

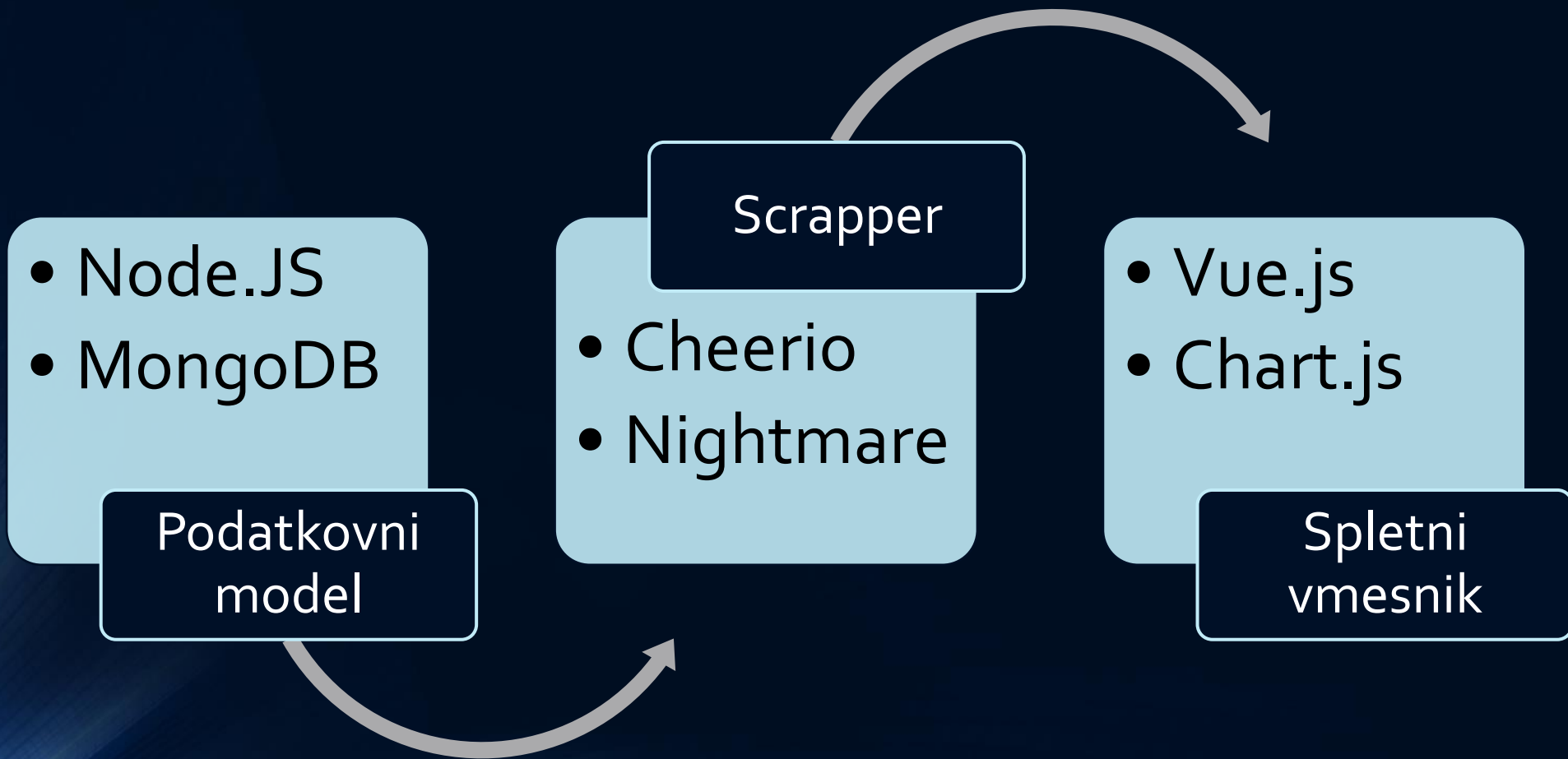
Analiza Masovnih Podatkov

KLEMEN NAPAST
GORAN KIROV
MATIC LAKOTA

Prometna infrastruktura

- Podatkovni model
- Scraper
- API
- Spletni vmesnik

Smo uporabili:



Prometna infrastruktura

ŠTEVCI PROMETA

Podatkovni model

ZBIRKE DOKUMENTOV

bin	Izgeneriran server za delo
controllers	Popravljanje tezav in dodajanje funkcionalnosti
models	Dodan rest o stanje prometa
public/stylesheets	Popravljanje tezav in dodajanje funkcionalnosti
routes	Popravljanje tezave
utils	Dodan converter za pisanje podatkov v json formatu
views	Popravljanje tezav in dodajanje funkcionalnosti
app.js	Dodan converter za pisanje podatkov v json formatu
package-lock.json	Popravljanje tezave
package.json	Popravljanje tezave
scraper.js	Dodan converter za pisanje podatkov v json formatu

VZOREC DOKUMENTA

```
models > JS TrafficMetersModel.js > ...
1  var mongoose = require('mongoose');
2  var Schema    = mongoose.Schema;
3
4  var TrafficMetersSchema = new Schema({
5    'Location' : String,
6    'Road'     : String,
7    'Direction' : String,
8    'NumberOfVehicles' : Number,
9    'Speed'    : Number,
10   'Condition' : String
11 });
12
13 module.exports = mongoose.model('TrafficMeters', TrafficMetersSchema);
14
```

Scraper

ŠTEVCI PROMETA

```
11 runScraper: async () => {
12   var data = [];
13
14   if(!loaded){
15     await nightmare
16       .goto(url)
17       .wait('body')
18       .evaluate(() => document.querySelector('body').innerHTML)
19       .then(response => {
20         data = parseData(response)
21         loaded = true;
22       }).catch(err => {
23         console.log(err);
24         data = null;
25         nightmare.refresh();
26         return err;
27       });
28   } else {
29     await nightmare
30       .evaluate(() => document.querySelector('body').innerHTML)
31       .then(response => {
32         data = parseData(response)
33       }).catch(err => {
34         console.log(err);
35         data = null;
36         nightmare.refresh();
37         return err;
38       });
39   }
40
41   return data;
42 }
```

CESTNE KAMERE

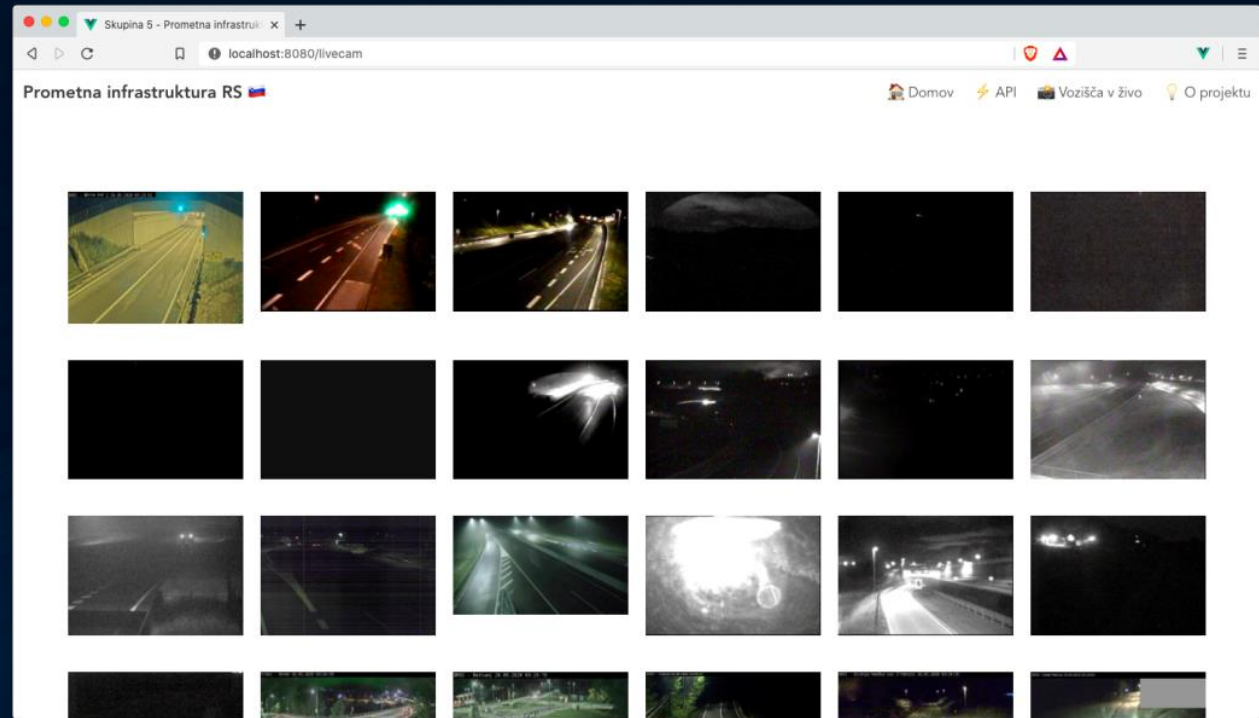
```
9 getCameras: async (req, res) =>{
10   var file = fs.readFileSync("kamere.json");
11
12   return JSON.parse(file)
13 },
14 scrapeToFile: async (req, res) =>{
15   if(req.params.password != "skupina5")
16     return "napačno geslo"
17   var imgs = []
18
19   await nightmare
20     .goto(url)
21     .inject('js', 'jquery-3.5.1.slim.min.js')
22     .wait('body')
23     .evaluate(function() {
24       var index = 0;
25       var imgs = []
26
27       $('.donotremove').each(function(){
28         var parent = $(this).parent();
29         var name = $(this).html();
30         var call = $(this).attr("href")
31
32         call = call.substring(0, call.length - 1)
33         call = call.replace("javascript:KamereCreateImg(", "")
34         call = call.replace(/(\s)/g, "")
35
36         var split = call.split(",");
37
38         if(split.length == 2){
39           KamereCreateImg(split[0], split[1])
40
41           var url = parent.find("img").first().attr('src')
42           url = url.replace(/(\s&dt=).+/, "")
43
44           imgs.push({name, url})
45         }
46         index++;
47       });
48
49       return imgs;
50     })
51 }
```

API

- Prikaz seznama akcij
- Demonstracija shranjevanje podatkov iz scrapperja v API - Postman
- Demonstracija shranjevanje podatkov iz senzorjev v API - Postman

Spletni Vmesnik

Demonstracija!



Spletni Vmesnik

Prikaz kamere!

