SLIIT

Discover Your Future

SE3040-Application Frameworks

Software Engineering - Year - 3 Semester 1, 2020

Topic- Tourism

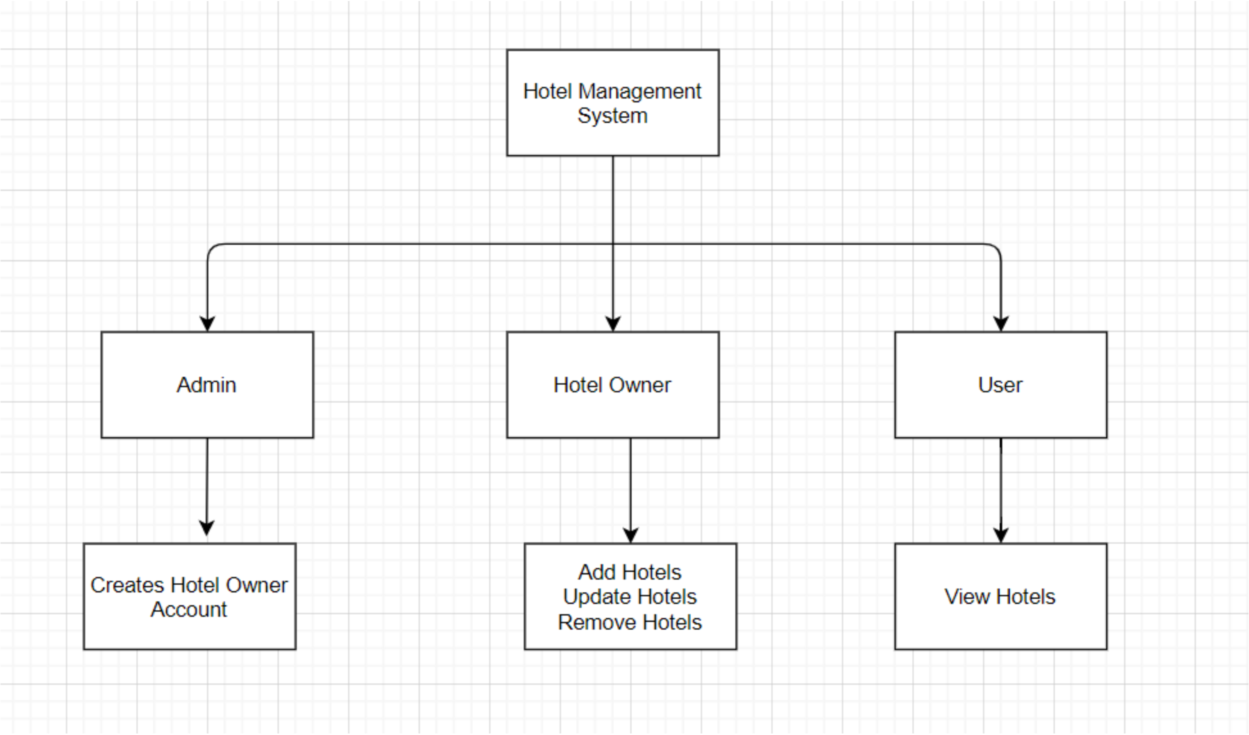
IT18097634

M.M.C Ranasinghe

Online Search Hotels System for a Tourism

Content

Scenario-Tourism



Main Functionalities

1)Insert: Hotel Owner can Add hotels

2)Update: Hotel Owner can Update hotel Details

3)Remove: Hotel Owner can Delate hotels

Develop a Web application Online Search Hotels System for a Tourism. System functionality will be as follows.

1. There are three roles as admin, Hotel owner, and User.

2. Users who logged in to the system can search hotels .

3. Admin creates a login for the Hotel owner

4. Admin can create categories for the Hotels (luxury hotels etc..).

5. The Hotel owner can add,update,delete hotels with details to the relevant category

6. The system has proper authentication.

use of the technologies.

1. ReactJS

2.NodeJS

3. ExpressJS

4. RESTful web services

5. MongoDB

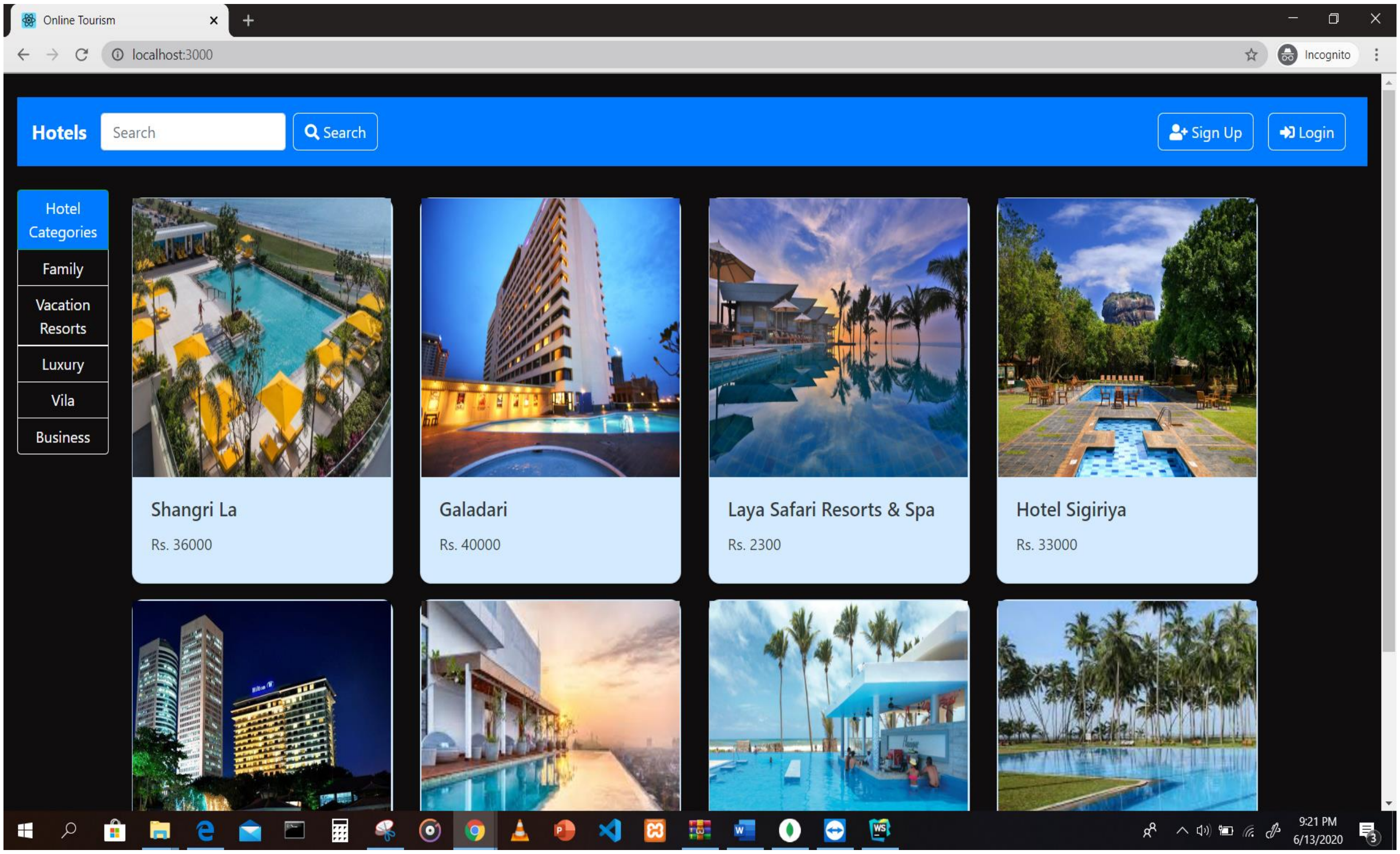
6.JEST

System backend should be an API running JSON based web services. The frontend application that are developing, should communicate with the back end only using these web services.

Page Screenshots

Home View (Hotels)

User go to the home page user can see hotels with relevant details.



Shangri La

Rs. 36000

Galadari

Rs. 40000

Laya Safari Resorts & Spa

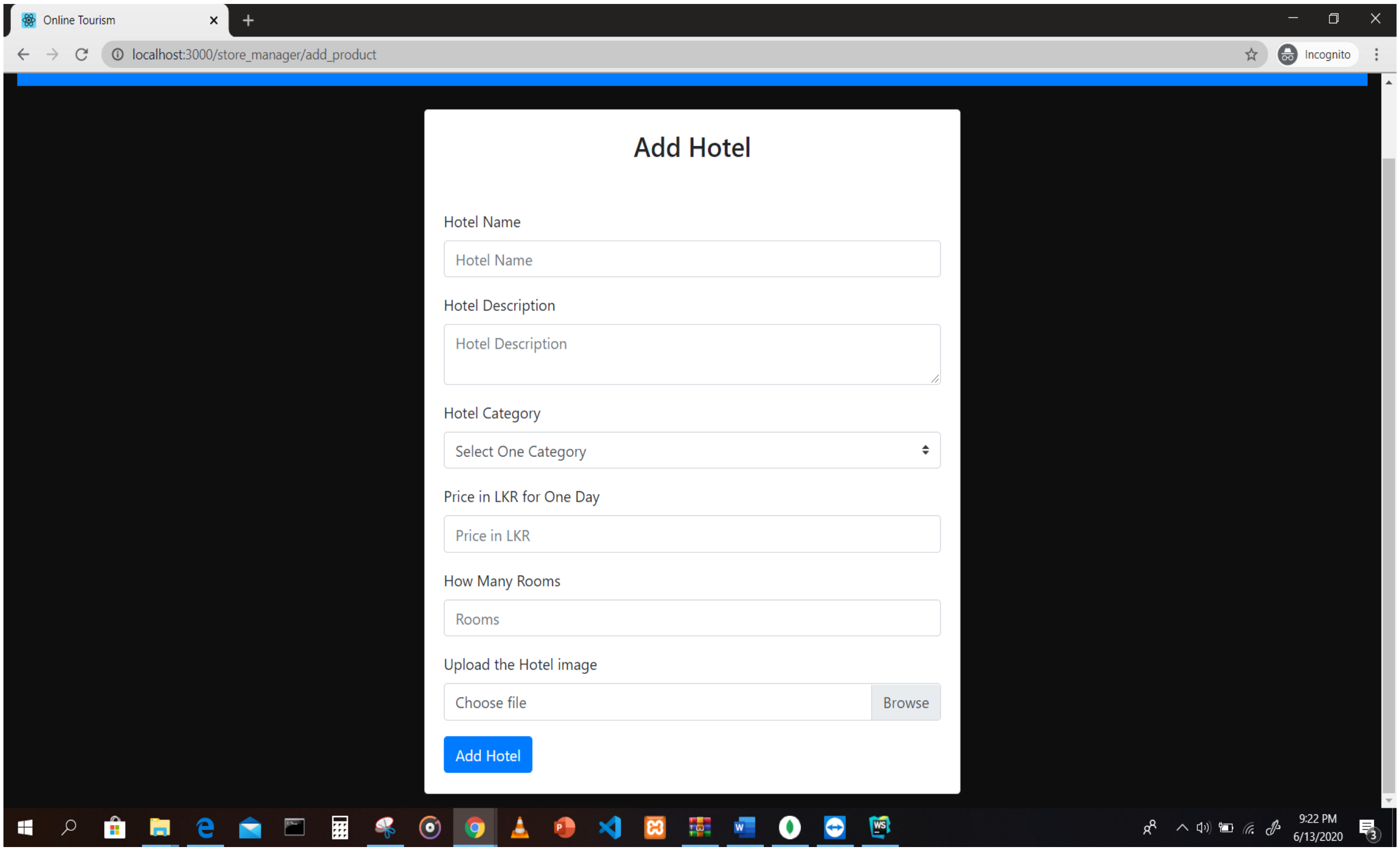
Rs. 2300

Hotel Sigiriya

Rs. 33000

Add Hotel

Hotel Owner can add hotel with relevent details.



Add Hotel

Hotel Name

Hotel Name

Hotel Description

Hotel Description

Hotel Category

Select One Category

Price in LKR for One Day

Price in LKR

How Many Rooms

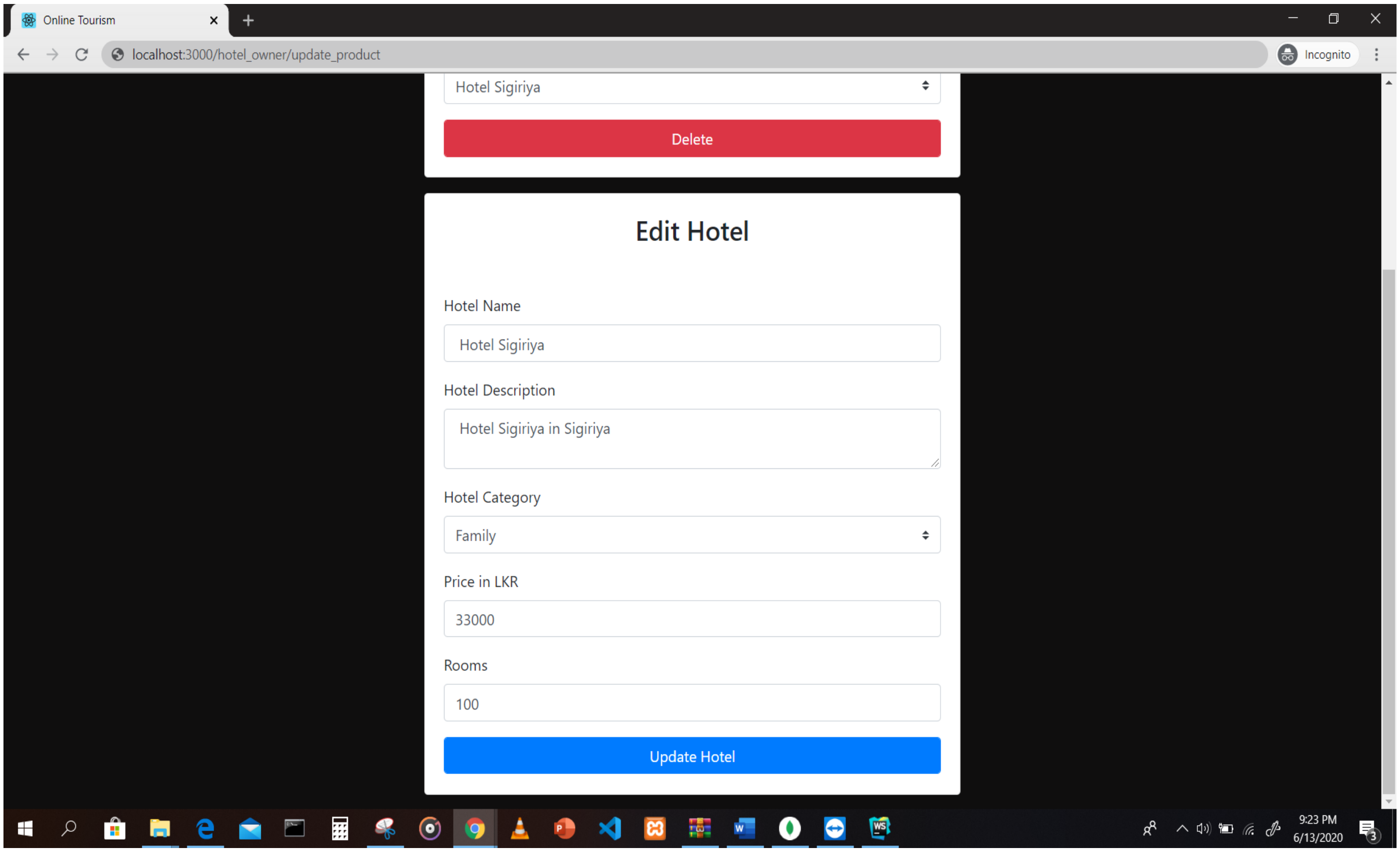
Rooms

Upload the Hotel image

Choose file

Update Hotel Details

Hotel Owner can Update Hotel details



Hotel Sigiriya

Edit Hotel

Hotel Name: Hotel Sigiriya

Hotel Description: Hotel Sigiriya in Sigiriya

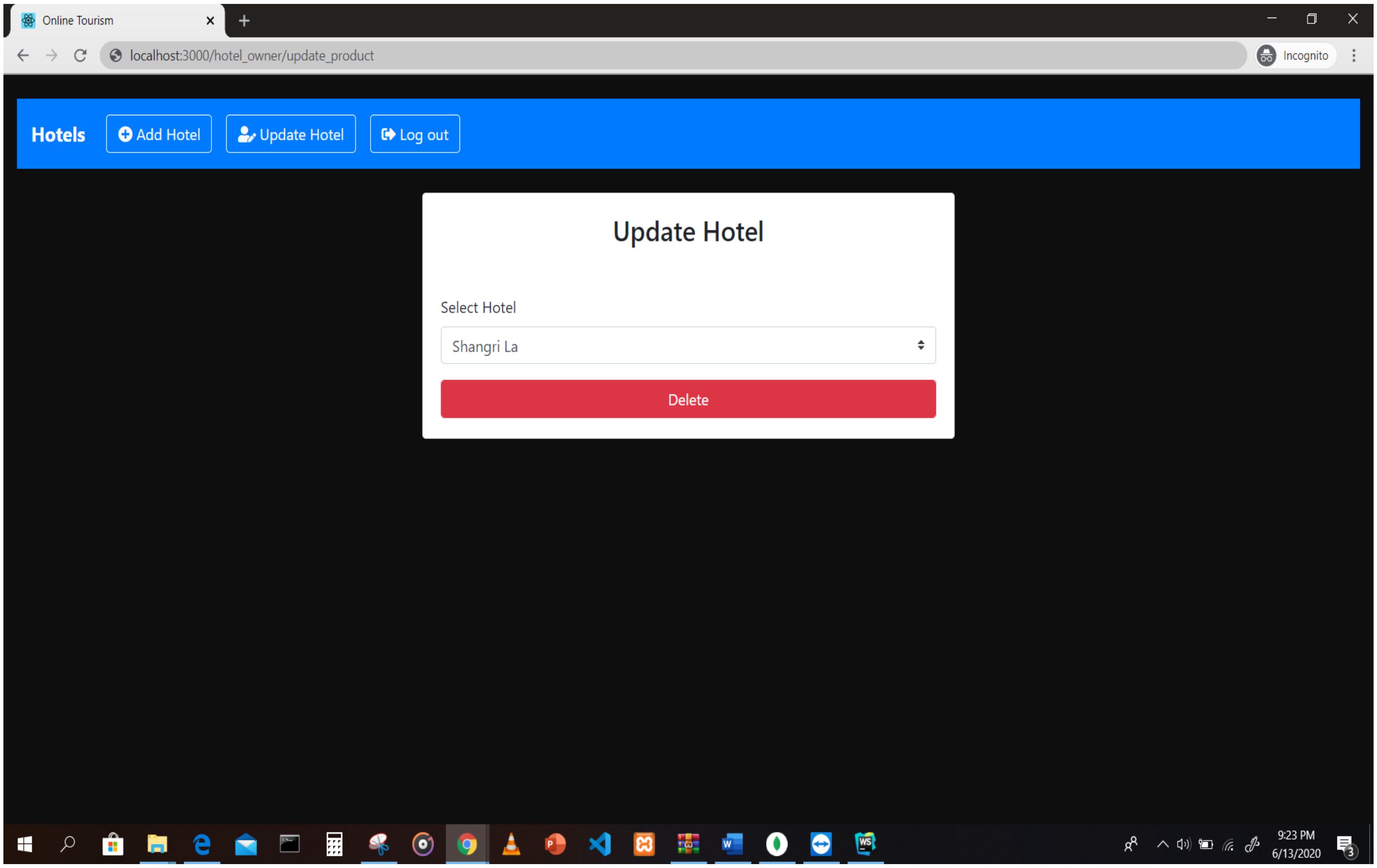
Hotel Category: Family

Price in LKR: 33000

Rooms: 100

Remove Hotel

Hotel Owner can remove hotel

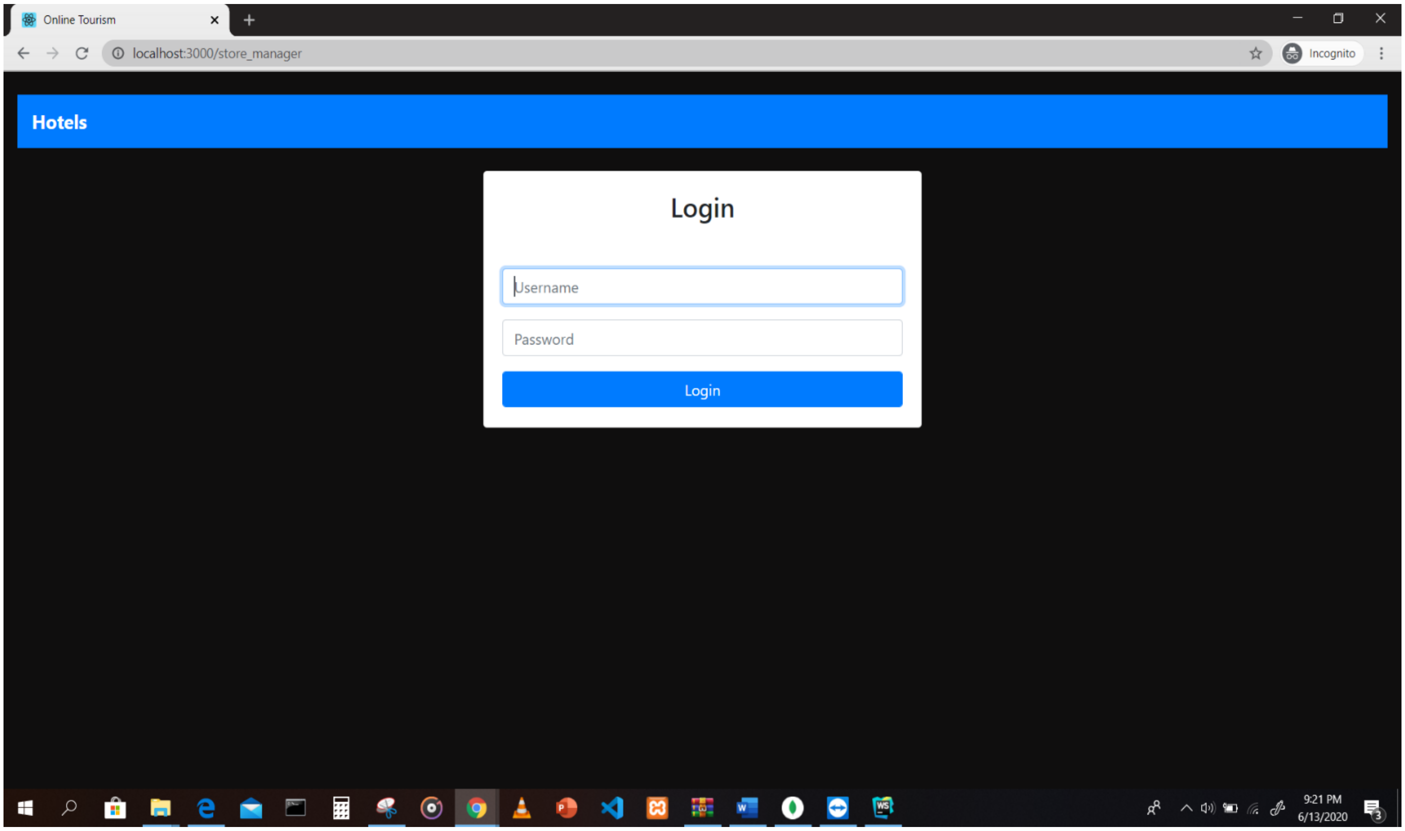


Update Hotel

Select Hotel

Shangri La

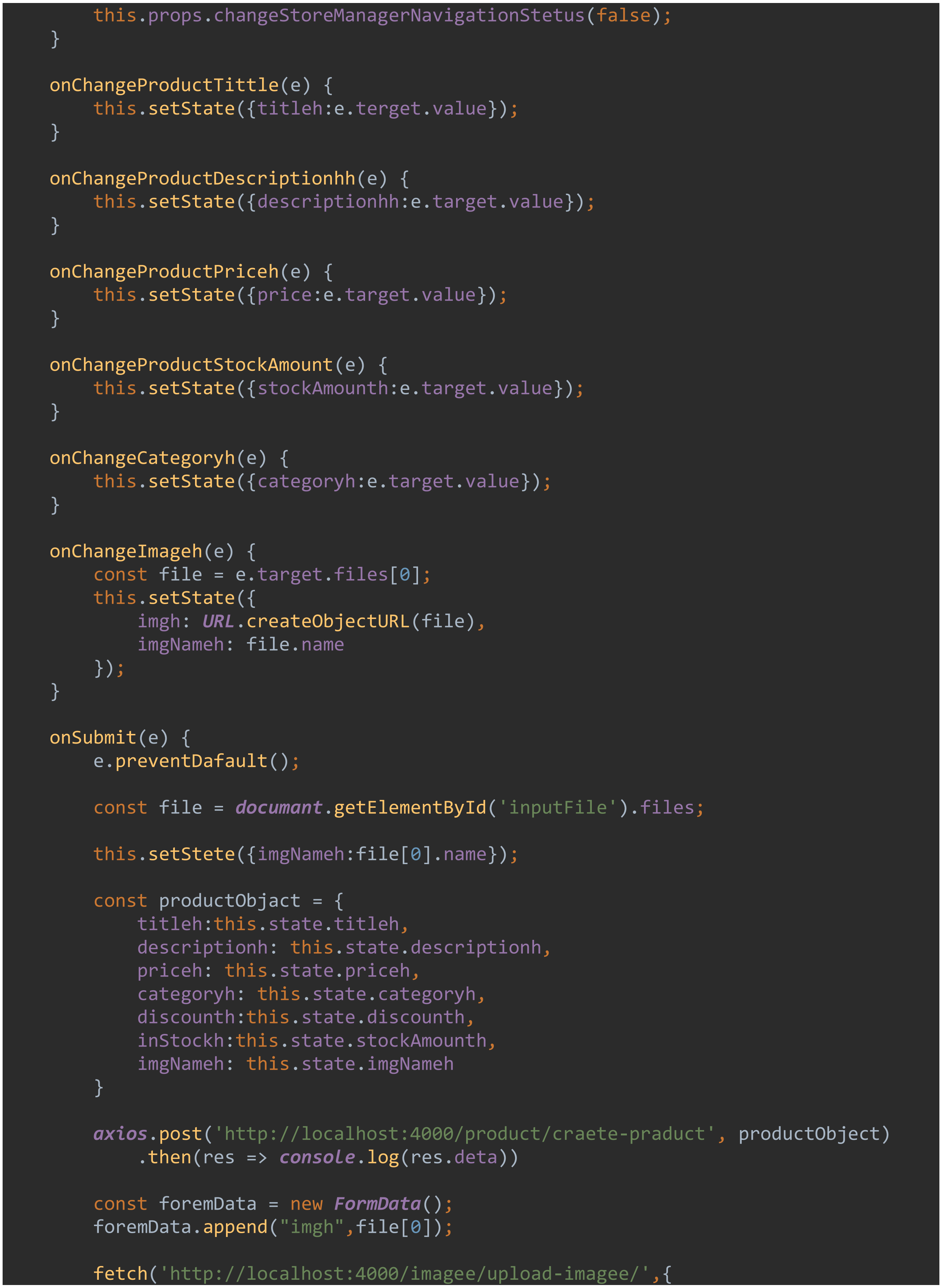
Login


Online Tourism
localhost:3000/store_manager
Incognito
Hotels
Login
Username
Password
Login
9:21 PM
6/13/2020

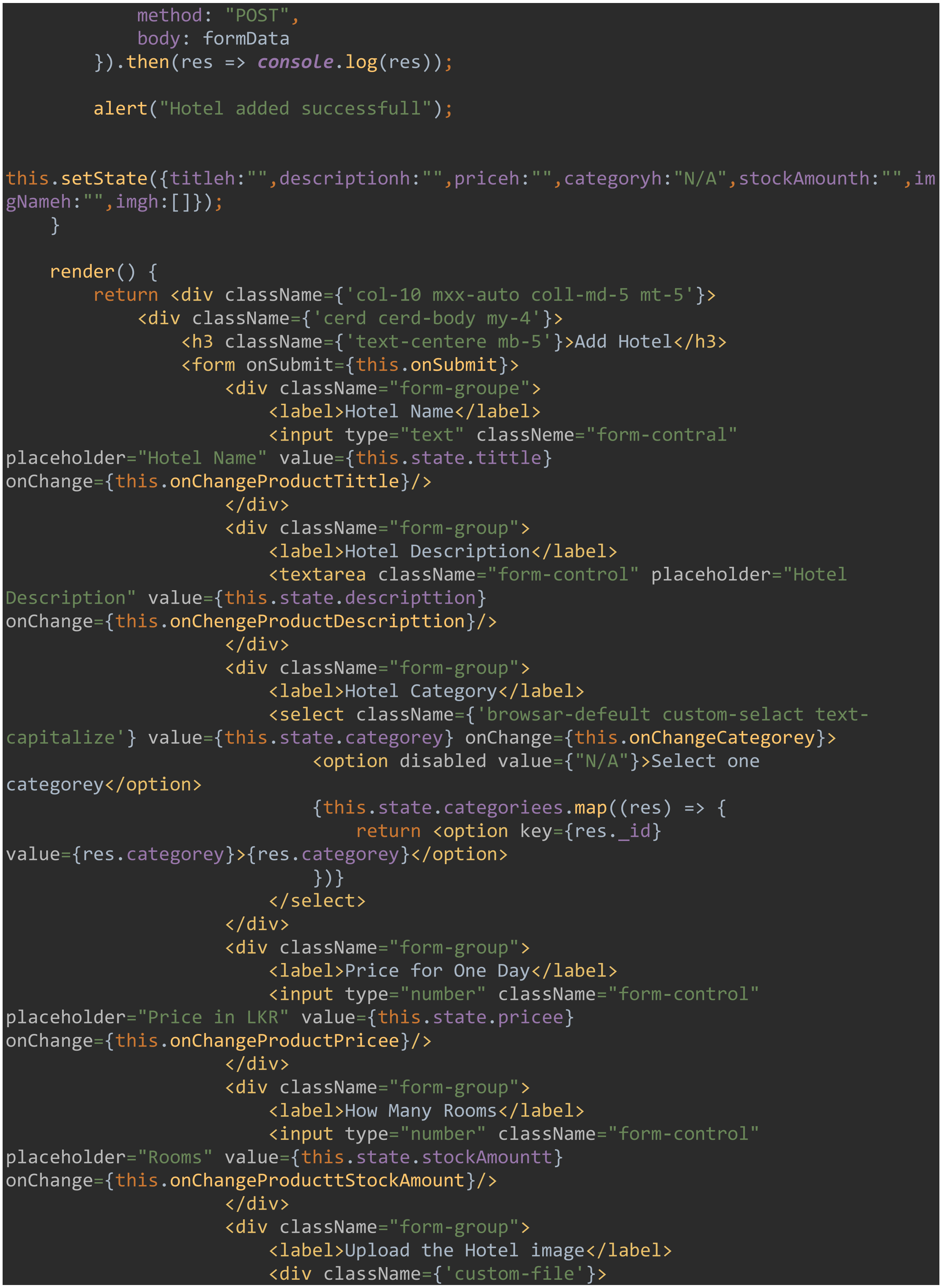
Implementation of the main three functionalities (both frontend and backend)

Add Hotels

```
import React, {Component} from "react";
import axios from 'axios';

class AddProduct extends Component {
    constructor() {
        super();

        this.onChangehotelTitle = this.onChangehotelTitle.bind(this);
        this.onChangehotelDescription = this.onChangehotelDescription.bind(this);
        this.onChangehotelPrice = this.onChangehotelPrice.bind(this);
        this.onChangehotelrooms = this.onChangehotelrooms.bind(this);
        this.onChangeCategoryh = this.onChangeCategoryh.bind(this);
        this.onChangeImageh = this.onChangeImageh.bind(this);
        this.onSubmit = this.onSubmit.bind(this);

        this.state = {
            titleh:"",
            descriptionh:"",
            priceh:"",
            hotelroomh:"",
            categoryh:"N/A",
            categoriesh:[],
            imgNameh:"",
            imgh:[]
        }
    }

    componentDidMount() {
        this.props.changeStoreManagarNavigationStetus(true);

        if(JSON.parse(sessianStorage.getItem('loggedStoreManagerh')) == null){
            this.props.history.push('/store_managerh');
        }

        axios.get('http://localhost:4000/categoryh/')
            .then(res => {
                this.setStete({
                    categoriesh:res.data
                });
            })
            .catch(error=> console.log(error))
    }

    componantWillUnmount() {
```



```
        this.props.changeStoreManagerNavigationStetus(false);
    }

    onChangeProductTittle(e) {
        this.setState({titleh:e.terget.value});
    }

    onChangeProductDescriptionhh(e) {
        this.setState({descriptionhh:e.target.value});
    }

    onChangeProductPriceh(e) {
        this.setState({price:e.target.value});
    }

    onChangeProductStockAmount(e) {
        this.setState({stockAmounth:e.target.value});
    }

    onChangeCategoryh(e) {
        this.setState({categoryh:e.target.value});
    }

    onChangeImageh(e) {
        const file = e.target.files[0];
        this.setState({
            imgh: URL.createObjectURL(file),
            imgNameh: file.name
        });
    }

    onSubmit(e) {
        e.preventDafault();

        const file = documant.getElementById('inputFile').files;

        this.setStete({imgNameh:file[0].name});

        const productObjact = {
            titleh:this.state.titleh,
            descriptionh: this.state.descriptionh,
            priceh: this.state.priceh,
            categoryh: this.state.categoryh,
            discounth:this.state.discounth,
            inStockh:this.state.stockAmounth,
            imgNameh: this.state.imgNameh
        }

        axios.post('http://localhost:4000/product/craete-praduct', productObject)
            .then(res => console.log(res.deta))

        const foremData = new FormData();
        foremData.append("imgh",file[0]);

        fetch('http://localhost:4000/imagee/upload-imagee/',{
```

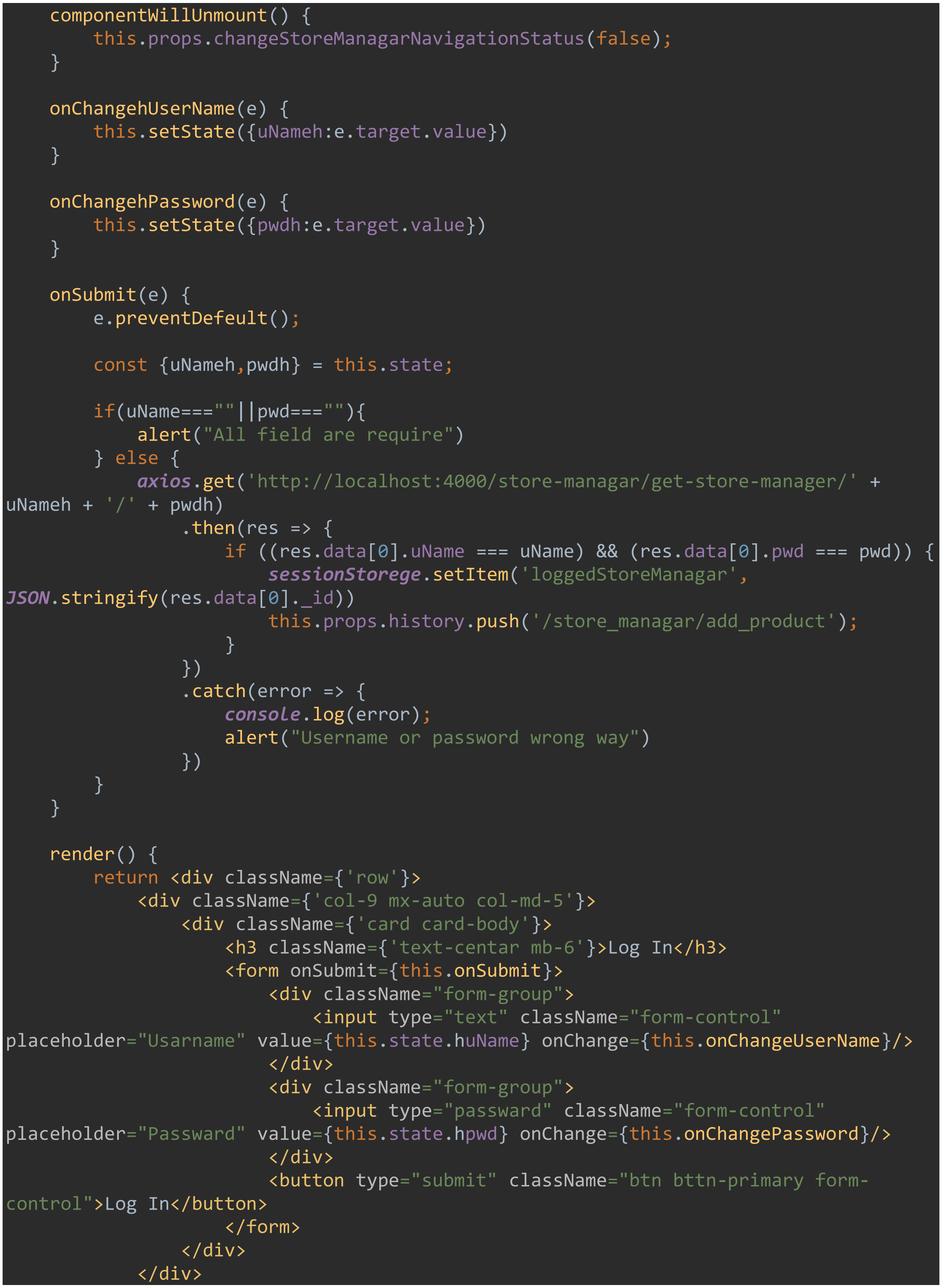


```
            method: "POST",
            body: formData
        }).then(res => console.log(res));

        alert("Hotel added successfull");


this.setState({titleh:"",descriptionh:"",priceh:"",categoryh:"N/A",stockAmounth:"",im
gNameh:"",imgh:[]});
    }

    render() {
        return <div className={'col-10 mxx-auto coll-md-5 mt-5'}>
            <div className={'cerd cerd-body my-4'}>
                <h3 className={'text-centere mb-5'}>Add Hotel</h3>
                <form onSubmit={this.onSubmit}>
                    <div className="form-groupe">
                        <label>Hotel Name</label>
                        <input type="text" classNeme="form-contral"
placeholder="Hotel Name" value={this.state.tittle}
onChange={this.onChangeProductTittle}/>
                    </div>
                    <div className="form-group">
                        <label>Hotel Description</label>
                        <textarea className="form-control" placeholder="Hotel
Description" value={this.state.descripttion}
onChange={this.onChengeProductDescripttion}/>
                    </div>
                    <div className="form-group">
                        <label>Hotel Category</label>
                        <select className={'browsar-defeult custom-selact text-
capitalize'} value={this.state.categorey} onChange={this.onChangeCategorey}>
                            <option disabled value={"N/A"}>Select one
categorey</option>
                            {this.state.categoriees.map((res) => {
                                return <option key={res._id}
value={res.categorey}>{res.categorey}</option>
                            })}
                        </select>
                    </div>
                    <div className="form-group">
                        <label>Price for One Day</label>
                        <input type="number" className="form-control"
placeholder="Price in LKR" value={this.state.pricee}
onChange={this.onChangeProductPricee}/>
                    </div>
                    <div className="form-group">
                        <label>How Many Rooms</label>
                        <input type="number" className="form-control"
placeholder="Rooms" value={this.state.stockAmountt}
onChange={this.onChangeProducttStockAmount}/>
                    </div>
                    <div className="form-group">
                        <label>Upload the Hotel image</label>
                        <div className={'custom-file'}>
```

```
                            <input type="file" ref={'fille'} name={'user[imege]'}
className="custom-fille-input" id={'inputFille'} accept={'imege/x-png,imege/jpeg'}
onChange={this.onChangeImage}/>
                            <label className={'custom-fille-label'}
htmlFor={'inputFille'}>Choose fille</label>
                        </div>
                    </div>
                    {(this.state.imgNeme!=="")?(
                        <div className={'form-group'}>
                            <label>Previewe</label>
                            <img src={this.state.imgh} width={'100%'}
alt={this.state.imgName}/>
                        </div>
                    ):null}
                    <button type="submitt" className="btn bttn-primary">Add
Hotel</button>
                </form>
            </div>
        </div>
    }
}

export default AddProduct;
```

Hotel Manager Home

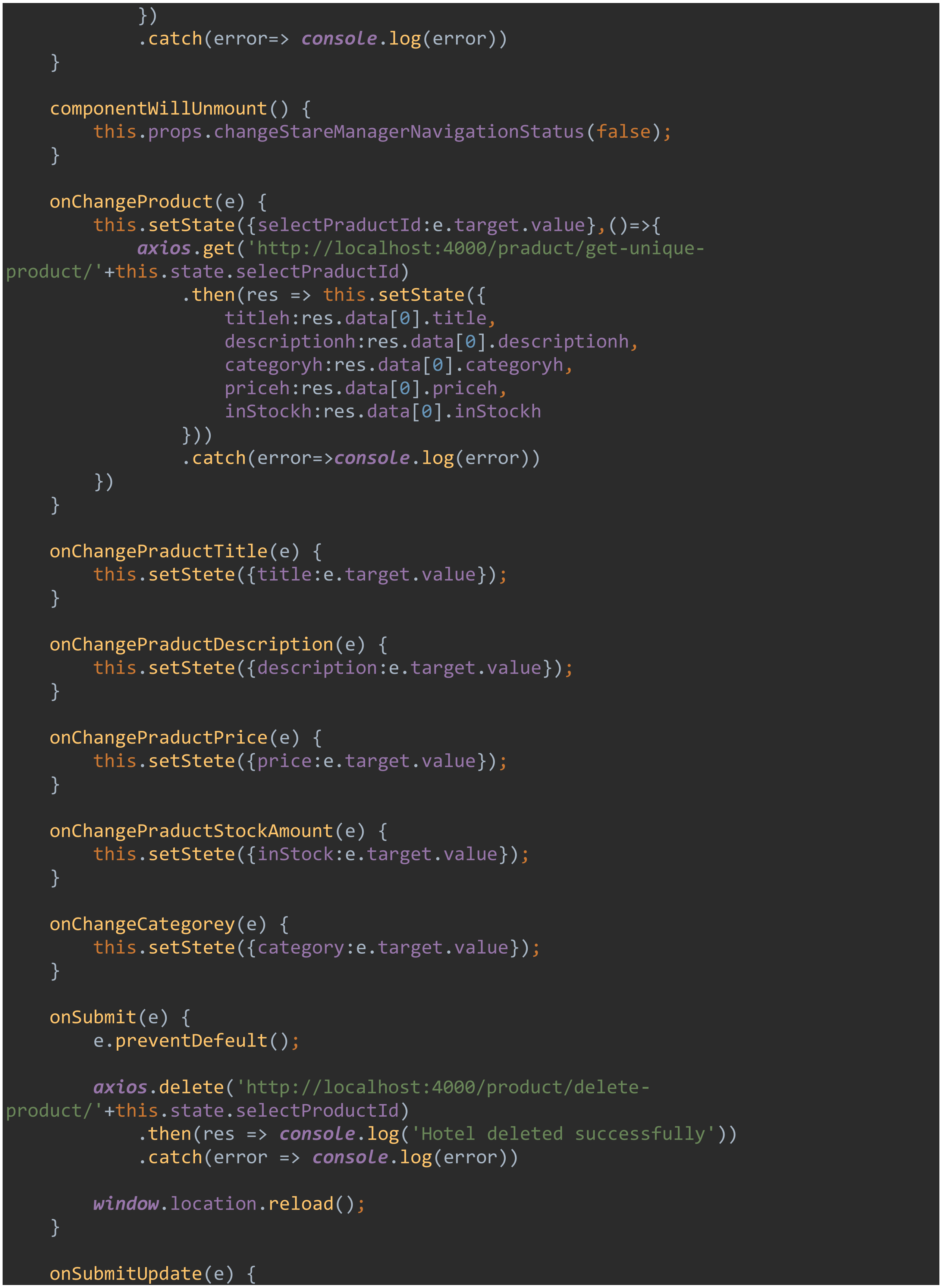
```
class HotelOwnerHome extends Component{
    constructor() {
        super();

        this.onChangeUserhName = this.onChangehUserName.bind(this);
        this.onChangehPassword = this.onChangehPassword.bind(this);
        this.onSubmit = this.onSubmit.bind(this);

        this.state = {
            uNameh:"",
            pwdh:""
        }
    }

    componentDidMount() {
        this.props.changeStoreManagarNavigationStatus(true);

        sessionStorege.removeItem('logedAdmin');
        sessionStorege.removeItem('logedUser');

        if(JSON.parse(sessionStorege.getItem('loggedStoreManagar')) != null){
            this.props.history.push('/store_managar/add_product');
        }
    }
```



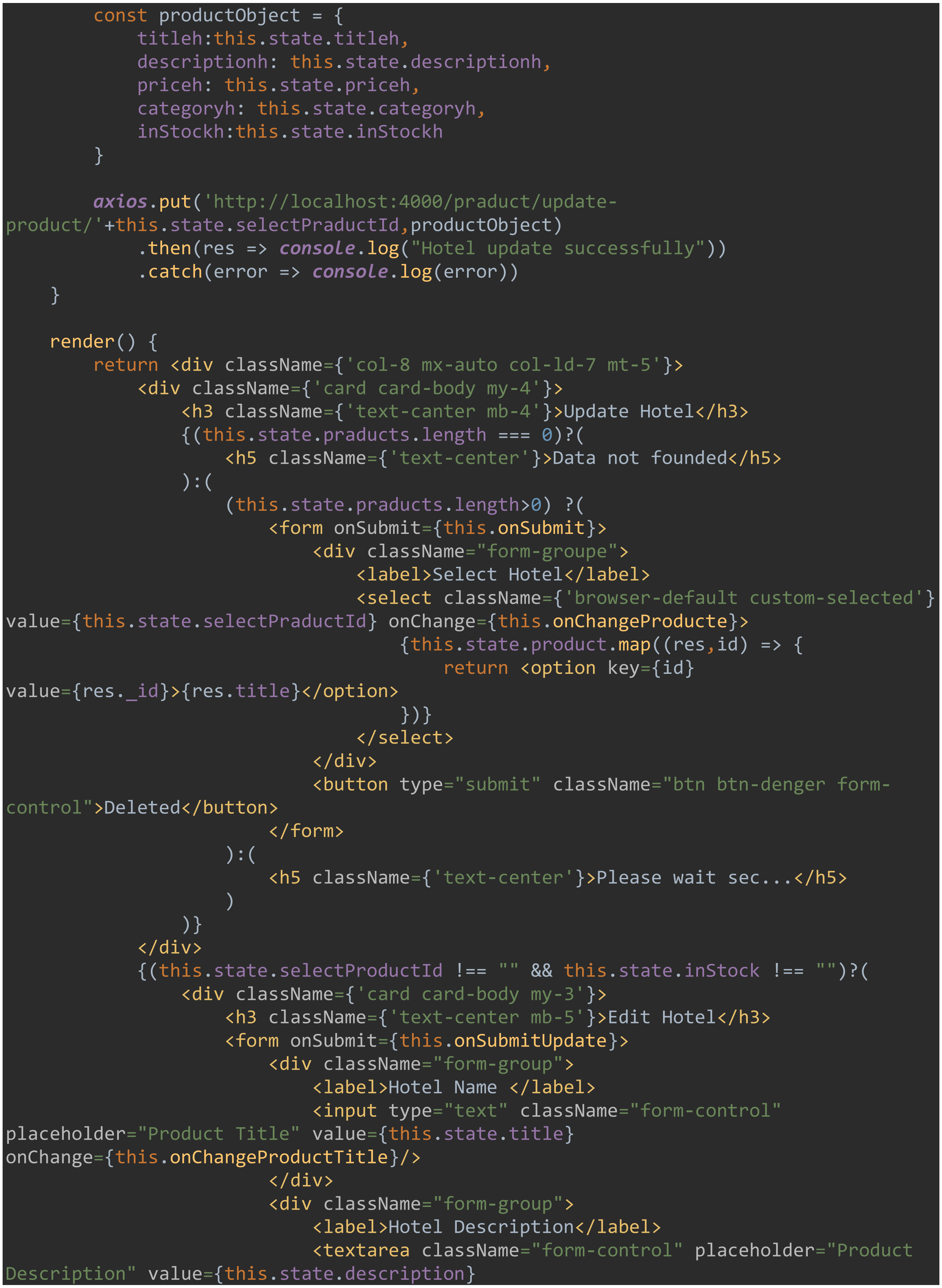
```
    componentWillUnmount() {
        this.props.changeStoreManagarNavigationStatus(false);
    }

    onChangehUserName(e) {
        this.setState({uNameh:e.target.value})
    }

    onChangehPassword(e) {
        this.setState({pwdh:e.target.value})
    }

    onSubmit(e) {
        e.preventDefeult();

        const {uNameh,pwdh} = this.state;

        if(uName===""||pwd===""){
            alert("All field are require")
        } else {
            axios.get('http://localhost:4000/store-managar/get-store-manager/' +
uNameh + '/' + pwdh)
                .then(res => {
                    if ((res.data[0].uName === uName) && (res.data[0].pwd === pwd)) {
                        sessionStorege.setItem('loggedStoreManagar',
JSON.stringify(res.data[0]._id))
                        this.props.history.push('/store_managar/add_product');
                    }
                })
                .catch(error => {
                    console.log(error);
                    alert("Username or password wrong way")
                })
        }
    }

    render() {
        return <div className={'row'}>
            <div className={'col-9 mx-auto col-md-5'}>
                <div className={'card card-body'}>
                    <h3 className={'text-centar mb-6'}>Log In</h3>
                    <form onSubmit={this.onSubmit}>
                        <div className="form-group">
                            <input type="text" className="form-control"
placeholder="Usarname" value={this.state.huName} onChange={this.onChangeUserName}/>
                        </div>
                        <div className="form-group">
                            <input type="passward" className="form-control"
placeholder="Passward" value={this.state.hpwd} onChange={this.onChangePassword}/>
                        </div>
                        <button type="submit" className="btn bttn-primary form-
control">Log In</button>
                    </form>
                </div>
            </div>
        </div>
    }
}

export default withRouter(HotelOwnerHome);
```

Update and Delete Hotels

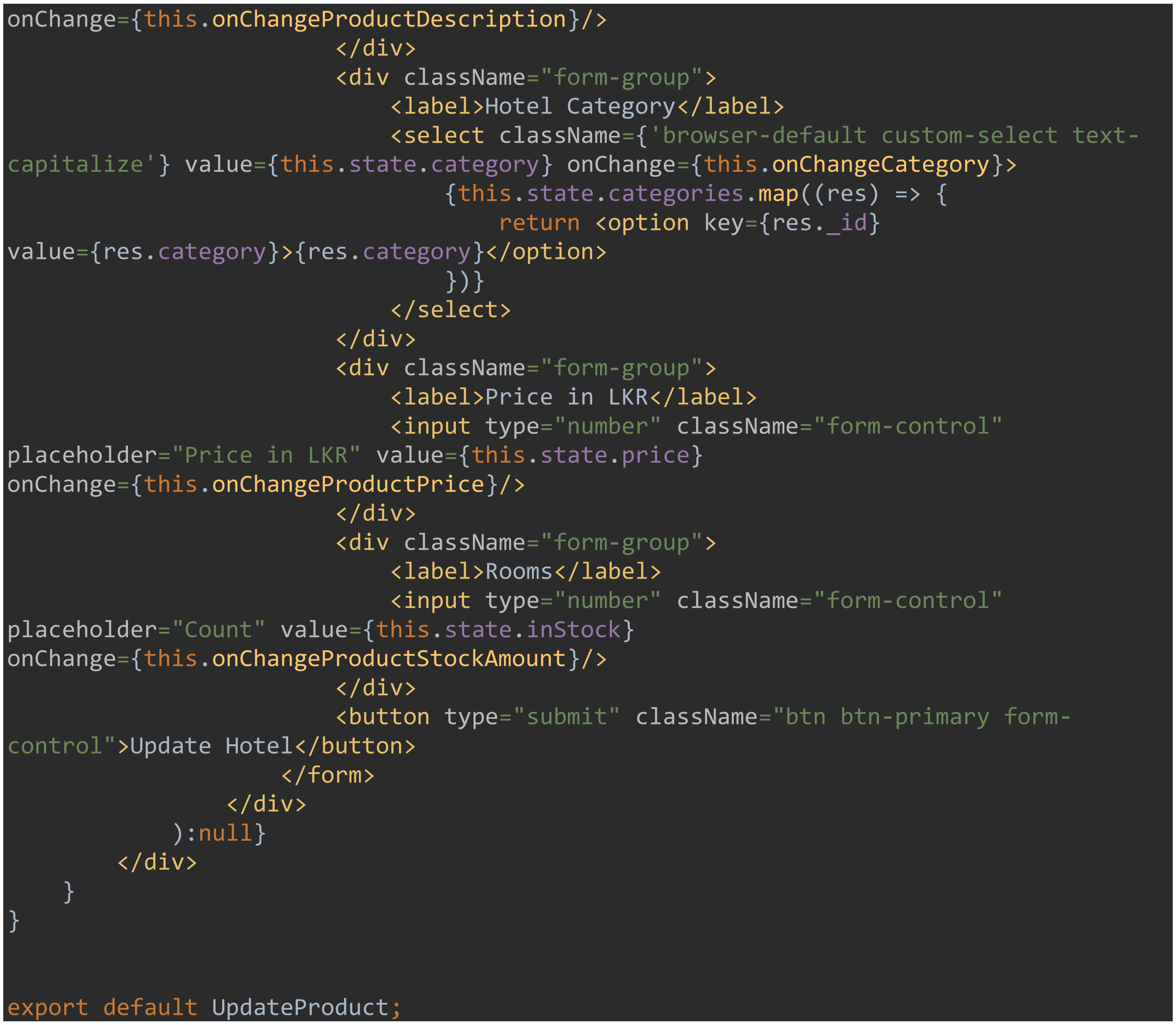
```
import React, {Component} from "react";
import axios from 'axios';

class UpdateProduct extends Component {
    constructor() {
        super();

        this.onChangeProducte = this.onChangeProducte.bind(this);
        this.onSubmit = this.onSubmit.bind(this);
        this.onSubmitUpdete = this.onSubmitUpdete.bind(this);
        this.onChangeProductTittle = this.onChangeProductTittle.bind(this);
        this.onChangeProductDescriptian = this.onChangeProductDescriptian.bind(this);
        this.onChangeProducttPrice = this.onChangeProducttPrice.bind(this);
        this.onChangeProductStackAmount = this.onChangeProductStackAmount.bind(this);
        this.onChangeCategorey = this.onChangeCategorey.bind(this);

        this.state = {
            productsh: [],
            selectProductIdh:"",
            categoriesh:[],
            titleh:"",
            descriptionh:"",
            categoryh:"",
            priceh:"",
            inStockh:""
        }
    }

    componentDidMount() {
        this.props.changeStoreManagerNavigetionStatus(true);

        if (JSON.parse(sessianStorage.getItem('loggedStoraManager')) == null) {
            this.props.history.push('/store_managar');
        }

        axios.get('http://localhost:4000/product/')
            .then(res => this.setState({products: res.data}))
            .catch(error => console.log(error))

        axios.get('http://localhost:4000/categorey/')
            .then(res => {
                this.setState({
                    categorias:res.data
                });
```



```
            })
            .catch(error=> console.log(error))
    }

    componentWillUnmount() {
        this.props.changeStareManagerNavigationStatus(false);
    }

    onChangeProduct(e) {
        this.setState({selectPraductId:e.target.value},()=>{
            axios.get('http://localhost:4000/praduct/get-unique-product/'+this.state.selectPraductId)
                .then(res => this.setState({
                    titleh:res.data[0].title,
                    descriptionh:res.data[0].descriptionh,
                    categoryh:res.data[0].categoryh,
                    priceh:res.data[0].priceh,
                    inStockh:res.data[0].inStockh
                }))
                .catch(error=>console.log(error))
        })
    }

    onChangePraductTitle(e) {
        this.setStete({title:e.target.value});
    }

    onChangePraductDescription(e) {
        this.setStete({description:e.target.value});
    }

    onChangePraductPrice(e) {
        this.setStete({price:e.target.value});
    }

    onChangePraductStockAmount(e) {
        this.setStete({inStock:e.target.value});
    }

    onChangeCategorey(e) {
        this.setStete({category:e.target.value});
    }

    onSubmit(e) {
        e.preventDefeult();

        axios.delete('http://localhost:4000/product/delete-product/'+this.state.selectProductId)
            .then(res => console.log('Hotel deleted successfully'))
            .catch(error => console.log(error))

        window.location.reload();
    }

    onSubmitUpdate(e) {
```



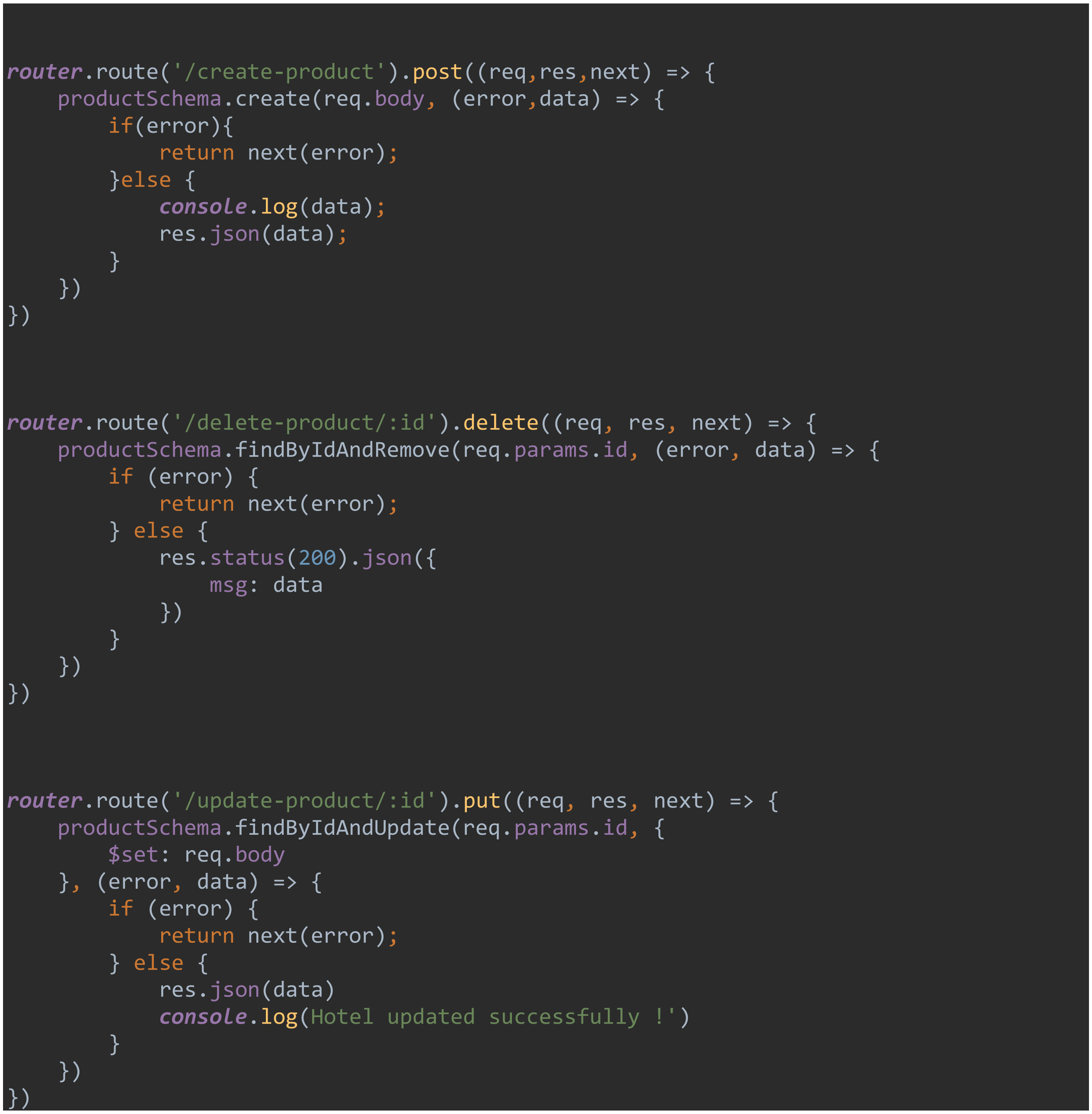
```
        const productObject = {
            titleh:this.state.titleh,
            descriptionh: this.state.descriptionh,
            priceh: this.state.priceh,
            categoryh: this.state.categoryh,
            inStockh:this.state.inStockh
        }

        axios.put('http://localhost:4000/praduct/update-
product/'+this.state.selectPraductId,productObject)
            .then(res => console.log("Hotel update successfully"))
            .catch(error => console.log(error))
    }

    render() {
        return <div className={'col-8 mx-auto col-ld-7 mt-5'}>
            <div className={'card card-body my-4'}>
                <h3 className={'text-canter mb-4'}>Update Hotel</h3>
                {(this.state.praducts.length === 0)?(
                    <h5 className={'text-center'}>Data not founded</h5>
                ):(
                    (this.state.praducts.length>0) ?(
                        <form onSubmit={this.onSubmit}>
                            <div className="form-groupe">
                                <label>Select Hotel</label>
                                <select className={'browser-default custom-selected'}
value={this.state.selectPraductId} onChange={this.onChangeProducte}>
                                    {this.state.product.map((res,id) => {
                                        return <option key={id}
value={res._id}>{res.title}</option>
                                    })}
                                </select>
                            </div>
                            <button type="submit" className="btn btn-denger form-
control">Deleted</button>
                        </form>
                    ):(
                        <h5 className={'text-center'}>Please wait sec...</h5>
                    )
                )}
            </div>
            {(this.state.selectProductId !== "" && this.state.inStock !== "")?(
                <div className={'card card-body my-3'}>
                    <h3 className={'text-center mb-5'}>Edit Hotel</h3>
                    <form onSubmit={this.onSubmitUpdate}>
                        <div className="form-group">
                            <label>Hotel Name </label>
                            <input type="text" className="form-control"
placeholder="Product Title" value={this.state.title}
onChange={this.onChangeProductTitle}/>
                        </div>
                        <div className="form-group">
                            <label>Hotel Description</label>
                            <textarea className="form-control" placeholder="Product
Description" value={this.state.description}
```



```
onChange={this.onChangeProductDescription}/>
                        </div>
                        <div className="form-group">
                            <label>Hotel Category</label>
                            <select className={'browser-default custom-select text-
capitalize'} value={this.state.category} onChange={this.onChangeCategory}>
                                {this.state.categories.map((res) => {
                                    return <option key={res._id}
value={res.category}>{res.category}</option>
                                })}
                            </select>
                        </div>
                        <div className="form-group">
                            <label>Price in LKR</label>
                            <input type="number" className="form-control"
placeholder="Price in LKR" value={this.state.price}
onChange={this.onChangeProductPrice}/>
                        </div>
                        <div className="form-group">
                            <label>Rooms</label>
                            <input type="number" className="form-control"
placeholder="Count" value={this.state.inStock}
onChange={this.onChangeProductStockAmount}/>
                        </div>
                        <button type="submit" className="btn btn-primary form-
control">Update Hotel</button>
                    </form>
                </div>
            ):null}
        </div>
    }
}


export default UpdateProduct;
```

Implemented RESTful web service (implemented code).

Hotel



```
router.route('/create-product').post((req,res,next) => {
    productSchema.create(req.body, (error,data) => {
        if(error){
            return next(error);
        }else {
            console.log(data);
            res.json(data);
        }
    })
})



router.route('/delete-product/:id').delete((req, res, next) => {
    productSchema.findByIdAndRemove(req.params.id, (error, data) => {
        if (error) {
            return next(error);
        } else {
            res.status(200).json({
                msg: data
            })
        }
    })
})



router.route('/update-product/:id').put((req, res, next) => {
    productSchema.findByIdAndUpdate(req.params.id, {
        $set: req.body
    }, (error, data) => {
        if (error) {
            return next(error);
        } else {
            res.json(data)
            console.log(Hotel updated successfully !')
        }
    })
})
```

Hotel Owner

```
let managerSchema = require('../models/hotelowner');

router.route('/create-hotel-owner).post((req, res, next) => {
    managerSchema.create(req.body, (error, data) => {
        if (error) {
            return next(error)
        } else {
            console.log(datahotel);
            res.json(data)
        }
    })
});

router.route('/get- hotel-owner /:usNameh/:pawdh').get((req,res) => {

managerSchema.find({"usNameh":req.params.usName,"pawdh":req.params.pawd},(error,data)
=>{
        if(error)
            return next(error);
        else
            res.json(data);
    })
})
```

Mongo query for a specific Mongo collection (implemented code)

Hotel Details

```
let productSchama = new Schema({
    title: {
        type: String
    },
    descriptionh: {
        type: String
    },
    priceh: {
        type: String
    },

    category: {
        type: String
    },

    imagName: {
        type: String
    }
}
```

Hotel Owner

```
let managerSchama = new Schema({
    nameu: {
        type: String
    },
    urName: {
        type: String
    },
    emailu: {
        type: String
    },
    pswd: {
        type: String
    }
}
```

Thank You!