Қазақстан Республикасы білім және ғылым министрлігі

Әл - Фараби атындағы Қазақ ұлттық университеті

Ақанова А.Е.

ЖЕДЕЛ КОРОНАРЛЫҚ СИНДРОМ РЕГИСТРІ АҚПАРАТТЫҚ ЖҮЙЕСІН ЖОБАЛАУ ЖӘНЕ ҚҰРУ

ДИПЛОМДЫҚ ЖҰМЫС

Мамандығы 5B070300 - «Aқпaрaттық жүйeлeр»

Қазақстан Республикасы білім және ғылым министрлігі

Әл - Фараби атындағы Қазақ ұлттық университеті

«Қорғауға жіберілген»

Хаттама № \_\_\_ , «\_\_\_»\_\_\_\_\_\_\_ 2022ж.

Кафедра меңгерушісі \_\_\_\_\_\_\_\_\_\_\_\_ ф. - м.ғ.к., Мусиралиева Ш.Ж.

(қолы және мөрі)

«ЖЕДЕЛ КОРОНАРЛЫҚ СИНДРОМ РЕГИСТРІ АҚПАРАТТЫҚ ЖҮЙЕСІН ЖОБАЛАУ ЖӘНЕ ҚҰРУ»

тақырыбында

ДИПЛОМДЫҚ ЖҰМЫС

5B070300 - «Aқпaрaттық жүйeлeр»

Орындаған Ақанова А.Е.

(қолы)

Ғылыми жетекші Самбетбаева А.К.

аға оқытушы (қолы)

Норма бақылаушы Сәндібек Д.Е.

(қолы)

**ТҮЙІНДЕМЕ**

Дипломдық жұмыс 70 беттен, 18 суреттен, 19 пайдаланған әдебиеттер тізімінен тұрады.

**Кілт сөздер**: ақпараттық жүйе, медициналық ақпараттық жүйе, функционалды компонент, web - технология, web - қосымша, Python, Django, SQLite 3, деректер қоры.

**Дипломдық жұмыстың пәні**: ЖКСР, диагностикалау процестерін автоматизациялау, медицинадағы АЖ.

**Дипломдық жұмыстың мақсаты**: ЖКСР АЖ - сін жобалау және программалау тілдері көмегімен осы жүйенің Web - қосымшасын жасау.

**Дипломдық жұмыстың** **әдістері**: объектілі - бағдарлы бағдарламалау принциптерінің негізінде медициналық ұйымының жүйесін жобалау.

**Дипломдық жұмыстың нәтижесі**: ЖКСР және оны диагностикалауға арналған АЖ - web - қосымша.

**Дипломдық жұмыстың практикалық маңызы**:Дипломдық жұмыс нәтижесінде ЖКСР процесстері сипатталанып, оларды автоматизациялайтын web - қосымша көмегімен жедел коронарлық синдромына шалдықтан адамдардың аурыларын алдын алып және диагностикалық жұмыстар жедел жасалынады.

**РЕФЕРАТ**   
Отчет состоит из 100 страниц.

Дипломная работа состоит из 70 страниц, 18 рисунков, 19 источников.

**Ключевые слова:** информационная система, медицинская информационная система, функциональный компонент, web - технология, web - приложение, Python, Django, SQLite 3, база данных.

**Предмет исследования**: РОКС, автоматизация процессов диагностики, ИС в медицине.

**Цель дипломной работы**:Разработка Web - приложения данной системы с помощью языков проектирования и программирования.

**Методы дипломной работы**: проектирование системы медицинской организации на основе принципов объектно - ориентированного программирования.

**Результат дипломной работы**:РОКС и ИС - web - приложение для его диагностики.

**Практическая значимость дипломной работы**:В результате дипломной работы описываются процессы РОКС и с помощью web - приложения, автоматизирующего их, оперативно разрабатываются диагностические и профилактические работы больных с острым коронарным синдромом.

**ABSTRACT**

Diploma work consists of 70 pages, 18 pictures, 19 references.

**Keywords**: information system, medical information system, functional component, web technology, web application, Python, Django, SQLite 3, database.

**The research subject:** ACS, automation of diagnostic processes, IS in medicine.

**The рurpose of research:** Development of a web application of this system using design and programming languages.

**Methods of diploma work:** Designing a medical organization system based on the principles of object - oriented programming.

**The results of the research:** ACS and IS web application for its diagnosis.

**The practical significance of the research:** As a result of the thesis, the ACS processes are described and, with the help of a web application that automates them, diagnostic and preventive work of patients with acute coronary syndrome is promptly developed.

**ҚЫСҚАРТУЛАР МЕН БЕЛГІЛЕУЛЕР**

ЖКСР - жедел коронарлық синдром регистрі.

АЖ - Ақпараттық жүйе.

АТ - Ақпараттық технология.

МАЖ - Медициналық ақпараттық жүйелер.

CRM - customer relationship management.

OББ - обьектіге бағытталған бағдарламалау.

ОЖ - операциялық жүйе.

МАЗМҰНЫ

[Кіріспе 7](#_Toc104374598)

[1 МЕДИЦИНА САЛАСЫНДА АЖ - ДІ ҚҰРУДЫҢ ЕРЕКШЕЛІКТЕРІ 10](#_Toc104374599)

[1.1 Медициналық ақпараттық жүйелер (МАЖ) 10](#_Toc104374600)

[1.2 МАЖ - дің классификациясы 10](#_Toc104374601)

[1.2.1 Құрылғы - компьютерлік медициналық жүйелер 11](#_Toc104374602)

[1.2.2 Мониторинг жүргізуге арналған жүйелер 12](#_Toc104374603)

[1.2.3 Емдеу процесстерін басқару жүйелері 13](#_Toc104374604)

[1.2.4 Телемедицина 14](#_Toc104374605)

[1.3 МАЖ - ді жетілдіру жолдары 15](#_Toc104374606)

[1.4 МАЖ шешуге мүмкіндік беретін мəселелер 16](#_Toc104374607)

[1.5 МАЖ - дің негізгі функционалды компоненттері 17](#_Toc104374608)

[1.6. МАЖ - ді қолданудың артықшылықтары 18](#_Toc104374609)

[1.7 Заманауи МАЖ - ге шолу 19](#_Toc104374610)

[1.8 ҚР Денсаулық сақтау министрлігінің АЖ - і 24](#_Toc104374615)

[2 ВЕБ - ТЕХНОЛОГИЯЛАРДЫҢ ҚОЛДАНУ АЙМАҒЫНДАҒЫ ӘДІСТЕРІ 27](#_Toc104374616)

[2.1 Web - қосымшаның серверлік бағдарламау тілі - Python 27](#_Toc104374617)

[2.1.1 Python бағдарламалау тілі 27](#_Toc104374620)

[2.1.2](#_Toc104374621)[[2.1.4 Python көмегімен web - қосымшаны құрудың этаптары 37](#_Toc104374621)](#_Toc104374641)

[[2.2 Web - қосымшаның фреймворкі - Django 38](#_Toc104374621)](#_Toc104374643)

[[2.2.1 Django фреймворкі ерекшеліктері 38](#_Toc104374621)](#_Toc104374644)

[[2.2.2 Django кодының жалпы көрінісі 40](#_Toc104374621)](#_Toc104374645)

[[2.2.2.1 Дұрыс View(urls.py) - ге сұранымдарды жіберу 41](#_Toc104374621)](#_Toc104374647)

[[2.2.2.2 Сұранымдарды өңдеу(views.py) 42](#_Toc104374621)](#_Toc104374648)

[[2.2.2.3 Қосымшаның деректерін анықтау (models.py) 43](#_Toc104374621)](#_Toc104374649)

[[2.2.2.4 Деректерге сұраным жасау(views.py) 43](#_Toc104374621)](#_Toc104374650)

[[2.2.2.5 Деректерді көрсету (HTML) 44](#_Toc104374621)](#_Toc104374651)

[Python web - фреймворктары 30](#_Toc104374621)

[2.1.3 Web - қосымша құруға арналған Python кітапханалары 37](#_Toc104374640)

[2.3 Деректер қорымен жұмыс жасау технологиясы - SQLite3 45](#_Toc104374652)

[3 ЖЕДЕЛ КОРОНАРЛЫҚ СИНДРОМ РЕГИСТРІ ҮШІН АҚПАРАТТЫҚ ЖҮЙЕНІ ҚҰРУ 54](#_Toc104374653)

[3.1 Жедел коронарлық синдром регистрі web - қосымшасына қойылатын талаптар және қосмышаның жалпы жұмысын сипаттау 54](#_Toc104374654)

[3.2 Web - қосымшаның деректер қорын құру 55](#_Toc104374655)

[3.3 Қосымшаны жобалаудың диаграммалары 56](#_Toc104374656)

[3.4 Web - қосымшаның жұмыс істеу функцияналын сипаттау 57](#_Toc104374657)

[Қорытынды 62](#_Toc104374658)

[Пайдаланылған әдебиеттер 63](#_Toc104374659)

[А Қосымшасы 65](#_Toc104374661)

[Ә Қосымшасы 66](#_Toc104374662)

[Б Қосымшасы 70](#_Toc104374663)

# **Кіріспе**

Жаңа заманда барлық әлемде ауқымды түрде өмірдің барлық дерлік сфераларында АТ - ды енгізу эрасы жүріп жатыр. Қандай да бір процесті автоматизациялау барысында белгілі бір функцияларды адам емес енді компьютер атқаратын болады.

Электр энергиясын жеткізу мен сумен қамтамасыздау, байланыс жүйелері мен транспортты басқару, қаржы ағындарын координациялау, медицина мен білім беру және т.б процестерін компьютерлік және АЖ - сіз қадағалау қазіргі таңда ойға келмес көрісністердің қатарында. АТ қарапайым адамдардың өміріне тікелей немесе жанама түрде әсер етуде, бұған қоса АЖ адамдар еңбегінің, тұрмысының, уақыт өткізуінің, өмір салтының және тіпті ойлау жүйесінің спецификасы мен сапасын өзгертуде. Қазірдің өзінде АЖ мен АТ дамуы тек қана жаңа технологиялық құрылымды ғана емес, сонымен қатар жаңа қоғамдық құрылымды құрастыруда және осындай технологиялар жай ғана тек қоғамның ғана емес, тіпті азаматтардың жеке өмірлерін қадағалаудың мүмкіншіліктерін арттыруда.

Қазіргі таңда медицина және денсаулық сақтау жүйелерін автоматизациялау және цифрландыру өзекті мәселелердің қатарында. Денсаулық сақтау ұйымдарының АЖ - ін жасаудағы негізгі мақсаты - осы ұйымдардағы автоматизациялауға бағынатын барлық дерлік процесстер мен операцияларды барынша жаңа АТ - мен іске асыру.

Заманауи медициналық ұйымдар үлкен көлемді ақпараттарды өндіруде және жинақтауда. Медициналық көмектің сапасы, халықтың жалпылама өмір деңгейі, мемлекеттің даму деңгейі жалпы алғанда денсаулық сақтау мекемеллеріндегі жұмысшылардың медициналық ақпараттарды қаншалықты сауатты әрі тиімді пайдалануына тәуелді келеді. Осы себептен медициналық мәселелерді (диагностикалық, терапевтік, статистикалық, басқарушылық және т.б) шешуде үлкен көлемдегі ақпараттарды қолдану қарқынды жүруде. Осындай жағдайлар емдеу - алдын алу мекемелерінде медициналық АЖ - ді құруға себеп болып жатыр.

Соңғы уақытқа дейін еліміздің денсаулық сақтау жүйесінде автоматизациялаудың қандай да бір белгілері тіпті байқалмайтын. Карталар, бюллетендер, процедуралық есептемелер, пациенттерді және дәрі - дәрмектерді тіркеудің барлығы қағаз түрінде жүретін. Бұл жылдам көмек берудің лездігіне, демек пациенттерді емдеудің сапасына әсерін берді, ал басқа жағынан қарағанда дәрігерлік, медициналық персоналдың жұмысын қиындатты, ал бұл өз кезегінде дәрігерлік қателіктерге және карталарды толтыруға, есептемелерді жүргізіге узақ уақыттың жұмсалуына бой алдырды. Нәтижесінде емдеу - алдын алу мекемелерінің басшылығының жұмысын ауырлатты.

Осы орайда денсаулық сақтау жүйесіне медициналық АТ - ды енгізудің эффективтілігі еліміздің әр тұрғынының денсаулығына әсер ететіні көзге анық көрінеді. Еліміздегі көптеген медициналық орталықтар өздерінің кәсіби жұмыстарында медициналық АЖ - дің қызметтеріне қарқынды түрде жүгінуде. Ал өз кезегінде медициналық АЖ түбегейлі жаңа деңгейде, емдеу мекемелерін басқаруды және медициналық көмек көрсетуді сапалы түрде атқаруға жағдай жасайтын әмбебап программалық өнім болып табылады.

Медициналық АЖ - дің айтарлықтай ерекшелігі олардың медициналық ақпаратпен локалды жұмыс істеуден интеграцияланған жүйеге көшуге мүмкіндік беруі, ал интеграцияланған жүйеде барлық ақпарат бірегей ақпараттық ортада біріктіріледі және қолжетімді болады. Заманауи медициналық АТ - ды қолдану медициналық қызмет көрсету сапасын көтереді, әр түрлі медициналық құрылымдардың басқаруын оптимизациялайды және дамыған елдердің медицина дейгейіне жетуге негіз болады.

Қазіргі уақытта ақпараттық нарықта көптеген автоматизацияланған медициналық АЖ бар. Олар әр түрлі сапалық және сандық характеристикаларға ие. Алайда олардың көптігіне қарамастан барлық талапқа сай, емделушінің жағдайын мониторлауға арналған АЖ - ді таңдау қиындық туғызады, себебі қай АЖ - нің барлық мониторлау процестерін барынша қамтитыны туралы толық көрініс жоқтың қасы. Дегенмен де елімізде жалпылама поликлиникаларда және жеке клиникаларда науқастарды тіркеуге, электрондық картаны жүргізуге және медклиника ішінде өзара ақпарат алмасуды іске асыратын АЖ - лер бар. Бірақ олардың кемшілігі сол жүйелердің науқастарға арналған бөлігінің жоқтығында, яғни қазірде науқас жалпылама поликлиникаларда дәрігерге тіркелуі қиындық туғызады, ал электронды амбулатурлы картаны олар көре алмайды, әлі де дәрігерлік рецепттер қағаз түрінде жүргізілуде. Осындай мәселердің көпшілігі жеке клиникаларда шешімін табуда.

Медициналық аурулар тобының бірі – жедел коронарлық синдромды диагностикалау және емдеу денсаулық сақтау жүйесіндегі өзекті мәселерге жатады. Оны алдын - алудағы немесе ушықтармаудағы негізгі принцип жедел диагностикалау болып табылады. Елімізде арнайы осы синдромға арналған ақпараттың жүйелер жоқтың қасы, сол себептен көптеген адамдар жедел коронарлық синдромға шалдыққанын байқамай да жүр.

**Дипломдық жұмыстың зерттеу пәні –** жедел коронарлық синдром регистрі, диагностикалау процестерін автоматизациялау, медицинадағы АЖ.

**Дипломдық жұмыстың мақсаты –** жедел коронарлық синдром регистрі АЖ - сін жобалау және программалау тілдері көмегімен осы жүйенің Web - қосымшасын жасау.

Дипломдық жұмыстың мақсатына жету үшін келесі міндеттерді қарастырған абзал:

* зерттеу тақырыбына толық сипаттама беру;
* медицинада қолданылып жатқан аж - ге шолу жасау;
* жедел коронарлық синдром регистрі жүйесінің жұмысын автоматтандыруға қойылатын талаптарды анықтау;
* АЖ - нің жұмыс істеу функцияналын анықтау;
* web - технологияны қолдана отырып жедел коронарлық синдром регистрі АЖ - сін құру.

**Дипломдық жұмыстың өзектілігі****–** қазіргі кезде АТ - дың интенсивті түрде дамуы барысында оларды денсаулық сақтау жүйесіне енгізу, оның жұмыс жасау тиімділігін арттыру.

**Дипломдық жұмыстың әдістері**. Объектілі - бағдарлы бағдарламалау принциптерінің негізінде медициналық ұйымының жүйесін жобалау.

**Дипломдық жұмыстың теориялық және әдіснамалық негіздері***.* Пәндік сипаттамаларды негізге ала отырып ЖКСР АЖ құрылады. Құрылатын АЖ ЖКСР - гі процестер жұмыстарын жеңілдетеді.

**Қолданылған технологиялар** *–* Python, HTML, CSS.

**Дипломдық жұмыстың құрылымы**.Дипломдық жұмыс кіріспе, негізгі бөлім, қорытынды мен пайдаланылған әдебиеттер тізімі бөлімдерінен құрылды.

Кіріспедедипломдық жұмыстың зерттеу тақырыбына арналған кіріспе мәліметтер және дипломдық жұмыста орындалатын мәселелер туралы баяндалады.

Негізгі бөлімніңбірінші бөлімінде медициналық ақпараттық жүйелер жайлы ауқымды ақпараттар беріледі, екінші бөлімде жасалынатын АЖ web - қосымшаны құруға арналған web - технологиялар сипатталынады және үшінші бөлімде құрылатын АЖ - ні жобалау және детальды түрде құру жұмыстары туралы жазылады*.*

Қорытындыдадипломдық жұмысты зерттеу барысындағы негізгі нәтижелер көрсетіледі*.*

# **1 МЕДИЦИНА САЛАСЫНДА АЖ - ДІ ҚҰРУДЫҢ ЕРЕКШЕЛІКТЕРІ**

## **1.1 Медициналық ақпараттық жүйелер (МАЖ)**

АЖ - лер медицина мен денсаулық сақтау барлық салаларында кездеседі. Саланың тұтас анық жұмыс істеуі және оны тиімді пайдалануы жүйенің реттілігіне байланысты. МАЖ - ді медициналық информатика қарастырады. Қазіргі уақытта медициналық информатика өзіндік пәні, зерттеу нысаны және бірқатар медициналық пәндерден орын алатын дербес ғылым саласы ретінде танылған. Медициналық информатика - бұл медицина мен информатиканың өзара әрекеттесуінің нәтижесі болып табылатын қолданбалы медициналық - техникалық ғылым, яғни, медицина күрделі әдіс - тәсілдерді ұсынса, ал информатика тапсырма - құралдар - әдістер - қабылдаулар жүйесінде құрылған күрделі әдіс - тәсілдерді бірыңғай әдістемелік тәсілде болуын қамтамасыз етеді.

Бұл жағдайда медицинакалық информатиканы зерттеу пәні әдістемелік - биологиялық, клиникалық және профилактикалық мәселелермен байланысты АЖ - лер болады. Медициналық информатиканы зерттеу объектісі денсаулық сақтау саласында жүзеге асырылатын АТ - лар болып табылады. Медициналық информатиканың негізгі мақсаты - халықтың денсаулығын сақтау сапасын арттыруды қамтамасыз ететін компьютерлік технологияларды қолдану арқылы медицинадағы АЖ - ді оңтайландыру [1].

## **1.2 МАЖ - дің классификациясы**

Денсаулық сақтауды ақпараттандырудың негізгі буыны АЖ болып табылады. МАЖ - дің жіктелуі иерархиялық принципке негізделген және денсаулық сақтаудың көп деңгейлі құрылымына сәйкес келеді. Келесідей деңгейлер ажыратылады:

1. базалық деңгейдегі медициналық ақпараттық жүйелердің негізгі мақсаты – әртүрлі мамандықтағы дәрігерлердің жұмысын компьютерлік қолдау болып табылады. Олар профилактикалық және зертханалық - диагностикалық жұмыстың сапасын арттыруға мүмкіндік береді, әсіресе білікті мамандардың уақыт тапшылығы кезінде жаппай қызмет көрсету кезінде жүзеге асырылады.

Шешілетін мәселелерге сәйкес келісідей ішкі жүйелер қарастырылады:

* ақпараттық - анықтамалық жүйелер (пайдаланушының сұранысы бойынша медициналық ақпаратты іздеуге және беруге арналған);
* консультациялық - диагностикалық жүйелер (әртүрлі бейіндегі аурулар кезінде емдеу тәсілдері бойынша болжамды және ұсынымдарды қоса алғанда, патологиялық жағдайларды диагностикалау үшін);
* аспаптық - компьютерлік жүйелер (науқастың денесімен тікелей байланыста болатын диагностикалық және емдеу процесін ақпараттық қолдау және/немесе автоматтандыру үшін);
* мамандардың автоматтандырылған жұмыс орындары (тиісті мамандық дәрігерінің бүкіл технологиялық процесін автоматтандыру және диагностикалық және тактикалық медициналық шешімдерді қабылдау кезінде ақпараттық қолдауды қамтамасыз ету);

1. емдеу - алдын алу мекемелері деңгейіндегі медициналық АЖ - лер.Келесі негізгі топтармен ұсынылады:

* кеңес беру орталықтарының АЖ - і (тиісті бөлімшелердің жұмыс істеуін қамтамасыз етуге және шұғыл жағдайларда кеңес беру, диагностикалау және шешім қабылдау кезінде дәрігерлерге ақпараттық қолдау көрсетуге арналған);
* медициналық қызмет туралы ақпарат банктері(мекеме қызметкерлерінің, тіркелген халықтың сапалық және сандық құрамы туралы жиынтық мәліметтерді, негізгі статистикалық мәліметтерді, қызмет көрсету аудандарының сипаттамаларын және басқа да қажетті мәліметтерді қамтиды);
* дербестендірілген тіркелімдермен (формализацияланған сырқатнама немесе амбулаторлық карта негізінде тіркелген немесе бақыланатын контингентке ақпаратты қамтиды);
* скринингтік жүйелермен (халықты дәрігерге дейінгі профилактикалық тексеруден өткізу үшін, сондай - ақ маманның көмегіне мұқтаж қауіпті топтар мен науқастарды анықтау үшін);
* емдеу - профилактикалық мекеменің ақпараттық жүйелері (барлық ақпараттық ағындарды бірыңғай жүйеге біріктіруге негізделген және мекеме қызметінің әр түрін автоматтандыруды қамтамасыз етеді);
* Ғылыми зерттеу институты және медициналық ЖОО ақпараттық жүйелері (3 негізгі міндеттерді - оқытудың технологиялық процесін ақпараттандыру, ҒЗИ және ЖОО ғылыми - зерттеу жұмыстары және басқару қызметін қарастырады) [2].

1. аумақтық деңгейдегі МАЖ - лер.Олар:

* Денсаулық сақтау органының аумақтық АЖ;
* АЖ мамандандырылған медициналық қызметтердің медициналық қызметкерлерінің қызметін ақпараттық қолдаумен қамтамасыз ететін медициналық - технологиялық міндеттерді шешуге арналған;
* аймақтық деңгейде бірыңғай ақпараттық кеңістік құруды қамтамасыз ететін компьютерлік телекоммуникациялық медициналық желілер.

1. денсаулық сақтау жүйесінің мемлекеттік деңгейін ақпараттық қолдауға арналған мемлекеттік деңгей.

### **1.2.1 Құрылғы - компьютерлік медициналық жүйелер**

Мамандандырылған медициналық АЖ - дің маңызды түрі – медициналық аспаптар мен компьютерлік жүйелер. Қазіргі уақытта медицинаны ақпараттандыру бағыттарының бірі медициналық жабдықты компьютерлендіру болып табылады. Медициналық тәжірибеде компьютерді өлшеу және басқару техникасымен бірге қолдану науқастың жағдайы туралы ақпаратты автоматты түрде жинауды, оны нақты уақыт режимінде өңдеуді және оның жағдайын басқаруды қамтамасыз етудің жаңа тиімді құралдарын жасауға мүмкіндік берді. Бұл процесс аспаптық зерттеу әдістері мен қарқынды терапияны жаңа сапалы деңгейге көтерген құрылғы - компьютерлік медициналық жүйелерді құруға әкелді. Құрылғы - компьютерлік медициналық жүйелер базалық деңгейдегі медициналық ақпараттық жүйелерге жатады. Бұл класс жүйелерінің басты айырмашылығы – зерттеу объектісімен тікелей байланыста және нақты уақыт режимінде жұмыс істеу.

Олар күрделі бағдарламалық - аппараттық кешендер. Медициналық аспаптар мен компьютерлік жүйелер жұмысы үшін есептеуіш техникадан басқа, арнайы медициналық аспаптар, жабдықтар, телетехника, байланыс құралдары қажет.

Медициналық аспаптар мен компьютерлік жүйелерінің типтік өкілдері науқастардың жай - күйін мониторингтеудің медициналық жүйелері болып табылады, мысалы, күрделі операцияларды жүргізу кезінде. Оларға томография деректерін компьютерлік талдау, ультрадыбыстық диагностика, радиография жүйелері, микробиологиялық және вирусологиялық зерттеулер деректерін автоматтандырылған талдау, адам жасушалары мен терілерін талдау жүйелері жатады.

Медициналық аспаптар мен компьютерлік жүйелерде үш негізгі компонентті бөлуге болады. Олар медициналық, аппараттық және бағдарламалық қамтамасыз ету.

Медициналық аспаптар мен компьютерлік жүйелерге қатысты медициналық қамтамасыз ету жүйенің аппараттық және бағдарламалық бөліктерінің мүмкіндіктеріне сәйкес шешілетін медициналық міндеттердің таңдалған шеңберін іске асыру әдістерін қамтиды. Медициналық қамтамасыз етуге пайдаланылатын әдістемелер жиынтығы, өлшенетін физиологиялық параметрлер және оларды өлшеу әдістері, жүйенің пациентке әсер етуінің тәсілдері мен жол берілетін шекараларын айқындау жатады.

Аппараттық құрал деп жүйенің техникалық бөлігін, соның ішінде медициналық және биологиялық ақпарат алу құралдарын, терапевтік әсер ету құралдарын және компьютерлік техниканы жүзеге асыру тәсілдерін түсіну керек.

Бағдарламалық жасақтамаға медициналық - биологиялық ақпаратты өңдеудің математикалық әдістері, алгоритмдер және бүкіл жүйенің жұмысын жүзеге асыратын нақты бағдарламалар кіреді [3].

### **1.2.2 Мониторинг жүргізуге арналған жүйелер**

Науқастың жағдайын жедел бағалау міндеті медицинадағы бірқатар маңызды практикалық бағыттарда және, ең алдымен, қарқынды терапия палаталарында, операциялық және операциядан кейінгі бөлімдерде науқасты үздіксіз бақылау кезінде туындайды.

Бұл жағдайда ағзаның физиологиялық жүйелерінің жағдайын сипаттайтын көптеген деректерді ұзақ және үздіксіз талдау негізінде емдеу кезінде асқынулардың жедел диагнозын ғана емес, сонымен қатар науқастың жағдайын болжауды қамтамасыз ету, сонымен қатар пайда болатын бұзылулардың оңтайлы түзетілуін анықтау қажет. Бұл мәселені шешу үшін медициналық аспаптар мен компьютерлік жүйелері мониторы арналған. Мониторинг барысында көп пайдаланылатын параметрлерге электрокардиограмма, тыныс алу жиілігі, әр түрлі нүктелердегі қан қысымы, қан газдарының мөлшері, температура қисығы, шығарылған ауадағы газдардың мөлшері және қан айналымының минуттық көлемін жатқызуға болады.

Функционалды диагностикаға арналған монитор жүйелерінің ақпараттық

құралы мен ұқсас жүйелер іс жүзінде мүлдем айырмашылығы жоқ. Монитор жүйелерінің маңызды ерекшелігі - жедел талдау және олардың нәтижелерін нақты уақыт режимінде визуализациялау құралдарының болуы. Бұл монитор экранында бақыланатын шамалардың әртүрлі туындыларының динамикасын көрсетуге мүмкіндік береді. Мұның бәрі әртүрлі уақыт ауқымында жүзеге асырылады. Сонымен қатар, жүйенің сапасы неғұрлым жоғары болса, бақыланатын және онымен байланысты көрсеткіштердің динамикасын бақылау мүмкіндігі соғұрлым жоғары болады. Көбінесе монитор жүйелері бірден 6 - ға жуық науқастың жағдайын бір уақытта бақылау үшін қолданылады және олардың әрқайсысында 16 - ға дейін негізгі физиологиялық параметрлерді зерттеуге мүмкіндік береді.

### **1.2.3 Емдеу процесстерін басқару жүйелері**

Емдеу және оңалту процестерін басқару жүйелеріне қарқынды терапияның автоматтандырылған жүйелері, биологиялық кері байланыс, сондай - ақ микропроцессорлық технология негізінде құрылған протездер мен жасанды органдар жатады.

Емдеу процесін басқару жүйелерінде бірінші кезекте жұмыстың сандық параметрлерін дәл мөлшерлеу, пациент денесінің физиологиялық сипаттамаларының өзгергіштігі жағдайында олардың белгіленген мәндерін тұрақты сақтау міндеттері тұрады.

Автоматтандырылған қарқынды терапия жүйелері деп ағзаның жағдайын емдік мақсатта басқаруға, сондай - ақ оны қалыпқа келтіруге, ауру адамның ағзалары мен физиологиялық жүйелерінің табиғи функцияларын қалпына келтіруге, оларды қалыпты деңгейде ұстауға арналған жүйелері қарастырылады. Олардағы құрылымдық конфигурацияға сәйкес қарқынды терапия жүйелері екі сыныпқа бөлінеді - бағдарламалық басқару жүйелері және жабық басқару жүйелері.

Бағдарламалық басқару жүйелеріне емдік әсер етуді жүзеге асыруға арналған жүйелер жатады. Мысалы, есептеу техникасымен жабдықталған түрлі физиотерапиялық аппаратура, дәрілік препараттарды құюға арналған құрылғылар, өкпені жасанды желдетуге және ингаляциялық наркозға арналған аппаратура, жасанды қанайналым аппараттары. Жабық қарқынды терапия жүйелері медициналық аспаптар мен компьютерлік жүйелеріне қарағанда құрылымдық жағынан күрделірек, өйткені олар науқастың жағдайын бақылау, бағалау және емдік әсерді басқару міндеттерін біріктіреді. Сондықтан, іс жүзінде жабық қарқынды терапия жүйелері өте жеке, қатаң бекітілген тапсырмалар үшін жасалады.

Биологиялық кері байланыс жүйелері пациентке оның ішкі мүшелері мен

жүйелерінің жұмысы туралы ағымдағы ақпаратты беруге арналған. Бұл пациенттің саналы ерікті әрекеті арқылы патологияның белгілі бір түрімен емдік әсерге қол жеткізуге мүмкіндік береді [4].

### **1.2.4 Телемедицина**

XXI ғасырдағы телемедицина жаһандық ақпараттық жүйелерді жаппай пайдалануды көздейтін «коммуникациялар ғасырына» айналуға тиіс. Мұндай жүйелерді медицинада қолдану келесідей сапалы жаңа мүмкіндіктер ашады:

* аймақтық клиникалардың ірі медициналық орталықтармен өзара әрекеттесуін қамтамасыз ету;
* жедел түрде соңғы ғылыми - зерттеу нәтижелерін алу;
* кадрларды даярлау және қайта даярлау.

Бұл мүмкіндіктерді бір жалпы тұжырымдама - телемедицинамен сипаттауға болады.

Телемедицина – медициналық ақпаратты қашықтықтан басқаруды көздейтін заманауи емдеу - диагностикалық әдістемелер кешені. Телемедицинаның пайда болуы, әдетте, ғарыштық ұшулар кезіндегі медициналық бақылаумен байланысты. Бастапқыда бұл ғарыштағы жануарлардың өмірлік маңызды көрсеткіштерін ғарыштық аппараттарда өлшеу, содан соң ғарышкерлердіде өлшейтін ғарыш аппараттарында болды.

Телемедицинаның тәжірибеге енуінің себебі коммуникациялық желілердің қарқынды дамуы, сондай - ақ бейне және аудио ақпараттармен және кез - келген ілеспе құжаттармен екі және көп жақты алмасуды қамтамасыз етуге мүмкіндік беретін ақпаратпен жұмыс істеу әдістері болды.

Телемедицина мүмкіндіктерін іске асырудың қарапайым жағдайы – дәрігердің қажетті анықтамалық ақпаратқа жылдам қол жеткізуі. Телемедицинаның негізгі қосымшасы - медициналық орталықтардан алыс немесе медициналық қызметтерге қол жетімділігі шектеулі халыққа қызмет көрсету.

Телемедицинаның тағы бір маңызды объектісі - бұл аймақтардың диагностикалық орталықтарының жүйесі. Көбінесе алыс қашықтықтарда орналасқан әртүрлі медициналық мекемелерде емдеуші дәрігер мен диагностикалық дәрігер арасында жедел байланыс қажет болған кезде қолданылады.

Телемедицинаның тағы бір маңызды бағыты - науқасты құтқару немесе күрделі жағдайларда емдеу тактикасын анықтау үшін орталық медициналық мекемелердің мамандарына, оның ішінде әлемдегі ең ірі медициналық орталықтардан жедел кеңес беру қажет болатын жедел жағдай және күрделі жағдайларын айтады.

Келесі бағыт – қашықтықтан медициналық білім беру. Қазіргі заманғы ақпараттық жүйелерді құрудағы ең перспективті тенденцияларды “модельдеуге негізделген сәулет” тұжырымдамасымен біріктіруге болады, бұл тәсілдің философиясы күрделі жүйеде барлық мүмкін сценарийлерді, жүйенің болашақ дамуын және т. б. болжау мүмкін емес. Сондықтан барлық қатысушылар үшін ортақ объектілік модель жасалынып, оны құру және қосымшаларды жүйеге біріктіру принциптерін анықтаған жөн. Аталған тұжырымдама мәселелерді жобалау және іске асыру міндеттерін бөлу арқылы шешеді. Бұл сенімді дәлелденген модельдерге негізделген жаңа кеңейтілген технологияларды қолдана отырып, өзара әрекеттесудің жаңа сипаттамаларын тез әзірлеуге және енгізуге мүмкіндік береді. Ақпараттық "модельдеуге негізделген сәулет” тұжырымдамасы құру процесі кез - келген күрделі ақпараттық жобаны әзірлеудің типтік циклі болып табылады. Оның ішіне талаптарды әзірлеу кезеңі - талдау кезеңі - іске асыру кезеңі кіреді. Әрбір фазаның шеңберінде оған қойылатын талаптарға сәйкестік, келісімділік және функционалдылық мәселелері пысықталады. Қазіргі заманғы АЖ - лер, әдетте, интернет сияқты ғаламдық желілерде орын алады. Телемедицина жүйелері де ерекшелік емес. Офлайн, жергілікті қосымшалардың уақыты артта қалып бара жатыр. Олардың орнын әртүрлі архитектуралармен, көп платформалармен, деректер мен хаттамалардың әртүрлі форматтарымен сипатталатын ақпараттық жүйелер басады.

## **1.3 МАЖ - ді жетілдіру жолдары**

Медициналық АТ - ға сыртқы ақпараттық факторлардың ағзаға әсер ету құралдары, оларды қолдану әдістері мен әдістерінің сипаттамасы және практикалық дағдыларды оқыту процесі кіреді. Тиісінше, осы технологияларды одан әрі дамыту келесі практикалық мәселелерді қарастыруды және шешуді талап етеді. Бірінші кезекте клиникалық практикаға қауіпсіздік және оларды қолданудың қарапайымдылығы, оларды қолданудың жоғары терапиялық тиімділігі сияқты талаптарға жауап беретін ақпараттық әсер етудің дәлелденген құралдары мен әдістерін кеңінен енгізу қажеттілігі туралы өзекті мәселе тұр. Келесі өзекті мәселе – ақпараттық медицинаның принциптері мен постулаттарына сәйкес келетін адам ағзасына әсер етудің жаңа құралдары мен әдістерін әзірлеуді және құруды ынталандыру және көтермелеу. Медицинаның осы саласын одан әрі дамыту және жетілдіру ақпараттық медицинаның принциптері мен постулаттарына сәйкес ағзадағы өзгерістерге барабар ақпараттық әсер етукезінде кері биологиялық байланыс құралдары мен әдістерін оңтайландырумен байланысты [5].

Қазіргі уақытта бірқатар медициналық, әлеуметтік және экономикалық мәселелерді шешудің негізгі жолдарының бірі - медицина қызметкерлерінің жұмысын ақпараттандыру. Бұл проблемаларға денсаулық сақтаудың маңызды үш көрсеткіштері - емдеу сапасын, пациенттердің қауіпсіздігі деңгейін, медициналық көмектің экономикалық тиімділігін арттыруды қамтамасыз ететін тиімді құралдарды іздеу жатады. Ақпараттандырудың негізгі буыны ауруханаларда шешім қабылдауды қолдау тетіктерімен жабдықталған заманауи клиникалық АЖ - ді пайдалану болып табылады.

Алайда, бұл жүйелер кең таралмады, өйткені клиникалық АЖ - ді құрудың ғылыми және әдіснамалық тәсілдері әлі әзірленбеген.

## **1.4 МАЖ шешуге мүмкіндік беретін мəселелер**

Әдетте, әрбір МАЖ медициналық мекеме қызметінің әртүрлі компоненттерін автоматтандыруға жауап беретін блоктардан тұрады. Оларға мыналар жатады:

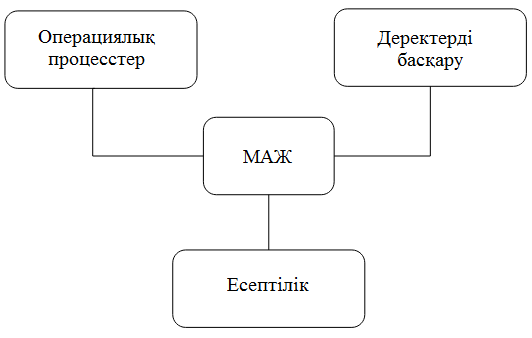
* пациенттерді тіркеу және электрондық медициналық карталар;
* медициналық зерттеулер туралы мәліметтер;
* дәрігер мен медбикенің жұмысы;
* мекеменің ресурстарын бөлу, олардың кестесі;
* қаржыны басқару және есепке алу;
* әкімшілік ақпарат және байланыс құралдары қызметкерлері;
* дәрі - дәрмектер, тағайындау журналы;
* медициналық көмек көрсету стандарттары және тағы басқалар.

МАЖ арқылы қандай мәселелерді шешуге болатындығын және бұл жұмысты ұйымдастыруға қалай әсер ететіні «1.1- суретте» көрсетілген.

Деректерді басқару және процестерді оңтайландыру.АЖ пациенттер туралы және медициналық ұйым қызметінің нәтижелері туралы деректердің үлкен ауқымын басқаруға мүмкіндік береді. Жүйеге кіргенде МАЖ - ға енгізілген барлық ақпаратты кез келген уақытта жүйенің қалаған нүктесінен пайдалануға қол жетімді болады. Осылайша пациенттерге деген көзқарас біріздендіріледі, ал медициналық құжаттама бір үлгі бойынша ресімделеді.

Cалдары мен жеке кабинеттеріне арналған электрондық құрылымдар құруға, бірнеше мекемелерді бірыңғай электрондық жүйеге біріктіруге мүмкіндік береді. МАЖ - дың көпшілігінде икемді алгоритмдер және есеп беру мен жүргізудің интуитивті құралдары бар.

Ақпараттың қолжетімділігі*.*МАЖ - дағы барлық ақпарат талдау және өңдеу үшін қол жетімді, ал бұл шын мәнінде үлкен электронды мұрағат. Жүйе әр түрлі пайдаланушы топтарына әртүрлі бөлімдерге қол жеткізуге мүмкіндік береді (мысалы, науқастарға арналған жеке порталды немесе байланыс және ақпарат алмасу мүмкіндігі бар дәрігерлерге арналған ішкі порталды қолдау).

****

1.1- сурет. МАЖ шешуге мүмкіндік беретін мәселелер

## **1.5 МАЖ - дің негізгі функционалды компоненттері**

Кешенді медициналық ақпараттық жүйелер, әдетте, модульдерден тұрады. Бұл әртүрлі типтегі мекемелер үшін қажетті конфигурациядағы МАЖ - н жинауға, конфигурациялауға және модульдерді одан әрі қосу не жою мүмкіндігімен қажетті функционалдылықты қамтамасыз етуге мүмкіндік береді. МАЖ - нің құрылымы - бірнеше үлкен топтарға біріктіруге болатын келесідей жеке компоненттерден тұрады [6]:

1. аналитикалық және басқару компоненттері.Бұл басқару есебін жүргізу модульдері мен құралдары, медициналық қызметтердің сапасы мен тиімділігін талдау құралдары. МАЖ - дың бұл компоненттері медициналық ұйымның жағдайын талдауға, проблемалық аймақтарды анықтауға және бизнес - процестерді оңтайландыруға мүмкіндік береді. Пайдаланушы деңгейінде, қолжетімділік деңгейі бойынша шектеулерді ескере отырып, кез келген өлшемшарттар бойынша медициналық жазбаларды іздеуге болады. Талдау нәтижелерін графиктер, кестелер немесе басып шығару түрінде көрсетуге болады.
2. медициналық компоненттер*.*Осыған пациенттерді тіркеумен байланысты барлық модульдер, электрондық медициналық карталардың тізілімін жүргізу, аурухана парақтарын есепке алу, емдеу хаттамаларын жүргізу, әртүрлі мекемелердегі (амбулатория, емхана, стационар) науқастарды емдеуді ақпараттық сүйемелдеу, медициналық статистика және аналитика, ауру тарихы және тағы басқалар кіреді.
3. қаржылық - экономикалық компоненттер*.*Оларға дәрі - дәрмектерді есепке алу құралдары, қорларды басқару, емдеудің өзіндік құны мен медициналық қызмет көрсету тарифтерін есептеу, дәрігерлерге үстемеақыны есептеу, ұйым қызметіне экономикалық талдау жүргізу құралдары және т. б. жатады.
4. деректер алмасу компоненттері.Біріздендірілген тізілімдерді, каталогтар мен анықтамалықтарды жүргізу, денсаулық сақтау мекемелері жүйесінде деректермен алмасу, алынған деректерді өңдеу кіреді.

## **1.6. МАЖ - ді қолданудың артықшылықтары**

Егер МАЖ сәтті таңдалса, оны енгізу ұйымның жұмысына оң әсер етеді. Яғни бәрі МАЖ түріне, оның функционалдығына және белгілі бір медициналық мекеменің ерекшелігіне байланысты болады. Біздің өмірімізді заманауи медицинасыз елестету мүмкін емес, сондықтан бұл процестің әртүрлі қатысушылары МАЖ қолданудың қандай артықшылықтарын алады деген сұраққа астыда жауап беріледі.

Медициналық мекемелеріндегі жұмысшылар үшін пайдасы:

* Қағаздарды толтырудан жеңілдетеді. Жазбаларды қайталаудың және ақпаратты басқа құжаттарға енгізудің қажеті жоқ яғни пациенттің картасына қол жеткізе алатын дәрігерлер мен персонал бірыңғай дерекқордан оның ауруының тарихымен, емделу барысымен, зерттеу нәтижелерімен және т.б. таныса алады.
* Қызмет көрсету сапасын арттыру және адам факторының әсерін азайту.Автоматтандырылған құжат айналымы зерттеулер мен көрсетілген қызметтер туралы өзекті ақпаратқа сүйене отырып, қағаз жұмысының санын азайтуға, клиенттер базасын сәтті жүргізуге мүмкіндік береді. Дәрігерлерге диагноз қою әлдеқайда оңай, маңызды деректерді жоғалту қаупі азаяды, өйткені бұл тест нәтижелері қағаз түрінде жиі кездеседі. Олар басқа біреудің картасына тіркелуі мүмкін, сондықтан сіз қайтадан зерттеу жүргізуіңіз керек.
* Телемедицина. Дәрігерлер диагноздың дұрыстығына (әсіресе шұғыл жағдайда), емдеуді тағайындау мен түзетуге қатысты нақты уақытта әріптестерімен және басқа мамандармен кеңесуге, науқастардың жағдайына қашықтықтан мониторинг жүргізуге және т. б. мүмкіндік алады.
* Жұмыстың жүйелілігі.Клиникалардың онлайн - тіркеуі болуы, пациенттердің жалпы ауруханалық базаларын жүргізуі, мамандардың жүктемесі мен кестесін ескере отырып, осы немесе өзге қызметтерге сұранысты бағалауға, ал жеке орталықтарға - баға белгілеуді қалыптастыруға мүмкіндігі бола отырып, оларды филиалдарға таратуы ыңғайлы.

Пациенттер үшін үшін пайдасы:

* МАЖ арқасында пациенттер өз деректеріне қол жеткізе алады, зертханалық талдаулардың нәтижелерін жедел ала алады және оларды емдеуші дәрігермен бірге бақылай алады, оған қабылдауға жазыла алады, Кері байланысты қолдай алады және т. б. Жалған және медициналық деректердің жоғалу ықтималдығы төмендейді, өйткені пациент оларды өздігінен бақылайды. Интернеттегі алдын - ала жазу жүйесі ауруханада кезекке тұрмауға мүмкіндік береді.

## **1.7 Заманауи МАЖ - ге шолу**

**Medods**

Medods – жеке медициналық орталықтар мен стоматологияларды, сонымен қатар, клиникалар желісін ұйымдастыруға арналған платформасы «1.2-суретте» көрсетілген. Бағдарлама жергілікті және бұлтты нұсқада қол жетімді. Жүйе барлық қажетті модульдерді қолдайды және пациенттерді қабылдауға жазуға, олардың электрондық карталарын жүргізуге, сайттан онлайн - жазбаны конфигурациялауға, келісімшарттар мен басқа құжаттарды автоматты түрде жасауға, шот - фактураларды жүргізуге және бақылауға мүмкіндік береді, төлемдер, қорларды есепке алу, маркетингтік акцияларды жоспарлау, email - және SMS - хабарламалар, жұмыстың жиынтық статистикасын алу және т.б фунционалы бар. Medods – пациенттердің кестесі мен жазбаларын және бизнес - аналитикалық құралдарды қолдайтын CRM жүйесінің элементтерін сәтті біріктірудің мысалы болып табылады.



1.2-сурет. Medods МАЖ - сі

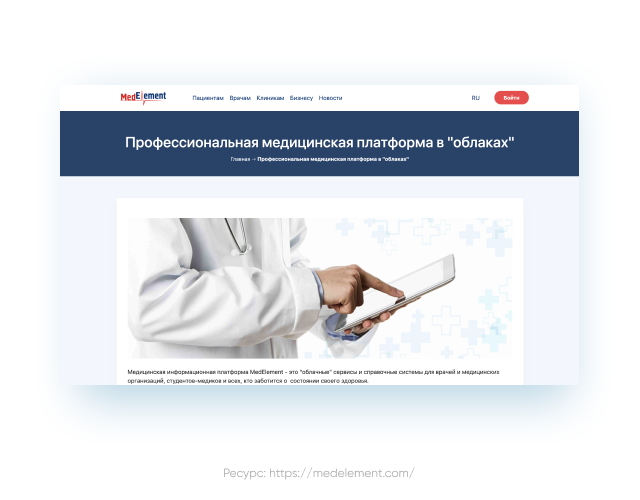
Артықшылықтары**.** Medods - тің маңызды артықшылықтары - онлайн жазбаның болуы, басшының жұмыс үстелі, UIS телефониясымен кіріктірілген интеграция, басқа API телефондарымен интеграция, мысалы, Asterisk, маркетингтік құралдарды қолдау, 1С интеграциясы және т.б саналады. Техникалық және клиенттік сүйемелдеу сатып алу құнына енгізілген. Ыңғайлы, интуитивті интерфейс жүйеде жасалған.

Кемшіліктері**.** Көп факторлы авторизацияға, бірнеше жерде сақтап көшірмелеуге, кіріктірілген интеграцияның шектеулі мүмкіндіктеріне, деректер қоймасына қолдау көрсетілмейді. Сынақ нұсқасы берілмейді.

Төлем**.** Лицензияны сатып алудың екі әдісі болады: абоненттік пайдалану немесе лицензияны бір реттік сатып алу. Төлем жазылу бойынша жүргізіледі.

# **MedElement**

MedElement – Қазақстанда жасалған МАЖ «1.3-суретте» көрсетілген. Бұлтты қызметтер мен дәрігерлерге, медициналық студенттерге және денсаулығына қамқорлық жасайтын барлық адамдарға арналған қуатты анықтамалық жүйенің үйлесімі. MedElement - клиникалардың, жеке клиникаларының, стоматологиялардың, дәріханалардың, тамақтану блоктарының, жеке медициналық практикалардың жұмысын автоматтандыру саласында қолданылады. Бұл МАЖ - ның қызықты ерекшелігі - негізгі модульдерді қолдаумен қатар, бұл қуатты анықтамалық жүйе болып табылады. Онда аурулардың, медициналық терминдердің, зертханалық көрсеткіштердің, дәрілік заттардың анықтамалары бар, әлемдік мерзімді басылымдарға шолулар орналастырылған және т.б. Сонымен қатар, бұлтты МАЖ артықшылықтарына барлық медициналық құжаттарды автоматтандыру, есептерді қалыптастыру, маркетингтік ақпаратты жинау, қаржы, қызметтерді есепке алу және т. б. жатады.



1.3-сурет. MedElement МАЖ - сі

Артықшылығы**.** Ыңғайлы Web - сервистері, дәрігерді жылдам іздеуге, қабылдауға жазуға және коммуникацияны жүргізуге арналған мобильді қосымшаның болуы. Бағдарламаға клиникалық шешімдер қабылдауда көмек көрсету технологиясы енгізілген - MedElement интерактивті медициналық анықтамалықтардың онлайн - базасымен байланысты.

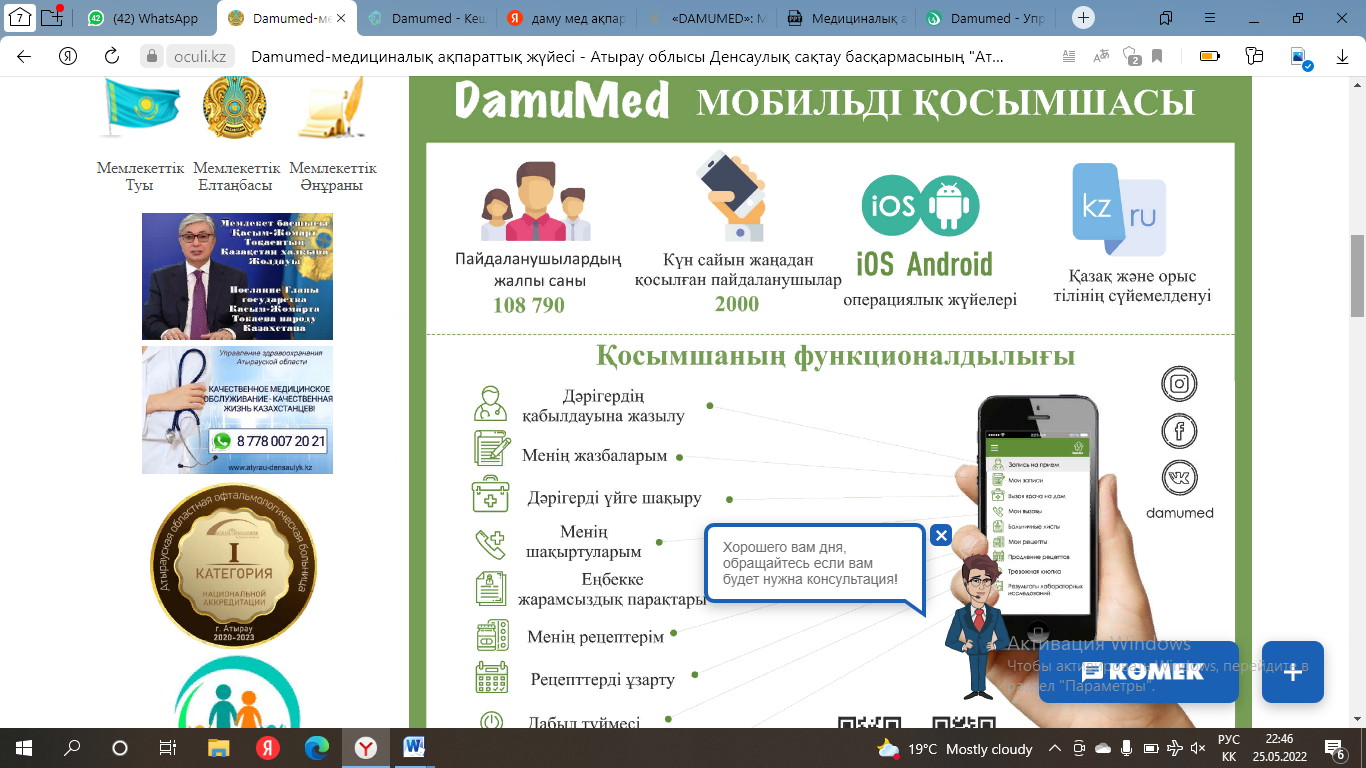
Кемшіліктері**.** Интерфейс өте ыңғайлы емес, "бәрін бірден" қолдау әрқашан плюс емес, бірақ бұл барлық бағдарламаларға қатысты.

Төлем**.** MedElement жүйесіне қосылу үшін төлем ай, жарты жыл, жыл үшін абоненттік төлем қағидаты бойынша жүзеге асырылады [7].

**Damumed**

**Damumed – бұл дәрігердің қабылдауына жазылу, дәрігерді үйге шақыру және медициналық құжаттарды қарау үшін медициналық ұйымға жылдам қол жеткізу «1.4-суретте» көрсетілген.**

Damumed қосымшасын Google Play немесе AppStore-дан жүктеуге болады. Қосымшаны іске қосқан кезде адамнан тіркеу аймағын таңдау сұралады. Қажетті аймақты таңдағаннан кейін Damumed қосымшасының бірінші терезесі ашылады. Тіркеуді бастау үшін "кіру"түймесін басыңыз. Әрі қарай өзіңіздің телефон нөміріңізді және ЖСН енгізіңіз. "Жалғастыру" Түймесін Басыңыз. Жүйе көрсетілген ЖСН таңдалған өңірде бекітілгенін тексереді, содан кейін көрсетілген деректермен пайдаланушыны іздейді. Егер пайдаланушы табылса, онда сіздің телефон нөміріңізге SMS-код жіберіледі, оны "SMS-тен алынған Код" өрісіне енгізу қажет және сіз қосымшаға кіре аласыз.



1.4-сурет. Damumed МАЖ-сі

DamuMed қосымшасында келесі мүмкіндіктер бар:

1. Учаскелік дәрігердің қабылдауына жазылу

2. Дәрігердің қабылдауына жазылу және дәрігерді үйге шақыру.

«Амбулаториялық-емханалық көмек көрсету порталы» ақпараттық жүйесінің» амбулаториялық деңгейдегі қызметтер тіркемесі 2013 жылдың қаңтар айынан бастап пайдаланылып келеді. Жобаның өзіндік бірқатар мақсаттары бар. Олар: КДПжО учаскелік дәрігерлер, дәрігер-мамандар мен дигнозшылардың қызметін, амбулаторлық-емханалық қызметін кешенді жан басына шаққандағы қаржыландыруды қолдану жағдайында тұрғындарға кеңес беру-диагностикалау көмек көрсетуге жолдамалар ағынын бағдарлау бөлігінде статистика жүргізушілердің қызметін автоматтандыру, ДСБҰЖ қағидаларын іске асыру, медициналық ұйым мен емдеуші дәрігерді еркін таңдау, бәсекеге қабілеттілік, қызмет көрсету үрдісінің ашықтығы.

«Медициналық қызмет көрсету сапасын басқару жүйесі»

«Даму» Ақпараттық Технологиялар Орталығы» ЖШС 2005 жылы Қазақстан Республикасы Денсаулық сақтау министрлігі жобаларының бірі болыып табылатын «Медициналық қызмет көрсету сапасын басқару жүйесі» жобасын іске асыра бастады. Жоба 2011 жылдан бастап пайдаланылуда. Негізгі мақсаттарына тұрғындарға медициналық көмек көрсету мен денсаулығын сақтауды арттыру, ҚР ДСМ МҚСБ комитетінің, оның құрылымдық бөлімшелердің тиімді қызметін қамтамасыз етуі сапаны бақылау және көрсетілетін медициналық қызметтердің толықтығы саласында мемлекеттік саясатты іске асыру бойынша бақылау міндеттерін өндірімді етіп жүзеге асыруға мүмкіндік беру кіреді.

«Тіркелген тұрғындар тіркемесі»

«Тіркелген тұрғындар тіркемесі» порталы 2011 жылдан бастап қолданыста. Науқас және оның сырқаты жөнінде диспансерлік есепке алынуы, туу/қайтыс болу туралы куәліктері туралы демографиялық мәліметтерінің жалпы ақпарат көзі болып табылатын жобаның негізгі мақсаттары – республиканың элктрондық тұрғындар санағының мәліметтерін жасау, электронды қызмет ұсынуды қамтамасыз ету.

**Doctor Eleks**

Doctor Eleks – кез келген көлемдегі және бейіндегі (жеке және мемлекеттік) клиникалардың жұмысын оңтайландыруға мүмкіндік беретін кешенді шешім «1.4-суретте көрсетілген». Әзірлеуші - Eleks компаниясы (Львов, Украина). Doctor Eleks пациенттің электрондық медициналық картасын, құжат үлгілерін өңдеу құралдарын, дәрігердің жеке кабинетін, тіркеу модулін және есептілікпен, қаржымен, персоналмен жұмыс істеуді қолдайды. Кестенің ішкі жүйесі дәрігерлер мен пациенттердің тілектерін ескере отырып, қызметкерлердің жұмыс кестесін құруға мүмкіндік береді. Doctor Eleks зертханалық ақпараттық жүйесін жеке бағдарламалық өнім ретінде пайдалануға болады. Қосымша мүмкіндіктердің ішінде құжаттарға қосуға болатын бейнелер мен суреттерді өңдеуге арналған толық редактор бар. Есеп берудің икемді технологиясы, медициналық құжаттардың аудитін жүргізу мүмкіндігі бар, PACS модулі, Web - клиент және басқалары қолдау көрсетеді.

# 

1.5-сурет. Doctor Eleks МАЖ - сі

Артықшылығы**.** Қуатты функционал, аралас АЖ - мен, сыртқы зертханалармен, сақтандыру компанияларымен HL7 форматында деректермен алмасу үшін коммуникациялық сервердің болуы. Toshiba ультрадыбыстық интеграциясы бар, DICOM суреттерін импорттауға, DICOM үйлесімді жабдықты қосуға және тағы басқаларға қолдау көрсетіледі. Doctor Eleks eHealth жүйесіне қосылған, жүйе тексеруден өткен.

Кемшіліктері**.** Егер олар бар болса да (мысалы, электронды бағыттарға қолдау көрсетілмейді), онда бұл кемшіліктер басқа мүмкіндіктермен өтеледі.

Төлем**.** Төлем лицензиялау тұрақты және уақытша лицензиялар негізінде жүргізіледі. Енгізу, интеграциялау, техникалық қолдау қызметтері енгізу аяқталғаннан кейін мобильді қосымшаны, Web - қызметтерді пайдалану және сақтандыру компанияларымен жұмыс істеу құны сияқты бөлек төленеді.

# **EMCiMED**

# EMCiMED – Украинаның eHealth жүйесіне қосылған медициналық мекемелерге, жеке клиникалар мен зертханаларға арналған Украинаның озық медициналық ақпараттық жүйесі «1.6-суретте» көрсетілген. Әрбір жеке мекеме үшін қажетті конфигурация оңай жинауға болатын модульдерден тұрады. Негізгі қолдау көрсетілетін модульдерге - тіркеу бөлімі, персоналды басқару, ұйымды басқару, емхана, аурухана, зертхана, серіктестікті басқару жатады. Қосымшадағы модульдерді - қызметтерді есепке алу, қорларды басқару, PACS медициналық бейнелеу мұрағаты және т.б еркін қолдануға мүмкіндік бар. Қажет болған жағдайда олар EMCiMED - емхана және EMCiMED - стационар пакеттерінің құрамында болуы мүмкін.



1.6-сурет. EMCiMED МАЖ - сі

Артықшылығы**.** Ұйым талаптарына сәйкес модульдерді, икемді параметрді, қуатты функционалды компонентті таңдау мүмкіндігі бар. Жүйе USB кілттерін пайдалану және барлық ақпаратты шифрлау арқылы қорғалған, соған қоса басқа өнімдерді, мысалы, 1С интеграциясын қолдайды. Тексерістен өткен және Украина медицина ұйымдары пайдалануға ұсынған.

Кемшіліктері**.** Егер бар болса, онда олар бағдарламамен жұмыс істеуге айтарлықтай әсер етпейді.

Төлем. Лицензиялар мерзімсіз, пайдаланудың бүкіл кезеңіне бір реттік төлемді көздейді. Жеткізу шартын жасау кезінде барлық шарттар тапсырыс берушімен егжей - тегжейлі талқыланады.

## **1.8 ҚР Денсаулық сақтау министрлігінің АЖ - і**

«Амбулаториялық - емханалық көмек порталы» АЖ - сінің«Амбулаториялық деңгейдегі қызметтерінің электрондық тіркелімдер» модулі

Аталған жүйе емделуші жайлы дербестендірілген мәліметтерді, оның амбулаториялық - емханалық мекемелерге жолығуы жайылы мәліметтерді қалыптастыруға бағытталған. АЖ үшін автоматизациялау объектісі ретінде амбулаториялық - емханалық көмек беретін медициналық денсаулық сақтау мекелемелерінің қызмет саласы қарастырылады. Қазіргі таңда АЖ - де "электрондық үкімет" порталы көмегімен "дәрігер қабылдауына жазылу" мен "дәрігерді үйге шақыру" қызметтерін қолдану мүмкіндіктері жасалынған. АЖ - ге кіру бірыңғай авторизациялану беті көмегімен <https://www.eisz.kz> сілтемесі бойынша іске асырылады.

Бастапқы медициналық - санитариялық көмек тарифіне қосымша компонент

Жоғарыда аталған порталдың мақсаты бойынша медицина қызметкерлерін материалдық деңгейлерін көтеретін, оларды кәсіптік оқуға жіберетін, жеткен нәтижелері үшін мекелемерін қаржыландыру мақсатында жан басына алғандағы нормативтік дамытушы құрамдасын айқындаудан құралады. Порталдың әрбір өңірлердегі ұйымдарының индикаторлары жүйесі ұпайлар құнын есептеуді автоматтандыруға және тарифке қосымша компонент сомасын есептеу негізінде айқындауға негізделген.

АЖ - ні құру мен енгізу мақсаты ретінде денсаулық сақтаудың алдынғы деңгейіндегі халыққа емдеу - алдын алу көмектерін көрсету және оның сапасын арттыру белгіленді.

Портал жұмысының басты нысанды келесідей көрсеткіштері бар:

* ана және бала өлімдері көрсеткіштерінің төмендеуі;
* көзге байқалатын онкологиялық ауруларды лезде анықтау мүмкіндігі. және туберкулезбен күрестің жеңілдеуі;
* емдеуге жатқызу деңгейінің төмендеуі және орташа деңгейдегі реттелген аурулар үшін лайықты өмір сүру сапасының қамтамасыз етілуі;
* бастапқы мен стационарлық деңгейлер арасындағы медициналық көмек көрсету бірлестігінің қамтамасыз етілуі;
* әлеуметтікке мақсатталған медицинаның дамытылуы.

Аталған тариф компоненті сайтына кіру <https://skpn.eisz.kz> сілтемесі арқылы іске асырылады [8].

«Жүкті әйелдер және фертильді жастағы әйелдер регистрі» АЖ - сі

«Жүкті әйелдер және фертильді жастағы әйелдер регистрі» АЖ - сін жасаудың басты мақсаты жүкті әйелдер және фертильді жастағы әйелдер топтарының денсаулық жағдайларының көрсеткіштеріне мониторинг жасау болып саналады. Жүйенің қолданушылары амбулаториялық деңгейіндегі медициналық ұйымдар жұмысшылары болып саналады (акушер - гинекологтар немесе терапевттер секілді).

«Жүкті әйелдер және фертильді жастағы әйелдер регистрі» АЖ негізгі функциялары келесідей болады:

* «жүкті мен босанған әйелдердің жеке картасы» есептік обьектісі негізінде жүкті әйелдердің топтарын орнату;
* фертильді жастағы әйелдер топтарын орнату;
* шығу обьектілерін жүргізу.

Жүйеге кіру <https://www.eisz.kz> сілтемесі бойынша іске асырылады.

«Жедел коронарлық синдром регистрі» АЖ

«Жедел коронарлық синдром регистрі» АЖ жүрек - қантамыр жүйелеріндегі кездесетін аурулары болатын емделушілерді тіркеуге, емделушілерді ары қарай бақылауға, диагностикалау алгоритмдері және емдеу хаттамаларының бұзылуын анықтау мен тіркеуге, басқарушылық шешімдерді жедел қабылдау үшін статистикалық және талдамалық есептілікті қалыптастыруға арналған.

«Жедел коронарлық синдром регистрі» АЖ басты функциялары келесідей:

* стационарлық деңгейдегі жүрек - қантамыр жүйелерінің аурулары болатын емделушінің электрондық картасын қадағалау;
* статистикалық пен аналитикалық кестелердің есептерін орындау.

Жүйеге кіру <https://www.eisz.kz> сілтемесі бойынша іске асырылады.

# **2 ВЕБ - ТЕХНОЛОГИЯЛАРДЫҢ ҚОЛДАНУ АЙМАҒЫНДАҒЫ ӘДІСТЕРІ**

## **2.1 Web - қосымшаның серверлік бағдарламау тілі - Python**

Python бүгінде ең танымал бағдарламалау тілдерінің бірі болып табылады. Таңқаларлық емес, көптеген бастаушы программисттер Python бағдарламалау тілін үйрену қиын ба және оны оқудың мағынасы бар ма деген сұрақтар мазалайды.

Python әр түрлі ақпараттық экожүйелерге енеді, бірақ көбінесе ол серверлік бөлігінде қолданылады. Google, Facebook, Stripe, Spotify, Netflix сияқты алыптары оны өз өнімдерінде қолданады. Сондықтан, Python бағдарламалау тілі бағдарламашы үшін мансаптың жақсы бастауы бола алатындығына күмән жоқ.

### **2.1.1 Python бағдарламалау тілі**

Python танымалдылығының себептері

“Интерпретируемость” – «Питонның» басты артықшылықтарының бірі. Есімізге түсіретін болсақ, интерпретация - бұл кодты құрастырудың қажеті жоқ бөлігі, бірақ оны бірден іске қосуға болады.

Python - дағы кодты кез - келген компьютерде іске қосуға болады және оған арналған аудармашылар барлық жерде бар [17]. Бұл дегеніміз, әзірлеуші нәтижені бірден көре алады. Алайда, түсінікті тіл бола отырып, Python құрастырылған тілдерге қарағанда жылдамдық жағынан төмен (өйткені ол тікелей компьютерде жұмыс істемейді).

* “интерпретируемость” тестілеуді және платформадан шағын код блоктарының платформасына ауысуды айтарлықтай жеңілдетеді. Көптеген операциялық жүйелермен үйлесімділік Python - ды әмбебап бағдарламалау тіліне айналдырады;
* бұл бағдарламалауды бастағандар үшін өте қолайлы. Ол жоғары деңгейлі тіл әзірлеушіге нақты қадамдар мен процедураларға емес, мәні мен мақсаттарына назар аударуға мүмкіндік береді. Осы себепті Python - да сценарийлер жазу көптеген басқа тілдерге қарағанда тезірек;
* оның табиғи тілге (ағылшын) ұқсастығы Python - ды басқа бағдарламалау тілдерімен салыстырған кезде оны үйренуді жеңілдетеді. Әзірлеушіге "питон" синтаксисін есте сақтау және кодты оқуды бастау оңай.
* шағын қолданбалы скриптардан басқа, Python - да үлкен коммерциялық қосымшаларды жаза алуға болады және оларды қажетті функционалдылықпен қалай қамтамасыз ету туралы ойлану қиын емес. Сенімділік Python бағдарламалау тілін өте танымал етеді.

Жоғарыда қөрсетілген артықшылықтар бағдарламашылар, хакерлер және дата - сайентистер тарапынан "питонға" деген сүйіспеншіліктің төрт себебі.

Python - ды таңдаудағы негізгі фактор – бұл бағдарламалау тілінің икемділігі және оның OББ парадигмасына сәйкестігі. Сондықтан ол Data Science және машиналық оқытуда белсенді қолданылады.

Python - ды қолданудың артықшылықтары мен кемшіліктері

Python бағдарламалау тілінің артықшылықтары:

* үйренудің қарапайымдылығы. Python бағдарламалау тілін нөлден үйрену оңай, өйткені ол анық, қисынды және тез оқылады. Күрделі синтаксис және шатастыратын ережелер «питон» туралы емес;
* тілге деген жоғары сұраныс**.** 2020 жылы IEEE Spectrum рейтингінде Python бірінші орынға ие болды. Бұл кең таралған ғана емес, сонымен қатар әртүрлі бағдарламалық өнімдерде де қолданылады. Python бағдарламашыларының табысы өте жақсы, әсіресе Web - әзірлеу, машиналық оқыту және Data Science;
* Python - ды жасанды интеллект және машиналық оқыту технологияларында қолдану. Python көмегімен күрделі есептеулер жүргізу ыңғайлы, сондықтан ол машиналық оқыту кезінде пайдалы болды. Ал нейронды жүйелерге жауапты кітапханалар жасанды интелект есептеріне өте ыңғайлы болып келді;
* Python бағдарламалау тілінде компьютер бағдарламасын, мобильді және Web - қосымшаны жазуға болады. «Питон» микро және макро жобаларды құру үшін көптеген кеңейтімдерге ие;
* ашық бастапқы код және ресми сайттағы тілдің ақысыз қол жетімділігі. Сонымен қатар, әркім Python - ның дамуына өз үлесін қосып, оған жаңа фичи қоса алады;
* тіл ОЖ - ге тәуелсіз және негізгі платформаларда - Linux, Windows, macOS - та бірдей тиімділікпен жұмыс істейді.

Python кемшіліктері:

* Баяу жұмыс. «Питон» жоғары өнімді қосымшаларды құру үшін нашар жұмыс істейді, бірден басқа тілді таңдаған дұрыс;
* Процессорды оңтайландыру үшін өте аз мүмкіндіктер бар. Оларды тілдің жадының арнайы моделі жоққа шығарады.

Бұл тілдің негізгі кемшіліктері - жылдамдықтың төмендігі және статикалық кодты талдау мүмкіндігінің болмауы. Бұл проблемалар бір - бірімен тығыз байланысты. Егер екіншісі шешілсе, онда бірінші шешім ашылады.

Python - ның қолдану аясы

1. Web - әзірлеу

Web - сайттың барлық сервер бөлігін «питонмен» жазуға болады. Бірақ таза Python - да емес, олар өз кезегінде танымал фреймворктерде (Django, Flask) жазылады. Бұл фреймворктар пайдаланушы көретін html - беттерін құру процестерін, дерекқорға сұраныстарды, мекен - жайларды өңдеуді жеңілдетеді.

Бүгінгі күні Web - қосымшаларды іске асыру үшін көптеген қосымша құралдар жазылды. Мысалы, үшінші тарап әзірлеушілерінің HTMLGen авторлығымен Python - да html - бетіне дайын кластар қосуға болады [18]. Ал mod\_python пакеті Apache сценарийлерін іске қосуға көмектеседі және Python Server Pages шаблондарының тұрақты жұмыс істеуін қамтамасыз етеді.

1. көрнекі интерфейс

Графика саласында көптеген мәселелер Python бағдарламалау тілінің көмегімен де шешіледі. Егер жасалған графикалық интерфейсті бағдарлама жұмыс істейтін операциялық жүйенің стиліне бейімдеу қажет болса, онда Python - ның қосымша Dabo және PythonCard графикалық кітапханаларымен қолдануға болады, ал бұл даму процесін едәуір жеңілдетеді.

1. деректер базасы

«Питонның» заманауи нұсқасы кез - келген мәліметтер базасымен мүмкіндігінше қарапайым және нақты әрекеттесетін етіп жасалған. Атап айтқанда, тілдің жұмыс ортасында SQL сұраныстары арқылы тікелей сценарийде жұмыс істеуге арналған бағдарламалық интерфейс бар. Python коды, егер оны Oracle және MySQL дерекқорлары үшін пайдалану қажет болса, минималды нақтылауды қажет етеді.

1. жүйелік бағдарламалау және басқару

Python - да Linux, Windows және т. б. жұмыс істейтін әртүрлі ОЖ қызметтерін басқаруға арналған интерфейстер бар. Сондықтан, «питонда» компьютерге арналған портативті қосымшаларды жазу ыңғайлы. Python бағдарламалау тілін жүйелік әкімшілер өз бағдарламаларын жазу үшін ұзақ уақыт бойы қолданып келеді. Python арқылы жүйелік файлдарды ашу мен іздеуді, бағдарламаларды іске қосуды, есептеулерді және басқа типтік тапсырмаларды тездетуге болады.

1. күрделі есептеулер

Есептеу саласында Python бағдарламалау тілі C++ және FORTRAN - нан кем емес. Математикалық есептеулерді жүргізу үшін теңдеулермен, деректер массивтерімен және олардың басқа форматтарымен жұмыс істейтін NumPy кеңейтімі жасалды. Осы кеңейтімді компьютерге орнатқан сәттен бастап «питон» бірден формула кітапханаларымен біріктіріледі.

Алайда, NumPy - ді есептеулерден басқа, анимациялар жасау және 3D нысандарын көрсету үшін де қолдануға болады. ScientificPython деп аталатын Python - ға тағы бір танымал қосымша ғылыми есептеулерге арналған өз кітапханалары бар. Яғни, Python есептеулерге ғана емес, сонымен бірге алынған деректерді визуализациялауға да көмектеседі.

1. машиналық оқыту

Машиналық оқытуда Python тілінің негізгі құралдары ғана емес, сонымен қатар рамкалар, сонымен қатар машиналық оқыту үшін арнайы «қайралған» қосымша кітапханалар белсенді қолданылады. Олардың ішіндегі ең танымалдары - TensorFlow және scikit - learn.

Python бағдарламалау тілінде машиналық оқытуға арналған функционал жазылады және оның көмегімен дауыстар мен тұлғаларды тану жүйелері жұмыс істейді. Python - ды терең оқыту мамандары мен нейрондық желілерді жасаушылар қолданады.

1. бизнес - процестерді автоматтандыру

Python қолданылатын ең танымал артықшылықтардың бірі - бірқатар жұмыс процестері мен стандартты процедураларды автоматтандыру үшін қысқа сценарийлер жазу. Мысалы, бұл кіріс хаттарды автоматты түрде өңдейтін шағын код. Ол оларды берілген негізгі тіркестердің болуымен сұрыптайды, ал бұл пайдаланушының өмірін айтарлықтай жеңілдетеді.

Мұндай сценарийлерді бірнеше жолдан бағдарламалау үшін Python тілінің тиімділігінің құпиясы неде деп сұралынса, бұл, ең алдымен, қарапайым және түсінікті синтаксис, оған сценарийлер жасау өте қызықты. Екіншіден, құрастыру кезеңінің болмауы және кодты бірден іске қосу және жөндеу мүмкіндігі.

1. Геймдев

Көптеген адамдар ойын индустриясына мән бермейді және ол бекер. Ойын индустриясы көптеген ыңғайлы және пайдалы гаджеттердің пайда болуына әкеліп соқты, сандық графика мен басқа да әзірлемелердің дамуына қуатты серпін берді. Python - да үлкен жоба жасау мүмкін емес - өйткені ол өте қарапайым орын алады және функциялардың тар жиынтығын орындайды.

Алайда, егер «питонның» жанкүйері болсаңыз, онда оған кішкене қосымшалар мен инди - ойыншықтарын жазуға болады. Мультиплатформалық ойындар туралы айтатын болсақ, әдетте C# - де жазылған Unity қозғалтқышы таңдалады бұл құрал тек ойындарды дамыту үшін жасалған.

Python - ды үйрену барысында қарапайым тапсырмаларды орындау арқылы қолды сынап көруі керек және қарапайым скриптерден бастаған жөн [19]. Бірте - бірте бағдарламалаудың нақты бағыты таңдалынып және оған мамандандырылса, дағдылар мақсатты түрде «арттырылады» және болашақ мансап құрылады.

### **2.1.2 Python web - фреймворктары**

Python фреймворктері кез - келген әзірлеушінің өмірін едәуір жеңілдетеді және тиімді қосымшалар мен web - сайттар құруға кең мүмкіндіктер береді. Көптеген шешімдерді оңай автоматтандырады, процестерді жеделдетеді және күнделікті тапсырмаларды азайтады.

Шын мәнінде, олардың функционалдығы айтарлықтай өзгереді. Кейбіреулері толыққанды күрделі көп деңгейлі даму үшін жасалған, ал басқаларында жеке модульдер мен блоктарды жасауға мүмкіндік беретін жеңілдетілген интерфейс бар. Қандай ортада жұмыс істеуді таңдамас бұрын, тапсырмаларды анықтап, мақсаттарды түсіну керек. Бұл жағдайда нақты не пайдалану керектігі белгілі болады.

**1) Django**

Django – еркін ашық коды бар толық full - stack фреймворгі. Бұл Python әзірлеушілері арасында ең негізгісі және танымалы. Ол прототиптен қысқа уақыт ішінде дайын жұмыс шешіміне көшуге көмектеседі, өйткені оның негізгі міндеті - процестерді автоматтандыру және бірлестіктер мен кітапханалар есебінен жұмысты жеделдету. Көптеген шешімдер дайын түрде болады, яғни оларды еш қиындықсыз қолдану керек.

Артықшылығы

Djangoның көптеген артықшылықтарға ие. Онда көптеген дайын шешімдер бар, ал бұл бағдарламаны құруды айтарлықтай жеңілдетеді. Әкімшілік тақтасы, мәліметтер базасын көшіру, әртүрлі формалар, пайдаланушының аутентификация құралдары бәрі дайын. Құрылым өте түсінікті және қарапайым.

Үлкен қауымдастық кез - келген мәселені шешуге көмектеседі. Қауіпсіздіктің жоғары деңгейі, ыңғайлы сәулет, ORM арқасында мәліметтер базасымен ыңғайлы жұмыс атқарады.

Кемшіліктері

Үлкен мүмкіндіктерге қарамастан, Python Web Framework Django - да кемшіліктер бар. Бұл өте массивті, монолитті, сондықтан ол баяу іске асыралады. Көптеген әмбебап модульдерге қарамастан, Django - ның даму жылдамдығы төмендейді.

Қолдану аймағы

Егер кем дегенде бірнеше тезис айтатын болсақ, Django - ны қолданудың келесідей мүмкіндіктері бар:

* API - нің серверлік бөлігін дамыту қажет болса;
* Web - қосымшаны жасау керек болса;
* кез - келген уақытта қосымшаны үлкейту және азайту бағытында масштабтау қажет болуы мүмкін;
* көптеген күрделі міндеттер бар болғанда, оларды өз бетіңізше шешу қиын жағдайында және қоғамдастықтың көмегі қажет болса;
* дерекқорға тікелей кірмеу үшін ORM қолдауы қажет болса;
* Машиналық оқыту сияқты жаңа технологияларды біріктіру қажеттілігі бар болса.

Бұл жағдайда Django керемет Python Web Framework болады және ол өз міндетін орындайды. Бұл таңқаларлық емес, ең танымал және миллиондаған әзірлеушілер белсенді оны қолданады.

**2) CherryPy**

Python фреймворктары full - stack және микро - фреймворк болып бөлінеді. CherryPy екінші типке жатады. Ол кез - келген операциялық жүйеде бағдарламаны іске қосуға қабілетті нақты мәселелерді шешуге арналған.

Артықшылығы

CherryPy фреймворкі ыңғайлы және түсінікті даму ортасына ие. Ол жақсы қосымшаларды жинауға болатын функционалды және толық фреймворк. Бастапқы код ашық, сондықтан платформа әзірлеушілер үшін толығымен тегін, ал қауымдастық тым үлкен болмаса да, өте жауапты және әрқашан мәселелерді шешуге көмектеседі.

Кемшілігі

Осы Python фреймворкінің кемшіліктері онша көп емес. Ол күрделі міндеттер мен функцияларды орындай алмайды, нақты, нүктелі шешімдерге, мысалы, белгілі бір плагиндер немесе модульдерді жасауға арналған. «Қораптан тыс» мүмкіндіктері кең емес.

Қолдану аясы

CherryPy келесі жағдайларда қолданылады:

* шағын өлшемді коды бар қосымшаны жасау керек болса;
* қосымшаның функционалдығы жүктелмеген, әдетте бір немесе бірнеше тапсырманы орындау жағдайында;
* бір уақытта бірнеше серверлерді басқару қажет болса;
* қосымшалардың жұмысын бақылау керек болса;
* андроид қосымшаларын әзірлеуде.

CherryPy нақты мәселелерді шешуге арналған Python Frameworks - ке жатады. Ол түсінікті, ыңғайлы.

**3) Pyramid**

Python Pyramid фреймворкі күрделі нысандарды бағдарламалауға және көп функциялы мәселелерді шешуге арналған. Оны кәсіби бағдарламашылар пайдаланады, дәлірек айтқанда, дәстүрлі түрде сәйкестендіру және бағыттау үшін қолданылады. Ол кең аудиторияға бағытталған және API прототиптерін жасауға қабілетті.

Артықшылығы

Pyramid негізгі қосымшаларды жылдам әзірлеуде өте жақсы. Үйрену өте икемді. Негізінде, бұл фреймворктің сәттілігінің кілті - ол толығымен негізгі принциптерге негізделген, қарапайым және негізгі бағдарламалау техникасы қолданылады. Бұл минималистік, бірақ сонымен бірге пайдаланушыларға кең еркіндік ұсынады. Шағын қосымшалармен де, қуатты көп функциялы бағдарламалармен де жұмыс істей алады.

Кемшіліктері

Негізгі принциптерден кету қиын. Бұл Python фреймворкі сіз үшін шешім қабылдайды деген сөз. Жеңіл бағдарламаларды жүзеге асыру өте оңай. Бірақ күрделі және ауқымды нәрсе жасау үшін басқа да фреймворктерді зерттеуге толығымен кірісіп, оған бағынуыңыз керек.

Қолдану аймағы

Pyramid фреймворкі келесідей қажет болған жағдайларда қолданылады:

* уақтылы түзетулер мен түзетулер енгізу үшін проблема индикаторларының құралдары үшін;
* бірден бірнеше бағдарламалау тілдерін қолдану кезінде;
* есептілік және қаржылық есептеулер, болжау үшін;
* тұрақты жұмыс туралы есеп бергенде, өйткені ол автоматты түрде жасалады;
* өте тез қарапайым қолданбаны жасауда.

Сонымен қатар, Python Web Framework Pyramid үлкен функционалдығы бар күрделі қосымшаларды құруға мүмкіндік береді.

4) Grok

Grok - бұл шаблондармен жұмыс істейтін Python фреймворкі. Оның негізгі міндеті - кодта қайталанбаушылықты қамтамасыз ету. Егер элемент қайта пайдаланылса, онда бұрын жасалған шаблон ғана қолданылады. Бұл жұмысты айтарлықтай жеңілдетеді және тездетеді.

Артықшылығы

Grok негізі - бұрын шығарылған Zope3 - тің «туындысы». Ол жеңілдетілген жұмыс құрылымына, модульдерді оңай орнатуға, кеңейтуге және икемділікке ие. Ол бірнеше қарапайым тапсырмалары бар шағын қосымшаларды жасау үшін жасалған. Ол күрделі жұмысқа арналмаған, бірақ оның функционалдығына байланысты жобаны тез жүзеге асыруға мүмкіндік береді.

Кемшіліктері

Grok қауымдастығы тым үлкен емес, өйткені ол кең танымал бола алмады. Алайда, оны Python бағдарламаушылары ыңғайлы әзірлеу үшін қолданады. Онда күрделі тапсырмаларды жүзеге асыру мүмкін емес. Құрамдас бөліктер мен компоненттік архитектураның жұмыс принципіне қарамастан, мүмкіндіктері өте шектеулі.

Қолдану аймағы

Grok келесідей жағдайларда әзірлеушілерге жарамды болады:

* бағдарламашылардың тәжірибесі аз және олар модульдерін жасауға әлі дайын емес, құрамдас бөліктерді қолдану оңайырақ кезінде;
* қарапайым қосымшаны тез әзірлеу қажеттілігі бар болса;
* қосымшаның функционалдығы қарапайым, түсінікті және интерфейс маңызды рөл атқармайтын болса.

Grok қарапайым болса да, ең жақсы Python Web Frameworks тізіміне енеді. Себебі ол түсінікті және ыңғайлы әзірлеу үшін жеткілікті мүмкіндіктерге ие.

TurboGears

TurboGears - бұл Web - қосымшаларды жасауға арналған ашық кодты фреймворк. Ол сапалы қызықты қосымшалар жасауға мүмкіндік береді, ыңғайлы шаблондарға ие, икемді ORM - ы бар және көптеген деректер базасын қолдайды. Ол әмбебап даму ортасы болып саналады және жаңадан бастаушыларға да, тәжірибелі бағдарламашыларға да ыңғайлы.

Артылықшылығы

TurboGears пәрмен жолының элементтеріне ие, көптеген мәліметтер базасын қолдайды және әртүрлі қызметтермен жұмыс істей алады. Бұл әзірлеуді едәуір жылдамдатуға көмектеседі және кодтардың саны азаяды, осылайша қосымшалар тезірек жұмыс істейді және қателіктер ықтималдығы айтарлықтай төмендейді. Оның жаңа виджет жүйесі қуатты және ауыр қосымшаларды құруды жеңілдетеді.

Кемшіліктері

Аталған Python фреймворгі үнемі жетілдіріліп, дамытылып отырады. Бірақ бастапқыда ол «асығыс» құрылды, осыған байланысты бірқатар жағымсыз пікірлер жиналды. Әрине, барлық жақсартулар жасалды және бүгінгі күні фреймворк жақсы жұмыс істейді.

Қолдану аймағы

TurboGears жиі қолданылады, әсіресе келесідей жағдайларда:

* салыстырмалы түрде қысқа мерзімде қуатты "ауыр" қосымшаны жасау керек болса;
* икемділік пен әртүрлі мәліметтер базасын пайдалану мүмкіндігі қажет болса;
* Formencode көмегімен тексеру қажет болса;
* командалық жол құралдарына қажет болса.

TurboGears Django - ға балама болып табылады және осы даму ортасымен үнемі бәсекелеседі. Қалай болғанда да, 2021 жылы осы Python Web Framework - не сұраныс жоғары болды.

**5) Web2Py**

Web2Py – бұл код редакторы, дебаггер, депла кіретін жеке IDE - сі бар Python кеңейтілетін фреймворгі. Ол өте жақсы жұмыс істейді, оны баптау немесе орнату қажет емес, деректердің жоғары қауіпсіздігін қамтамасыз етеді және әртүрлі платформаларда жұмыс істеуге жарамды.

Артықшылығы

Web2Py әртүрлі протоколдармен жұмыс істей алады, кірістірілген қате трекері бар және фреймворктің алдыңғы нұсқалары негізінде жұмыс істеуге көмектесетін кері үйлесімділік функциясы бар. Бұл кодты сақтау әлдеқайда жеңіл және арзан болады дегенді білдіреді. Ол тегін, яғни, ашық коды бар және өте икемді.

Кемшіліктері

Көптеген Python Web Frameworks арасында тілдің соңғы нұсқасын қолдануды қажет ететіндер көп емес. Web2Py - солардың бірі және Python 3 және одан төмен нұсқаларда жұмыс істемейді. Сондықтан жаңартуларды үнемі қадағалап отыру керек.

Қолдану аймағы

Web2Py келесідей жағдайларға керемет сәйкес келеді:

* әр түрлі операциялық жүйелерде даму қажеттілігі бар болса;
* жақтауды орнату және конфигурациялау мүмкіндігі жоқ болса;
* қаржылық қосымшаларды әзірлеу кезінде деректердің жоғары қауіпсіздігі қажет болса;
* қателерді тестілеу кезеңінде емес, даму барысында мұқият бақылау керек болса.

Web2Py өз міндеттерін өте жақсы орындайды, өте қарапайым және бәріне қол жетімді.

6) Flask

Flask - бұл модульдік дизайны бар микро - фреймворк. Ол көптеген қосымшаларды жасауға мүмкіндік береді, ал даму жылдамдығы қораптық шешімдерге байланысты үлкен. Flask кеңейтімдерді таңдау мүмкіндігі бар ыңғайлы Python фреймворгі.

Артылықшылығы

Flask - те кіріктірілген дебаггер және өз сервері бар, көптеген шаблондар мен дайын шешімдер әзірлеу процесін жеделдетеді, қауіпсіз кукиді қолдайды, сонымен қатар кез - келген ORM - ге қосыла алады.

Кемшіліктері

Олар іс жүзінде жоқ. Ол нақты міндеттер үшін арнайы жасалған. Жалғыз кемшілікті тым кең мүмкіндіктердің болмауы деп атауға болады, бірақ бұл кеңейтімдермен жұмыс істеуге және әзірлеушілерді қолдауға арналған микро - фреймворк. Көбінесе қосымша құрал ретінде қолданылады.

Қолдану аймағы

Flask - ты келесідей жағдайларда қолдануға болады:

* API құру қажеттілігі бар болса;
* RESTful қызметтерін дамыту үшін қолайлы кезде;
* қосымшаның серверлік бөлігі әзірленгенде.

Flask - нақты тапсырмаларға арналған Python Web Framework - і. 2022 жылы өзекті болатын көптеген бағдарламашылар үшін керемет қосымша құрал.

7) Bottle

Бастапқыда Bottle тек API құру үшін жасалды, бірақ болашақта функционалдылық біршама кеңейтілді. Көптеген функциялары - шаблондар, бағыттау, утилиталар және басқалар әдепкі бойынша қосылады. Бұл прототиптеу үшін өте ыңғайлы Python фреймворгі.

Артықшылығы

Bottle - де көптеген функцияларды қолдайтын кірістірілген HTTP әзірлеу сервері бар. Бұл кішігірім қосымшалар мен қарапайым прототиптерді жасау үшін өте ыңғайлы. Деректерге ыңғайлы қолжетімділік, файлдарды жүктеу мүмкіндігі, кукимен жұмыс істеу - фреймворктың толыққанды жұмысын қамтамасыз ететін құралдардың бір бөлігі ғана. Сондай - ақ, дамуды жеңілдететін және тездететін «қораптан тыс» шешімдер бар.

Кемшіліктері

Барлық Python Frameworks бірдей ауқымды әзірлеуге бағытталған емес, ал Bottle - солардың бірі. Ол үлкен жобаларға жарамайды, көптеген функциялармен жұмыс істей алмайды және күрделі операцияларды орындай алмайды. Оның міндеті - жеңіл функционалдығы бар жоғары сапалы жылдам шағын қосымшаларды жасау.

Қолдану аймағы

Bottle келесідей жағдайларда қолданылады:

* прототиптік қосымшаны тез жасап, негізгі мүмкіндіктерді көрсету керек болса;
* басқа даму орталарына енгізілуі мүмкін шағын модульдерді бөлек жасау керек болса;
* қосымша қарапайым және күрделі функционалдылықты қажет етпейтін кезде.

Bottle - қарапайым жұмыс үшін керемет фреймворк және қарапайым элементтерді тез құруға мүмкіндік береді.

8) Tornado

Tornado бір уақытта 10 000 - нан астам қосылыстарға төтеп бере алатын ауыр және жаппай қосымшалар үшін өте қолайлы. Бұғатталмайтын желілік енгізу - шығару арқасында келушілердің жоғары трафигі үшін мүмкіндіктерді кеңейтуге мүмкіндік беретін С10к мәселесі осы фреймворкте шешіледі.

Артықшылығы

Tornado - ның басты артықшылығы – көптеген сұраныстарды өңдеу және жоғары қуатты қосымшаларды әзірлеу мүмкіндігі. Бұл тек Python Web Framework ғана емес, сонымен қатар асинхронды желілік кітапхана. Үшінші тараптың авторизация жүйелері оңай қосылады, аудармаларды қолдау, өнімділіктің жоғары деңгейі бар. Ол күрделі қосымшаларды құруға арналған қуатты даму ортасы.

Кемшіліктері

Аталған фреймворкты үйрену уақытты қажет етеді. Қауымдастық өте дамыған, бірақ көптеген мүмкіндіктер көрсетілмеген, сондықтан кейбір мәселелерді шешуге уақыт қажет.

Қолдану аймағы

Tornado келесідей жағдайларда қолданылады:

* нақты уақыт режимінде нәтижелерді бақылау керек болса;
* үлкен жүктеме мен сабаққа қатысу қосымшасы жасалатын болса;
* үшінші тараптың аутентификация және авторизация қызметтерін қосу керек болса;
* қосымша бірден бірнеше тілде жасалатын болса.

Tornado - мамандарға арналған 2022 жылы Python - да әзірлеушілер арасындағы қуатты құрал болып есептелінеді.

9) BlueBream

BlueBream - дің басқа Zope3 атауы бар. Ол орташа және жоғары деңгейдегі тапсырмаларды өте жақсы орындайды және күрделі жобалармен жұмыс істеуге жарамды.

Артықшылығы

BlueBream құрастыру жүйесі өте күшті және күрделі мәселелерді шешуге жарамды. Онда функционалды қосымшаларды жасауға болады, ал компоненттерді қайта пайдалану принципі кодты жеңілдетеді. Бұл жағдайда даму жылдамдығы артады. Бағдарламалық жасақтама кеңейтіліп, транзакциялық объектілік дерекқорды пайдалану оны сақтаудың оңай жолын ұсынады. Бұл дегеніміз, сұраулар тез өңделеді және мәліметтер базасымен жұмыс істеу оңай [20].

Кемшіліктері

BlueBream орыс тілді қоғамдастықта өте танымал емес, оны өзіңіз зерттеуге тура келеді. Бұл әсіресе икемді фреймворк емес, одан не талап етілетінін алдын - ала білу керек. Сонымен қатар, ол үлкен жүктемеге төтеп бермейді. Бір уақытта 1000 пайдаланушымен жұмыс жасағанда, ол істен шығып, қателіктер жіберуі мүмкін. Сондықтан осал мәселелерді шешу үшін қолданылады.

Қайда қолданады

BlueBream Мұндай жағдайларда қолданылады:

* үлкен функционалдығы бар қуатты салмақты қосымшаны жасау қажет болған кезде;
* қолданбаға жүктеме үлкен болмауы керек, пайдаланушылардың бір уақытта саны жұмыс проблемаларын болдырмау үшін шектелуі керек;
* құжат айналымды процесін реттеу қажет, бұл мәселемен осы фреймворк өте жақсы жұмыс істейді.

Python фреймворктары көбінесе белгілі бір мәселелерді шешуге арналған. BlueBream - солардың бірі, ол мәліметтер базасымен жұмыс істеу маңызды рөл атқаратын қосымшаларға сәйкес келеді.

### **2.1.3 Web - қосымша құруға арналған Python кітапханалары**

Төменде web - әзірлеуге арналған бірнеше пайдалы Python кітапханалары берілген:

* Егер қолданба үшін деректерді шығару үшін іздеу роботы қажет болса, Scrappy бұл үшін өте жақсы. Бұл кітапхана көбінесе тазалау, деректерді іздеу, автоматтандырылған тестілеу және т.б. үшін қолданылады.
* Zappa – бұл AWS Lambda - да серверсіз қосымшаны жасауға арналған қуатты кітапхана.
* Requests – бұл HTTP сұрауларын оңай жіберуге мүмкіндік беретін кітапхана. Ол HTML беттерін немесе деректерді алуға мүмкіндік беретін қосымшамен байланыс үшін қолданылады.
* Dash – бұл деректерді визуализациялауға қатысты Web - қосымшаларды жасаушыларға арналған тағы бір пайдалы кітапхана. Фреймворк диаграммаларды, графиктерді, бақылау тақталарын және басқаларын ұсынады.

### **2.1.4 Python көмегімен web - қосымшаны құрудың этаптары**

Python көмегімен web - дамытуға арналған іс - әрекеттер келесідей болады:

1 - қадам: HTML + CSS

Web - дамытудың басында web - сайттар құру кезінде оқытудың негізі болып табылатын HTML және CSS - ті үйрену керек. Web - даму жолын бастау үшін жауап беретін статикалық беттерді құрылымдауды үйрену әлдеқайда маңызды. Сондай - ақ, интернет, HTTP, шолғыш, DNS, хостинг және басқалар сияқты ұғымдар туралы білу пайдалы [21].

Соған қоса, Materialize немесе Bootstrap сияқты CSS фреймворктер туралы білу жөн болады. Олар жұмысты едәуір жылдамдатады.

2 - қадам: Javascript

Келесі жасау керек қадам - Vanilla Javascript - ті үйрену. Деректер түрлері, айнымалылар, жалпы келісімдер, жолдармен жұмыс істеу, арифметика, басқару операторлары, циклдар және т. б. сияқты негізгі ұғымдармен танысу керек. Осы негіздерді білу JavaScript - ті клиенттік кодқа қолдануды жеңілдетеді.

3 - қадам: DOM & jQuery

Содан кейін тағы бір Javascript кітапханасын - jQuery - ді басқаруды үйрену керек. Оның көмегімен DOM - ны басқару процесі әлдеқайда жеңіл болады. Нәтижесінде динамикалық беттерді қалай құру туралы түсінік болады.

Фронтенд - фреймворк

Функционалды толық стек web - қосымшасын жасау үшін фронтенд фреймворкті зерттеу ұсынылады. Мұндай шеберлік көбінесе front - end немесе full - stack әзірлеушісін жалдау кезінде міндетті талап болып табылады.

4 - қадам: Python

Енді серверлік бөлік жасалынады. DOM - мен манипуляцияны бастамас бұрын, Python және Javascript негіздерін білу керек. Аталған негізгі білім Django - ге дайындайды. Python - ды үйрену қиын емес, өйткені көптеген ұғымдар Javascript - ты еске түсіреді.

5 - қадам: Django + мәліметтер базасы

Django көмегімен бэкенд - ортаны орнатып, бизнес логикасын қалыптастыра алуға болады. Сондай - ақ, сұраныстарды қалай құру, SQLite дерекқорын және CRUD функциясын зерттеу туралы білу керек. Нәтижесінде толық стек қосымшасын жасай алуға мүмкіндің бар.

## **2.2 Web - қосымшаның фреймворкі - Django**

Django – бұл қауіпсіз және қолдау көрсетілетін web - сайттарды тез құруға мүмкіндік беретін жоғары деңгейлі Python web - фреймворкі. Тәжірибелі әзірлеушілер жасаған Django web - дамудың көптеген қиындықтарын өз мойнына алады, сондықтан жаңа нәрселерді ойлап таппай - ақ, web - қосымшаны жазуға назар аудара алуға уақыт көбейеді. Бұл ақысыз және ашық, өсіп келе жатқан және белсенді қоғамдастық, керемет құжаттама және ақысыз және ақылы қолдаудың көптеген нұсқалары бар.

### **2.2.1 Django фреймворкі ерекшеліктері**

Django - ның бағдарламалық жасақтаманы жазудағы келесідей ерекшеліктері бар:

Толық

Django «барлығы қосылған» философиясын ұстанады және әзірлеушілер «қораптан тыс» жасағысы келетін барлық нәрсені ұсынады. Қажет барлық нәрсе бір «өнімнің» бөлігі болғандықтан, бәрі бірге жұмыс істейді, дизайнның дәйекті қағидаларына сәйкес келеді және кең және өзекті құжаттамаға ие.

Жан - жақты

Django әртүрлі web - қосымшларды, мазмұны бар басқару жүйелері және wiki - ден бастап әлеуметтік желілер және жаңалықтар сайтына дейін құру мақсатында қолданылады. Фреймворк түрлі клиенттік орталармен жұмыс жасай алады және әртүрлі форматта құрылатын қосымшаның мазмұнын жеткізе алады. Django қажет кез - келген функционалдылыққа арналған шешімдерді ұсынады (айталық, бірнеше танымал мәліметтер базасы, шаблондар және т.б.), сонымен қоса керек болған кезде оны үшінші тарап компоненттері кеңейте алады.

Қауіпсіз

Django сайттар әзірлеушілеріне сайтты автоматты түрде қорғау мақсатында «дұрыс затты жасауға» құрылған фреймворкты ұсына отырып көпшілік жалпылама қауіпсіздік қателіктерін болдыртпауға көмек береді. Мысалы, Django пайдаланушының есептік жазбалары мен құпия сөздерін басқарудың қауіпсіз әдісін ұсынады. Сонда сеанс туралы ақпаратты осал жерде cookie файлдарына орналастырырады немесе пароль хэшінің орнына құпия сөздерді тікелей сақтайды [22].

Пароль хэші – бұл криптографиялық хэш функциясы арқылы парольді өңдеу арқылы жасалған бекітілген символдар мәні. Django енгізген парольді хэш функциясы арқылы өткізіп, нәтижені сақталған хэш мәнімен салыстыру арқылы тексере алады. Функцияның «бір жақты» сипатына байланысты, сақталған хэш мәні бұзылған болса да, шабуылдаушыға бастапқы парольді анықтау қиынға соғады.

Django әдепкі бойынша көптеген осалдықтардан қорғайды, соның ішінде SQL инъекциясы, сайт аралық сценарий, сайт аралық сұраулар және кликджекинг секілді.

Масштабталады

Django компоненттік «ортақ - ескерту» архитектурасын қолданады. Бөлшектерді нақты бөлу арқылы, Django кез - келген деңгейде жабдықты қосу арқылы трафикті көбейте алады, яғни, кэштеу серверлері, дерекқор серверлері немесе қосымшалар серверлерін арттырады. Ең көп жүктелген сайттардың көбісі Django - ны сәтті масштабтайды (айталық, Instagram мен Disqus сайттары).

Сүйемелдеуде ыңғайлы

Django коды қолдау көрсетілетін және қайта пайдаланылатын кодты құруға көмектесетін принциптер мен дизайн шаблондарын қолдана отырып жазылған. Атап айтқанда, ол «Don't Repeat Yourself» (DRY, «қайталамаңыз») принципін қолданады, сондықтан қажет емес қайталану жоқ, бұл код көлемін азайтады. Django сонымен қатар байланысты функцияларды қайта пайдаланылатын «қосымшаларға» топтастыруға ықпал етеді және төменгі деңгейде байланысты кодты модульдерге топтайды (Model View Controller (MVC) үлгісіне сай).

Портативті

Django көпшілік платформаларда жұмыс жасайтын Python тілі негізінде құрастырылған. Бұл дегеніміз, белгілі бір серверлік платформаға қосылмаған және Linux, Windows және Mac OS X - тің көптеген нұсқаларында қосымшаларды іске қосуға болады. Бұған қоса, Django - ға көптеген web - хосттар мықты қолдау жасайды және олар көбіне Django - да құрылған сайттарды орналастыру үшін нақты бір инфрақұрылымдар мен құжаттаманы ұсынып жатыр.

Web - фреймворкті көбінесе "икемді" және "икемсіз"деп бөлуге болады.

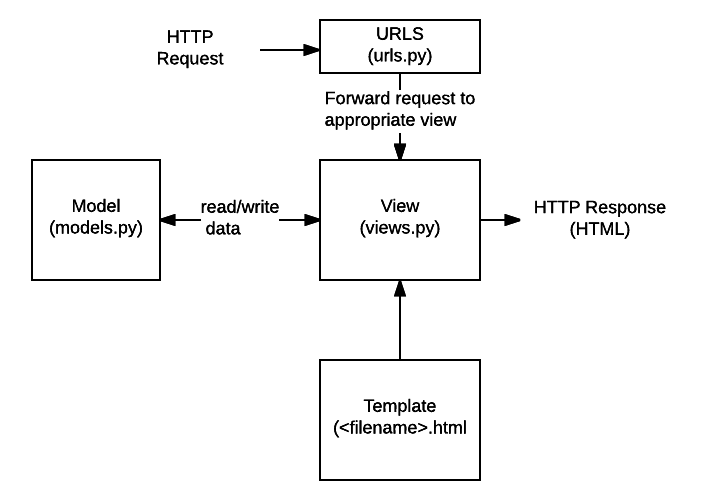
Икемсіз - бұл белгілі бір мәселені шешудің "дұрыс жолы" бары. Олар көбінесе белгілі бір аймақта жылдам орналастыруды қолдайды (белгілі бір типтегі мәселелерді шешу), өйткені кез - келген нәрсені жасаудың дұрыс әдісі әдетте жақсы түсініледі және жақсы құжатталады. Алайда, олар негізгі саладан тыс мәселелерді шешуде икемді болмауы мүмкін және олар қандай компоненттер мен тәсілдерді қолдана алатындығының аз нұсқаларын ұсынады.

Керісінше, икемді фреймворкте мақсатқа жету үшін немесе қандай компоненттерді қолдану керек болса да, компоненттерді байланыстырудың жақсы әдісіне шектеулер аз болады. Олар әзірлеушілерге белгілі бір тапсырманы орындау үшін ең қолайлы құралдарды қолдануды жеңілдетеді, дегенмен бұл компоненттерді жеке табу керек.

Django «орташа икемді», сондықтан «екі әлемнің ең жақсысын» қамтамасыз етеді. Ол көптеген web - әзірлеу тапсырмаларын өңдеуге арналған компоненттер жиынтығын және оларды қолданудың бір немесе екі әдісін ұсынады. Алайда, мұндай Django архитектурасы әдетте бірнеше түрлі опцияларды таңдай алу немесе қажет болған жағдайда жаңа нұсқаларға қолдауды қоса алу дегенді білдіреді.

### **2.2.2 Django кодының жалпы көрінісі**

Дәстүрлі ақпараттық web - сайтта web - шолғыштан (немесе басқа клиенттен) HTTP сұрауларын күтеді. Сұрау алынған кезде, бағдарлама URL - мекен - жайы және POST немесе get сұрауларындағы мәліметтер негізінде қажет нәрсені жасайды. Не талап етілетініне байланысты, ол мәліметтер базасынан ақпаратты оқи не жаза алады немесе сұранысты қанағаттандыру үшін қажет басқа тапсырмаларды орындай алады. Содан кейін бағдарлама жауапты web - шолғышқа қайтарады. Көбінесе HTML - бетті динамикалық түрде шолғышта көрсету үшін жасайды, алынған деректерді HTML - шаблонына енгізеді «2.1-суретте» көрсетілген.



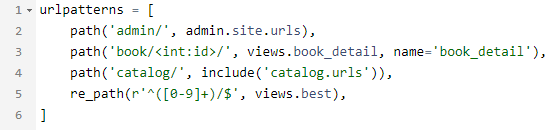
2.1 - сурет. Django кодының жалпы көрінісі

Django - да жазылған web - қосымшалар, әдетте, келесідей қадамдардың әрқайсысын жеке файлдарға өңдеп, жазылатын кодты топтайды:

* URLs: Әр URL - мекен - жайынан сұрауларды бір функциямен өңдеуге болатынына қарамастан, әр ресурсты өңдеу үшін бөлек функция жазу әлдеқайда ыңғайлы. URL - маршрутизаторы HTTP - сұраныстарын сұрау URL - мекен - жайы негізінде тиісті көрініске бағыттау үшін қолданылады. Сонымен қатар, URL - маршрутизаторы берілген шаблонға сәйкес URL - мекен - жайынан деректерді шығарып, оларды тиісті дисплей функциясына дәлелдер түрінде жібере алады.
* View: View - бұл HTTP - сұрауларын алатын және жауаптарды қайтаратын сұрау өңдегішінің функциясы. View функциясы сұраныстарды қанағаттандыру үшін қажетті деректерге қол жеткізе алады және жауаптарды үлгілер арқылы шаблондарға жібереді.
* Models: Модельдер - Python нысандары, олар қосымшаның деректер құрылымын анықтайды және басқару тетіктерін ұсынады (қосу, өзгерту, жою) және дерекқорға сұраныстарды орындауға қарастырады.
* Templates: Template - нақты мазмұнды көрсету үшін пайдаланылатын қойылмалы шеттері бар беттің құрылымын немесе орналасуын анықтайтын мәтіндік файл (мысалы, HTML - беті). View HTML - шаблондарын қолдана отырып және оларды модельден (model) мәліметтермен толтыру арқылы HTML - беттерін динамикалық түрде жасай алады. Шаблонды HTML - ға ғана емес, кез - келген типтегі файл құрылымын анықтау үшін пайдалануға болады.

### **2.2.2.1 Дұрыс View(urls.py) - ге сұранымдарды жіберу**

URL – мекен - жайы идентификаторы әдетте urls.py файлда болады. Төмендегі мысалда салыстырушы (urlpatterns) маршруттары мен сәйкес дисплей функциялары арасындағы салыстырулар тізімін анықтайды. Егер белгілі бір шаблонға сәйкес келетін URL - мекен - жайы бар HTTP - сұранысы алынса, онда байланысты дисплей функциясы шақырылып, сұрауға жіберіледі.



Urlpatterns нысаны path () және/немесе re\_path() функцияларының тізімі ретінде алынады(Python - да тізімдер төртбұрышты жақшалармен анықталады, олардың ішінде элементтер үтірмен бөлінген және қосымша аяқталатын үтір болуы мүмкін, мысалы: [item1, item2, item3,]).

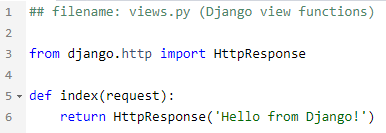
Path() әдісінде бұрыштық жақшалар URL - мекен - жайының түсірілген және көрсетілген аргументтер ретінде дисплей функциясына берілетін бөліктерін анықтау үшін қолданылады. Re\_path() функциясы үлгі өрнегі деп аталатын үлгіге сәйкес келудің икемді тәсілін қолданады.

Views.book\_detail белгілеуі функцияның book\_detail() деп аталатынын көрсетеді және оны views деп аталатын модульден табуға болады (яғни, аты views.py бар файлдың ішінде).

### **2.2.2.2 Сұранымдарды өңдеу(views.py)**

Көрініс(views) – webклиенттерден HTTP - сұрауларын қабылдайтын және HTTP - жауаптарын қайтаратын web - қосымшаның жүрегі. Сонымен қатар, олар мәліметтер базасына, визуализация үлгілеріне және т. б. қол жеткізу үшін басқа фреймворк ресурстарын пайдаланады.

Төмендегі мысалда алдыңғы бөлімдегі URL - мекен - жай салыстырушысынан туындауы мүмкін ең аз index() ұсыну функциясы көрсетілген. Барлық дисплей функциялары сияқты көрініс (view) функциясы Httprequest нысанын параметр ретінде алады және HttpResponse нысанын қайтарады. Бұл жағдайда сұраумен ештеңе істелінбейді және осындағы жауапты қатаң бағдарламаланған жолды қайтарады.

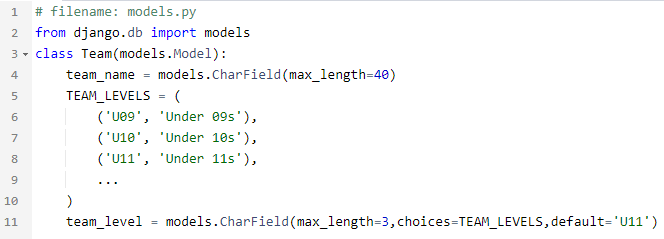


Көрініс (view) әдетте views.py файлда болады.

### **2.2.2.3 Қосымшаның деректерін анықтау (models.py)**

Django web - қосымшалары модельдер деп аталатын Python нысандары арқылы деректерді өңдейді және оларға сұраныc жібереді. Модельдер сақталатын деректердің құрылымын анықтайды, оның ішінде өрістердің түрлері және мүмкін олардың максималды мөлшері, әдепкі мәндер, таңдау тізімінің параметрлері, құжаттама үшін анықтама мәтіні, пішіндер үшін белгі мәтіні және т. б. Модельдің анықтамасы пайдаланылатын дерекқорға байланысты емес, яғни, модельдер олардың кез - келгенінде жұмыс істейді. Пайдалану керек келетін дерекқорды таңдағаннан кейін, оған тікелей байланысудың қажеті жоқ, яғни, жай ғана модель құрылымын және басқа код жазылады, ал Django автоматты түрде дерекқорға жүгінудің барлық «лас жұмыстарын» жасайды.

Төмендегі код үзіндісі Team нысаны үшін өте қарапайым Django моделін көрсетеді. Team класы models класынан мұра алады. Model пәрмен атауы мен пәрмен деңгейін таңбалар өрісі ретінде анықтайды және әр жазба үшін сақталатын таңбалардың максималды санын анықтайды. Team\_level бірнеше мәндердің бірі болуы мүмкін, сондықтан ол таңдау өрісі ретінде анықталады және көрсетілген опциялар мен сақталған деректер арасында әдепкі мәнмен салыстыруқамтамасыз етіледі.

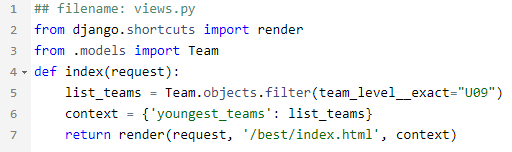


### **2.2.2.4 Деректерге сұраным жасау(views.py)**

Django моделі дерекқорды іздеу үшін қарапайым API сұрау ұсынады. Іздеуді әр түрлі критерийлерді (мысалы, exact ("дәл"), case - insensitive ("регистрді есептемегенде"), greater than ("көп") және т. б.) қолдана отырып, бір уақытта бірнеше өрістер арқылы жүргізуге болады және күрделі өрнектерді қолдай алады (мысалы, U11 командаларында іздеуді көрсетуге болады, команданың аты "Fr" - ден басталады немесе "al" - мен аяқталады).

Төмендегі код үзіндісі барлық U09 командаларын көрсету үшін view (ресурстар өңдегіші) функциясын көрсетеді. Бөлінген қалың жол барлық жазбаларды сүзгілеу үшін API - сұрау моделін қалай қолдана алатынынын көрсетеді, мұнда team\_level өрісінде 'U09' мәтіні бар (бұл критерийлердің filter() функциялары өріс атауы мен сәйкестік түрі бар аргумент ретінде қалай берілетініне назар аудару қажет: team\_level\_exact).

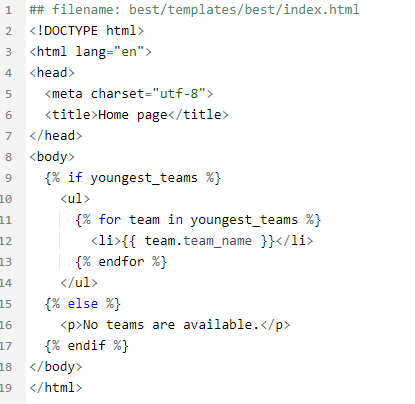
Бұл функция браузерге кері жіберілетін HttpResponse құру үшін render() функциясын пайдаланады. Бұл функция төте жол болып саналады. Ол көрсетілген HTML - шаблонын және кейбір деректерді шаблонға енгізу үшін біріктіру арқылы HTML - файлын жасайды («context» деп аталатын айнымалыда берілген).



### **2.2.2.5 Деректерді көрсету (HTML)**

Үлгі жүйелері бетті құру кезінде енгізілетін мәліметтер үшін толтырғыштарды қолдана отырып, шығыс құжатының құрылымын көрсетуге мүмкіндік береді. Шаблондар көбінесе HTML жасау үшін қолданылады, бірақ сонымен қатар басқа құжат түрлерін жасай алады. Django «қораптан тыс» өзінің шаблондық жүйесін де, Jinja2 деп аталатын басқа танымал Python кітапханасын да қолдайды (оны қажет болған жағдайда басқа жүйелерді қолдау үшін де қолдануға болады).

Төмендегі код үзіндісі жоғарыда айтылған render() функциясымен шақырылған HTML - шаблонының қалай көрінетінін көрсетеді. Бұл шаблон сурет салу кезінде youngest\_teams деп аталатын тізім айнымалысына қол жеткізе алады деген болжаммен жазылған(ол жоғарыдағы render() функциясының ішіндегі контекстік айнымалым). HTML қаңқасының ішінде алдымен youngest\_teams айнымалысының бар - жоғын тексеріп, содан кейін оны for циклде қайталайтын өрнегі болады. Әр қайталау кезінде шаблон <li> элементіндегі әр команданың team\_name мәнін көрсетеді.



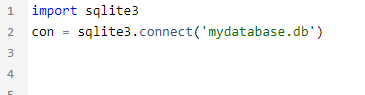
## **2.3 Деректер қорымен жұмыс жасау технологиясы - SQLite3**

SQLite3 - бұл жеке серверлік процесті қажет етпейтін және SQL сұрау тілін қолдана отырып дерекқорға кіруге мүмкіндік беретін жеңіл дискілік дерекқорды жүзеге асыратын кітапхана. Кейбір қолданбалар ішкі деректерді сақтау үшін SQLite қолдана алады. Сондай - ақ, SQLite көмегімен қосымшаның прототипін жасауға болады, содан кейін кодты PostgreSQL немесе Oracle сияқты көп функциялы дерекқорға беруге болады.

Sqlite3 модулі DB - API сипаттамаларына сәйкес келетін 2.0, PEP 249 сипатталған SQL интерфейсін жүзеге асырады.

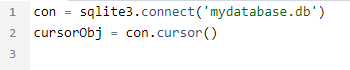
Байланыстарды құру жолы

Python - да SQLite3 пайдалану үшін Sqlite3 модулін импорттау керек, содан кейін дерекқорға қосылу нысанын жасау керек. Қосылу нысаны connect() әдісі арқылы жасалады:



Sqlite3 курсоры

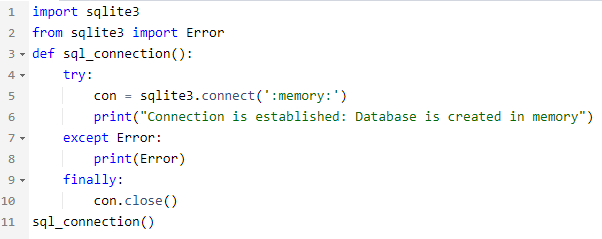
SQL мәлімдемелерін орындау үшін cursor() әдісімен жасалған курсор нысаны қажет. Sqlite3 курсоры қосылу нысаны әдісі. SQLite3 мәлімдемелерін орындау үшін алдымен қосылым орнатылады, содан кейін қосылым нысанын қолдана отырып курсор нысаны жасалады:



Енді кез - келген SQL - сұрауларын орындау үшін execute() әдісін шақыру үшін курсор нысанын пайдалануға болады.

Деректер базасын құру

SQLite - пен байланыс жасағаннан кейін, деректер базасы файлы болмаған жағдайда автоматты түрде жасалады. Бұл файл дискіде жасалады, бірақ connect әдісіндегі «: memory:" параметрін қолдана отырып, жедел жадта мәліметтер базасын құруға болады. Төмендегі кодты қарастырсақ, онда кез - келген ерекшеліктерді өңдеу үшін try, except және finally блоктары бар мәліметтер базасы жасалынды:



Sqlite3 модулі алдымен импортталады, содан кейін sql\_connection деп аталатын функция анықталады. Функция ішінде try блогы анықталған, онда connect() әдісі байланыс орнатылғаннан кейін қосылым нысанын қайтарады.

Содан кейін ерекшеліктер блогы анықталады, ол кез - келген ерекшелік жағдайында қате туралы хабарламаны басып шығарады. Егер қателер болмаса, байланыс орнатылады, содан кейін сценарий «Connection is established: Database is created in memory» мәтінін басып шығарады. Әрі қарай, байланыс finally блогында жабылады. Қосылымды жабу міндетті емес, бірақ бұл кез - келген пайдаланылмаған ресурстардан жадты босатуға мүмкіндік беретін жақсы бағдарламалау тәжірибесі.

Кесте құру

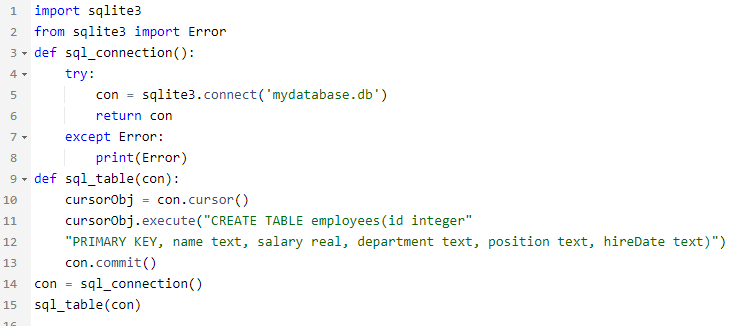
SQLite3 - те кесте құру үшін execute() әдісінде кесте құру сұрауы орындалады. Ол үшін келесі қадамдар тізбегі орындалады:

1. Қосылу нысанын жасау.
2. Cursor нысаны қосылу нысаны арқылы жасалады.
3. Курсор нысанын қолдана отырып, параметр ретінде құру кестесін сұрау арқылы execute әдісі шақырылады.

Келесі Employees бағандар жұмыс кестесін құрылсын:



Ол үшін код келесідей болады:



Жоғарыдағы кодта екі функция анықталған. Біріншісі - қосылымды орнатады, ал екіншісі - SQL оператор create table курсор нысанын қолдана отырып, объект құру кестесін орындайды.

Commit () әдісі барлық өзгерістерді сақтайды. Сценарийдің соңында екі функция да шақырылады. Кестелердің болғанын тексеру үшін деректер қоры - Sqlite3 - ті қолдайтын браузерін қолданамыз.

Кестеге деректерді енгізу

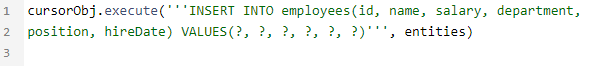
Кестеге деректерді енгізу үшін INSERT INTO операторын қолданылады.

Кодтың келесі жолы осы процесті қарастырады:



Сондай - ақ, execute() әдісіндегі INSERT мәлімдемесіне мәндерді/дәлелдерді жібере алуға болады. Әрбір мән үшін толтырғыш ретінде сұрақ белгісін (?) де қолдануға болады.

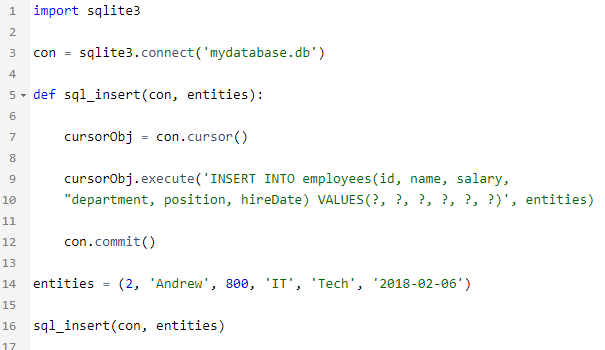
INSERT синтаксисі келесідей болады:



Мұнда entities картежі кестеде бір жолды толтыру үшін мәндерді қамтиды:



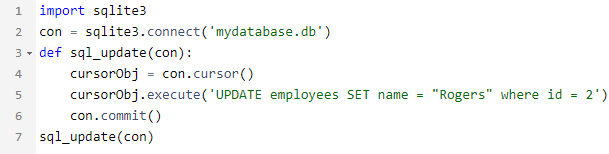
Код келесі бейнедегідей көрсетіледі:



Кестені жаңарту

Идентификаторы 2 - ге тең қызметкердің атын жаңарту керек делік. Жаңарту үшін UPDATE нұсқаулары қолданылады. Сондай - ақ, дұрыс қызметкерді таңдау үшін шарт ретінде WHERE предикаты қолданылады.

Келесі код қарастырылады:



Бұл Эндрюдің атын Роджерске өзгертеді.

SELECT операторы

SELECT операторы бір немесе бірнеше кестелерден деректерді таңдау үшін қолданылады. Егер кестеден барлық деректер бағандарын таңдау қажет болса, жұлдызшаны (\*) пайдалануға болады. Бұл үшін SQL синтаксисі келесідей болады:



SQLite3 - те таңдау нұсқаулары курсор нысанын орындау әдісінде орындалады. Мысалы, employee кестесінің барлық жолдары мен бағандары таңдалынады:



Егер кестеден бірнеше бағанды таңдау керек болса, оларды төменде көрсетілгендей бейнеленеді:



Мысалы,

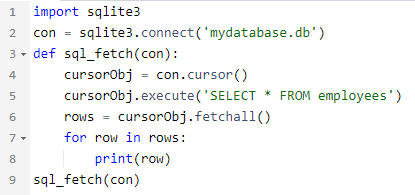


SELECT операторы деректер қорының жұмыс кестесінен барлық деректерді таңдайды.

Барлық деректерді іріктеу

Деректер базасынан деректерді алу үшін біз SELECT нұсқаулығын орындаймыз, содан кейін мәндерді айнымалыға сақтау үшін курсор нысанының fetchall() әдісін қолданамыз. Бұл жағдайда айнымалы мәліметтер базасының әр жолы тізімнің жеке элементі болатын тізім болады. Содан кейін айнымалы мәндерді санау және мәндерді басып шығару орындалады.

Код келесідей болады:

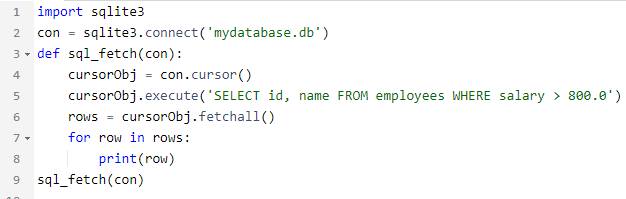


Сондай - ақ, бір жолда fetchall() операторын пайдалануға болады:



Егер деректер қорынан нақты деректерді алу қажет болса, WHERE предикатын қолданылады. Мысалы, жалақысы 800 - ден асатын қызметкерлердің идентификаторлары мен аттары таңдаланады. Ол үшін кестені көптеген жолдармен толтырылады, содан кейін сұранысты орындалады. Деректерді толтыру үшін INSERT мәлімдемесін қолдануға болады немесе оларды деректер базасының шолғыш бағдарламасына қолмен енгізіледі.

Енді жалақысы 800 - ден асатын қызметкерлердің есімдері мен идентификаторлары таңдалынады:



Жоғарыдағы SELECT мәлімдемесінде жұлдызшаның (\*) орнына id және name атрибуттары көрсетілген.

SQLite3 rowcount

SQLite3 жол есептегіші соңғы SQL сұранысы әсер еткен немесе таңдалған жолдардың санын қайтару үшін қолданылады.

Rowcount SELECT операторымен шақырылған кезде - 1 қайтарылады, өйткені таңдалған жолдардың саны олардың барлығы таңдалғанға дейін белгісіз болады. Келесі мысалды қарастыралық:



Сондықтан жолдардың санын алу үшін барлық деректерді алу керек, содан кейін нәтиженің ұзындығын алу керек:



DELETE операторы ешқандай шартсыз қолданылған кезде, кестедегі барлық жолдар жойылады және жойылған жолдардың жалпы саны rowcount арқылы қайтарылады.

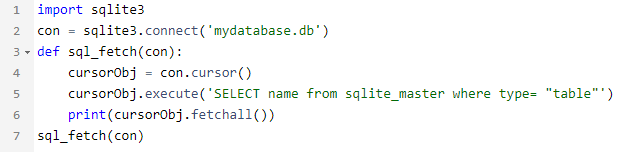


Егер ешқандай жол жойылмаса, 0 қайтарылады.

Кестелер тізімі

SQLite3 дерекқорындағы барлық кестелердің тізімін көрсету үшін sqlite\_master кестесіне сілтеме жасап, select операторынан нәтиже алу үшін fetchall() қолданылады.

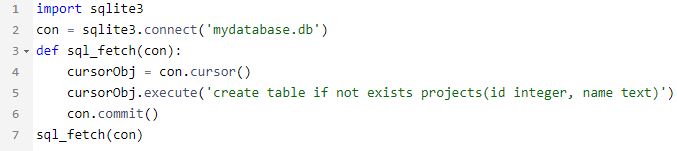
Sqlite\_master - бұл барлық кестелер сақталатын SQLite3 - тегі негізгі кесте.



Кестенің іске асуын тексеру

Кестені құру кезінде кестенің әлі жоқ екеніне көз жеткізу керек. Сол сияқты, кестені жойған кезде ол болуы керек.

Егер кесте әлі жоқ болса, тексеру үшін "if not exists" CREATE TABLE операторымен келесідей қолданылады:



Сол сияқты, кестені жою кезінде бар - жоғын тексеру үшін біз «if not exists» DROP TABLE нұсқаулығымен келесідей қолданылады:

Сондай - ақ, келесі сұрауды орындау арқылы қол жеткізуге болатын кесте бар - жоғын тексеріледі:



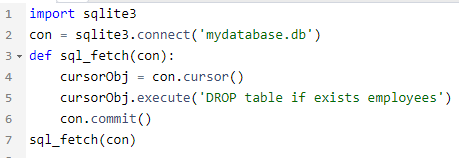
Егер көрсетілген кесте атауы болмаса, бос массив қайтарылады.

Кестені жою

Кестені жою DROP операторының көмегімен жүзеге асырылады. DROP операторының синтаксисі келесідей:



Кестені жою үшін кесте деректер қорында болуы керек. Сондықтан DROP операторымен «if exists» пайдалану ұсынылады. Мысалы, employees кестесін келесідей жойылады:



Sqlite3 “ерекшеліктері”

Басты ерекшелік - сценарий уақытының қателері. Python - да бағдарламалау кезінде барлық ерекшеліктер BaseException туынды класының экземпляры болып табылады.

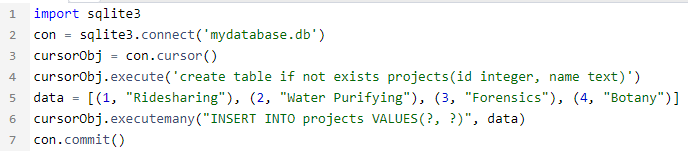
SQLite3 - те Python - ның келесі негізгі ерекшеліктері бар:

* DatabaseError. Деректер базасына қатысты кез - келген қате DatabaseError қатесін тудырады.
* IntegrityError.IntegrityError - бұл DatabaseError қосалқы класы және деректердің тұтастығы туралы мәселе туындаған кезде пайда болады, мысалы, барлық кестелерде сыртқы деректер жаңартылмаған кезде, бұл деректердің сәйкес келмеуіне әкеледі.
* ProgrammingError. ProgrammingError ерекшелігі синтаксистік қателер болған кезде пайда болады немесе кесте табылмаған немесе функция параметрлердің/дәлелдердің дұрыс емес санымен шақырылған кезде пайда болады.
* OperationalError. Бұл ерекшелік дерекқор операциялары сәтсіз болған кезде пайда болады, мысалы, ерекше ажырату кезінде. Бағдарламашының кінәсінен емес.
* NotSupportedError. Деректер базасы анықтамаған немесе қолдамайтын кейбір әдістерді қолданған кезде NotSupportedError ерекшелік пайда болады.

Sqlite - ге жолдарды жаппай енгізу

Бір уақытта бірнеше жолды енгізу үшін executemany операторы қолданылады.

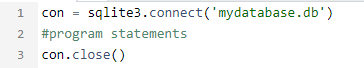
Келесі кодты қарастырсақ:



Мұнда екі бағаннан тұратын кесте жасалды, содан кейін "деректер" әр баған үшін төрт мәнге ие. Бұл айнымалы сұрау арқылы executemany() әдісіне беріледі. Мәндерді беру үшін толтырғыш қолданылғанын ескеріңіз.

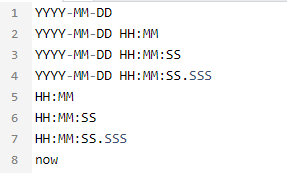
Байланысты жабу

Деректер қорымен жұмыс аяқталған кезде қосылымды жабу ұсынылады. Қосылымды close() әдісімен жабуға болады. Қосылымды жабу үшін close() әдісіне қоңырау шалу нысанын келесідей пайдаланылады:

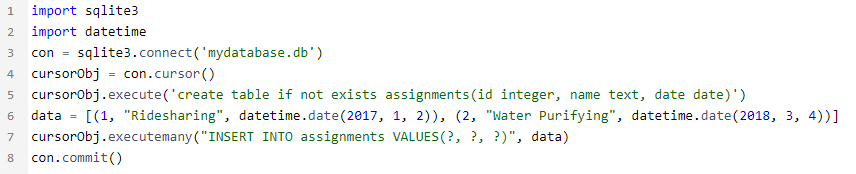


SQLite3 datetime

Python дерекқорында SQLite3 datatime модулін импорттау арқылы күнді немесе уақытты оңай сақтай алады. Келесі форматтар күн мен уақыт үшін жиі қолданылатын форматтар болып табылады:



Келесі кодты қарастырсақ:



Бұл кодта datetime модулі алдымен импортталады, содан кейін үш бағаннан тұратын assignments деп аталатын кесте жасалды. Үшінші бағанның деректер түрі - күн. Күнді бағанға енгізу үшін datetime.date қолданылды. Сол сияқты, сіз уақытты өңдеуге арналған datetime.time қолдана аласыз.

# **3 ЖЕДЕЛ КОРОНАРЛЫҚ СИНДРОМ РЕГИСТРІ ҮШІН АҚПАРАТТЫҚ ЖҮЙЕНІ ҚҰРУ**

## **3.1 Жедел коронарлық синдром регистрі web - қосымшасына қойылатын талаптар және қосмышаның жалпы жұмысын сипаттау**

Жедел коронарлық синдром регистрі ақпараттық жүйесі медициналық мекемелерде жедел коронарлық синдромымен ауыратын емделушілерге көрсетілетін көмектің есебі мен сапасын талдау құралы болып табылады.

ЖКС - пен ауыратын емделушілердің регистрлік деректер қорына қосудың келесідей критерийлері бар [16]:

* емделушінің жасы 18 - ге толған болу қажет;
* емделушінің аурулар тарихында ЖКС, миокарда инфарктісі диагноздары болу керек;
* ЖКС ауруынан айыққан болса;
* ЖКС - тан қайтыс болған болса;

ЖКС регистрін жүргізу үшін ақпараттық жүйені құру есебін анықтау қерек. Себебі кез - келген бағдарламалық өнімді құрудағы оның құрылымын, деректер қоры құрылымын және басқа да бағдарлама бөліктерін жобалау үшін ақпараттық жүйеге қойылатын нақты мақсаттары мен тапсырмалары білу маңызды.

Дипломдық жұмыстың мақсаты бойынша жедел коронарлық синдром регистрі ақпараттық жүйесін жобалау және программалау тілдері көмегімен осы жүйенің Web - қосымшасын жасау қажет. Web - қосымша көмегімен жүрек - қантамыр аурулары бар пациенттерді тіркеуге, пациенттерді одан әрі бақылауға, диагностика алгоритмдері мен хаттамаларының бұзылуын анықтауға және тіркеуге арналған.

ЖКС регистрі Web - қосымшасы алдына келесідей талаптар қойылады:

* емделушілерге толықтай ақпарат ұсына алу қажет;
* емделушілерге тіркелуге, содан соң жүйеге мүмкіндігі болу керек;
* жеке кабинетте емделуші өзі туралы ақпараттарды, дәрігергерлерге көріну жазбалары әрдайым қолжетімді болу қажет;
* дәрігер емделушунің ауру тарихын көре алуы, аурулар тарихына жазбалар енгізе алуы қажет;
* web - қосымшаның администраторы емделушілер және дәрігерлер туралы ақпаратты көре алуы, реттей алуы және емделушіге дәрігерге жолығуды ұйымдастыра алуы керек;
* администратор мен дәрігерлер үшін жүйеге бөлек тіркелу және кіру беттері болуы қажет;

Сонымен қатар жалпы алғанда Web - қосымшаны жобалауда сайттың келесідей беттері болады:

1. негізгі бет

* “Шапкада” - информациялық меню бөліктері - ЖКС деген не? Қалай алдын алуға болады? Ақпараттық байланыс контакттілері және тағы да бірнеше меню бөліктері болады.
* “Сайт денесінде” - ЖКС - на арналған бірнеше ақпараттық бөліктер, админ, пациент және дәрігерге сайтқа тіркелуге немесе кіруге арналған тіркелу/кіру формалары болады;
* “Футерда” - сайттың негізгі ақпараттық сілтемелері жинақталынады.

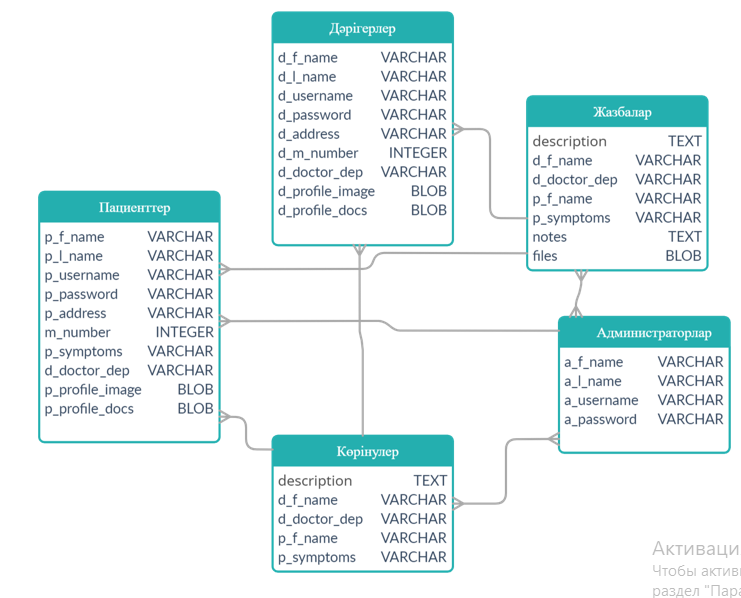
1. админка беті - бұл науқастарды тіркеуге, дәрігер қабылдауына жазуға, және т/б керекті іс әректерді жасауға арналған АДМИН БӨЛІМІ.
2. ациенттің жеке кабинеті - өзінің аурулары, дәрігерлеге көрінулері және дәрігерге жазылуға арналған іс - әрекеттері бар ЖЕКЕ КАБИНЕТ.
3. Дәрігердің жеке кабинеті - керекті іс - әрекеттері бар ЖЕКЕ КАБИНЕТ.

## **3.2 Web - қосымшаның деректер қорын құру**

ЖКС регистрін жүргізетін қосымшаның деректер қорын құру үшін алдымен осы жүйенің жұмыс жасау процестерін сипаттап алайық. ЖКС регистрі осы синдроммен ауыры мүмкін адамдардың алдын ала қаралуын немесе онымен ауыратын адамдардың диагностикадан өтуін және емделуін қадағалауға бағытталады.

ЖКС регистрі Web - қосымшасының обьект класстарын анықтап алайық. Оларға - Пациенттер, Дәрігерлер, Администраторлар, Көрінулер, Жазбалар жатады.

Аталған обьект класстары жалпы схемасы «3.1-суретте» көрсетілген.



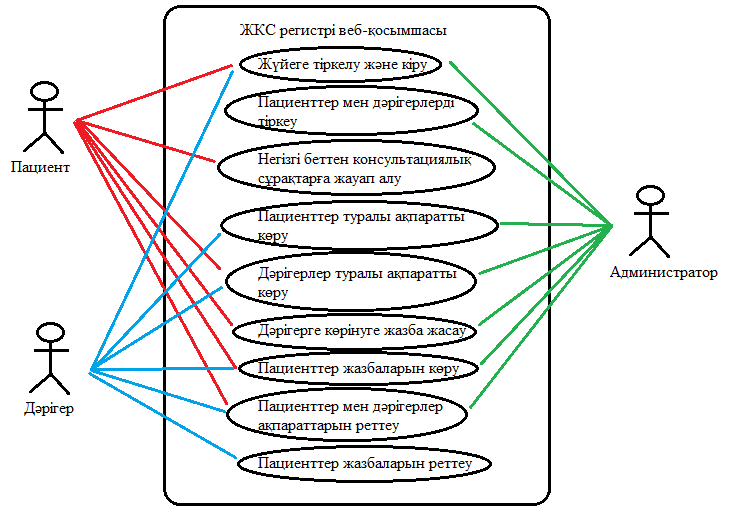
3.1-сурет. Деректер қорының жалпы схемасы

## **3.3 Қосымшаны жобалаудың диаграммалары**

Қолданудың таңдаулары диаграммасы

Қолданудың таңдаулары диаграммасы - қолданушылардың әр тобына құрылатын қосымшаның қандай функционалы қолжетімді екенін көрсететін диаграмма.

Жоғарыда жобалау бөлімінде айтып өткендей ЖКС регистрі Web - қосымшасында қолданушылардың 3 түрі - пациенттер, дәрігерлер және администраторлар болады. Осы қолданушылар үшін қолданудың таңдаулары диграммасы құрылды және ол келесі «3.2-суреттегідей» болады.

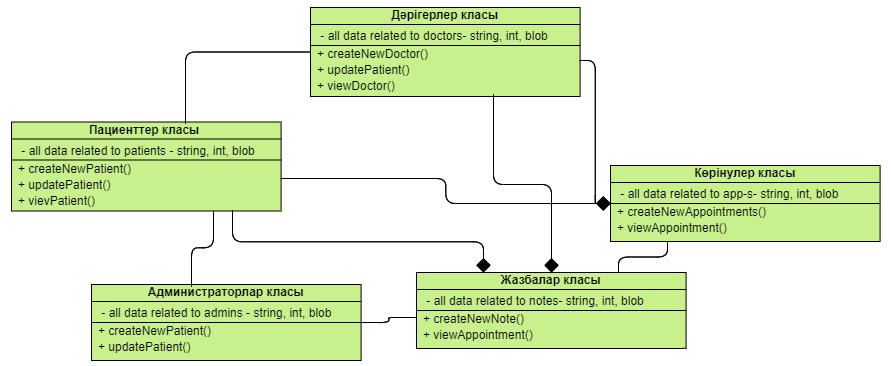


3.2-сурет. ЖКС регистрі Web - қосымшасының қолданулардың таңдаулары диаграммасы

Класстар диаграммасы

Класстар - бұл кез - келген обьектіге - бағытталған жүйенің негізгі элементтерінің бірі. Кластар біртекті обьектілердің олардың қасиеттерімен - атрибуттар, операциялар, өзара арақатынас, мағыналармен біріктірілген жиындарының сипаттамасы.

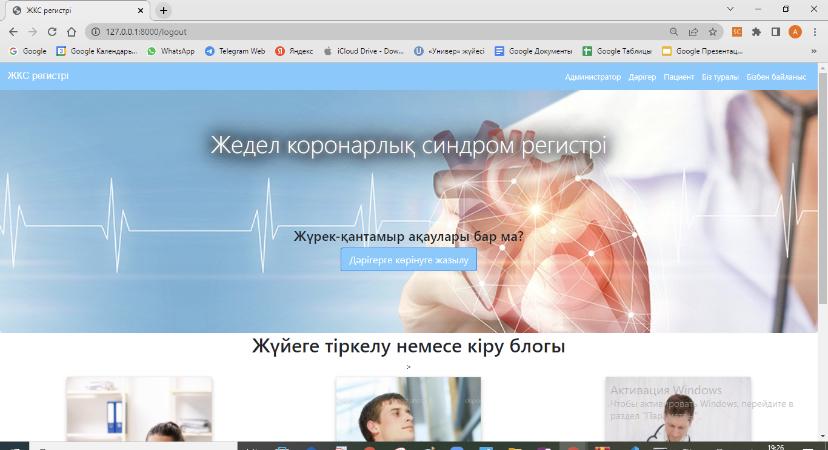
Кластар диаграммасы қосымшаның ішкі құрылымын класстар және олардың арасындағы байланыстар түрінде көрсетуге бағытталған. Деректер қорын жобалағанда, пациенттер, дәрігерлер, администраторлар, көрінулер, жазбалар секілді кластар анықталды. ЖКС регистрі Web - қосымшасы кластар диаграммасы келесідей болады «3.3-суретте көрсетілген».



3.3-сурет. ЖКС регистрі Web - қосымшасының класстар диаграммасы

## **3.4 Web - қосымшаның жұмыс істеу функцияналын сипаттау**

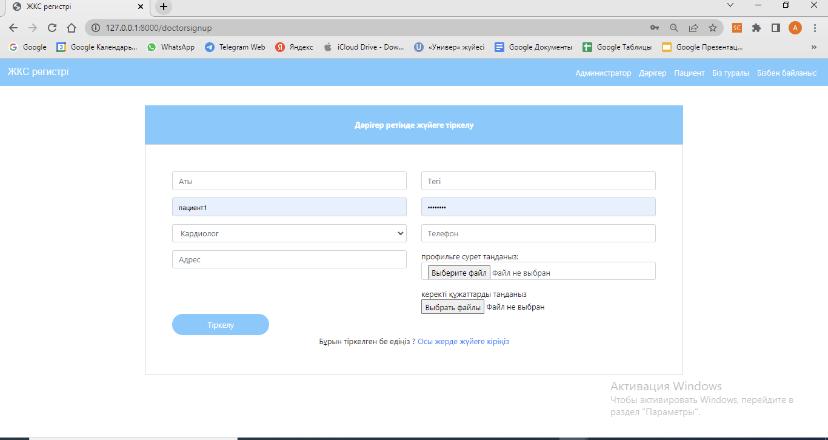
Web - қосымша компьютерлік серверде іске қосылады. Сайтты ашқан соң оның барлық функционалдарын қолдана аламыз. Негізгі функционалдары астындығы суреттерде бейнеленген. Негізгі беттің бейнесі «3.4-суретте» көрсетілген.



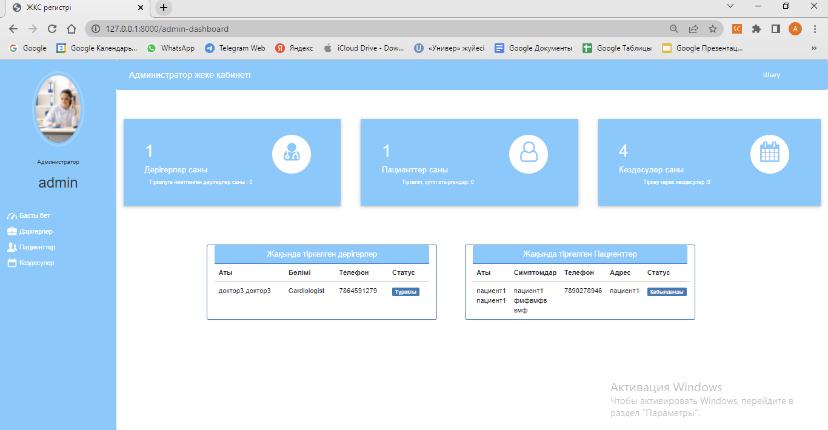
3.4-сурет. ЖКСР веб - қосымшасының негізгі беті

Сайттың функционалын сипаттау үшін әр қолданушы тобының жүзеге асыратын іс - әрекеттері негізге алынды. Яғни, мысалы, Пациент үшін жүйеге тіркелу немесе кіру, консультациялық сұрақтарға жауап алу, дәрігерлер туралы ақпаратты көру, дәрігерге көрінуге жазылу және өзі туралы ақпаратты реттеу іс - әрекеттері сайт көмегімен қалай жүзеге асатынын астыда баядалынатын болады [17].

Жүйеге тіркелу немесе тіркелу іс - әрекеті барлық қолданушылар - пациент, дәрігер және администратор үшін қолжетімді. Ол үшін сайттың patient - login, patient - signup беттеріне өтіп, кейін керекті деректерді толтырып жүйеге тіркелуге немесе кіруге болады «3.5-суретте» көрсетілгендей. Алайда пациент пен дәрігер жүйеге кіру үшін алдымен олардың тіркелуін администратор қарап, өзі қабыл алады, яғни тіркейді «3.6-суретте» көрсетілген. Содаң соң пациент те дәрігер де емін - еркін жүйені қолдана алады.



3.5-сурет. Дәрігердің тіркелу беті



3.6-сурет. Администратор жеке кабинеті негізгі беті

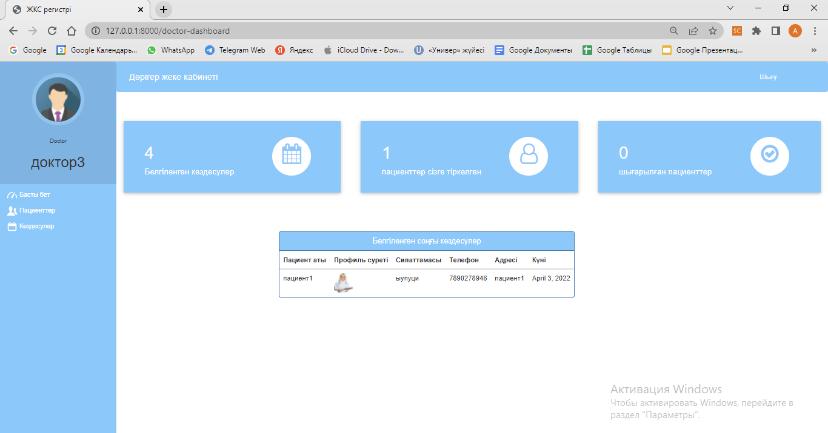
Негізгі беттен консультациялық сұрақтарға жауап алу іс - әрекеті негізен пациентке арналған. Пациент негізгі беттен ЖКС және ЖКС регистріне байланысты ақпаратты толықтай ала алады «3.7-суретте» көрсетілген.

Пациенттер және дәрігерлер туралы ақпаратты көру іс - әрекетін пациент те, дәрігер де және администратор да қолдана алады «3.8-суретте» көрсетілген. Пациент және дәрігер туралы ақпарат әр түрлі мақсатта қолданылуы мүмкін. Бұған қоса пациент те, дәрігер де өздеріне қатысты ақпаратты реттей алады және осы іс - әрекетті администратор да іске асыра алатын мүмкіндігі бар [18].

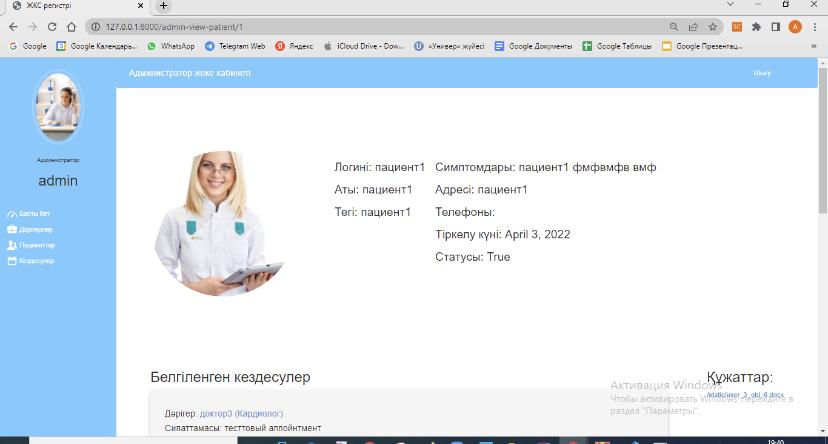
Дәрігерге көрінуге жазба жасау деп дәрігерге көрунуге электронды кезекке тұру деп түсінуге болады. Пациент дәрігерге көрінуге жазу үшін администратор көріну жазбасын тағайындай алады «3.9-суретте» көрсетілген [19].

Осы көріну жазбасы барлық қолданушыларға көрінетін болады «3.10-суретте» көрсетілген.

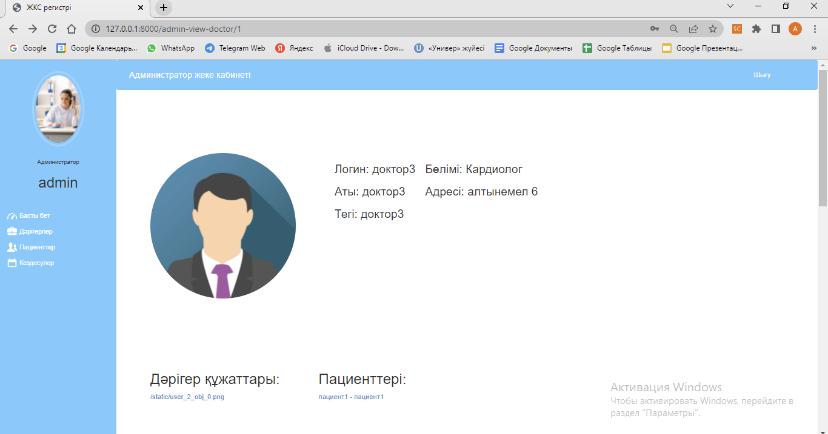
Кейін осы көріну жазбасын дәрігер ашып, керекті жазбаларды - диагноз, емделу хаттамаларын жаза алады және керекті құжаттарды тіркей алады «3.11-суретте» көрсетілген. Бұл пациенттер жазбаларын реттеу іс - әрекетіне жатады.



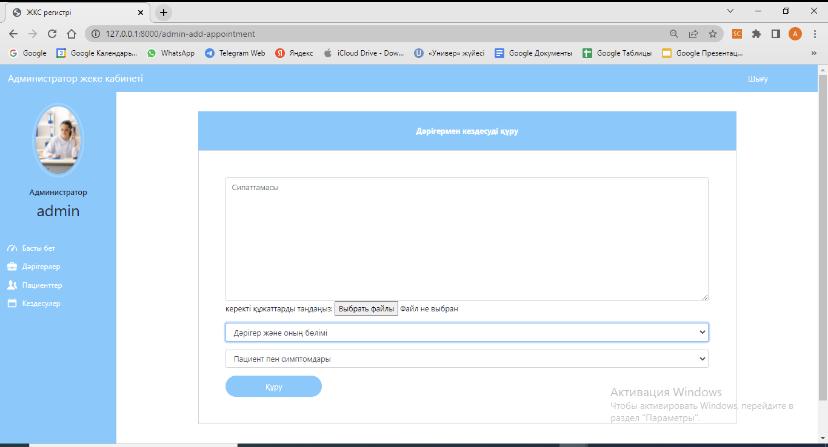
3.7-сурет. Дәрігер жеке кабинетінің негізгі беті



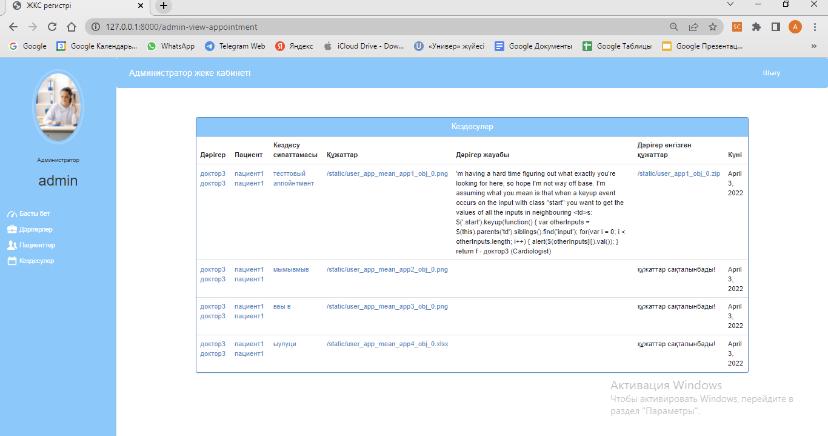
3.8-сурет. Администратор жеке кабинетінде пациент және дәрігер туралы ақпаратты көру беттері - а



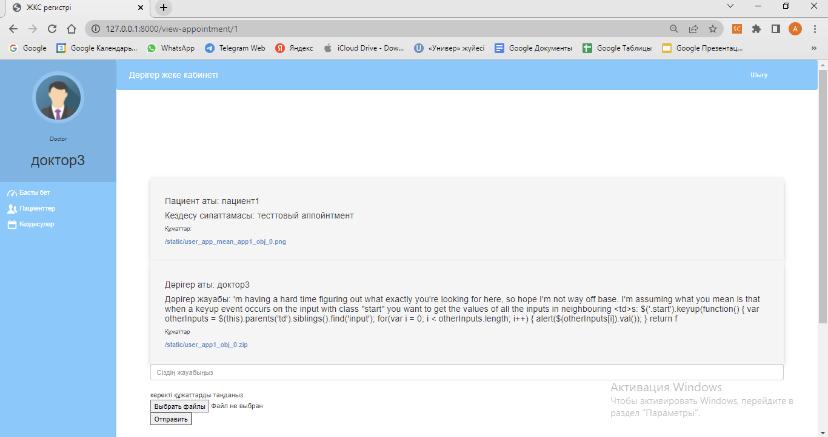
3.8-сурет. Администратор жеке кабинетінде пациент және дәрігер туралы ақпаратты көру беттері - б



3.9-сурет. Администратор жеке кабинетіндегі дәрігерге көріну жазбасын құру беті



3.10-сурет. Администратор жеке кабинетіндегі жазбаларды көру беті



3.11-сурет. Дәрігер жеке кабинетінде жазбаларды реттеу беті

# **Қорытынды**

Дипломдық жұмыстың мақсаты бойынша ЖКСР АЖ - сін жобалау және программалау тілдері көмегімен осы жүйенің Web - қосымшасын құру жұмыстың негізгі зерттеу нысаны болды. Медицина саласын ақпараттық технологиялармен қамтамасыз ету, басқаша айтқанда осы саланы цифрландыру казіргі әлемдік экономиканың басты мәселелерінің қатарына кіреді. Өйткені осындай технологияларды енгізу арқылы көрсетілетін медициналық көмектің сапасы артады, сәйкесінше халықтың денсаулық деңгейі өседі. Ал дені сау халықтың еңбекке деген қабілетін жоғарырақ болады. Сондықтан олардың экономикаға қосатын үлесі еселенеді. Ақпараттық технологияларды медицинаның жеке салаларына енгізу, оларды жалпылама енгізгеннен гөрі тиімдірек болады. Осындай салалардың бірі ретінде жедел коронарлық синдромы бар медицина бөлімін айтуға болады. Осы дауамен ауыратын адамдарды олардың ауру деңгейлеріне байланысты регистрге алу, синдромды кем дегенді ушықтырмауға мүмкіндік береді. Сондай - ақ, ЖКСР АЖ синдромды диагностикалауға, емдеу - алдын алу шараларын жүргізу жағдай жасайды.

Дипломдық жұмысты жазу барысында медициналық ақпараттық технологиялар мен жүйлері және оның түрлері, web - технологияларды қолдану амалдары толықтай сараланып шықты.

Дипломдық жұмыстың негізгі бөлімдерінде медициналық ақпараттық жүйелер мен технологиялар, Python бағдарламалау тілі, оны жүзеге асыру жайлы теоретикалық мәліметтер берілді. Сонымен қатар, жасалынған веб - қосымшаның сипаттамалары, диаграммалары және жұмыс істеу функционалы туралы баяндалды

Зерттелген жұмыстың негізгі мақсатына сай ЖКСР АЖ web - қосымшасы құрылды. Жұмысты орындау кезінде келесідей мәселелер қарастырылып, орындалды:

* зерттеу тақырыбына толық сипаттама берілді;
* медицинада қолданылып жатқан АЖ шолу жасалды;
* жедел коронарлық синдром регистрі жүйесінің жұмысын автоматтандыруға қойылатын талаптар анықталды;
* АЖ - нің жұмыс істеу функцияналы анықталды;
* web - технологияны қолдана отырып ЖКСР АЖ - сінің web - қосымшасы құрылды.

Жұмыстың нәтижесі ретінде ЖКСР жұмысындағы процесттерді басқарудағы АЖ web - қосымшасы ұсынылды. Аталған АЖ - ні ЖКСР - мен айналысатын медициналық орталықтар сәтті қолдана алады.

Қорытындылап айтқанда, жұмыстың басталу уақытында қойылған мақсаттар мен міндеттерге қол жеткізілді. Жұмыс нәтижесіндегі web - қосымшаны ЖКСР процестерін атқаруды одан да жеңілдететін әртүрлі жаңа функционалдық шешімдерді енгізу арқылы дамытуға болады.

# **Пайдаланылған әдебиеттер**

1. Кобринский Б. А., Зарубина Т. В. «Медицинская информатика», 2012 г., с. 56 - 72.

#### Автоматизация процессов, цифровые и информационные технологии в управлении и клинической практике лечебного учреждения: научные труды / Под ред. О. Э. Карпова. - М.: Деловой экспресс, 2016. - 388 с.

#### Воронцов И. М., Шаповалов В.В., Шерстюк Ю. М. ЗДОРОВЬЕ. Создание и применение автоматизированных систем для мониторинга и скринирующей диагностики нарушений здоровья. - Санкт - Петербург: 2016.

#### [Гельман В.Я. Медицинская информатика: практикум (2 - е изд.). - С. - Пб.: Питер, 2012. - 480 с.](https://armit.ru/books/gelman.php)

#### Гусев А. В., Романов Ф. А., Дуда­нов И. П., Воронин А. В. Информацион­ные системы в здравоохранении. Пет­розаводск: Изд - во ПетрГУ, 2012. - 120 с.

#### Дуданов И. П., Романов Ф. А., Гу­сев А.В. Информационная система в организации работы учреждений здра­воохранения: Практическое руково­дство / И. П. Дуданов, Ф. А. Романов, А. В. Гусев. ПетрГУ - Петрозаводск, Издательство ПетрГУ, 2015 - 238 с.

#### Зекий О.Е. Автоматизация здравоохранения. - М.: Новости, 2021. - 400 с.

#### Кудрина В. Г., Андреева Т. В., Дзеранова Н. Г. Эффективность обучения медицинских работников информационным технологиям. Москва: ИД "Менеджер здравоохранения", 2013. - 248 с.

#### Назаренко Г. И., Гулиев Я. И., Ерма­ков Д. Е. Медицинские информацион­ные системы: теория и практика. Под редакцией Г. И. Назаренко Г. С. Осипова. М.: ФИЗМАТЛИТ, 2015. - 320 с.

# Медицинские информационные системы - http://riac - almaty.kz/index.php/ru/informatsionnye - sistemy - mz - rk

1. Королюк И.П. «Основы медицинской информатики», 2012г., с. 122 - 131.
2. Медицинские информационные системы: анализ рынка // Гусев А. В., Дуданов И.П., Романов Ф. А. / PCWeek №47/2015. С38 - 40
3. Россум Г. Язык программирования python, 2020.
4. Б.А. Язык программирования python Ч1 - Ч3, 2014.
5. Лутц, М. Программирование на Python, в 2 т. Т. 1, 4 - е издание / Марк Лутц. - Санкт - Петербург: Символ - Плюс, 2021. - 992 с.
6. Форсье, Д. Django. Разработка Web - приложений на Python. 2020. - 458 c.
7. Чан, У. Python. Создание приложений / Уэсли Чан. - Санкт - Петербург: Вильямс, 2015. - 816 c.
8. Головатый, А. Django. Подробное руководство / Адриан Головатый, Джейкоб Каплан - Мосс. - Санкт - Петербург: Символ - Плюс, 2020. - 560 c.
9. Хахаева, И.А. Практикум по алгоритмизации и программированию на Python: учебно - методический комплекс / И. А. Хахаева. - Москва: Альт Линукс, 2021. - 226 с.

# **А қосымшасы**

Веб - қосымшаның басты файлдарының бірі - manage.py файлындағы код үзіндісі:

#!/usr/bin/env python

"""Django's command - line utility for administrative tasks."""

import os

import sys

def main():

os.environ.setdefault('DJANGO\_SETTINGS\_MODULE', 'hospitalmanagement.settings')

try:

from django.core.management import execute\_from\_command\_line

except ImportError as exc:

raise ImportError(

- - -

) from exc

execute\_from\_command\_line(sys.argv)

if \_\_name\_\_ == '\_\_main\_\_':

main()

# **Ә қосымшасы**

Веб - қосымшаның басты файлдарының бірі - settings.py файлындағы код үзіндісі:

"""

Django settings for hospitalmanagement project.

Generated by 'django - admin startproject' using Django 3.0.5.

For more information on this file, see

https://docs.djangoproject.com/en/3.0/topics/settings/

For the full list of settings and their values, see

https://docs.djangoproject.com/en/3.0/ref/settings/

"""

import os

# Build paths inside the project like this: os.path.join(BASE\_DIR, ...)

BASE\_DIR = os.path.dirname(os.path.dirname(os.path.abspath(\_\_file\_\_)))

TEMPLATE\_DIR = os.path.join(BASE\_DIR,'templates')

STATIC\_DIR=os.path.join(BASE\_DIR,'static')

# Quick - start development settings - unsuitable for production

# See https://docs.djangoproject.com/en/3.0/howto/deployment/checklist/

# SECURITY WARNING: keep the secret key used in production secret!

SECRET\_KEY = 'hpbv()ep00boce&o0w7z1h)st148(\*m@6@ - rk$nn)(n9ojj4c0'

# SECURITY WARNING: don't run with debug turned on in production!

DEBUG = True

ALLOWED\_HOSTS = []

# Application definition

INSTALLED\_APPS = [

'django.contrib.admin',

'django.contrib.auth',

'django.contrib.contenttypes',

'django.contrib.sessions',

'django.contrib.messages',

'django.contrib.staticfiles',

'hospital',

'widget\_tweaks',

]

MIDDLEWARE = [

'django.middleware.security.SecurityMiddleware',

'django.contrib.sessions.middleware.SessionMiddleware',

'django.middleware.common.CommonMiddleware',

'django.middleware.csrf.CsrfViewMiddleware',

'django.contrib.auth.middleware.AuthenticationMiddleware',

'django.contrib.messages.middleware.MessageMiddleware',

'django.middleware.clickjacking.XFrameOptionsMiddleware',

]

ROOT\_URLCONF = 'hospitalmanagement.urls'

TEMPLATES = [

{

'BACKEND': 'django.template.backends.django.DjangoTemplates',

'DIRS': [TEMPLATE\_DIR,],

'APP\_DIRS': True,

'OPTIONS': {

'context\_processors': [

'django.template.context\_processors.debug',

'django.template.context\_processors.request',

'django.contrib.auth.context\_processors.auth',

'django.contrib.messages.context\_processors.messages',

],

},

},

]

WSGI\_APPLICATION = 'hospitalmanagement.wsgi.application'

# Database

# https://docs.djangoproject.com/en/3.0/ref/settings/#databases

DATABASES = {

'default': {

'ENGINE': 'django.db.backends.sqlite3',

'NAME': os.path.join(BASE\_DIR, 'db.sqlite3'),

}

}

# Password validation

# https://docs.djangoproject.com/en/3.0/ref/settings/#auth - password - validators

AUTH\_PASSWORD\_VALIDATORS = [

{

'NAME': 'django.contrib.auth.password\_validation.UserAttributeSimilarityValidator',

},

{

'NAME': 'django.contrib.auth.password\_validation.MinimumLengthValidator',

},

{

'NAME': 'django.contrib.auth.password\_validation.CommonPasswordValidator',

},

{

'NAME': 'django.contrib.auth.password\_validation.NumericPasswordValidator',

},

]

# Internationalization

# https://docs.djangoproject.com/en/3.0/topics/i18n/

LANGUAGE\_CODE = 'en - us'

TIME\_ZONE = 'UTC'

USE\_I18N = True

USE\_L10N = True

USE\_TZ = True

# Static files (CSS, JavaScript, Images)

# https://docs.djangoproject.com/en/3.0/howto/static - files/

STATIC\_URL = '/static/'

STATICFILES\_DIRS=[STATIC\_DIR,]

MEDIA\_ROOT=os.path.join(BASE\_DIR,'static')

LOGIN\_REDIRECT\_URL='/afterlogin'

#for contact us give your gmail id and password

EMAIL\_BACKEND ='django.core.mail.backends.smtp.EmailBackend'

EMAIL\_HOST = 'smtp.gmail.com'

EMAIL\_USE\_TLS = True

EMAIL\_PORT = 587

EMAIL\_HOST\_USER = 'from@gmail.com' # this email will be used to send emails

EMAIL\_HOST\_PASSWORD = 'xyz' # host email password required

# now sign in with your host gmail account in your browser

# open following link and turn it ON

# https://myaccount.google.com/lesssecureapps

# otherwise you will get SMTPAuthenticationError at /contactus

# this process is required because google blocks apps authentication by default

EMAIL\_RECEIVING\_USER = ['to@gmail.com'] # email on which you will receive messages sent from website

# **Б қосымшасы**

Веб - қосымшаның басты файлдарының бірі - admin.py файлындағы код үзіндісі:

from django.contrib import admin

from .models import Doctor, Patient, Appointment, PatientDischargeDetails, ListOfSymptoms2

# Register your models here.

class DoctorAdmin(admin.ModelAdmin):

pass

admin.site.register(Doctor, DoctorAdmin)

class PatientAdmin(admin.ModelAdmin):

pass

admin.site.register(Patient, PatientAdmin)

class AppointmentAdmin(admin.ModelAdmin):

pass

admin.site.register(Appointment, AppointmentAdmin)

class PatientDischargeDetailsAdmin(admin.ModelAdmin):

pass

admin.site.register(PatientDischargeDetails, PatientDischargeDetailsAdmin)

class ListOfSymptoms2Admin(admin.ModelAdmin):

pass

admin.site.register(ListOfSymptoms2, ListOfSymptoms2Admin)