

**O'ZBEKISTON ALOQA VA AXBOROTLASHTIRISH AGENTLIGI  
TOSHKENT AXBOROT TEXNOLOGIYALARI UNIVERSITETI**

Himoyaga  
Kafedra mudiri

«\_\_\_\_\_»

«\_\_»\_\_\_\_\_2012 y.

**BAKALAVR BITIRUV MALAKAVIY  
ISHI**

Mavzu: Dasturlash texnologiyalari fani uchun elektron resurs yaratish

Bitiruvchi \_\_\_\_\_ Arislonov B.  
(imzo) (f.i.o.)

Rahbar \_\_\_\_\_ Reshetov E  
(imzo) (f.i.o.)

Taqrizchi \_\_\_\_\_ Qudratova D.Q.  
(imzo) (f.i.o.)

MM va TX \_\_\_\_\_ Ishmuhammedova V.I.  
(imzo) (f.i.o.)

**Toshkent – 2012**

Ushbu bitiruv malakaviy ishi dasturlash texnologiyasi fanini o'rgatuvchi elektron o'quv qo'llanma yaratilishiga bag'ishlangan. Ushbu ishning e'tiborli jihati shundaki, unda foydalanuvchilarning mana shu o'quv qo'llanmadan olgan bilimlarini tekshirish maqsadida tashkil etilgan test oyinasi mavjud. Yaratilgan elektron o'quv qo'llanma asosan web dasturlash sohasiga qiziquvchi yosh mutaxassislariga mo'ljallangan.

Bundan tashqari mehnat muhofazasi va texnika xavfsizligi bo'yicha umumiy masalalar ko'rib chiqilgan.

Данная выпускная квалификационная работа посвящена разработке технология для программирования, которая строена на основе PHP технологии. Особое внимание обращено к странице TEST, которая дает возможность проверять полученные знания. Данная работа полезна всем кто интересуется в PHP.

Также рассматриваются вопросы охраны труда и техники безопасности.

This Bachelor's final qualification work is dedicated to the development of the Web application for educational purpose which is built on the base of PHP. The main attention is made to the TEST page of the project for giving the chance to students to check their knowledge. Due to this, this application is suitable for everybody who wants to enter to the world of PHP.

Also the problems of labor safety and safety precautions are considered.

## MUNDARIJA

Kirish .....	
I . Web texnologiyalari va ularning hozirgi zamon AT muhitidagi o' rni va istiqbollari .....	
1.1 World Wide Web .....	
1.2 O' quv qo' llanmani yaratishda PHP web dasturlash tilidan foydalanish...	
1.2 PHP dasturlash tilining qisqacha tarixi web saytlar yaratishdagi o' rni...	
II . PHP web dasturlash tili, uning ma' lumotlar bazasi va web serverlar bilan bog' liqligi .....	
2.1 PHP web dasturlash tili haqida umumiy ma' lumotlar. Undagi funktsiya va ob' yektlar .....	
2.2. Ma' lumotlar bazasi va uning asosiy tushunchalari .....	
2.3 Serverlar haqida tushuncha. Apache serveri .....	
III . Dastur interfeysi va foydalanuvchi yo' riqnomasi .....	
3.1 Dasturiy tizimga qo' yiladigan talablar .....	
3.2 Dasturiy tizimdan foydalanish qo' llanmasini yaratish.....	
IV . Mehnat muhofazasi va texnika xavfsizligi .....	
4.1 Shaxsiy kompyuterda ishlashda texnika xavfsizligiga rioya qilish .....	
4.2 Elektr xavfsizligi shartlari tahlili .....	
Xulosa .....	
Foydalanilgan adabiyotlar.....	
Ilova.....	

## KIRISH

Mamlakatimiz prezidenti Islom Abdugʻanievich Karimovning “Jahon moliyaviy-iqtisodiy inqirozi, Oʻzbekiston sharoitida uni bartaraf etishning yoʻllari va choralari” asarida soʻnggi toʻrt yil mobaynida yiliga oʻrtacha 50 foiz oʻsishni taʼminlayotgan axborot-kommunikatsiya texnologiyalari sohasida hizmat koʻrsatish jadal rivojlanayotganligi alohida taʼkidlab oʻtilgan. Mazkur asarda prezidentimiz tomonidan inqirozga qarshi hal etilishi zarur boʻlgan masalalar belgilab berilgan boʻlib, unga koʻra korxonalarni modernizatsiya qilish, texnik va texnologik qayta jihozlashni yanada jadallashtirish, zamonaviy, moslashuvchan texnologiyalarni keng joriy etish inqirozga qarshi chora tadbirlarning eng asosiysi ekanligi koʻrsatib oʻtilgan.

Tarixga nazar tashlasak, texnika va texnologiya taraqqiyotining rivojlangan bosqichi XX asrga toʻgʻri keladi. Ushbu asrning “texnika” asri deb atalishining asosiy sabablaridan birini mana shu deb keltirsak yanglishmaymiz. Albatta, har bir texnika vositasi yaratilishiga inson ongi, tafakkuri, toʻplagan bilimlari va mehnati natijasi evaziga erishiladi. Bilimlar esa oʻz navbatida oʻqish, izlanish va tajriba asosida toʻplanadi. Bilim olishning asosiy manbai kitob hisoblansada, lekin hozirgi kunda internetning shiddat bilan rivojlanib, xonadonlarga kirib borishi uning oʻz - oʻzidan bilim olishni yana bir asosiy manbayiga aylanib bormoqda.

Insoniyat qadimdan oʻzi egallagan bilim va tajribalarini kelajak avlodlar uchun meros qilib qoldirishga harakat qilgan. Bu harakatlar turli xil koʻrinishlarda namoyon boʻlgan. Dastlab gʻor devorlariga turli xil shakl - belgilar orqali, tosh bitiklarga, hayvonlar terisiga, daraxt poʻstlogʻlariga, taxta va shu kabi narsalarga qoldirishgan.

Qog'oz paydo bo'lgandan keyin qog'ozga yozilib silindr idishlarda va undan kitoblar shaklida saqlanib qolgan. Bugungi kunga kelib esa axborot texnologiyalari rivojlangan bir paytda turli xil videokursli va web sayt ko'rinishdagi o'quv qo'llanmalar paydo bo'ldi.

Hozirgi paytda O'zbekistonda web dasturlash sohasiga bo'lgan talab o'sib bormoqda. Har bir korxonada, tashkilot, o'quv muassasalari, taniqli shaxslar o'z shaxsiy web saytlariga ega bo'lmoqdalar. Web dasturchilar bunday saytlarni yaratishda web dasturlash tillariga murojaat qilmoqdalar. Bunday holda ayniqsa o'zbek tilidagi web dasturlash tillarini o'rgatuvchi o'quv qo'llanmalarga ehtiyoj sezilyapti. Ushbu bitiruv malakaviy ishi ham xuddi shu kabi dolzarb masalalarni yechishga qaratilgan bo'lib, unda PHP web dasturlash tilini o'rgatuvchi elektron o'quv qo'llanma yaratish g'oyasi ilgari surilgan.

Ushbu elektron o'quv qo'llanmani yaratishdan maqsad o'rta-maxsus va oliy ta'lim o'quv muassasalari talabalari hamda shu sohaga qiziquvchilarga PHP web dasturlash tili haqida boshlang'ich bilimlar olishga yordamlashish. Bundan tashqari MySQL ma'lumotlar bazasi haqida tushunchalar hosil qilish va uni PHP dasturlash tili bilan bog'lashni o'rgatishdan iborat.

Yuqoridagi maqsadlarga erishish uchun quyidagi masalalarni yechish lozim:

- Internet, uning hozirgi zamon AT muhitidagi o'rni va uning asosiy tushunchalarini o'rganish;
- web dasturlash tillarining asosi bo'lgan HTML gipermatnli tili haqida tushunchaga ega bo'lish;
- PHP tilining imkoniyatlarini o'rganib chiqish;
- Ma'lumotlar bazasi bilan ishlash va uni boshqarish;

- Elektron o'quv qo'llanmadan foydalanish yo'riqnomasini yaratish.

Bitiruv malakaviy ishi kirish, to'rt bob, xulosa, foydalanilgan adabiyotlar va ilovadan iborat.

Birinchi bobda internet va web dasturlash tillarining hozirgi zamon AT muhitidagi o'rni, PHP web dasturlash tilining kelib chiqishi tarixi va uning afzalliklari haqida fikr yuritilgan.

Ikkinchi bobda PHP dasturlash tili va MySQL ma'lumotlar bazasi haqida umumiy ma'lumotlar berilgan bo'lib, ularning o'zaro bog'liklari to'g'risida gapirib o'tilgan. Dastur ishlashi uchun kerak bo'ladigan apache serverning PHP dasturlash tili bilan bog'liqligi to'g'risida ham to'xtalib o'tilgan.

Uchinchi bobda dasturiy tizimga qo'yiladigan talablar, dastur interfeysi va undan foydalanish yo'riqnomasi ishlab chiqilgan.

To'rtinchi bob texnika xavfsizligiga bag'ishlangan bo'lib, unda shaxsiy kompyuterda ishlashda texnika xavfsizligiga rioya qilish va elektr xavfsizligi shartlari tahlili to'g'risida ma'lumotlar berilgan.

Xulosa qismida ish davomida olingan asosiy masalalarning yechimi va bajarilgan ish haqida qisqacha ma'lumot berilgan.

## I . Web texnologiyalari va ularning hozirgi zamon AT muhitidagi o'рни va istiqbollari

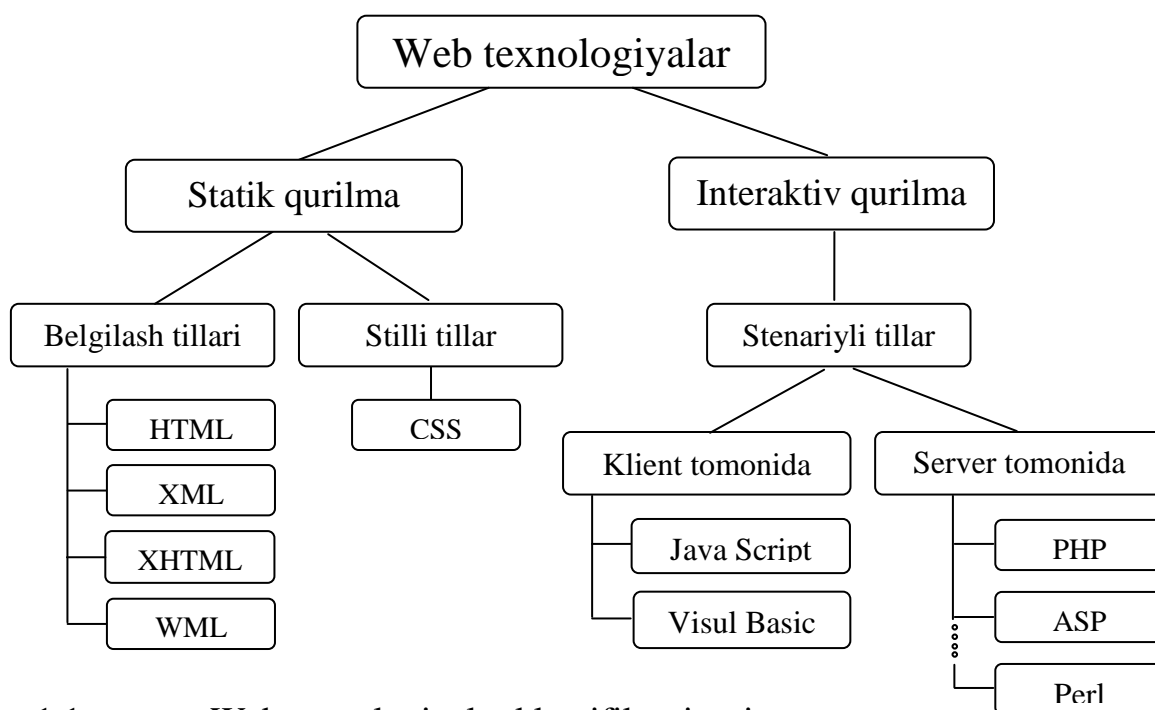
Bugungi kunda axborot texnologiyalari rivojlanishini web texnologiyalarisiz tasavvur qilib bo'lmaydi. Web texnologiyalari deganda - web ilovalar hosil qilish, web saytlar tashkil qilish, tarmoq va texnologiyalar jamlanmasi tushuniladi. Web texnologiyalarida quyidagi 3 ta asosiy tushuncha mavjud:

- Web sahifa
- Web sayt
- Web server

Web sahifa o'zining unikal adresiga ega bo'lgan va maxsus brauzer dasturlar orqali ochiladigan va ko'riladigan elektron hujjatdir.

Web sayt bir nechta web sahifalarning mantiqiy va funksional jihatdan birlashtirilgan sahifalar to'plami.

Web server tarmoqqa ulangan kompyuter hisoblanib umumiy resurs va dasturlarni boshqaradi. Web sahifa va web saytlar bilan ishlash uchun eng keng tarqalgan texnologiya WWW texnologiyasidir. Umumiy holda web texnologiyalarni quyidagi sxema ko'rinishida tasvirlash mumkin.



1.1 – rasm. Web texnologiyalar klassifikatsiyasi.

## 1.1. World Wide Web

### 1.1.1 Internet texnologiyalari

Internetning paydo bo'lishi tarixi 60-yillarning oxirida Amerika hukumati tomonidan asos solingan ARPANet (Advanced Research Projecj Agenci tashkiloti) hisoblash tarmog'iga borib taqaladi. Tarmoq harbiy tashkilotlarga xizmat qilgan. 1980 yillar boshlarida ma'lumotlarni uzatishni boshqarish protokoli TCP/IP (Transmissyon Control Protocol / Internet Protocol) ga asos solindi. Taxminan shu vaqtda ma'lum bo'ldiki, TCP/IP dan turli milliy va xalqaro darajadagi kompyuter tarmoqlarini bog'lashda foydalanish mumkin ekan.

1989 yilning oxirida ARPANet mukammal holga yetib keldi, lekin bu vaqtga kelib ko'pgina universitetlar va ilmiy muassasalar Internetga ulangan edilar. 1990 yillar boshlarida korporatsiyalar ham Internetdan elektron pochta orqali ma'lumotlar almashishda aktiv ishtirok etardilar. U vaqtlarda Milliy Ilmiy fond tijorat maqsadida Internetdan foydalanishni ta'qiqlagan edi. 1991 yilda bu cheklash bekor qilinadi va Internetdan tashkilot, muassasa, nodavlat tashkilotlarining foydalanish darajasi ortdi, shuningdek, tijorat maqsadida Internetdan keng foydalanila boshlandi (Internet magazinlar, Internet reklamalar va h.k.).

WWW (World Wide Web) – butun jahon tarmog'i. Ayni mana shu xizmat Internet bilan foydalanishni soddalashtirdi va ommaviylashtirdi. WWW asosida to'rtta poydevor mavjud:

1. Barcha hujjatlarning yagona formati (shakli)
2. Gipermatn
3. Hujjatlarni ko'rish uchun maxsus dasturlar
4. Yagona manzilni ko'rsatish tizimi

Hujjatlarning yagona formati (shakli)

Internetda barcha hujjatlar HTML (Hyper Text Markup Language – Gipermatnni belgilash tili) formatida tuziladi. HTML yordamida yozilgan hujjatlarni web-hujjat yoki web-sahifa deb atashadi. Birorta foydalanuvchi yoki tashkilotga qarashli bo'lgan, bir-biriga bog'langan web-sahifalar to'plami SAYT deb aytiladi.

Aslida web-sahifa oddiy matndan iborat va uning fayl kengaytmasi “\*.htm” yoki “\*.html” bo`ladi. Matnning o`lchamini, rangini, shaklini o`zgartirish uchun HTML teg (belgi)lari qo`llaniladi. HTML teglari yordamida sahifaga rasm, video o`rnatish, orqa fon (ko`rinish), musiqaviy ijro va yana ko`p narsalar qo`shish mumkin. Natijada, oddiy matn chiroyli va qulay ravishda taqdim etiladi.

### Gipermatn

WWW xizmatining eng inqilobiy qismi gipermatn orqali umuman klaviaturaga tegmasdan, faqat sichqonchanning chap tugmasini chertish yordamida Internetda sayohat qilish mumkin. Gipermatnning asosiy elementi giperko`rsatkich – tarmoqdagi boshqa hujjatga ko`rsatkich bo`lib, ular ikki xil: matn va grafik. Ko`rsatkichni aniqlash uchun web-sahifa ko`rilyatganda sichqoncha belgisi “qo`l” ko`rinishiga o`zgarsa, demak sichqonchanning tagidagi element giperko`rsatkichdir. Matn ko`rsatkichlar, odatda, oddiy matnlardan ajralib turishi uchun, tagida chiziqchali va boshqa rangda bo`yaladi.

### Grafik ko`rsatkich

Giperko`rsatkichga chertsangiz, huddi sehrli ravishda boshqa sahifaga o`tasiz. Hech qanday sahifaning manzilini, hech qayerga kiritish shart emas, chunki giperko`rsatkich barcha kerakli ma`lumotga ega.

### Brauzer web-sahifalarni ko`rish dasturi

Brauzer web-sahifada HTML teglarini topib, ular talabi bo`yicha ma`lumotni ekranga chiqaradi. Teglar, o`zlari, ekranda aks ettirilmaydi.

1993 yilda birinchi web-brauzer Mosaic paydo bo`ldi. Bugungi kunda yuz xil brauzerlar mavjud. Eng mashhurlari: Internet Explorer (Windows tarkibidagi dastur), Opera, FireFox.

## 1.1.2 HTTP va HTML

Gipermatn jo'natish protokoli (HTTP) web-brauzer va web-server muloqotining asosini tashkil etadi. Web-brauzer HTTP maxsus komandasi orqali server bilan bog'lanish uchun so'rov jo'natadi. Agar so'rov qoniqtirilsa, brauzer server bilan bog'lanadi. Qabul qiluvchi tomon nima qilishni o'zi xal qiladi yoki ekranda faylni ko'rsatadi, yoki uni diskda saqlab qo'yadi va h.k. HTTP Internetda foydalaniladigan yagona protokol emas. Fayllar uzatish protokoli (Post Office Protocole – POP, Simple Mayl Transport Protocole - SMTP), shifrlangan HTTP protokoli (SHTTP) kabi protokollar mavjud.

HTML so'zi qisqartma so'z bo'lib uning kengaytmasi Hyper Text Markup Language - ya'ni gipermatnli belgilash tili degan ma'no beradi. HTML da sayt yaratishda asosiy qoidalarga albatta e'tibor berish kerak. Sahifalar 2 qismga bo'linadi. Bosh qismi (HEAD) va tana qismi (BODY).

HEAD qismida siz ishlayotgan sahifaning SARLAVHAsi, uning METAsi, STYLElariga oid teglar joylashtiriladi.

BODY qismida esa asosiy ma'lumotlar, ya'ni siz sayt orqali ko'rsatib bermoqchi bo'lgan ma'lumotlar, rasmlar, musiqiy fayllar va shu kabi fayllarni Internet Explorer sahifasida namoyish etish uchun beriladigan buyruq teglari joylashadi.

Gipermatnlar tili (HTML) web-brauzerlar ekraniga ma'lumotlarni standart kodlar orqali chop etish imkoniyatini beradi. HTML dan foydalanib o'zida taxrirlangan matn, tasvir, multimedia elementlarini aks ettirgan web-sahifalarni yaratish mumkin. HTML ni dasturlash tillaridan farqlash lozim. HTML ni web-brauzerlarga matn yoki tasvirlarni qay holatda chop etish ko'rsatmalar to'plami deyish ham mumkin. Masalan HTML hujjat o'zida quyidagi matnni aks ettirgan bo'lsin:

```
<h1><b> Mening sahifamga xush kelibsiz </b></h1>
```

Bu matndagi <h1> va </h1> teglari asosiy matnni o'rtaga olib turibdi. Web-brauzer tushunadiki, bu teglar o'rtasidagi matn 1-darajali kattalikdagi sarlavha

holatida ekranga sop etilish kerak. <b> va </b> teglari esa, ular o'rtasidagi matn qalinlashtirilgan holda yozilishi kerakligini bildiradi.

Butunjahon o'rgimchak to'rining asosiy va HTML ning tarkibiy qismini gipermatnlar va gipermurojatlar tashkil etadi. Maxsus komandalar yordamida matnning ma'lum qismi shunday ajratiladiki, natijada o'sha matn ustiga sichqoncha tugmasi bosilsa boshqa matn yoki sahifa ochiladi. Bundan tashqari multimedia vositalarining ishlab ketishi yoki bo'lmasa, ma'lumotni diskda saqlash taklifi ham berilishi mumkin. Gipermatn yoki gipermurojat biror bir tasvirga ham qo'yilishi mumkinki uning ustiga bosilganda ham yuqorida aytilgan holatlar ro'y berishi mumkin. Har bir web-sahifa o'zida bir nechta gipermatn yoki gipermurojatlarni mujassam etishi mumkin. FTP protokoli kompyuterlar o'rtasida fayllar almashishda qo'llaniladi. FTP murojaatlarini hosil qilish uchun kerakli server nomini bilish kifoya:

```
<a href="ftp://ftp.microsoft.com/">Microsoft
```

```
kompaniyasining FTP sayti</a>
```

```
<a href="ftp://ftp.jspi.uz/downloads/program.zip"> Zip langan  
programma</a>
```

Ikkinchi misoldan ko'rinib turibdiki, kerakli programmani birato'la skachat qilib olish mumkin. Bunda brauzer olinayotgan dasturni qattiq diskka haqiqatdan ham saqlash kerak yoki kerakmasligini foydalanuvchidan so'raydi.

## 1.2. O'quv qo'llanmani yaratishda PHP web dasturlash tilidan foydalanish

### 1.2.1 PHP dasturlash tilining qisqacha tarixi

1994 yili php tilining yaratuvchisi Rasmus Lerdorf o'zining saytiga mehmonlar kirishini hisoblash uchun Perl dasturlash tilida maxsus qobiq yozib amalda qo'llagan. Ko'p o'tmay qobiqni ishlash unumdorligi juda past va sekinligi aniqlanganidan so'ng, dasturlarni yangidan "C" tilida yozib chiqishga to'g'ri keladi. Keyin, dastlabki dastur kodlari muallif tarafidan barchaga ko'rish uchun serverga nashr qilingan. Server foydalanuvchilari kodlar bilan qiziqib, uni ishlatish muxlislari ham paydo bo'lgan. Ko'p o'tmay, bu dasturlar alohida loyihaga aylanib, 1995 yilning iyul oyida dasturiy mahsulot PHP (Personal Home Page) nomi bilan birinchi nashri chiqarildi. Imkoniyatlari oddiy va sodda bo'lib: bir necha buyruqlarni tushunadigan kod analizatori (tekshiruvchisi), mehmonlar hisoblagichini, kitobini, chatini yaratish uchun foydali dasturlar to'plamidan iborat bo'lgan.

1996 yil aprel oyida dasturlar jiddiy qayta ishlanganidan so'ng, PHP/FI (Personal Home Page / Forms Interpreter) nomi bilan mahsulotning ikkinchi nashri paydo bo'ldi. Bu mahsulot html-kod ichiga yozilib, html-formalarni qayta ishlab, hozirgi PHP dasturlash tilining tayanch imkoniyatlarini ichiga olgan. PHP/FI kod yozilishi Perl tiliga juda o'xshagan, lekin soddaroq bo'lgan.

1997 yili PHP/FI 2.0 nashri chiqdi. O'sha paytda bu mahsulot bilan dunyo bo'yicha bir necha ming odam foydalanib, taxminan 50 ming domen bo'lib, Internetning 1% ini tashkil qildi.

1997 yilda Endi Gutmans va Ziv Suraski PHP/FI kodini boshqatdan yozib chiqishdi, chunki eski kod ular ishlatayotgan elektron tijorat tizimlari uchun yaroqsiz edi. Eski kodning mualliflaridan yordam olish uchun ular birlashishni taklif etib, PHP3 nomli loyihani PHP/FI -ni rasmiy vorisi deb e'lon qilishdi. Yangi loyiha uyushgandan keyin PHP/FI loyihasi ishlab chiqarilishi to'xtatilgan.

PHP 3.0 ning eng kuchli taraflaridan biri uning kengaytirala olinadigan yadrosi (tizimning boshqaruv qismi) bo'lib, bundan tashqari, ma'lumot jang'armalar bilan, turli protokollar va interfeyslar bilan birgalikda ishlash keng imkoniyatlari yaratildi. Muvaffaqiyatga erishishga ancha ahamiyatli fakt bu yangi

tilni boyligi va ob'ektlarga mo'ljallangan dasturlashni qo'llay olishi. Yangi loyiha bilan birga nafaqat tilni tashqi, ichki tuzulishi o'zgardi, balki o'zini nomi ham. Endi PHP qisqartmasi "PHP: Hypertext Preprocessor" ma'nosini anglatishi bildirildi.

1998 yilning oxirida PHP foydalanuvchilarning soni o'n minglardan oshdi. Yuz mingdan oshiq veb-saytlar bu tilni qo'llashini e'lon qilishdi. Taxminan Internetning 10% serverlarida PHP 3.0 o'rnatilgan edi.

1998 yilning iyun oyda PHP3 to'qqiz oy ommaviy tekshiruvidan keyin rasman e'lon qilindi. Shu yilning qishida Endi Gutmans va Ziv Suraski PHP yadrosini qaytadan ishlab chiqarishni boshlashgan. Ularning asosiy vazifasi PHP tiziminig unumdorligini ko'tarish va kodning modullarini yaxshilash edi.

1999 yilning o'rtalarida birinchi marta taqdim qilingan yangi yadro "Zend Engine" deb nomlangan ("Zend": mualliflar "Zeev" va "Andi" ismlardan tashkil topgan). Uni asosida tuzilgan yangi til PHP4 2000 yilning may oyida rasman chiqarilgan. Unumdorlik yaxshilangandan tashqari, PHP 4.0 muhim yangiliklarga ega bo'lib, sessiyalarni qo'llash, buferli chiqarish, kiritilgan ma'lumotlarni havfsiz qayta ishlash va yana bir necha yangi til tuzuvchilarini paydo bo'lishidan iborat.

Hozirgi kunlarda "Zend Engine" qayta yaxshilanib PHP5 tili ishlab chiqarildi. Asosiy o'zgarishlar ob'ektlarga mo'ljallangan dasturlash modelida bo'lib, tilning imkoniyatlari yanada kengaytirdi.

Hozirda PHP tilini yuz minglab dasturchilar ishatib, bir necha million saytlar yaratilib, Internetning 20% domenlarini tashkil etmoqda.

PHP tarixining jadvali

Nashr Sana Izoh

PHP 1.0 1995 yil, 8 iyun Rasmiy nomi "Personal Home Page Tools (PHP Tools)". Bu "PHP" qisqartmani birinchi marta ishlatilinishi.

PHP 2.0 1996 yil, 16 aprel Veb sahifalarni "Tez va oson ishlaydigan qurol" deb e'lon qilingan mahsulot.

PHP 3.0 1998 yil, 6 iyun Bir kishi ishlab chiqaruvchi loyihadan ko'p ishlab chiqaruvchilar loyihasiga aylandi. Endi Gutmans va Ziv Suraski kodni boshqatdan yozib chiqishdi.

PHP 4.0.0 2000 yil, 22 may Yangi yadro "Zend Engine" asosida tuzilgan

PHP 4.1.0 2001 yil, 10 dekabr Til konstruktsiyasiga superglobal imkoniyatlar qoshilgan (\$\_GET, \$\_POST, \$\_SESSION, etc.)

PHP 4.2.0 2002 yil, 22 aprel Xavfsizlikni oshirish maqsadida tizimning sozlash "register\_globals" parametri o'chirilgan.

PHP 4.3.0 2002 yil, 27 dekabr "CGI" (Common Gateway Interface)ga qo'shimcha "CLI" (command line interpreter) texnologiyasi kiritilgan.

PHP 4.4.0 2005 yil, 11 iyul phpize va php-config skriptlari uchun qo'llanma varaqlari qo'shilgan.

PHP 5.0.0 2004 yil, 13 iyul Zend Engine II – yangi ob'ekt modeli kiritilgan.

PHP 5.1.0 2005 yil, 24 noyabr PHP Engine ishlash unumdorligi yaxshilangan.

PHP 5.2.0 2006 yil, 2 noyabr Kengaytiradigan filtr modullarining php tizimida ishlatish imkoniyatlari ochiq holati o'rnatilgan (default).

## 1.2.2 PHP dasturlash tilining afzalliklari va web saytlar yaratishdagi o'рни

PHP tili asosan quyidagi afzalliklarga ega:

1. an'anaviylik
2. oddiylik
3. samaradorlik
4. xavfsizlik
5. moslashuvchanlik

An'anaviylik

PHP tarkibidagi til strukturasi C va Perl ga o'xshashlik xususiyati uning barcha versiyalarida saqlanib qolgan.

Oddiylik

PHP ning kodlarini boshqa belgilash tillariga oson joylashtirish imkoniyati mavjud.

```
<html>
<head>
<title>
<? Print "hello" ?>
</title>
<head>
<body>.....</body>
</html>
```

Samaradorlik

PHPda server ish jarayonini klientlarga taqsimlash yuqori darajada qo'yilgan, bunda klient so'rovlari imkon boricha qondirilishi yo'lga qo'yilgan.

Xavfsizlik

PHP da xavfsizlik 2 ta sathga bo'lingan :

- a) Tizimli xavfsizlik
- b) Taqdimot xavfsizligi

## Moslashuvchanlik

PHP da hosil qilingan stenariylar platformaga moslashuvchan bo'ladi. PHP ga tegishli bo'lgan barcha vositalar internetda to'g'ridan – to'g'ri tarqatiladi.

PHPning vazifasi serverni ma'lumotlar ombori bilan bog'lash (Apache+PHP+MySQL). PHPni o'rganishdan avval HTMLni bilishingiz kerak. PHP ishlashi uchun birinchi navbatda server dasturi o'rnatilgan bo'lishi kerak. Usiz PHP operatsiyalarini bajarib bo'lmaydi. Ular Apache, Denver. Apache'ni o'z kompyuteringizga o'rnatasiz.

PHP operatsiyalari sizga ko'rinmaydi. Siz faqat uning natijasini HTML-kod ko'rinishida olasiz. Dasturda xatolik yuz beradigan bo'lsa, qaysi qatordagiligini ham server ko'rsatadi. PHP kodlarini oddiy bloknotga ham yozsa bo'ladi. Ammo PHP tahrirlovchilarini ishlatgan ma'qulroq. Masalan, men Zend dasturini ishlataman. U PHP koddagi xatoliklarni ko'rsatib turadi.

```
?>
```

Ular orasida php-kodlari yoziladi. Birinchi misol:

```
echo("COMSOFT");
```

```
print("COMSOFT");
```

```
?>
```

Uni \*.php ko'rinishida saqlaymiz (\* - siz beradigan nom). Saqlangan faylni ...WebServers\home\localhost\www papkasiga tashlaymiz. (Bu papkalar Apache serverni o'rnatganingizda chiqadi. ... - Server o'rnatilgan joy). So'ngra brouzerni ochib manzillar paneliga localhost/\*.php deb yozamiz (\* - siz faylga bergan nom). Brouzerda nima yozuv chiqdi? COMSOFTCOMSOFT mi? Shunday bo'lsa hammasi joyida. "echo" va "print" nima ish bajarishini bilib olgan bo'lsangiz kerak. Ikkisi ham yozuvni ekranga chiqaradi. Ikkinchi misol - uni takomillashtiramiz:

```
echo("COMSOFT ");
```

```
print("COMSOFT\nFOREVER");
```

```
?>
```

ESLATMA: Har safar kodni o'zgartirib saqlaganingizdan keyin, brouzerni ham yangilab turing. Buning uchun CTRL+F5 yoki "Obnovit" tugmasini bosing.

echo yoki print ichida HTML kodlarini ishlatish mumkin. Faqat HTML-kodida yoziladigan `<font face="verdana">` PHPda `<font face='verdana'>` yoki `<font face=\"verdana\">` qilib yoziladi. Shunga o'xshash so'zlarni qo'shtirnoq ichida yoziladigani bo'lsa `\` qilib yoziladi. Bo'lmasa xatolik hisoblanadi. Bundan tashqari yozuv ichida `'` belgisini ``` (Yo harfi bilan birga, 1 tugmasi yonida `')` belgisiga almashtirish kerak. Masalan O'quvchi so'zini O`quvchi qilib yoziladi. `\n` belgisi esa probel vazifasini bajaradi. (Huddi HTMLdagi `<br />` kabi).

Izoh qo'shish:

Izoh qo'shish ba'zi holatlarda kod esdan chiqib qoladigan paytda, yoki kod vaqtincha kerak bo'lmagan holatda kerak bo'ladi. Uchinchi misol:

```
//echo("COMSOFT ");  
print("COMSOFT\nFOREVER");  
?>
```

Sizda faqat COMSOFT FOREVER yozuvi chiqadi. Sababi `//` dan keyin yozilgan kodni yo'q deb biladi. Zend dasturini ishlatayotgan bo'lsangiz, u kulrang rangda chiqib qoladi.

```
# belgisini // o'rniga ishlatsak ham bo'ladi. To'rtinchi misol:  
/*echo("COMSOFT ");  
print("COMSOFT\nFOREVER"); */  
?>
```

Hech narsa chiqmaydi. Sababi `/*` o'zidan keyingi barcha kodlarni `*/` gacha izohga qo'shib qo'yadi.

### Xulosa

Ushbu bobda internet va web dasturlash tillarining hozirgi zamon AT muhitidagi o'rnini, PHP web dasturlash tilining kelib chiqishi tarixi va uning afzalliklari haqida fikr yuritildi.

# **I . PHP web dasturlash tili, uning ma'lumotlar bazasi va web serverlar bilan bog'liqligi**

## **Masalaning qo'yilishi**

Ushbu elektron o'quv qo'llanmani yaratishdan maqsad o'rta-maxsus va oliy ta'lim o'quv muassasalari talabalari hamda shu sohaga qiziquvchilarga PHP web dasturlash tili haqida boshlang'ich bilimlar olishga yordamlashish. Bundan tashqari MySQL ma'lumotlar bazasi haqida tushunchalar hosil qilish va uni PHP dasturlash tili bilan bog'lashni o'rgatishdan iborat.

Yuqoridagi maqsadlarga erishish uchun quyidagi masalalarni yechish lozim:

- Internet, uning hozirgi zamon AT muhitidagi o'rni va uning asosiy tushunchalarini o'rganish;
- web dasturlash tillarining asosi bo'lgan HTML gipermatnli tili haqida tushunchaga ega bo'lish;
- PHP tilining imkoniyatlarini o'rganib chiqish;
- Ma'lumotlar bazasi bilan ishlash va uni boshqarish;
- Elektron o'quv qo'llanmadan foydalanish yo'riqnomasini yaratish.

## 2.1. PHP web dasturlash tili haqida umumiy ma'lumotlar. Undagi funksiya va ob'yektlar

### O'zgaruvchilar

O'zgaruvchilar hamma dasturlash tilining asosidir. Bularga O'zgaruvchi sonlar, belgilar va boshqalar kiradi.

Misol qilib sonlarni olaylik. Ular butun sonlar, qoldikli sonlar bo'lishi mumkindir va yana bularni orasiga ikkilik, sakizlik, o'n oltilik sanoq sistemasidagi sonlar ham kiradi. Yoki belgilardan &, %, /, \$, # va yana harflar ham kiradi.

Agarda biz qandaydir o'zgaruvchiga nimadir belgilab berish niyatimiz bo'lsa unda biz quyidagicha yozishimiz shartdir:

```
$ bilim = "kitob";
```

```
$ rang = "qizil";
```

Bulardan tashqari yana ishchi belgilarimiz bor ya'ni:

<i>Nomi</i>	<i>Ta'rifi</i>
\n	Yangi qator
\r	kursorni qaytishi
\t	gorizontal tabulyatsiya
\\	Teskari qiyshiq chiziq
\\$	Dollar belgisi
\"	Qo'shtirnoq belgisi
\[0-7]{1,3}	Sakizlik sanoq sistemasini yozilish
\x[0-9A-Fa-f]{1,2}	O'n oltilik sanoq sistemasi yozilishi

Bu belgilarni kerakli bo'lgan joylarda qulaygina ishlatishingiz mumkindir.

Bu yerda "" belgisi bilan " belgilari orasida farq bordir.

Misol:

```
$double_list = "item1\nitem2\nitem3";
```

```
$single_list = 'item1\nitem2\nitem3';
```

Ekkranga chiqarganingizda single\_list tamoman bir biri bilan birga va \n belgisi bilan chiqadi, double\_list esa bu belgilersiz bo'sh joy bilan birga chiqadi.

## Massivlar

Massiv: Bir qancha o'zgaruvchilarni birligidir. yozilishi: \$ismi[raqami]

misol:

```
$rang[0]="ko'k";  
$rang [1]="qizil";  
$rang [2]="yashil";
```

Agarda siz print \$rang[1] desangiz unda oynangizga "qizil" chiqadi. Yoki array() komandasini ishlatishingiz mumkin.

misol:

```
$kun=array("juma", "shanba", "yakshanba");
```

Agarda sizga massivlarni so'z bilan ajratish qilay bo'lsa unda PHP buni ham qabul qiladi. misol:

```
$maktab["ismi"]="Cho'lpon";  
$maktab["raqami"]="258";  
$maktab["shahri"]="Toshkent";
```

Ichma ich massivlar ham yoza olasiz.

yozilishi: \$ismi[raqami1][raqami2]..[raqamiN];

misol:

```
$shahmat[0][0]="Peshka";  
$shahmat[0][1]="Tora";  
$shahmat[1][1]="Ot";
```

## Mantiqiy funksiyalar

TRUE, FALSE

Solishtirish funksiyalariga "if else" funksiyasidir. Bunday solishtirishda sizda TRUE (Rost) yoki FALSE (yolg'on) chiqadi. keling misol bilan ko'rsak.

```
$flag=TRUE;  
if ($flag==TRUE):  
    print "The flag is true!";  
else:
```

```
print "The flag is false!";
endif;
```

Agarda flag o'zgaruvchimiz rost bo'lsa unda ekranimizga "The flag is true!" yoziladi. Agarda yolg'on bo'lsa unda ekranimizga "The flag is false!" yoziladi.

\$flag o'zgaruvchiga biz TRUE berganimizda bizning "The flag is true" yozuvimiz ekranga chiqadi. Huddi shunday agarda siz FALSE bersangiz unda ekraningizga "The flag is false" yozuvi chaqadi.

Shu yerda yozib o'tish o'rinli deb bildim. TRUE va FALSE yozuvlarini siz raqam bilan ham belgilab berishingiz mumin. TRUE=1 FALSE=0.

Agarda yuqoridagi misolda \$flag=1; deganingizda ham oynangizga "The flag is true!" yozuvi chiqqan bo'lardi.

### **Nomlash**

Ob'yektlarni, funksiyalarni, o'zgaruvchilarni nomlash uchun siz hamma belgilardan (lotincha) ishlatishingiz mumkin. Faqatgina boshlanishi sondan va yordamchi belgilardan bo'lishi kerak emas. Va ichida yordamchi belgilari bo'lmasligi kerak.

misol:

Mumkin	Mumkin emas
my_function	This&That
Size	!counter
somewords	4word
otherstring	!otherstring

### **O'zgaruvchilarni tanitish**

Yuqorida o'zgaruvchilarga qandaydir so'zni, yoki gapni tanitish, ya'ni berishni ko'rib chiqdik.

Lekin yana bir bor qaytarib o'tish yana ham ko'proq o'rganishingizga, agarda tushunmagan bo'lsangiz tushunib yetishingizga yordam beradi.

So'z yoki gaplarni quyidagicha qilib tanitamiz:

```
$sentence="Salom hayot!";
```

Sonlarni esa:

```
$price=43.22;
```

```
$ogirligi=185;
```

Ichki o'zgaruvchi

### **Ichki (lokal) o'zgaruvchi**

Ichki (lokal) o'zgaruvchi deb funksiyaning ichida tanitilgan o'zgaruvchiga aytiladi.

misol:

```
<?
```

```
$x=4;
```

```
function assignx(){
```

```
$x=0;
```

```
print “\ $x funksiya ichidagi $x”;
```

```
}
```

```
assignx();
```

```
print “<BR> \ $x tashqaridagisi $x”;
```

```
?>
```

Shunda ekraningizda \$x ning ikki xil raqami chiqadi. 4 bu umumiy tanitilgan raqami, 0 esa ichkisidir.

### **Global o'zgaruvchi**

Global, ya'ni umumiy o'zgaruvchanlar butun sahifamiz bo'yicha hamma joyda ishlatishingiz mumkindir. Yuqoridagi misolda \$x=4 bu bizning global o'zgaruvchimizdir. Agarda uni biz funksiya ichida ishlatmoqchi bo'lsak unda:

```
$sonlar==15;
```

```
function arifmet() {
```

```
GLOBAL $sonlar
```

```
$sonlar=$sonlar+10;
```

```
print “Bizning son $sonlar”;} 
```

```
arifmet();
```

## Static o'zgaruvchi

Static: Bu ichki o'zgaruvchi bo'lib, ammo funksiyaning ichidan chiqqaningizdan keyin ham o'z qiymatini saqlab qoladi, va shu funksiyaning keyingi bor ishlatganingizda o'z qiymati bilan ishlaydi. Misolda ko'raylik:

```
function keep_track(){  
    STATIC $count=0;  
    $count=$count + 10;  
    print $count;  
    print "<BR>";  
}  
keep_track();  
keep_track();  
keep_track();
```

Agarda siz shu kodni ishlatsangiz unda sizning ekraningizda 10 20 30 sonlari yoziladi. Ko'rib turganingizdek har gal yangittan ishlatganingizda count o'z qiymatini saqlab qoldi.

## Bir turdan ikkinchi turga o'tish

Agarda siz qandaydir son hoh butun son hoh qoldiqli bir turdan ikkinchisiga o'tkazish niyatingiz bo'lsa unda bu dars sizga yordam beradi.

Keling avval qanday turlari borligini yozib so'ngra bir turdan ikkinchisiga o'tkazishni ko'rib chiqamiz.

<i>Turi</i>	<i>ta'rifi</i>
(int) yoki (integer)	butun son
(real), (double), (float)	qoldiqli sonlar
(string)	harf, belgilar
(array)	Massivlar
(object)	Ob'yektlar

Keling bir turdan ikkinchi turga o'tkazishni ko'raylik.

```
$son=12;  
$son=(double)$son;
```

shunda 12 soningiz 12.0'ga aylanadi.

Yoki yana bir boshqa misol

```
$son1=4.0;
```

```
$son2=5;
```

```
$son3=(int)$son1+$son2;
```

Shunda son3 9 soniga teng bo'ladi.

```
$son1=7.8;
```

```
$son2=6;
```

```
$son3=(int)$son1+$son2;
```

Shunda son3=13'ga

Agarda siz ob'yekt bermoqchi bo'lsangiz unda:

```
$model="Toyota";
```

```
$new=(object)$model;
```

siz buni ekranga chiqarish uchun scalar so'zidan foydalanishingiz kerak, ya'ni:

```
print $new->scalar;
```

### **O'zgarmaslar**

Oldindan aniqlab berilgan son yoki sozlar tarkibidir.

Misol uchun agarda biz PI so'ziga 3.14 sonini bermoqchi bo'lsak va bu son hech ham o'zgarmasligini bilsak unda biz o'zgarmasni yozishimiz kerak (constant) misol:

```
define("PI", "3.14");
```

```
print "PI soni bu";
```

```
print PI;
```

Shu yerda bazi bir funksiyalarni ham ko'rib chiqaylik.

print() - Bu oynaga ma'lumot yuborishdir.

sprintf() - Bu qandaydir ma'lumotni formatlash(kerakli shaklga kiritish) uchun va uni qandaydir o'zgaruvchiga berish uchun ishlatiladi.

printf() - qandaydir ma'lumotni formatlab oynaga yuborish uchun ishlatiladi.

misol:

<?

```
$format=""The are %s monkeys in the $d";
```

```
printf($format, $num, $location);
```

?>

agarda biz \$num'ga 3 bersak va \$location'ga "tree" bersak unda quyidagi oynamizga chiqadi. "The are 3 monkeys in the tree" bo'ladi.

yoki

<?

```
$money1=68.75;
```

```
$money2=54.35;
```

```
$money=$money1+$money2;
```

```
$formatted=sprintf("%01.2f", $money);
```

?>

shunda \$formatted=123.00 bo'ladi, agarda biz sprintf siz oynaga chiqarganimizda unda 123 chiqgan bo'lardi.

Formatlashga kiritish turlari.

%-protsent belgisi.

b- son, ikkilik sanoq sistemasida (0 va 1)

c- son, songa mos keladigan ASCII belgilarini chiqaradi. (65=A belgisi)

d- son, belgili onlik sanoq sistemasida (+10 yoki -10) u-son, belgilanmangan onlik sanoq sistemasida (10, 5) f- qoldiqli son, qoldiqli son shaklida (1.25, 20.36)

o- son, sakizlik sanoq sistemasida (23, 65)

s- belgi, ya'ni harflar yoki boshqa belgilar

x- son, o'n oltilik sanoq sistemasida (harflari kichkinada yoziladi) (12ab, d23c)

X- son, o'n oltilik sanoq sistemasida (harflari kattada yoziladi) (12AB, D23C) Mana shular bilan siz foydalanib ma'lumotingizni formatlashingiz mumkindir.

## Operatorlar

Operatorlar	Ta'rifi
()	Qavs
new	Yangi, ob'yekt hosil qilish
! ~	Mantiqiy YO'Q
++ --	Bitta songa oshirish, kamaytirish
@	Hatolarni berkitish
/ * %	Bo'lish, ko'paytirish, qoldiq
+ - .	Qo'shish, ayirish, kesish
<< >>	Chapga, o'nga surish
< <= > >=	kichik, kichik va teng, katta, katta va teng
== != === <>	Teng, teng emas, o'xshash, teng emas
& ^	AND, XOR, OR
&&	AND, OR
?:	uchlik operatori
= += *= /= .=	Belgilab berish operatorlari
%= &=  = ^=	
<<= >>=	
AND XOR OR	Mantiqiy va, XOR, yoki

Keling yuqoridagi operatorlar bilan ba'zi bir misollarni ko'rib chiqaylik.

*\$a=5; // a teng bechga*

*\$a=3\*3\*4\*5;// a teng shu sonlar ko'paytmasiga*

*\$a=(5-(6-4));//avval 6-4 songra esa 5-(6-4) bo'ladi*

*\$a+=5; // \$a ning qiymatiga 5 qo'shilib yana \$a ga yozish.*

*\$a/=5; // \$a ning qiymati 5'ga bo'linib keyin esa qiymatni \$a'ga yozadi.*

*\$a++; // \$a ning qiymatiga 1 qo'shiladi*

*\$a--; // \$a ning qiymatidan 1 ayirib tashlanadi*

Lekin operandlarni joylashishini ham o'zgartirib yozishingiz mumkin, ammo uni bajaruvchi ishi ozgina o'zgaradi.

Misolda ko'raylik:

*\$qiymat=\$a++ // avval \$qiymat'ga \$a ning qiymatini yozadi,*

so'ngra  $\$a$  ning qiymatiga 1 qo'shadi.

$\$qiymat=++\$a$  //avval  $\$a$  ning qiymaga 1 qo'shiladi, so'ngra  $\$qiymat$  ga  $\$a$  ning qiymatini beradi.

agarda siz tenglikni tekshirmoqchi bo'lsangiz unda:

$\$a==\$b$  //  $\$a$  ni  $\$b$  bilan teng/tengmasligini tekshiradi

$\$a<\$b$  //  $\$a$  ni  $\$b$  dan kichikligini tekshiradi

Uchlik tekshirish.

$(\$a-12)?5:-1$  //agarda  $\$a=12$  bo'lsa unda 5 soni qaytadi, agarda teng bo'lmasa 1 soni qaytadi.

### **round()**

Bu funksiyamiz qandaydir qoldiqli sonni yahlitlash uchun ishlatiladi.

`float round(float son [,nuqtadan_keyin])`

bu yerda son bu biz yahlitlamoqchi bo'lgan sonidir, nuqtadan\_keyin bu nuqtadan so'ng qancha songacha yahlitlashni bildiradi. misol:

```
<?  
$son=12.3256;  
echo $round($son, 3);  
echo $round ($son, 2);  
echo $round ($son, 1);  
?>
```

shunda bizning oynaga quyidagilar chiqadi:

```
12.326  
12.33  
12.3
```

Bu yerda ko'rganingizdek 5 va undan yuqori bo'lsa unda oldindigi songa 1 qo'shadi agarda 5 dan kam bo'lsa unda faqatgina olib tashlaydi xolos.

### **ceil()/floor()**

`ceil()` bu keyingi songa yaxlitlaydi (12.3 bo'lsa unda 13 bo'ladi)

`floor()` bu bitta past songa yaxlitlaydi (12.6 bo'lsa unda 12 bo'ladi)

Ya'ni bu yerda qoldiqga qaralmaydi faqatgina butun qismini oladi. Demak bitta yuqorisini olish uchun ceil ishlatiladi va bitta pastgi sonni olish uchun esa floor funksiyasi ishlatiladi.

### **abs()**

Bu funksiya sonni oldidagi belgisiga olib tashlaydi, ya'ni agarda sizda manfiy son bo'lsa unda uni musbatga o'giradi.

misol:

<?

```
echo abs(4.2);
```

```
// oynaga 4.2 chiqadi
```

```
echo abs(4.2); // oynaga 4.2 chiqadi
```

```
echo abs(5); // oynaga 5 chiqadi
```

?>

### **rand()/srand()**

Qandaydir tasodifiy son olish uchun shu ikki funksiya ishlatiladi.

```
srand((double)microtime()*1000000)
```

```
$randomnumber=rand()
```

Mana shu ko'dni ishlatsangiz sizda haqiqatdan ham tasodifiy son bo'ladi. Agarda siz ikki son orasida son olmoqchi bo'lsangiz unda *rand(0, 10)* kabi ishlatishingiz mumkindir, shunda sizda 0 va 10 orasida qandaydir son chiqadi.

### **Sikl operatorlari**

#### **IF**

If operatori, bu bizning tekshirish operatirimizdir.

Yozilishi:

```
if (tekshirish) {
```

```
  yozuvimiz
```

```
}
```

```
yoki
```

```
if (tekshirish) {
```

```
  yozuvimiz
```

```
}
```

```
else {
```

```
  yozuvimiz
```

```
}
```

Misol:

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

```
  print “\ $a teng ekan \ $b ga”;
```

```
}
```

```
else {
```

```
  print “teng emas ekan”;
```

```
}
```

Bu tekshirishimiz, bizga \$a va \$b ni bir biri bilan solishtirib agarda ular teng bo'lsa unda ekranimizga “\$a teng ekan \$b ga” yozuvini chiqaradi, agarda teng bo'lmasa unda “teng emas ekan” yozuvini chiqaradi.

Keling butun bir misolni ko'rib chiqamiz.

```
<?
```

```
$a=15;
```

```
$b=15;
```

```
?>
```

```
<html>
```

```
<head>
```

```
<title>Iflar </title>
```

```
</head>
```

```
<body>
```

```
<?
```

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

```
  print “Ikki son bir biriga teng”;
```

```
}  
else{  
print "Ikki son bir biriga teng emas ekan";  
}  
?>  
</body>  
</html>
```

### **elseif**

Bu if operatorimizga yana bitta tekshirishni qo'shib beradi.

```
if (tekshirish) {  
yozuv  
}  
elseif (tekshirish) {  
yozuv  
}
```

Misolda ko'raylik.

```
if ($a<200) {  
print "Sonimiz 200 sonidan kichik ekan";  
}  
elseif ($a<500) {  
print "Sonimiz 200 dan katta, lekin 500 dan kichik ekan";  
}  
else {  
print "Sonimiz 500 dan ham katta ekan";  
}
```

Mana shu misolda sonimiz tekshirilib, uni qiymatiga qarab ekranimizga yozuv chiqadi.

IF operatorini yozishni ikkinchi turi ham bordir. Bu ikki tur bir biridan vazifasi bo'yicha hech qanday farq qilmaydi.

*if (tekshirish):*

*yozuu*

*else:*

*yozuu*

*endif;*

### **while**

Bu operatorimiz qandaydir ishni bir necha marotaba qayta qayta qilish uchun juda ham qulaydir.

*while (tekshirish):*

*yozuu*

*endwhile;*

keling bir misol bilan ko'rib chiqaylik.

*\$n=5;*

*\$ncopy=\$n;*

*\$factorial=1;*

*while (\$n>0):*

*\$factorial=\$n\*\$factorial;*

*\$n;*

*endwhile;*

*print "The factorial of \$ncopy is \$factorial";*

### **do..while**

Operator do..while huddi while kabi ishlaydi, lekin kichkina farqi tekshirish boshida emas oxirida bo'ladi, shuning uchun do..while eng kamida bir marotaba ishlaydi.

*do:*

*yozuu*

*while (tekshiruv);*

keling misolda ham ko'raylik.

*\$n=5;*

```

$ncopy=$n;
$factorial=1;
do:
$factorial=$n*$factorial;
$n;
while ($n>0);
print "The factorial of $ncopy is $factorial";

```

bu ham yuqoridagi misol kabi oynamizga “The factorial of 5 is 120” kabi yozuv chiqaradi.

### **for**

Bu operatorimiz biz uchun qandaydir programma kodimizni qanchadir marotaba qayta qayta ishlatib beradi.

```

for(tanitish:tekshiruv; o'zgarishi){
yozuv
}

```

Keling misol bilan ko'rib chiqamiz.

```

for ($i=10;$i<=100;i+=10){
print "\$i=$i <br> ";
}

```

shuni qandaydir html kodingizga qo'yib tekshirib ko'ring. Bu misolda bizning operatorimiz \$i ga boshida 10 sonini beradi va uni yuzdan kichik yoki teng bo'lgunicha ishlatadi, va har bir oshishi 10 sonidir, har gal \$i'ga 10 soni qo'shilib boradi.

### **break**

Bu qo'shimcha operatorimiz qandaydir ishni bo'lib qaytadan chiqib ketish uchundir.

Ya'ni misolga qarang:

```

$x=5;
for ( ;; $x+=2) {

```

```
print "$x ";  
if ($x==15):  
break;  
endif;  
}
```

Shunda for operatorimiz \$x ning 15 soni bo'lganigacha ishlaydi, for operatorini ishini esa break operatori bo'ladi.

### **foreach**

Massivlar bilan ishlash uchun for operatorining qulaylashtirilganidir.

```
foreach (massiv as $element) {  
yozuv  
}  
yoki  
foreach (massiv as $kalit=>$element){  
yozuv  
}
```

misol uchun:

```
$menu=array("juma", "shanba", "yakshanba");  
foreach ($menu as $item){  
print "$item <br>";  
}
```

### **switch**

Huddi if operatori kabidir, lekin undan farqi, birdaniga bir qancha tekshirishlar tekshiriladi.

Keling eng yaxshisi misol bilan ko'raylik.

```
switch(tanitish){  
case(shart):  
yozuv  
case(shart):
```

```
yoZuv
...
default:
yoZuv
}
```

bunda tanitilgan o'zgaruvchini shartlariga qarab turib nimani bajarishini belgilab berasiz. Agarda hech qanday shart bajarilmasa unda default'ning ichidagi ish bajariladi.

Misol:

```
$user_input="recipes";
switch($user_input){
case("search"):
print "Lets perform search";
break;
case("dictionary"):
print "What word would you like to look up?";
break;
case ("recipes"):
print "Here is a list of recipes... ";
break;
default:
print "Here is the menu... ";
break;
}
```

Mana ko'rganingizdek \$user\_input bir qancha so'zlarga tekshiriladi (search, recipes, dictionary) va recipes to'g'ri kelganligi uchun undagi ish nima bo'lsa shuni bajariladi. Nima uchun break ishlatdik, sababi ishni qilib bo'lgandan keyin biz switch operatorini ishini bitirishini hohlaymiz, qolganlarini tekshirishi bizga kerak emas, shunda ortiqcha vaqt ketmaydi.

## **continue**

Agarda for, yoki boshqa davomli operatorlaringizni ichida yozgan kodingizni qandaydir qismidan so'ng qolganlarini bajarilmasdan yana boshidan boshlanishini hohlasangiz unda continue operatorini ishlatasiz.

Keling misolda ko'raylik.

```
$boundary=100;  
for ($i=0;$i<=boundary; $i++){  
if (! is_prime($i)):  
continue;  
endif;  
$prime_counter++;  
}
```

Bu misolda toq son bo'lsa (is\_prime) unda for ishini davom etiradi, ya'ni \$prime\_counter++ operatori ishlamasdan boshiga for operatoriga ish beriladi. Boshiga ish berilishi degani for yana boshidan ishini boshlamaydi, vaholanki for o'z ishini davom etiradi, lekin for ning ifdan keyingi ishlari bajarilmasdan boshiga qaytadi.

## **Funksiya nima?**

Funksiya bu programmadagi bir programma parchasi bo'lib o'z ismi bo'ladi. Funksiya qandaydir aniq bor ishni bajaradi. Funksiyani ustunligi shundaki, funksiya bir marotaba yozilib uni xohlagan joyingizda qayta qayta ishlatishingiz va sizga kerak bo'lgan vaqtda uni osongina o'zgartirishingiz mumkin.

Yuqorida biz funksiyani ozgina bo'lsa ham ko'rib chiqdik, bu yerda esa sizga batafsil ya'ni to'la shakilda tushuntirib o'taman.

Yozilishi

```
function ($o'zgaruvchi1, $o'zgaruvchi2, $o'zgaruvchi3...){  
  yozuvlar  
}
```

Funksiyani siz bir birini ichiga qo'shib ham yozishingiz mumkindir. Va yana bir misol, agarda siz universal copyright funksiyani yozmoqchi bo'lsangiz unda sahifa yoki nimanidir nomini tashqaridan olishingiz mumkin. Misolga qarang:

```
function copyright($kitob){  
    print "Copyright &copy; 2006 $kitob , GNU GPL Litsensiyasi  
    ostida"  
}  
copyright("PHP davrasida");
```

Funksiyangiz qandaydir yozuv yoki sonni ham sizga qaytarishi mumkin. Ya'nikim, agarda siz hisob kitob ishini qilmoqchi bo'lsangiz va sizga uni natijasi kerak bo'lsa, unda siz funksiyangizdan qandaydir natijani qaytarishingiz mumkin.

Misolga qarang:

```
$narx=15;  
$son=3;  
function hisob($narx, $son){  
    return $narx*$son;  
}  
$total=hisob($narx, $son);  
print "Umumiy narhi: $total";
```

shunda sizga narxni songa qanchaligiga ko'paytirib natijasini chiqarib beradi.

Yana funksiyalarni ich ichiga, ya'ni o'zidan o'zini chaqirishingiz mumkindir, bu "recursive function" – "rekursiv funksiya" deb ataladi. Misol bilan ko'raylik:

```
function summation($count){  
    if ($count!=0):  
        return $count+summation($count-1);  
    endif;}  
$sum=summation(10);  
print "Summation=$sum";
```

bu yerda qilinadigan ishi shuki, avval 10 ga 9 ni qo'shadi ya'ni, o'zini yana chaqiradi faqat 9 soni bilan 9 sonidagi funksiya esa 8 bilan yana o'zini chaqiradi. Bu oxirida 0 bo'lguncha davom etadi. Shunda ekraningizda 1 dan 10 gacha sonlar yig'indisi chiqadi.

Agar siz juda ham ko'p funksiyalar ishlatish niyatida bo'lsangiz unda funksiyalar uchun alohida bir fayl yaratib shu faylga hammasini yozishingiz mumkin, keyin esa shu faylni asosiy oynangizda chaqirib ichidagi funksiyalarni ishlatishingiz mumkin.

Buning uchun qandaydir qo'shimcha .inc yaratib ya'ni fayl yaratib, uni ichiga bir qancha funksiyalar yozing.

Misol: qo'shimcha.inc

```
<?  
function copyright(){  
...  
}function hisob(){  
...  
}  
function summation (){  
...  
}  
?>
```

Keyin esa asosiy kodingizda *include (faylga yo'l);* operatori bilan qo'shing.

Ya'ni:

```
<?  
include (qoshimcha.inc);  
hisob();  
$sum=summation(10);  
...  
?>
```

## Object Oriented Programming (OOP)

Bu programmalash, yangi programmalash turidir, ya'ni, agarda siz C++, Visual Basic, Delphi kabi programmalash tillarini o'rgangan bo'lsangiz bu siz uchun yangilik bo'lmaydi, agarda bilmasangiz unda shu mavzuni ko'rib chiqqaningizdan so'ng sizda bu programmalash turi haqida ma'lumotingiz bo'ladi. OOP - Bu programmangizda qandaydir bir ob'yekt yaratib undan keyingi o'zingizga kerak bo'lgan yerlarda ishlatishdir. Ya'ni bo'lmagan bir ob'yekt hosil qilishdir.

Class

Class: bu OOPning asosidir. Ya'ni siz klas yaratasiz va undan keyinchalik foydalanishingiz mumkindir.

Keling misolda ko'raylik

```
class Class_name{ //Class_name nomli klas yaratdik
var $attribute_1; //ichida xohlaganimizcha o'zgaruvchi tanitamiz
...
var attribute_2;
function function1(){ //qancha istasak shuncha funksiya yaratamiz
...
}
...
function functionN(){
...
}
...
} //bu yerda esa Class_Name yopiladi.
```

Mana ko'rganingizdek biz bu yerda class bilan boshladik va ichida kerakli o'zgarmas, va kerakli funksiya yaratdik.

Agarda siz ob'yektingizni ichida shu ob'yektingizni qandaydir funksiyasiga yoki o'zgaruvchisiga murojat qilmoqchi bo'lsangiz unda `$this` funksiyasidan foydalanasiz.

Misol:

```
<?
class Webpage(){
var $bgcolor;
function setBgColor($color) {
$this>
bgcolor=$color;
}
function getBgColor(){
return $this>
bgcolor;
}}
?>
```

Mana biz o'zimizga klas yaratdik, endi ob'yekt yaratamiz.

`$home_page=new Webpage;` //home\_page nomli ob'yekt yaratdik, ya'ni u aynan Webpage kabidir.

`$home_page->setBgColor("black");` //Bu yerda esa biz home\_page'ga black rangini yubordik.

Siz bu yaratgan klaslardan yana bir qancha ichki klas yarata olasiz, ya'ni, Webpage klasidan foydalanib unga boshqa qo'shimchalar ham qo'shishingiz mumkindir.

Misol:

```
<?
class Webpage(){
var $bgcolor;
function setBgColor($color) {
```

```

$this>
bgcolor=$color;
} function getBgColor(){
return $this>
bgcolor;
}
}
class UzWebpage extends Webpage{
var admin;
function setSiteAdmin($adm){
$this>
admin=$adm;
}
function getSiteAdmin(){
return $this>
admin
}}
?>

```

Mana bu yerda biz Webpage'ni yana ham kengaytiridik, ya'ni UzWebpage qildik va unga qo'shimcha funksiya berdik, ya'ni sitening administratorini berib chiqdik.

### **Array get\_class\_methods(string class\_ismi)**

Bu funksiyamiz klasning ichida tanitilgan funksiyalarni ko'rsatib beradi.

Misol uchun yuqoridagini tekshirib ko'raylik ya'ni eng oxiriga

```

$test=get_class_methods(Webpage);
print $test[0];
print $test[1];

```

yozib tekshirib ko'ring, sizda setbgcolor, va getbgcolor chiqishi kerak.

### **Array get\_class\_var(string class\_ismi)**

Classning ichidagi o'zgaraslarni qaytaradi. Ya'ni yuqoridagi `get_class_methods` kabi ammo, farqi o'zgaraslarni qaytarishidadir.

### **Array `get_object_vars(ob'yekt_ismi)`**

Yaratilgan ob'yektning ichida agarda qandaydir o'zgarasga nimadir tanitilgan bo'lsa shularni hammasini ko'rsatib beradi.

Kelin misolga qarang:

```
<?
class Vehicle{
var $wheels;
}
class Land extends Vehicle{
var $engine;
}
class car extends Land{
var $doors;
function car($doors, $eng, $wheels){
$this>
doors=$doors;
$this>
engine=$eng;
$this>
wheels=$wheels;
} function get_wheels(){
return $this>
wheels;
}
}
$toyota=new car(2, 400, 4);
$vars=get_object_vars($toyota);
```

```
while (list($keys, $value)=each($vars)):  
print $keys==> $value <br>”;  
endwhile;  
?>
```

shunda oynangizda

```
doors==>2  
engine==>400  
wheels==>2
```

chiqadi.

**method\_exists()**

Ob'yektimizda qandaydir funksiya bor yo'qligini tekshirib chiqadi. Agarda bor bo'lsa unda TRUE agarda yo'q bo'lsa unda FALSE qaytadi.

**get\_class()**

Qandaydir ob'yektimiz qaysi klasni ichidaligini qaytaradi.

Misolga qarang:

```
Class Vehicle {  
...  
}  
class Land extends Vehicle {  
...  
}  
$car=new Land;  
$class_a=get_class($car);
```

shunda `class_a=Land` bo'ladi.

`get_parent_class()`

Aynan yuqoridagi kabi ammo, asosiy oyna classni qaytaradi, ya'ni

```
$class_a=get_parent_class($car);  
class_a=Vehicle bo'ladi.  
is_subclass_of()
```

Bu funksiya qandaydir ob'yekt qaysidir klasdan yaratilib yaratilmaganini tekshiradi.

*Bool is\_subclass\_of(object ob'yekt\_ismi, string class\_ismi)*

*get\_declared\_classes()*

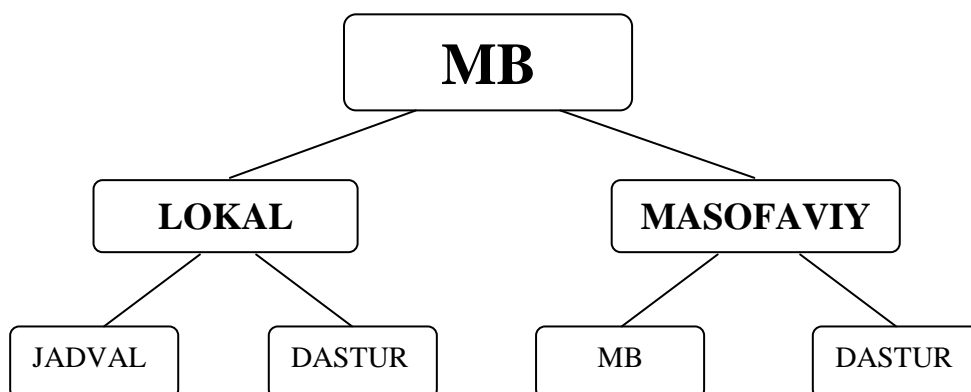
Bu funksiya esa, qanday klaslar yaratilgan ya'ni tanitilgan bo'lsa hammasini ekranga chiqaradi. Misol:

*Array get\_declared\_classes()*

## 2.2. Ma'lumotlar bazasi va uning asosiy tushunchalari

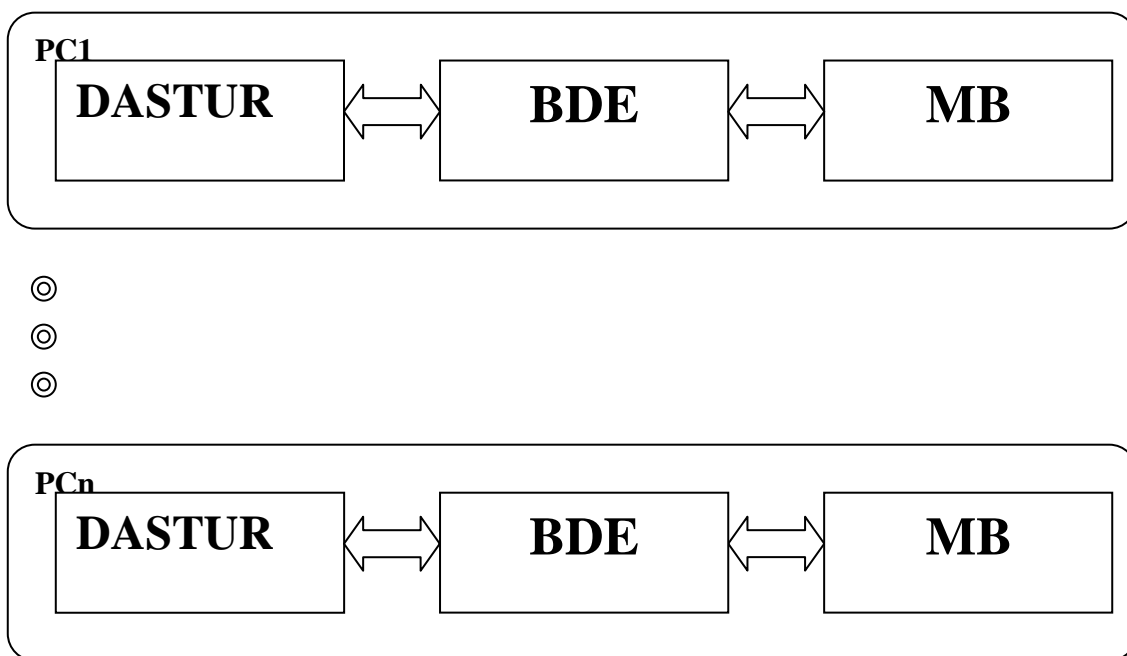
### Ma'lumotlar bazasi tarixi.

Ma'lumotlar bazasi lokal va masofaviy bo'lishi mumkin.



#### 2.2.1 – rasm. Ma'lumotlar bazasining struktura tuzilishi.

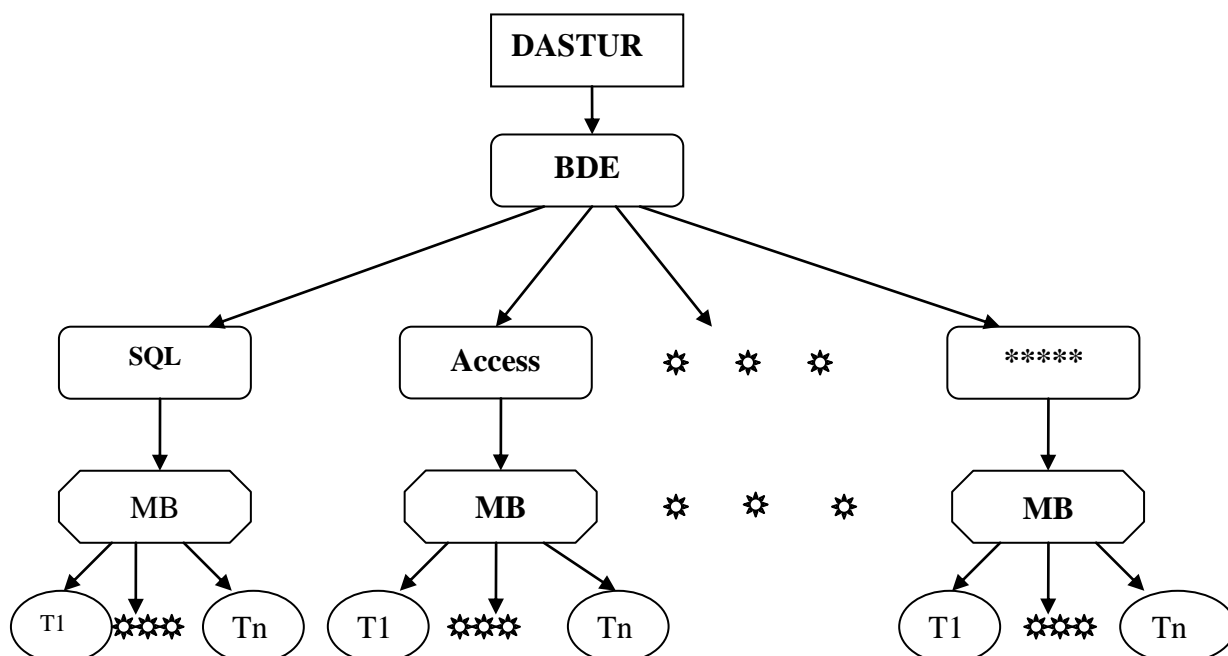
Lokal ma'lumotlar bazasi undan foydalanayotgan dastur bilan bitta kompyuterda joylashgan bo'ladi. Bunday hollarda axborot tizimlari lokal axborot tizimlarida qurilgan deb hisoblanadi. Ma'lumotlar bazasi bilan ishlash bir foydalanuvchi rejimida amalga oshiriladi.



#### 2.2.2 – rasm. Dastur va MB ning BDE orqali bog'lanishi.

Ikkita bir-birini tushunmaydigan odamlarni bir-birini tushunish uchun o'rtasiga tarjimon kerak bo'ladi. Dastur va ma'lumotlar bazasini ham bir-biriga bog'lash uchun BDE dan foydalanamiz.

Bunda dastur quyidagicha ishlaydi:



2.2.3 – rasm.Dastur va bir necha MB ning BDE orqali bog'lanishi.

Fayl server arxitekturasi ko'pincha foydalanuvchilari kam bo'lgan tarmoqlarda foydalaniladi. Ko'pincha bunday tizimlar Paradox yoki Dbase bazalari asosida quriladi. Fayl server arxitekturasi quyidagi kamchiliklarga ega:

Foydalanuvchi doimo lokal ko'chirish bilan ishlaidi. Agar jadvallarda o'zgarish bo'lsa lokal nusxani ma'lumotlar bazasini uzatadi va qayta qabul qiladi. Buning hisobiga tarmoqda aylanma ma'lumotlar oshib ketadi va yuklanish yuzaga kelib qoladi.

Ikkinchi kamchiligi shundaki har bir kompyuterda o'zini Ma'lumotlar bazasini lokal nusxasi bo'lgani uchun unda amalga oshirilayotgan o'zgarishlar qolgan foydalanuvchilarga bir o'z vaqt noma'lum bo'lib turadi.

### **Ma'lumotlar bazasidagi ma'lumotlarni qayta ishlash operatsiyalari**

Ma'lumotlar bazasi ma'lumotlarini qayta ishlashga quyidagi operatsiyalar kiradi:

Ma'lumotlar bazasini yaratish.

Ma'lumotlar bazasini o'zgartirish.

Ma'lumotlar bazasini yangilash.

Ma'lumotlar bazasini tartiblash.

Ma'lumotlar bazasidan ma'lumotlarni qidirish.

Ma'lumotlar bazasini fayl ko'rinishida qayta ishlash.

Ma'lumotlar bazasini diskdan yuklash.

Ma'lumotlar bazasidan foydalanib hisobotlar yaratish va bosmaga chiqarish.

### MySQL ma'lumotlar bazasi

MySQL (oqilishi "May es kyu el"). Relyatzional ma'lumotlar bazasini boshqarish sistemasi. Turli operatsion sistemalarda qo'llanishga moslashtirilgan Open-Source-mahsulot bo'lib, ko'pchilik dinamik web-sahifalar uchun asos hisoblanadi. MySQL Shvetsiyadagi MySQL AB firmasi tomonida yaratilgan bo'lib, 2008 yilning fevralida Sun Microsystems tomonida sotib olingan. Endlikda dastur kodini yaratishgda ushbu firma javobgar. MySQL Server ochiq, ya'ni tekin dasturiy ta'minot bo'lib, General Public License himoyasidadir. MySQL AB/Sun firmasi dastur kodiga mualliflik huquqiga egaligi sababli, ushbu dasturiy ta'minotning kommerzial turi ham bor. MySQL nomining kelib chiqishi aniq emas. 1996 yildan beri ko'plab biblioteka va tool (vosita)lar nomi oldida My prefiksi qo'yila boshlangan. Ba'zi ma'lumotlarga ko'ra esa firmani tashkil qilganlardan biri hisoblanuvchi Monty Wideniusning qizi ismidan va SQL, ya'ni Structured Query Language (Strukturaviy so'rovlar tili) birikmasidan tashkil topgan. Unix, MacOS X va Linuxdan tashqari, MySQL Windows, OS/2 va i5/OS (avvalgi OS/400) uchun ham moslashtirilgan. Windows uchun esa ba'zida ayrim cheklanishlar uchraydi. MySQL ko'pincha web-servis uchun ma'lumotlarni saqlashda qo'llaniladi, webserver Apache va PHP bilan birgalikda ishlatiladi. MySQL hozirda 6.000.000 dan ortiq foydalanuvchilar va kuniga 35.000 dan ortiq download qilinadigan dunyoning eng mashhur ma'lumotlarni boshqarish sistemasidir.

MySQL ning asosiy funksiyalari quyidagilar:

### **mysql\_connect()**

Funksiya MySQL serveriga bog'lanish uchun ishlatiladi. Ulangandan keyin esa kerakli bazani tanlashingiz mumkin.

```
Int mysql_connect (string host [:port] [, string login] [, string passowrd])
```

misol:

```
@mysql_connect("localhost", "user", "user") or die ("Could not connect to MySQL server!");
```

```
<?
```

```
$link=@mysql_connect("localhost", "user", "user") or die ("Could not connect to server!");
```

```
?>
```

### **mysql\_select\_db()**

MySQL bazasini tanlash uchun ishlatiladi.

```
Int mysql_select_db(string bazani_nomi [, int ulanish_korsatgichi])
```

misol;

```
<?
```

```
@mysql_connect("localhost", "user", "user") or die ("Could not connect to MySQL server!");
```

```
@mysql_select_db("guestbook");
```

```
?>
```

### **mysql\_close()**

Bu oldindan ochgan ulanishni yopish uchundir.

```
Int mysql_close([int ulanish_korsatgichi])
```

```
<?
```

```
@mysql_connect ("localhost", "user", "user") or die ("Could not connect to MySQL server!");
```

```
@mysql_select_db("guestbook");
```

```
print "Ulandik";
```

```
@mysql_close();
```

```
print "Uzildik";
```

```
?>
```

Bu yerda biz avval ulandik keyin esa uzdik bog'lanishni. Bu yerda biz ulanish ko'rsatgichini ko'rsatmadik sababi bizda faqatgina bitta ulanish bor.

### **mysql\_query()**

Bu funksiyamiz bazadan o'qish yoki bazaga yozish ishlari uchun, ya'ni query uchun ishlatiladi.

```
Int mysql_query(string query [, int ulanish_korsatgichi])
```

query SQL tilidagi komanda yoki komandalar to'plami bo'ladi.

Misolni keyingi funksiyada ko'rib chiqamiz.

### **mysql\_affected\_rows()**

Agarda biz bazaga yozsak mysql\_query() bilan va biz qancha qatorlarni o'zgarganini bilmoqchi bo'lsak unda shu funksiya bizga yordam beradi.

```
Int mysql_affected_rows([int ulanish_korsatgichi])
```

misol:

```
<?
```

```
@mysql_connect ("localhost", "user", "user") or die("Could not connect to Server!");
```

```
@mysql_select_db("uzinfo") or die("Could not connect to Database!");
```

```
$query="update guestbook set name='test' where email='sinch@uz';
```

```
$result=mysql_query($query);
```

```
print "Ozgargan qatorlar soni:", mysql_affected_rows();
```

```
mysql_close();
```

```
?>
```

Shunda oynangizga bitta o'zgargan ya'ni "Ozgargan qatorlar soni: 1" deb chiqadi.

### **mysql\_num\_rows()**

Funksiyamiz Select komandasidan qaytgan qatorlar sonini beradi.

```
Int mysql_num_rows(int qaytish_raqami)
```

Qaytish raqami oldindan ishlatilgan query'ning raqamidir.

Misol:

```
<?
```

```
@mysql_connect ("localhost", "user", "user") or die("Could not connect  
to Server!");
```

```
@mysql_select_db("uzinfo") or die("Could not connect to Database!");
```

```
$query="select name from guestbook";
```

```
$result=mysql_query($query);
```

```
print "Tanlangan qatorlar soni:", mysql_num_rows($result);
```

```
mysql_close();
```

```
?>
```

Shunda oynamizga qancha qatorlar tanlangan bo'lsa shularni sonini qaytaradi.

**mysql\_result()**

Bu funksiya mysql\_query() bilan birgalikda ishlatiladi va Select komandasi uchundir. Select'dan qaytgan ma'lumotni ya'ni so'rovni natijasini qaytaradi.

```
Int mysql_result(int qaytish_raqami, int yozuv [, mixed joy])
```

misol:

```
<?
```

```
@mysql_connect ("localhost", "user", "user") or die("Could not connect  
to Server!");
```

```
@mysql_select_db("uzinfo") or die("Could not connect to Database!");
```

```
$query="select * from guestbook";
```

```
$result=mysql_query($query);
```

```
$x=0;
```

```
while($x<mysql_num_rows($result)):
```

```
$name=mysql_result($result, $x, "name");
```

```
$email=mysql_result($result, $x, "email");
```

```
$comments=mysql_result($result, $x, "comments");
```

```

print "$name $email $comments";
print "<br>";
$x++;
endwhile;
mysql_close();
?>

```

Shunda oynamizga ism email comments ketma ket chiqadi.

Misol uchun shu kabi

```

sinch uzinfo@inbox.ru hayot gozal
azmi azmi@yahoo.com men keldim

```

...

mana shu kabi.

### **mysql\_fetch\_row()**

mysql\_result() kabidir faqat farqi shundaki birdaniga bitta qatorni hammasini o'zgarmasga yozadi.

```

Array mysql_fetch_row(int qaytish_raqami)

```

keling misolda ko'raylik.

```

<?
@mysql_connect ("localhost", "user", "user") or die("Could not connect
to Server!");
@mysql_select_db("uzinfo") or die("Could not connect to Database!");
$query="select * from guestbook";
$result=mysql_query($query);
while($row=mysql_fetch_row($result)):
print "$row[name] $row[email] $row[comments]";
print "<br>";
endwhile;
mysql_close();
?>

```

Yuqoridagi funksiya kabi chiqaradi.

### **mysql\_fetch\_array()**

Aynan `mysql_fetch_row()` kabidir, ammo farqi shundaki bu funksiyada siz o'zgarmaslarga qanday yozishni belgilaysiz, ya'ni ismi bilan, yoki raqamli yoki ham raqam ham ismi bilan yozishi mumkin.

Array `mysql_fetch_array(int qaytish_raqami [, int index_turi])`

`index_turi: MYSQL_ASSOC` – ismi kabi yozish (`$row["name"]`)

`MYSQL_NUM` – raqamli (`$row[1]`)

`MYSQL_BOTH` – ikkalasi

agarda hech qaysisini ko'rsatmasangiz unda `MYSQL_ASSOC` ni ishlatadi.

Misol:

```
<?
```

```
@mysql_connect ("localhost", "user", "user") or die("Could not connect  
to Server!");
```

```
@mysql_select_db("uzinfo") or die("Could not connect to Database!");
```

```
$query="select * from guestbook";
```

```
$result=mysql_query($query);
```

```
while($row=mysql_fetch_array($result)):
```

```
print "$row[name] $row[email] $row[comments]";
```

```
print "<br>";
```

```
endwhile;
```

```
mysql_close();
```

```
?>
```

Mana yuqoridagini aynan qaytardi.

### 2.3 Serverlar haqida tushuncha. Apache serveri

Tarmoqqa ulangan, doimiy ishlaydigan va tarmoqning turli xizmatlarini bajaradigan kompyuterga SERVER deyiladi. Server tarmoqdagi ma`lumotlarni saqlaydi, tarmoqning boshqa nuqtalariga uzatadi va asosiy xizmati – foydalanuvchilarni internetga ulaydi. Ayni mana shu xizmatlarni bajaradigan tashkilotni – provayder deb atashadi. Ularning serverlari internetga kirish darvozalari hisoblanadi.

Bundan tashqari server so`zi quyidagicha shakllarda ham ishlatiladi.

**Ajratilgan server.** Yirik kompaniyalar ko`pgina kompyuterlarga ega bo`lishadi. Tashkilotning barcha tarmoq xizmatlarini boshqarish uchun alohida kompyuter ajratilib, unga maxsus dasturlar o`rnatiladi, internetga doimiy holda ulanadi va uni ajratilgan server deb atashadi. Bu server orqali barcha kompyuterlar internetga chiqa olishadi.

**Virtual server.** Kecha-yu kunduz internetga ulangan kompyuterni ishlatish oddiy foydalanuvchiga qimmatga tushadi. Shuning uchun ular boshqa serverning xotirasidan bo`sh joyni ijaraga olishadi va virtual serverga ega bo`lishadi. Bu xuddi o`zingizning internetda vakolat xonangizga o`xshaganday. Albatta, foydalanuvchi jismoniy server bilan foydalanish huquqiga ega bo`lmaydi, lekin tarmoq orqali virtual serverga kirib, kerakli ma`lumotlarni yozib qo`yishi mumkin. Keyin tarmoqqa kiritilgan ma`lumotlarni boshqa foydalanuvchilar kecha-yu kunduz o`qishi mumkin. Odatda, bunday xizmatni provayderlar ko`rsatadi va uni “xosting” (inglizcha - hosting) deb atashadi. Serverdagi xosting xizmatlari ikki xil bo`ladi: pulli va bepul. Pulli xosting xizmatlarga boy va tez ishlaydi, bepulida esa xizmat turlari kam, ishlash tezligi past va saytingizda o`zgalarning reklamalari paydo bo`lib turadi.

**Proksi-server.** Proksi-server – foydalanuvchi va internet orasidagi maxsus vositachi dastur. Uni ishlatish shart emas, lekin bu dastur yordamida web sahifalarining yuklash jarayonini tezlashtirish mumkin. Proksi-server Internetdan

ko'p ko'radigan sahifalarni qattiq diskdagi omborga saqlaydi va yana shu sahifalar so'ralganda Internetdan emas, balki xotirasidan ko'rsatadi. Odatda, proksi-server ajratilgan serverga o'rnatiladi va tarmoq xizmatlarini ham bajaradi: mahalliy tarmoq kompyuterlarini Internetga ulaydi. Tarmoqdagi kompyuterlar internet so'rovlarini ajratilgan serverga uzatishadi, serverda esa proksi dasturi so'rovlarni internetga chiqaradi va tashqaridan kelgan ma'lumotlarni tarmoq kompyuterlariga yetkazib beradi. Yana proksi dasturi yordamida kompyuterlarning tarmoqdagi ish jarayonini boshqarib, kuzatib turish ham mumkin.

Web server tushunchasini ikki xil ma'noda ishlatish mumkin. Agap WWW xizmatini ko'rsatish haqida borsa, u holda Web server tarmoq mijozlariga Web sahifa va saytlardan foydalanish imkoniyatini yaratib beruvchi dastur ma'nosini anglatadi. Agar so'z internetning texnik ta'minoti haqida borsa, u holda Web server Web resurslari saqlanayotgan va uning dasturiy ta'minoti ishlab turgan kompyuter ma'nosini anglatadi. Internet tarmog'ining ixtiyoriy bir kompyuterida bir nechta server dasturlar ishlab turishi mumkin.

Apache server ham yuqoridagi serverlar bajargan vazifalarni bajaradi, lekin farq shundaki, bu ichki ya'ni lokal server hisoblanadi. Bunda foydalanuvchi kompyuteri o'zining serveri ham hisoblanadi. Siz apache serverni o'rnatganingizda kompyuteringizda virtual disk hosil bo'ladi. Sizning tuzgan web sahifalaringiz ma'lumotlari o'sha diskda saqlanadi. Yoki huddi shu disk ichidagi ma'lumotlar operatsion tizim o'rnatilgan diskda (odatda C diskda) WebServers papkasi ichida joylashgan bo'ladi. PHP da tuzilgan dasturlaringiz natijasini serversiz ko'rib bo'lmaydi.

### Xulosa

Ushbu bobda PHP dasturlash tili va MySQL ma'lumotlar bazasi haqida umumiy ma'lumotlar berildi va ularning o'zaro bog'liklari to'g'risida gapirib o'tildi. Dastur ishlashi uchun kerak bo'ladigan

apache serverning PHP dasturlash tili bilan bog'liqligi to'g'risida ham to'xtalib o'tildi.

### III . Dastur interfeysi va foydalanuvchi yo'riqnomasi

#### 3.1 Dasturiy tizimga qo'yiladigan talablar

Dastur PHP web dasturlash tilida tuzilgan bo'lib, uning natijasini faqat internet brauzerlarida ko'rish mumkin. Masalan, Opera, Internet Explorer, Firefox Mozella va shu kabi. PHP ishlashi uchun birinchi navbatda server dasturi o'rnatilgan bo'lishi kerak. Usiz PHP operatsiyalarini bajarib bo'lmaydi. Ular Apache, Denver. Apache'ni o'z kompyuteringizga o'rnatishingiz kerak.

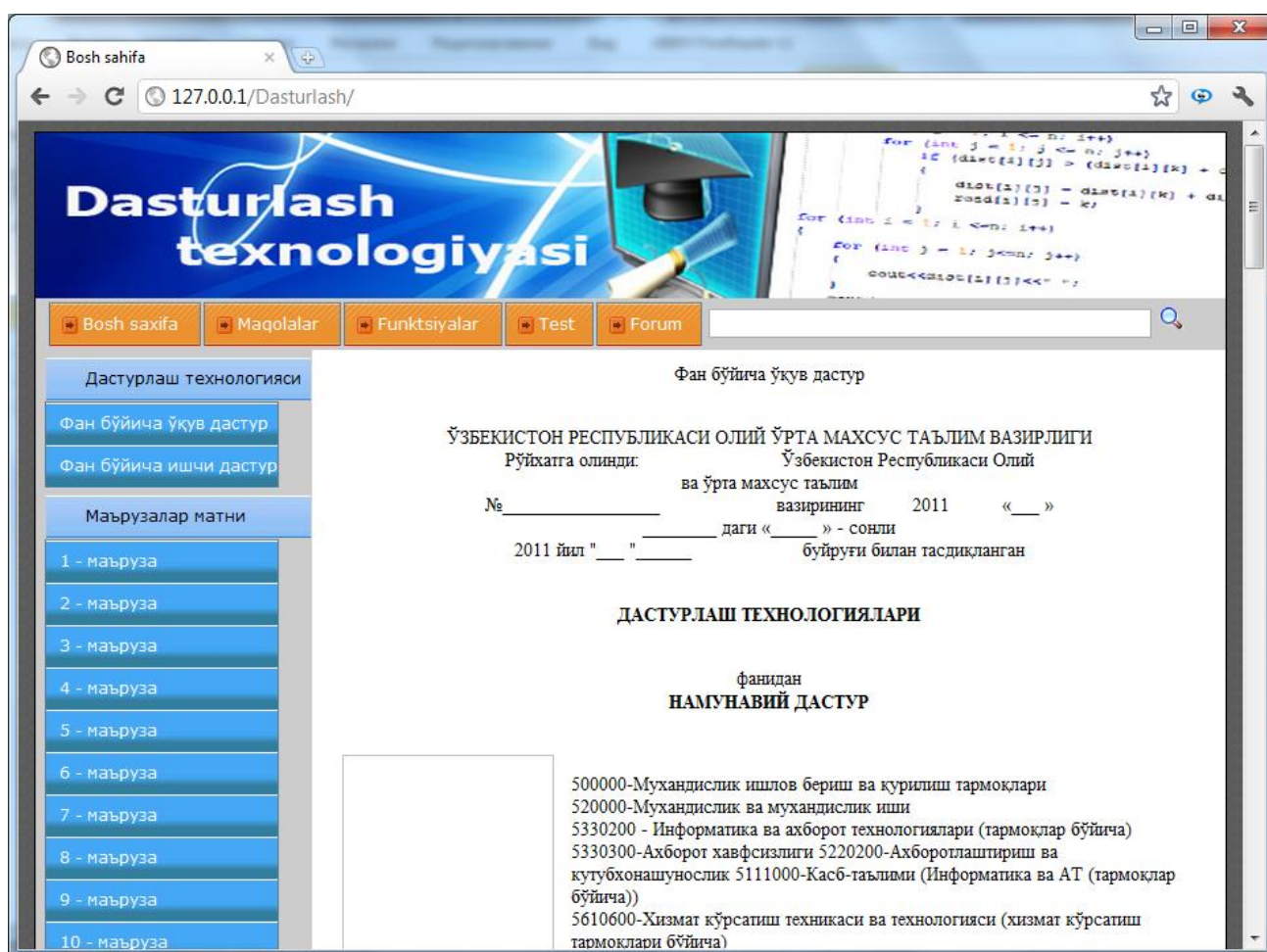
<b>Protsessor</b>	Minimum: 1.5 gigahertz (GHz)li Pentyum protsessori tavsiya etiladi
<b>Operatsion sistemalar</b>	Apache Linux yoki Unix sistemalariga qo'yilgan, lekin siz Windows ga ham o'rnatishingiz mumkin. Microsoft® Windows® 2000 Server SP4 Microsoft® Windows® XP Professional SP2 Microsoft® Windows® XP Home Edition SP2 Microsoft® Windows Server™ 2003 va boshqalar.
<b>RAM-Tezkor xotira</b>	Minimum: 256 MB
<b>Hard Disk- Qattiq disk</b>	Minimum: 600 Mb li bo'sh joy bo'lishi talab etiladi.
<b>CD yoki DVD Drive</b>	Talab qilinmaydi
<b>Display-Monitor</b>	Minimum: 800 x 600 256 rangli monitor tavsiya etiladi:

	1024 x 768 High Color - 16-bit
<b>Sichqoncha</b>	Microsoft sichqonchasi

### 3.2 Dasturiy tizimdan foydalanish qo'llanmasini yaratish

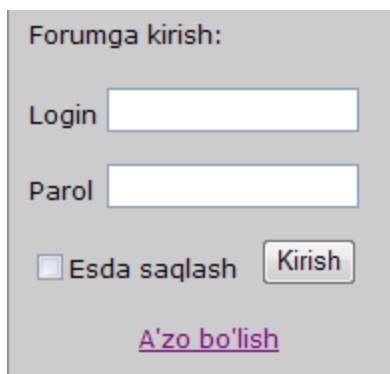
Microsoft firmasi PHP dasturlash tili, MySQL ma'lumotlar bazasi va Apache serverni birlashtirib (ya'ni PHP + MySQL + Apache) denver ishlab chiqqan. Mana shuni o'z kompyuteringizga o'rnatasiz. O'rnatganingizdan keyin operatsion tizim o'rnatilgan diskda (odatda C diskda) WebServers papkasi hosil bo'ladi va virtual disk yaratiladi. Bu papkada etc, home, tmp, usr deb nomlangan papkalar hosil bo'ladi. etc papkasidan Run.exe orqali server ishga tushiriladi. Endi brauzerni ochib adreslar satriga php.uz deb yozasiz. Mana endi siz dasturni ko'rishingiz mumkin.

Dasturning umumiy ko'rinishi quyidagicha:



### 3.2.1 – rasm. Dasturning umumiy ko'rinishi.

Dasturning asosiy menyusi Bosh sahifa , Maqolalar, Funksiyalar, MySQL, Test va Forum qismlaridan tashkil topgan. Dasturning chap qismida qo'shimcha menyular joylashgan bo'lib, unda bo'limlar va bo'limlar bo'yicha mavzular joylashtirilgan. U yerdan o'zingizga kerakli bo'limni va mavzuni darhol topishingiz mumkin. Dasturning o'ng qismida forumga kirish va unga a'zo bo'lish, ovoz berish bo'limlari mavjud.



Forumga kirish:

Login

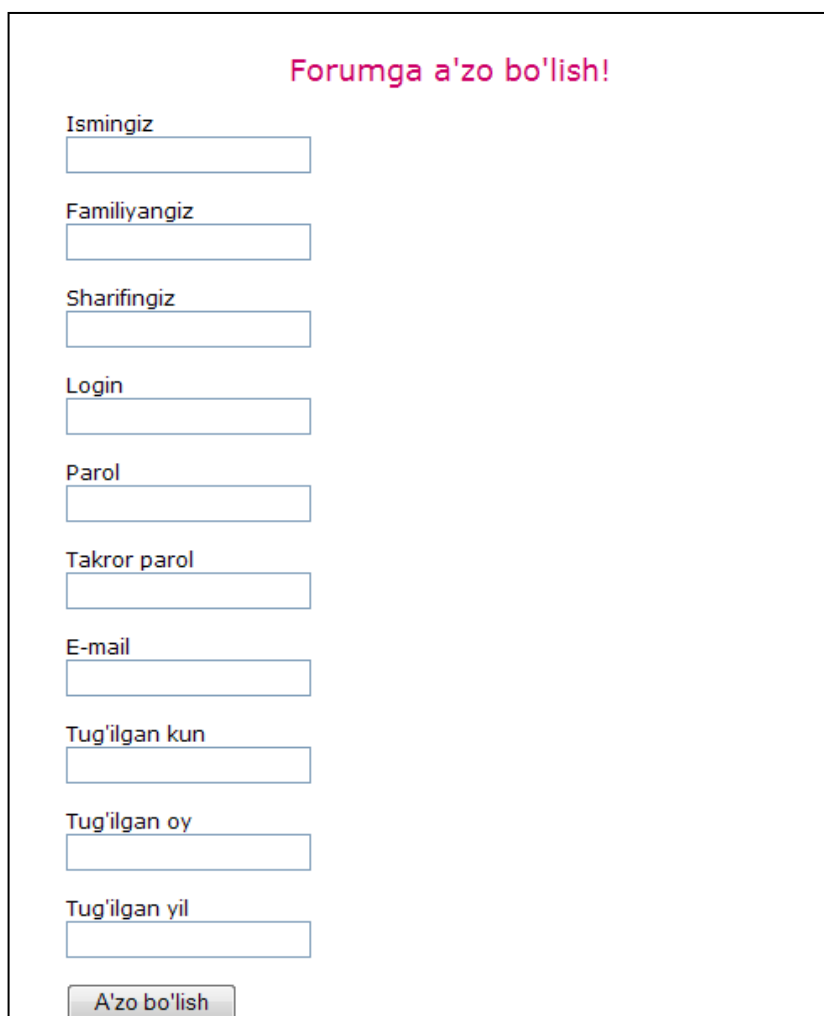
Parol

Esda saqlash

[A'zo bo'lish](#)

Ushbu oynada siz avvaldan forumda ro'yxatdan bo'lsangiz login va parolingizni yozib forumda ishtirok etishingiz mumkin. Agar a'zo bo'lmagan bo'lsangiz [A'zo bo'lish](#) yo'naltirgich(ссылка)ini bossangiz sizga quyidagi oyna chiqadi.

### 3.2.2 – rasm. Forumga kirish darchasi



**Forumga a'zo bo'lish!**

Ismingiz

Familiyangiz

Sharifingiz

Login

Parol

Takror parol

E-mail

Tug'ilgan kun

Tug'ilgan oy

Tug'ilgan yil

Siz bu yerdagi anketani to'ldirib forumga a'zo bo'lishingiz mumkin. Shundan so'ng forumdagilar bilan bermalol muloqotda bo'lib, o'z mulohazalaringizni qoldirishingiz mumkin.

### 3.2.3 – rasm. Forumga a'zo bo'lish darchasi

Sizning mulohazangiz

Mulohaza matni:

Qo'shish

### 3.2.3 – rasm. Forumga mulohaza qoldirish darchasi

Dasturda foydalanuvchilarning mana shu o'quv qo'llanmadan olgan bilimlarini tekshirish maqsadida test tashkil qilingan. Test yechish uchun ham a'zo bo'lish talab etiladi.

Testga kirish:

Login:

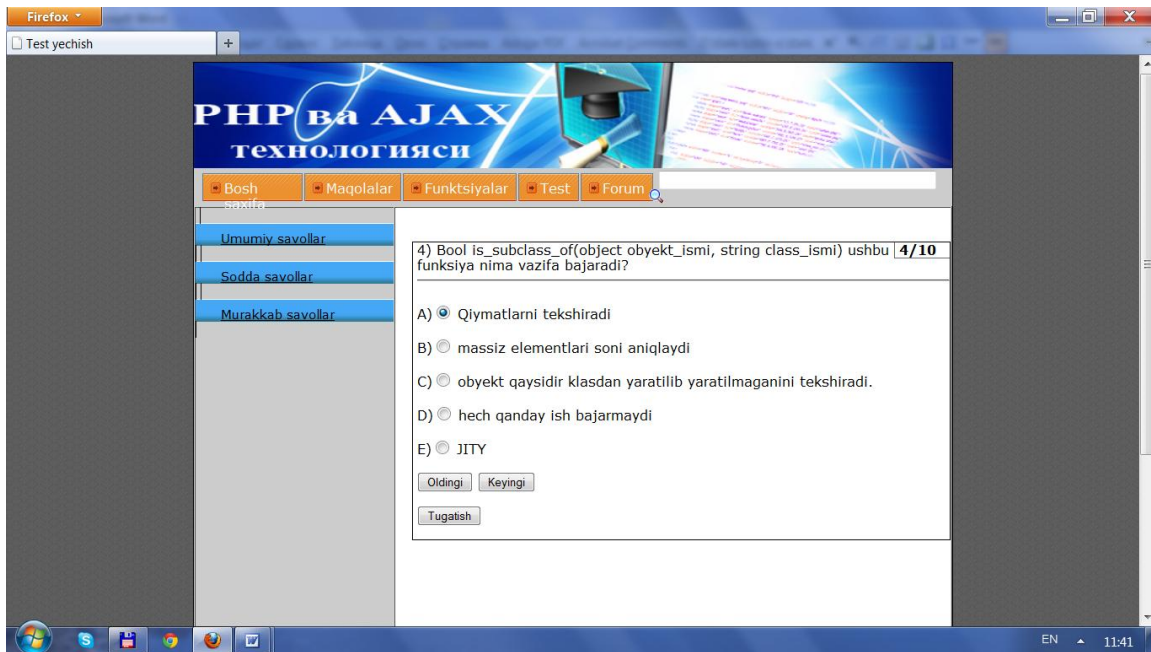
Parol:

Kirish

[Ro'yxatdan o'tish](#)

### 3.2.6 – rasm. Testga kirish darchasi

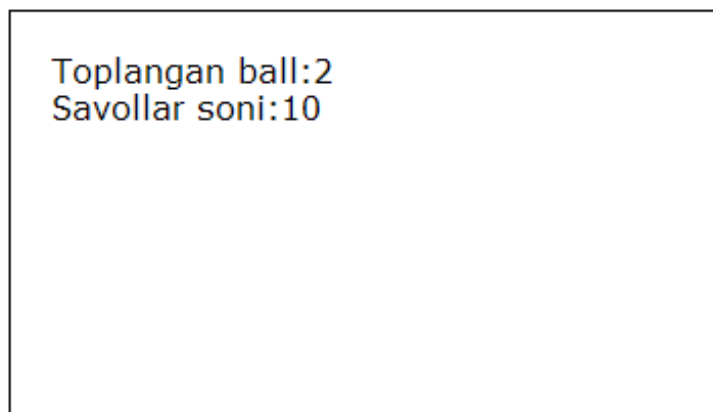
Login va parolingizni yozib kirganingizdan keyin sizda test yechish sahifasi paydo bo'ladi:



3.2.7 – rasm. Test yechish sahifasi

Bu oynada test savollari tartibi bilan beriladi. Siz bunda variantlar orasidan birortasini belgilab “Keyingi “ tugmasini bosasiz. Shunda sizga keyingi savol keladi. Yoki “Oldingi ” tugmasini bosgan holda siz avvalgi savolga qaytib boshqa variantni ham belgilashingiz mumkin.

Testlarni yechib bo’lib “Tugatish” tugmasini bossangiz shunda sizga natijani chiqarib beradi.



3.2.8 – rasm. Test natijasini ko’rish darchasi

## Xulosa

Ushbu bobda dasturiy tizimga qo’yiladigan talablar aytib o’tilib, dastur interfeysi va foydalanish yo’riqnomasi ishlab chiqildi.

## **IV . Mehnat muhofazasi va texnika xavfsizligi**

### 4.1 Shaxsiy kompyuterda ishlashda texnika xavfsizligiga rioya qilish

Kompyuter bilan ishlaganda quyidagi qoidalarga amal qilish kerak:

#### 1. Ishdan oldin:

- ish joyini to'g'ri tashkil qilish;
- kuchlanish kabellarini, ulovchi o'tkazgichlarni, rozetka, shtepsel vilkalarini sozligini tekshirish;
- mashinani shunday o'rnatish kerakki, tebranish(vibratsiya) va chayqalish turlari bartaraf etish imkoni bo'lsin;
- mashinaning hamma bloklarini sozligini va kompyuterni butun ishga tayyorlik darajasini tekshirish;

#### 2. Ish mobaynida:

- kompyuterda begona shaxslarni ishlashiga yo'l qo'ymaslik;
- begona buyumlarni mashinaning ishlayotgan bloklariga tushib qolmasligini nazorat qilib turish;

#### 3. Ishdan so'ng:

- ish joyini tartibga keltirish;

Ish mobaynida quyidagilar taqiqlanadi:

- kiyimning yuqori cho'ntaklarida begona metall buyumlar, ayniqsa o'zining tarkibida magnitlangan temirga ega bo'lgan buyumlarni saqlash;
- malakasi ikkinchi toifadan past bo'lgan shaxslarni xonada bir o'zi ishlashi;
- mashinaning tok oqib o'tadigan qismlariga teginish;
- manbaga ulangan kompyuterdan platalarni olish va qo'yish, shuningdek ajralishli yoki bog'lanishli ulanishlarni ajratish yoki ulash;
- tarmoqqa ulangan kompyuterning ustki qopqog'ini echish;
- axborotni tasvirlovchi modul (monitor) qopqog'ini ochish, chunki electron - nur trubkaga yuqori kuchlanish beriladi;

-ventilyatsion tirqishlarni berkitish;

-malakaviy mas'uliyatga ega bo'lmagan foydalanuvchilarga kompyuterda paydo bo'lgan nosozliklarni sozlashga ruxsat berish;

-boshqa nominalga ega bo'lgan saqlagichlarni ishlatish;

-nosoz mashinalar bilan ishlash;

-ish vaqtida uzoq tanaffuslarga mashinani yoqiq, holda tashlab ketish;

Kiritish chiqarish qurilmalari (monitor, klaviatura, va sichqoncha) ergonomik ekspluatatsiya va sog'liqni asrash uchun mo'ljallangan. Shuningdek, kiritish chiqarish qurilmalarini nosoz ishlashi, qo'lni noto'g'ri qo'yish yoki to'g'ri o'tirmaslik, uzoq vaqt uzluksiz klaviatura yoki sichqoncha orqali axborot kiritish inson tanasining zo'riqishiga va charchashiga olib kelishi mumkin. Ko'zni zo'riqishiga, mushaklarda asab tizimini susayishiga, bo'g'implarda og'riq paydo bo'lishiga ham sababchi bo'ladi. Bu holat quyidagi maslahatlarga amal qilinganda sezilarli darajada kamayishi mumkin.

Ish joyida monitor shunday joylashishi kerakki, yorug'lik iloji boricha yuqoridan burchak ostida tushishi kerak. Monitor derazaga nisbatan yoni bilan turishi, ish stoli esa, yorituvchi qurilmalar orasida turishi kerak. Ko'zni zo'riqtiradigan, to'g'ri (ko'rish maydoniga tushayotgan yorug'lik manbai) va qaytgan (ekrandan qaytgan yorug'lik) yorug'likdan iloji boricha himoyalani kerak. Buning uchun ish joyini o'zgartirish kerak. Monitordagi tasvirning yoritilganligi, ravshanligi va kontrasti bajarilayotgan ishga va xonaning yoritilganligiga bog'liq holda sozlanishi kerak. Monitor, hujjatlar va klaviatura shunday joylashishi kerakki, ularni yorug'lik manbasiga nisbatan joylashganligiga bog'liq bo'lgan yuzalaridagi ravshanlik 1:10 nisbatdan oshib ketmasligi kerak (1:3 nisbat tavsiya etiladi).

Klaviatura va sichqonchani shunday joylashtirish zarurki, ularning ekspluatatsiyasida sezilarli kuchlanishlar paydo bo'lmasligi kerak. Klaviaturani tananing yuqori qismiga nisbatan parallel ravishda, sichqonchani esa, aynan klaviatura sathida joylashtirish kerak. Stol cheti bilan klaviatura orasi 5—10 sm

bo'lishi kerak. Monitorni shunday joylashtirish kerakki, undagi tasvirni boshni yoki tanani burmasdan ko'rish mumkin bo'lsin. Monitorgacha bo'lgan masofa 70 sm dan oshmasligi zarur (optimal masofa 30 sm). Hujjatlashtirish qurilmalari va tez—tez ishlatib turilmaydigan texnik vositalar operatoridan o'ng tomonda, maksimal erishish zonasida joylashtirish, aloqa vositalari esa chap tomonda o'ng qo'lni yozish uchun bo'sh qoldirish maqsadida joylashtirish tavsiya etiladi. Hujjatgacha bo'lgan masofa huddi monitor uchun mo'ljallangan masofadek bo'lishi kerak. Stol yoki stulning balandligi operatorning bo'yiga qarab moslashtiriladi. Stulning suyanchig'i umurtqa pog'onasini bel qismiga tegib turadigan va operator to'g'ri o'tira oladigan bo'lishi kerak.

O'tirganda oyoqlarning bukilgan burchagi  $90^\circ$  va undan yuqorini tashkil qilishi zarur. Axborot kiritish vaqtida qo'lning tirsakdan to kaftgacha bo'lgan qismi pol bilan parallel bo'lishi, elkalar esa erkin holda bo'lishi kerak.

Monitor tasvirning yuqori chekkasi ko'z chizig'ida bulishi zarur.

Bu maslahatlarning hammasi kompyuter bilan ishlashda tana a'zolarini toliqtirmasdan, zo'riqtirmasdan ishlashini ta'minlaydi.

#### 4.2 Elektr xavfsizligi shartlari tahlili

Elektr tarmoqlari va elektr qurilmalarini 2 guruhga bo'lish qabul qilinadi:

- 1 guruh - 1000 V gacha kuchlanishli
- 2 guruh - 1000 V dan yuqori kuchlanishli

Elektr tarmoqlari tokli quvirlar soni bo'yicha quyidagilarga bo'linadi:

Bir o'tkazgichli, 2 o'tkazgichli, 3 o'tkazgichli, 4 o'tkazgichli.

Bir o'tkazgichli tarmoqda reles yoki yer ikkinchi o'tkazgich sanaladi. Bu sxema bo'yicha tramvaylar, elektrovozlar, gohida payvandlash qurilmalari ishlaydi.

2 o'tkazgichli tarmoqlari - doimiy va o'zgaruvchan tok 1 fazali tarmoqlari.

3 o'tkazgichli o'zgaruvchan tok tarmoqlari - izolyatsiyalanadi yoki yerga tutash neytralli 3-fazali tok tarmoqlari.

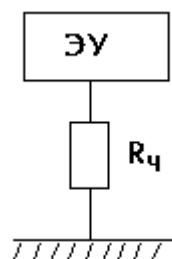
4 o'tkazgichli tarmoqlari - yerga tutash neytralli va nolli o'tkazgichli 3-fazali tok tarmoqlari.

Elektr xavfsizligining barcha tizimlari va elektr qurilmalari shartlari tahlilida yerga nisbatan 1000 V gacha hajmli fazada  $S_k = 0$  bo'lganda,  $X_c = \frac{1}{\omega C} = \infty$  deb hisoblanadi.

1 km dan oshmagan kabel liniyalarining umumiy uzunligidagi tarmoqlar 1000 V gacha kichik hajmli fazali U tarmoqlarga tegishlidir.

> 1000 V kuchlanishli havo tarmoqlar va  $\square$  1000 V kuchlanishli kabel tarmoqlarida 1-2 km dan ortiq liniya uzunligida isrof bo'ladigan toklar va hajmi toklar bilan hisoblashish zarur.

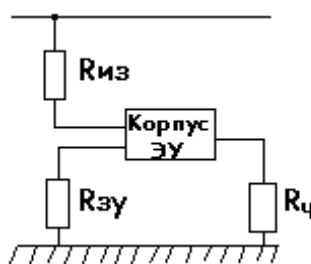
Kuchlanish ostidagi elektr qurilmalarning ishi chog'ida elektr toki urishining 2 xarakterli jihati bor:



4.2.1 – rasm. Elektr qurilmasining tok boradigan qismlariga tegib ketish

2. Insonning kuchlanish ostida normal bo'lmagan, biroq izolyatsiya uzilib qolishi holatida, kuchlanish ostiga tushib qolish holatida uskuna korpusiga tegib ketishida.

- $R_{из}$  – Izolyatsiya qarshiligi;
- $R_{з\у}$  – zazemliniya;
- $R_ч$  – inson tanasining qarshiligi;
- ЭУ – Elektr qurilmasi

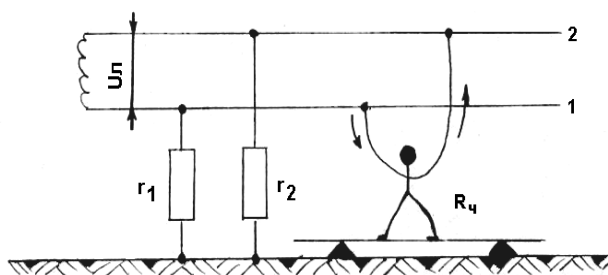


4.2.2 – rasm. Uskuna korpusiga tegib ketish

2 turdagi tegib ketish bor:

- ikki qutbo'li;
- bir qutbo'li.

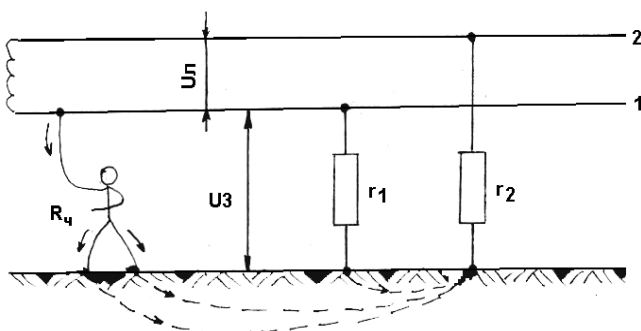
Insonning 1 fazali tarmoqqa ikki qutbli tegishi. Bunday holatda inson tanasi orqali qo'l -qo'l yo'li bo'yicha o'tadigan tok hajmi  $R_i$  inson tanasi qarshiligi va berilgan kuchlanish bilan belgilanadi.



4.2.3 – rasm. Insonning 1 fazali tarmoqqa ikki qutbli tegishi

$$I_u = \frac{U_{\text{н}}}{R_u} (A)$$

Tarmoq izolyatsiya qarshiligi, poyafzal, dielektrik qo'lqoplar va podstavkalar bunday tegib ketishda, insonni shikastlanishdan himoya qilolmaydi. Amaliyotda ko'pincha bir qutbli tegib ketish ko'p uchraydi.



4.2.3 – rasm. Bir qutbli tegib ketish

$$r_1 = r_2 = r_{iz}$$

Insonning yerdan izolyatsiya qilingan 2-o'tkazgichli tarmoq 1-o'tkazgichiga tegib ketishi. Insonning tok boruvchi qismlar (o'tkazgichlarga) tegib ketishigacha,

agar  $r_1 = r_2$  xar bir o'tkazgichning kuchlanishi yerga nisbatan  $\frac{1}{2}U_1 = U_e$ ,  $U_1 = 220V$ , unda  $U_e = 110V$  ni tashkil qiladi.

1 qutbli tegib ketishda inson  $r_1$  ga va keyin  $r_2$  ga ulangan 1-o'tkazgichga tegadi. Tegib ketgandan so'ng o'tkazgichlar kuchlanishining yerga nisbatan qayta taqsimlash sodir bo'ladi, zero, birinchi o'tkazgich bilan yer o'rtasida qarshilik kamayadi, ya'ni

$$I_u = \frac{U_n}{2R_u + r_{u3}}; \quad I_u = \frac{220}{2 \cdot 10^3 + 5 \cdot 10^3} = 0,031A$$

2 o'tkazgichli tarmoqqa 1 qutbo'li tegishida inson yerga nisbatan Riz tarmoq izolyatsiya ximoyasida bo'ladi.

1 o'tkazgichga tegayotgan inson ( $r_2 = 0$ ) 2-o'tkazgichning yerga qadalib qolishida to'liq hoshiyali kuchlanish ostiga kirib qoladi, ya'ni

$$I_u = \frac{U_n}{R_u} = \frac{220}{1000}$$

Birinchi o'tkazgichga daxldorlik 2 qutbli tegishdagidek xavfli bo'lib chiqishi mumkin, ya'ni oqibati o'lim bilan tugashi mumkin.

Ko'pincha elektr qurilmaga xizmat ko'rsatuvchi personal polda turib 1-qutbga tegadi (yalang'och o'tkazgich, kiskich). Uning qarshiligi inson tanasi bilan bog'liq tarzda bo'ladi.

Ushbu bobda texnika xavfsizligiga bag'ishlangan bo'lib, unda shaxsiy kompyuterda ishlashda texnika xavfsizligiga rioya qilish va elektr xavfsizligi shartlari tahlili to'g'risida ma'lumotlar berildi.

## Xulosa

Hozirgi kunda O'zbekistonda Web dasturlashga bo'lgan talab juda yuqori. Shuning uchun web texnologiyasi sohasida faoliyat yurituvchi mutaxassislarining malakasini oshirish va shu sohaga qiziquvchilari o'rgatish masalasi oldinga qo'yiladi.

Bunda web dasturlash tillarini o'rgatuvchi maxsus elektron darsliklar ham katta amaliy ahamiyatga ega. Yaratilgan elektron darslik albatta oddiy ko'rinishda emas, balki o'rganuvchi bilimni tekshirib nazorat qilib borishi ham kerak. Buning uchun maxsus test dasturlari, krasvordlar, o'yinlar nazariyasi, sinov savollarini kiritish va natijalarini tahlil qiluvchi dasturlarni kiritish lozim. Shuning uchun ushbu elektron o'quv qo'llanmada test va forum tashkil qilindi. Bu qo'llanmadan o'rta-maxsus va oliy ta'lim o'quv muassasalari talabalari hamda shu sohaga qiziquvchilar ham foydalanishlari mumkin.

Ushbu elektron o'quv qo'llanmani yaratish mobaynida quyidagi natijalarga erishildi:

1. Foydalanuvchilar dasturlash texnologiyasi haqida o'zbekcha ma'lumot olishlari mumkin bo'ldi.
2. Dasturlash texnologiyasi ma'lumotlari ma'ruza va laboratoriyalari haqida ma'lumotlar olishlari mumkin.
3. Foydalanuvchilar mana shu o'quv qo'llanmadan olgan bilimlarini test orqali tekshirish imkoniga ega bo'ldilar.
4. O'rganuvchilar o'zlari qiziqqan va bilmagan savollariga javob olishi uchun forum tashkil qilindi.
5. Qulay interfeys va qo'llanmadan foydalanish yo'riqnomasini yaratildi.

Qo'llanmada Dasturlash texnologiyasi haqida boshlang'ich va umumiy ma'lumotlar berildi. Kelajakda bu qo'llanmaning to'liq va kengaytirilgan versiyalarini ishlab chiqarish ham mumkin.

Ushbu elektron darslik o'z bilim va mahoratingizni mustahkamlashda, kelajakda yaxshi web loyihalarni amalga oshirishingizda sizga yordam beradi.

## FOYDALANILGAN ADABIYOTLAR

1. Karimov I. A. “Jahon moliyaviy-iqtisodiy inqirozi, O'zbekiston sharoitida uni bartaraf etishning yo'llari va choralari “ O'zbekiston, 2009. - 56 b.
2. Qosimov S. S “Axborot texnologiyalari” texnika OO'Yu talabalari uchun o'quv qo'llanma - Toshkent 2006. - 273 b.
3. Alimov S. “PHP davrasida” Toshkent 2006. – 85 b.
4. Котеров Д. В. «Самоучитель PHP4» БХВ-Петербург, 2001. — 572 с
5. Кузнецов М. В. «PHP 5 на примерах» БХВ-Петербург, 2005. — 577 с
6. Мазуркевич А. М. «PHP настольная книга программиста» Минск 000 Новое Знание 2003 – 480 с.
7. Орлов А.А.«PHP полезные приемы»  
Горячая линия – Телеком, 2004. – 225 с
8. Томсон Лаура «Разработка Web-приложений на PHP и MySQL»  
ДиаСофтЮП, 2003. — 672 с

### **Internet tarmog'idagi manbalar:**

1. [www.google.co.uz](http://www.google.co.uz)
2. [www.php.su](http://www.php.su)
3. [www.php.net](http://www.php.net)
4. [www.ziyonet.uz](http://www.ziyonet.uz)
5. [www.library.tuit.uz/konspekty.htm](http://www.library.tuit.uz/konspekty.htm)