

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ
Сәтбаев университеті

Ақпараттық және телекоммуникациялық технологиялар институты

Киберқауіпсіздік, ақпаратты өңдеу және сақтау кафедрасы

Ыдырыс Нұрсұлтан Сәкенұлы

«Мектеп асханасындағы менюді қашықтықтан көре алу және тапсырыс беру үшін мобильді веб-қосымшасын әзірлеу»

ДИПЛОМДЫҚ ЖҰМЫС

5B070300 – «Ақпараттық жүйелер» мамандығы


Алматы 2021

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

Сәтбаев университеті

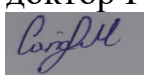
Ақпараттық және телекоммуникациялық технологиялар институты

Киберқауіпсіздік, ақпаратты өңдеу және сақтау кафедрасы

ҚОРҒАУҒА ЖІБЕРІЛДІ
КАӨЖС кафедра меңгерушісі,
тех.ғыл.канд, асс.профессор
 Н.А.Сейлова
« 03 » 06 2021 ж.

ДИПЛОМДЫҚ ЖҰМЫС

Тақырыбы: «Мектеп асханасындағы менюді қашықтықтан көре алу және тапсырыс беру үшін мобильді веб-қосымшасын әзірлеу»
5B070300 – «Ақпараттық жүйелер» мамандығы

Орындаған:
Бідырыс Н.С.
Ғылыми жетекші:
доктор PhD, аға оқушы
 М.Н.Сатымбеков
«25» мамыр 2021 ж.

Алматы 2021

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ


Сәтбаев университеті

Ақпараттық және телекоммуникациялық технологиялар институты

Киберқауіпсіздік, ақпаратты өңдеу және сақтау кафедрасы

БЕКІТЕМІН

КАӨС кафедра меңгерушісі,
тех.ғыл.канд, асс. профессор

 Н.А.Сейлова
«_03_»__06__2021 ж.

**Дипломдық жұмысты орындауға
ТАПСЫРМА**

Білім алушы: Ыдырыс Нұрсұлтан Сәкенұлы

Тақырыбы: «Мектеп асханасындағы менюді қашықтықтан көре алу және тапсырыс беру үшін мобильді веб-қосымшасын әзірлеу»

Университет Ректорының 2020 жылғы «24» 11 №2131-б бұйрығымен бекітілген

Аяқталған жұмысты тапсыру мерзімі 2021 жылғы «25» мамыр

Дипломдық жұмыстың бастапқы берілістері: диплом алдындағы практикалық жұмыс нәтижесі мен қорытындысы, дипломдық тақырыпқа шолу, теориялық деректер

Дипломдық жұмыстың қысқаша мазмұны:

- а) мектеп асханасының жұмыс істеу принципі;
- ә) веб-қосымша құру барысы;
- б) мәліметтер базасын құру;
- в) веб-қосымша әзірленуі.

Сызбалық материалдар тізімі: Power Point сызбалары

Сызба материалдар: ___ слайдтар көрсетілген

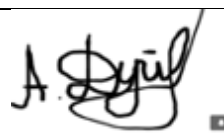
Ұсынылатын негізгі әдебиет: 8 атау

Дипломдық жұмысты дайындау

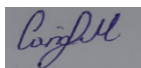
КЕСТЕСІ

Бөлімдер атауы, қарастырылатын мәселелер тізімі	Ғылыми жетекші мен кеңесшілерге көрсету мерзімдері	Ескерту
Мектеп асхана жұмысын талдау	20.01.2021 – 02.02.2021	
Деректер базасын әзірлеу	15.02.2021 – 11.03.2021	
Веб-қосымша құру	08.03.2021 – 19.04.2021	

Дипломдық жұмысының бөлімдерінің кеңесшілері мен норма бақылаушыларының аяқталған жобаға қойған **қолтаңбалары**

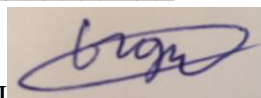
Бөлімдер атауы	Кеңесшілер, аты, әкесінің аты, тегі (ғылыми дәрежесі, атағы)	Қол қойылған күні	Қолы
Норма бақылаушы	Дүйсенбаева А.Н. Лектор, тех-ғыл магистрі	31..05.21	
Программалық қамтама			

Ғылыми жетекші



М.Н.Сатымбеков

Тапсырманы орындауға алған білім алушы



Н.С. Ыдырыс

Күні

«24» қараша 2021 ж

АҢДАТПА

Дипломдық жұмыста оқушылар үшін ыңғайлы және түсінікті интерфейсі бар, мектеп мәзіріне оңай қол жеткізуге мүмкіндік беретін қосымша әзірленеді.

Мектеп мәзірінің қосымшасы мектеп асханасының жұмысын орындайды. Оқушылардың ата-аналары өз балаларының ас мәзірін бақылай алады. Мектептегі түскі ас туралы маңызды хабарламалар алады және балаларыңызда пайда болуы мүмкін аллергияларды таңдап, күнделікті мәзірде алаңдаушылық тудыратын тағамдар ерекшеленеді.

Мектеп асханасының қосымшасы PostgreSQL мәліметтер базасын қолданады. Қосымша қолданушы интерфейсі React кроссплатформалық фреймворк көмегімен құрылды.

Қосымшада оқушыға тамақтануға қолжетімді тағамдар көрсетіледі. Керекті тамақты таңдап, тапсырыс беруге мүмкіншілік бар. Сонымен қатар ата-ана бақылауы жасалған.

АННОТАЦИЯ

В дипломной работе разработано приложение, имеющее удобный и понятный интерфейс для учащихся, позволяющий легко получить доступ к школьному меню.

Приложение школьного меню выполняет работу школьной столовой. Родители учащихся могут следить за меню своих детей. Получает важные сообщения о школьных обедах и выделяет тревожные продукты в ежедневном меню, выбирая аллергены, которые могут появиться у ваших детей.

Приложение школьной столовой использует базу данных PostgreSQL. Дополнительный пользовательский интерфейс был создан с помощью кроссплатформенного фреймворка React.

В приложении указываются продукты, доступные ученику для питания. Есть возможность выбрать и заказать нужную еду. Также разработан родительский контроль.

ANNOTATION

The thesis developed an application that has a user-friendly and intuitive interface for students, allowing easy access to the school menu.

The school menu app does the work of the school cafeteria. Parents of students can follow the menu of their children. Receives important messages about school lunches and highlights disturbing foods in the daily menu, selecting allergens that may appear in your children.

The school cafeteria app uses a PostgreSQL database. An additional user interface was created using the React cross-platform framework.

The appendix specifies the foods available to the student for nutrition. It is possible to choose and order the desired food. Parental controls have also been developed.

МАЗМҰНЫ

Кіріспе	9
1 Мектеп асханасына шолу	10
1.1 Мектеп асханасының оқушыларға әсері	10
1.2 Балалардың тамақтануы туралы фактілер	10
1.3 Есептің қойылымы	11
2 Веб-қосымшаның әзірленуі	13
2.1 PostgreSQL мәліметтер базасын басқару жүйесі	13
2.2 .NET Core модульді платформасы	14
2.3 React әзірлеу ортасы	14
2.4 C# бағдарламалау тілі	15
2.5 Мәліметтер базасының құрылымы	16
2.6 Іс-әрекеттер диаграммасы	18
2.7 Жоба прототипі	19
3 Жоба жұмысының сипаттамасы	24
3.1 Графикалық пайдаланушы интерфейсі	24
3.2 Веб-қосымша жұмысы	25
Қорытынды	35
Пайдаланынылған әдебиеттер тізімі	36
Приложение А	37

КІРІСПЕ

Біз қызықты уақытта өмір сүріп жатырмыз. Біз мұны тұтынушылық әдеттердің немесе демографияның өзгеруімен байланыстыра аламыз. Оған қоса технологиялардың дамуы да байланысты. Қазіргі технология көмегімен адамдар өздерінің физикалық іс-әрекеттерін жеңілдетіп, ол жұмысты компьютерлер, смартфондар және т.б. орындайтыны мәлім.

Білім беру мекемелерінде оқитын балалары жаңа технологиялар қолданады. Олардың көмегімен көптеген ақпараттаға ғаламтор арқылы аздаған уақытта лезде қол жеткізе алады. Сол себепті уақыт үнемдеу мен ыңғайлылық сұраныста. Сол сұранысты қанағаттандыру мақсатын орындау кезінде, оқушылардың тамақтану процессін жеңілдету үшін мектеп асханасының веб-қосымшасы жасалуы шешілді.

Мектептерде пайдалы тамақ болған кезде балалар тамақтану әдеттерін жақсартады. Мектептер өмір бойы дұрыс тамақтану әдеттерін қалыптастыруда маңызды рөл атқарады, балалар тамақтану бағдарламалары аясында толыққанды түскі ас ұсынулары да тиіс. Мектептегі түскі асқа сүт, жемістер, көкөністер және дәнді дақылдар кіреді, сонымен қатар кальций мен талшық сияқты маңызды қоректік заттар бар. Оқушылар мектеп қабырларында немесе үйден виртуалды түрде білім алған кезде, мектептер оқушылардың тамақтануға қол жетімділігін қамтамасыз ету үшін әртүрлі әдістерді қолдана алады.

Дипломдық жұмыс барысында әзірленген веб-қосымша мектеп асханасының функцияларын қамтып, жасалады. Оқушы мен мектеп қызметкерлерінің тамақтануын жақсартуында веб-қосымша үлкен рөл атқарады.

1 МЕКТЕП АСХАНАСЫНА ШОЛУ

1.1 Мектеп асханасының оқушыларға әсері

Мектеп асханасы - бұл көптеген мектептер ұсынатын қызмет.

Тамақтану әдеті бүкіл әлемдегі ата-аналарды алаңдатады, өйткені олар балаларына дұрыс тамақтану маңызды екенін біледі. Алайда, тамақтану уақыты үлкен ұрыс алаңына айналуы мүмкін, бұл алаңдаушылыққа әкелуі мүмкін, бұл біздің балаларымызбен қарым-қатынасымызға әсер етуі мүмкін.

Мектептегі асхана балаларға күнделікті тәртіп пен кестені меңгеруге көмектеседі. Мектеп асханаларында балалар күн сайын белгілі бір қадамдарды қайталайды. Сондықтан барлық балалар оларды үйренуіне мәжбүр болады. Бірінші жағынан, олардың тамақтануға уақыты шектеулі болғандықтан, мектеп асханалары тамақтану кезінде шоғырлануды арттыруға көмектеседі. Екіншіден, олар балаларға тамақтану уақытын ұзартпауға үйренуге көмектеседі.

1.2 Балалардың тамақтануы туралы фактілер

Мектеп асханасы – бұл балалар сабақтан демалып, қарым-қатынас орнатып, толықтай тамақтана алатын орын. Мектеп асханасы тамақтанушыларға бірінші кезекте қауіпсіз және пайдалы дайындауы тиіс.

Балалық шақта және жасөспірімде дұрыс тамақтану дұрыс өсу мен даму үшін, сондай-ақ әртүрлі аурулардың алдын алу үшін маңызды. Мектеп асханалары оқушыларға мынадай тағамдарды ұсынғандары тиіс:

- Түрлі жемістер мен көкөністер.
- Тұтас дәнді.
- Майсыз және майсыз сүт өнімдері.
- Ақуыз тағамдарының алуан түрлілігі.
- Майлар.

Адамдарға қатты майлардан (қаныққан және транс май қышқылдарының негізгі көздері) және қосылған қанттардан калория мөлшерін шектеуге, сонымен қатар натрий қабылдауын азайтуға кеңес беріледі. Өкінішке орай, балалар мен жасөспірімдердің көпшілігі кеңестерді орындамайды.

Дұрыс тамақтану адамдарға дұрыс дене салмағын көтеруге және сақтауға, маңызды қоректік заттарды тұтынуға және төменде жазылған аурулардың пайда болу қаупін азайтуға көмектеседі:

- Жоғары қан қысымы.
- Жүрек ауруы.
- 2 типті қант диабеті.
- Рак.
- Остеопороз.
- Темірдің жетіспеушілігі.
- Тіс кариесі (кариес).

Жастардың тамақтануы жайлы:

– 2005-2018 жылдар аралығында балалар мен жасөспірімдер арасында құрамында қант бар сусындарды тұтыну төмендеді, бірақ жалпы тұтынылатын калориялардың 10% құрайды.

– 2003-2019 жылдар аралығында балалар мен жасөспірімдер арасында жеміс-жидек пен тұтас жемістерді тұтыну өсті. Алайда, жастардың көпшілігі әлі де жемістер мен көкөністерге қатысты ұсыныстарды орындамайды;

– Қосылған қант пен қатты майлардан алынған бос калория 2-18 жас аралығындағы балалар мен жасөспірімдер үшін күнделікті калориялардың 40% құрайды, бұл олардың рационының жалпы сапасына әсер етеді. Бұл бос калориялардың жартысына жуығы алты заттан келеді: газдалған сусындар, жеміс сусындары, сүт десерттері, дәмді десерттер, пицца және тұтас сүт. Жастардың көпшілігі ұсынылған су мөлшерін тұтынбайды.

Диета және оқу үлгерімі:

– Оқушыларға оқып үйрену және тәжірибеге дұрыс тағамдарды енгізу мектептің жағдайы бар.

– Салауатты таңғы ас танымдық жақсарумен (әсіресе есте сақтау қабілетімен), серуендеу санының азаюымен және көңіл-күйдің жақсаруымен байланысты.

– Тиісті гидратация балалар мен жасөспірімдердегі когнитивтің функцияларды жақсартып алады, бұл оқу үшін маңызды.

Дұрыс тамақтану туралы хабарламалар таңертеңгі хабарландырулар, мектеп жиналыстары, ата-аналарға жіберілген материалдар, қызметкерлер жиналысы және ата-аналар жиналыстарында айтылып, жазылғаны дұрыс шешімдердің бірі.

Асхана қызметкерлері – бұл балалардың білім алуына өз үлестерін қосатын адамдар тобының бірі. Олар мұғалімдермен, жаттықтырушылармен, кеңесшілермен, кітапханашылармен және басқа мектеп қызметкерлерімен бірге балаларға мектеп жылдарынан барынша ләззат алуға көмектеседі.

1.3 Есептің қойылымы

Дипломдық жұмыстың басты мақсаты - балаларды жақсы тамақпен қамтамасыз ететін және олардың денсаулығы мен әл-ауқатын сақтайтын асханасына веб-қосымша жасау. Сабақ уақытында мектеп оқушылары дұрыс тамақтанбау мүмкінді жоғары. Оның бір себебі, сабақ арасындағы үзілісте асханаға барып, кезекте тұрып, тамақ таңдауымен көп уақыт кетеді. Соның шешімі мектеп асханасына смартфон арқылы қол жеткізу. Смартфонға тек қана әзірленген веб-қосымша керек. Оны ашу үшін тек бір-екі қадам жасауға болады. Қазіргі уақытта білім алушылардың көбісінің қолдарында мобильді құрылғы бар. Сол себепті, мектеп асханасына веб-қосымша әзірлеу оқушыларға да, асхана қызметкерлеріне де тиімді болып келеді.

Асханадан мүмкіндігінше жергілікті өнімдерді сатып алуды, тамақ гигиенасы, денсаулық және қауіпсіздікті қоса алғанда, ең жоғары кәсіби стандарттарды сақтауды талап ету керек.

Әзірленетін веб-қосымша жақсы нәтижеге жету үшін келесі міндеттерді орындау керек:

- Мектеп асханасының жұмысын зерттеу.
- Әр адамға түсінікті графикалық интерфейс құру.
- Қарапайым мәліметтер базасын жасау.

Веб-қосымша оқушылар мен асхана қызметкерлеріне арнайы ыңғайлы жасалуы тиіс.

2 Веб-қосымшаның әзірленуі ВЕБ-ҚОСЫМШАНЫҢ ӘЗІРЛЕНУІ

2.1 PostgreSQL мәліметтер базасын басқару жүйесі

PostgreSQL – бұл корпоративті деректер базасының қосымшаларына арналған ашық кодты дерекқорды басқару жүйесі. Ол SQL-ді де, JSON-ды да SQL-ді кеңейту және сәйкестендіру үшін реляциялық және реляциялық емес сұраулар үшін қолдайды. Сондай-ақ, ол мәліметтер базасының күрделі жүктемелерін сақтау және масштабтау үшін кеңейтілген мәліметтер типтері мен өнімділікті оңтайландыру мүмкіндіктерін ұсынады. Ол Postgres ретінде де белгілі.

PostgreSQL (бастапқыда Postgres деп аталды) информатика профессоры Майкл Стоунбрейкер және оның командасы жасаған. Бүгінгі таңда бұл танымал ашық кодты дерекқорлардың бірі.

Төменде PostgreSQL-дің негізгі ерекшеліктері келтірілген:

- Әзірлеушілерге қосымшалар жасауға көмектесу.
- Администраторларға деректердің тұтастығын қорғау арқылы қателіктерге төзімді орта құруға көмектесу.
- Барлық негізгі тілдерді және аралық бағдарламаларды қолданатын әртүрлі платформалармен үйлесімділік.
- Ол ең күрделі құлыптау механизмін ұсынады.
- Параллелизмді мультиверсиялық басқаруды қолдау.
- Серверлік бағдарламалаудың жетілген функционалдығы.
- ANSI SQL стандартына сәйкес келеді.
- Клиент-сервердің желілік архитектурасын толық қолдайды.
- SSL журналдары мен триггерлеріне негізделген репликациясы.
- Резервтік сервер және жоғары қол жетімділік.
- Нысанға бағытталған және ANSI-SQL2008 үйлесімді.
- JSON қолдауы NoSQL сияқты басқа деректер қоймаларымен байланысуға мүмкіндік береді, олар полиглоттардың мәліметтер базасына арналған біріктірілген хаб ретінде әрекет етеді.

PostgreSQL артықшылықтары:

- PostgreSQL динамикалық веб-сайттар мен веб-қосымшаларды LAMP стек опциясы ретінде іске қоса алады.
- PostgreSQL бастапқы ашық коды лицензия бойынша еркін қол жетімді. Бұл сіздің бизнесіңіздің қажеттіліктеріне сәйкес пайдалануға, өзгертуге және енгізуге еркіндік береді.
- PostgreSQL географиялық нысандарды қолдайды, сондықтан сіз оны орналасқан жер мен географиялық ақпараттық жүйелерге негізделген қызметтер үшін пайдалана аласыз.
- PostgreSQL географиялық нысандарды қолдайды, сондықтан оны орналасу және географиялық ақпараттық жүйелер негізінде қызметтер үшін геокеңістіктік деректер қоймасы ретінде пайдалануға болады.

– Postgres-ті үйрену үшін көп жаттығудың қажеті жоқ, өйткені оны пайдалану оңай.

– PostgreSQL-ді кірістірілген және корпоративті пайдалану үшін техникалық қызмет көрсету мен басқарудың төмен құны кетеді.

Егер сіздің веб-сайтыңыз секундына жүздеген, тіпті мыңдаған сұраныстарды өңдеуді қажет етсе, Postgres ең жақсы шешім болып келеді.

PostgreSQL Django, Node.js сияқты барлық заманауи веб-фреймворктарымен жақсы жұмыс. Ол сонымен қатар кез-келген дерекқор серверлерін масштабтауға мүмкіндік беретін репликация мүмкіндіктерін ұсынады.

2.2 .NET Core модульді платформасы

.NET Core – бұл .NET Framework-тің жаңа нұсқасы, Microsoft қолдайтын жалпы мақсаттағы ашық кодты платформасы. Бұл Windows, macOS және Linux операциялық жүйелерінде жұмыс істейтін кросс-платформалық орта.

.NET Core-ды мобильді, веб, бұлтты, IoT, машиналық оқыту, микросервис, ойындар және т. б. сияқты әртүрлі қосымшаларды құру үшін пайдалануға болады.

Қосымша мүмкіндіктер Nuget пакеттері түрінде беріледі, оларды қажет болған жағдайда әзірленіп жатқан бағдарламаңызға қосуға болады. Осылайша, .NET Core өнімділікті арттырады, жад көлемін азайтады және техникалық қызмет көрсетуді жеңілдетеді.

.NET Core сипаттамалары:

- Ашық кодты платформа.
- Кроссплатформалы орта.
- Архитектуралар арасындағы үйлесімділік.
- Қосымшалардың кең спектрі.
- Бірнеше тілді қолдайды.
- Модульді архитектура.
- Икемді орналастыру.

.NET Core Framework келесі бөліктерден тұрады:

- CLI құралдары: әзірлеу және орналастыру құралдарының жиынтығы.
- Roslyn: C # және Visual Basic үшін тіл құрастырушысы.
- CoreFX: жақтаулар жиынтығы.
- CoreCLR: JIT негізіндегі CLR ортасы (командалық тілдің орындалу ортасы).

2.3 React әзірлеу ортасы

React – бұл пайдаланушы интерфейстерін құруға арналған ашық кодты JavaScript интерфейсi. React-ті Facebook және жеке әзірлеушілер мен компаниялар қауымдастығы қолданады. Ол бір беттік веб-сайттар мен мобильді қосымшаларды құрудың негізі ретінде кеңінен қолданылады. Пайдалану өте

оңай және пайдаланушыларға UI компоненттерін қайта пайдалануға мүмкіндік береді.

ReactJS Ерекшеліктері:

1) JSX: JSX - бұл javascript кеңейтімі. React-те JSX қолдану міндетті емес, бірақ бұл жақсы мүмкіндіктердің бірі және оны пайдалану оңай.

2) Компоненттер: компоненттер javascript функцияларына ұқсас, олар логиканы бірнеше рет қолданылатын тәуелсіз кодқа бөлу арқылы кодты жеңілдетуге көмектеседі. Біз компоненттерді функциялар ретінде және класстар ретінде қолдана аламыз. Сондай-ақ, компоненттердің күйі, тіректері бар, бұл жұысты жеңілдетеді. Класс ішінде әр қасиеттің күйі сақталады.

3) Виртуалды DOM: React виртуалды үй жасайды, яғни жадтағы деректердің кэш құрылымын жасайды.

4) Javascript өрнектері: JS өрнектерін JSX файлдарында {} сияқты жақшаларды қолдана отырып қолдануға болады.

Пайдаланушы интерфейсі үшін бізге Html қажет, ал dom-дағы әр элементте өңделетін оқиғалар, күй өзгерістері және т. б. болады. React жағдайында бұл Html және javascript-ті бір файлда қолдануға және dom-дағы күй өзгерістеріне қамқор болуға мүмкіндік береді.

React қолдану кезіндегі маңызды артықшылықтары:

– React жадтағы деректер құрылымының кэшін қолданатын виртуалды dom қолданады. Қосымшаны тез жасауға қабілеті бар.

– Компоненттерді өз қалауынша пайдалануға болады. Компоненттерді қайта шақыруға болады, сонымен қатар олар кодты сақтау кезінде пайдалы болуы мүмкін.

– Reactjs - бұл JavaScript кітапханасы, сондықтан оны бастау оңай.

– Facebook пен Instagram-да ReactJS қысқа уақыт ішінде өте танымал болды. Оны Apple, Netflix және т.б. сияқты көптеген танымал компаниялар пайдаланады.

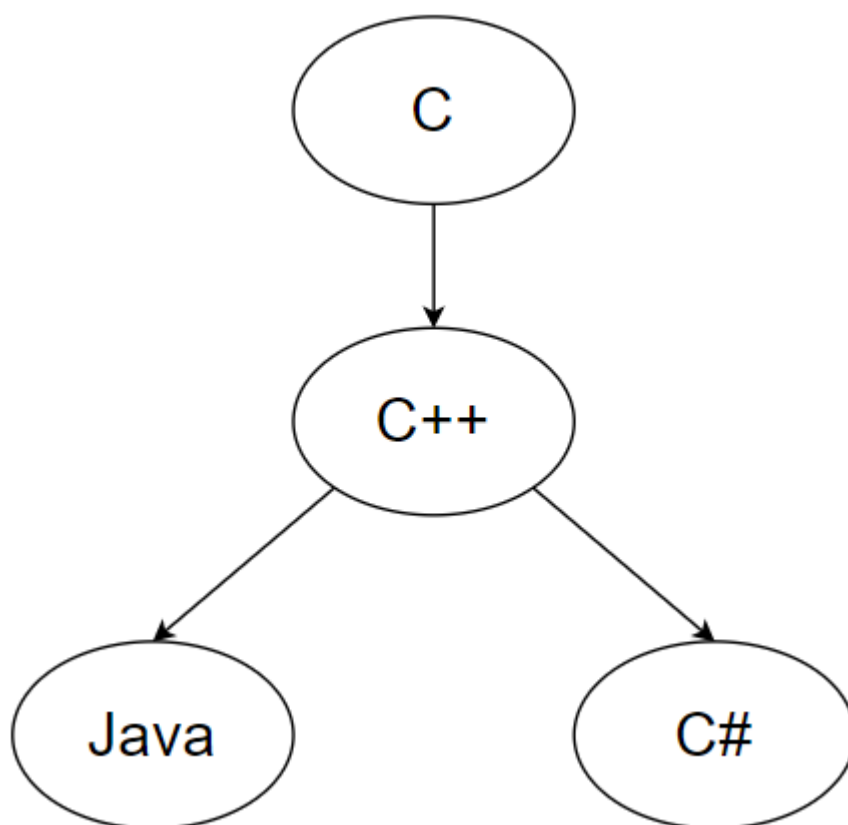
– Facebook ReactJS кітапханасын қолданады, сондықтан ол үнемі жаңартылып отырады.

– ReactJS-ті жұмыс үстелі мен мобильді қосымшалар үшін көп функциялы пайдаланушы интерфейсін жасау үшін пайдалануға болады.

– Күйін келтіру және тестілеу оңай, өйткені кодтың көп бөлігі Html емес, Javascript-те жасалады.

2.4 C# бағдарламау тілі

C# - Microsoft корпорациясының C ұрпағының көп парадигмалық бағдарламалау тілі. C# - бұл CIL-де құрастырылатын тіл, Windows, Mac OS және Linux-та орындалатын байт коды. C# тілі анықтап айтқанда, C++ баласы болып келеді. C тілінің немересі болып келеді. Ал Java ағайынды болады. Төменде көрсетілген суретте тілдердің иерархиясы суреттелді:



Сурет 1 - Бағдарламалау тілдерінің иерархиясы

Жоғарыда сызылған бағдарламалау тілдерінің бір-біріне қатысы бар екенін анықтай аламыз. Сол себепті `C#` тілі өзінің алдында құрылған тілдерінен керекті заттарын алып жеңілдетіп шығарылған. Яғни, тілдердің өзара қатынастары бар екені айқын.

`C#` бағдарламалау тілі өздігімен ауқатты болғанымен `.NET Framework` ортасымен тығыз байланысты. Бұл байланыс екі себеппен түсіндіріледі:

а) Бұл тіл бастапқыда `.NET Framework` ортасында орындалатын код болып жасалды;

ә) Ал екіншіден `C#`-тағы кітапханалар `.NET Framework` ортасына жасалған.

Тәжірибеде `C#` және `.NET Framework` бір-бірімен тығыз байланысты екенін білдіреді, бірақ теория жүзінде екеуін бөлек бөлуге болады.

2.5 Мәліметтер базасының құрылымы

Жобаның мәліметтер базасы PostgreSQL МББЖ арқылы құрылған. Біріншіден, біз PostgreSQL серверін және клиентін орнатуымыз керек. ОС үшін PostgreSQL-дің тиісті нұсқасын жүктеп, қол жетімді жаңа нұсқаны таңдаңыз.

Орнату кезінде сізден pgAdmin (клиент) таңдау сұралады. PostgreSQL үшін бірнеше жақсы клиенттер бар, бірақ pgAdmin бастама үшін жақсы нұсқа. Жаңадан жұмыс істеп жатқан адамға өте тиімді шешім, үйренуге тез.

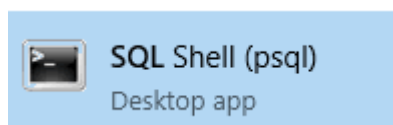
Сондай-ақ, сізден пайдаланушының паролін (postgres) енгізу сұралады. Бұл пароль негізгі пароль болып табылады және оны есте сақтап, жазып алу керек. Шатаспау үшін сіз әдепкі PostgreSQL порты болып табылатын портын

қалдыра аласыз. Егер сізде компьютерде PostgreSQL сервері болса, басқасын таңдауға болады.

Мәліметтер базасының құру үшін тиесілі кодты жазуымыз қажет:

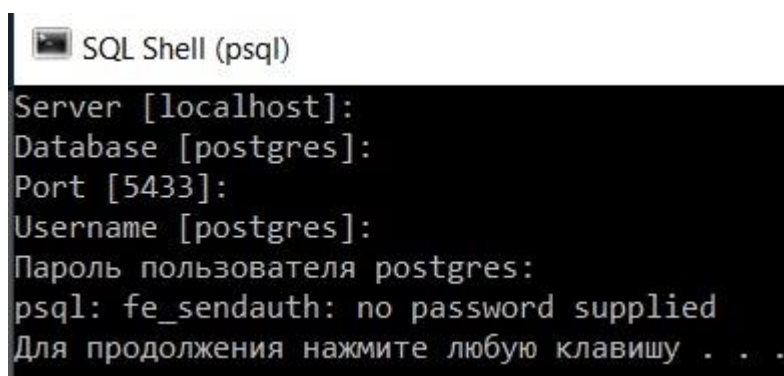
CREATE DATABASE МБаты.

Кодты жазу үшін мәліметтер базасын құру пәрмен жолын ашамыз.



Сурет 2 - Мәліметтер базасын құру пәрмен жолы

Дерекқорға қосылу үшін Enter пернесін бес рет басуы керек.



Сурет 3 - МБ-на қосылу

Бейнеленген код жолына МБ құру кодын жазып базаны құра аламыз. Осы жұмысты pgAdmin арқылы орындасақта болады.

PostgreSQL-де мәліметтер базасын құруға арналған толық синтаксис:

CREATE DATABASE dbName

OWNER = roleName

TEMPLATE = template

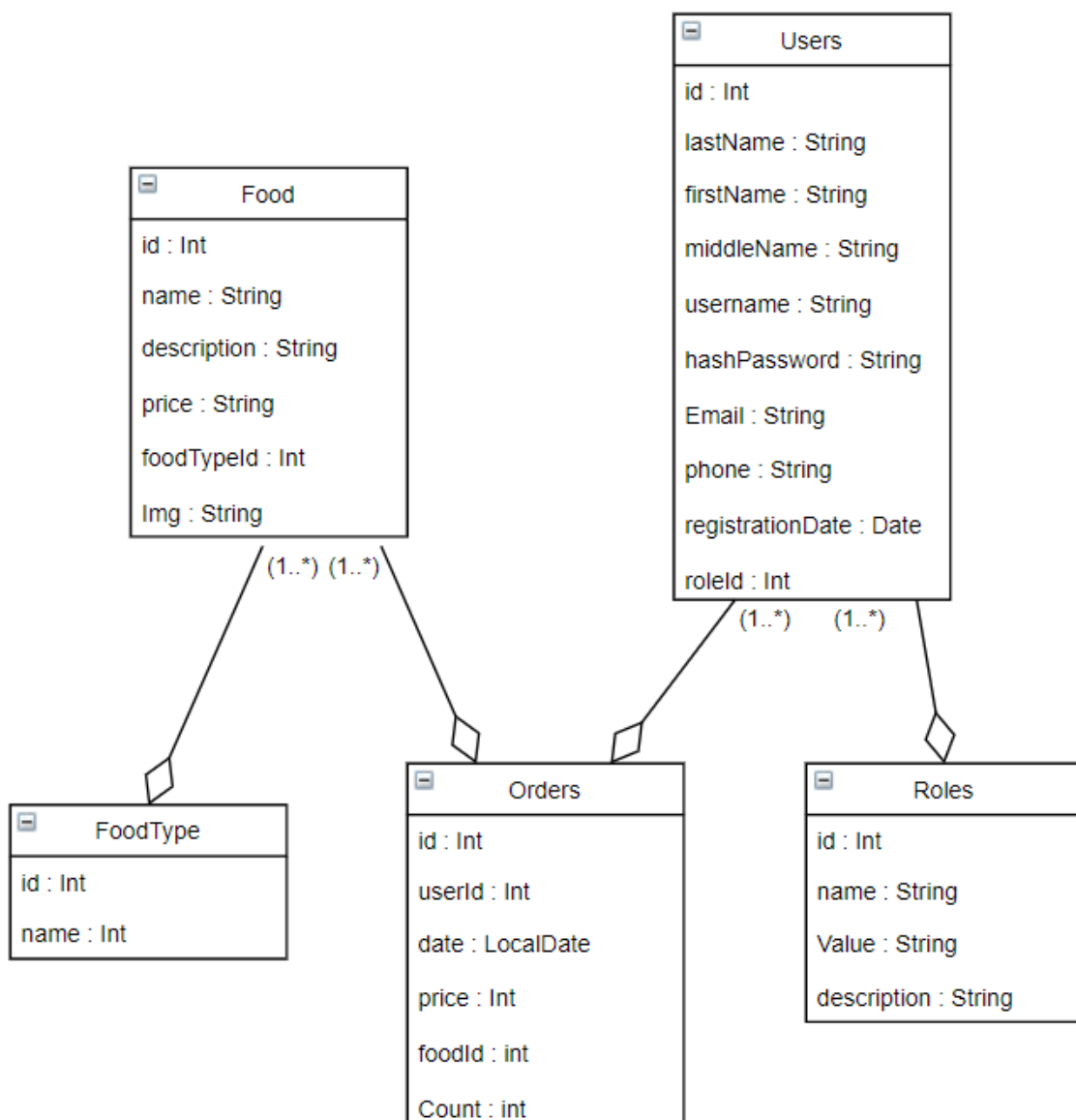
ENCODING = encoding

LC_COLLATE = collate

LC_CTYPE = CTypee

TABLESPACE = tableSpaceName

CONNECTION LIMIT = concurrentConnection



Сурет 4 - Мәліметтер базасының ег диаграммасы

4-суретте құрылған мәліметтер базасы кестелерінің байланысы бейнеленген. МБ-да бес кесте құрылған. Әр кестеде өзіне тиесілі деректер мен сол деректердің түрлері жазылған.

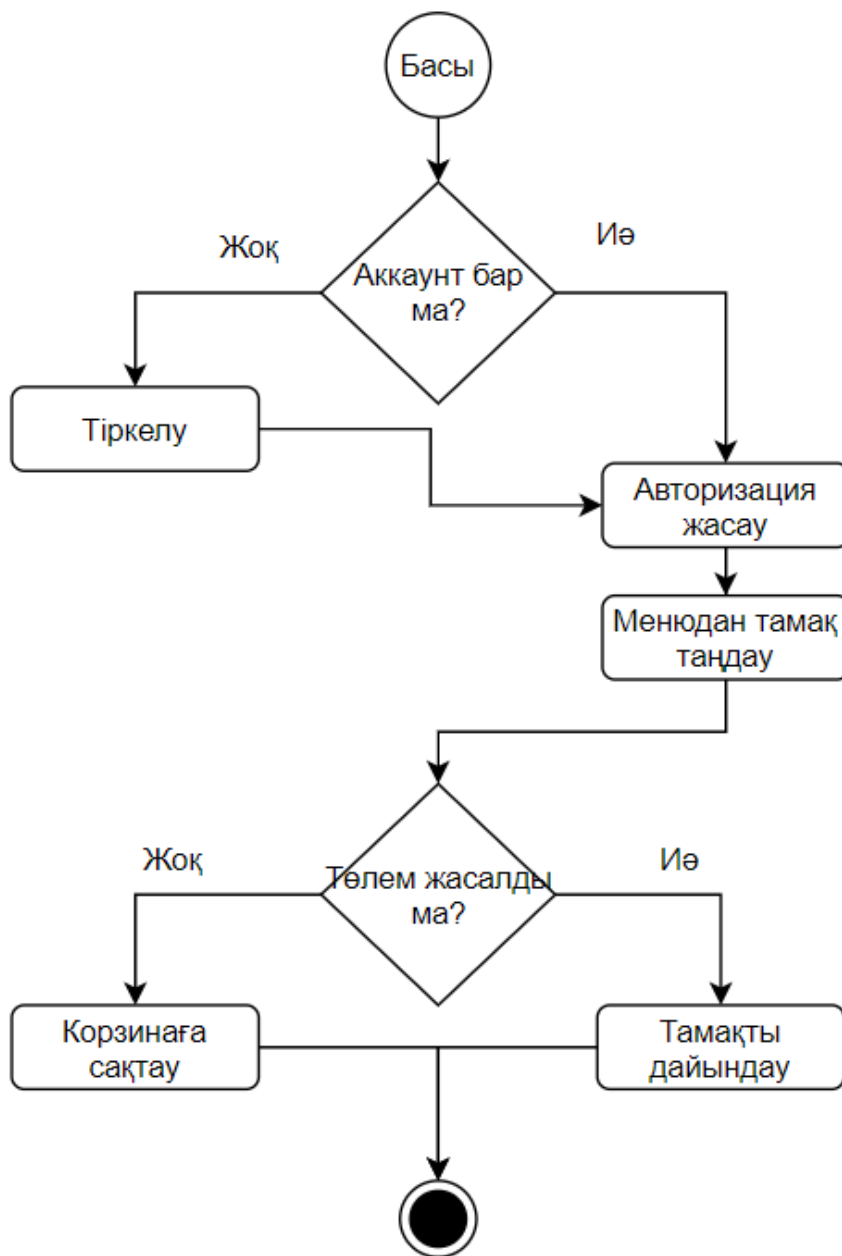
ER диаграммасы - бұл мәліметтер базасын жобалаудың графикалық тәсілі. Бұл мәліметтер элементтерін және олардың көрсетілген бағдарламалық жүйеге қатынасын анықтайтын жоғары деңгейлі деректер моделі. ER моделі нақты объектілерді көрсету үшін қолданылады.

2.6 Іс-әрекеттер диаграммасы

Іс-әрекеттер диаграммасы - бұл іс-әрекеттің бірінші қадамның келесіне ауысуды модельдейтін ағын схемасының кеңейтілген нұсқасы.

Процестерді модельдеуде іс-әрекеттер диаграммалары жиі қолданылады. Олар сонымен қатар пайдалану жағдайларының қадамдарын сипаттай алады.

Модельденген әрекеттер жүйелі және бір мезгілді болуы мүмкін. Екі жағдайда да іс-әрекеттер диаграммасында басталуы (бастапқы күй) және аяқталуы (соңғы күй) болады.



Сурет 5 - Іс-әрекеттер диаграммасының бейнесі

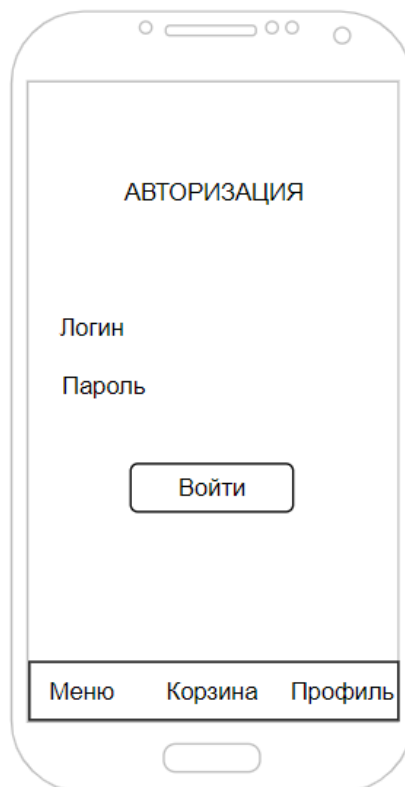
2.7 Жоба прототипі

Веб-дизайн тұрғысынан прототип-бұл сіздің веб-дизайныңыздың интерактивті орналасуы. Ол веб-сайтты түпкілікті құрастырудың барлық істеріне ие болмауы мүмкін, бірақ ол сізге функционалдылық, пайдаланушы жолы және веб-қосымша арқылы ағым туралы жақсы түсінік бере алады. Веб-қосымшаның прототипі – бұл шын мәнінде қосымшаның көрнекі нұсқасы болып

келеді, ол экрандарды байланыстыруға және веб-қосымшаны жасамас бұрын қалай жұмыс істейтінін көрсетуге мүмкіндік береді.

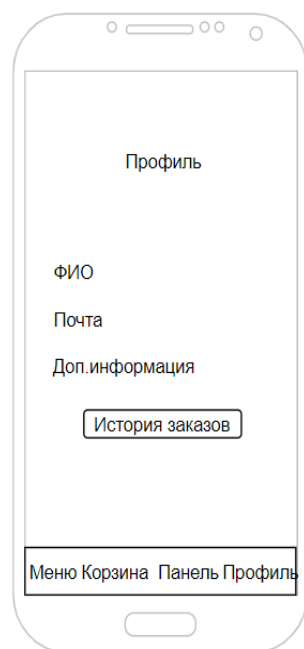
Прототиптеу жобалау процесінде өте пайдалы, өйткені веб-жобаның алғашқы кезеңдерінде көп уақытты үнемдеуге мүмкіндік береді. Жобалау кезеңінде жетілдіруді қажет ететін аймақтарды анықтауынан кейін сайт дамуға көшкеннен кейін осы мәселелерді анықтаудан гөрі оңайырақ. Прототиптеу сонымен қатар қосымшаның интерактивті нұсқасы сияқты қарапайым визуалды түсінбеуі мүмкін пайдаланушыларға көрсету үшін пайдалы.

Веб-қосымшаның прототипін жасауға көмектесетін көптеген құралдар бар. Прототип салынған қосымша - draw.io онлайн визуалдау құралы.



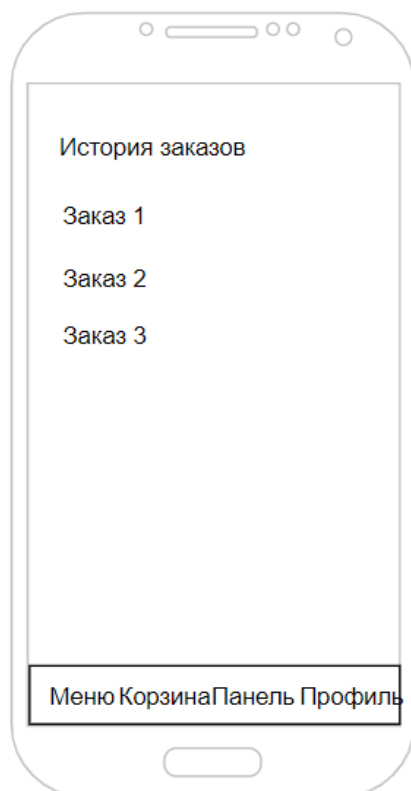
Сурет 6 - Кіру экранының прототипі

6-суретте қолданушының кіру парақшасы суреттелген. Кіру экранында логин мен пароль мәліметтерін жазуға арналған енгізетін жолдар жасалу керек. Мәліметтер толтырылғаннан кейін “кіру” батырмасын басып қосымшаға кіру мүмкіндігі болуы қажет. Төменде навбар сызылған. Навбарда қолданушы тез арада басқа парақшаларға өту үшін навигациялық батырмалар құрастырылуы керек.



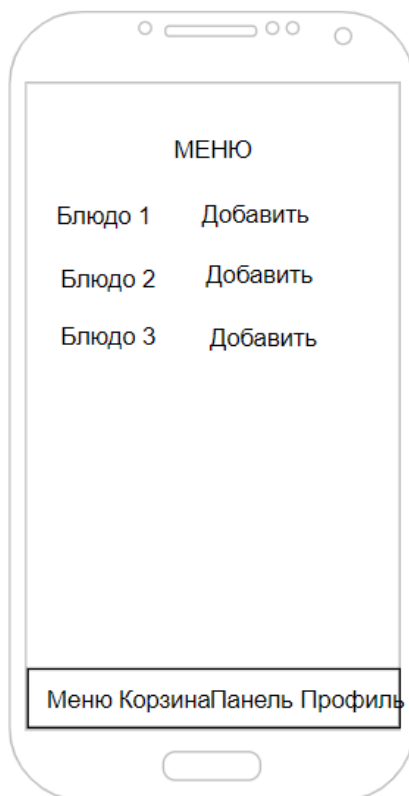
Сурет 7 - Қолданушы парақшасының прототипі

Қолданушы парақшасында қолданушыға тиесілі мәліметтер жазылып тұруы керек. Ол мәліметтер тіркелу уақытында немесе жаңарту уақытында енгізуге мүмкіндік болғаны тиіс. Қажет болған жағдайда қолданушы жайлы жазылған деректер парақшасынан тапсырыстар парақшасына өтуге болатындай батырма жасалады.



Сурет 8 - Тапсырыстар тарихының прототипі

Тапсырыстар аты мен нөмірлері жазылады.



Сурет 9 - Тамақтар көрсетілген мәзірінің прототипі

Меню - бұл пайдаланушыға ақпаратты табуға немесе бағдарлама функциясын орындауға көмектесетін компьютерлік қосымшаның пайдаланушысына ұсынылатын опциялар жиынтығы. Менюлар графикалық пайдаланушы интерфейстерінде жиі кездеседі (GUI).

Менюда екі негізге батырма болады. Ол батырмалар арқылы таңдалынған тамақты себетке салуға мүмкіндік болу керек. Тамақ жайлы қосымша ақпараттар да болуы мүмкін.

3 ЖОБА ЖҰМЫСЫНЫҢ СИПАТТАМАСЫ

3.1 Графикалық пайдаланушы интерфейсi

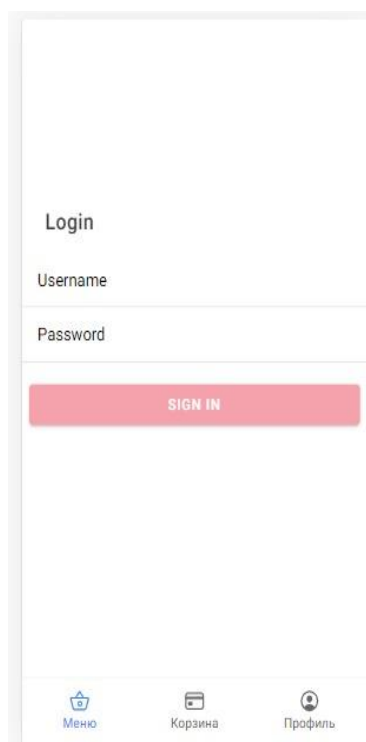
GUI - бұл пайдаланушыға машинамен өзара әрекеттесуді жеңілдету үшін берілген байланыстың визуалды көрінісі болып табылатын графикалық интерфейс. GUI графикалық пайдаланушы интерфейсін білдіреді. Бұл батырмалар мен белгішелер сияқты графикалық көріністі қамтитын жалпы пайдаланушы интерфейсi және байланыс қарапайым мәтіндік немесе командалық өзара әрекеттесу арқылы емес, сол белгішелермен өзара әрекеттесу арқылы жүзеге асырылуы мүмкін.

Графикалық пайдаланушы интерфейсiнің артықшылықтары:

- қарапайымдылық;
- көзге тартымды және кез-келген адамды машинамен жұмыс істеуге мәжбүр етеді;
- тіпті компьютерлік білімі жоқ жігіт компьютерді қолдана алады және негізгі функцияларды орындай алады. Бұл үшін GUI жауап береді;
- іздеу өте қарапайым болады, өйткені графикалық интерфейс файлдардың визуалды көрінісін және олар туралы егжей-тегжейлі ақпарат береді;
- компьютердің әр жауабы графикалық интерфейс арқылы визуалды түрде беріледі;
- компьютерлік білімі жоқ қолданушы графикалық интерфейс арқылы машинаны үйренуге кірісе алады, өйткені ол пайдаланушыларға үйренуге мүмкіндік береді және анықтауға мүмкіндік береді;
- пайдаланушы компьютерді немесе смартфонды интерфейссіз қолдана бастаса, онда ол әр тапсырманы орындау үшін машинаға командалар беруі керек. Қандай да бір мағынада, пайдаланушы бағдарламалау туралы қандай да бір білімге ие болуы керек.

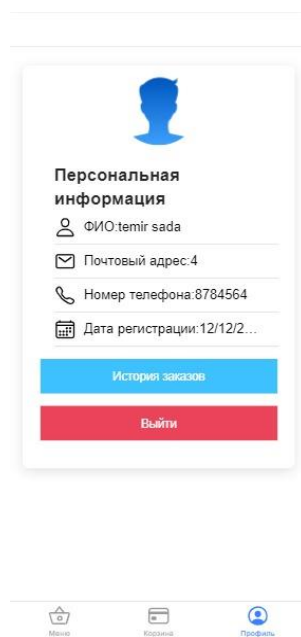
3.2 Веб-қосымша жұмысы

Веб-қосымша қолданыста қарапайым, ыңғайлы, түсінікті болуы үшін React әзірлеу ортасы қолданылды.



Сурет 10 - Авторизация экранының интерфейсі

Қосымша интерфейсі прототипте көрсетілгендей құралды. Қарапайым және көзге жағымды түс палитралары таңдалды. Менюдағы тағамдарды көріп, таңдау үшін авторизация парақшасы арқылы жүйге кіру керек.



Сурет 11 - Қолданушы парақшасы

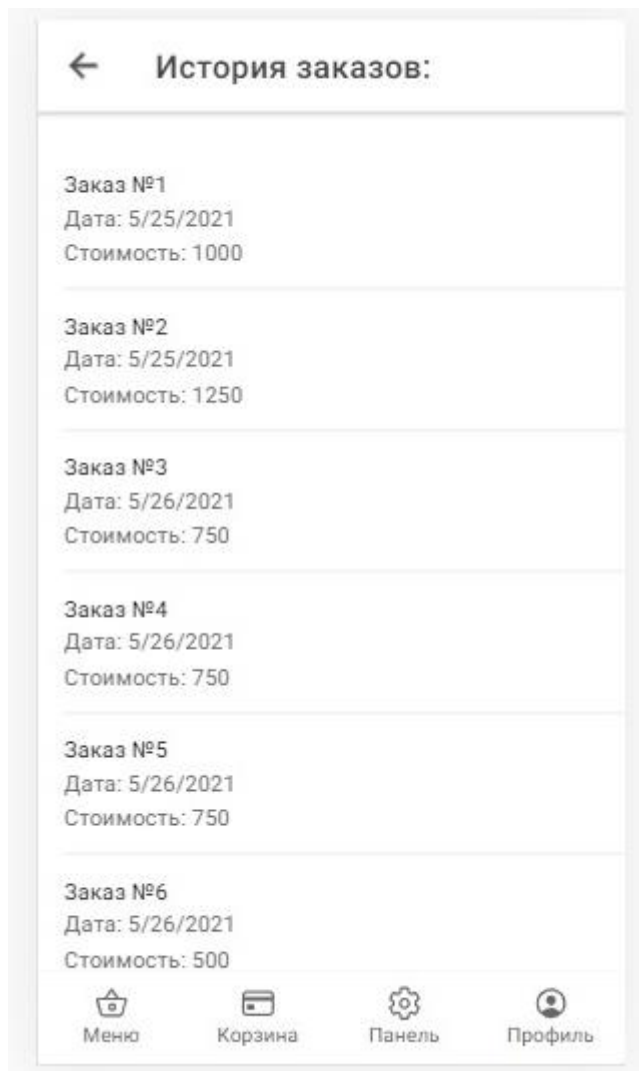
Қолдану парақшасында төмендегі мәліметтер көрсетіледі:

- аты-жөні;
- e-mail адресі;
- телефон;
- тіркелген уақыты.

Олардан бөлек екі батырма бар, олар:

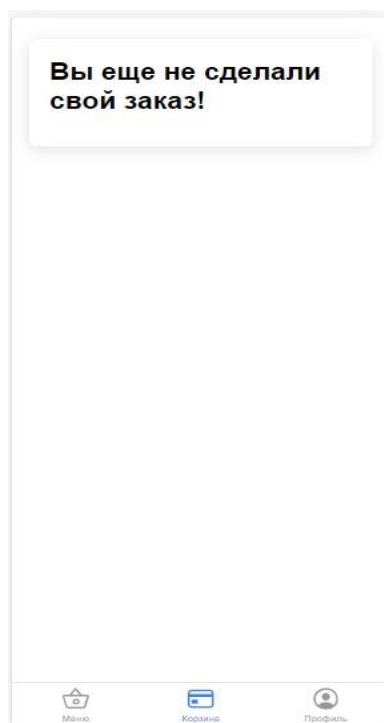
- тапсырыстар тарихы;
- жүйеден шығу.

Қосымшаның мақсаттарының бірі қарапайымдылық болғандықтан, керек емес заттарды қоспау жөн шешім болды.



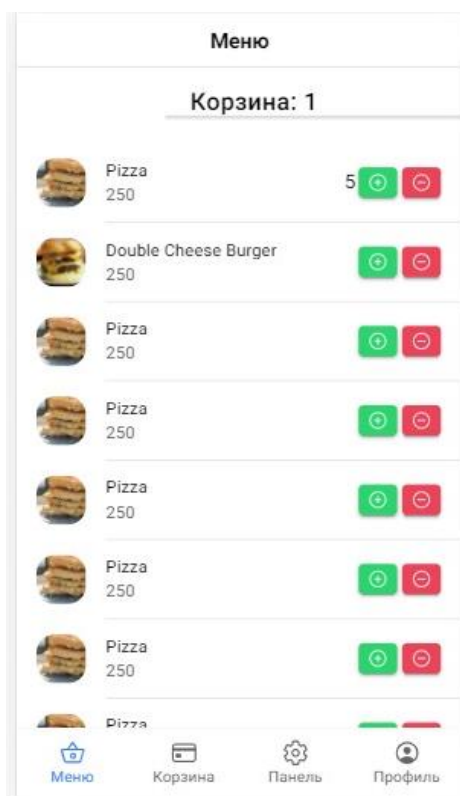
Сурет 12 - Тапсырыстар тарихы

12-суретте көрсетілген парақшада қолданушы тапсырыстарының тарихы бейнеленген. Тарихында тапсырыс нөмірі, датасы, және бағасы жазылған.



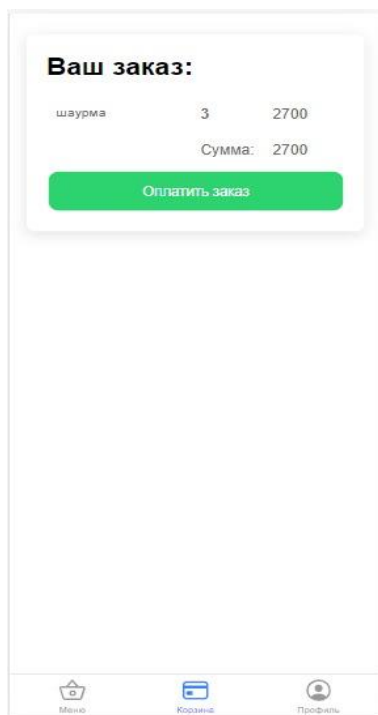
Сурет 13 - Тапсырыс қоржыны

Тапсырыс қоржыны - бұл пайдаланушы сатып алатын тағамдарды қоса алатын интерфейс. Бұл парақша арқылы қоржынға қандай тағамдар салынғанын көруге мүмкіндік бар. Осы парақша арқылы тағамдар салып алынады. Қоржынға бірде-бір тағам қосылмаса 13-суреттегідей хабарлама шығып тұрады.



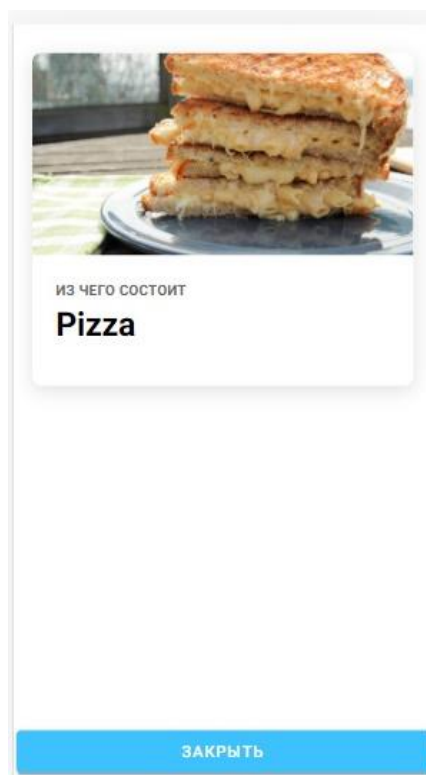
Сурет 14 - Ас мәзірі

Ас мәзірінде тағамдар аттары мен бағасы көрінеді. Өр тағам жанында қосуға немесе алуға болатын батырмалар жасалған. Батырмалар көмегімен тағамның санын өзгертуге мүмкін. Өзгерткен уақытта сол жолда қанша тағам саны қосылғаны көрініп тұрады. Қосқан жағдайда ол қоржынға тамақтың қанша түрі енгізілгені экранның жоғары жағында шығады.

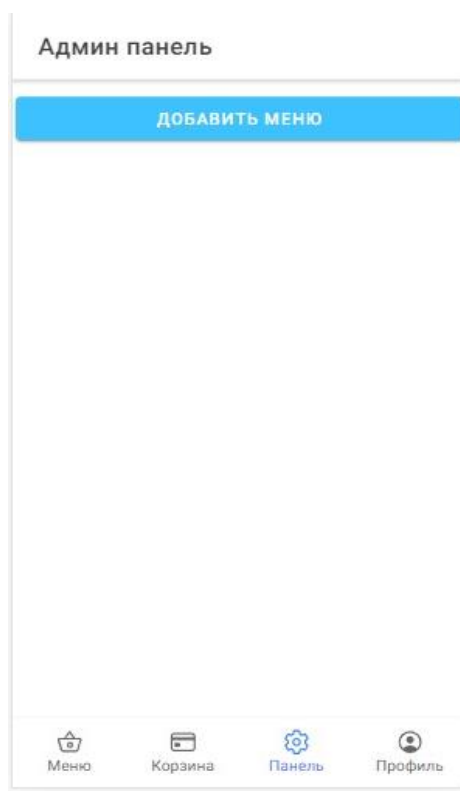


Сурет 15 - Қоржынға қосылған тағам

Қоржынға қосылған тағам 15-суретте көрсетілді. Қосылғаннан кейін 13-суретте бейнеленгендей хабарлама шықпайды.

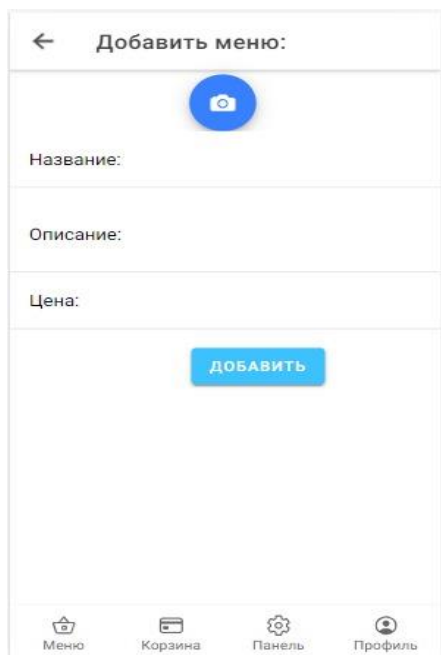


Сурет 16 - Тағамның неден жасалғаны жайлы ақпарат парақшасы
Ингредиенттер 16-суреттегі парақшада жазылып тұрады.



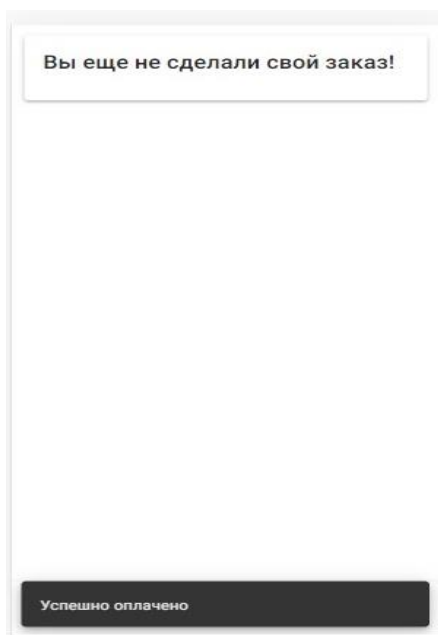
Сурет 17 - Админ панель

Админ панельде администратордың мәзірге тағам қосу мүмкіншілігі бар. Админ жүйеге кіргеннен кейін, навбарда жаңа навигациялық батырма шығады. Ол батырма көмегімен админ ас мәзірін өзгертуге мүмкіншілік алады.



Сурет 18 - Мәзірге тағам қосу парақшасы

Мәзірге тағамдарды қосу барысында, әр тағам жайлы ақпараттарын толтыру керек. Сол себепті 18-суретте ақпарат енгізу жолдары қосылған. Әр тағамның бейнесін салуға болатын жеке батырма жасалған.



Сурет 19 - Төлем жасалғаннан кейінгі хабарлама

Веб-қосымша өзінің бойында тиімді модульдар жиынтығын жинаған. Әр адам қосымшаны қолдануын тез арада үйрене алады.

ҚОРЫТЫНДЫ

Әзірленген веб-қосымша мектеп оқушылары мен асхана қызметкерлерінің жасалатын физикалық артық жұмыстарын жеңілдету үшін әзірленген.

Әзірлену құралдарына .NET Core, PostgreSQL, React кіреді. Бұл құралдар қазіргі уақытта өте тиімді және үлкен тәжірибені талап етпейді.

Жұмыстың мақсаты мектеп асханасының жұмысын автоматтандыру болды. Автоматтандыру әсері оқушы уақытын тиімді пайдалануына алып келеді.

Жобаны прототиптеу әзірленген веб-қосымшаға зор үлесін қосты деп санаймын. Прототиптеу онлайн құралымен жүзеге асты.

Қосымша арқылы оқушылар тамақ жайлы жаңа ақпараттар алуына болады. Асханамен бірлесе оқушылардың өз мәзірін құруына болады.

ПАЙДАЛАНЫЛҒАН ӘДЕБИЕТТЕР ТІЗІМІ

- 1 Мектепте дұрыс тамақтану [Электрондық ресурс]. – Ресурсқа сілтеме: https://www.cdc.gov/healthyschools/features/eating_healthier.htm
- 2 Фил Джапиксе, Эндрю Троелсен. Pro C# 8 with .NET Core 3: Foundational Principles and Practices in Programming, 2020. – 78 с.
- 3 Эндрю Стеллмен. Head First C#, 2007. – 103 с.
- 4 Technical questions about .NET Core [Электрондық ресурс]. – Ресурсқа сілтеме: <https://docs.microsoft.com/en-us/answers/topics/dotnet-runtime-core.html>
- 5 PostgreSQL ресми сайты [Электрондық ресурс]. – Ресурсқа сілтеме: <https://www.postgresql.org/about/>
- 6 React answers [Электрондық ресурс]. – Ресурсқа сілтеме: <https://www.reddit.com/r/reactjs/>
- 7 Грег Лим. Beginning React, 2017. – 39 с.
- 8 SQL Forum [Электрондық ресурс]. – Ресурсқа сілтеме: <https://www.sql.ru/forum/postgresql>

Приложение А

Веб-қосымша листингі

loginController.cs листингі

```
using Microsoft.AspNetCore.Http;
using Microsoft.AspNetCore.Mvc;
using Microsoft.Extensions.Configuration;
using Microsoft.IdentityModel.Tokens;
using SchoolCanteen.Models;
using System;
using System.Collections.Generic;
using System.IdentityModel.Tokens.Jwt;
using System.Linq;
using System.Security.Claims;
using System.Text;
using System.Threading.Tasks;
namespace SchoolCanteen.Controllers
{
    [Route("api/[controller]")]
    [ApiController]
    public class LoginController : ControllerBase
    {
        private readonly DbConnection _db;
        private readonly IConfiguration _config;
        public LoginController(DbConnection dbContext,
            IConfiguration config)
        {
            _db = dbContext;
            _config = config;
        }
        [HttpPost]
        public IActionResult Login([FromBody] Users loggedUser)
        {
            IActionResult response = Unauthorized();
            UserModel userModel = new UserModel();
            var user = Authenticate(loggedUser);
            if (user != null)
            {
                var token = GenerateJWTToken(user);
                response = Ok(new { token });
            }
            return response;
        }
    }
}
```


Продолжение приложения А

```
private UserModel Authenticate(Users loggedUser)
{
    try
    {
        UserModel user = null;
        if (_db.Users.Any(m => m.Username == loggedUser.Username.ToLower()
&& m.HashPassword == loggedUser.HashPassword))
        {
            Users u = _db.Users.FirstOrDefault(m => m.Username ==
loggedUser.Username.ToLower() && m.HashPassword ==
loggedUser.HashPassword);
            int? rolesId = u.RoleId;
            string role = _db.Roles.FirstOrDefault(m => m.Id == rolesId).Value;
            _db.SaveChanges();
            user = new UserModel()
            {
                Id = u.Id,
                Email = u.Email,
                Username = u.Username,
                RoleName = u.Roles.Name,
                Roles = u.Roles
            };
        }
        return user;
    }
    catch (Exception ex)
    {
        return null;
    }
}

private string GenerateJWTToken(UserModel user)
{
    var securityKey = new
SymmetricSecurityKey(Encoding.UTF8.GetBytes(_config["AuthOptions:SignInKey
"]));
    var credentials = new SigningCredentials(securityKey,
SecurityAlgorithms.HmacSha256);
    var claims = new[] {
        new Claim(JwtRegisteredClaimNames.Sub, user.Username),
        new Claim(JwtRegisteredClaimNames.Email, user.Email),
        new Claim(JwtRegisteredClaimNames.NameId, user.Id.ToString()),
        new Claim(JwtRegisteredClaimNames.Jti, Guid.NewGuid().ToString()),
    }
```

Продолжение приложения А

```
        new Claim(ClaimsIdentity.DefaultRoleClaimType,user.RoleName),
    };
    var token = new JwtSecurityToken(
        issuer: _config["AuthOptions:Issuer"],
        audience: _config["AuthOptions:Issuer"],
        claims,
        expires: DateTime.Now.AddMinutes(480),
        signingCredentials: credentials
    );
    var encodeToken = new JwtSecurityTokenHandler().WriteToken(token);
    return encodeToken;
    }
}
}
```

orderController.cs листингі

```
using Microsoft.AspNetCore.Http;
using Microsoft.AspNetCore.Mvc;
using Microsoft.Extensions.Configuration;
using Microsoft.Net.Http.Headers;
using SchoolCanteen.Models;
using SchoolCanteen.Models.Utils;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
namespace SchoolCanteen.Controllers
{
    [Route("api/[controller]")]
    [ApiController]
    public class OrdersController : ControllerBase
    {
        private readonly DbConnection _db;
        private readonly IConfiguration _config;
        public OrdersController(DbConnection dbContext, IConfiguration config)
        {
            _db = dbContext;
            _config = config;
        }
        [HttpPost]
        public IActionResult SaveOrder(List<Orders> ordersRequest)
        {
```

Продолжение приложения А

```
var orders = new List<Orders>();
var token =
Request.Headers[HeaderNames.Authorization].ToString().Replace("Bearer ", "");
var login = JWTHelper.GetUserLogin(token,
_config["AuthOptions:SignInKey"]);
var currentUser = _db.Users.SingleOrDefault(u => u.Username == login);
foreach (var order in ordersRequest)
{
    Orders newOrder = new Orders();
    newOrder.Date = DateTime.Now;
    newOrder.FoodId = order.FoodId;
    newOrder.Price = order.Price;
    newOrder.Count = order.Count;
    newOrder.UserId = currentUser.Id;
    orders.Add(newOrder);
}
_db.Orders.AddRange(orders);
_db.SaveChanges();
return Ok();
}
[HttpGet("MyOrders")]
public IActionResult GetMyOrders()
{
    var token =
Request.Headers[HeaderNames.Authorization].ToString().Replace("Bearer ", "");
var login = JWTHelper.GetUserLogin(token,
_config["AuthOptions:SignInKey"]);
var currentUser = _db.Users.SingleOrDefault(u => u.Username == login);
var myOrders = _db.Orders.Where(o => o.UserId == currentUser.Id);
return Ok(myOrders);
}
}
}
```

UserController.cs листинг1

```
using Microsoft.AspNetCore.Authorization;
using Microsoft.AspNetCore.Http;
using Microsoft.AspNetCore.Mvc;
using Microsoft.Extensions.Configuration;
using Microsoft.Net.Http.Headers;
using SchoolCanteen.Models;
```

Продолжение приложения А

```
using SchoolCanteen.Models.Utils;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
namespace SchoolCanteen.Controllers
{
    [Authorize]
    [Route("api/[controller]")]
    [ApiController]
    public class UserController : ControllerBase
    {
        private readonly DbConnection _db;
        private readonly IConfiguration _config;
        public UserController(DbConnection dbContext, IConfiguration config)
        {
            _db = dbContext;
            _config = config;
        }
        [HttpGet]
        [Authorize(Roles = "Administrator")]
        public IEnumerable<Users> GetUsers()
        {
            var token =
Request.Headers[HeaderNames.Authorization].ToString().Replace("Bearer ", "");
            var role = JWTHelper.GetUserRole(token,
_config["AuthOptions:SignInKey"]);
            var login = JWTHelper.GetUserLogin(token,
_config["AuthOptions:SignInKey"]);
            return _db.Users.ToList();
        }
        [Authorize]
        [HttpGet("GetLoggedUser")]
        public IActionResult GetLoggedUser()
        {
            var token =
Request.Headers[HeaderNames.Authorization].ToString().Replace("Bearer ", "");
            var login = JWTHelper.GetUserLogin(token,
_config["AuthOptions:SignInKey"]);
            return Ok(_db.Users.SingleOrDefault(u => u.Username == login));
        }
        [Authorize]
```

Продолжение приложения А

```
[HttpGet("GetMyRole")]
public IActionResult GetMyRole()
{
    var token =
Request.Headers[HeaderNames.Authorization].ToString().Replace("Bearer ", "");
    var role = JWTHelper.GetUserRole(token,
_config["AuthOptions:SignInKey"]);
    return Ok(role);
}
}
}
```

foodController.cs листинг

```
using Microsoft.AspNetCore.Http;
using Microsoft.AspNetCore.Mvc;
using Microsoft.EntityFrameworkCore;
using SchoolCanteen.Controllers.Foods;
using SchoolCanteen.Models;
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Threading.Tasks;
namespace SchoolCanteen.Controllers
{
    [Route("api/[controller]")]
    [ApiController]
    public class FoodController : ControllerBase
    {
        private DbConnection _db;
        public FoodController(DbConnection dbContext)
        {
            _db = dbContext;
        }
        [HttpGet]
        public IEnumerable<Food> GetFoods()
        {
            return _db.Food.Include(c => c.FoodType).ToList();
        }
        [HttpGet]
        [Route("getAll")]
    }
```

Продолжение приложения А

```
public IEnumerable<Food> GetAllFoods() => _db.Food.ToList();
[HttpPost]
public async Task<IActionResult> CreateFoodAsync(Food foodsRequest)
{
    _db.Food.Add(foodsRequest);
    _db.SaveChanges();
    return Ok();
}
[HttpDelete("{id}")]
public IActionResult DeleteFood(int foodId)
{
    var food = _db.Food.SingleOrDefault(f => f.Id == foodId);
    _db.Food.Remove(food);
    _db.SaveChanges();
    return Ok();
}
[HttpPut]
public IActionResult UpdateFood(Food updatedFood)
{
    _db.Food.Update(updatedFood);
    _db.SaveChanges();
    return Ok();
}
private async Task<string> saveFile(IFormFile file)
{
    byte[] fileBytes;
    using (var memoryStream = new MemoryStream())
    {
        await file.CopyToAsync(memoryStream);
        fileBytes = memoryStream.ToArray();
    }
    var content = Convert.ToBase64String(fileBytes);
    return content;
}
}
```

foodRequest.cs листинг

```
using Microsoft.AspNetCore.Http;
using Microsoft.AspNetCore.Mvc;
using Microsoft.AspNetCore.Mvc.ModelBinding;
using Newtonsoft.Json;
```

Продолжение приложения А

```
using SchoolCanteen.Models;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
namespace SchoolCanteen.Controllers.Foods
{
    public class FoodsRequest
    {
        [JsonProperty(PropertyName = "file")]
        public IFormFile File { get; set; }
        [ModelBinder(BinderType = typeof(FormDataJsonBinder))]
        public Food Food { get; set; }
    }
    public class FormDataJsonBinder : IModelBinder
    {
        public Task BindModelAsync(ModelBindingContext bindingContext)
        {
            if (bindingContext == null)
            {
                throw new ArgumentNullException(nameof(bindingContext));
            }
            string fieldName = bindingContext.FieldName;
            var valueProviderResult =
bindingContext.ValueProvider.GetValue(fieldName);
            if (valueProviderResult == ValueProviderResult.None)
            {
                return Task.CompletedTask;
            }
            else
            {
                bindingContext.ModelState.SetModelValue(fieldName,
valueProviderResult);
            }
            string value = valueProviderResult.FirstValue;
            if (string.IsNullOrEmpty(value))
            {
                return Task.CompletedTask;
            }
            try
            {

```

Продолжение приложения А

```
        object result = JsonConvert.DeserializeObject(value,
bindingContext.ModelType);
        bindingContext.Result = ModelBindingResult.Success(result);
    }
    catch (JsonException)
    {
        bindingContext.Result = ModelBindingResult.Failed();
    }
    return Task.CompletedTask;
}
}
```

foodTypeController.cs листинг

```
using Microsoft.AspNetCore.Http;
using Microsoft.AspNetCore.Mvc;
using SchoolCanteen.Models;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
namespace SchoolCanteen.Controllers
{
    [Route("api/[controller]")]
    [ApiController]
    public class FoodTypeController : ControllerBase
    {
        private DbConnection _db;
        public FoodTypeController(DbConnection dbContext)
        {
            _db = dbContext;
        }
        [HttpGet]
        public IEnumerable<FoodType> GetFoodTypes()
        {
            return _db.FoodType.ToList();
        }
        [HttpPost]
        public IActionResult CreateFoodType(FoodType foodType)
        {
            _db.FoodType.Add(foodType);
            _db.SaveChanges();
            return Ok();
        }
    }
}
```



```
}
[HttpDelete("{id}")]
public IActionResult DeleteFoodType(int foodTypeId)
{
    var foodType = _db.FoodType.SingleOrDefault(f => f.Id == foodTypeId);
    _db.FoodType.Remove(foodType);
    _db.SaveChanges();
    return Ok();
}
[HttpPut]
public IActionResult UpdateFoodType(FoodType updatedFoodType)
{
    _db.FoodType.Update(updatedFoodType);
    _db.SaveChanges();
    return Ok();
}
}
}
```

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

Сатпаев университет

ОТЗЫВ

НАУЧНОГО РУКОВОДИТЕЛЯ

Дипломная работа

Ыдырыс Нұрсұлтан Сәкенұлы

5B070300-Информационные системы

Тема: Мектеп асханасындағы менюді қашықтықтан көре алу және тапсырыс беру үшін мобильді веб-қосымшасын әзірлеу.

Дипломная работа представляет собой выпускную квалификационную работу по специальности 5B070300-«Информационные системы». Пояснительная записка состоит из введения, из введения, 3 глав, заключения, списка использованных источников и приложения.

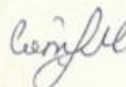
Автор дипломной работы все поставленные задачи выполнил и показал себя очень ответственным и коммуникабельным студентом.

Дипломная работа выполнена на достаточном профессиональном уровне и содержит все необходимые сведения для такого рода работ.

Считаю, что дипломная работа соответствует требованиям, предъявляемым к выпускным квалификационным работам по специальности 5B070300-«Информационные системы». Автор работы Ыдырыс Нұрсұлтан Сәкенұлы заслуживает присвоения академической степени бакалавра.

Научный руководитель

Доктор PhD



Сатымбеков М.Н.

« 03 » 06 2021

Активаци
Чтобы актив
"Параметры"



Метаданные

Название

Мектеп асханасындағы менюді Қашықтықтан көре алу және тапсырыс беру үшін мобильді веб-Қосымшасын Әзірлеу

Автор

Ыдырыс Нұрсұлтан

Научный руководитель

Максатбек Сатымбеков

Подразделение

ИКИИТ

Список возможных попыток манипуляций с текстом

В этом разделе вы найдете информацию, касающуюся манипуляций в тексте, с целью изменить результаты проверки. Для того, кто оценивает работу на бумажном носителе или в электронном формате, манипуляции могут быть невидимы (может быть также целенаправленное вписывание ошибок). Следует оценить, являются ли изменения преднамеренными или нет.

Замена букв		0
Интервалы		0
Микропробелы		0
Белые знаки		0
Парафразы (SmartMarks)		0

Объем найденных подоби

Обратите внимание! Высокие значения коэффициентов не означают плагиат. Отчет должен быть проанализирован экспертом.



КП1

25

Длина фразы для коэффициента подоби 2



КП2

2870

Количество слов



КЦ

23352

Количество символов

Подобия по списку источников

Просмотрите список и проанализируйте, в особенности, те фрагменты, которые превышают КП №2 (выделенные жирным шрифтом). Используйте ссылку «Обозначить фрагмент» и обратите внимание на то, являются ли выделенные фрагменты повторяющимися короткими фразами, разбросанными в документе (совпадающие сходства), многочисленными короткими фразами расположенные рядом друг с другом (парафразирование) или обширными фрагментами без указания источника ("криптоцитаты").

10 самых длинных фраз

Цвет текста

ПОРЯДКОВЫЙ НОМЕР	НАЗВАНИЕ И АДРЕС ИСТОЧНИКА URL (НАЗВАНИЕ БАЗЫ)	КОЛИЧЕСТВО ИДЕНТИЧНЫХ СЛОВ (ФРАГМЕНТОВ)
------------------	--	---

из базы данных RefBooks (0.00 %)

ПОРЯДКОВЫЙ НОМЕР	НАЗВАНИЕ	КОЛИЧЕСТВО ИДЕНТИЧНЫХ СЛОВ (ФРАГМЕНТОВ)
------------------	----------	---

Подбор по списку источников

Посмотрите список и проанализируйте, в особенности, те фрагменты, которые превышают КП №2 (выделенные жирным шрифтом). Используйте ссылку «Обозначить фрагмент» и обратите внимание на то, являются ли выделенные фрагменты повторяющимися короткими фразами, разбросанными в документе (совпадающие сходства), многочисленными короткими фразами расположенные рядом друг с другом (парафразирование) или обширными фрагментами без указания источника ("криптоцитаты").

10 самых длинных фраз

Цвет текста

ПОРЯДКОВЫЙ НОМЕР	НАЗВАНИЕ И АДРЕС ИСТОЧНИКА URL (НАЗВАНИЕ БАЗЫ)	КОЛИЧЕСТВО ИДЕНТИЧНЫХ СЛОВ (ФРАГМЕНТОВ)
------------------	--	---

из базы данных RefBooks (0.00 %)



ПОРЯДКОВЫЙ НОМЕР	НАЗВАНИЕ	КОЛИЧЕСТВО ИДЕНТИЧНЫХ СЛОВ (ФРАГМЕНТОВ)
------------------	----------	---

из домашней базы данных (0.00 %)



ПОРЯДКОВЫЙ НОМЕР	НАЗВАНИЕ	КОЛИЧЕСТВО ИДЕНТИЧНЫХ СЛОВ (ФРАГМЕНТОВ)
------------------	----------	---

из программы обмена базами данных (0.00 %)



ПОРЯДКОВЫЙ НОМЕР	НАЗВАНИЕ	КОЛИЧЕСТВО ИДЕНТИЧНЫХ СЛОВ (ФРАГМЕНТОВ)
------------------	----------	---

из интернета (0.00 %)



ПОРЯДКОВЫЙ НОМЕР	ИСТОЧНИК URL	КОЛИЧЕСТВО ИДЕНТИЧНЫХ СЛОВ (ФРАГМЕНТОВ)
------------------	--------------	---

Список принятых фрагментов (нет принятых фрагментов)

ПОРЯДКОВЫЙ НОМЕР	СОДЕРЖАНИЕ	КОЛИЧЕСТВО ИДЕНТИЧНЫХ СЛОВ (ФРАГМЕНТОВ)
------------------	------------	---

Протокол анализа Отчета подобию Научным руководителем

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

Автор: ~~Ыдырыс Нұрсұлтан~~

Название: ~~Мектеп асханасындағы менюді қашықтықтан кере алу және тапсырыс беру үшін мобильді веб-қосымшасын әзірлеу~~

Координатор: ~~Максатбек Сәтдібеков~~

Козэффициент подобию 1:0

Козэффициент подобию 2:0

Замена букв:0

Интервалы:0

Микропробелы:0

Белые знаки: 0

После анализа Отчета подобию 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:

Заимствования не обнаружены

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: Ыдырыс Нұрсұлтан

Название: Мектеп асханасындағы менюді қашықтықтан көре алу және тапсырыс беру үшін мобильді веб-қосымшасын әзірлеу

Координатор: Максатбек Сатымбеков

Коэффициент подобия 1:0

Коэффициент подобия 2:0

Замена букв:0

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, работа признается самостоятельной и допускается 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, работа не допускается к защите.

Обоснование:

Заимствования не обнаружены

.....
.....

Дата 03.06.2021



Подпись заведующего кафедрой /

начальника структурного подразделения

Заведующий кафедрой КОиХИ Сейлова Н.А.

Окончательное решение в отношении допуска к защите, включая обоснование:

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

Дата 03.06.2021



Подпись заведующего кафедрой /

начальника структурного подразделения

Заведующий кафедрой КОиХИ Сейлова Н.А.