Name : Raj Patel

Roll No : 21

Sub : Full Stack Web development 701

Msc IT 7th Sem

Github Repo Link : https://github.com/raj332/Assignment-2

Practical Assignment 2

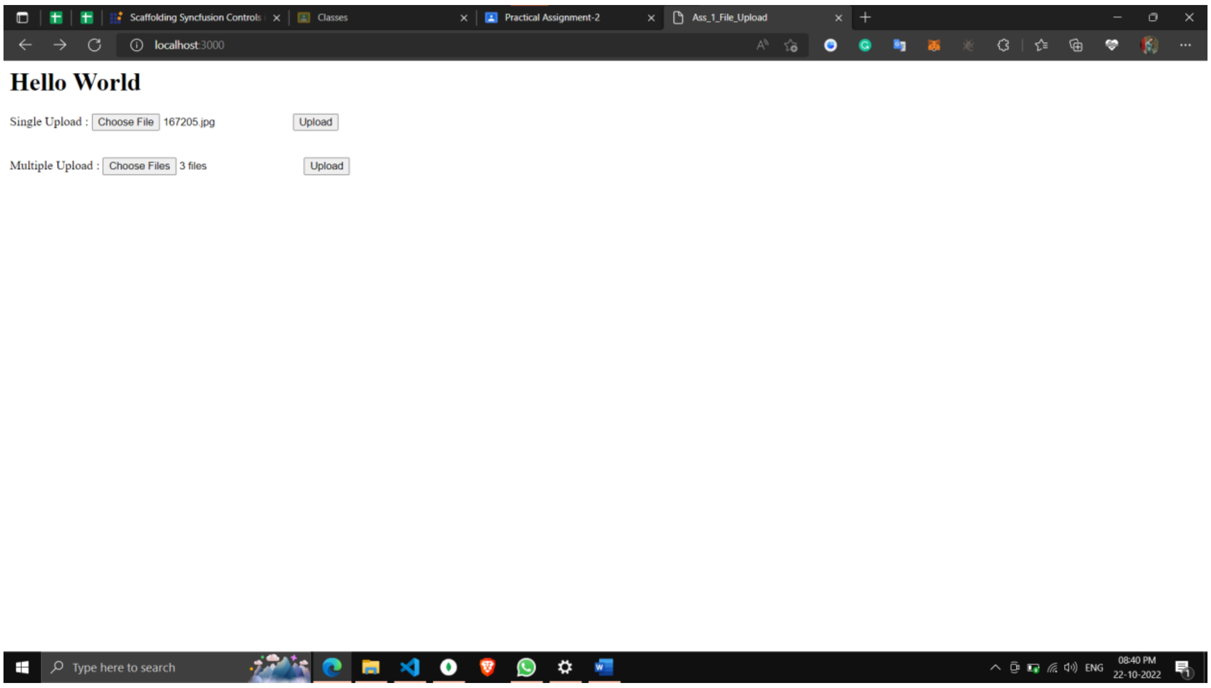
Q-1. Express File upload (single, multiple) with validations.

Index.js File

```
const express =require ('express')
const multer =require('multer')
const app= express();
app.set('view engine','ejs');
let options =multer.diskStorage({
    destination:(req,file,cb)=>{
        console.log(file.mimetype)
      if(file.mimetype !=='image/jpeg'){
        return cb('Invalid Format')
      }
      cb(null,'./imgs');
    },
    filename:(req,file,cb)=>{
      cb(null,Date.now()+file.originalname)
    }
})
let upload =multer({
    storage:options
})

app.get('/',(req,res)=>{
    res.render("index");
})
app.post('/upload',upload.single("images"),(req,res)=>{
    res.send("File Uploaded");
})
app.post('/uploads',upload.array("multi_images",2),(req,res)=>{
    res.send("Multi File Uploaded");
})
app.use(express.urlencoded({ extended: true }));
app.listen(3000,()=>{
    console.log("Listening 3000")
})
```

Index.ejs (View)

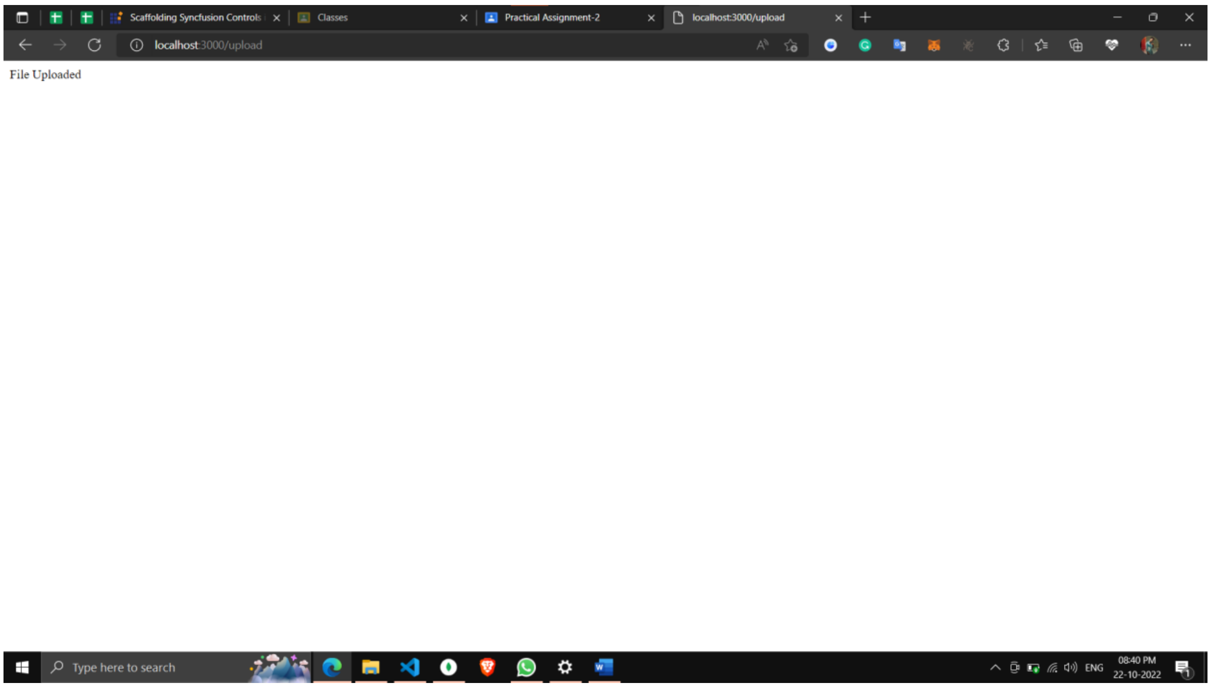
```
<html>
    <head>
        <title>
            Ass_1_File_Upload
        </title>
    </head>
    <body>
        <h1>Hello World</h1>
        <form method="post" action="/upload" enctype="multipart/form-data">
            Single Upload :
            <input type="file" name="images" >
            <button type="submit">Upload</button>
       <br></form><br>
       <form method="post" action="/uploads" enctype="multipart/form-data">
            Multiple Upload :
            <input type="file" multiple name="multi_images" >
            <button type="submit">Upload</button>
        </form>
    </body>
</html>
```



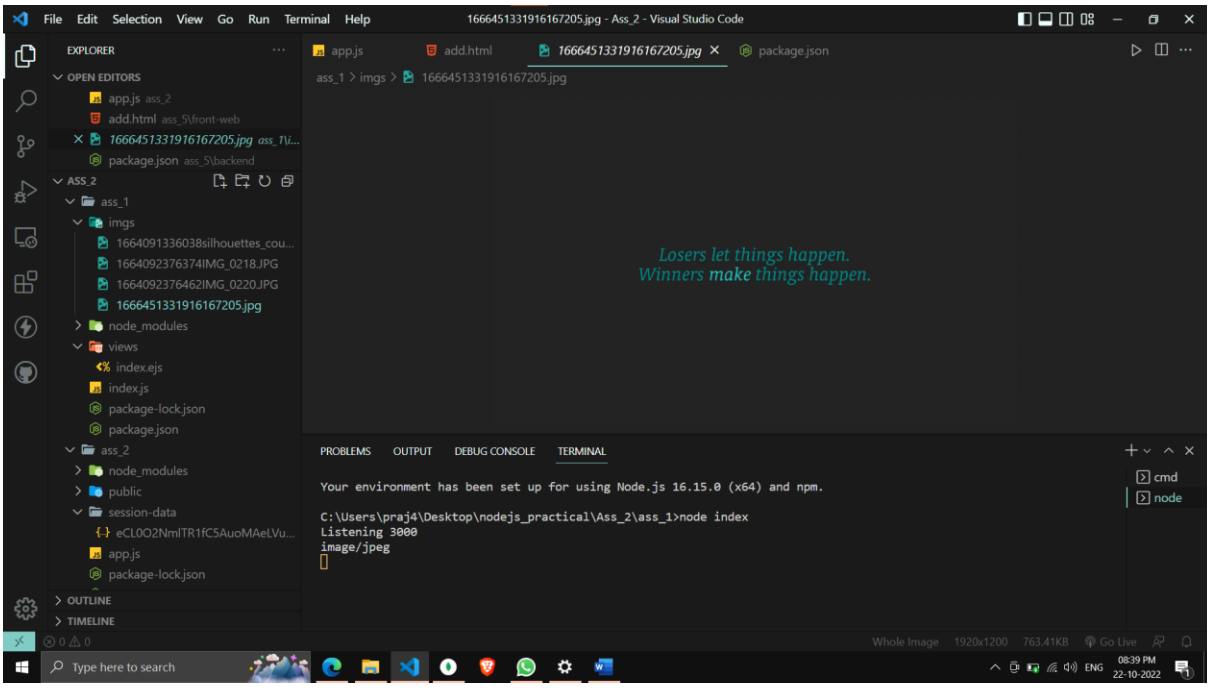
Hello World

Single Upload : Choose File 167205.jpg Upload

Multiple Upload : Choose Files 3 files Upload



File Uploaded

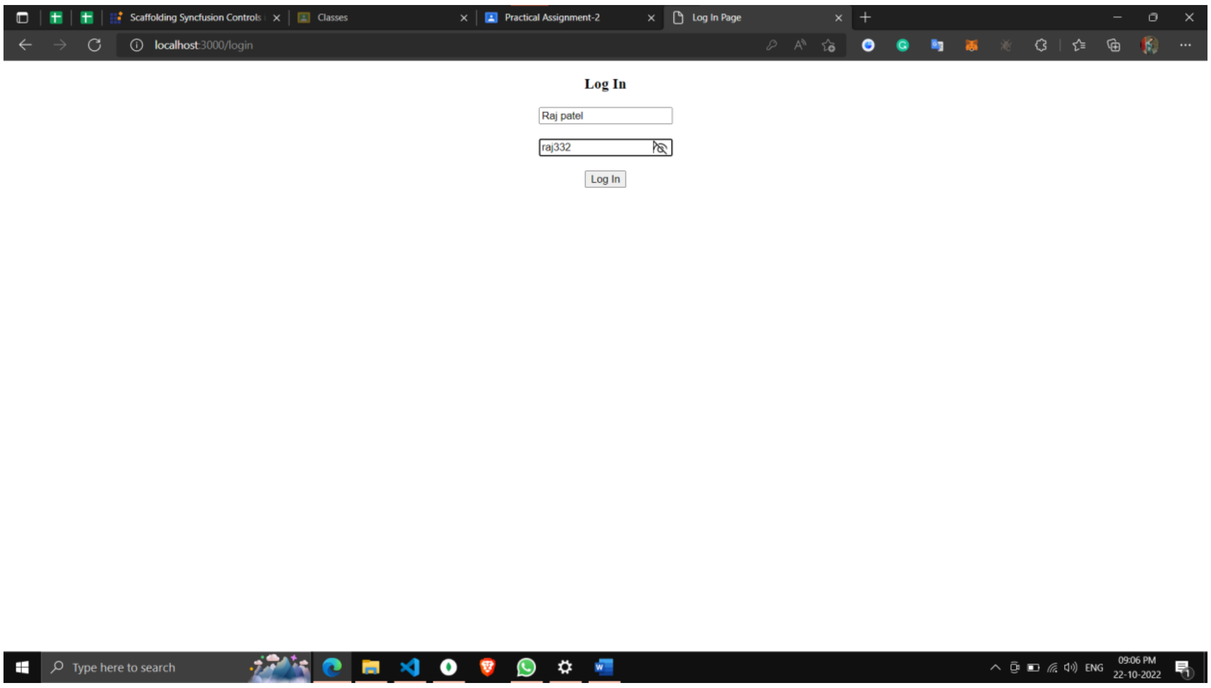

1666451331916167205.jpg - Ass_2 - Visual Studio Code
Losers let things happen.
Winners make things happen.

```
Your environment has been set up for using Node.js 16.15.0 (x64) and npm.

C:\Users\praj4\Desktop\nodejs_practical\Ass_2\ass_1>node index
Listening 3000
image/jpeg
```

2) Express Login application with file session store.

App.js file :

```
const express = require('express')
const app = express()
const session = require('express-session')
const path = require('path')
const FileStore = require('session-file-store')(session)
app.use(express.static('public'))
app.use(express.urlencoded({ extended: false }))
app.use(session({
    secret: 'secret',
    resave: false,
    saveUninitialized: false,
    store: new FileStore({ path: './session-data' })
}))

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

app.post('/login', (req, res) => {
    var username = req.body.username;
    var password = req.body.password;

    if (username && password) {
        req.session.loggedIn = true;
        req.session.username = username
        console.log(req.body)
        return res.redirect('/index')
    } else {
        return res.send('Please enter username and password !!!')
    }
})

app.get('/index', (req, res) => {
    if (req.session.loggedIn) {
        console.log('logged in')
        res.sendFile(__dirname + '/public/index.html')
    } else {
        console.log('not logged in')
        res.sendFile(__dirname + '/public/login.html')

    }
})

app.listen(3000, () => {
    console.log("Running on 3000");
})
```

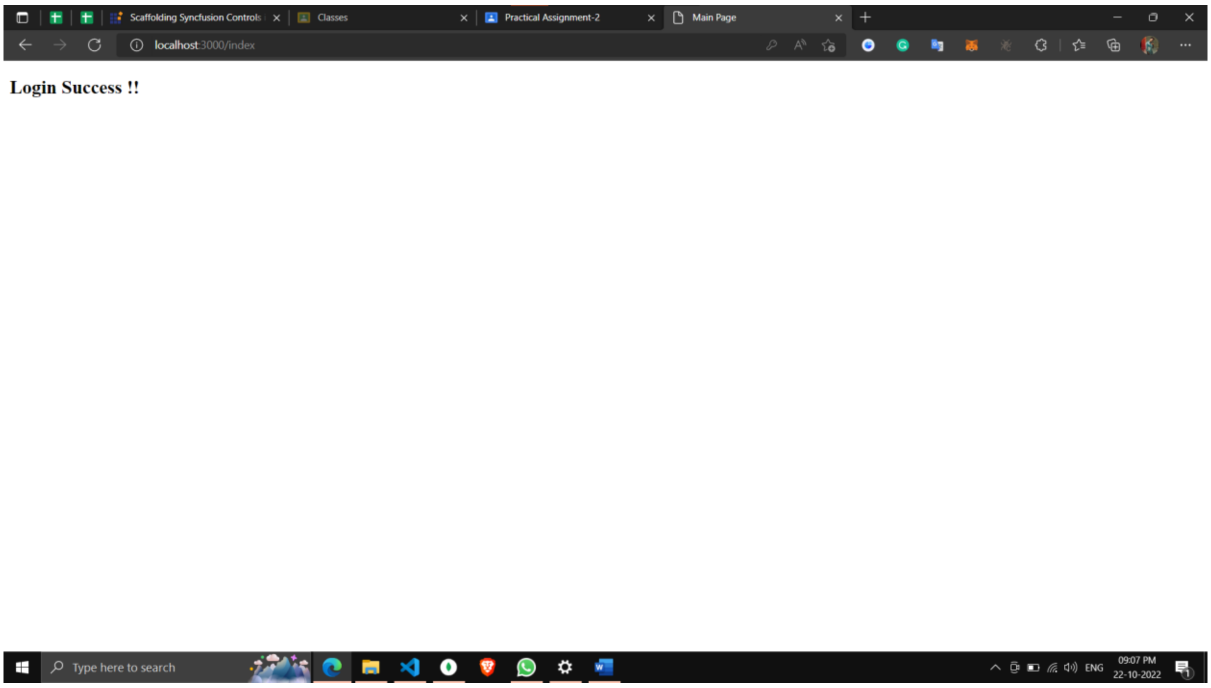


Log In

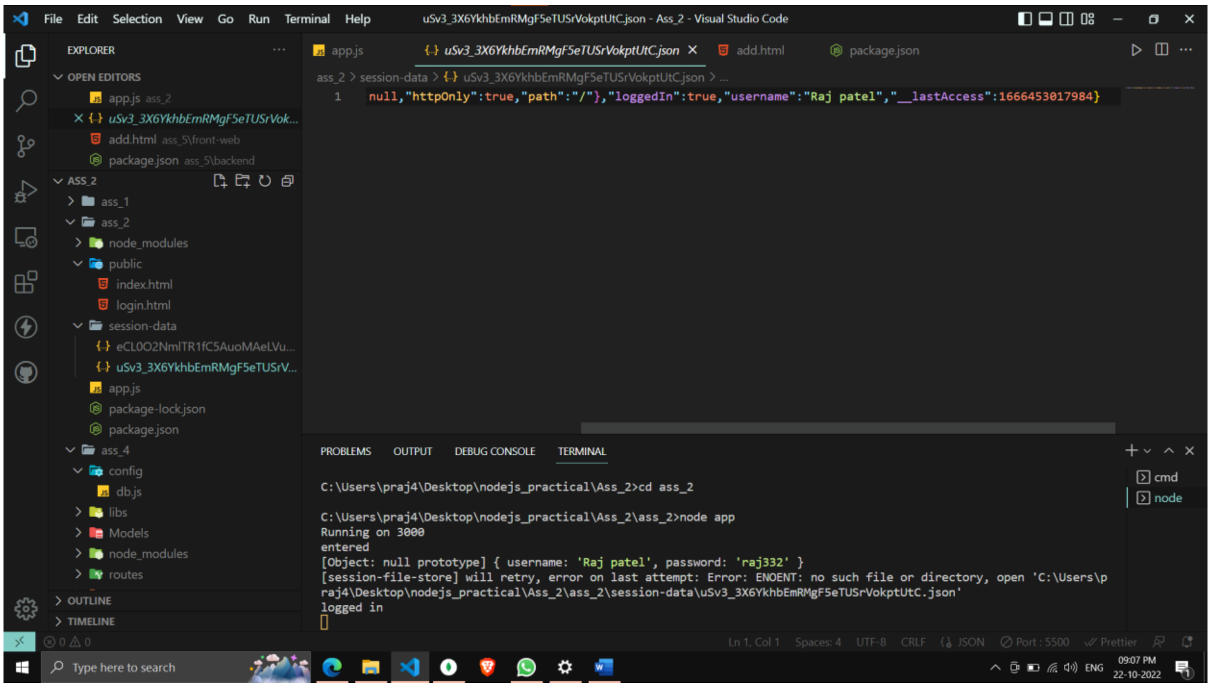
Raj patel

raj332

Log In



Login Success !!



```
null,"httpOnly":true,"path":"/"},"loggedIn":true,"username":"Raj patel","__lastAccess":1666453017984}
```

```
C:\Users\praj4\Desktop\nodejs_practical\Ass_2>cd ass_2

C:\Users\praj4\Desktop\nodejs_practical\Ass_2\ass_2>node app
Running on 3000
entered
[Object: null prototype] { username: 'Raj patel', password: 'raj332' }
[session-file-store] will retry, error on last attempt: Error: ENOENT: no such file or directory, open 'C:\Users\p
raj4\Desktop\nodejs_practical\Ass_2\ass_2\session-data\uSv3_3X6YkhbEmRMgF5eTUSrVokptUtC.json'
logged in
```

Q-4 :

App.js

```
const express = require('express');
const app = express();
const {verify} = require("./libs/jwt")
app.use(express.urlencoded({extended:true}));
app.use(express.json())
app.get('/',verify,(req,res)=>{
    res.send("Hello World")
})
app.set("view engine",'ejs');
app.use('/user',require('./routes/user'))
app.use('/student',require('./routes/student'))
app.listen(3000,()=>{
    console.log("Listening 3000")
});
```

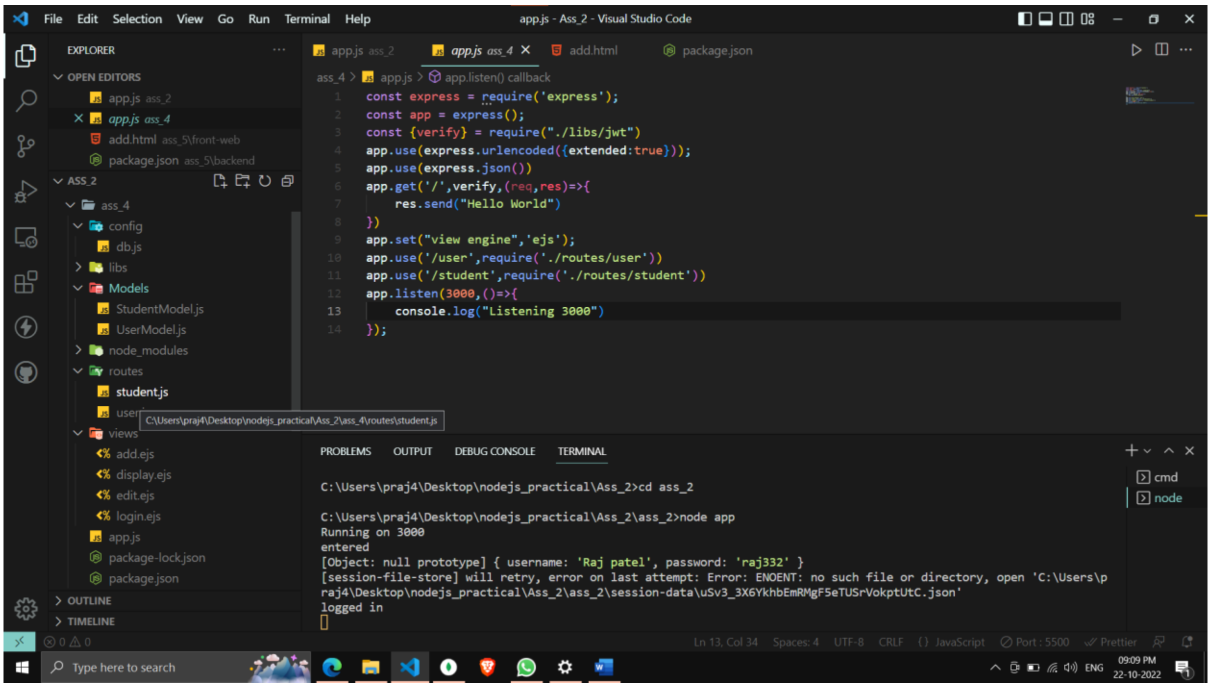
Student router:

```
const router = require("express").Router();
const { verify } = require("jsonwebtoken");
const Student = require("../Models/StudentModel");

router.get('/display',(req,res)=>{
        Student.find((err,students)=>{
```

```
        if(err){
            res.send(err);
        }
        res.render("display",{students:students})
         })
})

router.get('/add',(req,res)=>{
    res.render("add");
})

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

let student1 = new Student({
    name :req.body.name,
    rollno :req.body.rollno,
    semester :req.body.semester,
    grade :req.body.grade
})
    student1.save((err, student)=>{
     if(err){
        console.log(err);
     }
     res.redirect("/student/display");
    })
})
router.get("/edit/:id",(req,res)=>{
    Student.findOne({"_id":req.params.id },(err,student)=>{
        if(err){
            res.send(err)
        }
        if(!student){
            res.send("No Student found")
        }
        res.render("edit",{student :student})
    })
})
router.post("/update",(req,res)=>{
    Student.findOneAndUpdate({"rollno":req.body.rollno},
    {
        name:req.body.name,
        rollno:req.body.rollno,
        semester:req.body.semester,
        grade : req.body.grade
    },
    {new:true},function(err,student){
        if(err){
            res.send(err)
        }
        res.redirect("/student/display")
    }
    )
})

router.get("/delete/:id",(req,res)=>{

    Student.findByIdAndDelete({"_id":req.params._id},(err,student)=>{
        if(err){
            res.send(err)
        }
        res.redirect("/student/display")
    })
})

module.exports = router ;
```

User Router :

```
const router = require("express").Router();
let User = require("../Models/UserModel")
const {check ,validationResult }= require("express-validator")
const {sign} = require("../libs/jwt")
router.get("/login",(req,res)=>{
    res.render("login");
})
router.post("/login",
[
    check("email").isEmail()
],async (req,res)=>{
    const errors = validationResult(req);
    if(! errors.isEmpty()){
        return res.status(400).json({error:errors.array()});
    }
    let {email ,password}= req.body ;
    console.log(req.body)
    try {
        let user = await  User.findOne({email})
        if(!user){
           return res.status(400).json({error : "user not found"})
        }
 console.log(user)
 console.log(password)
        if(user.password == password){
            sign(req,res);

        }else{
            return res.status(400).json({error : "wrong password"})
        }
    }catch(err){
        res.send(err);
    }
})
module.exports = router;
```



```
const express = require('express');
const app = express();
const {verify} = require("./libs/jwt")
app.use(express.urlencoded({extended:true}));
app.use(express.json())
app.get('/',verify,(req,res)=>{
    res.send("Hello World")
})
app.set("view engine",'ejs');
app.use('/user',require('./routes/user'))
app.use('/student',require('./routes/student'))
app.listen(3000,()=>{
    console.log("Listening 3000")
});
```

```
C:\Users\praj4\Desktop\nodejs_practical\Ass_2>cd ass_2

C:\Users\praj4\Desktop\nodejs_practical\Ass_2\ass_2>node app
Running on 3000
entered
[Object: null prototype] { username: 'Raj patel', password: 'raj332' }
[session-file-store] will retry, error on last attempt: Error: ENOENT: no such file or directory, open 'C:\Users\p
raj4\Desktop\nodejs_practical\Ass_2\ass_2\session-data\uSv3_3X6YkhbEmRMgF5eTUSrVokptUtC.json'
logged in
```

StudentModel:

```
const mongoose = require('../config/db');

const StudentSchema = mongoose.Schema({
    name: String ,
    rollno : Number ,
    semester : Number ,
    grade : String
});

let Student = mongoose.model('Student',StudentSchema,'students')
module.exports = Student
```

User Model :

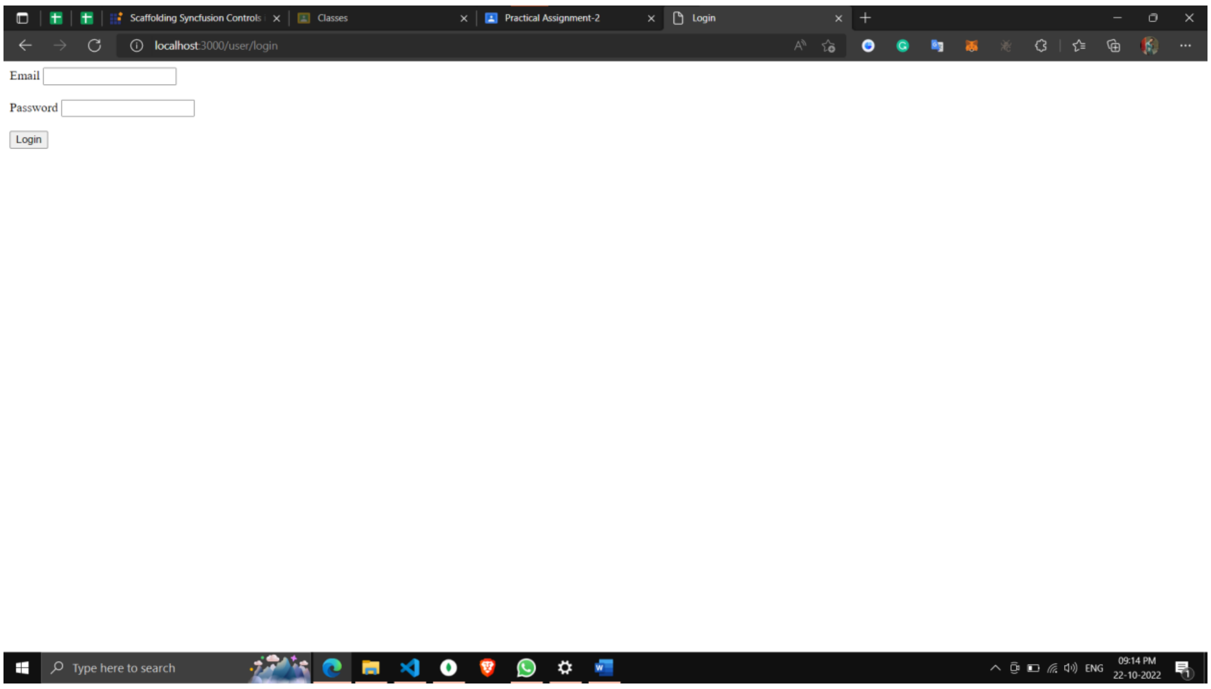
```
const mongoose = require('../config/db');
const UserSchema = mongoose.Schema({

    email : String,
    password :String

});
```

```
let User = mongoose.model('User',UserSchema,"users");
module.exports = User ;
```

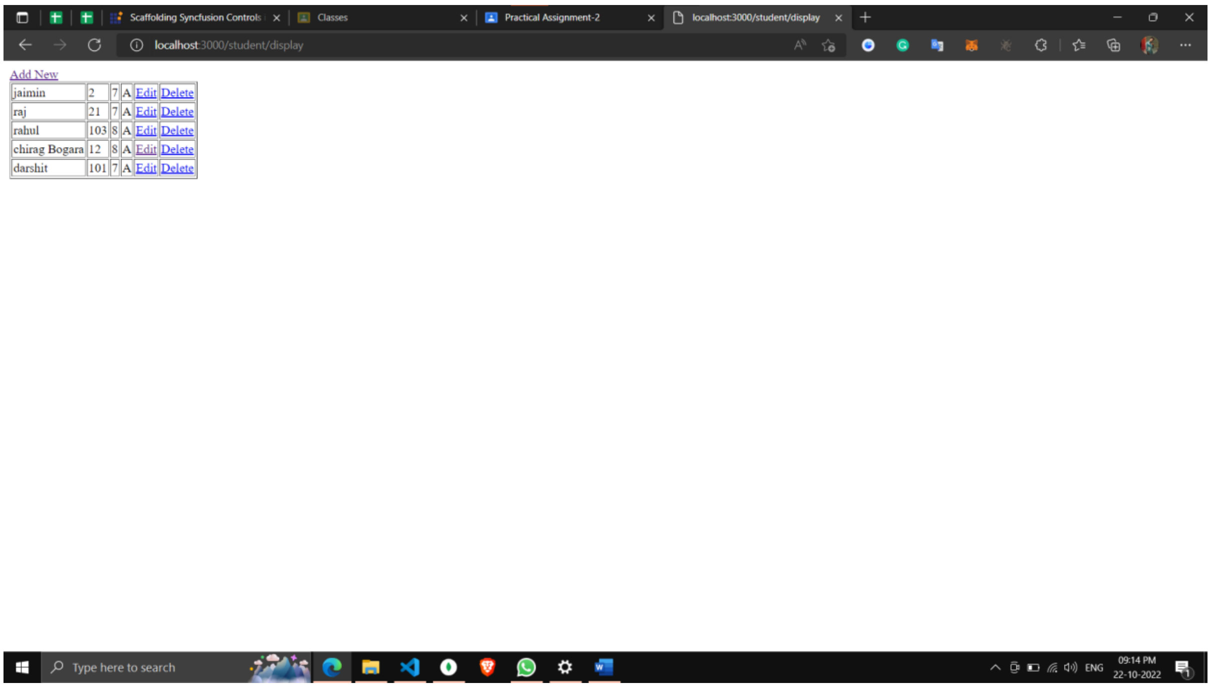
DB.js

```
const mongoose = require('mongoose');
mongoose.connect("mongodb://localhost:27017/User",{ useNewUrlParser: true
},(err)=>{
    if(error){
        console.log("Error in Db Connections")
    }else{
        console.log("DB Connected");
    }
});

module.exports = mongoose;
```

Login :



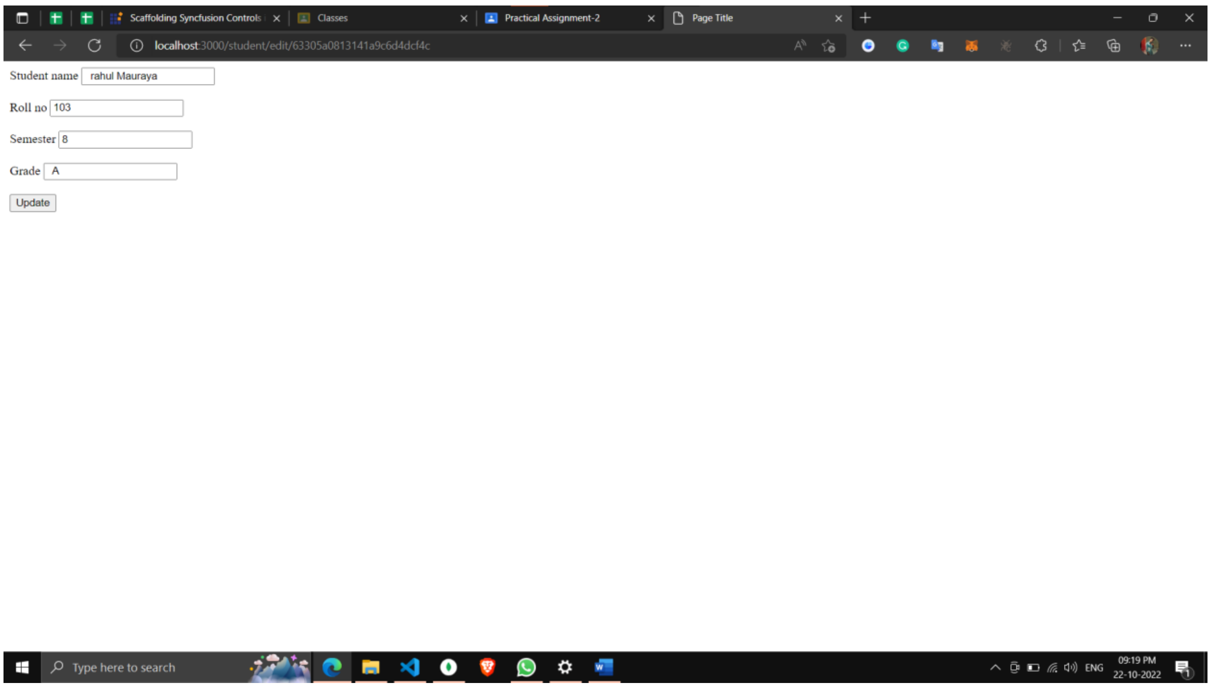
Display :


localhost:3000/student/display

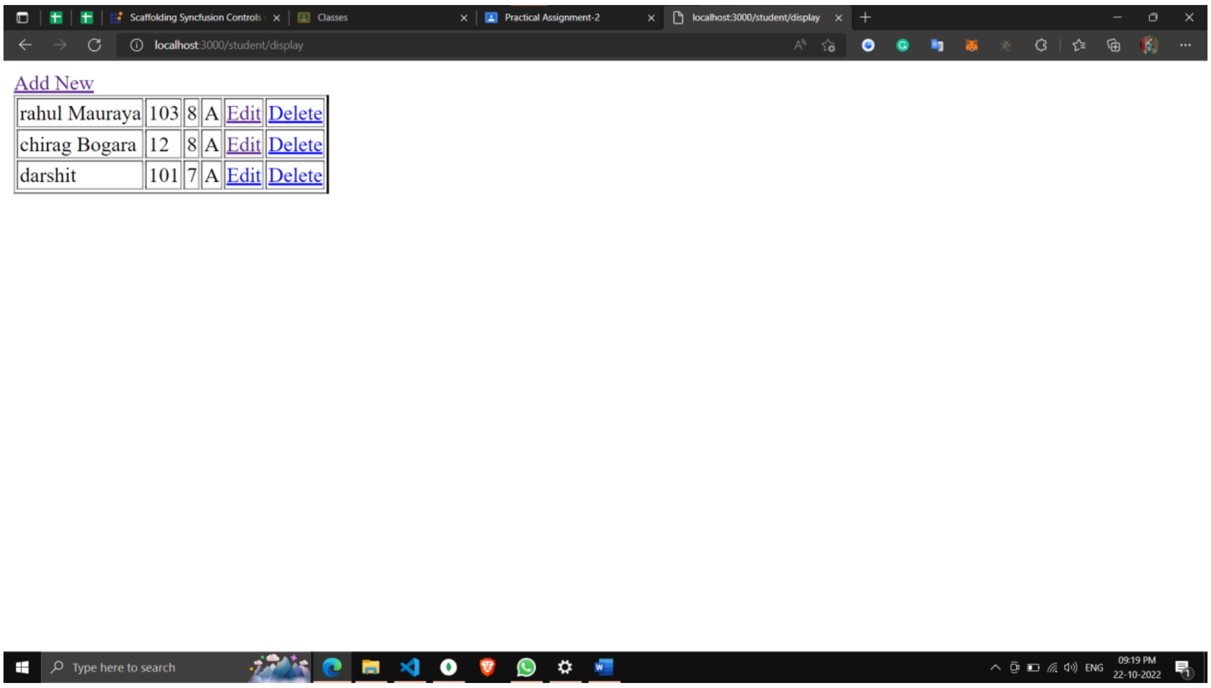
Add New

jaimin	2	7	A	Edit	Delete
raj	21	7	A	Edit	Delete
rahul	103	8	A	Edit	Delete
chirag Bogara	12	8	A	Edit	Delete
darshit	101	7	A	Edit	Delete

Update:


localhost:3000/student/edit/63305a0813141a9c6d4dcf4c

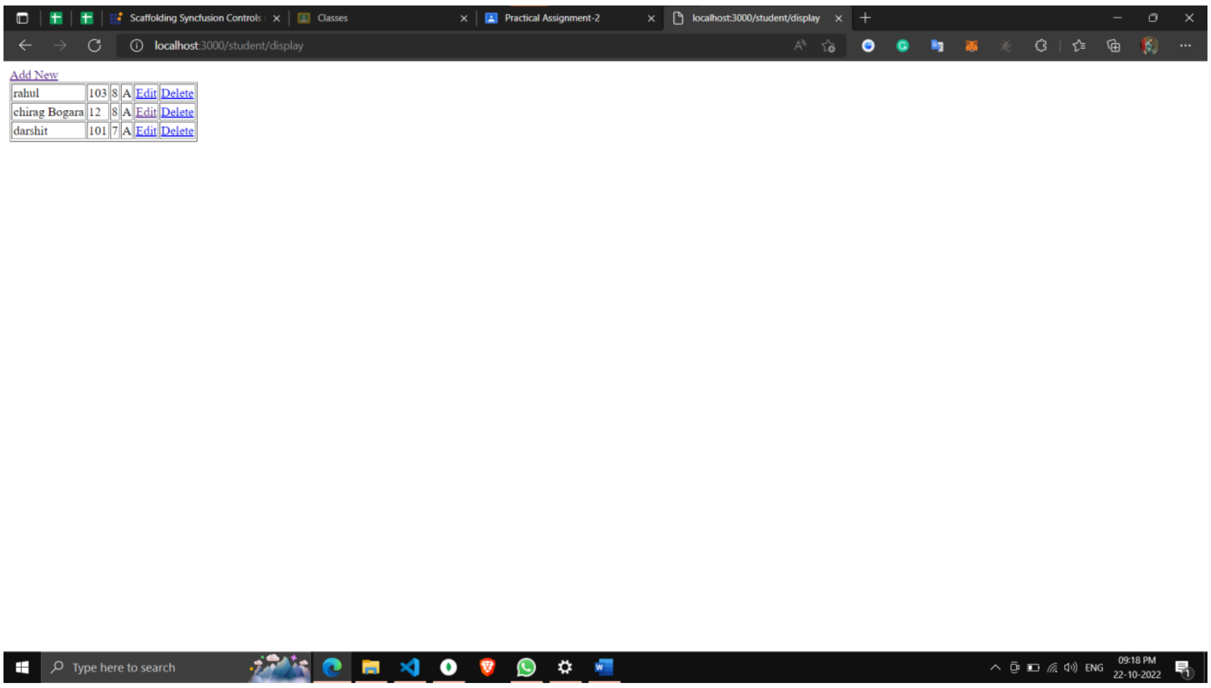
Student name	rahul Mauraya
Roll no	103
Semester	8
Grade	A



Add New

rahul Mauraya	103	8	A	Edit	Delete
chirag Bogara	12	8	A	Edit	Delete
darshit	101	7	A	Edit	Delete

After Deletion:



Add New

rahul	103	8	A	Edit	Delete
chirag Bogara	12	8	A	Edit	Delete
darshit	101	7	A	Edit	Delete

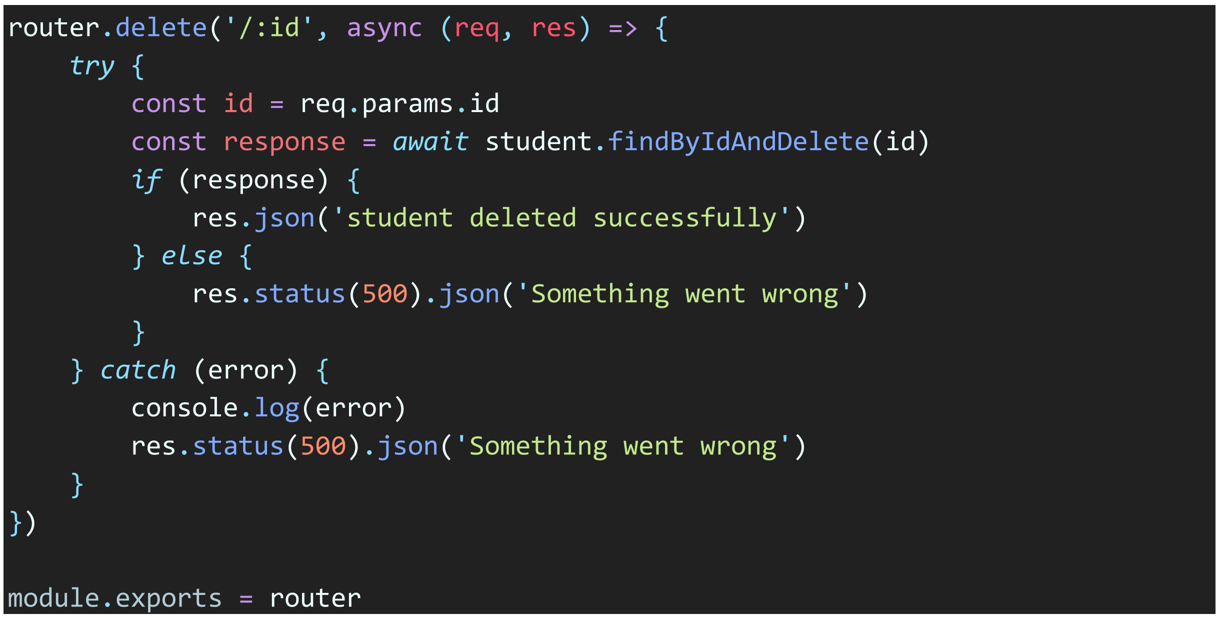
Q-4 :

App.js

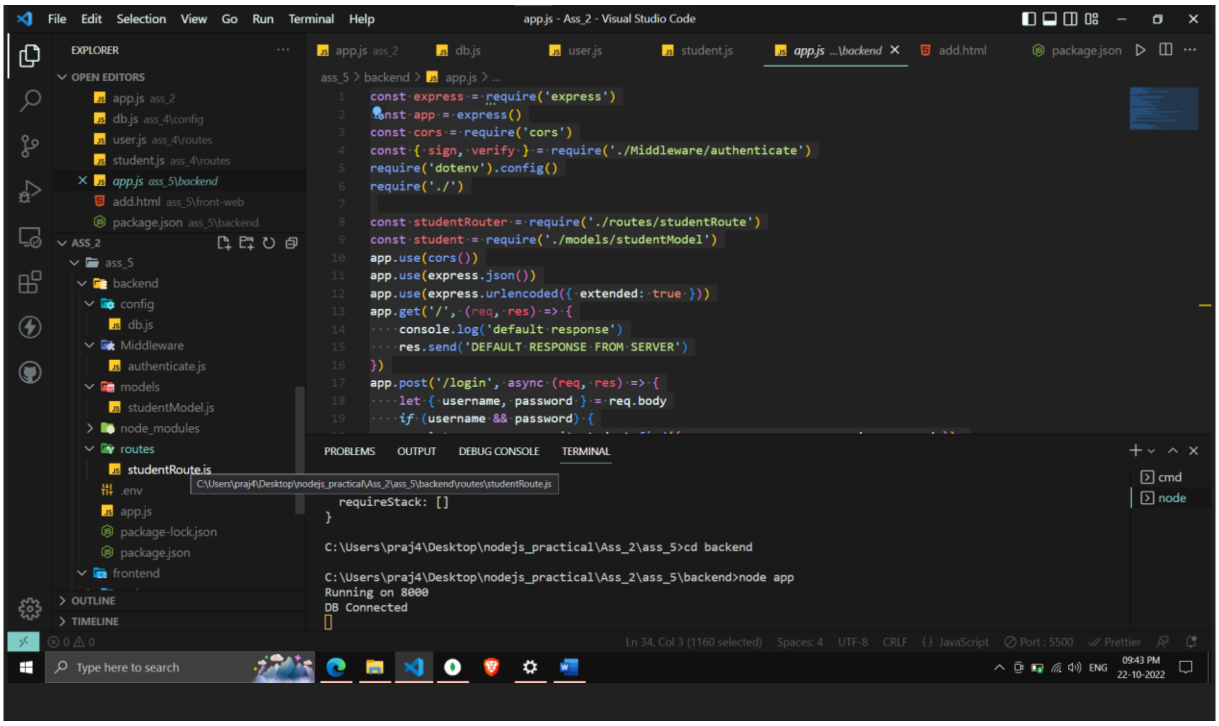
```
const express = require('express')
const app = express()
const cors = require('cors')
const { sign, verify } = require('./Middleware/authenticate')
require('dotenv').config()
require('./')

const studentRouter = require('./routes/studentRoute')
const student = require('./models/studentModel')
app.use(cors())
app.use(express.json())
app.use(express.urlencoded({ extended: true }))
app.get('/', (req, res) => {
    console.log('default response')
    res.send('DEFAULT RESPONSE FROM SERVER')
})
app.post('/login', async (req, res) => {
    let { username, password } = req.body
    if (username && password) {
        let response = await student.find({ name: username, password: password
});
        if (response.length === 1) {
            return res.json(sign({ username }))
        } else {
            return res.status(401).json('Invalid credentials')
        }
    } else {
        return res.status(401).json('Please provide username and password')
    }
})
app.use('/get-token', sign)
app.use('/student', verify, studentRouter)
app.listen(process.env.PORT, () => {
    console.log(`Running on ${process.env.PORT}`);
})
```

Student Route:

```
const express = require('express')
const router = express.Router()
const student = require('../models/studentModel')
router.get('/', async (req, res) => {
    try {
        const data = await student.find()
        if (data) {
            return res.send(data)
        } else {
            return res.status(404).send('No Data Found')
        }
    } catch (error) {
        console.error(error);
        return res.status(500).send('Something went wrong')
    }
})

router.get('/:id', async (req, res) => {
    const id = req.params.id
    try {
        const data = await student.findById(id)
        if (data) {
            return res.send(data)
        } else {
            return res.status(404).send('No Data Found')
        }
    } catch (error) {
        console.log(error)
        res.status(500).send('Something went wrong')
    }
})

router.post('/', async (req, res) => {
    try {
        const data = req.body
        const newStudent = new student({ ...data })
        const response = await newStudent.save()
        res.json("Student added successfully")
    } catch (error) {
        console.log(error)
        res.status(500).send('Something went wrong')
    }
})

router.put('/:id', async (req, res) => {
    try {
        const data = req.body
        const id = req.params.id
        const response = await student.findByIdAndUpdate(id, data)
        if (response) {
            return res.json("Student updated successfully")
        } else {
            return res.status(404).send('No data found !!')
        }
    } catch (error) {
        console.error(error);
        res.status(500).send('Something went wrong!!')
    }
})
```

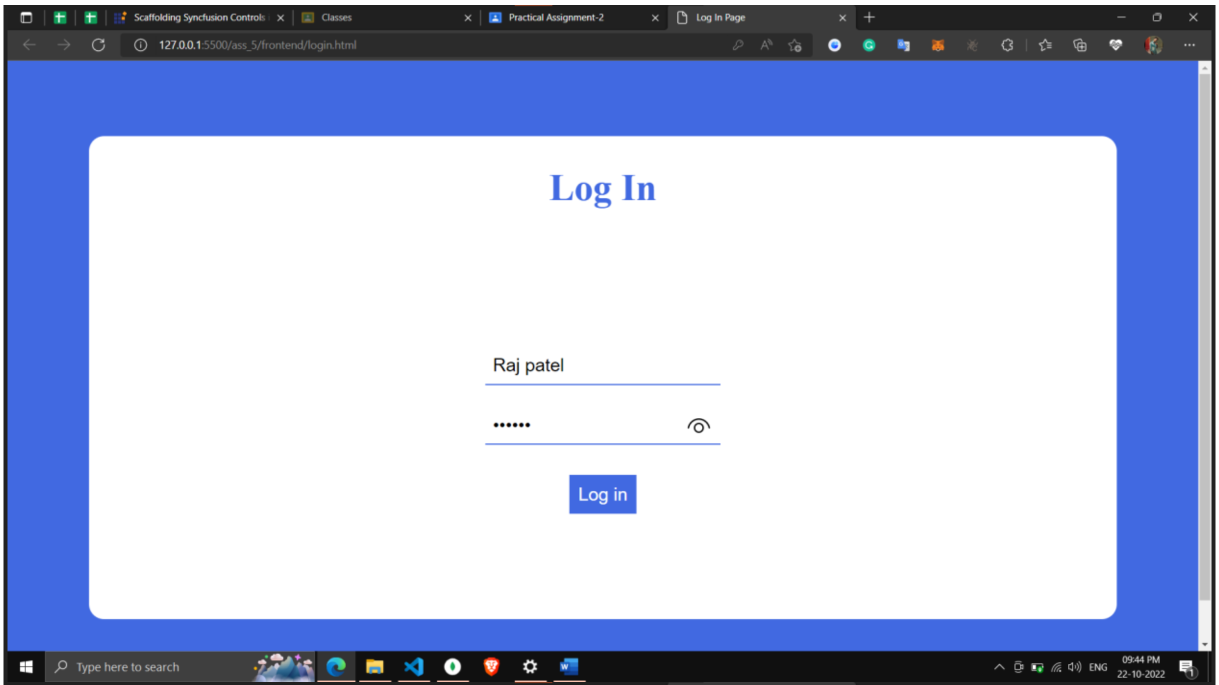


```javascript
router.delete('/:id', async (req, res) => {
    try {
        const id = req.params.id
        const response = await student.findByIdAndDelete(id)
        if (response) {
            res.json('student deleted successfully')
        } else {
            res.status(500).json('Something went wrong')
        }
    } catch (error) {
        console.log(error)
        res.status(500).json('Something went wrong')
    }
})

module.exports = router
```



```javascript
const express = require('express')
const app = express()
const cors = require('cors')
const { sign, verify } = require('./Middleware/authenticate')
require('dotenv').config()
require('./')

const studentRouter = require('./routes/studentRoute')
const student = require('./models/studentModel')
app.use(cors())
app.use(express.json())
app.use(express.urlencoded({ extended: true }))
app.get('/', (req, res) => {
    console.log('default response')
    res.send('DEFAULT RESPONSE FROM SERVER')
})
app.post('/login', async (req, res) => {
    let { username, password } = req.body
    if (username && password) {
```

```
  requireStack: []
}

C:\Users\praj4\Desktop\nodejs_practical\Ass_2\ass_5>cd backend

C:\Users\praj4\Desktop\nodejs_practical\Ass_2\ass_5\backend>node app
Running on 8000
DB Connected
```

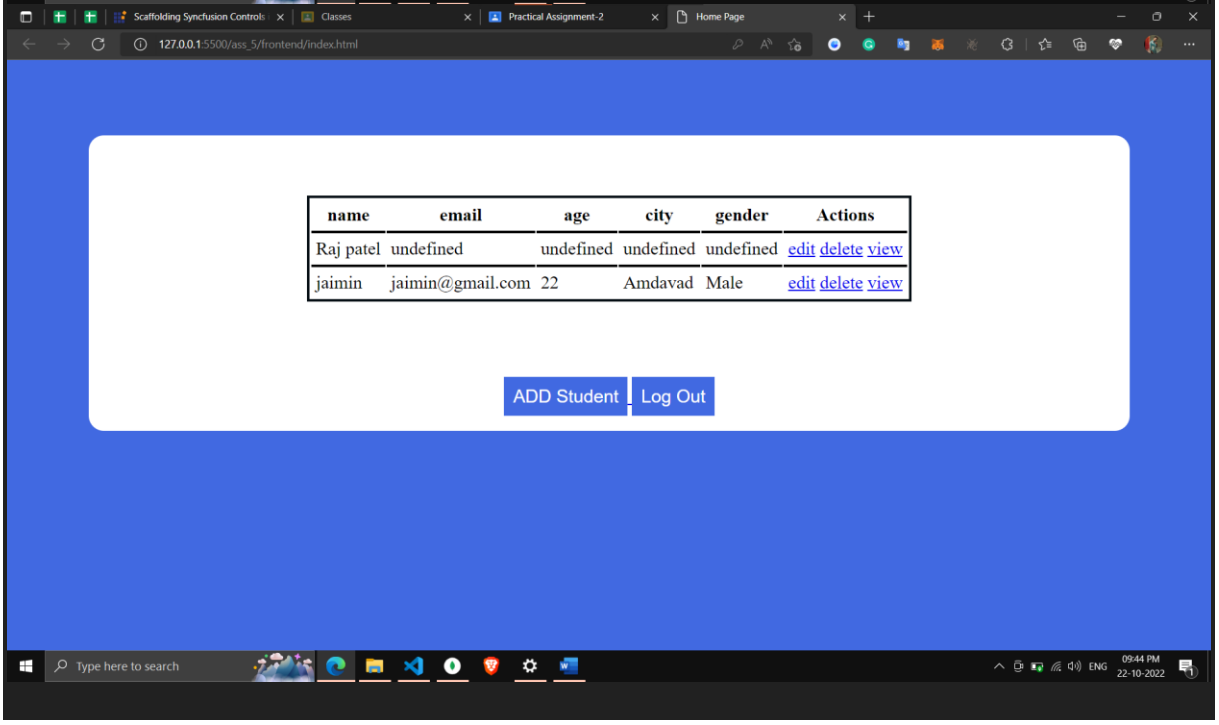

Scaffolding Syncfusion Controls
Classes
Practical Assignment-2
Log In Page
127.0.0.1:5500/ass_5/frontend/login.html

Log In

Raj patel

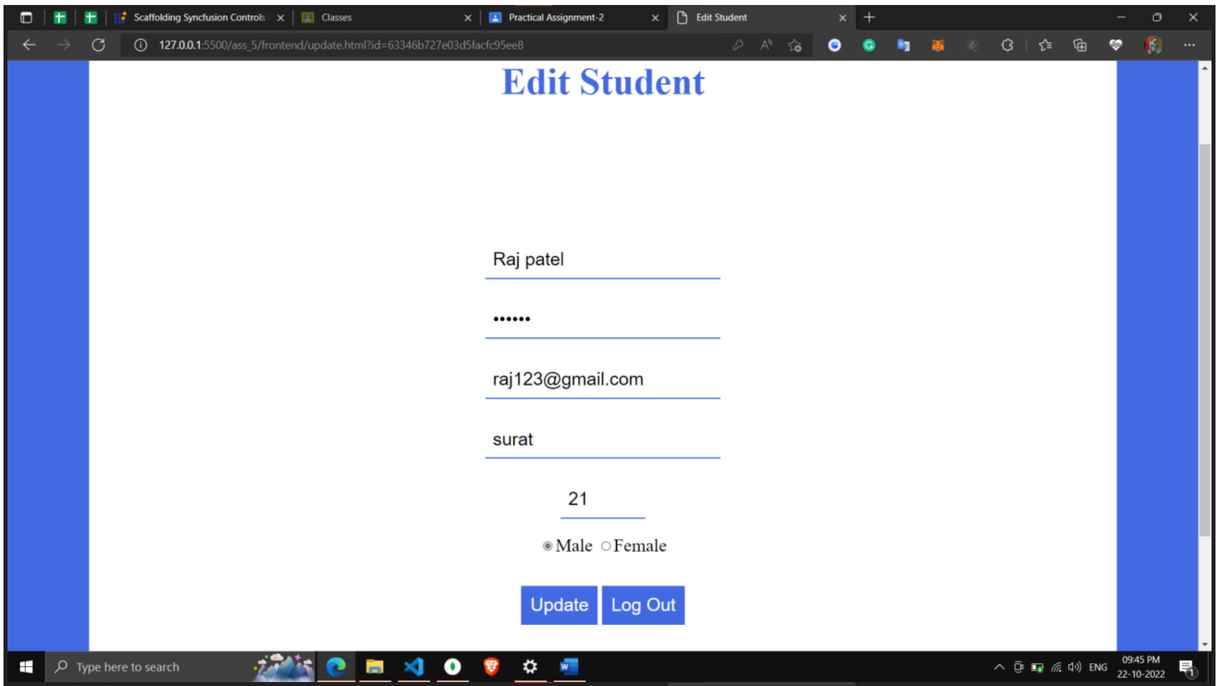
••••••

Log in


Scaffolding Syncfusion Controls
Classes
Practical Assignment-2
Home Page
127.0.0.1:5500/ass_5/frontend/index.html

name	email	age	city	gender	Actions
Raj patel	undefined	undefined	undefined	undefined	edit delete view
jaimin	jaimin@gmail.com	22	Amdavad	Male	edit delete view

ADD Student Log Out



Edit Student

Raj patel

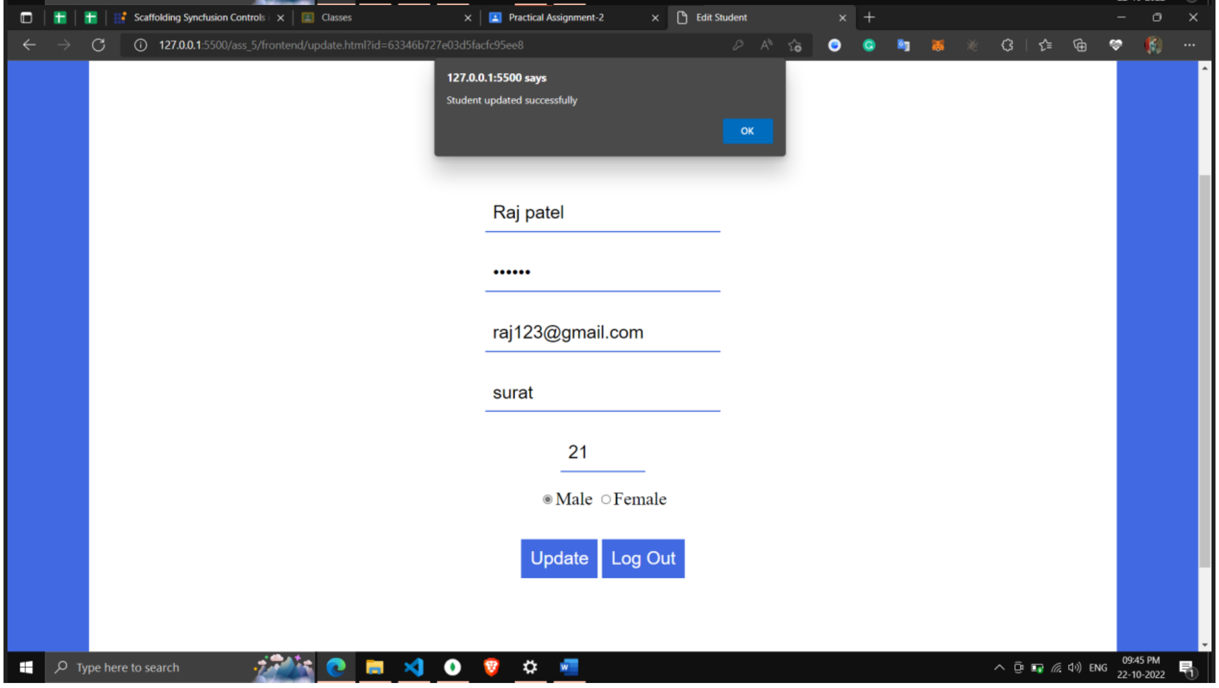
••••••

raj123@gmail.com

surat

21

[✓ Male] Female



127.0.0.1:5500 says

Student updated successfully

Raj patel

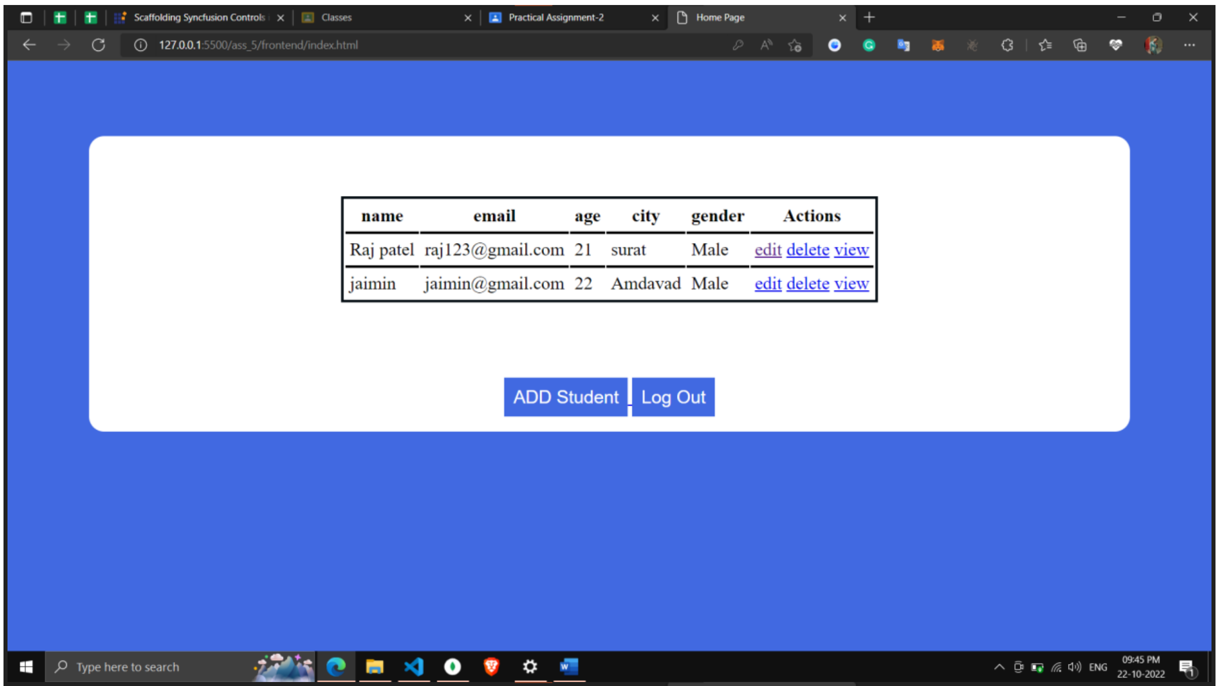
••••••

raj123@gmail.com

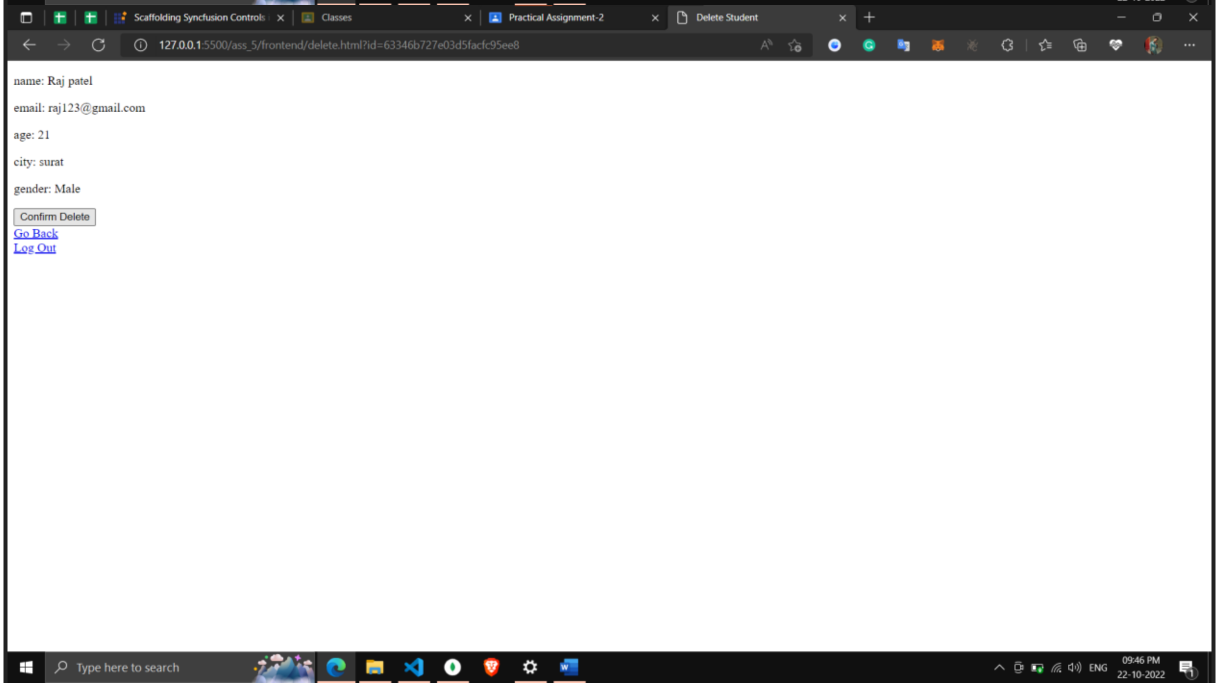
surat

21

[✓ Male] Female


Scaffolding Syncfusion Controls
Classes
Practical Assignment-2
Home Page
127.0.0.1:5500/ass_5/frontend/index.html

name	email	age	city	gender	Actions
Raj patel	raj123@gmail.com	21	surat	Male	edit delete view
jaimin	jaimin@gmail.com	22	Amdavad	Male	edit delete view


Scaffolding Syncfusion Controls
Classes
Practical Assignment-2
Delete Student
127.0.0.1:5500/ass_5/frontend/delete.html?id=63346b727e03d5facfc95ee8

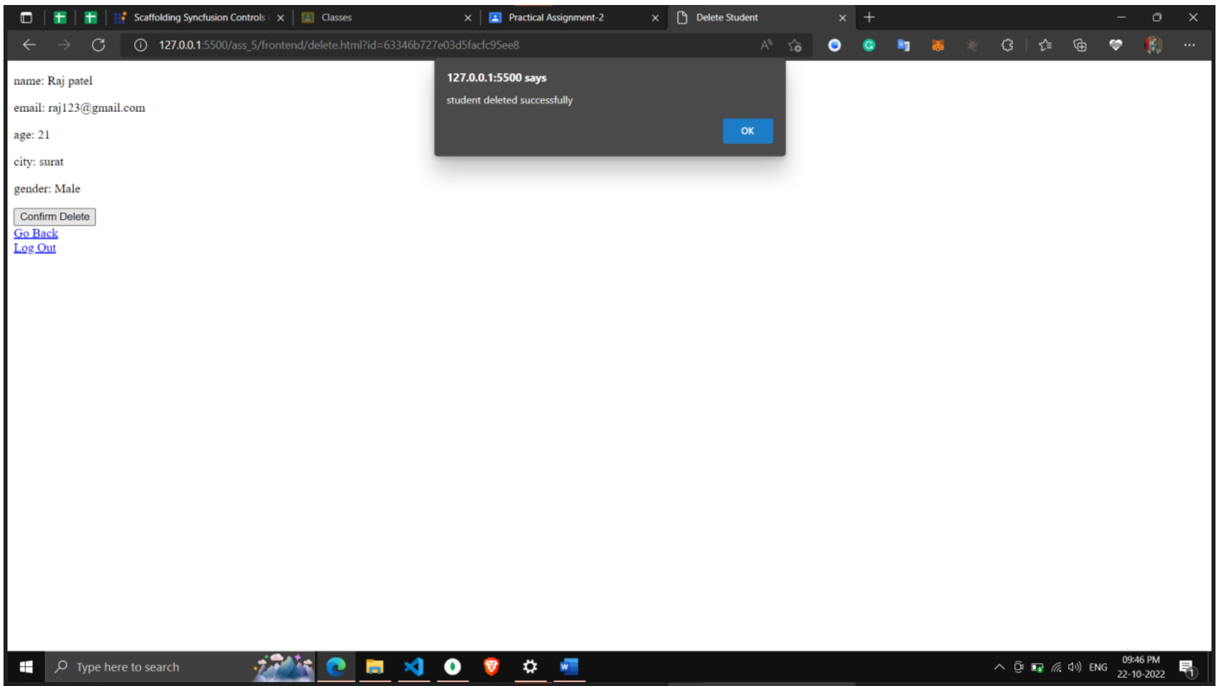
name: Raj patel

email: raj123@gmail.com

age: 21

city: surat

gender: Male


127.0.0.1:5500/ass_5/frontend/delete.html?id=63346b727e03d5facfc95ee8

127.0.0.1:5500 says

student deleted successfully

OK

name: Raj patel

email: raj123@gmail.com

age: 21

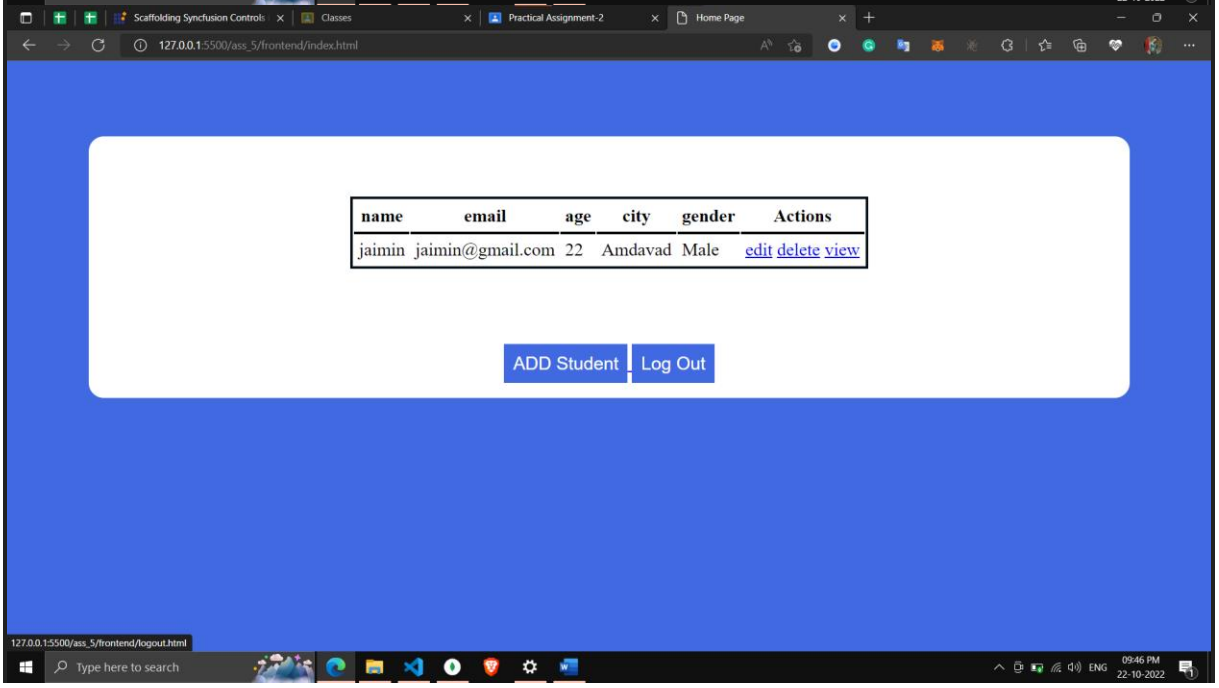
city: surat

gender: Male

Confirm Delete

Go Back

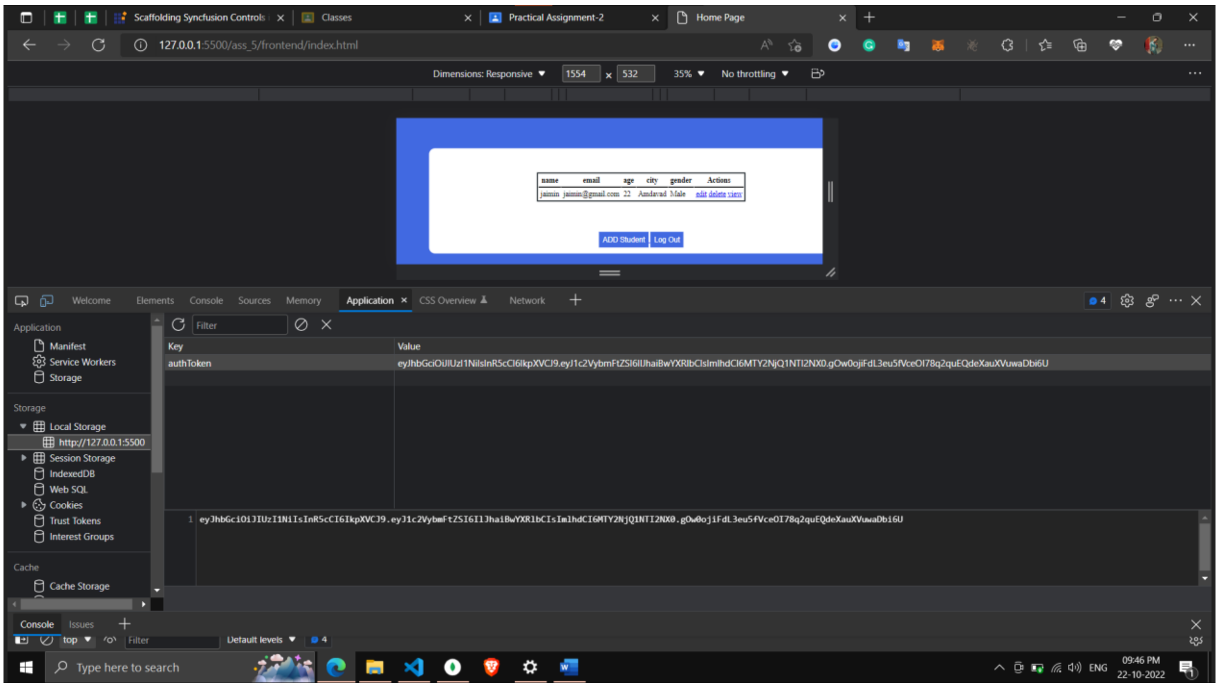
Log Out


127.0.0.1:5500/ass_5/frontend/index.html

name	email	age	city	gender	Actions
jaimin	jaimin@gmail.com	22	Amdavad	Male	edit delete view

ADD Student Log Out

127.0.0.1:5500/ass_5/frontend/logout.html



name	email	age	city	gender	Actions
jaimin	jaimin@gmail.com	22	Amdavad	Male	edit delete view

ADD Student Log Out

Key	Value
authToken	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VybmFtZSI6IlJhaiBwYXRlbCIsImlhdCI6MTY2NjQ1NTI2NX0.gOw0ojiFdL3eu5fVceOI78q2quEQdeXauXVuwaDbi6U

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VybmFtZSI6IlJhaiBwYXRlbCIsImlhdCI6MTY2NjQ1NTI2NX0.gOw0ojiFdL3eu5fVceOI78q2quEQdeXauXVuwaDbi6U
```