

Candidate Name: Ganesh Thapa

Try to explain in your own way. It can be 2-3 lines.

1. Are you familiar with this configuration?

Ps: Its related to mongodb

Your Answer: Yes, I'm familiar with the given MongoDB Network access control configuration cluster. It lists allowed IP addresses that can connect to the cluster with options to manage (e.g delete) entries. The configuration ensures security by restricting Ips which is common practices for protecting databases like MongoDB.

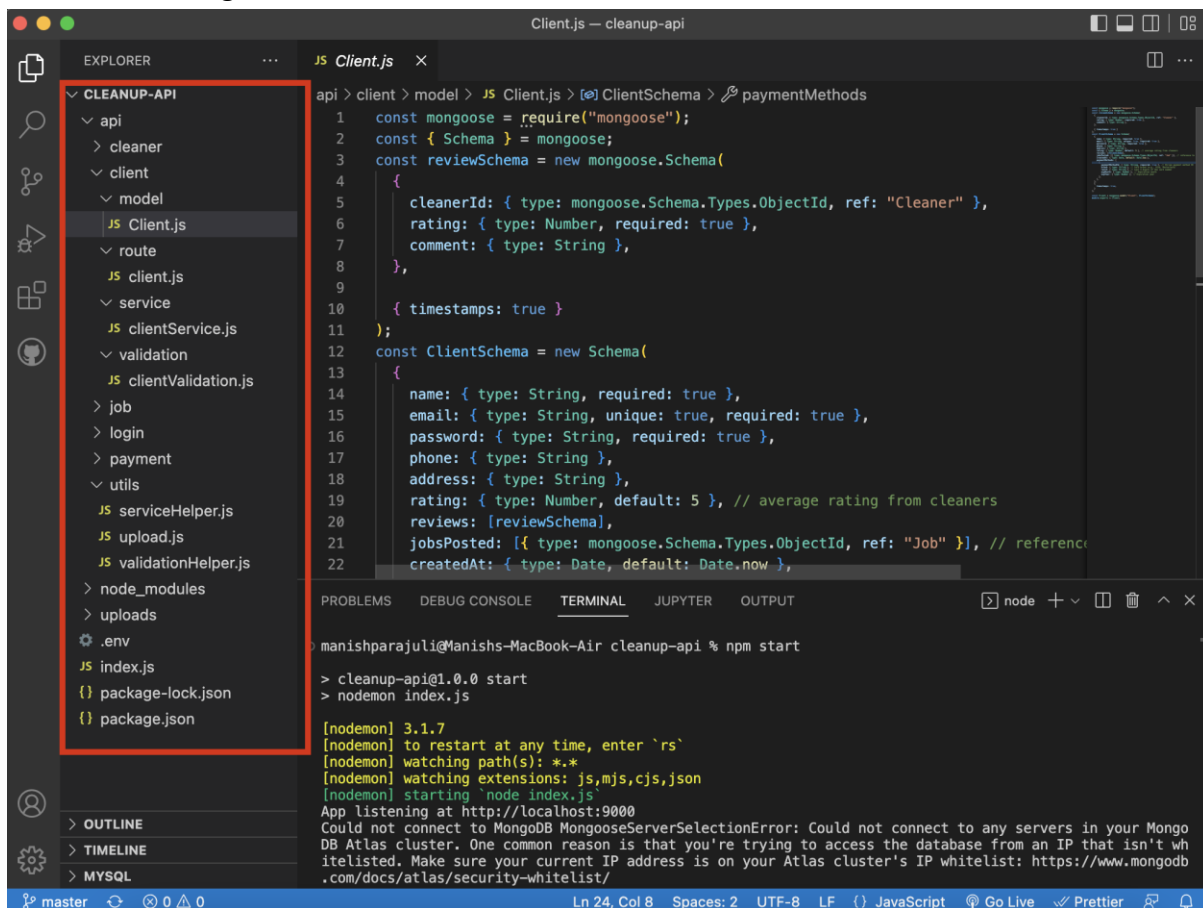
The screenshot displays the MongoDB Atlas 'Network Access' configuration page. The page title is 'Network Access' and it has three tabs: 'IP Access List', 'Peering', and 'Private Endpoint'. The 'IP Access List' tab is active. A yellow warning banner at the top states: 'You will only be able to connect to your cluster from the following list of IP Addresses:'. Below this, there is a table with the following columns: 'IP Address', 'Comment', 'Status', and 'Actions'. The table contains eight rows of IP addresses, all with a status of 'Active'. The 'Actions' column for each row contains 'EDIT' and 'DELETE' buttons. The left sidebar shows the navigation menu with 'Network Access' highlighted under the 'SECURITY' section.

IP Address	Comment	Status	Actions
208.98.222.4/32		Active	EDIT DELETE
99.225.19.65/32 (includes your current IP address)		Active	EDIT DELETE
208.98.222.111/32		Active	EDIT DELETE
199.7.157.95/32		Active	EDIT DELETE
99.229.236.56/32	Created as part of the Auto Setup process	Active	EDIT DELETE
199.119.233.152/32		Active	EDIT DELETE
99.230.16.140/32		Active	EDIT DELETE
208.98.222.63/32		Active	EDIT DELETE
208.98.222.84/32		Active	EDIT DELETE

2. Are you familiar with this code structure?

Ps: Its related to NodeJs

Your Answer: Yes, I am familiar with this Node.js project structure. It follows the MVC pattern, separating models, services, routes, and utilities for modular and scalable API development. I've worked on similar backend projects using Express.js and MongoDB



The screenshot shows a VS Code editor window titled "Client.js — cleanup-api". The Explorer sidebar on the left displays a project structure for "CLEANUP-API". The file tree includes:

- api
 - cleaner
- client
 - model
 - JS Client.js** (selected)
 - route
 - JS client.js
- service
 - JS clientService.js
- validation
 - JS clientValidation.js
- job
- login
- payment
- utils
 - JS serviceHelper.js
 - JS upload.js
 - JS validationHelper.js
- node_modules
- uploads
- .env
- JS index.js
- package-lock.json
- package.json

The main editor area shows the content of "JS Client.js":

```
1 const mongoose = require("mongoose");
2 const { Schema } = mongoose;
3 const reviewSchema = new mongoose.Schema(
4   {
5     cleanerId: { type: mongoose.Schema.Types.ObjectId, ref: "Cleaner" },
6     rating: { type: Number, required: true },
7     comment: { type: String },
8   },
9   { timestamps: true }
10 );
11
12 const ClientSchema = new Schema(
13   {
14     name: { type: String, required: true },
15     email: { type: String, unique: true, required: true },
16     password: { type: String, required: true },
17     phone: { type: String },
18     address: { type: String },
19     rating: { type: Number, default: 5 }, // average rating from cleaners
20     reviews: [reviewSchema],
21     jobsPosted: [{ type: mongoose.Schema.Types.ObjectId, ref: "Job" }], // reference
22     createdAt: { type: Date, default: Date.now },
```

The bottom panel shows the TERMINAL output:

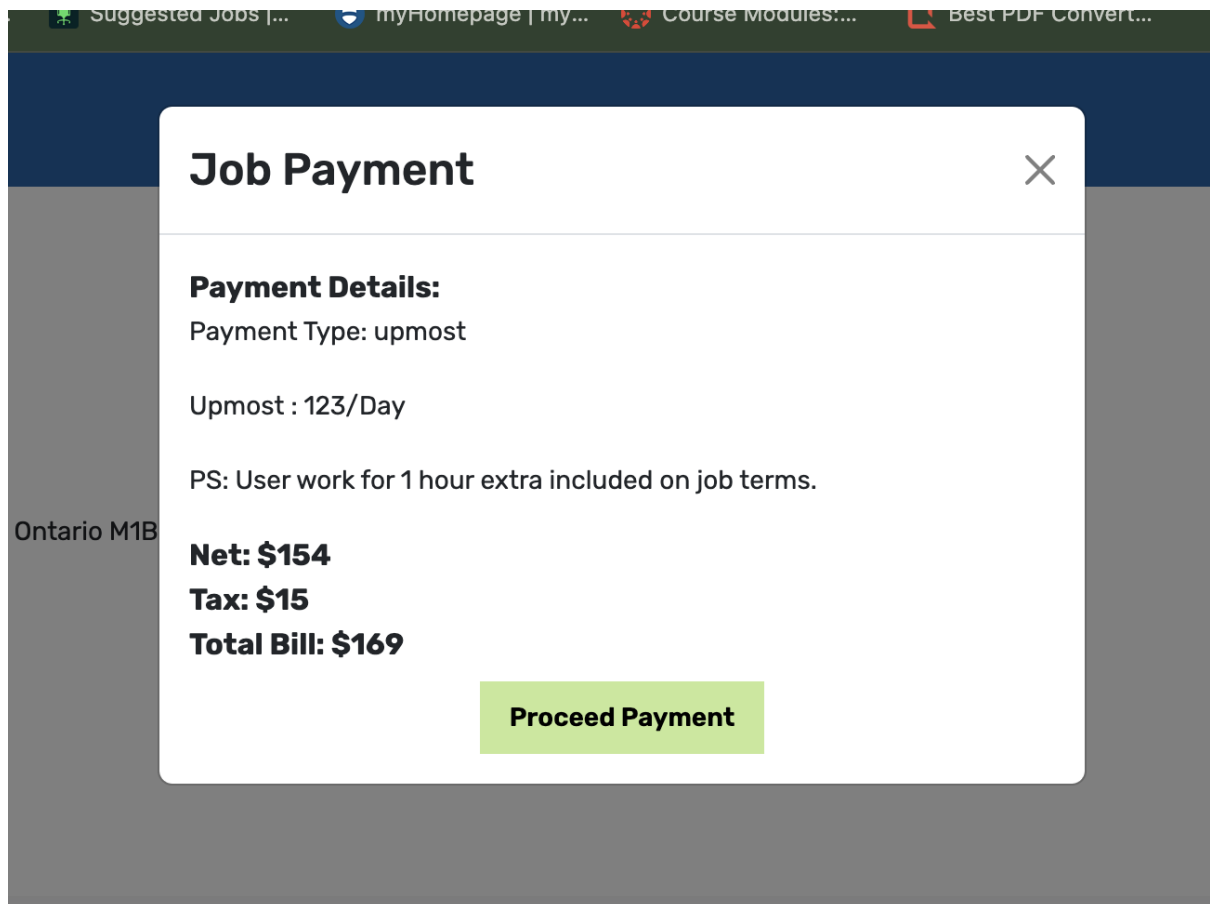
```
manishparajuli@Manishs-MacBook-Air cleanup-api % npm start
> cleanup-api@1.0.0 start
> nodemon index.js

[nodemon] 3.1.7
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting `node index.js`
App listening at http://localhost:9000
Could not connect to MongoDB MongooseServerSelectionError: Could not connect to any servers in your Mongo
DB Atlas cluster. One common reason is that you're trying to access the database from an IP that isn't wh
itelisted. Make sure your current IP address is on your Atlas cluster's IP whitelist: https://www.mongodb
.com/docs/atlas/security-whitelist/
```

3. Can you review this UI?

Ps: Its related to UI/UX

Your Answer: The UI appears clean and minimalistic with job payment data, which is a good start. However, improving spacing, color contrast, and adding consistent typography would enhance user experience. Accessibility and responsiveness should also be tested thoroughly.



4. Are you familiar with git and the following commands ?

<https://education.github.com/git-cheat-sheet-education.pdf>

Your answer: yes, I'm familiar with the given git cheat sheet pdf file git commands.

5. Are you familiar with formik ?

<https://formik.org/docs/overview>

Your answer: Yes, Formik is a popular React library for building and managing forms.

It simplifies form state handling, validation (often paired with Yup), and submission.

Key features include:

Reducing boilerplate code.

Supporting dynamic form fields and error messages.

Integrating seamlessly with React's ecosystem.