Practical Assignment - 2

Name: Saniya Baxi

Roll No: 50

Semester-7

1. **File Upload Using Express**

Code:

- App.js

```
const express = require('express')
const fileUpload = require('express-fileupload')
const path = require('path')

const app = express()

app.set('view engine','ejs')

app.use(fileUpload())

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

    res.render('index')

})

app.post('/single', async(req, res, next)=>{

    try{

const file = req.files.mFile
console.log(file)
const fileName = new Date().getTime().toString() +
path.extname(file.name) //to give new name to file while uploading
const savePath = path.join(__dirname, 'public','uploads',fileName)
```

```
if(file.truncated)
{
    throw new Error('File size is too big') //to show that file size is
too huge
}

if(file.mimetype!=='image/png')
{
    throw new Error("Only png is supported") //to show that only jpeg
files are supported
}

await file.mv(savePath)
res.redirect('/')
    }

    catch(error)
    {
        console.log(error)
        res.send("Error uploading file")
    }
})

app.post('/multiple', async(req, res, next)=>{

    try{

        const files = req.files.mFiles

const promises = files.map((file) => {
    const savePath = path.join(__dirname, 'public', 'uploads',
file.name)
    return file.mv(savePath)
})

        await Promise.all(promises)
        res.redirect('/')

    }catch(error){

        console.log(error)
        res.send("Error uploading file")
```

```
    }

})

app.listen(3000, ()=>
    console.log("server running on port 3000")
)
```

- Index.js

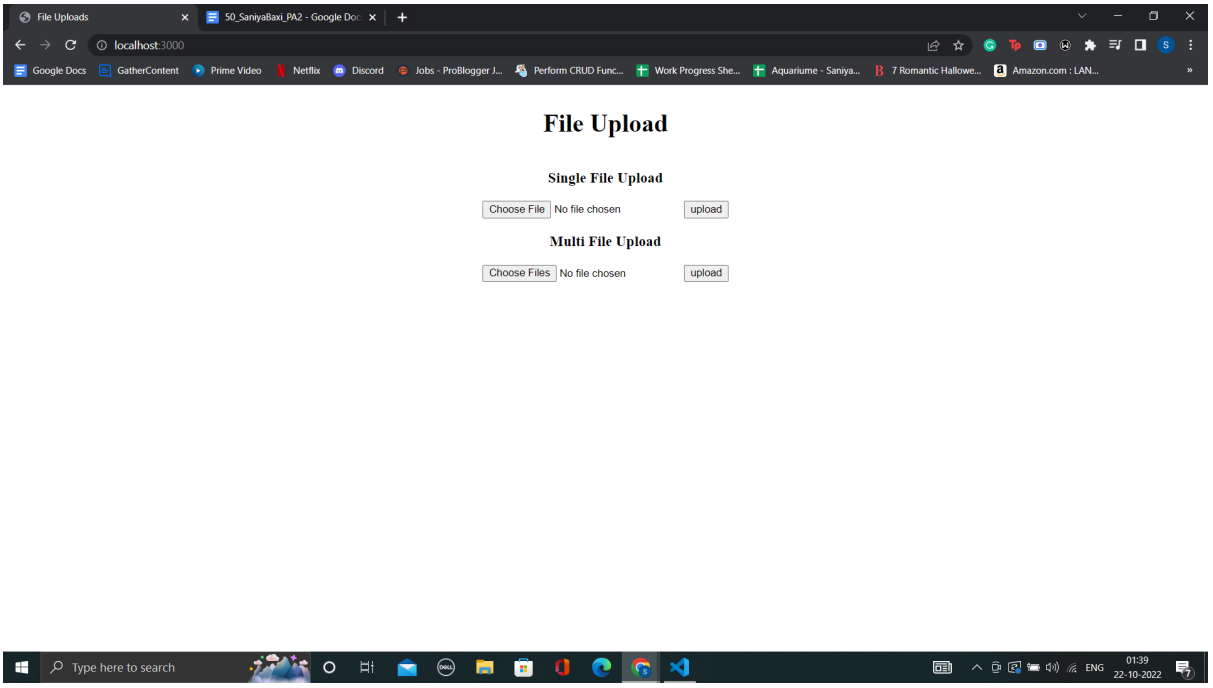
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>File Uploads</title>
    <style>
        body{
          display: grid;
          place-items: center;
        };
    </style>
</head>
<body>

    <h1> File Upload </h1>
    <h3> Single File Upload </h3>

    <form action="/single" method="post" enctype="multipart/form-data">

        <input type="file" name="mFile" id="" required
accept="image/jpeg, image/png, image/svg+xml, application/pdf" />
        <input type="submit" value="upload" />

    </form>

    <h3> Multi File Upload </h3>

    <form action="/multiple" method="post"
enctype="multipart/form-data">
```

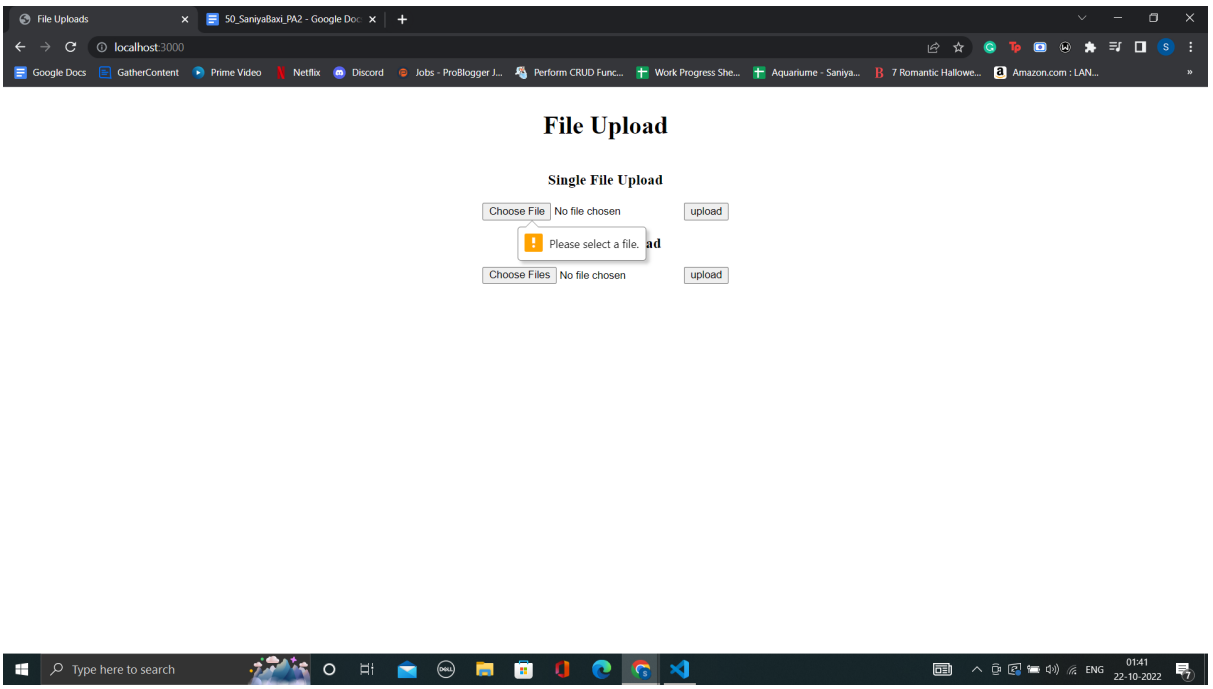
```
        <input type="file" name="mFiles" id="" required
accept="image/jpeg, image/png, image/svg+xml, application/pdf" multiple
/>
        <input type="submit" value="upload" />

    </form>


</body>
</html>
```

Screenshots:

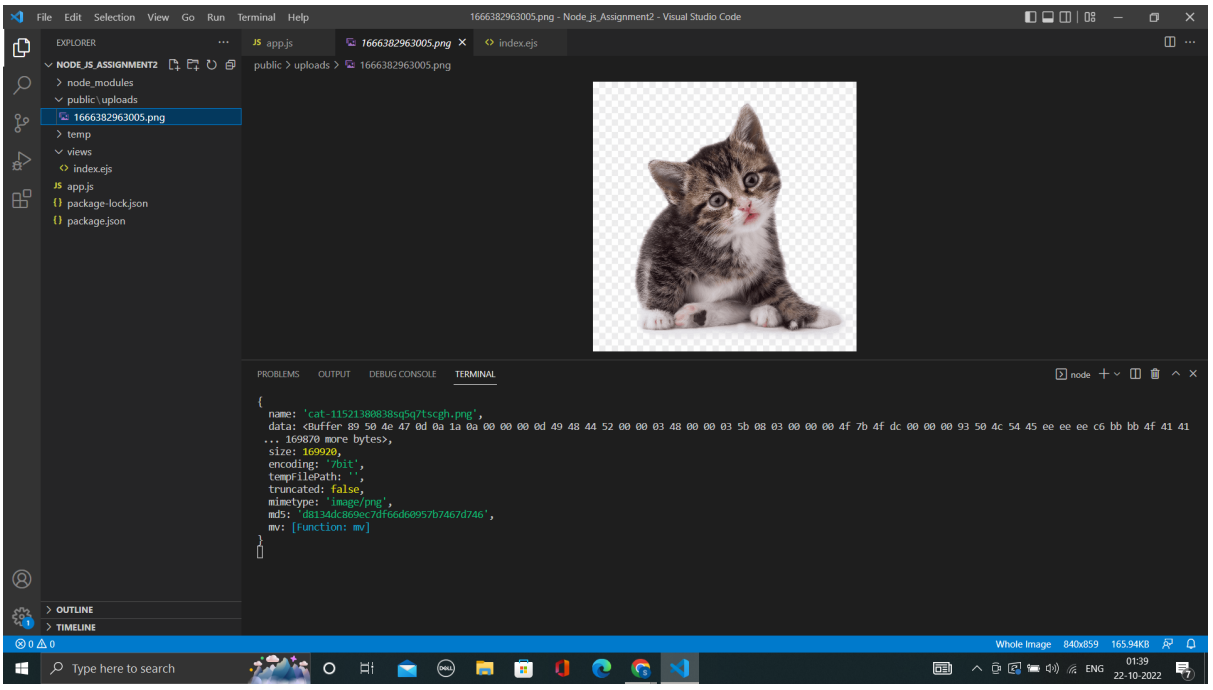
- Single File Upload

-> Inserting one file

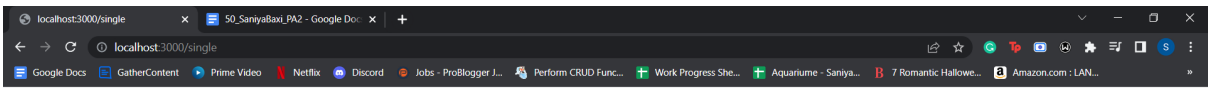


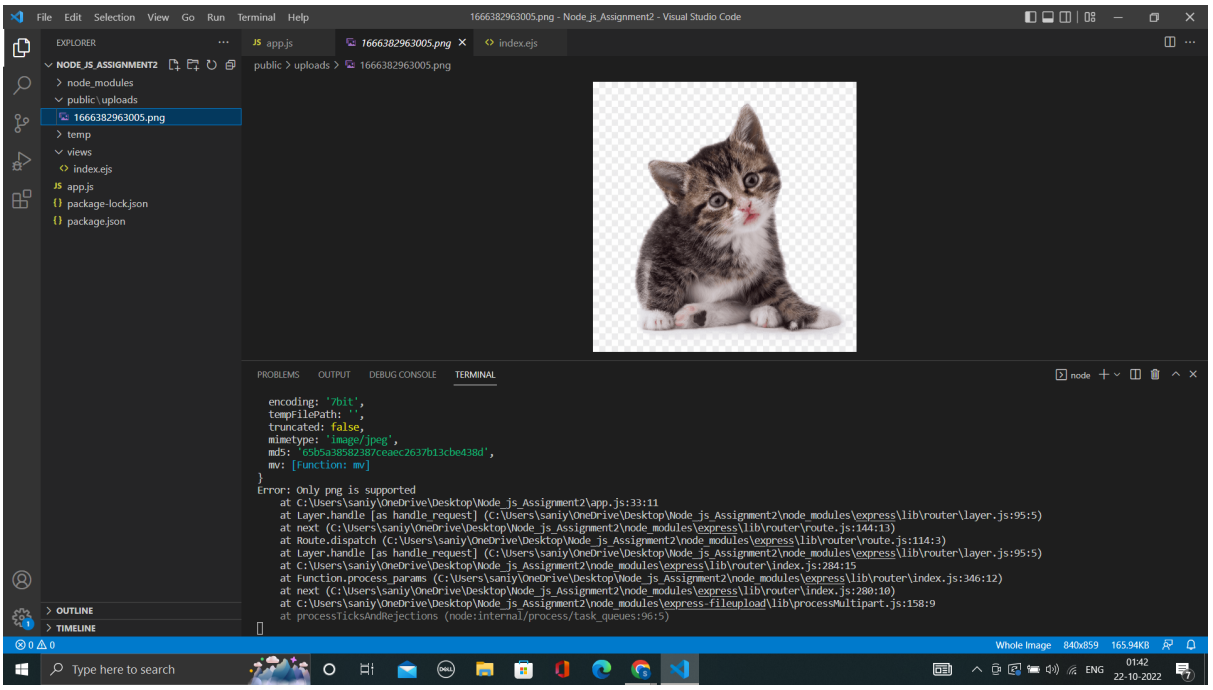


-> File inserted in the uploads folder. The name is changed as well.



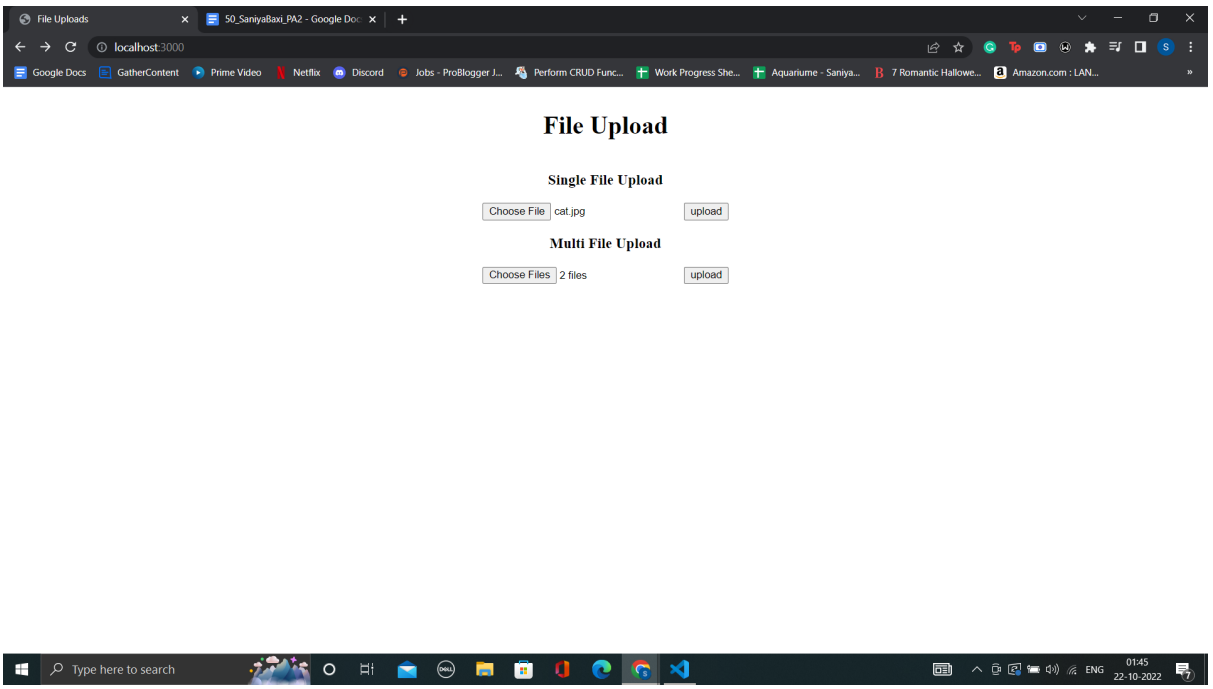
-> Error uploading file as only PNG is allowed



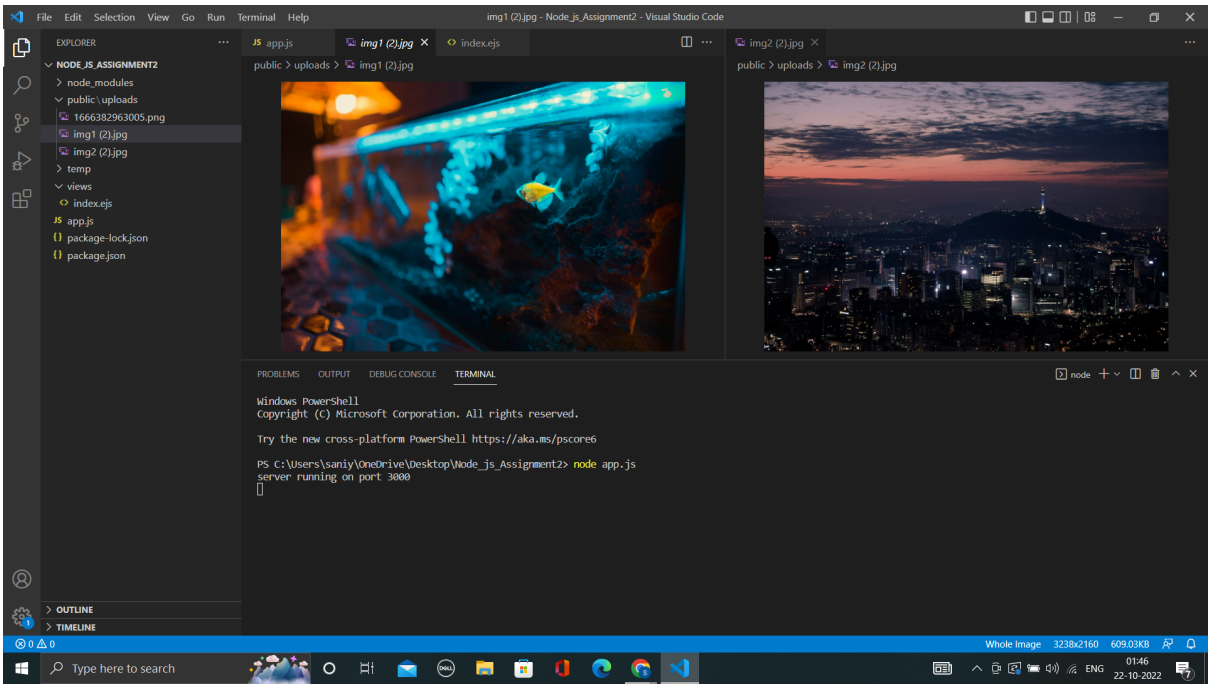


- **Multiple File Upload**

-> Uploading two files together



-> Both files stored in uploads folder



2. Login Using File Session Store

Code:

- Database.js

```
const mysql = require('mysql')

const connection = mysql.createConnection({
    host: 'localhost',
    user: 'root',
    password: '',
    database: 'test_tb'
})

connection.connect(function(error){
    if(error)
    {
        throw error
    }else
    {
        console.log('Connection established')
    }
})

module.exports = connection
```

- App.js

```
var createError = require('http-errors');
var express = require('express');
var path = require('path');
var cookieParser = require('cookie-parser');
var logger = require('morgan');

var session = require('express-session')
const FileStore = require('session-file-store')(session);


var indexRouter = require('./routes/index');
var usersRouter = require('./routes/users');

var app = express();

app.use(session({

  secret: 'secretword',
```

```
  resave: true,
  saveUninitialized: true,
  store: new FileStore({ path: './session-data'})


}))

// view engine setup
app.set('views', path.join(__dirname, 'views'));
app.set('view engine', 'ejs');

app.use(logger('dev'));
app.use(express.json());
app.use(express.urlencoded({ extended: false }));
app.use(cookieParser());
app.use(express.static(path.join(__dirname, 'public')));

app.use('/', indexRouter);
app.use('/users', usersRouter);

// catch 404 and forward to error handler
app.use(function(req, res, next) {
  next(createError(404));
});

// error handler
app.use(function(err, req, res, next) {
  // set locals, only providing error in development
  res.locals.message = err.message;
  res.locals.error = req.app.get('env') === 'development' ? err : {};

  // render the error page
  res.status(err.status || 500);
  res.render('error');
});

app.listen(3000)

module.exports = app;
```

- Index.js

```
var express = require('express');
const session = require('express-session');
const app = require('../app');
var router = express.Router();

var database = require('../database')

/* GET home page. */
router.get('/', function(req, res, next) {
  res.render('index', { title: 'Express', session : req.session});
});

router.post('/login',function(request, response, next){
  var user_email_address = request.body.user_email_address
  var user_password = request.body.user_password

  if(user_email_address && user_password)
  {
      query = `

      SELECT * from user_login WHERE user_email =
"${user_email_address}"

      `


    database.query(query, function(error, data){

      if(data.length > 0)
      {
          for(var count = 0; count < data.length ; count++)
          {
            if(data[count].user_password == user_password)
            {
                request.session.user_id = data[count].user_id
                response.redirect("/")
            }else
            {
              response.send('Incorrect Password')
            }
          }
      }
      else
```

```
      {
        res.send('Incorrect Email Address')
      }
    })

  }else{
    response.send('Please enter email address and password details')
    response.end()
  }
})

router.get('/logout', function(request, response, next){
  request.session.destroy();
  response.redirect("/")
})

module.exports = router;
```

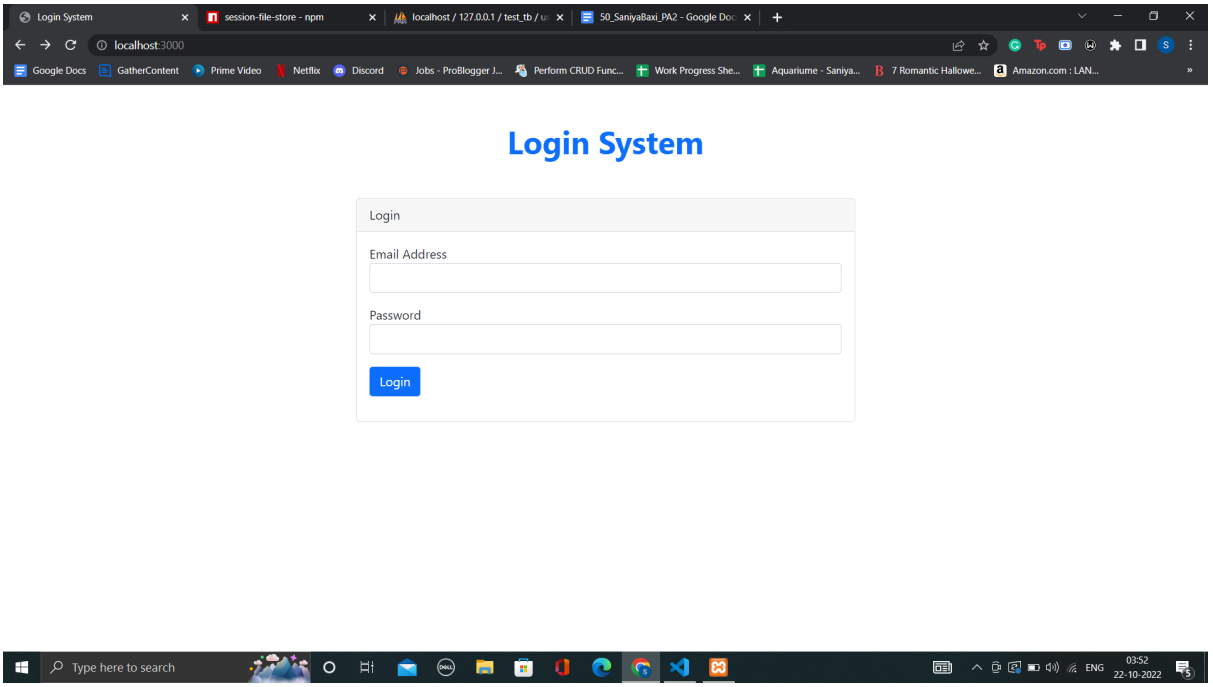
- Index.ejs

```
<!doctype html>
<html lang="en">
    <head>
        <!-- Required meta tags -->
        <meta charset="utf-8">
        <meta name="viewport" content="width=device-width,
initial-scale=1">

        <!-- Bootstrap CSS -->
        <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.m
in.css" rel="stylesheet"
integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65Vohh
puuCOmLASjC" crossorigin="anonymous">

        <title>Login System</title>
    </head>
    <body>

        <div class="container">
```

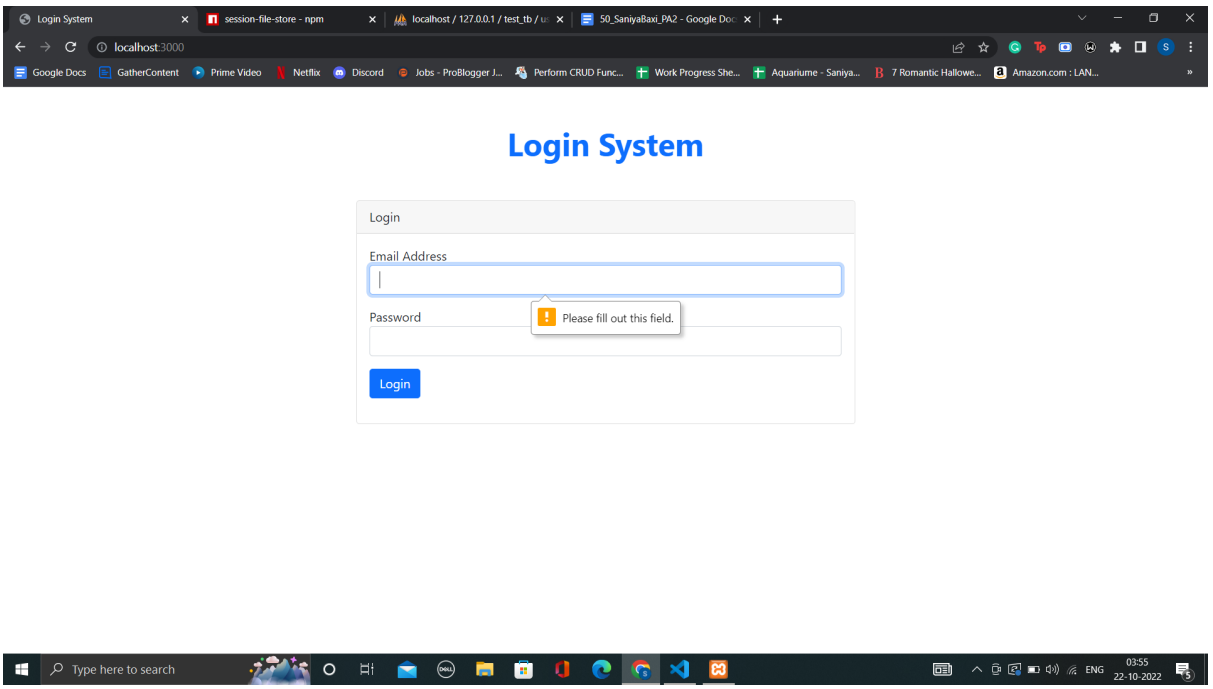
```
            <h1 class="mt-5 mb-5 text-center text-primary"><b>Login
System</b></h1>

            <div class="row mt-5">
                <div class="col-md-3"> </div>
                <div class="col-md-6">
                <% if(session.user_id) { %>

                    <h1>Welcome!</h1>

                    <a href="/logout" class="btn
btn-primary">Logout</a>

                <% } else { %>

                    <div class="card">
                        <div class="card-header">Login</div>
                        <div class="card-body">
                            <form method="post" action="/login">
                                <div class="mb-3">
                                    <label>Email Address</label>
                                    <input type="email"
name="user_email_address" class="form-control" />
                                </div>
                                <div class="mb-3">
                                    <label>Password</label>
                                    <input type="password"
name="user_password" class="form-control" />
                                </div>
                                <div class="mb-3">
                                    <input type="submit" class="btn
btn-primary" value="Login" />
                                </div>
                            </form>
                        </div>
                    </div>

                <% } %>
                </div>
            </div>
        </div>
    </body>
</html>
```

Screenshots:

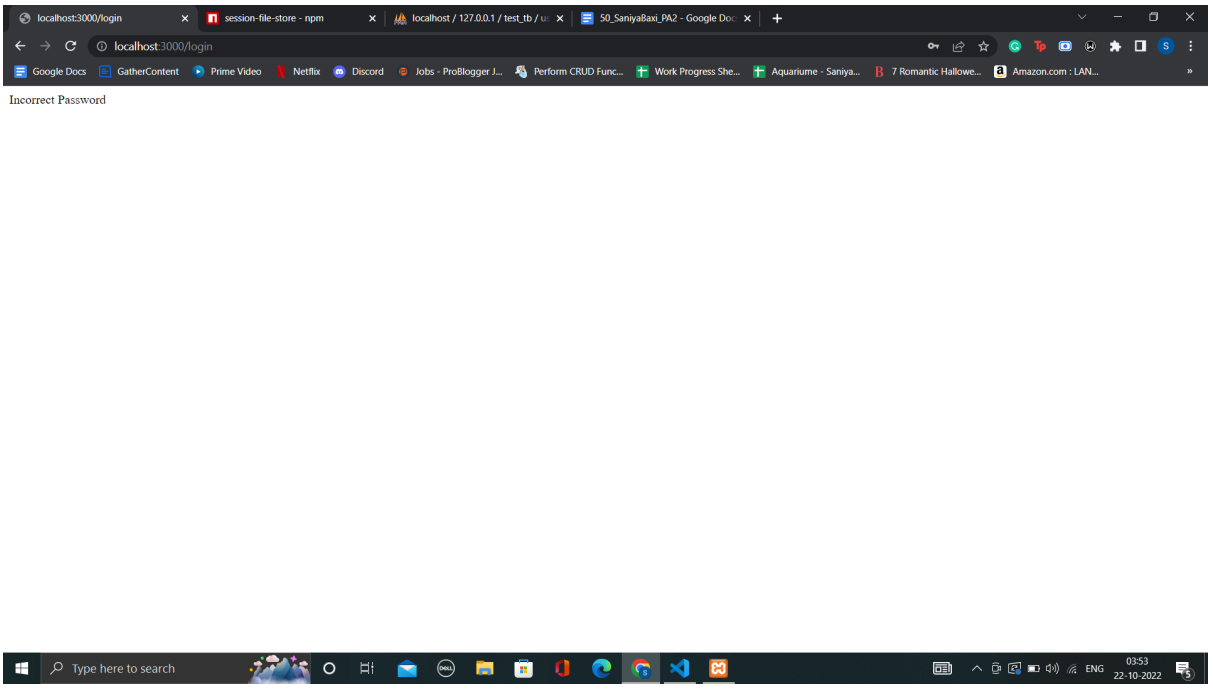
- **Main Login Page**



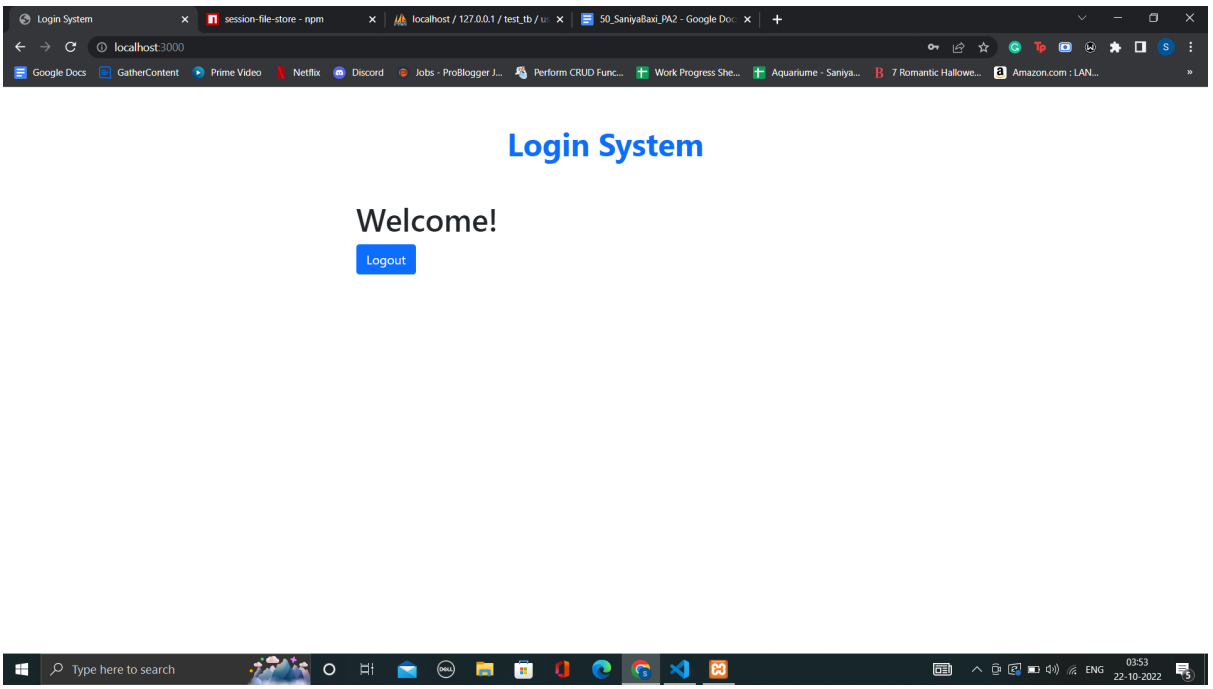
- **Empty Fields**



- **Incorrect Password**

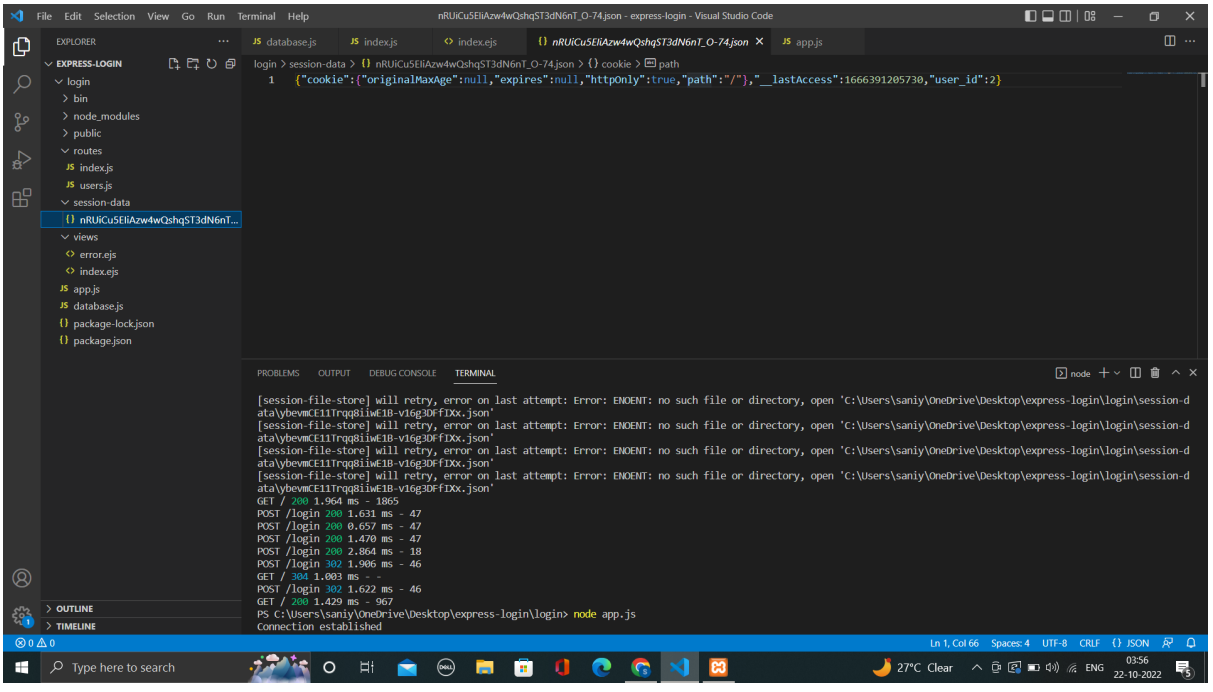


- **Login/Logout Page**

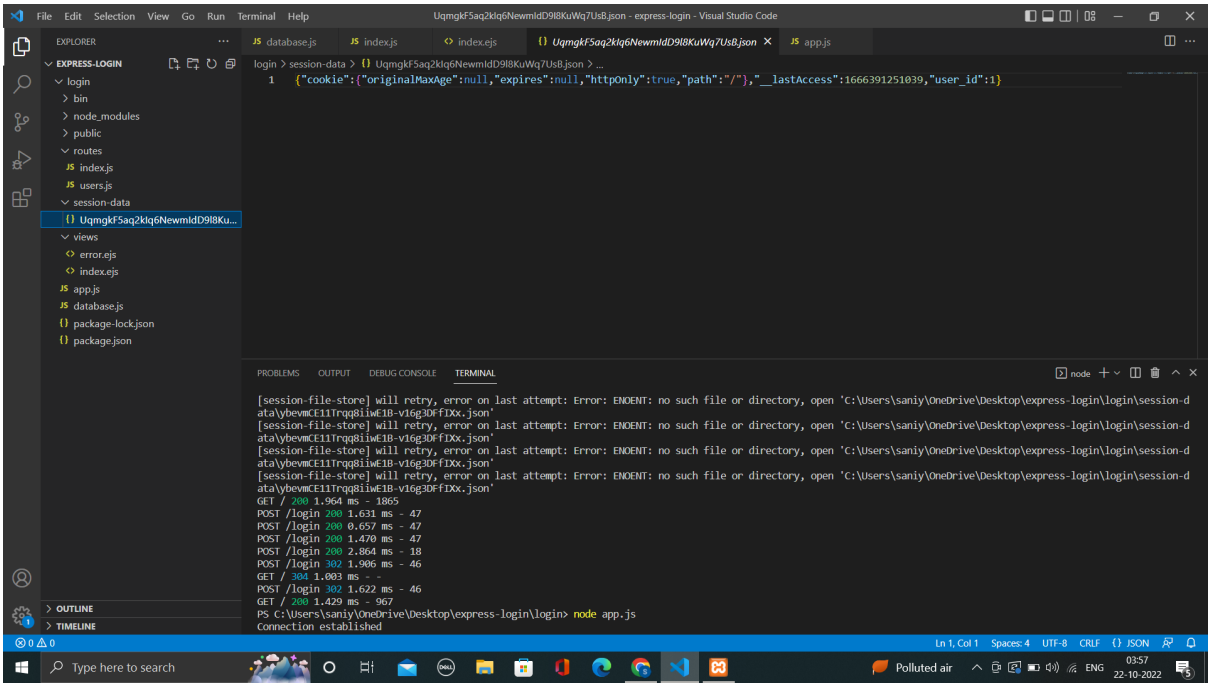


- **Session Details Stored in File**

-> user-id: 2



-> user-id: 1



3. Login (JWT, CRUD, Mongoose, Express, Template Engine, Logout)

Code:

Model File - student.js

```
const mongoose = require('mongoose')

const studentSchema = new mongoose.Schema({
    name: String,
    email: String,
    password: String,
    age: Number,
    address: String
})

module.exports = new mongoose.model('student', studentSchema,
'students')
```

Routes Folder - StudentRoute.js

```
const express = require('express');
const { verifyToken } = require('../middleware/token');
```

```
const student = express.Router();
const studentModel = require('../models/student')

student.use(verifyToken)

student.get('/add', (req, res) => {
    res.render('add');
})
student.post('/', async (req, res) => {
    try {
        const newStudent = req.body;
        const student = new studentModel({
            ...newStudent
        })
        await student.save()
        res.redirect('/')
    } catch (error) {
        res.send('Error creating student')
    }
})


student.get('/:id', async (req, res) => {
    try {
        const student = await studentModel.findById(req.params.id)
        student ? res.render('view', { data: student }) :
res.render('index', { noDataError: 'No students found' })
    } catch (error) {
        res.send('Error getting student')
    }
})

student.get('/', async (req, res) => {
    try {
        const students = await studentModel.find()
        students ? res.render('index', { data: students }) :
res.render('index', { noDataError: 'No students found' })
    } catch (error) {
        res.send('Error getting students')
    }
})
```

```
student.get('/edit/:id', async (req, res) => {
    try {
        const student = await studentModel.findById(req.params.id)
        student ? res.render('update', { data: student }) :
res.render('index', { noDataError: 'student not found' })
    } catch (error) {
        res.send('There was problem getting student')
    }
})

student.post('/edit/:id', async (req, res) => {
    console.log('student post /:id')
    try {
        const updatedStudent = req.body;
        const id = req.params.id;
        console.log('updated student: ', updatedStudent)
        delete updatedStudent['_id']
        await studentModel.findOneAndUpdate({ _id: id },
updatedStudent)
        return res.redirect('/')
    } catch (error) {
        return res.send('Couldnt update student')
    }
})


student.get('/delete/:id', async (req, res) => {
    try {
        const student = await studentModel.findById(req.params.id)
        student ? res.render('delete', { data: student }) :
res.render('index', { noDataError: 'student not found' })
    } catch (error) {
        res.send('Couldnt get student')
    }
})
student.post('/delete', async (req, res) => {
    console.log('delete student request')
    try {
        const id = req.body._id
        console.log("id: ", id)
        await studentModel.findOneAndDelete({ _id: id })
        res.redirect('/')
    } catch (error) {
```

```
        console.log('Error: ', error)
        res.send('Couldnt delete student')
    }
})

module.exports = student
```

Middleware Folder:

- Token.js

```
const jwt = require('jsonwebtoken')

const signToken = async (req, res) => {
    //TODO: implement sign token logic here
}

const verifyToken = async (req, res, next) => {
    const token = req?.cookies?.token
    if (token) {
        //verify token
        if (jwt.verify(token, process.env.TOKEN_SECRET)) {
            // return res.send('token found and verified');
            next()
        } else {
            return res.status(401).end('Unauthorized access');
        }
    }
    else {
        //redirect to login
        return res.status(401).send('Token Missing')
    }
}

module.exports = { signToken, verifyToken }
```

Views Folder:

- Index.ejs

```
<!DOCTYPE html>
<html lang="en">
```

```
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Home Page</title>

</head>

<body>
    <div class="container">
        <div class="error">
            <% if(typeof(noDataError)!=='undefined' ) { %>
                <p class="err">
                    <%= {noDataError} %>
                </p>
                <%}%>
                    <% if(typeof(accessToken)!=='undefined' ) { %>
                        <script>
                            localStorage.setItem('token', "<%=
JSON.stringify(accessToken) %>"); // <- setup token into localStorage,
(but i think it's not good place for that, and would be better get
token with another authorization request)
                        </script>
                        <%}%>
        </div>
        <div class="error">
            <% if(typeof(data)!=='undefined' ) { %>
                <table border="1">
                    <thead>
                        <tr>
                            <th>name</th>
                            <th>email</th>
                            <th>age</th>
                            <th>address</th>

                        </tr>
                    </thead>
                    <tbody>
                        <% data.forEach(student=>{
                            %>
                            <tr>
```

```
                                <td>
                                    <%= student.name %>
                                </td>
                                <td>
                                    <%= student.email %>
                                </td>
                                <td>
                                    <%= student.age %>
                                </td>
                                <td>
                                    <%= student.address %>
                                </td>

                                <td>
                                    <div class="mr-1">
                                        <a
href="/student/edit/<%=student._id%>" class="mr-1">edit</a>
                                        <a
href="/student/delete/<%=student._id%>" class="mr-1">delete</a>
                                        <a
href="/student/<%=student._id%>" class="mr-1">view</a>
                                    </div>
                                </td>
                            </tr>
                            <% }) %>
                        </tbody>
                    </table>
                    <%}%>
                        <div class="links mt-2">
                            <a href="/student/add">Add Student</a>
                        </div>
                        <div class="links mt-2">
                            <a href="/logout">Log Out</a>
                        </div>
            </div>
        </div>
    </body>

    </html>
```

- Add

```
<!DOCTYPE html>
```

```
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <!--  <link rel="stylesheet" href="/style.css" type="text/css"> -->
    <title>Add Student</title>
</head>

<body>
    <div class="container flex-center">
        <div class="header">
            <h3>Add Student</h3>
        </div>

        <div class="form-container mt-2 ">
            <form class="flex-center" action="/student/" method="post">

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

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

                Email:    <input type="email" name="email" required>
<br>

                Address    <input type="text" name="address" required>
<br>

               Age:     <input type="number" name="age"  min="10"
max="80"  required> <br>




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

            </form>


        <a href="/logout">Log Out</a>
```

```
</body>

</html>
```

- Delete

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Student Delete</title>

</head>

<body>


            <form action="/student/delete" method="post">
                <input type="hidden" name="_id" value="<%= data?._id
%>">

                Username:
                    <input type="text" name="name" value="<%=
data?.name %>" disabled> <br>

                Password:
                    <input type="password" name="password" value="<%=
data?.password %>"
                            disabled> <br>
                Email:
                    <input type="email" name="email" value="<%=
data?.email %>"
                        disabled> <br>

                Address:
                    <input type="text" name="address"  value="<%=
data?.address %>"
                        disabled> <br>
```

```
                    <input type="number" name="age" id="" min="10"
max="80"
                        value="<%= data?.age %>" disabled> <br>



                    <input type="submit" value="Delete">

            </form>


        <div class="links">
            <a href="/student/">Go Back</a>
        </div>
        <div class="links mt-2">
            <a href="/logout">Log Out</a>
        </div>

</body>

</html>
```

- View

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Student Detail</title>
</head>

<body>

        <div class="student-container">
            <p>name: <%= data.name %>
            </p>
            <p>email: <%= data.email %>
```

```
            </p>
            <p>age: <%= data.age %>
            </p>
            <p>Address: <%= data.address %>
            </p>
        </div>
        <div class="links">
            <a href="/student/">Go Back</a>
        </div>
        <div class="links mt-2">
            <a href="/logout">Log Out</a>
        </div>
        <% } else{ %>
           No Student!
            <%}%>
</body>

</html>
```

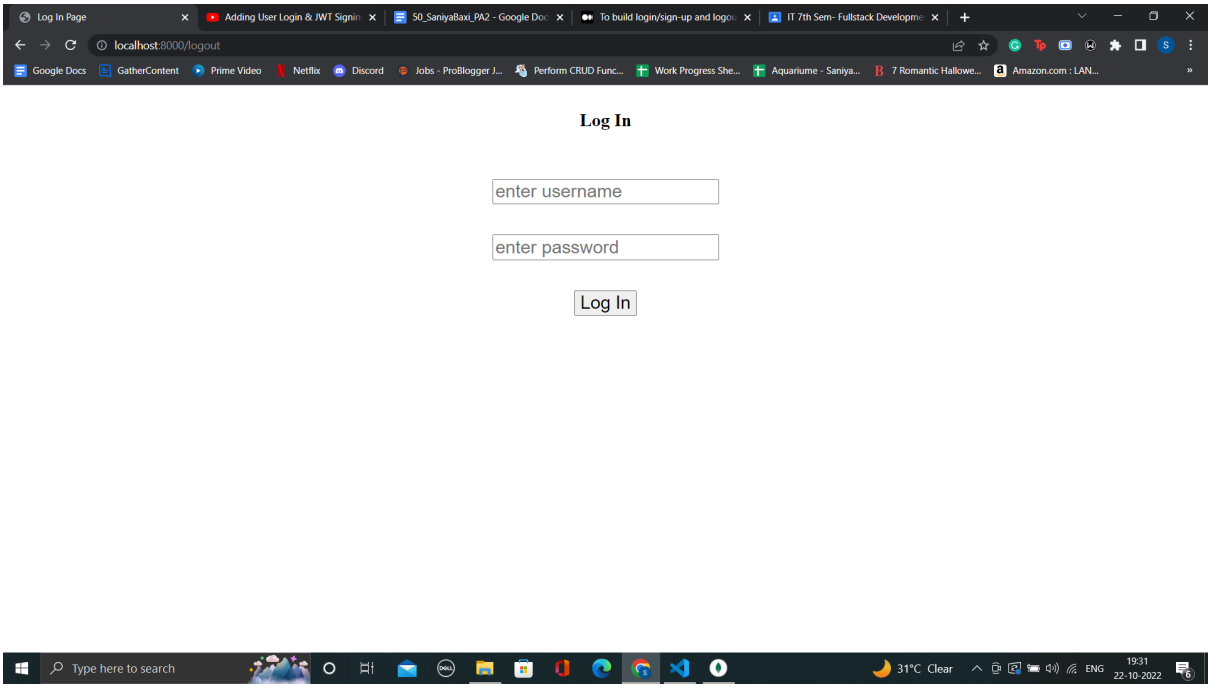
Index.js File:

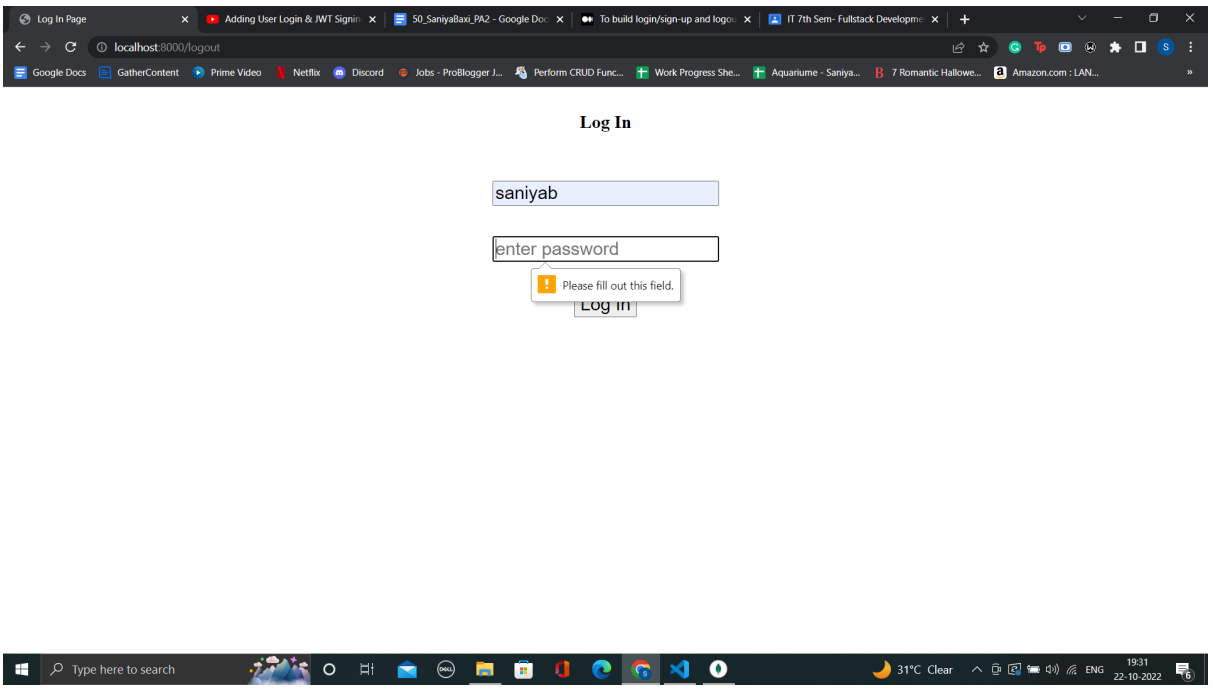
```
const express = require('express')
const app = express()
const PORT = 8000
const database = require('./db/database')
const student = require('./routes/StudentRoute')
require('dotenv').config()
const jwt = require('jsonwebtoken')
const cookieParser = require('cookie-parser')
const accessTokenSecret = process.env.TOKEN_SECRET;
const studentModel = require('./models/student')
const verifyToken = require('./middleware/token')

app.use(cookieParser())
app.set('view engine', 'ejs')
app.use(express.json())
app.use(express.urlencoded({ extended: true }))
app.use(express.static(__dirname + '/public'));

app.use('/student', student)

app.get('/', (req, res) => {
    const token = req.cookies.token
```

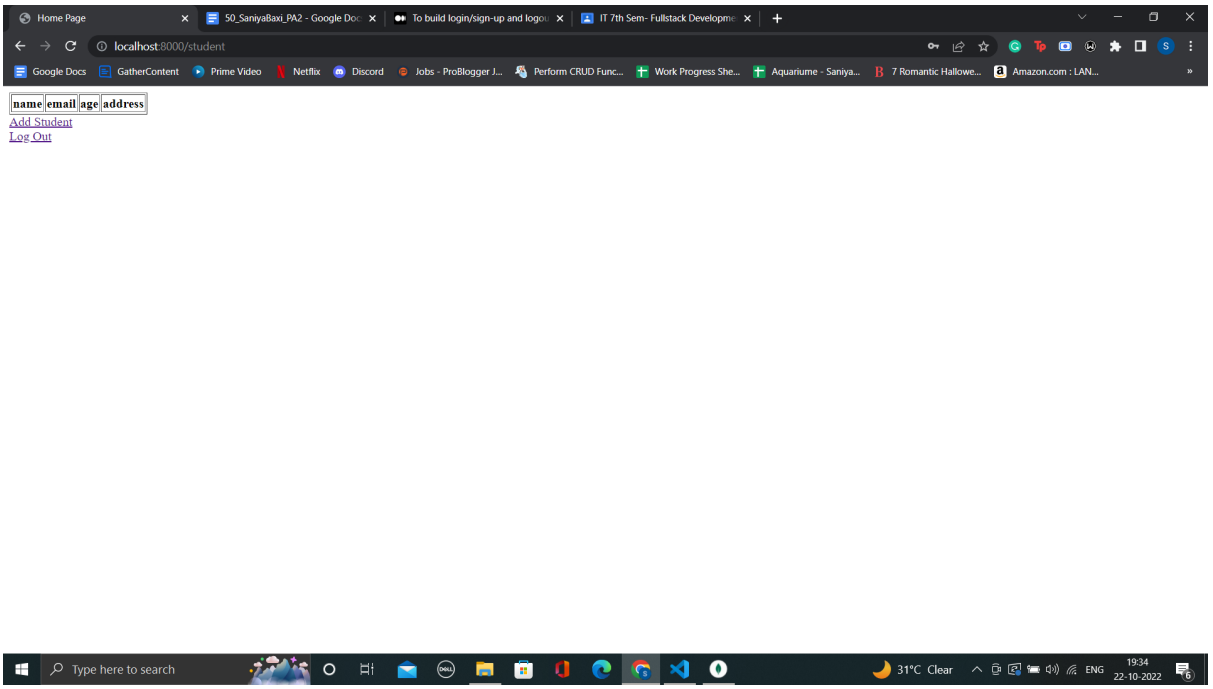
```
    if (token) {
        //verify token
        if (jwt.verify(token, process.env.TOKEN_SECRET)) {
            return res.redirect('/student')
        } else {
            return res.status(401).end('Unauthorized access');
        }
    }
    else {
        //redirect to login
        return res.render('login')
    }
})

app.post('/login', async (req, res) => {
    let { username, password } = req.body
    if (username && password) {
        try {
            let result = await studentModel.find({ name: username,
password: password })
           if (result.length === 1) {
                //Generate Token
                const accessToken = jwt.sign({ name: username,
password: password }, accessTokenSecret)
                res.cookie('token', accessToken, { httpOnly: true });
                return res.redirect('/student')

           } else {
               res.render('login', { errors: { validationError: 'Enter
Valid Username Or Password' } })
            }
        } catch (error) {
            res.render('login', { errors: { validationError: 'There Was
Problem While Log In.' } })
        }
    } else {
        res.render('login', { errors: { AllFieldsError: 'Please Enter
Username And Password' } })
    }
})

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

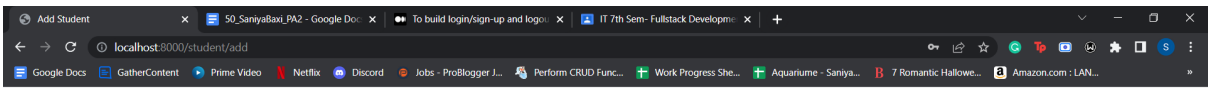
```
    res.render('login')
})

app.listen(PORT, () => {
    console.log(`app listening on http://localhost:${PORT}`)
})
```

Screenshots:



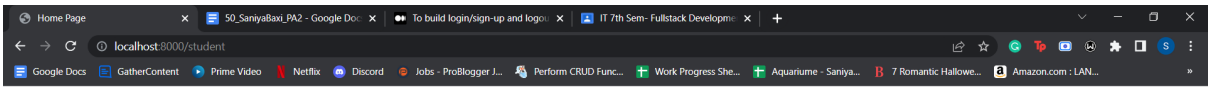

Log In Page
Adding User Login & JWT Signin
50_SaniyaBaxi_PA2 - Google Doc
To build login/sign-up and logou
IT 7th Sem- Fullstack Developme
localhost:8000/logout
Google Docs
GatherContent
Prime Video
Netflix
Discord
Jobs - ProBlogger J...
Perform CRUD Func...
Work Progress She...
Aquariume - Saniya...
7 Romantic Hallowe...
Amazon.com : LAN...
Log In
saniyab
enter password
Please fill out this field.
Log In
Type here to search
31°C Clear
ENG
19:31
22-10-2022


Home Page
50_SaniyaBaxi_PA2 - Google Doc
To build login/sign-up and logou
IT 7th Sem- Fullstack Developme
localhost:8000/student
Google Docs
GatherContent
Prime Video
Netflix
Discord
Jobs - ProBlogger J...
Perform CRUD Func...
Work Progress She...
Aquariume - Saniya...
7 Romantic Hallowe...
Amazon.com : LAN...
name
email
age
address
Add Student
Log Out
Type here to search
31°C Clear
ENG
19:34
22-10-2022



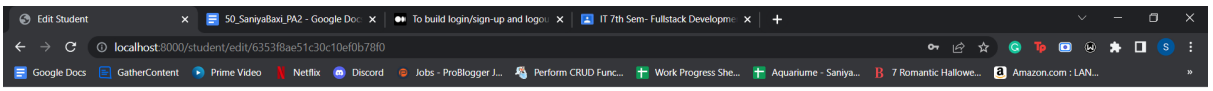
Add Student

Name: Saniya
Password: ••••••••
Email: saniya@gmail.com
Address chowk
Age: 21
Add
Log Out



name	email	age	address	
Saniya	saniya@gmail.com	21	chowk	edit delete view

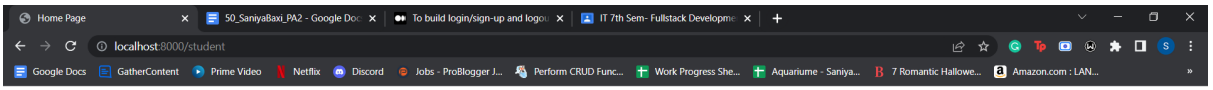
Add Student
Log Out



Edit Student

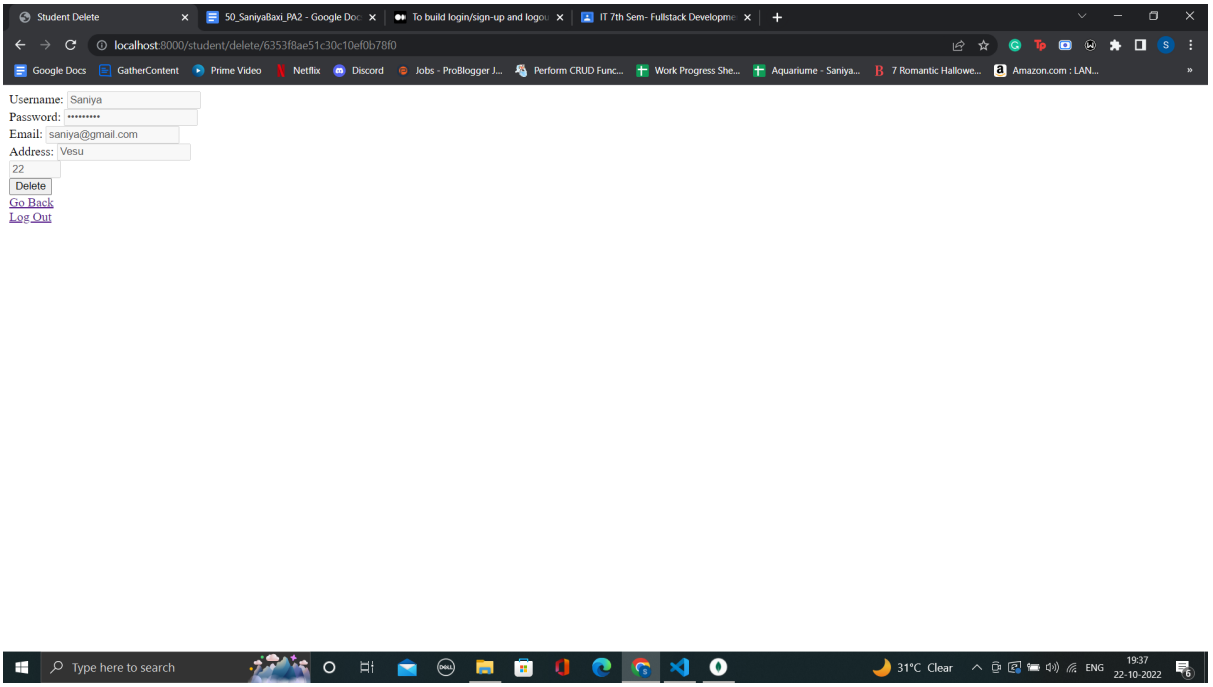
Name: Saniya
Password: ••••••••
Email: saniya@gmail.com
Address Vesu
Age: 22 Update

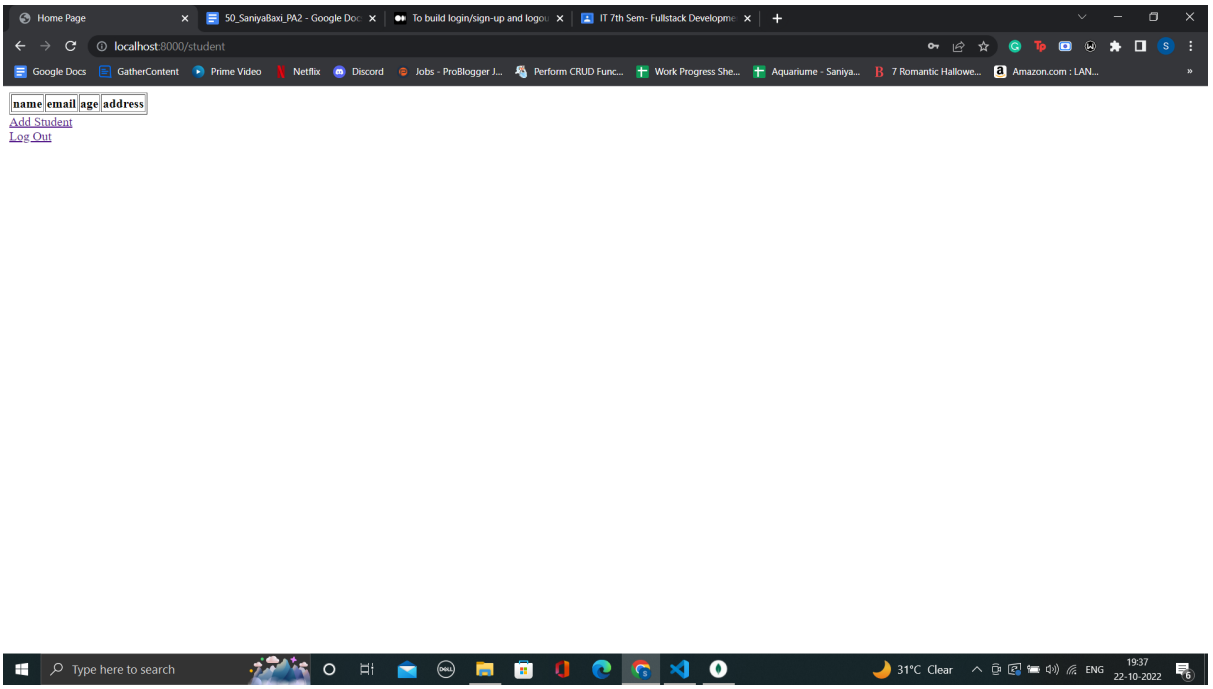
Log Out

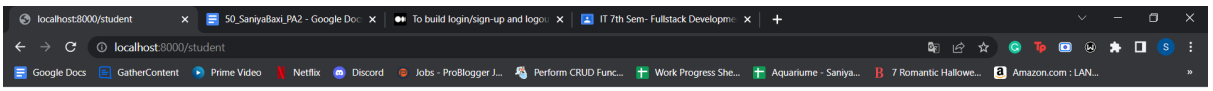


name	email	age	address	
Saniya	saniya@gmail.com	22	Vesu	edit delete view

Add Student
Log Out


Student Delete
50_SaniyaBaxi_PA2 - Google Doc
To build login/sign-up and logou
IT 7th Sem- Fullstack Developme
localhost:8000/student/delete/6353f8ae51c30c10ef0b78f0
Google Docs
GatherContent
Prime Video
Netflix
Discord
Jobs - ProBlogger J...
Perform CRUD Func...
Work Progress She...
Aquariume - Saniya...
7 Romantic Hallowe...
Amazon.com : LAN...
Username: Saniya
Password: ••••••••
Email: saniya@gmail.com
Address: Vesu
22
Delete
Go Back
Log Out
Type here to search
31°C Clear
ENG
19:37
22-10-2022


Home Page
50_SaniyaBaxi_PA2 - Google Doc
To build login/sign-up and logou
IT 7th Sem- Fullstack Developme
localhost:8000/student
Google Docs
GatherContent
Prime Video
Netflix
Discord
Jobs - ProBlogger J...
Perform CRUD Func...
Work Progress She...
Aquariume - Saniya...
7 Romantic Hallowe...
Amazon.com : LAN...
name
email
age
address
Add Student
Log Out
Type here to search
31°C Clear
ENG
19:37
22-10-2022



- Without login, a token won't be generated, and error will be shown while accessing other routes.

4. Login (JWT, CRUD, Mongoose, Express, Frontend, Logout)

- Auth Folder - authenticate.js

```
const jwt = require('jsonwebtoken')

const sign = (data) => {
    const token = jwt.sign({ ...data }, process.env.TOKEN_SECRET, {
algorithm: 'HS256', expiresIn: process.env.EXPIRY })
    return token
}

const verify = (req, res, next) => {
    let token = req.headers["authorization"] ||
req.headers["x-access-token"];
    token = token?.split(" ")[1]
    if (token) {
        try {
            if (jwt.verify(token, process.env.TOKEN_SECRET)) {
                return next()
            } else {
                return res.status(401).send('Unauthorized Access')
```

```
            }
        } catch (JsonWebTokenError) {
            return res.status(401).send('Something went wrong!');
        }
    } else {
        return res.status(401).send('Unauthorized Access')
    }
}

module.exports = { sign, verify }
```

- Model folder - studentModels.js

```
const mongoose = require('mongoose')
const studentSchema = new mongoose.Schema({
    name: String,
    email: String,
    password: String,
    age: Number,
    gender: String,
    city: String
})

module.exports = mongoose.model('student', studentSchema, 'students')
```

- StudentRoute.js

```
const express = require('express')
const app = express.Router()

//model
const student = require('../models/studentModel')

//getting all students
app.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')
    }
})

//getting single student
app.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')
    }
})

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

//updating student
app.put('/:id', async (req, res) => {
    try {
        const reqData = req.body
        const id = req.params.id
        const response = await student.findByIdAndUpdate(id, reqData)
        if (response) {
            return res.json("Student updated successfully")
```

```
        } else {
            return res.status(404).send('No Data Found For Given Id')
        }
    } catch (error) {
        console.error(error);
        res.status(500).send('Something went wrong')
    }
})

//deleting student
app.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 = app
```

- Index.js

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

//router
const studentRouter = require('./routes/studentRoute')

//model
const student = require('./models/studentModel')
```

```
//config
const PORT = process.env.PORT

//cors
app.use(cors())

//body-parser and json for sending json and reading body
app.use(express.json())
app.use(express.urlencoded({ extended: true }))

app.get('/', (req, res) => {
    console.log('default response')
    res.send('DEFAULT RESPONSE FROM SERVER')
})

//login request
app.post('/login', async (req, res) => {
    let { username, password } = req.body
    username = username
    password = password
    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 username or password')
        }
    } else {
        return res.status(401).json('Please provide username and
password')
    }
})

app.use('/get-token', sign)

app.use('/student', [verify], studentRouter)

app.listen(PORT, () => {
    console.log(`app listening on http://localhost:${PORT}`);
})
```

- Views (login, logout, add, delete, update, view)

- Login.html

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Log In Page</title>
    <style>
        .mb-2 {
            margin-bottom: 2rem;
        }

        .mt-2 {
            margin-top: 2rem;
        }

        .flex-center {
            display: flex;
            justify-content: center;
            flex-direction: column;
            align-items: center;
        }

        * {
            font-size: 1.2rem;
        }

        .err {
            color: red;
        }
    </style>
</head>

<body>
    <div class="container flex-center">
        <div class="header">
            <h3>Log In</h3>
        </div>
        <div class="err">
```

```
        </div>
        <div class="form-container mt-2 ">
            <div class="flex-center">
                <div class="username mb-2">
                    <input type="text" placeholder="enter username"
name="username" id="txtusername" required>
                </div>
                <div class="password mb-2">
                    <input type="password" name="password"
placeholder="enter password" id="txtpassword" required>
                </div>
                <div class="submit">
                    <input type="submit" value="Log In"
onclick="handleLogin()">
                </div>
            </div>
        </div>
    </div>
</body>

<script>
    const usernameField = document.getElementById("txtusername")
    const passwordField = document.getElementById("txtpassword")

    const handleLogin = async () => {
        let username = usernameField.value
        let password = passwordField.value
        postData('http://localhost:8000/login', { username, password })
            .then((data) => {
              if (data) {
                    localStorage.setItem('authToken', data)
                    window.location.replace('./index.html')
                }
            }).catch(error => {
                console.log(error)
            });
    }

    async function postData(url = '', data = {}) {
        // Default options are marked with *
        try {
            const response = await fetch(url, {
                method: 'POST', // *GET, POST, PUT, DELETE, etc.
```

```
                mode: 'cors', // no-cors, *cors, same-origin
                cache: 'no-cache', // *default, no-cache, reload,
force-cache, only-if-cached
                credentials: 'same-origin', // include, *same-origin,
omit
                headers: {
                    'Content-Type': 'application/json'
                    // 'Content-Type':
'application/x-www-form-urlencoded',
                },
                redirect: 'follow', // manual, *follow, error
                referrerPolicy: 'no-referrer', // no-referrer,
*no-referrer-when-downgrade, origin, origin-when-cross-origin,
same-origin, strict-origin, strict-origin-when-cross-origin, unsafe-url
                body: JSON.stringify(data) // body data type must match
"Content-Type" header
            });
            if (response.status === 200) {
                return response.json(); // parses JSON response into
native JavaScript objects
            } else {
                let msg = await response.json()
                alert(msg)
            }
        } catch (error) {
            alert(error)
        }

    }

</script>

</html>
```

- Logout.html

```
<html>

<head>
    <title>Log Out</title>
</head>

<body>
```

```
</body>
<script>
    window.localStorage.removeItem('authToken')
    window.location.replace('./login.html')
</script>

</html>
```

- Add.html

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <link rel="stylesheet" href="./style/style.css" type="text/css">
    <title>Add Student</title>
</head>

<body>
    <div class="container flex-center">
        <div class="header">
            <h3>Add Student</h3>
        </div>
        <div class="form-container mt-2 ">
            <div class="flex-center">
                <div class="username mb-2">
                    <input type="text" placeholder="enter your name"
id="txtname" name="name" required>
                </div>
                <div class="password mb-2">
                    <input type="password" name="password"
id="txtpassword" placeholder="enter password" id="" required>
                </div>
                <div class="email mb-2">
                    <input type="email" name="email" id="txtemail"
placeholder="enter your email" required>
                </div>
                <div class="city mb-2">
```

```
                    <input type="text" name="city" id="txtcity"
placeholder="enter your city" required>
                </div>
                <div class="age mb-2">
                    <input type="number" name="age" id="txtage"
min="10" max="80" placeholder="Age here" required>
                </div>
                <div class="gender mb-2">
                    <input type="radio" name="gender" value="Male"
id="txtmale" checked>Male
                    <input type="radio" name="gender" value="Female"
id="txtfemale">Female
                </div>
                <div class="submit">
                    <button style="padding:0.4rem"
onclick="handleAddStudent()">Add </button>
                </div>
            </div>
        </div>
    </div>
    <div class="links mt-2">
        <a href="./logout.html">Log Out</a>
    </div>
</body>

<script>
    const username = document.getElementById("txtname")
    const password = document.getElementById("txtpassword")
    const age = document.getElementById("txtage")
    const gender = document.getElementsByName("gender")[0]
    const male = document.getElementById("txtmale")
    const female = document.getElementById("txtfemale")
    const city = document.getElementById("txtcity")
    const email = document.getElementById("txtemail")

    const url = `http://localhost:8000/student`

    async function postData(url = '', data = {}) {
        // Default options are marked with *
        const response = await fetch(url, {
            method: 'POST', // *GET, POST, PUT, DELETE, etc.
            mode: 'cors', // no-cors, *cors, same-origin
```

```
            cache: 'no-cache', // *default, no-cache, reload,
force-cache, only-if-cached
            credentials: 'same-origin', // include, *same-origin, omit
            headers: {
                'Content-Type': 'application/json',
                'Authorization': `Bearer
${localStorage.getItem('authToken')}`
            },
            redirect: 'follow', // manual, *follow, error
            referrerPolicy: 'no-referrer', // no-referrer,
*no-referrer-when-downgrade, origin, origin-when-cross-origin,
same-origin, strict-origin, strict-origin-when-cross-origin, unsafe-url
            body: JSON.stringify(data) // body data type must match
"Content-Type" header
        });
        return response.json(); // parses JSON response into native
JavaScript objects
    }

    const handleAddStudent = () => {
        const payload = {
            name: username.value,
            email: email.value,
            password: password.value,
            age: age.value,
            gender: male.checked ? 'Male' : 'Female',
            city: city.value
        }
        postData(`${url}`, payload).then(response => {
            alert(response)
            window.location.replace('./index.html')
        })
    }
</script>

</html>
```

- Delete.html

```
<!DOCTYPE html>
```

```
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Delete Student</title>
</head>

<body>
    <div class="student-container" id="studentContainer">
    </div>
    <div class="btn">
        <button onclick="handleDelete()">Confirm Delete</button>
    </div>
    <div class="links">
        <a href="./student/">Go Back</a>
    </div>
    <div class="links mt-2">
        <a href="./logout.html">Log Out</a>
    </div>
</body>
<script>
    const studentContainer =
document.getElementById('studentContainer')
    const searchUrl = window.location.search
    id = searchUrl?.split("?")[1]?.split("=")[1]
    const url = `http://localhost:8000/student/${id}`
    const fetchStudent = async (id) => {
        const response = await fetch(url, {
            headers: {
                'Authorization': `Bearer
${localStorage.getItem('authToken')}`
            }
        })
        return response.json()
    }

    async function deleteData(url = '', data = {}) {
        // Default options are marked with *
        const response = await fetch(url, {
            method: 'DELETE', // *GET, POST, PUT, DELETE, etc.
```

```
            mode: 'cors', // no-cors, *cors, same-origin
            cache: 'no-cache', // *default, no-cache, reload,
force-cache, only-if-cached
            credentials: 'same-origin', // include, *same-origin, omit
            headers: {
                'Content-Type': 'application/json',
                'Authorization': `Bearer
${localStorage.getItem('authToken')}` || ''
            },
            redirect: 'follow', // manual, *follow, error
            referrerPolicy: 'no-referrer', // no-referrer,
*no-referrer-when-downgrade, origin, origin-when-cross-origin,
same-origin, strict-origin, strict-origin-when-cross-origin, unsafe-url
        });
        return response.json(); // parses JSON response into native
JavaScript objects
    }

    const handleDelete = () => {
        deleteData(url, id).then(response => {
            alert(response)
            window.location.replace('./index.html')
        })
    }

    if (id) {
        fetchStudent(id).then(data => {
            studentContainer.innerHTML = `
        <p>name: ${data.name}</p>
        <p>email: ${data.email}</p>
        <p>age: ${data.age}</p>
        <p>city: ${data.city}</p>
        <p>gender: ${data.gender}</p>
        `
        })
    } else {
        alert("Please provide student id")
    }
</script>

</html>
```

- View.html

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Student Detail</title>
</head>

<body>
    <div class="student-container" id="studentContainer">
    </div>
    <div class="links">
        <a href="./student/">Go Back</a>
    </div>
    <div class="links mt-2">
        <a href="./logout.html">Log Out</a>
    </div>
</body>
<script>
    const studentContainer =
document.getElementById('studentContainer')
    const fetchStudent = async (id) => {
        console.log('fetchStudent')
        const url = `http://localhost:8000/student/${id}`
        const response = await fetch(url, {
            headers: {
                'Authorization': `Bearer
${localStorage.getItem('authToken')}`
            }
        })
        return response.json()
    }
    const searchUrl = window.location.search
    id = searchUrl?.split("?")[1]?.split("=")[1]
    if (id) {
        fetchStudent(id).then(data => {
            studentContainer.innerHTML = `
        <p>name: ${data.name}</p>
        <p>email: ${data.email}</p>
        <p>age: ${data.age}</p>
```

```
        <p>city: ${data.city}</p>
        <p>gender: ${data.gender}</p>
        `
        })
    } else {
        alert("Please provide student id")
    }

</script>

</html>
```

- Update.html

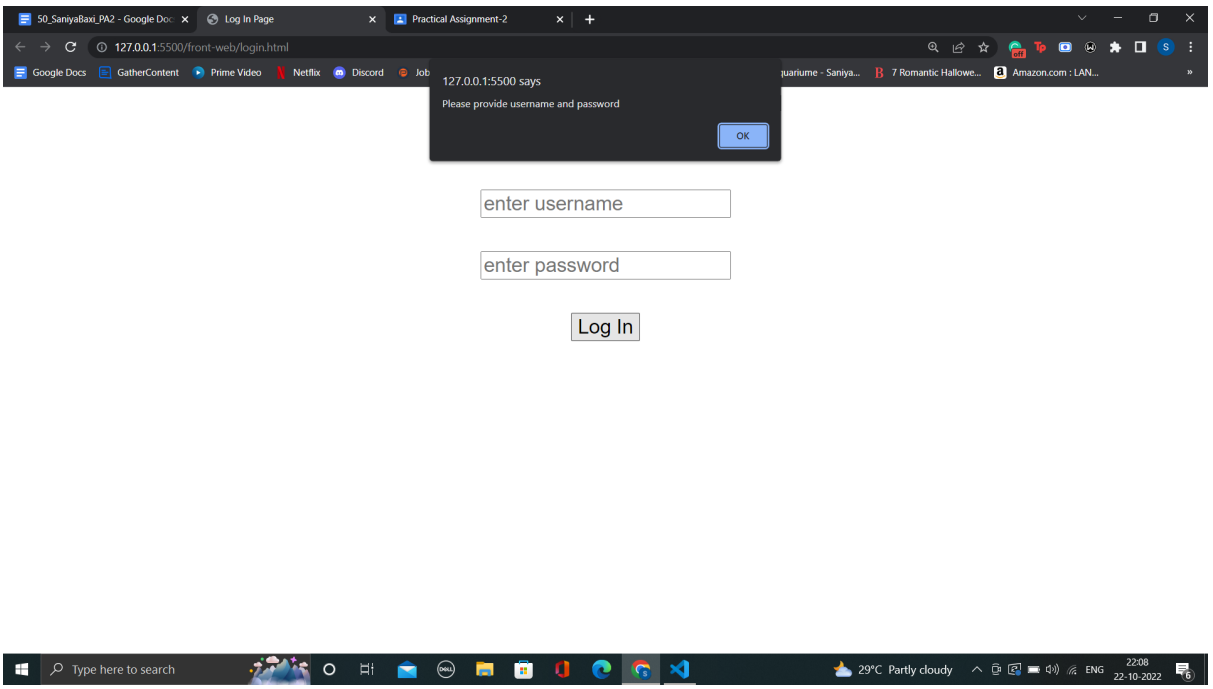
```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Edit Student</title>
    <link rel="stylesheet" href="./style/style.css" type="text/css">

</head>

<body>
    <div class="container flex-center">
        <div class="header">
            <h3>Edit Student</h3>
        </div>

        <div class="form-container mt-2 ">
            <div class="flex-center">
                <input type="hidden" name="_id">

                <div class="username mb-2">
                    <input type="text" placeholder="enter your name"
name="name" id="txtname" required>
                </div>
                <div class="password mb-2">
```

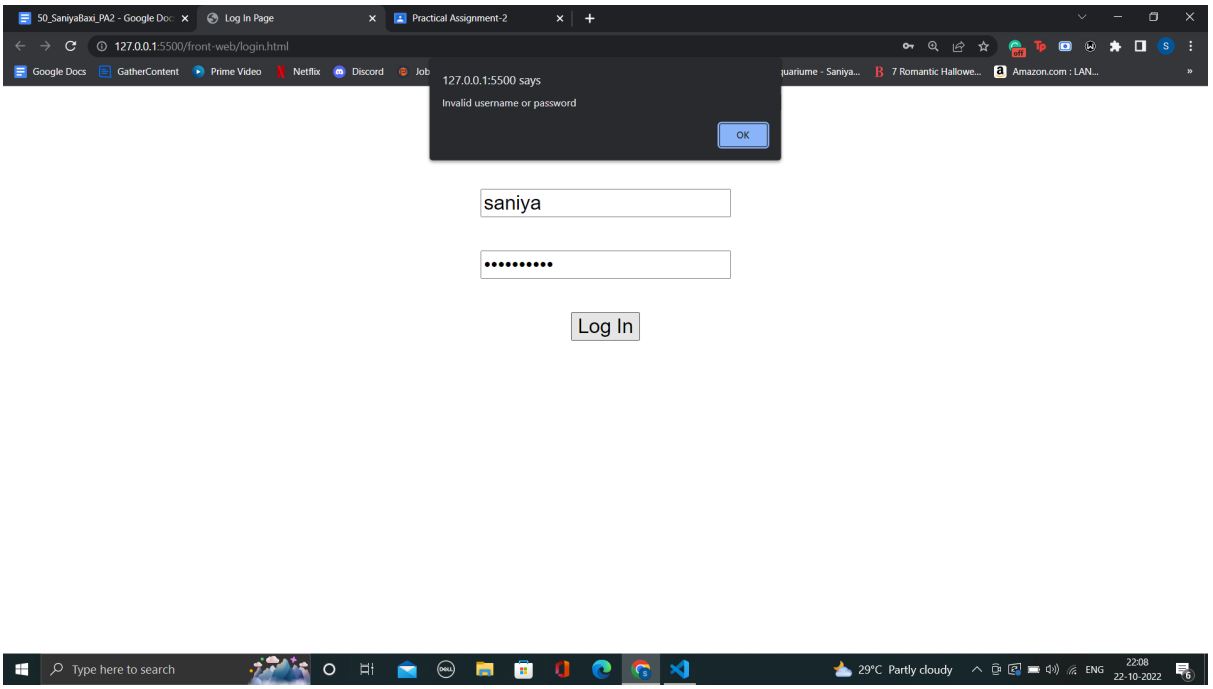
```
                    <input type="password" name="password"
placeholder="enter password" id="txtpassword" required>
                </div>
                <div class="email mb-2">
                    <input type="email" name="email" placeholder="enter
your email" id="txtemail" required>
                </div>
                <div class="city mb-2">
                    <input type="text" name="city" placeholder="enter
your city" id="txtcity" required>
                </div>
                <div class="age mb-2">
                    <input type="number" name="age" min="10" max="80"
placeholder="Age here" id="txtage" required>
                </div>
                <div class="gender mb-2">
                    <input type="radio" name="gender" value="Male"
id="txtmale">Male
                    <input type="radio" name="gender" value="Female"
id="txtfemale">Female
                </div>
                <div class="submit">
                    <button style="padding:0.4rem"
onclick="handleUpdate()">Update</button>
                </div>
            </div>
        </div>
    </div>
    <div class="links mt-2">
        <a href="./logout.html">Log Out</a>
    </div>
</body>

<script>
    const username = document.getElementById("txtname")
    const password = document.getElementById("txtpassword")
    const age = document.getElementById("txtage")
    const gender = document.getElementsByName("gender")[0]
    const male = document.getElementById("txtmale")
    const female = document.getElementById("txtfemale")
    const city = document.getElementById("txtcity")
    const email = document.getElementById("txtemail")
```

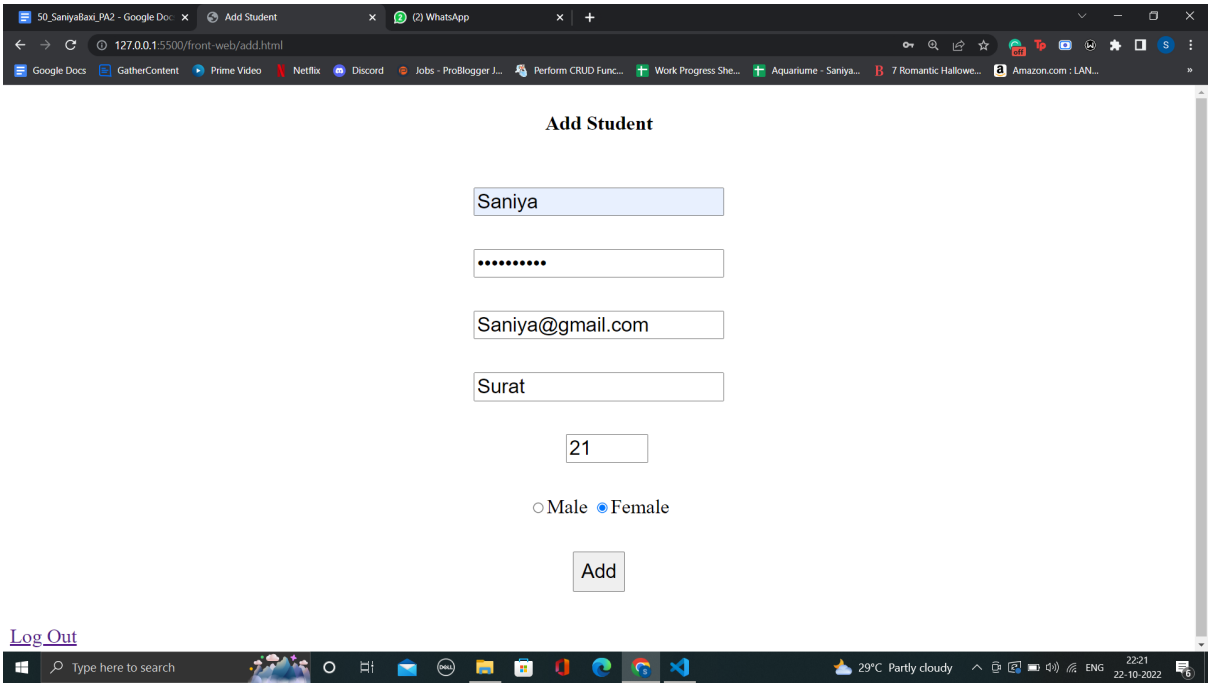
```
    const url = `http://localhost:8000/student`


    const searchUrl = window.location.search
    id = searchUrl?.split("?")[1]?.split("=")[1]

    const fetchStudent = async (id) => {
        const response = await fetch(`${url}/${id}`, {
            headers: {
                'Authorization': `Bearer
${localStorage.getItem('authToken')}`
            }
        })
        return response.json()
    }

    async function putData(url = '', data = {}) {
        // Default options are marked with *
        const response = await fetch(url, {
            method: 'PUT', // *GET, POST, PUT, DELETE, etc.
            mode: 'cors', // no-cors, *cors, same-origin
            cache: 'no-cache', // *default, no-cache, reload,
force-cache, only-if-cached
            credentials: 'same-origin', // include, *same-origin, omit
            headers: {
                'Content-Type': 'application/json',
                'Authorization': `Bearer
${localStorage.getItem('authToken')}`
            },
            redirect: 'follow', // manual, *follow, error
            referrerPolicy: 'no-referrer', // no-referrer,
*no-referrer-when-downgrade, origin, origin-when-cross-origin,
same-origin, strict-origin, strict-origin-when-cross-origin, unsafe-url
            body: JSON.stringify(data) // body data type must match
"Content-Type" header
        });
        return response.json(); // parses JSON response into native
JavaScript objects
    }


    const handleUpdate = () => {
        const payload = {
```

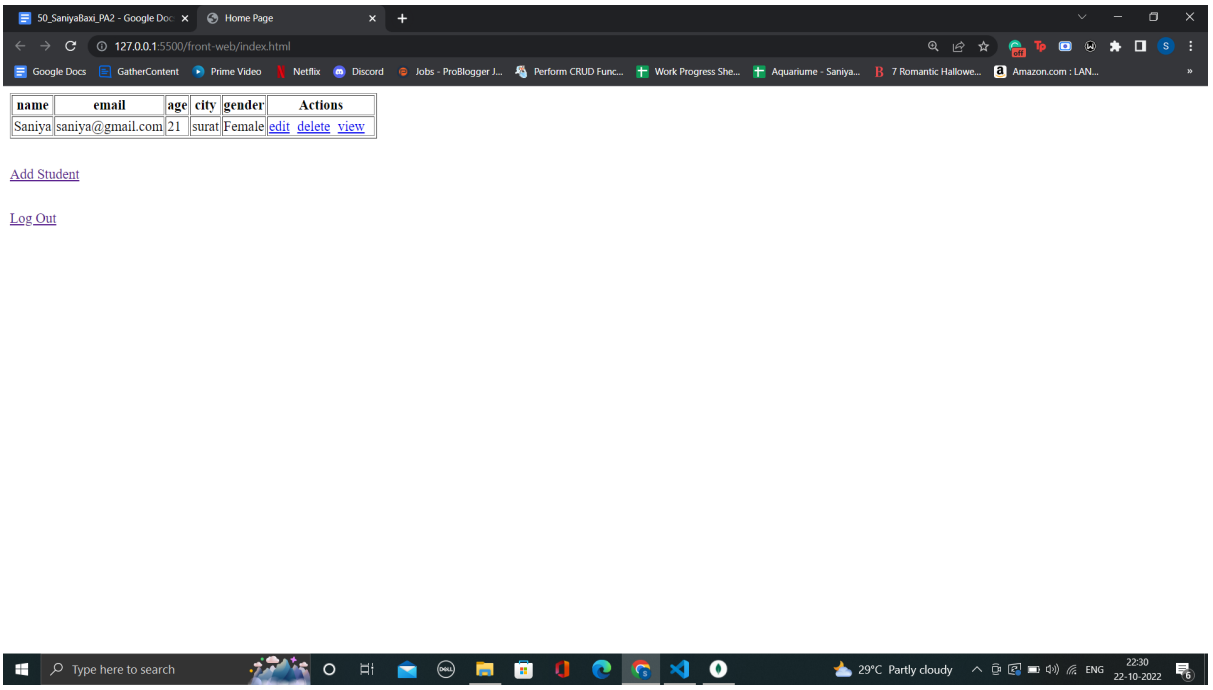
```
            name: username.value,
            email: email.value,
            password: password.value,
            age: age.value,
            gender: male.checked ? 'Male' : 'Female',
            city: city.value
        }
        putData(`${url}/${id}`, payload).then(response => {
            alert(response)
            window.location.replace('./index.html')
        })
    }

    if (id) {
        fetchStudent(id).then(data => {
            username.value = data.name
            email.value = data.email
            password.value = data.password
            city.value = data.city
            age.value = data.age
            data.gender.toLowerCase() === 'male' ? male.checked = true
: female.checked = true
        })
    } else {
        alert("Please provide student id")
    }
</script>

</html>
```

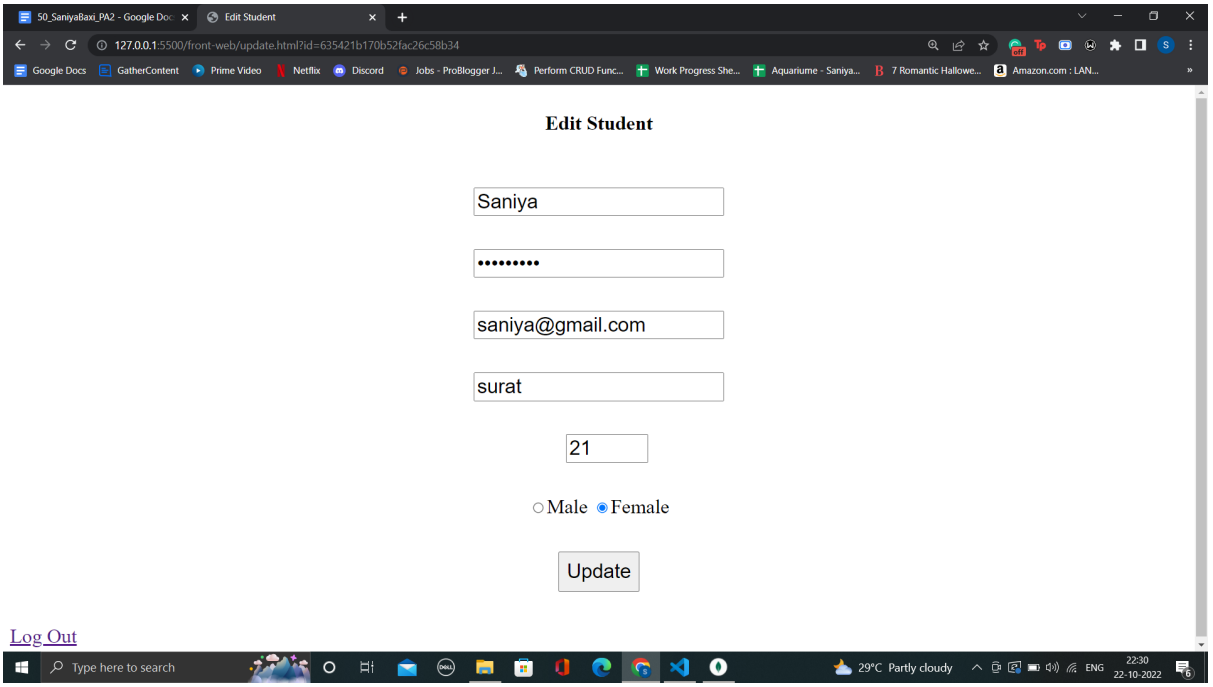
Screenshots:

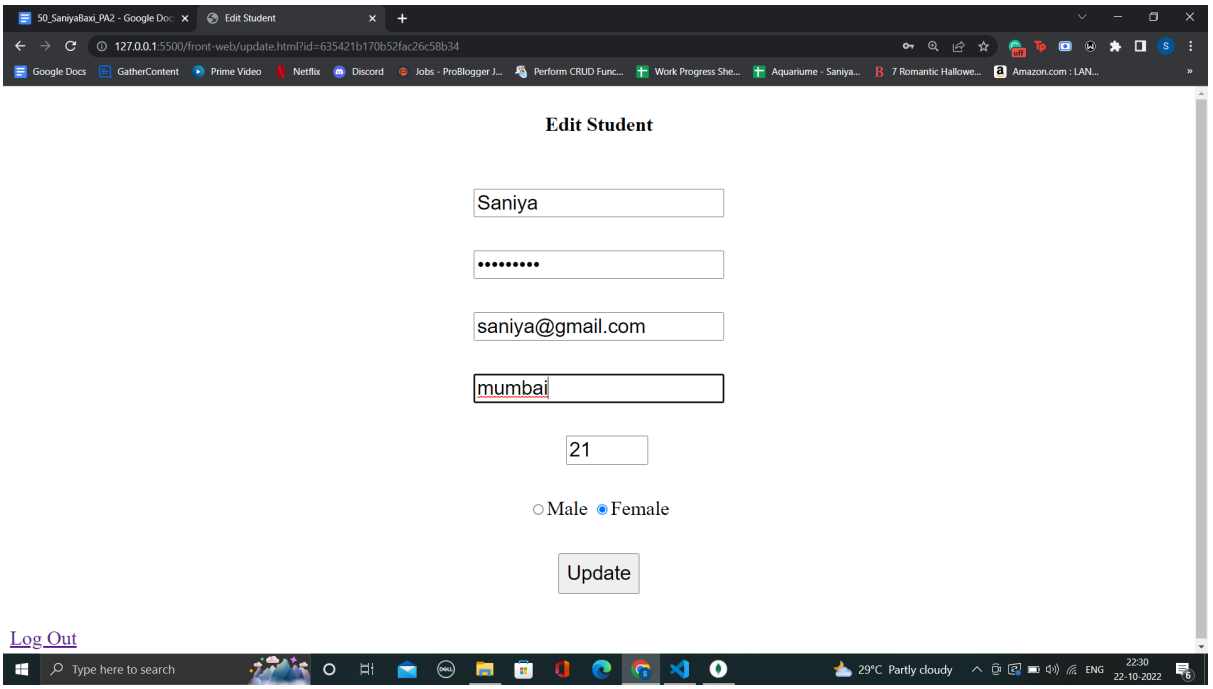

50_SaniyaBaxi_PA2 - Google Doc
Log In Page
Practical Assignment-2
127.0.0.1:5500/front-web/login.html
Google Docs
GatherContent
Prime Video
Netflix
Discord
127.0.0.1:5500 says
Please provide username and password
OK
enter username
enter password
Log In
Type here to search
29°C Partly cloudy
ENG
22:08
22-10-2022

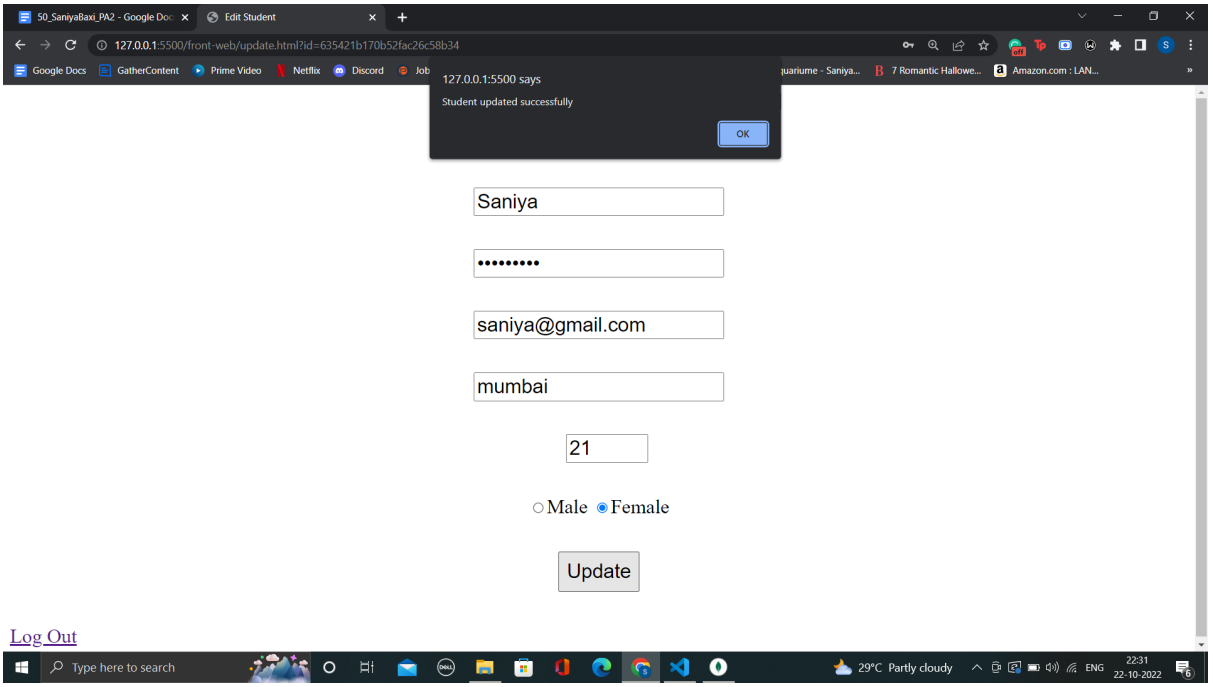

50_SaniyaBaxi_PA2 - Google Doc
Log In Page
Practical Assignment-2
127.0.0.1:5500/front-web/login.html
Google Docs
GatherContent
Prime Video
Netflix
Discord
127.0.0.1:5500 says
Invalid username or password
OK
saniya
Log In
Type here to search
29°C Partly cloudy
ENG
22:08
22-10-2022

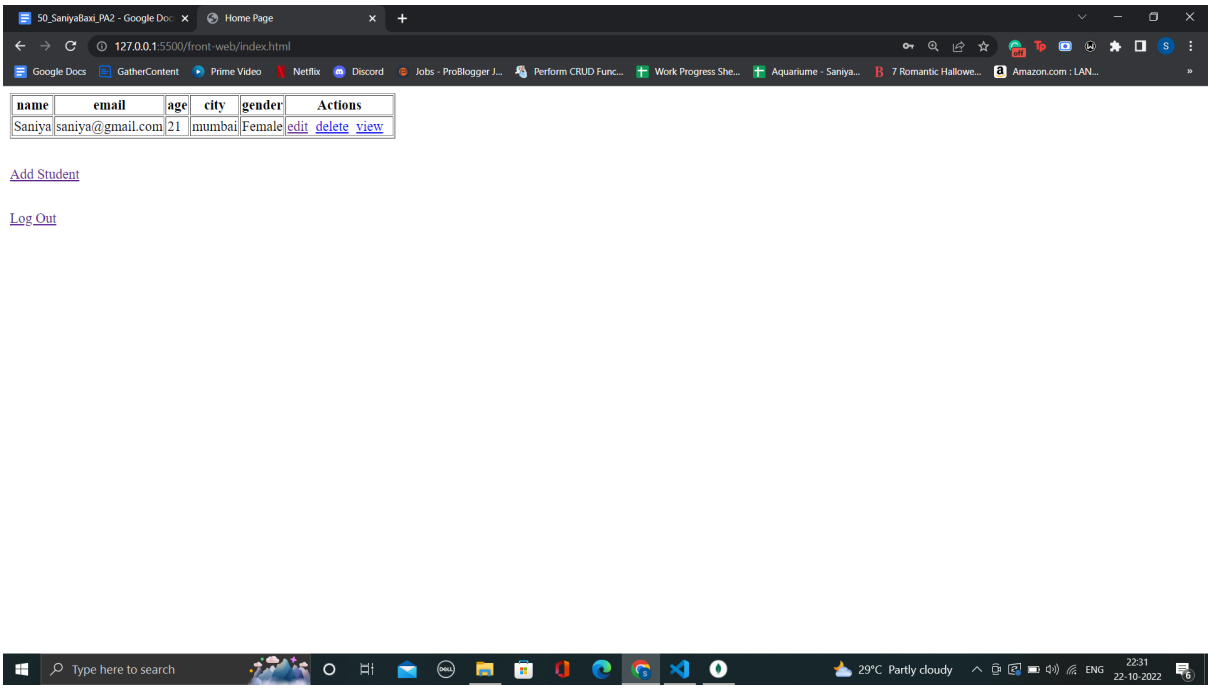

50_SaniyaBaxi_PA2 - Google Doc
Add Student
(2) WhatsApp
127.0.0.1:5500/front-web/add.html
Google Docs
GatherContent
Prime Video
Netflix
Discord
Jobs - ProBlogger J...
Perform CRUD Func...
Work Progress She...
Aquariume - Saniya...
7 Romantic Hallowe...
Amazon.com : LAN...
Add Student
Saniya
Saniya@gmail.com
Surat
21
Male
Female
Add
Log Out
Type here to search
29°C Partly cloudy
ENG
22:21
22-10-2022

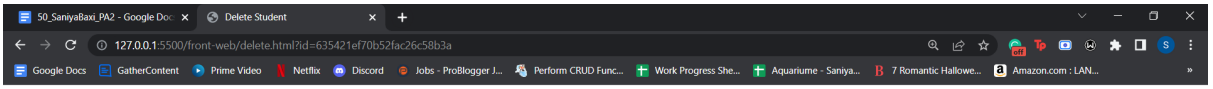

50_SaniyaBaxi_PA2 - Google Doc
Home Page
127.0.0.1:5500/front-web/index.html
Google Docs
GatherContent
Prime Video
Netflix
Discord
Jobs - ProBlogger J...
Perform CRUD Func...
Work Progress She...
Aquariume - Saniya...
7 Romantic Hallowe...
Amazon.com : LAN...
name
email
age
city
gender
Actions
Saniya
saniya@gmail.com
21
surat
Female
edit
delete
view
Add Student
Log Out
Type here to search
29°C Partly cloudy
ENG
22:30
22-10-2022


50_SaniyaBaxi_PA2 - Google Doc
Edit Student
127.0.0.1:5500/front-web/update.html?id=635421b170b52fac26c58b34
Google Docs
GatherContent
Prime Video
Netflix
Discord
Jobs - ProBlogger J...
Perform CRUD Func...
Work Progress She...
Aquariume - Saniya...
7 Romantic Hallowe...
Amazon.com : LAN...
Edit Student
Saniya
saniya@gmail.com
surat
21
Male
Female
Update
Log Out
Type here to search
29°C Partly cloudy
ENG
22:30
22-10-2022


50_SaniyaBaxi_PA2 - Google Doc
Edit Student
127.0.0.1:5500/front-web/update.html?id=635421b170b52fac26c58b34
Google Docs
GatherContent
Prime Video
Netflix
Discord
Jobs - ProBlogger J...
Perform CRUD Func...
Work Progress She...
Aquariume - Saniya...
7 Romantic Hallowe...
Amazon.com : LAN...
Edit Student
Saniya
saniya@gmail.com
mumbai
21
Male
Female
Update
Log Out
Type here to search
29°C Partly cloudy
ENG
22:30
22-10-2022


50_SaniyaBaxi_PA2 - Google Doc
Edit Student
127.0.0.1:5500/front-web/update.html?id=635421b170b52fac26c58b34
Google Docs
GatherContent
Prime Video
Netflix
Discord
127.0.0.1:5500 says
Student updated successfully
OK
Saniya
saniya@gmail.com
mumbai
21
Male
Female
Update
Log Out
Type here to search
29°C Partly cloudy
ENG
22:31
22-10-2022


50_SaniyaBaxi_PA2 - Google Doc
Home Page
127.0.0.1:5500/front-web/index.html
Google Docs
GatherContent
Prime Video
Netflix
Discord
name
email
age
city
gender
Actions
Saniya
saniya@gmail.com
21
mumbai
Female
edit
delete
view
Add Student
Log Out
Type here to search
29°C Partly cloudy
ENG
22:31
22-10-2022



name: nikita

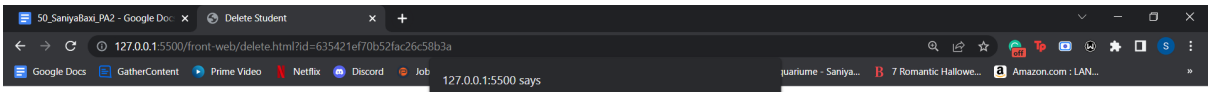
email: nikita@gmail.com

age: 20

city: surat

gender: Female

Confirm Delete
Go Back
Log Out



name: nikita

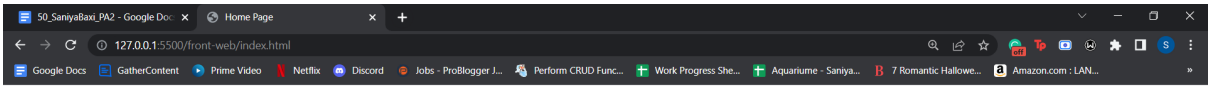
email: nikita@gmail.com

age: 20

city: surat

gender: Female

Confirm Delete
Go Back
Log Out



name	email	age	city	gender	Actions
Saniya	saniya@gmail.com	21	mumbai	Female	edit delete view

Add Student

Log Out