

ҚАЗАҚСТАН РЕСПУБЛИКАСЫНЫҢ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

СӘТБАЕВ УНИВЕРСИТЕТІ

Ақпараттық және телекоммуникациялық технологиялар институты

Программалық инженерия кафедрасы

Оразбек Қуаныш Оразбекұлы

Электронды силабус

Дипломдық жобаға
ТҮСІНІКТЕМЕЛІК ЖАЗБА

5B060200 – «Ақпараттану» мамандығы

Алматы 2019

ҚАЗАҚСТАН РЕСПУБЛИКАСЫНЫҢ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ
СӘТБАЕВ УНИВЕРСИТЕТІ

Ақпараттық және телекоммуникациялық технологиялар институты


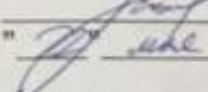
Программалық инженерия кафедрасы

ҚОРҒАУҒА ЖІБЕРІЛДІ

ПИ кафедра меңгерушісі

тех. ғыл. кандидаты,

ассистент-профессор

 Р. Юнусов
"  " 2019 ж.

Дипломдық жобаға
ТҮСІНІКТЕМЕЛІК ЖАЗБА

Тақырыбы: Электронды силабус

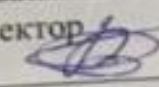
5B060200 – «Ақпараттану» мамандығы

Орындаған

Оразбек Қ. О.

Ғылыми жетекші

Лектор

 Д.А. Баймбетов
" 17 " 05 2019 ж.

Алматы 2019

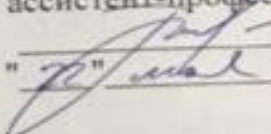
КАЗАҚСТАН РЕСПУБЛИКАСЫНЫҢ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ
СӘТБАЕВ УНИВЕРСИТЕТІ

Ақпараттық және телекоммуникациялық технологиялар институты
Программалық инженерия кафедрасы

5B060200 – «Ақпараттану»

БЕКІТЕМІН

ПИ кафедра меңгерушісі,
тех. ғыл. кандидаты,
ассистент-профессор

 Р. Юнусов
" 21 " мамыр 2019ж.

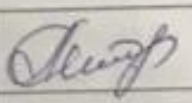
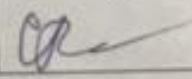
**Дипломдық жоба орындауға
ТАПСЫРМА**

Білім алушыға Оразбек Қуаныш Оразбекұлы
Тақырыбы: «Электронды сиплабус»
Университет ректоры бұйрығының № 1828-б "13" наурыз 2019 ж. шешімімен
бекітілген.
Орындалған жобаның өткізу мерзімі " 21 " мамыр 2019 ж.
Дипломдық жобаның бастапқы мәліметтері: Бұл дипломдық жобада
сиплабусты электронды түрде автоматтандыру. Жобаны құру барысында
Python бағдарламалау тілінің Django фреймворкі, Bootstrap фреймворкін
қолдандым.
Есеп – түсініктеме жазбаның талқылауға берілген сұрақтардың тізімі:
а) жобаны модельдеу, UML диаграммаларын құру;
б) қолданылатын кестелерді анализ жасау;
в) заманауи технологияларды салыстыру;
г) бағдарламаны құру, кітапханаларды қосу, деректерді қосу және тестілеу.
Графикалық материалдар тізімі (міндетті суреттердің нақты көрсетілуімен):
презентацияның 18 слайдпен берілген құжат түрінде ұсынылған.
Ұсынылған негізгі әдебиеттер: 10 пайдаланылған әдебиеттер тізімінен.

Дипломдық жобаны орындау
КЕСТЕСІ

Бөлімдердің атаулары, зерттелген мәселелердің тізімі	Ғылыми жетекшіге және кеңесшілерге ұсыну мерзімі	Ескерту
1. Диплом жұмысының жоспар құрылымын құру.	14.01.2019	жоқ
2. Тапсырма қойылымы және бағдарламалау ортасын таңдау	18.01.2019	жоқ
3. Зерттеу тақырыбы бойынша ғылыми теориялық материалдарды жинау және негізгі бөлім беру бойынша есеп беру жазбасын дайындау	01.02.2019	жоқ
4. Дипломның екінші бөлімі – жобалау сызбаларын дайындау.	15.02.2019	жоқ
5. Жобаның веб-қосымшасын тестілеуден өткізу.	18.03.2019	жоқ
6. Дипломдық жобаға түсіндірме жазба жазуды аяқтау	26.04.2019	жоқ

Дипломдық жұмыс бөлімдерінің кеңесшілерінің аяқталған жұмысқа қойған қолтаңбалары

Бөлімдер атауы	Кеңес берушілер (аты-жөні, тегі, ғылыми дәрежесі, атағы)	Қолтаңба қойылған мерзімі	Қолы
Нормалық бақылаушы	Алгожаева Р.С. Сениор-лектор	16.05.2019	
Бағдарламалық бөлім	Сман Н. О. Ассистент	16.05.2019	

Ғылыми жетекші  Д.А. Байымбетов

Тапсырманы орындауға қабылдап алған студент _____ К. О. Оразбек

Күні «17» 05 2018ж.

АҢДАТПА

Дипломдық жобада кейінгі технологияларды қолдана отырып «электронды силлабус» жобасының веб қосымшасын құру қарастырылған.

Бұл жоба қоғамға қажетті әрі ыңғайлы жүйе болып табылады. Негізгі идея жоғарғы оқу орындарының оқытушыларының уақытын үнемдеу және оқу жүйесін автоматтандыру болып табылады. Оқытушылар тек қана керек мәліметтерді қолдана отырып силлабус дайындап шығарады.

Бұл жобаны құру барысында кейінгі және ыңғайлы технологиялар қолданылды. Бұл технологиялар арқасында осы веб қосымша жылдам әрі қауіпсіз болып табылады. Оның барлығы технологияның мүмкіншіліктеріне жатады.

АННОТАЦИЯ

В данной дипломной работе рассмотрена разработка веб приложений “электронного курса” с использованием последних веб технологий.

Данный проект является удобной и нужной системой для общества. Основные задачи работы автоматизация учебной системы и экономия времени студентов высших учебных заведений. Используя только необходимые данные студенты, могут подготовить курс.

В процессе разработки проекта были использованы последние и удобные технологии. С помощью этих технологии веб приложение является быстрой и безопасной. Все это относится к возможностям данной технологии.

ANNOTATION

In this thesis work is considered the development of web applications of the “electronic syllabus” using the latest web technologies.

This project is a convenient and necessary system for society. The main tasks of the work are the automation of the educational system and the saving of time of students of higher educational institutions. Students can prepare a syllabus using only the necessary data.

During the development of the project, the latest and convenient technologies were used. Using these technologies, the web application is fast and secure. All this relates to the capabilities of this technology.

МАЗМҰНЫ

	Кіріспе	8
1	Жалпы бөлім	10
1.1	Силлабус жайлы түсінік	10
1.2	«Электронды силлабус» веб қосымшасына қысқаша шолу	12
1.3	«Электронды силлабус» веб қосымшасының негізгі идеясы	12
1.4	«Электронды силлабус» веб қосымшасының өзектілігі	12
1.5	«Электронды силлабус» веб қосымшасының қолдану ортасы	13
1.6	«Электронды силлабус» веб қосымшасында қолданылатын технологиялар	13
1.7	«Электронды силлабус» веб қосымшасының функционалдылығы	14
1.8	Аналогтар	15
2	Жобалау бөлімі	16
2.1	UML бірыңғайландырылған модельдеу тілі	16
2.2	Use Case Diagram жайлы	16
2.3	Sequence diagram жайлы	17
2.4	ER диаграмма жайлы	18
3	Қолданбалы бөлім	21
3.1	«Электронды силлабус» қосымшасы	21
3.2	«Электронды силлабус» веб қосымшасының сыртқы дизайны	31
	Қорытынды	33
	Пайдаланылған әдебиеттер тізімі	35
	А қосымшасы. Техникалық тапсырма	36
	Б қосымшасы. Бағдарлама мәтіні	39

КІРІСПЕ

Зерттеу барысында дипломдық жоба жасалды және ол үшін бірінші әдебиеттер тізімі жасалды. Екіншіден, тақырыптың практикалық маңызы мен өзектілігі анықталды. Осы зерттеу барысында ең керегі Веб-сайт болып табылды. Сондықтан қазіргі актуалды тақырып алынған болатын.

Интернет өмірімізге жылдам еніп кетті. Қазіргі уақытта адамдар өз өмірін интернетсіз елестете алмайды. Интернет бізге көптеген мүмкіншілік берді және алды. Көптеген адамдар электрондық пошта, жылдам хабар алмасу бағдарламалары, веб-сайттар, кез-келген ақпаратты, интернет-дүкендерді, блогтарды, әлеуметтік желілерді, онлайн-компьютерлік ойындарды, форумдарды және т.б. қосымшаларға қосылып алған. Шағын фирмалар мен үлкен Корпорациялар көптеген тапсырмаларды орындауға қабілетті корпоративтік Интернет-ресурстарсыз жұмыс істей алмайды. Ғалымдар көптеген компьютерлерді бір үлкен суперкомпьютерге біріктіретін үлкен есептеуіш ресурстар қажет ететін мәселелерді шеше алды.

Пайдаланушы стильді жобаланған, графикалық және анимациялармен шектелмеген, жылдам жүктелетін және веб-браузер терезесінде дұрыс көрсетілетін веб-беттерге кірген әлдеқайда ынғайлы.

Бұл жұмыстың мақсаты – силлабус жасауды автоматтандыру. Яғни таңдау операциялары арқылы ғана дайын pdf форматта силлабус жасап шығару. Осы жоба арқылы оқытушылардың уақытының басым бөлігін студенттермен өткізуге жұмсағанын қарастырдым. Себебі уақытының басым бөлігі документациялармен өтетіні айдан анық.

Дипломдық жобаның орындалуы керек міндеттер:

- веб-технологияларды үйрену;
- үлестірілген нұсқаларды басқару жүйесімен жұмыс жасау;
- жоғарғы оқу орнын зерттеу және талдау жасау;
- мәліметтер базасын оқу;
- Java Script тілдерін және веб-дизайндағы олардың мүмкіндіктерін үйрену;
- басқа жобаларды анализ жасап, керек функционалдарын қолдана отырып үйрену;
- силлабусты қарап, оны жүйелеу;
- Django фреймворкімен Spring Boot фреймворкін салыстырып, ынғайлы әрі функционалы жоғары технологияны таңдау.

«Электронды силлабус» веб қосымшасының ерекшеліктері өте көп. Мәліметтер базасымен сервер арасында байланыс өте жылдам болады және мәліметтер соған сәйкес тез алмасады. MVC паттерні арқылы осы қосымшаны көптеген функционалдар арқылы басқа жобалардан ерекшелеп көрсетеді. Әр қосымша жасалынып болған соң оның қауыпсіздігіне өте жоғары мән беріледі. Себебі қазіргі уақытта кибер шабуылдар жоғары дәрежеде жасалынады. Тағы

бір ерекшелігі ол бұл қосымшаның қауіпсіздігі болып табылады. Себебі Django фреймворкінің өзінің ішкі қауіпсіздігі өте жоғары дәрежеде жұмыс жасалынады.

1. Жалпы бөлім

1.1 Силлабус жайлы түсінік


Силлабус – бұл пәннің оқу-әдістемелік құралы болып табылады. Оған оқылатын пәннің сипаттамасы, мақсаттары мен міндеттері, әр сабақтың тақырыптары мен ұзақтығы, өзіндік жұмыс үшін тапсырмалар, кеңес беру уақыты, мұғалімнің талаптары, бағалау критерийлері және негізгі және қосымша әдебиеттер тізімі кіреді.

Әр семестрдің басында әрбір пәннен университеттің негізгі сайтында, оның ішінде кафедра бетінде керек силлабустар салынады. Және әрбір студентке еркін қолжетімді болатын қажетті силлабустар орналасады. Бұл оқытушы мен студенттердің бір-бірімен оңай әрі бірігіп немесе бөлек жұмыс істеуін оңтайландырады. Сырттай оқып жүрген студенттерге қолдануға оңай болып табылады.

Силлабустарды жазуға қатаң схема жоқ, мұғалім қажет деп санайтын нәрселерді қамтуға құқылы және студенттерге үйрететін пәндерді меңгеру үшін мәліметтерді салуға болады. Дегенмен, оқу жоспарының негізгі бөлімдерін ажыратуға болады:

- титулдық бет – білім беру мекемесінің атауы, сәйкес бөлім - әзірлеуші, пәннің атауы және коды, мамандықтың атауы немесе оқу бағыты, оқу семестрі, осы пәнді оқу үшін кредиттер саны (сағаттар), басқару нысаны.
- мұғалім туралы мәліметтер: мұғалімнің толық атауы, ғылыми дәрежесі мен атағы, лауазымы және кеңес беру орны мен уақыты.
- байланыс ақпараты: келу уақыты және байланыс телефондары (қызметтік, үй), электрондық пошта мекенжайы.
- кредиттер саны.
- күні: курстың жылын және семестрін көрсетеді
- пәннің сипаттамасы, пәннің өзектілігі мен мақсатын қамтиды.
- оқу және тақырыптық жоспардың уақытын сабақтарға байланысты бөлу. Оқу жоспары кесте түрінде ұйымдастырылады: «дәрістер», «практикалық», «зертхана», «жалпы сағат», «сабақ», «студенттердің өзіндік жұмысы».
- студенттерге арналған әдістемелік нұсқау. Силлабустың бұл бөлімі студентке арналған тапсырмалар тізімін, өзіндік жұмыс түрлерінің тізімі мен мазмұнын, оларды ұйымдастыру және енгізу бойынша ұсыныстарды қамтиды. Бұл силлабустың ең жан-жақты бөлімі, ол сабаққа дайындалуға арналған толық нұсқаулық болып табылады.
- негізгі және қосымша әдебиеттер. Негізгі әдебиет: негізгі әдебиеттер көздерінің оңтайлы саны 3-5 негізгі оқулықтар мен нормативтік актілер. Қосымша әдебиеттер: 10-ға дейін атаулар болуы мүмкін.

- бағалау туралы ақпарат.
 - бағалау саясаты.
 - постреквизит.
 - пререквизит.
- Силабус мысалын 1.1-суреттен көруге болады.

Силлабус		Казахский Национальный исследовательский технический университет им К.Сатпаева (КазНУТУ)	1
----------	---	--	---

УТВЕРЖДАЮ
Заведующий кафедрой
«Строительство и строительные
материалы»
Сартаев Д.Т.

« » 201 г.

SYLLABUS

КОД « Железобетонные конструкции I»

3 кредита

Семестр: 2, 2016 - 2017 уч. год

Преподаватель: ассоциированный проф., к.т.н. Жамбакина З.М.

Данные о преподавателе	Время и место проведения		Контактная информация		
	Аудиторных занятий	СРСП (офис часы)	Whats'Up	Аккаунт соцсети	e-mail
Ассоциированный профессор, к.т.н. Геотехника, строительные конструкции	45 часов: 30 час-лекции, 15 час-прак. занятия, 45 час-СРС, 45 час-СРСП, МУК	По расписанию на кафедре	8 701 75 7 7893	нет	zhambakin a@bk.ru

Пререквизит: для изучения дисциплины «Железобетонные конструкции I с основами сейсмостойкого строительства» студентам необходимы знания следующих дисциплин: математика, физика, инженерная механика, строительные материалы, архитектурные конструкции, геотехника, современные компьютерные расчеты.

Постреквизиты: знания данной дисциплины необходимы при дипломном проектировании..

Краткое описание курса: Данная дисциплина отражает современное состояние вопроса проектирования и расчета железобетонных конструкций в объеме, соответствующем программе курса.

Знания, полученные при прохождении дисциплины: студент учится владеть методами определения внутренних усилий, методами расчета прочности и деформативности железобетонных конструкций; приемами

1.1-сурет – Силлабус мысалы

1.2 «Электронды силлабус» веб қосымшасына қысқаша шолу

Қазіргі уақытта ІТ саласы қарқынды дамып жатыр. Көптеген салалар ІТ саласымен бірге жұмыс жасап жатыр. Медицина, экономика, білім беру және т.б. Мысалы, медицина саласында лазермен соқырішекті алып тастау. Сол сияқты басқада салалар қарқынды қолданып келе жатыр.

«Электронды силлабус» өте ыңғайлы автоматтандырылған қосымшасы болып табылады. Оқытушылар ол қосымшаны кез келген жерде қолдануға болады. Уйде, жұмыста, жолда немесе демалыс орындарында отырып қолдануға болады. Ол үшін интернет желісіне кіретін компьютер ғана қажет.

Бұл қосымша көмегімен оқытушылардың уақытын үнемдейміз және жүйке жүйесінің тозбауынан сақтайды. Себебі қазіргі уақытта оқытушылар көп документтермен жұмыс жасайды. Осы қосымшаға дейін силлабус құру үшін WORD программасымен жұмыс жасауға тура келді. Осы қосымшадан кейін оқытушылар уақытын үнемдейтін болады.

Әр оқытушының өз логині болады. Ал оған доступ администратор беретін болады. Авторизация бетінен өтіп болғаннан кейін негізгі бет ашылады. Ал ол бетте әр мұғалімнің өзіне байланысты мағлұматтар болады. Және таңдау операциясы арқылы керек мәліметтерді енгізіп баспаға шығарады. Админ беті арқылы керек мәліметтерді енгізуге, жоюға, өзгертуге болады. Ал қолданушы беті арқылы тек мәлімет қоса аламыз. Бұл қосымшаның негізінде сервер бөлігі жақсы қолданылады. Себебі ол жерде формалармен көп жұмыс жасаймыз. Формалар арқылы тікелей базаға қойылады.

Авторизация бетінде логин және пароль енгізу арқылы кіреді. Одан кейін негізгі бет болады. Осы бетте барлық оперциялар жүзеге асады. Әр аптаға тақырып және лабораториялық жұмыс таңдау қажет болады. Одан әрі сабақ, практикалық сабақтан беретін оқытушы, әдебиеттер тізімі, постреквизит және пререквизит таңдау қажет болады. Осыдан кейін ғана дайын силлабус PDF форматта шығады.

1.3 «Электронды силлабус» веб қосымшасының негізгі идеясы

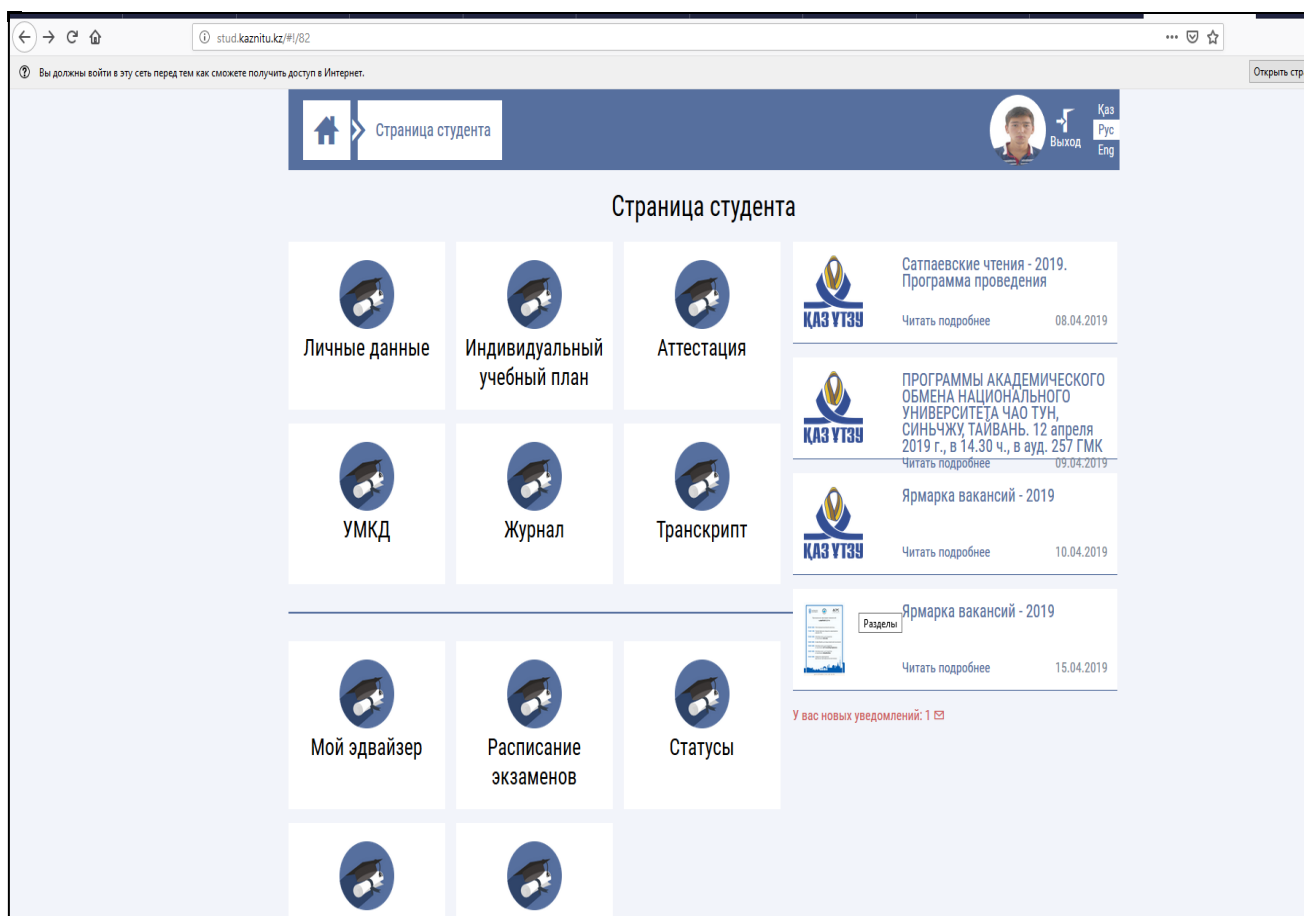
Бұл дипломдық жобаның негізі идеясы жоғары оқу орындарының оқу жүйесін автоматтандыру. Яғни оқытушылар және студенттер заманға сай өмір сүру принципі. Уақытының басым бөлігін документациядан да пайдалы әрекеттер жасауға мүмкіншілік беру. Осы негізгі идея арқылы дипломдық жоба жасалынып жатыр.

1.4 «Электронды силлабус» веб қосымшасының өзектілігі

Бұл дипломдық жобаның өзектілігі уақытты үнемдейді және орындалатын орны. Уақыт бүгінгі өмірде маңызды. Осы веб қосымша арқылы біз тек қана жұмыс орнында ғана емес барлық интернеті бар жерлерде жұмыс жасай аламыз. Жұмыс орнында үлгермей жатсаңыз сіз уайымдамай уйге барып жұмысыңызды жалғастыра аласыз.

1.5 «Электронды силлабус» веб қосымшасының қолдану ортасы

Бұл дипломдық жоба жоғары оқу орындарында қолданылады. Біз алғашында оны өзіміздің Сәтпаев Университетіне қолданып көргіміз келеді. Өзіміздің оқу орнымызда жақсы қолданысқа ие болса онда басқа жоғары оқу орындарына өзімнің жобамды ұсынып көремін. Осы қосымшаны университетіміздің негізгі сайтына салынатын болады. Алғашқыда ол бөлек қосымша болып жұмыс істейді. Негізгі сайттын көрінісі 1.2-суретте көрсетілген.



1.2-сурет – Негізгі сайт

1.6 «Электронды силлабус» веб қосымшасында қолданылатын технологиялар

Негізінен веб қосымша құру барысында 2-ге бөлінеді. Ол: клиент бөлігі және сервер бөлігі болып. Екеуі көбіне бірге жүреді. Менің жобам негізінен сервер бөлігін кеңінен қолданады. Себебі мәліметтермен көп жұмыс жасалынады.

Клиент бөлігі қолданушыларға көрінетін бөлік десек те болады. Оны тағы қолданушы интерфейсі деп айтады. Қолданушы интерфейсі өте маңызды бөлігі болып табылады. Себебі қолданушы веб бетке кірген сәтте ол адамға түсініксіздеу болса онда қолданғысы келмей қалады. Сол үшін кейінге технологияларды қолданған дұрыс болады. Менің жобамда клиент бөлігі JAVASCRIPT, BOOTSTRAP, CSS технологияларын қолдана отырып жасалған болатын.

Сервер бөлігі веб қосымшасының ең негізгі бөлігі болып саналады. Ол сайттың негізгі қызметтеріне жауап береді. Сервер бөлігі қолданушыға көрінбейтін болып қызмет жасайды. Ол мәліметтерді алмасуға және мәліметтер базасымен байланыс жасауға қолданылады. «Электронды силлабус» жобасында ол бөлігіне Python тілінің Django фреймворкін қолдандым. Ол қазіргі уақытта кеңінен қолданылып жатыр және өте жоғары деңгейде дамып келе жатыр. Негізгі қолданатын паттерн ол MVC. Model, View, Controller принципімен жұмыс жасалынады. Model дегеніміз мәліметтер базасына кесте қосу үшін осы модель бетіне жазып кету қажет. View түсінігі қолданушы интерфейсіне қандай мәліметтер көрсету қажеттілігін реттейді. Controller адресстерге тікелей жауап береді. Керек html беттерін қандай адреспен көрсететініне жауап беретін паттерн бөлігі.

1.7 «Электронды силлабус» веб қосымшасының функционалдылығы

Әр жасалған программалық қосымша жоғары функционалды болуы қажет. Егер функционалды болмаса онда ол бәсекелестік жағдайда ол қолданылмай қалуы мүмкін. Мысалы, Olx.kz қосымшасымен Market.kz қосымшасын салыстыратын болсақ, онда Olx.kz қосымшасының функционалдылығы жоғары болғандықтан оның қолданушылар өте көп.

Менің қосымшамда функционалдылығы жоғары десек те артық болмайды. Ең негізгі функционалдылығы ол жылдам жұмыс жасауы. Егер біздің жасалынып жатқан жобамыз асықпай жұмыс жасайтын болса онда қолданушылар қолданбай қалады.

Силлабус-қа кесте қосу үшін бөлек веб бет жасалған. Және сол веб бетке кіріп әр аптаға тақырып және зертханалық жұмыс жасалынады. Оның барлығы қолданушыға яғни оқытушы интерфейсіне көрініп тұрады.

Авторизация бетінде логинға деп тұрған жерге әр лектор өзінің логинімен кұпия сөзін жасған сәтте менің қосымшам жүйеде өңдеп сол лекторға ғана қатысты сабақтарды және лабораториялық жұмысты интерфейске шығарып береді. Ол функционалдың мақсаты, лекторға байланысты емес сабақтарды көрсетпеу. Егер қолданушы көретін болса көп жазбалар шығып тұрады. Оның барлығы SQL тілінде жасалады.

Бір ғана постреквизит, пререквизит, әдебиеттер таңдау ғана емес көп жазбалар таңдай алатын мүмкіндік. Себебі постреквизит, пререквизит, әдебиеттер көп болуы мүмкін.

1.8 Аналогтар

«Электронды силлабус» веб қосымшасына ұқсайтын қосымшалар мүлдем жоқ. Себебі оқу орындар автоматтандыруды кеңінен ашып қолданбайды. Сол мақсатта жоғары оқу орындар дәл осы сәттен бастап автоматтандыруды қолымызға алу қажет. Жалпы қазіргі оқу орындарға қатысты қосымшалар санаулы ғана. Оның бірі әр оқу орнының жеке веб сайты. Яғни портал. Ол веб қосымшаның ішінде бағалау беті, сабақ кестесін көретін бет және тағы басқа функционалдары бар.

2 Жобалау бөлімі

2.1 UML бірыңғайландырылған модельдеу тілі

UML тілі – бұл стандартты визуалды модельдеу тілі болып саналады. UML-дің мақсаты - барлық объектілі-бағдарланған әдістермен қолдануға болатын стандартты белгілерді ұсыну және предшестволық белгілердің үздік элементтерін таңдау және біріктіру. UML кең ауқымды қосымшаларға арналған. Осылайша, ол көптеген жүйелер мен әрекеттер үшін конструкцияларды қамтамасыз етеді (мысалы, бөлінген жүйелер, талдау, жүйені жобалау және орналастыру).

UML бағдарламалық жасақтаманы әзірлеу мен құжаттаудың айналасындағы хаос түрінде пайда болды. 1990-шы жылдарда бағдарламалық қамтамасыз ету жүйесін көрсету және құжаттаудың бірнеше түрлі жолдары болды. Бұл жүйені бейнелеудің неғұрлым бірыңғай әдісі қажет болды және нәтижесінде 1994-1996 жж. UML Rational Software-да жұмыс істейтін үш бағдарламашылар әзірледі. Кейінірек ол 1997 жылы стандарт ретінде қабылданды және бірнеше жаңартуларды ғана алғаннан бері стандартты қалды.[3]

Жүргізу диаграммалары жүйедегі объектілердің динамикалық мінез-құлқын көрсетеді, уақыт бойынша жүйеге өзгерістер тізбегі ретінде сипатталуы мүмкін, мінез-құлық диаграммасының жеті түрі бар:

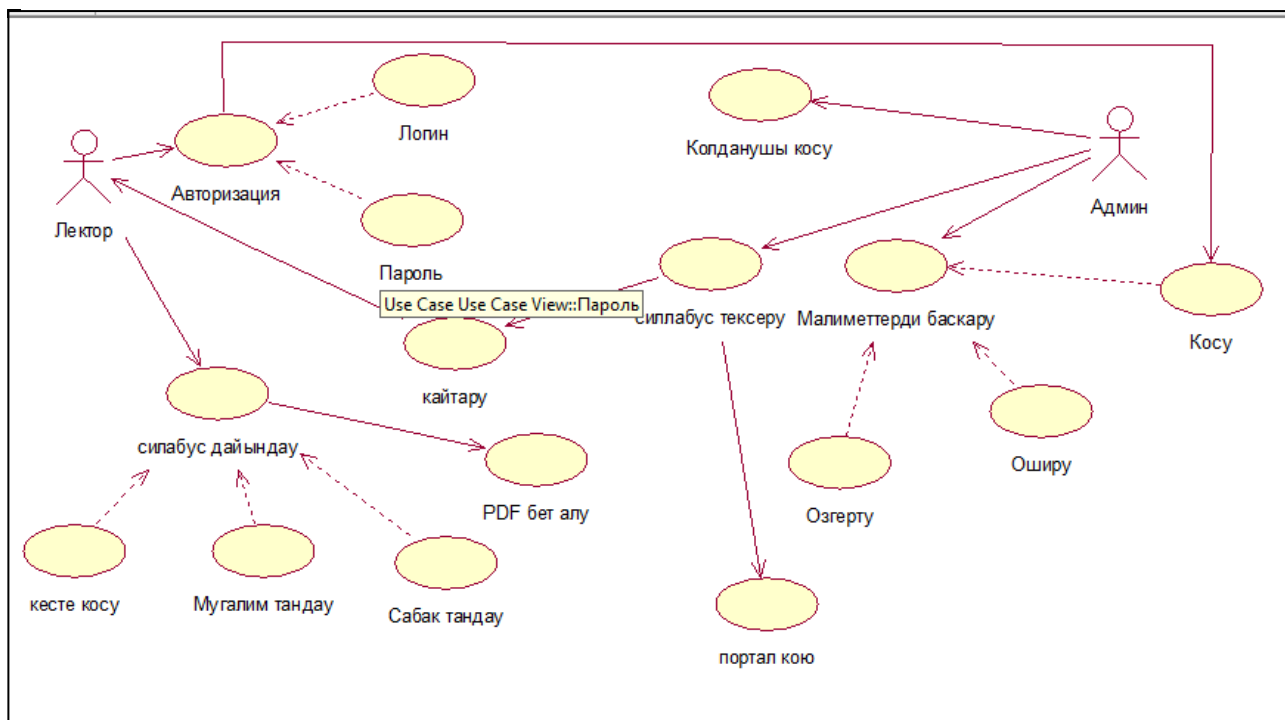
- Use Case Diagram;
- Activity Diagram;
- State Machine Diagram;
- Sequence Diagram;
- Communication Diagram;
- Interaction Overview Diagram;
- Timing Diagram.[8]

2.2 Use Case Diagram жайлы

Use Case диаграммасы ең алдымен, жүйенің функционалды талаптарын анықтау және бүкіл даму үдерісін бақылау үшін жасалған. Талдау, дизайн, тестілеу сияқты барлық негізгі іс-шаралар қолдану жағдайлары негізінде жүзеге асырылады. Актерлар мен қолдану варианттарының арасында әртүрлі байланыстар бар. Өзара әрекеттесудің негізгі түрлері:

- жай ассоциация;
- бағытталған ассоциация;
- мұрагерлік;

- кеңейту(extend);
 - қосу.[5]
- Жобаның Use Case диаграммасы 2.1-суретте көрсетілген.



2.1-сурет – Use Case диаграммасы

2.3 Sequence diagram жайлы

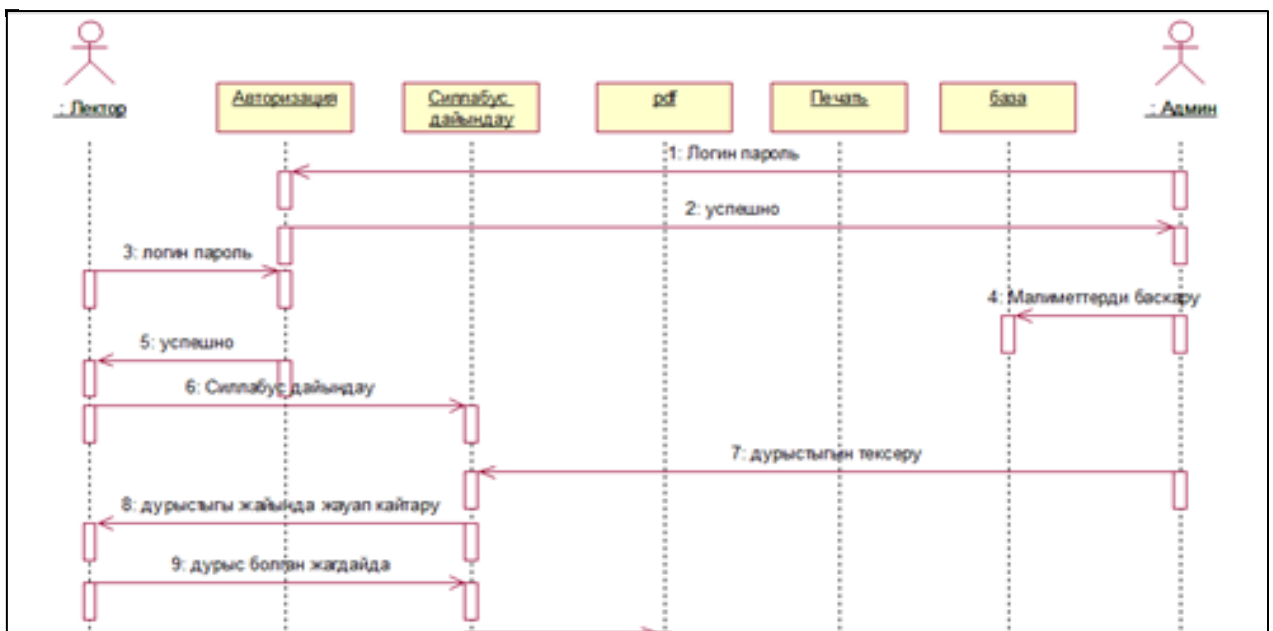
Уақытында ұйымдастырылған объектілердің өзара әрекетін көрсететін диаграмма. Атап айтқанда, ол өзара әрекеттесуге байланысты объектілерді және олар алмасатын хабарлардың дәйектілігін бейнелейді. Кезектілік диаграммасында екі өлшем бар: тік (уақыт) және көлденең (өзара әрекеттесуге тартылған нысандар). Әдетте, уақыт осі жоғарғыдан төменге бағытталған (қажет болса, ось бағыты алмасуы мүмкін). Әдетте, бұл маңызды хабарлар тізбегі, алайда нақты уақыттағы бағдарламалар үшін уақыт осі нақты метрика ретінде қызмет етуі мүмкін. Объектілердің көлденең орналасу тәртібі маңызды емес.

Әрбір объект жеке тік баған ретінде бейнеленген. Нысанның белгісі (астыңғы сызылған нысан атауы орналасқан тікбұрыш) бағанның үстіне, бұл нысан жасалатын хабардың көрсеткісінің соңында орналастырылады. Егер объект диаграммада көрсетілген бірінші операциядан бұрын болған болса, нысан таңбасы диаграмманың жоғарғы жағында, кез келген хабарлардан бұрын көрсетіледі. Одан объектінің жойылуына кедергі келтіретін сызық сызығы

пайда болады (егер бұл сәт диаграммамен сипатталған болса). Бұл желі өмір жолы деп аталады. нысан жұмысын тоқтатады, онда нүкте, онда үлкен крест (X), немесе объектінің бұзылуын тудырады, немесе объект өзін жояды нүктесінде хабарлама көрсете көрсеткі жағын тұрған. Объектінің қызметі кезеңі екі қатарлы сызық ретінде бейнеленген. Мұндай кезең, белсенді нысанның немесе пассивті нысанның жандандыру уақытта өмір - бұл (сондай-ақ, жүз туындаған операциялар қайтару операциясы күтіп жұмсалған уақытты қамтиды) объектінің кезінде жұмыс уақыты болып табылады. Нысан рекурсивті түрде, тікелей немесе жанама түрде қоңырау шалса, онда екіншісі бірінші қосарлы үздіксіз сызықтың жанында орналасады. Осылайша, қос активтендіру бейнеленеді (теориялық тұрғыдан, мұндай қос сызықтардың саны екіден көп болуы мүмкін). Нысандардың тәртібі маңызды емес, бірақ ыңғайлылық тұрғысынан көрсеткілердің өтуі тиіс өмір жолдарының арасындағы қашықтықты азайту жақсы. Терминалдың жанына активтендіру туралы түсініктеме қоюға болады. Жобаның Sequence диаграммасы 2.2-суретте көрсетілген.

Диаграмманың негізгі элементтері:

- орындалатын түйіндер;
- объектілер;
- ауысулар;
- басқару түйіндері;
- коннекторлар;
- топтастырылған элементтер.



2.2-сурет – Sequence диаграммасы

2.4 ER диаграмма жайлы

ER диаграммасы мәліметтер базасында кеңінен қолданады. Ол екі кестенің байланысын көрсетеді. Кестелер байланыссыз мүлдем мағынасы болмайды. Байланыстың 3 түрі болады: бірдің бірге, бірдің көпке, көптің көпке. Осы байланыстар иерархиялық байланыс деп атайды. Бірдің бірге байланысы бір жазбада бір ғана үлгі болуы қажет. Бірдің көпке қатынасы бір жазбаның көп үлгісі болуы қажет болып саналады. Ал көптің көпке қатынасы көп жазбаның көп үлгісі болуы қажет. Менің диаграммамда көрсетілгендей бірдің көпке қатынасы қолданылады. Қазіргі уақытта үлкен компаниялар, банктер осы диаграмманы қолданады. Себебі кестелердің байланысын көрсету қажет болады.

Қазіргі уақытта NoSQL кеңінен қолданылып жатыр. Ол жерде ER диаграммасы мүлдем қолданылмайды. Мүлдем байланыссыз. Байланыс болу үшін primary key және foreign key болуы қажет. Өлшемі аз мәліметтермен жұмыс жасауға ыңғайлы жасалған. Оның үлкен артықшылығы тез жұмыс жасауға мүмкіндік береді.

«Электронды силлабус» веб қосымшасы реляциялық мәліметтер базасын қолданады. Сол себепті ER диаграммасын қолданамын. «Электронды силлабус» веб қосымшасында 6 негізгі кесте бар. User, Teacher, Adebiet, Takugrup, Zert_jumys және Subject. User - ге лектор оқытушылар кіреді. Ол кестеде оқытушының логині, аты-жөні, почтасы, құпия сөзі, ұялы телефоны кіреді. Логиніне негізгі кілт жасалған. Teacher кестесінде университеттің барлық оқытушылары туралы ақпарат жиналған. Бұл кестенің сыртқы кілті User және Subject кестелерімен бірдің көпке және көптің көпке қатынасы арқылы байланыс жасайды. Себебі бір оқытушы көп сабақтан бере алады және бір сабақты көп оқытушы өткізе алады. Сол үшін Teacher және Subject кестелері көптің көпке қатынасымен байланысқан.

Subject кестесінде барлық дисциплиналар(сабақ) туралы мәліметтер сақталады. Осы кесте мынададай атрибуттардан тұрады: сабақ атауы, кредит саны, сабақтың толық анықтамасы. Және сыртқы кілттері Takugrup және Adebiet кестелерімен көптің көпке қатынасымен байланысып тұр. Әр сабақта бір немесе одан да көп тақырып болуы әбден мүмкін. Ал бір тақырып көп сабақта болады. Дәл сол сияқты бір пәнде бірден көп әдебиеттер тізімі болса бір әдебиет көп пәнде болуы мүмкін.

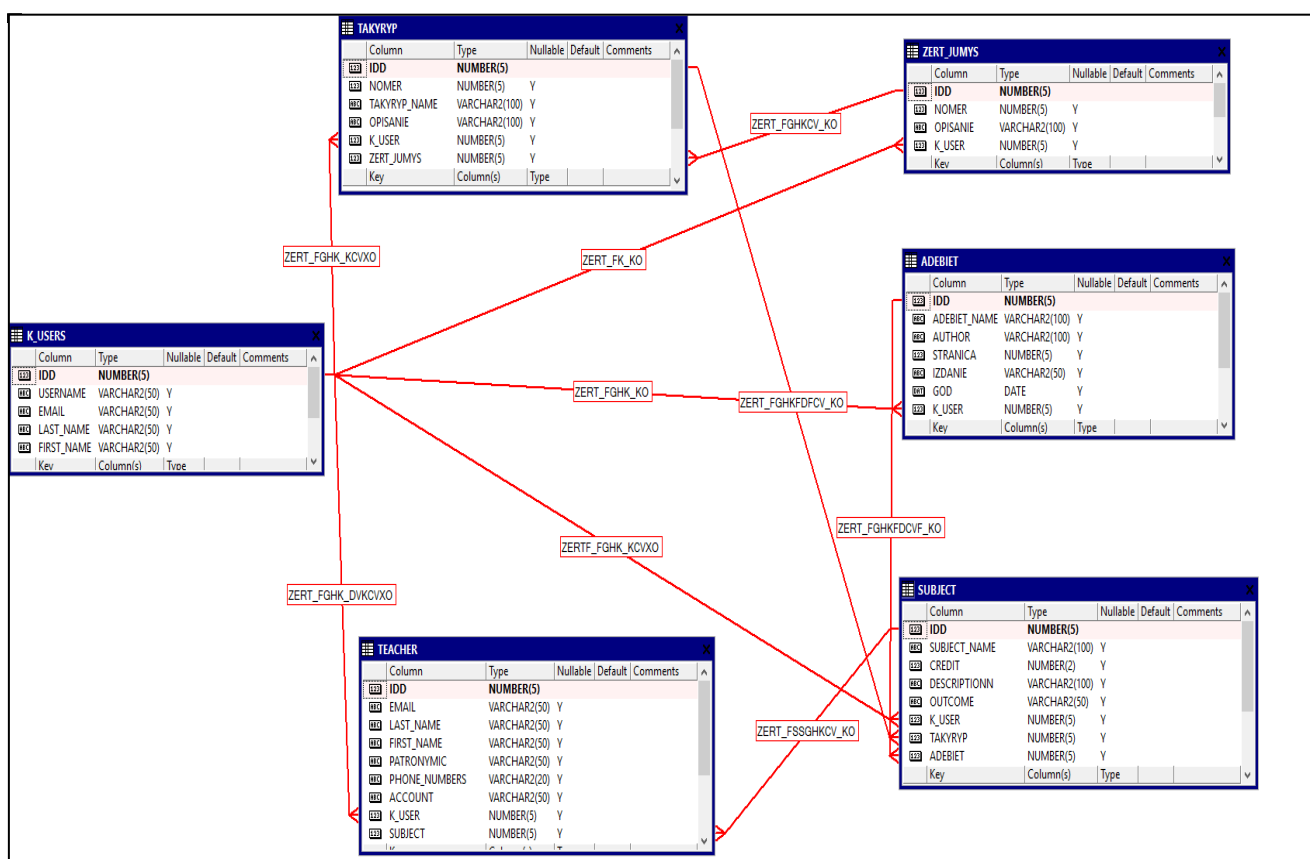
Takugrup кестесі номер, тақырыптың аты, толық анықтама деген атрибуттардан тұрады. Ол Zert_jumys кестесімен бірдің көпке қатынасымен байланыс жасайды. Бір тақырыпта көп зертханалық жұмыс болады және бір зертханалық жұмыс көп тақырыпта болмайды.

Әдебиеттер тізімі кестесінде тақырып атауы, типі, автор, шығарылған мекеме, жылы, бет деген атрибуттардан тұрады. Әдебиеттер тізімі тек қана әдебиеттер сақталынған кестесімен байланысқан.

Зертханалық жұмыс кестесі осындай атрибуттардан тұрады: номері және зертханалық жұмыс атауы. Зертханалық жұмыс тек қана тақырып кестесімен байланыс жасайды.

User кестесі барлық кестелермен байланыс жасалынған. Осы байланыс арқасында біз логин енгізіп авторизация бетінен өтіп негізгі бетке кірген сәтте кіріп тұрған қолданушыға қатысты мәліметтер ғана экранға көрінетін болады.

Тағы да негізгі кестелерден бөлек Keste және Lit деген қосымша кестелер бар. Ол кестелердің қызметі уақытша мәліметтерді сақтау үшін қолданылады. Осы веб қосымшаға авторизация жасаған сәтте қосымша кестелер тазарып отырады. Жобаның ER диаграммасы 2.3-суретте көрсетілген.



2.3-сурет – ER диаграммасы

3 Қолданбалы бөлім

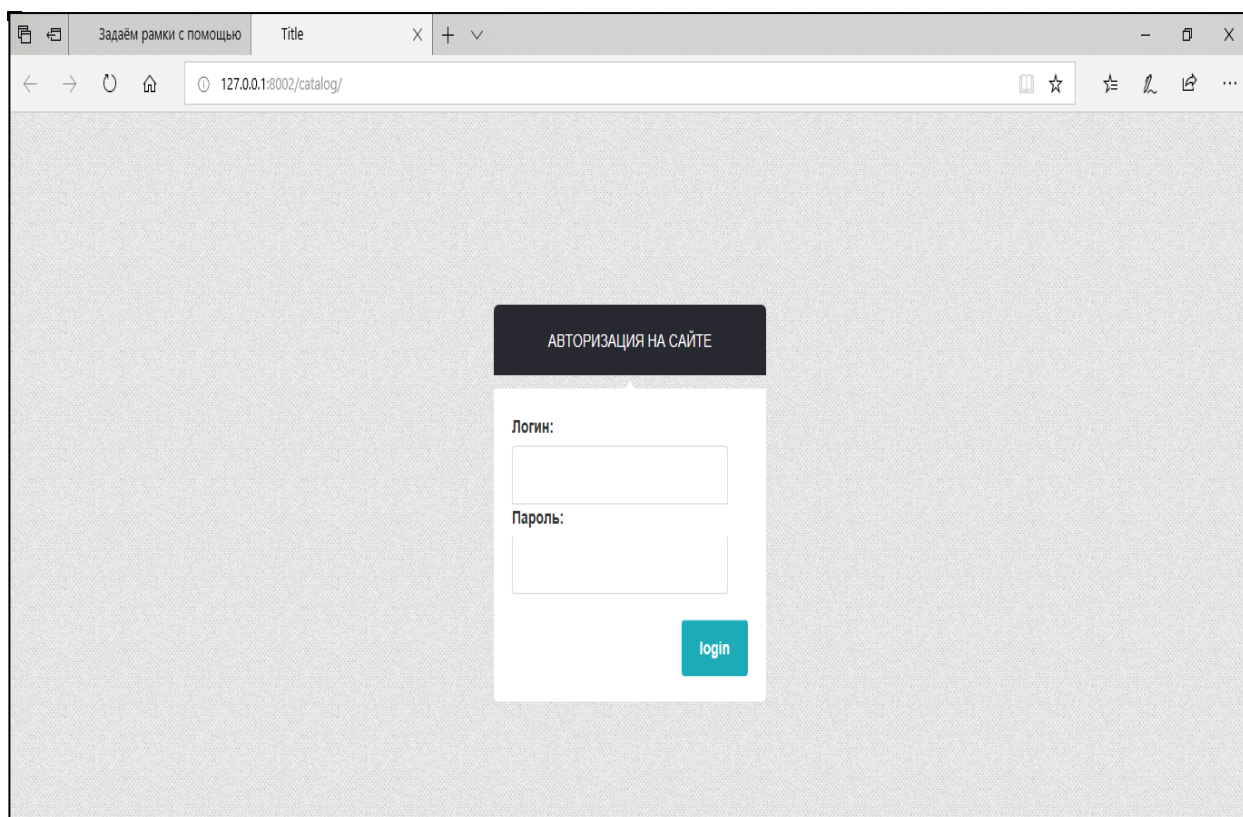
3.1 «Электронды силлабус» қосымшасы

«Электронды силлабус» қосымшасы веб – технологиялары арқылы жасалғаннан кейін ол міндетті түрде сервер бөлігінен және клиент интерфейсінен тұрады. Сервер бөлігінде көбіне формалармен жұмыс істеуге тура келеді. Және сервер бөлігінде мәліметтер базасымен жұмыс жасалынды. Клиент бөлігі HTML, JAVASCRIPT, BOOTSTRAP және CSS технологиялары арқылы жасалған. Ал сервер бөлігі Питон тілінің DJANGO фреймворкі көмегімен жасалған.

Қолданылатын мәліметтер базасы SQLite. Ол DJANGO фреймворкімен бірге жүреді. Ол аз мәліметтерді сақтай алады. Егер мәліметтер көлемі көп болатын болса онда ORACLE қолданған жөн болады. Бұл базада қолданылған байланыс түрі бірден көпке.

«Электронды силлабус» қосымшасының негізгі беттері 4 беттен тұрады. Олар: админ беті, авторизация беті, таңдау беті және дайын силлабус беті.

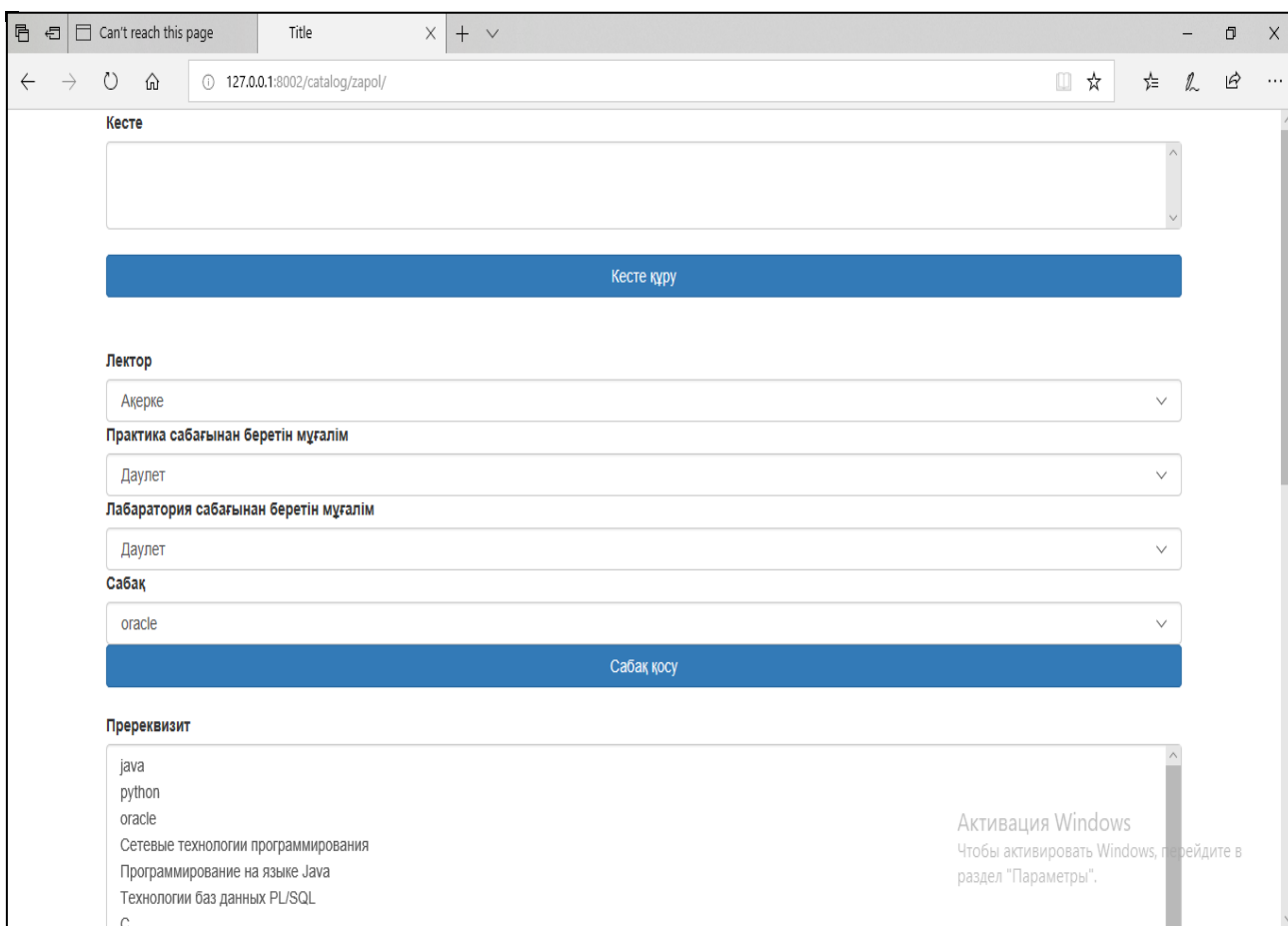
Бірінші бет авторизация беті. Яғни оқытушылар өзінің логині және құпия сөзін жазу керек болады. «Электронды силлабус» веб қосымшасының авторизация беті 3.1-суретте көрсетілген.



3.1-сурет – «Электронды силлабус» веб қосымшасының авторизация беті

Осы авторизация бетінен дұрыс өтіп болғаннан кейін таңдау бетіне өтеміз. Егер де дұрыс өтілмеген жағдайда Администраторға ескерту керек болады. Біздегі администратор мүмкіншілігі superuser болып табылады. Таңдау бетінде біз енгізу және таңдау операциялармен жұмыс жасаймыз. Суретте көрініп тұрғандай кесте құру батырмасын басқан кезде басқа веб бет ашылады. «Электронды силлабус» веб қосымшасының таңдау беті 3.2-суретте көрсетілген. Негізгі бетте қолданылатын функционалдар:

- кесте құру
- лектор таңдау
- практика сабағынан беретін мұғалім
- лаборатория сабағынан беретін мұғалім
- сабақ таңдау
- сабақ қосу
- постреквизит таңдау
- пререквизит таңдау
- әдебиеттер тізімін таңдау
- әдебиет қосу
- PDF форматта сақтау



3.2-сурет – «Электронды силлабус» веб қосымшасының таңдау беті

Кесте құру бетінде әр аптада тақырып таңдау операциясы және зертханалық жұмыс таңдау операциясын қолдана отырып бір семестрдің жобасы шығады. Осы бетте әр апта, тақырып және зертханалық жұмыс таңдап жіберу батырмасын басқан уақытта сол бетте кішігірім кесте визуалды түрде көрсетіледі. Барлық 15 аптаны толық толтырып болғаннан кейін кері қайту батырмасын басу қажет. Сол батырма басылған сәтте негізгі бетке қайтып оралады. Визуалды кесте де апта, лекция және зертханалық жұмыс көрсетіліп тұрады. «Электронды силлабус» веб қосымшасының кесте құру беті 3.3-суретте көрсетілген.

Апта	Лекция тақырыбы	Практикалық жұмыстың тақырыбы
1	Spring Data Jpa	kuka_zert
2	PL/SQL, Developer Орнату және айнымалылармен танысу	kuka_zert
3	Процедура Процедураның параметрлері	kuka_zert

3.3-сурет – «Электронды силлабус» веб қосымшасының кесте құру беті

Негізгі бетке қайта оралғаннан кейін сабақ қосу батырмасын басқан сәтте жаңа бет ашылады. Ол беттің қызметі мәліметтер базасына жаңа сабақ қосу. Яғни Subject кестесіне жаңа жазба құрады. Егер негізгі бет интерфейсінде осы оқытушының пәні болмаған жағдайда осы операция жұмыс жасайды. «Электронды силлабус» веб қосымшасының сабақ қосу беті 3.4-суретте көрсетілген.

localhost:8000/catalog/subform/

Сабақ аты:

Кредит:

Мазмұны:

Пән өту кезіндегі алған білім:

Постреквизит:

Тақырып:

- python
- Python тілін орнату
- Синтаксиспен танысу
- Цикл, шартты операторлар

Әдебиет:

- Программирование на Java
- Django для новичков
- Python and Django
- <https://www.python.com/>

3.4-сурет – «Электронды силлабус» веб қосымшасының сабақ қосу беті

Дәл алдыңғы веб бет сияқты әдебиеттер қосу беті де жұмыс жасайды. Adebiet кестесіне жаңа жазба жазады. Ол операцияның барлығы админ бетінде де қолданады. «Электронды силлабус» веб қосымшасының әдебиет құру беті 3.5-суретте көрсетілген.

localhost:8000/catalog/adform/

Әдебиет аты:

Автор:

Издание:

Бет саны:

Шығарылған жылы:

ДД. ММ. ГГГГ

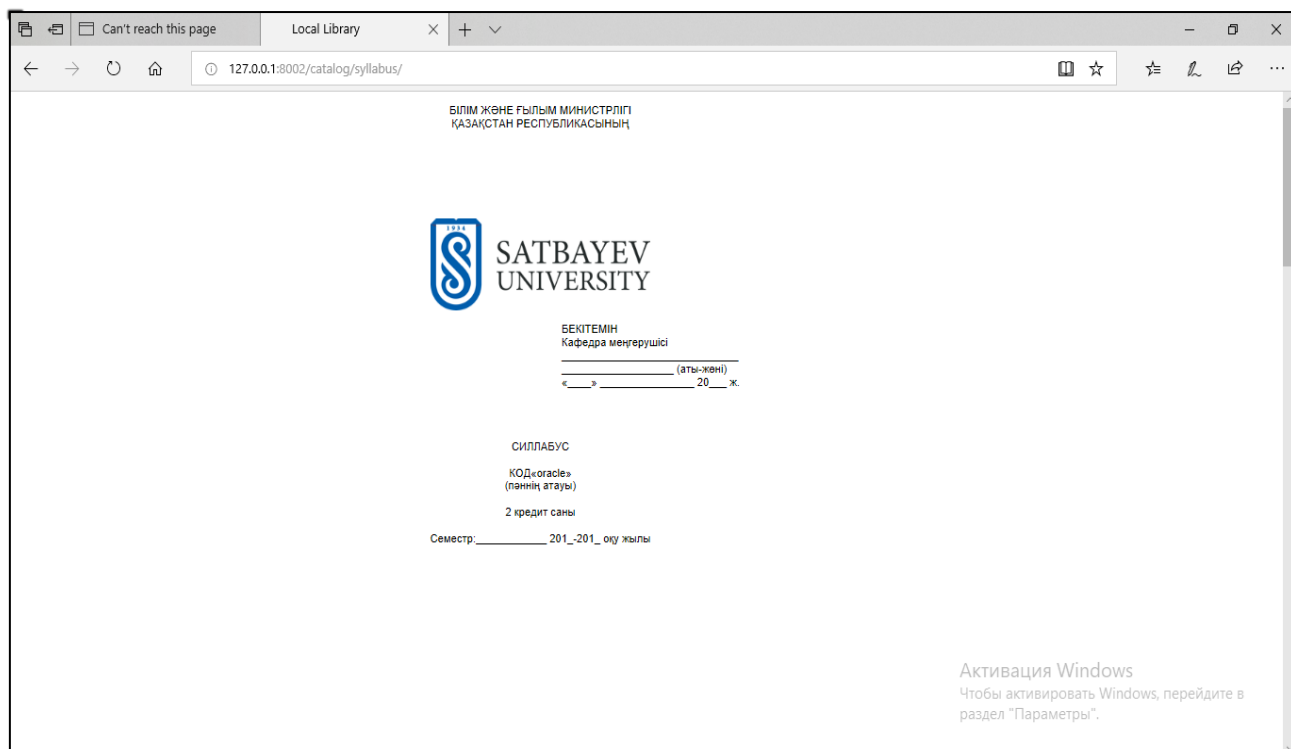
Типы:

Сақтау

3.5-сурет – «Электронды силлабус» веб қосымшасының әдебиет құру беті

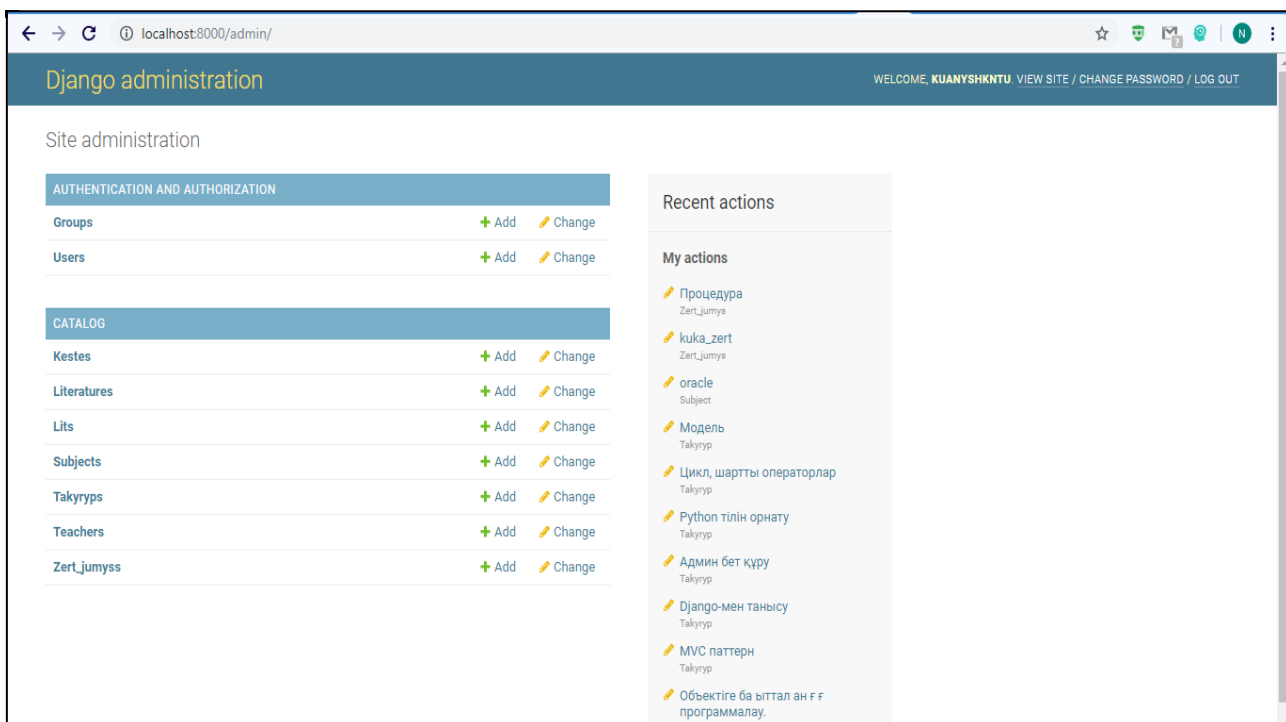
Таңдау бетінде барлығын толық таңдап болғаннан кейін «Басып Шығару» батырмасын басқан сәтте дайын силлабус беті шығатын болады. Осы бетте

«PDF» батырмасын басқан жағдайда бізге толық PDF форматта силлабус шығатын болады. «Электронды силлабус» веб қосымшасының дайын силлабус беті 3.6-суретте көрсетілген.



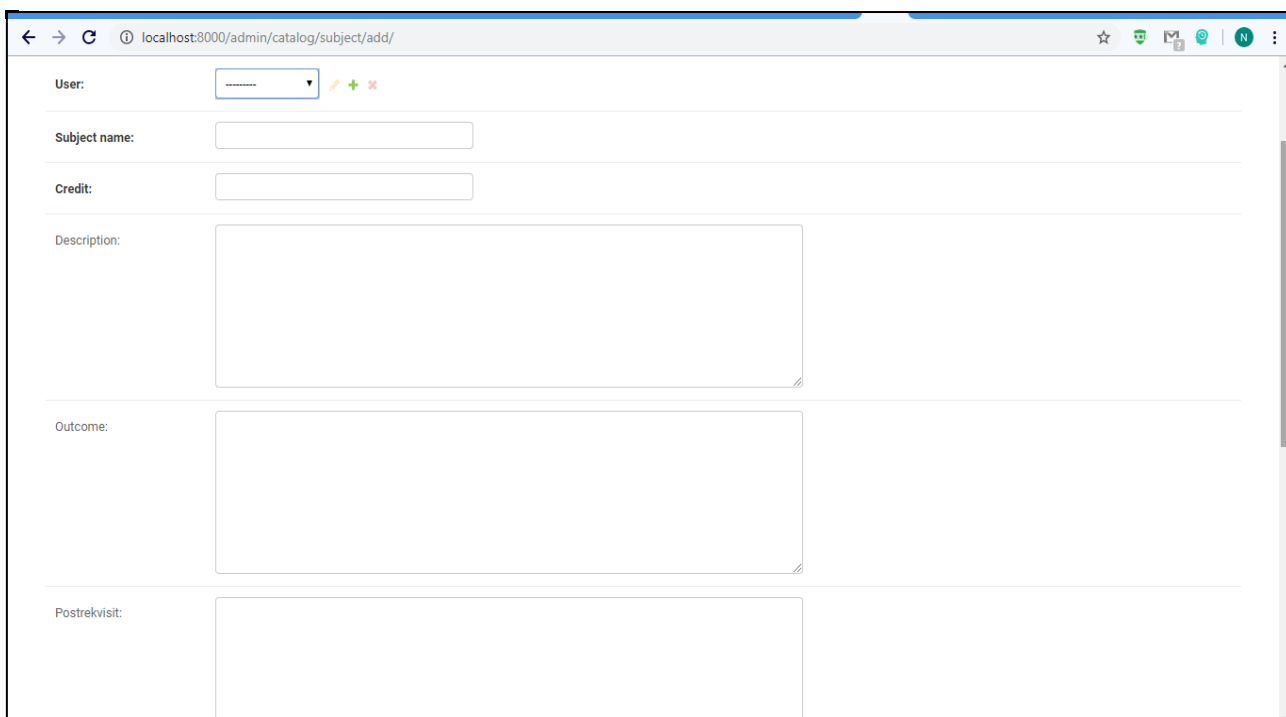
3.6-сурет – «Электронды силлабус» веб қосымшасының дайын силлабус беті

Қолданушы интерфейсіне толықтай шолу жасап болдық. Ендігі кезек сервер бөліктеріндегі ең негізгі бет – админ беті. Админ бетінде суперқолданушыларға мәліметтерді өзгерту, жою және құру мүмкіншіліктері болуы қажет. Админ бетіне өту үшін url жазатын орынға сілтеме/*admin* сөзін қосу қажет. Және бірінші суперқолданушы құру қажет болады. Оны *python3 manage.py createsuperuser* командасы арқылы құрылады. Админ бетінің негізгі бетінде барлық кестелер көрінеді. Яғни осы кестелерге DML операцияларын жасай аламыз. Delete, insert және update. Actions деген бөлімшеде соңғы жасалған операциялар көрсетіледі. «Электронды силлабус» веб қосымшасының админ беті 3.8-суретте көрсетілген.



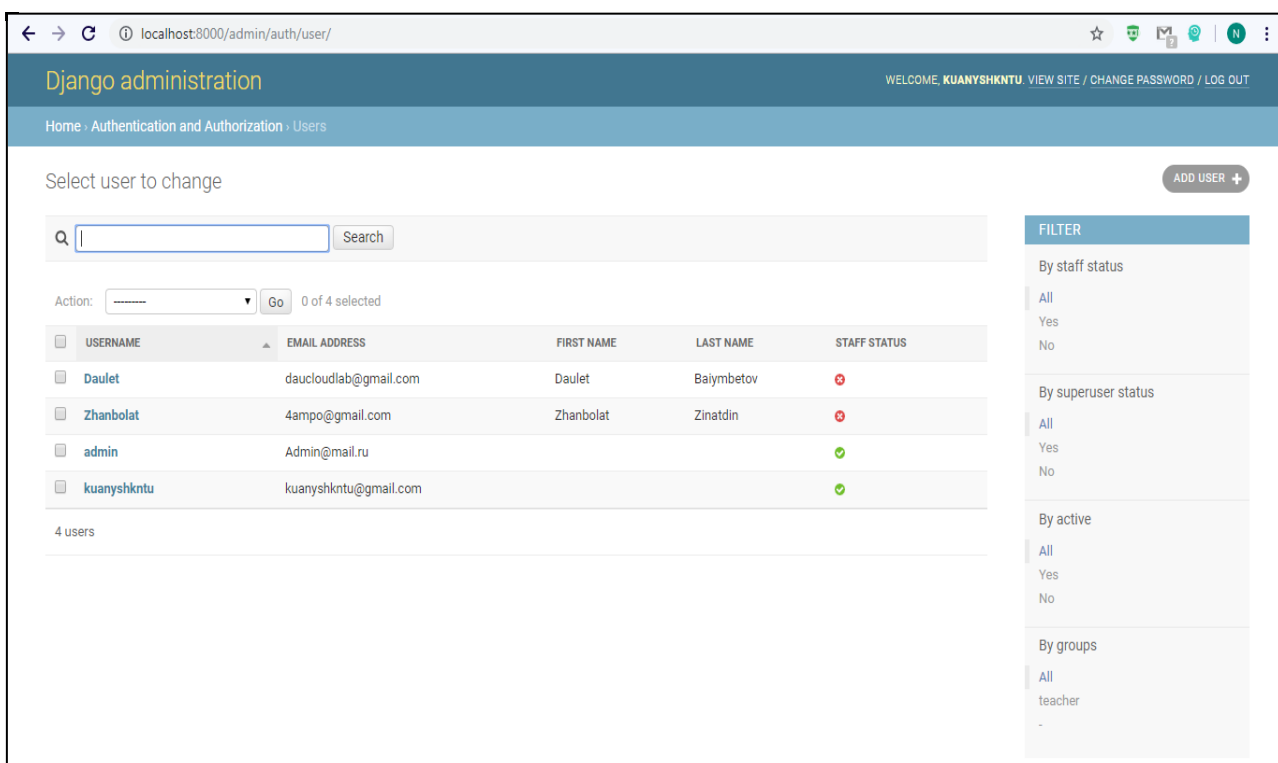
3.7-сурет – «Электронды силлабус» веб қосымшасының админ беті

Админ беті арқылы сабақ қосу операциясы төмендегі суретте көрсетілген. Дәл қолданушы интерфейсі сияқты жұмыс жасайды. «Электронды силлабус» веб қосымшасының администратор беті арқылы жаңа сабақ қосу беті 3.8-суретте көрсетілген.



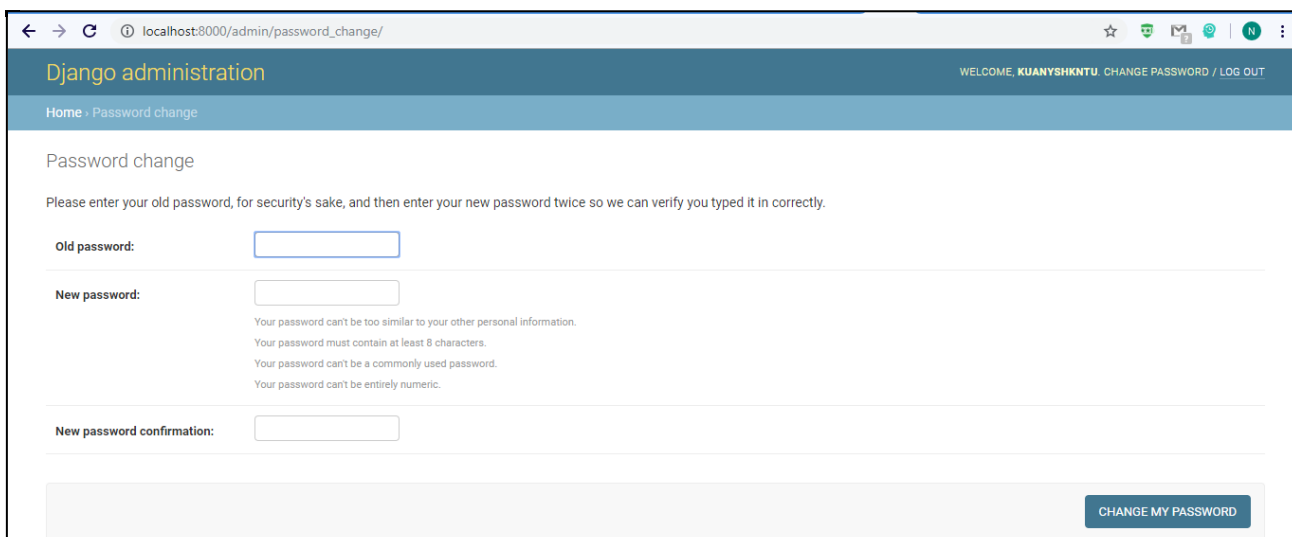
3.8-сурет – «Электронды силлабус» веб қосымшасының администратор беті арқылы жаңа сабақ қосу беті

Қолданушыларды өзгерту немесе өшіру үшін осы суреттегі сілтемеге кіріп керек операцияны қолдана аламыз. Осы кестенің сыртқы байланыс кестесі Groups деген кесте болып табылады. Ол жалпы Django фреймворкінде қолданылатын стандартты кесте. Олай бөлу себебі қолданушылардың әртүрлі ролі болу мүмкін. Менің қосымшамда ондай мүмкіндіктер жоқ. Кейін жобам дамып жатқан болса ол міндетті түрде қосылуы қажет әрі керек. «Электронды силлабус» веб қосымшасының администратор беті арқылы қолданушыларды қосу, өшіру, өзгерту беті 3.9-суретте көрсетілген.



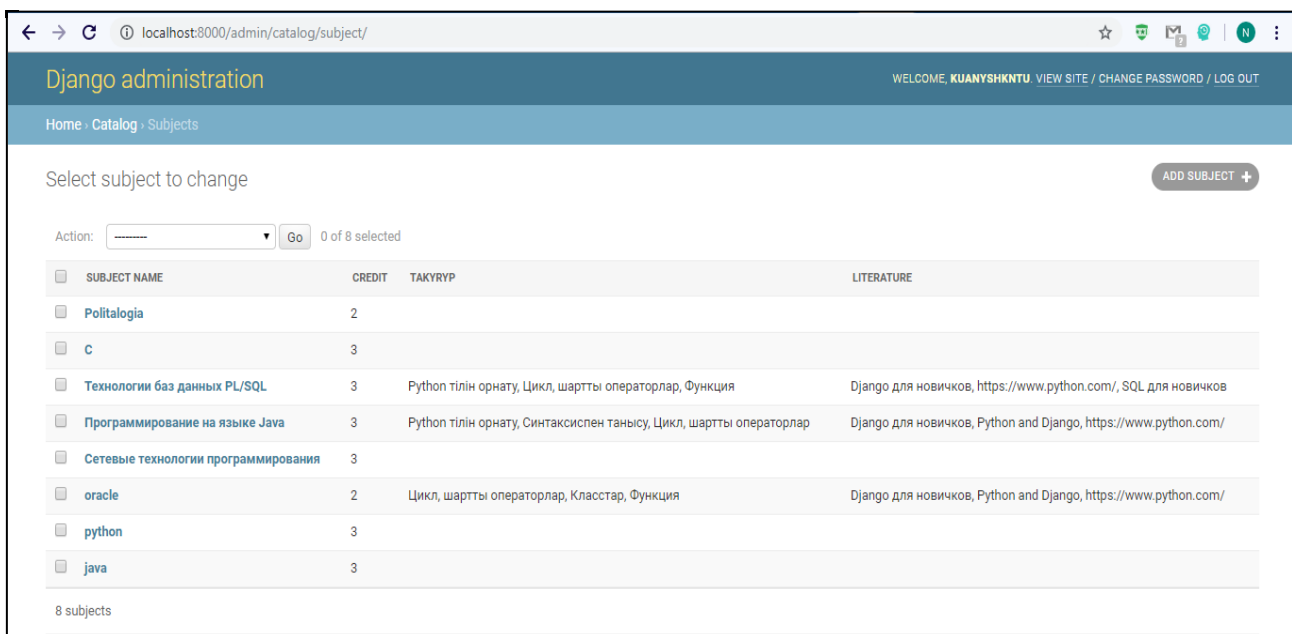
3.9-сурет – «Электронды силлабус» веб қосымшасының администратор беті арқылы қолданушыларды қосу, өшіру, өзгерту беті

Егер қолданушы өзінің құпия сөзін өзгерткісі келсе онда осы сілтеме арқылы кіріп өзгертсе болады. Бірінші жолға ескі құпия сөзді жазып, кейінгі екі жолға жаңа құпия сөзді жазамыз. Тексерістен сәтті өтілген жағдайда құпия сөз өзгертіледі. «Электронды силлабус» веб қосымшасының администратор беті арқылы құпия сөзді өзгерту беті 3.10-суретте көрсетілген.



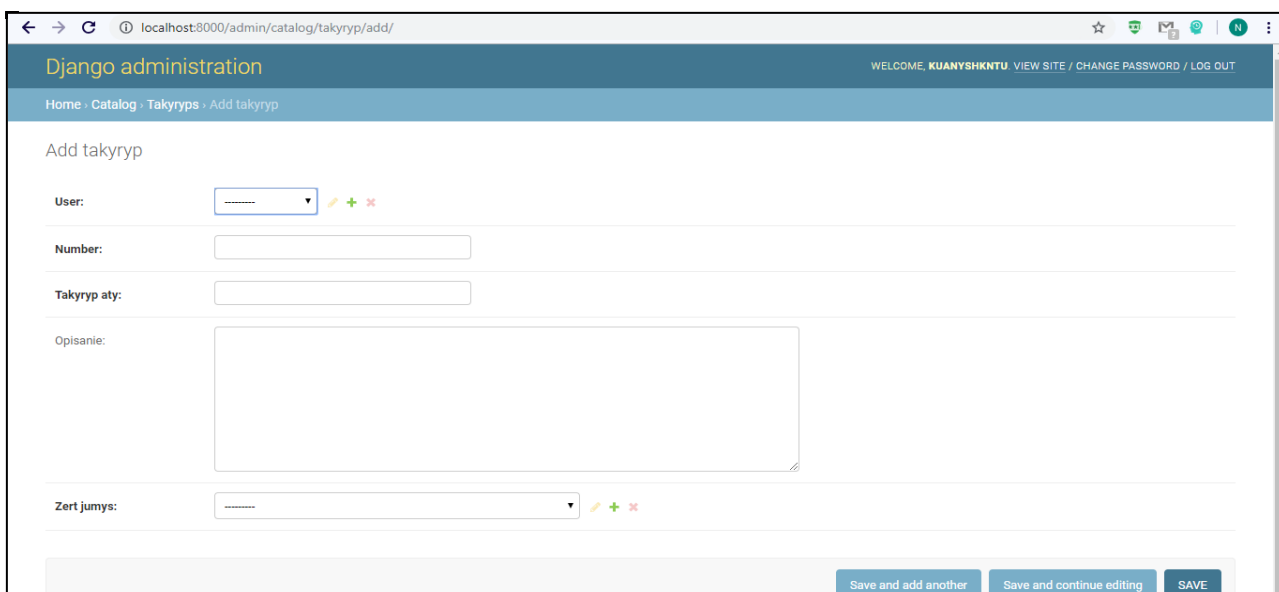
3.10-сурет – «Электронды силлабус» веб қосымшасының администратор беті арқылы құпия сөзді өзгерту беті

Төмендегі көрсетілген скриншотта жай Select for update операциясы орындалып жатыр. Сол операция арқылы мәліметтер базасына сұраныс жасап, керек жолды өзгерте алады. Ол үшін өзімізге қажет жолды таңдап, тінтуірдің сол жақ батырмасын екі рет басамыз. Сол уақытта Edit операциясы орындалады. Біз осы негізгі қолданушыға көрсетілетін атрибуттарды көрінетін жасау үшін баптау бетінен көрсетуіміз тиіс. Қазір біз көріп тұрғандай subject_name, credit, takyrup және adebiet деген атрибуттар көрсетіліп тұр. «Электронды силлабус» веб қосымшасының сабақ өзгерту беті 3.11-суретте көрсетілген.



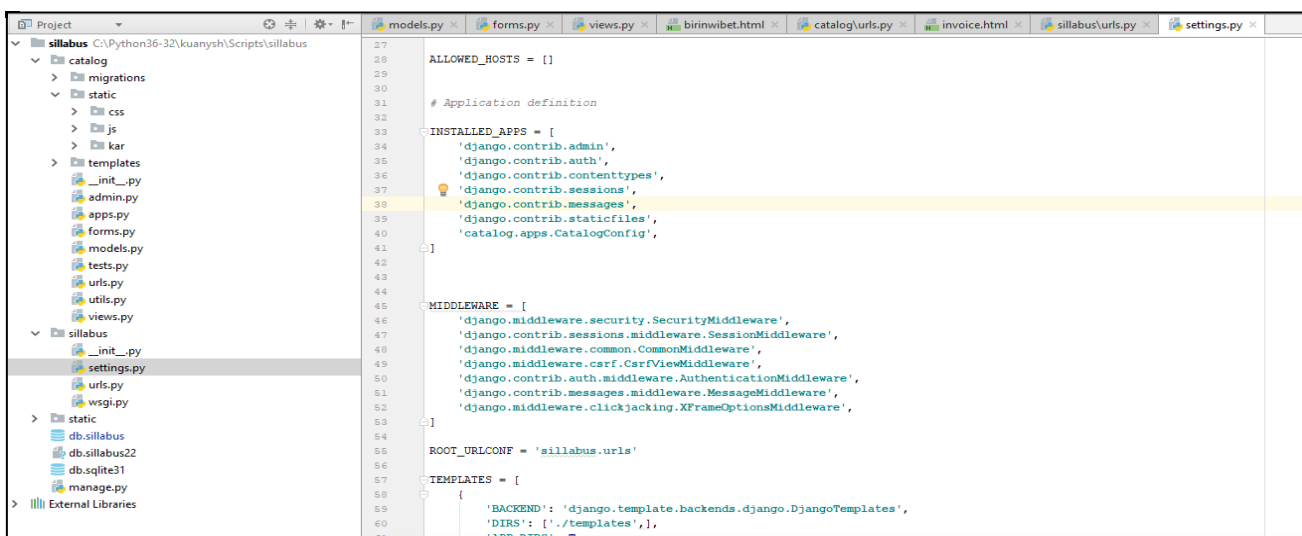
3.11-сурет – «Электронды силлабус» веб қосымшасының сабақ өзгерту беті

Келесіде көрсетілген суретте тақырып қосу беті. Бұл операция қолданушы интерфейсінде көрсетілмеген. Себебі тақырыптары супер қолданушы ғана қоса алады. «Электронды силлабус» веб қосымшасының администратор беті арқылы тақырып таңдау беті 3.12-суретте көрсетілген.



3.12-сурет – «Электронды силлабус» веб қосымшасының администратор беті арқылы тақырып таңдау беті

Қолданушы беттеріне толықтай шолу жасап болдық. Ендігі кезек кодтар болып табылады. Бұл кодтар PyCharm программасы арқылы жасалынды. Жоғарыда айтылып кеткендей MVC паттернін қолдану арқылы осы дипломдық жоба жасалынды. Төмендегі код скриншоттары Model, View және Controller(urls) беттерін көрсетеді.«Электронды силлабус» веб қосымшасының settings.py беті 3.13-суретте көрсетілген.



3.13-сурет – «Электронды силлабус» веб қосымшасының settings.py беті

Model құру бетінде класс құру арқылы мәліметтер базасында кесте құрылады. Және кестелер қатынасын осы жерде көрсетілу қажет. Модель құру бетінде класс құрып жасалынғаннан кейін осы класстарды Admin.py бетінде тіркеу қажет болады. Сол қадамнан өткеннен кейін ғана біздің кеселеріміз админ бетінде көрсетілетін болады. Және settings.py бетінде керек бауптаулар жасау міндетті. «Электронды силлабус» веб қосымшасының models.py беті 3.14-суретте көрсетілген.

```
64 number = models.CharField(max_length=100)
65 takyrp_aty = models.CharField(max_length=100)
66 opisanie = models.TextField(blank=True, null=True)
67 zert_jumys = models.ForeignKey('Zert_jumys', on_delete=models.SET_NULL, null=True)
68
69
70 def get_absolute_url(self):
71     """
72     Returns the url to access a particular author instance.
73     """
74     return reverse('subject-detail', args=[str(self.id)])
75
76 def __str__(self):
77     return '%s' % (self.takyrp_aty)
78
79
80 class Subject(models.Model):
81     user = models.ForeignKey(settings.AUTH_USER_MODEL, on_delete=models.SET_NULL, null=True)
82     subject_name = models.CharField(max_length=100)
83     credit = models.CharField(max_length=100)
84     description = models.TextField(blank=True, null=True)
85     outcome = models.TextField(blank=True, null=True)
86     takyrp = models.ManyToManyField(Takyrp, help_text="Select a genre for this book")
87     literature = models.ManyToManyField(Literature, help_text="Select a genre for this book")
88
89
90 def display_takyrp(self):
```

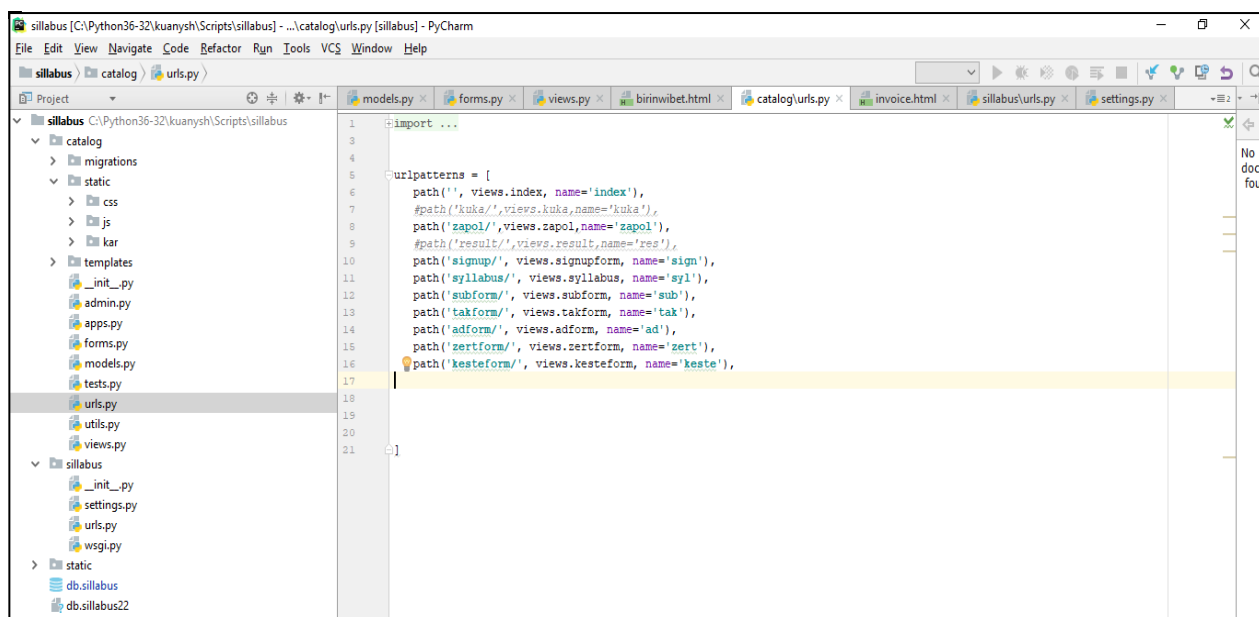
3.14-сурет – «Электронды силлабус» веб қосымшасының models.py беті

Келесі View бетінде қолданушыға қандай мәліметтерді көрсету қажет соны жасайды. Және мәліметтер қандай сұраныспен келіп жатқанын анықтап соған жауап қайтарады. Сұраныстың екі түрі болады. GET және POST. Ол Html беттің формасы арқылы келеді. «Электронды силлабус» веб қосымшасының views.py беті 3.5-суретте көрсетілген.

```
154 @login_required(login_url='/')
155 def zapol(request):
156     teachers = Teacher.objects.all()
157     user = auth.get_user(request).username
158     keste = Keste.objects.all()
159     teachers = Teacher.objects.filter(user_username=user)
160     subjects = Subject.objects.filter(user_username=user)
161     zertjumys = Zert_jumys.objects.filter(user_username=user)
162     takyrps = Takyrp.objects.filter(user_username=user)
163     literaturas = Literature.objects.filter(user_username=user)
164     rek = Subject.objects.all()
165
166     if request.method == 'POST' and request.POST.get('sub_name')!=None:
167         sub_tak = request.POST.getlist('sub_tak')
168         sub_lit = request.POST.getlist('sub_lit')
169         sub_name = request.POST.get('sub_name')
170         sub_cred = request.POST.get('sub_cred')
171         sub_desc = request.POST.get('sub_desc')
172         sub_out = request.POST.get('sub_out')
173         sub_post = request.POST.get('sub_post')
174         subjectt = Subject.objects.create()
175         #subjectt.user.create(Username=request.POST.get('uss'))
176         subjectt.user = auth.get_user(request)
177         subjectt.subject_name = sub_name
178         subjectt.credit = sub_cred
179         subjectt.description = sub_desc
180         subjectt.outcome = sub_out
181         subjectt.postrekvisit = sub_post
182         for i in sub_tak:
183             subjectt.takyrp.create(takyrp_aty=i)
```

3.15-сурет – «Электронды силлабус» веб қосымшасының views.py беті

Controller беті барлық қосымшамыздағы сілтемелерге жауап береді. Әр сілтемеге керек View бетінің функциясы шықырылады. «Электронды силлабус» веб қосымшасының urls.py беті 3.16-суретте көрсетілген.



```
1 import ...
2
3
4
5 urlpatterns = [
6     path('', views.index, name='index'),
7     #path('kuka/', views.kuka, name='kuka'),
8     path('zapol/', views.zapol, name='zapol'),
9     #path('result/', views.result, name='res'),
10    path('signup/', views.signupform, name='sign'),
11    path('syllabus/', views.syllabus, name='syl'),
12    path('subform/', views.subform, name='sub'),
13    path('takform/', views.takform, name='tak'),
14    path('adform/', views.adform, name='ad'),
15    path('zertform/', views.zertform, name='zert'),
16    path('kesteform/', views.kesteform, name='keste'),
17
18
19
20
21 ]
```

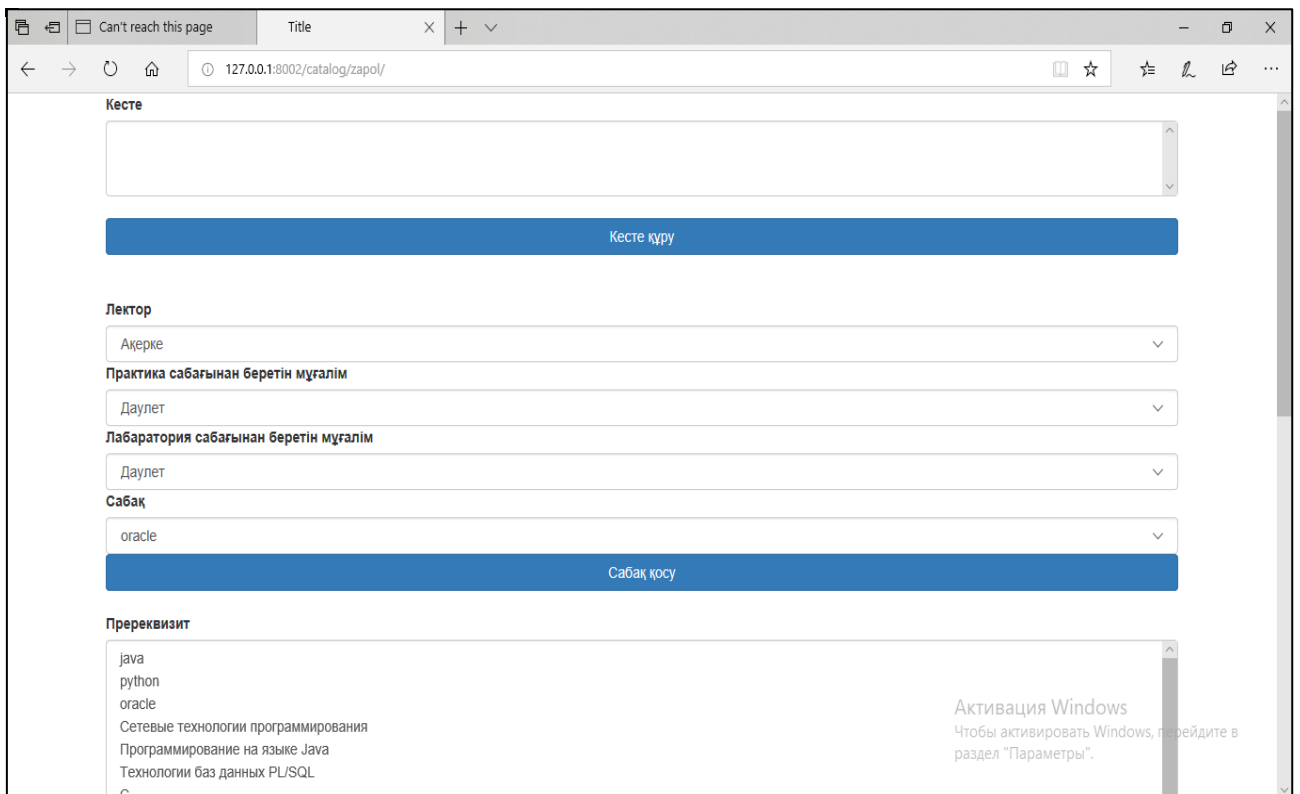
3.16-сурет – «Электронды силлабус» веб қосымшасының urls.py беті

3.2 «Электронды силлабус» веб қосымшасының сыртқы дизайны

Мен жоғарыда айтып кеткендей веб қосымшаның дизайні веб сайттың ең маңызды бөлігінің бірі болып табылады. Сондықтан менің ойымша дизайн қарапайым болуы қажет. Себебі адамдар түсініксіз дизайнмен мүлдем жұмыс жасағы келмейді.

«Электронды силлабус» веб қосымшасының дизайні өте қарапайым түрде жасалған десек болады. Ол жоғарғы оқу орны үшін жасалғандықтан қарапайым ақ және ашық көк түсті қолдандым. Кейін біздің оқу орнымыздың негізгі сайтына кіргізетін болсақ, сол сайтқа байланысты дизайн жасау қажет болады.

CSS және Bootstrap технологиясын қолданып мобильді телефондарға да дизайнды ыңғайландырып жасалынды. CSS арқылы негізгі дизайндық қаңқасын жасадым. Ал Bootstrap технологиясы арқылы оның сәндігін жетілдіріп және мобильді телефонның браузері арқылы ашқанды ыңғайлы интерфейс жасап шығарады. Егер осы технология қолданбайтын болсақ жай ғана қарапайым қисық дизайндар шығарып беретін еді. Негізі осы жобаға аса қатты әсемделген дизайн мүлдем жараспайды. Оқу орындарына қолданылатын қосымшалар қарапайым әрі түсінікті интерфейс болуы қажет. «Электронды силлабус» веб қосымшасының негізгі дизайні 3.17-суретте көрсетілген.



3.17-сурет – «Электронды силлабус» веб қосымшасының негізгі дизайні

ҚОРЫТЫНДЫ

«Электронды силлабус» веб қосымшасы – силлабусты онлайн түрде жасау қосымшасы. Осы жобаны мемлекетіміздің барлық жоғары оқу орындары қолдану үшін жасалған. Жобаны таңдау себебім оқу орындарына көмек көрсету мақсатында. Уақыт және өте аз орын алатын қосымша болып табылады.

Дипломдық жобаны жазу үшін көптеген технологиялар қолданылды. Яғни олар жаңа технологиялар десек те болады. Кейінгі технологиялар қолдану арқылы осындай жақсы жоба жасалынды.

Диплом жазу барысында қолданылған технологиялар:

- Python (Django);
- JavaScript;
- Bootstrap;
- JQuery;
- CSS;
- MySQL;
- Git.

Веб сайт құру үшін қолданылатын атақты фреймворктарының барлығы MVC паттернімен жұмыс жасайды. Осы MVC паттернін толықтай ашып және қолдану арқылы дипломдық жұмыс жасалды. MVC паттернің негізгі идеясы Model, View, Controller. Осы паттернді қолдайтын Django фреймворкі іске асырды.

Электрондық силлабус қосымшасы біздің университетімізге өте қажетті қосымша болып саналады. Бұл жұмыста оқытушылардың уақытын және жұмысын қысқарту мақсатында жасалған болатын.

Мен сабақ барысында алған білімімнің негізгі бөлігі іс жүзінде талап етілетініне тағы бір мәрте көзім жетті. Сондай-ақ қойылған міндеттерді шешуде әлемдік Интернет желісі үлкен көмек көрсетті, онда қазіргі уақытта көптеген пайдалы ақпарат табуға болады, сондай-ақ іскерлік электрондық хат алмасу құралы болып табылады.

Бұл жобаны жазу маған өте жоғары тәжірибе берді. Себебі осыған дейін жинаған теориялық біліміміздің барлығын практика жүзінде қолдандық. Осы жобаны жазу барысында да көптеген әдебиеттер оқылып және интернет желісінен көптеген мәләмет алынып, тәжірибелі оқытушылардың кеңесімен жасалынды. Осы тәжірибені қолданып алдыңғы уақытта осы сияқты басқа да жоғарғы оқу орындарының, мектептердің оқу процесстерін автоматтандыруды қолға аламыз.

«Электронды силлабус» жобасы жоғарғы оқу орындарына өте үлкен пайдасы тиетініне сенімдімін. Жоғарғы оқу орнының оқытушыларының уақытын үнемдейді. Уақыт ең үлкен байлық болғандықтан осы жобаның пайдасы тиеді.

Болашақта осы жобаны ары қарай жетілдіріп тағы көптеген функциялар қосып мемлекеттік дәрежеге шығарылуы тиіс. Онда й дәрежеге жету үшін ең соңғы шығып жатқан технолониялар (Vue.js, React т.б.) қолданылып жасау қажет. Себебі қолданушы интерфейсі қолдануға ыңғайлы болады.

ПАЙДАЛАНЫЛҒАН ӘДЕБИЕТТЕР ТІЗІМІ

- 1 Васильев А.Н. Python на примерах. — СПб.: Наука и Техника, 2016. — 40 с.
- 2 Прохоренок Н. А., Дронов В. А. Python 3 и PyQt 5. Разработка приложений на Python — СПб.: БХВ-Петербург, 2016. — 82 с.
- 3 Мэтиз Э. Изучаем Python. Программирование игр, визуализация данных, веб-приложения. — СПб.: Питер, 2017. — 154 с.
- 4 Gary Bradski and Adrian Kaehler. Learning OpenCV. — 22 p.
- 5 Кофлер М. Linux. Полное руководство. — СПб.: Питер, 2011. — 52 с.
- 6 P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, 2001, pp. I-511-I-518 vol.1.
- 7 Собель М. Linux. Администрирование и системное программирование. 2-е изд. — СПб.: Питер, 2011. — 115 с.
- 8 Майк МакГрат. Программирование на Python. — Москва: Эксмо, 2015. — 85 с.
- 9 Роберт Лав. Ядро Linux: описание процесса разработки. — М. : ООО “И.Д. Вильямс”, 2013. — 120 с.
- 10 R. Lienhart and J. Maydt, "An extended set of Haar-like features for rapid object detection," Proceedings. International Conference on Image Processing, 2002, pp. I-900-I-903 vol.1.
- 11 Мюллер А., Гвидо С. Введение в машинное обучение с помощью Python. — 67 с.
- 12 Jan Erik Solem. Programming Computer Vision with Python. — 202 p.
- 13 Рашка С. Python и машинное обучение. — М.: ДМК Пресс, 2017. — 75 с.
- 14 Педро Домингос. Верховный алгоритм: как машинное обучение изменит наш мир. — М. : Манн, Иванов и Фербер, 2016. — 143 с.
- 15 Уорд Б. Внутреннее устройство Linux. — СПб.: Питер, 2016. — 96 с.
- 16 Фленов М. Е. Linux глазами хакера: 3-е изд., перераб. и доп. — СПб.: БХВ-Петербург, 2010. — 124 с.
- 17 Dalal N., Triggs B. Histograms of Oriented Gradients for Human Detection. — IEEE Computer Society Conference on Computer Vision and Pattern Recognition. — San Diego, CA, USA — 2005.
- 18 Гудфеллоу Я., Бенджио И., Курвилль А. Глубокое обучение / пер. с англ. А. А. Слинкина. — 2-е изд., испр. — М.: ДМК Пресс, 2018. — 62 с.
- 19 OpenCV сайты: <https://opencv.org/>
- 20 Learn OpenCV сайты: <https://www.learnopencv.com/>

А Қосымшасы (міндетті)

Техникалық тапсырма

А.1 Кіріспе

Бұл жобаның негізгі мағынасы силлабусты автоматтандыру болып табылады. Осы жоба көптеген түзеулерден өткен болаты. Және көптеген технологиялар қолдану арқылы жасалған болатын.

Қолданушы басу арқылы дайын силлабус шығаруға болады. Уақытты жақсы үнемдейді. Көп дегенде 3 минут уақытты алады. Көп көңіл бөлінген сервер бөлігі, яғни формалармен жұмыс жасау. Себебі мәліметтермен көп жұмыс жасалынатын болады. Ал қолданушы бөлігі кеңінен қолданылмайды. Себебі қолданушы интерфейсі қатты қажет болмайды.

Жоба құру барысында кейінгі, яғни заманауи және тиімді фреймворктар мен технологиялар қолданылған болатын.

А.1.1 Өндеудің мақсаты мен қызметі

Осы жобаның негізгі мақсаты – уақытты өте жоғары деңгейде үнемдеу және артық іс – әрекетті азайту болып табылады. Word программасы арқылы жасау өте көп алғандықтан осы қосымша қолданысқа енгізу қажет болды. Ал ол қосымша арқылы өзі енгізіп оту қажет болған. Қосымша көмегімен тек тінтуір арқылы дайын силлабус жасап шығаруға мүмкіндік береді.

А.1.2 Қолдану саласы

Қолдану аймағы – университеттер мен білім беру орталықтары.

А.1.3 Сайт тілі

Қазақша.

А қосымшасының жалғасы

А.2 Мәтіндік ақпараттың көлемі мен құрамы

Дипломдық жоба 20 беттен және 3 бөлімнен тұрады.

А.2.1 Мәтіндік және графикалық ақпараттың электрондық түрдегі көлемі мен құрамы

Дипломдық жоба мәтіндік және графикалық ақпараттың электрондық көлемі 3,09 Мб тұрады.

А.2.2 Сайт аудиториясының шамаланған көлемі

Сайт аудиториясы негізінен 21-65 жас аралығындағы қолданушылар.

А.2.3 Сайт беттерінің саны

Сайт келесі веб парақшалардан тұруы қажет: Авторизация беті, Регистрация, Админ беті, Кесте толтыру, Негізгі таңдау беті, Дайын силлабус беті.

А.3 Сайт көлемі

«Электронды силлабус» веб қосымшасы 43,8 Мб тұрады.

А.4 Сайтты каталогтарда, рейтингтерде және топтарда тіркелу

«Электронды силлабус» веб қосымшасының GitHub жүйесіндегі сілтемесі:
<https://github.com/Kuanyshkntu/syllabus>

A.4.1 Сайтты құру мерзімі

«Why'm» веб қосымшасының құрылу мерзімі - 2017-2018.

Б қосымшасы

Бағдарлама мәтіні

1 Models. Кесте құру беті

```
from django.db import models
from django.conf import settings
class Zert_jumys(models.Model):
    user = models.ForeignKey(settings.AUTH_USER_MODEL,
on_delete=models.SET_NULL, null=True)
    number = models.CharField(max_length=100)
    opisanie = models.CharField(max_length=100,default='SOME STRING')
    def get_absolute_url(self):
        """
Returns the url to access a particular author instance.
        """
    return reverse('subject-detail', args=[str(self.id)])
    def __str__(self):
    return '%s' % (self.opisanie)
class Lit(models.Model):
    user = models.ForeignKey(settings.AUTH_USER_MODEL,
on_delete=models.SET_NULL, null=True)
    literature_name = models.CharField(max_length=100)
    author = models.CharField(max_length=100, blank=True)
    izdanie = models.CharField(max_length=100, blank=True)
    stranica = models.CharField(max_length=100, blank=True)
    god = models.DateField(null=True, blank=True)
    LOAN_STATUS = (
('H', 'Негізгі'),
('К', 'Қосымша'),
)
    typee = models.CharField(max_length=1, choices=LOAN_STATUS, blank=True,
default='m', help_text='Book availability')
    def get_absolute_url(self):
        """
Returns the url to access a particular author instance.
        """
    return reverse('subject-detail', args=[str(self.id)])
    def __str__(self):
    return '%s' % (self.literature_name
class Literature(models.Model):
    user = models.ForeignKey(settings.AUTH_USER_MODEL,
```


Б қосымшасы жалғасы

```
on_delete=models.SET_NULL, null=True)
literature_name = models.CharField(max_length=100)
author = models.CharField(max_length=100, blank=True)
izdanie = models.CharField(max_length=100, blank=True)
stranica = models.CharField(max_length=100, blank=True)
god = models.DateField(null=True, blank=True)
LOAN_STATUS = (
    ('H', 'Негізгі'),
    ('K', 'Қосымша'),
)
typee = models.CharField(max_length=1, choices=LOAN_STATUS, blank=True,
default='m', help_text='Book availability')
def get_absolute_url(self):
    """
    Returns the url to access a particular author instance.
    """
    return reverse('subject-detail', args=[str(self.id)])
def __str__(self):
    return '%s' % (self.literature_name)
class Takyryp(models.Model):
    user = models.ForeignKey(settings.AUTH_USER_MODEL,
on_delete=models.SET_NULL, null=True)
    number = models.CharField(max_length=100,unique=True)
    takyryp_aty = models.CharField(max_length=100)
    opisanie = models.TextField(blank=True, null=True)
    zert_jumys = models.ForeignKey('Zert_jumys', on_delete=models.SET_NULL,
null=True)
    def get_absolute_url(self):
        """
        Returns the url to access a particular author instance.
        """
        return reverse('subject-detail', args=[str(self.id)])
def __str__(self):
    return '%s' % (self.takyryp_aty)
class Subject(models.Model):
    user = models.ForeignKey(settings.AUTH_USER_MODEL,
on_delete=models.SET_NULL, null=True)
    subject_name = models.CharField(max_length=100)
    credit = models.CharField(max_length=100)
    description = models.TextField(blank=True, null=True)
    outcome = models.TextField(blank=True, null=True)
```

Б қосымшасы жалғасы

```
postrekvisit = models.TextField(blank=True, null=True)
takyryp = models.ManyToManyField(Takyryp, help_text="Select a genre for this
book")
literature = models.ManyToManyField(Literature, help_text="Select a genre for this
book")
def display_takyryp(self):
    """
    Creates a string for the Genre. This is required to display genre in Admin.
    """
    return ', '.join([ takyryp.takyryp_aty for takyryp in self.takyryp.all()[:3] ])
display_takyryp.short_description = 'Takyryp'
def display_literature(self):
    """
    Creates a string for the Genre. This is required to display genre in Admin.
    """
    return ', '.join([ literature.literature_name for literature in self.literature.all()[:3] ])
display_literature.short_description = 'Literature'
def get_absolute_url(self):
    """
    Returns the url to access a particular author instance.
    """
    return reverse('subject-detail', args=[str(self.id)])
def __str__(self):
    return '%s' % (self.subject_name)
class Teacher(models.Model):
    user = models.ForeignKey(settings.AUTH_USER_MODEL,
on_delete=models.SET_NULL, null=True)
    first_name = models.CharField(max_length=100)
    last_name = models.CharField(max_length=100)
    patronymic = models.CharField(max_length=100)
    email = models.EmailField()
    phone_numbers = models.CharField(max_length=100)
    account = models.CharField(max_length=100)
    subject = models.ManyToManyField(Subject, help_text="Select a genre for this
book")
def display_subject(self):
    """
    Creates a string for the Genre. This is required to display genre in Admin.
    """
    return ', '.join([ subject.subject_name for subject in self.subject.all()[:3] ])
display_subject.short_description = 'Subject'
```

Б қосымшасы жалғасы

```
def get_absolute_url(self):
    """
    Returns the url to access a particular author instance.
    """
    return reverse('teacher-detail', args=[str(self.id)])
def __str__(self):
    return '%s, %s' % (self.last_name, self.first_name)
class Keste(models.Model):
    apta = models.CharField(max_length=100)
    lekcia = models.CharField(max_length=100)
    zert = models.CharField(max_length=100)
    def get_absolute_url(self):
        """
        Returns the url to access a particular author instance.
        """
        return reverse('keste-detail', args=[str(self.id)])
    def __str__(self):
        return '%s, %s' % (self.apt, self.zert)
```

2. Views.py Көрініс беті.

```
from django.contrib.auth.decorators import login_required
from django.http import HttpResponseRedirect
from django.shortcuts import render
from django.shortcuts import render_to_response, redirect
from .models import Teacher, Subject, Literature, Takyryp, Zert_jumys, Keste, Lit
from .forms import SignupForm
from django.contrib import auth
from django.contrib import auth
def index(request):
    Keste.objects.all().delete()
    Lit.objects.all().delete()
    next = request.GET.get('next', '/')
    if request.method == 'POST':
        username = request.POST['username']
        password = request.POST['password']
        user = auth.authenticate(username=username, password=password)
        if user is not None and user.is_active:
            # Правильный пароль и пользователь "активен"
            auth.login(request, user)
            # Перенаправление на "правильную" страницу
```

Б қосымшасы жалғасы

```
return HttpResponseRedirect("/catalog/zapol/")
else:
# Отображение страницы с ошибкой

return HttpResponseRedirect("/")
return render(request, "log.html", {'redirect_to':next})
def syllabus(request):
kestes = Keste.objects.all()
techers = Teacher.objects.all()
if request.method == 'POST':
postrek = request.POST.getlist('postt')
pre = request.POST.getlist('pre')
teacher = request.POST.get('teacher')
lab = request.POST.get('lab')
prak = request.POST.get('prak')
print(teacher)
subject = request.POST.get('subject')
literature = request.POST.getlist('literature')
ll = Lit.objects.all()
postre = request.POST.get('post')
kkm = request.POST.get('kkm')
pokab = request.POST.get('pokab')
#subjectname = Subject.objects.only("credit").filter(subject_name=subject)
subject_credit = Subject.objects.get(subject_name=subject).credit
subject_outcome = Subject.objects.get(subject_name=subject).outcome
subject_description = Subject.objects.get(subject_name=subject).description
#takyryp_opisanie = Takyryp.objects.get(takyryp_aty=takyryp).opisanie
teacher_last = Teacher.objects.get(first_name=teacher).last_name
teacher_email = Teacher.objects.get(first_name=teacher).email
teacher_phone = Teacher.objects.get(first_name=teacher).phone_numbers
teacher_account = Teacher.objects.get(first_name=teacher).account
teacher_otch = Teacher.objects.get(first_name=teacher).patronymic
prak_teacher_last = Teacher.objects.get(first_name=prak).last_name
prak_teacher_email = Teacher.objects.get(first_name=prak).email
prak_teacher_phone = Teacher.objects.get(first_name=prak).phone_numbers
prak_teacher_account = Teacher.objects.get(first_name=prak).account
prak_teacher_otch = Teacher.objects.get(first_name=prak).patronymic
lab_teacher_last = Teacher.objects.get(first_name=lab).last_name
lab_teacher_email = Teacher.objects.get(first_name=lab).email
lab_teacher_phone = Teacher.objects.get(first_name=lab).phone_numbers
lab_teacher_account = Teacher.objects.get(first_name=lab).account
```

Б қосымшасы жалғасы

```
lab_teacher_otch = Teacher.objects.get(first_name=lab).patronymic
literature_author = []
literature_god = []
literature_izdanie = []
literature_stranica = []
literature_nam = []
if request.method == 'POST':
    for i in range(len(literature)):
        lit = Lit.objects.create()
        lit.literature_name =
        Literature.objects.get(literature_name=literature[i]).literature_name
        lit.author = Literature.objects.get(literature_name=literature[i]).author
        lit.izdanie = Literature.objects.get(literature_name=literature[i]).izdanie
        lit.stranica = Literature.objects.get(literature_name=literature[i]).stranica
        lit.god = Literature.objects.get(literature_name=literature[i]).god
        lit.typee = Literature.objects.get(literature_name=literature[i]).typee
        lit.save()
    context = {'kestes':kestes,'teacher': teacher, 'subject': subject, 'post': postrek, 'pre': pre,
    'outcome': subject_outcome,
    'description': subject_description,'ll': ll,
    'kkm': kkm, 'pokab': pokab, 'subject_credit': subject_credit, 'teacher_last':
    teacher_last,
    'teacher_email': teacher_email, 'teacher_phone': teacher_phone, 'teacher_account':
    teacher_account,
    'teacher_otch': teacher_otch, 'prak_teacher_last': prak_teacher_last,
    'prak_teacher_email': prak_teacher_email, 'prak_teacher_otch':
    prak_teacher_otch, 'prak': prak, 'lab_teacher_last': lab_teacher_last,
    'lab_teacher_email': lab_teacher_email, 'lab_teacher_otch': lab_teacher_otch, 'lab':
    lab}
    return render(
    request,
    'syll.html',
    context
    )
@login_required
def zapol(request):
    user = auth.get_user(request).username
    keste = Keste.objects.all()
    teachers = Teacher.objects.filter(user__username=user)
    subjects = Subject.objects.filter(user__username=user)
    zertjumyss = Zert_jumys.objects.filter(user__username=user)
```

Б қосымшасы жалғасы

```
takryyps = Takyryp.objects.filter(user__username=user)
literaturas = Literature.objects.filter(user__username=user)
rek = Subject.objects.all()
if request.method == 'POST' and request.POST.get('sub_name')!=None:
sub_tak = request.POST.getlist('sub_tak')
sub_lit = request.POST.getlist('sub_lit')
sub_name = request.POST.get('sub_name')
sub_cred= request.POST.get('sub_cred')
sub_desc = request.POST.get('sub_desc')
sub_out = request.POST.get('sub_out')
sub_post = request.POST.get('sub_post')
subjectt = Subject.objects.create()
subjectt.subject_name = sub_name
subjectt.credit = sub_cred
subjectt.description = sub_desc
subjectt.outcome = sub_out
subjectt.postrekvisit = sub_post
for i in sub_tak:
subjectt.takyryp.create(takyryp_aty=i)
for j in sub_lit:
subjectt.literature.create(literature_name=j)
subjectt.save()
if request.method == 'POST' and request.POST.get('tak_aty')!=None:
takyrypp = Takyryp.objects.create()
tak_zert = request.POST.get('tak_zert')
takyrypp.number = request.POST.get('tak_number')
takyrypp.takyryp_aty = request.POST.get('tak_aty')
takyrypp.opisanie = request.POST.get('tak_opisanie')
takyrypp.zert_jumys=Zert_jumys.objects.get(opisanie=tak_zert)
takyrypp.save()
if request.method == 'POST' and request.POST.get('ad_name')!=None:
liter = Literature.objects.create()
liter.literature_name = request.POST.get('ad_name')
liter.author = request.POST.get('ad_author')
liter.izdanie = request.POST.get('ad_izdanie')
liter.stranica = request.POST.get('ad_stranica')
liter.god = request.POST.get('ad_god')
liter.typee = request.POST.get('ad_tip')
liter.save()
if request.method == 'POST' and request.POST.get('zert_number') != None:
zert = Zert_jumys.objects.create(
```

Б қосымшасы жалғасы

```
zert.number = request.POST.get('zert_number')
zert.opisanie = request.POST.get('zert_opisanie')
zert.save()
if request.method == 'POST' and request.POST.get('apta') != None:
    apta = Keste.objects.create()
    op = Takyryp.objects.get(takyryp_aty=request.POST.get('tak')).opisanie
    apta.apta = request.POST.get('apta')
    apta.lekcia = request.POST.get('tak')+'.'+op
    apta.zert = request.POST.get('zert')
    apta.save()
    return render(request,'forma.html',{'rek':rek,'kestee':kestee,'teachers':teachers,
    'subjects': subjects, 'zertjumyss': zertjumyss, 'takyryps': takyryps,'literaturas':
    literaturas,'user':user})
def subform(request):
    takyryps = Takyryp.objects.all()
    literaturas = Literature.objects.all()
    return render(request,'subform.html',{'takyryps': takyryps,'literaturas': literaturas },)
def takform(request):
    zertjumyss = Zert_jumys.objects.all()
    return render(request,'takform.html',{'zertjumyss':zertjumyss },)
def adform(request):
    liter = Literature.objects.all()
    return render(request, 'adform.html', {'litters': liter}, )
def kesteform(request):
    kestes = Keste.objects.all()
    zert = Zert_jumys.objects.filter(user__username=auth.get_user(request).username)
    tak = Takyryp.objects.filter(user__username=auth.get_user(request).username)
    if request.method == 'POST' and request.POST.get('apta') != None:
        apta = Keste.objects.create()
        op = Takyryp.objects.get(takyryp_aty=request.POST.get('tak')).opisanie
        apta.apta = request.POST.get('apta')
        apta.lekcia = request.POST.get('tak')+'.'+op
        apta.zert = request.POST.get('zert')
        apta.save()
    return
    render(request,'kesteform.html',{'takyryps':tak,'zertjumyss':zert,'kestes':kestes })
def zertform(request):
    liter = Literature.objects.all()
    return render(request, 'zertform.html', {'litters': liter}, )
def signupform(request):
    #if form is submitted
```

Б қосымшасы жалғасы

```
if request.method == 'POST':
    #will handle the request later
    form = SignupForm(request.POST)
    #checking the form is valid or not
    if form.is_valid():
        #if valid rendering new view with values
        #the form values contains in cleaned_data dictionary
        return render(request, 'result.html', {
            'name': form.cleaned_data['name'],
            'email': form.cleaned_data['email'],
        })
    else:
        #creating a new form
        form = SignupForm()
        #returning form
        return render(request, 'signupform.html', {'form':form});
```




Төлеу түрі:	Сейлемісіз
Төлеу мерзімі:	Қазықасқа қаласы, Мейрамханалық көшесі, 10-11 аяқ
Алға:	Қазықасқа қаласы
Қарама-қарсы:	Дәулет Байдылдаев
Дата өткізілуі:	22.10.2019 12.22.10
Корреспонденциялық нөмірі № 1:	0,1%
Корреспонденциялық нөмірі № 2:	0,0%
Дәулет Байдылдаевтің корреспонденциялық нөмірі № 2:	0%
Қала мерзімі:	7 609
Қала мерзімі:	27 407
Адрес программаның бұл нұсқасы:	

ҒЫЛЫМИ ЖЕТЕКШІ
«Программалық инженерия»
кафедрасының лекторы Д.А. Баймбетов
 «17» 05 2019 жыл

Ғылыми жетекшінің пікірі

Дипломдық жоба

Оразбек Куаныш Оразбекұлы

5В060200 – Информатика

Тақырыбы: Электронды силлабус.

Диплом жобасын жасаушы Оразбек Куаныш Оразбекұлы операциялары арқылы дайын pdf форматта силлабус жасап шығару арқылы силлабусты автоматтандыру міндеттері қойылған. Веб-сайт құру барысында PyChart бағдарламалау ортасы және Python бағдарламалау тілі мен оның кітапханасы Django қолданылуы тиіс болатын.

Аталған технологиялар веб-сайтты құруда бағдарламалаушыларға таптырмас құрал және дүниежүзінде алдыңғы орында. Сондай-ақ клиенттен келген сұрауларды серверге жіберу мүмкіндіктеріне ие, сондықтан көптеген ірі компанияларда кеңінен қолданылады.


Оразбек Куаныш Оразбекұлы веб-сайтты, бұл сайтқа арналған PyChart бағдарламалау ортасы, Python бағдарламалау тілі, Bootstrap, JQuery технологияларын жақсы меңгере отырып веб-сайтты құру жолдарын жүзеге асырды. Бағдарлама құру туралы теориялық ақпараттар келтірілген. Нәтижені әрі қарай өңдеу және жақсарту талап етіледі.

Дипломдық жобаның негізгі мазмұнын құрайтын барлық нәтижелер студенттің өз бетімен алынған, ал жетекшіге мәселенің қойылуы, кеңестер мен нәтижелерді талдау ғана тиісті.

Қорытындылай келе, дипломдық жоба 5В060200 – «Информатика» мамандығының бітіру жұмыстарына қойылатын талаптарына сәйкес және 5В060200 – «Информатика» мамандығы бойынша «Техника және технологиялар бакалавры» академиялық дәрежесін тағайындауға болады және дипломдық жобаны қорғауға жіберіледі деп есептеймін.

Ғылыми жетекші

«Программалық инженерия»
кафедрасының оқытушысы, лектор

 Д.А.Баймбетов

«17» 05 2019 ж.