

Software Documentation : JYTHU

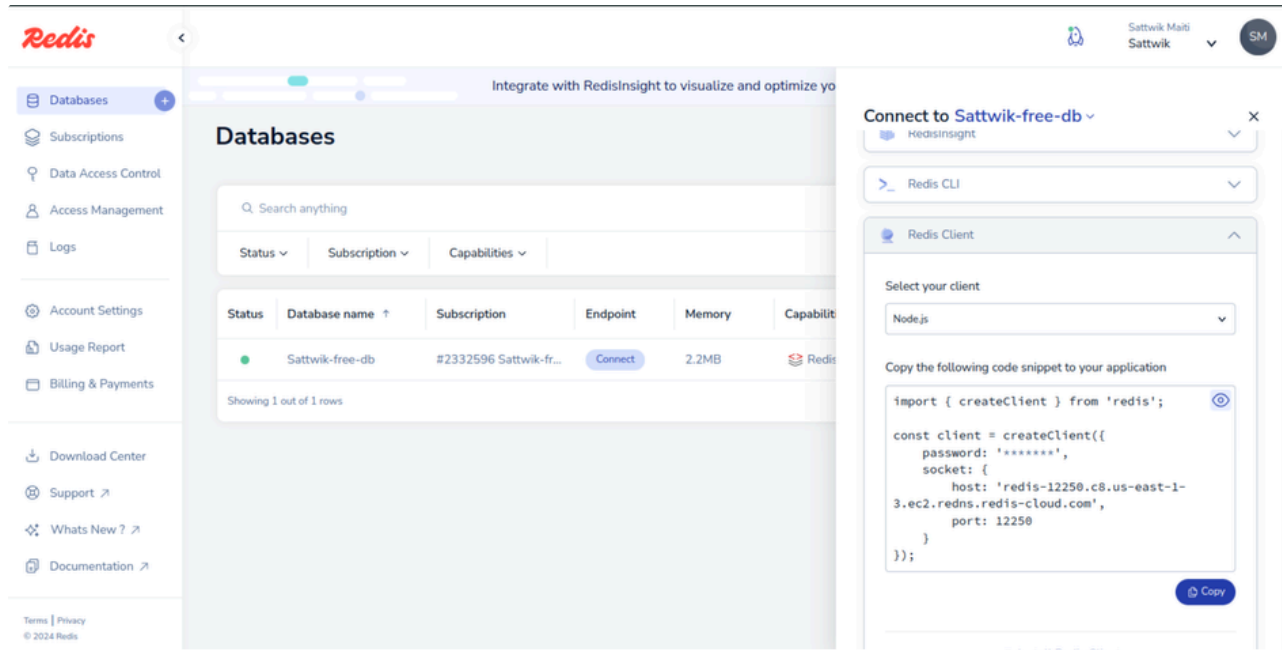


Environment Variables

1. Google oAuth 2.0
2. Cloudinary
3. Mongo DB
4. Rabbit Message Queue Cloud (Cloud AMQP)
5. Redis Cloud Host
6. Hosted Server URL

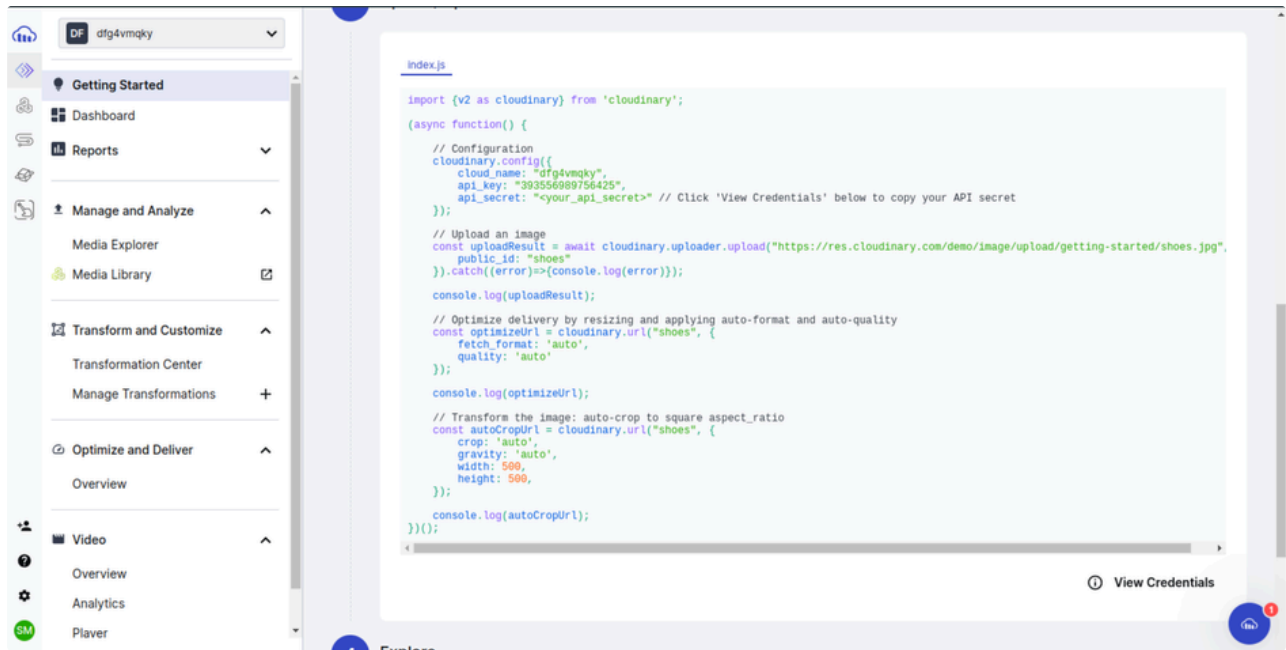
Redis Cloud Setup

1. Go To Redis Cloud Console [Redis Cloud Console](#)
2. Generate the Password and Redis Cloud Link. Click on Connect from Endpoints and go to Redis Client. Copy the Snippet to Your Application



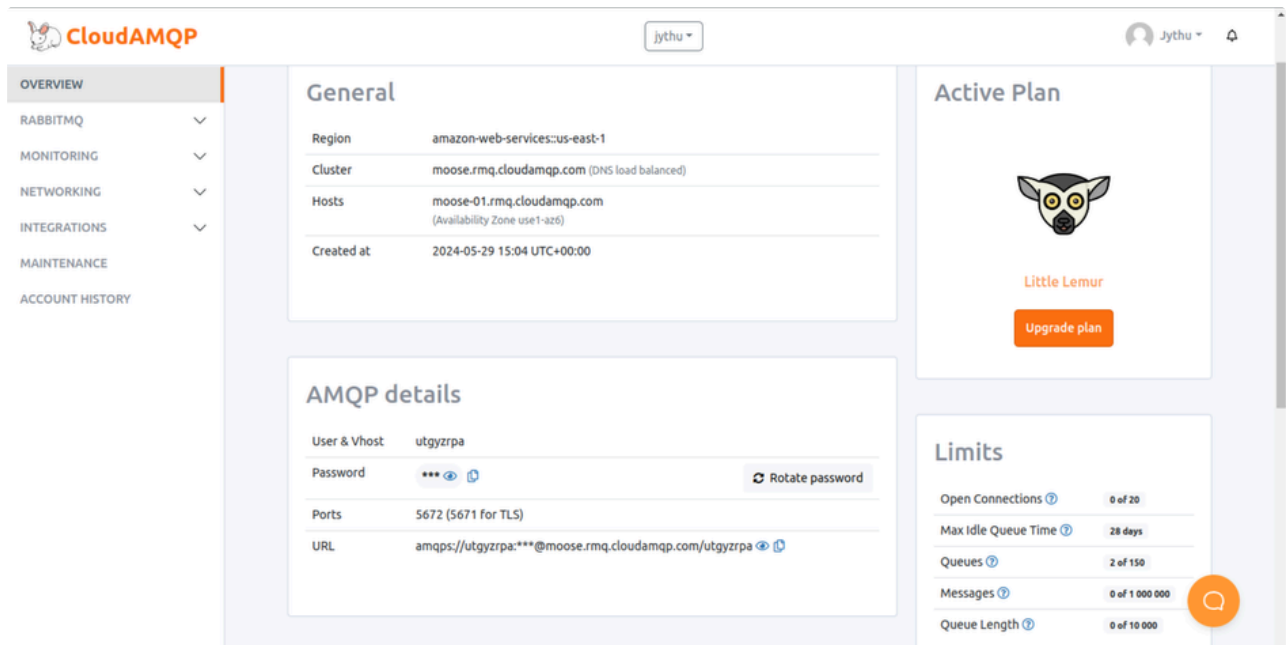
Cloudinary Setup

1. Go To Cloudinary Dashboard [Cloudinary](#)
2. Click On View Credentials and get the Cloudinary secret key .



Rabbit Message Queue

1. Create an Account on [Cloud AMQP](#)
2. Copy the AMQP host URL and use it in the Rabbit MQ host URL



Google OAuth 2.0

1. Follow Documentation only for Google Credential Settings [Implementing OAuth 2.0 to React for User Authorization](#)

2. Copy this below to your .env file

```
GOOGLE_CLIENT_ID=<the client ID you created earlier>
GOOGLE_CLIENT_SECRET=<the client secret you created earlier>
REDIRECT_URL=http://localhost:3000/auth/callback
CLIENT_URL=http://localhost:3000
TOKEN_SECRET="Jythu"
```

Replace Redirect and Client URLs with the client-side hosted URL

Mongo DB URL

1. Create an account on [MongoDB](#)

2. Follow This Documentation [How to Create a Database in MongoDB Atlas and connect your Database to your Application \(Step By...](#)

3. Just Add one extra IP address - 0.0.0.0 Description: entire Internet

Add entries to your IP Access List

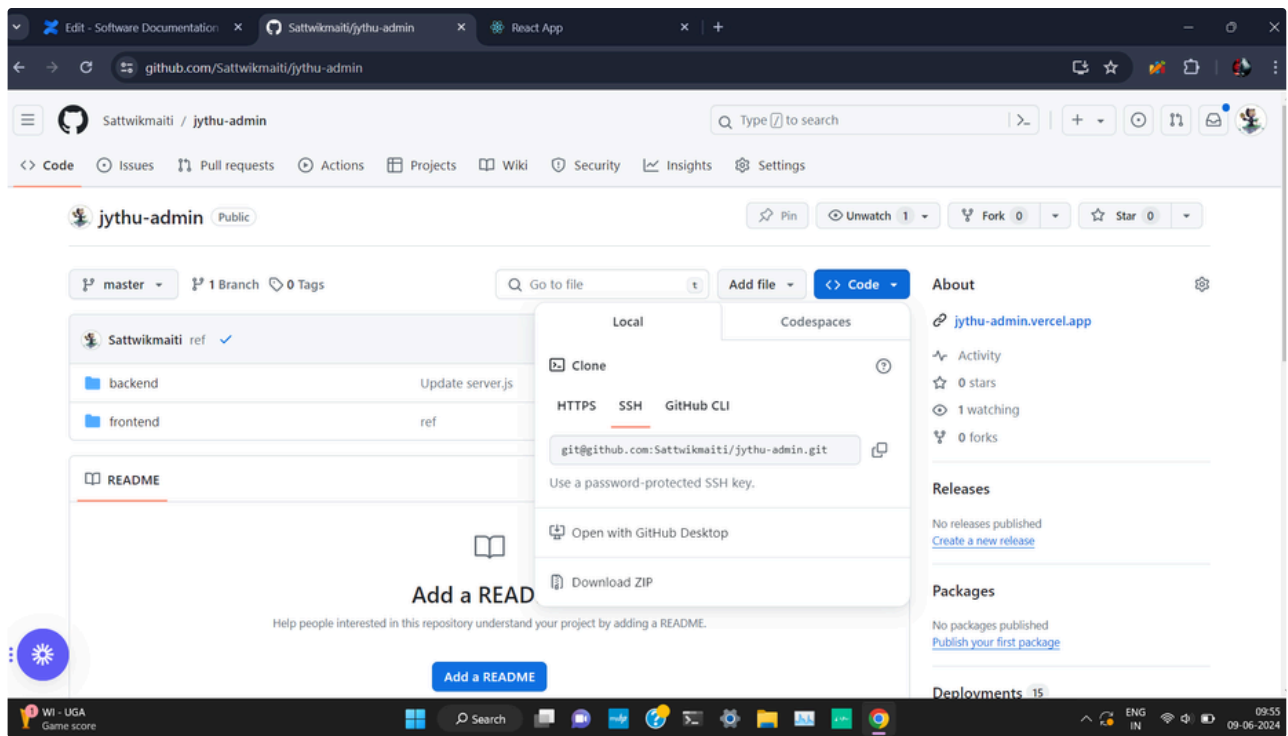
Only an IP address you add to your Access List will be able to connect to your project's clusters. You can manage existing IP entries via the [Network Access Page](#).

IP Address	Description	
<input type="text" value="Enter IP Address"/>	<input type="text" value="Enter description"/>	<input type="button" value="Add My Current IP Address"/>
<input type="button" value="Add Entry"/>		
<p>This IP address has already been added.</p>		

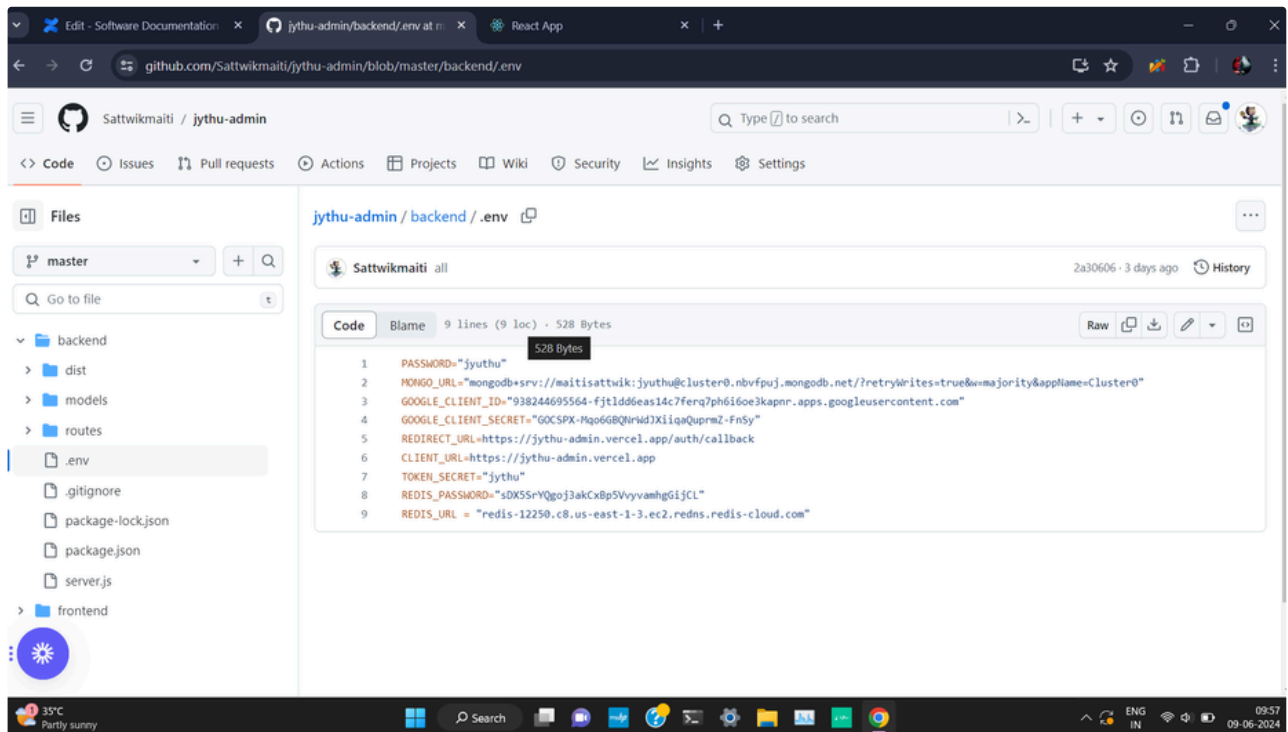
IP Access List	Description	
105.112.102.254/32	My IP Address	<input type="button" value="EDIT"/> <input type="button" value="REMOVE"/>

How to Setup Project

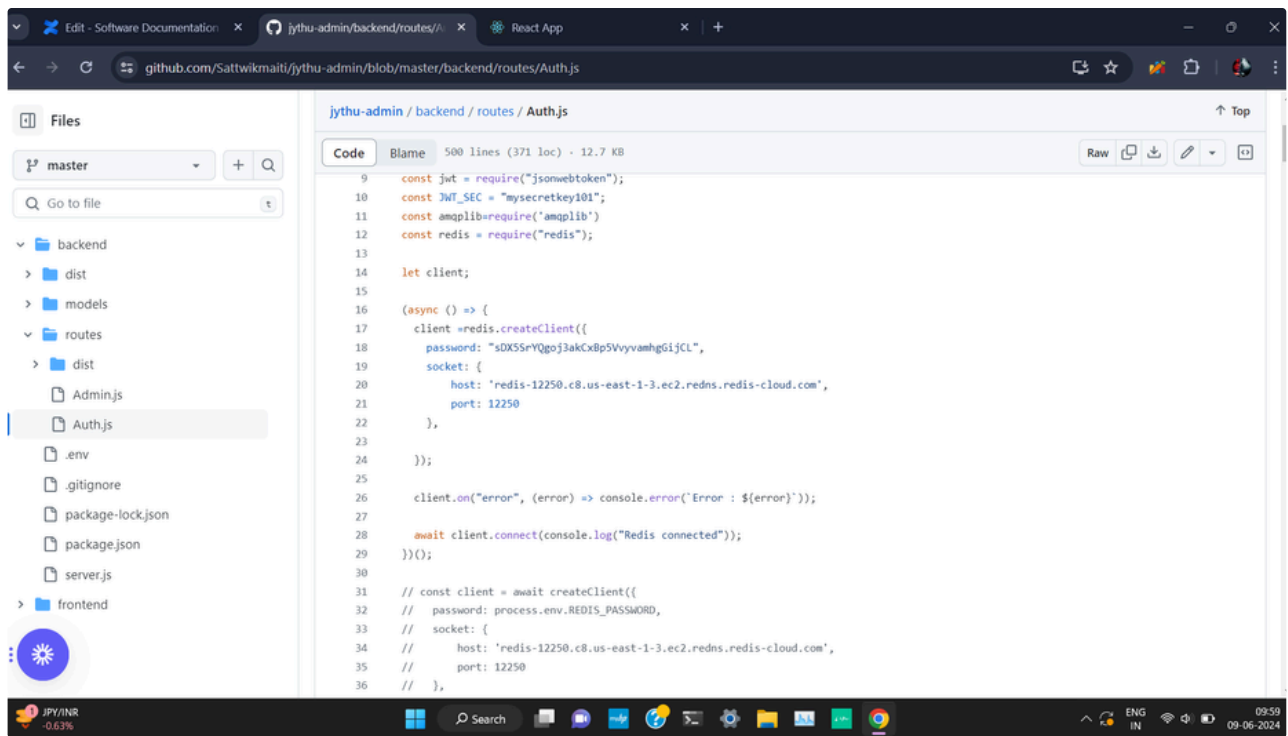
1. Download the zip folder from github



-
-
- Delete the git folder after extracting the files . File structure would be – git , backend , frontend . So delete the git folder.
- Go to backend folder

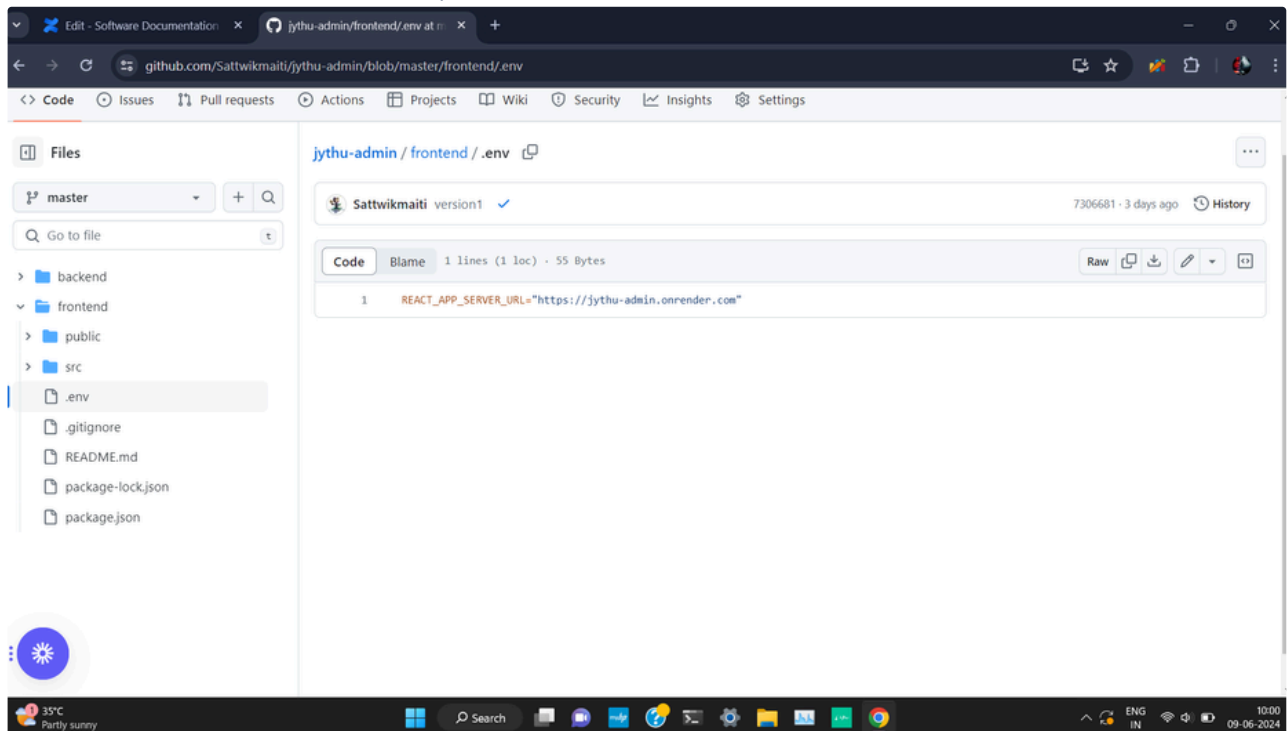


Edit all the details .



5. Edit the Redis URL and password . Port remains same

6. Go to Frontend folder . Create a .env file and paste the server url



7. Now Open terminal

> cd backend

>npm install

> npm start // starts the backend server

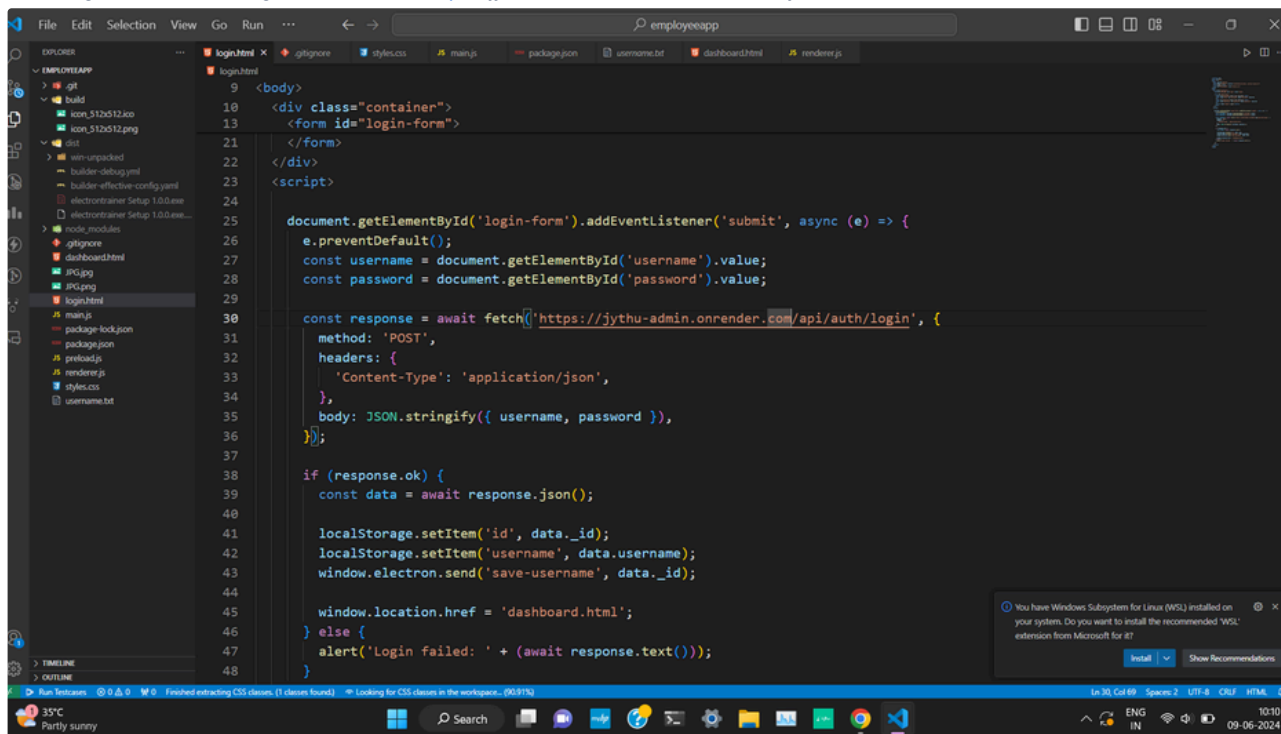
> cd ..

> cd frontend

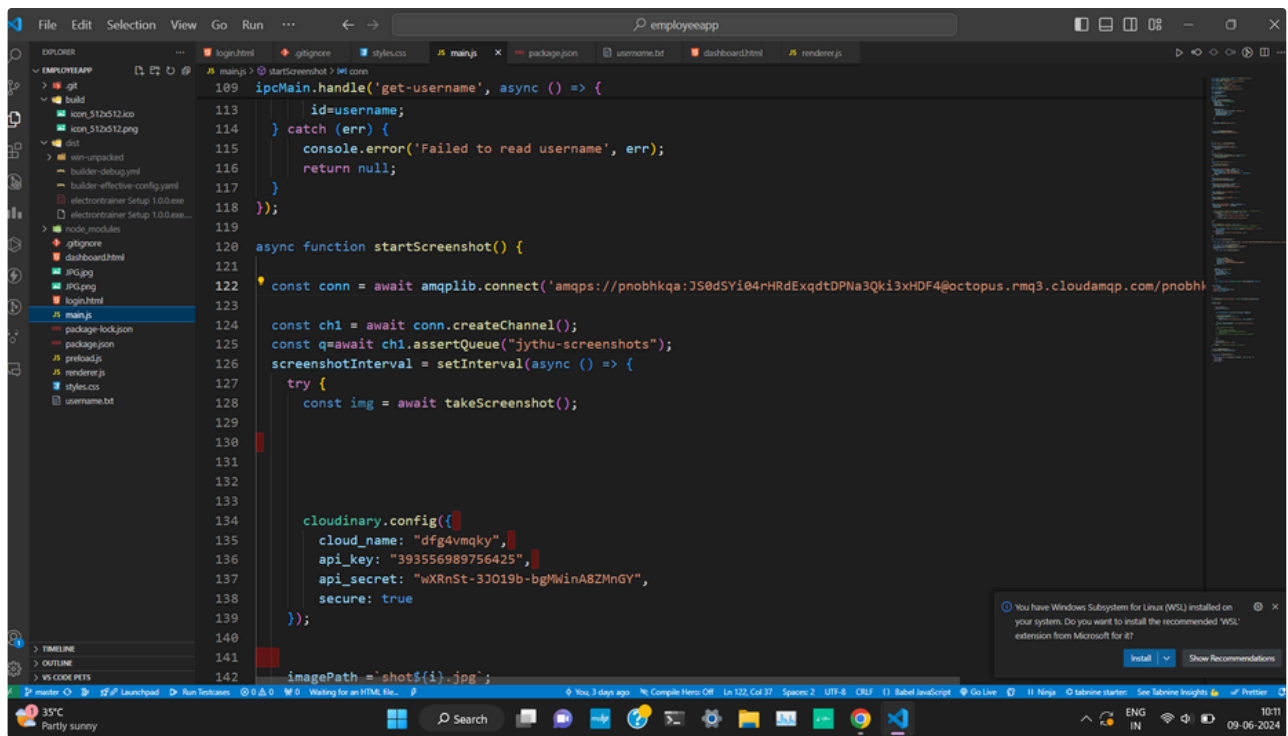
```
>npm install
> npm start // starts the frontend client
```

Electron App

1. Download the zip from github . Delete the git folder after extracting
2. Go to login.html file . Change the Server url <https://jythu-admin.onrender.com> with your hosted server url



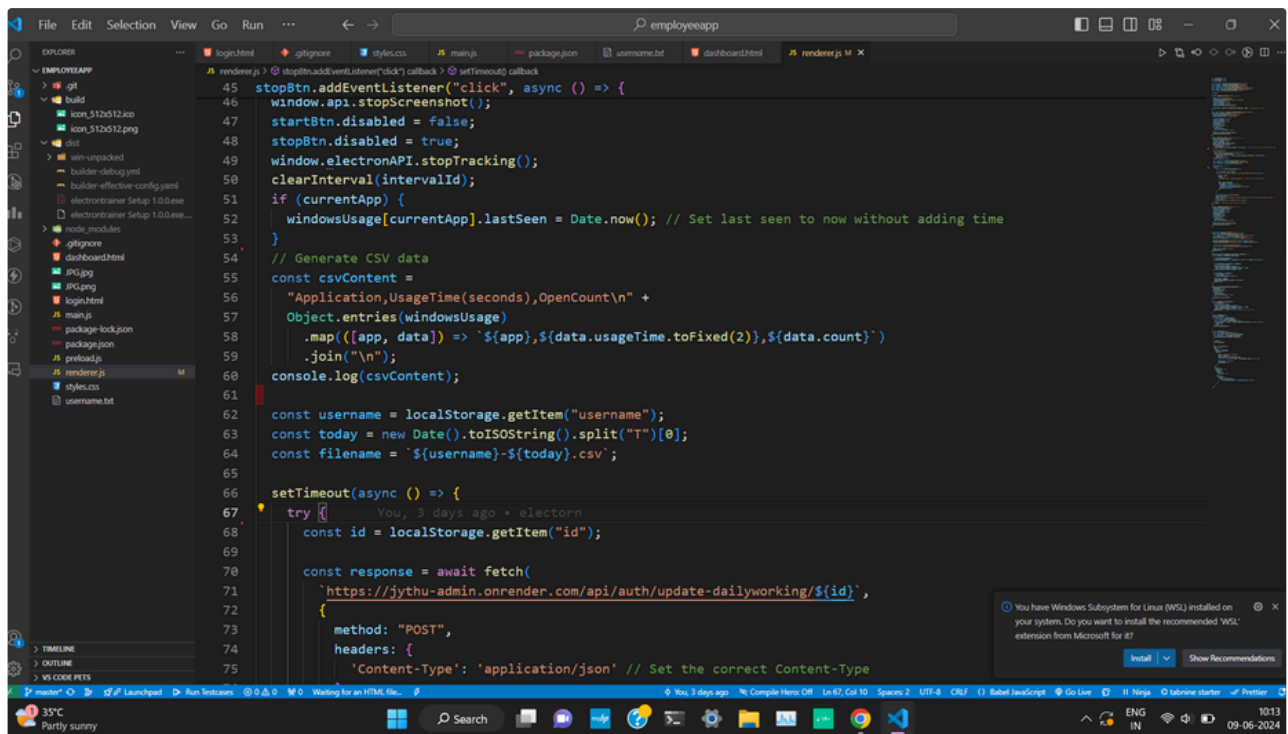
3. Go to main.js file and write your Rabbit CCloud AMQP link in the startScreenshot() function . Also Give your Cloudinary Cloud Details



```
109 ipcMain.handle('get-username', async () => {
110   id=username;
111 } catch (err) {
112   console.error('Failed to read username', err);
113   return null;
114 });
115
116 async function startScreenshot() {
117
118   const conn = await amqpLib.connect('amqps://pnobhkqa:3S0dSYi04rHRdExqdtDPNa3Qki3xHDF4@octopus.rm3.cloudamqp.com/pnobh
119
120   const ch1 = await conn.createChannel();
121   const q=await ch1.assertQueue("jythu-screenshots");
122   screenshotInterval = setInterval(async () => {
123     try {
124       const img = await takeScreenshot();
125
126       cloudinary.config({
127         cloud_name: "dfg4vmqky",
128         api_key: "393556989756425",
129         api_secret: "wXRnSt-3JO19b-bgMWinABZMnGY",
130         secure: true
131       });
132
133       imagePath = `shot${i}.jpg`;
134     } catch (err) {
135       console.error('Failed to upload screenshot', err);
136     }
137   }, 10000);
138 }
```

4.

5. Go to renderer.js file . Go to stopBtn.addEventListener('click') . Change the server url in the fetch response

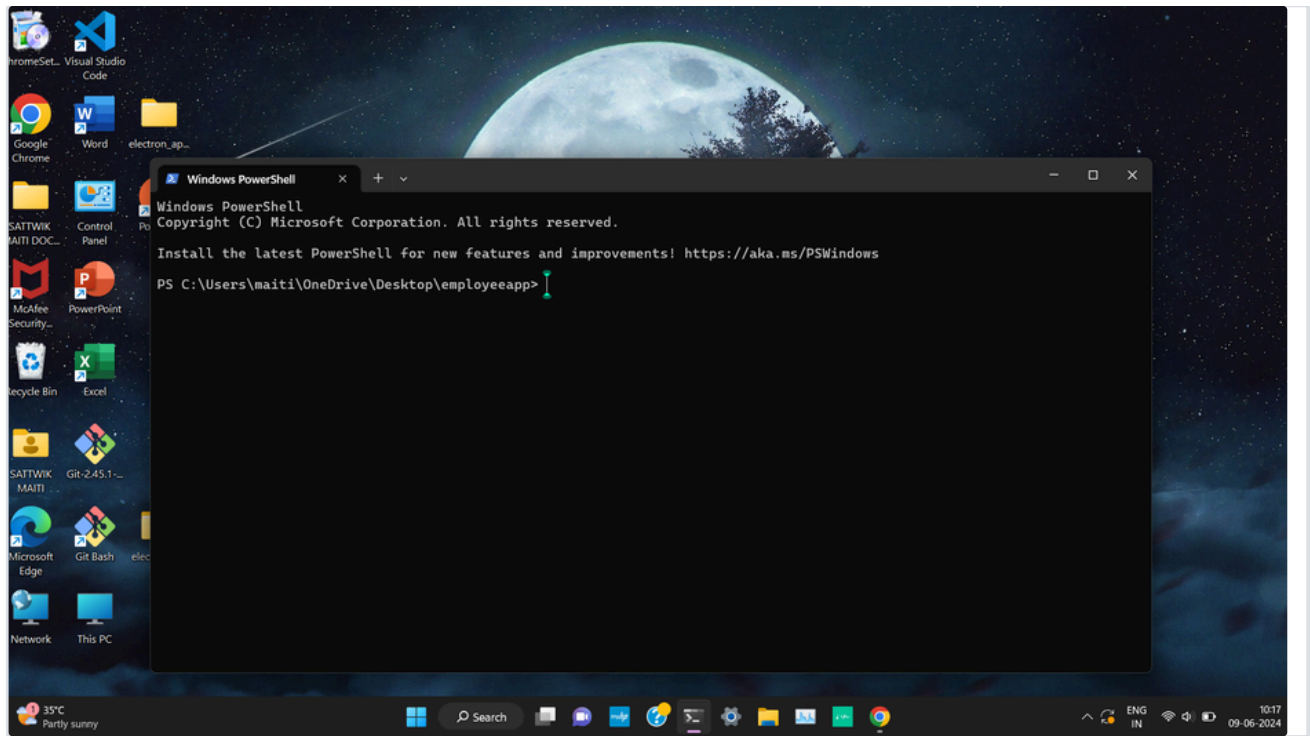


```
45 stopBtn.addEventListener("click", async () => {
46   window.api.stopScreenshot();
47   startBtn.disabled = false;
48   stopBtn.disabled = true;
49   window.electronAPI.stopTracking();
50   clearInterval(intervalId);
51   if (currentApp) {
52     windowsUsage[currentApp].lastSeen = Date.now(); // Set last seen to now without adding time
53   }
54   // Generate CSV data
55   const csvContent =
56     "Application,UsageTime(seconds),OpenCount\n" +
57     Object.entries(windowsUsage)
58       .map(([app, data]) => `${app},${data.usageTime.toFixed(2)},${data.count}`)
59       .join("\n");
60   console.log(csvContent);
61
62   const username = localStorage.getItem("username");
63   const today = new Date().toISOString().split("T")[0];
64   const filename = `${username}-${today}.csv`;
65
66   setTimeout(async () => {
67     try {
68       const id = localStorage.getItem("id");
69
70       const response = await fetch(
71         'https://jythu-admin.onrender.com/api/auth/update-dailyworking/${id}',
72         {
73           method: "POST",
74           headers: {
75             'Content-Type': 'application/json' // Set the correct Content-Type
```

6.

7. Go to terminal . npm install . npm start

8. If you want to make the app or windows apk . Run Windows PowerShell as administrator . Cd (change directory) to the folder where this project exists . To get the directory path yoy can also right click on the folder and open in terminal to get the path . copy the path and paste on the powershell as cd {path}



9. npm run dist

10. In the dist folder you will get the setup1.0.0.exe (executable) Application File (77074 kb) . You can install this app by clicking this .