Messenger

Project 2

**მოთხოვნა:**

დაკლონეთ git**-**ის გამოყენებით ატვირთლი [პროექტის შაბლონი](https://github.com/sabazed/MZIURI-Project-2), რომელშიც შემდგომში იმუშავებთ და pushგააკეთებთ ცალკე branch-ზე, რომელსაც ერქმევა სახელი შემდეგი ფორმატით: name\_surname, სადაც ჩაწერთ თქვენს სახელს და გვარს.  
  
**აღწერა:**

პროექტის ფარგლებში თქვენ უნდა გააკეთოთ პატარა საიტი, სადაც შეძლებენ მომხმარებლები დარეგისტრირებას, ანონიმური მესიჯების გაგზავნას და წაკითხვას.

1. **შექმენით ორი servlet**:

* MessageServlet (/message):

მეთოდები:

* + [GET]

მომხმარებლის მესიჯების წაკითხვის მეთოდი, რომელსაც პარამეტრად გადაეცემა username და password, რის შემდეგაც ამ მონაცემების შესაბამისი მომხმარებლის მესიჯები ჩაიწერება response ობიექტში და ისე დაბრუნდება.   
P.s: შეგიძლიათ, სიმარტივისთვის, ყველა მესიჯი ერთმანეთთან გადააბათ ახალი ხაზის (\n) ან რაიმე სხვა უნიკალური სიმბოლოთი, რომელიც მესიჯში არ იქნება (აკრძალავთ), რის შედეგადაც ბრაუზერში გამოსაჩენად შეძლებთ დაბრუნებული response body-ის დაყოფას ამ სიმბოლოთი (String-ის split)

* + [POST]

მომხმარებლისათვის მესიჯის გაგზავნის მეთოდი, რომელსაც პარამეტრებად გადმოეცემა username და message, რის შემდეგაც ამ username-ის მქონე მომხმარებელს გაუგზავნით message პარამეტრად გადმოცემულ ტექსტს, რაც ასევე შეინახება მონაცემთა ბაზაში. თუ ამ username-ს მქონე მომხმარებელი არ არსებობს ან message პარამეტრი შეიცავს ახალი ხაზის (\n) სიმბოლოს, response უნდა დაბრუნდეს 403 კოდით და ბაზაში არაფრის შენახვა არ უნდა მოხდეს.

* UserServlet (/user):  
  მეთოდები:
  + [POST]

მომხმარებლის სარეგისტრაციო მეთოდი, რომელსაც პარამეტრებად გადმოეცემა username და password, რის შემდეგაც ამ მონაცემებით მომხმარებელი შეინახება მონაცემთა ბაზაში. თუ იგივე username-ს მქონე მომხმარებელი უკვე დარეგისტრირებულია, response უნდა დაბრუნდეს 403 კოდით.

1. **გამოიყენეთ MySql მონაცემთა ბაზა**, სადაც გექნებათ user-ების და message-ების შესაბამისი table-ები. ბაზასთან სამუშაოდ გამოიყენეთ JDBC, რომლის შესაბამის dependency-საც დაამატებთ build.gradle-ში. შექმენით ცალკე ბაზასთან სამუშაო კლასი, რომლის ობიექტიც იქნება singleton ტიპის და ორივე სერვლეტს ექნება ამ ობიექტზე წვდომა.
2. **Bonus:** ბრაუზერიდან მიღებული მესიჯის ვალიდაციისათვის შექმენით ასევე ცალკე კლასი, რომლის ობიექტიც იქნება singleton და რომელზეც ექნება წვდომა შესაბამის სერვლეტს. ასევე დაწერეთ ამ მესიჯების ვალიდატორის შესაბამისი unit test-ები.
3. **Front-end:** თქვენ ვებსაიტს შეგიძლიათ დაუმატოთ უამრავი რამ, გამოიყენეთ კრეატიულობა, შეუცვალეთ სტილები და ა.შ. მთავარია, რომ საიტზე გქონდეთ შემდეგი სამი კომპონენტი:
   1. **მომხმარებლის სარეგისტრაციო ფორმა:**ფორმას უნდა ჰქონდეს ორი ველი, username-ის და password-ის, და ღილაკი, რომელიც ახდენს UserServlet-ზე POST request-ის გაგზავნას და ზემოთ არსებულ ველებში შეყვანილ მნიშნელობებს იყენებს პარამეტრებად
   2. **მესიჯის გაგზავნის ფორმა:**ფორმას უნდა ჰქონდეს ორი ველი, username-ის და message-ის, და ღილაკი, რომელიც ახდენს MessageServlet-ზე POST request-ის გაგზავნას და ზემოთ არსებულ ველებში შეყვანილ მნიშნელობებს იყენებს პარამეტრებად
   3. **მესიჯების წასაკითხი ფორმა:**ფორმას უნდა ჰქონდეს ორი ველი, username-ის და password-ის, და ღილაკი, რომელიც ახდენს UserServlet-ზე GET request-ის გაგზავნას და ზემოთ არსებულ ველებში შეყვანილ მნიშნელობებს იყენებს პარამეტრებად, ხოლო response body-ში მიღებულ მესიჯებს უჩვენებს მომხმარებელს ბრაუზერში (იმპლემენტაცია თქვენზეა, იყავით კრეატიულები)
   4. **მოთხოვნები:** 
      1. საჭიროა, რომ სამივე ფორმის წარმატებულად გაგზავნის შემდგომ (response status 200), ფორმის ველები უნდა დაცარიელდეს/გასუფთავდეს
      2. პაროლის ველი უნდა იყოს კონფიდენციალური (არ შეეძლოს მომხმარებელს შეყვანილი პაროლის წაკითხვა, გამოიყენეთ შესაბამისი input ელემენტის ატრიბუტი)
      3. ახალი მესიჯის გაგზავნისას კი, მომხმარებელმა ვერ უნდა შეძლოს მესიჯში ახალ ხაზზე გადასვლა (ანუ მესიჯი არ უნდა შეიცავდეს "\n"-ს)
      4. ყოველი request-ის გაგზავნის შემდგომ, მიღებული წარმატებული ან წარუმატებელი response-ის შესაბამისი მესიჯი უნდა ვუჩვენოთ მომხმარებელს alert()-ის ან რაიმე სხვა მეთოდით (მაგალითად თუ მომხმარებელი ცდილობს დარეგისტრირებას და პასუხი უარყოფილია, alert()-ით ვაჩვენოთ „შეყვანილი მამხმარებლის სახელი უკვე დარეგისტრირებულია“, ან დასტურის შემთხვევაში „თქენ წარმატებით დარეგისტრირდით“; საჩვენებელი ტექსტი შეგიძლიათ შეცვალოთ, მთავარია იყოს ლოგიკური)

**ინსტრუქცია:**

გადმოწერილი პროექტის გაშვების შემდგომ, შეგიძლიათ შეხვიდეთ <http://localhost:8989/messenger/index.html> და იხილოთ თქვენი საიტი. ხოლო საიტის კოდის შესაცვლელად შეგიძლიათ შეცვალოთ src/main/webapp/index.html ფაილი (ეს არის საითს მთავარი ფაილი, რომელიც არ უნდა წაშალოთ და უნდა გამოიყენოთ) და დაამათოთ თქვენი სასურველი ნებისმიერი ფაილი/რესურსი ამავე webapp ფოლდერში

**ვებსაიტის მაგალითი (დამაკმაყოფილებელი):**

A screenshot of a computer

Description automatically generated

**სერვერის დიზაინის მაგალითი:**

A diagram of a diagram

Description automatically generated