Task Management

შევქმნათ დავალებების მართვის სისტემა შემდეგი ფუნქციებით

1. პროექტები
2. დავალებები
3. მომხმარებლები
4. მომხმარებლის როლები და უფლებები

აპლიკაციაში შესაძლებელი უნდა იყოს მომხმარებლის რეგისტაცია და ავტორიზაცია.

ასევე ავტორიზაციის გავლის შემდეგ მომხმარებელმა უნდა შექმნას მინიმუმ ერთი პროექტი თავისი კონფიგურაციით

პროექტის კონფიგურაცია მოიცავს შემდეგ კონფიგურაციას

1. პროექტის დასახელება
2. ბორდის სახელი
   1. ბორდის კოლონები (მინიმუმ სამი კოლონა)
      1. ToDo
      2. In Progress
      3. Done
3. დავალების ტიპები (მაგ: Task, Bug, Test, Spike)

დავალებაზე შესაძლებელია დამატებითი ველების დამატება რომელიც იქნება ტექსტური ველები

1. პროექტზე მომხმარებლების მიმაგრება

პროექტის შექმნის შემდგომ აპლიკაციაში უნდა გამოჩნდეს პროექტი/ები რომლებიც დამატებული მაქვს ან რომელ პროექტშიც ვარ მონიშნული

პროექტის არჩევის შემდგომ გამოჩნდება ის ბორდი და დავალებები რომელიც ეკუთვნის ამ პროექტს.

ბორდზე უნდა გამოჩნდეს კანბან კოლონები რომელშიც ჩაიყრება დავალებები და შესაძლებელი იქნება ამ დავალებების მოძრაობა კოლონებს შორის, დავალებამ კოლონა უნდა შეიცვალოს სტატუსის მიხედვით, შესაძლებელიი უნდა იყოს ასევე დავალების გადატანა კოლონებს შორის drug and drop მეთოდითაც,

**მომხმარებლები**

მომხმარებლებს გააჩნიათ როლები და უფლებები

მომხმარბეელს ენიჭება როლი რომელზეც არის გაწერილი გარკვეული უფლებები

უფლებები გაწერილია მოქმედებების მიხედვით მაგ.: პროეტის შექმნა, დავალების დამატება, მომხმარებლის დამატება, როლის მინიჭება და ა.შ.

შესაძლებელი უნდა იყოს ასევე როლების დამატება/ რედაქტირება/ წაშლა, როლებზე უნდა მოხდეს შესაბამისი უფლებების გაწერა.

ყველა დამატებულ მომხმარებელს შეუძლია ავოტორიზაციის გავლა და იმ პროექტის მართვა რომელშიც არის დამატებული

აპლიკაციაში უნდა მოხდეს მოქმედებებზე შეზღუდვა უფლებებიდან გამომდინარე. ღილაკის დამალვა და route-ს შეზღუდვა

GUIDE

როგორ უნდა ვმართოთ პროცესი

გუნდი ქმნის რეპოზიტორს git-ზე, main და develop ბრენჩს.

გუნდის წევრი მუშაობს შემდეგი ნაბიჯებით,

დავალების მიხედვით develop ბრენშიდან ქმნის ახალ ბრენჩს დავალების სახელით, მუშაობს ამ დავალებაზე ამ ბრენჩზე ხოლო როდესაც დაასრულებს მუშაობას ვაკეთებთ pull request-ს develop ბრენჩზე, ვნიშნავთ ჩვენი გუნდის წევრებს და ლექტორს, რომ გადახედონ დავალებას და შენიშვნების ან რჩევის შემთხვევაში pull request-ში დაწეროს კომენტარი.

ხოლო როდესაც ყველა საკითხი ამოიწურება დაადასტუროს. გუნდის ყველა წევრის დადასტურების შემდგომ ლექტორი შემერჯავს ბრენჩი develop ბრენჩთან.

თუ არსებულ დავალების შემდგომ წარმოიშვა bug, ვასწორებთ იმავე ბრენჩზე რომელზეც გავაეთეთ დავალება.

ყოველი კვირის ბოლოს დასრულებული დავალება რომელიც შესაძლებელია რომ უკვე მოხვდეს რელიზში, ვაკეთებთ pull request-ს main ბრენჩზე.

Main- ბრენჩზე უნდა მოხვდეს მხოლოდ დასრულებული. დავალებები.

იმ შემთქვევაში თუ რელიზში არსებულ ვერსიაში ნაპოვნია bug, ვაკეთებთ ახალ ბრენჩს main ბრენჩიდან, ვმუშაობთ ამ ბრენჩზე, დასრულების შემდგომ გადაგვაქვს develop ბრენჩზე, ხოლო კვირის ბოლოს გასწორებული bug გადაგვაქვს main ბრენჩზე,

ასეთი ციკლით მოხდება ყველა დავალებების და bug-ების შესრულება.

დავალების დაგეგმვა/ მართვა

დავალებას გუნდი გეგმავს ერთობლივად. საკითხები იყოფა რაც შეიძლება პატარა ნაწილებად. და ხდება სირთულის შეფასება.

დავალებების განაწილება ხდება შეთანხმებით.

ასევე მნიშვნელოვანია რომ დავალებების გადანაწილება მოხდეს თანაბარი წონის მიხედვით.

დავალება უნდა ავიღოთ ისეთი წონის რომლის შესრულებასაც შევძლებთ ერთი კვირის განმავლობაში.

დავალების დაგეგმვა უნდა მოხდეს კვირაში ერთხელ, ხოლო კვირის ბოლოს მოხდეს ამ დავალებების ჩაბარება, თუ რომელიმე დავალება ვერ მოესწრო, უნდა მოხდეს ამ საკითხის გავლა და გაანალიზება რატომ ვერ მოესწრო, რათა შემდგომ კვირაში გამოსწორდეს თუ შესაძლებელია.

კვირის განმავლობაში დავალების შესრულებაში მაქსიმალურად ვეხმარებით ერთმანეთს გადაწყვეტილების და სოლუშენების პოვნაში, ასევე მივმართავთ ლექტორს დროულად თუ ვერ მოხდა სწორი სოლუშენის პოვნა.

საკომუნიკაციოთ ვიყენებთ ნებისმიერ არსებულ საშუალებას.

დავალებების სამარტავად შეგიძლიათ გამოიყენოთ: <https://trello.com>, <https://clickup.com/> ან ნებისმიერი სხვა.