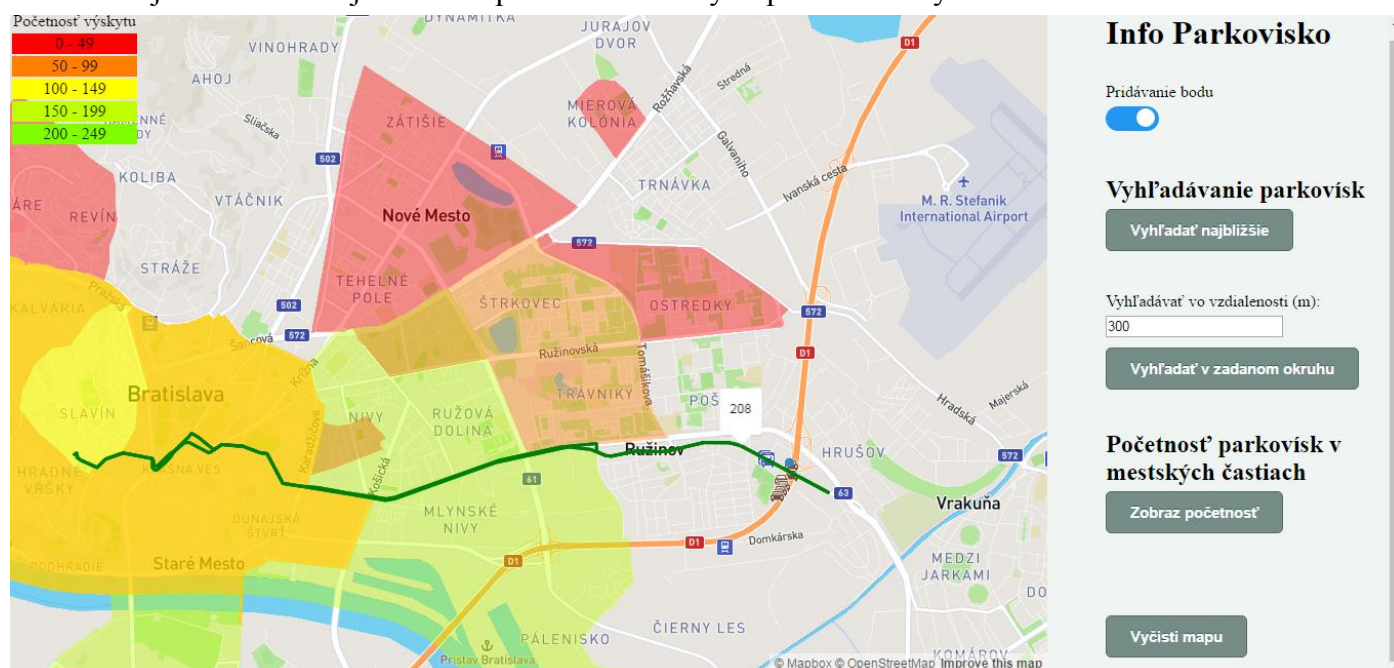


Prehľad

Webová aplikácia umožňuje používateľovi nasledovné operácie:

- Vybrať bod a od neho vyhľadať:
 - Najbližšie parkovisko.
 - Všetky parkoviská v zadanej vzdialenosti.
- Po kliknutí na nájdené parkovisko používateľ zadá vzdialenosť vyhľadávania a zobrazia sa mu všetky linky mestskej hromadnej dopravy, ktoré majú zastávku v zadanej vzdialenosti.
- Zobrazenie počtosti parkovísk pre jednotlivé mestské časti.

Na nasledujúcom obrázku je ukážka aplikácie so všetkými používateľskými scenármi.



Aplikácia sa skladá z dvoch častí – z frontendu, pri ktorom som použil HTML, CSS a JavaScript, a v ktorom využívam knižnicu [Mapbox GL JS](#), ktorá je určená na prácu s mapou. Druhá časť – backend sa skladá z PHP skriptov, ktoré získavajú dáta z databázy PostGIS. Dáta sa v skriptoch následne upravujú a sú poskytované cez http rozhranie frontendu.

Frontend

Frontend aplikácia je postavená na jednej statickej stránke HTML (*index.html*), ktorá sa skladá z dvoch častí: mapy a bočného panelu s možnosťami ovládania stránky.

Ďalšou časťou frontend aplikácie je JavaScript (*app/js/app.js*), v ktorom sa vykonáva celá logika na frontende. V tomto súbore sú definované jednotlivé udalosti, ktoré môžu nastať v aplikácii a to: kliknutie na tlačidlo alebo mapu, ukázanie na zobrazené veci na mape. Ďalej sú v tomto súbore riešené vyvolané akcie, čiže dopytovanie sa po dátach na backend a následne zobrazovanie dát na mape.

Poslednou časťou frontendu je CSS (*app/css/style.css*), pomocou ktorého som upravoval vzhľad aplikácie.

Backend

Backendová časť aplikácia je naprogramovaná v PHP (viacero skriptov v app/API/). Jedná sa o jednoduché skripty, ktoré sú volané z frontendu. Všetky skripty riešia len pripojenie do databázy, výber dát, spracovanie dát (spracovanie do GeoJSON formátu) a ich odoslanie na frontend.

Dáta

Všetky dáta pochádzajú z Open Street Maps. Databázu som si vytvoril pomocou aplikácie [Docker ToolBox](#). Do nej som si následne stiahol dáta Bratislavy, ktoré som získal zo [stránky predmetu](#). V databáze získavam GeoJSON dáta pomocou funkcie `st_asgeojson` a zároveň aj iné doplňujúce dáta ako napríklad názvy parkovísk, čísla liniek alebo názvy zastávok. Niektoré dáta preto ešte vyžadujú ďalšie spracovanie (napríklad umiestnenie názvu do *properties.name*, kde musí byť kvôli ďalšiemu zobrazovaniu na frontende), ktoré prebieha v PHP skriptoch.

Api rozhranie

V rámci rozhrania sú implementované štyri volania:

- Získanie najbližšieho parkoviska od bodu – POST <http://127.0.0.1:80/app/API/getNearest.php> data: {lat : lat, lng : lng}
- Všetky parkoviská vo zvolenej vzdialenosti od bodu vykliknutého na mape – POST <http://127.0.0.1:80/app/API/getAllNearest.php> data: {lat : lat, lng : lng, dist : dist}
- Zobrazenie počtu parkovísk pre jednotlivé mestské časti – POST <http://127.0.0.1:80/app/API/getRatio.php>
- Vyhľadanie liniek mestskej hromadnej dopravy, ktoré obsluhujú zastávky v okolí – POST <http://127.0.0.1:80/app/API/getStopAndLines.php> {lat : lat, lng : lng, dist : dist}

Odpoveď

Odpovede backendu sa líšia v závislosti od požiadavky. V prvých dvoch prípadoch je odpoveď nasledovná (v prípade viacerých parkovísk – viacej objektov):

```
"0": {
  "geometry": {
    "type": "Point",
    "coordinates": {
      "0": 17.0720708,
      "1": 48.1475957
    }
  },
  "properties": {
    "name": "No Name"
  }
},
```

V prípade početnosti parkovísk, sú odosielané zoradené polygóny obohatené o *count*, ktorý nesie informáciu o počte parkovísk:

```

{
  "count": 1,
  "geometry": {
    "type": "Polygon",
    "coordinates": [
      [
        [
          17.1554751,
          48.1758019
        ],
        .
        .
        .
        [
          17.1554751,
          48.1758019
        ]
      ]
    ]
  },
  "properties": {
    "name": "Mierová kolónia"
  }
},

```

V poslednom prípade sú ako dva objekty odosielané zastávky (typ Point) a linky (typ LineString):

```

[
  [
    {
      "type": "Feature",
      "properties": {
        "icon": "bus",
        "name": "Botanická záhrada"
      },
      "geometry": {
        "type": "Point",
        "coordinates": [
          17.0717803,
          48.1481357
        ]
      }
    },
    .
    .
    .
  ],
  [
    {
      "type": "Feature",
      "geometry": {
        "type": "LineString",
        "coordinates": [
          [
            16.9757339,
            48.1932594
          ],
          .
          .
          .
          [
            17.0623745,
            48.1508407
          ]
        ]
      },
      "properties": {
        "name": 141
      }
    }
  ]
]

```