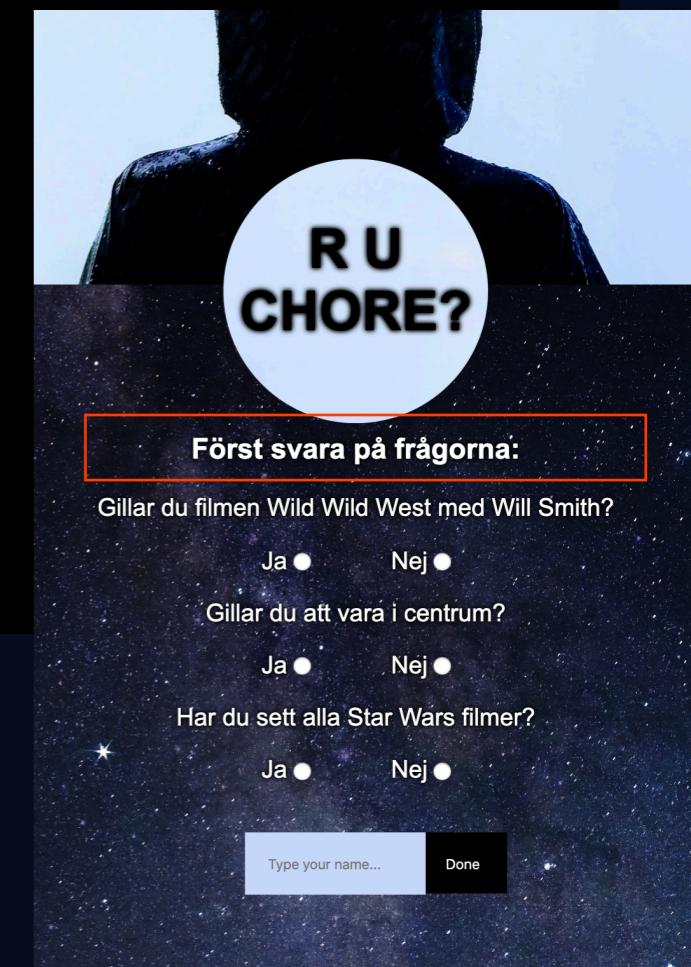
    if (
      this.state.question1 === true &&
      this.state.question2 === false &&
      this.state.question3 === false
    ) {
      this.props.onSubmit(profileData[1], this.state.name);
      // hämta steampunk från databasen spara det i state.
      //skicka state till layout och profileview.
    }
    if (this.state.question2 === true) {
      this.props.onSubmit(profileData[2], this.state.name);
      // hämta zlatan från databasen spara det i state.
      //skicka state till layout och profileview.
    }
    if (this.state.question2 === false && this.state.question3 === true) {
      this.props.onSubmit(profileData[3], this.state.name);
      // hämta nerd från databasen spara det i state.
      //skicka state till layout och profileview.
    } else if (
      this.state.question1 === false &&
      this.state.question2 === false &&
      this.state.question3 === false
    ) {
      //detta händer om man svarar nej på allt.
      this.props.onSubmit(profileData[4], this.state.name);
    }
  }
}
```



Title

src > components > viewContainer > Title.tsx > titleProps

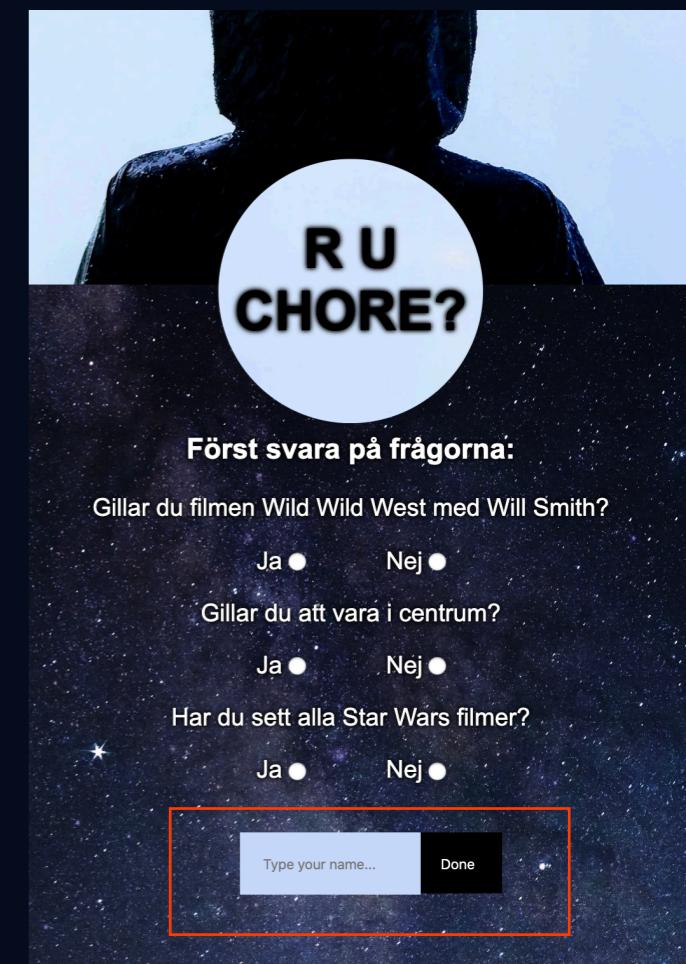
```
1 import React, { CSSProperties } from "react";
2
3 export interface titleProps {
4   titleContent: string;
5   name?: string;
6 }
7
8 export class Title extends React.Component<titleProps> {
9   render() {
10     return <h3 style={style}>{this.props.name} {this.props.titleContent}</h3>;
11   }
12 }
13
14 export default Title;
15
16 const style: CSSProperties = {
17   marginTop: "7rem",
18   textAlign: "center",
19   width: "100%"
20 };
21
```



FormField

```
components
  viewContainer
  # checkbox.css
  # fonts.css
  FormField.tsx
  Header.tsx
  LinkButton.tsx
  MainView.tsx
  TS ProfileData.ts
  ProfileImg.tsx
  ProfileView.tsx
  QuestionForm.tsx
  Title.tsx
  TodoList.tsx
  TodoListItem.tsx
  TS types.ts
  App.tsx
  Layout.tsx
```

```
src > components > viewContainer > FormField.tsx > inputField > backgroundColor
1 import React, { CSSProperties } from "react";
2
3 interface Props {
4   onChange?: (name: string) => void;
5   onSubmit?: (submit: string) => void;
6   defaultValue?: string;
7   placeHolder: string;
8   buttonText: string;
9 }
10
11 interface State {
12   value: string;
13 }
14
15 export class FormField extends React.Component<Props, State> {
16   constructor(props: Props) {
17     super(props);
18     this.state = {
19       value: this.props.defaultValue ? this.props.defaultValue : ""
20     };
21   }
22
23   private updateValue = (event: React.ChangeEvent<HTMLInputElement>) => {
24     if (this.props.onChange) {
25       this.props.onChange(event.target.value);
26     }
27     this.setState({
28       value: event.target.value
29     });
30   }
31
32   private clearInputField = () => {
33     if (this.props.onSubmit) {
34       this.props.onSubmit(this.state.value);
35     }
36     this.setState({
37       value: ""
38     });
39   }
40
41   render() {
42     return (
43       <div style={footerContainer}>
44         <input
45           style={inputField}
46           placeholder={this.props.placeHolder}
47           value={this.state.value}
48           type="text"
49           onChange={this.updateValue}
50         />
51         <button style={mainButton} onClick={this.clearInputField}>
52           {this.props.buttonText}
53         </button>
54       </div>
55     );
56   }
57 }
58
59 export default FormField;
```



ProfileView

components

viewContainer

checkbox.css

fonts.css

FormField.tsx

Header.tsx

LinkButton.tsx

MainView.tsx

TS ProfileData.ts

ProfileImg.tsx

ProfileView.tsx

QuestionForm.tsx

Title.tsx

TodoList.tsx

TodoListItem.tsx

TodoList.tsx

src > components > viewContainer > TodoList.tsx > TodoListProps > handleTodoRemove

```
1 import React, { CSSProperties } from "react";
2 import TodoList from "./TodoList";
3 import { Todo, ToggleTodo } from "./types";
4
5 interface TodoListProps {
6   todos: Array<Todo>;
7   toggleTodo: ToggleTodo;
8   handleTodoRemove: (text: string) => void;
9 }
10
11 const TodoList: React.FC<TodoListProps> = ({ todos, toggleTodo, handleTodoRemove }) => {
12   return (
13     <div style={{display: "flex", justifyContent: "center"}>
14       <ul style={centerTodos}>
15         {todos.map(todo => {
16           return (
17             <TodoListItem
18               key={todo.text}
19               todo={todo}
20               toggleTodo={toggleTodo}
21               handleTodoRemove={handleTodoRemove}
22             />
23           )
24         })
25       </ul>
26     </div>
27   );
28 };
29
30
31 export default TodoList
32 const centerTodos: CSSProperties = {
33   display: "flex",
34   justifyContent: "center",
35   alignItems: "flexStart",
36   flexDirection: "column-reverse",
37   listStyleType: "none"
38};
```

ProfileView.tsx

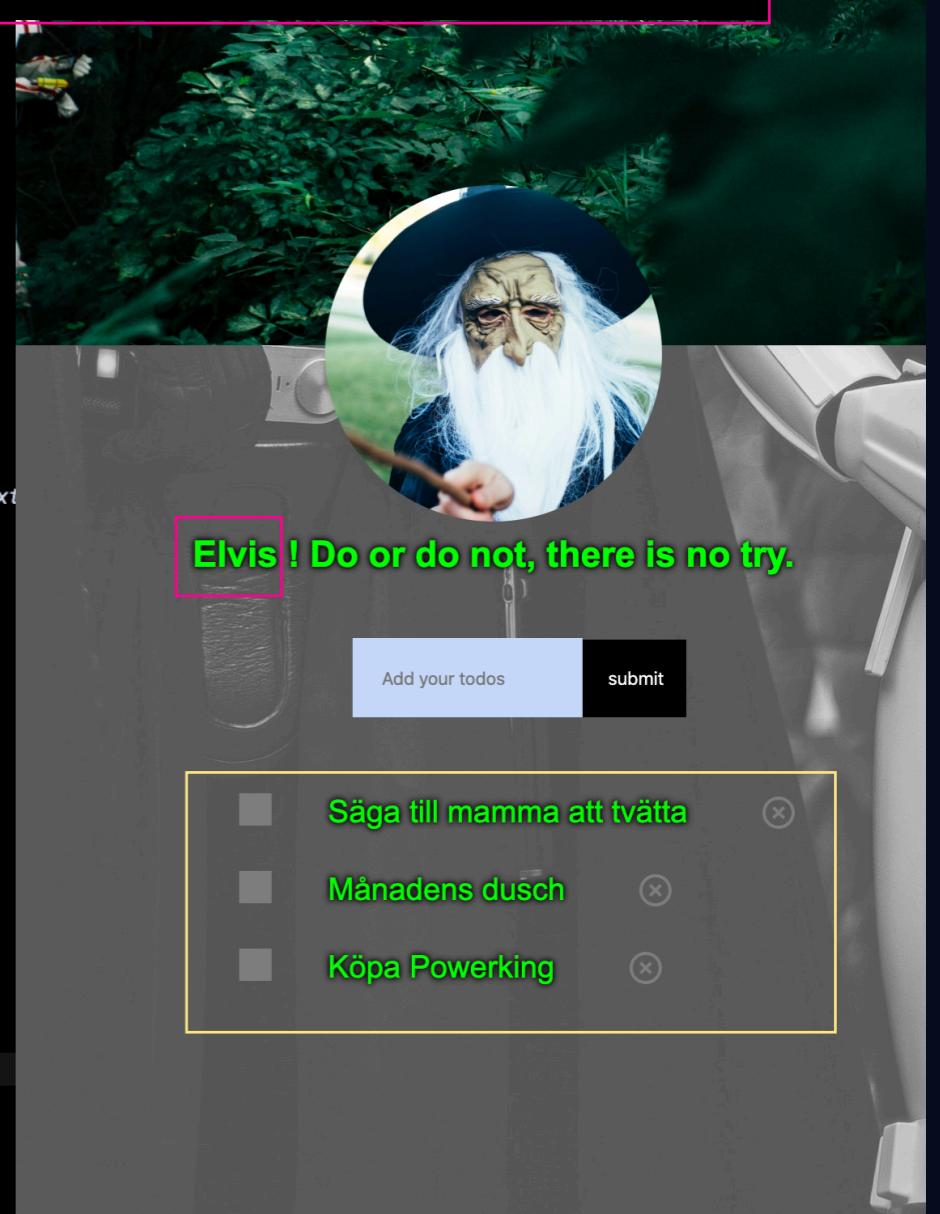
src > components > viewContainer > ProfileView.tsx > ProfileView

```
1 import React, { useState } from "react";
2 import TodoList from "./TodoList";
3 import { ToggleTodo, AddTodo, Todo } from "./types";
4 import FormField from "./FormField";
5 import { ProfileData } from "./ProfileData";
6 import Title from "./Title";
7
8
9 interface Props {
10   profile: ProfileData
11   name: string
12 }
13
14 const ProfileView: React.FC<Props> = props => {
15   const [todos, setTodos] = useState(props.profile.initialTodos);
16
17   const toggleTodo: ToggleTodo = selectedTodo => {
18     const newTodos = todos.map(todo => {
19       if (todo === selectedTodo) {
20         return {
21           ...todo,
22           complete: !todo.complete
23         };
24       }
25       return todo;
26     });
27     setTodos(newTodos);
28   };
29
30   const addTodo: AddTodo = newTodo => {
31     newTodo.trim() !== "" &&
32       setTodos([...todos, { text: newTodo, complete: false }]);
33   };
34
35   function handleTodoRemove(text: string) {
36     const newTodos: Todo[] = todos.filter((todo: Todo) => todo.text !== text);
37     setTodos(newTodos);
38   }
39
40
41   return (
42     <React.Fragment>
43       <Title name={props.name} titleContent={props.profile.title} />
44       <FormField
45         onSubmit={addTodo}
46         buttonText="submit"
47         placeHolder="Add your todos"
48       />
49
50       <TodoList
51         todos={todos}
52         toggleTodo={toggleTodo}
53         handleTodoRemove={handleTodoRemove}
54       />
55
56     </React.Fragment>
57   );
58
59   export default ProfileView;
```

Title.tsx

src > components > viewContainer > Title.tsx > titleProps

```
1 import React, { CSSProperties } from "react";
2
3 export interface titleProps {
4   titleContent: string;
5   name?: string;
6 }
7
8 export class Title extends React.Component<titleProps> {
9   render() {
10     return <h3 style={style}>{this.props.name} {this.props.titleContent}</h3>;
11   }
12 }
13
14 export default Title;
15
16 const style: CSSProperties = {
17   marginTop: "7rem",
18   textAlign: "center",
19   width: "100%"
20 };
21
```



```

✓ components
  ✓ viewContainer
    # checkbox.css
    # fonts.css
    ⚡ FormField.tsx
    ⚡ Header.tsx
    ⚡ LinkButton.tsx
    ⚡ MainView.tsx
  TS ProfileData.ts
    ⚡ ProfileImg.tsx
    ⚡ ProfileView.tsx
    ⚡ QuestionForm.tsx
    ⚡ Title.tsx
    ⚡ TodoList.tsx
    ⚡ TodoListItemIcon.tsx
  TS types.ts
    ⚡ App.tsx
    ⚡ Layout.tsx

```

TS ProfileData.ts

```

src > components > viewContainer > TS_ProfileData.ts
1 import (alias) const zlatanbackground: string
2 import zlatanbackground from './img/zlatan_bg.png'
3 import nerdbackground from './img/darth.jpg'
4 import mainviewbackground from './img/space.jpg'
5 import './fonts.css'

{
  id: 'zlatan',
  profileImg: require('./img/world-cup-trophy.png'),
  headerImg: require('./img/zlatan_header.png'),
  alt: 'zlatanProfile',
  style: {
    height: "100vh",
    fontSize: "1.2rem",
    fontFamily: 'Raleway, sans-serif',
    color: "rgb(254, 204, 0)",
    textShadow: 'none',
    backgroundImage: `url(${ zlatanbackground })`,
    backgroundPosition: 'center',
    backgroundSize: 'cover',
    overflow: 'scroll'
  },
  title: ' do not need a trophy to tell yourself that you are the best.',
  initialTodos: [
    { text:'Du har inga todos, du ger ut todos', complete: false},
  ]
}

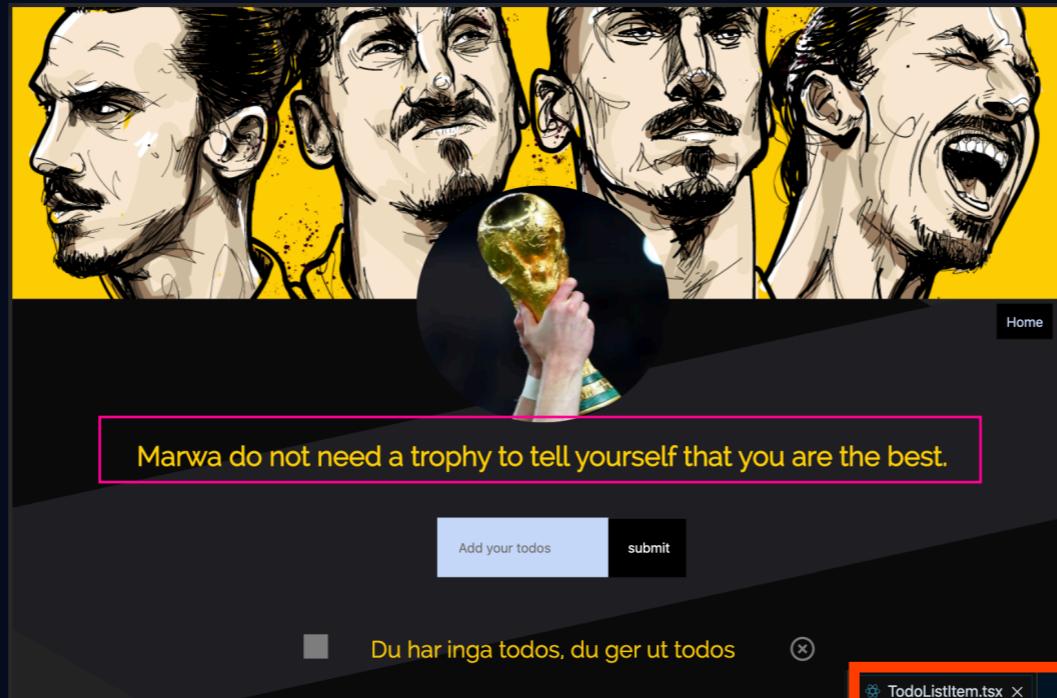
```

fonts.css

```

src > components > viewContainer > # fonts.css
1 @import url('https://fonts.googleapis.com/css?family=Lobster&display=swap');
2 /* Lobster, cursive */
3 @import url("https://fonts.googleapis.com/css?family=Raleway&display=swap");
4 /* Raleway, sans-serif */
5
6

```



TodoListItemIcon.tsx

```

src > components > viewContainer > TodoListItemIcon.tsx > TodoListItemIcon
1 import React, { CSSProperties } from "react";
2 import { Todo, ToggleTodo } from "./types";
3 import './checkbox.css'
4 import IconButton from '@material-ui/core/IconButton';
5 // import DeleteIcon from '@material-ui/icons/Delete';
6 import HighlightOffIcon from '@material-ui/icons/HighlightOff';

interface TodoListItemIconProps {
  todo: Todo;
  toggleTodo: ToggleTodo;
  handleTodoRemove: (text: string) => void;
}

const TodoListItemIcon: React.FC<TodoListItemIconProps> = ({ todo, toggleTodo, handleTodoRemove }) => {
  const completed: CSSProperties = {
    textDecoration: "line-through",
    color: "#444444",
    fontStyle: "italic"
  };
  return (
    <li>
      <label className="checkbox" style={todo.complete ? completed : undefined}>
        <input
          type="checkbox"
          checked={todo.complete}
          onChange={() => toggleTodo(todo)} />
        <span>{todo.text}</span>
        <IconButton style={ icons } aria-label="HighlightOff"
          onClick={() => {
            handleTodoRemove(todo.text);
          }}>
          <HighlightOffIcon />
        </IconButton>
        {/* <button onClick={() => handleTodoRemove(todo.text)}>X</button> */}
      </label>
    </li>
  );
}

const icons: CSSProperties = {
  color: 'grey'
}

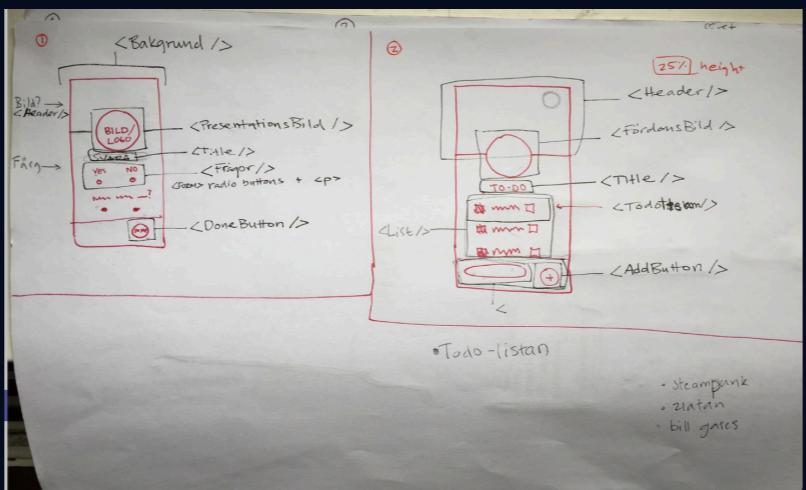
```

Planering och genomförande

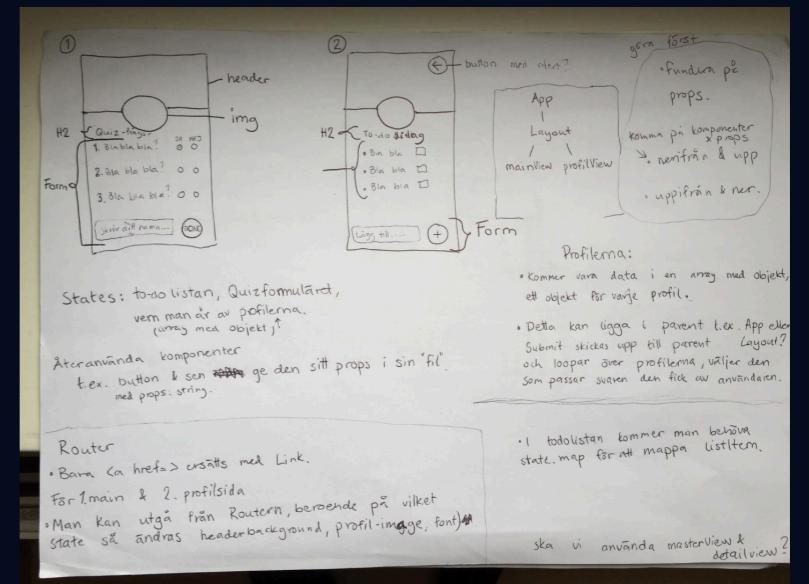
Första trädstrukturen

```
<App>
  <Layout>
    <Header>
    <ProfileImg>
    <Title>
    <MainView>
      <QuestionForm>
        <form>
        <p>
        <radiobutton>
      <ProfileView>
        <Todo>
          <TodoList>
          <TodoItem>
          <TodoForm>
        <Footer>
          <Input>
          <Submit>
```

Första mockup



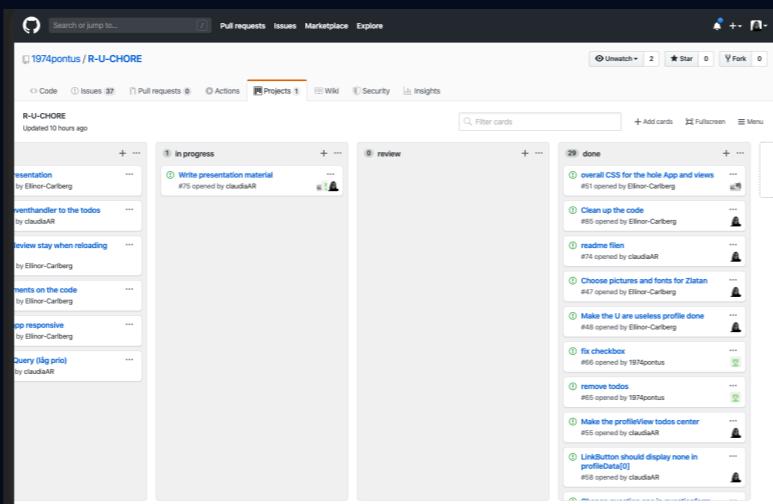
Mer detaljer



Github project med issues

Trädstrukturen idag
*= komponenten
använder ProfileData

```
App
  Layout *
    Mainview *
      Header *
      ProfileImg *
      QuestionForm *
        Title
        FormField
    Header *
      ProfileImg *
      LinkButton
      ProfileView *
        Title
        FormField
        Todolist
          TodolistItem
```



R-U Chore?

R-U Chore is basically a Todo-app that gives you a custom design depending on how you answer the questions. R-U Chore will scan your personality and refer you to your true self that comes with some custom chores. Please enjoy and have fun!

❤️ the R-U Chore team

**Vänta med att börja koda.
Strukturera först.**

**Många moduler som behöver
installeras via npm**

**Lättare att dela upp arbetet ett
med komponentbaserat flöde**

**Svårt att hitta ett bra sätt att
jobba med css på ett bra sätt.**



Tack och hej ,computer says no

Add your todos

submit



~~Say NO~~

