CSS Spinner – CSS Yukleme Animasyonu

* Once Spinner.js adinda klasorumuzu olusturuyoruz.
* Sonra google’dan Css Spinner diye aratip Spinnerimizi seciyoruz veya <https://projects.lukehaas.me/css-loaders/> bu siteden spinnerimizi seciyoruz. Ornekler asagida mevcut
* Daha sonra Spinner.js i kullanmak icin state’de loading:false degerini yaziyoruz. True oldugunda Spinneri gosterecegiz.
* Daha sonra asagidaki gibi belirleyecegimiz araliklarla Spinnerimizi gosterebiliriz.
* Mesela ben asagida once loading i tru yapiyorum ve true olunca spinnerim gozukuyor. Daha sonra axios isteginden sonra false oluyor ve bu istegin icindede false yapiyorum. Bu aradada Spinnerim gozukuyor.

  purchaseContinueHandler = () => {

    this.setState({ loading: true });

    //alert("You continue!");

    const order = {

      ingredient: this.state.ingredients,

      price: this.state.totalPrice,

      customer: {

        name: "Max Schwarzmuller",

        address: {

          street: "Cacesme Mahallesi Hakim Sokak",

          zipCode: "Teststreet 1",

          country: "Germany"

        },

        email: "test@test.com"

      },

      deliveryMethod: "fastest"

    };

    axios

      .post("/orders.json", order)

      .then(response => {

        this.setState({ loading: false, purchasing: false });

        console.log(response);

      })

      .catch(err => {

        this.setState({ loading: false, purchasing: false });

        console.error(err);

      });

  };

* Burada da orderSummary i bir degiskene atadim. Loading true olursa orderSummary Spinner olacak, false olursa orderSummery Hesap ozeti olacak.

    let orderSummary = (

      <OrderSummary

        ingredients={this.state.ingredients}

        purchaseCancelled={this.purchaseCancelHandler}

        purchaseContinued={this.purchaseContinueHandler}

        price={this.state.totalPrice}

      />

    );

    if (this.state.loading) {

      orderSummary = <Spinner />;

    }

    return (

      <Auxilliary>

        <Modal

          show={this.state.purchasing}

          modelClosed={this.purchaseCancelHandler}

        >

          {orderSummary}

        </Modal>

)

Ornek JS Spinner.js

import React from 'react';

const Spinner = () => (

    <div className="spinner">Loading...</div>

)

export default Spinner;

ORNEK CSS \_Spinner.scss

.spinner {

  color: #400040;

  font-size: 90px;

  text-indent: -9999em;

  overflow: hidden;

  width: 1em;

  height: 1em;

  border-radius: 50%;

  margin: 72px auto;

  position: relative;

  -webkit-transform: translateZ(0);

  -ms-transform: translateZ(0);

  transform: translateZ(0);

  -webkit-animation: load6 1.7s infinite ease, round 1.7s infinite ease;

  animation: load6 1.7s infinite ease, round 1.7s infinite ease;

}

@-webkit-keyframes load6 {

  0% {

    box-shadow: 0 -0.83em 0 -0.4em, 0 -0.83em 0 -0.42em, 0 -0.83em 0 -0.44em,

      0 -0.83em 0 -0.46em, 0 -0.83em 0 -0.477em;

  }

  5%,

  95% {

    box-shadow: 0 -0.83em 0 -0.4em, 0 -0.83em 0 -0.42em, 0 -0.83em 0 -0.44em,

      0 -0.83em 0 -0.46em, 0 -0.83em 0 -0.477em;

  }

  10%,

  59% {

    box-shadow: 0 -0.83em 0 -0.4em, -0.087em -0.825em 0 -0.42em,

      -0.173em -0.812em 0 -0.44em, -0.256em -0.789em 0 -0.46em,

      -0.297em -0.775em 0 -0.477em;

  }

  20% {

    box-shadow: 0 -0.83em 0 -0.4em, -0.338em -0.758em 0 -0.42em,

      -0.555em -0.617em 0 -0.44em, -0.671em -0.488em 0 -0.46em,

      -0.749em -0.34em 0 -0.477em;

  }

  38% {

    box-shadow: 0 -0.83em 0 -0.4em, -0.377em -0.74em 0 -0.42em,

      -0.645em -0.522em 0 -0.44em, -0.775em -0.297em 0 -0.46em,

      -0.82em -0.09em 0 -0.477em;

  }

  100% {

    box-shadow: 0 -0.83em 0 -0.4em, 0 -0.83em 0 -0.42em, 0 -0.83em 0 -0.44em,

      0 -0.83em 0 -0.46em, 0 -0.83em 0 -0.477em;

  }

}

@keyframes load6 {

  0% {

    box-shadow: 0 -0.83em 0 -0.4em, 0 -0.83em 0 -0.42em, 0 -0.83em 0 -0.44em,

      0 -0.83em 0 -0.46em, 0 -0.83em 0 -0.477em;

  }

  5%,

  95% {

    box-shadow: 0 -0.83em 0 -0.4em, 0 -0.83em 0 -0.42em, 0 -0.83em 0 -0.44em,

      0 -0.83em 0 -0.46em, 0 -0.83em 0 -0.477em;

  }

  10%,

  59% {

    box-shadow: 0 -0.83em 0 -0.4em, -0.087em -0.825em 0 -0.42em,

      -0.173em -0.812em 0 -0.44em, -0.256em -0.789em 0 -0.46em,

      -0.297em -0.775em 0 -0.477em;

  }

  20% {

    box-shadow: 0 -0.83em 0 -0.4em, -0.338em -0.758em 0 -0.42em,

      -0.555em -0.617em 0 -0.44em, -0.671em -0.488em 0 -0.46em,

      -0.749em -0.34em 0 -0.477em;

  }

  38% {

    box-shadow: 0 -0.83em 0 -0.4em, -0.377em -0.74em 0 -0.42em,

      -0.645em -0.522em 0 -0.44em, -0.775em -0.297em 0 -0.46em,

      -0.82em -0.09em 0 -0.477em;

  }

  100% {

    box-shadow: 0 -0.83em 0 -0.4em, 0 -0.83em 0 -0.42em, 0 -0.83em 0 -0.44em,

      0 -0.83em 0 -0.46em, 0 -0.83em 0 -0.477em;

  }

}

@-webkit-keyframes round {

  0% {

    -webkit-transform: rotate(0deg);

    transform: rotate(0deg);

  }

  100% {

    -webkit-transform: rotate(360deg);

    transform: rotate(360deg);

  }

}

@keyframes round {

  0% {

    -webkit-transform: rotate(0deg);

    transform: rotate(0deg);

  }

  100% {

    -webkit-transform: rotate(360deg);

    transform: rotate(360deg);

  }

}