**Assignment – 11**

Name : Om Kulkarni Date : 18/04/25

Class : TY CSE

PRN : 22510034 Batch : T-3

**Steps to follow :**1) Install neo4j desktop on the system from the below link :

<https://neo4j.com/download/>

Follow the steps for installation and make sure to create a new local DBMS and connect to it and also place the extracted .csv files into import folder of the created project so that the neo4j browser accesses it easily.

2)Download the CORA research paper classification dataset and extract the files in it.

<https://people.cs.umass.edu/~mccallum/data.html>

Convert all the files into .csv except for the actual citations or the research papers.

3) Design the Graph Data Model

**Nodes:**

Author (name)

Paper (title, id)

Classification (name)

**Relationships:**

(:Author)-[:WROTE]->(:Paper)

(:Paper)-[:CITES]->(:Paper)

(:Paper)-[:HAS\_CLASSIFICATION]->(:Classification)

(:Classification)-[:SUBCLASS\_OF]->(:Classification) (for hierarchy)

4) Neo4j queries for creating the appropriate relationships

// Create constraints (optional, but good practice)

CREATE CONSTRAINT FOR (a:Author) REQUIRE a.name IS UNIQUE;

CREATE CONSTRAINT FOR (p:Paper) REQUIRE p.id IS UNIQUE;

CREATE CONSTRAINT FOR (c:Classification) REQUIRE c.name IS UNIQUE;

// Load Authors (example)

LOAD CSV WITH HEADERS FROM 'file:///authors.csv' AS row

MERGE (a:Author {name: row.name});

// Load Papers

LOAD CSV WITH HEADERS FROM 'file:///papers.csv' AS row

MERGE (p:Paper {id: row.id, title: row.title});

// WROTE relationship

LOAD CSV WITH HEADERS FROM 'file:///wrote.csv' AS row

MATCH (a:Author {name: row.author}), (p:Paper {id: row.paper})

MERGE (a)-[:WROTE]->(p);

// CITES relationship

LOAD CSV WITH HEADERS FROM 'file:///citations.csv' AS row

MATCH (p1:Paper {id: row.source}), (p2:Paper {id: row.target})

MERGE (p1)-[:CITES]->(p2);

// Classifications

LOAD CSV WITH HEADERS FROM 'file:///classifications.csv' AS row

MERGE (c:Classification {name: row.name});

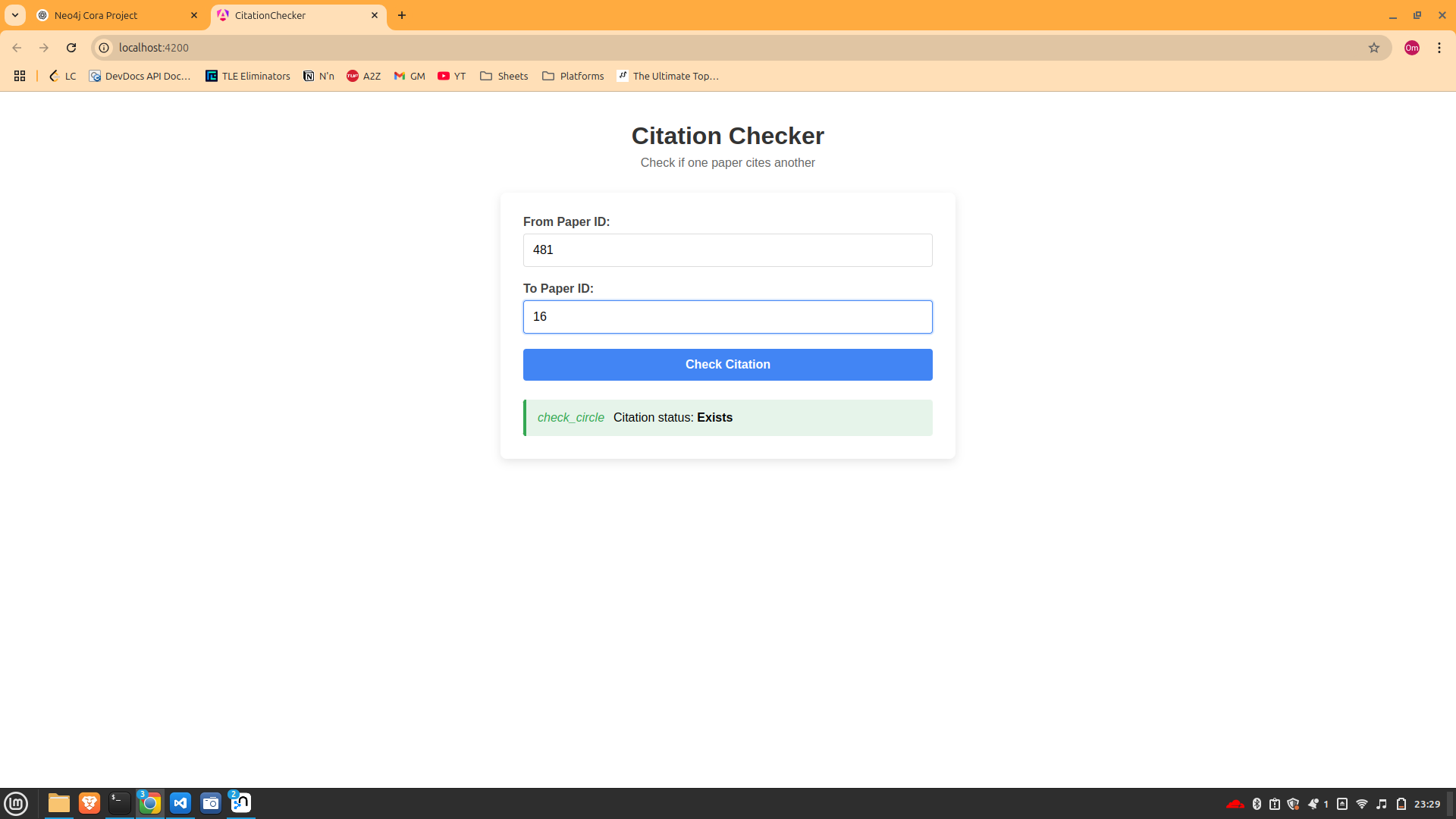
// HAS\_CLASSIFICATION

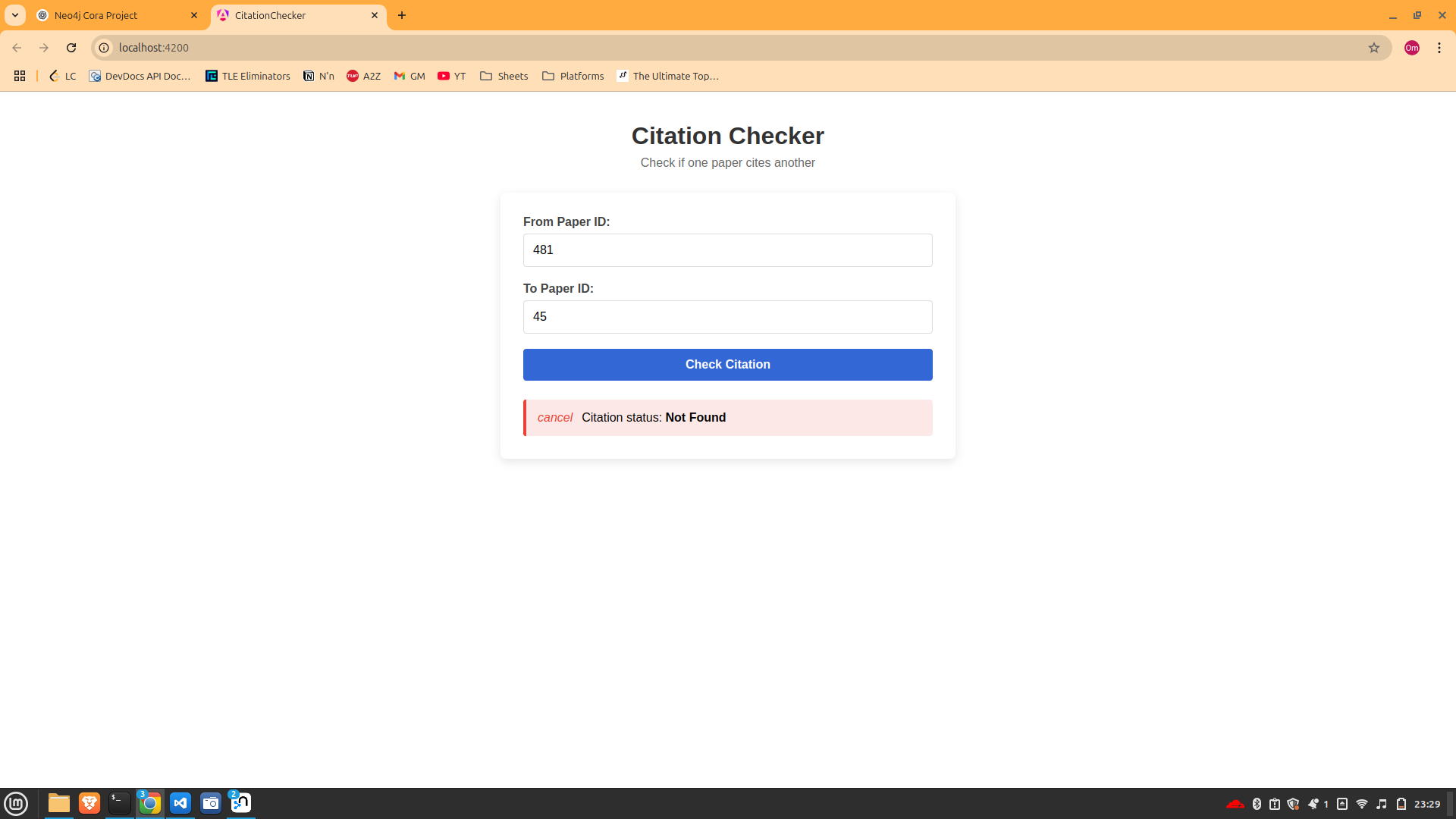
LOAD CSV WITH HEADERS FROM 'file:///paper\_class.csv' AS row

MATCH (p:Paper {id: row.paper}), (c:Classification {name: row.class})

MERGE (p)-[:HAS\_CLASSIFICATION]->(c);

This is the frontend angular form to check whether a paper cites another paper or not :





This is the neo4j part

