

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

Қ.Сәтбаев атындағы қазақ ұлттық техникалық зерттеу университеті

Ақпараттық және телекоммуникациялық технологиялар институты

Киберқауіпсіздік, ақпаратты өңдеу және сақтау кафедрасы

Жанұзақ Таңнұр Алтынбекқызы

Білім беру ұйымдарында тестілер өткізуге арналған
веб-қосымшаны әзірлеу

ДИПЛОМДЫҚ ЖҰМЫС

5B070300 – «Ақпараттық жүйелер» мамандығы

Алматы 2021

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ



Қ.Сәтбаев атындағы қазақ ұлттық техникалық
зерттеу университеті

Ақпараттық және телекоммуникациялық
технологиялар институты
Киберқауіпсіздік, ақпаратты өңдеу және сақтау
кафедрасы

ҚОРҒАУҒА ЖІБЕРІЛДІ

КАӨЖС кафедра меңгерушісі,
тех.ғыл.канд, ассоц.

доцент

Н.А.Сейлова

« 31 » 05 2021 ж.

ДИПЛОМДЫҚ ЖҰМЫС

Тақырыбы: «Білім беру ұйымдарында тестілер өткізуге арналған
веб-қосымшаны әзірлеу»

5B070300 – «Ақпараттық жүйелер» мамандығы

Орындаған:

Жанұзақ Т.А.

Ғылыми жетекші:

лектор, магистр
Дүйсенбаева А.Н.

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

Қ.Сәтбаев атындағы қазақ ұлттық техникалық зерттеу университеті

Ақпараттық және телекоммуникациялық технологиялар институты

Киберқауіпсіздік, ақпаратты өңдеу және сақтау кафедрасы

БЕКІТЕМІН

КАӨС кафедра меңгерушісі,

тех.ғыл.канд, ассоц. доцент

«_____ Н.А.Сейлова
31 » 05 2021 ж.

Дипломдық жұмысты орындауға ТАПСЫРМА

Білім алушы: Жанұзақ Таңнұр Алтынбекқызы

Тақырыбы: «Білім беру ұйымдарында тестілер өткізуге арналған
веб-қосымшаны әзірлеу»

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

Аяқталған жұмысты тапсыру мерзімі 2021 жылғы «20» мамыр

Дипломдық жұмыстың бастапқы берілістері: диплом алдындағы практикалық
жұмыс қорытындысы, тақырып бойынша әдебиеттерге шолу
нәтижелері, теориялық мәліметтердің жиыны

Дипломдық жұмыста қарастырылатын мәселелер тізімі:

а) қойылған мәселенің қазіргі жағдайын пайымдау

ә) деректер базасының моделін құру

б) программалық камтаманы құру

Сызбалық материалдар тізімі: Power Point бағдарламасындағы слайдтар

Сызба материалдар: 20 слайдпен көрсетілген

Ұсынылатын негізгі әдебиет: 11 атау

Дипломдық жұмысты дайындау

КЕСТЕСІ

Бөлімдер атауы, қарастырылатын мәселелер тізімі	Ғылыми жетекші мен кеңесшілерге көрсету мерзімдері	Ескерту
Мәселенің қазіргі жағдайына шолу және оны талдау	8.02.2021	
Ақпараттық қамтаманы құру	28.03.2021	
Программалық қамтаманы құру	3.05.2021	

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, магистр		
Программалық қамтама			

Ғылыми жетекші _____



_____Дүйсенбаева А.Н.

Тапсырманы орындауға алған білім алушы _____



Жанұзақ Т.А.

Күні _____

« 24» 11 2020 ж.

АҢДАТПА

Дипломдық жұмыстың мақсаты білім беру ұйымдарында тестілеу өткізу үшін арнайы веб қосымша жасау.

Жасалған веб-қосымша білім алушыларды тіркеу, оларды арнайы класстарға бөліп, белгілі бір пән бойынша тест және емтихан өткізетін оқытушылардың жұмысын автоматтандыратын веб-қосымша болып табылады. Бұл веб-қосымша оқытушылардың жұмыстарын жеңілдетіп, уақытты ұтымды пайдалануға мүмкіндік береді.

Бұл дипломдық жұмысты жасау барысында Postgre SQL мәліметтер базасы мен C#, React программалау тілдері және Microsoft Visual Studio ортасы қолданылды.

АННОТАЦИЯ

Целью дипломной работы является создание специального веб-приложения для тестирования в учебных заведениях.

Разработанное веб-приложение представляет собой веб-приложение, автоматизирующее работу учителей, которые регистрируют студентов, разделяют их на специальные классы и проводят тесты и экзамены по определенному предмету. Это веб-приложение облегчает работу учителей и позволяет им более эффективно использовать время.

При разработке данной дипломной работы использовались база данных PostgreSQL, языки программирования C#, React и среда Microsoft Visual Studio.

ANNOTATION

The purpose of the thesis is to create a special web application for testing in educational institutions.

The developed web application is a web application that automates the work of teachers who register students, divide them into special classes and conduct tests and exams in a specific subject. This web-based application makes the work of teachers easier and allows them to use their time more efficiently.

In the development of this thesis, PostgreSQL database, C#, React programming languages and Microsoft Visual Studio environment were used.

МАЗМҰНЫ

КІРІСПЕ	10
1. ПӘНДІК САЛАНЫ ЗЕРТТЕУ ЖӘНЕ ТАЛДАУ	11
1.1 Пәндік саланы талдау	11
1.2 Таңдалған тақырыптың өзектілігі	11
1.3 Қолданыстағы ұқсас веб-қосымшаларға шолу	12
1.4 Веб-қосымшаға қойылатын талаптар	12
1.5 Концептуалды қолдану үлгісі	12
1.6 Есептің қойылымы	14
2. ВЕБ ҚОСЫМШАНЫ ЖОБАЛАУ	16
2.1 Жалпы қолданбалы архитектура	16
2.2 МБ(Мәліметтер базасы) таңдау және ER диаграммасын құру	17
3 ПРОГРАММАЛЫҚ ҚАМТАМАНЫ ҚҰРУ	21
3.1 Разработка жасауға арналған программалау тілдерін таңдау	21
3.2 Функционалдық құрылымдық сұлба	21
ҚОРЫТЫНДЫ	39
ПАЙДАЛАНЫЛҒАН ӘДЕБИЕТЕР ТІЗІМІ	40
А қосымшасы	41
Б қосымшасы	44
С қосымшасы	49

КІРІСПЕ

Білім берудегі веб қосымшалар қазіргі кезде қоғамның ақпараттық өркениетке көшуінің басты баспалдағы болып табылады. Заманауи технологиялар білім беру үдерісін өзгертуге, пайдаланушыны ақпараттық-білім беру ортасына енуіне, білім сапасын арттыруға, ақпаратты қабылдау мен білімді игеру процестеріне түрткі болуға мүмкіндік береді. Жаңа ақпараттық технологиялар білім берумен қоса, әртүрлі қызмет салаларында компьютерлік ұйымдастыруды және басқаруды қолдау үшін жағдай жасайды. Ақпараттық технологияларды білім беру бағдарламаларына қолдану барлық деңгейлерде жүзеге асырылады: мектеп, университет және жоғары оқу орнынан кейінгі білім.

Қазіргі уақытта білім беру мекемелері білім сапасын арттыру мақсатында компьютерлік технологиялар мен ақпараттық технологияларды белсенді түрде қолданады.

Бүгінгі таңда білім беру мекемелерінің беделі тек білім сапасы деңгейіне байланысты емес, арнайы жүйенің тиімділігіне байланысты білім алушының білімін бақылау болып табылады. Ең жедел, қазіргі таңда өте жиі қолданылатын заманауи форма - компьютерлік тестілеуді өткізу.

Бұл дипломдық жұмыстың тақырыбын таңдаудағы негізгі себеп қазіргі қоғамда болып жатқан ақуалға байланысты болып келеді. Себебі біз қашықтықтан оқу форматына әлі дайын емеспіз және оған сай жетілдірілген платформалар да өте көп емес. Сол себепті дипломдық жұмыстың тақырыбын білім беру саласына қатысты етіп таңдадым. Нәтижесінде білім беру ұйымдарына көмек ретінде веб-қосымша әзірленетін болады.

1 ПӘНДІК САЛАНЫ ЗЕРТТЕУ ЖӘНЕ ТАЛДАУ

1.1 Пәндік саланы талдау

Қазіргі таңдағы ақпараттық-технологиялардың қарқынды дамуы білім беру тұрғысынан болашақ кез келген білікті маманды даярлап шығуға тікелей қатысты болып келеді. Сол себепті дамыған технологияларды кеңінен қолдану барлық білім беру мекемелері қызметкерлеріне және олардың кәсіби дайындығына қойылатын талаптар қатарына жатады.

Оқыту процесін жетілдірудің бір бағыты білім дағдыларын бақылаудың жедел жүйесін құру, білім алушының білімін объективті бағалауға мүмкіндік беру. Қазіргі уақытта білім алушыны бақылау мен бағалаудың көптеген әдістері бар. Соның ішінде кеңінен қолданысқа ие жүйе тестілеу болып табылады.

Білім беру ұйымдарындағы тестілеу білім алушының оқу бағдарламаларын білуіне объективті бақылау құралы ретінде қолданылады. Білім алушының білім деңгейін жүйелі бақылау, бір жағынан, білім алушының үлгерімін анықтаса, екінші жағынан, бұл көрсеткіш оқыту әдістерінің тиімділігі және оқу процесін ұйымдастыруға көмектеседі. Оқу үрдісіне әр түрлі тест тапсырмаларының енгізілуі студенттерді оқу материалын игеру бойынша жұмысты күшейтуге итермелейді және олардың қабілеттерін дамытуға деген ұмтылысты қалыптастырады.

Қазіргі уақытта білім беру ұйымдары тестілеудің екі түрін қолданады, компьютерлік және сауалнама түрінде.

Компьютерлік тестілеуді өткізудің бірнеше артықшылығы бар:

- тест нәтижелерін тез қалыптастыру;
- оқытушының жұмысын азайту;
- бағалаудағы объективтілік;
- уақытты үнемдеу.

Білім алушының білімін сауалнама түрінде тексеру кезінде оқытушы тарапынан қаржылық және ресурстар өте көп қолданылады (сәйкес тесттерді табу, оларды тексеру, баға қою және т.б.).

Егер тестілеу процедурасын сауалнама және компьютерлік түрінде салыстырар болсақ, компьютерлік тестілеу тиімдірек екені анық.

1.2 Таңдалған тақырыптың өзектілігі

Бұл веб қосымша базаға деректерді сақтауды, деректерді енгізу мен түзетуді, сондай-ақ әртүрлі есептерді қалыптастыруды ұсынады.

Сондықтан бұл веб қосымша білім беру ұйымдары үшін бүгінгі күні өзекті және жоғары сұранысқа ие болып табылады.

Қазіргі таңда, тестілеуді өткізу жүйелерінің көптеген түрі бар. Әр жүйе өз бетінше бірегей. Мен жасаған веб-қосымшаның өзектілігі – бұл жүйе, біздің білім беру мекемелеріміз үшін арнайы жасалған. Әрбір білім беру ұйымдары

өзгеше – сол білім ұйымының класстары, директоры, оқытушылары, студенттері және пәндері бөлек.

1.3 Қолданыстағы ұқсас веб-қосымшаларға шолу

Қазіргі таңда тест тапсыратын веб қосымшалар өте көп. Мысал ретінде «СИНТЕЗ» және «Конструктор тестов» жүйелері таңдап алынды.

«СИНТЕЗ» жүйесі (Система Интерактивного Тестирования Знаний)

«СИНТЕЗ» жүйесі:

- интернет арқылы жұмыс істеу мүмкіндігі бар;
- платформа: PHP, MySQL;
- осы бағдарламаның лицензиясы ақылы негізде таратылады.

Әр түрлі типтегі сұрақтарды құруға, оларды тақырыпқа бөлуге, сұрақ-жауап мәтініне суреттер, дыбыс, фильмдер, тізімдер, кестелер және т.б. енгізуге болады.

«Конструктор тестов» жүйесі:

- интернет арқылы жұмыс істеу мүмкіндігі жоқ;
- платформа: C ++;
- осы бағдарламаның лицензиясы ақысыз таратылады.

Тесттерде музыкалық және дыбыстық сүйемелдеуді, графикалық бейнелерді, бейнематериалдарды қолдануға болады. Білім алушыны тестілеу кезек-кезек бір компьютерде жүзеге асырылады. Әр тест тапсырушы бағдарламаға өзінің атымен кіреді.

1.4 Веб-қосымшаға қойылатын талаптар

Бұл дипломдық жұмыстың мақсаты – білім алушының білім, білік, дағды деңгейлерін компьютерлік тестілеуге арналған веб-қосымша құру. Веб-қосымша оқытушы мен білім алушыға уақытты тиімді жүзеге асыруға мүмкіндік береді.

Веб-қосымшаның негізгі міндеттері:

- білім алушылардың білім деңгейін анықтау;
- тестілеу кезінде оқытушының жұмысын жеңілдету және
- оның нәтижелерін өңдеу;
- интернет арқылы ақпараттық жүйеге қол жеткізуді қамтамасыз ету;
- тестілеу кезінде оқытушы мен білім алушылардың жұмсаған уақытын қысқарту.

Қазіргі кезде технологиялар әртүрлі функционалдығы бар күрделілігі жоғары веб-қосымшалар құруға мүмкіндік береді. Бүгінгі таңда веб-қосымшалар корпоративті желісі дамыған кәсіпорындарда кеңінен қолданылады, өйткені мұндай қосымшаларды қолдану әлдеқайда арзан және көп уақытты қажет етпейді.

1.5 Концептуалды қолдану үлгісі

Веб-қосымшаға кірген кезде, веб-қосымшаның ең негізгі артықшылықтары туралы ақпарат беретін бастапқы беті бар, ол бастапқы бетке өткенде оқу орнын іздейтін ұяшық пайда болады. Пайдаланушы өзіне керек оқу орнын басқанда сол оқу орны туралы мәліметтер пайда болады.

Веб қосымшада «Басты беттен» кейін әр жүйе пайдаланушысына байланысты бірнеше бөлімдерге бөлініп қарастырылған.

Тіркелген пайдаланушы - жүйенің тіркелген пайдаланушысы. Тіркелген пайдаланушы жүйелік мәліметтерге қол жеткізудің 4 түрлі деңгейі бар: «мұғалім», «білім алушы», «директор» және «админ».

Админ – жүйедегі өз құқықтары бар пайдаланушы. Оның бұл жүйеде негізгі құқықтары:

- білім алушыны тіркеу, оқу, өшіру және қайта жаңарту;
- класстарды тіркеу, оқу, өшіру және қайта жаңарту;
- мұғалімді тіркеу, оқу, өшіру және қайта жаңарту;
- директорды тіркеу, оқу, өшіру және қайта жаңарту;
- оқу орнын тіркеу, оқу, өшіру және қайта жаңарту.

Админ жүйе қолданушыларының және жүйенің жақсы жағдайда жұмыс істеуін тізімін қадағалайды.

Директор – директор құқығы бар пайдаланушы, ол өз кезегінде оқу орнының басшысы. Жүйемен жұмыс жасау барысындағы негізгі құқықтары:

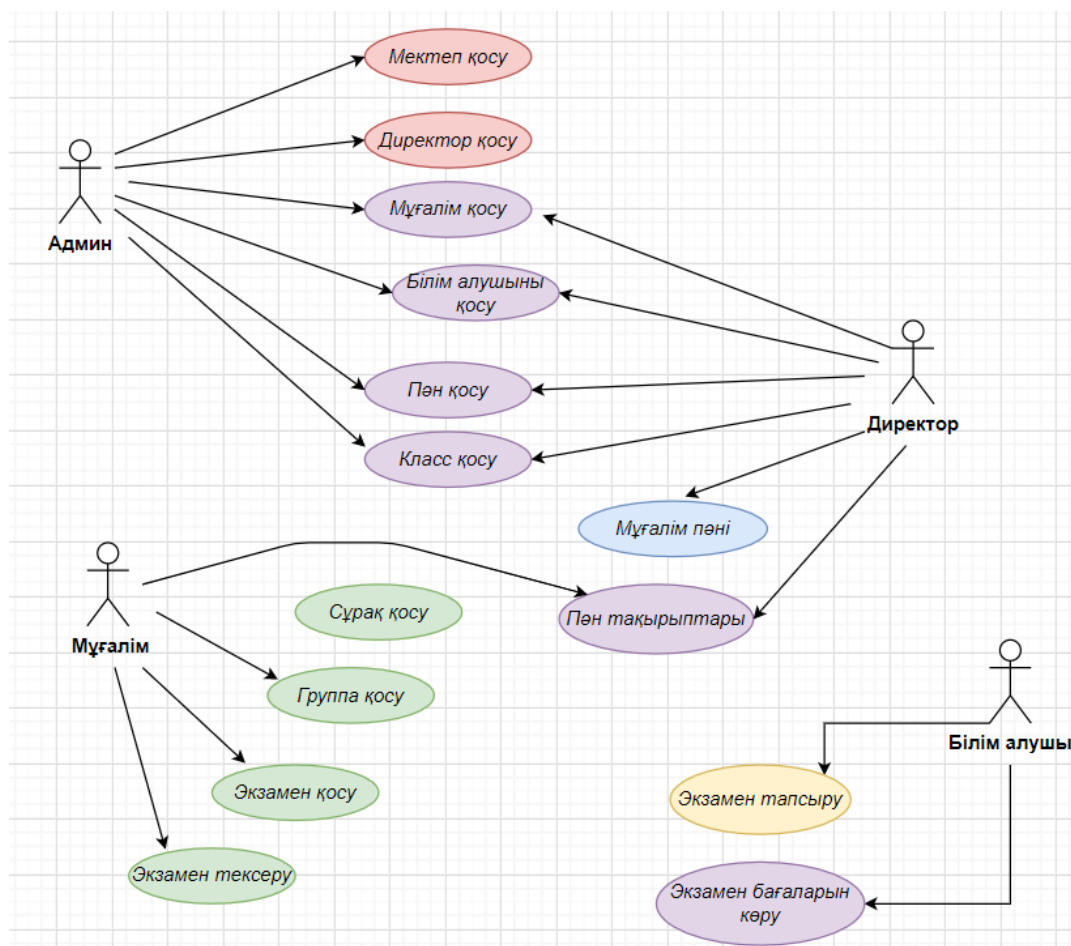
- мұғалімді тіркеу, оқу, өшіру және қайта жаңарту;
- білім алушыны тіркеу, оқу, өшіру және қайта жаңарту;
- класстарды тіркеу, оқу, өшіру және қайта жаңарту;
- мұғалімнің сабақтарын өзгерту, қосу және өшіру.

Мұғалім – мұғалімнің құқығы бар пайдаланушы, ол өз кезегінде оқу орнының оқытушысы. Жүйемен жұмыс жасау барысындағы негізгі міндеттері:

- керекті мәліметтерді енгізу, оқу, өшіру және қайта жаңарту;
- тесттерді құру, оқу, өшіру және қайта жаңарту;
- тесттің дұрыс өткізілгендігін тексеру.

Білім алушы – білім алушының құқықтары бар пайдаланушы. Бұл пайдаланушы жүйенің ең негізгі пайдаланушысы болып табылады. Білім алушы мұғалімнің жүктеген мәліметтерін, тесттерін көріп, тапсыра алады.

Жүйеге кіру – веб қосымша қолданушысы өз логині мен құпия сөзі арқылы кіріп, мұндай қолданушы бар болса, ол келесі мүмкіндіктері бар бетке өте алады, ал егер қате мәліметтер енгізілсе онда оған қате екендігі туралы хабарлама келеді.



1.5 сурет - Рөлдердің прецеденттер диаграммасы

1.6 Есептің қойылымы

Дипломдық жұмыстың негізгі мақсаты білім беру ұйымдарына ынғайлы қолданушы интерфейсін жасап ұсыну болып табылады. Веб-қосымша білім алушының білімін бақылауды және тестілеу кезінде білім алушы мен оқытушы тарапынан кететін уақыт пен шығынның азаюына мүмкіндік береді. Бұл веб-қосымшаны білім беру ұйымдары жалпылама қолдана алады, атап айтқанда мектеп, колледж, университет.

Веб-қосымшаны құру үшін келесі есептерді қалыптастыру қажет:

- пәндік саланы талдау;

- веб-қосымшаны жасау кезінде заманауи технологияларды қолдану;

- пайдаланушыға ыңғайлы интерфейс құру;
- мәліметтер базасының басқару жүйесін(МББЖ) таңдау;
- мәліметтер базасының құрылымын жобалау;
- веб-қосымшаны жасауға арналған бағдарламалық құралдарды таңдау;
- жүйенің ER-моделін құру;
- веб қосымшаның сұлбасын жасау;
- веб қосымшаны іске асыру және тестілеу.

Жасалған веб-қосымшаны келесі жүйе қолданушыларына арналған:
Білім беру ұйымының басшыларына:

- білім беру мекемесін басқаруға кететін уақытты үнемдеу;
- қажетті ақпараттарға жылдам қол жеткізу;
- білім беру ұйымында электронды құжат айналымын іске асыру;
- білім беру жүйесін және басқару тиімділігін арттыру.

Оқытушыларға:

- тест және емтихан құру кезіндегі уақытты және басқада шығындарды азайту;
- оқытушылардың жұмыстарын жеңілдету;
- студенттердің тапсырылған тест немесе емтихан нәтижелеріне онлайн қол жеткізу;
- тестілеу арқылы оқыту жүйесін арттыру және оқыту сапасын тексеруді ұйымдастыру;

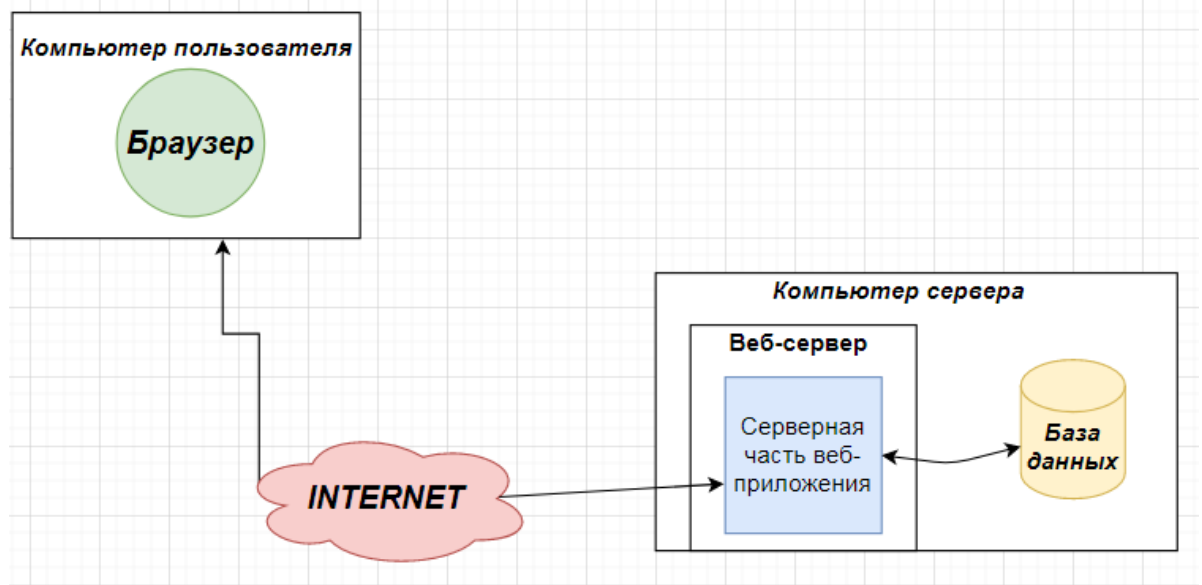
Білім алушыларға:

- жүйеге кіру арқылы тест немесе емтихан уақытын онлайн қадағалау;
- тест немесе емтихан тапсыратын уақытты ұтымды пайдалану;
- тест немесе емтиханнан алған нәтижесіне онлайн қол жеткізу;

2 ВЕБ ҚОСЫМШАНЫ ЖОБАЛАУ

2.1 Жалпы қолданбалы архитектура

Жасалып жатқан жүйенің өзегі - бұл клиент-сервер архитектурасы, онда жұмыс орындары немесе желі жүктемесі серверлер деп аталатын қызмет көрсетушілер мен клиенттер деп аталатын қызметтердің клиенттері арасында бөлінеді.



2.1 сурет - Жалпы қолданбалы архитектура

Веб-қосымша пайдаланушы мен мәліметтер базасы арасындағы сілтеме ретінде жұмыс істейді.

Мәліметтер базасында деректерді серверде сақтау келесі себептерге байланысты:

- сервердегі орталықтандырылған сақтау жергілікті машиналарда сақтауға карағанда сенімді: серверге физикалық және бағдарламалық қамтамасыз етудің қол жетімділігі шектеулі, деректердің сақтық көшірмесі үнемі жасалады;
- мәліметтер қорының реляциялық құрылымы байланысты деректерге жылдам қол жеткізуді қамтамасыз етеді;
- деректердің қажетсіз қайталануы алынып тасталынады.

Клиент-сервер архитектурасының артықшылықтары мен кемшіліктеріне тоқтала кетейік.

Негізгі артықшылықтары:

- есептеу жүйесінің функцияларын желідегі бірнеше бір-біріне тәуелсіз компьютерлер арасында бөлу мүмкіндігін қарастырады. Бұл дегеніміз

есептеу жүйесінің қызмет көрсетуін жеңілдететін ең басты мүмкіндік болып саналады және де серверді ауыстыру, жөндеу, қайта жаңарту немесе орынын ауыстыру клиенттерге әсер етпейді;

- көптеген деректер өте жақсы қорғалған серверде сақталады;
- бір сервердің ресурстарын әр түрлі клиенттер қолдана алады. Бұл клиенттер аппараттық платформалары мен операциялық жүйелері бар клиенттер.

Негізгі кемшіліктері:

- орталықтандырылған жүйені қолданған жағдайда, негізгі сервердің істен шығуы қолданыстағы жүйелердің барлығын жарамсыз етуі мүмкіндігі бар;
- осы ақпараттық жүйені басқару үшін білікті маман қажет;
- жабдықтардың құнының жоғарылығы.

2.2 МБ(Мәліметтер базасы) таңдау және ER диаграммасын құру

Веб қосымша разработкасын жасау үшін мәліметтер базасы ретінде PostgreSQL таңдап алдым.

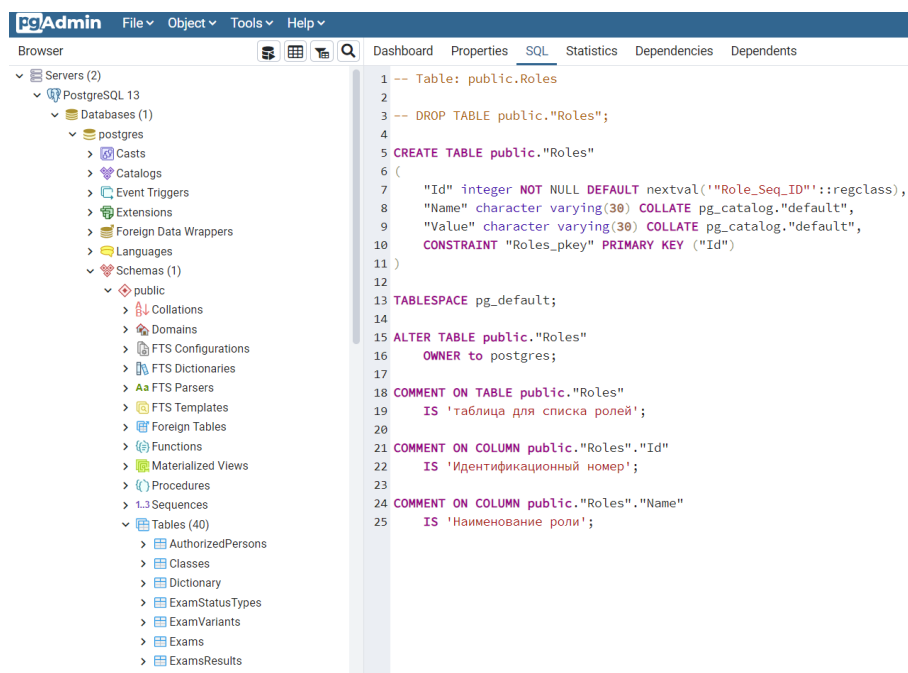
PostgreSQL - бұл ақысыз объектілік-реляциялық мәліметтер қорын басқару жүйесі (ДҚБЖ).

AIX, түрлі BSD жүйелері, HP-UX, IRIX, Linux, macOS, Solaris / OpenSolaris, Tru64, QNX және Microsoft Windows сияқты UNIX-ке ұқсас әр түрлі платформаларға арналған.

PostgreSQL SQL тіліне негізделген және SQL стандартының көптеген мүмкіндіктерін қолдайды.

PostgreSQL-дің мықты жақтары:

- жоғары өнімді және сенімді транзакциялар мен репликация механизмдері;
- мұрагерлік(наследование);
- геометриялық объектілерді индекстеу мүмкіндігі және оның негізінде PostGIS кеңейтуінің болуы;
- оларды индекстеу мүмкіндігімен JSON форматындағы жартылай құрылымды деректерге арналған қолдау;
- кеңейту мүмкіндігі (деректердің жаңа типтерін, индекстер типтерін, бағдарламалау тілдерін, кеңейту модульдерін құру, кез-келген сыртқы дерек көздерін қосу мүмкіндігі).



2.2 сурет - Мәліметтер базасы

ER диаграмма - бұл мәліметтер базасын жобалау процесін жеңілдетуге арналған семантикалық мәліметтер моделі. ER диаграммасынан мәліметтер базасының барлық түрлерін жасауға болады: реляциялық, иерархиялық, желілік, объектілік. ER-моделі «болмыс», «қатынас» және «атрибут» ұғымдарына негізделген.

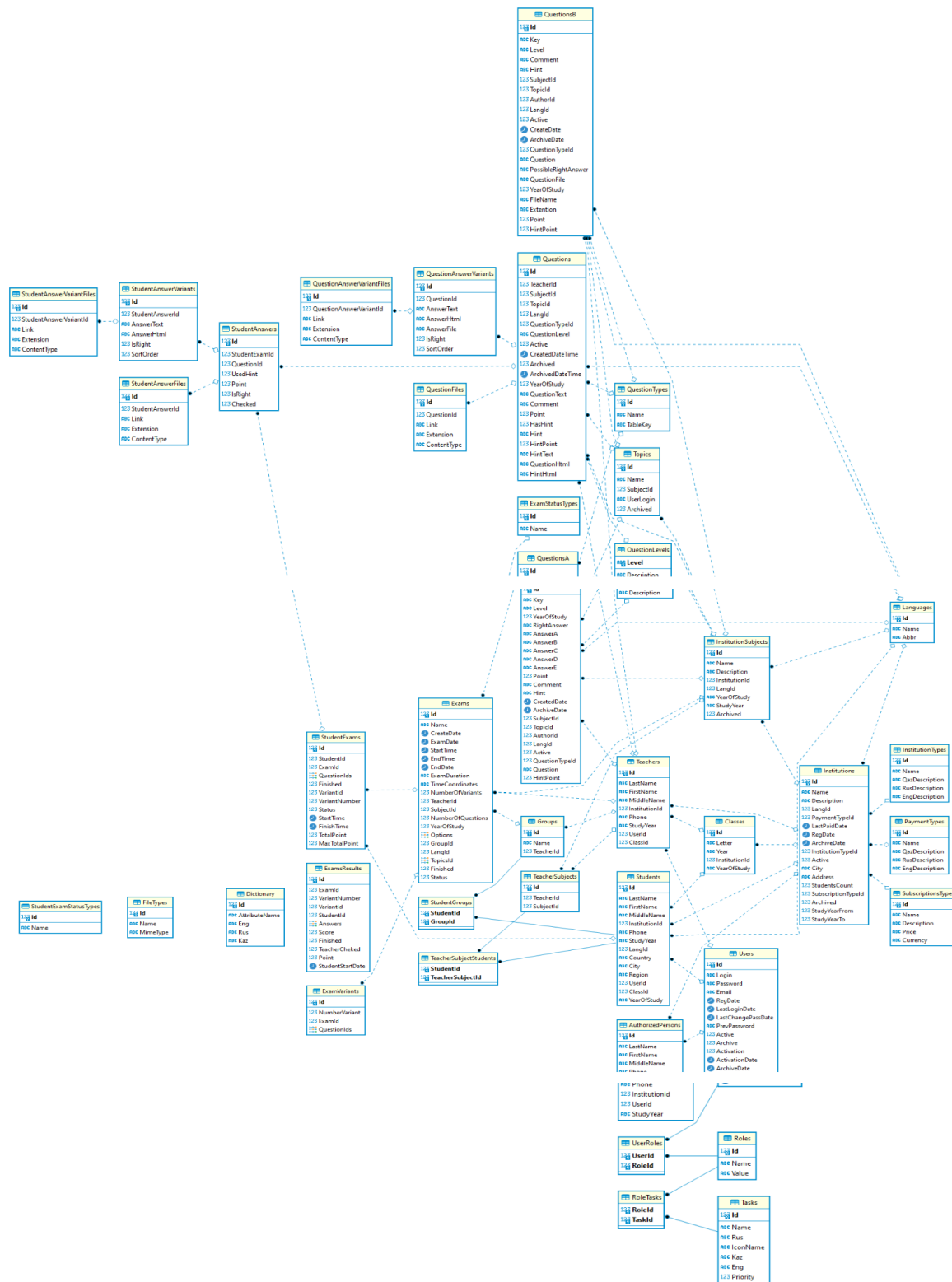
ER диаграмма- мәліметтер базасының графикалық диаграммалардағы визуалды көрінісі. ER диаграммасы белгілі бір пәндік аймақты анықтайтын процесті көзге елестетеді. Сонымен қатар ER диаграмма нысандарды, атрибуттар мен қатынастарды графикалық түрде бейнелейтін диаграмма.

Ірі деректер қорын басқару жүйесі үшін ER диаграммасын құру жобалау кезінде туындайтын қателіктерден аулақ болуға мүмкіндік береді.

Мәліметтер базасында әр кесте үшін басты кілт болады. Басты кілт-жазбаның бірегейлігін анықтайтын өріс. Барлық кестелер үшін басты кілт арнайы енгізілген идентификатор болып табылады.

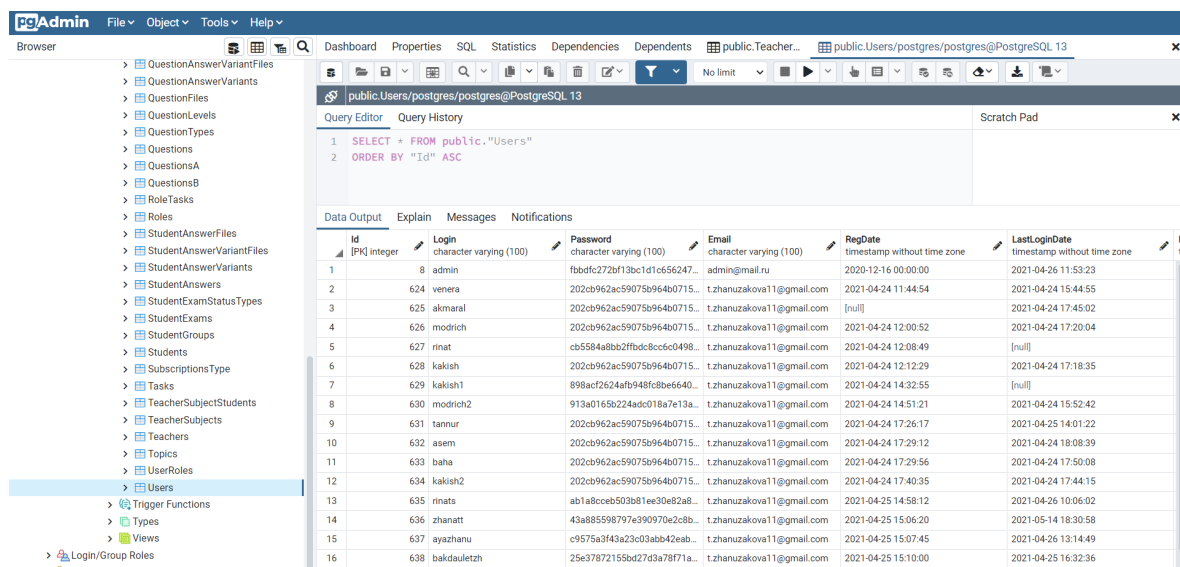
2.3 суретте көрсетілгендей әр кестеде өзінің басты кілті бар және ол «id» идентификаторы арқылы енгізілген. Басты кестелерді мысалға келтірер болсақ:

- «Users» кестесі үшін User_id;
- «User Roles» кестесі үшін Role_id;
- «Students» кестесі үшін Student_id;
- «Teachers» кестесі үшін Teacher_id .



2.3 сурет - Жүйенің ER диаграммасы

2.4 суреттен Postgre SQL мәліметтер базасындағы ең басты кестелердің бірі «Users» кестесінен жүйе бойынша тіркелген пайдаланушыларды көруге болады.



The screenshot shows the PgAdmin interface with the 'Users' table selected in the left sidebar. The table is displayed in the 'Data Output' tab, showing 16 rows of data. The columns are: Id (PK), Login, Password, Email, RegDate, and LastLoginDate. The data is sorted by Id in ascending order.

Id	Login	Password	Email	RegDate	LastLoginDate
1	admin	fbdfc272bf13bc1d1c565247...	admin@mail.ru	2020-12-16 00:00:00	2021-04-26 11:53:23
2	venera	202cb962ac59075b964b0715...	t.zhanuzakova11@gmail.com	2021-04-24 11:44:54	2021-04-24 15:44:55
3	akmaral	202cb962ac59075b964b0715...	t.zhanuzakova11@gmail.com	[null]	2021-04-24 17:45:02
4	modrich	202cb962ac59075b964b0715...	t.zhanuzakova11@gmail.com	2021-04-24 12:00:52	2021-04-24 17:20:04
5	rinat	cb5584a8bb2ffbd8c5c5c0498...	t.zhanuzakova11@gmail.com	2021-04-24 12:08:49	[null]
6	kakish	202cb962ac59075b964b0715...	t.zhanuzakova11@gmail.com	2021-04-24 12:12:29	2021-04-24 17:18:35
7	kakish1	898acf2624af948fc8be6640...	t.zhanuzakova11@gmail.com	2021-04-24 14:32:55	[null]
8	modrich2	913a0165b224dcd018a7e13a...	t.zhanuzakova11@gmail.com	2021-04-24 14:51:21	2021-04-24 15:52:42
9	tannur	202cb962ac59075b964b0715...	t.zhanuzakova11@gmail.com	2021-04-24 17:26:17	2021-04-25 14:01:22
10	asem	202cb962ac59075b964b0715...	t.zhanuzakova11@gmail.com	2021-04-24 17:29:12	2021-04-24 18:08:39
11	baha	202cb962ac59075b964b0715...	t.zhanuzakova11@gmail.com	2021-04-24 17:29:56	2021-04-24 17:50:08
12	kakish2	202cb962ac59075b964b0715...	t.zhanuzakova11@gmail.com	2021-04-24 17:40:35	2021-04-24 17:44:15
13	rinats	ab1a8cccb503b81ee30e82a8...	t.zhanuzakova11@gmail.com	2021-04-25 14:58:12	2021-04-26 10:06:02
14	zhanatt	43a885598797e390970e2c8b...	t.zhanuzakova11@gmail.com	2021-04-25 15:06:20	2021-05-14 18:30:58
15	ayazhanu	c9575a3f43a23c03abb42eab...	t.zhanuzakova11@gmail.com	2021-04-25 15:07:45	2021-04-26 13:14:49
16	bakdauletzh	25e37872155bd27d3a78f71a...	t.zhanuzakova11@gmail.com	2021-04-25 15:10:00	2021-04-25 16:32:36

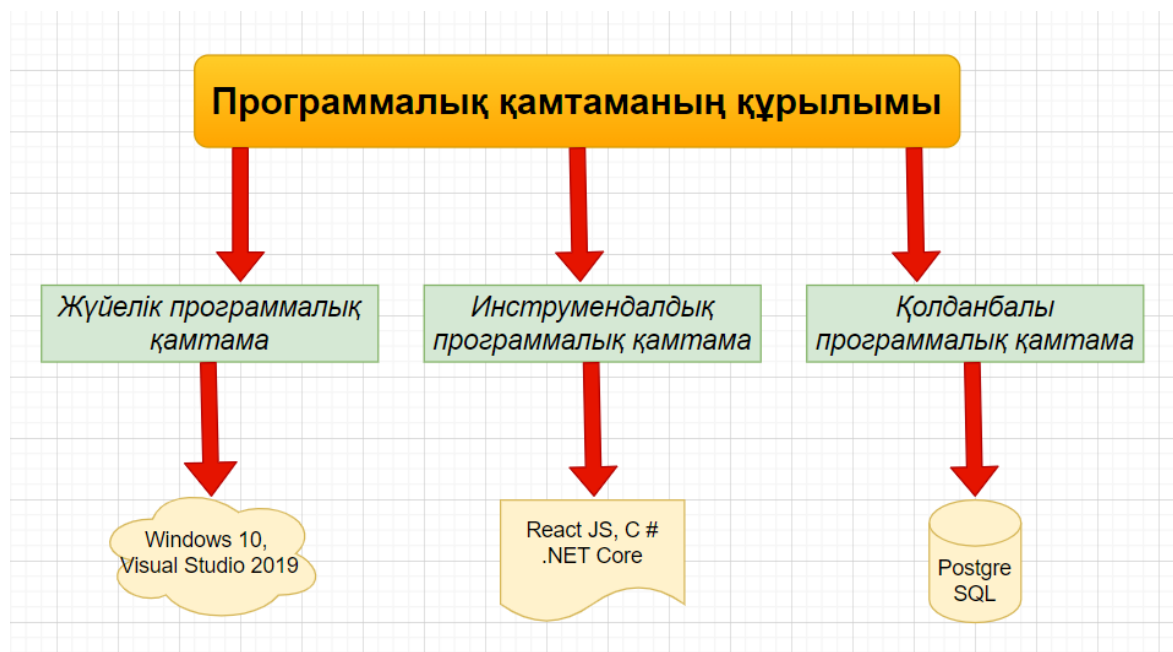
2.4 сурет - «Users» кестесі

3 ПРОГРАММАЛЫҚ ҚАМТАМАНЫ ҚҰРУ

3.1 Разработка жасауға арналған программалау тілдерін таңдау

Веб-қосымшаны әзірлеу үшін осы ортаға қолайлы тілдерді таңдаған жөн. Веб-қосымшаларды жасауға арналған келесі тілдерді қарастырамыз:

- C#;
- React.



3.1 сурет - Программалық қамтаманың құрылымы

Microsoft Visual Studio - бұл интеграцияланған бағдарламалық жасақтама ортасын және басқа да бірқатар құралдарды қамтитын Microsoft өнімдерінің желісі. Бұл өнімдер консольдық қосымшаларды да, графикалық интерфейсі бар қосымшаларды да, соның ішінде Windows Forms технологиясын қолдайтын бағдарламаларды да, веб-сайттарды, веб-қосымшаларды, веб-қызметтерді Windows платформасы қолдайтын барлық платформалар үшін жергілікті және басқарылатын кодта жасауға мүмкіндік береді, Windows Mobile, Windows CE, .NET Framework, Xbox, Windows Phone .NET Compact Framework және Silverlight.

C# (си шарп)- объектіге-бағытталған және қауіпсіз заманауи бағдарламалау тілі. C# программистерге .NET экожүйесінде жұмыс жасайтын қауіпсіз және сенімді қосымшалардың көптеген түрлерін жасауға мүмкіндік береді. C# белгілі C тілдер тобына жатады.Көбінесе C++, Java немесе JavaScript тілдерінде жұмыс жасағандарға таныс және синтаксистері де ұқсас болып келеді.

C# тілі арқылы көптеген бағдарламалық компоненттерді құруға және қолдануға болады. C# тілі пайда болғаннан бастап жаңа жүктемелерді және бағдарламалық жасақтаманы дамытудағы заманауи озық тәжірибелерді қолдау функцияларымен байытылды.

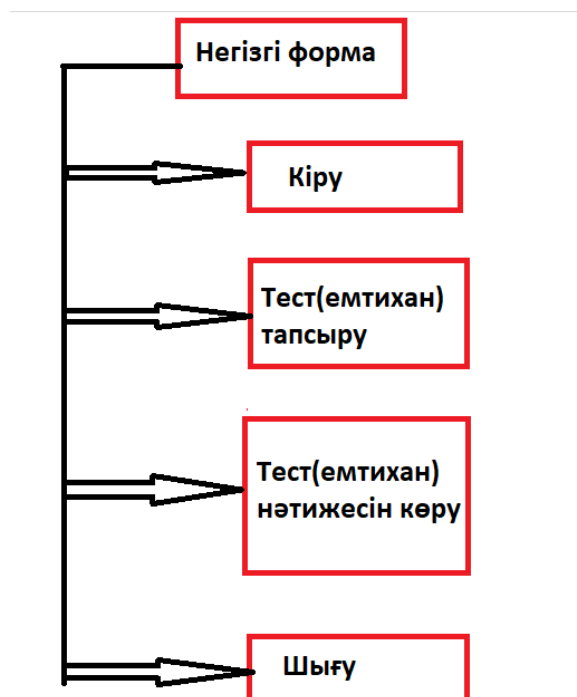
React(React.js немесе ReactJS) - бұл қолданушы интерфейстерін құруға арналған декларативті, тиімді және икемді JavaScript кітапханасы. Ол «интерфейстер» деп аталатын кішігірім, окшауланған код бөліктерінен күрделі UI құрастыруға мүмкіндік береді.

React-ты мобильді қосымшаларды әзірлеу үшін қолдануға болады. Оның мақсаты - жоғары жылдамдықты, қарапайымдылықты және масштабтылықты қамтамасыз ету. Пайдаланушы интерфейстерін дамытуға арналған кітапхана ретінде React MobX, Redux және GraphQL сияқты басқа кітапханалармен жиі қолданылады.

3.2 Функционалдық құрылымдық сұлба

Төмендегі сызбада программа жұмысының құрылымдық сұлбасы көрсетілген. Жүйеде төрт пайдаланушы бар: админ, директор, оқытушы, білім алушы.

Жүйенің негізгі пайдаланушысы білім алушының веб қосымшаға кіргеннен бастап әр қадамы көрсетілген.



3.2 сурет - Программа жұмысының құрылымдық сұлбасы

Жүйеге кіру терезесі арқылы авторизация жасау мүмкіндігіне ие. Бұл жерде пайдаланушының аты және құпия сөзді жазу жолы енгізілді.

* Имя пользователя: admin

* Пароль:

Войти

3.3 сурет - Жүйеге кіру терезесі

Админ — жүйенің ең негізгі пайдаланушысы. Басқа жүйе пайдаланушылары админ арқылы ғана жүйеге кіре алады. 3.4 суретте админнің жүйедегі атқаратын міндеттері көрсетілген.

Привязать задачи к ролям

Ученик
Учитель
Директор
Администратор

Добавить роль

Роль - "Администратор"

Название: admin Описание: Администратор Сохранить изменения в роли

Связанные задачи

☒ Роли ☒ Профиль ☐ Сдача экзамена ☐ Результаты экзамена ☒ Учителя ☒ Студенты ☒ Директоры ☒ Предметами ☐ Темы предметов

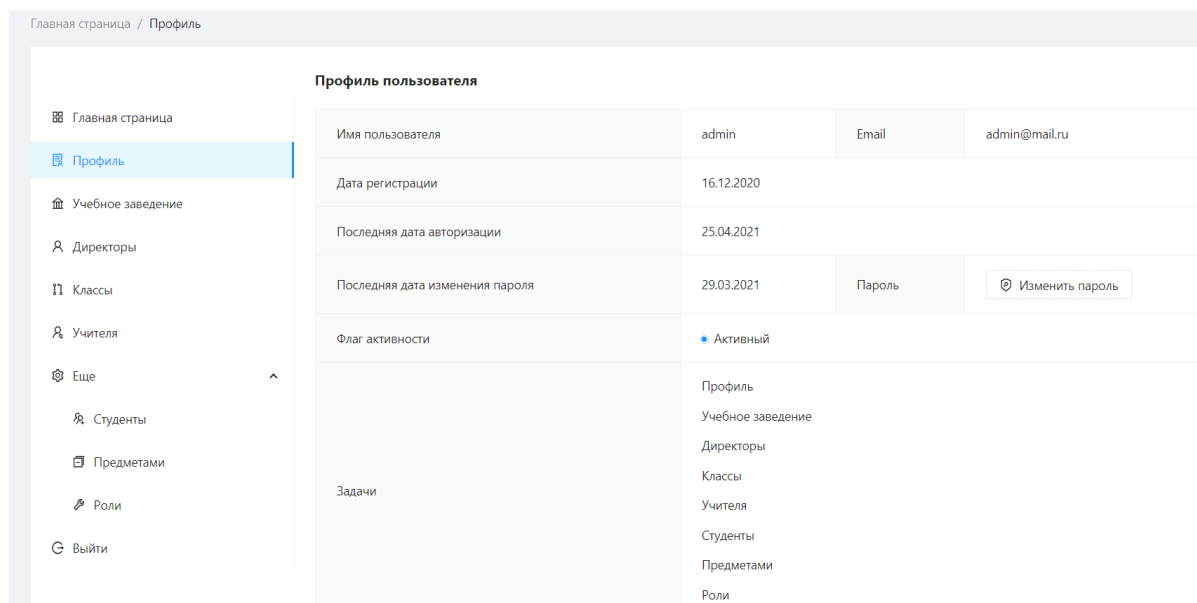
☒ Классы ☐ Предметы учителя ☐ Вопросы ☐ Экзамены ☐ Профиль учителя ☐ Группы ☐ Профиль ученика ☐ Профиль директора

☒ Учебное заведение ☐ История экзаменов студента ☐ Проверкой экзамена

3.4 сурет - Админнің негізгі міндеттері

Пайдаланушы жүйеге кірген кезде оның қандай міндеттері бар екенін «Роли» деген бөлімге өтіп көре аламыз.

Жүйеге кіргеннен кейін пайдаланушы өзінің профилін көре алады және әр пайдаланушыға сәйкес өзінің бөлімдері бөліп қарастырылған.



3.5 сурет - Админнің профилі

Админ жүйеге кіргеннен кейін оқу орнын қосады. Жүйеде оқу орнының 3 түрі көрсетілген: мектеп, колледж, университет.

[Добавить учебное заведение](#) Список учебных заведений

* Название:

* Описание:

* Язык: ▼

* Способ оплаты: ▼

* Количество учеников:

* Тип учебного заведения: ▼

* Город:

* Адрес:

* Начало года обучения:

* Конец года обучения:

3.6 сурет - Оқу орнын қосу

3.6 суретте оқу орнын қостық. Оның дұрыс қосылғандығын тізімнен көре аламыз.

Добавить учебное заведение [Список учебных заведений](#)

Школа Школа № 70

Редактирование учебного заведения

Название	Школа № 70		
Описание	Школа		
Язык	Казахский	Способ оплаты	Договор
Тип учебного заведения	Школа	Количество учеников	500
Город	Алматы	Активный	☒
Адрес	Сатпаева 22	Год обучения	1 ~ 11

3.7 сурет - Жүйеде қосылған оқу орындары

Оқу орнын қосқаннан кейін сол оқу орнына басшы тағайындайды, яғни директор қосады.

[Добавление уполномоченных лиц](#) [Список уполномоченных лиц](#)

* Фамилия: Сермахан

* Имя: Ринат

Отчество: Айдынулы

* Контактный телефон: +7 (777) 777-77-77

* Логин: rinats

* Электронная почта: t.zhanuzakova11@gmail.com

* Учебное заведение: Школа № 70, Алматы, Сатпаева 22

* Учебный год: 21/22

Добавить

3.8 сурет - Оқу орнының басшысын қосу

3.8 суретте көрсетілгендей оқу орнының басшысын қостық. 3.9 суреттен оның №70 мектепке директор болып қосылғанын көруге болады. Осы жерден қате кеткен жерді түзеуге немесе өшіруге болады.

Добавление уполномоченных лиц [Список уполномоченных лиц](#)

Школа Школа № 70 Сермахан Ринат Айдынулы

Редактирование уполномоченных лиц

Логин	rinats		
Электронная почта	t.zhanuzakova11@gmail.com		
Учебное заведение	Школа № 70	Учебный год	21/22
Фамилия	Сермахан	Имя	Ринат
Отчество	Айдынулы	Контактный телефон	+7 (777) 777-77-77
Активный	☒		

3.9 сурет - Тіркелген оқу орны басшыларының тізімі

Админ директорды қосқаннан кейін жүйенің басқа пайдаланушыларын қосу директорға да міндетті болып саналады. Директордың міндетін келесі 3.10 суреттен көруге болады.

Привязать задачи к ролям

Ученик

Учитель

Директор

Администратор

Добавить роль

Роль - "Директор"

Название: director Описание: Директор Сохранить изменения в роли

Связанные задачи

☐ Роли ☐ Профиль ☐ Сдача экзамена ☐ Результаты экзамена ☒ Учителя ☒ Студенты ☐ Директоры ☒ Предметами ☒ Темы предметов
☒ Классы ☒ Предметы учителя ☐ Вопросы ☐ Экзамены ☐ Профиль учителя ☐ Группы ☐ Профиль ученика ☒ Профиль директора
☐ Учебное заведение ☐ История экзаменов студента ☐ Проверкой экзамена

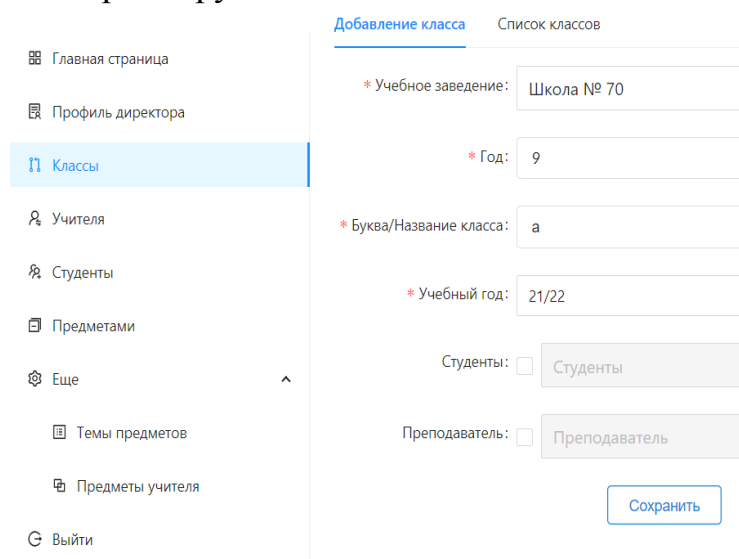
3.10 сурет - Директордың негізгі міндеттері

Жүйе пайдаланушысын қосар кезде электрондық пошта жазатын бөлім бар екенін көруге болады (3.8 сурет). Электрондық поштаға уақытша құпия сөз келеді. Сол құпия сөз арқылы жүйеге кіруге болады.

 **takostudy@gmail.com**
 кому: я ▼
 Ваш логин: rinats
 Ваш пароль: 8E\$8Z\$oG

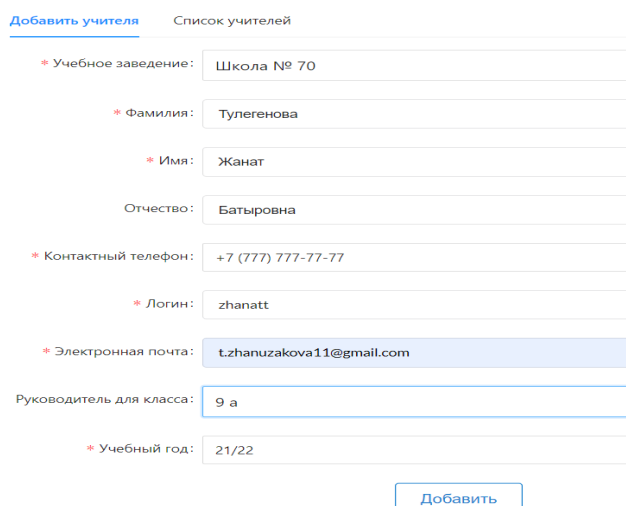
3.11 сурет - Электрондық поштаға келген директордың уақытша құпия сөзі

Директор жүйеге кіргеннен кейін ең алдымен класс қосады. 3.12суреттен директорға тиісті бөлімдерді көруге болады.

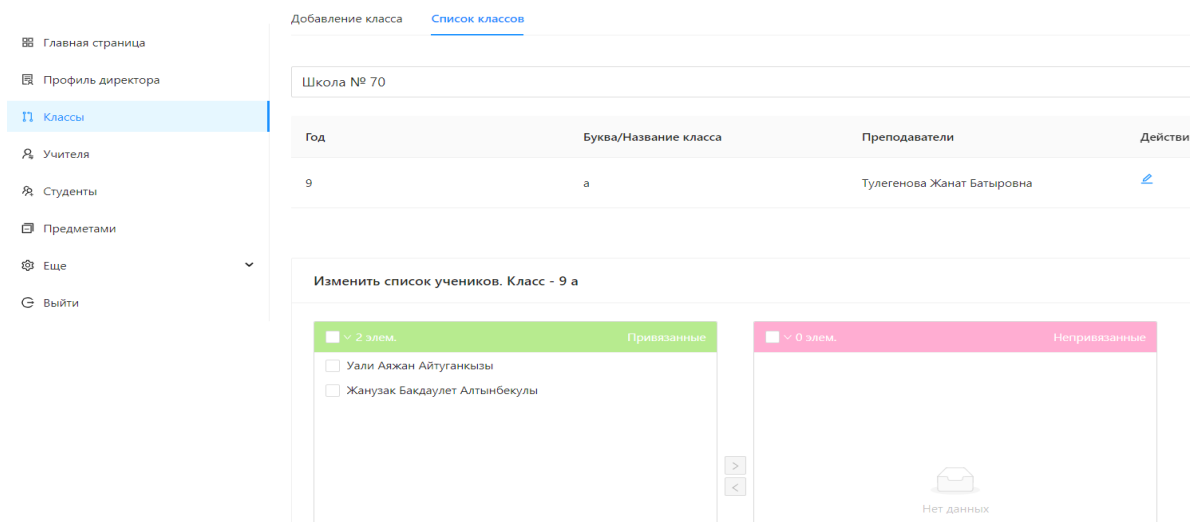


3.12 сурет - Класс қосу

Класс қосқаннан кейін мұғалім қосылады. 3.13 суретте көрініп тұрғандай мұғалімнің қазіргі уақытта қолданып жүрген электрондық поштасы жазылған. Жоғарыда айтып өткендей электрондық поштаға уақытша құпия сөз және логині жіберіледі. Сол арқылы ғана жүйеге кіре алады.



3.13 сурет - Мұғалім қосу



3.14 сурет - Класстар тізімі

3.12 суретте «9 а» класы қосылды. Сол кластың сынып жетекшісі ретінде (3.13 сурет) мұғалімді бекітеміз.

Директор класс және мұғалімді қосқаннан кейін студенттерді(оқушыларды) қосады.

3.15 сурет - Студенттерді қосу

Директор қосылған студенттерді классқа тіркейді. 3.16 суретте классқа тіркеудің екі жолы көрсетілген.

Добавить студента [Список студентов](#)

Школа Школа № 70

Выберите класс или наберите название для поиска

9 а

Уали Ааяжан Айтуганкызы

Жанузак Бакдаулет Алтынбекулы

3.16 сурет - Класска тіркеу

Осы қадамдардан кейін пән қосылады және сол пәнді бір мұғалімге тіркейді.

Привязка предметов к учителю

* Учитель: Жанат Батыровна Тулегенова

* Список предметов: Физика X

Сохранить

3.17 сурет - Пәнді мұғалімге тіркеу

Привязать задачи к ролям

Ученик

Учитель

Директор

Администратор

Добавить роль

Роль - "Учитель"

Название teacher Описание Учитель Сохранить изменения в роли

Связанные задачи

☐ Роли ☐ Профиль ☐ Сдача экзамена ☒ Результаты экзамена ☐ Учителя ☐ Студенты ☐ Директоры ☐ Предметами ☒ Темы предметов

☐ Классы ☐ Предметы учителя ☒ Вопросы ☒ Экзамены ☒ Профиль учителя ☒ Группы ☐ Профиль ученика ☐ Профиль директора

☐ Учебное заведение ☐ История экзаменов студента ☒ Проверкой экзамена

3.18 сурет - Мұғалімнің міндеттері

Мұғалім беретін пәні бойынша тақырып қосады.

Школа № 70

Физика

Название темы	Действие	Действие
Ньютон заңдары		

3.19 сурет - Тақырып қосу

Емтихан немесе тест өткізу үшін сұрақтар қосады. Сұрақтардың екі түрі бар: ашық және тест түрінде. Сұрақтардың үш деңгейі көрсетілген: оңай, орташа және қиын. Қойылатын балл сұрақ деңгейіне байланысты болады.

Создать вопрос

Список вопросов

Предмет:

Физика

Тема:

Ньютон заңдары

Год обучения:

9

Язык:

Казахский

Тип вопроса:

Открытый вопрос

Вопрос:

</>

B

I

≡

≡

≡

≡

A

T

≡

≡

≡

≡

≡

≡

≡

≡

↶

↷

↶

↷

↶

↷

Ньютонның бірінші заңы қалай тұжырымдалады?

≡

Chars: 39 Words: 5

Корректный ответ:

</>

B

I

≡

≡

≡

≡

A

T

≡

≡

≡

≡

≡

≡

≡

≡

↶

↷

↶

↷

↶

↷

Егер денеге басқа денелер әсер етпесе немесе әсерлер теңгерілсе, онда дене тыныштық күйін сақтайды немесе түзу сызықты бірқалыпты қозғалысын жалғастырады.

≡

Chars: 135 Words: 19

Уровень:

сложный

Балл за правильный ответ:

5

3.20 сурет - Ашық сұрақ қосу

* Тема:

* Год обучения:

* Язык:

* Тип вопроса:

* Вопрос:

</>

В

I

T

I

Ньютон екінші заңының формуласы

Chars: 28 Words: 4

* Вариант ответа 1:

</>

В

I

T

I

a=F/m

Chars: 5 Words: 1

Правильный вариант: ☒

3.21 сурет - Тест сұрақтарын қосу

Барлық қосқан сұрақтарды сұрақтар тізімінен көруге болады. Көріп қана қоймай сұрақты өзгертуге немесе өшіруге болады.

Создать вопрос

Список вопросов

Физика

Ньютон заңдары

Вопрос	Действие	Действие
Ньютонның бірінші заңы қалай тұжырымдалады?		
Ньютон екінші заңының формуласы		
2 м/с² үдеумен лифт арқылы жоғары көтеріліп бара жатқан массасы 70 кг адамның салмағы қаншаға артады?		
Құш		
Еркін түсу үдеуінің шамасы		

3.22 сурет - Сұрақтар тізімі

Сұрақтарды студенттерге тарату үшін группа құратын боламыз. Бұл жерде әр класс оқушыларынан немесе бір класс оқушыларынан группа құруға болады.

[Создать группу](#) [Список групп](#)

* Название:

* Список студентов:

Аяжан Айтуганкызы Уали ×

Бакдаулет Алтынбекулы Жанузак × |

3.23 сурет – Студенттерді группаға қосу

Создание экзамена

* Наименование экзамена:

Физика экзамен

* Предмет:

Физика

* Год обучения:

9

* Язык:

Казахский

* Группа:

Физика 1гр

* Название темы:

Ньютон заңдары

* Количество вариантов:

2

* Дата начала экзамена:

25/04/2021

* Время начала экзамена:

15:50

* Дата завершения экзамена:

25/04/2021

* Время завершения экзамена:

16:30

* Продолжительность экзамена в минутах:

40

* Общее количество вопросов:

5

3.24 суретте экзаменнің қалай қосылатыны көрсетілген.

Тип вопроса:

Открытый вопрос

* Количество вопросов:

1

* Уровень А:

0

* Уровень В:

0

* Уровень С:

1

Удалить

Тип вопроса:

Тестовый вопросы

* Количество вопросов:

4

* Уровень А:

3

* Уровень В:

1

* Уровень С:

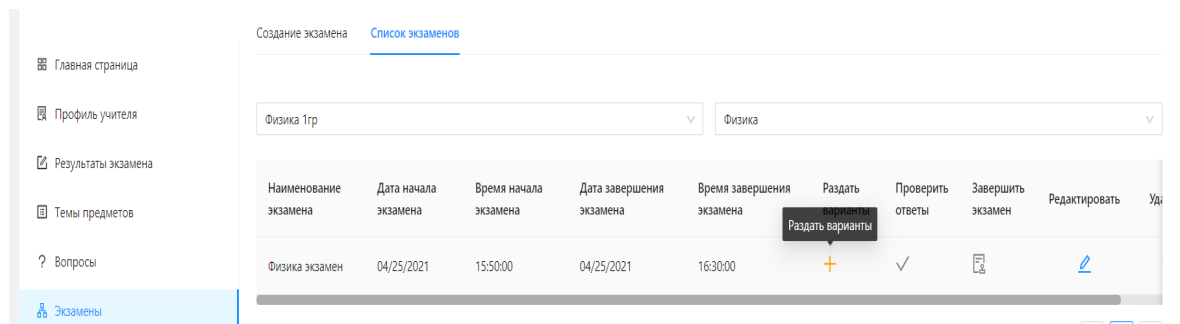
0

Удалить

+ Добавить еще

3.24 сурет - Экзамен қосу

Экзамен қосылғаннан кейін экзаменді бастау үшін мұғалім нұсқаларды тарату батырмасын басады. Бұл жерде нұсқалар автоматты түрде таратылады.



3.25 сурет - Нұсқа тарату

Емтихан аяқталғаннан кейін мұғалім студенттердің жұмысын тексереді.

Проверка экзаменов

Группа: Предмет: Экзамен:

Уали Аяжан Айтуганқызы	Номер варианта-1	Проверить ответы
Жанузак Бақдаулет Алтынбекулы	Номер варианта-2	Проверить ответы

3.26 сурет - Өткізілген экзаменді тексеру

Төменде келтірілген суреттерде студенттің сұрақтарға қалай жауап бергенін көруге болады. Ашық сұрақта дұрыс немесе дұрыс емес деген жауап ғана қабылданады.

Уали Аяжан Айтуганқызы

Еркін түсу үдеуінің шамасы

Ответ

10 Н

9,8Н/м

Корректный ответ 9,8м/с²

Ответ студента

10м/с²

Корректный ответ

3.27 сурет - Жауаптар тізімі

Ньютонның бірінші заңы қалай тұжырымдалады?

Ответ

Егер денеге сырттан күш әсер етпесе, онда ол тыныштық күйін немесе бірқалыпты түзу сызықты қозғалыстағы күйін сақтайды

Ответ правильный

☒

Закончить Назад

3.27.1 сурет - Жауаптар тізімі

Уали Аяжан Айтуганкызы

2 м/с² үдеумен лифт арқылы жоғары көтеріліп бара жатқан массасы 70 кг адамның салмағы қаншаға артады?

Ответ

Корректный ответ 140 Н

Ответ студента 35Н

130Н

120 Н

Некорректный ответ

3.27.2 сурет - Жауаптар тізімі

Мұғалім экзаменді аяқтап, экзамен нәтижелерін көре алады.

Создание экзамена

[Список экзаменов](#)

Физика 1гр

Физика

Наименование экзамена	Дата начала экзамена	Время начала экзамена	Дата завершения экзамена	Время завершения экзамена	Раздать варианты	Проверить ответы	Завершить экзамен	Редактировать	Удалить
Физика экзамен	04/25/2021	15:50:00	04/25/2021	16:30:00	+	✓			

3.28 сурет - Экзаменді аяқтау

Результаты экзамена

Физика

Физика 1гр

Физика экзамен

Фамилия	Имя	Отчество	Номер варианта	Набранный балл	Максимальный балл	Список вопросов
Уали	Аяжан	Айтуганкызы	1	10	12	Просмотр
Жанузак	Бакдаулет	Алтынбекулы	2	12	12	Просмотр

< 1 >

3.29 сурет - Экзамен нәтижелері

Студенттің(оқушы) негізгі міндеттерін 3.30 суреттен көре аламыз

Привязать задачи к ролям

Ученик

Учитель

Директор

Администратор

Добавить роль

Роль - "Ученик"

Название

student

Описание

Ученик

Сохранить изменения в роли

Связанные задачи

☐ Роли

☐ Профиль

☒ Сдача экзамена

☐ Результаты экзамена

☐ Учителя

☐ Студенты

☐ Директоры

☐ Предметами

☐ Темы предметов

☐ Классы

☐ Предметы учителя

☐ Вопросы

☐ Экзамены

☐ Профиль учителя

☐ Группы

☒ Профиль ученика

☐ Профиль директора

☐ Учебное заведение

☒ История экзаменов студента

☐ Проверкой экзамена

3.30 сурет - Студенттің жүйедегі міндеттері

Студент өзінің профилінен емтихан уақытын көре алады (3.31 сурет).

2021 Apr Месяц Год

Su	Mo	Tu	We	Th	Fr	Sa
28				01	02	03
04				07	08	09
11				14	15	16
18	19	20	21	22	23	24
25	26	27	28	29	30	01
02	03	04	05	06	07	08

• Физика экзамен (...)

Расписание экзаменов

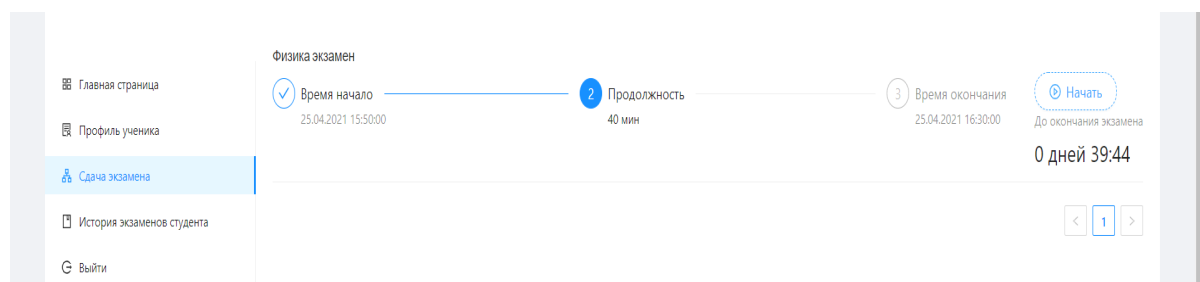
Предмет	Дата начала экзамена	Время начало
Физика экзамен	25.04.2021	15:50:00

< 1 >

OK

3.31 сурет - Экзамен басталу уақыты

Емтихан тапсыру бөліміне өтіп емтиханды тапсырады.



3.32 сурет - Экзамен мәртебесі

Экзамен уақыты басталғаннан кейін «Начать» бастырмасын басу арқылы сұрақтар тізіміне өтеді және емтихан сұрақтарына жауап беріп бастайды. «Далее» батырмасын басу арқылы келесі сұраққа өтеді. Барлық сұраққа жауап беріп болған жағдайда «Закончить» батырмасын басып емтиханды аяқтайды.

Предмет	Физика	Номер варианта	1
---------	--------	----------------	---

1 — 2 — 3 — 4 — 5

Сложность уровня вопроса - A

Еркін түсу үдеуінің шамасы

Ответ

☐ 10 Н

☐ 9,8Н/м

☒ 9,8м/с²

☐ 10м/с²

Далее

3.33 сурет - Студенттің емтихан сұрақтарына жауап беруі

✓ ————— ✓ ————— ✓ ————— ✓ ————— 5

Сложность уровня вопроса - C

Ньютонның бірінші заңы қалай тұжырымдалады?

Ответ

Егер денеге сырттан күш әсер етпесе, онда ол тыныштық күйін немесе бірқалыпты түзу сызықты қозғалыстағы күйін сақтайды

Chars: 102 Words: 17

Закончить Назад

3.34 сурет - Студенттің емтихан сұрақтарына жауап беруі

Емтихан мұғалім тарапынан тексеріліп біткен соң ғана студент өзінің алған нәтижесін көре алады (3.35 сурет).

Главная страница	Физика	Физика экзамен	
Профиль ученика			
Сдача экзамена			
История экзаменов студента			
Выйти			

Наименование экзамена	Дата начала экзамена	Предмет	Набранный балл
Физика экзамен	04/25/2021	Физика	10 / 12

< 1 >

3.35 сурет – Студенттің емтихан нәтижесі.

ҚОРЫТЫНДЫ

Бұл дипломдық жұмыста «Білім беру ұйымдарында тестілер өткізуге арналған» веб қосымша жасалынып шықты. Бұл веб қосымшаның мақсаты оқытушылар мен білім алушылардың уақытын тиімді өткізу, білім алушының оқу үдерісін бақылау болып табылады. Веб қосымшаның ең негізгі артықшылықтары:

- бұл жүйе - мектеп, колледж, университетке арналып жасалған;
- жүйеде мектепті, мектеп директорын, мұғалімдерді, пәндерді енгізуге және емтихан немесе тест өткізуге мүмкіндік бар;
- емтихан немесе тест өтілгеннен кейін оқытушы мен білім алушы жүйеден емтиханнан алған нәтижесін көре алады.

Веб қосымшаны құру кезінде жүйеге қойлатын негізгі талаптар белгіленді. Талаптар негізінде жүйені іске асыру үшін прецеденттер диаграммасы мен деректер қорының ER диаграммасы құрылды.

Веб қосымша қазіргі таңда өзекті болып табылады. Себебі, бұл веб қосымша білім беру жүйесін автоматтандыруға өз үлесін қосады және осындай веб қосымшаны қолдану уақытты тиімді пайдалануға мүмкіндік береді. Веб қосымша барлық қолданушыларға ыңғайлы әрі түсінікті етіп жасалынған.

Қорыта келгенде, осы дипломдық жұмыстың нәтижесінде C#, React, тілдері мен PostgreSQL дерекқорын қолдану арқылы берілген жұмыс орындалды. Жоғарыда келтірілген дипломдық жұмысқа қойылған талаптар орындалып, дипломдық жұмыстың басты мақсатына қол жеткізілді. Осы веб-қосымша білім беру ұйымдарын дамытуға өз үлесін қосады деп ойлаймын.

ПАЙДАЛАНЫЛҒАН ӘДЕБИЕТЕР ТІЗІМІ

- 1 Айтхожаева Е.Ж. Системы баз данных. Учебник. - Алматы, КазНТУ, 2002
- 2 Холл, М. Программирование для Web. Библиотека профессионала: учебник / М. Холл, Л. Браун. – М.: Вильямс, 2002
- 3 Хомоненко, А. Д. Базы данных: учебник / А. Д. Хомоненко, В.М. Цыганков, М.Г. Мальцев. – СПб.: КОРОНА-Век, 2009
- 4 Н.А. Гайдамакин Автоматизированные информационные системы, базы и банки данных, М.: «Гелиос», 2002
- 5 <https://ru.wikipedia.org/wiki/React>
- 6 https://ru.wikipedia.org/wiki/C_Sharp
- 7 П.Лузанов, Е.Рогов, И.Левшин PostgreSQL для начинающих, 2017
- 8 К. Максим, С. Игорь, Г. Сергей. PHP 5: Веб-сайттар құру тәжірибесі СПб.: 2007. – 139 б.
- 9 PostgreSQL.Для профессионалов. Дж.Уорсли, Дж.Дрейк, 2003 г
- 10 М.Пацианский «React.js для начинающих», 2016
- 11 Марк Тиленс Томас «React в действии», 2017.

А қосымшасы

LoginController.cs – аутентификация, авторизация, регистрация контроллері

```
using Intranet.Models;
using Microsoft.AspNetCore.Authentication.JwtBearer;
using Microsoft.AspNetCore.Authorization;
using Microsoft.AspNetCore.Http;
using Microsoft.AspNetCore.Mvc;
using Microsoft.EntityFrameworkCore;
using Microsoft.Extensions.Configuration;
using Microsoft.IdentityModel.Tokens;
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 Intranet.Controllers
{
    [Route("api/[controller]")]
    [ApiController]
    public class LoginController : ControllerBase
    {
        private readonly IConfiguration _config;
        private readonly BaseDbContext _baseDB;
        public LoginController(IConfiguration config, BaseDbContext context)
        {
            _config = config;
            _baseDB = context;
        }
        [HttpPost]
        public IActionResult Login([FromBody] Users loggedUser)
        {
            IActionResult response = Unauthorized();
            var user = Authenticate(loggedUser);
            if (user != null)
            {
                var token = GenerateJWT(user);
                response = Ok(new { token });
            }
            return response;
        }
    }
}
```

А қосымшасының жалғасы

```
private IdentityUser Authenticate(Users loggedUser)
{
    IdentityUser user = null;
    string password = Encryptor.MD5Hash(loggedUser.Password);
    if (_baseDB.Users.Any(m => m.Login == loggedUser.Login.ToLower() &&
m.Password == password))
    {
        Users u = _baseDB.Users.FirstOrDefault(m => m.Login ==
loggedUser.Login.ToLower() && m.Password == password);
        int? rolesId = _baseDB.UserRoles.FirstOrDefault(m => m.UserId ==
u.Id).RoleId;
        string role = _baseDB.Roles.FirstOrDefault(m => m.Id == rolesId).Value;
        u.LastLoginDate = DateTime.Now;
        _baseDB.Entry(u).State = EntityState.Modified;
        А қосымшасының жалғасы

_baseDB.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:Key"]));
    var credentials = new SigningCredentials(securityKey,
SecurityAlgorithms.HmacSha256);
    var claims = new[] {
        new Claim(JwtRegisteredClaimNames.Sub, user.Login),
        new Claim(JwtRegisteredClaimNames.Email, user.Email),
        new Claim(JwtRegisteredClaimNames.NameId, user.Id.ToString()),
        new Claim(JwtRegisteredClaimNames.Jti, Guid.NewGuid().ToString()),
        new Claim(ClaimsIdentity.DefaultRoleClaimType, user.Role)
    };
};
```

А қосымшасының жалғасы

```
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;  
  
}  
  
}
```

Б қосымшасы

UserController – қолданушылардың басты контроллері

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Security.Claims;
using Intranet.Controllers.Utills;
using Intranet.Models;
using Microsoft.AspNetCore.Authorization;
using Microsoft.AspNetCore.Mvc;
using Microsoft.EntityFrameworkCore;
using Microsoft.Extensions.Configuration;
using Microsoft.Extensions.Localization;
using Microsoft.Net.Http.Headers;
using Intranet.Controllers.ModelsController;
using Microsoft.AspNetCore.Http;

namespace Intranet.Controllers
{
    [Route("api/[controller]")]
    [ApiController]
    public class UserController : ControllerBase
    {

        private readonly BaseDbContext _baseDB;
        private readonly IStringLocalizer<Localization> _localizer;
        private readonly IConfiguration _config;

        public UserController(BaseDbContext context,
                             IStringLocalizer<Localization> localizer,
                             IConfiguration config)
        {
            _baseDB = context;
            _localizer = localizer;
            _config = config;
        }

        [HttpGet]
        [Route("userMain")]

        public IEnumerable<Users> GetUserList() => _baseDB.Users.ToList();
        //[Authorize(Roles = "teacher")]
    }
}
```

Б қосымшасының жалғасы

```
[HttpGet]
public Users Get(int id)
{
    var identity = HttpContext.User.Identity as ClaimsIdentity;
    IList<Claim> claim = identity.Claims.ToList();
    return _baseDB.Users.SingleOrDefault(m => m.Id == id);
}
```

```
[HttpGet]
[Route("userTasks")]
public List<Tasks> UserTasks(int userId)
{
    int? rolesId = _baseDB.UserRoles.FirstOrDefault(m => m.UserId ==
userId).RoleId;

    var tasks = _baseDB.RoleTasks
        .Where(m => m.RoleId == rolesId)
        .Join(_baseDB.Tasks, x => x.TaskId, y => y.Id, (x, y) => y)
        .OrderBy(m => m.Priority)
        .ToList();
    return tasks;
}
```

```
[HttpGet]
[Route("isloginexist")]
public bool IsLoginExist(string id)
{
    if (id != null)
    {
        if (_baseDB.Users.Any(m => m.Login == id.ToLower()))
        {
            return true;
        }
        return false;
    }
    else
    {
        return false;
    }
}
```

Б қосымшасының жалғасы

```
}

[Authorize()]
[HttpPut("changePassword")]
public IActionResult ChangePassword(ChangePassword request)
{
    var token =
Request.Headers[HeaderNames.Authorization].ToString().Replace("Bearer ", "");
    var id = JwtHelper.GetUserId(token, _config["Jwt:Key"]);
    if (!string.IsNullOrEmpty(id))
    {
        try
        {
            Users user = _baseDB.Users.SingleOrDefault(m => m.Id ==
int.Parse(id));
            if (user != null)
            {
                user.PrevPassword = user.Password;
                user.Password = Encryptor.MD5Hash(request.password);
                user.LastChangePassDate = DateTime.Now;
                _baseDB.Entry(user).State = EntityState.Modified;
                _baseDB.SaveChanges();
            }
            return Ok();
        }
        catch
        {
        }
    }
    return NotFound();
}

[HttpPost]
public Response Post([FromBody] Users user)
{
    Response response;
    try
    {
        _baseDB.Users.Add(user);
        _baseDB.SaveChanges();
        response = new Response() { code = 0, message = "Ok" };
    }
    catch (Exception ex)
```

Б қосымшасының жалғасы

```
{
    response = new Response() { code = -1, message = ex.Message };
}
return response;
}

[HttpGet("[action]")]
public IActionResult GetLoggedInUser()
{
    try
    {
        var token =
Request.Headers[HeaderNames.Authorization].ToString().Replace("Bearer ", "");
        var role = JwtHelper.GetUserRole(token, _config["Jwt:Key"]);
        var login = JwtHelper.GetUserLogin(token, _config["Jwt:Key"]);
        List<string> userInfo = new List<string>();
        userInfo.Add(login);

        if (role.Equals("director"))
        {
            AuthorizedPersons authorizedPerson =
_baseDB.AuthorizedPersons.SingleOrDefault(i => i.Users.Login.Equals(login));
            userInfo.Add(authorizedPerson.LastName + ' ' +
authorizedPerson.FirstName + ' ' + authorizedPerson.MiddleName);
            return Ok(userInfo);
        }
        else if (role.Equals("teacher"))
        {
            Teachers teacher = _baseDB.Teachers.SingleOrDefault(i =>
i.Users.Login.Equals(login));
            userInfo.Add(teacher.LastName + ' ' + teacher.FirstName + ' ' +
teacher.MiddleName);
            return Ok(userInfo);
        }
        else if (role.Equals("student"))
        {
            Students student = _baseDB.Students.SingleOrDefault(i =>
i.Users.Login.Equals(login));
```

Б қосымшасының жалғасы

```
userInfo.Add(student.LastName + ' ' + student.FirstName + ' ' +
student.MiddleName);
    return Ok(userInfo);          }
    return Ok(userInfo);
}
catch (Exception ex)
{
    return StatusCode(StatusCodes.Status500InternalServerError,
        new ActionResponse
        {
            Code = 500,
            Message = _localizer["InternalServerError"].Value,
            DeveloperMessage = ex.InnerException.Message
        });
}
}
}
}
```

С қосымшасы

ExamController.cs – экзамен өткізу, тексеру контроллері

```
using System;
using System.Collections.Generic;
using System.Linq;
using Intranet.Controllers.ModelsController;
using Intranet.Controllers.Utills;
using Intranet.Models;
using Microsoft.AspNetCore.Authorization;
using Microsoft.AspNetCore.Http;
using Microsoft.AspNetCore.Mvc;
using Microsoft.EntityFrameworkCore;
using Microsoft.Extensions.Configuration;
using Microsoft.Extensions.Localization;
using Microsoft.Extensions.Logging;
using Microsoft.Net.Http.Headers;

namespace Intranet.Controllers
{
    [Authorize(Roles = "teacher")]
    [Route("api/[controller]")]
    [ApiController]
    public class ExamsController : ControllerBase
    {
        private readonly BaseDbContext _dbContext;
        private readonly IConfiguration _config;
        private readonly IStringLocalizer<Localization> _localizer;
        private readonly ILogger<ExamsController> _logger;

        public ExamsController(BaseDbContext dbContext,
                               IConfiguration config,
                               IStringLocalizer<Localization> localizer,
                               ILogger<ExamsController> logger)
        {
            _dbContext = dbContext;
            _config = config;
            _localizer = localizer;
            _logger = logger;
        }

        [HttpGet("{id}")]
        public IActionResult GetById(int id)
        {
```

С қосымшасының жалғасы

```
var exam = _dbContext.Exams.SingleOrDefault(e => e.Id == id);

    if (exam != null)
        return Ok(exam);
    else
        return NotFound();
}

[HttpPost]
public IActionResult Post([FromBody] Exams exam)
{
    try
    {
        _logger.LogInformation("Exam Creating");

        var token =
Request.Headers[HeaderNames.Authorization].ToString().Replace("Bearer ",
    "");

        var login = JwtHelper.GetUserLogin(token, _config["Jwt:Key"]);
        TimeZoneInfo localZone = TimeZoneInfo.Local;
        exam.TimeCoordinates = localZone.DisplayName;
        exam.TeacherId = _dbContext.Teachers.FirstOrDefault(t =>
t.Users.Login.Equals(login)).Id;

        _dbContext.Exams.Add(exam);
        _dbContext.SaveChanges();

        CreateVariants(exam);

        return Ok();
    }
    catch (Exception exception)
    {
        _logger.LogInformation(string.Format("Exam Creating error-{0}",
exception.Message));

        return StatusCode(StatusCodes.Status500InternalServerError,
            new ActionResponse
            {
                Code = 500,
                Message = _localizer["InternalServerError"].Value,
                DeveloperMessage = exception.Message
            });
    }
}
```

С қосымшасының жалғасы

```
}  
}  
  
[HttpPut]  
public IActionResult Put([FromBody] Exams exam)  
{  
    try  
    {  
        TimeZoneInfo localZone = TimeZoneInfo.Local;  
        exam.TimeCoordinates = localZone.DisplayName;  
        _dbContext.Exams.Update(exam);  
        _dbContext.SaveChanges();  
  
        var variants = _dbContext.ExamVariants.Where(v => v.ExamId ==  
exam.Id);  
        if (variants != null)  
        {  
            _dbContext.ExamVariants.RemoveRange(variants);  
            _dbContext.SaveChanges();  
  
            CreateVariants(exam);  
        }  
  
        return Ok();  
    }  
    catch (Exception exception)  
    {  
        return StatusCode(StatusCodes.Status500InternalServerError,  
            new ActionResponse  
            {  
                Code = 500,  
                Message = _localizer["InternalServerError"].Value,  
                DeveloperMessage = exception.Message  
            });  
    }  
}  
  
[HttpDelete("{id}")]  
public IActionResult DeleteById(int id)  
{  
    var exam = _dbContext.Exams.SingleOrDefault(e => e.Id == id);  
  
    if (exam != null)
```

С қосымшасының жалғасы

```
{
    try
    {
        var variants = _dbContext.ExamVariants.Where(v => v.ExamId
== exam.Id);
        if (variants != null)
        {
            _dbContext.ExamVariants.RemoveRange(variants);
            _dbContext.SaveChanges();
        }

        _dbContext.Exams.Remove(exam);
        _dbContext.SaveChanges();

        return Ok();
    }
    catch (Exception exception)
    {
        return StatusCode(StatusCodes.Status500InternalServerError,
            new ActionResponse
            {
                Code = 500,
                Message = _localizer["InternalServerError"].Value,
                DeveloperMessage = exception.Message
            });
    }
}
else return NotFound();
}

private void CreateVariants(Exams exam)
{
    var variants = new List<ExamVariants>();

    var options = exam?.Options.ToList().Where(o => o.Amount > 0);

    for (int numberVariant = 1; numberVariant <=
exam.NumberOfVariants; numberVariant++)
    {
        var questions = new List<int?>();

        foreach (var option in options)        {
```

С қосымшасының жалғасы

```
var questionsIds = new List<int?>();

        questionsIds = _dbContext.Questions
            .Where(q => q.SubjectId == exam.SubjectId && q.LangId ==
exam.LangId &&
                q.TeacherId == exam.TeacherId && q.YearOfStudy ==
exam.YearOfStudy &&
                q.QuestionLevel == option.Level && q.QuestionTypeId ==
option.QuestionTypeId &&
                exam.TopicsId.Contains(q.TopicId))
            .Select(q => q.Id).ToList();

        if (questionsIds.Count > 0)
        {
            questions.AddRange(
                questionsIds.OrderBy(q =>
Guid.NewGuid()).Take(option.Amount));
        }
    }

    if (questions.Count > 0)
    {
        variants.Add(new ExamVariants()
        {
            ExamId = exam.Id,
            NumberVariant = numberVariant,
            QuestionIds = questions.ToArray()
        });
    }
}

if (variants != null)
{
    _dbContext.ExamVariants.AddRange(variants);
    _dbContext.SaveChanges();

    exam.Status = 2;
    _dbContext.Exams.Update(exam);
    _dbContext.SaveChanges();
}
}

[HttpPost("DistributeVariants/{id}")]
```

С қосымшасының жалғасы

```
public IActionResult DistributeVariants(int id)
{
    try
    {
        var exam = _dbContext.Exams.SingleOrDefault(e => e.Id == id);
        if (exam != null)
        {
            var existing = _dbContext.StudentExams.Where(se => se.ExamId
== exam.Id).ToList();
            if (existing.Count == 0)
            {
                var variants = _dbContext.ExamVariants.Where(v => v.ExamId
== exam.Id)
                .OrderBy(v => v.NumberVariant).ToArray();

                int variantsCount = variants.Length;

                if (variantsCount > 0)
                {
                    var results = new List<StudentExams>();

                    var studentsIds = _dbContext.StudentGroups.Where(sg =>
sg.GroupId == exam.GroupId)
                    .Select(sg => sg.StudentId).ToList();

                    int variantId = 0;
                    studentsIds.ForEach(sid =>
                    {
                        results.Add(new StudentExams()
                        {
                            StudentId = sid,
                            ExamId = exam.Id,
                            VariantId = variants[variantId].Id,
                            VariantNumber = variants[variantId].NumberVariant,
                            Finished = 0,
                            QuestionIds = variants[variantId].QuestionIds
                        });

                        variantId++;
                        if (variantId == variantsCount) variantId = 0;
                    });

                    if (results.Count > 0)
```

С қосымшасының жалғасы

```
{
    _dbContext.StudentExams.AddRange(results);
    _dbContext.SaveChanges();

    exam.Status = 3;
    _dbContext.Exams.Update(exam);
    _dbContext.SaveChanges();
}
}
}
}
return Ok();
}
catch (Exception exception)
{
    _logger.LogInformation(string.Format("Exam DistributeVariants
error-{0}", exception.Message));

    return StatusCode(StatusCode.Status500InternalServerError,
        new ActionResponse
        {
            Code = 500,
            Message = _localizer["InternalServerError"].Value,
            DeveloperMessage = exception.Message
        });
}

}

[HttpGet("myExamsAsTeacher")]
public IActionResult GetTeacherExams()
{
    try
    {
        var token =
Request.Headers[HeaderNames.Authorization].ToString().Replace("Bearer ",
""");
        var login = JwtHelper.GetUserLogin(token, _config["Jwt:Key"]);
        var teacher = _dbContext.Teachers.SingleOrDefault(s =>
s.Users.Login.Equals(login));
        var teacherExams = _dbContext.Exams.Where(exam =>
exam.TeacherId == teacher.Id);
        return Ok(teacherExams);
    }
}
```

С қосымшасының жалғасы

```
}  
    catch (Exception ex)  
    {  
        return StatusCode(StatusCodes.Status500InternalServerError,  
            new ActionResponse  
            {  
                Code = 500,  
                Message = _localizer["InternalServerError"].Value,  
                DeveloperMessage = ex.Message,  
            });  
    }  
}  
  
[HttpGet("myFilteredExamsAsTeacher/{groupId}/{subjectId}")]  
public IActionResult GetTeacherExamsByFilter(int groupId, int  
subjectId)  
{  
    try  
    {  
        var token =  
Request.Headers[HeaderNames.Authorization].ToString().Replace("Bearer ",  
"" );  
        var login = JwtHelper.GetUserLogin(token, _config["Jwt:Key"]);  
        var teacher = _dbContext.Teachers.SingleOrDefault(s =>  
s.Users.Login.Equals(login));  
        var teacherExams = _dbContext.Exams.Where(exam =>  
            exam.TeacherId == teacher.Id &&  
            exam.GroupId == groupId &&  
            exam.SubjectId == subjectId &&  
            exam.Status == 6);  
  
        return Ok(teacherExams);  
    }  
    catch (Exception ex)  
    {  
        return StatusCode(StatusCodes.Status500InternalServerError,  
            new ActionResponse  
            {  
                Code = 500,  
                Message = _localizer["InternalServerError"].Value,  
                DeveloperMessage = ex.Message,  
            });  
    }  
}
```


С қосымшасының жалғасы

```
[HttpGet("[action]/{groupId}/{subjectId}")]
public IActionResult GetFilteredTeacherExams(int groupId, int subjectId)
{
    try
    {
        var token =
Request.Headers[HeaderNames.Authorization].ToString().Replace("Bearer ",
    "");
        var login = JwtHelper.GetUserLogin(token, _config["Jwt:Key"]);
        var teacher = _dbContext.Teachers.SingleOrDefault(s =>
s.Users.Login.Equals(login));
        List<Exams> teacherExams = _dbContext.Exams
        .Where(exam =>
            exam.TeacherId == teacher.Id &&
            exam.GroupId == groupId &&
            exam.SubjectId == subjectId).ToList();

        return Ok(teacherExams);
    }
    catch (Exception ex)
    {
        return StatusCode(StatusCodes.Status500InternalServerError,
            new ActionResponse
            {
                Code = 500,
                Message = _localizer["InternalServerError"].Value,
                DeveloperMessage = ex.Message,
            });
    }
}

[HttpGet("passedExams/{groupId}/{subjectId}")]
public IActionResult PassedExams(int groupId, int subjectId)
{
    try
    {
        var token =
Request.Headers[HeaderNames.Authorization].ToString().Replace("Bearer ",
    "");
        var login = JwtHelper.GetUserLogin(token, _config["Jwt:Key"]);
```

С қосымшасының жалғасы

```
var teacher = _dbContext.Teachers.SingleOrDefault(s => s.Users.Login == login);
    if (teacher != null)
    {
        var exams =
            _dbContext.Exams.Where(e => e.TeacherId == teacher.Id &&
                e.GroupId == groupId &&
                e.SubjectId == subjectId &&
                e.Status == 3).ToList();

        return Ok(exams);
    }
    else
        return NotFound();
}
catch (Exception exception)
{
    return StatusCode(StatusCode.Status500InternalServerError,
        new ActionResponse
        {
            Code = 500,
            Message = _localizer["InternalServerError"].Value,
            DeveloperMessage = exception.Message
        });
}
}

[HttpPut("finishExam/{examId}")]
public IActionResult FinishExam(int examId)
{
    try
    {
        var exam = _dbContext.Exams.SingleOrDefault(e => e.Id == examId);
        if (exam == null) return NotFound();

        var studentExams = _dbContext.StudentExams.Where(x => x.ExamId == examId).ToList();
        if (studentExams.Count() == 0) return NotFound();

        var examEndTime = exam.EndDate.Value.Add(exam.EndTime.Value);
        var examExpired = examEndTime < DateTime.Now;
```

С қосымшасының жалғасы

```
var all = studentExams.Count();
var ready = studentExams.Where(se => se.Status == 0).Count();
var started = studentExams.Where(se => se.Status == 1).Count();
var finished = studentExams.Where(se => se.Status == 2).Count();
    var checkedByTeacher = studentExams.Where(se => se.Status ==
3).Count();

    if (finished > 0) return Ok("NotAllExamsChecked");

        if (!examExpired && ready + started > 0) return
Ok("NotAllExamsPassed");

    exam.Status = 6;
    _dbContext.Exams.Update(exam);
    _dbContext.SaveChanges();

    studentExams.ForEach(se =>
    {
        if (se.Status == 0) se.Status = 4;
        else if (se.Status == 1) se.Status = 5;
    });
    _dbContext.StudentExams.UpdateRange(studentExams);
    _dbContext.SaveChanges();

    return Ok();
}
catch (Exception exception)
{
    return StatusCode(StatusCodes.Status500InternalServerError,
        new ActionResponse
        {
            Code = 500,
            Message = _localizer["InternalServerError"].Value,
            DeveloperMessage = exception.Message
        });
}
}
}
}
```

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: Жанұзақ Таңнұр

Название: Білім беру ұйымдарында тестілер өткізуге арналған веб-қосымшаны әзірлеу

Координатор: Дүйсенбаева Асемгуль

Коэффициент подобия 1: 14,76

Коэффициент подобия 2: 5,33

Замена букв: 9

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:

Заимствования являются добросовестными и не обладают признаками плагиата.

24.05.2021

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: Жанұзақ Таңнұр

Название: Білім беру ұйымдарында тестілер өткізуге арналған веб-қосымшаны әзірлеу

Координатор: Дүйсенбаева Асемгуль

Коэффициент подобия 1: 14,76

Коэффициент подобия 2: 5,33

Замена букв: 9

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 являются добросовестными и не обладают признаками плагиата.

Дата 31.05.2021



Сейлова Н.А.

зав.кафедрой КБОиХИ

