

Cloud Computing Lab 3

Name: Som Singh Lodhi

Mis No: 112115157

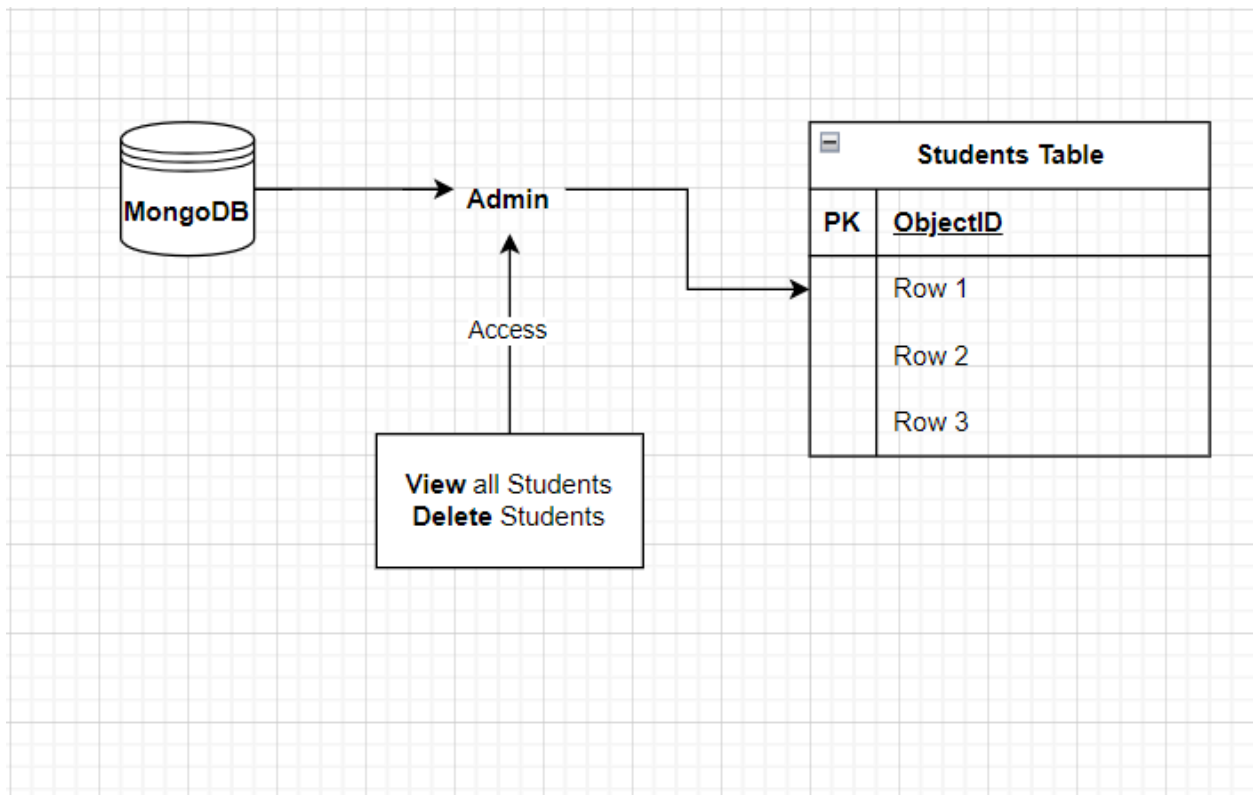
Problem Statement 2

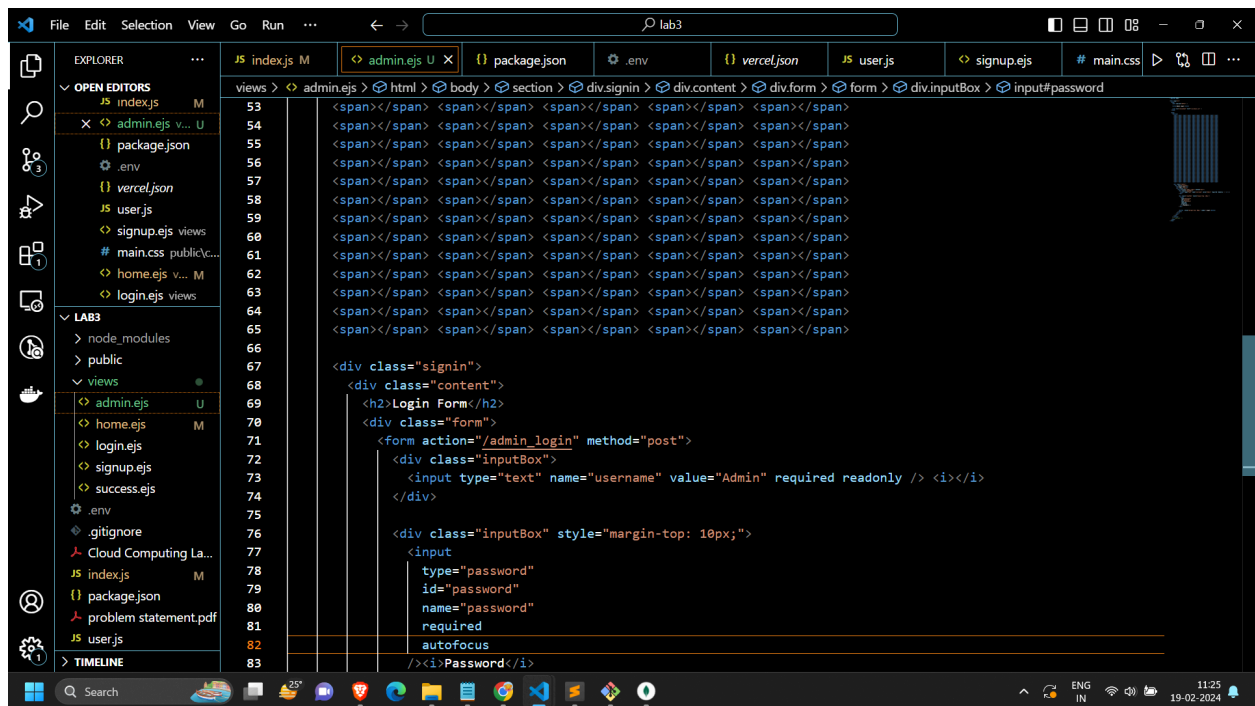
Deployment Link: <https://iiitp-cc.vercel.app/>

Github Repo : <https://github.com/SomSingh23/cc-lab3>

Admin Password: idon'tknow

Work-Flow





```
28 app.get("/admin", (req, res) => {
29   |   res.render("admin");
30   | });
31   // post routes start here
32
```

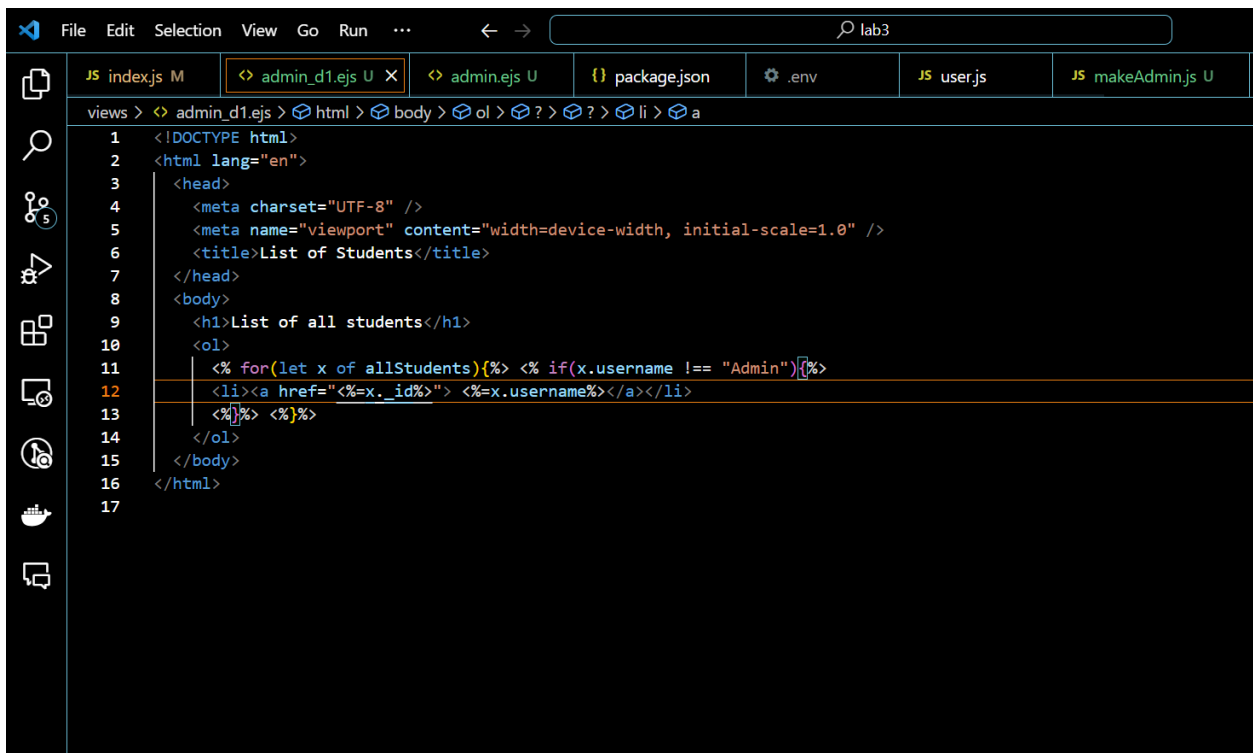
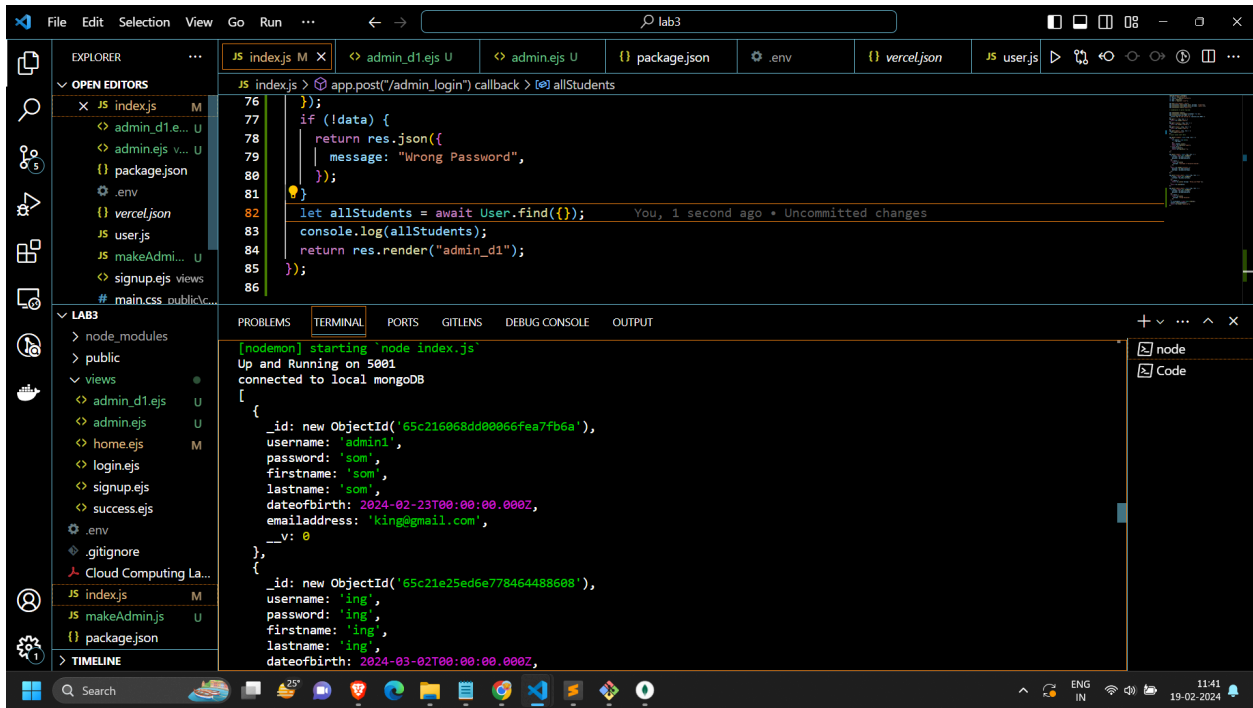
```
> app.post("/admin_login", (req, res) => {
>   |   res.json({
>   |     |   message: "ok 200",
>   |     | });
>   | });
```

JS index.js M	<> admin.ejs U	{ } package.json	⚙ .env	{ } vercel.json
---------------	----------------	------------------	--------	-----------------

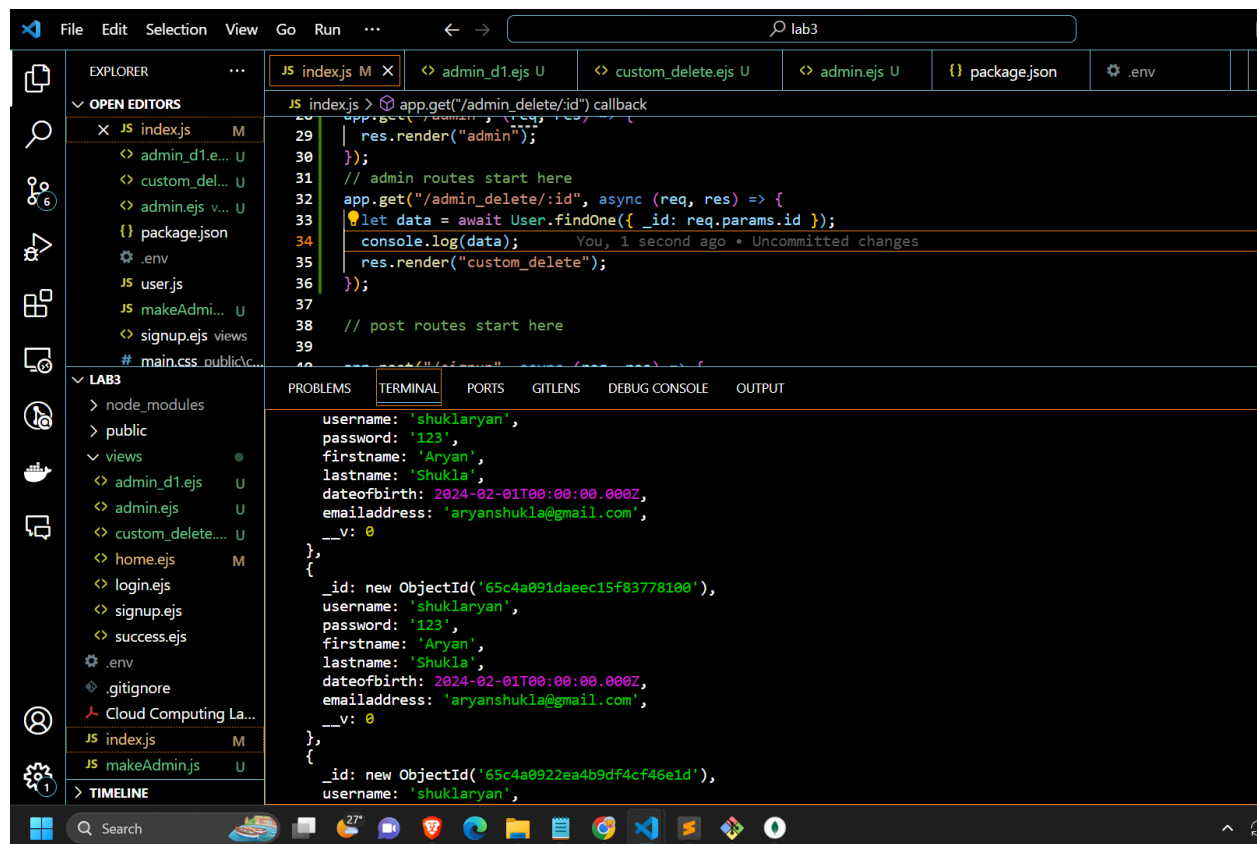
```
JS makeAdmin.js > [🔗] newAdmin
1   require("dotenv").config();
2   let User = require("../user");
3   let newAdmin = new User({
4     username: "Admin",
5     password: "", // something strong
6   });
7   newAdmin
8     .save()
9     .then((p) => console.log(p))
10    .catch((err) => console.log(err));
11
```

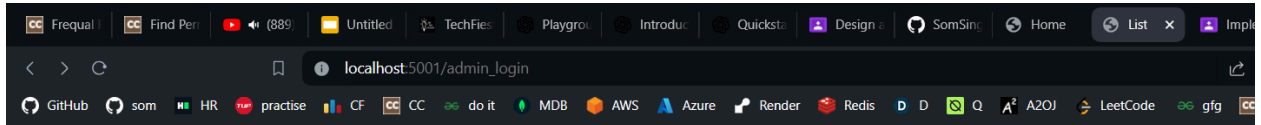
```
71
72  app.post("/admin_login", async (req, res) => {
73    let data = await User.findOne({
74      username: req.body.username,
75      password: req.body.password,
76    });
77    if (!data) {
78      return res.json({
79        message: "Wrong Password",
80      });
81    }
82    return res.send("by pass");
83  });
84
```

You, 1 second ago • Uncommitted changes



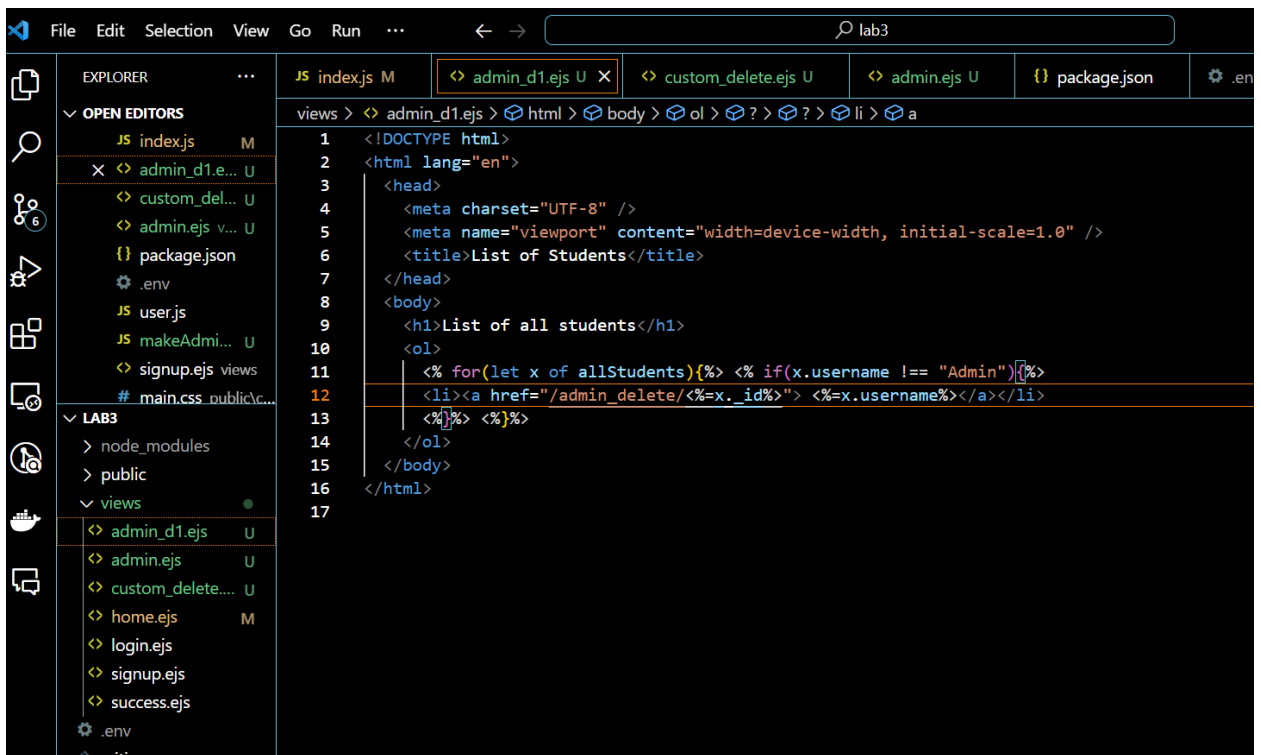
```
27   });
28   app.get("/admin", (req, res) => {
29     |   res.render("admin");
30   });
31   // admin routes start here
32   app.get("/admin_delete/:id", async (req, res) => {
33     |   res.json({
34     |     message: req.params.id,
35     |   });
36   });
37
```





List of all students

1. [admin1](#)
2. [ipg](#)
3. [king](#)
4. [tabfizm](#)
5. [Priyanshu_25](#)
6. [priyanshutyagi_25](#)
7. [priyanshutyagi_25](#)
8. [priyanshutyagi_25](#)
9. [shuklaryan](#)
10. [shuklaryan](#)
11. [shuklaryan](#)
12. [shuklaryan](#)
13. [shuklaryan](#)
14. [shuklaryan](#)
15. [shuklaryan](#)
16. [shuklaryan](#)
17. [aws](#)
18. [fb](#)
19. [sdf](#)



The screenshot shows the VS Code interface with the Explorer sidebar on the left. The Explorer sidebar shows a project named 'LAB3' with a 'views' folder containing several files: 'admin_d1.ejs', 'admin.ejs', 'custom_delete.ejs', 'home.ejs', 'login.ejs', 'signup.ejs', and 'success.ejs'. The 'custom_delete.ejs' file is selected and its content is displayed in the main editor area. The content is an HTML template for deleting a user, featuring a title 'Deleting', a header with the user's name, and a form with fields for Username, Email, and Date of Birth, followed by a confirmation message.

```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8" />
5     <meta name="viewport" content="width=device-width, initial-scale=1.0" />
6     <title>Deleting</title>
7   </head>
8   <body>
9     <hr></hr>
10    <h1><%=data.username%> Details</h1>
11
12    <p>Username: <b><%=data.username%></b></p>
13    <p>Email: <b><%=data.emailaddress%></b></p>
14    <p>Date of Birth : <b><%=data.dateofbirth%></b></p>
15    <p>Are you sure you want to delete User <b><%=data.username%></b></p>
16    <hr></hr>
17  </body>
18 </html>
19
```

The screenshot shows the VS Code interface with the content of 'app.js' displayed in the main editor area. The code defines a POST route for deleting a user, using the 'User.deleteOne' method to delete the user by ID and returning a JSON response with a message.

```
90   return res.render("admin_d1", { allStudents });
91 });
92 app.post("/admin_delete/:id", async (req, res) => {
93   await User.deleteOne({ _id: req.params.id });
94   return res.json({ message: `User with id=${_id} Deleted` });
95 });
96
```

JS index.js M

<> admin_d1.ejs U

<> custom_delete.ejs U X

<> admin.ejs U

{ package.j

views > <> custom_delete.ejs > html > body > form

1 <!DOCTYPE html>

2 <html lang="en">

3 <head>

4 <meta charset="UTF-8" />

5 <meta name="viewport" content="width=device-width, initial-scale=1.0" />

6 <title>Deleting</title>

7 </head>

8 <body>

9 <hr></hr>

10 <h1><%=data.username%> Details</h1>

11

12 <p>Username: <%=data.username%></p>

13 <p>Email: <%=data.emailaddress%></p>

14 <p>Date of Birth : <%=data.dateofbirth%></p>

15 <p>Are you sure you want to delete User <%=data.username%></p>

16

17 <form action="/admin_delete/<%=data._id%>" method="POST">

18 <button type="submit">Delete <%=data.username%></button>

19 <hr></hr>

20 </body>

21 </html>

22

31 // admin routes start here

32 app.get("/admin_delete/:id", async (req, res) => {

33 try {

34 let data = await User.findOne({ _id: req.params.id });

35

36 res.render("custom_delete", { data });

37 } catch (err) {

38 return res.json({ message: "Invalid ID or User is Deleted" });

39 }

40 });

41

42 // post routes start here

fb Details

Username: fb

Email: fb@fb.india

Date of Birth : Sat Aug 08 98989 05:30:00 GMT+0530 (India Standard Time)

Are you sure you want to delete User fb

Delete

