

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

Сәтбаев университеті

Ақпараттық және телекоммуникациялық технологиялар институты

Киберқауіпсіздік, ақпаратты өңдеу және сақтау кафедрасы

Тілеуқабыл Дидар Дәулетұлы

Балалардың ойлауын дамытуға арналған білім беру қосымшаларын құру

ДИПЛОМДЫҚ ЖҰМЫС

5B070300 – «Ақпараттық жүйелер» мамандығы

Сұрыпдаған:

Тілеуқабыл Д.Д.

Ғылыми бағамдаушы:

Ғылыми жетекші:

PhD докторы, доценттің проф.

Тыңдаушы:

О. А. Бағдатұлы

М. Б. Бағдатұлы

« 28 » мамыр 2019ж.

« 28 » мамыр 2019ж.

Алматы 2019

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

Сәтбаев университеті

Ақпараттық және телекоммуникациялық технологиялар институты

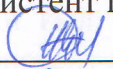
Киберқауіпсіздік, ақпаратты өңдеу және сақтау кафедрасы

ҚОРҒАУҒА ЖІБЕРІЛДІ

КАӨЖС кафедра меңгерушісі,

техн. ғыл. канд. ,

ассистент профессор

 Н.А.Сейлова

« 16 » мамыр 2019 ж.

ДИПЛОМДЫҚ ЖҰМЫС

Тақырыбы: Балалардың ойлауын дамытуға арналған білім беру
қосымшаларын құру

5B070300 – «Ақпараттық жүйелер» мамандығы бойынша

Орындаған :

Тілеуқабыл Д.Д.

Пікір беруші:

PhD докторы, ассистент проф.

 О. А. Баймуратов

« 16 » мамыр 2019ж.

Ғылыми жетекші :

Тьютор

 М. Б. Бауыржан

« 13 » мамыр 2019ж.

Алматы 2019

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

Сәтбаев университеті

Ақпараттық және телекоммуникациялық технологиялар институты

Киберқауіпсіздік, ақпаратты өңдеу және сақтау кафедрасы

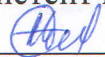
5B070300 – «Ақпараттық жүйелер» мамандығы

БЕКІТЕМІН

КАӨЖС кафедра меңгерушісі,

техн. ғыл. канд. ,

ассистент профессор

 Н. А. Сейлова

« 16 » мамыр 2019 ж.

**Дипломдық жұмысты орындауға
ТАПСЫРМА**

Білім алушы: Тілеуқабыл Дидар Дәулетұлы

Тақырыбы: Балалардың ойлауын дамытуға арналған білім беру
қосымшаларын құру

Университет Ректоры: «16» қазан 2018 ж. № 1162-б бұйрығымен
бекітілген

Аяқталған жұмысты тапсыру мерзімі: «15» мамыр 2019 ж.

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

Дипломдық жұмыста қарастырылатын мәселелер тізімі:

а) қойылған мәселенің қазіргі жағдайын пайымдау

ә) ақпараттық қамтаманы құру

б) программалық қамтаманы құру

Сызбалық материалдар тізімі: Power Point бағдарламасындағы слайдтар

Сызба материалдар: 20 слайдпен көрсетілген

Ұсынылатын негізгі әдебиет: 20 атау

Дипломдық жұмысты дайындау

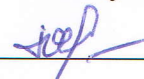
КЕСТЕСІ

Бөлімдер атауы, қарастырылатын мәселелер тізімі	Ғылыми жетекші мен кеңесшілерге көрсету мерзімдері	Ескерту
Мәселенің қазіргі жағдайына шолу және оны талдау	10.01.2019 – 08.03.2019	
Ақпараттық қамтаманы құру	05.02.2019 – 10.03.2019	
Программалық қамтаманы құру	11.03.2019 – 22.04.2019	

Дипломдық жұмысының бөлімдерінің кеңесшілері мен норма бақылаушыларының аяқталған жобаға қойған
қолтаңбалары

Бөлімдер атауы	Кеңесшілер, аты, әкесінің аты, тегі (ғылыми дәрежесі, атағы)	Қол қойылған күні	Қолы
Норма бақылаушы	А.А.Кабдуллин, кафедра ассистенті	13.05.19	
Программалық қамтама	М.Б. Бауыржан ,тьютор	8.05.19	

Ғылыми жетекші  М. Б. Бауыржан

Тапсырманы орындауға алған білім алушы  Д. Д. Тілеуқабыл

Күні “ 08 ” 01 2019 ж.

ҚАЗАҚСТАН РЕСПУБЛИКАСЫНЫҢ БІЛІМ ЖӘНЕ ҒЫЛЫМ
МИНИСТРЛІГІ
СӨТБАЕВ университеті

Ғылыми жетекшінің сын пікірі

Дипломдық жұмысқа

Тілеуқабыл Дидар Дәулетұлына

5B070300 – Ақпараттық жүйелер

Тақырыбы: Балалардың ойлауын дамытуға арналған білім беру қосымшасын құру

Бұл дипломдық жұмыс “Visual Studio” бағдарламалау ортасында құрастырылды.

Дипломдық жұмысқа түсіндірме хат кіріспеден, қорытындыдан, пайдаланылған әдебиеттер тізімінен тұрады.

Кіріспеде мәселенің маңыздылығы, жобаның мақсаты және қолданылған құралдар қысқаша келтірілген. Бұл дипломдық жұмыста балалардың ой өрісін дамытуға көмектесетін ойындар қарастырылған, сонымен қатар көптеген логиканы машықтыратын материалдар берілген.

Тілеуқабыл Дидар дипломдық жұмыс орындау барысында өз беттілік көрсетіп, әдебиеттермен жұмыс істеген. Дипломдық жұмыс “Ақпараттық жүйелер” мамандығының бітіру жұмыстарына қойылатын талаптарға сәйкес.

Қорытындылай келе, Тілеуқабыл Дидардың дипломдық жұмысын қорғауға жіберуге болады.

Ғылыми жетекші:

Техника ғылымдарының магистрі, тьютор

 М. Б. Бауыржан

«13» 05 2019 ж

РЕЦЕНЗИЯ

Дипломдық жұмыс

(жұмыс түрінің атауы)

Тілеукабыл Дидар Дәулетұлы

(білім алушының Т.А.Ә.)

5B070300 – Ақпараттық жүйелер

(мамандық атауы мен шифрі)

Тақырыбы: «Балалардың ойлауын дамытуға арналған білім беру қосымшаларын құру»

Орындалды:

түсініктеме _____ 52 бет

Тілеукабыл Д.Д. дипломдық жұмысы балалардың ойлау қабілетін арттыруға арналған. Осы тақырыпқа сәйкес ақпараттық жүйелер негізінде – балалардың ойлау қабілетін арттырудағы мәселелердің бірі болып табылады. Бұл жұмыс осы мәселені шешуге арналған. Сол үшін орындалған жұмыс практикалық маңызға ие.

ЖҰМЫСҚА ЕСКЕРТУ

Жұмысқа келесідей ескертулер жасалды:

- түсініктемелік жазбада/анықтамаларда грамматикалық және стилистикалық қателер кездеседі;
- ақпарат көздеріне сілтемелер толық келтірілмеген;
- деректер базасының кестелер құрылымы дипломдық жұмыста қарастырылмаған;
- жасалған бағдарламаның сипаттамасын мен талаптары келтірілмеген.

ЖҰМЫСТЫҢ БАҒАСЫ

Дипломдық жұмыс тапсырмаға сәйкес толық орындалған және «75» бағаға бағалап, ал жұмыстың авторы Тілеукабыл Дидар Дәулетұлы 5B070300 – «Ақпараттық жүйелер» мамандығы бойынша «бакалавр» академиялық дәрежесін алуға лайық деп санаймын.

РЕЦЕНЗЕНТ

PhD докторы, ассистент проф.

«16» мамыр 2019 ж.



О. А. Баймуратов

Протокол анализа Отчета подобия

заведующего кафедрой / начальника структурного подразделения

Заведующий кафедрой / начальник структурного подразделения заявляет, что ознакомился(-ась) с Полным отчетом подобия, который был сгенерирован Системой выявления и предотвращения плагиата в отношении работы:

Автор: Тілеукабыл Дидар

Название: Балалардың ойлау қабылетін дамытуға арналған қосымша кұру

Координатор: Мадина Бауыржан

Коэффициент подобия 1:0,5

Коэффициент подобия 2:0

Тревога:0

После анализа отчета подобия заведующий кафедрой / начальник структурного подразделения констатирует следующее:

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

Обоснование:

.....
.....
.....
.....
.....
.....

Дата 16.05.192

Подпись заведующего кафедрой /

начальника структурного подразделения




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: Тілеуқабыл Дидар

Название: Балалардың ойлау қабілетін дамытуға арналған қосымша құру

Координатор: Мадина Бауыржан

Коэффициент подобия 1: 0,5

Коэффициент подобия 2: 0

Тревога: 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:

.....
.....
.....
.....
.....
.....

13.05.19 г.

Дата



Подпись Научного руководителя

Самые длинные фрагменты, определенные, как подобные

Документы, в которых найдено подобные фрагменты: из RefBooks

Не обнаружено каких-либо заимствований

Документы, содержащие подобные фрагменты: Из домашней базы данных

Не обнаружено каких-либо заимствований

Документы, содержащие подобные фрагменты: Из внешних баз данных

Не обнаружено каких-либо заимствований

Документы, содержащие подобные фрагменты: Из интернета

Документы, выделенные жирным шрифтом, содержат фрагменты потенциального плагиата, то есть превышающие лимит в длине коэффициент подобия № 2

№	Источник гиперссылки	Количество одинаковых слов (количество фрагментов)
1	URL_ https://kazlib.kz/marketingtik-zhosparlau-turaly-sljajd/	24 (1)
2	URL_ http://www.diplomkaz.kz/wp-content/uploads/2013/02/%d0%94%d0%b8%d0%bf.-%d0%9a%d3%98%d0%9c%d0%95%d0%9b%d0%95%d0%a2%d0%9a%d0%95-%d0%a2%	5 (1)

АҢДАТПА

Дипломдық жұмыстың тақырыбы «Балалардың ойлауын дамытуға арналған білім беру қосымшаларын құру». Интерактивті білім беру мектебі бұл жұмыста, атап айтқанда оқу үрдісінде зерттеу объектісі болып табылады. Зерттеудің тақырыбы - мектептің оқу бағдарламаларын жақсы меңгеру үшін интерактивті оқытуды дамыту.

Дипломдық жұмыс үш бөлімнен тұрады. Кіріспе менің жұмысымның негізгі мақсатын, сондай-ақ интерактивті оқытудың өзектілігін сипаттайды. Бөлімдер даму құрылымынан құрылған толыққанды сайтты сипаттайды. Қорытындыда осы жұмыстың негізгі қорытындылары мен ұсыныстары келтірілген.

АННОТАЦИЯ

Тема дипломной работы "Разработка интерактивно-образовательного приложения для развития детского мышления". Школа интерактивного образования послужила объектом исследования в дипломной работе, а именно процесс обучения. Предметом исследования является разработка интерактивного обучения, для лучшего усвоения материала школьной программы.

Данная работа состоит из трех разделов. Во введении раскрыта главная цель моей работы, а также актуальность интерактивного обучения. Разделы описывают от структуры разработки до созданного полнофункционального сайта. В заключении представлены основные выводы и предложения по данной проделанной работе.

ANNOTATION

Theme of the thesis "Development of interactive and educational applications for the development of children's thinking." The school of interactive education has served as the object of study in the thesis work, namely the learning process. The subject of the study is the development of interactive learning, for better learning of the school curriculum material.

This work consists of three sections. The introduction describes the main purpose of my work, as well as the relevance of interactive learning. Sections describe from the development structure to the created full-featured site. In the conclusion, the main conclusions and proposals on this work done are presented.

МАЗМҰНЫ

	КІРІСПЕ	8
1	Web-қосымшалардың қазіргі жағдайы және ұқсас жұмыстарға шолу	9
1.1	Интернеттің қазіргі кездегі маңыздылығы	9
1.2	Балаларды оқыту үрдісінде ойындардың алатын орны	9
1.3	Білім беру веб-платформаларына шолу	11
1.3.1	"Смартария" білім беру платформасы	11
1.3.2	ИНТУИТ ұлттық университеті	11
1.3.3	Постнаука" білім беру платформасы	12
1.4	Есептің қойылымы	13
2	Балалардың ойлау қабілетін дамытуға арналған web-қосымша құру үшін тілдерді таңдау	15
2.1	Веб-қосымшаны әзірлеудің негізгі құралдары	15
2.2	HTML тілін қолдану негіздері	15
2.3	JavaScript скрипттер тілі	16
2.3.1	JavaScript тілінің қолданылуы	16
2.3.2	ReactJs технологиясы	16
2.3.3	NodeJs технологиясы	18
2.4	CSS стильдер кестесі	20
2.5	PostgreSql мәліметтер базасы	20
2.6	Мәліметтер базасын құру	21
2.7	Парақшалардың дизайнын жасау үшін "Figma" жіңішке клиенті	23
2.8	Web-бағдарламаның құрылымы	24
2.8.1	Web-платформада мәліметтерді жүктеу процесі	24
3	Балалардың ойлау қабілетін дамытуға арналған web-қосымша құру	26
3.1	Web-қосымшаның бағдарламалық қамтамасыз етуінің сипаттамасы	26
3.1.1	Клиентпен өзара әрекеттесу құрылымының жалпы мәліметтері	27
3.2	Қосымша прототипін жобалау	27
3.3	SPA әзірлеу	33
3.4	Серверлік бөлікті әзірлеу	34
3.4.1	Онлайн чат құру	36
3.5	"Дақ" ойынын құру	37
	ҚОРЫТЫНДЫ	41
	ПАЙДАЛАНЫЛҒАН ӘДЕБИЕТТЕР ТІЗІМІ	42
	А Қосымшасы	43

КІРІСПЕ

Дипломдық жұмыстың тақырыбы бойынша өскелең ұрпақтың білім алуына байланысты мәселелер келтіріледі. Білім беру туралы мәселе әр мемлекеттің маңызды бөлігі болып табылады. Қазіргі уақытта біздің елімізде тұрақтылық жоқ: министрлер, құрылымдар, білім беруге байланысты көптеген өзгерістердің болуы балалардың дамуына үлкен зиян келтіреді.

Бұл жұмыста барлық оқулықтар балаларға қол жетімді емес, көптеген оқулықтарды сатып алу өте қымбатқа түседі, білім беру орындарында балалардың қызығушылығын ояту үшін мультимедиялық құралдарды қолдану мәселесі қарастырылады.

Қазіргі кезде түрлі курстар, үйірмелер, онлайн-платформалар бар - бұл әрине жақсы, сапалы материал, жақсы әдістемелер бар, бірақ бір үлкен кемшілік – бұл материалдың бағасы. Барлық ата-аналар өз балаларын осындай курстарға жіберуге, онлайн-бағдарламаларды сатып алуға мүмкіндігі жоқ және қаржылық тұрғыдан бұл осы курстардың бағасы қымбат болғандықтан проблема болып табылады.

Теориялық материалды қызықты оқып-үйрену үшін қызықты практикалық бөлім құрылады, атап айтқанда жарыс түріндегі ойын. Өйткені ойын арқылы балаларға материалды жеткізу ыңғайлы болғандықтан барлық күш осы ойынды құруға бағытталады. Ойындар жарыс түрінде өтеді, жеңіске жеткен жеңімпаздар бонустар немесе ұпайға ие болады. Сондай-ақ, бір-бірімен байланыста болу үшін онлайн чат құрылады. Ойындар, жарыстар тек білім беру сипатында болады, мысалы, логикалық, ойлау, арифметикалық үдерістерді дамытады, өйткені бұл үш дағдыларға көптеген білім беру бағдарламаларында өте аз көңіл бөлінеді.

Жұмысты жасауда алға қойылған мақсат - балаларды білімге, ғылымға тарту және сапалы материал алу үшін қолайлы онлайн-платформа құру болып табылады.

Бұл дипломдық жұмыста қолданыстағы жүйелерді талдау, мәліметтер базасын жобалау, бағдарлама прототипін құру және бірбеттік қосымшаны әзірлеу бойынша жұмыстар жүргізіледі. Сонымен қатар қолданушылардың әртүрлі дағдыларын дамытуға көмектесетін ойындар құру келтіріледі.

1 Web-қосымшалардың қазіргі жағдайы және ұқсас жұмыстарға шолу

1.1 Интернеттің қазіргі кездегі маңыздылығы

Қазіргі уақытта әрқайсымызда смартфон немесе ноутбук және интернет бар болса, кез келген ақпаратты ыңғайлы уақыт аралығында алуға мүмкіндігіміз бар. Интернет арқылы жылдам коммуникация, қажетті ақпаратты іздеу мүмкіндігі пайда болды. Әрбір мәліметтер көзі әртүрлі ақпарат береді, сондықтан сапалы материалды табу қиынға соғады.

Интернеттің пайда болуымен жарнама мәселесі өзекті болып табылады. Кей кездері бізге санамыз зардап шегетін әртүрлі ақпараттарды оқуға тура келеді, соның салдарынан санамыз қажетсіз ақпаратпен ластанады. Білімді ересек адамдар мұндай айлаға сене бермейді, бірақ өкінішке орай балалар, жасөспірімдер, олардың өмірінде, жұмыста, отбасында үлкен жетістіктерге әкелмейтін ақпараттарға сенгіш болады.

Интернеттің түрлі теріс факторларынан басқа оң жақтары да бар. Интернет арқылы әртүрлі жастағы адамдардың білім алуымен тығыз байланысты түрлі онлайн курстар, платформалар пайда болды. Арнайы білім алу үшін арнайы қалаға барудың қажеті жоқ, қажетті платформаға тіркеліп, интерактивті форматта білім алуға болады. Мұндай курстар сіздің еліңізден басқа жерлерде болуы мүмкін, бірақ Интернет арқылы сіз Тіл тасымалдаушысынан шет тілін үйрене аласыз, Oxford университетінде оқылатын дәрістерге қатыса аласыз, электр сызбасын жинау және шамды қосу - мұның бәрі Интернет арқылы жүзеге асырылады.

Әртүрлі онлайн курстардың көмегімен, адамдар армандай алмайтын салаларда өздерін сынап көре алады. Дизайнерлер, суретшілер, жазушылар, бағдарламалаушылар – бұл тізімді ұзақ жалғастыруға болады, бірақ осы білімдердің барлығын үйден шықпай-ақ сатып алуға болады.

1.2 Балаларды оқыту үрдісінде ойындардың алатын орны

Біздің ойындар жайлы білетіндеріміздің барлығы теледидар, интернет бойынша беріле бермейді. Оқыту үрдісінде ойындардың негізгі артықшылықтарын қарастырайық (1.1-сурет).

Оқу жылдамдығы, экранда ойынға арналған практикалық нұсқаулар көз алдында тұрады, олар жылдам реакция мен өзара әрекеттесуді талап етеді. Ақпаратты жылдам қабылдап, жылдам жауап беру қажет болады. Ойнайтын балалар монитордан мәтінді лезде оқитыны ескерілген.



1.1-сурет – Компьютерлік ойындардың артықшылықтары

Жедел реакция, тіпті ересектердің өздері жиі ойнайтын атыспақ ойындарында тірі қарсы әрекет жағдайы талап етіледі, ауыспалы жағдайда теңбе-тең реакция талап етіледі - бұл өмір сүру үшін өте пайдалы сапа болып табылады.

Шағын моториканы дамыту, компьютерлік ойында әрекеттерді үйлестіру үшін, өз кезегінде, ұсақ моторика қажет (уақыт өте келе барлық саусақпен мәтінді тез теру).

Көру-моторлы үйлестірудің дамуы, қолданушылардың ойын ойнау барысында бір мезгілде өз саусақтарын және мониторды бақылап үлгермейді. Мінекей, бала да дәл осылай әрдайым үлгермейді. Ол өз қолдарының қозғалысын автоматтандыруға әкеледі, мұның бәрі мониторда оның көз алдында көрінеді.

Логикалық ойлау қабілетін дамыту, қойылған мақсатқа жету үшін бірдей қадамдар қабылдаймыз. Ал бүгінгі ойындардың майнкрафт сияқты вариативтілігін ескере отырып, бір мақсатқа жету үшін бірнеше жолдар бар – бұл ауқымды ойлау қабілетін жетілдіреді.

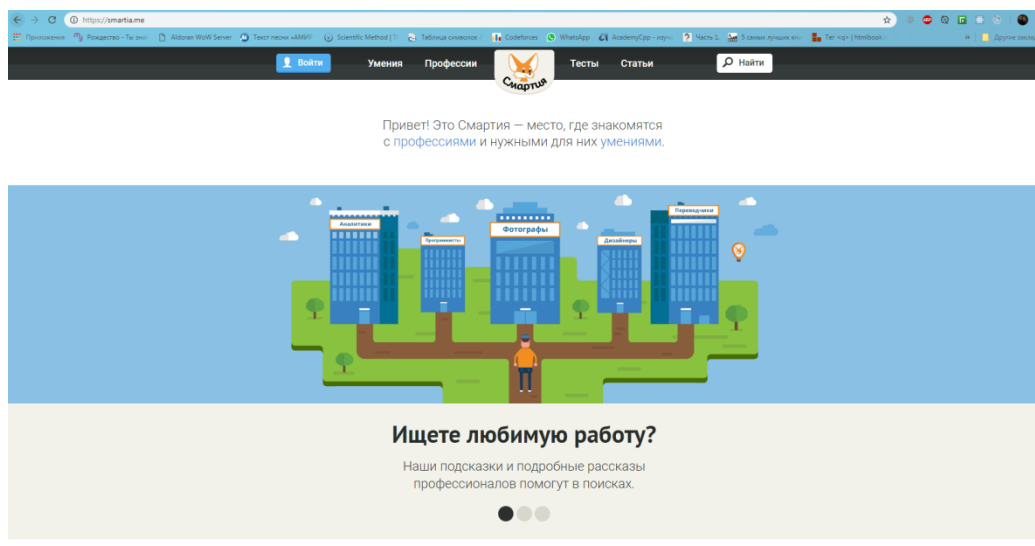
Эмоционалдық шиеленісті алып тастау, мектепте және үйде балалар стресс алу үшін жеткілікті себептер көп, бірақ оларды алып тастаудың тәсілдері жетіспейді. Бұл жағдайда барлық зұлымдарды құртып, әлемді құтқарып, әдетте болмайтын хандық құрып немесе ары қарайғы жетістіктер үшін алты қырдың ар жағына өтіп кетуге болатын компьютерлік ойындар көмектеседі [1].

1.3 Білім беру веб-платформаларына шолу

Веб-платформалардың бірнеше мысалын қарастырайық, олар арқылы әр түрлі білім алуға болады, олар бізге өмірде қажет болады. Шағын талдау жүргізіп, төменде көрсетілген жүйелердің әрқайсысының тәсілін қарастырамыз.

1.3.1 "Смартария" білім беру платформасы

"Смартария" платформасы сіз таңдаған бағыт бойынша материалды оқып-үйренуге арналған. Онда сіз қолданғыңыз келетін әр түрлі дағдыларды алуға немесе белгілі бір мамандықты таңдауға және осы кәсіппен байланысты материалды меңгеруге болады. Бұл платформа арқылы алуға болатын мамандықтар: Product manager, SMM-маман, веб-талдаушы, веб-дизайнер, ағылшын тілінен аудармашы, журналист, графикалық интерфейстерді жобалаушы, ВІ әзірлеуші, БҚ тестілеушісі, фотограф (1.2-сурет).



1.2-сурет – "Смартария" веб-платформасының басты беті

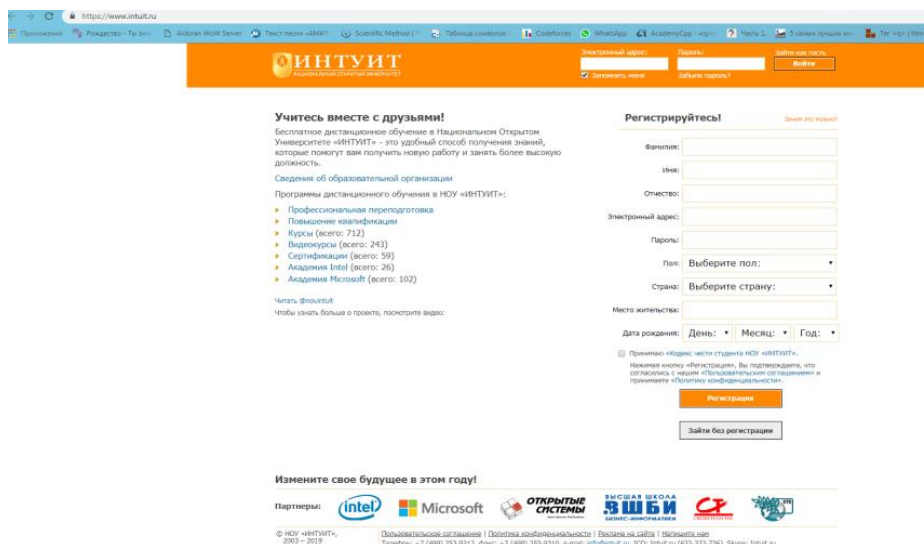
Бұл веб-платформа веб-қосымшаларды құрудың ескірген технологияларын пайдаланады. Әрбір шақыру back-серверге сұраныс жасайды, бұл өз кезегінде back-серверге жүктемені арттырады. Әрбір жаңа сұраныс жаңа html файлын жүктейді, ол браузермен қайта тексеріледі.

1.3.2 ИНТУИТ ұлттық университеті

ИНТУИТ ұлттық университет - әр түрлі материалды дәрістік форматта меңгеруге арналған веб-платформа. Түрлі курстардың үлкен таңдауы бар. Оқу келесі форматта өтеді: материалды оқу және тестілеуден өту, барлық тесттен

өткеннен кейін емтиханнан өтеді және емтиханды сәтті тапсыру кезінде пайдаланушының атына сертификат беріледі. Сондай-ақ, пайдаланушының өз сынақ кітапшасы бар, онда ол кез келген уақытта өз жетістіктерін көре алады.

Біраз уақыттан кейін дәріс оқып қана білім алу жалықтыра бастайды және оқыту қызықсыз бола бастайды.



1.3-сурет – "ИНТУИТ" тіркеу беті

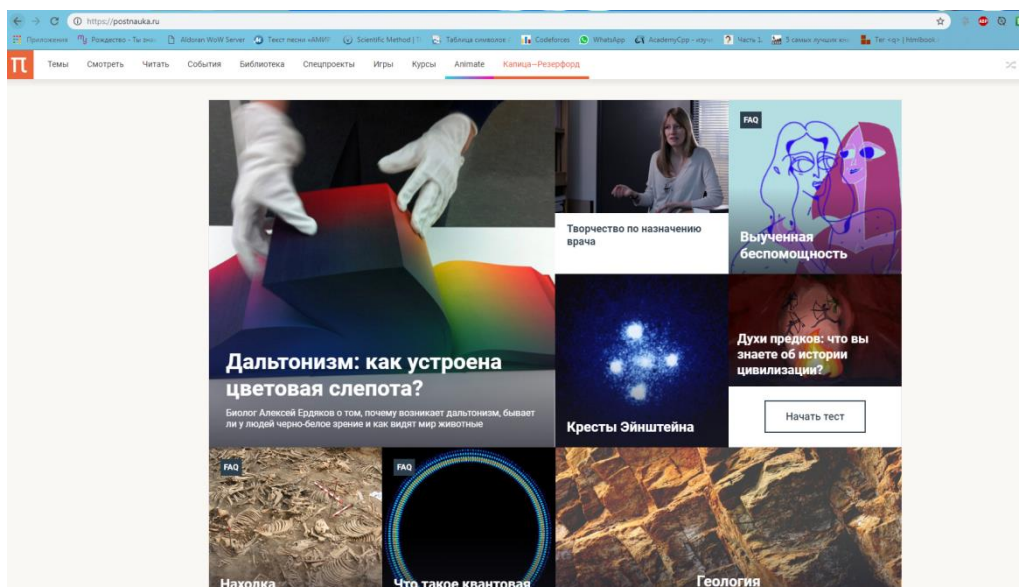
Бұл веб-платформа көптеген парактық архитектурадан тұрады, ол өз кезегінде серверді жүктейді. PHP тілінде жазылған "Drupal" бағдарламалық құралын пайдалану веб-қосымшаның жылдамдығына кепілдік бермейді. Бұл технология ескірген технологиялар қатарына жатады және үлкен жүктеме жоспарланбаған қарапайым сайт үшін қолайлы. Бұл бағдарламалық қамтама GPL лицензиясын пайдаланады, бұл бағдарлама тегін болып табылады және оны әрбір пайдаланушы пайдалана алады және кеңейте алады. Бұл тәсіл ыңғайлы, бірақ көптеген жаңа қосылған технологиялар ескі технологиялармен үйлесімді болмайды. Сайтты құрушылар қиын оқылатын және өзгертіліп отыратын кодты жаза алады.

Пайдаланушының осы платформаны пайдалануына қызықтыратын материалдар жоқ десе де болады. Пайдаланушыны осы сайтты қолдануына негіз болатын интерактивті материал жеткіліксіз (1.3-сурет).

1.3.3 "Постнаука" білім беру платформасы

"Постнаука" веб-порталы – интерактивті ойын-сауық стилінде ғылыми-көп тараған форматтағы материалдар жарияланатын білім беру порталы. Бұл веб-сайтта қызықты материал табуға болады, қолданушыны қызықтыратын кез келген тақырыпқа арналған материал, түрлі әлемдік компаниялардан курстар өтуге болады.

"Біз бірінші тұлғадан кез-келген қайта болжауды ұнатамыз және алдымызға ғалымдар тек қана академиялық дәрежеде ғана емес, кең аудиторияға танымал болатын платформа құру міндетін қоямыз. Барлық авторлар Постнауки - ғылыми пәндердегі сарапшылар."- "Постнаука" сайтының бетінен дәйексөз (1.4-сурет) [2].



1.4-сурет – "ПостНаука" платформасының басты беті

Бұл платформа пайдаланушымен өзара іс-әрекеттің жеткілікті ойластырылған технологиясынан тұрады. Интерактивті дизайн пайдаланушыны осы веб-қосымшамен жұмыс істеуге үйретеді. Негізгі мақсат материалды оқуға немесе қарауға бағытталған.

1.4 Есептің қойылымы

Қолданыстағы веб-жүйелерді талдай отырып, осы жүйелер клиенттерді тарту үшін тегін контентті пайдаланатыны туралы қорытынды жасалды, кейін материалдың ақылы нұсқасын сатып алу қажет болатындығы да белгілі. Сондай-ақ веб-платформаларды пайдаланушылардың нашар әлеуметтік байланысы келтірілген.

Бұл дипломдық жұмыстың мақсаты веб-платформада интерактивті-оқыту қосымшасын құру, веб-қосымшаның жақсартылған әлеуметтік бөлігін қосу болып табылады. Пайдаланушының материалды толық деңгейде меңгеруі үшін жақсартылған архитектура құру қажет. Ақылы курстарды төлей алмайтын халық үшін ой процестерін дамытуға арналған барынша ыңғайлы платформа құру.

Ақпараттық технологиялардың қазіргі әлемде интерактивті - білім беру веб-қосымшасын құру үшін келесі талаптарды орындау қажет:

- бар веб-технологияларға шолу жасау және салыстырмалы талдау жасау;
- қымбат техникалық құрылғыларды пайдаланбастан, платформамен өзара әрекет жасау мүмкіндігін қамтамасыз ету;
- бір-бірімен жеңіл өзара байланысты әлеуметтік кеңістік құру;
- веб-платформаны жобалау кезінде заманауи, берік технологияларды қолдану;
- SPA принциптерін қолдану;
- веб-платформа дизайнын жобалау;
- МБ жобалау;
- МБ концептуалды моделін құру;
- МБ-мен байланыс құру;
- МБ-на материалды қосу үшін платформаны әзірлеу;
- пайдаланушылар үшін қарапайым платформаны әзірлеу.

Дипломдық жұмысты құру барысында Node.js технологиясы, Visual Studio Code, PostgreSQL мәліметтер базасын қолданып, мультимедиялық веб-қосымша құрылады.

2 Балалардың ойлау қабілетін дамытуға арналған web-қосымша құру үшін тілдерді таңдау

2.1 Веб-қосымшаны әзірлеудің негізгі құралдары

Осы бағдарламаны әзірлеу үшін SPA стиліндегі бетті құру әдісі таңдалды (ағылш. Single page application), бұл тек бір html файлын пайдалануды білдіреді. Мұндай тәсіл серверге тікелей әсер етеді:

- ол html құжатты бір рет жібереді, кейін клиент элементтерді беттерге өзі өзгертеді;
- http сұраныстары арқылы клиент пен сервердің өзара әрекеттесуін жүктейді;
- клиент пен веб-сервер жағында кіріс және шығыс трафигін пайдалануды азайтады.

Веб-беттерді құру кезінде HTML (ағылш. Hyper Text Markup Language – веб-қосымша бетінде элементтерді орналастыру құрылымын өңдеу үшін қолданылады. CSS (ағылш. Cascading Style Sheets) - сайттың беттерінде элементтердің көрінісін сипаттайтын псевдо-тіл болып табылады. JavaScript-мультипарадигмалық бағдарламалау тілі, объектілі-бағытталған, императивті және функционалдық стиль ұғымын қолдайды.

Бұл дипломдық жұмысты жазу кезінде бағдарламалық кодты жазу үшін келесі құрал қолданылады: Visual Studio Code. Таңдалған тілдер үшін жазуға ыңғайлы және кодты тестілеу мүмкіндігі бар.

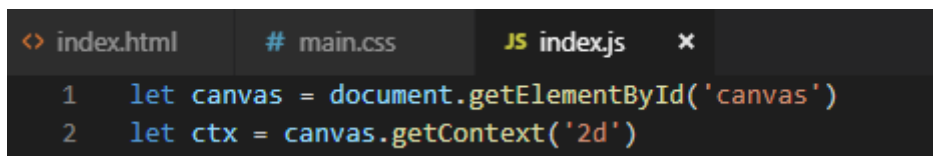
Мәліметтер базасын басқару жүйесі бағдарламалық өнімнің маңызды бөлігі болып табылады. PostgreSQL тегін МББЖ, веб-қосымшалармен өзара байланыс үшін өте қолайлы және МБ пайдалану мониторингі үшін GUI келтірілген.

2.2 HTML тілін қолдану негіздері

Браузерлер html-ді жақсы қабылдайды және оны түсінікті түрде түсіндіре алады. Жалпы, сайттың кез келген беті - браузер пайдаланушысы үшін достық интерфейс түріне аударатын html коды. Кез келген беттің коды барлығына қолжетімді. Html белгілеу тілі кең танымал болды. Қазіргі уақытта - бұл сайттың белгісі жасалатын жалғыз тіл. HTML тег-элементтердің бірі құрылым үшін қажет html құжаттар [3].

Осы гипермәтіндік белгілеулердің арқасында веб-қосымшамыздың қаңқасы, олардың бетте орналасуы және әртүрлі блоктардың шектеулері жасалады. HTML орнында тұрмайды, соңғы жаңартулардың шығуымен canvas сияқты тег қосылды. Бұл бет әртүрлі графикалық манипуляциялар жасауға мүмкіндік береді, ал бұл тегтің үлкен артықшылықтарының бірі ойындарды дамыту болып табылады. Әр түрлі ойындар жасау үшін кодтың келесі

жолдарын жазу қажет (2.1-сурет), осы екі жолдың арқасында біз хостта бөлінген кез келген нәрселерді тарта аламыз.



```
<> index.html    # main.css    JS index.js    x
1  let canvas = document.getElementById('canvas')
2  let ctx = canvas.getContext('2d')
```

2.1-сурет – 2D-контекст құру

2.3 JavaScript скрипттер тілі

JavaScript бастапқыда web-беттерді "тірілту" үшін құрылды. Бұл тілдегі бағдарламалар скрипттер деп аталады. Браузерде олар HTML-ге тікелей қосылады және бет жүктелгеннен кейін бірден орындалады. JavaScript бағдарламасы - қарапайым мәтін. Олар қандай да бір арнайы дайындықты талап етпейді. Бұл тұрғыда JavaScript Java деп аталатын басқа тілден өте ерекшеленеді [4].

2.3.1 JavaScript тілінің қолданылуы

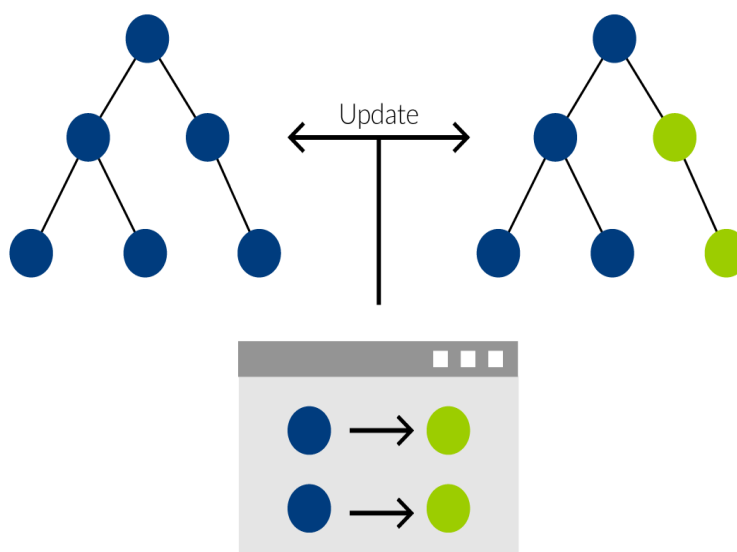
Қазіргі заманғы JavaScript – бұл жалпы мақсаттағы "қауіпсіз" бағдарламалау тілі. Ол жадымен, процессормен жұмыс істеудің төменгі деңгейлі құралдарын бермейді, себебі бастапқыда қажет емес браузерлерге бағытталған. Басқа мүмкіндіктерге келетін болсақ, олар JavaScript іске қосылған қоршаған ортаға байланысты. JavaScript браузерінде бет манипуляциясына, келушімен өзара іс-қимылға және сервермен қандай да бір шамада барлық нәрселерді жасай алады:

- жаңа HTML тегтерін жасау, қолданыстағыларын жою, элементтердің стилін өзгерту, жасыру, элементтерді көрсету;
- келушінің іс-әрекетіне жауап беру, тінтуірдің батырмаларын өңдеу, курсорды жылжыту, пернетақтаға басу;
- серверге сұраныстарды жіберу және бетті қайта жүктеусіз деректерді жүктеу (бұл технология "AJAX");
- cookie алу және орнату, деректерді сұрату, хабарламаларды шығару [5].

2.3.2 ReactJs Технологиясы

React - пайдаланушы интерфейстерін жасауға арналған кітапхана. Оның ерекшеліктерінің бірі JSX қолдану мүмкіндігі болып табылады, JavaScript компиляцияланатын HTML синтаксисіне жақын бағдарламалау тілі.

Әзірлеушілер жоғары өнімділік қосымшаларының көмегімен Virtual DOM-ды қолдануы мүмкін (2.2-сурет). React жайсыз жағдайдан құтылуға көмектесетін изоморфты қолданбаларды жасай алады, пайдаланушы деректерді жүктеу аяқталған кезде күтетін кезде және компьютер экранында жүктеу анимациясынан басқа бір нәрсе пайда болады. Құрылған компоненттер оңай өзгертілуі және жаңа жобаларда қайта пайдаланылуы мүмкін. Кодты қайта пайдаланудың жоғары пайызы тестілердің жабылуын арттырады, бұл өз кезегінде сапаны бақылаудың жоғары деңгейіне әкеледі. Android және iOS үшін React Native мобильді қосымшаларды пайдалана отырып, JavaScript және React әзірлеу тәжірибесін пайдалануға болады. Бұл әзірлеушілер үшін ойлауға мүмкіндік беретін техникалық ерекшеліктер болды.



2.2-сурет – Виртуалды DOM жұмысының үлгісі

Document Object Model немесе DOM-HTML, XHTML және XML құжаттарындағы нысандармен таныстыру және өзара әрекеттесу тәсілі. Осы модельге сәйкес, әрбір құжат Dom ағаш деп аталатын элементтердің иерархиялық ағашы болып табылады. Арнайы әдістерді пайдалана отырып, біз құжатымыздың белгілі бір элементтеріне қол жеткізе аламыз және оларды біз қалағандай өзгерте аламыз. Динамикалық интерактивті веб-бетті жасағанда, белгілі бір элементтің күйі өзгергеннен кейін, DOM тез жаңартылғанын қалаймыз. Осы мәселе үшін кейбір фреймворктар "dirty checking" деп аталатын әдісті қолданады және құжаттың жай-күйін үнемі зерттеу және деректер құрылымындағы өзгерістерді тексеру болып табылады. Бұл міндет жоғары жүктелген қосымшалар жағдайында нағыз бас ауруы болуы мүмкін. Virtual DOM, өз кезегінде, жадта сақталады. Сондықтан "нақты" DOM өзгергенде, react Virtual DOM-ді бір сәтте өзгерте алады. React "жинақтайды" мұндай өзгерістерді DOM жағдайымен салыстырады, содан кейін өзгерген компоненттерді қайта суреттейді. Бұл тәсілде сіз DOM тұрақты жаңартуды жасамайсыз. Сондықтан React қосымшаларының жоғары өнімділігіне қол

жеткізуге болады. Екінші салдары React изоморфты табиғатынан туындайды: сіз клиент жағында сияқты сервердің жағында рендеринг жасай аласыз.

2.3.3 NodeJs технологиясы

Веб-қосымшаларды жүзеге асыру үшін алдыңғы қатарлы және өте қызықты платформа - Node.js қолданылады. Жұмыс жасау барысында осы құралдың барлық артықшылықтарын егжей-тегжейлі қарастырамыз, құрылыстың ерекшеліктері туралы ақпарат береміз және осы платформаны нақты жобаларда қолданудың бірнеше мысалдарын талдаймыз. Node.js салыстырмалы жаңадан құрылған жоба: әзірлеушілер 2009 жылы веб-ресурстарды әзірлеудің идеялық жаңа технологиясының шығуы туралы мәліметтер берді. Содан бері платформа өсіп, бүкіл әлем бойынша көптеген бағдарламашылар жеткілікті ыңғайлы түрде пайдаланады.

Платформаның бірінші және ең пайдалы қасиеті, әсіресе жаңа адамдар үшін - қарапайымдылық пен қолжетімділік. Өз жобаларын жүзеге асыру үшін сіз осы өнім туралы кітапханаларды, пакеттерді, ақпаратты жарма арқылы қолмен жинаудың қажеті жоқ. Платформаны орнату бірнеше минуттарды алады, содан кейін Node жұмысқа дайын. Құрылымдық Node.js қозғалтқышты (V8), енгізу-шығару құралдарын (I/O) және кітапханалар жиынтығын біріктіру болып табылады. Сонымен қатар, қолданба немесе сайт компоненттерін жасау бір бағдарламалау тілінде жүргізіледі, ол JavaScript. Сонымен қатар қосымша ресурстарды үйрену немесе басқа тіл маманын тарту қажет емес, егер осы икемді тілді үйренуге мұқият қарайтын болса, оны үйрену қиын емес, JS-та кәсіптік деңгейде бағдарламалау дағдыларын меңгеру жеткілікті. Қорытынды ретінде сервер және клиент бір тілде жазылған (2.3-сурет).

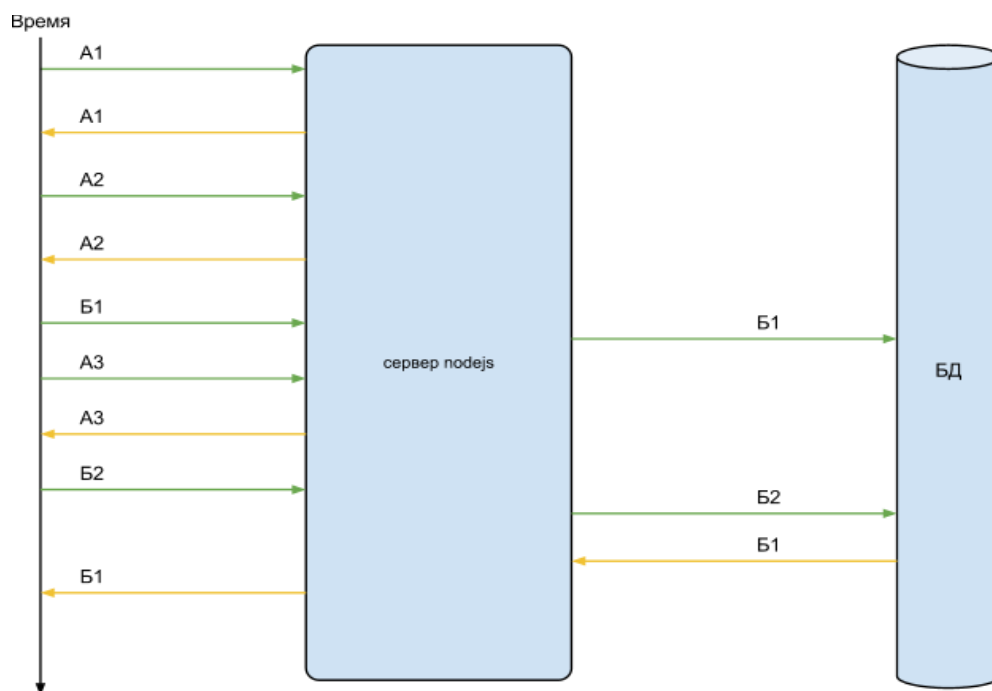


2.3-сурет – Клиент-серверлік қосымша

Кейбір қолданушылар клиентте және серверде бірыңғай кодты Node артықшылығы ретінде шығарады. Теориялық тұрғыдан ешкім мұны жоққа шығармайды, көптеген жобаларда да бар, бірақ ортақ кітапханалар мен жалпы бөлімдер үшін ғана қолданылады. Көптеген жағдайларда клиент пен сервердегі код іс жүзінде ерекшеленеді. Алайда, бұл платформамен жұмысты қиындатпайды. Сонымен қатар, Node бағдарламалаудың оқиғалы-бағытталған парадигмасын айтады, оған сәйкес оқиғалар бағдарламаны орындауды анықтайды, қосымшаларды құру орындау процесінде орын алуы мүмкін

көптеген оқиғаны өңдеушілерді бағдарламалаудың көмегімен жүзеге асырылады (2.4-сурет).

Бұл платформаның тағы бір артықшылығы - ОЖ барлық түрлері үшін құрылған. Node.js-ті орнату Mac және Windows және Unix жүйелерінде бірдей қарапайым болады. Осылайша, осы өнімнің қол жетімділігі мен тұрақтылығы нығая түседі.



2.4-сурет – Көптеген өңдеушілердің мысалы

Енді платформаның мәнін түсіну үшін тереңдете түсеміз. Бұған дейін V8 қозғалтқышы келтірілген. Қазіргі уақытта бұл JS үшін ең туынды бағдарламалардың бірі. Оның арқасында код жылдамырақ орындалады. V8 жадты тиімді басқарады:

- Қоқысты жинауға арналған үзіктер;
- Кодты орындау кезінде осы үзіктердің әсерін азайту;
- Көрсеткіштер мен объектілерді оңтайлы сақтау, соның салдарынан-жад жоғалту проблемасын жою;
- Ақпаратқа жылдам қол жеткізу.

V8-ден Тандем және Node.js кітапханалар жиынтығы желілік ресурстармен жұмыс бойынша асинхронды API жүзеге асырады. Енді сервер белгілі бір процесті сұрау үшін жауапты күтуге тәуелді емес, мысалы, деректер базасынан, ал бұл орындау процесінде болғанша басқа сұраныстармен жұмысты жалғастырады. Бұл интерактивті жүйелер үшін өте үлкен қадам, мұнда клиентте белгілі бір іс-әрекеттерге жауап беру жылдамдығы маңызды.

2.4 CSS стильдер кестесі

CSS-ті түстер, қаріптер, орналасу және құжат көрінісінің басқа да аспектілерін жасау үшін веб-беттерді жасаушылар қолданады. CSS әзірлеудің негізгі мақсаты мазмұнды бөлу (HTML немесе басқа таңба тілінде жазылған) және құжатты ұсыну (CSS-де жазылған) болып табылады. Бұл бөлу құжаттың қол жетімділігін арттырып, оны ұсынудың үлкен икемділігі мен басқаруына мүмкіндік береді, сондай-ақ құрылымдық мазмұндағы күрделілік пен қайталанушылықты азайтуы мүмкін. Сонымен қатар, CSS экрандық көрініс, басып шығару, дауыспен оқу (арнайы дауыстық браузер немесе экраннан оқу бағдарламасы) немесе Брайль шрифтін пайдаланатын құрылғылар шығарған кезде түрлі стильдерде немесе шығару әдістерінде бір құжатты ұсынуға мүмкіндік береді.

CSS стандарты, егер қандай да бір элементке бірнеше ереже бір уақытта сәйкес келсе, мәнерлер ережелері қолданылатын басымдықтарды анықтайды. Бұл ереже үшін басымдықтар немесе "салмақ" есептелетін "каскад" деп аталады.

CSS қасиеттерінің арқасында, біздің веб-қосымшамыздағы қажет емес блоктарды әр түрлі анимациясы бар әдемі, өз орындарына орналастырылған блоктарға айналдыруға болады.

2.5 PostgreSQL мәліметтер базасы

PostgreSQL тек реляциялық емес, объектілік-реляциялық МББЖ. Бұл MySQL, MariaDB және Firebird сияқты ашық бастапқы коды бар басқа SQL мәліметтер базасынан кейбір артықшылықтарды береді. Объектілік-реляциялық мәліметтер базасының іргелі сипаттамасы - бұл пайдаланушы объектілерді және олардың сипаттамасын, оның ішінде мәліметтер типтерін, функцияларын, операцияларын, домендерді және индекстерді қолдау. Бұл Постгресті керемет икемді және сенімді етеді. Сонымен қатар, ол мәліметтердің күрделі құрылымын жасай, сақтай және шығара алады. Төменде кейбір мысалдарда сіз стандартты РМББЖ қолдау емес, салынған және құрамдас конструкцияларды көресіз.

Постгресті қолдайтын мәліметтер түрлерінің кең таралған тізімі бар. Сандық, құбылмалы нүктелі, мәтіндік, буль және басқа да күтілетін мәліметтер типтерінен (сондай-ақ олардың көптеген нұсқалары) басқа, PostgreSQL uuid, акшалай, аударылатын, геометриялық, бинарлық типтерді, желілік адрестерді, биттік жолдарды, мәтіндік іздестіруді, xml, json, массивтерді, композиттік типтерді және диапазондарды, сондай-ақ объектілерді сәйкестендіру үшін кейбір ішкі типтерді қолдаумен мақтана алады. Әділдік үшін MySQL деп айтуға тұрарлық, MariaDB және Firebird де осы мәліметтер түрлерінің кейбірі бар, бірақ тек Постгрес оларды барлық түрін қолдайды. Өйткені Постгрес – бұл объектілік-реляциялық мәліметтер базасы, массивтер мәндер сақталуы мүмкін

көпшілігі үшін қолда бар мәліметтер типтерінің құрылымы. Мұны баған үшін мәліметтер түрінің ерекшелігіне шаршы жақшаларды қосу немесе ARRAY өрнегі арқылы жасауға болады. Массив мөлшері берілуі мүмкін, бірақ бұл міндетті емес.

Егер сіз осы уақыт ішінде кең ауқымды мәліметтер тізімі жеткіліксіз болса, онда сіз құрамдас, тізбектелген, диапазон және базалық мәліметтер сияқты жаңа мәліметтер түрлерін жасау үшін CREATE TYPE командасын пайдалана аласыз.

PostgreSQL көптеген мәліметтерді өңдей алады. Ағымдағы жарияланған шектеулер төменде көрсетілген (2.1-кесте) [9].

2.1-кесте. PostgreSQL шектеулер

Мәліметтер базасының ең үлкен мөлшері	Шектелмеген
Кестенің ең үлкен өлшемі	32 TB
Жолдың ең үлкен өлшемі	1.6 TB
Өрістің ең үлкен өлшемі	1 GB
Кестедегі жолдардың ең көп саны	Шектелмеген
Кестедегі бағандардың ең көп саны	250-1600 бағанның түріне байланысты
Кестедегі индекстердің ең көп саны	Шектелмеген

2.6 Мәліметтер базасын құру

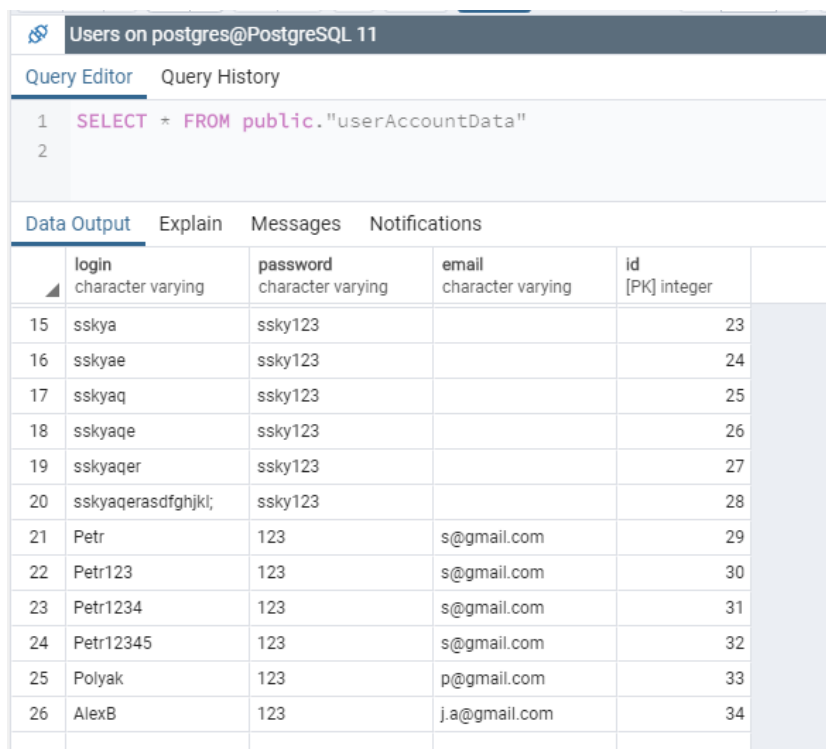
Мәліметтер базасын дұрыс жобалау үшін мәліметтер базасында не және қандай нысанда сақталуы тиіс деп ойластыру қажет. Жұмысты бастау үшін мәліметтер базасының тұжырымдамалық моделін әзірлеу қажет. Кесте атрибуттары тым көп болмауы тиіс, әйтпесе ақпараттың артықтығы болады, ал бұл әр түрлі сұраныстарды орындау үшін өте ыңғайсыз, өйткені мәліметтерді іріктеу және іздеу ұзағырақ жүргізіледі.

Пайдаланушылар мәліметтер базасын жасауды қолға алды, ол пайдаланушылар туралы барлық ақпаратты сақтайтын болады. Осы базада келесі кестелер болады:

- userAccountData, пайдаланушының есептік жазбасының мәліметтері;
- userPersonalData, пайдаланушының жеке мәліметтері;
- userStatisticsData, пайдаланушы статистикасының мәліметтері.
- user Account Data (2.5-сурет) кестесінде мәліметтер пайдаланушы аккаунт туралы ақпарат сақталады. Бұл кесте келесі өрістерге ие болады:
- user Account Id-сандық түрі, кестедегі жазбалардың бірегейлігіне жауап береді;

Жауап беруші: Толыкбаев Алмас Умирбекович Қазақстан Премьер – Министрі Кәрім Мәсімов отандық планшетті компьютерді төрт қазақстандық компаниямен бірлесе отырып өндіру мәселесін жетілдіру жөнінде тапсырма берді;

- password – жолдық тип, пайдаланушының паролі үшін жауап береді;
- email – жолдық тип, пайдаланушының электрондық поштасын сақтауға жауап береді.



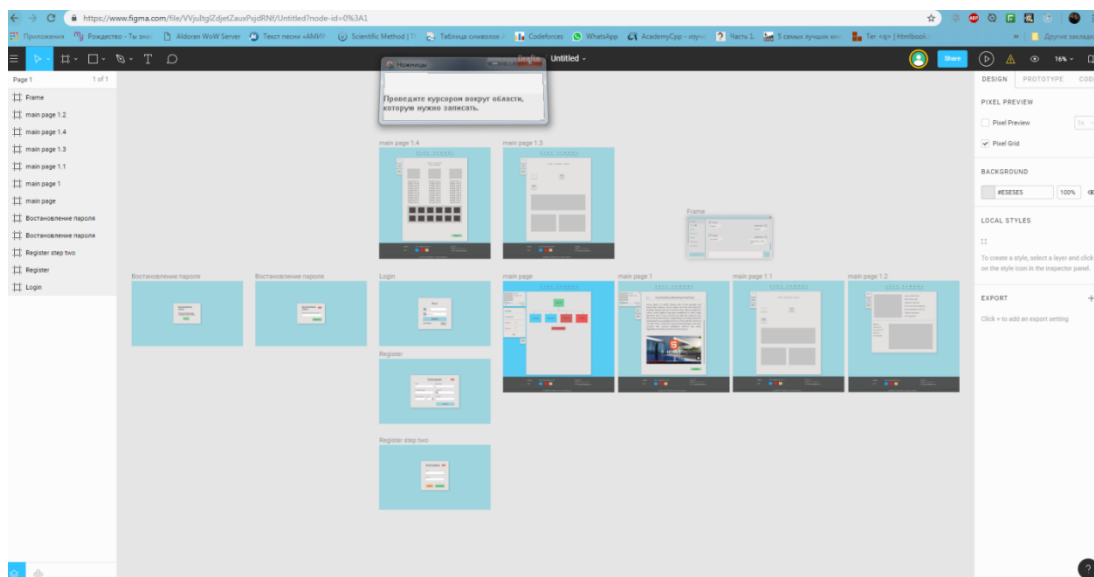
The screenshot shows a PostgreSQL query editor interface. At the top, it says "Users on postgres@PostgreSQL 11". Below that, there are tabs for "Query Editor" and "Query History". The query editor contains a SQL query: `1 SELECT * FROM public."userAccountData"`. Below the query, there are tabs for "Data Output", "Explain", "Messages", and "Notifications". The "Data Output" tab is selected, showing a table with 5 columns: `login` (character varying), `password` (character varying), `email` (character varying), and `id` ([PK] integer). The table contains 12 rows of data, with the first row having an `id` of 23 and the last row having an `id` of 34.

	login character varying	password character varying	email character varying	id [PK] integer
15	sskya	ssky123		23
16	sskyae	ssky123		24
17	sskyaq	ssky123		25
18	sskyaqe	ssky123		26
19	sskyaqer	ssky123		27
20	sskyaqerasdfghjkl;	ssky123		28
21	Petr	123	s@gmail.com	29
22	Petr123	123	s@gmail.com	30
23	Petr1234	123	s@gmail.com	31
24	Petr12345	123	s@gmail.com	32
25	Polyak	123	p@gmail.com	33
26	AlexB	123	j.a@gmail.com	34

2.5-сурет – userAccountData кестесі

2.6-суретте пайдаланушының жеке мәліметтері туралы ақпарат сақталатын кесте келтірілген. Бұл кестеде келесі мәліметтер болады:

- user Personal Id - сандық тип, осы кестедегі жазбалардың бірегейлігіне жауап береді, бастапқы кілт болып табылады;
- user Account Id - сандық тип, user Account Data кестесінің сыртқы кілті;
- name - жол типі, пайдаланушы атын сақтайды;
- surName - жол типі, пайдаланушының атын сақтайды;
- birthdate - күннің типі, пайдаланушының туған күнін сақтайды;
- sex - жол типі, пайдаланушының еденін сақтайды;
- phone - жол типі, пайдаланушы телефонын сақтайды;
- country-жол типі, пайдаланушының тұратын елін сақтайды.



2.7-сурет – Жобамен жұмыс істеуге арналған бет

2.8 Web-бағдарламаның құрылымы

Веб-қосымшаның құрылымын қарастырайық. Бұл веб-платформада бірнеше маңызды бөліктер бар: кіру, негізгі бет, профиль, сондай-ақ пайдаланушыға көмектесе алатын қосалқы компоненттер бар: біз туралы, достар, хабарламалар, байланыстар, тіркелу.

2.8.1 Web-платформада мәліметтерді жүктеу процесі

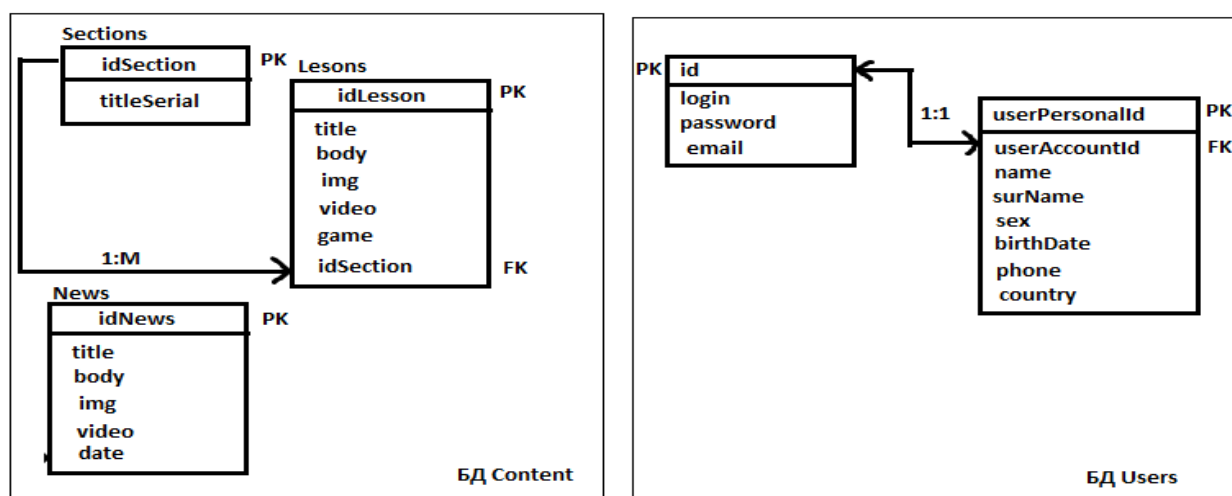
Пайдаланушыға қандай да бір мәлімет беру үшін ақпаратты таңдау платформаның өте маңызды бөлігі болып табылады. Барлық ақпарат түрлі кітап, электронды басылымдардан іріктелетін болады. Ақпаратты орналастыру алдында сайт редакторы өткен материалды толық оқып, зерделеп, іріктейді.

Пайдаланушы оқу материалынан басқа, алынған білімді бекіту үшін түрлі бейнематериалдарды көре алады.

Құрылымдық тәсілдің негізгі идеясы декомпозицияда, яғни жүйені функционалдық кіші жүйелерге бөлу болып табылады. Құрылымдық тәсіл оқыту курстарын шығару бойынша техникалық процесс (ТП) туралы түсінік алу талап етілетін АЖ жобалау кезінде қолданылады. Жүйенің атомдық деңгейге (қарапайым операцияға) дейінгі декомпозициясы абстрагирлеу принципін бұзуы мүмкін, яғни жүйенің маңызды және маңызды емес аспектілерінен алшақтату. Жоғарғы деңгейдегі жүйе контекстік диаграмма болып табылады. Платформаға кіру үшін Интернеттен алынған мәліметтер келіп түседі, шығу нәтижесі ретінде кітаптар оқуға дайын материал алады. Басқару құжаты ретінде "оқытуға жүйелік көзқарас", тетіктер ретінде - сервер,

платформа, серверге қызмет көрсету персоналы және платформа ретінде әрекет етеді.

2.8-суретте Users и Content мәліметтер базасының физикалық моделі келтірілген.

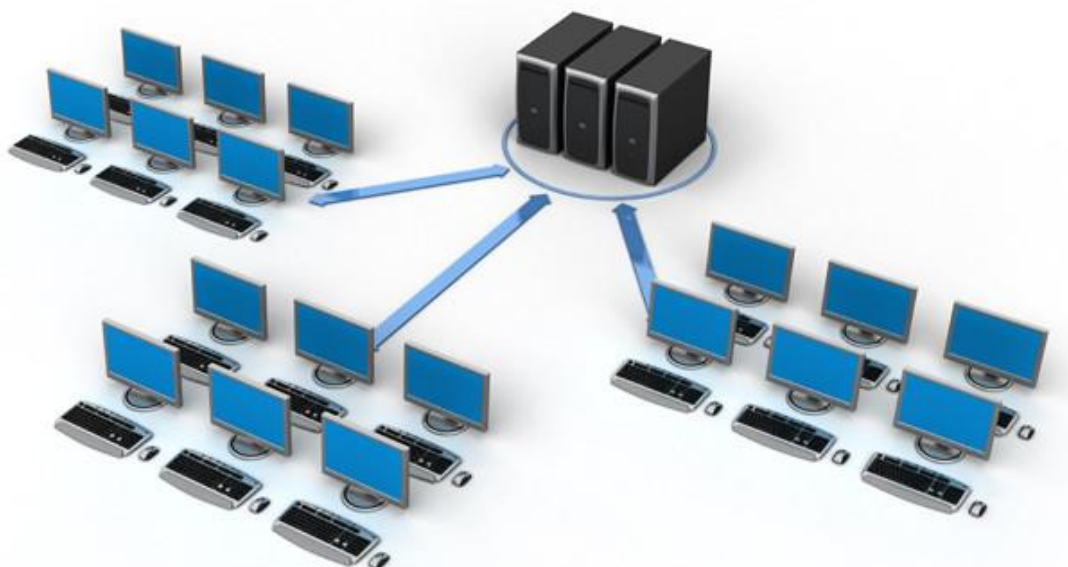


2.8-сурет – Мәліметтер базасының физикалық моделі

3 Балалардың ойлау қабілетін дамытуға арналған web-қосымша құру

3.1 Web-қосымшаның бағдарламалық қамтамасыз етуінің сипаттамасы

Веб-қосымша кездейсоқ таңдалмаған, 21 ғасырда ақпаратты іздеу үшін түрлі техникалық құралдар пайдаланылады. Әр түрлі NATIVE-қосымшалар белгілі бір ОЖ, платформаға сәйкес келеді, оларды "толық клиент" деп атайды, өйткені мұндай қосымшамен жұмыс істеу үшін құрылғының белгілі бір түрі қажет және оны торлық орнату жүргізіледі. Сонымен қатар, интернет желісі үшін Интернет байланысы мен браузері жеткілікті. Бұл тәсіл неғұрлым сенімді болып табылатындықтан көп қолданушылар осы тәсілді пайдаланады. Енді пайдаланушы аппараттық құрылғыға жаңартуларды, сәндеулерді, қосымшаларды орнатудың қажеті жоқ - мұның бәрі сервер жағында болады (3.1-сурет).



3.1-сурет – Жұқа клиент үлгісі

Сонымен қатар веб-қосымшаны ыңғайлы пайдалану үшін HTTP хаттамасы бойынша қарым-қатынас жасау керек болса, клиент өзінің техникалық құрылғысына осы тізімнен кез келген браузерді орнату қажет: Google Chrome нұсқасы 10.0 немесе одан жоғары, Яндекс браузері 12.0 және одан жоғары. Мұндай браузерлерді: Internet Explorer, Edge, Mozilla firefox, Amigo және т.б. пайдалану ұсынылмайды. JavaScript әр браузерге енгізілген болса да, бағдарламалау тілі, барлық жаңалықтар жоғарыда аталған қолдау емес браузерлер арқылы беріледі.

3.1.1 Клиентпен өзара әрекеттесу құрылымының жалпы мәліметтері

Бұл веб-бағдарлама клиент-серверлік қосымшаның парадигмасы бойынша жұмыс істейтін болады, бұл Http-сұраныс серверге жіберілуін және одан жауап күтуді білдіреді (3.1-диаграмма). Frontend және backend бөліктерін бөлу керек болады, екіншісін істен шыққан кезде серверлердің бірінің тұрақты жұмыс істеуі үшін.

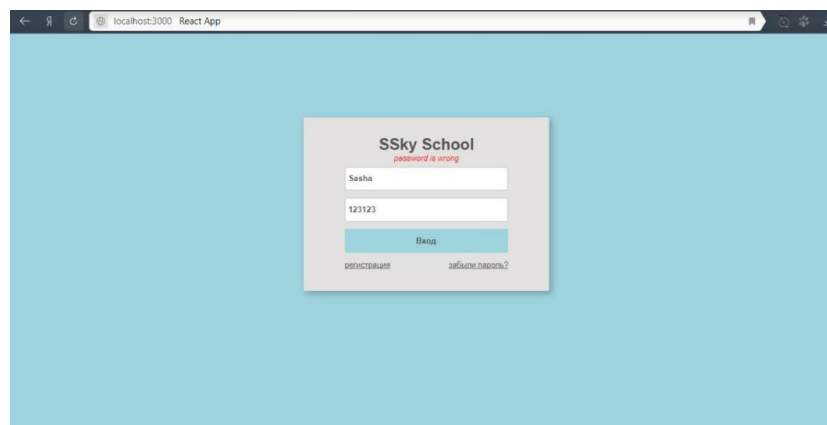
1. пайдаланушы браузер адресіне <http://sskyschool.kz/login> енгізеді;
2. пайдаланушының жеке компьютері (браузері) кіру бетін алу үшін front-серверге HTTP-сұраныс жібереді. Html-бетті жүктегеннен кейін, пайдаланушы кіру мәліметтерін енгізіп, "Кіру" батырмасын басады;
3. front-сервер авторландыру мәліметтері бар back-серверге сұраныс жібереді;
4. back-сервер PostgreSQL мәліметтер базасымен байланысты орындайды, процедураны шақыру;
5. мәліметтер базасы пайдаланушының енгізілген деректерінің болуын тексереді және жауапты қайтарады;
6. back-сервер мәліметтер базасынан жауапты өңдейді, құрылымды қалыптастырады және жауапты front-серверге жібереді;
7. front-сервер жауапты алады, өңдейді және клиенттің жеке компьютеріне жібереді немесе басты бетке өтеді немесе қате туралы хабарлама шығарады;
8. жеке компьютер front-серверден алынған ақпаратты көрсетеді.

Екі сервер бір-біріне байланысты жұмыс істейтінін байқауға болады. Қолданба сервері істен шыққан кезде пайдаланушы Html-беттерін ала алады, бірақ деректерді таңдау жүзеге асырылмайды, клиент оған түсінікті түрде қате туралы хабар алады.

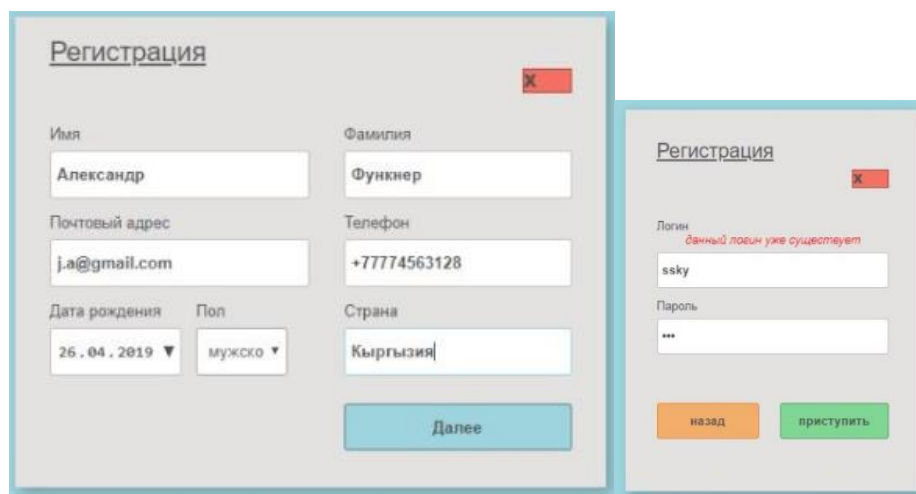
3.2 Қосымша прототипін жобалау

Прототипті, дизайнды жобалау үшін қолданба "Figma" жұқа клиентіне негізделген. Бұл бағдарламалық жасақтаманы браузерде де, жеке компьютерде немесе ноутбукте жергілікті іске қосуға болады. Figma әр түрлі фигураларды жасау үшін векторлық графиканы қолданады, бұл кескіндерді нақтылығын жоғалтпай кішірейтуге мүмкіндік береді.

Бағдарлама дизайны өте маңызды кезең болып табылады, өйткені ол пайдаланушыны осы өніммен жұмыс істеуге тартады. Бірінші болып кіру (3.2-сурет) және тіркелу жобаланған (3.3-сурет).



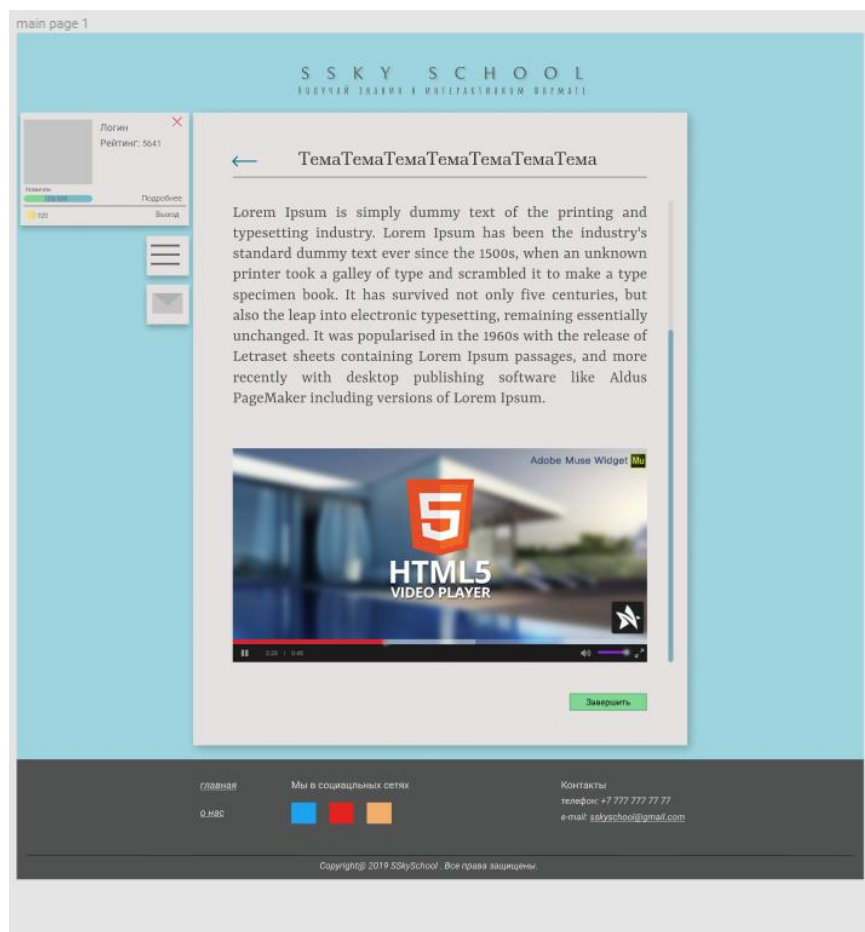
3.2-сурет – Веб-бағдарлама бетіне кіру прототипі



3.3-сурет –Веб-платформада тіркеудің 2 кезеңі

Бұл түсті гамма түсті психология тұрғысынан таңдалды. Көгілдір түс сенім мен тыныштықты білдіреді, көзді тітіркендірмейді.

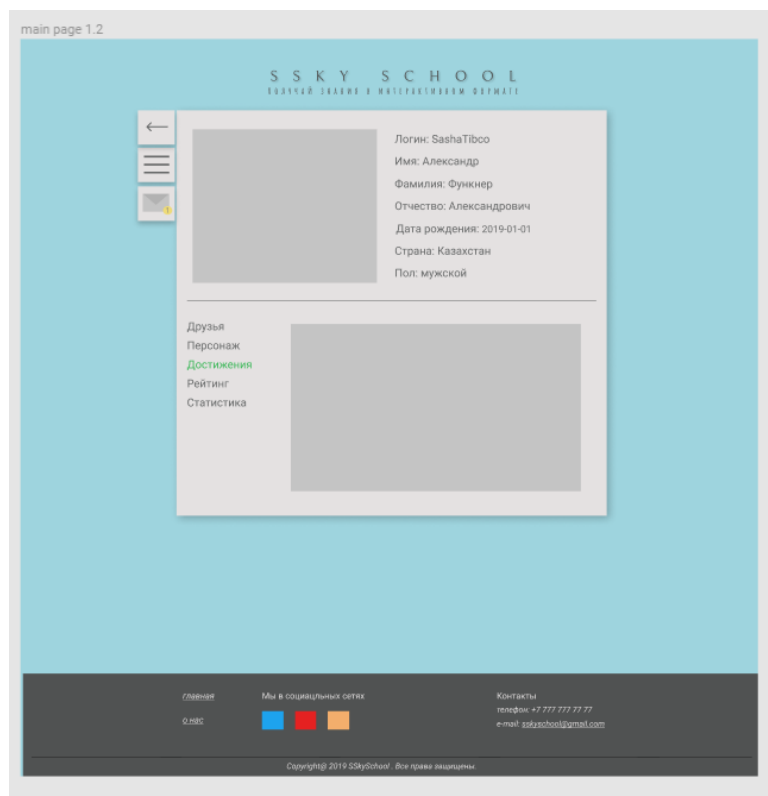
Сонымен қатар, веб-қосымшамен өзара әрекеттескен кезде, пайдаланушының динамикалық мәліметтерімен бір беті болады (3.4-сурет).



3.4-сурет – Веб-қосымшаның негізгі беті

Сол және оң жақ шеті бойынша әр түрлі элементтерді табу кезінде пайдаланушы назар, концентрацияны жоғалтады. Сондықтан веб-қолданбаның пайдаланушысы беттің сол жағындағы қосымша, қосымша элементтерді жасыра алады. Сондай-ақ, бет әдеттегі веб-платформаларда орын ретінде төмен айналмайды, ақпарат өлшемін өзгертпейтін басты блоктың ішінде айналуға болады.

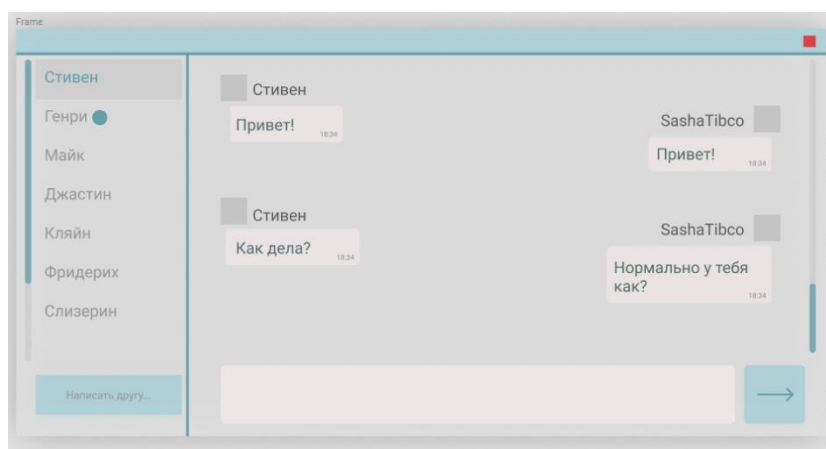
Ол үшін келесі бет қажет болады (3.5-сурет). "Жетістіктер" бөлімі бар екенін байқауға болады. Бұл бөлім пайдаланушыларды әртүрлі жарыстарға қатысуға, материалды зерттеуге және т.б. ынталандыратын болады.



3.5-сурет – Пайдаланушы туралы ақпаратты көрсету

Осы веб-қосымшаның әлеуметтік бөлігі пайдаланушыға достармен, таныстармен өзара іс-қимыл жасауға көмектеседі, жаңа достықты, танысуды бастайды, бұл құрамдас бөлік әр түрлі ақпаратпен қарым-қатынас жасауға және алмасуға мүмкіндік береді.

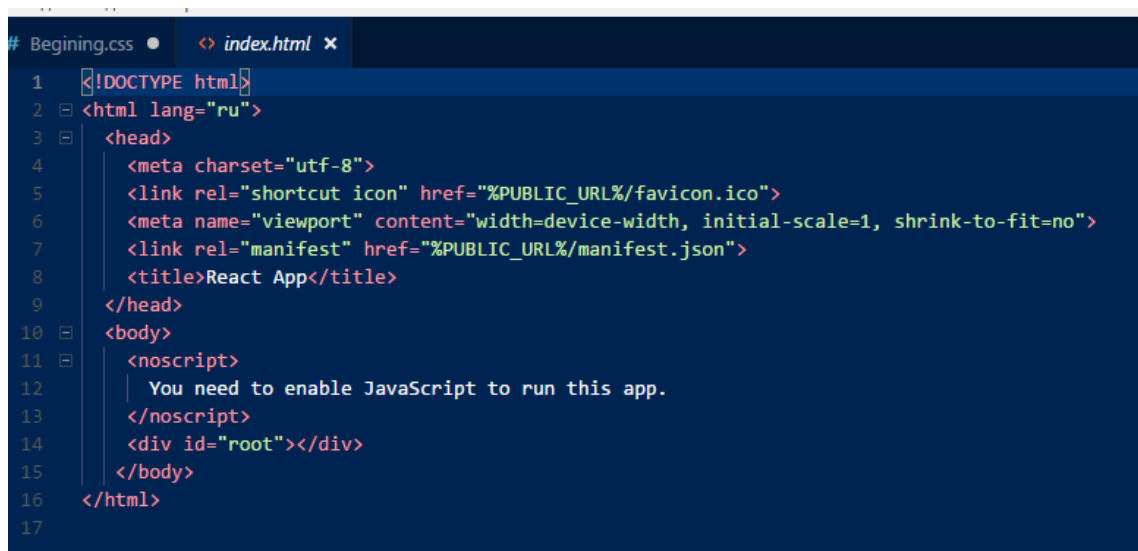
Қолданушылардың өзара әрекеттесу компоненті бөлек терезеде, барлық қалған материалдың үстінен болады. Дизайн негізгі веб-бағдарлама стилінде әзірленген, оның шағын нюансы бар, ол мәлдірлік пен жеңілдіктің әсерін сезінуге мүмкіндік береді (3.6-сурет).



3.6-сурет – Пайдаланушылардың байланыс прототипі

3.3 SPA әзірлеу

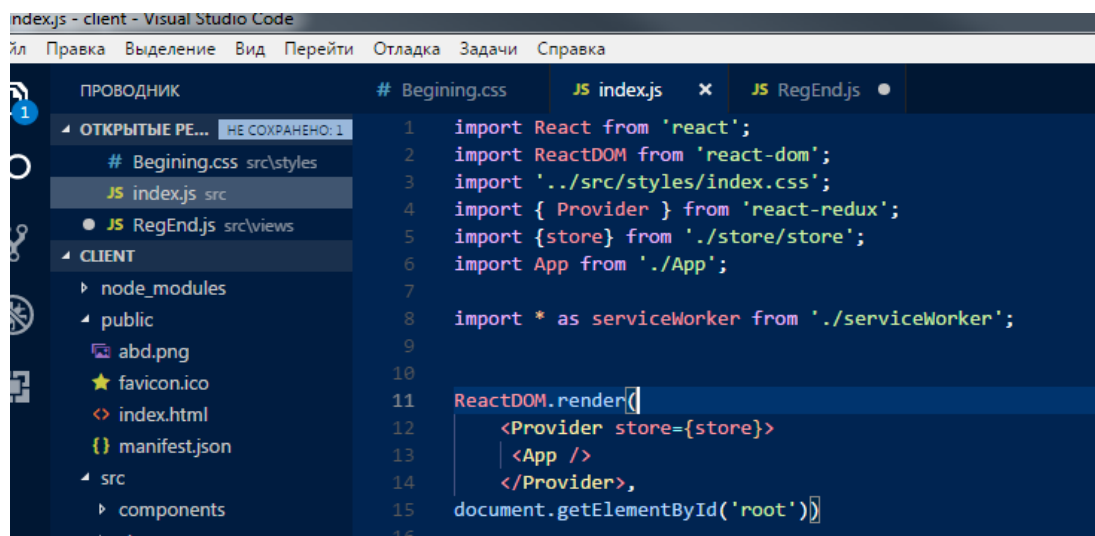
SPA-single page application (орыс. бір бетті қосымша) клиент-серверлік қосымшаның әмбебап және жақсартылған өзара әрекеті болып табылады. Барлық қолданба бір html құжатында орналасқан (3.7-сурет). Бұл тұжырымдаманың өз артықшылықтары мен кемшіліктері бар. Жаңа беттердің сұраныстарына сервердің жауабын плюстерге жатқызуға болады. Минустарға іске асыру қиындығын жатқызуға болады.



```
# Beginning.css • < index.html x
1 <!DOCTYPE html>
2 <html lang="ru">
3   <head>
4     <meta charset="utf-8">
5     <link rel="shortcut icon" href="%PUBLIC_URL%/favicon.ico">
6     <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
7     <link rel="manifest" href="%PUBLIC_URL%/manifest.json">
8     <title>React App</title>
9   </head>
10  <body>
11    <noscript>
12      You need to enable JavaScript to run this app.
13    </noscript>
14    <div id="root"></div>
15  </body>
16 </html>
17
```

3.7-сурет – Негізгі html құжат скриншоты

Бұдан әрі html құжатымен жұмыс істеу үшін екі файл құру қажет .js, негізінен index файлында. React жұмыс істеу үшін барлық қажетті модульдерді тарту қажет, React-функцияны пайдалана отырып, барлық қалған мазмұнды орналастыратын html құжатының id-ін көрсету қажет (3.8-сурет).



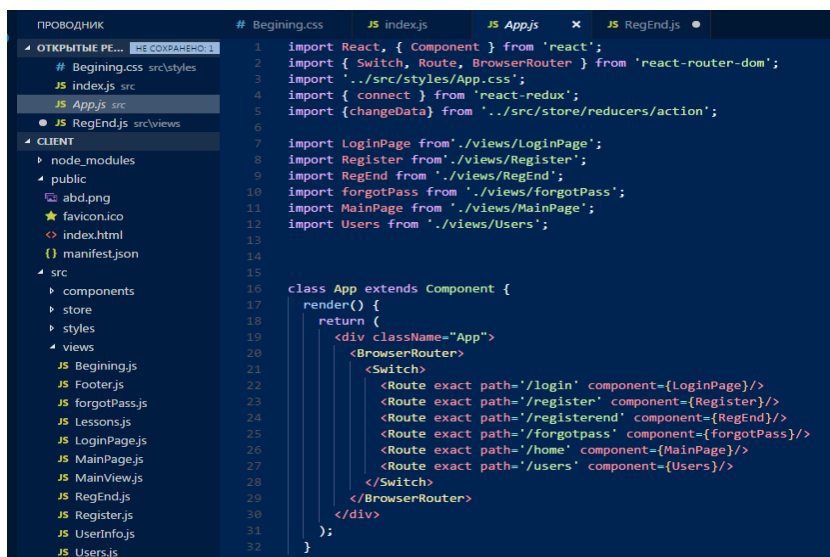
```
index.js - client - Visual Studio Code
Фл  Правка  Выделение  Вид  Перейти  Отладка  Задачи  Справка

ПРОВОДНИК
ОТКРЫТЫЕ РЕ... НЕ СОХРАНЕНО: 1
# Beginning.css src/styles
JS index.js src
JS RegEnd.js src/views
CLIENT
node_modules
public
abd.png
★ favicon.ico
<> index.html
{} manifest.json
src
components
store

# Beginning.css JS index.js x JS RegEnd.js •
1 import React from 'react';
2 import ReactDOM from 'react-dom';
3 import '../src/styles/index.css';
4 import { Provider } from 'react-redux';
5 import { store } from '../store/store';
6 import App from './App';
7
8 import * as serviceWorker from '../serviceWorker';
9
10
11 ReactDOM.render(
12   <Provider store={store}>
13     <App />
14   </Provider>,
15   document.getElementById('root'))
16
```

3.8-сурет – javaScript файл index.js

Енді файл келесі файл (бұдан әрі компонент) app.js-ты, сонымен қатар барлық react'a кітапханаларын және біздің меншіктегі әзірленген компоненттерімізді жүктеу қажет. Әр түрлі компоненттің көрсетілуін таңдау үшін "route" элементінде "Switch" және "BrowserRouter" модульдерін пайдаланамыз және ол үшін браузерде пайдаланушы енгізетін жолды және осы жолда көрсетілуі тиіс компонентті көрсету қажет (3.9-сурет).



3.9-сурет – javaScript файлы «App.js»

Енді HTML құжатымен өзара байланысу деңгейі дайын, енді оларды экранда көрсету үшін дайындаған компоненттерді әзірлеу қажет. Кіріс бетінен бастап, элементтерді көрсету үшін кодтың негізгі конструкциясына қарап шығамыз (3.10-сурет).



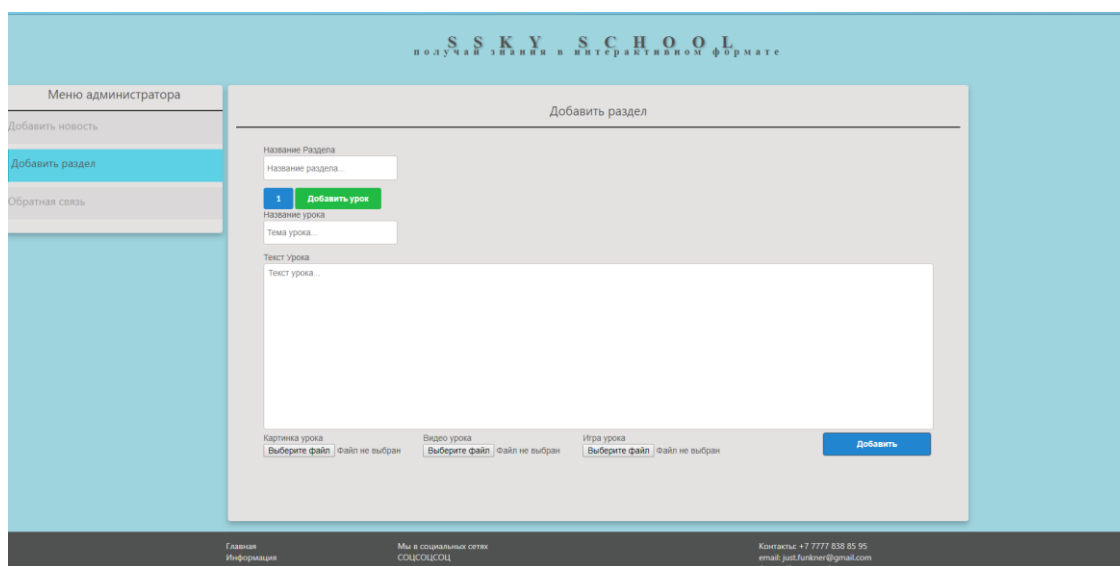
3.10-сурет – Кіру бетін көрсету коды

Сонымен қатар, ол html-ді JavaScript-ті пайдаланып, қарапайым html белгілеуді қолданатынын байқауға болады, бұл React ерекшеліктерінің бірі болып табылады, ол html белгілеуді javascript кодқа түрлендіреді, бұл компонентті жасауды жеңілдетеді.

Виртуалды DOM жасалады, онда біздің біріктірілген компоненттеріміз көрсетіледі және DOM html құжатына көшіріледі.

Бұл компоненттің күрделілігінің бірі серверге сұраныс жасау болып табылады. Ол үшін функцияны жазамыз (3.11-сурет), ол Promise-пен өзара әрекеттеседі, промис callback функциясы болып табылады және оның екі нәтижесі бар, бірі жауап алады немесе жоқ. Бұдан әрі серверден жауап өңдеу логикасы жүреді, егер http-қосылым мәртебесі 200 мәні болса, онда біз жауапты сұрайтын боламыз және пайдаланушы туралы ақпарат сақталатын басты бетке өтеміз. Әйтпесе, пайдаланушыға қате туралы хабарлама шығады және бұл қателіктің себебі ол F12 пернесін басып, консоль терезесін аша отырып, браузер консоліне қарай алады.

Барлық ақпаратты JSON форматында таратамыз, өйткені ол желі бойынша ақпарат беру үшін ең жақсы құрылым болып табылады. Ол қиын форматқа ие емес, және JavaScript үшін деректермен өзара әрекеттесудің ыңғайлы нысаны болып табылады.

The image shows a web application interface for 'SSKY SCHOOL'. The header has the school's name and tagline. A sidebar on the left contains a 'Menu administrator' and links for 'Add news', 'Add section', and 'Feedback'. The main content area is titled 'Add section' and contains a form with the following fields: 'Section Name' (with a placeholder), 'Lesson Name' (with a placeholder), 'Lesson Topic' (with a placeholder), and 'Lesson Text' (a large text area). Below these fields are three file upload sections: 'Lesson Image', 'Lesson Video', and 'Lesson Audio', each with a 'Choose file' button and a status 'File not selected'. A 'Add' button is at the bottom right of the form. The footer contains links for 'Main Information', 'Social Networks', and 'Contact Us' with a phone number and email address.

3.11-сурет – Басты бетке кіру функциясы

Тіркеу бөлімін қарастырайық. Тіркеу кезеңі екі кезеңге бөлінген, пайдаланушы туралы негізгі ақпаратты енгізу, бетке кіру үшін деректерді енгізу. Бірінші кезеңде біз жергілікті браузер сақтаушысымен өзара әрекеттесеміз, серверге сұраныс жібергенге дейін пайдаланушы туралы ақпаратты сақтауымыз қажет.

Осы компонентке оралған кезде, онда componentDidMount өмірлік циклінің функциясы орнатылған, соның арқасында беттің жаңа рендерингінде,

құрамдауышты көрсету алдында жергілікті қоймадан деректерді тартып, оларды пайдаланушы деректерін енгізу элементтеріне қоятын боламыз.

Пайдаланушыны тіркеуді аяқтау бетіне ауыстыратын функция пайдаланушы енгізген деректерді жинайды және өзінің жергілікті қоймасын жаңартады, бұл пайдаланушы беттен басқа бетке кездейсоқ көшкен кезде өзінің енгізілген деректерін жоғалтпауы үшін жасалады. Пайдаланушылар деректерін сақтау, веб-платформаның өте маңызды құрамдас бөлігі болып табылады.

Бұдан әрі ақпаратты толтыру кезінде пайдаланушы кіріс туралы ақпаратты енгізетін тіркеуді аяқтау бетіне өтеді, "кірісу" батырмасын басқан кезде пайдаланушы серверге сұрау жібереді және жауапты күтеді, тіркеу функциясын қарастырамыз.

Функция `promise` негізінде жұмыс істейді, серверге жіберу үшін, алдымен біз сұрау денесін қалыптастырамыз, белгілі бір өрістің `JSON`-объектісін жинаймыз. `Post` әдісімен `http` сұрау жібереміз және біз жоғарыда жинаған объектімізді жолдаймыз.

Пайдаланушыны құрғаннан кейін және басты бетке кіргеннен кейін, пайдаланушы туралы барлық ақпарат браузердің жергілікті қоймасынан, `token` объектісінен жүктеледі, ол біз жаңа пайдаланушыны кіргенде немесе регистрация жасаған уақытта шығатын болады. Бұл ДБ жиі шақыруларын болдырмау және қолданбалар серверін түсіру үшін жасалған.

Тіркеу кезінде кіріс `login` параметрінің бірегейлігіне тексеру жүргізіледі, яғни осы параметрді қайталаған кезде пайдаланушыға осындай `login` 'ом бар, пайдаланушы бар деген қате туралы хабарлама шығады.

3.4 Серверлік бөлікті әзірлеу

Серверлік бөлікті әзірлеу үшін `Node.js` және `Express` модулін пайдаланамыз. `Express`-бұл сервермен өзара әрекеттесуді жеңілдететін кітапхана.

Бастау үшін, бізге жергілікті сервер құру қажет портта 3001, өйткені порт 3000 бос емес, сервер қолдануда, ол қосымшаның алдыңғы бөлігін өңдеумен айналысады. Серверді дұрыс жұмыс істеу және іске қосу үшін, біз `POST`, `GET` сұраныстарды өңдеу үшін жасаған барлық қажетті модульдерді және модульдерді жүктеп алуымыз керек.

Сервермен өзара іс-қимылда біз `json` нысанын пайдаланамыз, себебі ол `body http`-сұраудың стандартты түрі болып табылмайды.

Сервермен не болып жатқанын түсіну үшін, біздің қосымшалар серверіне келетін кез келген оқиғаларды ұстап тұратын логгерді орнату қажет.

Сервермен өзара байланыста біз `json` объектісін пайдаланамыз, себебі ол `body http`-сұраудың стандартты түрі болып табылмайды.

Сервермен не болып жатқанын түсіну үшін, біздің қосымшалар серверіне келетін кез келген оқиғаларды ұстап тұратын логгерді орнату қажет.

Белгілі бір конфигурациясын құрылғаннан кейін, объект құру қажет, ол белгілі бір портты тындап тұрады.

Алдыңғы бөлім жағынан келетін сұраныстарды өңдеуді қарастырайық. Бірақ, ең алдымен, біздің базаға қосылуын қарастыру қажет. Базаға қосылу үшін PG модулін орнату қажет, ол PostgreSQL мәліметтер базасына қосылу үшін қажет. Содан кейін жаңа объект құру қажет және оған пайдаланушы, хост, мәліметтер базасының өзін, пароль және портты көрсете отырып, біздің базаға қосылу қасиеттерін қою қажет (3.12-сурет).

```
1  const pg = require('pg');
2  var pool = new pg.Pool({
3    user: 'sskySchool',
4    host: '127.0.0.1',
5    database: 'Users',
6    password: 'ssky1352',
7    port: '5432'});
8
9  module.exports = pool;
```

3.12-сурет – МБ қосу үшін объектіні құру

Мәліметтер базасына сәтті қосылғаннан кейін, біздің қосымшалар сервері үшін post-сұрау тындаушысын құру қажет. Басты сұраныстардың бірі пайдаланушының қосылуы, авторизациялау әрекеті кезінде енгізілген мәндерді тексеру.

Пайдаланушыдан мәліметтерді алған кезде, одан МБ-на сұраныс жасаймыз және мәліметтерді авторизация кезінде тексереміз, егер процедуралар бізге failed-ге тең жауап қайтарса, онда біз пайдаланушыға дұрыс емес мәліметтерді енгізгені туралы хабарлайтын боламыз, әйтпесе ол басты бетке өту үшін рұқсат алады және token алады, соның арқасында өзіңіз туралы ақпарат алатын болады.

Мәліметтердің өзара байланысын жеңілдету үшін деректер қорында token типін құру шешілді, онда әр түрлі кестеден тұратын жарамды ақпарат сақталады және пайдаланушыға беріледі (3.13-сурет).

token

General Definition Security SQL

Type: Composite

Composite Type					
	Member Name	Type	Length/precision	Scale	Collation
🗑	login	text			Select from the
🗑	email	text			Select from the
🗑	name	text			Select from the
🗑	surname	text			Select from the
🗑	sex	text			Select from the
🗑	birthdate	date			
🗑	phone	text			Select from the
🗑	country	text			Select from the

ⓘ ?
 Cancel
Reset
Save

3.13-сурет – Жекеменшік типін құру

Сонымен қатар, ол деректер қорының процедурасын шақыратын және табыстылық немесе бас тарту туралы белгілі бір жауапты алатын пайдаланушының тіркеу оқиғаларын өңдеуге жаңа сұраныс жасайды (3.22 сурет). Жауап JSON форматында, ол ақпаратты беру құрылымы бойынша ең қолайлы және қарапайым болып табылады.

3.4.1 Онлайн чат құру

Веб-қосымшаның идеяларының бірі - бір-бірімен қарым-қатынасты қолдау үшін онлайн чат құру болды. Бұл мақсат үшін socket.io модулін жүктеу қажет, соның арқасында онлайн режимінде түрлі форматтағы сұраныстарды жіберуге және алуға болады. Тыңдаушыны біздің сервер портына баптау қажет, консольге әрбір қосылғаннан кейін пайдаланушы сессияға қосылғаны туралы ақпарат жазылады. Өңдеу оқиғаларын құру хабарлама жіберу бойынша, функцияны құру, уақытты алуға және қабылдаған хабарламаны тарату адамдар тобына не белгілі бір тұлғаға беріледі.

Сонымен қатар, алдыңғы жақ бөліміне, хабарлардың көрсетілуіне жауап беретін компоненттердің жай-күйі жаңа хабар туралы оқиға алған кезде өзгеретінін көрсету қажет.

Сонымен қатар, мәліметтерді жіберу үшін, пайдаланушы енгізген мәтінді таңдап, socket.io-client модулінің көмегімен жіберетін функцияны құру қажет, ол біздің тыңдаушының серверлік чатпен өзара әрекеттеседі.

3.5 "Дақ" ойының құру

"Дақ" ойынын дамыту үшін HTML 5 жаңартылуы бар HTML жақында қосылған canvas технологиясын пайдаланамыз.

Біздің экранда блоктарды көрсету үшін әмбебап функцияны жазамыз (3.14-сурет).

```
123 function draw(block){
124     //console.log(block)
125     if(block.num != 0){
126         ctx.beginPath()
127         ctx.rect(block.x,block.y,width - 4,height - 4)
128         ctx.fillStyle = block.color
129         ctx.fill()
130         ctx.fillStyle='black'
131         ctx.stroke()
132         ctx.textAlign = 'center'
133         ctx.font = '32px Georgia'
134         ctx.fillText(block.num,block.x + width*0.4 +7,block.y + height/2 + 7)
135         ctx.closePath()
136     }
137 }
```

3.14-сурет – Блоктарды көрсету функциясы

Объектімен жұмыс істеу үшін, класс құру қажет (3.15-сурет), онда түстер, орналасуы x ережесі бойынша және y ережесі бойынша, блок нөмірі сақталады. Класты құру үшін функционалдық бағдарламалау мүмкіндігін қолданамыз.

```
77 function block(x,y,num,color ='red'){
78     this.x = x
79     this.y = y
80     this.num = num
81     this.color = color
82 }
```

3.15-сурет – Блоктар класын құру

Енді біздің кластың нысандарын құру және массивке орналастыру қажет. Ол үшін циклді және PUSH массивін қолданамыз.

Біздің массивте элементтерді араластыру үшін 0-ден 15-ке дейін кездейсоқ санды қайтару функциясын және swar функциясын қолданамыз, ол өзіне берілген екі нысанның нөмірлерін өзгертеді.

Блоктарды құру функциясын шақырғаннан кейін біз келесі суретті аламыз (3.16-сурет). Шын мәнінде, canvas тегін пайдалану кезінде біз жай ғана біздің блоктарды жылжыту үшін, біздің кенепте тінтуірдің сол жақ батырмасының басуын қадағалау қажет, бірақ браузер бетінде тінтуірді бақылау жоғарғы сол жақ жиегінен орын алады, біздің ойын түсірілетін блоктың қашықтығын, браузердің жоғарғы және сол жақ жиегінен алып тастау қажет.

1	2	3	4
5	6	7	8
9	10	11	12
13	15	14	

3.16-сурет – Блоктардың суреті

Енді біз тінтуірдің сол жақ батырмасының қысу оқиғасын бақылап, кенепті тазалағаннан кейін, қандай блоктарға кіргенімізді тексеріп, бос блоктарды суретке түсіріп, жеңіске жету жағдайын тексеріп отыратын функцияны қосу қажет.

Жеңіс шарттарын тексеру үшін біз функцияны жазамыз, ақиқат жағдайын қайтарамыз, ойынның шашыраңқы нұсқаларының бірін жинаймыз. Бұл ойында бос ұяшық төмен және оң жақта немесе жоғарғы және сол жақта болған кезде жеңудің екі нұсқасын ала аламыз.

Сонымен қатар функцияны құру қажет, ол тышқанның басу аймағын қадағалайтын болады, бұл анықтау аймағы арқылы жүзеге асырылады, яғни, тышқанды басу массивтің арқасында біздің элемент орналасқан аймақта болды ма, біз әрбір блоктың жағдайын сақтаймыз. Тексергеннен кейін, ұяшықтың жанында бос орын болса, элементтердің жылжуын қосу қажет. Сондай-ақ, біздің блоктар шеткі жағдайда, ең жоғарғы және ең төменгі жағдайларда, оқиғалар арқылы өңдеу қажет болады. Егер кенеттен біз басқан блоктардың бірі бос ұяшықтың жанында тұрса, онда біз swap функциясын шақыруымыз керек, ол блоктардың орнын ауыстырады және элементтердің жылжу саны үшін жауап беретін ауыспалы жолды жаңартады.

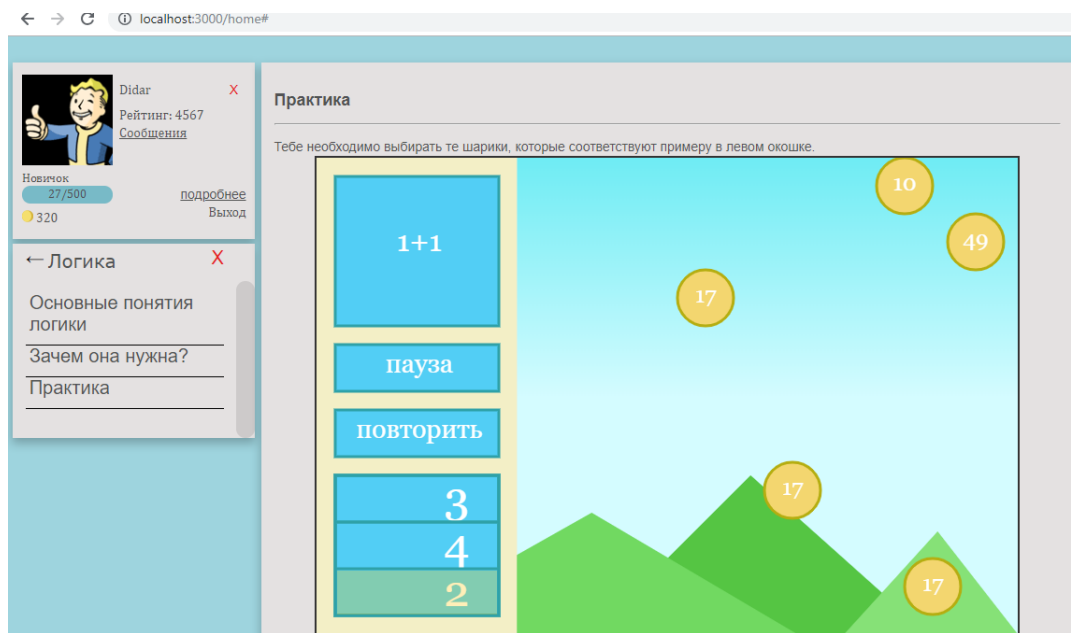
Кодпен жұмыс жасаудың барлық манипуляциясынан кейін қолданушының логикалық ойлауын дамытатын толыққанды ойын аламыз. Бұл ойын 1878 жылы ойлап табылған және әлі күнге дейін ойлау мен логиканы дамыту бойынша үздік ойындардың бірі болып табылады.

Стильдерді қосқаннан кейін біз келесілерді аламыз (3.17-сурет).



3.17-сурет – Ойынның соңғы көрінісі

Тағы бір құрастырылған ойын логиканы дамытуға арналған қарапайым ойын болып табылады. Мұнда ең алдымен жылдамдық және дұрыс жауапты табу болады (3.18-сурет).



3.18-сурет – Логиканы дамыту ойынының көрінісі

ҚОРЫТЫНДЫ

Ақпараттық технологиялардың соңғы уақыттағы даму үрдісі адам өміріне көптеген жаңалықтарды, мол мүмкіншіліктерді, жетістіктері мен нәтижелерін жедел түрде, қолма-қол пайдалана алуға мүмкіндік туғызатын жаңа жүйелерді, нысандарды енгізуге мүмкіндік береді.

Қазіргі уақытта балалар мен жасөспірімдерді оқытудағы жұмыс тәжірибесі арқылы білім алуға таңдалған тәсіл өте қолайлы деп есептеуге болады. Көптеген балалар, жасөспірімдер интернетті және компьютерді пайдаланады, бірақ пайдалы ақпарат алмайды, өз уақытын бос өткізеді.

Қазір біздің еркін және бақытты болашаққа қарсы ақпараттық күрес жүргізіліп жатыр, сондықтан біз өзіміздің ақпараттық кеңістігімізді алып, оны дұрыс арнаға бағыттап, оны мүлдем тегін жасауымыз қажет, өйткені біздің өміріміз балаларымыздың және болашақ ұрпақтың қолында, ал болашақ ұрпақ дұрыс білім алмаса біз болашаққа нық сеніммен ене алмаймыз.

Көптеген мектептерде зертханалық құрылғылар, химиялық реагенттер, биологиялық компоненттер жоқ, әртүрлі зертханалық жұмыстарды жүргізу үшін баланың белгілі бір салада немен айналысқысы келетінін түсіну үшін егер мектепте қарапайым химикаттармен, электр құрылғыларымен, микроскоптармен және т. б. жұмыс істеу үшін интернет арқылы материалдарды көру арқылы жасауға болады. Осы тәжірибені алу үшін заманауи технологияларды, интернетті, қосымшаларды, виртуалды шындықты қолдануға болады. Бізде осындай құрылымды құруға мүмкіндік, білім және күш те бар, бірақ ол үшін көп еңбектену қажет болады.

Бұл дипломдық жұмыста халыққа сапалы әрі тиімді қызмет көрсету мақсатында балалардың ойлау қабілетін дамытуға арналған веб-қосымшасы құрылды. Веб-қосымшаның мақсаты балалардың ойлау қабілетін дамыту мақсатында оларға материалды ойын түрінде көрсету әрі оның сапасын арттыру болып табылады. Ары қарай интернет технологиясына шолу жасалып, ақпараттық жүйені құру технологиясы жасалды. Бұл веб-беттегі ақпараттық жүйе HTML тілінде Node.js технологиясын қолданып Visual Studio Code ортасында PostgreSQL мәліметтер базасымен байланыстыра отырып, мультимедиялық веб-қосымша құрылып отыр.

Қорыта келгенде, осы жазылған дипломдық жұмыстың нәтижесінде HTML, JavaScript тілдері мен PostgreSQL дерекқорын қолдану арқылы берілген жұмыс орындалып, сенімді дерекқор мен желідегі ең танымал программалау тілінің өзара қатынасқан жұмысы жақсы нәтижеге жеткізілді.

Бұл дипломдық жұмыс қазіргі уақыттағы өзекті проблеманы қозғайды, өйткені білім алу мәселесі әр мемлекет үшін аса маңызды болып табылады.

ПАЙДАЛАНЫЛҒАН ӘДЕБИЕТТЕР ТІЗІМІ:

- 1 <https://postnauka.ru/about>
- 2 <https://zarabotat-na-sajte.ru/wiki/html.html>
- 3 <https://learn.javascript.ru/intro>
- 4 <https://xbsoftware.ru/blog/pochemu-stoit-ispolzovat-react-js-razrabotke-prilozhenij/>
- 5 https://zharikov.site/note/shortreview_nodejs.html
- 6 <http://figmadesign.ru/1-0-0-obzor-figma.html>
- 7 <https://spark.ru/startup/pixli/blog/30048/figma-moschnij-instrument-dlya-komandi-dizajnerov>
- 8 Node.js в действии. 2-е изд. — СПб.: Питер, 2018. — 432 с.: ил. — (Серия «Для профессионалов»).
- 9 Климов, Александр JavaScript на примерах / Александр Климов. - М.: "БХВ-Петербург", 2017. - 336 с.
- 10 Гаевский, А.Ю. 100% самоучитель. Создание Web-страниц и Web-сайтов. HTML и JavaScript / А.Ю. Гаевский, В.А. Романовский. - М.: Триумф, 2018. - 464 с.
- 11 Дронов, В. JavaScript в Web-дизайне / В. Дронов. - М.: СПб: БХВ, 2011. - 880 с.
- 12 Лазаро, Исси Коэн Полный справочник по HTML, CSS и JavaScript / Лазаро Исси Коэн, Джозеф Исси Коэн. - М.: ЭКОМ Паблишерз, 2017. - 120 с.
- 13 Хэррон, Дэвид Node.js Разработка серверных веб-приложений на JavaScript / Дэвид Хэррон. - М.: ДМК Пресс, 2014. - 144 с.
- 14 Сухов, Кирилл Node.js. Путеводитель по технологии / Кирилл Сухов. - М.: ДМК Пресс, 2015. - 416 с.
- 15 Изучаем Node.js. - М.: Питер, 2013. - 400 с.
- 16 Ригс, Саймон Администрирование PostgreSQL 9. Книга рецептов / Саймон Ригс, Ханну Кросинг. - М.: ДМК Пресс, 2015. - 364 с.
- 17 Уорсли, Дж. PostgreSQL. Для профессионалов (+ CD) / Дж. Уорсли, Дж. Дрейк. - М.: СПб: Питер, 2002. - 496 с.
- 18 Фрейен Бен HTML5 и CSS3. Разработка сайтов для любых браузеров и устройств; Питер - , 2014. - 304 с.
- 19 Хоган Брайан HTML5 и CSS3. Веб-разработка по стандартам нового поколения; Книга по Требованию - , 2012. - 272 с.
- 20 Mark Pilgrim HTML5: Up and Running; O'Reilly Media, 2010.- 224 с.

А қосымша

LoginPage.js

```
import React, { Component } from 'react';
import '../styles/LoginPage.css';
import { connect } from 'react-redux'
import {rootReducer} from '../store/reducers/reducer'
import {changeData} from '../store/reducers/action'
/*import {store} from '../reducers/reducer'
import {changeData} from '../reducers/action'
import {RegisterData} from '../reducers/register'
*/
class LoginPage extends Component{
  constructor(props){
    super(props);
    this.state ={
      auth: true,
      message: null,
    }
  }
  /*
  update(){
    console.log(store.getState())
    store.dispatch("SET_DATA");
  }*/
  signIn = (e) => {
    e.preventDefault();
    let uLogin = document.getElementById('login')
    let uPassword = document.getElementById('password')
    if(uLogin.value == 'admin'){
      window.location.href = "/admin"
    } else {
      fetch('/login',{
        method: 'POST',
        headers: {
          'Accept': 'application/json',
          'Content-Type': 'application/json'
        },
        body: JSON.stringify({
          login: uLogin.value,
          password: uPassword.value,
        })
      })
    }
  )
  .then(res => {
    if(res.status == 200){
      res.json().then(data => {
        this.setState({auth: data.auth})
        this.setState({message: data.message})
        console.log("status:", data.code, "\nmessage:", data.message, "\ncode:", res.status)
      })
    }
  })
}
```

```
if(data.code == 'ok'){
```

А қосымшаның жалғасы

```
.clear()
    localStorage.setItem('token',JSON.stringify(data.token))
    window.location.href="/home"
  }
  })
} else {
  console.log("status", "error", "message: ",res.statusText, "; code: ",res.status)
}
})
}
}
handleChange = (e) => {
  e.preventDefault();
  const {id} = e.currentTarget;
  this.props.changeDataAction({[id]:e.currentTarget.value});
}
sendInfo = (e) => {
}
render() {
  return(
    <div className="grid-containerLogin">
      <div className="item item-1L">
        <p className="logo">SSky School</p>
        {!this.state.auth && <span className="wrong">{this.state.message}</span>}
      </div>
      <div className="item item-2L">
        <input type = "text" placeholder = "User" id = "login"
onChange={this.handleChange}/>
      </div>
      <div className="item item-3L">
        <input type = "text" placeholder = "Password" id = "password"
onChange={this.handleChange}/>
      </div>
      <div className="item item-4L">
        <input type = "submit" value = "Вход" className="blueBut" onClick={this.signIn}/>
      </div>
      <div className="item item-5L">
        <a href="/register" onClick={this.sendInfo}>регистрация</a>
      </div>
      <div className="item item-6L">
        <a href="/forgotpass">забыли пароль?</a>
      </div>
    </div>
  )
}
}
const mapStateToProps = store => {
  /*console.log(store)*/
```

```
return{
```

А қосымшаның жалғасы

```
    user: store.user,
  }
}
const mapDispatchToProps = dispatch => ({
  changeDataAction: user => dispatch(changeData(user)),
})
export default connect(mapStateToProps,mapDispatchToProps)(LoginPage);
```

Register.js

```
import React, { Component } from 'react';
import '../styles/Register.css';
import { connect } from 'react-redux';
import { changeData } from '../store/reducers/action';
class Register extends Component{
  /*constructor(props){
    super(props);
  }*/
  endReg = (e) => {
    e.preventDefault();
    let birthDate = document.getElementById('birthDate').value
    let country = document.getElementById('country').value
    let email = document.getElementById('email').value
    let name = document.getElementById('name').value
    let phone = document.getElementById('phone').value
    let firstName = document.getElementById('firstName').value
    let sex = document.getElementById('sex').value
    localStorage.setItem('birthDate',birthDate)
    localStorage.setItem('country',country)
    localStorage.setItem('email',email)
    localStorage.setItem('name',name)
    localStorage.setItem('phone',phone)
    localStorage.setItem('firstName',firstName)
    localStorage.setItem('sex',sex)
    window.location.href="/registerend"
  }
  componentDidMount(){
    document.getElementById('name').value = localStorage.getItem('name')
    document.getElementById('country').value = localStorage.getItem('country')
    document.getElementById('email').value = localStorage.getItem('email')
    document.getElementById('phone').value = localStorage.getItem('phone')
    document.getElementById('firstName').value = localStorage.getItem('firstName')
    document.getElementById('birthDate').value = localStorage.getItem('birthDate')
    document.getElementById('sex').value = localStorage.getItem('sex')
  }
  render(){
    return(
      <div className="grid-containerRegister">
        <div className="item item-1">
```

<p className="title">Регистрация</p>

А қосымшаның жалғасы

</div>

<div className="item item-2">

<div className="redBut">

X

</div>

</div>

<div className="item item-3">

<p className="label">Имя</p>

<input type="text" placeholder="Имя" id="name" onChange={this.handleChange} />

</div>

<div className="item item-4">

<p className="label">Фамилия</p>

<input type="text" placeholder="Фамилия" id="firstName" />

</div>

<div className="item item-5">

<p className="label">Почтовый адрес</p>

<input type="text" placeholder="email@gmail.com" id="email" />

</div>

<div className="item item-6">

<p className="label">Телефон</p>

<input type="text" placeholder="+7 777 456 45 65" id="phone" />

</div>

<div className="item item-7">

<p className="label">Дата рождения</p>

<input type="date" id="birthDate"/>

</div>

<div className="item item-8">

<p className="label">Пол</p>

<select id="sex">

<option value="мужской">мужской</option>

<option value="женский">женский</option>

</select>

</div>

<div className="item item-9">

<p className="label">Страна</p>

<input type="text" placeholder="Казахстан" id="country" />

</div>

<div className="item item-10">

<input type="submit" value="Далее" className="blueBut" onClick={this.endReg}/>

</div>

</div>

)

};

};

const mapStateToProps = store => {

console.log(store)

return{

```
user: store.user, }}
```

А қосымшаның жалғасы

```
const mapDispatchToProps = dispatch => ({
  changeDataAction: user => dispatch(changeData(user)),
})
export default connect(mapStateToProps, mapDispatchToProps)(Register);
```

Users.js

```
import React, { Component } from 'react';

class Users extends Component {
  state = { users: [] }

  /*componentDidMount() {
    fetch('/users')
      .then(res => res.json())
      .then(users => this.setState({ users }));
  }*/

  getSome = () => {
    fetch('/users')
      .then(res => {
        if(res.status === 200){
          res.json().then(users => this.setState({ users }));
        } else {
          console.log("err: ", res.statusText, " code: ", res.status)
        }
      })
  }

  render() {
    return (
      <div>
        <button onClick={this.getSome}>Click</button>
        <h1>Users</h1>
        {this.state.users.map(user =>
          <div key={user.id}>{user.username}</div>
        )}
      </div>
    );
  }
}

export default Users;
```

Chats.js

```
import React, { Component } from 'react';
import { io } from '../socket/socket'
import '../styles/Chats.css';
class Chats extends Component {
  constructor(props){
    super(props)
    this.state = {
```



```
io:io.on('newMes',(data)=>{
```

А қосымшаның жалғасы

```
  this.setState({messages:this.state.messages.concat({author:data.author, message:data.msg, time:
data.time}})})
    }),
    messages:[
    ]
  }
}
sendMes = e =>{
  e.preventDefault();
  let socket = io;
  let author = JSON.parse(localStorage.getItem('token'))
  let text = document.getElementById('msg').value;

  socket.emit('sendMes', {author: author.login,msg: text})
}
render() {
  let author = JSON.parse(localStorage.getItem('token')).login
  return (
    <div className={this.props.className+" chatsWrapper"}>
      <div className="header">
        <button className="clsBtn" onClick={this.props.close}></button>
      </div>
      <div className = "friends">
        <div className = "friend" tabIndex='0'>
          <p>Стив</p>
        </div>
        <div className = "friend" tabIndex='0'>
          <p>Стив</p>
        </div>
        <div className = "friend" tabIndex='0'>
          <p>Стив</p>
        </div>
        <div className = "friend" tabIndex='0'>
          <p>Стив</p>
        </div>
        <div className = "friend" tabIndex='0'>
          <p>Стив</p>
        </div>
        <div className = "friend" tabIndex='0'>
          <p>Стив</p>
        </div>
        <div className = "friend" tabIndex='0'>
          <p>Стив</p>
        </div>
        <div className = "friend" tabIndex='0'>
          <p>Стив</p>
        </div>
        <div className = "friend" tabIndex='0'>
          <p>Стив</p>
        </div>
        <div className = "friend" tabIndex='0'>
          <p>Стив</p>
        </div>
      </div>
    </div>
  )
}
```

<p>Стив</p>

А қосымшаның жалғасы

```
</div>
  <div className = "friend" tabIndex='0'>
    <p>Стив</p>
  </div>
</div>
<div className="chat">
  <div className='messages'>
    {
      this.state.messages.map((item,key)=>(
        <div className={item.author == author?"me":"other"}>
          <p className={item.author ==
author?"mesAuthorMe":"mesAuthor"}>{item.author}</p>
          <div key={key} className={item.author ==
author?"mesTextMe":"mesText"}><p>{item.message}</p><p className={item.author ==
author?"mesTimeMe":"mesTime"}>{item.time}</p></div>
        </div>
      ))
    }
  </div>
  <div className='text'>
    <textarea id="msg" wrap="hard"></textarea>
    <button onClick={this.sendMes}>Отправить</button>
  </div>
</div>
</div>
);
}
}
export default Chats;
```

Game.js

```
import React, { Component } from 'react';
import '../styles/Game.css';
import startLogic from '../components/games/logicGame.js'
```

```
class Game extends Component{
  constructor(props){
    super(props)
    this.state={
      showBut: true,
    }
  }
}
```

```
start= () =>{
  this.setState({showBut:false})
  var canvas = document.getElementById('canvas')
  var ctx = canvas.getContext('2d')
  let canvasWidth = canvas.width
```

```
let canvasHeight = canvas.height
```

А қосымшаның жалғасы

```
let ballsArray = []
let questions = {
  firstLevel :
    [
      {title:'2+5', answer:7},
      {title:'12+5', answer:17},
      {title:'16-9', answer:7},
      {title:'99-33', answer:66},
      {title:'1+1', answer:2},
      {title:'250-190', answer:60},
      {title:'2+8', answer:10},
    ],
  secondLevel:
    [
      {title:'7*5', answer:35},
      {title:'3*3', answer:9},
      {title:'11*12', answer:132},
      {title:'256*14*0', answer:0},
      {title:'4*8', answer:32},
      {title:'7*8', answer:56},
      {title:'9*4', answer:36},
    ],
  thirdLevel:
    [
      {title:'(2+5)*7', answer:49},
      {title:'(17-10)*3', answer:17},
      {title:'9*(9-13)', answer:-36},
      {title:'(256-14)*0', answer:0},
      {title:'7*3-5*2', answer:10},
      {title:'50*50-190', answer:60},
      {title:'2+8*7+2', answer:70},
    ]
}
let question = null
let ballsCount = 4
let butColor = {pause:'#52CEF5',
  restart:'#52CEF5',
  startGame:'#52CEF5',
  mainMenu: '#FF1812',
  firstLevel:'#52CEF5',
  secondLevel:'#52CEF5',
  thirdLevel:'#52CEF5',
  firstLevelText:'#FFFFFF',
  secondLevelText:'#FFFFFF',
  thirdLevelText:'#FFFFFF',
} //#8DE3FC
let levelButBorderColor = {
  first:'#2EA1A8',
```

second:'#2EA1A8',

А қосымшаның жалғасы

third:'#2EA1A8',

```
}
let score = {caughtStyle:0,skippedStyle:0,failedStyle:0}
let cathed = 0
let skipped = 0
let failed = 0
let isWrongStart = false
let isMainMenu = true
let isGame = false
let curQuestion = null
  canvas.addEventListener('mousemove',(e) => {
    let x = e.pageX - canvas.offsetLeft
    let y = e.pageY - canvas.offsetTop
    hoverBut(x,y)
  })
  canvas.addEventListener('mousedown',(e) => {
    let x = e.pageX - canvas.offsetLeft
    let y = e.pageY - canvas.offsetTop
    clickBall(x,y)
    clickBut(x,y)
  })
  function hoverBut(x,y){
    //кнопка паузы в игре
    if(x > 20 && x < 195 && y > 200 && y < 250){
      butColor.pause = '#8DE3FC'
    } else {
      butColor.pause = '#52CEF5'
    }
    //кнопка рестарта в игре
    if(x > 20 && x < 195 && y > 270 && y < 320){
      butColor.restart = '#8DE3FC'
    } else {
      butColor.restart = '#52CEF5'
    }
    //кнопка начала игры в главном меню
    if(x > canvasWidth/2 - 75 && x < canvasWidth/2 - 75 + 175 && y > 25 && y < 75){
      butColor.startGame = '#8DE3FC'
    } else {
      butColor.startGame = '#52CEF5'
    }
    //кнопка первого уровня игры в главном меню
    if(x > canvasWidth/8 && x < canvasWidth/8 + 175 && y > 175 && y < 355){
      butColor.firstLevel = '#8DE3FC'
      butColor.firstLevelText = '#1C1C1C'
    } else {
      butColor.firstLevel = '#52CEF5'
      butColor.firstLevelText = '#FFFFFF'
    }
  }
```

//кнопка второго уровня игры в главном меню

А қосымшаның жалғасы

```
if(x > canvasWidth/2 - 75 && x < canvasWidth/2 - 75 + 175 && y > 175 && y < 355){
    butColor.secondLevel = '#8DE3FC'
    butColor.secondLevelText = '#1C1C1C'
} else {
    butColor.secondLevel = '#52CEF5'
    butColor.secondLevelText = '#FFFFFF'
}

//кнопка третьего уровня игры в главном меню
if(x > canvasWidth/2 + 130 && x < canvasWidth/2 + 130 + 175 && y > 175 && y < 355){
    butColor.thirdLevel = '#8DE3FC'
    butColor.thirdLevelText = '#1C1C1C'
} else {
    butColor.thirdLevel = '#52CEF5'
    butColor.thirdLevelText = '#FFFFFF'
}

//кнопка выхода в меню
if(x > canvasWidth - 190 && x < canvasWidth - 10 && y > 10 && y < 30){
    butColor.mainMenu = '#940804'
} else {
    butColor.mainMenu = '#FF1812'
}
}

function clickBut(x,y){
    if(isMainMenu){
        //условия нажатия на кнопку "начать игру"
        if(question!=null){
            if(x > canvasWidth/2 - 75 && x < canvasWidth/2 - 75 + 175 && y > 25 && y < 75){
                butColor.startGame = '#16ABD9'
                setTimeout(() => {
                    butColor.startGame='#8DE3FC'
                    curQuestion = getQuestion()
                    mainMenu()
                    clearInterval(startMainMenu)
                    createBall()
                    startGame = setInterval(main,1000/60)
                    updateQuestion = setInterval(()=>{
                        curQuestion = getQuestion()
                    },5000)
                    console.log('play')
                    isMainMenu = false
                    isGame = true
                },150)
            } else {
                butColor.startGame = '#52CEF5'
            }
        } else{
            isWrongStart = true
        }
    }
}
```

```
}
```

А қосымшаның жалғасы

```
//кнопка первого уровня игры в главном меню
if(x > canvasWidth/8 && x < canvasWidth/8 + 175 && y > 175 && y < 355){
    levelButBorderColor.first = 'green'
    levelButBorderColor.second = '#2EA1A8'
    levelButBorderColor.third = '#2EA1A8'
    question = questions.firstLevel
    isWrongStart = false
}
//кнопка второго уровня игры в главном меню
if(x > canvasWidth/2 - 75 && x < canvasWidth/2 - 75 + 175 && y > 175 && y < 355){
    levelButBorderColor.first = '#2EA1A8'
    levelButBorderColor.second = 'green'
    levelButBorderColor.third = '#2EA1A8'
    question = questions.secondLevel
    isWrongStart = false
}
//кнопка третьего уровня игры в главном меню
355){
    levelButBorderColor.first = '#2EA1A8'
    levelButBorderColor.second = '#2EA1A8'
    levelButBorderColor.third = 'green'
    question = questions.thirdLevel
    isWrongStart = false
}
}
if(!isMainMenu){
    if(x > 20 && x < 195 && y > 200 && y < 250){
        butColor.pause = '#16ABD9'
        setTimeout(() => {
            butColor.pause='#8DE3FC'
            main()
            if(isGame){
                clearInterval(updateQuestion)
                console.log('pause')
                isGame = false
            } else {
                updateQuestion = setInterval(()=>{
                    curQuestion = getQuestion()
                },5000)
                console.log('play')
                isGame = true
            }
        },150)
    }
    if(x > 20 && x < 195 && y > 270 && y < 320){
        butColor.restart = '#16ABD9'
        setTimeout(()=>{
```



```
butColor.restart='#8DE3FC'
```

А қосымшаның жалғасы

```
cathed = 0
    skipped = 0
    failed = 0
    ctx.clearRect(0,0,canvasWidth,canvasHeight)
    ballsArray = [];
    setTimeout(()=>{createBall()},5000)
},150)
}
//кнопка выхода в меню
if(x > canvasWidth - 190 && x < canvasWidth - 10 && y > 10 && y < 30){
    question = null
    skipped = 0
    cathed = 0
    failed = 0
    ballsArray = []
    levelButBorderColor.first = '#2EA1A8'
    levelButBorderColor.second = '#2EA1A8'
    levelButBorderColor.third = '#2EA1A8'
    isWrongStart = false
    isGame = false
    isMainMenu = true
    clearInterval(updateQuestion)
    clearInterval(startGame)
    startMainMenu = setInterval(mainMenu,1000/60)
}
}
}

function getQuestion(){
let i = getRandom(question.length,0)
return ({question: question[i].title,answer: question[i].answer})
}

function clickBall(x, y){
for(let i = 0; i < ballsArray.length ;i++){
    if( x >= ballsArray[i].x - 30 && x <= ballsArray[i].x + 30 && y > ballsArray[i].y - 30
&& y < ballsArray[i].y + 30){
        if(curQuestion.answer == ballsArray[i].num){
            cathed += 1
            score.catchedStyle = 0.9
        } else {
            failed += 1
            score.failedStyle = 0.9
        }
    }
    ballsArray[i].y = 0
    ballsArray[i].x = getRandom(canvasWidth - 30,canvasWidth/3 + 30)
    ballsArray[i].speed = getRandom(4,2)
    ballsArray[i].num = question[getRandom(question.length,0)].answer
}
}
```

}