



УНИВЕРЗИТЕТ У НОВОМ САДУ  
ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА  
Трг Доситеја Обрадовића бр. 6, Нови Сад




**Катарина Нинковић**

# **Трослојна архитектура као софтверско решење интернет продавнице**

МАСТЕР РАД  
- Мастер академске студије -

**Ментор:**  
**др Предраг Теодоровић**

Нови Сад, октобар 2022.

	УНИВЕРЗИТЕТ У НОВОМ САДУ • ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА 21000 НОВИ САД, Трг Доситеја Обрадовића 6	Датум:
	<b>ЗАДАТАК ЗА ИЗРАДУ МАСТЕР РАДА</b>	Лист/Листова:
		2/57

(Податке уноси предметни наставник - ментор)

Врста студија:	<input type="checkbox"/> Мастер академске студије
Студијски програм:	<b>Енергетика, електроника и телекомуникације</b>
Руководилац студијског програма:	<b>др Јован Бајић</b>

Студент:	<b>Катарина Нинковић</b>	Број индекса:	<b>Е1-61/2020</b>
Област:	<b>Електроника</b>		
Ментор:	<b>др Предраг Теодоровић</b>		

НА ОСНОВУ ПОДНЕТЕ ПРИЈАВЕ, ПРИЛОЖЕНЕ ДОКУМЕНТАЦИЈЕ И ОДРЕДБИ СТАТУТА ФАКУЛТЕТА ИЗДАЈЕ СЕ ЗАДАТАК ЗА МАСТЕР РАД, СА СЛЕДЕЋИМ ЕЛЕМЕНТИМА:

- проблем – тема рада;
- начин решавања проблема и начин практичне провере резултата рада, ако је таква провера неопходна;
- литература

### НАСЛОВ МАСТЕР РАДА:

<b>Трослојна архитектура као софтверско решење интернет продавнице</b>
--

### ТЕКСТ ЗАДАТКА:

У оквиру овог мастер рада потребно је урадити следеће:

- Подесити развојно окружење и инсталирати неопходне алате
- Креирати одвојене пројекте за серверски и клијентски део апликације
- Подесити базу података и преузимање иницијалних података о артиклима
- Омогућити прикупљање артикала од базе на клијентском претраживачу
- Имплементирати евиденцију корисника
- Имплементирати креирање и плаћање поруџбина
- Имплементирати администраторски простор за управљање сајтом
- Имплементирати активацију налога путем *e-mail* порука и осталих обавештења
- Подржати рецензије производа
- Креирати леп и пријатан кориснички интерфејс
- Тестирати пројектовани систем и написати мастер рад

Руководилац студијског програма:	Ментор рада:
др Јован Бајић	др Предраг Теодоровић

Примерак за: <input type="checkbox"/> - Студента; <input type="checkbox"/> - Ментора
--



УНИВЕРЗИТЕТ У НОВОМ САДУ • ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА  
21000 НОВИ САД, Трг Доситеја Обрадовића 6

## КЉУЧНА ДОКУМЕНТАЦИЈСКА ИНФОРМАЦИЈА

Редни број, <b>РБР:</b>			
Идентификациони број, <b>ИБР:</b>			
Тип документације, <b>ТД:</b>	Монографска публикација		
Тип записа, <b>ТЗ:</b>	Текстуални штампани материјал		
Врста рада, <b>ВР:</b>	Мастер рад		
Аутор, <b>АУ:</b>	Катарина Нинковић		
Ментор, <b>МН:</b>	др Предраг Теодоровић, доцент		
Наслов рада, <b>НР:</b>	Трослојна архитектура као софтверско решење интернет продавнице		
Језик публикације, <b>ЈП:</b>	Српски / ћирилица		
Језик извода, <b>ЈИ:</b>	Српски		
Земља публикавања, <b>ЗП:</b>	Република Србија		
Уже географско подручје, <b>УГП:</b>	Аутономна Покрајина Војводина		
Година, <b>ГО:</b>	2022.		
Издавач, <b>ИЗ:</b>	Ауторски репринт		
Место и адреса, <b>МА:</b>	Нови Сад; трг Доситеја Обрадовића 6		
Физички опис рада, <b>ФО:</b> (поглавља/страна/ цитата/табела/слика/графика/прилога)	(6/48/15/12/46/0/0)		
Научна област, <b>НО:</b>	Електротехника и рачунарство		
Научна дисциплина, <b>НД:</b>	Рачунарска техника		
Предметна одредница/Кључне речи, <b>ПО:</b>	Web, Node, React, Redux, MongoDB		
<b>УДК</b>			
Чува се, <b>ЧУ:</b>	Библиотека Факултета техничких наука, Нови Сад		
Важна напомена, <b>ВН:</b>			
Извод, <b>ИЗ:</b>	У овом чланку је описана имплементација и рад интернет продавнице. Реч је о трослојном систему који укључује имплементацију сервера и клијента, као и креирање и управљање базом података. Пројекат представља приступ развијања модерних web апликација уз последње технологије и програмска окружења као што су Node, React, Redux и MongoDB. Подржана је евиденција купаца и поруџбина, динамичко прикупљање производа и осталих информација, администраторски простор за управљање сајтом и друге функционалности које покрива савремена интернет продавница.		
Датум прихватања теме, <b>ДП:</b>			
Датум одбране, <b>ДО:</b>			
Чланови комисије, <b>КО:</b>	Председник:	др Калман Бабковић, ванредни професор	Потпис ментора
	Члан:	др Мирко Раковић, ванредни професор	
	Члан, ментор:	др Предраг Теодоровић, доцент	



UNIVERSITY OF NOVI SAD • FACULTY OF TECHNICAL SCIENCES  
21000 NOVI SAD, Trg Dositeja Obradovića 6

## KEY WORDS DOCUMENTATION

Accession number, <b>ANO</b> :			
Identification number, <b>INO</b> :			
Document type, <b>DT</b> :	Monographic publication		
Type of record, <b>TR</b> :	Textual printed material		
Contents code, <b>CC</b> :	Master thesis		
Author, <b>AU</b> :	Katarina Ninković		
Mentor, <b>MN</b> :	PhD Predrag Teodorović, docent		
Title, <b>TI</b> :	Three-layer architecture as a software solution for an online store		
Language of text, <b>LT</b> :	Serbian / Cyrillic		
Language of abstract, <b>LA</b> :	English		
Country of publication, <b>CP</b> :	Republic of Serbia		
Locality of publication, <b>LP</b> :	Autonomous Province of Vojvodina		
Publication year, <b>PY</b> :	2022.		
Publisher, <b>PB</b> :	Author's reprint		
Publication place, <b>PP</b> :	Faculty of Technical Sciences, Dositeja Obradovica sq. 6, 21000, Novi Sad		
Physical description, <b>PD</b> : (chapters/pages/ref./tables/pictures/graphs/appendixes)	(6/48/15/12/46/0/0)		
Scientific field, <b>SF</b> :	Electrical engineering and computer science		
Scientific discipline, <b>SD</b> :	Computer technology		
Subject/Key words, <b>S/KW</b> :	Web, Node, React, Redux, MongoDB		
<b>UC</b>			
Holding data, <b>HD</b> :	The Library of Faculty of Technical Sciences, Novi Sad, Serbia		
Note, <b>N</b> :			
Abstract, <b>AB</b> :	This article describes implementation and working of web shop. It's about a three-layer system which includes the implementation of server and client, as well as the creation and management of database. Project presents approach to development of modern web application including latest technologies such as Node, React, Redux and MongoDB. It supports notation of customers and orders, dynamical loading of products and rest of the information, admin space for site management and other functionalities which web shops cover nowadays.		
Accepted by the Scientific Board on, <b>ASB</b> :			
Defended on, <b>DE</b> :			
Defended Board, <b>DB</b> :	President:	Kalman Babković, PhD, associate professor	
	Member:	Mirko Raković, PhD, associate professor	Mentor's sign
	Member, Mentor:	Predrag Teodorović, PhD, docent	

## Изјава о академској честитости

Студент/киња: Катарина Нинковић

Број индекса: Е1-61/2020

Студент/киња: мастер академских студија Факултета техничких наука

Аутор/ка рада под називом:

Трослојна архитектура као софтверско решење интернет продавнице

Потписивањем изјављујем:

- да је рад искључиво резултат мог сопственог истраживачког рада;
- да сам рад и мишљења других аутора које сам користио/ла у овом раду назначио/ла или цитирао/ла и да су наведени у списку литературе/референци који су саставни део овог рада;
- да сам добио/ла све дозволо за коришћење ауторског дела који се у потпуности/целости уносе у предати рад и да сам то јасно навео/ла;
- да сам свестан/на да је плагијат коришћење туђих радова у било ком облику (као цитата, парафраза, слика, табела, дијаграма, дизајна, планова, фотографија, филма, музике, формула, веб сајтова, компјутерских програма и сл.) без навођења аутора или представљање туђих ауторских дела као мојих, кажњиво по закону (Закон о ауторским и сродним правима, Службени гласник Републике Србије, бр. 104/2009, 99/2011, 119/2012), као и других закона и одговарајућих аката Универзитета у Новом Саду;
- да сам свестан/на да плагијат укључује и представљање, употребу и дистрибуирање рада предавача или других студената као сопствених;
- да сам свестан/на последица које код доказаног плагијата могу проузроковати на предати рад и мој статус;
- да је електронска верзија рада идентична штампаном примерку и пристајем на његово објављивање под условима прописаним актима Универзитета.

Нови Сад, 28.10.2022

Потпис студента/киње:

---

## **Захвалност**

Посебну захвалност дугујем својим родитељима и дечку који су били велика подршка и чврст ослонац током досадашњег школовања.

Захваљујем се ментору др Предрагу Теодоровићу на саветима, стрпљењу и стручној помоћи приликом израде овог рада.

# Садржај

1. Увод .....	1
2. Теоријске основе.....	2
2.1 <i>Node.js</i> .....	2
2.2 <i>MongoDB</i> .....	3
2.3 <i>React</i> .....	4
2.4 <i>Redux</i> .....	5
3. Систем.....	6
3.1 Серверска страна.....	6
3.1.1 База података .....	7
3.1.2 Модели .....	9
3.1.3 Руте .....	10
3.1.4 Контролери .....	13
3.1.5 <i>Sendgrid</i> .....	15
3.2 Клијентска страна .....	16
3.2.1 <i>Axios</i> и <i>Redux</i> .....	17
3.2.2 Елементи и компоненте .....	20
3.2.3 <i>Bootstrap</i> .....	24
3.2.4 <i>PayPal</i> дугме .....	26
4. Функционалности .....	27
4.1 Кориснички мени .....	27
4.1.1 Почетна страна и странице производа .....	27
4.1.1 Детаљан приказ производа .....	29
4.1.2 Претрага производа.....	31
4.1.3 Регистрација и активација налога.....	32
4.1.4 Аутентикација.....	34
4.1.5 Пријава и одјава корисника.....	35
4.1.6 Ресетовање шифре.....	36
4.1.7 Ажурирање профила .....	37
4.1.9 Корпа .....	38
4.1.10 Поруџбине.....	39
4.1.11 Плаћање.....	40
4.2 Администраторски мени .....	41
4.2.1 Преглед и (де)активација корисника .....	42

4.2.2 Преглед и манипулација производима.....	43
4.2.3 Преглед, достављање и отказивање поруџбина .....	45
5. Закључак.....	47
6. Литература .....	48

## Списак слика

Слика 1. Блок шема целокупног система .....	1
Слика 2. Поређење Multi-threading vs. Node.js сервера [5] .....	2
Слика 3. Mongoose као спрега између MongoDB и Node.js .....	3
Слика 4. Приказ виртуалног DOM-а приликом промене објекта и приказ рендеровања истих у реалном DOM-у [6] .....	4
Слика 5. Релације између Redux компоненти [7] .....	5
Слика 6. Ток ажурирања стања на Redux-у [7] .....	5
Слика 7. Блок шема серверске стране.....	6
Слика 8. Приступање бази помоћу MongoDB Compass.....	8
Слика 9. SendGrid лого [8].....	15
Слика 10. Блок шема клијентске стране .....	16
Слика 11. Приказ компонената на почетној страници .....	20
Слика 12. PayPal Button .....	26
Слика 13. Приказ почетне стране и листе производа .....	27
Слика 14. Приказ ограничених производа по једној страни .....	28
Слика 15. Детаљан приказ једног од производа.....	29
Слика 16. Приказ падајућег менија у пољу количина .....	30
Слика 17. Промењено стање у падајућем менију након креирања поруџбине.....	30
Слика 18. Производи након извршене претраге .....	31
Слика 19. Прозор за регистрацију .....	32
Слика 20. Грешка приликом регистрације корисника који већ постоји .....	32
Слика 21. Приказ e-mail поруке за активацију корисничког налога .....	33
Слика 22. Порука о успешном активирању корисничког налога .....	33
Слика 23. Грешка приликом приступа корисничким подацима без токена .....	34
Слика 24. Прозор за пријаву .....	35
Слика 25. Грешка приликом погрешног уноса e-mail адресе или лозинке .....	35
Слика 26. Изглед навигационог менија након пријаве корисника .....	35
Слика 27. Прозор за слање захтева за ресетовање шифре .....	36
Слика 28. Приказ e-mail поруке за ресетовање шифре .....	36
Слика 29. Прозор за унос нове лозинке .....	36
Слика 30. Приказ корисничког профила .....	37
Слика 31. Приказ корпе у коју су смештени неки од производа .....	38
Слика 32. Детаљан преглед креиране поруџбине .....	39
Слика 33. Преглед статуса креиране поруџбине у корисничком профилу .....	39
Слика 34. Провера података на поруџбини и могућност њеног плаћања.....	40
Слика 35. Промена статуса поруџбине у плаћено.....	40
Слика 36. Ажурирано поље на плаћено у детаљном приказу поруџбине.....	40
Слика 37. Приказ навигационог менија администратора .....	41



Слика 38. Листа свих корисника .....	42
Слика 39. Грешка приликом покушаја пријаве корисника који је деактивиран .....	42
Слика 40. Листа свих производа .....	43
Слика 41. Прозор за ажурирање производа .....	43
Слика 42. Креирање новог производа .....	44
Слика 43. Листа свих поруџбина .....	45
Слика 44. Детаљан приказ једне од поруџбина коју је потребно доставити .....	45
Слика 45. Ажурирано стање на достављено у корисником профилу.....	46
Слика 46. Обавештење о достављеној поруџбини путем e-mail адресе.....	46

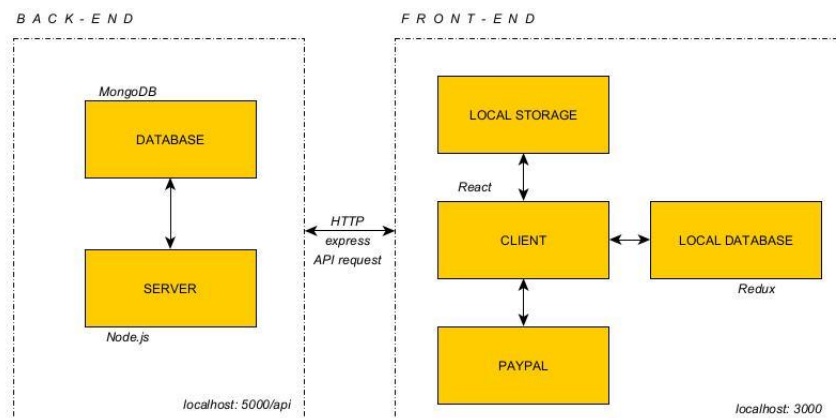
## Списак табела

Табела 1. Постојеће руте и одговарајући HTTP захтеви за производе .....	10
Табела 2. Постојеће руте и одговарајући HTTP захтеви за кориснике .....	11
Табела 3. Постојеће руте и одговарајући HTTP захтеви за поруџбине .....	12
Табела 4. Постојећи контролери за управљање производима .....	13
Табела 5. Постојећи контролери за управљање поруџбинама .....	13
Табела 6. Постојећи контролери за управљање корисницима .....	14
Табела 7. Листа редуктора и модела стања за продукте .....	17
Табела 8. Листа редуктора и модела стања за кориснике .....	18
Табела 9. Листа редуктора и модела стања за поруџбине .....	19
Табела 10. Постојеће front-end руте и одговарајући прикази .....	21
Табела 11. Постојеће компоненте и њихова релација са страницама (рутама) .....	22
Табела 12. Постојеће коришћене Bootstrap компоненте .....	24

# 1. Увод

У данашње време процес *online* куповине или куповине преко интернета постао је општеприхваћен и заступљен у модерном друштву. Динамика живота данашњег човека често оставља без могућности да пролази кроз многобројне радње у потрази за оним што му је потребно, што поред напорног обилажења, укључује и велик губитак слободног времена, те овакав вид куповине представља олакшавајућу околност. Путем интернет продавнице (енг. *Web Shop*) корисник је у могућности да прегледа све производе, одабере оне одговарајуће и “пошаље их у корпу”, одакле ће даље моћи да изврши куповину.

У оквиру овог пројекта реализован је систем који се састоји од два основна блока. Први блок представља, серверску страну, колоквијално названу *Back-end*, која укључује серверску апликацију базирану на *Node.js* [1] платформи и њој специфичном *JavaScript* (у даљем тексту *JS*) програмском језику, и базу података, која је у овом случају *MongoDB* [2], [10]. Други блок обухвата клијентску страну или *Front-end* који садржи *React.js* [3] кориснички интерфејс, који је такође базиран на *JS* програмском језику, али и на *HTML* и *CSS* језицима за креирање и дизајнирање *web* страница. Поред тога на “фронту” се користи *Local Storage* за складиштење података у претраживачу, те *Redux* [4] који контролише стања чувана у локалној бази података и *PayPal* као *third party* платни систем.



Слика 1. Блок шема целокупног система

Рад се састоји од следећих поглавља:

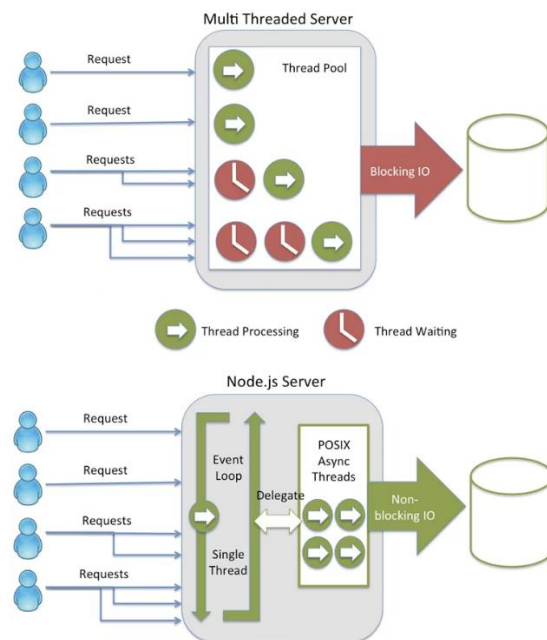
- Првог, уводног поглавља
- Другог поглавља, где су описане теоријске основе
- Трећег поглавља, где су детаљније описани блокови система
- Четвртог поглавља, у оквиру којег је урађен осврт на неке од кључних функционалности *web* апликације
- Петог поглавља, у којем су изнети закључак рада и могућа унапређења

## 2. Теоријске основе

У овом поглављу наведени су кључни аспекти и модули система и описани са теоријским освртом.

### 2.1 *Node.js*

*Node.js* представља *JS* платформу базирану на *Google V8 Engine*-у, која пружа једноставан и брз модел програмирања, а уз то подржава и асинхрони рад *web* апликације. Захтеве обрађује у само једном процесу, без потребе за креирањем додатне нити (енг. *Thread*) услед нових захтева. На следећој илустрацији (слика 2.) упоређено је понашање стандардног вишенитног сервера са *Node.js* приликом обраде нових захтева. Може се уочити да први приступ изискује чекање за извршење захтева, за разлику од *Node*-а, чији програм не чека да се захтев изврши већ апликација наставља даље са извршавањем наредне линије кода. Захваљујући томе омогућен је бржи процес.

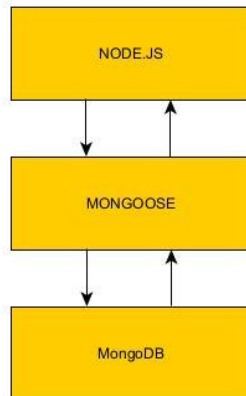


Слика 2. Поређење Multi-threading vs. *Node.js* сервера [5]

Уз *Node* долази и *Node Package Manager* - *npm*. У питању је алат који је задужен за добављање екстерних *Node.js* пакета, као и за складиштење и проналажење истих. Инсталација нових пакета обавља се помоћу *npm* команди командне линије (терминала), чиме се пакети аутоматски преузимају и додају у листу зависности (енг. *Dependency*) у датотеку *package* са *JSON* екстензијом.

## 2.2 MongoDB

*Node* поседује пакет (библиотеку) која омогућава приступање бази и извршавање различитих упита. Реч је о пакету који се назива *Mongoose* и пакету који поседује веома једноставан интерфејс за моделирање података објеката (енг. *Object Data Modeling* - *ODM*). Користи се као спрега између објеката у самом коду и њихову репрезентацију у бази.



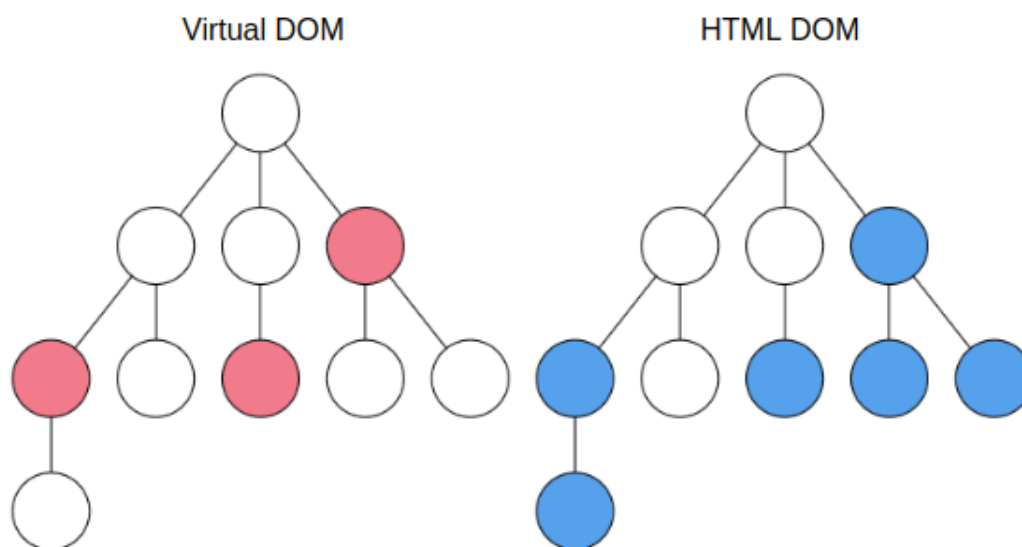
Слика 3. *Mongoose* као спрега између *MongoDB* и *Node.js*

*MongoDB* представља нерелациону врсту базе података или *NoSQL* [11], односно документациону базу, у оквиру које податке је могуће складиштити као *JSON* документе. Сваки документ у својој структури као обавезно поље мора садржати *\_id* као идентификатор по коме се разликује од осталих докумената. Уколико се не унесе, аутоматски ће бити додељен. Структура докумената може да варира, услед чега је смањена сложеност примене, а самим тим и бржи развој саме апликације.

## 2.3 React

Ова платформа функционише тако што се за сваки *HTML* елемент креира специјални *JS* објекат уз чију се помоћ врши контрола над *DOM* моделом сајта. Објекти могу бити елементи или компоненте (сложени елементи) и сваки од њих може укључити неке друге чиме се образује структура “стабла” у којем сваки чвор представља један објекат. Тако се креира виртуални *DOM* - идентична копија реалног модела, чија је улога упоређивање и ажурирање истог, који се користи за рендеровање странице. Уколико би дошло до промене стања неког од објеката виртуални *DOM* ће заправо ажурирати само њега, док остале неће дирати, за разлику од реалног модела који би ажурирао цело стабло чиме представља далеко спорију варијанту.

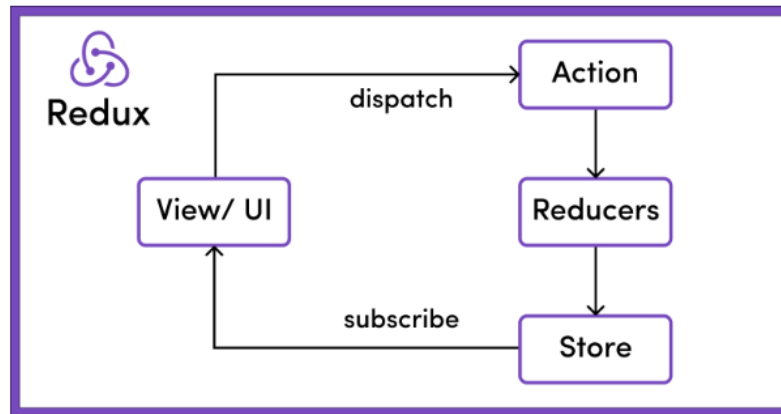
На следећој илустрацији (Слика 4.) приказан је процес интеракције између виртуалног и реалног модела. Виртуални *DOM*, као копија реалног, врло једноставно може да одреди локацију објекта који је потребно ажурирати. Након промене стања неког објекта виртуални модел се ажурира, затим упоређује своје претходно стање како би видео где се промена десила и након тога ажурира само те објекте у *HTML DOM*-у и све остале објекте који су му подређени.



Слика 4. Приказ виртуалног *DOM*-а приликом промене објекта и приказ рендеровања истих у реалном *DOM*-у [6]

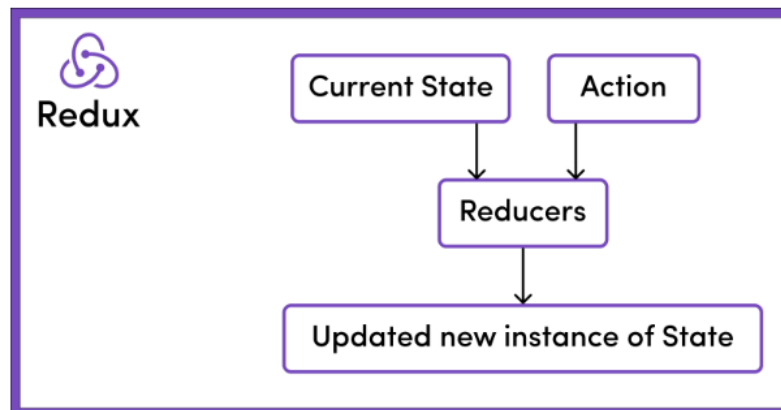
## 2.4 Redux

*Redux* је библиотека која се користи за управљање стањима. Стања из *Redux*-а могуће је само читати (енг. *Read-only*), док се за њихову промену и манипулацију користе акције и редуктори (енг. *Reducer*). Акција описује жељене измене, док редуктори те измене извршавају.



Слика 5. Релације између Redux компоненти [7]

Задатак редуктора је да прихвати акцију и тренутно стање апликације, па потом врати ново ажурирано стање. Редукторске методе се не позивају директно него уз помоћ специјалне функције *dispatch* из *Redux* библиотеке. *Dispatch* потом активира све редукторе, али се извршавају само они који се поклапају са акцијом која им је прослеђена. Потребно је имати на уму да је стање *read-only*, јер редуктори прво праве копију целог тренутног стања, изврше измене и враћају нову инстанцу стања.



Слика 6. Ток ажурирања стања на Redux-у [7]

### 3. Систем

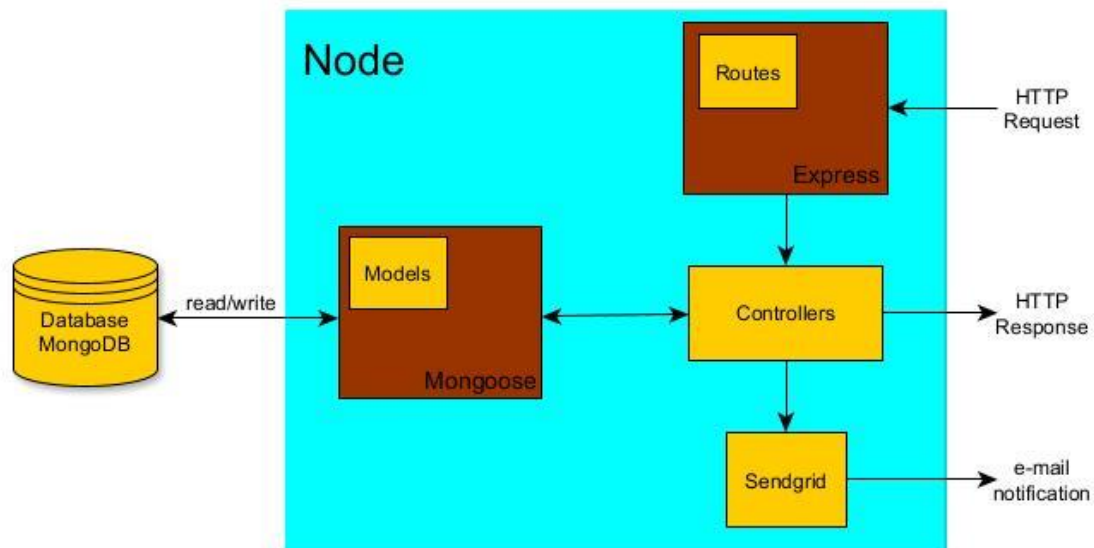
У овом поглављу детаљно су описани блокови система и релације између њих.

#### 3.1 Серверска страна

Као што је већ споменуто у уводном делу, у имплементацији система коришћен је *JS* програмски језик, како на *Front-end*, тако и на *Back-end* делу. *JS* је зависан програмски језик, који се раније могао користити искључиво на клијентској страни, тј. на претраживачу (енг. *Browser*). Међутим, данас овај програмски језик могуће је користити и на серверској страни захваљујући *Node.js* платформи.

Пакети који су коришћени на *back-end*-у су:

- *Mongoose* - за рад са *MongoDB*
- *Dotenv* - за учитавање променљивих из *env* датотеке у процесу
- *Express* [12] - за рутирање сервера и обраду *HTTP* захтева
- *BCrypts* - за енкриптовање шифре
- *JSONWebToken* [13] - за генерисање токена
- *Path* - за добијање *jpg* и *png*
- *Multer* [14] - за учитавање датотека
- *Morgan* - за приказ *HTTP* *zahteva* ради анализе
- *Sendgrid* - за слање *e-mail* порука ка корисницима



Слика 7. Блок шема серверске стране

Сервер је сачињен од следећих ентитета:

- Базе података
- Модела
- Рута
- Контролера и
- *Sendgrid* - *3PP e-mail* нотификатора

### 3.1.1 База података

Као база података коришћен је *MongoDB* који представља нерелациону врсту базе података или *NoSQL*. С обзиром да је ово база без шеме могу се складиштити *JSON* документи у њој, где структура ових докумената може да варира, што представља предност у односу на *SQL* јер убрзава развој апликације и смањује сложеност примене. Како би се могло руковати подацима у бази користе се следеће операције:

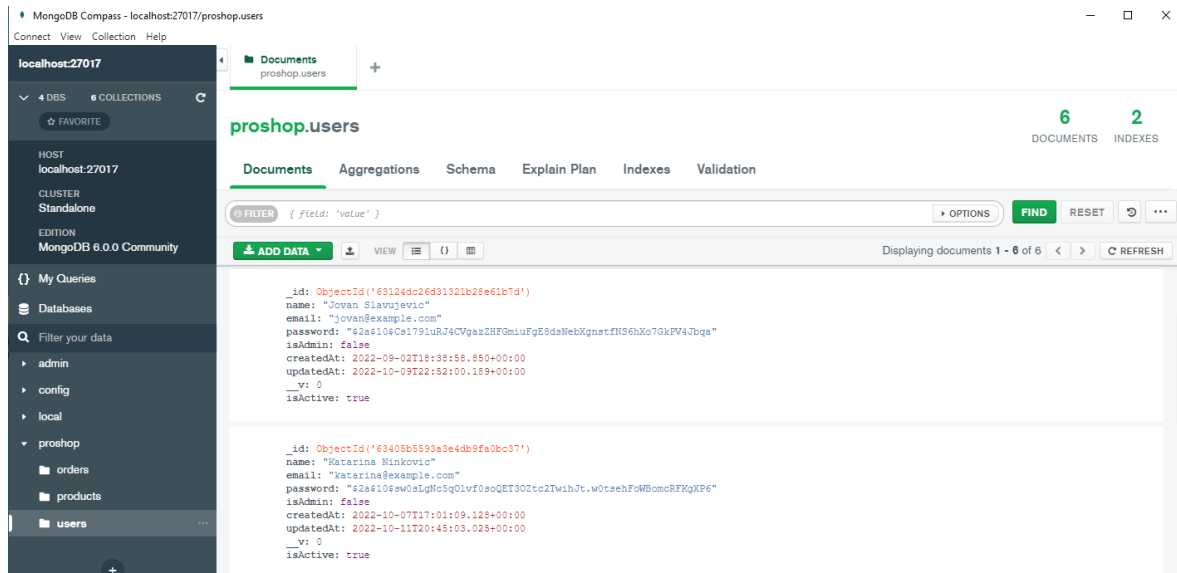
- *insertOne* - за појединачно додавање, где се као параметар прослеђује објекат који садржи потребне податке
- *insertMany* - за групни унос података у базу, где параметар представља низ објеката
- *find* - за претрагу свих података у бази без филтрирања. У случају филтрирања потребно је као параметар убацити жељени *JSON* објекат
- *findById* - за претрагу по *ID*
- *updateOne* - за појединачно ажурирање, где се као први параметар прослеђују поља по којима се објекат тражи, док други параметар представља поља која је потребно ажурирати у пронађеном објекту
- *updateMany* - за групно ажурирање, исти принцип као код *updateOne* с тим што се неће зауставити само на првом пронађеном објекту
- *replaceOne* - замена старих података новим на истој позицији у бази
- *deleteOne* - за појединачно брисање, где се као параметар прослеђује објекат који желимо да избришемо из базе
- *deleteMany* - за групно брисање, сви подаци који се налазе у бази ће бити обрисани

У почетној фази развоја, како би се избегла ситуација увлачења сваког производа и корисника посебно, креиран је групни унос који је писан у *JSON* форми. За коришћење тих података креиране су скрипте које се позивају директно у командој линији помоћу *npm*. Креиране су *import* за унос и *destroy* за брисање из базе. Приликом учитавања у базу прво је неопходно обрисати све до сада постојеће податке, што се постиже функцијом *deleteMany* над одговарајућим моделима. Након што су подаци обрисани могуће је учитати нове помоћу *insertMany* функције где се за параметар прослеђују подаци које је потребно учитати у бази.

Даље руковођење базом се врши помоћу одговарајућих метода, рута и функција у контролерима. Ако би се узео пример креирања производа, први корак био би дефинисање потребних упита у контролерима, где је потребно проверити да ли такав производ постоји у бази, што се постиже помоћу функције за претрагу, која је у овом случају *findById*. Прослеђује се само *ID* производа који је јединствен. Уколико се утврди да производ не постоји може се ићи даље на његово чување у бази, помоћу методе *save* над тим објектом. Након што је описана сва функционалност у контролерима, потребно је креирати *POST* захтев ка бази, креирањем одређене руте у којој се позива одговарајућу контролер.



У фази развоја за праћење и контролисање података коришћен је алат по имену *MongoDB Compass*. Наиме, овај алат се повезује са базом, на начин на који то ради и сервер, те је потребно проследити *IP* адресу, порт и име базе. Када *MongoDB Compass* успешно приступи бази, корисник добија графички приказ података, где је могуће ручно модификовати базу, ако за тим постоји потреба.



Слика 8. Приступање бази помоћу *MongoDB Compass*

### 3.1.2 Модели

Модели представљају ентитет у оквиру серверског блока који је задужен за дефинисање (моделирање) типова података који ће се складиштити. Модели су објекти који представљају *Mongoose* шеме, у оквиру којих је потребно навести све атрибуте које ће један модел поседовати. Атрибути поседују унапред дефинисан тип, али могу поседовати и индикаторе као што су: индикатор да ли је атрибут обавезан или опцион, да ли има и која му је уобичајена (енг. *default*) вредност, да ли се референцира на неки други модел итд. Након што су успешно дефинисани, могу се експортирати како би се моделима руководило у оквиру пројекта.

За потребе овог пројекта дефинисана су три типа *MongoDB* модела. Реч је о уопштеним моделима који су неопходни за рад интернет продавнице и то:

- *Product* је модел који је предвиђен за пласирање артикала које интернет продавница нуди. Састоји се од елемената као што су: име, опис, цена, количина (колико артикала има на стању) и друга потребна поља.
- *User* је модел уз чију помоћ се врши евиденција корисника. Састоји се од класичних поља као што су корисничко име, *e-mail* адреса и лозинка.
- *Order* је модел који је задужен за колекцију реализованих поруџбина. Састоји се од листе наручених производа, адресе доставе, начина плаћања, резултата плаћања, укупне цене и друго.

### 3.1.3 Руте

Иако се *Node.js* користи за креирање апликације на серверској страни, он не укључује руковођење *HTTP* захтевима са клијентске стране, па се у том случају користе *Node*-ове уграђене библиотеке попут *Express* пакета. *Express* пружа комуникација између сервера и базе података, где серверска страна тражи од модела извршавање операција од базе. Коришћене су *GET*, *POST*, *PUT* и *DELETE* методе.

Након што сервер пошаље захтев или добије исти, потребно је проследити одговор. Да би било лакше разазнати врсте захтева и упита, креирају се путање или руте (енг. *Routes*) које су креиране за одређену крајњу тачку (енг. *Endpoint*). Изворна (енг. *Root*) тачка у фази развоја јесте <http://localhost:5000/api>. Оваква путања могућа је само за машину у локалној мрежи, док би за остале уређаје у мрежи на рути уместо *localhost* кључне речи била прослеђена конкретна *IP* адреса уређаја на којем се сервер заправо извршава. У продукцији би крајња тачка била назив *web* странице, која би била *host*-ована. Руте служе за прослеђивање захтева одговарајућим контролерима, који даље обављају сву потребну функционалност.

Табела 1. Постојеће руте и одговарајући *HTTP* захтеви за производе

Руте	HTTP Методе	REQ	RES	Пристап	Опис
<a href="http://localhost:5000/api/products">http://localhost:5000/api/products</a>	GET		products	јавни	Преузимање свих производа
	POST	name, price, image, brand, category, countInStock, description	name, price, user, image, brand, category, countInStock, description	приватни, админ	Креирање производа
<a href="http://localhost:5000/api/products/:id">http://localhost:5000/api/products/:id</a>	GET	id	product	јавни	Преузимање једног производа
	DELETE	id	message	приватни, админ	Брисање једног производа
	PUT	name, price, image, brand, category, countInStock, description	new (name, price, image, brand, category, countInStock, description)	приватни, админ	Ажурирање једног производа
<a href="http://localhost:5000/api/products/:id/reviews">http://localhost:5000/api/products/:id/reviews</a>	POST	id, name, rating, comment	message	приватни	Креирање нове рецензије
<a href="http://localhost:5000/api/products/top">http://localhost:5000/api/products/top</a>	GET		products	јавни	Производи са најбољом рецензијом

Табела 2. Постојеће руте и одговарајући HTTP захтеви за кориснике

Руте	HTTP Метода	REQ	RES	Пристап	Опис
<a href="http://localhost:5000/api/users">http://localhost:5000/api/users</a>	GET		users	приватни, админ	Преузимање свих корисника
<a href="http://localhost:5000/api/users/pre-register">http://localhost:5000/api/users/pre-register</a>	POST	name, email, password	message	јавни	Активација корисничког налога
<a href="http://localhost:5000/api/users/register">http://localhost:5000/api/users/register</a>	POST	token	message	јавни	Регистрација новог корисника
<a href="http://localhost:5000/api/users/login">http://localhost:5000/api/users/login</a>	POST	email, password	id, name, email, token (user)	јавни	Ауторизација корисника и добијање токена
<a href="http://localhost:5000/api/users/forgot-password">http://localhost:5000/api/users/forgot-password</a>	POST	email	message	јавни	Заборављена лозинка
<a href="http://localhost:5000/api/users/reset-password">http://localhost:5000/api/users/reset-password</a>	PUT	token, password	message	јавни	Ресетовање лозинке
<a href="http://localhost:5000/api/users/profile">http://localhost:5000/api/users/profile</a>	GET	id	id, name, email (user)	приватни	Преузимање корисничког профила
	PUT	id, name, email, password	id, name, email, token (user)	приватни	Ажурирање корисничког профила
<a href="http://localhost:5000/api/users/:id">http://localhost:5000/api/users/:id</a>	GET	id	user	приватни, админ	Преузимање једног корисника
	PUT	id, name, email, isAdmin	_id, name, email, isAdmin	приватни, админ	Ажурирање корисничког профила
<a href="http://localhost:5000/api/users/:id/deactivate">http://localhost:5000/api/users/:id/deactivate</a>	POST	id	message	приватни, админ	Деактивирање корисничког налога
<a href="http://localhost:5000/api/users/:id/re-activate">http://localhost:5000/api/users/:id/re-activate</a>	POST	id	message	приватни, админ	Поновно активирање корисничког налога

Табела 3. Постојеће руте и одговарајући HTTP захтеви за поруџбине

Руте	HTTP Метод	REQ	RES	Пристап	Опис
<a href="http://localhost:5000/api/orders">http://localhost:5000/api/orders</a>	GET		orders	приватни, админ	Преузимање свих поруџбина
	POST	orderItems, shippingAddress, paymentMethod, itemsPrice, taxPrice, shippingPrice, totalPrice	orderItems, user.id shippingAddress, paymentMethod, itemsPrice, taxPrice, shippingPrice, totalPrice (order)	приватни	Креирање нове поруџбине
<a href="http://localhost:5000/api/orders/id">http://localhost:5000/api/orders/id</a>	GET	id	order	приватни	Преузимање једне од поруџбина
	DELETE	id, user, isAdmin	message	приватни	Брисање једно од поруџбина
<a href="http://localhost:5000/api/orders/id/pay">http://localhost:5000/api/orders/id/pay</a>	PUT	user id id, status, update_time, email from PayPal response	order	privatni	Ажурирање поруџбине након плаћања
<a href="http://localhost:5000/api/orders/id/deliver">http://localhost:5000/api/orders/id/deliver</a>	PUT	id, user.id	order	приватни, админ	Ажурирање поруџбине након доставе
<a href="http://localhost:5000/api/orders/myorders">http://localhost:5000/api/orders/myorders</a>	GET	id	orders	приватни	Преглед свих поруџбина на личном профилу

### 3.1.4 Контролери

Контролери функционишу тако што преко модела шаљу захтеве ка бази, креирају одређене упите у зависности од тога за шта су им потребни ти подаци и уколико испуњавају услове онда као одговор добијају жељене податке.

Уколико би се узео пример пријаве (енг. *Sign in*) корисника на *web* страници, први корак би био унос података (*e-mail* адреса и лозинка) на клијентској страни, након чега следи паковање унетих података у *HTTP* захтев ка рути на серверској страни <http://localhost:5000/api/users/login> који сервер преузима и потом прослеђује одговарајућем контролеру. Контролер даље за улогу има проверу да ли је корисник евидентиран у бази тако што шаље захтев ка бази, тражећи да прочита податке и изврши одређена испитивања као што је у овом случају постојање и поклапање унете *e-mail* адресе и шифре. Уколико корисник постоји као одговор од базе добијамо податке везане за тог корисника.

На следећим табелама су приказани сви имплементирани контролери.

Табела 4. Постојећи контролери за управљање производима

Контролери	Опис
getProducts	Преузимање свих производа
getProductById	Преузимање једног производа
createProduct	Креирање новог производа
updateProduct	Ажурирање једног од производа
createProductReview	Креирање нове рецензије
getTopProducts	Приказ производа са најбољом оценом

Табела 5. Постојећи контролери за управљање поруџбинама

Контролери	Опис
createOrder	Креирање нове поруџбине
deleteOrder	Брисање једне од поруџбина
getOrderById	Приказ једне од поруџбина
updateOrderToPaid	Ажурирање поруџбине након плаћања
updateOrderToDelivered	Ажурирање поруџбине након доставе
getMyOrders	Приказ поруџбина на личном профилу
getOrders	Приказ свих поруџбина

Табела 6. Постојећи контролери за управљање корисницима

Контролери	Опис
authUser	Ауторизација корисника и добијање токена
forgotPassword	Уколико је лозинка заборављена
resetPassword	Ресетовање лозинке
preRegisterUser	Активација корисничког налога
registerUser	Регистрација новог корисника
getUserProfile	Приказ корисничког профила
updateUserProfile	Ажурирање корисничког профила
getUsers	Приказ свих корисника
getUserById	Приказ једног корисника
updateUser	Ажурирање једног од корисника
deactivateUser	Деактивирање корисничког налога
reActivateUser	Поновно активирање корисничког налога

### 3.1.5 Sendgrid

У питању је екстерни (*third-party*) систем за реализацију слања *e-mail* порука ка корисницима од стране сервера. Ова платформа поседује подршку за *Node*, те се може интегрисати и користити као остали пакети. Неопходно је креирати *developer* налог и *API* кључ како би слање порука било могуће. *Sendgrid* је једна од платформи која изузетно брине о безбедности, те је потребно испоштовати неке од њихових процедура и гарантовати доставу сигурног садржаја, а њихова техничка подршка редовно надгледа рад налога и сигурност *API* кључа. Поред примаоца, у заглављу поруке, потребно је проследити *e-mail* адресу пошиљаоца, коју је претходно потребно верификовати.

У оквиру овог пројекта *Sendgrid* функционалност се користила приликом активације корисничког налога (један од корака приликом регистрације корисника), као и за слање битних нотификација као што су стање поруџбина корисника.



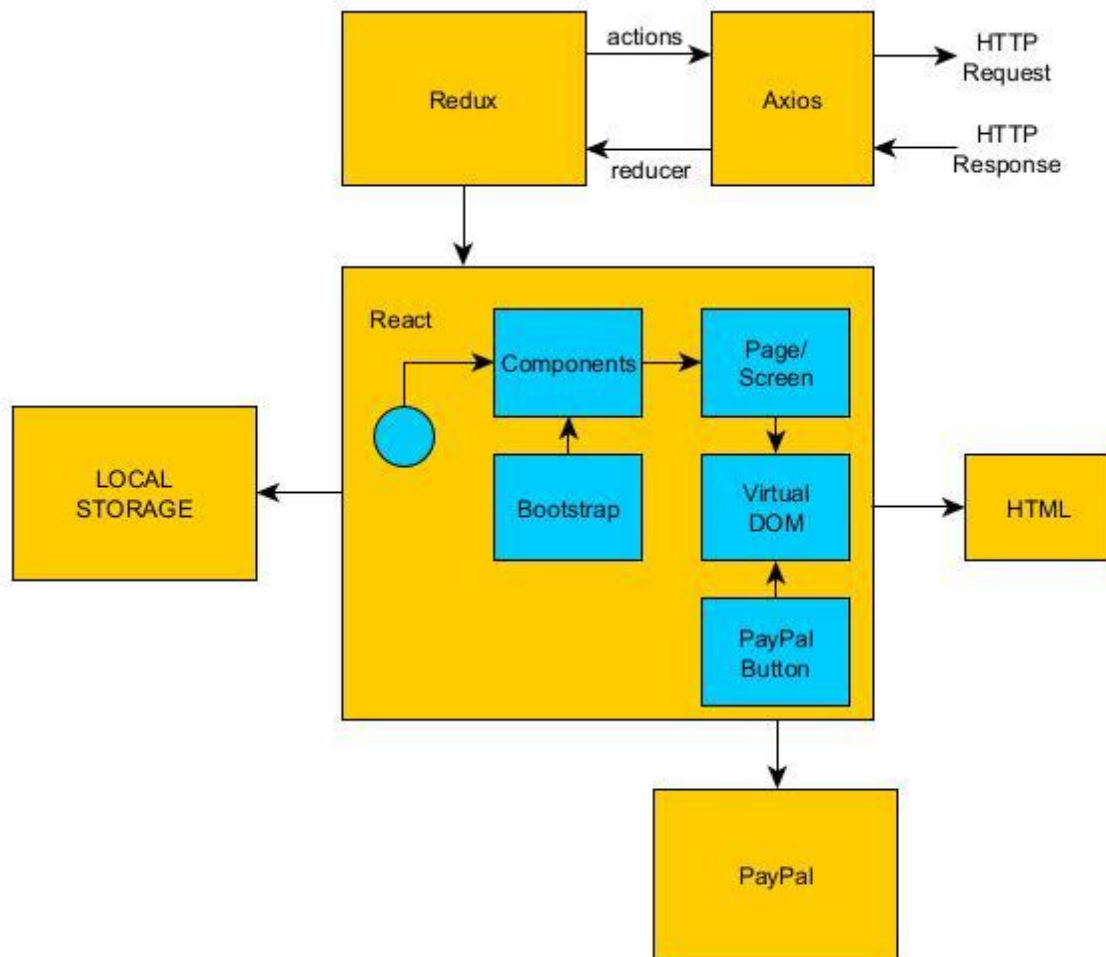
Слика 9. SendGrid лого [8]



## 3.2 Клијентска страна

За развој корисничког интерфејса коришћена је *React* библиотека са својом *ES6 JS* верзијом. Предности *React*-а су олакшано креирање динамичких рендерованих *web* апликација и могућности креирања виртуалног *DOM* (*Document Object Model*) захваљујући којем су побољшане перформансе и брже креирање апликације. Као и на *back-end* страни, за креирање пројекта и инсталацију пакета треће стране користи се *npm*. Кључни ентитети на *front-end* делу су:

- *React* - *JS front-end framework*
- *Axios* [15] - за слање *HTTP* захтева
- *Redux* - за контролу локалне базе података
- *Bootstrap* - за изглед корисничког интерфејса
- *PayPal* - систем за плаћање



Слика 10. Блок шема клијентске стране

### 3.2.1 Axios и Redux

*Axios* представља једно од својстава које се користи уз *React*. Користи се за руковођење *HTTP* захтевима на клијентској страни, као што то ради *Express* на серверској. *Axios* пружа комуникацију између сервера и клијента помоћу *GET*, *POST*, *PUT* и *DELETE* метода.

Помоћу *Axios*-а је подржано и асинхроно понашање које је засновано на обећањима (енг. *Promise*). Како би се обећања написала једноставније са што мање кода, избегавајући писање свих обећања посебно користе се *async* и *await* кључне речи.

*Axios* и *Redux* су у тесној релацији, пре свега јер се све *Axios* методе позивају преко *Redux*-а, односно преко његових редуктора.

Уколико би се узео пример ажурирања профила (енг. *Update Profile*), као први корак било би креирање акција за опис измена и потребног редуктора (енг. *Reducer*) који ће те измене извршити. У акцијама се помоћу *Axios*-а креира захтев за ажурирање (*PUT*) који као први аргумент садржи путању до странице на којој је потребно извршити промену. Други аргумент садржи податке о кориснику (*user*), док је трећи објекат *config* који у себи садржи место у бази где је потребно извршити измену и токен који је јединствен за сваког корисника и који се добија помоћу функције *getState*. Након што су све потребне измене дефинисане у акцији потребно их је извршити. Тада на сцену долази редуктор који прихвата ту акцију и тренутно стање апликације и након тога враћа ново ажурирано стање.

Табела 7. Листа редуктора и модела стања за продукте

Парче	Редуктори	Модел стања	Опис
Cart	cartAddItem, cartRemoveItem, cartSaveShippingAddress, cartSavePaymentMethod, cartReset	cartItems, shippingAddress, paymentMethod	Руковођење корпом
Product	productListRequest, productListSuccess, productListFail	products, pages, page, loading, error	Руковођење продуктима
ProductDetails	productDetailsRequest, productDetailsSuccess, productDetailsFail	product, loading, error	
ProductDelete	productDeleteRequest, productDeleteSuccess, productDeleteFail	loading, succes error	
ProductManage	productManageRequest, productManageSuccess, productManageFail, productManageReset	product, loading, succes, error	

ReviewCreate	productCreateReviewRequest, productCreateReviewSuccess, productCreateReviewFail, productCreateReviewReset	loading, succes, error	
TopRated	productTopRequest, productTopSuccess, productTopFail	products, loading, error	

Табела 8. Листа редуктора и модела стања за кориснике

Парче	Редуктори	Модел стања	Опис
UserLogin	userLoginRequest, userLoginSuccess, userLoginFail, userLoginReset, userLogout	userInfo, loading, error	Руковођење корисничким подацима
ForgotPassword	userForgotPasswordRequest, userForgotPasswordSuccess, userForgotPasswordFail, userForgotPasswordReset	loading, success, error	
UserRegister	userRegisterRequest, userRegisterSuccess, userRegisterFail, userRegisterReset	loading, success, error	
UserDetails	userDetailsRequest, userDetailsSuccess, userDetailsFail, userDetailsReset	user, loading, error	
UpdateProfile	userUpdateProfileRequest, userUpdateProfileSuccess, userUpdateProfileFail, userUpdateProfileReset	userInfo, loading, error, success	
UserList	userListRequest, userListSuccess, userListFail, userListReset	users, loading, error, success	
UserManipulate	userManipulateRequest, userManipulateSuccess, userManipulateFail, userManipulateReset	loading, error, success	
UserUpdate	userUpdateRequest, userUpdateSuccess, userUpdateFail, userUpdateReset	loading, error, success	

Табела 9. Листа редуктора и модела стања за поруџбине

Парче	Редуктори	Модел стања	Опис
OrderCreate	orderCreateRequest, orderCreateSuccess, orderCreateFail, orderCreateReset	loading, error	Руковођење поруџбинама
OrderDelete	orderDeleteRequest, orderDeleteSuccess, orderDeleteFail	loading, success, error	
OrderDetails	orderDetailsRequest, orderDetailsSuccess, orderDetailsFail, orderDetailsReset	order, loading, error	
OrderPay	orderPayRequest, orderPaySuccess, orderPayFail, orderPayReset	loading, success, error	
OrderDeliver	orderDeliverRequest, orderDeliverSuccess, orderDeliverFail, orderDeliverReset	loading, success, error	
OrderMyList	orderListMyRequest, orderListMySuccess, orderListMyFail, orderListMyReset	orders, loading, error	
OrderList	orderListRequest, orderListSuccess, orderListFail	orders, loading, error	
PayPalId	paypalIdRequest, paypalIdSuccess, paypalIdFail, paypalIdReset	ppId, loading, error	

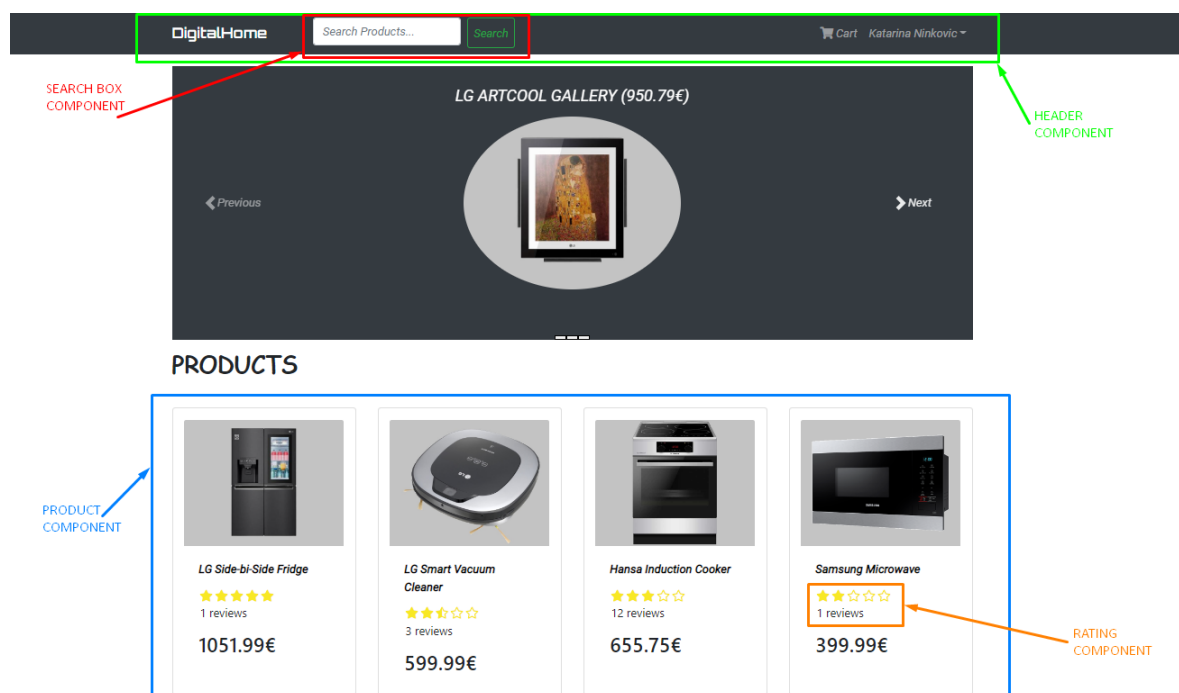
### 3.2.2 Елементи и компоненте

*React* елементи се користе за описивање изгледа *HTML* елемената. За њихов опис користи се *JS* синтаксно проширење *JSX* (енг. *JavaScript Syntax Extension*) који обезбеђује писање *HTML* структуре и *JS* кода у истој датотеци. За управљање тим елементима користе се компоненте.

*React* дели кориснички интерфејс на бројне компоненте које представљају градивне блокове, чиме је отклањање грешака олакшано. Могућности које компоненте обезбеђују су:

- Једна компонента се може користити у више области апликације, чиме убрзава процес развоја
- Компоненте могу садржати и друге компоненте
- Компонента може да прими својства (енг. *Properties*) или *props* коју прослеђује њен родитељ

У оквиру овог пројекта дељива особина *React* компоненти је у великој употреби због постојања више страница које морају укључити исте компоненте. Тиме је, очуван и идентичан стил на свим страницама сајта, али и битна функционалност. Пре свега, ово се односи на навигациони мени или *navbar* који је узет са једног од популарних *Bootstrap* шаблона.



Слика 11. Приказ компонента на почетној страници

Табела 10. Постојеће front-end руте и одговарајући прикази

Руте	Главна компонента	Опис
<a href="http://localhost:3000">http://localhost:3000</a>	HomeScreen	Приказ свих артикала
<a href="http://localhost:3000/register">http://localhost:3000/register</a>	RegisterScreen	Регистрација корисника
<a href="http://localhost:3000/account/activate/:token">http://localhost:3000/account/activate/:token</a>	EmailActivationScreen	Активација корисничког налога
<a href="http://localhost:3000/forgot/password">http://localhost:3000/forgot/password</a>	ForgotPasswordScreen	Заборављена лозинка
<a href="http://localhost:3000/password/reset/:token">http://localhost:3000/password/reset/:token</a>	ResetPasswordScreen	Ресетовање лозинке
<a href="http://localhost:3000/login">http://localhost:3000/login</a>	LoginScreen	Пријављивање корисника
<a href="http://localhost:3000/profile">http://localhost:3000/profile</a>	ProfileScreen	Кориснички прозор
<a href="http://localhost:3000/admin/user/:id/edit">http://localhost:3000/admin/user/:id/edit</a>	UserEditScreen	Ажурирање корисничког налога
<a href="http://localhost:3000/admin/userlist">http://localhost:3000/admin/userlist</a>	UserListScreen	Приказ свих корисника
<a href="http://localhost:3000/product/:id">http://localhost:3000/product/:id</a>	ProductScreen	Прозор специфичног артикла
<a href="http://localhost:3000/admin/product/create">http://localhost:3000/admin/product/create</a>	ProductManageScreen	Ажурирање и креирање нових производа
<a href="http://localhost:3000/admin/productlist/:pageNumber">http://localhost:3000/admin/productlist/:pageNumber</a>	ProductListScreen	Приказ свих поруџбина
<a href="http://localhost:3000/cart/:id?">http://localhost:3000/cart/:id?</a>	CartScreen	Приказ корпе
<a href="http://localhost:3000/shipping">http://localhost:3000/shipping</a>	ShippingScreen	Унос личних података за доставу

<a href="http://localhost:3000/payment">http://localhost:3000/payment</a>	PaymentScreen	Одабир начина плаћања
<a href="http://localhost:3000/placeorder">http://localhost:3000/placeorder</a>	PlaceOrderScreen	Приказ реализованих поруџбина
<a href="http://localhost:3000/order/:id">http://localhost:3000/order/:id</a>	OrderScreen	Плаћање поруџбине
<a href="http://localhost:3000/admin/orderlist">http://localhost:3000/admin/orderlist</a>	OrderListScreen	Приказ свих поруџбина

Табеле приказују побројане руте на *front-end* страни, те њихове главне компоненте (јединствене *DOM* моделе по страницама), као и компоненте које деле. Може се уочити да неке руте на крају имају део формата “:x”, што значи да те руте нису фиксне.

Табела 11. Постојеће компоненте њихова релација са страницама (рутама)

Компонента	Руте	Опис
Header	<a href="http://localhost:3000">http://localhost:3000</a> <a href="http://localhost:3000/register">http://localhost:3000/register</a> <a href="http://localhost:3000/login">http://localhost:3000/login</a> <a href="http://localhost:3000/profile">http://localhost:3000/profile</a> <a href="http://localhost:3000/product/:id">http://localhost:3000/product/:id</a> <a href="http://localhost:3000/cart/:id?">http://localhost:3000/cart/:id?</a>	Заглавље сваке странице, укључује навигациони мени
Footer	<a href="http://localhost:3000/shipping">http://localhost:3000/shipping</a> <a href="http://localhost:3000/payment">http://localhost:3000/payment</a> <a href="http://localhost:3000/placeorder">http://localhost:3000/placeorder</a> <a href="http://localhost:3000/order/:id">http://localhost:3000/order/:id</a>	Подножје сваке старнице, укључује <i>copyright</i> ознаку
Product	<a href="http://localhost:3000">http://localhost:3000</a>	Појединачни артикал
Message	<a href="http://localhost:3000">http://localhost:3000</a> <a href="http://localhost:3000/register">http://localhost:3000/register</a> <a href="http://localhost:3000/login">http://localhost:3000/login</a> <a href="http://localhost:3000/profile">http://localhost:3000/profile</a> <a href="http://localhost:3000/product/:id">http://localhost:3000/product/:id</a> <a href="http://localhost:3000/cart/:id?">http://localhost:3000/cart/:id?</a> <a href="http://localhost:3000/placeorder">http://localhost:3000/placeorder</a> <a href="http://localhost:3000/order/:id">http://localhost:3000/order/:id</a>	Обавештења (успех, грешка или упозорење)
Loader	<a href="http://localhost:3000">http://localhost:3000</a> <a href="http://localhost:3000/register">http://localhost:3000/register</a> <a href="http://localhost:3000/login">http://localhost:3000/login</a> <a href="http://localhost:3000/profile">http://localhost:3000/profile</a> <a href="http://localhost:3000/product/:id">http://localhost:3000/product/:id</a> <a href="http://localhost:3000/order/:id">http://localhost:3000/order/:id</a>	Учитавање или чекање одговора

Rating	<a href="http://localhost:3000/product/:id">http://localhost:3000/product/:id</a>	Просечна оцена артикла
FormContainer	<a href="http://localhost:3000/register">http://localhost:3000/register</a> <a href="http://localhost:3000/login">http://localhost:3000/login</a> <a href="http://localhost:3000/shipping">http://localhost:3000/shipping</a> <a href="http://localhost:3000/payment">http://localhost:3000/payment</a>	Форма за унос података
ChechoutSteps	<a href="http://localhost:3000/shipping">http://localhost:3000/shipping</a> <a href="http://localhost:3000/payment">http://localhost:3000/payment</a> <a href="http://localhost:3000/placeorder">http://localhost:3000/placeorder</a>	Кораци за реализацију куповине
Meta	<a href="http://localhost:3000">http://localhost:3000</a> <a href="http://localhost:3000/product/:id">http://localhost:3000/product/:id</a>	Промена назива странице у зависности који је производ одабран
Paginate	<a href="http://localhost:3000">http://localhost:3000</a> <a href="http://localhost:3000/admin/productlist/:pageNumber">http://localhost:3000/admin/productlist/:pageNumber</a>	Ограничен број производа по страни
ProductCarousel	<a href="http://localhost:3000">http://localhost:3000</a>	Приказ проивода са најбољом оценом
SearchBox	Header	Претрага производа
UserActiveButton	<a href="http://localhost:3000/admin/userlist">http://localhost:3000/admin/userlist</a>	Активација и деактивација корисничког налога



### 3.2.3 Bootstrap

*Bootstrap* је *framework* који је задужен за естетику, тј. за изглед корисничког интерфејса. Представља колекцију *CSS* класа које су задужене за унапред дефинисан и стандардизован изглед *HTML* елемената или, у овом случају, *React* компонената. Поред лепог изгледа, пружа и додатне функционалности кроз уграђени *jQuery*, али једна од најбитнијих одлика јесте респонзивни дизајн који пружа интернет страницама, односно скалиран и контролисан приказ елемената на различитим резолуцијама и величинама екрана. Поседује и *npm* пакет чиме обезбеђује и готове *React* компоненте.

Табела 12. Постојеће коришћене *Bootstrap* компоненте

Bootstrap Компоненте	Компоненте / Екрани	Опис
LinkContainer	ChechoutSteps, Header, Paginate	Интеграција између <i>React Router</i> и <i>React Bootstrap</i>
	OrderListScreen, ProductListScreen, ProfileScreen, UserListScreen	
Container	Footer, FormContainer, Header	Блок који поравнава садржај који је у њему
Nav	ChechoutSteps, Header	За изградњу свих компонената у навигационом менију
Navbar	Header	За навигациони мени
NavDropdown	Header	Падајући мени у оквиру навигационог менија
Row	Footer, FormContainer	За креирање редова
	HomeScreen, CartScreen, LoginScreen, OrderScreen, PlaceOrderScreen, ProductListScreen, ProductScreen, ProfileScreen, RegisterScreen	
Col	Footer, FormContainer	За креирање колона
	HomeScreen, CartScreen, LoginScreen, OrderScreen, PaymentScreen, PlaceOrderScreen, ProductListScreen, ProductScreen, ProfileScreen, RegisterScreen	
Table	OrderListScreen, ProductListScreen, ProfileScreen, UserListScreen	За формирање табела

ListGroup	CartScreen, OrderScreen, PlaceOrderScreen, ProductScreen	За приказивање низа садржаја
Button	SearchBox, UserActiveButton	Приказ дугмета уз подешавање изгледа и старања о догађајима
	CartScreen, ForgotPasswordScreen, ResetPasswordScreen, LoginScreen, OrderListScreen, OrderScreen, PaymentScreen, PlaceOrderScreen, ProductListScreen, ProductManageScreen, ProductScreen, ProfileScreen, RegisterScreen, ShippingScreen, UserEditScreen, UserListScreen	
Form	SearchBox	Креирање форме за попуњавање
	ForgotPasswordScreen, ResetPasswordScreen, LoginScreen, PaymentScreen, ProductManageScreen, ProductScreen, ProfileScreen, RegisterScreen, ShippingScreen, UserEditScreen	
FormControl	CartScreen, ProductScreen	Контрола над текстуалним садржајем (стил, величина...)
Spinner	Loader	Визуелни приказ приликом учитавања
Alert	Message	Обезбеђује враћање свих врста порука
Card	Product	Проширује контејнер са више опција
	CartScreen, OrderScreen, PlaceOrderScreen, ProductScreen	
Image	ProductCarousel	Подешавање слика
	CartScreen, OrderScreen, PlaceOrderScreen, ProductScreen	
Carousel	ProductCarousel	Обезбеђује “вртешку” слика
Pagination	Paginate	Изглед броја страна

### 3.2.4 PayPal дугме

Америчка компанија *Pay Pal* поседује дугу историју пословања у свету, један је од зачетника новчаних трансакција преко интернета, али и незаобилазни платни систем до данас. Уколико нека продавница жели да има велик домаћај (енг. *Reach*) ка корисницима, *Pay Pal* је у том случају *must-have*. Њена интеграција се врши уз помоћ скрипте која се учитава попут других *HTML* елемената приликом учитавања странице, где је потребно проследити претходно креирани *developer*-ски *API* кључ. Учитавањем скрипте омогућено је убацивање неког од њених популарних дугмића који прослеђују корисника на њен прозор. Овим је сва одговорност у вези са сигурношћу трансакција пребачена на страну разрађеног платног система, те је ризик од малверзација сведен на статистичку грешку.

У оквиру овог пројекта коришћен је пакет *react-paypal-js*, односно компоненте *PayPalScriptProvider* и *PayPalButtons*. Прва компонента задужена је за учитавање *PayPal* скрипте, у оквиру које се налази *JS* код задужен за рад дугмета који је потребно интегрисати. Како би се скрипта могла увући потребно јој је проследити претходно креирани *API* кључ, али и додатне опцију као што су валута у којој ће се трансакција обрачунавати. Друга компонента представља само дугме, те је задужена за изглед компоненте на страници, али и за отварање новог прозора и то *PayPal* сајта, у оквиру којег корисник може извршити плаћање. Овој компоненти је потребно проследити методе које ће бити позване приликом догађаја као што су успешна трансакција, где веб продавница одрађује свој део посла када је обавештена о успешној уплати средстава.



Слика 12. PayPal Button

## 4. Функционалности

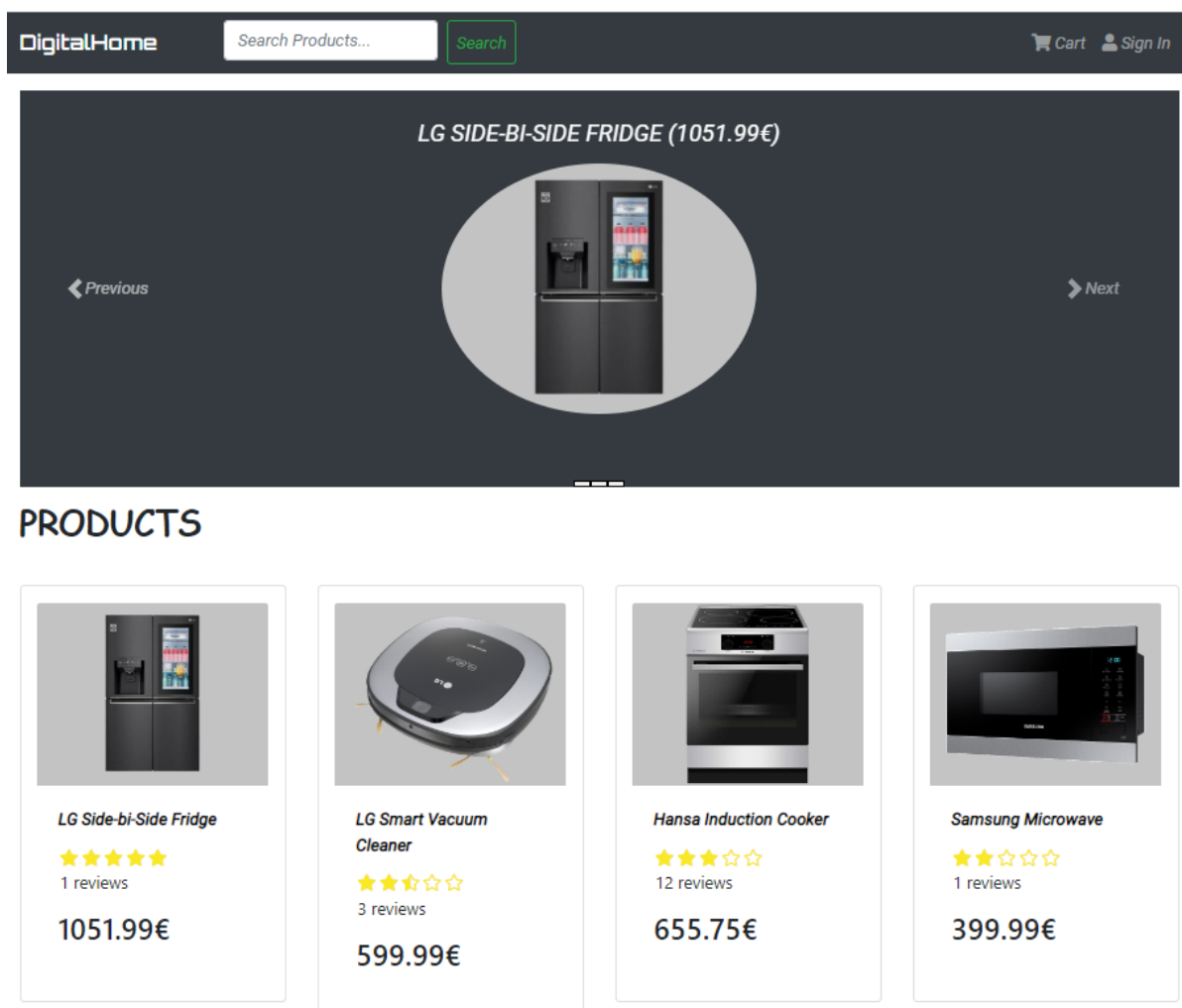
У овом поглављу, урађен је осврт на неке од кључних функционалности *web* апликације.

### 4.1 Кориснички мени

#### 4.1.1 Почетна страна и странице производа

На почетној страници *web* апликације, али и на свим осталим, као заглавље креиран је навигациони мени помоћу *Bootstrap*-а. С обзиром да он чини део сваке странице неопходно је да буде идентичан, што се постиже креирањем компоненте *Header* која се директно позива у *App JS* датотеци.

Поред већ поменутог на почетој страни приказан је и преглед свих производа. Наиме, приказани су основни подаци као што су цена производа, слика, назив и просечна оцена потрошача. Такође, приказани су и производи који имају најбољу оцену по мишљењу корисника.




Слика 13. Приказ почетне стране и листе производа

Како би уопште било могуће видети производе на *web* страници неопходно је преузети производе из базе података. То се постиже креирањем захтева од клијента ка серверу, који упит даље шаље ка бази. С обзиром да је потребно само читање без икаквих измена користи се *GET* метода. Помоћу *dispatch* функције се у специјалној *React* методи *useEffect* позивају потребне редукторске методе, чиме је обезбеђено прикупљање и приказивање производа на почетном екрану. Након што су подаци прикупљени уз помоћ *Redux*-а, односно његових акција и редуктора, може се даље манипулисати са њима.

Као још једна од ставки која је обезбеђена јесте ограничен максималан број производа на једној страни. За овим се појавила потреба како би се избегла ситуација да се на једној страни приказују сви производи, чији број није ограничен, и тиме би се знатно нарушио изглед и прегледност странице. То се обезбеђује креирањем нове компоненте *Paginate* где је дефинисано максимално 10 производа по страници. За изглед, односно приказ страна коришћен је *Pagination* из *Bootstrap*-а. На следећој слици приказано је како то изгледа, где је за пример промењена вредност максималног броја производа по страни на 3 производа.


## PRODUCTS



**LG Side-by-Side Fridge**

★★★★★  
1 reviews


**1051.99€**



**LG Smart Vacuum Cleaner**

★★★★☆  
3 reviews

**599.99€**



**Hansa Induction Cooker**

★★★★☆  
12 reviews

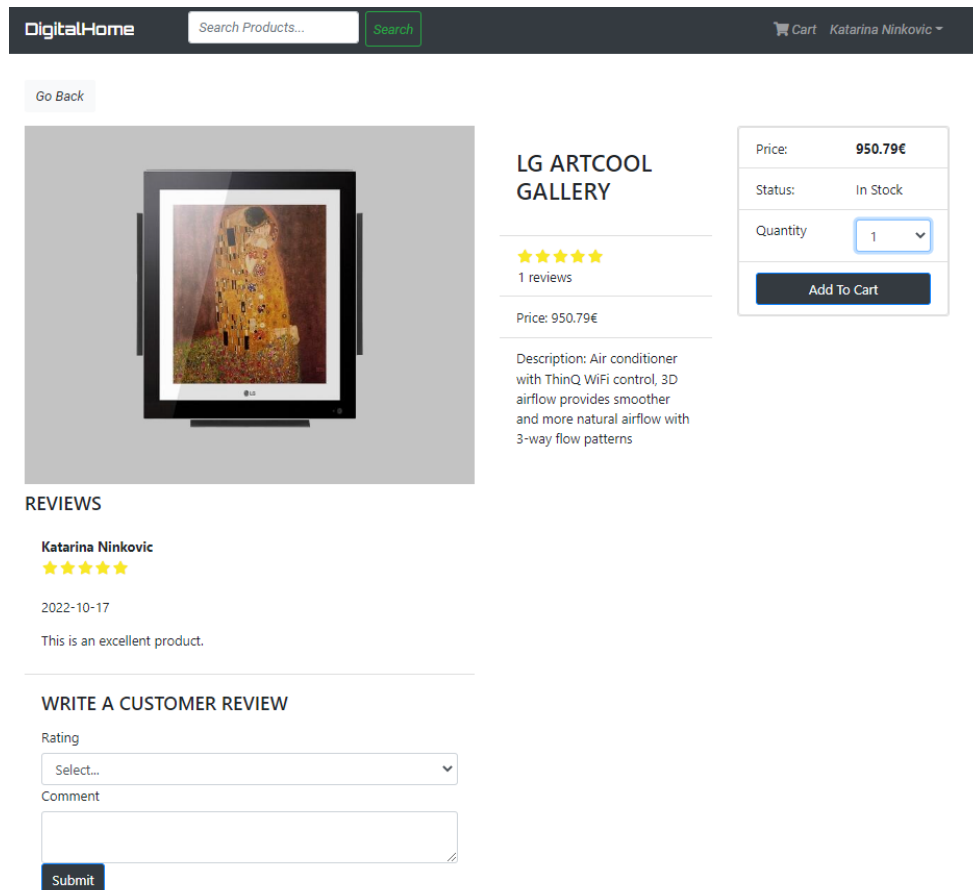
**655.75€**

1(current) 2

Слика 14. Приказ ограничених производа по једној страни

### 4.1.1 Детаљан приказ производа

Кликом на слику или назив производа на почетној страни помоћу *React Link*-а отвара се нова страница која садржи детаљнији приказ тог производа. Како би се знало о ком је производу реч, приликом уласка на страницу креира се путања која у себи садржи његов идентификациони број (*ID*). Након тога сервер купи *ID*, који је јединствен за сваки производ, и проверава да ли такав постоји у бази. Уколико постоји као одговор враћа тај производ, у супротном пријављује грешку.

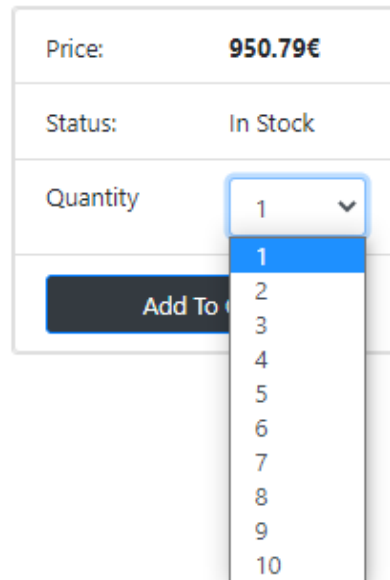


Слика 15. Детаљан приказ једног од производа

Постоји и могућност остављања и прегледа рецензија. Ово корисницима пружа додатно искуство приликом одабира производа, где може непосредно увидети искуства претходних купаца и видети да ли је овај производ стварно за њега. Сваки корисник може једном оставити оцену од 1 до 5 и опционо додати свој коментар, што се потом додаје у базу података и постаје видљиво осталим корисницима. Овим се просечна оцена аутоматски прерачунава за нову оцену. Ова опција омогућена је искључиво за оне кориснике који су пријављени.

Такође, на овој страници обезбеђено је и убацивање жељених производа у корпу, где су даље могућности блокиране све док се корисник не пријави. Сваки производ садржи поље количина (енг. *Quantity*), које купцу говори да ли га има на стању. Уколико га нема поље за додавање у корпу ће бити онеспособљено (енг. *disabled*), а уколико постоји кликом на стрелицу у пољу количина показаше се падајући мени који приказује колико тренутно има тог производа на стању.

Приликом убацивања производа у корпу корисник може произвољно узети онолико производа колико му је потребно, наравно максимално онолико колико тренутно има на стању.

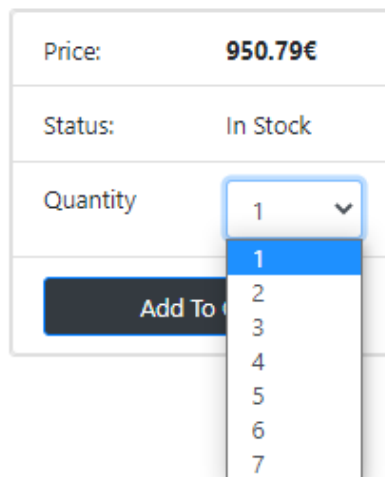


The image shows a product card with the following details:

- Price: 950.79€
- Status: In Stock
- Quantity: A dropdown menu is open, showing a list of numbers from 1 to 10. The number 1 is highlighted in blue.
- Below the quantity dropdown is a dark blue button with the text "Add To".

Слика 16. Приказ падајућег менија у пољу количина

Након креиране поруџбине стање се мења, односно смањује за онолико колико је корисник наручио. На слици 17. приказано је промењено стање након што је корисник наручио три иста производа. Ажурирање стања омогућено је администратору о чему ће више речи бити у наставку.



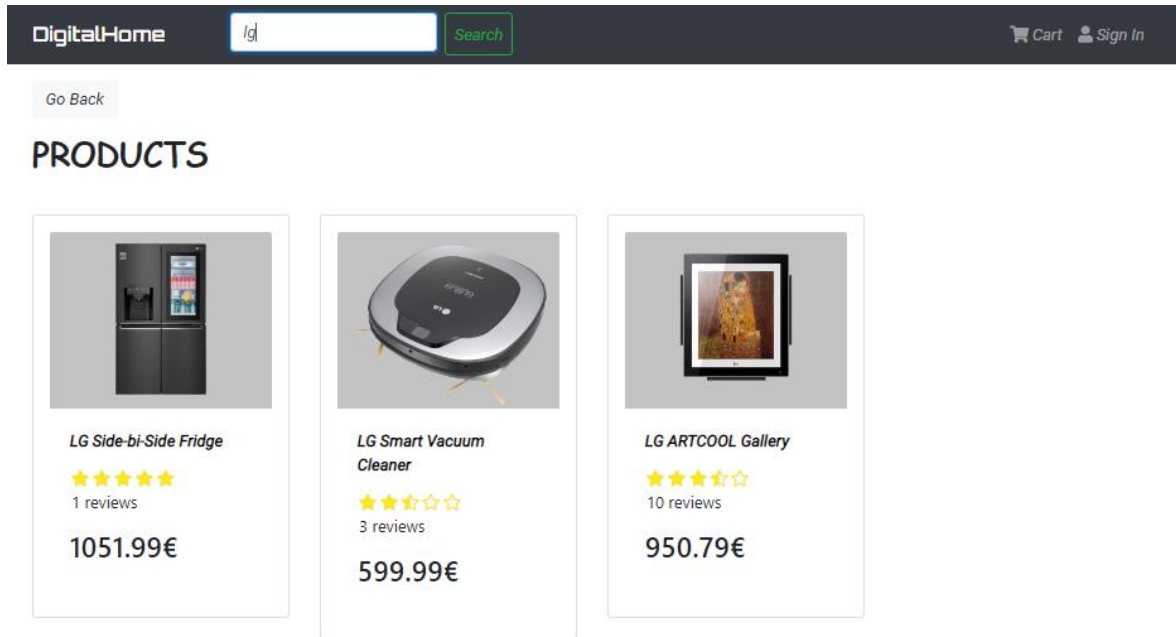
The image shows the same product card as in Slika 16, but with a different quantity dropdown menu:

- Price: 950.79€
- Status: In Stock
- Quantity: A dropdown menu is open, showing a list of numbers from 1 to 7. The number 1 is highlighted in blue.
- Below the quantity dropdown is a dark blue button with the text "Add To".

Слика 17. Промењено стање у падајућем менију након креирања поруџбине

## 4.1.2 Претрага производа

Како би кориснику било омогућено што лакше коришћење апликације, креирана је компонента *SearchBox*, која обезбеђује једноставну претрагу производа. Помоћу регуларних израза (енг. *Regular Expressions*) или *Regex* [9] обезбеђена је претрага где није неопходан унос идентичног назива већ сви производи који садрже тражене речи ће бити исфилтрирани и приказани на новој страници. Такође, обезбеђено је и игнорисање малих и великих слова, чиме је олакшана претрага тражених производа.



Слика 18. Производи након извршене претраге



### 4.1.3 Регистрација и активација налога

Уколико би корисник желео да креира поруџбину неопходно је да буде пријављен. Као нови корисник први корак би била регистрација где би се унели тражени подаци као што су име, *e-mail* адреса и шифра.

DigitalHome Search Products... Search

Cart Sign In

## SIGN UP

Name  
Katarina Ninkovic

Email Address  
kaca.ninkovic.69@gmail.com

Password  
.....

Confirm Password  
.....

Register

Have an Account? [Login](#)

Слика 19. Прозор за регистрацију

Како би се избегла ситуација за креирање истог корисника више пута приликом регистрације неопходно је извршити проверу постојања корисника са унетом *e-mail* адресом. Уколико се деси да се унета *e-mail* адреса поклапа са неком из базе, приликом регистрације као упозорење ће изаћи грешка као што је приказано на слици 20, чиме је регистрација заустављена.

## SIGN UP

User already exists

Name  
Katarina Ninkovic

Email Address  
kaca.ninkovic.69@gmail.com

Password  
.....

Confirm Password  
.....

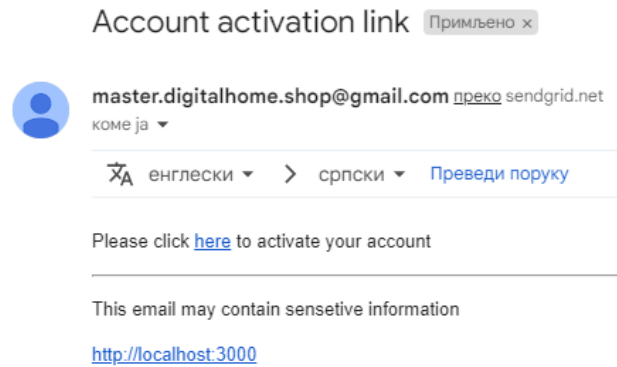
Register

Have an Account? [Login](#)

Слика 20. Грешка приликом регистрације корисника који већ постоји

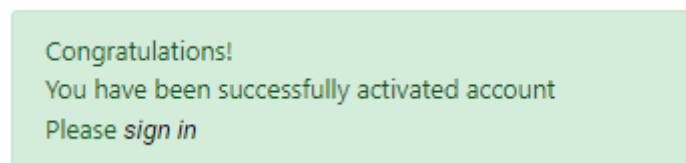
Ако је провером утврђено да корисник ипак не постоји, на *e-mail* адресу стиже порука за активацију корисничког налога преко *Sendgrid*-а. Садржај поруке је написан помоћу *HTML* елемената, међу којима је и линк који представља токен и чијом потврдом,

односно кликом на *here*, се завршава процес активације. Време истека токена је 10 минута након чега више није употребљив. У том случају корисник би морао поново да пошаље захтав за регистрацију.



Слика 21. Приказ e-mail поруке за активацију корисничког налога

Помоћу компоненте *Message* кориснику стиже обавештење да је налог успешно активиран и да може прећи на пријаву.



Слика 22. Порука о успешном активирању корисничког налога

#### 4.1.4 Аутентикација

Како би се заштитили подаци, као што је, на пример, профил корисника креиран је *authMiddleware* за аутентикацију. Заштита је обезбеђена помоћу *JWT* токена, чије се креирање врши уз помоћ *npm* пакета који се назива *JSONWebToken*. Приликом генерисања користи се функција која за први аргумент очекује некакав јединствени *ID*. Пожељно је да то буде податак који није од велике важности, као што би, на пример, био број рачуна, који треба да остане тајна. Као други аргумент ставља се тајна секвенца, односно креира се променљива која је позната власнику (креатору) сервера и стављена је у заштићеној *env* датотеци. За читавање те променљиве из *env* датотеке користи се *Dotenv* пакет. Трећи аргумент је опциони и представља време истека токена.

Како би се добили подаци о кориснику потребно је помоћу *jwt.verify* функције декодовати токен. Као први аргумент налази се токен који је потребно декодовати док је као други тајна комбинација која се чува у *env* датотеци и коју зна само креатор. Након тога као одговор се враћају сви подаци о кориснику изузев шифре.

Уколико се испостави да неко приликом приступања корисничким подацима нема потребан токен појавиће се грешка као на следећој слици.

DigitalHome Search Products... Search Cart Katarina Ninkovic

USER PROFILE MY ORDERS

Not authorized, token failed Not authorized, token failed

Name

Enter name

Email Address

Enter email

Password

Enter password

Confirm Password

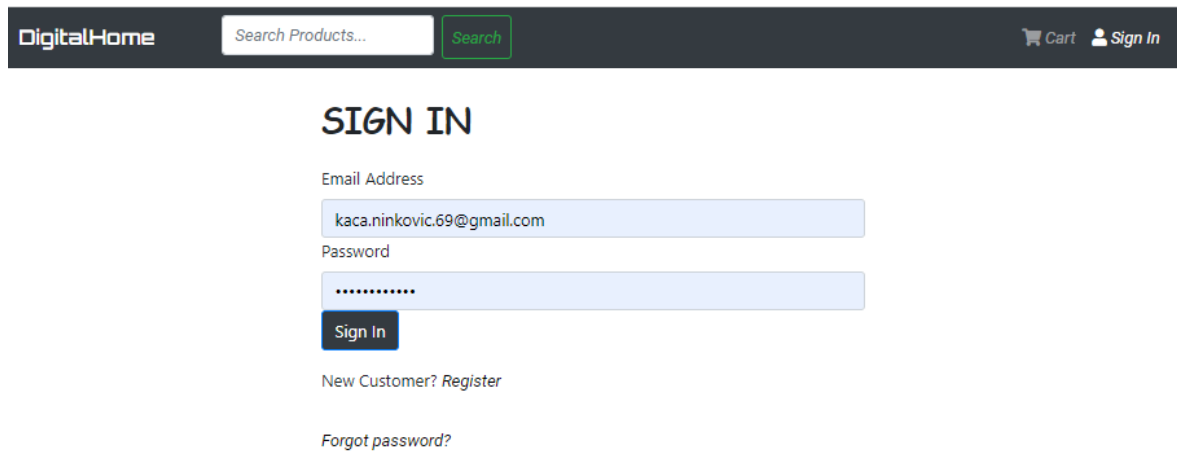
Confirm password

Update

Слика 23. Грешка приликом приступа корисничким подацима без токена

### 4.1.5 Пријава и одјава корисника

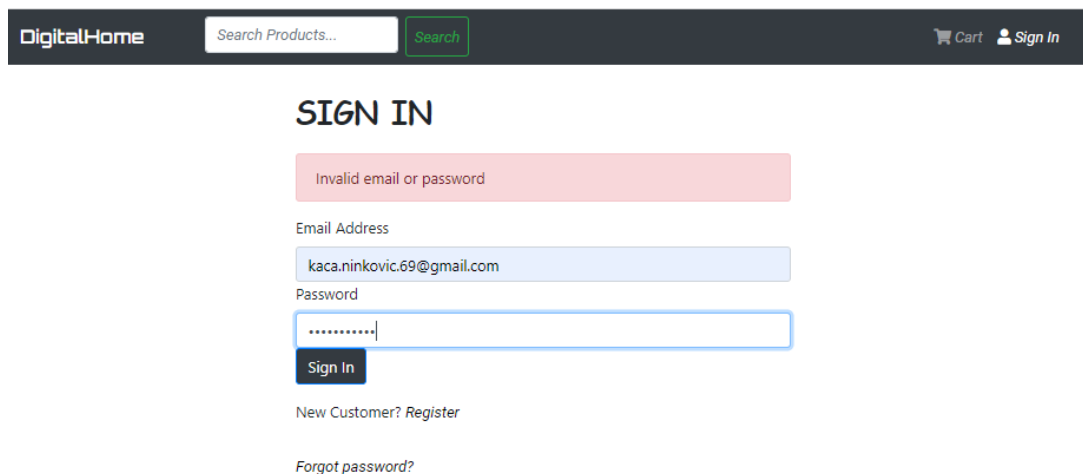
Након што је корисник креирао налог омогућена је пријава. Приликом пријаве долази до провере унетих података са подацима из базе.



The screenshot shows the 'SIGN IN' page of the DigitalHome application. At the top, there is a dark navigation bar with the 'DigitalHome' logo, a search bar with the placeholder 'Search Products...' and a green 'Search' button, and links for 'Cart' and 'Sign In'. The main content area has a light gray background. The 'SIGN IN' title is centered. Below it, the 'Email Address' field contains 'kaca.ninkovic.69@gmail.com' and the 'Password' field is filled with dots. A dark blue 'Sign In' button is positioned below the password field. At the bottom of the form, there are links for 'New Customer? Register' and 'Forgot password?'.

Слика 24. Прозор за пријаву

Уколико се подаци не поклапају долази до приказа грешке, коју је обезбедила компонента *Message*.



This screenshot shows the 'SIGN IN' page after an unsuccessful login attempt. A red error message box at the top of the form area displays the text 'Invalid email or password'. The 'Email Address' field still contains 'kaca.ninkovic.69@gmail.com' and the 'Password' field is empty. The 'Sign In' button and the links for 'New Customer? Register' and 'Forgot password?' are still visible at the bottom.

Слика 25. Грешка приликом погрешног уноса e-mail адресе или лозинке

У супротном корисник се усмерава на почетну страну апликације, те се дешава ажурирање навигационог менија које сада садржи његов профил. Као што је горе већ речено, профил је заштићен помоћу токена. Након пријаве, на *web* претраживачу се чувају подаци о кориснику у *Local Storage*-у докле год је он пријављен. По одјави, подаци о кориснику се бришу као и друга чувана поља.

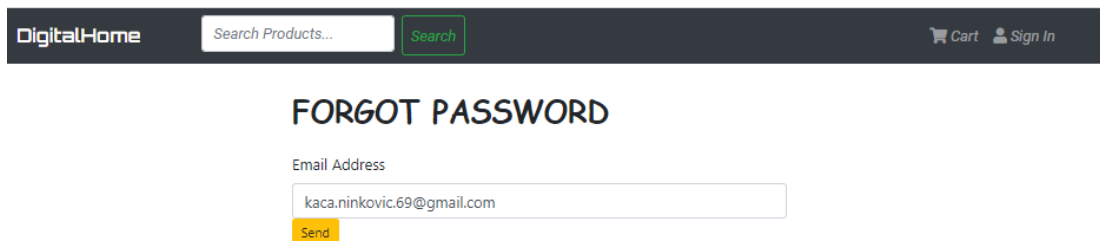


The screenshot shows the top navigation bar of the DigitalHome application after a user has logged in. The 'DigitalHome' logo is on the left. The search bar has the placeholder 'Search Products...' and a green 'Search' button. On the right, there is a 'Cart' icon and the user's name 'Katarina Ninkovic' followed by a dropdown arrow.

Слика 26. Изглед навигационог менија након пријаве корисника

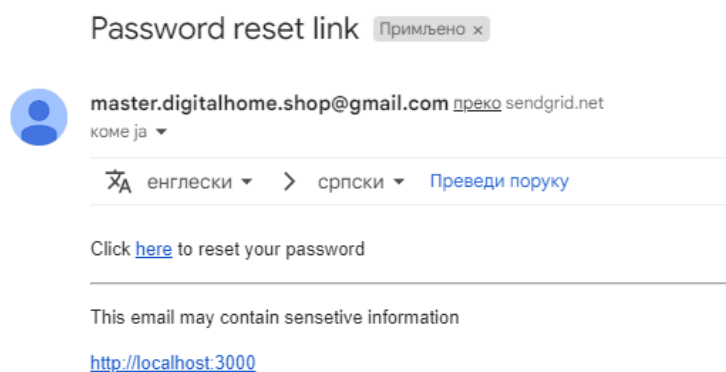
## 4.1.6 Ресетовање шифре

Као честа појава код апликација које користе шифру јесте заборављање исте. Како би се избегла ситуација не могућности приступа свим својим подацима и поруџбинама које су до сада реализоване обезбеђено је ресетовање шифре. Једино што је потребно да корисник уради јесте да унесе *e-mail* адресу коју је поставио приликом регистрације.



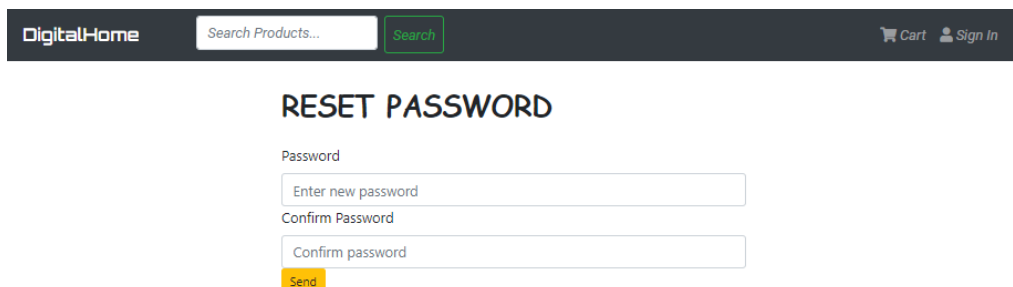
Слика 27. Прозор за слање захтева за ресетовање шифре

Кликом на дугме *send* на *e-mail* адресу стиже порука за ресетовање корисничке шифре преко *Sendgrid*-а.



Слика 28. Приказ *e-mail* поруке за ресетовање шифре

Уласком на линк кликом на хипер текст *here* отвара се нова страница на *web* апликацији на којој је потребно унети нову шифру и поновити исту. Тиме је извршено ресетовање шифре коју ће корисник убудуће користити приликом пријаве.



Слика 29. Прозор за унос нове лозинке

### 4.1.7 Ажурирање профила

Докле год је корисник пријављен може манипулисати подацима које је приложио. Та манипулација представља ажурирање профила и може се извршити на посебној страници преко навигационог менија. Ажурирање профила подразумева измену података као што су корисничко име, *e-mail* адреса и шифра. За промену корисничког имена или *e-mail* адресе проверава се да ли исти већ постоје у бази, а приликом ажурирања лозинке потребно је нову лозинку унети два пута.

DigitalHome

Search Products...

Search

Cart Katarina Ninkovic

USER PROFILE

Name

Katarina Ninkovic

Email Address

kaca.ninkovic.69@gmail.com

Password

Enter password

Confirm Password

Confirm password

Update

MY ORDERS

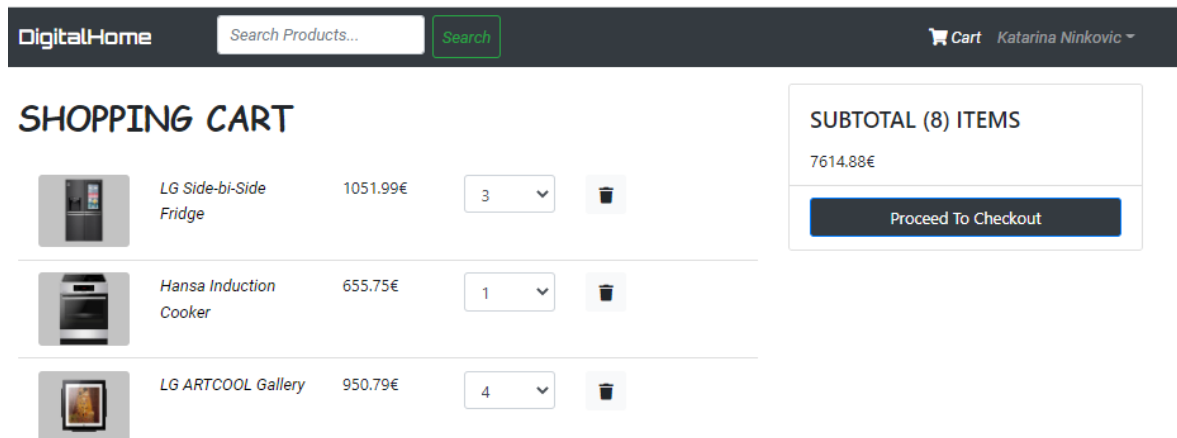
ID	DATE	TOTAL	PAID	DELIVERED		
634a82551156fe754fc5baf5	2022-10-15	689.99	2022-10-15	✗	Details	Cancel
634a82701156fe754fc5bb10	2022-10-15	754.11	2022-10-15	2022-10-15	Details	Cancel

Слика 30. Приказ корисничког профила

### 4.1.9 Корпа

Корпа представља складиште производа које корисник одабере за своју наредну поруџбину. Уласком у корпу корисник може прегледати све производе које је одабрао, заједно са израчунатом укупном ценом поруџбине. Уколико је одустао од куповине неког од продуката, исти може једноставно избацити кликом на икону кантице.

Садржај корпе се чува у *Local Storage*-у, те се приликом креирање поруџбе или одјаве корисника корпа „празни“, односно брише се корпа са *Local Storage*.



Слика 31. Приказ корпе у коју су смештени неки од производа

#### 4.1.10 Поруџбине

Када је корисник задовољан са одабиром производа које жели да купи прелази на следећи корак а то је креирање поруџбине. На овом прозору приказани су подаци о адреси корисника где ће се извршити достава, платни промет преко кога ће се вршити плаћање (*Pay Pal*), још једна провера да ли су то сви жељени производи и наравно обрачун поруџбине са коначном ценом.

The screenshot shows the checkout process on the DigitalHome website. At the top, there is a navigation bar with the logo, a search bar, and a user profile link. Below the navigation bar, there are links for 'Sign In', 'Shipping', 'Payment', and 'Place Order'. The main content area is divided into three sections: 'SHIPPING', 'PAYMENT METHOD', and 'ORDER ITEMS'. The 'SHIPPING' section shows the address: 'Fruskogorska 10, Popinci, 22428, Serbia'. The 'PAYMENT METHOD' section shows 'PayPal' as the selected method. The 'ORDER ITEMS' section lists three items: 'LG Side-by-Side Fridge' (3 units at 1051.99€ each), 'Hansa Induction Cooker' (1 unit at 655.75€), and 'LG ARTCOOL Gallery' (4 units at 950.79€ each). To the right of these sections is an 'ORDER SUMMARY' box showing the total items price (7614.88€), shipping (0.00€), tax (1142.23€), and a final total of 8757.11€. A 'Place Order' button is located at the bottom of the summary box.

ORDER SUMMARY	
Items	7614.88€
Shipping	0.00€
Tax	1142.23€
Total	8757.11€

Слика 32. Детаљан преглед креиране поруџбине

Након што је креирана поруџбина корисник може у било ком тренутку да провери њен статус као и да прегледа поруџбине креиране све до сада одласком на профил. Уколико се одлучи да ипак не жели да изврши куповину постоји могућност отказивања поруџбине кликом на дугме *Cancel*. Ово је могуће урадити само уколико корисник није платио исту. Када се отказивање деси сви производи који су је чинили враћају се на стање како би их нови корисник могао одабрати, ставити у корпу и креирати своју поруџбину.

The screenshot shows the user profile page on the DigitalHome website. The user's name is 'Katarina Ninkovic'. Below the name is a section for 'MY ORDERS'. This section contains a table with columns: ID, DATE, TOTAL, PAID, DELIVERED, and two empty columns. The first row of data shows an order with ID '634dac6ce389eebb6d3df616', DATE '2022-10-17', TOTAL '5021.21', and both PAID and DELIVERED status marked with a red 'X'. To the right of the table are links for 'Details' and 'Cancel'.

ID	DATE	TOTAL	PAID	DELIVERED		
634dac6ce389eebb6d3df616	2022-10-17	5021.21	X	X	Details	Cancel

Слика 33. Преглед статуса креиране поруџбине у корисничком профилу



### 4.1.11 Плаћање

Након што је корисник креирао поруџбину и проверио да ли су сви подаци добро унети може прећи на плаћање путем платног промета *Pay Pal*.

DigitalHome

Search Products...

Search

Cart Katarina Ninkovic

ORDER 634D8F6E2F150EE54A3C2A3D

SHIPPING

Name: Katarina Ninkovic

Email: kaca.ninkovic.69@gmail.com

Address: Fruskogorska 10, Popinci, 22428, Serbia




Not Delivered

PAYMENT METHOD

Method: PayPal

Not Paid

ORDER ITEMS

	LG Side-by-Side Fridge	3 x 1051.99€ = 3155.9700000000003€
	Hansa Induction Cooker	1 x 655.75€ = 655.75€
	LG ARTCOOL Gallery	4 x 950.79€ = 3803.16€

ORDER SUMMARY

Items	7614.88€
Shipping	0€
Tax	1142.23€
Total	8757.11€

PayPal

Debit or Credit Card

Powered by PayPal

Слика 34. Провера података на поруџбини и могућност њеног плаћања

Када је трансакција успешно извршена статус поруџбине се аутоматски мења у плаћено, што се може проверити одласком на профил.

DigitalHome

Search Products...

Search

Cart Katarina Ninkovic


USER PROFILE

Name

Katarina Ninkovic

Email Address

MY ORDERS

ID	DATE	TOTAL	PAID	DELIVERED		
634dac6ce389eebb6d3df616	2022-10-17	5021.21	2022-10-17		Details	Cancel

Слика 35. Промена статуса поруџбине у плаћено

Може се приметити да се у колони плаћено (енг. *Paid*) статус променио са црвеног *x* симбола на датум када је плаћање извршено. Такође, до ажирирања стања долази и на детаљнијем приказу где се поље за плаћање мења у плаћено као на следећој слици.

PAYMENT METHOD

Method: PayPal

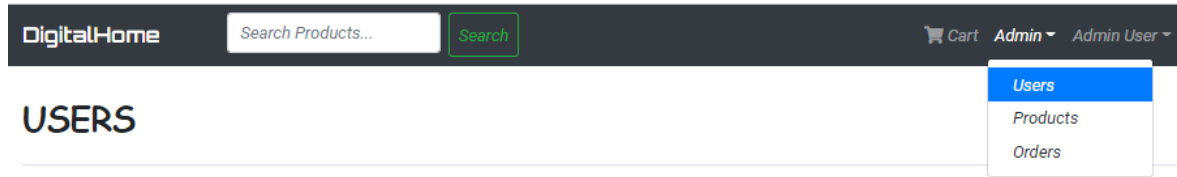
Paid on 2022-10-17T19:39:49.545Z

Слика 36. Ажурирано поље на плаћено у детаљном приказу поруџбине

## 4.2 Администраторски мени

Поред корисника креиран је и администратор који управља сајтом. У питању је напредни кориснички налог. Заправо администратор представља исти модел као и обичан корисник у бази података, с тим што му је индикатор админ постављен на 1.

Приступањем сајту са администраторским креденцијалима отвара се ново поље у навигационом менију са називом админ. Ово поље садржи мени са секцијама корисници, производи и поруџбине. Администратор има могућност прегледа и манипулације подацима из тих секција.



Слика 37. Приказ навигационог менија администратора

С обзиром да је администратор једини у могућности да управља овим подацима неопходно је креирати додатни вид заштите на тим путањама. Поред токена постоји и додатни прстен који проверава да ли је корисник администратор. Уколико су сви услови испуњени може се приступити тим путањама.

## 4.2.1 Преглед и (де)активација корисника

Уласком у секцију корисника, отвара се преглед свих корисника и њихов тренутни статус. Администратор има могућности ажурирања профила корисника са могућношћу постављања неког од корисника за админа.

DigitalHome

Search Products...

Search

🛒

Cart

Admin

Admin User

USERS

ID	NAME	EMAIL	ADMIN		
62f180f43b9a1c0b6d7edd90	Admin User	admin@example.com	✓		
63124dc26d31321b28e61b7d	Jovan Slavujevic	jovan@example.com	✗		
63405b5593a3e4db9fa0bc37	Katarina Ninkovic	katarina@example.com	✗		
63405d015ee881ea699feebb	kaca	kaca@example.com	✗		
63416ab9049ab9669c081dff	Nadja	nadja@example.com	✗		
6349e67164651df3cf4d83ee	Katarina Ninkovic	kaca.ninkovic.69@gmail.com	✗		

Слика 38. Листа свих корисника

Такође, администратор може деактивирати налог уколико то сматра потребним или га вратити у активно стање. Како би то било могуће потребно је прво преузети *ID* корисника којег је потребно деактивирати из *Redux*-а. Након тога помоћу контролера се мења стање у пољу за активацију на *False*. Деактивираном кориснику је онемогућена пријава, а тиме и све могућности везане за пријаву.

## SIGN IN

User is not active

Email Address

kaca.ninkovic.69@gmail.com

Password

.....

Sign In













New Customer? Register

Forgot password?

Слика 39. Грешка приликом покушаја пријаве корисника који је деактивиран

## 4.2.2 Преглед и манипулација производима

На овој страници приказан је преглед свих производа, са могућношћу креирања потпуно нових производа, измене постојећих, као и њихово брисање кликом на кантицу.

DigitalHome		Search Products...	Search	Cart Admin Admin User	
PRODUCTS					+Create Product
ID	NAME	PRICE	CATEGORY	BRAND	
62f180f43b9a1c0b6d7edd94	LG Side-bi-Side Fridge	1051.99€	Electronics	LG	 
62f180f43b9a1c0b6d7edd95	LG Smart Vacuum Cleaner	599.99€	Electronics	LG	 
62f180f43b9a1c0b6d7edd96	Hansa Induction Cooker	655.75€	Solid fuel stoves and fireplaces	Hansa	 
62f180f43b9a1c0b6d7edd97	Samsung Microwave	399.99€	Electronics	Samsung	 
62f180f43b9a1c0b6d7edd98	LG ARTCOOL Gallery	950.79€	Electronics	LG	 
634184e16385ae374b0cd7d8	Predator Laptop 22"	1415€	PC	Acer	 

Слика 40. Листа свих производа

Приликом ажурирања производа неопходно је прво проверити да ли тај производ постоји у бази. Како је *ID* јединствен за сваки производ он је искоришћен за такав вид провере. Уколико се покаже да производ постоји помоћу *useSelector Redux* методе се купе сва поља, односно детаљи о њему. Помоћу *React*-овог *useState*-а омогућено је манипулисање подацима, где је обезбеђена и могућност учитавања слика помоћу *Multer*-а чиме је олакшана измена без ручног уноса путање до слике.

### UPDATE PRODUCT

Name	<input type="text" value="LG Side-bi-Side Fridge"/>
Price	<input type="text" value="1051.99"/>
Image	<input type="text" value="/images/fridge.jpg"/> <input type="button" value="Choose File"/> No file chosen
Brand	<input type="text" value="LG"/>
Category	<input type="text" value="Electronics"/>
Count In Stock	<input type="text" value="5"/>
Description	<input type="text" value="LinearCooling technology reduces temperature fluctuation, thus keepir"/>
	<input type="button" value="Update"/>

Слика 41. Прозор за ажурирање производа

Приликом креирања новог производа добија се сличан мени као за ажурирање. На сликама 41. и 42. се може уочити да се сва поља подударају, с обзиром да су то све потребна поља која чине модел производа у бази. Приликом креирања новог производа шаље се *POST* захтев, чиме се нови подаци уносе у базу података, док се за ажурирање користи *PUT* метода.

## CREATE PRODUCT

Name  
Enter name

Price  
0

Image  
Enter image url  
Choose File No file chosen

Brand  
Enter brand

Category  
Enter category

Count in Stock  
0

Description  
Enter description

Create

Слика 42. Креирање новог производа

### 4.2.3 Преглед, достављање и отказивање поруџбина

Последња опција која је омогућена администратору јесте увид у све поруџбине које су креиране и њихове статусе. Уколико неку од поруџбина није могуће извршити постоји могућност отказивања, али само у случају да је корисник није платио. Дакле, као и корисник, администратор може отказати поруџбину, с тим што корисник бива обавештен о том догађају.

DigitalHome

Search Products...

Search

Cart

Admin

Admin User

ORDERS

ID	USER	DATE	TOTAL	PAID	DELIVERED		
63435783bb6ad498bffe63ea	kaca	2022-10-09	3105.91€	2022-10-09	✖	Details	Cancel
634a82701156fe754fc5bb10	unknown user	2022-10-15	754.11€	2022-10-15	2022-10-15	Details	Cancel
634ec8c7f588b3a82d16a643	unknown user	2022-10-18	3280.23€	✖	✖	Details	Cancel
635167f785b18f9c9c391b7c	Katarina Ninkovic	2022-10-20	2529.95€	2022-10-20	2022-10-20	Details	Cancel
6351bea683c99fd2a7bc73d0	Katarina Ninkovic	2022-10-20	1209.79€	✖	✖	Details	Cancel
6352a9f2d451f3ec4a6cf463	Jovan Slavujevic	2022-10-21	5009.33€	2022-10-21	2022-10-21	Details	Cancel
6352aa43d451f3ec4a6cf499	Nadja	2022-10-21	5467.04€	2022-10-21	✖	Details	Cancel
6352aa80d451f3ec4a6cf4ce	Katarina Ninkovic	2022-10-21	1209.79€	✖	✖	Details	Cancel
6352aaafd451f3ec4a6cf4f1	kaca	2022-10-21	3007.22€	✖	✖	Details	Cancel

Слика 43. Листа свих поруџбина

Како би администратор могао видети детаљније о поруџбини и извршио доставу неопходно је да кликне на поље *Details*. Након тога отвара се прозор као на слици испод.

#### ORDER 635167F785B18F9C9C391B7C

##### SHIPPING

Name: Katarina Ninkovic

Email: kaca.ninkovic.69@gmail.com

Address: Fruskogorska 10, Popinci, 22428, Serbia

Not Delivered

##### PAYMENT METHOD

Method: PayPal

Paid on 2022-10-20T15:24:01.456Z

##### ORDER ITEMS

 Samsung Microwave	1 x 399.99€ = 399.99€
 LG Smart Vacuum Cleaner	3 x 599.99€ = 1799.97€

##### ORDER SUMMARY

Items	2199.96€
Shipping	0€
Tax	329.99€
Total	2529.95€

Mark As Delivered

Слика 44. Детаљан приказ једне од поруџбина коју је потребно доставити

На овом прозору администратор проверава податке о купцу, да ли је извршено плаћање и ако су сви услови испуњени кликом на дугме *Mark as delivered* извршава се достава поруџбине. Након тога аутоматски се мења стање статуса поруџбине на профилу корисника.

## MY ORDERS

ID	DATE	TOTAL	PAID	DELIVERED		
635167f785b18f9c9c391b7c	2022-10-20	2529.95	2022-10-20	2022-10-20	<a href="#">Details</a>	<a href="#">Cancel</a>

Слика 45. Ажурирано стање на достављено у корисником профилу

Још једна од опција која је обезбеђена јесте обавештење о достави путем *e-mail* адресе, како би кориснику стигло обавештење одмах чим је поруџбина достављена, без да константно проверава њен статус на *web* апликацији.

### Order 635167f785b18f9c9c391b7c delivered



master.digitalhome.shop@gmail.com [преко sendgrid.net](#)  
коме ја ▼

Just disclaimer that order has been succesfully delivered.

Enjoy your new stuff :)

---

This email may contain sensitive information

<http://localhost:3000>

Слика 46. Обавештење о достављеној поруџбини путем *e-mail* адресе

## 5. Закључак

Може се закључити да систем чији је циљ био примена клијент-сервер комуникације са практичном применом успешно реализован. Пројекат је укључио развијање модерне *web* апликације са последњим технологијама за развијање истог. Пројекат је у потпуности развијан уз помоћ *JS* програмског језика, како на серверској тако и на клијентској страни, уз библиотеке и окружења као што су *React* и *Node*. Страница поседује респонзивни дизајн, захваљујући *Bootstrap* библиотеци, што омогућава лаку адаптацију на различитим екранима. Омогућен је извештај ниво безбедности услед енкрипције лозинке пре уноса у базу података, као и токенизације захтева са приватних рута захваљујући *JWT* пакету. Код је скалабилно структуриран, чиме је омогућена боља прегледност и коришћење заједничких функција система. Омогућено је динамичко учитавање производа и свих њихових података из базе, те је могуће податке изменити и да, притом, ажурирани подаци буду доступни корисницима. Омогућена су два корисничка простора за кориснике (купце) и администраторе. Администраторски простор обезбедио је да администратори странице могу бити и *Non-Tech* људи, јер не морају да имају директан додир са базом података или кодом како би управљали страницом, већ поседују једноставан интерфејс.

Предлози за побољшање:

- Категоризацију и филтрирање производа
- Подршка за *Android* и *iOS* кроз мобилне апликације
- *Chat bot* за добављање генеричких информација
- Додавање више слика на страници производа
- Прелазак са *HTTP* на *HTTPS* трансфер протокол као стандардизовани гарант сигурности и безбедности интернет странице
- Побољшање корисничког интерфејса у естетском смислу, укључујући доста анимација и покрета визуелних објеката
- Пооштравање критеријума приликом креирања корисничке лозинке, тако да лозинка мора укључити комбинацију слова, бројева и специјалних карактера
- Сервис за рекламацију производа
- Додатна опција активације налога преко *SMS* поруке уз помоћ *Sendgrid*-а
- Интеграција *Visa* и *Master Card* платних промета, за кориснике који не користе *Pay Pal* платформу
- Приказ цена и подршка наплате и обрачуна у динарској валути, поред постојеће наплате у еврима
- Опција резервација производа уколико га нема на стању



## 6. Литература

- [ 1 ] SHAH, DHRUTI NA. *Node .JS*. BPB PUBLICATIONS, 2018.
- [ 2 ] Chodorow, Kristina. *MongoDB*. O'Reilly, 2013.
- [ 3 ] Dunk, Jim. *React*. Van Duuren Media, 2019.
- [ 4 ] Lee, Stan, and Jack Kirby. *Redux*. Tunbridge Wells, Kent: Panini Publishing, 2018
- [ 5 ] [https://www.popwebdesign.net/popart\\_blog/2015/06/sta-je-node-js/](https://www.popwebdesign.net/popart_blog/2015/06/sta-je-node-js/), Шта је *Node.js*, *Pop Web Design* блог, приступљено: октобар 2022.
- [ 6 ] <https://medium.com/@mzarkovicm/lagani-uvod-u-react-a4d07034a604>, Лагани увод у *React*, *Medium* чланак, приступљено: октобар 2022.
- [ 7 ] <https://www.freecodecamp.org/news/what-is-redux-store-actions-reducers-explained/>, *What is Redux*, Free Code Camp, приступљено: октобар 2022.
- [ 8 ] <https://app.sendgrid.com/login>, *Sendgrid*, приступљено: октобар 2022.
- [ 9 ] Doleschal, Johannes. *Optimization and Parallelization of Regex Based Information Extraction*. Bayreuth, 2021.
- [ 10 ] <https://milannikolic.code.blog/2019/12/17/mongodb-crud-operacije/>, *MongoDB CRUD* операције, Милан Николић блог, приступљено: октобар 2022.
- [ 11 ] C., Stawart Delmar Thomas. *NoSQL: Computing, Database Management System, Relational Dataase Management System*. Beau Bassin: Dicho, 2012.
- [ 12 ] PERKINS, DANIEL. *Express.js: Master Express.js and Learn How to Develop Your Web Application*. S.l.: CREATESPACE INDEPENDENT P, 2016.
- [ 13 ] <https://jwt.io/> , *JSON Web Token - JWT*, приступљено: октобар 2022.
- [ 14 ] <https://www.npmjs.com/package/multer>, *Multer*, *npmjs*, приступљено: октобар 2022.
- [ 15 ] <https://www.freecodecamp.org/news/axios-react-how-to-make-get-post-and-delete-api-requests/>, *Axios requests*, Free Code Camp, приступљено: октобар 2022.