Lukas Sjöstrand and Malte Kasemo reporting on Simon Rödén's Lab 3

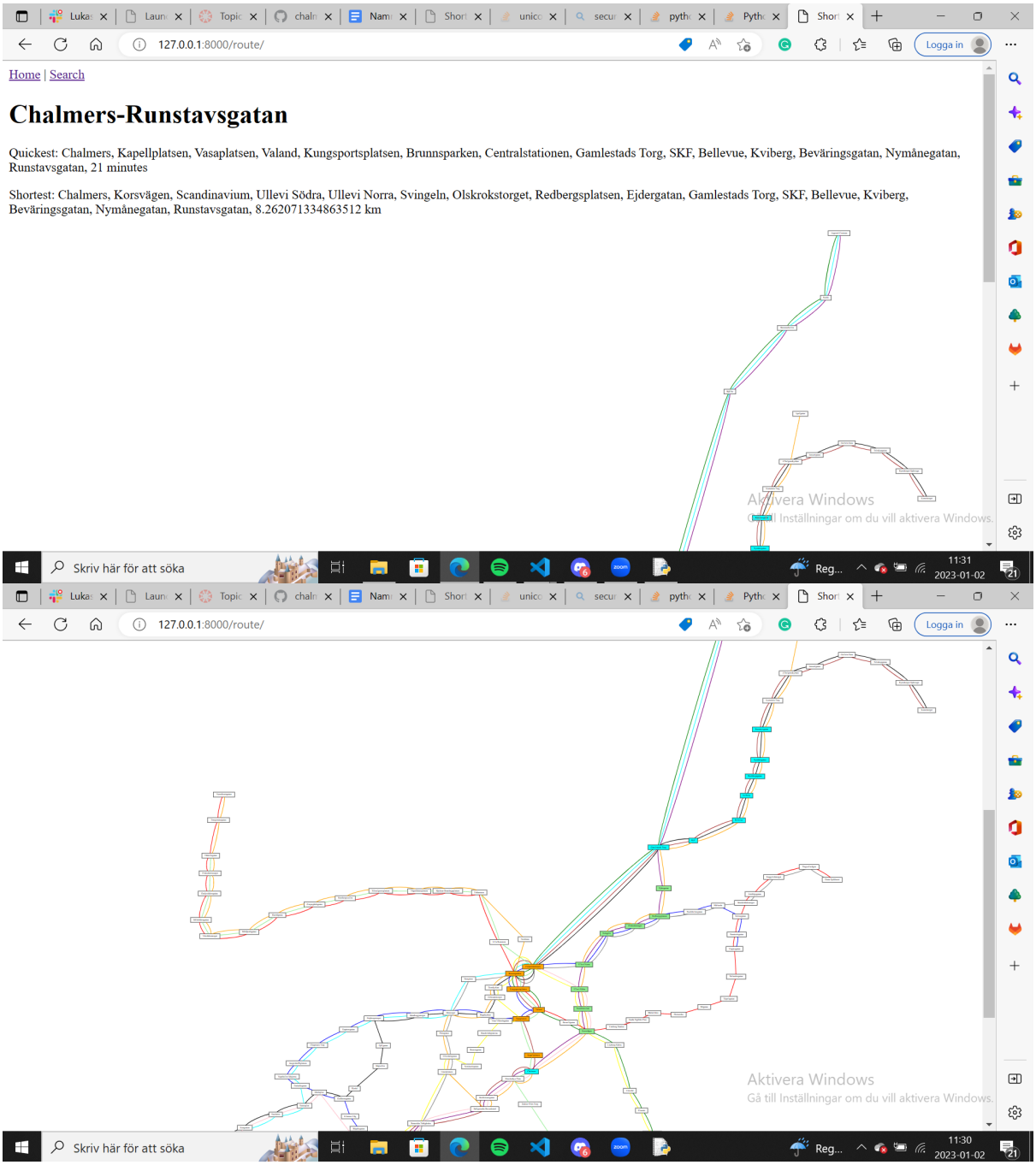
Section 1:
Q1: yes
Q2: yes
Q3: yes

Section 2:
B1: no
B2: yes

Section 3:
The code from lab 2 has been properly reused and dijkstra has been implemented and used as intended, where there is just one definition of the function. Just the cost function that have changed.

Section 4:

screenshot 1



Screenshot 2

```python
from django.conf import settings
import numpy as np

def show_shortest(dep, dest):
    network = readTramNetwork()

    quickest = dijkstra(network, dep, None)[dest]

    time = 0
    for idx, stop in enumerate(quickest[:-1]):
        if stop in network._timedict:
            if quickest[idx+1] in network._timedict[stop]:
                time += network._timedict[stop][quickest[idx+1]]
            else:
                time += network._timedict[quickest[idx+1]][stop]
        else:
            time += network._timedict[quickest[idx+1]][stop]

    def distance_between_stops(somedicts, stop1, stop2):
        data=somedicts
        phi1=lat1 = np.pi/180*float(data[stop1]["lat"])
        phi2=lat2 = np.pi/180*float(data[stop2]["lat"])
        r= 6371

        lambda1=lon1 = np.pi/180*float(data[stop1]["lon"])
        lambda2=lon2 = np.pi/180*float(data[stop2]["lon"])
        a = (np.square(np.sin((phi2-phi1)/2)) + np.cos(phi1) * np.cos(phi2) *
            np.square(np.sin((lambda2-lambda1)/2)))
        d = 2*r*np.arcsin(np.sqrt(a))
        return(d)


    shortest =dijkstra(network, dep, cost=lambda u,v: distance_between_stops(network._stopdict,u, v))[dest]

    distance = 0
    for idx, i in enumerate(shortest[:-1]):
        distance += distance_between_stops(network._stopdict, i, shortest[idx+1])

    timepath = 'Quickest: ' + ', '.join(quickest) + ', '+ str(time) + ' minutes'
    geopath = 'Shortest: ' + ', '.join(shortest) + ', ' + str(distance)+' km'

    def colors(v):
        if v in shortest and v in quickest:
            return 'cyan'
        elif v in shortest:
            return 'lightgreen'
        elif v in quickest:
            return 'orange'
        else:
            return 'white'
```