

**Faculty of Contemporary Sciences and Technologies**

**Skopje**

**Ambulance Management System**

**Software Engineering**

**Hani Alija**

Computer Science, Southeast European University, Skopje, North Macedonia

**Jasin Ismaili**

Computer Science, Southeast European University, Skopje, North Macedonia

**Uvejs Murtezi**

Computer Science, Southeast European University, Skopje, North Macedonia

**Mentor: Prof. Besnik Selimi**

2024, Skopje

Software Engineering

**ABSTRACT**

**-Ambulance Management System -** është një platformë inovative e zhvilluar për të përmirësuar dhe automatizuar proceset e termineve dhe menaxhimit të shërbimeve. Ky sistem ofron një eksperiencë të lehtë dhe të sigurtë për përdoruesit, duke i mundësuar atyre të caktojne termine, dhe menaxhojne te dhenat e tyre në mënyrë efikase. Pacientet mund të menaxhojne te dhenat e tyre si dhe te caktojne termine duke përdorur një platformë të sigurt dhe të besueshme. Doktoret dhe infermieret mund te menaxhojne me lehte me pacientet e tyre, te menaxhojne me orarin e termineve si dhe te shkruajne raporte per rezultatet e termineve.

**-Ambulance Management System -** gjithashtu përdor teknologji të fundit për të siguruar siguri dhe mbrojtje të të dhënave të përdoruesve. Ai është një hap drejt efikasitetit të lartë, lehtësimit të proceseve dhe përmirësimit të qasjes në shërbime shëndetësore urgjente .Sistemi gjithashtu ofron raporte të detajuara për takimet mes pacientëve dhe doktorëve.

# **TABLE OF CONTENTS**

**ABSTRACT** ……………………………………………………………………………………..2

**1. INTRODUCTION** ……………………………………………………………………..……..4

1.1 PROJECT AIMS AND OBJECTIVES …………………………..…………………..4

**2. SYSTEM ANALYSIS** …………………………………………………………………… . …5

2.1 SOFTWARE REQUIREMENT SPECIFICATION …………………………………. 5

2.1.1 GENERAL DESCRIPTION ………………………………………………………5

2.1.2 SYSTEM OBJECTIVES ………………………………………………………...5

2.1.3 SYSTEM REQUIREMENTS …………………………………………………….5

2.1.3.1 NONFUNCTIONAL REQUIREMENT ………………………………….5

2.1.3.2 FUNCTIONAL REQUIREMENTS ……………………………………...6

2.1.4 SOFTWARE REQUIREMENTS ………………………………………………8

2.2 SOFTWARE TOOL USED …………………………………..………………………9

**3. SYSTEM DESIGN**…………………………………………………………………………..12

3.1 TABLE DESIGN ……………………………………………………………………12

**4. SYSTEM IMPLEMENTATION** …………………………………………………………..15

4.1 SCREENSHOTS ………………………………………………………….………...15

**5. SYSTEM TESTING** ………………………………………………………………………...22

5.1 UNIT TESTING ……………………………………………………………………22

**6. CONCLUSION & FUTURE SCOPE** ………………………………………….…………..23

**7. REFERENCES** …………………………………………………….………………………..25

**CHAPTER 1**

**INTRODUCTION**

**1.1** **PROJECT AIMS AND OBJECTIVES**

Ambulance Management System është një zgjidhje innovative që synon adresimin e joefikasitetit të përhapur në proceset tradicionale administrative brenda mjediseve të kujdesit shëndetësor. Ky dokumentim përshkruan backgroundin, motivimin dhe deklaratën e problemit që nxit zhvillimin dhe zbatimin e këtij sistemi.

**- Backgroundi:** Në fushën e administrimit të kujdesit shëndetësor, mbështetja në procese manuale dhe të bazuara në letër ka sjellë sfida të konsiderueshme. Detyrat administrative që lidhen me menaxhimin e shërbimeve të ambulancës, informacionin e pacientëve, takimet dhe inventorin kanë qenë të prirura për vonesa, gabime dhe vështirësi në mbajtjen e të dhënave të sakta. Ky joefikasitet ndikon në cilësinë e përgjithshme të kujdesit ndaj pacientit dhe optimizimin e burimeve brenda objekteve të kujdesit shëndetësor.

**- Motivimi:** Motivimi i domosdoshëm per Ambulance Management System buron nga nevoja urgjente për të rritur efikasitetin dhe për të ngritur standardin e kujdesit ndaj pacientit. Ky sistem synon të thjeshtojë këto procese për të optimizuar burimet, për të reduktuar gabimet dhe përfundimisht për të përmirësuar cilësinë e shërbimeve të kujdesit shëndetësor që u ofrohen pacientëve.

**Problemi:** Proceset joefikase administrative

**Sfida:** Detyrat administrative manuale dhe të bazuara në letër që çojnë në vonesa, gabime dhe vështirësi në menaxhimin e informacionit të pacientit

**Rëndësia:** Përmirësimi i proceseve administrative është thelbësor për optimizimin e burimeve, reduktimin e gabimeve dhe përfundimisht përmirësimin e cilësisë së përgjithshme të kujdesit ndaj pacientit.

**- Përmbledhje e zgjidhjes:** Ambulance Management System ofron një qasje të integruar dhe të automatizuar për menaxhimin e shërbimeve të ambulancës dhe detyrave administrative të lidhura me to. Duke përdorur një aplikacion të bazuar në ueb, kjo zgjidhje centralizon administrimin e ambulancës, duke automatizuar detyrat kritike si vendosja e termineve, menaxhimi i të dhënave të pacientëve dhe mirëmbajtja e të dhënave të mjekëve dhe infermierëve.

**CHAPTER 2**

**SYSTEM ANALYSIS**

Në këtë kapitull, ne do të diskutojmë dhe analizojmë në lidhje me procesin e zhvillimit të Sistemit të Menaxhimit të Ambulances, duke përfshirë specifikimin e kërkesave të softuerit (SRS) dhe krahasimin midis sistemit ekzistues dhe atij të propozuar. Kërkesat funksionale dhe jo funksionale janë përfshirë në pjesën SRS për të siguruar përshkrimin e plotë dhe pasqyrën e përgjithshme të kërkesave të sistemit para se të kryhet procesi i zhvillimit.

**2.1 SOFTWARE REQUIREMENT SPECIFICATION**

**2.1.1 GENERAL DESCRIPTION**

Ambulance Management System është një zgjidhje innovative që synon adresimin e joefikasitetit të përhapur në proceset tradicionale administrative brenda mjediseve të kujdesit shëndetësor.

**2.1.2 SYSTEM OBJECTIVES**

* T’u ofrojme pacienteve nje sistem me te avancuar shendetesor
* T’u ofrojme punonjesve te menaxhojne me lehte te dhenat e pacienteve si dhe terminet.
* Nje databaze me e sigurte per ruajtjen e te dhenave te pacienteve dhe punonjesve.
* Menaxhim i lehte i takimeve mes pacienteve dhe doktorave si dhe shkruajtja e reporteve.

**2.1.3 SYSTEM REQUIREMENTS**

**2.1.3.1 NON FUNCTIONAL REQUIREMENTS**

* Kur do të implementohet një sistem i menaxhimit të ambulances, përdoruesi do të arrijnë qe te caktojne termine ne kohe shume te shpejte.
* Sistemi (administratori) do të kryejë me saktësi regjistrimin e doktoreve, infermiereve dhe pacienteve.
* Sistemi është krijuar për një mjedis miqësor për përdoruesit, në mënyrë që pacientet dhe doktoret(infermieret) të kryejnë detyra të ndryshme lehtësisht dhe në një mënyrë efektive.
* Në implementimin e tërë sistemit eshte perdorur html, css (bootstrap), javascript në pjesën e përparme me php (laravel) si server gjuhë skriptimi e cila do të përdoret për lidhjen e bazës së të dhënave dhe backend dmth pjesa e bazës së të dhënave zhvillohet duke përdorur mysql.

**2.1.3.2 FUNCTIONAL REQUIREMENTS**

1. **User Register (Admin)**

* Register: Vetem Admini do te kete qasje per te bere regjistrimin e: pacienteve, doktoreve dhe infermiereve). Per te bere regjistrimin do te futen te dhenat si: numri ame, emri, emaili, passwordi dhe tjera informacione)

*Kërkesat funksionale*

* Sistemi duhet të jetë në gjendje të kontrolloje informacionin
* Sistemi duhet të jetë në gjendje të mos lejojë dy paciente, infermiere ose doktore te regjistrohen me emailin dhe numrin ame te njejte
* Sistemi do ti jape ID (auto increment) perdoruesve qe do i regjistron admini.

1. **User Login (Patient, Nurse, Doctor, Admin)**

* Login: Kjo feature e përdorur nga perdoruesit për t'u identifikuar në sistem. Atyre u kërkohet të futin email-in dhe fjalëkalimin përpara se të lejohen të hyjnë në sistem. Email dhe fjalëkalimi i përdoruesit do të verifikohen dhe nëse identifikimi është i pavlefshëm përdoruesi nuk lejohet të hyjë në sistem.

*Kërkesat funksionale*

* Sistemi duhet të jetë në gjendje të verifikoje informacionin
* Sistemi duhet të lejojë që përdoruesi me email dhe fjalëkalim të vlefshëm të hyjë në sistem
* Përdoruesi duhet të jetë në gjendje të beje logout pasi të ketë përfunduar përdorimin e sistemit.

1. **Make an appointment (Patient, Nurses, Doctor)**

* Make an appointment: Nje feature per te caktuar termine si nga ana e pacienteve po edhe nga ana e doktoreve dhe infermiereve. Kerkon te futen te dhena si Id e pacientit, Id e doktorit, data, koha etj.

*Kërkesat funksionale*

* Sistemi duhet të jetë në gjendje të verifikoje informacionin
* Sistemi nuk duhet te lejoje dy appointment te krijohen me Id te njejte.

1. **Write a report(Doctor, Nurse)**

* Write a report: Doktoret dhe infermieret mund te shenojne raporte per takimet qe kane pasur me paciente, ndersa nga ana tjeter, pacientet mund te lexojne vetem per raportet e tyre. Kerkon te dhena si Id e terminit, Id e doktorit etj.

*Kërkesat funksionale*

* Sistemi duhet të jetë në gjendje të verifikoje informacionin
* Sistemi nuk duhet te lejoje dy raporte krijohen me Id te njejte.
* Sistemi nuk duhet te lejoje qe pacienti te shkruan raport.
* Sistemi duhet automatikisht ta fshije raportin nese termini me te cilin eshte i lidhur fshihet.

1. **Edit and Delete Users: ()**

* Edit: Secili User mund te beje edit vetem te dhenat e veta.
* Delete: Admini mund te fshije te gjithe perdoruesit, infermieret mund te fshijne pacientet dhe veten, doktoret dhe pacientet mund te fshijne vetem veten.

*Kërkesat funksionale*

* Sistemi nuk duhet te lejoje perdoruesit te editojne te dhenat e perdoruesve tjere.
* Sistemi nuk duhet te lejoje qe doktori te fshije paciente.
* Sistemi duhet ta fshije perdoruesin nese ai e ben delete accountin e vet.

**2.1.4 SOFTWARE REQUIREMENTS**

* Operating system- Windows 11 përdoret si sistem operativ pasi është i qëndrueshëm dhe mbështet më shumë karakteristika dhe është më miqësor për përdoruesit
* Database MYSQL-MYSQL përdoret si bazë e të dhënave, sepse është e lehtë për të ruajtur dhe marrë rekorde nga query të thjeshta, të cilat janë në gjuhën angleze, të cilat kuptohen lehtë dhe shkruhen lehtë.
* Development tools and Programming language- HTML përdoret për të shkruar të gjithë kodin dhe për të zhvilluar faqe në internet me CSS (Bootstrap) dhe javascript për punën e stilimit dhe php (Laravel) për skriptimin nga ana e serverit.

**2.2 SOFTWARE TOOLS USED**

* **HTML** - Hyper Text Markup Language është gjuha kryesore e shënjimit për krijimin e faqeve të internetit dhe informacioneve të tjera që mund të shfaqen në një shfletues uebi. HTML është shkruar në formën e elementeve HTML që përbëhet nga etiketa të mbyllura në kllapa këndi brenda përmbajtjes së faqes në internet. Etiketat HTML zakonisht vijnë në çifte si dhe, megjithëse disa etiketime përfaqësojnë elementë të zbrazët dhe kështu që janë të pa çiftuara, për shembull Error! Emri i skedarit nuk është specifikuar. Etiketimi i parë në një çift është etiketa startuese dhe etiketa e dytë është etiketa fundore (ato quhen gjithashtu etiketat e hapjes dhe etiketat mbyllëse). Në mes të këtyre etiketave, krijuesit e faqes në internet mund të shtojnë tekst, etiketa të mëtejshme, komente dhe lloje të tjerë të përmbajtjes së bazuar në tekst. Qëllimi i një shfletuesi në internet është të lexojë dokumente HTML dhe t'i kompozojë ato në faqe të dukshme ose të dëgjueshme. Shfletuesi nuk shfaq etiketat HTML, por përdor etiketat për të interpretuar përmbajtjen e faqes. Elementet HTML formojnë blloqet ndërtuese të të gjitha faqeve të internetit. HTML lejon që imazhet dhe objektet të nguliten dhe mund të përdoren për të krijuar forma ndërvepruese. Ai siguron një mjet për të krijuar dokumente të strukturuara duke treguar semantikën strukturore për tekstin siç janë tituj, paragrafë, lista, lidhje, citate dhe artikuj të tjerë. Mund të vendos skripte të shkruara në gjuhë të tilla si JavaScript të cilat ndikojnë në sjelljen e faqeve të HTML.

* **CSS (Bootstrap)**- Cascading Style Sheets(CSS) është një gjuhë e stilit që përdoret për përshkrimin e pamjes dhe formatimin e një dokumenti të shkruar në një gjuhë shënimi. Ndërsa përdoret më shpesh për të stiluar faqet e internetit dhe ndërfaqet e shkruara në HTML dhe XHTML, gjuha mund të zbatohet për çdo lloj dokumenti XML, duke përfshirë XML të thjeshtë, SVG dhe XUL. CSS është një specifikim i gurthemelit të uebit dhe pothuajse të gjitha faqet e internetit përdorin fletët e stilit CSS për të përshkruar prezantimin e tyre. CSS është krijuar kryesisht për të mundësuar ndarjen e përmbajtjes së dokumentit nga prezantimi i dokumentit, duke përfshirë elementë të tillë si paraqitja, ngjyrat dhe shkronjat. Kjo ndarje mund të përmirësojë aksesueshmërinë e përmbajtjes, të sigurojë më shumë fleksibilitet dhe kontroll në specifikim.

* **JAVA SCRIPT-** JavaScript(JS) është një gjuhë dinamike e programimit kompjuterik. Përdoret më së shpeshti si pjesë e shfletuesve të internetit, implementimet e të cilave lejojnë që skriptet e klientëve të ndërveprojnë me përdoruesin, të kontrollojnë shfletuesin, të komunikojnë në mënyrë asinkrone dhe të ndryshojnë përmbajtjen e dokumentit që shfaqet. Po përdoret gjithashtu në programimin nga ana e serverit, zhvillimin e lojërave dhe krijimin e aplikacioneve desktop dhe celular. JavaScript është një gjuhë skriptimi e bazuar në prototip me shtypje dinamike dhe ka funksione të klasit të parë. Sintaksa e saj u ndikua nga C. JavaScript kopjon shumë emra dhe konventa emërtimi nga Java, por të dy gjuhët përndryshe nuk kanë lidhje dhe kanë semantikë shumë të ndryshme. Parimet kryesore të dizajnit brenda JavaScript janë marrë nga gjuhët e programimit Self and Scheme.

* **PHP**- PHP është një gjuhë skriptimi nga ana e serverit e krijuar për zhvillimin e faqes në internet por përdoret gjithashtu si një gjuhë programimi me qëllim të përgjithshëm. PHP tani është instaluar në më shumë se 244 milion faqe në internet dhe 2.1 milion servera në internet. Krijuar fillimisht nga 15 Rasmus Lerdorf në 1995, zbatimi i referencës i PHP tani prodhohet nga The PHP Group. Ndërsa PHP fillimisht qëndronte për Personal Home Page, tani qëndron për PHP: Hypertext Preprocessor, një backronym rekursiv. Kodi PHP interpretohet nga një server me një modul të procesorit PHP, i cili gjeneron faqen që rezulton në internet: komandat PHP mund të nguliten direkt në një dokument burim HTML në vend që të thërrasin një skedar të jashtëm për të përpunuar të dhëna. Ai gjithashtu ka evoluar për të përfshirë një aftësi ndërfaqe të linjës komanduese dhe mund të përdoret në aplikacione grafike të pavarura. PHP është softuer falas i lëshuar nën Licencën PHP. PHP mund të vendoset në shumicën e serverave të internetit dhe gjithashtu si një shell i pavarur në pothuajse çdo sistem operativ dhe platformë, pa pagesë.
* **Laravel-** Laravel është një kornizë ueb PHP pa pagesë dhe me burim të hapur, e krijuar nga Taylor Otwell dhe e destinuar për zhvillimin e aplikacioneve në ueb duke ndjekur modelin arkitekturor model-view-controller (MVC) dhe i bazuar në Symfony. Disa nga veçoritë e Laravel janë një sistem paketimi modular me një menaxher të dedikuar varësie, mënyra të ndryshme për të hyrë në bazat e të dhënave relacionale, shërbime që ndihmojnë në vendosjen dhe mirëmbajtjen e aplikacionit.
* **MYSQL**- MySQL është (që nga korriku 2013) sistemi i dytë më i përdorur në botë i menaxhimit të bazës së të dhënave relacionale me burim të hapur (RDBMS). Emri i saj është vajza e bashkëthemeluesit Michael Widenius, My. Fraza SQL qëndron për Gjuhën e Strukturuar të Pyetjeve. Projekti i zhvillimit të MySQL e ka vënë në dispozicion kodin e tij burimor nën kushtet e Licencës së Përgjithshme Publike GNU, si dhe nën një mori marrëveshjesh pronësore. MySQL ishte në pronësi dhe sponsorizuar nga një firmë e vetme fitimprurëse, kompania suedeze MySQL AB, tani në pronësi të Oracle Corporation. MySQL është një zgjedhje popullore e bazës së të dhënave për përdorim në aplikacione në internet, dhe është një komponent qendror i burimit të hapur LAMP të përdorur gjerësisht pirg softuerësh për aplikime në internet (dhe pirgje të tjera 'AMP'). LAMP është një akronim për "Linux, Apache, MySQL, Perl / PHP / Python". Projektet me softuer të lirë me burim të hapur që kërkojnë një sistem të plotë menaxhimi të bazës së të dhënave shpesh përdorin MySQL. Për përdorim komercial, janë në dispozicion disa botime me pagesë dhe ofrojnë funksionalitete shtesë.

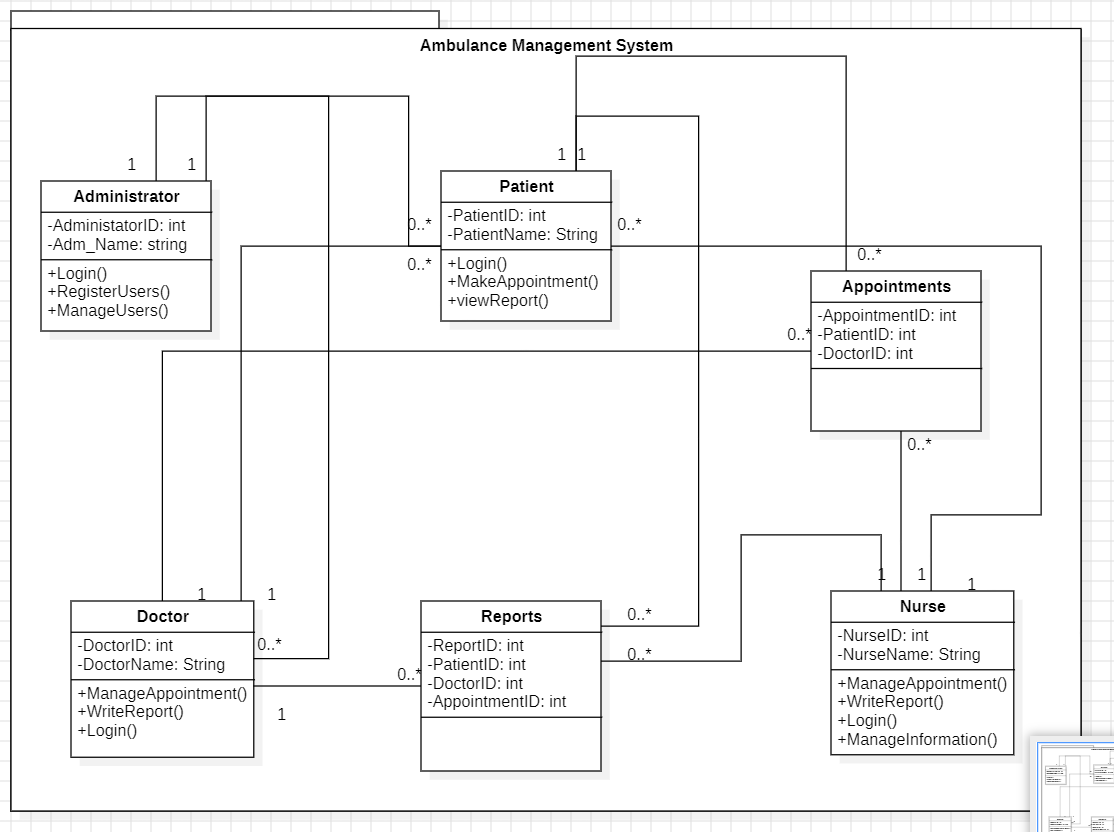
**CHAPTER 3**

**SYSTEM DESIGN**

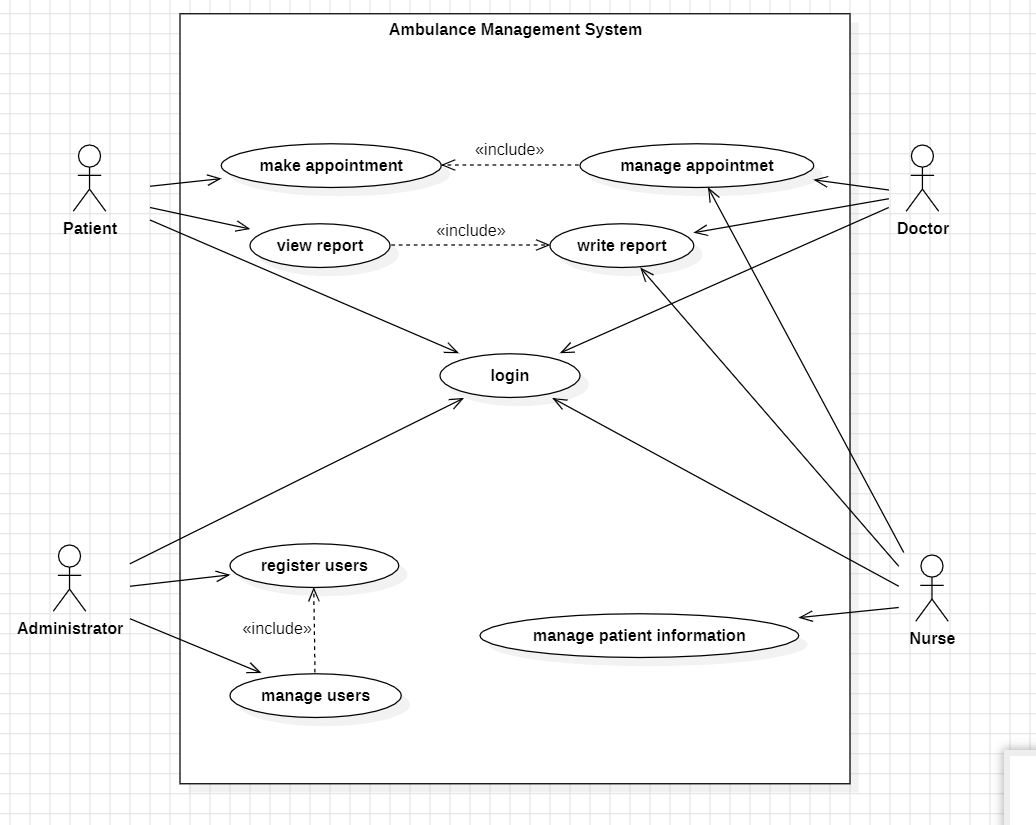
**3.1 Plani Akcional**

| **VEPRIMET** | **PERGJEGJSIT** | **KOHEZGJATJA** | **RESURSET** | **PENGESAT** | **REZULTATET** |
| --- | --- | --- | --- | --- | --- |
| **HAPI 1:**Analiza e kërkimit të informatave për sistemin që do realizojmë | Ne | 1 javë | Ne | Materiale te pakta per ndihmese | Një sistem që do jetë shumë fleksibil dhe do përmbush kërkesat e perdoruesve |
| **HAPI 2:** Dizajnimi sistemit | Ne | 1 javë | Ne | // | Nëse mendimi i dizajnimit del i suksesshëm atëher kalojme në hapin e kodimit |
| **Hapi 3:**Kodimi i sistemit | Ne | 1 javë | Ne | Njohurite e kufizuara | Arritje për testim të sistemit |
| **Hapi 4:**Testimi i sitemit | Ne | 1 javë | Ne | // | Arritje e qëllimit për sistemin e planifikuar |
| **Hapi 5:**Promovimi/Reklamimi i sistemit | Ne | Në të ardhmen | Ne | Kushtet e tanishme | Do të mundohet të bëhet promovimi |

**3.2 Class Diagram**

****

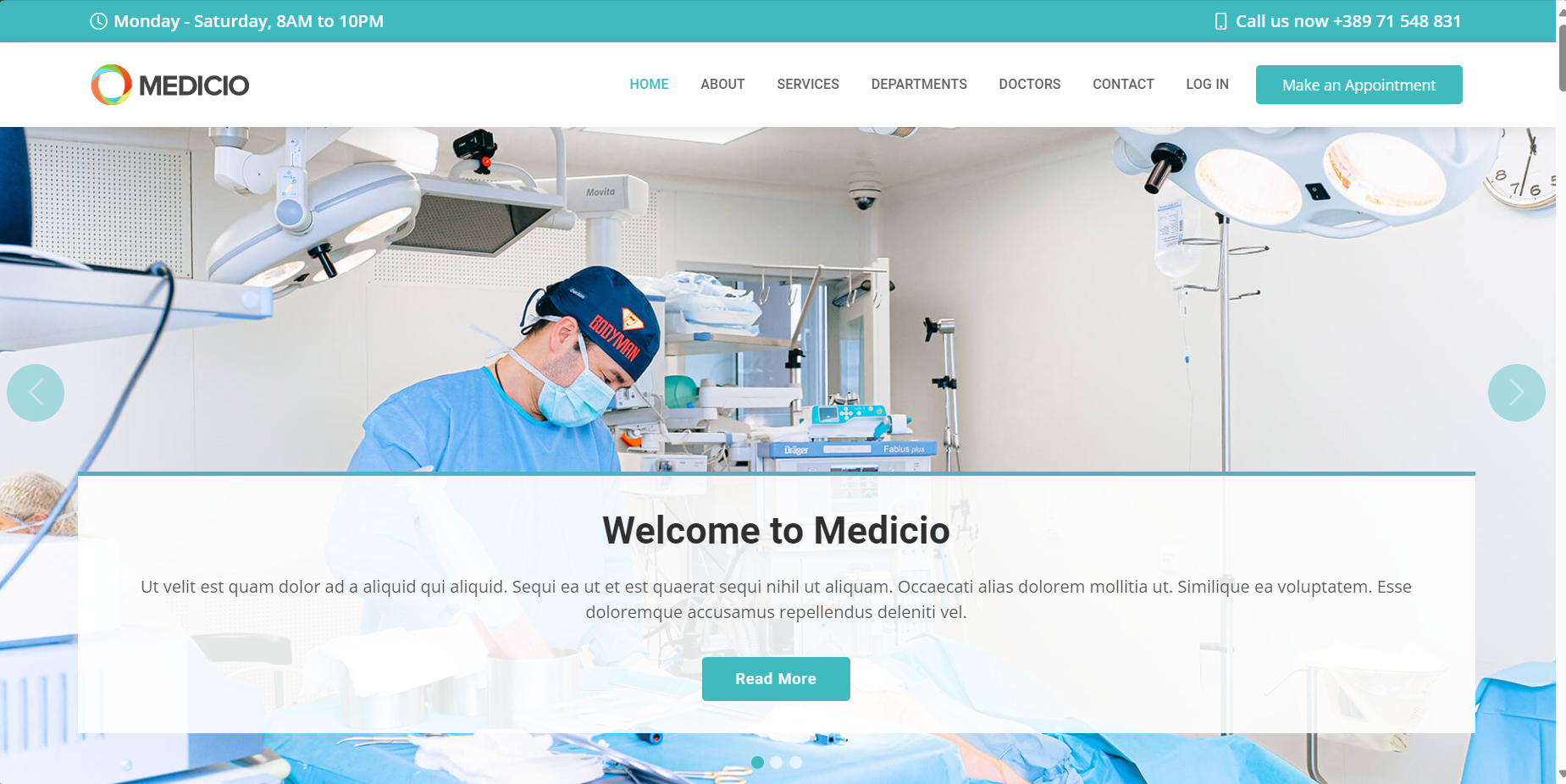
**3.3 Use Case Diagram**

****

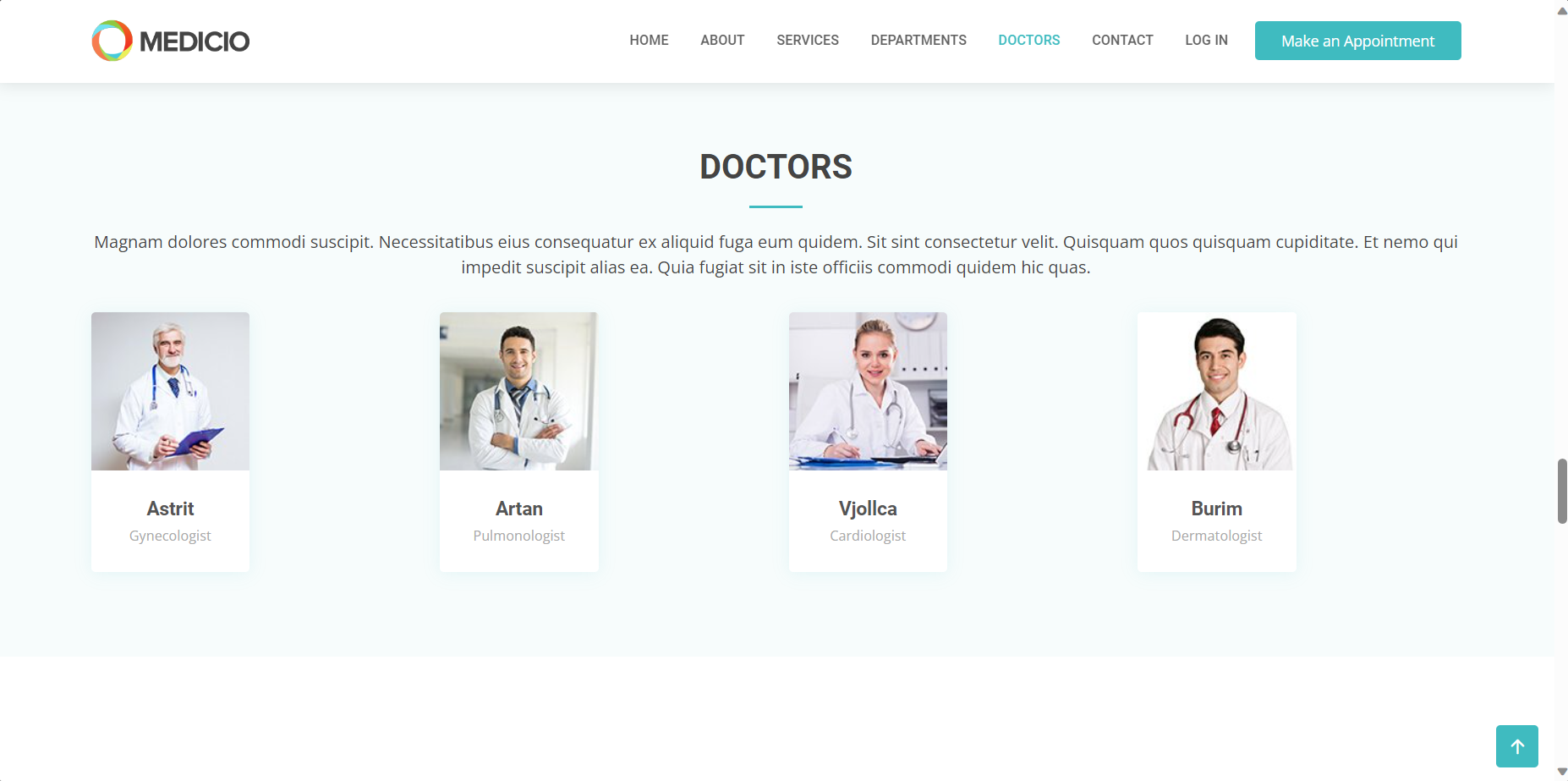
**CHAPTER 4**

**SYSTEM IMPLEMENTATION**

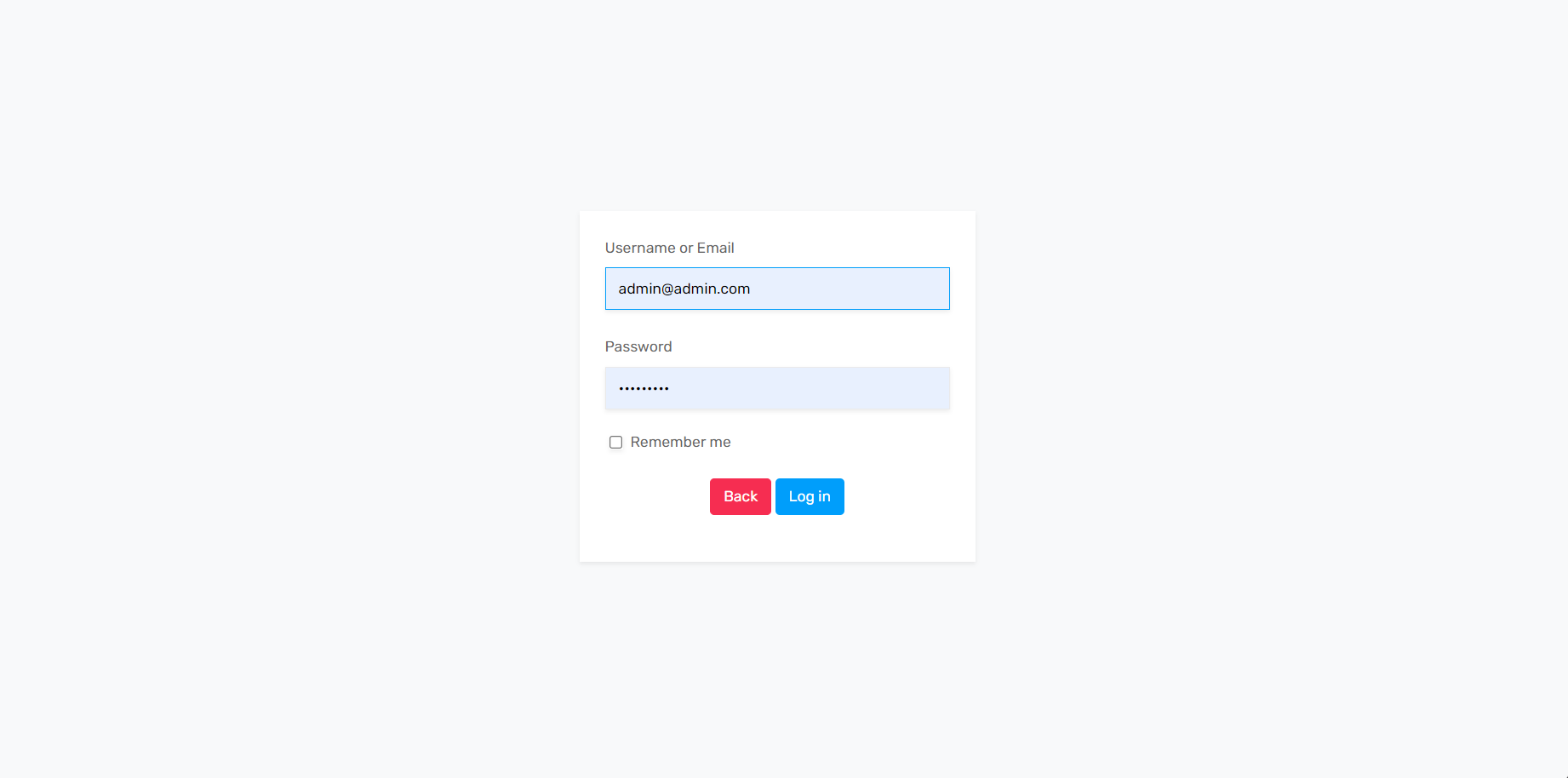
**4.1 Screenshot for welcome page**

****

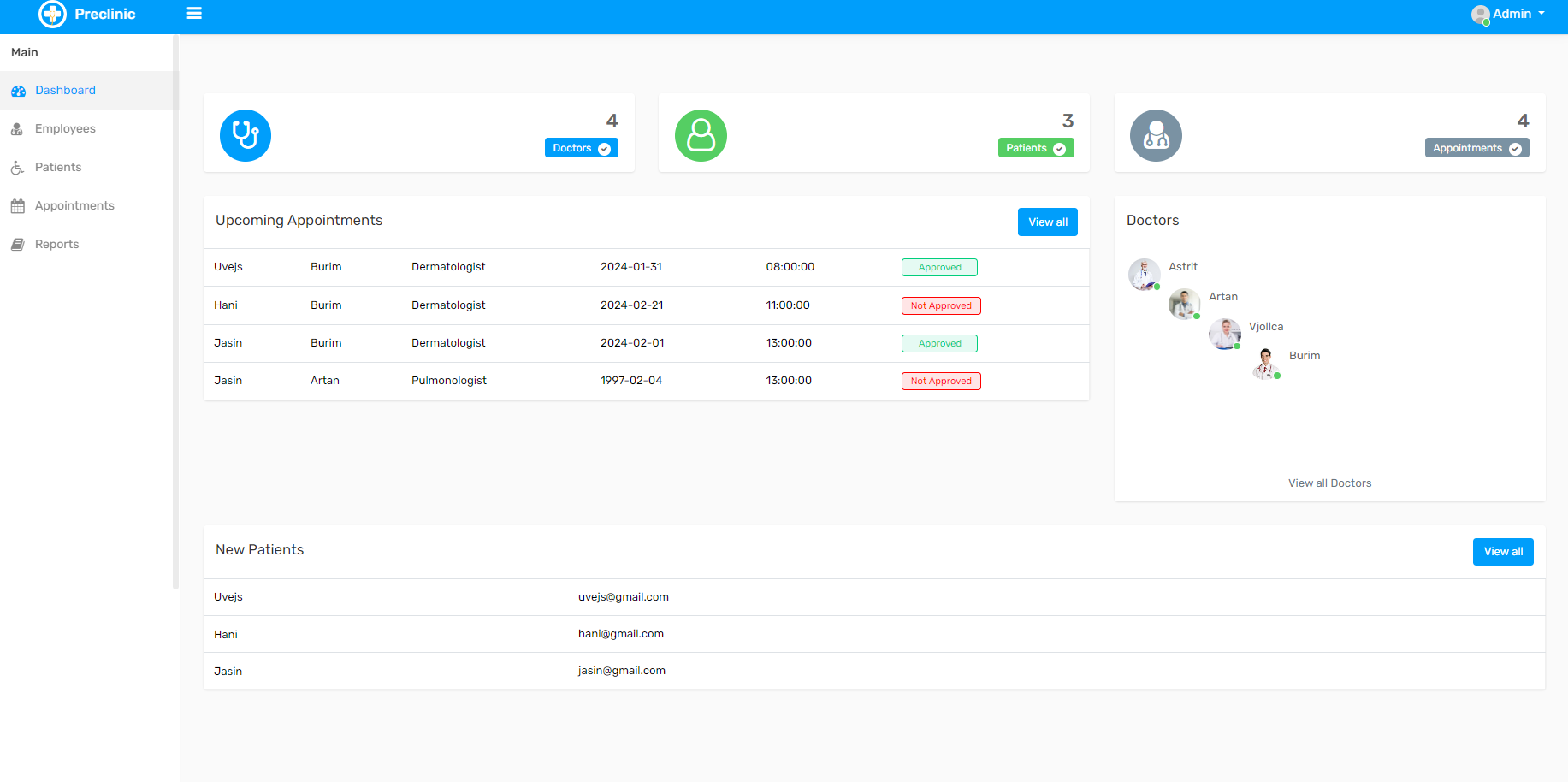
**4.2 Screenshot for welcome page -> doctors**

****

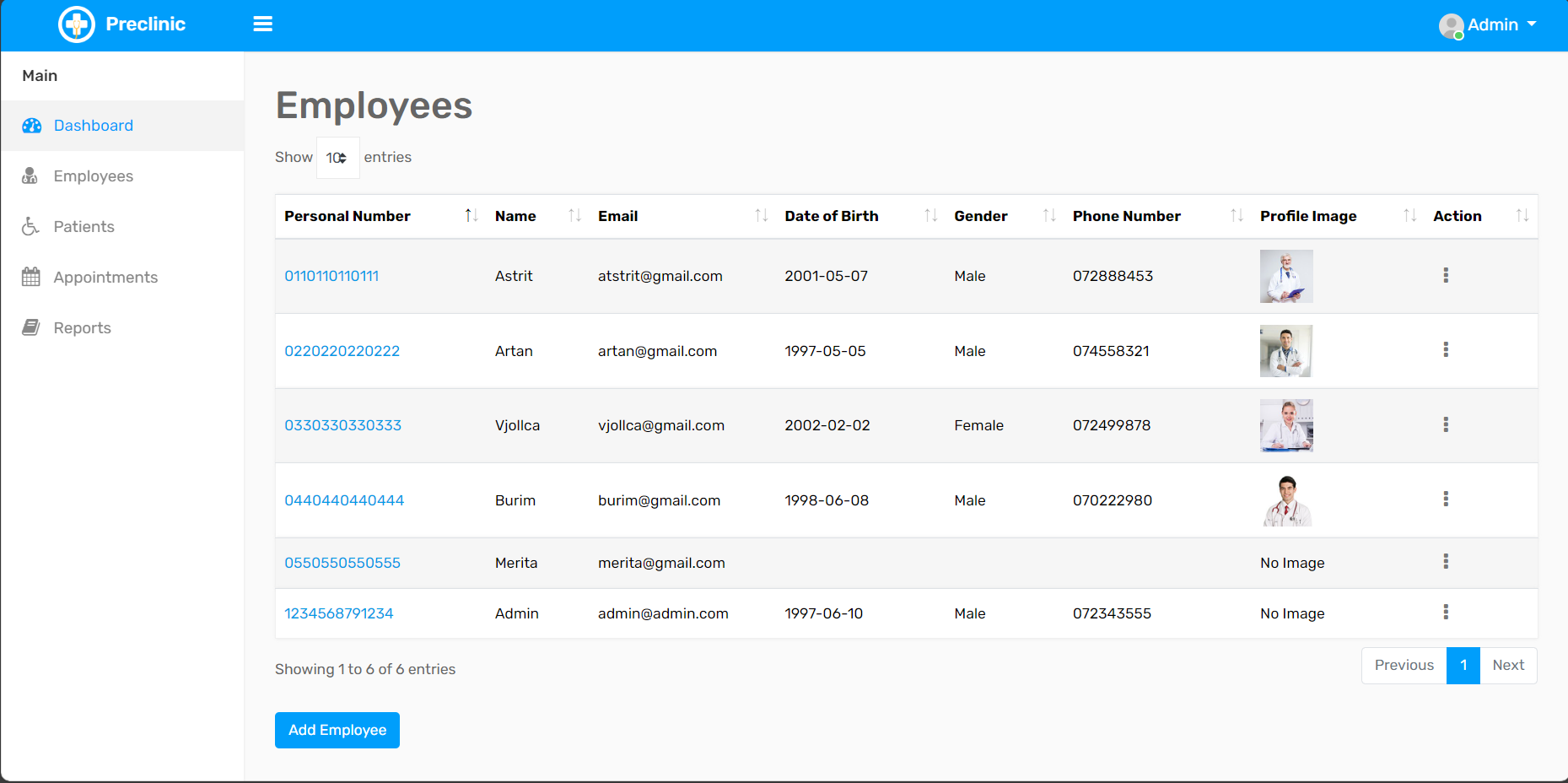
**4.3 Screenshot for login**

****

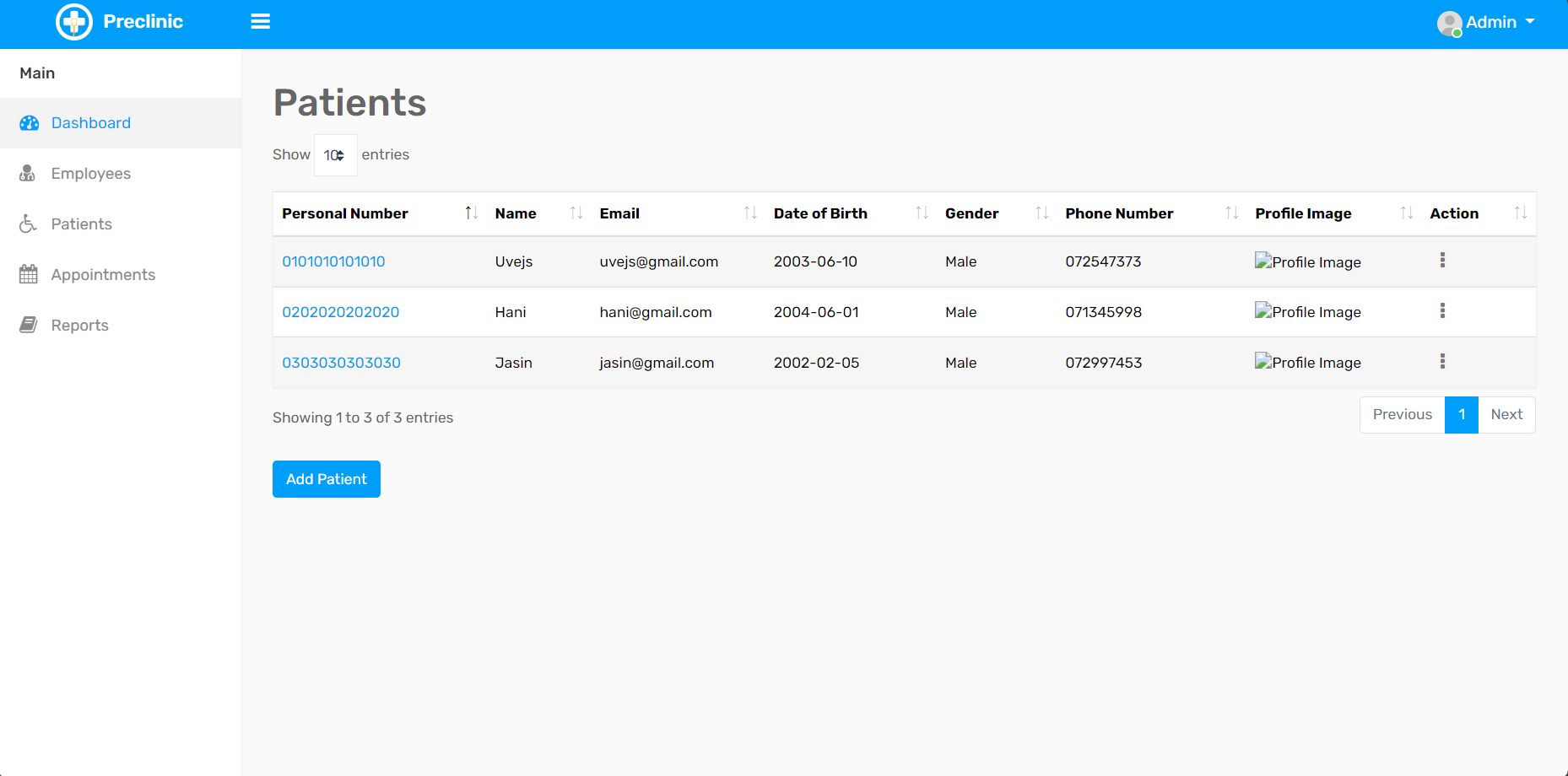
**4.4 Screenshot for admin dashboard**

****

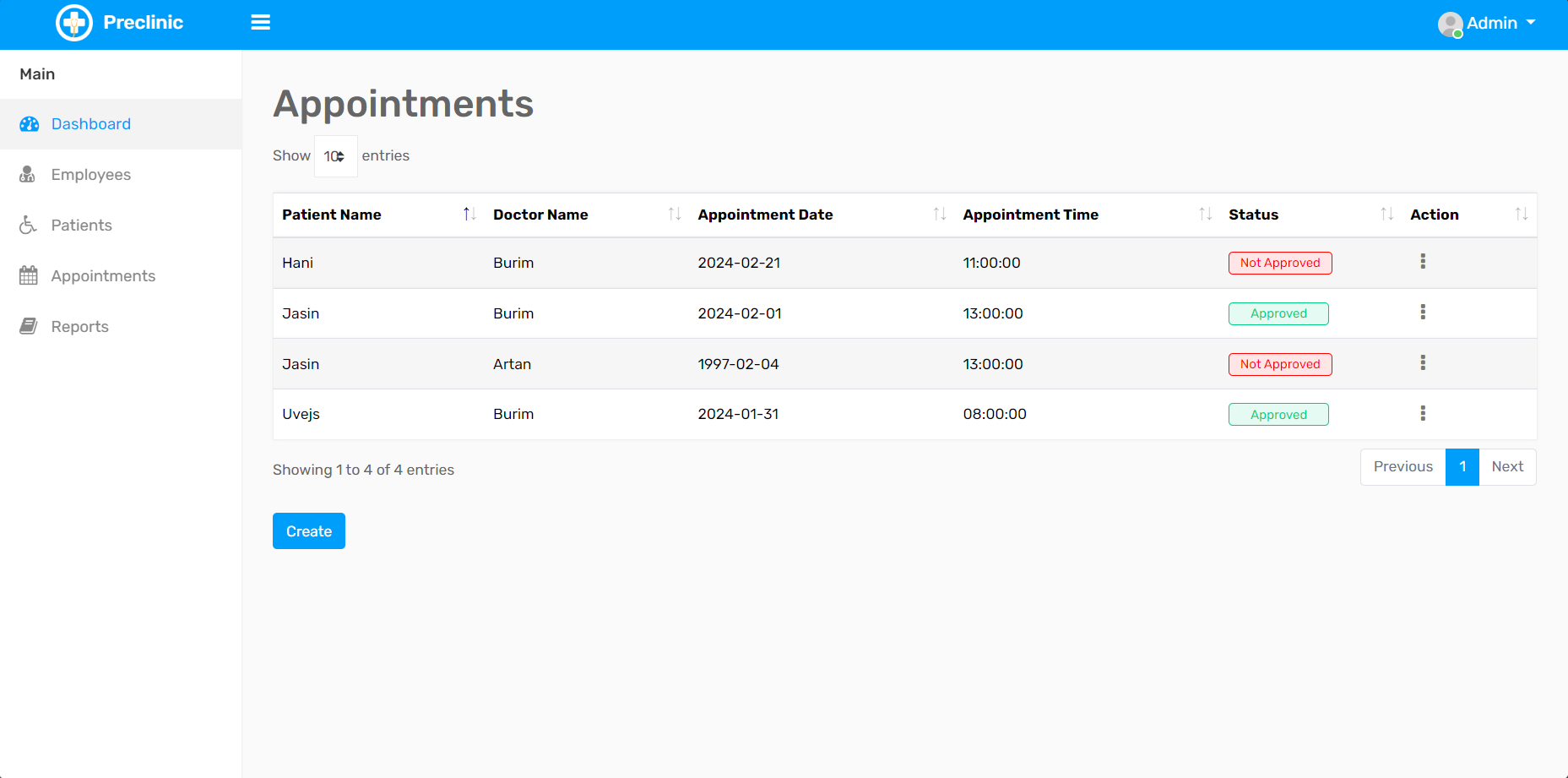
**4.5 Screenshot for list of employees**

****

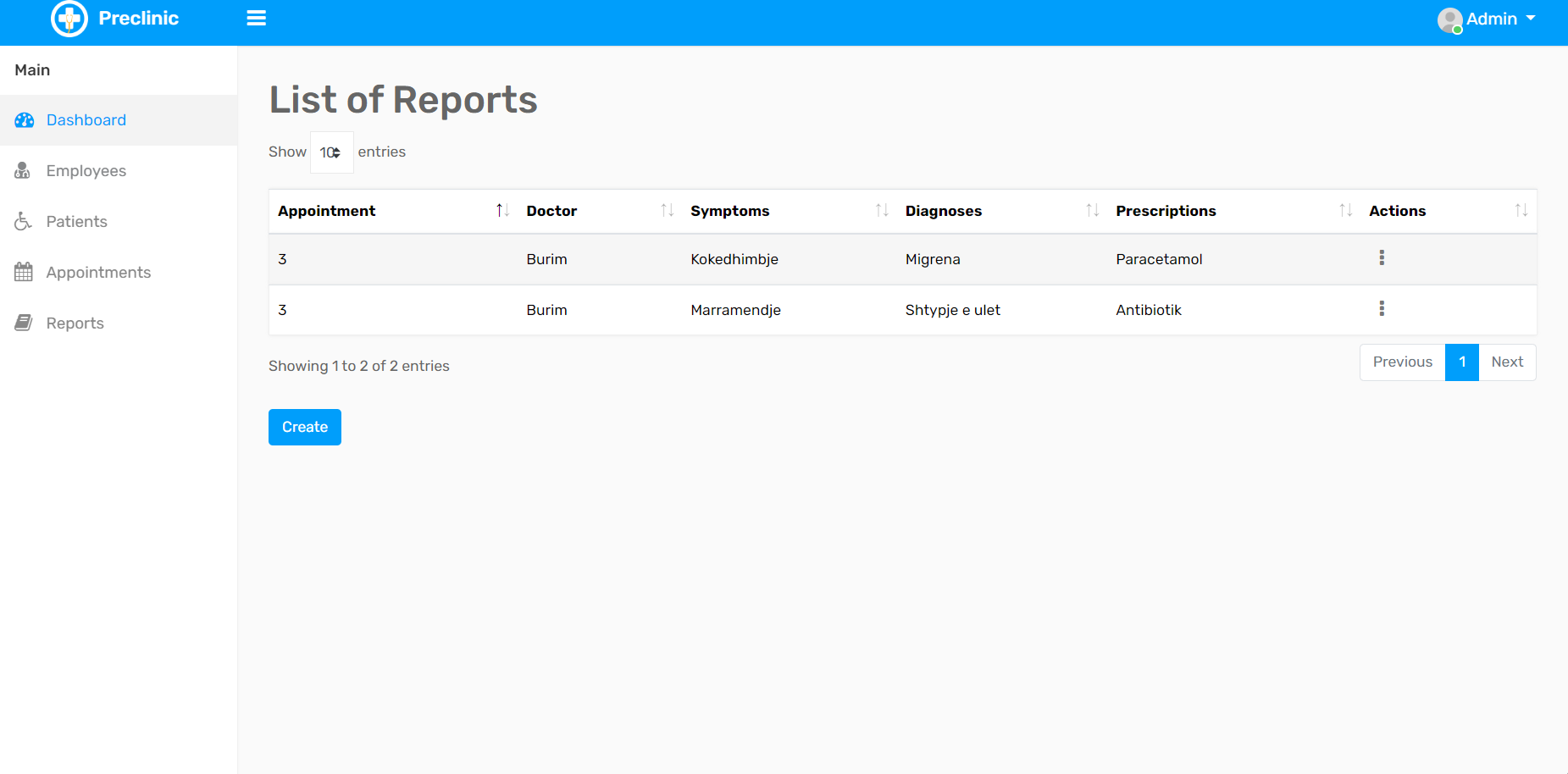
**4.6 Screenshot for list of patients**

****

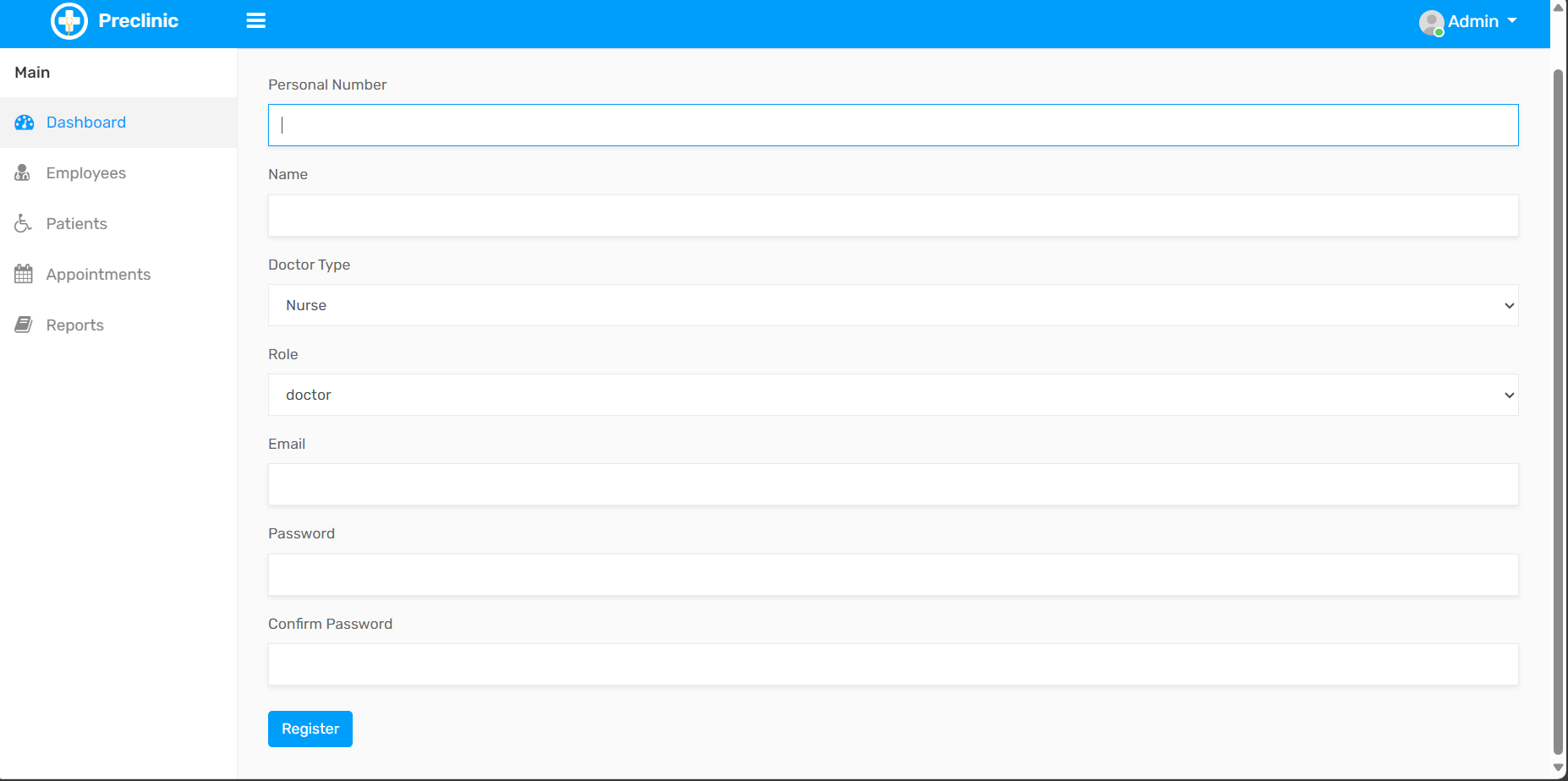
**4.7 Screenshot for list of appointments**

****

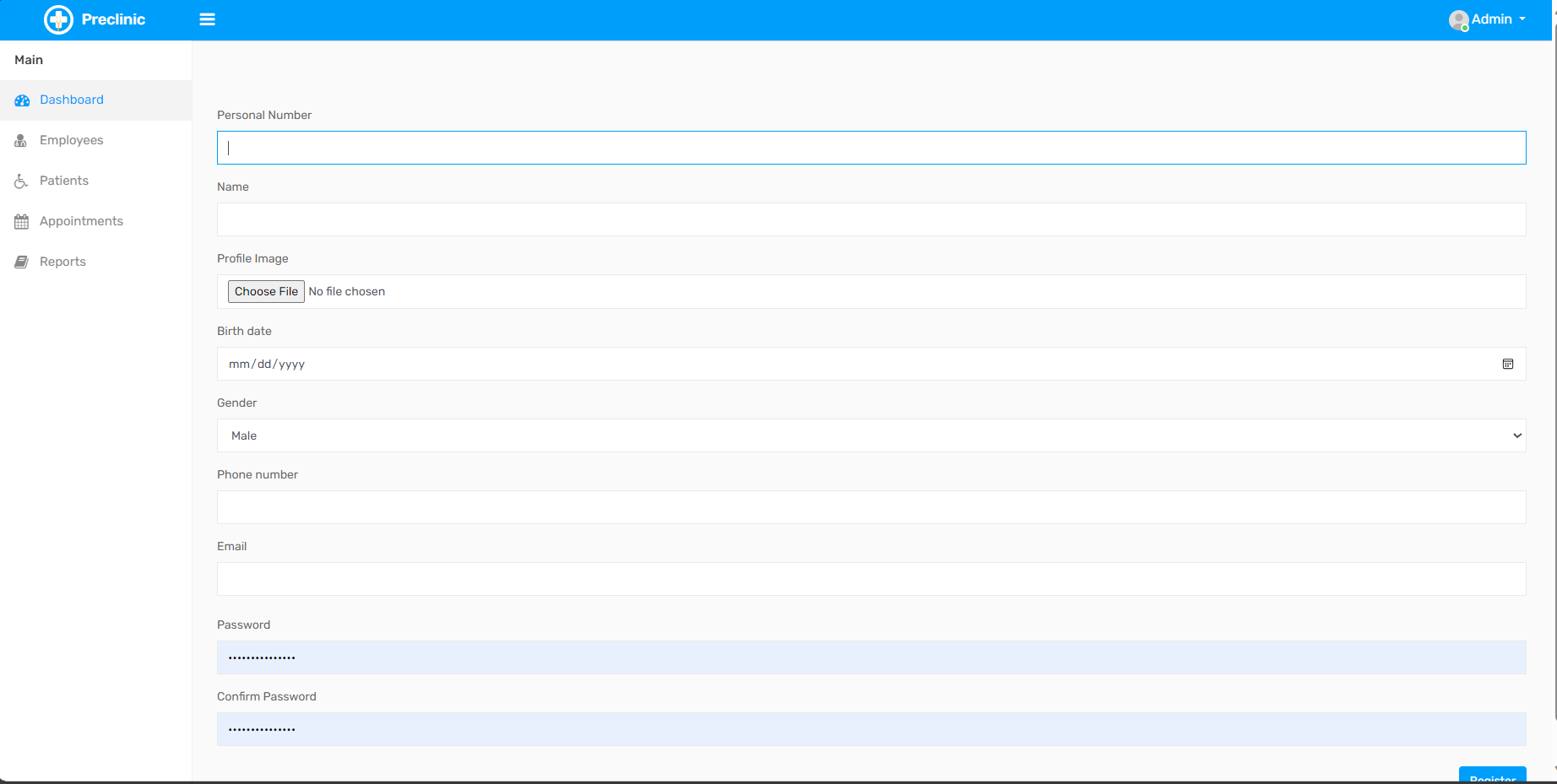
**4.8 Screenshot for list of reports**

****

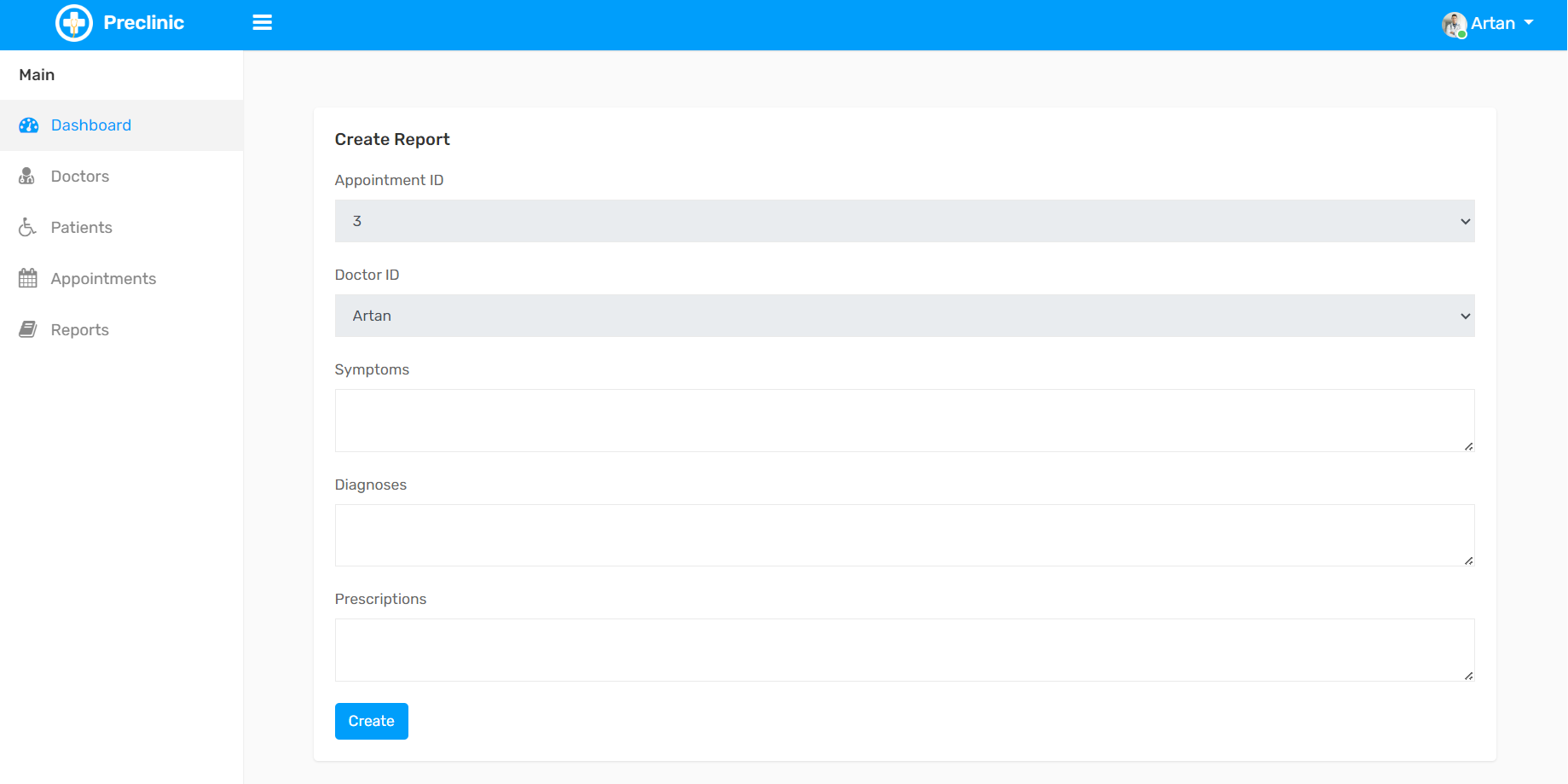
**4.9 Screenshot for admin register employee**

****

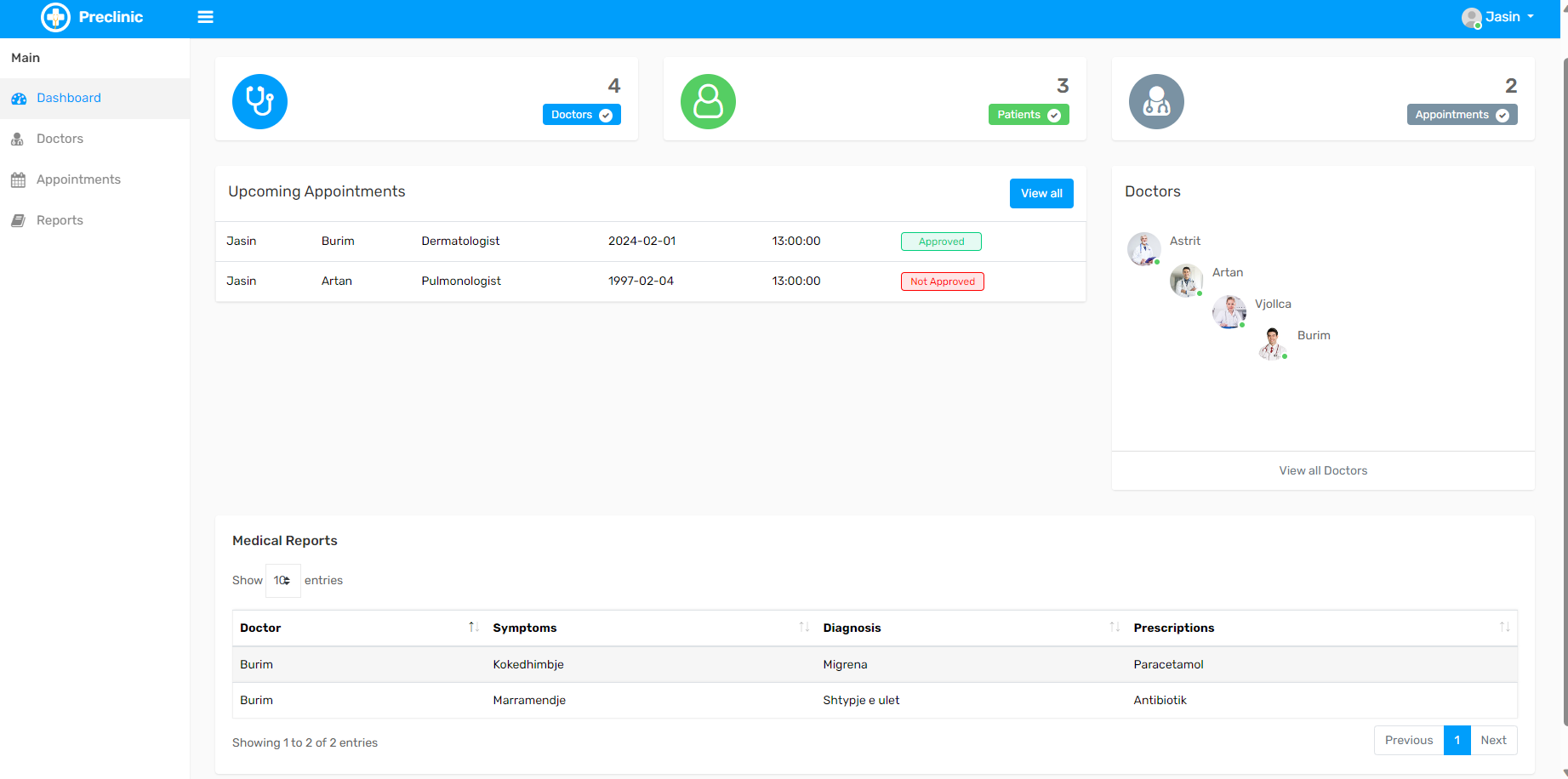
**4.10 Screenshot for register patient**

****

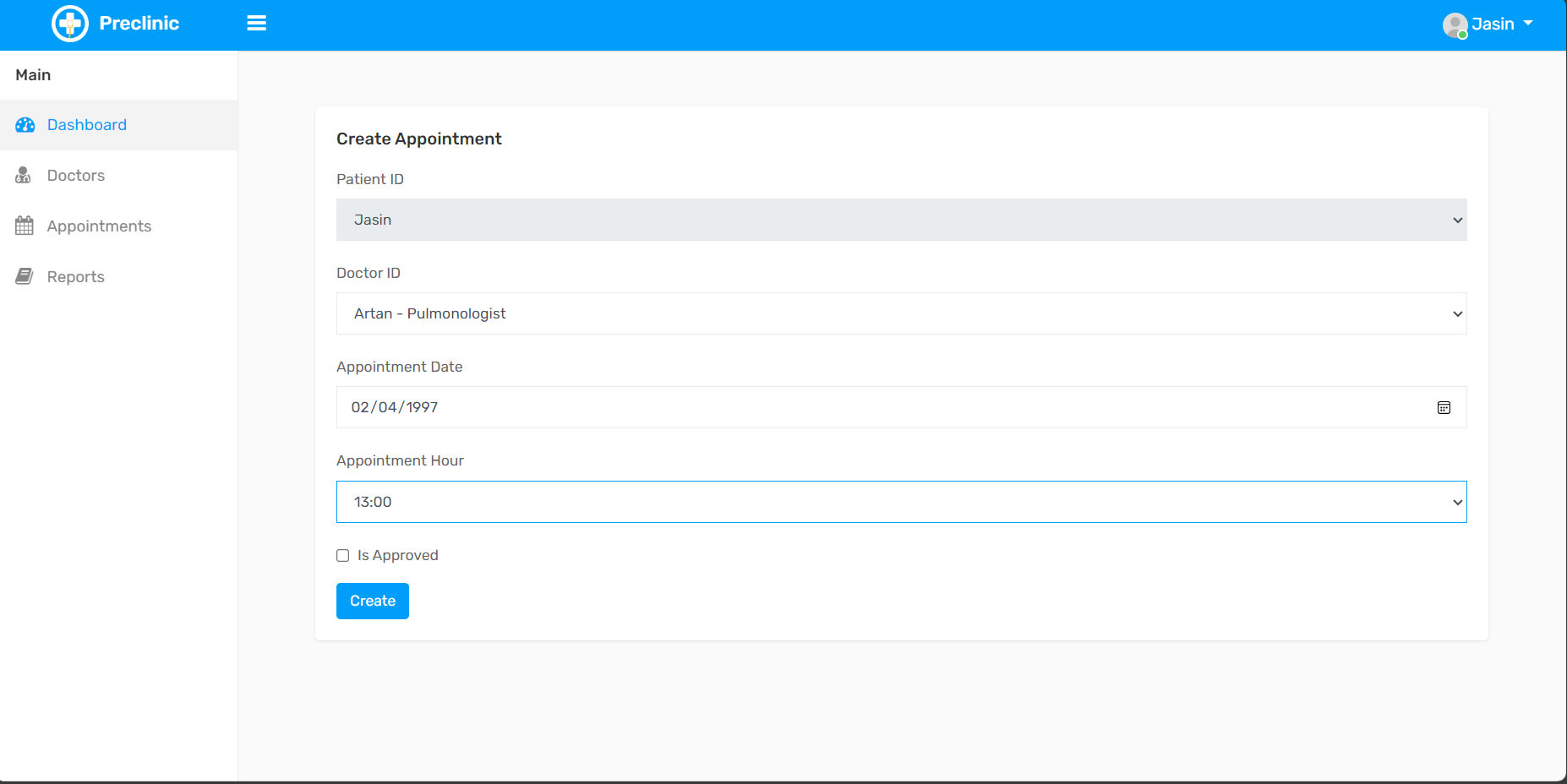
**4.11 Screenshot for doctor write report**

****

**4.12 Screenshot for patient dashboard**

****

**4.13 Screenshot for patient create appointment**

****

**CHAPTER 5**

**SYSTEM TESTING**

Qëllimi i procesit të testimit të sistemit ishte të përcaktojë të gjitha defektet në projektin tonë. Programi iu nënshtrua një sërë inputesh të testit dhe u bënë vëzhgime të ndryshme dhe bazuar në këto vëzhgime do të vendoset nëse programi sillet ashtu siç pritej apo jo.

*UNIT TESTING*

Testimi i njësisë ndërmerret kur një modul është krijuar me sukses. Në mënyrë që të testojmë një modul të vetëm, ne duhet të sigurojmë një mjedis të plotë.

* ***Admin register patient***

Ky test Laravel PHPUnit kontrollon nëse një admin mund të regjistroje me sukses nje pacient. Ai përfshin vendosjen e të dhënave të rreme të pacientit, dërgimin e një kërkese regjistrimi POST dhe kryerjen e pohimeve për të siguruar një regjistrim të suksesshëm. Testi verifikon ridrejtimin pas regjistrimit, ekzistencën e përdoruesit në bazën e të dhënave, caktimin e rolit të 'pacientit' dhe praninë e imazhit të profilit të përdoruesit në ruajtjen publike.

* ***Doctor create appointment***

Ky test Laravel PHPUnit siguron që një mjek, pas hyrjes me sukses ne website, mund të krijojë një takim me një pacient. Ai cakton rolet "mjeku" dhe "pacienti", krijon përdoruesit përkatës dhe regjistrohet si mjek. Testi më pas dërgon një kërkesë POST për të krijuar një takim dhe pohon statusin e përgjigjes, mesazhin e suksesit të seancës dhe praninë e takimit në bazën e të dhënave.

* ***Doctor create report***

Ky test Laravel PHPUnit kontrollon që një mjek mund të krijojë me sukses një raport mjekësor. Ai krijon një skenar me përdorues të rremë të mjekëve dhe pacientëve, regjistrohet si mjek, krijon një takim dhe më pas paraqet një kërkesë POST për të ruajtur një raport mjekësor. Pohimet verifikojnë që përgjigja ridrejton te '/reports' dhe se hyrja përkatëse e bazës së të dhënave përmban informacionin e pritur.

**CHAPTER 6**

**CONCLUSION & FUTURE SCOPE**

Sistemi i Menaxhimit të Ambulancës përfaqëson një hap të rëndësishëm përpara në adresimin e joefikasiteteve të qenësishme në proceset administrative tradicionale të kujdesit shëndetësor. Sidoqoftë, si çdo sistem, ai gjithashtu ka kufizimet dhe fushat e tij për përmirësime në të ardhmen.

**Kufizimet**

● **Sfidat e integrimit:** Integrimi i sistemit me sistemet ekzistuese të menaxhimit spitalor ose sistemet elektronike të të dhënave shëndetësore mund të paraqesë sfida, duke kufizuar potencialisht rrjedhën e pandërprerë të të dhënave dhe ndërveprueshmërinë.

● **Shqetësimet e shkallëzueshmërisë**: Ndonëse është projektuar për t'u shkallëzuar, sistemi mund të përballet me sfida të shkallëzueshmërisë kur vendoset në rrjete më të mëdha të kujdesit shëndetësor ose nën skenarë të përdorimit të rëndë, duke kërkuar optimizim të mëtejshëm.

● **Varësitë teknologjike**: Mbështetja në teknologji ose korniza specifike mund të përbëjë kufizime për sa i përket përputhshmërisë së platformës dhe përmirësimeve në terma afatgjatë.

● **Adoptimi nga përdoruesi**: Suksesi i sistemit varet shumë nga pranimi dhe miratimi i përdoruesit. Rezistenca ndaj ndryshimeve ndërmjet personelit të kujdesit shëndetësor ose trajnimi joadekuat mund të kufizojë efektivitetin e sistemit.

**Pune per te ardhmen**

**1. Ndërveprueshmëri e përmirësuar:** Përsëritjet e ardhshme mund të fokusohen në përmirësimin e aftësive të integrimit, duke siguruar shkëmbim të pandërprerë të të dhënave me sisteme të tjera të kujdesit shëndetësor për të krijuar një rrjet më të plotë informacioni për pacientët.

**2. Integrimi celular:** Zhvillimi i aplikacioneve celulare për të plotësuar sistemin e bazuar në ueb, duke mundësuar akses në lëvizje për reaguesit e urgjencës dhe stafin e kujdesit shëndetësor, mund të përmirësojë përdorshmërinë.

**3. Integrimi i makinerise dhe AI:** Përdorimi i algoritmeve të AI për analitikë parashikuese, optimizimin e shpërndarjes së burimeve dhe sistemet e mbështetjes së vendimeve mund të rrisë ndjeshëm aftësitë e sistemit.

**4. Përmirësimet e sigurisë:** Forcimi i masave të sigurisë për të mbrojtur informacionin e ndjeshëm të pacientit dhe për të siguruar përputhjen me rregulloret në zhvillim të privatësisë së të dhënave

**Perspektiva e përgjithshme**

Ndërsa Sistemi i Menaxhimit të Ambulancës trajton sfidat kritike administrative brenda kujdesit shëndetësor, njohja e kufizimeve të tij dhe planifikimi për përmirësimet e ardhshme do të jenë thelbësore për të siguruar efektivitetin dhe rëndësinë e tij të vazhdueshme në peizazhin gjithnjë në zhvillim të teknologjisë së kujdesit shëndetësor.

Duke u fokusuar në adresimin e kufizimeve, duke përqafuar përparimet teknologjike dhe duke iu përshtatur nevojave të përdoruesve, sistemi ka potencialin të evoluojë në një mjet edhe më të fuqishëm dhe të domosdoshëm për optimizimin e shërbimeve të ambulancës dhe përmirësimin e kujdesit ndaj pacientit.

**CHAPTER 7**

**REFERENCES**

**[1] [https://laravel.com/docs/10.x] Laravel documentation**

**[2] [**[**https://chat.openai.com**](https://chat.openai.com/)**] Chat GPT**

**[3] [**[**https://stackoverflow.com**](https://stackoverflow.com/)**] Stack Overflow**

**[4][**[**https://bootstrapmade.com/medicio-free-bootstrap-theme/**](https://bootstrapmade.com/medicio-free-bootstrap-theme/)**] Welcome Page Template**

**[5] [**[**https://getbootstrap.com/**](https://getbootstrap.com/)**] Bootstrap**

**[6][**[**https://bootstrap4.com/preclinic-medical-hospital-bootstrap-4-admin-template/**](https://bootstrap4.com/preclinic-medical-hospital-bootstrap-4-admin-template/)**] Dashboard Template**