

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

Коммерциялық емес акционерлік қоғамы  
АЛМАТЫ ЭНЕРГЕТИКА ЖӘНЕ БАЙЛАНЫС УНИВЕРСИТЕТИ

«Компьютерлік технологиялар» кафедрасы

«Қорғауға жіберілді»  
Кафедра меңгерушісі  
ф.-м.ғ.д., проф. Құралбаев З.Қ.

\_\_\_\_\_ (қолы)  
« \_\_\_\_\_ » \_\_\_\_\_ 2014 ж.

## ДИПЛОМДЫҚ ЖОБА

Тақырыбы: «Даулет-Жаркент» кәсіпқой спорт клубына арналған web-  
қосымша  
5B070400 – «Есептеу техникасы және бағдарламалық қамтамасыз ету»  
мамандығы бойынша

Орындаған Агимова Жансая Құдайбергенқызы тобы: ВТк-10-02

Жетекші аға оқытушы Г.Т.Мусатаева

Кеңесшілер :

Экономикалық бөлім бойынша :

\_\_\_\_\_ доцент Боканова Г.Ш.  
« 28 » 05 2014ж.  
(қолы)

Өмір тіршілігі қауіпсіздігі бойынша:

т.ғ.к., аға оқытушы Муташева Г.С.

\_\_\_\_\_ « 20 » 05 2014 ж.  
(қолы)

Есептеу техникасын қолдану бойынша :

\_\_\_\_\_ аға оқытушы Г.Т.Мусатаева  
« \_\_\_\_\_ » \_\_\_\_\_ 2014 ж.  
(қолы)

Мөлшер бақылаушы:

\_\_\_\_\_ аға оқытушы Рахимжанова З.М..  
« 3 » 06 2014 ж.  
(қолы)

Пікір жазушы :

\_\_\_\_\_ ф.-м.ғ.к., доцент Байтуленов Ж.Б.  
« \_\_\_\_\_ » \_\_\_\_\_ 2014 ж.  
(қолы)

Алматы 2014

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

Коммерциялық емес акционерлік қоғамы  
АЛМАТЫ ЭНЕРГЕТИКА ЖӘНЕ БАЙЛАНЫС УНИВЕРСИТЕТИ

«Ақпараттық технологиялар» факультеті  
«Есептеу техникасы және бағдарламалық қамтамасыз ету» мамандығы  
«Компьютерлік технологиялар» кафедрасы

жобаны орындауға берілген

ТАПСЫРМА

Студент Агимова Жансая Құдайбергенқызына

Жоба тақырыбы «Даулет-Жаркент» кәсіпқой спорт клубына арналған web-  
ректордың «24» <sup>қосымша</sup> бұйрық № 115 бұйрығы бойынша бекітілген.

Аяқталған жұмысты тапсыру мерзімі: «28» сәуір 2014 ж.

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

Web-ресурстар келесі қасиеттеріне ие болу керек:  
мәқалаларда жариялау және өңдеу;  
мәқалалар категориялар бойынша топталу;  
бар категорияларда өңдеу, мақаланы қосу;  
әр мақала бойынша пікір жазуға мүмкіндігі;

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

Жобаның басты мақсаты интернет технологияларын қолдана отырап «Даулет-Жаркент» спорт клубының мақалаларына және сайтта өзінің сайтқа ұстанатын адамдар арасында спорт мақалаларында пікір алмасуға, бағалауға және мақалалармен байланысты әртүрлі мақалаларды веб-ресурстар арқылы

Сызба материалдарының (міндетті түрде дайындалатын сызуларды көрсету) тізімі:

1. Деректердің ER диаграммасы
2. Web-қосымшаға байланысты берілетін мәстер және процедуралар диаграммасы
3. Деректер қорын құрудың тиімділігіне сыйласы.
4. Web-қосымшаға байланысты берілетін кластар диаграммасы

Негізгі ұсынылатын әдебиеттер:

1. Камалова Н., Мовлева Е. HTML. Самоучитель. 2-е изд. - СПб. Питер, 2011 - 288 с
2. Шадрин С. HTML, XHTML и CSS Базисная технология 5 изд. - М.: Диалектика, 2010 - 656 с
3. Шашиков П. Веб-мастеринг на 100%. HTML, CSS, Java Script, PHP, CMS, AJAX, разработка - СПб. Питер, 2010 - 512 с
4. Базылов К. Б., Амбаева С. А., Курмаганбетова С. С. Бітірілген ықпалының экономикалық бәлімі үшін әдістемелік құралдар - Алматы. АИЭС. 2009

Жоба тараулары бойынша кеңес берушілер және оның мерзімі:

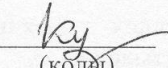
Бөлім	Кеңесші	Мерзімі	Қолы
Негізгі бөлім	Мусатаева Г.Т.		Мусаф
Тіршілік қауіпсіздігі	Муташева Г.С.		Муташева
Экономикалық бөлім	Боканова Г.Ш.		Боканова
Мөлшер бақылаушы	Рахимжанова З.М.		Рахимжанова
Есептеу техникасын қолдану	Мусатаева Г.Т.		Мусаф

диплом жобасын дайындау

КЕСТЕСІ

№ р/с	Тарау аттары, әзірленетін сұрақтардың тізімі	Жетекшіге ұсыну мерзімдері	Ескерту
	1. Интернет технологиялар	02.02.14	
	1.1. Қазіргі таңдағы веб-технологияның маңызы		
	1.2. Веб технологиялар арқылы ақпараттарды қолдану үйымдастыру	20.02.14	
	1.3. Интернеттер базасын құру тәсілдері	12.03.14 15.03.14	
	2. Интернет коммуникация түрлерінің қызметін тәсілдері арнаған Веб-қосымшаны құру		
	3. Бағдарламалық экран-шоттар	20.04.14	
	4. Адаптивтілік қауіпсіздігі	20.05.14	
	5. Техника-экономикалық келісімдер	15.05.14	

Тапсырманың берілген уақыты « \_\_\_\_\_ » \_\_\_\_\_ 2014 ж.

Кафедра меңгерушісі  (қолы) ф.-м.ғ.д., проф. Құралбаев З.Қ.

Жоба жетекшісі  (қолы) аға оқытушы Мусатаева Г.Т.

Орындалатын тапсырманы қабылдаған студент  (қолы) Агимова Жансая Құдайбергенқызы

**Андатпа**

Жобаның мақсаты пайдаланушыларға ыңғайлы болатын «Даулет–Жаркент» кәсіпқой спорт клубына арналған веб–қосымша құру. Дипломдық

жоба PHP5, CSS, HTML және JavaScript тілдерін қолдана отырып құрылған, ал деректер қоры MySQL негізінде жүргізілген.

«Тіршілік қауіпсіздігі» бөлімінде еңбек жағдайларына, жұмыс орнына талдау жасалынған. Инженердің жұмыс орнындағы жасанды жарықтандыру бойынша есептеулер жүргізілген.

Ал техника-экономикалық негізделуі бөлімінде барлық баптар бойынша жобаны жүзеге асырудағы шығындар анықталған және жобаның өзіндік құны мен зияткерлік еңбектің бағасы есептелген.

### **Аннотация**

Целью проекта является – разработка веб-приложения для профессионального спорт-клуба «Даулет–Жаркент» .

Дипломный проект разработан с использованием языков PHP5, CSS, HTML и JavaScript, а база данных проведена на основании MySQL.

В главе «Безопасности жизнедеятельности» анализируются трудовые условия и рабочее место. На основании полученных данных сделан расчет по искусственному освещению.

В главе «Технико-экономического обоснования проекта» по всем статьям определены расходы по осуществлению проекта, себестоимость и цена на реализацию.

### **Мазмұны**

Кіріспе .....	7
1 Ақпараттық жүйені құрудың негізгі технологиялары және қолданылатын бағдарламалық құралдар .....	13

1.1	Интернет туралы жалпы түсінік .....	13
1.2	Гипермәтін концепциясы .....	15
1.2.1	World Wide Web .....	15
1.3	Apache HTTP Server жобасы .....	17
1.3.1	Apache архитектурасы .....	18
1.4	Бағдарламалау құралдарын таңдауды негіздеу .....	19
1.4.1	HTML гипермәтіндік тілі туралы түсінік .....	20
1.4.2	MySQL деректер қорының сервері .....	24
1.4.3	Java Script скрипті бағдарламалау тілі .....	32
1.4.4	CSS стильдер тілінің негізі .....	36
1.4.5	PHP бағдарламалау тілі .....	37
1.4.6	Yii Framework платформасы.. <b>Ошибка! Закладка не определена.</b>	
2	Веб-порталды жобалау және өңдеу .....	45
2.1	Дипломдық жұмысқа қойылған есептер .....	45
2.2	Веб-порталды құру кезеңдері .....	45
2.3	Веб-портал жұмысының алгоритмі .....	48
2.4	Техникалық тапсырма .....	<b>Ошибка! Закладка не определена.</b>
2.5	Веб-портал құрылымы .....	51
2.6	Веб-порталдың функционалдығы <b>Ошибка! Закладка не определена.</b>	
2.7	Дизайн және беттеу .....	56
3	Техника-экономикалық негізделуі .....	57
3.1	Жұмыс сипаттамасы және қажеттілік дәлелдемесі <b>Ошибка! Закладка не определена.</b>	
3.2	Жұмыста пайдаланылатын еңбек ресурстары; ... <b>Ошибка! Закладка не определена.</b>	
3.3	Жобалау және өңдеу бойынша жұмыс құнын есептеу <b>Ошибка! Закладка не определена.</b>	
3.4	Амортизациялық шығындар өлшемі <b>Ошибка! Закладка не определена.</b>	
3.5	Зияткерлік еңбектің бағасы .....	<b>Ошибка! Закладка не определена.</b>
3.6	Жұмыстың техника-экономикалық негізделуіне қорытынды . <b>Ошибка! Закладка не определена.</b>	
4	Тіршілік қауіпсіздігі .....	57
4.1	Оператордың жұмыс жағдайын талдау .....	<b>Ошибка! Закладка не определена.</b>
4.2	Кондиционерлеу және кондиционерді таңдап, есептеу .....	<b>Ошибка! Закладка не определена.</b>
	Қорытынды .....	84
	Әдебиеттер тізімі .....	85
	Қосымша А .....	86

## Кіріспе

Бүгінгі күні адамдардың өмірін интернетсіз елестету мүмкін емес. Өйткені, ол күнделікті жаңалықтармен, аудио, видео өнімдермен танысу немесе әлеуметтік желілерде уақыт өткізу ғана емес, сонымен қатар әр түрлі оқуға, жұмысқа керекті ақпаратты іздеуге, танысуға, оқып білуге, сонымен бірге адамдардың өздерінің бизнестерін дамыту үшін керек.

Интернеттің басты тапсырмасы бұл – әр тәуліктік, жоғары сенімді байланыс. Интернетке қосылған кез келген екі компьютер (немесе басқа құрылғылар) бірбірімен кез келген уақытта хабарласа алады. Ары қарай “Желі” сөзін қолданғанда, Интернет сөзіне синоним ретінде Желімен интернет арқылы екі компьютерді байланыстыру мүмкіндігін және олардың өзара байланысын қамтамасыз етуді түсінеміз. Интернетке қосылған әрбір компьютер – бұл Желінің бір бөлігі.

WWW технологиясы алғашқысынан әр түрлі елдердің ғалымдары арасында ақпаратпен ыңғайлы алмасу үшін жасалған болатын. Арнайы бағдарламаны пайдалана отырып, жер бетінің бір жағындағы ғалымдар екінші жақтағы ғалымдардың дайындаған ақпараттарына қол жеткізе алатын болды. Сонымен бірге, олар қысқаша пікірлерімен алмасуға және құжаттарын дұрыстауға мүмкіндіктері пайда болды, бұл қызмет түрі қазір «электрондық пошта» деп аталады.

Қарым–қатынасу мен қызметтестіктің жаңа түрі өте жағымды болды, сондықтан да тез арада бәріне қол жетімді ресурсқа айналып, Интернет желісінің ең әйгілі мүмкіндігі деп танылды.

Менің дипломдық жобамның тақырыбы «Даулет–Жаркент» кәсіпқой спорт клубына арналған веб–қосымша құру». Менің бұл тақырыпты таңдаған себебім:

- біріншіден, қазіргі күнде қазақ тілінде веб–қосымша аз;
- екіншіден, әр адам сайтқа кіру арқылы қызықты әрі жаңа жаңалықтармен таныса алады және машықтану бағдарламаларын суреттелген немесе бейнематериалдардан қарауға болады. Бұл өте ыңғайлы.

Дипломдық жұмыс келесі бөлімдерден тұрады:

- кіріспе;
- жұмыс жасау әдістері мен бағдарламалық құралдарын таңдау;
- жұмыс жасалу барысы;
- қорытынды.

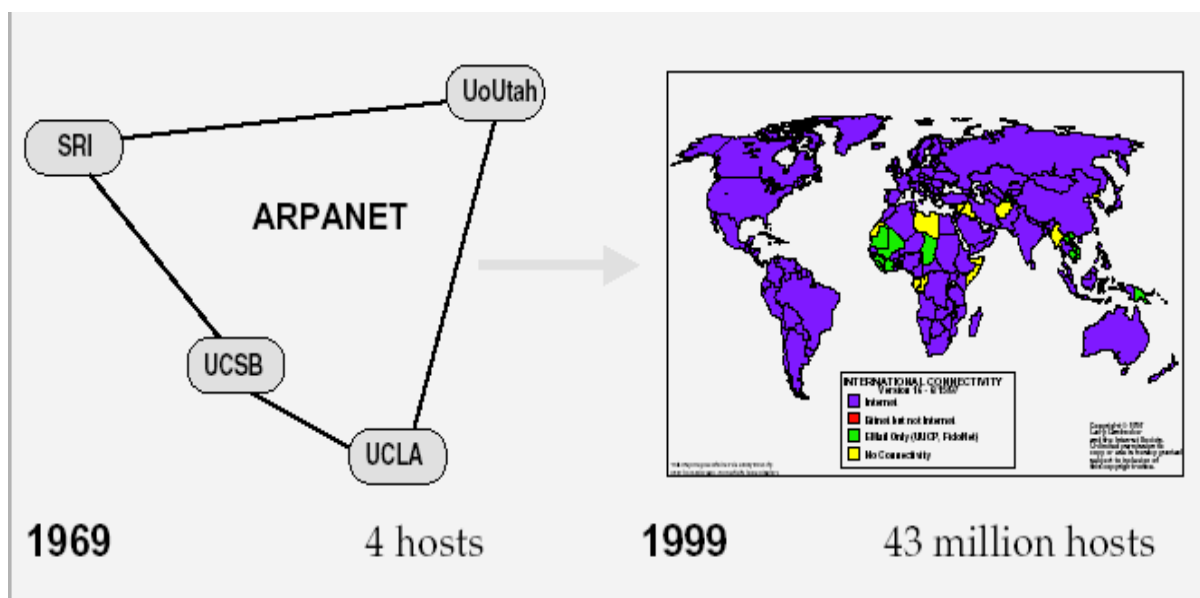
# 1 Ақпараттық жүйені құрудың негізгі технологиялары және қолданылатын бағдарламалық құралдар

## 1.1 Интернет туралы жалпы түсінік

Интернет дегеніміз компьютерлерді бір – бірімен белгілі хаттама арқылы (TCP/IP) байланыстыратын желі, ол әр компьютерлерді өзара әр түрлі жаңалықтар алмасуына ықпал етеді. Интернеттің кросс– қаттамалық тәуелсіздігі оның күн сайын қуат алуына және экспоненциалдық өсуіне себепші.

Сонымен, желі дегеніміз не? Интернетте мыңдаған компьютерлер бір бірімен байланысады. Бұл өте жеңіл сияқты көрінеді және көптеген адамдар Интернет сөзін күнде қолданады, санаулы адамдар ғана оның концепциясын түсіне біледі.

1957 жылы бірінші Спутник атты жасанды жер серігін жіберді. Бұл АҚШ –та үлкен әсер етіп, олар Advanced Research Projects Agency (ARPA) атты Department of Defense (DoD) әскери ұйым жанынан ғылыми –технологиялық бөлім құрады. 1969 жылы DoD APRANET атты төрт түйіннен тұратын желі құрады: California және Los Angeles (UCLA) университеттері, Stanford Research (SRI) институты, California Santa Barbara (UCSB) университеті және Utah университеті 1.1–суреттен көре аламыз.



1.1 сурет – Төрт түйіннен тұратын желі

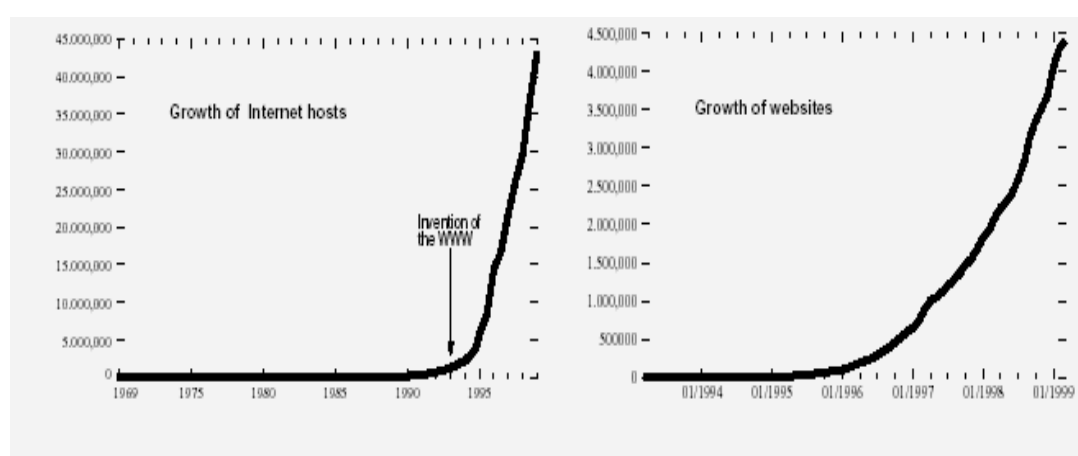
Бұл желі 50 Кбит/с жылдамдығымен Network Control Protocol (NCP) арқылы бірінші түйін – түйін қаттамаларынан тұрды. Жылдар бойы көптеген түйіндер ARPANET желісіне қосылып, және де ол желі қаттамалар мен бағдарламалық жабдықты қолдана бастады.



1974 жылы Vint Cerf және Bob Kahn “A Protocol for Packet Network Interconnection” атты Transmission Control Program (TCP) құрды.

Бағдарламасының құрылымы мен құрылысы туралы жинақ жарық көрді. 1978 жылы TCP екі қаттамадан тұрды: Transmission Control Protocol (TCP) және Internet Protocol (IP).

1982 жылы DoD TCP және IP (көбісі оны TCP/IP деп біледі) қаттамаларын ресми түрде қабылдады. 1983 жылдың қаңтар айында ARPANET ресми түрде NCP және TCP/IP қосылуы арқылы, Интернет пайда болды. Таңғажайып Интернеттің таралу жолдары: 1984 жылы қосылған түйіндердің саны 1000 болды, 1987 жылы ол 10000 –нан асты, 1989 жылы 100000–нан асты, 1992 жылы 1000000 болды, 1996 жылы 10 000 000 болды. Қазірде Интернет желілерінің саны 50 000 000–нан асты. Ол әлі де өсуде, оны 1.2 – суреттен көре аламыз.



1.2 сурет – Интернеттің өрістеу (қосылған түйіндер саны) және World Wide Web (серверлер саны)

## 1.2 Гипермәтін концепциясы

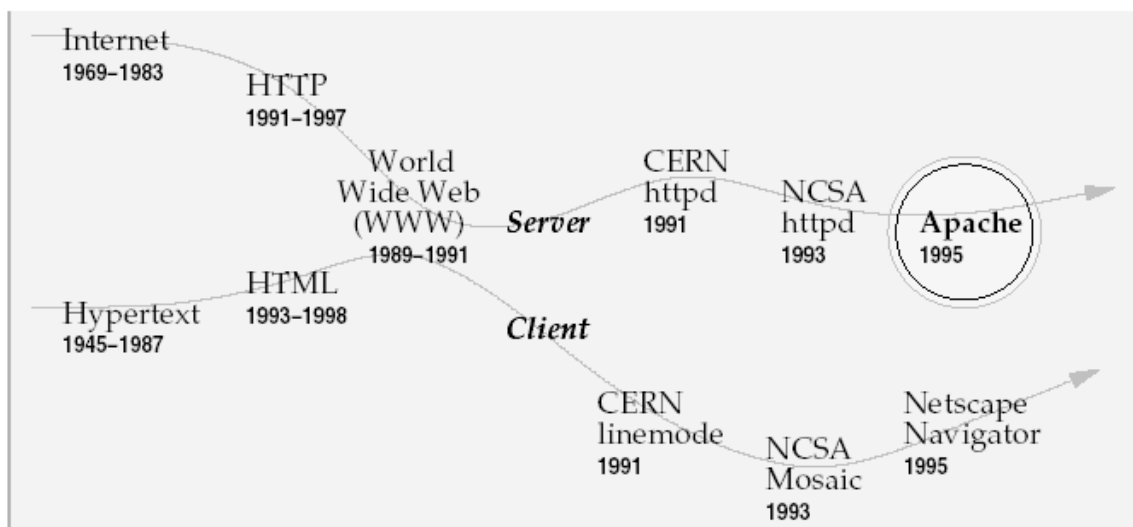
1965 жылы Ted Nelson гипермәтін концепциясын енгізді. Сонан соң, Brown Университетінде 1967 жылы Hypertext Editing System (HES) атты гипермәтіндік құжаттық жүйе жасады. 1968 жылы Douglas C. Engelbart (1963 жылы тышқанды ойлап тапқан) “ONLine System” атты жүйесін көрсетеді. Осы жағдай тышқан, гипермедия және бейне конференция туралы дау тудырды.

Сонан соң гипермәтін жобасын дамытатын көптеген жүйелер жасалды (1975: ZOG Carnegie Mellon университетінде; 1978: Aspen Movie Map; 1984: Filevision by Telos; 1985: Symbolics Document Examiner by Janet Walker; 1985: Intermedia by Norman Meyrowitz at Brown University; 1986: Guide from OWL, NoteCards from XeroxPARC, және т.б.). 1987 жылы Apple HyperCard атты Bill Atkinson жүйені жасап шығарды. Hyper –Card дегеніміз «программалық жабдықтың бастауы». ТЕХТ'87 атты конференция гипермәтінді концепцияға бағытталған болатын. Конференцияда қаралған бойынша: «Гипермәтінді

текстті немесе басқа жаңалықты көрсетеді. Біз жалғай алатын немесе түйін (хост) деп атайтын заттар линктер арқылы байланысады».

### 1.2.1 World Wide Web

The client side. WWW 1.3–суреттегі клиенттік жағы дегеніміз екі жағдай арқылы белгіленеді: Hypertext Markup Language (HTML) және қолданушылар компьютерінде орналасқан белгілі браузерлер арқылы. 1993 жылы бірінші HTML версиясы National Center for Supercomputing Applications (NCSA) ұйымында жасалып, Mosaic деп аталған.



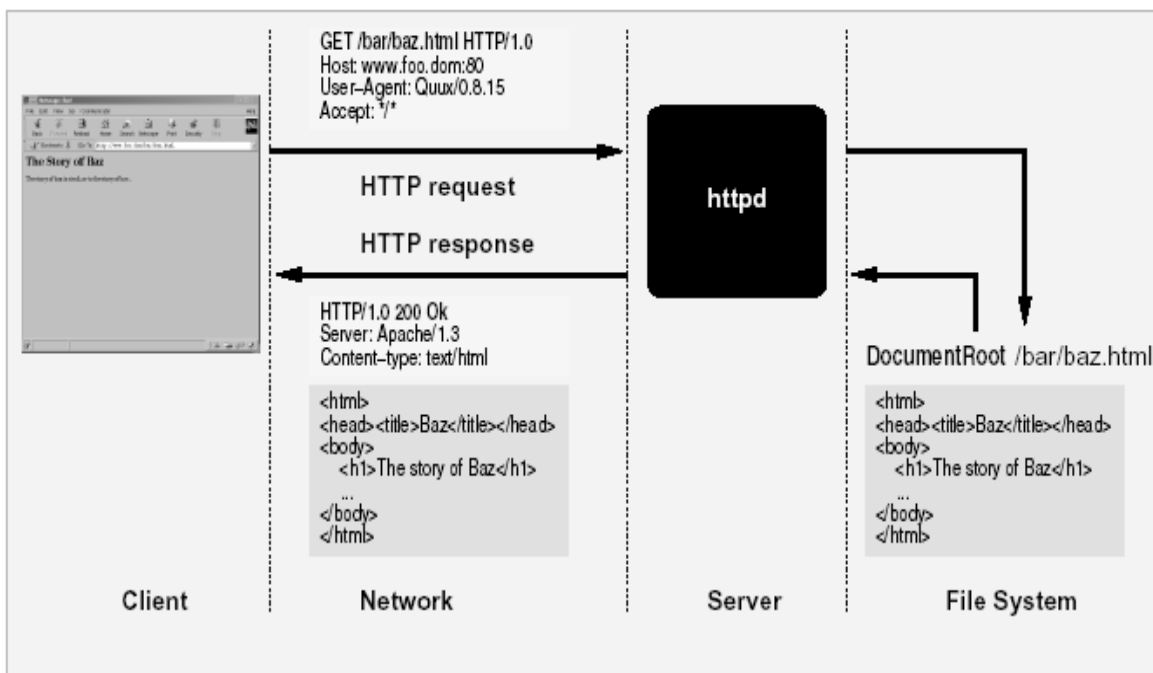
1.3 сурет – World Wide Web эволюциясы

Онан кейін белгілі Netscape Navigator браузері оның жалғасы болды.

HTML SGML атты кішкене гипертмәтіндік тілден тараған, сонан кейін ол үлкен комплексті гипермәтіндік тіл болды (мысалы, version 4.0). Басқа тілдермен қатар және объекті модельдермен (мысалы, JavaScript, DOM), графикалық форматтармен (мысалы, GIF, JPEG, PNG), және мультимедия (мысалы, audio, video), WWW клиенттік жағы түстерге өте бай, комплексті, кейде хаосты аудан болып табылады. Сол себепті көптеген адамдар клиент жағы өте әсерлі болғандықтан WWW мен байланыстырады да, Интернеттің басқа жағы – сервер туралы біле бермейді.

The server side. Сервер жағы бір қараған сәтте қарапайым қолданушыға қызықты да емес. Онда түрлі –түсті суреттер және т.б. жоқ. Бірақ бұл –Apache жері, яғни WWW–ның қызықты да және қиын жағы.

HTTP серверінің рөлі. Web сервер браузерден сұраныс алып, HTTP байланысы арқылы жауабын жібереді. 1.4 – суреттен көре аласыз.



1.4 сурет – Web сервер рөлі

Бұл жағдай бір қарағанда оңай сияқты көрінгенімен, көптеген сұраныстар анық болмайды және көп уақытты талап етеді (web сервер бір уақытта көптеген HTTP сұраныстарын қанағаттандыру керек), сервер оңай байланысты қалыптасуы керек, оңай орын ауысуы керек (Apache негізгі серверлі қаттамаларда жіберіле береді), сенімді және т.б. Бірақ, Apache ең тез және ықшам сервер болмаса да, дүние жүзіне тарауына себепші болып және сенімділігі болып табылады.

### 1.3 Apache HTTP Server жобасы

Apache HTTP server жобасы дегеніміз не? HTTP сервер жобасы деп Apache тобының жобасын айтамыз. Яғни жаңа коммерциялық пайдасы бар, өзгешелігі, қолдануда оңайлығы, тегін код арқылы жасалған HTTP сервер. Бұл серверді көбінесе “Apache” деп аталады. Ал бұл серверді жасаушы волонтерларды “the Apache Group” деп атайды.

Қалайша Apache HTTP server жобасы бастауын алған? Let Roy T. Fielding (HTTP әкесі деуге болады) Apache тобының мүшесі оны былайша суреттейді: “1995 жылдың ақпан айында ең атақты Web сервер бағдарламалық жабдығын Rob McCool мен National Center for Supercomputing Applications, University of Illinois, Urbana–Champaign. However ұйымдары жасады. Көптеген программистер күштерін бірлестіре отырып, интернет, электронды пошта арқылы серверді жаңа функциялармен, мәтіндерін жамау мен айналасты (оларды patch деп атаймыз).

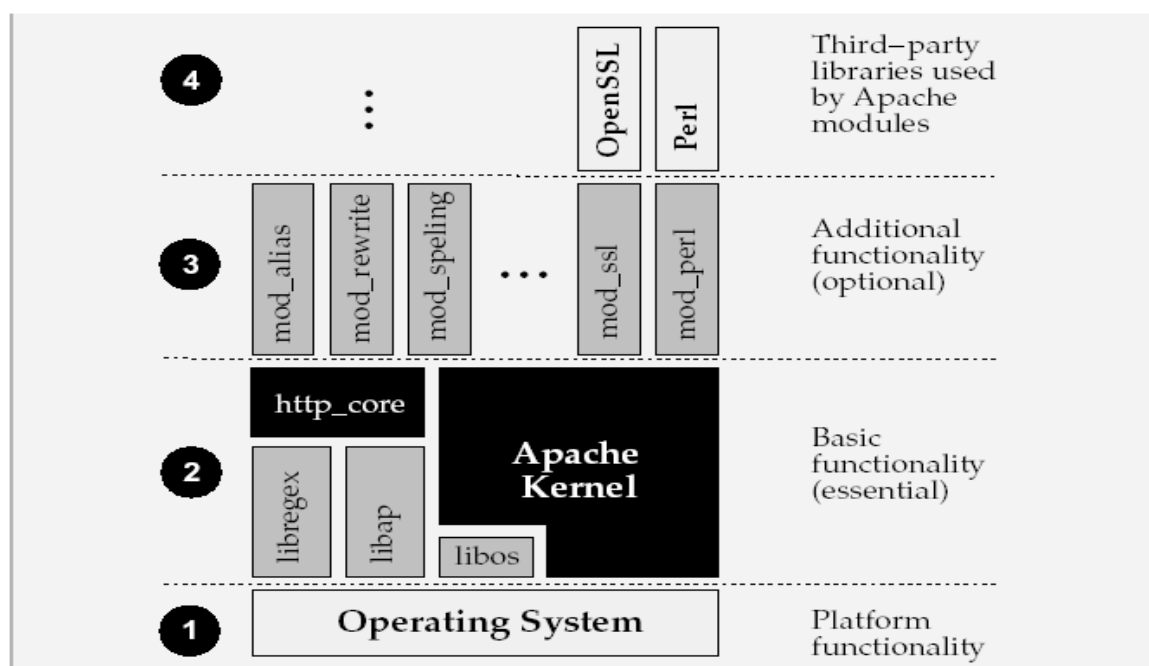
Көптеген волонтерлер Apache Web сервері үшін тер төгіп, ол дүние жүзінде ерекше орын алды. Оның эволюциялық версияларын келесі 1.1–кестеден көре аласыз.

К е с т е 1.1 – Apache эвалюциялық версиясы

Version	Lines of Code		
	Code	Comments	Total
1.0.5	11.551	6.099	17.650
1.1.3	18.896	9.786	28.682
1.2.6	33.526	15.715	49.241
1.3.3	52.341	24.956	77.297
1.3.12	69.646	31.041	100.687

### 1.3.1 Apache архитектурасы

1.5 – суретте сізге Apache web серверінің архитектурасын көрсетеді. Бұл архитектура бірнеше деңгейден тұрады.



1.5 сурет – Apache web серверінің архитектурасы

Let Roy T. Fielding (HTTP әкесі деуге болады) Apache тобының мүшесі оны былайша суреттейді: “1995 жылдың ақпан айында ең атақты Web сервер бағдарламалық жабдығын Rob McCool мен National Center for Supercomputing Applications, University of Illinois, Urbana–Champaign. However ұйымдары жасады. Көптеген программистер күштерін бірлестіре отырып, интернет, электронды пошта арқылы серверді жаңа функциялармен, мәтіндерін жамау мен айналасты (оларды patch деп атаймыз).

## 1.4 Бағдарламалау құралдарын таңдауды негіздеу

Спорттық веб–порталын жобалау үшін келесі программалық құралдар қолданылды:

- Apache 2.0.55;
- HTML 5;
- MySQL 5.0.24;
- JavaScript;
- CSS 3;
- PHP 5.0.4.

Apache – әлемдегі танымал веб–серверлердің бірі, оның ерекшеліктері: баптаудың үлкен мүмкіндіктері, веб–сервер жұмысы үшін көп түрлі қаттамалардың қолдауы, жоғары өнімділік, көптеген бағдарламалау тілдерімен және дерекқорлармен интеграциялану мүмкіндігі, ашық бастапқы код, түрлі операциялық жүйелермен (Windows, Unix, Linux) сәйкестігі.

Apache–дың тағы бір артықшылығы – оның көп тілге аударылған толық құжаттаманың болуы, және де баптау файлдарының салыстырмалы қарапайымдылығы.

Сонымен қатар оның жақсы масштабталушылығы мен кеңейтілімділігін, белгілі бір сервердің жұмысына лайық тек қажетті модульдердің бастапқы кодтарын жинақтау мүмкіндігін атап өткен жөн. Apache–дың аппараттық қамтамаға күй талғамайтындығы оның қазіргі компьютерлердің кез–келгеніне қосуға мүмкіндік ашады.

MySQL – бұл жылдам, сенімді және қолдануда жеңіл реляциялық деректер қорымен басқаратын жүйе. MySQL МББЖ GNU GPL лицензиясымен таратылатын бастапқы коды ашық программалық қамтама. Бірнеше жыл бойы MySQL өндірістік жағдайларда тиімді қолданады және қажетті тұрақтылық пен жұмыстың жоғары жылдамдығын қамтамасыз ете отыра, үлкен дерекқорлармен басқаруда қолданылады. Сонымен қатар, MySQL дамыған функционалдылық, қол жетерлік және қауіпсіздік сияқты қасиеттерге ие, және де осы қасиеттер Интернет арқылы дерекқорға қатынауды ұймдастыруға жақсы келеді. MySQL–дың танымалдығына тағы да оның әйгілі веб–программалау тілдерінде сәйкес модульдердің бар болуы үлес қосады.

MySQL клиенттердің мәліметтер серверіне қосылудың түрлі әдістерін қолдайды. Қосылу TCP/IP сокеті арқылы кез–келген қаттамада мүмкін. Одан басқа, түрлі қаттамаларда арнайы қосылу әдістері болады: Windows операциялық жүйесінде бұл атаулы арналар (named pipe), Unix–ортада бұл сокеттер. Connector/ODBC және Connector/JDBC интерфейстері MySQL–ге сәйкесінше ODBC және JDBC байланыстарын қолданатын клиенттік программаларды қолдайды.

Жобалаушылармен MySQL көптеген тілдерге аударылған. Сервер Unicod–ты қосқанда түрлі кодтық кестелерді қолдайды, қателер туралы хабрламаларды түрлі тілдерде жариялай алады, MySQL мәліметтер базасында ақпарат таңдалынған кодта сақталынады.

PHP – негізінен түрлі Интернет–қосымшалар саласында қолданылатын скрипттік программалау тілі. PHP құрылымының синтаксисі C++ программалау тілінің синтаксисіне ұқсайды. PHP – бұл Perl–дің орнына келген салыстырмалы жас тіл, ол негізінен веб–программалауға арналған, үйренуге күрделі емес, оның практикалық қолдануын кеңейтетін қосылатын түрлі модульдері бар. PHP–ді қолданудың басты мақсаты – қолданушының талабына сәйкес динамикалық HTML құру.

PHP–дің Perl тілімен салыстырғанда оның синтаксисінің жеңілдігінде. PHP веб–сервердің бір бөлігі сияқты жұмыс істейді, яғни хостингтің қорларын үнемдейді, CGI сценарийлерге қарағанда операциялық жүйеде жаңа үрдістерді тудырмайды. Әсіресе ол көп қаралатын көп бетті сайтта PHP кодының салыстырмалы қысқа үзінділері жазылған кезде байқалады. Дәл осы жағдайда Perl көптеген үрдістерді тудырап еді.

#### **1.4.1 HTML гипермәтіндік тілі туралы түсінік**

HTML (Hyper Text Markup Language) – гипермәтіндік өлшеуіш тілі күрделі SGML (Standard Generalized Markup Language) өлшеуіш тілінің мүшесі болып табылады. HTML кез келген тілдер сияқты программа құрудың стандартты құрылымынан тұрады. Бұл жағдайда ол HTML – құжат. HTML – тег директивалары бұрыштама жақшаларына алынады. Бұрыштама жақшаға алынбаған барлық объектілерді интерпретатор экранда бейнеленетін мәтін ретінде қабылдайды. Тег – браузер интерпретаторына тиісті нақты әр мәнді қалай өңдеуді көрсететін HTML командасы. Бұл мән тег атрибуты деп аталады. Тегтің атрибуты болуы да, болмауы да мүмкін. Мысалы, <HTML> тегінің атрибуты болмайды.

HTML – құжаты қысқа түрде құжат тақырыбынан және құжат денесінен тұрады. Осындай қолайлы да қызықты тілді үйрену арқылы өз білімімізді дамыта алдық.

Интернеттің барлық WEB парақтарына ортақ белгі бар, олардың барлығы HTML тілінде жазылған. WEB парақтарды құру процесі программалау процесіне өте ұқсас болғанменен, HTML программалаудың ерекше тілі. HTML–гипертексті белгілеу тілі. Ол жай мәтінді WEB парақ түрінде көрсететіндей ережелерді анықтайды.

Интернет дамуының жаңаша кезеңі 90-шы жылдары ақпарат алмасудың жаңа қаттамаларының пайда болуымен басталды. Бұл қаттама HTTP (Hyper Text Transfer Protocol – гипертексті мәтін беруқаттамасы). Бұл қаттамамен бірге файлдарды интернет арқылы беретін HTML серверінің ауқымды желісін көрсететін World Wide Web қызметі пайда болды. Бұл файлдардың негізгі бөлігі HTML тілінде жазылған Web парақтар арнайы файлдарды көрсетеді. Интернетте Web парақтар осындай файлдардың HTTP серверінде орналастырылуымен көрсетіледі.

Web парақтар әр түрлі болуы және еркін тақырыптарға арналуы мүмкін. Бірақ олардың барлығы бір негізді – HTML тілін қолданады. Әдетте, HTML

құжаттарының кеңейтулері мынадай болады: НТМ немесе HTML. HTML тілі, ол World Wide Web қызметімен қатар пайда болып, онымен бірге дамып отырды.

Гипермәтін – бұл форматтау, иллюстрация, мультимедиялық қойылымдар және гиперссылкалар сияқты қосымша элементтерді басқаратын арнайы кодтары бар мәтін. Бұл баяндамадағы ең маңыздысы – тіл.

Web парақтардың компьютер экранында өңделген түрде берілуіне қарамастан HTML тілі құжаттарды өңдеу үшін тағайындалмаған. Мысалы, Word мәтіндік процессор көмегімен мәтін өңделген кезде оның баспаға қандай принтер және қағазға басып шығарылатыны туралы алдын – ала анық болды. Ал құжат интернетте орналасса, оның қандай компьютерде қолданылатыны беймәлім болады.

Әртүрлі құрылғылар үшін құжаттарды дайындау қажет болғандықтан HTML тілі құжат форматын баяндауға арналмаған. Ол құжаттың функцияналды белгіленуі үшін, яғни мәтін фрагменттерінің тағайындалуын анықтау үшін қолданылады. Мысалы, егер мәтінде тақырып кездесе, онда HTML коды бұл фрагменттің тақырыбы екенін көрсетеді. Көру бағдарламасы (броузер) осындай кодты алып, тақырыппен не істеу керек екенін өзі шешеді.

Шығыс мәтінді белгілеу үшін қолданылатын HTML кодтары тегтер деп аталады. Тег – символдар жинағы. Барлық тегтер (<) символымен басталып, (>) символымен аяқталады. Осындай символдар жұбын кейде бұрыштық жақшалар деп те атайды. Ашылатын бұрыштық жақшадан кейін тегті анықтайтын кілттік сөз жүреді. HTML тілінде әр тегтердің арнайы тағайындалуы бар. Тегтер атауындағы әріптер регисторының маңызы жоқ – кіші әріп сияқты үлкен әріпті де қолдана беруге болады. HTML – дің 1 тегі құжаттың бір бөлігіне ғана әсер етеді. Мысалы, абзацқа. Мұндай жағдайда жұп: ашып–жабушы тегтері қолданылады. Ашушы тег әсер тудырады. Ал жабушы оның орындалуын тоқтатады. Жабушы тегтер (1) символымен басталады. Кейбір тегтер бір ретті әсер тудырады. Бұл кезде жабушы тегтің қажеті жоқ. Егер де тегте HTML тілінде жоқ кілттік сөз көрсетілсе, онда тег орындалмайды. Құжаттардың броузерде көрсету барысында тегтің өзі көрінбейді, бірақ құжаттың көріну әдісіне әсер етеді.

Кейде ашушы тегтерде эффектке әсер ететін атрибуттар болуы мүмкін. Атрибуттар – бұл қосымша кілттік сөздер. Кейбір атрибуттар, атрибуттардың мәнін көрсетуді талап етеді. Бұл мән кілттік сөзден (=) белгісімен бөлінеді.

Атрибут мәні тырнақшада тұру керек. Жабушы тегтердің ешқашан атрибуттары болмайды.

Бағдарламалау тілдерінде коментарий – мәтіндік қатарларды қолдану мүмкіндігі бар. Олар бағдарламаның бөлігі емес. HTML тілінің де осындай мүмкіндігі бар. HTML тілінде коментарий (<!—) арнайы тегімен басталады. Осы тегтің артынан жүретін сөз коментарий болып табылады. Коментарийді → символы аяқтайды. Коментарийде (<) символынан басқа, барлық символдар бола алады.

HTML құжаты құжаттың негізгі мәтінінен және жәй символдар жиыны болып табылатын белгілеу тегтерінен тұрады. Осылайша HTML құжаты жәй тексттік файл. Оны құру үшін тексттік редактор пайдаланылады.

Мәтіндік процессорлармен жұмыс істеу барысында әдетте тізімнің екі типі қолданылады: реттелген (нөмірленген) және реттелмеген. HTML тілі тізімнің S түрін құруға мүмкіндік береді. Бірақ қазіргі кезде тізімнің үш түрі кеңінен қолданылады: реттелген, реттелмеген, және анықтамалар тізімі.

Реттелген және реттелмеген тізімдер бір жолмен құрылады. Тізім әрқашан реттелген тізім үшін <OL> және </OL> реттелмеген үшін <UL> және </UL> ашушы және жабушы тегтерінің арасында орналасады. Тізім ішінде тізім элементтері бар. Олар үшін <LI> және </LI> тегтері қызмет етеді. <OL>, </UL> <LI> тегтерінде тізім элементтерін белгілеу әдісін көрсететін TYPE= атрибуты болуы мүмкін. Нөмірленген тізімді арап цифрларымен, рим цифрларымен, латын әріптермен нөмірлеуге болады. Реттелмеген тізімдер қара, ақ дөңгелек немесе тік бұрыш маркерлерімен белгіленеді. Тізім элементтерінде гиперсілтеуіш, сонымен қатар фраза элементтерін белгілейтін және өңдейтін тегтер болуы мүмкін. Анықтамалар тізімі <DL> тегінен басталып </DL> ден аяқталады. Маркерлеу және нөмірлеу бұндай тізімдерде қолданылмайды. Тізім анықталатын терминдер және сәйкес анықтамалардан тұрады. Анықталатын терминдер мен анықтамалар кезектеседі. Анықтамалар экранда сол жақ шеткі шегініспен көрінеді.

Мультимедияның кең таралуы World Wide Web қызметі бар болған кезіне сәйкес келеді. Аудио және видео файлдар әлі күнге дейін орналастырылған көмекші көмегімен орындалатын сыртқы объект ретінде қарастырылады. Видео және файлдың мультимедиялық дыбыстары әртүрлі формат қолданады. Жәй форматтар файлдың толық жүктелуін талап етеді. Жаңаша лектің форматтар дыбыс және видео көріністерді, олардың жүктелуі кезінде көрсетуге мүмкіндік береді. Интернеттегі лектік форматта радиотрансляциялар беріледі. Мультимедиялық объекті Web параққа қоюдың ең қарапайым әдісі бұл – <A> гиперсілтеуіш тегін қолдану. Кейбір жағдайларда Web параққа дыбыстық немесе мультимедиялық файлды енгізу қажет болады. Бұл мақсат үшін <EMBED> тегін қолданған ыңғайлы. Бұл тег Internet Explorer және Netscape Navigator броузерлармен танылады.

Кесте – бұл үлкен көлемді мәліметтерді ұсынудың ең ыңғайлы тәсілі. HTML тілі әртүрлі кестелер құрудың мүмкіндіктеріне бай. Кесте құруда қолданылатын негізгі тегтер:

– HTML тілінде кесте <TABLE> тегінен басталып, </TABLE> тегінен аяқталады. Кесте ішіндегі мәтін кесте элементтерін анықтайтын арнайы тегтерге алыну керек;

– <TABLE> және </TABLE> тегтерінің арасында кесте тақырыбын білдіретін <CAPTION> және </CAPTION> тегтері бір–ақ рет кездесе алады. Кесте тақырыбы кесте үстінде немесе кесте астында орналасады;



– онан кейін кесте қатарларын анықтайтын <TR> және </TR> тегтері жүреді. Кесте қатары келесі қатар алдында немесе кестемен аяқталатындықтан, жабушы тегін жазбауға болады;

– кестенің әрбір қатары ұяшықтардан тұрады. Олар қатар мен баған тақырыбы бар <TH> немесе жәй мәліметі бар <TD> тегтерімен белгіленеді;

– ұяшықтар ішінде құжат денесінде бола алатын кез келген мәлімет пен HTML тегтер болуы мүмкін. Мысалы, кесте ұяшығында қосалқы кесте болу мүмкін. Броузерлер автоматты түрде ұяшықтар мен бүкіл кестенің өлшемдерін есептейді;

– атрибуттардың бір BORDER = кестені және бөлек ұяшықтарды қоршауға мүмкіндік береді;

– төменде келтірілген мысалда ұяшықтар аралығын кеңейтетін және биіктігі үлкен ұяшықтар құратын атрибуттар қолданылған.

HTML тілі программа – браузер терезесін бірнеше бөлікке бөліп, соңында оларды бөлек құжат етіп көрсетуге мүмкіндік береді. Мұндай аймақтар деп аталады.

Фреймдерді құру үшін құрылымы жағынан ерекшеленетін ерекше HTML құжат қолданылады. Мұндай құжатта денесі және ешқандай мәтін болмайды. Оның орнына тегтерге алынған фреймдердің баяндалуы болады: <FRAMSET> және </FRAMSET>. Бұл баяндауда аумақтардың өлшемі мен браузер терезесінде орналасу ретін, сонымен қатар осы аумақтарда жүктелу керек құжаттарды көрсетеді.

<FRAMSET> тегінде терезені бөлу тәсілін анықтайтын COLS = немесе ROWS = атрибуттары болу керек. COLS = атрибутын қолданғанда терезе аумақтарға тік сызықтармен, ал ROWS = –та көлденең сызықтармен бөлінеді. Егер екі атрибут та берілсе, онда терезеде аумақ тор пайда болады. Бұл атрибуттардың мәні терезе аумағының биіктігін (енін) анықтайды. Әрбір қатар (бағанның) параметрлерін үтір арқылы, пиксельмен немесе пайызбен (%) береді. Соңғы параметр ретінде (\*) – ны қолдануға болады.

<FRAMSET> және </FRAMSET> тегтерінің арасында құрылған аумақтардың тағайындалуын көрсететін қосымша тегтер орналасады. Ол үшін терезені қосымша бөлетін <FRAMSET> немесе бөлек аумақтарда жүктелетін құжаттарды анықтайтын жалғыз <FRAME> тегтері қолданылады. <FRAMSET> және </FRAMSET> тегтерінің арасына салынған сан құрылған аумақ санына сәйкес келу керек.

<FRAME> тегінде берілген аумақта жүктелетін құжатты анықтайтын SRC= міндетті атрибуты болу керек.

Парақтың түрлі беттерін безендіру мен өзгерту үшін қолданылатын негізгі құралдарының бірі CSS стилі болып табылады.

Стандартты HTML–да белгілі бір элементке белгілі қасиеттерді (мысалы түс, өлшем, парақта орналасуы т.б.) меншіктеу үшін ол қасиеттерді әр қолданған кезде сипаттап отыру қажет болатын. Ал CSS болса басқа тиімді әрі үнемді әдіспен әрекет етеді. Кез–келген элементке ойлаған қасиеттеріңізді меншіктеу үшін сіз оларды бір рет стиль ретінде сипаттайсыз да, ары қарай сіз

қолданғыңыз келген элементтің стилі сондай екендігін көрсетіп қана отырасыз. Сонымен қатар, сіз стильдің сипаттамасын бөлек файлда да сақтай аласыз— бұл сізге стильдің сипаттамаларын бір ғана емес бірнеше Web–парақтарда қолдануыңызға мүмкіндік береді.

Сонымен қатар CSS беттерді шрифттік безендірудің стандартты HTML–дан да жоғары деңгейде, бетті графикамен ауырлатпай–ақ жасауымызға жол береді.

### **1.4.2 MySQL деректер қорының сервері**

MySQL – кішігірім, көп ағынды деректер базасы сервері. MySQL үлкен жылдамдықты, орнықтылығымен және қолданудың жеңілдігімен сипатталады.

MySQL тілі ТсХ компаниясы өзінің жеке қажеттілігіне арнап шығарылған тілі, өте үлкен деректер базасы тез өңдеуді қамтамасыз етеді. Компанияның айтуы бойынша MySQL–ді 40–тан астам ДБ серверлерінде 1996 жылдан бері қолданады, ДБ–лар 10000 кестеден, 500–ден көбі 7 миллионнан астам жолдан тұрады. MySQL кіші және орта қосымшаларға арналған идеалды шешім болып табылады. Сервердің түпнұсқа мәтіндері көптеген платформаларда компиляциядан өтеді. Серверлік қасиеттінің көптеген мүмкіндіктері көп ағынды қолдауы бар өндірісі артық Unix–серверлерде көрінеді. Windows нұсқасында MySQL Windows NT сервисі ретінде немесе Windows 95/98 қарапайым процесс ретінде орындалады.

MySQL белгілі ашық кодты SQL мәліметтер базасы, MySQL АВ қолдануымен жасалған. MySQL АВ –MySQL мәліметтер базасымен жұмыс жасайтын коммерциялық компания. MySQL –мәліметтер базасын ұйымдастырушы жүйе, ол реляциялық мәліметтер базаларымен жұмыс жасайды. Неліктен мен жұмысымда MySQL қолдандым? Себебі, MySQL –өте тез, сенімді және қолдануға жеңіл жүйе. MySQL үлкен мәліметтер базаларына негізделіп жасалғандықтан, көптеген жылдар бойы өз қуатын көрсетуде. Қазіргі кезде ол функцияларға бай және қолдануға қарапайым мәліметтер жүйесі. Қолдану, пайдалану қауіпсіздігі жоғары болғандықтан, MySQL жүйесін Интернетте көптеп қолдануда. MySQL техникалық өзгешілігіне келсек, ол клиент-серверлік жүйе және мульти –ағымдық SQL серверін қолдайды, көптеген клиенттік программалар мен библиотекаларды, административтік құралдар мен программалық интерфейстермен жұмыс жасайды.

Мәліметтер базасы – бұл құрамында белгілі бір ақпараты бар екіөлшемді өзара байланысқан кестелер жиыны. Мәліметтер базасын басқаратын (жаңа мәліметтер қорын, кестелерді құрады және де құрылған объектілерді қолдануға мүмкіндік береді) программалық қамтама мәліметтер базасын басқару жүйесі (МББЖ) деп аталады.

Клиенттердің сұрауларын сипаттау үшін тұтас тіл құрылған – ол SQL (Structured Query Language – құрылымдық сұраулар тілі). SQLсұраулары арқылы сіз келесі әрекеттерді орындай аласыз:

- мәліметтер базасымен кестелерді құруға;

- кестелерге ақпаратты қосуға;
- ақпаратты өшіруге;
- ақпаратты модификациялауға;
- қажет ақпаратты алуға.

Әрине, admin қолданушысы бар болудан басқа сәйкес құқықтарға ие болу керек. MySQL әрбір сұрауы үтірлі нүктемен аяқталуы тиіс.

MySQL – бұл ылғи да компьютерде жұмыс істеп тұратын сервер-программа. Клиенттік программалар (мысалға, сценари) оған арнайы сұрауларды сокет (демек, желілік құралдар арқылы) механизмі арқылы жіберіп отырады, ол оларды өңдеп нәтижесін сақтайды. Осыдан кейін, қайтадан клиенттің арнайы сұрауы бойынша бүкіл нәтиже немесе оның бір бөлігі қайта жіберілед. Неге ылғи тұтас нәтиже жіберілмейді? Өте қарапайым; мәселе мәліметтер жиынының нәтижесі өте үлкен болып, оны желі бойынша тасымалдауға көп уақыт кетіп.

Сокеттерді қолдану механизмі клиент–серверлік технологиясына ойластырылған, бұл жүйеде программалардан арнайы сұрауларды қабылдап және өңдейтін арнайы программа – MySQL – сервері қосылуы керек деген сөз.

MySQL AB қолдануымен жасалған. MySQL AB – MySQL мәліметтер базасымен жұмыс жасайтын коммерциялық компания. MySQL – мәліметтер базасын ұйымдастырушы жүйе, ол реляциялық мәліметтер базаларымен жұмыс жасайды. Неліктен мен жобамда MySQL қолдандым? Себебі, MySQL – өте тез, сенімді және қолдануға жеңіл жүйе. MySQL үлкен мәліметтер базаларына негізделіп жасалғандықтан, көптеген жылдар бойы өз қуатын көрсетуде. Қазіргі кезде ол функцияларға бай және қолдануға қарапайым мәліметтер жүйесі. Қолдану, пайдалану қауіпсіздігі жоғары болғандықтан, MySQL жүйесін Интернетте көптеп қолдануда. MySQL техникалық өзгешілігіне келсек, ол клиент – серверлік жүйе және мульти – ағымдық SQL серверін қолдайды, көптеген клиенттік программалар мен библиотекаларды, административтік құралдар мен программалық интерфейстермен жұмыс жасайды.

MySQL – интернеттегі МБЖБ (мәліметтер базасын жүйелік басқару). Ол ақпараттың үлкен көлемімен жұмыс жасауға арналмаған, бірақ оның көмегі үлкен және кіші сайттарды жасау кезінде зор.

MySQL жұмыс жасау жылдамдығы жақсы, сенімді, иілгіш келеді. Онымен жұмыс жасау әдетте үлкен қиындықтарды тудырмайды. MySQL серверін қолдау автоматты түрде PHP жеткізушіне қосылады.

Бұрындары ақпаратты ұзақ сақтау үшін файлмен жұмыс жасайтынбыз, оларға бірнеше жол орналастырып әрі қарай жұмыс жасау үшін кейін алып тастайтынбыз. Web–қосымшалардың программалауында ақпаратты ұзақ сақтау тапсырмасы жиі кездеседі: кірушілер санын есептеуіште санау, хабарламаларды форумде сақтау, сайттағы ақпарат құрамымен қашықтықтан басқару т.с.с.

Файлмен жұмыстың мамандандырылған әдістері өте ауыр, көп уақыт пе жұмысты талап етеді, ақпарат енгізуге, оларды реттеуге, алуға аса назар бөлу керек. Бұл әрекеттердің барлығы Unixтің бір нұсқасы тұрған хост–провайдер серверінде жүреді. Яғни файлдарға қатынас жасау құқығы мен оларды

орналастыру жайлы ұмытпаған дұрыс. Сонда кодтар көлемі үлкейіп программа кезінде қате жасау ықтималдығы жоғары болады.

Осы мәселелердің барлығын мәліметтер базасы шешеді. Мәліметтер базасы ақпараттың қауіпсіздігі мен оның реттелуіне жауап береді, бір жолдың көмегімен ақпаратты алып, орналастыруға мүмкіндік береді. Мәліметтер базасын қолданатын код әлдеқайда ықшам келеді, әрі түзетуге мүмкіндік береді. Жылдамдық жайлы да ұмытпаған дұрыс – мәліметтер базасынан ақпаратты іріктеп алу файлдан іріктеп алғанға қарағанда жылдамырақ жүреді.

Ақпаратты сақтау үшін мәліметтер базасын (MySQL) қолданатын PHPдегі қосымша файлдарға негізделіп жасалған қосымшаларға қарағанда әрдайым жылдамырақ жұмыс жасай алады. Себебі, мәліметтер базасы C++ тілінде жасалған, ал PHPде жазылған программаны қатты дискіде мәліметтер базасынан жылдамырақ жасайтын етіп жазу мүмкін емес, өйткені PHP тілінде жазылған программалар C++ тағы программа қарағанда баяуырақ келеді, себебі PHP интерпретатор, ал C++ компилятор.

Осылайша, мәліметтер базасы қатты дискімен жасалатын жұмысты өз мойнына алып тиімді жұмыс істейді.

MySQL жүйесіне кіру үшін өзіңіздің командалық жолдар интерфейсіне келесі жолды енгізіңіз:

```
> mysql -h hostname -u username -p
```

Командалық жауап жүйенің әртүрлілігіне байланысты әртүрлі болып келеді.

MySQL командасы MySQL мониторын жүктейді. Бұл MySQL серверімен байланыс орнататын командалық жолдың клиенті:

– h кілті хостты білдіреді. Хост – MySQL сервері орнатылған компьютер. MySQL сервері орнатылған машинаға бұл кілтті жазсаң, hostname параметрін жазбауға да болады. Басқа жағдайларда hostname параметрін сервер орнатылған компьютердің атымен ауыстыру керек;

– u параметріне қолданушының атын жазу керек. Егер қолданушының аты көрсетілмесе, онда үнсіздікпен сіздің операциялық жүйеге кірген атпен беріледі.

Егер MySQL сіздің компьютеріңізде немесе серверде орнатылса, жүйеге root деген атпен кіріп, өзіңіздің мәліметтер базаңызды жасау керек. Егер орнатылу бірінші рет жүзеге асырылып отырса, онда root жүйеге кіре алатын жалғыз қолданушы болады.

Егер MySQL администратордың компьютерінде қолданылса, өзіңіздің берілген қолданушы атымен кіріңіз;

– p кілті серверге сіз парольбен қосылуды қалайтыңызды көрсетеді. Егер сіз кіріп жатқан қолданушы атына пароль керек болмаса, парольды қолданбасаңыз да болады.

Егер сіз жүйеге root атымен кіретін болсаңыз, тезірек пароль жасауыңыз дұрыс. Парольсіз root қолданушысына жүйе қауіпсіз емес.

Жолға парольді енгізу міндетті емес. MySQL оны өзі сұрайды:

```
Enter password: ****
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 9 to server version: 3.22.34-
shareware-debug
Type 'help' for help.
mysql>
```

Егер мұндай жауап шықпаса, mysql\_install\_db жолы дұрыс орындалуын тексеріңіз.

MySQL мәліметтер базасының жүйесі әртүрлі мәліметтер база түрлерін қолдауы мүмкін. Әдетте бір қосымшаға бір мәліметтер базасы қолданылады.

MySQL командалық жолына келесі жолды енгізіңіз:

```
mysql> create database dbname;
```

Dbname орнына мәліметтер базасының атын жазу керек. Жауап мынандай түрде берілуі керек:

```
Query OK, 1 row affected (0.06 sec)
```

Егер бұндай жауап берілмесе, онда жолдың синтаксисінде қате бар.

Жол соңына “;” белгісі қойылса, ол жолдың аяқталғанын және оны жүктеу керектігін білдіреді.

MySQL жүйесі көп қолданушыларға арналған, root қолданышысы қауіпсіздік мәселесі бойынша администрациялық мақсаттарда ғана қолданылуы керек. Жүйедегі әрбір қолданушының аты мен пароли болуы керек. Олар басқа парольдерден бөлек болуы тиіс. MySQL–дің ең жақсы қасиеттерінің бірі деп күрделі жеңілдіктер жүйесінде болып табылады.

Жеңілдік – қолданушының белгілі бір объектіге, белгілі бір операцияны орындау мүмкіндігі 1.2–кестеде көрсет. Бұл концепция файлдарға қол жеткізу мүмкіндіктеріне өте ұқсайды. MySQL–ге қосылған кезде қолданушыға оның не істеуге мүмкіндігі бар екендігін білдіретін рұқсат беріледі.

GRANT командасы арқылы қолданушыларды енгізуге және оларға құқық беруге болады.

```
GRANT privileges [columns]
```

```
ON item
```

```
TO user_name [IDENTIFIED BY 'password' ]
```

```
[WITH GRANT OPTION]
```

```
REVOKE – ол құқықты алып тастауға арналған команда.
```

К е с т е 1.2 – Қолданушыларға арналған жеңілдіктер

Жеңілдіктер	Қолданылу өрісі	Сипаттамасы
SELECT	Кестеге, бағанға	Жолдарды таңдауға мүмкіндік береді.

Жеңілдіктер	Қолданылу өрісі	Сипаттамасы
INSERT	Кестеге, бағанға	Жаңа жолдарды кестеге енгізуге рұқсат береді.
UPDATE	Кестеге, бағанға	Кестенің жолдарының мәндерін өзгертуге рұқсат береді.
DELETE	Кестеге	Кестенің жолдарын жоюға рұқсат береді.
INDEX	Кестеге	Кестенің белгілі бір индекстерін құруға және жоюға рұқсат береді.
ALTER	Кестеге	Қолданушыларға кестенің құрылымын өзгертуге, яғни бағандар қосуға, баған және кесте аттарын өзгертуге рұқсат береді.
CREATE	МБ, кестеге	Жаңа МБ құруға рұқсат береді.

PHP–сценарийлерін MySQL мен байланыстыру үшін пайдаланушыны күйге келтіру керек.

Көп жағдайларда сценарийлерге кесте жолдарына тек қана осы SELECT, INSERT, DELETE және UPDATE операцияларды орындау керек. Келесі жағдайлармен орындауға болады:

```
mysql> grant select, insert, delete, update
->on books, *
->to bookorama identified by bookorama123*;
```

Қауіпсіздік туралы ұмытпаңыз! Мұндай пароль ешқайда жарамайды. Егер сіз Web–хостинг қызметін қолдансаңыз, сіздер үшін жасайтын мәліметтер базасы үшін басқа да пайдаланушы жеңілдіктеріне рұқсат алуға болатыны туралы болжауға болады. Команда жолымен (кесте құру және т.б.) және Web–сценарийлерін MySQL–ге (мәліметтер базасына қосылу үшін) қосылу үшін сізге тура сол пайдаланушы атын және пароль береді. Ол қауіпсіздік апатын төмендетеді. Пайдаланушыны осындай деңгей жеңілдігімен мынадай түрде орнатуға болады:

```
mysql > grant select, insert, update, delete, index, alter,
create, drop
->on books.*
->to bookorama identified by bookorama123*.
```

Енді екінші пайдаланушыны күйге келтіруге болады.

Пайдаланушы жүйесінен Root– ты шығару.

MySQL мониторын quit теру арқылы шығуға болады. Web–тен пайдаланушы ретінде жүйеге кіру қажет.

Қажетті мәліметтер базасын жүйеде пайдаланушы деңгейінде MySQL есептік жазуында болуыңыз керек және Web-серверде сіз немесе администратор орнатқан тестілеуге дайын болуыңыз керек.

Жүйеге кіргеннен кейін алдымен сіз жұмыс істейтін мәліметтер базасын анықтау керек. Оны төмендегіні тергеннен кейін жасауға болады:

```
mysql> use dbname  
мұнда dbname-мәліметтер базасына лайықты атау.
```

use командасын термеседе болады, бірақ жүйеге кірген кезде мәліметтер базасын көрсету керек:

```
mysql dbname -h hostname -u username -p
```

Бұл мысалда books мәліметтер базасын қолданамыз:

```
mysql> use books
```

Осы команданы енгізгеннен кейін MySQL мынадай жауап қайтару керек:

```
Database changed
```

Егер мәліметтер базасының жұмысының алдында таңдалмаса, онда MySQL қате туралы хабарлаумен жауап береді:

```
ERROR 1046: No Database Selected
```

Мәліметтер базасы серверінің моделі (DBS) – кейбір реляционды мәліметтер базаларында (Informix, MySQL, Ingres, Sybase, Oracle, mSQL) жүзеге асырылған.

Оның негізін сақталынған процедуралар механизмі құрайды. Олар – SQL-сервер программалауының құралдары. Процедуралар мәліметтер базасының сөздігінде сақталады. Бірнеше клиенттер арасында бөлінеді, SQL-сервер функцияланатын компьютерде орындалады. DBS-модельдегі ұсыну компоненті компьютере-клиентте орындалады. Қолданбалы компонент сақталынған процедуралар жиынтығы ретінде сақталынған және мәліметтер базасының компьютер –серверінде функционерленеді. Осында мәліметтерге қатынас жасау компоненті орындалады, яғни мәліметтер базасының ядросы.

Ақпараттық ресурс жайлы түсінік мәліметтер базасына шейін тарылған, себебі сақталынды процедуралар механизмі – DBS-модельдің ерекше қасиеті тек қана мәліметтер базасында ғана бар.

DBS – моделдің артықшылықтары:

- қолданбалы функцияларды орталықтан ұйымдастыру мүмкіндігі;
- трафиктің төмендеуі (SQL-сұраныстармен бірге желі бойынша сақталынған процедуралар шақырылады);
- процедураларды бірнеше қосымшалар арасында бөлу мүмкіндігі;

– бір кезде жасалынған процедураны орындау жоспарының арқасында компьютер ресурстарын үнемдеу мүмкіндігі.

Тәжірибе жүзінде әдетте аралас модельдер қолданылады. Бұл кезде мәліметтер базасының тұтастылығы мен кейбір қарапайым қолданбалы функциялар сақталынды функциялармен қамтамасыз етіледі. Ал одан да күрделі функциялар тек қана компьютер – клиентте (RDA–модель) орындалатын қолданбалы программаларда жүзеге асады.

MySQL МББЖ ықшам көпағымдық сервері. MySQL жоғарғы жылдамдығымен, төзімділігімен және қолданудың жеңілдігімен ерекшеленеді.

MySQL өте үлкен деректер қорын өңдеуге арналған ішкі қажеттіліктер үшін TeX компаниясымен құрылған. Компанияның айтуы бойынша 10000 кестеден (соның ішінде 24.01.2004 кейінгі 500–ден астам кестелер 7млн. жолдан тұрады) тұратын 40 деректер қоры бар серверде MySQL–ды 1996 жылдан бері қолданып келеді.

MySQL – бұл интернеттегі ең танымал және кең тараған МББЖ (Мәліметтер Базасын басқару жүйесі). Ол үлкен ақпараттық көлемдермен жұмыс істеуге арналған, алайда оны интернет сайттарында қолдану тиімді, олардың кішігірім не үлкен болуы шарт емес.

MySQL жақсы жұмыс жылдамдығымен, сенімділігімен, иілгіштігімен ерекшеленеді. Онымен жұмыс көбіне ешқандай қиындықтарға әкелмейді. MySQL серверін қолдау PHP–дің жеткізілуіне енгізілген. Оның ақысыздығы да маңызды факторлардың бірі. MySQL GNU (GPL, GNU Public License) сияқты жалпы лицензия шарттарына сай таралады.

Бұрын біз ақпаратты көпке дейін сақтау үшін біз файлдармен жұмыс істедік: оған белгілі бір жолдар санын енгізіп, ал сонан соң келесі жұмыс ішін ашатынбыз. Ақпаратты ұзақ уақытқа сақтау есебі Web–қосымшаларды программалауда көп кездеседі: есептеуіштегі келушілерді есептеу, форумдардағы хабарландыруларды сақтау, сайтта ақпарат мазмұны ұзақтай басқару және т.б.

Алайда, файлдармен жұмыс істегендегі профессионалды әдістерді қолдану өте қомақты уақытты қажет етеді: оларға ақпараттың жазылуын қамтамасыз ету, ақпаратты сұрыптау, шығару, сонымен бірге бұл істердің барлығының хост – провайдер серверінде, ал онда Unix болу ықтималдығы өте көп, сондықтан файлдарға қол жеткізу құрықтарымен оларды орналастыру құқықтары жайлы ойлануға келдік. Сонда оның өлшемі жетерліктей ұлғаяды да программада қате жасау ықтималдығы өте жоғары.

MySQL кіші және орташа қосымша программалар үшін өте жақсы шешім болып табылады. Сервердің бастапқы кодтары көптеген платформаларда компиляцияланады. Сервердің мүмкіндіктері көпағымдылықты қолдайтын Unix–серверлерінде толығырақ ашылады, ол өнімділікті арттыруға көмектеседі.

Бүгінгі күнде MySQL әлі құрылуда, бірақ 3.22 нұсқалары толық жұмысқа жарамды. MySQL–сервер коммерциялық емес қолдануға тегін, кері жағдайда лицензия алуға тура келеді.

MySQL мүмкіндіктері.



MySQL ANSI 92 стандарттарындағы сұраныс тілі SQL–ді қолдайды және бұл стандартқа басқа МББЖ–да жоқ көптеген кеңейтілімдерді қамтиды.

MySQL–дің негізгі кемшіліктері:

- деректер қорымен шектелмеген пайдаланушылар санының жұмыс істеу мүмкіндігі;
- кестедегі жолдар саны 50 млн–ға дейін жетуі мүмкін;
- командалардың жылдам орындалуы. Мүмкін MySQL – барлық серверлер ішіндегі ең жылдам сервер;
- қарапайым және тиімді қауіпсіздік жүйесі;
- MySQL шынында да ең жылдам сервер, бірақ ол мүмкіншілікті жүзеге асыру мақсатында құраушыларға реляциялық МББЖ–не қойылатын кейбір талаптарды қарастырмауға тура келеді.

MySQL–дің кемшіліктері:

- `SELECT * FROM table1 WHERE id IN (SELECT id FROM table2)` типті ішкі сұраныстарды қолданбайды;
- транзакцияларды қолдау орындалмаған. Оның орнына `LOCK/ON LOCK TABLE` қолдануға ұсыныс берілген.

Триггерлермен сақталатын процедураларды қолдау орындалмаған.

MySQL 4.1. MySQL4.1 2004 жылыдың 27 қазанда шықты. Ол келесідей жаңа мүмкіндіктерді қамтиды:

- ішкі сұраныстар және туынды кестелер;
- кодтау және сорттаудың жаңа жүйесі;
- жылдамдығы жоғарырақ клиент – сервер протоколы;
- Microsoft Windows және GNU/Linux үшін орнату мен реттеудің жаңа программасы;
- Open SSL арқылы қорғалған клиент–сервер байланысы;
- басқа программаларда қолдануға болытын жоғарғы оптимизацияланған кітапхана;
- юникодты (UTF–8 және UCS2) толық қолдауы;
- географиялық ақпаратты сақтау үшін арналған GIS стандартты кеңістіктегі деректер типі.

MySQL 5.0. MySQL 5.0–те функционалдық кеңейтілген, ол MySQL–ді коммерциялық МББЖ–мен бір қатарға қоюға мүмкіндік береді. Егер бұрын MySQL МББЖ–н SQL стандартын қажетті көлемде қолдамайды деп есептесе, онда бұл әйгілі мәліметтер базасының бесінші нұсқасының пайда болуымен SQL стандартының толық қолдауы пайда болды. MySQL5.0 келесі жаңа мүмкіндіктерге ие:

- сақталатын процедуралар мен функциялар;
- қателерді өңдеуіш;
- курсорлар;
- триггерлер;
- көріністер;
- ақпараттық сұлба (метадеректерді қамтитын жүйелік сөздік).

## MySQL5.1.

MySQL5.1 нұсқасы SQL2003 стандартының жолын жалғастырады және келесі жаңа мүмкіндіктері бар:

- тапсырмаларды жоспарлаушы;
- сегменттеу;
- SQL 2003 стандарттарымен толығырақ сәйкес келу үшін кейбір операторлардың өзгертуі;
- MySQL Cluster жұмысында үлкен өзгерістер, мысалы, кестелік деректерді дискте сақтау мүмкіндігі;
- row-based репликациясы;
- толық мәтінді іздеу парсерінің plug-in түрінде орындалуы.

MySQL6.0. MySQL6.0 нұсқасы әлі альфа – тестілеу деңгейінде. Алғашқы 5.2 нұсқасын құру жөнінде шешім қабылданған болатын, бірақ ол жақын арада 6.0 деп өзгертілген.

Дегенмен 6.0 нұсқаның жаңа мүмкіндіктердің ең бастысы – ол Oracle компаниясымен сатып алынған InnoDB компаниясының InnoDB механизмінің орнына MySQL компанияның falcon деректерді сақтаудың жаңа механизмі.

### 1.4.3 Java Script скрипті бағдарламалау тілі

JavaScript программалау тілі Netscape фирмасының Sun Microsystems фирмасымен серіктесуімен жасалған және интерактивті HTML–құжаттарды жасауға арналған. JavaScript қолданудың негізгі аймақтары:

- динамикалық беттерді жасау, яғни, құжатты жүктегеннен кейін мазмұны өзгертін беттер;
- серверге жібермес бұрын форманы толтырудың дұрыстығын алдын ала тексеру;
- сценарий мен басқа салалардың көмегімен логикалды тапсырмаларды шешу.

JavaScript клиенттің жағында да, сервердің жағында да қосымшаларды жасауға мүмкіндік береді. Екі түрде де қосымша жасау кезінде стандартты объектілердің анықтамасы сақталған ядро қолданылады. Клиенттік қосымшалар қолданушы машинасында браузердің көмегімен орындалады.

JavaScript тілінде жазылған программа (сценарий) браузерде орнатылған интерпретатордың көмегімен өңделеді. Жазылған сценарийдің кез келген браузерде жұмыс жасауына ұмтылу керек. Оқудың алғашқы қадамында бұған қол жеткізу қиын болады.

JavaScript тілінде жазылған программа (сценарий) операторлар тізбегін құрайды. Егер де бірнеше оператор бір жолда жазылған болса, араларына нүктелі үтір белгісін қою керек. Егер де әр оператор әр жолда жазылса бөлгішті жазудың қажеті жоқ. Бір оператор бірнеше жолда кездесуі мүмкін. Құрылымданған программалау қағидалары бойынша программаны блокты

құрылымы көрінетіндей етіп жазған дұрыс. Бұл программаны зерттеу мен қате табуды жеңілдетеді.

JavaScript тіліндегі программаларда түсініктемелерді жазып кетуге болады. Белгілі бір жоға түсініктеме жазғыңыз келсе мәтіннің алдына екі қиғаш сызықты (//) қойып кету керек. Егер де түсініктеме мәтін бірнеше жолды алатын болса, символдар арасына /\* және \*/ қойып кету керек.

JavaScriptте кіші және үлкен әріптер әр түрлі символ болып саналады. Скрипттік тілдер әлемді төңкертті, солардың арқасында беттермен қалағанды жасауға мүмкіндік беретін DHTMLдің пайда болды. Әдетте екі тіл көшбасшының орнына дәмеленді: Visual Basic–а және JavaScript. Осылардың екіншісіне біз ерекше тоқтала кетпекпіз. Скрипттік тілді жекелеп өзін ғана оқу ешқандай нәтиже алып келмейді: себебі олар браузердің объектілі моделімен тығыз байланыста.

Браузерлердің алғашқы версиялары енді шыққан кездерде дисплейге ақпарат түсу шамасына байланысты келіп отырды, ешқандай өзгертуге ұшырамады. Ал қазіргі кездері беттің дисплейге жету жолы ұзағырақ:

- бет сайттан жүктеліп компьютер жадысына орналасады;
- беттің талдауы жүреді, нәтижесінде құраушылардың қатарына енеді.

Беттің құрамындағы блоктар (<body></body>, <head></head>, <p></p> т.с.с.) объектілі моделдің құрамына сәйкес уақытша мәліметтер базасына ауысады. Мәліметтер базасына басқа программалар қатынас жасай алады, әсіресе бетті экранға шығаратын рендер. Бұл мәліметтер базасының мазмұнына қатынас жасау, басқару үшін браузер бізге объектілер механизмі мен қатынасты орындауға көмектесетін скриптті тілдерді ұсынады. Әрине файлдың мазмұны өзгеруі мүмкін, бұл DHTMLдің жұмыс жасауына мүмкіндік береді. Бұл бізге бет жүктелгеннен кейін аз шамада (HTML 4.0 нұсқаларында бетке толықтай) өзгерістер енгізуге мүмкіндік береді.

Біздің не айтқымыз келгенді түсіну үшін объектілі моделдің жалпы құрылымын қарастырайық. Төменде Internet Explorerдің 4.0 жоғары версияларының объектілі моделі келтірілген:

- window – браузерге қатынас жасауға мүмкіндік беретін объект;
- frames – фреймдерге қатынас жасауға мүмкіндік беретін объект;
- document – беттен тұратын объект;
- all – құжаттардың барлық тегтерінің толық топтамасы;
- forms – формалар топтамасы;
- anchors – зәкірлер топтамасы;
- applets – апплеттер топтамасы;
- embeds – енгізілген объектілер топтамасы;
- filters – фильтрлер топтамасы;
- images – кескіндер топтамасы;
- links – сілтемелер топтамасы;
- plugins – қосылатын модульдер топтамасы;
- scripts – <script></script> блоктарының топтамасы;

- selection – белгілеулер топтамасы;
- stylesheets – дербес берілген стильдері бар объектілер топтамасы;
- history – кірілген сілтемелер тарихына қатынас жасауға мүмкіндік беретін объект;
- navigator – браузер сипаттамаларына қатынас жасауға мүмкіндік беретін объект;
- location – ағымдағы URLден тұратын объект;
- event – оқиғаларға қатынас жасауға мүмкіндік беретін объект;
- screen – экран сипаттамаларына қатынас жасауға мүмкіндік беретін объект.

Netscape Navigatorтің объектілі моделі жоғарыда аталып өткеннен айырмашылығы бар, алайда, түпкі мағыналары бір. JavaScript қолдану бізге объектілі моделдер арасындағы айырмашылықтардан құтылуға мүмкіндік береді. Әр браузер моделіне кіретін объектіні нақтылап айқындаудың орнына сәйкес объектінің мәнін, қасиеттері мен әдістерін нақты мысалдарда талқылаймыз.

Объектілі моделдің құрылымы айтарлықтай күрделі, бірақ қатаң түрде анықталған, кез келген әдіс пен қасиетке қатынас жасаудың белгілі әдісі бар. Объектілі тілдерде қолданылатын синтаксис қолданылады. Төменде орналасқан объектілер жоғарғы нүктелерден айырмашылығы бар, нақты бір қасиетке қатынас жасау үшін қатынас жасау жолын нақтылап тұрғызу керек.

Кез келген программалау тілі сияқты JavaScript те де өзінің айнымалылар, операторлар, жиынтығы, енгізілген функциялары бар. JavaScript тілін оқу осыларға негізделеді. Объектілі моделден ажыратылып алынғаның пайдасы аз.

JavaScript скриптті тіл болғандықтан web–беттерде жұмыс жасауға арналған. Сондықтан да оның коды беттің HTML–кодында жазылады. JavaScript тіліндегі код екі бөлімнен тұрады:

- қандай да бір оқиғаға бет кодынан шақырылатын функциялар;
- функция шақыратын оқиғалар.

<head> </head> тегінде функцияның сипаттамалары орналасуы керек. Функцияны шақырған уақытта ол компьютер жадысында болады. Кодты орналастыру үшін арнайы <script> тегі қолданылады. JavaScript кәдімгі сипаттамасы:

```
<script language="JavaScript">
<!--
function somefunction()
{
    // бұнда функция коды орналасады
}
//-->
</script>
```

Кодтың өзі түсініктеме тегінде орналасқан. Бұл ескі браузерлер түсініксіз бейтаныс командаларды елемеулері үшін жасалған. Жаңа программалар

түсініктемеленген скрипттердің бар екенін біледі және бұл оларға кедергі жасамайды. Түсініктемелерді жабатын тек сәл өзгеше, ол екі қиғаш сызықтармен жабылады. Сызықтар – JavaScript тіліндегі түсініктемелер, яғни программа одан кейін тұрғанның бәрін елемейді. Бұл күрделілік HTML–түсініктемелерді белгісіз команда, қателік ретінде түсінетін Netscapeпен сыйысушылық жасау үшін қолданылады.

HTML–күжаттың интерпретациясы кезінде браузер JavaScript объектілерін жасайды. JavaScript тілінің объектілері сценарийлерді жасау кезіндегі негізгі құрал болып табылады. HTML тіліндегі тегтер параметрінің мәндерімен көптеген объектілердің қасиеттері анықталады. Күжаттың құрылымы HTML–тегтерге сәйкес келетін объектілердің иерархиялық құрылымымен көрсетіледі. Барлық тегтердің атасы иерархияның ең жоғарғы бөлігінде орналасқан windows болып табылады. Ол браузер терезесін ұсынып, браузерді іске қосқанда жасалады. JavaScriptсценариіне жаңа терезе ашу үшін , жаңа күжатты көрсету үшін open әдісі қолданылады. әдісі терезені жабу үшін қолданылады. Windows объектісінің alert о әдісі параметр ретінде берілген мәтіні бар диалогты терезені көрсетеді.

Берілген әдіс енгізілген мәліметтердің дұрыстығын форманың көмегімен тексерген жағдайларда қолданылады. Windows объектісінің қасиеттері барлық терезеге қатысты болады.

Document, history, location, frame объектілері Бағынышты объектілер болып саналады. History объектісінің қасиеттері бұрын жүктелген HTML–беттерді көрсетеді. Location объектісінің қасиеті URL–адрес арқылы көрсетілген күжатпен, frame объектісі мәліметтерді көрсетудің арнайы әдісімен байланысты.

Document объектісінің қасиеті күжаттың құрамымен анықталады:

– фон түсі, кескіндер, формалар және т.с.с.;

– document объектісінде өзінің құрамына байланысты бағынышты немесе еншілес объектілер бола алады. Әдетте document объектісіне form, image, link, area т.с.с. объектілері бағынышты болып табылады.

Әбір бет үшін бір document объектісі құрылады,и оның кейбір қасиеттері <BODY>тегінің параметрлеріне сай келеді: bgColor, fgcolor, linkcolor, alinkcolor, vlinkColor.

Write және writeln әдістері күжатқа берілген парамет бойынша мәтінді жазады.

Егер күжатта кескіндер болса, <img> тегінің name параметрінде көрсетілген айнымалының көмегімен кескінді анықтайтын объектіге қатынас жасауға болады. Image объектісінде күжаттағы барлық кескіндерге сілтемелері бар images қасиеті болады. Сілтемелер нөлден бастап нөмірленген.

HTML–күжатқа кескінді орнату үшін кескінді файлдың URL–адресін анықтайтын src міндетті параметрі бар <IMG> тегі арналған. Шығарылатын кескіннің параметрлерін көрсетуге болады. Width параметрінің мәні кескіннің енін, height параметрі кескіннің биіктігін білдіреді. Кескіннің ені мен

биіктігінің мәндері кескіннің шынайы өлшемдеріне сай келмеуі мүмкін, ол кезде, кескінді жүктеуде кескін өлшемін қайта масштабтау жүреді.

jQuery – JavaScript және HTML әректесуіне бағытталған JavaScript библиотекасы. Бұл библиотека көмегімен кез келген DOM элементіне қол жеткізіп, олардың атрибуттарына және құрамына қатынасып, басқаруға болады. Сонымен қатар jQuery библиотекасы Ajax–пен жұмыс істеу үшін ыңғайлы API ұсынады.

Библиотеканың мүмкіндіктері:

- CSS–селекторлардың кроссбраузерлік қозғалтқышы;
- DOM ағашы бойынша өту;
- оқиғалар;
- бейне эффекттері;
- AJAX қосымшалары;
- JavaScript плагиндері.

jQuery библиотекасының кең мәселелер тобын шешетін функционалдығы бар. Библиотеканың әмбебап ядросының және плагиндердің ықшамды құрылысы тек қажетті қорларды қолдануға мүмкіндік береді.

#### 1.4.4 CSS стильдер тілінің негізі

Web–түйіндерінің дизайны – бұл HTML–беттерінің компоненттерінің браузердің жұмыс істеу терезесінің бетінде бір–біріне қатысты орналасуы.

Мұндағы Web–дизайн анықтамасының дәлсіздігі анық. Онда HTML–беттерінің не түсі, не формасы, не басқа компоненттерінің қасиеттері байқалмаған. Бұл анықтамада HTML – өлшеуішінің мүмкіндіктерінің шектеулі екендігін көрсетеді. Бет үстінде компоненттерді позициялау бұл HTML–дің ең әлсіз жақтарының бірі.

Бет компоненттері: текст блоктары, графика және бетке кіргізілген қосымшалар. Осы компоненттердің әрқайсысының үлкендігі мен шектері HTML–өлшеуіші шегінде түрлі дәлдікпен беріледі. HTML–де тексттік блоктардың өлшемін беру мүмкін емес. Олар браузермен, шрифтін салыстырмалы үлкендігіне қатысты есептеледі.

Автор алдын ала қолданушының браузерінің сілтемелерін білмегендіктен, ақпаратты көрудің варианттарының санын азайтады.

Браузерлерді құрастырушылар осы мәселенің шешу мүмкіндіктерін қарастырмады деп айтуға болмайды. NEXT платформасына арналған CERN браузерлерінің алғашқы версияларында және WWWW браузерлерінде авторға браузерге қойылған қасиеттерді HTML–өлшеуіші арқылы өзгерту мүмкіндігі берілген болатын, Алайда бұл әдіс коммерциялық өнімдерде жалғасын таппады да, ұмытылып кетті.

Браузер қасиеттерін өзгертудің басқа жолы – JavaScript–те программалау. Осы тілдің қарқынды дамуы HTML–беттерін көрсетуді толық басқарылу мүмкіндіктері жайында айтқызуда. JavaScript кемшілігі – өлшеуіштің

(разметканың) декларативті түрінен бас тартуы және өлшеуіш элементтерінің қасиеттерін белгілеу үшін салыстырмалы түрдегі кодтың үлкендігі.

CSS (Cascading Style Sheets) спецификациясы декларативті түріндегі өлшеуіш шегінен шықпау және де HTML–өлшеуіші элементтерінің көрсетілетін формасы жайында толық басқарушылық мүмкіндігін береді.

Каскадты кестелер стильдері бір жағынан суреттердің және қосымшалардың көлемін анықтау үшін, ал екінші жағынан текст блоктарының және оның жазылуын анықтау жайындағы мәселелерді шешуге арналған.

Компоненттердің өлшемдерінен басқа, стильдер кестелері және текстілік фрагменттің жазылу тәсілін анықтайды, осы параметрлерді текстілік блоктың ішінде өзгертеді, басқа блоктар мен бет компоненттердің өлшемдеріне сай түзетулерді жүзеге асыруға мүмкіндік береді.

Алдында айтылған барлық мүмкіндіктер CSS жайында құжаттың логикалық құрылымы мен көрсету формасын бөлу құрылғысы ретінде айтуға болады. Құжаттың логикалық құрылымы HTML–өлшеуішінің элементтерімен, ал сол элементтердің әрқайсысының көрсету формасы элементтің CSS–баяндаушысымен анықталады.

CSS өлшеуіші элементінің көру формасын толықтай қайта анықтауға мүмкіндік береді.

Web–инженеринг үшін CSS практикалық маңыздылығы (құру технологиясы мен Web–түйіндерін алып жүрудің жиынтығы) түйінді құру бағдарламасын жариялап, келесідей қадамдар түрінде көрсетуге болады.

Ең алдымен беттер номенклатурасын белгілеу керек, яғни жобаланатын Web–түйінінің барлық беттерін типтерге бөлу: мысалға, үй беттері, новигациялық беттер, ақпараттық беттер, коммуникациялық беттер және т.б. Әрбіреу үшін бұл тізім әртүрлі болуы мүмкін.

Беттер типінің әрбіреуіне белгілі бір логикалық құрылым құрылады (бет компоненттерінің стандартты жиыны).

Содан кейін түйін новигациялық картасы құрастырылып, оның беттердегі реализацияның формасы белгіленеді.

Беттің әрбір стандартты компонентасы үшін бөлек оны көрсету стилі құрастырылады (CSS–бейнелеуші).

Енді тек қана суреттерді салу, анимацияны құру, программаларды жазу, қолдан текстіні енгізу және графиканы енгізу қалды немесе беттердің мазмұнын оларға қатынасу кезінде автоматты түрде генерациялау қалды.

### **1.4.5 PHP бағдарламалау тілі**

PHP атауы – «PHP: Hypertext Preprocessor» дегенді білдіретін рекурсивті аббревиатура. Алғашында веб–парақтарды өңдеуді оңайлату үшін Perl қондырмасы ретінде құрылған болатын.

Соңғы бес жылда Интернет жылдам түрде дамып, адамдардың өзара хабарласудың жаңа әдістері пайда болды. Бұл дамудың ең алдында World Wide Web (WWW) тұр. Күнделікті осы жаңа коммуникациялық аяда мыңдаған жаңа

сайттар ашылады, ал қолданушыларға жаңа қызмет түрлері ұсынылып жатады. Нарықтың кеңінен дамуымен қатар, оларды игерген жаңа технологиялар мен өндірушілерге деген үлкен сұраныс пайда болды. Егер сіз осы жолды оқып отырған болсаңыз, онда сіз қазір web-өндірушісіз немесе жақын арада web-өндіруші боласыз. Дегенмен сіздің мамандығыңыз қандай болса да, сіз керемет жаңа PHP-технологиясы туралы естідініз.

PHP (officially «PHP: Hypertext Preprocessor») – бұл Apache WEB-сервері үшін арналған WEB-қосымшаларды өндіру жүйесі. Apache – бұл скриптік тіл, ол командалары HTML-құжатқа кіретін (server-side HTML-embedded scripting language) сервер тарапында орындалады. PHP-дің құрамында CGI-интерфейс, тіл интерпретаторы және деректер қоры мен WWW – сервердің түрлі объектілеріне кіретін рұқсат функцияларының жиынтығы. Қазіргі кезде PHP деректер қорымен байланысу қосымшаларын өңдеу мен Интернеттегі жарты функционалдық ақпараттық жүйелерді құруды қосқандағы, WEB-қосымшаларын өндірудің әрі ыңғайлы, әрі күшті (әрі кеңінен танымал) құралы болып табылады.

SSI (Server Side Include) – модуліне ұқсас түрде белгілі mime типіндегі қайнар көзді өңдеудің бөлек модулі ретінде (Server Parsed Document) WEB-серверде орнатылады. PHP client-side JavaScript сияқты PHP тілінің командаларының қойылымымен бірге HTML-құжатты береді (MS IIS серверінің ASP технологиясында жасалатыны сияқты, бірақ мұндай технологияны ертеректе кімнің ұсынғаны белгісіз).

Мысалы:

```
<html>
<head>
<title>Example</title>
</head>
<body>
<?php echo "Hi, I am a PHP script!";?>
</body>
</html>
```

Осының арқасында PHP технологиясын қолданатын WEB-қосымшаларды жасау, басқа тілдердегі (мысалға C немесе Perl – егер де Apache серверінің Mod\_Perl жүйесін назарға алмасақ) қосымшаларды жасаудан гөрі әлдеқайда оңай.

PHP тілі (Perl-ға ұқсас) жиілікті сипаттамаларды өңдеу функциясының тілінің негізгі құрылымдарынан басқа, тағы кластар кітапханасы сондай-ақ, стандартты функциялар жиыны мен осыған сәйкес түрде өндірушіге үлкен мүмкіндіктер беретін кең ыңғайлылықтан тұратын C-программалау тіліне ұқсас болып келеді.

Біздің хостингіміздің барлық тарифтері PHP4 немесе PHP5 версияларын қолдайды. Тарифтарымен сіз осында ... таныса аласыз. PHP – кеңінен қолданылатын скрипттер тілі, ол арнайы түрде WEB-программалау үшін



өңделген. HTML–ге енгізу өте оңай. PHP – бұл Web үшін арнайы шығарылған және HTML–ге енгізуге болатын ортақ мақсаты бар сценарийлер скриптингінің кеңінен жайылған тілі. PHP–ң басқа, мысалға клиенттік JavaScript–тілінен айырмашылығы – кодтың серверде орындалатыны. Егер сізде серверде орналасқан скрипт болса, онда клиент бастапқы кодтың қандай болғанын анықтау мүмкіндігінсіз, осы скрипт жұмысының нәтижесін алады. Сіз, осымен қатар, өзіңіздің web–серверіңізді, ол PHP–дің көмегімен сіздің барлық HTML–файлдарыңызды өңдеп отыратындай етіп және осының арқасында қолданушы «жеңіңіздің ішінде» не бар екенін білмейтіндей етіп конфигурациялауыңызға болады. PHP–ң ең күшті бір қасиеті – ол бастаушылар үшін өте қарапайым болып келеді де, программист–мамандарға көптеген дамыған мүмкіндіктер ұсынады. PHP мүмкіндіктерінің ұзын тізімін көре сала шошымаңыз. Сіз оны тез меңгеріп, бірнеше сағаттан кейін код жаза бастайсыз. PHP өндірісі серверлік скриптингтің айналасына көз тіккен болса да, сіз PHP көмегімен әлдеқайда көп нәрсе жасай аласыз.

PHP құралнамасының шулы жетістігі Лердорфты PHP–ді одан әрі қарай дамытуға бет бұрғызды. Өз мақсатына жету үшін, Лердорф енді Perl–дан C–ға көшуді ұйғарды. PHP құралнамасының кеңейуі PHP2.0 немесе PHP–FI (Personal Home Page – Form Interpretator) пайда болуына әкелді. 2.0 версиясының дамуына бүкіл дүние жүзінің программистері қатысты. PHP–дің жаңа версиясы кеңінен қолданылып, соның арқасында өндірушілердің негізгі тобы құралды. Олар программалық кодты тікелей HTML–ға енгізудің бастапқы концепциясын сақтап, лексикалық анализ механизмін қайтадан жазды. Осы жағдай PHP 3.0–ң пайда болуына әкеліп соқты. 3.0–версиясы пайда болғанға дейін, 1997 жылы 50000–нан астам қолданушы өз web–беттерін жақсарту үшін PHP–ді қолданған болатын. 1997 жылы PHP–дің қысқарған түрі «Personal Home Page» емес, «PHP Hypertext Processor» болу керек екені ұйғарылды. Келесі екі жыл көлемінде PHP–дің дамуы жалғаса берді. Тілге жүздеген жаңа функциялар қосыла берді, ал қолданушылардың саны жылдам өсіп келе жатты. 1999 жылдың басында Neteraft қызметі минималды бағалар бойынша, PHP қолданушыларының саны 1000000–нан асқанын хабарлады, 1999 жылы PHP4.0 версиясының шығатыны хабарланды. PHP–дің ең күшті жақтарының бірі – сценарийлерді орындау әсерлілігі болған болса да, алғашқы өндіру кезеңінде, PHP–негізінде ірі масштабты қосымшалардың құрылатыны күтілмеген еді. Осы себеппен, Zend деп аталатын лексикалық анализ механизмін жақсартуды көздеген жұмыс басталды. Жұмыс жылдам жүріп, 2000 жылдың 22 мамырында 4.0 версиясындағы PHP–дің шығуымен аяқталды. Лексикалық Zend анализаторынан басқа, Zend Technologies (Израиль) компаниясы Zend оптимизаторын жою үстінде, ол лексикалық Zend анализаторын қолданған кездегі жылдам әрекеттіліктің ұтысын жоғарылатады. Тесттер оптимизаторларды қолдану нәтижесіндегі программа жұмысының жылдамдығы 40–тан 100%–ға дейін баратынын көрсетеді.

PHP–дің болашағы өте жарқын болып көрінеді, себебі бұл өнім күрделі web–беттерде қолданылғаны сияқты, бөлек қолданушылардың компьютерлерінде де қолданылып келеді.

PHP–ді бәрінен бұрын сервердің жағында жұмыс істейтін Web–сценарилерінің іштей құрылған тілі, ол өндірушілерге динамикалық Web–қосымшаларды жылдам әрі әсерлі түрде құрастыруға мүмкіндік береді. Грамматика мен синтаксис тұрғысынан, PHP программалық C тіліне ұқсас болып келеді, дегенмен өндірушілер оған басқа да Perl, Java және C++ сияқты тілдердің құралдарын қосудан тайынбаған. Бағалы мүмкіндіктердің ішінен жиілікті сипаттарды, массивтермен жұмыс жасаудың күшті құралдары, объектілік–бағытталған методологияны және деректер қорымен жұмыс жасаудың кең қолданушылығын кездестіруге болады.

Web–бетті (яғни HTML–ді) өндірудің дәстүрлік, статикалық методологиясынан тыс қосымшаларды жазу кезінде, PHP тілі JavaScript, стильдер, WML (Wireless Markup Language) және басқа да пайдалы тілдермен бірге қолданылатын динамикалық құрамды жасау және басқарудың бағалы құралы ретінде де қолданылуы мүмкін. Жүздеген стандартты функцияларының арқасында PHP өндірушінің ойына келген дерлік кез келген мәселені шеше алады. Онда графика жасауға, олармен түрлі операцияларды орындауға, математикалық есептеулерді, электрондық сауданы және XML (Extensible Markup Language), ODBC (Open Database Connectivity) және Macromedia Shockwave сияқты кеңінен танымал технологияларды жүзеге асыруға болады. PHP–дің үлкен мүмкіндіктерінің арқасында бөтен модульдерді қосып, қиын жұмыстарды жасаудан құтылуға болады, сондықтан, көптеген программистер PHP–ді таңдайды. PHP–дің басқа артықшылығы, оның тікелей HTML – кодына енгізілетіндегінде, сондықтан программистке қарапайым HTML–шығаруы үшін көптеген командалары бар программаны жазу қажет емес. HTML және PHP кодын керек болғанынша кезек–кезек қолдануға болады.

Құрылымның ішіндегі print командасы біткен программаны береді. Мұнда инициализацияның ұзын коды да, кітапхана элементтері де жоқ, тек берілген мәселені шешетін код қана бар!

1997 жылы екі израильдік программист Энди Тутманс (Andi Gutmans) және Зив Сураски (Zeev Suraski) израильдік технологиялар институтының (Technion) екі өндірушісі кодты нольден бастап қайтадан жазды: өндірушілер Хайфта (Израиль) орналасқан Университет жобасы үшін жұмыс істеген электрондық сауда қосымшасын өндіру үшін жеткіліксіз деп тапты. PHP 3.0–те өндірушілердің PHP/FI 2.0 деректер көмегімен бірігіп жұмыс істеу үшін Энди, Расмус және Зив бірігіп, PHP 3.0–ді PHP/FI–ң ресми мұрагері ретінде жариялауды ұйғарды, ал PHP/FI–ң өндірілуі толығымен тоқтатылды. PHP 3.0–ң ең күшті жақтарының бірі ядроны кеңейту мүмкіндігі еді. Нәтижеде, кеңейтулерді жазу интерфейсі PHP–ға көптеген бөтен өндірушілердің назарын аударта қызды, бұл өндірушілердің әрқайсысы өз модулімен жұмыс істейтін–ді, сондықтан PHP үшін деректер қорының үлкен мөлшерімен, протоколдармен жұмыс істеу және API–ң үлкен мөлшерін қолдау мүмкіндіктері ашылды.

PHP 3.0 ресми түрде 1998 жылдың маусымында, жалпы тесттен өткеннен 9 айдан кейін шығарылды. 1998 жылдың қысына таман, PHP 3.0-ң ресми шығысынан кейін бірден Энди Тутманс пен Зив Сураски PHP ядросын қайта өңдей бастады. Тапсырма – күрделі қосымшалардың өнімділігін арттырып, PHP кодының базисінің модулін одан әрі қарай жақсарту. Кеңеюлер PHP 3.0-ге деректер қорларының жиынтығымен жұмыс істеп, әр түрлі API мен протоколдардың үлкен мөлшерін қолдау мүмкіндіктерін берді, бірақ PHP 3.0 модульдердің сапалы қолдауына ие болмағандықтан, қосымшалар әсерлі түрде жұмыс істей алмады. Zend Engine деп аталған (Zend Technologies-тің негізін қалаған әрі Zend Engine-ң өндірушілері Зив пен Эндидің аттарынан шыққан) жаңа қадам қойылған тапсырманы жақсы орындап келді, 1999 жылдың ортасында бірінші рет ресми түрде танылған болатын. Осы қадамда негізі қаланған және өзімен бірге функциялардың қосымша жиынтығын әкелген PHP4.0 ресми түрде 2000 жылдың мамырында шығарылды, яғни өз алдында шыққан PHP 3.0-ң шығуынан кейін 2 жылдай уақыттан кейін шықты. өнімділігінің артуы үшін, PHP 4.0 тағы да бірнеше кілттік жаңашылдықтарға ие еді, мысалға, сессияны қолдау, шығыс буферизациясы, қолданушы тарапынан енгізілетін ақпаратты өңдеудің қауіпсіз тәсілдері мен тағы бірнеше жаңа тілдің құрылымдар. PHP 4-ң жаңаруы тек 2007 жылдың соңына дейін ғана шығарылды. Сол уақытқа дейін төртінші версияны ресми қолдау орындалады. Кейін 2008 жылдың 8 тамызына дейін, қажеттілік дәрежесі бойынша тек қауіпсіздік критикалық жаңарыстары пайда болды. 9 тамыздан бастап PHP-ң төртінші версиясы тарихқа енеді. PHP-ң бесінші версиясы өндірушілер тарапынан 2004 жылдың 13 шілдесінде шығарылған болатын. Онда пайда болған өзгеріс – Zend ядросының жаңаруы, бұл интерпретатордың әсерлілігін қатты ұлғайтады. XML-белгілеу тілдерінің қолдануы енгізілген. ОББ функциялары толығымен қайта өңделіп, Java-да қолданылатын модельге ұқсас болып қалды. Деструктор, ашық, жабық және қорғалған мүшелер мен әдістер, соңғы мүшелер мен әдістер, интерфейстер мен объектілердің клондануы енгізілген. Жаңашылдықтар, дегенмен тілдің ескі версиялары да кодпен ең үлкен сәйкестілікті сақтау мақсатымен жасалынған болатын. Қазіргі кезде ең тұрақты және ең көп қолданылатыны, PHP6 dev-версия пайда болған болса да, дәл осы 5.xx-версиясы болып табылады. PHP6-версиясы 2007 жылдың 6 қазанынан бері өндірілу үстінде. Онда, қазіргі кезде, көптеген жаңашылдықтар жасалған, мысалға, жиілікті сипаттың ядросынан POSIX және «ұзын» суперглобальдік массивтерді алып тастау, safe-mode, php-magic-quotes және register-globals дерективтерін конфигурациялық php.ini-файлынан өшіріп тастау. Юникодтың жүзеге асуы үшін көптеген уақыт бөлінген. GNU/Linux/BSD және Microsoft Windows үшін компиляцияланған версияларына арналған, өндіріліп жатқан код версиясын PHP Snapshots сайтынан алуға болады.

PHP тарихы 1995 жылы тәуелсіз Расмус Лердорф (Rasmus Lerdorf) атты программист-келісуші оның онлайндық резюмесін оқыған сайтқа кіруші адамдар санын есептеуге арналған Perl/CGI сценарийін жазуынан басталады.

Оның сценарийін екі есепті шешуге бағытталған: келушінің деректерін тіркеу және web-бетіне келушілердің санын шығару. WWW дамуы әзір тек басталғанда мұндай есептерді шешу үшін арнайы құралдар болған жоқ, сонда авторды халық сұрақтармен жауып тастады. Лерддорф Personal Home Page (PHP) немесе Hypertext Processor (гипермәтінді үрдіс) атты өзінің аспабын тегін тарата бастады.

PHP-дің сәтті жарыққа шығуы Лерддорфты PHP кеңейтілуін жобалауға итермеледі. Кеңейтілудің бірі HTML формасында енгізілген мәліметтерді символдық айнымалыларға түрлендіреді, ол оларды басқа да жүйелерге экспорттауға мүмкіндік береді. Берілген мақсатқа қол жеткізу үшін Лерддорф келешектегі жобаларда Perl-дан C-ге өтуге шешім қабылдады. PHP –дегі аспаптар жиынын кеңейтілуі PHP 2.0 –нің тууына себепші болды, немесе 2.0, или PHP-FI (Personal Home Page — Form Interpretator). 2.0 нұсқасын дамытуда бүкіл дүниежүзінің программисттері ат салысқан.

PHP-дің жаңа нұсқасы танымал атаққа ие болды, және жақын арада негізгі жобалаушылар командасы құрылды. Олар программалық кодты HTML-ге ендірудің түпнұсқалық концепциясын сақтай отырып, лексикалық анализ механизмін қайта жазды, осыдан келе PHP 3.0 жарыққа шықты. 3.0 нұсқасының шығуына жуыр 1997 жылы 50 000 астам қолданушы PHP-дің өздерінің web-беттерін жақсатуға қолданылған еді.

1997 жылы PHP-дің қысқартылуы «Personal Home page» деген мағынаны емес, «PHP Hypertext Processor» ұғымын білдіруі керектігі шешілді.

PHP (ағыл. PHP: Hyper Preprocessor – «PHP: гипермәтін процессоры») – HTML – беттерді веб-серверде генерациялау үшін және мәліметтер базасымен жұмыс істеу үшін құрылған бағдарламалау тілі. Қазіргі кезде хастинг – провайдерлердің басым көпшілігін қолдайды. LAMP – веб-сайттарды құру үшін арналған «стандартты» жиынға (Linux, Apache, MySQL, PHP (Python Perl)) кіреді.

PHP құрушылардың тобы оның ядросымен PHP-ді көбейту және сыбайлас жобалармен (мысалы, PERL немесе тілдің құжаттамасы) өз еріктерімен айналысатын жұмысшылардан тұрады.

PHP-дің қолданылуы.

Желіге арналған бағдарламалау облысында PHP – әйгілі скрипттік тілдердің бірі (JSP, Perl ASP.NET) ол өзінің қарапайымдылығына, орындалу жылдамдылығына жоғарғы функционалдылығына және PHP лицензиясының негізінде таратылатын бастапқы кодтарына байланысты PHP ядроның болуымен, қосылатын модульдердің болуымен, «кеңейтілулердің» болуымен ерекшеленеді, олар мәліметтер базасымен, сокеттермен, динамикалық графиктермен, криптографиялық кітапханалармен, PDF түріндегі құжаттар және т.с.с жұмыс істеуге арналған. Кез-келген адам өзінің жеке кеңейтуін құрып, қоса алады. Жүздеген кеңейтулер бар, бірақ стандартты жиынға тек оншақты кең таралған кеңейтулер кіреді. PHP интерпретаторы веб-серверге не сол серверге арнайы құрылған модуль арқылы (мысалы, Apache немесе IIS), не CGI – қосымша программа ретінде қосылады. Оған басқа UNIX, GNU/Linux,

Microsoft Windows, MacOS X және AmigaOS операциялық жүйелерінде әкімшілік есептерді шешу үшін қолданыла алады. Бірақ бұл қасиетімен ол бірінші орынға Perl, Python және VBScript-ті жіберіп, көп тарала қойған жоқ.

Қазіргі кезде PHP-ді жүздеген құраушылар қолданады. 20 миллион сайт PHP-мен жұмыс істейтінін айтамыз, ал бұл сан Интернет домендерінің бестен бір бөлігінен асады.

PHP-ді не үшін таңдау керек?

Web-қосымша программаларын құрушыларға web-беттер – ол мәтін мен суреттерге ғана емес екенін айтудың қажеті жоқ. Жақсы сайт пайдаланушымен қандай да бір деңгейдегі интерактивтілікті қолдау керек: ақпаратты іздеу өнімді сату, конференциялар және т.б. негізінен бұл мүмкіншіліктердің барлығы Perl-да жазылған CGI-скрипттермен орындалған. Бірақ CGI-скрипттердің масштабталу мүмкіндігі өте төмен. Әрбір CGI шақыру ядродан жаңа процесстің тууын талап етеді, ал ол процессор уақытын жоғарылатады және оперативті жадыны жұмсайды. PHP басқа нұсқаны ұсынады – ол Web-сервер бөлігі ретінде жұмыс істейді, осындай мүмкіндігімен ол Microsoft-тің ASP-не ұқсас.

PHP синтаксисі Си немесе Perl синтаксисіне өте ұқсас. Бағдарламалаумен таныс адамдар программаларды PHP тілінде жазуға өте тез үйренуге мүмкіндіктері бар. Бұл тілде деректерді типтеудің арнайы бағыты жоқ және жады бөлу мен босатудың керегі жоқ.

PHP тілінде жазылған программалар жеңіл оқылады. Жазылған PHP-кодты Perl программаларына қарағанда оқу және түсіну оңай. PHP кемшіліктеріне, біріншіден PHP интерпретацияланатын болғандықтан жылдамдығы бойынша компиляцияланатын Си тілімен сәйкес бола алмайды.

Дегенмен, кіші программаларды жазған кезде, жадыға жүктеу кезінде және Си тілінде жазылған CGI-программаларды шақырғанда үстеме шығындар іске қосылады. Екіншіден, дайын модульдер қоры үлкен емес, мысалы, Perl-дегі CRAN сияқты емес. Бұл жағдайды шешу үшін тек уақыт керек. PHP 4-те қараушылар арнайы PERL репозитариін қарастырған, ол CRAN-ға ұқсас.

PhpMyAdmin PHP тілінде жазылған және MySQL МББЖ администрациялау үшін арналған веб-интерфейс түріндегі ашық кодты веб-қосымша бағдарламасы.

PhpMyAdmin браузер арқылы MySQL серверін администрациялауды ұйымдастырады, SQL командаларын орындауға және кестелер мен деректер қорының құрамын мүмкіндік береді. Бұл бағдарлама веб-қараушылардың арасында өте әйгілі, себебі ол MySQL МББЖ-ін SQL командаларын енгізбей-ақ басқаруға мүмкіндік береді.

Басқа CGI программалар сияқты да PHP көптеген қызмет атқарады: мысалы, мәліметтер жинай алу, динамикалық бет жасау, жаңалықтар т.б. Бірақ ең негізгі PHP тілінің жетістігі болып оның көлемі мен қасиеті әр түрлі мәліметтер базаларын қолдануы болып табылады.

Сонымен қатар, PHP келесі протоколдарды қолдайды: IMAP, SNMP, NNTP, POP3 және HTTP.

PHP 1994 жылы Rasmus Lerdorf ойлап тапқан. Бірінші версиялары 1995 жылы жарық көріп Personal Home Page Tools деп аталған. Ол өте қарапайым анализатор мен белгілі бір макростардан тұрған және кішкене сайтқа пайдаланарлық боларлықтай болған, мысалы, қонақ кітабы, счетчик және т.б. Анализатор қайтадан 1995 жылдың ортасында жазылып PHP/FI Version 2 деп аталған. Ол Personal Home Page ден және mSQL функцияларынан тұрған.

Статистика мәліметтері бойынша 1996 жылы PHP/FI бағдарламасын 15,000 аса дүние жүзіндегі web сайттар пайдаланған. 1997 жылдың ортасында ол сан 50,000 жеткен.

## **2 Веб–қосымшаны жобалау және өңдеу**

### **2.1 Дипломдық жұмысқа қойылған есептер**

Дипломдық жобаның мақсаты интернет технологияларын қолдана отырып «Даулет–Жаркент» кәсіпқой спорт клубын таныстыратын және де салауатты өмір салтын ұстанатын адамдар арасында күрес тақырыбында пікір алмасуға, қызықты әрі жаңа мақалалармен танысуға және машықтану бағдарламаларын қарауға мүмкіндік беретін ыңғайлы web–қосымша құру.

Құрастырушыға қойылатын есептер:

– Сайт жабасын құрастыру, қажетті ақпараттар мен технологияларды жинастыру;

– жобаның бірнеше бөлікке бөлу, деректер қорын құрастыру, тапсырмаларды мәртебесі бойынша реттеу, құрастырылатын функционалдарды жинау уақытын белгілеу;

– деректер қорын жинау, функционалдарды құрастыру, жазылған жоба бөліктерінің мүлтіксіз жұмыс жасап тұрғанына көз жеткізу;

– жазылған жоба бөлімдерін жинастыру, тестілеуден өткізу;

– дайындалған ақпараттық жүйені тапсырыс берушіге тапсыру;

– жасалған ақпараттық жүйеге толық есеп беру, сызбаларын түсіру.

Бағдарламалық өнімді өңдеу, яғни сайтты құрастыру шығармашылық процесс болып табылады және өңдеу құралдарын таңдау бұл жерде маңызды орын алады. Таңдауға байланысты сайтты өңдеу жолы тәуелді болады, мысалы PHP–ді таңдасақ, онда скрипті бағдарламалау жолын таңдағанымыз. Осыған орай сайтты өңдеу мен оның жұмыс істеу процесі анықталады.

Программалау тілі ретінде PHP тілі таңдалынды. Деректер қорын басқару жүйесі ретінде MySQL жүйесін таңдадым. MySQL деректер қоры серверлік бағдарлама ретінде, бірнеше қолданушыларға бірнеше мәліметтер қорын қолдануды қамтамасыз етеді. PHP+MySQL тандемі веб–жобалардың көпшілігінде қолданылады және жан–жақты, күрделі сайттар мен қосымша өнімдерді өңдеуге болады. Қазіргі заманда PHP жүздеген мың дамытушылармен пайдалануда. Интернет желісі сайттарының 5–тен бірі осы тілде жазылған. Бұлардан басқа HTML тілі сайт қаңқасын құрастыру үшін, CSS оны өңдеу мен дизайнның келтіру үшін және Java Script + JQuery скрипттік объектіге бағытталған бағдарламалау тілі. Сайттағы динамикалық элементтерді құрастыру мен басқару үшін қолданылады.

### **2.2 Веб–қосымшаны құру кезеңдері**

Веб–қосымша құру бірнеше кезеңдерді қамтиды:

а) Бастапқы кезең. Бастапқы кезеңнің мақсаты – жобаның дизайны мен технологиялық негізінің талаптарын анықтау, сонымен қатар ресурстың мақсаттары мен міндеттерін белгілеп, жобаны құрастырудың жоспарын бекіту.

б) Жобаның графикалық безендірудің талаптарын қамтитын «дизайнның брифи» құрастырылады.

в) Жобаның дизайны мен архитектурасын жасау. Веб–қосымшаның дизайнымен байланысты жұмыс келесі кезеңдерді қарастырады:

1) Пайдаланушы интерфейсіне қойылатын талаптар анализі.

2) Дәтериялдардың интуитивті түрде түсінікті ұйымдастырылуы мен ойластырылған өтулер.

3) Дизайн концепциясының құрастырылуы. Жобаның барлық функционалдық мүмкіншіліктерін жақсы жағынан ұсынып, мақсатты аудиторияның сұраныстарын қанағаттандыруға көмектесу.

4) Дизайнды бекіту. Жобаның бастапқы дизайн концепциясына қажетті өзгертулер енгізіп, соңғы шешімге келу. Технологиялық негіздеме мен бекітілген дизайн концепциясы бойынша мәліметтер қорының оңтайлы құрылымы жасалады. Веб–қосымшаның қызмет көрсету функциялары мен материалдарын администраторға оперативті түрде өзгертуге мүмкіндік беретін веб–порталдың әкімшілік жүйесінің интерфейсі мен механизмдерін құрастыру жұмыстары жүргізіледі. Аталған кезеңде графикалық және мәтіндік материалдарды енгізуге болатын веб– қосымшаның жұмыс нұсқасы жасалады.

г) Контентпен толтыру. Бұл кезеңде сайттың мәліметтер қоры контентпен толтырылады. Тиісті шаблондар мен механизмдер құрастырылғаннан кейін барлық мәтіндер, файлдар және қажетті көрсетілімдер енгізіліп немесе әкімшілік интерфейс арқылы мәліметтер қорына көшіріліп сайтта жарияланады.

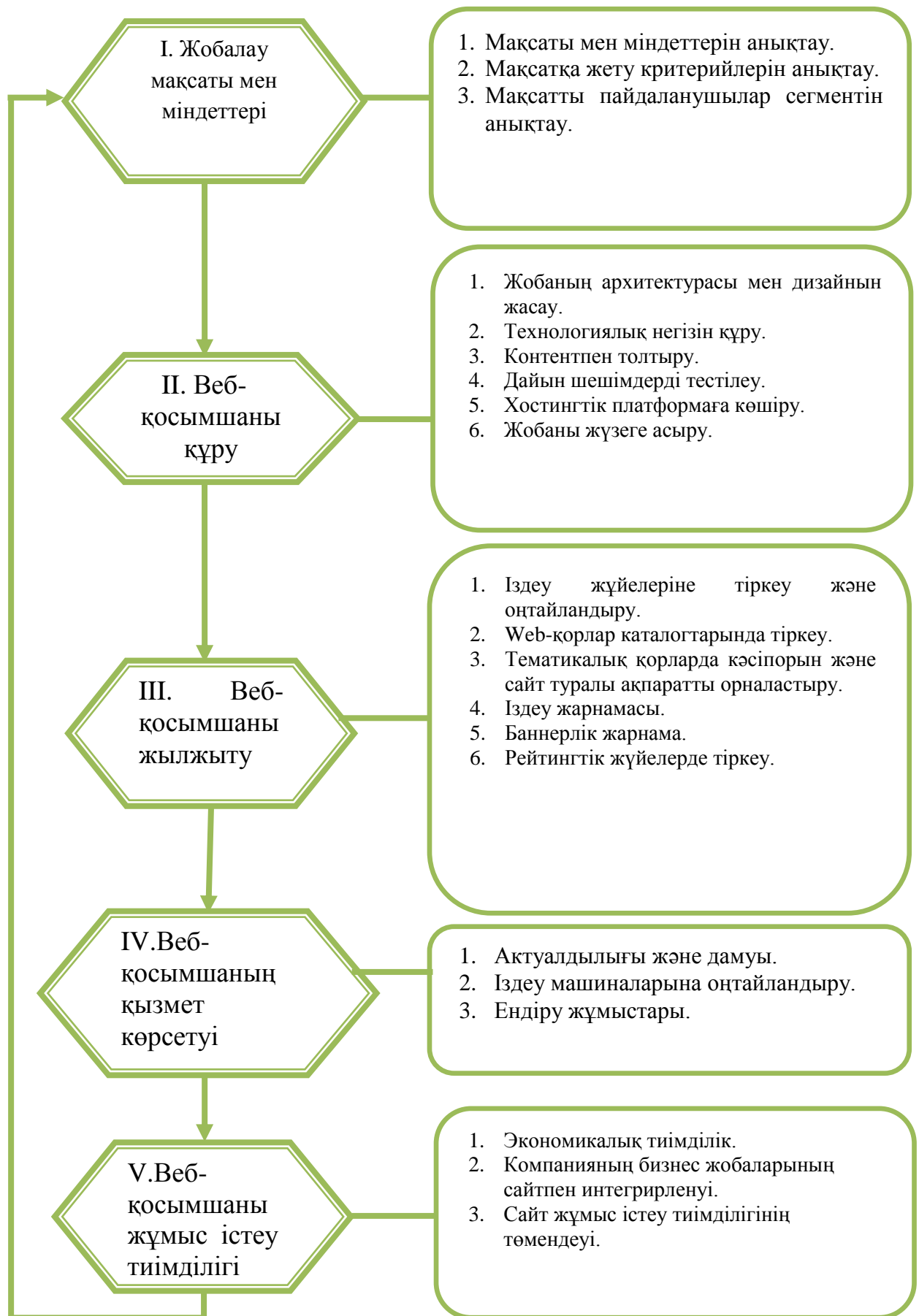
д) Дайын шешімді тестілеу. Жүйені ендіру мен интернет өкілеттіліктің құралдырының баптаулары аяқталған соң тестілеуге ұсынылады.

е) Құралдарды баптау және хостингтік платформаға көшіру. Мұнда провайдерді таңдау, хостингті алу, домендік атты таңдау және сайтты интернетке жүктеу әрекеттері жүргізіледі.

ж) Жобаны іске асыру. Барлық процедуралар аяқталғаннан кейін сайтты тестілеп, жобаны қолдау бойынша мамандардың нұсқауларынан өткеннен кейін сайт оның кірушілеріне ашылады.

Кез келген өнім сияқты веб–қосымшаның өзінің өмірлік циклының кезеңдері болады. Бұл кезеңдер 2.1 суретте көрсетілген.





2.1 сурет – Веб–қосымшаның өмірлік циклының кезеңдері

### 2.3 Веб–қосымша жұмысының алгоритмі

Сайт екі дербес бөліктен тұрады:

- қолданушылар;
- әкімшілік (администратор).

Қолданушы (қонақ, спортшы немесе администратор) сайтқа кіріп, сайт материалының навигациясымен танысады: бас мәзір навигациясы арқылы өзіне керек сілтеуді таңдайды. Таңдалған бетке өту арқылы өзіне керекті ақпаратқа қол жеткізуге мүмкіндік алады. Сайттағы ақпараттар рұқсат беру мәртебесіне қарай қорғалған. Қонақ және администратордың жеке беттері арнайы жасалған қорғаулармен шектелген. Администратор барлық ақпараттарды басқара алады және қонақтарды деректер қорына қосып, одан шығарып сала алады, қажетіне орай құпия сөздерін өзгерте алады.

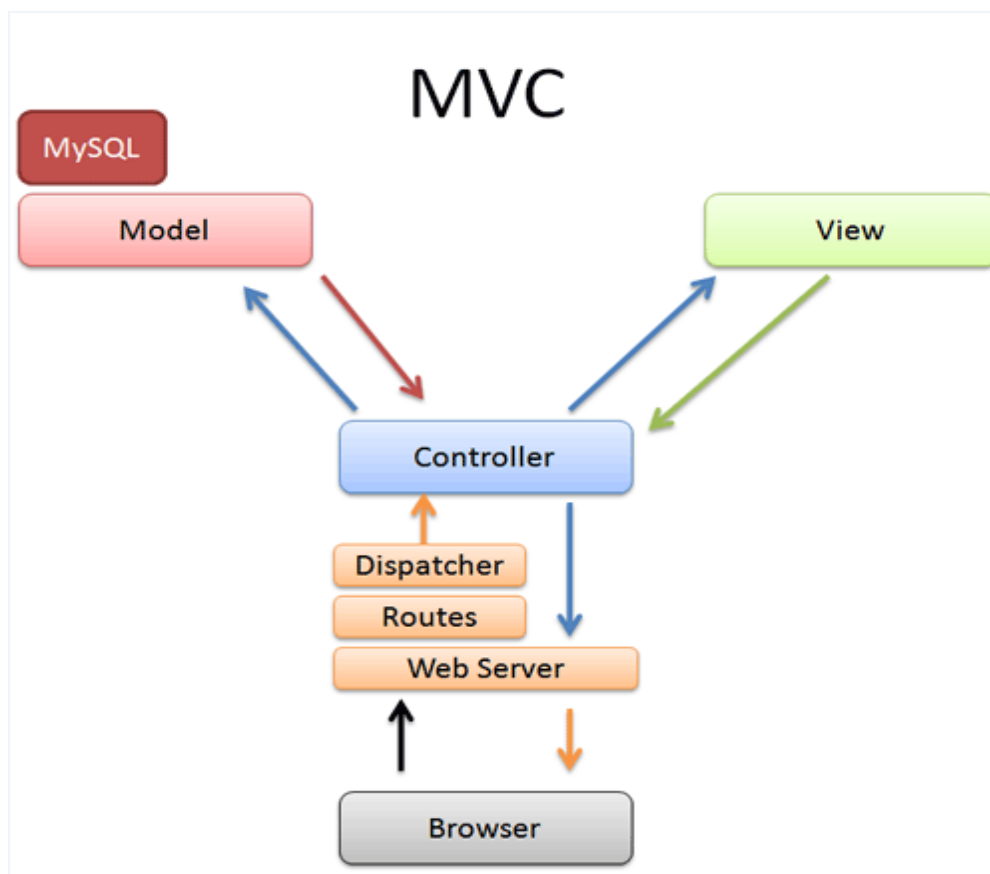
Алдында айтылғандай, сайттың кез–келген бетінен басқа бетке өту тышқанның сол жағын бір рет басу немесе оң жағынан тізімді шақырып, бетті ашу түрін таңдау арқылы жүзеге асады. Егер алгоритмді блок–схема ретінде қарайтын болсақ, онда 2.2 суретте көрсетілгендей болады.



2.2 сурет – Веб– қосымшаның жұмыс істеу алгоритмі

## 2.4 Функционалды моделді құрастыру

Бағдарламалық өнімдерді әзірлеу кезінде UML–диаграммалар мен модельдерді құрастыру маңызды болады. Оларды құрастыру өңдеушінің жұмысын жеңілдетеді. 2.3 суретте көрсетілген.



2.3 сурет– функционалды моделі

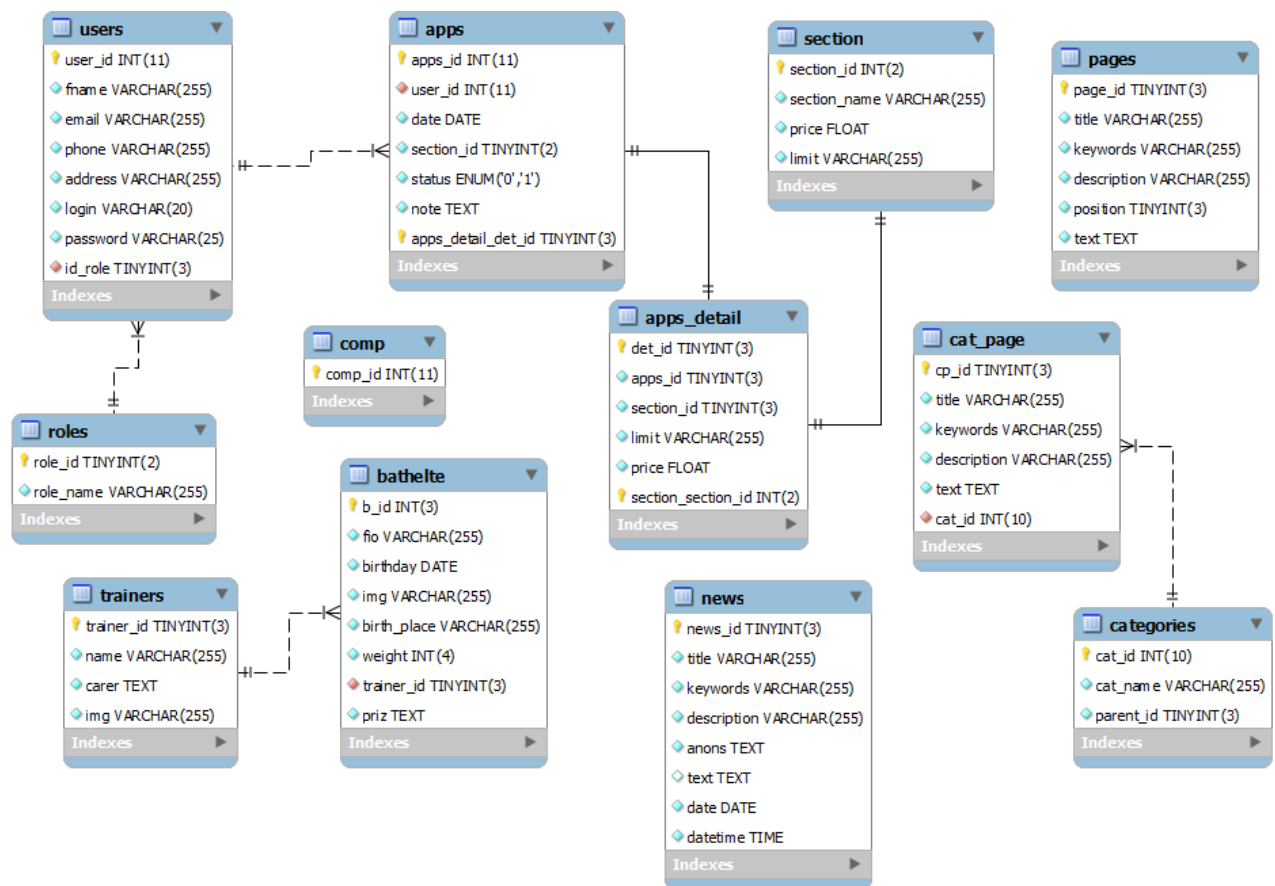
## 2.5 Сайттың бөлімдерінің сипаттамасы

Сайттың барлық беттері статикалық және динамикалық негізде жасалған. MySQL – деректер қорынан ақпараттар SQL – сұраныстар арқылы және PHP–да жазылған процедуралар арқылы шақырылып шығарылады.

Сайттың деректер қорының ER диаграммасы 2.4 суретте келтірілген.

Басты бет – бұл бетте осы сайттың қандай мақсатпен кімдер үшін құрастырылғаны туралы ақпарат енгізіледі.

Жаңалықтар – бұл бетте спортқа оның ішінде грек–рим күресіне қатысты барлық жаңалықтар енгізіледі. Бірінші бетте жаңалықтар шыққан күні бойынша реттеліп, тақырыбы және қысқаша сипаттамасымен жазылады. Тақырыбымен өткеннен кейін жаңалық толығымен ашылады. Бұл бөлімде бірнеше категориялар бойынша іздеу түрлері бар. Пайдаланушы өзіне қажетті ақпараттарды енгізіп қажетті мәліметтерді ала алады.



2.4–сурет Диаграмма

Жарыстар – бұл бетте грек–рим күресінен өтетін барлық жарыстардың уақыты, өтетін жері және де жарыстың сипаттамасы туралы барлық ақпараттар шығады.

Құрама команда – бұл бөлімде спорт клубтың барлық спортшылары туралы мәлімет болады. Олардың спортта жеткен жетістіктері туралы қысқаша сипаттамасы жазылады.

Байланысу – бұл бөлім осы сайттың иелерімен байланысуға арналған ақпараттарды көруге арналған. Бұл бөлімде байланысу аппараттарынан басқа, спорт клубтың мекен–жайы мен сол клубқа жету маршруты картадан көрсетілген.

Клуб туралы – бұл бөлімде осы клубтың басшылығы, клуб тарихы, осы уақытқа дейін жеткен нәтижелері, жаттығулардың уақыты, жұмыс жасайтын бапкерлер құрамы, қажетті құжаттар жайлы ақпараттар енгізілген.

Іздеу – батырмасы арқылы пайдаланушы сайттан барлық қажетті ақпараттарын таба алады. Егер де біздің сайттан таппаса, онда «Интернеттен іздеу» белгісін белгілеп, Google арқылы басқа қажетті ақпараттарын таба алады.

Өтініш беру– бұл жерде пайдаланушы тіркелу арқылы өтініш беріп, өзіне қажетті секциясын, жеке бапкермен қосымша дайындықты таңдап ала алады.

## 2.6 Веб– қосымша құрылымы

Сайт екі дербес бөліктен тұрады:

- қолданушылар жағы;
- әкімшілік жақ.

Қолданушы модулі 6 контроллерден құрылған:

а) main.php – 2.5 суретте көрсетілген веб сайттың басты бетін басқаратын файл. Бұл файл Controller және Model мен байланысқан. Бұл функция бастапқы бетте көрсету үшін сайттағы соңғы алты мақаланы және бейне материалды дайындап, оларды 2.5 суретте көрсетілгендей қолданушыға ұсынады.

б) жаңалықтарды дайындап, деректер қорынан сұрыптау жасауға модель файлындағы Get функциялары жауап береді.



**«Даулет-Жаркент» кәсіпқой спорт клубы**

Байланыс телефоны:  
8 707 822 15 32  
8 707 822 15 32

Жұмыс уақыты:  
Жұмыс күндері: сағ 9:00-дан 18:00-ға дейін  
Сенбі, Жексенбі демалыс күндері

Сайт бойынша іздеу?

НЕГІЗГІ    Жаңалықтар    Жарыстар    Құрама командалар    Байланыстар

**Клуб туралы**

- Басшылық
- Клуб тарихы
- Нәтижелер
- Жаттығу
- Бапкерлер құрамы
- Құжаттар

**Сайтқа кіру**

Қош келдіңіз, аға

[Шығу](#)

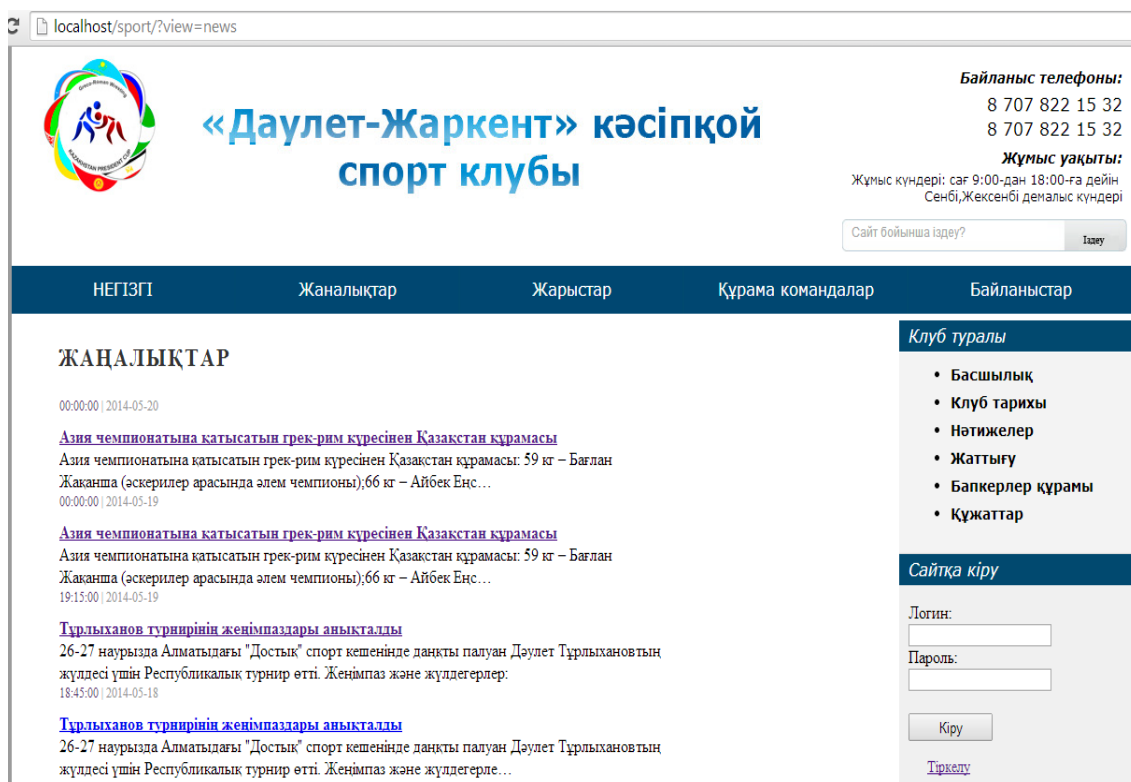
**Қабылдау**

**Өтініш беру**

ЖАҢА ЛЫҚТАР

2.5 сурет – Басты беті

News.PHP файлы–жаңалықтарды көрсетуді басқарады. Admin панелі арқылы жаңалықтарды категориялар бойынша сұрыптайды. Жаңалықтар беті 2.6 суретте көрсетілген. Құрамына get\_news,get\_news\_text функциялары кіреді.



2.6 сурет–Жаналықтар беті

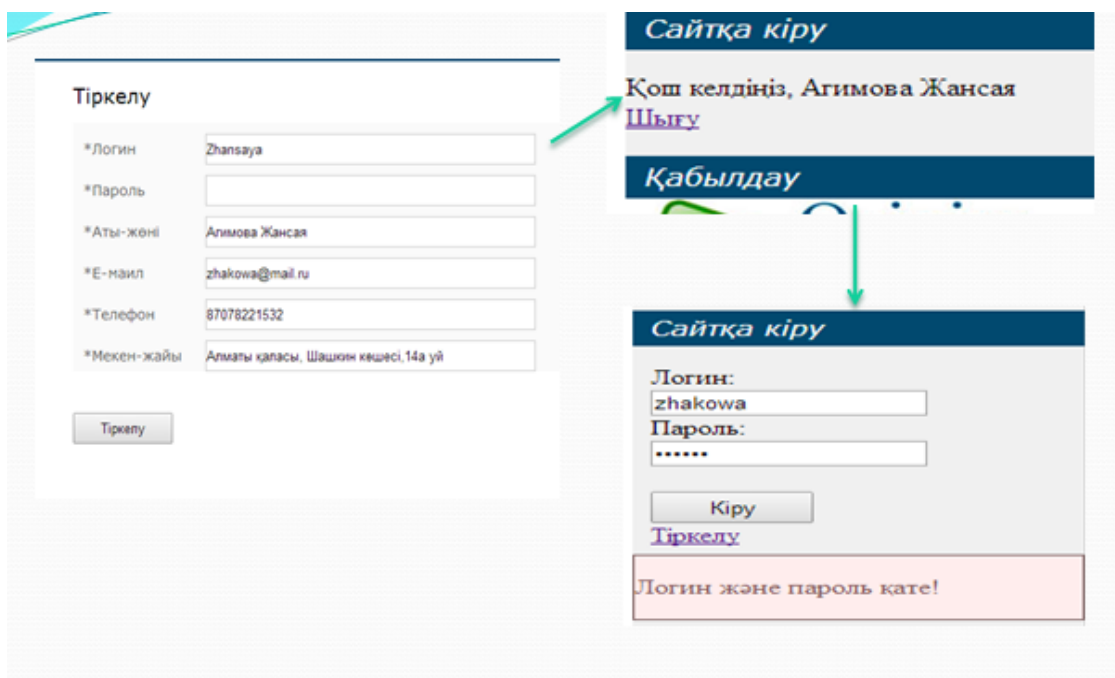
Get\_categories функциясы – категорияларды басқарады. Әр категория белгілі тақырыпқа арналған. Бұл жердегі get\_categories функциясы Categories моделі көмегімен сайттағы клуб туралы барлық категорияларды дерек қорынан алады. Көрсетілімде категориялар тізім түрінде шығарылады. 2.7 суретте клуб туралы мақалалардың категориялары көрсетілген.



2.7 сурет –Клуб туралы мақалалардың категориялары

в) Registration және authoration – сайтқа кіретін қолданушыларды авторизациялауын және рұқсат беруді басқарады. Аутентификация Zend\_Auth компоненті арқылы жүзеге асады. Бұл компонент қолданушының логин мен

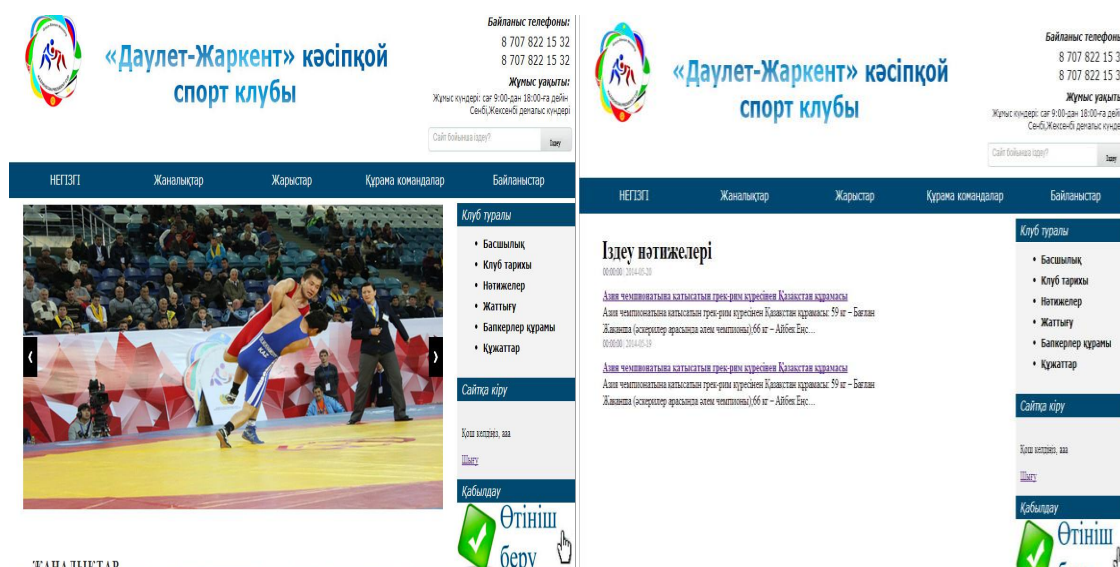
күпиясөзін қабылдап, арнайы токен жасайды. Осы токен арқылы сайт бойынша ұлықсат беру процесі өтеді. Деректер қорындағы қолданушылардың деректері get\_user моделі арқылы басқарылады. Сайтта тіркелу беті 2.8 суретте көрсетілген.



2.8 сурет– Сайтта тіркелу беті

## 2.7. Веб–порталдың функционалдығы

Сайтта пайдаланушыларға арналған жаналықтар бойынша ақпараттарды іздеу беті бар. Оның жұмыс жасау принципі 2.9 суретте көрсетілген.



2.9 сурет– Сайт бойынша іздеу бетінің нәтижесі

Сайттың басты бөлігі – өтініш беру беті. Бетте егер қолданушы тіркелген болса онда өзіне қажетті секцияны таңдап, ескертпе қалдырып, өтініш бере алады. Егер тіркелмеген болса өзінің аты–жөнін, емайлын, мекен–жайын толтырады. 2.10 суретте өтініш беру беті көрсетілген.

### Өтініш беру

**Сіз туралы ақпарат:**

*Аты-жөні	<input type="text"/>	<i>Мысалы: Сағындықов Ербол Тұрарбекұлы</i>
*Е-маил	<input type="text"/>	<i>Мысалы: test@mail.ru</i>
*Телефон	<input type="text"/>	<i>Мысалы: 8 937 999 99 99</i>
*Мекен-жайы	<input type="text"/>	<i>Мысалы: Алматы қаласы, Шашкин көшесі, 14а үй.</i>
*Секция	7 және 10 жас аралығында ▼	
Ескерту	<input style="height: 40px;" type="text"/>	<i>Мысалы: Маған жеке бапкермен қосымша дайындық керек</i>

2.10 сурет – Өтініш беру беті

Сайттың Жарыстар бетінде жарыстардың күнтізбесін көре алады. Күнтізбені өткізілу уақыты және қалалар бойынша сұрыптау мүмкіндігі қарастырылған. Жарыстар күнтізбесі беті 2.11 суретте көрсетілген.

<b>КҮНТІЗБЕ</b>		
Өткізілу уақыты	<input type="text" value="Жыл бойы (2014)"/>	Өткізілу орны
		<input type="text" value="Барлық қалалар"/>
<b>Қаңтар</b>		
<b>17 - 19</b> Қаңтар	Иван Поддубный Халықаралық турнирі	Тюмень
<b>20</b> Қаңтар	Международный турнир Голден Гран-при 30.02.-02.03 Самбатхей, Венгрия	Самбатхей
<b>17 - 19</b> Қаңтар	Иван Поддубный Халықаралық турнирі	Тюмень
<b>20</b> Қаңтар	Международный турнир Голден Гран-при 30.02.-02.03 Самбатхей, Венгрия	Самбатхей

2.11 сурет – Жарыстар күнтізбесі беті



Құрама командалар бетінде қолданушы клубтың спортшыларымен таныса алады. Бұл бетте әрбір спортшының аты–жөні, суреті, жаттықтырушысы және жеке мәліметтері келтірілген. Құрама командалар беті 2.12 суретте көрсетілген.

## Құрама командалар

### Грек-рим күресі



#### Қалмұрзаев Нұрсұлтан

Туған күні: 01 шілде 1989

Салмақ санаты: 55 кг

Жаттықтырушы: Байсейітов Б. Б.



#### Қалмұрзаев Нұрсұлтан

Туған күні: 01 шілде 1989

Салмақ санаты: 55 кг

Жаттықтырушы: Байсейітов Б. Б.

## 2.12 сурет – Құрама командалар беті

Сайтта клубта және басқада жерлерде өткізілген жарыстардың маңызды оқиғаларға байланысты суреттер галереясы бар. Ол 2.13 суретте көрсетілген.

## ФОТО

[Барлық фотосуреттер](#)

17:08 | 11 қыркүйек 2013



Олимпиада жүлдегері Ақжүрек Танатаровтың жүлдесі үшін жасөспірімдер арасындағы республикалық турнир (Мерке-2013)

17:08 | 11 қыркүйек 2013



Күрес түрлері бойынша ересектер арасындағы Қазақстан чемпионаты (Шымкент-2013)

## 2.13 сурет – Спортшылар суреті беті

## 2.7 Дизайн және беттеу

Сайттың беттеуін жасаған кезде BlueprintCSS фреймворкы қолданылған. Бұл фреймворк көмегімен парақшадағы элементтерді ыңғайлы орнатуға және реттеуге мүмкін болады. Беттеудің бұл әдісі кестелерге негізделген.

BlueprintCSS фреймворғында басты элемент «container» деп аталады. Оның ені 24 баған немесе 950px. Қалған элементтер контейнер ішіне кіріп, осы бағандарда орналасады. Бағандардағы орналасатын элементтерді жылжыту үшін сәйкес атрибутын өзгерту қажет. Бұл мақсатта BlueprintCSS фреймворкы дайын класстарды ұсынады. Мысалы «span-4» класы элементтің төрт бағанды алып тұратынын білдіреді.

Сайттағы анимацияны іске асыру үшін JavaScript скрипттік тіліне негізделген jQuery библиотекасы қолданылады. Бұл библиотека арқылы әр түрлі оқиғаларға сәйкес элементтердің атрибуттары өзгертіледі. Мысалы тінтуірді меню панеліне жақындатқан кезде меню элементінің орналасуы өзгереді. Бұл анимациясы қолданушыға бірқалыпты сырғанау ретінде көрінеді.

### 3 Техника–экономикалық негізделуі

#### 3.1 Ұсынылып отырған қызметтің сипаты мен артықшылықтары

Берілген дипломдық жобаның тақырыбы – «Даулет– Жаркент» кәсіпқой спорт клубына арналған web–қосымша құру». Салауатты өмір салты біздің игілігіміздің ең маңызды басымдықтарының бірі. Қазіргі кезде ең кең танымал ақпарат көздері веб–порталдар болып табылады, себебі кез келген қолданушы тез арада қалаған ақпаратына қол жеткізе алады және басқа адамдармен ақпарат алмасы алады. Бұл жоба салауатты өмір салтын ұстанатын қолданушыларға арналған пікір, ақпарат алмаса алатын портал болады. Бұл бөлімде берілген жобаны экономикалық жүзеге асыруды құрайтын уақытша, еңбектік және қаржы шығындары қарастырылады. Бағдарлама 1 өндіріс орнына енгізіледі деп жоспарланады, себебі бағдарлама арнайы бір ұйымға сол ұйымның қызмет түрлерін жарнамалауға және онлайн интернет арқылы төлем жасауға арналғандықтан бұл жоба сұранысқа ие болады деп есептеп, 1 өндіріс орнына сату ойластырылады. Жұмысты жазу барысында деректер қорының құрылымын және клиенттік қосымшаны жасау жұмыстары жүргізілген және енгізілген жүйеден алынған экономикалық пайданы есептеу жүргізілген. Бұл бөлімде берілген жобаны экономикалық жүзеге асыруды құрайтын уақытша, еңбектік және қаржы шығындары қарастырылады. Тұтас сегмент мен позициялау.

**Тұтас сегмент** – бұл қазіргі техникалық прогрестің мүмкіндігін толық пайдаланып жұмыс істеуді қажет ететін веб–қосымшалар жүйесі.

К е с т е 3.1 – Қызметті SWOT–талдау

<i>Артықшылықтары</i>	<i>Кемшіліктері</i>	<i>Мүмкіншілігі</i>	<i>Қауіпі</i>
Бұл жоба салауатты өмір салтын ұстанатын қолданушыларға арналған пікір, ақпарат алмаса алатын веб–қосымша болады.	Бұл веб–қосымша белгілі бір қолданушылар категориясына арналған.	Қосымша табыс көзі.	Қызмет түрінің рұқсатсыз таратылуы мүмкін.

#### 3.2 Маркетинг–микс қызметтері

Маркетинг–микс – маркетингтік құралдардың жиынтығы, бұл арқылы компания тауарлар немесе қызметтер сұранысына ықпал етуге тырысады.

Маркетинг–микс – бұл белгілі бір тауарға анықталған баға бойынша белгілі бір орында қозғалыс арқылы сұранысты қалыптастыратын жиындардың (стратегия) жүйесі.

Айта кету керек, стратегия тек компанияның белгілі бір мақсатта нарықтың кез келген сегментінде қозғалуына арналған ұмтылысы және стратегия жалпы, бөлінбейтін сипатқа ие. Жағдайды басқару мен оны назардан тыс шығармау үшін тек нақты орындалу уақыты анықталған жиындарды іске асыру керек, осы жиындарға жауапты адамдарды, қажетті ресурстарды (қаржылық, адами және т.б.) басқару қажет. іске асыру қажет. Міне сондықтан таңдалған стратегия үшін маркетинг–микс деп аталатын арнайы жиындар іске асырылады. Осы маркетингтің тактикалық деңгейі болып табылады.

Ереже бойынша, стандартты маркетинг–микс жиындарды төрт құраушы тактикалық іс бойынша сипаттайды: Маркетинг жиынтығы (marketing–mix) – бұл фирма біртұтас нарықтың оң жауабын алу үшін қолданатын басқаруға мүмкіндік беретін маркетингтік факторлардың жиыны. Маркетинг–микстің міндеті – біртұтас нарықта потенциалды тұтынушылардың қажеттерін өтеумен ғана шектелмейтін, сонымен қоса ұйымның әсерін жоғарлатуға тырысатын жиынтықтарды (mix) құрастыру.

Продукт (Product) – бұл фирманың тұтас нарыққа ұсынатын «бұйымдар мен тауарлар» жиынтығы.

Баға (Price) – тұтынушылардың тауарды алу үшін төлеуі қажет ақша суммасы.

Дистрибуция (Place) – тауар барлық бірыңғай тұтынушыларға қолжетімді болу үшін жүзеге асырылатын мүмкіндігі көп іс.

Қозғалу, даму (Promotion) – фирманың өзінің тауарлары туралы ақпарат таратуы мен тұтынушыларды осы тауарларды сатып алуға үгіттейтін мүмкіндігі зор іс–шаралар. Маркетинг–микс элементтері 3.2-кестеде көрсетілген.

К е с т е 3 . 2 – Маркетинг–микс элементтері

Маркетинг–микс элементтері	Қажетті істер мен талдаулар
Product	
“Даулет– Жаркент” кәсіпқой спорт клубына арналған web–қосымша құру	Жалпы тапсырманы шешу келесі бөлімдерден тұрады: 1) Қазіргі кезде ең кең танымал ақпарат көздері веб–порталдар болып табылады, себебі кез келген қолданушы тез арада қалаған ақпаратына қол жеткізе алады және басқа адамдармен ақпарат алмасы алады. Бұл жоба салауатты өмір салтын ұстанатын қолданушыларға арналған пікір, ақпарат алмаса алатын портал болады. 2) Егер web–қосымшаға уақыт өте келе сұраныс артатын болса, бұл қызмет түрі міндетті түрде ары қарай

Маркетинг–микс элементтері	Қажетті істер мен талдаулар
	дамытылады (яғни қызмет түрлерін таңдап, оған сұраныс тастау т.б. )
Price	
Базалық бағасы	«Даулет–Жаркент»спорт клубына арналған қызмет түрін көрсету мезгілінен бастап іске асады.
Promotion	
	Басында уақытша жарнамасыз жұмыс істеу, яғни тікелей сату
Place	
Спорт клуб	«Даулет– Жаркент» кәсіпқой спорт клубына және басқа да спорт мектептеріне арналған.

### 3.2 Бағдарламалық қамтаманың өңделуінің еңбек өнімділігінің есептелуі

Еңбекке кеткен шығынның базалық көрсеткіштері мына формула бойынша есептеледі

$$Q=q*c \quad (3.1)$$

мұндағы  $q$ – бағдарламалық өнімдегі (бастапқы команда) операторлар саны, менің жұмысымда операторлар саны 1800–ге тең, атап өткен  $q$  коэффициентінің таңдап алуға болады;

$c$ –бағдарлама күрделілігінің коэффициенті, ол менде 1 ге тең, себебі менің жобамның бағдарламалық тілі жоғарғы тілге жатады және күрделілік тобы үшінші топ, ал жаңалық дәрежесі бойынша В тобына,яғни типтік шешім қолданған бағдарламаларды дайындау тобына жатқандықтан осы мәндердің қиылысуынан арнайы кестеден алынды; Осы әдіспен базалық көрсеткішті табамыз

$$Q=1800*1=1800$$

Ары қарай бағдарламалық өнімді әзірлеуге кететін уақытты есептеу керек.Уақыт адам–сағатпен есептеледі, ал  $T_d$  нақты істелген уақытпен алынады, ал қалған кезеңнің уақыты  $Q$  командасының шартты санына байланысты есептік жолмен анықталады.

$T_{по}$  (мақсат сипатын дайындау уақыты), нақтылы деректер бойынша алынады және келесі мәнге тең деп алынады (3–тен 5 күнге дейін, 8 сағаттан)

$$T_{\text{ПО}}=24 \text{ адам/сағ}$$

$T_O$  (мақсат сипаттамасы уақыты) келесі формуламен анықталады

$$T_O=Q*B/(50*K) \quad (3.2)$$

$$T_O=1800*1,5/50*0,8=67,5 \text{ адам/сағ}$$

мұндағы  $B$  – мақсат есебі өзгерісінің коэффициенті,  $B$  коэффициенті мақсат күрделігіне және өзгеріс санына тәуелді – 1,2–ден 1,5–ке дейін;

$K$  – бағдарлама жасаушы біліктілігін ескеретін коэффициент, менің берілген салада жұмыс істеу өтілім 2 жылға дейінгі уақытты құрағандықтан,  $K$  коэффициенті мәні – 0,8–ге тең болады.

$T_A$  (алгоритм құруға кеткен уақыт) мына формуламен есептейміз

$$T_A=Q/(50*K) \quad (3.3)$$

$$T_A=1800/(50*0,8)=45 \text{ адам/сағ}$$

$T_{BC}$  (блок – сұлба құруға кеткен уақыт)  $T_A$  сияқты 3 формуламен есептеледі.

$T_H$  (бағдарламаның тілінде жазуға кеткен уақыт) келесі формуламен анықталады

$$T_H = Q*1/(50*K) \quad (3.4)$$

$$T_H=1800*1,5/(50*0,8)=67,5 \text{ адам/сағ}$$

$T_{II}$  (бағдарлама теру уақыты) келесі формуламен анықталады

$$T_{II}=Q*50 \quad (3.5)$$

$$T_{II}=Q*50=34 \text{ адам/сағ}$$

$T_{OT}$  (бағдарламаны реттеу және тестілеу уақыты) келесі формуламен анықталады

$$T_{OT}=Q*4,2/50*K \quad (3.6)$$

$$T_{OT}=1800*4,2/50*0,8=189 \text{ адам/сағ}$$

$T_D$  (құжаттарды рәсімдеу уақыты), нақтылы деректер бойынша алынады және құрылады (3–тен 5 күнге дейін, күніне 8 сағат)

$$T_D = 24 \text{ адам / сағ}$$

Еңбек шығындарының сомасы еңбек шығынының құрама сомасы ретінде формуламен есептеледі

$$T = T_{\text{ПО}} + T_{\text{ТО}} + T_A + T_{\text{БС}} + T_H + T_{\text{П}} + T_{\text{ОТ}} + T_D \quad (3.7)$$

$$T = 24 + 67,5 + 45 + 45 + 67,5 + 34 + 189 + 24 = 496 \text{ адам/сағ}$$

### 3.3 Бағдарламалық қамтаманы өңдеуге кеткен шығынның есептелуі

Еңбек құны екі құрамнан құралады: негізгі еңбек құны және қосымша еңбек құны.

Негізгі еңбек құны

$$Z_{\text{осн}} = t_{\Sigma} * TC / (t_{\text{ср}} * 8) \quad (3.8)$$

мұндағы  $t_{\Sigma}$  – (3.7) формуласы бойынша есептелетін жалпы еңбек шығыны;  
 $t_{\text{ср}}$  – айдағы орташа күн саны, 21 күнге тең, жұмыс күніндегі 8 сағатқа көбейтіледі.

Тарифтік мөлшерлеме минималды өлшемдегі еңбек ақы 01.01.2014ж бастап ҚР–да 19966 теңгені құрады

$$TC - \text{тарифтік қойылым} = 19966 / 8 = 2495,75 \text{ тең болды}$$

$t_{\Sigma}$  – (3.7) формуласы бойынша 496–ге тең.

Осылайша негізгі еңбек ақы

$$Z_{\text{осн}} = 496 * 2495,75 / 21 = 58947,24 \text{ теңге}$$

Қосымша еңбек ақы негізгі еңбек ақының 20%–ын құрайды. формуласы бойынша анықталады

$$Z_{\text{доп}} = 0,2 * Z_{\text{осн}} \quad (3.9)$$

$$Z_{\text{доп}} = 0,2 * 58947,24 = 11789,4 \text{ теңге}$$

Жалпы еңбек ақысы (еңбекті төлеу фонды) негізгі және қосымша еңбек ақының қосындысы ретінде анықталады

$$\text{ФОТ} = Z_{\text{доп}} + Z_{\text{осн}} \quad (3.10)$$

$$\text{ФОТ} = 11789,4 + 58947,24 = 70736,7 \text{ теңге}$$

### 3.4 Қосымша шығындар статьясы

Бағдарламалық қамтамасыз етудегі қосымша шығындар: материалдар мен құрамаға кеткен шығындар(құрылғының өз құны, яғни компьютерлер есепке алынбайды), социалды сақтандыруға аударым, жүкқұжатқа кеткен шығындар, амортизациялық аударым, техникалық қондырғылар қызметіне кеткен шығын, компьютермен жұмыс істеу кезінде кеткен электроэнергиялық шығындар).

Қондырғының құны бағдарламалық қамтама өңделуінің өзіндік құнына кірмегенімен, қосымша шығындар статьясындағы есептеулерде қолданылады. ЭЕМ – на бағдарлама жазу кезінде құрылғы ретінде дербес компьютер қолданылады. Құны  $C_{\text{обор}} = 189900$  теңге (дербес компьютердің қазіргі уақыттағы нарықтағы теңгемен берілген құны алынды).

Материалдар мен көмекші бөлшектер шығыны, бағдарламалық өнімді жазу барысында қолданылды ( $C_{\text{МжК}}$ ), сонымен қатар техникалық қызмет көрсету шығыны ( $C_{\text{ТО}}$ ), жабдықтың құнынан 1.5% және 2.5% (бұл мәндер бітіру жұмысының экономика бөлімін орындауға әдістемелік нұсқаулардан алынды Алматы: АЭЖБУ, 2014 – 27 б., 10–шы бет) құрайды және мына формулалар мен есептеледі (11)

$$C_{\text{МжК}} = 0,015 * C_{\text{обо}} \quad (3.11)$$

$$C_{\text{МжК}} = 0,015 * 189900 = 2848,5 \text{ теңге}$$

$$C_{\text{ТО}} = 0,025 * C_{\text{обор}} \quad (3.12)$$

$$C_{\text{ТО}} = 0,025 * 189900 = 4747,5 \text{ теңге}$$

Амортизациялық аударым өнімнің физикалық және моральдық жарамсыздығына байланысты сол өнімді толық қалпына келтіру үшін жиналатын ақша қоры болып есептеледі. Амортизациялық аударым амортизация нормасы бойынша орнатылады, формуласы бойынша есептеледі

$$A = \frac{C_{\text{обо}} * H_A * N}{100 * 12 * t} \quad (3.13)$$

мұндағы  $C_{\text{обор}}$  – компьютер құны;

$H_A$  – амортизация нормасы;

$N$  – жұмыс орындалуына кеткен күннің саны;



$t$  – дербес компьютерді қолдануға кеткен жалпы уақыт

$$N = T/24$$

$$N = 496/24 = 20,66 \text{ күн}$$

бұл жерде  $T$  еңбек шығынына тең, оның мәні – 496 адам/сағ тең болған.

Дербес компьютерде жалпы жұмыс істеу уақыты мына формуламен есептеледі

$$t = T_A + T_{BC} + T_H + T_{II} + T_{OT}$$

$$t = 45 + 45 + 67,5 + 34 + 189 = 380,5 \text{ адам/ сағ}$$

Амортизация нормасы (14) формуласы бойынша анықталады

$$H_A = \frac{C_{обор} - C_{ликв}}{T_{норм} \cdot C_{обор}} \cdot 100\% \quad (3.14)$$

мұндағы  $C_{ликв}$  – таратылым құны, жабдықтың құнынан 5% құрайды

$$C_{ликв} = 0,05 \cdot C_{обор} \quad (3.15)$$

$$C_{ликв} = 0,05 \cdot 189900 = 9495 \text{ теңге}$$

$T_{норм}$  – нормативті қызмет ету өтілі (дербескомпьютер үшін ҚР да  $T_{норм} = 4 \text{ жыл}$ ) осыдан

$$H_A = \frac{189900 - 9495}{4 \cdot 189900} \cdot 100\% = 23,75\%$$

$$A = \frac{C_{обор} \cdot H_A \cdot N}{100 \cdot 2 \cdot t} \quad (3.16)$$

$$A = 189900 \cdot 23,75 \cdot 20,66 / 100 \cdot 2 \cdot 380,5 = 204,1 \text{ теңге}$$

Бағдарламалық қамтаманы жазу кезіндегі электроэнергияның шығынын қоса есептеу керек. Электроэнергияның құны мына формула бойынша есептеледі

$$C_{ЭЭ} = M \cdot k_3 \cdot F_{эф} \cdot C_{кВт-ч} \quad (3.17)$$

мұндағы  $M$  – ЭВМ қуаты (450 Вт);  $k_3$  – жүктеу коэффициенті (0.8);  
 $C_{кВт.ч}$  – 1 кВт–сағ электрэнергияның құны, оның мәні ҚР – да 14,65  
 теңгеге тең

$F_{эф}$  – жұмыс уақытының эффективті фонды, мына формула бойынша  
 есептеледі

$$F_{эф} = D_{ном} * d * (1 - f / 100) \quad (3.18)$$

$$F_{эф} = 258 * 8 * (1 - 2,6 / 100) = 2010$$

мұндағы  $D_{ном} = 258$  – бір жылдағы жұмыс күнінің номиналды саны;

$d = 8$  – жұмыс күнінің ұзақтығы [сағ];

$f = 2.6 \%$  – ЭВМ –ды жөндеуге кеткен жоспарланған уақыт.

Берілген коэффициенттер мен параметрлер бойынша электрэнергияның  
 құны

$$C_{ЭЭ} = 0,45 * 0,8 * 2010 * 14,65 = 10601 \text{ теңге}$$

Дегенмен, алынған амортизациялық аударым мен электрэнергияға  
 шығын – жылдық шығынның мәні, оны жалпы жылдық эксплуатациондық  
 шығынмен анықталатын уақыт коэффициентімен коррекциялау (дұрыстау)  
 керек

$$\mathcal{E}_3 = t_{\Sigma'} * C_3 / F_{эф} \quad (3.19)$$

мұндағы  $C_3 = C_{ЭЭ} + C_{ТО} + A_{жыл}$  – ЭЕМ–ның жалпы жылдық эксплуатационды  
 шығын құны

$$C_3 = 10601 + 4747,5 + 204,1 = 15552,6 \text{ теңге}$$

мұндағы  $F_{эф}$  – жұмыс уақытының эффективті фонды, (3.18) формуласы  
 бойынша анықталды;

$t_{\Sigma'}$  – ЭВМ–нің формуласы бойынша анықталған есепті шешу үшін  
 жалпы қолданылған уақыты, бірақ бұл жерде тек компьютердегі жұмыс  
 уақытын ескере отырып анықталған

$$t_{\Sigma'} = t_{пр} + t_{отл} + t_d \quad (3.20)$$

$$t_{\Sigma'} = 45 + 189 + 24 = 258 \text{ сағ}$$

Сәйкесінше, ЭЕМ–ның жалпы жылдық эксплуатационды шығын құны

$$\mathcal{E}_3 = 258 * 15552,6 / 2010 = 1996,3 \text{ теңге}$$

ал уақыт коэффициенті мына формула бойынша есептеледі

$$w = \frac{\mathcal{E}_3}{C_{\mathcal{E}}} \quad (3.20)$$

$$w = \frac{1996,3}{15552,6} = 0,13$$

Осылайша, жалпы эксплуатационды шығынды уақыт коэффициентін ескеріп коррекциялаймыз(дұрыстаймыз)

Электрэнергияға шығын

$$C_{\mathcal{E}\mathcal{E}}^* = C_{\mathcal{E}\mathcal{E}} \cdot w \quad (3.21)$$

$$C_{\mathcal{E}\mathcal{E}}^* = 10601 * 0,13 = 1378,13 \text{ теңге}$$

Амортизационды аударым

$$A_{\text{жыл}}^* = A_{\text{жыл}} \cdot w \quad (3.21)$$

$$A_{\text{жыл}}^* = 204,1 * 0,13 = 26,53 \text{ теңге}$$

Сонымен қатар, формуласы бойынша есептелетін жұмыс жалақысына тәуелді шығындар бар. Осындай шығындар қатарына социалды сақтандыруға кететін аударымдар мен жүктеме аударымдар кіреді. Әлеуметтік салыққа кететін аударымдар бүкіл жалақының 11% құрайды [11%, ҚР ның Салық Кодексі], (3.22) формуласы бойынша анықталады

$$CC = (\text{ФОТ} - 0,1 * \text{ФОТ}) * 0,11 \quad (3.22)$$

$$CC = (70736,7 - 0,1 * 70736,7) * 0,11 = 7002,9 \text{ теңге}$$

Жүктеме шығындар қызмет көрсету мен басқаруға қатысты, құрылғының эксплуатациясы мен басқа да өндіріс процесін қамтамасыз ететін қосымша шығындар болып табылады, еңбек жалақысы фондының 50% құрайды, (3.23) формуласы бойынша анықталады

$$C_{\text{накл}} = 0,5 * \text{ФОТ} \quad (3.23)$$

$$C_{\text{накл}} = 0,5 * 70736,7 = 35368,35 \text{ теңге}$$

### 3.5 Өзіндік құн нәтижесінің кестесі

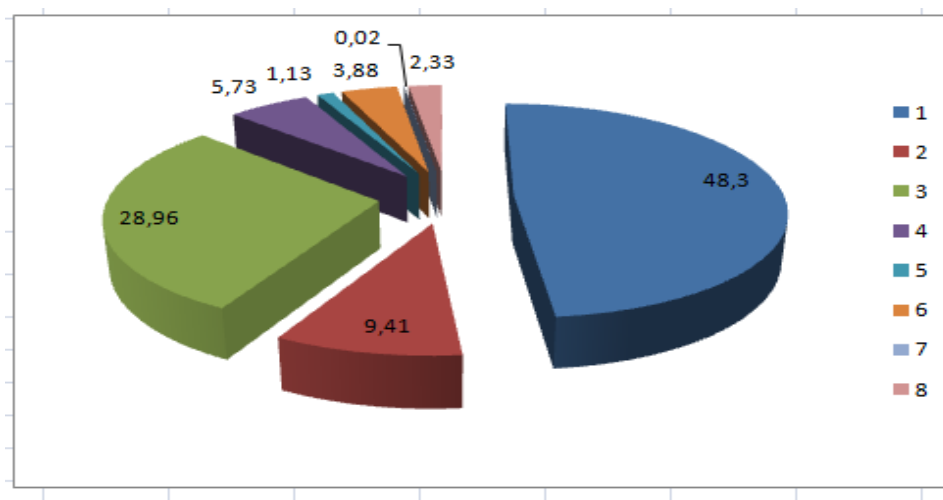
Бағдарламалық қамтаманы өңдеуге кеткен жалпы шығын еңбек жалақысы фондының, эксплуатационды шығынның, социалды сақтандыру, жүктеме шығындар мен материал және құрамалардың қосындысына тең болады.

Бағдарламалық қамтаманы өңдеудің өзіндік құны 3.3-кестеде көрсетілген.

Кесте 3.3 – Өзіндік құн нәтижесінің кестесі

Шығын статьялары		Сумма, теңге	Жалпы суммадан Процентпен
ФЗП	$Z_{\text{осн}}$	58947,24	48,3
	$Z_{\text{доп}}$	11789,4	9,41
Жүктеме шығындар, $C_{\text{накл}}$		35368,35	28,96
Әлеуметтік сақтандыру, $CC$		7002,9	5,73
Эксплуатационды Шығындар	$C_{\text{ээ}}^*$	1378,13	1,13
	$C_{\text{то}}$	4747,5	3,88
	$A_{\text{жыл}}^*$	26,53	0,02
Материалдар мен комплектілер, $C_{\text{мик}}$		2848,5	2,33
Барлығы:		122108,55	100

Өзіндік құнының диаграммасы 3.1-суретте көрсетілген



Сурет 3.1 – Өнімнің өзіндік құнының диаграммасы

Бағдарламалық өнімді жүзеге асыру бағасы оның құны мен таза кірістің қосындысынан тұрады

$$\text{Ц} = \text{C} + \text{П} \quad (3.24)$$

мұндағы C – өнім бағасы;

П – таза кіріс.

Бастапқы бағаны анықтауда бағдарламалық өнімді жүзеге асыру үшін керекті рентабелдік деңгейін анықтау қажет (20%)

$$\text{Ц}_{\text{п}} = \text{C} \cdot \left(1 + \frac{\text{P}}{100}\right) \quad (3.25)$$

мұндағы P – рентабельдік (20%)

$$\text{Ц}_{\text{п}} = 122108,55 \cdot (1 + 20/100) = 146530,26 \text{ теңге}$$

Бағдарламаның орындалу бағасы келесі формула арқылы табылады

$$\text{Ц}_{\text{р}} = \text{Ц}_{\text{п}} + \text{НДС} \quad (3.26)$$

Қазіргі таңда ҚР–да НДС 12% құрайды

$$\text{НДС} = \text{Ц}_{\text{п}} \cdot 12\% \quad (3.27)$$

$$\text{НДС} = 146530,26 \cdot 12\% = 17583,6312 \text{ теңге,}$$

$$\text{Ц}_{\text{р}} = 146530,26 + 17583,6312 = 164113,8912 \text{ теңге}$$

### **3.6 Бағдарламалық өнімді иемденудегі кәсіпорындардың бірмезгілдегі шығындарын есептеу**

Бағдарлама өнімін сатып алуға және оны өндіріске енгізу шығындары келесі шығындардан тұрады

$$\Sigma Z = C_C + C_{TP} + C_O \quad (3.28)$$

Бағдарламаның бағасы

$$C_C = 164113,8912 \text{ тг}$$

$C_{TP}$  – көлік шығыны, жүйе құнынан – 25 %, мың теңге

$$C_{TP} = 0,25 \cdot C_C \quad (3.29)$$

$$C_{\text{ТР}} = 164113,8912 * 0,25 = 41028,4728 \text{ теңге}$$

Жүйе құны үстінде есептелінді, ал қалғандары келесі түрде есептеледі. Өнімді игеруге деген маманды оқыту шығыны, оқытуға кеткен уақыт пен оған деген консалтингті фирмадағы мөлшерлемеден тұрады

$$C_{\text{O}} = T * C_{\text{ОП}} \quad (3.30)$$

$$C_{\text{O}} = 5 * 8 * 2600 = 104000 \text{ тг}$$

мұндағы  $T$  – оқытуға кеткен уақыт, сағ.

$$C_{\text{ОП}} = 2600 \text{ теңге}$$

Себебі қазіргі таңда сағатына 2500–3000 теңге аралығында.

$$\Sigma Z = C_{\text{C}} + C_{\text{ТР}} + C_{\text{O}}$$

$$\Sigma Z = 164113,8912 + 41028,4728 + 104000 = 309142,364 \text{ теңге}$$

Бағдарлама өнімін сатып алуға кеткен бір жолғы шығындар есебін 3.4–кестеде келтірілген.

**К е с т е 3 . 4** – Ақпарат жүйелерін енгізуге керекті бір жолымғы шығындар есебінің жиынтығы

Шығын бабы атауы	Сомасы, мың теңге
Жүйенің құны	164113,8912
Көлік шығыны	41028,4728
Жүйені оқуға кеткен шығыны	104000
Барлығы:	309142,364

### 3.7 Игеру саласындағы жылдық бір жолғы шығындар есебі.

Ақпараттық технологияларын қолданған кездегі жылдық шығындары келесі баптардан тұрады:

- жылдық ЕАҚ;
- әлеуметтік салық аударым;
- басқа да шығындар.

Осының барлығысын енді формула арқылы жазайық

$$C_{\Sigma} = 3\Pi + O_{\text{CH}} \quad (3.31)$$

АТ енгізгеннен кейінгі мамандардың жылдық еңбек ақы шығындары келесі формуламен есептеледі

$$ЗП = (O_C * t * K_p) * Ч * (1 + K_d) \quad (3.31)$$

$$ЗП = (3000 * 8 * 245) * 2 * (1 + 0,2) = 14112000 \text{ теңге,}$$

$$O_{CH} = (\text{ФОТ} - ЗА) * 11\% = (14112000 - 1411200) * 0,11 = 1397088 \text{ теңге,}$$

$$ЗА = \text{ЕАҚ} * 10\% = 14112000 * 0,1 = 1411200 \text{ теңге}$$

Басқа да шығындар – материалға деген шығындар, үстеме шығындар жылдық еңбек ақы шығындарынан 30 % құрайды және келесі формуламен есептеледі

$$Пр = ЗП * 0,3 \quad (3.32)$$

$$Пр = 14112000 * 0,3 = 4233600 \text{ теңге,}$$

$$Сэ = 14112000 + 1397088 + 4233600 = 19742688 \text{ теңге}$$

Ақпараттық технологияларын қолданған кездегі жылдық бір жолғы шығындарды келесі кестеге сомасын және әр баптарын көрсетіп толтырыңыз (7 кестені қара).

К е с т е 3 . 5 – Ақпараттық технологияларын қолданған кездегі жылдық бір жолғы шығындар

Шығын баптары атауы	Сомасы, мың теңге
Жылдық ЕАҚ	14112000
Әлеуметтік салық аударым	1397088
Басқа да шығындар	4233600
Барлығы:	19742688

### 3.8 Ақпараттық жүйе енгізуден үнем мен табыс мөлшерінің есебі.

АТ енгізгеннен кейінгі мамандардың жылдық еңбек ақы шығындары келесі формуламен есептеледі

$$ЗП = (O_C * t * K_p) * Ч * (1 + K_d) \quad (3.33)$$

$$ЗП = (3000 * 8 * 245) * 4 * (1 + 0,2) = 28224000 \text{ теңге}$$

$$O_{CH} = (28224000 - 2822400) * 0,11 = 2794176 \text{ теңге}$$

$$3A = 28224000 * 0,1 = 2822400 \text{ теңге}$$

Басқа да шығындар – материалға деген шығындар, үстеме шығындар жылдық еңбек ақы шығындарынан 30 % құрайды және келесі формуламен есептеледі

$$Pr = 3П \times 0,3 \quad (3.34)$$

$$Pr = 28224000 * 0,3 = 8467200 \text{ теңге}$$

$$C_3 = 42336000 + 4191264 + 12700800 = 49356720 \text{ теңге}$$

Ұйымда жүйе енгізілмей тұрғанда жылдық бір жолғы шығындар 3.6 кестеде келтірілген

**К е с т е 3 . 6** – Ұйымда жүйе енгізілмей тұрғанда жылдық бір жолғы шығындар

Шығын баптары атауы	Сомасы, мың теңге
Жылдық ЕАҚ	28224000
Әлеуметтік салық аударым	2794176
Басқа да шығындар	8467200
Барлығы:	39485376

Еңбек өнімділігі өсуінен түскен үнемділікті келтірінді шығындардың базалық ( $C_0$ ) және ұсынылған ( $C_1$ ) нұсқалар айырмасы ретінде шығарып алуға болады

$$\Delta_{yr} = C_0 - C_1 \quad (3.35)$$

$$\Delta_{yr} = 39485376 - 19742688 = 19742688 \text{ теңге}$$

**3.9 Салыстырмалы экономикалық тиімділіктің көрсеткіштерін есептеу.**

Нормативтік күрделі қаржы салымының экономикалық тиімділігінің коэффициенті келесі формуламен есептеледі

$$E_n = \frac{1}{T_n} \quad (3.36)$$

$$E_n = 0,25$$



мұндағы  $T_n$  – нормативтік күрделі қаржы салымын өтелу мерзімі, жыл.

( $T_n = 1, 2, 3 \dots n$ ) бағдарлама өнімдерінің өтеу мерзімі 4 жыл.

Есептік күрделі қаржы салымының экономикалық тиімділігінің коэффициенті

$$E_p = \frac{\mathcal{E}_{yz}}{K} \quad (3.37)$$

$$E_p = \frac{19742688}{39485376} = 0,5$$

Есептік күрделі қаржы салымын өтелу мерзімі

$$T_p = \frac{1}{E_p} \quad (3.38)$$

$$T_p = \frac{1}{0,5} = 2$$

Дисконтты есептегенде өтелу мерзімі 2 жыл.

Бағдарлама өнімін енгізудің салыстырмалы экономикалық тиімділігінің көрсеткіштері 3.7-кестеде көрсетілген.

3.7 к е с т е – Бағдарлама өнімін енгізудің салыстырмалы экономикалық тиімділігінің көрсеткіштері

Көрсеткіштер атауы	Мәні
Шартты жылдық шығынды үнемдеу, мың тенге	19742688
Күрделі қаржы салымының экономикалық тиімділігінің коэффициенті ( $E_p$ )	0.5
Күрделі қаржы салымын өтелу мерзімі ( $T_p$ ), жыл	2

Бағдарлама өнімін әзірлеуінің және енгізуінің экономикалық пайдалылығының көрсеткіштері 3.8-кестеде көрсетілген.

3.8 к е с т е – Бағдарлама өнімін әзірлеуінің және енгізуінің экономикалық пайдалылығының көрсеткіштері

Көрсеткіштер атауы	Мәні
Бағдарлама өнімін әзірлеуге және енгізуге шығын, мың теңге	39485376
Бағдарлама өнімін енгізгеннен кейінгі болжалды үнем, мың теңге	19742688
Таза дисконттық табыс, мың теңге	2102230,67
Табыстық индекс	1,05

Көрсеткіштер атауы	Мәні
Ішкі табыстық мөлшері	0,234
Дисконтталған өтелу мерзімі, жыл	2жыл 10ай
Моральдық ескіру мерзімі, жыл	3

$$T_{ок} = t + \frac{K - (d\Pi_1 + d\Pi_2)}{d\Pi_3} = 2 + \frac{39485376 - 30162440}{11425167} = 2,816 = 2\text{жыл}10\text{ай}$$

Ақшаның жылжуы 3.9-кестеде көрсетілген.

3.9 кесте – Ақшаның жылжуы

Атауы	Жылдар			
	2014	2015	2016	2017
Бірреттік шығын,кВ мың,теңге	39485376			39485376
Үнемдеуден шыққан пайда		19742688	19742688	39485376
Пайдаға салық		3948537,6	3948537,6	7897075,2
Таза пайда	- 39485376	15794150,4	15794150,4	30965348,25
Ағымдағы шығын		39485376	39485376	78970752
Дисконттау еселігі (20% мөлшерлеменен)	1	0,83	0,694	
Таза дисконтталған пайда	- 39485376	13109144,83	10961140,38	23883604,0976
Жалпы өсім бойынша ТДП	- 39485376	-26376231,17	-15415090,79	8468513,3052

Егер инвестициялық жобадан таза дисконттық табыс (ЧДД) оң болса, жоба тиімді (берілген дисконттау мөлшерінде).

$E$  – тұрақты дисконт мөлшері, 20 % [6].

Онда барлық есеп шегін ескергендегі таза дисконттық табыс сомасы

$$ЧДД = \mathcal{E}_1 \frac{1}{1+E} + \mathcal{E}_2 \frac{1}{(1+E)^2} + \mathcal{E}_3 \frac{1}{(1+E)^3} - K$$

$$ЧДД = \frac{19742688}{1,2} + \frac{19742688}{1,2^2} + \frac{19742688}{1,2^3} - 39485376 = 2102230,67$$

Таза дисконттық табыстың оң мәні  $ЧДД > 0$ , инвестицияның тиімділігін көрсетеді, ақпараттық жүйелер керекті көлемде пайда алып келеді

Табыстық индексі (ИД) келтірілген әсерлердің сомасының күрделі қаржы салымына қатынасы. Ол келесі формуламен есептеледі

$$ИД = \frac{1}{K} \sum_{t=1}^T (P_t - Z_t) \cdot \frac{1}{(1+E)^t}$$

мұндағы  $K$  – күрделі қаржы салымы немесе инвестицияның құны

$$ИД = \frac{41587606,67}{39485376} = 1,05.$$

Егер  $ИД > 1$ , жоба тиімді, егер  $ИД < 1$  – тиімді емес.

Ішкі табыстық мөлшері (ВНД немесе IRR) – келтірілген әсер, келтірілген күрделі қаржы салымдарға тең болғандағы дисконттау мөлшерін ( $r_{вн}$ ) көрсетеді.

Бұл көрсеткіш  $r_{вн}$  (ВНД) келесі теңдеуден шығады

$$\sum_{t=0}^T \frac{P_t - Z_t}{(1+r_{вн})^t} = \sum_{t=0}^T \frac{K_t}{(1+r_{вн})^t}. \quad (38)$$

Егер  $r_{вн}$  мәні инвесторға керек табыс мөлшері салынған капиталдан көп не Ішкі табыстық мөлшері (ВНД)

егер  $E_1 \rightarrow ЧДД_1 > 0$

$E_2 \rightarrow ЧДД_2 < 0$

$$ВНД = E_1 + \frac{ЧДД_1}{ЧДД_1 - ЧДД_2} \cdot (E_2 - E_1)$$

егер  $E_1 \rightarrow ЧДД_1 > 0$

$E_2 \rightarrow ЧДД_2 > 0$

$$ВНД = E_1 + \frac{ЧДД_1}{ЧДД_1 + ЧДД_2} \cdot (E_2 - E_1)$$

$E_1 = 0,20$

$$ЧДД_1 = \frac{19742688}{1,2} + \frac{19742688}{1,2^2} + \frac{19742688}{1,2^3} - 39485376 = 2102230,67$$

$E_2 = 0,25$

$$ЧДД_2 = \frac{19742688}{1,25} + \frac{19742688}{1,25^2} + \frac{19742688}{1,25^3} - 39485376 = 29446900,94 - 39485376 = -947649,024$$

$$E_1 \rightarrow ЧДД_1 > 0;$$

$$E_2 \rightarrow ЧДД_2 > 0$$

$$IRR = 0,20 + \frac{2102230,67}{2102230,67 + 947649,024} \cdot (0,25 - 0,20) = 0,234$$

Сонымен, дисконттау мөлшері келесі аралықта 20 %-23,44 %.

$$IRR = 23,44\%$$

Егер мына шарт орындалса, формула дұрыс:  $r_a < IRR < r_b$  и  $NPV_a > 0 > NPV_b$

## 4 Өміртіршілік қауіпсіздігі

### 4.1 Жасанды жарықтандыру жүйесін есептеу

Бұл дипломдық жоба «Даулет–Жаркент кәсіпқой спорт клубына арналған веб–қосымша құру» болып табылады. Жобаны жасау барысында компьютер негізгі рөл атқарады. Жоба Жаркент қаласындағы Даулет–Жаркент кәсіпқой спорт клубында қолданылады. Бөлмеде 2 оператор жұмыс істейді, жұмыс графигі – аптасына 5 күн, күніне 8 сағат.

Сондықтан, жұмысшылардың компьютермен жұмыс істеген кездегі еңбектің қауіпсіздігімен жұмысқа қабілеттілігін сақтауымыз қажет. Еңбек қорғау саласы бойынша бұл технологияға қатысты жасанды жарықтандырудың сипаттамалары мен әдістерін, түрлерін қысқаша сипаттап жазып есеп жүзінде дәлелдеуге тырыстым. Еңбек қауіпсіздігі көз– қарасы бойынша көру мүмкіндігі мен көруге арналған жайлылық адам ағзасы үшін өте–мөте маңызды болып табылады. Жақсы жарықтандырудың болмауы еңбек шартының жағдайы жақсы болмауына кепілдік береді. Сондықтан жарықты еңбек шартымызға ыңғайлы етіп пайлануымыз қажет. Жұмысшылардың еңбек жұмысы кезінде немесе құрылғылар арқылы қызмет көрсету кезіндегі қанағаттандырмайтын жарықтандыру немесе қателіктер, жарақаттану, сондай–ақ жұмыс сапасы мен еңбек өнімділігінің төмендеуіне алып келеді.

Жасанды жарықтандыру түрлері. Жасанды жарықтандыру конструктивтік жасалу бойынша үш түрге бөлінеді–жалпы, жергілікті(локальді) және біріккен. Жалпы жарықтандыру жүйесін бөлменің барлық аумағын (сварка, гальваникалық цехтар) қамтитын жұмыс барысында қолданылатын жарықтандыруды айтамыз, сондай–ақ административтік, конторлық, складтық бөлмелер жатады. Жалпы бірқалыпты жарықтандыру (жұмыс орнының орналасуына қарамастан бөлменің бүкіл ауданына бірқалыпты болып таралады) және жалпы жергілікті жарықтандыру (жұмыс орнының орналасуын ескере отырып таралатын ) деген түрге бөлінеді. Нақты көру жұмыстарымен айналысатын (мысалы , темір ұстасы), сондай–ақ құрылғылар қараңғы, көлеңкелі жерлерде орналасуына байланысты жарықтандыру ұйымдастырылуын жергілікті жарықтандыру деп атаймыз. Жергілікті және жалпы жарықтандыру қосындысын біріккен жарықтандыру деп атайды. Менің жұмысым бойынша жасанды жарықтандырудың барлық түрлерін қолдануға болады, менің ойымша біріккен жарықтандыруды қолдану керек. Ал жұмыс орнында жергілікті жарықтандыруды қолданылуына жол берілмейді, өйткені ол өткір көлеңке туғызып соның салдарынан көру қабілеті тез шаршатып өндіріс қабілеттілікті төмендетуге алып келеді.

Жұмыс мақсаты–тапсырмаға байланысты жекелеген өндіріс бөлмесінің жасанды жарықтандырылуын орындап есеп жүзінде дәлелдеу,  $E_{min}$  қажетті жарық шамдарының санын анықтап нормаланған жарықтандыруды есептеу.

Жалпы айырбастау орындарын салыстыра отырып, мынадай өлшемдер алдым.

Жұмыс аумағының ауасы, жалпы санитарлық гигиеналық талаптар бөлмедегі жұмысшылардың жұмыс істеу категориясы 1а жеңіл физикалық жұмыс категориясына жатады, өйткені құрылғыны аралықтан компьютер арқылы басқару болады. Адам организміне жұмыс категориясы бойынша энергия шығару 4.1 кестеде көрсетілген.

4.1 кесте – Адам организмінің жұмыс категориясы бойынша энергия шығыны

Жұмыс категориясы	Категория	Адам организмінің энергия шығыны, Ккал/сағ.	Жұмыс сипаттамасы
Жеңіл физикалық жұмыс 1а	1 а	138–172	Жұмыс отыру күйінде өтеді

#### 4.2 Жұмыс орнының микроклимат жағдайы ,өрт қауіпсіздігі

Жұмыс бөлмесін талдау барысында жұмыс аумағының қалыпты микроклиматтық шарттарын міндетті түрде қарастыру қажет. Қажетті микроклиматтық шарттарды сақтау үшін бөлме кондиционермен жабдықталған. Бөлменің терезелер арқылы түсетін табиғи жарықтануы, және тәуліктің қараңғы уақытында жұмыс істеу мүмкіндігін беретін жасанды жарықтануы бар. Жасанды жарықтану жоғары дәлдікті көру жұмысының I, а разрядының талаптарына сәйкес келеді. Жасанды жарықталу люминесцентті шамдар арқылы жүзеге асырылады.

Микроклиматтың күйін қадағалау операторлар залында қолайлы жағдайды орнатуға көмектеседі. Ал жұмыс орнының қолайлы жағдайларында адамдардың жұмыс істеу қабілеттері жақсарады, жұмысшылардың ауруға шалдығуы азаяды сонымен қатар, еңбек өндірісі артады. Жұмыс орнындағы нормаланған климаттық параметрлер 4.2 – кестеде келтірілген.

4.2 кесте – Температураның, ылғалдылық пен ауа қозғалысының нормалары

Жұмыс мезгілі	Ауа температурасы, °С	Салыстырмалы ауа ылғалдылығы, %	Ауа қозғалысының жылдамдығы, $\frac{м}{с}$
Суық	22 – 24	40–60	0,1
Жылы	23 – 25	30–50	0,2

Жазғы микроклиматтың қысқыға қарағанда, ауа баптауы сәл жоғарырақ. Кесте бойынша, ауа температурасы 23–25°, ылғалдылығы 30–50%, ауа қозғалысының жылдамдығы 0,2м/с. 55% дейінгі ауа ылғалдылығы құрғақ деп есептеледі, 56–70% – деңгейінде орташа құрғақ, 71–85% – орташа ылғалды және 85% жоғары – тым ылғалды. Ең үлкен ылғалдылық пен салыстырмалы ылғалдылық арасындағы айырым қанығу тапшылығы деп аталады. Қанығу тапшылығы қаншалықты үлкен болса, соншалықты ылғалды ауа жұтады. Ауаның температурасы 23–24<sup>0</sup>С – де ауа қозғалысының жылдамдығы 0,15 м/с дейін болған жөн.

Кәсіпорындық ғимараттарды жарықтанудың жоғары болуы жұмыс жағдайына дұрыс әсерін тигізеді.

Авариялық жарықтану, жалпы жарықтанудан 5%–ке аз жарықтануды қамтамасыз етуі керек, бірақ ғимарат ішінде оның деңгейі 2 лк–дан аз болмауы керек.

Эвакуациялық жарықтану, адамның қауіпті деген өтетін жерлеріне орналасады, ол өтетін жолдардың еденінде, баспалдақтарда орналасады, оның деңгейі ғимарат ішінде 0,1лк, ал ашық территорияда 0,2 лк болу керек.

4.3 кесте – Жарықтану нормасы

Жұмыс орны	Жарықтандыру беті, еденнен биіктігі	Жарықтану нормасы, лк		K <sub>п</sub> , % артық емес
		Комбинациялық жарықтану	Жалпы жарықтану	
Машиналық зал	Г – 0,8	750	400	15
ЭЕМ эксплуатациялық зал	Г – 0,8	750	400	15
Техникалық қызмет көрсету залы	Г – 0,8	750	400	15
Ақпарат тасығыш құралдарды сақтау залы	–	–	300	20

Біздің жағдайымызда электр қауіпсіздігін қамтамасыз ету үшін ГОСТ 12.1.030–81 бойынша жерлендіру мүмкіндігін қарастыру қажет. Біздің жағдайымыздағы кернеу – 220В, сондықтан жерлендіру мен нөлдеу міндеттелмейді, бірақ ұсынылады. Жерлеуді электрқауіпсіздігі мақсатымен қолданады. Ал өрт қауіпсіздігі үшін бөлмеде өрт сөндіргіш қойылған.

Бөлмедегі шуды азайту үшін шу көзін жауып, қашықтатып отыруға болады. Шуды көп бөлетін технологиялық құрылғыларды айырбастау мүмкін болмағанда дыбыс шағылғыштар, дыбыс тұтқыштар қолданады. Менің алынған жобам бойынша шу шығаратын зат ол дербес компьютерлер. Ал есептеу орталықтарының өндірістік бөлмелерінде пайда болатын шу көздерін азайтуды қоршайтын конструкцияларда дыбысты оқшаулауды арттыра отырып, қол жеткізуге болады.

### 4.3 Жұмыс орынның жасанды жарықталуы, электр қауіпсіздігі

Өндірістік кәсіпорындарындағы жасанды жарықталуының шарттары адамның көру жұмыс қабілеттігіне, физикалық және моральдық күйіне үлкен әсер етеді, ал бұл еңбек өнімділігіне және өнімнің сапасына әсер етуіне әкеледі.

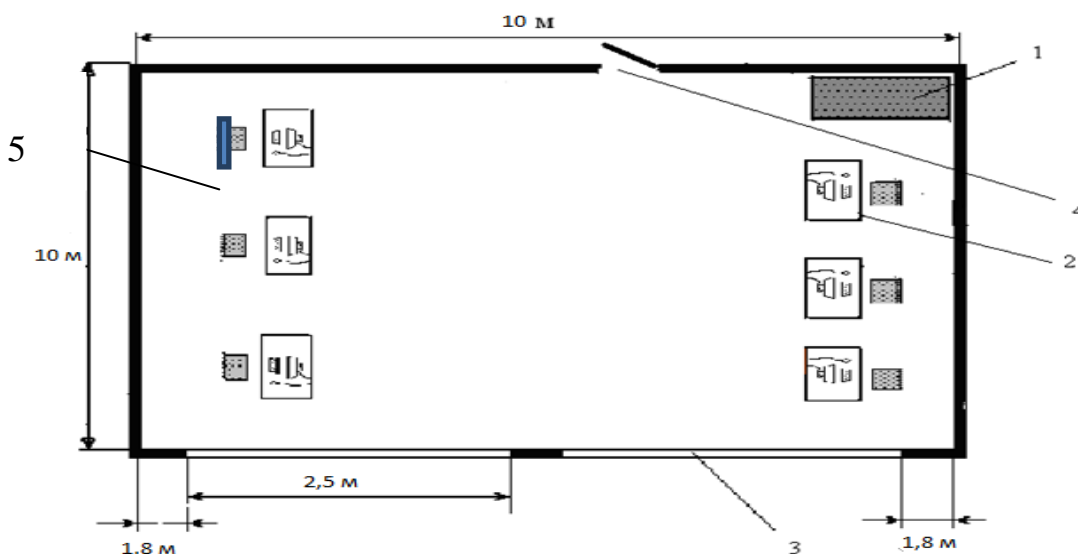
Еңбектің жағымды шарттарын құру үшін өндірістік жарықталуы келесі талаптарға сай болу керек:

- жұмыс орындағы жарықталуы тазалық нормаға сәйкес болу керек;
- жұмыс бетіндегі және қоршаған кеңістіктің шектегі жарық мүмкіндігінше біркелкі таратылу керек;
- жұмыс бетіндегі кенет көлеңкенің болуы жарықтың біркелкі емес таратылуына әкеледі, сондықтан оларды мүмкіндігінше жою керек;
- көру алаңында жарқылдық (тура немесе шағылысқан) болмау керек.

Берілген бөлмеміз барлық электр қауіпсіздігі қанағаттандырады, яғни бөлме ылғал емес 60% аспайды, тақтай еденнен жасалған. Жерге тұйықтау және нөлдеу сымдарының қимасы, олардың қосылыстары мен жалғанымдарының бүтіндігі және беріктігі тексерілген.

### 4.4 Бөлме жоспары

Оператор бөлмесінің жоспары-4.1 суретте көрсетілген



4.1 сурет – Оператор бөлмесінің жоспары



Бөлменің өлшемдері келесі: Ұзындығы  $L = 10$  м, ені  $B = 10$  м, биіктігі  $H = 3$  м. Бөлме ішінде 6 жұмыс орны және 1 шкаф, 2 терезе, 1 өрт сөндіргіш бар.

Қызмет көрсетушілердің қауіпсіздігін қамтамасыз ету үшін станцианың қызметкерлеріне әсер ететін барлық мүмкін факторларды талдау қажет. Жұмыс бөлмесінің бір жақтық екі терезе ретіндегі әр қайсысы 1,1м табиғи жарықтандыруы бар.

Қаралып жатқан бөлмеде жұмыс істеуге қажетті жарықтандыру қалып бойынша  $E_n=200$ лк, осылайша бөлмедегі қажетті жарықтандыруды қамтамасыз етеміз.

Бөлмеде құрылғылардың жұмысын жүзеге асыру үшін 2 дербес компьютер қолданылады.

Құрылғының техникалық көрсеткіштері:

- дербес компьютер Intel Core i5–3317U Processor E7500 2.60 GHz (SIS751+SB+VGA)/2GB DDR II/160 Gb SATA Seagate/SP/CD 400W;
- монитор 17" LG LI720 0.26dpi;
- электрқоректенуі – айнымалы тоқ 220–250 В, жиілігі 60 Гц, қуаты 400 Вт.

#### 4.5 Жарықтың көзін таңдау

Жасанды жарық үшін қызу және газоразрядтық шамдар қолданылады. Жалпы жұмыстық қыздыру шамдардың типтерін белгілеу шартындағы келесі әріптер мынаны білдіреді: В– вакуумдық, Г– газбен толтырылған, Б– биспиральдық, БК– биспиральдық криптондық. Шамдардың жарықтық ағымдары, жалпы жұмыстық қыздырушылық және олардың қуаты 4.3 кестеде келтірілген.

4.3 кесте– қыздырушылық шамның қуаты мен жарықтық ағым

Шамның типі	Қуат Вт	220	200–235	Шамның типі	Қуат Вт	200	200–235
		кернеуіндегі жарықтық ағым				кернеуіндегі жарықтық ағым	
В	15	105	85	Б	150	2100	1840
В	25	220	190	Г	200	2800	–
Б	40	400	300	Б	200	2920	2540
БК	40	460	–	Г	300	4600	4000
Б	60	716	550	Г	500	8300	7200
БК	60	790	–	Г	750	13100	–
Б	100	1350	1090	Г	1000	18600	–
БК	100	1450	–	Г	1500	2900	–
Г	150	2000	–				

Кіші қысымды газоразрядтық шамдар ең үнемді болып саналады (ЛД, ЛДЦ, ЛХБ және т.б.). Газоразрядтық шамдардың кейбір техникалық сипаттамалары 4.4 кестеде келтірілген.

4.4 кесте – кіші қысымды газоразрядтық шамдардың кейбір сипаттамалары

Номиналдық қуат, Вт	НОМИНАЛДЫ ЖАРЫҚТЫҚ АҒЫМ ЛМ., ШАМДАР ТИПІ					Шамның өлшемі, мм.	
	ЛДЦ	ЛД	ЛХБ	ЛТБ	ДБ	Диаметр	Штырьдың ұзындығы
15	500	590	675	700	760	27	451,6
20	820	920	935	975	1060	40	604,0
30	1450	1640	1720	1720	2100	27	908,8
40	2100	2340	3000	3000	3120	40	1213,6
65	3050	3570	3820	3980	4650	40	1514,2
80	3740	4070	4440	4440	5220	40	1514,2

#### 4.6 Жасанды жарықтандыруды есептеу

Жарықтандыру табиғи және жасанды болуы мүмкін. Жасанды жарықтандыру екі жүйелі болуы мүмкін: жалпы және аралас. Аралас жарықтандыру кезінде жалпыға жергілікті жарықтандыру қосылады.

Жасанды жарықтандыруды есептеу үшін пайдалану коэффициенті және нүктелік әдістер қолданылады. Пайдалану коэффициенті әдісі үлкен көлеңке түсіретін заттар болмаған жағдайдағы біркелкі горизонталь беттердегі жарықтандыруды есептеуге арналған. Бұл әдіс арқылы  $\eta$  коэффициентінің мәні анықталады. Ол есептелінетін бетке түсетін жарық ағынының жарықтандыру құралының толық ағынына қатынасы болып табылады.

Біздің есептейтін бөлменің өлшемдері мынадай:

- габариттері: 4x3x2,5;
- шағылу коэффициенттері:
  - $\rho_{\text{төбе}} = 70\%$ ;
  - $\rho_{\text{қабырға}} = 50\%$ ,
  - $\rho_{\text{еден}} = 30\%$ ;
- жарықтандырудың бірқалыпсыздық коэффициенті  $\lambda = 1,1 \div 1,2$ ;
- еденнен жұмыс орнына дейінгі биіктік:  $h_{\text{еден}} = 0,8$  м.

Нормаланған жарық қабырғалары, төбесі әктелген, терезелері жабылмаған бөлме үшін  $E=200$  лк. Жарықтандыру жүйесіне люминесцентті ЛД шамдарын қолданамыз, оның қуаты 20 Вт, ал жарықтық ағыны 920 лм.

Ілінудің есептік ұзындығын анықтаймыз. Жұмыс орнының еденнен биіктігі  $h_{\text{еден}} = 0,8$  м, ал шамның іліну биіктігі  $h_{\text{іліну}} = 0,2$  м болғандықтан, ол келесі шамаға тең

$$h = h - (h_{\text{еден}} + h_{\text{іліну}}) \quad (4.1)$$

$$h = 3 - (0,8 + 0,2) = 2,1 \text{ м.}$$

Шырақтар арасындағы ең қолайлы қашықтық төмендегі өрнекпен анықталады

$$z = \lambda * h \quad (4.2)$$

$$z = 1,3 * 2,1 = 1,8 \text{ м}$$

Шырақтардың қатарлар санын санайық

$$n = \frac{B}{L} \quad (4.3)$$

мұнда  $B$  – бөлме ені,  $B = 10\text{м}$ ;

$L$  – шырақтар арасындағы қашықтық,  $L = 2.8 \text{ м}$ .

Осыдан

$$n = \frac{10}{2.8} \approx 3$$

Сәйкесінше шырақтарды бір немесе екі қатарға орналастыруға болады.

Бөлмеге 3 қатар шырақ орнатамыз, қатарларының ара қашықтығы 2,8м, ал қабырға мен шырақтар арасындағы қашықтық 1,3м.

Бөлме индекcін анықтаймыз:

$$i = \frac{S}{h(A + B)} \quad (4.4)$$

мұндағы:  $A$  – бөлме ұзындығы;

$B$  – бөлменің ені;

$S = A \cdot B = 100 \text{ м}^2$  – ауданы;

$$i = \frac{100}{2 * (10 + 10)} = 2,5$$

Кесте бойынша пайдалану коэффициенті  $\eta = 55\%$ , қор коэффициенті

$$K_k = 1,5$$

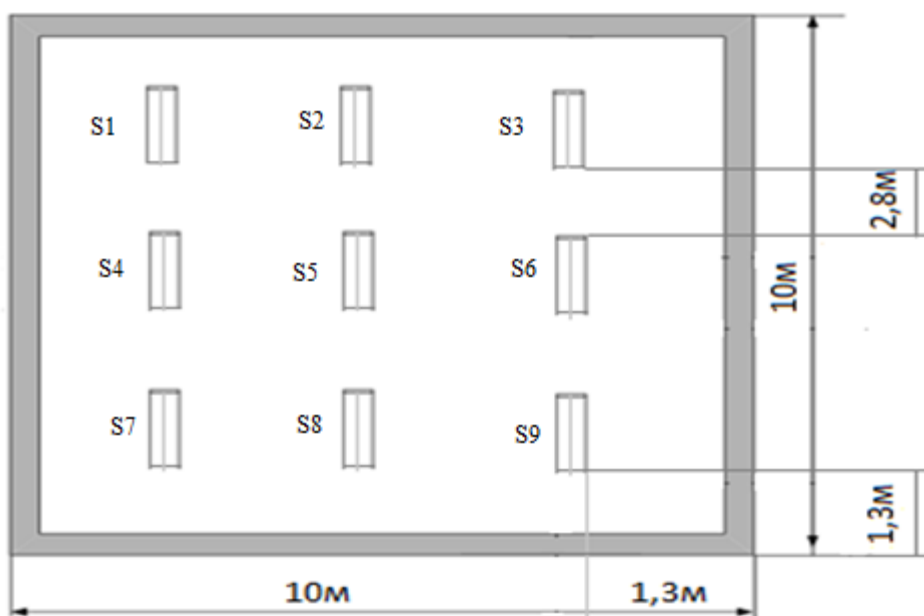
Қажетті шамдар санын анықтаймыз

$$N = \frac{E \cdot K_k \cdot S \cdot Z}{n \cdot \Phi_d \cdot \eta} \quad (4.5)$$

мұндағы  $E$  – берілген жарықтық;  
 $K_k$  – қор коэффициенті;  
 $S$  – бөлменің ауданы,  $m^2$ ;  
 $Z$  – жарықтандырудың бірқалыпсыздық коэффициенті.

$$N = \frac{200 \cdot 1.7 \cdot 100 \cdot 1.1}{2 \cdot 3570 \cdot 0.55} \approx 9 \text{ шам.}$$

Әр қатарға шам орнатылады, олардың арасындағы қашықтық (шамның ұзындығын ескереміз) 28м, ал қабырға мен шырақтар арасындағы қашықтық 1,2 м. Шамдардың бөлме ішінде орналасу сұлбасы сурет 4.2–те келтірілген.



4.2 сурет – Жасанды жарықтандыру сұлбасы

Нормаланған 200 лк жарықтандыруды қамтамасыз ету үшін қуаты 20Вт болатын ЛД20–2 типті 9 шам қажет. ЛД20–2 типті шамдар қажетті есептелінген жарық ағынын (920 лм) қамтамасыз етеді. өзге шамдарға қарағанда құрылымы мен сыртқы көрінісі желінің номиналды кернеуіне (220В) және бөлме ішіндегі қоршаған орта шарттарын орындай алады. Бұл түрдегі шамның пайдалы әсер коэффициенті 85% – ке тең.

#### 4.7 Меншікті қуат әдісімен жарықтандыруды есептеу

Меншікті қуат әдісімен жарықтандыруды шешудің маңызы, шырақтардың түрі мен оны орнатудың орнына байланысты, жұмыс бетінен

ілгіштің биіктігі, горизонтальды беттегі жарықтандыру мен бөлменің ауданымен меншікті қуат мәні анықталаумен қорытындыланады.

ЛДР шырақтарын орнатуды қабылдаймыз және параллель 2 қатарға, бір–бірден орналыстырамыз.

$$N = 3 \cdot 3 = 9 \text{ дана.} \quad (4.6)$$

Шырақтың меншікті қуаты  $\omega = 12 \text{ Вт/м}^2$ .

Бір шырақтың қуаты

$$\rho = \frac{\omega \cdot S}{N} = \frac{12 \cdot 100}{9} = 133 \text{ Вт.} \quad (4.7)$$

9 шырақ орнатамыз. Әр шамның қуаты

$$\rho_g = \frac{133}{2} = 66 \text{ Вт}$$

Сәйкесінше, екі әдіспен келтірілген есептер негізінде екі шырақты таңдап аламыз, олардың әр қайсысы 66 Вт тұрады.

#### 4.8 Өнеркәсіптегі жасанды жарықтандыру

Өнеркәсіптегі жарықтандыру адамның көру жұмыс әрекетіне, физикалық және моральді жағдайына, яғни жұмыс өнімділігіне, сапасына және өнеркәсіптік жарақаттануға үлкен әсерін тигізеді.

Еңбек етудің қолайлы жағдайын жасау үшін өнеркәсіп бөлмелері келесі талаптарға жауап беруі керек:

- жарықтану гигиеналық нормаларға сай болу керек;
- жұмыс бетіндегі және айналадағы жарық бірқалыпты болу керек;
- көлеңкеге өтуі күрт болмауы тиіс;
- жұмыс бетінен шағылысулар болмау керек;
- дұрыс жарық берілу үшін жарықтың қажетті спектралді құрамы қамтамасыз етілуі керек.

Өндіріс бөлмесін құру жарық көзін таңдаумен, оларды орналастырумен, жарықты–техникалық есептеулермен және жарықтандыру құрылғыларын анықтаумен сипатталады.

Сызықты – аппарат бөлмесіне ең қолайлысы ЛД типті люминесцентті шам болып табылады. Төбеге бекітілетін шамдар шаңдануды аздату үшін төбеден 50 мм–ден аспай ілінуі керек. Шам қалпақшалары өткізу коэффициенті 0,7–ден кем болмайтын сәуле шағылыстырушы материалдан жасалады.

Бөлменің жарықтану нормасы ( $E_{min}$ ) сол бөлмедегі минималды объектке қатысты жасалынатын жұмысқа қатысты көру жұмысының разрядына байланысты.

## Қорытынды

Бұл дипломдық жобада барлық қойылған мақсаттар іске асырылды. «Даулет–Жаркент» кәсіпқой спорт клубына арналған веб–қосымша құрылды. Веб–қосымшаны өңдеу барысында барлық нюанстар ескерілді.

Веб–қосымшаны жасау барысында бірнеше веб–программалау тілдері қолданылды. Олардың қатарында HTML гипермәтіндік программалау тілі, MySQL деректер қорын басқару жүйесі, CSS, Java Script, PHP5 тілдері, JQuery кітапханасы және Apache сервері пайдаланылады. Жаңа түскен ақпаратқа байланысты деректерді тек администратор ғана енгізе алады. Сонымен қатар, авторизация қолдану арқылы деректердің қауіпсіздік пен құпиялылық мәселелері қарастырылды.

Бұл бағдарламалық өнім «Даулет–Жаркент» кәсіпқой спорт клубында қолданылады.

«Тіршілік қауіпсіздігі» бөлімінде еңбек жағдайларына, жұмыс орнына талдау жасалынды. Жұмыс орнында жасанды жарықтандыру бойынша есептеулер жүргізілген.

Техника – экономикалық негізделуі бөлімінде барлық баптар бойынша жобаны жүзеге асырудағы шығындар анықталды және жобаға экономикалық пайдалылық бағасы берілді.

## Әдебиеттер тізімі

1. Котеров Д., Костарев А. PHP 5 в подлиннике. – СПб, 2012.
2. Yii Framework: Разработка веб-приложений на PHP. / Перевод В. Васвани. – СПб: Питер, 2012. – 432 с.
3. Сайт <http://yiiframework.ru/>
4. Базылов Қ.Б., Алибаева С.А., Нурмагамбетова С.С. Бітіруші жұмысының экономикалық бөлімі үшін әдістемелік нұсқаулар 050719–Радиотехника, электроника және телекоммуникация мамандығының барлық оқу түрінің студенттеріне арналған. – Алматы; АИЭС, 2009.
5. Экономика промышленного предприятия: Учебник / И.Н. Иванов. – М.: ИНФРА–М, 2011. – 395 с.
6. Экономика труда: Учебник / А.И. Рофе. – М.: КноРус, 2010. – 400 с.
7. Экономика труда: Учебное пособие / Ю.М. Остапенко. – М.: ИЦ РИОР, 2010. – 160 с.
8. Абдимуратов Ж.С., Дюсебаев М.К., Еңбекті қорғау және тіршілік қауіпсіздігінің негіздері. Дәрістер жинағы. – Алматы: АЭЖБИ, 2007. – 35 б.
9. ГОСТ 12.1.005–88 ССБТ. Система стандартов безопасности труда. Общие санитарно-гигиенические требования к воздуху рабочей зоны. – М.: Издательство стандартов, 1988.
10. МамаевЕ. MS SQL Server. Проектирование и реализация баз данных. Сертификационный экзамен. – СПб.: ВHV, 2004. – 416с.
11. Л.Бейли, М. Моррисон. Изучаем PHP и MySQL. – Эксмо, 2010.– 800 с.
12. Б. Лоусон, Р. Шарп. Изучаем HTML5. – Питер , 2011. – 272 с.
13. Сидерхолм Д. CSS3 для веб-дизайнеров. – Москва, 2013. – 124 с.
14. Николай Прохоренок. HTML, JavaScript, PHP и MySQL. Джентльменский набор Web-мастера. – БХВ–Петербург, 2010. – 900 с.

## ҚосымшаА

```
<?php
defined('SPORT') or die('Access denied');
session_start();
// подключение модели
require_once MODEL;
//print_r($_SERVER['DOCUMENT_ROOT']);
// подключение библиотеки функций
require_once 'functions/functions.php';

// получение массива каталога
$cat = catalog();
// получени массива страниц
$pages = pages();

$section=section();
// регистрация
if($_POST['reg']){
    registration();
    redirect();
}

// авторизация
if($_POST['auth']){
    authorization();
    if($_SESSION['auth']['user']){
        // если пользователь авторизовался
        echo "<p>Қош келдіңіз, {$_SESSION['auth']['user']}</p>";
        exit;
    }else{
        // если авторизация неудачна
        echo $_SESSION['auth']['error'];
        unset($_SESSION['auth']);
        exit;
    }
}
// ВЫХОД ПОЛЬЗОВАТЕЛЯ
if($_GET['do'] == 'logout'){
    logout();
    redirect();
}
```



```
// получение динамической части шаблона #content
$view = empty($_GET['view']) ? 'main' : $_GET['view'];
switch($view){
    case('main'):
        $news=get_all_news();
        //
        $meta['title'] = TITLE;
        $meta['description'] = TITLE;
        $meta['keywords'] = TITLE . ", Спорт клуб";
    break;
    case('page'):
        // отдельная страница
        $page_id = abs((int)$_GET['page_id']);
        $get_page = get_page($page_id);
        $meta['title'] = "{$get_page['title']} | " .TITLE;
        $meta['description'] = "{$get_page['description']} | " .TITLE;
    break;
    case('cat'):
        // товары категории
        $cat_id = abs((int)$_GET['category']);
        $text_category=get_text_category($cat_id);
        break;
    case('fullnews'):
        $news_id=abs((int)$_GET['news_id']);
        $news_text=get_news_text($news_id);

    break;
    case('news'):
        $get_news=get_all_news();
        break;
    case('reg'):
        break;
    case('order'):
        if($_POST['order']){
            add_order();
            redirect();
        }
    break;
    case('search'):
        $result_search=search();
        $count_rows = count($result_search);
    break;
```

```
case('order'):

    break;
    default:
        // если из адресной строки получено имя несуществующего вида
        $view = 'main';

    }
    // подключение вида
    require_once
$_SERVER['DOCUMENT_ROOT'].'/sport/views/sport/index.php';
http://sport/views/ishop/index.php //

<?php defined('SPORT') or die('Access denied'); ?>
<?php

/* =====Каталог – получение массива===== */
function catalog(){
    $query = "SELECT * FROM categories ORDER BY parent_id,cat_id";
    $res = mysql_query($query) or die(mysql_query());

    //массив категорий
    $cat = array();
    while($row = mysql_fetch_assoc($res)){
        if(!$row['parent_id']){
            $cat[$row['cat_id']] = $row['cat_name'];
        }else{
            $cat[$row['parent_id']]['sub'][$row['cat_id']] = $row['cat_name'];
        }
    }
    return $cat;
}
/* =====Каталог – получение массива===== */
/* =====Страницы===== */
function pages(){
    $query = "SELECT page_id, title FROM pages ORDER BY position";
    $res = mysql_query($query);

    $pages = array();
    while($row = mysql_fetch_assoc($res)){
```

*А қосымшасының жалғасы*

```
        $pages[] = $row;
    }
    return $pages;
}
/* ===Страницы=== */

/* ===Отдельная страница=== */
function get_page($page_id){
    $query = "SELECT title, keywords, description, text FROM pages WHERE
page_id = $page_id";
    $res = mysql_query($query);

    $get_page = array();
    $get_page = mysql_fetch_assoc($res);
    return $get_page;
}
/* ===Отдельная страница=== */

function get_all_news(){
    $query = "SELECT news_id,title, anons, date,datetime FROM news
ORDER BY date DESC LIMIT 6";
    $res = mysql_query($query);

    $all_news = array();
    while($row = mysql_fetch_assoc($res)){
        $all_news[] = $row;
    }
    return $all_news; }
/* ===Получение текста информера=== */
function get_text_category($cat_id){
    $query = "SELECT cp_id,title,keywords,description,text
FROM cat_page
LEFT JOIN categories ON categories.cat_id = cat_page.cat_id
WHERE cp_id = $cat_id";
    $res = mysql_query($query);

    $text_category= array();
    $text_category= mysql_fetch_assoc($res);
    return $text_category;
}
```

*А қосымшасының жалғасы*

```
/* ===Получение текста информера=== */

/* ===Отдельная новость=== */
function get_news_text($news_id){
    $query = "SELECT title, text, date,datetime FROM news WHERE news_id
= $news_id";
    $res = mysql_query($query);

    $news_text = array();
    $news_text = mysql_fetch_assoc($res);
    return $news_text;
}
/* ===Отдельная новость=== */

/* ===Регистрация=== */
function registration(){
    $error = ""; // флаг проверки пустых полей

    $login = trim($_POST['login']);
    $pass = trim($_POST['pass']);
    $name = trim($_POST['name']);
    $email = trim($_POST['email']);
    $phone = trim($_POST['phone']);
    $address = trim($_POST['address']);

    if(empty($login)) $error .= '<li>Логинді толтырыңыз!</li>';
    if(empty($pass)) $error .= '<li>Парольді толтырыңыз!</li>';
    if(empty($name)) $error .= '<li>Аты-жөніңізді толтырыңыз!</li>';
    if(empty($email)) $error .= '<li>Emailді толтырыңыз!</li>';
    if(empty($phone)) $error .= '<li>Телефонды толтырыңыз!</li>';
    if(empty($address)) $error .= '<li>Мекен-жайыңызды толтырыңыз!</li>';

    if(empty($error)){
        // если все поля заполнены
        // проверяем нет ли такого юзера в БД
        $query = "SELECT user_id FROM users WHERE login = '"
.clear($login). "' LIMIT 1";
        $res = mysql_query($query) or die(mysql_error());
        $row = mysql_num_rows($res); // 1 – такой юзер есть, 0 – нет
        if($row){
            // если такой логин уже есть
```

*А қосымшасының жалғасы*

```
$_SESSION['reg']['res'] = "<div class='error'>Сайтта бұл логинмен  
тіркелген қолданушы бар. Басқа логин енгізіңіз.</div>";  
    $_SESSION['reg']['name'] = $name;  
    $_SESSION['reg']['email'] = $email;  
    $_SESSION['reg']['phone'] = $phone;  
    $_SESSION['reg']['address'] = $address;  
}else{  
    // если все ок – регистрируем  
    $login = clear($login);  
    $name = clear($name);  
    $email = clear($email);  
    $phone = clear($phone);  
    $address = clear($address);  
    $pass = md5($pass);  
    $query = "INSERT INTO users (fname, email, phone, address, login,  
password)  
VALUES ('$name', '$email', '$phone', '$address', '$login',  
'$pass')";  
    $res = mysql_query($query) or die(mysql_error());  
    if(mysql_affected_rows() > 0){  
        // если запись добавлена  
        $_SESSION['reg']['res'] = "<div class='success'>Сіз сәтті  
тіркелдіңіз.</div>";  
        $_SESSION['auth']['user'] = $_POST['name'];  
        $_SESSION['auth']['user_id'] = mysql_insert_id();  
        $_SESSION['auth']['email'] = $email;  
    }else{  
        $_SESSION['reg']['res'] = "<div class='error'>Қате!</div>";  
        $_SESSION['reg']['login'] = $login;  
        $_SESSION['reg']['name'] = $name;  
        $_SESSION['reg']['email'] = $email;  
        $_SESSION['reg']['phone'] = $phone;  
        $_SESSION['reg']['address'] = $address;  
    }  
}  
}else{  
    // если не заполнены обязательные поля  
    $_SESSION['reg']['res'] = "<div class='error'>Міндетті орындар  
толтырылмады: <ul> $error </ul></div>";  
    $_SESSION['reg']['login'] = $login;  
    $_SESSION['reg']['name'] = $name;  
    $_SESSION['reg']['email'] = $email;
```

*А қосымшасының жалғасы*

```
        $_SESSION['reg']['phone'] = $phone;
        $_SESSION['reg']['address'] = $address;
    }
}
/* ===Регистрация=== */

/* ===Авторизация=== */
function authorization(){
    $login = mysql_real_escape_string(trim($_POST['login']));
    $pass = trim($_POST['pass']);

    if(empty($login) OR empty($pass)){
        // если пусты поля логин/пароль
        $_SESSION['auth']['error'] = "Логин және парольді толтырыңыз!";
    }else{
        // если получены данные из полей логин/пароль
        /* $pass = md5($pass);*/

        $query = "SELECT user_id, fname, email FROM users WHERE login =
'$login' LIMIT 1";
        $res = mysql_query($query) or die(mysql_error());
        if(mysql_num_rows($res) == 1){
            // если авторизация успешна
            $row = mysql_fetch_row($res);
            $_SESSION['auth']['user_id'] = $row[0];
            $_SESSION['auth']['user'] = $row[1];
            $_SESSION['auth']['email'] = $row[2];
        }else{
            // если неверен логин/пароль
            $_SESSION['auth']['error'] = "Логин және пароль кате!";
        }
    }
}
/* ===Авторизация=== */
/* ===Поиск=== */
function search(){
    $search = clear($_GET['search']);
    $result_search = array(); //результат поиска

    if(mb_strlen($search, 'UTF-8') < 4){
        $result_search['notfound'] = "<div class='error'>Іздеу сөзі 4 символдан
кем болмауы тиіс!</div>";
    }
}
```

```
}else{
    $query = "SELECT news_id, title, anons,text,date,datetime
              FROM news
              WHERE MATCH(text) AGAINST('{ $search }*' IN BOOLEAN
MODE)";
    $res = mysql_query($query) or die(mysql_error());

    if(mysql_num_rows($res) > 0){
        while($row_search = mysql_fetch_assoc($res)){
            $result_search[] = $row_search;
        }
    }else{
        $result_search['notfound'] = "<div class='error'>Сіздің сұранысыңыз
бойынша мәлімет табылмады</div>";
    }
}

return $result_search;
}
/* ===Поиск=== */

function section(){
    $query = "SELECT section_id, section_name,price FROM section";
    $res = mysql_query($query);

    $sections = array();
    while($row = mysql_fetch_assoc($res)){
        $sections[] = $row;
    }
    return $sections;
}

/* ===Добавление заказа=== */
function add_order(){
    $section_id = (int)$_POST['section'];
    $prim = trim($_POST['prim']);
    if($_SESSION['auth']['user']) $user_id = $_SESSION['auth']['user_id'];

    if(!$_SESSION['auth']['user']){
        $error = ""; // флаг проверки пустых полей
    }
}
```

*А қосымшасының жалғасы*

```
$name = trim($_POST['name']);
$email = trim($_POST['email']);
$phone = trim($_POST['phone']);
$address = trim($_POST['address']);

if(empty($name)) $error .= '<li>Аты–жөніңізді толтырыңыз</li>';
if(empty($email)) $error .= '<li>Emailді толтырыңыз</li>';
if(empty($phone)) $error .= '<li>Телефонды толтырыңыз</li>';
if(empty($address)) $error .= '<li>Адресі толтырыңыз</li>';

if(empty($error)){
    // добавляем гостя в заказчики (но без данных авторизации)
    $user_id = add_customer($name, $email, $phone, $address);
    if(!$user_id) return false; // прекращаем выполнение в случае
    возникновения ошибки добавления гостя–заказчика
} else {
    // если не заполнены обязательные поля
    $_SESSION['order']['res'] = "<div class='error'>Міндетті жолдар
    толтырылмады: <ul> $error </ul></div>";
    $_SESSION['order']['name'] = $name;
    $_SESSION['order']['email'] = $email;
    $_SESSION['order']['phone'] = $phone;
    $_SESSION['order']['address'] = $address;
    $_SESSION['order']['prim'] = $prim;
    return false;
}
}
$_SESSION['order']['email'] = $email;
save_order($user_id,$section_id,$prim);
}
/* ===Добавление заказа=== */

/* ===Добавление заказчика–гостя=== */
function add_customer($name, $email, $phone, $address){
    $name = clear($_POST['name']);
    $email = clear($_POST['email']);
    $phone = clear($_POST['phone']);
    $address = clear($_POST['address']);
    $query = "INSERT INTO users (fname, email, phone, address)
    VALUES ('$name', '$email', '$phone', '$address)";
    $res = mysql_query($query);
    if(mysql_affected_rows() > 0){
```



*А қосымшасының жалғасы*

```
// если гость добавлен в заказчики – получаем его ID
return mysql_insert_id();
}else{
// если произошла ошибка при добавлении
$_SESSION['order']['res'] = "<div class='error'>Өтініш жіберу кезінде
кате болды</div>";
$_SESSION['order']['name'] = $name;
$_SESSION['order']['email'] = $email;
$_SESSION['order']['phone'] = $phone;
$_SESSION['order']['address'] = $address;
$_SESSION['order']['prim'] = $prim;
return false;
}
}
/* ===Добавление заказчика–гостя=== */

/* ===Сохранение заказа=== */
function save_order($user_id, $section_id, $prim){
$prim = clear($prim);
$query = "INSERT INTO apps (user_id, date, section_id, note)
VALUES ($user_id, NOW(), $section_id, '$prim)";
mysql_query($query) or die(mysql_error());
if(mysql_affected_rows() == -1){
// если не получилось сохранить заказ – удаляем заказчика
mysql_query("DELETE FROM users
WHERE user_id = $user_id AND login = ''");
return false;
}
$order_id = mysql_insert_id(); // ID сохраненного заказа

if($_SESSION['auth']['email']) $email = $_SESSION['auth']['email'];
else $email = $_SESSION['order']['email'];
//mail_order($order_id, $email);

// если заказ выгрузился
unset($_SESSION['cart']);

$_SESSION['order']['res'] = "<div class='success'>Өтініш бергеніңізге
рахмет. Жакын уақыт аралығында сізбен менеджер хабарласады.</div>";
return true;
}
```

*А қосымшасының жалғасы*

```
/* ===Сохранение заказа=== */

/* ===Отправка уведомлений о заказе на email=== */
function mail_order($order_id, $email){
    //mail(to, subject, body, header);
    // тема письма
    $subject = "Заявка на сайте";
    // заголовки
    $headers .= "Content-type: text/plain; charset=utf-8\r\n";
    $headers .= "From: SPORT";
    // тело письма
    $mail_body = "Благодарим Вас за заявку!\r\nНомер Вашего заявки –
{$order_id}
\r\n\r\nЗаявка : \r\n";
    /* // атрибуты товара
        $mail_body .= "Наименование: {$value['name']}\r\n";
    }
    $mail_body .= "\r\nИтого: {$_SESSION['total_quantity']} на сумму:
{$_SESSION['total_sum']}";*/

    // отправка писем
    mail($email, $subject, $mail_body, $headers);
    mail(ADMIN_EMAIL, $subject, $mail_body, $headers);
}
/* ===Отправка уведомлений о заказе на email=== */

<?php defined('SPORT') or die('Access denied'); ?>
<div id="right-bar">
    <div class="right-bar-cont">
        <h2><span>Клуб туралы</span></h2>
        <div>
            <ul class="nav-catalog" id="accordion">
                <?php foreach($cat as $key => $item): ?>
                    <?php if(count($item) > 1): // если это родительская
категория ?>
                        <h3><li><a href="#"><?=$item[0]?></a></li></h3>
                        <ul>
                            <li>
                                <a href="?view=cat&category=?=$key?">Барлығы</a></li>
                                <?php foreach($item['sub'] as $key => $sub): ?>
                                    <li>
                                        <a href="?view=cat&category=?=$key?"><?=$sub?></a></li>

```

```
<?php endforeach; ?>
</ul>
<?php elseif($item[0]): // если самостоятельная категория ?>
<li><a
href="?view=cat&category=<?=$key?>"><?=$item[0]?></a></li>
<?php endif; ?>
<?php endforeach; ?>
</ul>
</div>
<div class="enter">
<h2><span>Сайтқа кіру</span></h2>
<div id="logins" class="authform">
<?php if(!$_SESSION['auth']['user']): ?>
<form method="post" action="#">
<label for="login">Логин: </label><br />
<input type="text" name="login" id="login" /><br />
<label for="pass">Пароль: </label><br />
<input type="password" name="pass" id="pass" /><br /><br />
<input type="submit" name="auth" id="auth" value="Кіру" />
<p class="link"><a href="?view=reg">Тіркелу</a></p>
</form>

<?php
if(isset($_SESSION['auth']['error'])) {
    echo '<div class="error">' . $_SESSION['auth']['error'] . '</div>';
    unset($_SESSION['auth']);
}
?>
<?php else: ?>
<p>Қош
<?=htmlspecialchars($_SESSION['auth']['user'])?></p>
<a href="?do=logout">Шығу</a>
<?php endif; ?>
</div>
<h2><span>Қабылдау</span></h2>
<div class="zayavka">
<a href="<?=PATH?>?view=order"></a>
</div>
</div>

</div>
```

келдіңіз,