

**BAHRIA UNIVERSITY (KARACHI CAMPUS**)

OPEN ENDED LAB 2 – Fall 2022

(CLO-5)

**Course Title:**  Web Engineering  **Course Code**: SEL 310

**Course Instructor: Engr. Adnan ur Rehman Class**: BSE-5(A)

**Lab Engineer:** **Engr. Rehan Baig Name: ZAIN SHAHID**

**Max. Marks:** 6 Marks **Reg no: 69983**

**Time: 120** Min

**Note:**

* **The probability of similarity is 0% hence Copied OEL will be marked as ZERO**
* Save Document (Containing Screenshot of Code and Result) of your solution with your name+enrollment like xyz015
* **Must Upload your code folder to Github Public Repository and share your repo URL in doc file.**

**[Marks 6]**

**Scenario:** The **US** based company **Rently** in its second phase of development now wanted to expand its Web SaaS Application to Mobile Devices and also have requirements for low end mobile devices for liteApp. Following are the Requirements

* **Fully Responsive UI (100% mobile optimized) For the Following Pages**
  1. Login
  2. Sign-up
  3. Dashboard
  4. Pay Rent
  5. Collect Rent
  6. Delete Account (With Confirmation – (hint: Use Soft Delete from database just set **isDeleted** check true))
* **Database (you are flexible to implement any database for this application)**
* **REST API’s (Well Commented code for understanding)  
  Following are the requirements for the API’s** 
  1. Use Rest API verbs
  2. APIs contains two types of URL’s one is **Public** **(Any User Can Use)** and other one is **Private** **(Authenticated Users can Access only)**  
     Login/Sign-up Api’s will use public URL’s  
     Dashboard/PayRent/CollectRent/DeleteAccount API’s will use Private URL’s
  3. All Delete Operations for the above API’s are Soft Delete Operations Means Every Delete Operation IsDeleted column will be true against the deleted record but data will not remove.  
       
     **Important Note**  
       
     You as an Engineer responsible for this application design, Storage and implementation phases from scratch with best practices of software engineering.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* GOOD LUCK \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

***REACT APP CODE:***

import React from 'react'

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

import Navbar from './components/Navbar';

import Signup from './components/Signup';

import Dashboard from './components/Home';

import Collect\_Rent from './components/About';

import Pay\_Rent from './components/PayRent';

const App = () =>{

return(

  <>

  <Navbar/>

 <Signup/>

 <Collect\_Rent/>

<Dashboard/>

<Pay\_Rent/>

</>

   )

}

export default App

***Navbar Code:***

import React from 'react'

//import 'bootstrap/dist/css/bootstrap.css'

const Navbar = () =>{

return(

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

  <a class="navbar-brand" href="/">Navbar</a>

  <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="./Home.js">Dashboard<span class="sr-only">(current)</span></a>

      </li>

      <li class="nav-item">

        <a class="nav-link" href="./About.js">Collect Rent</a>

      </li>

      <li class="nav-item">

        <a class="nav-link" href="./Contact.js">Pay Rent</a>

      </li>

      <li class="nav-item">

        <a class="nav-link" href="./Login.js">Login</a>

      </li>

      <li class="nav-item">

        <a class="nav-link" href="./Signup.js">Registeration</a>

      </li>

    </ul>

  </div>

</nav>

)

}

export default Navbar

***Signup and login Code:***

import React from 'react'

const Signup = () =>{

return(

    <section className='Signup'>

<div className='container mt-5'>

<div className="Signup-content">

<div className="Signup-form">

     <center><h2 className="form-title">Signup</h2></center>

     <form className='register-form ' id='register-form'>

     <form>

     <form>

  <div class="form-row">

    <div class="form-group col-md-6">

      <label for="inputEmail4">Name</label>

      <input type="text" class="form-control" id="inputEmail4" placeholder="Name"/>

    </div>

    <div class="form-group col-md-6">

    <label for="inputAddress">Email</label>

    <input type="email" class="form-control" id="inputAddress" placeholder="abc@gmail.com"/>

  </div>

  <div class="form-group col-md-6">

    <label for="inputAddress">Phone</label>

    <input type="number" class="form-control" id="inputAddress" placeholder="+92 3152143937"/>

  </div>

  <div class="form-group col-md-6">

    <label for="inputAddress2">Work</label>

    <input type="text" class="form-control" id="inputAddress2" placeholder="Software Engineer"/>

  </div>

  <div class="form-group col-md-6">

      <label for="inputPassword4">Password</label>

      <input type="password" class="form-control" id="inputPassword4" placeholder="Password"/>

    </div>

  </div>

  <div class="form-row">

    <div class="form-group col-md-6">

      <label for="inputCity">Confirm Password</label>

      <input type="Password" class="form-control" id="inputCity"/>

    </div>

    </div>

  <div class="form-group">

    <div class="form-check">

      <input class="form-check-input" type="checkbox" id="gridCheck"/>

      <label class="form-check-label" for="gridCheck">

        Check me out

      </label>

    </div>

  </div>

  <center><button type="submit" class="btn btn-primary">Sign Up</button></center>

</form>

</form>

     </form>

</div>

</div>

</div>

    </section>

)

}

export default Signup

***Signup Output:***

***Graphical user interface, text, application, email

Description automatically generated***

***Graphical user interface, application

Description automatically generated***

ZAIN

***DATA IN MONGO DB***

***Graphical user interface, text, application

Description automatically generated***

ZAIN

ZAIN

***LoginCode:***

import React from 'react'

const Login = () =>{

return(

    <section className='Sign In'>

    <div className='container mt-5'>

    <div className="Signup-content">

    <div className="Sign In-form">

         <center><h2 className="form-title">Sign In</h2></center>

        <center> <form className='register-form ' id='register-form'>

    <div class="form-group col-md-6">

    <label for="inputAddress">Email</label>

    <input type="email" class="form-control" id="inputAddress" placeholder="abc@gmail.com"/>

  </div>

  <div class="form-group col-md-6">

    <label for="inputAddress">Password</label>

    <input type="Password" class="form-control" id="inputAddress" placeholder="............................."/>

    </div>

    <center><button type="submit" class="btn btn-primary">Login</button></center>

</form>

</center>

</div>

</div>

</div>

</section>

)

}

export default Login

***Output:***

***Graphical user interface, application, email

Description automatically generated***

***Dashoboard Code:***

***PayRent:***

  return (

      <form onSubmit={onSubmit}>

          <h1>Add Rent</h1>

          {/\* Description \*/}

          <div className="form-control">

            <label htmlFor="description">Description</label>

            <input type="text" id='description' name='description' value={transaction.description} onChange={onchange} placeholder='Enter Description' />

          </div>

          {/\* Amount \*/}

          <div className="form-control">

            <label htmlFor="amount">Amount</label>

            <input type="number" id='amount' name='amount' value={transaction.amount} onChange={onchange} placeholder='Enter Amount' />

          </div>

          {/\* Type \*/}

          <div className="form-control">

            <label htmlFor="">Type</label>

            <div>

              <input type="radio" id='income' name='type' value={"cash"} checked={transaction.type === 'income'} onChange={onchange} /> <label htmlFor="income">Cash</label>

              <input type="radio" id='expenses' name='type' value={"check"} checked={transaction.type === 'expenses'} onChange={onchange} /> <label htmlFor="expenses">check</label>

            </div>

          </div>

          {/\* Submit \*/}

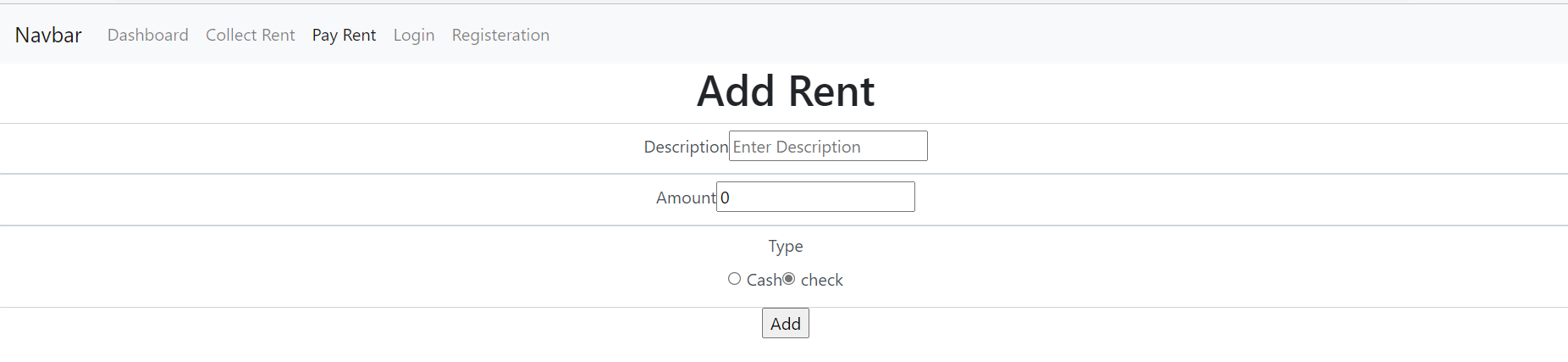
          <input type="submit" value="Add" />

      </form>

    )

  }

  export default Form

******

***Collect Rent:***

 return (

      <form onSubmit={onSubmit}>

          <h1>Collect Rent</h1>

          {/\* Description \*/}

          <div className="form-control">

            <label htmlFor="description">Description</label>

            <input type="text" id='description' name='description' value={transaction.description} onChange={onchange} placeholder='Enter Description' />

          </div>

          {/\* Amount \*/}

          <div className="form-control">

            <label htmlFor="amount">Amount</label>

            <input type="number" id='amount' name='amount' value={transaction.amount} onChange={onchange} placeholder='Enter Amount' />

          </div>

          {/\* Type \*/}

          <div className="form-control">

            <label htmlFor="">Type</label>

            <div>

              <input type="radio" id='income' name='type' value={"cash"} checked={transaction.type === 'income'} onChange={onchange} /> <label htmlFor="income">Cash</label>

              <input type="radio" id='expenses' name='type' value={"check"} checked={transaction.type === 'expenses'} onChange={onchange} /> <label htmlFor="expenses">check</label>

            </div>

          </div>

          {/\* Submit \*/}

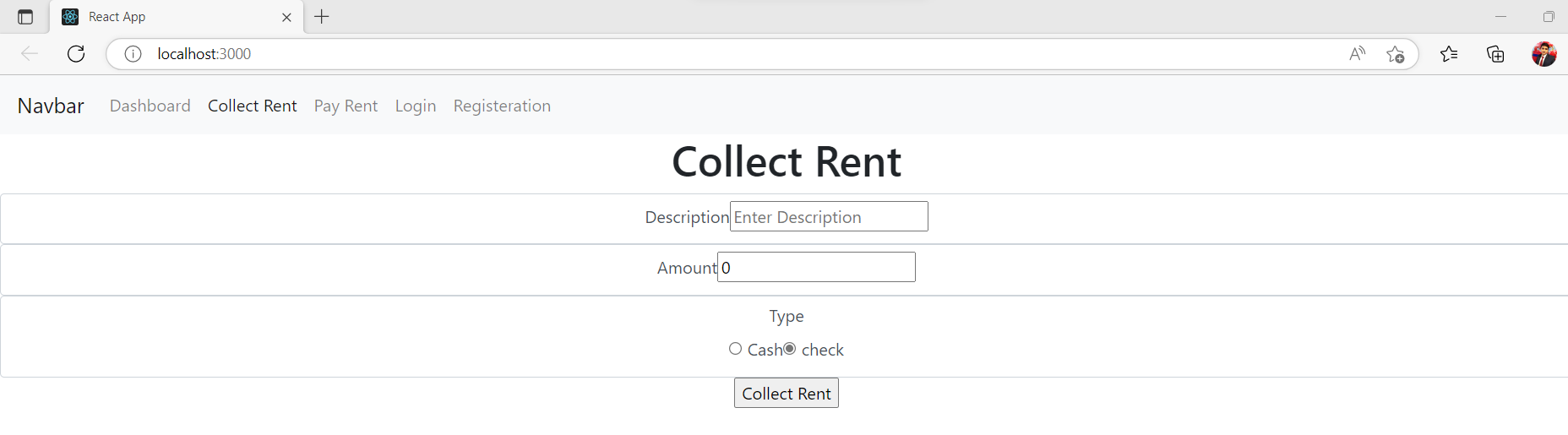
          <input type="submit" value="Collect Rent" />

      </form>

    )

  }

  export default Form

******

***BACKEND CODE OF NODE:***

***App.js Class Code:***

const dotenv = require("dotenv");

const mongoose = require('mongoose');

const express = require('express');

const app = express();

dotenv.config({path: './config.env'});

require('./db/conn');

//const User = require('./model/userSchema');

app.use(express.json());

//link the router file

app.use(require('./router/auth'));

const PORT = process.env.PORT;

// middleware

const middleware =(req,res,next) =>{

console.log('Hello My Middle ware');

next();

}

//middleware();

//app.get('/',(req,res)=> {

  //  res.send('Hello World from server app.js')

   // });

app.get('/about',middleware,(req,res)=> {

  console.log('hello my about');

    res.send('Hello  about World from server')

    });

    app.get('/contact',(req,res)=> {

        res.send('Hello contact  World from server')

        });

        app.get('/signup',(req,res)=> {

            res.send('Hello signup World from server')

            });

            app.get('/signin',(req,res)=> {

                res.send('Hello signin World from server')

                });

app.listen(PORT,() =>{

    console.log('server is running at port '+PORT );

});

***Config.env***

PORT=4000

DATABASE=mongodb://0.0.0.0:27017/2707

***Package.json***

{

  "name": "mernapp",

  "version": "1.0.0",

  "description": "",

  "main": "index.js",

  "scripts": {

    "test": "echo \"Error: no test specified\" && exit 1"

  },

  "keywords": [],

  "author": "",

  "license": "ISC",

  "dependencies": {

    "bcryptjs": "^2.4.3",

    "dotenv": "^16.0.3",

    "express": "^4.18.2",

    "jsonwebtoken": "^9.0.0",

    "mongoose": "^6.8.3",

    "nodemon": "^2.0.20"

  }

}

***Router Code:***

const { response } = require('express');

const express = require('express');

const router = express.Router();

const bcrypt = require('bcryptjs');

require('../db/conn');

const User = require("../model/userSchema");

router.get('/get',(req,res)=> {

    res.send('Hello World from server router js')

    });

    router.post('/register', (req, res) =>{

        const { name , email, phone , work , Password , ConfirmPassword} = req.body;

if(!name || !email|| !phone|| !work|| !Password|| !ConfirmPassword){

    return res.status(422).json({error:"Please fill the all required fields"});

}

User.findOne({ email: email}).then((userExist) =>{

if(userExist){

    return res.status(422).json({error:"This Email has already registered"});

}

else if (Password != ConfirmPassword){

    return res.status(422).json({error:"Password is not Identical"});

}

else{

    const user = new User({name ,email , phone, work, Password, ConfirmPassword });

   user.save();

    res.status(201).json({message:"User Succesfully Registered"});

}

}).catch(err =>{console.log(err);});

//response.send("response");

//res.json({message:req.body});

    });

//Login Route

router.post('/SignIn', async(req, res) => {

   //console.log(req.body);

    //res.json({message: "awesome"})

    try{

        const {email,Password} = req.body;

        if(!email || !Password){

            return res.status(400).json({error:"Please fill the data"})

        }

        const userLogin = await User.findOne({email:email});

//console.log(userLogin);

if(userLogin){

    const isMatch = await bcrypt.compare(Password, userLogin.Password);

    if(!isMatch){

        res.status(400).json({ error:"Invalid Credidentials pass"});

    }

    else{

        res.json({ message:"User Sign In SuccesFully"});

    }

}

else{

    res.json({ error:"Invalid Credidentials"});

}

    } catch(err){

         console.log(err);

    }

});

    module.exports = router;

***UserSchema Code:***

 const mongoose = require('mongoose');

const bcrypt = require('bcryptjs');

 const userSchema = new mongoose.Schema({

name:{

    type: String,

    required: [true,"Please Add a Name"],

},

email:{

    type: String,

    required: [true,"Please Add An Email"],

    unique: true,

    match:[

        /^(([^<>()[\]\\.,;:\s@"]+(\.[^<>()[\]\\.,;:\s@"]+)\*)|(".+"))@((\[[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\])|(([a-zA-Z\-0-9]+\.)+[a-zA-Z]{2,}))$/,

        "Please Enter a Valid Email"

    ]

},

phone:{

    type: Number,

    required: [true,"Please Add a Number"],

},

work:{

    type: String,

    required: [true,"Please Add a Work"],

},

Password:{

    type: String,

    required: [true,"Please Add a Password"],

},

ConfirmPassword:{

    type: String,

    required: [true,"Please Add a Confirm Password"],

}

 })

 // hashing of password

 userSchema.pre('save', async function(next){

console.log("hi from inside");

    if(this.isModified('Password')){

            this.Password = await bcrypt.hash(this.Password,12);

            this.ConfirmPassword = await bcrypt.hash(this.ConfirmPassword,12);

    }

    next();

 });

 const User = mongoose.model('DB',userSchema);

 module.exports = User;

***Connection.cs***

const mongoose = require('mongoose');

mongoose.set('strictQuery', true);

const db = process.env.DATABASE;

mongoose.connect(db,{

}).then(() => {

console.log('connection successfull');

}).catch((err) => console.log('no connection'));