

**ЎЗБЕКИСТОН РЕСПУБЛИКАСИ АЛОҚА,
АХБОРОТЛАШТИРИШ ВА ТЕЛЕКОММУНИКАЦИЯ
ТЕХНОЛОГИЯЛАРИ ДАВЛАТ ҚЎМИТАСИ
ТОШКЕНТ АХБОРОТ ТЕХНОЛОГИЯЛАРИ УНИВЕРСИТЕТИ**

Ҳимояга

Кафедра мудири

« _____ »

« ____ » _____ 2014й.

БАКАЛАВР БИТИРУВ МАЛАКАВИЙ ИШИ

Мавзу: YII Framework имкониятларидан фойдаланиб WEB
саҳифаларни ишлаб чиқиш

Битирувчи	_____	_____
	(имзо)	(ф.и.о.)
Раҳбар	_____	_____
	(имзо)	(ф.и.о.)
ҲФХ бўйича консультант	_____	_____
	(имзо)	(ф.и.о.)
Такризчи	_____	_____
	(имзо)	(ф.и.о.)

Тошкент – 2014

ТОШКЕНТ АХБОРОТ ТЕХНОЛОГИЯЛАРИ УНИВЕРСИТЕТИ

Факультет: Компьютер инженеринг

Кафедра: Ахборот технологиялари

Йўналиш(мутахассислик): 5521900 Информатика ва ахборот
технологиялари

«ТАСДИҚЛАЙМАН»

Кафедра мудири _____

«__» _____ 2014 й.

Битирув малакавий ишига

ТОПШИРИҚ

Махмадиев Феруз Расулович

(фамилия, исми, отасининг исми)

1. БМИ мавзуси: YII Framework имкониятларидан фойдаланиб WEB саҳифаларни ишлаб чиқиш

2. 2014 йилдаги №____ сонли буйруқ билан тасдиқланган

3. Ишни ҳимояга топшириш муддати _____

4. Ишга оид дастлабки маълумотлар: илмий ва техник адабиётлар, интернет сайтлари, дастурлаш тили.

5. Кириш, Web технологиялар ва инструментал воситалар таҳлили Web саҳифаларни лойиҳалаштириш, “Маърифат юлдузи таълим маркази” веб-сайтнинг таснифи, Ҳаёт фаолияти хавфсизлиги, Хулоса.

6. График материаллар рўйхати: Microsoft Power Point дастурида намойиш слайдлари

7. Топшириқ берилган сана _____

Раҳбар _____
(имзо)

Топшириқ олдим _____
(имзо)

8. Ишнинг айрим бўлимлари бўйича маслаҳатчилар

Қисм	Маслаҳатчи ўқитувчининг Ф.И.Ш	Имзо, сана	
		Топшириқ берилди	Топшириқ олинди
Асосий қисм ХФХ			

9. Ишни бажариш графиги

№	БМИ бўлимларининг номи	Бажариш муддати	Раҳбар (маслаҳатчи) имзоси
1	Кириш	17.03.2014	
2	Web технологиялар ва инструментал воситалар таҳлили	17.04.2014	
3	Web саҳифаларни лойиҳалаш	07.05.2014	
4	“Маърифат юлдузи таълим маркази” веб-сайтнинг таснифи	15.05.2014	
5	Ҳаёт фаолияти хавфсизлиги	23.05.2014	
6	Хулоса	25.05.2014	

Битирувчи _____
(имзо)

2014 йил « ____ » _____

Раҳбар _____
(имзо)

2014 йил « ____ » _____

МАЗМУННОМА

Битирув малакавий иши хозирги вақтда долзарб бўлган Yii Framework имкониятларидан фойдаланиб WEB саҳифаларни ишлаб чиқишдан иборат.

Ушбу битирув малакавий ишда WEB технологиялар асосида ишлаб чиқиладиган дастурларнинг ёзилишини осонлаштирадиган метод ва кутубхоналар кўрсатилган бўлиб, унда дастур коди ёзилиши камайтиради. Ишда Yii Framework ишлатган ҳолда тузилган ВЕБ саҳифалар мисол сифатида кўрсатилган.

Кўрсатилган дастурдаги асосий кодлар PHP дастурлаш тилида ишланган ва дастур базаси MySQL маълумотлар базасида шакллантирилган.

АННОТАЦИЯ

Данная выпускная квалификационная работа посвящена к актуальной в данный момент проблеме - разработке WEB страниц используя возможности Yii Framework.

В данной выпускной квалификационной работе указываются методы и библиотеки облегчающие записи программ разрабатываемых на основе ВЕБ технологий. При этом уменьшется записи кодов. В качестве примера приводится ВЕБ страницы составленные применяя Yii Framework.

Основные коды программы разработаны на языке программирования PHP и база программы составлена на базе данных MySQL.

ANNOTATION

This Graduate work is devoted to the actual problem at the moment - the development of WEB pages using the capabilities Yii Framework. In this final qualifying work specified methods and facilitate biblioteki razrabotyvaemyh record programs based on WEB technologies. Thus umenshetsya recording codes. As an example, a Web page created by using the Yii Framework. The main program codes developed in the programming language PHP and database programs compiled on a database MySQL.

Мундарижа

Кириш.....	6
1. БОБ WEB ТЕХНОЛОГИЯЛАР ВА ИНСТРУМЕНТАЛ ВОСИТАЛАР ТАҲЛИЛИ	8
1.1. Веб технологиялар таҳлили	8
1.2. PHP тилида дастурлаш	9
1.3. MySQL маълумотлар базасини бошқариш тизими	17
2. БОБ WEB САҲИФАЛАРНИ ЛОЙИҲАЛАШ	22
2.1. Web-саҳифаларни лойиҳалаш воситалари.....	22
2.2. Web-саҳифаларда матн ва графикани тақдим этиш.	25
2.3. Сайт яратишнинг асосий қоидалари ва босқичлари.	29
3. БОБ “МАЪРИФАТ ЮЛДУЗИ ТАЪЛИМ МАРКАЗИ” ВЕБ-САЙТНИНГ ТАСНИФИ.....	35
3.1. Маълумотлар базасининг архитектураси.....	35
3.2. Веб-сайтнинг асосий модулларининг таснифи.....	38
3.3. Веб-сайтнинг фойдаланиш тартиби.....	43
4. БОБ ҲАЁТ ФАОЛИЯТИ ХАВФСИЗЛИГИ.....	52
4.1. Ишлаб чиқаришда сунъий ёритилганлик	52
4.2. Ёнғин хавфсизлиги	55
Хулоса.....	59
Фойдаланилган адабиётлар рўйхати.....	60
Илова.....	61

КИРИШ

Ҳозирда Ахборот технологиялари жадал суръатлар билан ривожланиб бормоқда ва жамиятнинг барча соҳаларига тадбиқ этилмоқда. Ахборот технологияларидан кенг ва унумли фойдаланиш мақсадга тез ва сифатли эришишга имконият яратади. XXI аср – ахборот асрида ахборот коммуникация технологиялари (АКТ) кириб бормаган соҳа ёйинки даргоҳ қолмади. Жаҳон ҳамжамиятининг кўзланган ривожланиш ва фаровонликка эришиш учун, ахборот технологиялари(АТ)га бўлган эҳтиёжи катта суръатлар билан ошиб бормоқда. Иқтисодий ўсишнинг фаоллашуви, дунё аҳолиси яшаш даражасининг яхшиланиши ахборот технологияларининг кундалик ҳаётимизга сингиб кетгани натижасидир. Дунё тажрибаси шуни кўрсатадики эркин ахборот оқимининг таъминланиши бозор иқтисодиётига ўтишни тезлаштиради ва ижтимоий фаровонликни оширади .

Глобал тармоқ-интернетдан фойдаланиш мамлакатлараро алоқаларнинг янада ривожланишига туртки бўлди. Ҳозирда Ўзбекистон Республикасининг жаҳон ҳамжамиятига интеграциялашув жараёни жадал суръатлар билан амалга ошиб бораётган даврда ахборот технологиялари ютуқларини ҳаётимизга кенг жалб этиш замон талабидир[3].

Ишдан мақсад:

1. Веб-сайт Yii framework асосида яратилган.
2. Yii бу юқори эффектли PHP фреймворк компонент структурасига асосланган йирик Веб иловаларни қайта ишловчи ҳисобланади.
3. Yii бу йирик Веб иловаларни қайта ишловчи компонент структурасига асосланган PHP фреймворкдир. У кодларни қайта ишлаш концепциясини максимал қулай имконини бериб веб қайта ишлаш жараёнини сезиларли тезлаштиради.

Вазифаси:

1. Yii бу юқори эффектли PHP фреймворк компонент структурасига асосланган йирик Веб иловаларни қайта ишловчи ҳисобланади.

2.ҮШ бу йирик Веб иловаларни қайта ишловчи компонент структурасига асосланган PHP фреймворкдир. У кодларни қайта ишлаш концепциясини максимал қулай имконини бериб веб қайта ишлаш жараёнини сезиларли тезлаштиради. ҮШ (U i :) деб ўқилади) деганини оддий (easy), эффектив (efficient), кенгаювчи (extensible) маъноларини беради

Битирув малакавий иш кириш, тўртта боб, хулоса, фойдаланилган адабиётлар рўйхати ва иловадан иборат.

Биринчи бобда Web технологиялар, PHP тилида дастурлш ва MySQL маълумотлар базаси бошқариш тизими таҳлил этилган.

Иккинчи бобда Web саҳифаларни лойиҳалаш масаласи кўриб чиқилган бўлиб, яратилаётган Web сайт учун матн ва графикани тақдим этиш, қоида ва босқичлари кўриб чиқилган.

Учунчи бобда “Маърифат юлдузи” таълим маркази Web сайти ишлаб чиқилган бўлиб, унинг тавсифи ва функционал имкониятлари ёритиб берилган.

Туртинчи боб ҳаёт фаолияти ҳавфсизлигига бағинланган унда ишлаб чиқаришда суний ёритилганлик ва ёнғин ҳавфсизлиги масалалари кўриб чиқилган.

Ҳаёт фаолияти ҳавфсизлиги бўлимида ишлаб чиқаришда сунъий ёритилганлик, иш майдонларида ёритишга қўйилган меъёрлар, ёритиш усуллари ҳақида ва ёнғин ҳавфсизлиги, ёнғин ҳавфсизлиги бўйича категорияларни аниқлаш, ёнғин портлаш ҳавфсизлигини аниқлаш ҳақида гапирилган.

Яратилган талаба автоматлаштирилган ахборот тизимини такомиллаштириш ва ишлаб чиқаришда қўллаш истиқболлари, мавзуга яқунлар битирув-малакавий ишимнинг хулоса қисмида ўз ифодасини топади.

1.БОБ. WEB ТЕХНОЛОГИЯЛАР ВА ИНСТРУМЕНЪАЛ ВОСИТАЛАР ТАҲЛИЛИ

1.1. Веб-технологиялар таҳлил

Веб-технологияни ўрганишни веб-дизайннинг қуйидаги учта тушунчасини ўрганишдан бошланади: веб-саҳифа, веб-сайт ва веб-сервер[5].

Веб-саҳифа – ўзининг уникал адресига эга бўлган ва махсус кўриш дастури ёрдамида (браузер) кўрилувчи ҳужжатдир. Унга матн, графика, овоз, видео ёки анимация маълумотлар бирлашмаси - мультимедияли ҳужжатлар, бошқа ҳужжатларга гипермуружаатлар кириши мумкин.

Веб-сайт – бир қанча веб-саҳифаларнинг мантиқий бирлашмаси.

Веб-сервер – тармоққа уланган компьютер ёки ундаги дастур ҳисобланиб, умумий ресурсларни клиентга тақдим этиш ёки уларни бошқариш вазифаларини бажаради. Веб-серверлар маълумотлар базалари ва мультимедияли маълумотларни бир бирига мослаштиради; веб-серверда веб-саҳифа ва веб-сайтлар сақланади.

Биз Интернет тармоғидаги веб-саҳифаларни кўришимиз учун **WWW (World Wide Web)** деб аталувчи сервисдан фойдаланамиз.

WWW - компьютер тармоқларида керакли маълумотни кўришни гипермуружаат деб аталувчи усул билан компьютер тармоқларида жойлаштириш усулидир.

WWW – World Wide Web ном Тим Бернерс – Лее (ЦЕРН лабораторияси) томонидан киритилгандир. У бошқача қилиб, ўргимчак яшаши учун турли йўллар ташкил қилиб, бу йўллар орқали турли нуқталарга юришга ўхшаб WWWда ҳам турли йўллар орқали тегишли маълумотларга етиб бориш ва уни кўриш имконияти борлигидир ва бутун дунё “ўргимчаклари” деб аталади. WWWда нуқталар ролини компьютер ўйнайди. Йўллар сифатида телефон йўллари ишлатилади. Веб саҳифалар одатда HTML ҳужжат, яъни HTML (Hyper Text Markup language- гиперматни белгилаш тили) тилида ёзилган ҳужжат сифатида тайёрланади.

WWWнинг яратилиш тарихига назар ташласак, 1989 йили ЦЕРН (Европа электрон зарралар физикаси лабораторияси) тадқиқотчилари ўз олдларига шундай система яратиш масаласини қўйишадики, бу система турли илмий гуруҳлар ўзаро алоқа қилишларини таъминлаши керак эди.

Баъзи жамоалар WWWдан фойдаланувчилар учун ишлашни осонлаштирувчи дастурлар ёзишга киришди. 1993 йилдан бошлаб WWW Интернетнинг ресурслари ичида энг оммавийсига айланди[5].

1.2. PHP тилида дастурлаш

Ҳозирги кунда интернет кенг оммалашгани сабабли, замон тараққиётини веб-технологиясиз тасаввур этиш мумкин эмас. Веб технологияларига талаб ошган сари Веб-дастурлаш тилларини билиш ҳар бир дастурчи учун муҳим вазифа саналмоқда.

PHP тарихи. Кўпгина бошқа дастурлаш тилларидан фарқли равишда, PHP қандайдир ташкилот ёки кучли дастурчи томонидан яратилган эмас. Уни оддий фойдаланувчи Расмус Лердорф 1994 йили ўзининг бош саҳифасини интерактив услубда кўрсатиш учун яратган. Унга Personal Home Page (PHP – шахсий бош саҳифа) деб ном берган[4,8].

1995 йили Расмус PHPни ўзининг HTML формалари билан ишлайдиган бошқа дастур билан умумлаштириб PHP/FI Version 2 ("Form Interpretator") ҳосил қилди. 1997 йилга бориб PHP дан фойдаланувчи сайтлар 50 мингдан ошди. Шундан сўнг веб технология усталари PHP ғояси асосида мукамал тил яратишга Зива Сураски ва Энди Гутманс асосчилигида киришилди. PHPни самарали деб ҳисобланмагани учун деярли нолдан бошлаб, мавжуд C ва Perl тилларидан ибрат олиб PHP3 талқинини яратилди. 1999 йилга келиб PHP асосида қурилган сайтлар миллиондан ошиб кетди. 2000 йилда эса Zend Technologies ширкати янги кўпгина функцияларни қўшган ҳолда PHP4 шарҳловчисини яратди.

PHP – веб технология тили. PHPни ўрганиш учун аввал HTML ва дастурлаш тилидан хабардор бўлиш талаб қилинади. HTML/CSS ва JavaScript

ларни мукамал билувчилар учун PHP ни ўрганиш мураккаблик туғдирмайди. PHP нинг вазифаси HTML файлини яратиб бериш. JavaScript ёрдамида бажариладиган кўпгина операцияларни PHP орқали ҳам амалга ошириш мумкин, аммо эътибор қилиш лозимки, PHP – серверда; JavaScript – клиент томонда бажарилади. PHP да ёзилган код сервернинг ўзида бажарилиб, клиентга HTML шаклида етиб боради. Бу хавфсизлик жиҳатдан анча мақсадга мувофиқдир. JavaScript ёрдамида код ёзиш, маълумот узатиш ва қабул қилишни биров тезлаштирсанда, кодни клиент кўриш имкониятига эга бўлади. Барибир ҳар иккисини бошқаси боса олмайдиган ўз ўрни бор, равшанки бу ўрин PHP да муҳимроқ ва каттароқ[4,8,9].

PHP имкониятлари. «PHP да ҳар қандай дастурни яратса бўлади», – деган эди унинг яратувчиси. Биринчи навбатда PHP тили сервер томонидан бажариладиган скриптлар яратиш учун фойдаланилади ва айнан шунинг учун у яратилган. Бироқ PHP тили фойдаланиладиган бошқа соҳалар ҳам мавжуд. Бу соҳаларни биз унга асосий қисмга бўламиз:

➤ Биринчи соҳа – биз юқорида айтиб ўтганимиздек, сервер томонидан бажариладиган иловалар (скриптлар) яратиш. PHP тили бундай турдаги скриптларни яратиш учун жуда кенг қўлланилади. Бундай иш кўрсатиш учун PHP-парсер (яъни *php*-скриптларни қайта ишловчи) ва скриптларни қайта ишловчи Веб-сервер, скриптларни натижасини кўриш учун браузер ва албатта *php*-кодини ёзиш учун қандай бўлса ҳам матн муҳаррири керак бўлади. PHP-парсер CGI-дастурлар кўринишида ёки сервер модуллари кўринишида тарқалган.

➤ Иккинчи соҳа – буйруқлар сатрида бажариладиган скриптларни яратиш. Яъни PHP тили ёрдамида бирор-бир компьютерда браузер ва Веб-серверлардан мустақил равишда ўзи бажариладиган скриптларни ҳам яратиш мумкин. Бу ишларни бажариш учун ҳеч бўлмаганда PHP-парсер (бу ҳолатда биз уни буйруқлар сатри интерпретатори (CLI, command line interpreter) деб атаيمиз) талаб этилади. Бундай ишлаш услуги турли масалаларни

режалаштириш ёрдамида бажарилиши учун керак бўлган скриптлар ёки оддий матнни қайта ишлаш учун керак бўлган масалага ўхшаш ишлайди.

➤ Учинчи соҳа – бу мижоз томонидан бажариладиган GUI-иловаларни (график интерфейс) яратиш. Бу соҳа PHP тилини эндигина ўрганаётган фойдаланувчилар учун унча муҳим бўлмаган соҳадир. Бирок агарда сиз PHP тилини чуқур ўрганган бўлсангиз, бу соҳа сиз учун анча муҳимдир. PHP тилини бу соҳага қўллаш учун php кенгайтмали махсус ёрдамчи – PHP-GTK талаб этилади.

PHP дастурлаш тилида дастурлашнинг иккита ҳаммабоп парадигмалари ишлатилади, булар процедурали ва объектли дастурлаш. PHP4 дастурлаш тили процедурали дастурлашни бутунлай қўллаб қувватлайди, бироқ объектли стилдаги дастурларни ҳам қўлласа бўлади. PHP5 дастурлаш тилининг биринчи тестлаш версиясида PHP4 дастурлаш тилида учрайдиган объектга йўналтирилган дастурлаш моделларининг камчиликлари тўлдирилган. Шундай қилиб, ҳозирда таниш бўлиб улгурган ишлаш принципини танлаш керак.

PHP дастурлаш тили ёрдамида тасвирлар, PDF-файллар, флэш-ролиқлар яратиш мумкин; ҳозирги вақтдаги замонавий маълумотлар базасини қўллаб қувватлайди; ихтиёрий матнли файл форматлари билан, ҳамда XML ва файллар тизими билан ишлайдиган функциялар ҳам қўшилган. PHP тили турли сервислар ўртасидаги протоколларнинг ўзаро алоқасини қўллаб қувватлайди. Буларга мисол тариқасида папкаларга киришни бошқариш протоколи LDAP, тармоқ қурилмалари билан ишлайдиган протокол SNMP, маълумотларни узатиш протоколлари IMAP, NNTP ҳамда POP3, гиперматнларни узатиш протоколи HTTP ва бошқаларни олиш мумкин.

PHP асосий синтаксиси. Биринчи навбатда PHP дастурлаш тили синтаксисига нисбатан нималарни билиш керак. Бу HTML-код ичига ўрнатилган ва PHP дастурлаш тилидаги коддир, уни интерпретатор фарқлай

билади. Кўп ҳолларда мисолларда `<?php ?>` вариант ўрнига қисқартирилган `<? ?>` теглардан фойдаланишни айтиб ўтамыз.

Инструкцияларни ажратилиши. PHP дастурлаш тилидаги дастур (ихтиёрий дастурлаш тилидаги) – бу буйруқлар (инструкциялар) тўпламидир. Дастурни қайта ишлаш учун бир буйруқни бошқа буйруқдан фарқини билиш керак. Бунинг учун махсус символлар – ажратгичлардан фойдаланилади. PHP дастурлаш тилида инструкцияларни худди C ёки Perl дастурлаш тиллари каби ажратилади, яъни ҳар бир ифода нуқтали вергул (“;”) билан тугайди[9].

«?>» ёпиладиган тег ҳам инструкцияни тугагини англатади, шунинг учун ундан олдин нуқтали вергул қўйилмайди. Масалан, қуйидаги икки фрагментлар эквивалентдир:

```
<?php
echo "Hello, world!"; // буйруқлар охирида
                        // нуқтали вергул
                        // қўйиш шарт
?>
<?php
echo "Hello, world!" ?>
<!-- "?"> борлиги учун
        нуқтали вергул ташлаб кетилди -->
```

Ўзгарувчилар, ўзгармаслар ва операторлар. Ҳар бир дастурлаш тилида муҳим элементлардан бири бу ўзгарувчилар, ўзгармаслар ва улар қўлланиладиган операторлардир. PHP дастурлаш тили бу элементларни қандай белгилаши ва қайта ишлашини кўриб чиқамиз.

Ўзгарувчилар. PHP дастурлаш тилида ўзгарувчилар олдида доллар белгиси (“\$”) қўйиб эълон қилинади, масалан, `$my_var`.

Ўзгарувчилар номлари регистрларни фарқлайди, яъни `$my_var` ҳамда бош ҳарфли `$My_var` ўзгарувчилари турли хил ўзгарувчилардир.

Ўзгармаслар. Скрипт бажарилиш жараёнида ўзгармайдиган қийматли катталикларни сақлаш учун ўзгармаслардан фойдаланилади. Бундай

катталиқлар математик ўзгармаслар, пароллар, файлларнинг йўллари ва бошқалар бўлиши мумкин. Ўзгармасларнинг ўзгарувчилардан асосий фарқи шуки, уларни фақат бир мартагина ўзлаштирилади ва уни қийматини эълон қилингандан кейин бекор қилиб бўлмайди. Бундан ташқари ўзгармаслар олдида доллар белгиси қўйилмайди ҳамда уни оддий қиймат ўзлаштириш каби қараш мумкин эмас. Ўзгармаслар қандай аниқланади? Бунинг учун махсус `define()` функцияси мавжуд, унинг синтаксиси қуйидагичадир:

```
define("Ўзгармас номи",  
      "Ўзгармас қиймати",  
      [регистрга_сезгирлиги_кичик])
```

Амаллар. Ўзгарувчилар, ўзгармаслар ва ифодалар устида турли ҳисоблашларни бажарадиган бу амаллардир. Биз ҳали бу ифодалар ҳақида тўхтаб ўтганимиз йўқ. Ифодалар қийматини ушбу амаллар ёрдамида аниқланади. Ўзгарувчилар ва ўзгармаслар – бу ифодаларнинг асосий ва жуда содда шаклидир.

PHP да массивлар. PHP дастурлаш тилида массив типни тартибланган карталарга ўхшайди ва қийматини калитга ўзлаштирадиган типдир. Бу тип бир неча йўналишларда оптималлаштирилади, шунинг учун сиз уни хусусий массив, рўйхат (вектор), хеш-жадвали (картани амалга ошириш учун ишлатилади), стек, навбат ва бошқалар сифатида фойдаланишингиз мумкин. Модомики, PHP дастурлаш тилида бир массивни қийматини бошқасига ўзлаштириш учун дарахтлардан фойдаланилади.

Массивларни `array()` конструкцияси ёрдамида аниқланади ёки элементларига қиймат бериш билан аниқланади.

```
array() конструкцияси ёрдамида аниқлаш.  
array ([key] => value,  
       [key1] => value1, ... )
```

Агарда TRUE ёки FALSE калит сифатида қўлланилса, у ҳолда унинг қиймати мос равишда `integer` типининг бир ва нолига ўзлаштирилади. Агар NULL дан фойдаланилса, у ҳолда калит ўрнига бўш сатр ҳосил бўлади. Бу

бўш сатрни калит сифатида фойдаланса бўлади, аммо уни қўштирноққа олиш керак бўлади. Бу усул бўш квадрат қавсни ишлатиш каби эмас. Массивлар ёки объектлар калити сифатида фойдаланиш мумкин ҳам эмас.

Массивга қиймат бериш орқали массив яратиш мумкин. Биз юқорида айтиб ўтганимиздек, массив элементи қийматига эга бўлиш учун квадрат қавс ичига унинг калити кўрсатилиши керак, масалан, `$subject["php"]`. Агарда янги калит ва янги қиймат кўрсатсангиз қуйидагича бўлади: `$subject["new_key"]="new_value"` ҳамда массивга янги элемент қўшилади. Агарда калитни кўрсатмай фақат қийматни ўзлаштирак, яъни `$subject[]="new_value"`, у ҳолда массивга янги элемент қўшилади ва уни калити мавжуд максимал қийматга бир қўшилади. Агарда биз қиймат берган массив яратилмаган бўлса, у ҳолда биз қиймат бергандан кейин у яратилади.

Альтернатив синтаксислар. PHP дастурлаш тили ўзининг бир нечта *if*, *while*, *for*, *foreach* ҳамда *switch* бошқариладиган структуралари учун альтернатив синтаксисни тақдим этади. Ҳар бир ҳолатда очиладиган қавс икки нуқтага (:), ёпиладигани эса мос равишда `endif`;, `endwhile`; ва ҳоказоларга ўзгартирилади.

Масалан, `if` шарт оператори синтаксисини қуйидагича ифодалаш мумкин:

```
if (ифода) : бажариладиган_блок endif;
```

Маъноси ўзгармасдан қолади: агар `if` шарт оператори думалок қавси ичидаги шарт рост бўлса, икки нуқтадан «:» то `endif`; буйруғигача барча код бажарилади. Бундай синтаксисдан фойдаланиш `html`-код ичида қурилган `php`-код учун қулайдир.

```
<?php
$names = array("Карим","Салим","Содик");
if ($names[0]=="Карим"):
?>
Салом, Карим!
<?php endif ?>
```

Агарда `else` ҳамда *elseif* конструкцияларидан фойдаланилса, у ҳолда ҳам альтернатив синтаксисдан фойдаланса бўлади:

```
<?php
if ($a == 5):
    print "a ўзгарувчи 5 га тенг";
    print "... ";
elseif ($a == 6):
    print "a ўзгарувчи 6 га тенг ";
    print "!!!";
else:
    print "a ўзгарувчи на 5 га ва на 6 га тенг ";
endif; ?>
```

Синфлар ва объектлар. Объектга йўналтирилган дастурлашнинг асосий тушунчалари – синфлар ҳамда объектлардир. Бу тушунчаларни қуйидагича тушуниш мумкин: объект – бу дастурда қўлланиладиган тушунча ёки бирор физик предмет ҳақида маълумот берадиган структураланган ўзгарувчидир, синфлар эса бу объектларнинг тавсифи ва улар устида бажариладиган ҳаракатлардир[4,9].

PHP дастурлаш тилида синфлар қуйидаги синтаксис ёрдамида аниқланади:

```
class Синф_номи{
    var $хусусият_номи;
    /*хусусиятлар рўйхати*/
    function метод_номи(){
        /* усулларнинг танаси */
    }
    /*усуллар рўйхати*/
}
```

Маълумки, PHP билан ишлаш даврий ҳолатда HTML режимида юкланиши мумкин. Бу ҳолда дастур бир неча коднинг бўлаклари

(блоклар)дан ташкил топади. Синфни ифодалаш php-коднинг ҳар хил блоклари бўйича ва қолаверса ҳар хил файллар бўйича тарқатилмаслиги керак.

Ўзгарувчиларни инициаллаштириш. Баъзан айрим синф атрибутларига маънони синф иштирокчисини тузиш биланок ўзлаштириш керак бўлади. Синф атрибутларининг бошланғич маъноларини бериш учун махсус иккита стандарт усул мавжуд. PHP4 да маънони var оператори ёки конструктор функцияси ёрдамида инициаллаштириш мумкин. var ёрдамида фақат констант маъноларни инициаллаштириш мумкин. Констант бўлмаган маъноларни бериш учун объект синфдан ажраб чиққанда ўз-ўзидан ишга тушувчи конструктор функциясидан фойдаланилади. Конструктор-функция у ифодаланган бутун синфга мос келувчи номга эга бўлиши керак.

Объектлар. Синф – бу объект типдаги маълумотларнинг бир туридаги ифодаланишидир. Синфлар реал ўзгарувчилар учун шаблон вазифасини ўтайди. Керакли типдаги ўзгартувчи new оператори ёрдамида синфдан тузилади. Объектни тузиб, биз барча усулларни қўллашимиз ва барча синф ифодасида кўрсатиб ўтилган хусусиятларни олишимиз мумкин бўлади. Бунинг учун қуйидагича синтаксисдан фойдаланилади: \$объект_номи->хусусият ёки

_номи\$объект_номи->усул_номланиши(аргументлар_рўйхати).

Хусусиятлар ёки усулар номлари олдидан \$ белгиси қўйилмайди.

Хусусиятлар маъносини ўрнатиш. Синфни тузиб, бу синфнинг объекти қандай номга эга бўлишини била олмаймиз, қолаверса объектлар жуда кўп бўлиши ва уларнинг барчаси ҳар хил номга эга бўлиши мумкин. Синфни юзага чиқариш ичида объектга қандай муносабатда бўлишни билмаймиз. Синф юзага чиқиши ичида функциялар ва ўзгарувчиларга эркин кириш учун, \$this ўриндош ўзгарувчисидан фойдаланиш керак. Масалан, \$this->title шундай синф объектнинг title ини қайтаради. Баъзан бу ўзгарувчини “менинг хусусий мулким” (хусусиятга муносабат тариқасида) деб ўқишни таклиф қилинади.

1.3. MySQL маълумотлар базасини бошқариш тизими

MySQL тилида 1.1-жадвалдаги асосий маълумотлар типлари ишлатилади.

1.1-жадвал. MySQL да маълумотлар типлари

TINYINT	Диапазони -128 дан 127 гача сонларни сақлаши мумкин;
SMALLINT	Диапазони -32768 дан 32767 гача сонларни сақлаши мумкин
MEDIUMINT	Диапазони -8 388 608 дан 8 388 607 гача сонларни сақлаши мумкин
INT	Диапазони -2 147 483 648 дан 2 147 483 647 гача сонларни сақлаши мумкин
BIGINT	Диапазони -9 223 372 036 854 775 808 дан 9 223 372 036 854 775 807 гача сонларни сақлаши мумкин
FLOAT	Аниқлиги унча катта бўлмаган сўзувчи нуқтали сон
DOUBLE	Иккилик аниқликка эга бўлган сўзувчи нуқтали сон
REAL	DOUBLE учун синоним
DECIMAL	Сатрлар шаклида сақланувчи каср сон
NUMERIC	DECIMAL учун синоним
VARCHAR	255 дан ортиқ бўлмаган символларни сақлаши мумкин
TINYTEXT	255 дан ортиқ бўлмаган символларни сақлаши мумкин
TEXT	65 535 дан ортиқ бўлмаган символларни сақлаши мумкин
MEDIUMTEXT	16 777 215 дан ортиқ бўлмаган символларни сақлаши мумкин
LONGTEXT	4 294 967 295 дан ортиқ бўлмаган символларни сақлаши мумкин
DATE	-махсус команда орқали аниқланувчи форматдаги сана (Sysbase да кўзда тутилган бўйича уу/мм/дд);
TIME	- махсус команда орқали аниқланувчи форматдаги вақт (кўзда тутилган бўйича hh.mm.ss).

Жадваллар билан ишлаш. Жадваллар CREATE TABLE командаси билан яратилади. Бу команда қаторларсиз бўш жадвал яратади. CREATE TABLE командаси жадвал номини ва жадвал ўзини маълум тартибда кўрсатилган устунлар номлари кетма - кетлиги таърифи кўринишида аниқлайди. У маълумотлар типлари ва устунлар ўлчовини аниқлайди. Ҳар бир жадвал жуда бўлмаганда битта устунга эга бўлиши керак[4].

CREATE TABLE командаси синтаксиси:

```
CREATE TABLE <table-name >  
( <column name> <data type>[(<size>)],  
  <column name> <data type>[(<size>)], ... );
```

Аргумент қиймати катталиги маълумот турига боғлиқдир. Агар сиз махсус кўрсатмасангиз, тизим автоматик қийматни ўрнатади.

Жадвални ўчириш имконига эга бўлиш учун, жадвал эгаси (яъни яратувчиси) бўлишингиз керак. Фақат бўш жадвални ўчириш мумкин. Қаторларга эга бўлган, тўлдирилган жадвални ўчириш мумкин эмас, яъни жадвал ўчиришдан олдин тозаланган бўлиши керак. Жадвални ўчириш командаси куйидаги кўринишга эга:

```
DROP TABLE < table name >;
```

Жадвални ўзгартириш учун ALTER TABLE командасидан фойдаланилади. Бу команда жадвалга янги устунлар кўшиш, устунларни ўчириш, устунлар катталигини ўзгартириш, ҳамда чекланишларни кўшиш ва олиб ташлаш имкониятларига эга. Бу команда ANSI стандарти қисми эмас, шунинг учун ҳар хил тизимларда ҳар хил имкониятларга эга.

Жадвалга устун кўшиш учун команданинг типик синтаксиси:

```
ALTER TABLE <table name> ADD <column name>  
  <data type> <size>;
```

MySQL операторлари. SELECT оператори МБ жадвалларидан натижавий тўплам олиш учун мўлжалланган ифодадир. Биз SELECT оператори ёрдамида сўров берамиз, у бўлса маълумотлар натижавий тўпламини қайтаради. Бу маълумотлар жадвал шаклида қайтарилади. Бу

жадвал кейинги SELECT оператори томонидан қайта ишланиши мумкин ва ҳоказо.

```
SELECT -- ALL ----- schema , ustun ----
      -- DISTINCT -- ---- * -----
FROM   -- schema , Jadval .. -----
WHERE  -- izlash sharti -----
GROUP BY -- schema , ustun -----
HAVING -- izlash sharti -----
ORDER BY – tartiblash spetsifikatori -----
```

Биринчи қоида, SELECT ифодаси ўз ичига албатта FROM ифодасини олиши керак. Қолган ифодалар керак бўлса ишлатилади. SELECT ифодасидан сўнг сўровда қайтарилувчи устунлар рўйхати ёзилади.

FROM ифодасидан сўнг сўровни бажариш учун жадваллар номи ёзилади.

WHERE ифодасидан сўнг агар маълум сатрларни қайтариш лозим бўлса, излаш шarti ёзилади.

GROUP BY ифодаси гуруҳларга ажратилган натижавий сўров яратишга имкон беради.

HAVING ифодасидан гуруҳларни қайтариш шarti ёзилади ва GROUP BY билан бирга ишлатилади.

ORDER BY ифодаси маълумотлар натижавий тўпламини тартиблаш йўналишини аниқлайди.

Жадвалга маълумот қўшиш. Ҳамма сатрлар MySQL да INSERT командаси ёрдамида киритилади. INSERT қуйидаги форматга эга:

```
INSERT INTO <table name | view name> [(column [,column] ...)]
VALUES ( <value> [,<value>] ... );
```

Сатрларни ўчириш. Сатрларни жадвалдан DELETE командаси билан ўчириш мумкин. У алоҳида қийматларни эмас фақат сатрларни ўчиради. DELETE қуйидаги форматга эга:

```
DELETE FROM <table name | view name>
```

[WHERE search-condition];

Майдон қийматларини ўзгартириш. Бу ўзгартириш UPDATE командаси ёрдамида бажарилади. Бу командада UPDATE ифодасидан сўнг жадвал номи ва SET ифодасидан сўнг маълум устун учун ўзгартириш кўрсатилади. UPDATE икки форматга эга. Улардан биринчиси:

```
UPDATE <table name | view name>
```

```
SET column = expression [,column = expression] ...
```

```
[WHERE search-condition]
```

бу ерда expression - бу устун | ифода | константа | ўзгарувчи

Иккинчи вариант:

```
UPDATE <table name>
```

```
SET column = expression, ...
```

```
[ FROM table-list ]
```

```
[ WHERE search-condition ]
```

Мантиқий операторлар. BETWEEN ифодаси бу қийматлар диапазонига тегишлиликни текширишдир. Ифода синтаксиси қуйидагича:

-- текширилаётган ифода -- BETWEEN -- қуйи ифода AND юқори ифода

Ифода IN тўпلامга тегишлиликни текширади. Команда синтаксиси қуйидагича:

--- текширилаётган ифода --- IN --- (--- const---)

Бирламчи калитлар чекланишлари. SQL бирламчи калитларни тўғридан тўғри бирламчи калит (PRIMARY KEY) чекланиши орқали таърифлайди. PRIMARY KEY жадвални ёки устунларни чеклаши мумкин. Бу чекланиш UNIQUE чекланиши каби ишлайди, фақат жадвал учун фақат битта бирламчи калит (ихтиёрий сондаги устунлар учун) аниқланиши мумкин бўлган ҳолдан таққари. Бирламчи калитлар NULL қийматга эга бўлиши мумкин эмас.

Мисол:

```
CREATE TABLE `fakultet`
```

```
(id int(11) NOT NULL PRIMARY KEY,  
name varchar (255),  
name_long varchar (255),  
dean varchar(255));
```

PRIMARY KEY чекланиши қийматлар уникал комбинатсиясини ташкил қилувчи бир неча майдонлар учун қўлланиши мумкин. Масалан PRIMARY KEY чекланишини жуфтликлар учун қўллаш мумкин:

```
CREATE TABLE `fakultet`  
(id int(11) NOT NULL,  
name varchar (255) NOT NULL,  
name_long varchar (255),  
dean varchar(255),  
PRIMARY KEY(name,id));
```

Майдон қийматларини текшириш (CHECK чекланиши). CHECK чекланиши жадвалга киритилаётган маълумот қабул қилинишидан олдин мос келиши лозим бўлган шарт киритишга имкон беради. CHECK чекланиши CHECK калит сўзи кўрсатилган майдондан фойдаланувчи предикат ифодадан иборатдир[4,10].

CHECK чекланишидан майдонга маълум қийматларини киритишдан ҳимоя қилиб, хатолар олдини олиш учун фойдаланиш мумкин.

2. БОБ. WEB САҲИФАЛАРНИ ЛОЙИҲАЛАШ

2.1. Web-саҳифаларни лойиҳалаш воситалари

Қайд қилинган ўлчамли Web-саҳифани ишлаб чиқишда унинг учун экран ўлчамини танлашга тўғри келади. Бунда саҳифани энг кўп фойдаланувчилар эркин фойдаланадиган (ва тўғри акс эттириладиган) қилиб яратиш керак.

Саҳифаларни кўриш жараёнида горизонтал айлантириш (прокрутка)ни ишлатишга тўғри келганлиги учун кўпчилик дизайнерлар саҳифаларни 640x480 форматида ишлаб чиқишни тавсия қиладилар. Горизонтал айлантириш ҳамма вақт ўзлаштиришни қийинлаштиради, шунинг учун дизайнерлар анъанага кўра уни рад этадилар.

Ҳозирги кунга келиб кўпчилик ишлаб чиқувчилар 800x600 ни стандарт ажрата олиш деб ҳисобляпти. Бундан ҳам юқориқ ажрата олишларга мўлжалланган саҳифаларни эса жуда камчилик ишлаб чиқадиладар. Албатта, сизнинг қарорингиз биринчи навбатда аудиторияга боғлиқ бўлади. Масалан, агар сайтнинг ресурслари графика дизайнерлари учун мўлжалланган бўлса, улар энг камида 800x600 ажрата олишли дисплейларга эга деб ҳисоблаймиз ва шунга мувофиқ саҳифа ишлаб чиқиладди.

Эътиборга лойиқ Web-дизайн чекланган имкониятли, хусусан, кўриш ва эшитишда қийналадиган фойдаланувчилар эркин фойдаланиши мумкин бўлган саҳифаларни ишлаб чиқишни ўз ичига олади. World Wide Web консорциуми Web дан барча фойдаланувчиларнинг янада эркин фойдаланишини мақсад қилиб қўйган Web Accessibility Initiative (WAI) ташаббусини эълон қилди. Бироқ ушбу ташаббуснинг муваффақияти унда қўйилган вазифаларга мувофиқ Web-сайтларни ярата оладиган (ёки ярата олмайдиган) оддий ишлаб чиқувчиларнинг иштирок этишига боғлиқ.

HTML 4.0 воситалари. HTML 4.0 спецификацияси Web-хужжатларни анча кенг доирадаги фойдаланувчилар эркин фойдаланадиган қилиш учун махсус яратилган бир қатор янги артибутлар ва тегларни ўз ичига олади.

HTML 4.0 нинг баъзи янги имкониятларини қисқача санаб ўтамиз. (<http://www.w3.org/WAI/References/HTML4-access> сайтида имкониятларнинг кенгайтирилган рўйхати, <http://www.w3.org/TR/REC-html40> сайтида эса ушбу версиянинг тўлиқ спецификациялари жойлашган.) HTML 4.0 эркин фойдаланишни таъминлайдиган қуйидаги янги имкониятларни таклиф этади:

- кейинчалик ҳужжат тузилиши ва унинг ташқи кўринишини янада бўлиш. HTML 4.0 стили тўғрисидаги ахборотни каскадли стиллар жадвалларида жойлаштиришни таклиф этади;
- навигация ёрдами, масалан, эркин фойдаланиш клавишлари ва фақат клавиатурани ишлатиш билан саҳифа элементларидан эркин фойдаланиш учун табуляция тартибини индексация қилиш;
- график ва матнли мурожаатларни бирлаштирадиган янги миждоз карта-тасвирига тегишли тавсиялар;
- нутқ ва бошқа қурилмаларга қисқартма ва акронимларни талқин қилишга ёрдам берадиган `<abbr>` ва `<acronym>` янги теглари;
- жадвалларнинг қаторлари ва устунларини мантиқан гуруҳлашнинг имконияти, жадвалларни талқин қилишни осонлаштириб, уларни сарлавҳалар, резюме ва ичидаги нарсанинг узун тавсифлари билан таъминлаш;
- формаларни бошқариш элементларини гуруҳлаш ва ўзлаштириш учун анча равшан бўлган узун танлов рўйхатларини яратиш имконияти. Формалар элементларидан, шунингдек табуляция ва тезкор эркин фойдаланиш клавишлари орқали эркин ҳаракат қилиш;
- муқобил матнни яратишнинг такомиллашган механизми. Энди `alt` атрибути `` теги учун мажбурийдир. Тасвирларга анча узун матнли изоҳлар билан алоқани таъминлаш учун `longdesc` атрибути киритилган.

Ихтиёрий элемент тўғрисида ахборот қўшиш учун `title` атрибутини ишлатиш мумкин.

CSS воситалари. Каскадли стиллар жадваллари ёки CSS (инглизча Cascading Style Sheets дан олинган) HTML ни янада ривожланишининг

ҳисобланади ҳисобланади ва бизга ахборот тақдим этишнинг кейинги поғонасига ўтишга имкон беради. Стиллар жадваллари саҳифанинг ичидаги нарсани мазмунан бўлишга ва унинг безатилишига имкон беради.

Янги стандартда HTML концепциясининг манбаига қайтишга ҳаракат қилиб кўрилди. Тўртинчи версия биринчи версия каби саҳифаларни шундай қилиб яратишни тавсия этадики, улар ихтиёрий қурилмада: акс этсин – хоҳ у 21" дисплейда, хоҳ уяли телефоннинг кичкина оқ-қора экранда акс этсин.

CSS2 (Cascading Style Sheets, Level 2) – каскадли стиллар жадваллари бўйича энг сўнгги тавсия, саҳифаларни график ва визуал бўлмаган қурилмалар томонидан яхшироқ талқин қилиниши учун механизмларни тақдим этади. Такмиллаштиришлар қуйидагичадир:

- фойдаланувчи яратган стиллар жадвали- каскаддаги анча юқори поғонадаги барча стиллар жадвалларини алмаштириши мумкин бўлган механизмлар. Бу сўнгги фойдаланувчига акс эттиришни тўлиқ бошқаришга имкон беради. Фойдаланувчи махсус талабларга мувофиқ саҳифаларни чиқариш учун созланадиган стиллар жадвалларини яратиш имкониятини олади;

- юкланаётган шрифтлар учун махсуслаштирилган кўллаб-қувватлаш – шу тариқа саҳифанинг ташқи кўринишини яхшилаш учун матнни графикага жойлаштириш тенденцияси камаяди;

- ичидаги нарсани ташқи кўринишдан ажратадиган вазиятни белгилаш ва текислаш механизмлари. Бу стиллар жадваллари HTML тегларидан акс эттиришнинг алоҳида эффектларини яратиш учун нотўғри фойдаланишни бартараф этиши керак. HTML тегларидан ҳужжатни мантиқан тузилишини белгилашда фойдаланиш мумкин ва бу билан уни визуал бўлмаган воситачиларнинг талқин қилиши учун анча оддийроқ қилинади.

- Web бўйича етказилган ахборотни овозли чиқариш учун бошқариш воситалари;

- Мўлжални тўғри олиш мақсадида ҳужжатга қўшиш мумкин бўлган рақамли маркерлар каби яхшиланган навигация воситалари.

2.2. Web-саҳифаларда матн ва графикани тақдим этиш

Web учун профессионал графикани яратишда четлари текисланган матндан фойдаланилади. Текислаш нотекис четлардаги енгилгина “суркалиш”лар бўлиб, ранглар ўртасидаги ўтишларни текислайди. Текисланмаган четлар, аксинча, кертик-мертик ва поғонали кўринади. Жуда кичик ўлчамдаги (10 пункт ёки ундан кам) матн бу умумий қоидадан истиснодир, чунки текислашни қўллаш уни деярли фарқлаб бўлмайдиган қилади. Кичик ўлчамлардаги матн силлиқлашсиз анча яхши кўринади.

Икки комплектдаги шрифтлар. Web-саҳифани базавий HTML воситалари билан ишлаб чиқишда иккита комплектдаги шрифтлар мавжуд: мутаносиб ва белгиланган кенгликдаги шрифт. Муаммо шундан иборатки, акс эттиришда уларнинг қайси бири ва қайси ўлчами ишлатилиши номаълум.

Мутаносиб шрифт, бошқача қилиб айтганда «ўзгарувчан кенгликдаги шрифт» ҳар бир белги учун унинг чизилишига қараб турли миқдордаги жойни ажратади. Масалан, мутаносиб шрифтда бош «W» ҳарфи горизонтал бўйича катта «I» ҳарфига қараганда кўп жой эгаллайди. Times, Helvetica ва Arial каби гарнитуралар мутаносиб шрифтларнинг намунаси ҳисобланади.

Web-браузерлар Web-саҳифадаги аксарият матнлар учун, шу жумладан, асосий матн, сарлавҳалар, рўйхатлар, цитаталар ва бошқалар учун мутаносиб шрифтларни ишлатади. Одатда, асосий матннинг катта бўлаклари мутаносиб шрифтларда босмадан чиқарилган бўлса, уларни ўқиш қулайроқдир. Кўпчилик фойдаланувчилар “сукут” бўйича (автоматик тарзда) ўрнатилган шрифтларни алмаштиришга вақтлари бўлмаганлиги сабабли, сизнинг саҳифангиздаги матн 10 ёки 12 ўлчамдаги Times (Netscape) ёки Helvetica (Microsoft Internet Explorer) шрифти билан акс этишини жуда катта эҳтимол билан кутиш мумкин. Бу фақатгина умумий қоидадир, холос.

Белгиланган кенгликдаги шрифт шрифтнинг барча белгилари учун бир хил жой тақдим этади. Бош «W» ҳарфи бош «I» ҳарфидан кўп жой эгалламайди. Courier ва Monaco каби гарнитуралари белгиланган кенгликдаги шрифтлар ҳисобланади. Web-браузерларда белгиланган кенгликдаги шрифтлардан қуйидаги HTML-теглар: <pre>, <tt>, <code>, <kbd>, <samp>, <xtp> ичидаги ихтиёрий матнни акс эттириш учун фойдаланади

Кўпчилик сукут бўйича ўрнатилган шрифтларни созлашни ўзгартирмаганлиги сабабли, кўрсатилган тегларда жойлашган матн Courier типидagi шрифтларнинг бирида чиқарилади.

Тасвирлардаги матн. Дизайнерлар шрифтлар устидан абсолют назоратни ўрнатишнинг энг тўғри усули – матнни тасвирга жойлаштириш эканлигини тез тушундилар. GIF файллари кўринишида бажарилган сарлавҳалар, кичик сарлавҳалар ва эълонларни тез-тез кўриш мумкин. Кўпгина Web-саҳифалар фақат графикада тақдим этилган бўлиб, саҳифанинг бутун матнини ўз ичига олади. HTML-матннинг ўрнига графикадан фойдаланишнинг афзалликлари шубҳасиз аниқдир:

- шрифтнинг тури, ўлчами, интерлиньяж, ҳарфлар ўртасидаги оралик, ранг ва текислашни, яъни фақат HTML да қийинчилик туғдирадиган барча атрибутларни аниқлаш мумкин;
- сизнинг саҳифангиз барча графика браузерларида чиқаришда бир хил бўлади.

Лекин бу усулнинг ҳам қатор камчиликлар мавжуд:

- тасвир матнга қараганда узоқроқ юкланади, чунки графика файллари одатда HTML-матнларга қараганда анча катта бўлади;
- нографика браузерларида ичидаги нарса йўқотилади. Графикани кўра олмайдиган (ёки кўришни хоҳламаётган) фойдаланувчилар матнни ҳам кўришмайди. Графика тасвирининг ўрнидаги муқобил матн (Alt атрибутидан фойдаланилади) ёрдам беради, лекин унинг имкониятлари чегараланган ва бу

хар доим ҳам мавжуд бўлмаган графика ахборотини тенглаштиришнинг ишончли усули бўлмайди;

- тасвирда жойлашган ахборотни индексациялаш ёки уни излашни ташкил қилиш мумкин эмас. Натижада ҳужжатдан ахборотнинг муҳим қисмлари чиқариб ташланади.

Шрифт ўлчами. Одатда, шрифтнинг ўлчами пунктларда белгиланади (72 пункт (пт) = шрифт баландлигининг 1 дюйми), лекин афсуски, бу ўлчамлар платформалар ўртасида етарлича аниқ ўтказилмайди. Бу қисман уларнинг операцион тизимлари турли ажрата олишли дисплейларни бошқариши сабабли рўй беради. Одатда, Windows – 96 нукта/дюйм, MacOS – 72 нукта/дюйм экраннинг ажрата олишидан фойдаланади. MultiScan мониторларининг ажрата олиши анча юқорирок бўлади.

Web-саҳифаларда графикани тақдим этишда ҳозирги пайтда Web даги барча тасвирлар икки форматда: GIF ва JPEG форматларида тақдим этилган. Эслатиб ўтиш ўринли бўлган учинчи рақиб, PNG формати, браузерларнинг кўллаб-қувватлаши ва эътибори учун курашмоқда.

GIF. GIF – Graphics Interchange Format ни Web нинг анъанавий формати деб аташ мумкин. У Web-браузерлар кўллаб-қувватлаган файлларнинг биринчи формати бўлган ва ҳозирги кунгача ҳам Web нинг асосий графика формати бўлиб келмоқда. Унинг хусусиятлари қуйидагилардан иборат:

- 256 дан ортиқ бўлмаган рангни кўллаб-қувватлайди (кам бўлиши мумкин ва кўпинча шундай ҳам бўлиши керак);
- ранглар палитрасидан фойдаланади;
- LZW усули бўйича ахборотни йўқотишсиз зичлашдан фойдаланади (бу усул PKZIP архиваторида қўлланиладиган зичлашга ўхшаш ва бинобарин, GIF-файллар кейинчалик яна зичланмайди);
- қаторлараро ёйишни кўллаб-қувватлайди;
- оқимли формат ҳисобланади, яъни расмни кўрсатиш уни ола бошлаш вақтида бошланади;

- палитрадаги рангларнинг бирига шаффоф атрибутини белгилашга имкон беради, бу шаффоф деб аталадиган GIF ларни яратишда қўлланилади;
- бир файлда бир неча тасвирларни сақлаш имкониятига эга, бу анимация қилинган GIF ларни тайёрлашда қўлланилмоқда;
- файлга изоҳ қўйиш, тасвирларни кўрсатиш орасида ушланиб қолишни бажариш ва бошқаларга имкон берадиган файлга бошқарувчи блокларни қўйиш имкониятини қўллаб-қувватлайди.

JPEG. Web даги иккинчи энг машхур графика формати JPEG – Joint Photographic Experts Group ҳисобланади. У ранг тўғрисида 24-разрядли ахборотга эга бўлади. Бу GIF форматидаги 256 рангдан фарқли ўлароқ 16,77 млн. ранг деганидир. JPEG да йўқотишли зичлашдан фойдаланилади. Бу шуни англатадики, зичлаш жараёнида тасвир тўғрисидаги баъзи ахборот чиқариб юборилади, лекин, аксарият ҳолларда, тасвир сифатининг ёмонлашиши зиён келтирмайди ва ҳаттоки кўпинча сезилмайди ҳам.

Ранглари уйғун ҳолда бир-бирига ўтадиган фотосурат ёки ихтиёрий тасвирларни JPEG-форматида сақлаган яхши, чунки у кичик ҳажмдаги файлга сиғадиган тасвирнинг анча юқори сифатини таклиф этади. Шунга қарамасдан, JPEG бир хил рангдаги графика тасвирлари учун энг яхши қарор ҳисобланмайди, чунки бу формат рангларда хол-хол доғлар ҳосил қилади ва сўнгги файл, одатда, ўшандай тасвир учун GIF-файлга қараганда бироз катта бўлади.

PNG. Web да доимий ишлатиш учун рақобатлашаётган учинчи графика формати ҳам мавжуд. Бу баъзи афзалликларига қарамасдан 1994 йилдан бери у ёки бу жиҳатдан унча кўп фойдаланилмаётган PNG – Portable Network Graphic форматидир. Браузерлар ичига ўрнатилган графика сифатида PNG форматини эндигина қўллаб-қувватлай бошлади, бироқ PNG Web да оммавий формат бўлиш учун барча имкониятларга эга. Айнан шунинг учун ҳам у бу ерда «катта учлик»ка киритилган. PNG йўқотишсиз зичлаш схемасидан фойдаланган ҳолда 8-разрядли индексация қилинган рангларни, 16-разрядли

ярим тонларни ёки 24-разрядли тўла рангли тасвирларни кўллаб-қувватлаши мумкин. Бу тасвирнинг анча юқори сифатини, баъзан эса GIF формати билан таққослаганда кичикроқ ҳажмдаги файлни таъминлайди. Бундан ташқари, PNG файллари бир неча ажойиб функцияларга эга, масалан, гамма коэффицентини бошқариш ва шаффофликнинг ўзгартириладиган даражалари (бу фон расмини тарқоқ майин соялар орқали кўрсатишга имкон беради).

Тасвирни 72 ppi ажрата олиш билан яратиш қулайдир (экранда тақдим этиш учун бу энг яхши вариантдир), бунда пикселлардаги умумий ўлчамларга эътибор бериш керак. Web да графика яратиш жараёнида дюймларни умуман ишлатмаслик мумкин. Тасвирнинг саҳифадаги бошқа тасвирлар билан таққосланган ўлчами ва браузер ойнасининг умумий ўлчами муҳимдир. Масалан, кўпчилик фойдаланувчилар ҳамон 640x480 пикселдаги ажрата олишли 14-дюймли дисплейларни ишлатади. Экраннинг барча жойини графика кўйилмаси билан тўлиқ тўлдириш учун унинг кенглигини 600 пикселдан ошмайдиган қилиш керак (бунда ўнг ва чап томондаги пикселларнинг бир қисми ойна учун ва айлангириш йўли учун ишлатилиши эътиборга олинади). Саҳифадаги бошқа тугмачалар ва тасвирларнинг ўлчамини 600 пикселли кенгликка эга банерга нисбатан пикселларда ўлчаш керак.

Ушбу муносабатда Web-дизайнер учун ягона энг муҳим қоида бор: графика тасвири файлининг ўлчами имкон қадар кичик бўлиши керак! Тармоқ бўйича узатишга мўлжалланган тасвирларни яратиш ишлаб чиқувчилар учун юклаш вақти муаммосига жиддий ёндашиш масъулиятини юклайди.

2.3. Сайт яратишнинг асосий қоидалари ва босқичлари

Браузерларни танлаш ва янги технологиялардан фойдаланиш тўғрисида тўғри қарор қабул қилиш кўп ҳолларда фойдаланувчиларнинг танланган аудиториясига боғлиқ бўлади. Янги сайт ишлаб чиқишдан аввал тадқиқотлар

Ўтказиб, қандай платформалар, браузерлар, техникавий янгиликлар афзал кўрилаяпти, фойдаланувчилар уланишнинг қандай тезликларидан фойдаланаётганлигини аниқлаш зарур. Агар мавжуд сайт қайта ишланаётган бўлса, олдин серверда рўйхатга олинган ёзувлар, сайтдан фойдаланиш тўғрисида сақланаётган ахборот таҳлил қилинади.

Кўп мақсадли сайтларнинг аксарияти учун «фарқларни ажратиш» ёндашувидан фойдаланиш маъқулдир, агар ресурслар имкон берса, бир нечта версияларни яратинг ва уларга тегишлича хизмат кўрсатинг.

Ойнанинг энг кичик иш макони браузернинг барча кўринадиган асбоблари (тугмачалар, жойлашиш панеллари ва айлантириш полосаси кабилар) билан ўлчанади. Ойнанинг энг катта иш макони барча мажбурий бўлмаган элементларни ёпган ҳолда ўлчанади, бу браузер ойнасининг ушбу ажрата олиш учун энг катта бўлишини таъминлайди.

Лекин бу фақат иккита энг чекка ҳолатлардир. Амалиётда браузер ойнасининг ўлчамлари ўзгариб туради. Тугмачаларнинг бир қисми экранда кўриниб туради, лекин ҳаммаси эмас. Айлантириш полосалари автоматик тарзда уланиб ва ўчириб турилади, шунинг учун уларни ҳисобга олиш қийин. Ажрата олиши жуда юқори бўлган (1024 пиксел ва ундан юқори) дисплейдан фойдаланувчилар ҳар доим ойнани тўла очадилар, кўпинча эса бир вақтнинг ўзида бир нечта ойнани очадилар. Булар билан бир қаторда яна кўплаб маълум бўлмаган омиллар борки, улар ҳам браузер ойнаси ўлчамига таъсир қилиши мумкин.

Дисплей ранглариининг ўзига хос хусусиятлари. Мониторлар, агар улар рангли бўлса, тасвирлаб берадиган ранглари бўйича фарқланади. Бу ишлаб чиқувчининг қарорига таъсир қиладиган яна битта омил. Мониторлар одатда 24-разрядли (16,77 млн. ранг), 16-разрядли (65 536 ранг) ёки 8-разрядли (256 ранг) бўлади.

«Ҳақиқий», рангларнинг 24-разрядли маконида олинган ранглар 8-разрядли дисплейларда браузерлар билан ишлов берилганда бузилади (доғлар ва майда холлар билан чиқарилади).

Лекин 216 рангдан иборат тўплам мавжуд бўлиб, у MacOS ва Windows тизимий палитралари рангларидан тузилган ва 8-разрядли дисплейларда бузилмайди. Бу ранглар тўпламининг номларидан бири – Web-палитра (Web Palette). Кўпчилик дизайнерлар саҳифа барча фойдаланувчилар учун бирдек кўриниши учун Web-графики ва HTML элементларини яратишда шу палитрадан фойдаланишни афзал биладилар.

Web-саҳифанинг аниқ ўлчамини аниқлашдаги биринчи қадам дисплей таъминлаб берадиган энг катта маконни аниқлаш бўлиши зарур. Компьютер дисплейлари турли стандарт ўлчамларга эга бўлади ва, одатда, дюймларда ўлчанади. Дисплейларнинг баъзи бир типик ўлчамларини келтириш мумкин – 14", 15", 17", 19" ва 21".

Дисплейнинг ажрата олиши – экрандаги пикселларнинг умумий сони (picture's element – сурат элементи) анча муҳим тавсиф ҳисобланади. Ажрата олиши қанчалик юқори бўлса, тасвир шунча муфассал ва аниқ кўриниши мумкин. Пикселларнинг сонини билган ҳолда унга мувофиқ равишда тасвир (пикселларда ўлчанадиган) ва саҳифа элементларини яратиш мумкин.

Шуни ёдда тутиш муҳимки, ушбу дисплейда ажрата олиш қанчалик юқори бўлса, экраннинг мавжуд маконида шунча кўп пикселлар прессланган бўлади. Натижада пикселлар камроқ бўлиб қолади ва тегишлича саҳифанинг бошқа элементлари ва тасвири ҳам кичиклашади.

Айнан ана шу сабабга кўра Web да ўлчовлар дюймларда эмас, балки пикселларда ўлчанади. Сизнинг тизимингизда дюймларда берилган катталиқ бошқа фойдаланувчиларнинг дисплейларида камроқ ёки кўпроқ кўриниши мумкин. Сиз пикселлар билан ишлаётганингизда элементлар ўлчамининг бир-бирига нисбати қандайлигини биласиз.

Эндиликда Web нафақат шахсий компьютерлар учун мўлжалланибгина қолмай, Web-браузерлар WebTV, чўнтак PDA-қурилмалари ва ҳатто уяли телефонлар шаклида яшаш хоналари, портфеллар ва машиналарга тобора дадил кириб бормоқда. Дизайнерларнинг янги ташвишлари улар дисплейларининг ўлчамлари жуда кичиклиги билан боғлиқдир.

Чўнтак қурилмалари. Web нинг янада кўпроқ оммалашуви алоқанинг чўнтак компьютерлари, PDA ва уяли телефонлар каби чўнтак қурилмаларидан фойдаланишнинг тобора кенгайиб бориши билан биргаликда шунга олиб келдики, Web-браузерлар жуда кичик ўлчамларгача зичлашди. Бундай қурилмаларнинг аксарияти сайт мазмунини акс эттириш учун «юпқа» мижоз (thin client) – мижоз-терминал каби браузерлардан фойдаланади. Браузернинг бу тури процессорга жуда кам талаблар қўйган ҳолда мижоз томонида ишлатилади ва ишларнинг асосий ҳажмини бажаришни серверларга қолдиради.

Ўзгарувчан саҳифалар (ўлчами ўзгаради ва ойнанинг турли ўлчамларига соزلанади) ва ўлчами қайдланган (ишлаб чиқувчига саҳифалар ўлчамини яхшироқ бошқариш имконини беради) саҳифалар ўртасида улардан бирини танлаш масаласи дисплейнинг ажрата олиш хусусиятлари билан боғлиқ. Ҳар бир ёндашув фойдасига анча салмоқли далиллар мавжуд. Табиийки, уларнинг ҳар бирига қарши чиқиш ва ёқлаш учун яхшигина сабабларни топиш мумкин.

Мослашувчан саҳифаларни ишлаб чиқиш. Web-саҳифалар сукут бўйича мослашувчан қилиб ишлаб чиқилади. HTML-файл матни ва элементлари браузер ойнасига тушади, бунда улар, дисплейнинг ўлчамидан қатъи назар, бутун маконни тўлдиради. Агар браузер ойнасининг ўлчами ўзгарса, элементлар янги ўлчамларга созлаш учун бошқатдан киритилади. Web нинг моҳияти мана шунда намоён бўлади. Кўплаб дизайнерлар онгли равишда саҳифаларни улар Web-ойна кенгайишлари ва зичланишларига бардош бериши учун шундай қилиб ишлаб чиқадилар. Бу ёндашув ўзининг афзалликлари ва камчиликларига эга.

Афзалликлари:

- Web-саҳифалар ажрата олиши турлича бўлган дисплейларда акс эттирилади, мослашувчан саҳифани ҳар қандай дисплейга чиқариш учун созлаш мумкин;

- дисплейнинг бутун макони тўлдирилган, аксарият ҳолларда ўлчами қайдланган саҳифаларни ишлаб чиқувчилар томонидан режалаштириладиган бўлмаслиги маъқул бўлган очик жойлар йўқ;

- мослашувчан саҳифалар дизайни ўз табиатига кўра ҳеч қандай қийинчилик туғдирмайдиган бўлиб, «олтин ўрталик»ка анча яқин туради. Бундай стандартларга мувофиқ фойдаланувчиларнинг аксарияти учун қулай бўлган саҳифа «яхши» деб ҳисобланади.

Камчиликлари:

- катта дисплейларда қатор узунлиги ҳаддан ортиқ бўлиши мумкин, бунда матн браузер ойнасининг бутун эини тўлдиради. Узун қаторлар экрандан ўқиш учун, айниқса, ноқулайдир, шунинг учун ойна ёки фреймнинг бутун эни матн билан тўлдирилганда, кўп фойдаланувчилар учун ўқиш шароитлари анча ёмонлашади;

- катта дисплейларда элементлар экранда етарли даражада уйғун жойлашган бўлади, кичик дисплейларда улар зич бўлади;

- мослашувчан дизайн натижаларини олдиндан айтиш қийин ва саҳифалар турли фойдаланувчиларда турлича бўлади.

Ўлчами қайдланган саҳифаларни ишлаб чиқиш. Кимда-ким саҳифаларни белгилаш устидан кўпроқ назорат ўрнатишни истаса, эни қайдланган саҳифалар ишлаб чиқиш керак бўлади, дисплей ўлчами ёки ойна ўлчамлари ўзгаришларидан қатъи назар у, барча фойдаланувчилар учун доимий бўлади. Бу ёндашув ноширлик ишида саҳифалар яратиш принципларига асосланган бўлиб, бу принциплар доимий тўрни, саҳифада жойлашган элементларнинг нисбатини ва қаторларнинг қулай узунлигини сақлаб туриш кабилардан иборатдир.

Афзалликлари:

- кенглиги қайдланган саҳифа ва устунлар узун қаторларни яхшироқ бошқаришни таъминлайди. Катта дисплейларда кўришда қаторлар жуда ҳам узун бўлиб кетмаслиги учун жадваллардан фойдаланиш мумкин.

Камчиликлари:

- агар браузернинг ушбу ойнасининг ўлчами саҳифа тўридан кам бўлса, саҳифанинг баъзи қисмлари кўринмай қолади ва бўйламасига айлантириш талаб этилиши мумкин. У деярли ҳар доим ишни қийинлаштирувчи тўсиқ сифатида қабул қилинади, шунинг учун ҳам аксарият ҳолларда ундан қочиш зарур. Ечимлардан бири – кўпчиликка тўғри келадиган саҳифа ўлчамини танлаш;

- илгаригидек браузерларда чиқариладиган символлар ўлчамини назорат қилиш мураккаб иш, элементлар ишлаб чиқиш вақтида фойдаланилган ўлчамга нисбатан каттароқ ёки кичикроқ ўлчамдан фойдаланиш натижасида олдиндан айтиб бўлмайдиган даражада сурилиб кетиши мумкин;

- саҳифа тасвирини тўлалигича назорат қилишга интилиш бу муҳитга қарши чиқиш, демакдир. Web – бу босма нашр эмас, унинг ўз кучли томонлари ва хусусиятлари бор. Мослашувчан дизайн стратегияси ҳимоячилари Web да қайдланган дизайнга ўрин йўқ, деб айтадилар.

Албатта, Web-саҳифалар тўлалигича қайдланган ёки мослашувчан бўлиши шарт эмас. Ҳар иккала ёндашувни бирлаштирувчи саҳифалар ишлаб чиқиш мумкин.

Битта барчага маълум бўлган усул шундан иборатки, жадваллардан фойдаланиб қайдланган саҳифа белгилари яратилади, сўнгра у саҳифада шундай марказлаштириладики, катта дисплейларга чиқарилганда янада мувозанатлашган ҳолатда кўринади (ўнг томондан экран худди бўшдек кўринмайди). Бу усулдан фойдаланишда жадвални фон тасвирига нисбатан аниқ жойлаштириш мумкин эмас.

Иккинчи ёндашув жадваллар ёки фреймлар гуруҳидан фойдаланишдан иборат бўлиб, улар абсолют ва нисбий ўлчамларининг комбинациялари белгиланган устунлар (ёки фреймлар) дан таркиб топади. Бу ҳолда ойна ўлчами ўзгарганида, битта устун ёки фрейм олдинги кенглигини сақлаб қолади, шу вақтнинг ўзида бошқалари ўлчамини ўзгартиради ва бўш жойларни тўлдирди.

3. БОБ. “МАЪРИФАТ ЮЛДУЗИ ТАЪЛИМ МАРКАЗИ” ВЕБ-САЙТНИНГ ТАСНИФИ

3.1 Маълумотлар базасининг архитектураси

Бу сайтнинг маълумотлар базаси яратилиш жараёнида MySQL маълумотлар базасининг бошқариш тизимидан фойдаланилди. Яратилган маълумотлар базаси 15 та жадвалдан иборат. Улар қуйидагилар:

1. “tbl_q_yunalish” жадвалида йуналишлар хақида маълумотлар сақланади.



3.1- расм. «tbl_q_yunalish» жадвали

2. “tbl_page” жадвалида веб саҳифалардаги жойлашган маълумотлар сақланади.



3.2- расм. «tbl_q_page» жадвали

3. “tbl_q_fanlar” жадвалида фанлар хақида маълумотлар сақланади.



3.3- расм. «tbl_q_fanlar» жадвали

4. “tbl_q_test” жадвалида тестлар хакида маълумотлар сақланади.



3.4- расм. «tbl_q_test» жадвали

5. “tbl_menu” жадвалида меню ссылгалари сақланади.



3.5- расм. «tbl_menu» жадвали

6. “tbl_jurnalis” жадвалида журналистлар хакида маълумотлар сақланади.



3.6- расм. «tbl_jurnalis» жадвали

7. “tbl_q_answer” жадвалида тестларнинг жавоблари сақланади.



3.7- расм. «tbl_q_answer» жадвали

8. “tbl_comment” жадвалида комментарийлар сақланади.



3.8- расм. «tbl_comment» жадвали

9. “tbl_gallery” жадвалида расмлар сақланади.



3.9- расм. «tbl_gallery » жадвали

10. “tbl_news” жадвалида янги хабарлар сақланади.



3.10- расм. «tbl_news » жадвали

11. “tbl_profiles” жадвалида аккаунтлар хакида маълумот сақланади.



3.11- расм. «tbl_profiles» жадвали

12. “tbl_users” жадвалида фойдаланувчилар хакида маълумотлар сақланади.

marifat.tbl_users	
id	int(11)
username	varchar(20)
password	varchar(128)
email	varchar(128)
activkey	varchar(128)
createtime	int(10)
lastvisit	int(10)
superuser	int(1)
status	int(1)

3.12- расм. «tbl_users» жадвали

3.2. Веб-сайтнинг асосий модулларининг таснифи

Веб-сайт Yii framework асосида яратилган.

Ҳа бу юқори эффеќтли РНР фреймворк компонент структурасига асосланган йирик Веб иловаларни қайта ишловчи ҳисобланади.

Ҳа бу йирик Веб иловаларни қайта ишловчи компонент структурасига асосланган РНР фреймворқдир. У кодларни қайта ишлаш концепциясини максимал қулай имконини бериб веб қайта ишлаш жараёнини сезиларли тезлаштиради. Ҳа (Yii) деб уќилади) деганини оддий (easy), эффеќтив (efficient), кенгаювчи (extensible) маъноларини беради.

Талаблар

Ҳа тузулган веб иловаларни ишга тушуриш учун 5.1.0. версиядаги РНР ни кутарадиган веб сервер керак. Ҳа ни ишлатишни хоҳловчилар учун объектга йўналтирилган дастурлаш (ОЙД) конценциясини тугатиш жуда муҳим. Чунки Ҳа қатъий объектга йўналтирилган фреймворк.

Ҳа бу умумбелгиланган веб дастурлаш фреймворк бўлиб, уни хар қандай веб иловаларни қайта ишлашда қуллаш мумкин. Узини енгиллиги ва замонавий хэшлаш мавжудлиги учун Ҳа катта трафик оқимига эга булган порталлар, форумлар контентни бошқариш тизимлари (CMS), электрон

тижорат тизимлари ва бошқаларнинг иловаларини қайта ишлаш учун жуда мос келади.

ҮII ни бошқариш фреймворклар билан таққослаш.

ҮII бошқа кўпчилик PHP фреймворклар каби MVC фреймворкдир. ҮII ни бошқа фреймворклардан афзаллиги унинг эффективлиги кенг имкониятлари ва хужжатлар сифатлигидадир. ҮII бошқа мураккаб веб иловаларни қайта ишлашнинг барча талабларига жавоб берадиган фреймворк сифатида лойихалаштирадиган фреймворк, ҮII бирор лойихани ёрдамчи маҳсулоти ёки чет ечимларининг тўплами эмас. У веб иловаларни қайта ишлаш соҳасида муаммоларнинг катта тажрибаси натижаси ҳамда машхур веб фреймворк ва иловалар тадқиқоти натижасидир.

ҮII PHP веб илова структурасига асосланган юқори эффектли объектга йўналтирилган очиқ манбаси ҮII “Үее” деб эълон қилинади.

ҮII лойихаси 2008 йил 1 январда Prado структураси камчиликларини кўрсатиш учун яратилиши бошланди. Масалан, Prado мураккаб саҳифалар билан ишлашда секин ишлайди кўпгина бошқариш воситаларини созлаш қилиш қийин. ҮII да бу ишлар анча осон ва эффективроқ 2008 йил октябрда 10 ой орасида ривожланишдан кейин, ҮII нинг август версияси чиқарилади.

ҮII нинг сўнгги турғун версияси ҮII 1.1, сиз бу версияни узингизнинг ишлаб чиқарадиган веб иловаларингизда бемалол ишлатишингиз мумкин.

ҮII PHP нинг тез структурасидир. ҮII сизга зарур булган иловаларнигина юклайди. Бунда хэшлашнинг кучли томонларини мужассамлашган.

ҮII PHP нинг хавфсиз структураси хавфсизлик ҮII учун стандарт ҳисобланади. Бунга маҳсулотларни филтрлаш SQL инъенкциясини олдини олишни кундалик участкаси SCRIPTING лар киради.

ҮII PHP нинг профессионал структураси. ҮII сизга кўп марта ишлатиш кодекси ва софт ишлашга ёрдам беради. Бу MVC наъмуналаридан келиб, чиқиб мантиқ ва тасаввурни аниқ ажратади.

ҮІІ хусусиятлари

1. Маълумотлар базасига кириш объектлари
 - 3 jQuery билан интеграция
 - 4 Кириш формалари ва ратификация.
 - 5 Ajax- рухсат этилган виджетлар.
 - 6 Туғриланган билимни қуллаш.
 - 7 Тозалаш ва theming
 - 8 WSDL хизмати курсатиш мураккаб спецификаяли автоматик равишда веб хизмат талабини қайта ишлашни бошқариш.
 - 9 Интернационллаш ва локаллаштириш бу хабарлар таржимаси вақт ва сана форматлаш ва интерфейсни локаллаштириш
 - 10 Хэшлашнинг қатламли схемаси бу маълумотларни хэшлаш саҳефаларини сарогментларни хэшлаш динамик мазмун қуллайди маълумот тошувчисини хэшлаш ўзгартириши мумкин.
 - 11 Хатолар билан ишлаш хатолар қайта ишлашни тушинарли равишда ифодаланган маълумотлар руйхатидан ўтган ва филтрланган.
 - 12 Хавфсизлик тадбирлари қайтариш кундаланг ўчаскасига SCRIPTING (XSS) жойлашган.
 - 13 Функционал имконият ва бирлашмалар PHP UNION селенга асосланган.
 - 14 Йиғиштирилган сутга автоматик кодлаш авлоди.
 - 15 ҮІІ компоненталар ва сатр камандаси XHTML стандартига бўйсинади.
 - 16 ГРУШМ кодексини ҮІІ да ишлатиш (НОК) мумкин.
- Веб-сайт қўйидаги модуллардан ташкил топган:
- CSS модули сайтнинг стили ва дизайннинг кодларини ўз ичига олади

..				<Папка> 18.06.2014 16:15
form	css	1 613	21.06.2014 16:37	
lightbox	css	3 767	21.06.2014 16:37	
main	css	12 218	21.06.2014 16:37	
navi	css	3 167	21.06.2014 16:37	
slide	css	706	21.06.2014 16:37	
tables	css	0	21.06.2014 16:37	

3.13- расм. «css» модули

- Images модули сайтнинг расмларини ўз ичига олади

..				<Папка> 18.06
fullscreen				<Папка> 21.06
gallery				<Папка> 21.06
thumbnails				<Папка> 21.06
bullet	gif	49	21.06	
delete	gif	315	21.06	
down	gif	82	21.06	
edit	gif	559	21.06	
favicon	gif	1 182	21.06	
left	gif	498	21.06	
loading	gif	8 476	21.06	
right	gif	498	21.06	
up	gif	81	21.06	
1	jpg	78 401	21.06	
2	jpg	54 966	21.06	
3	jpg	60 358	21.06	
4	jpg	92 837	21.06	
home	jpg	762	21.06	
logotip	jpg	70 950	21.06	
mail	jpg	760	21.06	
manzil	jpg	71 409	21.06	
back_menu	png	148	21.06	
bg-checker	png	950	21.06	
box	png	11 732	21.06	
close	png	498	21.06	
donate	png	5 125	21.06	

3.14- расм. «images» модули

- JS модули JavaScript кодларини ўз ичига олади

..				<Папка> 18.06
functions	js	619	21.06	
jquery	js	265 218	21.06	
jquery-ui	js	63 368	21.06	
lightbox	js	11 617	21.06	
quiz_func	js	62	21.06	
slide	js	36 635	21.06	
slides.min.jquery	js	6 784	21.06	

3.15- расм. «js» модули

- Controller модули фреймворкни контроллерларнинг кодларини ўз ичига олади

→ ..		<Папка>	18.06
GalleryController	php	1 442	21.06
MenuController	php	2 313	21.06
NewsController	php	2 612	21.06
PageController	php	1 350	21.06
QuizController	php	4 369	21.06
RoyxatController	php	1 000	21.06
SiteController	php	721	21.06
StatController	php	869	21.06
TestController	php	4 738	21.06

3.16- расм. «controller» модули

- Models модули фреймворкни моделларининг кодларини ўз ичига олади

→ ..		<Папка>	18.06
Answers	php	645	21.06
Comment	php	539	21.06
EmailForm	php	821	21.06
Fanlar	php	638	21.06
Gallery	php	384	21.06
Jurnalis	php	1 328	21.06
Menu	php	561	21.06
News	php	650	21.06
PageModel	php	483	21.06
SomeModel	php	208	21.06
Stat	php	634	21.06
Test	php	1 668	21.06
Yunalishlar	php	966	21.06

3.17- расм. «models» модули

- Views модули фреймворкни viewларининг кодларини ўз ичига олади

→ ..		<Папка>	18.06
gallery		<Папка>	18.06
layouts		<Папка>	18.06
menu		<Папка>	18.06
news		<Папка>	18.06
page		<Папка>	18.06
quiz		<Папка>	18.06
royxat		<Папка>	18.06
site		<Папка>	18.06
stat		<Папка>	18.06
test		<Папка>	18.06

3.18- расм. «views» модули

- Widgets модули фреймворкни widgetларининг кодларини ўз ичига олади

..		<Папка>	18.06
views		<Папка>	21.06
FooterWidget	php		113 21.06
MenuWidget	php		203 21.06
NewsWidget	php		224 21.06
SlideWidget	php		111 21.06
StatWidget	php		109 21.06
TopWidget	php		253 21.06

3.19- расм. «widgets» модули

- Modules модули фреймворкни moduleларининг кодларини ўз ичига олади

elfinder		<Папка>	18.06
user		<Папка>	18.06

3.20- расм. «modules» модули

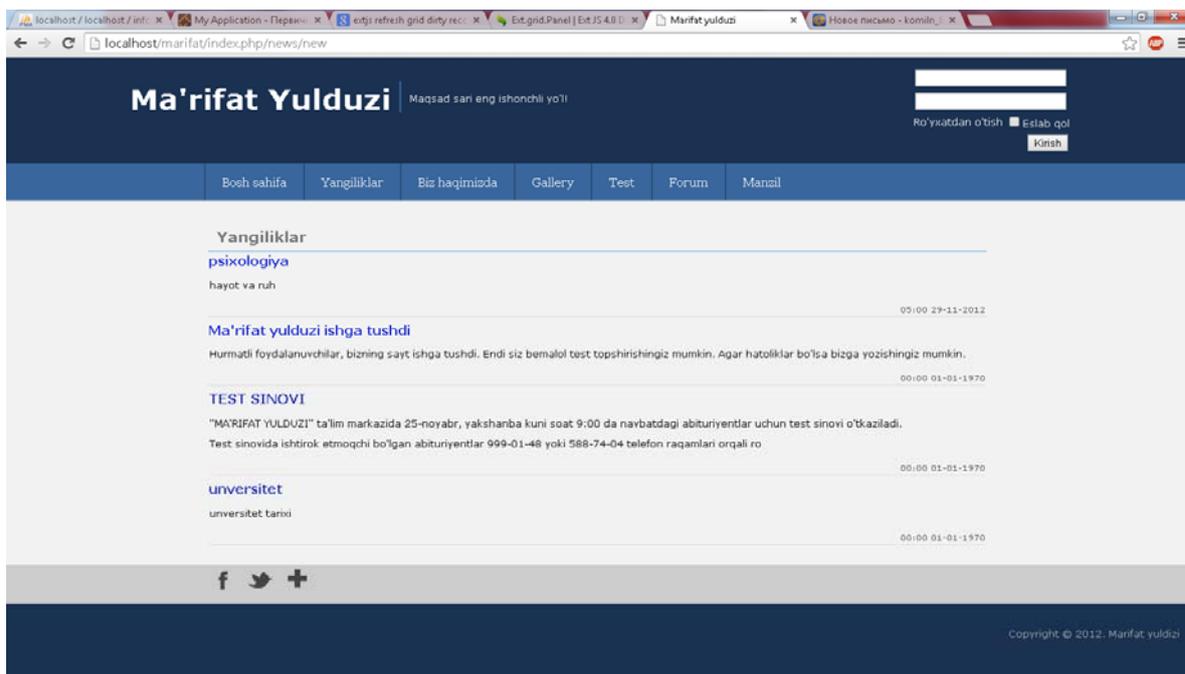
3.3. Веб-сайтнинг фойдаланиш тартиби

Қуйидаги расмда веб-сайтнинг бош саҳифаси кўрсатилган:



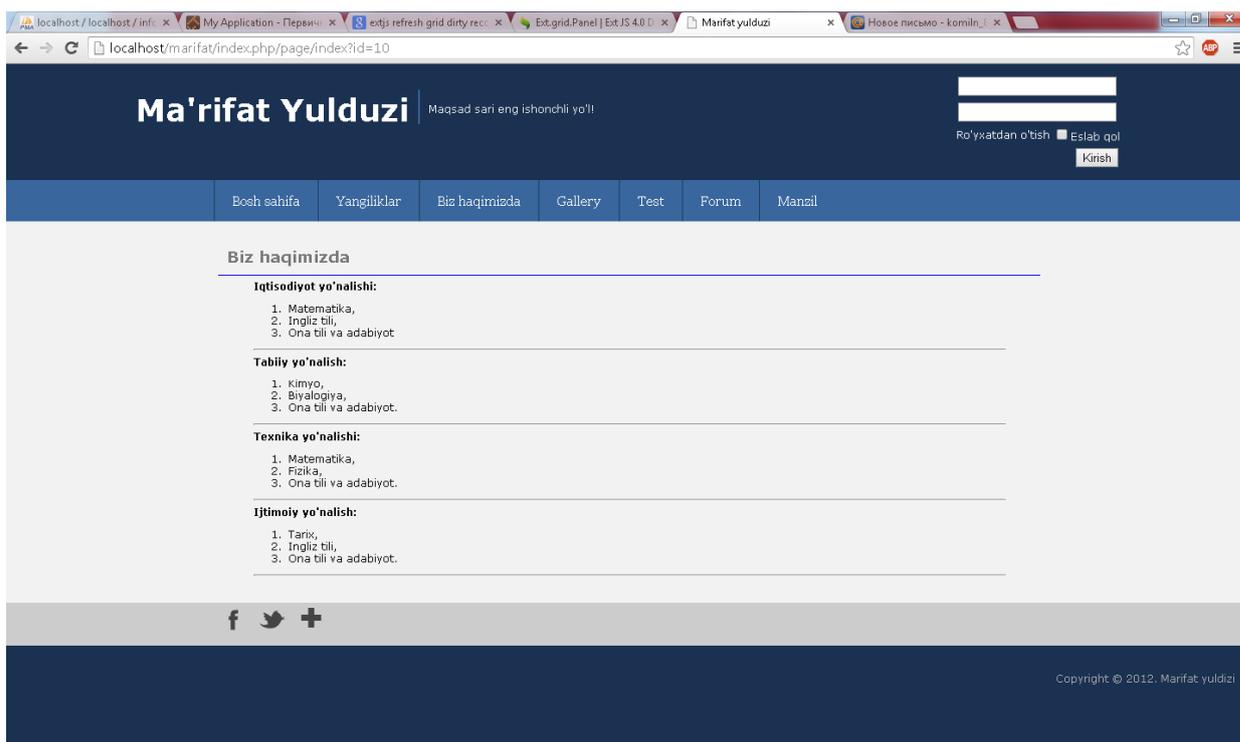
3.21- расм. Бош саҳифа

“Янгиликлар” ссылкаси босилганда, янгиликлар саҳифаси юкланади:



3.22- расм. “Янгиликлар” саҳифаси

“Биз ҳақимизда” ссылкаси босилганда, қуйидаги саҳифа юкланади:



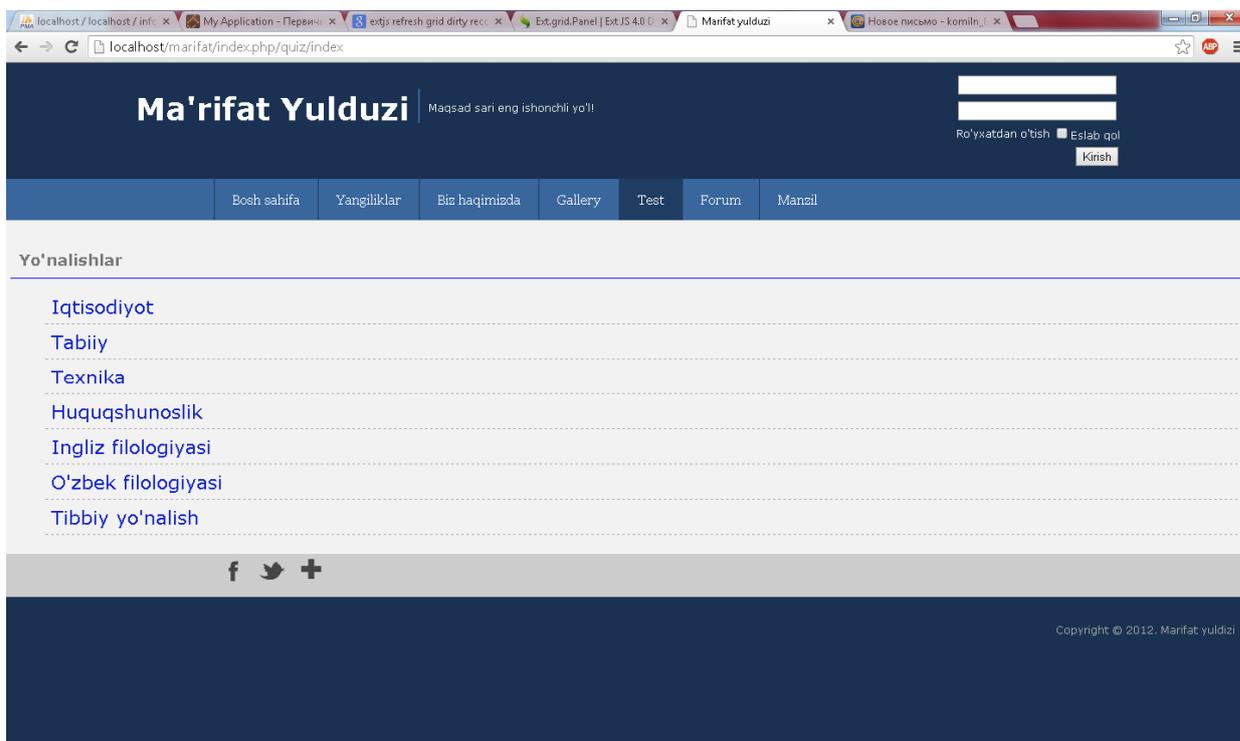
3.23- расм. “Биз ҳақимизда” саҳифаси

“Галерея” саҳифасида расмларни ўз ичига олади:



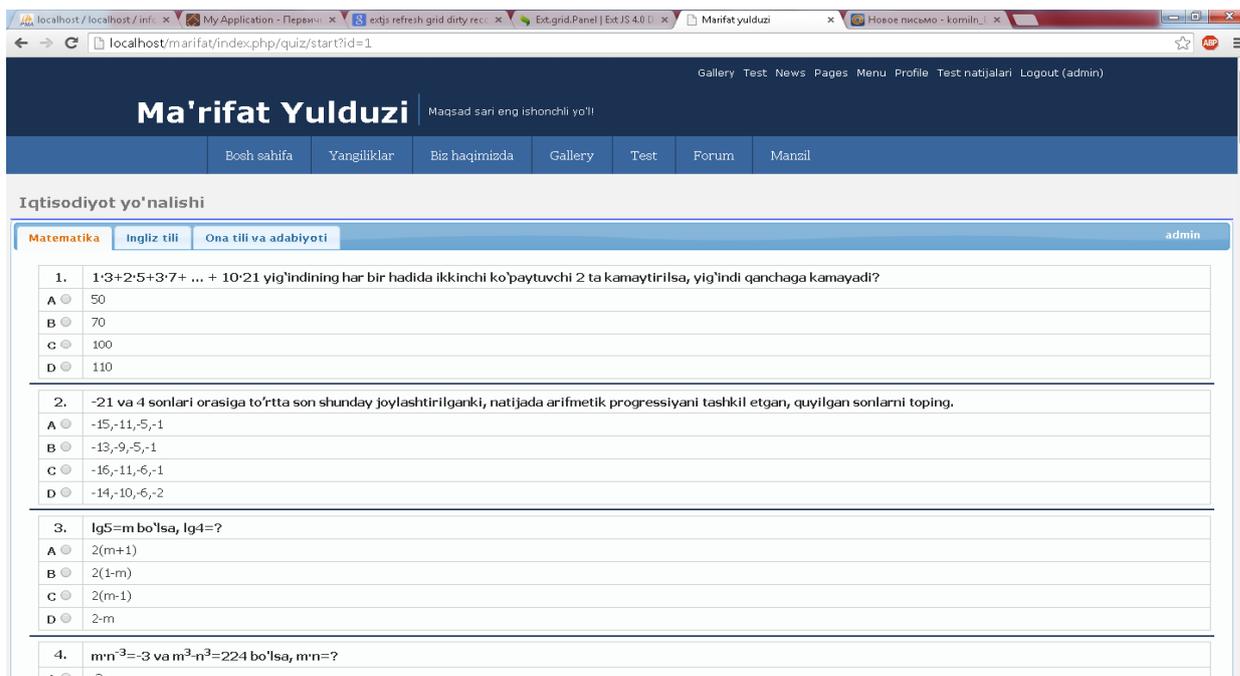
3.24- расм. “Галерея” саҳифаси

“Тест” саҳифасида тестларни ўз ичига олади:



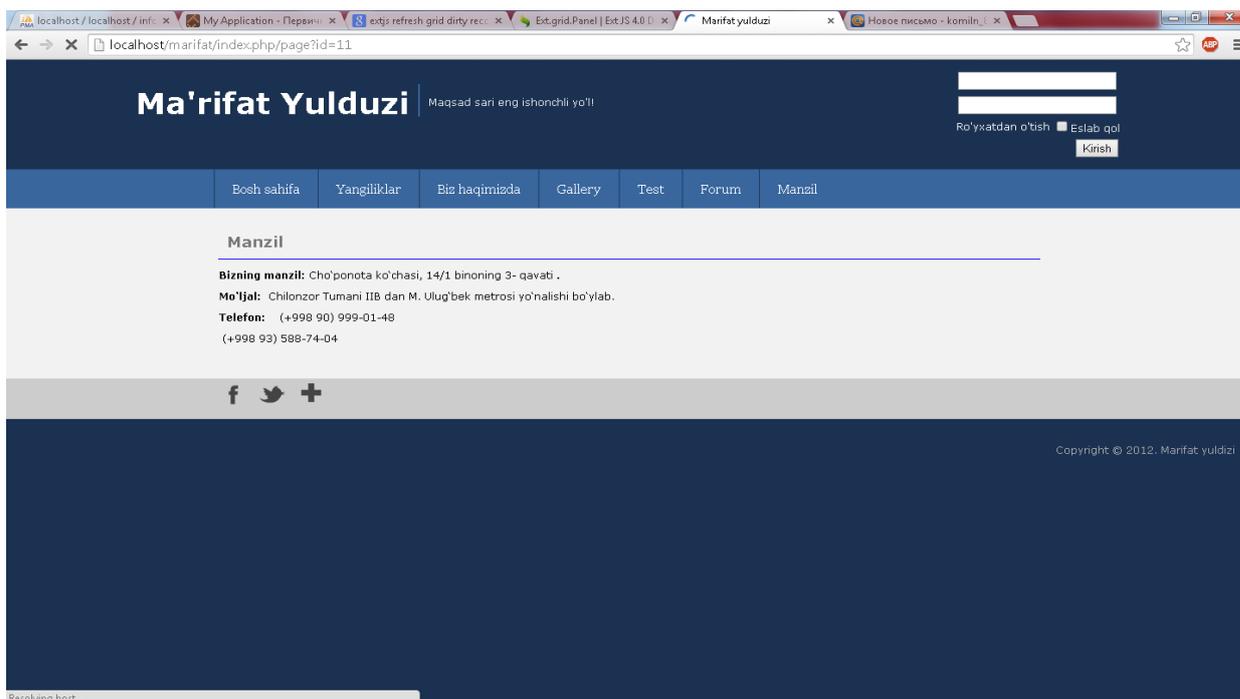
3.25- расм. “Тест” саҳифаси

“Тест” саҳифасида бўлим танланса, шу бўлимга тегишли тестлар юкланади ва тест топширилади:



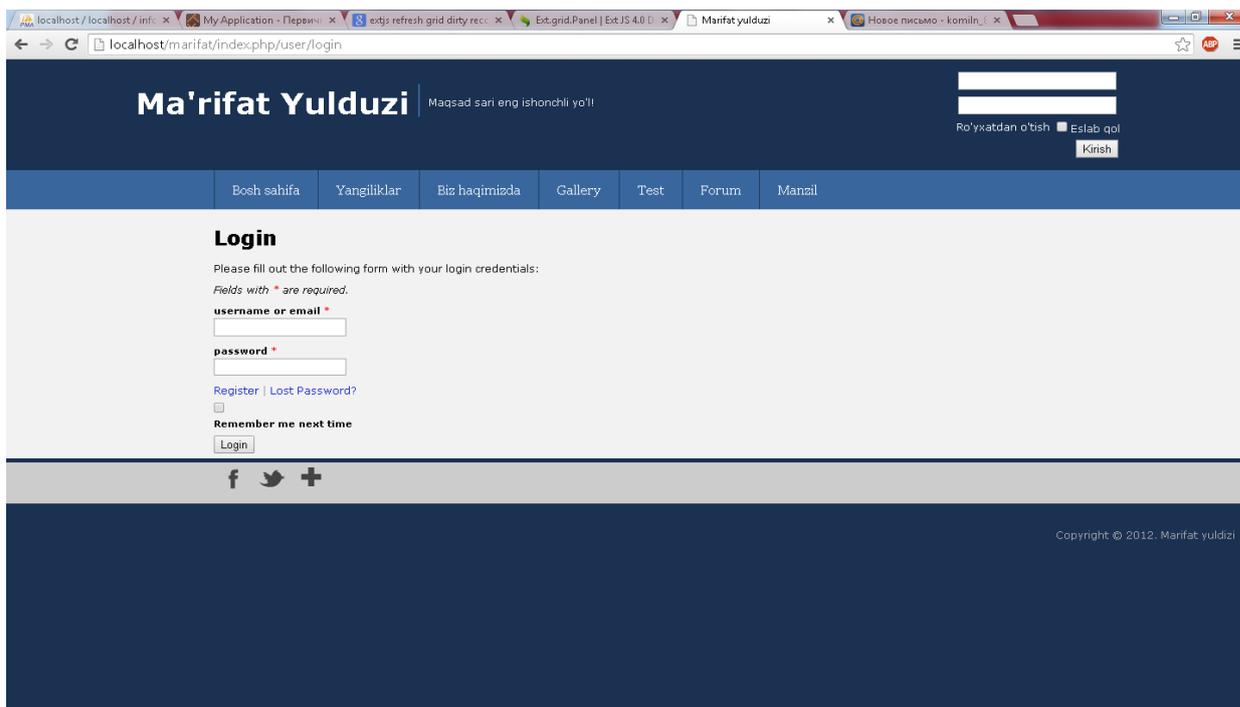
3.26- rasml. "Тест топшириш" sahifasi

"Манзил" sahifasida markazning manzili va kontakt ma'lumotlari keltilirilgan:



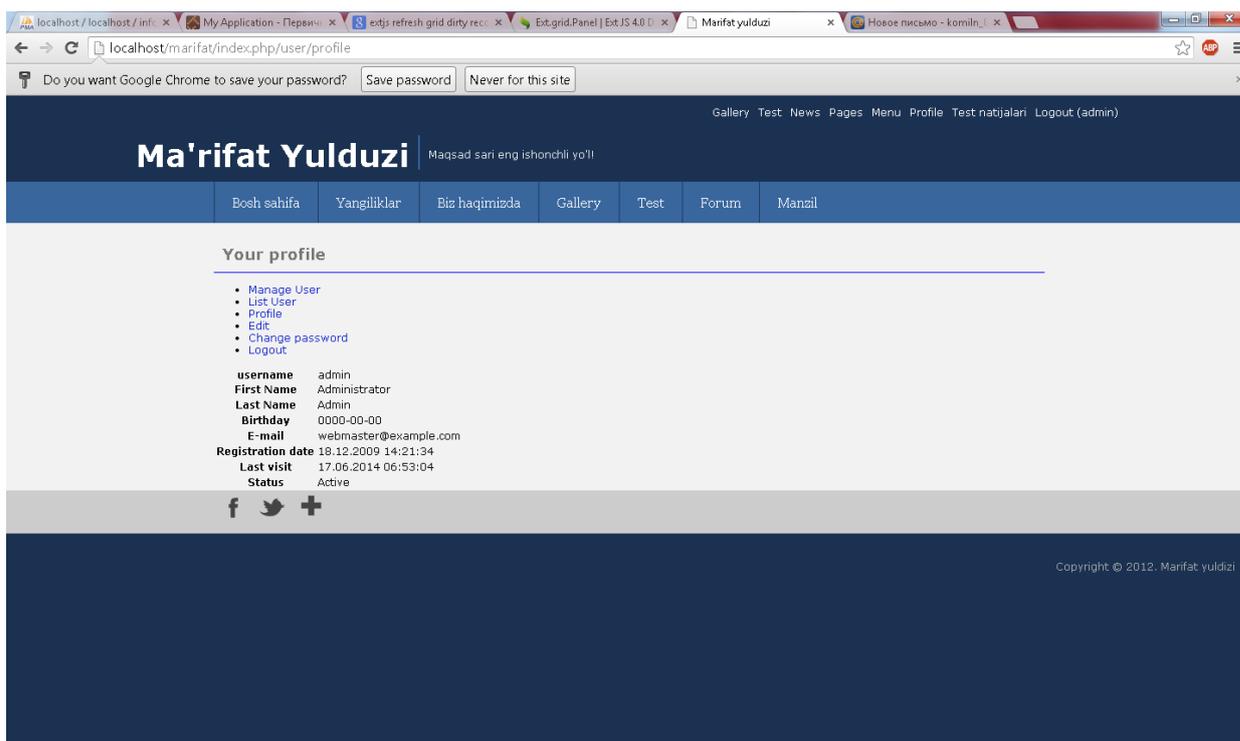
3.27- rasml. "Манзил" sahifasi

Saytga avtorizatsiya(kiirish) mumkin, login va parol kiriritish lozim:



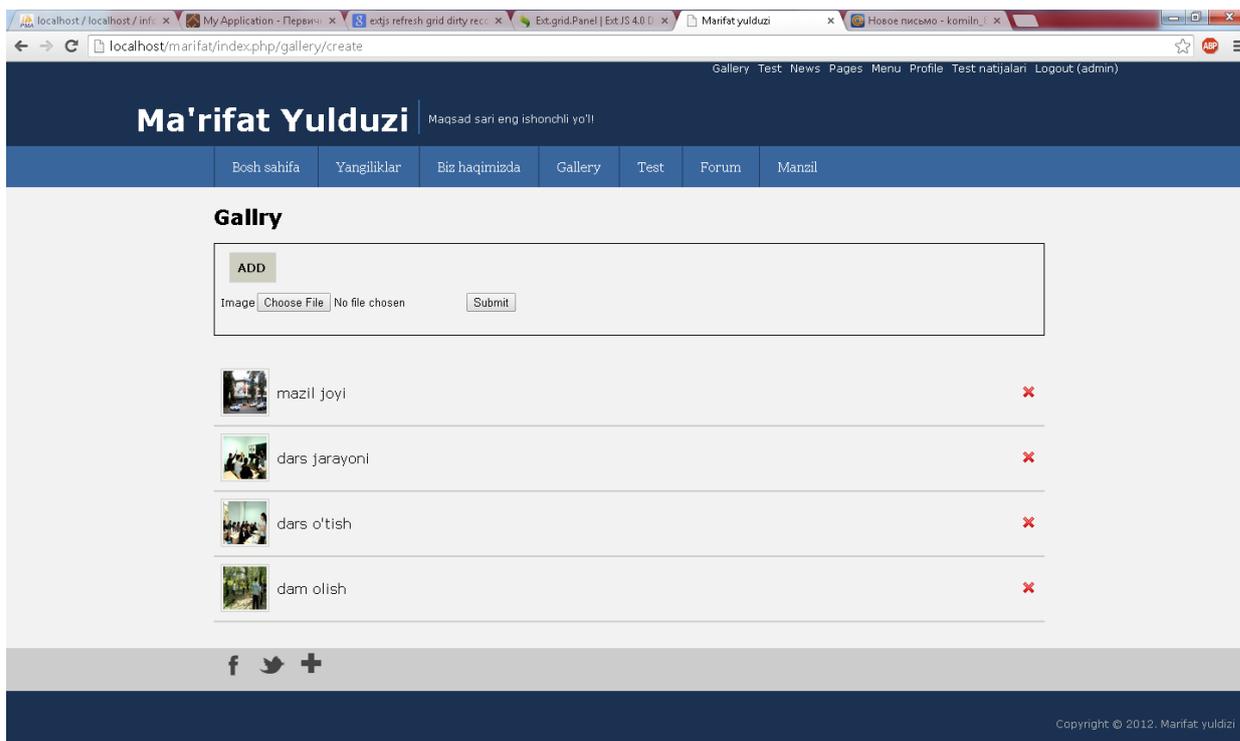
3.28- расм. “Авторизация” ойнаси

Маълумотларни кўриш учун “Profile” ссылкасини босиш лозим:

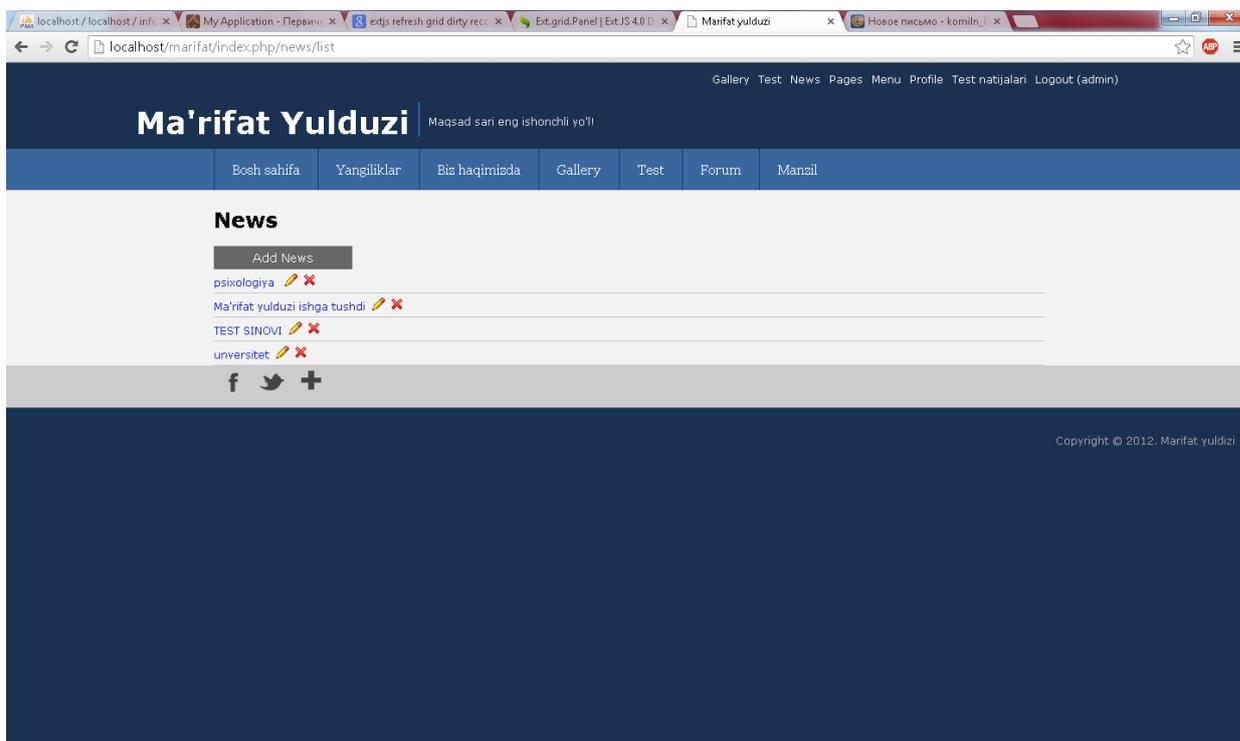


3.29- расм. “Маълумотлар кўриш” ойнаси

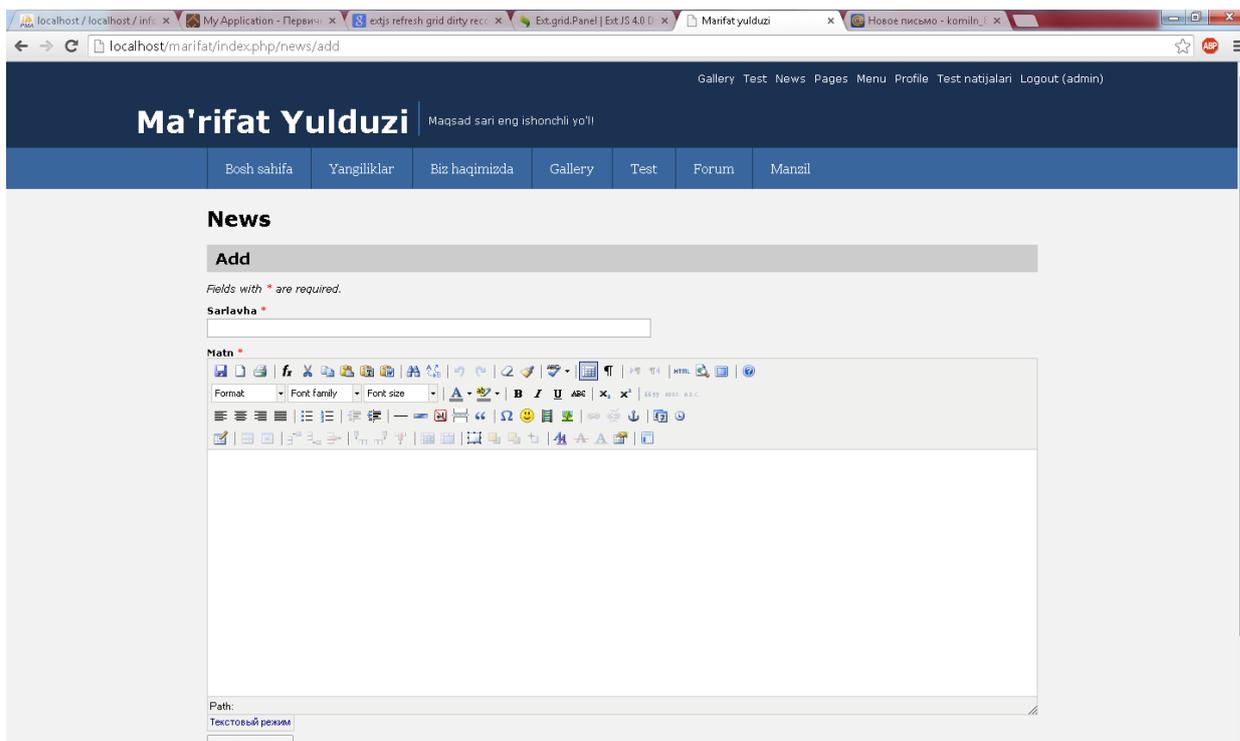
Сайтда галерея шакллантириш функцияси мавжуд:



3.30- расм. “Галереяни шакллантириш” ойнаси
Сайтда янгиликлари бошқариш функцияси мавжуд:

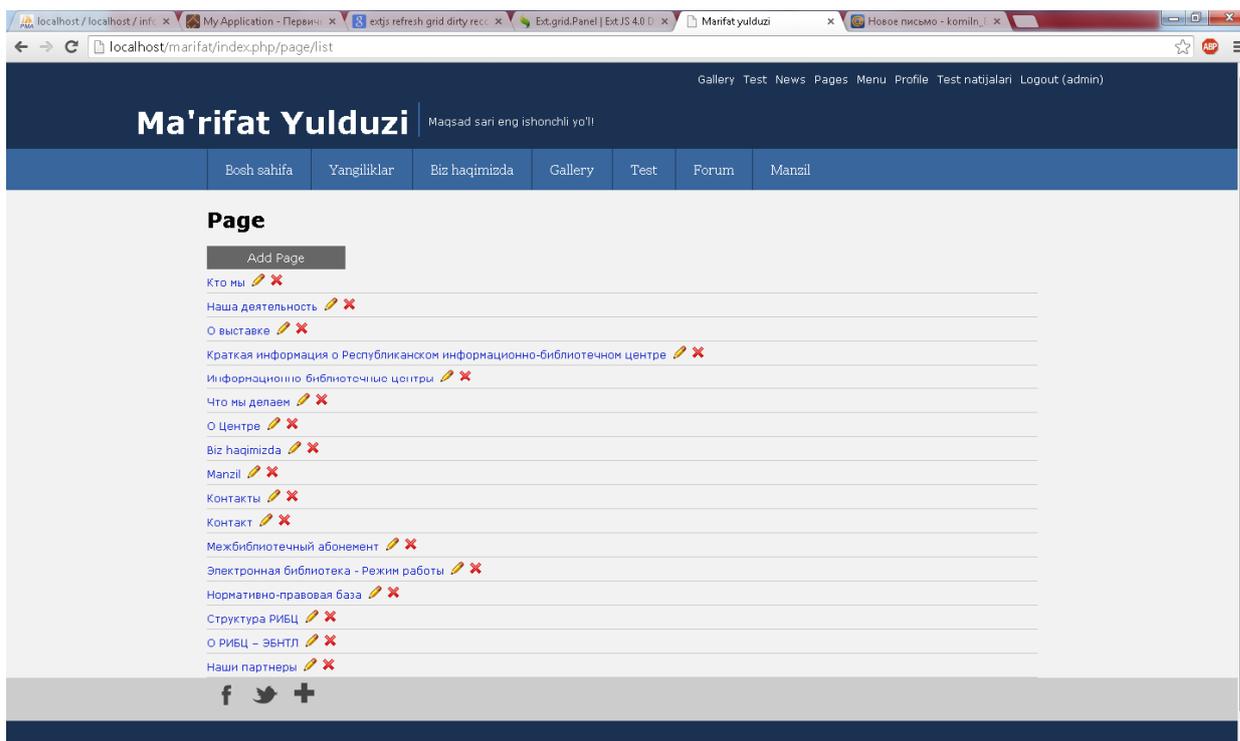


3.31- расм. “Янгиликларни бошқариш” ойнаси

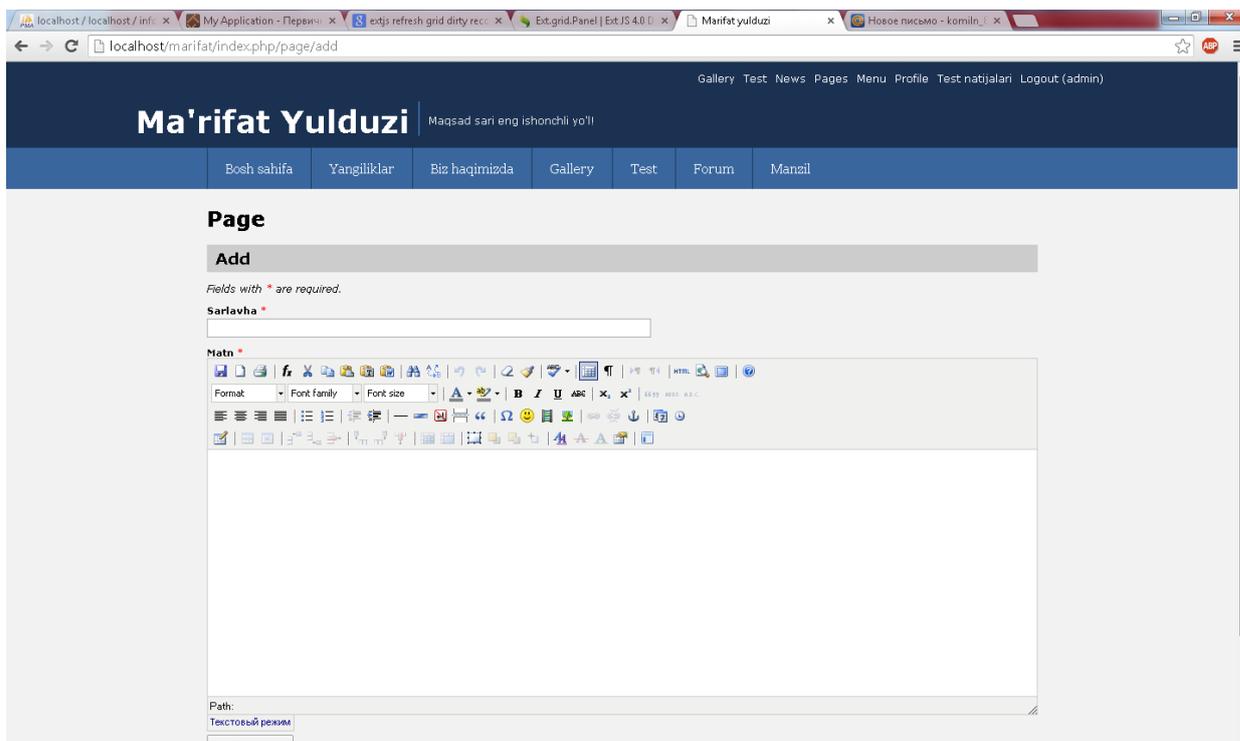


3.32- расм. “Янгиликларни киритиш” ойнаси

Сайтда саҳифаларни бошқариш функцияси мавжуд:

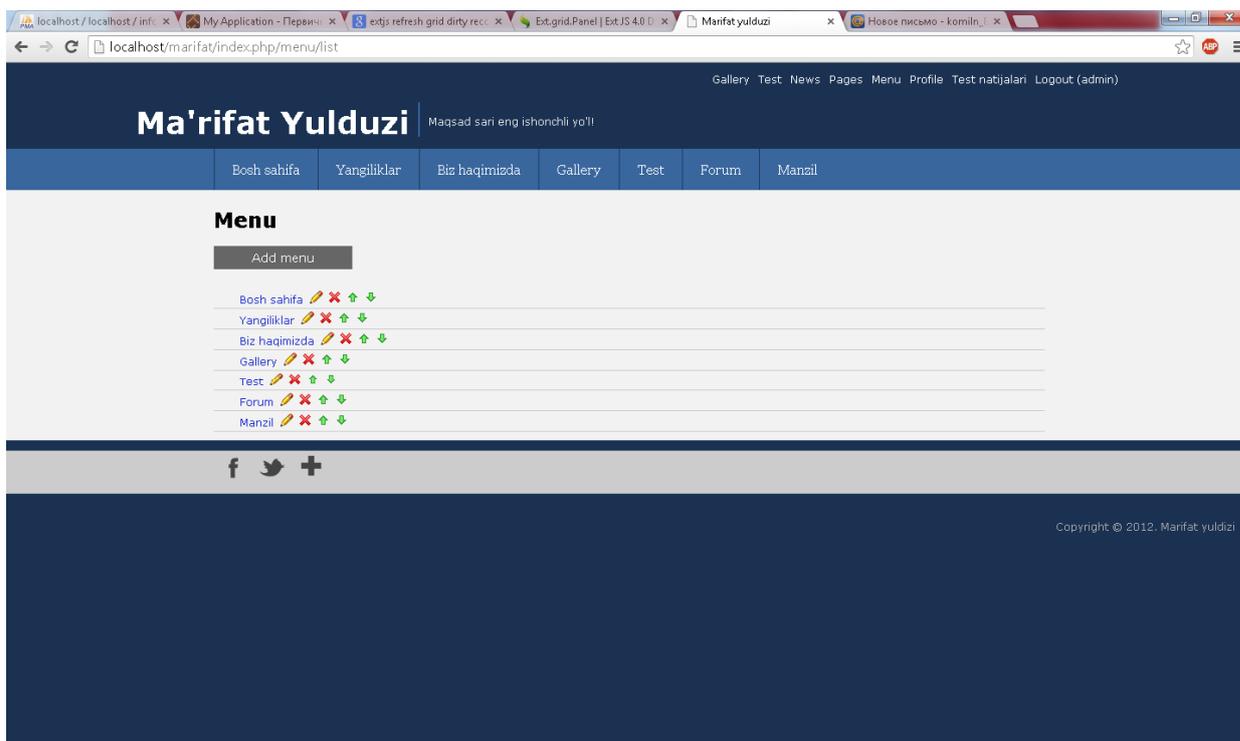


3.33- расм. “Саҳифаларни бошқариш” ойнаси

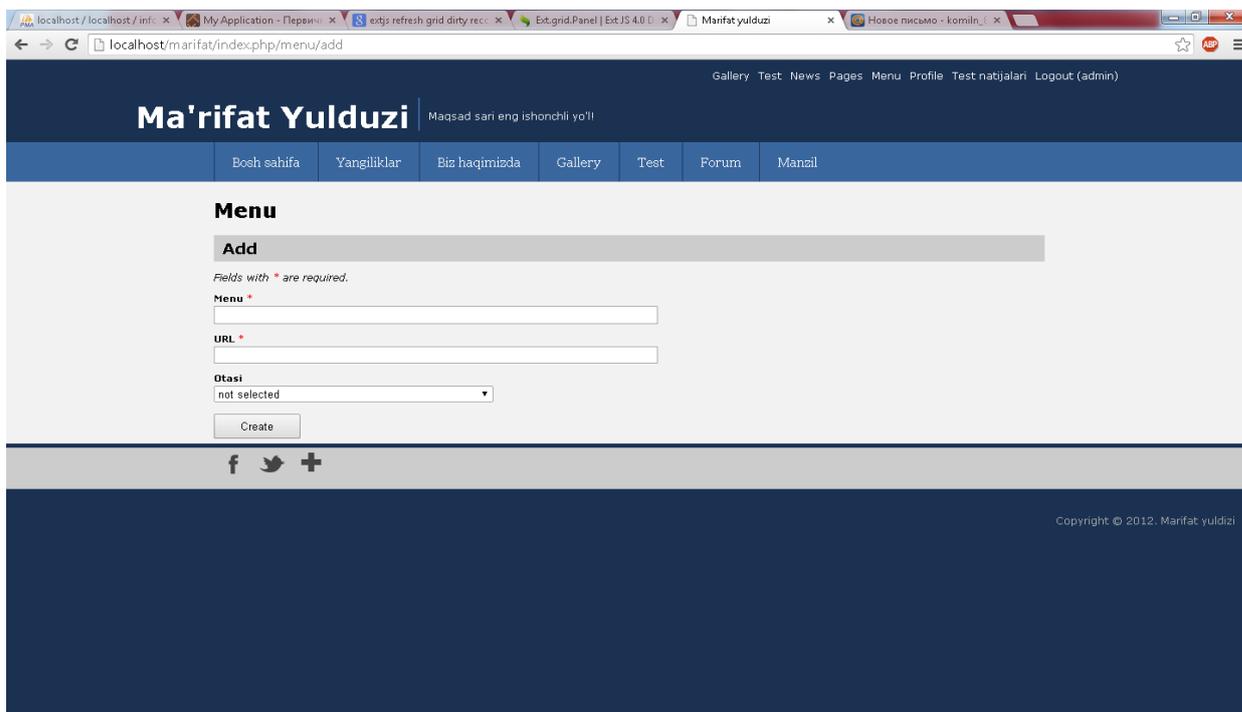


3.34- расм. “Янги саҳифани қўшиш” ойнаси

Сайтда менюларни бошқариш функцияси мавжуд:

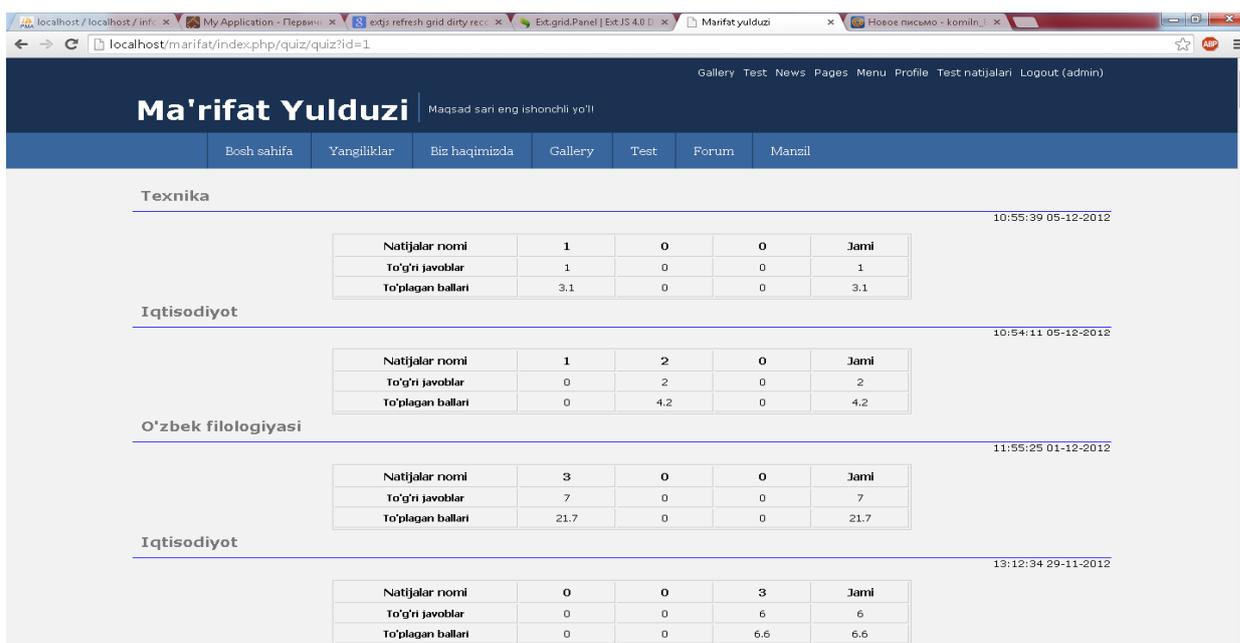


3.35- расм. “Менюларни бошқариш” ойнаси



3.36- расм. “Янги менюни қўшиш” ойнаси

Сайтда тест натижаларини кўриш функцияси мавжуд:



3.37- расм. “Тест натижаларини кўриш” ойнаси

4.БОБ. ҲАЁТ ФАОЛИЯТИ ХАВФСИЗЛИГИ

4.1. Ишлаб чиқаришда сунъий ёритилганлик.

Ёруғлик инсон фаолияти давомида жуда муҳим рол ўйнайди. Кўриш инсон учун асосий маълумот манбаи ҳисобланади. Умумий олинadиган маълумотнинг тахминан 90% кўз орқали олинади.

Табиатда мутлоқ қора нарса бўлмагани сабабли фонни қайтаришда нур қайтариш коэффициентининг 0,02 дан 0,95 гача бўлган чегаралари ҳисобга олинади. Ёруғлик манбаиларига нисбатан ишлаб чиқариш корхоналарини ёритиш икки усулда:

1) табиий қуёш ёруғлиги ёрдамида ёритиш (бунда қуёш тарқатаётган нурдан тўғридан-тўғри фойдаланилади ёки қуёш нурининг таъсирида ёруғлик тарқатаётган осмоннинг диффузия ёруғлигидан фойдаланилади);

2) қуёш ёрдамида ёритишнинг иложи бўлмаган ишлаб чиқариш корхоналари хоналарини ва қуёш ботгандан кейин умуман ишлаб чиқаришда корхоналарини электр нурлари ёрдамида сунъий ёритиш йўли билан амалга оширилади.

Табиий ёруғлик ўзининг барча хусусиятлари билан сунъий ёритишдан кескин фарқ қилади. Табиий ёруғлик инсон кўриш органлари ва бошқа физиологик жараёнларнинг бориши учун ҳарур бўлган ультрабинафша нурларга бой ва бу нурлар билан ёритилган хоналарда ишлаш кўз учун жуда фойдали. Табиий ёруғлик ёритилиш зонаси бўйлаб бир текис тарқалади[11].

Ишлаб чиқариш хоналарининг ва иш жойларининг ёритилганлиги, меҳнат гигиенасининг муҳим курсаткичларидан бири ҳисобланиб, меҳнатни илмий асосда ташкил этишнинг ва ишлаб чиқариш маданиятининг ажралмас қисми ҳисобланади.

Ёритилганлик инсоннинг ташқи муҳит билан боғланишини аниқловчи ва инсон миясига келувчи ташқи дунё туғрисидаги маълумотларнинг сифатини ифодаловчи асосий кўрсаткичларидан биридир.

Тўғри ва нормал миқдордаги ёритилганлик иш куруллари ва жиҳозларнинг рангини, ўлчамларини тезда аниқлашга имкон беради ва ишчининг меҳнат қобилиятини узоқ муддатгача сақланиб қолишига, меҳнат унумдорлигини ошишига, ишлаб чиқилган маҳсулотнинг сифатли бўлишига шароит яратиб, меҳнат хавфсизлигини оширади.

Ёритилганликни меъёрлашни енгиллатиш мақсадида барча ишлар аниқлик даражасига кўра 6 разрядга бўлинган: ўта юқори аниқликдаги ишлар- I-разряд; жуда юқори аниқликдаги ишлар- II-разряд; юқори аниқликдаги ишлар-III-разряд; ўта аниқликдаги ишлар- IV-разряд; кам аниқликдаги ишлар- V-разряд; дағал ишлар-VI-разряд.

Энг юқори ёритилганлик 1 разряддаги ишлар учун белгиланган бўлиб 5000 Лк.ни ташкил этади, кичик ёритилганлик эса V-разряддаги ишлар учун 75 Лк. қилиб белгиланган.

Ташқи муҳитда бажариладиган ишларда иш разрядига боғлиқ ҳолда ёритилганлик 2 дан 50 Лк.гача бўлади. Масалан, МТА ларида машиналарнинг олд қисмидаги ёритилганлик 5 Лк, ишчи аъзолардаги ёритилганлик 10 Лк булиши мумкин.

Сунъий ёритилишни ҳисоблашдан асосий мақсад ишлаб чиқариш корхоналарида ишлатилаётган ёритиш лампаларининг сонини аниқлаш, уларни оқилона жойлаштириш ва электр энергияси иқтисодини таъминлаган ҳолда саноат корхоналари хоналаридаги нормаланган ёруғликдаги иш жойларини таъминлашдан иборат. Бу масалаларни ҳал қилишда бирмунча мустақил масалаларни ҳал қилишга тўғри келади.

1. Ёритиш манбаларини танлаш. Умуман саноат корхоналарини ёритишда люминесцент лампалардан фойдаланиш мақсадга мувофиқдир.

2. Ёритиш системасини танлаш. Ёритиш системасини танлаганда комбинацияли ёритилиш иқтисодий самарадор, аммо умумий ёритилиш эса санитар-гигиеник томондан анча мукамал ҳисобланади, чунки умумий ёритилиш зонани бир текис ёритади. Ёруғлик нурларини маълум участкага тўплаб йўналтириш йўли билан иш жойларида иқтисодий томондан

самарадор бўлган ҳолда умумий ёритилишнинг юқори даражаларини ҳосил қилиш мумкин. I-IV, Va ва Vб разряддаги ишларни бажаришда комбинацияли ёритиш системасидан фойдаланиш тавсия этилади.

3. Иш зонасидаги ҳавонинг тоза-ифлослиги ва ҳаво муҳитидаги газ ва портловчи моддаларнинг буғлари мавжудлигига қараб лампаларнинг турларини танлаш.

4. Лампаларнинг сонини аниқлаш ва уларни иш зонасига жойлаштириш. Лампалар шахмат тартибида, ромб кўринишда ва бошқача усулларда жойлаштирилиши мумкин.

5. Иш жойларида нормаланган ёритилиш миқдорини аниқлаш. Бунинг учун фонга нисбатан контрастлигини белгилаш ва иш жойлари фонини ҳисобга олган ҳолда жадвалга асосан танланган ёритилиш системаси ва лампанинг турига қараб иш жойларидаги минимал нормаланган ёритилишни аниқлаш керак бўлади. Сунъий ёритилишни аниқлаш учун уч усулдан фойдаланилади.

Горизонтал ишчи юзаларни ёритганда нур оқими усулидан фойдаланилади. Бу усул ёритилаётган юзаларга тушаётган нур оқимидан фойдаланиш коэффициенти усули деб ҳам юритилади. Юзаларга тушаётган нур оқими чўғланувчи ва люминесцент лампалардан фойдаланганда куйидаги формула билан аниқланади.

$$F = \frac{E_H S k Z}{N \eta} \quad (1)$$

Бунда,

E_H - минимал ёритиш, лк; F - лампанинг нур оқими, лм;

S - ёритилаётган хонанинг юзаси, м²; Z - минимал ёритилиш коэффициенти, одатда ўртача ёритилиш минимал ёритилишга нисбати олинади ва 1,1 - 1,5 чегаралар атрофида бўлади;

k - эҳтиёт коэффициенти, жадвалдан олинади; N - хонадаги лампалар сони; η - нур оқимидан фойдаланиш коэффициенти.

Ишлаб чиқариш шароитлари амалиётида олинган миқдор ҳисобланган миқдордан - 10 ва + 20% фарқ қилишига йўл қўйилади, акс ҳолда бошқа лампаларнинг жойлаштириш схемаси танланади.

Нуқтали усулдан йўналтирилган иш жойларини ёритишни ҳисоблашда фойдаланилади. Бу усул асосида қуйидаги боғланиш ётади:

$$E = \frac{I\alpha \cdot \cos \alpha}{r^2} \quad (2)$$

бунда: I - ёруғлик манбаидан иш олиб бориладиган юзага йўналтирилган ёруғлик кучи, кд; r - лампадан ёритилаётган юзагача бўлган масофа, м; α - ишчи юза билан манбадан тушаётган нур оқимининг йуналиши орасидаги бурчак; градусларда ўлчанади.

Сунъий ёритилишни ҳисоблаш. Маълумки ишлаб чиқариш корхоналари люминесцент лампалар ёрдамида ёритилади. Сунъий ёритиш нормаларига ҳамма ишлаб чиқариш корхоналарида ёритилиш $E \geq 200$ лк дан кам бўлмаслиги керак.

Иш олиб бориладиган стол баландлигини 0,8 м деб олсак ва лампани шифтга осилган баландлигини шифтга 0,4 м ҳисобласак, лампа иш жойига нисбатан:

$$h = 72 - 0,8 - 0,4 = 6 \text{ м баландликда бўлади.}$$

Бизнинг ҳолда нур тарқатиш борасида ҳеч қандай талаблар қўйилмаганлиги сабабли, биз ёритиш воситаси сифатида ЛБ-80 люминесцент лампасидан фойдаланамиз.

$$\text{Унинг нур оқими миқдори } F_{\text{л}} = 43250 \text{ лм.}$$

4.2. Ёнғин хавфсизлиги бўйича умумий талаблар.

Ёнғин – бу махсус манбадан ташқарида содир бўладиган ва катта материал зарар ҳамда талофатлар келтириб чиқарадиган назоратсиз ёниш жараёнидир.

Ёнғин вақтида содир бўладиган турли хил хавфли ва заррали факторлар таъсирида материал бойликлар нобуд бўлиши ва бахтсиз ҳодисалар рўй

бериши мумкин. Ёнғиннинг хавfli ва заррали факторларига асосан қўйидагиларни киритишимиз мумкин: очик аланга, атроф-муҳитнинг ва ёнғинда қолган буюмларнинг юқори ҳарорати, ёниш вақтида ҳосил бўладиган турли хил захарли газ ва буғлар, тутунлар, кислороднинг кам концентрацияда бўлиши, қурилиш конструкциялари ва материалларининг қулаб тушаётган қисмлари, ёнғин вақтида содир бўладиган портлаш, портлашдаги тўлқин зарбаси, портлаш таъсирида учиб кетган материаллар ва зарарли моддалар ва б[12].

Ёнғин хавфсизлиги бўйича категорияларини аниқлаш. Ишлаб чиқариш корхоналарининг ишлаб чиқариш бинолари, энергетика, транспорт ва омбор хўжалиги бинолари технологик жиҳатдан қандай мақсадда ишлатишлари жиҳатидангина эмас, айрим жараёнларнинг ёнғин жиҳатидан қанчалик хавfliлигига қараб ҳам бир-биридан фарқ қилади.

Амалдаги нормаларга кўра, барча турдаги корхоналар ёнғин жиҳатидан хавfliлигига қараб бешта категорияга бўлинади: А,Б,В,Г,Д.

Уй-жой бинолари, жамоат бинолари ва иншоатлар ёнғин чиқиш хавfli жиҳатидан категорияларга бўлинмайди. Лекин қуриладиган биноларнинг неча қават бўлиши кераклиги, шунингдек ёнғинга қарши деворлар орасидаги майдоннинг йўл қўйиладиган максимал катталиги шу биноларнинг ўтга чидамлилиқ даражасига боғлиқ.

Ишлаб чиқариш бино ва қурилмаларини лойиҳалашда кўзда тутилган ёнғинга қарши тадбирлар, энг аввало, ишлаб чиқариш жараёнининг ёнғин жиҳатидан хавfliлиқ даражасига боғлиқ бўлади.

Лойиҳалаш меъёрларига мувофиқ, ишлаб чиқаришлар ёнғин жиҳатидан хавfliлиқ даражасига кўра олтига тоифага бўлинади.

А тоифага буғларнинг чақнаш ҳарорати 28° ва ундан паст бўлган жуда кўп суюкликлар ёки пастки портлаш чегараси ҳаво ҳажмига нисбатан 10% ва ундан кам бўлган ёнувчи газлар ишлатиладиган ишлаб чиқариш бинолари киради. Бу тоифага бензин, кальций карбид сақланадиган омборлар, шунингдек, буғининг чақнаш ҳарорати $28-61^{\circ}$ ва

пастки портлаш чегараси 10% дан юқори бўлган осон алангаланувчи суюқликлар ҳамда газлар ишлатиладиган ишлаб чиқаришлар киради.

В тоифага ёнувчи қаттик моддалар, шунингдек портлаш чегараси 65 г/м³ бўлган чанг ёки толалар ажратиб чиқарадиган моддаларга ёки буғларнинг чакнаш ҳароратидан юқори бўлган суюқликларга ишлов бериладиган ва улар ишлатиладиган ишлаб чиқаришлар киради. Бундай ишлаб чиқаришлар қаторига: ёғоч тилиш, дурадгорлик ва омихта ем ва озуқа тайёрлаш цехлари, дон сақланадиган омборлар, тегирмонларнинг дон тозалаш бўлимлари, зиғир, пахтага дастлабки ишлов бериш цехлари, бензинсиз ёнилғи-мойлаш материаллари сақланадиган омборлар, ёпик кўмир омборлари, электр тақсимлаш қурилмалари киради.

Бензинда ҳаракатланадиган транспорт воситаларини сақлаш гаражалари ҳам шу тоифага киради.

Г тоифага ёнилғи(шу жумладан, газ) ёқиладиган ёки ёнилғидан ёнмайдиган моддаларга ишлов бериладиган ишлаб чиқаришлар киради. Ёнмайдиган моддаларга қиздирилган ёки суюқлантирилган ҳолатда ишлов берилади ва бунда нур энергия ажралади. Бундай ишлаб чиқаришларга иссиқлик электрстансияларидаги қозонхоналар, темирчилик хоналари, машина заллари, мотор синаш станциялари киради.

Д тоифага ёнмайдиган моддаларга деярли совуқлайин ишлов бериладиган ишлаб чиқаришлар, масалан, ёнмайдиган суюқликлар ҳайдаладиган насос станциялари, сабзавотлар, сут, балиқ ва гўшт маҳсулотларига ишлов бериш цехлари, биологик ёки техникавий усулда иситиладиган иссиқхона ва парниклар киради. Газ билан иситиладиган парник ва иссиқхоналар бундан мустасно, улар Г тоифага тааллуқли.

Е тоифага кирадиган ишлаб чиқаришларда суюқ фазасиз ёнувчи газлар ва портловчи чанглар шундай миқдорда ишлатиладики, бунда улар ҳажми хона ҳажмининг 5% идан кўп бўлган ҳаво билан аралашиб портловчи аралашмалар ҳосил қилади. Бунда портлаш технологик жараён шартларига кўра фақат кейинги ёнишсиз содир бўлади. Бу категорияга, масалан,

аккумуляторлар хоналари, водородли ёки асетиленли баллонлар омборини киритиш мумкин.

Ёнғин портлаш хавфсизлигини аниқлаш. Моддалар ёнаётганда содир бўладиган барча химиявий ва физик жараёнларнинг асл моҳиятини яхши билган тақдирдагина ёнғинга қарши қаратилган тадбирларни тўғри ва самарали ташкил этиш ҳамда ёнғин ўчириш воситаларини тўғри танлаш мумкин[12].

Ёниш пайтида аланга ҳосил бўлиши ҳам, бўлмаслиги ҳам мумкин. Ёнаётган моддадан ёнувчи газ ажралиб чиққанида аланга ҳосил бўлади. Бундай ҳолларда аланга газ қобиғидан иборат бўлиб, газ ва буғлар шу қобик ичида ёнади; ёғоч, тошкўмир ва ёнувчи суюқликлар шундай ёнади. Писта кўмир, кокс алангасиз ёнади.

Ёнувчи модда алангаланаши ва ёна бошлаши учун уни маълум ҳароратгача қиздириш керак; бу ҳарорат турли моддалар учун турлича бўлади. Модда қандай ҳароратда алангаланса ва ёна бошласа, шу ҳарорат унинг алангаланиши ҳарорати деб аталади. Ҳавода кислород бўлган тақдирдагина шундай ҳодиса рўй беради.

Алангаланиш ҳарорати модданинг ўзигагина боғлиқ бўлмай, шу билан бирга, атмосфера босимига, ҳаводаги кислород миқдорига ва бошқа сабабларга боғлиқ. Ҳатто битта модданинг алангаланиш ҳарорати ҳам жуда ўзгариб туриши мумкин. Масалан, ёғочнинг алангаланиш ҳарорати 250-350, торфники 225-280 атрофида бўлади ва ҳоказо.

Моддаларнинг ҳарорати ошиши алангаланишга сабаб бўлади. Алангаланиш натижасида оловнинг таъсири, ёруғлик нурларидан чиққан иссиқлик, электр учқунлари, қуёш нурларининг иссиқлиги, яшин чакмоқлари ва бошқалар шундай сабаблардандир. Материалнинг алангаланиш ҳарорати қанчалик паст бўлса, унинг ёниб кетиш хавфи шунчалик кучли бўлади.

Моддаларнинг кимёвий хусусиятидан маълумки, атроф - муҳит ҳарорати қанчалик юқори бўлса, кимёвий рақциялар, шу жумладан ёниш жараёни ҳам шунчалик тез ўтади.

ХУЛОСА

Биттирув малакавий ишини бажариш орқали қўйдаги асосий натижаларга эришилди:

1. Web технология ва уларнинг асосий авфзалликлари ва камчиликлари кўриб чиқилди белгилаб олинди.

2. Сайтни ишлаб чиқиш учун инструминтал воситалар таҳлил этилиб PHP ва MySQL муҳитлари танлаб олинди.

3. Биттирув малакавий ишида, “Маърифат юлдузи таълим маркази” ишлаб чиқилган бўлиб унинг авфзаллиги юқори тезликда ишлаши қулай ва содда интерфейсга эга эканлиги ҳамда тингловчилар учун етти хил фандан ўз билимини синаш учун тест топшириқларини киритилганлигини келтириш мумкин.

4. Ҳаёт фаолияти ҳавфсизлиги бўйича ишлаб чиқишда сунъий ёритилганлик ва ёнғин ҳавфсизлигини таъминлаш бўйича тавсиялар ишлаб чиқилган.

Фойдаланилган адабиётлар рўйхати

1. Каримов И.А. Ўзбекистон буюк келажак сари . Ўзбекистон, 1998.
2. И.Каримов Ўзбекистон XXI аср бўсағасида: Хавфсизликка таҳдид, барқарорлик шартлари ва тараққиёт кафолатлари. Т.: Ўзбекистон, 1997.
3. Ғуломов С.С. ва бошқалар. Ахборот тизимлари ва технологиялари. «Шарқ», Тошкент, 2000.
4. Назиров Ш.А., Қобулов Р.В. Дастурлаш фани бўйича мутахассислар учун АКТ тизимларида дастурлаш. Тошкент, 2006
5. Мингликулов З.Б., асс. Агзамходжаева М.Р. “Веб-дастурлаш” фани маъруза матнларини /ТАТУ, Тошкент, 2009., 125 с
6. Хамдамов Р.Х., Кобилов С.С. Проблемы внедрения новых информационных технологий в высшем образовании и пути их решения. Журнал «Таълим муаммолари». 2000. - № 4. –с. 67-72.
7. Алимов С. JavaScript Ҳаммага. Тошкент 2006, 95 с.
8. Максим Кузнецов, Игорь Симдянов, Сергей Голышев. PHP 5 на примерах. Серия: На примерах. Издательство: БХВ-Петербург, 2010 г., 576 стр.
9. Поль Дюбуа MySQL MySQL Серия: Landmark Другие издания: Твердую переплет Аналогии: Твердую переплет Издательство: Вильямс, 2012 г., 1064 стр.
10. Ёрматов Р. Меҳнат фаолияти хавфсизлиги. Тошкент , 1998.
11. Йўлдошев У.Р. Ёнғин хавфсизлик асослари. Т.1995
12. <http://www.intuit.ru/department/pl/plphp/>.
13. http://w3schools.com/html/html_examples.asp
14. http://codeigniter.com/user_guide/.
15. <http://www.php.net/manual/en/> .
16. <http://dasturchi.uz/questions?tag=Mysql>
17. <http://ru.wikipedia.org/wiki/Yii>

Ilova

```
<?php

// change the following paths if necessary

$yii=dirname(__FILE__).'/yii-core/framework/yii.php';
$config=dirname(__FILE__).'/protected/config/main.php';

// remove the following lines when in production mode

defined('YII_DEBUG') or define('YII_DEBUG',true);

// specify how many levels of call stack should be shown in each log message
defined('YII_TRACE_LEVEL') or define('YII_TRACE_LEVEL',3);

require_once($yii);

Yii::createWebApplication($config)->run();

<?php

// uncomment the following to define a path alias

// Yii::setPathOfAlias('local','path/to/local-folder');

// This is the main Web application configuration. Any writable
// CWebApplication properties can be configured here.

return array(

    'basePath'=>dirname(__FILE__).DIRECTORY_SEPARATOR.'..',

    'name'=>'My Web Application',

    // preloading 'log' component

    'preload'=>array('log'),

    // autoloading model and component classes

    'import'=>array(

        'application.models.*',

        'application.components.*',

        'application.widgets.*',

        'application.modules.user.models.*',

        'application.modules.user.components.*',
```

```

),

'aliases' => array(
    'xupload' => 'ext.xupload'
),

'modules'=>array(
    // uncomment the following to enable the Gii tool
    /*
    'gii'=>array(
        'class'=>'system.gii.GiiModule',
        'password'=>'Enter Your Password Here',
        // If removed, Gii defaults to localhost only. Edit carefully to taste.
        'ipFilters'=>array('127.0.0.1', '::1'),
    ),
    */
    'user'=>array(
        'sendActivationMail' => false,
        'activeAfterRegister' => false,
    ),
),

// application components
'components'=>array(
    //assetManager' => array(
        //linkAssets' => true,
    //),
    'user'=>array(
        // enable cookie-based authentication
        'allowAutoLogin'=>true,
        'loginUrl' => array('/user/login'),
    ),
    // uncomment the following to enable URLs in path-format

    'urlManager'=>array(
        'urlFormat'=>'path',
        'rules'=>array(
            '<controller:\w+>/<id:\d+>'=>'<controller>/view',

```

```

        '<controller:\w+>/<action:\w+>/<id:\d+>'=>'<controller>/<action>',
        '<controller:\w+>/<action:\w+>'=>'<controller>/<action>',
    ),
),

// uncomment the following to use a MySQL database

'db'=>array(
    'connectionString' => 'mysql:host=localhost;dbname=info',
    'emulatePrepare' => true,
    'username' => 'root',
    'password' => "",
    'charset' => 'utf8',
    'tablePrefix' => 'tbl_'
),

'errorHandler'=>array(
    // use 'site/error' action to display errors
    'errorAction'=>'site/error',
),

'log'=>array(
    'class'=>'CLogRouter',
    'routes'=>array(
        array(
            'class'=>'CFileLogRoute',
            'levels'=>'error, warning',
        ),
        // uncomment the following to show log messages on web pages
        /*
        array(
            'class'=>'CWebLogRoute',
        ),
        */
    ),
),
),
),

```

```

// application-level parameters that can be accessed

// using Yii::app()->params['paramName']

$params=>array(

    // this is used in contact page

    'adminEmail'=>'webmaster@example.com',

),

);

<?php

class Controller extends CController

{

    public $layout='//layouts/main';

    public $breadcrumbs=array();

    public function error_r($error_code = 404, $error_text = 'The requested page does not exist.') {

        echo $error_code." - ".$error_text;

        Yii::app()->end();

    }

    protected function beforeAction($action) {

        if($this->getId() === "quiz")

            $this->layout = "quiz" ;

        if($this->getId() === "gallery") {

            $this->layout = "main";

            $cs=Yii::app()->getClientScript();

            $cs->registerScriptFile(Yii::app()->baseUrl . '/js/jquery.js', CClientScript::POS_HEAD);

        }

        $is_admin = $this->isAdmin();

        $allowed_actions = array(

            'add' => 0,

            'edit' => 1,

            'delete' => 2,

            'create' => 3,

            'list' => 4,

```

'managetest' => 5

);

```
if($is_admin) {  
    return true;  
}  
  
if($this->getId() === "test" && $is_admin)  
    return true;  
else if($this->getId() === "profile" && $action->id === "edit")  
    return true;  
else if(!$is_admin && !array_key_exists($action->id, $allowed_actions))  
{  
    return true;  
}  
else {  
    echo "i'm so sorry, do not try this action";  
    Yii::app()->end();  
}  
  
return false;  
}
```

```
public function isAdmin() {  
    return Yii::app()->getModule('user')->isAdmin();  
}  
}
```

<!DOCTYPE html>

<html>

<head>

<title>Marifat yulduzi</title>

<meta http-equiv="content-type" content="text/html; charset=utf-8" />

<link rel="stylesheet" type="text/css" href="<?php echo Yii::app()->request->baseUri; ?>/css/main.css">

<link rel="stylesheet" type="text/css" href="<?php echo Yii::app()->request->baseUri; ?>/css/slide.css">

<link rel="stylesheet" type="text/css" href="<?php echo Yii::app()->request->baseUri; ?>/css/form.css">

<link rel="stylesheet" type="text/css" href="<?php echo Yii::app()->request->baseUri; ?>/css/lightbox.css">

<link rel="Shortcut Icon" type="image/x-icon" href="/favicon.ico" />

```

<?php

/*Yii::app()->clientScript->registerCssFile(

    Yii::app()->assetManager->publish(

        Yii::app()->basePath . '/vendors/jquery.ui/redmond/'

    ).

    '/jquery-ui.css', 'screen'

);*/

//Yii::app()->clientScript->registerCoreScript('jquery');

$cs=Yii::app()->getClientScript();

//$cs->registerScriptFile(Yii::app()->baseUrl . '/js/jquery.js', CClientScript::POS_HEAD);

//$cs->registerScriptFile(Yii::app()->baseUrl . '/js/jquery-ui.js', CClientScript::POS_HEAD);

$cs->registerScriptFile(Yii::app()->baseUrl . '/js/slides.min.jquery.js', CClientScript::POS_HEAD);

$cs->registerScriptFile(Yii::app()->baseUrl . '/js/slide.js', CClientScript::POS_HEAD);

$cs->registerScriptFile(Yii::app()->baseUrl . '/js/lightbox.js', CClientScript::POS_HEAD);

$cs->registerScriptFile(Yii::app()->baseUrl . '/js/functions.js', CClientScript::POS_HEAD);

?>

<script type="text/javascript">

<?php echo "site_url = \".Yii::app()->createUrl("").\"; ?>

<?php echo "base_url = \".Yii::app()->request->baseUrl.\"; ?>

</script>

</head>

<body>

<div>

<div class="wrapper col1">

<?php $this->widget("TopWidget");?>

</div>

<div class="wrapper col2">

<?php $this->widget("MenuWidget");?>

</div>

<div class="w">

<div class="we">

```

```

        <div class="wrapper col4">
            <?php echo $content; ?>
        </div>
    </div>
</div>

<div class="wrapper col6">
    <?php $this->widget("FooterWidget");?>
</div>

</div>
</body>
</html>

```

```
<?php
```

```
class Answers extends CActiveRecord
```

```
{
```

```
    static public function model($className = __CLASS__)
```

```
    {
```

```
        return parent::model($className);
```

```
    }
```

```
    public function tableName()
```

```
    {
```

```
        return '{{q_answer}}';
```

```
    }
```

```
    public function rules() {
```

```
        return array(
```

```
            array('tid, right', 'numerical'),
```

```
            array('name', 'required'),
```

```
        );
```

```
    }
```

```
    public function attributeLabels() {
```

```
        return array(
```

```
            'name' => "Javob",
```

```
            'right' => "T"
```

```

        );
    }

    static public function loadAnswer($id) {
        $model = Answers::model()->findByPk($id);
        if($model == null) return new Answers;
        return $model;
    }
}

<?php
class Comment extends CActiveRecord
{
    static public function model($className = __CLASS__)
    {
        return parent::model($className);
    }

    public function tableName()
    {
        return '{{comment}}';
    }

    public function rules() {
        return array(
            array('content, nid, date, uid', 'required'),
        );
    }

    public function relations() {
        return array(
            'author' => array(self::BELONGS_TO, 'User', 'uid'),
        );
    }

    public function attributeLabels() {
        return array(
            'content' => 'Matn',

```

```

        );
    }
}

<?php
class EmailForm extends CFormModel
{
    public $email;
    public $to;
    public $subject;
    public $message;
    public $from;

    /**
     * Declares the validation rules.
     */
    public function rules()
    {
        return array(
            // name, email, subject and body are required
            array('email, to, subject, message', 'required'),
            // email has to be a valid email address
            array('email', 'email'),
            // verifyCode needs to be entered correctly
            //array('verifyCode', 'captcha', 'allowEmpty'=>!CCaptcha::checkRequirements()),
        );
    }

    /**
     * Declares customized attribute labels.
     * If not declared here, an attribute would have a label that is
     * the same as its name with the first letter in upper case.
     */
    public function attributeLabels()
    {
        return array(
            'verifyCode'=>'Verification Code',
        );
    }
}

```

```
}  
}
```

```
<?php
```

```
class Fanlar extends CActiveRecord
```

```
{
```

```
    static public function model($className = __CLASS__)
```

```
    {
```

```
        return parent::model($className);
```

```
    }
```

```
    public function tableName()
```

```
    {
```

```
        return '{{q_fanlar}}';
```

```
    }
```

```
    public function rules() {
```

```
        return array(
```

```
            array('name', 'required'),
```

```
            array('name', 'length', 'max' => 255),
```

```
        );
```

```
    }
```

```
    public function attributeLabels() {
```

```
        return array(
```

```
            'name' => "Fan",
```

```
        );
```

```
    }
```

```
    public static function items() {
```

```
        $items = array();
```

```
        $model = self::model()->findAll();
```

```
        foreach ($model as $value)
```

```
            $items[$value->id] = $value->name;
```

```
        return $items;
```

```
    }
```

```
}
```

```
<?php
```

```
class Gallery extends CActiveRecord
```

```
{
```

```
    static public function model($className = __CLASS__)
```

```
    {
```

```
        return parent::model($className);
```

```
    }
```

```
    public function tableName()
```

```
    {
```

```
        return '{{gallery}}';
```

```
    }
```

```
    public $image;
```

```
    public function rules()
```

```
    {
```

```
        return array(
```

```
            array('image', 'file', 'types'=>'jpg, gif, png'),
```

```
        );
```

```
    }
```

```
}
```

```
<?php
```

```
class Jurnalis extends CActiveRecord
```

```
{
```

```
    public $kon_name;
```

```
    public $n_cmi;
```

```
    public $fio;
```

```
    public $mansab;
```

```
    public $tel;
```

```
    public $faks;
```

```
    public $email;
```

```

public $web_sayt;

static public function model($className = __CLASS__)
{
    return parent::model($className);
}

public function tableName()
{
    return '{{jurnal}}';
}

public function rules()
{
    return array(
        array('kon_name, n_cmi, fio, mansab, tel, faks, email, web_sayt', 'required'),
        array('kon_name', 'length', 'max' => 255),
        array('n_cmi', 'length', 'max' => 255),
        array('fio', 'length', 'max' => 255),
        array('mansab', 'length', 'max' => 255),
        array('tel', 'length', 'max' => 255),
        array('faks', 'length', 'max' => 255),
        array('email', 'length', 'max' => 255),
        array('web_sayt', 'length', 'max' => 255),
    );
}

public function attributeLabels()
{
    return array(
        'kon_name' => 'Название выставки/конференции, на которую необходима аккредитация*',
        'n_cmi'      => 'Название СМИ:',
        'fio'       => 'ФИО:',
        'mansab'   => 'Должность:',
        'tel'      => 'Телефон:',
        'faks'     => 'Факс:',
        'email'    => 'E-mail:',
        'web_sayt' => 'Веб-сайт:',
    );
}

```

```

        );
    }

}

<?php
class Menu extends CActiveRecord
{
    static public function model($className = __CLASS__)
    {
        return parent::model($className);
    }

    public function tableName()
    {
        return '{{menu}}';
    }

    public function rules() {
        return array(
            array('name, link', 'required'),
            array('name', 'length', 'max' => 50),
            array('link', 'length', 'max' => 255),
            array('parent', 'numerical')
        );
    }

    public function attributeLabels() {
        return array(
            'name' => 'Menu',
            'link' => 'URL',
            'parent' => 'Otasi'
        );
    }
}
}

<?php

```

```

class GalleryController extends Controller {

    public function actionIndex() {

        $data = Gallery::model()->findAll();

        $this->render('index', array('data' => $data));

    }

    public function actionCreate()
{
    $model=new Gallery;

    if(isset($_POST['Gallery']))
    {
        $model->attributes=$_POST['Gallery'];

        $model->image=CUploadedFile::getInstance($model,'image');

        if($model->save()
        {
            $model->image->saveAs(Yii::getPathOfAlias('webroot').'/images/gallery/'.$model->image);
        }
    }

    $criteria = new CDbCriteria();

    $count=Gallery::model()->count($criteria);

    $pages=new CPagination($count);

    // элементов на страницу
    $pages->pageSize=10;

    $pages->applyLimit($criteria);

    $data = Gallery::model()->findAll($criteria);

    $this->render('create', array(
        'data' => $data,
        'model'=>$model,
        'pages' => $pages
    ));
}

```

```

}

public function actionAgallerydelete() {

    $model = $this->loadgallery();

    $model->delete();

}

public function loadgallery() {

    if(!isset($_GET['id']))

        $this->error_r();

    $model = Gallery::model()->findByPk((int)$_GET['id']);

    if($model == null) $this->error_r();

    return $model;

}

}

<?php

class MenuController extends Controller

{

    public function actionList()

    {

        $data = Menu::model()->findAll(array('order'=>'parent, order_m'));

        $this->render('list', array('data' => $data));

    }

    public function actionAdd()

    {

        $model = new Menu();

        if(isset($_POST['Menu'])) {

            $model->attributes = $_POST['Menu'];

            $criteria=new CDbCriteria;

            $criteria->select = 'max(order_m) AS order_m';

            $criteria->condition = "parent=:col_val";

            $criteria->params = array(':col_val' => $model->parent);

            $row = $model->model()->find($criteria);

            if($row['order_m']!=null){

```