

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

СӘТБАЕВ УНИВЕРСИТЕТІ

Кибернетика және ақпараттық технологиялар институты

Программалық инженерия кафедрасы

**ҚОРҒАУҒА ЖІБЕРІЛДІ**

Кафедра меңгерушісі

тех. ғыл. кандидаты,

ассоц. профессор

Р. Юнусов

“ \_\_\_\_\_ ” \_\_\_\_\_ 2020 ж.

## **ДИПЛОМДЫҚ ЖҰМЫС**

**Тақырыбы** «Оқытушылар құрамының конкурсы» веб қосымшасы

**Мамандығы** 5B070400 – Есептеу техникасы және бағдарламалық қамтамасыз ету

Орындаған

Зәкір К.Т

Ғылыми жетекші  
техн. ғыл. магистрі,  
лектор

І.Н. Ерсари

“ \_\_\_\_\_ ” \_\_\_\_\_ 2020 ж.

Алматы 2020

## Протокол анализа Отчета подобия Научным руководителем

Заявляю, что я ознакомился(-ась) с Полным отчетом подобия, который был сгенерирован Системой выявления и предотвращения плагиата в отношении работы:

**Автор:** Зәкір Кенесары Тәжібайұлы, Ерсари Іңкәр Нұрхатқызы

**Название:** Веб қосымша - оқытушылар құрамының конкурсы

**Координатор:** Жибек Алибиева

**Коэффициент подобия 1:0**

**Коэффициент подобия 2:0**

**Замена букв:4**

**Интервалы:0**

**Микропробелы:0**

**Белые знаки: 0**

**После анализа Отчета подобия констатирую следующее:**

- обнаруженные в работе заимствования являются добросовестными и не обладают признаками плагиата. В связи с чем, признаю работу самостоятельной и допускаю ее к защите;
- обнаруженные в работе заимствования не обладают признаками плагиата, но их чрезмерное количество вызывает сомнения в отношении ценности работы по существу и отсутствием самостоятельности ее автора. В связи с чем, работа должна быть вновь отредактирована с целью ограничения заимствований;
- обнаруженные в работе заимствования являются недобросовестными и обладают признаками плагиата, или в ней содержатся преднамеренные искажения текста, указывающие на попытки сокрытия недобросовестных заимствований. В связи с чем, не допускаю работу к защите.

Обоснование:

.....  
.....

*Дата*

*Подпись Научного руководителя*

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

СӘТБАЕВ УНИВЕРСИТЕТІ

Кибернетика және ақпараттық технологиялар институты

Программалық инженерия кафедрасы

5B070400 – Есептеу техникасы және бағдарламалық қамтамасыз ету

**БЕКІТЕМІН**

Кафедра меңгерушісі  
тех. ғыл. кандидаты,  
ассоц. профессор

\_\_\_\_\_ Р.Юнусов  
“ \_\_\_\_\_ ” \_\_\_\_\_ 2020 ж.

**Дипломдық жұмыс орындауға  
ТАПСЫРМА**

Білім алушы Зәкір Кенесары Тәжібайұлы

Тақырыбы «Оқытушылар құрамының конкурсы» веб қосымшасы

Университет Ректорының 20\_\_ жылғы "\_\_\_" \_\_\_\_\_ №\_\_-б бұйрығымен бекітілген  
Аяқталған жұмысты тапсыру мерзімі 2020 жылғы "\_\_\_" \_\_\_\_\_

Дипломдық жұмыстың бастапқы берілістері: Ұсынылатын дипломдық жобада  
«Оқытушылар құрамының конкурсы» үміткер оқытушыларға конкурс  
жариялайтын веб-қосымша жасау.

Дипломдық жұмыста қарастырылатын мәселелер тізімі

а) Негізгі бөлім;

б) Технологиялар бөлімі;

в) Жобаны талдау бөлімі;

г) Жоба құрылым бөлімі;

Сызба материалдар тізімі (міндетті сызбалар дәл көрсетілуі тиіс)  
Жобаның презентациялық 15 слайды ұсынылған

Ұсынылатын негізгі әдебиет 15 әдебиеттер тізімімен.

Дипломдық жұмысты дайындау  
КЕСТЕСІ

Бөлімдер атауы, қарастырылатын мәселелер тізімі	Ғылыми жетекші мен кеңесшілерге көрсету мерзімдері	Орындалған жөніндегі белгі
1. Дипломдық жоба тақырыбына байланысты әдебиеттерді іздеу және талдау, дипломдық жұмыстың негізін рәсімдеу	24.01.2020	
2. UML модельдеу тілі арқылы дипломдық жұмысты жобалау	05.02.2020	
3. Дипломдық жобаның веб қосымшасын дайындау және іске асыру	21.02.2020	
4. Жобаны диплом алды есеп алу комиссия мүшелеріне таныстыру, ескертулерін түзеу және қорытындылау.	27.04.2020	
5. Қосымша материалдар мен презентацияны дайындау	01.05.2020	

Дипломдық жұмыс бөлімдерінің кеңесшілері мен норма бақылаушының аяқталған жұмысқа қойған  
қолтаңбалары

Бөлімдер атауы	Кеңесшілер, аты, әкесінің аты, тегі (ғылыми дәрежесі, атағы)	Қол қойылған күні	Қолы
Бағдарламалық бөлім	Куникеев А. Техн. ғыл. магистрі, сениор-лектор		
Нормалық бақылаушы	Марғұлан Қ. Техн. ғыл. магистрі, лектор		

Ғылыми жетекші \_\_\_\_\_ І.Н. Ерсари

Тапсырманы орындауға алған білім алушы \_\_\_\_\_ К.Т. Зәкір

Күні «\_\_» \_\_\_\_\_ 2020 ж.

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

СӘТБАЕВ УНИВЕРСИТЕТІ

Кибернетика және ақпараттық технологиялар институты

Мамандығы 5B070400 – Есептеу техникасы және бағдарламалық қамтамасыз ету

Зәкір Кенесары Тәжібайұлы

Дипломдық жұмыс

**Тақырыбы:** Оқытушылар құрамының конкурсына арналған веб-қосымша

### **ҒЫЛЫМИ ЖЕТЕКШІНІҢ ШІКІРІ**

Диплом жобасын жасаушы Зәкір Кенесары алдына оқытушылар құрамының конкурсына арналған веб-қосымша жасау тапсырмасы қойылған.

Дипломдық жобада үміткер оқытушылардың өтініштерін қабылдау, модерациядан өткізу, дауыс беру арқылы конкурстан өткізу және конкурс нәтижесін шығару процесстерін үйлестіретін кешенді веб қосымша құрылған.

Жұмыс барысындағы басты мәселелер: өтініштерді электронды түрде жіберу және қабылдау, оқытушылардың өтініштерін мұқият модерациядан өткізу, кадрлар бөлімі мен үміткер оқытушылардың байланысын қамтамасыз ету, онлайн түрде үміткерге дауыс беру және конкурс нәтижелерін дереу шығаруды қамтамасыз ету болып табылады. Бұл өз кезегінде университет пен болашақ оқытушылар арасындағы уақыт ресурсын үнемдеуде таптырмас веб қосымша бола алады.

Менің пікірімше, диплом жазушы алдына қойылған тапсырманы толығымен орындады және ақпараттық жүйелердің заманауи технологияларын меңгергендігін көрсетті.

Жоба жетекшісі ретінде бұл дипломдық жобаны өз деңгейіне сәйкес деп есептей отырып Зәкір Кенесарыға 5B070400 – «Есептеу техникасы және бағдарламалық қамтамасыз ету» мамандығы бойынша «Техника және технологиялар бакалавры» академиялық дәрежесін тағайындауға болады деп есептеймін.

**Ғылыми жетекші:** «Программалық инженерия» кафедрасының лекторы  
\_\_\_\_\_ І.Н. Ерсари

«\_\_» \_\_\_\_\_ 2020 жыл

## АҢДАТПА

«Оқытушылар құрамының конкурсы» веб қосымшасын құру жобасында жоғарғы оқу орнына үміткер оқытушылардың өтініштері электронды түрде қабылданып, мұқият модерациядан өту арқылы комиссия мүшелерімен сұқбаттасуға шақырылады.

Дипломдық жұмыс кіріспеден, негізгі үш бөлімнен және қорытындыдан құрылған:

Кіріспеде жалпы тақырыптың өзектілігі ашылып, жобаның басты мақсаты мен тапсырмасы айқындалды.

Бірінші бөлімде оқытушылардың өтініштерін қабылдау процесстері, сонымен қатар конкурс өту сатылары мен модерация кейінгі іс-әрекеттер жазылды.

Екінші бөлімде «Оқытушылар құрамының конкурсы» веб қосымшасына UML модельдеу тілі арқылы диаграммалар келтіріліп, жобада қолданылған бағдарламалар мен бағдарламау тілдері туралы кеңінен жазылған. Дипломдық нарықтағы соңға веб технологиялармен жасалынды.

Үшінші бөлімде жоба құрылымы сипатталған және веб қосымшаның барлық интерфейстерінің скриншоттары бейнеленген.

«Оқытушылар құрамының конкурсы» түсініктемесі жалпы 38 бет, оның ішінде 20 сурет, 1 қосымшадан тұрады. Жұмысты жазуға 32 әдебиеттер мен мақалалар қолданылды.

## АННОТАЦИЯ

В проекте «Конкурс преподавательского состава» заявки преподавателей-кандидатов в высшие учебные заведения принимаются в электронном виде и приглашаются для обсуждения с членами комиссии путем тщательной модерации.

Дипломный проект состоит из введения, трех основных разделов и заключения:

В введении рассмотрена актуальность общей темы, определены основные цели и задачи проекта.

В первой части изложены процессы приема заявлений преподавателей, а также этапы проведения конкурса и дальнейшая действий модерации.

Во второй части приведены диаграммы на языке UML к веб приложению "Конкурс преподавательского состава", где предусмотрены основы работы с используемыми в приложении средами и языками программирования. Проект был разработан с помощью нескольких технологий.

В третьей части описана структура проекта и изображены скриншоты всех интерфейсов веб-приложения.

Объяснение "Конкурс преподавательского состава" состоит 38 из страниц, из них 20 рисунков, 1 приложение. К написанию работы использовались 32 литературных статей.

## ANNOTATION

In the project “Competition of teaching staff”, applications of teachers-candidates for higher educational institutions are accepted in electronic form and are invited for discussion with members of the commission through thorough moderation.

The graduation project consists of an introduction, three main sections and a conclusion:

In the introduction, the relevance of the general topic is considered, the main goals and objectives of the project are determined.

The first part outlines the processes for accepting teachers' applications, as well as the stages of the competition and further moderation actions.

The second part contains diagrams in the UML language for the web application "Competition of the teaching staff", which provides the basics of working with the programming environments and languages used in the application. The project was developed using several technologies.

The third part describes the structure of the project and shows screenshots of all interfaces of the web application.

The explanation "Competition of the teaching staff" consists of 38 pages, 20 of them are drawings, 1 appendix. By writing, the work used 32 literary articles.



## МАЗМҰНЫ

	Кіріспе	10
1	Негізгі бөлім	11
1.1	Жобаның өзектілігі	11
1.2	Веб қосымша жасаудың негіздері	11
1.3	Өтінішті қабылдау процессіне шолу	12
1.4	Жүйені модельдеу	13
1.5	Ұқсас бағдарламаға шолу	14
2	Технологиялар бөлімі	16
2.1	MEAN stack веб қосымшасы	16
2.2	MongoDB кроссплатформалық деректер базасы	17
2.3	Express.js фреймворкі	18
2.4	Angular 2.0 фреймворкі	19
2.4.1	Angular 2.0 қосымшасының архитектурасы	20
2.5	Node.js серверлік платформа	21
2.6	Pusher.js нақты уақыттағы деректер және Chart.js	22
3	Жоба құрылымы	23
3.1	«Satbayev Hunters» жобасы туралы	23
3.2	Веб қосымша интерфейстері	23
	Қорытынды	31
	Пайдаланылған әдебиеттер тізімі	32
	А Қосымшасы. Техникалық тапсырма	33
	Б Қосымшасы. Бағдарлама мәтіні	37
	Спецификация беті	43

## КІРІСПЕ

Қазіргі технологиялардың қарқынды даму сатысында әрбір адамның дүниежүзілік желіге шығатын компьютері бар. Бұл ғаламторда ұсынылған, өзекті ақпаратты пайдаланушылар үшін оңай қол жетімді етеді және кез келген мақсаты орындау үшін қолданылады. Бұл өз кезегінде интернеттегі веб қосымшалар арқылы жүзеге асады. Веб қосымша (веб сайт) – белгілі бір ақпараттарды тарату үшін немесе қызмет көрсету сервистерін ұсынатын ғаламторда жарияланған бағдарламалық қосымша. Веб қосымшалардың негізінде қандайда бір кәсіпорынның қызмет көрсету сапасын арттырып, ақпарат беру немесе ғаламтор арқылы сауда жасау мақсатында құрылатын қосымшаларды айтуға болады. Angular десктопты немесе мобильді веб қосымшаларды жасауға арналған Javascript программалау тілінің фрейворкі.

Оның бір-беттік қосымша (single page application (SPA)) жасауға немесе қосымша компоненттерінің бір-бірімен динамикалық түрде байланысуына мүмкіндік беретін артықшылықтары бар болғандықтан, Javascript бағдарламаушыларының арасында көп пайдаланылатын фреймворк болып табылады. Сол себепті “Satbayev Hunters” вакансияға өтініш қалдыру веб қосымшасы Angular фреймворкінде құрылды.

Веб қосымшаның негізгі аудиториясы – білікті мамандар, яғни “Satbayev Hunters” веб қосымшасы арқылы белгілі бір мамандықты тандап, университеттің кадр бөліміне жеке өтінішін және резюме құжаттарын үміткер ретінде жіберуге арналған.

Дипломдық жұмыстың мақсаты: Сәтбаев Университетінде жұмыс істегісі келетін үміткер білікті мамандардың өтінішін және құжаттарын электронды түрде қабылдап, комиссия мүшелеріне бір жүйе арқылы үміткерге дауыс беру болып табылады.

Дипломдық жоба өзектілігі: “Satbayev Hunters” веб қосымшасы өтініш беруші үміткер мен кадр бөлімінің арасындағы өтініш беру формасын электронды форматқа көшіру болып табылады. Бұның бірқатар артықшылықтарына көз жеткізуге болады, біріншіден үміткер мен кадр бөлімі арасында қағаз ресурсін үнемдеуге көмектеседі, екіншіден үміткер университетке келмей қашықтықтан өтініш бере алады, яғни үміткер өз уақытын құртпайды. Өтініш беру жүйесі білімді мамандардың университетімізде көп болуына және білім сапасын жақсартуға ықпалын тигізетін таптырмас веб қосымша болады.

## **1 Негізгі бөлім**

### **1.1 Жобаның өзектілігі**

Елімізде мобильді қосымшаларға ұқсайтын веб қосымшалардың үлесі артып келеді, оны күнделікті пайдаланып жүрген отандық веб қосымшалардан байқауымызға болады. Тікелей ғаламтор желісіне қосылуды талап ететін бұл веб қосымшаның түрі барлық құрылғылардың браузерлерінде қолжетімді.

Satbayev Hunters веб қосымшасы негізінен университетте жұмыс істегісі келетін мамандарға арналған. Үміткер мамандар кез-келген құрылғылардың браузерлерінен веб қосымшаға кіру арқылы өтініш беру құжаттар тізімін, жалпы өтініш ережесін және сұқбат өтетін орын мен уақытымен таныса алады. Сонымен қатар үміткер кез-келген уақытта өтінішін онлайн түрде кадр бөліміне жібере алады, ал сұқбат өтіп болғаннан кейін комиссия мүшелері үміткерге дауыс береді. Бұл қосымша тек Сәтбаев Университетіне арналған жоба болып табылады.

Бұл жоба үш түрлі веб қолданушыға арналған, яғни атап өтетін болсақ, үміткер өтініш беретін парақша, комиссия мүшелеріне арналған парақша, администраторға арналған жеке парақша. Веб сервер Node.js программалау тілімен жазылған, барлық деректер қоры MongoDB негізінде жасалынған бұлттық mLab сервисінде сақталынған. Веб қосымшамен серверді бір-бірімен байланыстыру үшін Express.js және REST-API технологиясы қолданылды. Ал қолданушыға көрінетін бетті Angular 2.0 фреймворкімен жасалынды, белгілі кандидатқа дауыс беру компонентінде нақты уақыттағы (real-time) дауыстарды санап, оны басқа да комиссия мүшелеріне бірдей уақытта көрсету үшін Pusher.js фреймворкі қолданылды. Диаграммаларды дер кезінде шығару үшін Chart.js фреймворк тектес кітапхана пайдаланылды.

### **1.2 Веб қосымша жасаудың негіздері**

Веб қосымша – бұл гиперсілтемелермен байланысты веб беттер жиынтығы. Веб беттер немесе гипермәтін құжаттары тег деп аталатын арнайы командалар (tags) бар мәтін болып табылады. Бұл тегтер бет элементтерін пішімдеуді қамтамасыз етеді және графикалық нысандарды, суреттерді, гиперсілтемелерді және т.б. орналастыруға мүмкіндік береді. Веб қосымшаны құру үшін негізгі қажеттер:

- сайтты құру мақсатын анықтау;
- техникалық тапсырмаларды әзірлеу;
- сайттың доменін белгілі бір аймақта тіркеу (com, ru, kz, net)
- сайтты әзірлеу;
- хостингке сайт орналастыру;

– іздеу жүйелері мен тақырыптық каталогтарда тіркеу.

Кез-келген веб қосымшаның негізгі мақсаты клиентті веб сервермен байланыстырып, қалаған сұраныстарды жүргізуге рұқсат беру болып табылады. Веб қосымшаның бірқатар артықшылықтарын атап өтсек:

- ғаламторға кірген кез-келген қолданушыға қолжетімді;
- бағдарламалық қосымша орнатудың қажеті жоқ;
- іздеу жүйелерінде (Google, Yandex) SEO оптимизация жасауға ыңғайлы, яғни веб қосымшаны іздеу жүйелеріндегі тізбекте бастапқы орындарға шығару болып табылады;
- уақытылы жаңартулар оңай жасалынады;
- жадыда орын алмайды.

### **1.3   Өтінішті қабылдау процессіне шолу**

Өтінішті жіберу кезінде қолданушы формада берілген орындарды толтырып, жеке құжаттарын тіркеу арқылы өтініш қалдырады. Өтініш администратордың үміткерлер кестесіне түседі. Үміткерлердің өтініштері мен құжаттары бір рет мұқият тексеруден (модерация) администратор өткізеді, екінші кезеңде комиссия мүшелері сұхбаттасу кезінде тексереді. Егер өтініште қажетті құжаттар жеткіліксіз болса немесе жалпы мәліметтері дұрыс болмаса, үміткердің почтасына модерациядан өтпегені туралы хабарлама жіберіледі, қателіктерін дұрыстап қайтадан жіберуі сұранылады, ал егер барлығы дұрыс болып, модерациядан өткен жағдайда үміткердің почтасына сұхбаттасуға шақыру туралы хабарлама жіберлетін болады. Стандартқа сәйкес келмеген өтініштер администратор арқылы автоматты түрде кейін өшіріледі. Сұхбаттасу кезінде комиссия мүшелері үміткерге сұрақтар қойып болған соң, ортақ бір шешімге келу үшін дауыс береді. Сол жоғарғы дауыстың нәтижесі бойынша үміткер универге жұмысқа тұрады, ал керісінше жағдайда үміткердің сұхбаттан өтпегені хабарланады. Сұхбаттасу кезіндегі үміткер нәтижелері мен құжаттары веб қосымша деректер базасында сақталады.

Satbayev Hunters қосымшасы өтінішті қабылдау және дауыс беру кезінде бірқатар мәселелерді шешті:

- деректер мен құжаттар орталықтандырылған деректер қорында сақталған, әрбір комиссия мүшесіне кез-келген ақпарат қолжетімді;
- үміткерге онлайн дауыс беру мүмкіншілігі;
- үміткерге модерация нәтижесін почтасына жіберу, хабарландыру.

## 1.4 Жүйені модельдеу

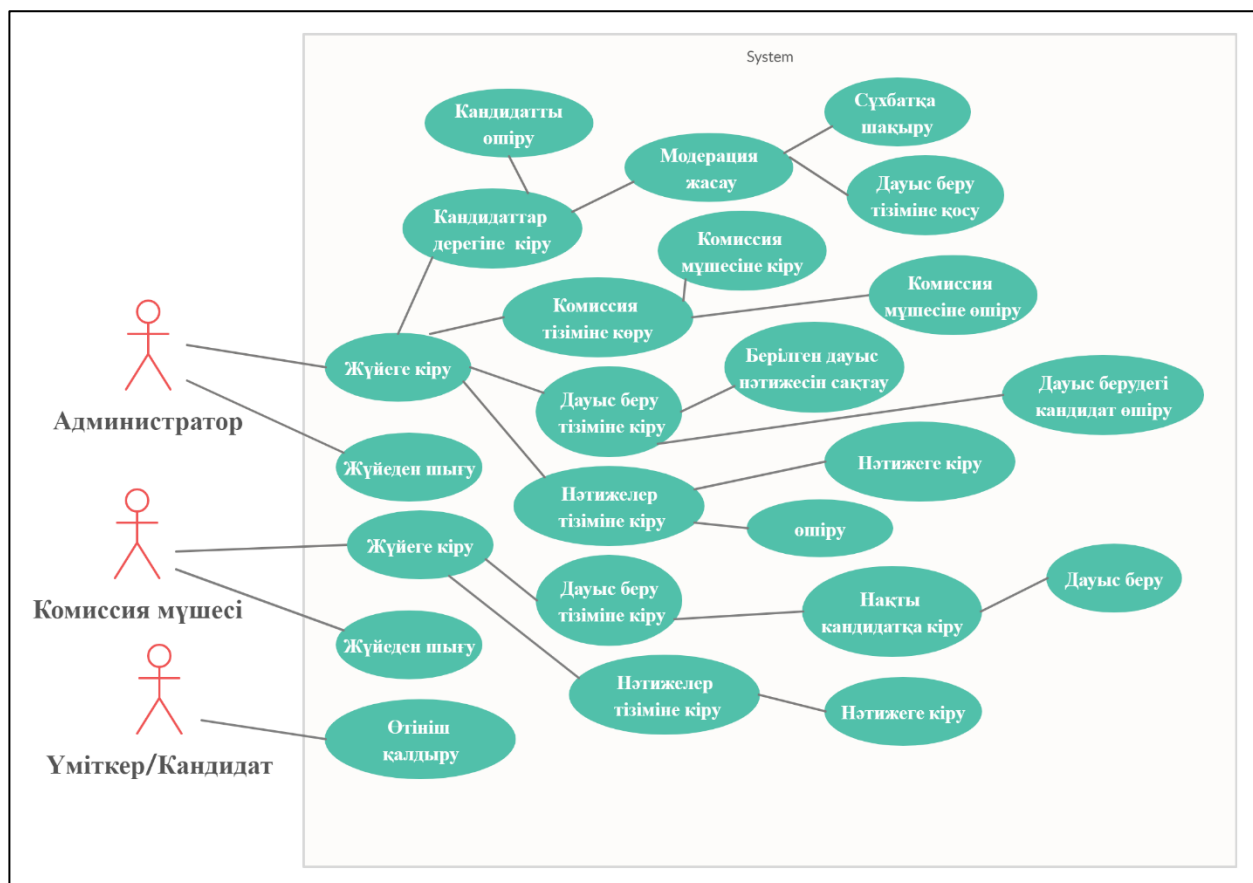
Веб қосымшаның техникалық тапсырмаларын әзірлеу барысында жобаны модельдеу жұмыстары қатар жүреді. Ауқымды жүйе құруда нақты мақсатқа жету үшін бірыңғай модельдеу тілі қолданылады.

UML (Unified Modeling Language) – бағдарламалық жүйенің артефактілерін визуализациялау, құжаттау және модельдеу тілі. Модельдеу жүйені түсіну үшін қажет. Сонымен қатар, жүйені құру кезінде бір модель түрі жеткіліксіз болады. Сондықтан, кез-келген тривиалды емес жүйені түсіну үшін бір-бірімен өзара байланыстағы көптеген модельдерді жасау қажет. UML арқылы қосымшаның функция және архитектурасын диаграммалық, тексттік, сызба түрінде құруға болады. Бағдарламалық жасақтамаға қолданған кезде, бұл оның жүйелік архитектурасының әртүрлі көріністерін бейнелеу үшін қолданылатын тіл екенін білдіреді.

Бірыңғай модель диаграммаларында нысандар және олардың арасындағы қатынастар түрінде ұсынылады. UML моделдеу тілінде диаграммалардың көптеген түрлері бар, бірақ осы дипломдық жұмыста, түсінікті әрі оңай болғандықтан прецеденттер диаграммасын қолдандым.

Прецеденттер диаграммасы – UML диаграммаларының бірі, жүйенің динамикалық аспектілерін модельдеу үшін қолданылады. Прецеденттер диаграммалары элементтің іс-әрекетін визуализациялау, нақтылау және құжаттау үшін маңызды. Олар жүйелерді, ішкі жүйелерді немесе класстарды түсінуді жеңілдетеді, бұл элементтерді контексте қалай қолдануға болатындығы туралы сыртқы көріністі ұсынады. Прецеденттер диаграммасы экторлар (actor) мен олардың іс-әрекеттері арқылы сипатталады. Эктор (actor) – жүйедегі белгілі бір қолданушылар. Модельдеу жүйесіне қойылатын талаптар:

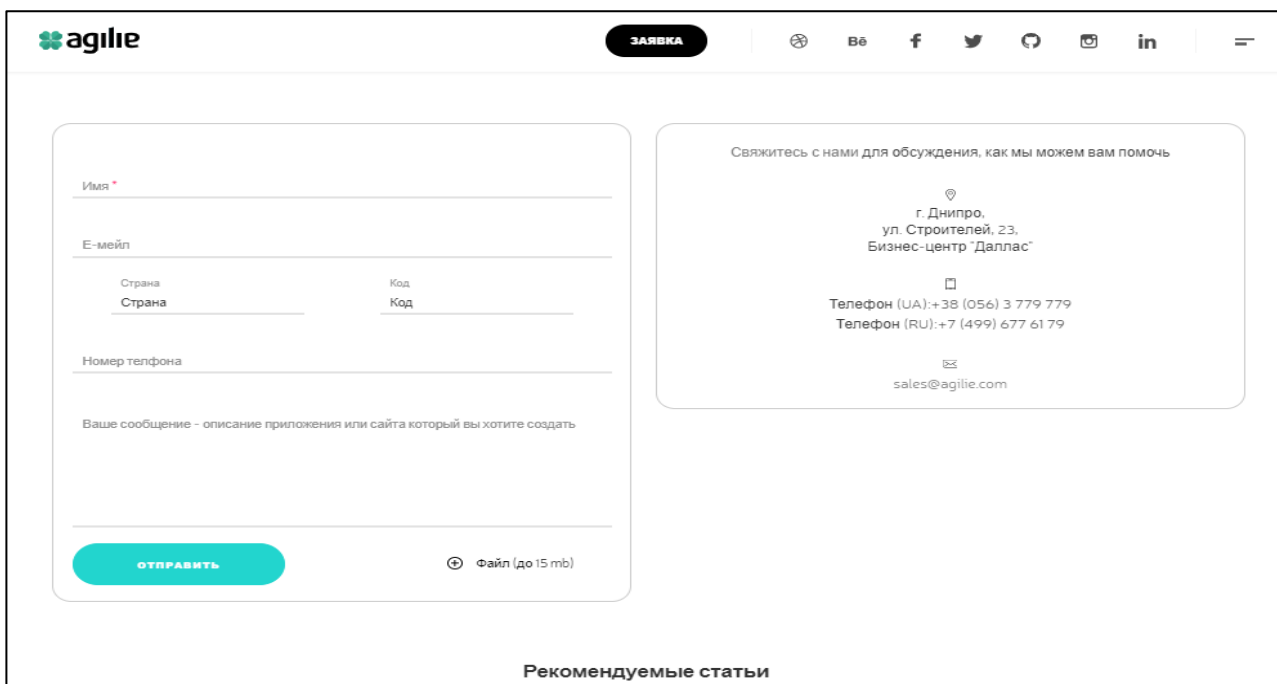
- жүйедегі негізгі экторларды анықтап алу;
- әр эктор үшін жүйеден күтілетін немесе талап етілетін іс-әрекеттерін қарастыру;
- белгілі экторлармен байланысатын іс-әрекеттерді біріктіріп диаграммаға сызу.



1.1-сурет – Прецеденттер диаграммасы (Use-case)

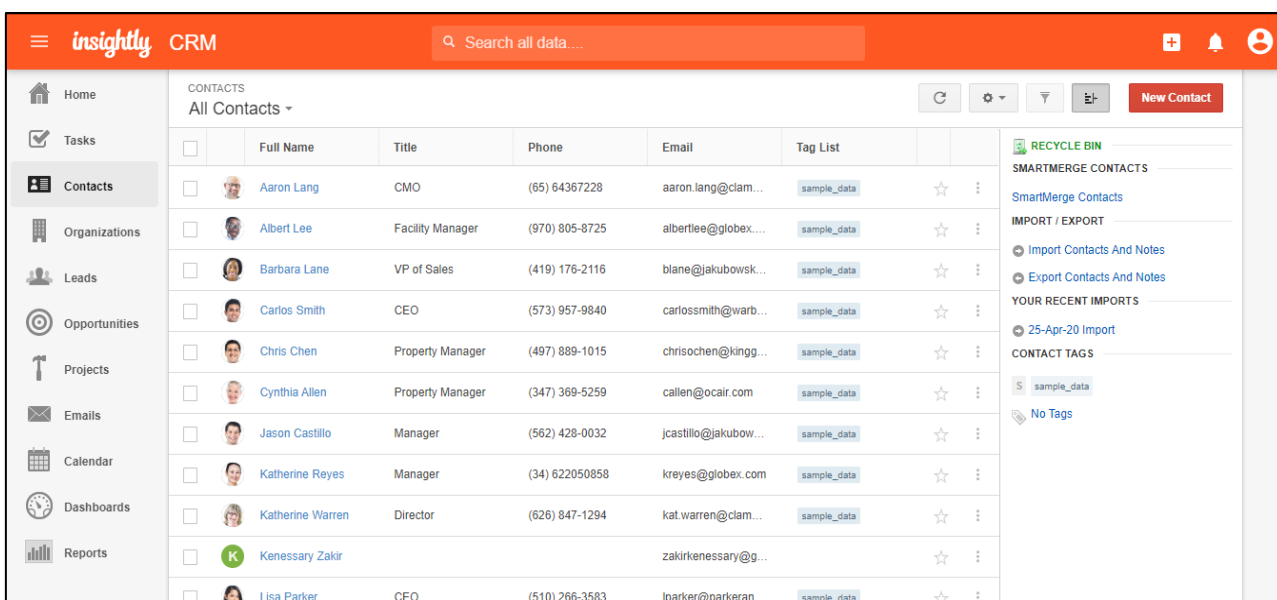
### 1.5 Ұқсас бағдарламаға шолу

IT- нарығында көптеген жекелеген кәсіпорындар өздерінің дайын өнімін клиенттерге веб қосымшасы арқылы толық ақпарат беріп, өнімді онлайн түрде сатып, веб қосымшасын табыс көзінің біріне айналдырады. Веб қосымша арқылы жекелеген кәсіпорындар өздерінің бәсеке қабілетті екенін дәлелдей алады. Соның бір мысалы ретінде Украиналық веб және мобильді қосымшаларды жасайтын компания (agilie.com). Бұл компания заманауи технологияларды қолдану арқылы сапалы Full-stack веб және мобилді қосымшасын дайындауға бейімделген және пайдалы программалауға байланысты блогтар жүргізіп тұрады. Аталмыш компанияға өзіңіздің бизнесіңізге веб қосымша жасату үшін өтініш қалдыру формасы “Satbayev Hunters”-тың өтініш беру формасына функционалдығы жағынан бірдей. Сипатталған көріністер 1.1-суретте бейнеленген.



## 1.2-сурет – “Agilie” компаниясының өтініш беру формасы

Бұл жобада клиенттік бет пен қатар комиссия және администратордың жеке беттері бар. Ол веб беттер үміткерлерге көрінбейді. Администратор веб беттің CRM (Customer Relationship Management) веб қосымшасы ретінде жасалынды, яғни қазіргі кезде көп компаниялар өздерінің қызметкерлерін басқару үшін қолданады. Осындай веб қосымшаның бірі – Insightly (insightly.com). Бұл компания дайын CRM өнімдерің клиенттерге ұсынады.



## 1.3-сурет – “Insightly” компаниясының CRM платформасы

## 2 Технологиялар бөлімі

### 2.1 MEAN stack веб қосымшасы

Бұл веб қосымша қазіргі кездегі программалаудағы соңғы технологиялардың бірігуімен жасалынған. Солардың технологиялардың бірі MEAN stack. MEAN stack веб-қосымшаларды әзірлеу үшін пайдаланылатын JavaScript технологиясы жиынтығына жатады. Осылайша, клиенттен серверге және деректер базасына дейін барлығы JavaScript тілінде негізделген. MEAN – бұл жылдам және сенімді веб-қосымшаны әзірлеу үшін пайдаланылатын Javascript технологияларының жиынтығы. MEAN – пайдаланушыға ыңғайлы динамикалық веб-сайттар мен қосымшаларды құру үшін тамаша шешім болып табылатын стек. MEAN стек төрт түрлі технологиядан тұрады:

- MongoDB – бұл NoSQL деректер қоры;
- Express JS – веб-қосымшаларды құру үшін қолданылатын фреймворк;
- Angular – Google компаниясы әзірлеген JavaScript-фреймворк;
- Node.js – JavaScript серверлік ортасы.

MongoDB – C++ тілінде жазылған ашық бастапқы коды бар кросс-платформалық деректер базасы. Ол деректерді JSON сияқты мәліметтердің екілік түрін қолдана отырып, кілт жұптарында сақтайды. Бұл құжат – NoSQL-ге бағытталған деректер қоры болып табылады. MongoDB – бұл деректер қоры жүйесі миллиондаған деректер кестелерін басқару үшін тамаша таңдау. Сонымен қатар, MongoDB-ге жаңа кесте қосу оңай, себебі ол бүкіл кестені жаңартуды талап етпейді.

Express – жеңіл серверлік фреймворк. Ол бір беттік, көп беттік және гибриді веб-қосымшаларды құруға арналған. Бұл жеңіл фреймворк үлгіні қолдану үшін Pug қозғалтқышын пайдаланады. Express – бұл Node.js үшін де-факто фреймворк болып саналады.

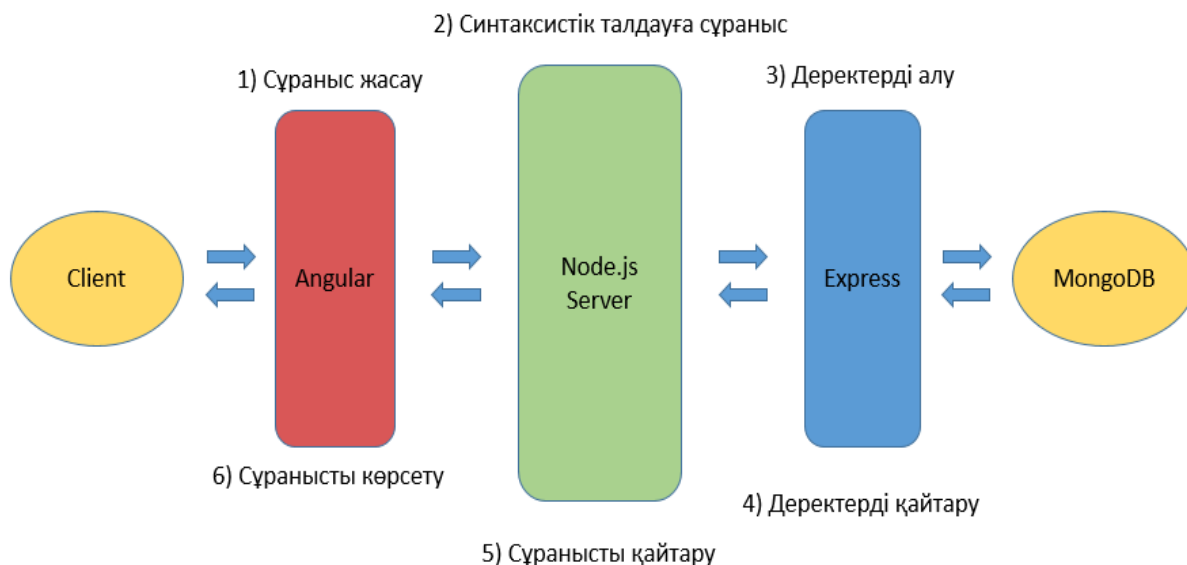
Angular – ашық бастапқы коды бар JavaScript фреймворк. Angular-ды Google компаниясы қолдайды. Бұл платформаның мақсаты MVC(Model View Controller) архитектурасын браузерлік қосымшаға енгізу болып табылады, бұл әзірлеу және тестілеу процесін жеңілдетеді. Бұл платформа дербестендіруді қолдайтын ақылды веб қосымшаны жасауға көмектеседі. Angular үлгі тілі ретінде HTML гипермәтіндік тілді пайдалануға мүмкіндік береді. Осылайша, қолданба компоненттерін көрсету үшін HTML синтаксисін кеңейте аласыз.

Node.js веб-серверлерді жасауға және оларға веб-қосымшаларды жасауға мүмкіндік береді. Бұл Javascript-тің серверлік ортасы. Node.js басқару енгізушығару моделін пайдаланады. Бұл оны оңай әрі тиімді етеді. MEAN stack архитектурасының әр түрлі компоненттерінің негізгі функциялары:

- Angular: сұрауларды қабылдау / соңғы пайдаланушыға нәтижелерді көрсету;
- Node.js: клиенттік және серверлік сұраныстарды өңдеу;
- Express JS: деректер базасына сұраныстарды орындау / жауапты қайтару;



– MongoDB: деректерді сақтау/шығару.



## 2.1-сурет – MEAN stack архитектурасының диаграммасы

- біріншіден, клиент Angular өңдейтін сұраныс жасайды;
  - содан кейін сұрау Node.js бөліміне жылжиды, ол сұранысты талдайды;
  - express деректерді алу немесе орнату үшін MongoDB-ге сұраныс жасайды;
  - mongoDB-ден қажетті деректерді алып, Express-ке сұранысты қайтарады;
  - Node.js клиентке сұранысты қайтарады;
  - алынған дерек Angular клиент жағында көрінеді.
- Жоғарыда аталған технологияларды келесі бөлімдерде жеке қарастыратын боламыз.

## 2.2 MongoDB кроссплатформалық деректер базасы

MongoDB – бұл құжатталған NoSQL, яғни үлкен көлемді деректерді сақтау үшін қолданылады. Таблицалар мен жолдарды дәстүрлі реляциялық деректер базасындағы сияқты пайдаланудың орнына MongoDB коллекциялар мен құжаттарды пайдаланады. Құжаттардың кілттік мәні – MongoDB негізгі деректер бірлігі болып табылатын мән. Коллекцияда реляциялық деректер қоры кестелерінің баламасы болып табылатын құжаттар мен функциялар жиынтығы бар. MongoDB ерекшеліктері:

- әрбір дерекқор коллекциядан тұрады, олар өз кезегінде құжаттардан тұрады. Әрбір құжат әр түрлі ортада болады. Әрбір құжаттың мөлшері мен мазмұны бір-бірінен өзгеше болуы мүмкін.

– құжат құрылымы көбінесе тиісті бағдарламалау тілдерінде өз сыныптары мен объектілерін қалай құратынына байланысты сәйкес келеді.

– mongoDB қол жетімді деректер моделі иерархиялық қатынастарды оңай ұсынуға, массивтер мен басқа да күрделі құрылымдарды сақтауға мүмкіндік береді.

## 2.1-кесте – MongoDB архитектурасының негізгі компоненттері

<b>_Id</b>	<b>SubId</b>	<b>CandidateName</b>	<b>docsSrc</b>
563479cc8a8a4246bd27d784	111	Kenessary	File.pdf
563479cc7a8a4246bd47d784	112	Zakir	Text.txt
563479cc9a8a4246bd57d784	113	Keness	Docs.exe

*\_id* – әрбір MongoDB құжатында қажетті компонент. *\_id* компоненті MongoDB құжатында бірегей мән болып табылады. *\_id* компоненті құжаттың бастапқы кілтіне ұқсайды. Егер жаңа құжатты *\_id* компонентсіз жасасаңыз, MongoDB автоматты түрде бұл компонентті жасайды. Мысалы, жоғарыда келтірілген Submission кестесінің мысалын көрсек, MongoDB коллекциядағы әрбір құжатқа 24 таңбалы бірегей идентификаторды қосады.

*Коллекция* – MongoDB құжаттарын топтастырады. Коллекция – Oracle немесе MS SQL сияқты кез келген басқа RDMS-де жасалатын кестенің баламасы. Коллекция бір дерекқор ретінде бола алады. Коллекциялар ешқандай құрылымды бұзбайды.

*Курсор* – бұл нәтижені сұрау жиынының көрсеткіші. Клиенттер нәтиже алу үшін курсор арқылы итерацияны орындай алады.

*Деректер қоры* – коллекцияларға арналған контейнер. MongoDB сервері бірнеше деректер қорын сақтай алады.

*JSON* – бұл JavaScript объектілік нотасы. Бұл құрылымдалған деректерді білдіру үшін ыңғайлы, қарапайым мәтіндік пішім. Қазіргі уақытта JSON көптеген бағдарламалау тілдерінде қолдау көрсетеді.

## 2.3 Express.js фреймворкі

Express.js – Node.js-тың веб қосымшалар жасауға көмектесетін фреймворкі. Веб қосымшаны тез әрі оңай жасауға ықпал ететін ерекшелігі бар, қосымша жасау барысында тек Node.js-ты қолданса көп уақыт кетеді.

Express.js аралық бағдарламалық қамтамасыз ету модулі connect деп аталады, ол өз кезегінде http-модульді пайдаланады. Осылайша, connect-ке негізделген кез келген аралық бағдарламалық жасақтама Express-пен жұмыс істейтін болады.



**2.2-сурет – Express.js архитектурасының диаграммасы**

Express.js-тің артықшылықтары:

- Node.js-тің тез әрі оңай жұмыс істеуіне әсер етеді;
- қолданба роутерлерін(routers) HTTP және URL-адрестері негізінде анықтайды;
- сұрау және жауап бойынша қосымша тапсырмаларды орындау үшін пайдалануға болатын аралық бағдарламалық(middlewares) қамтамасыз етудің түрлі модульдерін қамтиды;
- қателерді өңдеу үшін аралық бағдарламалық жасақтаманы анықтауға мүмкіндік береді;
- REST API серверін жасауға мүмкіндік береді;
- MongoDB, Redis, MySQL сияқты деректер қорына оңай қосылады.

## **2.4 Angular 2.0 фреймворкі**

Angular – Javascript фреймворкі мобильді және десктопты веб-қосымшаларды әзірлеу платформасы. Angular үлкен және күрделі бизнес логикалық қосымшалар жасауға мүмкіндік береді. Angular AngularJS-тың жаңартылған нұсқасы болғандықтан көптеген артықшылықтар мен ерекшеліктер пайда болды, және осы тұрғыдан жоғары масштабтауға ие болды. Angular 2.0 фреймворкінің бірқатар артықшылықтары :

- Google, Microsoft компанияларынан қолдау тапқан;
- әзірлеуші құралдарымен қамтамасыз етеді (CLI);
- жобаның бірыңғай құрылымы бар;
- TypeScript (қатаң типизацияланған кодты жазу мүмкіндігі);
- Rxjs реактивті бағдарламалауы бар;
- HTML кеңейтуіне негізделген үлгілер;
- кроссбраузерлік Shadow DOM;
- Http, WebSockets, Service Workers кроссбраузерлік қолдау;
- React, Vue деңгейіндегі фреймворк;

– үлкен комьюнити.

Angular кез келген күрделілік деңгейіндегі өндірістік және жылдам жұмыс істейтін қосымшаларды әзірлеу үшін келесі тұжырымдамаларда сипатталған:

*Form Builder* – күрделі формаларды жасау үшін декларативті форманың орнына реактивті формаларды қолдану қажет. Жақсы мысалдардың бірі (реактивті форма + валидация);

*Change Detection* – Angular деректер үлгісін екі жақты байланыстыруды пайдаланғандықтан, мұндай деректердің үлкен көлемімен жұмыс істегенде қолданбалар баяу жұмыс істейді, сондықтан кейбір жағдайларда өзгерістерді анықтаудың дұрыс стратегиясын ойластыру керек. Түрлі Open Source жобаларын көре аласыз: Priming, Angular Material, Clarity UI, Angular Bootstrap және басқалары, олар барлық ChangeDetection.OnPush пайдаланады.

*Routing* – бұл веб-қосымшаларды әзірлеудегі негізгі құбылыстардың бірі. Бұл жерде маршруттау компоненттер сияқты өмірлік циклы бар екенін түсіну өте маңызды, оны түсіну арқылы қосымшаларды жазуға болады. Айта кету керек, егер маршруттардың қайсыбірінде сіз осы маршрут бойынша бетті көрсетуге жауап беретін компонент емес, модульді енгізсеңіз, модуль талап ету бойынша бетте жүктелетін болады.

### 2.4.1 Angular 2.0 қосымшасының архитектурасы

#### *Модуль*

Angular қосымшаның проектілік архитектурасы модульдерден басталады. Модуль дегеніміз – қосымшаны құру үшін басқа модульдермен біріктіруге болатындай, өзара байланысты компоненттерді, директиваларды, каналдар мен қызметтерді топтастырудың тетігі. Олардың әрқайсысында өзіндік құрылымдық элементтер бар:

- компонент (component) – веб-беттің бөлігіне жауап береді және HTML шаблонын, CSS стильдерін және мінез-құлық логикасын қамтиды;
- сервис (service) - компонент үшін мәліметтер жеткізушісі;
- директива (directive) – DOM белгілі бір бөлігін белгілі бір жолмен түрлендіреді.

Жоғарыда айтылғандардың барлығы AppModule деп аталатын түбірлік модульде жинақталады.

@NgModule () – объектіні сипаттайтын модульді қабылдайтын декоратор.

#### *Компонент*

Компонент – бұл өзіндік логикасы бар қолданбалы интерфейс бөлігі. Angular қосымшасының барлық көрінетін бөлігі компоненттерді қолдану арқылы орындалады. Компоненттердің жасалуына @Component () декораторы жауап береді. Декоратор қабылдайтын негізгі мәндер:

- селектор – компоненттің атауы;

- шаблон (немесе `templateUrl`) – HTML ретінде жолды белгілеу (немесе HTML файлына апаратын жол);
- провайдерлер – компонент үшін деректерді ұсынатын қызметтер тізімі;
- стильдер – жасалатын құрамдас бөлікке арналған стильдерден тұратын CSS файлдарына апаратын массив.

### *Сервис*

Қызметтер компоненттерге деректер беру үшін қажет. Бұл тек серверге сұраулар ғана емес, сонымен бірге берілген деректерді алгоритм бойынша түрлендіретін функциялар болуы мүмкін. Олар `ngular` қосымшасының архитектурасын неғұрлым икемді және масштабталатын етуге мүмкіндік береді.

### *Директива*

Директивалардың мақсаты – DOM-ды белгілі бір жолмен түрлендіру, элементті мінез-құлқы бойынша ерекшелеу. Директиваның екі түрі бар:

- құрылымдық – DOM-ға элементтерді қосу, алып тастау және ауыстыру;
- атрибуттар – элементке басқа мінез-құлық береді.

Олар конфигурация объектісі бар `@Directive ()` декораторының көмегімен жасалады.

## **2.5 Node.js серверлік платформа**

`Node.js` – бұл жылдам және масштабталатын желілік қосымшаларды оңай құру үшін JavaScript орындау ортасында құрылған платформа. `Node.js` басқару оқиғаларын және енгізу-шығару моделін пайдаланады, бұл оны жеңіл және тиімді етеді, әртүрлі құрылғыларда жұмыс істейтін нақты уақыттың қарқынды қосымшалары үшін мінсіз.

`Node.js` – серверлік және желілік қосымшаларды әзірлеу үшін ашық бастапқы коды бар кросс-платформалық орта. Сонымен қатар JavaScript модульдерінің бай кітапханасын ұсынады. `Node.js` асинхронды бағдарламалауды пайдаланады! Веб-сервер үшін әдеттегі тапсырма серверде файлды ашу және деректерді клиентке қайтару болып табылады. `Node.js` файл сұранысын өңдеу процесстері:

- сұранысты компьютердің файлдық жүйесіне жібереді;
- келесі сұранысқа жауап беруге дайын;
- файлдық жүйе файлды ашып, оқығанда сервер сұраныстағы деректерді клиентке қайтарады;
- `Node.js` ұзақ күтілімді жояды және келесі сұранысты орындауды жалғастырады.

## 2.6 Pusher.js нақты уақыттағы деректер және Chart.js

Pusher.js – бұл нақты уақыттағы деректер мен функцияларды веб және мобильді қосымшаларға қосуды жеңілдететін хостинг қызметі. Аталмыш хостинг қызметі деректерге мәліметтер қосылу кезінде, веб бетті жаңартпай, деректерді нақты уақытта дереу алуға ықпалын тигізеді.

Pusher нақты уақытта сіздің серверлеріңіз бен клиенттеріңіз арасында байланыс орнатады. Pusher, мүмкін болса, клиенттермен тұрақты байланыс орнатады және HTTP қосылымына жүгінеді – сервер деректерді қабылдап алғаннан кейін дереу клиент бөліміне деректерді шығарады.

Pusher барлық негізгі жұмыс орталары мен шеңберлеріне интеграция үшін кітапханаларды ұсынады. Сервердегі PHP, Ruby, Python, Java, Node және JavaScript тілдерін қолдайды.

Chart.js – бұл деректерді визуализациялауға арналған ашық коды бар JavaScript кітапханасы. Бұл сандық деректерді диаграмма, график және тізбек түрінде шығаруға көмектеседі.

Бұл дипломдық жұмыста үміткерлерге дауыс беру кезінде берілген дауыстар барлық комиссия мүшелеріне бір уақытта көрініп тұруы үшін Pusher.js және Chart.js кітапханаларын пайдаландым.

## 3 Жоба құрылымы

### 3.1 «Satbayev Hunters» жобасы туралы

«Satbayev Hunters» жобасы көпбетті веб қосымша болып табылады. Веб қосымша университет қабырғасында жұмыс істегісі келетін білікті мамандардың өтінішін қабылдауға арналған. Бұл жоба Express.js технологиясын қолдана отырып жасалынды.

Express.js веб қосымшаны қолданушы клиент пен сервердің арасындағы байланысты қамтамасыз етеді және ақпаратты тез алуға септігін тигізеді. Жобаның негізгі қызметі жұмысқа үміткерлердің өтініштері мен құжаттарын қабылдау болып саналады. Сол себепті маңызы бар құжаттарды электронды түрде жіберу кезінде веб қосымша қауіпсіздік жоғарғы деңгейде қамтамасыз етеді.

### 3.2 Веб қосымшаның интерфейстері

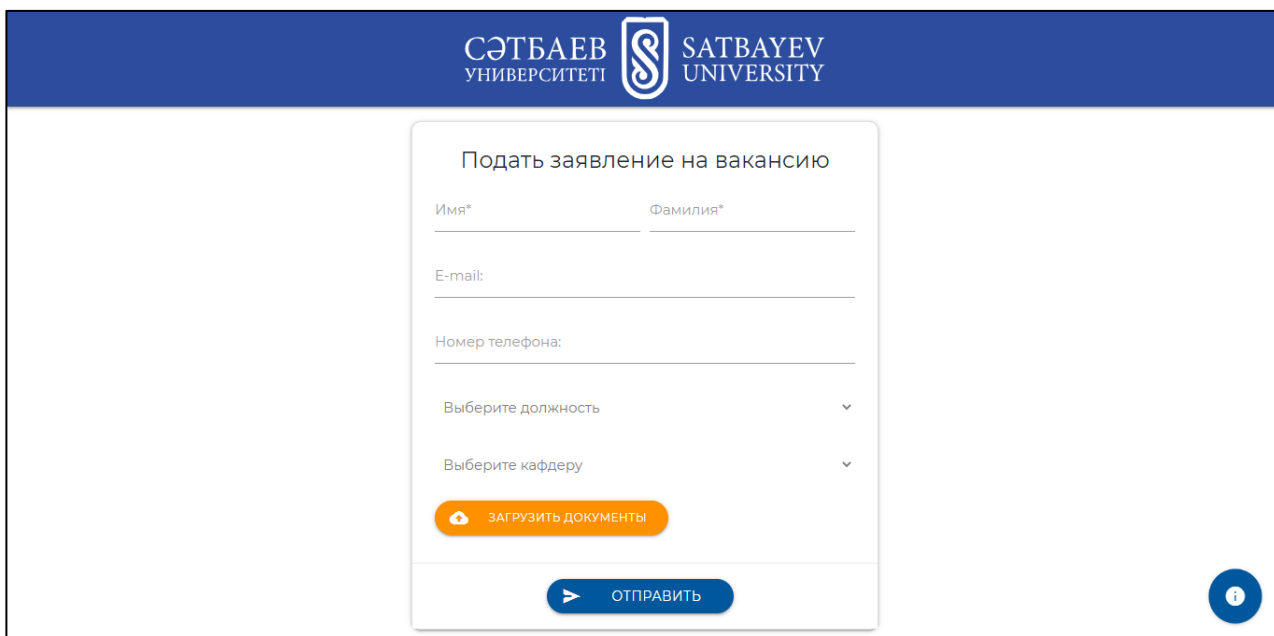
«Satbayev Hunters» веб қосымшасының интерфейсі басты бет және бірнеше бөлімнен тұрады. Олар: басты бет, өтініш қалдыру беті, интервью туралы бет, комиссия мүшелерінің беті (авторизация/регистрация), администратордың беті (авторизация).

Басты бетте “Өтініш қалдыру” батырмасын басып өтініш беру бетіне өтуге болады. Төменде 3.1-суретте көрсетілген.



3.1-сурет – Басты бет көрінісі

“Өтініш қалдыру” батырмасын басқан соң, өтініш беру беті ашылады. Бұл бетте жұмысқа үміткер маман аты-жөнің, электронды почтасын, телефон нөмерін, қалаған лауазымын, таңдаған кафедрасын және қажетті құжаттарын тіркеу арқылы өтінішін қалдыра алады. Өтініш беруші өзінің құжаттарын бір .pdf файлына біріктіру қажет және файлдың жалпы көлемі 5 МБ-тан аспау қажет. Өтініш қалдыру формасы 3.2-суретте көрсетілген.



СЭТБАЕВ  
УНИВЕРСИТЕТИ

SATBAYEV  
UNIVERSITY

Подать заявление на вакансию

Имя\* \_\_\_\_\_

Фамилия\* \_\_\_\_\_

E-mail: \_\_\_\_\_

Номер телефона: \_\_\_\_\_

Выберите должность ▾

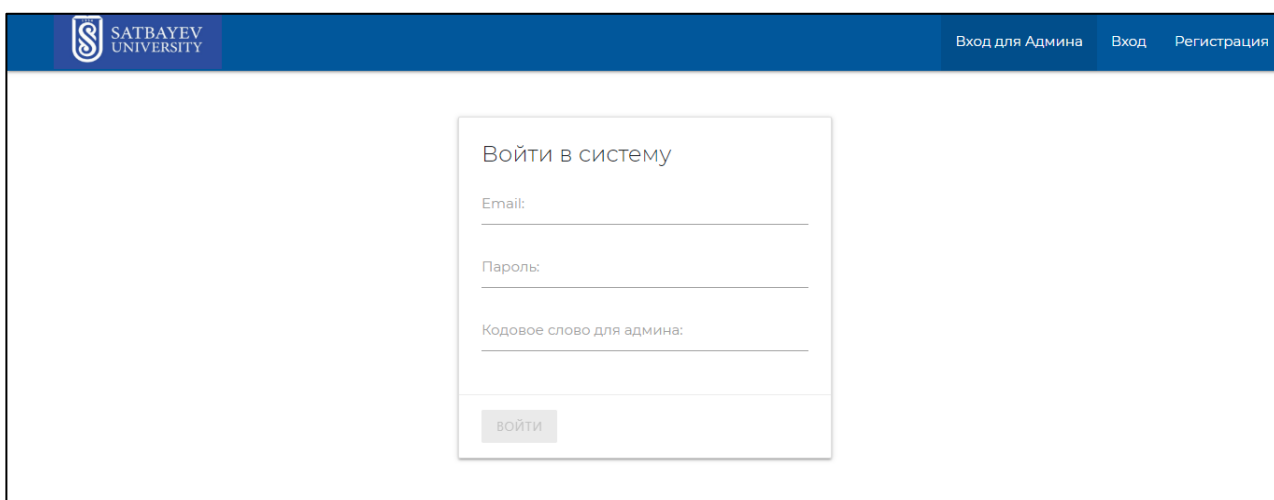
Выберите кафедру ▾

ЗАГРУЗИТЬ ДОКУМЕНТЫ

ОТПРАВИТЬ

### 3.2-сурет – Өтініш қалдыру формасы

Түскен өтініштерді сараптау үшін және үміткерді конкурсқа тағайындау үшін администратордың атымен кіру қажет болады. Администратордың атымен кіру үшін электронды почтаны, парольді және кодтық сөзді енгізу қажет болады.



SATBAYEV  
UNIVERSITY

Вход для Админа Вход Регистрация

Войти в систему

Email: \_\_\_\_\_

Пароль: \_\_\_\_\_

Кодовое слово для админа: \_\_\_\_\_

ВОЙТИ

### 3.3-сурет – Администратор авторизациясы



Авторизация сәтті орындалғаннан кейін, администратор “Үміткерлер” бөліміне өтеді. Бұл бетте администратор барлық түскен өтініштердің тізімін көре алады. Сонымен қатар кез келген үміткерді аты-жөні, электронды почтасын теру арқылы тауып алуға болады және өтініштерді түсу уақытымен фильтрлей алады. Сипаттамаларды 3.4-суреттен көре аласыздар.

ФИО	Email	Должность	Время подачи заявок	Подробнее
Магжан Жумабаев	kenessaryz@mail.ru	Заведующий(ая) кафедры	28.05.2020 [03:08]	
Сакен Сейфуллин	kenessaryz@gmail.com	Лектор	28.05.2020 [03:08]	
Ыбырай Алтынсарин	kenessaryz@gmail.com	Лектор	28.05.2020 [03:03]	
Nuralim Zakir	nuralim@gmail.com	Ассоциированный профессор	28.05.2020 [00:08]	

[ЗАГРУЗИТЬ ЕЩЕ](#)

### 3.4-сурет – Үміткерлер тізімінің беті

Әрбір үміткердің жеке мәліметтерің білу үшін сол үміткердің аты-жөні тұрған тұсты басып, үміткердің өтініш бергендегі мәліметтерін көруге болады, және дұрыс жіберілмеген өтініштерді “Өшіру” батырмасымен қате үміткерді өшіре алады. Аталмыш іс-әрекеттер 3.5 – суретте бейнеленген.

Кандидаты > Заявка кандидата

Имя, Фамилия: Магжан Жумабаев

Email: kenessaryz@mail.ru

Телефон: 87056234589

Должность: Заведующий(ая) кафедры  
Кафедра Транспортная техника (Институт промышленной инженерии)

[+ ДОБАВИТЬ К ГОЛОСОВАНИЮ](#)

Пригласить кандидата на собеседование

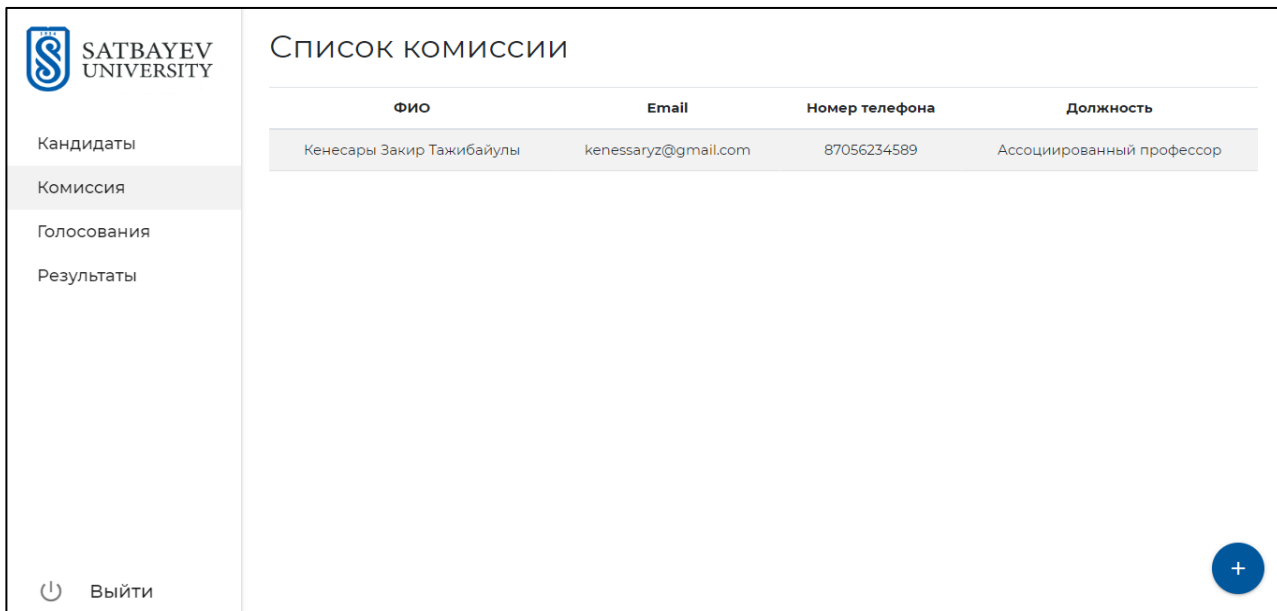
E-mail\*

Сообщение для приглашения

[ОТПРАВИТЬ](#)

### 3.5-сурет – Үміткер туралы жеке ақпараттар

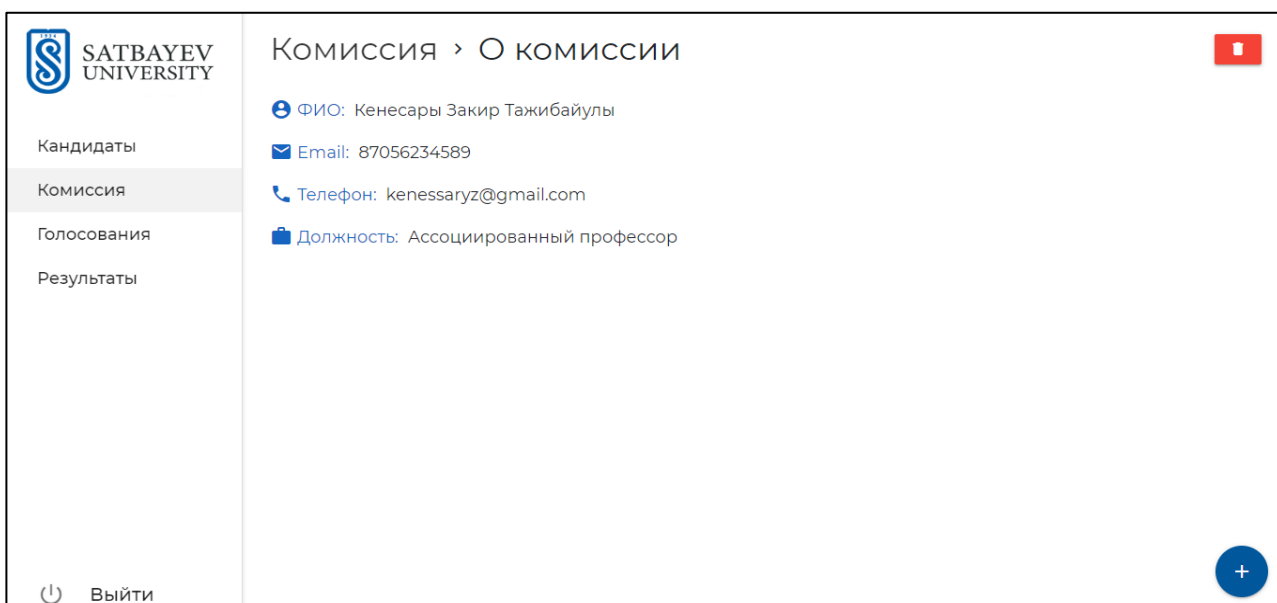
Жоғарыда көрсетілген 3.5 – суретте қызыл батырманы басу арқылы үміткердің құжаттардың онлайн түрде браузердің ішінен көре беруге болады. Келесі бөлімде, яғни “Комиссия” бөлімінде жүйеге тіркелген комиссиялардың тізімін көреміз. Бұл тізімдегі комиссиялар жұмысқа үміткерге дауыс беріп шешім қабылдау үшін жүйеге тіркелу қажет.



ФИО	Email	Номер телефона	Должность
Кенесары Закир Тажибайулы	kenessaryz@gmail.com	87056234589	Ассоциированный профессор

### 3.6-сурет – Жүйеге тіркелген комиссиялар тізімі

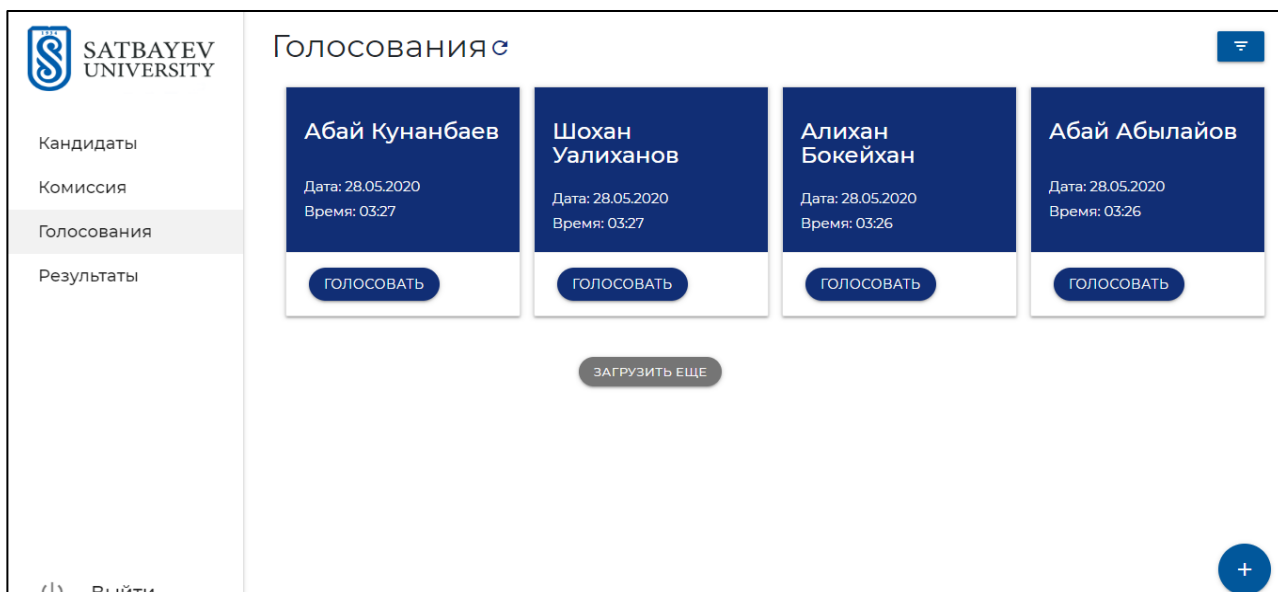
Кез-келген комиссия мүшесі туралы толық ақпарат білу үшін комиссия мүшесінің аты-жөні тұрған жерді басып келесі бетке өтуге болады. Аккаунты қатып қалған комиссия мүшесін өшіру батырмасын басып өшіруге болады. Комиссия мүшесі туралы ақпарат 3.7-суретте бейнеленген.



ФИО: Кенесары Закир Тажибайулы
Email: 87056234589
Телефон: kenessaryz@gmail.com
Должность: Ассоциированный профессор

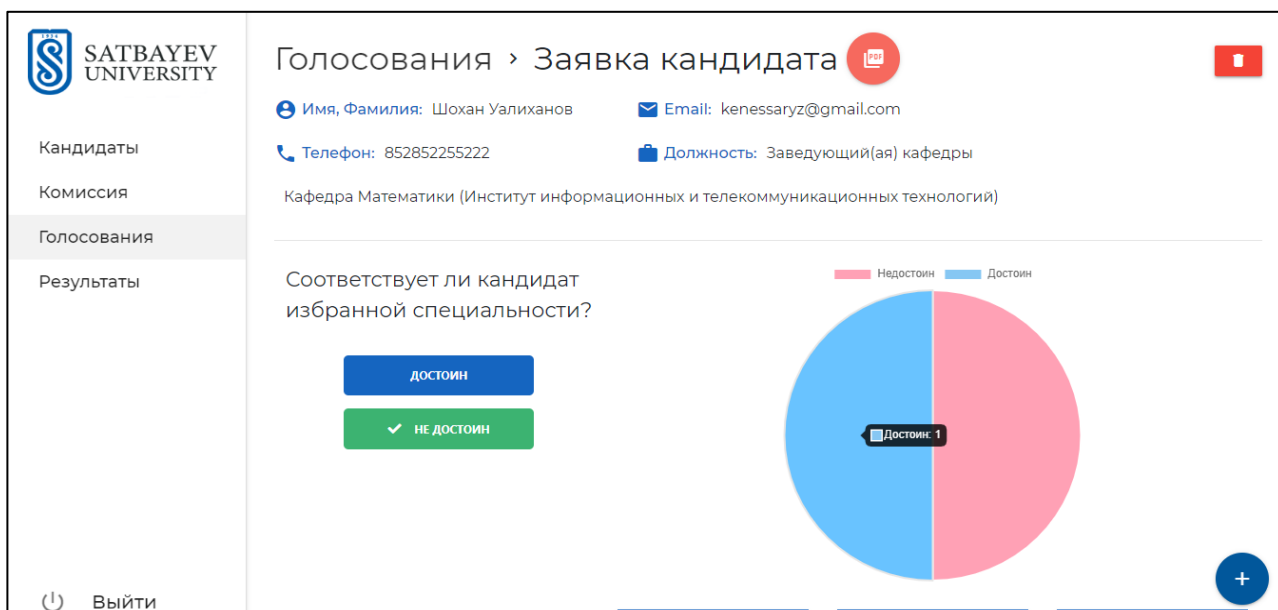
### 3.7-сурет – Комиссия мүшесі туралы толық ақпарат

Кандидатқа дауыс беру нәтижелерін сақтау үшін Дауыс беру бөліміне кіру қажет. Дауыс беру бөлімінде кезек күтілімі бойынша кандидаттар тізімі шығады. Әрбір кандидатқа кіру арқылы сол кандидатқа дауыс берілген нәтижелерін сақтауға болады. Сипаттамалар 3.8-суретте көрсетілген.



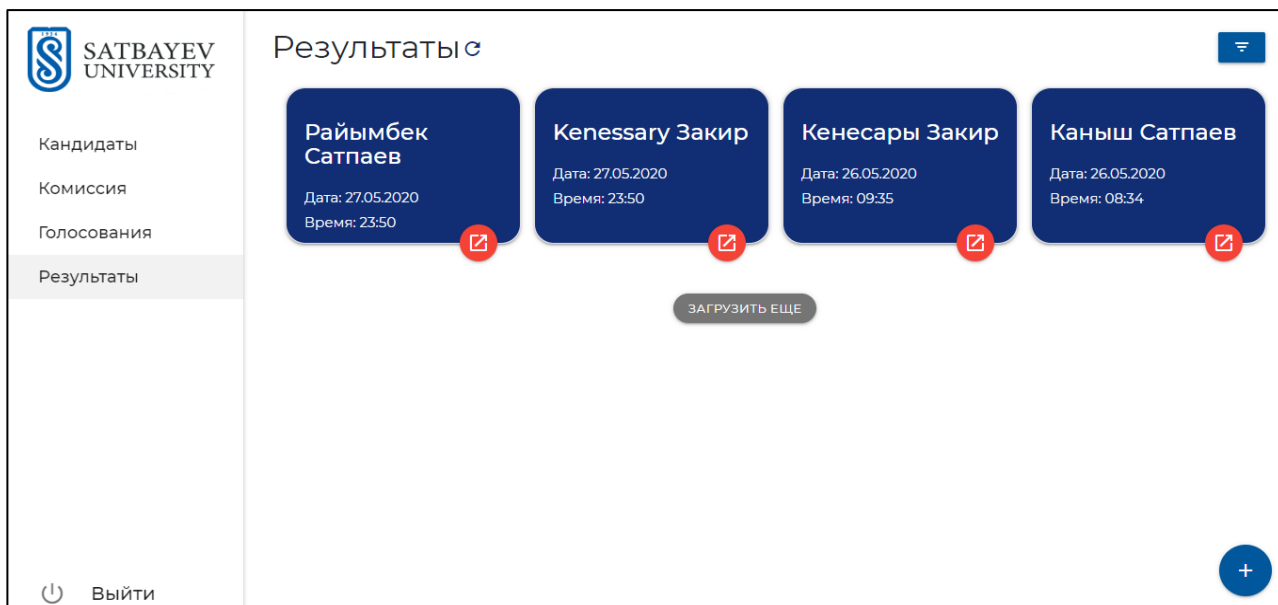
**3.8-сурет – Дауыс беру күтіліміндегі кандидаттар тізімі**

Комиссия мүшелері берген дауыстарды сақтау үшін жеке бір кандидаттың бетіне кіру арқылы сақтауға болады. Дауыстар нәтижесі диаграмма түрінде барлық комиссиялар мүшелері мен администратордың бетінде шығып тұрады. Айтылған сипаттамаларды 3.9-суреттен көре аласыздар.



**3.9-сурет – Дауыс беру кезіндегі кандидаттың нәтижесін сақтау**

Кандидаттардың сақталған нәтижелерін “Результаты” бөліміне кіріп қарауға болады. Айтылған сипаттамаларды 3.10-суреттен көре аласыздар



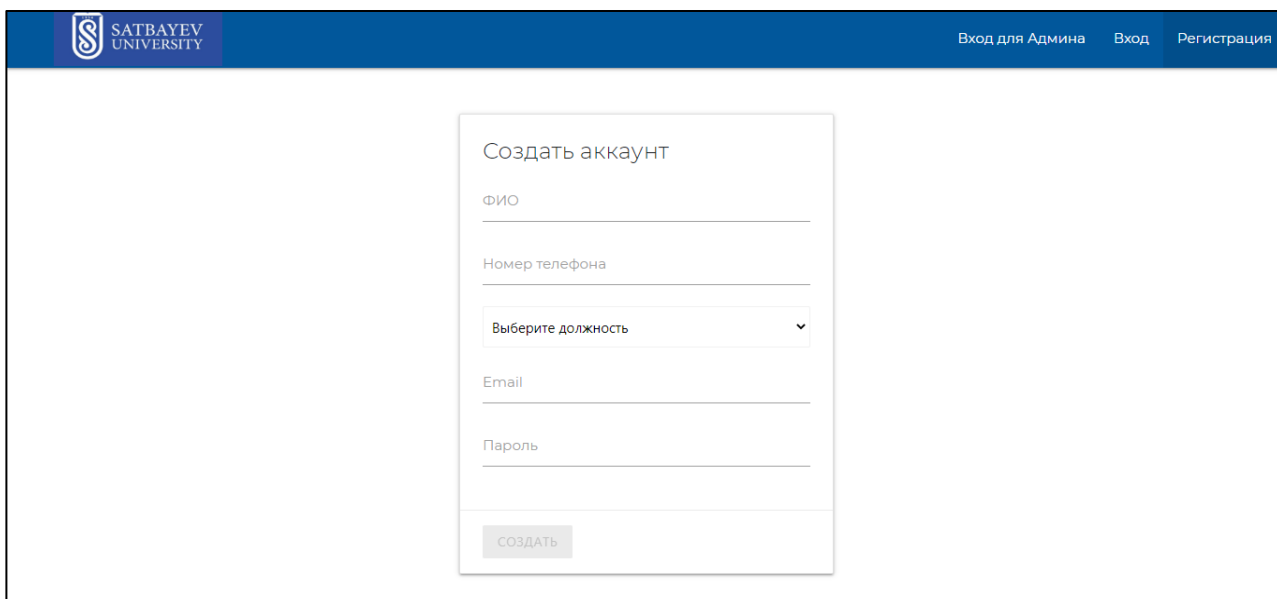
### 3.10-сурет – Нәтижелер бөлімі

Нәтижелер бөліміндегі жеке бір кандидаттың нәтижесің қарау үшін “толығырақ” батырмасын басып қарауға болады. Айтылған сипаттамаларды 3.11-суреттен көре аласыздар



### 3.11-сурет – Комиссия мүшесінің регистрация жасау бөлімі

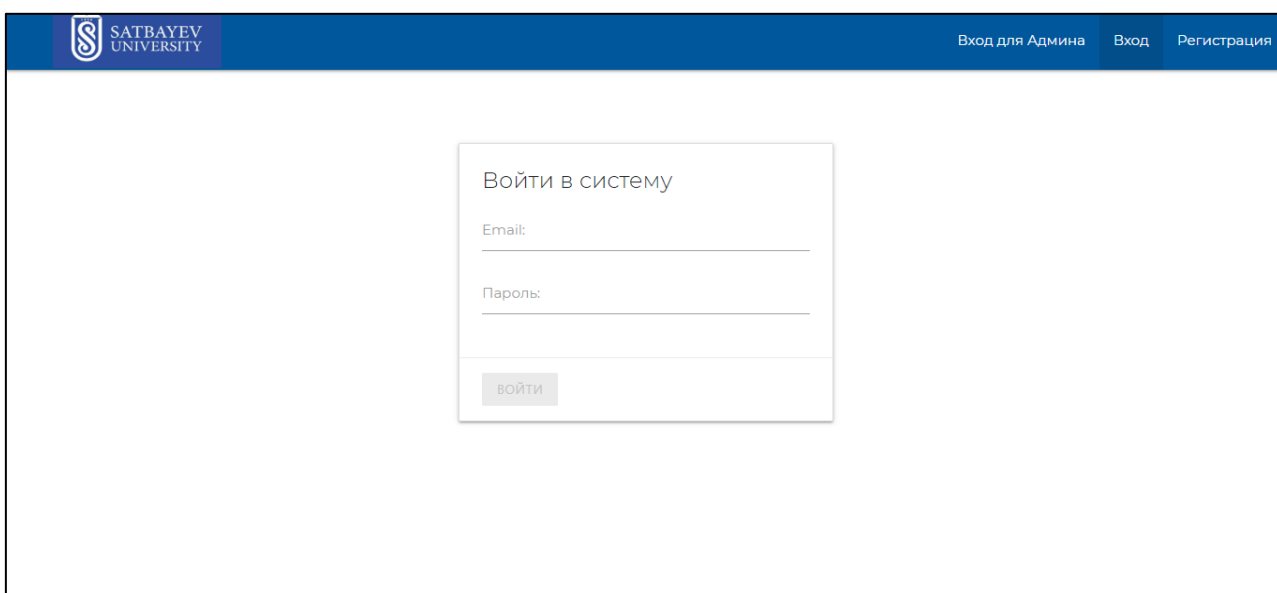
Комиссия мүшесі болып жүйеге кіру үшін “Регистрация” батырмасын басасыз. Келесі бетте толық аты-жөн, телефон-нөмір, лауазым, электронды почта, парольді қою арқылы сіз жүйеге комиссия мүшесі болып кіруіңізге болады. Айтылған сипаттамаларды 3.12-суреттен көре аласыздар



The screenshot shows the registration page of the Satbayev University website. The header includes the university logo and name, and navigation links for 'Вход для Админа', 'Вход', and 'Регистрация'. The main content area features a form titled 'Создать аккаунт' (Create account). The form contains the following fields: 'ФИО' (Full name), 'Номер телефона' (Phone number), a dropdown menu for 'Выберите должность' (Select position), 'Email', and 'Пароль' (Password). A 'СОЗДАТЬ' (CREATE) button is located at the bottom of the form.

### 3.12-сурет – Комиссия мүшесінің регистрация жасау бөлімі

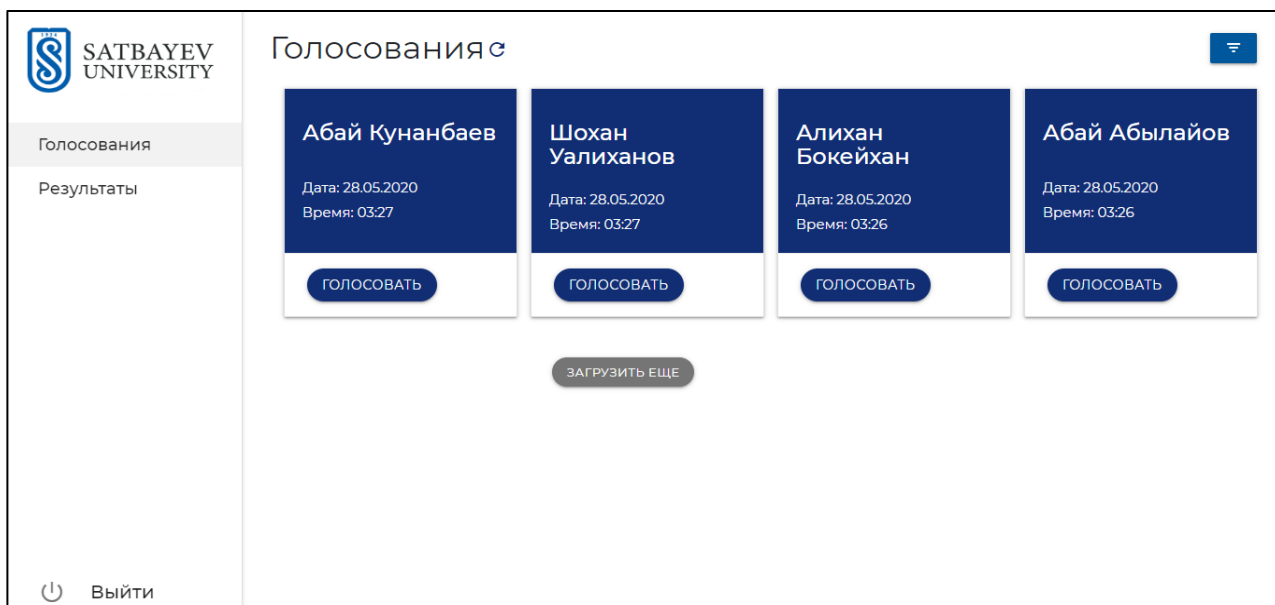
Жүйеге тіркеліп болғаннан кейін комиссия мүшесін веб қосымша автоматты түрде жүйеге тіркелу бөліміне өткізеді. Бұл жерде комиссия мүшесі пошта адресін және парольын жазу арқылы жүйеге ене алады. Айтылған сипаттамаларды 3.13-суреттен көре аласыздар



The screenshot shows the login page of the Satbayev University website. The header is identical to the registration page. The main content area features a form titled 'Войти в систему' (Log in to the system). The form contains two fields: 'Email:' and 'Пароль:' (Password). A 'ВОЙТИ' (LOG IN) button is located at the bottom of the form.

### 3.13-сурет – Комиссия мүшесінің жүйеге кіру бөлімі

Коммисия мүшесінің беті тек екі бөлімнен тұрады: дауыс беру және нәтижелер бөлімі. Бұл бөлімдерде кандидаттарға дауыс беріп, артынша нәтижесін көруге болады. Айтылған сипаттамаларды 3.14-суреттен көре аласыздар



### 3.14-сурет – Комиссия мүшесінің беті

Үміткерлер өтініштерін бермес бұрын жалпылыма мәліметтерді және жұмысқа кіру құжаттарын біле алады. Барлық институттар мен кафедраларға арналған сұқбаттасу орыны және уақыты кестеде көрсетіледі. Айтылған сипаттамаларды 3.15-суреттен көре аласыздар.




#### Конкурс на замещение вакантных должностей Satbayev University

Лица, не являющиеся работниками ППС КазНИТУ, для участия в конкурсе подают следующие документы:

- заявление (по форме Приложения 4);
- резюме (по форме Приложения 2);
- копии дипломов о высшем образовании, академической и ученой степени, ученом звании и подлинники указанных документов для сверки;
- копии сертификатов о переподготовке и повышении квалификации (при наличии) и подлинники для сверки;
- копии документов, подтверждающих трудовой стаж претендентов, и подлинники для сверки;
- заверенный по месту работы список выполнявшихся научных проектов, научных работ и изобретений за последние 3 года.

[Квалификационные требования](#)  
[Форма резюме](#)  
[Форма заявления](#)

Место проведения конкурса: Нефтяной корпус, каб. 316.

№	Институт (по кафедрам)	День аттестации	Приблиз. время начала аттестации*
1	Институт базового образования Кафедра общественных дисциплин	10 июня	10:30

### 3.15-сурет – Үміткерге арналған мәліметтер

## ҚОРЫТЫНДЫ

Дипломдық жобаны қорытындылай келе, жалпы Javascript тілімен жазылған “Satbayev Hunters” веб қосымшасы университеттегі білікті мамандарды көбейтіп, өтініш формасын электрондандыру үшін жасалынды. Веб сайт жасалу барысында IT – нарығындағы соңғы технологияларды қолданып, көптеген кітаханаларды жұмысты тиімді әрі тездету үшін қолданылды. Веб қосымша вебте кеңінен қолданылатын Angular Javascript тілінің фрейворкімен жасалынды.

Веб қосымшаның негізгі ерекшелігі – көпфункционалдығы. Нақтырақ айтқанда, өтініш беру, өтінішті сараптау, өтінішке дауыс беру іс-әрекеттері бір жүйеде біріктірілген. Университетке түсетін өтініштердің көптігіне байланысты, электронды өтініш қалдыру жүйесі білікті мамандар үшін және кадр бөліміндегі қызметкерлердің жұмысын жеңілдету үшін таптырмас қосымша болады.

Веб қосымша әртүрлі платформалар мен браузерлерде функционалдарын тест сынақтан өткіздім. Нәтижесінде ескі нұсқадағы браузерлерде кейбір функционалдар мен дизайны көрінбей қалады. Дегенмен көптеген қолданушылар қазіргі кезде соңғы нұсқадағы браузерлерді қолданады және бұл веб қосымшаны мобильді браузерлер де қолдайды.

Фреймворктың көптеген функционалдықтарын үйрене отырып, веб қосымшаға болашақта көптеген жаңартулар енгізіледі. Бұл өз кезегінде қолданушылардың қауіпсіздігін жоғарғы деңгейде қамтамасыз етіп, кадр бөліміндегі қызметкерлердің жұмысын жеңілдететін қосымша беттерді жасауға мүмкіндік береді.

Сайт негізінен клиент жағы үміткерге көрінетін беттерден, комиссия мүшесінің жеке кабинеті (авторизация/регистрация жасау арқылы кіреді), администратордың жеке кабинеті(авторизация жасау арқылы кіреді) беттерден құрылған. Клиент пен серверді байланыстыру үшін Express.js серверлік фреймворкі қолданылды.

## ПАЙДАЛАНЫЛҒАН ӘДЕБИЕТТЕР ТІЗІМІ

1. Тема 8. Унифицированный язык моделирования UML // Сайттың электрондық нұсқасы <http://samara.mgpru.ru/~dzhadzha/dis/15/200.html>
2. Colin J Ihrig, Adam Bretz - Full Stack JavaScript Development With MEAN – SitePoint, 2014 – 40-page
3. Nicholas McClay - MEAN cookbook. The meanest set of mean stack solutions around - Packt Publishing Ltd, 2017 – 83-page
4. Архитектура приложения Angular. Используем NgModules // Сайттың электрондық нұсқасы <https://habr.com/ru/post/351678/>
5. Руководство по безопасности MongoDB // Сайттың электрондық нұсқасы <https://habr.com/ru/company/pgdayrussia/blog/329306/>
6. Работа с модулями Node.js // Сайттың электрондық нұсқасы <https://metanit.com/web/nodejs/2.2.php>
7. Why do we actually need Express.js? // Сайттың электрондық нұсқасы
8. <https://stackoverflow.com/questions/12616153/what-is-express-js>
9. Brad Dayley, Brendan Dayley, Caleb Dayley - Angular: A Hands-On Guide to Angular 2 and Angular 4 - Addison-Wesley, 2017 – 105-page
10. Minko Gechev - Switching to Angular 2 - Packt Publishing Ltd, 2016 – 139-page
11. Pusher - Leader In Real-time Technologies // Сайттың электрондық нұсқасы <https://github.com/pusher>
12. Отправка E-mail // Сайттың электрондық нұсқасы <https://www.xsltdev.ru/nodejs/tutorial/email/>
13. Simon Holmes – Getting MEAN with Mongo, Express, Angular and Node 1<sup>st</sup> Edition – Paperback, 2015 – 204-page
14. Ethan Brown - Web Development with Node and Express: Leveraging the JavaScript Stack 2nd Edition - O'Reilly Media, 2019 – 122-page
15. Greg Lim - Beginning Node.js, Express & MongoDB Development - Independently published, 2019 – 78page
16. Marijn Haverbeke - Eloquent JavaScript, 3rd Edition: A Modern Introduction to Programming, 2018 – 43-pages.



## **А Қосымшасы** (міндетті)

«Satbayev Hunters» веб қосымшасын құруға арналған  
техникалық тапсырма

### **А.1 Кіріспе**

Қазіргі кезде кеңсе жұмыстарының жұмысын қағаз ресурсынсыз жекеленген кәсіпке немесе кеңсе жұмыстарына арналған кросс-платформалық веб қосымшалар пайда болуда. Оны күнделікті қолданып жүрген отандық немесе әлемдік үздік веб қосымшалаынан байқауға болады. Бұл өз кезегінде дәстүрлі жолмен жұмыс істеуді жылжытып, барлық құжаттарды электрондандыру жүйесіне әкеліп соғады. Сондықтан кеңсе жұмыстарына қажетті веб қосымшалар қазіргі күннің өзекті тақырыбы. Десек те Satbayev Hunters веб қосымшасы кеңсе мәселерлерін шешу үшін жасалынды.

#### **А.1.1 Жобаның мақсаты мен міндеті**

Satbayev Hunters веб қосымшасы негізінен университетте жұмыс істегісі келетін мамандарға арналған. Үміткер мамандар кез-келген құрылғылардың браузерлерінен веб қосымшаға кіру арқылы өтініш беру құжаттар тізімін, жалпы өтініш ережесін және сұқбат өтетін орын мен уақытымен таныса алады. Сонымен қатар үміткер кез-келген уақытта өтінішін онлайн түрде кадр бөліміне жібере алады, ал сұқбат өтіп болғаннан кейін комиссия мүшелері үміткерге дауыс береді. Бұл қосымша тек Сәтбаев Университетіне арналған жоба болып табылады.

#### **А.1.2 Қолдану саласы**

Қолдану саласы – компьютер немесе мобильді телефон браузерлерін қолдана білетін кез-келген қолданушылар пайдалана алады.

### **А.1.3 Анықтамалар, терминдер және қысқартулар**

Анықтамалар, терминдер және қысқартулар А.1-кестеде көрсетілген.

#### **А.1-кесте – Анықтамалар, терминдер және қысқартулар**

Терминдер немесе қысқартулар	Анықтамалар
Браузер	гипермәтіндік байланысу мүмкіндігі бар файлдармен жұмыс істеуге арналған бағдарлама. Ол әркімнің таңдауы бойынша кәдімгі мәтіннің, статикалық бейненің немесе графиктердің экранда бейнеленуін және олармен байланысқан басқа файлдарды іздестіруді қамтамасыз етеді.
Фреймворк	сайтты жедел жасауға арналған құралдар жиынтығы, белгілі бір міндеттерді шешуге бағытталған. Бұл бағдарлама өнімін немесе сайтты әзірлеуді жеңілдететін көптеген түрлі кітапханалардан тұратын жүйе. Яғни фреймворк - бірнеше кітапханалардың жиынтығы. Фреймворктар өзінің мүмкіндіктері мен функциялары бойынша әр түрлі.
Angular	Javascript фреймворкі мобильді және десктопты веб-қосымшаларды әзірлеу платформасы. Angular үлкен және күрделі бизнес логикалық қосымшалар жасауға мүмкіндік береді.
MEAN stack	Бұл жылдам және сенімді веб-қосымшаны әзірлеу үшін пайдаланылатын Javascript технологияларының жиынтығы. MEAN - пайдаланушыға ыңғайлы динамикалық веб-сайттар мен қосымшаларды құру үшін тамаша шешім болып табылатын стек.  MEAN стек төрт түрлі технологиядан тұрады: MongoDB, Express.js, Angular, Node.js
Кросс-платформа	Әртүрлі құрылғыларда жұмыс істейтін веб қосымша

## **A.2 Жалпы сипаттамасы**

Жалпы веб қосымша дерекқоры бұлттық режимде жұмыс істейтін mLab онлайн дерекқор сервисінде сақталынған. Дерекқор мен клиент арасындағы сұранымдарды Express.js жүргізеді, кейін дерекқордан алынған ақпараттар клиенттік бетке дереу шығарылатын болады.

### **A.2.1 Пайдаланушы интерфейстер**

Кросс-платформалық құрылғыларда жұмыс істейтін бұл веб қосымшаның түрін қолданушылар қол жетімді әрі түсінікті интерфейсте қолдана алады.

### **A.2.2 Аппараттық интерфейстер**

Компьютерлерге қойылатын жалпы талаптар:

- процессор – Intel Pentium немесе AMD;
- оперативті жады – 2GB;
- дисктегі кеңістік – 52 MB
- ғаламтор желісі;
- соңғы нұсқадағы браузерлер (Google Chrome, Firefox, Opera, Microsoft Edge, Safari)
- пернетақта.

### **A.2.3 Программалық интерфейстер**

Жобаға қатысты бағдарламалық компоненттер:

- Windows 10 операциялық жүйесі (соңғы жаңартудағы нұсқа);
- Visual Studio Code – бастапқы кодты өзгертетін редактор ;
- Figma – веб дизайнерлік бағдарлама.
- Postman бағдарламасы – REST сұранымдарды жүргізу бағдарламасы.
- mLab – MongoDB деректерін сақтайтын онлайн сервис .
- Google Chrome браузері.

#### **А.2.4 Коммуникациялық интерфейстер**

Қолданушылардың жоғары жылдамдықты интернет желісі болу керек. Сервер мен клиенттерді байланыстыратын TCP/IP протоколы қолданылады.

#### **А.2.5 Жады бойынша шектеулер**

Деректер сақталатын mLab бұлттық сервисінде бірқатар шектеулер бар. Ашып айтқанда, бұл онлайн деректер сервисінде тегін және ақылы бөлімдері бар. Веб қосымша жасау барысында тегін бөлімін таңдадым, өйткені оның мүмкіншіліктері жеткілі болды. Веб қосымшаны хостқа салғаннан кейін веб деректер сервисын ақылы бөлімге көшірге жөн, жады бойынша шектеулері жойылып, жадының көлемі өседі.

#### **А.2.6 Адаптация бойынша талаптар**

Satbayev Hunters веб қосымшасы Android, iOS, Windows, Linux платформаларындағы браузерлерінде жұмыс істей алады. Ескі нұсқадағы браузерлерді кейбір элементтер шықпай қалуы мүмкін.

## Б Қосымшасы (міндетті)

### Бағдарлама мәтіні

#### 1. Жүйеге логин мен пароль арқылы кіру бетінің мәтіні

```
module.exports.login = async function(req, res){
  const candidate = await Admin.findOne({email: req.body.email})
  if(candidate){
    //проверка пользователя, пользователь существует
    const passwordResult = bcrypt.compareSync(req.body.password,
candidate.password)
    if (passwordResult){
      //генерация токена пароли совпали
      const token = jwt.sign({
        email: candidate.email,
        adminId: candidate._id
      }, keys.jwt, {expiresIn: 60 * 60})

      res.status(200).json({
        token: `Bearer ${token}`
      })
    }else{
      // пароли не совпали
      res.status(401).json({
        message: 'Пароли не совпадают. Попробуйте снова'
      })
    }
  } else {
    //пользователь нет ошибка
    res.status(404).json({
      message: 'Пользователь с таким email не найден.'
    })
  }
}
```

#### 2. Жүйеге тіркелу мәтіні

```
module.exports.registeradmin = async function(req, res){
  const candidate = await Admin.findOne({email: req.body.email})
  if (candidate){
    // пользователь существует, нужно отправить ошибку
    res.status(409).json({
```

```
    message: 'Такой email уже занят. Попробуйте другой'
  })
} else {
  // нужно создать пользователя
  const salt = bcrypt.genSaltSync(10)
  const password = req.body.password
  const admin = new Admin({
    fullname: req.body.fullname,
    phone: req.body.phone,
    position: req.body.position,
    email: req.body.email,
    password: bcrypt.hashSync(password, salt)
  })
  try{
    await admin.save()
    res.status(201).json(admin)
  } catch(e){
    errorHandler(res, e)
  }
}
}
```

### 3. Комиссиялар тізімінің мәтіні

```
<div class="page-title">
  <h4>Список комиссии</h4>
</div>
<div class="table-wrapper" *ngIf="commissions$ | async as commissions; else
loader">
  <div class="centereds" *ngIf="commissions.length !== 0; else empty">
    <table class="table table-striped table-hover centered">
      <thead>
        <tr>
          <th>ФИО</th>
          <th>Email</th>
          <th>Номер телефона</th>
          <th>Должность</th>
        </tr>
      </thead>
      <tbody >
        <tr *ngFor="let commission of commissions" [routerLink]="['/commission',
commission._id]">
```

## Б Қосымшасының жалғасы

```
<td class="collection-item ">{{ commission.fullname }}</td>
<td class="collection-item ">{{ commission.email }}</td>
<td class="collection-item ">{{ commission.phone }}</td>
<td class="collection-item ">{{ commission.position }}</td>
</tr>
</tbody>
</table>
</div>
</div>
<ng-template #empty>
  <div class="center">
    У вас нет ни одного членов комиссии
  </div>
</ng-template>
<ng-template #loader>
<app-loader></app-loader>
</ng-template>
```

### 4. Кадрлар бөлімінің атынан үміткерге хабарлама жіберу мәтіні

```
app.post("/api/sendmail", (req, res)=>{
  const {body} = req;
  const user = {...body}

  sendMail(user, info => {
    console.log(`Ok`);
    res.send(info);
  });
})

async function sendMail(user, callback) {
  let transporter = nodemailer.createTransport({
    service:'gmail.com',
    auth: {
      user: process.env.EMAIL,
      pass: process.env.PASSWORD
    }
  });
});

let mailOptions = {
  from: 'satbayevuniversity.hr@gmail.com',
  to: user.email,
```

```
subject: "Отдел кадров Satbayev University",  
html: `

## ${user.text}</h2>` };


```

```
let info = await transporter.sendMail(mailOptions);
```

```
callback(info);  
}
```

## 5. Өтініш жіберу мәтіні

```
onSubmit(){  
  let obs$  
  let empty = "  
  
  obs$ = this.candidateService.create(  
    this.form.value.name,  
    this.form.value.surname,  
    this.form.value.email,  
    this.form.value.phone,  
    this.form.value.position,  
    this.form.value.faculty,  
    this.pdf)  
  
  obs$.subscribe(  
    candidate => {  
      this.candidate = candidate  
      this.modal.open()  
      this.form.reset()  
    }, error => {  
      this.modal.close()  
      MaterialService.toast('Заполните форму')  
    }  
  )  
}
```

## 6. Кандидаттарды фильтрлеу мәтіні

```
export class CandidateFilterComponent implements OnDestroy, AfterViewInit {  
  @Output() onFilter = new EventEmitter<Filter>()  
  @ViewChild('start') startRef: ElementRef  
  @ViewChild('end') endRef: ElementRef
```



```
start: MaterialDatepicker
end: MaterialDatepicker
name: string
surname: string
isValid = true
constructor() { }
ngOnDestroy(){
    this.start.destroy()
    this.end.destroy()
}
ngAfterViewInit(){
this.start = MaterialService.initDatepicker(this.startRef, this.validate.bind(this))
this.end = MaterialService.initDatepicker(this.endRef, this.validate.bind(this))
}
validate(){
    if(!this.start.date || !this.end.date){
        this.isValid = true
        return
    }
    this.isValid = this.start.date < this.end.date
}
submitFilter(){
    const filter: Filter = {}
    if(this.name){
        filter.name = this.name
    }
    if(this.surname){
        filter.surname = this.surname
    }
    if(this.start.date){
        filter.start = this.start.date
    }
    if(this.end.date){
        filter.end = this.end.date
    }
    this.onFilter.emit(filter)}}}
```

#### 7. Дауыс беру нәтижесінің мәтіні

```
<div class="can">
  <canvas      baseChart      [chartType]="chartType"      [labels]="chartLabels"
  [data]="chartData">
```

```
</canvas>
</div>
<div class="sss">
  <div class="stats">
    <div class="stat">
      <div class="header">
        <p>Количество голосов</p>
      </div>
      <div class="body"><h4>{{ totalVote }}</h4></div>
    </div>
    <div class="stat">
      <div class="header">
        <p>Достоин</p>
      </div>
      <div class="body">
        <h4>{{ posVotePercent || 0 }}</h4>
        <span class="percent">%</span>
      </div>
    </div>
    <div class="stat">
      <div class="header">
        <p>Недостоин</p>
      </div>
      <div class="body">
        <h4>{{ negVotePercent || 0 }}</h4>
        <span class="percent">%</span>
      </div>
    </div>
  </div>
</div>
<div>
</div>
</div>
<app-stats></app-stats>
```

Форматы	Зонасы	Поз.-сы.	Белгілеулер			Аталулары	Саны	Қорытынды
Өлш	Бет	Құжат №	қолы	күні	Дипломдық жоба			Бет
								43