

ფინალური პროექტი

მითითებული დიზაინის მიხედვით შექმენით ვებ გვერდი, რომელშიც იქნება ორი გვერდი.

პირველ გვერდზე გამოიტანეთ News-ების სია, რომელიც დაბრუნდება სერვერიდან, თითოეული ჩანაწერის გასწვრივ დაამატეთ წაშლის ღილაკი რომელზე დაჭერაც გამოიძახებს სერვერს და წაშლის კონკრეტულ News-ს, ასევე წაშლის დროს სიიდან მისი წაშლა განახორციელეთ ანიმაციის საშუალებით.

News-ების სია ავტომატურად უნდა იტვირთებოდეს სერვერიდან გვერდის ჩატვირთვის დროს, და წაშლის დროს ავტომატურად იშლებოდეს ანიმაციის საშუალებით.

მეორე გვერდიდან შესაძლებელი უნდა იყოს ახალი News-ის შექმნა. ასევე უნდა უზრუნველყოთ ვალიდაცია, რადგან სავალდებულო ველების გარეშე არ მოხდეს პოსტის შექმნა. კატეგორიის ველი უნდა იყოს ასარჩევი Select, სადაც თქვენი სურვილისამებრ შეგიძლიათ დაამატოთ კატეგორიები. ასევე Save ღილაკის დაჭერისას ავტომატურად გადაამისამართეთ News List გვერდზე სადაც მოხდება ახალი დამატებული News-ის გამოტანა.

ბონუს დავალება: პროექტის ფარგლებში სავალდებულო არ არის, მაგრამ შეგიძლიათ გააკეთოთ თითოეული News პოსტის განახლების ფუნქციონალიც, ამ შემთხვევაში შესაძლებლობა გექნებათ მიიღოთ ბონუს ქულა, თუმცა არ შესრულების შემთხვევაში ქულა არ დაგაკლდებათ.

დიზაინი შეგიძლიათ იხილოთ:

<https://www.figma.com/design/EKazCsZqpzCW0yrobPmhUK/Figma-basics?node-id=1669-162202>

API მისამართები:

1. News-ების სიის წამოღება

HTTP Method: GET

URL: <https://btu-exam-cb6c3fdf3b9d.herokuapp.com/news>

2. News-ის წაშლა

HTTP Method: DELETE

URL: <https://btu-exam-cb6c3fdf3b9d.herokuapp.com/news/{:id}>

3. News-ის შექმნა

HTTP Method: POST

URL: <https://btu-exam-cb6c3fdf3b9d.herokuapp.com/news>

Body: {
 title,
 description,
 category,
 editorFirstName,
 editorLastName
}

4. News-ის განახლება

HTTP Method: PUT

URL: <https://btu-exam-cb6c3fdf3b9d.herokuapp.com/news/{:id}>

Body: {
 title,
 description,
 category,
 editorFirstName,
 editorLastName
}

5. ერთი News-ის წამოღება

HTTP Method: GET

URL: <https://btu-exam-cb6c3fdf3b9d.herokuapp.com/news/{:id}>