Работа во облак

- вежби -



Firebase

Firebase од Google нуди два типа решенија за решавање на проблемот со чување достапност на податоците. База на податоци во реално време и Cloud Firestore се бази податоци базирани на облак на кои поддржуваат синхронизирање на податоци во реално време. База на податоци во реално време е ефикасно решение и е наменета за мобилни апликации бараат кои синхронизација во реално време. Cloud Firestore е исто така наменета за мобилни апликации и е наследник на база на податоци во реално време со нов и поинтуитивен модел на податоци. Cloud Firestore побогата, побрза и поскалабилна од базата на податоци во реално време.





Cloud Firestore

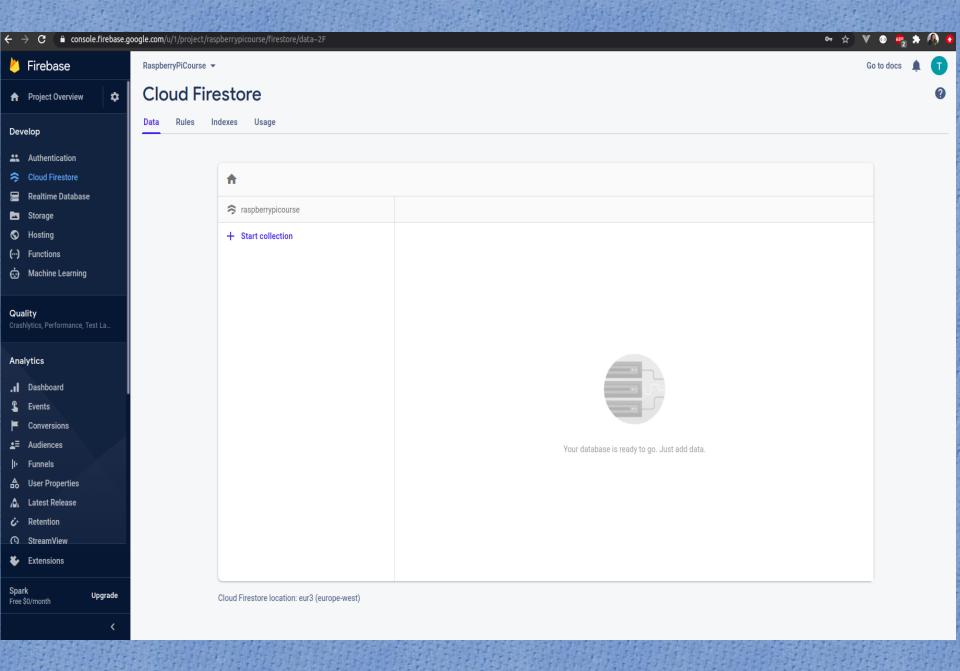
Cloud Firestore претставува база со податоци базирана на документи, која нуди подобро структурирање и сите податоци се сочувани во документи кои може да вклучуваат податоци од различен тип (текст, цел број, децимален број или пак објект). Сите документи се групирани во колекции.

Едноставни CRUD операции - Cloud Firestore овозможува додавање, бришење, модифицирање и читање на податоци од базата. Таа овозможува синхронизација на податоци помеѓу различни апликации и сервиси преку посебни интерфејси. За читање на податоци од базата постојат две опции и тоа читање на цела колекција и читање на документ.

Податоците се чуваат на различни локации и тоа доведува до глобална скалабилност и достапност

Синхронизација помеѓу различни апликации и сервиси во реално време (мобилни и веб апликации).





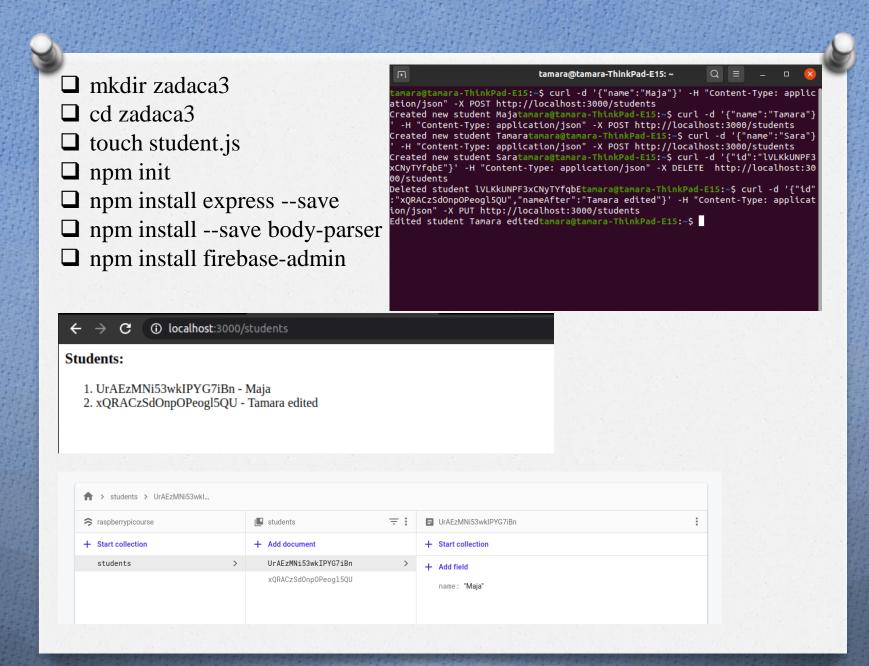
Тамара Митевска Факултет за електротехника и информациски технологии





Задача 2

Креирај node.js express сервер на raspberry рі кој податоците ќе ги
сочувува во Cloud Firestore база на податоци и ќе може да се
справи со следниве барања:
□ /students – GET метод кој ќе врати HTML документ со листа од
студенти
□ /students – POST метод кој ќе се користи за додавање на нови
студенти (се проследува само име на студент како патаметар)
□ /students – PUT метод кој ќе се користи за изменување на веќе
постоечки студент (се проследува id и ново име на студент како
патаметри)
□ /students – DELETE метод кој ќе се користи за бришење на
студент (се проследува само id на студент како патаметар)



```
Open
        | ▼ | .....
        express = require('express')
        bodyParser = require('body-parser');
                                                                                            36 .post(1
                                                                                                             (req, res) {
                                                                                            37
                                                                                                      name = req.body.
        app = express()
                                                                                            38 db.collection('students').add({
        port = 3000
                                                                                                name: name,
 7 app.use(bodyParser.json()) // support json encoded bodies
                                                                                            41
                                                                                                    res.send('Created new student ' + name)
                                                                                            42
                                                                                                    res.end()
 8 app.use(bodyParser.urlencoded({ extended: true })) // support encoded bodies
                                                                                            44 })
        admin = require('firebase-admin')
10
                                                                                            45 .put(f
                                                                                                          n (req, res) {
11
                                                                                                      id = req.body.id
        serviceAccount = require('./firebaseKey.json')
                                                                                            47
                                                                                                      studentNameAfter = req.body.nameAfter
                                                                                            48 db.collection('students').doc(id).set({
                                                                                                name: studentNameAfter,
14 admin.initializeApp({
                                                                                            50 }).then(() => {
                                                                                                   res.send('Edited student ' + studentNameAfter)
15 credential: admin.credential.cert(serviceAccount)
                                                                                                   res.end()
16 });
17
        db = admin.firestore();
                                                                                            54 })
18
                                                                                                               (req, res) {
19
                                                                                                      id = req.body.id
      students = []
                                                                                            57 db.collection('students').doc(id).delete().then(() => {
                                                                                                   res.send('Deleted student ' + id)
21 app.route('/students')
                                                                                                   res.end()
22 .get(fun
                                                                                            60 });
                (req, res) {
                                                                                            61 })
    res.writeHead(200, { 'Content-Type': 'text/html' })
                                                                                            62
24
        data = '<html><body>'
                                                                                            63 app.listen(port, () => {
25
                                                                                            64 console.log(`Example app listening at http://localhost:${port}`)
                                                                                            65 })
    db.collection('students').get().then(doc => {
27
        (doc) {
28
          data+='<h3>Students:</h3>'
29
          doc.forEach(student => data+='' + student.id + ' - ' + student.data().name + '')
30
          data+=''
31
32
      res.write(data + '</body></html>')
33
      res.end()
34 });
```

Тамара Митевска

Факултет за електротехника и информациски технологии





Задача 2

Креирај node.js express сервер на raspberry рі кој податоците ќе ги
сочувува во Cloud Firestore база на податоци и ќе може да се
справи со следниве барања:
□ /courses – GET метод кој ќе врати HTML документ со листа од

- курсеви

 /courses POST метод кој ќе се користи за додавање на нови курсеви (име на курс, почетен и краен датум како патаметри)
- □ /courses/:id PUT метод кој ќе се користи за изменување на веќе постоечки курс (се проследува име на курс, почетен и краен датум како патаметар)
- □ /courses/:id DELETE метод кој ќе се користи за бришење на курс

Прашања?

БЛАГОДАРАМ НА ВНИМАНИЕТО! [©]