**Chhotu Grocery**

**A PROJECT REPORT**

**Submitted By**

**SHUBHAM**

**University Roll No 1802914015**

**Submitted in partial fulfillment of the**

**Requirements for the Degree of**

**MASTER OF COMPUTER APPLICATION**

**Under the Supervision of**

**DR. SANGEETA ARORA**

**ASSOCIATE PROFESSOR**

**KIET Group of Institutions, Ghaziabad**



**Submitted to**

**Department Of Computer Applications**

**Dr. A.P.J. Abdul Kalam Technical University**

**Uttar Pradesh-226031**

**(JUNE 2021)**

**CERTIFICATE**

Certified **that SHUBHAM (University Roll No 1802914015)** has carried out the project work having “**Chhotu Grocery**” for Master of Computer Applications from Dr. A.P.J. Abdul Kalam Technical University (AKTU**)** (formerly UPTU),Technical University, Lucknow under my supervision. The project report embodies original work, and studies are carried out by the student himself/herself and the contents of the project report do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

**DR. SANGEETA ARORA**

**Associate Professor**

**Department of Computer Applications**

**KIET Group of Institutions, Ghaziabad**

**Signature of Internal Examiner Signature of External Examiner**

**Dr. Ajay kr Shrivastava**

**Head, Department of Computer Application**

**KIET Group of Institutions, Ghaziabad**

**Date:**

**CHOTU GROCERY**

**SHUBHAM**

**ABSTRACT**

The purpose of “Chhotu Grocery” it is a online grocery order system is to automate the existing manual system by the help of computerized equipment and full-fledged computer software , fulfilling their requirements, so that their valuable data/information can be stored for a longer period with easy accessing and manipulation of the same. The required software and hardware are easily available and easy to work with.

Online grocery ordering system , as describe above , can lead to error free, secure ,reliable and fast management system.it can assist the user to concentrate their activities rather to concentrate on the record keeping. Thus it will help organization in better utilization of resources. The organization can maintain computerizes records without redundant entries. That means that one need not be distracted by information that is not relevant, while being able to reach the information.

The aim to automate its existing manual system by the help if computerized equipment and full-fledged computer software, fulfilling their requirements, so that their valuable data/information can be stored for a longer period with easy accessing and manipulation of the same. Basically the project described how to manage for good performance and better services for the clients.

**ACKNOWLEDGEMENT**

Success in life is never attained single handedly. My deepest gratitude goes to my Project supervisor, **DR. SANGEETA ATORA** for her guidance, help and encouragement throughout my research work. Their enlightening ideas, comments, and suggestions.

Words are not enough to express my gratitude to **Dr. Ajay Kumar Srivastava, Professor and Head, Department of Computer Applications**, for his insightful comments and administrative help at various occasions.

Fortunately, I have many understanding friends, who have helped me a lot on many critical conditions.

Finally, my sincere thanks go to my family members and all those who have directly and indirectly provided me moral support and other kind of help. Without their support, completion of this work would not have been possible in time. They keep my life filled with enjoyment and happiness.

**SHUBHAM**

**(1802914015)**

**TABLE OF CONTENT**

|  |  |  |
| --- | --- | --- |
|  | Certificate | ii |
|  | Abstract | iii |
|  | Acknowledgement | iv |
|  | Table of content | v-vi |
|  | List of Figures | vii |
|  |  |  |
| **CHAPTER 1** | **Introduction** | 1-7 |
|  | 1.1 Project description | 1 |
|  | 1.2 Project Purpose | 1 |
|  | 1.3 Software Requirement Specification of “Chhotu Grocery” | 2-7 |
|  | 1.3.1 Introduction | 2 |
|  | 1.3.1.1 Purpose of this topic | 2 |
|  | 1.3.1.2 Audience and Reading Suggestions | 2 |
|  | 1.3.1.3 Scope of Project and Document | 3 |
|  | * + 1. General Description | 3-4 |
|  | * + - 1. Quality Function Deployment of “Chhotu Grocery” | 3-4 |
|  | Specific Requirement | 4-5 |
|  | 1.3.3.1External Interface Requirements of the System7 | 4 |
|  | 1.3.3.2 User Characteristics for the System | 5 |
|  | 1.3.4 Project Model Used | 6-7 |
|  |  |  |
| **CHAPTER 2** | **Feasibility Study** | 8-10 |
|  |  |  |
|  | 2.1 Introduction | 8 |
|  | 2.1.1 Technical feasibility | 8 |
|  | 2.1.2 Economical Feasibility | 9 |
|  | 2.1.3 Operational Feasibility | 9 |
|  | 2.1.4 Social and Behavioral Feasibility | 9 |
|  | 2.1.5 Legal Feasibility | 9 |
|  |  |  |
| **CHAPTER 3** | **System Design** | 10 |
|  |  |  |
|  | 3.1 Introduction | 10 |
|  | 3.2 System Architecture | 10 |
|  | 3.3 Module in the System | 10 |
|  | 3.4 Methodology Used | 11 |
|  | 3.5 Data Flow Diagrams | 13 |
|  | 3.5.1 0-level DFD | 14 |
|  | 3.5.2 1-level DFD | 15 |
|  | 3.6 Entity Relationship Diagram | 15 |
| **CHAPTER 4** | **Source Code** | 16 |
| **CHAPTER 5** | **Snapshot** | 53 |
| **CHAPTER 6** | **Testing** | 64 |
|  | 6.1 Introduction | 65 |
|  | 6.2 Types of Testing | 67 |
| **CHAPTER 7** | **Conclusion and Future Scope** | 68 |
|  | 7.1 Conclusion | 68 |
|  | 7.2 Future Scope | 68 |
| **CHAPTER 8** | **References** | 69 |

**List of Figures**

|  |  |
| --- | --- |
| [**Fig. 1**](#_gjdgxs)**.1** [-Phases of Iterative Model](#_gjdgxs) | 7 |
| [**Fig.**](#_30j0zll) **3.1**[-Use Case Diagram](#_30j0zll) | 12 |
| [**Fig. 4.1** -0-Level DFD](#_1fob9te) | 15 |
| [**Fig. 4**](#_3znysh7)**.2**[-1 Level DFD](#_3znysh7) | 15 |
| [**Fig.**](#_2et92p0) **4.3**[-1-ERD](#_2et92p0) | 16 |
| [**Fig.**](#_tyjcwt) **5.1**[-Home Page](#_tyjcwt) | 53 |
| [**Fig.**](#_3dy6vkm) **5.2**[-Admin Dashboard](#_3dy6vkm) | 54 |
| [**Fig.**](#_1t3h5sf) **5.3**[-login page](#_1t3h5sf) | 55 |
| [**Fig**](#_1t3h5sf)**. 5.4**[-Mobile no. verification](#_1t3h5sf) | 56 |
| [**Fig.**](#_4d34og8) **5.5**[-Logged in successful](#_4d34og8) | 57 |
| [**Fig.**](#_4d34og8) **5.6**[-Manage category](#_4d34og8) item | 58 |
| [**Fig.**](#_4d34og8) **5.7**[-Delivery address](#_4d34og8) | 59 |
| [**Fig.**](#_4d34og8) **5.8**[-Proceed to pay](#_4d34og8) | 60 |
| [**Fig.**](#_4d34og8) **5.9**[-Selecting Date and time](#_4d34og8) | 61 |
| **Fig. 5.10-**Selecting payment method | 62 |
| **Fig.5.10-**Ordered Confirmed | 63 |

**CHAPTER 1**

**INTRODUCTION**

* 1. **PROJECT DESCRIPTION**

The “Online Chotu Grocery” has been developed to override the problems prevailing in the practicing manual system. This software is supported to eliminate and in some cases reduce the hardships faced by this existing system. Online Chotu Grocery can lead to error free, secure and reliable fast news management system.

Online Chotu Grocery allows users to read up to date news related to many fields like entertainment, national, international, business, sports etc. without any payment or login. He can also contact us to give suggestions and can also give us feedback related to our site.

New news can be added only by the admin and only admin have the right to update or delete any news.

Managing your online Chotu Grocery management system may seem tricky, but this is part of user service system.

* 1. **PURPOSE OF PROJECT**
* The objective of this project is to develop a web application for Online News Paper website that can aware the people
* The objective of this project is to provide the daily news.
* Anytime, anywhere, anyone can know about the news or information by internet at low cost
  1. **Software Requirements Specification of “Chhotu Grocery”**

This topic covers the requirements specification of our “Chhotu Grocery”. It includes the specification of this documentation with general description, specific requirements, and analysis of models. It also includes changes management of this requirements specification in case of any change

* + 1. **Introduction**

In this section, the documentation of this report is specified. It specifies the document convention, document scope and provides a suggestion for the readers of the document.

* + - * 1. **Purpose of this Topic**

This Software Requirements Specification (SRS) part is intended to give a complete overview of our Project the “Chhotu Grocery” including the action flow, initial user interface and story therein. The SRS document details all features upon which we have currently decided with reference to the manner and importance of their implementation.

* + - * 1. **Intended Audience and Reading Suggestions**

This project is a prototype for Chhotu Grocery. This has been implemented under the guidance of college professors. This project is useful for anyone to check daily activities around the world.

Although the document may be read from front to back for a complete understanding of the project, it was written in section and hence can be read as such. For an overview of the document and the project itself, see Overall Description. For a detailed description of the system elements and their interaction with the user, read System Features. Readers interested in the system interface and navigation between different front-end menus should go through External Interface Requirements.

* + - * 1. **Scope of Project and Document**

The project has a wide scope, as it is not intended to a particular organization. This project is going to develop generic software, which can be applied by any business’ organization. Moreover it provides facility to its users. Also the software is going to provide a huge amount of summary data.It helps to user to read up to date news related to many fields like entertainment, national, international, business, sports etc. without any payment or login. The purpose of project is to online management of news i.e adds latest news and also categorizes them

**1.3.2 General Description**

It specifies the QFD (Quality Function Deployment) of our software and the User Story of it.

* + - * 1. **Quality Function Deployment of “Chhotu Grocery”**

Quality Function Deployment is a technique that translates the needs of the customer into technical requirements for software. It concentrates on maximizing customer satisfaction from the Software engineering process. With respect to our project the following requirements are identified by a QFD.

* Normal Requirements.
* Expected Requirements.
* **Normal Requirements**

Normal requirements consist of objectives and goals that are stated during the meeting with the relevant people. Normal requirements of our project are:

1. Minimum maintenance cost.
2. Availability of expected requirements within the PC configuration.
3. Easy to operate.
4. The software with measured coding, professional thinking.

* **Expected Requirements**

These requirements are implicit to the system and may be so fundamental that the relevant person does not explicitly state them. Their absence will be a cause for dissatisfaction.

1. Maximum high regulation with minimum hardware.
2. We may provide a user-friendly interface to users for easy commenting and form filling.
3. Easy to update.
   * 1. **SPECIFIC REQUIREMENTS**

This section covers the project external requirements of our software and indicates the User Characteristics for this project.

**1.3.3.1External Interface Requirements of the System**

* **User Interfaces**

Every website must have a home page and a site map to make the software user friendly enough and user can easily fulfill their need. Nav Bar is also an important component. We have added the Home Page, different Forms’ snapshots.

* **Hardware Interfaces**

“Chhotu Grocery” application designed specifically for Windows platform and is functional on both Desktop and Laptop. And a browser which supports HTML&Javascript.

* **Software Interface**

“Chhotu Grocery” has been developed using a series of Website Development with React JS. Working tools and Platform

* **Operating System-** Windows/Linux
* **Technology Used-** React JS(javascript)
* **Front-end Technologies-**HTML5,CSS,Bootstrap,Javascript
* **Code Editor-** Node js,Visual Studio code

**1.3.3.2 User Characteristics for the System**

It helps to user to read up to date news related to many fields like entertainment, national, international, business, sports etc. without any payment or login.

* + 1. **PROJECT MODEL USED**

**Iterative Enhancement Model**

* This model has the same phases as the waterfall model, but with fewer restrictions.
* Generally the phases occur in the same order as in the waterfall model, but they may be conducted in several cycles.
* Useable product is released at the end of the each cycle, with each release providing additional functionality. Customers and developers specify as many requirements as possible and prepare a SRS document. Developers and customers then prioritize these requirements. Developers implement the specified requirements in one or more cycles of design, implementation and test based on the defined priorities.

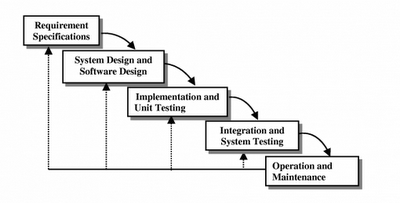
The procedure itself consists of the initialization step, the iteration step, and the Project Control List. The initialization step creates a base version of the system. The goal for this initial implementation is to create a product to which the user can react. It should offer a sampling of the key aspects of the problem and provide a solution that is simple enough to understand and implement easily. To guide the iteration process, a project control list is created that contains a record of all tasks that need to be performed. It includes such items as new features to be implemented and areas of redesign of the existing solution. The control list is constantly being revised as a result of the analysis phase.

The iteration involves the redesign and implementation of iteration is to be simple, straightforward, and modular, supporting redesign at that stage or as a task added to the project control list. The level of design detail is not dictated by the iterative approach. In a light-weight iterative project the code may represent the major source of [documentation](http://en.wikipedia.org/wiki/Software_documentation) of the system; however, in a critical iterative project a formal [Software Design Document](http://en.wikipedia.org/wiki/Software_Design_Document) may be used. The analysis of an iteration is based upon user feedback, and the program analysis facilities available. It involves analysis of the structure, modularity, [usability](http://en.wikipedia.org/wiki/Usability), reliability, efficiency, & achievement of goals. The project control list is modified in light of the analysis results.

### PHASES:

Incremental development slices the system functionality into increments (portions). In each increment, a slice of functionality is delivered through cross-discipline work, from the requirements to the deployment. The unified process groups increments/iterations into phases: inception, elaboration, construction, and transition.

* Inception identifies project scope, requirements (functional and non-functional) and risks at a high level but in enough detail that work can be estimated.
* Elaboration delivers a working architecture that mitigates the top risks and fulfills the non-functional requirements.
* Construction incrementally fills-in the architecture with production-ready code produced from analysis, design, implementation, and testing of the functional requirements.
* Transition delivers the system into the production operating environment.

****

**Fig. 1.1 Phases of Iterative Model**

**CHAPTER 2**

**FEASIBILITY STUDY**

**2.1 INTRODUCTION**

A feasibility study is an analysis that takes all of a project's relevant factors into account—including economic, technical, legal, and scheduling considerations—to ascertain the likelihood of completing the project successfully. Project managers use feasibility studies to discern the pros and cons of undertaking a project before they invest a lot of time and money into it.

The feasibility study is conducted to check whether the candidate system is feasible. The system which is selected to be the best against the criteria is there after designed and developed. The feasibility study takes in to consideration, the risks involved in the project development beforehand. Therefore in this phase we have to do feasibility study which is the test of the website according to its work ability, impact on the organization, ability to meet user need and effective use of resources. We do the feasibility study for website to analyze the risks, costs and benefits relating to economics, technology and user organization. There are several types of feasibility depending on the aspect they cover. Import of these includes:

**2.1.1 TECHNICAL FEASIBILITY**

This is an important outcome of preliminary investigation. It comprise of following questions:-

* Can the work of project be one with the current equipment, existing software and available man power resource?
* If Technology is required what are the possibilities that it can be developed?

We can strongly say that it is technically feasible, since there will not be much difficulty in getting resources for the development and maintaining the system as well. All the resources needed for the development of the software as well as the maintenance of the same is available in the organization.

**2.1.2 ECONOMICAL FEASIBILITY**

It deals with question related to the economy. It comprise of the following questions:-

* Are there sufficient benefits in creating the system to make the cost acceptable?
* Are the costs of not creating the system so great that the project must be undertaken?

Development of this application is highly economically feasible. The organization needed not spend much money for the development of the system already available. The only thing is to be done is making an environment for the development with an effective supervision. Even after the development, the organization will not be in condition to invest more in the organization.

**2.1.3 OPERATIONAL FEASIBILITY**

The operational feasibility consists of the following activity:-

* + - * Will the system be useful if it is developed &implemented?

**2.1.4 SOCIAL AND BEHAVIORAL FEASIBILITY**

It deals with the various issues related to the human behavior like:-

* + - * Whether the user be able to adapt a new change or not?
      * Whether the ambiance we are providing suits the user or not?

**2.1.5 LEGAL FEASIBILITY**

It deals with the question related to the legal issues. It comprise of the following questions:-

* Contract Signing
* Software License agreement
  + - * Issues related to cyber laws.
      * Legal issues relating to the man power contract.

**CHAPTER 3**

**SYSTEM DESIGN**

**3.1 INTRODUCTION**

**System design** is the phase that bridges the gap between problem domain and the existing system in a manageable way. This phase focuses on the solution domain, i.e. *“how to implement?”*

**3.2SYSTEM ARCHITECTURE**

System design is the process of defining system architecture, modules and interfaces for the proposed system to satisfy specified requirement.

**3.3 Modules in the System**

The system comprises of 2 major modules as follows:

1. **Admin:**
   1. Secure admin login system
   2. Admin dashboard
   3. **Category** – In this section admin can add/update/delete the category. Admin can also restore deleted category
   4. **Sub- Category** – In this section admin can add/update/delete the Subcategory. Admin can also restore deleted Subcategory
   5. **Post –** Admin can add /update / delete news posts. admin can also view deleted news post in trash post section and restore deleted posts.
   6. **Pages –**Admin can manage the contact of about us and contact us page.
   7. **Comments –**Admin can approve/ unapprove / delete reader comments.
2. **User:**

Anyone can read the news and also search for particular news. The reader can leave comments on the particular news

**3.4METHODOLOGY USED**

**3.4.1 Javascript**

Javascript is a widely used general-purpose, high level programming language. It was initially designed by Guido van Rossum in 1991 and developed by Javascript Software Foundation. It was mainly developed for emphasis on code readability, and its syntax allows programmers to express concepts in fewer lines of code.

Javascript is a programming language that lets you work quickly and integrate systems more efficiently.

Javascript is dynamically typed and garbage-collected. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming. Javascript is often described as a "batteries included" language due to its comprehensive standard library.

**3.4.2 HTML**

HTML (Hypertext Markup Language) is the set of markup symbols or codes inserted in a file intended for display on a World Wide Web browser page. The markup tells the Web browser how to display a Web page's words and images for the user. Each individual markup code is referred to as an element (but many people also refer to it as a tag). Some elements come in pairs that indicate when some display effect is to begin and when it is to end.

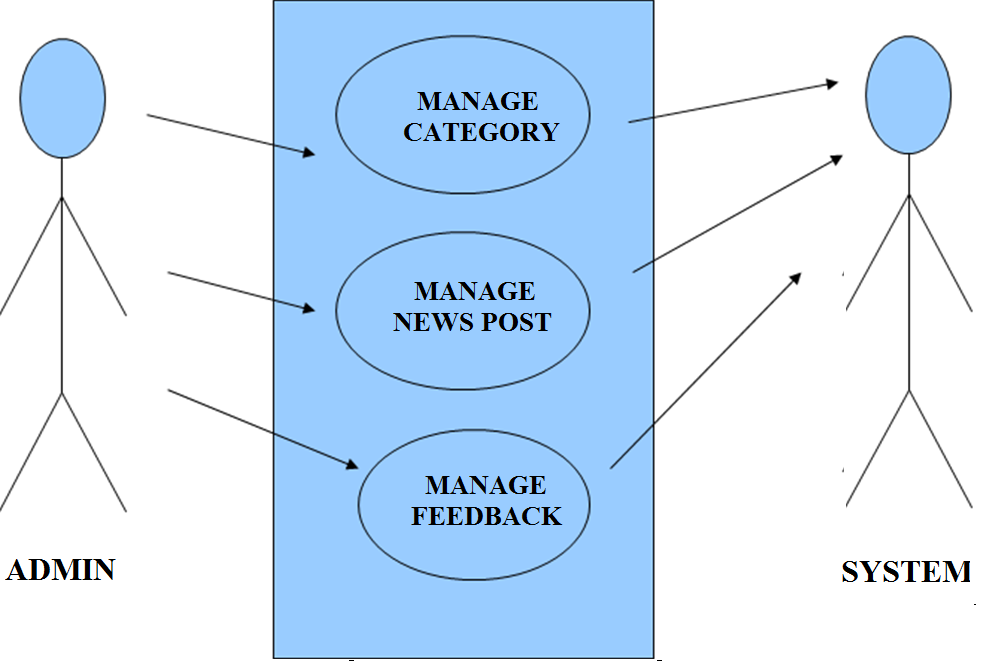
**3.4.3Javascript**

JavaScript is a programming language commonly used in web development. It was originally developed by Netscape as a means to add dynamic and interactive elements to websites. While JavaScript is influenced by Java, the syntax is more similar to C and is based on ECMAScript, a scripting language developed by Sun Microsystems.

JavaScript is a client-side scripting language, which means the source code is processed by the client's web browser rather than on the web server. This means JavaScript functions can run after a webpage has loaded without COMMUNICATING with the server. For example, a JavaScript function may check a web form before it is submitted to make sure all the required fields have been filled out. The JavaScript code can produce an error message before any information is actually transmitted to the server.

**3.4.4React JS**

React JS is a web application framework written in Javascript programming language. It is based on MVT (Model View Template) design pattern. The React JS is very demanding due to its rapid development feature. It takes less time to build application after collecting client requirement.



**Fig. 3.1 Use Case Diagram**

**Data Flow Diagram**

A Data Flow Diagram (DFD) is a graphical representation of the "flow" of data through an Information System. A data flow diagram can also be used for the visualization of Data Processing. It is common practice for a designer to draw a context-level DFD first which shows the interaction between the system and outside entities. This context-level DFD is then "exploded" to show more detail of the system being modeled.

A DFD represents flow of data through a system. Data flow diagrams are commonly used during problem analysis. It views a system as a function that transforms the input into desired output. A DFD shows movement of data through the different transformations or processes in the system.

Dataflow diagrams can be used to provide the end user with a physical idea of where the data they input ultimately has an effect upon the structure of the whole system from order to dispatch to restock how any system is developed can be determined through a dataflow diagram. The appropriate register saved in database and maintained by appropriate

authorities.

**DFD Level 0 Diagram**

**Diagram

Description automatically generated**

**Fig-4.1-**DFD LEVEL O DIAGRAM

It is a dfd diagram which shows flow of data on website through information system.

**Entity Relationship Diagrams (ER-Diagrams):**

An entity-relationship (ER) diagram is a specialized graphic that illustrates the interrelationships between entities in a database. ER diagrams often use symbols to represent three different types of information. Boxes are commonly used to represent entities. Diamonds are normally used to represent relationships and ovals are used to represent attributes

**ENTITY RELATIONSHIP(ER) DIAGRAM**

Diagram

Description automatically generated

**Fig-4.2-ER Diagram**

It is aER diagram which shows how interrelationships between entities in database.

**CHAPTER 4**

**SOURCE CODE**

* 1. **Home Page Coding**

**import React from 'react';**

**import ReactDOM from 'react-dom';**

**import './index.css';**

**import reportWebVitals from './reportWebVitals';**

**import Routes from "./lib/Routes";**

**ReactDOM.render(**

**<>**

**<React.StrictMode>**

**<Routes />**

**</React.StrictMode>**

**</>,**

**document.getElementById('root')**

**);**

**Index js**

**import React from "react";**

**import ReactDOM from "react-dom";**

**import Base from './core/Base';**

**import Cartvalue from './component/Cartvalue';**

**import Mainindex from "./lib/Mainindex";**

**function App() {**

**return (**

**<>**

**<Mainindex/>**

**</>**

**);**

**}**

**App JS**

**import { withRouter } from "react-router-dom";**

**import React, { useState, useEffect } from 'react';**

**import IconButton from '@material-ui/core/IconButton';**

**import MenuItem from '@material-ui/core/MenuItem';**

**import withWidth from '@material-ui/core/withWidth';**

**import Menu from '@material-ui/core/Menu';**

**import ListItemIcon from '@material-ui/core/ListItemIcon';**

**import ListItemText from "@material-ui/core/ListItemText";**

**import SwipeableDrawer from '@material-ui/core/SwipeableDrawer';**

**import AccountCircle from '@material-ui/icons/AccountCircle';**

**import HomeIcon from '@material-ui/icons/Home';**

**import HistoryIcon from '@material-ui/icons/History';**

**import HamburgerMenu from "../container/menu/HamburgerMenu";**

**// import OrderHistory from "../container/orderHistory/OrderHistory";**

**// import Profile from "../container/profile/Profile";**

**// import { signout, isAuthenticated } from "../services/auth";**

**// import {getCartCountValue} from "../services/user";**

**function MenuBar({ history, width, reload }) {**

**const [anchorEl, setAnchorEl] = React.useState(null);**

**const [anchorE2, setAnchorE2] = React.useState(null);**

**const [openOrderDialog, setOpenOrderDialog] = React.useState(false);**

**const [openCartDialog, setOpenCartDialog] = React.useState(false);**

**const [openProfileDialog, setOpenProfileDialog] = React.useState(false);**

**const [cartCount, setCartCount] = useState(0);**

**const [refreshCount, setRefreshCount] = useState(false);**

**const isMenuOpen = Boolean(anchorEl);**

**const isProfileMenuOpen = Boolean(anchorE2);**

**// const userId = isAuthenticated() ? isAuthenticated().user.id : null;**

**// const token = isAuthenticated() ? isAuthenticated().token : localStorage.getItem("guestToken");**

**const handleProfileMenuOpen = (event) => {**

**setAnchorEl(event.currentTarget);**

**};**

**const handleMenuOpen = (event) => {**

**setAnchorE2(event.currentTarget);**

**};**

**const handleMenuClose = () => {**

**setAnchorEl(null);**

**setAnchorE2(null)**

**};**

**const menuId = 'primary-search-account-menu';**

**const renderMenu = (**

**<Menu**

**anchorEl={anchorE2}**

**anchorOrigin={{ vertical: 'top', horizontal: 'right' }}**

**id={menuId}**

**keepMounted**

**transformOrigin={{ vertical: 'top', horizontal: 'center' }}**

**open={isProfileMenuOpen}**

**onClose={handleMenuClose}**

**>**

**<MenuItemonClick={() => {**

**history.push("/")**

**handleMenuClose()**

**}}>**

**<ListItemIcon><HomeIcon /></ListItemIcon>**

**<ListItemText>Home</ListItemText>**

**</MenuItem>**

**<MenuItemonClick={() => {**

**}}>**

**<ListItemIcon>**

**<AccountCircle />**

**</ListItemIcon>**

**<ListItemText>Profile</ListItemText>**

**</MenuItem>**

**</Menu>**

**);**

**let menui = 'primary-search-account-menu';**

**return (**

**<HamburgerMenu**

**handleMenuOpen={handleMenuOpen}**

**setOpenCartDialog={setOpenCartDialog}**

**handleProfileMenuOpen={handleProfileMenuOpen}**

**cartCount={cartCount}**

**>**

**{renderMenu}**

**</HamburgerMenu>**

**);**

**}**

**export default MenuBar;**

**Cartvalue.js**

**import React, {useRef,useState} from "react";**

**import { Button, Overlay,Popover} from "react-bootstrap";**

**import {} from ".././../node\_modules/bootstrap/dist/css/bootstrap.min.css";**

**import {cartvalue,maggiImage,deleteImage} from "../lib/utils";**

**import {minus,pay} from "../lib/utils";**

**import RemoveIcon from "@material-ui/icons/Remove";**

**import AddIcon from "@material-ui/icons/Add";**

**import CloseIcon from '@material-ui/icons/Close';**

**import {Route} from "react-router-dom";**

**const Cartvalue=() => {**

**const[show, setshow] = useState(false);**

**const[target,setTarget]=useState(null);**

**const[data,setData]=useState(false);**

**const ref = useRef(null);**

**const handleclick = (event)=>{**

**setshow(!show);**

**setTarget(event.target);**

**};**

**const Handlepage=()=>{**

**setData(false);**

**window.location.href = '/cart' ;**

**}**

**return(**

**<>**

**<div ref={ref}>**

**<p onClick={handleclick}>Cart</p>**

**<Overlay**

**show={show}**

**target={ref.current}**

**placement="bottom"**

**container={ref.current}**

**containerPadding={50}**

**>**

**<Popover id="popover-contained">**

**<Popover.Title as="h2"style={{background:'003A305',color:'white',width:'480px',padding:'15px'}}>**

**<imgsrc={cartvalue}/>My Cart</Popover.Title>**

**<Popover.Content style={{width:'470px',background:'white'}}>**

**<strong>**

**<imgsrc={maggiImage} style={{height:'60px'}}/>**

**</strong>maggi 2- minute masala,700(pack os 12)**

**<imgsrc={deleteImage} style={{marginLeft:'35px'}}/>**

**<Button variant="outline-success"style={{marginLeft:'60px'}}><RemoveIcon/></Button>{''}**

**<span style={{marginLeft:'15px'}}>1</span>**

**<Button variant="outline-success"style={{marginLeft:'20px'}}><AddIcon/></Button>{''}**

**<CloseIcon style={{marginLeft:'10px'}}></CloseIcon>**

**<span style={{marginLeft:'10px'}}>135 /-</span>**

**<span style={{marginLeft:'90px',fontWeight:'bold'}}>135/-</span>**

**</Popover.Content>**

**<Popover.Content style={{width:'470px',background:'white'}}>**

**<strong>**

**<imgsrc={maggiImage} style={{height:'60px'}}/>**

**</strong>maggi 2- minute masala,700(pack os 12)**

**<imgsrc={deleteImage} style={{marginLeft:'35px'}}/>**

**<Button variant="outline-success"style={{marginLeft:'60px'}}><RemoveIcon/></Button>{''}**

**<span style={{marginLeft:'15px'}}>1</span>**

**<Button variant="outline-success"style={{marginLeft:'20px'}}><AddIcon/></Button>{''}**

**<CloseIcon style={{marginLeft:'10px'}}></CloseIcon>**

**<span style={{marginLeft:'10px'}}>135 /-</span>**

**<span style={{marginLeft:'90px',fontWeight:'bold'}}>135/-</span>**

**</Popover.Content>**

**<Popover.Content style={{width:'480px',background:'white'}}>**

**<p style={{marginTop:'160px',fontWeight:'bold',fontSize:'14px'}}>sub total**

**<span style={{float:'right'}}>360/-</span></p>**

**<p style={{fontWeight:'bold',fontSize:'14px'}}>delivery charges**

**<span style={{float:'right'}}>49/-</span></p>**

**<p style={{fontWeight:'bold',fontSize:'14px'}}>total**

**<span style={{float:'right'}}>400/-</span></p>**

**</Popover.Content>**

**<Popover.Content style={{width:'480px',background:'white'}}>**

**<Button onClick={Handlepage} style={{padding:'8px 58px'}}variants="success">**

**<span style={{marginRight:'80px',fontSize:'14px'}}>2 item 400/-</span>**

**proceed to pay <imgsrc={pay}/>**

**</Button>{''}**

**</Popover.Content>**

**</Popover>**

**</Overlay>**

**</div>**

**</>**

**)**

**}**

**export default Cartvalue**

**Date.jsx**

**import React, { useState } from "react";**

**import {ListGroup,Button,Form} from "react-bootstrap";**

**const Date=()=>{**

**const[data,setData]=useState(true);**

**const handlepage=()=>{**

**setData(false);**

**window.location.href="/Payment";**

**}**

**return(**

**<>**

**<div style={{maxHeight:"45px",marginBottom:"10px",marginTop:"20px"}}>**

**<ListGroup horizontalclassName="nchange\_13"style={{marginLeft:"139px",marginRight:"139px"}}**

**ListGroup.ItemclassName="listing" style={{border:"none",borderBottom:"5px solid green",color:"#03A305"}}>My cart</ListGroup.Item>**

**<ListGroup.ItemclassName="listing" style={{border:"none",borderBottom:"5px solid green",color:"#03A305"}}>Delivery Address</ListGroup.Item>**

**<ListGroup.ItemclassName="listing" style={{border:"none",borderBottom:"5px solid green",color:"#03A305"}}>Date & Time</ListGroup.Item>**

**<ListGroup.ItemclassName="listing" style={{border:"none"}}>Payment</ListGroup.Item>**

**</ListGroup>**

**</div>**

**<ListGroup>**

**<ListGroup.Item variant="light" style={{marginLeft:"139px",marginRight:"139px",height:"538px",marginBottom:"10px",marginTop:"12px"}} className="bigcontent">**

**<ListGroup>**

**<ListGroup.ItemclassName="firstlist" style={{marginLeft:"16px",marginRight:"16px",height:"84px"}}>**

**<p className="timeone">Delivery Slots</p>**

**{/\* <span style={{marginLeft:"60px",marginRight:"94px"}} className="timetwo">Mon</span>**

**<span style={{marginRight:"94px",borderBottom:"5px solid green"}} className="timetwo">Tue</span>**

**<span style={{marginRight:"94px"}} className="timetwo">Wed</span>**

**<span style={{marginRight:"94px"}} className="timetwo">Thu</span>**

**<span style={{marginRight:"94px"}} className="timetwo">Fri</span>**

**<span style={{marginRight:"94px"}} className="timetwo">Sat</span>**

**<span style={{marginRight:"60px"}} className="timetwo">Sun</span> \*/}**

**<span className="timetwo">Mon</span>**

**<span className="timetwo" style={{borderBottom:"2px solid green"}}>Tue</span>**

**<span className="timetwo">Wed</span>**

**<span className="timetwo">Thu</span>**

**<span className="timetwo">Fri</span>**

**<span className="timetwo">Sat</span>**

**<span className="timetwo">Sun</span>**

**</ListGroup.Item>**

**<ListGroup.Item style={{marginLeft:"16px",marginRight:"16px",height:"84px",marginTop:"7px",border:"none"}} >**

**<span style={{marginLeft:"60px", marginRight:"74px"}}>22Feb</span>**

**<span style={{marginRight:"74px",color:"green"}}>23Feb</span>**

**<span style={{marginRight:"74px"}}>24Feb</span>**

**<span style={{marginRight:"74px"}}>25Feb</span>**

**<span style={{marginRight:"74px"}}>26Feb</span>**

**<span style={{marginRight:"74px"}}>27Feb</span>**

**<span style={{marginRight:"60px"}}>28Feb</span>**

**</ListGroup.Item>**

**<ListGroup.Item style={{marginLeft:"16px",marginRight:"16px",height:"84px",marginTop:"7px",border:"none"}} >**

**{/\* <Button variant="outline-success" className="timethree"><p>6am-9am**

**<span>**

**<input**

**class="form-check-input"**

**type="radio"**

**name="radioNoLabel"**

**id="radioNoLabel1"**

**value=""**

**aria-label="..."**

**/></span></p>**

**</Button>{' '}**

**<Button variant="outline-success">9am-1pm</Button>{' '}**

**<Button variant="outline-success">4pm-7pm</Button>{' '}**

**<Button variant="outline-success">7pm-11pm</Button>{' '} \*/}**

**<button class="button2 button3" style={{padding:"3px 13px"}}>6am-9am**

**<span style={{float:'right'}}>**

**{['radio'].map((type) => (**

**<div key={`inline-${type}`} >**

**<Form.Check type={type} id={`inline-${type}-6`} name="formHorizontalRadios" />**

**</div>**

**))}</span>**

**</button>**

**<button class="button2 button3" style={{padding:"3px 10px"}}>9am-1pm**

**<span style={{float:'right'}}>**

**{['radio'].map((type) => (**

**<div key={`inline-${type}`} >**

**<Form.Check type={type} id={`inline-${type}-6`} name="formHorizontalRadios" />**

**</div>**

**))}</span>**

**</button>**

**<button class="button2 button3" style={{padding:"3px 10px"}}>4pm-7pm**

**<span style={{float:'right'}}>**

**{['radio'].map((type) => (**

**<div key={`inline-${type}`} >**

**<Form.Check type={type} id={`inline-${type}-6`} name="formHorizontalRadios" />**

**</div>**

**))}</span></button>**

**<button class="button2 button3" style={{padding:"3px 10px"}}>7pm-11pm**

**<span style={{float:'right'}}>**

**{['radio'].map((type) => (**

**<div key={`inline-${type}`} >**

**<Form.Check type={type} id={`inline-${type}-6`} name="formHorizontalRadios" />**

**</div>**

**))}</span></button>**

**</ListGroup.Item>**

**<Button variant="success" onClick={handlepage} style={{marginLeft:"720px",marginRight:"16px",marginTop:"200px"}}>Next</Button>{' '}**

**</ListGroup>**

**</ListGroup.Item>**

**</ListGroup>**

**</>**

**)**

**}**

**export default Date**

**DeliveryAddress**

**import React, { useState } from "react";**

**import {ListGroup,Button,Modal,Form} from "react-bootstrap";**

**import {Home} from "../lib/utils";**

**import {addbutton,Address} from "../lib/utils";**

**const Deliveryaddress=()=>{**

**const [show, setShow] = useState(false);**

**const [data,setData]=useState(true);**

**const handleClose = () =>setShow(false);**

**const handleShow = () =>setShow(true);**

**const handlepage=()=>{**

**setData(false);**

**window.location.href="/Date";**

**}**

**function Example(props) {**

**return (**

**<>**

**<Modal**

**show={show}**

**onHide={handleClose}**

**backdrop="static"**

**keyboard={false}**

**>**

**<Modal.HeadercloseButton style={{border:"none",padding:"0px"}}>**

**<Modal.Title></Modal.Title>**

**</Modal.Header>**

**<Modal.Body style={{marginLeft:"18px",marginRight:"18px"}}>**

**<p className="delivone" >Add new address</p>**

**<p className="delivtwo" style={{marginLeft:"29px",marginRight:"29px"}}><b>Enter Accurate Address To Get Delivery Fast To Your Doorstep</b></p>**

**<p className="delivthree">Area/Locality</p>**

**<div class="container">**

**<form class="form-inline" action="/action\_page.php">**

**<div class="form-group">**

**<Form.Control as="select" id="formGroupExampleInput" style={{width:"270px",marginRight:"10px"}} >**

**<option>search area</option>**

**<option>2</option>**

**<option>3</option>**

**<option>4</option>**

**<option>5</option>**

**</Form.Control>**

**<Button variant="outline-success"><imgsrc={Address}></img>Get Location</Button>{' '}**

**</div>**

**</form>**

**</div>**

**<form>**

**<div class="form-group">**

**<label for="formGroupExampleInput"><p className="delivthree">Full Name</p></label>**

**<input type="text" class="form-control" id="formGroupExampleInput" placeholder="enter full name"></input>**

**</div>**

**</form>**

**<form>**

**<div class="form-group">**

**<label for="formGroupExampleInput"><p className="delivthree">Complete Address</p></label>**

**<input type="text" class="form-control" id="formGroupExampleInput" placeholder="enter complete address"></input>**

**</div>**

**</form>**

**<Form>**

**{['radio'].map((type) => (**

**<div key={`inline-${type}`} className="mb-3">**

**<Form.Check inline label="Home" name="group1" type={type} id={`inline-${type}-1`} />**

**<Form.Check inline label="Office" name="group1" type={type} id={`inline-${type}-2`} />**

**<Form.Check inline label="Others" name="group1" type={type} id={`inline-${type}-1`} />**

**</div>**

**))}**

**</Form>**

**<Button variant="success" style={{width:"302px",marginLeft:"70px",marginRight:"70px",marginBottom:"25px"}}><span className="delivfive">Continue Shopping</span></Button>{' '}**

**</Modal.Body>**

**</Modal>**

**</>**

**);**

**}**

**return(**

**<>**

**<div style={{maxHeight:"45px",marginBottom:"10px",marginTop:"20px"}}>**

**<ListGroup horizontal className="nchange\_13"style={{marginLeft:"139px",marginRight:"139px"}} >**

**<ListGroup.ItemclassName="listing" style={{border:"none",borderBottom:"5px solid green",color:"#03A305"}} >My cart</ListGroup.Item>**

**<ListGroup.ItemclassName="listing" style={{border:"none",borderBottom:"5px solid green"}}>Delivery Address</ListGroup.Item>**

**<ListGroup.ItemclassName="listing" style={{border:"none"}}>Date & Time</ListGroup.Item>**

**<ListGroup.ItemclassName="listing" style={{border:"none"}}>Payment</ListGroup.Item>**

**</ListGroup>**

**</div>**

**<ListGroup>**

**<ListGroup.Item variant="light" className="bigcontent" style={{marginLeft:"139px",marginRight:"139px",height:"538px",marginBottom:"10px",marginTop:"12px"}}>**

**<ListGroup>**

**<ListGroup.ItemclassName="firstlist" style={{marginLeft:"16px",marginRight:"16px",height:"94px"}}>**

**<imgsrc={Home}/>**

**<bold className="home" style={{marginLeft:"17px"}}>Home</bold>**

**<p className="address" style={{marginLeft:"43px"}}>228,Lajpat Rai Market,Chandani Chowk Delhi 110006 <span className="edit" style={{marginLeft:"450px"}}>Edit</span></p>**

**</ListGroup.Item>**

**<ListGroup.ItemclassName="firstlist" style={{marginLeft:"16px",marginRight:"16px",height:"94px",marginTop:"7px"}}>**

**<button onClick={handleShow} className="btnshow">+</button>**

**<Example**

**handleClose = {() =>setShow(false)}**

**handleShow = {() =>setShow(true)} />**

**<bold className="addressnew" style={{marginLeft:"17px"}}>Add Another/New Address</bold>**

**</ListGroup.Item>**

**<Button variant="success" onClick={handlepage} style={{marginLeft:"720px",marginRight:"16px",marginTop:"250px"}}>Confirm Address</Button>{' '}**

**</ListGroup>**

**</ListGroup.Item>**

**</ListGroup>**

**</>**

**)**

**}**

**export default Deliveryaddress**

**Discount.jsx**

**import React from 'react';**

**import {} from "../../node\_modules/bootstrap/dist/css/bootstrap.min.css";**

**import {Card} from 'react-bootstrap';**

**import Discountimage from "../lib/Discountimage";**

**import "../../src/App.css"**

**function Discount(props)**

**{**

**return(**

**<>**

**<div>**

**<Card style={{ width: '15rem' }} className="cardItem">**

**<Card.Title>**

**<span className="span\_value">{props.offer}</span>**

**</Card.Title>**

**<Card.Img variant="top" src={props.image} className="img\_type" />**

**<Card.Body>**

**<div>**

**<center><h5 style={{fontSize:"0.85rem"}}>{props.name}</h5></center>**

**</div>**

**</Card.Body>**

**</Card>**

**</div>**

**</>**

**)**

**}**

**export default Discount;**

**Dropdownmenu.jsx**

**import React, { useEffect, useState } from 'react';**

**import { } from "../../node\_modules/bootstrap/dist/css/bootstrap.min.css";**

**import {Nav, Navbar,NavDropdown } from "react-bootstrap";**

**import axios from "axios";**

**function Dropdownmenu() {**

**const[item,setitem]=useState([]);**

**useEffect(()=>{**

**axios.get("https://chotugrocery.com/ApiData/getCategoryWithSubcategory")**

**.then(res=>{**

**setitem(res.data.categories)**

**})**

**.catch(error=>{**

**console.log(error);**

**})**

**},[])**

**return(**

**<>**

**<Navbar bg="light" expand="lg" className="dropdown">**

**<Navbar.Toggle aria-controls="basic-navbar-nav" />**

**<Navbar.Collapse id="basic-navbar-nav">**

**<Nav className="mr-auto">**

**{item.map((menu,index)=>(**

**<NavDropdown title={menu.category\_name} id="basic-nav-dropdown" style={{fontWeight:'bold'}}>**

**{menu.subcategory.map((sub,i)=>(**

**<NavDropdown.Itemhref="#action/3.1">{sub.category\_name}</NavDropdown.Item>**

**))}**

**</NavDropdown>**

**))}**

**</Nav>**

**</Navbar.Collapse>**

**</Navbar>**

**</>**

**)**

**}**

**export default Dropdownmenu;**

**HomeRouting.jsx**

**import React from 'react';**

**import { Redirect, Route, Switch, withRouter } from 'react-router';**

**import Mycart from "./Mycart";**

**import Menu from "../core/Menu";**

**const Homerouting = () => {**

**return (**

**<>**

**<Menu/>**

**<Mycart/>**

**</>**

**)**

**};**

**export default withRouter(Homerouting);**

**Lastdailydeals.jsx**

**import React from "react";**

**import { ListGroup,Button } from "react-bootstrap";**

**import Dailydeals from "../core/Dailydeals";**

**import Dailydealsimage from "../lib/Dailydealsimage";**

**const Lastdailydeals=()=>{**

**return(**

**<>**

**<ListGroup horizontal>**

**<ListGroup.Item style={{width:"230px",height:"825px",padding:"0px",border:"none",marginLeft:"20px"}} className="deals\_item">**

**<ListGroup variant="flush">**

**<ListGroup.ItemclassName="item\_1" style={{height:"57px"}} ><bold>Dailydeals</bold></ListGroup.Item>**

**<ListGroup.ItemclassName="item\_2" >Cold Drinks</ListGroup.Item>**

**<ListGroup.ItemclassName="item\_2" >Tea & coffee</ListGroup.Item>**

**<ListGroup.ItemclassName="item\_2" >Juices & Drinks</ListGroup.Item>**

**<ListGroup.ItemclassName="item\_2" >HealthlyEnery Drinks</ListGroup.Item>**

**<ListGroup.ItemclassName="item\_2" >Spices</ListGroup.Item>**

**</ListGroup>**

**</ListGroup.Item>**

**<ListGroup.Item style={{width:"100%",padding:"0px",border:"none"}}>**

**<ListGroup>**

**<ListGroup.ItemclassName="daily" style={{border:"none"}}>Home Dailydeals</ListGroup.Item>**

**</ListGroup>**

**<ListGroup>**

**<ListGroup.Item style={{width:"100%",height:"320px",border:"none"}} className="mainitem">**

**<div>**

**<h3 style={{ marginTop: '40px', marginLeft: '50px' }}>Daily Deals**

**<Button variant="success" style={{ float: 'right', marginRight: '20px' }}>View All</Button></h3>**

**{Dailydealsimage.map((val)=>{**

**return(**

**<Dailydeals image={val.image}**

**name={val.name}**

**link={val.link}**

**price={val.price}**

**price1={val.price1}**

**price2={val.price2}**

**rating={val.rating}**

**offer={val.offer}/>**

**)**

**})}**

**</div>**

**</ListGroup.Item>**

**</ListGroup>**

**</ListGroup.Item>**

**</ListGroup>**

**</>**

**)**

**}**

**export default Lastdailydeals**

**Myaddress.jsx**

**import React from "react";**

**import {ListGroup} from "react-bootstrap";**

**import {} from "../../node\_modules/bootstrap/dist/css/bootstrap.min.css";**

**const Myaddress=()=>{**

**return(**

**<>**

**<ListGroup horizontal >**

**<ListGroup style={{marginLeft:"139px",width:"240px"}}>**

**<ListGroup.Item style={{height:"548px",padding:"0px",marginTop:"15px",border:"none",marginBottom:"7px"}} className="order\_7">**

**<ListGroup style={{width:"230px"}}>**

**<ListGroup.Item style={{height:"50px",marginBottom:"18px",border:"none"}} className="order\_6"><imgsrc="./image/smile.svg" style={{marginRight:"22px"}}/>+91-65768768</ListGroup.Item>**

**<ListGroup.Item style={{height:"50px",marginBottom:"5px",border:"none",color:"#03A305"}} className="order\_6"><imgsrc="./image/order.svg"orders style={{marginRight:"22px"}}/>My Orders</ListGroup.Item>**

**<ListGroup.Item style={{height:"50px",marginBottom:"5px",border:"none"}} className="order\_6"><imgsrc="./image/place.svg" style={{marginRight:"22px"}}/>My Addresses</ListGroup.Item>**

**<ListGroup.Item style={{height:"50px",marginBottom:"5px",border:"none"}} className="order\_6"><imgsrc="./image/logout.svg" style={{marginRight:"22px"}}/>Logout</ListGroup.Item>**

**</ListGroup>**

**</ListGroup.Item>**

**</ListGroup>**

**<ListGroup style={{width:"788px",marginRight:"139px",marginTop:"15px",marginLeft:"10px"}}>**

**<ListGroup.Item style={{height:"72px",paddingTop:"3px",border:"none"}} className="order\_8">**

**<imgsrc="./image/Home.svg"></img>**

**<bold className="home" style={{marginLeft:"17px"}}>Home</bold>**

**<p className="address" style={{marginLeft:"43px"}}>228,Lajpat Rai Market,Chandani Chowk Delhi 110006 <span className="edit" style={{marginLeft:"210px",marginRight:"38px"}}>Edit</span><span className="order\_9">Delete</span></p>**

**</ListGroup.Item>**

**<ListGroup.Item style={{height:"72px",paddingTop:"3px",border:"none",marginTop:"15px"}} className="order\_8">**

**<imgsrc="./image/Plus.svg"></img>**

**<bold className="addressnew" style={{marginLeft:"17px"}}>Add Another/New Address</bold>**

**</ListGroup.Item>**

**</ListGroup>**

**</ListGroup>**

**</>**

**)**

**}**

**export default Myaddress**

**Mycart.jsx**

**import React, { useState } from "react";**

**import { } from"../../node\_modules/bootstrap/dist/css/bootstrap.min.css";**

**import {ListGroup,Button} from "react-bootstrap";**

**import {maggiImage,redlabel} from "../lib/utils";**

**import RemoveIcon from '@material-ui/icons/Remove';**

**import AddIcon from '@material-ui/icons/Add';**

**import CloseIcon from '@material-ui/icons/Close';**

**const Mycart=()=>{**

**const[data,setData]=useState(true);**

**const Handlepage=()=>{**

**setData(false);**

**window.location.href="/DeliveryAddress";**

**}**

**return (**

**<>**

**<div style={{maxHeight:"45px",marginBottom:"10px",marginTop:"20px"}}>**

**<ListGroup horizontal className="nchange\_13"style={{marginLeft:"139px",marginRight:"139px"}} >**

**<ListGroup.ItemclassName="listing" style={{border:"none",borderBottom:"5px solid green",color:"green"}}>My cart</ListGroup.Item>**

**<ListGroup.ItemclassName="listing" style={{border:"none"}}>Delivery Address</ListGroup.Item>**

**<ListGroup.ItemclassName="listing" style={{border:"none"}}>Date & Time</ListGroup.Item>**

**<ListGroup.ItemclassName="listing" style={{border:"none"}}>Payment</ListGroup.Item>**

**</ListGroup>**

**</div>**

**<div style={{maxHeight:"80px",marginBottom:"3px"}}>**

**<ListGroup horizontal className="nchange\_13"style={{marginLeft:"139px",marginRight:"139px",padding:"0px",maxHeight:"80px"}}>**

**<ListGroup.ItemclassName="list\_1" style={{border:"none"}}>1</ListGroup.Item>**

**<ListGroup.ItemclassName="nchange\_4" style={{border:"none"}}><imgsrc={maggiImage} style={{height:"45px",width:"40px"}}></img><span className="list\_2" >Maggi 2-minute Noodles 70g(pack of 12)</span>**

**<p><span className="nchange\_1" style={{marginLeft:"66px"}}>135/-</span><span className="nchange\_2">MRP 150</span><span className="nchange\_3">Save 15</span></p>**

**</ListGroup.Item>**

**<ListGroup.ItemclassName="list\_3" style={{border:"none"}}>**

**<a href="#" className="nchange\_7">Remove</a></ListGroup.Item>**

**<ListGroup.ItemclassName="list\_4" style={{border:"none"}}><Button variant="outline-success" style={{padding:"0px",float:"left",marginLeft:"0px"}} ><RemoveIcon /></Button>{' '}**

**<span >1</span>**

**<Button variant="outline-success" style={{padding:"0px"}}><AddIcon /></Button>{' '}**

**<CloseIconclassName="nchange\_6"></CloseIcon>**

**<span className="nchange\_9">135 /-</span>**

**</ListGroup.Item>**

**<ListGroup.ItemclassName="list\_5" style={{border:"none"}}><span className="nchange\_8">135/-</span></ListGroup.Item>**

**</ListGroup>**

**</div>**

**<div style={{maxHeight:"80px"}} >**

**<ListGroup horizontal className="nchange\_13" style={{marginLeft:"139px",marginRight:"139px",padding:"0px",maxHeight:"80px"}}>**

**<ListGroup.ItemclassName="list\_1" style={{border:"none"}} >2</ListGroup.Item>**

**<ListGroup.ItemclassName="nchange\_4" style={{border:"none"}}><imgsrc="./image/redlabel.png" style={{height:"45px",width:"40px"}}></img><span className="list\_2" >Red Label Tea Leaf 1 kg</span>**

**<p><span className="nchange\_1" style={{marginLeft:"66px"}}>476/-</span><span className="nchange\_2"> MRP 490 </span><span className="nchange\_3">Save 14</span></p>**

**</ListGroup.Item>**

**<ListGroup.ItemclassName="list\_3" style={{border:"none"}}>**

**<a href="#" className="nchange\_7">Remove</a></ListGroup.Item>**

**<ListGroup.ItemclassName="list\_4" style={{border:"none"}} ><Button variant="outline-success" style={{ marginLeft: '0px',padding:"0px" ,float:"left"}}><RemoveIcon /></Button>{' '}**

**<span>1</span>**

**<Button variant="outline-success" style={{ padding:"0px" }}><AddIcon /></Button>{' '}**

**<CloseIconclassName="nchange\_6"></CloseIcon>**

**<span className="nchange\_9" >476 /-</span>**

**</ListGroup.Item>**

**<ListGroup.ItemclassName="list\_5" style={{border:"none"}}><span className="nchange\_8">476 /-</span></ListGroup.Item>**

**</ListGroup>**

**</div>**

**<ListGroup horizontal style={{marginTop:"209px",marginLeft:"139px",marginRight:"139px",width:"1088px"}}>**

**<ListGroup.Item style={{border:"none"}}>**

**<p className="nchange\_12">2items</p>**

**</ListGroup.Item>**

**<ListGroup.Item style={{width:"300px",marginLeft:"605px",border:"none"}} >**

**<p className="nchange\_10">Sub Total:**

**<span style={{ float: 'right' }}>611 /-</span></p>**

**<p className="nchange\_10">Delivery Charges:**

**<span style={{ float: 'right' }}>49 /-</span></p>**

**<p className="nchange\_10"> Total:**

**<span style={{ float: 'right' }}>660/-</span></p>**

**<Button variant="success" onClick={Handlepage} className="nchange\_11" style={{width:"280px"}}>Next</Button>{' '}**

**</ListGroup.Item>**

**</ListGroup>**

**</>**

**)**

**}**

**export default Mycart**

**Myorder.jsx**

**import React from "react";**

**import {ListGroup} from "react-bootstrap";**

**//import {smile,orders,place,logout} from "../lib/utils"**

**const Myorders=()=>{**

**return(**

**<>**

**<ListGroup horizontal >**

**<ListGroup style={{marginLeft:"139px",width:"240px"}}>**

**<ListGroup.Item style={{height:"548px",padding:"0px",marginTop:"15px",border:"none",marginBottom:"7px"}} className="order\_7">**

**<ListGroup style={{width:"230px"}}>**

**<ListGroup.Item style={{height:"50px",marginBottom:"18px",border:"none"}} className="order\_6"><imgsrc="./image/smile.svg" style={{marginRight:"22px"}}/>+91-65768768</ListGroup.Item>**

**<ListGroup.Item style={{height:"50px",marginBottom:"5px",border:"none",color:"#03A305"}} className="order\_6"><imgsrc="./image/order.svg" style={{marginRight:"22px"}}/>My Orders</ListGroup.Item>**

**<ListGroup.Item style={{height:"50px",marginBottom:"5px",border:"none"}} className="order\_6"><imgsrc="./image/place.svg"style={{marginRight:"22px"}}/>My Addresses</ListGroup.Item>**

**<ListGroup.Item style={{height:"50px",marginBottom:"5px",border:"none"}} className="order\_6"><imgsrc="./image/logout.svg" style={{marginRight:"22px"}}/>Logout</ListGroup.Item>**

**</ListGroup>**

**</ListGroup.Item>**

**</ListGroup>**

**<ListGroup style={{width:"788px",marginRight:"139px",marginTop:"15px",marginLeft:"10px"}}>**

**<ListGroup.Item style={{height:"88px",paddingTop:"3px",border:"none"}} className="order\_8">**

**<p className="order\_1">1 February 2021</p>**

**<p style={{marginBottom:"0px"}}><span className="order\_2">5 items</span><span style={{marginLeft:"480px"}} className="order\_4">Repeat</span><span style={{marginLeft:"32px"}} className="order\_5">Remove</span></p>**

**<p className="order\_3" style={{marginBottom:"0px"}}> Total Amount: rs 615</p>**

**</ListGroup.Item>**

**</ListGroup>**

**</ListGroup>**

**</>**

**)**

**}**

**export default Myorders**

**Payment.jsx**

**import React, { useState } from "react";**

**import {ListGroup,Button,Modal} from "react-bootstrap";**

**import {} from "../../node\_modules/bootstrap/dist/css/bootstrap.min.css";**

**import { verified } from "../lib/utils";**

**//import {googlepay, money,creditcard,paytm,centerimage,verify} from "../lib/utils";**

**const Payment=()=>{**

**const [modalShow, setModalShow] = React.useState(false);**

**function MyVerticallyCenteredModal(props) {**

**return (**

**<Modal style={{marginLeft:"455px",marginRight:"455px",width:"390px"}}**

**{...props}**

**size="lg"**

**aria-labelledby="contained-modal-title-vcenter"**

**centered**

**>**

**<Modal.Header style={{border:"none"}}>**

**<Modal.Title id="contained-modal-title-vcenter" >**

**Order Confirmed**

**</Modal.Title>**

**</Modal.Header>**

**<Modal.Body>**

**<imgsrc="./image/verified.svg" className="paysix"/>**

**{/\* <Button variant="success" style={{margin:"auto"}} onClick={props.onHide}>Continue Shopping</Button> \*/}**

**<Button variant="success" size="lg" style={{padding:"3px",width:"230px",marginLeft:"auto",marginRight:"auto",display:"block",marginTop:"20px"}}>**

**<span className="getotp" style={{ margin: "auto" }}>Continue Shopping</span>**

**</Button>**

**</Modal.Body>**

**</Modal>**

**);**

**}**

**return(**

**<>**

**<div style={{maxHeight:"45px",marginBottom:"10px",marginTop:"20px"}}>**

**<ListGroup horizontal className="nchange\_13"style={{marginLeft:"139px",marginRight:"139px"}} >**

**<ListGroup.ItemclassName="listing" style={{border:"none",borderBottom:"5px solid green",color:"green"}}>My cart</ListGroup.Item>**

**<ListGroup.ItemclassName="listing" style={{border:"none",borderBottom:"5px solid green",color:"green"}}>Delivery Address</ListGroup.Item>**

**<ListGroup.ItemclassName="listing" style={{border:"none",borderBottom:"5px solid green",color:"green"}}>Date & Time</ListGroup.Item>**

**<ListGroup.ItemclassName="listing" style={{border:"none",borderBottom:"5px solid green",color:"green"}}>Payment</ListGroup.Item>**

**</ListGroup>**

**<ListGroup>**

**<ListGroup.ItemclassName="bigcontent" variant="light" style={{marginLeft:"139px",marginRight:"139px",height:"550px",marginBottom:"10px",marginTop:"12px"}}>**

**<ListGroup horizontal style={{width:"350px",margin:"0px"}}>**

**<ListGroup.Item style={{border:"none"}}>**

**<p className="paytwo">Select Payment Method</p>**

**<button class="button button1" style={{padding:"16px 32px"}}><imgsrc="./image/Money.svg"/></button>**

**<p style={{marginLeft:"25px"}} className="payone">cash</p>**

**<button class="button button1" style={{padding:"16px 32px"}}><imgsrc="./image/Creditcard.svg"/></button>**

**<p style={{marginLeft:"10px"}} className="payone">Credit/Debitcard</p>**

**<button class="button button1" style={{padding:"16px 25px"}}><imgsrc="./image/Paytm.svg"/></button>**

**<p style={{marginLeft:"25px"}} className="payone">Paytm</p>**

**<button class="button button1" style={{padding:"16px 25px"}}><imgsrc="./image/Googlepay.svg"/></button>**

**<p style={{marginLeft:"25px"}} className="payone">Googlepay</p></ListGroup.Item>**

**<ListGroup.Item style={{border:"none"}}><imgsrc="./image/Payment.svg" className="paythree"/>**

**<Button variant="success" onClick={() =>setModalShow(true)} className="payfive" style={{marginTop:"100px",marginLeft:"480px",width:"200px",marginRight:"20px"}}><span className="payfour">Confirm Order </span></Button>{' '}**

**<MyVerticallyCenteredModal**

**show={modalShow}**

**onHide={() =>setModalShow(false)}**

**/>**

**</ListGroup.Item>**

**</ListGroup>**

**</ListGroup.Item>**

**</ListGroup>**

**</div>**

**</>**

**)**

**}**

**export default Payment**

**Placeorder.jsx**

**import React from "react";**

**import {ListGroup,Button} from "react-bootstrap";**

**//import { redlabel,share,deliver,cashon,doorstep, bucketimage, bucketimage\_1} from "../lib/utils";**

**const Placedorder=()=>{**

**return(**

**<>**

**<ListGroup>**

**<ListGroup.Item style={{width:"100%",height:"400px",marginTop:"18px"}}>**

**<ListGroup>**

**<ListGroup horizontal>**

**<ListGroup.Item style={{width:"120px",height:"360px",marginLeft:"40px",padding:"0px",marginRight:"12px",border:"none",marginTop:"16px"}}>**

**<ListGroup>**

**<ListGroup.Item style={{width:"110px",height:"80px"}}></ListGroup.Item>**

**<ListGroup.Item style={{width:"110px",height:"80px",marginTop:"12px"}}></ListGroup.Item>**

**<ListGroup.Item style={{width:"110px",height:"80px",marginTop:"12px"}}></ListGroup.Item>**

**<ListGroup.Item style={{width:"110px",height:"80px",marginTop:"12px"}}></ListGroup.Item>**

**</ListGroup>**

**</ListGroup.Item>**

**<ListGroup.Item style={{width:"380px",height:"360px",marginTop:"16px"}} className="placed\_11">**

**<imgsrc="./image/redlabel.svg" style={{width:"270px"}}/>**

**</ListGroup.Item>**

**<ListGroup.Item style={{width:"542px",height:"360px",marginTop:"16px",border:"none"}}>**

**<p className="placed\_1" style={{marginBottom:"8px"}}>Red Label Tea Leaf,1Kg<imgsrc="./image/share.svg" style={{marginLeft:"15px"}}/></p>**

**<p style={{marginBottom:"8px"}}><span className="placed\_2">₹ 476</span><span className="placed\_3">MRP ₹ 490</span><span className="placed\_4">Save ₹ 14</span></p>**

**<button class="button2 button3" style={{padding:"3px 22px"}}>1Kg</button>**

**<button class="button2 button3" style={{padding:"3px 10px"}}>500gm</button>**

**<button class="button2 button3" style={{padding:"3px 10px"}}>500gm</button>**

**<p style={{marginBottom:"10px",marginTop:"10px"}}>**

**<Button variant="success">Add to Cart<imgsrc="./image/bucketimage.svg" style={{marginLeft:"10px"}}/></Button>{' '}</p>**

**<p className="placed\_5" style={{marginBottom:"5px"}}>Services</p>**

**<p className="placed\_6" style={{marginBottom:"5px"}}><span><imgsrc="./image/deliver.svg" style={{marginRight:"10px"}}/> Schedule Your delivery</span>**

**<span><imgsrc="./image/cashon.svg" style={{marginRight:"10px",marginLeft:"35px"}} />Cash on delivery available</span></p>**

**<p className="placed\_6"><imgsrc="./image/doorstep.svg" style={{marginRight:"10px"}}/> Easy Doorstep Returns</p>**

**<p className="placed\_7" style={{marginBottom:"5px"}}>Description</p>**

**<p className="placed\_8">Enjoy and share a healthy cup of Brooke Bond Red Label tea view more<button onClick={Handlepage} style={{background:"none",border:"none"}}>view more</button></p>**

**</ListGroup.Item>**

**<ListGroup.Item style={{width:"252px",height:"360px",marginRight:"40px",marginTop:"16px",border:"none",padding:"0px"}} className="placed\_9">**

**<ListGroup>**

**<ListGroup.ItemclassName="placed\_10" style={{border:"none",font:"normal normal bold 20px/30px Nunito"}}>Beverages</ListGroup.Item>**

**<ListGroup.ItemclassName="placed\_10" style={{border:"none",marginTop:"17px",font:"normal normal 600 16px/30px Nunito"}} >Cold Drinks</ListGroup.Item>**

**<ListGroup.ItemclassName="placed\_10" style={{border:"none" ,marginTop:"5px",font:"normal normal 600 16px/30px Nunito"}}>Juices & Drinks</ListGroup.Item>**

**<ListGroup.ItemclassName="placed\_10" style={{border:"none" ,marginTop:"5px",font:"normal normal 600 16px/30px Nunito"}}>Healthy Energy Drinks</ListGroup.Item>**

**<ListGroup.ItemclassName="placed\_10" style={{border:"none" ,marginTop:"5px",font:"normal normal 600 16px/30px Nunito"}}>Spices</ListGroup.Item>**

**</ListGroup>**

**</ListGroup.Item>**

**</ListGroup>**

**</ListGroup>**

**</ListGroup.Item>**

**</ListGroup>**

**<ListGroup>**

**<ListGroup.Item style={{width:"100%",height:"420px",marginTop:"15px"}}>**

**<div className="row">**

**<div>**

**<h3 style={{ marginTop: '20px', marginLeft: '50px',float:"left"}} className="mainhead">Similar Products**

**<Button variant="success" style={{ float: 'right', marginRight: '20px',marginLeft:"900px" }}>View All</Button></h3></div>**

**{Dailydealsimage.map((val)=>{**

**return(**

**<Daily image={val.image}**

**name={val.name}**

**link={val.link}**

**price={val.price}**

**price1={val.price1}**

**price2={val.price2}**

**rating={val.rating}**

**offer={val.offer}/>**

**)**

**})}**

**</div>**

**</ListGroup.Item>**

**</ListGroup>**

**</>**

**)**

**}**

**export default Placedorder**

**CHAPTER 5**

**SNAPSHOTS**

**Home Page:**

**Graphical user interface, website

Description automatically generated**

**This is the front page of chhotu grocery**

**Fig. 5.1-Home Page:**

When customer visit the page it will come to this page which is shown in above figure.

**Admin Dashboard Page**

Graphical user interface, application, website

Description automatically generated

**Fig. 5.2-Admin Dashboard:**

**Login Page**

**Graphical user interface, text, application

Description automatically generated**

**Fig. 5.3-Login Page**

Here we input customer’s mobile number for making its unique id.

**Mobile Number verification**

A screenshot of a computer

Description automatically generated with medium confidence

**Fig. 5.4-Mobile Number Verification:**

Here otp verification code will come to verify the mobile number.

**Logged in successful**

**Graphical user interface, text, application

Description automatically generated**

**Fig. 5.5-Logged in successful:**

After verifying otp customer will come to this page.

**Manage Category Page**

Graphical user interface, application

Description automatically generated

**Fig. 5.6-Manage Category Page**

It is category here we select the type of product user wants to buy.

**Delivery Address**

**Graphical user interface, application, email

Description automatically generated**

**Fig. 5.7-Delivery Address**

Address of delivery the item

**Proceed to Pay**

**Graphical user interface, application

Description automatically generated**

**Fig. 5.8-Proceed to pay:**

Here we go for payment gateway.

**Selecting Date and Time**

**Graphical user interface, application

Description automatically generated**

**Fig. 5.9-Date & Time:**

Before the payment gateway we can choose the timing and date of delivery.

**Selecting Payment Method**

**Graphical user interface, application, website

Description automatically generated**

**Fig. 5.10-Selecting Payment Method:**

Here customer choose any payment option he/she is comfortable.

**Order confirmed**

**Graphical user interface, application

Description automatically generated**

**Fig.5.11-Order Confirmed:**

After payment the order is confirmed customer can continue the shopping or it can leave the page .

**CHAPTER 6**

**TESTING**

**6.1 INTRODUCTION**

Testing is the integral part of any System Development Life Cycle insufficient and interested application tends to crash and result in loss of economic and manpower investment besides user’s dissatisfaction and downfall of reputation.

“Software Testing can be looked upon as one among much process, an organization performs, and that provides the last opportunity to correct any flaws in the developed system. Software Testing includes selecting test data that have more probability of giving errors.” The first step in System testing is to develop the plan that all aspect of system .Complements, Correctness, Reliability and Maintainability.

Software is to be tested for the best quality assurance, an assurance that system meets the specification and requirement for its intended use and performance.

System Testing is the most useful practical process of executing the program with the implicit intention of finding errors that makes the program fail.

**6.2 Types of Testing**

**Black Box (Functional) Testing:**

Testing against specification of system or components. Study it by examining its inputs and related outputs. Key is to devise inputs that have a higher likelihood of causing outputs that reveal the presence of defects. Use experience and knowledge of domain to identify such test cases. Failing this a systematic approach may be necessary. Equivalence partitioning is where the input to a program falls into a number of classes, e.g. positive numbers vs. negative numbers. Programs normally behave the same way for each member of a class. Partitions exist for both input and output. Partitions may be discrete or overlap. Invalid data (i.e. outside the normal partitions) is one or more partitions that should be tested.

Internal System design is not considered in this type of testing. Tests are based on requirements and functionality.

This type of test case design method focuses on the functional requirements of the software, ignoring the control structure of the program. Black box testing attempts to find errors in the following categories:

* Incorrect or missing functions.
* Interface errors.
* Errors in data structures or external database access.
* Performance errors.
* Initialization and termination errors.

**White Box (Structural) Testing:**

Testing based on knowledge of structure of component (e.g. by looking at source code). Advantage is that structure of code can be used to find out how many test case need to be performed. Knowledge of the algorithm (examination of the code) can be used to identify the equivalence partitions. Path testing is where the tester aims to exercise every independent execution path through the component. All conditional statements tested for both true and false cases. If a unit has no control statements, there will be up to 2n possible paths through it. This demonstrates that it is much easier to test small program units than large ones. Flow graphs are a pictorial representation of the paths of control through a program (ignoring assignments, procedure calls and I/O statements). Use flow graph to design test cases that execute each path. Static tools may be used to make this easier in programs that have a complex branching structure. Tools support. Dynamic program analyzers instrument a program with additional code. Typically this will count how many times each statement is executed. At end print out report showing which statements have and have not been executed. Problems with flow graph derived testing:

# Data complexity could not take into account.

# We cannot test all paths in combination.

# In really only possible at unit and module testing stages because beyond that complexity is too high.

This testing is based on knowledge of the internal logic of an application’s code. Also known as a Glass Box Testing .Internal software and code working should be known for this type of testing. Tests are based on coverage of code statements, branches, paths, conditions.

**Unit Testing:**

Unit testing concentrates on each unit of the software as implemented in the code. This is done to check syntax and logical errors in programs. At this stage, the test focuses on each module individually, assuring that it functions properly as a unit. In our case, we used extensive white-box testing at the unit testing stage.

A developer and his team typically do the unit testing do the unit testing is done in parallel with coding; it includes testing each function and procedure.

**Incremental Integration Testing:**

Bottom up approach for testing i.e. continuous testing of an application as new functionality is added; Application functionality and modules should be independent enough to test separately done by programmers or by testers.

**Integration Testing:**

Testing of integration modules to verify combined functionality after integration .Modules are typically code modules, individual applications, client and server and distributed systems.

**Functional Testing:**

This type of testing ignores the internal parts and focus on the output is as per requirement or not .Black box type testing geared to functionality requirements of an application.

**System Testing:**

Entire system is tested as per the requirements. Black box type test that is based on overall requirement specifications covers all combined parts of a system.

**CHAPTER 7**

**CONCLUSION &FUTURE SCOPE**

**7.1 CONCLUSION-**

The project entitled “Online Chotu Grocery” is developed using HTML, CSS and Bootstrap as front end and Javascript, Sqlite database in back end to computerize the process of online management of news post. This project covers only the basic features required

**7.2 FUTURE SCOPE-**

Thisweb application involves almost all the features of the online news posting. The future implementation will be online help for the users and chatting with website administrator.

**CHAPTER 8**

**REFERENCES**

1. Trost and A. Žemva, "Configurable hardware components generator in Javascript," *2015 4th Mediterranean Conference on Embedded Computing (MECO)*, 2015, pp. 96-99, doi: 10.1109/MECO.2015.7181876.
2. A. Kumar and S. P. Panda, "A Survey: How Javascript Pitches in IT-World," *2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)*, 2019, pp. 248-251, doi: 10.1109/COMITCon.2019.8862251
3. Y. Liu, "JSOptimizer: An Extensible Framework for JavaScript Program Optimization," *2019 IEEE/ACM 41st International Conference on Software Engineering: Companion Proceedings (ICSE-Companion)*, 2019, pp. 168-170, doi: 10.1109/ICSE-Companion.2019.00069.
4. H. Park, W. Jung and S. Moon, "Javascript ahead-of-time compilation for embedded web platform," *2015 13th IEEE Symposium on Embedded Systems For Real-time Multimedia (ESTIMedia)*, 2015, pp. 1-9, doi: 10.1109/ESTIMedia.2015.7351768.
5. S. Delcev and D. Draskovic, "Modern JavaScript frameworks: A Survey Study," *2018 Zooming Innovation in Consumer Technologies Conference (ZINC)*, 2018, pp. 106-109, doi: 10.1109/ZINC.2018.8448444.
6. M. Akbar, F. N. Azizah and G. A. P. Saptawati, "Integration of HTML tables in web pages," *2015 International Conference on Data and Software Engineering (ICoDSE)*, 2015, pp. 132-137, doi: 10.1109/ICODSE.2015.7436985.
7. S. -. Lim and Y. -. Ng, "Extracting structures of HTML documents," *Proceedings Twelfth International Conference on Information Networking (ICOIN-12)*, 1998, pp. 420-426, doi: 10.1109/ICOIN.1998.648420
8. Y. C. Chou and H. C. Liao, "A Webpage Data Hiding Method by Using Tag and CSS Attribute Setting," *2014 Tenth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, 2014, pp. 122-125, doi: 10.1109/IIH-MSP.2014.37.
9. Vamsi Krishna Myalapalli and BhupatiLohith Ravi Teja, "High performance PL/SQL programming," *2015 International Conference on Pervasive Computing (ICPC)*, 2015, pp. 1-5, doi: 10.1109/PERVASIVE.2015.7087001.
10. T. Q. Dam, S. Cheon and Y. Won, "On the IO Characteristics of the SQLite Transactions," *2016 IEEE/ACM International Conference on Mobile Software Engineering and Systems (MOBILESoft)*, 2016, pp. 214-224, doi: 10.1109/MobileSoft.2016.047.
11. J. Chou, L. Chen, H. Ding, J. Tu and B. Xu, "A Method of Optimizing React JS Based on Greedy Strategy," *2013 10th Web Information System and Application Conference*, 2013, pp. 176-179, doi: 10.1109/WISA.2013.41.
12. T. Naumovic, M. Despotovic-Zrakic, B. Radenkovic, L. Zivojinovic and I. Jezdovic, "Development of a Continuous System Simulation Engine in Javascript Programing Language," *2020 19th International Symposium INFOTEH-JAHORINA (INFOTEH)*, 2020, pp. 1-5, doi: 10.1109/INFOTEH48170.2020.9066334.
13. F. Rigueira, J. Bernardino and I. Pedrosa, "Extraction of information from log files Using Javascript Programming and Tableau," *2020 15th Iberian Conference on Information Systems and Technologies (CISTI)*, 2020, pp. 1-7, doi: 10.23919/CISTI49556.2020.9140844.
14. M. A. P. Subali and S. Rochimah, "A new model for measuring the complexity of SQL commands," *2018 10th International Conference on Information Technology and Electrical Engineering (ICITEE)*, 2018, pp. 1-5, doi: 10.1109/ICITEED.2018.8534782.
15. Y. Guo, N. Li, J. Offutt and A. Motro, "Automatically Repairing SQL Faults," *2018 IEEE International Conference on Software Quality, Reliability and Security (QRS)*, 2018, pp. 500-511, doi: 10.1109/QRS.2018.00063.