GROCERY MANAGEMENT SYSTEM

Problem Statement: Build a Model of Grocery Management chain where enormous amounts of data are collected and that need to be stored efficiently and suitable frontend for the ease of use.

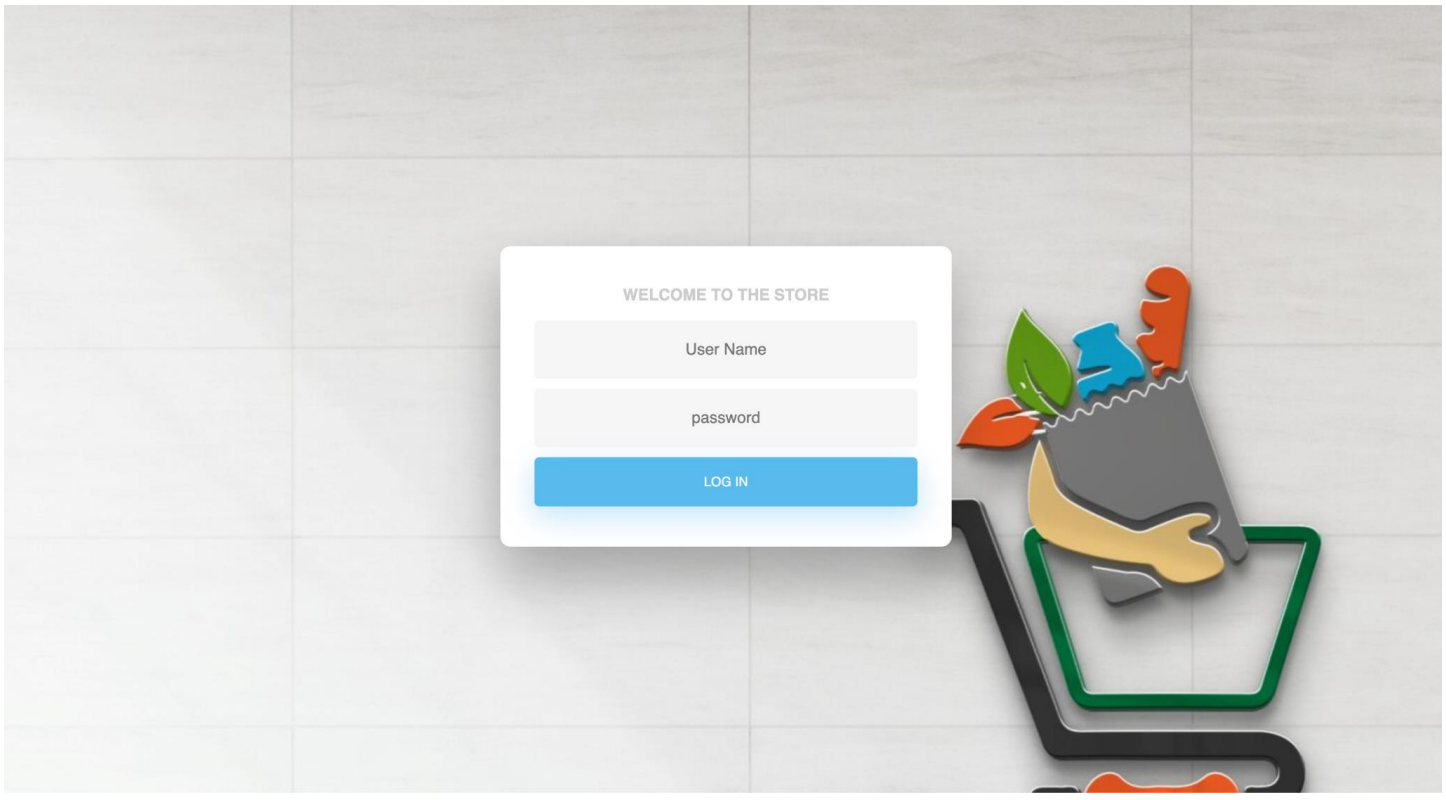
Front - End Design and Implementation

We in the grocery management system used NodeJS and EXPRESS servers for the implementation of backend and used HTML,CSS,BOOTSTRAP for the beautiful simple front-end. This design and implementation is focussed on making the system easy to understand and easy to use.

Details

The queries are used in server side for update,select,deleting data.

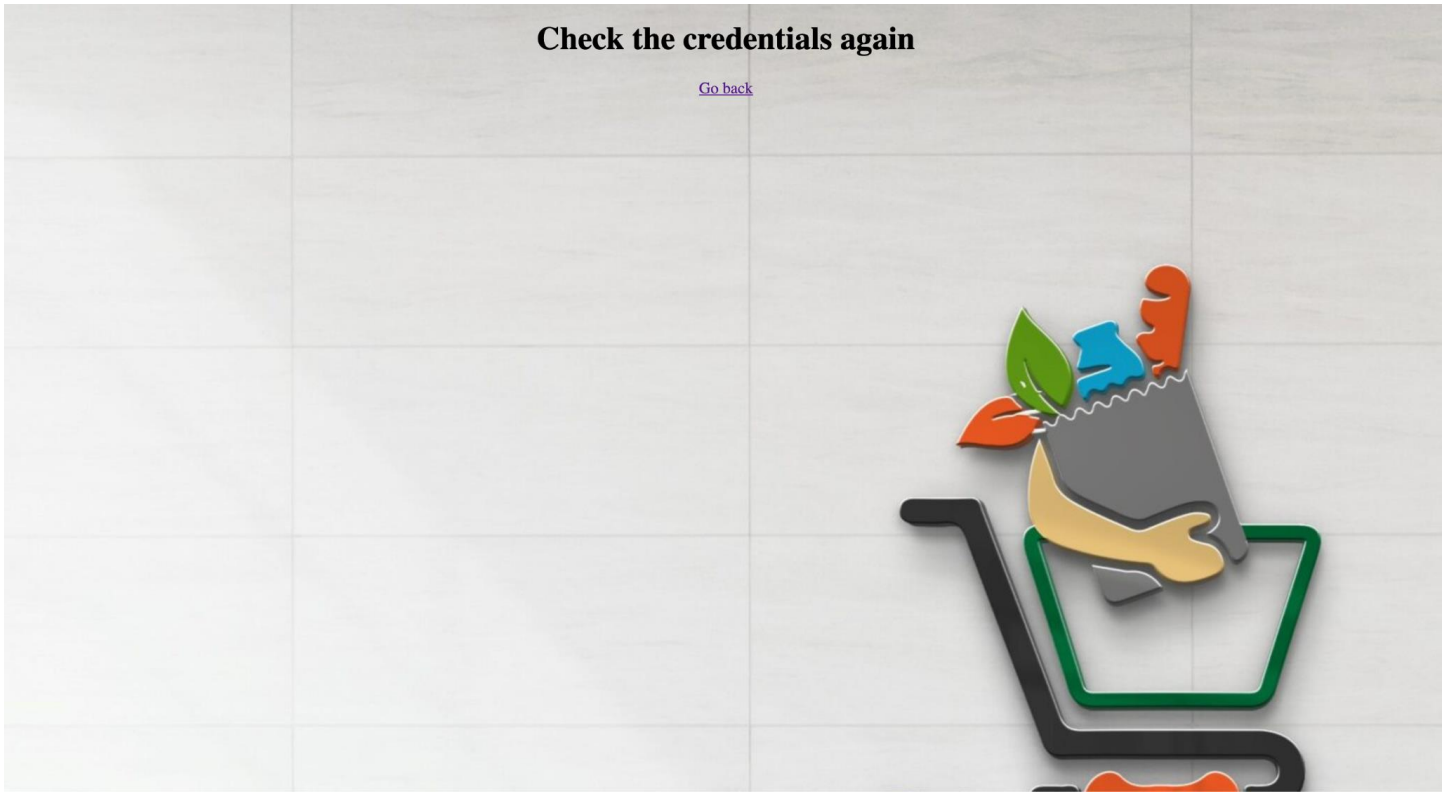
1.LOGIN PAGE



There are 2 types of users who use this system,

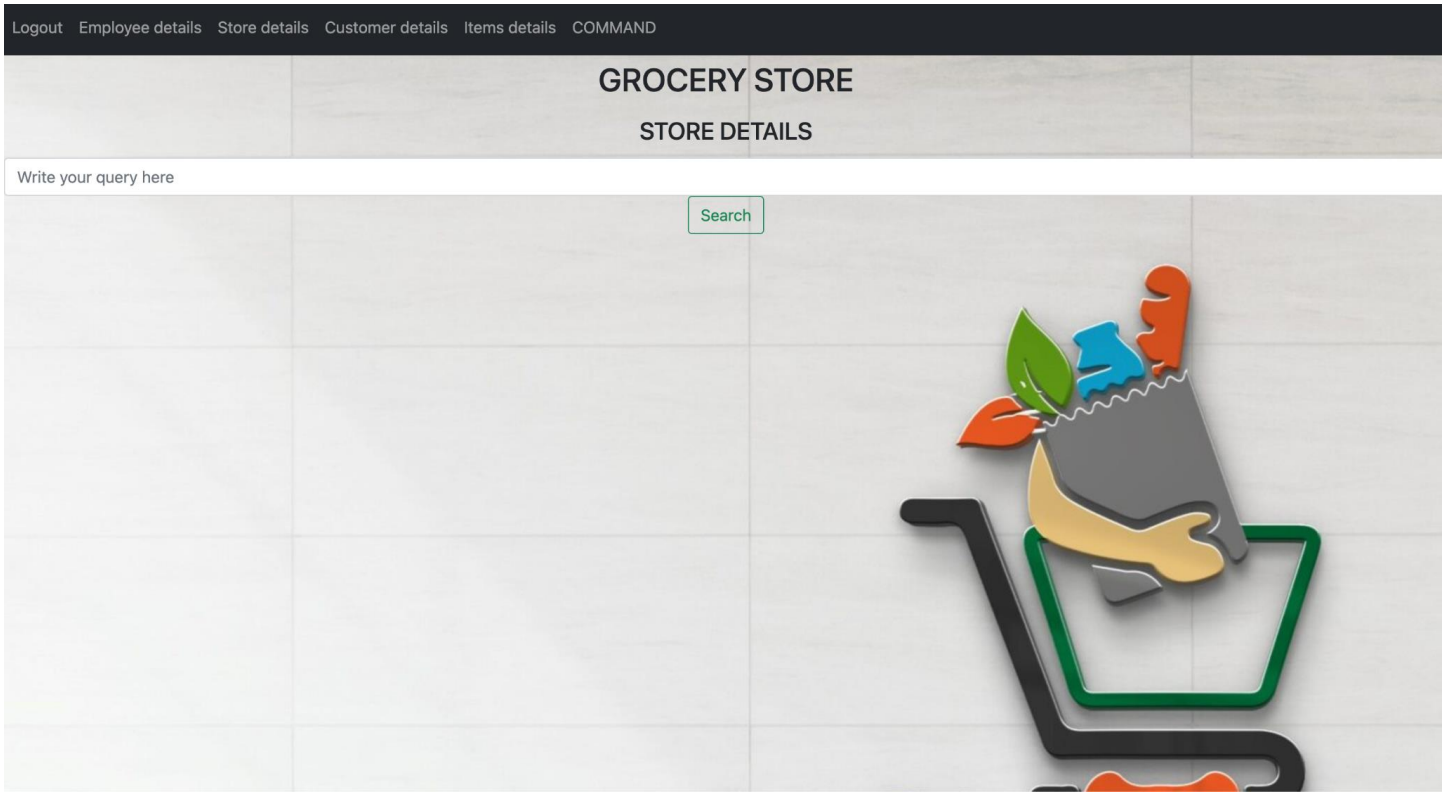
- Employees at checkout desk
- Manager

Users has to login through this page with proper credentials, any mismatch results in failure of connection to the database and shows the error message

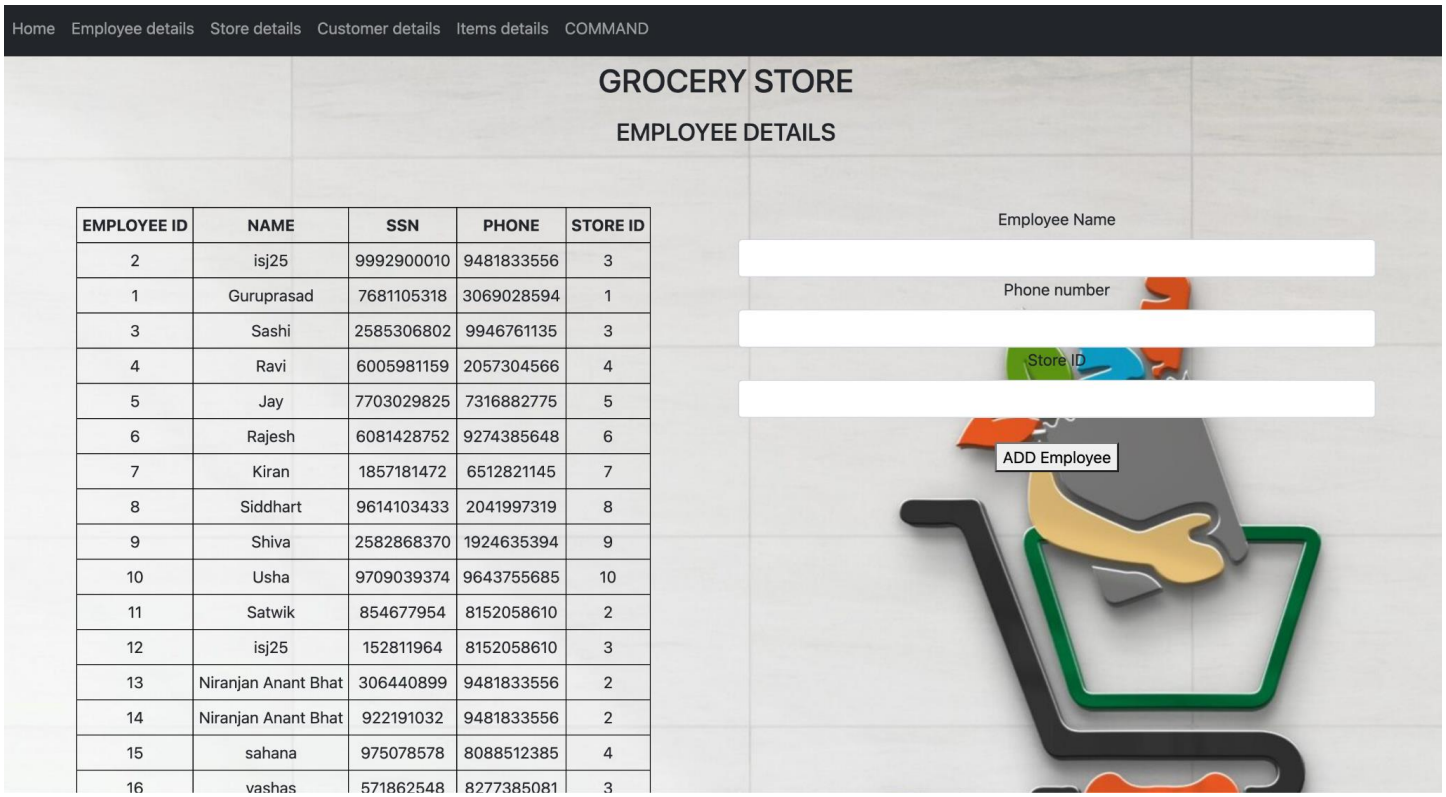


Manager:

After successful login, For manager a page with many options related to the store is shown and he can navigate between those pages to update and view the data stored.

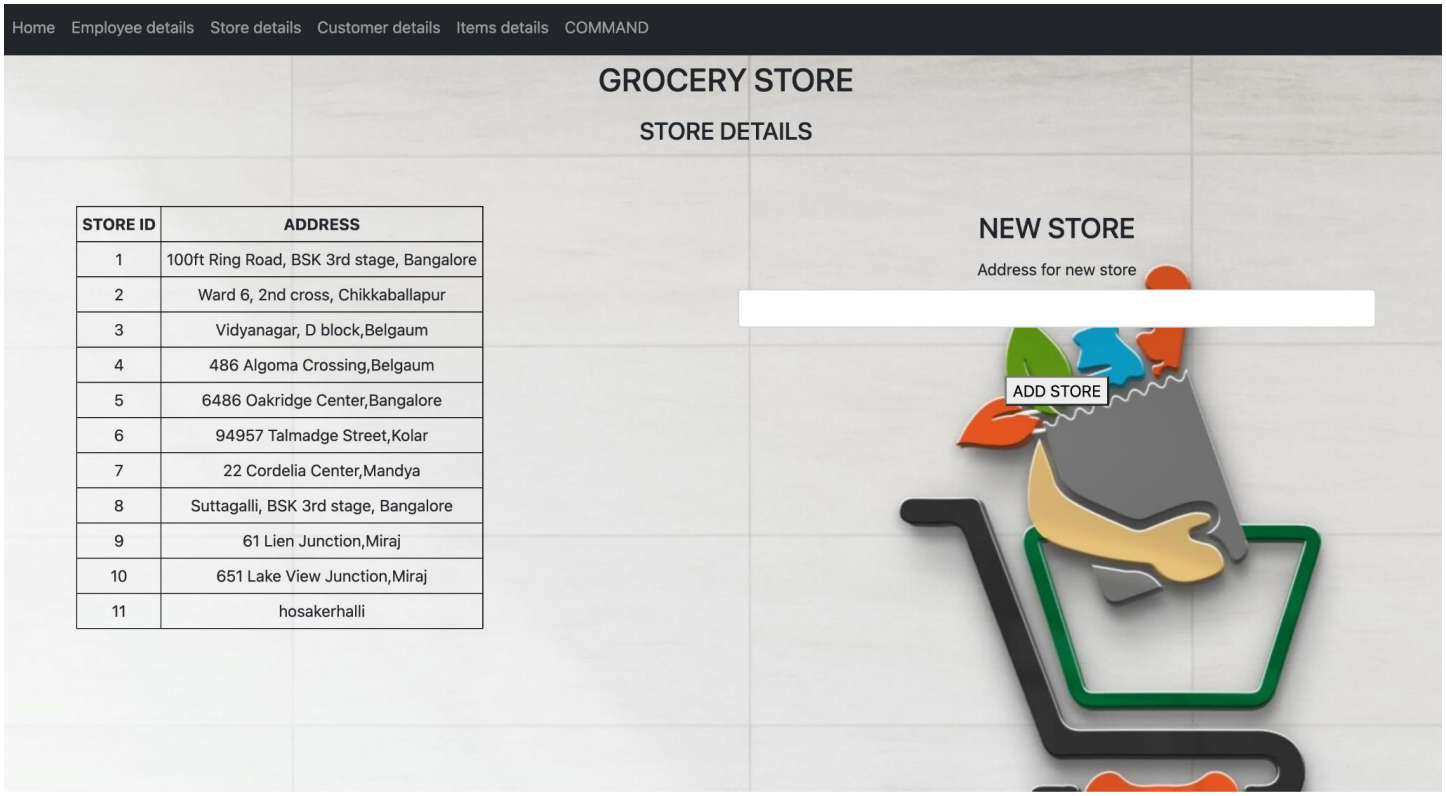


Manager has the option to Add employees and remove employees from the work and he can update the database with ease.



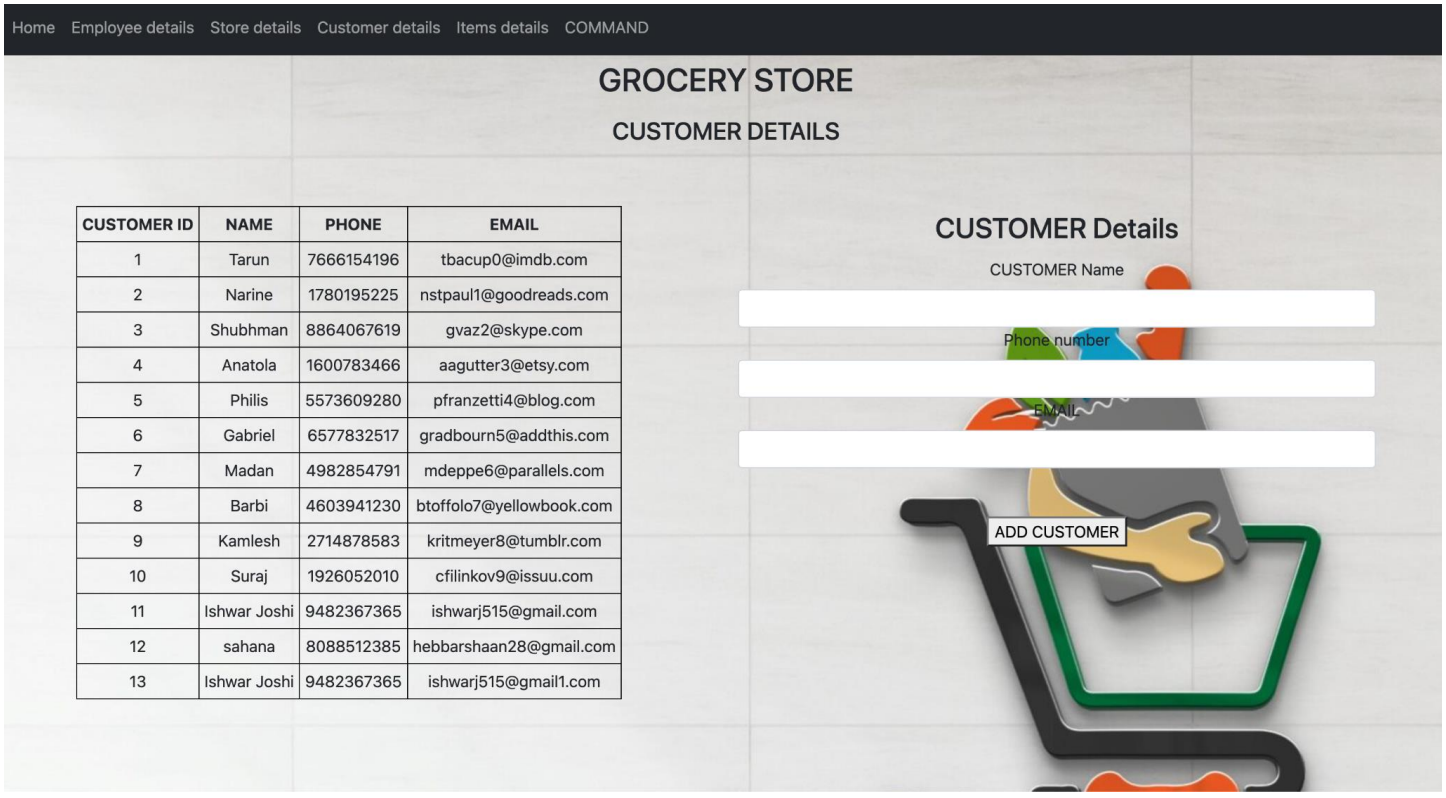
EMPLOYEE ID	NAME	SSN	PHONE	STORE ID
2	isj25	9992900010	9481833556	3
1	Guruprasad	7681105318	3069028594	1
3	Sashi	2585306802	9946761135	3
4	Ravi	6005981159	2057304566	4
5	Jay	7703029825	7316882775	5
6	Rajesh	6081428752	9274385648	6
7	Kiran	1857181472	6512821145	7
8	Siddhart	9614103433	2041997319	8
9	Shiva	2582868370	1924635394	9
10	Usha	9709039374	9643755685	10
11	Satwik	854677954	8152058610	2
12	isj25	152811964	8152058610	3
13	Niranjan Anant Bhat	306440899	9481833556	2
14	Niranjan Anant Bhat	922191032	9481833556	2
15	sahana	975078578	8088512385	4
16	vashas	571862548	8277385081	3

He also has the option to add new stores to list and view already existing stores.



STORE ID	ADDRESS
1	100ft Ring Road, BSK 3rd stage, Bangalore
2	Ward 6, 2nd cross, Chikkaballapur
3	Vidyanagar, D block,Belgaum
4	486 Algoma Crossing,Belgaum
5	6486 Oakridge Center,Bangalore
6	94957 Talmadge Street,Kolar
7	22 Cordelia Center,Mandya
8	Suttagalli, BSK 3rd stage, Bangalore
9	61 Lien Junction,Miraj
10	651 Lake View Junction,Miraj
11	hosakerhalli

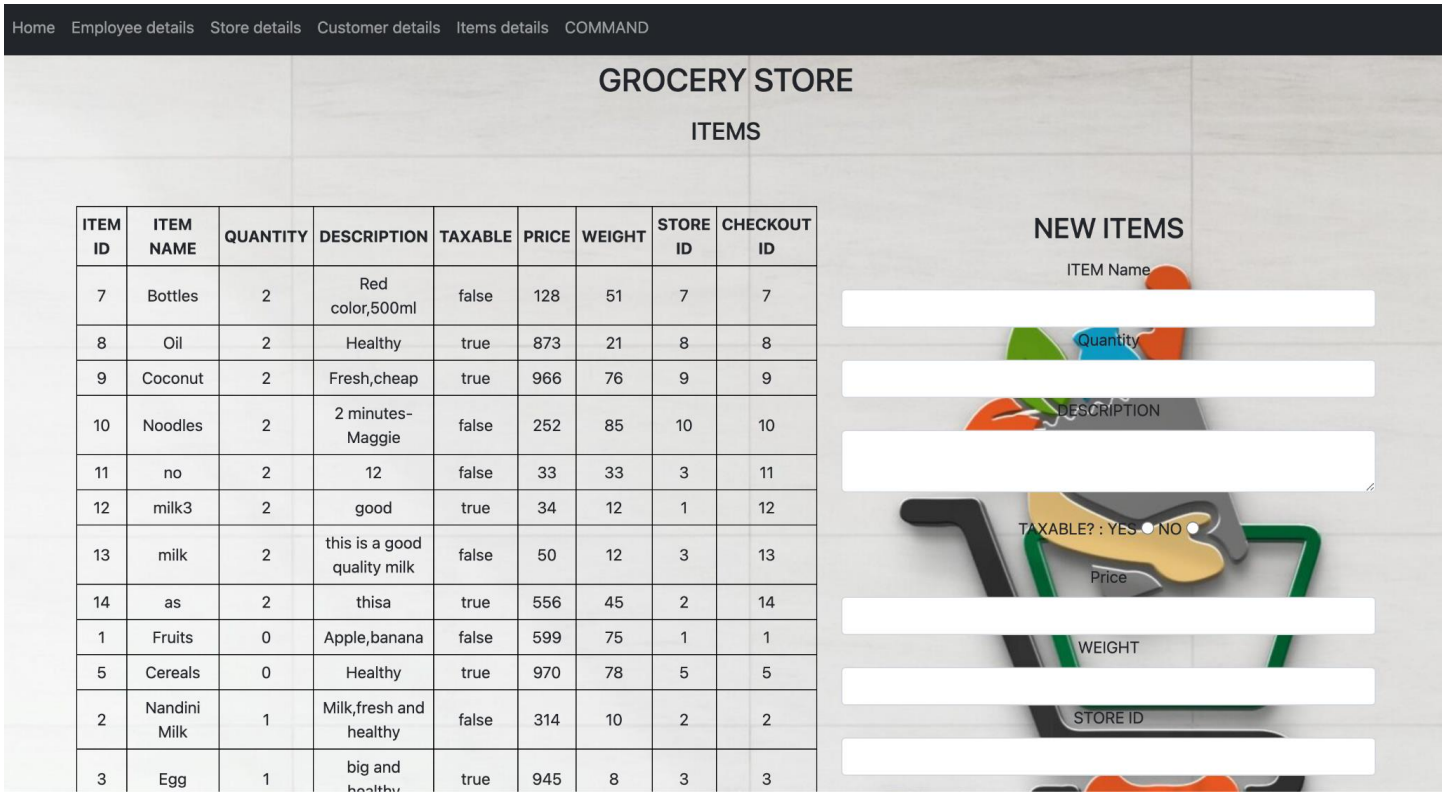
By this simple front-end manager can easily maintain the database of customers and details related to them,and can use them for promotion or advertisement purposes.



CUSTOMER ID	NAME	PHONE	EMAIL
1	Tarun	7666154196	tbacup0@imdb.com
2	Narine	1780195225	nstpaul1@goodreads.com
3	Shubhman	8864067619	gvaz2@skype.com
4	Anatola	1600783466	aagutter3@etsy.com
5	Philis	5573609280	pfranzetti4@blog.com
6	Gabriel	6577832517	gradbourn5@addthis.com
7	Madan	4982854791	mdeppe6@parallels.com
8	Barbi	4603941230	btoffolo7@yellowbook.com
9	Kamlesh	2714878583	kritmeyer8@tumblr.com
10	Suraj	1926052010	cfilinkov9@issuu.com
11	Ishwar Joshi	9482367365	ishwarj515@gmail.com
12	sahana	8088512385	hebbarshaan28@gmail.com
13	Ishwar Joshi	9482367365	ishwarj515@gmail1.com

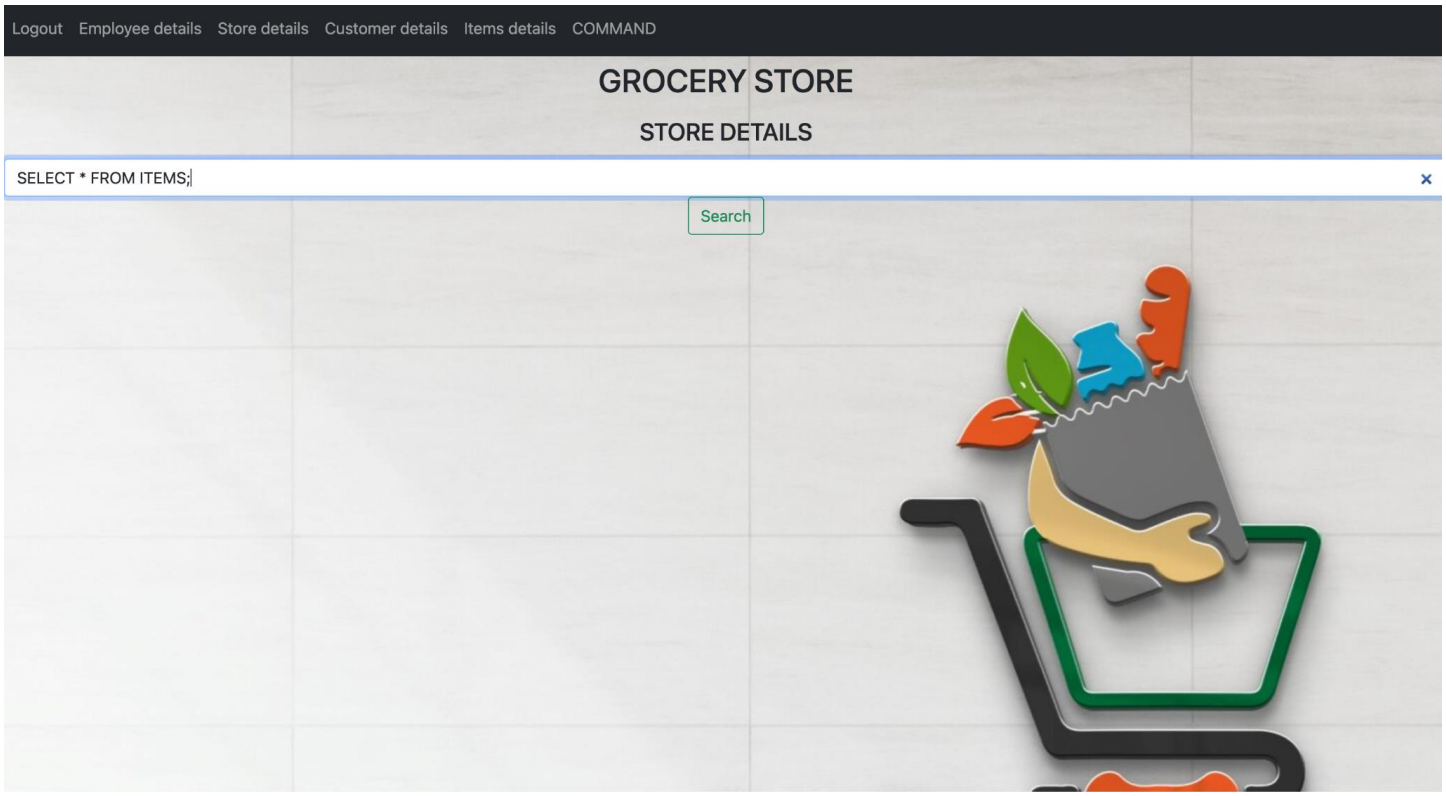
Item Details

Manager has the permission to add new items to the database with proper details.



ITEM ID	ITEM NAME	QUANTITY	DESCRIPTION	TAXABLE	PRICE	WEIGHT	STORE ID	CHECKOUT ID
7	Bottles	2	Red color,500ml	false	128	51	7	7
8	Oil	2	Healthy	true	873	21	8	8
9	Coconut	2	Fresh,cheap	true	966	76	9	9
10	Noodles	2	2 minutes-Maggie	false	252	85	10	10
11	no	2	12	false	33	33	3	11
12	milk3	2	good	true	34	12	1	12
13	milk	2	this is a good quality milk	false	50	12	3	13
14	as	2	thisa	true	556	45	2	14
1	Fruits	0	Apple,banana	false	599	75	1	1
5	Cereals	0	Healthy	true	970	78	5	5
2	Nandini Milk	1	Milk,fresh and healthy	false	314	10	2	2
3	Egg	1	big and	true	945	8	3	3

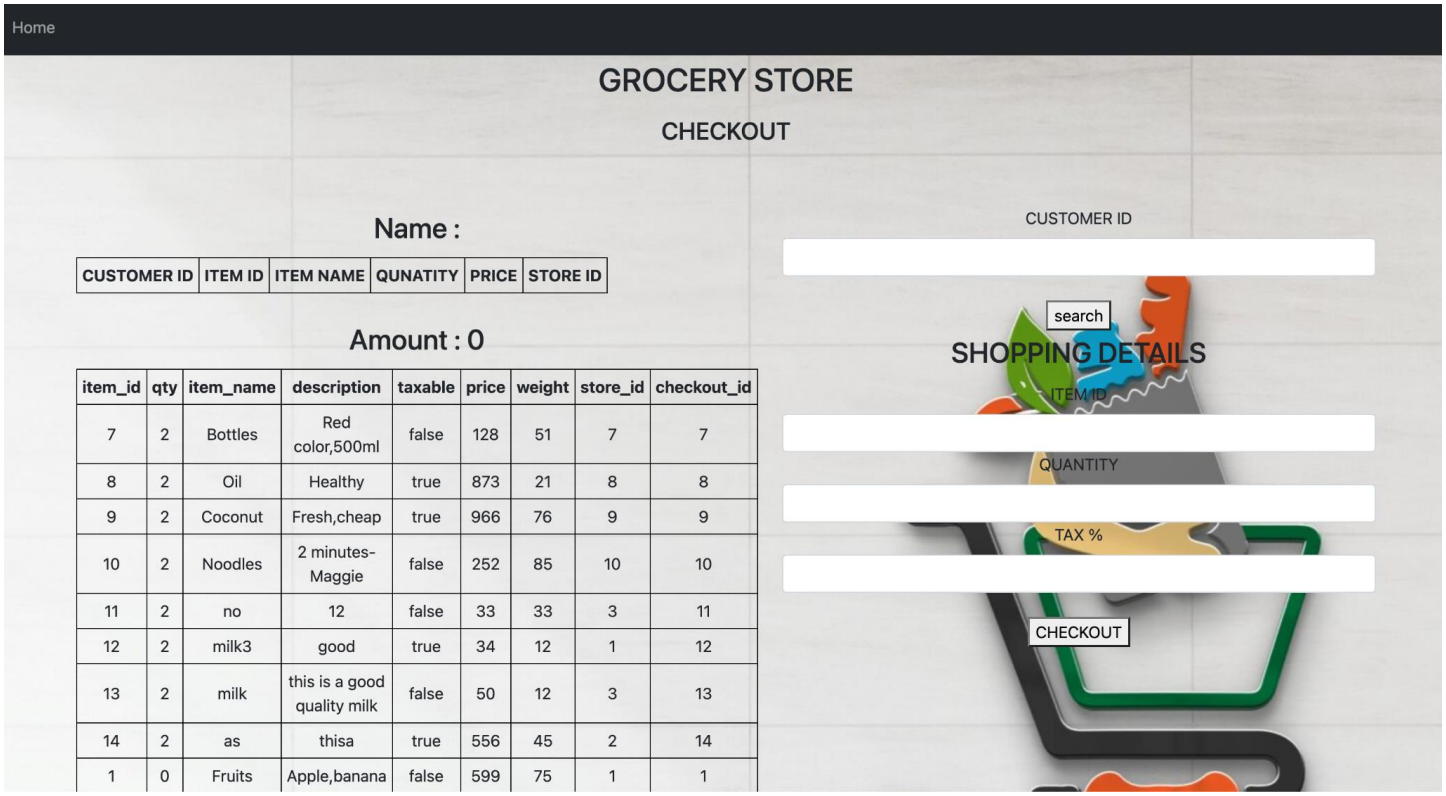
For the project demonstration purpose and to view actual database implementation an optional field of search bar is provided where one can write the query and obtain the results.



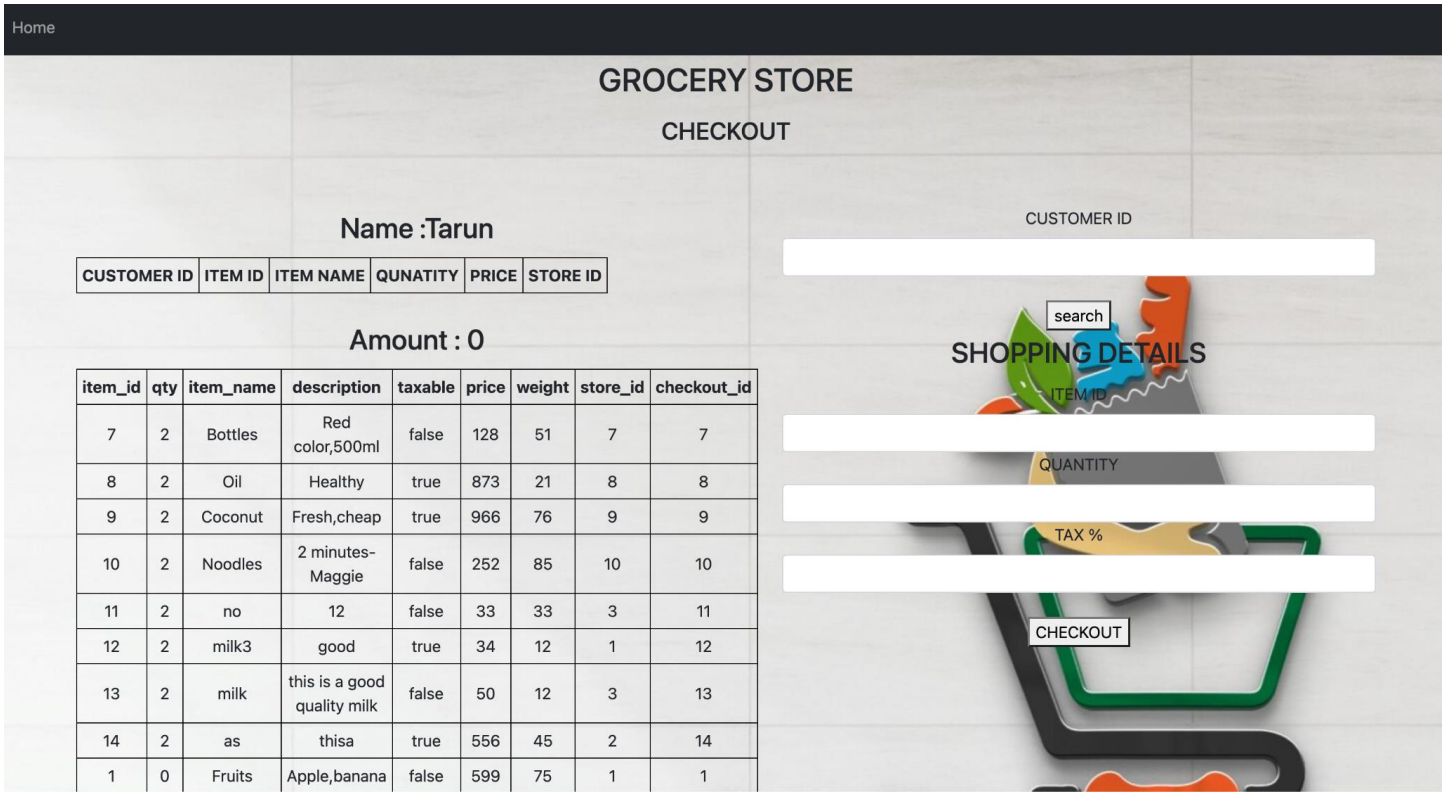
2.Checkout Desk

Employees at the checkout table have to login to the system with the credentials given to them.
Once the login is successful ,the page is redirected to the checkout page.

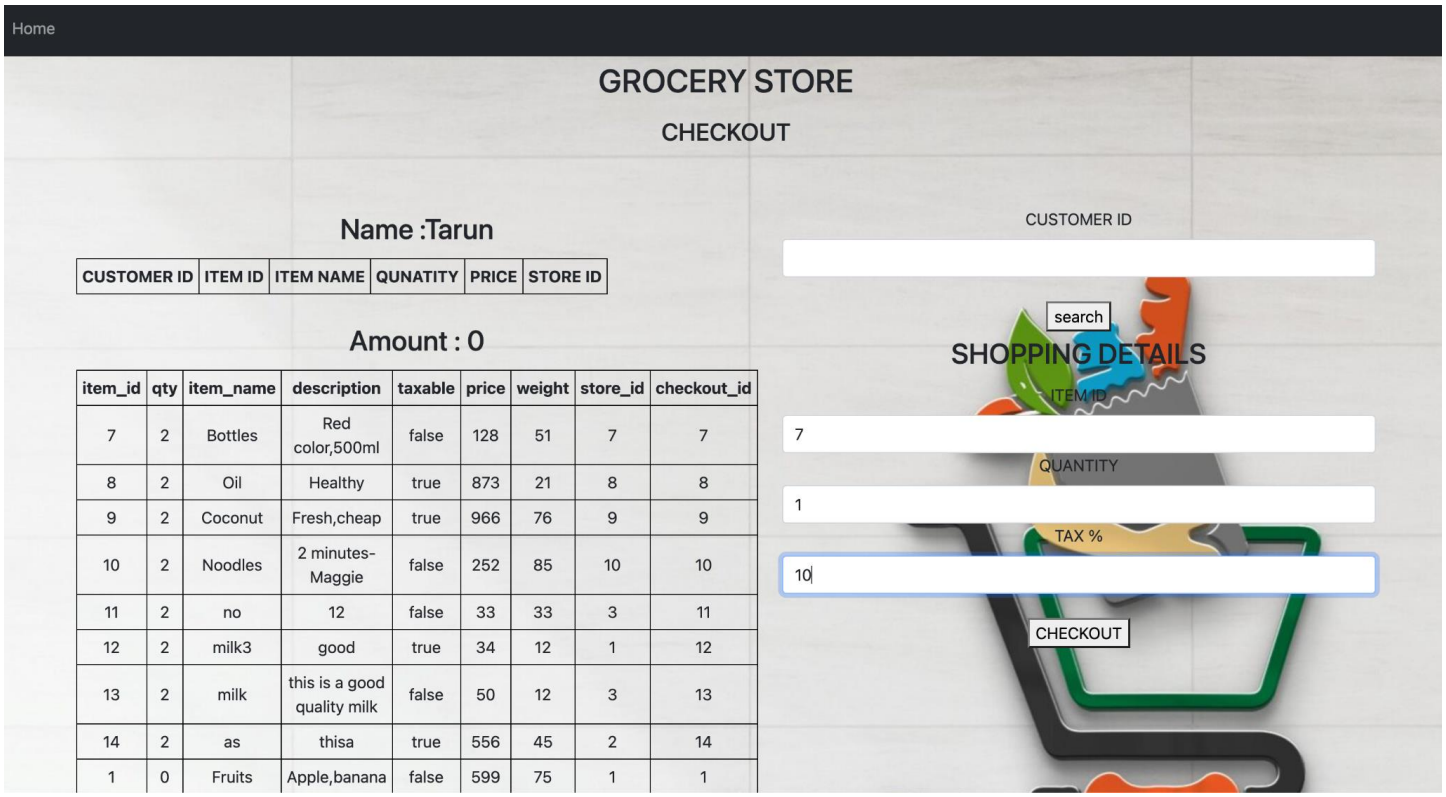
Checkout page:
This is the main working page of the system where a bill is prepared for the items that are bought by the customer. After every successful purchase of items, the database is updated and the available items are shown to the employees.



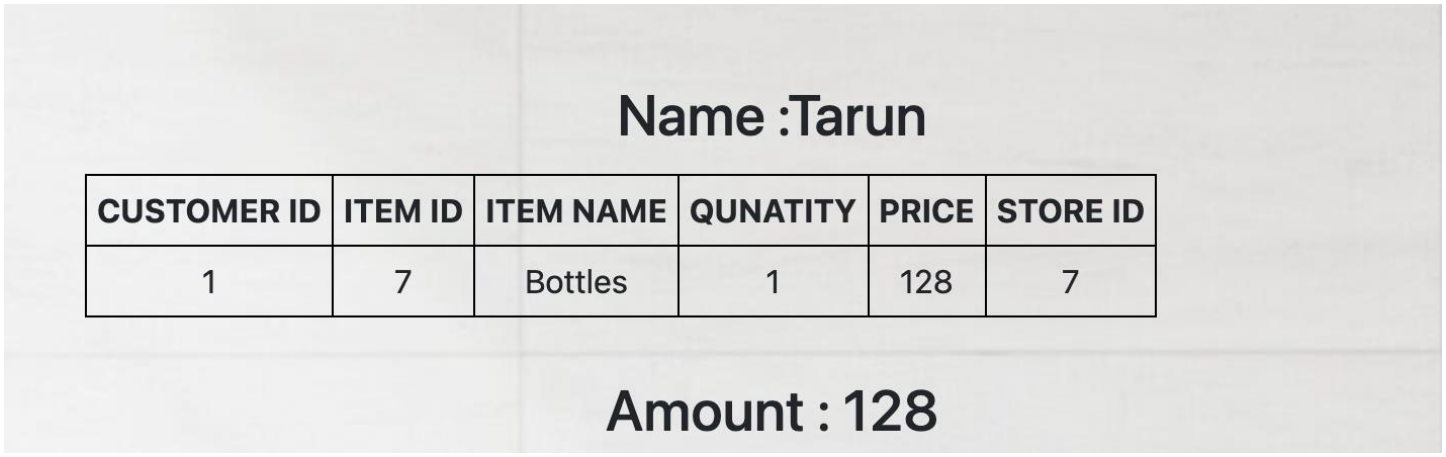
Checkout page



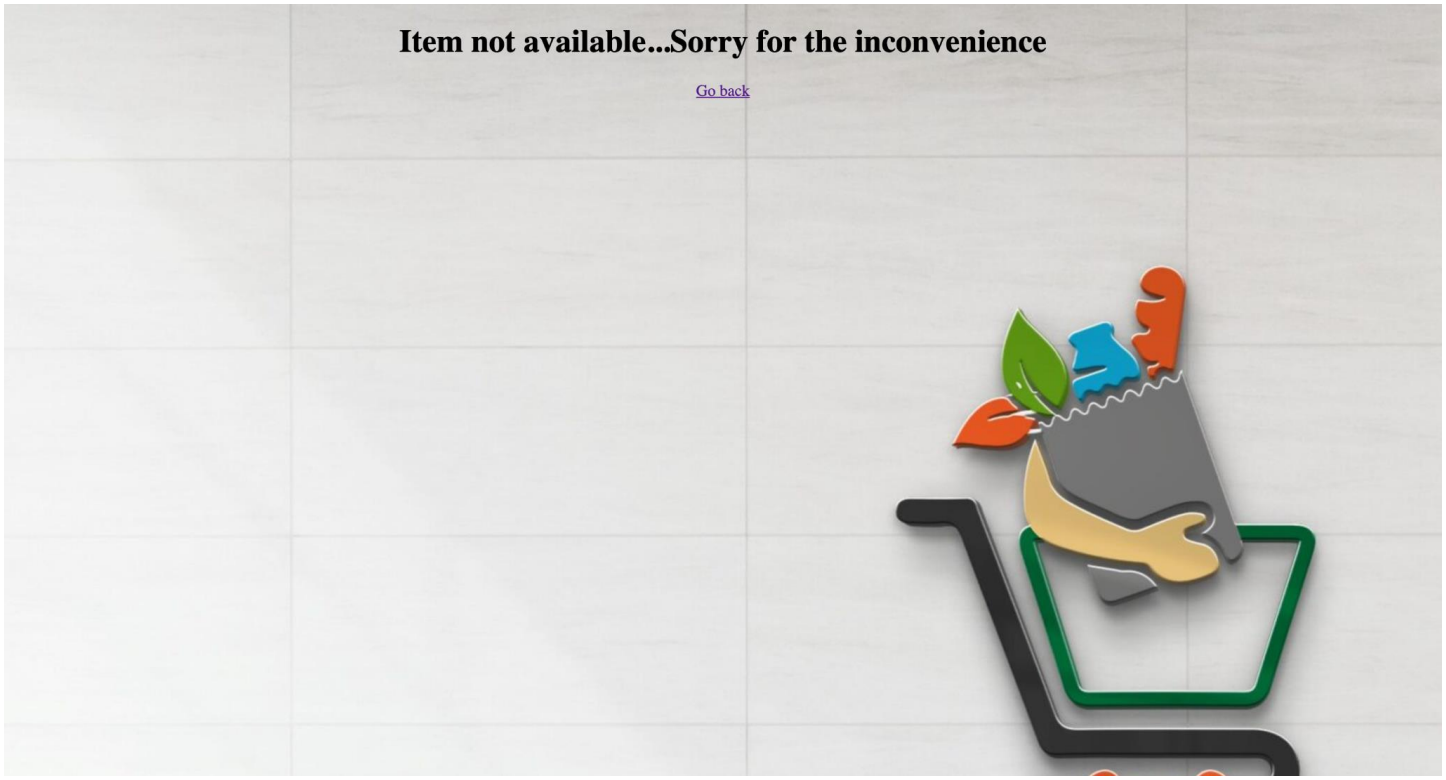
Customer name is retrieved from the database



Items are added to the bill



every item is billed and amount is calculated with given tax



Items not available

GROCERY STORE

STORE DETAILS

feedback_id	rating	cust_feedback	cust_id	item_id
1	7.5	Quality is not soo good	1	1
2	7.0	it was tasy	2	2
3	2.0	not worth	3	3
4	6.0	it was not so good	4	4
5	2.0	horrible	5	5
6	9.8	Fantastic	6	6
7	1.4	Bad	7	7
8	3.0	Normal	8	8
9	7.0	healthy and good	9	9
10	9.0	Awsome	10	10

Go back

Customer feedback

Changes in the constraints and schema

- Two tables(Dependents and finance) from the schema is dropped because of extra details of the table to this project.
- The employee table is slightly modified.
- Database permissions are modified
- checkout employees are given select permission on customer table
- checkout employee are given select and update permission on items table

```
create user manager with password 'manager@123';
grant select,insert,update,delete on all tables in schema public to manager;
revoke update on finance from manager;
-- create check1 with password "check1";

create user checkout1 with password 'checkout1';
grant select,insert,update,delete  on table checkout,cancellation to checkout1;
grant select on table customer to checkout1;
grant select,update on table items to checkout1;



create user checkout2 with password 'checkout2';
grant select,insert,update,delete on table checkout,cancellation to checkout2;
grant select on table customer to checkout2;
grant select,update on table items to checkout2;


create user checkout3 with password 'checkout3';
grant select,insert,update,delete  on table checkout,cancellation to checkout3;
grant select on table customer to checkout3;
grant select,update on table items to checkout3;



create user user1 with password 'user1';
grant update,select on table feedback to user1;
```

Database migration and support

The Postgresql database which is used to implement this system can handle large amounts of data.

Limit	Value
Maximum Table Size	32 TB
Maximum Row Size	1.6 TB
Maximum Field Size	1 GB
Maximum Rows per Table	Unlimited

This amount of data is not stored in the simple grocery mart system,hence migration of databases is not required.

If required then what? : Front-end interface must be kept the same as before and database schema and relations must be created in the new database which stores exactly the same data and there should be an option for modification.
Suitable migration support must be used as there is a risk of data loss.

Support for the database:
Postgresql provides good support for it’s customers ,for more information
https://www.postgresql.org/support/

Implementation Details

1.Web app is served using node js - express server.
command : npm install express;
index.js is the server file which is the core of the backend,which serves all the page according to the user click.

```
const express = require("express");
const bodyParser = require("body-parser");
const { urlencoded } = require("body-parser");

const { Pool, Client } = require('pg')
let pool;
const ejs = require("ejs");
const { json } = require("stream/consumers");
const app = express()
app.set("view engine","ejs");
app.use(bodyParser.urlencoded({extended:true}))
app.use(express.static('public'));


app.get("/",(req,res)=>{
    res.sendFile(__dirname+"/login.html");
});


app.post("/",async(req,res)=>{
    nm = req.body.loginid;
    psd = req.body.loginpassword;
    try{
        pool =  new Pool({
            user: nm,
            password: psd,
            host: "localhost",
            port: "5432",
            database: "grocery"
        });


        pool.connect((err) => {
            if (err) {
                res.render("error",{errormessage:"Check the credentials again"});
            }else
            {
                if(nm == "manager")
                {
                    res.redirect("/manager");
                }else
                {
                    res.redirect("/checkout");
```

2. body-parser

body parser is used to retrieve the input data which is entered by the user of the system.

```
const bodyParser = require("body-parser");
const { urlencoded } = require("body-parser");
```

3. Database connection

‘pg’ is used to connect backend server to the database server

```
const { Pool, Client } = require('pg')
let pool;
```

```
nm = req.body.loginid;
psd = req.body.loginpassword;
try{
    pool =  new Pool({
        user: nm,
        password: psd,
        host: "localhost",
        port: "5432",
        database: "grocery"
    });
```

Front End

Front end is designed using HTML CSS and BOOTSTRAP

```html
<!DOCTYPE html>
<html lang="en">
<head>

</head>
<title>login</title>
<link href="//maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css" rel="stylesheet" id="bootstrap-css">
<script src="//maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"></script>
<script src="//cdnjs.cloudflare.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
<link rel="stylesheet" href="styles.css">
<body>
    <!------ Include the above in your HEAD tag ---------->

    <div class="wrapper fadeInDown">
        <div id="formContent">
            <!-- Tabs Titles -->

            <!-- Icon -->
            <!-- <div class="fadeIn first">
                <img src="http://danielzawadzki.com/codepen/01/icon.svg" id="icon" alt="User Icon" />
            </div> -->

            <!-- Login Form -->


            <h2>WELCOME TO THE STORE </h2>


            <form method="post">
                <input type="text" id="login" class="fadeIn second" name="loginid" placeholder="User Name" autocomplete="off">
                <input type="password" id="password" class="fadeIn third" name="loginpassword" placeholder="password">
                <input type="submit" class="fadeIn fourth" value="Log In">
            </form>

            <!-- Remind Passowrd -->
            <!-- <div id="formFooter">
                <a class="underlineHover" href="#">Forgot Password?</a>
            </div> -->

        </div>
    </div>
```

HTML

```css
html {
  background-color: #56baed;
}

body {
  font-family: "Poppins", sans-serif;
  height: 100vh;
  background-image: url("g.jpg");
}

a {
  color: #92badd;
  display:inline-block;
  text-decoration: none;
  font-weight: 400;
}

h2 {
  text-align: center;
  font-size: 16px;
  font-weight: 600;
  text-transform: uppercase;
  display:inline-block;
  margin: 40px 8px 10px 8px;
  color: #cccccc;
}



/* STRUCTURE */

.wrapper {
  display: flex;
  align-items: center;
  flex-direction: column;
  justify-content: center;
  width: 100%;
  min-height: 100%;
  padding: 20px;
}
```

CSS

```html
<body style="background-image: url('g.jpg');">


    <nav class="navbar navbar-expand-lg navbar-dark bg-dark">

        <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarSupportedContent"
            aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">
            <span class="navbar-toggler-icon"></span>
        </button>

        <div class="collapse navbar-collapse" id="navbarSupportedContent">
            <ul class="navbar-nav mr-auto">
                <li class="nav-item active">
                    <a class="nav-link" href="/">Logout</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" href="/employee">Employee details</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" href="/store">Store details</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" href="/customer">Customer details</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" href="/items">Items details</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" href="/manager">COMMAND</a>
                </li>

            </ul>
        </div>
    </nav>
```

BOOTSTRAP

changes in Business/Application changes/expansion

advancement of this business application is to take all workings to the cloud platform so that system is available to all stores at all time. Instead of storing the data locally and processing it , storing it in the cloud helps for various future works.

PES1UG19CS174	GURUPRASAD N B
PES1UG19CS176	H V SHASHIKANTH REDDY
PES1UG19CS191	ISHWAR SITARAMA JOSHI