

Sri Lanka Institute of Information Technology



Distributed Systems

Assignment - 2

Student Name – E.M.Y.L Edirisooriya

Student ID- IT16161870

Architecture

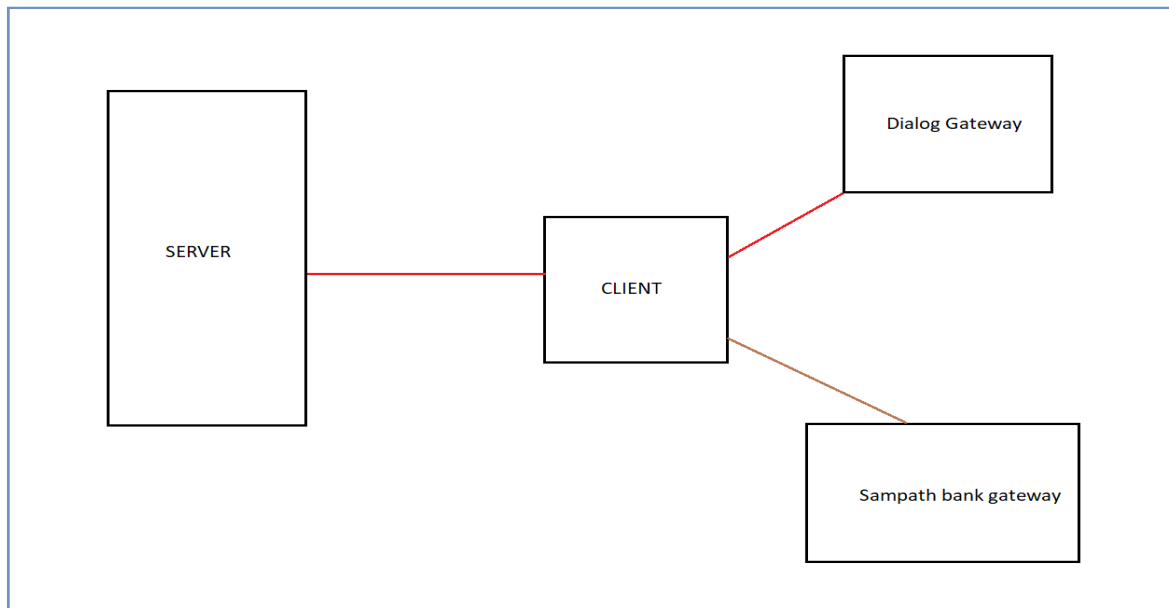


Figure1.13

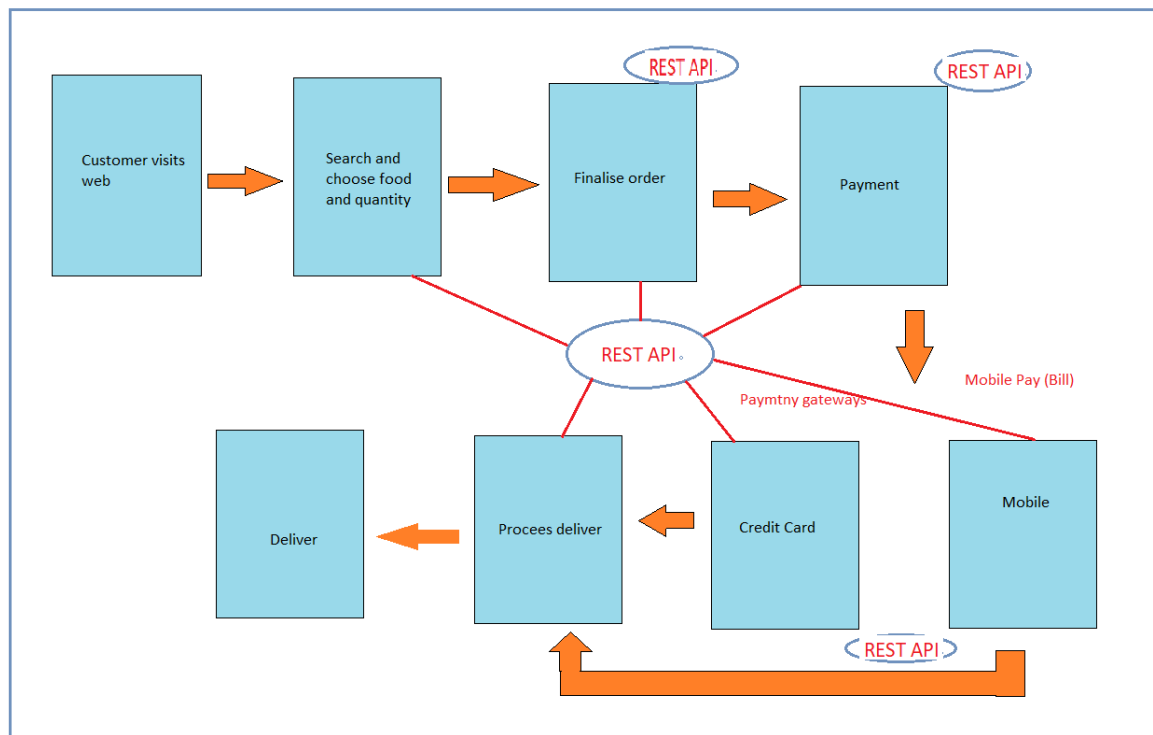


Figure1.2

Interfaces

Login

Username

admin


Password

...

SignUp

Login

Figure 1.1

EMYLs online fast food

Yasith Edirisooriya

logout

Admin Dashboard

Admin DashBoard

Food Items 🍴

Food	Price
Pastry	Rs.45.00
Vege Patis	Rs.30.00
Cake	Rs.40.00
Fish Roll	Rs.40.00
Pizza	Rs.340.00

Add Food

Food Item Name

Price Rs.

Add Food

Customer Dashboard

Figure1.2

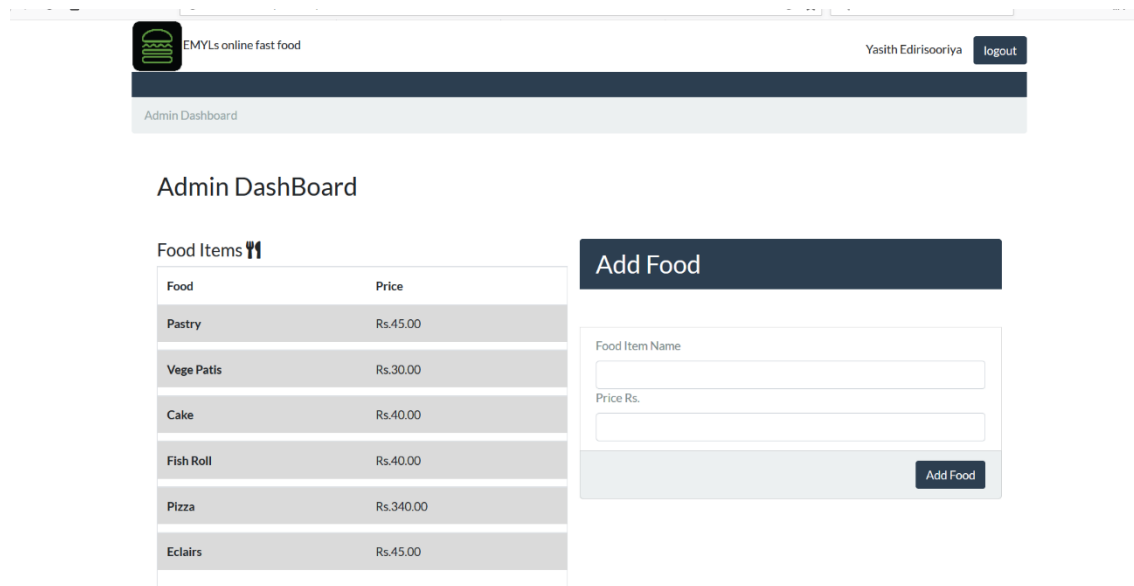


Figure1.3

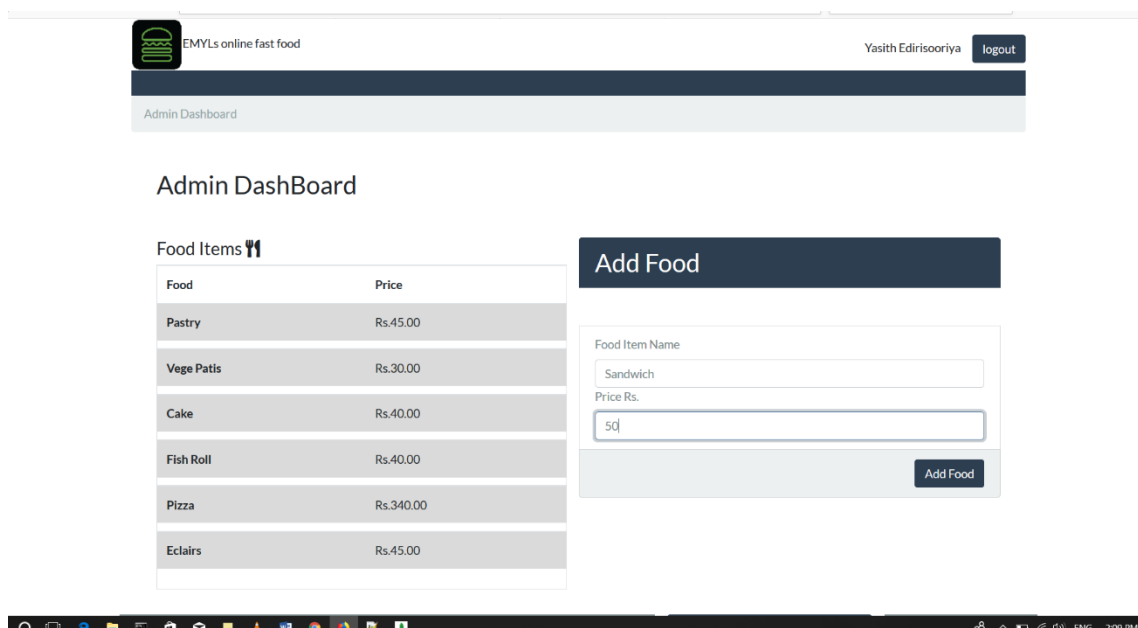


Figure1.4

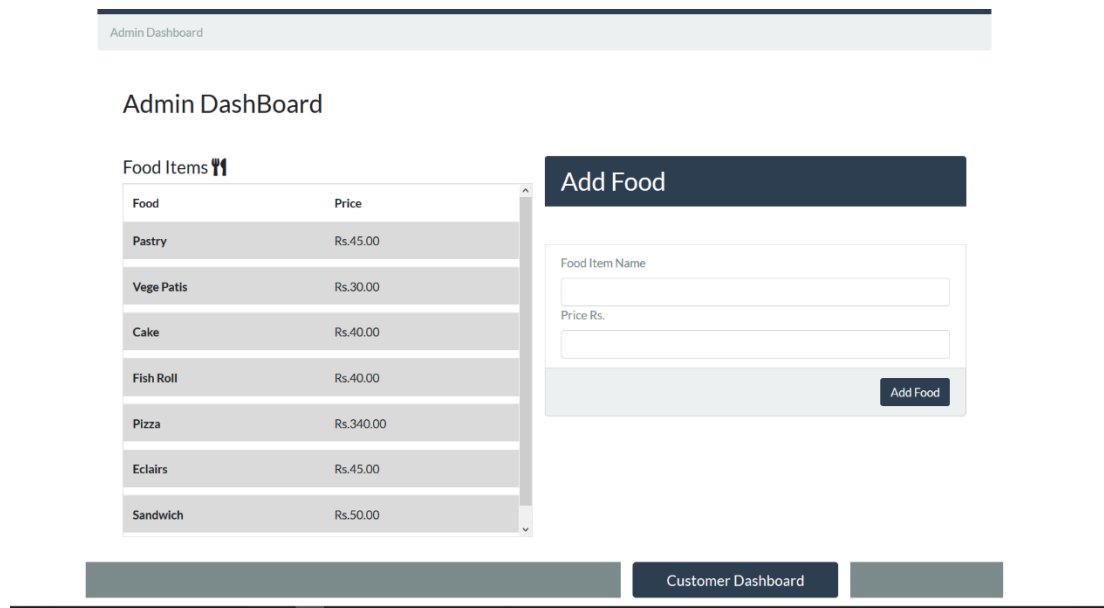


Figure1.5

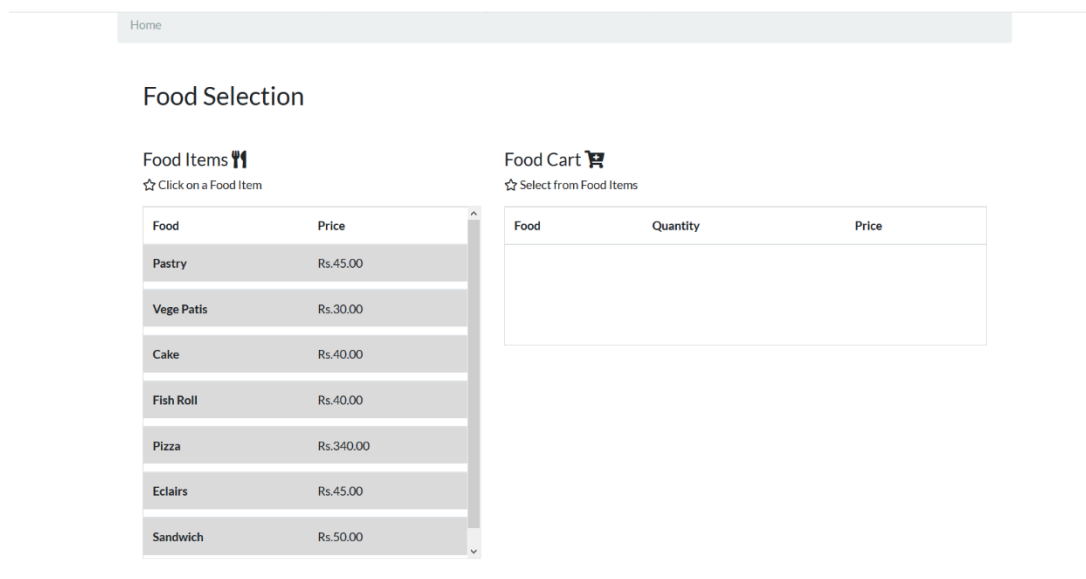


Figure1.6

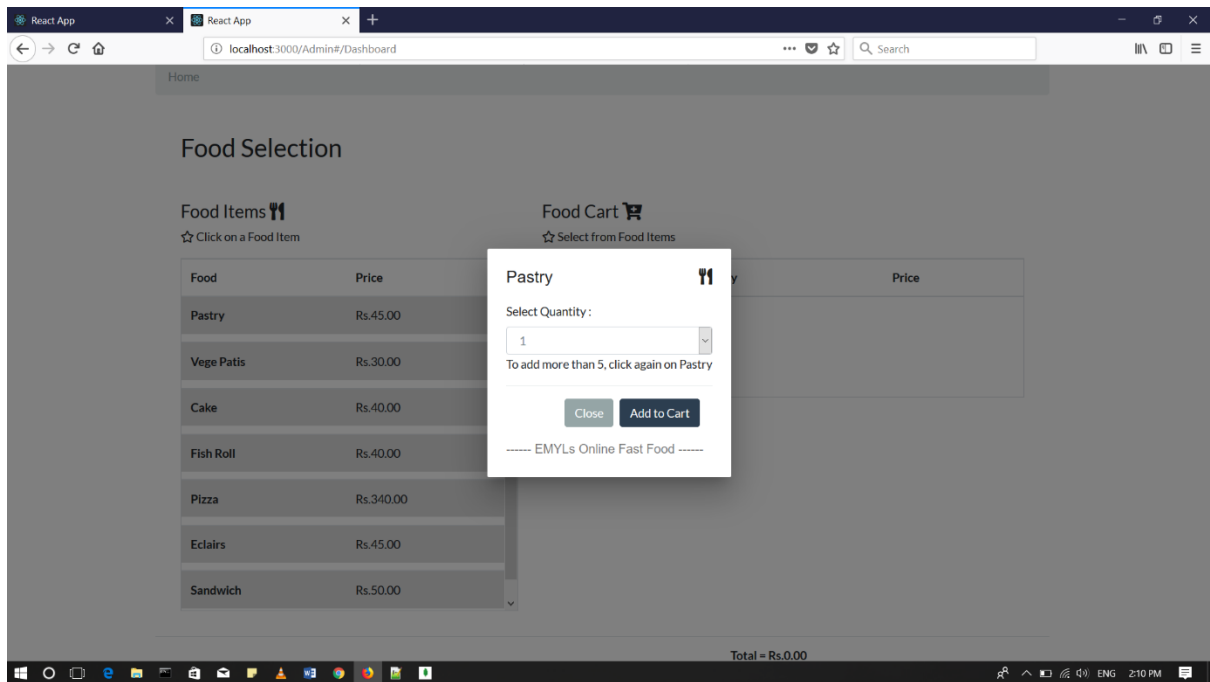


Figure1.7

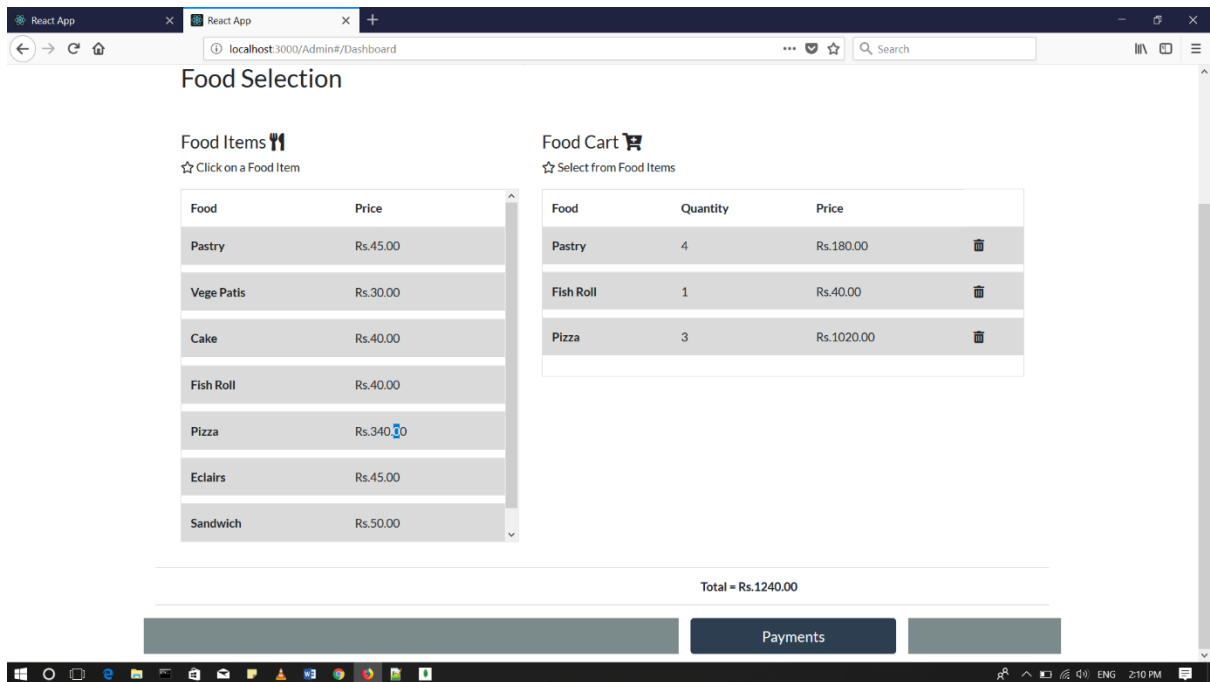


Figure1.8

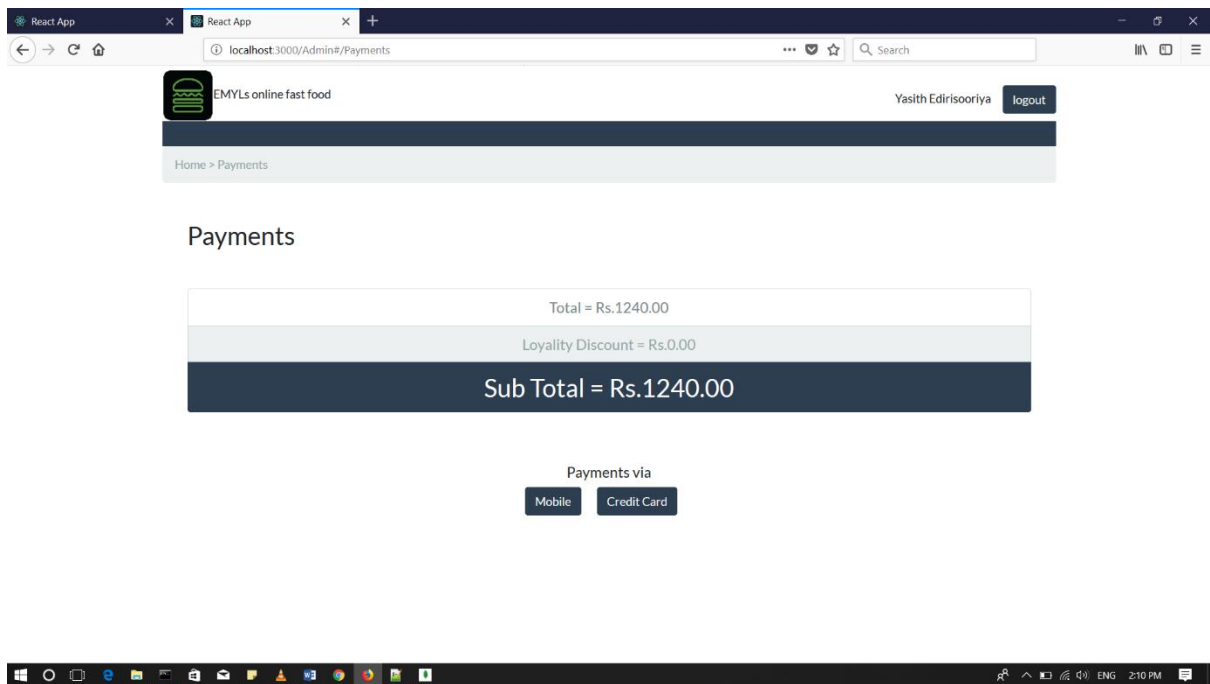


Figure1.9

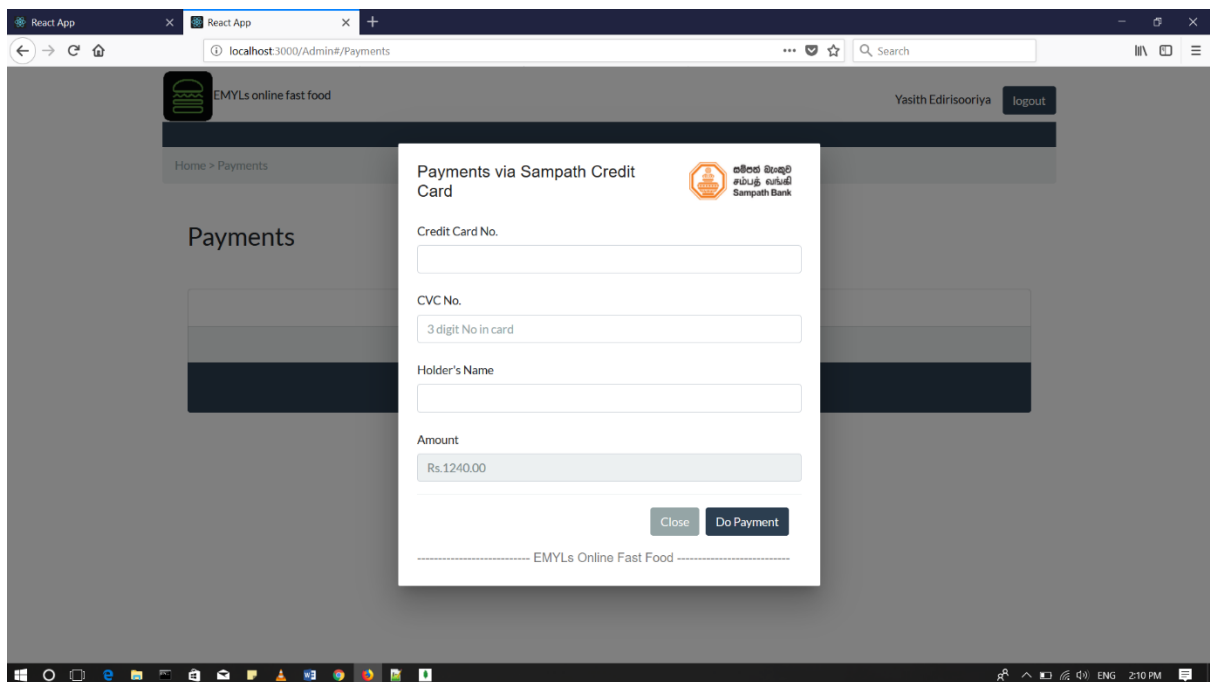


Figure1.10

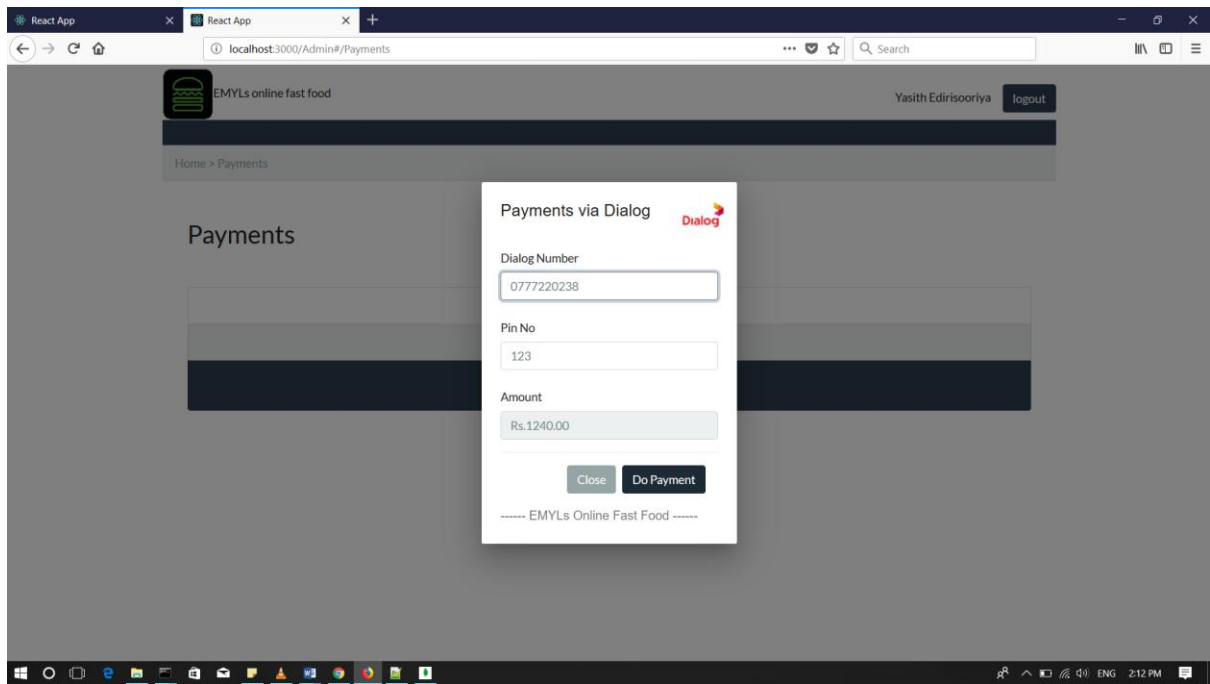


Figure1.11

RESTful API

App.js

```
const express = require('express');

const app = express();

const bodyParser = require('body-parser');
app.use(bodyParser.json());

const PORT = process.env.port || 3002 || 3001 || 3000;

// Server startup
app.listen(PORT, function (err) {
  if(err) {
    console.log(err);
    process.exit(-1);
  }
  console.log('Food company server is running on port ' + PORT);
});

const food = [{ name:"Pastry", price:45},
  { name:"Vege Patis",price:30},
  { name:"Cake",price:40},
  { name:"Fish Roll",price:40},
  { name:"Pizza",price:340}];

const customers = [{
  id:1,
  name:"Yasith Edirisooriya",
  type:"admin",
  mobileNo:"0711236234",
  username:"admin",
```

```
        password:"123",
        email:"edir@gmail.com",
        loyaltyPoints:0},

    {
        id:2,
        name:"Sheena Daniels",
        type:"user",mobileNo:"0777234543",
        username:"user",
        password:"123",
        email:"sheena@gmail.com",
        loyaltyPoints:5
    }
];
```

```
const payments = [];
```

```
const dialog = [{phoneNo:"0771234567",pin:1234}];
```

```
const sampath = [{creditCardNo:1234,cvc:123,holdersName:'ds'}];
```

```
// To resolve cross origin issues
```

```
app.use(function (req, res, next) {
    res.header('Access-Control-Allow-Origin', "*");
    res.header('Access-Control-Allow-Methods', 'GET,PUT,POST,DELETE');
    res.header('Access-Control-Allow-Headers', 'Content-Type');
    next();
});
```

```
// REST APIs
```

```
// Customers service
```

```
app.get('/customers', function(req, res) {
```

```
  res.status(200).send(customers);
```

```
});
```

```
// Customer get by id
```

```
app.get('/customers/:id', function (req,res) {
```

```
  const customer = customers.find(c => c.id === parseInt(req.params.id));
```

```
  if(!customer) res.status(404).send('cannot find the customer');
```

```
  else res.status(200).send(customer);
```

```
});
```

```
// POST a customer
```

```
app.post('/customers', function (req, res) {
```

```
  if(!req.body) {
```

```
    res.status(404).send('cannot find the customer');
```

```
    return;
```

```
  }
```

```
  const customer = {
```

```
    id:customers.length +1,
```

```
    name: req.body.name,
```

```
    type: req.body.type,
```

```
    mobileNo: req.body.mobileNo,
```

```
    username: req.body.username,
```

```

        password: req.body.password,
        email: req.body.email,
        loyaltyPoints: "0"
    };

    customers.push(customer);
    res.status(200).send(customer);
    fs.writeFile('./Files/Customers.txt', JSON(customer), function (err) {
        if (err) throw err;
        console.log('Saved!');
    });
});

// Check Login

app.post('/customers/login', function (req, res) {
    if(!req.body) {
        res.status(404).send('cannot find the customer');
        return;
    }

    const customer = customers.find(c => (c.username === req.body.username) &&
(c.password === req.body.password));

    if(!customer) res.status(404).send('cannot find the customer');

    else res.status(200).send(customer);
});

// Food Service

app.get('/food', function(req, res) {

```

```
    res.status(200).send(food);
  });

app.get('/food/:id', function (req,res) {
  const food = food.find(c => c.id === parseInt(req.params.id));

  if(!food) res.status(404).send('cannot find food');

  else res.status(200).send(food);
});

app.post('/food', function (req, res) {
  if(!req.body) {
    res.status(400).send('Bad request');
    return;
  }
  const foodi = {
    name: req.body.name,
    price: req.body.price
  };

  food.push(foodi);
  res.status(200).send(foodi);
});
```

// Dialog Payment Gateway

```
app.post('/dialog', function (req, res) {
  if(!req.body) {
    console.log('invalid');
```

```

        return;
    }

    const holder = dialog.find(c => (c.phoneNo === req.body.phoneNo) && (c.pin ===
parseInt(req.body.pin)));

    if(!holder) res.status(404).send('Not Found');

    else {
        const body = {
            amount:req.body.amount,
            userId:req.body.userId,
            date:new Date()
        };
        payments.push(body);
        res.status(200).send('payments are success');

    }
    console.log(holder);
});

//Sampath Payment Gateway
app.post('/sampath', function (req, res) {
    if(!req.body) {
        console.log('invalid');
        return;
    }

    const holder = sampath.find(c => (c.creditCardNo === parseInt(req.body.creditCardNo))
&&

(c.cvc === parseInt(req.body.cvc)) &&

```

```
        (c.holdersName === (req.body.holdersName))));

if(!holder) res.status(404).send('Not Found');

else {
    const body = {
        amount:req.body.amount,
        userId:req.body.userId,
        date:new Date()
    };
    payments.push(body);
    res.status(200).send('payments are success');

}
console.log(holder);
});
```

dbAPI.js

```
const express = require('express');

const app = express();

const bodyParser = require('body-parser');
app.use(bodyParser.json());

const PORT = process.env.port || 3002 || 3001 || 3000;

const mongoose1 = require('./DBSchema/CustomerSchema');
const mongoose2 = require('./DBSchema/FoodSchema');
const mongoose3 = require('./DBSchema/PaymentGateways/Dialog');
const mongoose4 = require('./DBSchema/PaymentGateways/Sampath');
const mongoose5 = require('./DBSchema/PaymentSchema');

const CustomerSchema = mongoose1.model('Customer');
const FoodSchema = mongoose2.model('Food');
const dialogSchema = mongoose3.model('dialog');
const sampathSchema = mongoose4.model('sampath');
const PaymentSchema = mongoose5.model('Payment');

const Mongo_DB_Base_URL = "mongodb://127.0.0.1:27017";

// Server
app.listen(PORT, function (err) {
  if(err) {
    console.log(err);
    process.exit(-1);
  }
  console.log('Food company server is running on port ' + PORT);
});
```



```
// To resolve cross origin issues
app.use(function (req, res, next) {
  res.header('Access-Control-Allow-Origin', "*");
  res.header('Access-Control-Allow-Methods', 'GET,PUT,POST,DELETE');
  res.header('Access-Control-Allow-Headers', 'Content-Type');
  next();
});

// Db connection
mongoose1.connect(Mongo_DB_Base_URL+"/FoodCompany", function (err) {
  if(err) {
    console.log(err);
    process.exit(-1);
  }
  console.log('FoodCompany Db is connected');
});

mongoose2.connect(Mongo_DB_Base_URL+"/Food", function (err) {
  if(err) {
    console.log(err);
    process.exit(-1);
  }
  console.log('Food Db is connected');
});

mongoose3.connect(Mongo_DB_Base_URL+"/dialog", function (err) {
  if(err) {
    console.log(err);
    process.exit(-1);
  }
  console.log('connected to Dialog Payment gateway ...');
});
```

```

mongoose4.connect(Mongo_DB_Base_URL+"/sampath", function (err) {
  if(err) {
    console.log(err);
    process.exit(-1);
  }
  console.log('Connected to Sampath Bank Payment gateway...');
});

mongoose5.connect(Mongo_DB_Base_URL+"/Payments", function (err) {
  if(err) {
    console.log(err);
    process.exit(-1);
  }
  console.log('Payments DB is connected');
});

// REST API Customers
app.get('/customers',function(req, res) {
  CustomerSchema.find().exec().then(function (customers) {
    res.status(200).send(customers);
  }).catch(function (err) {
    res.status(400).send('Bad Reques');
  });
});

app.get('/customers/:id',function(req, res) {
  CustomerSchema.find({_id:req.params.id}).exec().then(function (customers) {
    res.status(200).send(customers);
  }).catch(function (err) {
    res.status(400).send('Bad Reques');
  });
});

```

```
app.post('/customers', function (req, res) {

    const data = new CustomerSchema(req.body);
    console.log(req.body);
    data.save().then(function (data) {
        res.status(200).send(data);
    }).catch(function () {
        res.status(400).send({message:'Bad request'});
    });
});

app.post('/customers/login', function (req, res) {
    if(!req.body) {
        res.status(404).send('cannot find the customer');
        return;
    }

    CustomerSchema.find({ username:req.body.username,password:req.body.password}).exec().then(function (customers) {
        res.status(200).send(customers[0]);
    }).catch(function (err) {
        res.status(404).send('cannot find the customer');
    });
});

app.put('/customers/:id', function (req, res) {
    CustomerSchema.update({_id:req.params.id}, req.body).then(function (customer) {
        res.status(200).send({customer: customer, message: 'updated'});
    }).catch(function () {
        res.status(400).send({message:'Bad request'});
    });
});

app.delete('/customers/:id', function (req, res) {
    CustomerSchema.remove({_id: req.params.id}).then(function () {
        res.status(200).send({message:'deleted'});
    }).catch(function () {
```

```

        res.status(400).send({ message: 'Bad request' });
    });
});

// REST API food
app.get('/food', function (req, res) {
    FoodSchema.find().exec().then(function (food) {
        res.status(200).send(food);
    }).catch(function (err) {
        res.status(400).send('Bad Reques');
    });
});

app.get('/food/:name', function (req, res) {
    FoodSchema.find({ name: req.body.name }).exec().then(function (food) {
        res.status(200).send(food);
    }).catch(function (err) {
        res.status(400).send('Bad Reques');
    });
});

app.post('/food', function (req, res) {
    const food = new FoodSchema(req.body);
    food.save().then(function (food) {
        res.status(200).send(food);
    }).catch(function (err) {
        res.status(400).send('Bad Reques');
    });
});

app.put('/food/:id', function (req, res) {
    CustomerSchema.update({ _id: req.params.id }, req.body).then(function (customer) {
        res.status(200).send({ customer: customer, message: 'updated' });
    });
});

```

```
    }).catch(function () {
        res.status(400).send({ message: 'Bad request' });
    });
});

app.put('/food/:id', function (req, res) {
    FoodSchema.update({ _id: req.params.id }, req.body).then(function (food) {
        res.status(200).send({ customer: food, message: 'updated' });
    }).catch(function () {
        res.status(400).send({ message: 'Bad request' });
    });
});
```

```
app.delete('/food/:id', function (req, res) {
    FoodSchema.remove({ _id: req.params.id }).then(function () {
        res.status(200).send({ message: 'deleted' });
    }).catch(function () {
        res.status(400).send({ message: 'Bad request' });
    });
});
```

// Dialog Payment Gateway APIs

```
app.post('/dialog', function (req, res) {
    if (!req.body) {
        console.log('invalid');
        return;
    }
    const dialog = new dialogSchema({
        amount: req.body.amount,
        phoneNo: req.body.phoneNo,
        date: new Date()
    });
});
```

```

const payment = new PaymentSchema({
  amount:req.body.amount,
  type:req.body.type,
  date:new Date(),
  userid:req.body.userId
});

if(req.body.pin === '1234') {
  dialog.save().then(function () {
    payment.save().then(function () {
      res.status(200).send({ message:'payment success via dialog'});
    }).catch(function () {
      res.status(400).send({ message:'Bad request'});
    });
    // res.status(200).send();
  }).catch(function () {
    res.status(400).send({ message:'Bad request'});
  });
}

else res.status(400).send({ message:'Bad request'});
});

// Sampath Payment Gateway APIs
app.post('/sampath', function (req, res) {
  if(!req.body) {
    console.log('invalid');
    return;
  }
  const sampath = new sampathSchema({
    amount:req.body.amount,

```

```
    holdersName:req.body.holdersName,  
    creditCardNo:req.body.creditCardNo,  
    date:new Date(),  
  });
```

```
const payment = new PaymentSchema({  
  amount:req.body.amount,  
  type:req.body.type,  
  date:new Date(),  
  userid:req.body.userId  
});
```

```
if(req.body.cvc === '123') {  
  sampath.save().then(function () {  
    payment.save().then(function () {  
      res.status(200).send({ data:payment,message:'payment success via sampath'});  
    }).catch(function () {  
      res.status(400).send({ message:'Bad request'});  
    });  
    //res.status(200).send();  
  }).catch(function () {  
    res.status(400).send({ message:'Bad request'});  
  });  
}
```

```
else res.status(400).send({ message:'Bad request'});  
});
```

// Payments APIs

```
app.get('/payments',function (req,res) {  
  PaymentSchema.find().exec().then(function (data) {
```

```
        res.status(200).send(data);
    }).catch(function (err) {
        console.log(err);
    })
});
```

DBSchema

PayemntSchema.js

```
const mongoose = require('mongoose');
const Schema = mongoose.Schema;
```

```
const PaymentSchema = new Schema({
  amount: {
    type:String,
    require:true
  },
  type:{
    type:String,
    require:true
  },
  date:{
    type:String,
    require:true
  },
  userid:{
    type:String,
    require:true
  }
});
```

```
mongoose.model('Payment', PaymentSchema);
```



```
module.exports = mongoose;
```

Web client

App.jsx

```
import React, { Component } from 'react';
import { HashRouter, Route } from 'react-router-dom';
import Login from './app/containers/Login';
import Dashboard from './app/containers/Dashboard';
import Payments from './app/containers/PaymentGateway';
import SignUp from './app/containers/SignUp';
import AdminDashboard from './app/containers/AdminDashboard';

class App extends Component {
```

```
render() {  
  const isLoggedIn = localStorage.getItem('logged');  
  const type = localStorage.getItem('type');  
  return (  
    <HashRouter>  
      {  
        isLoggedIn === 'ok'  
        ?  
          type === 'user'  
          ?  
            <div>  
              <Route exact path="/" component={Dashboard} />  
              <Route path="/Dashboard" component={Dashboard} />  
              <Route path="/Payments" component={Payments}/>  
            </div>  
          :  
            <div>  
              <Route exact path="/" component={AdminDashboard} />  
              <Route path="/Dashboard" component={Dashboard} />  
              <Route path="/Payments" component={Payments}/>  
              <Route path="/Admin" component={AdminDashboard} />  
            </div>  
          :  
            <div>  
              <Route exact path="/" component={Login}/>  
              <Route path="/Login" component={Login}/>  
              <Route path="/SignUp" component={SignUp} />  
            </div>  
      }  
    </HashRouter>  
  );  
}
```

```
    }  
  }  
  export default App;
```

Components

Header.js

```
import React, { Component } from 'react';  
import logo from '../images/logo.png'  
import '../scss/Header.css';  
class Header extends Component {  
  signOut = () => {  
    window.localStorage.clear();  
    window.location.href = '/login';  
  }  
  render() {  
    return(  
      <div>  
        <span><img src={logo} /></span>  
        <span>EMYLS online fast food</span>  
  
        <span className='ds-logout'>  
          <span>{localStorage.getItem('user')}</span>  
          <button type="button" class="btn btn-primary" onClick={this.signOut}>logout</button>  
        </span>  
  
        <span className="navbar navbar-expand-lg navbar-dark bg-primary" />  
      </div>  
    );  
  }  
}  
  
export default Header;
```

ButtonBar.js

```
import React, { Component } from 'react';
import { Link } from 'react-router-dom';
import './scss/ButtonBar.css';
import Payments from '../containers/PaymentGateway';

class ButtonBar extends Component {

  render() {
    let redirect='/Payments';
    this.props.work === 'Payments' ? redirect='/Payments' : redirect='/Dashboard';
    return(
      <div className="row">
        <div className="col-md-7 ds-buttonbar">

          </div>

          <div className="col-md-3">
            <Link to={redirect}>
              <button type="button" className="btn btn-primary btn-lg btn-block">{this.props.work}</button>
            </Link>
          </div>
          <div className="col-md-2 ds-buttonbar">

          </div>
        </div>
      );
    }
  }
```

```
export default ButtonBar;
```

Breadcrum.js

```
import React, {Component} from 'react';  
class Breadcrumb extends Component {  
  render() {  
    return(  
      <ol className="breadcrumb">  
        <li className="breadcrumb-item active">{this.props.path}</li>  
      </ol>  
    );  
  }  
}
```

```
export default Breadcrumb;
```

AllFoodItems.jsx

```
import React, {Component} from 'react';  
import '../scss/FoodItems.css'  
import FoodRaw from './FoodRaw';  
import * as API from '../Constants/APIs';  
import axios from 'axios';
```

```
class AllFoodItems extends Component {
```

```
  constructor(props) {  
    super(props);  
    this.state = {  
      allFood:[],  
      isResponseFollFill:false  
    }  
  }
```

```
}
```

```
componentDidMount() {
```

```
  let url = API.Get_All_Food;
```

```
  let food = this;
```

```
  axios.get(url).then(function (response) {
```

```
    food.setState({
```

```
      allFood:response.data,
```

```
      isResponseFollFill:true
```

```
    });
```

```
    console.log(response.data);
```

```
  }).catch(function (err) {
```

```
    console.log(err);
```

```
  });
```

```
}
```

```
render() {
```

```
  return(
```

```
    <div>
```

```
      <h4>Food Items <i className="fas fa-utensils"/></h4>
```

```
      {this.props.disable === false ?
```

```
        <p> <i className="far fa-star"/> Click on a Food Item</p>
```

```
      :
```

```
      null}
```

```
    <div className="ds-content-wrapper">
```

```
      <table className="table table-hover ">
```

```
        <thead >
```

```
          <tr>
```

```
            <th scope="col">Food</th>
```

```
            <th scope="col">Price</th>
```

```
          </tr>
```

```

        </thead>

        {
            this.state.allFood.map((food, index) => {
                return(

                    <FoodRaw key={index} index={index} food={food}
                        addFoodToSelectedFood={this.props.addFoodToSelectedFood}
                        disable={this.props.disable}/>

                )
            })
        }
    </table>

</div>
</div>

);
}

}

export default AllFoodItems;

```

TotalRows.jsx

```

import React, {Component} from 'react';

class TotalRow extends Component {
    constructor(props) {
        super(props);

        this.state = {
            selectedFood:[{ name:'Pastry',quantity:2,price:90.00}],
            total:0.00
        }
    }

    calculateTotal() {

```

```

    const selectedFood = this.props.selectedFood;

    let total = 0;
    for(let i=0; i<selectedFood.length; i++) {
        total = total + selectedFood[i].price;
    }
    localStorage.setItem('total',total);
    return total;
}

render() {
    return(
        <div>
            <table className="table table-hover">
                <thead>
                    <tr>
                        <th scope="col"> </th>
                        <th scope="col"> </th>
                        <th scope="col"> </th>
                        <th scope="col">Total = Rs.{this.calculateTotal()}.00</th>
                    </tr>
                </thead>
            </table>
        </div>
    );
}
}

```

```
export default TotalRow;
```

SelectItem.js

```

import React, {Component} from 'react';
import '../scss/FoodItems.css'

class SelectedFood extends Component {
    render() {

```



```

return(
  <div>
    <h4>Food Cart <i className="fas fa-cart-plus" /></h4>
    <p> <i className="far fa-star"/> Select from Food Items</p>
    <div className="ds-content-wrapper">
      <table className="table table-hover">
        <thead>
          <tr>
            <th scope="col">Food</th>
            <th scope="col">Quantity</th>
            <th scope="col">Price</th>
          </tr>
        </thead>

        {
          this.props.selectedFood.map((food, index) => {
            return(
              <tbody key={index}>
                <tr className="table-active">
                  <th scope="row">{ food.name }</th>
                  <td>{ food.quantity }</td>
                  <td>Rs.{ food.price }.00</td>
                  <td><i className="fas fa-trash-alt" style={{ cursor:'grab' }}
onClick={ ()=>this.props.deleteFood(index) }/></td>
                </tr>
                <tr className="row" style={{ border:'0px solid white',padding:'5px' }}>

                </tr>

              </tbody>
            )
          })
        }
      </div>
    </div>
  )
)
}

```

```

        </table>
      </div>
    </div>
  );
}
}
export default SelectedFood;

```

FoodRow.js

```

import React, { Component } from 'react';
import Dialog from '@material-ui/core/Dialog';
import DialogContent from '@material-ui/core/DialogContent';
import DialogContentText from '@material-ui/core/DialogContentText';
import DialogTitle from '@material-ui/core/DialogTitle';
import Slide from '@material-ui/core/Slide';
import '../scss/FoodItems.css'

function Transition(props) {
  return <Slide direction="up" {...props} />;
}

class FoodRow extends Component {
  constructor(props) {
    super(props);
    this.state = {
      allFood:[{ name:'Pastry',price:45.00}, { name:'Patis',price:30.00}],
      open: false
    }
  }

  handleClickOpen = () => {
    this.setState({ open: true});
  };

```

```

handleClose = () => {
  this.setState({ open: false });
};
addFoodToSelectedFood = () => {
  const food = this.props.food;
  const id = "exampleSelect" + this.props.index;
  const quantity = parseInt(document.getElementById(id).value);

  this.props.addFoodToSelectedFood({ name: food.name, quantity: quantity, price: (food.price * quantity) });
  this.handleClose();
}
render() {
  return(
    <tbody>
      <tr className="ds-food-column table-active " onClick={this.handleClickOpen}>
        <th scope="row">{this.props.food.name}</th>
        <td>Rs.{this.props.food.price}.00</td>
      </tr>
      <tr className="row" style={{ border: '0px solid white', padding: '5px' }}>

    </tr>

    {
      this.props.disable === false ?
        <Dialog
          open={this.state.open}
          TransitionComponent={Transition}
          keepMounted
          onClose={this.handleClose}
          aria-labelledby="alert-dialog-slide-title"
          aria-describedby="alert-dialog-slide-description"
        >

```

```
<DialogTitle id="alert-dialog-slide-title">
```

```
  <div className="row">
```

```
    <div className="col">
```

```
      {this.props.food.name}
```

```
    </div>
```

```
    <div className="col ds-dialog">
```

```
      <i className="fas fa-utensils"/>
```

```
    </div>
```

```
  </div>
```

```
</DialogTitle>
```

```
<DialogContent>
```

```
  <div className="form-group">
```

```
    <label>Select Quantity : </label>
```

```
    <select className="form-control" id={ "exampleSelect"+this.props.index }>
```

```
      <option value="1">1</option>
```

```
      <option value="2">2</option>
```

```
      <option value="3">3</option>
```

```
      <option value="4">4</option>
```

```
      <option value="5">5</option>
```

```
    </select>
```

```
    To add more than 5, click again on {this.props.food.name}
```

```
  </div>
```

```
  <div className="modal-footer">
```

```
    <button type="button" className="btn btn-secondary" data-dismiss="modal"
onClick={this.handleClose}>Close</button>
```

```
    <button type="button" className="btn btn-primary"
onClick={this.addFoodToSelectedFood}>Add to Cart</button>
```

```
  </div>
```

```
<DialogContentText id="alert-dialog-slide-description">
```

```
  ----- EMYLs Online Fast Food -----
```

```
</DialogContentText>
```

```

        </DialogContent>
      </Dialog>
    :
    null
  }
  </tbody>
);
}
}
export default FoodRaw;

```

Containers

```

Sign import React, {Component} from 'react';
import axios from 'axios';
import * as API from '../Constants/APIs';
import '../scss/Login.css';

```

```

class SignUp extends Component {

```

```

  signup() {
    const pss1 = document.getElementById('password1').value;
    const pss2 = document.getElementById('password2').value;

    if( pss1.length>0 && pss1 === pss2){
      const body = {
        name:document.getElementById('name').value,
        mobileNo:document.getElementById('mobile').value,
        email:document.getElementById('email').value,
        username:document.getElementById('username').value,
        password:document.getElementById('password1').value,
        loyaltyPoints:0,

```

```

        type:document.getElementById('type').value
    };
    console.log(pss1.length,pss2,body);
    axios.post(API.Get_All_Customers,body).then(function (response) {
        if(response.status==200) {
            window.alert('Welcome to EMYLs Food world')
            localStorage.setItem('logged','ok');
            localStorage.setItem('user', response.data.name);
            localStorage.setItem('type', response.data.type);
            localStorage.setItem('id', response.data.id || response.data._id);
            localStorage.setItem('points', response.data.loyaltyPoints);
            console.log()

            response.data.type === 'admin' ? window.location.href = '/Admin' :
window.location.href = '/Dashboard';

        }
        else window.alert('cannot create :()
    }).catch(function (err) {
        window.alert('cannot create :()
    });
}
else window.alert('Password mismatch');
}
render() {
    return(
        <div className="list-group ds-signUp">
            <h3 className="list-group-item list-group-item-action active">SignUp</h3>

            <div className="form-group list-group-item list-group-item-action">
                <label className="col-form-label">Name</label>
                <input type="text" className="form-control" id="name" />

```

```
<label className="col-form-label">Mobile No.</label>

<input type="text" className="form-control" id="mobile" />

</div>
```

```
<div className="form-group list-group-item list-group-item-action">

  <label>Email address</label>

  <input type="email" className="form-control" id="email" aria-
describedby="emailHelp" placeholder="Enter email" />

  <small id="emailHelp" className="form-text text-muted">We'll never share your
email with anyone else.</small>

</div>
```

```
<div className="form-group list-group-item list-group-item-action">

  <label className="col-form-label">Username</label>

  <input type="text" className="form-control" id="username" />

</div>
```

```
<div className="form-group list-group-item list-group-item-action">

  <label>Password</label>

  <input type="password" className="form-control" id="password1"/>

</div>
```

```
<div className="form-group list-group-item list-group-item-action">

  <label>Re-Enter Password</label>

  <input type="password" className="form-control" id="password2" />

</div>
```

```
<div className="form-group list-group-item list-group-item-action">

  <label>User Type</label>

  <select className="form-control" id="type" >

    <option value="admin">Admin</option>
```

```

        <option value="user">User</option>
    </select>

    <small id="emailHelp" style={{ color:'red' }}>This is provide to create an admin to
log as admin</small>

</div>

<div className="list-group-item list-group-item-action disabled ds-button">
    <button type="button" className="btn btn-primary btn-lg ds-button"
onClick={this.signup}>SignUp</button>
</div>
</div>
);
}
}

```

export default SignUp;up.js

PaymentGateway.js

```

import React,{Component} from 'react';
import Header from '../components/Header';
import AmountDisplay from '../components/Payments/AmountDisplay';
import BreadCrumb from '../components/BreadCrumb';
import PaymentSelectionBar from '../components/Payments/PaymentSelectionBar';
import '../scss/Payments.css';
class PaymentGateway extends Component {
    render() {
        console.log(this.props)
        return (
            <div className="ds-container">
                <Header/>
                <BreadCrumb path={"Home > Payments"}/>
                <div className="ds-payments">

```



```

        <h3>Payments</h3>

        <AmountDisplay/>

    </div>

    <PaymentSelectionBar/>

</div>

);

}

}

```

Login.js

```

import React, { Component } from 'react';
import { Link } from 'react-router-dom';
import * as API from '../Constants/APIs';
import axios from 'axios';
import '../scss/Login.css';

class Login extends Component {
  login() {
    let login = this;

    const username = document.getElementById('username').value;
    const password = document.getElementById('password').value;
    const body = { username:username,password:password };

    axios.post(API.LOGIN,body).then(function (response) {
      if(response.status === 200) {
        console.log(response);
        localStorage.setItem('logged','ok');
        localStorage.setItem('user', response.data.name);
        localStorage.setItem('type', response.data.type);
        localStorage.setItem('id', response.data.id);
        localStorage.setItem('points', response.data.loyaltyPoints);
        response.data.type === 'admin' ? window.location.href = '/Admin' : window.location.href =
'/Dashboard';
      }
    })
  }
}

```

```

    }).catch( function (err) {
        console.log(err);
    });
}

render() {
    return(
        <div className="list-group ds-login">
            <h3 className="list-group-item list-group-item-action active">
                Login
            </h3>

            <div className="form-group list-group-item list-group-item-action">
                <label >Username</label>
                <input type="text" className="form-control" id="username" aria-
describedby="emailHelp"/>
            </div>

            <div className="form-group list-group-item list-group-item-action">
                <label>Password</label>
                <input type="password" className="form-control" id="password"
placeholder="Password" />
            </div>

            <div className="list-group-item list-group-item-action disabled ds-button">
                <Link to={"/SignUp"}>
                    <button type="button" className="btn btn-primary btn-lg ds-button">SignUp</button>
                </Link>
                <button type="button" className="btn btn-primary btn-lg"
onClick={this.login}>Login</button>
            </div>
        </div>
    );
}
}

```

```
export default Login;
```

Dashboard.js

```
import React, {Component} from 'react';
import Header from '../components/Header';
import AllFoodItems from '../components/FoodSelection/AllFoodItems';
import SelectedFood from '../components/FoodSelection/SelectedFood';
import ButtonBar from '../components/ButtonBar';
import Breadcrumb from '../components/BreadCrumb';
import TotalRow from '../components/FoodSelection/TotalRow';
import '../scss/FoodItems.css'

class Dashboard extends Component {

  constructor(props) {
    super(props);

    this.state = {
      selectedFood:[]
    }

    this.addFoodToSelectedFood = this.addFoodToSelectedFood.bind(this);
    this.deleteFoodFromSelectedFood = this.deleteFoodFromSelectedFood.bind(this);
  }

  addFoodToSelectedFood(food) {
    let selectedFood = this.state.selectedFood;
    let previous = null;
```

```
for(let i=0; i<selectedFood.length; i++) {  
  if(food.name === selectedFood[i].name) {  
    previous = selectedFood[i];  
    previous.quantity = previous.quantity + food.quantity;  
    previous.price = previous.price + food.price;  
  }  
}
```

```
if(previous === null) {  
  selectedFood.push(food);  
}  
this.setState(selectedFood);  
}
```

```
deleteFoodFromSelectedFood (index) {  
  const selectedFood = this.state.selectedFood;  
  
  selectedFood.splice(index,1);  
  
  this.setState(selectedFood);  
}
```

```
render() {  
  return(  
    <div className='ds-container'>  
  
      <Header/>  
      <Breadcrumb path={'Home'}/>  
  
      <div className="ds-food-selection">
```

```

    <h3>Food Selection</h3>

    <div className="row">

        <div className="col-md-5">

            <AllFoodItems addFoodToSelectedFood={this.addFoodToSelectedFood}
disable={false}/>

        </div>

        <div className="col">

            <SelectedFood selectedFood={this.state.selectedFood}
deleteFood={this.deleteFoodFromSelectedFood}/>

        </div>

    </div>

</div>

<div className="row ds-total">

    <div className="col">

        <TotalRow selectedFood={this.state.selectedFood} />

    </div>

</div>

<ButtonBar work={'Payments'}/>

</div>

);
}
}

export default Dashboard;

```

Admins Dashboard.js

```
import React, {Component} from 'react';
import * as API from '../Constants/APIs';
import axios from 'axios';

import Header from '../components/Header';
import AllFoodItems from '../components/FoodSelection/AllFoodItems';
import SelectedFood from '../components/FoodSelection/SelectedFood';
import ButtonBar from '../components/ButtonBar';
import BreadCrumb from '../components/BreadCrumb';
import TotalRow from '../components/FoodSelection/TotalRow';
import '../scss/FoodItems.css'

class Dashboard extends Component {
  constructor(props) {
    super(props);

    this.state = {
      selectedFood:[]
    }

    this.addFoodToSelectedFood = this.addFoodToSelectedFood.bind(this);
    this.deleteFoodFromSelectedFood = this.deleteFoodFromSelectedFood.bind(this);
  }
  addFoodToSelectedFood(food) {
    let selectedFood = this.state.selectedFood;
    let previous = null;

    for(let i=0; i<selectedFood.length; i++) {
      if(food.name === selectedFood[i].name) {
```

```

        previous = selectedFood[i];
        previous.quantity = previous.quantity + food.quantity;
        previous.price = previous.price + food.price;
    }
}
if(previous === null) {
    selectedFood.push(food);
}
this.setState(selectedFood);
}

deleteFoodFromSelectedFood (index) {
    const selectedFood = this.state.selectedFood;
    selectedFood.splice(index,1);
    this.setState(selectedFood);
}

addFood() {
    let addFood = this;
    const body = {
        name:document.getElementById('name').value,
        price:document.getElementById('price').value
    }
    axios.post(API.Get_All_Food, body).then(function (response) {
        //window.alert(response.status);
        window.location.reload();
    }).catch(function (err) {
        console.log(err);
    });
}

render() {
    return(
        <div className='ds-container'>
            <Header/>

```

```

    <Breadcrumb path={'Admin Dashboard'}/>
    <div className="ds-food-selection">
      <h3>Admin DashBoard</h3>
      <div className="row">
        <div className="col-md-6">
          <AllFoodItems addFoodToSelectedFood={this.addFoodToSelectedFood}
            disable={true}/>
        </div>
        <div className="list-group col">
          <h3 className="list-group-item list-group-item-action active">
            Add Food
          </h3>
          <div className="form-group list-group-item list-group-item-action">
            <label>Food Item Name</label>
            <input type="text" className="form-control" id="name" aria-
describedby="emailHelp"/>
            <label>Price Rs.</label>
            <input type="text" className="form-control" id="price" aria-
describedby="emailHelp"/>
          </div>
          <div className="list-group-item list-group-item-action disabled ds-button">
            <button type="button" className="btn btn-primary"
onClick={this.addFood}>Add Food</button>
          </div>
        </div>
      </div>
    </div>
    <div>
      <ButtonBar work={'Customer Dashboard'}/>
    </div>
  );
}
}

export default Dashboard;

```


Appendix

Figure	Description
Figure 1.1	Login
Figure 1.2	Admins dashboard
Figure 1.3	Adding items
Figure 1.4	Updated Table
Figure 1.5	Customer dashboard
Figure 1.6	Select required food
Figure 1.7	Add quantity
Figure 1.8	Updated table
Figure 1.9	Total Price
Figure 1.10	Sampath bank gateway
Figure 1.11	Dialog gateway form
Figure 1.12	Hardware Architecture
Figure 1.13	High level architecture