

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

Қ.И. Сәтбаев атындағы Қазақ ұлттық техникалық зерттеу университеті

Кибернетика және ақпараттық технологиялар институты

Киберқауіпсіздік, ақпаратты өңдеу және сақтау кафедрасы

Сатыбалдин Ербол Дәуренұлы

«Автосервис және автокөлікті жөндеу» WEB-қосымшасын әзірлеу

## **ДИПЛОМДЫҚ ЖҰМЫС**

5B070300 – «Ақпараттық жүйелер» мамандығы

Алматы 2021

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

Қ.И. Сәтбаев атындағы қазақ ұлттық техникалық зерттеу университеті


Кибернетика және ақпараттық технологиялар институты

Киберқауіпсіздік, ақпаратты өңдеу және сақтау кафедрасы

**ҚОРҒАУҒА ЖІБЕРІЛДІ**

КАӨЖС кафедрасы меңгерушісі

техн. ғыл. канд., асс. профессор

 Н.А. Сейлова

«31» 05 2021 ж.

## ДИПЛОМДЫҚ ЖҰМЫС

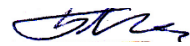
Тақырыбы: «Автосервис және автокөлікті жөндеу» WEB-қосымшасын  
әзірлеу

5B070300- Ақпараттық жүйелер мамандығы

Орындаған  Сатыбалдин Е.Д.

Ғылыми жетекші

техн.ғыл.канд, ассоц.профессор

 Б.И.Жумағалиев

« 31 » 05 2021 ж.

Алматы 2021

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

Қ.И. Сәтбаев атындағы Қазақ ұлттық техникалық зерттеу университеті

Кибернетика және ақпараттық технологиялар институты

Киберқауіпсіздік, ақпаратты өңдеу және сақтау кафедрасы

5B070300 – Ақпараттық жүйелер

**БЕКІТЕМІН**

КАӨЖС кафедрасы меңгерушісі

техн. ғыл. канд., асс. профессор

 Н.А. Сейлова

«31» 05 2021 ж.

**Дипломдық жұмысты орындауға  
ТАПСЫРМА**

Білім алушы Сатыбалдин Ербол Дәуренұлы

Тақырыбы: «Автосервис және автокөлікті жөндеу» WEB-қосымшасын әзірлеу.

Университет Ректорының 2020 жылғы «24» қараша №2131-б бұйрығымен бекітілген.

Орындалған жұмыстың өткізу мерзімі «25» мамыр 2021 ж.

Дипломдық жұмыстың бастапқы мәліметтері: Берілген тақырып бойынша әдебиеттерге шолу кезінде жиналған мәліметтер, теориялық материалдар жинау.

Дипломдық жұмыста қарастырылатын мәселелер тізімі:


1. Тақырып аясын талдау және шолу
2. Деректер базасының моделін құру
3. Программалық қамтаманы құру.

Графикалық материалдардың тізімі (міндетті түрде қажет сызбалар көрсетілген): жұмыстың 17 слайдтан тұратын презентациясы көрсетіледі. Ұсынылған негізгі әдебиет 6 кітаптан тұрады.


Дипломдық жұмысты даярлау  
**КЕСТЕСІ**

Бөлім атауы, қарастырылатын мәселелер тізімі	Ғылыми жетекші мен кеңесшілерге көрсету мерзімі	Ескерту
Тақырып аясын талдау және шолу	18.01.2021 – 09.02.2021	
Деректер базасының моделін құру	09.02.2021 – 21.03.2021	
Программалық қамтаманы құру	21.03.2021 – 29.04.2021	

Дипломдық жұмыс бөлімдерінің кеңесшілері мен норма бақылаушының аяқталған жұмысқа қойған  
**қолтаңбалары**

Бөлімдердің атауы	Кеңесшілер (аты-жөні, тегі, ғылыми дәрежесі, атағы)	Қол қойылған мерзімі	Қолы
БҚ жасау			
Норма бақылаушы	М.А.Кабдуллин, тьютор	21.05.2021	

Ғылыми жетекшісі  Б.И.Жумағалиев

Тапсырманы орындауға қабылдаған білім алушы  Сатыбалдин Е.Д.

Күні «24» 11. 2020 ж.

## АҢДАТПА

Бұл дипломдық жұмыстың мақсаты - автокөлік иелеріне диагностикаға жазылуды қамтамасыз ететін және веб-қосымша арқылы қажетті авто бөлшектерді сатып алатын «Автосервис және автокөлікті жөндеу» веб-қосымшасын әзірлеу. Жұмыс барысында пәндік салаға талдау жасалып, жұмыстың орындалуы жоспары құрылды. Веб-қосымшаланы әзірлеу үш бөлікке бөлінді: сервер, клиент және мәліметтер базасын жобалау. Қосымшаның сервер жағы RESTFull әрі архитектурасын және .NET CORE, Entity Framework технологияларын қолдана отырып жасалған. Клиенттік жағы React, HTML, CSS және JavaScript қолдана отырып жасалған. Мәліметтер базасын жобалау UML диаграммаларын қолдану арқылы жүзеге асырылды.

Қосымша практикалық қолдануға арналған.

## АННОТАЦИЯ

Целью данной дипломной работы является разработка веб-приложения «Автосервис и авторемонт», которое обеспечит автовладельцам возможность записаться на диагностику и приобретать необходимые автозапчасти через веб-приложение. В процессе работы был проведён анализ предметной области и был разработан план выполнения работы. Разработка веб приложения состоит из трех частей: серверная, клиентская и проектирование базы данных. Серверная часть приложения была разработана с применением архитектуру RESTFull api и технологиями .NET CORE, Entity Framework. Клиентская сторона с помощью React, HTML, CSS и JavaScript. Проектирование базы данных было осуществлено с использованием UML диаграмм.

Приложение предназначено для практического использования.

## ANNOTATION

The purpose of this thesis is to develop a web application "Car Service and Auto Repair", which will provide car owners with an appointment for diagnostics and purchase the necessary auto parts through the web application. In the course of the work, the analysis of the subject area was carried out and the performance of the work was developed. Web application development was divided into three parts: server and client and database design. The server side of the application was developed using the RESTFull api architecture and .NET CORE, Entity Framework technologies. Client side with React, HTML, CSS and JavaScript. Database design was carried out using UML diagrams.

The application is intended for practical use.

## МАЗМҰНЫ

КІРІСПЕ .....	9
1. Пәндік саланы талдау .....	10
1.1 Бизнес-процестің сипаттамасы .....	10
1.2 Концептуалды қолдану үлгісі .....	12
1.3 Есептің қойылымы .....	15
2 «Автосервис және автокөлікті жөндеу» WEB-қосымшасын жобалау .....	16
2.1 Веб-қосымшаның архитектурасы .....	17
2.2 Мәліметтер базасын жобалау .....	18
2.3 Деректер моделі .....	19
2.4 Мәліметтер базасының сипаттамасы .....	22
3 Программалық қамтаманы құру .....	28
3.1 Программалау тілін таңдауды негіздеу .....	28
3.2 Бағдарламаны әзірлеу ортасын таңдау .....	28
3.3 WEB-қосымшаға шолу .....	29
ҚОРЫТЫНДЫ .....	36
ПАЙДАЛАНЫЛҒАН ӘДЕБИЕТТЕР ТІЗІМІ .....	37
А қосымшасы .....	38
Б қосымшасы .....	42
С қосымшасы .....	47



## КІРІСПЕ

Қазіргі уақытта өмірді интернет-технологиясыз елестету мүмкін емес, оны пайдаланушылар әр түрлі деңгейдегі адамдар, жасы, жынысы, мүдделері және т.б. сондықтан бүгінгі таңда өзекті тақырып-бұл компаниялар коммерциялық және кәсіпорынды басқару мақсатында пайдаланатын веб-сайттарды дамыту.

Адамның күнделікті өміріне ғаламдық желінің әсер ету дәрежесін бағалау қиын. Интернет барлық қызмет салаларына еніп кетті. Біз Интернетті тауарларды таңдау және сатып алу, авиабилеттерді іздеу және сатып алу, қонақүйлерді брондау, баратын жерге дейін ең қысқа жолды табу, дәрігердің қабылдауына жазылу үшін қолдануға дағдыланғанбыз. Интернет қызметтері бізге ауа райы туралы, кептелістер туралы хабарлайды, әуежайға жету үшін үйден қашан шығатындығын айтады.

Бүгінгі таңда Интернет тек компаниялар мен қызметтер туралы ақпаратты орналастыруға арналған кеңістік емес. Бүгінде Интернет - әлеуметтік желілер, жедел хабар алмасу қызметтері, интернет-дүкендер, аудио-видео хостинг, геоақпараттық жүйелер, онлайн-банктер, анықтамалық жүйелер, білім беру порталдары, шоттарды төлеудің автоматтандырылған жүйелері және жол қозғалысына айыппұлдар, коммуналдық төлемдерді бақылау және балаларыңыздың виртуалды журналдары.

Автосервис және көлік жөндейтін компаниялар үшін веб-қосымшаны құру көптеген себептерге байланысты кезек күттірмейтін міндет. Оларды атап атап өтсек:

1. Сайттың келушілеріне, кәсіпорынның клиенттері үшін тиісті ақпаратты жариялау;
2. Адам күшің үнемдеу;
3. Аудиторияны, яғни дәл осы өнім немесе қызмет түріне мұқтаж адамдарды қызықтырып тарту;
4. Кәсіпорынды басқарудың кейбір функцияларын қамтамасыз ету.

Бұл дипломдық жұмыстың мақсаты автокөліктерге қызмет көрсететін компанияның веб-қосымшасын әзірлеу.

## **1. Пәндік саланы талдау**

### **1.1 Бизнес-процестің сипаттамасы**

Автосервис кәсіпорнының негізгі бизнес-бағыты автокөлікті жөндеу және қызмет көрсету бойынша қызметтер көрсету болып табылады.

Автосервисте ұйымдастырылатын сервистік қызмет көрсетудің негізгі мақсаты: автомобильдерге сервистік қызмет көрсетудің заманауи жүйесін толыққанды қамтамасыз ету.

Автосервис кәсіпорнының мақсатына қол жеткізу үшін негізгі міндет қызмет көрсетудің тиісті деңгейін қамтамасыз етуге дайын жоғары білікті кадр персоналын іріктеу болып табылады.

Қазіргі заманғы автосервис кәсіпорнының қызметкерлері мыналармен қамтамасыз етілуі керек:

- кәсіпорын директоры;
- бухгалтерлік персоналмен;
- қосалқы басқарушы персонал (заңгерлер, маркетингтар);
- автожөндеу кәсіпорнының бастығы;
- әкімші.

автомобиль сервисінің әр түрлі салаларының қабылдаушы шеберлері (автокөлік жуу, іштен жанатын қозғалтқыштарды жөндеу аймағы, автомобиль диагностикасы, электр жабдықтарын жөндеу және монтаждау алаңы, автокөліктің қосымша жабдықтарын орнату аймағы)

Жөндеу және техникалық қызмет көрсету персоналы (автокөлік жуушылар, әр түрлі бағыттағы автомобиль механиктері, автоэлектриктер, диагностиктер)

Кәсіпорын басшысы (директор), бухгалтерия, заң бөлімі және кәсіпорынның маркетинг бөлімі кәсіпорынның тыныс-тіршілігін қамтамасыз етуге және кәсіпорынның тыныс-тіршілігін қамтамасыз ететін шарттық қатынастар мен қатынастар режимінде басқа кәсіпорындармен өзара іс-қимылды келісуге бағытталған жалпы іс-шараларды жүзеге асыра отырып, кәсіпорынның клиентімен тікелей байланысқа қатыспайды.

Сервистік қызмет көрсетуді немесе жөндеуді қажет ететін автокөлік иесінің тікелей байланысы автосервис әкімшісімен кездесуден басталады.

Іскери этикет қағидаларын міндетті түрде сақтау шартымен (сыпайылық, клиентке деген шынайы қызығушылық және т.б.), әкімші ұсынған қызметкер клиенттің көлігімен байланысты мәселелерді (немесе тілектерді) анықтайды.

Жүгінудің проблемалық аспектілерін анықтағаннан кейін әкімші клиентке ақаулықтарды жою бойынша жұмыстарды жүргізу мүмкіндігі туралы кеңес береді, қызмет көрсету сапасын жақсартуға бағытталған қосымша қызметтерді ұсынады (қажет болған жағдайда) (қосымша қызметтерді ұсынған кезде әкімші

клиенттің жүгінуі себебіне қатысы жоқ жұмыстарды немесе клиентке қызмет көрсету сапасын жақсартпайтын қызметтерді дұрыс таңдамауға тиіс).

Қажетті жұмыстардың тізбесі анықталғаннан және келісілгеннен кейін әкімші клиент ұсынған жөндеуге берілетін автомобильді пайдалануға және оған қызмет көрсетуге өкілеттіктерінің болуы туралы құжаттардың негізінде тапсырыс-наряд не қызмет көрсету туралы шарт ресімдейді. Әкімші мен тапсырыс беруші жөндеу жұмыстарының құнын, жұмыстың болжамды аяқталуын, сметалық құнын және қызмет көрсетуге қажетті қосалқы бөлшектер мен шығын материалдарын сатып алу тәртібін келіседі.

Әкімшінің одан арғы іс-әрекеттері көлік құралын қабылдаушы-шеберге беру және жөнделіп жатқан көлік құралына қызмет көрсету кезеңінде клиенттің жайлы күтуі үшін жағдай жасау болып табылады.

Жөндеу жүргізу кезеңінде клиент жүргізілетін жұмыстардың тиімділігі мен толықтығын тексеру мақсатында жөнделетін көлік құралына еркін қол жеткізе алуы тиіс.

Шебер-қабылдаушы әкімшіден кілттері мен ілеспе құжаттары бар қызмет көрсетуге қабылданған көлік құралын алып, автомобильді тікелей жөндеу учаскесіне беруді қамтамасыз етеді және оны жөндеу жұмыстарының басталу уақытын белгілей отырып, қызмет көрсетуді жүзеге асыратын маманға береді.

Мамандандырылған персонал көлік құралын қабылдаушы шеберден қабылдайды және ақауларды тікелей жоюға не тапсырыс беруші клиентпен келісілген технологиялық операцияларды жүргізуге кіріседі. Көлік құралымен тікелей келісілген жұмыстарды жүргізу кезінде маман бұрын алған білімдері мен дағдыларын негізге ала отырып, автожөндеу кәсіпорнының директоры мен бастығы тікелей бекіткен лауазымдық нұсқаулықтарды міндетті түрде сақтай отырып, технологиялық операцияларды жүзеге асырады. Лауазымдық міндеттердің көлемі мен тізбесі қызметкердің мамандандырылған бағытына байланысты өзгереді (автомобиль механикі-жаяу жүргінші, автокөлік механик-ойлаушы, автоэлектрик, автокөлік сырлаушы және т.б.).

Бұрын келісілген жөндеу немесе алдын алу жұмыстарын жүргізу аяқталғаннан кейін маман автомобильді қабылдаушы-шеберге береді, ал қабылдаушы-шебер жұмыстың аяқталу уақытын бекітеді.

Орындалған жұмыстың дұрыстығы мен толықтығын көзбен тексереді. Клиент-тапсырыс берушіге әрі қарай беру үшін көлікті алады. Көлік құралын маманнан алып, шебер-қабылдаушы автомобильді шығу алаңына жылжытады және кілттері мен ілеспе құжаттарын әкімшіге береді.

Әкімші төлем құжаттарын түпкілікті ресімдеуді жүргізеді, көлік құралын тексеру және жұмыстардың дұрыс орындалуын алдын ала бағалау үшін клиентке береді, түпкілікті есеп айырысуды жүргізу үшін клиентті кәсіпорынның кассасына ертіп жүреді. Соңғы есептеулерді жасағаннан кейін, әкімші клиентке автокөліктің кілттерін береді және оған орындалған жұмыс үшін кепілдік міндеттемелерді қосымша түсіндіре отырып, кәсіпорын аумағынан кетуге көмектеседі.

## 1.2 Концептуалды қолдану үлгісі

Автосервис қосымшасына кіргенде, ең бірінші бетте қолданушыны қарсы алып, қосымшаның қандай мүмкіндіктер орындай алатыны көрсетілген.

Оң жақта бізде бағдарлама бойынша шарлауға мүмкіндік беретін навигациялық мәзір бар және келесі бөлімдер бар:

- басты бет;
- жеке кабинет;
- көліктер;
- тапсырыс-наряд;
- автоервис шығындары;
- есептер;
- қызметкерлер;
- клиенттер.

Басты бетте веб қосымшадағы негізгі функционал туралы ақпарат орналасқан, оларды атап өтсек:

- клиенттер мен олардың көлік құралдары туралы мәліметтер базасын жүргізу;
- автокөлік қызметтерінің шығындары туралы мәліметтер базасын жүргізу;
- автокөлік бойынша жүргізілген жөндеу жұмыстарының есебін жүргізу;
- автосервис қызметкерлерінің есебін жүргізу;
- атқарылған жұмыстар туралы толық есеп беру.

Жүйеде жұмыс істеу үшін жүйеге алдымен тіркеліп кіру қажет. Жүйеге кіргеннен кейін пайдаланушы бірнеше рөлдердің біріне ие болуы мүмкін. Жүйеде 4 рөл бар. Жүйенің рөлдері:

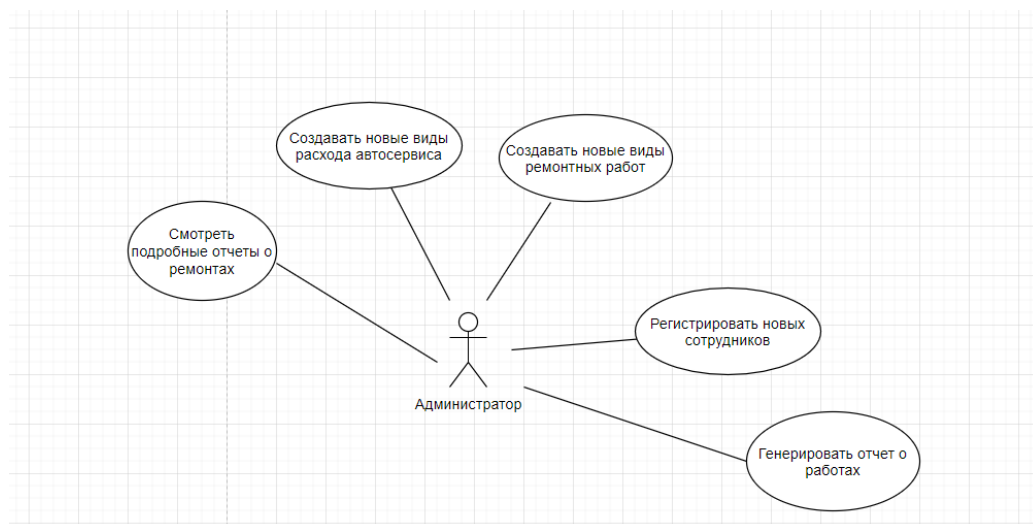
- әкімші;
- пайдаланушы;
- қызметкер.

1.1 кесте – Рөлдерге тапсырмалардың қол жетімділігі.

Тапсырма\Рөл	Әкімші	Пайдаланушы	Қызметкер
Басты бет	Бар	Бар	Бар
Жеке кабинет	Бар	Бар	Бар
Көліктер	Бар	Бар	Бар
Тапсырыс-наряд	Бар	Жоқ	Бар
Автоервис шығындары	Бар	Жоқ	Жоқ

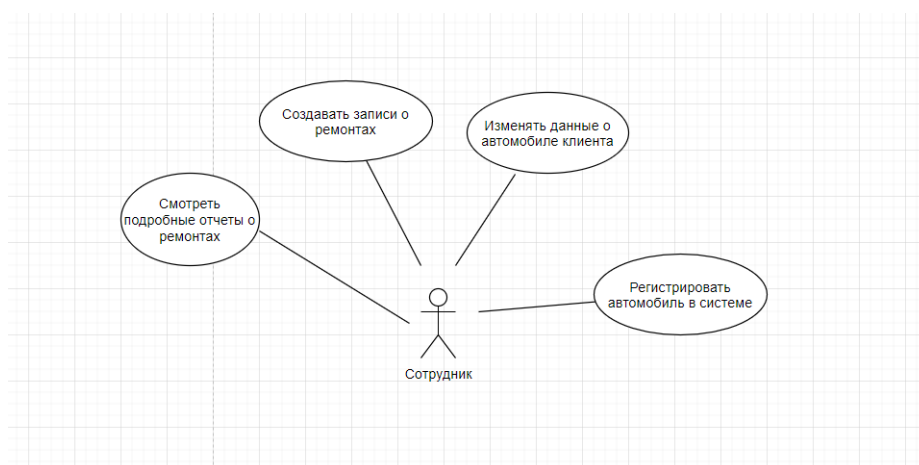
Есептер	Бар	Жоқ	Жоқ
Қызметкерлер	Бар	Жоқ	Жоқ
Клиенттер	Бар	Жоқ	Бар

Әкімші-бұл жүйенің жұмысына жауап беретін пайдаланушы. Оның іс-әрекет диаграммасы келесі суретте көрсетілген (1.1-сурет).



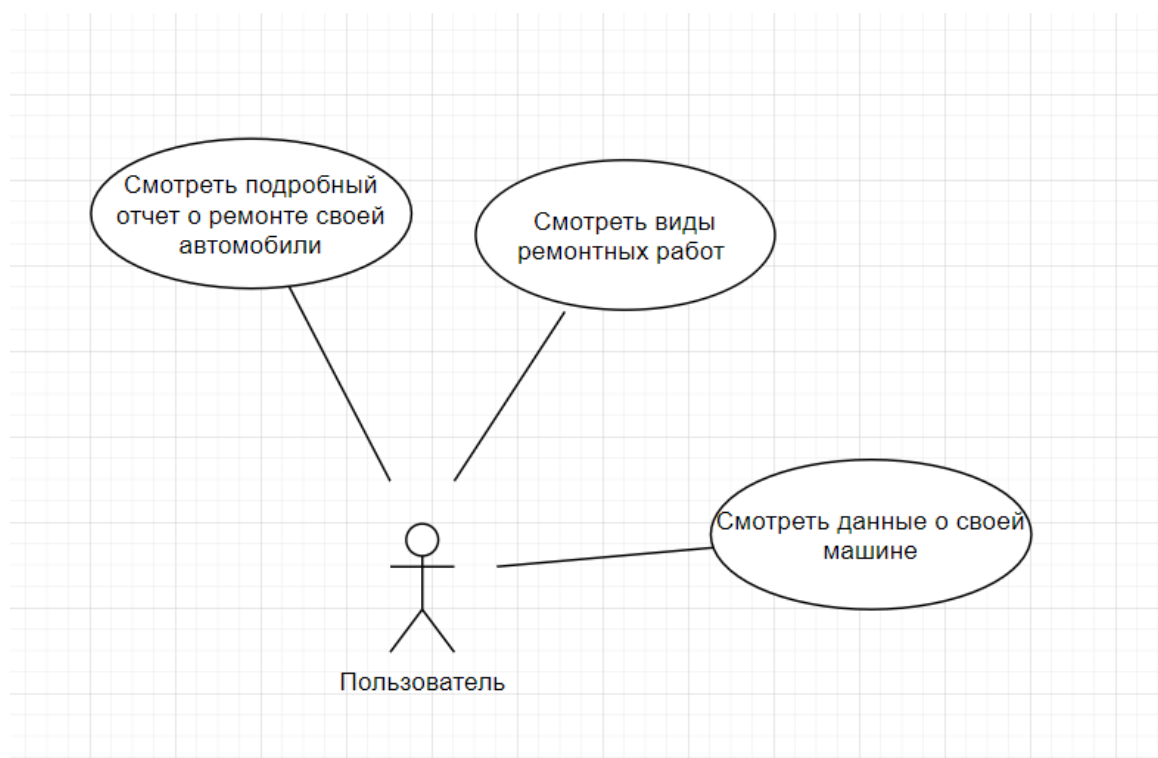
1.1-сурет – Әкімші рөлінің іс-әрекет диаграммасы

Қызметкер - бұл машиналарды тікелей жөндеумен айналысатын пайдаланушы. Оның іс-әрекет диаграммасы келесі суретте көрсетілген (1.2-сурет).



1.2-сурет – Қызметкер рөлінің іс-әрекет диаграммасы

Пайдаланушы - бұл көлікті жөндеуге тапсырған қарапайым пайдаланушы. Оның іс-әрекет диаграммасы келесі суретте көрсетілген (1.3-сурет).



1.3-сурет – Пайдаланушы рөлінің іс-әрекет диаграммасы

Осылайша, жүйеде әрбір пайдаланушының өз рөлі мен міндеттері бар. Сондай-ақ, кез-келген уақытта сіз белгілі бір рөлдің функционалдығын өзгерте аласыз, осылайша жүйеде жұмыс істеу міндеттерін өзгерте аласыз.

### 1.3 Есептің қойылымы

Веб-қосымшаның бағдарламалық модульдерінің өзара әрекеттесуінің ыңғайлы және қарапайым интерфейсін қамтамасыз ету үшін модульдер арасында функционалдылықты бөлуді қамтамасыз ету қажет. Мәселені шешу пайдаланушының өзара әрекеттесуі үшін интерактивті интерфейсін дамытуды қажет ететіндіктен, веб-қосымшаның клиенттік бөлігі интерфейсін көрсету үшін қуатты құралдарды қажет етеді. Тиісінше, осы функционалдылықты бір модульде оқшаулау үшін барлық әрекеттерді клиент қосымшасында интерфейс дисплейін қалыптастыру үшін шоғырландыру қажет. Шешім ретінде бір парақ қосымшасын құрудың тәсілі таңдалды. Бір парақты қосымша клиенттің қолданбалы модуліне логиканың көп бөлігін алып тастауды көздейді: көрсету үшін деректерді өңдеу, көрсету, пайдаланушы оқиғаларын өңдеу, серверге деректерді жіберу.

Бір беттік қосымша логиканың көп бөлігін клиенттік қосымшаның модуліне шығаруды қамтиды: деректерді көрсету, көрсету, пайдаланушы оқиғаларын өңдеу, деректерді серверге жіберу. Осылайша, клиент модуліне арналған интерфейс-бұл шикі деректерді дисплейсіз қабылдау және жіберу әдістері.

Клиенттік қосымшаның мәселелерін шешу үшін келесі технологиялар таңдалды:

- пайдаланушы интерфейсін көрсетумен жұмыс істеуге арналған React кітапханасы[1];
- серверге сұраныс жасау үшін Axios кітапханасы;
- веб-қосымшаның дизайнын жасау үшін Antd кітапханасы.

Бір беттік веб-қосымшаның тәсілін жүзеге асырудағы серверлік қосымшаның негізгі міндеттері:

- мәліметтер базасымен өзара әрекеттесі;
- клиенттік қосымшаның сұраныстарын өңдеу және қосымшаның іскери логикасын орындау;
- пайдаланушының бірінші сұранысына арналған веб-парақты қалыптастыру.

Серверлік қосымшаның мәселелерін шешу үшін келесі технологиялар таңдалды:

- .NET - веб-серверді іске асыруға, пайдаланушы сұрауларын бағыттауға, бағдарламаның логикасын жеке ішкі жүйелерге бөлуге мүмкіндік беретін платформа;
- Entity Framework Core - веб-серверде мәліметтер базасымен жұмыс істеуге көмектесетін технология;
- мәліметтер қорын қосымшаға қосу үшін Npgsql кітапханасын

пайдалану.

Заманауи талаптарға сай веб-қосымшаны әзірлеу үшін, дипломдық жұмыста келесі талаптарды орындау қажет:

- веб-қосымшаның құрылымын жасау;
- веб-қосымшаның клиенттік модулін құру;
- веб-қосымшаның серверлік модулін құру;
- мәліметтер базасының модулін құру;
- клиент пен сервер арасындағы мәліметтер алмасу хаттамасын ұйымдастыру;
- клиенттің модулі мен сервер бөліктері арасындағы өзара әрекеттесуге арналған интерфейсті жобалау;
- сервер модулі мен мәліметтер базасы арасындағы өзара әрекеттесу интерфейсін жобалау.

Дипломдық жұмыстың орындау нәтижесі автокөлік жөндеу саласында өз қызметтерін ұсынатын автосервис кәсіпорнына арналған веб-қосымша болуы тиіс.



## 2 «Автосервис және автокөлікті жөндеу» WEB-қосымшасын жобалау

### 2.1 Веб-қосымшаның архитектурасы

Веб-қосымша клиенттік және серверлік бөліктен тұрады, осылайша "клиент - сервер" желілік архитектурасын қалпына келтіреді. Клиент бөлігі пайдаланушы интерфейсіне жүзеге асырады, серверге сұраулар жасайды және одан жауаптарды өңдейді. Сервер пайдаланушының сұраныстарын қабылдайды, қажетті есептеулерді орындайды, содан кейін веб-бетті қалыптастырады және оны HTTP немесе HTTPS протоколын қолдана отырып, желі арқылы клиентке жібереді.

REST (Representational state transfer) – бұл веб-қызметтерді құру үшін қолданылатын таратылған жүйелік архитектурасының стилі[2].

Rest архитектуралық стилінде клиенттің орындалуы мен сервердің орындалуы бір-бірін білместен бір-біріне тәуелсіз жүзеге асырылуы мүмкін. Бұл клиенттік кодты кез-келген уақытта сервердің жұмысына әсер етпестен өзгертуге болатындығын және серверлік кодты клиенттің жұмысына әсер етпестен өзгертуге болатындығын білдіреді.

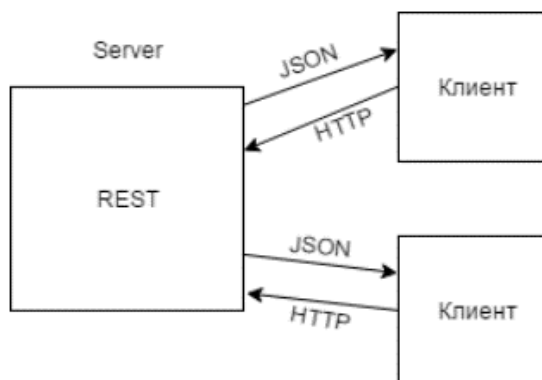
REST жүйесіндегі ресурстармен өзара әрекеттесу үшін біз сұраныстарда қолданатын 4 негізгі HTTP командалары бар:

- GET - белгілі бір ресурстарды (идентификатор бойынша) немесе ресурстар жиынтығын алу;
- POST - жаңа ресурс жасау;
- PUT - белгілі бір ресурстарды жаңарту (идентификатор бойынша);
- DELETE - идентификатор бойынша белгілі бір ресурстарды жою.

REST клиенттен сервердегі деректерді алу немесе өзгерту үшін серверге сұраныс жасауды талап етеді.

Сұраныс әдетте мыналардан тұрады:

- операцияның қандай түрін жасау керектігін анықтайтын HTTP етістігі;
- клиентке сұрау туралы ақпаратты жіберуге мүмкіндік беретін тақырып;
- ресурсқа апаратын URL жол;
- деректерді қамтитын қосымша хабарлама денесі.



## 2.1 Сурет – REST серверінің жұмыс схемасы

Бұл дипломдық жұмыста мен REST архитектурасын қолдануды шештім, өйткені архитектураның өзі өте икемді және жүйені жетілдіруге және масштабтауға мүмкіндік береді . Жобада HTTP протоколындағы PUT, POST, DELETE, GET әдістері қолданылады, ал деректермен алмасу JSON форматын қолдану арқылы жүзеге асырылады.

## 2.2 Мәліметтер базасын жобалау

ДББЖ ретінде PostgreSQL таңдалды[3]. Тестілеу және қосымшаны әзірлеу жергілікті компьютерде жүргізілгендіктен, pgAdmin 4 қосымшасын жүктеу және орнату қажет болды. ДББЖ орнатудың стандартты кезеңдерінен кейін ол жұмысқа дайын болды.

Қосымшаны тұжырымдамалық әзірлеу кезеңінде болашақ мәліметтер базасының моделі жасалды, оған кестелер мен бағдарламаның функционалдығын кеңейту мүмкіндігі туралы мәліметтер енгізілді.

Мәліметтер базасын мен кестелерді визуалды редакторда да, PostgreSQL сұрауларының көмегімен де жасауға болады. Жұмыс барысында сұраныс тілі қолданылды, өйткені сервер үшін және сервердің қосымшаның жұмыс істеуі үшін сұрау тілін қолдану керек.

Мысал ретінде, PostgreSQL көмегімен жасалған кестенің бір нұсқасы. “Clients” кестесін құру үшін, келесі сұраныс қолданылды.

```
CREATE TABLE "Clients" (  
    Id serial PRIMARY KEY,  
    LastName varchar(250) NOT NULL,  
    FirstName varchar(250) NOT NULL,  
    UserId integer NOT NULL,  
    MiddleName varchar(250),  
    Phone varchar(20) NOT NULL,  
    Email varchar(100) NOT NULL,  
    Login varchar(50) NOT NULL,  
    Password varchar(250) NOT NULL
```

Қалған дерекқор кестелерін, кестелердің өрістерін және деректер түрлерінің аттарын өзгерту арқылы ұқсастықпен жасауға болады.

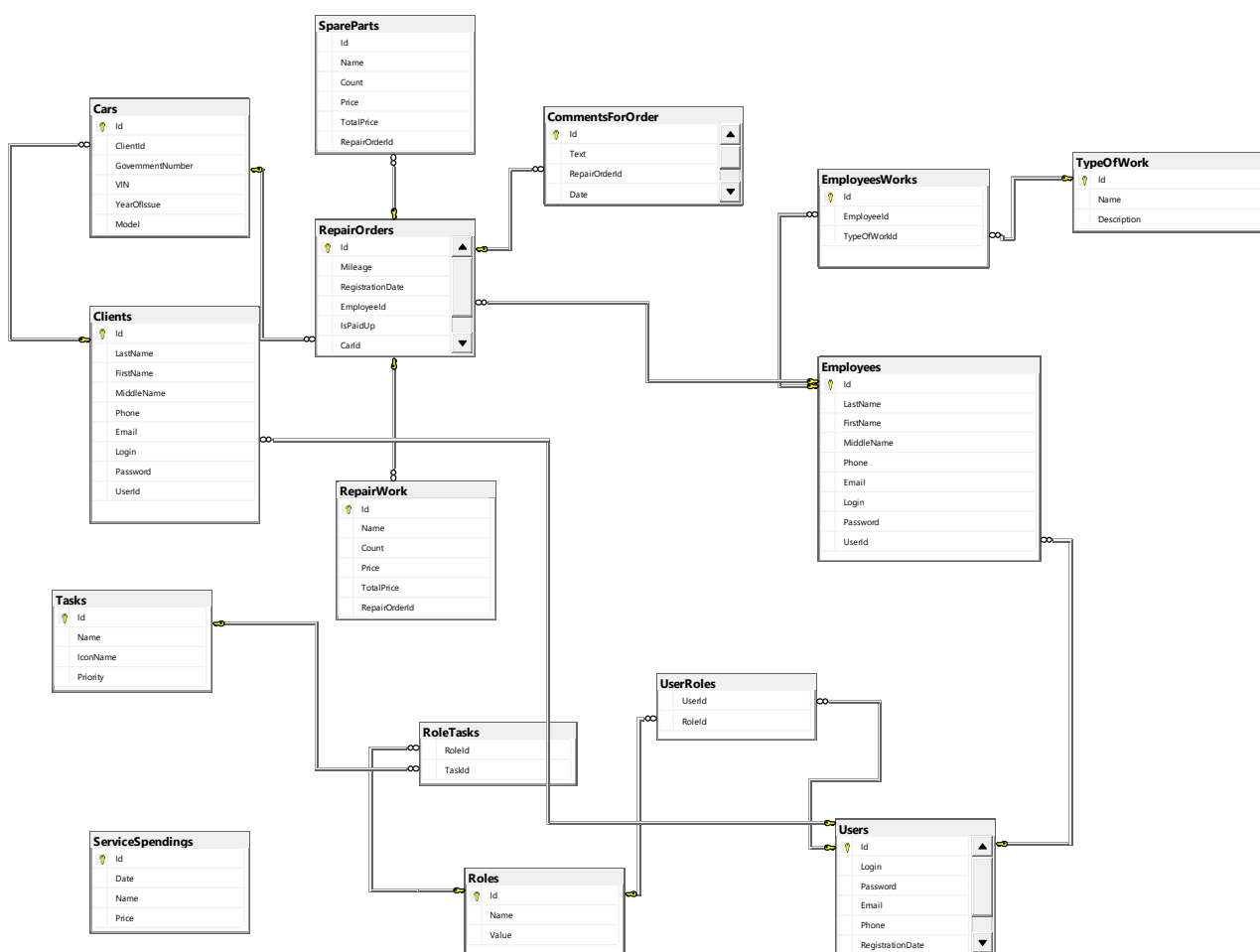
Pgadmin 4 интерфейсінде мәліметтер базасының құрылымын көре аламыз, сонымен мәліметтер базасымен әр түрлі операцияларды жасай аламыз.

	Data Output	Explain	Messages	Notifications
	Id [PK] integer	LastName character varying (200)	FirstName character varying (200)	MiddleName character varying (200)
1	1	Иванов	Иван	Иванович
	Phone character varying (20)	Email character varying (50)	Login character varying (100)	
	220833	ivan@mail.ru	ivan	

2.2 Сурет – PostgreSQL pgAdmin 4 интерфейсі

### 2.3 Деректер моделі

Деректер базасының схемасы 2.2 суретте көрсетілген.



2.2-сурет – Деректер базасының схемасы

Автосервис кәсіпорнының пәндік аймағында келесі объектілерді , олардың атрибуттары мен байланыстарын ажыратуға болады:

1. Көліктер
  - Идентификатор

- Мемлекеттік нөмір
  - VIN
  - Машинаның шығарылған күні
  - Модель атауы
2. Клиенттер
- Идентификатор
  - Тегі
  - Аты
  - Әкесінің аты
  - Логин
  - Күпия сөз
  - Пошталық адрес
  - Телефон нөмері
  - Жүйеде тіркелген күні
3. Қызметкерлер
- Идентификатор
  - Тегі
  - Аты
  - Әкесінің аты
  - Логин
  - Күпия сөз
  - Пошталық адрес
  - Телефон нөмері
  - Жүйеде тіркелген күні
4. Жұмыс түрлері
- Идентификатор
  - Аты
  - Сипаттамасы
5. Жұмыскердің жұмыс түрлері
- Идентификатор
  - Жұмыскердің идентификаторы
  - Жұмыс түрінің идентификаторы
6. Рөлдер
- Идентификатор
  - Атауы
  - Мәні
7. Тапсырмалар
- Идентификатор
  - Атауы
8. Рөлдің тапсырмалары
- Рөлдің идентификаторы
  - Тапсырманың идентификаторы
9. Пайдаланушылардың рөлдері

- Пайдаланушының идентификаторы
  - Рөлдің идентификаторы
10. Тапсырыс-нарядтар
- Идентификатор
  - Көліктің идентификаторы
  - Тіркеу сәтіндегі жүгіріс
  - Тіркеу күні
  - Жұмыскердің идентификаторы
  - Қосалқы бөлшектердің жалпы сомасы
  - Жөндеудің жалпы сомасы
  - Төленді
11. Тапсырыс туралы түсініктемелер
- Идентификатор
  - Пікір мәтіні
  - Тапсырыс-нарядтың идентификаторы
  - Күні
  - Жұмыскердің идентификаторы
12. Қосалқы бөлшектер
- Идентификатор
  - Атауы
  - Саны
  - Бағасы
  - Жалпы сомасы
  - Тапсырыс-нарядтың идентификаторы
13. Жұмыстар
- Идентификатор
  - Атауы
  - Саны
  - Бағасы
  - Жалпы сомасы
  - Тапсырыс-нарядтың идентификаторы
14. Автосервистің шығындары
- Идентификатор
  - Күні
  - Атауы
  - Бағасы

Бизнес-процестерді талдау және пәндік саланы сипаттау негізінде келесі кестелерден тұратын мәліметтер базасының схемасы жасалды:

- Cars – автокөліктер
- Clients - клиенттер
- CommentsForOrder – тапсырыстар үшін түсініктеме
- Employees – жұмыскерлер

- RepairOrders - тапсырыстар
- RepairWork – тапсырыстағы жұмыстар
- RoleTasks – рөлдердің тапсырмалары
- Roles – рөлдер
- ServiceSpending- автосервис шығындары
- SpareParts - тапсырыстағы қосалқы бөлшектер
- Tasks - тапсырмалар
- TypeOfWork – жөндеу жұмыс түрлері
- UserRoles – пайдаланушылардың рөлдері
- Users – пайдаланушылар

## 2.4 Мәліметер базасының сипаттамасы

Деректердің толық сипаттамасы 2.1 - 2.15 кестелерінде келтірілген.

2.1 кесте – Рөлдер тізіміне арналған кесте Roles кестесі

Атауы	Деректер түрі	Ұзындығы	Ескерту
Id	integer	11	PRIMARY_KEY, NOT NULL, AUTO_INCREMENT
Name	character varying	250	NOT NULL
Value	character varying	250	NOT NULL

2.2 кесте – Автомобильдерді жөндеуге арналған тапсырыстарды сақтауға арналған кесте

RepairOrders кестесі

Атауы	Деректер түрі	Ұзындығы	Ескерту
Id	serial	11	PRIMARY_KEY , NOT NULL , AUTO_INCREMENT
Mileage	integer	11	NOT NULL
RegistrationDate	date		NOT NULL
EmployeeId	integer	11	NOT NULL
TheAmountsOf Parts	integer	11	NOT NULL
TheAmountsOf Works	integer	11	NOT NULL
IsPaidUp	boolean		NOT NULL

2.3 кесте – Жалпы пайдаланушы деректерін сақтауға арналған кесте.

Users кестесі

Атауы	Деректер түрі	Ұзындығы	Ескерту
Id	serial	11	PRIMARY_KEY , NOT NULL , AUTO_INCREMENT
Login	character varying	50	NOT NULL
Password	character varying	250	NOT NULL
Email	character varying	75	NOT NULL
Phone	character varying	20	NOT NULL
RegistrationDate	date		NOT NULL

2.4 кесте – Клиенттерді сақтауға арналған кесте.  
Clients кестесі

Атауы	Деректер түрі	Ұзындығы	Ескерту
Id	serial	11	PRIMARY_KEY, NOT NULL, AUTO_INCREMENT
LastName	character varying	250	NOT NULL
FirstName	character varying	250	NOT NULL
MiddleName	character varying	250	
UserId	integer	11	NOT NULL

2.5 кесте – Пайдаланушы рөлдерін сақтауға арналған кесте.  
UserRoles кестесі

Атауы	Деректер түрі	Ұзындығы	Ескерту
UserId	integer	11	PRIMARY_KEY, NOT NULL, AUTO_INCREMENT
RoleId	integer	250	NOT NULL

2.6 кесте – Автосервис қызметкерлерін сақтауға арналған кесте.  
Employees кестесі

Атауы	Деректер түрі	Ұзындығы	Ескерту
Id	serial	11	PRIMARY_KEY, NOT NULL, AUTO_INCREMENT
LastName	character varying	250	NOT NULL
FirstName	character varying	250	NOT NULL
MiddleName	character varying	250	
UserId	integer	11	NOT NULL



2.7 кесте – Тапсырыстардан пікірлерді сақтауға арналған кесте.

CommentsForOrder кестесі

Атауы	Деректер түрі	Ұзындығы	Ескерту
Id	serial	11	PRIMARY_KEY , NOT NULL , AUTO_INCREMENT
Text	character varying	250	NOT NULL
RepairOrderId	character varying	250	NOT NULL
Date	date		NOT NULL
EmployeeId	integer	11	NOT NULL

2.8 кесте – Рөлдік тапсырмаларды сақтауға арналған кесте.

RoleTasks кестесі

Атауы	Деректер түрі	Ұзындығы	Ескерту
RoleId	integer	11	PRIMARY_KEY, NOT NULL, AUTO_INCREMENT
TaskId	integer	11	NOT NULL

2.9 кесте – Қызметкерге арналған жұмыс түрлерін сақтауға арналған кесте.

EmployeesWorks кестесі

Атауы	Деректер түрі	Ұзындығы	Ескерту
Id	integer	11	PRIMARY_KEY, NOT NULL, AUTO_INCREMENT
EmployeeId	integer	11	NOT NULL
TypeOfWorkId	integer	11	NOT NULL

2.10 кесте – Жөндеу жұмыстарының түрлерін сақтауға арналған кесте.  
TypeOfWork кестесі

Атауы	Деректер түрі	Ұзындығы	Ескерту
Id	integer	11	PRIMARY_KEY , NOT NULL , AUTO_INCREMENT
Name	character varying	250	NOT NULL
Description	character varying	250	

2.11 кесте – Жөндеу жұмыстары бойынша ақпаратты сақтауға арналған кесте.

RepairWork кестесі

Атауы	Деректер түрі	Ұзындығы	Ескерту
Id	serial	11	PRIMARY_KEY , NOT NULL , AUTO_INCREMENT
Name	character varying	50	NOT NULL
Price	integer	11	NOT NULL
TotalPrice	integer	11	NOT NULL
RepairOrderId	integer	11	NOT NULL

2.12 кесте – Жөндеу жұмыстары бойынша ақпаратты сақтауға арналған кесте.

ServiceSpendingс кестесі

Атауы	Деректер түрі	Ұзындығы	Ескерту
Id	integer	11	PRIMARY_KEY , NOT NULL , AUTO_INCREMENT
Date	date		NOT NULL
Name	character varying	250	NOT NULL
Price	integer	11	NOT NULL

2.13 кесте – Жөндеуге арналған қосалқы бөлшектер туралы ақпаратты сақтауға арналған кесте.

SpareParts кестесі

Атауы	Деректер түрі	Ұзындығы	Ескерту
Id	serial	11	PRIMARY_KEY, NOT NULL, AUTO_INCREMENT
Name	character varying	50	NOT NULL
Count	integer	11	NOT NULL
Price	integer	11	NOT NULL
TotalPrice	integer	11	NOT NULL
RepairOrderId	integer	11	NOT NULL

2.14 кесте – Рөлдерге арналған тапсырмаларды сақтауға арналған кесте.

Tasks кестесі

Атауы	Деректер түрі	Ұзындығы	Ескерту
Id	integer	11	PRIMARY_KEY, NOT NULL, AUTO_INCREMENT
Name	character varying	250	NOT NULL
Priority	integer	11	

2.15 кесте – Көліктер жайлы ақпараттарды сақтауға арналған кесте.

Cars кестесі

Атауы	Деректер түрі	Ұзындығы	Ескерту
Id	serial	11	PRIMARY_KEY, NOT NULL, AUTO_INCREMENT
ClientId	character varying	50	NOT NULL
Government Number	character varying	250	NOT NULL
VIN	character varying	75	NOT NULL
Model	character varying	50	NOT NULL

## 3 Программалық қамтаманы құру

### 3.1 Программалау тілін таңдауды негіздеу

C#-қарапайым, заманауи және объектіге бағытталған бағдарламалау тілі. Қазіргі уақытта C # бағдарламалау тілі IT саласындағы ең қуатты, жылдам дамып келе жатқан және сұранысқа ие тілдердің бірі болып табылады[4]. Қазіргі уақытта ол әртүрлі қосымшаларды жазады: кішігірім жұмыс үстелі бағдарламаларынан бастап күн сайын миллиондаған пайдаланушыларға қызмет көрсететін үлкен веб-порталдар мен веб-қызметтерге дейін.

C# - бұл C синтаксисі ұқсас тіл және осыған байланысты C++ және Java-ға жақын. Сондықтан, егер осы тілдердің бірімен таныс болсаңыз, онда C# меңгеру оңайырақ болады.

C# объектіге бағытталған және осыған байланысты Java және C++ тілдерінен көп функцияны қабылданды. Мысалы, C# полиморфизмді, мұрагерлікті, операторлардың шамадан тыс жүктелуін, статикалық теруді қолдайды. Объектіге бағытталған тәсіл үлкен, бірақ сонымен бірге икемді, кеңейтілетін және кеңейтілетін қосымшаларды құру мәселелерін шешуге мүмкіндік береді. C# белсенді дамуды жалғастыруда және әр жаңа нұсқада лямбда, динамикалық байланыстыру, асинхронды әдістер және т. б. сияқты қызықты функционалдылықтар пайда болады.

C # артықшылықтарын айта кетсек :

- C# - бұл толықтай объектіге бағытталған тіл, онда тіпті тілге енгізілген типтер де класстармен ұсынылған;

- C# - мұрагерлік және әмбебаптандыру мүмкіндіктері бар қуатты объект тілі;

- C# - бұл танымал бағдарламалау тілдерінің ең жақсы ерекшеліктерін сақтай отырып, C/C++ тілдерінің мұрагері;

- Кросс-платформалық даму-C# көптеген платформаларда бағдарламаларды жасауға өте қолайлы;

- Қоқыс жинау автоматты түрде жүреді – C# - де қажет болған жағдайда қоқысты автоматты түрде жинауға және жоюға арналған ыңғайлы жүйе бар. Бұл бағдарламаны орындау кезінде проблемалардың туындау ықтималдығының төмендеуіне әкеледі.

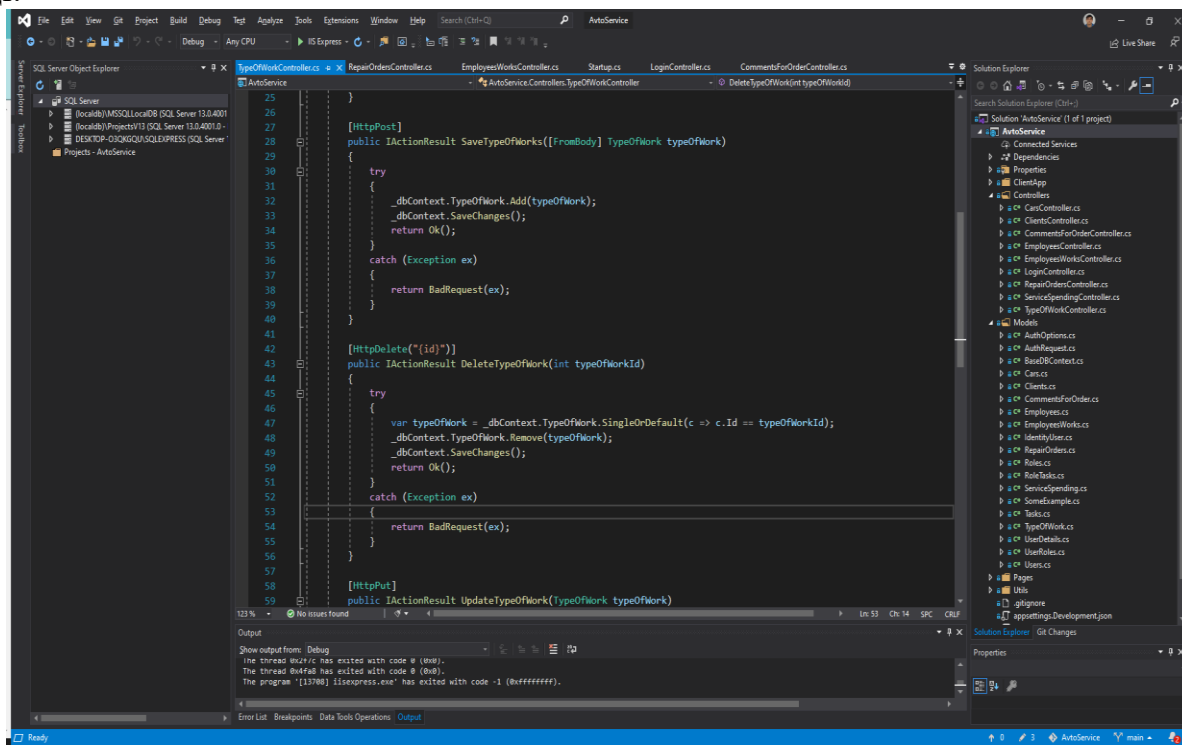
### 3.2 Бағдарламаны әзірлеу ортасын таңдау

Visual Studio 2019 - бұл барлық сұранысқа ие клиенттер мен пайдаланушыларды қанағаттандыратын бай қосымшаларды бағдарламалауға арналған заманауи өнім.

Visual Studio 2019-те HTML, JavaScript, C#, .NET және C ++ тілдеріндегі

қуатты қосымшаларды бағдарламалауға қажетті барлық құралдар бар[5].

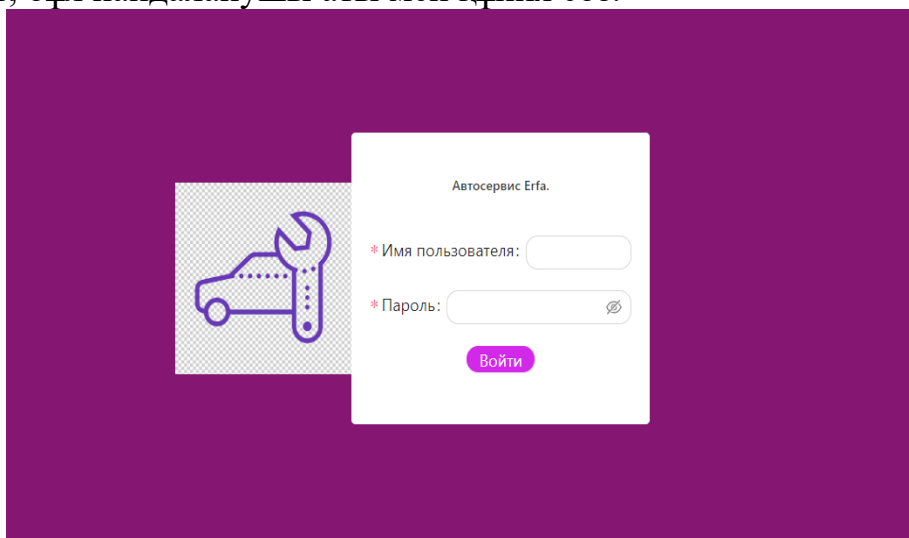
Дипломдық жұмысты Visual Studio 2019 бағдарламасында жұмыс жасауды шештім, өйткені ол тегін және жоба құрылымын оңай басқаруға мүмкіндік береді.



3.1 Сурет – Visual Studio 2019 бағдарламаның интерфейсі

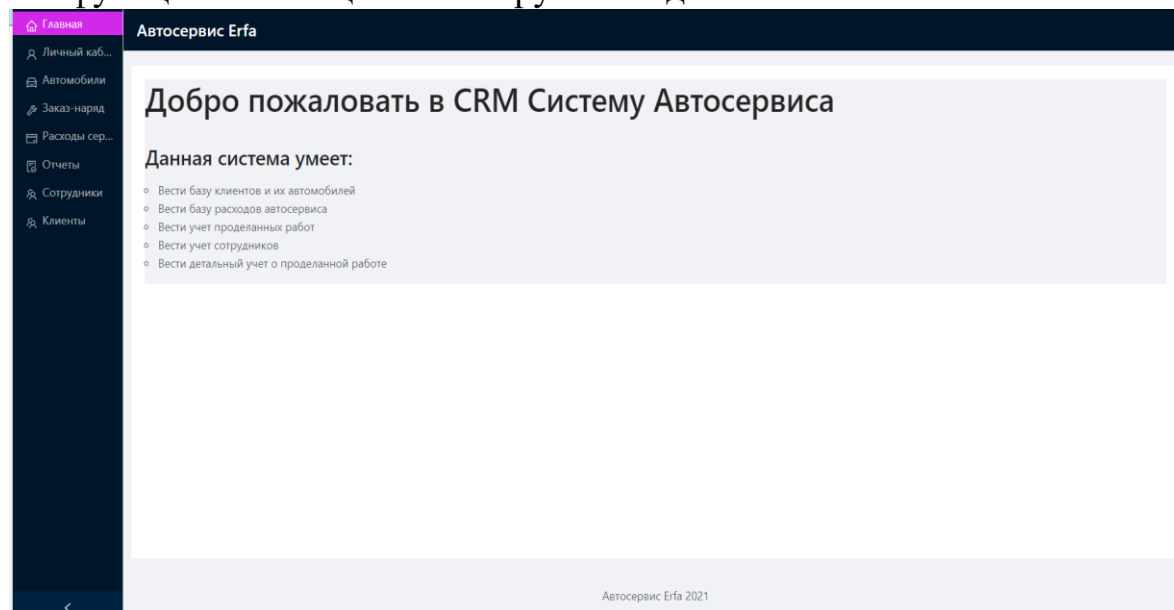
### 3.3 WEB-қосымшаға шолу

Қосымшаға кіргенде, ең алдымен, авторизация парағы көрінеді. Авторизация қажет, өйткені біз қандай қолданушының қосымшада белгілі бір әрекетті жасағанын білуіміз керек. Авторизацияны өту үшін екі өрісті толтыру қажет, бұл пайдаланушы аты мен құпия сөз.



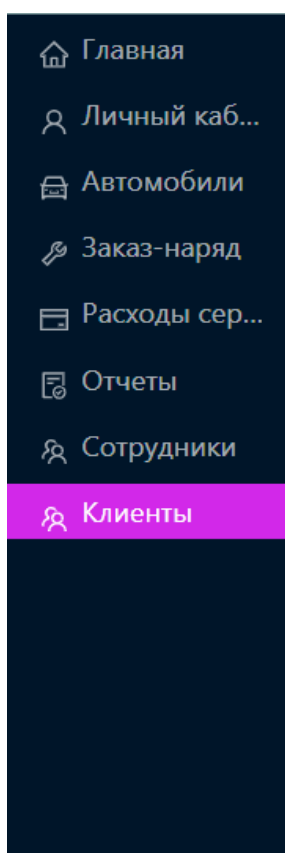
3.1 Сурет – Жүйеге кіру терезесі

Авторизациядан өткеннен кейін бізге қосымшаның басты беті ашылады, онда сайт функционалының тізімін көруге болады.



### 3.2 Сурет – Бастапқы бет

Бағдарламаның сол жағында сіз сайттың навигациялық менюсын көре аламыз. Бұл навигация меню кірген пайдаланушыға қол жетімді әр түрлі парақтарды көрсетеді.



### 3.3 Сурет – Сайттың навигациялық мәзірі

«Клиент» бетінде автокөлік қызметін пайдаланған барлық клиенттердің тізімі бар кесте орналасқан. Бұл парақта клиенттерді құру, жою, редакциялау батырмалары бар.

Фамилия	Имя	Отчество	Номер телефона	Логин пользователя	Почта	Действие	Действие
Satybaldin	Yerbol	Дауренулы	+7 (708) 887-45-30	admin	TestUser@mail.ru	<a href="#">↗</a>	<a href="#">✖</a>
Admin	Admin	Admin	+7 (708) 887-45-30	admin	dasdsada	<a href="#">↗</a>	<a href="#">✖</a>
Какиш	Ерасыл	Ринатулы	+77088874530	kakish	kakish@mail.ru	<a href="#">↗</a>	<a href="#">✖</a>
Еспенбетова	Алия	Темиржановна	+77088874528	alia	alia@mail.ru	<a href="#">↗</a>	<a href="#">✖</a>
Семейханов	Асхат	Сакенович	+7 (705) 843-98-16	ask	ask@mail.ru	<a href="#">↗</a>	<a href="#">✖</a>

### 3.4 Сурет – Клиент парақшасы

Егер сіз жаңа клиент қосқыңыз келсе, ол туралы барлық ақпаратты толтырып, сақтау түймесін басып деректер қорына сақтай аламыз.

Создание клиента

\* Фамилия:

\* Имя:

\* Отчество:

\* Номер телефона:

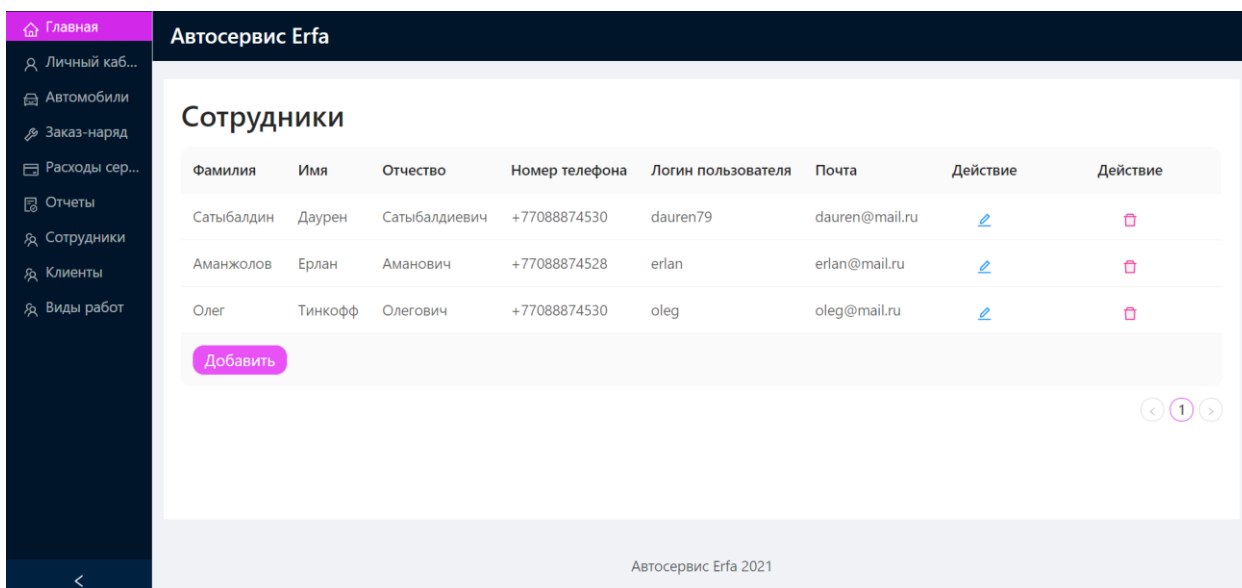
\* Логин пользователя:

\* Пароль:

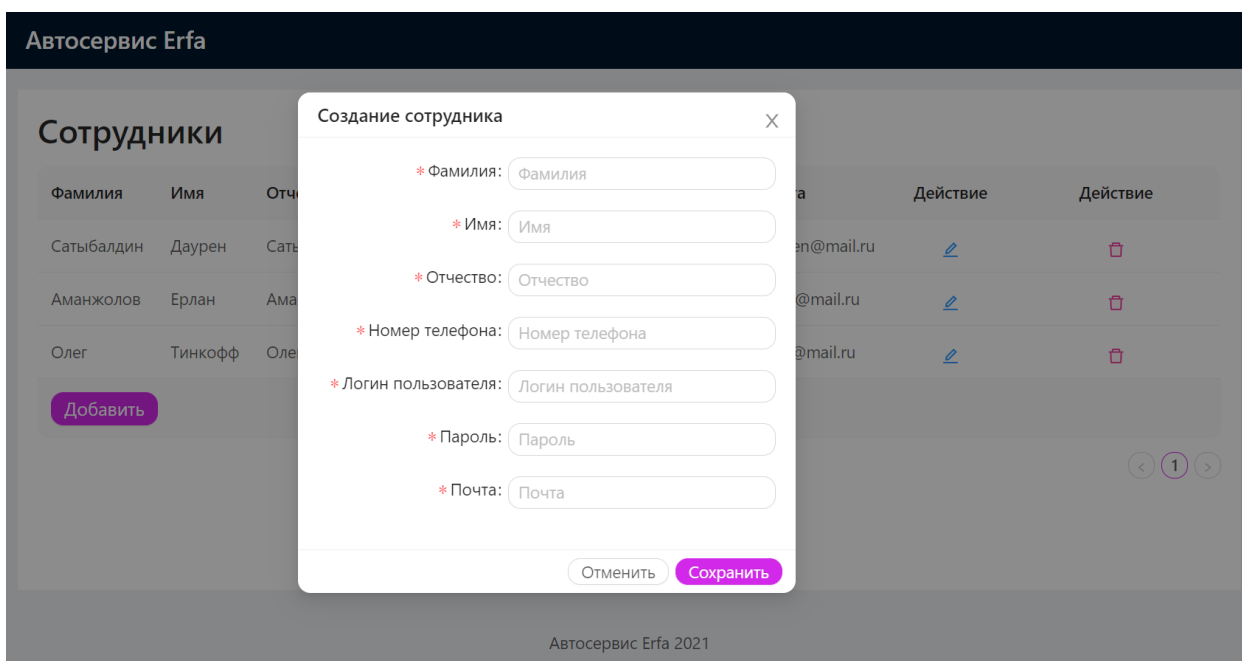
\* Почта:

### 3.5 Сурет – Клиенттерді қосуға арналған модальді терезе

Қызметкерлер парағында автокөлік қызметтерінің барлық қызметкерлерінің тізімі бар кесте бар. Бұл парақта қызметкерлерді құру, жою, редакциялау батырмалары бар.



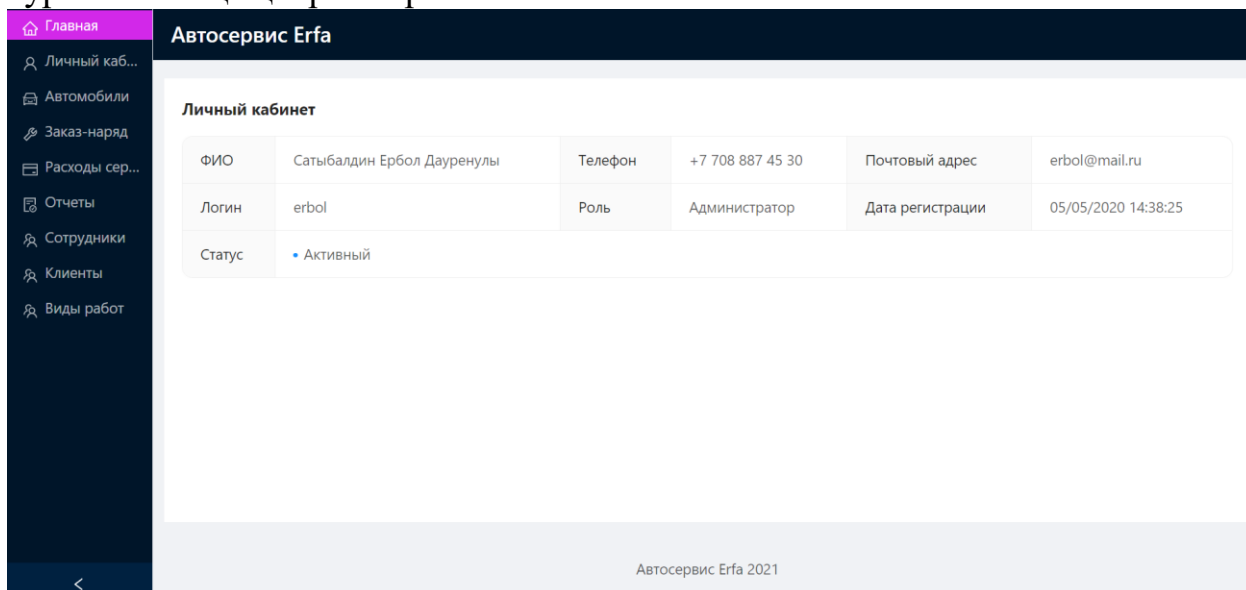
3.6 Сурет – Қызметкерлер парақшасы



3.7 Сурет – Жұмыскерлерді қосуға арналған модальді терезе

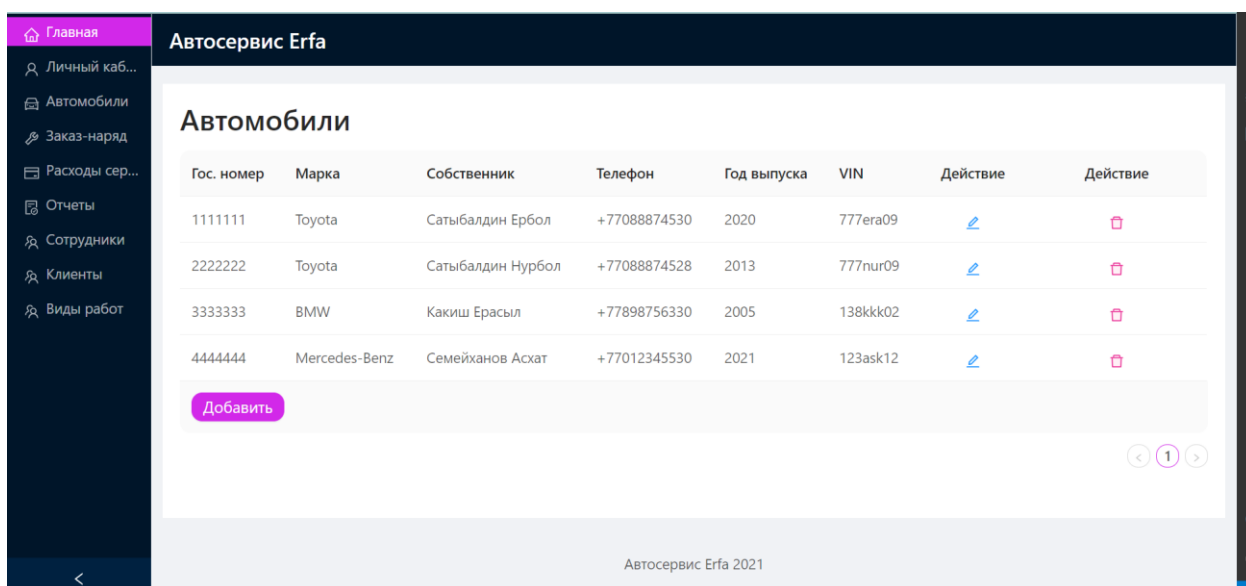


Жеке кабинеттің парағында қазіргі уақытта жүйеге кірген пайдаланушы туралы толық ақпарат бар.

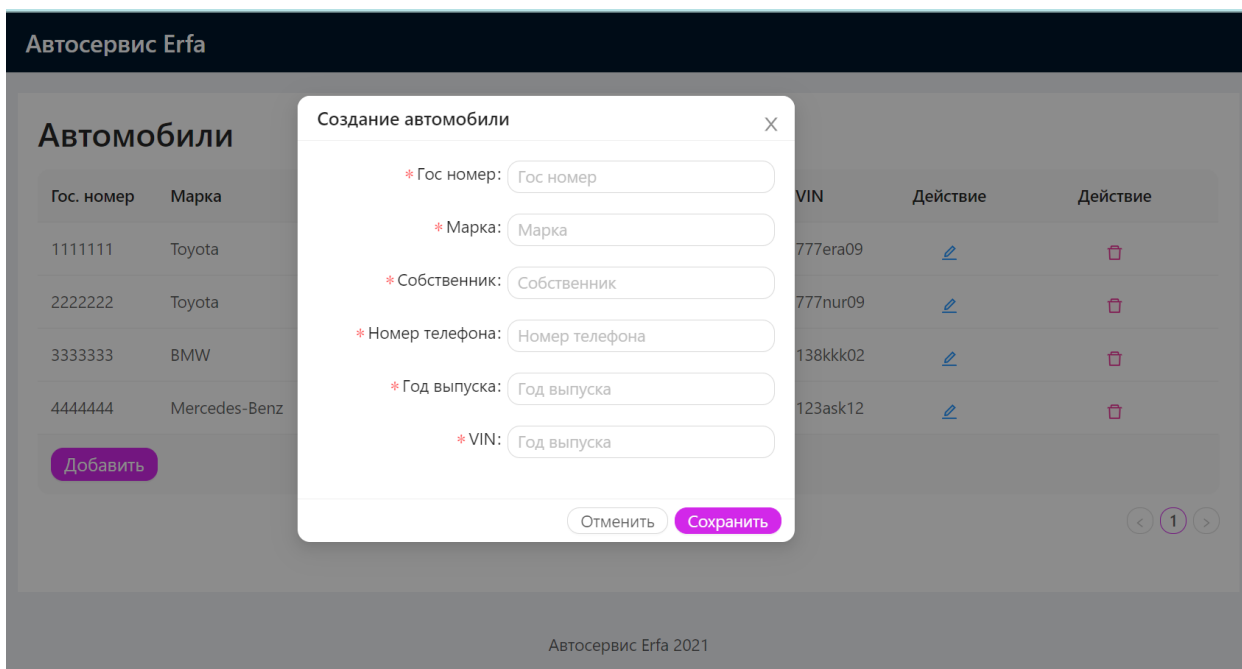


### 3.8 Сурет – Жеке кабинеттің парағы

Автокөліктер парағында жүйеге тіркелген барлық машиналардың тізімі келтірілген. Бұл кестеде машинаның маркасы, иесі, шығарылған жылы, нөмірі туралы мәліметтер келтірілген.

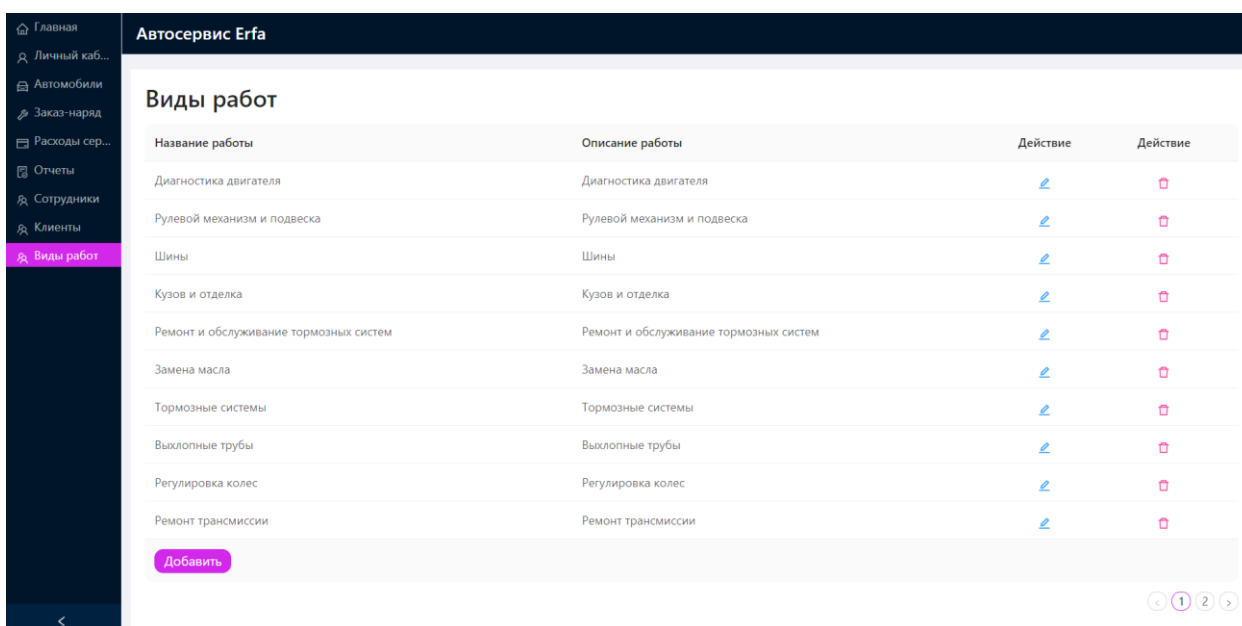


### 3.9 Сурет – Көліктер парақшасы

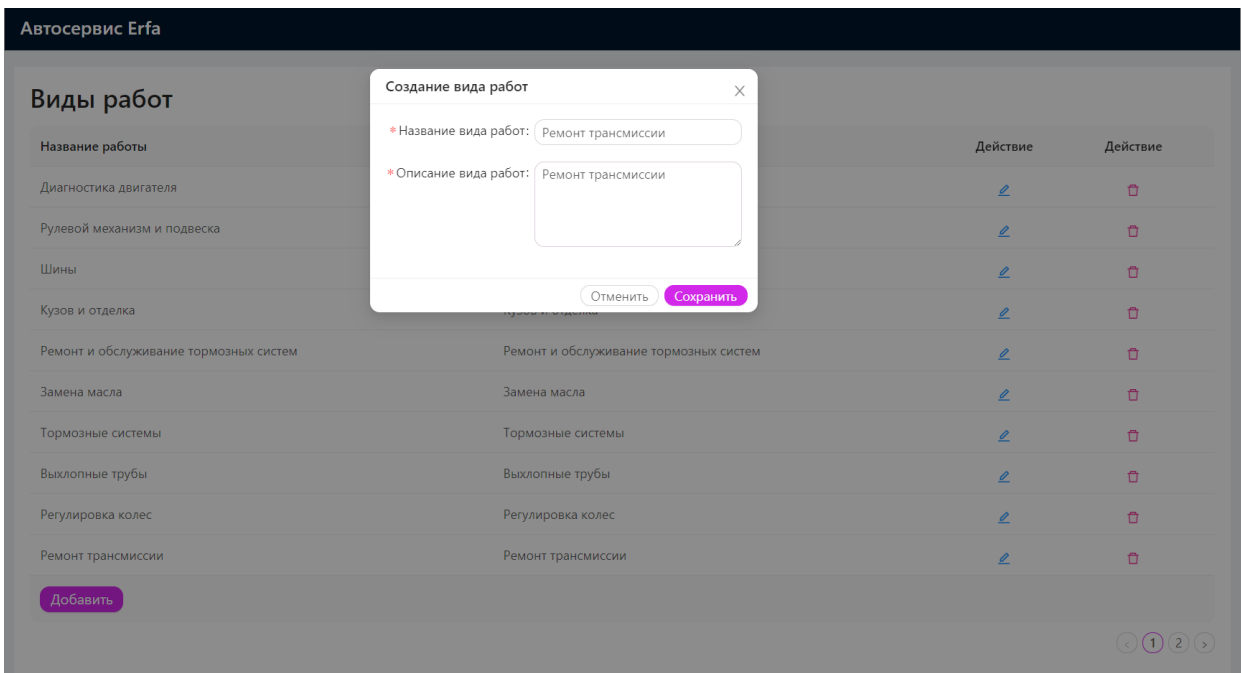


### 3.10 Сурет – Көліктерді жүйеге қосуға арналған модальді терезе

Жұмыс түрінің парағында автосервисте қандай жұмыс түрін ұсынуға болатындығы сипатталған.



### 3.11 Сурет – Жұмыс түрі парағы



3.12 Сурет – Жұмыс түрі парағы

## ҚОРЫТЫНДЫ

Дипломдық жұмыстың мақсаты – Автокөлік иелеріне автокөлік қызметіне жазылуға және автомобильдерге қосалқы бөлшектер сатып алуға мүмкіндік беретін WEB-қосымшасын құру.

WEB-қосымшаны әзірлеу барысында оңай масштабталатын және икемді қосымшаны құру үшін заманауи технологиялар қолданылды. Атап өткенде .NET Core ,Entity Framework Core , React , JavaScript, HTML, CSS және PostgreSQL.

Дипломдық жұмыс орындау барысында келесі нәтижелерге қол жеткізілді:

- Пәндік аймақ зерттелді.
- Қосымшаны пайдаланудың барлық пайдаланушылық сценарийлерін ескеретін жүйе әзірленді: авторизация, автокөліктерді есепке алу, нәтижелерді қарау.
- Есеп жүргізу үшін деректер базасы әзірленді.
- Веб-қосымшаның заманауи дизайны жасалды.
- Қосымшаның серверлік жағы жасалды.
- Қосымшаның клиенттік жағы жасалды.

## ПАЙДАЛАНЫЛҒАН ӘДЕБИЕТТЕР ТІЗІМІ

- 1 <https://ru.reactjs.org/docs/getting-started.html>
- 2 Mark Masse , REST API Design Rulebook , 2011, 166 с
- 3 Моргунов, Е. П. , PostgreSQL. Основы языка SQL: учеб. пособие / Е. П. Моргунов; под ред. Е. В. Рогова, П. В. Лузанова. — СПб.: БХВ-Петербург, 2018. — 336 с
- 4 Марк Дж. С# 7 и .NET Core: кросс-платформенная разработка для профессионалов: пер. с англ. / Марк Дж. Прайс; – 3-е изд. – Санкт-Петербург [и др.]: Питер, 2018. – 636 с
- 5 <https://docs.microsoft.com/en-us/dotnet/>
- 6 Дженнифер Нидерст Роббинс "HTML5, CSS3 и JavaScript. Исчерпывающее руководство" 529 с
- 7 Линн Б., Майкл М. PHP және MySQL, 2010 — 85 б
- 8 Леон Шкляр, Рич Розен , Архитектура веб-приложений , 2011, 634 с

## А қосымшасы

Login.cs – авторизацияның бетінің логиасы

```
using AvtoService.Models;
using AvtoService.Utills;
using Microsoft.AspNetCore.Mvc;
using Microsoft.Extensions.Configuration;
using Microsoft.Extensions.Options;
using Microsoft.IdentityModel.Tokens;
using System;
using System.Collections.Generic;
using System.IdentityModel.Tokens.Jwt;
using System.Linq;
using System.Security.Claims;
using System.Security.Cryptography;
using System.Text;

namespace AvtoService.Controllers
{
    [Route("api/[controller]")]
    [ApiController]
    public class LoginController : ControllerBase
    {
        private readonly BaseDBContext _dbContext;
        private readonly AuthOptions _options;
        private readonly IConfiguration _config;

        public LoginController(BaseDBContext dbContext,
                               IOptions<AuthOptions> options,
                               IConfiguration config)
        {
            _dbContext = dbContext;
            _options = options.Value;
            _config = config;
        }

        [HttpPost]
        public IActionResult Login([FromBody] Users loggedUser)
        {
            IActionResult response = Unauthorized();
            var user = Authenticate(loggedUser);
            if (user != null) {
```

### А қосымшасының жалғасы

```
var token = GenerateJWT(user);
response = Ok(new { token });
}
return response;

}

private IdentityUser Authenticate(Users loggedUser)
{
    IdentityUser user = null;
    if (_dbContext.Users.Any(m => m.Login == loggedUser.Login.ToLower()
    && m.Password == loggedUser.Password))
    {
        Users u = _dbContext.Users.FirstOrDefault(m => m.Login ==
loggedUser.Login.ToLower() && m.Password == loggedUser.Password);
        int? rolesId = _dbContext.UserRoles.FirstOrDefault(m => m.UserId ==
u.Id).RoleId;
        string role = _dbContext.Roles.FirstOrDefault(m => m.Id == rolesId).Value;

        _dbContext.SaveChanges();
        user = new IdentityUser()
        {
            Id = u.Id,
            Email = u.Email,
            Login = u.Login,
            Role = role
        };
    }
    return user;
}

private string GenerateJWT(IdentityUser user)
{
    var securityKey = new
SymmetricSecurityKey(Encoding.UTF8.GetBytes(_config["Jwt:SignInKey"]));
    var credentials = new SigningCredentials(securityKey,
SecurityAlgorithms.HmacSha256);
    var claims = new[] {
        new Claim(JwtRegisteredClaimNames.Sub, user.Login),
        new Claim(JwtRegisteredClaimNames.Email, user.Email),
    }
}
```

### А қосымшасының жалғасы

```
new Claim(JwtRegisteredClaimNames.NameId, user.Id.ToString()),
new Claim(JwtRegisteredClaimNames.Jti, Guid.NewGuid().ToString()),
new Claim(ClaimsIdentity.DefaultRoleClaimType, user.Role)
};
var token = new JwtSecurityToken(
    issuer: _config["Jwt:Issuer"],
    audience: _config["Jwt:Issuer"],
    claims,
    expires: DateTime.Now.AddMinutes(480),
    signingCredentials: credentials
);
var encodeToken = new JwtSecurityTokenHandler().WriteToken(token);
return encodeToken;

}

private string CreateToken(IEnumerable<Claim> claims)
{
    var securityKey = new
SymmetricSecurityKey(_options.GetSignInKeyBytes());
    var credentials = new SigningCredentials(
        securityKey,
        SecurityAlgorithms.HmacSha256);

    var token = new JwtSecurityToken(
        issuer: _options.Issuer,
        audience: _options.Audience,
        claims: claims,
        expires: DateTime.UtcNow.AddHours(_options.Expires),
        signingCredentials: credentials);

    return new JwtSecurityTokenHandler().WriteToken(token);
}

private string CreateHash(string value)
{
    if (string.IsNullOrEmpty(value))
    {
        throw new ArgumentException("Invalid password.", nameof(value));
    }

    using var sha = new SHA256Managed();
```



### **А қосымшасының жалғасы**

```
byte[] textData = Encoding.UTF8.GetBytes(value);
byte[] hash = sha.ComputeHash(textData);

return BitConverter
    .ToString(hash)
    .Replace("-", string.Empty)
    .ToLower();
}

private static Claim[] GetClaims(SomeExample user)
{
    return new[] {
        new Claim(ClaimTypes.Name, user.Name),
        new Claim(ClaimTypes.Email, user.Email),
        new Claim(ClaimTypes.Role, user.Email),
        new Claim(ClaimTypes.NameIdentifier, user.Login)};
}
}
```

## Б қосымшасы

index.js – навигациялық бет

```
import React, { useState,useContext } from 'react';
import {
  BrowserRouter as Router,
  Route,
  Link,
  Switch,
  Redirect } from 'react-router-dom';
import { Layout, Menu, PageHeader } from 'antd';
import {
  Login,
  MainPage,
  Reports,
  UserProfile,
  Employees,
  Clients,
  Cars,
  OrderRepair,
  ServiceSpending,
  WorkTypes,
} from '../pages';
import NotFound from '../pages/Error/NotFound';
import {
  MailOutlined,
  MenuFoldOutlined,
  MenuUnfoldOutlined,
  UserOutlined,
  CarOutlined,
  TeamOutlined,
  ToolOutlined,
  FileDoneOutlined,
  HomeOutlined,
  CreditCardOutlined,
} from '@ant-design/icons';
import './index.css';
import { Context } from '../common';
import jwt from 'jwt-decode';

const { Header, Content, Footer, Sider } = Layout;
```

## Б қосымшасының жалғасы

```
export const groupBy = (xs, key) => {
  return xs.reduce((rv, x) => {
    (rv[x[key]] = rv[x[key]] || []).push(x);
    return rv;
  }, {});
};

export const dataURLToFile = (dataurl, filename) => {

  var arr = dataurl.split(','),
      mime = arr[0].match(/:(.*?);/)[1],
      bstr = atob(arr[1]),
      n = bstr.length,
      u8arr = new Uint8Array(n);

  while (n--) {
    u8arr[n] = bstr.charCodeAt(n);
  }

  return new File([u8arr], filename, { type: mime });
}

export const shuffleVariants = (array) => {
  for (let i = array.length - 1; i > 0; i--) {
    const j = Math.floor(Math.random() * (i + 1));
    const temp = array[i];
    array[i] = array[j];
    array[j] = temp;
  }
  return array;
}

export const getBase64 = (file) => {
  return new Promise((resolve, reject) => {
    const reader = new FileReader();
    reader.readAsDataURL(file);
    reader.onload = () => resolve(reader.result);
    reader.onerror = error => reject(error);
  });
}
```

## Б қосымшасының жалғасы

```
export const PrivateRoute = ({ component: Component, roles, ...rest }) => {

const { token } = useContext(Context);

const grantPermission = (props) => {
  if (token) {
    const persistedToken = jwt(token);
    if (Date.now() >= persistedToken.exp * 1000) {
      return (
        <Redirect
          to={{
            pathname: '/error/sessionexpired',
            state: { from: props.location },
          }}
        />
      );
    }
  } else {
    return (
      <Redirect
        to={{
          pathname: '/login',
          state: { from: props.location },
        }}
      />
    );
  }
};

return <Route {...rest} render={(props) => grantPermission(props)} />;
};

const BaseLayout = () => {
  const [collapsed, setCollapsed] = useState(false);
  const names = [{
    name: 'Главная', title: 'Главная',
    name: 'Личный кабинет', title: 'Личный кабинет',
    name: 'Автомобили', title: 'Автомобили',
    name: 'Клиенты', title: 'Клиенты',
    name: 'Сотрудники', title: 'Сотрудники',
    name: 'Отчет', title: 'Отчет',
    name: 'Заказ-наряд', title: 'Заказ-наряд',
  ]
};
```

## Б қосымшасының жалғасы

```
name:'Страница входа',title:'Страница входа',
}]

return (

<Router>
  <Layout style={{ minHeight: '100vh' }}>
    <Sider collapsible collapsed={collapsed} onCollapse={(e) =>
setCollapsed(!collapsed)}>
      <div className="logo" />
      <Menu theme="dark" defaultSelectedKeys={['1']} mode="inline">
        <Menu.Item key="1" icon={<HomeOutlined />}>
          <span>Главная</span>
          <Link to="/" />
        </Menu.Item>
        <Menu.Item key="2" icon={<UserOutlined />}>
          <span>Личный кабинет</span>
          <Link to="/userProfile" />
        </Menu.Item>
        <Menu.Item key="3" icon={<CarOutlined />}>
          <span>Автомобили</span>
          <Link to="/cars" />
        </Menu.Item>
        <Menu.Item key="4" icon={<ToolOutlined />}>
          <span>Заказ-наряд</span>
          <Link to="/orderRepair" />
        </Menu.Item>
        <Menu.Item key="5" icon={<CreditCardOutlined />}>
          <span>Расходы сервиса</span>
          <Link to="/spendingService" />
        </Menu.Item>
        <Menu.Item key="6" icon={<FileDoneOutlined />}>
          <span>Отчеты</span>
          <Link to="/reports" />
        </Menu.Item>
        <Menu.Item key="7" icon={<TeamOutlined />}>
          <span>Сотрудники</span>
          <Link to="/employees" />
        </Menu.Item>
        <Menu.Item key="8" icon={<TeamOutlined />}>
          <span>Клиенты</span>
          <Link to="/clients" />
        </Menu.Item>
      </Menu>
    </Sider>
  </Layout>
</Router>
)
```

## Б қосымшасының жалғасы

```
    </Menu.Item>
    <Menu.Item key="9" icon={<TeamOutlined />}>
      <span>Виды работ</span>
      <Link to="/workTypes" />
    </Menu.Item>
  </Menu>
</Sider>

<Layout>
  <Menu theme="dark">
    <PageHeader className="site-page-header" title="Автосервис Erfa" />
  </Menu>
  <Content style={{ margin: '24px 16px', padding: 24, background: '#fff',
minHeight: 280 }}>
    <Switch>
      <Route exact={true} path="/" component={MainPage} />
      <Route path="/cars" component={Cars} />
      <Route path="/orderRepair" component={OrderRepair} />
      <Route path="/reports" component={Reports} />
      <Route path="/spendingService" component={ServiceSpending} />
      <Route path="/userProfile" component={UserProfile} />
      <Route path="/employees" component={Employees} />
      <Route path="/clients" component={Clients} />
      <Route path="/workTypes" component={WorkTypes} />

      <Route exact path={'/login'} component={Login} />
      <Route exact={true} path="*" component={NotFound} />
    </Switch>
  </Content>
  <Footer style={{ textAlign: 'center' }}>Автосервис Erfa 2021</Footer>
</Layout>
</Layout>
</Router>
);
};

export default BaseLayout;
```

## С қосымшасы

OrderRepair.js – көліктер парақшасы

```
import { Card, Col, Row, Table } from 'antd';  
import React from 'react';
```

```
function OrderRepair() {  
  const commentColumns = [  
    {  
      title: `№`,  
      dataIndex: 'governmentNumber',  
      key: 'governmentNumber',  
    },  
    {  
      title: `Коментарии`,  
      dataIndex: 'governmentNumber',  
      key: 'governmentNumber',  
    },  
    {  
      title: `Дата Добавления`,  
      dataIndex: 'model',  
      key: 'Model',  
    },  
  ];  
  const orderRepairColumns = [  
    {  
      title: `Дата`,  
      dataIndex: 'governmentNumber',  
      key: 'governmentNumber',  
    },  
    {  
      title: `Гос.номер`,  
      dataIndex: 'governmentNumber',  
      key: 'governmentNumber',  
    },  
    {  
      title: `Марка`,  
      dataIndex: 'model',  
      key: 'Model',  
    },  
    {  
      title: `Пробег (км)`,  
      dataIndex: 'governmentNumber',  
      key: 'governmentNumber',  },  
  ],
```

## С қосымшасының жалғасы

```
{
  title: `Клиент`,
  dataIndex: 'model',
  key: 'Model',
},
{
  title: `Телефон`,
  dataIndex: 'governmentNumber',
  key: 'governmentNumber',
},
{
  title: `Ответственный сотрудник`,
  dataIndex: 'model',
  key: 'Model',
},
{
  title: `Сумма запчастей (тг.)`,
  dataIndex: 'governmentNumber',
  key: 'governmentNumber',
},
{
  title: `Сумма работ (тг.)`,
  dataIndex: 'model',
  key: 'Model',
},
{
  title: `Оплачен`,
  dataIndex: 'model',
  key: 'Model',
},
},
];
const gridStyle = {
  width: '50%',
  textAlign: 'center', };
const style = { background: '#0092ff', padding: '1px 0' };
return (
  <div>
    <Row gutter=[[16, 24]]>
      <Col className="gutter-row" span={14}>
        <div>
          <Table columns={orderRepairColumns} bordered scroll={{ x: '15px' }} />
        </div>
      </Col>
    </Row>
  </div>
);
```



## С қосымшасының жалғасы

```
</Col>
<Col className="gutter-row" span={10}>
  <Card title="Информация">
    <Card.Grid hoverable={false} style={gridStyle}>
      Марка
    </Card.Grid>
    <Card.Grid hoverable={false} style={gridStyle}>
      Самру
    </Card.Grid>
    <Card.Grid hoverable={false} style={gridStyle}>
      Год выпуска
    </Card.Grid>
    <Card.Grid hoverable={false} style={gridStyle}>
      2021
    </Card.Grid>
    <Card.Grid hoverable={false} style={gridStyle}>
      VIN
    </Card.Grid>
    <Card.Grid hoverable={false} style={gridStyle}>
      АААААААААААААААААААА
    </Card.Grid>
    <Card.Grid hoverable={false} style={gridStyle}>
      Собственник
    </Card.Grid>
    <Card.Grid hoverable={false} style={gridStyle}>
      Ербол
    </Card.Grid>
    <Card.Grid hoverable={false} style={gridStyle}>
      Телефон
    </Card.Grid>
    <Card.Grid hoverable={false} style={gridStyle}>
      +77088874530
    </Card.Grid>
  </Card>
</Col> <Col className="gutter-row" span={14}>
  <div></div> </Col> <Col className="gutter-row" span={10}> <Table
columns={commentColumns} bordered scroll={{ x: '15px' }} />
  </Col> </Row> </div>
);
}

export default OrderRepair;
```

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

Сәтбаев университеті

**Ғылыми жетекшінің пікірі**

Дипломдық жұмыс

Сатыбалдин Ербол Дәуренұлы

5B070300 – Ақпараттық жүйелер

Тақырыбы: «Автосервис және автокөлікті жөндеу»WEB-қосымшасын әзірлеу

Бұл дипломдық жұмыс өзінің логикалық құрылымымен ерекшеленген. Түсіндірме жобаның құрамы кіріспеден, 3 бөлімнен, қорытындыдан, әдебиеттер тізімінен және қосымшадан тұрады.

Менің пікірімше, диплом жобалаушы алдына қойылған тапсырманы толығымен орындады және кейінгі технологияларын меңгергендігін көрсетті.

Жалпы дипломдық жоба профессионалдық деңгейде орындалған. Түсіндірме жазба сауатты бейнеленген, жоба бойынша барлық қажетті ақпараттар бар.

Кемшілік ретінде кейбір шағын стилистикалық қателерді атап кетуге болады.

Жоғарыда айтылғандарға байланысты, дипломдық жұмыс 5B070300 – «Ақпараттық жүйелер» мамандығының бітіру жұмыстарына қойылатын талаптарына сәйкес және дипломдық жұмыс қорғауға жіберіле алады, ал оның авторы Сатыбалдин Ербол Дәуренұлы

бакалавр академиялық дәрежесін алуға лайықты деп есептеймін.

Ғылыми жетекші

Ассоц.проф, т.ғ.к.



Жумағалиев Б.И.

«31» мамыр 2021 ж.

## 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:** Сатыбалдин Ербол

**Название:** «Автосервис және автокөлікті жөндеу» WEB-қосымшасын әзірлеу

**Координатор:** Жумагалиев Биржан

**Коэффициент подобия 1:** 5,43

**Коэффициент подобия 2:** 2,09

**Замена букв:** 19

**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:

Замены букв было связано с применением общих принятых сокращений в виде аббревиатур на английском языке.....

31 мамыр 2021

Дата

.....  .....

Подпись Научного руководителя

**Протокол анализа Отчета подобия  
заведующего кафедрой / начальника структурного подразделения**

Заведующий кафедрой / начальник структурного подразделения заявляет, что ознакомился(-ась) с полным отчетом подобия, который был сгенерирован Системой выявления и предотвращения плагиата в отношении работы:

**Автор:** Сатыбалдин Ербол

**Название:** «Автосервис және автокөлікті жөндеу» WEB-қосымшасын әзірлеу

**Координатор:** Жумагалиев Биржан

**Коэффициент подобия 1:** 5,43

**Коэффициент подобия 2:** 2,09

**Замена букв:** 19

**Интервалы:** 0

**Микропробелы:** 0

**Белые знаки:** 0

**После анализа отчета подобия заведующий кафедрой / начальник структурного подразделения констатирует 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:**

.....  
.....Заимствования не обнаружены.....  
.....

Дата 31.05.2021



Подпись заведующего кафедрой КОиХИ / Сейлова Н.А.

начальника структурного подразделения

**Окончательное решение в отношении допуска к защите, включая обоснование:**

.....  
.....Допускается к защите.....  
.....

Дата 31.05.2021



Подпись заведующего кафедрой КОиХИ / Сейлова Н.А.

начальника структурного подразделения



## Metadane

Tytuł

Автосервис және автокөлікті жөндеу»WEB-қосымшасыназірлеу

Autorizy

Сатыбалдин Ербол Дауренұлы

Promotor

Биржан Жумағалиев

Jednostka organizacyjna

ИКИИТ

## Alerty

W tej sekcji znajdują się statystyki występowania w tekście zabiegów edytorskich, które mogą mieć na celu zaburzenie wyników analizy. Niewidoczne dla osoby zapoznającej się z treścią pracy na wydruku lub w pliku, wpływają na frazy porównywane podczas analizy tekstu (poprzez celowe błędy pisowni) w celu ukrycia zapożyczeń lub obniżenia wyników w Raporcie podobieństwa. Należy ocenić, czy zaznaczone wystąpienia wynikają z uzasadnionego formatowania tekstu (nadwrażliwość systemu), czy są celową manipulacją.

Znaki z innego alfabetu		18
Rozstrzelenia		0
Mikrospacje		0
Białe znaki		0
Parafrazy		19

## Metryka podobieństw

Należy pamiętać, że wysokie wartości Współczynników nie oznaczają automatycznie plagiatu. Raport powinien zostać przeanalizowany przez kompetentną / upoważnioną osobę. Wyniki są uważane za wymagające szczegółowej analizy, jeśli WP 1 wynosi ponad 50%, a WP 2 ponad 5%.



WP1

25

Długość frazy dla WP 2



WP2

3351

Liczba słów



CYT

27322

Liczba znaków

## Aktywne listy podobieństw

Uwagi wymagają szczególnie fragmenty, które zostały włączone do WP 2 (zaznaczone pogrubieniem). Użyj linku "Pokaż w tekście" i zobacz, czy są to krótkie frazy rozproszone w dokumencie (przypadkowe podobieństwa), skupione wokół siebie (parafraza) lub obszerne fragmenty bez wskazania źródła (tzw. "kryptocytaty").

### 10 najdłuższych fragmentów

Kolor w tekście

LP	TYTUŁ LUB ADRES URL ŹRÓDŁA (NAZWA BAZY)	IDENTYCZNYCH SŁÓW (FRAGMENTÓW)	
1	<a href="https://itacademy.kz/">https://itacademy.kz/</a>	43	1.28 %
2	<a href="https://mlky.ru/sample-page/%D0%BC%D0%B0%D1%82%D0%B5%D1%80%D0%BB%D0%B0%D0%BB%D1%8B-%D0%BF%D0%BE-%D1%81%D1%83%D0%B1%D0%BA/">https://mlky.ru/sample-page/%D0%BC%D0%B0%D1%82%D0%B5%D1%80%D0%BB%D0%B0%D0%BB%D1%8B-%D0%BF%D0%BE-%D1%81%D1%83%D0%B1%D0%BA/</a>	27	0.81 %
3	Дипломка Аманжолқызы.doc Аманжолқызы Карагат 5/15/2019 E.A.Buketov Karaganda State University (Математики и информационных технологий)	23	0.69 %

4	Інформаційна система для ІТ компанії "Верда" ІТІТ-21 Леманіч 12/10/2018 National University "Lviv Politechnika"2 (NULP2)	16	0.48 %
5	yulia_miloslavska_diplom.docx Мілославська Юлія 9/11/2019 Sumy State University (Кафедра комп'ютерних наук, секція ІКТ)	15	0.45 %
6	<a href="https://aldebbaran.ru/author/kj_rzayis_mark/kniga_c_7_i_net_core_kross_platfornennaya_razr/">https://aldebbaran.ru/author/kj_rzayis_mark/kniga_c_7_i_net_core_kross_platfornennaya_razr/</a>	15	0.45 %
7	<a href="https://gist.github.com/kota8/cc0c3f21d8e0822b481ec6ccd488457f">https://gist.github.com/kota8/cc0c3f21d8e0822b481ec6ccd488457f</a>	13	0.39 %
8	<a href="http://books.ictedu.info/bootime/programim-ne-webushtrime/kapitulli-19-cookie-dhe-sesionet">http://books.ictedu.info/bootime/programim-ne-webushtrime/kapitulli-19-cookie-dhe-sesionet</a>	11	0.33 %
9	yulia_miloslavska_diplom.docx Мілославська Юлія 9/11/2019 Sumy State University (Кафедра комп'ютерних наук, секція ІКТ)	8	0.24 %
10	<a href="http://books.ictedu.info/bootime/programim-ne-webushtrime/kapitulli-19-cookie-dhe-sesionet">http://books.ictedu.info/bootime/programim-ne-webushtrime/kapitulli-19-cookie-dhe-sesionet</a>	6	0.18 %

#### z bazy RefBooks (0.00 %)

LP	TYTUL	IDENTYFICACIJA SLOV (FRAGMENTOV)
----	-------	----------------------------------

#### z bazy macierzystej (0.00 %)

LP	TYTUL	IDENTYFICACIJA SLOV (FRAGMENTOV)
----	-------	----------------------------------

#### z Programu Wymiany Baz (1.85 %)

LP	TYTUL	IDENTYFICACIJA SLOV (FRAGMENTOV)	
1	yulia_miloslavska_diplom.docx Мілославська Юлія 9/11/2019 Sumy State University (Кафедра комп'ютерних наук, секція ІКТ)	23 (2)	0.69 %
2	Диплома Аманжолқызы.doc Аманжолқызы Каражат 5/15/2019 E.A.Buketov Karaganda State University (Математики и информационных технологий)	23 (1)	0.69 %
3	Інформаційна система для ІТ компанії "Верда" ІТІТ-21 Леманіч 12/10/2018 National University "Lviv Politechnika"2 (NULP2)	16 (1)	0.48 %

#### z Internetu (3.58 %)

LP	ADRES URL ZHODLA	IDENTYFICACIJA SLOV (FRAGMENTOV)	
1	<a href="https://itcademy.kz/">https://itcademy.kz/</a>	43 (1)	1.28 %
2	<a href="https://miky.ru/sample-page/%D0%BC%D0%B0%D1%82%D0%B5%D1%80%D0%B8%D0%B0%D0%BB%D1%8B-%D0%BF%D0%BE-%D1%81%D1%83%D0%B1%D0%B4/">https://miky.ru/sample-page/%D0%BC%D0%B0%D1%82%D0%B5%D1%80%D0%B8%D0%B0%D0%BB%D1%8B-%D0%BF%D0%BE-%D1%81%D1%83%D0%B1%D0%B4/</a>	27 (1)	0.81 %
3	<a href="http://books.ictedu.info/bootime/programim-ne-webushtrime/kapitulli-19-cookie-dhe-sesionet">http://books.ictedu.info/bootime/programim-ne-webushtrime/kapitulli-19-cookie-dhe-sesionet</a>	22 (3)	0.66 %
4	<a href="https://aldebbaran.ru/author/kj_rzayis_mark/kniga_c_7_i_net_core_kross_platfornennaya_razr/">https://aldebbaran.ru/author/kj_rzayis_mark/kniga_c_7_i_net_core_kross_platfornennaya_razr/</a>	15 (1)	0.45 %
5	<a href="https://gist.github.com/kota8/cc0c3f21d8e0822b481ec6ccd488457f">https://gist.github.com/kota8/cc0c3f21d8e0822b481ec6ccd488457f</a>	13 (1)	0.39 %