

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ ҒЫЛЫМ ЖӘНЕ БІЛІМ МИНИСТРЛІГІ

Қ.И.Сәтбаев атындағы Қазақ ұлттық техникалық зерттеу университеті

Автоматика және ақпараттық технологиялар институты

«Программалық инженерия» кафедрасы

Әбдірешев Бағлан Жеңісұлы

Дайындық орталығы үшін ақпараттық жүйені жобалау және әзірлеу

ТҮСІНДІРМЕ ЖАЗБА

дипломдық жобаға

5B070400 – «Есептеу техникасы және бағдарламалық қамтамасыз ету»

Алматы 2022

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ ҒЫЛЫМ ЖӘНЕ БІЛІМ МИНИСТРЛІГІ

Қ.И.Сәтбаев атындағы Қазақ ұлттық техникалық зерттеу университеті


Автоматика және ақпараттық технологиялар институты

«Программалық инженерия» кафедрасы



ҚОРҒАУҒА ЖІБЕРІЛДІ

ПИ кафедрасының меңгерушісі
физ.-мат. ғыл.канд, профессор

 А.Н. Молдагулова

« 20 » 05 2022 г.

ТҮСІНДІРМЕ ЖАЗБА

дипломдық жобаға

Тақырыбы: «Дайындық орталығы үшін ақпараттық жүйені жобалау және әзірлеу»

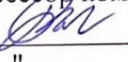
5B070400 – «Есептеу техникасы және бағдарламалық қамтамасыз ету»
мамандығы


Орындаған

Әбдірешев Б.Ж.

Рецензент
техника ғылымдар кандидаты,
профессор көмекшісі

Ғылыми жетекші
лектор

 Куламбаев Б. О.
" " " 2022 ж.

 Омаров Б.С.
" " " 2022 ж.

Алматы 2022

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ ҒЫЛЫМ ЖӘНЕ БІЛІМ МИНИСТРЛІГІ

Қ.И.Сәтбаев атындағы Қазақ ұлттық техникалық зерттеу университеті

Автоматика және ақпараттық технологиялар институты

«Программалық инженерия» кафедрасы



БЕКІТЕМІН

ПИ кафедрасының меңгерушісі

физ.-мат. ғыл.канд, профессор

М.А. А.Н. Молдагулова

« 20 » 05 2022 г.

**Дипломдық жобаны орындауға
ТАПСЫРМА**

Білім алушыға Әбдірешев Бағлан Жеңісұлы

Тақырыбы: «Дайындық орталығы үшін ақпараттық жүйені жобалау және әзірлеу»

Академиялық мәселелер жөніндегі проректоры бұйрығының

№ 489-П/В " 24 " 12 2021 ж. шешімімен бекітілген.

Орындалған жобаның өткізу мерзімі " 25 " 05 2022 ж.

Дипломдық жобаның бастапқы мәліметтері: Жобаның мақсаты, өзектілігі, деректер қоры, жоба диаграммалары, программалық қаматамаларды жасау жүргізілген.

Есеп – түсініктеме жазбаның талқылауға берілген сұрақтардың тізімі:

а) пән саласын талдау, техникалық тапсырма әзірлеу;

б) дипломдық жобаның веб қосымшасын әзірлеу;

в) жүйе функционалдығын дамыту;

г) тестілеу және безендіру;

Графикалық материалдар тізімі (міндетті суреттердің нақты көрсетілуімен):



презентацияның 29 слайдпен берілген құжат түрінде ұсынылған.

Ұсынылған негізгі әдебиеттер: 22 пайдаланылған әдебиеттер тізімінен

Дипломдық жобаны орындау
КЕСТЕСІ

Бөлімдердің атаулары, зерттелген мәселелердің тізімі	Ғылыми жетекшіге және кеңесшілерге ұсыну мерзімі	Ескерту
1. Дипломдық жобаның жоспарын құру	14.01.2022	орындалды
2. Тапсырма қойылымы және программалау ортасын таңдау	18.01.2022	орындалды
3. Зерттеу тақырыбы бойынша ғылыми теориялық материалдарды жинау және негізгі бөлім бойынша есеп беру жазбасын дайындау	01.02.2022	орындалды
4. Дипломның екінші және үшінші бөлімдерін, жобалау сызбаларын дайындау	15.02.2022	орындалды
5. Жобаның веб-қосымшасын тестілеуден өткізу	18.03.2022	орындалды
6. Дипломдық жобаға түсіндірме жазба жазуды аяқтау	26.04.2022	орындалды

Дипломдық жұмыс бөлімдерінің кеңесшілерінің аяқталған жұмысқа қойған
қолтаңбалары

Бөлімдер атауы	Кенес берушілер (аты-жөні, тегі, ғылыми дәрежесі, атағы)	Қолтаңба қойылған мерзімі	Қолы
Нормалық бақылаушы	Жекамбаева М.Н. PhD, қауымдастырылған-профессор	18.05.22	
Бағдарламалық бөлім	Марғұлан Қ. тех.ғыл.магистрі, лектор	18.05.22	

Ғылыми жетекші _____

Омаров Б.С.

Тапсырманы орындауға қабылдаған студент _____

Әбдірешев Б.Ж.

Күні _____

«14» 11 2021

АНДАТПА

Бұл дипломдық жұмыстың мақсаты дайындық орталығы үшін ақпараттық жүйені жобалау және әзірлеу. Қазіргі кезде көптеген жерлерге адамдарды даярлауға , оқуға және де дамытуға арналған дайындық орталықтары қажет. Ақпараттық жүйе SQLITE дерекқорларды басқару жүйесінде жасалған және Python Django бағдарламалық ортасында құрылған веб - қосымшаны құрайды. Дипломдық жұмыста ақпараттық жүйені дайындау, деректер қорың құру және веб қосымшаны жасау жұмыстары жүргізілген.

Дипломдық жұмыс кіріспеден, үш негізгі бөлім және қорытындыдан құралған:

Кіріспеде жалпы тақырыптың мақсаты және жұмысы айқындалады.

Бірінші бөлімде жобаның мақсаты , жобаның өзектілігі , UML модельдеу тілі арқылы диаграммалар көрсетілген.

Екінші бөлімде «Дайындық орталығы үшін ақпараттық жүйені жобалау және әзірлеу» веб қосымшасының жобадағы қолданылған барлық бағдарламау тілдері және бағдарлама қосымшасы туралы жазылған.

Үшінші бөлімде веб қосымшаның өзі бейнеленген және барлық интерфейстерінің скриншоттары көрсетілген.

АННОТАЦИЯ

Целью данной дипломной работы является проектирование и разработка информационной системы для учебного центра. Сегодня во многих местах нужны учебные центры для обучения, воспитания и развития. Информационная система представляет собой сайт, созданный в системе управления базами данных SQLITE и созданный в программной среде Python Django. Диссертация включала разработку информационной системы, создание базы данных также создание веб-приложения.

Диссертация состоит из введения, трех основных разделов и заключения:

Введение определяет цель и работу общей темы.

В первом разделе показаны цель проекта, актуальность проекта, диаграммы с использованием языка моделирования UML.

Во втором разделе описаны все используемые в проекте языки программирования и применение веб-приложения «Проектирование и разработка информационных систем учебного центра»

В третьем разделе показано само веб-приложение и скриншоты всех интерфейсов.

ANNOTATION

The purpose of this thesis is to design and develop an information system for the training center. Today, many places need training centers for training, education and development. The information system is a site created in the SQLITE database management system and created in the Python Django software environment. Thesis included the development of an information system, creation of a database and the creation of a web application.

Thesis consists of an introduction, three main sections and a conclusion:

The introduction defines the purpose and work of the general theme.

The first section shows the purpose of the project, the relevance of the project, diagrams using the UML modeling language.

The second section describes all the programming languages and program applications used in the project of the web application "Design and development of information systems for the training center".

The third section shows the web application itself and screenshots of all interfaces.

МАЗМҰНЫ

	Кіріспе	9
1	Негізгі бөлім	10
1.1	Жобаның мақсаты	10
1.2	Терминдер мен қысқартулар	10
1.3	Жобаның өзектілігі	11
1.4	Веб қосымша жасаудың негіздері	11
1.5	Жүйені модельдеу	12
1.5.1	ER диаграммасы	12
1.5.2	USE Case диаграммасы	13
2	Технологиялар бөлімі	14
2.1	SQLite деректер базасы	14
2.2	Бағдарламау тілі	14
2.2.1	Python	14
2.2.2	Django фреймворкі	14
2.3	HTML бағдарламалау тілі	15
2.4	CSS кестелер стилі	15
2.5	JQuery , Javascript	16
2.6	Код редакторы	16
3	Жобаның құрылымы	18
3.1	«Дайындық орталығы» веб қосымшасы	18
3.1.1	«Дайындық орталығы» веб қосымшасының функционалың сипаттау	19
3.1.2	Студенттің бет көріністері	19
3.1.3	Қызметкердің бет көріністері	22
3.1.4	Админ бет көрінісі	26
3.2	SQLite деректер қорындағы ақпараттардың сақталуы	29
	Қорытынды	32
	Пайдаланылған әдебиеттер тізімі	33
	А Қосымшасы. Техникалық тапсырма	35
	Б Қосымшасы. Бағдарлама мәтіні	37

КІРІСПЕ

Бұл дипломдық жұмыста дайындық орталығы үшін ақпараттық жүйені жобалау және әзірлеу үшін веб - қосымша бағдарламасы жобаланды. Дайындық орталығы үшін ақпараттық жүйені жобалау қазіргі кезде ең қолданбалы орталықтардың бірі болып саналады. Қазіргі кезде барлық жұмыс орталықтарына белгілі бір тәжірибелік қабілеттер керек, сол қабілеттерді алу үшін дайындық орталықтары қолданылады. Дайындық орталықтары қарқынды дамуда, мысалыға айтқанда, Kaspersky lab және Kolesa academia сияқты Қазақстандағы ірі компаниялар адамдарды даярлауға жаңа білім беруге және де сәтті өткен жағдайда оларға жұмыс ұсынады, осы мысал сияқты көптеген компаниялар өз өзіне жұмыскерлерді белгілі бір немесе өзінің дайындық орталығы арқылы оқытып өзіне керек мамандарды дайындайды.

Дайындық орталықтарында қазіргі кезде келесі қызметкерлер жұмыс жасауы шарт, олар: білім немесе тәжірибе беретін мекеме осы мекемеге арнап дайындық жүргізетін қызметкер. Осы мекемеде білім немесе тәжірибе алатын оқушы. Және де осының бәрін басқаратын басты қызметкер.

1 Негізгі бөлім

1.1 Жобаның мақсаты

Жобаның мақсаты қазіргі кездегі жұмыс тәжірибесі, білімі жоқ адамдарға және қызметкер іздеп жүрген кәсіпорындарға арналған дайындық орталығының ақпараттық жүйесін құру. Негізгі қолданушыларға оқуды енді бітірген жас өспірімдерге және қызметкер ідеп жүрген кәсіпорындарға арналған. Жас өспірімдер оқу бітірген сон дайындық орталығына жазылып, керекті және нақты тәжірибелік білім алып кәсіпорындарға жұмысқа тұра алады. Бұл жоба негізінен жұмыс тәжірибесі, жұмыс туралы түсінігі және білімі жоқ адамдарға арналған. Барлық ақпаратты орталықтандыру. Қолдануға ыңғайлы және де көзге жағымды веб қосымша әзірлеу.

1.2 Терминдер мен қысқартулар

Дипломдық жобаға түсіндірме жазбада қолданылатын терминдер мен аббревиатуралар төмендегі кестеде келтірілген.

1-кесте – Терминдер мен аббревиатуралар, сонымен қатар олардың анықтамалары

Терминдер немесе қысқартулар	Анықтамалар
SQL	Құрылымдық сұраныс тілі
ДҚ	Деректер қоры
АЖ	Ақпараттық жүйе
JSON	JavaScript нысан белгісі
ER	(ағылшынша Entity Relationship сөзінен қысқартылған) тұлға-қатынас
Фреймворк	бағдарламалық жүйенің құрылымын анықтайтын бағдарламалық платформа
Кітапхана	Бағдарламалауда басқа программаларды әзірлеу үшін біріктірілген арнайы программалар мен объектілер.
API	(ағылшын тілінен қысқартылған. Application Programming Interface) қолданбалы бағдарламалау интерфейсі. Әртүрлі бағдарламалардың бір бірімен байланысың қамтитын құралдар жиынтығы

1 – кестенің жалғасы

Терминдер немесе қысқартулар	Анықтамалар
PyChram	Бағдарламалық қамтамасыз етуді әзірлеу ортасы
Django	Python тілінде жазылған веб қолданбалы құрылым
CRUD	Деректер қорымен жұмыс жасау, құру, оқу, өзгерту, жою кезіндегі функциялар

1.3 Жобаның өзектілігі

Қазіргі кезде барлық жұмыс орталықтарына белгілі бір тәжірибелік қабілеттер керек, сол қабілеттерді алу үшін дайындық орталықтары қолданылады. Дайындық орталықтары қарқынды дамуда мысалыға айтқанда Қазақстандағы ірі компаниялар Kaspi lab және Kolesa academia сияқты адамдарды даярлауға жана білім беруге және де сәтті өткен жағдайда оларға жұмыс ұсынады, осы мысал сияқты көптеген компаниялар өз өзіне жұмыскерлерді белгілі бір немесе өзінің дайындық орталығы арқылы оқытып өзіне керек мамандарды дайындайды.

Кәсіпорындардың қызметкерлердің білімін дамыту немесе қайта дайындау, сондай-ақ жаңа қызметкерлерді және осы бейіндегі білім беру қызметтерін қажет ететін басқа адамдарды оқыту мақсатында құрылған мамандандырылған оқыту жүйесі болып келеді.

Бұл жобаны жүзеге асыру барысындағы білім және жұмыс беру жүйелері үшін өте пайдалы. Өйткені, бүгінгі таңда адамдардың білім алып, білімін дамытуы өте маңызды.

1.4 Веб қосымша жасаудың негіздері

Веб-бағдарлама – бұл клиент браузері арқылы веб-сервермен әрекеттесетін клиент – сервер қолданбасы.

Веб-қосымшаның жұмыс істеу принципі сервер мен клиент арасында таратылады, деректер негізінен серверде сақталады, ақпарат желі арқылы алмасады. Бұл тәсілдің бір артықшылығы клиенттердің пайдаланушының арнайы операциялық жүйесіне тәуелсіз болуы, сондықтан веб-қосымшалар платформааралық қызметтер болып табылады.

Веб-қосымшаны құруға қойылатын негізгі талаптар:

- веб-қосымшаның мақсаты;

- техникалық шарттарды әзірлеу;
- платформаны таңдау;
- веб-қосымшаларды әзірлеу;
- веб-қосымшаларды орналастыру;
- клиентті веб-серверге қосу.

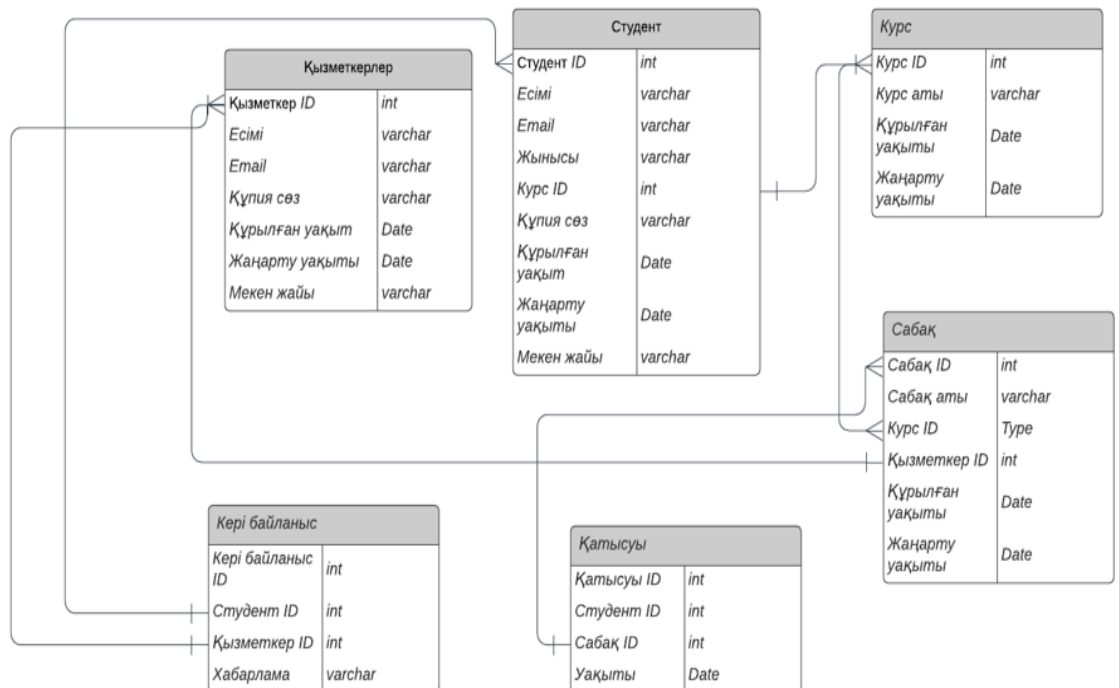
1.5 Жүйені модельдеу

1.5.1 ER диаграммасы

UML диаграммасының бұл түрі негізінен дерекқорды жобалауда қолданылады. Мәліметтер қоры әзірленіп жатқан жүйенің міндетті бөлігі болғандықтан, оның құрылымы мен логикасын осы диаграмма арқылы сипаттау қажет. Ол негізгі объектілерді, сондай-ақ олардың арасындағы қатынастарды көрсетеді.

Бұл диаграмма анықтауға көмектеседі:

- мәліметтер қорындағы негізгі субъектілер, олардың арасындағы байланыс;
- тұтас логиканы тұтастай алдын ала бағалауға мүмкіндік береді;
- мәліметтер қорын реляциялық түрлендіруге көмектеседі;
- дерекқорда қандай деректерді сақтау керектігі туралы жақсы түсінік алуға көмектеседі



1.1-сурет – Басты кестелердің ER диаграммасы

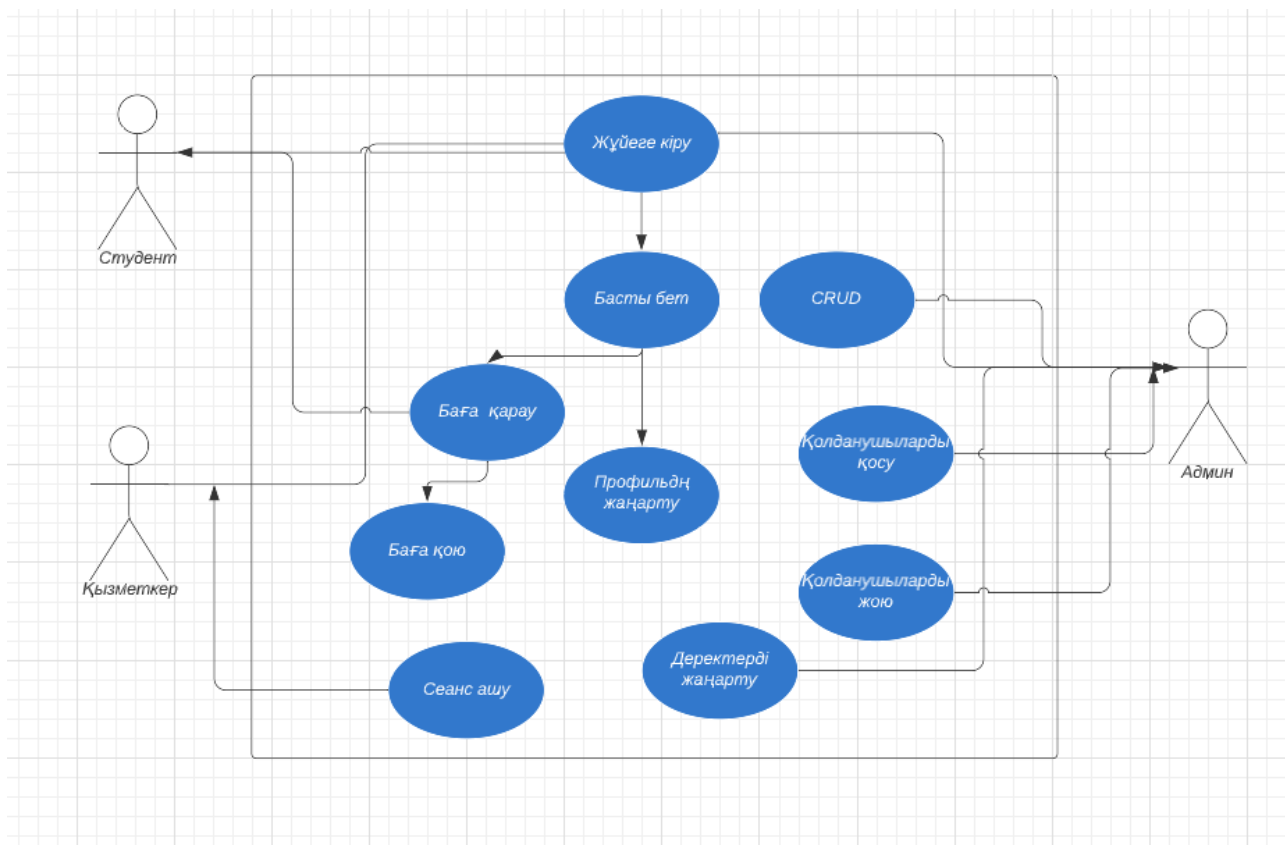
1.5.2 Use Case диаграммасы

Use Case диаграммасы - Flexberry Designer қолдайтын қолжетімді диаграмма түрлерінің бірі.

Диаграмманың негізгі элементтері қатысушы (актер) және прецедент (опция) болып табылады.

Қатысушы — жұмыс жағдайларымен немесе объектілермен өзара әрекеттесу кезінде орындалатын байланысты рөлдердің жиынтығы. Бұл қатысушыны немесе объектіден тыс нәрсені білдіретін басқа жүйе, ішкі жүйе және т.б. Графикалық түрде қатысушы «кішкентай адам» ретінде бейнеленген.

Прецедент (пайдалану жағдайы) – қатысушы бақылайтын нәтижеге жеткізетін жүйе орындайтын оқиғалардың сипаттамасы. Пайдалану жағдайында қатысушылар мен жүйенің бір бірімен байланысын көрсетеді. Прецедент белгілі нақты орындалатынын әрекеттерді көрсетеді. Қолдану жағдайлары - шеңбер түрінде көрсетілген, оның ішінде оның атауы көрсетілген.



1.2-сурет – Жүйенің Use case диаграммасы

2 Технологиялар бөлімі

2.1 SQLite деректер базасы

SQLite – пайдалану оңай, ықшам, бірақ өнімді және жаңартылған ДҚБЖ, жүйе бұл өнімді негізгі ДҚБЖ ретінде таңдады. Бұл әлемдегі ең танымал ДҚБЖ. Бастапқы коды ашық. SQLite реляциялық ДҚБЖ болып табылады және өзіндік мүмкіндіктері мен қосымша мүмкіндіктері бар SQL сұрау тілін пайдаланады. Қазіргі уақытта SQL-92 тіл стандартына қолдау көрсетіледі. Бұл ДҚБЖ пайдасына таңдау жасалды, өйткені оны орналастыру оңай, ол жұмыс кезінде бөлек процесті қажет етпейді, дерекқорды кез келген жолмен конфигурациялау немесе басқарудың қажеті жоқ, ол жеңіл және жүйені жүктемейді. SQLite кітапханасының функциялық шақырулары (API) пайдаланылады. Бұл тәсіл қосымша шығындарды, жауап беру уақытын қысқартады және бағдарламаны жеңілдетеді. SQLite бүкіл дерекқорды (соның ішінде анықтамаларды, кестелерді, индекстерді және деректерді) бағдарлама жұмыс істеген құрылғыдағы бір стандартты файлда сақтайды. Жүзеге асырудың қарапайымдылығы жазу транзакциясын орындаудың басталуына дейін мәліметтер базасын сақтайтын бүкіл файлдың блокталуына байланысты қол жеткізіледі; ACID [1] функцияларына басқа нәрселермен қатар журнал файлы жасау арқылы қол жеткізіледі.

2.2 Бағдарламалау тілі

2.2.1 Python

Кодты жазу үшін пайдаланылған бағдарламалау тілі Python 3.10 нұсқасы. Бұл бағдарламалау тілі өзінің сұранысы мен танымалдығының шыңында, ол көптеген тапсырмаларда жиі қолданылады. Оның әзірлеушілер арасында көптеген жанкүйерлері, кез келген қажеттіліктерге арналған көптеген кітапханалар мен фреймворктар бар. Сондай-ақ, жобаның кодын жазу үшін көптеген кітапханалар пайдаланылды.

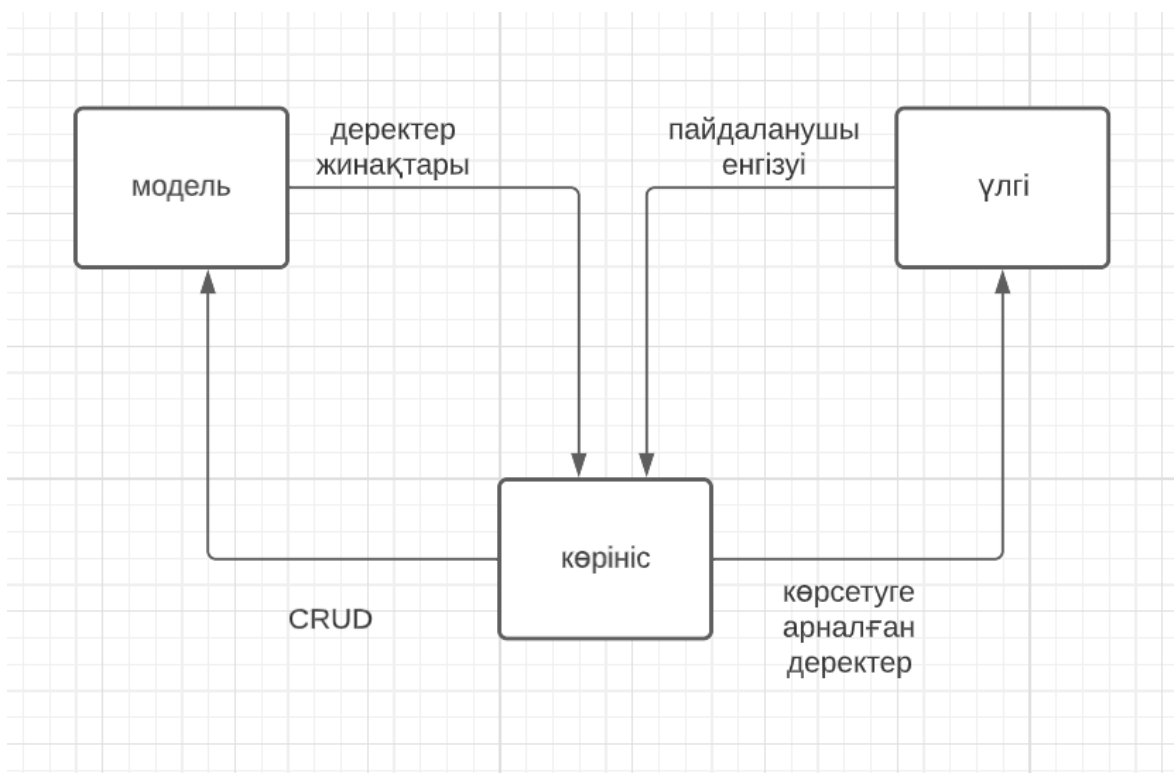
2.2.2 Django фреймворкі

Django Python тілінде жазылған ең жақсы веб фреймворк болып саналады. Django — Python тілінде жазылған ақысыз және бастапқы коды ашық веб-бағдарлама құрылымы. Фреймворк - бұл веб-сайттарды жылдам және оңай жасауға көмектесетін компоненттер жиынтығы.

Веб-сайтты жасаған сайын сізге ұқсас компоненттер қажет: пайдаланушыларды аутентификациялау тәсілі (логин, шығу, тіркеу), сайттың басқару тақтасы, пішіндер, файлдарды жүктеп салу құралдары және т.б.

Бақытымызға орай, басқа адамдар веб-әзірлеуде осындай проблемаларды байқады, сондықтан олар бірігіп, бізге пайдалану үшін дайын үлгілерді ұсынатын фреймворктарды (Django және басқалар) жасады.

Даму процесін жеңілдету және доңғалақты қайта ойлап таппауға мүмкіндік беретін жақтаулар бар.



2.1-сурет – Джанго фреймворкінің жұмыс істеу принципі(MVT)

2.3 HTML бағдарламалау тілі

HTML гипермәтін бетін белгілеу тілі болып табылады. Ол браузерге жүктелген сайтты қалай көрсету керектігін айту үшін қолданылады.

Тіл тегтерден тұрады – бұл пайдаланушы браузерінде көрнекі нысандарға түрлендірілетін ерекше командалар. Мысалы, тегі кескіндерді бетке орналастыру үшін қолданылады. Оның файлға сілтемені көрсететін талап етілетін src төлсипаты бар.

2.4 CSS кестелер стилі

CSS - каскадты стиль кестелері. Шын мәнінде, бұл HTML құжатының сыртқы түрін сипаттауға жауапты тіл. Қазіргі веб-сайттардың басым көпшілігі HTML + CSS тіркесіміне негізделген

CSS әртүрлі құрылғылар мен экран өлшемдері үшін дизайн, орналасу және орналасу нұсқаларын қоса алғанда, құжат дизайнына арналған стильдерді (ережелерді) анықтау. Бұл пішімдеу әдісінің бірнеше артықшылығы бар:

- тегтер қайталанбайды;
- құжатты сақтау оңайырақ;
- әр беттің пішімдеуін реттеудің орнына бүкіл сайттың сыртқы түрін орталықтан өзгертуге болады.

2.5 JQuery, Javascript

jQuery - JavaScript және HTML арасындағы өзара әрекеттесуге бағытталған JavaScript мүмкіндіктері жинағы. jQuery кітапханасы кез келген DOM элементіне қол жеткізуді, DOM элементтерінің атрибуттары мен мазмұнына қол жеткізуді және оларды өңдеуді жеңілдетеді. JQuery кітапханасы сонымен қатар AJAX-пен жұмыс істеу үшін ыңғайлы API ұсынады.

CSS көрсетуді HTML құрылымынан бөлетіні сияқты, jQuery де мінез-құлықты HTML құрылымынан бөледі. Мысалы, түймені басу оқиғасының өңдегішіне тікелей нұсқаудың орнына, басқару түймелерді анықтайтын, содан кейін оны басу оқиғасының өңдегішіне түрлендіретін JQuery-ге беріледі. Мінез-құлық пен құрылымның бұл бөлінуі JavaScript-тің назар аудармайтын принципі деп те аталады.

jQuery кітапханасында тапсырмалардың ең кең ауқымы үшін пайдалы функциялар бар. Дегенмен, кітапхана әзірлеушілерінің мақсаты jQuery-де барлық жерде сәйкес келетін функцияларды біріктіру емес еді, өйткені бұл көптеген кодтарға әкеледі, олардың көпшілігі қажет емес. Сондықтан кітапхана мен плагиндердің ықшам әмбебап ядросының архитектурасы жүзеге асырылды [6]. Бұл ресурс үшін сұранысқа ие болатын JavaScript функционалдығын дәл жинауға мүмкіндік береді.

2.6 Код редакторы

PyCharm — Python бағдарламалау тіліне арналған біріктірілген әзірлеу ортасы. Кодты талдау құралдарын, графикалық отладчикті, бірлік сынақ жүгірткісін қамтамасыз етеді және Django веб-әзірлеуіне қолдау көрсетеді. PyCharm IntelliJ IDEA негізінде JetBrains әзірлеген.

PyCharm — Windows сияқты барлық операциялық жүйелерде қолданылады. PyCharm код редакторының екі түрі бар 1 шісі бұл PyCharm

Community Edition (тегін), ал 2 шісі PyCharm Professional шығарылымы (ақылы) меншікті болып табылады, Apache компаниясымен лицензияланған.

Мүмкіндіктері:

- PyDev көмегімен кодты жөндеу;
- Кодты рефакторинг;
- Git, SVN, Mercurial және басқа нұсқаларды басқару жүйелерін қолдау;
- Кодты автоматты аяқтау.

Пайдаланушылар өздерінің плагиндерін жаза алады, осылайша PyCharm мүмкіндіктерін кеңейтеді. Басқа JetBrains IDE құрылғыларының кейбір плагиндері PyCharm-мен жұмыс істей алады. PyCharm-пен үйлесімді мыңнан астам плагиндер бар.

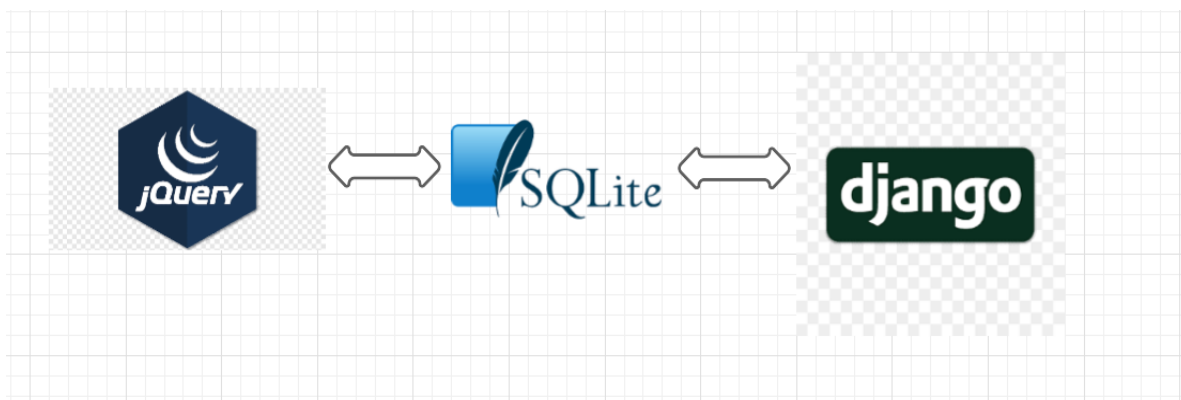
3 Жобаның құрылымы

3.1 «Дайындық орталығы» жобасы туралы

Бұл жоба дайындық орталығының дәстүрлі түрде оқуды жобалау және де оңайлату болып табылады. Веб қосымша дайындық орталығынның студенттер мен қызметкерлердің байланысың көрсетеді. Компанияның ішкі бизнес-процестерін автоматтандыруға арналған веб-қосымша. Барлық деректер орталықтандырылған және өте ыңғайлы. Бұл жүйеде үш қолданушылар бар: “Администратор”, “Қызметкер”, “Студент”.

Admin	Қызметкер	Студент
Жүйеге кіру	Жүйеге кіру	Жүйеге кіру
Студенттер мен Қызметкерлерге кері байланыс беру	Студентке хабарлама жіберу	Қатысуды көру
Студенттерді басқару	Кері байланыс	Студент профилі
Қызметкерлерді басқару	Демалысқа өтініш беру	Демалысқа өтініш беру
хабарлама жіберу	Студенттің сабаққа қатысуы	Кері байланыс
қатысуды көру	Қызметкер профилі	Хабарландыруды қараңыз
Курстар мен сабақтарды басқару	Жүйедең шығу	Жүйедең шығу
Жүйедең шығу		

3.1-сурет – Қолданушылар түрлері және қызметтері



3.2-сурет – Бағдарламалық тілдердің бір бірімен байланысы

3.1.1 «Дайындық орталығы» веб қосымшасының функционалын сипаттау

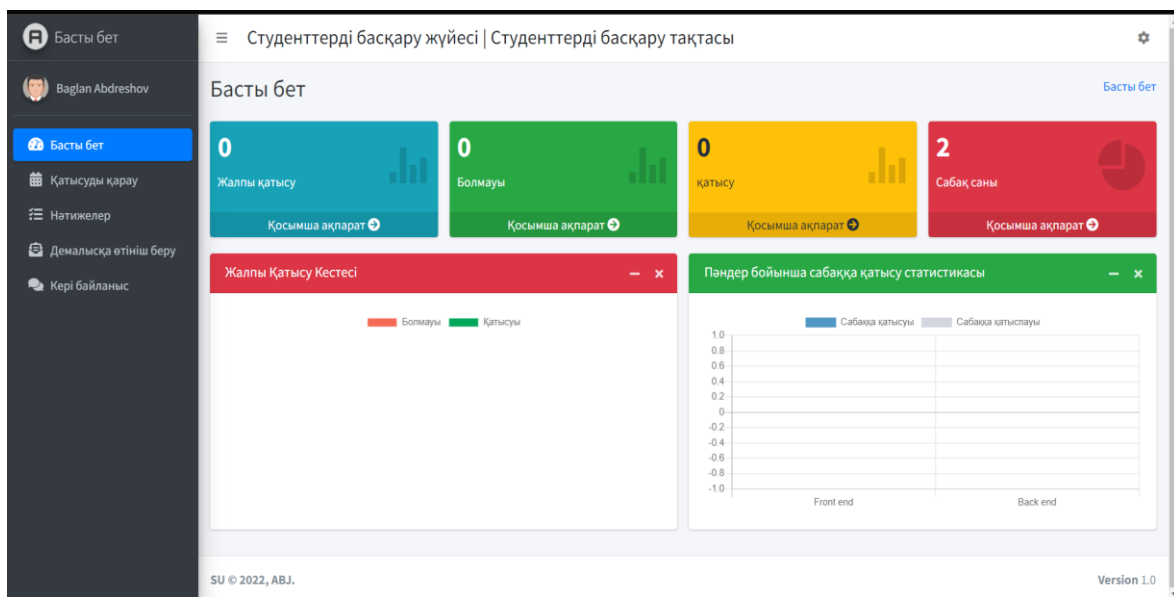
Веб қосымшаға кіру беті 2 бөлімнен тұрады ол біздің пайдаланушы аты және құпия сөз (3.3-сурет).

3.3-сурет – Авторизация беті

3.1.2 Студенттің бет көріністері

Студент атынан кірген кезде бізде басты бет осылай көрінеді. Басты бетте

Жалпы қатысуы, болмауы , қатысуы және де сабақ саны көрсетіледі. Сол жағынаң ыңғайлы беттер арасына ауысуға арналған интерфейс орналастырылған (3.4-сурет).



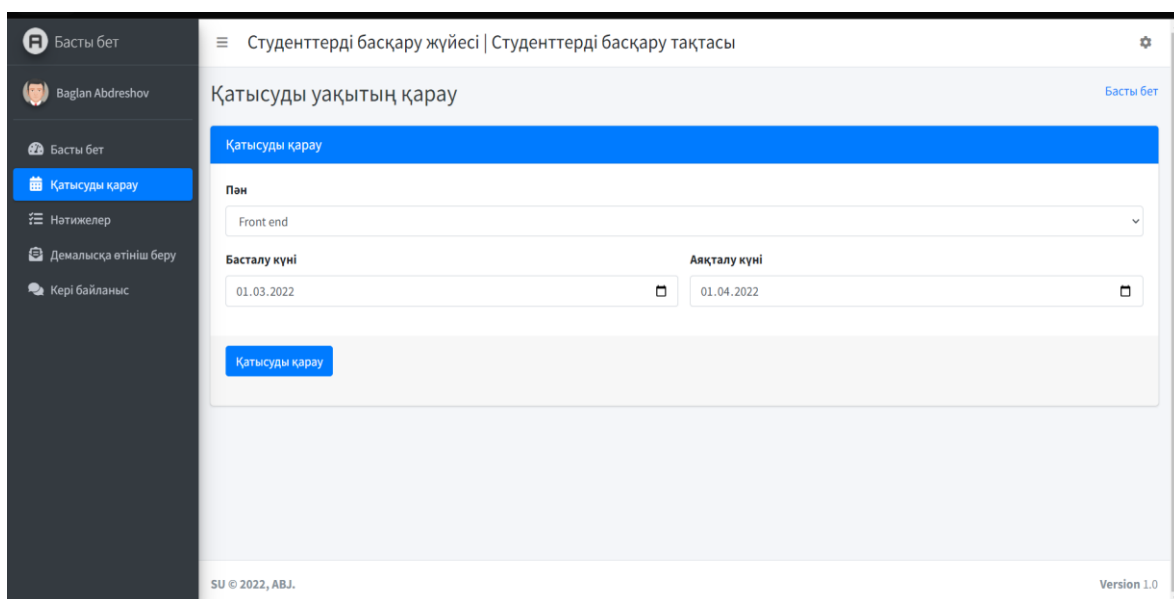
3.4-сурет – Студенттің басты бет көрінісі

Бұл біз өз профилімізге басқан кезде пайда болатың бет. Осы бетте біз өзіміздің профильді жанарта аламыз егерде бір нәрсені өзгерту қажет болса. Ол бізде пайдаланушы аты, пошта, есімі, тегі, мекен жайы, және құпия сөзден тұрады (3.5-сурет).

3.5-сурет – Студенттің профиль жаңарту бет көрінісі

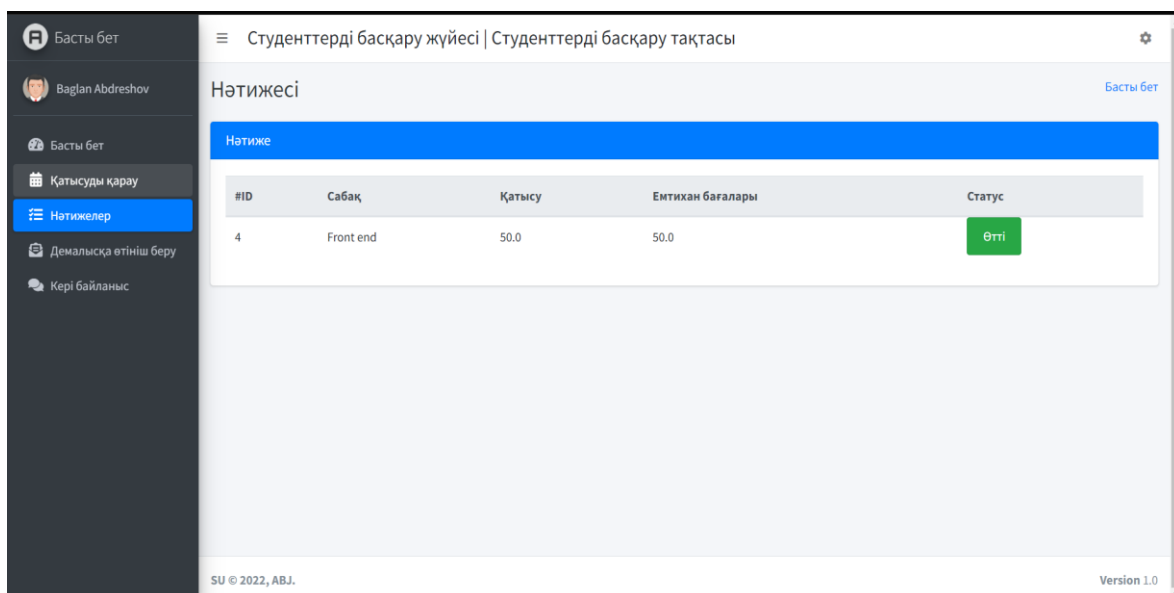
Қатысуды қарау бөлімінде біз пәнді таңдап сол пәннің қай уақыт аралығын

таңдап қатысуды қарай аламыз. Қатысу уақытың қарау беті бізде пән, басталу күні және аяқталу күнінен тұрады (3.6-сурет).



3.6-сурет – Студенттің қатысу бет көрінісі

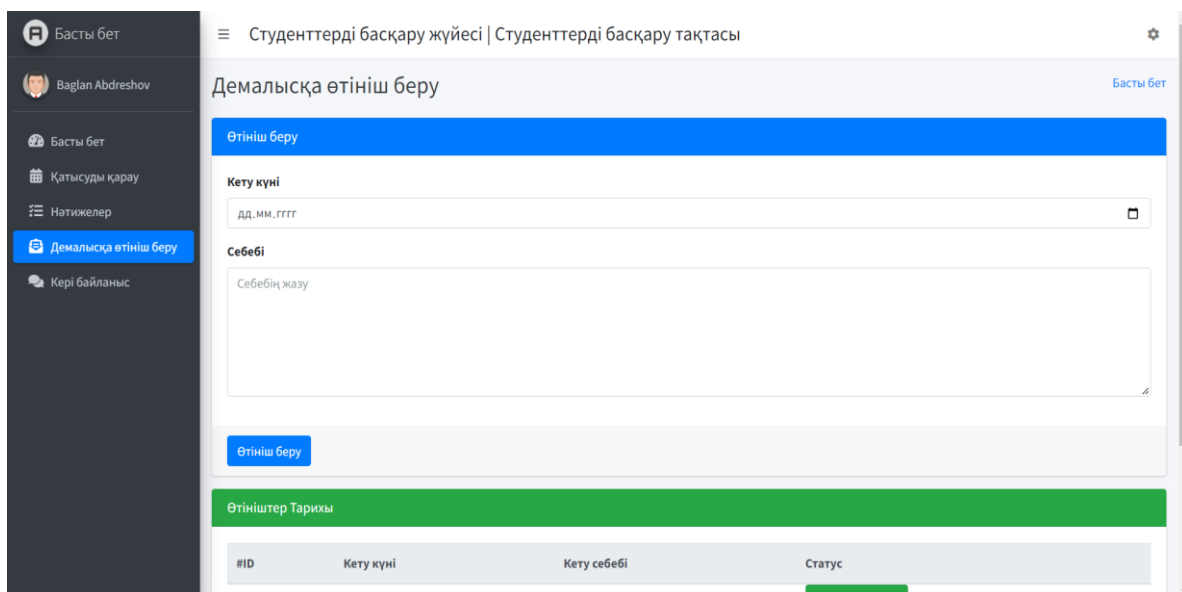
Нәтижелер бетінде біз Сабақтар бойынша қатысу және де емтихан бойынша ақпараттарды қарап біз осы сабақтаң өттік немесе өтпедік қарай аламыз (3.7-сурет).



3.7-сурет – Студенттің нәтижелер бет көрінісі

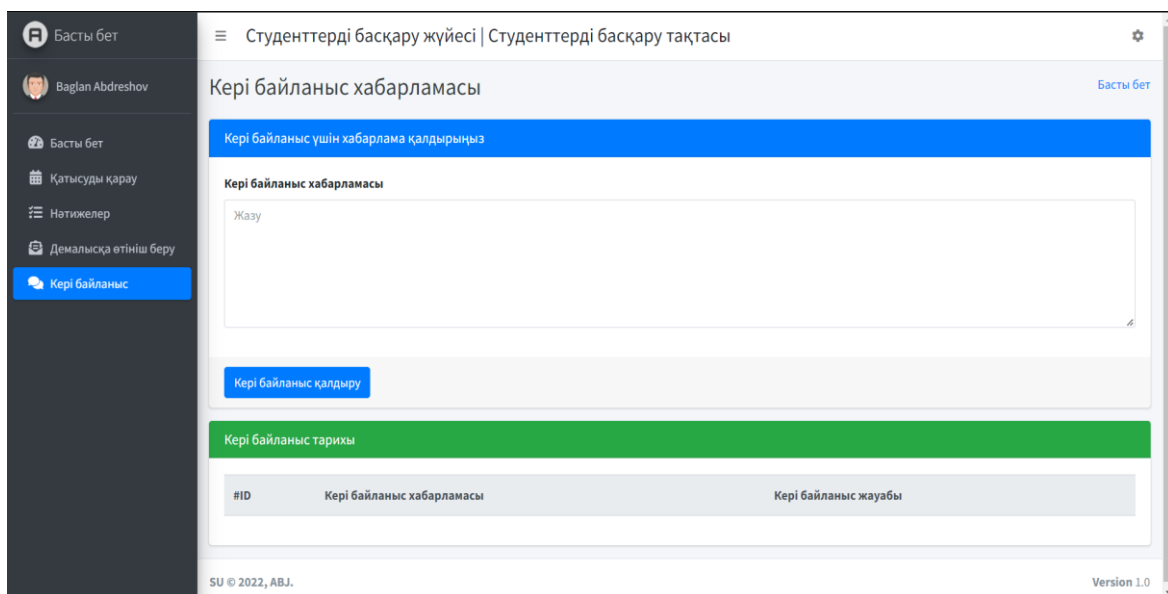
Демалысқа өтініш беру бетінде егер де студентке белгілі бір себептермен сабаққа қатыса алмау жағдайында өтініш бере аламыз ол өтінішті қызметкер немесе администратор қарай алады. Өтініш беру беті бізде кету күні және

себебінен тұрады осы 2 бөлімді толтырып өтініш жібере аламыз (3.8-сурет).



3.8-сурет – Студенттің демалысқа өтініш беру бет көрінісі

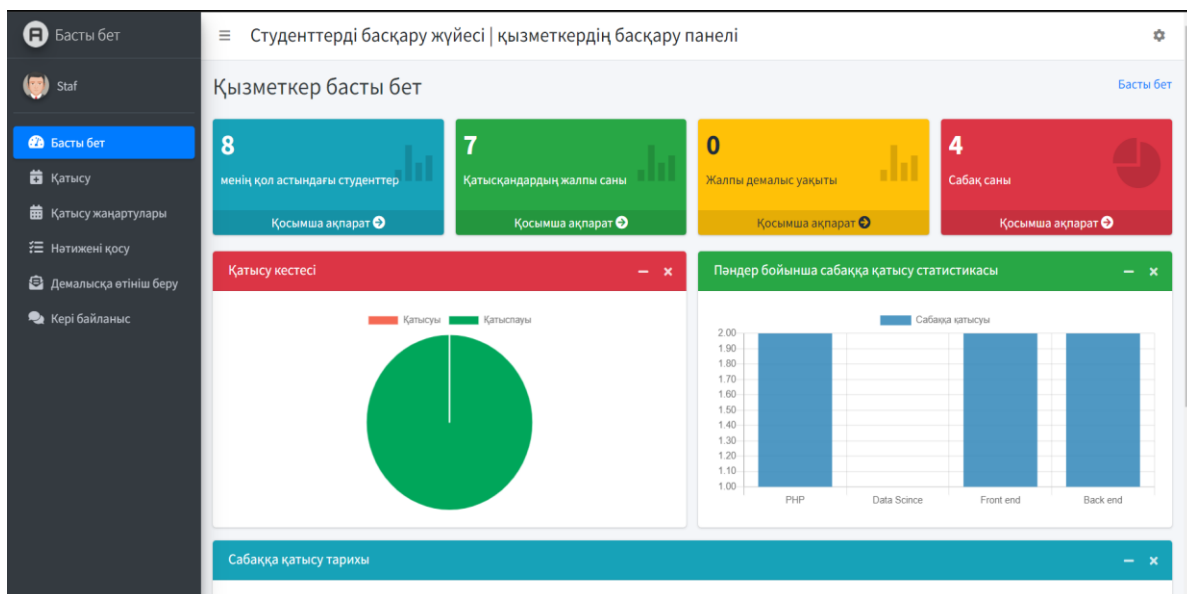
Кері байланыс бетінде біз администраторға бізде белгілі бір сұрақ немесе қиындық туындап жатса қалдыра аламыз. Кері байланыс беті бізде бір толтыру бөлімінен тұрады, сол толтыру бөліміне бізге керекті сұрақты жазамыз (3.9-сурет).



3.9-сурет – Студенттің кері байланыс бет көрінісі

3.1.3 Қызметкердің бет көріністері

Қызметкер атынан кірген кезде бізде басты бет осылай көрінеді. Басты бетте бізде студенттер санын, қатысу жалпы саны, демалыс уақыты және қанша сабақ санына жауап береді соны көре аламыз. Сол жағынаң ыңғайлы беттер арасына ауысуға арналған интерфейс орналастырылған. (3.10-сурет).



3.10-сурет – Қызметкердің басты бет көрінісі

Бұл біз өз профилімізге басқан кезде пайда болатың бет. Осы бетте біз өзіміздің профиліңді жанарта аламыз егерде бір нәрсені өзгерту қажет болса. Ол бізде пайдаланушы аты, пошта, есімі, тегі, мекен жайы, және құпия сөзден тұрады (3.11-сурет).

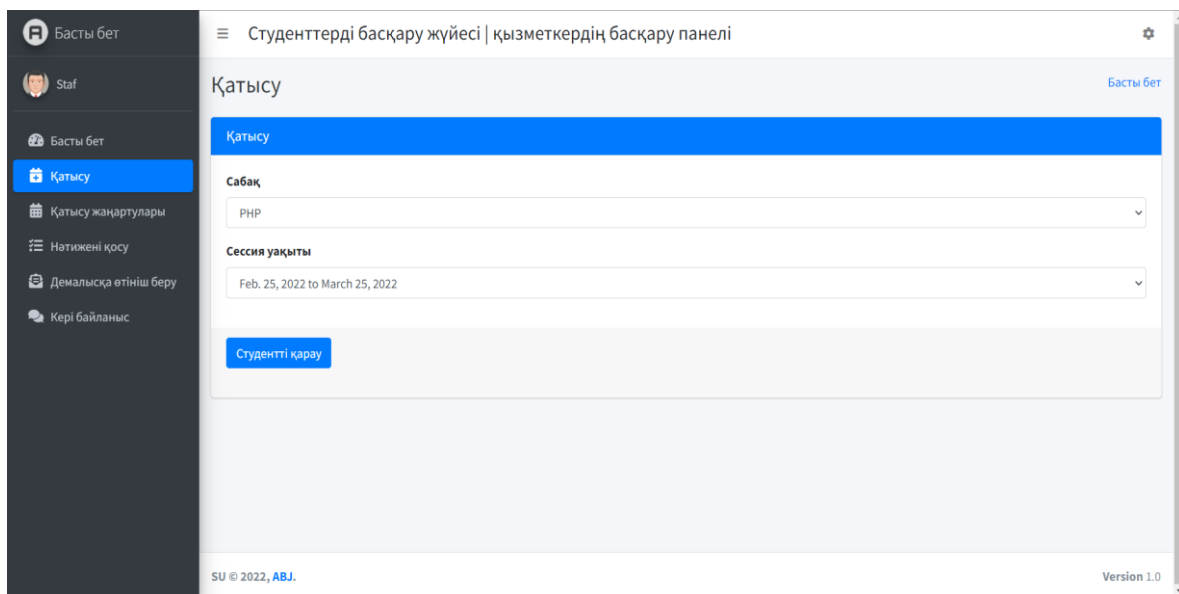
The screenshot shows a form titled 'Профильді жаңарту'. The form contains the following fields:

- Пайдаланушы аты**: Input field with 'staf@gmail.com' entered.
- Email**: Input field with 'staf@gmail.com' entered.
- Есімі**: Input field with 'Staf' entered.
- Тегі**: Empty input field.
- Мекен жайы**: Empty text area.
- Құпия сөз**: Empty input field.

At the bottom of the form, there is a blue button labeled 'Профильді жаңарту'.

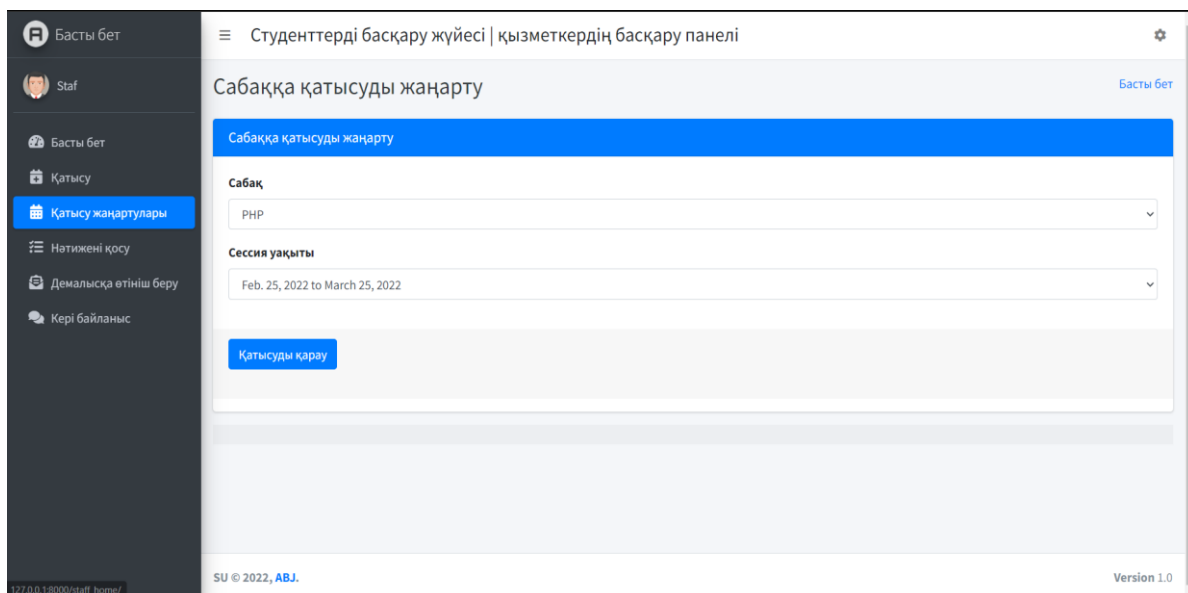
3.11-сурет – Қызметкердің профиль жаңарту бет көрінісі

Қатысуды қарау бөлімінде біз сабақты таңдап сол сабаққа қай студент қатысты қарап соңы сақтаймыз. Қатысу беті 2 бөлімнен тұрады ол керекті сабақ және сол сабақтың сессия уақыты. Студентті қарау батырмасын басқан соң студенттер тізімі шығады (3.12-сурет).



3.12-сурет – Қызметкердің қатысу бет көрінісі

Қатысуды жаңарту бөлімінде біз сабақты таңдап сол сабаққа қай студент қатысты соңы жаңарта аламыз. Сабаққа қатысуды жаңарту беті бізде сабақтаң және сессия уақытынан тұрады (3.13-сурет).



3.13-сурет – Қызметкердің қатысуды жаңарту бет көрінісі

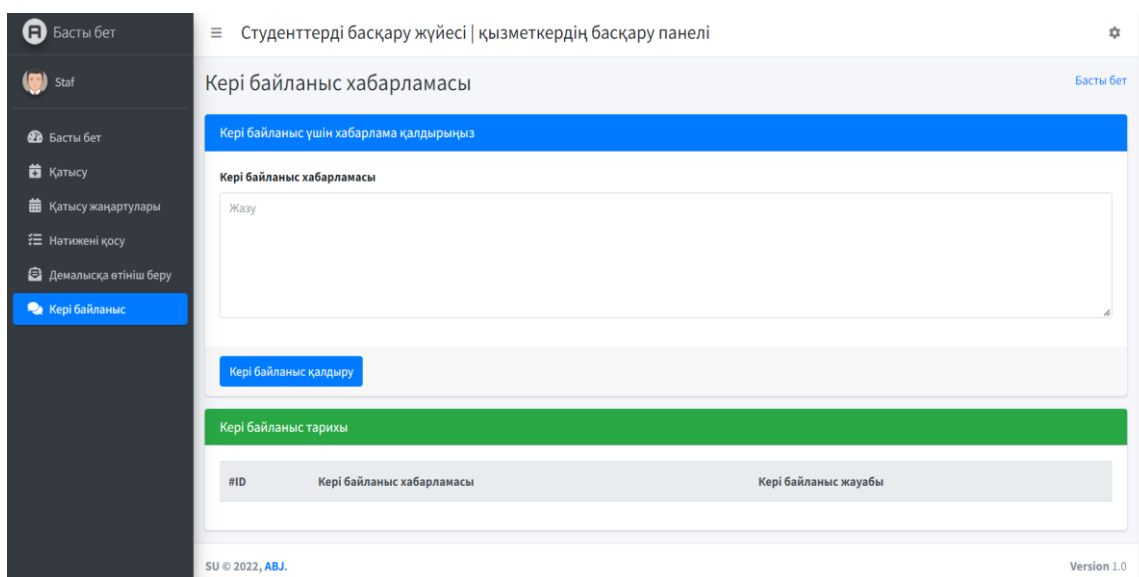
Нәтиже қосу бетінде біз қызметкер жауапты сабақтарға студенттерге баға қоя аламыз. Нәтиже қосу беті сабақ, сессия уақытынан тұрады. Студентті табу батырмасын басқан соң бізде студенттер тізімі, тапсырма белгілері және емтихан бағалары пайда болады. Осы ұяшықтарды толтырып нәтижені сақтаймыз (3.14-сурет).

3.14-сурет – Қызметкердің нәтижелерді қосу бет көрінісі

Демалысқа өтініш беру бетінде егер де студентке белгілі бір себептермен сабаққа қатыса алмау жағдайында өтініш бере аламыз ол өтінішті администратор қарай алады. Өтініш беру беті бізде кету күні және себебінен тұрады осы 2 бөлімді толтырып өтініш жібере аламыз (3.15-сурет).

3.15-сурет – Қызметкердің демалысқа өтініш беру бет көрінісі

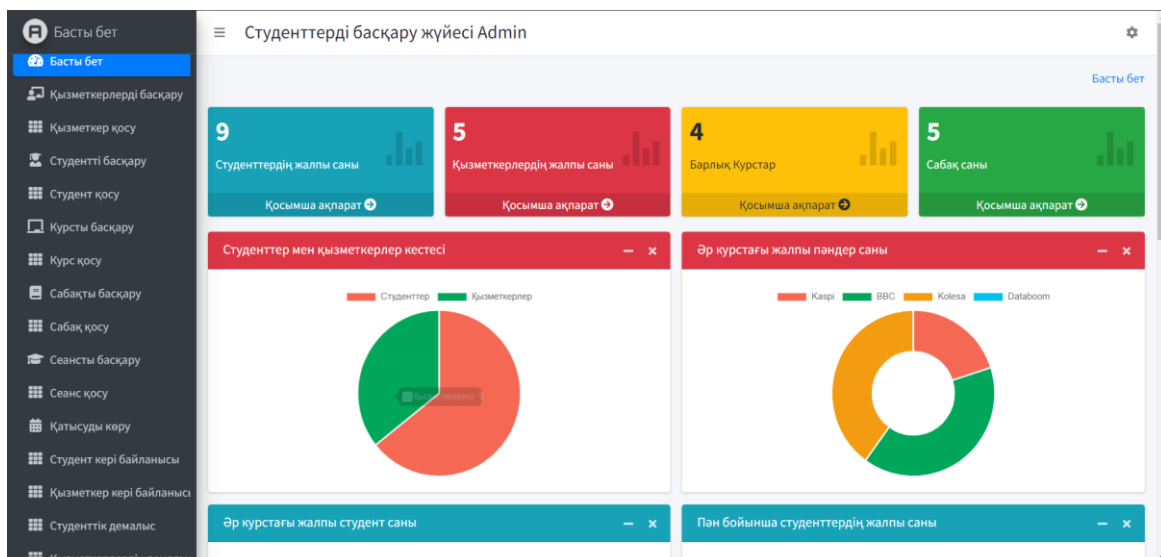
Кері байланыс бетінде біз администраторға бізде белгілі бір сұрақ немесе қиындық туындап жатса қалдыра аламыз. Кері байланыс беті бізде бір толтыру бөлімінен тұрады, сол толтыру бөліміне бізге керекті сұрақты жазамыз (3.16-сурет).



3.16-сурет – Қызметкердің кері байланыс бет көрінісі

3.1.4 Админ бет көрінісі

Астынғы суретте біз администратордың интерфейсің көре аламыз. Администраторда барлық құқықтар бар. Қызметкерлер , Студенттерді басқару (қосу , өшіру , өзгерту) Курстарды басқару (қосу , өшіру , өзгерту), Сабақтарды басқару (қосу , өшіру , өзгерту). Кері байланыстарға, демалыстарға жауап беру, қатысуды қарау және де сеанстарды басқару болып келеді. Басты бетінде біз жалпы дайындық орталығында қанша студент , қызметкер , курс және сабақ сандарың көре аламыз және де солар график түрінде көрсетілен. (3.17-сурет).



3.17-сурет – Администратордың бет көрінісі

Қызметкерлерді басқару бетінде жүйедегі барлық қызметкерлер тізімі тұрады. Сол тізім ішінде қызметкер жайлы ақпараттар көрсетілген олар:тегі, пайдаланушы аты, email, мекен жайы, соңғы кіру, қосылған күні және де әрекет батырмасынан тұрады. Әрекет батырмасында бізде өзгерту немесе жою функциялары қолданылған (3.18-сурет).

Tегі	Пайдаланушы аты	Email	Мекен жай	Соңғы кіру	Қосылған күні	Әрекет	
Burch	kunifi	cihezymo@mailinator.net	Nisi reiciendis pari	None	June 26, 2020, 4:01 p.m.	Өзгерту Жою	
is	Strickland	gimamomuh	sikoka@mailinator.com	Ea eos ipsam dolor	July 9, 2020, 4:45 p.m.	June 26, 2020, 4:02 p.m.	Өзгерту Жою
ег	Carter	cuqenana	rynex@mailinator.com	Ut quos autem aliqui	None	July 7, 2020, 3:15 p.m.	Өзгерту Жою
лн	Strong	misola	caracax@mailinator.net	Voluptatem consecte	None	July 7, 2020, 3:15 p.m.	Өзгерту Жою
		staf@gmail.com	staf@gmail.com	April 29, 2022, 6:59 a.m.	March 16, 2022, 11:31 a.m.	Өзгерту Жою	
nder	Hardy	lutozu	zicyj@mailinator.com	Aut dolor ad laborio	None	April 5, 2022, 11:33 a.m.	Өзгерту Жою
er	Reeves	byropymiz	qabib@mailinator.com	Eos ut ut deleniti	None	April 5, 2022, 11:33 a.m.	Өзгерту Жою

3.18-сурет Қызметкерді басқару бет көрінісі

Студенттерді басқару бетінде жүйедегі барлық қызметкерлер тізімі тұрады. Сол тізім ішінде қызметкер жайлы ақпараттар көрсетілген олар:есімі, тегі, username, email, мекен жайы, жынысы, профиль суреті, басталу уақыты, аяқталу уақыты, курс, соңғы кіру, қосылған күні және де әрекет батырмасынан тұрады. Әрекет батырмасында бізде өзгерту немесе жою функциялары қолданылған (3.19-сурет).

Студентті басқару

+ Студентті қосу

Студенттік мәліметтер

Жыныс	Профиль суреті	Басталу уақыты	Аяқталған уақыт	Курс	Соңғы кіру	Қосылған күні	Әрекет
Male	[Profile Picture]	Jan. 25, 2022	March 25, 2022	BBC	July 9, 2020, 4:45 p.m.	June 26, 2020, 4:07 p.m.	Өзгерту Жою
Female	[Profile Picture]	March 1, 2022	April 1, 2022	Databoom	None	June 26, 2020, 4:08 p.m.	Өзгерту Жою
Male	[Profile Picture]	Jan. 25, 2022	March 25, 2022	BBC	July 3, 2020, 7:37 a.m.	July 1, 2020, 9:01 a.m.	Өзгерту Жою
Male	[Profile Picture]	Feb. 25, 2022	March 25, 2022	Kaspi	None	July 5, 2020, 11:55 a.m.	Өзгерту Жою
Female	[Profile Picture]	Jan. 25, 2022	March 25, 2022	Kolesa	None	July 5, 2020, 11:55 a.m.	Өзгерту Жою
Female	[Profile Picture]	Jan. 25, 2022	March 25, 2022	BBC	None	July 5, 2020, 11:55 a.m.	Өзгерту Жою
Female	[Profile Picture]	Jan. 25, 2022	March 25, 2022	BBC	None	July 5, 2020, 11:55 a.m.	Өзгерту Жою

3.19-сурет Студентті басқару бет көрінісі

Курсты басқару бетінде жүйедегі барлық курстар тізімі тұрады. Сол тізім ішінде курстар жайлы ақпараттар көрсетілген олар: id, курс аты, құрылған уақыты, жаңартылған уақыты және де әрекет батырмасынан тұрады. Әрекет батырмасында бізде өзгерту немесе жою функциялары қолданылған (3.20-сурет).

Студенттерді басқару жүйесі Admin

Курсты басқару

+ Курс қосу

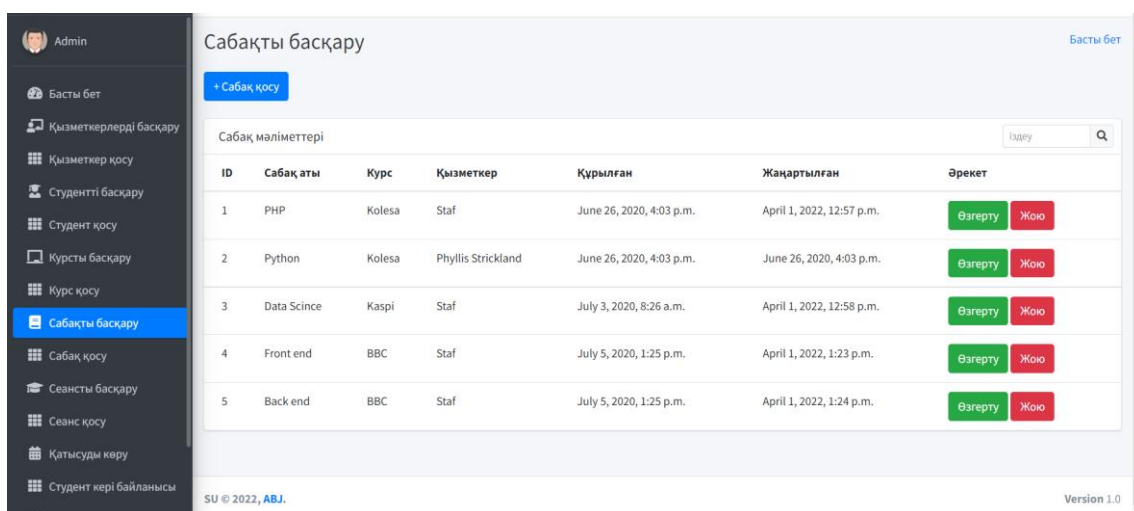
Курс мәліметтері

ID	Курс аты	Құрылған	Жаңартылған	Әрекет
1	Kaspi	June 26, 2020, 4:02 p.m.	April 1, 2022, 12:54 p.m.	Өзгерту Жою
2	BBC	June 26, 2020, 4:02 p.m.	April 1, 2022, 12:54 p.m.	Өзгерту Жою
3	Kolesa	June 26, 2020, 4:02 p.m.	April 1, 2022, 12:54 p.m.	Өзгерту Жою
4	Databoom	June 26, 2020, 4:02 p.m.	April 1, 2022, 12:55 p.m.	Өзгерту Жою

SU © 2022, ABJ. Version 1.0

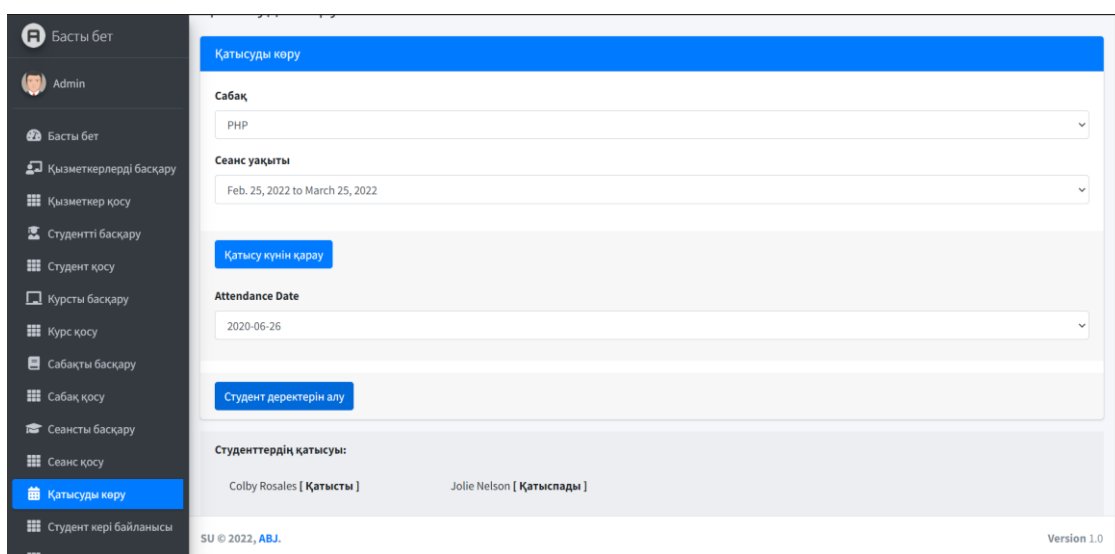
3.20-сурет Курсты басқару бет көрінісі

Сабақты басқару бетінде жүйедегі барлық сабақтар тізімі тұрады. Сол тізім ішінде сабақтар жайлы ақпараттар көрсетілген олар: id, сабақ аты, курс аты, осы сабақты беретін қызметкер, құрылған уақыты, жаңартылған уақыты және де әрекет батырмасынан тұрады. Әрекет батырмасында бізде өзгерту немесе жою функциялары қолданылған (3.21-сурет).



3.21-сурет Сабақты басқару бет көрінісі

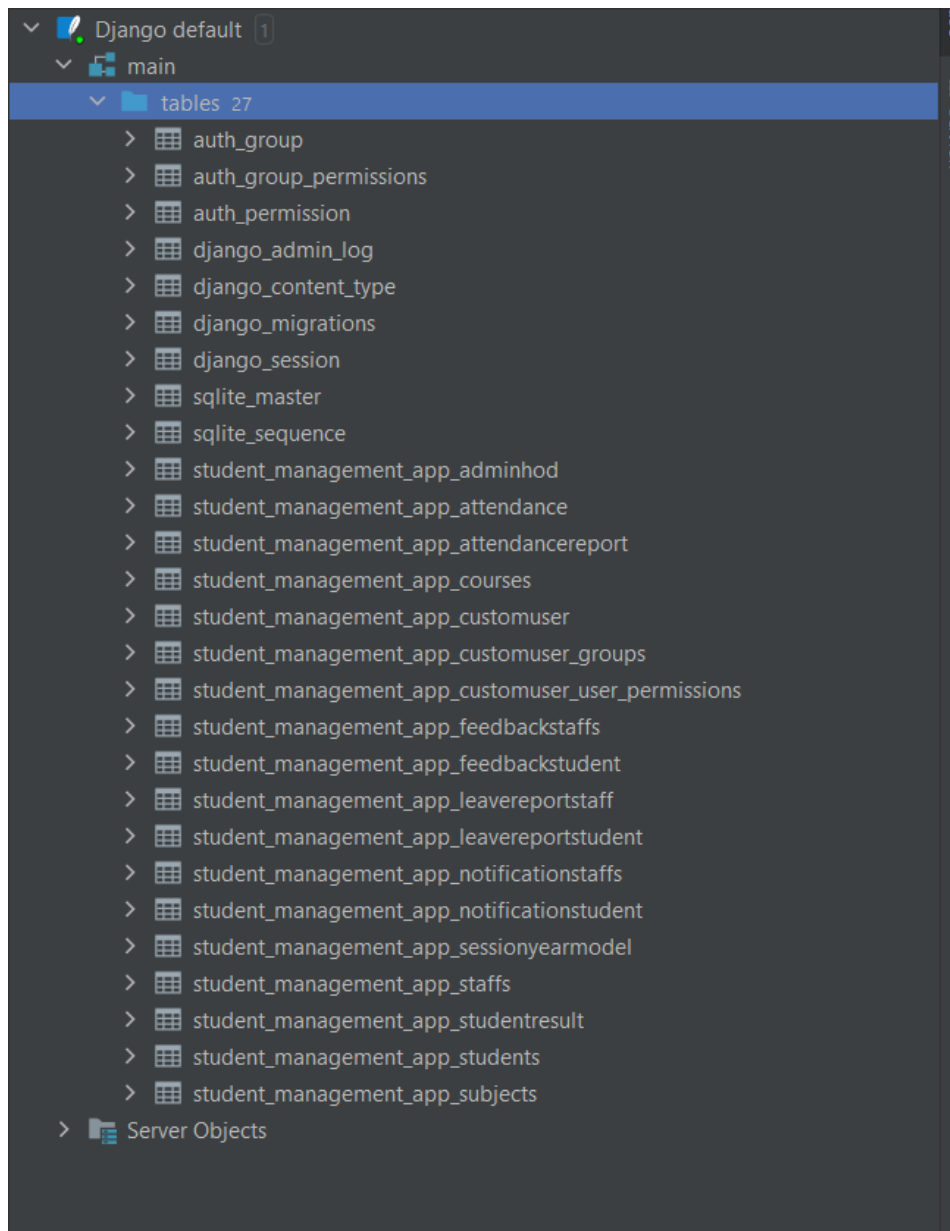
Қатысуды көру бетінде сабаққа сеанс уақытында қандай студенттер қатысты қарай аламыз. Қатысу көрі беті негізгі ұяшықтары бұл: сабақ, сеанс уақыты осы екеун таңдап болған сон қатысу күнін көре аламыз және астында бізде қатысқан студенттер шығады (3.22-сурет).



3.22-сурет Қатысуды қарау бет көрінісі

3.2 SQLite деректор қорындағы ақпараттардың сақталуы

Барлық кестелердің көрінісі Pycharm IDE программалық жүйесінде SQLite деректер қоры кіріктірілген сондықтан да біз сол арқылы барлық кестелерді pycharm программалық жүйесінен бақылай аламыз (3.23-сурет).



3.23-сурет – Кестелердің көрінісі

Басты кесте көрінісі. Осы кестеде біз барлық қолданушылар туралы ақпаратты қарай аламыз. Басты кестеде көрсетілген ақпараттар бұл: id, құпия сөз, есімі, тегі, поштасын көре аламыз (3.24-сурет).

id	password	Last_login	is_superuser	username	first_name	Last_name	email	is_staff	is_active
1	pbkdf2_sha256...	2022-04-29 07:38:4...	1	admin	Admin		admin@gmail.com	1	1
2	pbkdf2_sha256...	<null>	0	kunifi	Ivor	Burch	cihezymo@mailinator.net	0	1
3	pbkdf2_sha256...	<null>	0	rizubin	Gretchen	Powell	tafu@mailinator.net	0	1
4	pbkdf2_sha256...	2020-07-09 16:45:1...	0	gimamomuh	Phyllis	Strickland	sikoka@mailinator.com	0	1
5	pbkdf2_sha256...	2020-07-09 16:45:2...	0	disany	Colby	Rosales	zetewyv@mailinator.com	0	1
6	pbkdf2_sha256...	<null>	0	zakygena	Jolie	Nelson	silas@mailinator.com	0	1
7	pbkdf2_sha256...	2020-07-03 07:37:4...	0	rojopop	Wesley	Cervantes	voku@mailinator.com	0	1
8	pbkdf2_sha256...	<null>	0	viwewood	Alan	Decker	luqoj@mailinator.net	0	1
9	pbkdf2_sha256...	<null>	0	gasif	Bruce	Obrien	nosylun@mailinator.com	0	1
10	pbkdf2_sha256...	<null>	0	hunurezomi	Mohammad	Golden	dobezyxacy@mailinator.net	0	1
11	pbkdf2_sha256...	<null>	0	cesohigago	Ifeoma	Collier	kiso@mailinator.com	0	1
12	pbkdf2_sha256...	<null>	0	tywifajak	Irene	Galloway	gixakibag@mailinator.net	0	1
13	pbkdf2_sha256...	<null>	0	givudu	Larissa	Grimes	ponec@mailinator.net	0	1
14	pbkdf2_sha256...	<null>	0	tacig	Cullen	Curry	tejeso@mailinator.com	0	1
15	pbkdf2_sha256...	<null>	0	cugenana	Skyler	Carter	rynex@mailinator.com	0	1
16	pbkdf2_sha256...	<null>	0	misola	Fulton	Strong	caracax@mailinator.net	0	1
17	pbkdf2_sha256...	<null>	0	xokefy	Candice	Blackburn	tamozlo@mailinator.net	0	1
18	pbkdf2_sha256...	<null>	0	sapijuwa	Hoyt	Simpson	guduh@mailinator.com	0	1
19	pbkdf2_sha256...	2022-04-29 07:37:4...	0	b.abdreshov@mail.ru	Baglan	Abdreshov	b.abdreshov@mail.ru	0	1
20	pbkdf2_sha256...	2022-04-29 06:59:2...	0	staf@gmail.com	Staf		staf@gmail.com	0	1
21	pbkdf2_sha256...	<null>	0	lutozu	Alexander	Hardy	zicyj@mailinator.com	0	1
22	pbkdf2_sha256...	<null>	0	bypopymiz	Walker	Reeves	qabib@mailinator.com	0	1
23	pbkdf2_sha256...	<null>	0	caxuvediv	Colorado	Shelton	junoke@mailinator.com	0	1
24	pbkdf2_sha256...	<null>	0	robatyhib	Zeph	Gregory	paqi@mailinator.com	0	1
25	pbkdf2_sha256...	<null>	0	dubypazany	Armand	West	fyfez@mailinator.com	0	1
26	pbkdf2_sha256...	<null>	0	nuwysne	Scarlett	Padilla	lunu@mailinator.com	0	1

3.24-сурет – Басты кесте көрінісі

Сабақтар кестесінің SQLite тағы сақталу көрінісі. Сабақтар кестесі негізгі 6 бөлімнен тұрады олар: id, сабақ аты, құрылған уақыты, жаңартылған уақыты және байланысқан курс id мен жауап беретін мұғалім id ден тұрады (3.25-сурет).

id	subject_name	created_at	updated_at	course_id_id	staff_id_id
1	PHP	2020-06-26 16:03:11.580536	2022-04-01 12:57:56.271777	3	20
2	Python	2020-06-26 16:03:18.923926	2020-06-26 16:03:18.923985	3	4
3	Data Science	2020-07-03 08:26:56.663451	2022-04-01 12:58:03.150134	1	20
4	Front end	2020-07-05 13:25:08.878218	2022-04-01 13:23:57.795926	2	20
5	Back end	2020-07-05 13:25:18.353760	2022-04-01 13:24:02.801434	2	20

3.25-сурет – Сабақтар кестесі

Студенттердің бағалар кестесінің SQLite тағы сақталу көрінісі. Бағалар кестесінің SQLite тағы сақталу көрінісі. бағалар кестесі негізгі 7 бөлімнен тұрады олар: id, сабақтаң алған бағасы, сабақтағы қатысуы, құрылған уақыты, жаңартылған уақыты және байланысқан студент id мен сабақ id ден тұрады (3.26-сурет).

id	subject_exam_marks	subject_assignment_marks	created_at	updated_at	student_id_id	subject_id_id
1	50	50	2020-07-07 05:40:29.583150	2022-04-01 13:27:06.908915	1	1
2	68	50	2020-07-07 05:41:40.091522	2020-07-07 05:41:40.091565	6	1
3	60	50	2020-07-07 05:48:19.201166	2020-07-07 05:48:19.201211	2	1
4	50	50	2022-04-01 13:27:14.774057	2022-04-01 13:27:14.774057	12	4

3.26-сурет – Бағалар кестесі

ҚОРЫТЫНДЫ

Дипломдық жұмысты қорытындылай , Python Django фреймворкі негізінде құрылған веб қосымшасы дайындық орталығы саласына үлкен үлесін қосады деп ойлаймын. Бұл дегеніміз ең бастысы барлық жұмысты артық қағаздан арылуына зор септігін тигізеді.

Қазіргі кездегі көптеген ірі компанияларға қажетті жобалардың бірі болып саналады. Барлық жұмыс орталықтарына белгілі бір тәжірибелік қабілеттер керек, сол қабілеттерді алу үшін дайындық орталықтары қолданылады. Дайындық орталықтары қарқынды дамуда, мысалыға айтқанда, Қазақстандағы ірі компаниялар Kaspi lab және Kolesa academia сияқты адамдарды даярлауға жаңа білім беруге және де сәтті өткен жағдайда оларға жұмыс ұсынады, осы мысал сияқты көптеген компаниялар өз өзіне жұмыскерлерді белгілі бір немесе өзінің дайындық орталығы арқылы оқытып өзіне керек мамандарды дайындайды. Бұл проект қазіргі уақыттағы дайындық орталығы саласындағы көптеген қиындықтарды шешуге көмегін беретін қосымша.

Қорытындылай келе, қазіргі кездегі көп қолданбалы және де қарқынды дамып жатқан жүйелердің бірі. Бұл жобаның мақсаты дайындық орталығын дамыту және де оңайлату болып табылады. Ең басты ерекшелігі адамдарға оқуға үйренуге деген оңайлатылған түрін көрсетеді, көптеген ірі компанияларды байланыстырып оларға бір дайындық орталығының базасын құруға болады.

ПАЙДАЛАНЫЛҒАН ӘДЕБИЕТТЕР ТІЗІМІ

- 1 Тема: унифицированный язык моделирования UML // Сайттың электронды нұсқасы <http://samara.mgpru.ru/~dzhadzha/dis/15/200.html>
- 2 Диаграмма прецедентов // Сайттың электронды нұсқасы <https://lucid.app/documents#/dashboard>
- 3 Python Django Материал из Национальной библиотеки им. Н.Э.Баумана <https://ru.bmstu.wiki/Django#:~:text=%D0%94%D0%B6%D0%B0%D0%BD%D0%B3%D0%BE%20%D0%B1%D1%8B%D0%BB%20%D1%81%D0%BE%D0%B7%D0%B4%D0%B0%D0%BD%20%D0%BE%D1%81%D0%B5%D0%BD%D1%8C%D1%8E%202003,%D0%B2%20%D1%87%D0%B5%D1%81%D1%82%D1%8C%20%D0%B3%D0%B8%D1%82%D0%B0%D1%80%D0%B8%D1%81%D1%82%D0%B0%20Django%20Reinhardt.>
- 4 Djanog // Сайттың электронды нұсқасы <https://ru.wikipedia.org/wiki/Django>
- 5 Что такое SQLite? // Сайттың электронды нұсқасы <https://habr.com/ru/post/149356/>
- 6 Что такое SQLite и зачем она нужна ? // Сайттың электронды нұсқасы <https://unetway.com/tutorial/sqlite>
- 7 Введение в SQLite // Сайттың электронды нұсқасы <https://ru.wikipedia.org/wiki/SQLite>
- 8 Архитектура SQLite // Сайттың электронды нұсқасы <https://russianblogs.com/article/1781106567/>
- 9 Все что нужно знать о JQuery // Сайттың электронды нұсқасы <https://ru.wikipedia.org/wiki/JQuery>
- 10 Test-Driven Development with Python: Obey the Testing Goat: Using Django, Selenium, and JavaScript: PacktPublishing O'Reilly Media; 2nd edition (August 2, 2017), – 945 б
- 11 Django. Разработка веб-приложений на Python | Форсье Джефф, Биссекс Пол : Packt Publishing Символ-Плюс, 2020. – 40 б
- 12 Python: Основы программирования // Сайттың электронды нұсқасы <https://vk.cc/ccVYBa>.
- 13 What is Use Case Diagram? // Сайттың электронды нұсқасы <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-use-case-diagram/>.
- 14 Peterson R. ER Diagram: Entity Relationship Diagram Model, DBMS Example // Сайттың электронды нұсқасы <https://www.guru99.com/er-diagram-tutorial-dbms.html>
- 15 Особенности SQLite // Сайттың электронды нұсқасы <https://unetway.com/tutorial/sqlite#:~:text=Особенности%20SQLite,в%20одном%20окросс-платформенном%20диске.>
- 16 PyCharm // Сайттың электронды нұсқасы <https://www.jetbrains.com/ru-ru/pycharm/>
- 17 Что же представляет из себя Django // Сайттың электронды нұсқасы

<https://tutorial.djangogirls.org/ru/django/>

18 Документация Django // Сайттың электронды нұсқасы

<https://www.djangoproject.com/>

19 William S. Vincent – Django for APIs: Build Web APIs with Python and Django
// Chapter 7: User Authentication

20 Дронов В. Django: практика создания Web-сайтов на Python – СПб.: БХВ – Петербург, 2016. – 528 б.

21 Django Introduction // Сайттың электронды нұсқасы
<https://developer.mozilla.org/en-US/docs/Learn/Server-side/Django/Introduction>

22 USE Case диаграмма // Сайттың электронды нұсқасы
https://flexberry.github.io/ru/fd_use-case-diagram.html

А Қосымшасы

(міндетті)

А.1.1 Дайындық орталығы үшін ақпараттық жүйені жобалау және әзірлеу техникалық тапсырма

Дайындық орталығы үшін ақпараттық жүйені жобалау және әзірлеу үшін сайт бағдарламасы жобаланды. Дайындық орталығы үшін ақпараттық жүйені жобалау қазіргі кезде ең қолданбалы орталықтардың біреуі болып саналады. Қазіргі кезде барлық жұмыс орталықтарына белгілі бір тәжірибелік қабілеттер керек сол қабілеттерді алу үшін дайындық орталықтары қолданылады.

А.1.2 Жобаның мақсаты

Жобаның мақсаты қазіргі кездегі кәсіпорындарға арналған дайындық орталығының ақпараттық жүйесін құру. Бұл жоба негізінен Дайындық орталығы саласындағы кәсіпорындарға арналған. Қазіргі кездегі көптеген кәсіпорындардағы жұмыстарды бір жүйеге келтіріп жұмыстарың жеңілдету. Барлық ақпаратты орталықтандыру. Қолдануға ыңғайлы және де көзге жағымды веб қосымша әзірлеу.

А.1.3 Жалпы сипаттамасы

Веб қосымша жалпы Python Django программалау тілі және SQLite деректер қорында әзірленген. Барлық ақпараттар SQLite деректер қорында сақталады. Басты іс әрекеттер Python Django программалау тілі арқылы жасалады.

А.1.4 Қауіпсіздік бойынша талаптар

Сайт шабуылдардың негізгі түрлеріне қарсы негізгі қорғанысты қамтамасыз етуі керек: сайттаралық сценарийлер (XSS), SQL инъекциясы, CSRF осалдықтары. Қауіпсіздік сертификаттарымен қамтамасыз ету. Мысалы SSL сертификаты.

SSL сертификаттары веб-сайт пен браузер арасында сенімдірек деректерді тасымалдауды қамтамасыз етеді. Жіберілетін ақпарат электрондық

А қосымшасының жалғасы
(міндетті)

Цифрлық қолтаңбамен қорғалған, бұл оның құпиялылығын қамтамасыз етуге көмектеседі және қауіпсіз қосылуға кепілдік береді.

Бағдарламалық құралдың бір көшірмесін сыртқы медиада сақтау ұсынылады. Сондай-ақ, егер хостингтік компания мұндай қызметті ұсынбаса, дерекқор файлын үнемі сыртқы медиаға көшіру ұсынылады.

Б Қосымшасы (міндетті)

Бағдарлама мәтіні

1 Дайындық орталығы жүйесіне кіру

```
<a href="{ % url 'login' % }"><b></b>Дайындық орталығы </a>
</div>
<!-- /.login-logo -->
<div class="card">
  <div class="card-body login-card-body">
    <input type="email" class="form-control" placeholder="Email"
name="email">
    <input type="password" class="form-control" placeholder="Password"
name="password">

  <p class="login-box-msg">Кіру</p>
```

2 Студенттің басты бет көрінісінің HTML коды

```
<h4 style="margin-left: 10px; margin-top: 5px;">Студенттерді басқару
жүйесі | Студенттерді басқару тақтасы</h4>
```

```
<ul class="navbar-nav ml-auto">

<!-- Notifications Dropdown Menu -->
<li class="nav-item dropdown">
  <a class="nav-link" data-toggle="dropdown" href="#" aria-expanded="false">
    <i class="fas fa-cog"></i>
  </a>
  <div class="dropdown-menu dropdown-menu-lg dropdown-menu-right"
style="left: inherit; right: 0px;">

    <a href="{ % url 'student_profile' % }" class="dropdown-item">
      <i class="fas fa-user-edit mr-2"></i> Профильді жаңарту
    </a>

    <div class="dropdown-divider"></div>
    <a href="{ % url 'logout_user' % }" class="dropdown-item">
      <i class="fas fa-power-off mr-2"></i> Жүйеден шығу
    </a>
```

Б қосымшасының жалғасы (міндетті)

```
<div class="dropdown-divider"></div>
</div>
</li>
</ul>
```

2 Қызметкердің басты бет көрінісінің HTML коды

```
<h4 style="margin-left: 10px; margin-top: 5px;">Студенттерді басқару
жүйесі | қызметкердің басқару панелі</h4>
```

```
<ul class="navbar-nav ml-auto">
```

```
<!-- Notifications Dropdown Menu -->
```

```
<li class="nav-item dropdown">
```

```
<a class="nav-link" data-toggle="dropdown" href="#" aria-expanded="false">
```

```
<i class="fas fa-cog"></i>
```

```
</a>
```

```
<div class="dropdown-menu dropdown-menu-lg dropdown-menu-right" style="left:
inherit; right: 0px;">
```

```
<a href="{ % url 'staff_profile' % }" class="dropdown-item">
```

```
<i class="fas fa-user-edit mr-2"></i> Профильді жаңарту
```

```
</a>
```

```
<div class="dropdown-divider"></div>
```

```
<a href="{ % url 'logout_user' % }" class="dropdown-item">
```

```
<i class="fas fa-power-off mr-2"></i> Шығу
```

```
</a>
```

```
<div class="dropdown-divider"></div>
```

```
</div>
```

3 Админ басты бет көрінісінің HTML коды

```
<h4 style="margin-left: 10px; margin-top: 5px;">Студенттерді басқару
жүйесі Admin</h4>
```

Б қосымшасының жалғасы (міндетті)

```
<ul class="navbar-nav ml-auto">

<!-- Notifications Dropdown Menu -->
<li class="nav-item dropdown">
  <a class="nav-link" data-toggle="dropdown" href="#" aria-expanded="false">
    <i class="fas fa-cog"></i>
  </a>
  <div class="dropdown-menu dropdown-menu-lg dropdown-menu-right" style="left:
inherit; right: 0px;">

    <a href="{ % url 'admin_profile' % }" class="dropdown-item">
      <i class="fas fa-user-edit mr-2"></i> Профильді жаңарту
    </a>

    <div class="dropdown-divider"></div>
    <a href="{ % url 'logout_user' % }" class="dropdown-item">
      <i class="fas fa-power-off mr-2"></i> Шығу
    </a>
    <div class="dropdown-divider"></div>
  </div>
```

4 Models.py деректер қорының құрылуы және бастты кілттері

```
from django.contrib.auth.models import AbstractUser
from django.db import models
from django.db.models.signals import post_save
from django.dispatch import receiver
```

```
class SessionYearModel(models.Model):
    id = models.AutoField(primary_key=True)
    session_start_year = models.DateField()
    session_end_year = models.DateField()
    objects = models.Manager()
```

Б қосымшасының жалғасы (міндетті)

```
# Overriding the Default Django Auth User and adding One More Field  
(user_type)
```

```
class CustomUser(AbstractUser):  
    user_type_data = ((1, "HOD"), (2, "Staff"), (3, "Student"))  
    user_type = models.CharField(default=1, choices=user_type_data,  
max_length=10)
```

```
class AdminHOD(models.Model):  
    id = models.AutoField(primary_key=True)  
    admin = models.OneToOneField(CustomUser, on_delete = models.CASCADE)  
    created_at = models.DateTimeField(auto_now_add=True)  
    updated_at = models.DateTimeField(auto_now=True)  
    objects = models.Manager()
```

```
class Staffs(models.Model):  
    id = models.AutoField(primary_key=True)  
    admin = models.OneToOneField(CustomUser, on_delete = models.CASCADE)  
    address = models.TextField()  
    created_at = models.DateTimeField(auto_now_add=True)  
    updated_at = models.DateTimeField(auto_now=True)  
    objects = models.Manager()
```

```
class Courses(models.Model):  
    id = models.AutoField(primary_key=True)  
    course_name = models.CharField(max_length=255)  
    created_at = models.DateTimeField(auto_now_add=True)  
    updated_at = models.DateTimeField(auto_now=True)  
    objects = models.Manager()
```

```
# def __str__(self):  
#     return self.course_name
```

```
class Subjects(models.Model):  
    id =models.AutoField(primary_key=True)
```


Б қосымшасының жалғасы
(міндетті)

```
subject_name = models.CharField(max_length=255)
course_id = models.ForeignKey(Courses, on_delete=models.CASCADE,
default=1) #need to give default course
staff_id = models.ForeignKey(CustomUser, on_delete=models.CASCADE)
created_at = models.DateTimeField(auto_now_add=True)
updated_at = models.DateTimeField(auto_now=True)
objects = models.Manager()
```

```
class Students(models.Model):
    id = models.AutoField(primary_key=True)
    admin = models.OneToOneField(CustomUser, on_delete = models.CASCADE)
    gender = models.CharField(max_length=50)
    profile_pic = models.FileField()
    address = models.TextField()
    course_id = models.ForeignKey(Courses, on_delete=models.DO_NOTHING,
default=1)
    session_year_id = models.ForeignKey(SessionYearModel,
on_delete=models.CASCADE)
    created_at = models.DateTimeField(auto_now_add=True)
    updated_at = models.DateTimeField(auto_now=True)
    objects = models.Manager()
```

```
class Attendance(models.Model):
    # Subject Attendance
    id = models.AutoField(primary_key=True)
    subject_id = models.ForeignKey(Subjects, on_delete=models.DO_NOTHING)
    attendance_date = models.DateField()
    session_year_id = models.ForeignKey(SessionYearModel,
on_delete=models.CASCADE)
    created_at = models.DateTimeField(auto_now_add=True)
    updated_at = models.DateTimeField(auto_now=True)
    objects = models.Manager()
```

```
class AttendanceReport(models.Model):
    # Individual Student Attendance
    id = models.AutoField(primary_key=True)
    student_id = models.ForeignKey(Students, on_delete=models.DO_NOTHING)
```

Б қосымшасының жалғасы
(міндетті)

```
attendance_id = models.ForeignKey(Attendance, on_delete=models.CASCADE)
status = models.BooleanField(default=False)
created_at = models.DateTimeField(auto_now_add=True)
updated_at = models.DateTimeField(auto_now=True)
objects = models.Manager()
```

```
class LeaveReportStudent(models.Model):
    id = models.AutoField(primary_key=True)
    student_id = models.ForeignKey(Students, on_delete=models.CASCADE)
    leave_date = models.CharField(max_length=255)
    leave_message = models.TextField()
    leave_status = models.IntegerField(default=0)
    created_at = models.DateTimeField(auto_now_add=True)
    updated_at = models.DateTimeField(auto_now=True)
    objects = models.Manager()
```

```
class LeaveReportStaff(models.Model):
    id = models.AutoField(primary_key=True)
    staff_id = models.ForeignKey(Staffs, on_delete=models.CASCADE)
    leave_date = models.CharField(max_length=255)
    leave_message = models.TextField()
    leave_status = models.IntegerField(default=0)
    created_at = models.DateTimeField(auto_now_add=True)
    updated_at = models.DateTimeField(auto_now=True)
    objects = models.Manager()
```

```
class FeedBackStudent(models.Model):
    id = models.AutoField(primary_key=True)
    student_id = models.ForeignKey(Students, on_delete=models.CASCADE)
    feedback = models.TextField()
    feedback_reply = models.TextField()
    created_at = models.DateTimeField(auto_now_add=True)
    updated_at = models.DateTimeField(auto_now=True)
    objects = models.Manager()
```

```
class FeedBackStaffs(models.Model):
    id = models.AutoField(primary_key=True)
```

Б қосымшасының жалғасы
(міндетті)

```
staff_id = models.ForeignKey(Staffs, on_delete=models.CASCADE)
feedback = models.TextField()
feedback_reply = models.TextField()
created_at = models.DateTimeField(auto_now_add=True)
updated_at = models.DateTimeField(auto_now=True)
objects = models.Manager()
```

```
class NotificationStudent(models.Model):
    id = models.AutoField(primary_key=True)
    student_id = models.ForeignKey(Students, on_delete=models.CASCADE)
    message = models.TextField()
    created_at = models.DateTimeField(auto_now_add=True)
    updated_at = models.DateTimeField(auto_now=True)
    objects = models.Manager()
```

```
class NotificationStaffs(models.Model):
    id = models.AutoField(primary_key=True)
    staff_id = models.ForeignKey(Staffs, on_delete=models.CASCADE)
    message = models.TextField()
    created_at = models.DateTimeField(auto_now_add=True)
    updated_at = models.DateTimeField(auto_now=True)
    objects = models.Manager()
```

```
class StudentResult(models.Model):
    id = models.AutoField(primary_key=True)
    student_id = models.ForeignKey(Students, on_delete=models.CASCADE)
    subject_id = models.ForeignKey(Subjects, on_delete=models.CASCADE)
    subject_exam_marks = models.FloatField(default=0)
    subject_assignment_marks = models.FloatField(default=0)
    created_at = models.DateTimeField(auto_now_add=True)
    updated_at = models.DateTimeField(auto_now=True)
    objects = models.Manager()
```

#Creating Django Signals

It's like trigger in database. It will run only when Data is Added in CustomUser

Б қосымшасының жалғасы
(міндетті)

```
@receiver(post_save, sender=CustomUser)
# Now Creating a Function which will automatically insert data in HOD, Staff or
Student
def create_user_profile(sender, instance, created, **kwargs):
    # if Created is true (Means Data Inserted)
    if created:
        # Check the user_type and insert the data in respective tables
        if instance.user_type == 1:
            AdminHOD.objects.create(admin=instance)
        if instance.user_type == 2:
            Staffs.objects.create(admin=instance)
        if instance.user_type == 3:
            Students.objects.create(admin=instance, course_id=Courses.objects.get(id=1),
session_year_id=SessionYearModel.objects.get(id=1), address="", profile_pic="",
gender="")
```

```
@receiver(post_save, sender=CustomUser)
def save_user_profile(sender, instance, **kwargs):
    if instance.user_type == 1:
        instance.adminhod.save()
    if instance.user_type == 2:
        instance.staffs.save()
    if instance.user_type == 3:
        instance.students.save()
```