

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

Коммерциялық емес акционерлік қоғамы
АЛМАТЫ ЭНЕРГЕТИКА ЖӘНЕ БАЙЛАНЫС УНИВЕРСИТЕТИ

Компьютерлік технологиялар
кафедрасы

«Қорғауға жіберілді»
Кафедра меңгерушісі

(аты-жөні, ғылыми дәрежесі, атағы)

« » 20 ж.

(қолы)

ДИПЛОМДЫҚ ЖОБА

Тақырыбы: Қаладағы мейрамханаларға онлайн-брондау қызметін енгізу

Есептеу техникасы және бағдарламалық қамсыздандыру мамандығы бойынша
Орындаған Балтабек Асхат БТК-12-1

(аты - жөні)

(тобы)

Жетекші Қортымбаев Т.Т. К.ғ.ғ.-м.ғ.н. доцент

(аты-жөні, ғылыми дәрежесі, атағы)

Кеңесшілер :

Экономикалық бөлім бойынша :

Ғ.Ғ.И. доцент Турсалбаев Б.С.

(ғылыми дәрежесі, атағы, аты-жөні)

« 26 » 05 20 16 ж.

(қолы)

Өмір тіршілігі қауіпсіздігі бойынша:

Проф. Т.Е.Ғ. Хакимжанов Т.Е.

(ғылыми дәрежесі, атағы, аты-жөні)

« 25 » мамыр 20 16 ж.

(қолы)

Есептеу техникасын қолдану бойынша :

К.ғ.ғ.-м.ғ.н. доцент Қортымбаев Т.Т.

(ғылыми дәрежесі, атағы, аты-жөні)

« 2 » маусым 20 16 ж.

(қолы)

Мөлшер бақылаушы: К.ғ.ғ.-м.ғ.н. доцент Қортымбаев Т.Т.

(ғылыми дәрежесі, атағы, аты-жөні)

« 2 » маусым 20 16 ж.

(қолы)

Пікір жазушы :

Техника ғылымдарының кандидаты, доцент, Сванбаев Е.А.

(ғылыми дәрежесі, атағы, аты-жөні)

« 31 » мамыр 20 16 ж.

(қолы)

Алматы 2016

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

Коммерциялық емес акционерлік қоғамы
АЛМАТЫ ЭНЕРГЕТИКА ЖӘНЕ БАЙЛАНЫС УНИВЕРСИТЕТІ

Ақпараттық және ақпараттық технологиялар факультеті
Есептеу техникасы және бағдарламалық қамтамасыз мамандығы
компьютерлік технологиялар кафедрасы

жобаны орындауға берілген

ТАПСЫРМА

Студент Балтабек Ахат Балтабекұлы
(аты - жөні)

Жоба тақырыбы "Қала мейрамханаларына онлайн-брондау қызметін енгізу"
ректордың «10» наурыз 2016 ж. №21 бұйрығы бойынша бекітілген.

Аяқталған жұмысты тапсыру мерзімі: «06» маусым 2016 ж.

Жобаға бастапқы деректер (талап етілетін жоба нәтижелерінің параметрлері және нысанның бастапқы деректері)

Web-технологияларға негізделген
web-қосымша жасау. Бұл қызмет мейрам-
хана тұтынушыларына онлайн-брондауға,
талаптар арқылы мейрамханаларға мәтін-
сөйле беруге мүмкіндік береді.

Диплом жобасындағы әзірленуі тиіс сұрақтар тізімі немесе диплом жобасының қысқаша мазмұны:

Мейрамханалардағы ақпараттық жүйе.
Web-қосымша жасауда қажетті құралдар.
Деректер қорының архитектурасы және оны
жобалау.

- техника-экономикалық негізделуі
- жұмыс орнамдағы тиімділік көрсеткіштері

Диплом жобасын дайындау

КЕСТЕСІ

№ р/с	Тарау аттары, әзірленетін сұрақтардың тізімі	Жетекшіге ұсыну мерзімдері	Ескерту
1.	Мейраханна іскері ақпарат жүйесі	30.07.16	
1.1.	Мейраханна кәсібінде ақпараттық жүйесін қолдану	02.02.16	
1.2.	Web-технологиялар арқылы ақпараттарды іске жіюдісі	05.02.16	
2.	Қолданбалы технологиялар	07.02.16	
2.1.	HTML интерактивтік тілі	15.03.16	
2.2.	CSS кәсіпкер кестелер тізімі	17.03.16	
2.3.	JavaScript скрипт жазу тілі	19.03.16	
2.4.	PHP қойылым қолдану тілі	23.03.16	
2.5.	MySQL РҚБЖ	27.03.16	
3.	Бағдарламаны сыпаттау	30.03.16	
3.1.	Бағдарлама шартнамасы	02.04.16	
3.2.	Деректер қорын жобалау	03.04.16	
3.3.	Көрсеткі құру, жана, оңарту	09.04.16	
3.4.	Сұрамастар	15.04.16	
4.	Техника экономикалық негізін	20.04.16	
4.1.	Еңбекке шартнамалар	20.02.16	
4.2.	Еңбекке шартнамалар бойынша құк есептеу	15.03.16	
4.3.	Еңбекке шартнамалар бойынша құк есептеу	17.04.16	
5.	Тиражлік қолдану тілі	16.05.16	
5.1.	Программист әріпке қазіп факторларға таңдау	21.01.16	
5.2.	Жарыстандар уға есептеу	25.02.16	
5.3.	Векторлық және есептеу	27.03.16	
		29.04.16	

Тапсырманың берілген уақыты « 30 » қаңтар 2016 ж.

Кафедра меңгерушісі

(колы)

(аты-жөні, ғылыми дәрежесі, атағы)

Жоба жетекшісі

(колы)

(аты-жөні, ғылыми дәрежесі, атағы)

Орындалатын тапсырманы қабылдаған студент

(колы)

(аты-жөні)

Аннотация

Цель дипломного проекта – представить клиентам ресторана сервис онлайн-бронирование.

Проект написан при использовании возможностей фреймворка Laravel 5.2. Верстка веб страниц была сделана на языке HTML 5. В качестве СУБД был выбран MySQL.

В разделе охрана труда проведен анализ система искусственных и естественных освещений в ресторанах и определены вредные факторы на рабочем месте оператора. А также предусмотрены и сделаны расчеты вентиляционных систем в комнатах ресторана.

В технико-экономическом основание сделаны денежные расчеты эффективности внедрении проекта. И был определен срок окупания стоимости проекта.

Annotation

The purpose of the degree project - to present customers of the restaurant online service for booking.

The project is written using features of the framework Laravel 5.2. Imposition of web pages has been made in the language of HTML 5. As the database was selected MySQL.

In the occupational health and safety analysis system of artificial and natural lighting in restaurants and identified hazards to the operator workplace. Also provided and made calculations of ventilation systems in the rooms of the restaurant.

The technical and economic grounds to make cash payments Effective implementation of the project. And there was a time limit paying off the cost of the project.

Аңдатпа

Менің дипломдық жобамның мақсаты – қаладағы мейрамханалардың тұтынушыларына онлайн-брондау қызметін ұсыну.

Бұл жоба Laravel 5.2 фреймворкінің мүмкіншіліктерін қолдана отырып жасалды. Веб-парақшалары HTML 5 тілінде таңбаланды. Деректер қорын басқару жүйесі ретінде MySQL таңдалынып алынды.

Тіршілік қауіпсіздігі бөлімінде оператордың өміріне зиян келтіретін факторларға, мейрамхана бөлмелерінің жасанды және табиғи жарықтандыру жүйесіне жалпы таңдау жасалды. Мейрамхана ішіндегі ауа алмасу процесі жіті бақыланып, есепке алынды.

Техника-экономикалық негіздеу бөлімінде жобаны енгізудің қаржылық тиімділігі есептелді. Жобаға салынған қаржының өзін-өзі ақтау уақытын анықтадық.

Мазмұны

Кіріспе	7
1 Мейрамхана ісіндегі ақпараттық жүйе	9
1.1 Мейрамхана кәсібінде ақпараттық жүйесін қолдану	9
1.2 Веб-технологиялары негізінде ақпараттандыру ісін ұйымдастыру	10
1.3 Веб-қосымшаны құрастыру тәсілдері.....	13
2 Қолданылған технологиялар.....	15
2.1 HTML таңбалар тілі	15
2.2 Каскадты кестелер стилі CSS	17
2.3 JavaScript – қойылым жасау тілі.....	19
2.4 MySQL – деректер қорын басқару жүйесі.....	21
3 Бағдарламаның сипаттамасы	26
3.1 Бағдарламаның интерфейсі.....	26
3.2 Деректер қорын жобалау.....	36
3.3 Кестелерді құру, өзгерту, жою	40
3.4 Сұраныстар. Көріністер.....	40
4 Техника-экономикалық негіздері	41
4.1 Еңбекақы шығындарын есептеу.....	41
4.2 Әлеуметтік алым бойынша шығындарды есептеу	45
4.3 Амортизациялық алымдар шығынын есептеу.....	46
4.4 Электр энергия шығының есептеу.....	47
4.5 Пайдалану шығындары	49
4.6 Барлық шығындар тармақтары бойынша құн есептеу	49
4.7 Ителлектуалды еңбек құны.....	50
5 Тіршілік қауіпсіздігі.....	51
5.1 Программисттің жұмысына әсер ететін негізгі қауіпті әрі зиянды факторларды талдау	51
5.2 Дербес компьютердің шу шығару деңгейін есептеу.....	52
5.3 Жарықтандыруды бағалау.....	53
5.4 Жасанды жарықтандыруды есептеу	54
5.4.1. Жарықтандыруды коэффициентті қолдану әдісімен есептеу	54
5.4.2. Жарықтандыруды нүктелік әдіс бойынша есептеу.....	57
5.5 Серверлік бөлменің вентиляция жүйесін есептеу	59
5.5.1. Вентиляцияны есептеу	59
Қорытынды	62
Әдебиеттер тізімі.....	63
Қосымша А	65

Кіріспе

Қазіргі заманда мейрамхана ісін басқаруда қолданбалар өте көп, дегенімен, қолданбалардың көбеюі олардың сапасына кері әсерін тигізері анық. Осы мәселеге қатысты автоматтандырылған және жұмыс процесін жылдамдатуға бағытталған өз бағдарламамды жасау туралы шешім қабылдадым.

Интернет желісінің жедел дамуына байланысты мейрамханалар өз ісін осы бағытта да басқаруды жөн санайды. Қолданушыға көп мүмкіндіктер жасау әр мейрамхананың басты мақсаты болып табылады. Кез-келген жаңа кәсіпорынның алдыңғы қатардан көрінуіне себепкер тек көп инвестиция ғана емес, жаңа автоматтандырылған технологияларды қолдану да үлкен себепкер.

Мейрамхана ісін басқару жүйесі бұл – автоматтандырылған ақпараттық көп функцияналдық жүйе. Автоматтандырудың негізгі мақсаты – жұмыс істеу үрдісін жеделдету, кейбір ережелерге қайшы келетін істерді болдырмау. Автоматтандыру жүйесі осы екі қасиетті тиімді әрі сапалы түрде қолдануға мүмкіндік береді.

Бұл қызметтің негізгі басты артықшылығын айта кетсек:

– онлайн-брондау ыңғайлылығы, қаланың кез-келген жерінен үселге тапсырыс беруге болады;

– маркетинг ісіне үлкен, алдыңғы қатарлы, жаңа үрдістер енгізеді;

– көп уақыт алатын күнделікті жұмыстарды біршама азайтады.

Осындай автоматтандырылған жүйе арқылы кәсіпорын басқармасы көптеген бәсекеге қауқарлы мүмкіндіктерге ие болады.

Бұл қызметтің деректер базасы қолданушыға мынадай ақпараттар бере алады:

– мейрамхана асханасы туралы жалпы мәлімет;

– жұмыс істеу уақыты;

– адам басына орташа есеп шығыны;

– мейрамхана мекен-жайы немесе хабарласу телефондары

– және өте маңызды емес көптеген ақпараттар.

Тамақтандыру кәсіпорындарын жіктеу үшін бірқатар өлшемдерді пайдалану қажет. Аталмыш өлшемдердің ішінде ең маңыздыларына мыналар жатады:

– сауда-өндірістік қызметтің сипаты;

– орналасқан жері (барлық қызмет түрі үшін маңызды фактор);

– қызмет көрсетілетін клиенттердің құрамы (сегменті);

– өнімнің түрлері (маманданымы);

– үй-жайдың сыйымдылығы;

– қызмет ету нысаны;

– қызмет көрсетілетін уақыт;

– қызмет көрсету деңгейі (сапасы).

Сонымен бірге ұлттық менталитетпен қалыптасқан қызмет көрсету мәдениетінің де сол немесе бөтен елде тамақтандыру кәсіпорындарын ұйымдастыру нысандарына әсер ететінін ескеру қажет.

1. Мейрамхана ісіндегі ақпараттық жүйе

1.1 Мейрамхана кәсібінде ақпараттық жүйесін қолдану

Қазіргі кезде мейрамхана ісі елімізде қарқынды түрде дамып келеді. Табысты, әрі тиімді кәсіп түрі болып табылады. Алдығы кеңестік дәуірдегі қоғамдық тамақтандыру орталықтары деңгейінен бірнеше есе өскен. Халықаралық есептеуде қонақ үй және мейрамхана ісі барынша дамыған салалардың алдыңғы қатарында тұр.

Мейрамхана бұл – қонақтарға тамақтар, түрлі сусындар, сонымен қатар, фирмалық және күрделі әзірленген кондитерлік тағамдардың кең түрлерін ұсынатын тамақтану орны. Мейрамханалардағы қызмет жасаудың жоғары сатылы, тәжірибелі аспазшыларымен, даяшыларымен қамтамасыз етіледі және демалыс-көңіл көтеру шарасын ұйымдастырады.

Қазіргі кезден біраз уақыт бұрын компьютер арқылы ақпарат таратуды тек тәжірибелі қолданушылар ғана жасай алған болса, бүгінде ғаламтор желісін пайдалану компьютер танымалдылығы секілді, дәл сондай танымал бола бастады. Осы кезде компьютердің коммуникацияларын қолдану үшін аса білім талап етілмейді. Кез-келген қатардағы қолданушыға түсінікті, әрі еш қиындықсыз интернет желісін пайдалануы веб-технологияның жетістігі десе болады. Себебі ғаламтор желісіндегі ақпараттар белгілі бір технология негізінде құрылады. Қарапайым тілмен түсіндірсек, кітапханадағы кітаптар жік-жігімен қалай реттеліп, қол жеткізгенге оңай орналастырылса, ғаламторте де дәл солай реттеліп, веб порталдар, веб сайттар, веб парақшалар кейпінде келеді.

Веб-сайт немесе сайт (ағыл. website, web – өрмек және site – орын) – бұл HTTP/HTTPS протоколдарының арқасына жүзеге асырылатын және тек бір ғана адреске бағдарланған бір немесе бірнеше веб-беттердің жиынтығы. Сайт беттері ортақ, басты адреспен, сонымен қоса, негізгі атаумен, логикалық құрылымымен, автормен және бет келбетімен қосылған. Жалпы қол жетімді сайттар жиыны дүниежүзілік торды құрайды.

Веб-сайт – бұл жарнама орны. Онда мекеме жайлы барлық қажетті ақпаратты (мекеме жұмысын, атқарылған қызметтерін, байланыс телефондарын, реквизиттерді және т.б.) жайғастыруға болады.

Сайт барлық уақытта жұмыс жасайды және ақпаратты қолданушыға қажетті уақытта жеткізуге мүмкіндік жасайды.

Сайттың адресін бүкіл жарнамалық материалда (визиткалар немесе билбордтарда, әр түрлі анықтамалық кітаптарда және т.б.) жайғастыруға болады. Әрдайым түзетулер енгізу, жаңалықтарды жаңалау, хабарламаларды орнату тез және жеңіл орындалады. Сонымен қоса, өте аз уақыт ішінде аз күш және қаражат жұмсалады.

Сайтқа кірген адам тұтынушы болып саналады. Қолданушы санына қарай сайт мәртебесі артады. Веб-сайт арқылы потенциалды пайдаланушы шеңбері кеңейеді. Пайдаланушылар іздеу серверлері арқылы сайт адресін таба

алады, бұл арқылы мекеме туралы ақпараттармен танысып, хабарласып, нағыз тұтынушыға айналады.

Сайттағы пайдаланушылар ғаламтор арқылы қажетті деректерді көрсетіп, сұраныс жасап және тапсырыс беруі мүмкін. Сондай қызмет жасау арқылы пайдаланушылар сайттан қажет деректерін тез тауып ала алады.

Веб-сайт – қолданушылармен қарым-қатынас жасау үшін керек. Сайт көмегімен пайдаланушыларға жаңалықтар мен түрлі ақпараттарды және т.б. жеткізуге болады. Сонымен қоса, сайтта «Сұрақ-жауап» бөлімін жасауға болады, онда қолданушылар өз сұрақтарын, пікірлерін жазады және сол сұрақтарға толық, жоғары дәрежеде жауап жазуға болады.

Мінекей, жоғарыда айтылған жағдайларды ескерсек, веб-технологиясының біздің күнделікті өмірімізде алатын орыны ерекше. Қандай салада да болмасын халықтың жылдам және ыңғайлы жұмыс істеуіне мүмкіндік береді. Инновация кезеңіне сай осы салада жаңа қарқынмен дамуда. Енді осы технологияның өзге де жақтарынан көрінуін тосудамыз.

1.2 Веб-технологиялары арқылы ақпараттандыру ісін ұйымдастыру

Ақпарат бұл – кейбір байланыс формаларының белгіленуі немесе материалдық сфераның белгілі бір заңдылықтарына сүйенетін және оның адам санасындағы көрінісі болатын объектілердің, үдерістердің, құбылыстардың тәуелділіктері.

Ақпарат көзінің тәуелдігіне, өзімен жұмыс жасау тәсілдеріне қатысты белгілі бір қасиеттерімен сипатталады. Осылардың маңызды түрлері төменде көрсетілген:

– ақпарат қоршаған орта туралы өзі келгенге дейін болмаған жаңа мәліметтер береді;

– ақпарат таратушыда өзінің белгілері мен сигнал түрінде келетіндігіне қарамастан ақпарат материалды болмайды;

– таңбалар мен сигналдарды тек оны танып және қабылдай алатын қабылдаушыға ғана ақпарат жеткізеді;

– ақпарат тасушыдан бөлінбейді, сонымен қатар белгілі бір тасушыға байланысты емес және белгілі бір тілге қатысты емес;

– дискреттік ақпарат – бұл хабарлама түрінде берілген нақты мәліметтерден құралады;

– осымен қоса, ақпарат үздіксіз – ол бірігіп және қадамдап дамиды.

Ақпараттың сапасы көрінетін негізгі қасиеттер төменде көрсетілгендей:

– репрезентативтігі – таңдау жасау дұрыстығы және көрсетілетін үрдіске ақпараттың пара-пар құрылуы;

– мазмұндылығы – хабарламада семантикалық хабарлама саны өңделетін ақпараттар көлемі қатынасына тең келетін ақпараттардың семантикалық мөлшері;

– жеткілікті болуы – шешім қабылдау үшін хабарлаушыға көрсеткіштің жиынтық нығыздығы;

– қол жетімділігі – ақпаратты жеткізу формасы ыңғайлығы;

– өзектілігі – ақпараттардың сол мезетте қолданудағы құндылығының деңгейі;

– құндылығы – нақты бір мәселені шешудегі маңыздығы;

– түсінікті болуы – қолданушылардың білім деңгейіне ақпарат мазмұнының сай келуі;

– ықшам болуы – хабарланатын мәлімдемелердің жеткізілуінің сығылу деңгейі.

«Технология» сөзі өнер, шеберлік, істей алу дегенді білдіретін гректің «techne» сөзінен шыққан, гректің «logos» сөзі – түсінік, оқу дегенді білдіреді.

Технология – ол ғылыми және инженерлік білімдердің қосындысы. Бұл өнім немесе қызмет көрсетуді іске асыратын материалдық, техникалық, энергетикалық, өндірістің еңбек факторларының жиынтығы.

Ақпараттық технология – бұл ақпаратты сақтаумен, өңдеумен айналысатын адамдардың еңбегін тиімді ұйымдастыру тәсілдерін үйрететін, өзара байла нысқан ғылыми, технологиялық, инженерлік дисциплиналар кешені.

Негізінен, барлық ақпараттық технологиялар қажет ақпараттарды шартты етілген сапасы мен белгіленген тасымалдаушы арқылы жеткізуге бағытталған.

Ақпараттық технологиялар басты үш компоненттерден құралады:

– техникалық құралдардың кешені – есептеуіш, телекоммуникациялық және техникалық ұйымдастыру;

– программада құралдар жүйесі – барлық (жүйелік) және функционалды (қолданбалы) программаның жабдықтамасы;

– ұйымдастырушы – әдістемелі қамтамасыздандыру жүйесі.

Жүйе бір мақсатқа жетуге бағдарланған бүтіндей, өзара байланысқан элементтер жиыны болып табылады. Қоғамның дамуының қазіргі деңгейі жалпы ақпараттану үрдісімен көрнектеледі, бұл дегеніміз барлық ғылымдар мен техника салаларын компьютерлендіру және ақпараттық технологиялардың енуімен сипатталады. Осыған байланысты негізгі мағынасы болып, күнделікті қызметте ақпаратты өңдейтін құралдар мен әдістердің қолда нуы болады. Жаңа ақпараттық технологияның әсер етуімен өндірісте және басқару процестерінде үлкен өзгерістер болып отырады. World Wide Web – әлемдік компьютерлік торап құрамында алуан түрлі ақпараттары бар миллиондаған сайттардан тұрады. Адамдар Internet технологияларын қолдану арқылы сол ақпараттарға қол жеткізеді. WWW кеңістігінде пайдалану үшін арнайы программалар – веб техноглогиясы қолданылады. Web браузердегі бүкіл ақпарат WWW негізгі байт элементтері болып табылатын Web парақтар түрінде беріледі.

Веб-парақтар мультимедия технологиясын қолдағандықтан, өзіне түрлі ақпараттарды біріктіре алады: текст, графика, дыбыс, анимация, видео. Веб-парақтардың қаншалықты сапалы және әдемі жасалуы көп жағдайда торапқа байланысты.

Қолданушылар тартымды және графикамен, анимация шегімен безендірілген, сонымен қоса, дұрыс, жылдам жүктелетін веб-парақтарды қарайды.

WWW технологиялардың жоғарғы мүмкіндіктері WWW серверлерінің өсу нәтижелеріне және интернет желісінің өсуі себебінен пайда болады. WWW сервер – бар ақпаратты сақтайтын, қолданушыға гипермәтінді құжатқа қол жеткізілетін, жергілікті немесе ішкі корпоративті торап. Қолданушылар WWW-серверлерімен қарым-қатынас жасауы үшін арнайы программалық жабдықтамалар – браузер (англ. browser) немесе көру бағдарламасы деп айтатын жабдық қолданады.

WWW-серверлерінің жұмыс жасау схемасы төмендегідей:

Тораптарды қолданушылар браузер деп аталатын бағдарламалық жабдықтамалар пакеттерін жеткізеді. Мұның қатарына мынадай функциялар қосылады. Сервермен қатынас орнатады, одан кейін:

- керек құжаттарды алады;
- алынған құжаттарды көрсетеді;
- қолданушының әрекетіне жауап беру, жаңа құжатқа ие болу.

Қолданушылардың ықпал етуімен не болмаса браузерді жіберген соң, WWW серверлерімен байланыс жасайды және бұған берілген құжаттарды алуға сұраныстар жібереді.

WWW-серверлері сұраған құжаттарды іздейді және браузерге нәтижелерді қайтадан қайтарады.

Ал браузер құжаттарды қабылдап, қолданушыларға ұсынады және оның әрекетін тосады.

Сайт құрылысы, сол сияқты гипермәтіндік құрылымындағы оқу-әдістемелік материалдарды ұсыну мәліметтерді гипермәтінді белгілеу тілі HTML, CSS, JavaScript және басқа да веб бағдарламалау тілінің варианттарының көмегімен жүзеге асырылады.

HTML мәтіні сан түрлі арнайы конвертер және редакторлармен жасалатындығынан, HTML тілін меңгерместен вебте жұмыс істеуге болады. Жалпы, тікелей HTML-де кодтау оңай. Ол тіпті, HTML-редакторлар немесе конвертерлерді үйренуден қиын емес, себебі бұл жерде өз мүмкіндігі шектелген, сан алуан платформаларда жұмыс жасайтын сапасыз HTML кодтарын жасайды және қателіктер жіберіледі.

HTML тілі бірнеше вариантта қолданылады және дамып отырады, бірақ оның құрылымы кейінде пайдаланылуы мүмкін. HTML үйреніп және оны терең түсіне отырып, оны мүмкіндігінше ұлғайта отырып, Веб көптеген браузерімен қарауға болатын құжат жасай аласыз. Бұл басқа әдістерді пайдалану мүмкіндігін шығармайды, мысалы, Netscape Navigator, Internet Explorer немесе кейбір басқа да бағдарламалар ұсынатын ұлғайтылған мүмкіндіктер әдісі. HTML-де жұмыс істеу – бұл нақты қажет болған кезде ұлғайтуды пайдалана отырып, стандартталған тілде құжат жасау ерекшелігін меңгеру тәсілі.

HTML World Wide Web Consortium-мен бекітілген болатын. Бұл бірнеше кең таралған браузерлерде қолданылады және Вебке қатысы бар барлық бағдарламалық қамтамасыз етудің негізі болуы мүмкін.

HTML шегінде негізінен, студенттің игеруіне қолайлы түрдегі ақпарат беруді жүзеге асырады. JAVA-Script пайдалану түрлендіруге және ақпарат берілуін «жандандыруға», сол сияқты интерактивтің кейбір элементтерін енгізуге мүмкіндік береді.

Ғаламтордың көптеген бөтен технологияларының ішінен пайдаланушыдан жылдамдықты арналардың болуын талап етпейтіндерді пайдалану мүмкін. Мұндай жағдайда оқу сайты немесе гипермәтіндік оқулық әзірлеу кезінде клиент жақта жүктелетін файлдардың өлшемі бойынша оны ыңғайлыландыруға бөлінеді. Осындай тәсіл тәжірибе жүзінде Flash, VRML, мультимедиа және кейбір бөтен технологияларды пайдалануға жол бермейді.

Тіпті қарапайым графикалық суреттер мұқият өңдеуді талап етеді. Осындай шектеулер жүйелік емес болжамдармен салыстырғанда, оқу сайтының немесе гипермәтіндік оқулықтың сапасын едәуір төмендетеді. Мәселен, гипермәтіндік оқулықтың әдістемелік құрылысы, кәдімгі баспа оқулығынан ерекшеленбейді. Ақпарат тарауларға, параграфтарға және т.б. бөлініп, қажетті сурет және сызбамен өрнектеледі. Тарауларды қолдану тәртібі тақырыпты оқу логикасымен анықталады. Гипермәтіндік оқулықты немесе оқу сайтының өзін бөлімге және бөлімшелерге бөлу қолданудағы жылдам навигация мен қолдану икемдігін қамтамасыз етеді.

Оқу-әдістемелік материалдарды гипермәтінді ұсынудың электрондық оқулықтар істеген сәтте ескеру қажет болатын, бірнеше ерекшеліктері болады.

Ережелерді сақтау табыстарға қажет мүмкіндіктер мен ережелердің шартты жиынтығын жасауға мүмкіндік береді:

- 12000Б/сек қосу жылдамдығында бір минуттан аспайтын жүктеменің уақыты, фрагментке берілген оқулық сайтының жалпы көлемін бөлу;
- навигациялық элементтерді пайдалану;
- күрделі көркемделген суреттерден бас тарту;
- басып шығаруға арналған құрал варианттарының бар болуын қамтамасыз ету.

Айтқандай келтірілген ережелер оның толымды болуына талаптар қоймайды, бірақ олар кем дегенде барынша ыңғайлы гипермәтіндік оқулықтардың дайын болуын қамтамасыз етеді. Бірақ, мейрамхана сайты әзірлеуге бұл ережелер жеткіліксіз. Мейрамхана сайты дайындау үшін веб-мастерингтің дәстүрлі әдістері тиімді.

1.3 Веб-қосымшаны құрастыру тәсілдері

Бүгінгі таңда портал құрудың әдістері өте көп. Порталдың түрі мен мақсаттарына, бағдарламалаушының біліктілігіне және тапсырыс берушілердің қаржылық мүмкіндігіне байланысты өзара ерекшеліктері болады. Дегенімен, әдістерді алуан түрлі қасиеттері бойынша топсатыруға болады:

- қолданылатын бағдарламалық қосымша бойынша;
- сервердің құрылымына байланысты;
- технологиялық мүмкіндіктеріне сүйене отырып;
- маркетингтік сипаттамалары арқылы.

Келешекте порталдың иелері үшін іске асыру мерзімі мен бағдарламаның құны өте маңызды. Сол себептен бір-біріне ұқсамайтын бірнеше тәсілдер болғанымен, бағдарлама құны мен оны құруға кеткен уақыт мөлшері ең маңызды факторлар деп қарайтын болсақ, технологиялық іске асыру бойынша негізгі үш әдіс қолданылады.

1) Басынан бастап жасау (нөлден құру)

Бұл әдіс ең ұзақ әрі ең қиын болып есептеледі. Бұл әдісті жүзеге асыру үшін веб бағдарламалау тілдерінің (PHP, HTML, CSS), сонымен қатар деректер қорымен (MySQL) жұмыс істеу біліктілігін шарт етеді. Сонымен қоса Dreamweaver, FrontPage, WebSiteX5 сияқты бағдарламаларды пайдаланып порталды құруға болады.

Бұл әдісте бүкіл жұмысты бағдарламалаушылардың өздері атқарады. Алайда нәтиже кейде өте қарапайым бір түрде көрініс беруі мүмкін. Сонымен бірге бұл әдісті іске асыру үшін өте көп уақыт пен күш жұмсалады.

2) Конструкторларды қолдану

Білікті веб-бағдарламалаушының жоғары тәжірибесін қажет етпегендігінен, ең қарапайым тәсіл болып саналады. Дегенімен жоғары сапалы нәтижеге үміттенбеуге де болады. Шаблон мен макет негізінде жасалған көп конструкторлар бар. Ең көп тарағандары «wix.com», «setup.ru», «umi.ru», «ucoz.ru» және т.б.

Басында конструкторларды қолданып, уақыт өте бағдарламалаушы оны түсінуге кеткен уақыты үшін қынжылуы мүмкін. Өйткені, бұл тәсілдің мынадай үлкен кемшіліктері бар:

- конструкторлардың мүмкіншіліктерінің шектеулілігі, тек өзінде қарастырылған функцияларды жасай алуы;
- арнайы кодтау фрагменттерінің болуы немесе бұл секілді ресурстың танымалдылығын арттыру қиындығы.

3) Құрылымды басқару жүйелерін қолдану.

Ең тез және кеңінен қолданылатын әдіс. Бүгінде WordPress, Joomla, Django, 15 Drupal және т.б. жүйелер сайттар мен блоктарға кеңінен қолданылады. Ал форумдар үшін JAVAbb, ExBB, PunBB, SMF жүйелері.

Басқару жүйелерін (CMS) қолдану шаблондар мен құрылым сұлбаларын пайдалануға мүмкіндік береді, және бағдарламалаушы жұмысын жылдамдатып, жеңілдетеді.

2.Қолданылған технологиялар

Бұл жобаны іске асыру барысында келесідей технологиялар қолданылды: HTML таңбалау тілі, PHP скрипт жасау тілі, CSS каскадты кестелер стилі, MySQL деректер қоры, JavaScript қойылым жасау тілі, Sublime Text 3 программалау редакторы және Xampp виртуалды хостинг.

2.1 HTML таңбалар тілі

HTML – бүкіләлемдік тордағы құжаттарды орнастырудың стандартты тілі. Көп веб-беттерде HTML тілінде орнастырудың сипаттамасы болады. HTML тілі браузерлармен интерпретицияланады және пайдаланушыға лайықты құжат ретінде көрсетіледі.

Бүкіләлемдік торда HTML-беттер, әдетте, браузер арқылы желіден HTTP немесе HTTPS хаттамасымен қарапайым мәтін түрінде немесе сығылу арқылы. Бастапқыда HTML тілі құжаттарды қосу (шығару) құралдарына байлаусыз құрау мен өңдеу құрылғысы ретінде ойластырылған және жасалған. Идеал түрде HTML белгісімен мәтін стилистикалық және құрылымдық түзетулерсіз әртүрлі техникалық қамтамасыздандырулармен (қазіргі компьютердердің түсті экраны, организер монохромды экраны, өлшемі шектелеген ұялы телефон экраны немесе мәтінді дыбыстық қосатын құрылғы және бағдарлама) құрылғыда қосылуы қажет. Алайда HTML қазіргі кезде қолдануы бастапқы мақсатынан өзгеше. Мысалы, TABLE тегі құжатта кесте жасау үшін қолданылады, бірақ элементтердің бетте орналасуының безені ретінде жиі қолданылады. Уақыт өте келе HTML тілінің платформаға тәуелсіздігінің бастапқы идеясы қазіргі заман мультимедия және графикалық безен қажетілігіне құрбанға шалынды.

Бұл аббревиатура Hyper Text Markup Language ретінде ашылады және гипермәтінді белгі тілі деген мағына береді. Оны әйгілі WWW бүкіләлемдік тор негізін құраған Тим Бернерс-Ли (сол уақытта бар SQL тілі негізінде), қазіргі уақытқа дейін интернеттің W3c консорциум ішінде (оны тағы W3C (validator) деп атайды) жаңа стандарттарын жасау жұмысына қатысып жүр.

Айта кетсек, HTML тілінің бірінші нұсқасы өткен ғасырдың тоқсаныншы жылдарында пайда болды және ең біріншіден ақпаратты ғылым ортасына беруге бағдарланған, алайда осы кезден бастап бүкіләлемдік тордың әйгілігін бағалай бастаса болады.

Бұған дейін интернет тек бірнеше білетін және қызығатын адамдардың пайдалануында болатын, бірақ HTML және оның жөндем кодын пайдаланушыға түсінікті және ыңғайлы түрге келтіретін (веб бет немесе, жиі айтылатын, веб құжат) бірінші браузерлардың пайда болуынан бастап, бүкіләлемдік тор өзінің салтанатты шеруін бастады. Мүмкін, гипермәтінді белгі тілінің пайда болуын мәтіндіктен графикалық операциялық жүйеге ауысына теңестірсек болады.

Менің ойымша көбіне қызық болатын осы белгі тілінің дамуының шапшыңдығы мен нюанстары. Сонымен, бірінші нұсқаның пайда болғаннан

бірнеше жыл өткеннен кейін Тим Бернерс-Лидің қолға алуымен W3C (World Wide Web Consortium) консорциумы жасалынды, бұл стандарттарды орнықтырушы болуы және нашар салдарға келтіруі мүмкін құрастырушылардың шашырап кету мен бос жүруіне жол бермеуге тиісті болатын. 1994 жылы гипермәтінді белгі екінші нұсқасы жасалынып жатыр, ал 1995 жылы CSS (каскадты кестелер стилі) қолдауымен HTML 3 жұмысы жүріп жатыр. Сол уақытта жаңа браузер мозаика пайда болып бедел жинады, тез уақыт аралығында Netscape Navigator-ға қайта жасалынды. Microsoft Netscape Navigator-ды өзінің Windows интеграциясы үшін сатып алмақ болды, бірақ бұл браузердің құрастырушылары келісім берген жоқ (нәтижесінде Mozilla Firefox пайда болды), нәтижесінде біз қазіргі күнге дейін Microsoft өзі ойлап тапқан - Internet Explorer - Мозаиканың ашық код базасынан жасалынғанын қолданып жүрміз. Бір қызығы, IE (ең белгілі операциялық жүйеде қайта орнатылғандығы үшін) браузерлар рейтингінде Netscape Navigator-ды шығара алды, алайда орнына жаңа ойыншыларды тапты (тегін браузер Opera, Mozilla, бүгінгі көшбасшы Google Chrome және басқалары). Осы уақыт аралығында браузер құрастырушылары жиі "паравоз алдында жүгірді"(валидатор) және өздерінің стандарттарын енгізді, өйткені W3C-дағы HTML форматымен жұмысы айтарлықты баяу жүрді.

Осындай шаруына көріп консорциум бір 1997 жылда үлкен секіріс жасады – белгі тілі екі өзгеріске ұшырады, 3.2 версиясынан 4.0 версияна дейін, содан кейін (1999 жылы) және қазіргі күнге дейін қолданатын версияға дейін - HTML 4.01. Осыдан кейін, сенесіз бе, стандарт шамамен он екі жыл ауысқан жоқ (барлығына бәрі ұнады, браузер құраушыларына да, W3C валидатор командасына да).

Қазір консорциум браузер құраушылар конгломератының бастауымен форматпен жұмыс жасау активті түрде жүріп жатыр, алайда оның пайда болуын жақын болшақта күту керек емес шығар. Дегенменде кейбір жаңадан енген HTML 5 форматы кейбір браузерлермен қолданып жатыр. Шамамен дәл осылай кейін де жалғаса береді - жаңа формат бөлік-бөлік болып ене бастайды, бірақ оның толық барлық браузерлермен қолдауы көп уақытты алады.

Сонымен, HTML көмегімен біз веб беттер (құжаттар) құрамыз. Көптеген бірдей домен аттарға тиесілі web беттер сайт деп аталады. Сайт құрамына кіретін веб беттер физикалық түрде бір жерде болуы және кез-келген пайдаланушыға 24 сағатта ашылуы керек екені түсінікті. Осы мақсаттарға веб желілер арналған, әдетте арнайы құралдарылған, мекемеде орналасқан және осы мақсатты орындауға келтірілген қарапайым компьютер түрінде. Веб желілер тұрақты және кеңсіздікті интернетке рұқсаты бар, өйткені пайдаланушылар жердің кез- келген нүктесінде және кез-келген аумақта сіздің саттыңызға рұқсат алу үшін.

Сайттарды орналастыруда орын беру қызметі аталмыш хостерларды ұсынады, және де, әрине, осығағ ақша жасайды бірақ та табиғатта тегін хостинг ретінде жағымды ерекше жағдай болады. Сіздің сайтыңызды көру

санына байланысты сізге веб желі толығымен (ерекшеленген желіні көрсету қызметі) және оның кішкене бөлігі (виртуалды хостинг немесе ерекшеленген виртуалды хостинг) керек болуы мүмкін. Сіздің сайтыңыздың кез-келген беті (күжаты) өзінің қайталанбас мекен-жайы немесе, басқа сөзбен айтқанда, URL (Uniform Resource Locator) болады. Пайдаланушы компьютер экранында осы беттің мазмұнын көру үшін оның браузеры желімен http сұрау және http жауап алу арқылы диалог құруы керек.

Осы диалог нәтижесінде браузер күжаттың HTML кодын алады, оны қарастырады, қажет қосымша бетті безендіру элементтерін (суреттер, css файлдар, скриптар) жүктейді және пайдаланушы экранда сайт бетін көреді.

2.2 Каскадты кестелер стилі CSS

HTML негізін білу, сіз үшін кішкене шындықты түсінуге жеткілікті: гипермәтінді белгі тілдің қажетті элементтерін жазу – автор қалағанындай пайдаланушының бетінде барлық ақпараттың орналасуына кепілдеме бермейтін, өте күрделі және шаршататын жұмыс. Елестетіңіз, өзіңіздің өңдеп отырған веб-бетіңіз браузер терезесінің барлық көлеміне орналастыру керек болған мәтіннің он абзацынан және экранның ортасы бойынша туырлау керек болған бес көркемдеуден тұрады. Бұл дегеніміз, HTML-күжаттың кодында `<p align="justify"></p>` тегін он рет және бес рет `` тегінде `align="center"` атрибутын көрсетуіңіз керек болады. Егер сізге, осыдан кейін барлық абзацтағы мәтін түрін өзгерту керек болатын болса, мысалы, түсі мен мөлшерін өзгерту, онда мұндай өзгерістерді он рет енгізуге тура келеді. Нәтижесінде жұмыс көлемі, уақыты ұлғаяды және веб-шебер ретіндегі құны да. Сонымен қатар, күжатқа орнатылып жатқан тег көлемі мен нәтижелік файлдың көлемі ұлғайғандықтан, оның серверден жүктелуі де көбейеді. Яғни, бұл дегеніміз, сіз өз сайтыңыздың пайдаланушыларын жоғалтасыз! HTML-белгілердің мүмкіндіктері шектеулі. Графиканың өлшемін пиксельге дейін нақты көрсетуге болады. HTML-де мәтіндік блоктар өлшемін көрсетуге болмайды: олар үнсіз келісім бойынша шрифтке қатысты өлшемнен шығып, браузермен есептеледі. Беттің авторы пайдаланушының браузер күйге келтірулерін алдын-ала анықтай алмайды, бұл бетте ақпаратты ұсынудың нұсқаларын шектейді. Осы және басқа да мәселелерді шешу үшін W3C консорциумының HTML стандартын өңдеуші-компания бағдарламалаушыларымен жаңа CSS (Cascading Style Sheets – стильдердің каскадты кестелері) стандарты жасалынды. CSS, HTML-ге қарағанда веб-бет алгоритм элементтерін жазуды басқа алгоритмін қолданады. Мәтіндік файлдағы әрбір элементтің қасиетін бір рет көрсету арқылы, кейін сіз осы файлды HTML күжатқа қосып, браузерге веб-беттің әрбір компонентінің мәнін сол жақтан оқыта аласыз. Бұған қоса, сіз осы стильдер жазбасымен жеке файлды түрлі күжаттардың шексіз мөлшеріне қоса аласыз. Бұдан басқа веб-беттерді дайындаудың алдында CSS –тің сөзсіз бір ерекшелігі бар. Өзіңіздің сайтыңыздың немесе бірнеше сайттардың барлық веб-беттердің қандайда бір элементін стилін өзгерту үшін, бір файлдағы кодтің бір қатарын туырлаудың

өзі жеткідікті болады. Бұдан басқа, барлық веб-беттер үшін ішкі файлдан берілген +стильді оған керемет түр беру үшін, кез-келген нақты веб-беттің ішінен анықтауға болады. Сонымен қатар кез-келген бірлік тегтің стилінде өзгертуге болады.

HTML-пен танысқаннан кейін сайт құрастырушалары екі негізгі категорияға бөлінеді. Бір бөлігі HTML көмегімен сайтта барлығын немесе барлығына жақынын жасауға болады, ал басқасы веб-құжаттарды безендіру үшін белгі құралдары жеткіліксіз екенін түсінеді. Шынында сайт жасауды үйренуде HTML тек бірінші фазасы. Келесі қадам стильдерді үйрену немесе CSS (Cascading Style Sheets, каскадты кесте стилі).

Стилмен файл кез-келген веб-документтен сілтеме жасай алғандықтан қайталанатын мәліметтер өлшемінің азаюына алып келеді. Кодттардың бөлінуінің және безенуінің арқасында құжат түрінің басқаруының икемділігі артады және сайт жұмысының жалдамдығы артады.

CSS HTML -мен сәйкестенетін, тек кейбір айырмашылықтары, мысала түсті ажырату әдісі, бар өздігінен жеке тіл ретінде ұсынады.

Бір құжатта бірге қолданыла алатын бірнеше стиль типтері ажыратылады. Бұл браузер стилі, автор стилі және пайдаланушы стилі.

Браузермен веб-беттерге үнсіз келісім бойынша қолданылатын безендіру. Бұл безендіруді "жалаңаш" HTML -де көруге болады: құжатқа ешқандай стиль қолданбаған жағдай. Мысалы, тәгімен өңделетін тақырыбы көп браузерларда сапасы 24 пункт өлшемімен шрифт шығарылады.

Пайдаланушы стилі - бұл браузер баптауынан сайт пайдаланушы қоса алатын стиль. Бұндай стиль үлкен приоритетке ие және құжаттың бастапқы безенін қайта анықтайды.

Стильдер веб-беттерді беттеуде және құжатты, сілтемені, көрініс және басқа элементтерді безендіруде ыңғайлы, епті және әсерлі құрал болып табылады. Стильді қолданудың басты жақсы жақтарына қарамастан бір көргенде білінбейтін CSS -ң барлық артықшылықтарын қарастырамыз.

HTML коды түсті қою, шрифт өлшемі және басқа параметрлер сияқты безендіру элементтерінен бос болу идеясы әлем сиқты ескі. Ең дұрысы веб-бет тек логикалық өңдеу тәгтары болу керек, ал элементтер түрі стиль арқылы тағайындалады. Осылай бөлген жағдайда дизайн жұмысы мен сайт саралауы параллель түрде жүре алады. Әр түрлі құралдардың әт түрлі безені.

Стиль көмегіме әт түрлі шығару құралдары үшін веб-беттің түрін анықтай алады: монитор, принтер, смартфон, планшет және басқалары. Мысалы монитор экранында беттің бір безендіруін шығара алады, ал басу кезінде - басқа. Бұл мүмкіндік тағы әртүрлі құрылғыларда шығарғанда құжаттың кейбір элементтерін жасыра немесе көрсете алады.

HTML-мен салыстырғанда элементі безендіру әдістері кеңейтілген.

HTML қарағанда стильдер веб-құжаттар элементтерін безендіруде көбірек мүмкіндіктері бар. Қарапайым құралдармен элементтің негізгі түсін өзгертуге, жақша қоюға, шрифт қоюға өлшемін, орнын анықтауға және басқаларын орнатуға болады.

Стилдерді жеке файлда сақтауда, ол кэшталады және қайта шақырғанда браузер кэшынан алынады. Кэш есебінен және стиль жеке файлда сақталатындықтан веб-беттің коды азаяды және құжатты жүктеу уақыты азаяды.

Кэш деп пайдаланушының жергілікті компьютеріндегі бірден сайтты шақырғанда браузер файлдарды сақтайтын арнайы орын. Келесі ретте сайтты шақырғанда бұл файлдар желіден жүктелмейді, жергілікті дискіден алынады. Бұндай жол веб-бетті жүктеу жылдамдығын әлдеқайда жоғарылатады.

Сайт - бұл бір-бірімен байланысқан құжаттар ғана емес, сонымен қатар негізгі блоктардың орналасуы мен түрі. Тақырыптардың, негізгі мәтіннің және басқа элементтердің бірдей безендіруін қолдану беттердің үздіксіздігін жасайды және пайдаланушының сайтпен жұмысын және толық қабылдауын жеңілдетеді. Ал құрастырушыларға стилді қолдану дизайнды жобалауды айтарлықтай оңайлатады.

Ортақтандырылған сақтау. Стилдер, әдеттегідей, сайт документтерінің барлығында сілтемелері бар арнайы бір немесе бірнеше файлдарда сақталады. Осының арқасында стиль жөндеуді бір орында орындаған ыңғайлы, сонымен қатар элементтерді безендіру автоматты түрде көрсетілген файлмен байланысқан барлық беттерде өзгереді. Ондаған HTML-файлдарды модификациялағанша стилмен бір файлды түзетіп, керекті құжаттарда безендіру лезде ауысады.

Көрсетілген CSS-файлдардың минималдау әдістерінен басқа gzip утилитасымен файлдарды тура желіде архивтеуге және сығылған түрде браузерге беруге мүмкіндік беретін шешімдер бар. Қазіргі заманғы браузерлар gzip-ті анық түсінеді және оны орнында ашады. Барлық процесс автоматты түрде өтеді және желіден берілетін файлдар өлшемінің айтарлықтай азаюына алып келеді.

Бұл жағдайда сіздің ресурстағы әрбір кодтың басты бөлігінде (HEAD тэгтар арасында) HTML-құжаттар (веб беттер) сыртқы CSS-файлға дейін жолы арнайы тэгте жазылу керек (link) , өйткені кері жағдайда сіздің жобаңыздың келушілері оның сұрқайы түрін көруі мүмкін.

Егер коды бар жол CSS-файлға қосылуы орны болса, онда браузер осы жолды оқығаннан кейін оның жүктеуін бастайды және келушілерге "киінбеген" сайтты көрсетпей стилді безендіруін қосады. Сыртқы стилдерді қосу жолы.

Егер сіз CMS-ті қолдансаңыз HTML-ң басты бөлігі сіздің қолданған күй безенінде JAVA файлдардың біреуінде қалыптасады. Joomla, Wordpress, SMF қозғалтқыштары үшін ашатын және жабатын HEAD тегін қалыптастыратын осы файлдың орнын білсеңіз болады.

2.3 JavaScript – қойылым жасау тілі

Java Script тілінің мақсаты - Веб-бетті активті жұмыс істеуге мүмкіндік береді. Аттарының ұқсастығына қарамастан Java Script тілі Java тіліне ешқандай қатысы жоқ. Java құрастырушылардың өтініміне қажетті

аплиттердің браузерлармен байланысы үшін біраз құралы бар, сол уақытта Java Script дәл осы мақсат үшін жасалынған. Java Script Java-ға қарағанда интерпретация жасай алатын тіл. Java Script тіліндегі бағдарлама бастапқы мәтін түрінде тура HTML- құжатта орналасады және құжатты браузерға жүктеу кезінде автоматты түрде байт-кодқа компиляцияланады. HTML-құжаттарының авторлары үшін Java Script пайдаланушымен әрекет жасайтын HTML-құралдарын жасау жолындағы маңызды алға қадам болып табылады. Осылайша Java Script тілінің басты мақсаты HTML- құжаттардың пайдаланушымен бір-бірімен белсенді әрекетін қамтамасыз ету. Бұл тіл Java, C сияқты үлкен масштабты бағдарламалау тілі болуға ұмтылмайды. Негізінде ол пайдаланушының нақты бір браузермен жұмысын жеңілдететін HTML тілінің кеңейтімі болып табылады. JavaScript-бағдарламалар құжат мазмұнын орындайтыны маңызды факт: оларды активтендіретін құжаттардан тыс болытан Java-апплетке қарағанда ол физикалық түрде HTML -құжаттарының ішінде орналасады.

JavaScript сахнаға 1995 жылдың соңында шықты және қазіргі уақытқа дейін өте әйгілі, көп браузерлармен қолданатын болды. JavaScript тілі Sun Microsystems серіктестігімен Netscape Communication Corporation компаниясымен жасалынды. JavaScript тілінің бастапқы атауы LiveScript болды. Сосын, Java тілі әлемге әйгілі болғанда, LiveScript-ті JavaScript-ке ауыстырды. Мүмкін бұл Netscape компаниясымен коммерциялық қараудан жасалынды: өзінің атауында кілтті сөзі Java болатын барлығы үлкен қызығушылық туғызды.

JavaScript бағдарламалау тілін дамытуда ол кәсіби емес пайдаланушыларға арналған болып болжанған. Қарапайым қолданысты JavaScript көбіне Basic-ке ұқсас, дегенменде оның арқасында квалификациялы бағдарламаушылар жеткілікті байыпты және пайдалы мәселелерді шеше алады. JavaScript автономды бағдарламалар немесе апплет жасауға қолайлы емес, және осыда ол Java, C сияқты басқа тілдерден қатты өзгеше. JavaScript құрылымы HTML құжатының басты мәтініне тура орналасады және браузермен осы құжаттарды жүктеу барысында интерпретацияланады. Сонымен қатар JavaScript көмегімен сіз динамикалық түрде жүктеліп жатқан HTML құжатының мәтінін өңдеп және өзгерте аласыз.

JavaScript тілінің тағы бір ерекшелігі - оның объектілі бағдарланғандығы. JavaScript-тің Java мен C++ қарағанда объектілі-бағдарланғандық мүмкіндіктері кедейлеу болғанымен, олар осы тілде бар. Бағдарламаушыларға көптеген браузерға енгізілген объектілер мен оған жүктелген фреймдар, өлшемдер, басқа құжаттарға сілтемелер сияқты HTML - құжаттар қолжетімді.

Өздік класстарды жасауға рұқсат етілген, сонымен қатар объектілерді топтау және басты классты кеңейтуге арналған мәлімет құрылымына рұқсат етілген. HTML-құжатына енгізілген JavaScript конструкция көмегімен оқиғаларды өңдеуге болады. Бұл оқиғалар пайдаланушының браузер терезесіне жүктелген HTML-құжаттарына әр түрлі операциялар орындау

нәтижесінде пайда болады. Бұл әр түрлі визуалды эффектілерді алуға мүмкіндік береді, мысалыға тышқан сілтемесі тұрғандағы басқару органдарының сыртқы түрінің өзгерісі. Пайдаланушының web желісіне өңдеуге үшін жібермей тұрып форма жолдарына енгізген мәліметтерді алдынала тексеруге болады, сонымен қатар браузер терезесіне HTML-құжатын жүктеуден бұрын іс-әрекет орындаудайды және керекті жұмыстарды орындайды.

Тағы айта кететін жағдай, JavaScript қиын бағдарламалар жасауға арналмаған. Осындай тапсырмаларды басқа бағдарламалау тілдерін қолданып шешкен дұрыс, мысалы Java көмегімен. Алайда, JavaScript көмегімен сіз жеткілікті қиын мінез-құлықты жылдам белсенді беттерді жасай аласыз. Бұл тіл оқуға қиын емес, ал осы тілде жазылған бағдарламаларды ретке келтіру оңайға соғады.

2.4 MySQL – деректер қорын басқару жүйесі

MySQL - бос объектілі-бағдарланған деректер қорын басқару жүйесі (ДҚБЖ). Әр түрлі AIX6 n8hk3 BSD-жүйелер, HP-UX6, IRIX, Linux, Mac OS, X6 Solaris/OpenSolaris6 Tru6 QNXUnix-тектес платформалар күйінде жұмыс істеуге негізделген және SQL:2011 стандартының мүмкіндіктерін қолдайды.

Жоғары жүктейтін сайт жасай отырып, көптеген тауарларды қолдануға болады.

Веб-сервер, операциялық жүйе, веб-сервер бағдарламалық қамсыздандыруы, деректер базасын басқару жүйесі және бағдарламалу тілі немесе сценарий құру үшін аппараттық қамтамасыздандыруды таңдау қажет. Осы компоненттердің таңдалуына тәуелді. Мысалы барлық операциялық жүйелер кез-келген жабдықта жұмыс істей алмайды, барлық сценарий жасау тілдері барлық деректер базасына қосуды жүзеге асыра алмайды. Осы дипломдық жұмыста веб-сервердің аппараттық қамсыздандыруы, операциялық жүйелеріне және бағдарламалы қамтамасыздандыруына аса назар аударылмайды. Бұл бізге қажет емес. JAVA-ның ең жақсы қасиеттерінің бірі ол Microsoft Windows рұқсаты бар, UNIX-ң көптеген версияларына және толық функционалдық веб-серверлардың кез-келгеніде орындалады. MySQL жүйесі дәл осындай икемділікке ие.

Бұны көрсету үшін жоғары жүктемелі сайттың дамуы мен тестілеуі Apache Веб-сервері арқылы Microsoft Windows операциялық жүйесінің басқаруымен жергілікті серверде жүргізілді, және Қазақстандық серверларда Debian GNU/Linux операциялық жүйе басқаруымен және Nginx веб-сервисімен PS хостинг компаниясында жасалынды.

Мастер-база мәліметтерді немесе мәлімет сызбаларын өзгертетін әрбір сұранысты арнайы файлға - аталмыш binary-log жазып отырады. Бірінші бинарлы лог белгілі бір өлшемге жеткенде жазба келесі файлға өтеді. Бинарлы логтарға арнайы индекс бар, сондай-ақ оларды басқару үшін командала тізбегі бар(мысалы ескі бинарлогтарды өшіру үшін.)

Слейв желі арқылы осы файлдарды "шикі" күйде алады(мастердің іске асуын оңайлатады) және оны өзінің мәліметтеріне қолданады. Слейв бинарлы логтар оқыған орнын есте сақтайды, сол себепті қайта қосқанда мастер керекті орыннан логты жіберуді сұрайды. Репликацияның екі режимі бар - statement және row. Бірінші режимде мастер бинарлы логқа мәліметтерді өзгерту үшін шығыс сұраныстарды жазады. Слейвта барлық сұраныстар мастерде орындалған сияқты орындала береді.

ROW режимында 5.1 версиялы MySQL рұқсат берілген бинарлы логқа сұраныстар жазылмайды, ал осы сұраныстармен өзгертіліп қойған мәліметтер (DDL)мәліметтер сызбасын өзгертетін сұраныстардың жазылуы бірдей.

ROW -режиміндегі оқиға былайша:

– Бір жол деректер мәліметтер - командалар үшін INSERT DELETE, сәйкесінше енгізілген жол жазылады, өшірілген жол;

– Екі жол - BEFORE ЖӘНЕ AFTER - UPDATE ҮШІН.

Мәліметтерді үлкен өзгерістер кезінде осындай сұраныстың өзін жазғанға қарағанда бинарлы логтар әлдеқайда көп болады, бірақ бұл жеке жағдай. Өзінің демонда ROW-репликациясын қолдану бізге сұранысты жасауды білу қажет еместігімен ыңғайлы және біз тез керегімізді аламыз - өзгертілген жолдар.

Айта кетсек, өзіміздің ең-маңызды-мастер-серверде ROW-репликациясын қосу міндетті емес. Мастер-сервер әдеттегі STATEMENT репликация (резерв, бэкаптар) арқылы бірнеше слейвті қамтамасыз ете алады, осы слейвтердің бөлігін ROW-лог жаза алады, сондай-ақ демондар мәліметтер сұрай алады. Деректер базасы тіркелген барлық пайдаланушылар туралы барлық ақпаратты, жасалған галерия категориялары, қосымша модуль, жасалған тақырыптар мен хаттар, пайдаланушы топтар, пайдаланушы қосымшалар, қосымша файлдар, компоненттар, виджеттар, конфигурацифлык жөндеулер, суреттерге рұқсат алу, сұраныстарды сақтау керек.

Әкімшілік бөлігіне кіру керек: жаңа пайдаланушылар туралы ақпарат, сайт жөндеу бөлігі, әкімшілік форум мүмкіндігі, форум арқылы статистика шығару мүмкіндігі, пайдаланушылардың есептік жазбаларын басқару және басқа мүмкіндіктер. Мәліметтерді жобалау процесін шартты түрде екі сатыға бөлуге болады: логикалық моделдеу және физикалық жобалау. Біріншінің нәтижесі аталмыш мәліметтердің логикалық(немесе концептуалды) моделі, әдетте "зат- байланыс" диаграммасымен беріледі немесе осындай диаграммалрды шығару үшін стандартты нотация біреуі ER (entity-Relationship) диаграммасымен беріледі. Екінші саты нәтижесі дайын деректер базасы немесе оның жасалуы үшін DDL- скрипт болып табылады. Мәліметтердің реляционды моделінде "біреуі-бәріне" деген байланысы қолдайтын болғандықтан, онда рұқсат берілген категориялар көмегімен реляционды модельге лайықты емес арнайы көптеген байланысты көрсететін механизм қажет. Бұл "біреуі-бәріне" қарым-қатынасты байлындыратын арнайы қосымша енгізулер арқылы жасалынады, бұл қарым- қатынас атрибуты қарым-қатынас байланыстыратын бастапқы кілттер болады.

Сонымен қатар жаңа қарым-қатынастың әрбір атрибут сыртқы кілт болып табылады (foreign key), ал бірге олар жаңа затты байланыстыратын бастапқы кілтті құрайды (primary key). Қарым-қатынасты өзгерту ассоциативті кестелер арқылы жасалынады. Енді ассоциативті кестелерді қосқанда барлық заттарды және олардың қарым-қатынасын SQL тіліндегі барлық кестелердің нақты сипаттамасын келтірейік. Деректер базасының маңызды бөлігінің бірі рұқсат құқығының шығарылуы, өйткені оған тіркелмеген рұқсат алудан қорғаныс үшін керек және рұқсат алудан қорғаныс болуы керек. Сәтсіздіктен қорғаныс үшін резервті көшіру стратегиясы жасалынады. Әрбір пайдаланушының тіркелмеген кіргеннен қорғау үшін пайдаланушыға оның рұқсат алу құқығына алынады. Әрбір деректер базасында объектіге рұқсат алуын өзіндік ролі бар. Көп администраторлар бұл құқықтармен жүргісі келмейді сол себепті бәрін дерлік шешетін үндемей public орнатады.

Біздің сайттың ДҚ-да үш негізгі рол бар:

– Администратор. Жөндеу құқығы бар, жаңалаулар, өшірулер, жаңа материалдар жасау және сайтта пайдаланушылар. Тура супер пайдаланушы болады.

– Түзетулер. Сайт материалдарын жасау және өндеу құқығы бар.

– Тіркелген пайдаланушылар. Жабық беттерді көру құқығы бар пайдаланушылар. Сайттың әкімшілік бөлігіне рұқсат жоқ.

Резервті көшіру (back copy) - мәліметтердің қайта жазу үшін бастапқы немесе бұзылу жағдайында жаңа орында тасушыларда (қатқыл диск, дискета) көшірмесін жасау процесі. Резервті көшіру жұмыс көшірмесі жоғалып немесе бұзылғанда жылдам және арзан ақпаратты қалпына келтіру керекті (құжаттар, бағдарлама, жөндеулер және тағы басқалары).

Веб-сайт керекті функционалдығын қамсыздандыратын оригиналды графикалық дизайнға ие болу керек. Веб-сайт үшін фирмалық стиль мен визуалды идентификациясы ретінде уникалды элементтер жиыны жасалынды. Белгі, логотип, түсті шешімі, интерфейс элементтері үшін шартты белгілер. Дизайн келесі негізгі шарттарды қанағаттыру керек:

– графикалық элементтері веб-беттерде спецификасына қарай жасалуы керек;

– негізгі ақпаратқа рұқсат максималды тез болу керек және пайдаланушы назары маңызды бағдарлық элементтерде аударылуы керек;

– негізгі ақпарат, мәзір және басқа бағдарлық элементтер ашық браузер терезесінде горизонталды бұраусыз рұқсат алынуы керек;

– басты бетте іздеу блогында негізгі ақпараттар "толығырақ" деген сілметелер қойған жөн. Бұл пайдаланушыны керек тарауға өткізуге көмектеседі.

– тарауларда бағдар интуитті-түсінікті болу керек;

– тақырып пен безендіруге назар аудару керек.

Веб-сайт дизайнын құрастырғанда келесі принциптарды қадағалау керек:

- Дизайнмен жұмысқа жүйелі түрде жол беру, алдыңғы қатарлы тенденцияларды ортақтандыру;
- Келушілерге және ақпаратқа назарлану (user centered design);
- Ресурстағы бағдардың жеңіл болуы (easy navigation);
- Веб-сайтты пайдаланушылардың оңай түсінуі;
- Беттердің минималды "салмағы" (килобайт);
- Элементтер қайталануын жүйенің ішінде болмауын қадағалау (modularity);
- Сапалы және толық графика;
- Басты және басқа беттерді қолдануда көріністер ақпаратқа сәйкес және өлшемі аз болуы керек. Стилдің ортақтылығы болуы керек;
- Көрініс элементтері тіпті жай түстер мен басты түстер мәтінде объектілерді оқуға кедергі болмау керек;
- Көрініс объектілері негізгі ақпаратты бірінші терезе шекарасынан шығарып тастамау керек.
- Төменде біздің порталдың деректер қорының кестелерінің тізімі көрсетілген.

```
Schema::create('restaurants', function (Blueprint $table) {
    $table->increments('id');
    $table->string('title');
    $table->text('description');
    $table->string('seat');
    $table->string('avg_price');
    $table->string('kitchen');
    $table->string('allowed_self');
    $table->boolean('parking');
    $table->string('features');
    $table->string('extra');
    $table->string('site');
    $table->timestamps();
});
```

```
Schema::create('restaurant_places', function (Blueprint $table) {
    $table->increments('id');
    $table->integer('restaurant_id');
    $table->string('coordinates');
    $table->string('address');
    $table->string('phones');
    $table->string('working_hours');
    $table->timestamps();
});
```

```
Schema::create('password_resets', function (Blueprint $table) {
```



```
$table->string('email')->index();  
$table->string('token')->index();  
$table->timestamp('created_at');  
});
```

```
Schema::create('users', function (Blueprint $table) {  
    $table->increments('id');  
    $table->string('name');  
    $table->string('email')->unique();  
    $table->string('password');  
    $table->rememberToken();  
    $table->timestamps();  
});
```

3. Бағдарламаның сипаттамасы

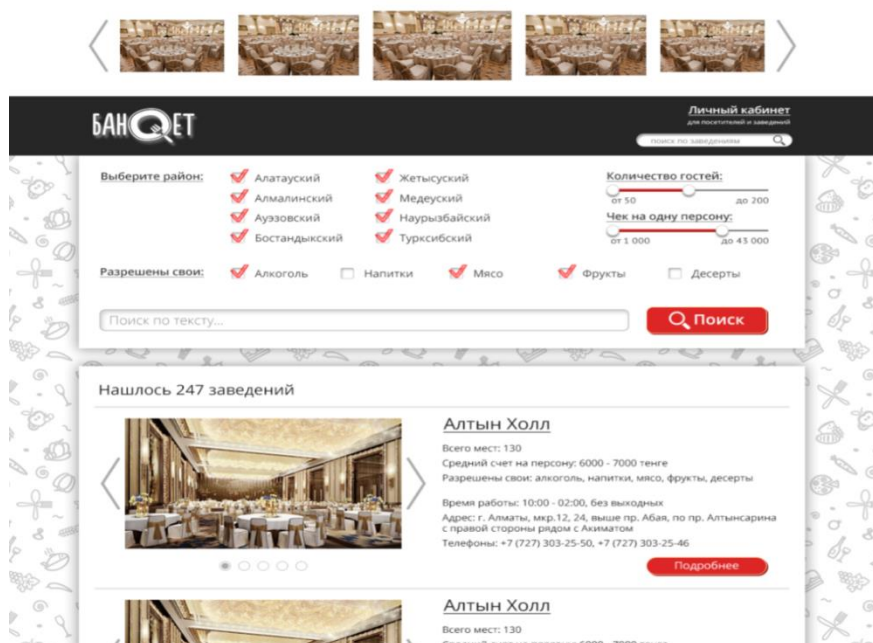
Менің дипломдық жобамда қала мейрамханаларының тұтынушыларына онлайн-брондау қызметі жасалады. Бұл жасалған өнім қаладағы біздің қызметке тіркелген барлық мейрамхана туралы ақпараттар береді және ғаламтор желісі арқылы үстелден орын алу қызметін атқарады. Біздің қызметті қолданған адамдар үшін көптеген тиімді әрі қолайлы мүмкіндіктер жасалған.

3.1. Бағдарлама интерфейсі

“Banqet.kz” сайтына кірген кезде, сіз өзіңізге жақын орналасқан мейрамханалармен қоса адам басына шаққандағы орташа шығын көлемін, телефон номерін, мекен-жайы туралы ақпаратты көре аласыз.



3.1-сурет. Веб- қолданбаның бастапқы беті(1)



3.2-сурет. Веб-қолданбаның бастапқы беті(2)

Сайт негізгі 4 құрылымдық бөліктен тұрады:

- Сайттың басы(HEADER);
- Іздеу формасы(SEARCH);
- Контент(CONTENT);
- Сайттың төлесі(FOOTER).

Негізгі бетті шығаруға арналған контроллердің коды

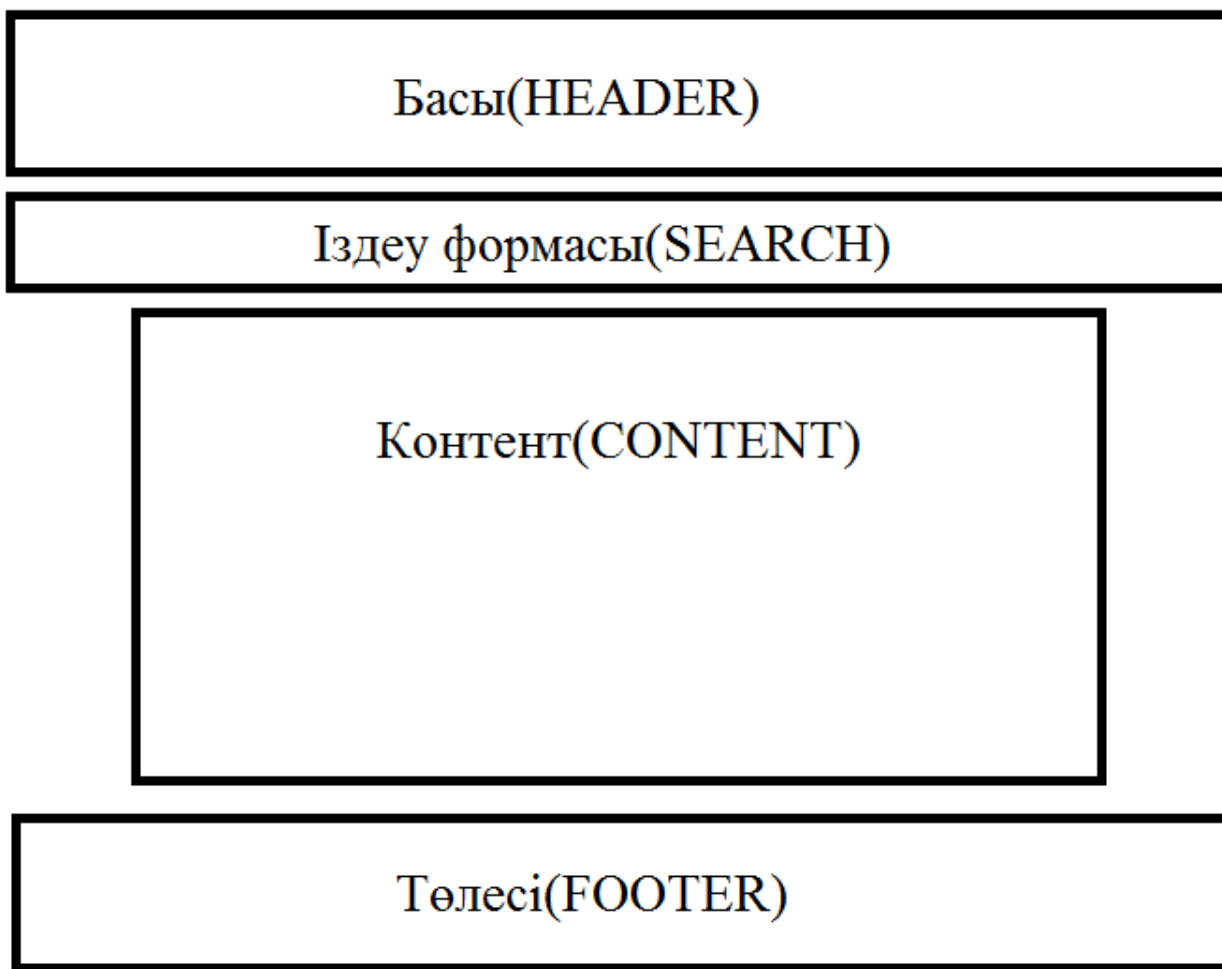
```
<?php
namespace App\Http\Controllers;
use App\Http\Requests;
use Illuminate\Http\Request;

class HomeController extends Controller
{
    public function __construct()
    {
        $this->middleware('auth');
    }
    public function index()
    {
        return view('home');
    }
}
```

Жеке парақшаға кіру коды

```
<?php
namespace App;
use Illuminate\Foundation\Auth\User as Authenticatable;

class User extends Authenticatable
{
    protected $fillable = [
        'name', 'email', 'password',
    ];
    protected $hidden = [
        'password', 'remember_token',
    ];
}
```



3.3-сурет. Сайттың негізгі құрылымы

Бұл веб-қосымшаның басы «Жеке парақшаға» кіру немесе шығу батырмасынан, логотиптен және қысқаша іздеу формасынан тұрады.



3.4-сурет. Сайттың басы(HEADER)

3.4-суреттегі іздеу формасы тек мейрамхана аты бойынша ғана іздеуге арналған.

Выберите район:

- Алатауский
- Жетысуский
- Алмалинский
- Медеуский
- Ауэзовский
- Наурызбайский
- Бостандыкский
- Турксибский

Количество гостей:

от 50 до 200

Чек на одну персону:

от 1 000 до 43 000

Разрешены свои:

- Алкоголь
- Напитки
- Мясо
- Фрукты
- Десерты

Поиск по тексту... 🔍 Поиск


3.5-сурет. Іздеу формасы(SEARCH)

Іздеу формасы мейрамхана орналасқан район бойынша, қонақтарды қабылдайтын орын бойынша, бір адамға шаққанда орташа чек шығыны бойынша, тыстан алып келетуге рұқсат етілген тағамдар бойынша және жалпы мейрамханада кездесетін сөздік қор бойынша мүмкіндік береді.

Сайттың контентінен біз мейрамхана туралы мынадай ақпараттар ала аламыз:

- мейрамхана аты;
- қызмет көрсете алатын адамдардың жалпы саны;
- адам басына шаққандағы орташа есеп;
- тұтынушы өзімен алып келуіне болатын азық түрлерін;
- жұмыс жасау уақыты;
- мейрамхана адресі;
- телефон номері.

Нашлось 247 заведений



⏪ ⏩


● ○ ○ ○ ○

Алтын Холл

Всего мест: 130
Средний счет на персону: 6000 - 7000 тенге
Разрешены свои: алкоголь, напитки, мясо, фрукты, десерты

Время работы: 10:00 - 02:00, без выходных
Адрес: г. Алматы, мкр.12, 24, выше пр. Абая, по пр. Алтынсарина с правой стороны рядом с Акиматом
Телефоны: +7 (727) 303-25-50, +7 (727) 303-25-46

Подробнее



⏪ ⏩

● ○ ○ ○ ○

Алтын Холл

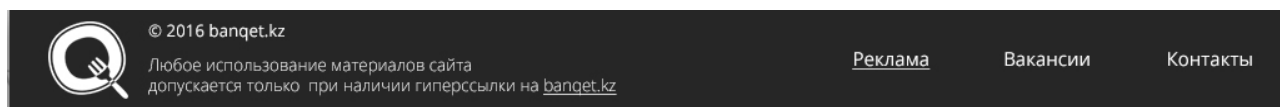
Всего мест: 130
Средний счет на персону: 6000 - 7000 тенге
Разрешены свои: алкоголь, напитки, мясо, фрукты, десерты

Время работы: 10:00 - 02:00, без выходных
Адрес: г. Алматы, мкр.12, 24, выше пр. Абая, по пр. Алтынсарина с правой стороны рядом с Акиматом
Телефоны: +7 (727) 303-25-50, +7 (727) 303-25-46

Подробнее

3.6-сурет. Контент(CONTENT)

Сайттың төлесінде кишкентай логотип және авторлық құқық туралы ақпар алуға болады, сонымен қатар ізбен хабарласу контактісіне, жұмыс жайлы хабарламаға, жарнама беруге арналған сілтемелер менюі орналасқан.

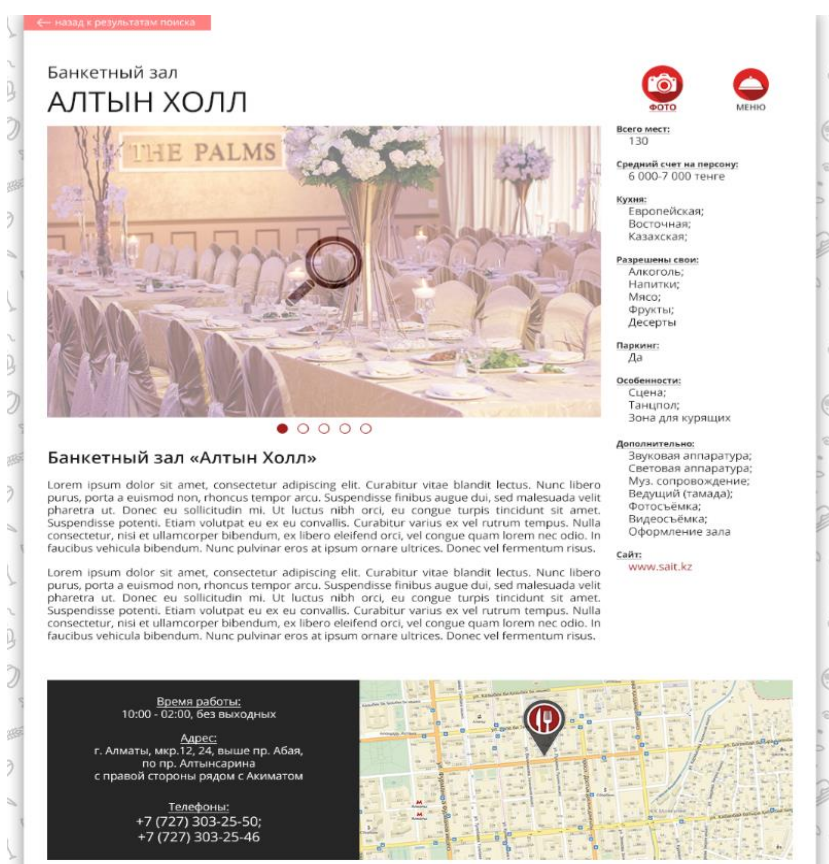


3.7-сурет. Сайттың төлесі(Footer)

Сайттың контентіндегі мейрамханалар тізімінен біреуін таңдап ішкі парақшаға өтсек, сол мейрамхана туралы толығырақ ақпараттар алуға болады. Олар мынадай:

- мейрамхана мәзірі;
- асханасының ассортименті;
- паркинг;
- мейрамхана артқшылықтары;
- қосымша ұсынылатын қызметтер;
- мейрамхананың жеке сайтына сілтеме;
- мейрамхана туралы кішкене мәтін;
- карта

және т.б.



3.8-сурет. Мейрамхана туралы толық ақпарат парақшасы

Бұл парақшаны ДҚ-нан шығару коды:

```
class PersonalController extends Controller
{
    public function index()
    {
        $restaurants = Restaurant::with('places')->all();
        return view('pages.personal', ['restaurants' => $restaurants]);
    }
}
```

```
Route::get('/personal', 'PersonalController@index');
```

Мейрамхананың өзге де суреттерін көру үшін 3.8-суреттегі фото сілтемесіне жүгінеміз.



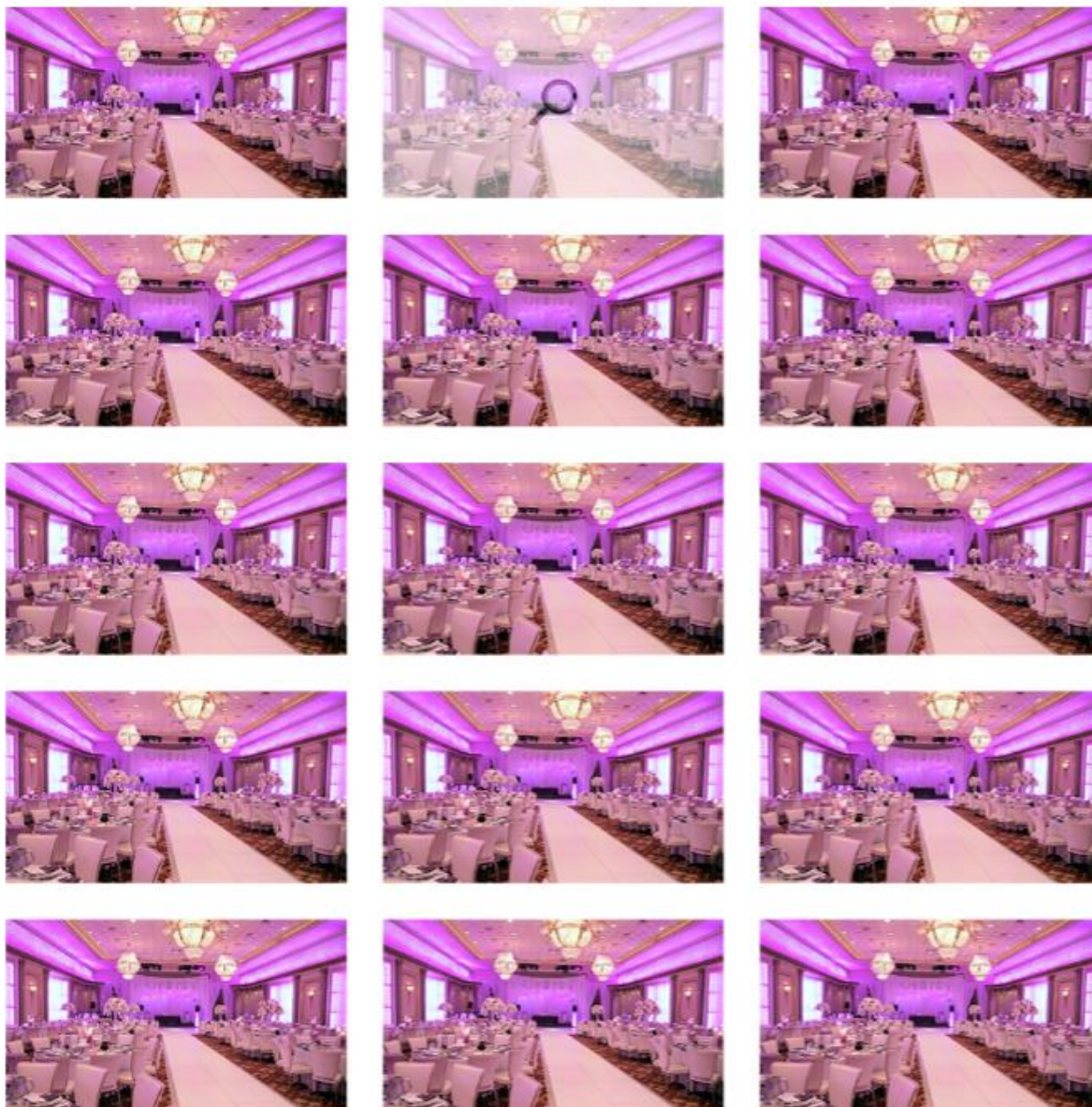
1 2 3 4 5 6 7 8 9 Следующая

Добавить заведение

3.9-сурет. Мейрамхана альбомдарының парақшасы

Альбом ішіндегі қосымша суреттер келесідей көрініс табады.

Банкетный зал «АЛТЫН ХОЛЛ» > Альбомы > Интерьер

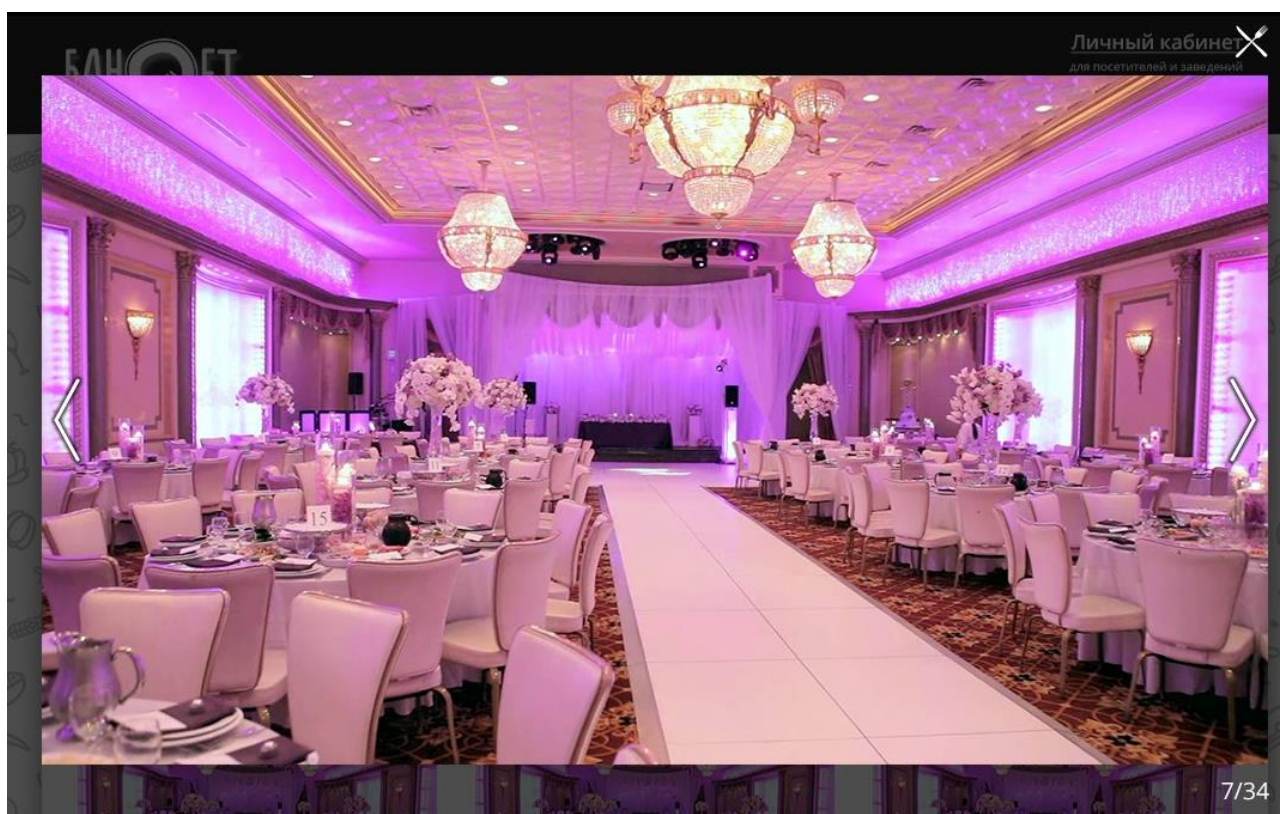


Предыдущая [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) Следующая

Добавить заведение

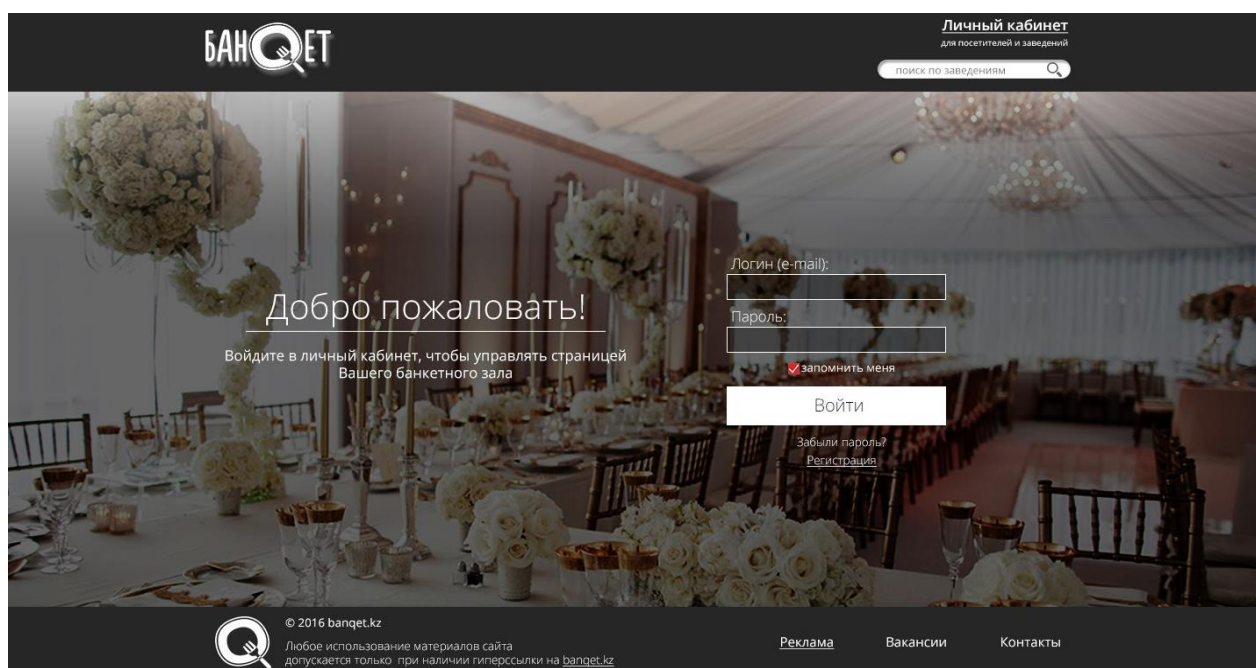
3.10-сурет. Альбом ішіндегі жеке суреттер

3.10-суретте мейрамхананың барлық суреттерімен танысамыз. Және ол суреттерді үлкейтіп көруге болады.



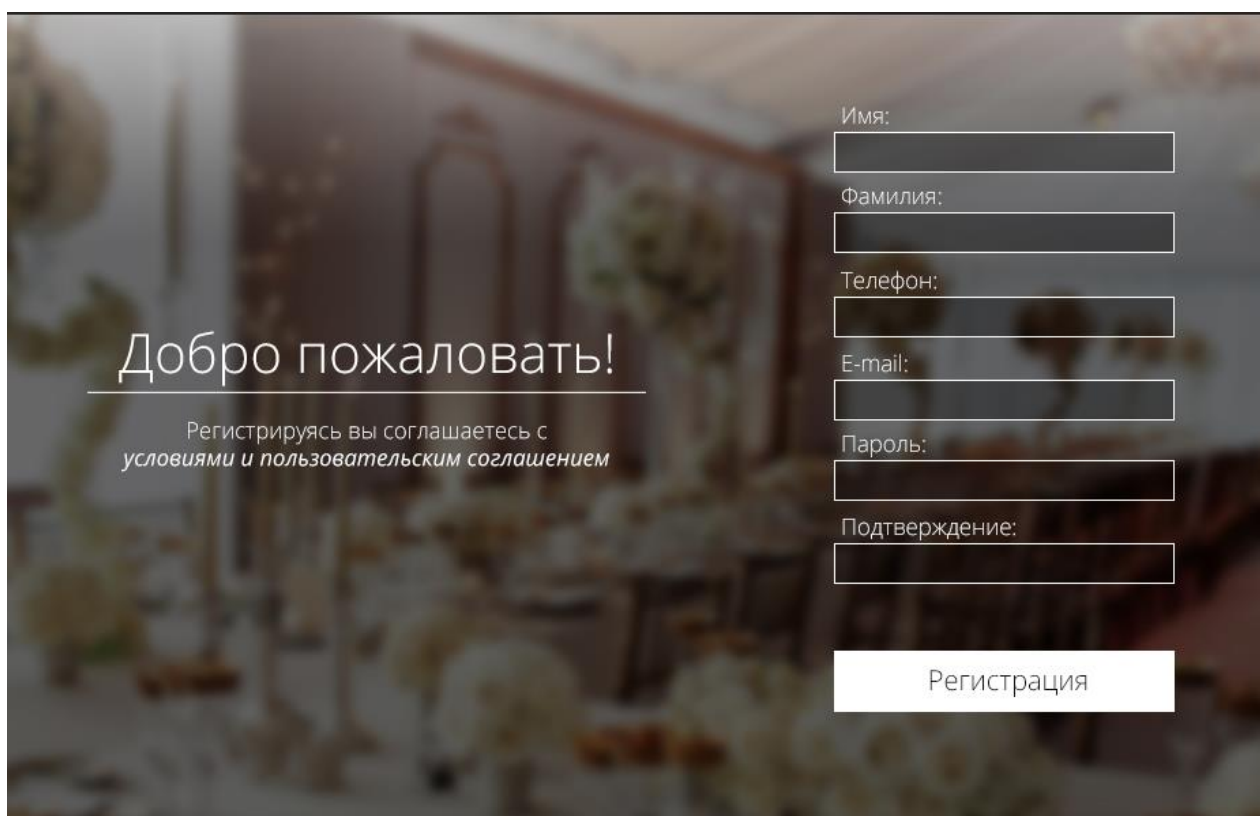
3.11-сурет. Суреттің үлкейген сәттегі көрінісі

Тіркелген қолданушыларға өзінің басқару бетіне кіруі үшін почта адресін және құпия сөзін тереді.



3.12-сурет. Жеке парақшаға кіру беті

Ал тіркелу парақшасында сіз өзіңіздің атыңызды, тегіңізді, телефон номерін, почтаның адресі мен құпия сөзді енгізу керексіз.



Добро пожаловать!

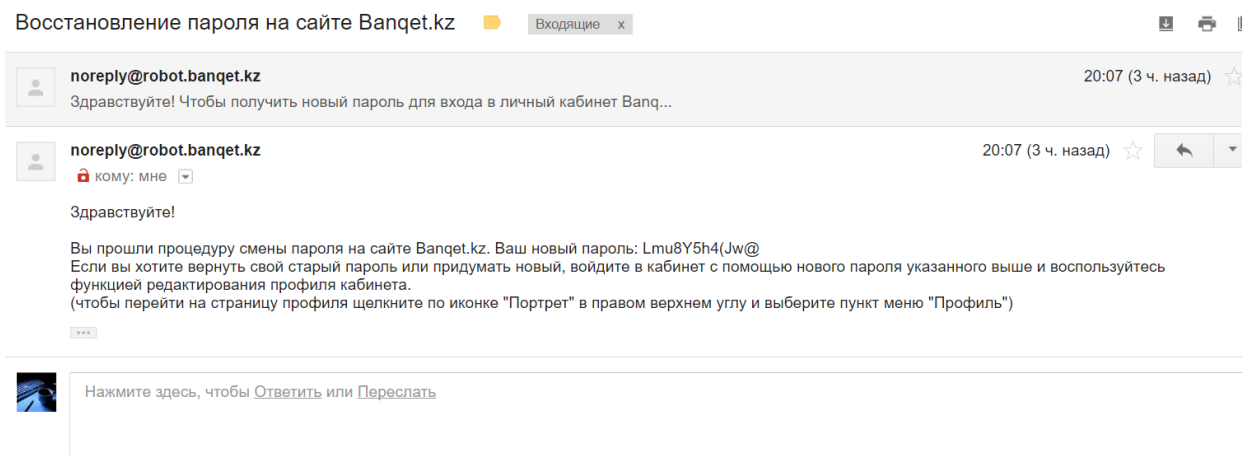
Регистрируясь вы соглашаетесь с условиями и пользовательским соглашением

Имя:
Фамилия:
Телефон:
E-mail:
Пароль:
Подтверждение:

Регистрация

3.13-сурет. Тіркелу парақшасы

Тіркелген қолданушылар өз құпия сөздерін ұмытып қалған жағдайда оны қайта қалпына келтіру мүмкіндігі бар. Ол үшін сіз өзіңіздің почтаңызды терсеңіз жеткілікті. Сонан соң, сіздің почтаңызға жаңа құпия сөзді енгізу парақшасына сілтеме хат келеді. Сол арқылы сіз өз құпия сөзіңізді өзгерте аласыз.



3.14-сурет. Почтаға келген құпия сөзді қайта қалпына келтіру хаты

Біздің қызметке қосылғыңыз келсе, өз мейрамханаңызды тіркеуіңіз керек. Ол үшін сіз мынадай анкетаны толтыруыңыз қажет:

- мейрамхана аты;
- қалаңызды(қазіргі кезде тек Алматы қаласы);
- мейрамхана адресін;
- бір немесе бірнеше телефон номерін;
- компания логотипін.

Ескерте кететін сәт: сізге 30 күнге тегін біздің сайтта тұра аласыз. Содан кейінгі қызметтерге қаржы төлеуіңізге тура келеді.

Добавление нового заведения

Название:

Город: **Алматы**

Адрес:

Телефоны:

+7

+7

+7

[добавить еще](#)

Логотип:



Добавить банкетный зал
30 дней бесплатно

3.15-сурет. Жана мейрамхана қосу парақшасы

Және осы беттен сіз жиі қойылатын көптеген сұрақтар мен оларға жауаптар оқи аласыз. Ол келесідей көрініс табады.

Часто задаваемые вопросы:

<p>ЧТО ПРОИЗОЙДЁТ, КОГДА ЭТОТ СРОК ИСТЕЧЁТ?</p> <p>Вам будет предложено оплатить дальнейшее размещение, согласно выбранному пакету. Мы пришлём вам письмо-напоминание ближе к окончанию этого срока.</p>	<p>КАК ВЕРНУТЬСЯ НА ПЛАТНЫЙ ПАКЕТ?</p> <p>Здесь же, в вашем личном кабинете, выберите один из пакетов — стандартный или премиум — и внесите оплату со своего счёта, пополнив его любым удобным способом, например, банковской карточкой.</p>
<p>А ЕСЛИ Я НЕ ОПЛАЧУ?</p> <p>Страница вашего заведения автоматически перейдёт на бесплатный тарифный план, и на ней перестанут отображаться фотографии в альбомах, новости и меню (при этом они сохранятся в вашем личном кабинете на случай, если вы решите вернуться к платному размещению). А ещё при бесплатном размещении приоритет показа заведения среди других самый низкий — в конце списка.</p>	<p>МОЖНО ЛИ МЕНЯТЬ КОЛИЧЕСТВО РАЗДЕЛОВ ДЛЯ МОЕГО ЗАВЕДЕНИЯ И РАЗМЕР АБОНПЛАТЫ?</p> <p>Конечно! Вы всегда сможете сменить один платный пакет на другой, отключить больше ненужные вам разделы или добавить новые. Каждое такое изменение будет приводить к перерасчёту суммы абонентской платы.</p>

3.16-сурет. Жиі қойылатын сұрақтар

3.2. Деректер қорын жобалау

“Banqet.kz” сайтының деректер қоры MySQL деректер қорын басқару жүйесінде құрылған.

Сайттың деректер қорының тізімі келесідегідей:

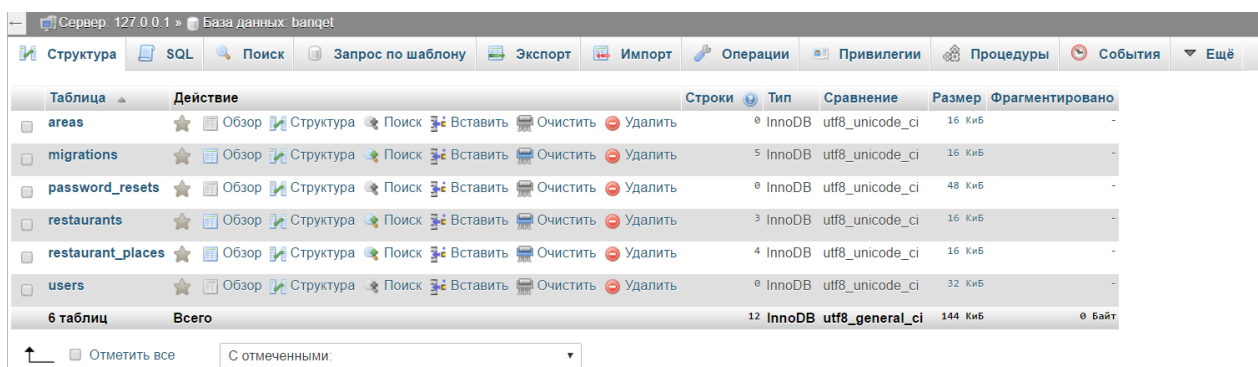
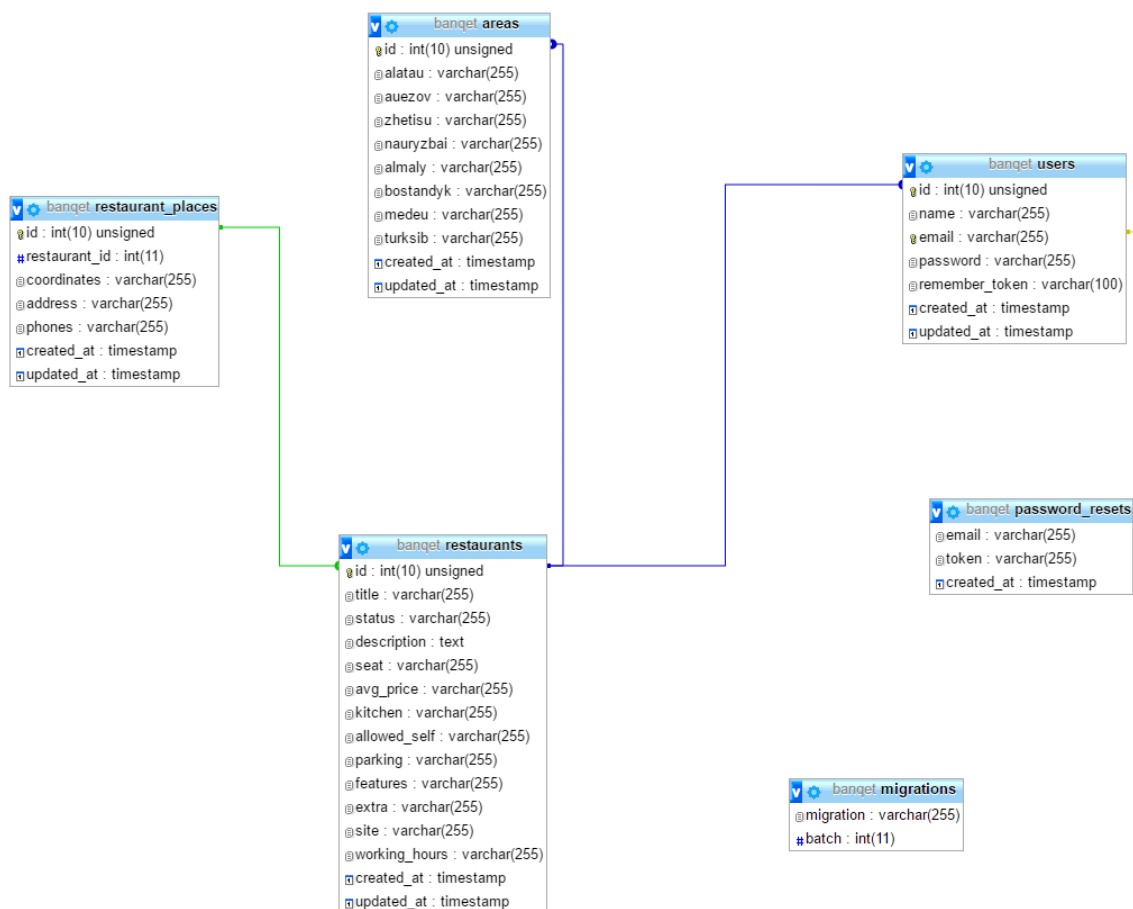


Таблица	Действие	Строки	Тип	Сравнение	Размер	Фрагментировано
areas	Обзор Структура Поиск Вставить Очистить Удалить	0	InnoDB	utf8_unicode_ci	16 Киб	-
migrations	Обзор Структура Поиск Вставить Очистить Удалить	5	InnoDB	utf8_unicode_ci	16 Киб	-
password_resets	Обзор Структура Поиск Вставить Очистить Удалить	0	InnoDB	utf8_unicode_ci	48 Киб	-
restaurants	Обзор Структура Поиск Вставить Очистить Удалить	3	InnoDB	utf8_unicode_ci	16 Киб	-
restaurant_places	Обзор Структура Поиск Вставить Очистить Удалить	4	InnoDB	utf8_unicode_ci	16 Киб	-
users	Обзор Структура Поиск Вставить Очистить Удалить	0	InnoDB	utf8_unicode_ci	32 Киб	-
6 таблиц	Всего	12	InnoDB	utf8_general_ci	144 Киб	0 Байт

3.17-сурет. “banqet” деректер қоры

Веб-қосымшаның деректер қорының архитектурасы мынадай түрде кескінделеді:



3.18-сурет. Деректер қорының сызбасы(ER диаграммасы)

Деректер қоры 6 кестеден тұрады:

– users – тіркелген адамдардың аттарының тізімі, почталарының адресі, құпия сөздері сақталады;

– password_resets – бұл кестеде құпия сөзді ұмытып қалған жағдайда қайта қалпына келтіруге қажетті деректер сақталады;

– restaurants – мейрамхана туралы ақпараттар сақталады. Мейрамхана атауы, статусы, мейрамхана туралы қысқаша анықтама, отыратын орындар саны, орташа бағасы, мейрамхана асханасы, тұтынушы өзімен бірге сырттан алып келуге рұқсат етілетін тағамдар тізімі, паркинг, артықшылықтары және т.б. ақпараттар осы кестеде сақталады;

– restaurant_places – мейрамхананың картадағы координатасы, адресі, телефон номері сақталады;

– areas – қала райондары сақталады;

– migrations – MySQL мен Laravel фреймворк арасындағы байланысты атқаратын миграция деректері сақталады.

Олар өзара барынша ыңғайлы және тиімді түрде байланысқан. Ол деректер қоры мен клиент арасындағы деректер алмасуын жылдамдататын факторлардың бірі болып табылады.

Деректер қорын біз Laravel фреймворкінің мүмкіндіктерін қолдана отырып мынадай кодпен жасадық:

```
1 <?php
2
3 use Illuminate\Database\Schema\Blueprint;
4 use Illuminate\Database\Migrations\Migration;
5
6 class CreateUsersTable extends Migration
7 {
8     /**
9      * Run the migrations.
10     *
11     * @return void
12     */
13     public function up()
14     {
15         Schema::create('users', function (Blueprint $table) {
16             $table->increments('id');
17             $table->string('name');
18             $table->string('email')->unique();
19             $table->string('password');
20             $table->rememberToken();
21             $table->timestamps();
22         });
23     }
24
25     /**
26     * Reverse the migrations.
27     *
28     * @return void
29     */
30     public function down()
31     {
32         Schema::drop('users');
33     }
34 }
```

3.19-сурет. “users” кестесін құратын код

```

1 <?php
2
3 use Illuminate\Database\Schema\Blueprint;
4 use Illuminate\Database\Migrations\Migration;
5
6 class CreatePasswordResetsTable extends Migration
7 {
8     /**
9      * Run the migrations.
10     *
11     * @return void
12     */
13     public function up()
14     {
15         Schema::create('password_resets', function (Blueprint $table) {
16             $table->string('email')->index();
17             $table->string('token')->index();
18             $table->timestamp('created_at');
19         });
20     }
21
22     /**
23     * Reverse the migrations.
24     *
25     * @return void
26     */
27     public function down()
28     {
29         Schema::drop('password_resets');
30     }
31 }
32

```

3.20-сурет. “passwords” кестесін құратын код

```

1 <?php
2
3 use Illuminate\Database\Schema\Blueprint;
4 use Illuminate\Database\Migrations\Migration;
5
6 class CreateRestaurantsTable extends Migration
7 {
8     /**
9      * Run the migrations.
10     *
11     * @return void
12     */
13     public function up()
14     {
15         Schema::create('restaurants', function (Blueprint $table) {
16             $table->increments('id');
17             $table->string('title');
18             $table->string('status');
19             $table->text('description');
20             $table->string('seat');
21             $table->string('avg_price');
22             $table->string('kitchen');
23             $table->string('allowed_self');
24             $table->string('parking');
25             $table->string('features');
26             $table->string('extra');
27             $table->string('site');
28             $table->string('working_hours');
29             $table->timestamps();
30         });
31     }
32
33     /**
34     * Reverse the migrations.
35     *
36     * @return void
37     */
38     public function down()
39     {
40         Schema::drop('restaurants');
41     }
42 }

```

3.21-сурет. “restaurants” кестесін құратын код

```

1 <?php
2
3 use Illuminate\Database\Schema\Blueprint;
4 use Illuminate\Database\Migrations\Migration;
5
6 class CreateRestaurantPlacesTable extends Migration
7 {
8     /**
9      * Run the migrations.
10     *
11     * @return void
12     */
13     public function up()
14     {
15         Schema::create('restaurant_places', function (Blueprint $table) {
16             $table->increments('id');
17             $table->integer('restaurant_id');
18             $table->string('coordinates');
19             $table->string('address');
20             $table->string('phones');
21             $table->timestamps();
22         });
23     }
24
25     /**
26     * Reverse the migrations.
27     *
28     * @return void
29     */
30     public function down()
31     {
32         Schema::drop('restaurant_places');
33     }
34 }

```

3.22-сурет. “restaurant_places” кестесін құратын код

```

3 use Illuminate\Database\Schema\Blueprint;
4 use Illuminate\Database\Migrations\Migration;
5
6 class Areas extends Migration
7 {
8     /**
9      * Run the migrations.
10     *
11     * @return void
12     */
13     public function up()
14     {
15         Schema::create('areas', function (Blueprint $table) {
16             $table->increments('id');
17             $table->string('alatau');
18             $table->string('auezov');
19             $table->string('zhetisu');
20             $table->string('nauryzbai');
21             $table->string('almaly');
22             $table->string('bostandyk');
23             $table->string('medeu');
24             $table->string('turksib');
25             $table->timestamps();
26         });
27     }
28
29     /**
30     * Reverse the migrations.
31     *
32     * @return void
33     */
34     public function down()
35     {
36         //
37     }
38 }

```

3.23-сурет. “areas” кестесін құратын код

3.3 Кестелерді құру, өзгерту, жою

Кестелерді деректер қорында құру үшін CREATE TABLE командасын қолдану керек.

Кестелерді құрудың синтаксисі келесідей болады:

CREATE TABLE кесте аты (бірінші баған аты типі, екінші баған аты типі,, соңғы баған аты типі);

Егер кестелерге өзгерістер енгізу керек болып жатса, онда Alter table командасын қолдану қажет.

Деректер қорынан кестені түгелімен жою қажет болса, Drop table командасын қолдану қажет.

3.4 Сұраныстар. Көріністер

Сұраныстар деректер қорындағы мәліметтерді көруді, енгізуді, жоюды немесе өзгертуді жеңілдетеді. Сұраныстарды қолданудың басқа мақсаттарына келесілерді жатқызсақ болады:

1) Белгілі шарттарды қолдану арқылы фильтрлеу әдісімен белгілі бір деректерді тез арада іздеу;

2) Деректерді санау немесе мәлімдеу;

3) Деректердің автоматты басқарылуы, мысалы өзекті деректерді тұрақты бақылау.

Деректер қорынан ақпараттарды алудың негізгі құралы — бұл сұраныс, яғни SELECT командасы болып табылады. Осы команданың негізгі мүмкіндіктері қарастырылатын болады. Сонымен қатар, MySQL Serverмен салыстырғандағы өзгешеліктер аталып кетеді.

Барлық басты деректер қорларында ANSI SQL стандарты қолданылатын болғандықтан, Oracleдағы және басқа да деректер қорындағы (Microsoft SQL Server, IBM DB2, Informix, Sybase, Microsoft Access) негізгі мүмкіндіктері сәйкес болады.

Oracle-дегі SELECT командасының басты синтаксисі келесідей болады:

SELECT [DISTINCT] бағандар тізімі

FROM дерек көзі WHERE фильтр ORDER BY сұрыптау өрнегі

Көрініс (ағыл. view) — көріністі қолдану кезінде ішкі сұраныс сияқты қолданатын атауланған сұратуды қолданатын виртуалды (логикалық) кесте.

Реляционды деректер қорының қарапайым кестелерінен өзгеше, көріністер дерекқорда сақталатын мәліметтер топтамасының бөлігі болмайды.

Көрініс мазмұны динамикалық түрде бар кестелердің мәліметтерін арқылы шығарылады. Бар кестедегі өзгеріс, осы кесте арқылы шығарылатын барлық көріністердің мазмұнын өзгертеді.

4. Техника-экономикалық негіздері

4.1 Еңбекақы шығындарын есептеу

Виртуалды белгіленген сервер құрастыруға қатысқан жұмысшылар еңбек ақысын төлеуге кеткен шығындар төмендегідей формула бойынша есептеледі

$$\Phi OT = Z_{осн} + Z_{доп} \quad (4.2)$$

бұл жерде $Z_{осн}$ – негізгі еңбек ақы, теңге;

$Z_{доп}$ – қосымша еңбек ақы, теңге.

Негізгі еңбек ақы көлемін есептеу үшін келесі ақпарат қажет:

- проект құрастыруда қатысқан жұмысшылар саны;
- қызметкер еңбекақысының айлық мөлшері;
- әрбір қызметкердің еңбекақысының бір сағаттық мөлшері;
- әрбір жұмыс түрінен еңбек көлемі;
- жобаны құрастыру кезеңін орындау циклінің ұзақтығы.

Қосымша еңбекақы (сияқы және т.б.) негізгі еңбек ақысынан 10% бөлігін құрайды және төменгі формуламен есептелінеді

$$Z_{доп} = Z_{осн} \cdot 0,1 \quad (4.3)$$

Әрбір жұмысшы бір сағат жұмыс еңбек ақысы өткен ай бойынша жұмыс күндер санының жұмысшының айлық еңбек ақысына және жұмыс күнінің сағат санына қатынасы ретінде анықталады. Төмендегідей формула бойынша анықталады

$$Z_q = \frac{Z}{N \cdot n} \quad (4.4)$$

бұл жерде Z – жұмысшының айлық еңбек ақысы, теңге;

N – айдағы жұмыс күндер саны, бес күндік жұмыс аптасы үшін $N = 21$ күн;

n – жұмыс күнінің ұзақтығы, $n = 8$ сағат.

Бір сағат жұмысының еңбек ақысы:

басқарушы үшін:

$$Z_q = \frac{90000}{21 \cdot 8} = 535,7 \text{ теңге/сағат};$$

жүйелік администратор үшін:

$$Z_q = \frac{100000}{21 \cdot 8} = 595,2 \text{ теңге/сағат};$$

техникалық қолдау маманы үшін:

$$z_q = \frac{60000}{21 \cdot 8} = 357,14 \text{ теңге/сағат.}$$

Әрбір кезең бойынша цикл ұзақтығын күн мөлшерінде анықтау жұмыстың әрбір түрі бойынша жобаны іске асыруға кеткен еңбек шығынын анықтауға мүмкіндігін береді

Әрбір жұмыс түрі күн бойынша цикл ұзақтығы дөнгелектенген және төмендегідей формула бойынша анықтаймыз

$$t_n = \frac{T}{q_n \cdot z \cdot K} \quad (4.5)$$

бұл жерде T – кезеңнің еңбек көлемі, норма-сағат;

q_n – кезең бойынша орындаушылар саны;

z – жұмыс күн ұзақтығы, $z = 8$ сағат;

K – норма уақыт бойынша орындау коэффициенті, $K = 1,1$.

Алынған t_n саны бүтін күндер санына дейін аздап дөнгелектеледі

$$t_1 = \frac{24}{1 \cdot 8 \cdot 1,1} \approx 3; \text{ Басқарушы, міндеттің қойылуы;}$$

$$t_1 = \frac{16}{1 \cdot 8 \cdot 1,1} \approx 2; \text{ Басқарушы, қажетті сервер туралы ақпаратты жинау;}$$

$$t_1 = \frac{8}{1 \cdot 8 \cdot 1,1} \approx 1; \text{ Басқарушы, есептеуіш орталықтағы Colocation-ға тапсырыс}$$

беру;

$$t_1 = \frac{24}{1 \cdot 8 \cdot 1,1} \approx 3; \text{ Жүйелік администратор, сервердің жұмысқа қабілеттігін}$$

тексеру;

$$t_1 = \frac{8}{1 \cdot 8 \cdot 1,1} \approx 1; \text{ Жүйелік администратор, есептеуіш орталыққа орнату;}$$

$$t_1 = \frac{8}{1 \cdot 8 \cdot 1,1} \approx 1; \text{ Жүйелік администратор, хост машинаға және клиент}$$

серверге операциялық жүйені орнату;

$$t_1 = \frac{8}{1 \cdot 7 \cdot 1,1} \approx 1; \text{ Жүйелік администратор, KVM гипервизорын орнату және}$$

баптау;

$$t_1 = \frac{16}{1 \cdot 8 \cdot 1,1} \approx 2; \text{ Техникалық қолдау маманы, виртуалы машина}$$

менеджмент жүйесін орнату;

$$t_1 = \frac{24}{1.8 \cdot 1,1} \approx 3; \text{ Жүйелік администратор, хост машина және клиент}$$

серверін тестілеу;

$$t_1 = \frac{16}{1.8 \cdot 1,1} \approx 2; \text{ Жүйелік администратор, хост машина және клиент}$$

серверді ретке келтіру;

$$t_1 = \frac{24}{1.8 \cdot 1,1} \approx 3; \text{ Техникалық қолдау маманы, хост машиналарды}$$

менеджмент және биллинг жүйесіне енгізу;

$$t_1 = \frac{8}{1.8 \cdot 1,1} \approx 1; \text{ Басқарушы, жаңа қызмет туралы ақпаратты енгізу және}$$

жарнамаға жіберу;

$$t_1 = \frac{8}{1.8 \cdot 1,1} \approx 1; \text{ Басқарушы, «Қазақстандағы VPS (VDS)» бөлімін}$$

дайындау;

$$t_1 = \frac{24}{1.8 \cdot 1,1} \approx 3; \text{ Басқарушы, «ТҚ» бөлімін дайындау;}$$

$$t_1 = \frac{24}{1.8 \cdot 1,1} \approx 3; \text{ Жүйелік администратор, ҒЗЖ рәсімдеу;}$$

$$t_1 = \frac{16}{2 \cdot 8 \cdot 1,1} \approx 1; \text{ Жүйелік администратор, есеп беруді тексеру мен тапсыру;}$$

$$t_1 = \frac{24}{2 \cdot 8 \cdot 1,1} \approx 2; \text{ Басқарушы, проверка и сдача отчета.}$$

4.5 кестесінде KVM технологиясын қолдану арқылы виртуалды белгіленген сервер платформасында хостингты құрастыру кезінде іске қатысқан жұмысшылар еңбек ақысына жұмсалатын қаражатты есептеудегі жиынтық нәтижелер көрсетілген.

Қобасымша еңбек ақы негізгі еңбек ақысының 10% құрайды және төмендегі формула бойынша есептеледі

$$Z_{\text{доп}} = 150476,19 \cdot \frac{10}{100} = 15047,62 \text{ теңге.}$$

Нәтижесінде жұмысшылардың жалпы еңбек ақы фондының көлемі:

$$\Phi OT = 150476,19 + 15047,62 = 165523,81 \text{ теңге.}$$

4.5 кесте – Негізгі еңбек ақысына шығынды есептейтін жиынтық нәтижелер

Жұмыстың аталуы және	Орындаушы	Еңбек көлемі	Цикл	Бір сағат	Жалпы еңбек
----------------------	-----------	--------------	------	-----------	-------------

мазмұны		Норма сағат	толық еңбек көлемінен бөлігі%	ұзақтығы, күн	жұмыс үшін еңбек ақы	ақы, теңге
1 Міндеттің қойылуы	Басқарушы	24	8,33	3	535,71	12857,14
2 Қажетті сервер туралы ақпаратты жинау	Басқарушы	16	5,56	2	535,71	8571,43
3 Есептеуіш орталықтағы Colocation-ға тапсырыс беру	Басқарушы	8	2,78	1	535,71	4285,71
4 Сервердің жұмысқа қабілеттігін тексеру	Жүйелік администратор	24	8,33	3	595,24	14285,71
5 Есептеуіш орталыққа орнату	Жүйелік администратор	8	2,78	1	595,24	4761,90
6 Хост машинаға және клиент серверге операциялық жүйені орнату	Жүйелік администратор	8	2,78	1	595,24	4761,90
7 KVM гипервизорын орнату және баптау	Жүйелік администратор	8	2,78	1	595,24	4761,90
8 Виртуалы машина менеджмент жүйесің орнату	Техникалық қолдау маманы	16	5,56	2	357,14	5714,29
9 Хост машина және клиент серверін тестілеу	Жүйелік администратор	24	8,33	3	595,24	14285,71

4.5 кестенің соңы

Жұмыстың аталуы және	Орындаушы	Еңбек көлемі	Цикл	Бір сағат	Жалпы еңбек
----------------------	-----------	--------------	------	-----------	-------------

мазмұны		Норма сағат	толық еңбек көлемінен бөлігі %	ұзақтығы, күн	жұмыс үшін еңбек ақы	ақы, теңге
10 Хост машина және клиент серверді ретке келтіру	Жүйелік администратор	16	5,56	2	595,24	9523,81
11 Хост машиналарды менеджмент және биллинг жүйесіне енгізу	Техникалық қолдау маманы	24	8,33	3	357,14	8571,43
12 Жаңа қызмет туралы ақпаратты енгізу және жарнамаға жіберу	Басқарушы	8	2,78	1	535,71	4285,71
13 «Қазақстандағы VPS (VDS)» бөлімін дайындау	Басқарушы	8	2,78	1	535,71	4285,71
14 «ТҚ» бөлімін дайындау	Басқарушы	24	8,33	3	535,71	12857,14
15 ҒЗЖ рәсімдеу	Жүйелік администратор	24	8,33	3	595,24	14285,71
16 Есеп беруді тексеру мен тапсыру	Жүйелік администратор	16	5,56	1	595,24	9523,81
	Басқарушы	24	8,33	2	535,71	12857,14
Барлығы		280	100	33		150476,19

4.2 Әлеуметтік алым бойынша шығындарды есептеу

Әлеуметтік алым жұмысшы табысының 11% (ҚР СК ст. 358 п. 1) құрайды және төмендегі формула бойынша есептеледі

$$Oc = (\Phi OT - ПO) \cdot 0,11 \quad (4.6)$$

Бұл жерде $ПО$ зейнеткерлік алымдар, ол еңбек ақы қорының 10% құрайды және оған әлеуметтік салық салынбайды, төмендегідей формула бойынша есептеледі

$$ПО = ФОР \cdot 0,1 \quad (4.7)$$

$$ПО = 165523,81 \cdot 0,1 = 16552,38 \text{ теңге}$$

Нәтижесінде әлеуметтік қажеттілікке алымдар көлемі:

$$Ос = (165523,81 - 16552,38) \cdot 0,11 = 16386,86 \text{ теңге}$$

4.3 Амортизациялық алымдар шығынын есептеу

Амортизациялық алымдар төмендегідей формула бойынша есептеледі

$$A_i = \frac{H_A \cdot C_{ПЕР} \cdot N}{100 \cdot 12 \cdot n} \quad (4.8)$$

бұл жерде H_A - амортизациялау нормасы;

$C_{ПЕР}$ - құрал жабдықтардың бастапқы құны;

N - жұмысты орындауға кеткен күндер саны;

n - жұмыс айының күндер саны.

Серверлік техникаға H_A амортизациялау нормасы құрал жабдықтың бастапқы құнының 15% құрайды [9].

Нәтижесінде 4.8 формула бойынша қолданылатын құрал жабдықтардың амортизациялық алымдар көлемі:

Клиент сервер HP ProLiant DL120 G7:

$$A_1 = \frac{15 \cdot 145000 \cdot 33}{100 \cdot 12 \cdot 21} = 2848,21 \text{ теңге.}$$

Хост машина HP ProLiant DL360p Gen8 Base:

$$A_2 = \frac{15 \cdot 738000 \cdot 33}{100 \cdot 12 \cdot 21} = 14496,43 \text{ теңге.}$$

Қатты диск HP Enterprise:

$$A_3 = \frac{15 \cdot 76350 \cdot 33}{100 \cdot 12 \cdot 21} = 1499,73 \text{ теңге.}$$

Қатты диск HP Dual Port Enterprise:

$$A_4 = \frac{15 \cdot 48450 \cdot 33}{100 \cdot 12 \cdot 21} = 951,69 \text{ теңге.}$$

Қатты диск HP Dual Port Midline

$$A_5 = \frac{15 \cdot 99750 \cdot 33}{100 \cdot 12 \cdot 21} = 1959,38 \text{ теңге.}$$

Жедел жады Kingston HyperX memory

$$A_6 = \frac{15 \cdot 22500 \cdot 33}{100 \cdot 12 \cdot 21} = 441,96 \text{ теңге.}$$

4.6 кестеде амортизациялық алымдар есептеудегі жиынтық нәтижелер көрсетілген.

4.6 к е с т е – амортизация шығыны бойынша мәліметтер

Наименование изделий	Бірлік саны	Бірлік құны, теңге	Амортизация нормасы, %	Бірлік үшін амортизация құны, теңге	Жалпы сомма, теңге
1 Клиент сервер HP ProLiant DL120 G7	1	145000	15	2848,21	2848,21
2 Хост машина HP ProLiant DL360p Gen8 Base	1	738000	15	14496,43	14496,43
3 Қатты диск HP Enterprise	8	76350	15	1499,73	11997,86
4 Қатты диск HP Dual Port Enterprise	2	48450	15	951,70	1903,39
5 Қатты диск HP Dual Port Midline	2	99750	15	1959,38	3918,75
6 Жедел жады Kingston Hyper Xmemory	2	22500	15	441,96	883,93
Барлығы					36048,57

4.4 Электр энергия шығының есептеу

Өндіріс процесінде электр құрылғылар қолданылатындықтан, электр энергиясына шығынды есептеу қажет. Өндірістік қажеттілік үшін электр энергия шығын құрамына құрылғыларға жұмсалатын электр энерия және қосымша қажеттіліктер кіреді [10]

$$\mathcal{E} = \mathcal{Z}_{ЭЛ.+ЭН.+ОБОР.} + \mathcal{Z}_{ДОП.НУЖ.} \quad (4.9)$$

бұл жерде $\mathcal{Z}_{ЭЛ.+ЭН.+ОБОР.}$ - құрылғыларға жұмсалатын электр энерия %;

$\mathcal{Z}_{ДОП.НУЖ.}$ - қосымша қажеттіліктер үшін жұмсалатын электр энерия шығыны;

Құрылғыларға жұмсалатын электр энергия бойынша шығындар төмендегі формула бойынша есептелінеді

$$Z_{ЭЛ.+ЭН.+ОБОР.} = W \cdot T \cdot S \cdot K_{ИСП} \quad (4.10)$$

бұл жерде W - тұтынылатын қуат, Вт;

T- жұмыс уақыты;

S-тариф (1кВт = 12,34тг/кВт*сағат);

$K_{ИСП}$ - қолдану коэффициенті ($K_{ИСП}=0,9$).

$$Z_{ЭЛ.+ЭН.+ОБОР.-1} = 0,4 \cdot 840 \cdot 12,34 \cdot 0,9 = 3731,62 \text{ теңге}$$

$$Z_{ЭЛ.+ЭН.+ОБОР.-2} = 0,46 \cdot 840 \cdot 12,34 \cdot 0,9 = 4291,36 \text{ теңге}$$

Негізгі құрал жабдықтарға жұмсалатын электр энергия соммасы:

$$Z_{ЭЛ.+ЭН.+ОБОР.} = 3731,62 + 4291,36 = 8022,97 \text{ теңге}$$

Қосымша қажеттіліктер үшін жұмсалатын сомма 5% деңгейінде дөнгелектелген көрсеткіш бойынша алынады:

$$Z_{ДОП.НУЖ.} = 8022,97 \cdot 0,05 = 401,15 \text{ теңге}$$

Нәтижесінде электр энергия шығынының жалпы соммасы:

$$\Sigma = 8022,97 + 401,15 = 8424,12 \text{ теңге}$$

4.7 кестеде электр энергия шығыны есептеудегі жиынтық нәтижелер көрсетілген.

4.7 кесте – Электр энергия шығыны

Құрылғылар аты	W, кВт	Жұмыс күндерінің саны	$K_{ИСП}$	Құрылғының жұмыс істеу уақыты, сағат	ΣW кВт*ч	Құны, теңге
Клиент сервер HP ProLiant DL120 G7	0,4	35	0,9	840	336	3731,62
Хост машина HP ProLiant DL360p Gen8 Base	0,46	35	0,9	840	386,4	4291,36
Барлығы					722,4	8022,9744

4.5 Пайдалану шығындары

Пайдалану шығындары барлық шығындардың 25% құрайды және төмендегідей формула бойынша есептеледі

$$H = (\Phi OT + O_C + A + \mathcal{E} + C_{PP}) \cdot 0,25 \quad (4.11)$$

$$H = (165523,81 + 16386,86 + 36048,57 + 8424,12) \cdot 0,25 = 56595,84 \text{ теңге}$$

4.6 Барлық шығындар тармақтары бойынша құн есептеу

4.1 формула бойынша виртуалды сервер құрастыру өнімін шығару үшін шығындардың құны:

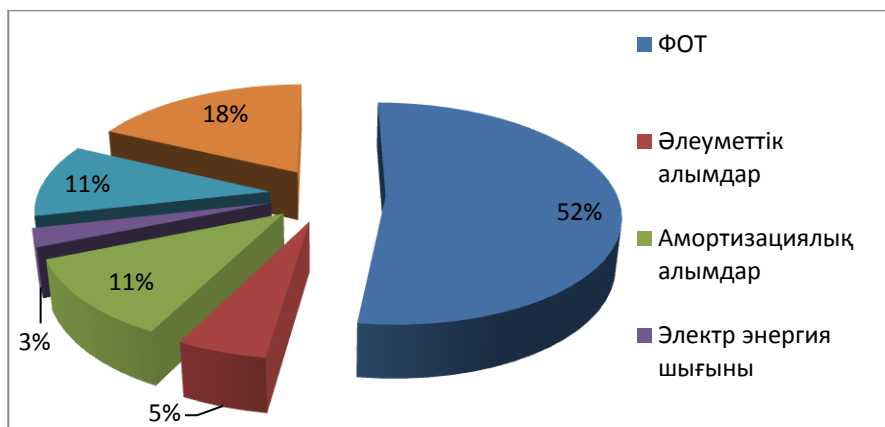
$$C = 165523,81 + 16386,86 + 36048,57 + 8424,12 + 33777 + 56595,84 = 316756,20 \text{ теңге}$$

4.8 кестесінде KVM технологиясын қолдану арқылы виртуалды белгіленген сервер платформасында хостингты құрастыру шығындарын есептеудегі жиынтық нәтижелер көрсетілген [11].

4.8 кесте – Барлық шығындар тармақтары бойынша KVM технологиясын қолдану арқылы виртуалды белгіленген сервер платформасында хостингты құрастыру құны

Шығын тармағының аты	Шығын соммасы, тенге
ФОТ	165523,81
Әлеуметтік алымдар	16386,86
Амортизациялық алымдар	36048,57143
Электр энергия шығыны	8424,12312
Бағдарламалық қамтамасыздандыру	33777
Пайдалану шығындары	56595,84
Барлығы	316756,20

4.1 суретінде KVM технологиясын қолдану арқылы виртуалды белгіленген сервер платформасында хостингты құрастыру шығындарын есептеудегі жиынтық нәтижелері графигалық түрде көрсетілген.



Сурет 4.1 – Хостинг қызметін құрастыру бойынша шығындар құрылымы

4.7 Ителектуалды еңбек құны

Вируалды белгіленген сервер қызметін жүзеге асыру құны баға және таза табыстан құралады, және төмендегідей формула бойынша есептеледі

$$Ц=C+П \quad (4.12)$$

бұл жерде: С – қызмет бағасы;

П – таза табыс;

Виртуалды белгіленген сервер қызметін жүзеге асыру үшін бастапқы бағаны анықтау кезінде күтілетін рентабельділік (30%) деңгейімен берілу керек (4.13 формуласы)

$$Ц_{п} = C \cdot \left(1 + \frac{P}{100}\right) \quad (4.13)$$

бұл жерде P – рентабельділік (30%);

$$Ц_{п} = 316756,20 \cdot \left(1 + \frac{30}{100}\right) = 411783,06 \text{ теңге}$$

Дайын қызметтің жүзеге асыру бағасы төмендегі формула бойынша есептелінеді

$$Ц_{р} = Ц_{п} + ҚҚС \quad (4.14)$$

12% ставкасымен есептелінетін қосымша құн салығы төмендегідей формуламен есептелінеді

$$ҚҚС = Ц_{п} \cdot 0,12 \quad (4.15)$$

$$ҚҚС = 411783,06 \cdot 0,12 = 49413,97 \text{ теңге};$$

$$Ц_{р} = 411783,06 + 49413,97 = 461197,03 \text{ теңге}.$$

5. Тіршілік қауіпсіздігі

5.1. Программисттің жұмысына әсер ететін негізгі қауіпті әрі зиянды факторларды талдау

Дербес компьютерде жұмыс жасау бұл – қолданушымен тез әрі нақты қабылданатын ақпаратты диспейде көрсету.

ЭЕМ жұмыс жасайтын адамдардың еңбек нәтижесіне тікелей әсер ететін фактор – жайлылық және қауіпсіздік.

Дербес компьютермен жұмыс жасайтын еңбек тұтынушысының шарты анықталады:

- 1) жұмыс орнының ұйымдастырылу ерекшелігімен;
- 2) өндірістік ортаның жағдайымен (жарықтандырумен, микроклиматпен, шуммен, электромагниттік және электростатикалық өріспен, дисплейдің әртүрлі параметрлерімен, т.с.с.);
- 3) адам мен ЭЕМ-ның өзара іс-қимыл мінездемелерімен.

Қызметкерлердің жұмысы тікелей компьютермен байланысты, сондықтан олардың қызмет тиімділігін айтарлықтай төмендетер қосымша факторлар өте көп. Олар:

- 1) дұрыс емес жарықтандыру;
- 2) микроклиматтың бұзылуы;
- 3) қысымның болуы.

Дербес компьютер қолданушыларында пайда болатын жағымсыз әсерлер мен ауыруды басу үшін жиі үзілістер және ыңғайлы позада жұмыс жасауға және босансуға мүмкіндік беретіндей жұмыс орнын дұрыс орналастыру қажет.

Компьютер пайдаланушыларының денсаулығын нашарлататын тағы бір фактор – электромагниттік және электростатикалық өріс, акустикалық шу, ауаның иондық құрамы мен микроклиматтың өзгеруі. Жұмыс орнының жарықтығы, монитор мен орындықтың орналасуы да біршама рөл атқарады.

Психофизиологиялық қауіпті әрі зиянды факторлар: назар мен көріністің қиындығы; интеллектуалды, эмоционалды және ұзақ уақыттық бірқалыпты қысым; еңбектің бірыңғайлығы; уақыт бірлігінде орындалатын үлкен көлемдегі ақпарат; жұмыс орнының тұрақсыз орналасуы.

Кез-келген ДК қолданушысы жұмыс аяғында сезінетін белгілі сезімдер: көздің шаршауы, бастың ауыруы, мойын бұлшықеттерінің ауыруы, назар аударудың нашарлауы.

ГОСТ 12.1.005-88 ССБТ «Жұмыс категориясына байланысты тиімді және рұқсат етілетін микроклимат нормаларына» сүйенсек, бөлмеде қызмет жасау жеңіл 1а категориясына жатады. Бөлмеде ЭЕМ-мен жұмыс жасау барысында мынадай қағидаларды сақтау қажет:

Суық мезгілде:

– бөлменің тиімді температурасы 22-24 С°, рұқсат етілетін температура 18-26 С°;

– салыстырмалы ауа ылғалдылығы 40-60%, 75% рұқсат етіледі;

- салыстрмалы және рұқсат етілетін ауа қозғалысы жылдамдығы 0,1м/с.
- Жылы мезгілде:
- бөлменің тиімді температурасы 23-25 С°, рұқсат етілетін температура 20-30 С°;
- салыстырмалы ауа ылғалдылығы 40-60%, 55% рұқсат етіледі;
- салыстрмалы және рұқсат етілетін ауа қозғалысы жылдамдығы 0,1м/с.

5.2. Дербес компьютердің шу шығару деңгейін есептеу

Жұмыстың мақсаты – тұтынушылар үшін үстелді ғаламтор арқылы қолайлы түрде броннау қызметін ұсыну. Бұл қызмет түрі мейрамхана қолданушыларына үлкен ыңғайлылықтар мен көптеген ақпараттар береді.

Жұмыс барысында мынадай конфигурациялы компьютер қолданылады:

- Аналық тақта: Intel HM76;
- Процессор: Intel Core i7-3610QM 2.3GHz;
- Жедел жады: 8Gb;
- Қатқыл жады: Seagate Momentus SpinPoint 1Tb;
- Видеокарта: NVIDIA GeForce GTX 660M 2024Mb;

Негізгі шу шығаратын құрылғы процессор болып табылады. Сонымен қатар DVD/CD, қатқыл жады және кулерлер де шу шығарады. Принтер және сканер де шу шығарғанымен, олар өте сирек қолданылғандықтан есепке алған жоқпын.

Қатқыл жады. Жүйеде 1 жады қолданылады, айналым саны 5400айн/мин. Әдетте қатқыл диск 25дБ шу шығарады, жұмыс барысында 32дБ-ге дейін барады.

DVD/CD қосымшасы. Жүйеде 1 DVD-RW құрылғысы орнатылған. Ол максималды жұмыс жасау кезінде 40-45дБ шу шағарады, дегенмен қазіргі кезде бұл құрылғы да сирек қолданылады.

Процессорда орналасқан жүйедегі жалғыз кулер Ø 60мм, 2700айн/мин, 22дБ шу шығарады.

Жақын орналасқан шу шығару орталықтарының жалпы шу деңгейі келесідей анықталады:

1. Егер екі шу шығару көздерінің деңгейі шамалас болса, әрқайсының шу шығару деңгейі 3дБ-ге артады.

2. Егер айырмашылық 10дБ болса, жалпы шу шығару деңгейі екі көздің үлкеніне тең болады.

3. Егер айырмашылықтар 10дБ-ге жетпесе, төменде келтірілген кестеге сүйеніп есептейміз. Шу шығару көздері 2-ден көп болса да есептеу әдісі өзгермейді.

Сонымен, үнемі шу шығарып отыратын шу көздерін есепке алайық(бірен-саран қолданылатын құрылғылар есепке алынбайды).

1-кесте. Шу шығару көздері туралы мәліметтер

Шу көзі	Саны	Диаметр	Жиілігі	Шу деңгейі
---------	------	---------	---------	------------

Процессордағы кулер	1 дана	60мм	2700айн/мин	22дБ
Қатқыл жады	1 дана	50мм	5400айн/мин	32дБ

Жалпы шу шығару деңгейін анықтау формуласы

$$L_{x,y} = L_y + \Delta L$$

бұл жерде L_y – үлкен шу көзі,

ΔL – кестеде көрсетілген мән.

Демек,

$$L_{1,2} = 22\text{дБ} + 2,6\text{дБ} = 24,6\text{дБ}.$$

$$L_{2,3} = 32\text{дБ} + 2,1\text{дБ} = 34,1\text{дБ}.$$

Жалпы шу шығару деңгейі 34,1дБ-ге тең болады.

Алынған есептеулер нәтижесіне сүйенсек, шу шығару деңгейі рұқсат етілген шамадан аспайды.

Операторлардың едәуір ақыл-ой қуаты және жүйке-эмоционалдық жүктеме компьютермен жұмыс істеумен характерлінеді, компьютердің пернетақтасымен жұмыс істегенде едәуір үлкен көру қуатымен және қолдың және белдің бұлшық еттерге едәуір үлкен жүктме болады. Жұмыс столдың элементтерінің және құрылыстың рационалды жайғастырылуы маңызы үлкен, сонымен қатар адам жұмысының оптималды қалып формасын жықтауы өте қажетті.

5.3 Жарықтандыруды бағалау

Аудиторияда табиғи жарықтандыру көзі ретінде 1000x2000 мм өлшемді 1 терезе бар. Жарық өтетін материал – қағазды әйнек. Ашылу түрі – 2 жаққа ашылу. Күнге қарсы құрылғылар – тұрақты жалюздер мен перделер.

Бұлтты күні мен қысқы күні жұмыс уақыты кезінде керек жарықты шығару үшін жасанды жарық көзі қолданылады. Жасанды жарық көзі ретінде люминесценттік шамдар қолданылады.

Бұл бөлмедегі жұмыс жасау үшін керек жарық: 300 лк.

СНиП РК 2.04.-05-2002 бойынша жарықтандыру нормасы 5.3-кестеде көрсетілген.

5.3-к е с т е – Ұсынылған жарықтану (СНиП РК 2.04.-05-2002)

Түсті айыруға талап бойынша көру жұмысының сипаттамасы	Жарықтану, лк	Жарық көзінің түстаратуын ың максималды индексі, Ra	Жарық көзінің температурасын ың түстер диапазоны	Жарық көзінің қолданатын түрі
Түсті айыруға жоғары емес талап	≥500	50	3500-6000	ЛБ, ЛХБ, МГЛ
	300, 400	50	3500-5500	ЛБ, НЛВД+МТЛ

бойынша түсті объектілерді айыру	150, 20	45	3000-4500	ЛБ, ЛХБ, НЛВД+МТЛ, ДРЛ
	150 \geq	40	2700-3500	ЛБ, ДРЛ, НЛВД+МТЛ (ЛН, КГ)

5.4 Жасанды жарықтандыруды есептеу

Жарық көзі адам өмір сүруінің ең маңызды шарттарының бірі болып табылады. Ол ағза жұмысына әсер етеді, дұрыс орнатылған жарық жоғарға нер қызметіндегі процесстердің жүруіне жағдай жасайды және жұмыс істеу қабілетін көтереді. Жарық көзі жеткіліксіз болған жағдайда адам жұмысы өнімсіз болады, тез шаршайды, қате шешім қабылдау ықтималдылығы өседі, мұның өзі жарақаттану ықтималдылығын асырады.

Бөлме офистің күн түспейтін жақта орналасқандықтан, жасанды жарықтандыруды қолданамыз. Аудиторияның өлшемдері: ұзындығы 6 м, ені 3 м, биіктігі 4 м.

Есептеу негізінен екі әдіспен орындалады: қолдану коэффициенті әдісі және нүктелік әдіс.

- Қолдану коэффициенті әдісі – үлкен көлеңке түсіретін заттар болмаған жағдайда горизонталды беттерді біркелкі жарықтандыруды жалпы есептеуге арналған.

- Ал нүктелік әдіс бойынша – көлеңке түсіретін заттар болған кезде, жалпы локалданған және біркелкі жалпы жарықтандыруды есептеуге арналған.

5.4.1 Жарықтандыруды коэффициентті қолдану әдісімен есептеу

Бұл әдіс бөлмедегі горизонталдық беттегі ортақ бірқалыпты жарықтандыруды есептеуге арналған. Толқын ұзындығына байланысты жарық (қызғылт сары-қызыл) қоздырушы немесе жайландырушы (сары -жасыл) әсер көрсетеді.

Еңбек шарттарының жайлы болуы үшін өндірістік жарық келесі талаптарға сай болады:

- жұмыс орнындағы жарықтандыру гигиеналық нормаларға сәйкес;
- жұмыс бетіне және қоршаған кеңістік шегінде жарықтық бірқалыпты таралады;

- жұмыс бетінде айқын көлеңкелер жоқ, олардың болуы жарықтықтың бірқалыпсыз таралуын тудырады. Жылтырау (тура немесе шағылған) жазықтықта жоқ;

- жарық заттың түсін дұрыс беруге қажетті жарық спектрлерімен қамтамасыз етеді.

Жарықтандыруды коэффициентті қолдану әдісімен есептеу үшін мына формуланы қолданамыз

$$\Phi = \frac{E \cdot S \cdot k_z \cdot Z}{N \cdot \eta} \quad (5.1)$$

мұнда Φ – әр шамдағы қажетті жарықтық арна, лм;
 E – нормалық минималды жарықтану, лк, кестеден алынады;
 k_z – қор коэффициенті;
 S – жарықтандыратылатын аудан, кв м;
 z – 1,1 мен 1,5 арасындағы жарықтандырудың минималды коэффициенті (оптималды жағдайда қыздыру шамдар мен ДРЛ шамдар арасындағы қашықтық $z=1,15$ және шамдар үшін $z=1,2$);
 N – аудиториядағы шамдар саны;
 η – жарықтық арнаны қолдану коэффициенті.
 Көру жұмысының дәрежесі – V. Жарықтандырудың нормалауы 5.4-кестеде келтірілген – 300 лк.

5.4-кесте – Көру жұмысының дәрежесі (СНиП РК 2.04.-0.5-2002)

Заттың минималады ажыратылатын аралығы	Заттан жұмысшы көзіне шейін аралығы	Көру жұмысының дәрежесі
1-10 мм	500 мм	V

$E=300$ лк; $k_z=1,2$; $Z=1,2$ қолданамыз.

Жарықтандыру үшін газразрядты шамдарды қолданамыз.

Жарықтандырылатын аудан мына формуламен есептелінеді

$$S = A \cdot B$$

мұнда S – жарықтандырылатын аудан, м²;

A – аудиторияның ұзындығы, м;

B – аудиторияның ені, м.

$$S=6 \cdot 3=18 \text{ м}^2.$$

Ілгіштің есептік биіктігі мына формуламен анықталады

$$h = H - h_{\%} - h_p$$

мұнда H – аудиторияның биіктігі, м;

$h_{\%}$ – шамның іліну биіктігі, м;

h_p – жұмыс орнының еденнен биіктігі, м.

$H=4$ м, $h_{\%}=0,7$ м, $h_p=0,3$ м.

$h=4-0,7-0,3=3,0$ м.

Шамдар арасындағы арақашықтықты анықтаймыз

$$L = \lambda \cdot h$$

$$L=1,2 \cdot 3,0=3,2 \text{ м.}$$

Қабырғадан жақын шамға дейінгі арақашықтық:

$$l = (0,25 \div 0,3) \cdot L$$

$$l = 0,25 \cdot 3,6=0,9 \text{ м.}$$

η қолдану коэффициентін табу үшін i бөлме индексін табамыз

$$i = \frac{A \cdot B}{h \cdot (A + B)}$$

мұнда h – ілгіштің есептік биіктігі, м.

$$i = \frac{6 \cdot 3}{3 \cdot (6 + 3)} = 0,66.$$

Шыққан мәнді жуықтап аламыз, сонда $i=1$ болады.

$$\rho_e = 70\%, \rho_k = 50\%, \rho_{жс} = 30\%.$$

ρ_e – еден, ρ_k – қабырға, $\rho_{жс}$ - жұмыс орнының беті.

Берілген шамға i мен ρ мәндерін табамыз

ЛБ40-4 шам түрін аламыз, оның $\eta=0.70$ %-ке тең.

Формула (5.1) бойынша әр шамға керек жарық ағынын табамыз:

$$\Phi = \frac{300 \cdot 18 \cdot 3 \cdot 1.2}{2 \cdot 0.70} = 12457$$

Люминесценттік лампалардың санын мын формуламен анықтаймыз

$$N_1 = \frac{E \cdot k_z \cdot S \cdot Z}{\Phi \cdot \eta}$$

мұнда k_z – қор коэффициенті;

E – берілген жарықтандырудың миналы, $E=300$ лк.;

Z – жарықтандырудың әркелкілік коэффициенті, $Z=1,2$;

Φ – лампаның жарық ағанының коэффициенті, $\Phi=3120$ лм.;

η - пайдалу коэффициенті, $\eta=70\%$.

$$N_1 = \frac{300 \cdot 3 \cdot 18 \cdot 1.2}{3120 \cdot 0.7} = 7.97 \approx 8.$$

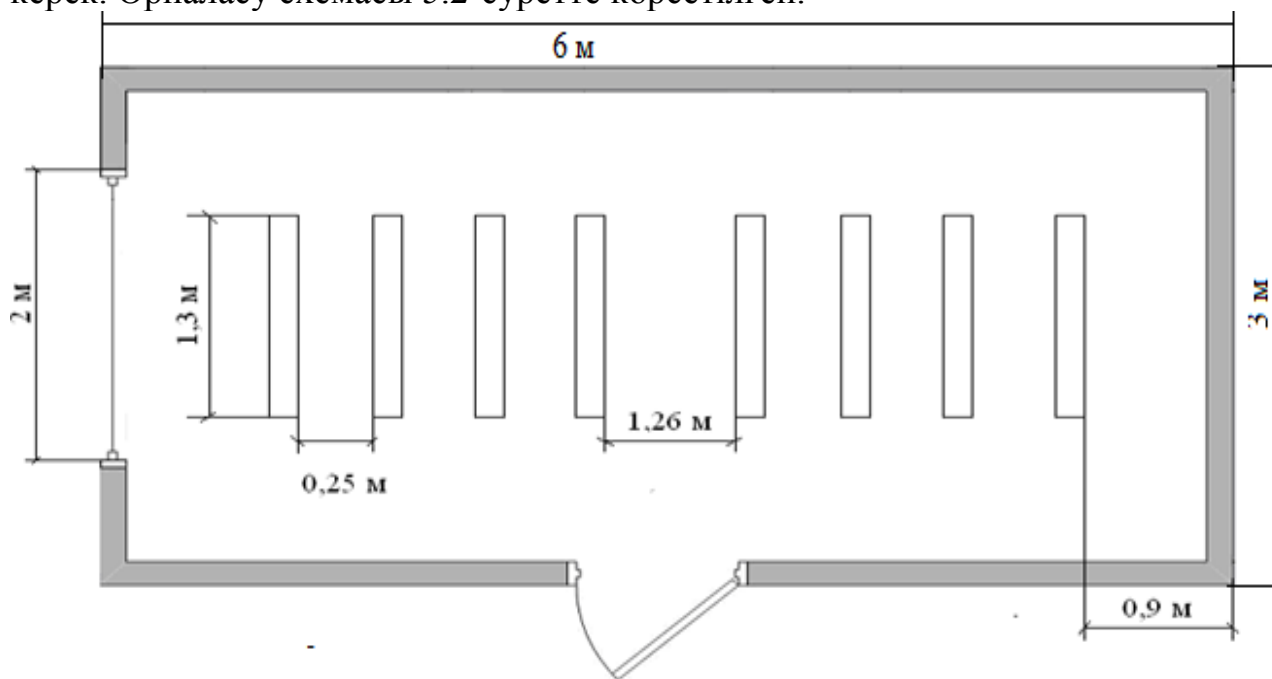
Таңдалған лампа түрі – ЛБ40-4. Оның сипаттамлары 5.5-кестеде көрсетілген.

5.5-к е с т е – ЛБ40-4 шамның техникалық сипаттамасы

Қуат	Шамның жарық	Масса,	Габариті, мм	Жану
------	--------------	--------	--------------	------

ы, Вт	ағынының номиналы, лм	кг	Ұзындығы, мм	Ені, мм	Биіктігі, мм	сұлбасы
40	3120	2	1300	100	125	C(1x40)

Есептеулерден көріп отырғандай, бөлмені жарықтандыру үшін 8 шам керек. Орналасу схемасы 5.2-суретте көрсетілген.

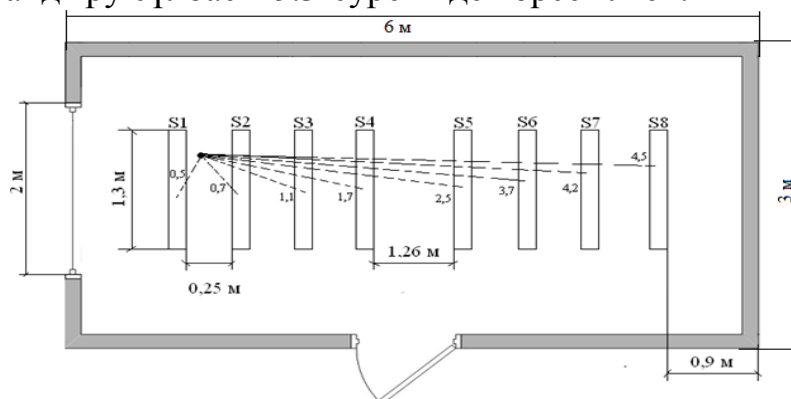


5.2-сурет – Шамдардың орналасу жобасы

5.4.2 Жарықтандыруды нүктелік әдіс бойынша есептеу

Көру жұмысының дәрежесі– V. Жарықтандырудың нормалауы 5.4 кетесе келтірілген – 300 лк. Жалпы жарықтандыру жүйесінде люминесцендык параметры ЛБ40-4 шам қолданылады, 5.5 кетесінде оның техникалық сипаттамасы көрсетілген. Шамның іліну биіктігі $h=3$ м, қор коэффициенті $k_Z = 1,2$ тең.

Жарықтандыру сұлбасы 5.3-суретінде көрсетілген.



5.3-сурет – Жарықтандырылу есептеуінің сұлбасы (S1-S2 – шамдар)

О бақылау нүктесін белгілейміз. Әр шам үшін бақылау нүктесіне дейін арақашықтық анықталады, содан кеңістік изолюкс шарты графигі бойынша әр шам үшін бақылау нүктесіндегі жарықтандырудың шартын табамыз.

Бірінші шам:

$$S_1=0,5 \text{ м};$$

$$E_1=16 \text{ лк.}$$

Екінші шам:

$$S_2=0,7 \text{ м};$$

$$E_2=14 \text{ лк.}$$

Үшінші шам:

$$S_3=1,1 \text{ м};$$

$$E_3=10 \text{ лк.}$$

Төртінші шам:

$$S_4=1,7 \text{ м};$$

$$E_4=9 \text{ лк.}$$

Бесінші шам:

$$S_5=2,5 \text{ м};$$

$$E_5=8 \text{ лк.}$$

Алтыншы шам:

$$S_6=3,7 \text{ м};$$

$$E_6=7 \text{ лк.}$$

Жетінші шам:

$$S_7=4,2 \text{ м};$$

$$E_7=6 \text{ лк.}$$

Сегізінші шам:

$$S_8=4,5 \text{ м};$$

$$E_8=5 \text{ лк.}$$

Барлық шамдардың О бақылау нүктесіндегі жарықтандырудың жалпы шамасын есептейміз ΣE_0 , лк.

$$\Sigma E_0 = E_1 + E_2 + E_3 + E_4 + E_5 + E_6 + E_7 + E_8$$

$$\Sigma E_0 = 16 + 14 + 10 + 9 + 8 + 7 + 6 + 5 = 75 \text{ лк.}$$

Берілген формула бойынша жарық ағынының есептейміз

$$\phi = \frac{1000 \cdot E \cdot \kappa_z}{\mu \cdot \Sigma E_0}$$

мұнда, μ – коэффициент, «алыстатылған» шамдар әрекетін есептейді (1,1-1,2 коэффициенті алынады);

E – бөлменің жарықтандыру нормасы.

$$\phi = \frac{1000 \cdot 300 \cdot 1,2}{1,15 \cdot 75} = 4173,91 \text{ лм.}$$

Демек, жұмыс орнының дәл үстінен 2 ЛБ40-4 шамндарын орнату қажет, әр-қайсысы 3000 лм жарық ағынымен [14].

5.4. Серверлік бөлменің вентиляция жүйесін есептеу

5.4.1. Вентиляцияны есептеу

Жылы не суық ауа адамдардан тыс бағытында, жылыту жүйесі және кондиционерлеу жүйесі орнатылуы тиісті. Өндірісте динамикалық климатты белгілі деңгей ауысуымен бірге жасауды ұсынады. Ауа температурасы беттің нобайлы бейнесі және адам басының деңгейінде бір-бірімен 5 градустан ажыратылуы болдырмауға керек. Вентиляция жүйесінің характеристикасын сипаттайтын басты параметрлері, еселік аурбастауы болады, ол деген ауаның бөлмедегі қаншы рет ауысқаны.

Ауа асуының еселегін 6м x 3м x 3м бір терезесімен бар бөлмені есептейік, терезенің өлшемдері 2м x 1м болып келеді. Серверлік бөлмеде бір адам жұмыс істейді. 5.6-кестедегі көрсетілген электржабдығықтарды қолданамыз.

5.6-к е с т е – Қолданылатын электржабдығықтар

Атауы	Саны (дана)	Қолданылатын қуат, Вт
Дербес компьютер	6	600
Тоқтаусыз қоректендірі көзі	3	700
Firewall – Cisco Firewall PIX 515	1	170
Маршрутизатор – Cisco 2811 Voice Bundle w/ PVDM2-16,FL-CCME-36,SP Serv,64F/256D	3	100

Ауа асуының еселегін мына формула бойынша есептейміз

$$K = \pm \frac{V_{\text{вент}}}{V_{\text{бл}}}$$

мұнда $V_{\text{вент}}$ – бөлмеге берілетін ауа көлемі, м³;

$V_{\text{бл}}$ – бөлме көлемі, м³.

Ауысуға керекті ауа көлемін $V_{\text{вент}}$ жылулық тепе-теңдік балансы арқылы анықтаймыз.

$$V_{\text{вент}} = \frac{3600 \cdot Q_{\text{аж}}}{C \cdot (t_{\text{кт}} - t_{\text{кл}}) \cdot Y}$$

мұнда $Q_{\text{аж}}$ – артық жылу, Вт;

$C = 1000$ – ауаның сыбағалы жылуөткізгіштілігі, Дж/кг·К;

$Y = 1.2$ – ауаның тығыздығы, кг/м³.

Тыс кететін ауаның температурасы мына формула бойынша есептелінеді

$$t_{\text{кт}} = t_{\text{км}} + (H - 2) \cdot t$$

мұнда t – бөлменің 1 метр биікке температуның өзгеруі, 1 ден 5 градусқа дейін мәндерін алады.

$t_{\text{км}}$ – жұмыс орындағы температурасы, 23 градусқа тең.

H – бөлменің биіктігі, 4 метрге тең.

$t_{\text{кл}}$ – келетін ауаның температурасы, 18 шрадусқа тең.

5.12-формула бойынша кететін ауаның температурасы мынағын тең:

$$T_{\text{кт}} = 23 + (3 - 2) \cdot 2 = 25.$$

Артық жылуды мына формула бойынша есептейміз

$$Q_{\text{аж}} = Q_{\text{аж1}} + Q_{\text{аж2}} + Q_{\text{аж3}}$$

мұнда $Q_{\text{аж1}}$ – электоржабдықтардан және жарықтандырудан артық жылу, Вт;

$Q_{\text{аж2}}$ – күннің сәулесінен келген жылу, Вт;

$Q_{\text{аж3}}$ – адмамның жылу беруі, Вт.

Электоржабдықтардан және жарықтандырудан артық жылуың мына формула бойынша есептейміз

$$Q_{\text{аж1}} = E \cdot P$$

мұнда E – жылу бұру энергиясының шығын коэффициенті;

P – электр жабдықтар қуаты, Вт.

Элетрэнергияның шығыны коэффициенті келесі мәндерді қабылдайды:

$E=0.55$ жарықтандыру үшін;

$E=0.4$ дербес компьтер үшін.

Келесі формула бойынша жарықтандырудың және электр жабдықтардың жалпы қуатын есептейміз

$$P = P_{\text{дана}} \cdot K$$

мұнда $P_{\text{дана}}$ – бір дана жабдықтың қуат өлшемі, Вт;

K – жабдықтардың сан данасы.
Сөйтіп, 5.16- формула бойынша:

$$P_{\text{жарықтандыру}} = 150 \cdot 3 = 450 \text{ Вт},$$

$$P_{\text{компьютеры және т.б.}} = 600 \cdot 6 + 700 \cdot 3 + 170 + 10 \cdot 3 = 5900 \text{ Вт}.$$

5.15- формула бойынша электоржабдықтардан және жарықтандырудан артық жылуын есептейміз:

$$Q_{\text{аж1}} = 0.55 \cdot 450 + 0,4 \cdot 5900 = 2607.5 \text{ Вт}.$$

Жасанды жарықтандырудан артық жылуын келесі формула бойынша есептейміз

$$Q_{\text{аж3}} = m \cdot S \cdot k \cdot Q_{\text{T}}$$

мұнда m – терезе саны;
 S – бір терезенің ауданы, м^2 ;
 k - әйнектеуді ескеретін коэффицент. Екі қабатты әйнектеудің коэффиценті – $k = 0.8$;
 Q_{T} – терезеден жылу келіуіб, 127 Вт/м тең.

5.17-формула бойынша жасанды жарықтандырудан артық жылуын есептеймік:

$$Q_{\text{аж2}} = 1 \cdot 2 \cdot 0.8 \cdot 127 = 203 \text{ Вт}.$$

Адамның жылу беруің келесі формула бойынша есептейміз

$$Q_{\text{аж3}} = n \cdot q$$

мұнда q – бір адамның жылуы бөлгіштігі, 80 Вт/ад тең;
 n – адам саны, 1 тең.

5.18-формула бойынша адамның жылу беруің есептейміз:

$$Q_{\text{аж3}} = 1 \cdot 80 = 80 \text{ Вт}.$$

5.12-формула бойынша артық жылуды есептейміз:

$$Q_{\text{аж}} = 2607.5 + 203 + 80 = 2890.5 \text{ Вт}.$$

Сөйтіп, 5.12-формула бойынша ауысуға керекті ауа көлемін есптейміз:

$$V_{\text{вент}} = \frac{3600 \cdot 2890,5}{1000 \cdot (25 - 18) \cdot 1,2} = 1238,78.$$

Демек, оптималды шартты тудыру үшін ауаны кондиционерлеуін жаса жүзеге асыру керек, ол деген бөлменің сыртында ауаның өзгеруіне және бөлменің ішінде өзгеруіне қармастан белгілі бір нұсқаудың шарты бойынша (берілген температура, ылғалдылық, ауаның ауысуы) автоматты түрде ауаны сақтандыруды айтады.

Қорытынды

Менің дипломдық жобамда «Banqet.kz» атты, жоғары потенциалды, сапалы, қала мейрамханалары мен адамдар арасын едәуір жақындататын қызмет көрсету веб-қосымшасы құрылды. Бұл қызмет мейрамханаларға қойылатын талаптар мен өзара бәсекелестікті арттырады және үлкен жарнама орны, сонымен қатар тұтынушылардың талап-талғамын дұрыс қанағаттандырады.

«Banqet.kz» сайты өз қызметін қазіргі кезде тек Алматы қаласындағы мейрамханаларға ғана ұсынады. Келешекте еліміздің барлық мейрамханаларына осы қызметті енгіземіз. Сонымен бірге, тек мейрамхана ғана емес өзге де көңіл көтеретін ойын-сауық отауларын да өз базамызға қосамыз.

Қаламыздағы мейрамхана қонақтарының онлайн-брондау қызметін қолдануы өз уақытын, қаржысы мен күшін сарп етпеуіне мүмкіндік береді.

Қолданушының тұрған жеріне байланысты оған ең жақын мейрамханалар тізімін шығарып беретін геолокация жүйесі біздің қызметтің басқа да ұқсас қызметтерден бірден-бір үлкен артықшылығы болып саналады.

Пайдаланылған әдебиеттер

1. ГОСТ 2.105–95. Общие требования к текстовым документам. – Взамен ГОСТ 2.105–79, ГОСТ 2.906–71. – М.: Изд-во стандартов, 1996. – 23 с.
2. ГОСТ 19.XXX-78. Единая система программной документации.
3. ГОСТ 19.701–90. Схемы алгоритмов, программ, данных и систем. – Взамен ГОСТ 19.002–80, ГОСТ 19.003–80. – М.: Изд-во стандартов, 1991. – 26 с.
4. ГОСТ 34.602–89. Техническое задание на создание автоматизированной системы: Утвержден и введен 24.03.89. – М.: Изд-во стандартов, 1989. – 20 с.
5. ГОСТ 34.201–89. Виды, комплектность и обозначение документов при создании автоматизированных систем: Утвержден и введен 24.03.89. – М.: Изд-во стандартов, 1989. – 14 с.
6. ГОСТ 34.601–90. Автоматизированные системы стадии создания. ОКСТУ 0034; Введен 01.01.92. – М.: Издательство стандартов, 1991. – 9 с.
7. РД 50–34.698–90. Методические указания. Информационная технология. Комплекс стандартов и руководящих документов на автоматизированные системы. Автоматизированные системы. Требования к содержанию документов: Введен 01.01.1992, Изд-во стандартов, 1996. – 14 с.
8. Ерекешова З.Д., Боканова Г.Ш., 5B070400 – Есептеу техникасы және бағдарламамен қамтамасыз ету мамандығының студенттері үшін диплом жұмысының экономика бөлімін орындауға әдістемелік нұсқаулар. – Алматы: АЭЖБУ, 2014-276.
9. Современная экономика: Учебное пособие /Под ред. О. Ю. Мамедова. – Ростов-на-Дону: Феникс, 2011. – 456 с.
10. В.М. Пономарева, В.И. Жукова, Безопасность жизнедеятельности. - 2014. -608 с.
11. Виноградова С.Н. Организация и технология торговли. Мн.: Высшая школа, 2007. 460 с.
12. Дашков Л.П., Памбухчиянц В.К. Организация, технология и проектирование торговых предприятий: Учебник. М.: ИВЦ "Маркетинг", 2008. 336 с.
13. Уодтке Кристина Информационная архитектура: чертежи для сайта. – М.: Кудиц-образ, 2004. – 320 с.
14. Мир CMS: мир управления сайтами. [Электронный ресурс] URL: <http://www.cmsworld.ru/> (дата обращения: 10.03.2010).
15. Кочеткова, О.В. ВСЕГДА ГОТОВ! Методические рекомендации по выполнению дипломной работы, для студентов специальности «Прикладная информатика (в экономике)» / О.В. Кочеткова, Э.Е. Кудряшова. – Волгоград.: МУПК, 2003 г. – 91 с.
16. Эффективная работа с СУБД А. Рубен, А. Горев, С. Макшарипов СПб.: Питер, 2009. – 822 с
17. Киселев Ю.Н. Электронная коммерция: практическое руководство.-

СПб: ООО «ДиаСофтЮП», 2006.

18. Завалеев В. Курс ЦИТ «Internet-технологии в проектах с пластиковыми карточками», Центр Информационных Технологий, 2007.

19. Быков В.А. Электронный бизнес и безопасность /.-М.: Радио и связь, 2000.-200 с.:ил.

20. www.wikipedia.kz сайты.

Қосымша А

```
<?php
```

```
namespace App\Http\Controllers;
```

```
use Illuminate\Foundation\Bus\DispatchesJobs;  
use Illuminate\Routing\Controller as BaseController;  
use Illuminate\Foundation\Validation\ValidatesRequests;  
use Illuminate\Foundation\Auth\Access\AuthorizesRequests;  
use Illuminate\Foundation\Auth\Access\AuthorizesResources;
```

```
class Controller extends BaseController
```

```
{  
    use AuthorizesRequests, AuthorizesResources, DispatchesJobs,  
    ValidatesRequests;  
}
```

```
<?php
```

```
namespace App\Http\Controllers;
```

```
use App\Http\Requests;  
use Illuminate\Http\Request;
```

```
class HomeController extends Controller
```

```
{  
    /**  
     * Create a new controller instance.  
     *  
     * @return void  
     */  
    public function __construct()  
    {  
        $this->middleware('auth');  
    }  
  
    /**  
     * Show the application dashboard.  
     *  
     * @return \Illuminate\Http\Response  
     */  
    public function index()  
    {  
        return view('home');
```

```
    }  
  }  
<?php
```

```
namespace App\Http\Controllers;
```

```
use DB;  
use Illuminate\Http\Request;  
use App\Restaurant;  
use App\RestaurantPlace;  
use App\Http\Requests;
```

```
class MainController extends Controller  
{  
    public function index()  
    {  
        $restaurants = Restaurant::with('places')->paginate(3);  
        return view('pages.main', ['restaurants' => $restaurants]);  
    }  
}  
<?php
```

```
namespace App\Http\Controllers\Auth;
```

```
use App\User;  
use Validator;  
use App\Http\Controllers\Controller;  
use Illuminate\Foundation\Auth\ThrottlesLogins;  
use Illuminate\Foundation\Auth\AuthenticatesAndRegistersUsers;
```

```
class AuthController extends Controller  
{  
    /*  
    |-----  
    | Registration & Login Controller  
    |-----  
    |  
    | This controller handles the registration of new users, as well as the  
    | authentication of existing users. By default, this controller uses  
    | a simple trait to add these behaviors. Why don't you explore it?  
    |  
    */  
  
    use AuthenticatesAndRegistersUsers, ThrottlesLogins;
```

```

/**
 * Where to redirect users after login / registration.
 *
 * @var string
 */
protected $redirectTo = '/';

/**
 * Create a new authentication controller instance.
 *
 * @return void
 */
public function __construct()
{
    $this->middleware($this->guestMiddleware(), ['except' => 'logout']);
}

/**
 * Get a validator for an incoming registration request.
 *
 * @param array $data
 * @return \Illuminate\Contracts\Validation\Validator
 */
protected function validator(array $data)
{
    return Validator::make($data, [
        'name' => 'required|max:255',
        'email' => 'required|email|max:255|unique:users',
        'password' => 'required|min:6|confirmed',
    ]);
}

/**
 * Create a new user instance after a valid registration.
 *
 * @param array $data
 * @return User
 */
protected function create(array $data)
{
    return User::create([
        'name' => $data['name'],
        'email' => $data['email'],
    ]);
}

```

```

        'password' => bcrypt($data['password']),
    ];
}
}
<?php

```

```

namespace App\Http\Controllers\Auth;

```

```

use App\Http\Controllers\Controller;
use Illuminate\Foundation\Auth\ResetsPasswords;

```

```

class PasswordController extends Controller

```

```

{
    /*
    |-----
    | Password Reset Controller
    |-----
    |
    | This controller is responsible for handling password reset requests
    | and uses a simple trait to include this behavior. You're free to
    | explore this trait and override any methods you wish to tweak.
    |
    */

```

```

    use ResetsPasswords;

```

```

    /**
     * Create a new password controller instance.
     *
     * @return void
     */

```

```

    public function __construct()
    {
        $this->middleware('guest');
    }

```

```

}
<?php

```

```

/*
|-----
| Application Routes
|-----
|
| Here is where you can register all of the routes for an application.

```

| It's a breeze. Simply tell Laravel the URIs it should respond to
| and give it the controller to call when that URI is requested.

```
|  
*/
```

```
Route::get('/personal', 'PersonalController@index');
```

```
Route::get('/', 'MainController@index');
```

```
Route::auth();
```

```
Route::get('/home', 'HomeController@index');
```

```
<?php
```

```
namespace App;
```

```
use Illuminate\Database\Eloquent\Model;
```

```
class Restaurant extends Model
```

```
{
```

```
    /**
```

```
     * Get the places for the restaurant.
```

```
    */
```

```
    public function places()
```

```
    {
```

```
        return $this->hasMany('App\RestaurantPlace');
```

```
    }
```

```
}
```

```
<?php
```

```
namespace App;
```

```
use Illuminate\Database\Eloquent\Model;
```

```
class RestaurantPlace extends Model
```

```
{
```

```
    /**
```

```
     * Get the restaurant that owns the place.
```

```
    */
```

```
    public function restaurant()
```

```
    {
```

```
        return $this->belongsTo('App\Restaurant');
```

```
    }
```

```
}
```

```
<?php
```

```

namespace App;

use Illuminate\Foundation\Auth\User as Authenticatable;

class User extends Authenticatable
{
    /**
     * The attributes that are mass assignable.
     *
     * @var array
     */
    protected $fillable = [
        'name', 'email', 'password',
    ];

    /**
     * The attributes that should be hidden for arrays.
     *
     * @var array
     */
    protected $hidden = [
        'password', 'remember_token',
    ];
}
<?php

```

```

use Illuminate\Database\Schema\Blueprint;
use Illuminate\Database\Migrations\Migration;

class CreateUsersTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('users', function (Blueprint $table) {
            $table->increments('id');
            $table->string('name');
            $table->string('email')->unique();
            $table->string('password');
        });
    }
}

```

```

        $table->rememberToken();
        $table->timestamps();
    });
}

/**
 * Reverse the migrations.
 *
 * @return void
 */
public function down()
{
    Schema::drop('users');
}
}
<?php

```

```

use Illuminate\Database\Schema\Blueprint;
use Illuminate\Database\Migrations\Migration;

```

```

class CreatePasswordResetsTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('password_resets', function (Blueprint $table) {
            $table->string('email')->index();
            $table->string('token')->index();
            $table->timestamp('created_at');
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::drop('password_resets');
    }
}

```

```

    }
}
<?php

use Illuminate\Database\Schema\Blueprint;
use Illuminate\Database\Migrations\Migration;

class CreateRestaurantsTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('restaurants', function (Blueprint $table) {
            $table->increments('id');
            $table->string('title');
            $table->string('status');
            $table->text('description');
            $table->string('seat');
            $table->string('avg_price');
            $table->string('kitchen');
            $table->string('allowed_self');
            $table->string('parking');
            $table->string('features');
            $table->string('extra');
            $table->string('site');
            $table->string('working_hours');
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::drop('restaurants');
    }
}

```



```
<?php
```

```
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Database\Migrations\Migration;
```

```
class CreateRestaurantPlacesTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('restaurant_places', function (Blueprint $table) {
            $table->increments('id');
            $table->integer('restaurant_id');
            $table->string('coordinates');
            $table->string('address');
            $table->string('phones');
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::drop('restaurant_places');
    }
}
<?php
```

```
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Database\Migrations\Migration;
```

```
class Areas extends Migration
{
    /**
     * Run the migrations.
     *

```

```

* @return void
*/
public function up()
{
    Schema::create('areas', function (Blueprint $table) {
        $table->increments('id');
        $table->string('alatau');
        $table->string('auezov');
        $table->string('zhetisu');
        $table->string('nauryzbai');
        $table->string('almaly');
        $table->string('bostandyk');
        $table->string('medeu');
        $table->string('turksib');
        $table->timestamps();
    });
}
/**
 * Reverse the migrations.
 *
 * @return void
 */
public function down()
{
    //
}
}
<?php

```

```

use Illuminate\Database\Seeder;

```

```

class DatabaseSeeder extends Seeder
{
    /**
     * Run the database seeds.
     *
     * @return void
     */
    public function run()
    {
        $this->call(UsersTableSeeder::class);
    }
}
<?php

```

```
use Illuminate\Database\Seeder;
```

```
class RestaurantsTableSeeder extends Seeder
```

```
{  
    /**  
     * Run the database seeds.  
     *  
     * @return void  
     */  
    public function run()  
    {  
        factory(App\Restaurant::class, 50)->create()->each(function($restaurant) {  
            $restaurant->places()->save(factory(App\RestaurantPlace::class)->make());  
        });  
    }  
}  
*{  
    margin: 0 0;  
    padding: 0 0;  
}  
@font-face{  
    font-family: 'Open Sans';  
    src: url(../fonts/open-sans-r.ttf);  
}  
@font-face{  
    font-family: 'Open Sans Bold';  
    src: url(../fonts/open-sans-b.ttf);  
}  
@font-face{  
    font-family: 'Open Sans Light';  
    src: url(../fonts/open-sans-l.ttf);  
}  
body{  
    min-width: 1024px;  
    font-family: 'Open Sans';  
    font-weight: 300;  
    color: #141414;  
    background: url(../img/bg.jpg);  
}  
a{  
    color: inherit;  
    text-decoration: none;  
    cursor: pointer;
```

```

}
p{
  cursor: default;
}
input{
  outline: none;
}

ul, li{list-style: none}

.wrapper{
  width: 1024px;
  margin: 0 auto;
  padding: 0 0;
}
.top-banner>.wrapper{
  box-shadow: none;
}

.main-body{
  background: url(../img/bg_main.jpg) no-repeat;
  background-size: cover;
}
.main-head{
  position: relative;
  width: 100%;
  float: left;
  background: none;
}
.search{
  float: left;
  margin-top: 33px;
  margin-left: 5px;
  position: relative;
}
.search input[type=text]{
  font-family: 'Open Sans', sans-serif;
  height: 23px;
  width: 225px;
  border: 1px solid #c4c4c4;
  border-radius: 23px;
  padding-left: 15px;
  outline: none;
  color: #c4c4c4;
}

```

```

    -webkit-box-shadow: inset 2px 2px 2px 0px #c4c4c4;
    -moz-box-shadow: inset 2px 2px 2px 0px #c4c4c4;
    box-shadow: inset 2px 2px 2px 0px #c4c4c4;
}
.search input[type=image]{
    position: absolute;
    top: 4px;
    right: 10px;
    outline: none;
}
.search input:focus{
    color: #000;
}
.enter{
    float: right;
    margin-top: 30px;
    font-family: 'Open Sans', sans-serif;
    color: #fff;
    text-align: right;
    margin-right: 5px;
}
.enter a{
    font-weight: 400;
    font-size: 22px;
    text-decoration: underline;
}
.enter span{
    font-size: 13px;
}

.main-sec{
    float: inherit;
    position: relative;
    top: 82px;
    text-align: center;
}
.main-sec>.wrapper{
    background: none;
    box-shadow: none;
}
.main-logo{
    display: inline-block;
    width: 565px;
    height: 324px;
}

```

```

    background: url(../img/logo_main.png);
    margin-top: 165px;
}

.main-foot{
    position: fixed;
    bottom: 0;
    width: 100%;
    text-align: center;
}

.scroll{
    display: inline-block;
    width: 120px;
    height: 86px;
    background: url(../img/scroll.png);
}

.top-banner{
    height: 170px;
    position: relative;
    background: #fff;
}

.navbar-static-top{
    height: 100px;
    width: 100%;
    background-color: #262626;
}

.logo{
    display: inline-block;
    float: left;
    margin-top: 20px;
    margin-left: 10px;
}

.reg-search{
    float: right;
    margin-right: 10px;
}

.reg-search .enter{
    float: none;
    margin-top: 10px;
    line-height: 20px;
    color: #fff;
}

.reg-search .enter a{

```

```

    font-size: 20px;
}
.reg-search .enter span{
    font-size: 12px;
}
.reg-search .search{
    margin-top: 10px;
}
.reg-search .search input[type="text"]{
    width: 220px;
    height: 22px;
}

#search{
    height: 325px;
}
#search>.wrapper{

}
section>.wrapper{
    background: #fff;
    height: 100%;
    -webkit-box-shadow: 5px 3px 40px 0px #969696;
    -moz-box-shadow: 5px 3px 40px 0px #969696;
    box-shadow: 5px 3px 40px 0px #969696;
}
form{
    color: #363636;
    display: inline-block;
}
form .raion{
    width: 605px;
    font-size: 17px;
    display: inline-block;
    float: left;
    margin: 10px 0 0 30px;
}
form b{
    float: left;
    margin-top: 10px;
    text-decoration: underline;
}
form ul{
    display: inline-block;

```

```

        width: 450px;
        float: left;
    }
    form li{
        float: left;
        margin-left: 30px;
        width: 180px;
    }
    input[type="checkbox"],
    input[type="radio"] {
        display:none;
    }
    input[type="checkbox"] + label,
    input[type="radio"] + label {
        font-family: "Open Sans", sans-serif;
        font-weight: 400;
        color: #444;
        cursor: pointer;
    }
    input[type="checkbox"] + label::before,
    input[type="radio"] + label::before {
        content: "";
        display: inline-block;
        height: 24px;
        width: 26px;
        margin: 10px 5px -3px 0;
        background-image: url(../img/checkbox.png);
        background-repeat: no-repeat;
    }
    input[type="checkbox"] + label::before {
        background-position: -29px;
    }
    input[type="checkbox"]:checked + label::before {
        background-position: 0px;
    }
    }

.svoji{
    display: inline-block;
    margin-left: 30px;
    margin-top: 20px;
}
.svoji label:nth-child(3){
    margin-left: 37px;
}

```



```

.svoji label{
  margin-left: 60px;
}

.text-search{
  display: inline-block;
  margin-left: 30px;
  margin-top: 50px;
  position: relative;
}
.text-search input[type="image"]{
  position: absolute;
  top: -7px;
  right: -200px;
}
.text-search input[type="text"]{
  font-family: "Open Sans", sans-serif;
  font-size: 18px;
  font-weight: 300;
  padding-left: 15px;
  width: 745px;
  height: 36px;
  border-radius: 10px;
  -webkit-box-shadow: inset 3px 3px 6px 0px #c4c4c4;
  -moz-box-shadow: inset 3px 3px 6px 0px #c4c4c4;
  box-shadow: inset 3px 3px 6px 0px #c4c4c4;
  outline: none;
  box-sizing: border-box;
}

.formCost {
  float: left;
  margin-bottom: 10px;
  display: inline-block;
  width: 375px;
}
.content{
  margin-top: 30px;
}
.content h1{
  display: inline-block;
  margin: 30px;
  font-weight: 400;
}

```

```

.result{
    display: inline-block;
    margin-left: 22px;
    border-top: 2px solid #ddddd;
    width: 980px;
}
.result-img{
    float: left;
    margin: 30px 30px 30px 15px;
}
.result-info{
    float: right;
    padding: 20px 0 20px 0;
}
.result-info li h2{
    font-weight: 400;
    font-size: 27px;
    text-decoration: underline;
}
.result-info li{
    font-size: 15px;
    font-weight: 400;
    line-height: 25px;
}
.result-info a{
    float: right;
    display: inline-block;
    width: 172px;
    height: 35px;
    color: #fff;
    text-align: center;
    font-weight: 400;
    line-height: 33px;
    background: url(../img/more.png);
    margin-top: 15px;
}

.pagination{
    margin: 30px 0 25px 30px;
    text-align: center;
}
.pagination li{
    display: inline-block;
    font-size: 19px;
}

```

```

    font-weight: 400;
    line-height: 18px;
    margin-left: 5px;
}
.pagination li:last-child{
    font-weight: 300;
    font-size: 15px;
    margin-left: 15px;
}
.pagination .active{
    text-decoration: underline;
    font-size: 30px;
    color: #dd2626;
}
.add{
    display: block;
    width: 221px;
    height: 47px;
    background: url(../img/add.png);
    color: #fff;
    font-weight: 400;
    font-size: 16px;
    text-align: center;
    line-height: 45px;
    margin: 0 0 30px 370px;
}

.back-to-results{
    display: block;
    width: 236px;
    height: 23px;
    background: #ff8181;
    color: #fff;
    padding-left: 7px;
}
.back-to-results:hover{
    background: #dd2626;
}
.content-inner{
    margin-top: 0;
}
.personal-title{
    margin: 50px 0 5px 30px;
    display: inline-block;
}

```

```

}
.personal-title p{
    font-weight: 400;
    font-size: 23px;
}
.personal-title h3{
    font-weight: 400;
    font-size: 40px;
}

.menu-photo{
    float: right;
    margin-top: 60px;
    margin-right: 30px;
}
.personal-photos, .personal-menu{
    display: block;
    width: 49px;
    height: 60px;
    background: url(../img/menu-photo.png);
    float: left;
}
.personal-photos:hover, .personal-menu:hover{
    background: url(../img/menu-photo-hover.png);
    width: 53px;
    height: 66px;
}
.personal-menu:hover{
    background-position: -53px;
}
.personal-menu{
    background-position: -48px;
    margin-left: 65px;
}

.personal-left{
    display: inline-block;
    margin: 0 0 0 30px;
    width: 716px;
}
.personal-left h4{
    font-weight: 600;
    font-size: 22px;
}

```

```

.personal-left p{
    font-weight: 400;
    font-size: 15px;
    text-align: justify;
}
.content-inner .fotorama__nav__frame.fotorama__active .fotorama__dot{
    background: #9d1f1f;
}
.content-inner .fotorama__dot{
    width: 10px !important;
    height: 10px !important;
    border: 1px solid #9d1f1f !important;
    border-radius: 50% !important;
}

.personal-right{
    display: inline-block;
    float: right;
    margin-right: 70px;
}
.personal-right ul, .personal-address ul{
    margin-bottom: 10px;
}
.personal-right ul:last-child .personal-site a{
    color: #9d1f1f;
}
.personal-contacts{
    margin: 60px 30px 30px 30px;
    width: 963px;
    height: 260px;
}
.personal-address{
    display: inline-block;
    padding-top: 22px;
    background: #262626;
    width: 403px;
    height: 238px;
}
.personal-address ul{
    text-align: center;
    color: #fff;
}
.personal-address li{
    font-weight: 400 !important;
}

```

```

}
.personal-address .personal-options{
    font-size: 15px !important;
}
.personal-right li .personal-address li{
    font-weight: 400;
    font-size: 15px;
    margin-left: 12px;
}
.personal-right .personal-options, .personal-address .personal-options{
    text-decoration: underline;
    font-weight: 700;
    font-size: 13px;
    margin-left: 0;
}

section.content-add-institution{
    margin-top: 0;
    background: #fff;
}
section.content-add-institution .wrapper{
    box-shadow: none;
    font-weight: 400;
}
.add-institution-form{
    position: relative;
    margin-left: 30px;
}
.add-institution-form input[type='file']{
    display: none;
}
.add-institution-form .add-logo{
    display: inline-block;
    width: 250px;
    height: 250px;
    background: url(../img/label-bg.png) center no-repeat;
    overflow: hidden;
    position: relative;
    cursor: pointer;
    margin-bottom: 11px;
    float: left;
    margin-right: 11px;
    text-align: center;
    transition: 0.3s ease-in-out;
}

```

```

}
.add-logo>.add-delete{
  width: 100%;
  height: 30px;
  position: absolute;
  bottom: -30px;
  background: #474747;
  -webkit-transition: bottom 0.3s ease-in-out 0.3s;
  -moz-transition: bottom 0.3s ease-in-out 0.3s;
  -o-transition: bottom 0.3s ease-in-out 0.3s;
  transition: bottom 0.3s ease-in-out 0.3s;
}
.add-logo:hover .add-delete{
  bottom: 0;
}
.add-delete a{
  color: #ededed;
  margin: 4px 10px 4px 10px;
  font-size: 13px;
}
.add-delete #add_logo{
  float: left;
}
.add-delete #delete_logo{
  float: right;
}
input[type='text']{
  width: 620px;
  height: 29px;
  background: #f0f0f0;
  border-bottom: 2px solid #cacaca;
  border-top: none;
  border-left: none;
  border-right: none;
  box-sizing: border-box;
  padding-left: 15px;
  font-size: 16px;
  color: #2d2d2d;
  font-family: 'Open Sans';
}
.add-institution-form form>div{
  margin-top: 30px;
}
.add-institution-form select{

```

```

border: none;
outline: none;
text-decoration: underline;
color: #db3a3a;
font-size: 16px;
display: inline-block;
width: 150px;
}
.add-institution-form option{
  color: #474747;
}
.add-institution-form option:hover{
  background: #db3a3a;
}
.add-institution-form .sity-select{
  position: relative;
}
.add-institution-form .sity-select:after{
  content:' ';
  font:11px "Consolas", monospace;
  color:#aaa;
  -webkit-transform:rotate(90deg);
  -moz-transform:rotate(90deg);
  -ms-transform:rotate(90deg);
  transform:rotate(90deg);
  right:8px; top:2px;
  padding:0 0 2px;
  border-bottom:1px solid #ddd;
  position:absolute;
  pointer-events:none;
}
.add-institution-form .sity-select:before {
  content:"";
  right:6px; top:0px;
  width:20px; height:20px;
  background:#fff;
  position:absolute;
  pointer-events:none;
  display:block;
}
.logo-upload{
  display: inline-block;
  position: absolute;
  top: 0px;

```



```

    right: 30px;

}
.phones-area>span{
    display: block;
    background: none;
    margin-top: -5px;
}
.phones-area>span:nth-child(3){
    margin-top: 15px;
}
.phones-area input{
    padding: 0;
    text-align: center;
}
.phones-area .phone{
    width: 60px;
    margin: 0 15px 0 15px;
}
.phones-area .phone1{
    width: 100px;
}

footer{
    height: 84px;
    margin-bottom: 65px;
}
footer.content-add-institution{
    background: #262626;
}
footer.content-add-institution .wrapper{
    box-shadow: none;
    background: none;
}
footer .wrapper{
    background: #262626;
    padding: 0 30px 0 30px;
    box-sizing: border-box;
    -webkit-box-shadow: 5px 3px 40px 0px #969696;
    -moz-box-shadow: 5px 3px 40px 0px #969696;
    box-shadow: 5px 3px 40px 0px #969696;
}
.foot-logo{
    display: inline-block;

```

```

    width: 64px;
    height: 61px;
    background: url(../img/foot-logo.png);
    margin: 10px 0 10px 0;
}
.copyright{
    display: inline-block;
    position: relative;
    top: -10px;
    left: 20px;
    color: #fff;
    font-size: 13px;
    margin-top: 10px;
}
.copyright span{
    display: block;
    font-weight: 400;
    margin-bottom: 7px;
}
.copyright a{
    text-decoration: underline;
}
.foot-menu{
    display: inline-block;
    float: right;
    margin-top: 30px;
}
.foot-menu li{
    float: left;
    color: #fff;
    margin-left: 60px;
}
.foot-menu li:hover{
    text-decoration: underline;
}
/* Ion.RangeSlider, Simple Skin
// css version 2.0.3
// © Denis Ineshin, 2014 https://github.com/IonDen
// © guybowden, 2014 https://github.com/guybowden
//
=====
=====*/

```

```

/*
=====
=====
// Skin details */

.irs {
    height: 55px;
}
.irs-with-grid {
    height: 75px;
}
.irs-line {
    height: 10px; top: 33px;
    background: #EEE;
    background: linear-gradient(to bottom, #DDD -50%, #FFF 150%); /* W3C */
    border: 1px solid #CCC;
    border-radius: 16px;
    -moz-border-radius: 16px;
}
.irs-line-left {
    height: 8px;
}
.irs-line-mid {
    height: 8px;
}
.irs-line-right {
    height: 8px;
}

.irs-bar {
    height: 10px; top: 33px;
    border-top: 1px solid #428bca;
    border-bottom: 1px solid #428bca;
    background: #428bca;
    background: linear-gradient(to top, rgba(66,139,202,1) 0%, rgba(127,195,232,1)
100%); /* W3C */
}
.irs-bar-edge {
    height: 10px; top: 33px;
    width: 14px;
    border: 1px solid #428bca;
    border-right: 0;
    background: #428bca;
}

```

```

        background: linear-gradient(to top, rgba(66,139,202,1)
0%,rgba(127,195,232,1) 100%); /* W3C */
        border-radius: 16px 0 0 16px;
        -moz-border-radius: 16px 0 0 16px;
    }

.irs-shadow {
    height: 2px; top: 38px;
    background: #000;
    opacity: 0.3;
    border-radius: 5px;
    -moz-border-radius: 5px;
}
.lt-ie9 .irs-shadow {
    filter: alpha(opacity=30);
}

.irs-slider {
    top: 25px;
    width: 27px; height: 27px;
    border: 1px solid #AAA;
    background: #DDD;
    background: linear-gradient(to bottom, rgba(255,255,255,1)
0%,rgba(220,220,220,1) 20%,rgba(255,255,255,1) 100%); /* W3C */
    border-radius: 27px;
    -moz-border-radius: 27px;
    box-shadow: 1px 1px 3px rgba(0,0,0,0.3);
    cursor: pointer;
}

.irs-slider.state_hover, .irs-slider:hover {
    background: #FFF;
}

.irs-min, .irs-max {
    color: #333;
    font-size: 12px; line-height: 1.333;
    text-shadow: none;
    top: 0;
    padding: 1px 5px;
    background: rgba(0,0,0,0.1);
    border-radius: 3px;
    -moz-border-radius: 3px;
}

```

```

.lt-ie9 .irs-min, .lt-ie9 .irs-max {
  background: #ccc;
}

.irs-from, .irs-to, .irs-single {
  color: #fff;
  font-size: 14px; line-height: 1.333;
  text-shadow: none;
  padding: 1px 5px;
  background: #428bca;
  border-radius: 3px;
  -moz-border-radius: 3px;
}
.lt-ie9 .irs-from, .lt-ie9 .irs-to, .lt-ie9 .irs-single {
  background: #999;
}

.irs-grid {
  height: 27px;
}
.irs-grid-pol {
  opacity: 0.5;
  background: #428bca;
}
.irs-grid-pol.small {
  background: #999;
}

.irs-grid-text {
  bottom: 5px;
  color: #99a4ac;
}

.irs-disabled {
}
/*!
 * Fotorama 4.6.4 | http://fotorama.io/license/
 */
.fotorama__arr:focus:after, .fotorama__fullscreen-icon:focus:after,
.fotorama__html, .fotorama__img, .fotorama__nav__frame:focus
.fotorama__dot:after, .fotorama__nav__frame:focus .fotorama__thumb:after,
.fotorama__stage__frame, .fotorama__stage__shaft, .fotorama__video iframe {
  position: absolute;
}

```

```

width: 100%;
height: 100%;
top: 0;
right: 0;
left: 0;
bottom: 0;
}
.fotorama--fullscreen, .fotorama__img {
max-width: 99999px!important;
max-height: 99999px!important;
min-width: 0!important;
min-height: 0!important;
border-radius: 0!important;
box-shadow: none!important;
padding: 0!important;
}
.fotorama__wrap .fotorama__grab {
cursor: move;
cursor: -webkit-grab;
cursor: -o-grab;
cursor: -ms-grab;
cursor: grab;
}
.fotorama__grabbing * {
cursor: move;
cursor: -webkit-grabbing;
cursor: -o-grabbing;
cursor: -ms-grabbing;
cursor: grabbing;
}
.fotorama__spinner {
position: absolute!important;
top: 50%!important;
left: 50%!important;
}
.fotorama__wrap--css3 .fotorama__arr, .fotorama__wrap--css3
.fotorama__fullscreen-icon, .fotorama__wrap--css3 .fotorama__nav__shaft,
.fotorama__wrap--css3 .fotorama__stage__shaft, .fotorama__wrap--css3
.fotorama__thumb-border, .fotorama__wrap--css3 .fotorama__video-close,
.fotorama__wrap--css3 .fotorama__video-play {
-webkit-transform: translate3d(0, 0, 0);
transform: translate3d(0, 0, 0);
}

```

```

.fotorama__caption, .fotorama__nav:after, .fotorama__nav:before,
.fotorama__stage:after, .fotorama__stage:before, .fotorama__wrap--css3
.fotorama__html, .fotorama__wrap--css3 .fotorama__nav, .fotorama__wrap--css3
.fotorama__spinner, .fotorama__wrap--css3 .fotorama__stage, .fotorama__wrap--
css3 .fotorama__stage .fotorama__img, .fotorama__wrap--css3
.fotorama__stage__frame {
  -webkit-transform: translateZ(0);
  transform: translateZ(0);
}
.fotorama__arr:focus, .fotorama__fullscreen-icon:focus, .fotorama__nav__frame {
  outline: 0;
}
.fotorama__arr:focus:after, .fotorama__fullscreen-icon:focus:after,
.fotorama__nav__frame:focus .fotorama__dot:after, .fotorama__nav__frame:focus
.fotorama__thumb:after {
  content: "";
  border-radius: inherit;
  background-color: rgba(0, 175, 234, .5);
}
.fotorama__wrap--video .fotorama__stage, .fotorama__wrap--video
.fotorama__stage__frame--video, .fotorama__wrap--video
.fotorama__stage__frame--video .fotorama__html, .fotorama__wrap--video
.fotorama__stage__frame--video .fotorama__img, .fotorama__wrap--video
.fotorama__stage__shaft {
  -webkit-transform: none!important;
  transform: none!important;
}
.fotorama__wrap--css3 .fotorama__nav__shaft, .fotorama__wrap--css3
.fotorama__stage__shaft, .fotorama__wrap--css3 .fotorama__thumb-border {
  transition-property: -webkit-transform, width;
  transition-property: transform, width;
  transition-timing-function: cubic-bezier(0.1, 0, .25, 1);
  transition-duration: 0ms;
}
.fotorama__arr, .fotorama__fullscreen-icon, .fotorama__no-select,
.fotorama__video-close, .fotorama__video-play, .fotorama__wrap {
  -webkit-user-select: none;
  -moz-user-select: none;
  -ms-user-select: none;
  user-select: none;
}
.fotorama__select {
  -webkit-user-select: text;
  -moz-user-select: text;
}

```

```

    -ms-user-select: text;
    user-select: text;
}
.fotorama__nav, .fotorama__nav__frame {
    margin: auto;
    padding: 0;
}
.fotorama__caption__wrap, .fotorama__nav__frame, .fotorama__nav__shaft {
    -moz-box-orient: vertical;
    display: inline-block;
    vertical-align: middle;
    *display: inline;
    *zoom: 1;
}
.fotorama__nav__frame, .fotorama__thumb-border {
    box-sizing: content-box;
}
.fotorama__caption__wrap {
    box-sizing: border-box;
}
.fotorama--hidden, .fotorama__load {
    position: absolute;
    left: -99999px;
    top: -99999px;
    z-index: -1;
}
.fotorama__arr, .fotorama__fullscreen-icon, .fotorama__nav,
.fotorama__nav__frame, .fotorama__nav__shaft, .fotorama__stage__frame,
.fotorama__stage__shaft, .fotorama__video-close, .fotorama__video-play {
    -webkit-tap-highlight-color: transparent;
}
.fotorama__arr, .fotorama__fullscreen-icon, .fotorama__video-close,
.fotorama__video-play {
    background: url(fotorama.png) no-repeat;
}
@media (-webkit-min-device-pixel-ratio:1.5), (min-resolution:2dppx) {
    .fotorama__arr, .fotorama__fullscreen-icon, .fotorama__video-close,
.fotorama__video-play {
        background: url(fotorama@2x.png) 0 0/96px 160px no-repeat;
    }
}
.fotorama__thumb {
    background-color: #7f7f7f;
    background-color: rgba(127, 127, 127, .2);
}

```



```

@media print {
  .fotorama__arr,      .fotorama__fullscreen-icon,      .fotorama__thumb-border,
.fotorama__video-close, .fotorama__video-play {
  background: none!important;
}
}.fotorama {
  min-width: 1px;
  overflow: hidden;
}
.fotorama:not(.fotorama--unobtrusive)>*:not(:first-child) {
  display: none;
}
.fullscreen {
  width: 100%!important;
  height: 100%!important;
  max-width: 100%!important;
  max-height: 100%!important;
  margin: 0!important;
  padding: 0!important;
  overflow: hidden!important;
  background: #000;
}
.fotorama--fullscreen {
  position: absolute!important;
  top: 0!important;
  left: 0!important;
  right: 0!important;
  bottom: 0!important;
  float: none!important;
  z-index: 2147483647!important;
  background: #000;
  width: 100%!important;
  height: 100%!important;
  margin: 0!important;
}
.fotorama--fullscreen .fotorama__nav, .fotorama--fullscreen .fotorama__stage {
  background: #000;
}
.fotorama__wrap {
  -webkit-text-size-adjust: 100%;
  position: relative;
  direction: ltr;
  z-index: 0;
}

```

```

.fotorama__wrap--rtl .fotorama__stage__frame {
  direction: rtl;
}
.fotorama__nav, .fotorama__stage {
  overflow: hidden;
  position: relative;
  max-width: 100% }
.fotorama__wrap--pan-y {
  -ms-touch-action: pan-y;
}
.fotorama__wrap .fotorama__pointer {
  cursor: pointer;
}
.fotorama__wrap--slide .fotorama__stage__frame {
  opacity: 1!important;
}
.fotorama__stage__frame {
  overflow: hidden;
}
.fotorama__stage__frame.fotorama__active {
  z-index: 8;
}
.fotorama__wrap--fade .fotorama__stage__frame {
  display: none;
}
.fotorama__wrap--fade .fotorama__fade-front, .fotorama__wrap--fade
.fotorama__fade-rear, .fotorama__wrap--fade
.fotorama__stage__frame.fotorama__active {
  display: block;
  left: 0;
  top: 0;
}
.fotorama__wrap--fade .fotorama__fade-front {
  z-index: 8;
}
.fotorama__wrap--fade .fotorama__fade-rear {
  z-index: 7;
}
.fotorama__wrap--fade .fotorama__fade-rear.fotorama__active {
  z-index: 9;
}
.fotorama__wrap--fade .fotorama__stage .fotorama__shadow {
  display: none;
}

```

```

.fotorama__img {
  -ms-filter: "alpha(Opacity=0)";
  filter: alpha(opacity=0);
  opacity: 0;
  border: none!important;
}
.fotorama__error .fotorama__img, .fotorama__loaded .fotorama__img {
  -ms-filter: "alpha(Opacity=100)";
  filter: alpha(opacity=100);
  opacity: 1;
}
.fotorama--fullscreen .fotorama__loaded--full .fotorama__img, .fotorama__img--full {
  display: none;
}
.fotorama--fullscreen .fotorama__loaded--full .fotorama__img--full {
  display: block;
}
.fotorama__wrap--only-active .fotorama__nav, .fotorama__wrap--only-active
.fotorama__stage {
  max-width: 99999px!important;
}
.fotorama__wrap--only-active .fotorama__stage__frame {
  visibility: hidden;
}
.fotorama__wrap--only-active .fotorama__stage__frame.fotorama__active {
  visibility: visible;
}
.fotorama__nav {
  font-size: 0;
  line-height: 0;
  text-align: center;
  display: none;
  white-space: nowrap;
  z-index: 5;
}
.fotorama__nav__shaft {
  position: relative;
  left: 0;
  top: 0;
  text-align: left;
}
.fotorama__nav__frame {
  position: relative;
}

```

```

    cursor: pointer;
}
.fotorama__nav--dots {
    display: block;
}
.fotorama__nav--dots .fotorama__nav__frame {
    width: 18px;
    height: 30px;
    margin-left: 10px;
}
.fotorama__nav--dots .fotorama__nav__frame--thumb, .fotorama__nav--dots
.fotorama__thumb-border {
    display: none;
}
.fotorama__nav--thumbs {
    display: block;
}
.fotorama__nav--thumbs .fotorama__nav__frame {
    padding-left: 0!important;
}
.fotorama__nav--thumbs .fotorama__nav__frame:last-child {
    padding-right: 0!important;
}
.fotorama__nav--thumbs .fotorama__nav__frame--dot {
    display: none;
}
.fotorama__dot {
    display: block;
    width: 17px;
    height: 16px;
    position: relative;
    top: 12px;
    left: 6px;
    background: url(../img/point2.png);
    background-position: -1px;
}
.fotorama__nav__frame:focus .fotorama__dot:after {
    padding: 1px;
    top: -1px;
    left: -1px;
}
.fotorama__nav__frame.fotorama__active .fotorama__dot {
    background: url(../img/point1.png);
    background-position: -1px -1px;
}

```

```

}
.fotorama__nav__frame.fotorama__active .fotorama__dot:after {
  padding: 3px;
  top: -3px;
  left: -3px;
}
.fotorama__thumb {
  overflow: hidden;
  position: relative;
  width: 100%;
  height: 100% }
.fotorama__nav__frame:focus .fotorama__thumb {
  z-index: 2;
}
.fotorama__thumb-border {
  position: absolute;
  z-index: 9;
  top: 0;
  left: 0;
  border-style: solid;
  border-color: #00afea;
  background-image: linear-gradient(to bottom right, rgba(255, 255, 255, .25),
rgba(64, 64, 64, .1));
}
.fotorama__caption {
  position: absolute;
  z-index: 12;
  bottom: 0;
  left: 0;
  right: 0;
  font-family: 'Helvetica Neue', Arial, sans-serif;
  font-size: 14px;
  line-height: 1.5;
  color: #000;
}
.fotorama__caption a {
  text-decoration: none;
  color: #000;
  border-bottom: 1px solid;
  border-color: rgba(0, 0, 0, .5);
}
.fotorama__caption a:hover {
  color: #333;
  border-color: rgba(51, 51, 51, .5);
}

```

```

}
.fotorama__wrap--rtl .fotorama__caption {
  left: auto;
  right: 0;
}
.fotorama__wrap--no-captions .fotorama__caption, .fotorama__wrap--video
.fotorama__caption {
  display: none;
}
.fotorama__caption__wrap {
  background-color: #fff;
  background-color: rgba(255, 255, 255, .9);
  padding: 5px 10px;
}
@-webkit-keyframes spinner {
  0% {
    -webkit-transform: rotate(0);
    transform: rotate(0);
  }
  100% {
    -webkit-transform: rotate(360deg);
    transform: rotate(360deg);
  }
}
@keyframes spinner {
  0% {
    -webkit-transform: rotate(0);
    transform: rotate(0);
  }
  100% {
    -webkit-transform: rotate(360deg);
    transform: rotate(360deg);
  }
}
.fotorama__wrap--css3 .fotorama__spinner {
  -webkit-animation: spinner 24s infinite linear;
  animation: spinner 24s infinite linear;
}
.fotorama__wrap--css3 .fotorama__html, .fotorama__wrap--css3 .fotorama__stage
.fotorama__img {
  transition-property: opacity;
  transition-timing-function: linear;
  transition-duration: .3s;
}
.fotorama__wrap--video .fotorama__stage__frame--video .fotorama__html,
.fotorama__wrap--video .fotorama__stage__frame--video .fotorama__img {

```

```

    -ms-filter: "alpha(Opacity=0)";
    filter: alpha(opacity=0);
    opacity: 1;
}
.fotorama__select {
    cursor: auto;
}
.fotorama__video {
    top: 32px;
    right: 0;
    bottom: 0;
    left: 0;
    position: absolute;
    z-index: 10;
}
@-moz-document url-prefix() {
    .fotorama__active {
        box-shadow: 0 0 0 transparent;
    }
}
.fotorama__arr, .fotorama__fullscreen-icon, .fotorama__video-close,
.fotorama__video-play {
    position: absolute;
    z-index: 11;
    cursor: pointer;
}
.fotorama__arr {
    position: absolute;
    width: 27px;
    height: 88px;
    top: 50%;
    margin-top: -44px;
}
.fotorama__arr--prev {
    left: 0px;
    background: url(..img/left.png);
}
.fotorama__arr--next {
    right: 0px;
    background: url(..img/right.png);
}
.fotorama__fullscreen-icon {
    width: 32px;
    height: 32px;
    top: 2px;
}

```

```

    right: 2px;
    background-position: 0 -32px;
    z-index: 20;
}
.fotorama__arr:focus, .fotorama__fullscreen-icon:focus {
    border-radius: 50% }
.fotorama--fullscreen .fotorama__fullscreen-icon {
    background-position: -32px -32px;
}
.fotorama__video-play {
    width: 96px;
    height: 96px;
    left: 50%;
    top: 50%;
    margin-left: -48px;
    margin-top: -48px;
    background-position: 0 -64px;
    opacity: 0;
}
.fotorama__wrap--css2 .fotorama__video-play, .fotorama__wrap--video
.fotorama__stage .fotorama__video-play {
    display: none;
}
.fotorama__error .fotorama__video-play, .fotorama__loaded .fotorama__video-
play, .fotorama__nav__frame .fotorama__video-play {
    opacity: 1;
    display: block;
}
.fotorama__nav__frame .fotorama__video-play {
    width: 32px;
    height: 32px;
    margin-left: -16px;
    margin-top: -16px;
    background-position: -64px -32px;
}
.fotorama__video-close {
    width: 32px;
    height: 32px;
    top: 0;
    right: 0;
    background-position: -64px 0;
    z-index: 20;
    opacity: 0;
}

```



```

.fotorama__wrap--css2 .fotorama__video-close {
  display: none;
}
.fotorama__wrap--css3 .fotorama__video-close {
  -webkit-transform: translate3d(32px, -32px, 0);
  transform: translate3d(32px, -32px, 0);
}
.fotorama__wrap--video .fotorama__video-close {
  display: block;
  opacity: 1;
}
.fotorama__wrap--css3.fotorama__wrap--video .fotorama__video-close {
  -webkit-transform: translate3d(0, 0, 0);
  transform: translate3d(0, 0, 0);
}
.fotorama__wrap--no-controls.fotorama__wrap--toggle-arrows .fotorama__arr,
.fotorama__wrap--no-controls.fotorama__wrap--toggle-arrows
.fotorama__fullscreen-icon {
  opacity: 1;
}
.fotorama__wrap--no-controls.fotorama__wrap--toggle-arrows
.fotorama__arr:focus, .fotorama__wrap--no-controls.fotorama__wrap--toggle-
arrows.fotorama__fullscreen-icon:focus {
  opacity: 1;
}
.fotorama__wrap--video .fotorama__arr, .fotorama__wrap--video
.fotorama__fullscreen-icon {
  opacity: 0!important;
}
.fotorama__wrap--css2.fotorama__wrap--no-controls.fotorama__wrap--toggle-
arrows .fotorama__arr, .fotorama__wrap--css2.fotorama__wrap--no-
controls.fotorama__wrap--toggle-arrows.fotorama__fullscreen-icon {
  display: none;
}
.fotorama__wrap--css2.fotorama__wrap--no-controls.fotorama__wrap--toggle-
arrows .fotorama__arr:focus, .fotorama__wrap--css2.fotorama__wrap--no-
controls.fotorama__wrap--toggle-arrows.fotorama__fullscreen-icon:focus {
  display: block;
}
.fotorama__wrap--css2.fotorama__wrap--video .fotorama__arr, .fotorama__wrap--
css2.fotorama__wrap--video.fotorama__fullscreen-icon {
  display: none!important;
}

```

```

.fotorama__wrap--css3.fotorama__wrap--no-controls.fotorama__wrap--
slide.fotorama__wrap--toggle-arrows .fotorama__fullscreen-icon:not(:focus) {
  -webkit-transform: translate3d(32px, -32px, 0);
  transform: translate3d(32px, -32px, 0);
}
.fotorama__wrap--css3.fotorama__wrap--video .fotorama__fullscreen-icon {
  -webkit-transform: translate3d(32px, -32px, 0)!important;
  transform: translate3d(32px, -32px, 0)!important;
}
.fotorama__wrap--css3.fotorama__wrap--video .fotorama__arr--prev {
  -webkit-transform: translate3d(-48px, 0, 0)!important;
  transform: translate3d(-48px, 0, 0)!important;
}
.fotorama__wrap--css3.fotorama__wrap--video .fotorama__arr--next {
  -webkit-transform: translate3d(48px, 0, 0)!important;
  transform: translate3d(48px, 0, 0)!important;
}
.fotorama__wrap--css3 .fotorama__arr:not(:focus), .fotorama__wrap--css3
.fotorama__fullscreen-icon:not(:focus), .fotorama__wrap--css3 .fotorama__video-
close:not(:focus), .fotorama__wrap--css3 .fotorama__video-play:not(:focus) {
  transition-property: -webkit-transform, opacity;
  transition-property: transform, opacity;
  transition-duration: .3s;
}
.fotorama__nav:after, .fotorama__nav:before, .fotorama__stage:after,
.fotorama__stage:before {
  content: "";
  display: block;
  position: absolute;
  text-decoration: none;
  top: 0;
  bottom: 0;
  width: 10px;
  height: auto;
  z-index: 10;
  pointer-events: none;
  background-repeat: no-repeat;
  background-size: 1px 100%, 5px 100% }
.fotorama__nav:before, .fotorama__stage:before {
  background-image: linear-gradient(transparent, rgba(0, 0, 0, .2) 25%, rgba(0, 0,
0, .3) 75%, transparent), radial-gradient(farthest-side at 0 50%, rgba(0, 0, 0, .4),
transparent);
  background-position: 0 0, 0 0;
  left: -10px;
}

```

```

}
.fotorama__nav.fotorama__shadows--left:before,
.fotorama__stage.fotorama__shadows--left:before {
    left: 0;
}
.fotorama__nav:after, .fotorama__stage:after {
    background-image: linear-gradient(transparent, rgba(0, 0, 0, .2) 25%, rgba(0, 0, 0, .3) 75%, transparent), radial-gradient(farthest-side at 100% 50%, rgba(0, 0, 0, .4), transparent);
    background-position: 100% 0, 100% 0;
    right: -10px;
}
.fotorama__nav.fotorama__shadows--right:after,
.fotorama__stage.fotorama__shadows--right:after {
    right: 0;
}
.fotorama--fullscreen .fotorama__nav:after, .fotorama--fullscreen
.fotorama__nav:before, .fotorama--fullscreen .fotorama__stage:after, .fotorama--
fullscreen .fotorama__stage:before, .fotorama__wrap--fade .fotorama__stage:after,
.fotorama__wrap--fade .fotorama__stage:before, .fotorama__wrap--no-shadows
.fotorama__nav:after, .fotorama__wrap--no-shadows .fotorama__nav:before,
.fotorama__wrap--no-shadows .fotorama__stage:after, .fotorama__wrap--no-
shadows .fotorama__stage:before {
    display: none;
}
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">

    <title>Laravel</title>

    <!-- Fonts -->

    <!-- Styles -->
    {{-- <link href="{{ elixir('css/app.css') }}" rel="stylesheet" --}}
    <link rel="stylesheet" type="text/css" href="css/style.css">
    <link rel="stylesheet" type="text/css" href="css/fotorama.css">
    <link rel="stylesheet" type="text/css" href="css/ion.rangeSlider.css">
    <link
        rel="stylesheet"
        type="text/css"
href="css/ion.rangeSlider.skinHTML5.css">
    <link rel="stylesheet" type="text/css" href="css/normalize.css">

```

```

</head>
<body id="app-layout">
  <nav class="navbar navbar-default navbar-static-top">
    <div class="wrapper">
      <div class="navbar-header">

        <!-- Collapsed Hamburger -->

        <!--<button type="button" class="navbar-toggle collapsed" data-
toggle="collapse" data-target="#app-navbar-collapse">
  <span class="sr-only">Toggle Navigation</span>
  <span class="icon-bar"></span>
  <span class="icon-bar"></span>
  <span class="icon-bar"></span>
</button> -->

        <!-- Branding Image -->
        <a class="navbar-brand logo" href="{{ url('/') }}">
          
        </a>
      </div>

      <div class="collapse navbar-collapse reg-search" id="app-navbar-collapse">
        <!-- Left Side Of Navbar -->
        <ul class="nav navbar-nav enter">
          @if (Auth::guest())
            <li><a href="{{ url('/login') }}">Login</a></li>
            <li><a href="{{ url('/register') }}">Register</a></li>
          @else
            <a href="{{ url('/home') }}" class="">
              {{ Auth::user()->name }} <span class="caret"></span>
            </a>

            <span><li><a href="{{ url('/logout') }}">Logout</a></li></span>
          </i></span>
          @endif
        </ul>

        <!-- Right Side Of Navbar -->
        <ul class="nav navbar-nav navbar-right">
          <!-- Authentication Links -->
          <form action="" method="">
            <div class="search">

```

```

        <input type="image" src="img/search.png">
        <input type="text" name="term" id="term" class="form-control"
placeholder="поиск по заведениям">
    </div>
</form>
</ul>
</div>
</div>
</nav>

```

```
@yield('content')
```

```

<footer>
    <div class="wrapper">
        <div class="foot-logo"></div>
        <div class="copyright">
            <span>© 2016 banqet.kz</span>
            <p>Любое использование материалов сайта<br> допускается только
при наличии гиперссылки на <a href="{{url('.')}}">banqet.kz</a></p>
        </div>
        <div class="foot-menu">
            <ul>
                <li><a href="#">Реклама</a></li>
                <li><a href="#">Вакансии</a></li>
                <li><a href="#">Контакты</a></li>
            </ul>
        </div>
    </div>
</footer>

```

```

<!-- JavaScripts -->
<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/2.2.3/jquery.min.js"
integrity="sha384-
I6F5OKECLVtK/BL+8iSLDEHowSAfUo76ZL9+kGAgTRdiByINKJaqTPH/QVN
S1VDb" crossorigin="anonymous"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/twitter-
bootstrap/3.3.6/js/bootstrap.min.js" integrity="sha384-
0mSbJDEHialfmuBBQP6A4Qrprq5OVfW37PRR3j5ELqxs1yVqOtnepnHVP9aJ7
xS" crossorigin="anonymous"></script>
<script type="text/javascript" src="../js/main.js"></script>
{{-- <script src="{{ elixir('js/app.js') }}"></script> --}}
<script src="js/jquery-1.12.3.min.js"></script>
<script type="text/javascript" src="js/ion.rangeSlider.js"></script>
<script type="text/javascript" src="js/fotorama.js"></script>

```

```

    <script type="text/javascript" src="js/main.js"></script>
</body>
</html>
@extends('layouts.app')

@section('content')
<section class="content">
    <div class="wrapper">
        <h1>Нашлось заведений: {{ count($restaurants) }}</h1>

        <div class="col-md-10 col-md-offset-1">
            @foreach($restaurants as $rest)
                <div class="result">
                    <div class="result-img">
                        <div class="fotorama" data-width="460" data-height="220">
                            
                            
                            
                            
                            
                        </div>
                    </div>
                    <div class="result-info">
                        <ul>
                            <li><h2>{{ $rest->title }}</h2></li>
                            <li>{{ $rest->status }}</li><br>
                            <li>Всего мест: <span>{{ $rest->seat }}</span></li>
                            <li>Паркинг: <span>{{ $rest->parking }}</span></li>
                            <li>Средний счет на персону: <span>{{ $rest->avg_price }}</span></li>
                            <li>Разрешены свои: <span>{{ $rest->allowed_self }}</span></li><br>
                            <li>Время работы: <span>{{ $rest->working_hours }}</span></li>
                            @foreach($rest->places as $place)
                                <li>Адрес: <span>{{ $place->address }}</span></li>
                                <li>Телефоны: <span>{{ $place->phones }}</span></li>
                            @endforeach
                        </ul>
                        <a href="#" class="more">Подробнее</a>
                    </div>
                </div>
            @endforeach
            {{ $restaurants->links() }}

```

```

    </div>
  </div>
</section>
@endsection
@extends('layouts.app')

@section('content')
<section class="content">
  <div class="wrapper">
    <div class="col-md-10 col-md-offset-1">
      @foreach($restaurant as $res)
        <div class="result">
          <div class="result-img">
            <div class="fotorama" data-width="460" data-height="220">
              
              
              
              
              
            </div>
          </div>
          <div class="result-info">
            <ul>
              <li><h2>{{ $res->title }}</h2></li>
              <li>{{ $res->status }}</li><br>
              <li>Всего мест: <span>{{ $res->seat }}</span></li>
              <li>Средний счет на персону: <span>{{ $res-
>avg_price }}</span></li>
              <li>Разрешены свои: <span>{{ $res-
>allowed_self }}</span></li><br>
              <li>Время работы: <span>{{ $res-
>working_hours }}</span></li>
            </ul>
            <a href="#" class="more">Подробнее</a>
          </div>
        </div>
      @endforeach
    </div>
  </div>
</section>
@endsection
@extends('layouts.app')

@section('content')

```

```

    <div class="input-group">
        <input type="image" src="img/search.png">
        <input type="text" name="term" id="term" class="form-control"
placeholder="поиск по заведениям">
    </div>
@endsection
@extends('layouts.app')

@section('content')
<div class="container">
    <div class="row">
        <div class="col-md-10 col-md-offset-1">
            <div class="panel panel-default">
                <div class="panel-heading">Dashboard</div>

                <div class="panel-body">
                    You are logged in!
                </div>
            </div>
        </div>
    </div>
</div>
@endsection
@extends('layouts.app')

@section('content')
<div class="container">
    <div class="row">
        <div class="col-md-10 col-md-offset-1">
            <div class="panel panel-default">
                <div class="panel-heading">Welcome</div>

                <div class="panel-body">
                    Your Application's Landing Page.
                </div>
            </div>
        </div>
    </div>
</div>
@endsection
<?php

class TestCase extends Illuminate\Foundation\Testing\TestCase
{

```



```

/**
 * The base URL to use while testing the application.
 *
 * @var string
 */
protected $baseUrl = 'http://localhost';

/**
 * Creates the application.
 *
 * @return \Illuminate\Foundation\Application
 */
public function createApplication()
{
    $app = require __DIR__.'/../bootstrap/app.php';

    $app->make(Illuminate\Contracts\Console\Kernel::class)->bootstrap();

    return $app;
}
}

```