Name: Parmar Henisha Manishkumar

Roll No : 28

Class: MSc(ICT) – Sem 3

Subject: 304 – Open Source Web Development

Git Url: https://github.com/HenishaTechie/Node Assignment2

Practical Assignment2

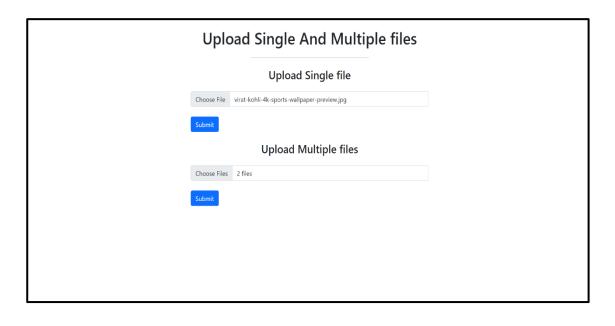
1. Develop a user registration form and store its data in any database using Express. Form should also contain file upload (single, multiple) with validations.

index.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">
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC"
crossorigin="anonymous">
    <title>Upload single/multiple image</title>
    <style type="text/css">
        form{
     width: 50%;
     margin: 10px 25%;
    </style>
</head>
<body>
    <div class="container">
         <div class="single-upload">
                <div class="text">
```

```
<h1 class="text-center my-4">Upload Single And Multiple
files</h1>
                    <hr class="w-25 m-auto bg-dark">
                </div>
                <h3 class="text-center my-4">Upload Single file</h3>
            <form action="/upload_single" method="post" enctype="multipart/form-</pre>
data">
                <input type="file" class="form-control" name="userFile" id=""><br>
                <input type="submit" class="btn btn-primary" value="Submit">
            </form>
        </div>
        <div class="multi-upload">
            <div class="header-single">
                <h3 class="text-center my-4 mx-3">Upload Multiple files</h3>
            </div>
            <form action="/upload_multiple" method="post" enctype="multipart/form-</pre>
data">
                <input type="file" class="form-control" name="userFiles" id=""</pre>
multiple="multiple"><br>
                <input type="submit" class="btn btn-primary" value="Submit">
            </form>
        </div>
    </div>
</body>
</html>
                                   server.js
const express = require('express')
const app = express();
const multer = require('multer')
var fs = require('fs')
var path = require('path')
var Storage = multer.diskStorage({
    destination: (req, file, callback) => {
        const destFolder = './uploads'
        //check if folder exist or not
        if (fs.existsSync(destFolder)) {
            callback(null, destFolder)
        } else {
            fs.mkdir(destFolder, (err) => {
                err ? console.error(err.stack) : callback(null, destFolder)
            })
        }
    },
    filename: (req, file, callback) => {
        callback(null,
`${file.fieldname}_${Date.now()}${path.extname(file.originalname)}`)
    }
})
```

```
const acceptedTypes = ["image/jpeg", "image/jpg", "image/png", "image/svg",
"image/gif"]
var upload = multer({
    storage: Storage, fileFilter: (req, file, callback) => {
        if (acceptedTypes.includes(file.mimetype)) {
            callback(null, true)
        } else {
            callback(null, false)
            return callback(`only ${acceptedTypes.toString(',')} format allowed`)
        }
    }
})
app.post('/upload_single', upload.single('userFile'), (req, res) => {
    return res.send('file is uploaded');
})
app.post('/upload_multiple', upload.array('userFiles', 4), (req, res) => {
    return res.send('files uploaded successfully.');
})
app.get('/', (req, res) => {
    return res.sendFile(__dirname + '/views/index.html');
})
app.use(function (err, req, res, next) {
    if (err instanceof multer.MulterError) {
        console.log("ERRRR");
        res.status(500).send("file upload err " + err.message);
    }
    else
        next(err);
});
app.listen(8000, () => {
    console.log(`App listening`)
})
```



files uploaded successfully.	

2 Express Login application with file session store.

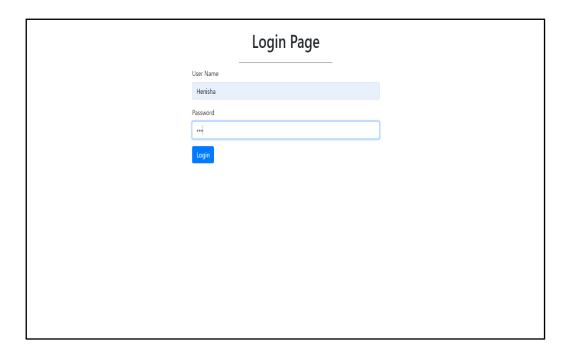
login.html

```
<!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">
    <link rel="stylesheet"</pre>
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
    <style type="text/css">
  form
  {
      width: 50%;
      margin: 10px 25%;
  }
    </style>
</head>
<body>
    <div class="container">
        <div class="text">
            <h1 class="text-center my-4">Login Page</h1>
            <hr class="w-25 m-auto bg-dark">
        </div>
        <form class="flex-center" action="/login" method="post">
            <div class="form-group">
                <label for="name">User Name</label>
                <input type="text" class="form-control" placeholder="Enter</pre>
Username" name="username" required>
            </div>
            <div class="form-group">
                <label for="name">Password</label>
                <input type="password" class="form-control" name="password"</pre>
placeholder="Enter Password" id="" required>
            </div>
            <div class="form-group">
                <input type="submit" value="Login" class="btn btn-primary">
            </div>
        </form>
    </div>
</body>
</html>
```

server.js

```
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' })
}))
// Define a list of valid users with their passwords
const validUsers = [
    { username: 'Henisha', password: '123' }
1;
app.get('/', (req, res, next) => {
    res.sendFile(__dirname + '/views/login.html')
})
app.post('/login', (req, res) => {
    const username = req.body.username;
    const password = req.body.password;
    // Check if provided username and password match any valid user
    const validUser = validUsers.find(user => user.username === username &&
user.password === password);
    if (validUser) {
        // Valid credentials, store data in session
        req.session.loggedIn = true;
        req.session.username = username;
        return res.redirect('/home');
    } else {
        return res.send('Invalid Username or Password!');
    }
})
app.get('/home', (req, res) => {
    if (req.session.loggedIn) {
        console.log('logged in')
        res.sendFile(__dirname + '/views/home.html')
    } else {
        console.log('not logged in')
        res.sendFile(__dirname + '/views/login.html')
```

```
}
})
app.get('/logout', (req, res) => {
    req.session.destroy();
    res.redirect('/');
});
app.listen(8080, () => {
    console.log(`server listening`)
})
                                     Home.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>Home Page</title>
</head>
<body>
    <div class="container">
        <h2>Home Page, You Are Authentic User</h2>
        <a href="/logout">Logout</a>
    </div>
</body>
</html>
```



Home Page, You Are Authentic User
Logout

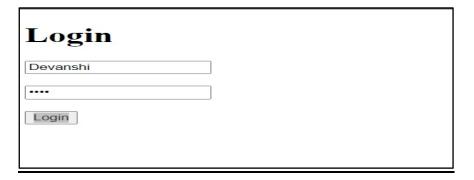
3.Express Login application with redis session store.

server.js

```
const express = require("express");
const session = require("express-session");
const { createClient } = require("redis");
const hbs = require("hbs")
const path = require("path")
const app = express();
// Initialize client.
const redisClient = createClient();
redisClient.connect().then(()=>{
   console.log("redis client is connected");
}).catch((error)=>{
    console.log("not connected ",error)
});
// Initialize store.
const RedisStore = require("connect-redis").default
const redisStore = new RedisStore({
  client: redisClient,
  prefix: "Expressredis:",
});
// Initialize session storage.
app.use(
  session({
    store: redisStore,
    resave: false,
    saveUninitialized: false,
    secret: "keyboard cat",
 })
);
app.use(express.json())
app.use(express.urlencoded({extended:true}))
app.set("views",path.join(__dirname,"/src/views"))
app.set("view engine",hbs)
const routes = require("./src/Routes/routes")
app.use("/",routes)
// Start the server
```

```
app.listen(3000, () => {
 console.log("Server started on port 3000");
});
                                routers => routers.js
const router = require("express").Router()
 router.get("/", (req, res) => {
    res.render("Login.hbs")
 });
 const isLoggedIn = (req, res, next) => {
    if (req.session.userId) {
     next();
   } else {
     res.redirect("/login");
    }
  };
  router.post("/login", (req, res) => {
   try {
        const password = req.body.password;
        req.session.userId = Date.now();
        return res.render("dashboard.hbs");
    } catch (error) {
       console.log(error);
   }
 });
 router.get("/logout", (req, res) => {
    // Destroy session and redirect to login page
    req.session.destroy();
    res.redirect("/");
 });
module.exports = router
                             Controller => Students.js
exports.Login= async(req,res) =>{
   try {
    } catch (error) {
}
                                 views=> login.hbs
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
```

```
</head>
<body>
     <h1>Login</h1>
      <form method="POST" action="/login">
        <input type="text" name="username" placeholder="Username" required> <br><br>
        <input type="password" name="password" placeholder="password"</pre>
required><br><br>
        <button type="submit">Login</button>
     </form>
</body>
</html>
                              views=> dashboard.hbs
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <h1>Dashboard</h1>
      Welcome, User!
      <a href="/logout">Logout</a>
</body>
</html>
```



Dashboard		
Welcome, User!		
Logout		

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

db => database.js

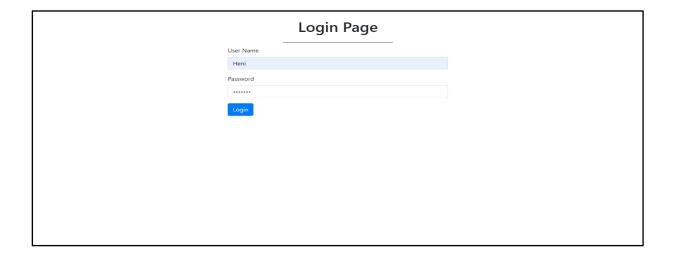
```
var mongoose = require('mongoose');
//database config
const server = "127.0.0.1:27017"
const Database = "studentDB"
class database {
    constructor() {
       this._connect()
    async _connect() {
        try {
            await mongoose.connect(`mongodb://${server}/${Database}`)
            console.log(`database connection successfull`)
        } catch (error) {
            console.error('Database connection error: ' + error);
        }
    }
}
module.exports = new database()
                                models => student.js
const mongoose = require('mongoose')
const studentSchema = new mongoose.Schema({
    name: String,
    email: String,
    password: String,
    age: Number,
    gender: String,
    city: String
})
module.exports = new mongoose.model('student', studentSchema, 'students')
                              middleware => token.js
const jwt = require('jsonwebtoken')
const signToken = async (req, res) => {
}
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 }
                                   index.js
const express = require('express')
const app = express()
const PORT = 8000
const database = require('./db/database')
const student = require('./routes/student')
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')
            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')
                // return res.render('index', { session: result[0], accessToken:
accessToken })
                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}`)
})
                                routes => student.js
const express = require('express');
const { verifyToken } = require('../middleware/token');
const student = express.Router();
const studentModel = require('../models/student')
student.use(verifyToken)
//create
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('There was problem creating students')
   }
})
//read
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('There was problem 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('There was problem getting students')
    }
})
//update
student.get('/edit/:id', async (req, res) => {
        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('There was problem updating students')
```

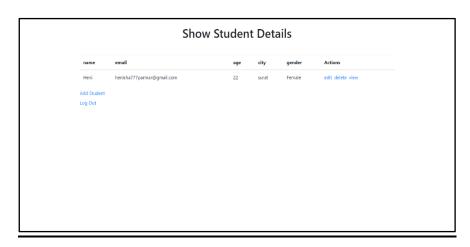
```
}
})
//delete
student.get('/delete/:id', async (req, res) => {
    try {
        const student = await studentModel.findById(req.params.id)
        student ? res.render('delete', { data: student }) : res.render('index', {
noDataError: 'student not found' })
    } catch (error) {
        res.send('There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



views => 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">
   <link rel="stylesheet"</pre>
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
</head>
<body>
   <div class="container">
      <div class="text">
          <h1 class="text-center my-4">Show Student Details</h1>
      </div><br>
      <div class="error">
          <% if(typeof(noDataError)!=='undefined' ) { %>
             <%= {noDataError} %>
             <%}%>
                <% if(typeof(accessToken)!=='undefined' ) { %>
                    <script>
                       localStorage.setItem('token', "<%=</pre>
JSON.stringify(accessToken) %>");
                    </script>
                    <%}%>
      </div>
      <div class="error">
          <% if(typeof(data)!=='undefined' ) { %>
             <thead>
                    name
                       email
                       age
                       city
                       gender
                       Actions
                    </thead>
                 <% data.forEach(student=>{
                       %>
                       >
                              <%= student.name %>
```

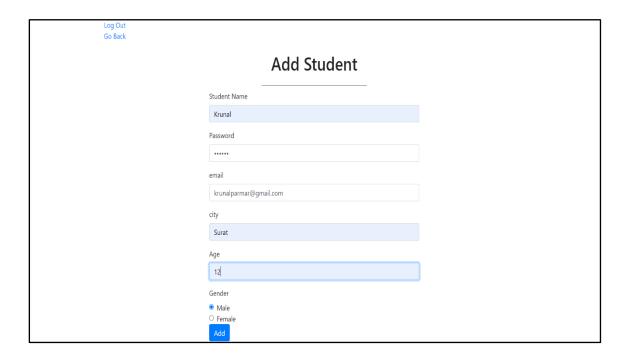
```
<%= student.email %>
                            <%= student.age %>
                            <%= student.city %>
                            >
                                <%= student.gender %>
                            >
                                <div class="mr-1">
                                   <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>
                            <% }) %>
                 <%}%>
                 <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.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">
    <!-- <link rel="stylesheet" href="/style.css" type="text/css"> -->
    <link rel="stylesheet"</pre>
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
    <title>Add Student</title>
    <style type="text/css">
        form
        {
            width: 50%;
            margin: 10px 25%;
          </style>
</head>
<body>
    <div class="container">
        <div class="links">
            <a href="/logout">Log Out</a>
        </div>
        <div class="links">
            <a href="/student/">Go Back</a>
        </div>
        <div class="text">
            <h1 class="text-center my-4">Add Student</h1>
            <hr class="w-25 m-auto bg-dark">
        </div>
        <div class="err">
            <% if(typeof errors!=='undefined' ){ %>
                >
                    <%= errors.validationError %>
                <% } %>
        </div>
        <form class="flex-center" action="/student/" method="post">
            <div class="form-group">
                <label for="name">Student Name</label>
```

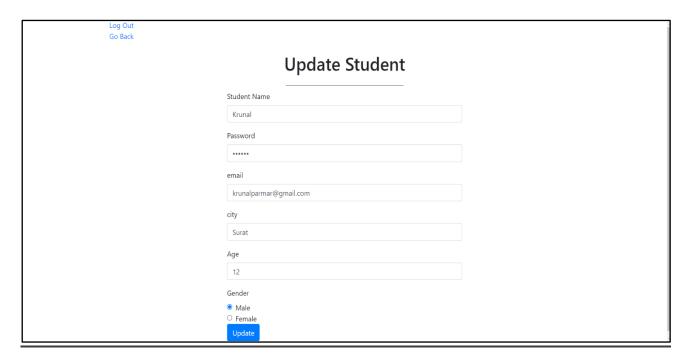
```
<input type="text" class="form-control" placeholder="Enter Student</pre>
name" name="name" required>
            </div>
            <div class="form-group">
                <label for="name">Password</label>
                <input type="password" class="form-control" name="password"</pre>
placeholder="Enter Password" id="" required>
            </div>
            <div class="form-group">
                <label for="name">email</label>
                <input type="email" class="form-control" name="email"</pre>
placeholder="Enter email" id="" required>
            </div>
            <div class="form-group">
                <label for="name">city</label>
                <input type="text" class="form-control" name="city"</pre>
placeholder="Enter city" id="" required>
            </div>
            <div class="form-group">
                <label for="name">Age</label>
                <input type="text" class="form-control" name="age" id="" min="10"</pre>
max="80" placeholder="Enter Age" id="" required>
            </div>
            <label for="name">Gender</label>
            <div class="form-check">
                 <input class="form-check-input" type="radio" name="gender"</pre>
value="Male" id="" checked>
                <label class="form-check-label" for="flexRadioDefault2">
                    Male
                </label>
              </div>
              <div class="form-check">
                <input class="form-check-input" type="radio" name="gender"</pre>
value="Female" id="">
                <label class="form-check-label" for="flexRadioDefault2">
                    Female
                </label>
              </div>
            <div class="form-group">
                <input type="submit" value="Add" class="btn btn-primary">
            </div>
        </form>
    </div>
</div>
</body>
</html>
```



Update.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>Edit Student</title>
    <link rel="stylesheet"</pre>
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
    <style type="text/css">
    form
    {
        width: 50%;
        margin: 10px 25%;
    }
      </style>
</head>
<body>
    <div class="container">
        <div class="links">
            <a href="/logout">Log Out</a>
        </div>
        <div class="links">
```

```
<a href="/student/">Go Back</a>
        </div>
        <div class="text">
            <h1 class="text-center my-4">Update Student</h1>
            <hr class="w-25 m-auto bg-dark">
        </div>
        <div class="err">
            <% if(typeof errors!=='undefined' ){ %>
                    <%= errors.validationError %>
                <% } %>
        </div>
        <form class="flex-center" action="/student/edit/<%= data?._id %>"
method="post">
            <input type="hidden" name="_id" value="<%= data?._id %>">
            <div class="form-group">
                <label for="name">Student Name</label>
                <input type="text" class="form-control" placeholder="Enter Student</pre>
name" name="name" value="<%= data?.name %>" required>
            </div>
            <div class="form-group">
                <label for="name">Password</label>
                <input type="password" class="form-control" name="password"</pre>
placeholder="Enter Password" id="" value="<%= data?.password %>" required>
            </div>
            <div class="form-group">
                <label for="name">email</label>
                <input type="email" class="form-control" name="email"</pre>
placeholder="Enter email" id="" value="<%= data?.email %>" required>
            </div>
            <div class="form-group">
                <label for="name">city</label>
                <input type="text" class="form-control" name="city"</pre>
placeholder="Enter city" id="" value="<%= data?.city %>" required>
            </div>
            <div class="form-group">
                <label for="name">Age</label>
                <input type="text" class="form-control" name="age" id="" min="10"</pre>
max="80" placeholder="Enter Age" value="<%= data?.age %>" id="" required>
            </div>
            <label for="name">Gender</label>
            <div class="form-check">
                <input class="form-check-input" type="radio" name="gender"</pre>
value="Male" id="" <% if(data?.gender==='Male' ) {%>checked<%} %> checked>
                <label class="form-check-label" for="flexRadioDefault2">
                    Male
                </label>
              </div>
```



View.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>Student Detail</title>
</head>
<body>
   <% if(typeof(data)!=='undefined' ) { %>
       <div class="student-container">
           name: <%= data.name %>
           email: <%= data.email %>
           age: <%= data.age %>
           city: <%= data.city %>
           gender: <%= data.gender %>
           </div>
       <div class="links">
           <a href="/student/">Go Back</a>
       </div>
       <div class="links mt-2">
           <a href="/logout">Log Out</a>
       </div>
       <% } else{ %>
           Student Not Available!
           <%}%>
</body>
</html>
```

Delete.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>Student Detail</title>
    <link rel="stylesheet"</pre>
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
    <style type="text/css">
        form
        {
            width: 50%;
            margin: 10px 25%;
          </style>
</head>
<body>
    <% if(typeof(data)!=='undefined' ) { %>
    <div class="container">
        <div class="links">
            <a href="/logout">Log Out</a>
        </div>
        <div class="links">
            <a href="/student/">Go Back</a>
        </div>
        <div class="text">
            <h1 class="text-center my-4">Delete Student</h1>
            <hr class="w-25 m-auto bg-dark">
        </div>
        <div class="err">
            <% if(typeof errors!=='undefined' ){ %>
                >
                    <%= errors.validationError %>
                <% } %>
        </div>
        <form class="flex-center"action="/student/delete" method="post">
            <input type="hidden" name=" id" value="<%= data?. id %>">
            <div class="form-group">
                <label for="name">Student Name</label>
                <input type="text" class="form-control" placeholder="Enter Student</pre>
name" name="name" value="<%= data?.name %>" disabled>
```

```
</div>
            <div class="form-group">
                <label for="name">Password</label>
                <input type="password" class="form-control" name="password"</pre>
placeholder="Enter Password" value="<%= data?.password %>"
                id="" disabled>
            </div>
            <div class="form-group">
                <label for="name">email</label>
                <input type="email" class="form-control" name="email"</pre>
placeholder="Enter email" id="" value="<%= data?.email %>"
                disabled>
            </div>
            <div class="form-group">
                <label for="name">city</label>
                <input type="text" class="form-control" name="city"</pre>
placeholder="Enter city" id="" ue="<%= data?.city %>"
                disabled>
            </div>
            <div class="form-group">
                <label for="name">Age</label>
                <input type="text" class="form-control" name="age" id="" min="10"</pre>
max="80" placeholder="Enter Age" value="<%= data?.age %>" disabled>
            </div>
            <label for="name">Gender</label>
            <div class="form-check">
                <input class="form-check-input" type="radio" name="gender"</pre>
value="Male" id="" <% if(data?.gender==='Male' ){%>checked<%} %>
                <label class="form-check-label" for="flexRadioDefault2">
                    Male
                </label>
              </div>
              <div class="form-check">
                <input class="form-check-input" type="radio" name="gender"</pre>
value="Female" id="" <% if(data?.gender==='Female' ){%>checked<% }%>
                <label class="form-check-label" for="flexRadioDefault2">
                    Female
                </label>
              </div>
            <div class="form-group">
                <input type="submit" value="Confirm Delete" class="btn btn-primary">
            </div>
        </form>
    </div>
</div>
```

<% } else{ %>
 Student Not Available! <%}%>
</html>

