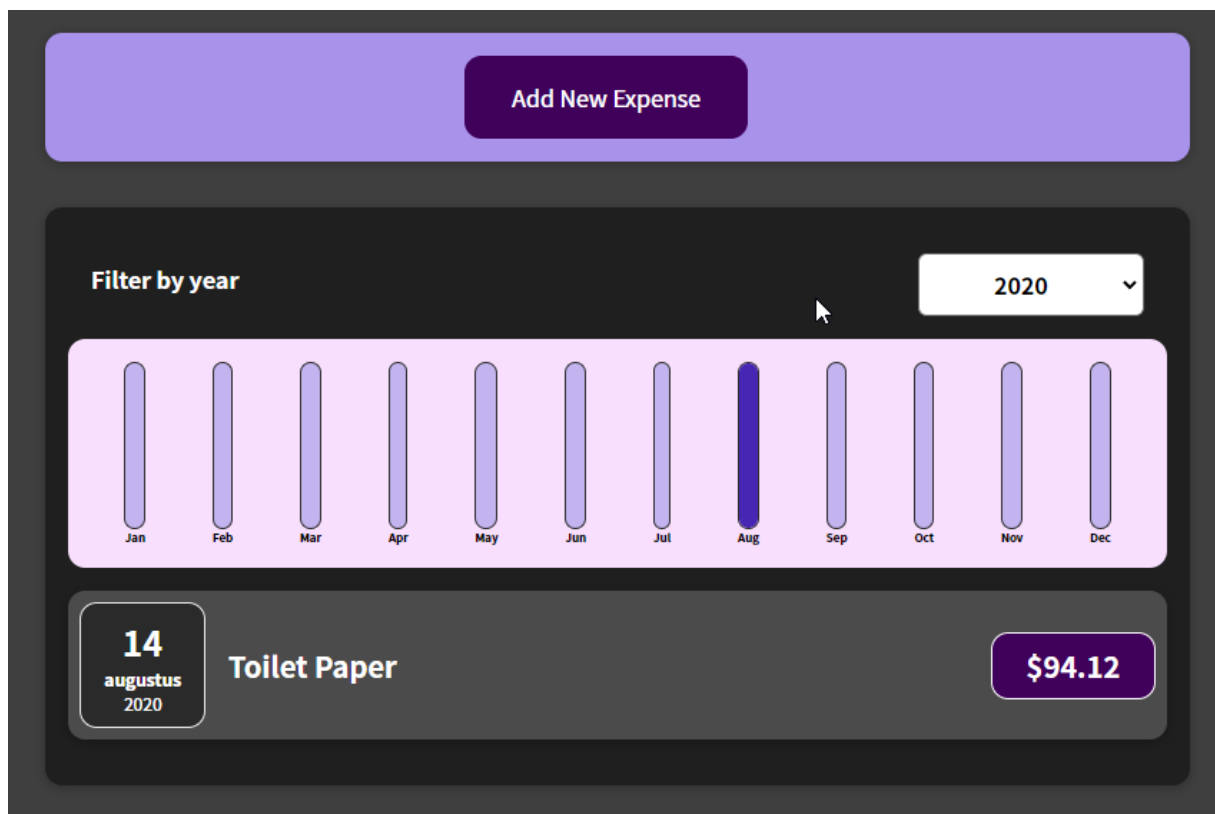


H3 react-complete-guide



```
// Van de array met 12 objecten maak je een nieuwe array met 12 waarden
const dataPointValues = props.dataPoints.map((dataPoint) => dataPoint.value);

const totalMaximum = Math.max(...dataPointValues);

barFillHeight = Math.round((props.value / props.maxValue) * 100) + "%";

const month = props.datum.toLocaleString("nl-NL", { month: "long" });
const day = props.datum.toLocaleString("nl-NL", { day: "2-digit" });
const year = props.datum.getFullYear();

Expenses, ExpenseForm, NewExpense useState

<select value={props.selected} onChange={dropdownChangeHandler}>

const dropdownChangeHandler = (event) => {
  console.log(event.target.value);

id: Math.random().toString()

// Aangezien filteredYear een string is, moet je toString gebruiken
expense.date.getFullYear().toString() === filteredYear;
```

```

// props.expenses is een array dus gebruik je of en niet in
for (const expense of props.expenses) {
  const expenseMonth = expense.date.getMonth(); // starting at 0 => January
=> 0
  chartDataPoints[expenseMonth].value += expense.amount;
}

const expenseData = {
  ...enteredExpenseData,
  id: Math.random().toString()
};

setExpenses((prevExpenses) => {
  return [expense, ...prevExpenses];
});

```

H6 styling-react

Course Goal

Add Goal

Do all exercises!

Finish the course!

```
npm install --save styled-components
```

```

if (event.target.value.trim().length > 0) {
  setIsValid(true);
}

```

```
className={` ${classes["form-control"]} ${!isValid && classes.invalid}` }
```

```
<form onSubmit={formSubmitHandler}>
```

```

<input type="text" onChange={goalInputChangeHandler} />

// The unshift() method adds one or more elements to the beginning of an array
and returns the new length of the array.
setCourseGoals((prevGoals) => {
  console.table("addGoalHandler (prevGoals): " + prevGoals);
  console.log("addGoalHandler (enteredText): " + enteredText);
  const updatedGoals = [...prevGoals];
  updatedGoals.unshift({ text: enteredText, id: Math.random().toString()
});
  console.table("addGoalHandler (updatedGoals): " + updatedGoals);
  return updatedGoals;
});

setCourseGoals((prevGoals) => {
  console.table("deleteItemHandler (prevGoals): " + prevGoals);
  console.log("Te verwijderen goalId: " + goalId);
  const updatedGoals = prevGoals.filter((goal) => goal.id !== goalId);
  console.table("deleteItemHandler (updatedGoals): " + updatedGoals);
  return updatedGoals;
});

className={` ${classes["form-control"]} ${!isValid && classes.invalid}`}

let content = (
  <p style={{ textAlign: "center" }}>No goals found. Maybe add one?</p>
);

```

H7 debugging-react

Course Goal

Add Goal

Do all exercises!

Finish the course!

```
if (enteredValue.trim().length === 0) {
  setIsValid(false);
  return;
}

const [courseGoals, setCourseGoals] = useState([
  { text: "Do all exercises!", id: "g1" },
  { text: "Finish the course!", id: "g2" },
]);

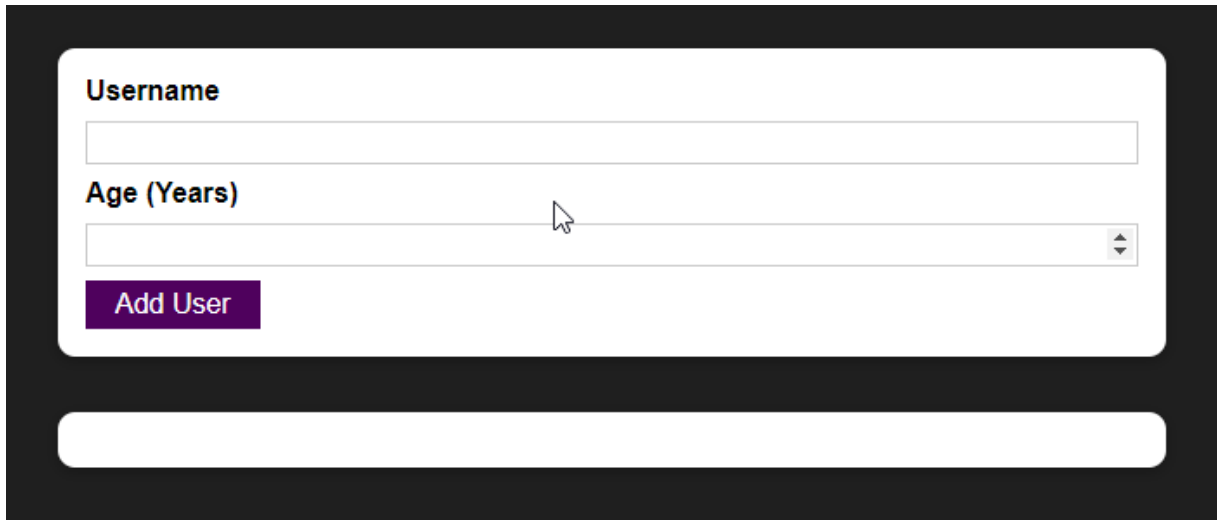
fetch("https://www.marchouben.nl/php/cursussen-post.php", {
  method: "POST",
  body: JSON.stringify({ cursus: enteredText }),
  headers: {
    //"Content-Type": "application/json",
    "Content-Type": "text/plain", // To prevent preflight, change
application/json into text/plain
  },
});

return (
  <li className="goal-item" onClick={deleteHandler}>
    {props.children}
  </li>
);

<p style={{ textAlign: "center" }}>No goals found. Maybe add one?</p>
```

```
className={` ${styles['form-control']} ${!isValid && styles.invalid}` }
```

H8 reusable-card-button



```
import classes from "./Button.module.css";

type={props.type || "button"}

<div className={` ${classes.card} ${props.className}` }>

// onClick prop is available on every default HTML component

// htmlFor i.p.v. for (className i.p.v. class)
// event.preventDefault() voor een submit om te voorkomen dat een request
wordt verstuurd
// return statement ends the function execution
// enteredAge is a string, not a number : +enteredAge zorgt ervoor dat het een
number wordt

    if (enteredUsername.trim().length === 0 || enteredAge.trim().length ===
0){
        // Dankzij setError wordt opnieuw een render uitgevoerd, waardoor error
het object met de title en message bevat
        setError({
            title: 'Invalid input',
            message: 'Please enter a valid name and age (non-empty values).'
        });
        return;
    }

<Card className={classes.users}>
  <ul>
    {props.users.map((user) => (
      <li key={user.id}>
```

```

        {user.name} ({user.age} years old)
      </li>
    )})
  </ul>
</Card>

const AddUserHandler = (uName, uAge) => {
  setUsersList((prevUsersList) => {
    return [...prevUsersList, { name: uName, age: uAge, id:
Math.random().toString() }]);
  });
};

```

// Hier wordt de functie uitgevoerd, dus met ()

```

{!isVoornaamValid && (
  <p style={{ textAlign: "center", color: "red" }}>
    Vul een voornaam in
  </p>
)}

```

H9 fragments-portals-refs

```

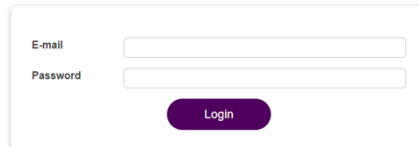
// import React is alleen nodig als je JSX gebruikt
// props.children is alles tussen open en sluit tag

// ReactDOM.createPortal verwacht een JSX element, het tweede argument is waar
het terecht moet komen
// Use Portal to move components HTML content somewhere else
{ReactDOM.createPortal(
  <Backdrop onConfirm={props.onConfirm} />,
  document.getElementById("backdrop-root") // Zie index.html (public)
)}

```

H10 side-effects-reducers-context

A Typical Page



A typical login form with two input fields labeled 'E-mail' and 'Password', and a 'Login' button.

```
import React, {useContext} from 'react';
const AuthCtx = useContext(AuthContext);
<Button onClick={AuthCtx.onLogout}>Logout</Button>

const emailReducer = (state, action) => {
  if (action.type === 'USER_INPUT') {
    return { value: action.val, isValid: action.val.includes('@') };
  }
  if (action.type === 'INPUT_BLUR') {
    return { value: state.value, isValid: state.value.includes('@') };
  }
  return { value: '', isValid: false };
};

const [emailState, dispatchEmail] = useReducer(emailReducer, {
  value: '',
  isValid: null,
});

const { isValid: emailIsValid } = emailState;

const emailChangeHandler = (event) => {
  dispatchEmail({ type: 'USER_INPUT', val: event.target.value });
};

const validateEmailHandler = () => {
  dispatchEmail({ type: 'INPUT_BLUR' });
};

const submitHandler = (event) => {
  event.preventDefault();
  if (formIsValid){
    AuthCtx.onLogin(emailState.value, passwordState.value);
  } else if (!emailIsValid){
    emailInputRef.current.focus();
  }
};
```

```

    } else {
      passwordInputRef.current.focus();
    }
  };

  <form onSubmit={handleSubmit}>
    <Input
      ref={emailInputRef}
      id="email"
      label="E-mail"
      type="email"
      isValid={emailIsValid}
      value={emailState.value}
      onChange={emailChangeHandler}
      onBlur={validateEmailHandler}

<header className={classes['main-header']}>

className={` ${classes.button} ${props.className}`}

import React, { useRef, useImperativeHandle } from "react";

const Input = React.forwardRef((props, ref) => {
  const inputRef = useRef();

  // Your own method
  const activate = () => {
    inputRef.current.focus();
  };

  // focus is de externe naam, activate de interne.
  useImperativeHandle(ref, () => {
    return {
      focus: activate,
    };
  });

  return (
    <div
      className={` ${classes.control} ${
        props.isValid === false ? classes.invalid : ""
      }`}
    >

const AuthContext = React.createContext({
  isLoggedIn: false,
  onLogout: () => {},
  inLogin: (email, password) => {}
});

```

```

export const AuthContextProvider = (props) => {
  const [isLoggedIn, setIsLoggedIn] = useState(false);

  useEffect(() => {
    const storedUserLoggedInInformation = localStorage.getItem("isLoggedIn");

    if (storedUserLoggedInInformation === "1") {
      setIsLoggedIn(true);
    }
  }, []);

  const loginHandler = (email, password) => {
    localStorage.setItem("isLoggedIn", "1");
    setIsLoggedIn(true);
  };

  const logoutHandler = () => {
    localStorage.removeItem("isLoggedIn");
    setIsLoggedIn(false);
  };

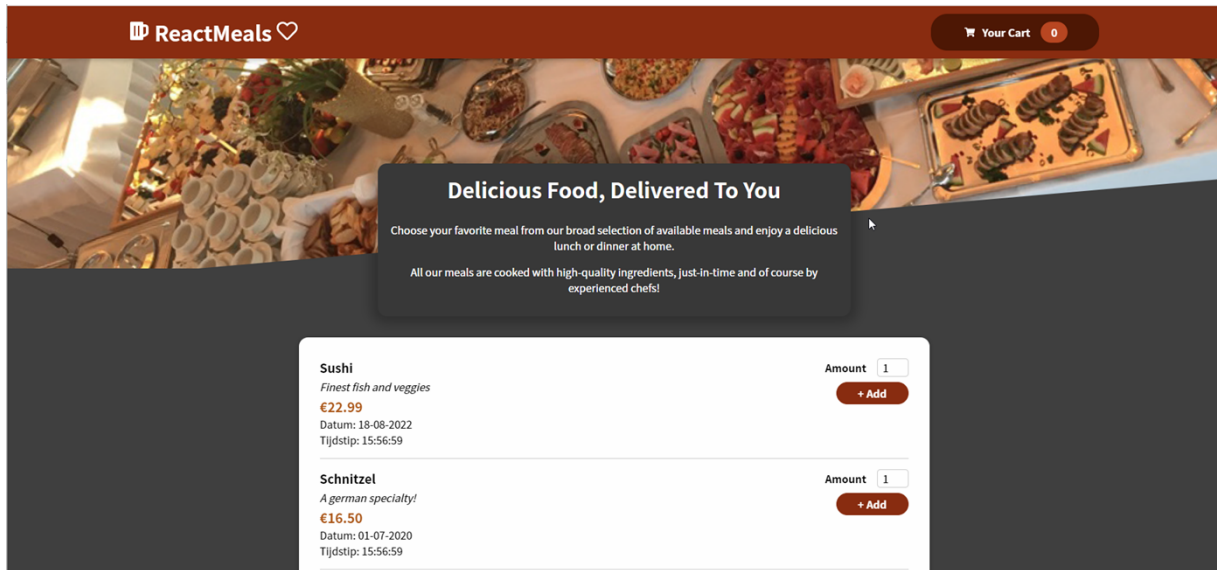
  return (
    <AuthContext.Provider value={{isLoggedIn: isLoggedIn, onLogout:
logoutHandler, onLogin: loginHandler}}>
      {props.children}
    </AuthContext.Provider>
  );
};

import AuthContext from './store/auth-context';
const ctx = useContext(AuthContext);
{!ctx.isLoggedIn && <Login />}

root.render(<AuthContextProvider><App /></AuthContextProvider>);

```

H11 food-order-app



H12 optimization-techniques

Hi there!

Allow Toggling

Toggle Paragraph!

My List

1

3

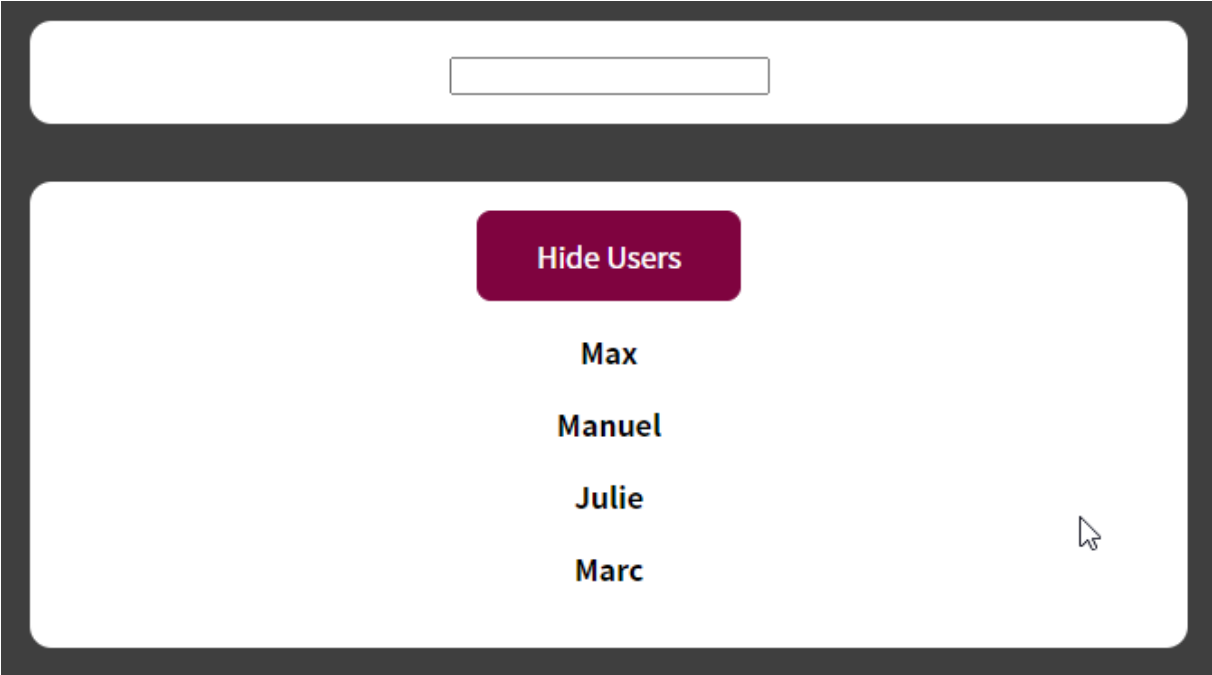
5

9

10

Change List Title

H13 class-based



H14 http-requests

Title

Opening Text

Release Date

Add Movie

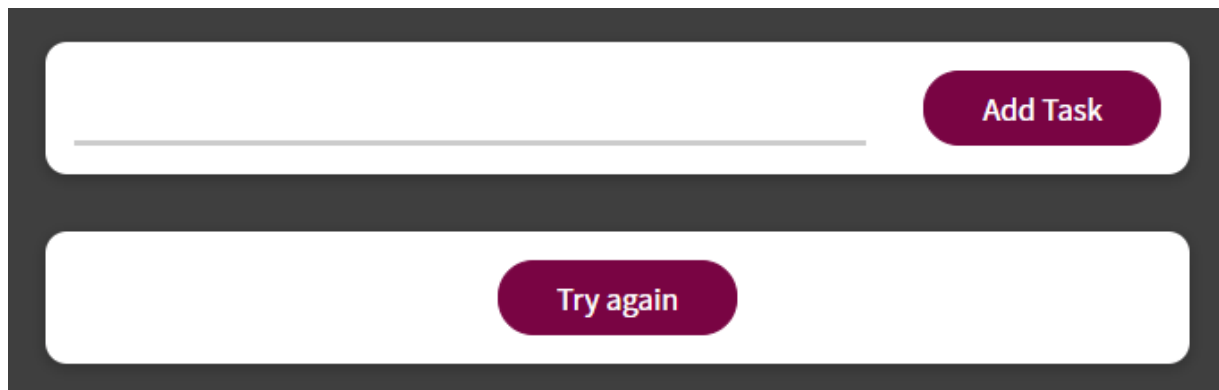
Fetch Movies

Something went wrong: 401

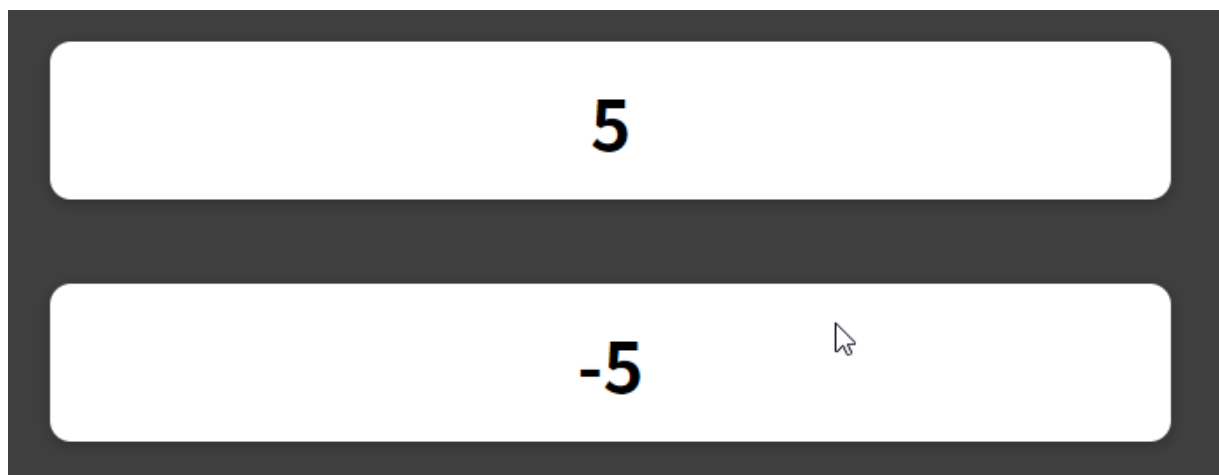


Marc Houben

H15 custom-react-hooks



H15 forward-backward-counter



H16 forms-user-input

Your Name

Your E-Mail

Submit

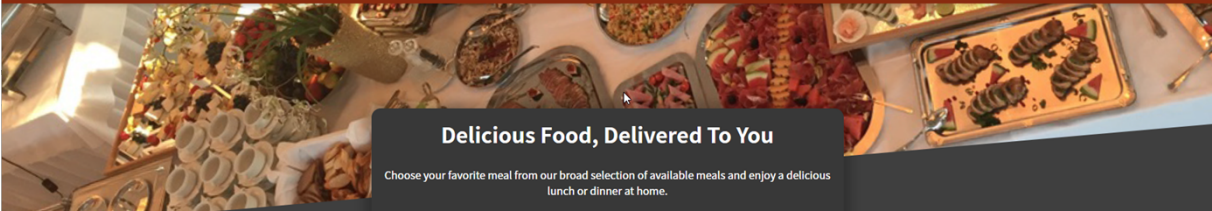
First Name **Last Name**

E-Mail Address

Submit

H17 http-forms-food-order

ReactMeals Your Cart 0



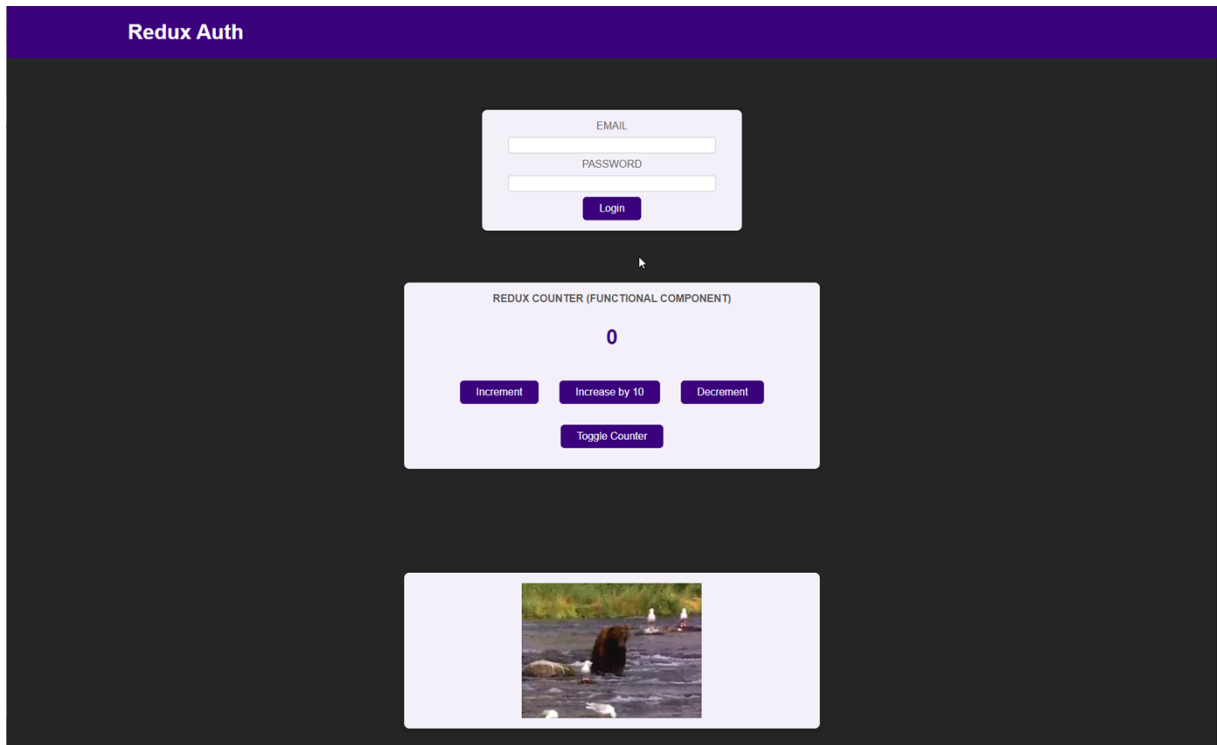
Delicious Food, Delivered To You

Choose your favorite meal from our broad selection of available meals and enjoy a delicious lunch or dinner at home.

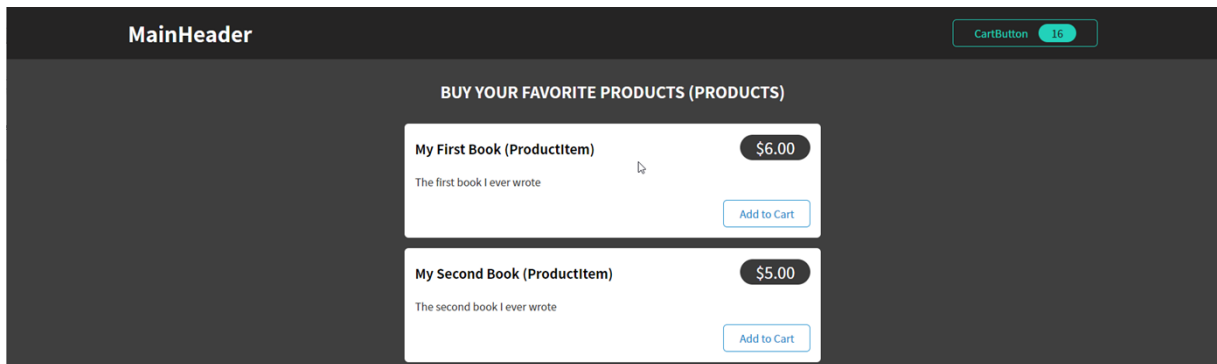
All our meals are cooked with high-quality ingredients, just-in-time and of course by experienced chefs!

Sushi <i>Finest fish and veggies</i> \$22.99	Amount <input type="text" value="1"/> + Add
Schnitzel <i>A German specialty!</i> \$16.50	Amount <input type="text" value="1"/> + Add

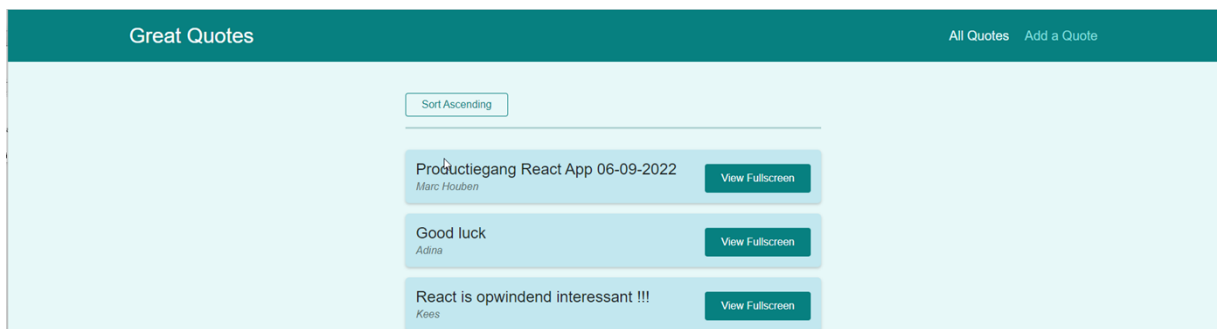
H18 redux



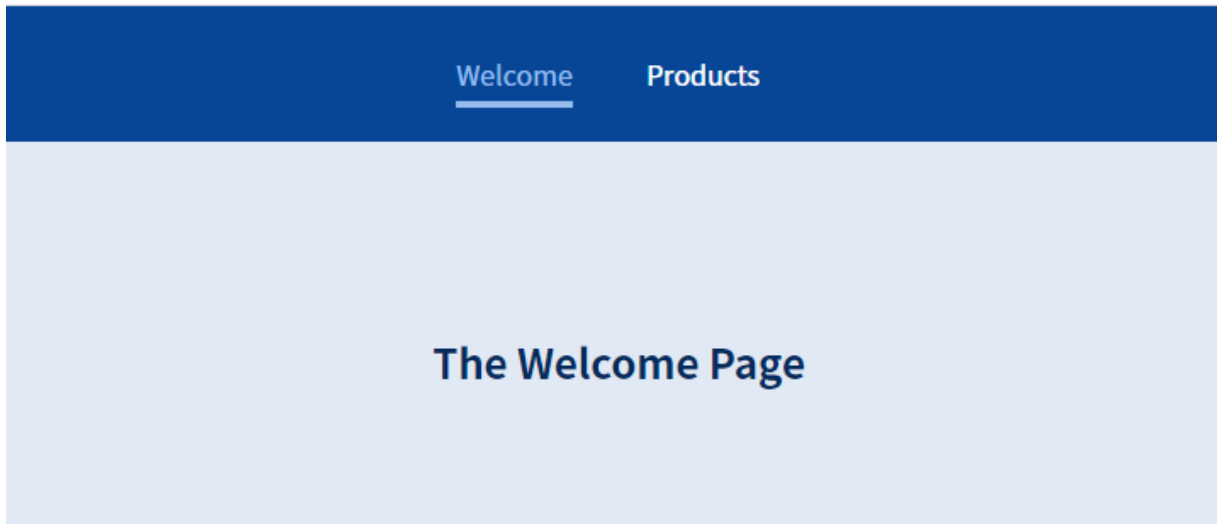
H19 advanced-redux



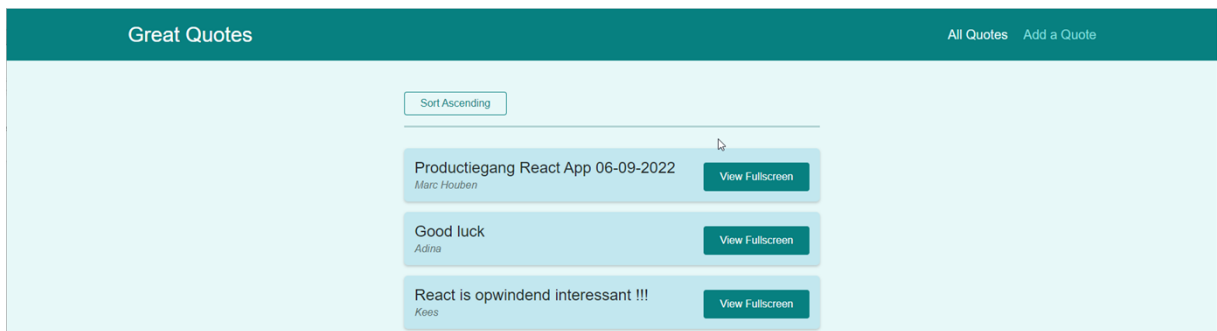
H20 add-quotes-v6



H20 multi-page-spa



H21 deploying-react



H22 authentication

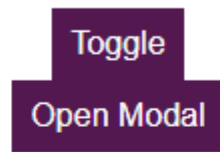


H23 meetups

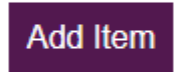
H23 nextjs-course

H24 animating-react

React Animations



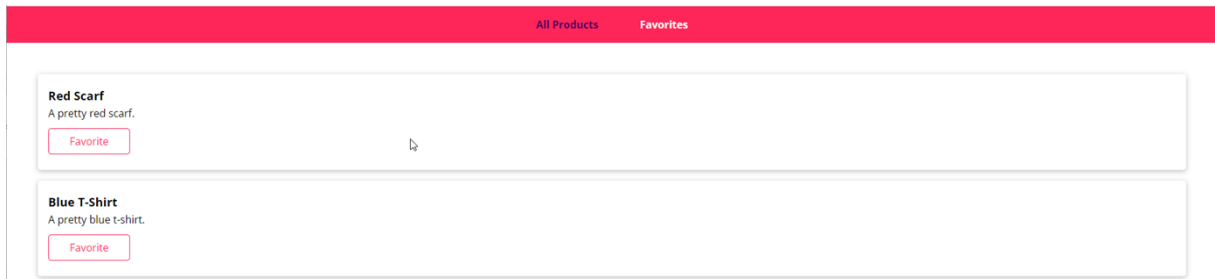
Animating Lists



Click Item to Remove.

1
2
3

H25 replace-redux-with-hooks



H26 testing-react-app

Learn React

sunt aut facere repellat provident occaecati excepturi optio reprehenderit
qui est esse
ea molestias quasi exercitationem repellat qui ipsa sit aut
eum et est occaecati
nesciunt quas odio
dolorem eum magni eos aperiam quia
magnam facilis autem
dolorem dolore est ipsam
nesciunt iure omnis dolorem tempora et accusantium
optio molestias id quia eum
et ea vero quia laudantium autem
in quibusdam tempore odit est dolorem
dolorum ut in voluptas mollitia et saepe quo animi
voluptatem eligendi optio
eveniet quod temporibus
sint suscipit perspiciatis velit dolorum rerum ipsa laboriosam odio
fugit voluptas sed molestias voluptatem provident
voluptate et itaque vero tempora molestiae
adipisci placeat illum aut reiciendis qui
doloribus ad provident suscipit at
asperiores ea ipsam voluptatibus modi minima quia sint
dolor sint quo a velit explicabo quia nam
maxime id vitae nihil numquam
autem hic labore sunt dolores incidunt

H27 react-typescript

Todo text

Add Todo

H28 react-hooks

Name

Amount

Add Ingredient


Filter by Title

Loaded Ingredients

Tomaten	3x
Bananen	7x
Kipsaté	25x
Bami	3x
Saté	100x
Kip	25x
Mandarijnen	275x

React Meetups All Meetups Add New Meetup My Favorites 0

All Meetups



Guys...
Google
Is your friend!

Meetup MH
Dam 200, Amsterdam
Leuk Uitje !!!

To Favorites

H29 my-todos

My Todos

- Learn React**
Delete
- Master React**
Delete
- Explore the full React course**
Delete