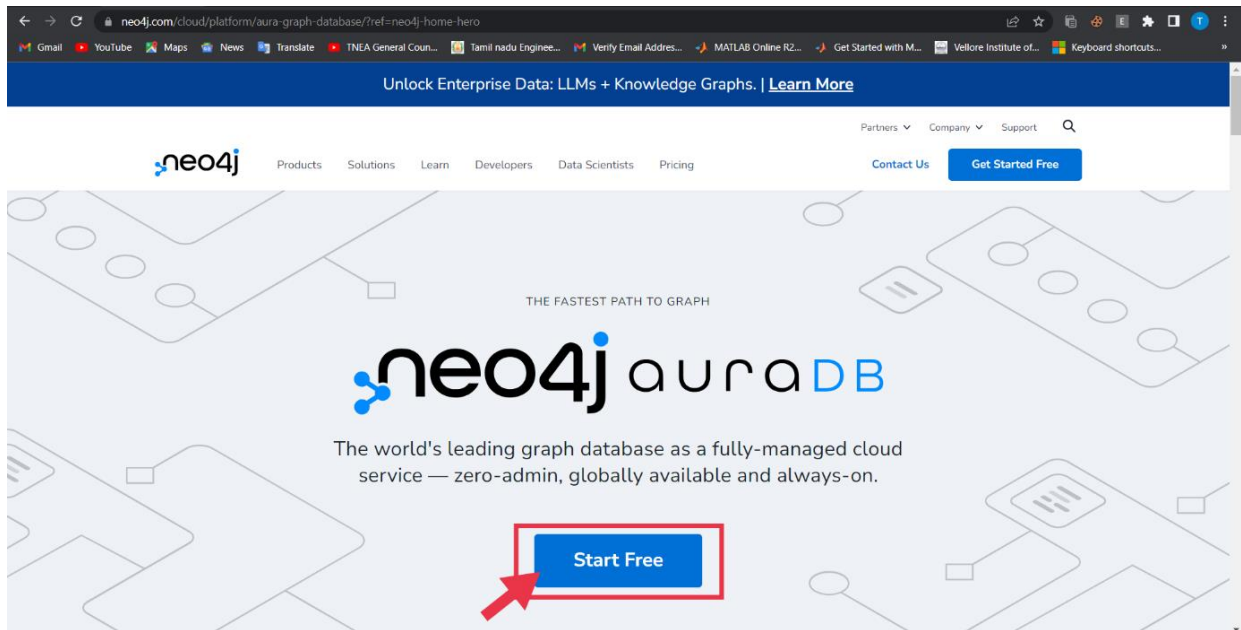


MEDICAL GRAPH VISUALIZER

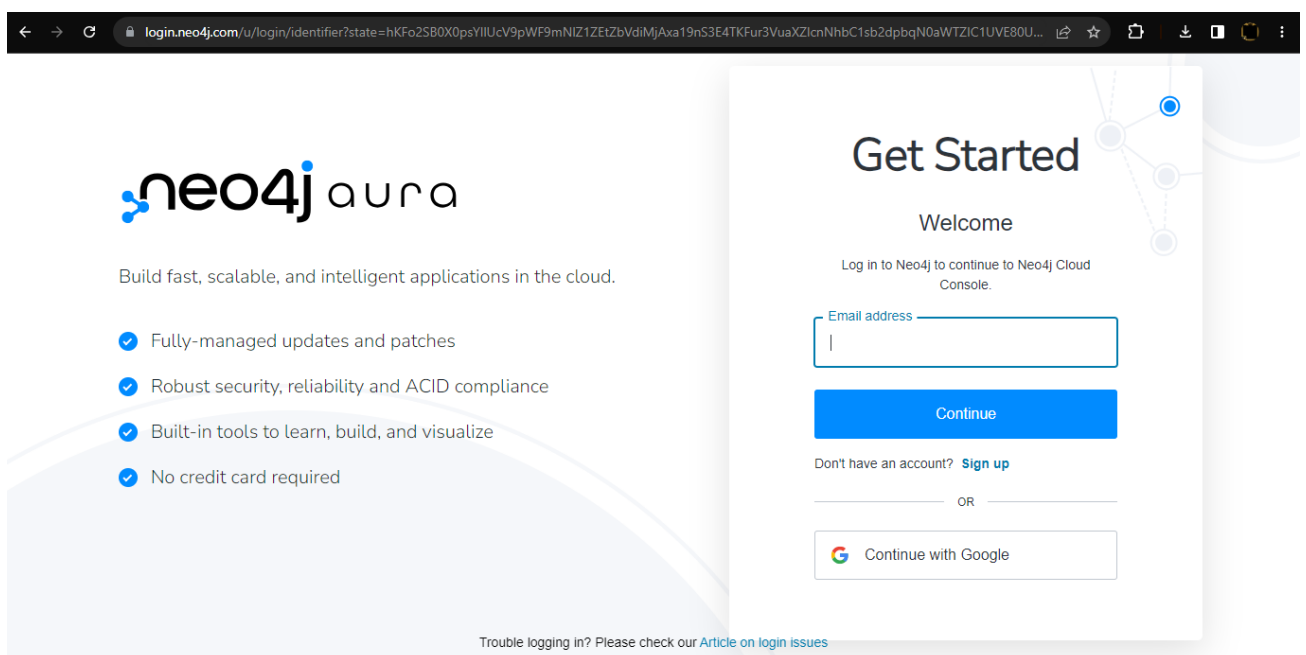
APPLICATION SET-UP GUIDE

Neo4j Aura Login and Graph creation

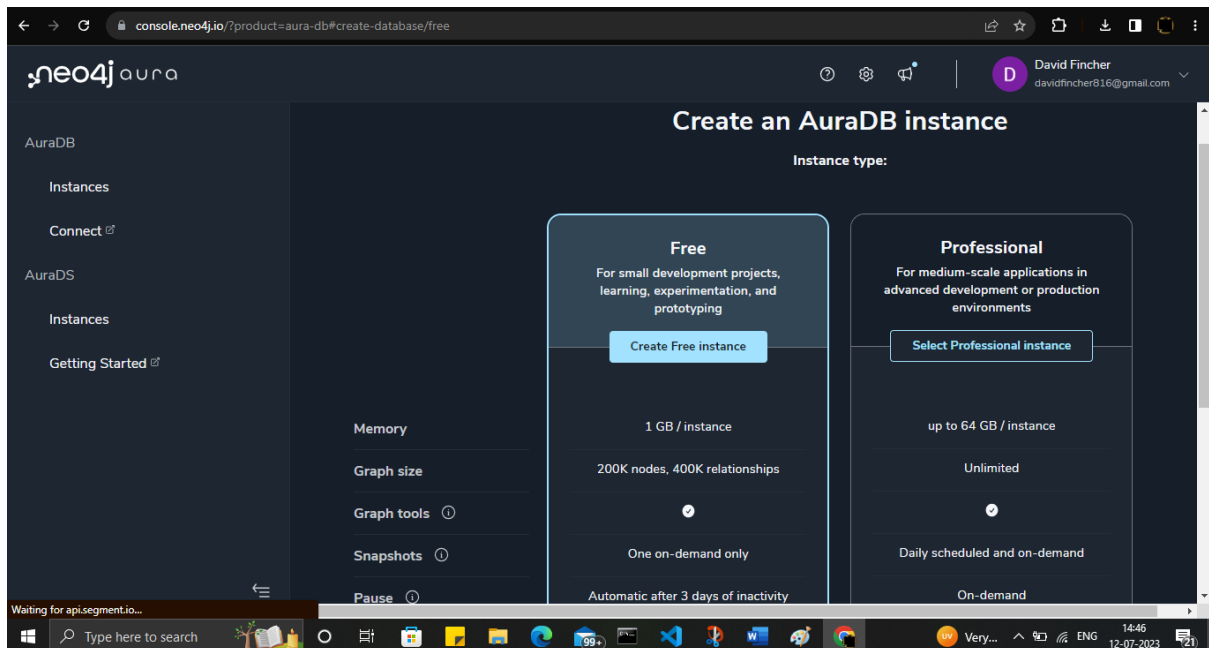
Step 1: Go to [Neo4j Aura Website](https://neo4j.com/cloud/platform/aura-graph-database/) and click “Start Free”



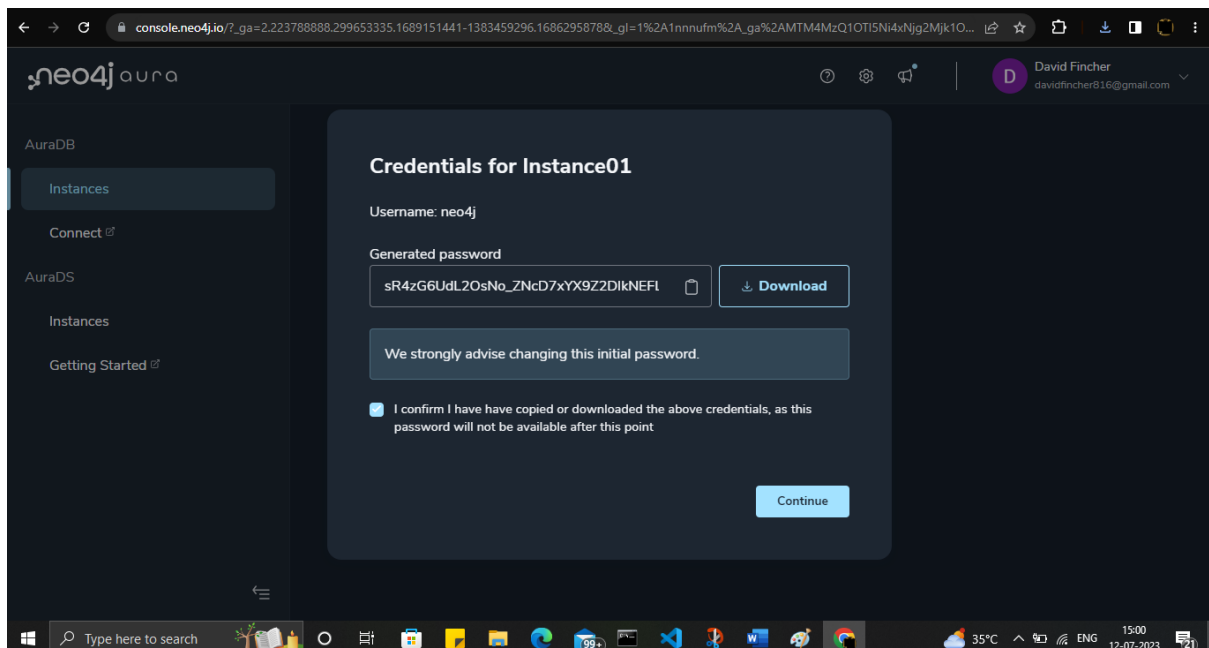
Step 2: Create Neo4j account using your email



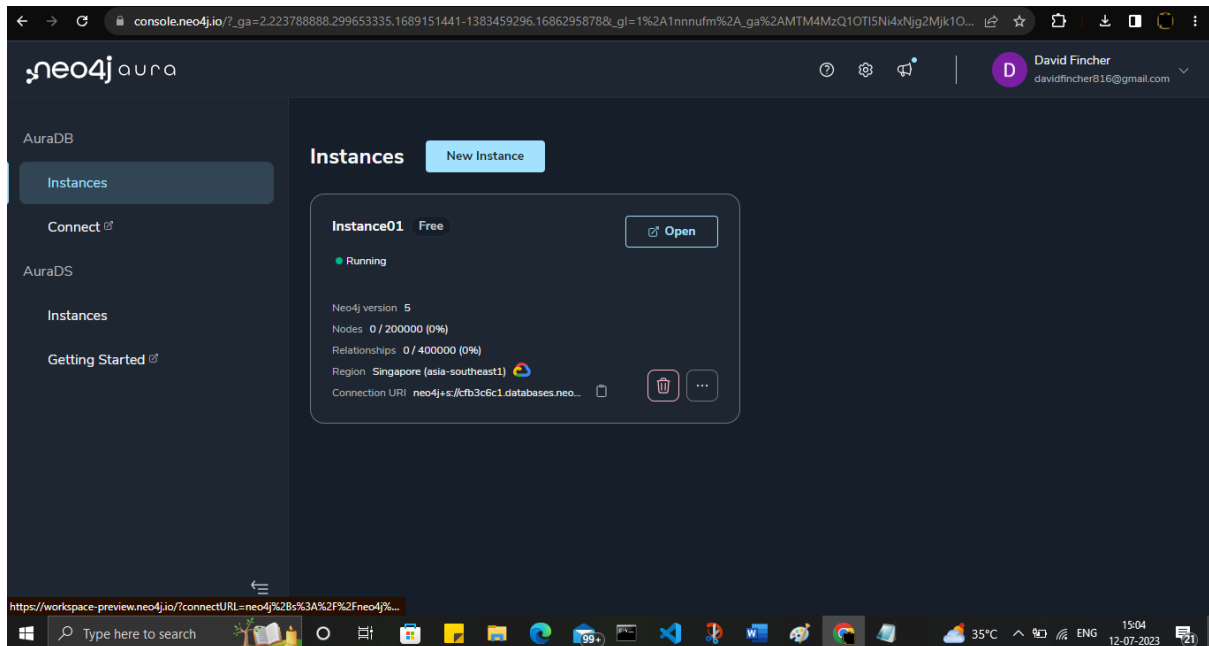
Step 3: After account creation, click on “**Create Free Instance**”



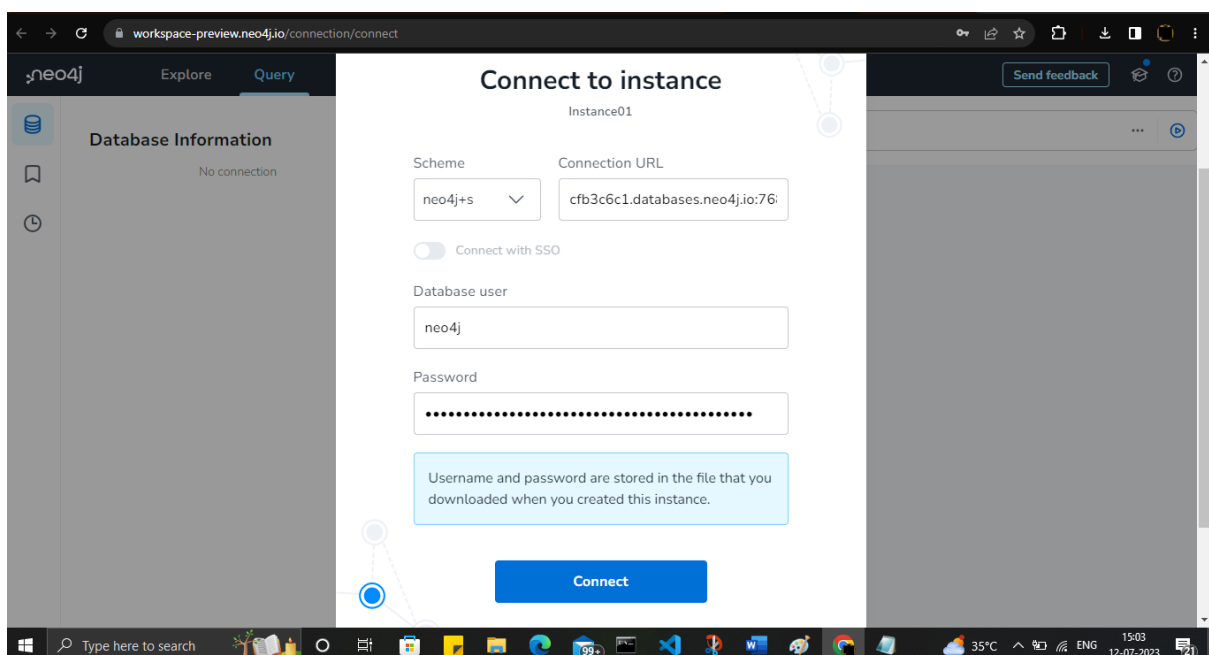
Step 4: The username and password for the instance will be displayed. **It is highly important to download the password before clicking “Continue”.**




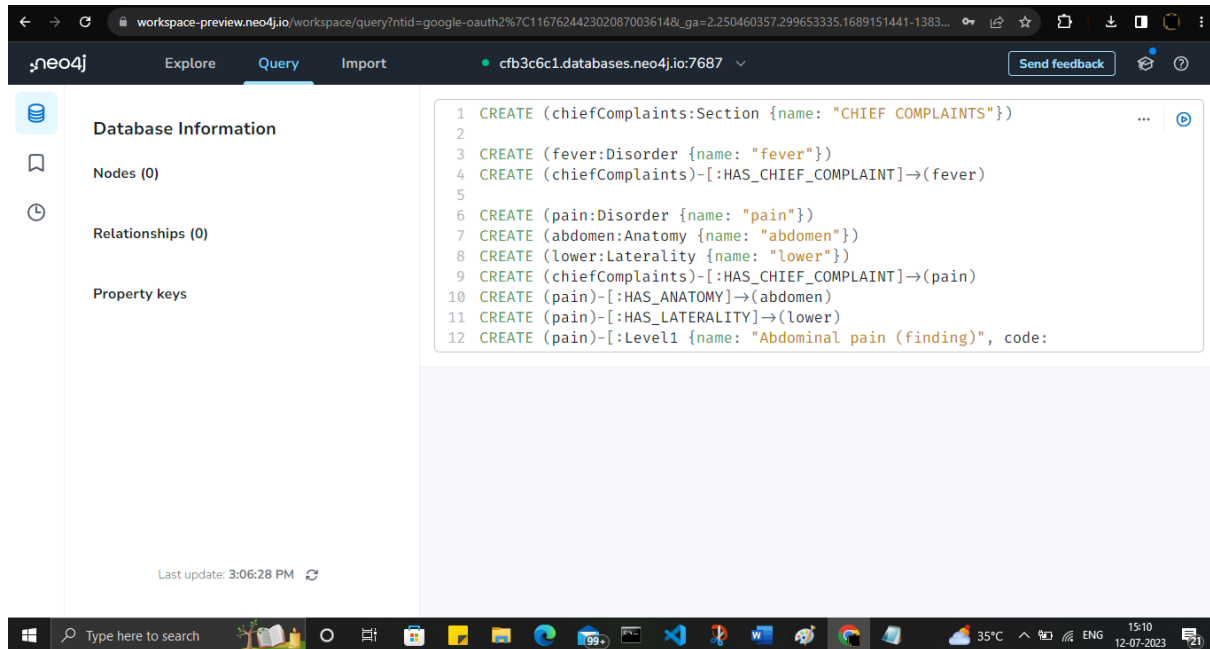
Step 5: A free instance will be created. Ensure that its status is “**Running**”. Now click on “**Open**”.



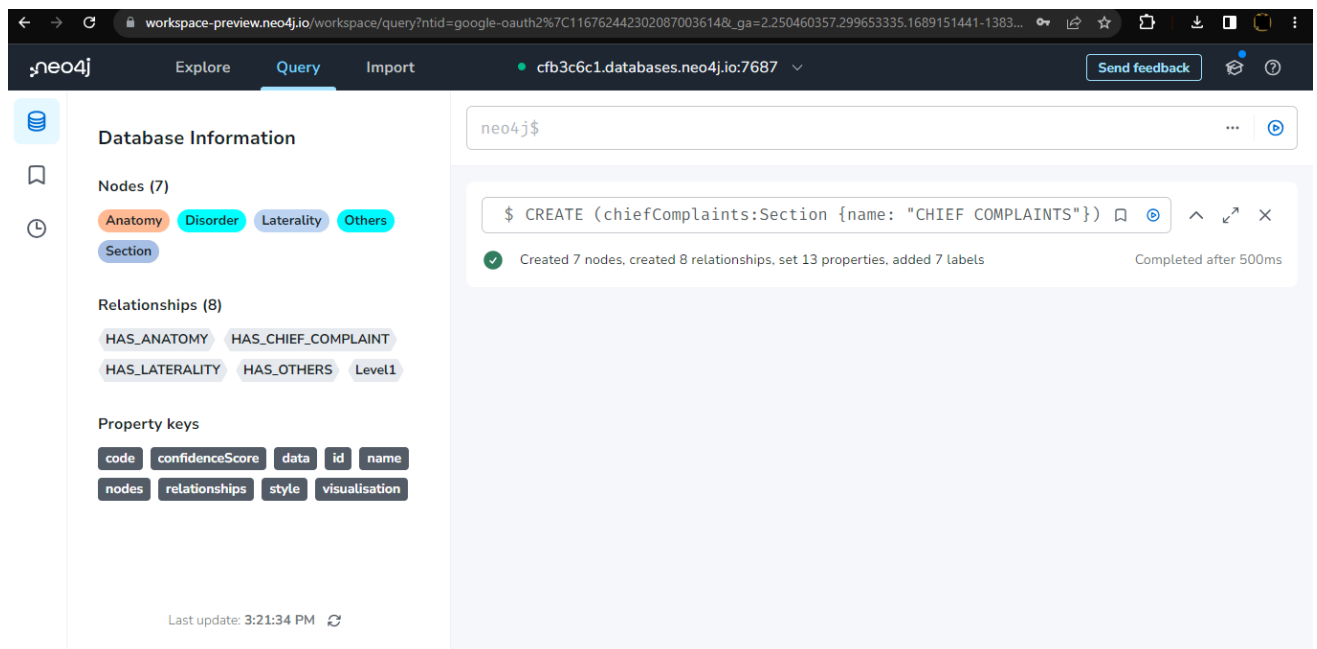
Step 6: The Neo4j workspace will be launched. Use the password that was downloaded earlier and click on “**Connect**”.




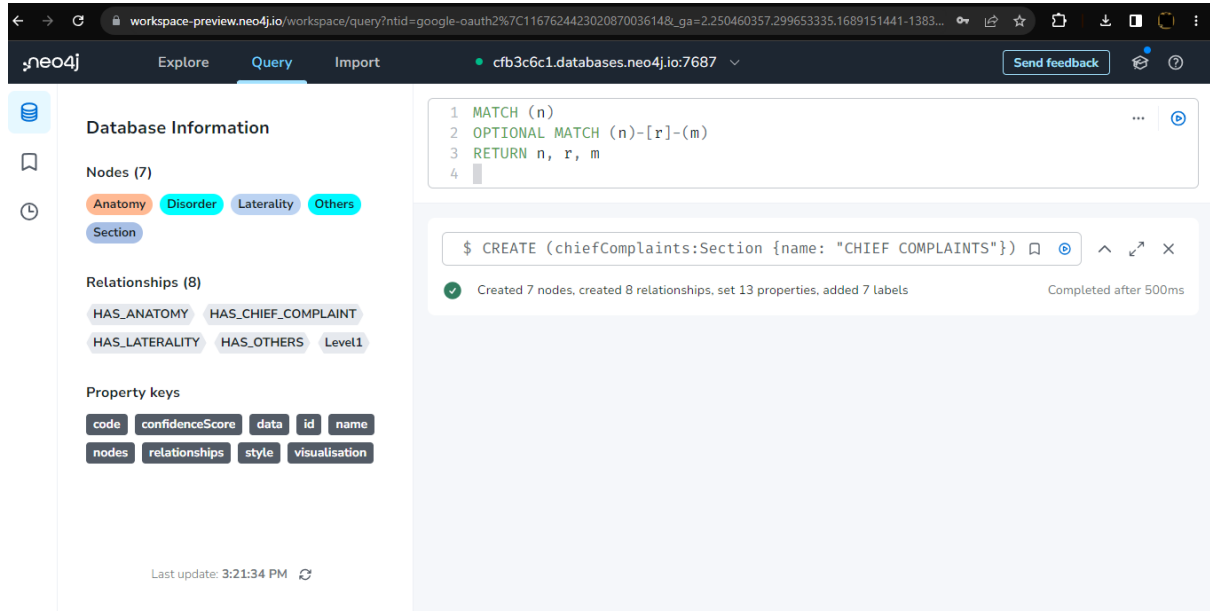
Step 7: The workspace will be opened. Here write the cypher queries for creating the Graph Database with required nodes, edges and relationships and click on  button. The cypher query text file for creation (**CQ_creation.txt**) is attached with the zip file.



Step 8: After executing query successfully, the Graph will be created with nodes and relationships.



Step 9: Then, write the cypher query for visualizing the Graph and click on  button. The cypher query text file for visualizing (**CQ_vis.txt**) is attached with the zip file.

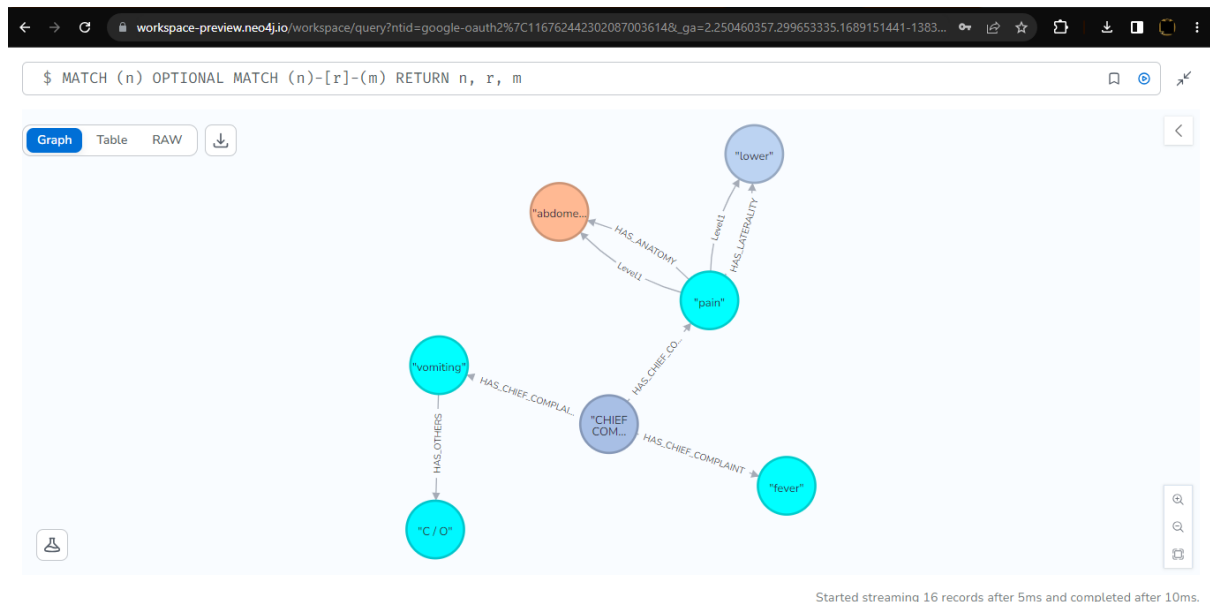


The screenshot shows the Neo4j workspace interface. On the left, the 'Database Information' panel displays details for the database 'cfb3c6c1.databases.neo4j.io:7687'. It lists 7 nodes (Anatomy, Disorder, Laterality, Others) and 8 relationships (HAS_ANATOMY, HAS_CHIEF_COMPLAINT, HAS_LATERALITY, HAS_OTHERS, Level1). Below this, property keys are listed: code, confidenceScore, data, id, name, nodes, relationships, style, and visualisation. The main area shows a Cypher query:

```
1 MATCH (n)
2 OPTIONAL MATCH (n)-[r]-(m)
3 RETURN n, r, m
4
```

 Below the query, a status message indicates: 'Created 7 nodes, created 8 relationships, set 13 properties, added 7 labels. Completed after 500ms'.

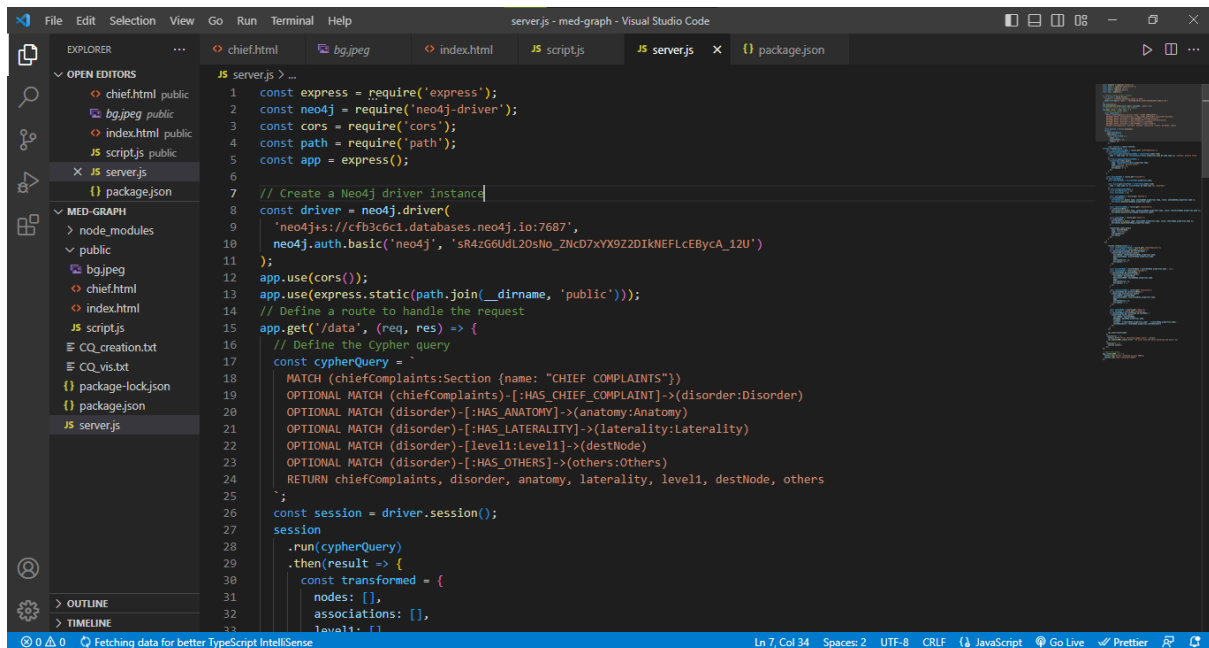
Step 10: After executing the query successfully, the Graph will be visualized.



Unzipping the project folder and deploying

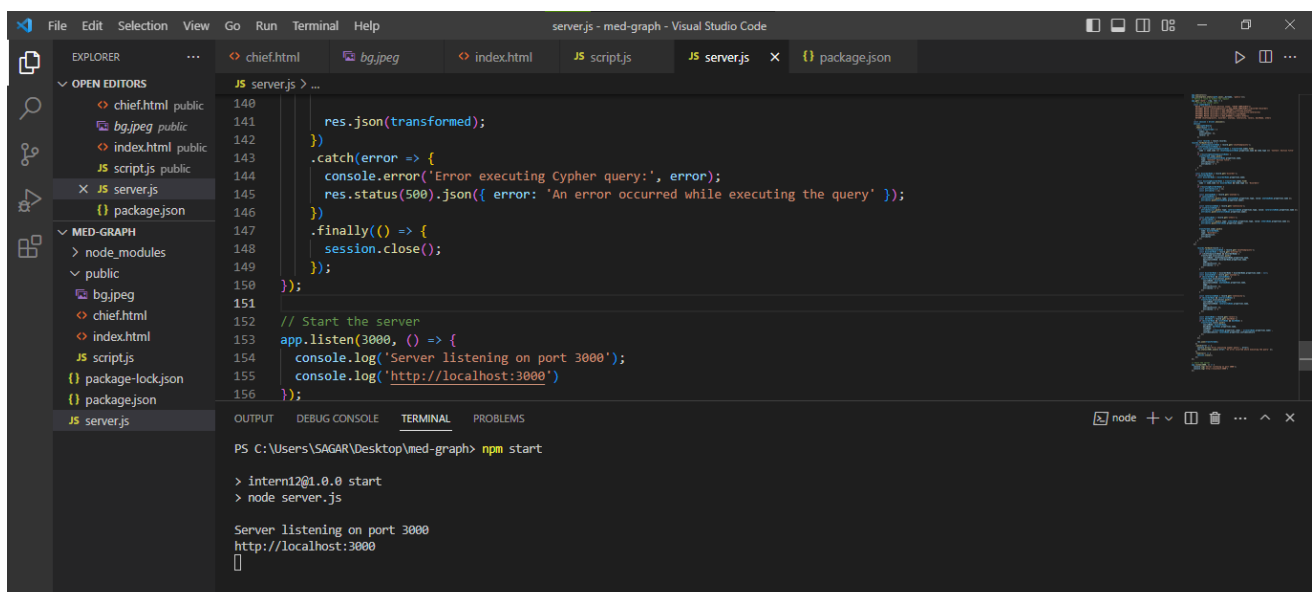
Step 1: After unzipping, launch the project directory in Visual Studio Code and open the server side script file (**server.js**).

This script uses the neo4j database credentials to retrieve the data from the graph using cypher query and pushes it to the client side as json.



```
1 const express = require('express');
2 const neo4j = require('neo4j-driver');
3 const cors = require('cors');
4 const path = require('path');
5 const app = express();
6
7 // Create a Neo4j driver instance
8 const driver = neo4j.driver(
9   neo4j+s://cfb3c6c1.databases.neo4j.io:7687',
10  neo4j.auth.basic('neo4j', 'sR4zG6UdL20sNo_ZNcD7xYX922DIkNEFLcEBycA_12U')
11 );
12 app.use(cors());
13 app.use(express.static(path.join(__dirname, 'public')));
14 // Define a route to handle the request
15 app.get('/data', (req, res) => {
16   // Define the Cypher query
17   const cypherQuery = `
18     MATCH (chiefComplaints:Section {name: "CHIEF COMPLAINTS"})
19     OPTIONAL MATCH (chiefComplaints)-[:HAS_CHIEF_COMPLAINT]->(disorder:Disorder)
20     OPTIONAL MATCH (disorder)-[:HAS_ANATOMY]->(anatomy:Anatomy)
21     OPTIONAL MATCH (disorder)-[:HAS_LATERALITY]->(laterality:Laterality)
22     OPTIONAL MATCH (disorder)-[:level1:Level1]->(destNode)
23     OPTIONAL MATCH (disorder)-[:HAS_OTHERS]->(others:Others)
24     RETURN chiefComplaints, disorder, anatomy, laterality, level1, destNode, others
25   `;
26   const session = driver.session();
27   session
28     .run(cypherQuery)
29     .then(result => {
30       const transformed = {
31         nodes: [],
32         associations: [],
33         level1: []
34       };
35       result.records.forEach(record => {
36         const chiefComplaints = record.get('chiefComplaints');
37         const disorder = record.get('disorder');
38         const anatomy = record.get('anatomy');
39         const laterality = record.get('laterality');
40         const level1 = record.get('level1');
41         const destNode = record.get('destNode');
42         const others = record.get('others');
43         transformed.nodes.push(chiefComplaints);
44         transformed.associations.push(disorder);
45         transformed.associations.push(anatomy);
46         transformed.associations.push(laterality);
47         transformed.associations.push(destNode);
48         transformed.associations.push(others);
49         transformed.level1.push(level1);
50       });
51       res.json(transformed);
52     })
53     .catch(error => {
54       console.error('Error executing Cypher query:', error);
55       res.status(500).json({ error: 'An error occurred while executing the query' });
56     })
57     .finally(() => {
58       session.close();
59     });
60 });
61
62 // Start the server
63 app.listen(3000, () => {
64   console.log('Server listening on port 3000');
65   console.log('http://localhost:3000');
66 });
```

Step 2: Open the Terminal in VS code and type the command **“npm start”** and click enter. The node server will be started and website will be hosted in local host with port 3000. (<http://localhost:3000>)



```
PS C:\Users\SAGAR\Desktop\med-graph> npm start

> intern12@1.0.0 start
> node server.js

Server listening on port 3000
http://localhost:3000
```

Step 3: Ctrl + Click on the address in the terminal (<http://localhost:3000>) to launch the web application. Click on the View button in the Chief complaints grid to traverse to the Chief complaints graph visualization page.

