KATHMANDU UNIVERSITY

SCHOOL OF ENGINEERING

DEPARTMENT OF GEOMATICS ENGINEERING

REPORT ON

FINDING SHORTEST ROUTE FROM

DHANGADHI TO KATHMANDU UNIVERSITY

USING PgRouting

Submitted by: Aarya Pant (029022-21)

Level: UNG/ III Year/ I Semester

14th June 2024

Introduction

The problem of determining the shortest path is currently very interesting to discuss in everyday life. This problem is related to finding the shortest route and the minimum budgeted costs. Unintentionally, we experience this problem almost every day. For example, traveling from one place to another, telephone networks, electricity networks and so on. Another example is the online bikes and taxies services which is currently very popular in Nepal. Online bikes and taxies trips always look for the shortest path from the starting city point to the final destination city point. This is so that every online bike or taxi driver can quickly get customers according to the target set by the company every day. There it appears that information technology is increasingly advanced and sophisticated today. Therefore, many people who want to travel by looking for the shortest route or distance to reach the trip using technology (Alam & Faruq, 2019; CHOOSUMRONG et al., 2012; Nurhasanah et al., 2021; pgRouting Community, 2016; Pritee et al., 2019; Seeta, 2021; Singh et al., 2015). So, here we use Dijkstra's algorithm to find the shortest distance. Dijkstra's algorithm calculates the path based on the shortest distance travelled. For our project we also used PgRouting (an extension for PostgreSQL).

Geographic information systems (GISs) are becoming the most popular field in recent years. A GIS is an application or system which is designed to capture, storage, manipulation, analysis, and presentation of spatial or geographic information. Geographic location is the key term or information for the geographic information without which the data can't be said to be spatial or geographic. This project explores the working of QGIS and Databases technologies to find out the shortest path between two places (source and destination) in road network (geospatial data) using various algorithm approaches of PgRouting [extension of PostgreSQL database]. In this project, Dijkstra's algorithm is implemented to calculate the best and an optimal shortest path between two nodes(CHOOSUMRONG et al., 2012).

The purpose of this project is to find the shortest route from Dhangadhi to Kathmandu University using PgRouting. This study aims to provide an efficient route planning solution that can be utilized for various applications, including logistics and transportation.

Dijkstra's Algorithm

Dijkstra's algorithm is a popular algorithm in computer science used for finding the shortest path between nodes in a graph. It's named after Dutch computer scientist Edsger W. Dijkstra, who developed it in 1956 (Alam & Faruq, 2019).

(Seeta, 2021) The result of Dijkstra's algorithm is the shortest path tree from the source node to all other reachable nodes in the graph, with the minimum distance from the source node to each node. It's often used in routing and network algorithms, such as finding the shortest route in transportation networks or the optimal path in communication networks. The working of this algorithm has been explained using six steps:

I. Choose search region providing set S of vertices and select source and destination vertices.
II. Initialize the source node to zero which is solved and identify rest unsolved nide connected to source node which is assumed as origin.
III. Calculate candidate distance by adding distance to the solved node and length of arc connecting between solved and unsolved nodes.
IV. After that algorithm chooses smallest candidate distance.
V. Then unsolved is changed into solved node.
VI. If there will be tie in candidate distance, then choose arbitrary.
VII. Repeat these steps with all unsolved node until the destination is reached.

PgRouting

PgRouting is an extension for PostgreSQL, the open-source object-relational database system. It adds routing capabilities to PostgreSQL(one of the Structured Query Language specially used for Relational Databases), allowing users to perform routing queries and computations within the database. This extension is particularly useful for applications that require routing functionality, such as navigation systems, logistics, and transportation planning (Pritee et al., 2019). PgRouting provides a set of functions and algorithms for various routing tasks, including shortest path calculation, traveling salesman problem solving, and network analysis. It utilizes graph theory and algorithms like Dijkstra's algorithm and A* search to find the most efficient paths between points on a network. With PgRouting, users can store and manage spatial data in PostgreSQL databases and perform routing computations directly within the database, leveraging the power and flexibility of SQL queries. This integration simplifies the development and maintenance of routing applications by centralizing data storage and processing (Singh et al., 2015).

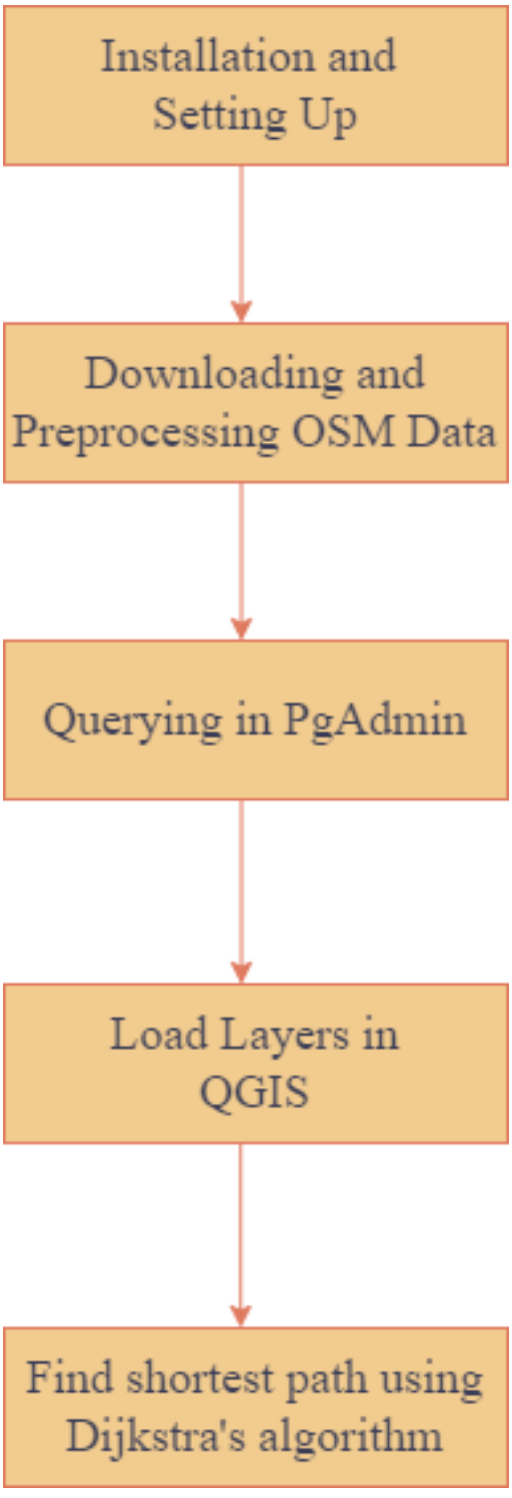
Advantages of the database routing approach are:

Data and attributes can be modified by many clients, like Quantum GIS and uDig through JDBC, ODBC, or directly using Pl/pgSQL. The clients can either be PCs or mobile devices. Data changes can be reflected instantaneously through the routing engine. There is no need for pre calculation. The “cost” parameter can be dynamically calculated through SQL and its value can come from multiple fields or tables. PgRouting provides functions for: Shortest Path Dijkstra: routing algorithm without heuristics; Shortest Path A-Star: routing for large datasets (with heuristics); Shortest Path Shooting-Star: routing with turn restrictions (with heuristics), Traveling Salesperson Problem, Driving Distance Calculation (Isolines)(Nurhasanah et al., 2021).

This project encompasses the development and implementation of an efficient route planning solution using Geographic Information Systems (GIS) and PgRouting, a powerful extension of the PostgreSQL database. The scope includes the acquisition and management of road network data within QGIS, the application of Dijkstra’s algorithm via PgRouting to determine the shortest path between Dhangadhi and Kathmandu University, and the evaluation of this method's performance in dynamic, real-world scenarios. By leveraging dynamic cost calculations, this project aims to adapt to real-time variations in the road network, such as traffic conditions and road closures. The outcome will be a robust and scalable solution for optimal route planning, applicable to various fields including transportation, demonstrating the practical benefits of integrating advanced geospatial data analysis with database management systems.

The study area for this project is the road network between Dhangadhi and Kathmandu University in Nepal. This region encompasses a diverse range of geographic and infrastructural features, making it an ideal candidate for evaluating the effectiveness of GIS-based routing solutions. Dhangadhi is located in the farwestern part of Nepal, Dhangadhi is one of the major cities in the Sudurpashchim Province, characterized by a mix of urban and semi-urban road networks whereas Kathmandu University is situated in Dhulikhel, approximately 30 kilometers east of Kathmandu, the university represents a significant endpoint in the hilly terrain of central Nepal.

Workflow


Installation and Setting Up
Downloading and Preprocessing OSM Data
Querying in PgAdmin
Load Layers in QGIS
Find shortest path using Dijkstra's algorithm

Description

1. Database download:

The installation of PostgreSQL had to be done. PostGIS is a spatial extension for PostgreSQL. It can be installed using the link provided above. Also, After installing PostgreSQl on your system, PostGIS can be installed using Stack Builder Application. PgRouting is included with the PostGIS bundle.

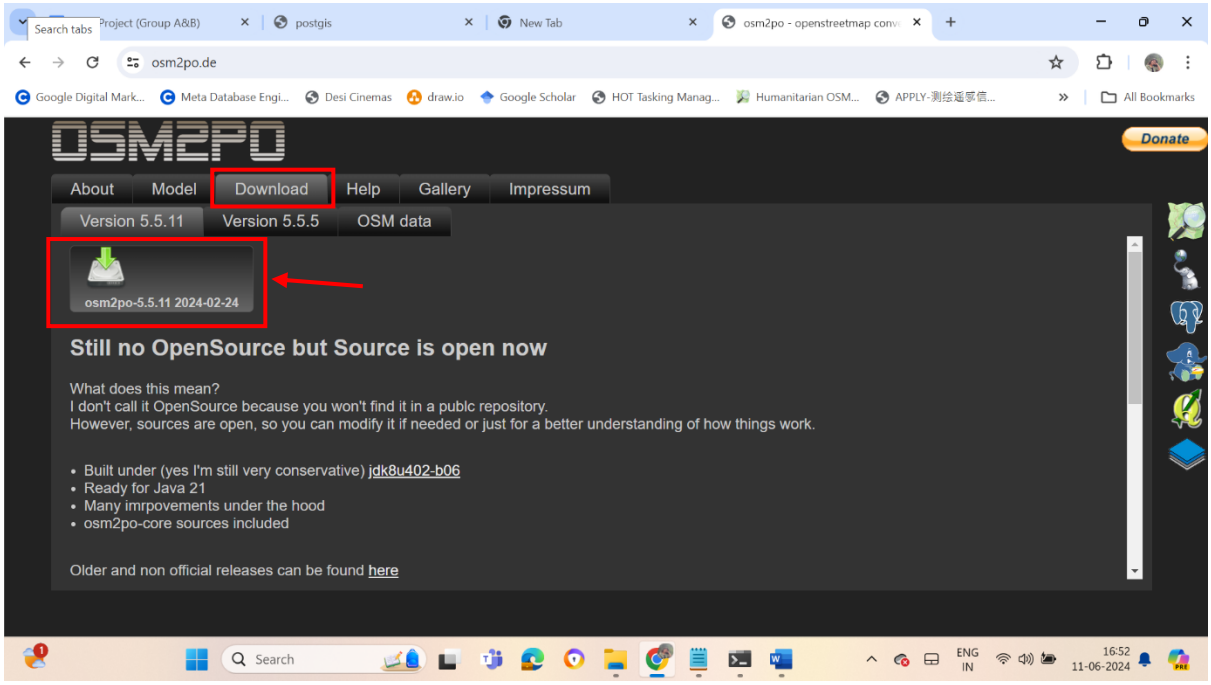
2. GIS:

This Open-Source GIS applicaton was to be downloaded.

3. osm2po:

The osm2po package was downloaded from the link provided by the lecturer. This package takes as input the raw OSM data (eg. .PBF format) and converts the data into a SQL file which can be imported into our Spatial Database and used for the pgRouting implementation. The following steps can be followed for the installation of osm2po.

- Visit: https://osm2po.de/
- Download: osm2po-5.5.11

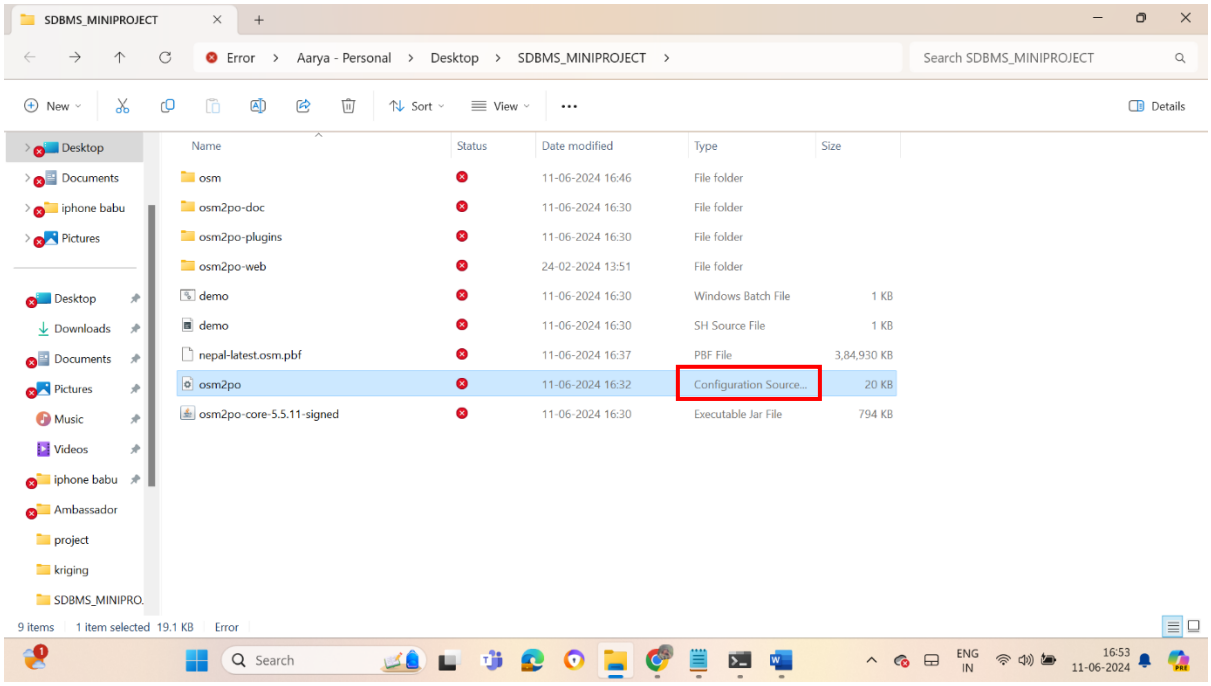


4. Extracting the zipped file:

- Extract files from: osm2po-5.5.11

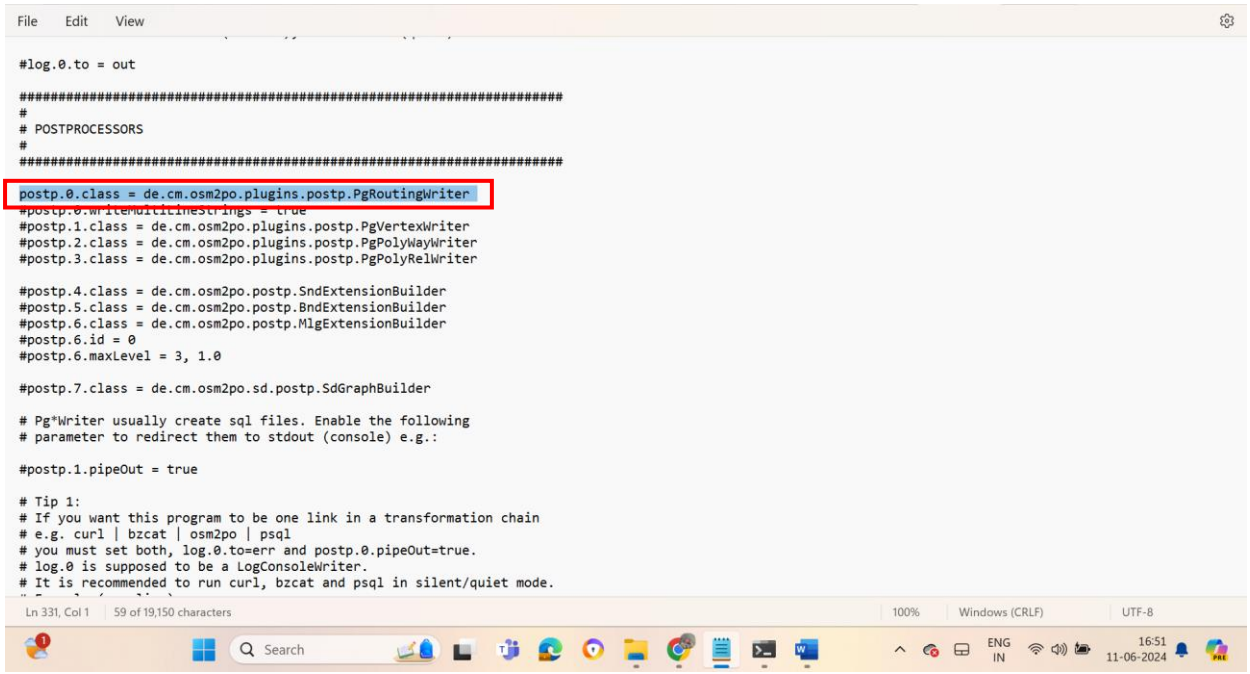
5. Editing the data

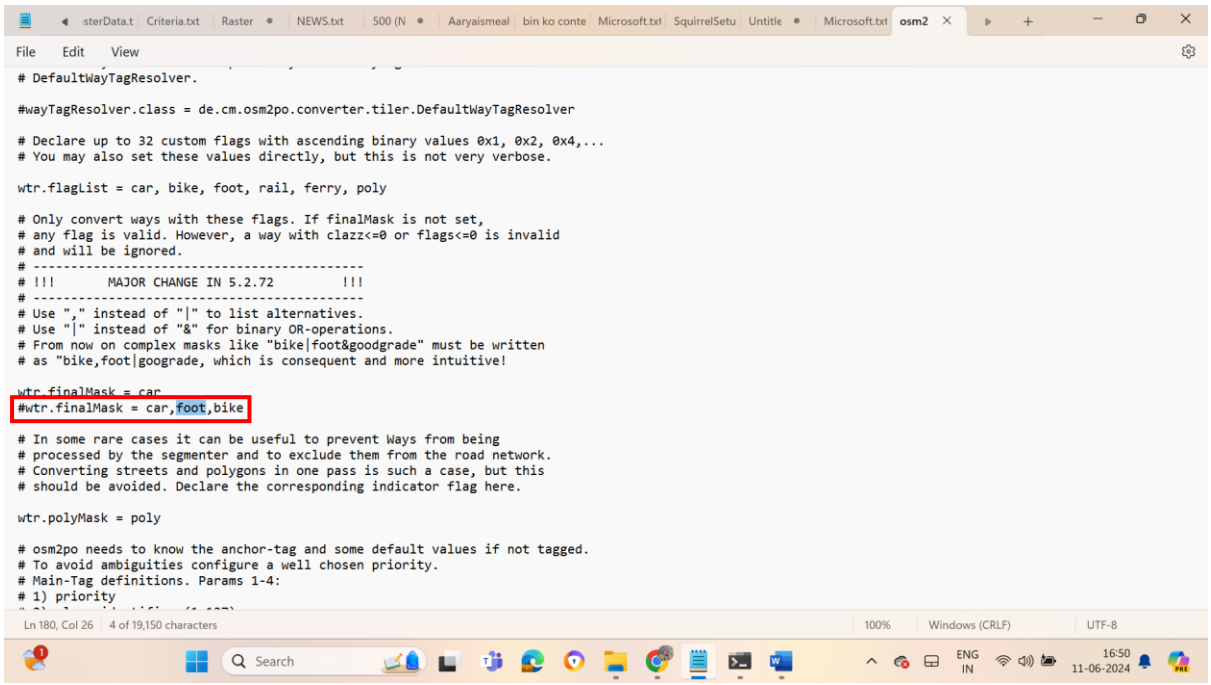
- osm2po.config was to be opened using Notepad



Editing was to be done on:

- Line 330: Removed # in the beginning
- Line 179: wtr.finalMask = car,foot,bike





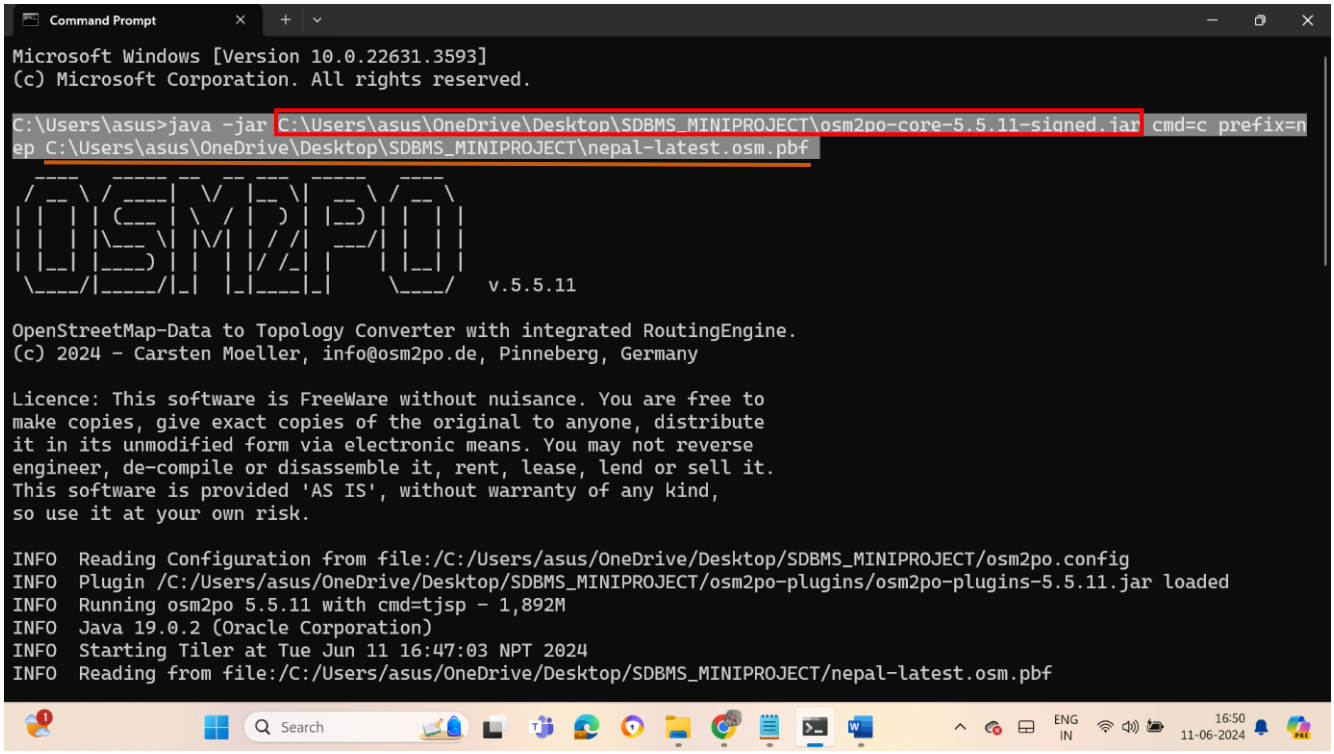
6. Pre-processing OSM Data in osm2po:

- Open Command Prompt and type the following command:

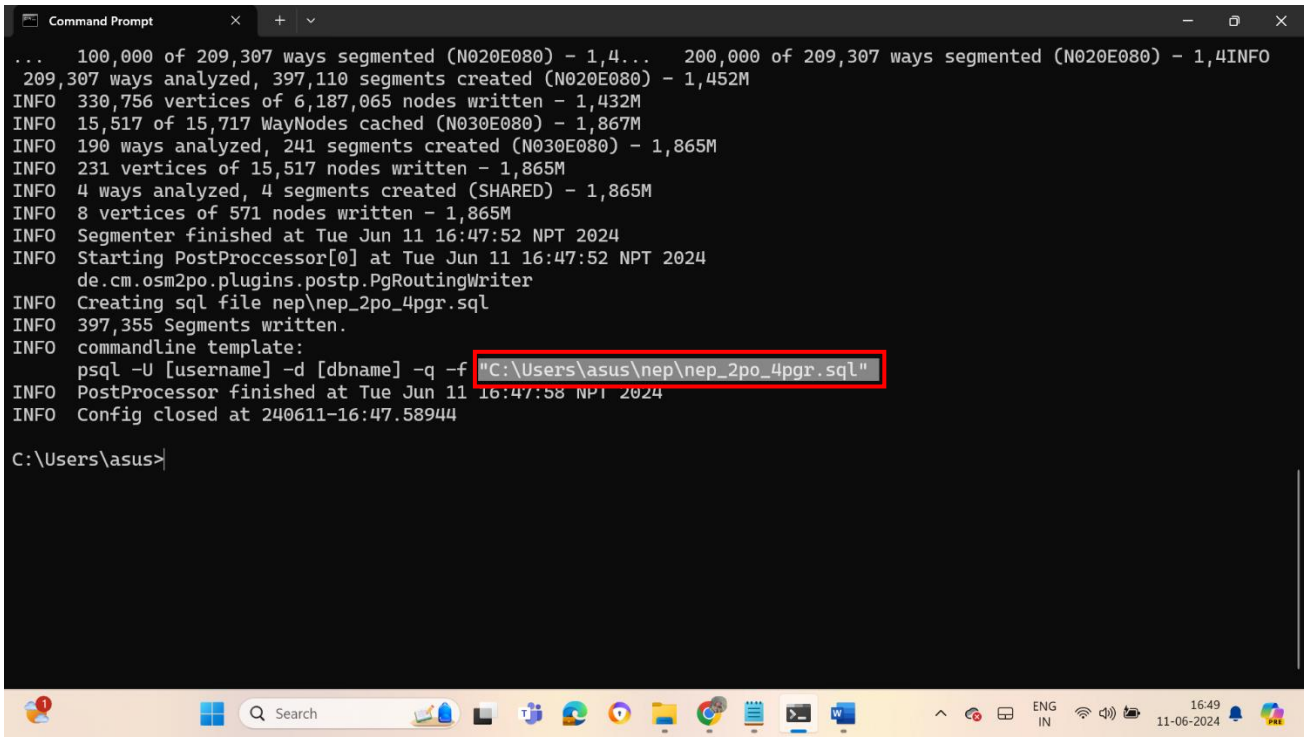
C:\Users\asus>java -jar C:\Users\asus\OneDrive\Desktop\SDBMS_MINIPROJECT\osm2po-core-5.5.11-signed.jar cmd=c prefix=nep C:\Users\asus\OneDrive\Desktop\SDBMS_MINIPROJECT\nepal-latest.osm.pbf

Location of osm2po file

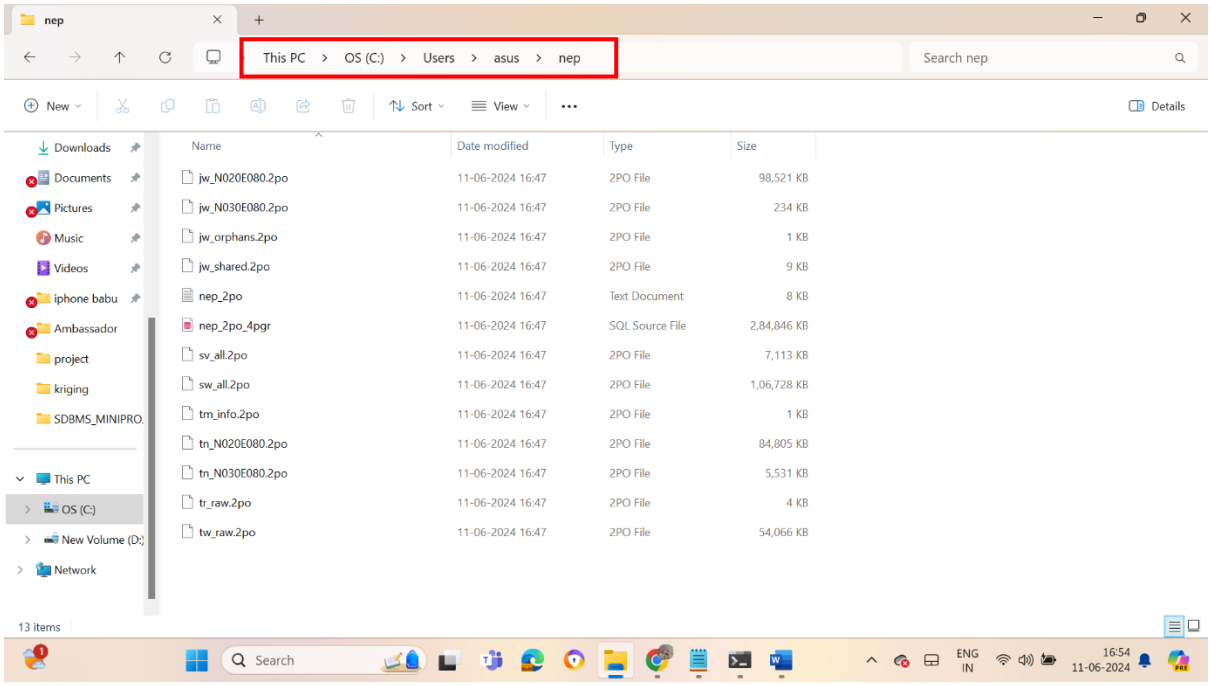
Location of osm data file



- Navigate to the converted file.

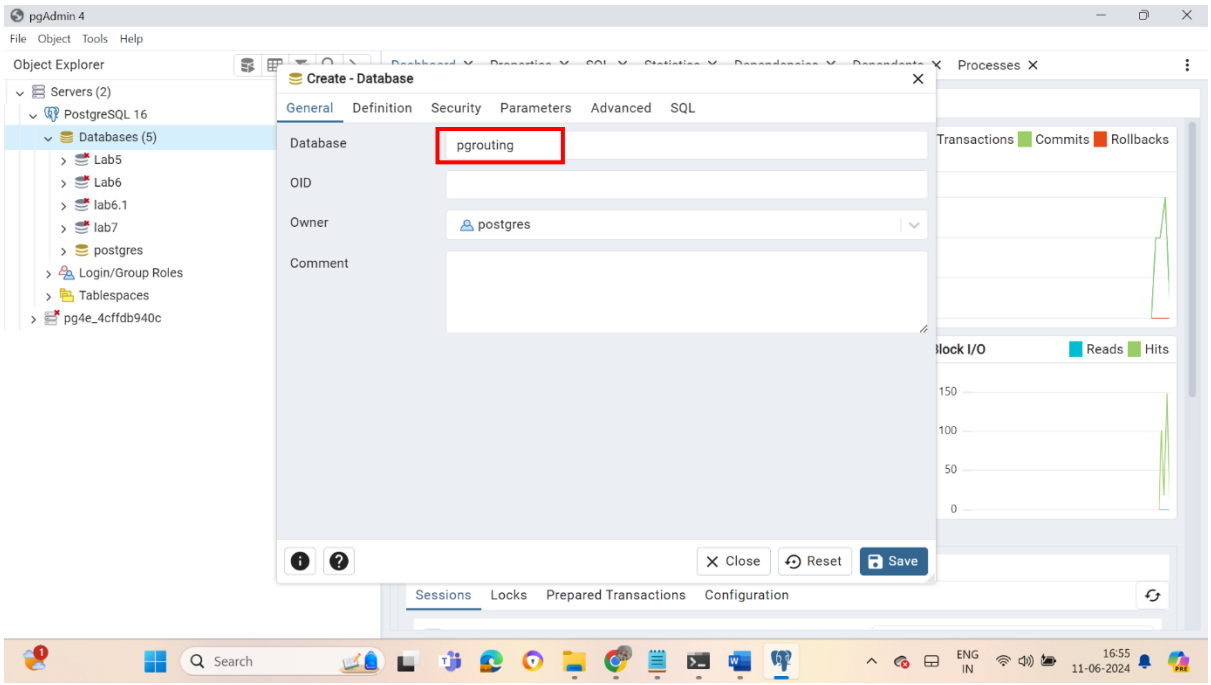


- Navigate to the converted file in Windows Explorer.

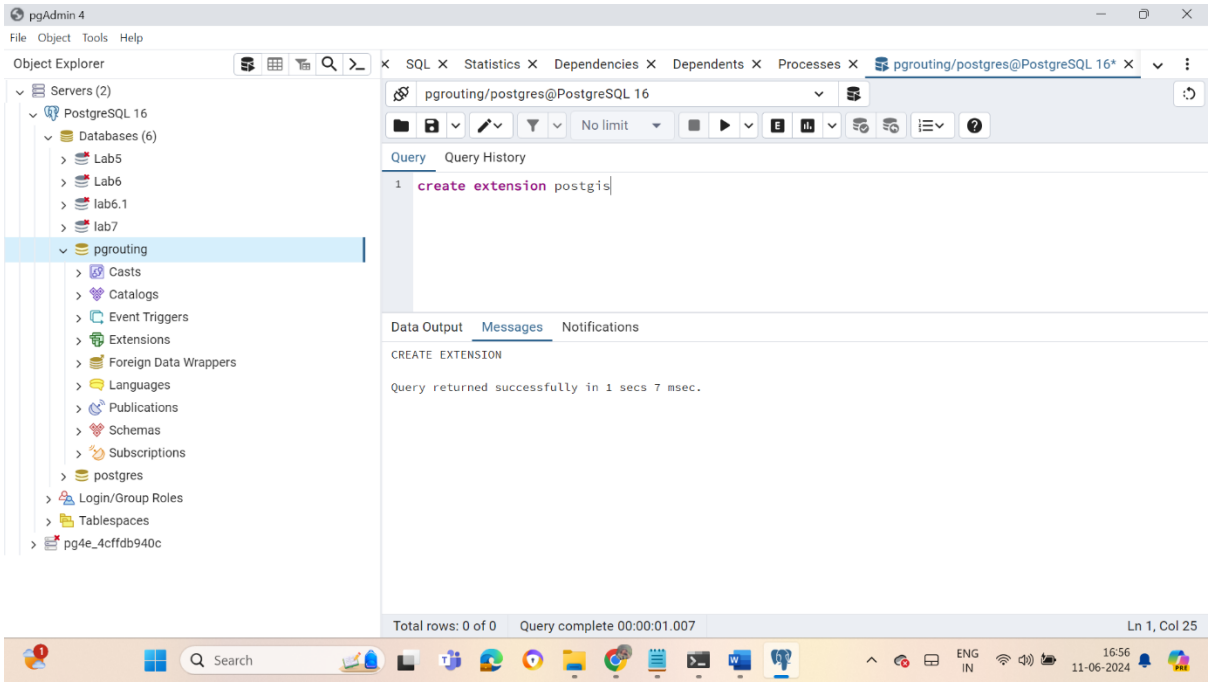


7. pgAdmin:

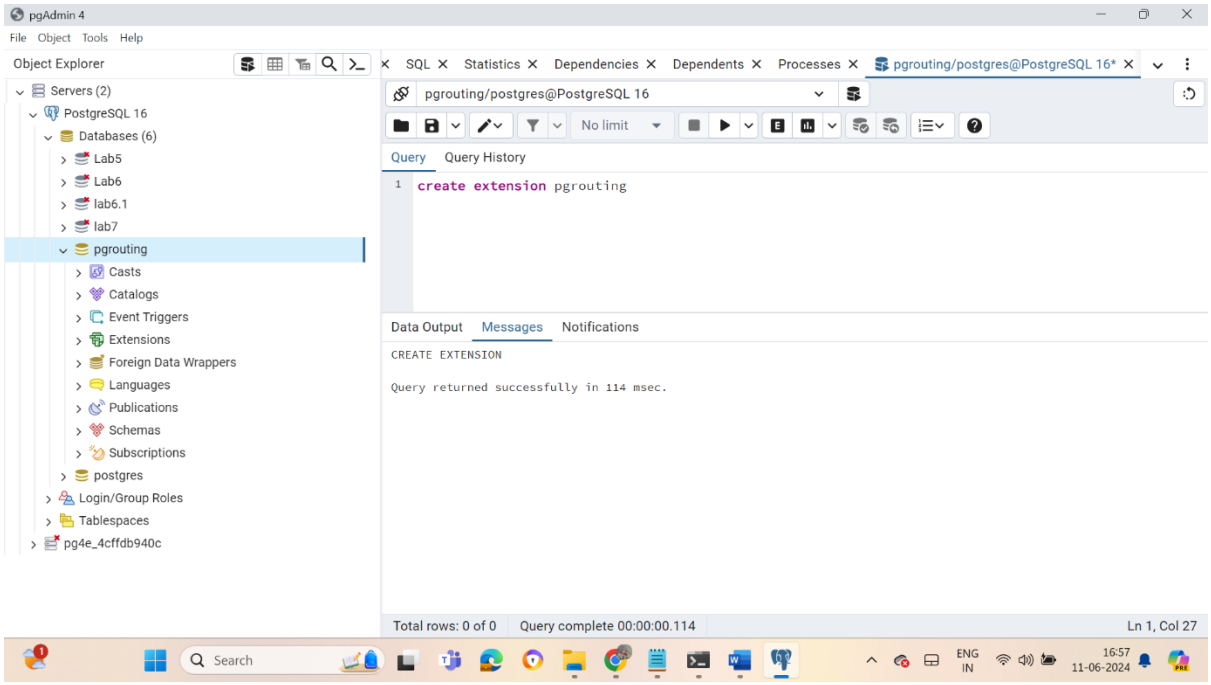
- Create Database pgrouting.



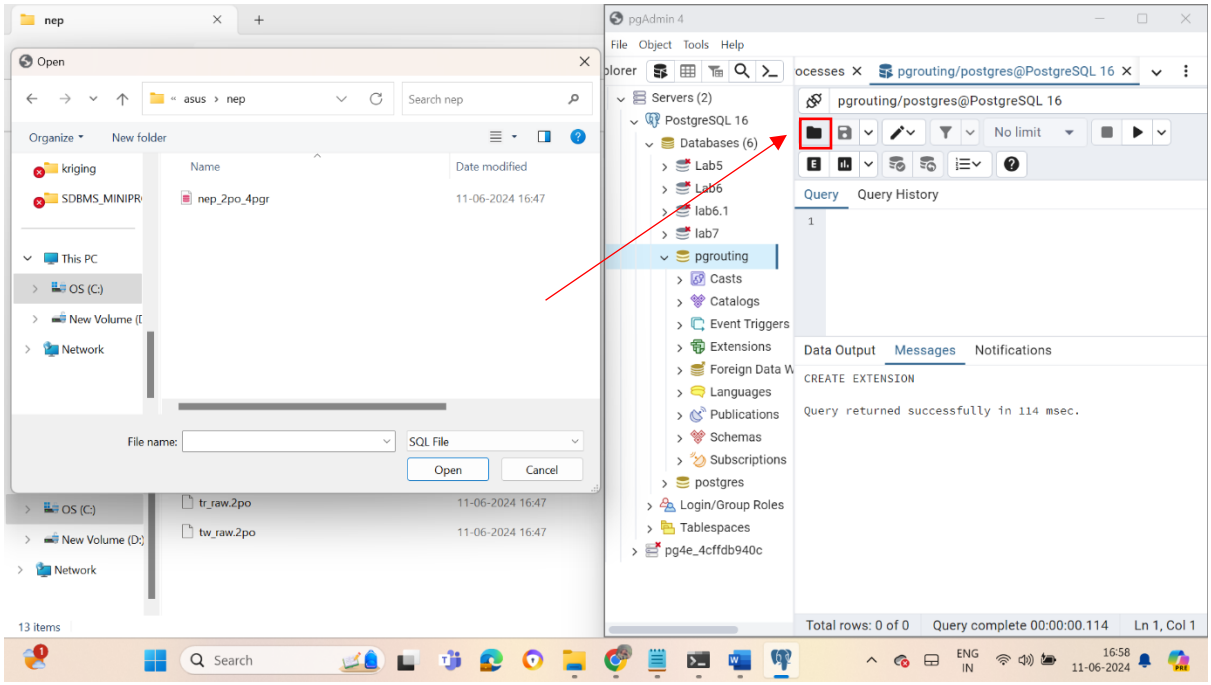
- Create Extension postgis.

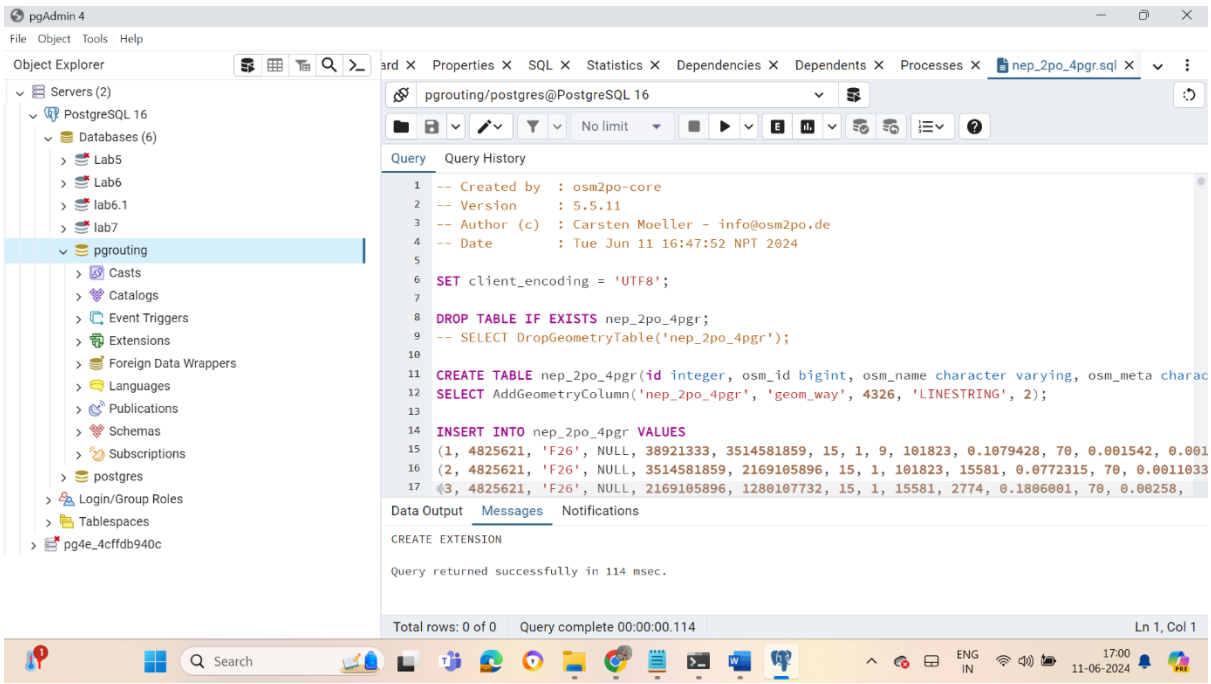


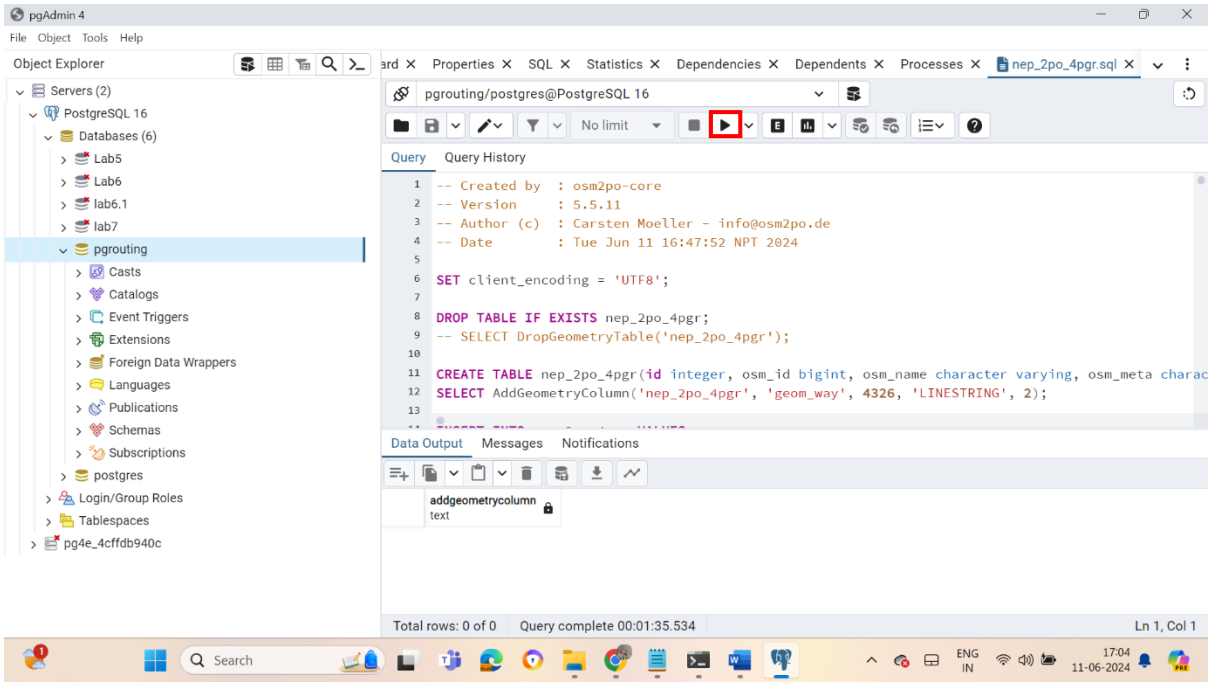
- Create Extension pgrouting.

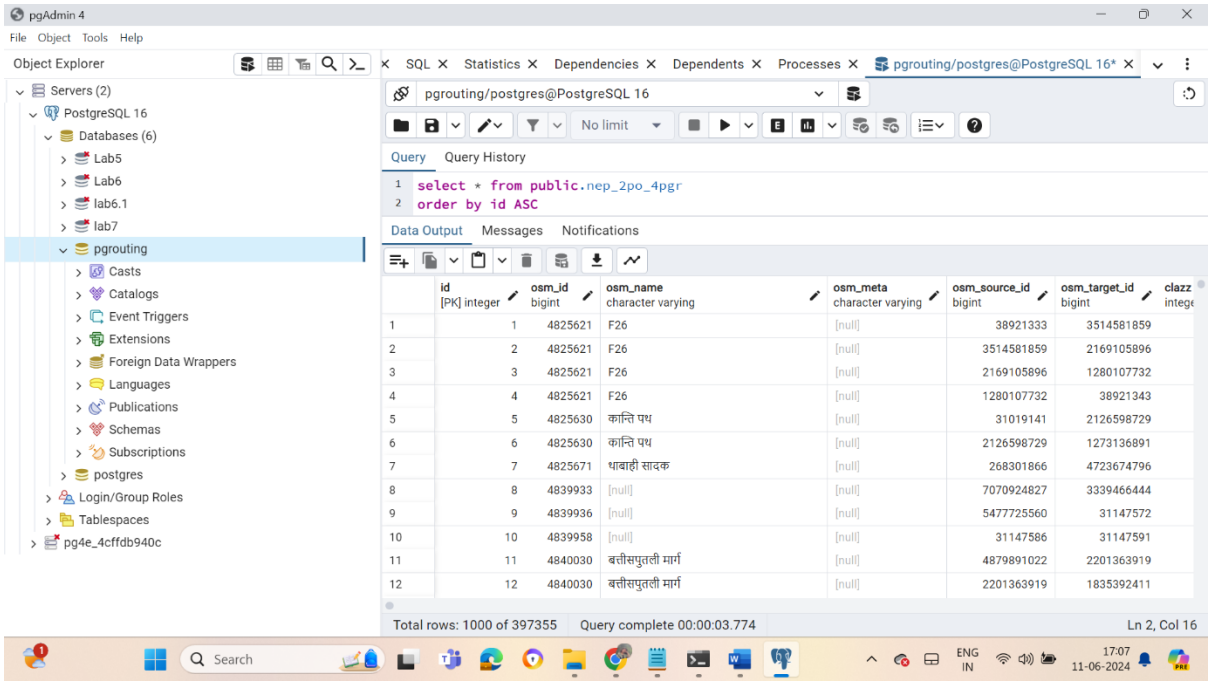


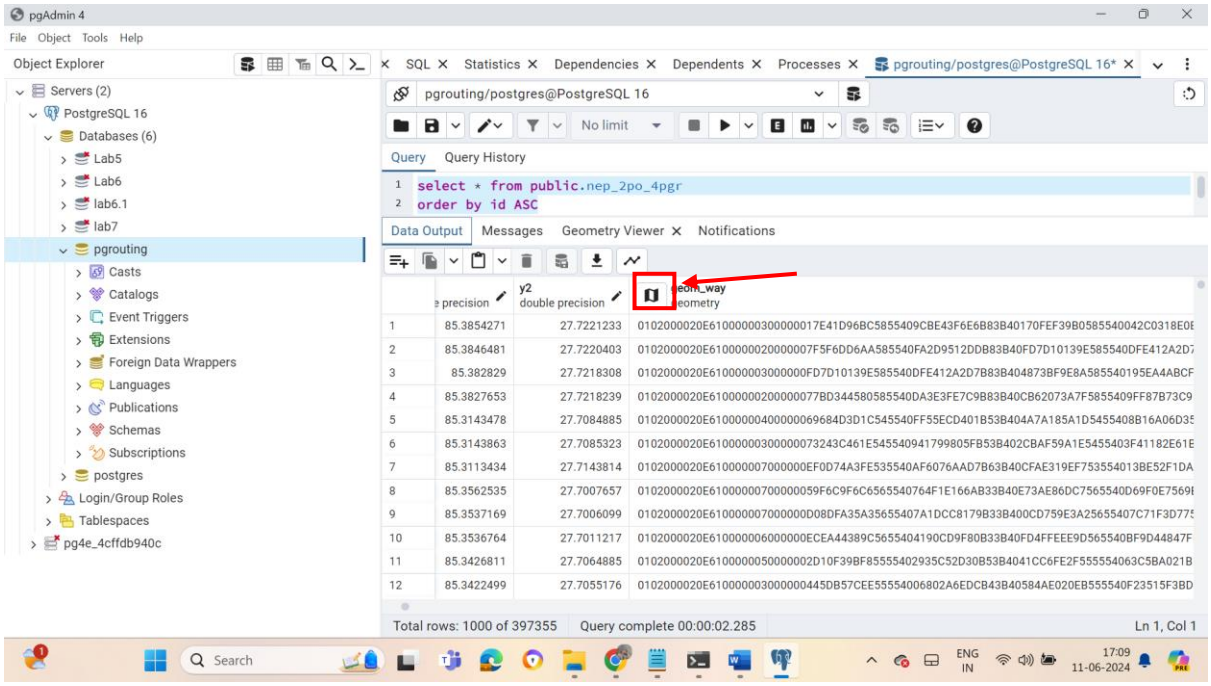
- Open .sql file converted earlier and execute.

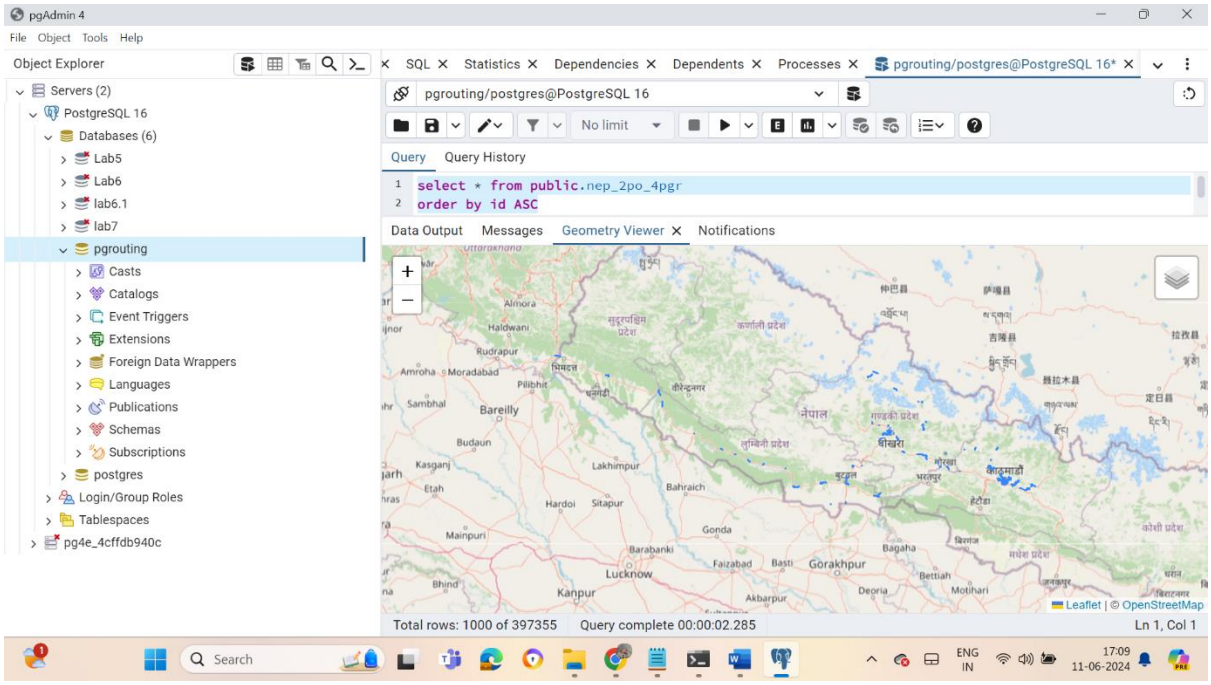





pgAdmin 4
File Object Tools Help
Object Explorer
Servers (2)
PostgreSQL 16
Databases (6)
Lab5
Lab6
lab6.1
lab7
pgrouting
Casts
Catalogs
Event Triggers
Extensions
Foreign Data Wrappers
Languages
Publications
Schemas
Subscriptions
postgres
Login/Group Roles
Tablespaces
pg4e_4cffdb940c
Properties
SQL
Statistics
Dependencies
Dependents
Processes
nep_2po_4pgr.sql
pgrouting/postgres@PostgreSQL 16
No limit
Query
Query History
-- Created by : osm2po-core
-- Version : 5.5.11
-- Author (c) : Carsten Moeller - info@osm2po.de
-- Date : Tue Jun 11 16:47:52 NPT 2024
SET client_encoding = 'UTF8';
DROP TABLE IF EXISTS nep_2po_4pgr;
-- SELECT DropGeometryTable('nep_2po_4pgr');
CREATE TABLE nep_2po_4pgr(id integer, osm_id bigint, osm_name character varying, osm_meta charac
SELECT AddGeometryColumn('nep_2po_4pgr', 'geom_way', 4326, 'LINESTRING', 2);
Data Output
Messages
Notifications
addgeometrycolumn
text
Total rows: 0 of 0
Query complete 00:01:35.534
Ln 1, Col 1
Search
ENG
IN
17:04
11-06-2024

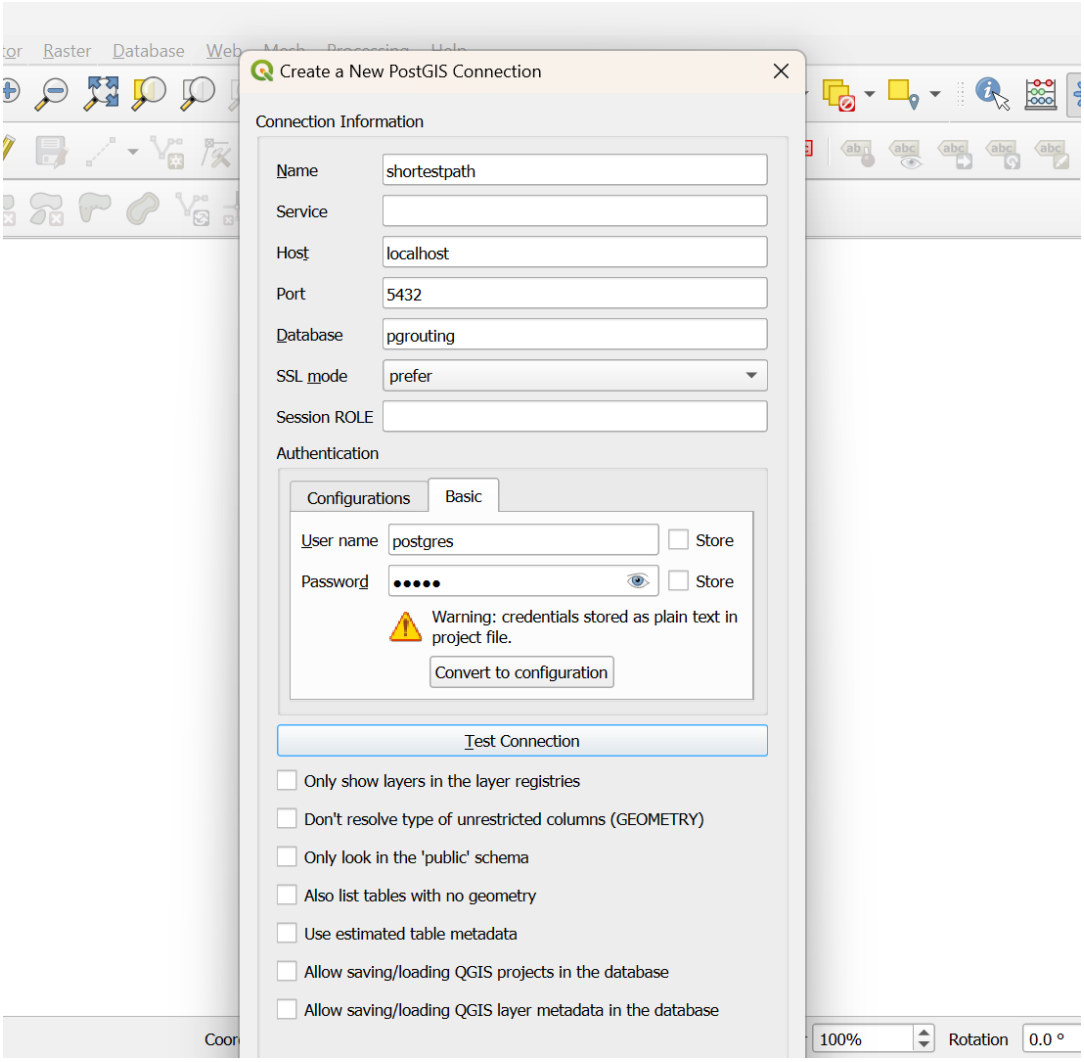

pgAdmin 4
File Object Tools Help
Object Explorer
Servers (2)
PostgreSQL 16
Databases (6)
Lab5
Lab6
lab6.1
lab7
pgrouting
Casts
Catalogs
Event Triggers
Extensions
Foreign Data Wrappers
Languages
Publications
Schemas
Subscriptions
postgres
Login/Group Roles
Tablespaces
pg4e_4cffdb940c
SQL
Statistics
Dependencies
Dependents
Processes
pgrouting/postgres@PostgreSQL 16*
pgrouting/postgres@PostgreSQL 16
No limit
Query
Query History
select * from public.nep_2po_4pgr
order by id ASC
Data Output
Messages
Notifications
id [PK] integer
osm_id bigint
osm_name character varying
osm_meta character varying
osm_source_id bigint
osm_target_id bigint
clazz integ
1 1 4825621 F26 [null] 38921333 3514581859
2 2 4825621 F26 [null] 3514581859 2169105896
3 3 4825621 F26 [null] 2169105896 1280107732
4 4 4825621 F26 [null] 1280107732 38921343
5 5 4825630 कान्ति पथ [null] 31019141 2126598729
6 6 4825630 कान्ति पथ [null] 2126598729 1273136891
7 7 4825671 थाबाही सादक [null] 268301866 4723674796
8 8 4839933 [null] [null] 7070924827 3339466444
9 9 4839936 [null] [null] 5477725560 31147572
10 10 4839958 [null] [null] 31147586 31147591
11 11 4840030 बत्तीसपुतली मार्ग [null] 4879891022 2201363919
12 12 4840030 बत्तीसपुतली मार्ग [null] 2201363919 1835392411
Total rows: 1000 of 397355
Query complete 00:00:03.774
Ln 2, Col 16
Search
ENG
IN
17:07
11-06-2024

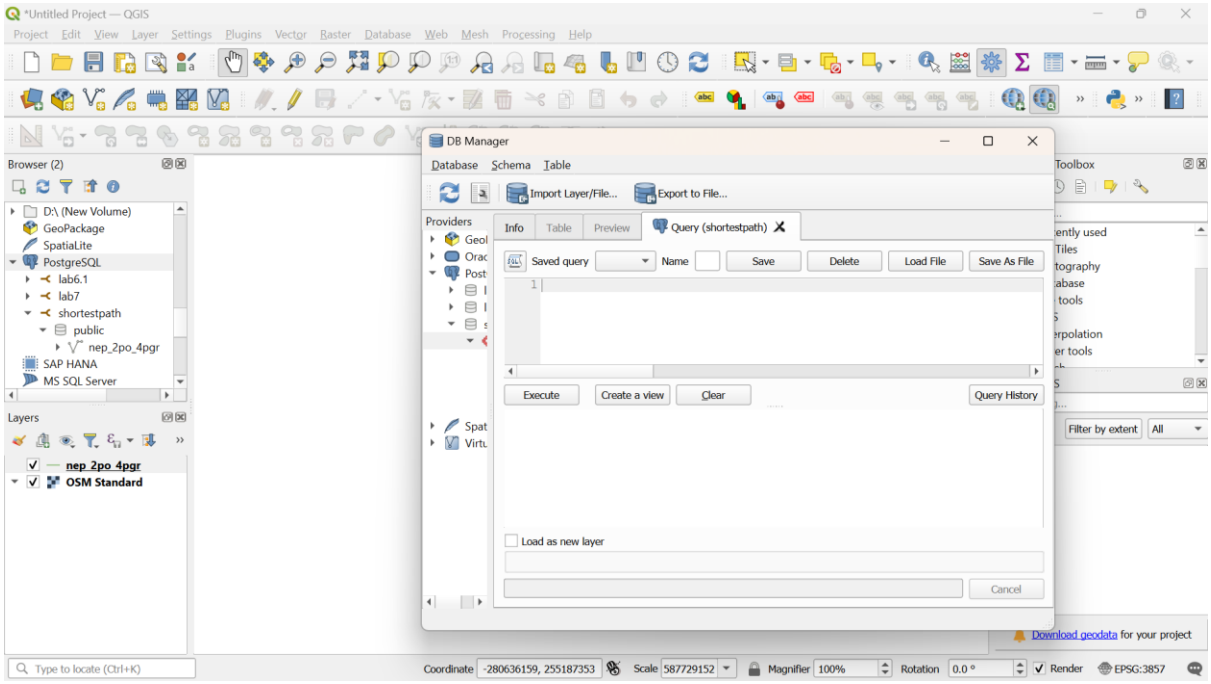

pgAdmin 4
File Object Tools Help
Object Explorer
Servers (2)
PostgreSQL 16
Databases (6)
Lab5
Lab6
lab6.1
lab7
pgrouting
Casts
Catalogs
Event Triggers
Extensions
Foreign Data Wrappers
Languages
Publications
Schemas
Subscriptions
postgres
Login/Group Roles
Tablespaces
pg4e_4cffdb940c
SQL
Statistics
Dependencies
Dependents
Processes
pgrouting/postgres@PostgreSQL 16*
pgrouting/postgres@PostgreSQL 16
No limit
Query
Query History
select * from public.nep_2po_4pgr
order by id ASC
Data Output
Messages
Geometry Viewer
Notifications
y2
double precision
Total rows: 1000 of 397355
Query complete 00:00:02.285
Ln 1, Col 1
Search
ENG
IN
17:09
11-06-2024


pgAdmin 4
File Object Tools Help
Object Explorer
Servers (2)
PostgreSQL 16
Databases (6)
Lab5
Lab6
lab6.1
lab7
pgrouting
Casts
Catalogs
Event Triggers
Extensions
Foreign Data Wrappers
Languages
Publications
Schemas
Subscriptions
postgres
Login/Group Roles
Tablespaces
pg4e_4cffdb940c
SQL
Statistics
Dependencies
Dependents
Processes
pgrouting/postgres@PostgreSQL 16*
pgrouting/postgres@PostgreSQL 16
No limit
Query
Query History
select * from public.nep_2po_4pgr
order by id ASC
Data Output
Messages
Geometry Viewer
Notifications
Almora
Haldwani
Rudrapur
Moradabad
Bareilly
Budaun
Lakhimpur
Hardoi
Sitapur
Bahraich
Gonda
Barabanki
Lucknow
Faizabad
Basti
Gorakhpur
Kanpur
Akbarpur
Deoria
Bettiah
Motihari
Leaflet | © OpenStreetMap
Total rows: 1000 of 397355
Query complete 00:00:02.285
Ln 1, Col 1
Search
ENG
IN
17:09
11-06-2024

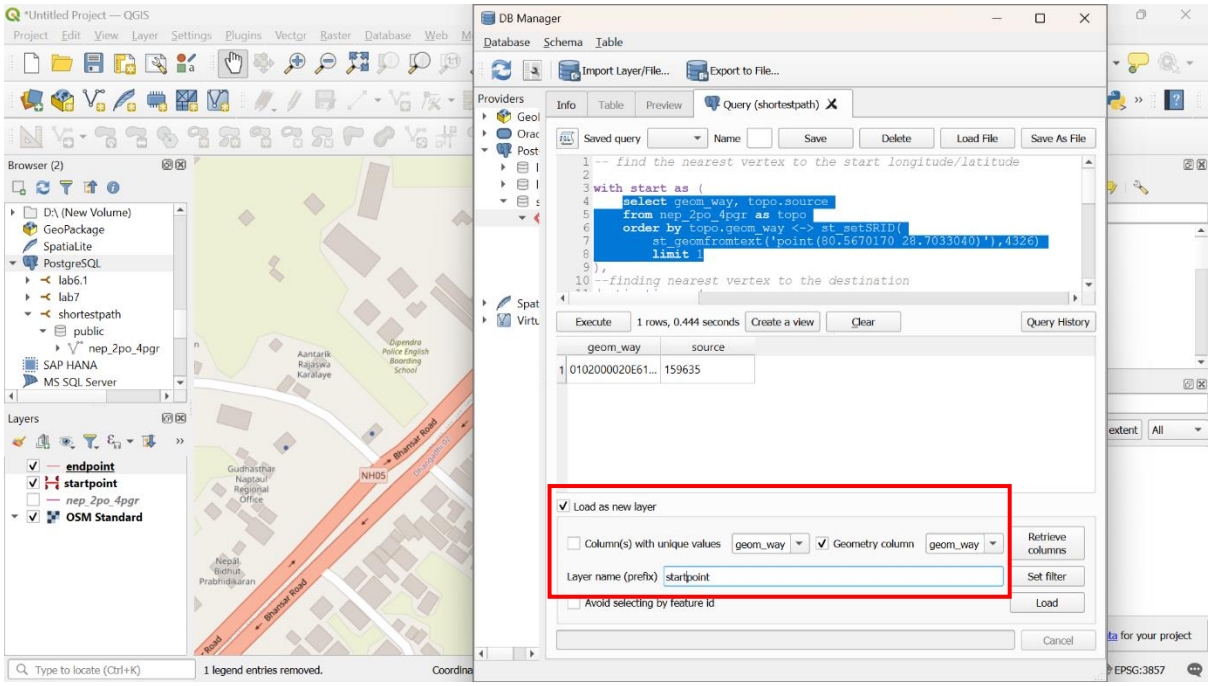
8. QGIS

- Open QGIS and Connect to database pgrouting created earlier.

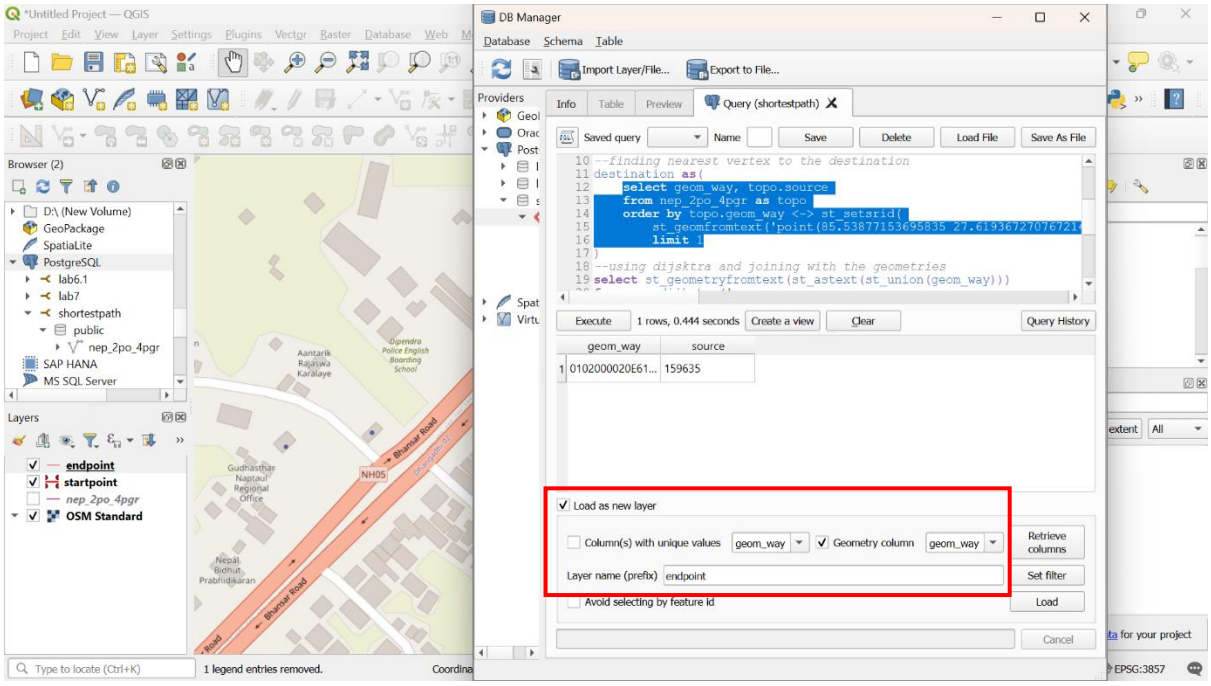


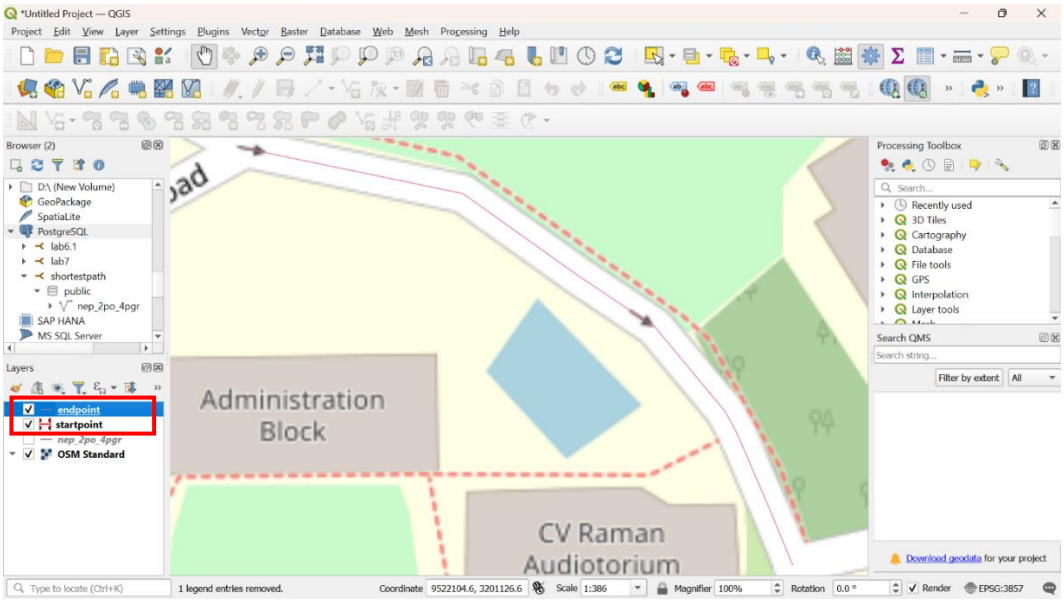


- We find the neartest vertex to start
- Origin: Dhangadhi
- While executing query, we also checked the ‘load as new layer’ option and named the layer ‘startpoint’.

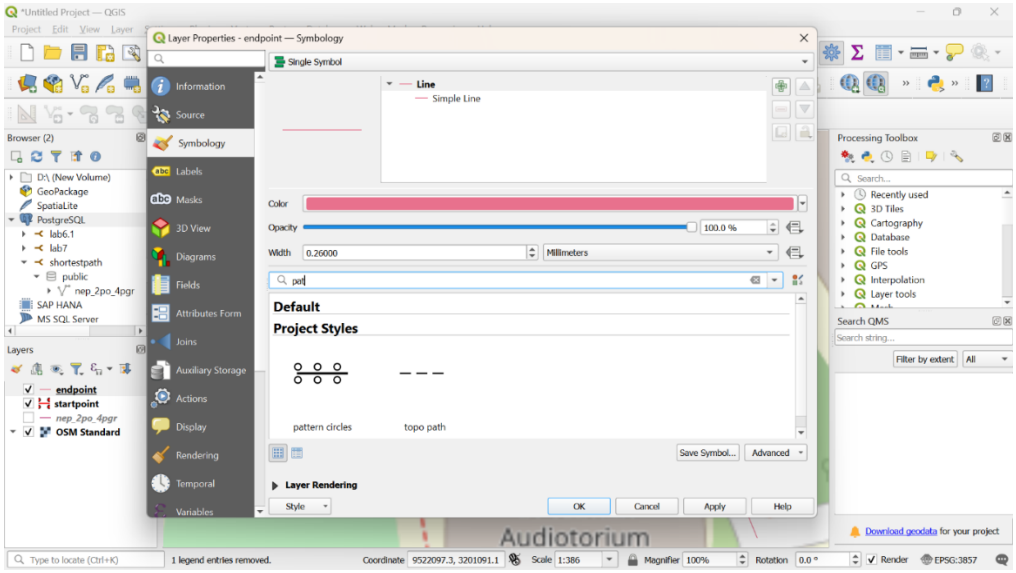


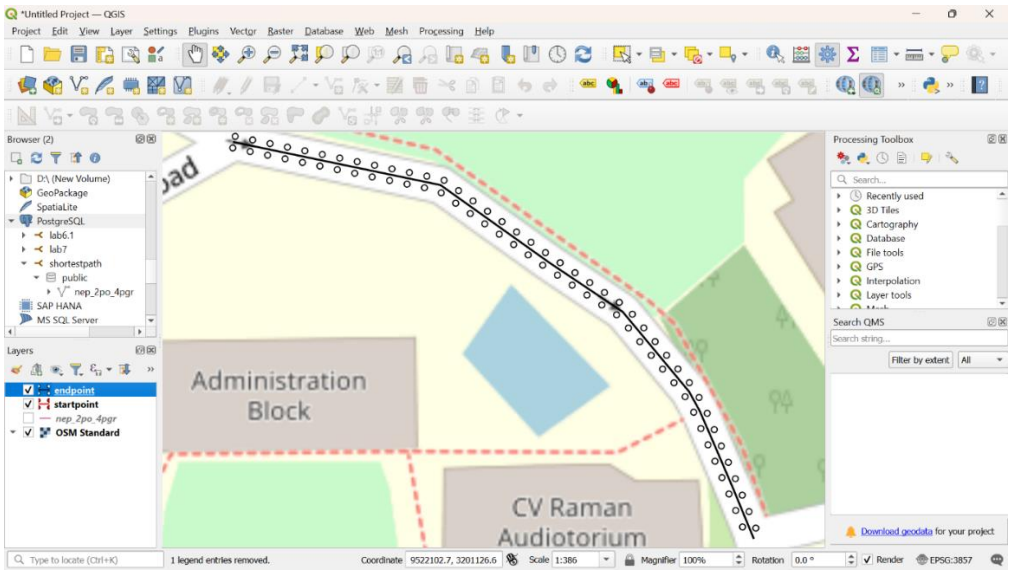
- Destination: Kathmandu University
- While executing query, we also checked the ‘load as new layer’ option and named the layer ‘endpoint’.



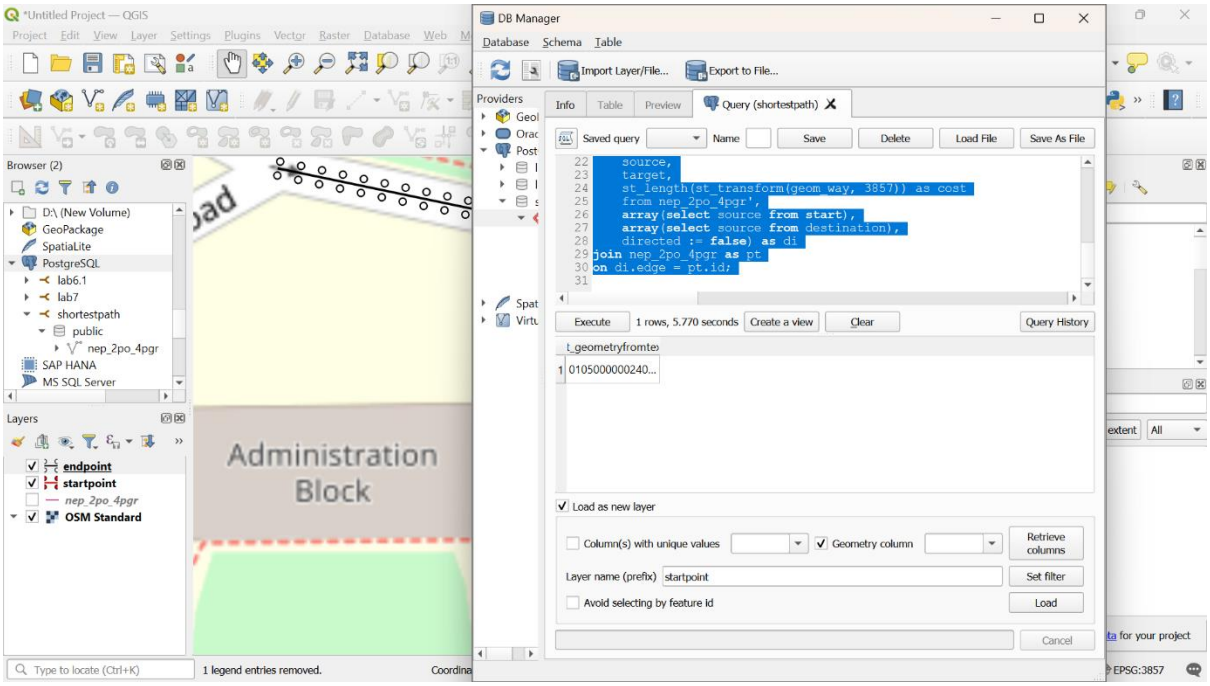


- Changing the symbols to more appropriate ones:

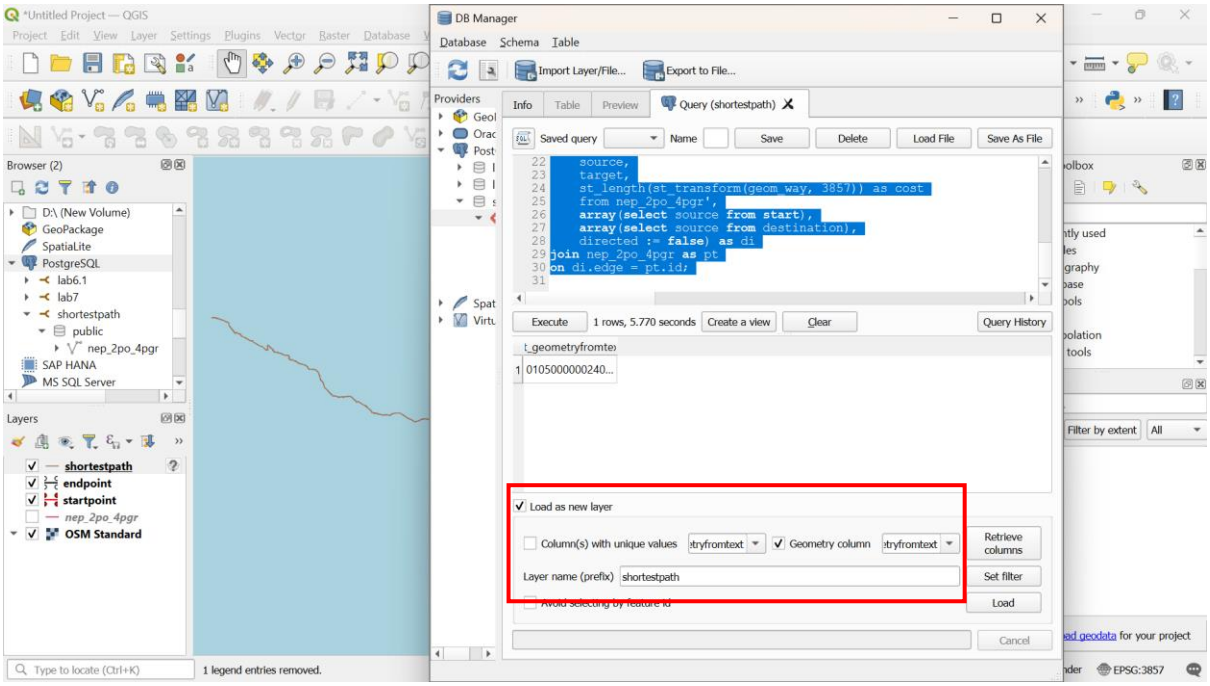




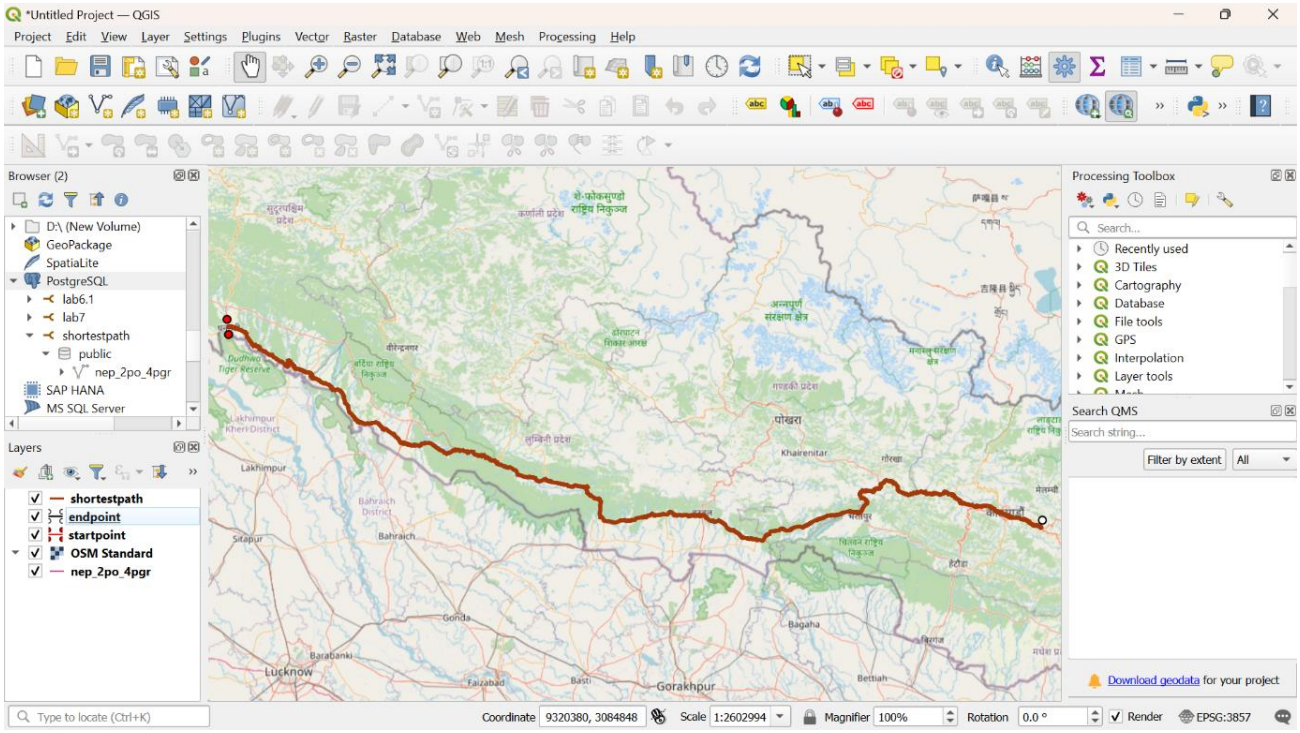
- Applying the Dijkstra's Algorithm to find the shortest path:



- Loading a new layer named as 'shortestpath'.



Final Result:



References

Alam, Md. A., & Faruq, Md. O. (2019). Finding Shortest Path for Road Network Using Dijkstra's Algorithm. *Bangladesh Journal of Multidisciplinary Scientific Research*, *1*(2). https://doi.org/10.46281/bjmsr.v1i2.366

CHOOSUMRONG, S., RAGHAVAN, V., & BOZON, N. (2012). Multi-Criteria Emergency Route Planning Based on Analytical Hierarchy Process and pgRouting. *GEOINFORMATICS*, *23*(4). https://doi.org/10.6010/geoinformatics.23.159

Nurhasanah, F. Y., Gata, W., Riana, D., Jamil, Muh., & Saputra, S. F. (2021). Shortest Path Finding Using Dijkstra's Algorithm. *PIKSEL : Penelitian Ilmu Komputer Sistem Embedded and Logic*, *9*(1). https://doi.org/10.33558/piksel.v9i1.2365

pgRouting Community. (2016). pgRouting Project — Open Source Routing Library. In *pgRouting*.

Pritee, K., Garg, R. D., & Ohri, A. (2019). Windows Implementation of PgRouting to find Shortest Path using Dijkstra's Algorithm for Varanasi Road Network. *Journal of Basic and Applied Engineering Research*, *2*(2).

Seeta, V. (2021). Finding the Shortest Path using Dijkstra's Algorithm. *International Journal of Innovative Research in Engineering & Multidisciplinary Physical Sciences*, *9*(5). https://doi.org/10.37082/ijirmps.2021.v09si05.017

Singh, P. S., Lyngdoh, R. B., Chutia, D., Saikhom, V., Kashyap, B., & Sudhakar, S. (2015). Dynamic shortest route finder using pgRouting for emergency management. *Applied Geomatics*, *7*(4). https://doi.org/10.1007/s12518-015-0161-4