

**ЎЗБЕКИСТОН РЕСПУБЛИКАСИ АЛОҚА, АХБОРОТЛАШТИРИШ  
ВА ТЕЛЕКОММУНИКАЦИЯ ТЕХНОЛОГИЯЛАРИ ДАВЛАТ  
ҚЎМИТАСИ  
ТОШКЕНТ АХБОРОТТЕХНОЛОГИЯЛАРИ УНИВЕРСИТЕТИ  
ФАРҒОНА ФИЛИАЛИ  
«АХБОРОТ ТЕХНОЛОГИЯЛАРИ» КАФЕДРАСИ**

«ҲИМОЯГА»

Кафедра мудири

---

« \_\_\_\_ » \_\_\_\_\_ 2014 й

**КИТОБ САВДОСИНИ АМАЛГА ОШИРУВЧИ WEB САЙТ ЯРАТИШ**

**МАВЗУСИДА**

**МАЛАКАВИЙ БИТИРУВ ИШИ**

**БИТИРУВЧИ:**

**М.Алимов  
610-10 гуруҳ талабаси**

**Фарғона 2014**

## АННОТАТСИЯ

Тошкент ахборот технологиялар университети Фарғона филиали Информатика ва Ахборот технологиялари йўналиши 610-10 гуруҳ талабаси Алимов Мохибжон “Китоб савдосини амалга оширувчи веб сайт яратиш” мавзусидаги битирув малакавий ишига аннотация.

Ушбу малакавий иш замонавий компьютер имкониятларидан кенг кўламда фойдаланган холда PHP дастурлаш мухитидан ёрдамида яратилди. Битирув малакавий иши кириш қисми, аналитик қисм, лойиҳа қисми, тадбиқ қилишни ташкил этиш ва самарадорлиги, меҳнат муҳофазаси, хулоса, илова ва фойда- ланилган адабиётлардан ташкил топган.

Кириш қисмида ахборот технологияларини ривожланиши ҳақида гап боради.

Аналитик қисмда ахборот технологияларни ривожланиши ва тарихи, серверлар, HTML, CSS, Javascript ва PHP дастурлаш тили ҳақида гап боради.

Битирувч малакавий ишнинг лойиҳа қисми 3 қисмдан иборат бўлиб дастурлаш технологияси ва унинг hozirgi кундаги ўрни, PHP дастурлаш мухити ва унинг имкониятлари, CMS, масаланинг қўйилиши, дастурий воситадан фойдаланиш бўйича йўриқнома.

Тадбиқ қилишни ташкил этиш ва лойиҳалар самарадорлиги қисмида дастурини самарадорлиги ва тадбиқ қилиниши кўриб чиқилган.

Меҳнат муҳофазаси бўлимида ишлаб чиқаришдаги зарарли омиллар ва уларнинг организмга таъсири, эргонометрик хавфсизлик, электр хавфсизлигини таъминлаш.

Бундан ташқари битирув малакавий иши якунида хулоса, фойдаланилган адабиётлар рўйхати ҳамда илова ўз ўрнини топган. Битирув малакавий ишида 23 та расмлардан фойдаланилди.

## КИРИШ

Хозирги кунда ахборот технологиялари жаддалик билан ривожланиб бормоқда. Интернет технологияларининг яратилиши турли манбалардан тез ва осон йўл билан ахборот олиш имкониятларини ҳамма учун оддий фуқародан тортиб йирик ташкилотларгача мисли кўрилмаган даражада ошириб юборди. Давлат муассасалари, фан-таълим муассасалари, тижорат корхоналари ва алоҳида шахслар ахборотни электрон шаклда яратиб-сақлай бошладилар. Бу муҳит аввалги физикавий сақлашга нисбатан катта қулайликлар туғдиради. Сақлаш жуда ихчам, узатиш эса бир онда юз беради ва тармоқ орқали бор маълумотлар базаларига мурожаат қилиш имкониятлари жуда кенг. Ахборотдан самарали фойдаланиш имкониятлари ахборот микдорининг тез кўпайишига олиб келди. Бизнес қатор тижорат соҳаларида бугун ахборотни ўзининг энг кимматли мулки деб билади. Бу албатта оммавий ахборот ва ҳамма билиши мумкин бўлган ахборот ҳақида гап борганда ўта ижобий ходиса. Лекин пинхона

(конфиденсиал) ва махфий ахборот оқимлари учун интернет технологиялари қулайликлар билан бир қаторда янги муаммолар келтириб чиқарди. Интернет муҳитида ахборот хавфсизлигига таҳдид кескин ошди:

- Ахборот ўғирлаб қўйиш;
- Ахборот мазмунини бузиб қўйиш, эгасидан изнсиз ўзгартириш;
- Тармоққа ва серверларга ўғринча суқилиб кириш;
- Тармоққа тажовуз қилиш;
- Аввал қўлга киритилган трансаксия(амалларнинг яхлит кетма-кетлиги)ларни қайта юбориш, "хизматдан ё ахборотга дахлдорликдан буйин товлаш", жўнатмаларни рухсат берилмаган йўл орқали йўналтириш.

Маълумки, хозирги кунда веб-саҳифаларни яратиш энг замонавий дастурлашлардан ҳисобланади. Бир неча йил аввал кўплаб веб-саҳифаларини яратувчилар статистик маълумотларгагина қўллар эдилар. Веб-саҳифалар

HTML-файллардан ташкил топиб, фойдаланувчи ўз браузерини ишга юклар ва ўқий олар еди.

Секин аста веб-тармоқида фойдаланувчилар билан алоқа учун формалар яратилиб, қидирув функциялари ҳамда ўзаро ахборот алмашилиш имкониятлари яратила бошланди, лекин саҳифага мурожат қилувчига статистик маълумотлар аввалгидек кўринишда ифодалаб берилди. Энг эътиборли томони тармоқ магазинлари пайдо бўлиб, олди-сотди ишларини шу тармоқ асосида амалга оширувчи дастурлар ишлаб чиқарила бошланди. Шу кунларда эса, веб-саҳифаларини яратиш дастурлари энг асосий куруллардан бирига айланиб қолди. Компютердан фойдаланувчи ўз шахсий иш режаларига эга бўлиб, бу ишларни, яъни электрон почтани текшириш, кўриш режаларини тартибга солиш кабиларни шу саҳифа асосида амалга оширади. Хатто баъзи бир махсус маълумотларни тармоқ орқали миждозга намоиш этиш учун ҳам саҳифа кўринишида узатилиб берилади.

Агар биз веб-саҳифаларини тайёрловчи сифатида асосан HTML, JavaScript ва график дизайнерлик билан шуғуллансангиз ёки биз дастурчи бўлсак, лекин веб-дастурлаш технологияси билан таниш бўлмасак у холда бу фан (соха) биз учундир. У сизга юқори энг замонавий йўналишда кадам кўйишингизга ёрдам беради. PHP тили Perl, ASP ва Java тиллари билан бир қаторда динамик веб-саҳифаларни яратишда юқори ўринлардан бирини эгаллайди.

## I. АНАЛИТИК ҚИСМ

### 1.1. Веб технологияларининг ривожланиши ва тарихи

Ахборот технологиялари ва замонавий техника ютуқлари билан ўзаро алмашиш эҳтиёжи глобал компютер тармоқларини мамлакатлараро ҳамкорлик дастурини амалга оширишни ажралмас қисми қилиб қўйди. Илмий ва маориф мақсадлари бизнес учун кўплаб компютер тармоқлари ташкил этилган. Кўплаб тармоқларни бирлаштира олувчи ва дунё ҳамжамиятига кириш имконини берувчи тармоқ бу-интернет. Интернет фойдаланувчига чексиз ахборот ресурсларини тақдим этади. Ушбу ресурсларга кириш учун мос келувчи амалий дастурий таъминотдан фойдаланиш керак. Дўстона график интерфейсли интернет хизматидан ҳар бир кишининг фойдалана олиши учун имконият яратади. Бундай дастурларнинг кўпи фойдаланувчи учун қулай бўлган Windows OT муҳитида ишлайди. График интерфейсли дастурлар муҳим хусусиятларга эга. Улар фойдаланувчига барча тизимли архитектура билан ва ҳар қандай компютер платформасида сақланадиган маълумотлар билан ишлаш имконини беради.

Замонавий ахборот технологиясига ҳисоблаш техникасидан ва электрон узатиш тизимидан фойдаланмай туриб, замон талабига жавоб берувчи молиявий муассасаларни ташкил этиш мумкин эмас. Шу боис бундай муассасалар дастурли аппарат комплекси «уникасия» воситаси сифатида энг йирик истеъмолчилар ҳисобланади. Ташкилотларнинг алоҳида автоматлаштирилган комплексларини боғловчи глобал тармоқлар миллий ва халқаро даражада ҳисоб-китоблар ўтказиш имконини беради. 20 йил аввал АҚШ нинг мудофаа министрлиги тармоқ ишлаб чиқди. Бу интернет нинг бошланиши эди ва у арпанет дейиларди.

Арпанет тажриба тармоғи бўлиб, у харбий ишлаб чиқариш сферасидаги илмий текширишлар учун яратилган. Шунингдек бирмунча носозликларда тармоқлар бемалол ишлаши учун ҳам яратилган. Бу интернетни тузилиши ва

структурасини тушуниш учун калит беради. Арпанетда ҳар доим манба компютери ва қабул қилувчи компютер орасида алоқа бўлади.

Прион тармоғи ишончсиз дейиларди, чунки тармоқнинг ҳар қандай қисми хоҳлаган пайтда учиб кетиши мумкин. Фақат тармоққа эмас, шунингдек боғланаётган компютерларга ҳам алоқани тузиш ва қўллаб туриш юкланилади. Асосий принциплардан бири шуки, ҳар бир компютер бошқа бир компютер билан тенгма-тенг боғланиши мумкин. Тармоқда маълумотларни бериш интернет -IP баёни асосида ташкил қилинган.

### **Тармоқлар**

МДХ да янги тижорат тармоқларини оммавий равишда барпо этиш 1991-йилдан бошланди ва доимий давом этиб келмоқда.

Тармоқ хизматига кирадиган ҳудуд жадаллик билан кенгаймоқда, тақдим этиладиган сервис даражаси ва сифати ўсмоқда.

Телекоммуникация хизматининг ривожланиши хўжалик ва тижорат фаолиятини ҳаракатлантиришнинг, иқтисодий ривожланишнинг муҳим шартларидан биридир. Фаол маркетинг ва техник сиёсатни олиб борувчи айрим тармоқларни шарҳлаб ўтаимиз:

#### ***Relkom тармоғи.***

1990 йилда Атом энергияси институтининг ва бир қанча илмий тадқиқот институтлари (ИТИ), Unix - компютерлари тизимини ишлаб чиқарувчи мутахассислар ва фойдаланувчиларни бошқарувчи унча катта бўлмаган тармоқ ташкил этилди. Тез орада тармоқ 30 дан ортиқ ташкилотларни бирлаштирди ва интернетнинг - ғарбий европадаги EUnet тармоғига уланди, ҳамда рўйхатга олинди.

Ҳозирда Relkom умумий мақсадларга мўлжалланган тармоқ сифатида ривожланмоқда. У илмий ва тижорат ташкилотларини, давлат идоралари ва муассасаларини бирлаштиради. Техник жиҳатдан Relкомни ўз минтақасида

фойдаланувчилар ўртасида маълумотлар алмашишини таъминловчи ва зарур пайтда минтақавий алоқа узеллари жамламаси деб тасаввур этиш мумкин. Relkom минтақа ичида ва EUNET, интернет, Bitnet абонент тармоқларида электрон почта хизматини таъминлайди. Тармоқ Usenet телеконференцияларидан фойдаланиш имконини беради.

Relkom орқали Relis тижорат ахборот тизими билан ишлаш осон ва қулай. Бу тизим кундалик янгиликларни, дайжестлар, таҳлилий шарҳлар ва бошқа кўпгина мавзуларда ахборотларни тақдим эта олади.

### ***Sprintnet тармоғи.***

Sprintnet маълумотларни узатиш тармоғи бўлиб, ўнлаб мамлакатларнинг юзлаб шаҳарларига кириш узелларига эга.

Sprintnet тармоғига кенг соҳадаги маълумотларга эга мингта маълумот базаси уланган. Sprintnet тармоғи ахборотларни катта тезликда алмашиш имконини беради.

### ***Glasnet тармоғи.***

Ошкоралик кучайган пайтда ташкил этилган ва уни ўз атамасига айлантирган (Glasnet - ошкоралик) Glasnet тармоғи 1990 йилдан бошлаб Россия аҳолисига ва МДХдаги бир қанча давлатлар учун жаҳон интернет компьютер тармоғига кириш хизматини таъминлайди. Хусусий мижозлар ва тармоқ хизматидан фойдаланувчи кичик бизнес вакилларига эътибор бериш- Glasnetнинг ажралиб турувчи хусусияти саналади.

### ***Sovam Teleport тармоғи.***

Sovam Teleport Халқаро компьютер ахборот тармоғи San Francisco Moscow Teleport (AKSH), Cable & Wireless (Буюк Британия) компаниялари ҳамда Автоматлаштирилган тизимлар институти (Россия) томонидан 1990 йилда ташкил этилган. Тармоқ аввало жорий вақт режимида телекс ва

телефакс халқаро ахборот мўлжалланган. Халқаро тармоққа чиқиш ижарадаги йўлдан ғарбий европага алоқа қилиш каналлари орқали чиқилади.

### **Потенциал истеъмолчилар, интернетга йўл.**

Интернетнинг фойдалилиги ҳисоблаш техникасининг ривожланишига нисбатан 10 йил кечиқиш билан ошиб келган. 80-йиллар охирида шахсий компьютерлар пайдо бўлиши информатиканинг билимдонлар салтанатидан кенг оммага олиб чиқди. Ўзининг ривожи ва тарқалишида интернет худди шундай жараёни олиб боради. Ҳисоблаш техникасига ўхшаб, интернет ҳам экспертлар ўйинчоғидан то кундалик эҳтиёж асбобигача йўл босиб ўтди. Уларнинг шу йўлни босиб ўтиш жараёнлари ҳам ўхшашдир. Тармоқдан фойдаланиш аста-секин соддалашди, бунинг сабаби, бир томондан, ускуналар яхшиланиши, иккинчи томондан тармоқнинг ўзини тезкор ва ишончли бўлиб қолишидир. Аввалида ишонмаганлар ишонч ҳосил қилиб, интернет дан фойдалана бошладилар. Бугинги истеъмолчилар, янги ресурс ва янги ускуналарга катта эҳтиёжни вужудга келтирдилар. Эски усуллар ривож топди, янги ресурсларга ўтиш учун янги усуллар яратилди, бу эса тармоқдан фойдаланишни осонлаштирди.

Худди шунингдек янги истеъмолчилар ҳам интернет фойдасини тушундилар. Жараён қайтарилди. Бу айланма ҳаракат шу кунгача ривожланиб келмоқда.

Умуман, барча интернет фойдаланувчилари бир нарсани истайдилар: компьютерлар орқали мулоқот қилиш. Интернет ресурслари ҳақида унутиш осон, аммо улар жуда муҳим худди яхши компьютердек. Интернет тинчликсевар ва дўстликсевар мухит. Бу ерда худди ўзингиздек инсонларни учратиш мумкин. Сиз шубҳасиз доимий истеъмолчисиз, агарда масалан сиз:

- ❖ Генетика картасига муҳтож биолог бўлсангиз:
- ❖ Руҳий тушуниш ва ўртоқлик истаётган бўлсангиз:
- ❖ Ислоний давлатдаги буддист бўлсангиз:

❖ Ефирдаги поп-мусиқадан бедор бўлган, классика ва рок мухлиси бўлсангиз:

❖ Нозик муносабатлар ҳақида қонун билан сирлашмоқчи ёки психотерапевт бўлсангиз:

Интернет шу барча инсонларга Ўз ҳамфикрларини топишда ажойиб имконият тақдим этади. Хоҳлаган мавзуга электрон мулоқот клуб топиш мумкин, яъни жуда ҳам осон (ҳозир булар бир ярим мингга яқин), ёки ҳали ҳеч ким билмайдиган янги клуб ихтиро қилиш мумкин. Интернет яна бу инсонларга янги компьютер ресурсларига йўл очиб беради. "Билим" жамияти лектори NASA компютери билан боғланиш мумкин, у эса ўтмиш келажак ва ҳозирги пайтдаги космик билимлари ва АҚШ дастурлари ҳақида маълумот беради. Руҳоний эса, Қуръон ,Таврот ёки Инжилдан керакли парчалар топиши мумкин. Юрист "Иран-контрас" ишида АҚШ олий судининг доклад нушаларини ўз вақтида топиши мумкин. Саккизинчи синф ўқувчиси

В. Сойнинг лирикаси ҳақида тенгдошлари билан муҳокама қилиши мумкин. Бу фақат бошланиши.

### **Интернетдаги мавзули доменлар**

<b>Домен</b>	<b>Мавзу белгиси</b>
com.....	Тижорат корхоналари
edu.....	Ўқув муассасаси( масалан, университет)
gov.....	Нохарбий ҳукумат муассасаси
mil.....	Харбий муассасалар
net.....	Тармоқ ташкилотлари
org.....	Бошқа ташкилотлар

## *Интернетдаги географик доменлар*

<b>Домен</b>	<b>Мамлакатлар</b>
au-----	Австралия
sa-----	Канада
de-----	Германия
fr-----	Франсия
jp-----	Япония
it-----	Италия
ru-----	Россия
uk-----	Буюк Британия
us-----	АҚШ
uz-----	Ўзбекистон

### **Файлларни юбориш (FTP).**

FTP-File Transfer Protokol - файлларни юбориш баёни - файлларни биринчи компютердан иккинчисига юбориш қоидаларини аниқлаш баёни. Шунингдек FTP-амалий таъминотдаги дастурнинг номи. FTP-баёни асосан файлларни юбориш учун ишлатилади. ftp-ни қўллаш нуқтаи назарига кўра кўпинча Telnetга ўхшаб кетади. яъни ftp билан ишлаш учун, яъни ўзингизга файлларни ўтказиш, масофадан туриб машинага кириш учун кириш номини ва тўғри келадиган паролни билишингиз керак. Кириш камида диал-уп (чақириш бўйича) типига бўлиши шарт. FTP ни қўллаш учун сеансни ўтказишда қўлланиладиган иш машинасининг номи билан FTP буйруғини бериш лозим.

Шунингдек FTP масофадан туриб машиналарда файлларни излашга имкон беради, яъни каталогдан-каталокка ўтиш, бу каталог ва файлларни таркибини кўриб чиқиш. Файлларни ва унинг гуруҳларини, шунингдек бутун каталогни, ҳоҳлаган чуқурликка солинган қисм каталогларни ҳам, барчасини

юборишга имкон беради. Маълумотларни иккилик, ахборот, ёки ASCII (текст) каби кўринишдаги юборишга имкон беради. ASCII-юбориш, бошқа кодли алфавитга компьютердаги текстни юборишда маълумотларни автоматик кодлашга имконият беради, бу матннинг олдинги ўқиш кўринишини сақлайди. Маълумотларни юборишда қисиш ва ундан кейин олдинги кўринишга келтириш мумкин. Шунингдек FTP нинг аноним FTP си ҳам мавжуд. Анонимлиги шундаки, агар FTP нинг иш бошланишида ўзини тўғри идентификация қилиши, кириш номини киритиш ва паролни кўрсатиш керак бўлса, FTP нинг бу турини қўллаб турадиган машинага кириш ва иш бошлаш учун буни қайтариш сўралмайди. FTP баёни, буйруқлар баёнига ва узатиладиган маълумотларга бўлинади. Буйруқлар 4% фоизни ташкил этса, маълумотлар бутун тармоқ графигини ўртача 40% ни ташкил этади. Шунингдек FTP ни бирмунча манба(манба(сервер))ларнинг e-mail пакет режимида ишлатса бўлади, лекин тўғридан-тўғри мулоқот жуда ноқулай ва ишни секинлаштиради. Аммо йўқдан кўра яхши.

### **Электрон манзил. e-mail ни имкониятлари.**

E(mail)-( Elektronik mail).

Бизни давлатимизда ҳам бугунги кунда интернет энг замонавий ва келажаги бор алоқа. Бахолашларга қараганда, бугунги кунда дунёда электрон почтадан 100 миллиондан ортиқ одамлар фойдаланади. Дунё бўйича электрон почта графика (баён SMTP) тармоқнинг 3,7% ни ташкил қилади.

Электрон почта тез ривожланиш сабаби, бугунги кунда зарурлиги ва кўпинча уланиш йўли (модемга кириш) UUSR имконияти бор. Электрон почта бу оддий почтани замонавий тури. Шу почта орқали сиз ҳар хил хабарномаларни жўнатиш ва қабул қилиш имкониятингиз бор. Хатларга жавоб бериш, хатларни кўпайтириб бирданига ҳар-хил манзилга жўнатиш, оддий манзилни ўрнига, мантиқий исмли манзилдан фойдаланиш, қабул қилаётган маълумотнома, хабарнома ва бошқа ҳужжатларни почта яшигингизда бўлимларга ажратиш, хатларга текст файлларни кўшиш,

мухбирлар гурухи билан мулоқотда "Ойнома почта" орқали тартиб билан фойдаланиш ва бошқа турлари чексиз.Керакли бўлимларни манзилини ҳамда тармоқни чиқадиган форматини билиб туриб, интернет орқали сиз қўшни тармоқларга почтани жўнатишингиз мумкин.

Электрон почта билан ишлаб туриб, сиз осинхрон режимда FTP дан фойдаланишингиз мумкин. Ҳозирги кунда кўп манба (сервер)лар шу хизматга мослашган.Шу буйруқ тизимига тўғри келадиган электрон почтани юқоридаги хизмат ташкилотига жўнатасиз. Мисол: Ташкилотга листинг бериш ёки сизга керакли файл жўнатиш электрон почта орқали шу листинг ёки файл орқали сизга автоматик жавоб келади. Бу режимда оддий FTP ҳамма буйруқ тўпламлардан фойдаланиш мумкин. Шунақа манба(манба(сервер))лар борки, файлни FTP орқали фақат ўзидан эмас, хоҳлаган FTP манба(манба(сервер))дан сиз кўрсатган йўлланма билан электрон почта орқали файл олиши мумкин.

Электрон почта орқали мулоқотлар ва телеконференциялар ўтказишга имкони бор. Бунинг учун машинада ўрнатилган mail пefестор-лардан фойдаланиш керак.

Сиз кўрсатилган жойга хабарнома жўнатиб (қайси мулоқот ёки конференция), орқасидан қатнашчиларни хабарномасини нусхасини қабул қиласиз.

Почта ойнаси электрон хатни қабул қилгандан кейин, нусхаларини ҳамма қатнашчиларга жўнатади.

### 1.1.1. HTML, CSS, JavaScript ва PHP дастурлаш тили

Энди текстлар билан ишлашни кўриб чихамиз. Биз саҳифамизга маълумотларнинг асосий қисмини текст сифатида қўямиз. Шунинг учун текстларни тартиб билан чиройли дизайнда намоиш этиш саҳифанинг ўқимишли бўлишига ёрдам беради. Текстлар ёзилиши каторида янги катордан, абзатсдан ёзилиши мумкин. Бунинг учун Демак текстларга ранг танлаш ҳам алоҳида нозик дид талаб қилади. Биз текстга `<p>` ва `</p>` тегларидан фойдаланамиз.

Матнни қалин, *ётиқ* ва тагига чизилган ҳолда ёзиш учун `<b></b>`, `<i></i>` ва `<u></u>` теглари орасида ёзиш кифоя. Текст орқали бошқа саҳифага йўл корсатмоқчи бўлсангиз албатта HREF тегида фойдаланасиз.

Мисол учун  
`<a href="http://www.uz"> www.www.uz</a>`

Кўринишда йўл кўрсатсак албатта www.www.uz саҳифаси очилади. Ана энди саҳифага расм қўйиш учун фойдаланиладиган тег IMAGE билан танишамиз. IMAGE теги фақатгина саҳифага расмларни жойлаштириш учун фойдаланилади. Расмлар саҳифаларни қизикроқ ва тасавурга бойроқ қилиб кўрсатади. Демак расм қўйиш учун `<IMAGE>` тегини киритамиз ва `src=""` қўйиб қўштирноқ орасига расм олинадиган манзил кўрсатилади. Албатта `width=""` ва `height=""` буйруқларида расмнинг кенглиги ва баландлик параметрлари кўрсатилади. `alt=""` тегидан фойдаланиб расмга изоҳ ёзиш мумкин.

#### CSS нима ўзи у?

CSS - бу стиллар билан ишлай оладиган каскадли стиллар мажмуасидир. Веб саҳифаларга хар хил стиллар бериш учун ишлатилади. HTML хужжат ичида теглар билан бирга фойдаланилади. CSS кодларини компиляция қилиш учун қандайдир компилляторлар, қандайдир дастурлар, ва бу код тушуна оладиган қандайдир редакторлар керак бўлмайди. Бу кодларни HTML сингари веб браузернинг ўзи компиляция қилади ва натижани чиқариб беради. Бу стиллар жадвали(CSS) тўлиқ ҳолатда 1997

йилда ташкил топиб, "WWW Consorcium" ида кўллаб қувватланган ва фойдаланишга тақдим этилган. Ўша пайтларда HTML нинг 3 версияси оммабоп эди. CSS ни дастлаб Netscape Navigator 4.0 ва Internet Explorer 4.0 броузерларида кўриш мумкин бўлган. ҳозирда барча броузерлар CSS да ёзилган кодларни тушуна олади. Бу стилларга оид дастурлаш тилини ўрганиш содда бўлиб, янги стиллар кўшилиши натижасида кенгайиб бормоқда.

CSS ни HTML хужжати ичида ишлатилишини бирин кетин кўриб ўтамиз:

1.Веб саҳифада ишлатиладиган CSS кодлари орқали ҳосил қилинадиган стилларни ҳаммасини

битта файлга ёзиб, уни серверда сақлаймиз. Сўнг, саҳифадан, сервердаги сақлаб қўйилган файлга мурожаат қилиб, керакли стилни оламиз. Бу усул катта ҳажмдаги веб саҳифалар яратишда фойдали ҳисобланади. Файлга мурожаат, <HEAD>теги ичида амалга оширилади ва қуйидаги кўринишга эга бўлади.

```
<link rel=stylesheet type="text/css" href="url">
```

<link> теги саҳифа бирор файлга мурожатни амалга оширмоқда дегани, REL - файл қандайдир стиллардан иборат дегани, TYPE - стиллар CSS кодларидан иборат ва файл жойлашган манзил ёзилади.

1.CSS кодлар веб саҳифа ичидаги <HEAD> теги орасида берилади. Бунда яратилган стиллар шу саҳифага тегишли бўлади, бу дегани фақат шу саҳифада ишлатилиш мумкин дир.

```
<head>< style type="text/css">
```

```
<!--
```

```
A {text-decoration:none; }
```

```
-->
```

```
</--STYLE></head>
```

2.Навбатдаги тури, бу ҳар бир элемент учун алоҳида тег ичида стил бериб чиқишдан иборат. Мисол учун, "p" теги ичида бирорматн ёзилса, шу матн учун тег ичида алоҳида стил берилади ва бу стил шу тег учун хусусий бўлади. Бу усулдан фойдаланишни маслаҳат бермиман, чунки бу усул кодларни чалкаш ва тушунарсиз бўлишига олиб келади. Шошилиш вазиятларда ёки стил қай тартибда намоён бўлиши кўриб олиш учун бу усулдан фойдаланишни маслаҳат бераман.Ундан ташқари <style> теги орқали CSS файлни импорт қилиш мумкин бўлади.

```
@import: url(mystyles.css);
```

CSS стилларини ишлатишда, қуйидаги гуруҳлаш усулларида фойдаланиш, тартибли веб саҳифа тузишга олиб келади:

-Кодларни ихчамлаш учун, ҳар ҳил элементлар учун бир хил типдаги стилларни гуруҳлаб ишлатиш мақсадга мувофиқдир.

```
H1 {font-family: Verdana}
```

```
H2 {font-family: Verdana}
```

Юқоридаги стилни гуруҳлаб қуйидагича ёзамиз.

```
H1, H2 {font-family: Verdana}
```

-Гуруҳлашда бир хил элемент учун бир неча стилларни бирлаштириш мумкин.

```
H2 {font-weight: bold}
```

```
H2 {font-size: 14pt}
```

```
H2 {font-family: Verdana}
```

Барча стилларни биттага йғамиз.

```
H2 {font-weight: bold; font-size: 14pt; font-family: Verdana;}
```

- Баъзи стиллар битта стил ичида берилиши ҳам мумкин.

```
H2 {font: bold 14pt Verdana}
```

CSS стиллари билан ишлаганда, элемент ичидаги элементлар ҳам ўзидан юқори турган элемент стилини қабул қилади, яъни ўзидан олдинги элемент "ота" вазифасини бажаради ва ўз "фарзандлари"га ҳам ўзида борини беради. Мисол учун, <P> теги ичидаги матн кўк рангда ёзилиши лозим бўлса

(P {color:blue}), <P> теги ичидаги <em> тегига тегишли бўлган матн ҳам кўк рангда ёзилади. Баъзи стиллар фақат яқка тартибда ишлайди, яъни ички тегларга стиллари ўтмайди(мисол учун, background), шунинг учун барча стилларни бирма бир ишлатиб, ўрганиб олиш лозим. <BODY> тегига стил бериб, дастлабки ўрнатилиш жараёнини ҳосил қилиш мумкин. Шунда бирор элементга стил бериш эсдан чиқиб қолса, <BODY> тегига таълуқли стил эсдан чиққан элементга ўрнатилади.

```
BODY {color: green;  
font-family: "Verdana"; background: url(joke.gif) white; }
```

### **CSS 3 ҳақида**

CSS - бу стиллар билан ишлаш учун мўлжалланган кодлар мажмуасидир. Хозирги кунга келиб янги CSS3 версияси ишлаб чиқилмоқда, лекин хали кўп браузерлар бу стилларни акс эттира олмаяпти. Шунинг учун CSS3 яратувчилари браузер ўртасида келишмовчиликларни олдини олиш мақсадида браузер турига қараб ҳар хил префикслар ишлаб чиқишди ва бу префикслардан CSS3 тўлиқ яратилмагунча ишлатиш тавсия этилган, CSS3 тўлиқ яратилгандан сўнг бу префикслар олиб ташланади. Бу префикслар қуйдагилар:

- Safari va Chrome browserlari -webkit-
- Opera browseri -o-
- Firefox browseri -moz-
- Internet Explorer(IE) browseri учун эса -ms-

Мен бу префиксларни ишлатмаган ҳолда мисоллар келтирдим, агар сизда бу мисоллар ишламаса олдига префиксларни қўйиб ишлатиб кўринг. CSS3 да энг асосий қўшилган стиллардан бири бу анимациялардир, яъни стиллар алмашилиш жараёни бирданига эмас, балки секин асталик билан содир бўлишидир. қуйида CSS3 версиясига янги қўшилган баъзи стилларни мисол тариқасида кўрсатиб ўтаман.

"Элементларни(матн, расм, фон,..) оч(прозрачный) кўринишда акс эттириш.

Background-color: rgba(10,0,255,0.7); Бу ерда саҳифа фониға ранг бериб, шу рангни 0.7 даражали оч тусға келтирилмоқда, танишинг CSS3 нинг янги элементи rgba."Элементларнинг исталган бурчагига(қисмиға) бир вақтнинг ўзида ҳар хил расмларни жойлаштириш. Бу сайтни верстка қилиш жараёнини енгиллаштиради.

```
background:
url(top.gif) top left no-repeat,
url(center.png) top 11px no-repeat,
url(bottom.png) bottom left no-repeat,
url(middle.png) left no-repeat;
```

"Resize буйруғи. Бу буйруқ орқали фойдаланувчи исталган элементнинг ўлчамини ўзгартириш мумкин бўлади.

```
div.resize {
width: 25px;
height: 35px;
resize: both;
}
```

"CSS3 нинг яна бир янгилиги бу - элементлар(шакллар, блоклар) бурчагини исталган радиусда буриш мумкинлигидир. Кўп веб сайтлар форма яратиш жараёнида, форманинг бурчакларини буришни верстка пайтида бирор расм орқали амалға оширишади, бу эса қўшимча иш ва вақт йўқотишға олиб келади, CSS3 да бу муаммо бартараф этилган ва қуйидагича амалға ошириш мумкин бўлади.

```
#forma {
border-bottom-right-radius: 2em;
border-bottom-left-radius: 1em;
border-top-left-radius: 5em;
```

```
border-top-right-radius: 3em;
}
```

"CSS3нинг соялар билан ишлаш қисми. Барча "p" теглари учун соялар хосил қилиш.

```
p {
text-shadow: #003471 // 2px /* 2px /* razmitie*/;
}
```

"Шрифтлар. Интернетда кўп фойдаланувчилар вердана шрифти билан ишлайди. Нега? Чунки бу шрифт барча компьютерларда мавжуд ва броузерда чиройли кўринишга эга. Агар стилларда қўлланилган шрифт фойдаланувчи компютерида мавжуд бўлмаса, броузер маттни исталган бошқа шрифтда кўрсатиши мумкин. Бу эса шрифтлар ранг баранглигига олиб келади. CSS3 да шрифтлар билан ишлаш учун янги команда @font-face.

```
@font-face.
```

```
@font-face {
```

```
font-family: shrift_akm; src: url('http://blabla.uz/fonts/shrift_akm.ttf');}
```

```
h1 {font-family: shrift_akm;}
```

"Веб саҳифада колонкалар. Бунисига нима дейсиз? Элемент ичидаги матнларни бир неча устунли колонкаларда чиқаришингиз мумкин.

```
div {column-width: 15em; column-gap: 2em; /* Яшил рангда */ column-
rule: 4px solid red; /* Қизил рангда */ padding: 5px; /* Қора рангда */}-----
```

## PHP тарихи



1994 йили PHP тилиниг яратувчиси Расмус Лердорф ўзиниг сайтига меҳмонлар киришини ҳисоблаш учун Perl дастурлаш тилада махсус қобик ёзиб амалда қўллаган. Кўп ўтмай қобикни ишлаш унумдорлиги жуда паст ва секинлиги аниқланганидан сўнг, дастурларни янгидан "C" тилида ёзиб чиқишга тўғри келади. Кейин, дастлабки дастур кодлари муаллиф тарафидан барчага кўриш учун серверга нашр қилинган.

Сервер фойдаланувчилари кодлар билан қизиқиб, уни ишлатиш мухлислари ҳам пайдо бўлган. Ҳадемай, бу дастурлар алоҳида лойиҳага айланиб, 1995 йилнинг июн ойида дастурий маҳсулот PHP (Personal Home Page) номи билан биринчи нaшри чиқарилди. Имкониятлари оддий ва содда бўлиб: бир неча буйруқларни тушунадиган код анализатори(текширувчиси), меҳмонлар ҳисоблагичини, китобини, чатини яратиш учун фойдали дастурлар тўпламидан иборат бўлган.

1996 йил апрел ойида дастурлар жиддий қайта ишланганидан сўнг, PHP/FI (Personal Home Page / Forms Interpreter) номи билан маҳсулотнинг иккинчи нaшри пайдо бўлди. Бу маҳсулот HTML-код ичига ёзилиб, HTML-формаларни қайта ишлаб, ҳозирги PHP дастурлаш тилининг таянч имкониятларини ичига олган. PHP/FI код ёзилиши Perl тилига жуда ўхшаган, лекин соддароқ бўлган.

1997 йили PHP/FI 2.0 нaшри чиқди. Ўша пайтда бу маҳсулот билан дунё бўйича бир неча минг одам фойдаланиб, тахминан 50 минг домен бўлиб, интернетнинг 1%-ни ташкил қилди.

1997 йилда Энди Гутманс ва Зив Сураски PHP/FI кодини бошқатдан ёзиб чиқишди, чунки эски код улар ишлатаётган электрон тижорат тизимлари учун яроқсиз эди. Эски коднинг муаллифларидан ёрдам олиш учун улар бирлашишни таклиф этиб, PHP3 номли лойиҳани PHP/FI -ни расмий вориси деб эълон қилишди. Янги лойиҳа уюшгандан кейин PHP/FI лойиҳаси ишлаб чиқарилиши тўхтатилган.

PHP 3.0 нинг энг кучли тарафларидан бири унинг кенгайтирала олинадиган ядроси(тизимнинг бошарув қисми) бўлиб, бундан ташқари, маълумот жамғармалар билан, турли протоколлар ва интерфейслар билан биргаликда ишлаш кенг имкониятлари яратилди. Муваффақиятга эришишга анча аҳамиятли факт бу янги тилни бойлиги ва объектларга мўлжалланган дастурлашни қўллай олиши. Янги лойиҳа билан бирга нафақат тилни ташқи, ички тузулиши ўзгарди, балки ўзини номи ҳам. Энди PHP қисқартмаси “PHP: Hypertext Preprocessor” маъносини англатиши билдирилди.

1998 йилнинг охирида PHP фойдаланувчиларнинг сони ўн минглардан ошди. Юз мингдан ошиқ веб-сайтлар бу тилни қўллашни эълон қилишди. Тахминан Интернетнинг 10% серверларида PHP 3.0 ўрнатилган эди.

1998 йилнинг июн ойда PHP3 тўққиз ой оммавий текширувидан кейин расман эълон қилинди. Шу йилнинг қишида Энди Гутманс ва Зив Сураски PHP ядросини қайтадан ишлаб чиқаришни бошлашган. Уларнинг асосий вазифаси PHP тизимининг унумдорлигини кўтариш ва коднинг модулларини яхшилаш эди. 1999 йилнинг ўрталарида биринчи марта тақдим қилинган янги ядро “Zend Engine” деб номланган (“Zend”: муаллифлар “Зив” ва “Andi” исмлардан ташкил топган). Унинг асосида тузилган янги тил PHP4 2000 йилнинг май ойида расман чиқарилган. Унумдорлик яхшилангандан ташқари, PHP 4.0 муҳим янгиликларга эга бўлиб, сессияларни қўллаш, буферли чиқариш, киритилган маълумотларни ҳавфсиз қайта ишлаш ва яна бир неча янги тил тузувчиларини пайдо бўлишидан иборат.

Ҳозирги кунларда “Zend Engine” қайта яхшиланиб PHP5 тили ишлаб чиқарилди. Асосий ўзгаришлар объектларга мўлжалланган дастурлаш моделида бўлиб, тилнинг имкониятлари янада кенгайтирди.

Ҳозирда PHP тилини юз минглаб дастурчилар ишатиб, бир неча миллион сайтлар яратилиб, Интернетнинг 20% доменларини ташкил этмоқда.

### **PHP тарихининг жадвали**

Нашр	Сана	Изоҳ
PHP 1.0	1995 йил, 8 июн	Расмий номи “Personal Home Page Tools (PHP Tools)”. Бу “PHP” қисқартмани биринчи марта ишлатилиниши.
PHP 2.0	1996 йил, 16 апрел	Веб саҳифаларни “Тез ва осон ишайдиган курул” деб эълон қилинган маҳсулот.
PHP 3.0	1998 йил, 6 июн	Бир киши ишлаб чиқарувчи лойиҳадан кўп

		ишлаб чиқарувчилар лойиҳасига айланди. Энди Гутманс ва Зив Сураски кодни бошқатдан ёзиб чиқишди.
PHP 4.0.0	2000 йил, 22 май	Янги ядро “Zend Engine” асосида тузилган
PHP 4.1.0	2001 йил, 10 декабр	Тил конструктсиясига суперглобал имкониятлар қўшилган (\$_GET, \$_POST, \$_SESSION, etc)
PHP 4.2.0	2002 йил, 22 апрел	Хавфсизликни ошириш мақсадида тизимнинг созлаш “Register_globals” параметри очирилган
PHP 4.3.0	2002 йил, 27 декабр	“CGI” (Common Gateway Interface)га қўшимча “CLI” (command line interpreter) технологияси киритилган.
PHP 4.4.0	2005 йил, 11 июл	phpize va php-config скриптлари учун қўлланма варақлари қўшилган.
PHP 5.0.0	2004 йил, 13 июл	Zend Engine II – янги объект модели киритилган.
PHP 5.1.0	2005 йил, 24 ноябр	Zend Engine II ишлаш унумдорлиги яхшиланган.
PHP 5.2.0	2006 йил, 2 ноябр	Кенгайтирадиган филтр модулларининг PHP тизимида ишлатиш имкониятлари очиқ ҳолати ўрнатилган (default).

### 1.1.2. Системалар, серверлар ва маълумотлар базалари

PHP турли хил системаларда ишлай олади. У система Windows, Unix нинг кўплаб версиялари, шунингдек Linux ва хатто Macintosh бўлиши мумкин. PHP кўплаб тармоқ серверларида, хусусан Apache, Microsoft Internet Information Server, Web Site Pro, Iplanet Web Server ва Microsoft Personal Web Server – ларда ишлаши мумкин. Агар ўзимиз тузган дастурларимизни

Windows тизимида текширишни хохлаш охири санаб ўтилган сервердан фойдаланишимиз мумкин, хатто Apache сервери Windows системаси бошқарувида ишласа ҳам.

PHP интерпретатори ёрдамида дастурни алоҳида мустақил кўринишда компиляция қилиш мумкин. У ҳолда дастурни мустақил ишга тушириш мумкин. PHP тилини яратишда маълумотлар базаси билан боғланиш талабларини алоҳида эътиборга олинган. Кўплаб маълумотлар базаларини PHP да ўқиш мумкин. Масалан буларга Adabas D, InterBase, Gold, dBase, mSQL, Sybase, Empress, MySQL, Velocis, FilePro, Oracle, Unixdbm, Informix ва ҳоказоларни келтиришимиз мумкин. Шунингдек, PHP ODBC стандартини ҳам ўқий олади. Ушбу қўлланмада эса биз Linux, Apache ва MySQL серверлар асосида фикр юритамиз. Бу учта дастурлар мажмуалари кенг фойдаланиш мумкин бўлган.

## **1.2 PHP дастурлаш тили**

### ***PHP/FI***

PHP бошқа продукт, PHP/FI ривожлантирилиши натижасида яратилди. PHP/FI 1995 йилда Расмус Лердорф томонидан ўзининг “Online-gezuyme”сига мурожаатни кузатиш учун Perl-скриптлар содда тўплами сифатида яратилди. У бу скриптлар тўпламини ` Personal Contents Page Tools деб атади. Катта функционаллик талаб қилингани учун, Расмус маълумотлар базаси билан ишлаш имконига эга бўлган C кенгайтирилган реализациясини яратди ва фойдаланувчиларга содда динамик веб-иловалар яратиш имконини берди. Расмус ҳар бир фойдаланувчи кенгайтириш ва ўзгартириш имкониятига эга бўлиши учун PHP/FI кодини кенг оммага эълон қилди. PHP/FI, Personal Contents Page / Forms Interpreter, ҳозирги PHP асосий функционалигига эга эди. У Perl-каби ўзгарувчилар, форма ўзгарувчилари автоматик интерпретацияси ва HTML га қўлланган синтаксисга эга эди. Синтаксис ўзи Perl ни эслатарди, фақат чекланган, соддалаштирилган ва тўла бўлмаган. 1997 йилда PHP/FI 2.0, C-реализация

иккинчи версияси, бутун дунё бўйича бир неча минг мухлисларга эга бўлиб, тахминан 50,000 доменларга ўрнатилган эди. Бу ҳамма интернет доменларнинг 1% ташкил қилар эди. Кўп одамлар ўзларининг код блокларини бу лойиҳа учун таклиф қилганлари учун, у бир кишининг лойиҳаси бўлмай қолди. PHP/FI 2.0 расмий равишда фақат 1997 йил ноябрида чиқарилди.

Унгача у асосан бета-релизлар шаклида мавжуд эди. Шундан сўнг кўп ўтмасдан биринчи alpha PHP 3.0 пайдо бўлди.

### ***PHP 3***

PHP 3.0 бугунги PHP га ўхшаган биринчи версия эди. Уни Анди Гутманс ва Зив Сураски 1997 йилда тўла қайтадан ёзилган тил сифатида яратдилар, чунки улар PHP/FI 2.0 тилини ўзларининг е-commerce-иловаларини яратиш учун этарли имкониятга эга эмас деб топдилар. Кучларни бирлаштириш учун, Анди, Расмус ва Зив PHP 3.0 ни PHP/FI 2.0 расмий вориси сифатида яратдилар ва эълон қилдилар. Натижада PHP/FI 2.0 ривожланиши ўхтади. PHP 3.0 энг кучли томонларидан бири уни кенгайтириш имкони эди. Чекли фойдаланувчиларга ҳар хил маълумотлар базалари, протокол ва IP лар учун мустахкам инфраструктура, ҳамда кенгайтириш имкониятини яратгани учун, ўнлаб фойдаланувчиларни Янги кенгайган модуллар яратишга ундар эди. Балким PHP 3.0 ўта машхурлиги сабаби шундадир. PHP 3.0 асосий хусуиятларидан бири объектга йўналтирилган синтаксис эди. Тилга PHP: Hypertext Preprocessor деб ном берилди. 1998 йил охирига келиб PHP ўнг минглаб фойдаланувчилар ва юз минглаб веб-сайтлар учун асос бўлиб қолди. Энг машхур бўлган пайтда PHP 3.0 тахминан Интернет веб-серверларининг 10% га ўрнатилган эди. PHP 3.0 расмий равишда 1998-июнида 9 ойлик ошкора тестлашжандан сўнг чиқарилган эди.

### ***PHP 4***

1998-йил қишида PHP 3.0 расмий эълон қилингандан сўнг, Анди Гутманс и Зив Сураски катта амалий дастурлар билан ишлашда

унумдорлигини ошириш ва PHP кодли базаси модуллигини ошириш мақсадида PHP ядросини қайта ишлашга киришдилар. Бундай Амалий дастурлар яратиш PHP 3.0 да мумкин эди, лекин PHP 3.0 мураккаб комплексли амалий дастурларга хизмат қилиш учун яратилмаган эди. Янги машина, `Zend Engine` (яратувчилар номлари асосида - Зив ва Анди), бу масалаларни муваффақиятли хал қилди ва 1999 ўртасида пайдо бўлди. Бу машинага асосланган ва кўпчилик Янги имкониятларга эга бўлган PHP 4.0, расмий равишда 2000 йил майида, PHP 3.0 дан икки йил кейин чиқарилди. Анча ошган унумдорликдан ташқари бу версияда PHP 4.0 қуйидаги имкониятларни киритди. Катта сондаги веб-серверлар, http-сессияларни қўллаш, фойдаланувчи киритиши билан хавфсиз ишлаш усуллари ва тилнинг турли янги конструксиялари. PHP 4 хозирда PHPнинг охириги версиясидир. Zend Engine PHP 5.0 га интегратсия қилиш учун яхшилаш ва модификация қилиш устида иш бошланган. Хозирги кунда PHP юз минглаб дастурчилар томонидан фойдаланади ва бир неча миллион сайтлар уни ўрнатилгани ҳақида маълумот берадилар, бу еса интернет доменларнинг 20% дан ортиқдир. PHP яратувчилари командаси ўнлаб дастурчилардан ҳамда, PHP билан боғлиқ PEAR ва хужжатлаш лойиҳалари каби лойиҳалар устида ишловчилардан иборат.

### **1.2.1 PHP да цикл операторлари**

#### **PHPда If ва цикл операторлари**

If else(агар, акс ҳолда) сўров оператори. Барча дастурлаш тилларида бўлгани каби PHP дастурлаш тилида ҳам бу оператор мавжуд. Бирор дастур тузишда бу операторнинг ўрни каттадир. If else орқали текшириш жараёнлари амалга оширилади, яъни агар амал рост бўлса, бирор буйруқни бажар, акс ҳолда бошқасини.

```
if(мантиқий амал)
оператор_1;
else
оператор_2;
```

Мантикий амал рост бўлса, оператор\_1 акс холда оператор\_2 ишлайди.

Мисол\_1

```
If($a>$b)
```

```
Echo "a katta b dan";
```

```
Else
```

```
Echo "a kichik b dan";
```

Агар а ўзгарувчи б ўзгарувчидан катта бўлса, кавс ичидадан true (рост) натижа чиқади ва ундан кейинги оператор ишлайди, else дан кейинги оператор ишламайди. Агар кавс ичидан false(ёлғон) жавоб чиқса(а кичик б бўлса) if дан кейинги оператор ишламайди, аксинча else дан кейинги оператор ишлайди. Мантикий амал ичида гуруҳли солиштириш ҳамбўлиши мумкин, яъни бир пайтнинг ўзида бир неча солиштириш.

агар бир вақтнинг ўзида ҳам ўзгарувчи 100 дан катта ёки тенг бўлиб, 5000 дан кичик ёки тенг бўлса, экранда амал тўғри деган ёзув акс холда(бирорта амал хато бўлса).

Агар бир неча теркшириш бирин кетинлик билан амалга ошириш керак бўлса, ҳар бир қисм блокларга

```
олинади({ });
```

Мисол\_3

```
If($a>$b)
```

```
{ Echo "a katta b"; $c=$b; }
```

```
Else if($a==$b)
```

```
{ Echo "a teng b"; $c=$a; }
```

```
Else
```

```
{ Echo "a kichik b dan"; $c=$a; }
```

```
Echo "Kichik son: $c";
```

**While цикли.**

Бу оператор дастурлашда цикллар билан ишлаш учун мўлжалланган. Шарт терширилади ва цикл ишга тушади, токи шарт false(ёлғон) бўлсагина циклдан чиқилиб кейинги операторлар ишлай бошлайди. Унинг структураси:

```
While(мантикий_амал)
```

```
Операторлар;
```

Мисол\_4

```
$i=1; $p=1;
```

```
While($i Echo $p, " ";
```

```
$p=$p*2;
```

```
$i=$i+1;}
```

While цикл операторининг бошқа шакли

```
While(мантикий амал)
```

```
Операторлар;
```

```
end While;
```

```
do-While.
```

Бу ҳамцикл оператори. Фақат, дастлаб операторлар бажарилади ва кейин шарт текширилди. Бу циклда оператор(операторлар гуруҳи) камида бир маротаба бажарилади.

Унинг структураси:

```
do{Оператор;}
```

```
While(мантикий амал);
```

**For цикл оператори**

Бу цикл оператори универсал ҳисобланади. Бу операторнинг универсаллиги шундаки, барча жараёнлар for операторининг ичида содир бўлади. Унинг структураси.

```
For(ўзгарувчиларни эълон қилиш;цикл шарти;циклдан кейинги жараён)
```

```
Цикл танаси;
```

```
Ёки
```

```
For(---/----/----/)
```

Операторлар;

end if;

Кўриб турганингиздек, цикл ишлашидан олдин барча қилиниши керак ишлар қилиниб олинади, кейин цикл ишга тушади.

Мисол\_6

```
For($i=0, $j=0, $k="ok"; $i  
{ $k = $k."."; Echo $k; }
```

### **Foreach**

PHP4 дан қўшилган. Массивларни чиқариш учун ишлатилади.

Структураси:

```
Foreach(массив ас $калит=>$қиймат)
```

Операторлар;

Мисол\_7

(Барча глабал массивларни экранга чиқариш)

```
Foreach($_SERVER as $k = $v)  
{ Echo "$k=>$v"; }
```

Калитга массивнинг ҳарбир қийматини бериб, уни экранга чиқармоқда.

Агар калит керак бўлмаса, қуйидагича кўринишга эга бўлади.

```
Foreach(массив ас $қиймат)
```

Операторлар;

Switch-case

Switch орқали бир ўзгарувчи текширилади ва сасе орқали ўзгарувчи қийматини топишга ҳаракат қилинади. Структураси қуйидагича:

```
Свитч(бирор бир ўзгарувчи){  
Сасе қиймат1:команда1; break;  
Сасе қиймат2:команда2; break;  
Сасе қиймат3:команда3; break;  
Сасе қиймат4:команда4; break;  
.....  
Сасе қийматН:командаН; break;
```

```
default
```

```
Командалар break;
```

```
}
```

```
Мисол_8
```

```
Switch($a){
```

```
Case 0; echo "a тенг 0" break;
```

```
Case 1; echo "a тенг 1" break;
```

```
Case 2; echo "a тенг 2" break;
```

```
Case 3; echo "a тенг 3" break;
```

```
Default Echo "a 0,1,2,3 га тенг эмас";
```

```
}
```

а ўзгарувчи текширилмоқда, агар  $a=0$  бўлса, а тенг 0 деган ёзув чиқади, ва ҳаказо агар а ўзгарувчи ҳеч бирига тенг бўлмаса а 0,1,2,3 га тенг эмас деган ёзув чиқади.

### 1.2.2. PHP да функциялар

Функциялар - бу дастурлашнинг асоси ҳисобланади. Дастурлашнинг барча тилларида мавжуддир. Функциялар орқали кодларни тартиблаш, дастурнинг ишлаш тезлигини ошириш ва кодларни ихчамлаштириш каби ютуқларга эришиш мумкин.

Функциянинг тузилиши:

```
Функцион функция_номи(параметр1, параметр2,...)
```

```
{
```

```
оператор_1;
```

```
оператор_2;
```

```
.....
```

```
}
```

" Функцияни чақириш:

```
Функция_номи(параметр_1, параметр_2);
```

Мисол\_1

```
function printMenu($name)
```

```
{ print("$name"); }
```

```
//функцияни чақирамиз:
```

```
$names = "Hello users";
```

```
printMenu($names);
```

Натижа:

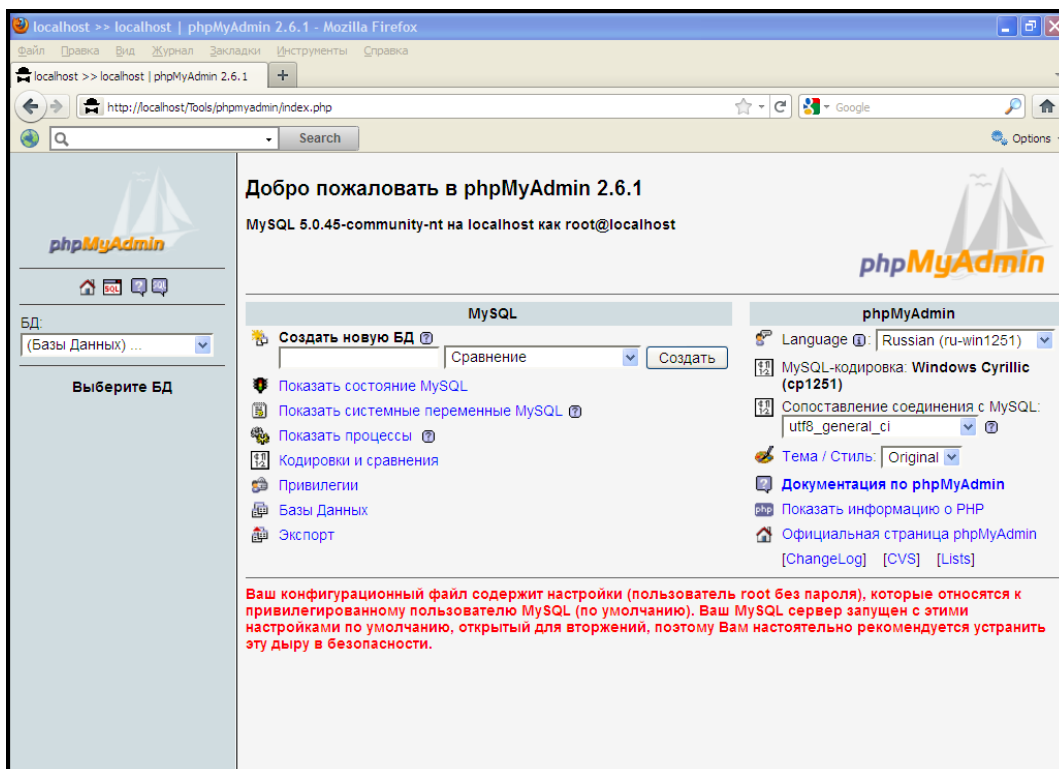
Hello users

### 1.2.3 PHP да MySQL базаси билан ишлаш

#### MySQL ҳақида бошланғич тушунчалар

MySQL - бу энг машхурва жуда кўп фойдаланиладиган маълумотлар базасини бошқариш тизими (МББТ) ҳисобланади. Бу тизим жуда катта маълумотлар билан ишлаш учун яратилмаган, аксинча бироз кичик ҳажмдаги базалар билан катта тезликда ишлаш учун яратилган. Унинг асосий ишлаш доираси, сайтлар ҳисобланади. Ҳозиргикунда жуда кўп сайт ва блоklarнинг маълумотлари айнан шу МББТ сақланади. Хўш, сайтларнинг нималари базада сақланиши мумкин? Дастлаб, маълумотлар сақлаш учун дастурчилар файллардан фойдаланишган, яъни файл очилиб керакли маълумотлар у ерга сақланиб, керакли пайтда чақириб ишлатилган. Кейинчалик файллардан воз кечилиб (ноқулайликлар юзага келган, яъни файлни очиш, ўқиш, ёпиш,.. кўп вақтни олиб қўйган, у ердан қидириш, хуллас жуда кўп), маълумотлар базасига ўтилган. Базада сайтдаги мақолалар, сайт фойдаланувчилари ҳақидаги маълумотлар, сайт контентлари, қолдирилган комментарийлар, савол-жавоблар, ҳисоблагич натижалари ва шунга ўхшаш жуда кўп маълумотлар сақланади. MySQL шундай маълумотларни ўзида сақлайди. MySQL - жуда катта тезликда ишловчи ва қулай ҳисобланади. Бу тизимда ишлаш жуда содда ва уни ўрганиш қийинчилик туғдирмайди. MySQL тизими тсх компанияси томонидан, маълумотларни тез қайта ишлаш учун корхона

миқёсида ишлатишга яратилган. Кейинчалик оммалашиб, сайтларнинг асосий базаси сифатида ёйилди. Denwegнинг ичида куйидагича кўринишда бўлади.(1-расм)



1-расм(“localhost”да база яратилиши)

Сўровлар SQL тили орқали амалга оширилади. Бу МББТ релясион маълумотлар база ҳисобланади. Бу дегани база жадваллар, жадваллар эса устунлардан ташкил топгандир. MySQL МББТ 2 хил турдаги лицензияга эга. Биринчиси бепул, яъни MySQLни кўчириб олиш ва ишлатиш ҳеч қандай ҳаражат талаб қилмайди ва GPL(GNU Public License, GNU) лицензиясига асосланади. Иккинчи тури, GPL шартига кўра, агар сиз MySQL кодларини бирор дастурингизда ишлатсангиз, бу дастурингиз ҳам GPL(бепул) бўлиши керак. Бу эса дастурчига тўғри келмайди. Шунинг учун, бу дастурингизни пуллик қилишингиз учун MySQL пуллик лицензиясини сотиб олишингиз керак. MySQL логотипи делфин ҳисобланади. Бу делфинни исми "Сакила"дир. Бу логотипни OpenSource тузувчиларидан бири Ambrose Twebaze га тегишлидир. MySQL жуда кўп операцион тизимлар билан ишлай олади.

Буларни ёзадиган бўлсам: AIX, BSDi, FreeBSD, HP-UX, Linux, Mac OS X, NetBSD, OpenBSD, OS/2 Warp, SGI IRIX, Solaris, SunOS, UnixWare, Windows 95, Windows 98, Windows NT, Windows 2000, Windows XP, Windows Server 2000, Windows Vista, Windows 7,... MySQL швед корхонаси MySQL AB га тегишли бўлган, кейинчалик SUN фирмаси MySQL ни ўзиники қилиб олган. Бир неча йил олдин эса, гигант корхоналардан бири Oracle, SUN фирмасини сотиб олди ва ҳозирда MySQL Oracle фирмаси маҳсулоти сифатида чиқиб келмоқда. Бу бир ҳисобда Microsoft SQL Сервер га катта рақобатдош эканлигини англатади. MySQL бир неча серверларнинг бир қисми ҳисобланади. Мисол учун, wamp, appserv, lamp, denwer клиентлар MySQL серверига маълум бир кутубхоналар орқали уланади. MySQL га қуйидаги дастурлаш тиллари уланиб ишлаши мумкин: Delphi, C, C++, Java, Perl, Php, Python, Ruby ва бошқалар.

### **MySQL типлари ва версия тушунчаси**

MySQL бу - кроссплатформали дастур ҳисобланади, яъни бир хил типга эга бўлган(ишлаш технологияси бир хил), лекин ўрнатилувчи файллари бироз бошқачадир. ҳозирда MySQL Oracle фирмаси томонидан ишлаб чиқарилмоқда, олдин SUN фирмасига тегишли бўлган. Шунинг учун MySQL 4 версияси MySQL 5 версиясидан фарқ қилади. ҳозирда энг сўнги версияси 5.6.16 дир. Хар бир дастур, маълум бир сонлар асосида версияларга ажратилади. Дастур исми ўзгармайди, фақат сонлар ўзгариб, дастурнинг янги ёки эскилигини ажратиб туради. Янги ёки эскилиги, дастурнинг даражасини белгилайди, яъни янги дастур эскисига қараганда янги қирралари мавжудлигини англатади. MySQL дастури ҳам ҳамма дастурлар каби ўз версиясига эга. Демак биринчи рақам асосий рақам дейилади. Бизнинг ҳолатда бу 5 рақами. Бу рақам камдан кам ўзгаради, фақатгина дастурнинг асоси ўзгарса ўзгаради. Кейинги рақам 6, бу рақам асосий рақамга ёрдамчи рақам ҳисобланади ва асосий рақамга қараганда тезроқ ўзгаради. Бунинг ўзгариши, дастурнинг ишлаш технологияси бироз ўзгариши натижасида

содир бўлади. Кейинги рақам буилд дейилади. Бизнинг ҳолатда бу 16 рақам. Дастур ишлашида бирор нуқсон пайдо бўлса, дастурчилар дарров шу нуқсонни йўқотишади ва рақамни кейингисига ўзгартириб қўйишади. Бундан кўриниб турибдики бу рақам тез ўзгаради. Бу ўзгартириш дастурчилар тилида Path дейилади.

Ўз тажрибамдан келиб чиқиб фикр билдирсам, бари бир янги чикган Patch, эски нуқсонларни бартараф этиб, янгисини пайдо қилади .

MySQL асосан 2 турга бўлинади. Биринчиси, пуллик версия бўлиб, MySQL Enterprise Server дейилади. Бу туридан фойдаланганда, ҳар йил маълум бир маблағ тўлаш талаб этилади. Иккинчиси, MySQL Community Server деб номланиб, бепул ҳисобланади. MySQL Enterprise Server MySQL Community Server дан фарқи, Enterprise типи ўзида барча утилитларни камраб олади ва чиқаётган барча янги Path ва ўзгаришлар дастлаб шу типига чиқарилади ва анча вақтдан сўнг Community типига чиқади. Ундан ташқари Enterprise типига кластер технологияси ҳам мавжуд. Биз учун Community типи ҳам етарли ҳисобланади ва шу типини ўрганиб борамиз. MySQL Community типининг ҳам 2 тури мавжуд бўлиб, бири Essentials(бошланғич) версияси, иккинчиси Installer дейилади. Essentials турида фақат сервер мавжуд бўлиб, ҳеч қандай қўшимча компонентлар йўқ, Installerда эса бир неча қўшимча компонентлар мавжуд бўлиб, ҳажми ҳам каттароқ. Essentials версияси зип архивда берилади, Installer эса .msi файли кўринишида бўлади. Ундан ташқари MySQL ни кўчириб олишда, операцион тизимнинг қайси битлигини ҳам инобатга олиш лозим(операцион тизим битлари ҳақида). Сайтда ҳам, 64 ва 32 учун алоҳида MySQL дастурини файллари таклиф қилинади. MySQL ишлаши учун, қхч қандай Apache сервери, ҳеч қандай PHP керак эмас. Фақатгина базани ўзини ўрганмоқчи бўлсангиз, Apache PHP ларни ўрнатманг. Булар фақат тўлиқ сайт яратиш учун керак. Тўғри буларни ўрнатиш ҳам қийин эмас, лекин ортиқча нарсалар. Базани мукамал ўрганганингиздан сўнг қолган нарсаларни ўрнатиб, созловчи файллар орқали бир бирларини боғлаб оласиз. Яъна бир нарса, кўпчилик сайтларда олдин

Apache қўйинг, кейин MySQL ва охирида PHP ни ўрнатинг деган мақолаларни учратиш мумкин. Умуман олганда кетма-кетликнинг умуман фарқи йўқ, биргаликда ишлашса ҳам, булар ҳаммаси алоҳида - алоҳида дастурлар. Агар ҳали ҳам MySQL нималигини англай олмаган бўлсангиз, охириги тушунтириш гапларини ёзаман, бу - маълумотлар базасини бошқариш тизими бўлиб, бу тизимга синоним тизим қилиб, Oracle, MS SQL SERVER ларни мисол қилиш мумкин.

#### 1.2.4 Ўзгарувчилар

Ўзгарувчи – бу шундай жойки, бу жойни ташкил этиш ва у ерда дастурда керак бўладиган маълумотларни сақлаш мумкин. Барча ўзгарувчилар \$ белгиси билан бошланувчи алоҳида номга эга бўлади. Ўзгарувчи номида ҳарф ва сонлардан ташқари бошқа белгиларни қўйиш, шунингдек бўшлиқ (пробел) қўйиш мумкин эмас. Қуйида ўзгарувчини ташкил этишнинг тўғри келтирилган ҳолатини кўрайлик:

```
$a;  
$a_long-ish_variable_name;  
$2453;  
$sleppyZZZZ;
```

Агар аҳамият берсак, ўзгарувчи охирида нуқтали вергул (;) белгиси қўйиляпти, лекин бу белги ўзгарувчи номига тегишли бўлмайди. Балки у ўзаро ажратувчи символ ҳисобланади.

Ўзгарувчини шундай қурол сифатида тасаввур қилиш мумкинки, у бир хил типдаги белгиларни ўзида сақловчи махсус яшиқ дейиш мумкин. Бу ўзгарувчида сонлар, символлар, объектлар, массивлар ёки мантиқий қийматлар сақланиши мумкин. Ўзгарувчидаги берилган қийматларни ихтиёрий вақтда ўзгартириш мумкин.

Динамик ўзгарувчилар.

Бизга маълумки, ўзгарувчи \$ белги асосида эълон қилинади ва ундан сўнг ўзгарувчи номи келтирилади. Лекин биз учун ўзгарувчини бошқа ўзгарувчида сақлаш албатта янгилик. Лекин бунинг иложи бор. Масалан:

```
$user="bob";
```

сифатида келтирилган ўзгарувчи қуйидаги билан эквивалент ҳисобланади:

```
$holder="user";
```

```
$$holder="bob";
```

Бу ерда \$holder га user ўзлаштирилгани учун \$ кейинги \$holder ни user деб тушуниши мумкин. Динамик ўзгарувчини қаторли ўзгармас ёрдамида эълон қилиш мумкин. Бунинг учун ўзгарувчи номини {} ичида берилиши лозим :  `${"user"}="bob" ;`

Динамик ўзгарувчига мурожат қилиш учун худди шундай кетма кетлик бажарилади:

```
$ user="bob" ;
```

```
print=$user ;
```

Бу қуйидаги билан эквивалент ҳисобланади ;

```
$ user="bob" ;
```

```
$holder="user";
```

```
print $ holder;
```

Лекин ўзгарувчи номини чоп қилиш учун бошқачароқ берилади:

```
$user="bob" ;
```

```
$holder="user";
```

```
print "$$ holder";
```

Экранда эса \$user чоп қилиб берилади.

Юқорида келтирилиб ўтган фикримизга асос ва тушунарли бўлиши учун қуйидаги мисолни келтириб ўтамиз:

```
<html>
```

```
<head>
```

```
<title> Динамик ўзгарувчини яратиш </title>
```

```

</head>
<body>
<?php
$holder="user";
$$holder="bob" ;
print"$user<br>";           // натижа "bob" чикади
print"$holder";           // натижа "bob" чикади
print"<br>";
print"${ $holder}<br>";     // натижа "bob" чикади
print"${ 'user' }<br>";    // натижа "bob" чикади
?>
</body>
</html>

```

#### Маълумотлар ва уларнинг типлари

Турли типдаги маълумотлар хоотирадан тирли хил жой эгаллайди ва уларни қайта ишлаш ҳам хар хил бўлади. Шунинг учун дастурлаш тилларининг кўпида ўзгарувчини эълон қилиш олдиндан бажарилиши талаб қилинади. PHP тили эса, бундай қаттиқ талаблардан йироқ бўлиб бу типларни алоҳида эълон қилишни талаб қилмайди. Берилган қийматга қараб типни ўзи ажратиб олади. Бу усул осон лекин камчиликка эга ҳисобланади. Чунки ўзгарувчилар кўпайиб кетганда уларни назорат қилиш дастурчига қийинчиликлар келтиради.

**Қуйидаги жадвалда PHP тилида фойдаланиладиган асосий типларни келтириб ўтайлик:**

Тип	Мисол	Берилиши
Integer	5	Бутун сон

Double	3.234	Ҳақиқий сон
String	“hello”	Символлар
Boolean	true	Мантиқий
Object		Объектлар

PHP4 тилида шундай функция борки, у ўзгарувчи типини аниқлаб бера олади. Агар `gettype()` функцияси ичига ўзгарувчи номини берилса, у натижа сифатида типни беради. Қуйидаги мисолда `gettype()` асосида турли хил қийматларни олувчи битта ўзгарувчи типини аниқлаш дастурини кўриб ўтайлик:

```

<html>
<head>
<title> Ўзгарувчи типин аниқлаш </title>
</head>
<body>
<?php
    $testing=5;
    print gettype($testing); // 1 integer;
    print <BR>;
    $testing="five";
    print gettype($testing); // string;
    print <br>;
    $testing=5.12;
    print gettype($testing); // double;
    print <br>;
    $testing=true;
    print gettype($testing); // boolean;
    print <br>;
?>

```

```
</body>
```

```
</html>
```

Бу дастур куйидаги натижаларни чоп қилиб беради:

```
Integer;
```

```
String;
```

```
Double
```

```
Boolean.
```

### Ўзгарувчи типини ўзгартириш

Ўзгарувчини типини ўзгартириш учун PHP4 да Set Type, функциясидан фойдаланилади. Бунинг учун функцияда ўзгарувчининг типини ва унинг янги типини кўрсатилиши лозим . SetType функциясидан дастурдан фойдаланишни куйидаги мисол ёрдамида кўрайлик :

```
<html>
```

```
<head>
```

```
    <title> Ўзгарувчи типини ўзгартириш </title>
```

```
</head>
```

```
<body>
```

```
<?php
```

```
    $ss=3.14 ;
```

```
    print gettype ($ss) ; // double
```

```
    print “- - $ss <br>” ; // 3.14
```

```
    SetType ($ss, string);
```

```
    print gettype ($ss) ; // string
```

```
    print “- - $ss <br> “ ; // 3.14
```

```
    SetType ($ss, Integer);
```

```
print gettype ($ss) ; // integer
```

```

print "-- $ss <br> "; // 3
SETTYPE($SS, double);
print gettype ($ss) ; // double
print "--$ss <br>"; // 3.6
?>
</body>
</html>

```

### Операторлар ва ифодалар

Оператор - бу, бир ёки бир неча қийматлардан янги қийматларни ҳосил қилишда ишлатиладиган символ ёки символлар тўплами. Операнда – бу, оператор ёрдамида қўшилувчи қиймат ёки ўзгарувчи. Масалан: иккита операнда ва битта оператор ёрдамида янги сон ҳосил қилишни кўрайли:

Бу ерда 4 ва 5 лар операнда «Қ» оператор ҳисобланади. Ифода бу, операторлар ёрдамида бажарилувчи ихтиёрий ссон, ўзгарувчи ва функцияларнинг комбинацияси ҳисобланади. РНР 4 тилида ўзлаштириш операторларини «=» белигиси билан белгиланади. Масалан, `print($name="bob");`

РНР4 да ишлатилувчи асосий арифметик операторни қуйидаги жадвал кўринишида келтириб ўтайлик:

Оператор	Номланиши	Мисол	Натижа
Қ	қўшиш	10Қ3	13
-	айриш	10-3	7
/	бўлиш	10/3	3,33333
*	кўпайтириш	10*3	30
%	модул бўйича бўлиш	10%3	1

PHPда конкатенация оператори оддий нуқта ҳисобланади. Бу оператор ўнгидаги қаторни чапга қўшиб бериш (боғлаш) да ишлатилади.

Масалан, “hello”.”world”; натижа эса қуйидагича бўлади: “helloworld”

Ўзлаштиришнинг қўшимча операторлари. Аслида ўзлаштириш оператори битта, лекин бир нечта арифметик ва ўзлаштириш операторлари комбинациясидан ташкил топган операторлар ҳам мавжуд. Бу операторларни қуйидаги жадвалда келтириб ўтайлик:

Оператор	Мисол	Эквиваленти
+ =	$\$x+=5$	$\$x=\$x +5$
- =	$\$x-=5$	$\$x=\$x -5$
/ =	$\$x/=5$	$\$x=\$x /5$
* =	$\$x*=5$	$\$x=\$x *5$
% =	$\$x\%=5$	$\$x=\$x \%5$
. =	$\$x.=\text{”test”}$	$\$x=\$x.\text{”test”}$

Солиштириш операторлари.

Солиштириш операторлари иккита операндани солиштириш учун ишлатилади. Улар иккита операндани солиштириш натижаси тўғри бўлса “true” акс холда “false” натижасини беради. Қуйида солиштириш операторлари келтирилган жадвални келтириб ўтайлик:

Оператор	Номланиши	Бажарилиш шарти	Миқдор	Натижа
=	Тенглик Тенг	Чап ва ўнг операндлар тенг Чап ўнга тенг эмас	$\$x=$ $=5$	Fals e

!	эмас		\$x!	Tru
=		Тенг ва бир хил	=5	e
	Бир хил			
=			\$x=	Fals
= =			= =5	e

## Ўзгармаслар

Ўзгарувчилардан фойдланиш маълумотларни сақлаш ва қайта ишлашнинг энг ажойиб усулларида бири хисобланади. Биз бу ўзгарувчилар қийматини, хаттоки типини хоҳлаган вақтимизда ўзгартира олишимиз мумкин. Лекин дастур яратиш жараёнида шундай ҳолат бўлишини истаб қоламизки, бунда бизни берган кийтимиз ёки маълумотимиз дастурни яратиш ва фойдаланиш жараёнида, керак бўлса тасодифан ўзгариб кетишдан ҳоли булсин, яъни ўзгармасин. Бунинг учун ўзгармаслардан фойдаланишимиз зарур. PHP4 да ўзгармаслар билан ишловчи `define()` функцияси мана шу мақсадда ишлатилади. Бу функция ёрдамида яратилган ўзгармас (константа) дастурни яратиш ва фойдаланиш жараёнида ўзгартирилмайди. Ўзгармасни ташкил этиш учун `define()` функциясида ўзгармас номи ва унинг қиймати берилади.

`Define ("nomi",42)`

Ўзгармас қабул қиладиган маълумот қиймат ва символ бўлиши мумкин. Ўзгармас номи эса фақат катта ҳарфлар билан берилади. Ўзгармасдан фойдаланишда фақат катта ҳарфлар билан \$ белгисини олдиға куймасдан ёзилади. Қуйидаги ўзгармасдан фойдаланиш дастурда қандай ишлатилишига мисол сифатида келтириб ўтамыз:

`<html>`

`<head>`

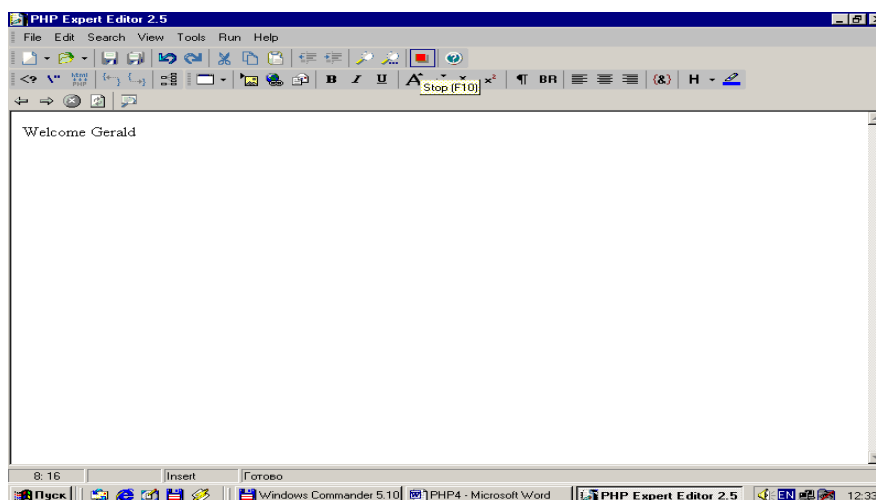
`<title>Uzgarmaslarni tashkil etish</title>`

`</head>`

`<body>`

```
<?php  
define("USER",Gerald);  
print "Welcome".USER;  
?>  
</body>  
</html>
```

Агар дастурга аҳамият берсак, «Welcome» ёзувидан сўнг конкатенация (.) белгисидан фойдаландик. Дастур бажарилиши натижасида эса, экранда куйидаги кўриниш ҳосил бўлади(2-расм)



2-расм(PHPда “welcome” сўзи ёзилмоқда)

PHPда яна бир имконият борки, у фойдаланувчига автоматик равишда бир нечта ўзгармас ташкил этиб беради. Масалан, `_FILE_` ўзгармаси фақат фойдаланилаётган актив файл номини узида сақлайди. `_LINE_` ишлатилаётган файлдаги актив қаторни аниқлаб беради ва х.к

## II. ЛОЙИҲА ҚИСМИ

### 2.1 CMSлар турлари

Барчага маълумки, бутунжаҳон тармоғидаги веб-саҳифалар кундан кунга кўпайиб бормоқда. Бу ҳолат Ўзбекистонда, яъни узнетда ҳам жадаллик билан амалга ошмоқда. Кимдир ўз шахсий саҳифасини яратса, компания ва фирмалар корпоратив сайтларини яратмоқда. Яна кимдир ўз кундаликларини, яъни блогларини олиб борадилар.

Шу билан бирга, йирик порталлар фаолияти ҳам ривожланмоқда. Буларнинг ҳаммаси эса бирор дастурий восита ёрдамида яратилади. Кимдир HTML тилида, кимдир PHP да, яна кимлардир CMS деб аталувчи тизимлар ёрдамида веб-сайтларни яратмоқда. Хўш, CMS дегани ўзи нима? Бу тизимни нима эканини тушуниш учун, келинг, аввало, веб-саҳифалар қандай яратилиши ҳақида билиб оламиз.

1989-йилда Тим Бернерс-Ли томонидан World Wide Web, яъни Бунунжаҳон ўргимчак тўри ихтиро қилинганидан бошлаб унинг ҳар бир фойдаланувчиси оддий бўлса-да, ўз веб-саҳифасини ярата олиш имкониятига эга бўлди. Лекин, бунинг учун HTML (Hyper Text Markup Language) тилини билиш шарт эди. Турли хил АТ-компаниялари томонидан бу ишни осонлаштирувчи бир қанча дастурий пакетлар, жумладан Adobe Dreamweaver ва Microsoft FrontPage кабилар яратилди. Бундай дастурлардан баъзилари сайт яратишнинг барча босқичларини (HTML-кодни ёзишдан бу кодни серверга юклашгача) бошқаришни ўз зиммасига олади. Шу билан бирга, кўпгина ҳолларда (айнан оддий саҳифаларни яратишда) фойдаланувчи бу кодни билиши шарт бўлмаган. Сабаби бу дастурларда веб-саҳифаларни яратишнинг визуал усулларидан фойдаланилган.

Бироқ, HTML-редакторларининг барча афзалликларига қарамай, сайтни бошқариш вақт ўтгани сайин ва янги маълумотлар қўшилиши билан қийинлашиб бораверди. Чунки ахборот технологиялари ривожланиши билан веб-сайтлар ҳам ривожланиб борди. Сайт харитасини яратишдек оддий функцияни қўшиш учун ҳам веб-яратувчилар томонидан анча-мунча меҳнат

талаб этилар эди. Ахир бунинг учун барча иловаларни кўриб чиқиш, саҳифалар орасидаги боғлиқликларни текшириш, менюни яратиш, форумни кўшиш каби ишларни бажариш лозим эди. Бу эса доимий диққатни ва жуда кўп вақтни талаб этарди.

**CMS “ Content Management System ”** деб ёйиб ёзилади, яъни контентни бошқариш тизими ёки сайтни бошқариш тизими. Бу сайтни тез яратиш Bitrix (Сайтни Бошқариш Тизими) ишини соддалаштириш учун хизмат килади.

**Биринчи:** Сайтни бошқариш тизими буюртмачига сайтни мустақил эксплуатация қилишда, яъни бизнинг мутахассис билан эмас, балки ўзингизнинг компания хизматчингиз (менеджер, котиб, маркетинг) билан сайтдаги ихтиёрий маълумотни ва унинг структурасини ўзгартиришингизга имкон беради. Агар сиз сайтингиз маркетингли ускуна ва ундаги ҳар кунги маълумотида компаниянинг охири маркетингли акциялари акс этишини хоҳласангиз у ҳолда CMS ни сизнинг сайтингизга қўлланилиши шарт. Бу сайтингизни ривожлантириш ёки модернизациялаш ишларида ўйланмай қилинган ҳаракатлар ёки шунчаки хатолардан асрайди.

**Иккинчи:** Сайтни бошқариш тизими ёрдамида сайтни ривожланиши сезиларли соддалашади ва жонланади. Сайтда янги бўлим яратиш, ихтиёрий саҳифада реклама блокларини, ички баннерларни ёки товарларнинг махсус таклифларини жойлаштиришни буюртмачи мустақил ва дақиқалар билан ўлчанадиган вақт мобайнида бажариши мумкин. Бундан ташқари, Буюртмачи мустақил бизнинг ёрдимимизсиз, CMS ёрдамида сайтни янги интерфаол бўлимларини яратиши ( масалан, forum) ёки у ёки бу саҳифаларни парол билан ёпиб ва уларга фақат танланган фойдаланувчиларгагина киришга имкон бериш мумкин.

**Учинчи:** Бизнинг сайтни бошқариш тизимимиз (CMS, Битрих) сизнинг сайтингиз асосида ётувчи бўлган, бошқа проектарда кўп мартаба ўрнатилгани (3500 мартадан кўпроқ), учун у Сайтингизни уни эксплуатация қилиш жараёнида дастурий хатолардан асрашига кафолат беради.

**CMS**–бу Content Management System дан олинган кенгайтма, у сўзма-сўз таржима қилинганда “Сайт контентини Бошқариш тизими” ёки “Сайтнинг Бошқариш Тизими” ларни аниқлатади. Баъзида CMS ни сайтнинг “Ҳаракатлантирувчиси” деб аташади.

**CMS**–бу динамик маълумотли Веб-саҳифаларни ишлаб чиқариш ва таъминлашга имкон берувчи дастурий таъминотдир. Ҳар хил CMS ҳар хил мураккабликдаги сайтларни проектлашга имкон беради. Интернет магазинлардан то инфорацион порталларгача, ҳаммасидан ҳам, CMS ахборотли ва контентли сайтларни яратиш учун мос келади.

### **CMS нинг иккита асосий тури бор:**

1. Интернетда ишловчи ва жойлашувчи CMS.
2. Компютерга ўрнатилган ва сайт билан автоматик ёки файлларни янгилаш интерфейси орқали уланган CMS.

CMS нинг сайт билан интеграциясидан кейин компаниянинг хизматчиси сайт таркиби ва структурасини бошқаришлари, янгиликларни чоп қилишлари, матнларни таҳрир қилишлари ва графикани жойлаштиришлари , каталогдаги маҳсулотни узгартиришлари ва маҳсулотларни қўшишлари янги саҳифаларни, бўлимларни яратишлари ва кўпгина бошқа ишларни бажаришлари мумкин. CMS асосида қурилган динамик сайтларнинг устунлиги дизайннинг ахборотли тўлдиришдан ажратилганида бу ҳужжат айланмасини, бизнес жараёнларни шахсийлаштириш механизмларини автоматлаштиришга имкон беради. Бундан ташқари, CMS сайт таркибини осон, содда ва тез бошқаришга имкон беради. Бу CMS дан фойдаланишнинг иқтисодий фойдаси сезиларли даражада бўлишини таъминлайди.

**CMS нинг мақсад ва вазифалари** – мижоз томонидан маълумот ва графикаларни бошқариш, тўлдириш ва таҳрир қилиш. Соддароқ қилиб

айтганда, бу тизим – сизга бўлимларни ўзгартириш ва тўлдириш, сайт таркибини таҳрир қилиш, сайт графика ва дизайнни ўзгартиришга имкон беради. CMS сиз бундай сайтни бошқариш масалалари қиммат турувчи дастурчилар аралашиммаса ҳал қилинмайди.

### **CMS сайт контентни бошқариш тизими**

Веб – сайтларни яратиш ва таъминлаш нархини пасайтиради, шунингдек материалларни янгилаш тезлигини оширади, яъни сайт хўжайнида материал пайдо бўлганидан то сайтда чоп этилганигача кетган вақт CMS ни шарофати билан минимал бўлади. Бунинг устига CMS да маълумотларни чоп этишга тайёрлаш жараёни жуда ҳам оддий. CMS контентни бошқариш тизимининг шубҳасиз ижобий томони шундаки, сайтни бошқариш ва таъминлаш нархи пасаяди. Бу веб – мастер ойлик маошига ҳужжатларни излашга иккиланиш ва ҳатоликларни олдини олишга кетган ёқотишларнинг пасайиши, шериклар ва миждозлар билан боғланиш тезлигининг ортиши ... ҳисобига юз беради. Сизга лойиҳани таъминлаш ва ривожлантириш учун мутахассис ёллашингизга тўғри келмайди. CMS ёрдамида сиз сайтга жойлаштирилган маълумотларнинг ҳажми сайт нархига таъсир қилмаслигига гувоҳ бўласиз. CMS контентни бошқариш тизими сизга сайтда ихтиёрий ҳажмдаги маълумотларни жойлаштиришга имкон беради. Одатда CMS га асосланган сайт саҳифалари бутунлигича сақланмайди, балки уларга мурожат қилишда жуда тез шаклланади, бу саҳифаларни юклашдаги тезликни оширишга имкон беради, чунки саҳифа шаблони (дизайнерлик расмийлаштириш) фақат бир марта юкланади ва кейинчалик ҳар бир янги саҳифа сўровида фойдаланувчи компютеридан очилаверади. Сайтни бошқариш тизимида одатда кешлаштириш механизми амалга оширилади, бу қурилмаларга тушадиган юкламаларни камайтиришга имкон беради ва ишлашнинг ва сайтнинг киритиш қобилиятининг тезлиги ошишига имкон беради. Бу ишлар маълумотлар базаси ва статис материалларга сўров натижаларини сақлаш эвазига содир бўлади. Одатда CMS тизимини сайтнинг компания ишлаб

чиқарувчиси ўрнатишини таклиф қилишади. Бу анча тўғри вариант чунки тизим аввал бошиданок сизнинг сайтингизга жойлашиб, мослашиб кетади. Шунингдек, CMS тизимлари қўшимча функцияларни таклиф этади ва охир оқибат сайтда форум, меҳмон китоби, саволлар ва жавоблар ва ҳоказоларни жойлаштиришга имкон беради. CMS нинг бепуллари ҳам мавжуд, улар сайт таркибини бошқариш учун етарлича катта имкониятларни намоиш қилади. Бепул CMS лар кўп фойдаланувчиларга эга, улар бепул контентни бошқариш тизими ёрдамида сайтни бошқариш муаммоларини ҳал қилишга ёрдам беради. Бепул CMS ларга катта миқдордаги кенгайтма ва қўшимчалар мавжуд, улар сайтни бошқариш жараёнини анча қулай ва функционал қилишга имкон беради. Аммо бепул CMS ларнинг салбий томони расмий техник қўллаб қувватлашнинг йўқлиги ва сайт таркибини бошқариш тизимида критик хатоларни тўғрилаш, тезлигининг паслигидадир.

Қайси CMS дан фойдаланган маъқул?

Ҳозирги даврга келиб, CMS асосига қурилган бир қанча тизимлар яратилган. Уларга мисол қилиб, Drupal, PHP-Nuke, WebDirector, Joomla!, NetCat, Slaed, Microsoft CMS, WordPress, PHPShop кабиларни келтириш мумкин. Бу тизимларнинг баъзилари умуман Бепул бўлса, баъзиларидан эса ҳаққини тўлабгина фойдаланиш мумкин.

CMS ни танлаш бир томондан осон ва жўндек кўринсада, бу жуда жиддий масала. Интернетда уларнинг турли йўналишдаги ўнлаб (агар юзлаб бўлмаса) юз долларлаб турадиганидан то очик кодли бепул туригача, маълум бир соҳагагина тааллуқлисидан ихтиёрий мураккабликдаги сайт қуриш имконини берадиганигача, осон бошқариладиганидан администратор учун катта қийинчиликлар келтирадиганигача турини топишингиз мумкин. Хуллас, агар CMS лар билан ишлаш бўйича ҳеч қандай тажрибангиз бўлмаса, кераклисини танлаш қийинлашиб кетиши тайин.

Дастлаб келинг, пуллик CMS ларни ишлатмасликка ҳаракат қиламиз. Ахир катта пул тўлаб, CMS ни ишлатишдан бесамара фойдаланишга нахожат. Шу ерда бир нарсани айтиб ўтиш керак: бу билан ўзимизга қийин

киламиз. Бошқа тарафдан эса, қанчалик мураккаб бўлса, шунчалик қизик бўлиши аниқ, шундай эмасми ?

Энди CMS ларга яна бир талаб қўямиз: универсаллик. Бу дегани, биз ишлатмоқчи бўлган CMS ихтиёрий конфигурациядаги компьютерларда ҳам ишлай олишидир. Шунингдек, сайтни локал сервердан ҳақиқий серверга кўчирганимизда ёки бевосита сервернинг ўзида яратаётганимизда CMS билан ҳеч қандай қийинчиликлар туғилмаслиги керак. Бу талабга эса, афсуски, кўпчилик бепул CMS лар жавоб бера олмайди.

Барча дастурий таъминотларга қўйиладиган талаблардан яна бири – хавфсизлик (ахир сиз ҳафталаб-ойлаб яратган сайтингизни шунчаки қизиқиш учун бир кунда бузишларини хоҳламасангиз керак?). Шунинг учун CMS ларнинг хавфсизлик даражалари доимий янгиланиб туриши шарт.

Натижада, талабларимизга деярли тўлиқ жавоб берадиган қуйидаги: PHP Nuke, Drupal, Joomla!, Slaed CMS тизимларигина қолади. Энди ана шу CMS ларни батафсилроқ кўриб чиқамиз ва ўзимизга керакли баъзи хулосалар чиқарамиз.

**PHP Nuke.** Энг биринчи CMS лардан бири бўлиб, афсуски, анча вақтлардан буён янгиланмай келмоқда. Бироқ, унинг коди кўплаб адашлари учун асос бўлиб хизмат қилмоқда (масалан, DotNetNuke, PostNuke). Бу бири-бирига ном жиҳатдан яқин бўлган тизимлар фойдаланувчи диққатига навбатма-навбат чиқиб туради. Функционаллик жиҳатидан тизим анча мукамал бўлиб, унинг учун кўплаб қўшимча модуллар яратилган. Масалан, хабарлар лентаси, форум, мақолалар, сўровлар ташкил қилиш, статистика олиб бориш ва бошқалар. Аммо, сайтларнинг тузилиши жиҳатидан бир хиллик кузатилади. Шу билан бирга, бу тизим юкланиши қийин бўлади, яъни қўшимча модуллар саҳифа юкланишини секинлаштиради. Бу эса кўплаб фойдаланувчиларга ёқмаслиги тайин. Юкланиш нафақат саҳифага, шунингдек, серверга ҳам тушади, бу эса хостинг хизматини тақдим этувчиларга ёқмайди. Агар сизни шундай вазият қониқтирса, айнан шу CMS ни танлашингиз мумкин.

**Drupal.** Турли хил ўзгартиришлар орқали ўзимизга мослаб олиш бўйича энг мукамал CMS деб айтиш мумкин. Шу ерда шуни айтиш керакки, “Томчи” (бу тизим номи инглиз тилидан айнан шундай таржима қилинади) аслида, CMS эмас, балки “CMS ни тузувчи тизим”. Унинг ёрдамида сиз тайёр блоклардан керакли CMS ни йиғасиз. Шуни айтиш мумкинки, функционаллик ва дизайн жиҳатидан ўзингизни қаноатлантирувчи CMS ни ҳосил қиласиз. Стандарт модуллар тўпламига блог, форум, хабарлар лентаси, файлларни юклаш, овоз бериш, қидирув кабилар киради. Бунда кўринадиган камчилик битта: PHP ва MySQL ни мукамал билиш лозим. Агар бу қисқартмаларни тушунмаган бўлсангиз, кейинги CMS га ўтаверинг, акс ҳолда эса бу тизимга жиддийроқ эътибор беринг.

**Slaed CMS.** Бугунги кунда анча оммабоп CMS бўлиб, турли хил сайтларда ишлатилади. Икки хил варианты мавжуд: пуллик ва бепул (OpenSlaed). Иккови ҳам битта ядрога қурилган (бу дегани хавфсизлик даражаси бир хил) ва фақатгина функционалликка таъсир қилувчи модуллар сони билан фарқланади. Бепул вариантыда бирор мавзудаги нисбатан кичик портални яратишингиз мумкин. Аммо, бундан ортиғини эмас. Дизайни ҳам унчалик катта қийинчиликларсиз ўзгартирилиши мумкин. Тизим аслида Россияда яратилгани учун рус тилидаги ёрдам ҳақида қайғурмаса ҳам бўлади.

**Joomla!** Бугунги кунда энг мукамал движоклардан бири (агар бепуллари ичида энг мукамали бўлмаса) бўлиб, унинг учун шунчалик кўп қўшимча модуллар яратилганки, улар ёрдамида ўзингизнинг сайтингизга деярли ихтиёрий функционалликни беришингиз мумкин. Бу дизайнга ҳам тегишли бўлиб, уни ўзгартириш учун ҳам жуда кўп шаблонлар мавжуд. Бироқ, бу ерда бир камчилик ҳам бор: шаблонлар тизими шундай тузилганки, саҳифаларни яратиш ва тўлдириш жараёнида сайт бўлимчалари кичик-кичик қисмлар бўлиб, ажралиб қолиши мумкин. Яъни, бу қисмлар жадваллар асосида яратилган. Бу эса сайтингизга ўз шахсий кўриниш беришингизда бир қатор ноқулайликлар туғдиради: юқорида тилга олинган қисмчалар қутилмаган

жойда чиқиб қолиб, жаҳлингизни чиқаради, шунингдек, уларни йўқотиш учун вақтингиз кетади. Умуман олганда, агар сизни тайёр шаблонлар кониқтирса (улар эса жуда кўп ва хилма-хил), у ҳолда бу тизим сизга жуда мос келади. Агар сиз Joomla! ни танлаб ўзига хос оригинал сайт яратмоқчи бўлсангиз озгина терлашингизга тўғри келади. Расмий ёрдам сайти: [joomla.org](http://joomla.org). Шундай қилиб, юқорида келтирилган тавсиялар асосида сиз ўз сайтингизни тез ва осон равишда яратишингиз мумкин. Аммо ҳеч қайси CMS сизга тайёр ҳолдаги сайтни тақдим этмайди. Бунинг учун сиздан ҳаракат ва меҳнат талаб этилади. Яратаётган сайтингиз бошқаларникига ўхшамаслиги, дизайн жиҳатдан такрорланмас ва оригинал бўлиши учун эса шахсий фантазиянгиз ва қобилиятингизни ишга солишингиз керак.

## **2.2 CMS ларнинг афзалликлари ва камчиликлари.**

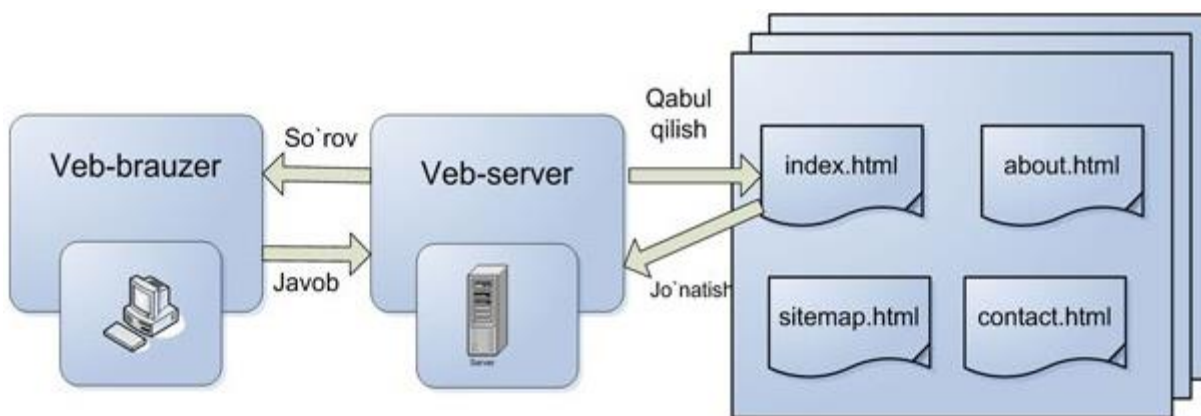
### **CMS ларнинг афзалликлари -Статик ва динамик веб-саҳифалар**

Енди савол туғилиши мумкин: оддий HTML кодида ёзилган саҳифа билан CMS нимаси билан фарқ қилади? Авваломбор, HTML да ёзилган сайт статик ҳолда бўлса, замонавий сайтлар динамик кўринишга эга.

Оддий сайт қандай принцип асосида ишлашини кўриб чиқамиз. Фойдаланувчи бирор-бир, масалан, page.HTML саҳифасига мурожаат қилди дейлик. Веб-сервер ўзидаги веб-саҳифалар каталогини кўриб чиқади ва агар page.HTML бўлса, фойдаланувчига уни кўрсатади, акс ҳолда эса – 404-чи хабарни (“Файл топилмади”) етказди.

Динамик сайт эса биров бошқача ишлайди. Фойдаланувчи бирор саҳифага мурожаат қилади, бироқ бу мурожаат дискдаги аниқ бир файлга эмас, балки сайт движокига (агар движок PHP тилида ёзилган бўлса, одатда бу файл index.php бўлади) қаратилган бўлади. Движок фойдаланувчига керакли маълумотни бирор манбадан (масалан, маълумотлар базаси ёки дискдаги файлдан) олади, уни HTML-саҳифага айлантиради ва веб-серверга узатади, у эса ўз навбатида, фойдаланувчи браузерига жўнатади.

Унда яна бир савол туғилади: агар битта файлнигина сўраб, уни браузерга узатиш мумкин бўлса, бунча мураккаблаштиришнинг ўзи нимага керак? Шунинг учунки, бу сайтни бошқариш қулай бўлсин. Айтайлик, сиз 100 саҳифалик (бу унчалик кўп эмас) оддий сайт яратдингиз. Эртами-кеч сиз бу сайт дизайнини, кўринишини ўзгартирмоқчи бўласиз. У ҳолда эса сиз шу 100 саҳифани қўлда ўзгартириб чиқишингизга тўғри келади. Устига-устига, сиз аввалига бу 100 саҳифани компютерингиздаги локал серверга юклаб олишингиз, ундан сўнг таҳрирлаб, керакли ўзгартиришларни киритишингиз ва охирида яна серверга юклашингиз керак бўлади.(3-расм)

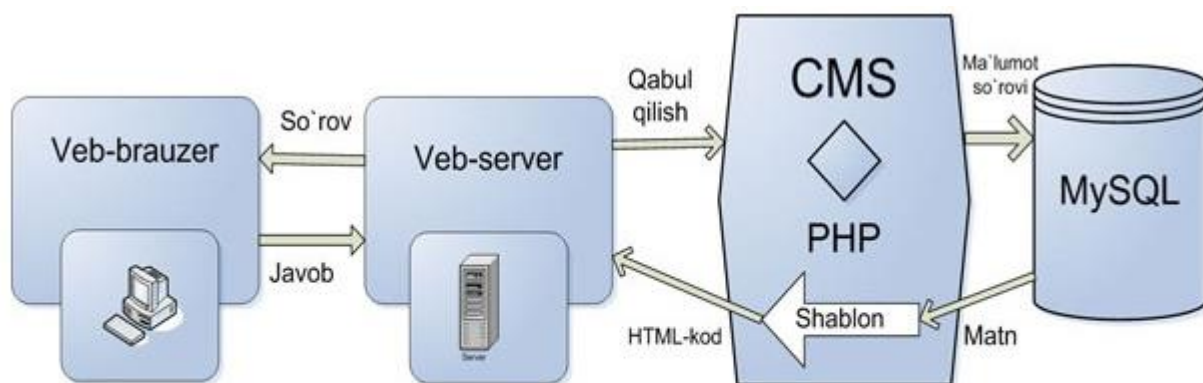


Veb-serverning CMS siz holda qanday ishlashi

3-расм(веб сервернинг CMSиз холда ишлаши)

Ана энди динамик сайт қандай ишлашини кўриб чиқайлик. Асосий воқеа – “HTML-саҳифани яратиш”. Яъни движок контентни – матнли маълумотни (у расм ва жадваллар билан ҳам бўлиши мумкин) олади ва уни дизайнга “кийинтиради”. Демак, сайтнинг дизайнини ўзгартириш учун сиз фақатгина тайёр график мавзунини ўрнатишингиз ёки бир-иккита файлни – сайт шаблони файлларини ўзгартиришингиз керак.

CMS тизими саҳифаларни динамик кўринишга келтиради: улар фойдаланувчи саҳифани кўрмоқчи бўлгандагина амалда яратилади. Бу дегани, саҳифалар фойдаланувчи томонидан бевосита кўрилаётганда янгиланиши ёки ўзгартирилиши мумкин.(4-расм)



Veb-serverning CMS bilan qanday ishlashi

#### 4-расм(веб сервернинг CMS билан бирга ишлашиши)

##### CMS афзалликлари

Бу тизимларнинг асосий жиҳати шундаки, динамик веб-саҳифаларни осон ярата олиш ва уларни турли хил ахборот билан тўлдириш имкониятини беришдир. CMS тизимлари одатда мураккаб бўлиб, улар ўз ичига хабарларни тарқатиш хизматлари (Rss), форумлар ва интернет-дўкониларни олади. Шу билан бирга, уларни ўзгартириш ҳам осон.

CMS нафақат сайтдаги турли маълумотларни бошқаришни автоматлаштиради, балки шу билан бирга, сайт тузилишининг техник томонлари билан таниш бўлмаганлар, яъни ёзувчи ва журналистларга сайтда ўз материалларини бевосита фойдаланувчи интерфейси орқали кўшиш имкониятини беради. Шу нарса тушунарли бўлдики, энди фойдаланувчиларга сайтда ўз материалларини кўриш учун HTML ни ҳам, сайтдаги маълумотлар янгиланишининг технологиясини ҳам билиши шарт эмас.

CMS тизими, шунингдек, веб-сайт яратишни турли хил мутахассислар орасида тақсимлаб чиқиш имкониятини беради. Шунга кўра, веб-дизайнер ўз диққатини сайтнинг кўринишига (дизайнни мукамаллаштириш ва шаблонларни ўрнатишга) қаратса, техник йўналишда бўлмаган мутахассислар уни маълумотлар (матн, тасвир ва аниматсиялар) билан тўлдиришга ҳаракат қилади.

Бу ҳам ҳаммаси эмас. Маълумки, движок – CMS нинг бир қисми холос. У дизайн шаблони асосида сайт маълумотларини чиқариб беради. Бундан ташқари, CMS да сайтдаги маълумотларни бошқариш (улар сайт бўлимларини ва категорияларини яратишга ва маълумотларни улар орасида бўлиб чиқишга имкон беради) ва бу маълумотларни таҳрирлаш воситалари ҳам мавжуд. Шу билан бирга, фойдаланувчи сайт маълумотлари айнан қаерда (маълумотлар базасида, дискда HTML-файл кўринишида ёки умуман ташқи серверда) сақланаётганини билиши шарт эмас.

CMS қўлланилиши билан кўпгина қийинчиликлар бартараф этилиши мумкин. Юқорида мисол сифатида кўрилган сайт харитаси ҳам автоматик равишда тузилади, бунинг учун фойдаланувчидан ҳеч қандай меҳнат талаб этилмайди. Шу билан бирга, CMS сайтдаги қўшимча функцияларни, масалан, суратлар галереяси ёки рўйхатдан ўтиш ойнасини қўллаш имконини беради – шу каби функциялар CMS кенгайтмалари орқали амалга оширилади. Фақатгина у ёки бу кенгайтмани ўрнатиш керак холос.

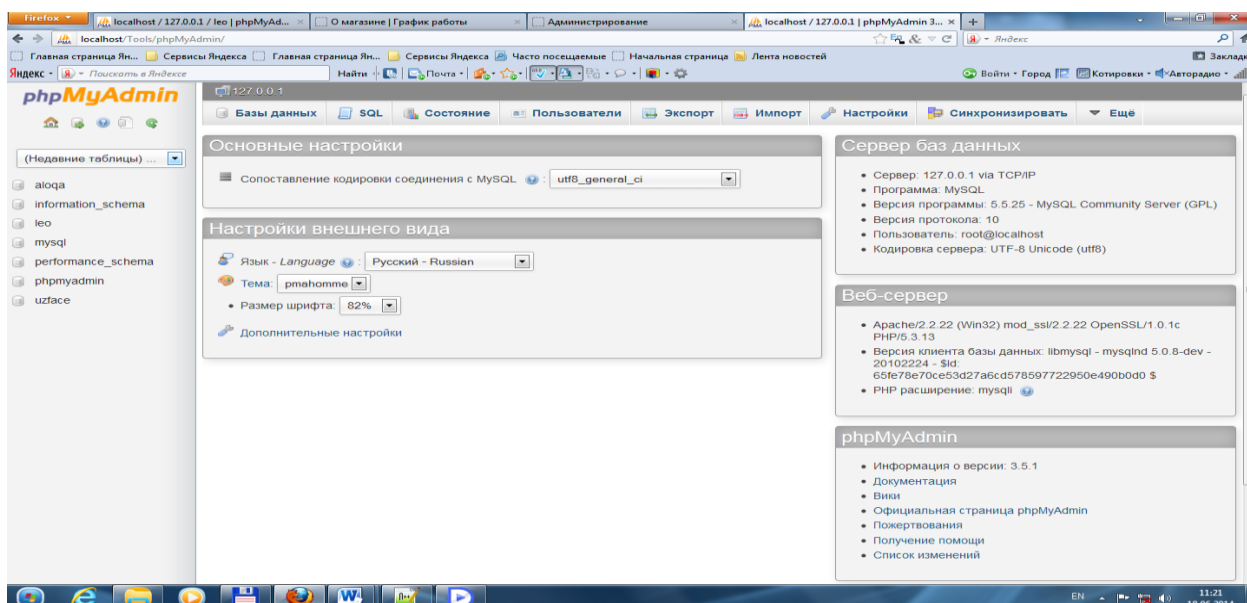
CMS га ўтишнинг ўзига хос қийинчиликлари

Хўш, CMS ларнинг шунча афзалликлари бўлганда нима учун барча сайт эгалари Joomla! ёки Drupal каби CMS ларга отишга интилмапти? Бунга бир қанча сабабларни кўрсатиш мумкин. Биринчи сабаб, инертсия бўйича кетиш ёки бошқача айтганда, ўзи билган нарсага ўрганиб қолиш. Инсон HTML ни ишлатишга ўрганиб қолади ва CMS га ўтиш сайтни бошқаришни автоматлаштириш бўйича қанча қулайликлар келтирмасин, бирор нарсани ўзгартиришни хоҳламайди. HTML дан CMS га ўтиш-йиллар ичида амалга ошиши мумкин. Иккинчи сабаб, мавжуд минглаб HTML-саҳифаларни CMS маълумотлар базасига мослаб ўзгартириш керак бўлади. Аксарият ҳолларда буни қўлда бажариш керак бўлади. Катта сайт учун бу жараён анча кўп вақтни олади, бу эса сайтни вақтинчалик фаолиятининг тўхташига ва балки, компания томонидан кўриладиган молиявий зарарга олиб келади. Албатта, CMS га ўтиш сайтни бошқариш бўйича анча вақт тежалишини ҳисобга олса, келажакда бу ўзини оқлайди.

Энди, CMS га ўтишга ҳалақит берувчи фақатгина битта тўсиқ қолади – техник ҳолат. Деярли барча замонавий CMS ишлаши учун хостингда PHP интерпретатори ва MySQL маълумотлар базаси сервери бўлиши керак. Беш-олти-йил аввал шу каби функцияларни тақдим этувчи хостинг-провайдерлар кам эди. Ҳозирга келиб, вазият яхши тарафга ўзгарди ва PHP ҳамда MySQL ни қоллайдиган хостингни Ўзбекистоннинг ўзида топиш қийинчилик туғдирмайди.

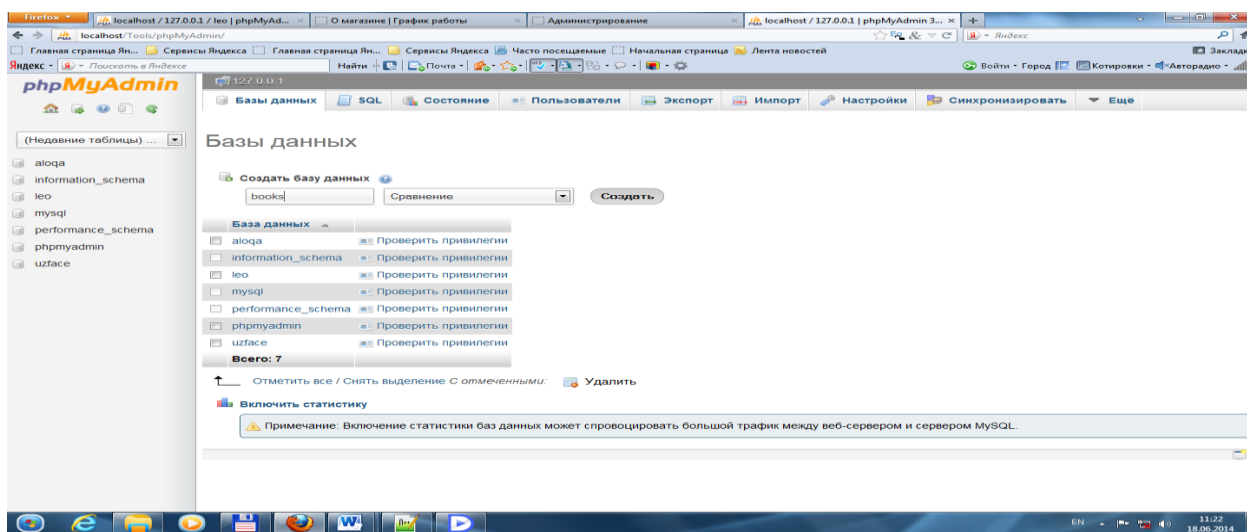
## 2.3 CMS технологияси – Lego Spда сайт яратиш.

CMS технологияларидан бири бўлган Lego Spда сайт яратиш учун дастлаб виртуал сервер “Denwer” ни ўрнатамиз. Ундан сўнг ҳосил бўлган ўзингиз кўрсатган дискга (одатда “z” билан белгиланади) кириб ичидаги “home” папкасига сайт номи остида папка очамиз, унинг ичига “www” папка очамиз ва уни ичига Lego Sp сайтини ташлаймиз ва браузерни ишга тушираемиз. Браузерни ишга туширганимиздан сўнг “localhost” номини ёзамиз ва уни чақирамиз. “localhost”да биз сайтимишнинг базасини яратамиз. Бу эса пастдаги расмда акс этган.



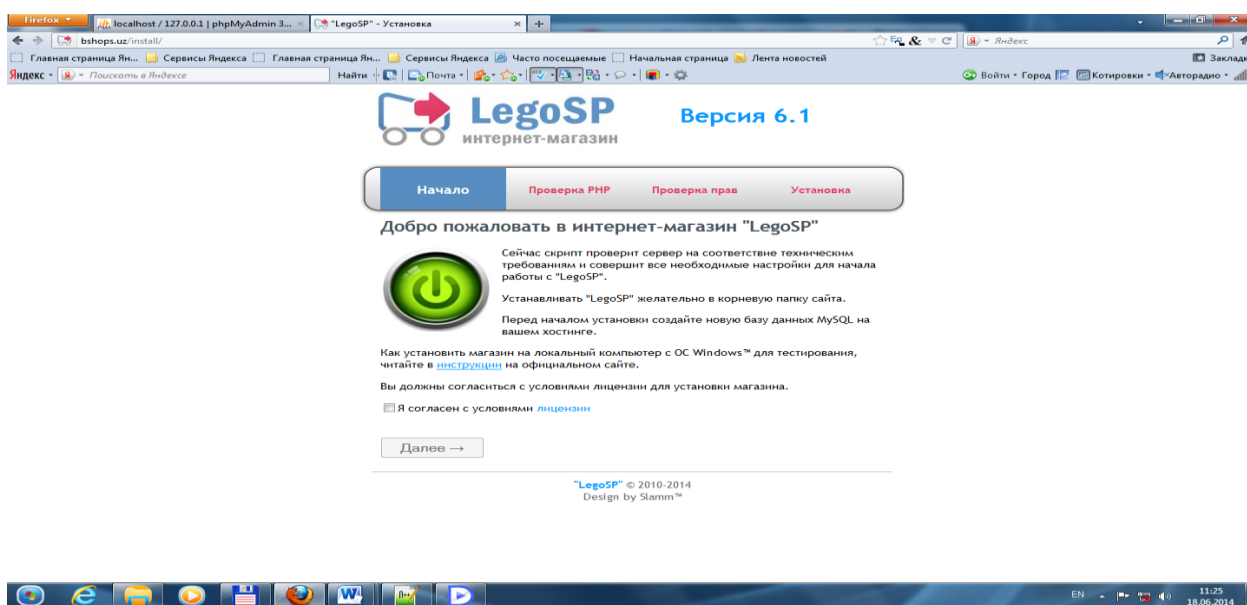
5-расм (“localhost” чақирилиши)

Бу ерда биз “Базы данных” тугмасини босамиз ва базамизнинг номини киритиб “Создать” тугмасини босамиз. Шу билан “books” номли база яратилди.



6-расм(база яратилиши)

Базамизни яратиб олганимиздан сўнг сайтимизни ўрнатишни бошлаймиз. Браузердан сайтимизни чақирамиз. Натижада куйидаги ойна ҳосил бўлади. Бу ойнада шартларга розилик билдирилади ва галичка қўйилиб “Далее” тугмаси босилади.



7-расм(Lego SP ни ўрнатиш бошланмоқда)

Қуйидаги ойнада эса сайтимиз янада яхшироқ ишлаши учун РНРнинг баъзи бир модулларини ўрнатамиз бунинг учун эса модулларни текшираемиз ва “Далее” тугмаси босамиз.



## 8-расм(РНР модуллари ўрнатилмоқда)

Кейинги ойнада эса ўрнатилиши керак бўлган файлларни текшираемиз ва яна “Далее” тугмаси босамиз.



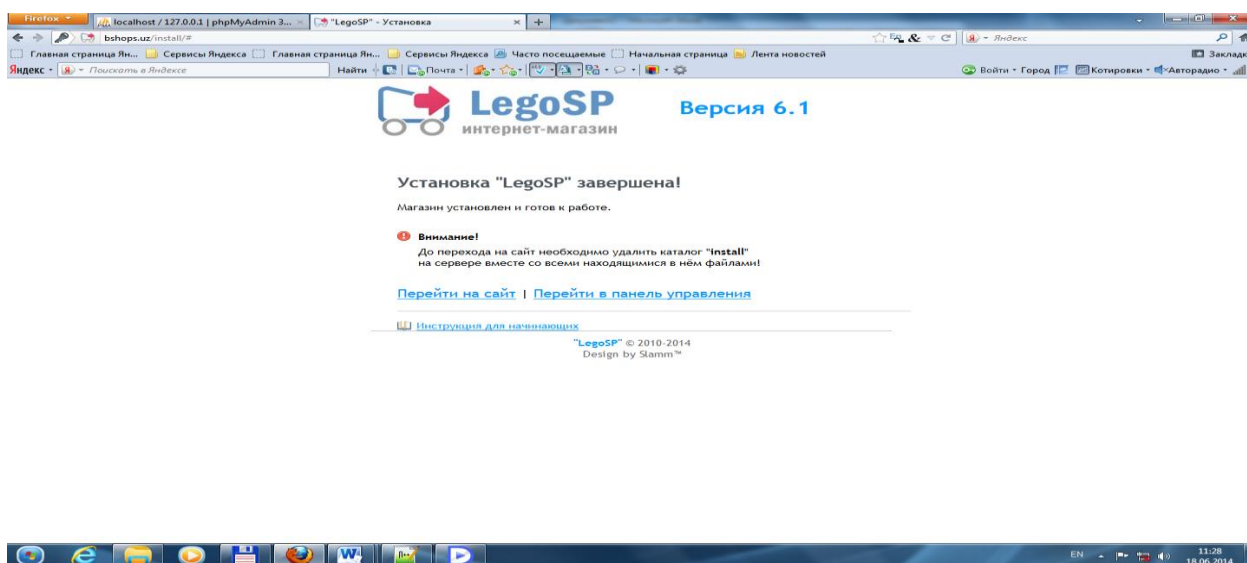
## 9-расм(файллар ўрнамоқда)

Бу ойнада эса сайтни ўрнатишни охириги босқичини амалга оширамиз яни бу ойнада сайтимиз номи,алоқа учун “email” манзилимизни, сайтнинг администратори номи вапаролини, базамиз номи ва паролини киритамиз. Сўнгра «Установить» тугмасини босамиз.



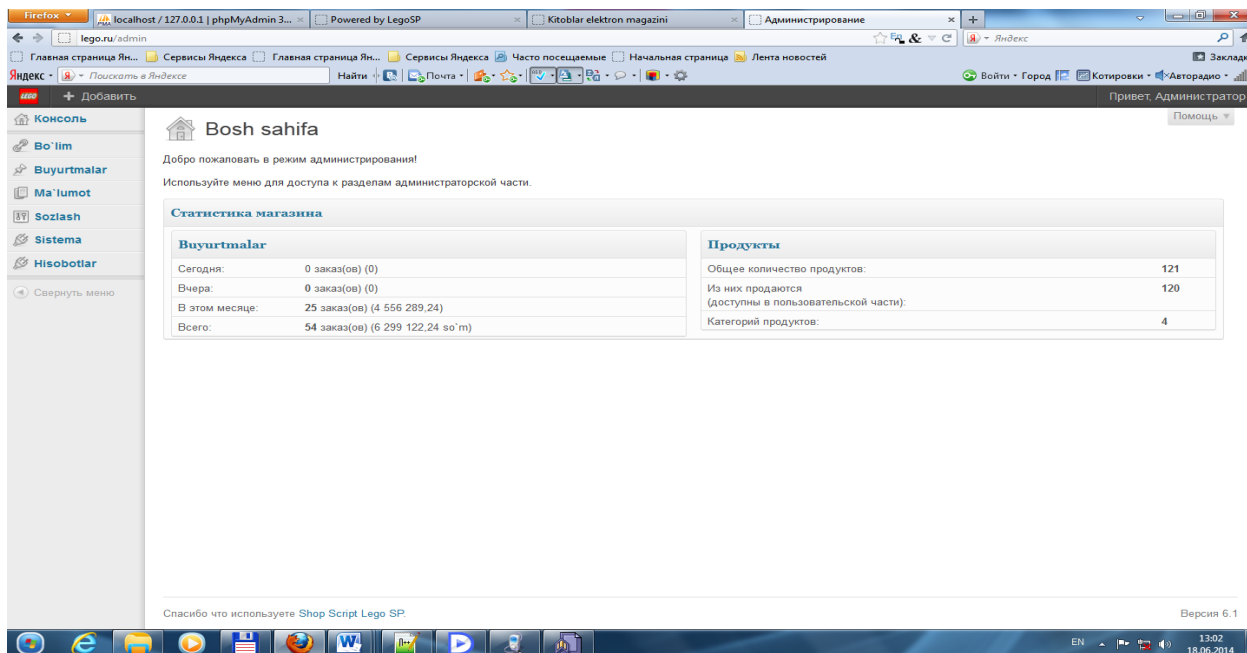
### 10-расм(сайт ва база номлари киритилмоқда)

Шу билан ўрнатиш жараёни тугайди. Охириги ойнада эса сатимизнинг “install” папкасини ўчириб юбориш кераклигини эслатмоқда ва биз бу папкани ўчириб юборамиз.



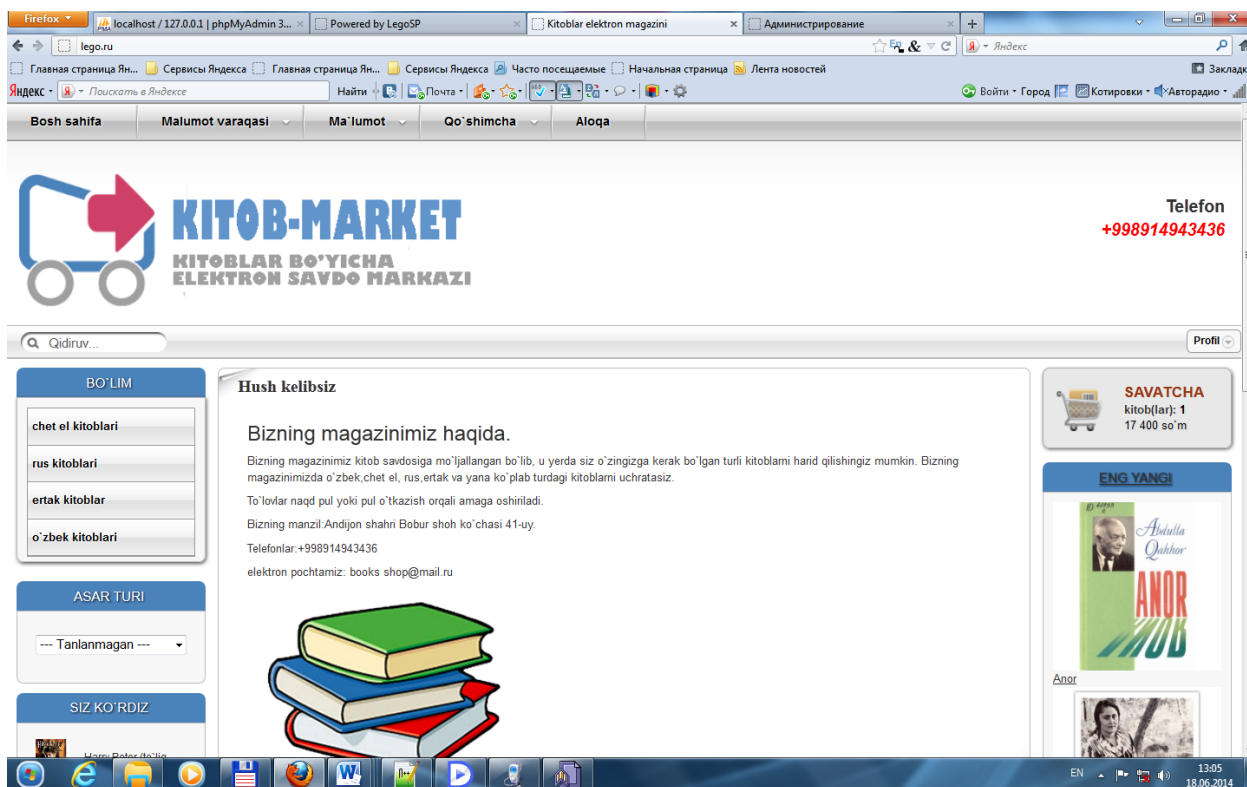
### 11-расм(ўрнатиш жараёни якунланмоқда)

Бу ойнада сайтимизни администратор бўлими акс этган.



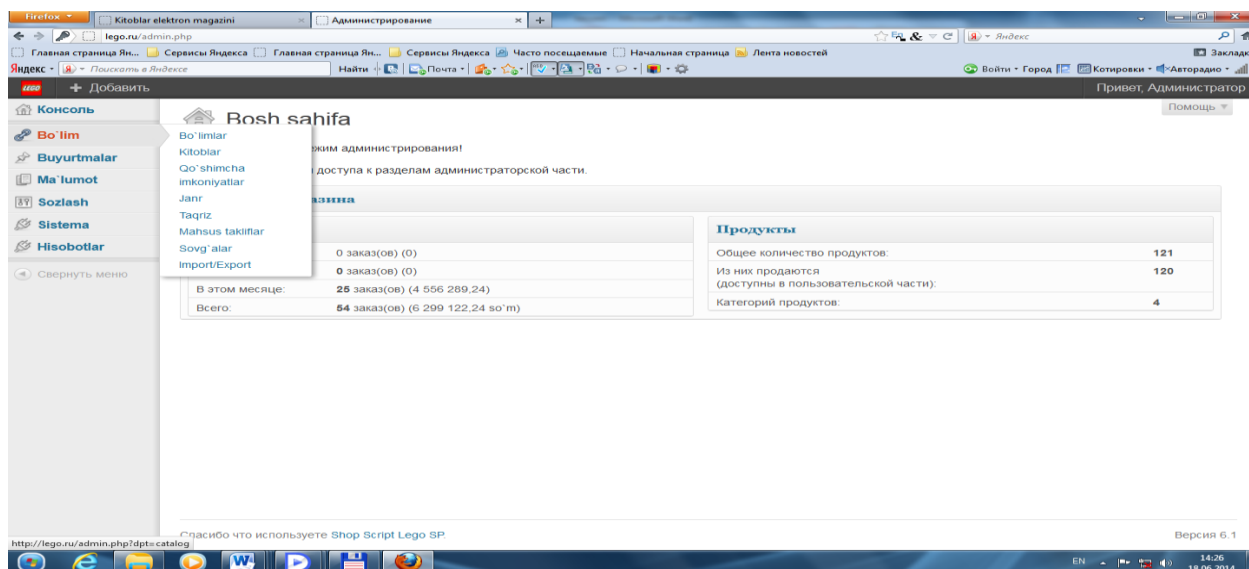
12-расм(сайтнинг администратор бўлими)

Бу ойнада сайтимизни кўриниши акс этган.



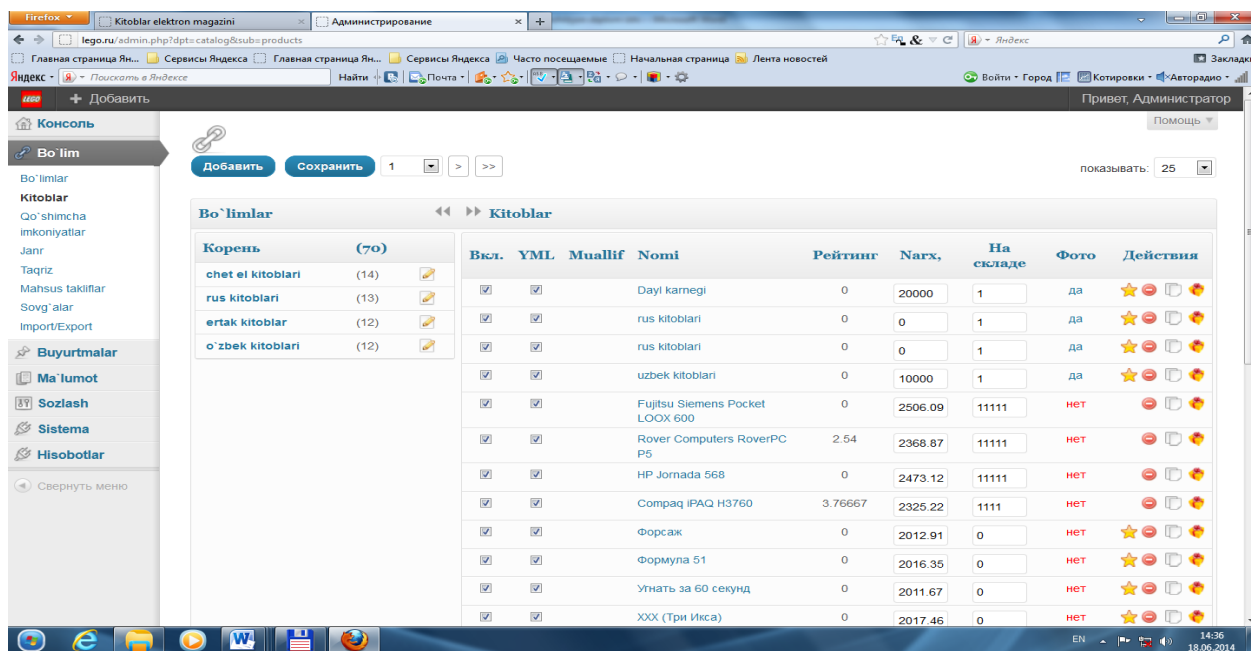
13-расм(сайтнинг бош саҳифаси)

Энди биз сизлар билан администратор томонидан базага янги китоблар кўшишни кўриб чиқамиз. Бунинг учун биз сайтнинг администратор қисмига кирамиз ва “Во`лимлар” менюсига танлаймиз. У ердан “Kitoblar” бўлимига кирамиз.



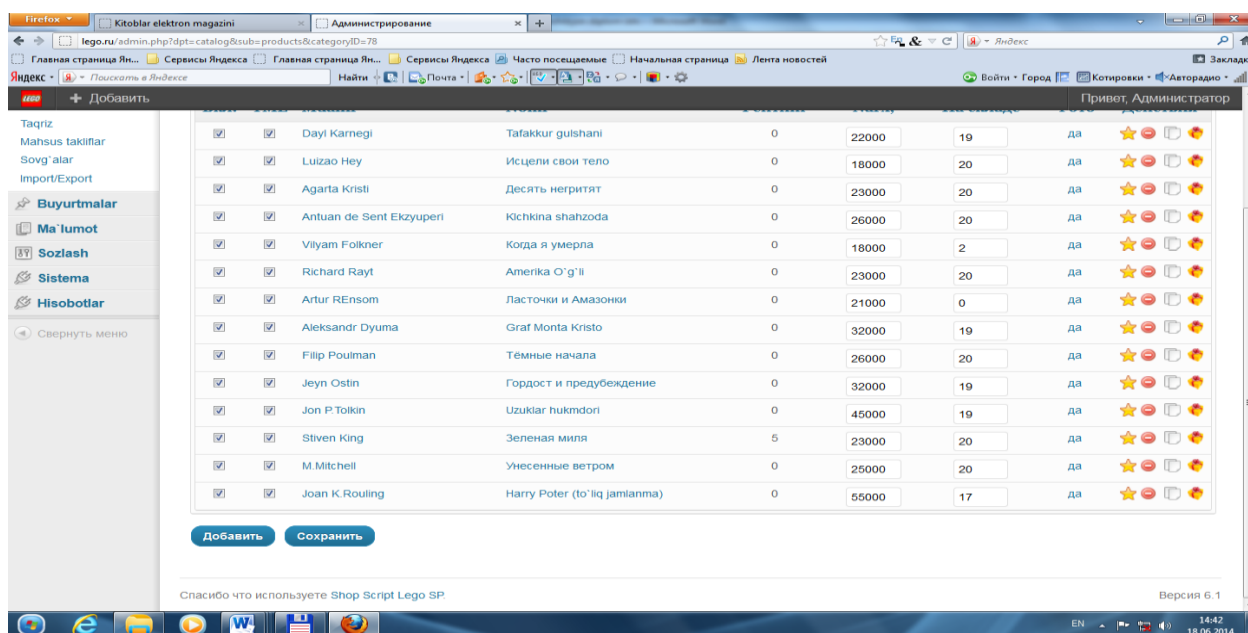
14-рasm(Администратор бўлимининг менюлари)

Бу ерда китобларнинг турлари жойлашган бўлиб, биз кўшмоқчи бўлган китоб қандай асар эканлигига қараб бирор бўлимни танлаймиз. Мисол учун “chet el kitoblari” бўлимини танлаймиз.



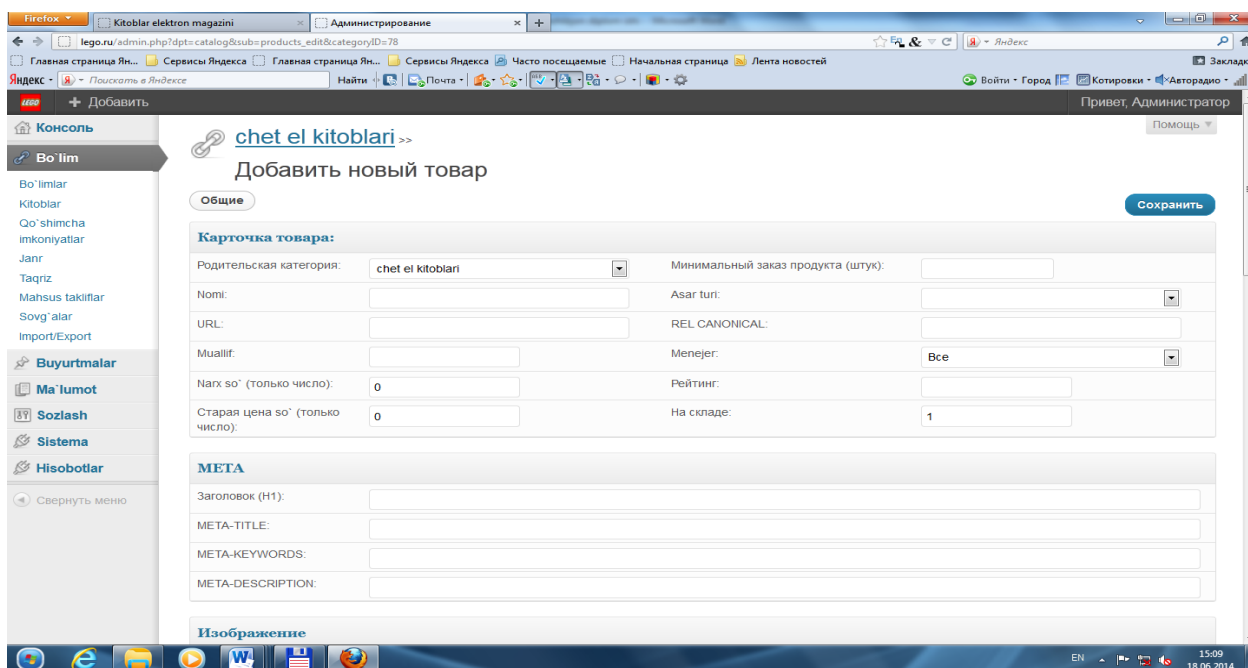
15-рasm(китоб тури танланмоқда)

Шундан сўнг бу бўлимдаги китоблар рўйхати чиқади ва уларнинг қуйи қисмида эса “добовить” тугмаси жойлашганб биз шу тугмани босамиз.



16-расм(янги китоб кўшилмоқда)

Сўнгра экранимизда қуйидаги ойна пайдо бўлади. Бу ойнада китобнинг номи, китоб муаллифи, қайси жанрда ёзилганлиги, расми, китобга тавсифлар, нархлари ёзилади ва “сохранить” тугмаси босилади.



17-расм(янги китоб кўшилди)

### **III. Тадбиқ қилишни ташкил қилиш ва лойиҳа самарадорлиги**

#### **3.1 Веб сайтлардан фойдаланишнинг самарадорлиги**

Интернет технологияларининг яратилиши турли манбалардан тез ва осон йўл билан ахборот олиш имкониятларини ҳамма учун-оддий фуқародан тортиб йирик ташкилотларгача мисли кўрилмаган даражада ошириб юборди. Давлат муассасалари, фан-таълим муассасалари, тижорат корхоналари ва алохида шахслар ахборотни электрон шаклда яратиб-сақлай бошладилар. Бу мухит аввалги физикавий сақлашга нисбатан катта қулайликлар туғдиради: сақлаш жуда ихчам, узатиш эса бир онда юз беради ва тармоқ орқали бор маълумотлар базаларига мурожаат қилиш имкониятлари жуда кенг. Ахборотдан самарали фойдаланиш имкониятлари ахборот миқдорининг тез кўпайишига олиб келди. Бизнес қатор тижорат соҳаларида бугун ахборотни ўзининг энг қимматли мулки деб билади.

Бундан ташқари давлат бошқарув органларининг ҳам электрон ҳукумат шаклига ўтиши интернетнинг қай даражада қулай ва самарадорлигини тасдиқлайди.

Самарали эканлиги яна бир далил сифатида қилиб қуйидагини кўрсатиш мумкин. Бир неча сониялар ичида фан, таълим, технология ютуқлари ҳақида маълумот олишингиз, уни кўришингиз ва хаттоки сотиб олишингиз ҳам мумкин.

### **3.2 Веб сайтларнинг бугунги кундаги зарурияти.**

Веб сайтларнинг нечоғлик зарур эканлигини, интернетнинг нечоғлик зарур эканлиги билан солиштирсак менимча муьолаға бўлмаса керак. Чунки интернет тармоғини Веб сайтлар ташкил қилади.

Дунёнинг бу бурчагидан туриб, у бурчагида маълумотни тез ва қулай тарзда ўқиш имконияти ва дунёдаги янгиликлардан боҳабар бўлиш имконияти билан интернет ҳозирги замоннинг энг керакли ва энг зарур тармоғи десак муьолаға бўлмаса керак. Бугунги кунда исталган жойдаги маълумот, янгилик ва хабарларни бир неча сония ўқиш ва кўриш имконияти мавжуд.

Бугунги кунда замонавий технологиялар фойдаланмайдиган инсоннинг ўзи бўлмаса керак. Оддий мактаб ўқувчидан тортиб университет талабасигача, оддий ўқитувчидан тортиб фан доктори, профессоргача ҳозирги пайтда инфарматсион технология ютуқларидан фойдаланади. Албатта бу технологияларни интернетсиз тасаввур қилиш анча қийин. Интернет бизнинг ҳаётимизга шунчалар тез ва жадал суратлар билан кириб келмоқдаки, усиз ҳаётимизни тасаввур қилиш қийин. Интернет ва унда фойдаланиладиган веб сайтлар жамиятимизнинг барча сохаларига чуқур кириб борган. Хусусан спорт, таълим, тижорат, саёҳат ва бошқа бир қатор йўналишларда веб сайтларнинг ўрни беқиёсдир.

## IV. МЕҲНАТ МУХОФАЗАСИ

### 4.1 Замонавий техника жойлашган хонасига талаблар.

Хонани шифти оқ кўк фон билан оқланиши ва деворлари эса яшил рангга оқланиши керак. Бу ранглар офтоб нурланишини бизга ранг иқлимини яратиб беради. Хоналарга қўйилган талаблар ишчи мухит ишчининг (оператор) иш жойи ташқи мухит факторлари йиғиндиси бўлиб улар қўйидаги ишлардан иборат: физик, химик, биологик, ахборот, социал-психологик ва эстетик факторлар ташқи мухит хоссалари бўлиб операторга таъсир этади. Ишчи мухит турлича бўлиши мумкин: иш жойида хаёт фаолиятини таъминловчи воситалар операторнинг талаб этилган меҳнат қобилияти шароитини хосил қилади ва уни нохуш факторлар таъсиридан химоя қилади.

Ходимлар самарали фаолият кўрсатиш учун шароит яратиш ва техник воситаларни ишлаш учун хоналар ёруғ, тоза, товуш ва тебранишдан изоляцияланган ҳолатда лойихаланади. Шкаф ва деворлар товуш ютувчи плиткалар билан қопланиши мақсадга мувофиқдир. Хона харорати оптимал хароратда 21-23<sup>o</sup>C да оптимал намлик 40-60 %, чанг концентрасияси 0,2 Мг/м<sup>3</sup> дан ва чанг максимал заррача ўлчаш 3 Мк дан ошмаслиги лозим. Хоналарда бундай шароитни ушлаб туриш мақсадида, хоналарни хаво алмаштириб туриш кўзда тутилади.

Операторнинг узок вақт давомида монитор ортида ўтириши натижасида кўриш апаратининг зўриқиши, ишдан қоникмаслик, бош оқриқи, уйқунинг бузилиши чарчоқ ва кўз, бўйин, бел, қўлларда оғриклар сезила бошланади.

ЕХМ операторининг иш жойи дейилганда техник манбалар ва ёрдамчи қурилмалар билан жиҳозланган конкрет ишлаб чиқариш масаларни ечишга мўлжалланган "оператор -одам" иш фаолияти билан шуғуланадиган худуд тушинилади.

Иш жойни меҳнат ҳавфсизлиги қоидалари ва стандартлар талабларига мос равишда жиҳозлаш керак. Иш жойи элеменларини жойлаштиришда

қуйидагиларга эътибор бериш керак : оператор одамнинг ишчи позаси операторга керакли харакатларни амалга оширувчи жой оператор ва усқунани боғловчи жисмоний, кўриш ва эшитиш алоқаси.

ишчи жойидан ташқарини кўриш имконияти ёзиш ҳамда оператор томонидан ишлатиладиган хужжатларни сақлаш имконияти усқунанинг ташқи ва конструктив кўринишини жиҳозлаш минимал чарчаш учун шароит яратади.

Операторнинг иш жойини тўғри ташкил этилганида унинг меҳнат унумдорлиги 8-20% ошади. Компютер ўрнатиладиган хонага компютер сонига қараб туриб қуйидаги талаблар қўйилади: ахборотлаштириш, бутингловчиларни ёки ишловчиларнинг компютерда назарий ва амалий машғулотлар ўтказиш билан бажарилади. Шунинг учун компютер хонасида 2 тадан 5 тагача компютер ўрнатилиши мумкин. Шунинг учун компютер хонасини ўлчамлари қуйидагича бўлиши керак (3х6х2,8 м).

Стол ва стулларга талаблар:

Компютер хонасида стол ва стулларга талаблар мавжуд бўлиб стол баландлиги ердан 68-77 см бўлиб, стуллар эса айланувчан бўлиши керак ва орқасида суянчиғи бўлиши керак. Чунки стол стуллар ўз габарити билан тўғри келмаса фойдаланувчи тезда чарчаб қолади. Стол ва стуллар шундай жойлаштирилиши керакки, улар инсонларга туриб юришга халақит бермаслиги керак. Бундан ташқари, операторлар бемалол ҳар бир операторлар олдига бориб бирга ишлай олиши керак.

Иш жойининг конструктивиги ва элеменларининг жойлашинуви (ўтирғичлар, ахборотнинг кўрсатиш, бошқариш органлари) антропометрик, физиологик ва психологик талабларга ҳамда ишнинг характерига тўғри келиши керак.

Шундай конструкцияланган иш жойи мотор майдонидан ташқаридаги бажарилиши қийин бўлган операсияларни бажариш имконини беради. Ахборотнинг кўрсатиш манбалари бу ҳолда ЭХМ нинг дисплеи СНИП 2.01.02-85 (5)га тўғри келади. Кўзга тушаётган нағрузкани камайтириш учун

дисплей эргономика нуктаи назаридан оптимал ўрнатилиши керак, дисплейнинг тепа бурчаги кўз билан бир текисликда бўлиши керак, экрангача масофа 28-60 см бўлиши керак. Экранинг милтилаши мил>70 Гс бўлиши керак.

Антропометрик мос тушиши операторнинг иш бораётган вақтида фазода, кенгликда тананинг жойланиши имконияти ва турли позани эгаллаши назарда тутилади. Бу масалани хал қилиш учун биринчи навбатда бошқариш пулти асбобларидан операторнинг оёғи бориб етадиган зона аниқланади. Бу мос келишини таъминлаш қийинчилик билан эришилади, чунки хар бир кишининг антропометрик кўрсаткичлари турлича. Ўрта бўйли кишини қониктирган ўриндик, баланд ёки паст бўйли бўлган кишига ноқулай бўлиши мумкин.

Хавфсиз фаолият кўрсатиш мақсадида инсон танаси ўлчамлари куйидаги ҳолатларда ҳисобга олинади:

- полдан ёки иш майдонидан, машиналар ишлашини назорат қилиш, тўғрилаш зонаси, сигнализасия ва назорат асбобларига бўлган сатхни оптимал баландлигини ўлчашда.

- баландликда кўлда бошқариладиган машиналар фронтини жойлаштиришда, айникса авария органларининг пухта жойлаштиришда.

- бошқариш органларини шакли ва ўлчамларини танлашда.

Машиналарни лойихалашда инсон антропометрик кўрсаткичларни тўғри танлаш учун узини топография қилиш усули ёки моделлаш усули қўлланилади. Ўзини топография қилишда инсон ишчи танасини турли ҳолатларини схематик конструкиялаш ва ишчи бажарадиган ишлар ва операсиялар билан боғлаш киради. Моделлаш усулига инсон фигурасини ҳажмий ва текисликда моделлаш киради. Инсоннинг антрапометрик куйидагича: ўртача баландлиги 1 метр 72 см, елка кенглиги 39 см, қўллар ёйилмаси 160 см агар бу антропометрик ўлчовлар ҳисобга олинмаса операторлар иш пайтида бир-бирига халақит бериши мумкин. Шунинг учун антропометрик ўлчовларни ҳисобга олиш катта аҳамиятга эга.

Монитордан инсоннинг кўзигача бўлган оптимал масофа

Монитор кўздан озгина пастрокда ва 50 см дан кам бўлмаган масофада жойлашиши керак. Монитор ва кўз орасидаги масофа 80 см гача бўлиши тавсия қилинади, бу масова кичик бўлса инсоннинг кўзи тез чарчайди. Мониторни дизайни ва ранги ўзига эътиборни жалб қилмаслиги керак. Шунинг учун мониторнинг сирт томонида хар хил реклама ёпиштиргичлар бўлмаслиги керак. Мониторнинг экрани зангори ва кўк рангларга бўялиши мақсадга мувофиқ хисобланади. Чунки бу ранглар инсон кўзига энг яхши ранглардан хисобланади. қисман манитор олдидаги ўтириш хавфсизликни ва иш жойини расионал ташкил этиш лозим. Фойдаланувчи усул асосий хавфсизлик видеоманитор экран дисплейдан чиқади деб бўлмайди. Энг кучли нурланиш одатда маниторни ён ва орқа томонидан хам тарқалади. Шунинг учун фойдалануви жойини бирнеча кампютер карама-қарши турган жойда ундан хам ёмони орқама-кетин жойлаштиришдир.

Видеоманитор хиллари орасидаги тавсия этладиган орадаги масофа 2 м дан кам бўлмаслиги ва ён томондаги масофа 1,2 м дан кам бўлмаслиги лозим. Кампютерлар жойлашган хона етарли даражада кенг ва доимий равишда хавоси алмашиб туриши керак. Битта дисплей учун минимал стандарт норма 6м ни, минимал хажим эса 20 м ташкил этиши керак.

Компютер билан ишлаганда чарчаш сабаблари

Компютер билан ишлаш вақтида инсон қуйидаги факторлардан чарчайди:

- экранинг меъёридан ортиқ ёруғлиги
- контраст ва фон ўртасидаги аниқлиги
- компютерда ишлаш пайтидаги иссиқликдан нурланиши
- компютерда нурланишнинг инсонга таъсири
- компютер бузуқлиги

Компютердан нурланишнинг олдини олиши учун химоя филтрларидан фойдаланилади.

## 4.2 Иш жойининг ёритилганлиги

Иш жойини лойихалаш вақтида суъний ва табиий ёритиш масаласи ҳал қилиниши керак. Ёритиш нафақат ишлаб чиқариш масаларини хал қилиш балки у ишлаётган одамнинг психологик ҳамда физик ҳолатига тасир кўрсатади. Ишлаб чиқариш жойларидаги рационал ёритганликка кўйилган талаблар :

- ёруғлик манбаи ва ёритиш тизимини тўғри танлаш.
- ишлаб чиқариш тепалигларини керакли даражадаги ёруғлик даражаси билан таъминлаш.
- кўзни оладиган ёруғликни чеклаш
- текис ёруғликни ташкилаш
- ёруғлик оқимининг вақтда тебранишин йўқотиш ёки чеклаш

Керакли даражадаги ёритилмаганлик оқибатида ва кўриш ҳолатининг зўриқишида бажарилётган иш давомида кўзнинг чарчаши кучайяди умумий ишлаши ва ишлаб чиқариш унумдорлиги тушиб кетади ва хатолар сони кўпаяди. Иш жойидаги ёритганлик гигиеник талабларга биниоан меҳнатнинг кўриш шароитларига тўғри келиши керак. Тарқатилган ёритишдан, шіфтларнинг, деворларнинг, ускуналарнинг оч рангларга бўяш кўланилади. Операторнинг кўриш майдонида ёруғлик майдони бўлса тўғри ялтираш, кўриш майдони ичида қайтарадиган ёруғлик текисликлари мавжуд бўлса қайтарувчи ялтираш дейилади. Тўғри ялтирашни кўриш майдонидан ярқилаган ёруғликни 60 см камайтириш йўли билан камайтириш мумкин. қайтарувчи ялтирашдан эса ёруғликни тарқатувчи манбалар ҳамда полировка қилинган

текисликлар ўрнига матовий ишлатиш йўли билан камайтириш мумкин. Экрaн мониторидаги бликларни камайтириш учун тасвирни контрастлигини кучайтирувчи ва бликларни камайтиртурувчи экран филтирларидан фойдаланиш керак ёки антиблик қопламаси мавжуд мониторлардан фойдаланиш зарур.

Ёруғликни турини танлаш мухим масала хисобланади (табий ёки суъний). Таъбий ёруғликдан фойдаланиш кўп камчиликларга эга :

- ёруғлик тушиши фақат бир томондан
- ёруғликни вақтда ва хажмда бир хил бўлмаганлиги
- равшан куёш нурларининг кўзни олиши ва бошқалар.

Суъний ёруғликдан фойдаланиш юқоридаги камчиликларни бартараф этади ва оптимал ёруғлик режимини яратишга ёрдам беради. лекин ойналарсиз иншотлардан фойдаланиш инсонларда ўзига ишончсизлик ва уялувчанликни келтириб чиқаради. Тўғри ёруғлик узатишни ташкилетиш учун куёш нурларига яқин суъний ёруғликни танлаш керак.

### **4.3 Електр ва ёнғин хавфсизлиги**

Электр токидан ерга улаб муҳофаза қилишда ерга улаш қурилмаси юзаки бўлиши мумкин, бунда шохсимон ҳолда ёки нурсимон тартибда ер остига 1 м чуқурликда 30 м дан кам бўлмаган нурлардан ташкил топган металл таёқлар ётиқ ҳолатда кўмилади. Чуқурлаштирилган ерга улаш қурилмаси эса 2-3 м дан кам бўлмаган металл трубаларни ёки бошқа шаклдаги металл таёқларни ер юзасидан 0,7-0,8 м чуқурликкача тик ҳолда қоқилади. Бу таёқчалар ва юқори усулдаги нурлар ўзаро металл тасмалар билан туташтирилиб кавшарланади. Бундай ерга улаш қурилмасининг электр токини ўтказишга қаршилиги 10 Ом дан ошмаслиги керак. Ҳар қандай электр қурилмаси, агар унинг металл корпусларида электр кучланиши ҳосил бўлиш хавфи бўлса, қайси жойда ва қандай бинода ишлатилишидан қатъи назар, унинг корпусини ерга улаб қўйилади ва бу электр қурилмаларини ерга улаб муҳофаза қилиш деб аталади.

Ерга улаб муҳофаза қилишнинг асосий моҳияти ишлатилаётган электр асбобларининг металл корпусларида электр кучланиши пайдо бўлса, уни ерга ўтказиб юборишдан иборат электр қурилмаларни ерга улаб муҳофаза қилишнинг асосий хусусияти, қурилма корпусига ўтиб кетган кучланишни хавфсиз кучланиш даражасига тушириш, шунингдек, ерга уланган жой

атрофида потенциаллар айирмаси ҳосил бўлмаслигини таъминлашдан иборат. Химояловчи ерга улаш қурилмалари 2 хил:

1. Контурли ерга улаш

2. Ташқарига чиқарилган ерга улаш - бу усул кўпинча уловчи асбоб-ускуналар турган жойдан ташқарига чиқариб маълум бир майдончага тўпланиб ўрнатилади. Ерга улашнинг бу тури асосан кучланиши 1000 В гача бўлган қурилмаларда ишлатилади. Бунинг афзаллиги шундаки, электрод вазифасини бажарувчи қозикларни ерга қоқиш учун қаршилиги кам бўлган ерларни танлаш имкони бор.

Электр токидан зарарланган инсонга биринчи ёрдам кўрсатиш:

- Шикастланган одамни электр токи таъсирида бир неча усуллар билан халос қилиш мумкин.

- Агар ток урган хушидан кетган бўлса унга медицина ёрдамини кўрсатиш

Ёнғин хавфсизлиги

Электр тоқларини қисқа туташуви натижасида кучланиш ортиб қизиш юзага келади. Натижада ёнғин чиқиш хавфи туғилади. Бу ёнғин чиқиш сабабларидан бири хисобланади. Хоналарнинг ёнғинга қарши тоифасига қараб бўлинади. Агар ёнғин чиққан пайтда оператор дархол ёнғинни сабабини билиши ва уни бартараф этиш усулларини кўриши лозим. Бунинг учун электр токидан ёнғин чиққан бўлса линияни электр токидан узиб, сўнг ўчиришга киришиш керак. Шу билан биргаликда ўт ўчирувчи гуруҳларга хабар қилиши лозим. Ёнғин кучайган ҳолатда эвакуация йўллари орқали (қўшимча чиқиш эшиклари) ишчиларни эвакуация қилиш керак. Эвакуация йўлларининг ёритилиши камида 5 лк бўлиши,

йўлакларининг эни эса ишчилар сонига нисбатан кенг бўлиши керак. Хар бир ташкилотда ёнғин учун сув таъминоти бўлиши керак.

Карбонат ангидрид ёрдамида ўт ўчириш.

Карбонат ангидрид CO<sub>2</sub> - инертли, рангсиз газ, солиштирма оғирлиги 1.524, кучсиз хидли ва сувда яхши эрийди 100 С температурада ва 36 атм.

босими суюқ холатга келади. Ўз хажмидан 0.002 га кўпаяди: суюқ холатидаги карбонат ангидрид буқланганда 509 л газ хосил қилади. Ўт ўчирувчи мосламадан чиққан карбонат ангидрид (5000 мертгача) кенгаяди ва қаттиқ дур холатига ( қор кўринишига) келади. Карбонат ангидрид ўт ўчириш хусусиятини шундан иборатки қор кўринишидаги угле кислород ёнғин ўчоғи тушгандан катта миқдордаги иссиқлик ютиб келади. Карбонат ангидрид осон ёнадиган суюқликларни, қаттиқ моддаларни, кучланиш остида электр мосламаларини ўчиришда. Айниқса ўқув хоналаридаги лабораторияларда ёнғин ўчиришда яхши фойда беради. У текган нарсалар зараланмайди шунинг учун қимматли ва ноёб материалларни ўчиришда карбонат ангидрид қўлланилади. Унинг камчилиги шундан иборатки, у хўллаш хусусиятига эга эмас ва юқори босимда суюқ холатга ўтиш учун катта хажмдаги ва оғир металдан ясалган балонлардир

## V.ХУЛОСА

Хулоса ўрнида шуни айтишим мумкинки мен бу битирув малакавий ишини бажариш давомида "PHP дастурлаш тили" билан пухта танишдим. Унинг қачон яратилгани, тарихи, операторлари, массивлар имкониятларини пухта ўргандим. Бундан ташқари Drupal, PHP-Nuke, WebDirector, Joomla!, NetCat, Microsoft CMS, WordPress, PHPShop каби CMS лар ҳақида кўплаб маълумотларни ўргандим. Lego SPда сайт яратишни мукамал даражада ўргандим.

Замонавий веб дастурлаш тилларига кирувчи PHP дастурлаш мухити модуллар билан ишладим. Замонавий дастурлаш технологиялари PHP дастурлаш муқотида дастур ёзиш фойдаланувчилар ишини енгиллаштиришини билдим. Дастурларни мустақил тузишни ўрганмоқчи бўлган фойдаланувчилар учун яратилган PHP веб дастурлаш тили электрон қўлланмадан дарс машғулотларда фойдалансалар мақсадга мувофиқ бўлар еди.

Мен ўз битрурув малакавий ишимда китоб савдосини амалга оширурувчи веб сайт яратдим. Бу сайт фойдаланувчилар учун жуда қулай бўлиб,улар ўзларига керакли бўлган китобларни интернет орқали ҳам харид қилишлари ҳам мумкин. Сайтда қидирув тизими,сайт харитаси тизимлари мавжуд бўлиб менимча бу имкониятлар сайтдан фойдаланувчилар учун жуда ҳам катта қулайлик туғдиради. Ҳозирги кунда бизнинг веб сайтимиз интернет тармоғида ишламоқда.

Веб сайтни яратиш давомида шунга гувоҳ бўлдимки юртимизда бу сохани ривожлантириш юзасидан қатор ишлар олиб борилмоқда лекин шунга қарамай Ўзбекистонда бу борада хал қилинадиган ишлар кўплиги ва бу муамолларни электрон савдо хизмати тараққий этган давлатлар тажрибсидан фойдаланиб бартараф этиш мақсадга мувофиқ эканлини

айтишим мумкин. Бир ўйлаб кўрайлик хозирги пайтда интернет ҳаётимизнинг бир қисмига айланиб улгурди. Интернетдан ҳар куни миллионлаб, миллиардлаб одамлар фойдаланади. Интернет орқали маълумот узатиш жуда қулай ва самарили эканлигини ҳисобга олсак маҳсулотларни интернет орқали харид қилиш жуда фойдали ҳисобланади. Чунки сиз ўзингизга керак бўлган товарни бозорга бормасдан, уйиғизда ёки офисингизда ўтириб харид қилишингиз мумкин бўлади. Бу орқали эса ўз вақтингизни тежаб қолишга эришасиз. Албатта бу соҳани йўлга қўйиш унчалик ҳам осон эмас аммо биз ёшлар бунга астойдил ҳаракат қилсак албатта эришамиз. Зеро юртимизда бунга етарлича имкониятлар етарли.

Битирув малакавий ишни бажариш билан мен келгусида мураккаб дастурий маҳсулотлар яратишга ҳаракат қилиш мумкинлигини билдим. Келажакда электрон тижорат соҳаси янада ривожланиши ва бу орқали инсонларнинг юмушлари янада енгиллашишига ишонаман.

## VI. Фойдаланилган адабиётлар рўйхати

1. “Мамлакатимизни модернизация ва иқтисодиётимизни барқарор ривожлантириш йўлида” 2008й.
2. “Ватанимизнинг босқичма-босқич ва барқарор ривожланишини таъминлаш – бизнинг олий мақсадимиз” 2009й.
3. Ларри Ульман. Основы программирования на PHP. Самоучитель. ДМК пресс. 2001
4. “ВБ НЕТ Сомплете”. БПБ Публикасион, НьюДелхи-2002 й.
5. Матросов, Сергев, Чаунин. HTML 4.0 в подлиннике. БХВ-СПб, 2000
6. Вайнам Л, Вайнам В, Динамический HTML. Диасофт 2001
7. Уилтон П. Жава Скрипт. Основы. Символ-плюс. 2002
8. «Освой самостоятельно PHP4». Пер.с англ. – М.:Издательский дом «Вильямс», 2001.
9. Энциклопедия. Кибернетика.1-2 т. Киев 1974
10. Н.Е.Кобринский Бведение в экономическую кибернетику. М. 1975.
11. <http://www.ziyonet.uz>
12. <http://www.php.ru>
13. <http://yiiframework.ru>

## VII. ИЛОБА.

```
<?php
$ref = $_SERVER['HTTP_USER_AGENT'];
if (!preg_match ("/yandex/i",$ref)) session_start();
if (is_dir('./install/')) {header("Location: ./install/"); exit;}
function get_mtime() {
    $mtime = microtime();
    $mtime = explode(' ', $mtime);
    $mtime = $mtime[1] + $mtime[0];
    return $mtime;
}
function convert($size) {
    $unit = array('b', 'kb', 'mb', 'gb', 'tb', 'pb');
    return round($size / pow(1024, ($i = floor(log($size, 1024)))), 2) . $unit[$i];;
}
function mamory() {
    $memory = (!function_exists('memory_get_usage')) ? " :
round(memory_get_usage() / 1024 / 1024, 4) . 'MB';
    echo "<div>" . $memory . "<div>";
}
$time1 = get_mtime();
//core file
ini_set("display_errors", "1");
ini_set("magic_quotes_runtime", "0");
//include core files
define('ROOT_DIR', dirname(__FILE__));
define('INCLUDE_DIR', ROOT_DIR . '/includes/');
include_once ("./cfg/connect.inc.php");
include_once ("./includes/database/mysql.php");
```

```

include_once (".cfg/general.inc.php");
include_once (".cfg/functions.php");
include_once (".core/core_errors.php");
include_once (".cfg/appearance.inc.php");
include_once (".cfg/category_functions.php");
include_once (".cfg/language_list.php");
include_once (".cfg/company.inc.php");
include_once (".cfg/product.inc.php");
include_once (".cfg/shipping.inc.php");
include_once (".cfg/votes.inc.php");
include_once (".cfg/redirect.inc.php");
//init Smarty
if (DB_CHARSET == 'cp1251') define('SMARTY_RESOURCE_CHAR_SET',
DB_CHARSET);
include ROOT_DIR . '/smarty/smarty.class.php';
$smarty = new Smarty; //core smarty object
$smarty_mail = new Smarty; //for e-mails
$smarty->register_modifier("print_price", "print_price");
//set Smarty include files dir
$smarty->template_dir = ROOT_DIR . "/css/css_" . CONF_COLOR_SCHEME .
"/theme";
$smarty_mail->template_dir = ".css/css_" . CONF_COLOR_SCHEME .
"/theme/mail";
//select a new language?
if (isset($_GET["new_language"]) && $_SESSION["current_language"] !=
$_GET["new_language"]) {
    $_SESSION["current_language"] = $_GET["new_language"];
    header("Location: " . $_SERVER["HTTP_REFERER"]);
}
//current language session variable

```

```

if (!isset($_SESSION["current_language"]) || $_SESSION["current_language"] < 0
|| $_SESSION["current_language"] > count($lang_list))
$_SESSION["current_language"] = 0; //set default language
//include a language file
if (isset($lang_list[$_SESSION["current_language"]]) && file_exists(ROOT_DIR
."/languages/" . $lang_list[$_SESSION["current_language"]]->filename)) include
(ROOT_DIR . "/languages/" . $lang_list[$_SESSION["current_language"]]-
>filename); //include current language file
else {
    die("<font color=red><b>ERROR: Couldn't find language file!</b></font>");
}
//connect to the database
db_connect(DB_HOST, DB_USER, DB_PASS) or die(db_error());
db_select_db(DB_NAME) or die(db_error());
mysql_query('SET NAMES ' . DB_CHARSET);
currency();
$currents = db_arAll("select * from " . CURRENCY_TABLE);
$smarty->assign("current_all", $currents);
unset($currents);
//get currency ISO 3 code
$currency_iso_3 = (defined('CONF_CURRENCY_ISO3')) ?
CONF_CURRENCY_ISO3 : "USD";
$smarty->assign("currency_iso_3", $currency_iso_3);
unset($currents, $currency_iso_3);
//hurl's
#$esql=1;
if (isset($_GET["categoryID"]) && CONF_CHPU && db_r('select hurl from
'.CATEGORIES_TABLE.' where categoryID='.(int)$_GET["categoryID"]'))
    include_once('./core/core_404.php');

```

```

if (isset($_GET["productID"]) && CONF_CHPU && db_r('select hurl from
'.PRODUCTS_TABLE.' where productID=).(int)$_GET["productID"]))
    include_once('./core/core_404.php');

if (isset($_GET["catalog_hurl"])) {
    $row = db_r("SELECT categoryID FROM " . CATEGORIES_TABLE . " where
enabled=1 and hurl=" . mysql_real_escape_string($_GET["catalog_hurl"]) . "");
    if ($row && CONF_CHPU) $_POST["categoryID"] = $row;
    else include_once('./core/core_404.php');
} elseif (isset($_GET["tags_hurl"])) {
    $tagID = db_r('SELECT tag FROM ' . TAGS_TABLE . ' WHERE
hurl='.int_text($_GET['tags_hurl']). ' OR tag='.int_text($_GET['tags_hurl']));
    if ($tagID && CONF_CHPU) {
        $_POST["tagID"] = $tagID;
    } else include_once('./core/core_404.php');
} elseif (isset($_GET["product_hurl"])) {
    $row = db_r("SELECT productID FROM " . PRODUCTS_TABLE . " where
enabled=1 and hurl=".int_text($_GET['product_hurl']));
    if ($row && CONF_CHPU) $productID = $row;
    else include_once('./core/core_404.php');
}
//set $categoryID
if (isset($_GET["categoryID"]) || isset($_POST["categoryID"]))
{
    $categoryID = isset($_GET['categoryID']) ? (int)$_GET['categoryID'] :
(int)$_POST['categoryID'];
}
else $categoryID = 0;
$categoryID = (int)$categoryID;
//$productID

```

```

if (isset($_GET['productID']) || isset($_POST['productID'])) $productID =
isset($_GET['productID']) ? (int)$_GET['productID'] : (int)$_POST['productID'];
//and different vars...
if (isset($_GET["register"]) || isset($_POST["register"])) $register =
isset($_GET["register"]) ? $_GET["register"] : $_POST["register"];
if (isset($_GET["update_details"]) || isset($_POST["update_details"]))
$update_details = isset($_GET["update_details"]) ? $_GET["update_details"] :
$_POST["update_details"];
if (isset($_GET["order"]) || isset($_POST["order"])) $order =
isset($_GET["order"]) ? $_GET["order"] : $_POST["order"];
if (isset($_GET["check_order"]) || isset($_POST["check_order"])) $check_order =
isset($_GET["check_order"]) ? $_GET["check_order"] : $_POST["check_order"];
if (isset($_GET["proceed_ordering"]) || isset($_POST["proceed_ordering"]))
$proceed_ordering = isset($_GET["proceed_ordering"]) ?
$_GET["proceed_ordering"] : $_POST["proceed_ordering"];
if (!isset($_SESSION["vote_completed"])) $_SESSION["vote_completed"] =
array();
//checking for proper $offset init
if (isset($_GET["offset"]) && !is_numeric($_GET["offset"]))
include_once('./core/core_404.php');
$offset = isset($_GET["offset"]) ? $_GET["offset"] : 0;
if ($offset < 0 || $offset % CONF_PRODUCTS_PER_PAGE) $offset = 0;
// sort order
if (isset($_GET["sort"])) {
    switch ($_GET["sort"]) {
        case "name":
            $_SESSION["sort"] = "name";
            break;
        case "Price":
            $_SESSION["sort"] = "Price";

```

```

break;
case "in_stock":
    $_SESSION["sort"] = "in_stock";
break;
case "product_code":
    $_SESSION["sort"] = "product_code";
break;
case "customers_rating":
    $_SESSION["sort"] = "customers_rating";
break;
default:
    $_SESSION["sort"] = CONF_SORT_PRODUCT;
break;
}
$_GET["order"] = strtoupper($_GET["order"]);
switch ($_GET["order"]) {
    case "ASC":
        $_SESSION["order"] = "ASC";
break;
    case "DESC":
        $_SESSION["order"] = "DESC";
break;
    default:
        $_SESSION["order"] = CONF_SORT_PRODUCT_BY;
break;
}
} else {
    if (!isset($_SESSION["sort"])) $_SESSION["sort"] =
CONF_SORT_PRODUCT;

```

```

    if (!isset($_SESSION["order"])) $_SESSION["order"] =
CONF_SORT_PRODUCT_BY;
}
// $smarty -> assign("sort", $_SESSION["sort"]);
// $smarty -> assign("order", $_SESSION["order"]);
define("CURR_USD", 1);
define("CURR_EUR", 1);
if (CONF_CURRENCY_AUTO == 1) {
    if (getAddrByHost("www.cbr.ru", 1))
getCURRENCY('http://www.cbr.ru/scripts/XML_daily.asp');
}
$smarty->assign("product_category_path", "");
// -----SET SMARTY VARS AND INCLUDE SOURCE FILES-----//
if (isset($productID)) { //to rollout categories navigation table

    $categoryID = db_r('SELECT categoryID FROM ' . PRODUCTS_TABLE . '
WHERE productID=' . $productID);
    if (!$categoryID) include_once('./core/core_404.php');
}
//set Smarty main page
$f_cnt = file('./core/aux_pages/live_counts');
$out_cnt = implode("", $f_cnt);
$smarty->assign("live_counts", $out_cnt);
//assign core Smarty variables
$smarty->assign("lang_list", $lang_list);
$smarty->assign("lang_list_count", count($lang_list));
if (isset($_SESSION["current_language"])) $smarty->assign("current_language",
$_SESSION["current_language"]);
// - following vars are used as hidden in the customer survey form
$smarty->assign("categoryID", $categoryID);

```

```

if (isset($productID)) $smarty->assign("productID", $productID);
if (isset($_GET["currency"])) $smarty->assign("currency", $_GET["currency"]);
if (isset($_GET["user_details"])) $smarty->assign("user_details",
$_GET["user_details"]);
if (isset($_GET["aux_page"])) $smarty->assign("aux_page",
$_GET["aux_page"]);
if (isset($_GET["show_price"])) $smarty->assign("show_price",
$_GET["show_price"]);
if (isset($_GET["adv_search"])) $smarty->assign("adv_search",
$_GET["adv_search"]);
if (isset($_GET["searchstring"])) $smarty->assign("searchstring",
$_GET["searchstring"]);
if (isset($register)) $smarty->assign("register", $register);
if (isset($order)) $smarty->assign("order", $order);
if (isset($check_order)) $smarty->assign("check_order", $check_order);
//set default main_content template to homepage
$smarty->assign("main_content_template", "home.tpl.html");
if (isset($_POST['token'])) {
    $url = 'http://ulogin.ru/token.php?token=' . $_POST['token'] . '&host=' .
CONF_SHOP_URL;
    $authData = legosp_http_request($url);
    $authData = json_decode($authData, true);
    $em = "";
    if (isset($authData['email']) && trim($authData['email'])) $em = ' or
cust_email=\" . $authData['email'] . "\";
    if (DB_CHARSET == 'cp1251') {
        if (isset($authData['first_name'])) $authData['first_name'] =
Utf8Win($authData['first_name']);
        else $authData['first_name'] = "";
    }
}

```

```

        if (isset($authData['last_name'])) $authData['last_name'] =
Utf8Win($authData['last_name']);
        else $authData['last_name'] = "";
    }
    $custID = db_r('select count(*) FROM ' . CUST_TABLE . ' WHERE
(provider=\'\" . $authData['network'] . \'\" and openID=\'\" . $authData['uid'] . \'\"')\" .
$em);
}
// includes all .php files from includes/ dir
foreach(glob(INCLUDE_DIR . \"*.php\") as $php_file) include ($php_file);
$includes_dir = './css/css_' . CONF_COLOR_SCHEME . '/includes/';
if (is_dir($includes_dir)) foreach(glob($includes_dir . \"*.php\") as $php_file)
include ($php_file);
if (isset($_GET) && !empty($_GET) && $smarty-
>tpl_vars[main_content_template]->value===\'home.tpl.html\')
{
    include_once('./core/core_404.php');
}
if (isset($_SERVER[\"HTTP_REFERER\"]) &&
!strpos($_SERVER[\"HTTP_REFERER\"], 'cart')) $_SESSION[\"go_back\"] =
$_SERVER[\"HTTP_REFERER\"];
if (isset($_SESSION[\'go_back\'])) $go_back = $_SESSION[\'go_back\'];
else $go_back = \"\";
$smarty->assign(\"go_back\", $go_back);
if (isset($_GET[\'debug\'])) $smarty->debugging = true;
$smarty->display(\"./css/css_\" . CONF_COLOR_SCHEME .
\"/theme/index.tpl.html\");
?>

```