Name: Patel Riya Nitinbhai

041

Pratical assignment -2

1. Express File upload (single, multiple) with validations.
2. const express = require('express');
3. const multer = require('multer');
4. const path = require('path');
5. const app = express();
6. app.use(express.static('public'));
7. var option=multer.diskStorage({destination:function(req,file,cb){
8. if(file.mimetype !=='image/jpeg'){
9. return cb('invaid file formate');
10. }
11. cb(null,'./uploads');
12. },filename:function(req,file,cb){
13. cb(null,(Math.random().toString(30)).slice
14. (5,10) + Date.now() +
15. path.extname(file.originalname));
16. }});
17. var upload=multer({storage:option});
18. app.get("/",function(req,res){
19. res.send("<a href='page\_upload.html'>upload your file here</a>");
20. })
21. app.post("/file\_upload",upload.single("profile\_pic"),function(req,res){
22. res.send("file is uploaded");
23. });
24. app.post("/upload-multiple-images",upload.array("multiple\_images",2),function(req,res){
25. res.send("multiple file is uploded");
26. })
27. app.listen(8000);

<html>

    <head>

        <title>file uploading</title>

    </head>

    <body>

        <h3>upload file : </h3>

        select file for upload:<br>

        <form action="/file\_upload" method="POST" enctype="multipart/form-data">

            <div>

                <label>Select your profile picture:</label>

                <input type="file" name="profile\_pic" size="500" />

            </div>

            <div>

                <input type="submit"  value="Upload file" />

            </div>

        </form>

    </body>

</html>

<br>

<form method="POST" action="/upload-multiple-images" enctype="multipart/form-data">

    <div>

        <label>Select multiple images:</label>

        <input type="file" name="multiple\_images" multiple />

    </div>

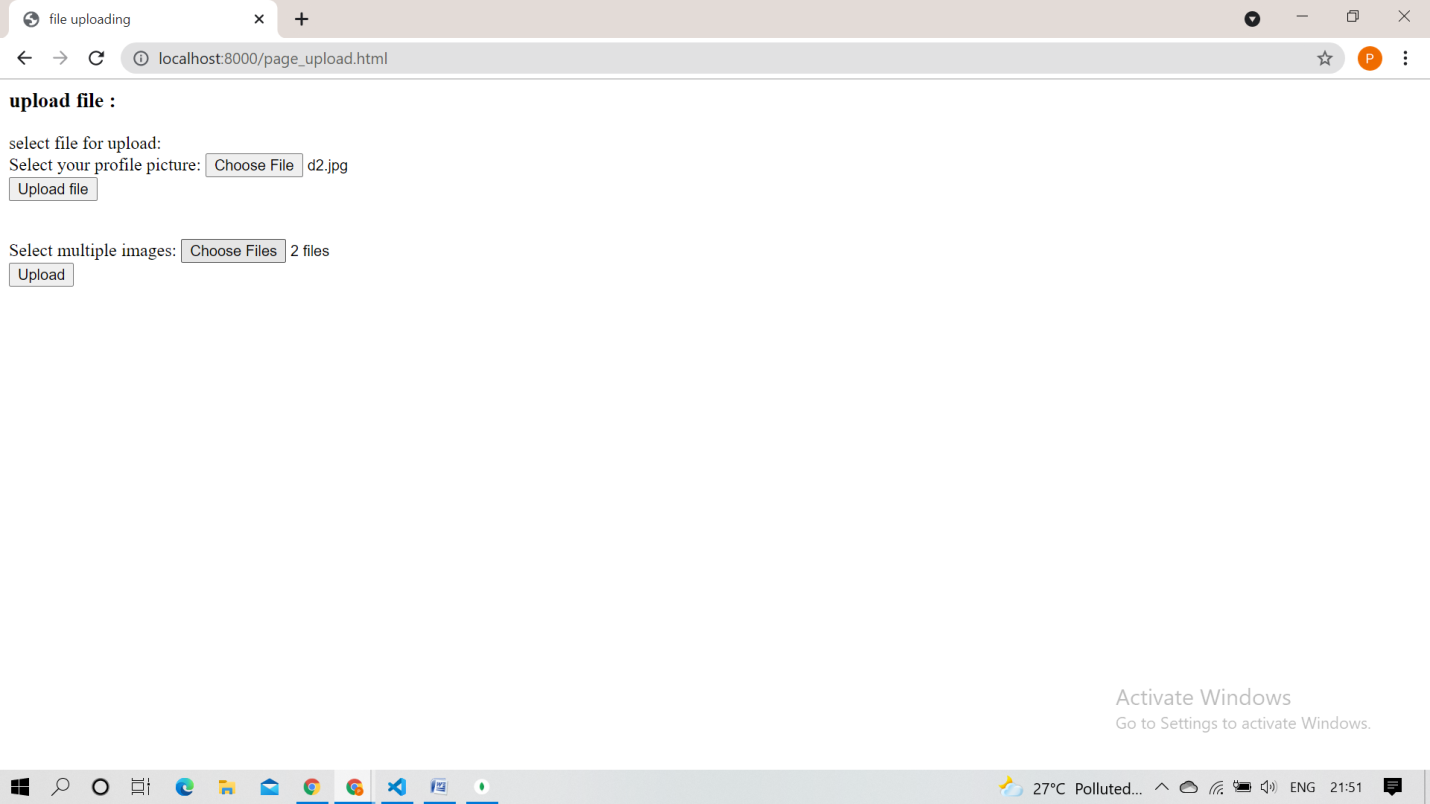
    <div>

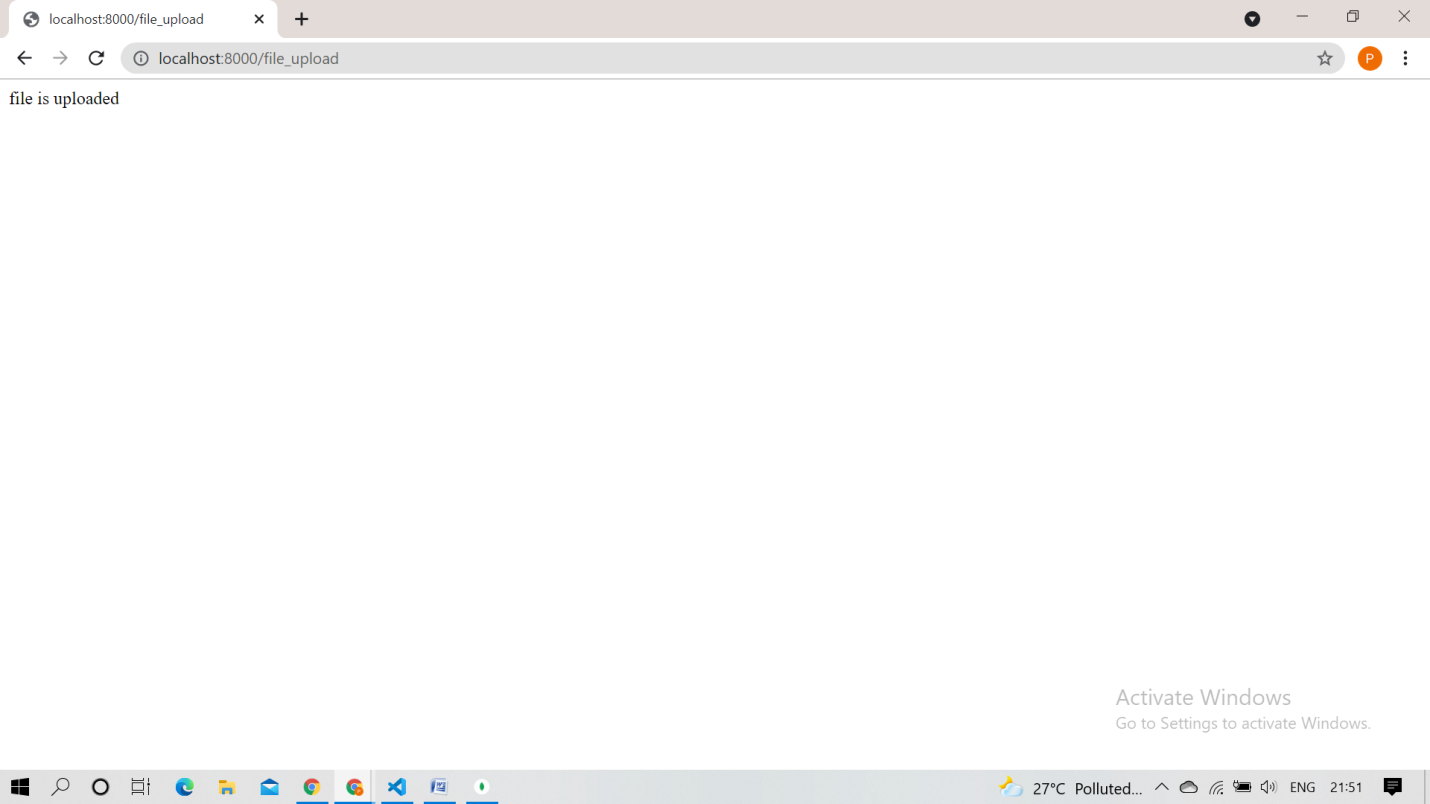
        <input type="submit" value="Upload" />

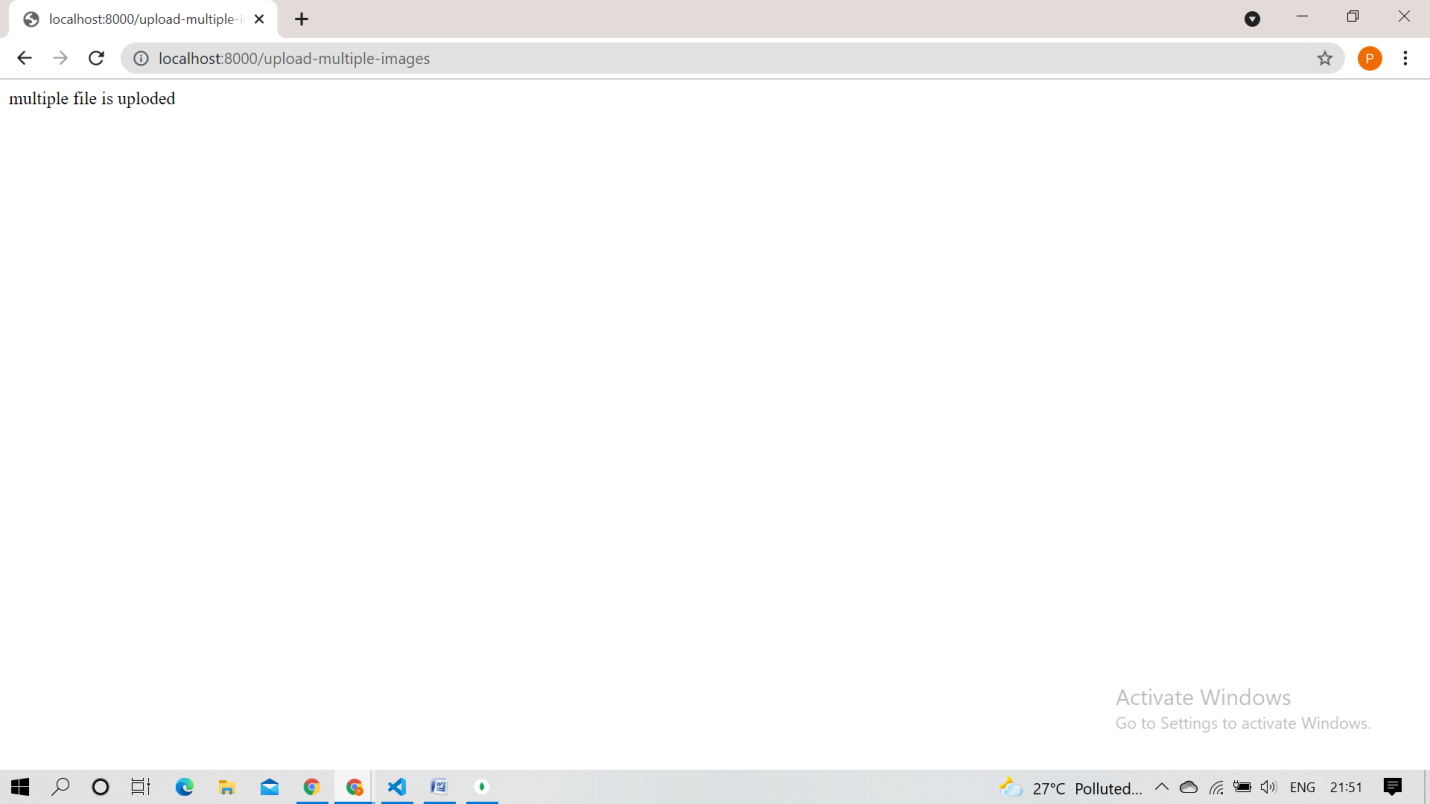
    </div>

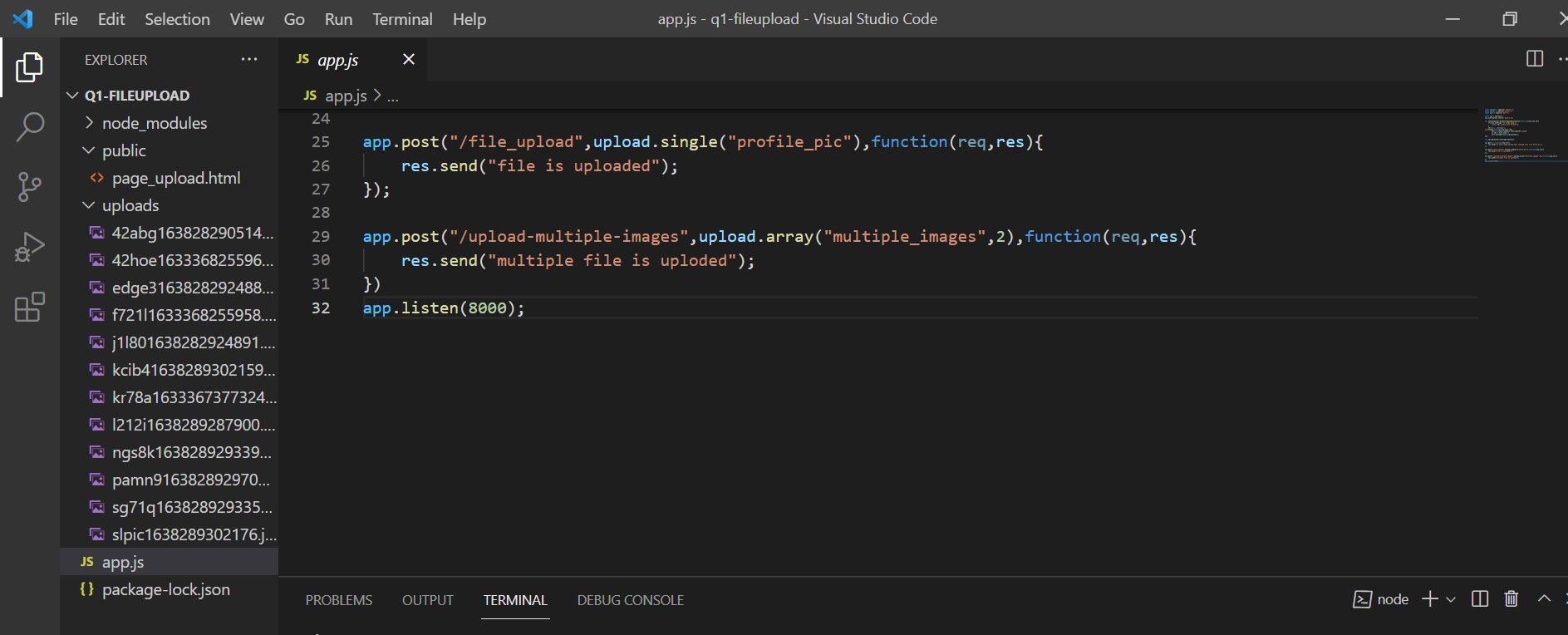
</form>

Output:









2) Express Login application with file session store

const express = require('express');

const cookieParser = require("cookie-parser");

const sessions = require('express-session');

const app = express();

const PORT = 4000;

const oneDay = 1000 \* 60 \* 60 \* 24;

//session middleware

app.use(sessions({

    secret: "thisismysecrctekeyfhrgfgrfrty84fwir767",

    saveUninitialized:true,

    cookie: { maxAge: oneDay },

    resave: false

}));

// parsing the incoming data

app.use(express.json());

app.use(express.urlencoded({ extended: true }));

//serving public file

app.use(express.static(\_\_dirname));

// cookie parser middleware

app.use(cookieParser());

//username and password

const myusername = 'user1'

const mypassword = 'mypassword'

// a variable to save a session

var session;

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

    session=req.session;

    if(session.userid){

        res.send("Welcome User <a href=\'/logout'>click to logout</a>");

    }else

    res.sendFile('views/index.html',{root:\_\_dirname})

});

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

    if(req.body.username == myusername && req.body.password == mypassword){

        session=req.session;

        session.userid=req.body.username;

        console.log(req.session)

        res.send(`Hey there, welcome <a href=\'/logout'>click to logout</a>`);

    }

    else{

        res.send('Invalid username or password');

    }

})

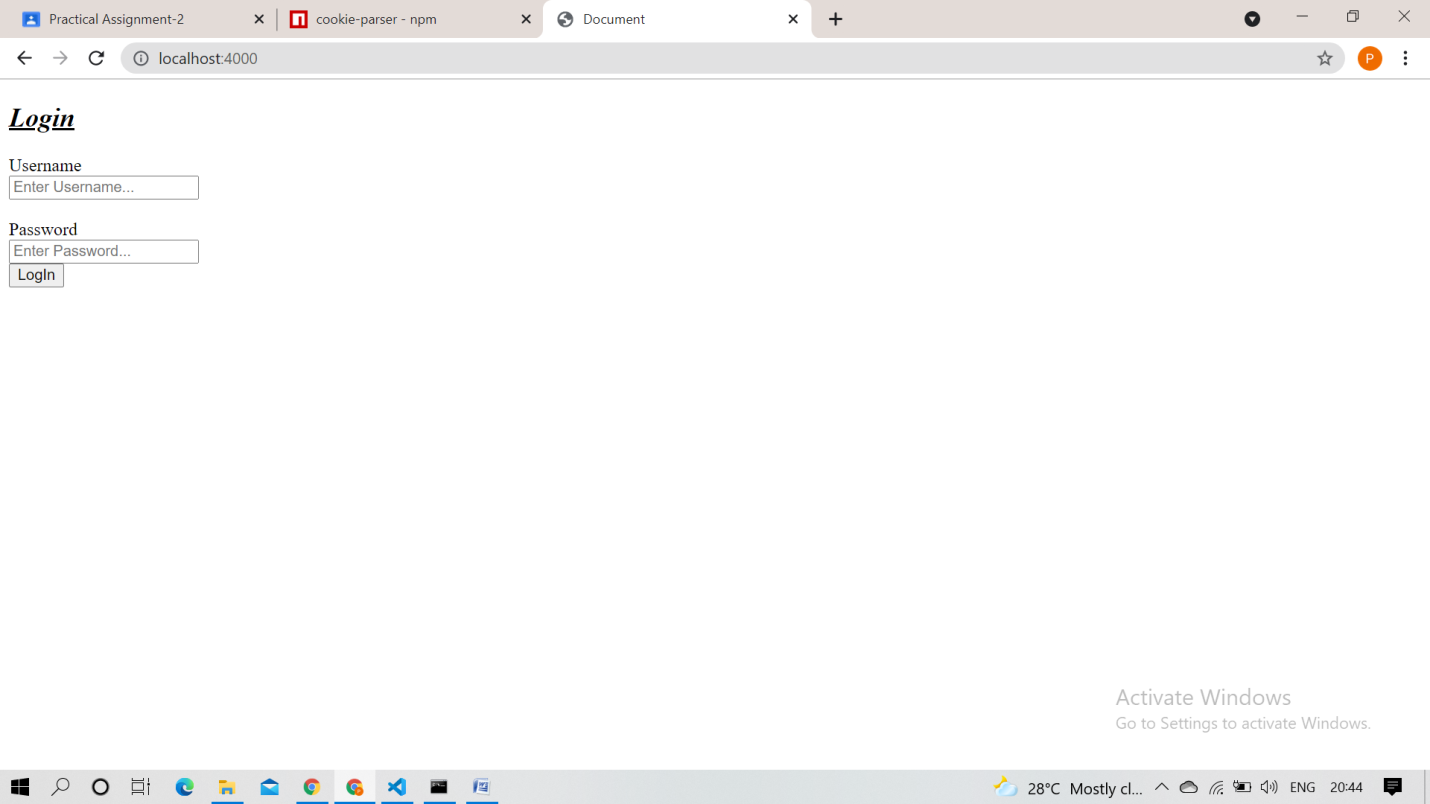
app.get('/logout',(req,res) => {

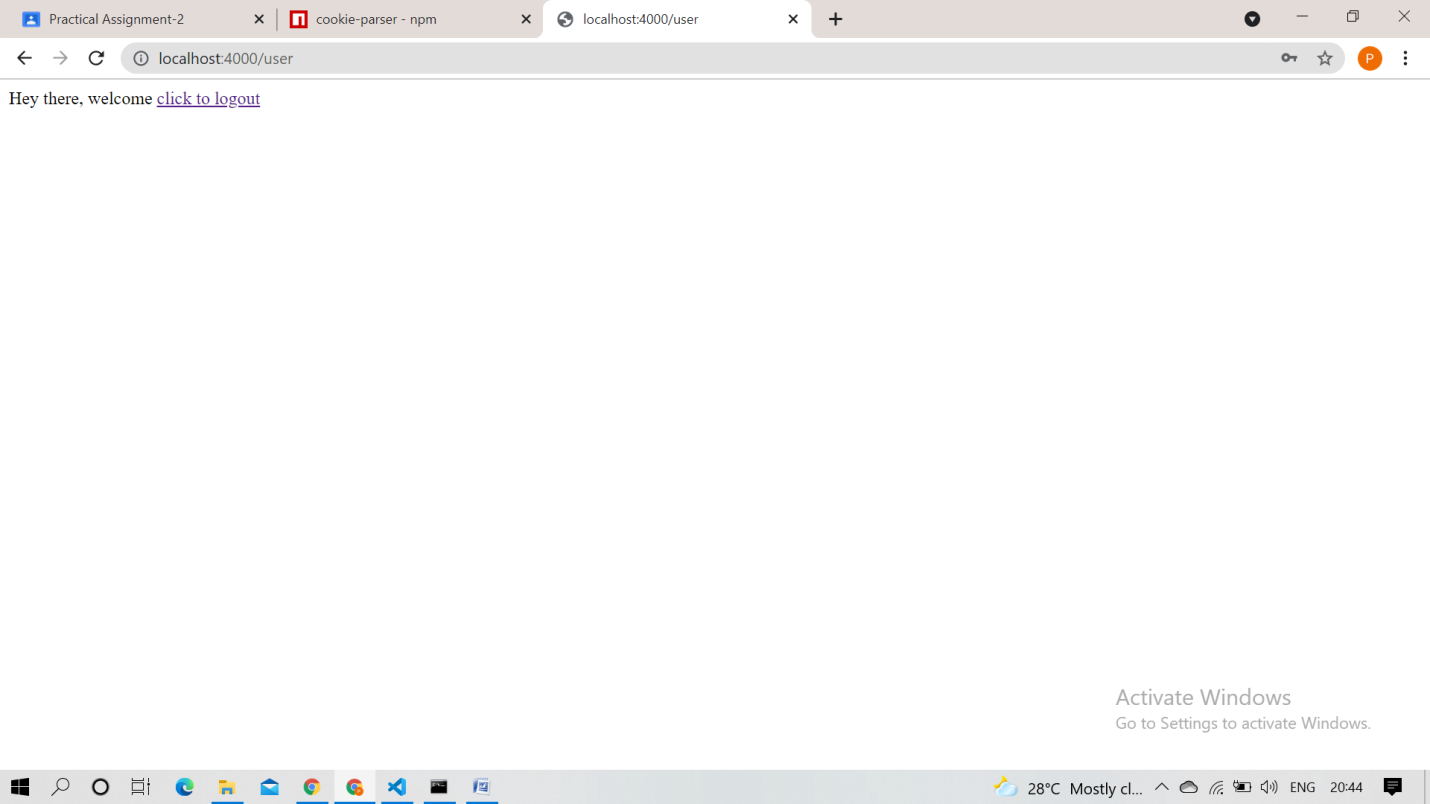
    req.session.destroy();

    res.redirect('/');

});

app.listen(PORT, () => console.log(`Server Running at port ${PORT}`));





3) Login with JWT, CRUD operations for students table with mongoose, express and any one  
  
template engine, Logout.

require("dotenv").config();

require("./config/db").connect();

const Student = require("./model/student");

const express = require("express");

const bcrypt = require("bcryptjs");

const jwt = require("jsonwebtoken");

var cookieParser = require("cookie-parser");

const { verify } = require("./middleware/auth");

const app = express();

app.use(cookieParser());

app.use(express.json());

app.use(express.urlencoded({ extended: true }));

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

  res.sendFile(\_\_dirname + "/views/login.html");

});

app.post("/login", async (req, res) => {

  try {

    console.log(req.body);

    const data = await Student.findOne({ email: req.body.email });

    if (data) {

      console.log(data.password);

      const auth = await bcrypt.compare(req.body.password, data.password);

      if (auth) {

        let \_token = jwt.sign(

          { email: data.email },

          process.env.ACCESS\_TOKEN\_SECRET,

          {

            expiresIn: 86400,

          }

        );

        res.cookie("jwt", \_token, { secure: true, httpOnly: true });

        res.send("You are authenticated");

      } else {

        res.send("invalid Data");

      }

    } else {

      res.send("invalid data");

    }

  } catch (err) {

    console.log(err);

  }

});

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

  res.clearCookie("jwt");

  res.redirect("/display");

});

app.listen(5000, () => {

  console.log("Running");

});

<html>

  <body>

    <h1>Login</h1>

    <form action="/login" method="POST">

      <label for="">Email</label>

      <input type="text" name="email" /><br /><br />

      <label for="">Password</label>

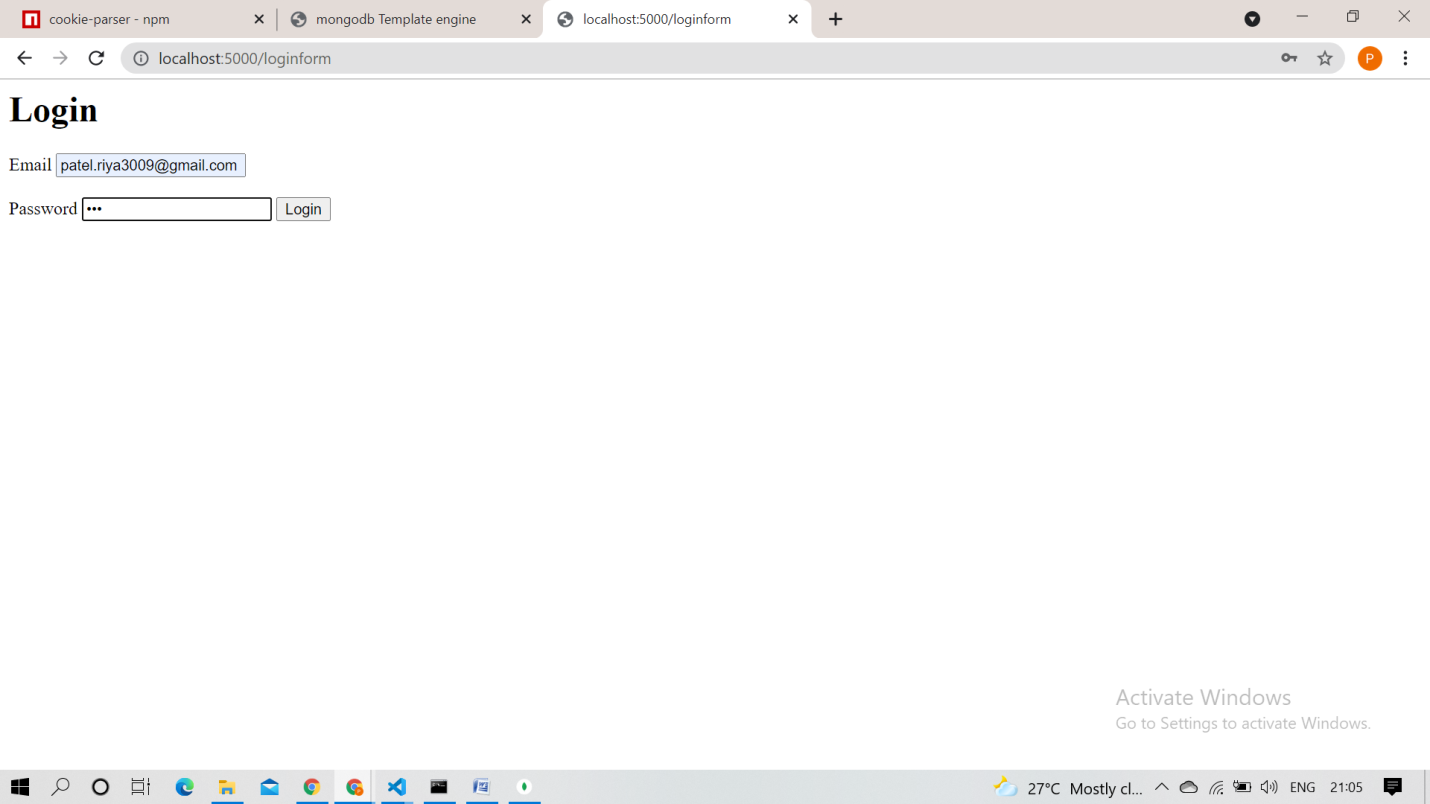
      <input type="password" name="password" />

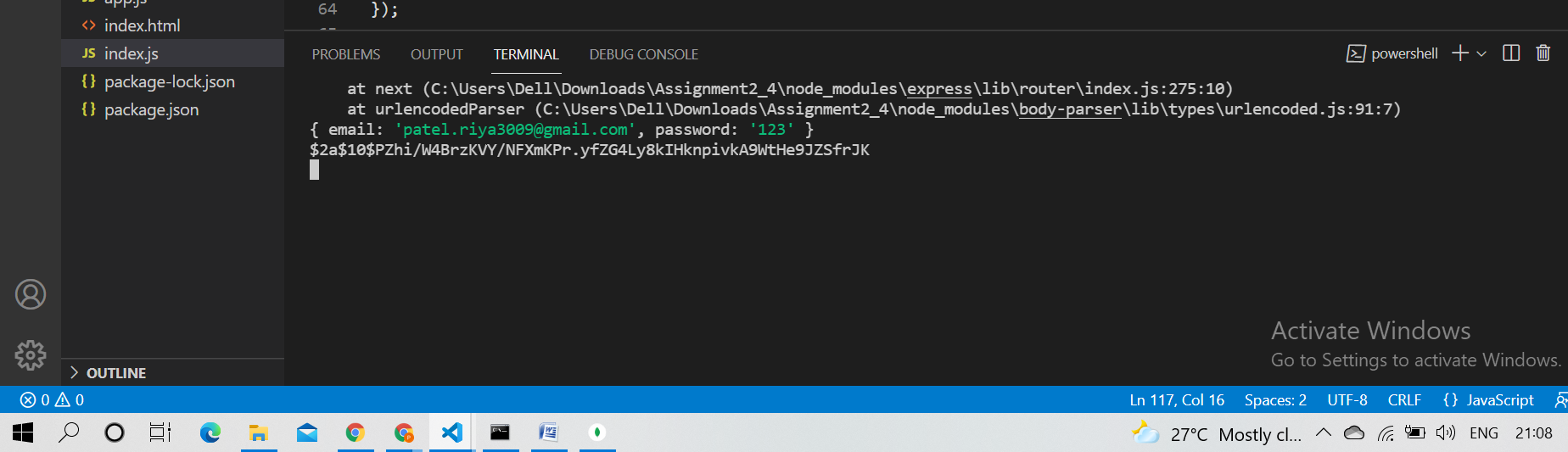
      <input type="submit" value="Login" />

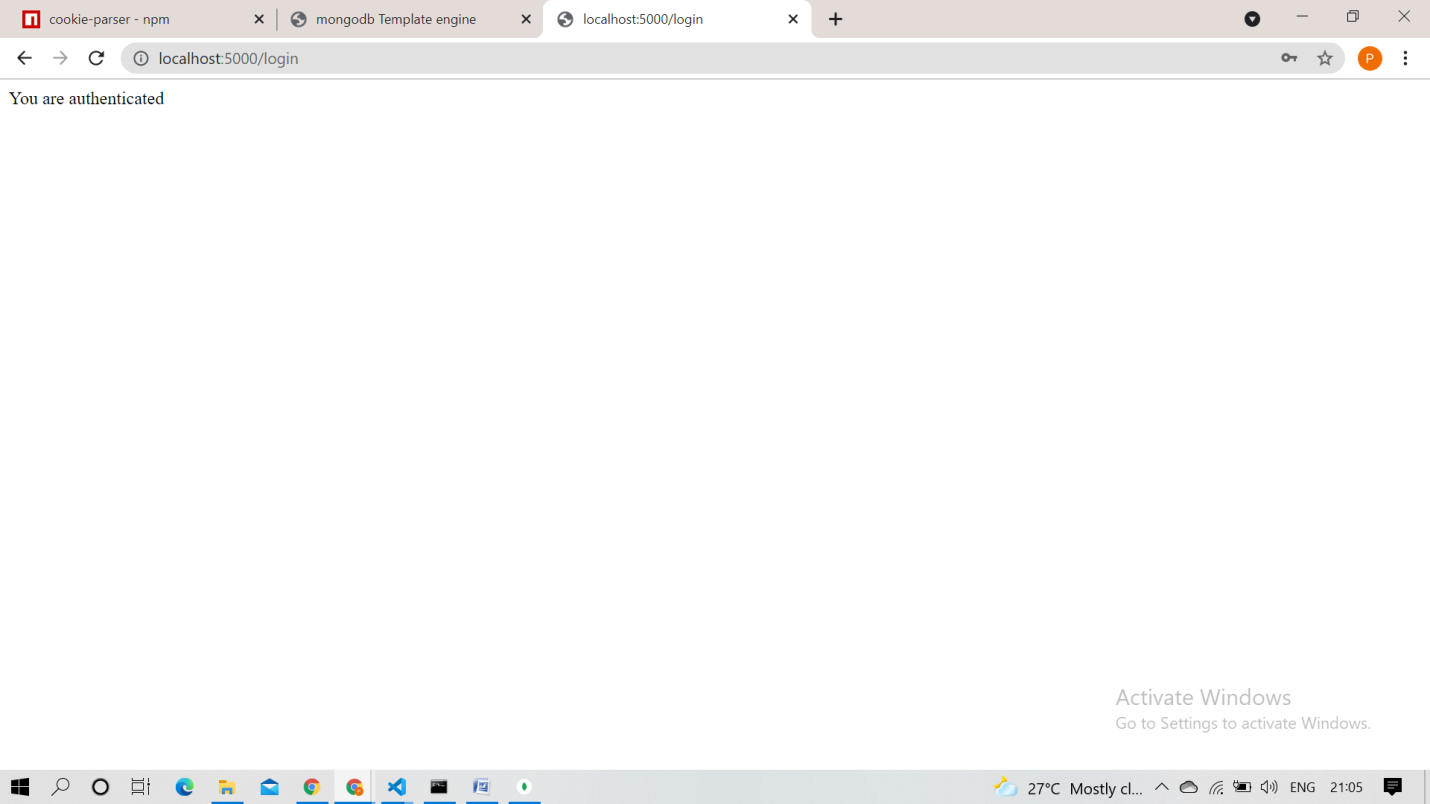
    </form>

  </body>

</html>







require('./models/db');

const express =require('express');

const app=express();

const path=require('path');

const exphbs=require('express-handlebars');

const multer=require('multer');

const studentContoller =require('./controller/studentController');

const bodyParser = require('body-parser');

var storage =multer.diskStorage({

    destination :(req,file,cb)=>{

        cb(null,"public/upload");

    },

    filename:(req,file,cb)=>{

        cb(null,file.fieldname+ "\_"+Date.now()+path.extname(file.originalname))

    }

})

app.use(bodyParser.urlencoded({

    extended:true

}));

app.use(multer({

    storage:storage

}).single('file'));

app.use(bodyParser.json());

app.set('views',path.join(\_\_dirname,'/views/'));

app.use('/public',express.static(path.join(\_\_dirname,'public')));

app.engine('hbs',exphbs({

    extname :'hbs',

    defaultLayout:'mainLayout',

    layoutsDir:\_\_dirname +'/views/layouts/'

}));

app.set('view engine','hbs');

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

    req.logout();

    res.redirect("/");

});

function isLoggedIn(req,res,next) {

    if(req.isAuthenticated()){

        return next();

    }

    res.redirect("/login");

}

app.listen(3000,()=>{

    console.log('ecpress server started at port : 3000');

});

app.use('/student',studentContoller);

<!DOCTYPE html>

<html>

    <head>

        <title>mongodb Template engine</title>

    </head>

    <body class="bg-info">

        <div class="row">

            <div class="col-md-6 offset-md-3" style="background-color:rgb(229, 233, 238);padding: 20px;">

                {{{body}}}

            </div>

        </div>

    </body>

</html>

<h3>{{viewTitle}}</h3>

<form action="/student" method="POST" enctype="multipart/form-data">

<table>

    <tr>

        <input type="hidden" name="\_id" value="{{student.\_id}}">

        <th><label>Roll no :</label></th>

        <td><input type="text" class="form-control" name="rollno" placeholder="roll no" value="{{student.rollno}}"></td>

        {{!-- <div class="text-danger">{{student.rollnoError}}</div> --}}

    </tr>

    <tr>

        <th><label>Name :</label></th>

        <td><input type="text" class="form-control" name="name" placeholder="name" value="{{student.name}}"></td>

    </tr>

    <tr>

        <th><label>City :</label></th>

        <td>

            <input type="text" class="form-control" name="city" placeholder="city" value="{{student.city}}">

        </td>

    </tr>

    <tr>

        <th><label>Department :</label></th>

        <td><input type="text" class="form-control" name="department" placeholder="department name" value="{{student.department}}"></td>

    </tr>

    <tr>

        <th>Image :</th>

        <td><input type="file" name="file" value=""></td>

    </tr>

    <tr>

        <th></th>

        <td><button type="submit" class="btn btn-info">Submit</button>

        <button type="button"><a  href="/student/list">View All</a></button></td>

    </tr>

</table>

</form>

<h3><button type="button"><a href="/student"> Create New</a> </button>students List</h3>

<table class="table table-striped" border="1">

    <thead>

        <tr>

            <th>Roll no</th>

            <th>Name</th>

            <th>City</th>

            <th>Department</th>

            <th>Image</th>

            <th>Action</th>

        </tr>

    </thead>

    <tbody>

        {{#each list}}

        <tr>

            <td>{{this.rollno}}</td>

            <td>{{this.name}}</td>

            <td>{{this.city}}</td>

            <td>{{this.department}}</td>

            <td><img src="/{{this.image}}" height="100px" width="100px"></td>

            <td>

                <a href="/student/{{this.\_id}}">Edit</a>

                <a href="/student/delete/{{this.\_id}}" onclick="return confirm('Are u sure u want to delete record ?');">

                Delete</a>

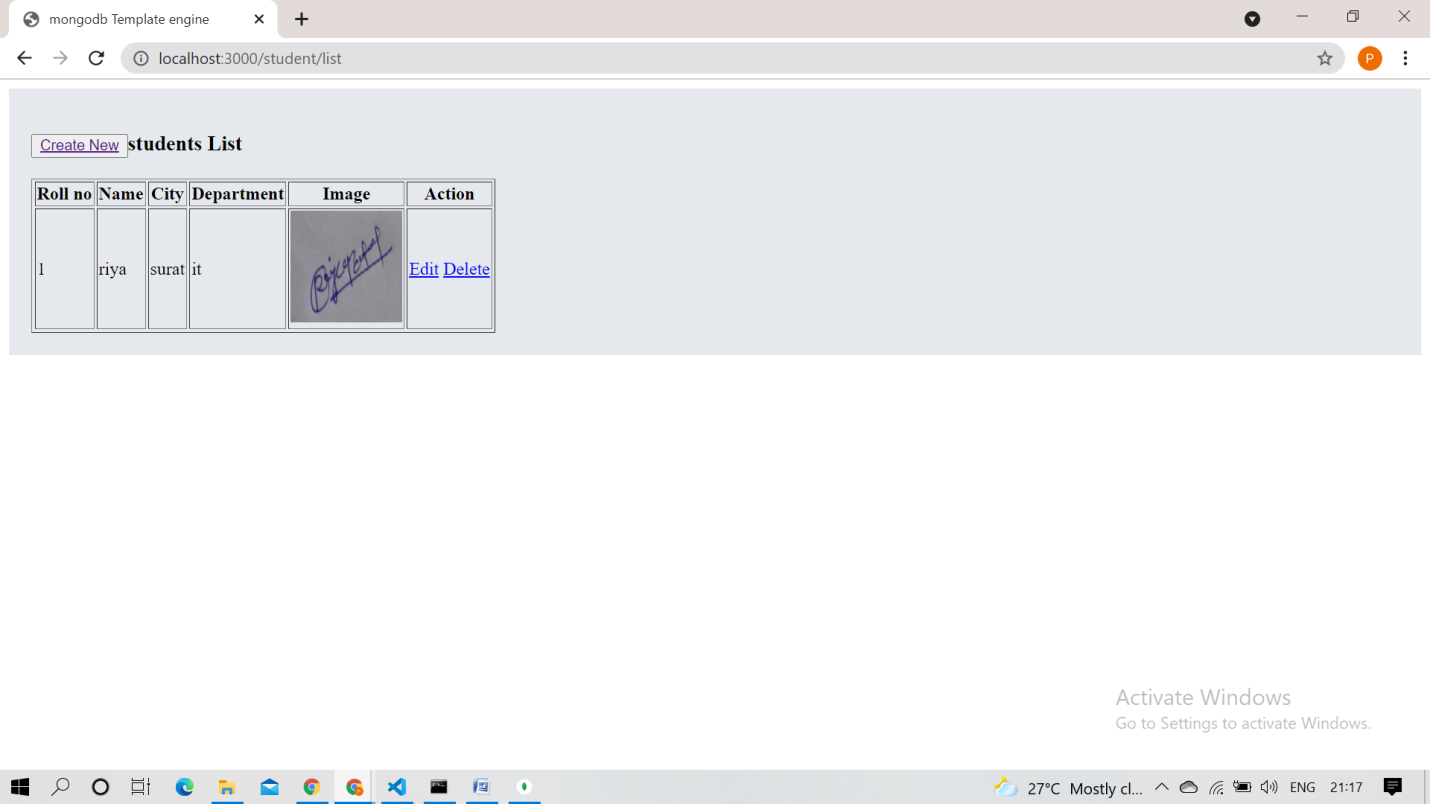
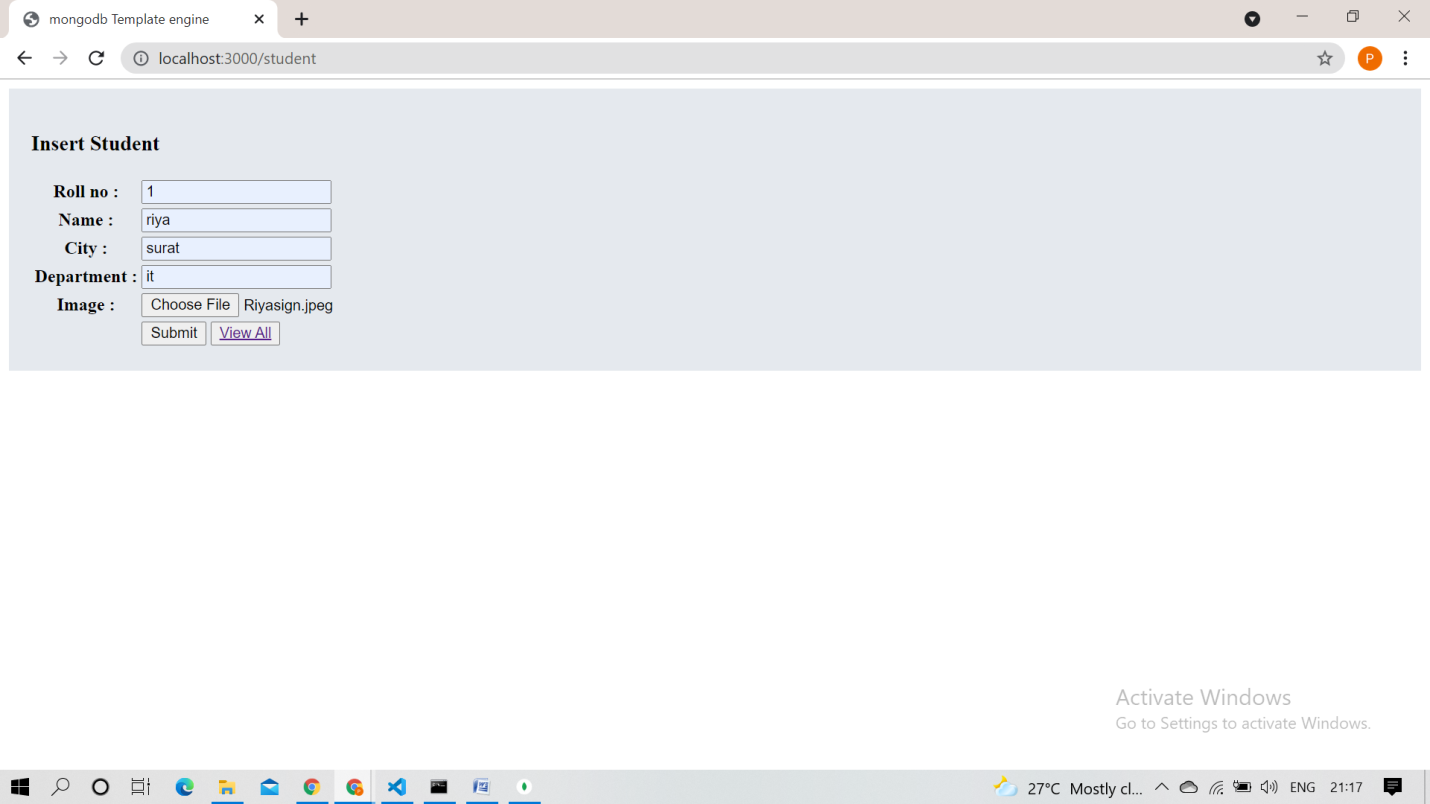
            </td>

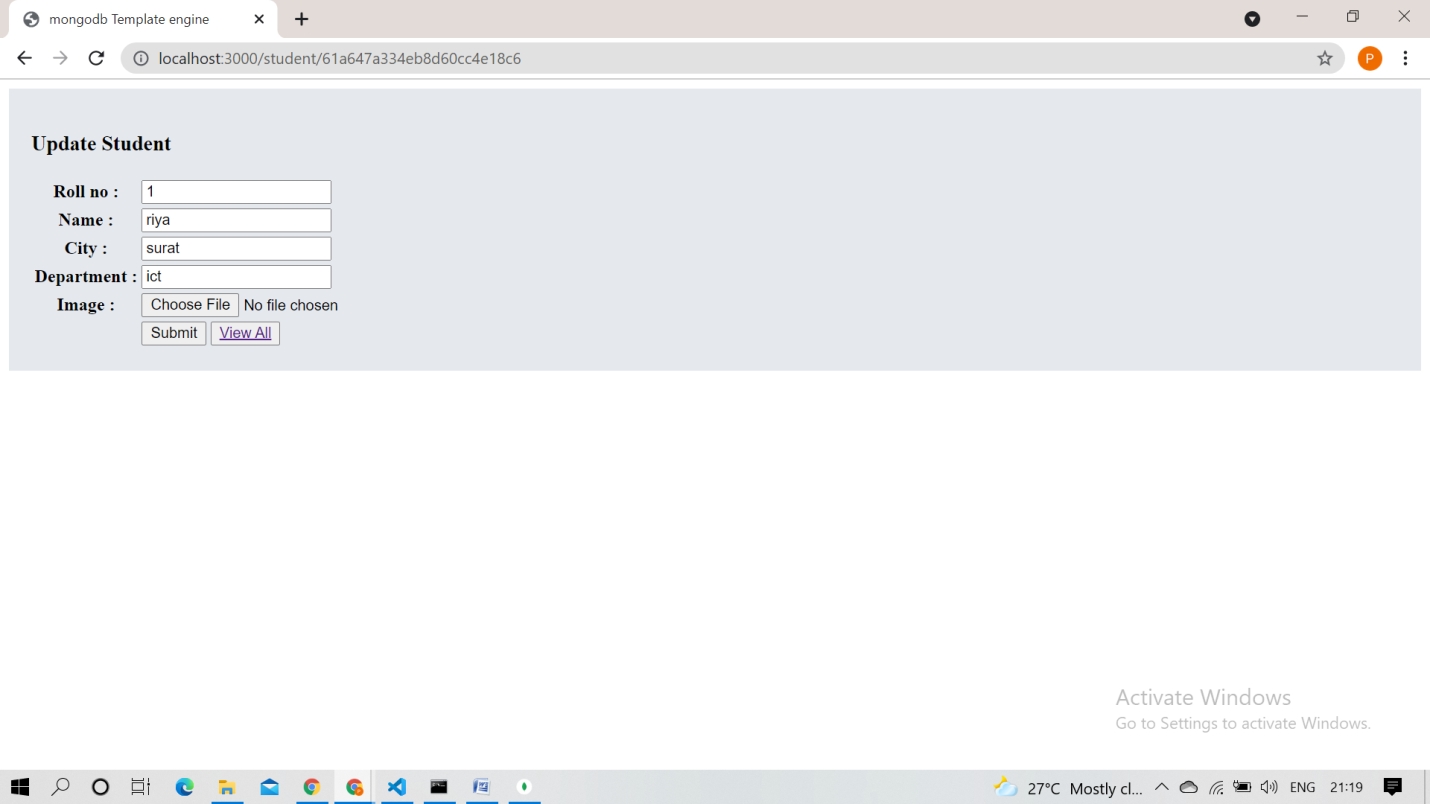
        </tr>

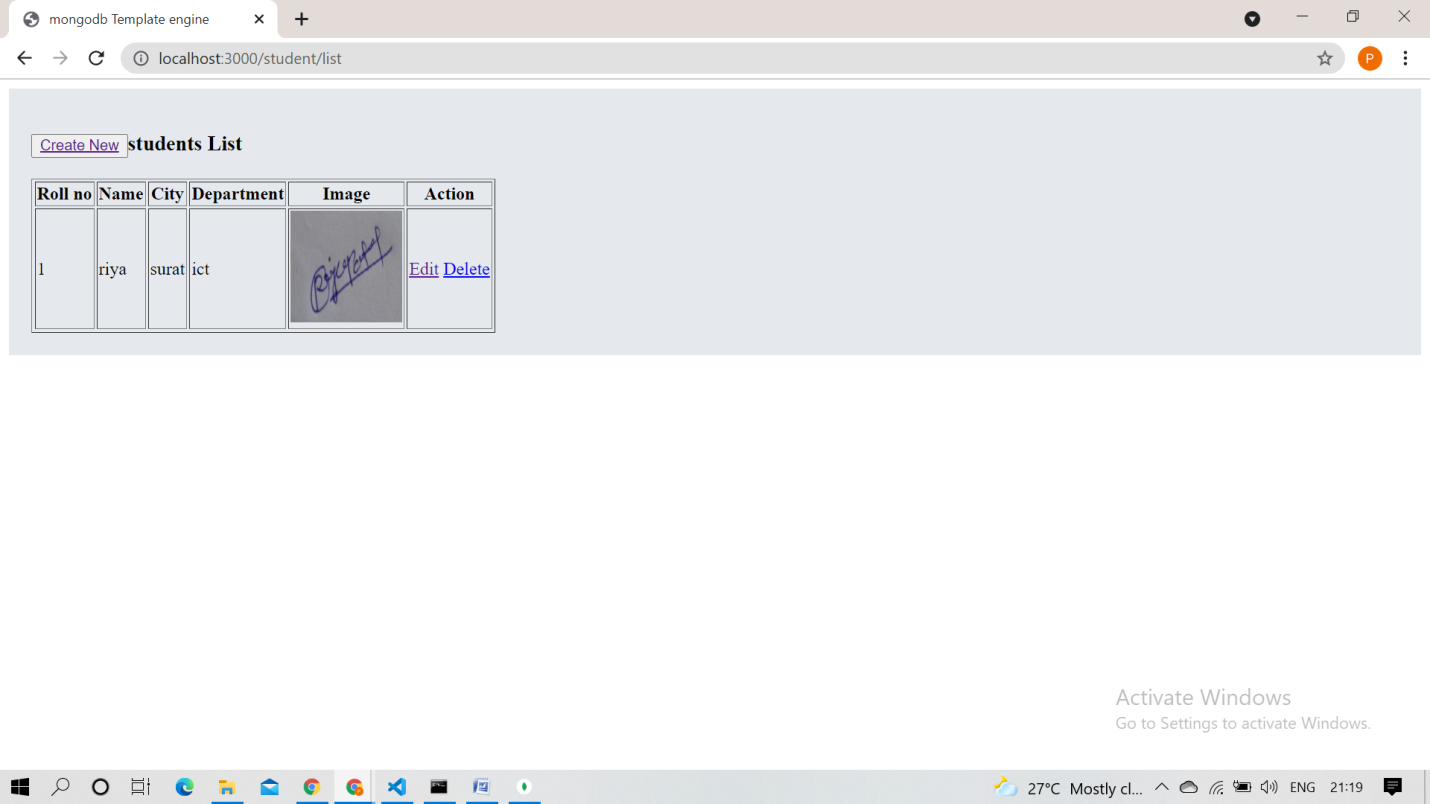
        {{/each}}

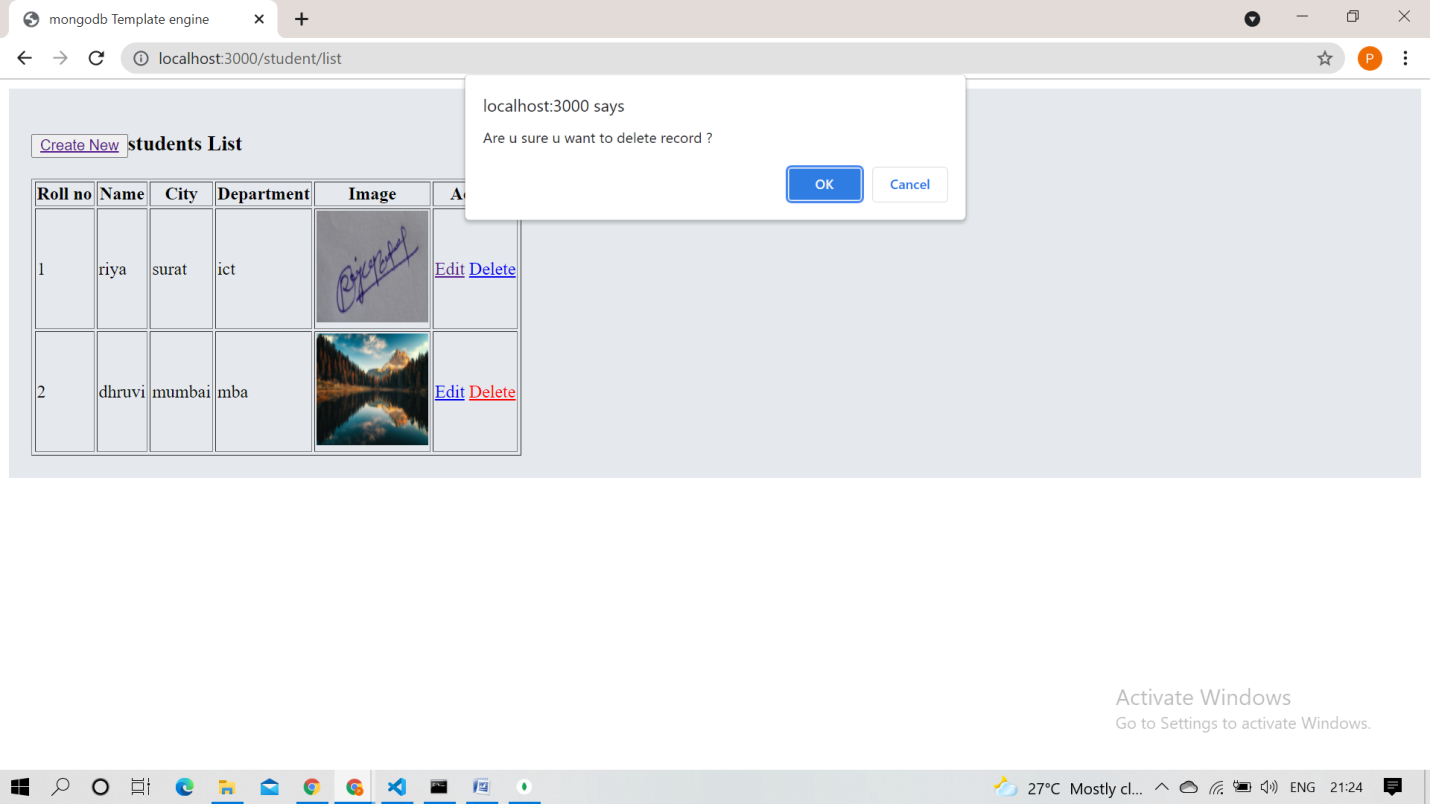
    </tbody>

</table>









4) Login with JWT, CRUD operations for students table with mongoose, express and  
frontend(html,css,javascript/jquery/angularjs), Logout.

require("dotenv").config();

require("./config/db").connect();

const Student = require("./model/student");

const express = require("express");

const bcrypt = require("bcryptjs");

const jwt = require("jsonwebtoken");

var cookieParser = require("cookie-parser");

const { verify } = require("./middleware/auth");

const app = express();

app.use(cookieParser());

app.use(express.json());

app.use(express.urlencoded({ extended: true }));

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

  res.send("working");

});

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

  res.sendFile(\_\_dirname + "/views/create.html");

});

app.post("/store", (req, res) => {

  console.log(req.body);

  req.body.password = bcrypt.hashSync(req.body.password);

  Student.insertMany(req.body)

    .then((result) => {

      res.redirect("/display");

    })

    .catch((error) => console.log(error));

});

app.get("/display", verify, (req, res, next) => {

  Student.find({}, (err, data) => {

    if (err) {

      console.log(err);

    } else {

      res.sendFile(\_\_dirname + "/views/display.html");

    }

  });

});

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

  Student.find({}, (err, data) => {

    if (err) console.log(err);

    res.json({ data: data });

  });

});

app.get("/delete/:id", function (req, res) {

  Student.findByIdAndRemove(req.params.id, function (err, data) {

    if (err) console.log(err);

    else res.redirect("/display");

  });

});

app.get("/edit/:id", (req, res) => {

  res.sendFile(\_\_dirname + "/views/edit.html");

});

app.post("/editstudent", (req, res) => {

  Student.findOne({ \_id: req.body.\_id }, (err, data) => {

    res.json({ data: data });

    console.log(req.body);

  });

});

app.post("/edit", (req, res) => {

  console.log(req.body);

  Student.updateOne(

    { \_id: req.body.\_id },

    {

      name: req.body.name,

      email: req.body.email,

      address: req.body.address,

    },

    function (err, data) {

      if (err) console.log(err);

      else res.redirect("/display");

    }

  );

});

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

  res.sendFile(\_\_dirname + "/views/login.html");

});

app.post("/login", async (req, res) => {

  try {

    console.log(req.body);

    const data = await Student.findOne({ email: req.body.email });

    if (data) {

      console.log(data.password);

      const auth = await bcrypt.compare(req.body.password, data.password);

      if (auth) {

        let \_token = jwt.sign(

          { email: data.email },

          process.env.ACCESS\_TOKEN\_SECRET,

          {

            expiresIn: 86400,

          }

        );

        res.cookie("jwt", \_token, { secure: true, httpOnly: true });

        res.send("You are authenticated");

      } else {

        res.send("invalid Data");

      }

    } else {

      res.send("invalid data");

    }

  } catch (err) {

    console.log(err);

  }

});

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

  res.clearCookie("jwt");

  res.redirect("/display");

});

app.listen(5000, () => {

  console.log("Running");

});

<html>

  <body>

    <h1>Login</h1>

    <form action="/login" method="POST">

      <label for="">Email</label>

      <input type="text" name="email" /><br /><br />

      <label for="">Password</label>

      <input type="password" name="password" />

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

    </form>

  </body>

</html>

<html>

  <body>

    <h1>Enter Student Record</h1>

    <form action="/store" method="post">

       Name : <input type="text" name="name" /> <br /><br />

      Email :<input type="text" name="email" /> <br /><br />

      Password : <input type="password" name="password" /> <br /><br />

      Address  : <input type="text" name="address" /> <br /><br />

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

    </form>

  </body>

</html>

<html>

  <body>

    <table border="1" >

      <thead>

        <th>Name</th>

        <th>Email</th>

        <th>Password</th>.

        <th>Address</th>

        <th>Edit</th>

        <th>Delete</th>

      </thead>

      <tbody id="studentdata"></tbody>

    </table>

    <a href="/logout">Logout</a>

    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

    <script>

      $(document).ready(async function () {

        try {

          var res = await fetch("/studentdata", {

            method: "GET",

            headers: { "Content-Type": "application/json" },

          });

          var data = await res.json();

          var value = "";

          data.data.forEach((element) => {

            value += ` <tr>

                    <td >${element.name}</td>

                    <td >${element.email}</td>

                    <td >${element.password}</td>

                    <td >${element.address}</td>

                    <td ><a href="/edit/${element.\_id}">Edit</a></td>

                    <td ><a href="/delete/${element.\_id}">Delete</a></td>

                    </tr>`;

          });

          $("#studentdata").append(value);

        } catch (err) {

          console.log(err);

        }

      });

    </script>

  </body>

</html>

<html>

  <body>

    <h3>Edit Student</h3>

    <form action="/edit" method="post">

      <input type="hidden" name="\_id" id="\_id" value="" />

       Name :

      <input type="text" name="name" id="name" value="" />

      <br /><br />

      Email :<input type="text" name="email" id="email" value="" /> <br /><br />

      Address :<input type="text" name="address" id="addresss" value="" />

      <br /><br />

      <input type="submit" value="Edit" /><br/><br/>

    </form>

    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

    <script>

      let param = window.location.href.split("/");

      var \_id = param.slice(-1)[0];

      $(document).ready(async function () {

        var res = await fetch("/editstudent", {

          method: "POST",

          headers: {

            "Content-Type": "application/json",

            Accept: "application/json",

          },

          body: JSON.stringify({ \_id: \_id }),

        });

        var data = await res.json();

        $("#\_id").val(\_id);

        $("#firstname").val(data.data.firstname);

        $("#lastname").val(data.data.lastname);

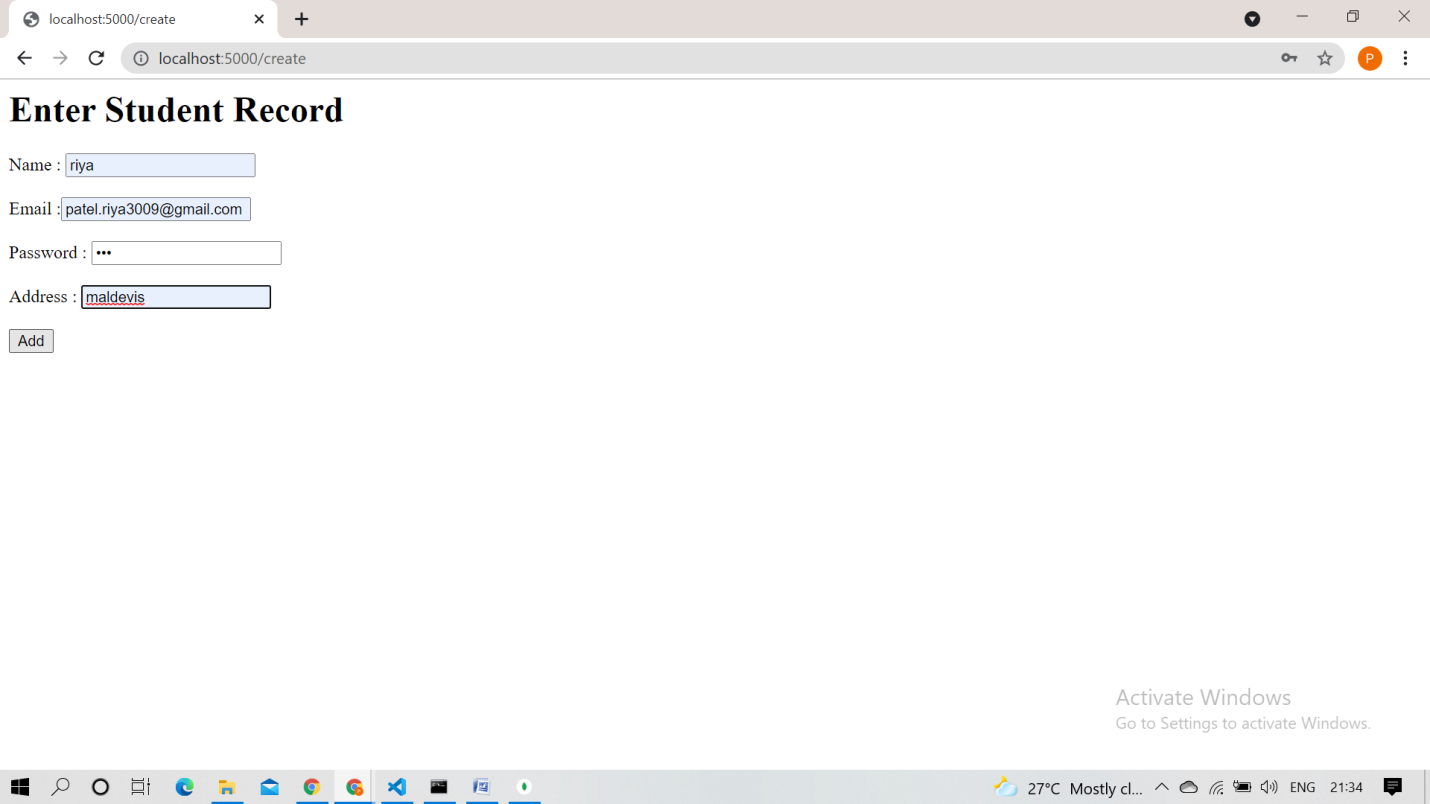
        $("#email").val(data.data.email);

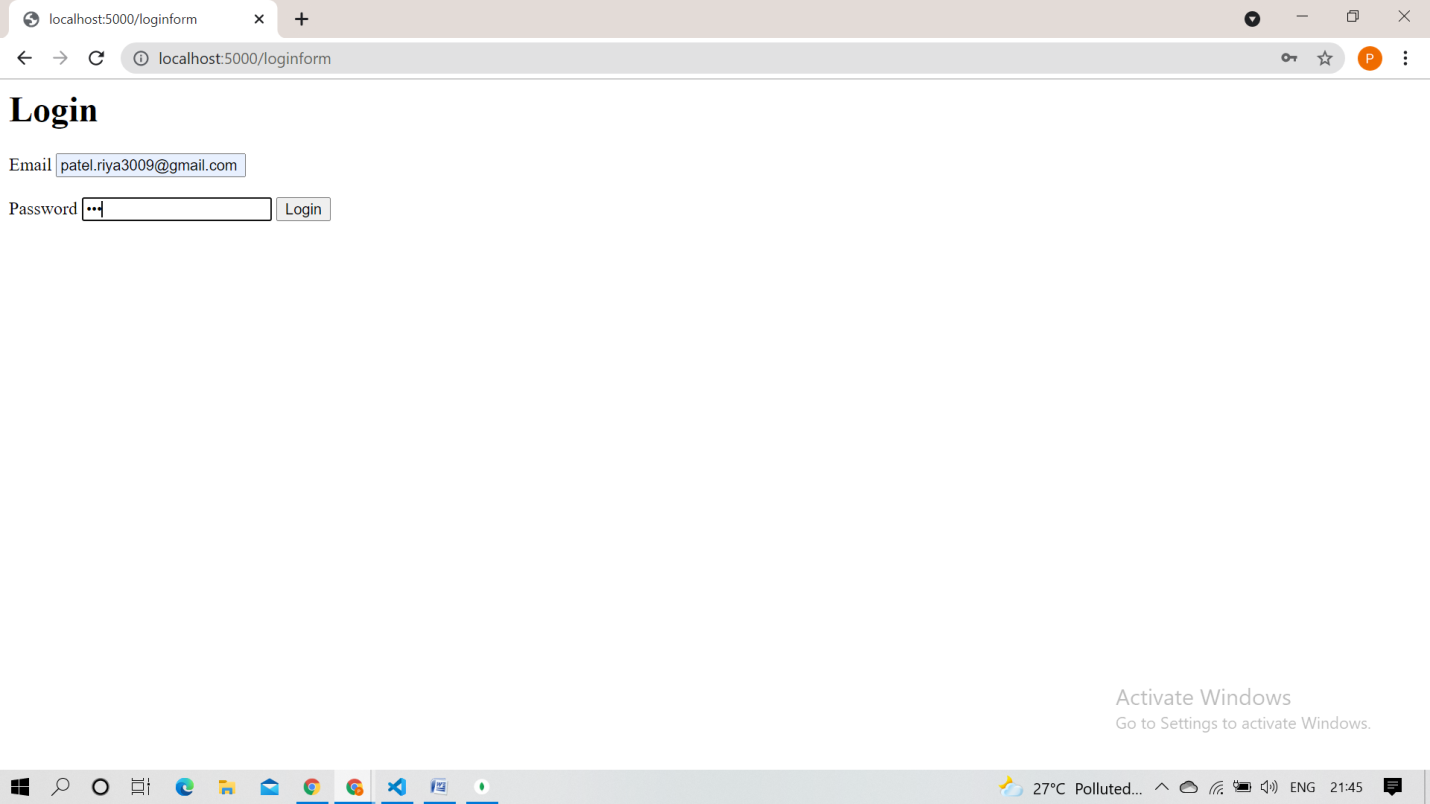
      });

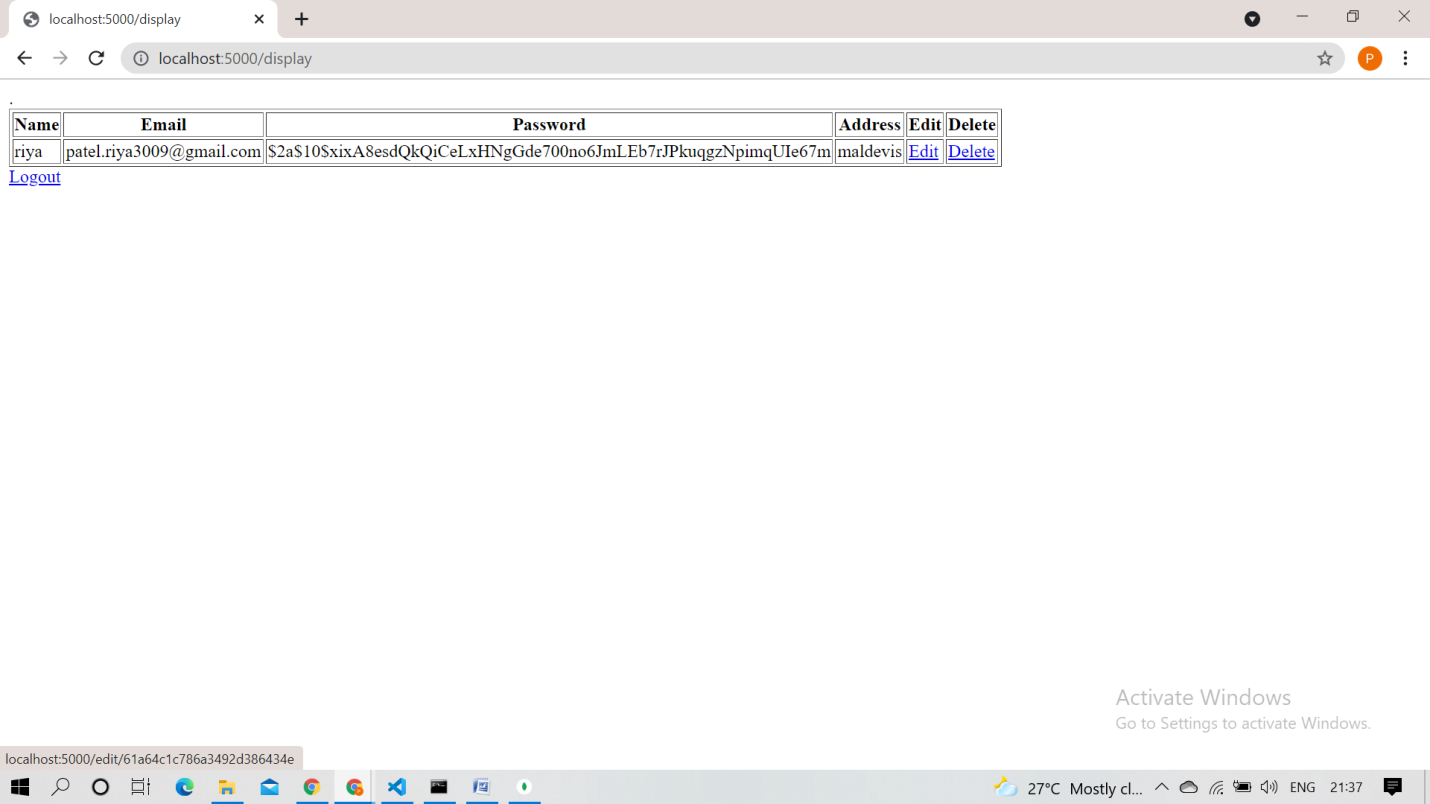
    </script>

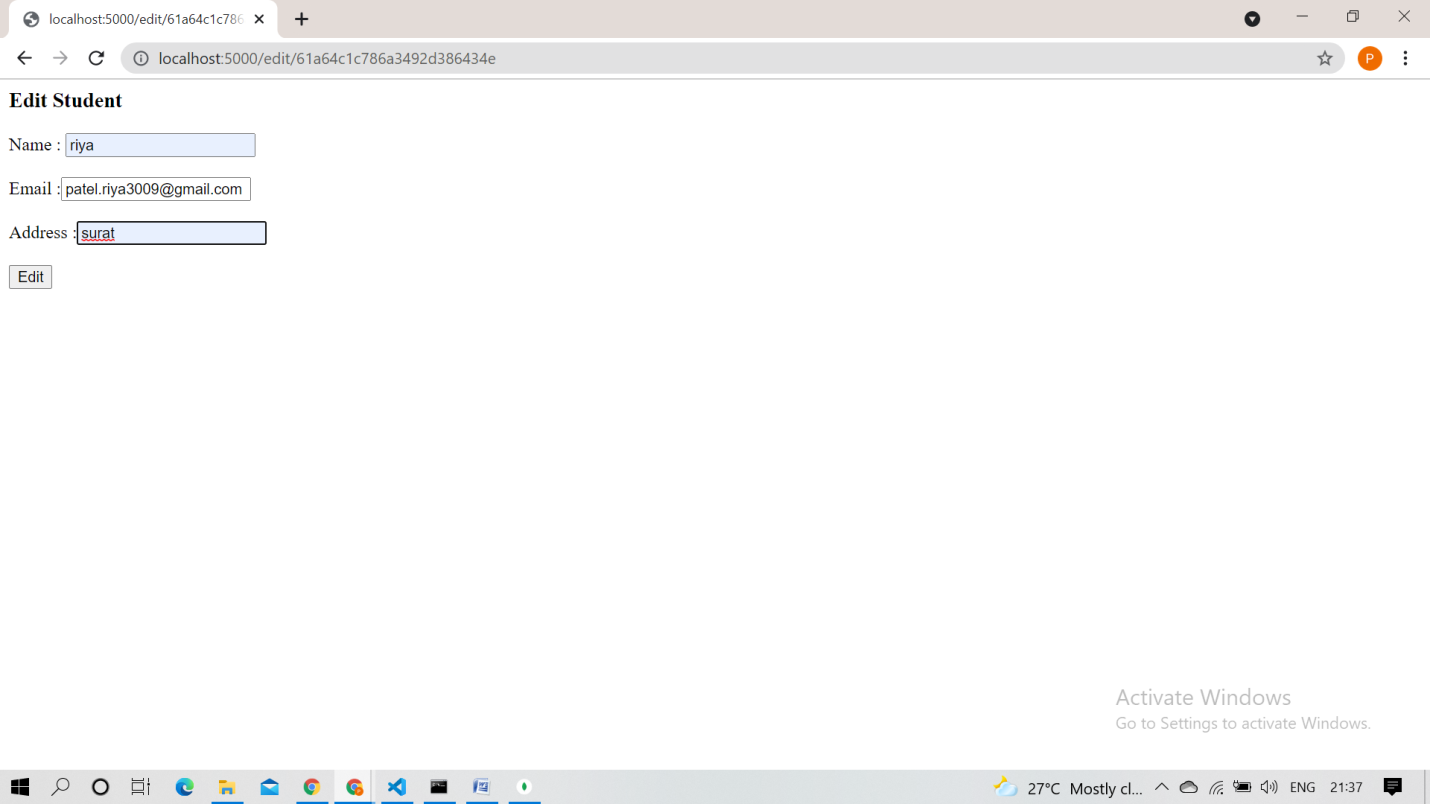
  </body>

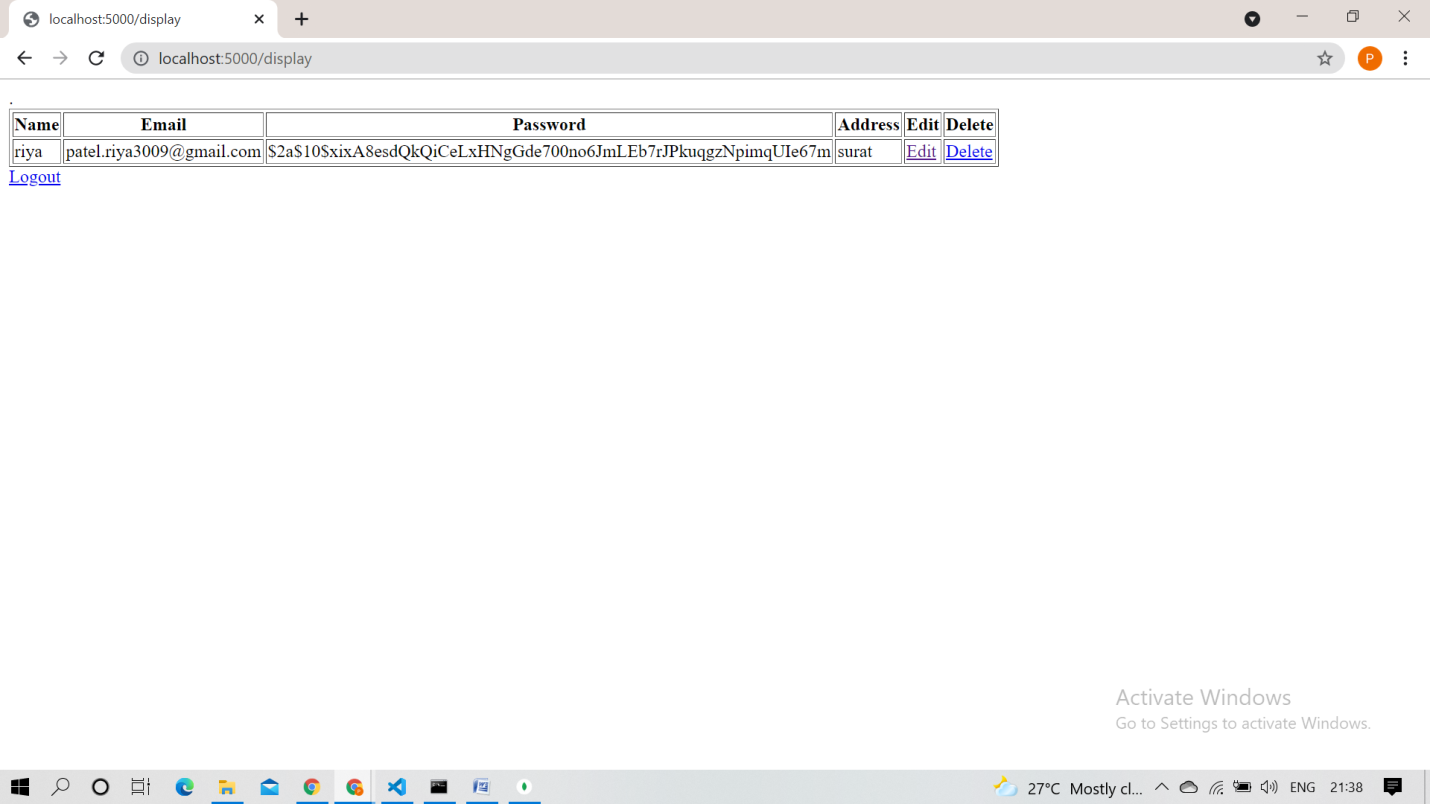
</html>



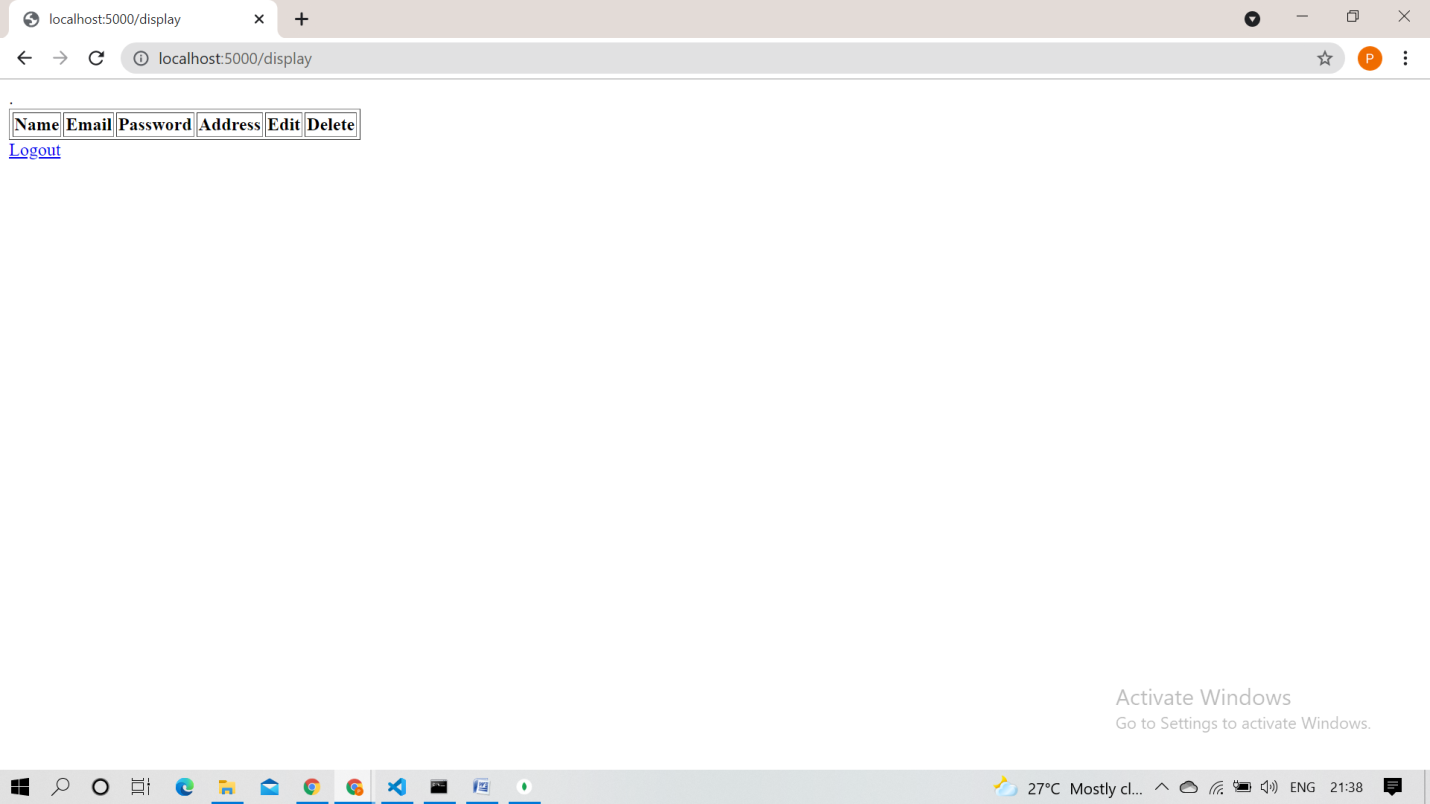








Delete:



Logout

