

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

СӘТБАЕВ УНИВЕРСИТЕТІ

Кибернетика және ақпараттық технологиялар институты

«Программалық инженерия» кафедрасы

Мукашева Дамира

«Дипломдық жобалау» бизнес үдерісін автоматтандыру

ТҮСІНІКТЕМЕЛІК ЖАЗБА

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

Алматы 2020

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

СӘТБАЕВ УНИВЕРСИТЕТІ

Кибернетика және ақпараттық технологиялар институты

«Программалық инженерия» кафедрасы

ҚОРҒАУҒА ЖІБЕРІЛДІ
Кафедра меңгерушісі
Тех. ғыл. кандидаты,
ассоциированный профессор
_____ Р.Юнусов
“_____” _____ 2020 ж.

ТҮСІНІКТЕМЕЛІК ЖАЗБА

Тақырыбы: «Дипломдық жобалау» бизнес үдерісін автоматтандыру

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

Орындаған

Мукашева Д.Б.

Ғылыми жетекші
Техн.ғыл. магистрі,
сениор-лектор
_____ Куникеев А.Д.

Алматы 2020

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

СӘТБАЕВ УНИВЕРСИТЕТІ

Кибернетика және ақпараттық технологиялар институты

«Программалық инженерия» кафедрасы

БЕКІТЕМІН

Кафедра меңгерушісі

Тех. ғыл. кандидаты,

ассоциированный профессор

Р.Юнусов

“_____” _____ 2020 ж.

ТАПСЫРМА

Білім алушы Мукашева Дамира Бекзатовна

Тақырыбы: «Дипломдық жобалау» бизнес үдерісін автоматтандыру

Университет Ректорының 20__жылғы "___" _____ №__-б бұйрығымен
бекітілген

Аяқталған жұмысты тапсыру мерзімі 2020 жылғы "02" маусым

Дипломдық жұмыстың бастапқы берілістері: Ұсынылатын дипломдық жобада
«Дипломдық жобалау» бизнес үдерістерін автоматтандыруға бағытталған
веб-қосымша жасау.

Дипломдық жұмыста қарастырылатын мәселелер тізімі

а) зерттеу бөлімі

б) технологиялық бөлімі

в) жобалау бөлімі

г) программалау бөлімі

Сызба материалдар тізімі (міндетті сызбалар дәл көрсетілуі тиіс)

Сызба материалдарының 10 бет слайдта көрсетілген.

Ұсынылатын негізгі әдебиет 7 атаудан тұрады.

Дипломдық жұмысты (жобаны) дайындау
КЕСТЕСІ

Бөлімдер атауы, қарастырылатын мәселелер тізімі	Ғылыми жетекші мен кеңесшілерге көрсету мерзімдері	Ескерту
Дипломдық жоба тақырыбына байланысты ізденіс жасау, зерттеу жүргізу.	15.11.2019	
Дипломдық жұмысты жобалау: деректер базасын құру.	10.02.2020	
Дипломдық жобаның веб- қосымшасын дайындау, іске асыру.	20.03.2020	
Дипломдық жобаны диплом алды есеп алу комиссия мүшелеріне таныстыру.	27.04.2020	
Дипломдық жобаға түсіндірме жазба жазу.	30.05.2020	

Дипломдық жұмыс (жоба) бөлімдерінің кеңесшілері мен
норма бақылаушының аяқталған жұмысқа (жобаға) қойған
қолтаңбалары

Бөлімдер атауы	Кеңесшілер, аты, әкесінің аты, тегі (ғылыми дәрежесі, атағы)	Қол қойылған күні	Қолы
Бағдарламалық бөлім	Куникеев А.Д. Техн.ғыл. магистрі, сениор- лектор		
Норма бақылау	Марғұлан Қ. Техн.ғыл. магистрі, лектор		

Ғылыми жетекші _____ Куникеев А.Д.

Тапсырманы орындауға алған білім алушы _____ Мукашева Д.Б.

Күні " _____ " _____ 20__ ж.

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

СӘТБАЕВ УНИВЕРСИТЕТІ

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

Мукашева Дамира Бекзатовна

Дипломдық жоба

Тақырыбы: «Дипломдық жобалау» бизнес үдерісін автоматтандыру

ҒЫЛЫМИ ЖЕТЕКШІНІҢ ШҚІРІ

Дипломдық жобаны жасаушы Мукашева Дамира дипломдық жобалау кезіндегі процесстерді анықтау, потенциалды қолданушылардан нақты проблемаларды білу және дипломдық жобалауды автоматтандыруға арналған қосымша жасау міндеттері қойылды.

Мамандық бойынша оқу жоспарына сай алған білімін және практикалық тәжірибесін ескере отырып, MEAN стегін таңдады. MEAN стегінің технологияларымен таныс болған, дипломдық жоба орындау барысында білімін жетілдірді.

Қосымшаның бастапқы бөлігі оқу жылының барысында университет және кафедра басшылықтарының өтініші бойынша жасалды. Осы кезде білім алушының тиянақтылығы, жауапкершілігі байқалды. Бұл жоба топтық болған соң, білім алушы, яғни, Мукашева Дамира қосымшаның мұғалім бетін, тақырыпты қосу, өшіру және оған өзгерту енгізу бөлігін, студенттен келген өтінішті қабылдау немесе кері қайтару бөлігін, студенттерге тапсырма беру бетін жасады.

Дипломдық жобаны Мукашева Дамира Бекзатовна жоспарлы түрде, қойылған талаптарға сай, оңтайлы нәтижелер алып, жұмысты қойылған талаптарға сай орындап, тақырып төңірегінде жазып шыққанын ескере отырып, Мукашева Дамира Бекзатовнаға Есептеу техникасы және бағдарламалық қамтамасыз ету – 5B070400 бакалавры академиялық дәрежесін беруге лайық деп есептеймін.

Ғылыми жетекші: «Программалық инженерия» кафедрасының техникалық ғылымдарының магистрі, сениор-лектор _____ Куникеев А.Д.

«__» _____ 2020ж

Протокол анализа Отчета подобия Научным руководителем

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

Автор: Мукашева Дамира Бекзатовна

Название: Бизнес үдерісті автоматтандыру "Дипломдық жобалау"

Координатор: Жибек Алибиева

Коэффициент подобия 1:1

Коэффициент подобия 2:1

Замена букв:0

Интервалы:0

Микропробелы:0

Белые знаки: 0

После анализа Отчета подобия констатирую следующее:

- обнаруженные в работе заимствования являются добросовестными и не обладают признаками плагиата. В связи с чем, признаю работу самостоятельной и допускаю ее к защите;
- обнаруженные в работе заимствования не обладают признаками плагиата, но их чрезмерное количество вызывает сомнения в отношении ценности работы по существу и отсутствием самостоятельности ее автора. В связи с чем, работа должна быть вновь отредактирована с целью ограничения заимствований;
- обнаруженные в работе заимствования являются недобросовестными и обладают признаками плагиата, или в ней содержатся преднамеренные искажения текста, указывающие на попытки сокрытия недобросовестных заимствований. В связи с чем, не допускаю работу к защите.

Обоснование:

.....
.....

.....

Дата

.....

Подпись Научного руководителя

АҢДАТПА

Бұл дипломдық жобада университет аясында әр жыл сайын 4 немесе 5 курс студенттерінің тапсыратын дипломдық жұмыстарын орындау процесстерін автоматтандыруға арналған жүйе құру қарастырылады. Жасалған жүйе дипломдық жұмысты орындау кезеңдерін айқындауға бағытталған. Бұл қосымша студенттің дипломдық жұмыс тақырыбын таңдаудан бастап аяқтауына дейінгі кезеңдерінен тұрады.

Берілген дипломдық жоба негізгі төрт бөлімнен тұрады. Біріншісі, зерттеу бөлімі – бұл бөлімде проблеманың өзектілігі анықталады. Екіншісі, технологиялық бөлім, мұнда қосымшаны құру барысында қолданылған технологиялар тізімі және солар туралы жалпы мәліметтер қарастырылады. Үшіншісі, жобалау бөлімі – қосымшаның дұрыс жұмыс жасауы үшін қажетті функционалдылығы мен логикасы құрастырылады. Төртіншісі, программалау бөлімі – бұнда қосымшаның бөліктеріне сипаттама беріледі.

АННОТАЦИЯ

Этот дипломный проект предусматривает создание системы для автоматизации процесса выполнения дипломных работ, представляемых студентами 4-го и 5-го курса каждый год в университете. Разработанная система направлена на определение этапов дипломной работы. Это приложение состоит из этапов от выбора темы дипломной работы до ее завершения.

Этот дипломный проект состоит из четырех основных разделов. Первый – это раздел исследований, в котором определяется актуальность проблемы. Вторым является раздел технологий, в котором представлен список технологий, используемых при создании приложения и общая информация о них. Третий раздел проектирования создает функциональность и логику, необходимые для правильной работы приложения. Четвертый раздел программирования, который описывает части приложения.

ANNOTATION

This graduation project provides for the creation of a system to automate the process of completing the thesis submitted by 4th and 5th year students every year at the university. The developed system is aimed at determining the stages of the diploma project. This application consists of stages from the selection of the topic of the diploma project to its completion.

This graduation project consists of four main sections. The first is the research section, which determines the relevance of the problem. The second is the technology section, which presents a list of technologies used to create the application and general information about them. The third is the design section creates the functionality and logic necessary for the proper operation of the application. The fourth programming section that describes the parts of the application.

МАЗМҰНЫ

Кіріспе	11
1 Зерттеу бөлімі	12
1.1 Зерттеу барысы	12
1.2 Зерттеу нәтижесі	17
2 Технологиялық бөлім	18
2.1 MEAN STACK	18
2.2 MongoDB жалпы шолу	19
2.3 Express туралы жалпы түсінік	20
2.4 Angular туралы жалпы түсінік	20
2.5 Node.js туралы жалпы мәлімет	21
3 Жобалау бөлімі	23
3.1 Веб қосымшаның деректер базасының құрылымы	23
3.2 Веб қосымшаның логикалық құрылымы	24
4 Программалау бөлімі	26
Қорытынды	30
Пайдаланылған әдебиеттер тізімі	31
А Қосымшасы	32
Б Қосымшасы	34

КІРІСПЕ

Дипломдық жоба – студенттердің білім деңгейлерін объективті басқаруға және кәсіптік қызмет түрлеріне сәйкес типтік міндеттерді шешудегі қабілеттер мен дағдыларын тексеруге арналған студенттердің қорытынды біліктілік жұмысы. Барлық студенттерге мамандығы бойынша жоба тақырыбын алдын-ала таңдау мүмкіндігі беріледі. Тақырып студенттің ғылыми және практикалық қызығушылығына байланысты анықталады. Студент дипломдық жобасының тақырыбын өзінің кафедрасы ұсынған тізімнен таңдай алады. Тақырыбын таңдаған соң, сипаттамасымен танысып, кафедра меңгерушісінің атына өтініш жібереді. Кафедра меңгерушісімен бекітілгеннен кейін, студент жетекшісімен дипломдық жобасын орындауға кіріседі.

Дипломдық жобаның мақсаты – дипломдық жобалау барысындағы барлық процесстерді қашықтықта жүргізе алатындай веб қосымша құру.

Дипломдық жобаның міндеттері:

- дипломдық жобалау кезіндегі процесстерді анықтау;
- потенциалды қолданушылардан нақты проблемаларды анықтап алу;
- анықталған проблемаларды шешетін веб қосымша әзірлеу.

1 Зерттеу бөлімі

1.1 Зерттеу барысы

Біз Satbayev University мұғалімдері мен 4 курс студенттері арасында сауалнама жүргіздік. Сауалнаманың мақсаты – дипломдық жұмысты орындау барысындағы процесстерді автоматтандыруға деген көзқарасын анықтау.

Сауалнамаға қатысқан мұғалімдердің жауабы бойынша келесідей мәселелер анықталды:

– мұғалімдер көбінесе дипломдық жұмысты талқылауға уақыт бөле алмайды, 1.1.1-суретте көрсетілген;



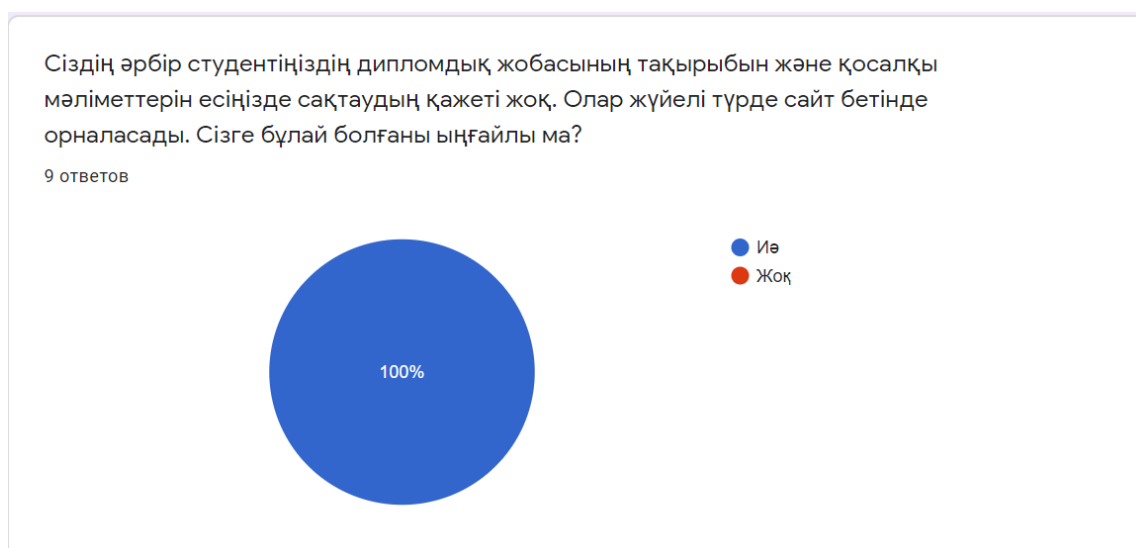
1.1.1-сурет – Сауалнама сұрағы және жауаптар

– мұғалімдердің басым бөлігіне қашықтықтан, тікелей жолығусыз студенттің жасаған жұмысын қадағалап отыру ыңғайлы, 1.1.2-суретте көрсетілген;



1.1.2-сурет – Сауалнама сұрағы және жауаптар

– мұғалімдерге әрбір студентінің дипломдық жұмысының тақырыбы және басқа да мәліметтері бір жерде сақталып тұрғаны ыңғайлы, 1.1.3-суретте көрсетілген;



1.1.3-сурет – Сауалнама сұрағы және жауаптар

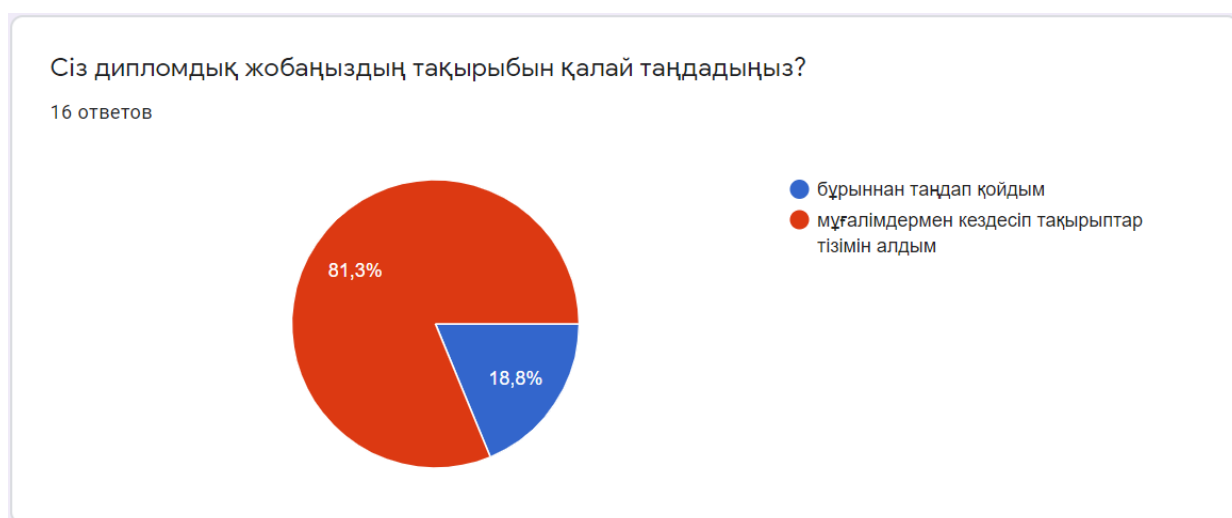
– мұғалімдерге әр студентімен хат алмасу арқылы байланыста болу ыңғайлы, 1.1.4-суретте көрсетілген;



1.1.4-сурет – Сауалнама сұрағы және жауаптар

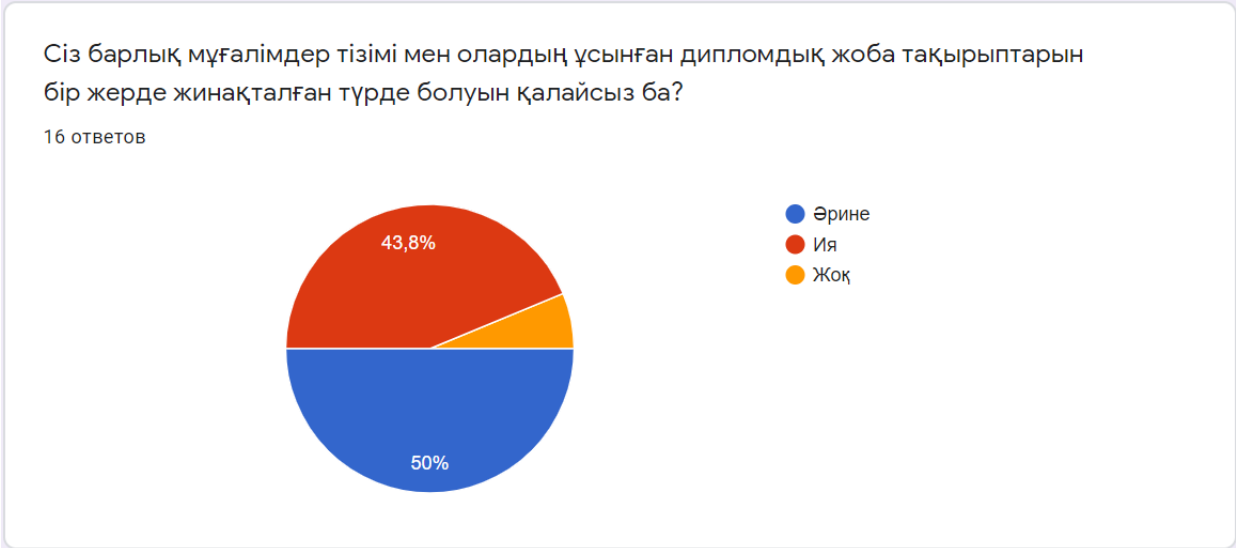
Сауалнамаға қатысқан студенттер жауабына сәйкес келесідей мәселелер анықталды:

– студенттер дипломдық жобалардың тақырыптары мен жетекші болатын мұғалімдерді білу үшін әр мұғаліммен кездесу керек болды, 1.1.5-суретте көрсетілген;



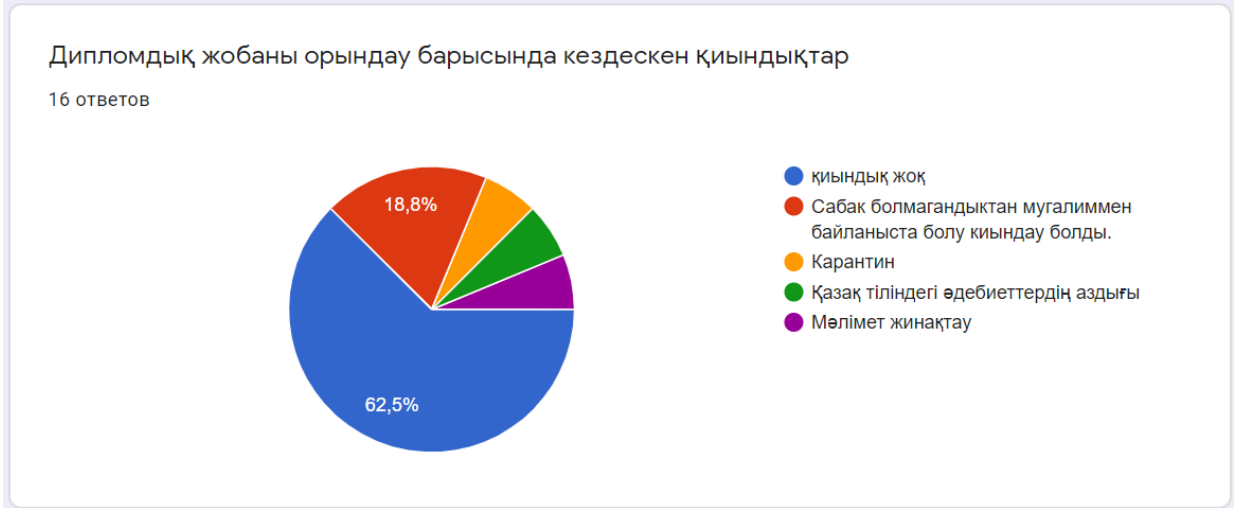
1.1.5-сурет – Сауалнама сұрағы және жауаптар

– студенттерге жетекші болатын мұғалімдер мен олардың тақырыптарын бір жерде жинақтап көру ыңғайлы, 1.1.6-суретте көрсетілген;



1.1.6-сурет – Сауалнама сұрағы және жауаптар

– дипломдық жұмысты орындау барысында студенттердің аз бөлігі кейбір қиындықтарға кезікті, 1.1.7-суретте көрсетілген;

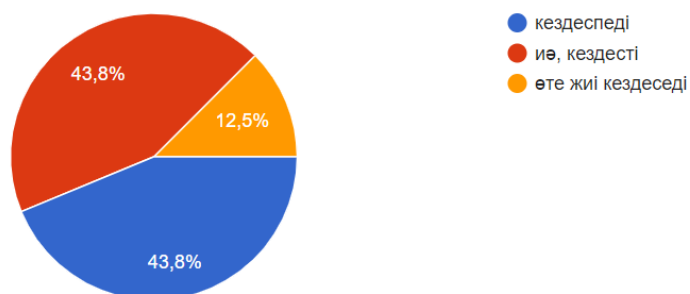


1.1.7-сурет – Сауалнама сұрағы және жауаптар

– дипломдық жұмыс бойынша сұрақтарды талқылауға студенттер мен жетекші мұғалімнің уақыттары көбінесе сәйкес келмейді, 1.1.8-суретте көрсетілген;

Жоба бойынша сұрақтарыңызды жетекшіңізбен талқылау үшін уақыттарыңыздың сәйкес келмеген кездері жиі кездесті ма?

16 ответов

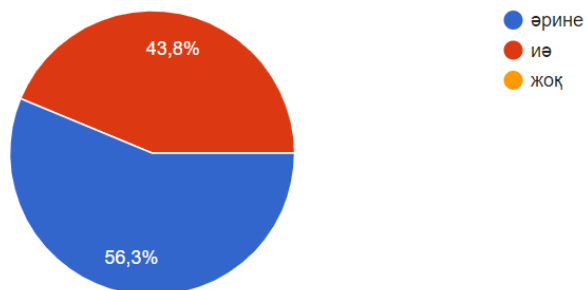


1.1.8-сурет – Сауалнама сұрағы және жауаптар

– студенттерге өзіне берілген тапсырмаларды бір жерде жинақтап көрген ыңғайлы, 1.1.9-суретте көрсетілген;

Сіздің дипломдық жобаңызға қатысты тапсырмаларды бір жерде жүйеленген түрде көруді қалайсыз ба?

16 ответов

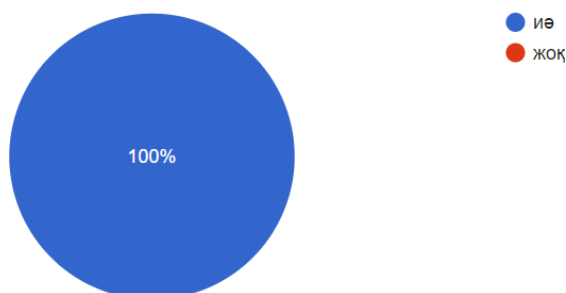


1.1.9-сурет – Сауалнама сұрағы және жауаптар

– студенттерге жетекші мұғалімімен хат алмасу арқылы байланыста болу ыңғайлы, 1.1.10-суретте көрсетілген.

Сіз жетекшіңізбен тәулік бойы байланысты болып, хат алмаса аласыз. Бұлай болғаны сізге ұнайтын ба еді?

16 ответов



1.1.10-сурет – Сауалнама сұрағы және жауаптар

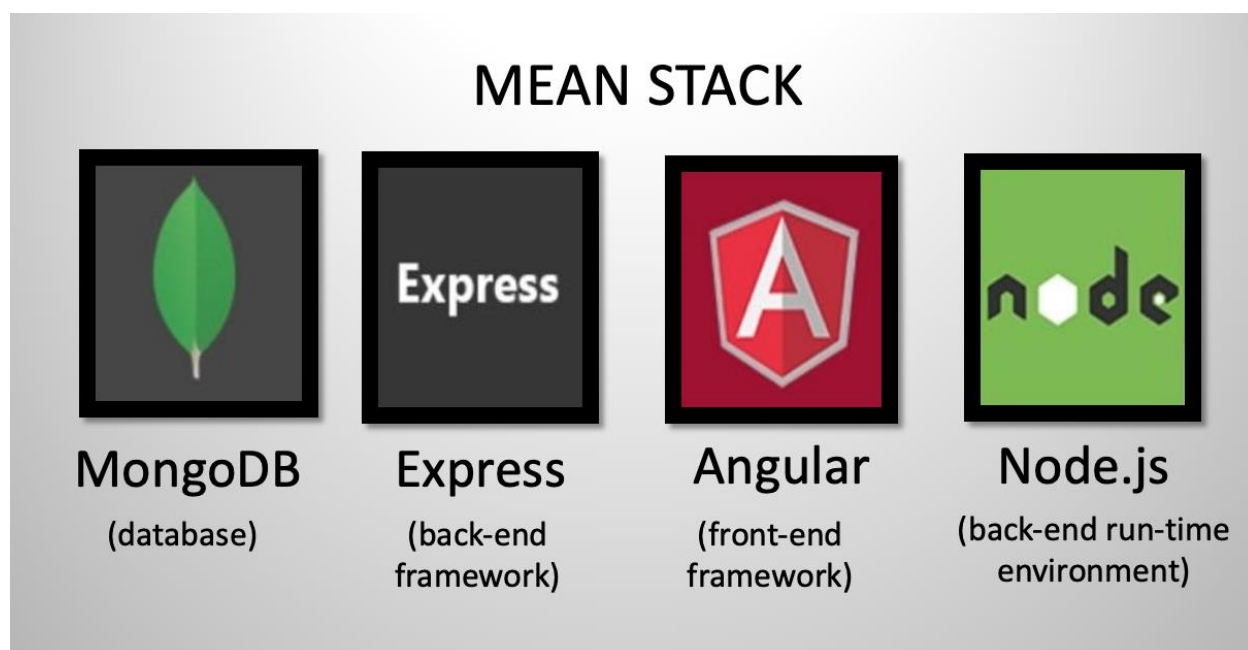
1.2 Зерттеу нәтижесі

Осы сауалнаманың көмегімен біз мұғалімдер мен студенттер қалауын анықтадық. Нәтижесінде студент пен мұғалімнің арасындағы байланысты веб қосымша арқылы жүзеге асыру ыңғайлы болатындығы расталды. Осылайша, веб қосымша арқылы біз уақыт бөле алмаушылық және қашықтықтан студенттің жұмысын бақылай алу секілді мәселелерін шешеміз. Сонымен қатар, студенттер үшін барлық мұғалімдердің тақырыптары бір қосымшада жинақталып тұруы, олардың уақытын үнемдейді.

2 Технологиялық бөлім

2.1 MEAN STACK

Біз қазір цифрлық заманда өмір сүріп жатырмыз және қазіргі уақытта заманауи технологиялардың мүмкіндігі зор. Технологиялардың дамуына байланысты веб қосымшалар жетілдіріле түсті, оларға деген сұраныс және талаптар күшейді. Сондықтан веб қосымшаны құруды жоспарлаған кезде, веб қосымша орындайтын барлық функцияларын дұрыс жүзеге асыра алатын бағдарламалар немесе технологиялар қажет. Қазіргі кезде мұндай технологиялар өте көп, сол себептен веб немесе басқа қосымшалардың әзірлушілері қосымшаны дамыту және жақсы нәтижелерге қол жеткізу үшін әртүрлі технологиялардың комбинацияларын қолданады. Бұндай комбинациялардың арасындағы ең кеңінен танымалы MEAN стегі. Бұл дипломдық жобада осы стек қолданылды. MEAN стегімен веб қосымша құру JavaScript тілін қолдану арқылы жүзеге асады [1].



2.1.1-сурет – MEAN STACK

MEAN стек бұл қысқартылған сөз, мұндағы әрбір әріп стекте қолданылатын технологияларды білдіреді. Олар MongoDB, Express, Angular және Node.js. Әр компоненттің веб қосымшаны құру кезінде атқаратын белгілі бір қызметі бар.

2.2 MongoDB жалпы шолу

MySQL, Oracle және тағы басқа көптеген деректер базалары бар. Бұл деректер базасы «реляциялық деректер қоры» деп аталады. Реляциялық деректер базасы кестелерден тұрады және олар бастапқы кілт, сыртқы кілт сияқты шектеулерді қолдана отырып, бір-бірімен байланысты. Реляциялық деректер базасы Big Data / Үлкен деректерді сақтауда тиімді емес. Үлкен деректер - бұл дәстүрлі сақтау құрылғыларын немесе реляциялық дерекқорларды пайдалану арқылы сақтау қиын деректердің үлкен жиынтығы. NoSQL реляциялық емес мәліметтер базасын білдіреді және Big Data-мен жұмыс істей алады. Сонымен қатар, NoSQL дерекқорларын басқару оңай [7].

MongoDB – бұл реляциялық емес, құжатқа негізделген деректер қорын басқару жүйесі. MongoDB-да барлық мәліметтер құжат ретінде сақталады, 2.2-сурет. Құжат дегеніміз – бұл кілттер мен мәндердің кез-келген санын қамтитын мәліметтер құрылымы. Мысалы, студент құжатында оның аты-жөні, оқу жылы, мамандығы және оның мәні осы құжатта кілт-жұп ретінде сақталады. Бірдей құрылымы бар құжаттар коллекцияға жинақталады. Бұл коллекциялар реляциялық деректер қорындағы кестелер болып табылады [7].

Схемалардың жетіспеушілігі – SQL дерекқоры мен NoSQL арасындағы маңызды айырмашылықтардың бірі. NoSQL-де біз оның алдын-ала анықталған схемасы жоқ деп айтамыз - құжатта BSON форматындағы кез-келген нәрсе болуы мүмкін. Нақты болу үшін кез-келген деректер құрылымы дерекқорға сызбасыз сәйкес келеді, сондықтан құрылымдалмаған деректерді сақтауға жарамды. Бұл әзірлеуші үшін өмірді жеңілдетеді, өйткені реляциялық деректер базасында бекітілген схема бар. Бірақ MongoDB-де кез-келген қасиеттер жиынтығымен кез-келген деректер құрылымын қоя аламыз. MongoDB схемалық болмаса да, жобалау кезінде біз сол құрылымдалған құжаттарды логикалық түрде коллекцияға орналастырамыз, сондықтан ол анық емес схеманы қолдайды [3].

Терминология MySQL и MongoDB

MySQL	MongoDB
Таблица	Коллекция
Строка	Документ
Столбец	Поле
Вторичный индекс	Вторичный индекс
JOIN-ы	Вложенные документы, \$lookup & \$graphLookup
GROUP_BY	Aggregation Pipeline

2.2.1-сурет – MySQL және MongoDB атауларын салыстыру

2.3 Express туралы жалпы түсінік

Express – бұл Javascript тілінде жазылған, тиімді қосымшалар мен API жасау үшін қажетті функциялар қабатын жүзеге асыратын Node.js ішінде жұмыс істейтін танымал фреймворк. Оны қолдану кодтың жазылуын едәуір азайтады, яғни қосымшаны жасауға кететін уақыт азаяды. Express HTTP сұрауларын өңдеуге дайын функциялары бар және сол HTTP әдістерінің өзіндік функциялары бар болғандықтан бұл әсіресе REST API жасау кезінде ыңғайлы [4].

Express негізгі белгілері :

- сұранысты бағыттаудың икемді жүйесі;
- қайта бағыттау;
- динамикалық көріністер;
- мазмұнды нақтылау;
- орындауға көңіл бөлу;
- көріністерді өңдеу және жартылай шаблондарды қолдау;
- ортаға негізделген конфигурацияны қолдау;
- API кәші.

2.4 Angular туралы жалпы түсінік

Angular 7 – веб қосымшаның клиенттік бөлігін, интерфейсін жасауға арналған фреймворк. Бір беттік қосымшалар жасауға негізделген.

SPA немесе бір беттік қосымша дегеніміз – веб бетті қажетті кодымен бірге жүктеу арқылы жұмыс істейтін бір парақты веб бағдарлама. Барлық CSS сілтемелері мен барлық қажетті сценарийлері бар жоба парағы «Веб-бет» деп аталады делік. Негізгі беті бар файл әдетте index.html деп аталады. Пайдаланушыларға осы ресурс ішінде ауысуға арналған беттер «модульдер» деп аталады. Бір параққа арналған қосымшалар болған жағдайда, бүкіл бетті жаңартудың орнына, мазмұнның қажетті бөлігі ғана жаңартылады. Бұл веб-сайттың жылдамдығын едәуір жақсартады. Бағдарламаның қызмет ету мерзімі ішінде HTML немесе сценарийлер немесе CSS сияқты көптеген ресурстар тек бір рет жүктеледі. Бір парақ қосымшасында жергілікті деректерді кәштеуге және тиімді пайдалануға болады. SPA серверге бір сұраныс жібергеннен кейін алынған барлық деректерді сақтайды. Ол тіпті деректерді пайдалану арқылы офлайн режимінде жұмыс істей алады. Егер пайдаланушы қазіргі уақытта нашар байланыста болса, нақты байланыс болған кезде сервер жергілікті деректермен синхрондалуы мүмкін. SPA қосымшаларының басты артықшылығы олардың әмбебаптылығы. Олар дербес компьютерлерде де, мобильді құрылғыларда да жақсы жұмыс істейді. Планшеттер, смартфондар және тіпті қарапайым ұялы телефондар SPA негізінде жасалған жобаларда қиындықсыз жұмыс істейді.

Сондықтан, бір беттік қосымшалардың бірінші артықшылығы құрылғыларды кең қамту деп санауға болады. Сондықтан, бір беттік қосымшаны жасау, веб-әзірлеудің стандартты әдістерін қолданумен салыстырғанда едәуір үлкен мақсатты аудиторияға сенуге мүмкіндік береді.

Angular HTML-ді шаблон тілі ретінде қолдана алады, сондай-ақ HTML синтаксисін кеңейтіп, қолданбаның құрамдас бөліктерін анық және дәл сипаттайды.

Angular екі жақты байланыстыру сияқты функционалдылықты қамтамасыз етеді, бұл модельдік мәліметтерді басқа жерде, шаблондарда, маршруттауда және т.б. өзгерту кезінде интерфейстің бір орнында деректерді динамикалық түрде өзгертуге мүмкіндік береді.

Angular фреймворкының артықшылықтары:

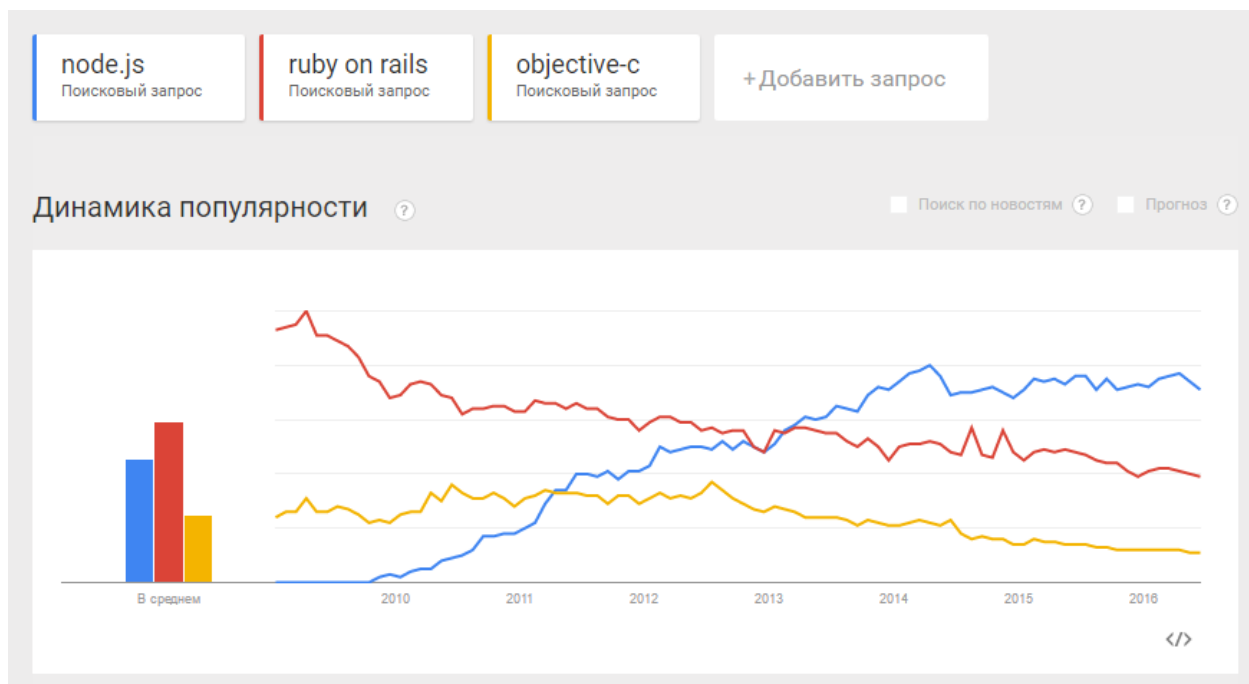
– Angular JavaScript фреймворкы болып саналғанымен, TypeScript-ке негізделген. TypeScript кодын JavaScript-те құрастыру оңай. JavaScript-пен салыстырғанда, TypeScript-тің ең жақсы артықшылығы – кодты енгізу кезіндегі қателерді анықтау, әзірлеушілерге қателерді бірден түзетуге мүмкіндік береді;

– Angular 2-ден бастап, фреймворктың барлық нұсқалары сәйкес келеді. Бұл жобаның кодтық базасын өзгертпестен Angular жаңа нұсқасына көшіруге болатындығын білдіреді. Сонымен қатар, Angular Google-мен жұмыс істейді, бұл фреймворктың тұрақтылығын одан әрі арттырады;

– Angular қосымшалар модульдер түрінде ұйымдастырылған. Бұл код әртүрлі функцияларға жауап беретін жеке модульдерден тұрады дегенді білдіреді. Модульдік жүйе тестілеуді жеңілдетеді, өйткені жеке компоненттерді сынау оңайырақ. Сонымен қатар, әртүрлі тестілеу технологиялары бар. Мысалы, Protractor ақырғы тестілеу үшін тиімді, ал Karma қондырғыны тестілеу үшін тиімді [6].

2.5 Node.js туралы жалпы мәлімет

Node.js – бұл веб-сервер сияқты жұмыс жасайтын JavaScript жұмыс ортасы. Бүгінгі таңда Node вебті дамытудың жетекші платформаларының бірі болып саналады, және көптеген веб-құралдар, сервер мен клиент осы құралмен жұмыс істейді [6].



2.5.1-сурет – Google-дің 2009 жылдан бері Node.js-қа қызығушылығы

Платформаның маңызды артықшылықтарының бірі – асинхрондылық оқиғалық тәсілмен біріктірілген. Ағындық тәсілге қарағанда, оқиғаларға бағытталған бағдарламалау кейбір сыртқы әрекеттерге негізделген, яғни, бағдарламаның орындалуы пайдаланушының әрекеттеріне байланысты. Бұл шешім енгізу-шығару көмегімен жұмыс істейтін интерактивті қосымшаларды бағдарламалауды жеңілдетеді. Бұл онлайн чаттар немесе ойын қосымшалары, түрлі-түсті интерактивті веб-қызметтер немесе калькуляторлардың барлық түрлері, рейтингтер, сауалнамалар және т.б. болуы мүмкін.

Сонымен қатар, платформаның артықшылықтары:

- платформа өзінің жеке құралдары мен мүмкіндіктерін қамтиды, меншікті кітапханалары және басқа да қызықты шешімдері бар. Бірақ негізінен JavaScript мүмкіндіктері мен синтаксисі қолданылады;
- платформа бастапқыда көптеген мүмкіндіктерге ие болды, ал жаңа нұсқаларында кітапхана толықтырылды және жетілдірілді;
- NPM пакет менеджерін қолдану Node жүйесін үнемі дамытуға мүмкіндік береді. Бүгінгі таңда ондағы ашық бастапқы пакеттер саны 500 мыңнан асып, үнемі өсіп келеді.

MEAN стегін қолдану барысында клиент пен сервер арасында ауысу оңай. Мұнда бәрі бір тілде жазылғандықтан, бұл клиент пен сервер арасында алмасуды жеңілдетеді. Мысалы, егер Node.js веб-сервер үшін қолдансақ, LAMP стеки немесе Apache қажет емес. Егер белгілі бір функция бұзылса, ол туралы бірден білінеді, себебі автоматты тестілеу мүмкіндігі бар. Бұл жобаны жетілдіруге көбірек уақыт бөлуге мүмкіндік береді. Осылайша, уақытты үнейдей отырып, сапалы қосымша жасауға болады [2].

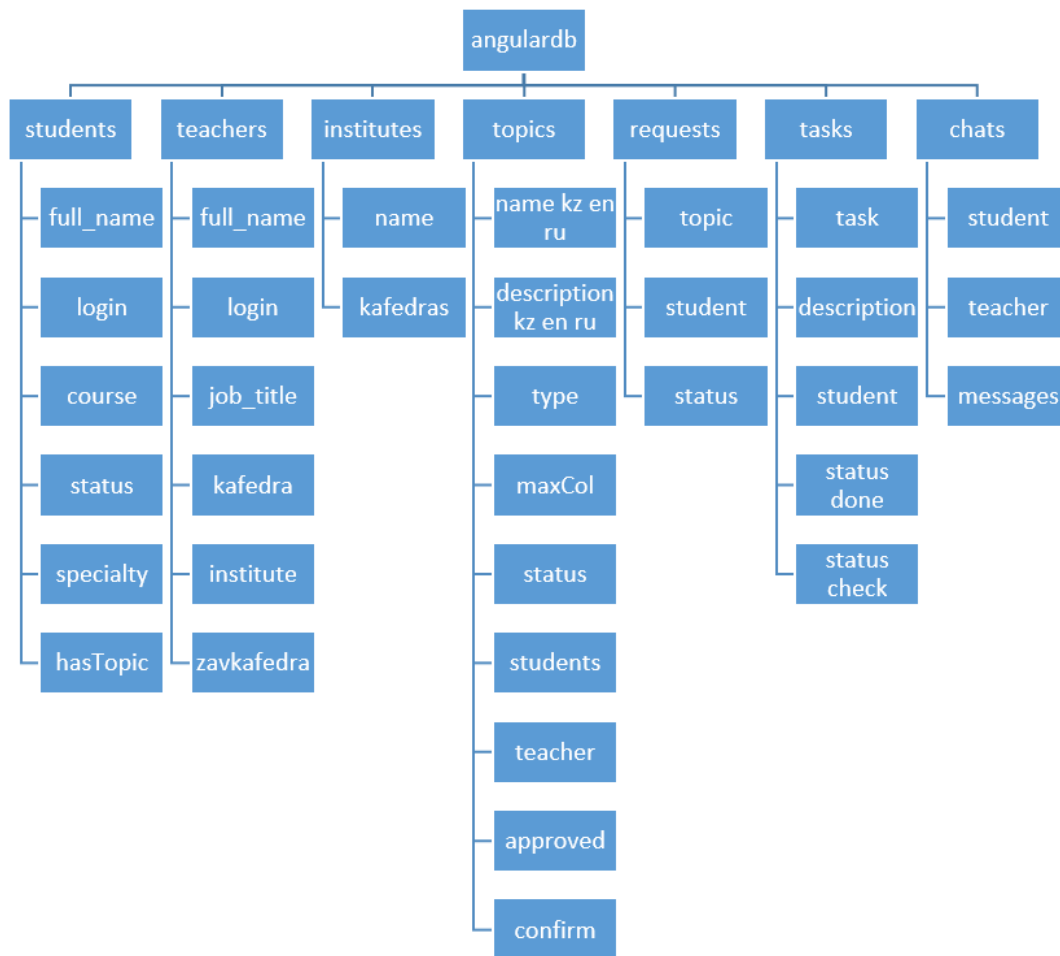
3 Жобалау бөлімі

Бұл дипломдық жұмыстың мақсаты дипломдық жобалаудың бизнес процесстерін автоматтандыру. Сондықтан жасалынған веб қосымша студент пен жетекші мұғалімнің арасында байланысты жеңілдету үшін арналған. Қосымшаның пайдаланушылары – студенттер, мұғалімдер және кафедра меңгерушілері.

3.1 Веб қосымшаның деректер базасының құрылымы

Дерекқордың атауы angulardb. Сызбасы 3.1.1-суретте көрсетілген. Ол келесі коллекциялардан тұрады:

- 1) Students коллекциясы. Бұл коллекция әр студент жайында ақпарат сақталған құжаттардан тұрады.
- 2) Teachers коллекциясы. Бұл коллекция әр мұғалім жайында ақпарат сақталған құжаттардан тұрады.
- 3) Institutes коллекциясы. Бұл коллекция әр институт жайында ақпарат сақталған құжаттардан тұрады.
- 4) Topics коллекциясы. Бұл коллекция әр мұғалімнің тақырыбы және сол тақырыпқа тіркелген студенттерді сақтайтын құжаттардан тұрады.
- 5) Requests коллекциясы. Бұл коллекция әр дипломдық жобаға жіберілген студенттердің өтінішін сақтайтын құжаттардан тұрады.
- 6) Tasks коллекциясы. Бұл коллекция әр студентке жетекші мұғалім тарапынан берілген тапсырмалардан тұратын құжаттарды сақтайды.
- 7) Chats коллекциясы. Бұл коллекция студент пен жетекші мұғалім арасындағы хаттарды сақтайды.



3.1.1-сурет – деректер қорының сызбасы

3.2 Веб қосымшаның логикалық құрылымы

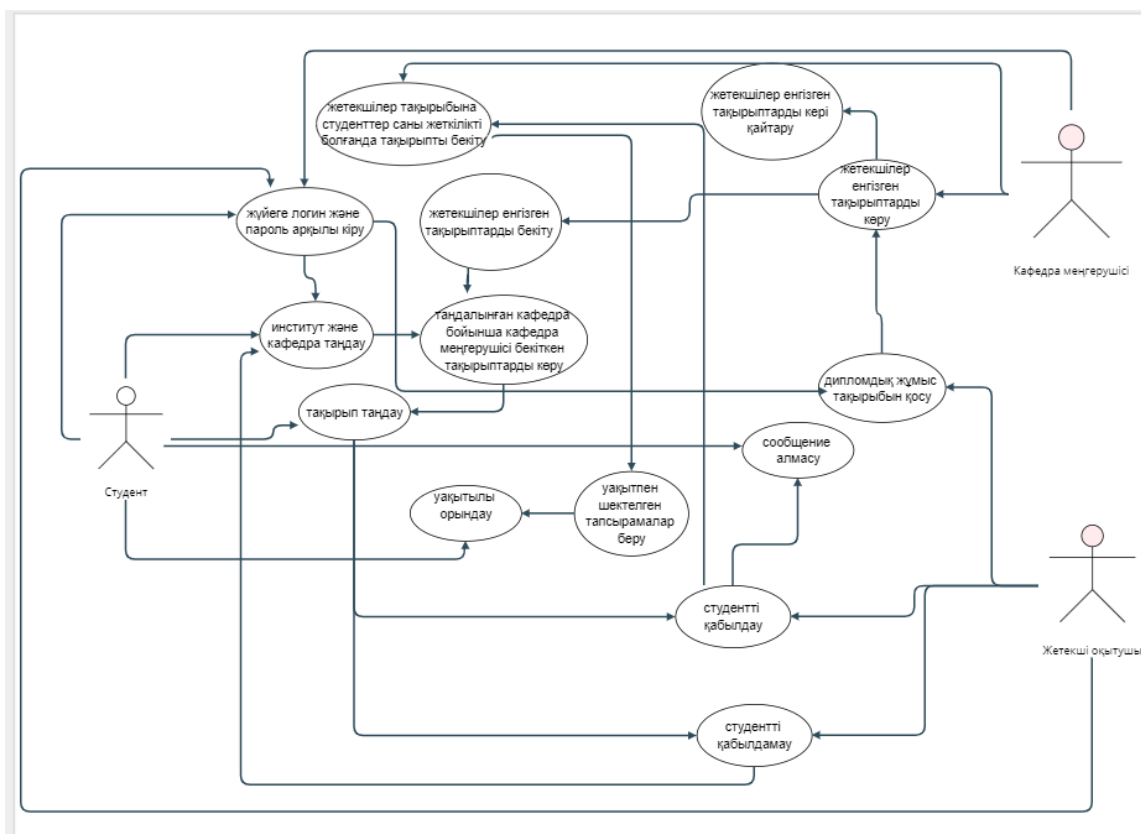
Веб қосымшада пайдаланушылардың жүзеге асыратын іс-әрекеттері 3.2.1-суретте көрсетілген.

Жетекші мұғалім жүйеге кіріп, өзінің дипломдық жұмыс тақырыптарын енгізеді. Сонымен қатар, өзіне тіркелген студенттерге тапсырма бере алады және әрбір студентімен хат алмасу мүмкіндігі бар.

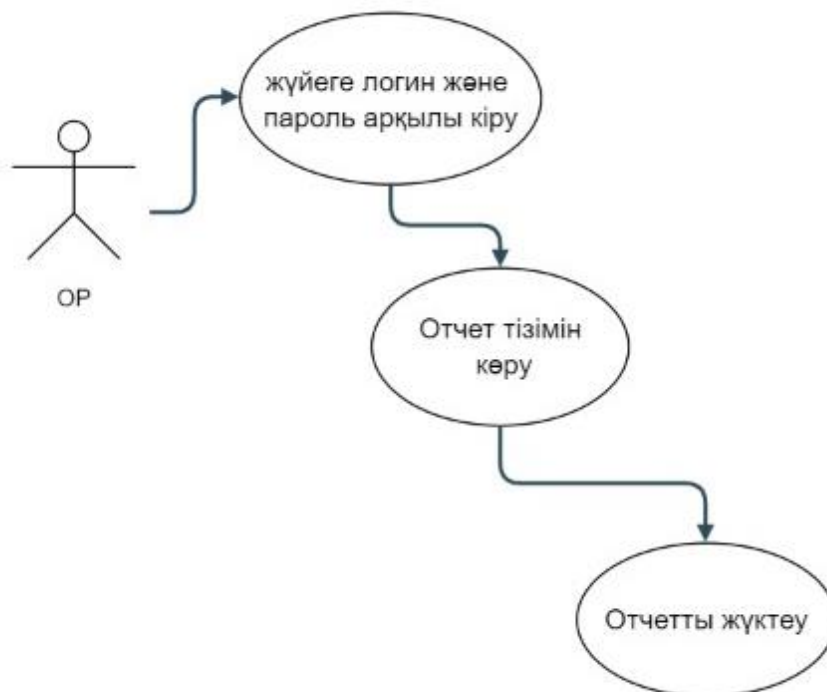
Студент жүйеге кіріп өзіне дипломдық жұмыс тақырыбын таңдайды. Жетекші мұғалім өзіне қабылдағаннан кейін, берілген тапсырмаларды орындай алады және жетекші мұғалімімен хат алмаса алады.

Кафедра меңгерушісі жетекші мұғалімдердің дипломдық жұмыс тақырыптарын бекітеді.

Офис регистратор жүйеге кіріп, отчет тізімін көріп, өзіне керек отчетты жүктей алады.



3.2.1-сурет – веб қосымшаның логикалық құрылымы

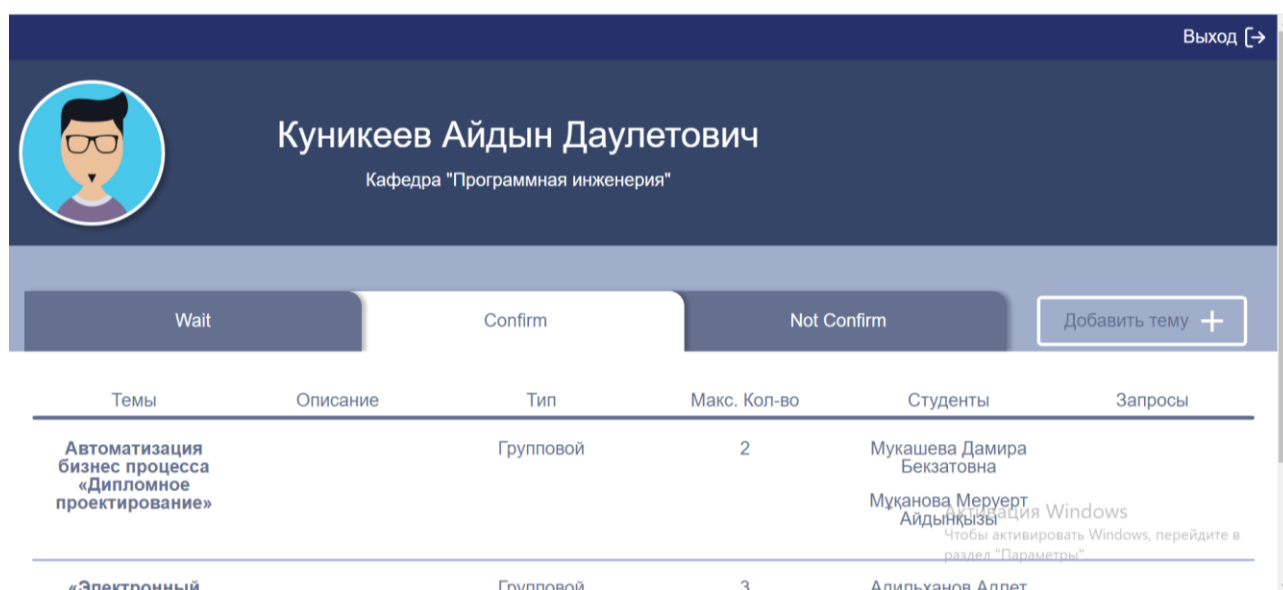


3.2.2-сурет – веб қосымшаның логикалық құрылымының жалғасы

4 Программалау бөлімі

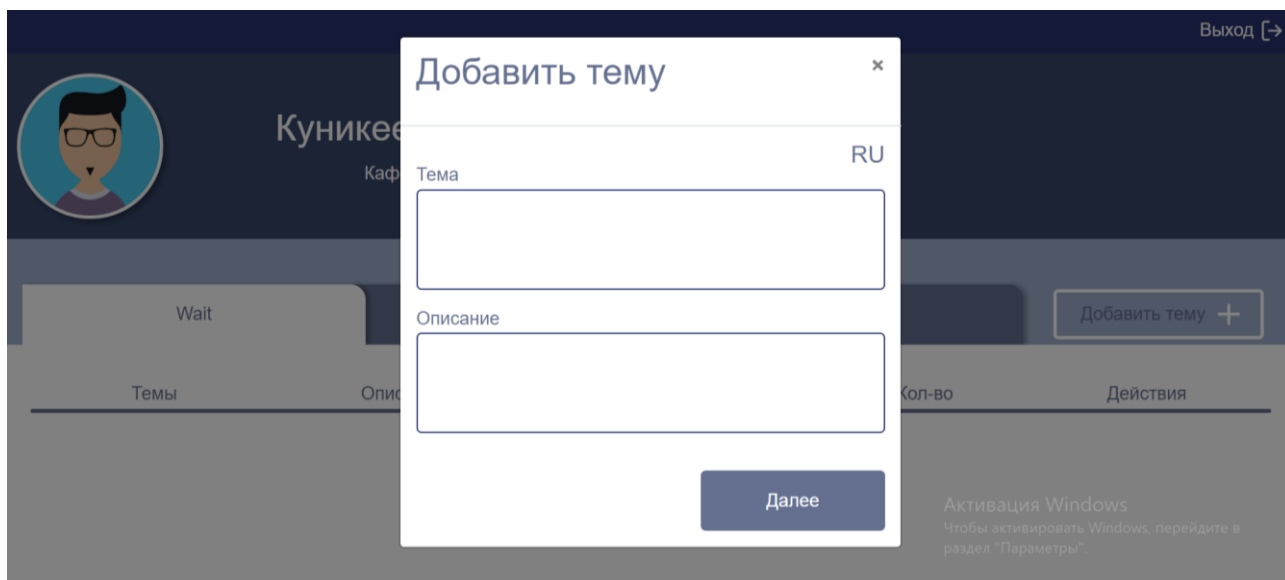
Дипломдық жұмысты автоматтандырудың веб қосымшасы келесі компоненттерден тұрады:

1) Teacher page компоненті. Интерфейсі 4.1-суретте көрсетілген. Мұғалім жүйеге кірген соң өзіне жаңа тақырып қоса алады. Жаңадан қосылған тақырыптар wait таблицасына қосылады. Кафедра меңгерушісі тақырыптарды қабылдайды немесе кері қайтарады. Қабылдаған жағдайда студенттер сол тақырыптарды көре алады. Қабылдамаған жағдайда not confirm таблицасына қосылады.



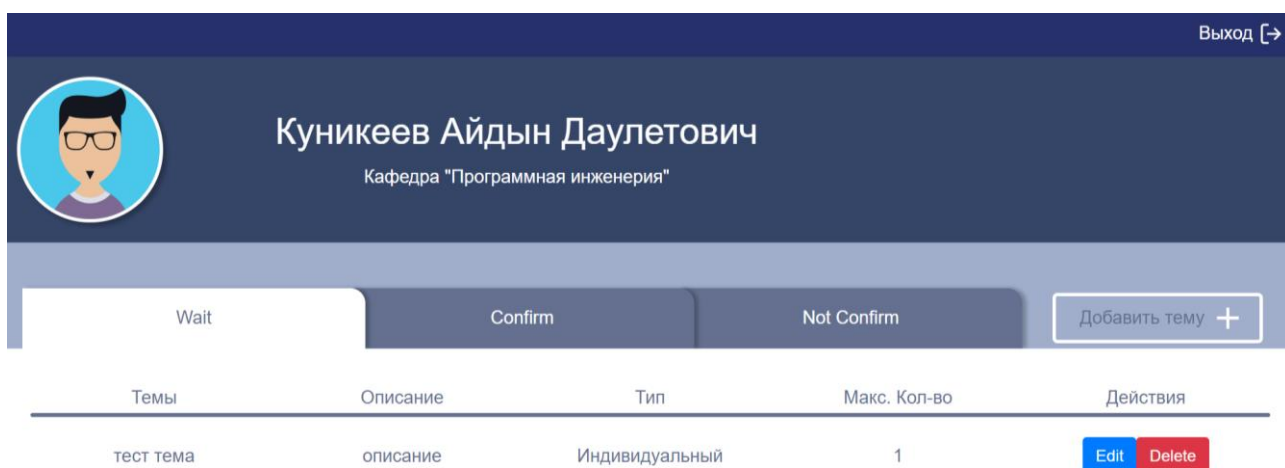
4.1-сурет – Teacher page интерфейсі

2) Мұғалімнің өзіне тақырып қосу мүмкіндігі бар, интерфейсі 4.2-суретте көрсетілген.

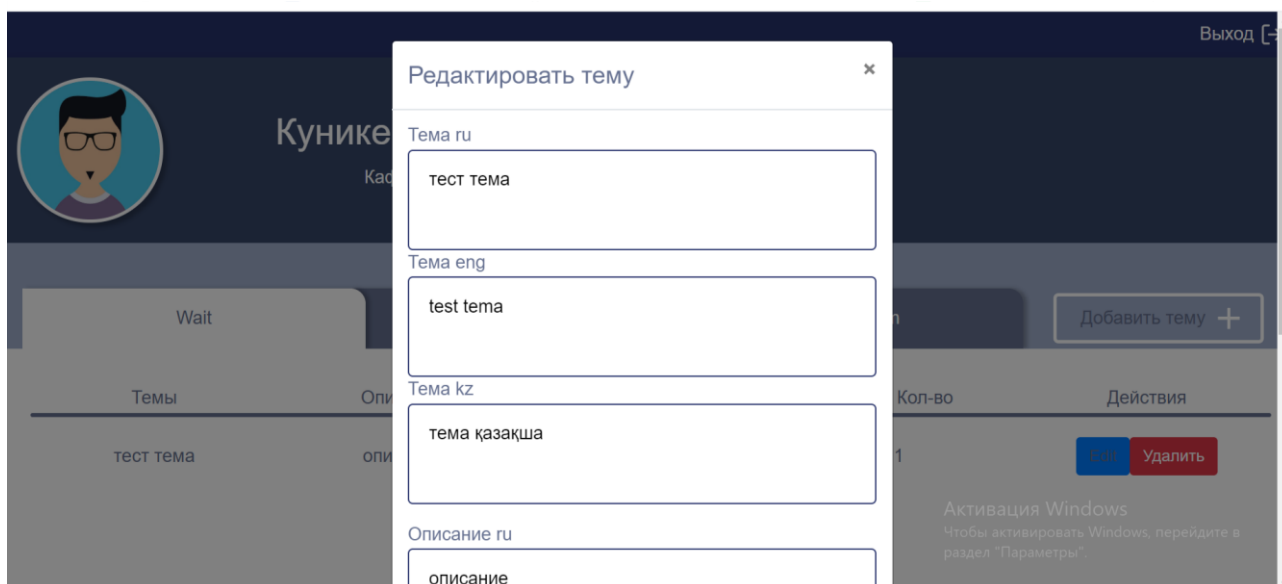


4.2-сурет – Teacher page тақырып қосуы интерфейсі

3) Жаңадан қосылған тақырыптарды мұғалім кафедра меңгерушісі бекіткенге дейін өзгерте алады немесе өшіре алады, интерфейсі 4.3-суретте көрсетілген.



4.3-сурет – Teacher page тақырыпты өзгерту немесе өшіру мүмкіндігі



4.4-сурет – Teacher page тақырыпты өзгерту интерфейсі

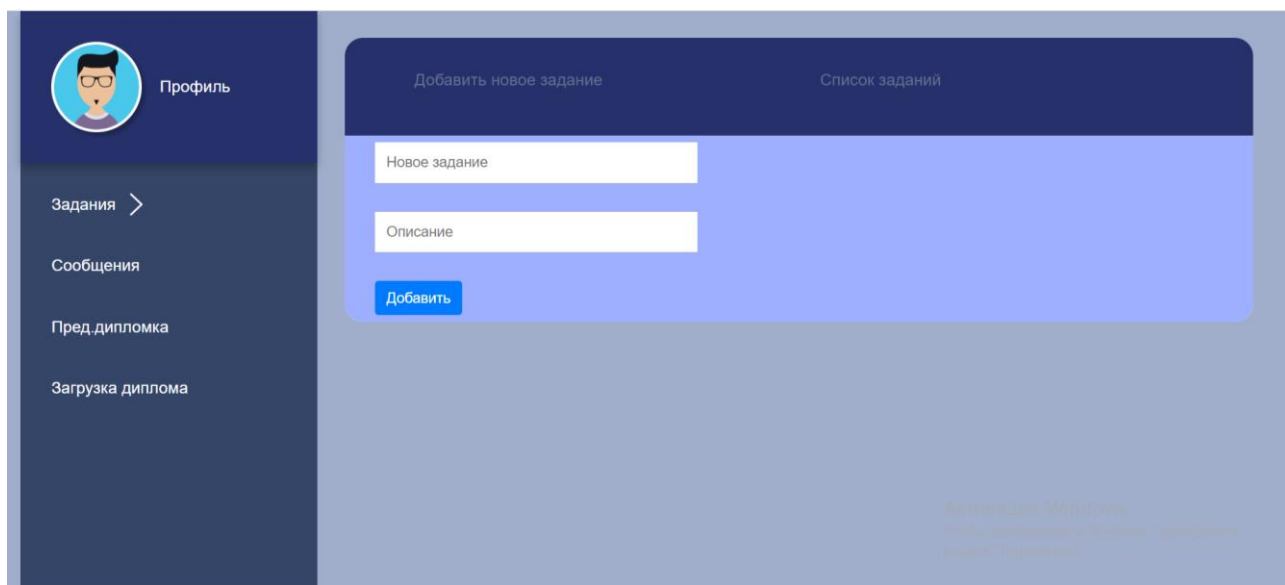
4) Студенттер жүйеге кіріп өзінің кафедрасы бойынша берілген тізімнен тақырып таңдайды. Мұғалім студенттен келген өтінішті қабылдайды немесе кері қайтарады. Кері қайтарылған жағдайда студентте басқа тақырып таңдау мүмкіндігі болады. Қабылдаған жағдайда кафедра меңгерушісінің бекітуін күтеді. Бекітілгеннен кейін жұмысқа кіріседі.

Темы	Описание	Тип	Макс. Кол-во	Студенты	Запросы
Автоматизация бизнес процесса «Дипломное проектирование»		Групповой	2	Мукашева Дамира Бекзатовна Мұқанова Меруерт Айдынкызы	
«Электронный табель» - автоматизация табель выходов		Групповой	3	Адилханов Адлет Муратович Ахетова Қарлығаш Ғалымқызы Берікқалиева Айтолқын Бекбергенқызы	
weg	df	Индивидуальный	1	Хамитов Ғабиден Ғабидұлы	
тест тема	описание	Индивидуальный	1		Абдрахман Абылай Бағдатұлы Принять Отказать

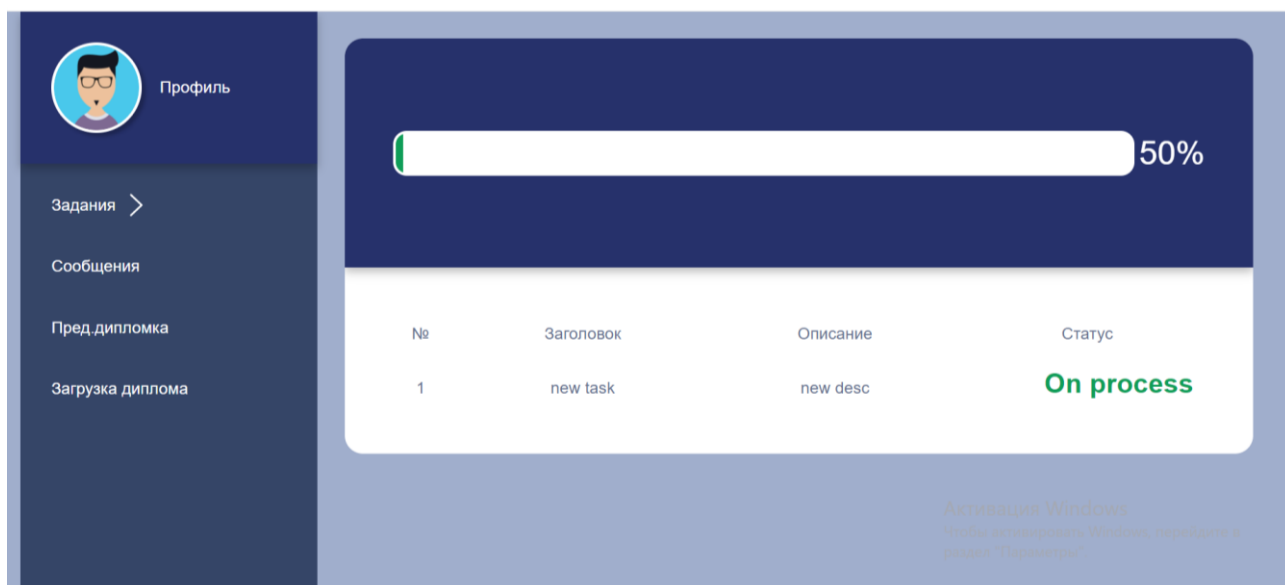
4.5-сурет – Teacher page студенттен келген өтінішті қабылдау немесе кері қайтару

5) Кафедра меңгерушісі мұғалімнің тақырыптарын студенттерімен бекіткеннен кейін, мұғалім әрбір студентіне тапсырмаларды системада белгіленген күнге дейін ғана өшіріп, қоса алады, 4.6-суретте көрсетілген.

Студенттер сол тапсырмаларды системада белгіленген күннен кейін ғана көре алады, орындауды бастайды, 4.7-суретте көрсетілген.



4.6-сурет – Мұғалім тапсырма қосатын бет интерфейсі



4.7-сурет – Тапсырмалардың студент бетінде көрсетілуі

ҚОРЫТЫНДЫ

Дипломдық жоба барысында дипломдық жобалау кезіндегі бизнес процесстерді автоматтандыруға арналған веб қосымша жасалынды. Жасалған жүйеге тек жетекшілік ететін мұғалімдер, кафедра меңгерушілер мен бітіруші студенттер ғана кіре алады. Жүйенің басты мақсаты мұғалім мен студент арасында байланыс орнатату және дипломдық жобаны орындау ыңғайлығын қамтамасыз ету болды.

Зерттеу бөлімінде университет студенттері мен мұғалімдері арасында сауалнама жүргізілді. Нәтижесінде қолданушыларға қажетті функциялар анықталды.

Технологиялық бөлімінде әртүрлі технологияларды зерттей келе, MEAN стегі таңдалынды. MEAN стек жүйесінде қолданылатын технологиялармен жұмыс істеу шеберлігіміз артып, тәжірибе жинадық.

Жобалау бөлімінде қосымшаның дұрыс жұмыс істеуі үшін керекті процесстер анықталып, қосымшаның жалпы көрінісі айқын болды. Деректер базасы жобаланды. Сонымен қатар, қолданушылардың жүйедегі іс-әрекеттері анықталды.

Программалау бөлімінде жүйенің интерфейсі және деректер қорымен байланысы, сонымен қатар жалпы жүйенің функционалы мен логикасы жасалынды.

Дипломдық жобалаудың нәтижесінде жасалған жүйені қолдану студенттер мен мұғалімдер үшін ыңғайлы болып, уақытын үнемдеуге көмектеседі деген сенімдеміз.

ПАЙДАЛАНЫЛҒАН ӘДЕБИЕТТЕР ТІЗІМІ

1 Why Choose Mean Stack For Your Web & Mobile App Development Projects? // Электрондық нұсқасы <https://medium.com/@ronak8036/why-mean-stack-ec42aa82818>

2 Want to understand the MEAN Stack quickly? Here's documentation with useful diagrams. // Электрондық нұсқасы <https://www.freecodecamp.org/news/cjn-understanding-mean-stack-through-diagrams/>

3 Лабазин, А. А. Интеграция Java-приложения с базой данных MongoDB / А. А. Лабазин. — Текст : непосредственный // Молодой ученый. — 2018. — № 24 (210). — С. 28-32. — URL: <https://moluch.ru/archive/210/51448/> (дата обращения: 16.05.2020).

4 10. Node.js модули, о которых вы должны знать: express.js // Электрондық нұсқасы <https://nodeguide.ru/doc/modules-you-should-know/express/>

5 Зачем вам нужны разработчики Angular и как их нанимать // Электрондық нұсқасы <https://senior.ua/articles/zachem-vam-nuzhny-razrabotchiki-angular-i-kak-ih-nanimat>

6 Что такое Node.js? // Электрондық нұсқасы <https://netology.ru/blog/node>

7 NoSQL мен MongoDB арасындағы айырмашылық // Электрондық нұсқасы <https://kk.bccrwp.org/compare/difference-between-nosql-and-mongodb/>

А Қосымшасы

Техникалық тапсырма

А.1.1 Дипломдық жобалау процесстерін автоматтандыруға арналған жүйеге техникалық тапсырма

Бұл техникалық тапсырма дипломдық жобалау кезіндегі студенттер мен мұғалімдер арасындағы қарым-қатынасты жүзеге асыратын қосымшаға бағытталады. Болжам бойынша бұл жүйені университеттің студенттері, мұғалімдері және офис регистратор қолданады. Дипломдық жобалаудың барлық процесстерін автоматтандыратын жүйе әр студенттің жұмысын орындау динамикасы туралы ақпарат береді. Жүйе студенттер мен мұғалімдердің уақытын үнемдейді.

А.1.2 Тағайындалуы

Бітіруші студенттер мен мұғалім арасында байланысты қашықтықтан жүргізіп, ақпаратты сақтау және өңдеуге арналған.

А.1.3 Функционалдық талаптар

Жүйе келесі функцияларды орындай алуы керек:

- авторизация (қолданушы өзінің жеке мәліметтерін енгізіп, жүйеге кіреді);
- мұғалім өзіне 3 тілде (қазақша, орысша, ағылшынша) тақырып енгізу керек және оны сақтау керек, бекітілгенге дейін өзгерту және өшіре алатын мүмкіндігі болуы керек;
- кафедра меңгерушісі өз кафедрасы бойынша мұғалімдердің тақырыптарын бекіту керек;
- бекітілген тақырыптар студентте көрінуі тиіс;
- студентте институт және кафедра таңдау мүмкіндігі болуы керек;
- студентте таңдаған кафедрасы бойынша мұғалімдердің тақырыптарын көріп, тек біреуін ғана таңдай алуы керек;
- мұғалімде студенттен түскен өтінішті қабылдай немесе кері қайтара алуы керек;
- кафедра меңгерушісі студенттер саны жеткілікті тақырыптарды бекіте алуы тиіс;

А Қосымшасының жалғасы

– студентте дипломдық жұмысты орындауды бастайтын бет ашылуы керек, бұл бетте 5 бөліктен (профиль, задания, сообщения, преддипломка, диплом) тұратын мәзір болуы керек, кірген бетте профиль беті ашық тұруы керек;

– белгілі бір уақытқа дейін мұғалім әр студентіне тапсырмалар тізімін жасап, оны растай алуы керек және белгіленген уақыт аяқталған соң, студентте тек расталған тапсырмалар тізімі болуы керек, расталмаған тапсырмаларды жүйе өшіреді;

– студент белгілі бір тапсырманы орындағаннан кейін done батырмасын басады, мұғалім ол тапсырманы тексергеннен кейін қабылдайды немесе кері қайтарады. Қабылдаған жағдайда тапсырма аяқталған болып саналады, кері қайтарылған жағдайда студент сол тапсырманы қайта орындауға кіріседі;

– мұғалім мен студент арасында хат алмасу мүмкіндігі болуы керек;

– мұғалім пред.дипломдық қорғаудан кейін студенттің бағасын жүйеге енгізе алуы керек;

– студент аяқталған жұмысын жүйеге жүктеу керек, ал мұғалім өз кезегінде оның бағасын қоюы керек.

А.1.4 Сенімділікке қойылатын талаптар

– Енгізілген ақпаратты уақытылы бақылауды және қамтамасыз ету.

– Сақталатын ақпараттың тұтастығын қамтамасыз ету

А.1.5 Техникалық жабдықтың құрамы мен параметрлеріне қойылатын талаптар

Жүйе IBM-үйлесімді дербес компьютерлер жұмыс істеуі тиіс. минималды конфигурация: процессор типі – Pentium және одан жоғары, жедел жадтау құрылғысының көлемі 512 Мб және одан жоғары.

Б Қосымшасы

Программаның мәтіні

```
<!--This is Teacher Page component html --->
<div class="body">
  <input type="hidden" id="refresh" value="no">
  <div class="page">
    <header class="header">
      <div class="header__top">
        <div class="container">
          <div class="header__top-inner">
            <a class="header__btn" *ngIf="teacher?.zavkafedra"
(click)="openPageZK()" style="color:white">
              Зав.Кафедра
            </a>
            <a class="header__exit-btn" href="#">
              Выход
            </a>
          </div>
        </div>
      </div>
    </div>
    <div class="header__bottom">
      <div class="container">
        <div class="header__bottom-inner">
          <div class="header__avatar" style="background-image:
url(../../assets/images/avatar.jpg);"></div>
          <div class="header__info">
            <div class="header__name">
              {{ teacher?.full_name }}
            </div>
            <div class="header__type">
              {{ teacher?.kafedra }}
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>
<section class="themes">
  <div class="themes__header">
    <div class="container-fluid">
```

```
<div class="themes__header-inner">
  <ul class="themes__tabs">
    <li class="themes__tab active" [class.active]="viewMode == 'tab1'"
rel="tab1" (click)="viewMode = 'tab1'">Wait</li>
    <li class="themes__tab" [class.active]="viewMode == 'tab2'" rel="tab2"
(click)="viewMode = 'tab2'">Confirm</li>
    <li class="themes__tab" [class.active]="viewMode == 'tab3'" rel="tab3"
(click)="viewMode = 'tab3'">Not Confirm</li>
  </ul>

  <a class="themes__add-btn" data-src="#modal"
(click)="openModal(teacher)" >
    Добавить тему
  </a>
</div>
</div>
</div>
<div class="themes__main" [ngSwitch]="viewMode" >
  <div class="container-fluid">
    <div class="themes__main-inner">

      <div id="tab1" *ngSwitchDefault>
        <div class="themes__table-header">
          <div class="themes__table-col">Темы</div>
          <div class="themes__table-col">Описание</div>
          <div class="themes__table-col">Тип</div>
          <div class="themes__table-col">Макс. Кол-во</div>
          <div class="themes__table-col">Действия</div>
        </div>
        <div class="themes__table-main">
          <div class="themes__table-row" *ngFor="let topic of waitTopics">
            <div class="themes__table-col">{{ topic.name }}</div>
            <div class="themes__table-col">{{ topic.description }}</div>
            <div class="themes__table-col">{{ topic.tip }}</div>
            <div class="themes__table-col">{{ topic.maxCol }}</div>
            <div class="themes__table-col">
              <button (click)="openEditModal(topic)" class="btn btn-
primary">Edit</button>
              <button class="btn btn-danger" mwlConfirmationPopover
[popoverTitle]="popoverTitle" [popoverMessage]="popoverMessage"
placement="left" (confirm)="deleteTopic(topic)"
(cancel)="delete=false">Delete</button>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>
```

```
</div>
  </div>
</div>
</div>

<div id="tab2" *ngSwitchCase="tab2">
  <div class="themes__table-header">
    <div class="themes__table-col">Темы</div>
    <div class="themes__table-col">Описание</div>
    <div class="themes__table-col">Тип</div>
    <div class="themes__table-col">Макс. Кол-во</div>
    <div class="themes__table-col">Студенты</div>
    <div class="themes__table-col">Запросы</div>
  </div>
  <div class="themes__table-main">
    <div class="themes__table-row" *ngFor="let topic of confirmTopics">
      <div class="topic"> {{ topic.name }}</div>
      <div class="trow"> {{ topic.description }}</div>
      <div class="trow"> {{ topic.tip }}</div>
      <div class="trow"> {{ topic.maxCol }}</div>
      <div class="trow" >
        <div *ngFor="let s of topic.students" >
          <ng-container>
            <p [routerLink]="['/teacherDiploma', s.student._id]"
>{{ s.student.full_name }} </p>
          </ng-container>

        </div>
      </div>
    </div>
  </div>
  <div class="trow">
    <div *ngFor="let stud of requests">

      <ng-container *ngIf="stud.topic.name == topic.name">

        {{ stud.student.full_name }}
      <div class="themes__table-row">
        <div>
          <button class="btn btn-success"
(click)="acceptStudent(stud.student,topic._id,stud._id)">Принять</button>
        </div>
      </div>
    </div>
  </div>
</div>
```

Б Қосымшасының жалғасы

```
<button class="btn btn-danger"
(click)="deleteRequest(stud._id)">Отказатъ</button>
  </div>
</div>

  </ng-container>
</div>
</div>
</div>
</div>
</div>

<div id="tab3" *ngSwitchCase="tab3">
  <div class="themes__table-header">
    <div class="themes__table-col">Темы</div>
    <div class="themes__table-col">Описание</div>
    <div class="themes__table-col">Тип</div>
    <div class="themes__table-col">Макс.кол-во</div>
    <div class="themes__table-col">Действия</div>
  </div>
  <div class="themes__table-main">
    <div class="themes__table-row" *ngFor="let topic of notconfirmTopics">
      <div class="themes__table-col">{{ topic.name }}</div>
      <div class="themes__table-col">{{ topic.description }}</div>
      <div class="themes__table-col">{{ topic.tip }}</div>
      <div class="themes__table-col">{{ topic.maxCol }}</div>
      <div class="themes__table-col">
        <button class="btn btn-danger"
(click)="deleteTopic(topic)">Удалить</button>
      </div>
    </div>
  </div>
</div>
</div>
</div>
</div>

</section>
</div>
</div>
```

```
<!--This is Teacher Page component ts --->
import { Component, OnInit } from '@angular/core';
import { HttpClient } from '@angular/common/http';
import { DiplomaService } from '../diploma.service';
import { BsModalService, BsModalRef } from 'ngx-bootstrap/modal';
import { ActivatedRoute, Router } from '@angular/router';
import { Pipe, PipeTransform } from '@angular/core';
import { ShareServiceService } from '../share-service.service';
import { MatSnackBar } from '@angular/material/snack-bar';
import { Topic } from '../Models/Topic';
import { Teacher } from '../Models/Teacher';
import { Request } from '../Models/Request';
import { CreateTopicComponent } from '../create-topic/create-topic.component';
import { EditTopicComponent } from '../edit-topic/edit-topic.component';

@Component({
  selector: 'app-teacher-page',
  templateUrl: './teacher-page.component.html',
  styleUrls: ['./teacher-page.component.css']
})
export class TeacherPageComponent implements OnInit {
  topics: Topic[] = [];
  confirmTopics: Topic[] = [];
  notconfirmTopics: Topic[] = [];
  waitTopics: Topic[] = [];
  teacher: Teacher;
  requests: Request[];
  test;
  message;
  selectedTopic;
  warning = true;
  delete = false;
  bsModalRef: BsModalRef;
  public popoverTitle: string = 'Удалить тему';
  public popoverMessage: string = 'Вы уверены?';
  public viewMode = "tab1";
  constructor(private http: HttpClient, private service: DiplomaService, private router:
  Router,
    private modalService: BsModalService, private route: ActivatedRoute, private
  shareService: ShareServiceService, private snackBar: MatSnackBar) { }
```

```
ngOnInit() {

  this.service.get(this.route.snapshot.params['id']).subscribe((res: Teacher) => {
    this.teacher = res;
    this.getTopics(this.teacher);
    this.getCT(this.teacher._id)
    this.getNCT(this.teacher._id)
    this.getWT(this.teacher._id)
  });

  this.topicsFromCreatePage();

  this.shareService.getTopicEvent.subscribe(
    (something: any) => {
      if (something.name === 'GetTopicsFromEdit') {
        this.getWT(this.teacher._id)
        console.log("in teacher page ")
      }
    }
  )
  this.getRequests();
}

topicsFromCreatePage(){
  console.log("topicsfromcreatepage")
  this.topics = this.shareService.topics;
  this.shareService.getTopicsEvent.subscribe(
    (something: any) => {
      if (something.name === 'GetTopics') {
        console.log("in teacher page ", something.data)
        this.waitTopics = something.data;
      }
    }
  );
}

openPageZK() {
  this.router.navigateByUrl('/zavkafedra/' + this.teacher._id)
}

openModal(teacher) {
  console.log(teacher);
}
```

```
this.bsModalRef = this.modalService.show(CreateTopicComponent);
  this.bsModalRef.content.closeBtnName = 'Close';
  this.getTeacher(teacher);
}
```

```
openEditModal(topic) {
  console.log(topic);
  this.selectedTopic = topic
  this.bsModalRef = this.modalService.show(EditTopicComponent);
  this.bsModalRef.content.closeBtnName = 'Close';
  this.shareService.topic = this.selectedTopic;
  this.shareService.topics = this.waitTopics;
  console.log("topiiic", this.shareService.topic)
  const gs = { name: 'GetTopic', topic:this.shareService.topic,
data:this.shareService.topics }
  this.shareService.getTopicEvent.emit(gs as any);
  console.log("GS", gs)
}
```

```
getTeacher(teacher): void {
  console.log("workGetTeacher");
  this.shareService.teacher = teacher;
  const gs = { name: 'GetTeacher', data: this.shareService.teacher }
  this.shareService.getTeacherEvent.emit(gs as any);
  console.log("gs" + gs);
}
```

```
getRequests() {
  console.log("get req tekseru")
  this.requests = []
  console.log(this.requests)
  this.service.getRequests().subscribe((res: Request[]) => {
    this.requests = res;
  });
}
```

```
acceptStudent(student, id, req_id) {
  student.hasTopic = "true"
  this.service.updateTopicStud(student, id).subscribe((res) => {
    if (res == false) {
      this.message = "Набор закрыт";
      this.snackBar.open(this.message, 'Закрыть', {
```



```
        duration: 3000,
    });

}
else {
    this.service.updateStudent(student).subscribe((result) => {
        student = result;
        this.snackBar.open('Вы успешно добавили студента!', 'Заккрыть', {
            duration: 3000,
        });
        this.deleteRequest(req_id)
        this.getCT(this.teacher._id)
        // this.getTopics(this.teacher)
        this.getRequests()
    })
}
})
}
```

```
deleteRequest(id) {
    this.service.deleteRequest(id).subscribe(() =>
        console.log('Deleted'));
    this.getTopics(this.teacher);
    this.getRequests();
}
```

```
}
getCT(id) {
    this.service.getCT(id).subscribe((res: Topic[]) => {
        this.confirmTopics = res;
    })
}
```

```
getNCT(id) {
    this.service.getNCT(id).subscribe((res: Topic[]) => {
        this.notconfirmTopics = res;
    })
}
```

```
getWT(id) {
    this.service.getWT(id).subscribe((res: Topic[]) => {
```

```
        this.waitTopics = res;
    })
}
getTopics(teacher) {
    this.service.getTopics(teacher._id).subscribe((res: Topic[]) => {
        this.topics = res;

    });
}

redirectToAdd() {
    this.share(this.teacher);
    this.router.navigateByUrl('/create')
}

share(teacher) {
    this.shareService.teacher = teacher;
    const gt = { name: 'GetTeacher', data: this.shareService.teacher }
    this.shareService.getTeacherEvent.emit(gt as any);
}

deleteTopic(topic) {
    if (topic.students.length == 0) {
        this.service.deleteTopic(topic._id).subscribe(() =>
            console.log('Deleted')
        );
        this.getWT(this.teacher._id);
        this.getNCT(this.teacher._id)
    }
    else {
        this.warning = true;
        this.snackBar.open('Вы не можете удалить! У Вас есть студенты.', 'Закреть', {
            duration: 3000,
        });
        this.delete = false;
    }
}
}
```

Форматы	Зонасы	Поз-сы.	Белгілеулер			Аталулары	Саны	Қорытынды
					Дипломдық жоба			Бет
								43
Өлш	Бет	Құжат №	колы	күні				