Contents

[for css style: 4](#_Toc75436167)

[In config->jest->webpack.config(2nd last file): 4](#_Toc75436168)

[In config->jest->webpack.config(last last file): 4](#_Toc75436169)

[-TO CREATE WRAPPER 5](#_Toc75436170)

[USE IT LIKE THIS: 5](#_Toc75436171)

[FOR LAYOUT: 6](#_Toc75436172)

[INSTALL PROPSTYPES 6](#_Toc75436173)

[RESPONSIVE MEDIA QUERY IN CSS 7](#_Toc75436174)

[PROJECT STURCTURE 8](#_Toc75436175)

[App.js 8](#_Toc75436176)

[Layout.js 8](#_Toc75436177)

[Layout.css 8](#_Toc75436178)

[PizzaBuilder.js 9](#_Toc75436179)

[Pizza.js 10](#_Toc75436180)

[Pizza.css 10](#_Toc75436181)

[PizzaIngredients.js 11](#_Toc75436182)

[PizzaIngredients.css 12](#_Toc75436183)

[BuildControls.js 15](#_Toc75436184)

[BuildControl.css 16](#_Toc75436185)

[BuildControl.js 16](#_Toc75436186)

[ADD AND REMOVE BUTTON WITH DISABLE LESS BUTTON: 18](#_Toc75436187)

[PizzaBuilder.js 18](#_Toc75436188)

[BuildControls.js 19](#_Toc75436189)

[BuildControl.js 20](#_Toc75436190)

[ADDING PRICE ON BASIS OF ADDITION OF TYPES OF INGREDIENTS: 21](#_Toc75436191)

[PizzaBuilder.js 21](#_Toc75436192)

[BuildControls.js 21](#_Toc75436193)

[ORDER BUTTON SHOWN ON CONDITION: 22](#_Toc75436194)

[PizzaBuilder.js 22](#_Toc75436195)

[CREATING ORDER SUMMARY MODAL: 25](#_Toc75436196)

[Modal.js: 25](#_Toc75436197)

[Modal.css: 26](#_Toc75436198)

[PizzaBuilder.js 26](#_Toc75436199)

[OrderSummary.js 27](#_Toc75436200)

[ADDING ANIMATION: 28](#_Toc75436201)

[PizzaBuilder.js 28](#_Toc75436202)

[BuildControls.js 28](#_Toc75436203)

[IMPLEMENT BACKDROP 29](#_Toc75436204)

[Backdrop.js 29](#_Toc75436205)

[Modal.js 30](#_Toc75436206)

[Pizzabuilder.js 30](#_Toc75436207)

[purchasedCancelHandler function. 31](#_Toc75436208)

[CREATING BUTTON UI 31](#_Toc75436209)

[Button.js 31](#_Toc75436210)

[Button.css 31](#_Toc75436211)

[OrderSummary.js 32](#_Toc75436212)

[PizzaBuilder.js 33](#_Toc75436213)

[CANCEL AND CONTINUE FUNCTIONS: 33](#_Toc75436214)

[ADDING TOOLBAR: 34](#_Toc75436215)

[TOOLBAR.JS 34](#_Toc75436216)

[Toolbar.css 35](#_Toc75436217)

[LAYOUT.JS 35](#_Toc75436218)

[LAYOUT.CSS 36](#_Toc75436219)

[ADDING LOGO IN NAVBAR 36](#_Toc75436220)

[Logo.js 36](#_Toc75436221)

[Logo.css 36](#_Toc75436222)

[Toolbar.js 36](#_Toc75436223)

[ADDING REUSABLE NAVIGATION ITEMS: 37](#_Toc75436224)

[Layout.js 37](#_Toc75436225)

[Toolbar.js 38](#_Toc75436226)

[NavigationItems.js 38](#_Toc75436227)

[NavigationItems.css 39](#_Toc75436228)

[NavigationItem.js 39](#_Toc75436229)

[NavigationItem.css 39](#_Toc75436230)

[CREATING RESPONSIVE SIDE BAR: 40](#_Toc75436231)

[SideDrawer.js 40](#_Toc75436232)

[SideDrawer.css 41](#_Toc75436233)

[Layout.js 41](#_Toc75436234)

[ADJUSTMENTS OF RESPONSIVE NAVBAR: 42](#_Toc75436235)

[NagivationItem.css: 42](#_Toc75436236)

[NavigationItems.css 43](#_Toc75436237)

[NAVIGATIONS ITEMS UNVISIBLE IN MOBILE VIEW. 43](#_Toc75436238)

[Toolbar.css 43](#_Toc75436239)

[Toolbar.js 43](#_Toc75436240)

[REUSING BACKDROP IN NAVBAR 44](#_Toc75436241)

[TAKE REFERENCE FROM MODAL.js 44](#_Toc75436242)

[SideDrawer.js 44](#_Toc75436243)

[SideDrawer.css 45](#_Toc75436244)

[Layout.js 46](#_Toc75436245)

[Adding SideDrawer toggle button: 47](#_Toc75436246)

[DrawerToggle.js 47](#_Toc75436247)

[DrawerToggle.css 47](#_Toc75436248)

[Toolbar.js 47](#_Toc75436249)

[Layout.js 48](#_Toc75436250)

# for css style:

-npm run eject

# In config->jest->webpack.config(2nd last file):

{

test:

use:

require...

{

options:{

importLoaders:1,

//ADD THESE TWO LINES

modules: true,

localIdentName: '[name]\_\_[local]\_\_[hash:base64:5]'

},

}

}

## In config->jest->webpack.config(last last file):

fallback:require.........),

use: [

{

loader: req.....

options:{

importLoader:1,

.

.

modules: true,

localIdentName: '[name]\_\_[local]\_\_[hash:base64:5]'

}

}

]

# -TO CREATE WRAPPER

1.create hoc folder and inside it Auxilliary.js

const Aux = (props) => props.children;

export default Aux;

## USE IT LIKE THIS:

import React from 'react';

import Aux from '../../hoc/Auxilliary';

const Layout = (props) => {

return(

<Aux>

<div>ToolBar,SideDrawer,Backdrop</div>

<main>

{props.children}

</main>

</Aux>

);

}

# FOR LAYOUT:

import React from 'react';

import Aux from '../../hoc/Auxilliary';

import './Layout.css';

const Layout = (props) => {

return(

<Aux>

<div>ToolBar,SideDrawer,Backdrop</div>

<main className='Content'>

{props.children}

</main>

</Aux>

);

}

export default Layout;

# INSTALL PROPSTYPES

npm install --save prop-types

CODE:

import PropTypes from 'prop-types';

PizzaIngredients.propTypes = {

type: PropTypes.string.isRequired

}

# RESPONSIVE MEDIA QUERY IN CSS

@media (min-width: 500px) and (min-height: 400px){

.Pizza{

width: 350px;

height: 300px;

}

}

@media (min-width: 500px) and (min-height: 401px){

.Pizza{

width: 450px;

height: 400px;

}

}

@media (min-width: 1000px) and (min-height: 700px){

.Pizza{

width: 700px;

height: 600px;

}

}

# PROJECT STURCTURE

## App.js

import React from 'react';

import Layout from '../components/Layout/Layout';

import PizzaBuilder  from './PizzaBuilder/PizzaBuilder';

const App = () => {

  return(

    <div className="App">

      <Layout>

        <PizzaBuilder/>

      </Layout>

    </div>

  )

}

export default App;

**FOR LAYOUT:**

## Layout.js

import React from 'react';

import Aux from '../../hoc/Auxilliary';

import  './Layout.css';

const Layout = (props) => {

    return(

        <Aux>

            <div>ToolBar,SideDrawer,Backdrop</div>

            <main className='Content'>

                {props.children}

            </main>

        </Aux>

    );

}

export default Layout;

## Layout.css

.Content{

    margin-top: 16px;

}

**FOR CONTENT:**

## PizzaBuilder.js

NOTE: AUX FOR WRAPPER. THIS COMPONENT MANAGES STATES AND PASS FUNCTION TO OTHERS STATELESS COMPONENTS.

import React,{useState} from 'react';

import Aux from '../../hoc/Auxilliary';

import Pizza from '../../components/Pizza/Pizza';

import BuildControls from '../../components/Pizza/BuildControls/BuildControls';

const PizzaBuilder = () => {

    const [state, setstate] = useState({

        ingredients: {

            salad: 0,

            bacon: 0,

            cheese: 0,

            meat: 0

        },

    });

    return(

        <Aux>

             <Pizza ingredients={state.ingredients}/>

             <BuildControls />

        </Aux>

    );

}

export default PizzaBuilder;

## Pizza.js

NOTE: IT WILL MANAGES PIZZA INGREDIENTS

import React from 'react';

import './Pizza.css';

import PizzaIngredients from './PizzaIngredients/PizzaIngredients';

const Pizza = (props) => {

    let transfromedingredients = Object.keys(props.ingredients)

    .map(igKey => {

        return [...Array(props.ingredients[igKey])].map((\_, i) => {

            return <PizzaIngredients key={igKey + i} type={igKey} />

        }) //[] of two elemenst

    })

    .reduce((arr, el) => { // convert aboves arrays into one array

        return arr.concat(el)

    },[]);

    console.log(transfromedingredients);

    if(transfromedingredients.length === 0){

        transfromedingredients = <h4>Please add some ingredients</h4>

    }

    return(

        <div className='Pizza'>

        <PizzaIngredients type="bread-top" />

        {transfromedingredients}

        <PizzaIngredients type="bread-bottom" />

        </div>

    );

}

export default Pizza;

## Pizza.css

.Pizza {

    width: 100%;

    margin:auto;

    height: 250px;

    overflow: scroll;

    text-align: center;

    font-weight: bold;

    font-size: 1.2rem;

}

@media (min-width: 500px) and (min-height: 400px){

    .Pizza{

        width: 350px;

        height: 300px;

    }

}

@media (min-width: 500px) and (min-height: 401px){

    .Pizza{

        width: 450px;

        height: 400px;

    }

}

@media (min-width: 1000px) and (min-height: 700px){

    .Pizza{

        width: 700px;

        height: 600px;

    }

}

## PizzaIngredients.js

NOTE: CREATE PIZZAINGREDIENTS FOLDER AND CREATE THESE FILES INSIDE IT. SWTICH CASE IS USED TO ADD AND REMOVE INGREDIENTS AS PER INGREDIENTS NAME. PROPSTYPES IS USED TO MAKE EASIER FOR TEAM TO LOOK WHICH PROPS IS PASSED IN THIS FILE.

import React from 'react';

import './PizzaIngredients.css';

import PropTypes from 'prop-types';

const PizzaIngredients = (props) =>{

    let ingredients = null;

        switch(props.type) {

            case('bread-bottom'):

                ingredients = <div className='BreadBottom'></div>;

                break;

            case('bread-top'):

                ingredients = (

                    <div className='BreadTop'>

                        <div className='Seeds1'></div>

                        <div   className='Seeds2'></div>

                    </div>

                );

                break;

            case('bread-bottom'):

                ingredients = <div className='BreadBottom'></div>;

                break;

            case('meat'):

                ingredients = <div className='Meat'></div>;

                break;

            case('cheese'):

                ingredients = <div className='Cheese'></div>;

                break;

            case('bacon'):

                ingredients = <div className='Bacon'></div>;

                break;

            case('salad'):

                ingredients = <div className='Salad'></div>;

                break;

        }

    return ingredients;

}

PizzaIngredients.propTypes = {

    type: PropTypes.string.isRequired

}

export default PizzaIngredients;

## PizzaIngredients.css

.BreadBottom {

    height: 13%;

    width: 80%;

    background: linear-gradient(#F08E4A, #e27b36);

    border-radius: 0 0 30px 30px;

    box-shadow: inset -15px 0 #c15711;

    margin: 2% auto;

}

.BreadTop {

    height: 20%;

    width: 80%;

    background: linear-gradient(#bc581e, #e27b36);

    border-radius: 50% 50% 0 0;

    box-shadow: inset -15px 0 #c15711;

    margin: 2% auto;

    position: relative;

}

.Seeds1 {

    width: 10%;

    height: 15%;

    position: absolute;

    background-color: white;

    left: 30%;

    top: 50%;

    border-radius: 40%;

    transform: rotate(-20deg);

    box-shadow: inset -2px -3px #c9c9c9;

}

.Seeds1:after {

    content: "";

    width: 100%;

    height: 100%;

    position: absolute;

    background-color: white;

    left: -170%;

    top: -260%;

    border-radius: 40%;

    transform: rotate(60deg);

    box-shadow: inset -1px 2px #c9c9c9;

  }

.Seeds1:before {

    content: "";

    width: 100%;

    height: 100%;

    position: absolute;

    background-color: white;

    left: 180%;

    top: -50%;

    border-radius: 40%;

    transform: rotate(60deg);

    box-shadow: inset -1px -3px #c9c9c9;

  }

  .Seeds2 {

    width: 10%;

    height: 15%;

    position: absolute;

    background-color: white;

    left: 64%;

    top: 50%;

    border-radius: 40%;

    transform: rotate(10deg);

    box-shadow: inset -3px 0 #c9c9c9;

  }

  .Seeds2:before {

    content: "";

    width: 100%;

    height: 100%;

    position: absolute;

    background-color: white;

    left: 150%;

    top: -130%;

    border-radius: 40%;

    transform: rotate(90deg);

    box-shadow: inset 1px 3px #c9c9c9;

  }

.Meat {

    width: 80%;

    height: 8%;

    background: linear-gradient(#7f3608, #702e05);

    margin: 2% auto;

    border-radius: 15px;

}

.Cheese {

    width: 90%;

    height: 4.5%;

    margin: 2% auto;

    background: linear-gradient(#f4d004, #d6bb22);

    border-radius: 20px;

}

.Salad {

    width: 85%;

    height: 7%;

    margin: 2% auto;

    background: linear-gradient(#228c1d, #91ce50);

    border-radius: 20px;

}

.Bacon {

    width: 80%;

    height: 3%;

    background: linear-gradient(#bf3813, #c45e38);

    margin: 2% auto;

}

NOTE: IN PIZZABUILDER.JS, WE HAVE

return(

        <Aux>

             <Pizza ingredients={state.ingredients}/>

             <BuildControls/>

        </Aux>

    );

## BuildControls.js

import React from 'react';

import './BuildControls.css';

import BuildControl from './BuildControl/BuildControl';

const controls = [

    {label: 'Salad', type: 'salad'},

    {label: 'Bacon', type: 'bacon'},

    {label: 'Cheese', type: 'cheese'},

    {label: 'Meat', type: 'meat'},

];

const BuildControls = (props) => (

    <div className='BuildControls'>

         { controls.map(ctrl =>(

             <BuildControl

             key={ctrl.label}

             label={ctrl.label}

             />

         ))}

    </div>

);

export default BuildControls;

## BuildControl.css

.BuildControls {

    width: 100%;

    display: flex;

    flex-flow: column;

    align-items: center;

    box-shadow: 0 2px 1px #ccc;

    margin: auto;

    padding: 10px 0;

}

## BuildControl.js

NOTE: BUTTON AND LABEL ARE HERE

import React from 'react';

import './BuildControl.css';

const BuildControl = (props) => {

    return(

        <div className='BuildControl'>

            <div className='Label'>{props.label}</div>

            <button

             className='Less'

            >Less</button>

            <button

            className='More'

            >More</button>

        </div>

    );

}

export default BuildControl

BuildControl.css

.BuildControl {

    display: flex;

    justify-content: space-between;

    align-items: center;

    margin: 5px 0;

}

.BuildControl button{

    display: block;

    font: inherit;

    padding: 5px;

    margin: 0 5px;

    width: 80px;

    border: 1px solid #AA6817;

    cursor: pointer;

    outline: none;

}

# ADD AND REMOVE BUTTON WITH DISABLE LESS BUTTON:

## PizzaBuilder.js

import React,{useState} from 'react';

import Aux from '../../hoc/Auxilliary';

import Pizza from '../../components/Pizza/Pizza';

import BuildControls from '../../components/Pizza/BuildControls/BuildControls';

const INGREDIENTS\_PRICES = {

    salad:0.5,

    cheese: 0.4,

    meat: 1,

    bacon: 2,

}

const PizzaBuilder = () => {

    const [state, setstate] = useState({

        ingredients: {

            salad: 0,

            bacon: 0,

            cheese: 0,

            meat: 0

        },

        totalPrice: 4,

    });

    const addIngredientsHandler = (type) => {

        const oldCount = state.ingredients[type];

        const updatedCount = oldCount +1;

        const updatedIngredients ={

            ...state.ingredients

        };

        updatedIngredients[type] =updatedCount;

        const priceAddition = INGREDIENTS\_PRICES[type];

        const oldPrice= state.totalPrice;

        const newPrice = oldPrice + priceAddition;

        setstate({ingredients: updatedIngredients, totalPrice: newPrice})

    }

    const removeIngredientsHandler = (type) => {

        const oldCount = state.ingredients[type];

        if( oldCount <= 0) {

            return;

        }

        const updatedCount = oldCount -1;

        const updatedIngredients ={

            ...state.ingredients

        };

        updatedIngredients[type] =updatedCount;

        const priceDeduction = INGREDIENTS\_PRICES[type];

        const oldPrice= state.totalPrice;

        const newPrice = oldPrice - priceDeduction;

        setstate({ingredients: updatedIngredients, totalPrice: newPrice})

    }

    const disableInfo = {

        ...state.ingredients

    }

    for( let key in disableInfo){

        disableInfo[key] = disableInfo[key] <=0

    }//{salad: true, meat:false, ...}

    return(

        <Aux>

             <Pizza ingredients={state.ingredients}/>

             <BuildControls

             removeIngredient= {removeIngredientsHandler}

             moreIngredient={addIngredientsHandler}

             disabled={disableInfo}

             />

        </Aux>

    );

}

export default PizzaBuilder;

## BuildControls.js

import React from 'react';

import './BuildControls.css';

import BuildControl from './BuildControl/BuildControl';

const controls = [

    {label: 'Salad', type: 'salad'},

    {label: 'Bacon', type: 'bacon'},

    {label: 'Cheese', type: 'cheese'},

    {label: 'Meat', type: 'meat'},

];

const BuildControls = (props) => (

    <div className='BuildControls'>

         { controls.map(ctrl =>(

             <BuildControl

             key={ctrl.label}

             label={ctrl.label}

            //  type={ctrl.type}

            removed={() => props.removeIngredient(ctrl.type)}

            //kun type chai add garne vanera type ctrl props bata pathako

             added={() => props.moreIngredient(ctrl.type)}

             disabled ={props.disabled[ctrl.type]}

             />

         ))}

    </div>

);

export default BuildControls;

## BuildControl.js

import React from 'react';

import './BuildControl.css';

const BuildControl = (props) => {

    return(

        <div className='BuildControl'>

            <div className='Label'>{props.label}</div>

            <button

             className='Less'

            onClick={props.removed}

            disabled={props.disabled}

            >Less</button>

            <button

            className='More'

            onClick={props.added}

            >More</button>

        </div>

    );

}

export default BuildControl

# ADDING PRICE ON BASIS OF ADDITION OF TYPES OF INGREDIENTS:

## PizzaBuilder.js

return(

        <Aux>

             <Pizza ingredients={state.ingredients}/>

             <BuildControls

             removeIngredient= {removeIngredientsHandler}

             moreIngredient={addIngredientsHandler}

             disabled={disableInfo}

             price={state.totalPrice}

             />

        </Aux>

    );

## BuildControls.js

NOTE: ToFixed(2) MEANS DECIMAL UPTO 2 DIGITS.

const BuildControls = (props) => (

    <div className='BuildControls'>

    <p>Current Price: <strong>{props.price.toFixed(2)}</strong></p>

         { controls.map(ctrl =>(

             <BuildControl

             key={ctrl.label}

             label={ctrl.label}

# ORDER BUTTON SHOWN ON CONDITION:

## PizzaBuilder.js

import React,{useState} from 'react';

import Aux from '../../hoc/Auxilliary';

import Pizza from '../../components/Pizza/Pizza';

import BuildControls from '../../components/Pizza/BuildControls/BuildControls';

import Modal from '../../components/UI/Modal/Modal';

import OrderSummary from '../../components/Pizza/OrderSummary/OrderSummary';;

const INGREDIENTS\_PRICES = {

    salad:0.5,

    cheese: 0.4,

    meat: 1,

    bacon: 2,

}

const PizzaBuilder = () => {

    const [state, setstate] = useState({

        ingredients: {

            salad: 0,

            bacon: 0,

            cheese: 0,

            meat: 0

        },

        totalPrice: 4,

    });

    const [purchases, setpurchases] = useState({

        purchasable: false,

        purchasing: false

    })

    function updatePurchaseState(ingredients) {

        const sum = Object.keys( ingredients )

        .map( igKey => {

            return ingredients[igKey];

        } )

        .reduce( ( sum, el ) => {

            return sum + el;

        }, 0 );

        setpurchases( { purchasable: sum > 0 } );

    }

    const addIngredientsHandler = (type) => {

        const oldCount = state.ingredients[type];

        const updatedCount = oldCount +1;

        const updatedIngredients ={

            ...state.ingredients

        };

        updatedIngredients[type] =updatedCount;

        const priceAddition = INGREDIENTS\_PRICES[type];

        const oldPrice= state.totalPrice;

        const newPrice = oldPrice + priceAddition;

        setstate({ingredients: updatedIngredients, totalPrice: newPrice});

        updatePurchaseState(updatedIngredients);

    }

    const removeIngredientsHandler = (type) => {

        const oldCount = state.ingredients[type];

        if( oldCount <= 0) {

            return;

        }

        const updatedCount = oldCount -1;

        const updatedIngredients ={

            ...state.ingredients

        };

        updatedIngredients[type] =updatedCount;

        const priceDeduction = INGREDIENTS\_PRICES[type];

        const oldPrice= state.totalPrice;

        const newPrice = oldPrice - priceDeduction;

        setstate({ingredients: updatedIngredients, totalPrice: newPrice});

        // PurchasesUpdate();

    }

    const disableInfo = {

        ...state.ingredients

    }

    for( let key in disableInfo){

        disableInfo[key] = disableInfo[key] <=0

    }//{salad: true, meat:false, ...}

    return(

        <Aux>

            <Modal show={purchases.purchasing}>

                <OrderSummary ingredients={state.ingredients}/>

            </Modal>

             <Pizza ingredients={state.ingredients}/>

             <BuildControls

             removeIngredient= {removeIngredientsHandler}

             moreIngredient={addIngredientsHandler}

             disabled={disableInfo}

             purchasabled = {purchases.purchasable}

             price={state.totalPrice}

             />

        </Aux>

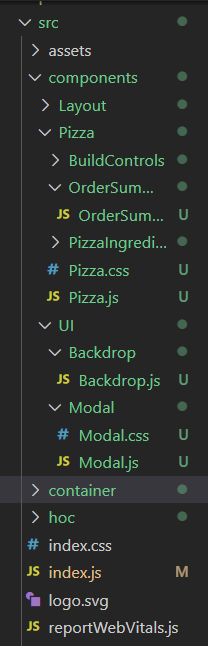
    );

}

export default PizzaBuilder;

# CREATING ORDER SUMMARY MODAL:

Folder structure:



## Modal.js:

**NOTE: Modal is used as wrapper. {props.children} will render anything passed.**

import React from 'react';

import './Modal.css';

const Modal = (props) => (

    <div className='Modal'>

        {props.children}

        {/\* children can be anything even your components  \*/}

    </div>

);

export default Modal;

## Modal.css:

.Modal {

    position: fixed;

    z-index: 500;

    background-color: white;

    width: 70%;

    border: 1px solid #ccc;

    box-shadow: 1px 1px 1px black;

    padding: 16px;

    left: 15%;

    top: 30%;

    box-sizing: border-box;

    transition: all 0.3s ease-out;

}

@media (min-width: 600px) {

    .Modal {

        width: 500px;

        left: calc(50% - 250px);

    }

}

## PizzaBuilder.js

return(

        <Aux>

            <Modal>

                <OrderSummary ingredients={state.ingredients}/>

            </Modal>

             <Pizza ingredients={state.ingredients}/>

             <BuildControls

             removeIngredient= {removeIngredientsHandler}

             moreIngredient={addIngredientsHandler}

             disabled={disableInfo}

            //  purchasable = {state.purchasable}

             price={state.totalPrice}

             />

        </Aux>

    );

## OrderSummary.js

NOTE: ordersummary will render list of ingredients added in burger. ingredientsSummary will convert ingredients objects passed as state into array through map in this components inside JSX code list <li></li>; igKey is unique for every ingredients.

import React from 'react';

 import Aux from '../../../hoc/Auxilliary';

const OrderSummary = (props) => {

    //AS we know we have in object format not in array format so we need to transformation into array.

    const ingredientsSummary = Object.keys(props.ingredients)

    .map(igKey => {

        return (

        <li>

        <span>{igKey}</span>

        :{props.ingredients[igKey]}

        </li>

        );

    })

    return (

        <Aux>

        <h3>YOUR ORDER</h3>

        <p>ingredients list:</p>

        <ul>{ingredientsSummary}</ul>

        <p>COntinue to check out?</p>

        </Aux>

    );

};

export default OrderSummary;

# ADDING ANIMATION:

## PizzaBuilder.js

const [purchases, setpurchases] = useState({

        purchasable: false,

        purchasing: false

    })

 purchaseHandler() {

        setstate({purchasing: true});

    }

 <BuildControls

             removeIngredient= {removeIngredientsHandler}

             moreIngredient={addIngredientsHandler}

             disabled={disableInfo}

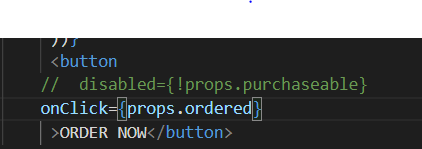
             ordered = {purchaseHandler}

             purchasabled = {purchases.purchasable}

             price={state.totalPrice}

             />

## BuildControls.js



# IMPLEMENT BACKDROP

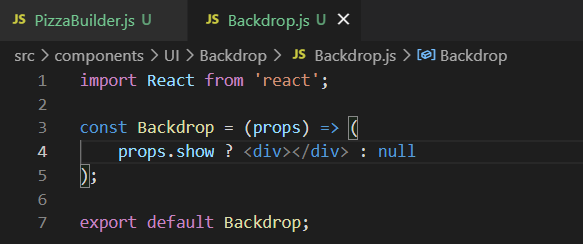
NOTE: props.show is passed from PizzaBuilder.

<Modal show={purchases.purchasing}>

                <OrderSummary ingredients={state.ingredients}/>

            </Modal>

INTO this component. IF ELSE CONDITION IS USED.



BACKDROP.CSS

.Backdrop {

    width: 100%;

    height: 100%;

    position: fixed;

    z-index: 100;

    left: 0;

    top: 0;

    background-color: rgba(0, 0, 0, 0.5);

}

## Backdrop.js

NOTE: Hamile modal card ko backside tira click garyo vane clicked vane props call garchum.

import React from 'react';

import './Backdrop.css';

const Backdrop = (props) => (

    props.show ? <div className='Backdrop' onClick={props.clicked}></div> : null

);

export default Backdrop;

## Modal.js

NOTE: hamile backdrop import garim yesma kina vane.. jabo modal dekhauxa teti bela hamile cancel garnako lagi backdrop chahinxa. Hamile show ra click backdrop ma as props pathiyim tara yini haru pani hamile Pizzabuilder vane component bata hamile as props leyko ho.

import Aux from '../../../hoc/Auxilliary';

import Backdrop from '../Backdrop/Backdrop';

const Modal = (props) => (

    <Aux>

        <Backdrop show={props.show} clicked={props.modalClosed} />

        <div className='Modal'

        style ={{

        transform: props.show ? 'translateY(0)' : 'translateY(-100vh)',

        opacity: props.show? '1' : '0'}}>

        {props.children}

        {/\* children can be anything even your components  \*/}

    </div>

    </Aux>

);

export default Modal;

## Pizzabuilder.js

NOTE: modalClosed ne hamile purchasedCancelHandler function lai as props pathayeko chum.

 return(

        <Aux>

            <Modal show={purchases.purchasing}

 modalClosed={purchasedCancelHandler}>

                <OrderSummary ingredients={state.ingredients}/>

            </Modal>

             <Pizza ingredients={state.ingredients}/>

             <BuildControls

             removeIngredient= {removeIngredientsHandler}

             moreIngredient={addIngredientsHandler}

             disabled={disableInfo}

             ordered = {purchaseHandler}

             purchasabled = {purchases.purchasable}

             price={state.totalPrice}

             />

        </Aux>

    );

## purchasedCancelHandler function.

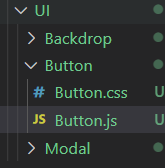
const purchasedCancelHandler = () => {

        setpurchases({purchasing:false})

    }

# CREATING BUTTON UI

FOLDER STRUCTURE:



## Button.js

import React from 'react';

import classes from './Button.css';

const button = (props) => (

    <button

    className={['Button',[props.btnType]].join(' ')}

    onClick={props.clicked}>{props.children}</button>

);

export default button;

## Button.css

.Button {

    background-color: transparent;

    border: none;

    color: white;

    outline: none;

    cursor: pointer;

    font: inherit;

    padding: 10px;

    margin: 10px;

    font-weight: bold;

}

.Button:first-of-type {

    margin-left: 0;

    padding-left: 0;

}

.Success {

    color: #5C9210;

}

.Danger {

    color: #944317;

}

## OrderSummary.js

import React from 'react';

import Aux from '../../../hoc/Auxilliary';

import Button from '../../UI/Button/Button';

const OrderSummary = (props) => {

    //AS we know we have in object format not in array format so we need to transformation into array.

    const ingredientsSummary = Object.keys(props.ingredients)

    .map(igKey => {

        return (

        <li>

        <span>{igKey}</span>

        :{props.ingredients[igKey]}

        </li>

        );

    })

    return (

        <Aux>

        <h3>YOUR ORDER</h3>

        <p>ingredients list:</p>

        <ul>{ingredientsSummary}</ul>

        <p>Continue to check out?</p>

        <Button btnType="Danger" clicked={props.purchaseCancel}>CANCEL</Button>

        <Button btnType="Success" clicked={props.purchaseContinue}>

CONTINUE</Button>

        </Aux>

    );

};

export default OrderSummary;

## PizzaBuilder.js

## CANCEL AND CONTINUE FUNCTIONS:

const purchasedCancelHandler = () => {

        setpurchases({purchasing:false})

    }

    const purcharseContinueHandler = () => {

        alert('continue')

    }

<OrderSummary

                ingredients={state.ingredients}

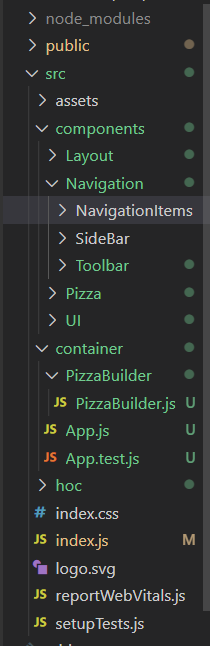
                purchaseCancel={purchasedCancelHandler}

                purchaseContinue ={purcharseContinueHandler}

                />

# ADDING TOOLBAR:

FOLDER STRUCTURE:



## TOOLBAR.JS

import React from 'react';

import './Toolbar.css'

const Toolbar = (props) => (

    <header className='Toolbar'>

        <div>MENU</div>

        <div>LOGO</div>

        <nav>

            ....

        </nav>

    </header>

);

export default Toolbar;

## Toolbar.css

.Toolbar {

    height: 56px;

    width: 100%;

    position: fixed;

    top: 0;

    left: 0;

    background-color: #9e7f61;

    /\* if you don't know about flex please look it  \*/

    display: flex;

    justify-content: space-between;

    align-items: center;

    padding: 0 20px;

    box-sizing: border-box;

    z-index: 90;

}

.Toolbar nav {

    height: 100%;

}

.Logo {

    height: 80%;

}

@media (max-width: 499px) {

    .DesktopOnly {

        display: none;

    }

}

## LAYOUT.JS

import React from 'react';

import Aux from '../../hoc/Auxilliary';

import Toolbar from '../Navigation/Toolbar/Toolbar';

import  './Layout.css';

const Layout = (props) => {

    return(

        <Aux>

            <Toolbar />

            <main className='Content'>

                {props.children}

            </main>

## LAYOUT.CSS

.Content{

    margin-top: 72px;

}

# ADDING LOGO IN NAVBAR

NOTE: Create folder assets and images inside it then copy your images inside it.

## Logo.js

import React from 'react';

import  './Logo.css';

import burgerLogo from '../../assets/images/burger-logo.png';

const logo =(props) => (

    <div className='Logo'>

        <img src={burgerLogo} alt="Food receipe" />

    </div>

);

export default logo;

## Logo.css

.Logo{

    background-color: wheat;

    padding:  8px;

    height: 80%;

}

.Logo img{

    height: 100%;

}

## Toolbar.js

import React from 'react';

import './Toolbar.css';

import Logo  from '../../Logo/Logo';

const Toolbar = (props) => (

    <header className='Toolbar'>

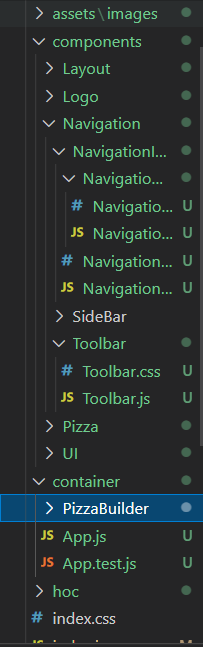
        <div>MENU</div>

        <Logo />

        <nav>

# ADDING REUSABLE NAVIGATION ITEMS:

Folder structure:



## Layout.js

NOTE: this layout is for whole app layout.

import React from 'react';

import Aux from '../../hoc/Auxilliary';

import Toolbar from '../Navigation/Toolbar/Toolbar';

import  './Layout.css';

const Layout = (props) => {

    return(

        <Aux>

            <Toolbar />

            <main className='Content'>

                {props.children}

            </main>

        </Aux>

    );

}

export default Layout;

## Toolbar.js

NOTE: It is used for heading section of app layout.

import React from 'react';

import './Toolbar.css';

import Logo  from '../../Logo/Logo';

import NavigationItems from '../NavigationItems/NavigationItems';

const Toolbar = (props) => (

    <header className='Toolbar'>

        <div>MENU</div>

        <Logo />

        <nav>

            <NavigationItems />

        </nav>

    </header>

);

export default Toolbar;

## NavigationItems.js

NOTE: active -- active is boolean so it can be passed like this

import React from 'react';

import './NavigationItems.css';

import NavigationItem from './NavigationItem/NavigationItem';

const NavigationItems = () => (

    <ul className='NavigationItems'>

        <NavigationItem link="/" active>

        {/\* //active is boolean so it can be passed like this \*/}

            Burger Builder

        </NavigationItem>

        <NavigationItem link="/" >

            Check out

        </NavigationItem>

    </ul>

);

export default NavigationItems;

## NavigationItems.css

.NavigationItems {

    margin: 0;

    padding: 0;

    list-style: none;

    display: flex;

    flex-flow: column;

    align-items: center;

    height: 100%;

}

## NavigationItem.js

NOTE: if item is active then it will call class active of css.

import React from 'react';

import './NavigationItem.css';

const NavigationItem = (props) => (

    <li className='NavigationItem'>

    <a

    href={props.link}

    className={props.active ? 'active' : null}

    >

    {props.children}

    </a>

    </li>

);

export default NavigationItem;

## NavigationItem.css

.NavigationItem {

    margin: 10px 0;

    box-sizing: border-box;

    display: block;

    width: 100%;

}

.NavigationItem a {

    color: #8F5C2C;

    text-decoration: none;

    width: 100%;

    box-sizing: border-box;

    display: block;

}

.NavigationItem a:hover,

.NavigationItem a:active,

.NavigationItem a.active {

    color: #40A4C8;

}

# CREATING RESPONSIVE SIDE BAR:

Folder Structure:



# SideDrawer.js

import React from 'react';

import './SideDrawer.css'

import Logo from '../../Logo/Logo';

import NavigationItems from '../NavigationItems/NavigationItems';

const SideDrawer = (props) => {

    //...

    return (

        <div className='SideDrawer'>

            <Logo />

            <nav>

                <NavigationItems />

            </nav>

        </div>

        );

}

export default SideDrawer;

# SideDrawer.css

.SideDrawer {

    position: fixed;

    width: 280px;

    max-width: 70%;

    height: 100%;

    left: 0;

    top: 0;

    z-index: 200;

    background-color: white;

    padding: 32px 16px;

    box-sizing: border-box;

    transition: transform 0.3s ease-out;

}

@media (min-width: 500px) {

    .SideDrawer {

        display: none;

    }

}

.Open {

    transform: translateX(0);

}

.Close {

    transform: translateX(-100%);

}

.Logo {

    height: 11%;

    margin-bottom: 32px;

}

# Layout.js

import React from 'react';

import Aux from '../../hoc/Auxilliary';

import Toolbar from '../Navigation/Toolbar/Toolbar';

import  './Layout.css';

import SideDrawer from '../Navigation/SideBar/SideDrawer';

const Layout = (props) => {

    return(

        <Aux>

            <Toolbar />

            <SideDrawer />

            <main className='Content'>

                {props.children}

            </main>

        </Aux>

    );

}

export default Layout;

# ADJUSTMENTS OF RESPONSIVE NAVBAR:

ADD CODES IN:

## NagivationItem.css:

@media (min-width: 500px) {

    .NavigationItem {

        margin: 0;

        display: flex;

        height: 100%;

        width: auto;

        align-items: center;

    }

    .NavigationItem a {

        color: white;

        height: 100%;

        padding: 16px 10px;

        border-bottom: 4px solid transparent;

    }

    .NavigationItem a:hover,

    .NavigationItem a:active,

    .NavigationItem a.active {

        background-color: #8F5C2C;

        border-bottom: 4px solid #40A4C8;

        color: white;

    }

}

## NavigationItems.css

@media (min-width: 500px) {

    .NavigationItems {

        flex-flow: row;

    }

}

# NAVIGATIONS ITEMS UNVISIBLE IN MOBILE VIEW.

## Toolbar.css

@media (max-width: 499px) {

    .DesktopOnly {

        display: none;

    }

}

## Toolbar.js

import React from 'react';

import './Toolbar.css';

import Logo  from '../../Logo/Logo';

import NavigationItems from '../NavigationItems/NavigationItems';

const Toolbar = (props) => (

    <header className='Toolbar'>

        <div>MENU</div>

        <Logo/>

        <nav className='DesktopOnly '>

            <NavigationItems />

        </nav>

    </header>

);

export default Toolbar;

# REUSING BACKDROP IN NAVBAR

## TAKE REFERENCE FROM MODAL.js

import React from 'react';

import './Modal.css';

import Aux from '../../../hoc/Auxilliary';

import Backdrop from '../Backdrop/Backdrop';

const Modal = (props) => (

    <Aux>

        <Backdrop show={props.show} clicked={props.modalClosed} />

        <div className='Modal'

        style ={{

        transform: props.show ? 'translateY(0)' : 'translateY(-100vh)',

        opacity: props.show? '1' : '0'}}>

        {props.children}

        {/\* children can be anything even your components  \*/}

    </div>

    </Aux>

);

export default Modal;

## SideDrawer.js

NOTE: HIGHLIGHTED PART IS USED FOR CSS CLASS TO OPEN AND CLOSE THE SIDEDRAWER WHEN IT IS CLICKED IN BACKDROP.

import React from 'react';

import './SideDrawer.css'

import Logo from '../../Logo/Logo';

import NavigationItems from '../NavigationItems/NavigationItems';

import Backdrop from '../../UI/Backdrop/Backdrop';

import Aux

 from '../../../hoc/Auxilliary';

const SideDrawer = (props) => {

    let attachedClasses = ['SideDrawer','Close'];

    if(props.open) {

        attachedClasses = ['SideDrawer','Open'];

    }

    return (

        <Aux>

        <Backdrop show={props.open} clicked={props.closed} />

        <div className={attachedClasses.join(' ')}>

            <Logo />

             <nav>

                <NavigationItems />

            </nav>

        </div>

        </Aux>

        );

}

export default SideDrawer;

## SideDrawer.css

.SideDrawer {

    position: fixed;

    width: 280px;

    max-width: 70%;

    height: 100%;

    left: 0;

    top: 0;

    z-index: 200;

    background-color: white;

    padding: 32px 16px;

    box-sizing: border-box;

    transition: transform 0.3s ease-out;

}

@media (min-width: 500px) {

    .SideDrawer {

        display: none;

    }

}

.Open {

    transform: translateX(0);

}

.Close {

    transform: translateX(-100%);

}

.Logo {

    height: 11%;

    margin-bottom: 32px;

}

## Layout.js

import React,{useState} from 'react';

import Aux from '../../hoc/Auxilliary';

import Toolbar from '../Navigation/Toolbar/Toolbar';

import  './Layout.css';

import SideDrawer from '../Navigation/SideBar/SideDrawer';

const Layout = (props) => {

    const [state, setstate] = useState({

        showSideDrawer: true,

    });

    const SideDrawerCLosedHandler = () => {

        setstate({showSideDrawer: false})

    }

    return(

        <Aux>

            <Toolbar />

            <SideDrawer

open={state.showSideDrawer}

 closed={SideDrawerCLosedHandler} />

            <main className='Content'>

                {props.children}

            </main>

        </Aux>

    );

}

export default Layout;

# Adding SideDrawer toggle button:

When clicking on the button, sidedrawer will apper.

## DrawerToggle.js

import React from 'react';

const DrawerToggle = (props) => (

    <div onClick={props.clicked} className=”DrawerToggle”>

<div></div>

<div></div>

<div></div>

</div>

);

export default DrawerToggle;

## DrawerToggle.css

.DrawerToggle {

    width: 40px;

    height: 100%;

    display: flex;

    flex-flow: column;

    justify-content: space-around;

    align-items: center;

    padding: 10px 0;

    box-sizing: border-box;

    cursor: pointer;

}

.DrawerToggle div {

    width: 90%;

    height: 3px;

    background-color: white;

}

@media (min-width: 500px) {

    .DrawerToggle {

        display: none;

    }

}

## Toolbar.js

import React from 'react';

import './Toolbar.css';

import Logo  from '../../Logo/Logo';

import NavigationItems from '../NavigationItems/NavigationItems';

import DrawerToggle from '../SideBar/DrawerToggle/DrawerToggle';

const Toolbar = (props) => (

    <header className='Toolbar'>

        <DrawerToggle clicked={props.drawerToggleClicked}/>

        <Logo/>

        <nav className='DesktopOnly '>

            <NavigationItems />

        </nav>

    </header>

);

export default Toolbar;

## Layout.js

import React,{useState} from 'react';

import Aux from '../../hoc/Auxilliary';

import Toolbar from '../Navigation/Toolbar/Toolbar';

import  './Layout.css';

import SideDrawer from '../Navigation/SideBar/SideDrawer';

const Layout = (props) => {

    const [state, setstate] = useState({

        showSideDrawer: true,

    });

    const SideDrawerCLosedHandler = () => {

        setstate({showSideDrawer: false})

    }

    const sideDrawerToggleClicked = () => {

        // setstate({

        //     showSideDrawer: !state.showSideDrawer

        // })

        // THIS WILL THROW AN ERROR

        setstate((prevState) => {

            return {showSideDrawer: !prevState.showSideDrawer};

        })

    }

    return(

        <Aux>

            <Toolbar drawerToggleClicked={sideDrawerToggleClicked} />

            <SideDrawer open={state.showSideDrawer} closed={SideDrawerCLosedHandler} />

            <main className='Content'>

                {props.children}

            </main>

        </Aux>

    );

}

export default Layout;