**MINI-PROJECT – II**

**(2021-2022)**

****

**Institute of Engineering & Technology**

**Mid Term Report: Burger App**

**Submitted By: Submitted to:**

Mansi Gupta(181500371) Faculty : Mr. Akash Choudhary

Gaurav Sharma(181500235)

Vishal Singh(181500809)

Harsh yadav(181500250)

**DECLARATION**

We hereby declare that the work which is being presented in the MINI PROJECT **“Title…BURGER APP…”, will be fulfilled** of the requirements for Mini Project viva voce, is an authentic record of my work carried under the supervision of **Mr. Akash Choudhary.**

Signature of Candidates:

Name of Candidate/Roll No:

MANSI GUPTA (181500371)

GAURAV SHARMA (181500235)

VISHAL SINGH (181500809)

HARSH YADAV (181500250)

Course: B-tech CSE

Year: III

Semester: 6th

**ACKNOWLEDGEMENT**

We express our sincere indebtedness towards our guide mentor. Mr. Akash Choudhary, Computer Science & Engineering.GLA University, Mathura for his invaluable guidance, suggestions, and supervision throughout the work. Without his kind patronage and guidance, the project would not have taken shape. We would also like to express our gratitude and sincere regards for his kind approval of the project, time-to-time counseling, and pieces of advice.

We would also like to thank HOD Mr. Anand Singh Jalal Department of Computer Science & Engineering.GLA University, Mathura for her expert advice and counseling from time to time.

We owe sincere thanks to all the faculty members in the department

of Computer Science & Engineering for their kind guidance and

encouragement from time to time.

Mansi Gupta(181500371)

Gaurav Sharma(181500235)

Vishal Singh(181500809)

Harsh yadav(181500250)

**Abstract**

In this we will build a app using the concept of React by using the front end and back end technologies in which Front end technology includes HTML,CSS and JavaScript and Back-end technology includes Node javaScript that we have used in our App which makes the App efficient and effective. This application has various functions to perform such as the option of adding and removing the ingredients from a burger and make your own burger as you want. So here we are going to implement one more feature along with the normal adding and removing the ingredients such as you can order that burger you make on that application. By which the customers had various options available in this application. This application can enhance the order-making feature.

|  |  |  |
| --- | --- | --- |
| S.NO | Topic | Pages No. |
|  |  |  |
| 1 | Declaration | 2 |
|  |  |  |
| 2 | Acknowledgment | 3 |
|  |  |  |
| 3 | Abstract | 4 |
|  |  |  |
| 4 | Introduction | 6 |
|  |  |  |
| 5 | System and Hardware Requirements | 6 |
|  |  |  |
| 6 | Frontend and Backend | 7 |
|  |  |  |
| 7 | Idea and Objective | 8 |
|  |  |  |
|  |  |  |
| 8 | Module Description | 9 |
| 9 | Flow Chart | 10 |
| 10 | DFD 0level ,1level | 11 |
| 11 | Codes and Screenshorts | 14 |
| 12 | BIBLIOGRAPHY | 35 |

**INDEX**

**INTRODUCTION**: -

This application has various functions to perform such as the option of adding and removing the ingredients from a burger and make your own burger as you want. So here we are going to implement one more feature along with the normal adding and removing the ingredients such as you can order that burger you make on that application. By which the customers had various options available in this application. This application can enhance the order-making feature.

**System Requirements:** -

**Supported Operating system: -**

Windows 10 Windows 8 Windows 7

**Software Required: -**

* VS Code

**Hardware Requirements: -**

**For VS Code: -**

**Intel i3 6th Gen (1.6 GHz minimum)**.

**1 GB** of RAM.

Internet Connection

Frontend and Backend: -

**Frontend**

Front-end web development, also known as client-side development is the practice of producing HTML, CSS and JavaScript, React for a website or Web Application so that a user can see and interact with them directly. The challenge associated with front end development is that the tools and techniques used to create the front end of a website change constantly and so the developer needs to constantly be aware of how the field is developing.

The objective of designing a site is to ensure that when the users open up the site, they see the information in a format that is easy to read and relevant. This is further complicated by the fact that users now use a large variety of devices with varying screen sizes and resolutions thus forcing the designer to take into consideration these aspects when designing the site. They need to ensure that their site comes up correctly in different browsers (cross-browser), different operating systems (cross-platform) and different devices (cross-device), which requires careful planning on the side of the developer.

**Backend**

It has a Node.Js in backend with implementation of Firebase for the database and hosting. The project was built using the Redux library. The user can create their own account, login and check their orders.

**Idea: -**

Idea behind this app is to create a platform for the customer to make your own burger with your own ingredients and order it.

**Objective: -**

1. The main objective of the project is to make your own burger easier.
2. It will help customers to do order on that application.

**Module Description:**

**Main activity:** The main activity consists of the set of adding and removing the ingredients as you want and then order that burger.

**Home Module:**  In this module user can select which burger he/she has to opt (veg /non-veg) further user has to select ingredients which user is going to use in burger.

**User Module:**  In this module user can login and fill in information and choices what he want and can view profile and order history.

**Admin Module:** In this module admin a view users order history, can change the ingredients as per availability in the shop and admin can enable or disable any kind of burger which is currently not available.

**Scope: -** The scope of this project is quite wide. We cannot use this application on the burger but also use in a different type of food.

**Flow Chart:**

App

Layout

Toolbar

BackDrop

{props.childern}

Side Drawer

Different Page

Burger Page

Burger control

Burger

Modal

{props.Children}

* Burger control
* Burger control
* ………..
* …………..
* Ingredients
* Ingredients
* ………………
* …………….

**States**

* Ingredients

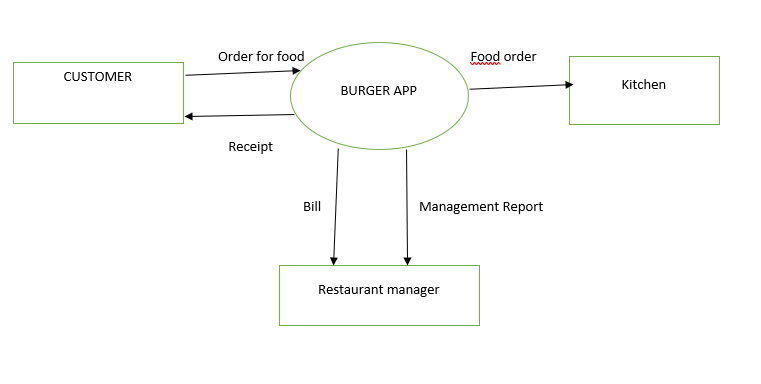
{meat,cheese ,bacon,salad}

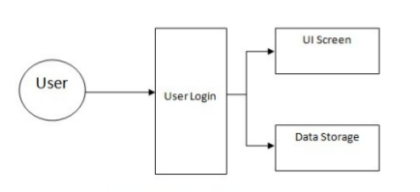
* Purchases :True
* Total Price:$\_\_\_\_

Order Button

**DFD: -** A data flow diagram (DFD) illustrates how data is processed by a system in terms ofinputs and outputs. As its name indicates its focus is on the flow of information, where data comes from, where it goes and how it gets stored.

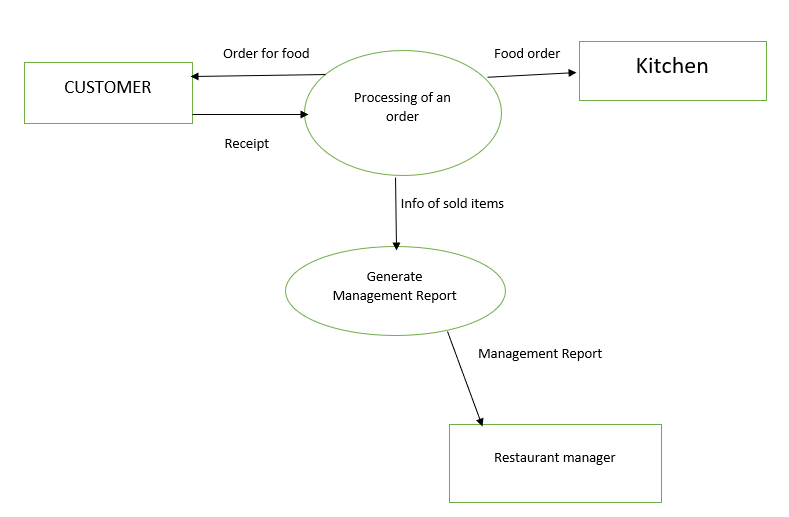
**0 Level DFD: -**DFD Level 0 is also called a Context Diagram. It’s a basic overview of the whole system orprocess being analyzed or modeled. It’s designed to be an at-a-glance view, showing the system as a single high-level process, with its relationship to external entities. It should be easily understood by a wide audience, including stakeholders, business analysts, data analysts and developers.

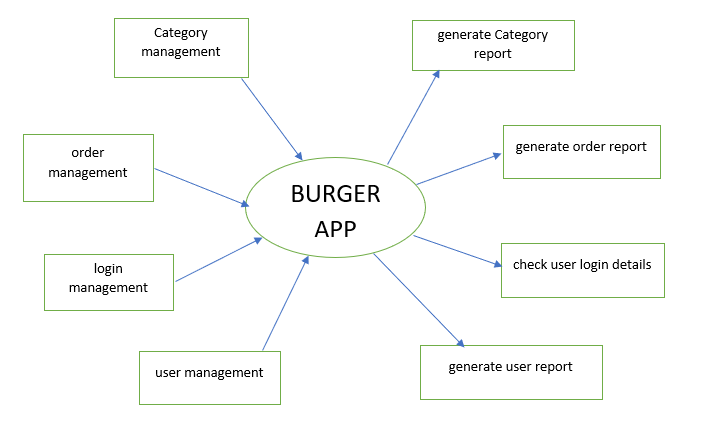




**1 level DFD: - DFD Level 1 provides a more detailed breakout of pieces of the Context**

**Level Diagram. You will highlight the main functions carried out by the system, as you breakdown the high-level process of the Context Diagram into its sub processes.**



****

**Codes and ScreenShorts**

Webpack.config.dev.js

{

        test: /\.(js|jsx)$/,

        enforce: 'pre',

        use: [

          {

            options: {

              formatter: eslintFormatter,

              eslintPath: require.resolve('eslint'),

            },

            loader: require.resolve('eslint-loader'),

          },

        ],

        include: paths.appSrc,

      },

      {

        oneOf: [

          {

            test: [/\.bmp$/, /\.gif$/, /\.jpe?g$/, /\.png$/],

            loader: require.resolve('url-loader'),

            options: {

              limit: 10000,

              name: 'static/media/[name].[hash:8].[ext]',

            },

          },

          {

            test: /\.(js|jsx)$/,

            include: paths.appSrc,

            loader: require.resolve('babel-loader'),

            options: {

              cacheDirectory: true,

            },

          },

          {

            test: /\.css$/,

            use: [

              require.resolve('style-loader'),

              {

                loader: require.resolve('css-loader'),

                options: {

                  importLoaders: 1,

                  modules: true,

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

                },

              },

              {

                loader: require.resolve('postcss-loader'),

                options: {

                  ident: 'postcss',

                  plugins: () *=>* [

                    require('postcss-flexbugs-fixes'),

                    autoprefixer({

                      browsers: [

                        '>1%',

                        'last 4 versions',

                        'Firefox ESR',

                        'not ie < 9',

                      ],

                      flexbox: 'no-2009',

                    }),

                  ],

                },

              },

            ],}

Buildcontrol.js

import React from 'react';

import classes from './BuildControls.module.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={classes.BuildControls}>

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

        {controls.map(*ctrl* *=>* (

            <*BuildControl*

                key={*ctrl*.label}

                label={*ctrl*.label}

                added={() *=>* *props*.ingredientAdded(*ctrl*.type)}

                removed={() *=>* *props*.ingredientRemoved(*ctrl*.type)}

                disabled={*props*.disabled[*ctrl*.type]} />

        ))}

        <button

            className={classes.OrderButton}

            disabled={!*props*.purchasable}

            onClick={*props*.ordered}>ORDER NOW</button>

    </div>

);

export default buildControls;

Buildercontrolmodules.css

.BuildControls {

*width*: 100%;

*background-color*: #CF8F2E;

*display*: flex;

*flex-flow*: column;

*align-items*: center;

*box-shadow*: 0 2px 1px #ccc;

*margin*: auto;

*padding*: 10px 0;

}

.OrderButton {

*background-color*: #DAD735;

*outline*: none;

*cursor*: pointer;

*border*: 1px solid #966909;

*color*: #966909;

*font-family*: inherit;

*font-size*: 1.2em;

*padding*: 15px 30px;

*box-shadow*: 2px 2px 2px #966909;

}

.OrderButton:hover, .OrderButton:active {

*background-color*: #A0DB41;

*border*: 1px solid #966909;

*color*: #966909;

}

.OrderButton:disabled {

*background-color*: #C7C6C6;

*cursor*: not-allowed;

*border*: 1px solid #ccc;

*color*: #888888;

}

.OrderButton:not(:disabled) {

*animation*: enable 0.3s linear;

}

@keyframes *enable* {

    0% {

*transform*: scale(1);

    }

    60% {

*transform*: scale(1.1);

    }

    100% {

*transform*: scale(1);

    }

}

Burgeringredients.js

import React, { Component } from 'react';

import PropTypes from 'prop-types';

import classes from './BurgerIngredient.module.css';

*class* BurgerIngredient extends Component {

    render () {

*let* ingredient = null;

        switch ( this.props.type ) {

            case ( 'bread-bottom' ):

                ingredient = <div className={classes.BreadBottom}></div>;

                break;

            case ( 'bread-top' ):

                ingredient = (

                    <div className={classes.BreadTop}>

                        <div className={classes.Seeds1}></div>

                        <div className={classes.Seeds2}></div>

                    </div>

                );

                break;

            case ( 'meat' ):

                ingredient = <div className={classes.Meat}></div>;

                break;

            case ( 'cheese' ):

                ingredient = <div className={classes.Cheese}></div>;

                break;

            case ( 'bacon' ):

                ingredient = <div className={classes.Bacon}></div>;

                break;

            case ( 'salad' ):

                ingredient = <div className={classes.Salad}></div>;

                break;

            default:

                ingredient = null;

        }

        return ingredient;

    }

}

BurgerIngredient.propTypes = {

    type: PropTypes.string.isRequired

};

export default BurgerIngredient;

Burgeringredientsmodules.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;

}

Ordersummary.js

import React from 'react';

import Aux from '../../../hoc/Aux1';

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

*const* orderSummary = ( *props* ) *=>* {

*const* ingredientSummary = *Object*.keys( *props*.ingredients )

        .map( *igKey* *=>* {

            return (

                <li key={*igKey*}>

                    <span style={{ textTransform: 'capitalize' }}>{*igKey*}</span>: {*props*.ingredients[*igKey*]}

                </li> );

        } );

    return (

        <*Aux*>

            <h3>Your Order</h3>

            <p>A delicious burger with the following ingredients:</p>

            <ul>

                {ingredientSummary}

            </ul>

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

            <p>Continue to Checkout?</p>

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

            <*Button* btnType="Success" clicked={*props*.purchaseContinued}>CONTINUE</*Button*>

        </*Aux*>

    );

};

export default orderSummary;

Burger.js

import React from 'react';

import classes from './Burger.module.css';

import BurgerIngredient from './BurgerIngredient/BurgerIngredient';

*const* burger = ( *props* ) *=>* {

*let* transformedIngredients = *Object*.keys( *props*.ingredients )

        .map( *igKey* *=>* {

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

                return <*BurgerIngredient* key={*igKey* + *i*} type={*igKey*} />;

            } );

        } )

        .reduce((*arr*, *el*) *=>* {

            return *arr*.concat(*el*)

        }, []);

    if (transformedIngredients.length === 0) {

        transformedIngredients = <p>Please start adding ingredients!</p>;

    }

    return (

        <div className={classes.Burger}>

            <*BurgerIngredient* type="bread-top" />

            {transformedIngredients}

            <*BurgerIngredient* type="bread-bottom" />

        </div>

    );

};

export default burger;

Burger.modules.css

.Burger{

*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){

    .Burger{

*width*: 350px;

*height*: 300px;

    }

}

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

    .Burger{

*width*: 450px;

*height*: 400px;

    }

}

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

    .Burger{

*width*: 700px;

*height*: 600px;

    }

}

Layout.js

import React, { Component } from 'react';

import Aux from '../../hoc/Aux1';

import classes from './Layout.module.css';

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

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

*class* Layout extends Component {

    state = {

        showSideDrawer: false

    }

    sideDrawerClosedHandler = () *=>* {

        this.setState( { showSideDrawer: false } );

    }

    sideDrawerToggleHandler = () *=>* {

        this.setState( ( *prevState* ) *=>* {

            return { showSideDrawer: !*prevState*.showSideDrawer };

        } );

    }

    render () {

        return (

            <*Aux*>

                <*Toolbar* drawerToggleClicked={this.sideDrawerToggleHandler} />

                <*SideDrawer*

                    open={this.state.showSideDrawer}

                    closed={this.sideDrawerClosedHandler} />

                <main className={classes.Content}>

                    {this.props.children}

                </main>

            </*Aux*>

        )

    }

}

export default Layout;

Layout.module.css

.Content{

*margin-top*: 60px;

}

Logo.js

import React from 'react';

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

import classes from './Logo.module.css';

*const* logo = (*props*) *=>* (

    <div className={classes.Logo} style={{height: *props*.height}}>

        <img src={burgerLogo} alt="MyBurger" />

    </div>

);

export default logo;

Logo.modules.css

.Logo {

*background-color*: white;

*padding*: 8px;

*height*: 100%;

*box-sizing*: border-box;

*border-radius*: 5px;

}

.Logo img {

*height*: 100%;

}

Aux1.js

*const* aux = (*props*) *=>* *props*.children;

 export default aux;

App.js

import React, { Component } from 'react';

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

import BurgerBuilder from './containers/BurgerBuilder/BurgerBuilder';

*class* App extends Component {

  render () {

    return (

      <div>

        <*Layout*>

          <*BurgerBuilder* />

        </*Layout*>

      </div>

    );

  }

}

export default App;

Apptest.js

import React from 'react';

import ReactDOM from 'react-dom';

import App from './App';

it('renders without crashing', () *=>* {

*const* div = document.createElement('div');

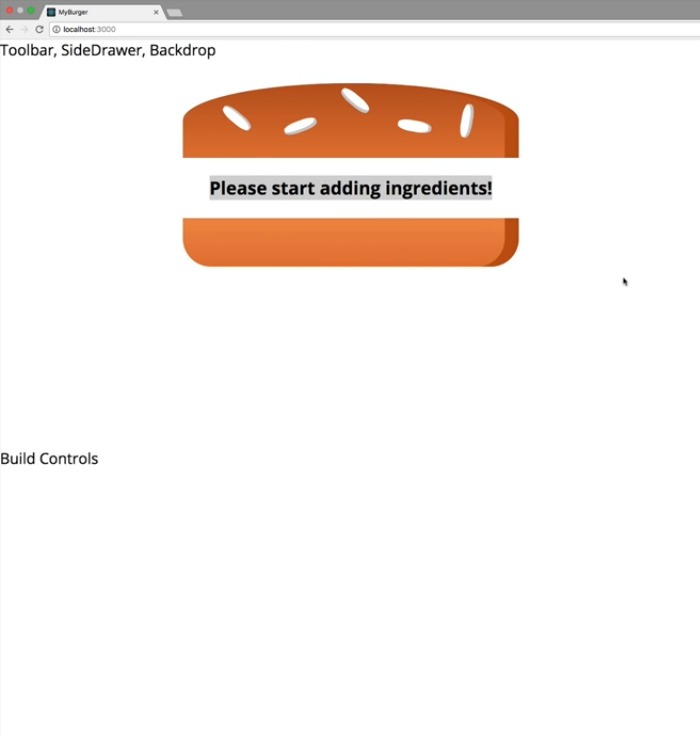
  ReactDOM.render(<*App* />, div);

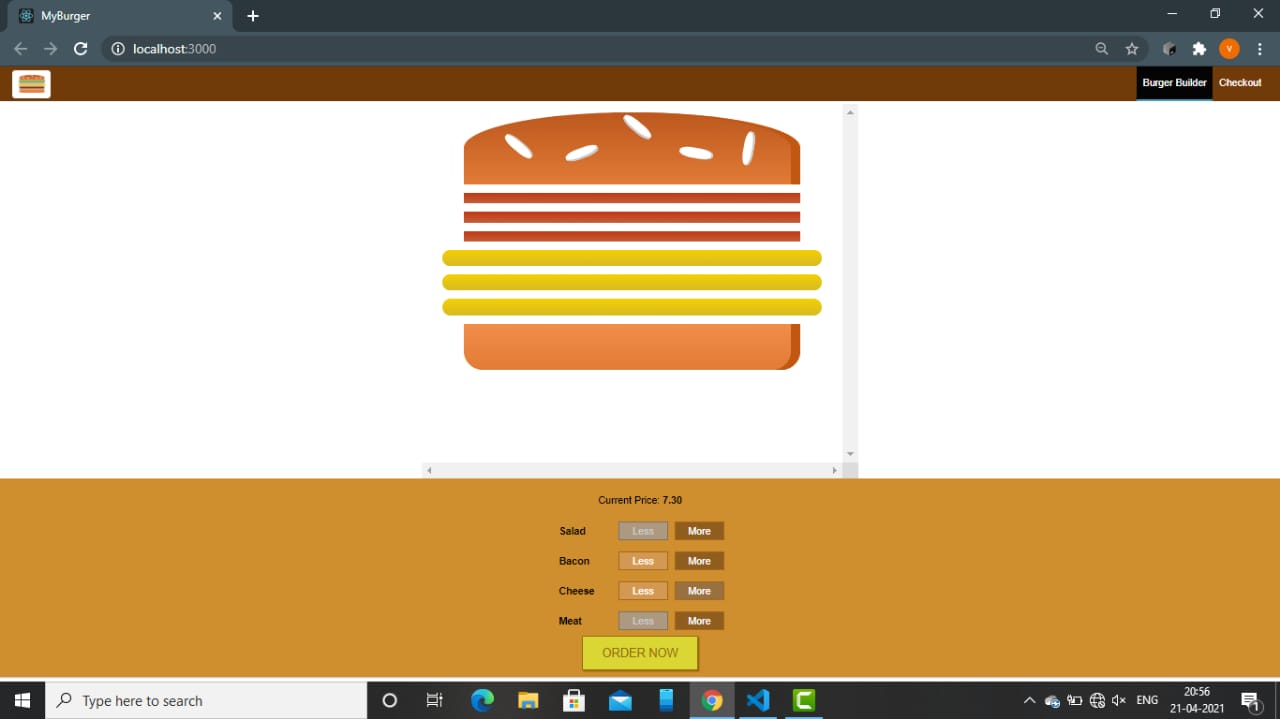
  ReactDOM.unmountComponentAtNode(div);

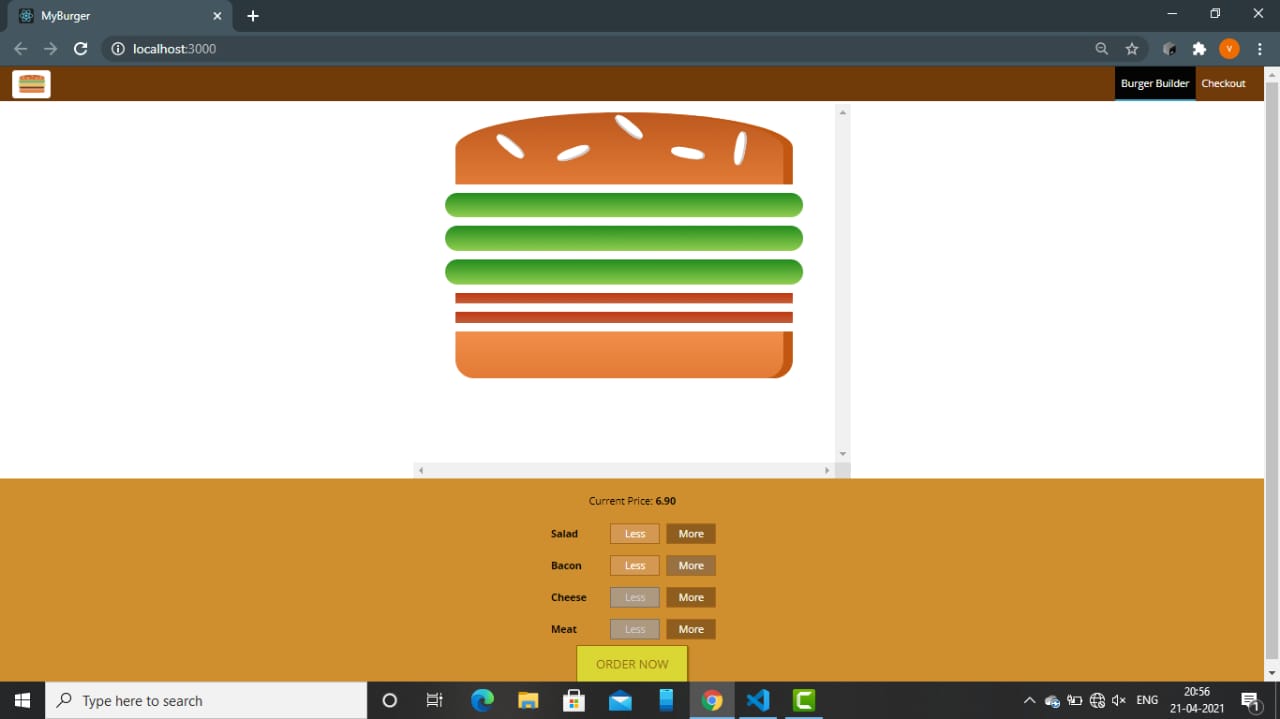
});

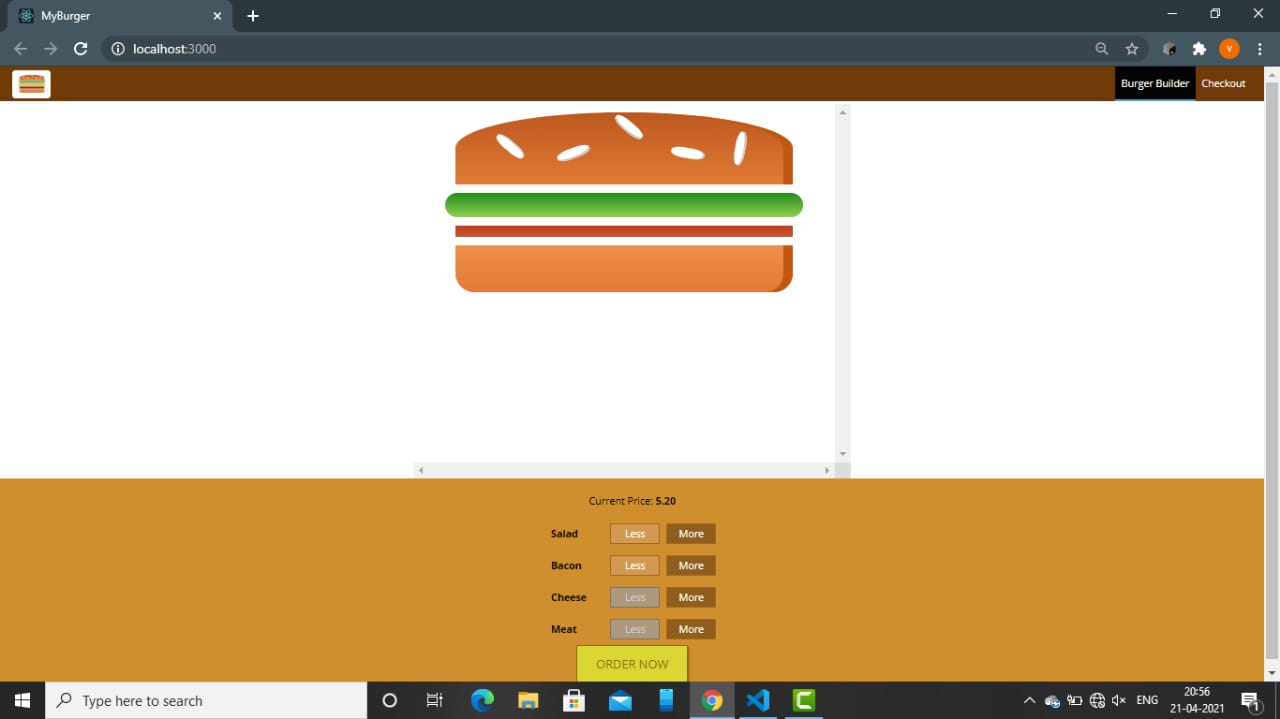
Output:

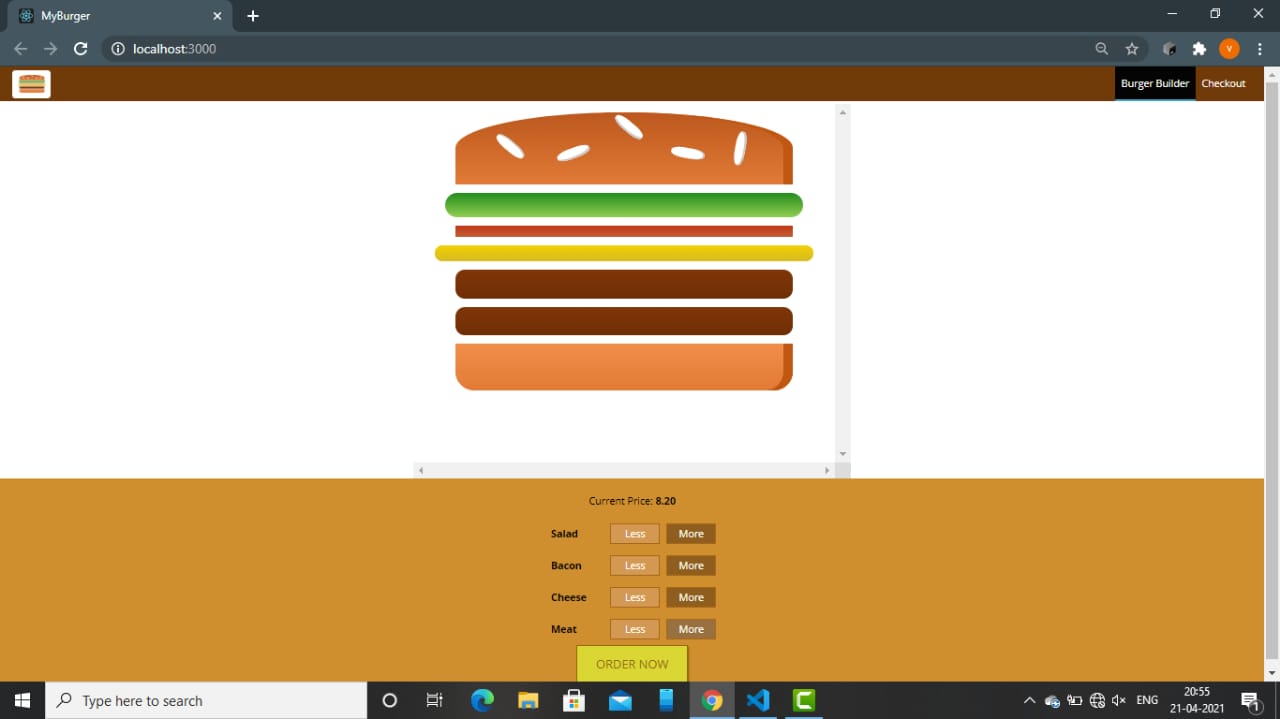


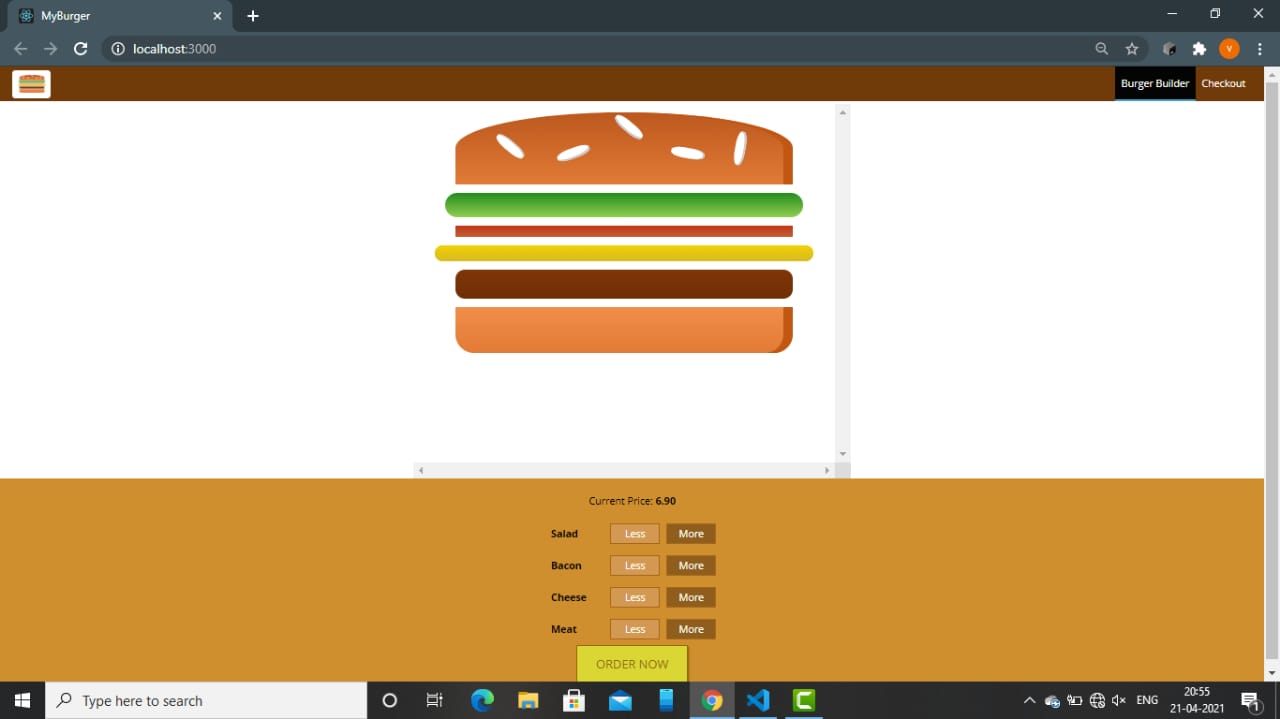


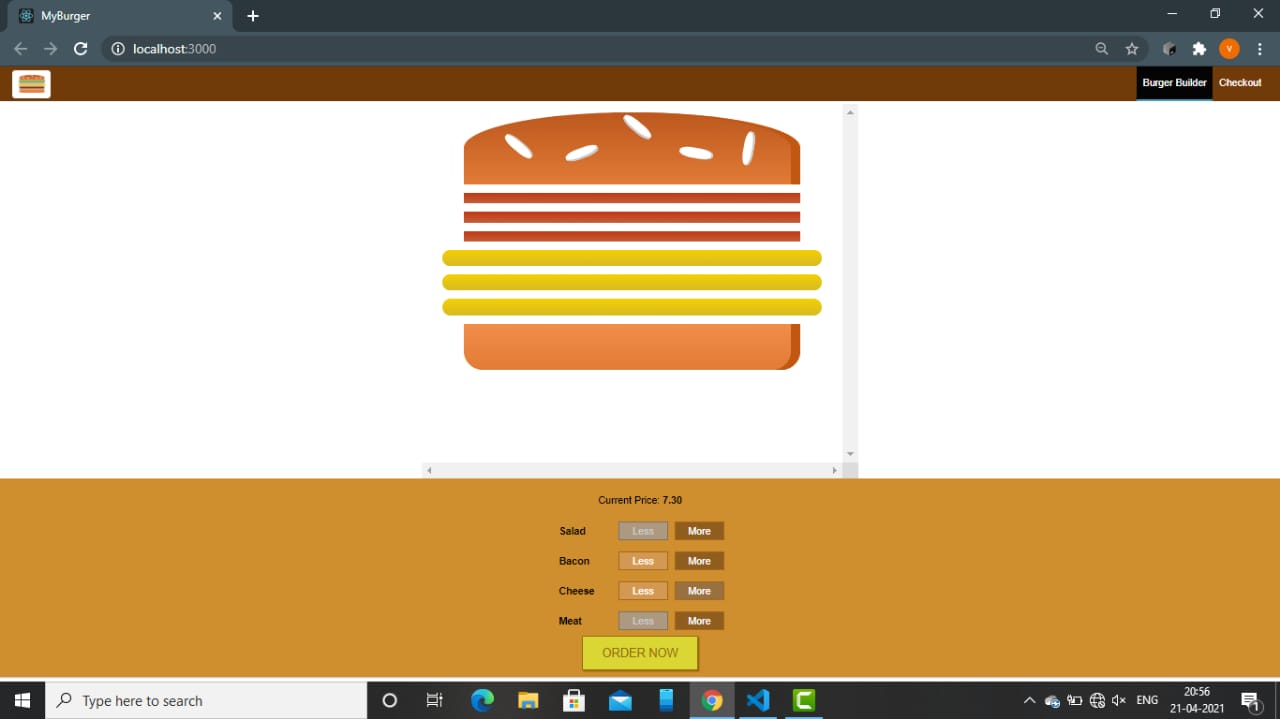


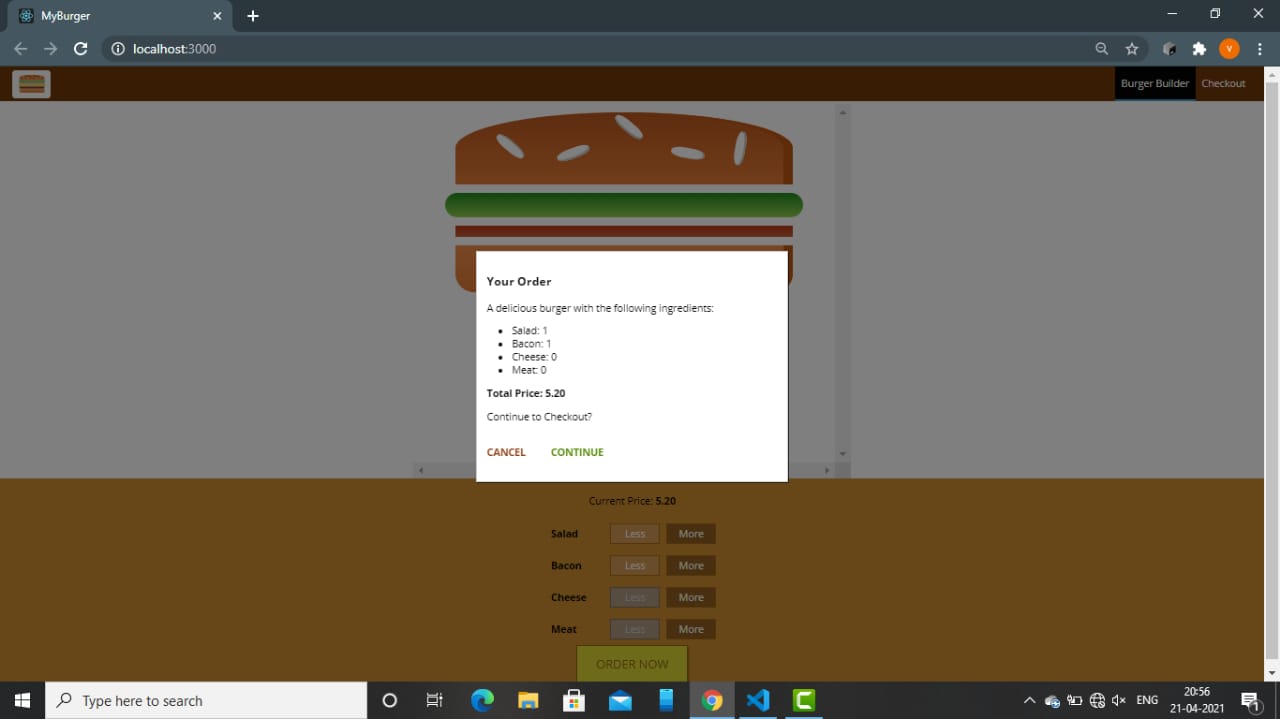


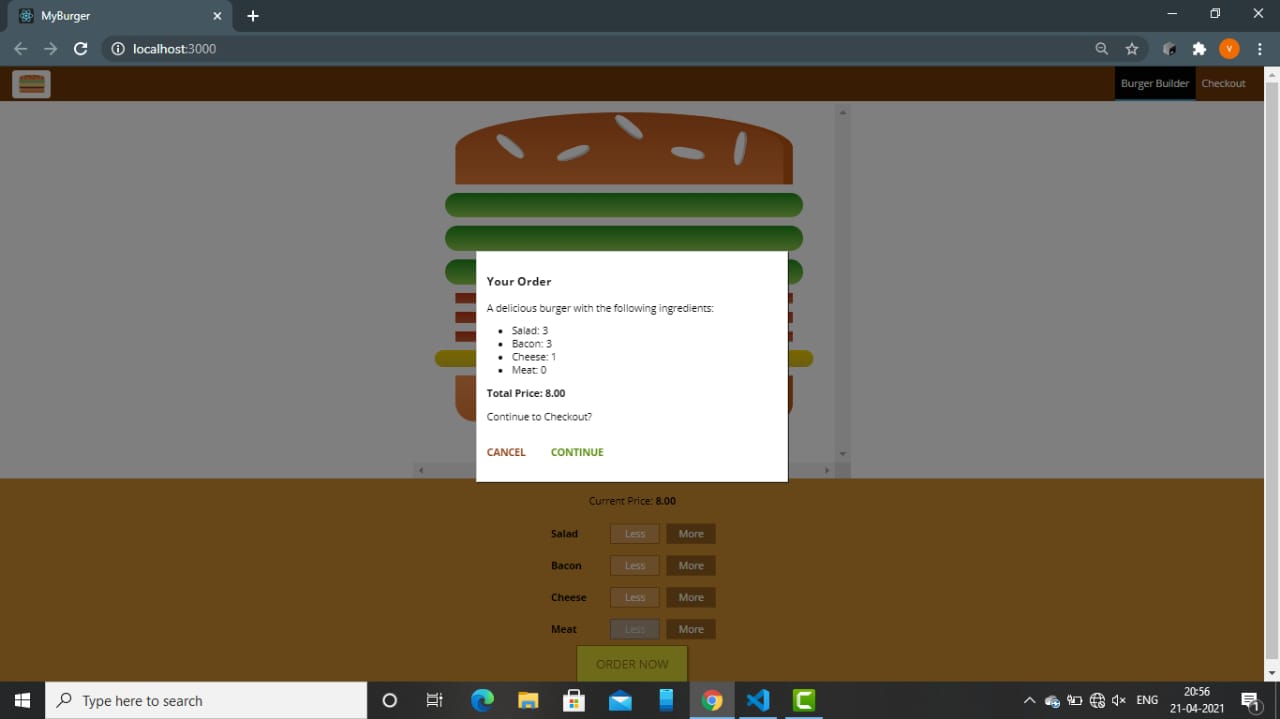


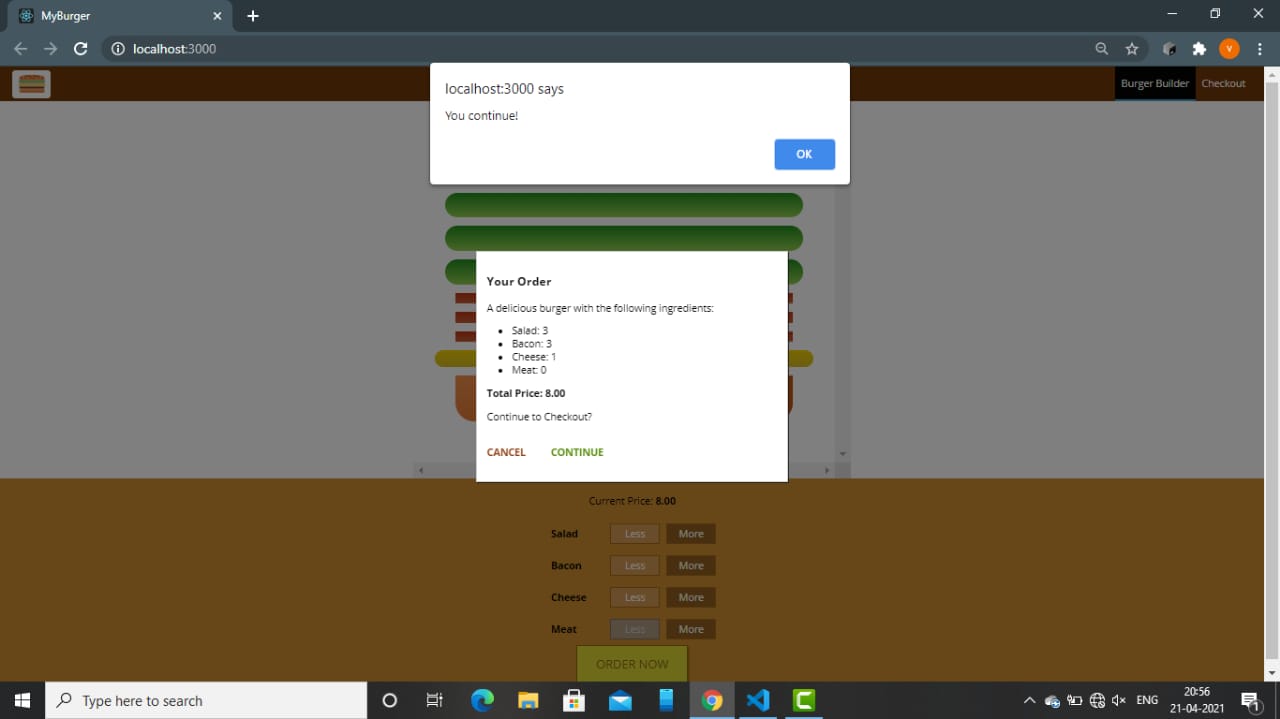






****

****

****

**BIBLIOGRAPHY**

* + - * <https://www.w3schools.com/>
      * <https://stackoverflow.com/>
      * <https://www.tutorialspoint.com/>
      * https://www.youtube.com/