

Instalirati biblioteku geopy. **pip3 install geopy**

Na osnovu unetih parametara odrediti lokaciju

```
import os
from geopy.geocoders import Nominatim

def set_proxy():
    proxy_addr = 'http://{}:{}'.format(
        address='proxy.uns.ac.rs', port=int('8080'))
    os.environ['http_proxy'] = proxy_addr
    os.environ['https_proxy'] = proxy_addr

def unset_proxy():
    os.environ.pop('http_proxy')
    os.environ.pop('https_proxy')

#set_proxy()
geolocator = Nominatim(user_agent="test")

# Odredjivanje pozicije na osnovu stringa

#location = geolocator.geocode('Dr Ilije Djuricica 1 Novi Sad',
#timeout=10)
#print(location.latitude, location.longitude)

# Na osnovu geografske duzine i sirine
#latitude = 45.249540754858295
#longitude = 19.87405135159598
#location = geolocator.geocode('%s, %s' % (latitude,
#longitude), timeout = 10)
#print(location.address)

#unset_proxy()
```

Instalirati biblioteku za prikaz mapa

pip3 install folium

```
import io
import sys
import folium
from PySide6 import QtWidgets, QtWebEngineWidgets
#from geopy.geocoders import Nominatim

#geolocator = Nominatim(user_agent="test")
#location = geolocator.geocode('Dr Ilije Djuricica 1 Novi Sad',
#timeout=10)
#long = location.longitude
#lat = location.latitude

if __name__ == "__main__":
    app = QtWidgets.QApplication(sys.argv)
```

```

m = folium.Map(location=[14, 14])
#m = folium.Map(location=[lat,long], zoom_start = 20)

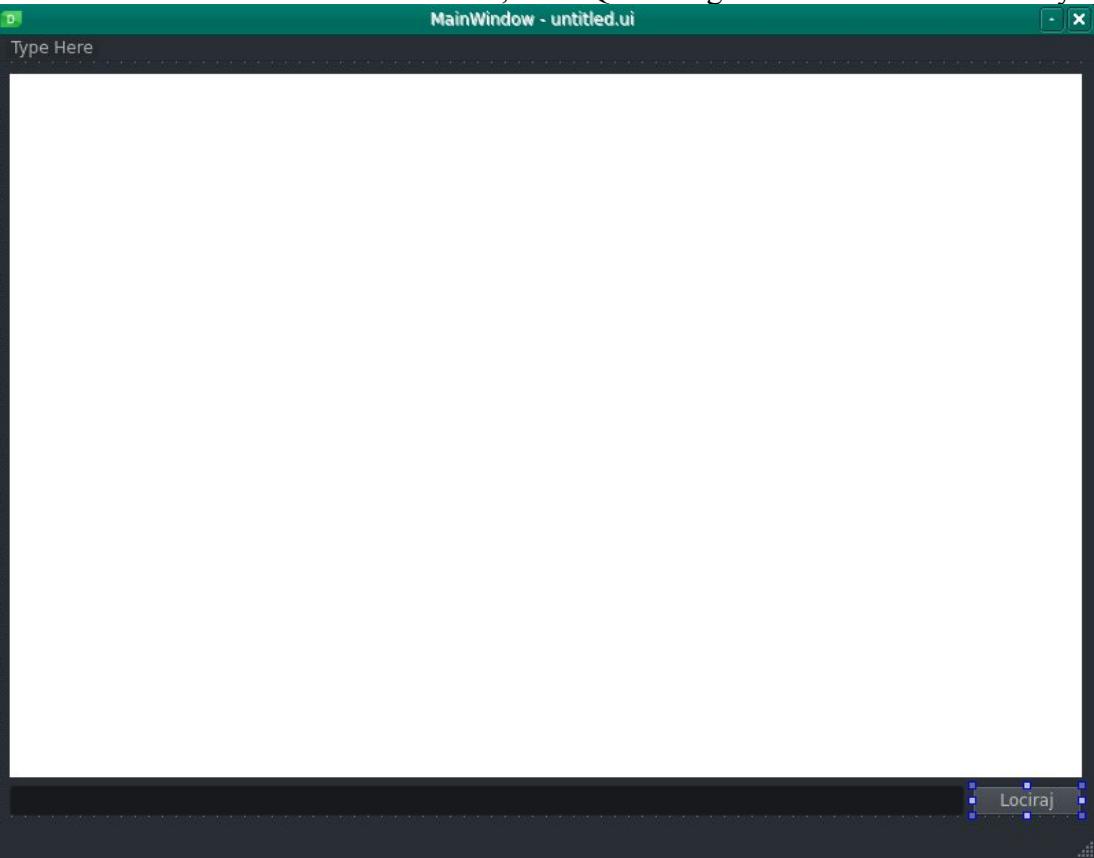
data = io.BytesIO()
m.save(data, close_file=False)

w = QtWebEngineWidgets.QWebEngineView()
w.setHtml(data.getvalue().decode())
w.resize(640, 480)
w.show()

sys.exit(app.exec())

```

U Main window dodati taster i text edit, kao i QWebEngineView. Postaviti Grid layout.



Nakon čuvanja .ui fajla generisati py fajl. U QWebEngineView ubaciti html mapu generisanu folium-om.

```

from PySide6.QtWidgets import QApplication, QMainWindow
from p2uic import Ui_MainWindow
from geopy.geocoders import Nominatim
import folium
import io

class MainWindow(QMainWindow):
    def __init__(self):
        super(MainWindow, self).__init__()
        self.ui = Ui_MainWindow()
        self.ui.setupUi(self)

```

```

        self.ui.pushButton.clicked.connect(self.locate)
        self.geolocator = Nominatim(user_agent="test")

    def locate(self):
        str = self.ui.lineEdit.text()
        location = self.geolocator.geocode(str, timeout=10)
        lat = location.latitude
        long = location.longitude
        self.map = folium.Map(location = [lat,long], zoom_start = 20)
        #folium.Marker(location = [lat,long]).add_to(self.map)
        data = io.BytesIO()
        self.map.save(data, close_file=False)

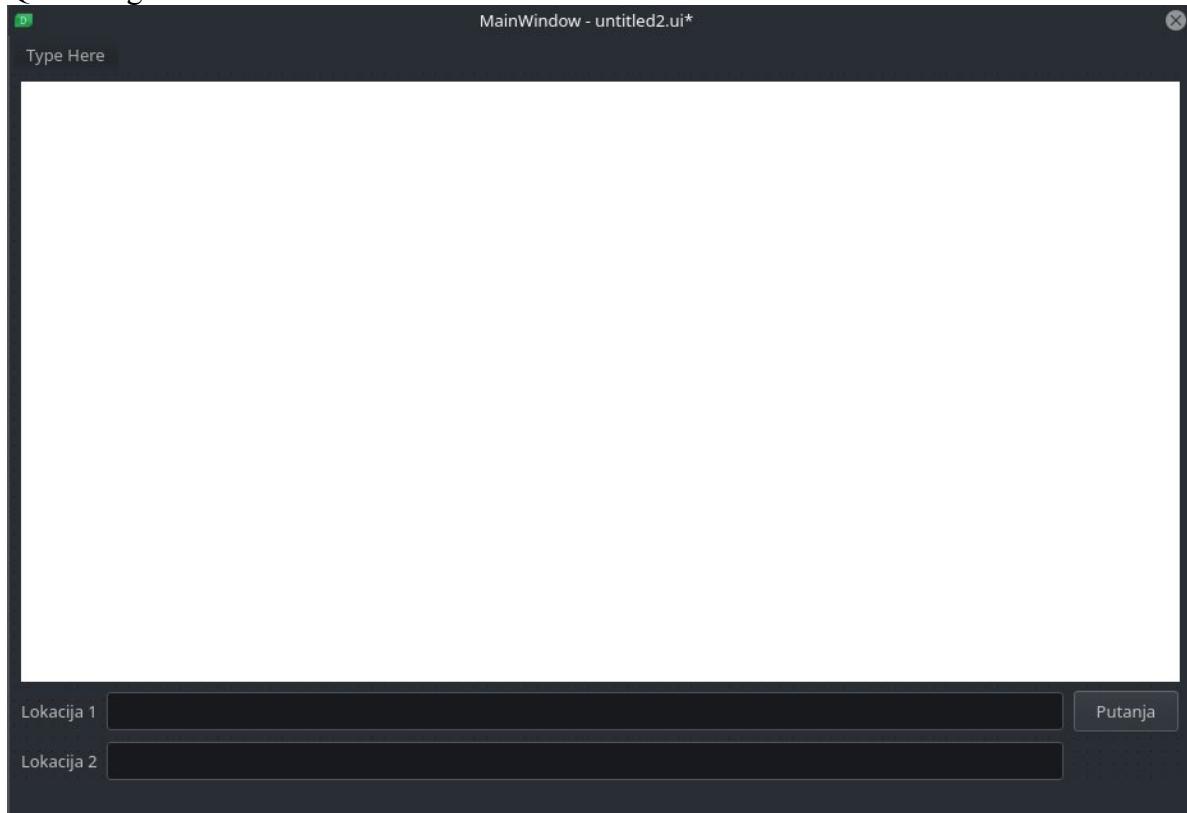
        self.ui.webEngineView.setHtml(data.getvalue().decode())

app = QApplication()
w = MainWindow()
w.show()

app.exec()

```

Dodati dva lineEdit-a, dve labele i taster sa nazivom putanja. Za prikaz mape koristi se QWebEngineView.



```

from PySide6.QtWidgets import QApplication, QMainWindow
from p4uic import Ui_MainWindow
from geopy.geocoders import Nominatim
import folium

```

```

import io

class MainWindow(QMainWindow):
    def __init__(self):
        super(MainWindow, self).__init__()
        self.ui = Ui_MainWindow()
        self.ui.setupUi(self)

        self.ui.pushButton.clicked.connect(self.locate)
        self.geolocator = Nominatim(user_agent="test")

    def locate(self):
        str1 = self.ui.lineEdit.text()
        str2 = self.ui.lineEdit_2.text()
        location1 = self.geolocator.geocode(str1, timeout=10)
        location2 = self.geolocator.geocode(str2, timeout=10)

        self.map=folium.Map([location1.latitude,location1.longitude],
                           zoom_start = 15)

        folium.Marker(location =
                      [location1.latitude,location1.longitude]).add_to(self.map)
        folium.Marker(location =
                      [location2.latitude,location2.longitude]).add_to(self.map)

        folium.PolyLine([(location1.latitude,location1.longitude),(location2.latitude,location2.longitude)], color='red', weight=7,
                        opacity=0.8).add_to(self.map)

        data = io.BytesIO()
        self.map.save(data, close_file=False)

        self.ui.webEngineView.setHtml(data.getvalue().decode())

app = QApplication()
w = MainWindow()
w.show()

app.exec()

```

Instalirati biblioteku za traženje najbliže putanje
pip3 install osmnx

```

from PySide6.QtWidgets import QApplication, QMainWindow
from p4uic import Ui_MainWindow
from geopy.geocoders import Nominatim
import folium
import io
import osmnx as ox
import networkx as nx

```

```

class MainWindow(QMainWindow):
    def __init__(self):
        super(MainWindow, self).__init__()
        self.ui = Ui_MainWindow()
        self.ui.setupUi(self)

        self.ui.pushButton.clicked.connect(self.locate)
        self.geolocator = Nominatim(user_agent="test")

    def locate(self):
        str1 = self.ui.lineEdit.text()
        str2 = self.ui.lineEdit_2.text()
        location1 = self.geolocator.geocode(str1, timeout=10)
        location2 = self.geolocator.geocode(str2, timeout=10)

        G = ox.graph_from_point((location1.latitude,
                                location1.longitude), dist=10000, network_type='drive')

        orig_node = ox.nearest_nodes(G, location1.longitude,
                                    location1.latitude)
        dest_node = ox.nearest_nodes(G, location2.longitude,
                                    location2.latitude)

        route = nx.shortest_path(G, source=orig_node,
                                target=dest_node, weight='length')
        ox.config(log_console=True, use_cache=True)

        route_map = ox.plot_route_folium(G, route)

        #m = folium.Map([location1.latitude,location1.longitude],
        #               zoom_start = 15)
        #fg = folium.FeatureGroup(name='legend name', show=True)

        #ox.plot_route_folium(G, route, route_color='green',
        #                      route_map=fg, tooltip='some tooltip')
        #m.add_child(fg)
        #data = io.BytesIO()
        #m.save(data, close_file=False)

        data = io.BytesIO()
        route_map.save(data, close_file=False)
        self.ui.webEngineView.setHtml(data.getvalue().decode())

app = QApplication()
w = MainWindow()
w.show()
app.exec()

```

Dodatni primeri korišćenja folium-a:

https://nbviewer.org/github/python-visualization/folium_contrib/tree/master/notebooks/