

Spring 3-2021

## **FIND-DOCTOR APPLICATION**

Kastriot Beha

*University for Business and Technology - UBT*

Follow this and additional works at: <https://knowledgecenter.ubt-uni.net/etd>



Part of the [Computer Sciences Commons](#)

---

### **Recommended Citation**

Beha, Kastriot, "FIND-DOCTOR APPLICATION" (2021). *Theses and Dissertations*. 2612.  
<https://knowledgecenter.ubt-uni.net/etd/2612>

This Thesis is brought to you for free and open access by the Student Work at UBT Knowledge Center. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of UBT Knowledge Center. For more information, please contact [knowledge.center@ubt-uni.net](mailto:knowledge.center@ubt-uni.net).



Programi për Shkenca Kompjuterike dhe Inxhinierise

**FIND-DOCTOR APPLICATION**

Shkalla Bachelor

Kastriot Beha

Mars/2021  
Prishtinë



Programi për Shkenca Kompjuterike dhe Inxhinierise

Punim Diplome  
Viti Akademik 2017-2020

Kastriot Beha

**FIND-DOCTOR APPLICATION**

Mentori: Ramiz Hoxha

Mars / 2021

Ky punim është përpiluar dhe dorëzuar në përmbushjen e kërkesave të pjesshme për Shkallën Bachelor

## **ABSTRAKT**

Nëpërmjet kësaj ideje unë synojë të krijojë një web aplikacion i cili u mundëson përdoruesëve të kërkojnë mjekë të departamenteve të ndryshme sipas nevojave të secilit përdorues. Renditja e mjekëve pas kërkimit nga ana e përdoruesve apo pacientëve bëhet përmes rating(vlerësimit) nga doktori më i mirë pra doktori me rating më të lartë deri tek doktori me rating më të ulët.

Përveq renditjes së mjekëve ne ofrojmë edhe statistika apo charts për të treguar top 5 mjekët më të mirë, në mënyrë që përdoruesve të ju lehtësojmë gjetjen e doktorit të caktuar dhe të duhurit për nevojat e tyre.

Poashtu përdoruesit mund edhe të vlerësojnë mjekët në aplikacion si dhe të shtojnë komente të caktuara për secilin mjekë.

Kërkimi i mjekëve në aplikacion mund të bëhet përmes çfarëdo pajisje elektronike qoftë ai laptop apo telefon-mobil pasi që ne do të ofrojmë atyre një responsivitet të plotë. Të gjitha të dhënat do të ruhen në NoSQL databazë (MongoDB) dhe ato të dhëna do të shfaqen të përdoruesi, aplikacioni do të dizajnohet në dy module së pari për pacientët i cili përdor aplikacionin dhe kërkon mjekët e ndryshëm përmes web aplikacionit dhe për mirëmbajtësit e web aplikacionit të cilët përditësojnë mjekët, të dhënat e tyre në bazë të arritjeve dhe punës të secilit mjekë.

Regjistrimi i të dhënave në web-aplikacion do të mund të bëhet përmes ekraneve të ndryshme dhe vetëm nga personat e autorizuar.

## **MIRËNJOHJE/FALENDERIME**

Së pari do të doja të falënderoja mentorin tim, profesorin Ramiz Hoxha, ekspertiza e të cilit ishte e paçmuar në formulimin e pyetjeve dhe metodologjisë së hulumtimit. Reagimi juaj i zgjuar më shtyu të theksoj të menduarit dhe e solli punën time në një nivel më të lartë.

Unë gjithashtu do të doja të falënderoja tutorët e mi, Betim Gashi ,Arbër Kadriu dhe Medina Shamolli, për udhëzimet e tyre të vlefshme gjatë studimeve të mia. Ju më siguroat mjetet që më duheshin për të zgjedhur drejtimin e duhur dhe për të përfunduar me sukses disertacionin tim. Përveç kësaj, unë do të doja të falënderoja prindërit e mi për këshillat e tyre të mençura dhe motivimet të cilat më kanë japur. Ju jeni gjithmonë aty për mua. Më në fund, nuk mund ta kisha përfunduar këtë disertacion pa mbështetjen e miqve të mi, kolegëve, të cilët ofruan diskutime stimuluese si dhe shpërqendrim të lumtura për të pushuar mendjen time jashtë hulumtimit tim.

# PËRMBAJTJA

<b>LISTA E FIGURAVE.....</b>	<b>V</b>
<b>LISTA E TABELAVE.....</b>	<b>VI</b>
<b>FJALORI I TERMAVE.....</b>	<b>VII</b>
<b>1 HYRJE.....</b>	<b>1</b>
<b>2 SHQYRTIMI I LITERATURES .....</b>	<b>2</b>
2.1 Fushëveprimi I Projektit(Project Scope) .....	2
2.2 Analiza Teknike Dhe Teknologjike E Sistemit.....	3
2.3 Analiza e Projektit X(doctorondemand.com).....	7
2.4 Analiza Teknike Dhe Teknologjike(doctorondemand.com).....	7
2.4.1 Hyrje .....	7
2.4.2 Përparësitë e Java.....	8
2.4.3 Përparësitë e Vue.js.....	10
2.5 Krahasimi I Gjuhëve Programuese .....	10
2.5.1 Node.js VS Java.....	10
2.5.2 Krahasimi i front-end: React.js dhe Vue.js:.....	13
Menaxhimi i Gjendjës së të Dhënave .....	15
Performanca .....	15
Shkallzueshmëria .....	16
Verdikti .....	17
2.6 MONGODB VS SQL .....	17
2.6.1 Historia.....	18
2.6.2 Si ruhen të dhënat.....	18
2.6.3 Besueshmëria .....	19
2.6.4 Skema.....	20
2.6.5 Query.....	20
<b>3 DEKLARIMI I PROBELMIT.....</b>	<b>23</b>
3.1 QFAR OFRON KY SISTEM.....	23
3.2 TEKNOLOGJITE E PERDORURA.....	23
<b>4 METODOLOGJIA .....</b>	<b>26</b>

4.1	Scrum .....	26
<b>5</b>	<b>REZULTATET .....</b>	<b>28</b>
5.1	Rezultatet e aplikacionit ne fjale .....	28
<b>6</b>	<b>DISKUTIMI DHE PERFUNDIMI.....</b>	<b>30</b>
<b>7</b>	<b>REFERENCAT .....</b>	<b>33</b>
<b>8</b>	<b>APPENDIXES.....</b>	<b>34</b>

## LISTA E FIGURAVE

Figura 1. UBT Logo.....	1
Figura 2.4.1 Procesi i Threads.....	9
Figura 2.5.1. Lista e Frameworks.....	14
Figura 3.1. Procesi I Funksionit te Redux.....	25
Figura 4.1 Scrum Metodologjia.....	27
Figura 5.1 Gjetja e Doktorit sipas departamentit.....	28
Figura 5.2 Listimi i Doktorave sipas departamentit te caktuar.....	29
Figura 5.3 Listimi i Doktorave.....	32



## **LISTA E TABELAVE**

Tabela 1. Specifikimi i sistemit operativ.....	6
Tabela 2. Dallimi Java – Node.js.....	12
Tabela 3. Specifikimi i Sprintave.....	31

## **FJALORI I TERMAVE**

CMS – Content Management System

SSD – Solid State Drive

TDD – Test Driven Development

CPU – Central Processing Unit

SEO – Search Engine Optimization

JSON - Java Script Object Notation

JDK - Java Development Kit



# 1 HYRJE

Nëpërmjet kësaj ideje unë synoj të krijoj një web aplikacion i cili u mundëson përdoruesëve të kërkojnë mjekë të departamenteve të ndryshme sipas nevojave të secilit përdorues. Renditja e mjekëve pas kërkimit nga ana e përdoruesve apo pacientëve bëhet përmes rating(vlerësimit) nga doktori më i mirë pra doktori me rating më të lartë deri tek doktori me rating më të ulët.

Përveq renditjes së mjekëve ne ofrojmë edhe statistika apo charts për të treguar top 5 mjekët më të mirë, në mënyrë që përdoruesve të ju lehtësojmë gjetjen e doktorit të caktuar dhe të duhurit për nevojat e tyre.

Poashtu përdoruesit mund edhe të vlerësojnë mjekët në aplikacion si dhe të shtojnë komente të caktuara për secilin mjekë.

Kërkimi i mjekëve në aplikacion mund të bëhet përmes çfarëdo pajisje elektronike qoftë ai laptop apo telefon-mobil pasi që ne do të ofrojmë atyre një responsivitet të plotë. Të gjitha të dhënat do të ruhen në NoSQL databazë (MongoDB) dhe ato të dhëna do të shfaqen të përdoruesi, aplikacioni do të dizajnohet në dy module së pari për pacientët i cili përdor aplikacionin dhe kërkon mjekët e ndryshëm përmes web aplikacionit dhe për mirëmbajtësit e web aplikacionit të cilët përditësojnë mjekët, të dhënat e tyre në bazë të arritjeve dhe punës të secilit mjekë.

Regjistrimi i të dhënave në web-aplikacion do të mund të bëhet përmes ekraneve të ndryshme dhe vetëm nga personat e autorizuar.

## 2 SHQYRTIMI I LITERATURES

### 2.1 Fushëveprimi i Projektit(Project Scope)

Qëllimi kryesor i këtij projekti është zhvillimi i një web aplikacion i cili u mundëson përdoruesëve të kërkojnë mjekë të departamenteve të ndryshme sipas nevojave të tyre. Ky aplikacion do të ketë të gjitha tiparet për përdoruesit, për të ju mundësuar atyre një përdorim sa më të lehtë të sistemit. Do të mundësojë vizitorëve të kërkojnë mjekë të departamenteve të ndryshme, pasi që aplikacioni përmban një kërkim të sofistikuar për përdoruesit për të kërkuar mjekë specifike për nevojat e tyre.

Fusha e kërkimit (Search bar) ofron një mënyrë të lehtë dhe të përshtatshme për kërkimin e mjekëve ku automatikisht do të përmirësojë mjekët e kërkimit në dispozicion bazuar në të dhënat e përdoruesit. Përdoruesi pastaj mund të shohë specifikat e plotë të secilit mjekë. Ata do të kenë mundësinë të shkruajnë përshtypjet e tyre dhe të japin një vlerësim (rating ) për ndonjë mjekë të caktuar.

Web-aplikacioni në fjalë është i dizajnuar dhe do të qëndrojë online 24orë në 7 ditët e javës e që kjo do të thotë se përdoruesit mund të shfletojnë në web-aplikacionin tonë në çfarëdo kohe që ata dëshirojnë gjithashtu vlen të përmendet se sistemi ofron edhe një kontakt form ku ne si ekip do të jemi në dispozicion për përdoruesit për të marrë kritika positive apo negative, se qfarë duhet të përmisojmë apo të shtojmë në aplikacion.

Pasi që projekti është i bazuar në aktivitete që zhvillohen tërësisht online në duhet të mendojmë edhe për sigurinë dhe ta vendosim si një ndër qështjet primare të projektit të find-doctor, pra ne si ekip do ta ndërtojmë sistemin në fjalë të atillë që të ketë një siguri në nivelin e caktuar pra do të ndajmë grupet e përdoruesve, pra do të kemi autentifikim që do të thotë do të ndajmë userat në 3 grupe , usera pa mode të autentifikimit pra guests pastaj usera me modin user në mënyrë që të

mund të bëjnë vlerësime dhe të shtojmë komente për mjekët e caktuar, pastaj usera me modin moderator këta do të jenë përgjegjës për regjistrimin e mjekëve dhe admin i cili u jep qasje userave në mode të caktuar, e gjithë kjo do të arrihet përmes mongodb apo node js

passport gjithashtu në mënyrë që web të jetë i mbrojtur nga ddos sulmet do të përdorim cdn e që në rastin tonë cloudflare,

dhe nëse ne përcjellim dhe zbatojmë hapat e lartpërmendur ne do tu ofrojmë përdoruesve të platformës sonë të qasën dhe të vazhdojnë me kërkimet e tyre për nevojat e tyre në çdo kohë dhe nga çdo platformë që ata përdorin. Pasi që web-aplikacioni është tërësisht online ai do të jetë i lehtë për tu qasur në çdo kohë dhe çdo paisje.

## **2.2 Analiza Teknike Dhe Teknologjike E Sistemit**

Disa konsiderata duhet të merren parasysh për të vendosur metodologjinë që do të përdoret në këtë projekt siç janë shtrirja, koha, rreziqet dhe objektivat të cilat ne duhet ti përmbushim. Në fillim do të kemi vetëm modulet normale të webit për kërkim të mjekëve. Ku si fillim do të regjistrojmë mjekët e departamenteve të caktuara për të parë se si do të jetë reagimi i përdoruesve për web-aplikacione të tilla si Find-Doctor dhe se si përdoruesit do të ambientohen në një web të tillë, por para kësaj ne fillimisht vetë si staf (ekip) do të bëjmë testimin dhe të shohim se sistemi a është i përshtatshëm për përdoruesit, që ata të mundën me gjetë mjekun e duhur për nevojat e tyre në mënyrë sa më të lehtë dhe më të shpejtë, dhe pastaj përdoruesit do të kenë mundësi për feedback.

Shtimi i moduleve të reja të cilat vijnë së bashku me evolucionin e projektit, pasi që është edhe diçka e re për klientët mbrenda një kohe të caktuar do të shtohet nevoja për krijim të moduleve të reja si dhe për oferta të ndryshme për klientët shtimi i mjekëve do të shtohet dhe do të rritet pasi do të vërehet se kjo është mënyra më e mirë për të i plotësuar nevojat e secilit përdorues, kjo është edhe diçka e mirë edhe për përdoruesit sepse përdoruesit do të kenë mundësi të ndryshme të përzgjedhjes së mjekëve të preferuar. Në web aplikacionin tonë do të kemi akses (qasje) vetëm userat me modin moderator për të vlerësuar dhe komentuar mjekët e ndryshëm në aplikacion.

Përdoruesit tanë do të kenë mundësinë e kërkimit dhe vlerësimit të mjekëve ndërsa ata pastaj vetëm kontaktojnë në forma të caktuara me ne se cilë doktorë apo departamente dëshirojnë ti postojnë në Find-Doctor. Viruset (spam email), ngacmimi virtual cyber bullying, hyrja ne web faqen tonë nga link-u/adresa zyrtare, kriptimi i te dhënave, mbrojtja antivirus, firewall, antispyware, Bllokimi / Lock out Automatik. Pas nje numri tentativash të dështuara për të hyrë në llogarinë online, ne ç'aktivizojme llogarine tuaj. Per te riaktivizuar llogarinë tuaj juve ju duhet të kontaktoni sherbimin Help ose administratën.

Time out / Skadimi kohor i Sesionit. Nëse ju harroni të dilni në mënyre korrekte (log off) paisja pasi e përdorni atë ose nëse kompjuteri juaj qëndron jo aktiv gjate nje sesioni atëherë sistemi ynë ju nxjerr automatikisht nga llogaria juaj.Login do të ofrohet nga sistemi i jonë i autentifikimit por ne do tu ofrojmë përdoruesëve të Find-Doctor edhe mënyra tjera të qasjes në sistem ku si sistem të jashtëm kemi vendosur që të përdoret vetëm google login,pasi është një sistem in-house nuk ka nevoj për përdorim e sistemeve tjera.Të ilustrojm pak se përse përdoret google login.Ne e dimë që kryesisht aplikacionet të cilat përdoren sot që kanë të bëjnë me porosi si dhe pagesa kryesisht përdoret një login form,në këtë drejtim google është zhvilluar mjaft shumë duke mundsuar implementimin e një login gjenerik për qfardo lloj aplikacioni.Arsyeja se pse ne kemi vendosur që të përdoret google login është për lehtësim në përdorim te aplikacionit tonë nga ana e përdoruesit duke mundësuar kështu përmes një gmail account të mundet të kyqet në aplikacionin tonë kështu duke mos e frustruar atë në krijimin e një email të ri të cilin e kemi hasur në disa prej aplikacioneve të cilat krijohen në ditet e sotme duke mos qenë dhe një zgjidhje adekuate,ndërsa sa i përket sistemit të kyqjes që do të zhvillohet nga ne si ekip do të përdorim mongodb si databazë real time dhe sistemin e autentifikimit për verifikimin e userave të cilët qasen në Find-Doctor.

Ne si ekip jemi të pregaditur për zhvillimin e projektit në fjalë falë përvojës te antareve te zhvillimit.Projekti ynë do të bazohet në teknologjinë MEVN që do të thotë (Mongo-si databzë,Express.js-si framework të cilin e përdorë Node.js,React-si front-end framework për SPA web aplikacione dhe Nodejs si javascript runtime environment dhe Mongodb krejt në fund e cila do të na shërbej për ruajtjen e përdoruesëve dhe autentifikimin e tyre në Find-Doctor).Ruajtja e mjekëve të cilët gjenden në Find-Doctor do të ruhen në databazën e llojit NoSql pra që në këtë rast

ne do të përdorim si databazë të llojit të tillë MongoDB pse? Sepse është më e shpejtë në krahasim me databazat e tipit SQL pastaj ka performancë të lartë , shkallëzim automatik , e thjeshtë për instalim dhe mirëmbajtje gjithashtu e lehtë për tu implementuar dhe gjithashtu përdor dokumente JSON apo BSON formate.Kjo databazë do të komunikojë me front-end përmes Node JS e cila do të jetë përgjegjëse për request që bëhen nga përdoruesit dhe pastaj të merr ato të dhëna dhe ti shfaq tek përdoruesi në formate të caktuara.

Arsyet pse do të përdori node js dhe jo ndonjë platformë tjetër është sepse i tërë sistemi është i bazuar në javascript si gjuhë programuese dhe si të tillë edhe node js është i bazuar në javascript pastaj është lightweight dhe i shpejtë për arsye që është asinkron konsumimi i API është më i lehtë dhe mbi të gjitha përdoret për teknologji real-time dhe për fund si teknologji do të përdorim react js ngase është një front-end javascript framework dhe na mundëson ndërtimin e web-aplikacioneve si single (SPA) gjë e cila e bën një web app jashtëzakonisht të shpejtë e sidomos në request e gjitha kjo arrihet falë router i cili na mundëson React teknologjia ndërsa bashkë me atë do të instalojmë edhe plugins të tjerë përmes npm package manager si psh axios për get apo post ose put requesta etj,shkurtimisht ky ishte një përshkrim i asaj se çka do të përdorim gjatë zhvillimit në front-end dhe në back-end.

Ndërsa sa i përket hostimit (deploy) aplikacionin tonë do ta bëjmë deploy në sisteme operative të bazuara në kernelin e Linux pse?sepse do të ulim kostot e projektit sepse nëse do të kishim përdorur windows server do të kishim pasur kosto shumë më të larta për arsye të licensave që duhet blerë e për këtë arsye ne si ekip gjatë diskutimit dhe analizimit kemi vendosur që të përdorim linux sisteme operative e konkretisht ubuntu distro të blerë nga digital ocean të shoqëruar me apache web server së bashku me nginx web server duke e përdorur atë si proxy server dhe e fundit domain do të blihet fillimisht për një periudhë një vjeqare nga regjistrar të domain sistemeve siq janë go daddy apo HostGator ne do të zgjedhim në mes të këtyre dy kompanive varësisht qmimit që ata na e ofrojnë ne.Për të arritur maksimumine dyqanit tonë online në mënyrë që të tërheqim sa më shumë vizitorë dhe numri i klientëve të rritet ne do të përdorim analiza të informacioneve siq janë Google Analytics,analiza e web duhet të bëhet përdoruesit të dukshme për ndryshe bëhet shumë vonë , analiza në kohë të rregullt e Find-Doctor do të ndihmojë në identifikimin e problemeve që përdoruesit tonë hasin si psh gjatë kërkimit të një mjeku.



Gjithashtu ne përmes analizave të web kemi mundësi të shohim se cili doktor kerkohet me shume nga perdoruesit e aplikacionit tone.Pra ne përmes google analytics ne do të arrijmë të optimizojmë Find-Doctor web aplikacionin . Pra krejt në fund mund të themi se zhvillimi i sistemit të Find-Doctor përmban këto aktivitete të cilat po përpiqem ti listojë më poshtë: sistemi do të jetë userfriendly përgjatë gjithë moduleve të sistemit,mënyra se si është i ndërtuar sistemi në fjalë e bënë atë lehtë të menaxhueshëm,pastaj garantojmë se nuk do të ketë keq menaxhim të të dhënave të përdoruesëve ngase ato do të enkriptohen dhe si të tilla ruhen në Mongoddb,mundëson nivel të lartë të autentifikimit të përdoruesëve në mënyrë të parandalimit të keqpërdorimit të tyre,dhe krejt në fund sistemi do të ketë një performancë jashtzakonisht të shpejtë dhe optimale në krahasim me aplikacionet tjera.Disa nga funksionet tjera që do të ketë sistemi jonë e që më lart nuk janë të përmendura janë :CMS për menagjim të webit ,lojalitete,vouchers,promocione me email etj.

Ndërsa sa i përket mënyrës së deploy ky web aplikacion do të bëhet host me një server virtual Vps që do ta blejmë nga kompania Digital Ocean i bazuar ne sitemin operativ ubuntu dhe me këto specifika:

Ram	V-Cpu	Transfer bw	SSD	Qmimi/muaj
16Gb	8Core	6TB	320GB	80\$

Tabela1-Specifikat e sistemit operativ

Kur klienti qaset në Find-Doctor ai ka mundësinë e navigimit për të parë mjekë të ndryshëm, ka mundësinë e vlerësimit dhe për të shtuar komente për secilit mjekë por si fillim ai duhet të jetë i regjistruar dhe i kyqur.Përdoruesi mund të përdorë edhe login with google për të u kyqur ne aplikacionin Find-Doctor.Si dhe duhet të ceket se përdoruesit te cilet regjistrohen ne aplikacion atëherë duhet të verifikojn email adresen e tyre, pra nuk munden me qdo email te regjistrohen, kjo ka rendesi sepse nese kemi nevojë te dergojm email-e te ndryshme tek perdoruesit e ndryshem per ndonje feedback athere duhet te sigurohemi qe te gjith perdoruesit e sistemit janë real me te dhenat e tyre.

## **2.3 Analiza e Projektit X(doctorondemand.com)**

Qëllimi kryesor i këtij projekti doctorondemand.com është zhvillimi i një aplikacioni i cili do të ju mundësoj përdoruesve të bëjnë video chat me doktorin e caktuar. Ky aplikacion i ka të gjitha tiparet për klientët të cilët janë të intereuar dhe kanë nevojë të komunikojn. Të bëjnë video chat me një mjek të caktuar.Do të ju mundësoj përdoruesve te sistemit te regjistrohen, te kyqën ne aplikacion, dhe pastaj ata kanë mundësin që të zgjedhin llojin e mjekut pra departamentin e pastaj e zgjedhin mjekun e caktuar për të bërë video chat me të.

Përdoruesi pastaj mund të shohë specifikat e plota të secilit mjekë, ngjashem pra sikur në aplikacionin Find-Doctor.Web-aplikacioni në fjalë është i dizajnuar dhe do të qëndrojë online 24orë në 7 ditët e javës e që kjo do të thotë se klientët mund të shfletojnë në web-aplikacionin dhe të kërkojnë mjekë të caktuar per video-chat në çfarëdo kohe që ata.Ky web-aplikacion është tërësisht online dhe është i lehtë për tu qasur në çdo kohë dhe çdo paisje, duke ofruar responsivitet të plotë.

## **2.4 Analiza Teknike Dhe Teknologjike(doctorondemand.com)**

### **2.4.1 Hyrje**

Aplikacioni doctorondemand.com si back-end përdorin java, kurse ne front-end vue.js si dhe json formatin për shkëmbimin e informacioneve midis klientit dhe serverit. Le të analizojm pak gjuhët programuese të përdorura nga doctorondemand.com:

Java: Java është një gjuhë programuese me qëllim të përgjithshëm që ndjek paradigmen e programimit të orientuar drejt objektit dhe qasjen: ‘Shkruaj një herë ekzekuto kudo’. Java përdoret për aplikacione desktop, ueb, mobile. Java nuk është vetëm një gjuhë por një ekosistem mjetesësh që mbulon pothuajse gjithçka që mund t’ju duhet për zhvillimin e Java development. Kjo përfshin:

Java Runtime Environment (JRE) - mjet për shpërndarjen e softverit që përmban një Makinë Virtuale Java të pavarur, librarinë standarde Java (Java Class Library) dhe një mjet konfigurimi.

Java mund të gjendet kudo që të shikoni. Është një gjuhë kryesore për zhvillimin e Android. Do ta gjeni në aplikacione në internet, faqet e internetit qeveritare dhe teknologjitë e të dhënave të mëdha siç janë Hadoop dhe Apache Storm. Dhe është gjithashtu një zgjedhje klasike për projekte shkencore, veçanërisht përpunimin e gjuhës natyrore (natural language processing). Java ishte duke dominuar mobilen edhe në ditët para smartphone-it - lojërat e para me celularë në fillimin e viteve 2000 u bënë kryesisht në Java. Pra, është e drejtë të thuhet se Java, falë historisë së saj të gjatë, ka fituar vendin e saj në Sallën e Famës të Programimit. Indeksi TIOBE, një nga renditjet më të njohura të programimit në botë, përdor rezultatet e motorëve të kërkimit për llogaritjen. Përkundër popullaritetit në rritje të Go dhe Python, Java ka mbetur në krye të listës për më shumë se një dekadë.

### **2.4.2 Përparësitë e Java**

**Fleksibiliteti:** Java provoi që kodi procedural, i alokuar manualisht dhe i varur nga platforma i C nuk ishte gjithçka. Falë Java-s, më shumë njerëz filluan të adoptojnë programim të orientuar drejt objektit, një praktikë e përdorët edhe sot.

**Test-driven development:** Me Java, TDD nuk ishte më një praktikë eksperimentale, por mënyra standarde për të zhvilluar softuerin. Futja e JUnit në vitin 2000 konsiderohet si një nga kontributet më të mëdha të Java-s.

**Multithreading:** Në programim, një thread është njësi më e vogël e përpunimit. Për të maksimizuar shfrytëzimin e kohës së CPU, Java ju lejon të ekzekutoni këto thread njëkohësisht - në një proces të quajtur multithreading. Threads ndajnë të njëjtën zonë të memories kështu që kalimi midis tyre kërkon pak kohë.

Ata janë gjithashtu të pavarur, kështu që nëse një thread përballet me exception, nuk ndikon në threads e tjera. Kjo është veçanërisht e dobishme për lojëra dhe programe të rënda për animacione.



Fig 2.4.1-Proceset e threads

Sa i përket front-end nga ondemand.com kanë përdorur Vue.js:

Historia e Vue.js fillon në 2013 kur Evan You po punonte në Google duke krijuar shumë prototipa brenda një shfletuesi. Për atë qëllim, Evan përdori praktika të dobishme nga kornizat e tjera me të cilat ai punoi dhe lëshoi Vue.js zyrtarisht në 2014.

Vue.js është një kornizë progresive për JavaScript që përdoret për të ndërtuar ndërfaqe në internet dhe aplikacione me një faqe. Jo vetëm për ndërfaqet në internet, Vue.js përdoret gjithashtu si për zhvillimin e aplikacioneve në desktop dhe për mobile me kornizën Electron. Shtrirja HTML dhe baza JS shpejt e bënë

Vue një mjet të preferuar të përparmë, të dëshmuar nga adoptimi nga gjigandë të tillë si Adobe, Behance, Alibaba, Gitlab dhe Xiaomi. Vue.js bazohet ne arkitekturen mvc(model –view-controller).

### **2.4.3 Përparësitë e Vue.js**

Madhësia e vogël: vetem 18 KB dhe kjo është anë positive sepse ndikon edhe ne SEO(search engine optimization).

Virtual Dom: Kur një përdorues ndërvepron me faqen, objektet ndryshojnë gjendjen e tyre, kështu që një shfletues duhet të azhurnojë informacionin dhe t'i japë ato në ekran. Por, azhurnimi i tërë DOM-it është i ngadalshëm dhe kërkon kohë. Për shkak të shpejtësisë, Vue.js shfrytëzon DOM virtual: Mendoni për këtë si një kopje të një DOM origjinal që gjen se cilat elemente duhet të bëhen update, pa rirenditur të gjithë DOM-in. Kjo qasje e bën shpërndarjen e faqes mjaft të shpejtë dhe përmirëson performancën e aplikacionit.

## **2.5 Krahasimi I Gjuhëve Programuese**

Krahasimi i aplikacionit find-doctor dhe ondemand.com:

### **2.5.1 Node.js VS Java**

Java është një gjuhë programuese e orientuar në objekte, me qëllim të përgjithshëm dhe e bazuar në klasë. Zhvilluesit(programerët) mund të përdorin parimin - "shkruaj një herë, ekzekuto kudo" me Java. Java si gjuhë programuese është e lëvizshme, që do të thotë një program i shkruar për çdo platformë duhet të ekzekutohet në mënyrë të ngjashme në një kombinim të pajisjes dhe sistemit operativ. Node JS është një librari dhe ambient ekzekutimi i cili është ndër-platformor(cross-platform) dhe përdoret për krijimin e aplikacioneve ne javaScript të ekzekutuara jashtë shfletuesit(browserit). Është një burim i lirë (pra open source) dhe i hapur dhe përdoret për krijimin e aplikacioneve nga ana e serverit. Node JS lejon zhvilluesit të ekzekutojnë kodin e tyre nga ana e serverit. Ai siguron një mënyrë më të shpejtë për të shkruar skripta që janë të lehta dhe të shpejta.

## **Java**

Java u zhvillua fillimisht nga James Gosling në Sun Microsystem. Pjesa më e madhe e sintaksës së Java-s rrjedh nga C++, e cila gjithashtu është e bazuar në klasë dhe e orientuar drejt objektit. Java po transportohet në formën e JDK (Java Development Kit), i cili përfshin komponentë të ndryshëm që kërkohen për të ekzekutuar një program java, megjithatë, jo të gjithë përbërësit e JDK janë të detyrueshëm për të ekzekutuar Java. Versioni i fundit është Java 10, lëshuar në Mars 2018. Java ka mbështetje të gjerë për konkurrencën, rrjetëzimin(networking) dhe GUI (Ndërfaqja Grafike e Përdoruesit). Ekziston një numër i madh i frameworks që janë ndërtuar duke përdorur Java. Duhet të sigurohemi se JRE duhet të instalohet në një sistem operativ të kërkuar i cili mund të shkarkohet nga faqja zyrtare e Java-s. Një përkthyes(compiler) do të shndërronte kodin burimor Java në bytecode, dhe JVM (Java Virtual Machine) e cila është krijuar brenda JRE (Java Runtime Environment) do të shndërronte bytecode në kodin e makinës.

## **Node Js**

Dikush mund të përdorë lehtësisht Node JS për pjesën e front-end, si dhe për zhvillimin e back-end pasi lejon përdorimin e të njëjtit JavaScript. Zhvilluesit mund të shkruajnë aplikacione në kohë reale, dhe në të njëjtën kohë, ajo ofron hapësirë për zhvillimin e aplikacioneve mobile. Aftësitë nga ana e serverit sigurohen gjerësisht në Node JS, një zhvillues mund të dëgjojë dhe t'u përgjigjet kërkesave HTTP në kompjuter, të dëgjojë rrjetin e trafikut dhe në të njëjtën kohë mund të hyjë në bazën e të dhënave nga një kompjuter direkt. Node JS përdor një model të bazuar në ngjarje për të adresuar shkallëzimin dhe për të lejuar librari të pasura JavaScript për modulet JavaScript-it që ndihmon në thjeshtësimin e kodimit. Ka shumë framework të bazuara në Node JS si Express JS, Partial JS, etj. Kur bëhet fjalë për ndërtimin e aplikacioneve të shpejta nga ana e serverit, Node JS është mjaftë i preferueshëm.

JAVA	NODE JS
Gjuha e bazuar në klasë, e orientuar drejt objektit që rrjedh nga C ++	Framework i shkruar në C,C++,Javascript
Kompleks web based, highly concurrent application	Për aplikacione të shpejta në server side
E përshtatshme për projekte komplekse në web	E përshtatshme për projekte me madhësi të vogël
Messaging, web application, highly concurrent application	Highly useful when scalable and faster application is required
Require JDK(Java Development Kit) to run Java	Need to install archive file on system
Dominate server-side interaction	Developers can utilize it on client and server side
Simplify application development with object oriented approach	More suited for application like real time collaborative drawing/edit like Google docs

Tabela2-Krahasimi Java vs Node

Java është një mundësi e shkëlqyeshme për ndërtimin e një aplikacioni kompleks në web, shumë bashkëkohor, ndërsa Node JS është ideale për zhvillimin e projekteve me madhësi të vogël. Java kërkon Java Development Kit i cili përfshin kompajlerin Java dhe runtime environment për të ekzekutuar Java në çdo makinë, ndërsa Node JS duhet të instalohet përmes skedarit(fileve) të instalueshëm të arkivit. Kodi burimor i shkruar në Java përpilohet në bytecode dhe më pas bytecode shndërrohet në kod makine, i gatshëm për t'u ekzekutuar në çdo platformë ndërsa, me Node JS, kodi burimor i shkruar në një skedar burimi është JavaScript, interpretuesi Node JS do të përdoret për të interpretuar dhe ekzekutuar kodin e JavaScript-it. Java është një gjuhë e orientuar drejt objektit duhet të përpilohet dhe ekzekutohet brenda JRE (Java Runtime Environment) ndërsa Node JS është një sistem dhe cross-platform për aplikacione të shkruara në javascript. Detyra intensive e CPU-së që kërkon një fuqi të madhe llogaritëse mund të trajtohet me Java ndërsa operacionet e lidhura me I / O si biseda në kohë reale, transmetimi i mediave etj mund të trajtohen më mirë me Node JS. Për projekte në shkallë të gjerë që përfshinin njëkohësi(concurrency) java rekomandohet shumë ndërsa. Java është gjuha më e popullarizuar e programimit në botë ndërsa Node JS është një framework që është e lidhur ngusht me javascript.

## **Konkluzion**

Java është një gjuhë programimi ndërsa Node JS është një framework e shkruar në C, C ++, JavaScript. Ka shumë gjëra që mund të bëhen me Java në mënyrë efikase, por jo me Node JS dhe anasjelltas. Java ka një grup të gjerë dhe të pasur librarish dhe frameworka të cilat po dominojnë zhvillimin e faqeve në të gjithë globin për vite me rradhë. Node JS është një framework e cila është shumë e përshtatshme për zhvilluesit e JavaScript. Funksionon në mënyrë efikase për klientin dhe pjesën anësore të serverit të një aplikacioni.

Si përfundim krejt varet nga ajo se qfarë po zhvillojm, pra qfarë llojë aplikacioni. Çdo gjë që bëhet me Node JS mund të arrihet në mënyrë efikase me libraritë dhe framworkat. Aplikacionet të shpejta mund të ndërtohen në Java, ndërsa aplikacionet me I / O mund të shkruhen në mënyrë efikase me framworkun Node JS.

### **2.5.2 Krahassimi i front-end: React.js dhe Vue.js:**

#### **React:**

Ndërvepron me dokumentet HTML përmes të ashtuquajturit DOM virtual (d.m.th. një kopje e DOM-it real me të gjitha nyjet të përfaqësuara si objekte javaScripti) dhe ndjek një stil deklarativ të programimit. Qasja e saj e bazuar në komponente ju jep më shumë shpejtësi dhe fleksibilitet kur ndërtoni aplikacione komplekse. Vue.js është një tjetër framework open source për ndërtimin e faqeve komplekse. Përmirëson shumë koncepte që gjenden në Angular dhe React. Vue reklamohet si një framework progresive që do të thotë që ju mund të migroni projektet tuaja ekzistuese në të njëjtën kohë. Në varësi të nevojave tuaja, Vue është midis një frameworku dhe një librerie. Ashtu si React, vue përdorë komponenta, të dhëna dy-drejtimitëshe(two-way data binding), dhe DOM virtual.

Gjatë dy viteve të fundit, Vue u bë jashtëzakonisht e popullarizuar në mesin e zhvilluesve dhe fitoi mbështetjen zyrtare nga Laravel. Popullariteti i një librerie ndikon në numrin e zhvilluesve(developerave). Por më e rëndësishmja, kjo do të thotë që dikush atje tashmë ka zgjidhur problemet që mund të hasni gjatë zhvillimit të projektit tuaj. React është një fitues i qartë



në këtë kategori. Komuniteti i tij i madh përkthehet në më shumë udhëzime, kurse në internet, artikuj dhe 3+ herë më shumë pyetje në Stackoverflow. Një komunitet më i madh nënkupton një ekosistem të madh të librarive, packages, tools, si dhe mbështetjen nga të gjithë IDE-të kryesore.

Për më tepër, libraria është zhvilluar dhe mirëmbajtur nga Facebook, e cila shumë pak garanton mbështetjen e saj afatgjatë. React është përdorur në shumë projekte të facebook dhe secili ekip mund të bëjë update librarinë.

Me React-in që është biblioteka më e popullarizuar, ka inxhinierë më me përvojë në dispozicion. Sipas sondazhit të vitit 2019, më shumë se 48% e zhvilluesve mund të përdorin React në një nivel të përshtatshëm për projektet e tyre (kundrejt 23% për Vue). Raporti i Shkathtësive të Zhvilluesve nga HackerRank përmend që 33.2% e kompanive kanë nevojë për zhvillues React ndërsa vetëm 19% e inxhinierëve kanë aftësitë e kërkuara. Për Vue, mungesa është edhe më e lartë (10% kundrejt 5.1%). Edhe pse fitoi tërheqje vetëm disa vjet më parë, Vue vjen i katërti në listën e teknologjive që programuesit do të donin të mësonin në vitin 2020.

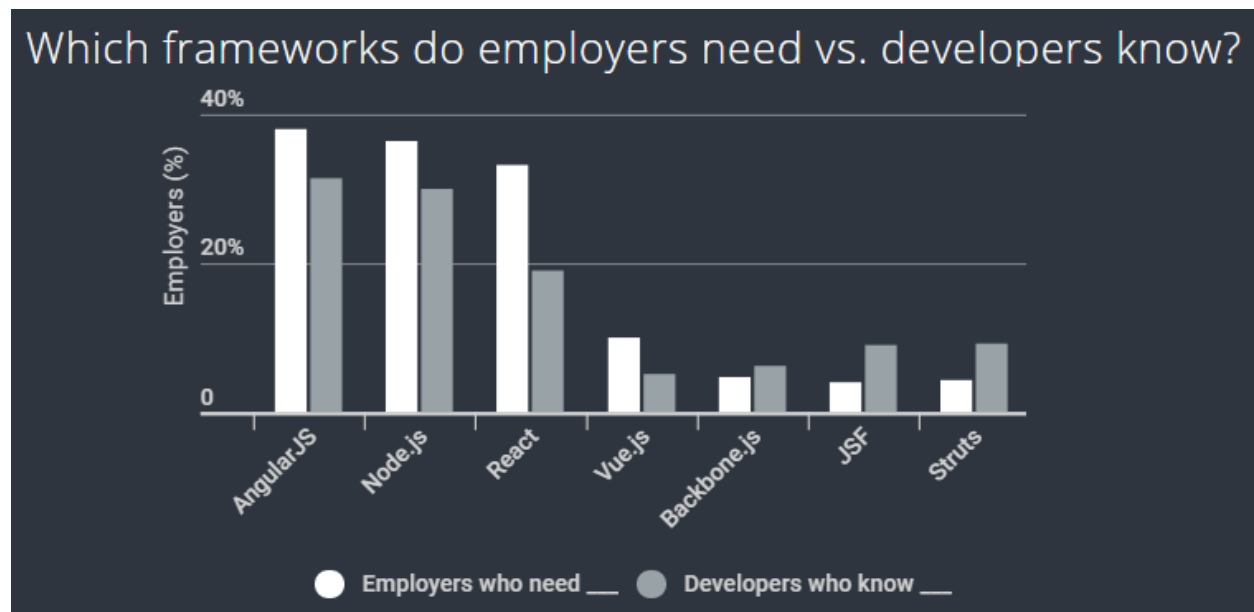


Fig 2.5.1 – Lista e frameworks

Dokumentacioni i React në krahasim me Vue.js, kalon nëpër bazat e zhvillimit të React-it dhe përfshin të gjitha konceptet, por prezantimi nuk është i strukturuar mirë në krahasim me dokumentacionin e Vue.js.

Vue.js ka një dokumentacion më të mirë, dhe më të pasur dhe më i kuptueshëm nga fillestarët. Dokumentacionet janë të shkruara mirë, të qarta dhe të arritshme, duke trajtuar pothuajse gjithçka që ju duhet të dini për të krijuar aplikacione me Vue. Ashtu si në React, dokumentet përkthehen në disa gjuhë përveç anglishtes.

React përdor JSX ku të dy HTML dhe CSS shprehen përmes JavaScript me sintaksën XML. Kjo ju lejon të ndërtoni përbërës të pavarur GUI të cilët përmbajnë të gjitha udhëzimet e renderimit.

Vue ka një qasje më tradicionale me komponentet dhe blloqe të dallueshme për shabllonet, stilet dhe JS HTML. Kjo ndarje është menjëherë e njohur për çdo zhvillues të front-end duke e bërë frameworkun më të lehtë për t'u mësuar.

## **Menaxhimi i Gjendjës së të Dhënave**

State i referohet të dhënave të ndara midis UI komponenteve. Ndërsa sistemi juaj bëhet më i madh, bëhet më e vështirë për të menaxhuar këto të dhëna pa efekte anësore të cilat mund të ndikojnë negativisht në sistemin tuaj. Menaxhimi i state (të dhënat e aplikacionit) është thelbësor për aplikacionet React. Ekzistojnë një numër frameworks si Redux që mund t'ju ndihmojnë dhe lehtësojnë mjaftë punën. Ato ofrojnë një mënyrë të vetme për të modifikuar gjendjen e cila e bën më të thjeshtë korrigjimin e gabimeve.

Në Vue.js nuk ka nevojë për local state sepse ne mund të i modifikojmë të dhënat duke përdorur vetitë e të dhënave të një objekti në Vue.js. Për aplikacione më të mëdha na nevojitet një librari për menaxhimin e të dhënave si psh Vuex.

## **Performanca**

Ekzistojnë dy metrika kryesore që përcaktojnë shpejtësinë e një aplikacioni: koha e fillimit dhe performanca e ekzekutimit.

Të dy libraritë kanë madhësi të vogla pako që përshpejton ngarkesën fillestare (31 KB për Vue / 84,4 KB të pakompresuara dhe 32,5 / 101,2 KB për React).

Për performancën e kohës së ekzekutimit, ju kontrolloni këtë pikë referimi që krahason kohën e fillimit, ndarjen e memories dhe kohëzgjatjen e operacioneve për frameworkat dhe libraritë e njohura të JS.

Të dy libraritë kanë performancë të shkëlqyeshme. Vue është pak më i mirë në caktimin e memories dhe kohën e fillimit ndërsa React është pak më i shpejtë në kohën e ekzekutimit.

Megjithatë, nuk duhet të gjykoni për performancën e librarisë vetëm në bazë të standardeve. Shpejtësia mund të ndryshojë shumë në varësi të madhësisë së aplikacionit dhe përpjekjeve tuaja për optimizim.

## **Shkallzueshmëria**

React është mjaft i lehtë për t'u përdorur në aplikacione me shumë faqe (Facebook e bën këtë në shumicën e faqeve të tij). Ju mund të importoni librarinë dhe të filloni të shtoni veçori të JS me komponentet React. Libraria është gjithashtu e përshtatshme për SPA të mëdha.

Për të filluar projektin tuaj, do t'ju duhen packages të formës së palëve të treta dhe zgjidhje për routing (p.sh. Flux dhe Redux).

Vue është i shkëlqyeshëm për krijimin e aplikacioneve interaktive me shumë faqe. Kjo ju lejon të importoni shpejt librarinë kryesore dhe të vendosni Vue në faqet ekzistuese. Ju as nuk keni nevojë të përdorni përbërës për veçori më të thjeshta.

Me Vue CLI 4, mund të filloni ndërtimin e një projekti të ri në pak kohë. Ai siguron skedar(scaffolding) që mund të përdoren me sisteme ndërtimi si Webpack ose Browserify (ose pa sistem ndërtimi fare).

Ju duhet të instaloni packages që ju nevoiten për ndërtimin e aplikacionit tuaj. Instalimi i librarive të ndryshme mund të bëhet shumë lehtë përmes npm.

## Verdikti

React është udhëheqësi i testuar nga beteja me mbështetjen nga korporatat dhe një komunitet i madh me burim të hapur(open source). Libraria shkallëzon më mirë duke ju lejuar të krijoni aplikacione komplekse për shkallën e ndërmarrjes.

Nuk ka mungesë të zhvilluesve të kualifikuar të React. Ekosistemi i tij i madh do të thotë që ju mund të gjeni një zgjidhje për pothuajse çdo problem që mund të hasni në projektin tuaj.

Më në fund, React Native është zgjidhja më e mirë cross-platform në treg.

Vue është ylli në rritje i zhvillimit të frontit. Është e lehtë për të mësuar dhe një kënaqësi për të punuar me të. Ajo ka një sintaksë më tradicionale që ju lejon të migroni gradualisht projektet ekzistuese në Vue.

Dokumentacioni i tij është më i miri në klasë dhe performanca është edhe më e mirë se sa me React.

Vue ofron më shumë mjete jashtë kutisë me mbështetje nga ekipi kryesor. Me Vue CLI 4, ju mund të vendosni shpejt një projekt dhe të jepni një zgjidhje të gatshme për tregun brenda disa javësh.

## 2.6 MONGODB VS SQL

Bazat e të dhënave MongoDB dhe SQL janë dy anët polare të kundërta të botës së back-end-it. E para merret me të dhëna kaotike të pastrukturuara, ndërsa e dyta punon me të dhëna të strukturuar të organizuara. Të dy databazat kanë avantazhet dhe disavantazhet e tyre dhe janë të destinuara për lloje të ndryshme të rasteve të përdorimit. Në këtë artikull, ne do të bëjmë një krahasim të thelluar midis bazave të të dhënave MongoDB vs SQL, (për të qenë të saktë bazën e të dhënave MySQL) dhe gjithashtu do të prekim temën e rëndësishme se si mund të kryejmë analitikë MongoDB të ngjashme me lehtësinë me të cilën bëhen analitikat e SQL.

### **2.6.1 Historia**

MongoDB i përket familjes së bazave të të dhënave NoSQL e cila përdoret për ruajtjen e dokumenteve të pastrukturuara në formatin JSON. Ajo u lançua për herë të parë në vitin 2009 dhe që atëherë është bërë një nga bazat e të dhënave kryesore në hapësirën e NoSQL.

MySQL është një bazë e të dhënave relacionale SQL me burim të hapur(open source), e cila përdoret për ruajtjen e të dhënave të strukturuara në një format të ngjashëm me tabelën. Ajo u lançua për herë të parë në vitin 1995 dhe tani menaxhohet nga Oracle. Meqenëse është falas, ajo është bërë një zgjedhje shumë e popullarizuar në kërkesë për bazat e të dhënave SQL.

### **2.6.2 Si ruhen të dhënat**

Në MySQL, të dhënat ruhen në tabela, ku kolona tregon atributin dhe rreshti përfaqëson një rekord të veçantë. Nga ana tjetër, këto tabela gjenden brenda bazave të të dhënave. Në MongoDB, të dhënat ruhen në koleksione që janë analoge me tabelat e MySQL. Një koleksion mund të përbëhet nga shumë dokumente në të cilat të dhënat ruhen në formatin JSON të vlerës kryesore. Mund të ketë qindra koleksione të tilla brenda një baze të të dhënave MongoDB. Bazat e të dhënave SQL kanë një veti relacionale ku tabela të ndryshme lidhen me njëra-tjetrën me çelësa të huaj, çelësa primarë. P.sh. Kolona e EmployeeID e cila do të ishte një çelës kryesor i tabelës së Punonjësve do të ishte e pranishme si një çelës i huaj në tabelën e Pagesave, duke lidhur kështu dy tabelat me pronën referuese. Kjo marrëdhënie siguron që të mos ketë asnjë hyrje pagese të një punonjësi, detajet e të cilit nuk janë të pranishme në tabelën master të punonjësve. Kjo është arsyeja pse bazat e të dhënave SQL si MySQL quhen gjithashtu baza të të dhënave relacionale.

Nga ana tjetër, në MongoDB, ne nuk mund të vendosim një marrëdhënie të tillë midis të dhënave të pastrukturuara të koleksioneve. Prandaj, ajo konsiderohet si një bazë e të dhënave jo-relacionale.

Arkitektura e bazave të të dhënave SQL si MySQL rregullohet nga parimet e pronës ACID.

ACID qëndron për Atomicitetin, Konsistencën, Izolimin dhe Qëndrueshmërinë. Këto prona përqendrohen në qëndrueshmërinë dhe besueshmërinë e transaksionit të bërë në bazën e të dhënave.

MongoDB është ndërtuar mbi parimet e Teoremës së CAP, e cila përqendrohet në qëndrueshmërinë, disponueshmërinë dhe ndarjen. Ndryshe nga vetitë ACID të bazave të të dhënave SQL, teorema e CAP përqendrohet në disponueshmërinë e të dhënave në rastin e MongoDB.

### **Shkallzueshmëria**

Baza e të dhënave MySQL ose bazat e të dhënave SQL, në përgjithësi, mund të shkallëzohen vetëm vertikalisht duke rritur madhësinë e memories, hapësirën e diskut ose fuqinë kompjuterike të serverit. Shkallëzimi vertikal mund të jetë i shtrenjtë me rritjen e kostove me shpejtësi për bazat e të dhënave të mëdha me vëllim të lartë.

Bazat e të dhënave NoSQL si MongoDB mbështesin shkallëzimin horizontal, i njohur gjithashtu si ndarje. Në këtë rast, në vend që të rritet konfigurimi i serverit, një server i ri shtohet me qëllim të shkallëzimit. Kjo qasje është zakonisht më pak e kushtueshme sepse një grup i pajisjeve të mallrave me kosto të ulët mund të plotësojnë së bashku kërkesat për të mbështetur vëllimin e lartë të pyetjeve në një mënyrë me kosto efektive.

### **2.6.3 Besueshmëria**

Besueshmëria dhe disponueshmëria janë metrika kryesore për të matur se sa i fortë është çdo sistem i bazës së të dhënave. Shumica e bazave të të dhënave SQL fillimisht ishin krijuar për servera të pavarur. Për të zbutur rrezikun e dështimit, arkitektura e tyre u zhvendos drejt një baze të dhënash të shpërndarë, ku baza e të dhënave funksionon në një tufë nyjesh, duke rritur kështu rezistencën. Edhe nëse një nyje në grumbull është poshtë, baza e të dhënave do të vazhdonte të funksiononte në nyjet e tjera.

Bazat e të dhënave NoSQL si MongoDB u krijuan fillimisht duke mbajtur parasysht qëndrueshmërinë. Ekziston në një grup të pajisjeve të mallrave dhe përsërit të dhënat nëpër nyje për besueshmëri dhe disponueshmëri të lartë. Ndryshe nga bazat e të dhënave SQL, besueshmëria dhe disponueshmëria është një tipar integral i arkitekturës MongoDB dhe jo një mendim i mëvonshëm. Prandaj, dështimi automatik në MongoDB është më i shpejtë dhe më pak kompleks krahasuar me MySQL dhe bazat e të dhënave të tjera SQL.

#### 2.6.4 Skema

Bazat e të dhënave MySQL, si çdo bazë tjetër e të dhënave SQL, kanë një skemë të paracaktuar, të cilës të dhënat duhet t'i përmbahen. Për shembull, numri i kolonave në një tabelë së bashku me llojin e saj të të dhënave duhet të përcaktohet gjatë krijimit të tabelës. Çdo e dhënë që ruhet në tabelë duhet të përputhet me strukturën e tabelës, përndryshe, do të japë një gabim.

Nga ana tjetër, në MongoDB, nuk ka nevojë të paracaktohet ndonjë skemë. Një koleksion mund të ruajë lloje të ndryshme të dokumenteve pa ndonjë problem. Nuk ka asgjë për t'u shqetësuar nëse mbërrin një lloj i ri i dokumentit, ai lehtë mund të ruhet.

Natyra dinamike e skemës MongoDB është e dobishme sepse shumica e të dhënave që gjenerohen nga aplikacionet e internetit dhe pajisjet IoT janë jo të strukturuar të cilat nuk mund të ruhen në një bazë të dhënash tradicionale SQL.

#### 2.6.5 Query

Baza e të dhënave MySQL mund të kërkohet me ndihmën e Structured Query Language ose SQL. Në fakt, MySQL ndjek standardet ANSI SQL që është një standard i zakonshëm SQL i miratuar nga pothuajse të gjitha bazat e të dhënave relacionale si Oracle, PostgreSQL, Sybase, etj.

Quer-it e SQL janë miqësore(friendly) për zhvilluesit dhe të vendosura mirë. SQL mund të përdoret për të kryer funksione të përparuar të analizave si filtra, bashkime, bashkime dhe grumbullime në të dhëna gjithashtu. Kjo e bën SQL një mundësi të fuqishme për kryerjen e analizave të përparuar.

MongoDB nuk i mbështet pyetjet tradicionale SQL ashtu si e bën MySQL. Sidoqoftë, MongoDB mbështetë pyetjen e dokumenteve, por tipari është i pazhvilluar dhe i kufizuar - veçanërisht në krahasim me SQL.

Një shembull i kësaj është që pyetjet MongoDB nuk mbështesin bashkimet, i cili është një operacion vendimtar për të nxjerrë informacion nga burime të shumta të të dhënave.

## Analizimi I Json Formatit

Aplikacioni Find-Doctor për komunikim përdorë json formatin, për të mundësuar bartjen e të dhënave mes back-end(Node.js) dhe front-end(React.js) prandaj është e nevojshme të analizojm json formatin, se cilat janë përparësit dhe mangësit e tijë.

Json(Java Script Object Notation) – është një paraqitje pa skemë, e bazuar në tekst, e të dhënave të strukturuar e cila bazohet në key-value pairs.Json rrjedh nga Javascript, json suportohet përmes librave të ndryshme të cilat mund të instalohen përmes npm në shumicën e gjuhëve programuese.Json kryesisht përdoret për shkëmbimin e informacionit midis klientit(browserit) dhe serverit.Gjatë 15 viteve të fundit, json është pothuajse formati më i përdorur në internet. Popullariteti i JSON gjithashtu ka rezultuar në mbështetjen të JSON nga shumë baza të të dhënave.

Bazat e të dhënave relacionale si PostgreSQL dhe MySQL tani dërgojnë mbështetje për ruajtjen e të dhënave JSON. Bazat e të dhënave NoSQL si MongoDB dhe Neo4j gjithashtu mbështesin JSON, megjithëse MongoDB përdor një version binar të JSON paksa të modifikuar, prapa skenave. JSON është një format i përgjithshëm i të dhënave me një numër minimal të llojeve të vlerës: vargjet(arrays), numrat, booleans, listat, objektet dhe null. Megjithëse shënimi është një nëngrup i JavaScript, këto lloje përfaqësohen në të gjitha gjuhët e zakonshme të programimit, duke e bërë JSON një kandidat të mirë për të transmetuar të dhëna.Në aplikacionin Find-Doctor kam përdorur json format për të shkëmbyer të dhëna në mes të back-end(Node.js) dhe front-end(React.js).

Krahasimit mes JSON dhe XML:

Struktura e modelit të të dhënave JSON përputhet me të dhënat: Struktura e të dhënave e JSON është një tree(pemë), ndërsa XML është një tree(pemë). Megjithëse një tree (vetëm çiftet e çelësit / vlerës –key-value pairs) mund të jetë kufizuese, kjo është ajo që duam, sepse është më e lehtë për t'u interpretuar dhe është e parashikueshme.

Në kod: Artikujt përfaqësohen në të njëjtën mënyrë në kod. Në shumë gjuhë, posaçërisht ato dinamike, thjesht mund të përdorim JSON 'dhe menjëherë të keni objektin e domenit tuaj. Është e lehtë të kalosh nga objektet në JSON tek objektet në kod sepse ato përafrohen. Kur shkoni nga



objektet në XML në objektet në kod ato nuk përputhen dhe ka shumë hapësirë për interpretim, dhe ndonjëherë edhe gabime.

JSON është kufizuese, pra të limiton, por kjo është një gjë e mirë: JSON është e kufizuar për sa i përket asaj që objektet mund të modelohen. Disa mund të mendojnë se XML është më e mirë sepse më shumë objekte mund të modelohen dhe nuk i ndalon zhvilluesit. Por edhe pse JSON ndalon zhvilluesit, është në një mënyrë pozitive, duke e bërë kodin më të thjeshtë, më të parashikueshëm dhe të lehtë për t'u lexuar. XML mund të formatohet për të parë dhe funksionuar në çdo mënyrë që një kompani dëshiron, por kjo e bën të vështirë për zhvilluesit të lexojnë, kuptojnë dhe konvertojnë. Në shumicën e rasteve njerëzit besojnë se XML është më e mirë sepse zhvilluesit mund të bëjnë gjithçka, por ne 15 vitet e fundit JSON është një alternative më e mirë.

## **3 DEKLARIMI I PROBELMIT**

### **3.1 Çfarë ofron ky sistem**

Nëpërmjet kësaj ideje unë synoj të krijoj një web aplikacion i cili u mundëson përdoruesëve të kërkojnë mjekë të departamenteve të ndryshme sipas nevojave të secilit përdorues. Renditja e mjekëve pas kërkimit nga ana e përdoruesve apo pacientëve bëhet përmes rating(vlerësimit) nga doktori më i mirë pra doktori me rating më të lartë deri tek doktori me rating më të ulët.Përveq renditjes së mjekëve ne ofrojm edhe statistika apo charts për të treguar top 5 mjekët më të mirë, në mënyrë qe përdoruesve të ju lehtësojm gjetjen e doktorit të caktuar dhe të duhurit për nevojat e tyre.

Poashtu përdoruesit mund edhe të vlerësojn mjekët në aplikacion si dhe të shtojn komente te caktuara per secilin mjekë.Kërkimi I mjekëve ne aplikacion mund te behet përmes çfarëdo paisje elektronike qoftë ai laptop apo telefon-mobil pasi që ne do tu ofrojmë atyre një responsivitet të plotë.Të gjitha të dhënat do të ruhen në NoSQL databazë (MongoDB) dhe ato të dhëna do të shfaqen te përdoruesi ,aplikacioni do të dizajnohet në dy module së pari për pacientët i cili përdor aplikacionin dhe kërkon mjeket e ndryshem përmes web aplikacionit dhe për mirëmbajtësit e web aplikacionit të cilët përditësojnë mjekët, të dhenat e tyre në bazë të arritjeve dhe punes te secilit mjekë.

Regjistrimi i të dhënave në web-aplikacion do të mund të bëhet përmes ekraneve të ndryshme dhe vetëm nga personat e autorizuar.

### **3.2 Teknologjite e perdorura**

Pra aplikacioni Find-Doctor përdorë si gjuhë programuese ne back end Node.js, ne front end kam përdorur React.js, kurse si databaz për ruajtjen e të dhënave kam përdorur MongoDB.

Arsyeja e përdorimit të node.js në pjesën e back-end është sepse është një gjuhë mjaft e përshtatshme për zhvillimin e aplikacioneve web si dhe shumë lehtë mund të kombinohet me react-in në front. Ka shumë library të cilat mund të instalohen përmes npm-it dhe me ndihmën e tyre shumë lehtë mund të kryejmë query nga më të ndryshmet.

Arsye tjetër është edhe se node.js ofron fleksibilitet në krijimin e api-ve, po ashtu edhe të micro-serviceve.

### **Pse React?**

Aplikacioni është i ndarë në komponente, dhe kjo është përparësi sepse përmes komponenteve mundësohet ripordorshmëria e kodit.

Arsye e përdorimit të react është se shumë lehtë mund të qasemi në api, në back-end (node.js) përmes axios, dhe pastaj të dhënat i marrim në json format, dhe shumë lehtë i renderojmë të dhënat në komponente.

Pastaj arsye tjetër dhe mjaft e rëndësishme është state management (menaxhimi i të dhënave të aplikacionit) e që shumë lehtë mund të arrihet përmes react-it, përkatësisht përmes redux, ku në të ruhen të gjitha të dhënat e aplikacionit dhe komponenten qasen në të dhëna përmes redux.

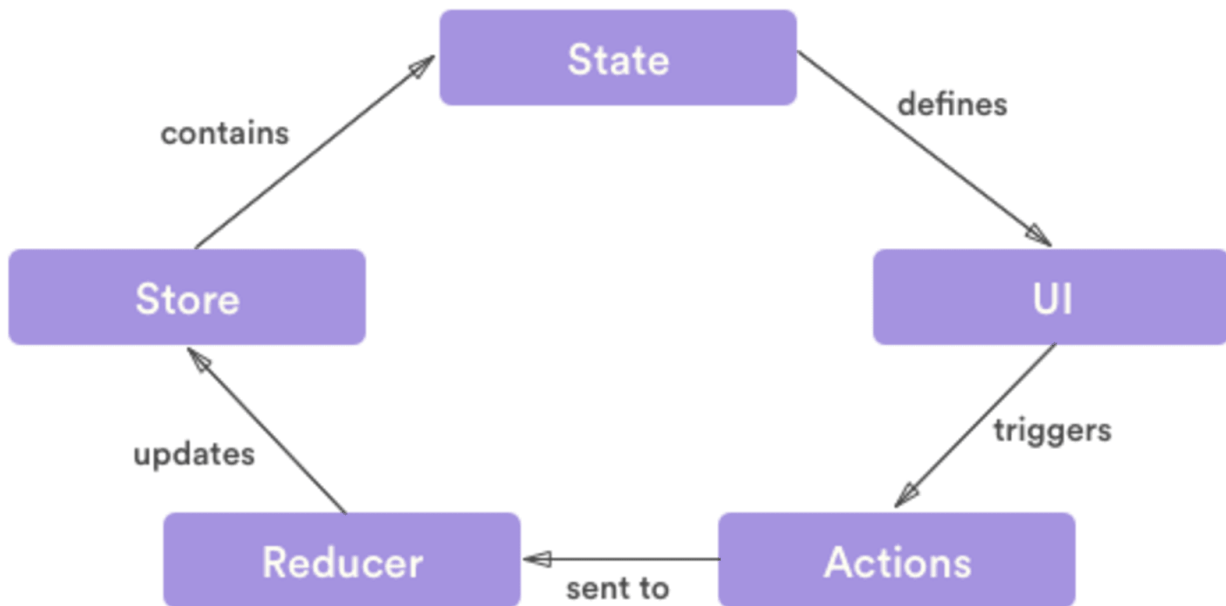


Fig 3.1-Redux State Managment

Kur klienti qaset në Find-Doctor ai ka mundësinë e navigimit për të parë mjekë të ndryshëm, ka mundësinë e vlerësimit dhe për të shtuar komente për secilit mjekë por si fillim ai duhet të jetë i regjistruar dhe i kyqur.

Përdoruesi mund të përdorë edhe login with google për të u kyqur ne aplikacionin Find-Doctor.Si dhe duhet të ceket se përdoruesit te cilet regjistrohen ne aplikacion atëherë duhet të verifikojn email adresen e tyre, pra nuk munden me qdo email te regjistrohen, kjo ka rendesi sepse nese kemi nevojë te dergojm email-e te ndryshme tek perdoruesit e ndryshem per ndonje feedback athere duhet te sigurohemi qe te gjith perdoruesit e sistemit janë real me te dhenat e tyre.

## 4 METODOLOGJIA

Në temën në fjalë unë implementoj një metodologji eksperimentale për aplikacionin në fjalë, duke ju përshtat metodologjise së implementimit të sistemeve softuerike, përkatesisht procesi Scrum.

### 4.1 Scrum

Mënyra për t'iu qasur shumicës së projekteve të softuerit nga përsëritje (interacionit), domethënë periudha më e shkurtër kohore (zakonisht javësh) ku ndodh i gjithë procesi (planifikimi, zhvillimi, testimi, lirim, reagimi i klientit). Një kornizë-punë e njohur për menaxhimin e softuerit që zbaton këtë ide quhet Scrum.

Scrum jo vetëm që i lejon klientit të fillojë përdorimin e softuerit më herët (duke shtuar vlerën në biznesin e tij), por gjithashtu i jep mundësi atij që të ofrojë reagime (feedback) më të shpejta, të kërkojë ndryshime më herët, të ndryshojë prioritetet e funksionit dhe kështu me radhë.

SCRUM kërkon një ndryshim mendor në mënyrën se si njerëzit mendojnë: oPreferenca mbi njerëzve: të kuptuarit për të zgjedh probleme komplekse kërkon fuqi truri, jo receta;

Duke kuptuar se produktet më të mira janë zhvilluar duke patur një fokus në nevojat e përdoruesit dhe jo duke u mbështetur në dokumentin e kërkesave;

Preferencat në Ekipet mbi vetë-organizuesin v.s liderit të vetmuar ose menaxhimit i kontrollimit të fort. Duke kuptuar/ralizuar që secili antarë është pjesë e një ekipi të zhvillimit të produktit dhe se ata nuk janë thjesht njerëz/puntorë që bëjnë thjeshtë punë.

### Modeli Scrum

Scrum është një kornizë-punë (framework) e përdorur nga ekipet për të menaxhuar punën e tyre.

Ekzistojnë tri faza në Scrum:

Faza fillestare është një fazë planifikuese në të cilën ju vendosni objektivat e përgjithshme për projektin dhe dizajnoni arkitekturën e softuerit.

Kjo pasohet nga një sërë cikle sprintesh, ku çdo cikël zhvillon një interacion të sistemit.

Faza e mbyllëse e projektit përfundon projektin, plotëson dokumentacionin e kërkuar ndimës së sistemit dhe doracakët e përdorimit dhe vlerëson mësimet e nxjerra nga projekti.

Scrum zbaton parimet e Agile si një bashkësi e artefakteve, praktikave dhe roleve dhe parimeve.

Korniza e punës së scrum-it përcakton: 3-role, 5-ngjarje(evente), 3-artefakte (objekte), dhe 6-principe (parime).

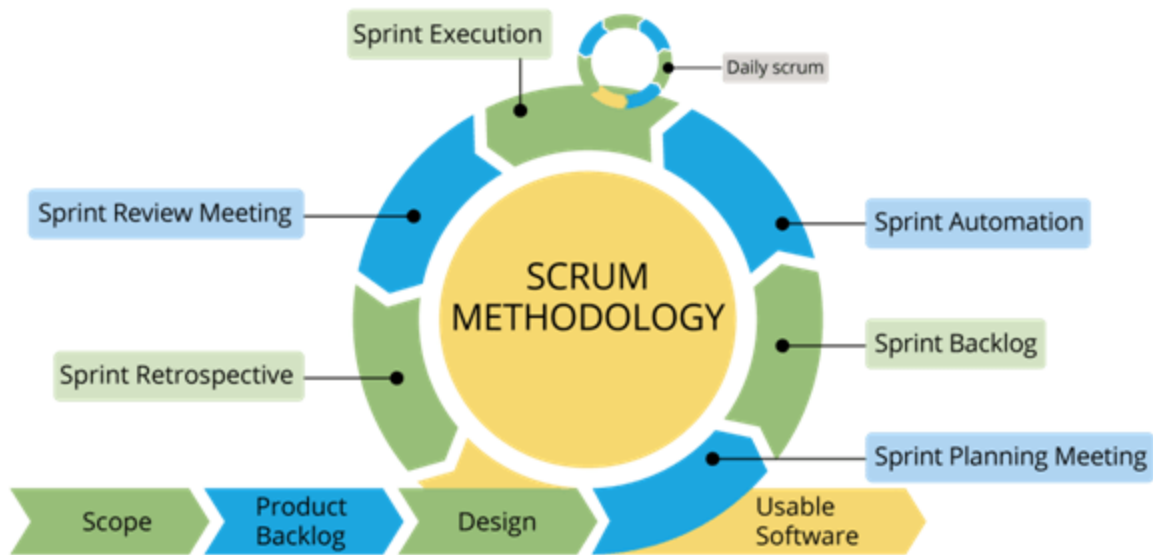


Fig 4.1-Modeli Scrum

## 5 REZULTATET

### 5.1 Rezultatet e aplikacionit ne fjale

- Gjetja e doktorit sipas departamentit te caktuar

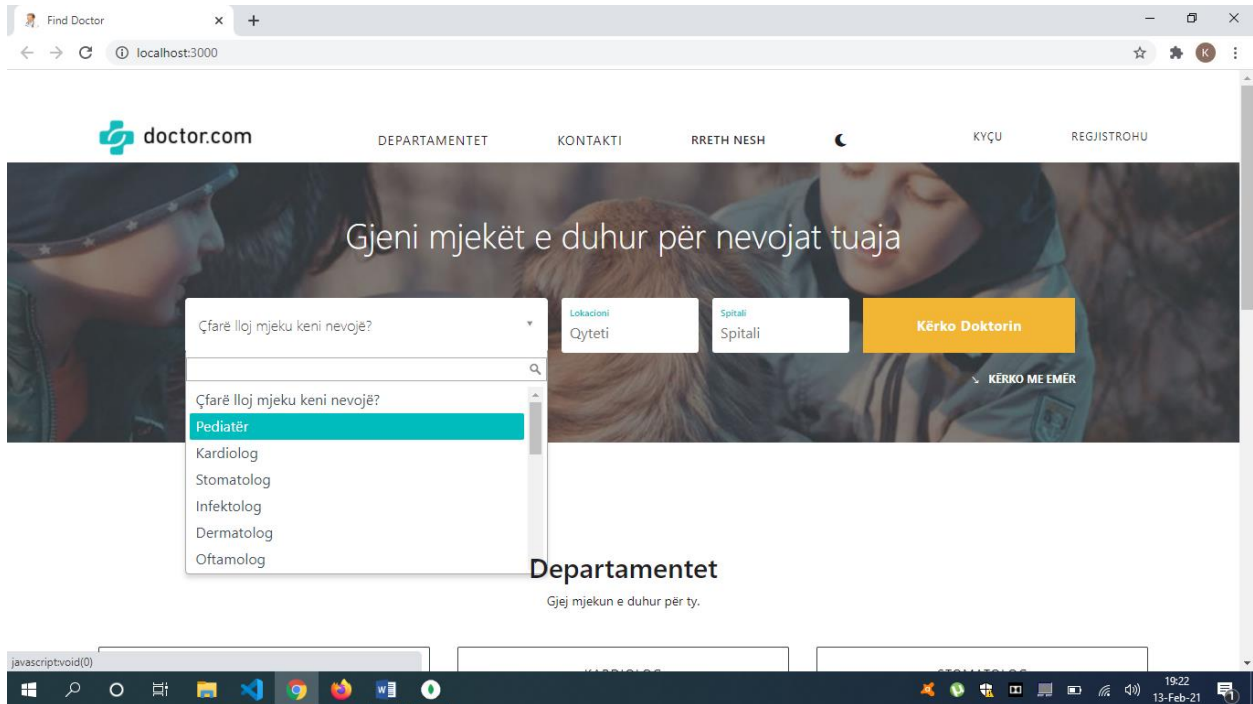


Fig 5.1-Gjetja e doktorit sipas departamentit

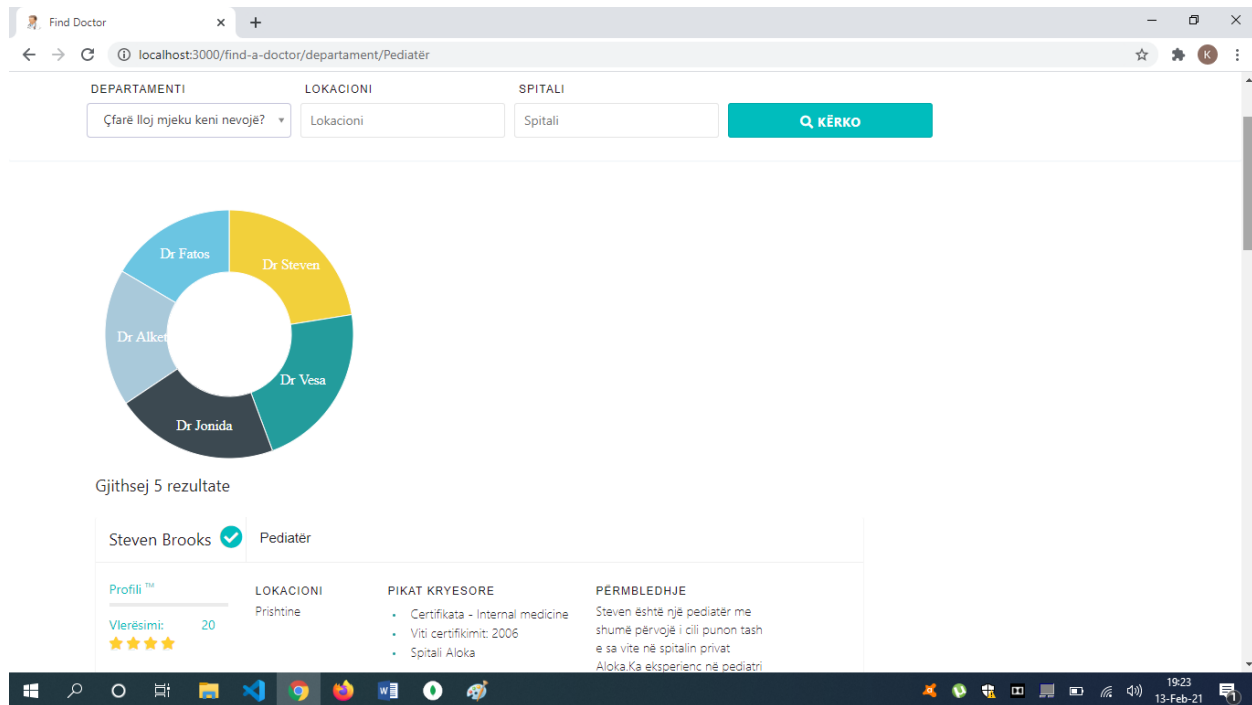


Fig 5.2-Listimi i doktorave sipas departamentit te caktuar

- Gjetja e doktorit ne baze te lokacionit.
- Reset Password.
- Votimi I doktorit nga ana e perdoruesve (pacientave).
- Password hashing. Kta e argumentoj posht tek diskutimi pse e kom bo edhe me qka e kom bo.
- Definimi i top mjekëve sipas departamentit te tyre (shembull top 5 kardiologët më të njohur).
- Implementimi i Login with google.



## 6 DISKUTIMI DHE PERFUNDIMI

### Funksionet e Aplikacionit të Find-Doctor janë:

- Useri ka mundësinë të kërkojë mjekë të ndryshëm në aplikacionin Find-Doctor.
  - Useri ka mundësinë e çasjes nga telefonët mobil, tablet, laptop ngase aplikacioni është responsiv.
  - Useri ka mundësinë e regjistrimit përmes google, si dhe përmes email-it.
  - Useri ka mundësinë të shtojë komente për mjekë të ndryshëm.
  - Useri ka mundësinë e fshirjes së komenteve që u ka bërë mjekëve.
  - Useri ka mundësinë që të vlerësojë doktorët e ndryshëm.
  - Useri ka mundësinë të shohë mjekët më të vlerësuar në aplikacion.
  - Useri ka mundësinë e editimit të komenteve që u ka bërë mjekëve.
  - Sistemi ofron siguri të të dhënave, të dhënat janë të enkriptuara.
  - Sistemi lejon vetëm adminin të regjistroj një mjekë.
- Mongoddb si database në këtë project e kam përdorur sepse: Modeli i të dhënave të dokumentit është një mënyrë e shpejt dhe e lehtë për të ruajtur dhe rimarrë të dhëna që më lejon të qasem shpejt në të dhëna. Pastaj arsye tjetër është arkitektura horizontale, e shkallëzuar e MongoDB mund të mbështesë sasi të mëdha si të të dhënave, ashtu edhe të trafikut.
- Password hashing është për arsye të sigurisë, në mënyrë që përdoruesit e aplikacionit të jenë më të sigurtë, në rast se nese bjen databaza si rezultat i ndonjë sulmi, të dhënat e përdoruesve të jenë të sigurta.
  - Password reset - Në rast se përdoruesi ka harruar fjalëkalimin e tij të mund të e rikthej atë, dhe kjo do të i mundësojë që ai të mos i humb të dhënat e ndryshme për mjekët, komentet të cilat i ka bërë.
  - Votimi i mjekëve nga ana e përdoruesve është bërë për arsye se të jemi në gjendje të dime se cili nga mjekët është më i sukseshëm nga pacientet, dhe të rradhitet më mire në aplikacion kur kërkohet nga përdoruesit.
  - Definimit i top mjekëve sipas departamentit është bërë sepse pacienti me pasë më të lehtë me e pa se cili është mjeku më i mire dhe më i përshtatshëm për nevojat e tij si pacient.

SPRINT	START	FINISH
Database Design	2/01/2021	3/01/2021
Home page and othet front pages	4/01/2021	7/01/2021
Login, Register	8/01/2021	9/01/2021
Database Tables	10/01/2021	11/01/2021
View Doctor List	12/01/2021	14/01/2021
Middlewares backend-frontend	14/01/2021	17/01/2021
Implement Search	18/01/2021	20/01/2021
Email Notifications	21/01/2021	23/01/2021
Doctor Profile	24/01/2021	25/01/2021
Password Reset	26/01/2021	27/01/2021
Google Login	28/01/2021	30/01/2021
Password Hashing	31/01/2021	01/02/2021

Tabela3-Definimi i Sprintave

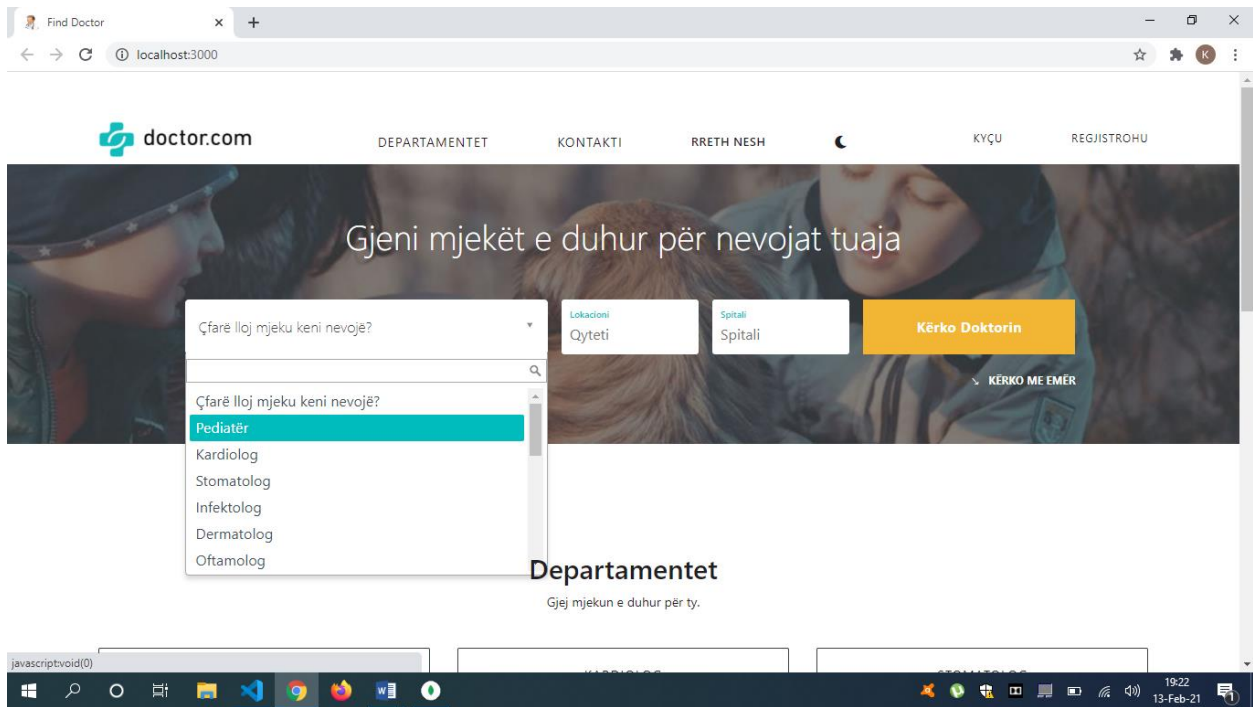


Fig 5.3-Listimi i Doktorave

## 7 REFERENCAT

- [1] Joshua Bloch. Effective Java. third edition
- [2] Cary S.Horstmann. Core Java Volume – Fundamentals
- [3] Brett D. McLaughlin, Gary Pollice & David West. Head First Object-Oriented Analysis Design
- [4] Mario Casciaro, Luciano Mammino. Node.js Design Patterns
- [5] Greg Sidelnikov. Learning React Javascript Library From Scratch
- [6] Callum Macrae. Vue.js Up & Running
- [7] Kyle Banker. MongoDB in Action
- [8] Charles Bell. Introducing the MySQL Document Store
- [9] Tom Marris. JSON at Work
- [10] Marijn Haverbeke. Eloquent Javascript
- [11] Ramiz Hoxha: Slidet, 2019
- [12] Referenca e fotos 2.4.1 – Slidet e prof Ramiz Hoxha
- [13] Referenca e fotos 2.5.1 – Callum Macrae. Vue.js Up & Running
- [14] Referenca e fotos 3.1 – Robin Wieruch. The Road to Learn React
- [15] Referenca e fotos 4.1 – Slider e prof Ramiz Hoxha

## 8 APPENDIXES

### Precedenca e Operatorit Në Java

Java ka rregulla të përcaktuara mirë për specifikimin e radhës në të cilën vlerësohen operatorët në një shprehje kur shprehja ka disa operatorë. Për shembull, shumëzimi dhe pjesëtimi kanë një përparësi më të lartë se mbledhja dhe zbritja. Rregullat e përparësisë mund të mbivendosen nga kllapa të qarta.

Rendi i përparësisë. Kur dy operatorë ndajnë një operator, operatori me përparësi më të lartë shkon i pari. Për shembull,  $1 + 2 * 3$  trajtohet si  $1 + (2 * 3)$ , ndërsa  $1 * 2 + 3$  trajtohet si  $(1 * 2) + 3$  meqenëse shumëzimi ka një përparësi më të lartë se mbledhja.

Shoqërimi. Kur një shprehje ka dy operatorë me të njëjtën përparësi, shprehja vlerësohet sipas shoqërimit të saj. Për shembull  $x = y = z = 17$  trajtohet si  $x = (y = (z = 17))$ , duke lënë të tre variablat me vlerën 17, pasi që operatori = ka shoqërim djathtas-majtas (dhe një deklaratë caktimi vlerëson në vlerën në anën e djathtë). Nga ana tjetër,  $72/2/3$  trajtohet si  $(72/2) / 3$  pasi që operatori / ka shoqërim nga e majta në të djathtë. Disa operatorë nuk janë asociativë: për shembull, shprehjet  $(x <= y <= z)$  dhe  $x ++ -$  janë të pavlefshme.

Precedenca dhe shoqërimi i operatorëve Java. Tabela më poshtë tregon të gjithë operatorët Java nga përparësia më e lartë tek më e ulta, së bashku me shoqërimin e tyre. Shumica e programuesve nuk i mësojnë përmendësh të gjithë, madje edhe ata që përdorin akoma kllapa për qartësi.