

რეგისტრაციის ფორმა

დავალებაში უნდა ააწყოთ რეგისტრაციის ფორმა, რომ იქნება დანერგილი VueJS 3.0 Framework-ზე და სასურველია TypeScript-ის გამოყენებით. გარდა ამისა უნდა გამოიყენოთ Vuex (State Management), რომ შეინახოთ მონაცემები და მოახდინოთ მასზე მანიპულაციები, როგორც არის ჩანერა, ნაკითხვა და სერვერიდან წამოღება.

საიტი უნდა ერგებოდეს როგორც კომპიუტერს, ასევე მობილურს (Responsive Layout).

საიტი უნდა შედგებოდეს ორი გვერდისგან

1. <http://localhost:3000/>
2. <http://localhost:3000/dashboard/>

პირველზე უნდა გამოდიოდეს რეგისტრაციის ფორმა, ხოლო მეორეზე ჩამონათვალი და ფილტრი უკვე რეგისტრირებულების.

რეგისტრაციის შემდეგ უნდა გადამისამართდეს dashboard-ის გვერდზე, როუტერისთვის უნდა გამოიყენოთ Vue-Router.

რეგისტრაციის ველები

1. სახელი
2. გვარი
3. პირადობის ნომერი (11 ციფრი)
4. დაბადების თარიღი (html5-ის კალენდარი)
5. სქესი (ასარჩევი ველი)

სასურველია რაიმე სახის ვალიდაცია ედიტოს პირადობის ნომერს.

ფორმამ შენახვისას არ უნდა შეცვალოს გვერდი, მისი მონაცემები უნდა fetch-ით, ან axios-ით გაიგზავნოს POST მეთოდით სერვერზე და სერვერიდან წარმატების დაბრუნების შემთხვევაში უნდა გადაასამიშროდ მეორე გვერდზე.

პრაქტიკაში ვიყენებთ axios პაკეტს, ხოლო fetch ბრაუზერში ჩაშენებული ფუნქცია არის და index.html-ში დანერგილია მაგალითი, როგორ შეიძლება ამ ფუნქციით post მეთოდის გაშვება.

რეგისტრირებულების გვერდი

1. მონაცემები უნდა იფილტრებოდეს:
 - სახელი, გვარდით
 - დაბადების თარიღით
 - პირადობის ნომრით
 - სქესით
2. უნდა იყოს არსებული მომხმარებლების ჩამონათვალი, html5-ის table-ში, შემდეგი სვეტებით
 - სრული სახელი (სახელი და გვარი)
 - პირადობის ნომერი
 - სქესი
 - დაბადების თარიღი

ფილტრის დროს მონაცემების განახლება უნდა მოხდეს დინამიურად.

სერვერი

server.js ფაილში აწყობილია მინიმალური სერვერ express.js და body-parser-ის გამოყენებით, ორივე npm-ით ყენდება. სიმარტივისთვის ბაზის როლში გამოყენებულია json ფაილი. სერვერის ფაილში განერილია api ლინკები, საიდანაც უნდა წამოიღოთ და ჩაწეროთ ინფორმაცია.