

Q-1)

App.js :=

```
const express = require('express')
const app = express()
const multer = require('multer')
const port = 6969
let fs = require('fs')
let path = require('path')

let Storage = multer.diskStorage({

  destination: (req, file, callback) => {
    const destinationFolder = './uploads'
    if (fs.existsSync(destinationFolder)) {
      callback(null, destinationFolder)
    } else {
      fs.mkdir(destinationFolder, (err) => {
        err ? console.error(err.stack) : callback(null,
destinationFolder)
      })
    }
  },
  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"]

let 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('/singlefile', upload.single('singleFile'), (req, res) => {
  return res.send('file is uploaded');
})

app.post('/multiplefiles', upload.array('multipleFiles', 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("ERRRRR");
    res.status(500).send("file upload err " + err.message);
  }
  else
    next(err);
});
```

```

app.listen(port,()=>{
  console.log(`🚀 Server running on port ${port}`)
})
```

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">
  <title>Upload single/multiple image</title>
  <style>
    * {
      font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif
    }

    .container {
      display: flex;
      flex-direction: column;
      justify-content: space-between;
      height: 30vh;
    }
  </style>
</head>
<body>
```

```
<div class="container">

  <div class="single-upload">
    <div class="header-single">
      <h3>Upload single File</h3>
      <form action="/singlefile" method="post"
enctype="multipart/form-data">
        <input type="file" name="singleFile" >
        <input type="submit" value="submit">
      </form>
    </div>
  </div>

  <div class="multiple-upload">
    <div class="header-single">
      <h3>Upload Multiple File</h3>
      <form action="/multiplefiles" method="post"
enctype="multipart/form-data">
        <input type="file" name="multipleFiles"
multiple="multiple">
        <input type="submit" value="submit">
      </form>
    </div>
  </div>
</div>
</body>
</html>
```



**Upload single File**

No file selected.

**Upload Multiple File**

No files selected.

Q-2)

Index.js:=

```
const express = require('express')
const app = express()
const session = require('express-session')
const path = require('path')
const FileStore = require('session-file-store')(session)

const PORT = 9696

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('/', (req, res, next) => {
  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('/home')
  } else {
    return res.send('Enter Username And Password!')
  }
})

app.get('/home', (req, res) => {
  if (req.session.loggedIn) {
    console.log('Login Success !! ☑☑')
    res.sendFile(__dirname + '/public/home.html')
  } else {
    console.log('Error while login !! ✕✕')
    res.sendFile(__dirname + '/public/login.html')
  }
})
```

```
  })  
  
  app.listen(PORT, () => {  
    console.log(`🚀🚀 server listening on port : ${PORT}`)  
  })  
})
```

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;  
      }  
    </style>  
  </head>  
  
  <body>  
    <div class="container flex-center">  
      <div class="header">  
        <h2>Log In</h2>  
      </div>  
      <div class="form-container mt-2 ">  
        <form class="flex-center" action="/login" method="post">  
          <div class="username mb-2">  
            <input type="text" placeholder="Username" name="username"  
required>
```

```
        </div>
        <div class="password mb-2">
            <input type="password" name="password"
placeholder="Password" id="" required>
        </div>
        <div class="submit">
            <input type="submit" value="Log In">
        </div>
    </form>
</div>
</div>
</body>
</html>
```

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">
        <h1>This is Home Page 😊 Yor are now Authenticate User ..</h1>
    </div>
</body>

</html>
```



Log In

Log In



This is Home Page 🎉 Yor are now Authenticate User ..



Q-4)

Db.js :=

```
var mongoose = require('mongoose');

//database configuration
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 Connected !! ☒`)
    } catch (error) {
      console.error(`❌❌ Database connection error : ' + error);
    }
  }
}

module.exports = new database()
```

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')
  }
}
```

Students.js (model) :=

```
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')
```

students.js (router) :=

```
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) => {
  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('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
```

index.js :=

```
const express = require('express')
const app = express()
const PORT = 9090
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')
      } 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 Port :${PORT}`)
})
```

Add.ejs :=

```

<div class="container flex-center">
  <div class="header">
    <h3>Add Student</h3>
  </div>
  <div class="err">
    <% if(typeof errors!=='undefined' ){ %>
      <p><%= errors.validationError %><p>
    <% } %>
  </div>
  <div class="form-container mt-2 ">
    <form class="flex-center" action="/student/" method="post">
      <div class="username mb-2">
        <input type="text" placeholder="enter your name"
name="name" required>
      </div>
      <div class="password mb-2">
        <input type="password" name="password" placeholder="enter
password" id="" required>
      </div>
      <div class="email mb-2">
        <input type="email" name="email" id="" placeholder="enter
your email" required>
      </div>
      <div class="city mb-2">
        <input type="text" name="city" id="" placeholder="enter
your city" required>
      </div>
      <div class="age mb-2">
        <input type="number" name="age" id="" min="10" max="80"
placeholder="Age here" required>
      </div>
      <div class="gender mb-2">
        <input type="radio" name="gender" value="Male" id=""
checked>Male
        <input type="radio" name="gender" value="Female"
id="">Female
      </div>
      <div class="submit">
        <input style="padding:0.4rem" type="submit" value="Add">
      </div>
    </form>
  </div>
</div>
<div class="links mt-2">
  <a href="/logout">Log Out</a>
</div>

```

update.ejs :=

```

<div class="container flex-center">
  <div class="header">
    <h3>Edit Student</h3>
  </div>
  <div class="err">
    <% if(typeof errors!=='undefined' ){ %>
      <p><%= errors.validationError %></p>
    <% } %>
  </div>
  <div class="form-container mt-2 ">
    <form class="flex-center" action="/student/edit/<%= data?._id %>"
method="post">
      <input type="hidden" name="_id" value="<%= data?._id %>">

      <div class="username mb-2">
        <input type="text" placeholder="enter your name"
name="name" value="<%= data?.name %>" required>
      </div>
      <div class="password mb-2">
        <input type="password" name="password" placeholder="enter
password" value="<%= data?.password %>"
id="" required>
      </div>
      <div class="email mb-2">
        <input type="email" name="email" id="" placeholder="enter
your email" value="<%= data?.email %>"
required>
      </div>
      <div class="city mb-2">
        <input type="text" name="city" id="" placeholder="enter
your city" value="<%= data?.city %>"
required>
      </div>
      <div class="age mb-2">
        <input type="number" name="age" id="" min="10" max="80"
placeholder="Age here"
value="<%= data?.age %>" required>
      </div>
      <div class="gender mb-2">
        <input type="radio" name="gender" value="Male" id="" <%
if(data?.gender==='Male' ) {%>checked<%} %>
>Male
        <input type="radio" name="gender" value="Female" id=""
<% if(data?.gender==='Female' ){%>
checked<% }%>>Female
      </div>
      <div class="submit">

```

```

        <input style="padding:0.4rem" type="submit"
value="Update">
      </div>
    </form>
  </div>
</div>
<div class="links mt-2">
  <a href="/logout">Log Out</a>
</div>

```

delete.ejs :=

```

<body>
  <% if(typeof(data)!='undefined' ) { %>
    <div class="student-container">
      <form class="flex-center" action="/student/delete" method="post">
        <input type="hidden" name="_id" value="<%= data?._id %>">

        <div class="username mb-2">
          <input type="text" placeholder="enter your name"
name="name" value="<%= data?.name %>" disabled>
        </div>
        <div class="password mb-2">
          <input type="password" name="password" placeholder="enter
password" value="<%= data?.password %>"
            id="" disabled>
        </div>
        <div class="email mb-2">
          <input type="email" name="email" id="" placeholder="enter
your email" value="<%= data?.email %>"
            disabled>
        </div>
        <div class="city mb-2">
          <input type="text" name="city" id="" placeholder="enter
your city" value="<%= data?.city %>"
            disabled>
        </div>
        <div class="age mb-2">
          <input type="number" name="age" id="" min="10" max="80"
placeholder="Age here"
            value="<%= data?.age %>" disabled>
        </div>
        <div class="gender mb-2">
          <input disabled type="radio" name="gender" value="Male"
id="" <% if(data?.gender==='Male' )
            {>checked<%} %>
            >Male

```





```

        <th>email</th>
        <th>age</th>
        <th>city</th>
        <th>gender</th>
        <th>Actions</th>
    </tr>
</thead>
<tbody>
    <% data.forEach(student=>{
        %>
        <tr>
            <td>
                <%= student.name %>
            </td>
            <td>
                <%= student.email %>
            </td>
            <td>
                <%= student.age %>
            </td>
            <td>
                <%= student.city %>
            </td>
            <td>
                <%= student.gender %>
            </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>

```

login.ejs :=

```

<body>
  <div class="container flex-center">
    <div class="header">
      <h3>Log In</h3>
    </div>
    <div class="err">
      <% if(typeof errors!=='undefined' ){ %>
        <p>
          <%= errors.validationError %>
        </p>
      <% } %>
    </div>
    <div class="form-container mt-2 ">
      <form class="flex-center" action="/login" method="post">
        <div class="username mb-2">
          <input type="text" placeholder="enter username"
name="username" required>
        </div>
        <div class="password mb-2">
          <input type="password" name="password" placeholder="enter
password" id="" required>
        </div>
        <div class="submit">
          <input type="submit" value="Log In">
        </div>
      </form>
    </div>
  </div>
</body>

```

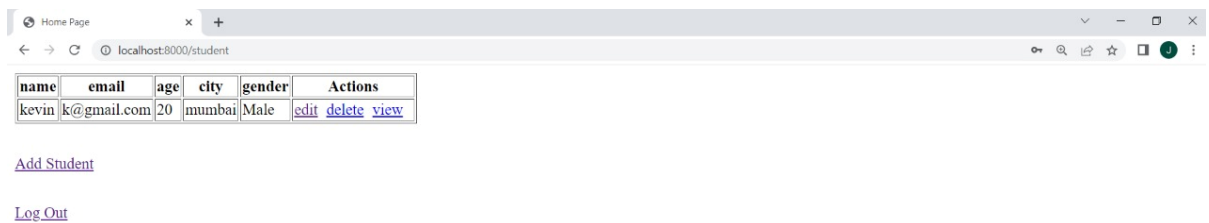
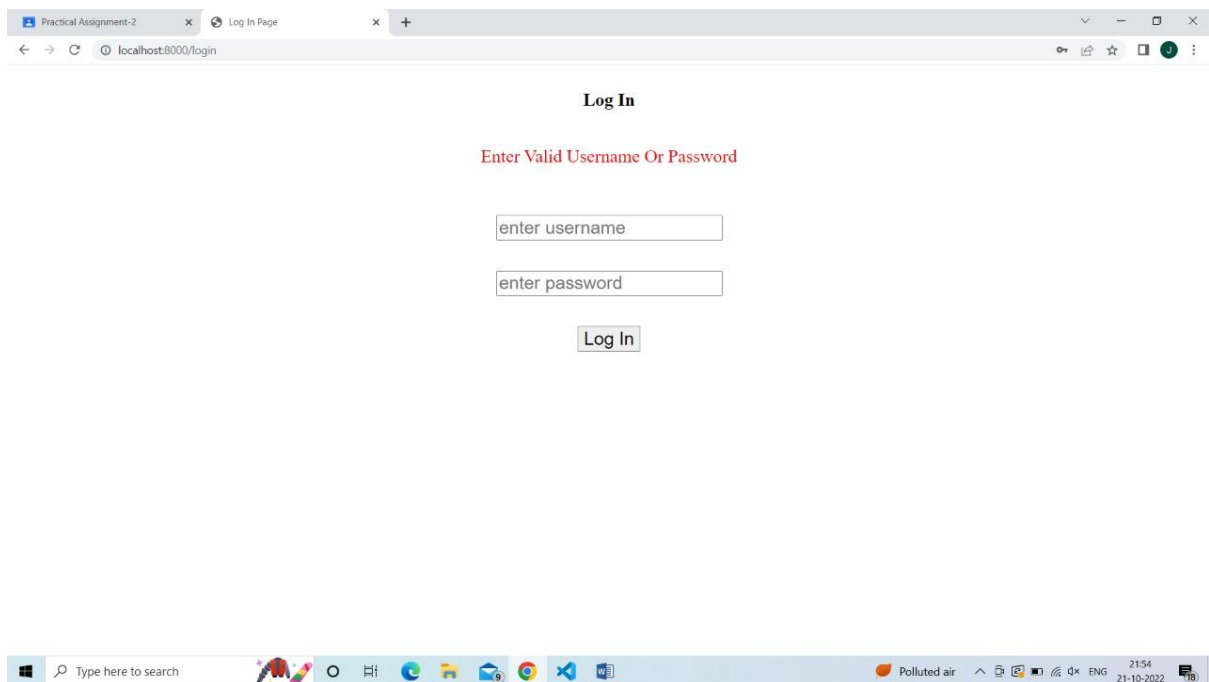
view.ejs :=

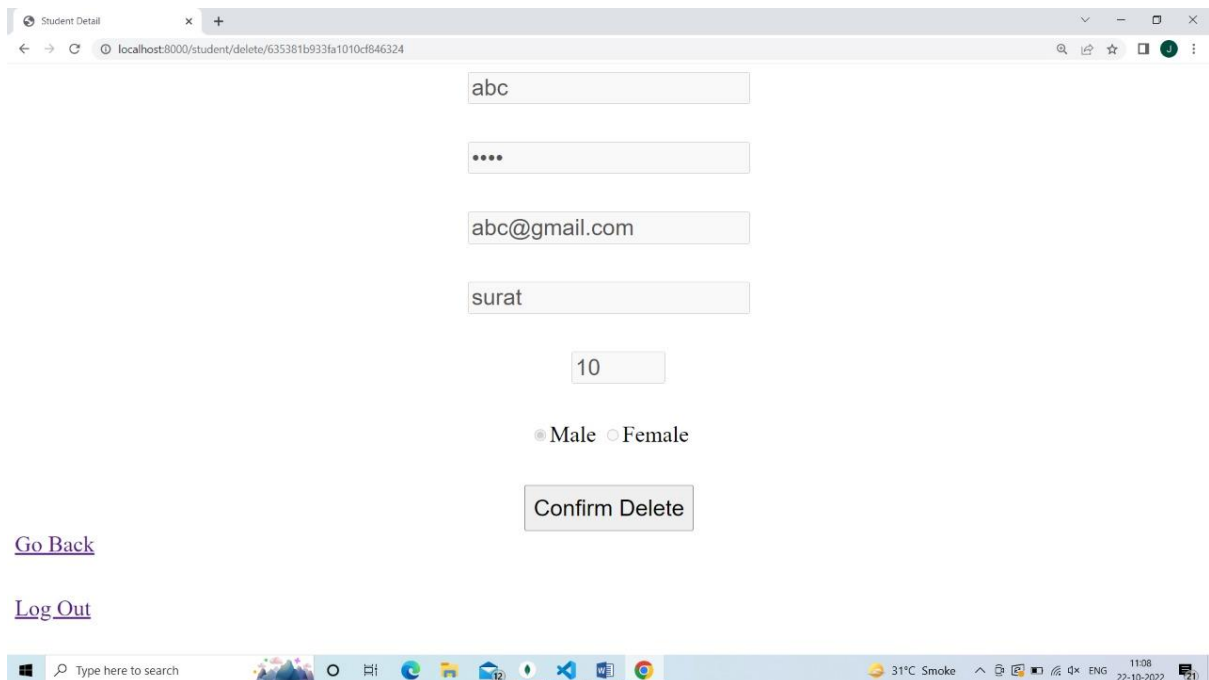
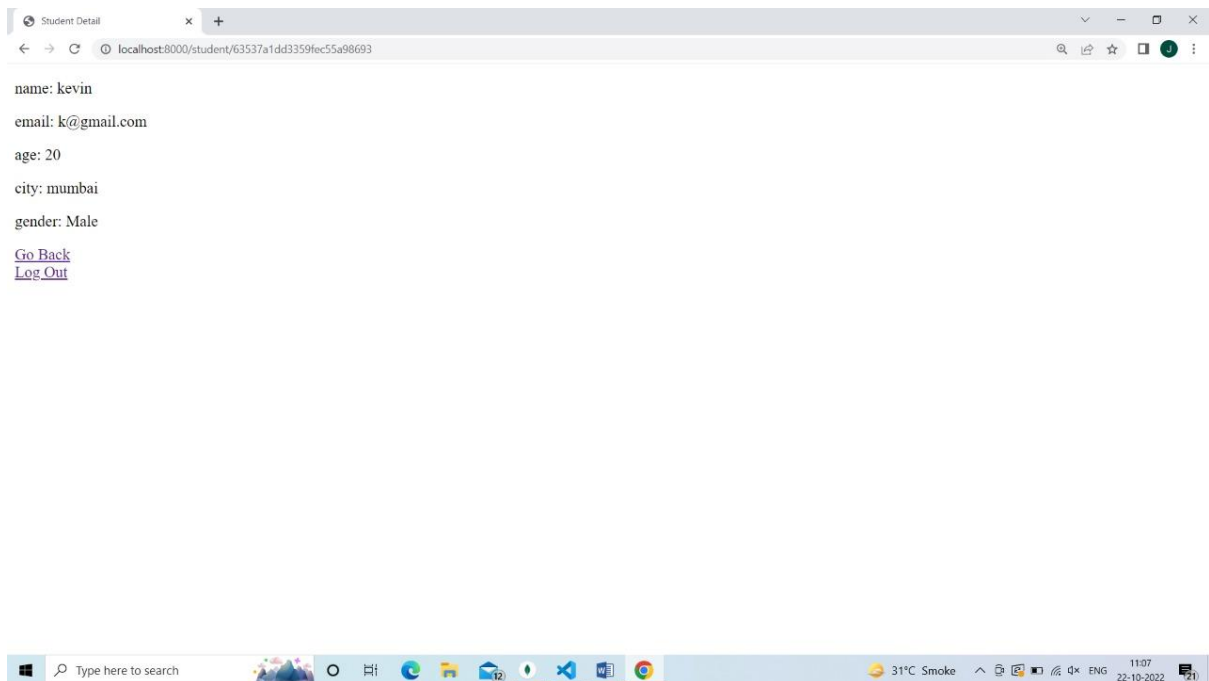
```

<body>
  <% if(typeof(data)!=='undefined' ) { %>
    <div class="student-container">
      <p>name: <%= data.name %>
      </p>
      <p>email: <%= data.email %>
      </p>
      <p>age: <%= data.age %>
      </p>
      <p>city: <%= data.city %>
      </p>
    </div>
  <% } %>

```

```
<p>gender: <%= data.gender %>
</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{ %>
  Student Not Available!
<%}%>
</body>
```





Q-5)

**Backend Part :=**

Database.js

```
var mongoose = require('mongoose');
const DB_URL = process.env.DB_URL
class database {
  constructor() {
    this._connect()
  }
  async _connect() {
    try {
      await mongoose.connect(DB_URL)
      console.log(`database connection successfull`)
    } catch (error) {
      console.error('Database connection error: ' + error);
    }
  }
}

module.exports = new database()
```

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 }
```

studentmodel.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(`🚀🚀 Server listening on Port : ${PORT}`);
})

```

**Frontend Part :-**

Add.html

```

<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 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 = {}) {

        const response = await fetch(url, {
            method: 'POST',
            mode: 'cors',
            cache: 'no-cache',
            credentials: 'same-origin',
            headers: {
                'Content-Type': 'application/json',
                'Authorization': `Bearer ${localStorage.getItem('authToken')}`
            },
            redirect: 'follow',
            referrerPolicy: 'no-referrer',
            body: JSON.stringify(data)
        });
        return response.json();
    }
}
```

```

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>

```

delete.html

```

<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 = {}) {
    const response = await fetch(url, {

```

```

        method: 'DELETE',
        mode: 'cors',
        cache: 'no-cache',
        credentials: 'same-origin',
        headers: {
            'Content-Type': 'application/json',
            'Authorization': `Bearer ${localStorage.getItem('authToken')}`
        },
        redirect: 'follow',
        referrerPolicy: 'no-referrer',
    });
    return response.json();
}

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>

```

update.html

```

<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>
        </div>
    </div>

```

```
        <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 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',
    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',
    body: JSON.stringify(data)
  });
  return response.json();
}

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>

```

index.html

```

<body>
    <div class="container">
        <div class="error">
            <table border="1">
                <thead>
                    <tr>
                        <th>name</th>
                        <th>email</th>
                        <th>age</th>
                        <th>city</th>
                        <th>gender</th>
                        <th>Actions</th>
                    </tr>
                </thead>
                <tbody id="tableBody">
                </tbody>
            </table>
            <div class="links mt-2">
                <a href="./add.html">Add Student</a>
            </div>
            <div class="links mt-2">
                <a href="./logout.html">Log Out</a>
            </div>
        </div>
    </div>
</body>

<script>
    const tableBody = document.getElementById('tableBody')
    const fetchStudents = async () => {

```

```
const url = "http://localhost:8000/student"
const response = await fetch(url, {
  headers: {
    'Authorization': `Bearer ${localStorage.getItem('authToken')}`
  }
})
return response.json()
}
fetchStudents().then(data => {
  data.forEach(student => {
    tableBody.innerHTML = tableBody.innerHTML + `
      <tr>
        <td>
          ${student.name}
        </td>
        <td>
          ${student.email}
        </td>
        <td>
          ${student.age}
        </td>
        <td>
          ${student.city}
        </td>
        <td>
          ${student.gender}
        </td>
        <td>
          <div class="mr-1">
            <a
href="./update.html?id=${student._id}" class="mr-1">edit</a>
            <a
href="./delete.html?id=${student._id}" class="mr-1">delete</a>
            <a
href="./view.html?id=${student._id}" class="mr-1">view</a>
          </div>
        </td>
      </tr>
    `
  });
})
</script>
```



view.html

```
<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>
```

login.html

```
<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>
</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)
      });
  }

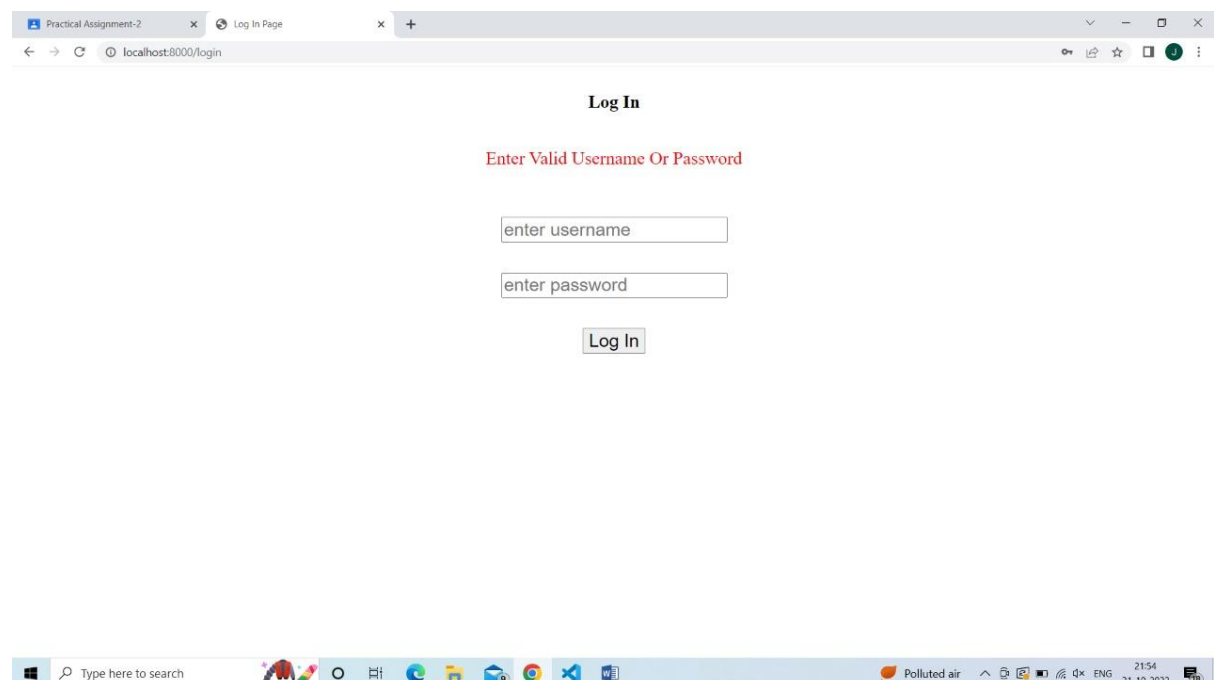
  async function postData(url = '', data = {}) {
    // Default options are marked with *
    try {
      const response = await fetch(url, {
        method: 'POST', // *GET, POST, PUT, DELETE, etc.
        mode: 'cors', // no-cors, *cors, same-origin
        cache: 'no-cache',
        credentials: 'same-origin',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

```
        'Content-Type': 'application/json'
      },
      redirect: 'follow', // manual, *follow, error
      refererPolicy: 'no-referrer',
      body: JSON.stringify(data)
    });
    if (response.status === 200) {
      return response.json();
    } else {
      let msg = await response.json()
      alert(msg)
    }
  } catch (error) {
    alert(error)
  }
}

</script>
```

logout.html

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



**Add Student**

aaa

\*\*\*\*

a@gmail.com

surat

20

☒ Male ☐ Female

Add

[Log Out](#)

name	email	age	city	gender	Actions
kevin	k@gmail.com	20	mumbai	Male	<a href="#">edit</a> <a href="#">delete</a> <a href="#">view</a>

[Add Student](#)

[Log Out](#)

**Edit Student**

kevin

\*\*\*\*

k@gmail.com

mumbai

20

☒ Male ☐ Female

Update