דף הבית:

import React, { useState } from 'react';

import { useSpring, animated } from 'react-spring';

const images = [

   "./a.jpg", "./b.jpg", "./c.jpg", "./d.jpg"

]

const Home=()=>{

   const [index, setIndex] = useState(0);

   const props = useSpring({

       opacity: 1,

       from: { opacity: 0.85 },

       reset: true,

       config: { duration: 3000 },

       onRest: () => setIndex((index + 1) % images.length),

   });

   return (

       <dive>

         <br></br>

           <div style={{ height: "", opacity: "80%",marginTop:"100px" }}>

           <animated.div style={props}>

               <img src={images[index]} width="100%" alt={`Image ${index + 1}`} />

           </animated.div></div>

           <div style={{ backgroundColor: "#C08F48", height: "5px" }}></div>

       </dive>

      // <>home page</>

   )

}

export default Home

navbar:

לוגי:

1. לאפשר שלמנהל לא יהיה את התגית <basket>

עיצוב:

1. ב- return : style ...

 <div style={{backgroundColor:'#C08F48',opacity:'0.9',padding:'5px',position: 'fixed',width:'95%',left: '2.5%',zIndex:'100'}}>

1. start:- להפוך ללינק לדף הבית וה style ...

<Link to={'/'} ><img alt="logo" src="logo.png" height="60" className="mr-2" style={{paddingLeft:'15px'}}></img></Link>

1. end:

<div className="flex align-items-center gap-2" style={{paddingRight:'25px'}}>

            <Button icon="pi pi-shopping-cart" rounded text severity="info" onClick={() => setVisibleRight(true)} style={{color:'#C08F48'}}>

login:

לוגי:

עיצוב:

1. ב return: - style…

 <>

        <br></br>

        <div className="card" style={{marginTop:"150px"}}>

products:

לוגי:

1. ביטול האפשרות להראות את השגיאות (כדי שהדף לא יפול...)
2. מחיקת ה רשאןמע (הכוכבים, לא קשור...)

עיצוב:

1. ב return:

<> <br></br>

<div style={{width:'100%',left:'8%',textAlign:'center',marginTop:'100px',opacity:'0.8', backgroundImage:'url(products.jpg)',height:'200px'}}>

<br></br>

<br></br>

<h1 style={{marginTop:'30px',fontSize:'50px'}}>Our Products:</h1>

</div>

<div>

<DataView value={products?products:[]} listTemplate={listTemplate} layout={layout} header={header()} style={{ padding: '5%' }} />

</  div> </>

1. ב- greaditem:

בdiv שעוטף את הכל:

<div className="col-14 sm:col-6 lg:col-12 xl:col-4 p-4" key={product.\_id}>

להוריד span של קטגוריה...

ללינק של התמונה להוסיף style:

<Link to={`/product/${product.\_id}`} style={{width:'100%',marginLeft:'20%'}}>

לdiv של שם המוצר להוסיף style:

<div className=" text-2xl font-bold " style={{textAlign:'center', height:'70px'}}>{product.name}</div>

1. ב listitem:

בשם של המוצר להוסיף style:

<div className="text-2xl font-bold text-900" style={{textAlign:'center'}}>{product.name}</div>

סל צדדי:

לוגי:

1. יפתח כאשר יתווסף מוצר לסל ויהיה אפשרות משם ללכת לסל המלא...
2. כאשר הסל יהיה ריק יראו עיצוב לסל ריק

עיצוב:

1. לסל ריק:

 <>

<img className=" xl:w-15rem  xl:block mx-auto" src={'emptyCart.png'} alt={'emptyCart'} style={{marginTop:'100px'}}/>

<h2 style={{textAlign:'center'}}>Oups! Your cart is empty,</h2>

<Button onClick={()=>{setVisibleRight(false);navigate('/products')}} style={{marginLeft:'75px',color:'white',backgroundColor:'transparent',border:'none'}}>continue shopping</Button>

</>

1. לסל צדדי מלא:

ב imageBodyTemplate:

return <div className='flex p-overlay-badge'>

<img src={"http://localhost:1234/uploads/" + p.product.imageURL[0].split("\\")[2]} alt={p.product.name} className="w-6rem shadow-2 border-round" />

<Badge value={p.quantity} style={{backgroundColor:'white'}} ></Badge>

</div>;

ב header:

 <div className="flex flex-wrap align-items-center justify-content-between gap-4">

<span className="text-xl text-900 font-bold" >Basket</span>

        </div>

ב- Return:

<div >

<DataTable value={products} scrollable scrollHeight="480px" tableStyle={{ minWidth: '80px', minHeight: '430px' }}>

<Column field="name" body={nameBodyTemplate} header="Name"></Column>

<Column header="Image" body={imageBodyTemplate}></Column>

<Column field="price" header="Price" body={priceBodyTemplate}></Column>

</DataTable>

<div className="mt-auto">

<hr className="flex flex-wrap align-items-center justify-content-between gap-2" />

<h3 style={{ textAlign: 'center' }}>Total payment {basket.payment} ₪</h3>

<Button onClick={() => { setVisibleRight(false); navigate('/basket') }} style={{ marginLeft: '85px', color: 'white', backgroundColor: 'transparent', border: 'none', position: 'fixed', Button: '0', zIndex: '100', fontSize: '200' }}><b>to full basket</b></Button>

</div>

                </div>

דף מוצר:

לוגי:

עיצוב:

1. ב itemTemplate:

return <img src={"http://localhost:1234/uploads/" + item.split("\\")[2]} alt={item.alt} style={{ width: '100%',maxHeight:'420px' }} />;

1. ב :thumbnailTemplate

        return <img src={"http://localhost:1234/uploads/" + item.split("\\")[2]} alt={item.alt} style={{ width: '50px' }} />;

1. ב return:

<>

<br></br>

<div className="" style={{ direction: 'rtl', marginTop: '100px',width:'80%',marginLeft:'10%'}}>

<div className="flex flex-column md:flex-row" >

          <div className="flex-column md:flex-row" style={{ maxWidth: '500px', minWidth: '300px'}}>

                        <h1><b>{product.name}</b></h1>

                        <h3><b>{product.description}</b></h3>

                        <h3><b>₪{product.price}</b></h3>

                        <InputNumber inputId="minmax-buttons" value={value} onValueChange={(e) => setValue(e.value)} mode="decimal" showButtons min={0} max={product.quantity} /><br></br><br></br>

                        <Button icon="pi pi-shopping-cart" className="p-button-rounded" onClick={addProductToBasket} disabled={product.quantity === 0}></Button>

                        <br></br><br></br>

                    </div>

                    <Divider layout="vertical" className="hidden md:flex"> </Divider>

                    <div className="flex flex-column md:flex-row" style={{ maxWidth: '500px', minWidth: '300px' }}>

                        <Galleria value={product.imageURL} responsiveOptions={responsiveOptions} numVisible={5} circular style={{ maxWidth: '500px'}}

                            showItemNavigators showItemNavigatorsOnHover item={itemTemplate} thumbnail={thumbnailTemplate} />

           </div>

      </div>

</div>       </>

עריכת פרטים אישיים:

לוגי:

עיצוב:

1. ב return:

<>

<br></br>

<div style={{marginTop:'200px'}}>

<form onSubmit={formik.handleSubmit}  className="flex flex-wrap  gap-3 p-fluid" style={{width:'50%',marginLeft:'25%'}}>

<Toast ref={toast} />

<div className="flex-auto">

<label htmlFor="value" className="font-bold block mb-2">FirstName</label>

                <AutoComplete

                    inputId="ac\_FirstName"

                    name="firstName"

                    value={formik.values.firstName}

                    className={classNames({ 'p-invalid': isFormFieldInvalid('firstName') })}

                    onChange={(e) => {

                        formik.setFieldValue('firstName', e.value);

                    }}

                />

                {getFormErrorMessage('firstName')}

                </div>

                <div className="flex-auto">

                <label htmlFor="value" className="flex-auto font-bold block mb-2">LastName</label>

                <AutoComplete

                    inputId="ac\_LastName"

                    name="lastName"

                    value={formik.values.lastName}

                    className={classNames({ 'p-invalid': isFormFieldInvalid('lastName') })}

                    onChange={(e) => {

                        formik.setFieldValue('lastName', e.value);

                    }}

                />

                {getFormErrorMessage('lastName')}

                </div>

            <div className="flex-auto">

                <label htmlFor="value" className="font-bold block mb-2">Phone</label>

                <AutoComplete

                    inputId="ac\_Phone"

                    name="phone"

                    value={formik.values.phone}

                    className={classNames({ 'p-invalid': isFormFieldInvalid('phone') })}

                    onChange={(e) => {

                        formik.setFieldValue('phone', e.value);

                    }}

                />

                {getFormErrorMessage('phone')}

                </div>

                <div className="flex-auto">

                <label htmlFor="value" className="font-bold block mb-2" >User\_id</label>

                <AutoComplete

                    inputId="ac\_user\_id"

                    name="user\_id"

                    value={formik.values.user\_id}

                    className={classNames({ 'p-invalid': isFormFieldInvalid('user\_id') })}

                    onChange={(e) => {

                        formik.setFieldValue('user\_id', e.value);

                    }}

                />

                {getFormErrorMessage('user\_id')}

                </div>

                <Button label="Submit" type="submit" icon="pi pi-check" style={{backgroundColor:'#C08F48',border:'black'}} />

            </form>

        </div>

        </>

רישום:

לוגי:

עיצוב:

1. ב return:

<br></br>

        <div style={{ marginTop: '200px' }}>

            <form onSubmit={formik.handleSubmit} className="flex flex-wrap  gap-3 p-fluid" style={{ width: '50%', marginLeft: '25%' }}>

                <Toast ref={toast} />

                <div className="flex-auto">

                    <label htmlFor="firstName" className="block">

                        FirstName

                    </label>

                    <AutoComplete

                        inputId="ac\_FirstName"

                        name="firstName"

                        value={formik.values.firstName}

                        className={classNames({ 'p-invalid': isFormFieldInvalid('firstName') })}

                        onChange={(e) => {

                            formik.setFieldValue('firstName', e.value);

                        }}

                    />

                    {getFormErrorMessage('firstName')}

                </div>

                <div className="flex-auto">

                    <label htmlFor="value">LastName</label>

                    <AutoComplete

                        inputId="ac\_LastName"

                        name="lastName"

                        value={formik.values.lastName}

                        className={classNames({ 'p-invalid': isFormFieldInvalid('lastName') })}

                        onChange={(e) => {

                            formik.setFieldValue('lastName', e.value);

                        }}

                    />

                    {getFormErrorMessage('lastName')}

                </div>

                <div className="flex-auto">

                    <label htmlFor="value">Password</label>

                    <Password

                        inputId="in\_password"

                        name="password"

                        rows={5}

                        cols={30}

                        className={classNames({ 'p-invalid': isFormFieldInvalid('password') })}

                        value={formik.values.password}

                        feedback={false}

                        onChange={(e) => {

                            formik.setFieldValue('password', e.target.value);

                        }}

                    />

                    {getFormErrorMessage('password')}

                </div>

                <div className="flex-auto">

                    <label htmlFor="value">Email</label>

                    <AutoComplete

                        inputId="ac\_email"

                        name="email"

                        value={formik.values.email}

                        //  suggestions={items}

                        //  completeMethod={search}

                        className={classNames({ 'p-invalid': isFormFieldInvalid('email') })}

                        onChange={(e) => {

                            formik.setFieldValue('email', e.value);

                        }}

                    />

                    {getFormErrorMessage('email')}

                </div>

                <div className="flex-auto">

                    <label htmlFor="value">Phone</label>

                    <AutoComplete

                        inputId="ac\_Phone"

                        name="phone"

                        value={formik.values.phone}

                        className={classNames({ 'p-invalid': isFormFieldInvalid('phone') })}

                        onChange={(e) => {

                            formik.setFieldValue('phone', e.value);

                        }}

                    />

                    {getFormErrorMessage('phone')}

                </div>

                <div className="flex-auto">

                    <label htmlFor="value">User\_id</label>

                    <AutoComplete

                        inputId="ac\_user\_id"

                        name="user\_id"

                        value={formik.values.user\_id}

                        className={classNames({ 'p-invalid': isFormFieldInvalid('user\_id') })}

                        onChange={(e) => {

                            formik.setFieldValue('user\_id', e.value);

                        }}

                    />

                    {getFormErrorMessage('user\_id')}

                </div>

                <Button label="Submit" type="submit" icon="pi pi-check" style={{ backgroundColor: '#C08F48', border: 'black' }} />

            </form>

        </div>

    </>

סל מלא:

לוגי:

1. לבטל אפשרות של סל ריק

עיצוב:

1. ב return:

<>

            <br></br>

            <div style={{ marginTop: '100px' }}>

                <h1 style={{textAlign:'center'}} >Basket</h1>

                <div className="card">

                    <div className="flex flex-column md:flex-row" >

                        <p className="flex flex-column md:flex-row" style={{ minWidth: '75%' }}>

                            <DataView value={products} listTemplate={listTemplate} header={null} />

                        </p>

                        <Divider layout="vertical" className="hidden md:flex"> </Divider>

                        <p style={{ minWidth: '25%', textAlign: "center" }}>

                            <h3>{`₪${basket.payment} לתשלום`}</h3>

                            <Button onClick={() => checkUser()} style={{ backgroundColor: '#C08F48', border: 'black' }}>להזמנה</Button>

                        </p>

                    </div>

                </div>

            </div>

        </>

הזמנה:

לוגי:

1. לבטל אפשרות של סל ריק

עיצוב:

1. ב return:

 <>

            <div style={{ margin: '20px' }}>

                {

                    !submit ?

                        <form onSubmit={formik.handleSubmit} className="flex flex-wrap  gap-3 p-fluid" style={{width:'80%',marginLeft:'10%'}}>

                            <Toast ref={toast} />

                            <div className="flex-auto">

                                <label htmlFor="value" className="block">firstName</label>

                                <AutoComplete

                                    inputId="ac\_FirstName"

                                    name="firstName"

                                    value={formik.values.firstName}

                                    className={classNames({ 'p-invalid': isFormFieldInvalid('firstName') })}

                                    onChange={(e) => {

                                        formik.setFieldValue('firstName', e.value);

                                    }}

                                />

                                {getFormErrorMessage('firstName')}

                            </div>

                            <div className="flex-auto">

                                <label htmlFor="value" className="block">lastName</label>

                                <AutoComplete

                                    inputId="ac\_LastName"

                                    name="lastName"

                                    value={formik.values.lastName}

                                    className={classNames({ 'p-invalid': isFormFieldInvalid('lastName') })}

                                    onChange={(e) => {

                                        formik.setFieldValue('lastName', e.value);

                                    }}

                                />

                                {getFormErrorMessage('lastName')}

                            </div>

                            <div className="flex-auto">

                                <label htmlFor="value" className="block">City</label>

                                <AutoComplete

                                    inputId="ac\_City"

                                    name="city"

                                    value={formik.values.city}

                                    className={classNames({ 'p-invalid': isFormFieldInvalid('city') })}

                                    onChange={(e) => {

                                        formik.setFieldValue('city', e.value);

                                    }}

                                />

                                {getFormErrorMessage('city')}

                            </div>

                            <div className="flex-auto">

                                <label htmlFor="value" className="block">Street</label>

                                <AutoComplete

                                    inputId="ac\_Street"

                                    name="street"

                                    value={formik.values.street}

                                    className={classNames({ 'p-invalid': isFormFieldInvalid('street') })}

                                    onChange={(e) => {

                                        formik.setFieldValue('street', e.value);

                                    }}

                                />

                                {getFormErrorMessage('street')}

                            </div>

                            <div className="flex-auto">

                                <label htmlFor="value" className="block">houseNumber</label>

                                <AutoComplete

                                    inputId="ac\_houseNumber"

                                    name="houseNumber"

                                    value={formik.values.houseNumber}

                                    className={classNames({ 'p-invalid': isFormFieldInvalid('houseNumber') })}

                                    onChange={(e) => {

                                        formik.setFieldValue('houseNumber', e.value);

                                    }}

                                />

                                {getFormErrorMessage('houseNumber')}

                            </div>

                            <div className="flex-auto">

                                <label htmlFor="value" className="block">apartment</label>

                                <AutoComplete

                                    inputId="ac\_apartment"

                                    name="houseNumber"

                                    value={formik.values.apartment}

                                    className={classNames({ 'p-invalid': isFormFieldInvalid('apartment') })}

                                    onChange={(e) => {

                                        formik.setFieldValue('apartment', e.value);

                                    }}

                                />

                                {getFormErrorMessage('apartment')}

                            </div>

                            <div className="flex-auto">

                                <label htmlFor="value" className="block">postalCode</label>

                                <AutoComplete

                                    inputId="ac\_postalCode"

                                    name="postalCode"

                                    value={formik.values.postalCode}

                                    className={classNames({ 'p-invalid': isFormFieldInvalid('postalCode') })}

                                    onChange={(e) => {

                                        formik.setFieldValue('postalCode', e.value);

                                    }}

                                />

                                {getFormErrorMessage('postalCode')}

                            </div>

                            <div className="flex-auto">

                                <label htmlFor="value" className="block">Phone</label>

                                <AutoComplete

                                    inputId="ac\_Phone"

                                    name="phone"

                                    value={formik.values.phone}

                                    className={classNames({ 'p-invalid': isFormFieldInvalid('phone') })}

                                    onChange={(e) => {

                                        formik.setFieldValue('phone', e.value);

                                    }}

                                />

                                {getFormErrorMessage('phone')}

                            </div>

                            <div className="flex-auto">

                                <Checkbox onChange={e => setChecked(e.checked)} checked={checked}>שמירה ככתובת דפולטיבית</Checkbox>

                            </div>

                            <Button label="שמירת כתובת" type="submit" style={{backgroundColor:'#C08F48',border:'black'}}/>

                        </form> :

                <div style={{marginLeft:'35%', border: "5px solid #C08F48", padding: "", width: "30%",textAlign:'center'  }}>

                <h4>{formik.values.firstName} {formik.values.lastName}</h4>

                            <h4>{formik.values.street} {formik.values.houseNumber}, {formik.values.apartment}</h4>

                            <h4>{formik.values.city}</h4>

                            <h4>Israel</h4>

                            <h4>{formik.values.phone}</h4>

                        </div>

                }

            </div>

        </>

כתובת:

לוגי:

1. לבטל אפשרות של סל ריק

עיצוב:

1. ב return:

<>

            <div style={{ margin: '20px' }}>

                {

                    !submit ?

                        <form onSubmit={formik.handleSubmit} className="flex flex-wrap  gap-3 p-fluid" style={{width:'80%',marginLeft:'10%'}}>

                            <Toast ref={toast} />

                            <div className="flex-auto">

                                <label htmlFor="value" className="block">firstName</label>

                                <AutoComplete

                                    inputId="ac\_FirstName"

                                    name="firstName"

                                    value={formik.values.firstName}

                                    className={classNames({ 'p-invalid': isFormFieldInvalid('firstName') })}

                                    onChange={(e) => {

                                        formik.setFieldValue('firstName', e.value);

                                    }}

                                />

                                {getFormErrorMessage('firstName')}

                            </div>

                            <div className="flex-auto">

                                <label htmlFor="value" className="block">lastName</label>

                                <AutoComplete

                                    inputId="ac\_LastName"

                                    name="lastName"

                                    value={formik.values.lastName}

                                    className={classNames({ 'p-invalid': isFormFieldInvalid('lastName') })}

                                    onChange={(e) => {

                                        formik.setFieldValue('lastName', e.value);

                                    }}

                                />

                                {getFormErrorMessage('lastName')}

                            </div>

                            <div className="flex-auto">

                                <label htmlFor="value" className="block">City</label>

                                <AutoComplete

                                    inputId="ac\_City"

                                    name="city"

                                    value={formik.values.city}

                                    className={classNames({ 'p-invalid': isFormFieldInvalid('city') })}

                                    onChange={(e) => {

                                        formik.setFieldValue('city', e.value);

                                    }}

                                />

                                {getFormErrorMessage('city')}

                            </div>

                            <div className="flex-auto">

                                <label htmlFor="value" className="block">Street</label>

                                <AutoComplete

                                    inputId="ac\_Street"

                                    name="street"

                                    value={formik.values.street}

                                    className={classNames({ 'p-invalid': isFormFieldInvalid('street') })}

                                    onChange={(e) => {

                                        formik.setFieldValue('street', e.value);

                                    }}

                                />

                                {getFormErrorMessage('street')}

                            </div>

                            <div className="flex-auto">

                                <label htmlFor="value" className="block">houseNumber</label>

                                <AutoComplete

                                    inputId="ac\_houseNumber"

                                    name="houseNumber"

                                    value={formik.values.houseNumber}

                                    className={classNames({ 'p-invalid': isFormFieldInvalid('houseNumber') })}

                                    onChange={(e) => {

                                        formik.setFieldValue('houseNumber', e.value);

                                    }}

                                />

                                {getFormErrorMessage('houseNumber')}

                            </div>

                            <div className="flex-auto">

                                <label htmlFor="value" className="block">apartment</label>

                                <AutoComplete

                                    inputId="ac\_apartment"

                                    name="houseNumber"

                                    value={formik.values.apartment}

                                    className={classNames({ 'p-invalid': isFormFieldInvalid('apartment') })}

                                    onChange={(e) => {

                                        formik.setFieldValue('apartment', e.value);

                                    }}

                                />

                                {getFormErrorMessage('apartment')}

                            </div>

                            <div className="flex-auto">

                                <label htmlFor="value" className="block">postalCode</label>

                                <AutoComplete

                                    inputId="ac\_postalCode"

                                    name="postalCode"

                                    value={formik.values.postalCode}

                                    className={classNames({ 'p-invalid': isFormFieldInvalid('postalCode') })}

                                    onChange={(e) => {

                                        formik.setFieldValue('postalCode', e.value);

                                    }}

                                />

                                {getFormErrorMessage('postalCode')}

                            </div>

                            <div className="flex-auto">

                                <label htmlFor="value" className="block">Phone</label>

                                <AutoComplete

                                    inputId="ac\_Phone"

                                    name="phone"

                                    value={formik.values.phone}

                                    className={classNames({ 'p-invalid': isFormFieldInvalid('phone') })}

                                    onChange={(e) => {

                                        formik.setFieldValue('phone', e.value);

                                    }}

                                />

                                {getFormErrorMessage('phone')}

                            </div>

                            <div className="flex-auto">

                                <Checkbox onChange={e => setChecked(e.checked)} checked={checked}>שמירה ככתובת דפולטיבית</Checkbox>

                            </div>

                            <Button label="שמירת כתובת" type="submit" style={{backgroundColor:'#C08F48',border:'black'}}/>

                        </form> :

                <div style={{marginLeft:'35%', border: "5px solid #C08F48", padding: "", width: "30%",textAlign:'center'  }}>

                <h4>{formik.values.firstName} {formik.values.lastName}</h4>

                            <h4>{formik.values.street} {formik.values.houseNumber}, {formik.values.apartment}</h4>

                            <h4>{formik.values.city}</h4>

                            <h4>Israel</h4>

                            <h4>{formik.values.phone}</h4>

                        </div>

                }

            </div>

        </>

    )