

O'ZBEKISTON ALOQA VA AXBOROTLASHTIRISH AGENTLIGI
TOSHKENT AXBOROT TEXNOLOGIYALARI UNIVERSITETI

Himoyaga.
Kafedra mudiri

_____2010 y

**««Klient-server» texnologiyasi asosida ma'lumotlarga ishlov berish tizimini
yaratish» mavzusida**

**Bakalavrning
bitiruv malakaviy ishi**

Bitiruvchi: Abduraxmonov K.R

Raxbar: Raxmatov F.A.

Toshkent

O'ZBEKISTON ALOQA VA AXBOROTLASHTIRISH AGENTLIGI
TOSHKENT AXBOROT TEXNOLOGIYALARI UNIVERSITETI

Fakultet: Axborot texnologiyalari
Kafedra: Kompyuter tizimlari
Yo'nalish (mutaxassislik): 5811200 – Servis (axborot servisi)

TASDIQLAYMAN
Kafedra mudiri _____
“ ____ ” _____ 2010 y.

Abduraxmonov Karim Raxmatullaevich

bitiruv malakaviy ishiga

T O P S H I R I Q

1. Ish mavzusi: «Klient-server» texnologiyasi asosida ma'lumotlarga ishlov berish tizimini yaratish».
2. «____» _____ 20__ yil ____ - sonli buyruq bilan tasdiqlangan.
3. Ishni himoyaga topshirish muddati: 31.05.10 y.
4. Ishga oid dastlabki ma'lumotlar: kompyuter qurilmalari xaqida ma'lumotlar, web-ilovalar yaratish bo'yicha kitob va boshqa ilmiy-texnik adabiyotlar.
5. Hisoblash-tushuntirish yozuvlarining mazmuni (ishlab chiqiladigan masalalar ro'yxati): HTML dashturlash tili. Dinamik saytlarni yaratish texnologiyasi (PHP) . Web texnologiyalari imkoniyatlari oshirish vositalari (Java, Java script). MS SQL ma'lumotlar bazasini boshqarish tizimi. ORACLE ma'lumotlar bazasini oshkarish tizimi va uning avzalliklari. My SQL ma'lumotlar boshqarish tizimidan foydalanishning ahamiyati. Tizimning tafsifi. Foydalanuvchiga qo'llanma. Dastur ishlash uchun qo'yiladigan talablar. Texnika xavfsizligi va mehnat muxofazasi. Xulosa. Foydalangan dadiyotlar. Ilova.
6. Grafik materiallar ro'yxati: Ma'ruza bo'yicha prezentatsion slaydlar.
7. Topshiriq berilgan sana: 15.01.10 y.

Rahbar _____
(imzo)

Topshiriqni oldim _____
(imzo)

8. Ishning ayrim bo'limlari bo'yicha maslahatchilar

Qism	Maslahatchi o'qituvchi F.I.O	Imzo, sana	
		Topshiriq berildi	Topshiriq olindi
Asosiy qism Xayot faoliyati hafsizligi	Raxmatov F.A.		

9. Ishni bajarish grafigi

T/r	Ish qismlari nomi	Bajarish muddati	Rahbar (maslahatchi) belgisi
1	Web texnologiyalar taxlili.	15.01.10-26.02.10	
2	Internetda MB ni tashkil etish.	27.02.10-01.04.10	
3	“IT-prise” tizimini yaratish.	05.04.10-10.05.10	
4	Xayot faoliyati hafsizligi	11.05.10-15.05.10	
5	Prezentatsiya tayyorlash	17.05.10-31.05.10	

Bitiruvchi _____ «____» _____ 2010 yil
(imzo)

Rahbar _____ «____» _____ 2010 yil
(imzo)

Bitiruv malakaviy ishida «Klient-server» texnologiyasi asosida ma'lumotlarga ishlov berish tizimi ya'ni orgtexnikaga oid ma'lumotlarini yig'sh, saqlash, saralash va ularni aks ettirish imkonini beruvchi «IT-PRICE» tizimi yaratilgan.

В выпускной квалификационной работе разработана система «IT-PRICE», которая позволяет обеспечить, сохранение, отбор и отображение информации по средствам оргтехники.

In final qualifying work system "IT PRICE" which allows to provide, preservation, selection and information display on organizers is developed.

MUNDARIJA

Kirish.....	6
1. WEB TEXNALOGIYALAR TAXLILI	
1.1 HTML dashturlash tili.....,,.....	7
1.2. Dinamik saytlarni yaratish texnologiyasi.....	19
1.3. Web texnologiyalar imkoniyatlarini oshirish vositalari.....	32
2. INTERNETDA MA'LUMOTLAR BAZASINI TASHKIL ETISH	
2.1. MS SQL ma'lumotlar bazasini boshqarish tizimi.....	47
2.2. ORACLE ma'lumotlar bazasini boshkarish tizimi va uning avzalliklari	58
2.3. My SQL ma'lumotlar boshqarish tizimidan foydalanishning axamiyati.....	59
3. "IT-PRISE" DASTURIY TIZIMINI YARATISH	
3.1. Tizimning tafsifi.....	69
3.2 Foydalanuvchiga qo'llanma.....	70
3.3. Dastur ishlash uchun qo'yiladigan talablar.....	75
4. XAYOT FAOLIYATI HAFSIZLIGI	
4.1 Vibratsiya(titrash) ishlab chiqarishdagi zararli omil sifatida.....	77
4.2 Favkulodda vaziyatlar.....	82
Xulosa.....	90
Foydalangan adabiyotlar.....	91
Ilova.....	92

Kirish

Bugungi kunda axborot texnologiyalari kirib bormagan biror bir oxani ko'rsatib o'tish juda mushkul. Ayni vaqtda ilm-fan, texnika taraqqiyotini innovatsion texnologiyalar, web dasturlashsiz tasavvur etib bo'lmaydi. Ayni paytda ushbu texnologiyalarning ta'lim tizimiga kirib kelishi sezilarli yutuqlarni vujudga keltirdi. Bu esa o'z-o'zidan ham ma'naviy, ham moddiy yutuqlarga asos bo'ladi.

Vazirlar mahkamasining 2001 – yil 23 – maydagi “Internetning xalqaro axborot tizimlariga keng kirib borishini ta'minlash dasturini ishlab chiqish tashkil etish chora tadbirlari to'g'risida” gi qarori, O'zbekiston Respublikasi Prezidentining 2002 – yil 30 – maydagi “Kompyuterlashtirish va axborot texnologiyalarini rivojlantirish to'g'risida” gi farmoni va korxonalarni modernizatsiya qilish texnik va texnologik qayta jihozlashni yanada jadallashtirish, zamonaviy, moslashuvchan texnologiyalarni keng joriy etishdan ko'rinib turibdiki hozirgi kunda fuqoralarimizning internetdan keng ko'lamda foydalanishiga imkon beradi.

Internet - sayyoramiz miqyosidagi kompyuter tarmoqlarining birlashmasidir. Agar kosmik stantsiya a'zolarining E-mail dan foydalanishini inobatga olsak internetning sayyoramiz sarhadlaridan chiqqanini ham ko'rishimiz mumkin.

Hozirga kelib Internet uchun web ilovalar har xil turdagi axborotlar juda tezkor ravishda va hohlagan paytda hohlagan qiziquvchiga yekazish imkonini beradi.

Web dasturlashning juda jadallik bilan tarqalishiga turtki bo'lgan yana bir manba bu uning tekin ravishda tarqatilishi va undan foydalanish juda soddaligi deb aytish mumkin.

Web dasturlash bundan buyon rivojlanishda davom etib o'zida shu davr talablariga mos yechimlarni mujassamlashtirgan holda faqat oldinga qarab qadam bosadi va kompyuter insoniyatga xizmat qilguncha xizmat qiladi.

1. Web texnologiyalar taxlili

1.1. HTML dashturlash tili

Butun dunyo o'rgimchak to'ri WWW (World Wide Web) - bu ko'p sonli o'zaro bir-biriga bog'langan hujjatlardir. Web sahifada boshqa Web sahifalar bilan bog'lovchi chekli gipermurojaat bo'ladi. Gipermurojaat mexanizmini bir necha so'z bilan izohlash qiyin, lekin kamida bir marta Web sahifaga kirib ishlagan odam buni darhol tushunadi.

Gipermurojaat butunjahon o'rgimchak to'ridan hujjatning URL deb nomlanuvchi manzilini qidirishda ishlatiladi. Biz bilamizki Web sahifaning asosiy mohiyati axborotni taqvirolash va uni har bir foydalanuvchi uchun ochib berishdir. Bunda bir qancha funktsional cheklanishlar mavjud, ya'ni biz oldindan Web sahifani ko'ruvchining kompyuteri qanday, monitorining imkoniyati qanday va ko'rish oynasining qanday o'lchamda o'rnatilganligini bilmaymiz. Biz hattoki foydalanuvchining qanday operatsion muhitda ishlayotganini ham bilmaymiz. Web sahifa esa INTEL mashinada ham Makintoshda ham bir xil ko'rinishda bo'lishi kerak. Foydalanuvchi o'z operatsion muhitida qanday yozuv shriftlar o'rnatganligi Yoki uning videokartasi qancha rangni ko'tara olishi ham malum emas. Yuqoridagi malumotlarning yo'qligi Yoki etarli emasligi umumaxborot almashish tilining paydo bo'lishiga to'siq bo'lishi kerak edi, lekin bunday bo'lmadi.

Gap shundaki 1986 yili xalqaro standartlashtirish tashkiloti (ISO) tomonidan qog'ozdagi hujjatni ekranda tasvirlash imkonini beruvchi hamma variantlarni etiborga olgan SGML (Standart Generalized Markup Language) tili yaratildi. Hamma imkoniyatlarni etiborga olgan kuchli sistema ishlab chiqildi.

Web sahifa uchun bu yaxshi yo'l deb hisoblangan edi. Lekin bu til qonun qoidalari izohi yuzlab sahifalarni egalladi. Bunday til asosida yaratilgan sahifani tasvirlovchi dasturni ishlab chiqish uchun juda ko'p vaqt ketgan bo'lardi, shuning uchun Internet ehtiyoji uchun SGML tilining malum bir qismi tanlab olindi va HTML (Hyper Text Markup Language) deb nomlandi. HTML tilida yozilgan Web sahifa faylida axborotning qanday tasvirlanishi haqidagi yo'l yo'riqlar ketma – ket

joylashgan bo'ladi. Bu fayl oddiy matnli fayl bo'lib uni maxsus brauzer dasturlarsiz o'qish qiyin. Grafik rasmlarni esa umumman ko'rib bo'lmaydi chunki uning o'rnida brauzer dasturga biror rasm kerakligini ko'rsatuvchi TEG turadi. Agar biz o'z Web sahifamizni yaratmoqchi bo'lsak, albatta HTML tilini yaxshi bilishimiz kerak bo'ladi.

HTML da yozilgan Web sahifa odatda brauzer deb nomlanuvchi maxsus dastur yordamida quriladi. Bu so'zning o'zbekchaga to'g'ridan – to'g'ri tarjimai «sharxlovchi dastur» manosini beradi. Brauzerning asosiy vazifasi foydalanuvchi talabiga muvofiq manzilni internetdan topish va qiyinchiliksiz uni tasvirlashdir. Oldiniga brauzer HTML tilida yozilgan «instruktsiyani» tahlil qiladi va bu instruktsiya yordamida Web sahifada mavjud bo'lgan axborotni tasvirlaydi.

Hozirgi kunda eng ko'p ishlatiladigan 2 ta Web brauzer mavjud bo'lib Web sahifa tayorlovchilar bu brauzerlarning xususiyatlarini e'tiborga olishlari kerak. Bular "Netscape Navigator" va "Microsoft Internet Explorer" (IE) brauzerlaridir. Aslida brauzerlar juda ko'p, lekin ular dunyoda ishlatilayotgan brauzerlarning bor - yo'g'i 1% ni tashkil qiladi. Qolgan 75% ni IE, 24% ni esa "Netscape Navigator" egallaydi. Demak shunday ekan ko'pchilik hollarda shu 2 ta brauzer haqida so'z yuritiladi.

Keling endi brauzerlar tarixiga nazar tashlaylik. 1989-yilda SHveytsariyada CERN (The European Laboratory for Particle physics) olimlari tomonidan WWW tizimiga asos solingandan keyin uning axborot manbalarini ochib ko'rish muammosi paydo bo'ldi.

Bu muammoni hal qiluvchi dastlabki dasturlardan biri bu *Lynx* tipidagi satrli brauzerdir. SHunday qilib dastlabki satrli brauzerlar paydo bo'ldi. Vaqt o'tishi bilan brauzerlardan faqat matnlarni emas, balki ovoz, grafika va tasvirlarni tasvirlash ham talab qilindi. Grafik ma'lumotlarni matnli ma'lumotlar kabi ko'rish masalasi birinchi bo'lib Mosaic brauzeri tomonidan amalga oshirilgan .

Mosaic brauzeri UNIX, PC va Macintosh platformalarida ishlashga mo'ljallangan bo'lib bepul brauzerdir. Keyinchalik Mosaic va Silicon Graphics asoschilari birlashib hozirgi kunda etakchi brauzerlardan bo'lgan Netscape

brauzeriga asos soldilar . HTML ning konstruktsiyasi TEG lar deyiladi. Brauzer TEG larni oddiy matnlardan farqlashi uchun ular burchak qavslarga olinadilar. TEG tasvirlash jarayoni hatti harakatlarining boshlanishini bildiradi. Agar bu harakat butun hujjatga talluqli bo'lsa, bunday teg o'zining yopiluvchi juftiga ega bo'lmaydi. Juft teglarning ikkinchisi birinchisining harakatini yakunlaydi. Masalan, har bir Web sahifa <html> tegi bilan boshlanib </html> tegi bilan yopilishi kerak. Etibor bergan bo'lsangiz yopiluvchi teg ochiluvchidan « / » belgisi bilan farq qiladi. Teg nomlari katta Yoki kichik harflar bilan yozilishi mumkin, bularni brauzer bir xil qabul qiladi. HTML tilida boshqa kompyuter tillaridagi kabi izoh berish imkoniyati mavjud. Izoh quyidagi «<- - » va «<- ->» belgilar orsiga yoziladi.

Masalan:

<-- Bu izox -->

Har qanday Web sahifa ikkita qismdan tashkil topadi. Bular sarlavha qismi va asosiy qism. Sarlavha qismida Web sahifa haqidagi malumot joylashadi, asosiy qismda esa Web sahifaning mazmuni bilan tasvirlanish qoidalarini joylashadi. Sarlavha qismi quyidagi ochiluvchi <head> va yopiluvchi </head> teglari orasida joylashadi. Asosiy qism esa <body> va </body> teglari orasida joylashadi. Odatda sarlavha qismi oldidan qo'llanilayotgan HTML standartlari haqida malumot yoziladi. Har qanday Web sahifaning umumiy ko'rinishi quyidagicha bo'ladi:

Misol 1:

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN ">

<html>

<head>

<title>

Hujjat sarlavhasi

</title>

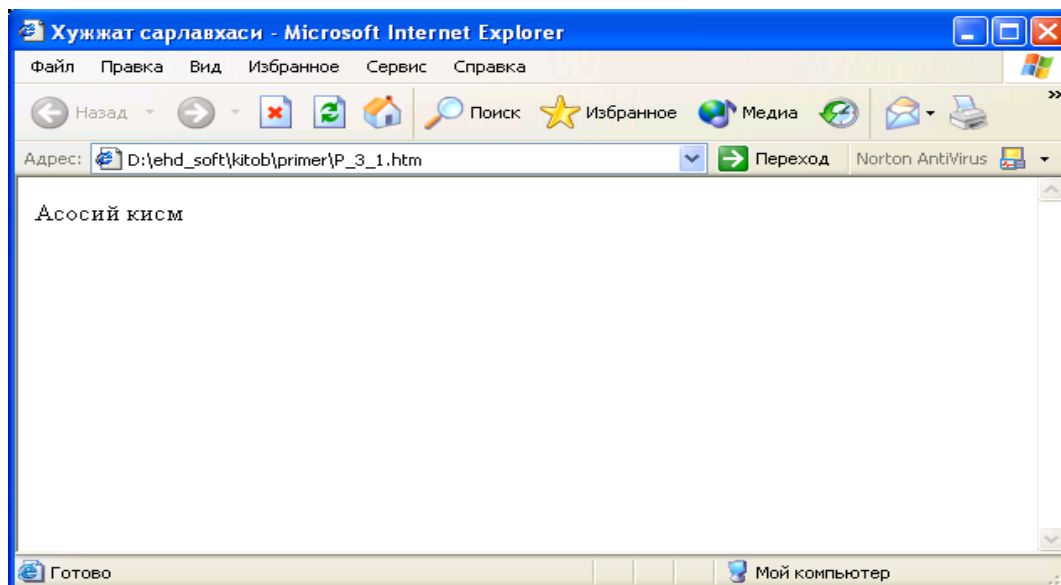
</head>

<body>

Asosiy qism

</body>

</html>



1.1.1-rasm. Hujjat sarlavhasi

Birinchi <Doctype> tegi o'zining parametrlari bilan brauzerga ushbu Websahifani qaysi HTML versiyada yozilganligi haqida malumot beradi.

Web sahifa ishga tushirilganda brauzerning eng yuqori satrida yuklanayotgan hujjat mazmunini anglatuvchi qisqacha yozuv turadi. Bu yozuvni hosil qilish uchun quyidagi ochiluvchi <title> va yopiluvchi </title> teglaridan fodalanamiz.

Misol 2:

<html>

<head>

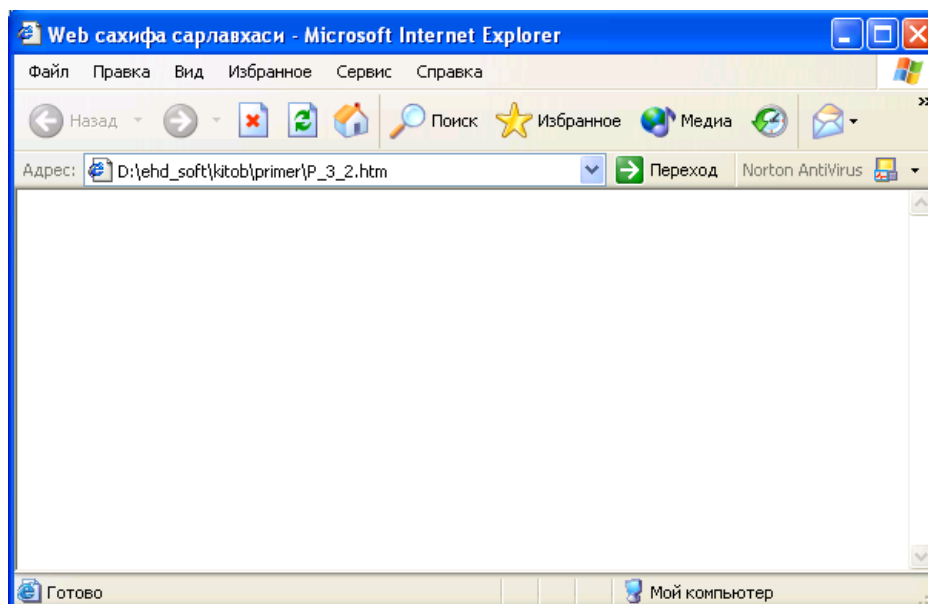
<title>Web sahifa sarlavhasi</title>

</head>

<body>

</body>

</html>



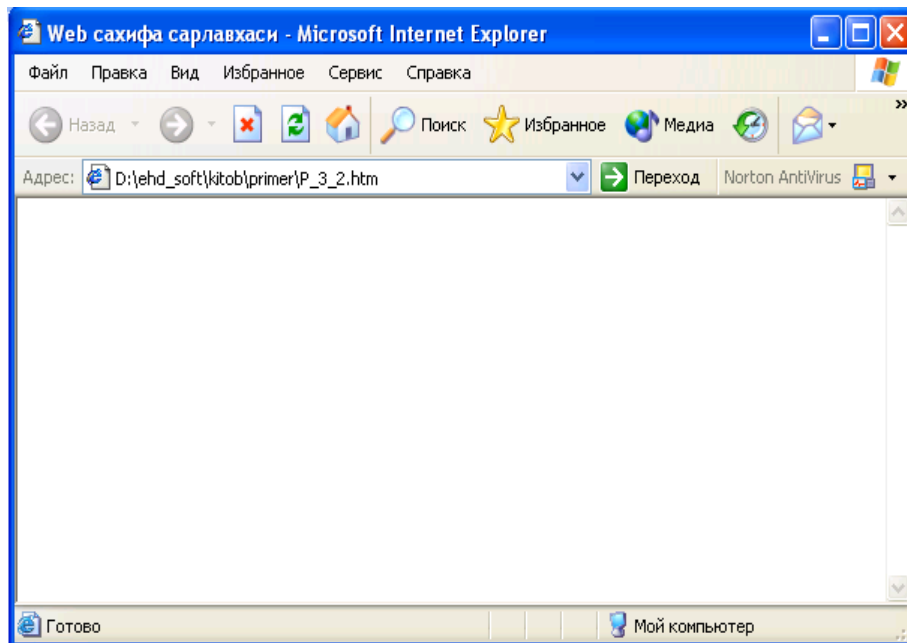
1.1.2- rasm. Web sahifa sarlavhasi

Web sahifaning asosiy qismi `<body>` va `</body>` teglari orasida joylashadi. Bu oddiy matn bo'lishi mumkin. Brauzer bu matnni tug'ridan to'g'ri interpretatsiya qilib ekranda tasvirlaydi. Bizga dastlabki Web sahifamizni yaratish uchun oddiy «Bloknot» matn muharriri kifoya. Quyida ko'rsatilgan misolni matn muharririda yozib, uni xotiraga yozishda kengaytmasini `html` Yoki `htm` deb kiritishimiz kerak.

Misol 3:

```
<html>
  <head>
    <title>Mening birinchi Web sahifam </title>
  </head>
  <body>
    Mening bu sahifamga kiruvchilarga alangali salom
  </body>
</html>
```

Bu faylni ishga tushirish uchun sichqoncha ko'rsatkichini shu fayl ustiga keltirib chap tugmasini ikki marta bosish kerak. Natijada ekranda quydagi ko'rinishdagi natija hosil bo'ladi:



1.1.3-rasm Mening birinchi Web sahifam

`<body>` tegi bir qancha qo'shimcha parametrlarga ega. Bu parametrlar tegning ochiluvchi qismida joylashadi. Parametrlar ikki qismdan iborat bo'ladi: parametr nomi va parametr qiymati. Masalan `bgcolor` parametri tasvirlanayotgan Web sahifa fonining rangini belgilaydi.

Masalan: `<body bgcolor = "green">`

Parametrlarning satrli qiymatlari qo'shtirnoq ichida yoziladi. Biz quyida `<body>` tegining parametrlari bilan tanishamiz.

Background - fon sifatida biror bir grafik tasvirdan foydalanish. Parametr qiymati sifatida grafik tasvir joylashgan manzil (URL) beriladi.

Text - tasvirlanayotgan matn rangi.

Link - Web sahifadagi matnli gipermurojat rangi.

Vlink-foydalanuvchi tomonidan oldin murojat qilingan gipermurojaat rangi.

Alink - foydalanuvchi tomonidan tanlangan gipermurojaat rangi.

Lang – Web sahifa matni yozilgan tilni aniqlash.

Metama'lumotlar: Endi biz metama'lumotlar bilan tanishib chiqamiz. Web sahifalarda meta ma'lumotlarini hosil qilish uchun <meta> tegi ishlatiladi, uning umumiy ko'rinishi quyidagicha:

<meta name="o'zgaruchi nomi" content="o'zgaruvchi qiymati">

Agarda biz Web sahifadagi avtor haqida ma'lumot yozmoqchi bo'sak uni quyidagi ko'rinishda yozish mumkin:

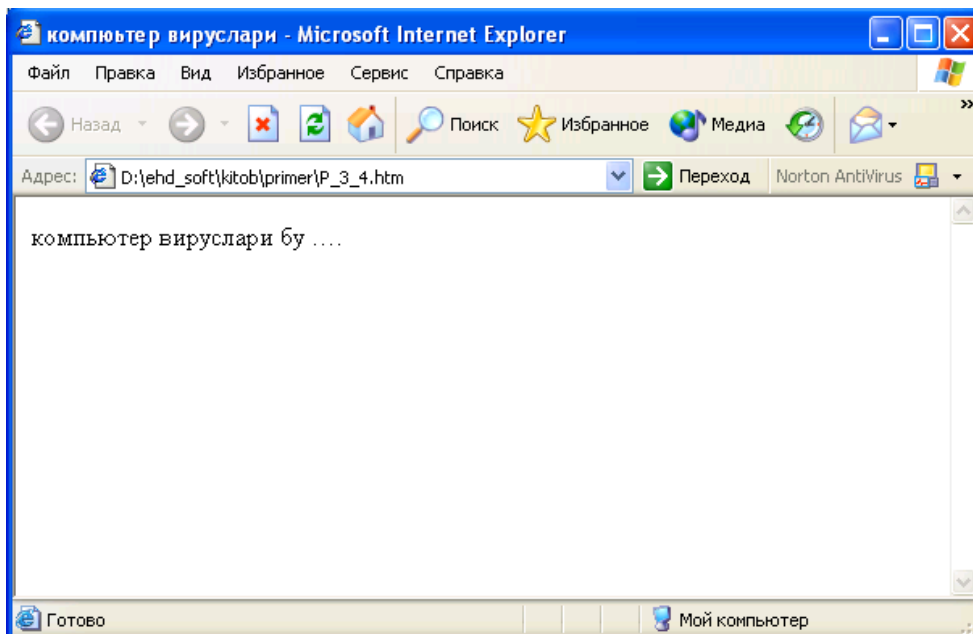
<meta name="Author" content="Bu meniki!!!">

Meta ma'lumotlar asosan Internet da joylashgan qidirish mashinalari uchun zarur. Qidirish mashinalari Web sahifalar haqidagi ma'lumotlarni qaerdan oladi degan savol paydo bo'ladi. Xuddi shu ma'lumotlarini qidirish mashinalari metao'zgaruvchilardan oladi. (Web sahifa qaysi sohaga tegishli, unda qanday ma'lumotlar borligi) <meta> tegida keymards va description o'zgaruvchisi bor. Keymards o'zgaruvchili Web sahifadagi kalitli so'zlar ro'yxatini o'zida saqlaydi. description o'zgaruvchi esa Web sahifaning qisqacha ma'lumotini o'zida saqlaydi. Masalan, bizning Web sahifamiz kompyuter viruslari haqida tayyorlangan bo'lsin u holda HTML hujjatda meta ma'lumotlarni quyidagicha yozish mumkin:

Misol 4.

```
<html>
  <head>
    <title>kompyuter viruslari </title>
    <meta name="keywords" lang="ru" content="virus,
      kompter, antivirus,...">
    <meta name="description" content="Web sahifadagi kompter
      viruslariga bag'ishlangan.
  </head>
  <body>
    kompyuter viruslari bu ....
```

<body>
</html>



1.1.4-rasm Kompyuter viruslari ko'rinishi xosil qilish

Brauzerlar foydalanuvchi tomonidan ochib ko'rilgan Web sahifalarini kesh xotirada saqlab qoladi. Agar foydalanuvchi yana shu sahifalarga murojaat qilsa, Web brauzer oldindan kesh xotirada mavjud bo'lgan (yana yangi sahifani internetdan olmasdan) nusxasini olib ko'rsatadi. Bu jarayon foydalanuvchining vaqtini va iqtisodini tejaydi. Endi Web sahifa qachon yangilanadi degan savol paydo bo'ladi. Bu savolga metama'lumotlardagi *expres* o'zgaruchilardan javob olish mumkin. Bu o'zgaruvchida Web sahifaning yaroqlilik muddati ko'rsatiladi. Agar kesh xotiradagi web sahifa yaroqlilik muddati o'tgan bo'lsa, brauzer tarmoqdan Web sahifani qaytadan o'qib oladi.

Misol:

<meta http-equiv="Expres" content="Tue,uf Aug 2002 14:56:27 Gmt">

Web sahifalarda ma'lumotlar tez o'zgarishi mumkin, masalan chatlarda va birja sahifalarida ma'lumotlar tez o'zgaradi. Bunday hollarda refresh o'zgaruvchisidan foydalaniladi va qiymatlari sekundlarda beriladi.

Masalan:

<meta http-equiv="Refresh" content=10>

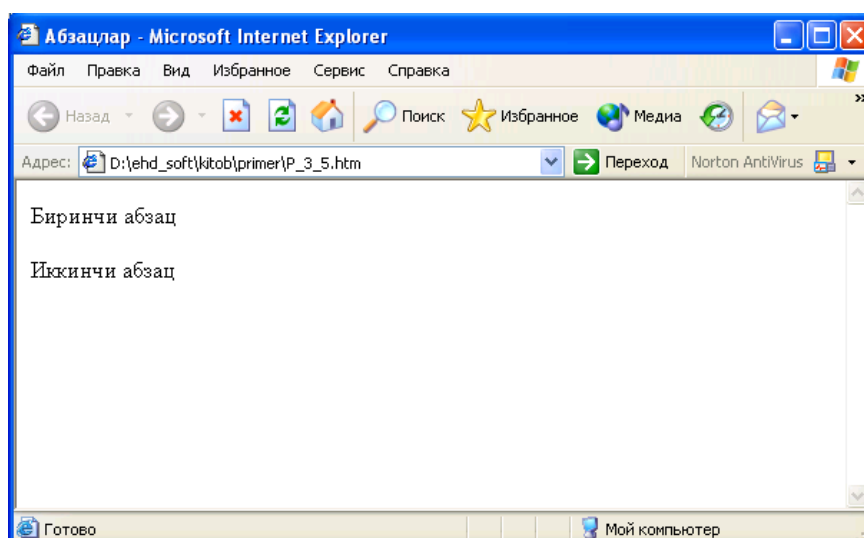
Bu yozuvdan keyin Web sahifa har 10 sekunddan keyin avtomatik tarzda o'zi qayta yuklanadi.

Identifikatorlar: HTML tilida har bir qo'llanilgan tegga unikal idintifikator berish imkoniyati mavjud. Masalan matn bir necha abzatlardan tashkil topgan bo'lsin. Har bir abzatsga mos maxsus nom berish mumkin, keyinchalik HTML tilining qo'shimcha imkoniyatlari yordamida bu abzatlarni boshqarish mumkin, ya'ni ularning birortasini ko'rinmas qilish Yoki shrifti rangini o'zgartirish.

Yuqoridagi ishlar faqat abzatslar uchun emas, balki Web sahifaning ixtiyoriy qismi uchun ta'luqlidir. Biror bir tegni nomlash uchun id parametri ishlatiladi. Abzatslar <p> va </p> orqali ko'rsatiladi.

Misol 5.

```
<html>  
  
  <head><title>Abzatslar</title></head>  
  
  <body>  
  
    <p id="p1"> Birinchi abzats </p>  
    <p id="p2"> Ikkinchi abzats </p>  
  
  </body>  
  
</html>
```



1.1.5-rasm Abzatslar ko'rinishi xosil qilish

HTML hujjatda id parametri qiymatlari takrorlanmasligi lozim, aks holda bu qiymatlar e'tiborga olinmaydi.

class parametri faqat shakl bezash ishlarida ishlatiladi. Biz Web sahifaning ayrim elementlarini sinflarga bo'lamiz, keyinchalik sinfni tasvirlash qoidalari yozuvini bir joydan o'zgartirishimiz mumkin va bu o'zgartirish avtomatik ravishda shu sinfga kirgan barcha teglarga tarqaladi.

HTML hujjatini tashkil etuvchi barcha elementlar ikki turga bo'linadi: inline elementlar va blokli elementlar. Inline elementlar oddiy matn elementlari bo'lib satr qismi ham bo'lishi mumkin, blokli elementlar esa har safar yangi satrdan boshlanishi shart. Blokli elementlar boshqa blokli elementlar va Inline elementlaridan tashkil topishi mumkin, lekin Inline elementlar blokli elementlarni o'z ichiga olmaydi. Web sahifa elementlarini bloklarga birlashtirish ularga birvarakayiga shakl berish imkonini beradi, yani, birlashtiruvchi yagona tegni aniqlab blok joylashuvini o'zgartirish mumkin. Tabiiyki bu Web sahifa elementlarining har birining joylashuvini alohida o'zgartirishdan oson.

Blokli tip elementlarini birlashtirish uchun <div> va </div> teglari qo'llaniladi. Inline elementlari esa va teglari orqali birlashtiriladi. YUqorida aytilganlarga asosan <div> tegi tegi ichida joylasha olmaydi.

Misol 6:

```
<html>
```

```
<head>
```

```
<title>div blokni hosil qilish</title>
```

```
</head>
```

```
<body>
```

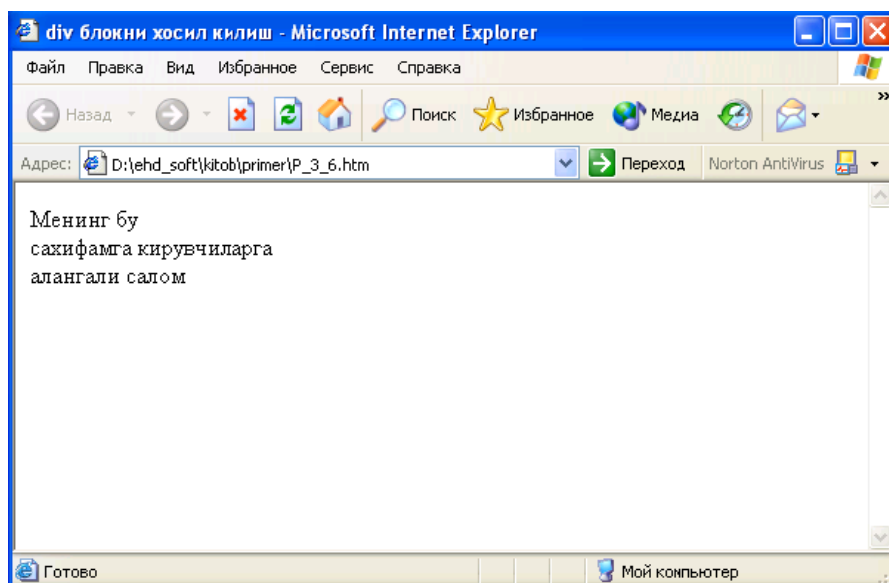
```
<div>
```

```
Mening bu <div> sahifamga kiruvchilarga </div> alangali salom
```

```
</div>
```

```
</body>
```


</html>



1.1.6-rasm Div blokni hosil qilish

 va <div> teglari qo'shimcha parametrlarni ham o'z ichiga olishi mumkin. Bizga malum bo'lgan id va class parametrlaridan tashqari style va align parametrlari ham ishlatiladi. style parametri shu blokdaqi malumot stilini o'rnatadi, align parametri shu malumotni qanday tekislashni aniqlaydi. HTML hujjatida sarlavhaning o'z teglari mavjud bo'lib, ular oltitadir. Eng yuqori darajasi bu birinchidir. Har bir sarlavhaning o'z tegi va o'z tasvirlanish qoidasi mavjud. Eng katta yani birinchi darajali sarlavha <h1> va </h1> teglari orqali, ikkinchi darajali sarlavha <h2> va </h2> teglari orqali va oxirgi oltinchi darajali sarlavha <h6> va </h6> teglari orqali ifodalanadi. Quyidagi misolda biz sarlavhalar tasvirini ko'ramiz:

Misol 7:

<html>

<head>

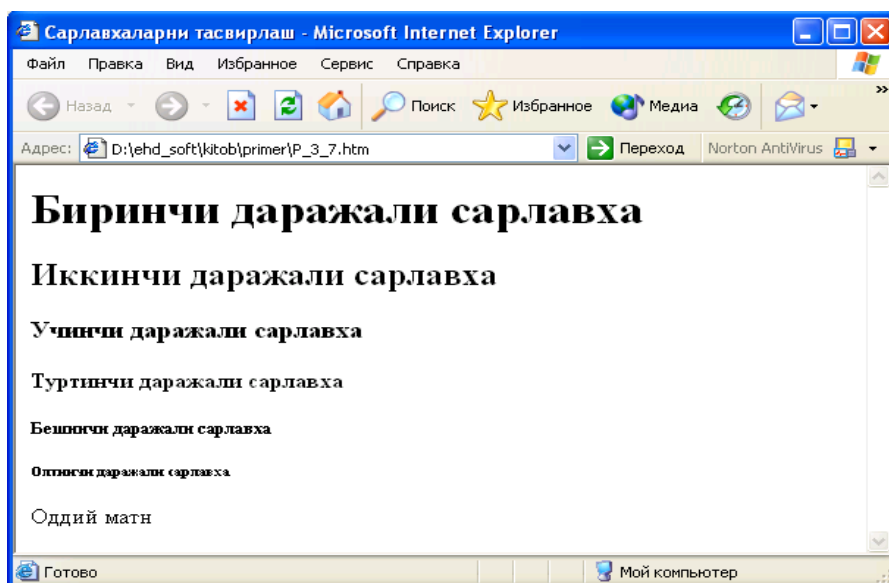
<title>Sarlavhalarni tasvirlash</title>

</head>

<body>

<h1>Birinchi darajali sarlavha</h1>

```
<h2>Ikkinchi darajali sarlavha</h2>
<h3>Uchinchi darajali sarlavha</h3>
<h4>To'rtinchi darajali sarlavha</h4>
<h5>Beshinchi darajali sarlavha</h5>
<h6>Oltinchi darajali sarlavha</h6>
<p>Oddiy matn</p>
</body>
</html>
```



1.1.7-rasm Sarlavhalarni tasvirlash

Sarlavha teglari `` va `<div>` teglari kabi id, class, style, align parametrlarini o'z ichiga oladi.

Ishlatiladigan belgilar: Kompyuterda har bir belgi qandaydir sonidan iboratdir. Operatsion sistema har bir songa mos keluvchi belgini ekranda tasvirlaydi. Sonlarga mos keluvchi belgilar jadvali kodirovka (kodlash) deyiladi. Hozirning o'zida rus tili belgilari uchun kamida 5 ta kodlash usullari mavjud. Agar Web sahifa yaratilayotganda qaysi kodlash usulidan foydalanilganni brauzer aniqlay olmasa, ekranda tushunarsiz belgilar paydo bo'ladi. HTML tili Web sahifa qaysi usulda kodlanganini ko'rsatib turish imkoniyatiga ega. Buning uchun `<meta>` tegi ishlatiladi.

HTTP protokolida (bayonnomasida) oldindan aniqlangan Content– Type nomli o'zgaruvchi mavjud. U o'zida Web sahifa tilini va kodlash usulini saqlaydi. Umumiy ko'rinish quyidagicha bo'ladi:

```
<META http-equiv = "Content-Type" content = "text/HTML;  
charset=ISO-8858-5">
```

Yuqoridagi misolda o'zgaruvchining qiymati “;” belgi bilan ajratilgan ikki qismdan iborat. Birinchi qism matnning HTML teglari yordamida yaratilgan oddiy matn ekanligini bildirsa ikkinchi qism foydalanilgan kodlash usulini ko'rsatib turadi. Yuqoridagi misolda xalqaro standartlashtirish tashkiloti (ISO) tomonidan tasdiqlangan standart kodlash usuli ko'rsatilgan. Afsuski brauzerlar matnda uchraydigan ba'zi bir belgilarni ekranda aks ettira olmaydi. Agar brauzer matnda “kichik” tengsizlik belgisini uchratsa uni teg uchun ochiluvchi qavs deb tushunadi. Matnda bu belgidan keyin hech qanday teg uchramasa matnning biror bir qismi e'tiborsiz qoldiriladi va ekranda aks ettirilmaydi. Bunday xatoliklarning oldini olish uchun matnda bunday belgilar o'rniga qo'shtirnoq ichiga olingan maxsus belgilar ketma – ketligi qo'llaniladi.

1.2. Dinamik saytlarni yaratish texnologiyasi (PHP)

PHP dasturlari ikki usulda bajarilishi mumkin: Web-server tomonidan stsensariy ilovasi va konsol dasturi sifatida.

Bizning maqsadimiz web ilovalarni dasturlash bo'lgani uchun asosan birinchi usulni ko'ramiz.

PHP odatda Internet bilan bog'liq dasturlar yaratish uchun ishlatiladi. Lekin PHP dan komanda satrlar interpretatori, asosan *nix tizimlarda foydalanish mumkin. Oxirgisi CORBA va COM interfeyslar hamda PHP-GTK kengaytmasi yordamida mumkin. Bu xolda quyidagi masalalarni echish mumkin:

- Interaktiv komanda qatorlari yordamida ilovalar yaratish;
- Kross-platformali GUI ilovalarni PHP-GTK bibliotekasi yordamida yaratish;

- Windows va Linux uchun ba'zi masalalarni avtomatizatsiya qilish.

Serverga brouzerning murojlat qilishida yordamida php-stsenariylari bajarilishini ko'rib chiqamiz. Avval borouzer php kengaytmali sahifani so'raydi, so'ngra web-server dasturni PHP mashinadan o'tkazadi va natijani html-kod shaklida qaytaradi. Agar standart HTML sahifani olib, kengaytmasini .php ga o'zgartirilsa va PHP mashinadan o'tkazilsa, foydalanuvchiga o'zgartirmasdan qaytaradi. Bu faylga PHP komandani qo'shish uchun, PHP komandalani maxsus teglar ichiga olish kerak. Bu teglarning 4 xil shakli mavjud bo'lib, ixtiyoriysidan foydalanish mumkin:

XML qayta ishlash instruktsiyasi:

<?php

...

?>

SGML qayta ishlash instruktsiyasi:

<?

...

?>

HTML stsenariylari qayta ishlash instruktsiyasi:

<script language = "php">

...

</script>

ASP uslubidagi instruktsiya:

<%

...

%>

Biz XML Yoki SGML uslubiga rioya qilamiz.

Xususan biror blok ichida PHPdan chiqish mumkin, faqat keyinchalik yana uning ichiga kirib kodni tugatish sharti bilan, quyidagi konstruktsiya mumkin:

<?

```

if(5<3){
    echo("<p>Hello, world!<p>");
?>

<p>Hello!</p>
// bu qator PHP kodi sifatida qaralmaydi
// va kod bloki bajarilayotgan bo'lsa chiqariladi
<?
    echo("<p>Hello, world!<p>");
}
?>

```

PHP da echo komandasi web – sahifalarda uchraydigan har qanday ma'lumotni(matn, HTML ajratuvchi simvoli, son) chiqarish uchun qo'llanadi. Uning ma'nosi misolda lo'rsatilgan.

Izoxlar

PHP tilida izoxlarni joylash uchun bir necha usullar mavjud. Eng soddasi ikkilik slesh (//) dan foydalanish, Shundan so'ng PHP satrlar oxirigacha yozilganni o'tkazib yuboradi. Bundan tashqari S (/*...*/) uslubidagi ko'p qatorli izoxlardan foydalanish mumkin. Bir qatorli izoxlar uchun (#) simvoldan foydalanish qulay.(UNIX script tillaridagi izox).

```

<php
    echo("<p>Hello</p>"); // izox
    echo("<p>Hello</p>"); # izox
/*
    bu ham izox
*/
?>

```

Shuni esdan chiqarmaslik lozimki PHP uslubi izoxlari faqat PHP chegaranishlari orasida ta'sir qiladi. Agar PHP bu izoxlar simvollarini

chegaranishlari tashqarisida uchratsa, ularni boshqa matnga oʻxshab, html-sahifaga joylashtiradi.

Masalan:

```
<php
    echo("<p>Hello</p>"); // normal izox
?>
```

// bu izox brouzerda koʻrinadi.

<!-- HTML izoxi.

Bu izox HTML kodda koʻrinadi, brouzerda emas -->

Izoxlarni faqat operator oxiriga emas, quyidagicha joylash ham mumkin:

```
<?
    $a = "Hello, world";
    echo strstr($a,"H");
    // bu funktsiyani keyinchalik qarab chiqamiz
?>
```

Oʻzgaruvchilar va konstantalar

RNR da oʻzgaruvchilar dollar (\$) belgisidan boshlanadi. Bu simvoldan ixtiyoriy sondagi harf, raqam va ostiga chizioʻ simvollar kelishi mumkin, lekin birinchi simvol albatta harf boʻlishi kerak. Shuni esda tutish kerakki, RNRda oʻzgaruvchilarning nomlari kalit soʻzlardan farqli registrga bogʻliqdir.

RNR da oʻzgaruvchilarni taʼriflaganda oshkora tipini koʻrsatish shart emas va dastur davomida itta oʻzgaruvchi har xil tiplarga ega boʻlishi mumkin.

Oʻzgaruvchi unga qiymat berilganda initsializatsiya qilinadi va dastur bajarilguncha mavjud boʻladi. Yaʼni web-sahifa xolida to soʻrov tugamaguncha.

Tashqi oʻzgaruvchilar

Klient soʻrovi veb-server tomonidan taxlil qilinib, RNR mashinaga uzatilgandan soʻng, u soʻrovga tegishli maʼlumotlarni oʻz ichiga olgan va bajarish davomida murojaat qilish mumkin boʻlgan bir necha oʻzgaruvchilarni yaratadi.

Oldin RNR sizni tizimingiz atrof muxit o'zgaruvchilarini oladi vash u nomdagi va shu qiymatdagi RNR stsenariysi atrofidagi o'zgaruvchilarni yaratadi, toki servedagi stsenariylarga klient tizimi xususiyatlari bilan ishlash mumkin bo'lsin. Bu o'zgaruvchilar \$HTTP_ENV_VARS assotsiativ massivga joylashtiriladi.

Tabiiyki \$HTTP_ENV_VARS massivi o'zgaruvchilari tizimga bog'liqdir (chunki ular aslida atrof muxit o'zgaruvchilaridir). Atrof muxito'zgaruvchilari qiymatlarini sizni mashinangiz uchun env (Unix) Yoki set (Windows) komandasi yordamida ko'rishingiz mumkin.

So'ngra RNR u GET-o'zgaruvchilarning guruxini yaratadi. Ular so'rov satrini taxlil qilishda yaratiladi. So'rov satri \$QUERY_STRING o'zgaruvchida saqlanadi va so'ralgan URL dagi "?" simvoldan keyingi informatsiyadan iborat. RNR so'rov satrini & simvollar bo'yicha aloxida elementlarga ajratadi, va har bir elementda "=" belgisini qidiradi. Agar "=" belgisi topilgan bo'lsa, tenglik chap tomonidagi simvollaridan iborat o'zgaruvchi yaratadi. Quyidagi formani ko'ramiz:

```
<form action = "http://localhost/PHP/test.php" method="get">
```

```
  HDD: <input type="text" name="HDD"/><br>
```

```
  CDROM: <input type="text" name="CDROM"/><br>
```

```
<input type="submit"/>
```

Agar siz bu formada HDD qatorda "Maxtor", CDROM qatorda "Nec" tersangiz, quyidagi so'rov shaklini xosil qiladi:

```
http://localhost/PHP/test.php?HDD=Maxtor&CDROM=Nec
```

Bizning misolimizda RNR quyidagi o'zgaruvchilarni yaratadi: \$HDD = "Maxtor" va \$CDROM = "Nec".

Siz o'zingizni scriptingizdagi (bizda – test.php) bu o'zgaruvchilar Bilan oddiy o'zgaruvchilar bilan ishlagandek ishlashingiz mumkin. Bizning misolimizda ular ekranga chiqariladi:

```
<?
```

```
  echo("<p>HDD is $HDD</p>");
```

```
  echo("<p>CDROM is $CDROM</p>");
```

```
?>
```

Agar sahifa so'rovi POST usuli yordamida bajarilsa, POST-o'zgaruvchilarning guruxi yaratilib, interpretatsiya qilinadi va \$HTTP_POST_VARS massivga joylashtiriladi.

Konstantalar

Konstantalar RNR da define() funktsiyasi yordamida e'lon qilinadi:

```
define(CONSTANT, value)
```

Bu funktsiya birinchi parametri – konstant nomi, ikkinchisi – uning qiymati.

Konstantadan foydalanilganda nomi bo'yicha ilova qilinadi:

```
<?
define(CONSTANT1,15);
define(CONSTANT2,"\x20"); // kod probela
define(CONSTANT3,"Hello");
echo(CONSTANT1);
echo(CONSTANT2);
echo(CONSTANT3);
?>
```

Odatga ko'ra konstantalar nomlari yuqori registr harflari bilan yoziladi. Bu faqat odat bo'lsa ham unga rioya qilishni maslaxat beramiz, chunki yaxshi odatlarga rioya qilmaydigan dasturchilardan yomon dasturchilar chiqadi. Konstantalar aniqlanganligini defined() funktsiyasi yordamida tekshirish mumkin:

```
<?
define(CONSTANT,"Hello");
if(defined("CONSTANT"))
{
    echo("<p>CONSTANT is defined</p>");
}
?>
```


RNR da ma'lumotlar tiplari. Tiplarni o'zgartirish

Yuqorida aytilganidek PHP tilida bitta o'zgaruvchini dastur bajarilish davomida satr Yoki son sifatida ishlatish mumkin. Shu bilan birga PHP tilida o'zgaruvchilar bilan ishlanganda oshkor ko'rsatilishi mumkin bo'lgan asosiy ma'lumotlar tiplari to'plami mavjud:

```
integer;  
string;  
boolean;  
double;  
array;  
object;
```

RNR o'zgaruvchiga tayinlagan tipni qaytaruvchi `gettype()` funktsiyasi mavjud:

```
<?  
$var = "5";  
$var1 = 5;  
echo(gettype($var));  
echo "<br>";  
echo(gettype($var1));  
?>
```

Birinchi xolda RNR string qaytaradi, ikkinchi xolda integer.

Tipni oshkora o'rnatuvchi `settype()` funktsiyasi mavjuddir:

```
<?  
$var = "5";  
echo(gettype($var));  
settype($var, integer);  
echo "<br>";  
echo(gettype($var));  
?>
```

Kodning bu fragmentini bajarish, avvalgisini bajarish Bilan bir xil natijaga olib keladi.

RNR tilida `settype()` funktsiyasidan tashqari tipni o'zgartirish operatoridan foydalanish mumkin. Tipni o'zgartirish qavslarga olingan Yangi tipni ko'rsatish orqali bajariladi:

```
$var = (int)$var;
```

RNR quyidagi kodni bajarish natijasida, integer qaytaradi:

```
<?
```

```
$var = "5"; // string tip
```

```
$var = (int)$var; // int ga o'zgartiramiz
```

```
echo(gettype($var));
```

```
?>
```

Operatorlar

1.2.1-jadval. Arifmetik operatorlar

+	Qo'shish
-	Ayirish
*	Ko'paytirish
/	Bo'lish
%	Modul bo'yicha qoldiqni hisoblash (masalan: 5 % 2=1)

Razryadli operatorlar

Bu operatorlar guruxi butun qiymatli operatorlarning bitli tasvirlari bilan ishlaydi. Asosan bu operatorlar mantiqiy bayroqlar to'plamini yaratish uchun ishlatiladi. Bizning misollarimizda bu operatorlarni kam ishlatamiz, lekin quyida shu operatorlardan foydalanishga misol keltiramiz.

	Razryadli kon'yunktsiya (AND)
	Razryadli diz'yunktsiya (OR)
	Razryadli ajratuvchi YOKI (XOR)
	Razryadli inkor (NOT)
<	Chap butun qiymatli operand bitli tasvirini o'ng butun qiymatli operand qiymatiga teng razryadlar soniga chapga siljitish
>	Chap butun qiymatli operand bitli tasvirini o'ng butun qiymatli operand qiymatiga teng razryadlar soniga o'ngga siljitish

Misol:

<?

```
echo(4<<2); // teng 16
```

```
echo"<br>";
```

```
echo(5>>1); // teng 2
```

```
echo"<br>";
```

```
echo(6&5); // teng 4
```

```
echo"<br>";
```

```
echo(6|5); // teng 7
```

```
echo"<br>";
```

```
echo(6^5); // teng 3
```

>?

Birinchi misolda 4 sonining ikkilik kodi 100 ga teng, chapga 2 razryad surilganda 100 kod 10000 ga teng bo'ladi. Bu o'nlik 16 ga teng bo'ladi. Boshqa misollarni ham shunga o'xshab qarab chiqish mumkin. Chapga n pozitsiyaga surish 2^n ga ko'paytirishga teng, o'ngga surish 2^n marta kamaytirib, kasr qismini tashlab yuborish natijasiga teng (shuning uchun $5 \gg 1$ teng 2). Bu operatorlar asosiy vazifasi tez hisoblashlar. Ular apparat darajasida qo'llangani uchun, bu operatorlardan foydalanib tuzilgan algoritmlar o'ta unumdor bo'ladi.

1.2.1-jadval. Mantiqiy operatorlar

| | |
|------------|-------------------|
| or ili | Yoki |
| Xor | Rad qiluvchi Yoki |
| and ili && | Va |
| ! | Inkor |

Bu gurux operatorlari razryadli operatorlardan farqli mantiqiy (boolean) o'zgaruvchilar bilan ishlaydi va boshqaruvchi konstruktsiyalar: sikllar va shartlarda ishlatiladi. Mantiqiy o'zgaruvchilar Yoki to'g'rirog'i Boolean tipidagi o'zgaruvchilar faqat ikki qiymatga egalar: true(rost) va false(yolg'on). Ifodalarda true va false qiymatlarini 1(0 dan farqli ixtiyoriy son) va 0 ga almashtirish mumkin. Bu imkoniyat maxsus true i false qiymatlariga ega bo'lmagan S tilining merosidir.

Mantiqiy operatorlar Boolean tipidagi o'zgaruvchilar uchun oddiy o'zgaruvchilar uchun qo'shish, ayirish operatorlari rolini o'ynaydi. Operator if(cond) avval cond shartni tekshiradi, agar uning qiymati true bo'lsa if dan keyingi figurali qavs ichidagi kod bajariladi, agar false bo'lsa else operatoridan keyingi kod bajariladi. Agar cond qiymati false bo'lsa teskarisi.

```
<php
    $flag = true; // Rost
    if($flag)
    {
        echo "<p> flag o'zgaruvchisi true qiymatga ega </p> ";
    } else {
        echo "<p> flag o'zgaruvchisi false qiymatga ega</p>";
    }
?>
```

Script ishlashi natijasi quyidagi ko'rinishga ega:

flag o'zgaruvchisi true qiymatga ega

Agar

```
$flag = true; // Rost
qator
$flag = false; // YOlg'on
qatorga almashtirilsa
teg qaytaradigan qiymat o'zgaradi:
flag o'zgaruvchisi false qiymatga ega
Ya'ni biz else bloka kirdik, chunki if operatorida shart yolg'on.
Shart bitta bo'lsa, hammasi sodda. Shartlar ko'proq bo'lsa mantiqiy
operatorlar ishga tushadi:
```

```
<php
    $flag1 = true; // Rost
    $flag2 = true; // Rost
    if($flag1 && $flag2) // I $flag1 va $flag2 rost
    {
        echo "<p>Shart: true (Ikkala bayroq rost)</p>";
        // Xa
    } else {
        echo "<p>Shart: false (Bayroqlardan biri yolg'on)</p>";
        // Yo'q, bittasi Yoki ikkovi yolg'on
    }
?>
```

Bu script quyidagini chiqaradi: Shart: true (Ikkala bayroq rost). Agar operatorlardan biri false bo'lsa(Yoki ikkovi) – bquyidagini chiqaradi: Shart: false (Bayroqlardan biri yolg'on).

Albatta shartlar bilan o'ynab ko'ring: agarda siz shartli operatorlarni yaxshi bilmasangiz hech qaysi dasturlash tilida yaxshi dasturchi bo'lolmaysiz. Agarda sizga shartli operator kerak bo'lmasa, demak sizga dasturlash tili ham kerak emas.

&& (VA) ni || (YOKI) ga almashtiramiz

```
<php
    $flag1 = true; // Rost
```

```

$flag2 = false; // Rost
if($flag1 || $flag2)
    // Agar bayroqlardan juda bo'lmasa biri ($flag1 $flag2) rost
{
    echo "<p>Shart: true (Bayroqlardan biri rost)</p>";
    // Xa
} else {
    echo "<p>Shart: false (Ikala bayroq yolg'on)</p>";
    // Yo'q, Ikkovi qiymati false
}
?>

```

Natija: Shart: true.

Va hamda Yoki operatorlarining ikki ko'rinishi mavjud bo'lib, o'larning ustivorligi har xildir. Mantiqiy operatorlar ustivorligi: or > xor > and > || > && > !
Bazida shartlarda ustivorlikni oshkor ko'rsatish uchun qavslardan foydalaniladi:

```

<php
    $flag1 = true;
    $flag2 = true;
    $flag3 = false;
    if( $flag1 && $flag2 || $flag3)
    {
        echo "<p>Shart rost <p>";
    }
    else
    {
        echo "<p>Shart yolg'on <p>";
    }
?>

```

Natijada «Shart rost» jumlasini chiqariladi, chunki ustivorlik asosida avval && operatori bajariladi. Buning sababi ikala bayroq true bo'lgani uchun—true

qaytaradi, soʻngra bu ifoda `$flag3 (true || $flag3)` Bilan solishtiradi va yana true qaytaradi, chunki solishtirilayotganlardan biri true ga teng.

Agar quyidagini yozsak

```
<php
    $flag1 = true;
    $flag2 = true;
    $flag3 = false;
    if($flag1 && ($flag2 || $flag3))
    {
        echo "<p>Shart rost <p>";
    }
    else
    {
        echo "<p>Shart yolgʻon <p>";
    }
?>
```

natija «Shart yolgʻon».

Umuman olganda dasturchilar ustivorliklar tartibini yodlashni yoqtirmaydilar, shuning uchun kodingiz tushunarli boʻlishi uchun qavslardan koʻproq foydalanish kerak. Yaʼni birinchi scriptda `if(($flag1 && $flag2) || $flag3)` – yozgan maqulroq.

Baʼzida shartlarni yolgʻon Yoki rostlikka tekshirish kerak boʻladi. Masalan, funktsiya `mysql_query($query)` –true qaytaradi agar `$query` satrida joylashtirilgan MySQL maʼlumotlar bazasiga murojaat muvaffaqiyatli bajarilsa, va false – aks xolda.

Agar siz muvaffaqiyatli murojaatga javob berishingiz kerak boʻlsa, quyidagicha yozishingiz mumkin:

```
<php
    if(mysql_query($query))
    {
```

```
        echo "<p>Ma'lumotlar muvaffaqiyatli ma'lumotlar bazasiga  
kiritilgan.<p>";  
    }  
?>
```

Agar siz muvaffaqiyatsiz murojaatga javob berishingiz kerak bo'lsa, quyidagicha yozishingiz mumkin:

```
<php  
    if(mysql_query($query))  
    {  
    else  
    {  
        echo"<p> Ma'lumotlar ma'lumotlar bazasiga kiritilmagan.<p>";  
    }  
?>
```

Lekin bu chalkashtiruvchi usuldir. Bu misolda inkor operatoridan foydalanish kerak: !, uning o'zgaruvchiga qo'llanishi qiymatini true dan false ga, false dan bo'lsa true ga:

```
<php  
    if(!mysql_query($query))  
    {  
        echo"<p> Ma'lumotlar ma'lumotlar bazasiga kiritilmagan.<p>";  
    }  
?>
```

1.3. Web texnologiyalar imkoniyatlarini oshirish vositalari (Java, Java script)

Webda sahifalarni ayratishda "klient-server" arxitekturasini bilan bogliq muammo yuzaga chiqadi. Sahifalarni ham klient tomonida, ham server tomonida

yaratish mumkin. 1995 yilda Netscape kompaniysi muttaxislari JavaScript dasturlash tilini ishlab chiqib, sahifalarni klient tomonida boshqarish mexanizmini yaratishdi.

Shunday qilib, Javascript – bu Webni gipermatnli sahifalarini klient tomonida ko'rish tsenariyalarini boshqarish tili. Yanada aniqroq aytadigan bo'lsa, Javascript – bu na faqat klient tomonidagi dasturlash tili. Liveware Javascript tilining avlodi bo'lib, Netscape serveri tomonida ishlovchi vosita bo'ladi. Ammo Javascript tilini mashhur qilgan narsa bu klient tomonida dasturlashdir.

Javascriptning asosiy vazifasi – HTML-konteynerlar atributlarining qiymatlarini va ko'rsatuvchi muhitining hossalirini HTML-sarlavxalarni ko'rish jarayonida foydalanuvchi tomonidan o'zgartirish imkoniyatlarida, boshqacha aytganda ularni dinamik sarlavxalar qilish (DHTML). Yana shuni aytish joizki, sarlavxalar qayta yuklanmaydi

Amalda buni, masalan, quydagicha ifodalash mumkin, sarlavxaning fonini rangini Yoki xujjatdagi rasmni o'zgartirish, yangi oyna ochish Yoki ogoxlantirish oynasini chiqarish.

“JavaScript” nomi Netscape kompaniyasining hususiy maxsuloti. Microsoft tomonidan amalga oshirilgan til rasman Jscript deb nomlanadi. Jscript versiyalari Javascriptning mos versiyalari bilan mos keladi (aniqroq qilib aytganda oxirigacha emas).

Javascript – ECMA (European Computer Manufacturers Association – Evropa Kompyuter Ishlab Chiqaruvchilar Assotsiyatsiyasi) tomonidan standartlashtirilgan. Mos standartlar quydagicha nomlanadi ECMA-262 va ISO-16262. Ushbu standartlar bilan Javascript 1.1ga taqriban ekvivalent ECMAScript tili aniqlanadi. Eslatish joizki, bugungi kunda Javascript ning hamma versiyalari ham ECMA standartlariga mos kelavermaydi. Mazkur kurs Yoki qo'llanmada barcha hollarda biz Javascript nomidan foydalanamiz.

Javascriptning asosiy hususiyatlari. Javascript – bu Internet uchun katta bo'lmagan klient va server ilovalarni yaratishga mo'ljallangan nisbatan oddiy ob'ektga yo'naltirilgan til. Javascript tilida tuzilgan dasturlar HTML-xujjatning

ichiga joylashtirilib ular bilan birga uzatiladi. Kurish dasturlari (brauzerlar – browser ingliz suzidan) Netscape Navigator va Microsoft Internet Explorer xujjat matniga joylashtirilgan dasturlarni (Scriptkod) uzatishadi va bojarishadi.

Shunday qilib, Javascript – interpretatorli dasturlash tili xisoblanadi. Javascriptda tuzilgan dasturlarga foydalanuvchi tomonidan kiritilayotgan ma'lumotlarni tekshirayotgan Yoki xujjatni ochganda Yoki yopganda biror bir amallarni bagaruvchi dasturlar misol bo'lishi mumkin.

JavaScript da yaratilgan dasturlarga misol sifatida foydalanuvchi tomonidan kiritilgan ma'lumotlarni tekshiruvchi, dokumentni ochish Yoki yopish vaqtida qandaydir amallarni bajaruvchi dasturlarni keltirish mumkin. Bunday dasturlar foydalanuvchi tomonidan berilgan kursatmalarga – sichqoncha tugmachasini bosilishiga, ma'lumotlarni ekran orqali kiritishiga Yoki sichqonchani sahifa buylab siljilinishiga kura ish bajaradi. Bundan tashqari JavaScript dagi dasturlar brauzerning uzini va dokumentning atributlarini ham boshqarishi mumkin.

JavaScript dasturlash tili sintaktik jihatdan Java dasturlash tiliga, ob'ektli modellashtirishni istisno qilgan holda, o'xshash ketsada, lekin ma'lumotlarni statik tiplari va qat'iy tiplashtirish kabi xususiyatlarga ega bulmaydi. JavaScript da Java dasturlash tilidan farq qilib, sinf (klass) tushunchasi bu tilning asosiy sintaktik qurilmasi hisoblanmaydi. Bunday asos sifatida foydalanilayotgan tizim tomonidan qullab-quvvatlanayotgan, oldindan aniqlangan ma'lumot tiplari: sonli, mantiqiy va satrli; mustaqil ham bulishi, ob'ektning metodi (JavaScriptda metod tushunchasi funktsiya/qism-dastur ning uzi) sifatida ham ishlatilishi mumkin bulgan funktsiyalar; katta sondagi uz hossalariga va metodlariga ega bulgan oldindan aniqlangan ob'ektlardan iborat ob'ektli model va yana dastur ichida foydalanuvchi tomonidan yangi ob'ektlarni berish qoidalari hisoblanadi.

JavaScript da dasturlar yaratish uchun hech qanday qo'shimcha vositalar kerak bulmaydi – faqatgina tegishli versiyadagi JavaScript qullanishi mumkin bulgan brauzer va DHTML-dokumentlarni yaratishga imkon beruvchi matn muharriri kerak bo'ladi. JavaScript dagi dastur bevosita HTML –dokumentlarni

ichiga joylashtirilganligi uchun dastur natijasini dokumentni brauzer yordamida kurish orqali tekshirish mumkin va kerakli holda uzgartirishlar kiritilishi mumkin.

JavaScript dasturlash tilining imkoniyatlari. Uning yordamida HTML – dokumentlarning ko'rinishi va tuzilishini dinamik ravishda boshqarish mumkin. Ekranda tasvirlanayotgan dokumentga brauzer tomonidan yuklangan dokumentning sintaktik tahlil qilish jarayonida istalgan HTML-kodlarni joylashtirish mumkin. “Dokument” ob’ekti yordamida foydalanuvchining oldingi bajargan amallari Yoki boshqa bir faktorlarga kura yangi dokumentlarni avtomatik hosil qilish mumkin.

JavaScript yordamida brauzer ishini boshqarish mumkin. Masalan, Window ob’ekti suzib yuruvchi oynalarni ekranga chiqarish, brauzerning yangi oynalarini yaratish, ochish va yopish, oynalarning yugurdagi va o'lchamlarining rejimlarini o'rnatish va hokazolarni imkoniyatini beruvchi metodlarga ega.

JavaScript dokumentdagi ma'lumotlar bilan bog'lanish imkoniyatini beradi. Document ob’ekti va undagi mavjud ob’ektlar dasturlarga HTML-dokumentlarning qismlarini o'qish va bazida ular bilan bog'lanish imkoniyatini beradi. Matnning o'zini o'qish mumkin emas, lekin masalan berilgan dokumentdagi gipermatnli o'tishlar ro'yhatini olish mumkin. Hozirgi vaqtda Form ob’ekti va undagi mavjud bo'lishi mumkin bo'lgan ob’ektlar: Button, Checkbox, Hidden, Password, Radio, Reset, Select, Submit, Text va Textarealar dokumentdagi ma'lumotlar bilan bo'g'lanish uchun keng imkoniyatlar beradi.

JavaScript foydalanuvchi bilan aloqa qilishga imkon beradi. Bu tilning eng muhim hususiyati unda amalga oshirilgan hodisalarni qayta ishlashni aniqlash imkoniyati – ma'lum bir hodisaning (odatda foydalanuvchi tomonidan bajarilgan amal) ro'y berish vaqtida bajariladigan dastur kodining ixtiyoriy qismi hisoblanadi. JavaScript hodisalarni qayta ishlovchi sifatida ixtiyoriy yangi oldindan berilgan funktsiyalardan foydalanish imkoniyatini beradi. Masalan, foydalanuvchi sichqoncha ko'rsatkichini gipermatnli o'tishlar ustiga keltirsa, holatlar satrida mahsus habarni chiqaruvchi Yoki ma'lum bir amalni bajarishni tasdiqlashni so'rovchi dialogli oynani ekranga chiqaruvchi Yoki foydalanuvchi

tomonidan kiritilgan qiymatlarni tekshiruvchi va hatolik yuz bergan holda kerakli kursatmalarni berib, tug'ri qiymatni kiritishni so'rovchi dasturlarni yaratish mumkin.

JavaScript ixtiyoriy matematik hisoblashlarni bajarish imkoniyatini beradi. Bundan tashqari bu tilda vaqt va sanalarning qiymatlari bilan ishlovchi yuqori darajada rivojlangan vositalar mavjud. JavaScript CGI-dasturlarga va Perl dasturlash tiliga va to'diruvchi sifatida ayrim hollarda Java tiliga muqobil til sifatida yaratilgan.

Har bir boshlovchi dasturchining asosiy savoli: "Dasturlar qanday tuziladi va bajariladi? ". Bu savolga iloji boricha soddaroq, lekin JavaScript-kodlarini qo'llanilishining barcha usullarini unutmagan holda javob berishga harakat qilamiz.

Birinchidan, JavaScript-kodlari brauzer tomonidan bajariladi. Unda mahsus JavaScript interpretatori mavjud. Unga kura programmaning bajarilishi interpretator tomonidan boshqaruvni qachon va qay tarzda olishiga bog'liq bo'ladi. Bu esa, o'z navbatida kodning funktsiyaviy qo'llanilishiga bo'g'liq bo'ladi. Umuman olganda JavaScript ning funktsional qo'llanilishining 4 hil usulini ajratib ko'rsatish mumkin:

1. gipermatnli o'tish (URL sxemasi);
2. hodisalarni qayta ishlash (handler);
3. o'rniga qo'yish(entity)
4. qo'yish (SCRIPT konteyneri).

JavaScript bo'yicha o'quv qo'llanmalarida JavaScript ni qo'llashning bayoni odatda SCRIPT konteyneridan boshlanadi. Lekin dasturlash nuqtai nazaridan bu unchalik ham to'g'ri emas, chunki bunday tartib asosiy savol: "JavaScript-kodi boshqaruvni qanday oladi?" ga javob bermaydi. Ya'ni JavaScript da yozilgan va HTML-dokumentning ichiga joylashtirilgan dastur qanday tarzda chaqiriladi va bajariladi.

HTML-sahifa muallifining kasbi va uning dasturlash asoslaridan habardarligining darajasiga qarab JavaScript ni o'zlashtirishga kirishishni bir necha hil variantlari mavjud. Agar siz klassik tillar (C, Fortran, Pascal va h.) bo'yicha dasturlovchi bo'lsangiz, u holda dokument matni ichida dasturlashdan boshlagan ma'qul, agar siz Windows sistemasida dasturlashga o'rgangan bo'lsangiz, u holda hodisalarni qayta ishlashni dasturlashdan boshlaganingiz ma'qul, agar siz faqat HTML bo'yichagina tajribaga ega bo'lsangiz Yoki anchadan beri dasturlash bilan shug'ullanmayotgan bo'lsangiz, u holda gipermatnli o'tishlarni dasturlashdan boshlaganingiz ma'qul.

URL-sxemali JavaScript. URL sxemasi (Uniform Resource Locator)- bu Web-texnologiyalarning asosiy elementlaridan biri. Web dagi har bir informatsion resurs uzining o'ziga hos URLiga ega bo'ladi. URL A konteynerining HREF atributida, IMG konteynerining SRC atributida, FORM konteynerining ACTION atributida va h.larda ko'rsatiladi. Barcha URL lar resursga ruhsatning protokoliga bog'liq bo'lgan ruhsat sxemalariga bo'linadi, masalan, FTP-arxiviga kirish uchun ftp sxemasi, Gopher-arxiviga kirish uchun gopher sxemasi, elektron maktublarni jo'natish uchun smtp sxemasi qo'llaniladi. Sxemaning tipi URLning birinchi komponentasiga ko'ra aniqlanadi: <http://directory/page.html>

Bu holatda URL http bilan boshlanayapti- mana shu kirish sxemasini aniqlashdir (sxema http).

Gipermatnli sistemalar uchun dasturlash tillarining asosiy vazifasi gipermatnli o'tishlarni dasturlashdir. Bu shuni bildiradiki, u Yoki bu gipermatnli o'tishlarni tanlashda gipermatnli o'tishni amalga oshiruvchi dastur chaqiriladi. Web-texnologiyalarida standart dastur sifatida sahifani yuklash dasturi hisoblanadi. HTTP protokoli bo'yicha standart o'tishni JavaScript da dasturlangan o'tishdan farq qilish uchun yaratuvchilar yangi URL sxemasi – JavaScript ni kiritishdi:

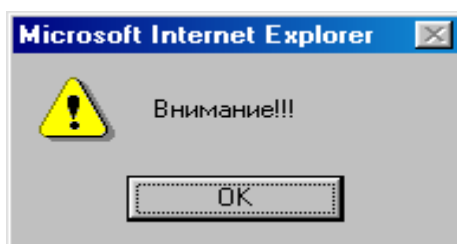
```
<A HREF="JavaScript:JavaScript_код">...</A>
```

```
<IMG SRC="JavaScript:JavaScript_код">
```

Bu holatda "JavaScript_код" matni birinchi holatda gipermatnli o'tishni tanlanganda chaqiriladigan va ikkinchi holatda rasmni yuklashda chaqiriladigan JavaScript da yaratilgan dasturiy-qayta ishlovchilarni bildiradi.

Masalan, Внимание!!! gipermatnli o'tishga keltirilgan holda ogohlantiruvchi oynani chiqarish mumkin:

```
<A HREF="JavaScript: alert('Внимание!!!');"> Внимание!!!</A>
```



1.3.1-rasim Formadagi submit tipidagi tugmachani bosish orqali shu formadagi matnli maydonni to'ldirish mumkin:

```
<FORM NAME=f METHOD=post  
ACTION="JavaScript: window.document.f.i.VALUE='Нажали кнопку  
Click';void(0);">  
<TABLE BORDER=0>  
<TR>  
<TD><INPUT NAME=i></TD>  
<TD><INPUT TYPE=submit VALUE=Click></TD>  
<TD><INPUT TYPE=reset VALUE=Reset></TD>  
</TABLE>  
</FORM>
```

URLda murakkab dasturlar va funktsiya chaqirilishlarini joylashtirish mumkin. Faqatgina shuni yodda tutish kerakki, JavaScript sxemasi hamma brauzerlarda ham ishlamaydi, faqatgina Netscape Navigator va Internet Explorer larning to'rtinchi versiyalaridan boshlab ishlaydi.

Shunday qilib gipermatnli o'tishlarni dasturlashda interpretator boshqaruvni foydalanuvchi sichqoncha tugmasini gipermatnli o'tishga "bosgandan" keyingina oladi.

Hodisalarni qayta ishlovchilar. Hodisalarni qayta ishlovchi dasturlar (handler) shu hodisa bog'liq bo'lgan konteynerlarning atributlarida ko'rsatiladi. Masalan, tugmachani bosishda onClick hodisasi ro'y beradi:

```
<FORM><INPUT TYPE=button VALUE="Кнопка"
onClick="window.alert('Внимание!!!');"></FORM>
```

O'rniga qo'yish. O'rniga qo'yish (entity) Web-sahifalarda anchagina kam uchraydi. Shunga qaramay u brauzer tomonidan HTML-sahifani hosil qilish uchun ishlatiladigan kerakli darajadi kuchli instrument hisoblanadi. O'rniga qo'yishlar HTML-konteynerlarning atributlari qiymatlari sifatida ishlatiladi. Masalan, foydalanuvchining shahsiy sahifasini aniqlovchi forma maydonining boshlang'ich qiymati sifatida joriy sahifaning URLi ko'rsatiladi:

```
<SCRIPT>
function l()
{
    str = window.location.href;
    return(str.length);
}
</SCRIPT>
<FORM><INPUT VALUE="&{window.location.href};"
SIZE="&{l()};">
</FORM>
<SCRIPT>
<!-- Это комментарий ...JavaScript-код...// -->
</SCRIPT>
<BODY>
```

```
Тело документа ...  
</BODY>  
</HTML>
```

HTML-shathlar bu yerda sahifaning berilgan bo'lagini eski brauzerlardagi HTML-parserlar tomonidan interpretatsiya qilinishidan saqlanish uchun qo'yilgan (yuqori boshqaruvdagilarda hali ham uchraydi). O'z navbatida HTML-sharhining ohiri JavaScript interpretatori tomonidan interpretatsiya qilinishidan himoya qilingan (satr boshidagi // belgisi). Bundan tashqari konteiner boshlanishidagi tegning LANGUAGE atributining qiymati sifatida "JavaScript" ko'rsatilgan. JavaScript ga muqobil sifatida ko'riluvchi VBScript keng qo'llaniluvchi tildan ko'ra ko'proq ekzotika hisoblanadi, shuning uchun bu atributni tushirib qoldirish mumkin – "JavaScript" qiymati o'z-o'zidan qabul qilinadi.

O'z –o'zidan ayonki, dokument sarlavhasida matnni hosil qilishni joylashtirish mantiqsizdir – u brauzer tomonidan namoyish qilinmaydi. Shuning uchun sarlavhada keyinchalik dokument ichida foydalaniladigan umumiy o'zgaruvchilar va funktsiyalar e'lon qilinadi. Bunda Netscape Navigatorda Internet Explorerga nisbatan talab qat'iyroqdir. Agar funktsiya sarlahada e'lon qilinmagan bo'lsa, dokument ichida bu funktsiya chaqirilsa, bu funktsiya aniqlanmaganligi to'g'risidaga habarni olish mumkin.

Funktsiyani joylashtirish va ishlatishga oid misolni ko'ramiz:

```
<HTML>  
<HEAD>  
<SCRIPT>  
function time_scroll()  
{  
    d = new Date();  
    window.status = d.getHours()+":"+d.getMinutes()+":"+d.getSeconds();  
    setTimeout('time_scroll();',500);  
}
```



```
</SCRIPT>
</HEAD>
<BODY onLoad=time_scroll(>
<CENTER>
<H1>Час в строке статуса</H1>
</CENTER>
</BODY>
</HTML>
```

O'rniga qo'yishlar bo'lgan holda brauzer (parser komponent) tomonidan HTML-dokumentni tahlil qilish vaqtida interpretator boshqaruvni o'z qo'liga oladi. Parser konteyner atributida &{..} konstruktsiyasini uchratishi bilan boshqaruvni JavaScript interpretatoriga beradi, u esa o'z navbatida bu kodni bajargandan so'ng boshqaruvni yana parserga qaytaradi. Shunday qilib bu operatsiya HTML-sahifaga grafikani yuklashga o'hshab ketadi.

Qo'yish (SCRIPT konteyneri- interpretatorni majburiy chaqirish). SCRIPT konteyneri bu o'rniga qo'yishlarni JavaScript-kod tomonidan dokument matnini hosil qilish imkoniyati darajasigacha rivojlantirilishidir. Bu ma'noda SCRIPTni qo'llanilishi Server Side Includesga o'hshab ketadi, ya'ni server tomonidan dokumentlarning sahifalarini hosil qilishga. Lekin bu yerda biz sal ilgarilab ketdik. Dokumentni tahlil qilish vaqtida HTML –parser SCRIPT konteyneri boshlanishi tegini uchratgandan so'ng boshqaruvni interpretatorga beradi. Interpretator SCRIPT konteynerining ichidagi barcha kod bo'lagini bajaradi va SCRIPT konteyneri ohirini ko'rsatuvchi tegdan so'ng sahifa matnini qayta ishlash uchun boshqaruvni HTML-parserga qaytaradi.

SCRIPT konteyneri quyidagi 2 ta asosiy funktsiyani bajaradi:

1. HTML-dokument ichiga kodni joylashtirish;
2. brauzer tomonidan HTML-belgilarni shartli hosil qilish.

Birinchi funktsiya keyinchalik o'tishlar dasturi, hodisalarni qayta ishlovchilar va almashtirishlar sifatida ishlatish uchun o'zgaruvchilar va

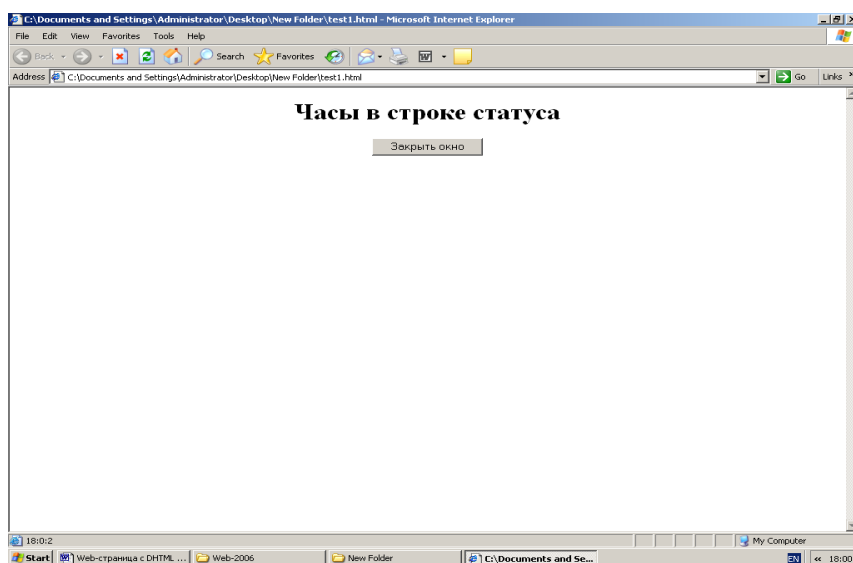
funktsiyalarni e'lon qilinishiga o'hshab ketadi. Ikkinchisi bu dokumentni yuklash Yoki qayta yuklash paytida JavaScript-kod bajarilishi natijalarini o'rniga qo'yishdir.

HTML-dokument ichiga kodni joylashtirish. Hususan olganda bu yerda hech qanday ajralib turadigan har hillik yuq. Kodni dokument sarlavhasida, HEAD konteynerining ichida Yoki BODY ning ichiga joylashtirish mumkin. So'nggi usul va uning hususiyatlari "Brauzer tomonidan HTML-bo'limlarini shartli hosil qilish" bo'limida ko'rib chiqiladi. Shuning uchun diqqatimizni dokument sarlavhasiga qaratamiz.

Sarlavhada kod SCRIPT konteinerining ichiga joylashtiriladi:

```
<HTML>
<HEAD>
<SCRIPT>
function time_scroll()
{
    d = new Date();
    window.status = d.getHours()+":"+d.getMinutes()+":"+d.getSeconds();
    setTimeout('time_scroll();',500);
}
</SCRIPT>
</HEAD>
<BODY onLoad=time_scroll()>
<CENTER>
<H1>Час в строке статуса</H1>
<FORM>
<INPUT TYPE=button VALUE="Закрыть окно" onClick=window.close()>
</FORM>
</CENTER>
</BODY>
```

</HTML>



1.3.2-rasim Time_scroll funktsiyani dokument sarlavhasi

Bu misolda biz time_scroll() funktsiyani dokument sarlavhasida e'lon qildik, keyinchalik esa uni BODY konteyneri boshining tegida hodisalarni qayta ishlovchi load sifatida chaqirdik (onLoad=time_scroll()).

O'zgaruvchilarni e'lon qilishga misol sifatida avvalgi-oyina tomonidan keyingi-oyinaning statusini o'zgartirishni ko'ramiz.

Quyidagi funktsiya yordamida hosilaviy oynani e'lon qilish va keyin chaqirish orqali yaratamiz:

```
function sel()
{
    id = window.open("", "example", "width=500,height=200,status,menu");
    id.focus();
    id.document.open();
    id.document.write("<HTML><HEAD>");
    id.document.write("<BODY>");
    id.document.write("<CENTER>");
    id.document.write("<H1>Change text into child window.</H1>");
}
```

```

id.document.write("<FORM NAME=f>");
id.document.write("<INPUT TYPE=text NAME=t SIZE=20 MAXLENGTH=20
VALUE='Это тест'>");
id.document.write("<INPUT TYPE=button VALUE='Закрыть окно'
onClick=window.close()></FORM>");
id.document.write("</CENTER>");
id.document.write("</BODY></HTML>");
id.document.close();
}
<INPUT TYPE=button VALUE="Изменить поле статуса в окне примера"
onClick="id.defaultStatus='Привет'; id.focus();">

```

Keyingi oynani ochishda id o'zgaruvchiga oyna ob'ektiga bo'lgan ko'rsatkich id=window.open() ni joylashtirdik. Endi biz uni Window sinfining o'bekti identifikatori sifatida ishlatishimiz mumkin. Bizning holatda id.focus() dan foydalanish shart. "Misoldagi oynadagi status maydonini o'zgartirish" tugmasini bosish orqali e'tiborni asosiy oynaga o'tkaziladi. U ekran o'lchamida bo'lishi mumkin. Bunda asosiy oyna tomonidan berkitib turilgan ikkinchi oynada o'zgarishlar ro'y beradi. O'zgarishlarni ko'rish uchun diqqatni yana ikkinchi oynaga berish kerak. id o'zgaruvchi qandaydir funktsiya tashqarisida aniqlangan bo'lishi kerak. Bu holatda u oynaning hususiyati ga aylanadi. Agar biz uni keyingi oynani ochish funktsiyasi ichiga joylashtirsak, u holda hodisalarni qayta ishlovchi click orqali unga murojaat qila olmaymiz.

Brauzer tomonidan HTML-bo'limlarini shartli hosil qilish. Serverdan o'zimizning brauzer imkoniyatlariga Yoki foydalanuvchiga moslashtirilgan sahifalarni olish doimo yoqimli. Bunday sahifalarni hosil qilishning faqatgina 2 imkoniyati bor: server tomonidan Yoki bevosita klient tomonidan. JavaScript – kod klient tomonida bajariladi (aslida Netscape kompaniyasi serverlari JavaScript-kodni server tomonida ham bajarishi mumkin, lekin bu holda u LiveWire-kod

nomini oladi; LiveConnect bilan adashtirilmasin), shuning uchun faqatgina klient tomonida hosil qilinishini ko'rib chiqamiz.

HTML-bo'limlrini hosil qilish uchun SCRIPT konteyneri dokument asosiy qismining ichiga joylashtiriladi. Oddiy misol- mahalliy vaqt sahifasiga moslashtirish:

```
<BODY>
...
<SCRIPT>
d = new Date();
document.write("<BR>");
document.write("Момент загрузки страниц:
"+d.getHours()+":"+d.getMinutes()+":"+d.getSeconds());
document.write("<BR>");
</SCRIPT>
...
</BODY>
```

JavaScriptda operatorlar, ifodalar, funktsiyalar

Operatorlar: arifmetik amallar, o'zlashtirish, orttirish, kamaytirish. Shartli ifodalar:

Qo'shish "+", ayirish "-", ko'paytirish "*", bo'lish "/", qoldikli bo'lish "%".

Bu ifodalar har qanday sonli ifodalarda uchrashi mumkin.

Shuni ko'rish mumkinki, JavaScriptdagi o'zlashtirish operatorlari C va Java dagilar bilan bir hil: "=", "+=", "-=", "*=", "/=", "%=".

x=y ifoda boshqa ko'pgina dasturlash tillaridagi kabi "x" o'zgaruvchiga "y" qiymatni berishni bildiradi.

Quyidagi operatorlar C dagi mos operatorlar bilan bir hil sintaktik qoidalariga ega:

$y+=x$ bilan $y=y+x$ ekvivalent

$y-=x$ bilan $y=y-x$ ekvivalent

$y*=x$ bilan $y=y*x$ ekvivalent

$y/=x$ bilan $y=y/x$ ekvivalent

$y\%=x$ bilan $y=y\%x$ – y ni x ga butun bo'lgandagi qoldiq- ekvivalent.

Shartli ifodalar quyidagi ko'rinishga ega

$(shart)?qiymat1:qiymat2$

2. Internetda MB ni tashkil etish

2.1. MS SQL ma'lumotlar bazasini boshqarish tizimi

Ma'lumotlar bazasining bor imkoniyatlari u bilan ishlashga mo'ljallangan instrumentlar orqali ochiladi. Biz quyida bunday instrumentlarni hozirgi davrda keng tarqalgan dasturlash tillari yordamida yaratishni ko'ramiz. Biznes uchun veb Amaliy dasturlari yaratish misolida biz API Lar, hamda MySQL va mSQL imkoniyatlaridan maksimal foydalanish uchun zarur bo'lgan instrumentlarni ko'rib chiqamiz. Biz avval ma'lumotlar bazasi Bilan ishlovchi amaliy dasturlar arxitekturasini va SGI- dasturlashga to'xtalamiz. Asosan MySQL va mSQL uchun har xil tillarda dasturlashni ko'ramiz.

Ma'lumotlar bazasi bilan ishlash

Ma'lumotlar bazasi Bilan ishlash uchun har xil tillarda amaliy dasturlar yaratish xususiyatlarni ko'rishdan oldin, bu amaliy dasturlarni loyihalashni ko'rib chiqish kerak. Biz kontseptual tushuncha Ya'ni ma'lumotlar bazasini dasturlash asosida yotadigan klient/ server arxitekturasini ko'rib chiqamiz. Bu masalalar MySQL va mSQL da dasturlash uchun muxim, lekin faqat ularga xos emas. Aksincha ular ma'lumotlar bazalarini dasturlash ixtiyoriy muxiti uchun muximdir. Agar arxitektura printsiplarini hisobga olmasa, sizni amaliy dasturlaringiz talablaringizga javob berolmaydi va o'zgaruvchi muxitga moslasha olmaydi. Biz quyida murakkab mavzularga Ya'ni oddiy tkki bo'g'inli arxitektura, ob'ektlar va relyatsion ma'lumotlar orasidagi munosabat, hamda Yangi uch bo'g'inli klient/server arxitekturasiga to'xtalib o'tamiz.

Klient/server arxitekturasini

Sodda xolda klient/server arxitektura amaliy dasturdagi qayta ishlashni ikki Yoki undan ko'p mantiqiy qismlarga ajratishga asoslangan. Ma'lumotlar bazasi qandaydir amaliy dastur tomonidan foydalanish uchun yaratilgandir.

Soddalashtirib aytish mumkinki ma'lumotlar bazasi klient/server arxitekturasining bir qismini tashkil qiladi. Ma'lumotlar bazasi «server», undan foydalanuvchi har qanday amaliy dastur «klient». Odatda klient va server har xil mashinalarda joylashgan; ko'p xollarda klient amaliy dasturi ma'lumotlar bazasiga do'stona interfeysdir. Quyidagi ris. 2-1 grafik shaklda klient/server sodda tizimi tasviri berilgan.

Balkim siz Internetda bunday strukturani uchratgan bo'lishingiz mumkin. Moxiyat bo'yicha biz Internet uchun klient/server amaliy dasturlarini yaratish masalasiga murojaat qilamiz. Masalan WWW ulkan klient/server tipidagi amaliy dastur bo'lib, unda Web-brouzer klient Web-server bo'lsa serverdir. Bu stsenariyda server relyatsion ma'lumotlar bazasi serveri emas, maxsus fayl-serverdir. Serverning muxim xususiyati shundaki, u klientga ma'lumotlarni ma'lum shaklda uzatadi.



2.1.1-rasim Klient server arxitekturasini

Ma'lumotlar bazasi bilan ishlaydigan amaliy dasturlar yaratilganda avvalambor klientni ma'lumotlar bazasi bilan bog'lash imkoniyatiga ega bo'lish kerak. Ma'lumotlar bazalari yaratuvchila dasturchilardan konkret tilga mo'ljallangan, API yordamida bog'lanish asosiy mexanizmlarni berkitadilar. Ma'lumotlar bazasi bilan ishlovchi dastur yaratganingizda sizni so'rovlarinigizni tarmoq orqali ma'lumotlar bazasi serveriga uzatiluvchi TCP/IP paketlariga translyatsiya qiladi.

Ma'lumotlar bazasiga murojaat API larining tashqi ko'rinishi har xil va dasturlash tillariga, ko'p xollarda ma'lumotlar bazasining o'ziga bog'liq. MySQL

uchun API lar mSQL bilan o'xshash qilib yaratilgani uchun, biz ko'radigan API lar orasidagi farq minimaldir.

Ma'lumotlarni qayta ishlash

Ma'lumotlar bazasi ishlashga mo'ljallangan dastur tranzaktsiyalarni boshqarish va natijaviy to'plamlarni qayta ishlash uchun mo'ljallangan instrumentdir. Masalan, sizni sizni amaliy dasturingiz adres kitobi bo'lsa, natijaviy to'plamlarni qayta ishlash, jadvaldan hamma satrlarni ajratish va foydalanuvchiga ko'rsatishdan iborat bo'ladi. Tranzaktsiyalarni boshqarish address va person jadvallarda o'zgarishlar birgalikda amlga oshirishni ta'minlashdan iborat bo'ladi.

MySQL da tranzaktsiyalar qo'llanmaydi. Ma'lumotlar bazasida har qanday o'zgarish avtomatik ravishda siz so'raganingizda yuz beradi. Bu cheklanish ma'lumotlar bazasiga ikki bog'liq murojlat oralig'ida rad etish natijasiga ma'lumotlar yaxlitligi buzilmasligi uchun maxsus usullar qo'llashga majbur qiladi.

Amaliy dastur ishidagi ikki muxim moment bu ulanish va uzilishdir. Tushunarliki so'rovni bajarishdan oldin ma'lumotlar bazasiga ulanish lozim. Lekin ko'pincha masalaning ikkinchi tomoni haqida unutiladi, Ya'ni «o'zidan keyin tozalash». Hardoim egallangan ma'lumotlar bazasi resurslarini ozod qilishlozim, agar ular kerak bo'lmasa. Ko'p yashovchi Amaliy dasturlarda masalan Internet demoni kabi, beparvo yozilgan tizim ko'p vaqt maelumotlar bazasi resurslarini egallashi mumkin va oxir borib tizimni blokirovka qilishi mumkin.

«O'zidan keyin tozalash» xatolarni to'g'ri qayta ishlashdan iborat. YAxshi dasturlash tillari g'ayri odiy xolatlar (tarmoqning rad qilishi, qo'shilganda qaytariluvchi kalitlar, SQL sintaksisi xatosi va xokazo) qayta ishlovilarini o'tkazib yuborishni qiyinlashtiradi. Qaysi tilni tanlagandan qat'iy nazar, API bu chpqirig'ida qanday g'ayri oddiy xolatlar yuzaga kelishi mumkinligini va har bir g'ayri oddiy xolatda nima qilish kerakligini bilishingiz kerak. MySQL uchun S-bibliotekalari ma'lumotlar bazalarini satrlar to'plami sifatida tasvirlashga asoslangan. Bush uni bildiradiki S bibliotekalari ma'lumotlar Bilan ma'lumotlar bazasida qanday tasvirlangan bo'lsa shu shaklda qaya ishlashga imkon beradi.

Ob'ektga-yo'naltirilgan muxitdan relyatsion ma'lumotlar bazasiga murojaat aloxida paradoksni eltirib chiqaradi: relyatsionny Dune ma'lumotlar ustida ish olib borsa, ob'ektlar dunyosi faoliyat sxemalariasosida ma'lumotlarni inkapsulyatsiya qilish Bilan shug'ullanadi. Ob'ektga-yo'naltirilgan Amaliy dasturlar ma'lumotlar bazasi ob'ektlarni saqlash vositasi bo'lib xizmat qiladi. Ob'ektga-yo'naltirilgan dasturlash so'rov ma'lumotlarni satrlar to'plami deb emas, ob'ektlar to'plami deb qaraydi.

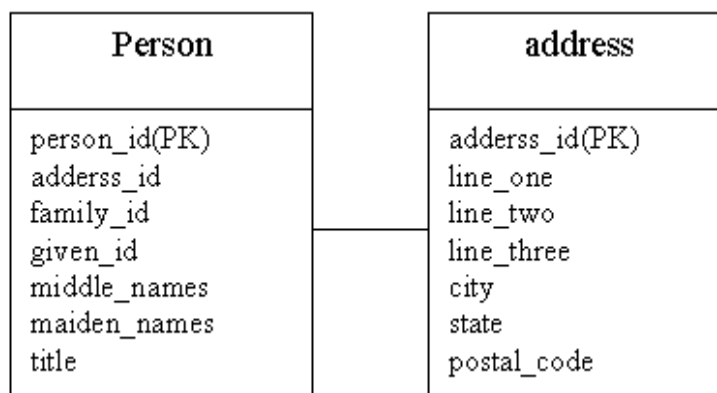
Ob'ektli/relyatsion modellashtirish

Ob'ektga yo'naltirilgan amaliy dastur yaratuvchisi oldida relyatsion ma'lumotlar bazasidan foydalanishda tug'iladigan masala, relyatsion ma'lumotlarni qanday qilib ob'ektlarga akslantirishdir. Birinchi yo'l ob'ekt atributlarini jadval maydonlariga akslantirish. Lekin bu usul ancha noqulaydir.

- Ob'ektlar o'z atributlarida faqat berilganlarni saqlamaydilar. Ularda kollektsiyalar va i boshqa ob'ektlar Bilan bog'lanishlar saqlanadi.
- Ko'pgina relyatsion ma'lumotlar bazalarida vorislikni modellashtirish imkoniyati yo'qdir.
- Ob'ektno-relyatsion modellashtirish uchun amaliy qoidalar
- Har bir saqlanuvchi sinf uchun ma'lumotlar bazasida o'zining jadvali mavjud.
- Ob'ektlarning sodda tipli (butun, simvol, satr va xokazo) maydonlariga ma'lumotlar bazasidagi mos jadvalning ustunlariga mos qo'yilgan.
- Ma'lumotlar bazasi jadvalining har bir satri saqlpnuvchi sinf nusxasiga mos keladi.
- Har bir ob'ektlar orasidagi «ko'pga-ko'p» turdagi bog'lanish jadval – bog'lovchini talab qiladi, xudi shundek ma'lumotlar bazasi «ko'pga-ko'p» turdagi ob'ektlari uchun shu talab qilinadi.
- Vorislik sinf va ostki sinfga mos keluvchi jadvallar orasidagi «birga-bir»

munosabat yordamida modellashtiriladi.

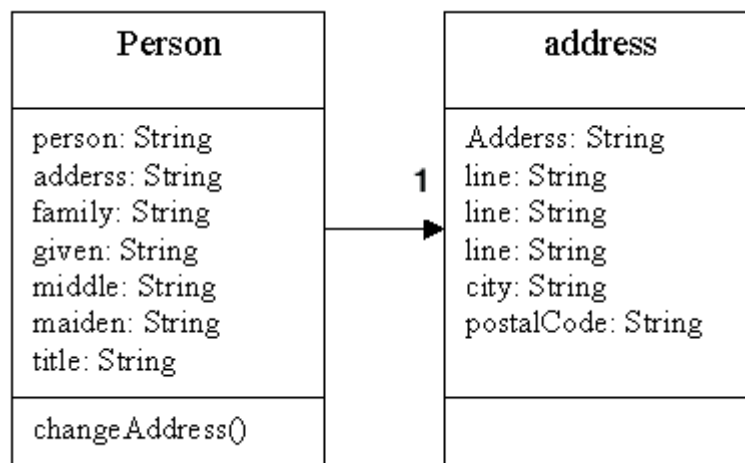
Oldin ko‘rilgan adreslar kitobini eslang. Masalan address va person jadvallariga ega bo‘lsin, 2.1.1- rasmdagi kabi.



2.1.2-rasmi Adres kitobi amaliy dasturi ma‘lumotlar modeli

Dasturchilarga uchraydigan ravshan bo‘lmagan muammo mavjuddir. Ob‘ektga-yo‘naltirilgan dasturlashni relyatsion ma‘lumotlarga qo‘llash - eto, poluchiv bu ma‘lumotlarni olib, darxol ob‘ekt nusxasini yaratishdir. Amaliy dastur ma‘lumotlar Bilan faqat ob‘ektlar orqali ishlashi kerak. Ko‘pginatraditsion dasturlash usullari, shu qatorda S, PowerBuilder va VisualBasic da dasturlash, yaratuvchi bazadan ma‘lumotlarni olib, keyin qayta ishlashga asoslangan. Asosiy farq shundaki ma‘lumotlar bazalarini ob‘ektga-yo‘naltirilgan dasturlashda siz mao‘lumotlar Bilan emas, ob‘ektlar bilan ishlaysiz.

2.1.3-rasmda ko‘rsatilgan ob‘ektli model, 2.1.2-rasmda ko‘rsatilgan ma‘lumotlar modeliga mos keladi. Ma‘lumotlar bazasi har bir satri dasturiy ob‘ektga aylantiriladi. SHunday qilib Amaliy dastur natijaviy to‘plamni qabul qiladi va har qaytariluvchi satr uchun Address Yoki Person Yangi nusxasini yaratadi. Eng qiyini yuqorida ko‘rsatilgan muammoni xal qilish: qanday qilib amaliy dasturda odam va uning adresi orasida bog‘liqlikni o‘rnatish. Ob‘ekt Person, albatta shu odamga tegishli ob‘ekt Address ga ilovaga ega, lekin ob‘ekt Address ni relyatsion baza person jadvalida saqlashmumkin emas. Ma‘lumotlar modeli ob‘ektlar orasidagi bog‘lanishni tashqi kalitlar yordamida saqlaydi. Buning uchun person jadvaliga address_id kiritiladi.



2.1.3-rasim Adres kitobi sodda amaliy dasturi ob'ektli modeli

Ob'ektli modelning ozgina murakkablashi bizning ob'ektlarimiz va ma'lumotlar modellari orasida moslikni o'rnatishda katta muammolarga olib keladi.

Masalan Person sinfi Entity avlodi va Company sinfi ham Entity avlodi bo'lsin. Entity ni Person dan YokiCompanynidan qanday ajratish mumkin? Yuqorida keltirilgan qoida faqat yo'llanmadir. Bazi xollarda asos sinf abstrakt bo'lib ma'lumotlar bazasida unga bog'liq ma'lumotlarga ega emas. Bu xolda bu sinf uchun ma'lumotlar bazasida ob'ekt mavjud bo'lmaydi.

Uch bo'g'inli arxitektura

Shu paytgacha biz WWW va biznes amaliy dasturlari bilan ishlash eng sodda arxitekturasi klient/server arxitekturasini muxokama qildik. Lekin bu arxitekturani Amaliy dasturlar rivojlanishi bilan takomillashtirish ancha murakkabdir. Bu arxitekturada ob'ektga-yo'naltirilgan dasturlash imkoniyatlaridan foydalanish ham qiyin. Birinchi muammo «nozik klientlar» haqidagi baxslarda o'z aksini topdi. Nozik klientlarga bo'lgan talab, klientga uzatilayotgan ma'lumotlar o'sib boish tendetsiyasidan kelib chiqdi. Bu muammo PowerBuilder va VisualBasic larda ko'rindi. Ular bazadan ma'lumotlarni GUI ga oladi va bu ma'lumotlar ustidagi hamma amallarni GUI da bajaradi.

Foydalanuvchi interfeysini baza yadrosiga bog‘lab qo‘yish foydalanuvchilar soni va ma’lumotlar xajmi oshishi bilan o‘zgartirish va masshtablash qiyin bo‘lgan dasturlar yaratilishiga olib keladi. Agar sizda foydalanuvchi interfeysi yaratish tajribasi bo‘lsa, foydalanuvchi xoxishiga qarab interfeysni qayta ishlab chiqish muammosiga duch kelgansiz. Bunday qayta ishlashni kamaytirish yo‘li GUI uchun faqat bita vazifa – foydalanuvchi interfeysi vazifasini qoldirish kerak. Foydalanuvchi bunday interfeysi chindan ham nozik klientdir.

Masshtablanishga ta’sir o‘tkazish boshqa tomondan ham ko‘rinadi. Agar foydalanuvchilar soni va ma’lumotlar xajmi oshgani munosabati bilan amaliy dasturni qayta ishlab chiqish kerak bo‘lsa, modifikatsiya ma’lumotlar bazasiga o‘zgartirish kiritish yo‘li bilan amalga oshirilishi mumkin. Masalan ma’lumotlar bazasini bir necha serverlarga taqsimlash yo‘li bilan. Interfeysni ma’lumotlar bazasiga bog‘lab qo‘yish masshtablash muammosini xal qilish uchun GUI ni o‘zgartirishga majbur qiladi. Aslida esa bu server bilan bog‘liq muammolardir.

Nozik klientlar – bugunda yagona yo‘nalish emas. Boshqa yo‘nalish- koddan qayta foydalanish. Xaer xil amaliy dasturlar uchun kod biznes logika deb atalgan qayta ishlashga yo‘naltiriladi. Agar biznes logika foydalanuvchi interfeysida joylashgan bo‘lsa, koddan qayta foydalanishni ta’minlash qiyin bo‘ladi. Bu muammoni xal qilish yo‘li Amaliy dasturni ikki qismga emas uch qismga ajratishdir. Bunday arxitektura uch bo‘g‘inli deyiladi.

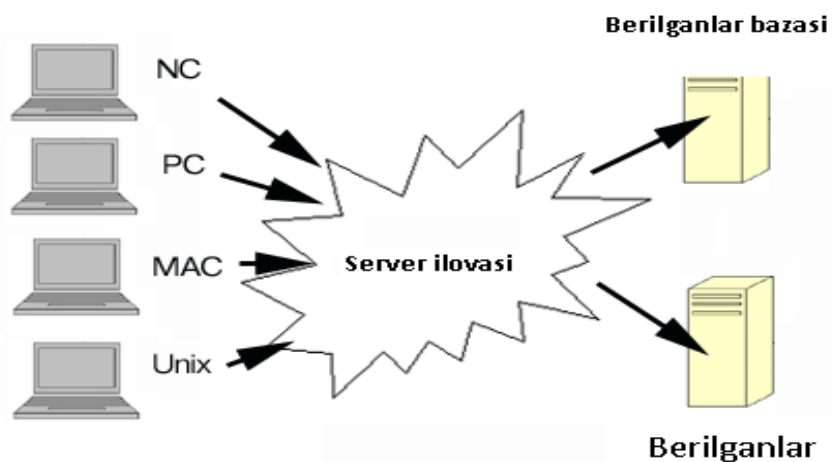
Klientdagi foydalanuvchi interfeys haqida gapirganimizda, mantiqiy farqni nazarda tutamiz. Nozik klient bir turi «O‘ta nozik klient», bo‘lib ko‘pchilik Web-sahifa deb qabul qiladi. Web-sahifa dinamik tarzda Web-serverda yaratilishi mumkin. Bu holda klient ishining ko‘p qismi serverda HTML-sahifalarni dinamik generatsiya qilish shaklida bajariladi.

2.1.1-rasmda ko‘rsatilgan ikki bo‘g‘inli arxitekturani 2.1.4-rasmda ko‘rsatilgan uch bo‘g‘inli arxitektura bilan solishtiring. Biz foydalanuvchi interfeys va ma’lumotlar bazasi orasida qo‘shimcha qatlam joylashtirdik. Bu Yangi qatlam amaliy dasturlar serveri o‘zida biror soha uchun umumiy bo‘ldgan Amaliy dastur ish mantiq‘i – biznes mantiqni oladi. Klient o‘rta yarus

ob'ektlarini ko'rish vositasi, ma'lumotlar bazasi bo'lsa shu ob'ektlar omboriga aylanadi.

Sizni eng katta yutug'ingiz, - foydalanuvchi interfeysi bilan ma'lumotlar bazasini ajratilishidir. Endi sizga ma'lumotlar bazasi haqidagi ma'lumotni GUI ga kiritishga to'g'ri kelmaydi. Balkim, ma'lumotlar bazasi bilan qanday ishlash haqidagi hamma ma'lumotlar, o'rta yarusda joylashishi mumkin.

Amaliy dasturlar serverining ikki asosiy vazifasi – ma'lumotlar bazasiga ulanishlarni izolyatsiya qilish va biznes mantiq uchun markazlashgan omborni ta'minlash. Foydalanuvchi interfeysi faqat ma'lumotlarni kiritish va akslantirish bilan shug'ullanadi, ma'lumotlar bazasi yadrosi bo'lsa faqat ma'lumotlar bazasi muammolari bilan shug'ullanadi. Ma'lumotlarni qayta ishlashni markazlashtirish Amaliy dasturlar serverining bitta dasturini har xil foydalanuvchi interfeyslari ishlatishi mumkin va har gal Yangi amaliy dastur yaratilganda ma'lumotlarni qayta ishlash qoidalarini yozish kerak bo'lmay qoladi.



2.1.4-rasim Uch bo'g'inli arxitektura

2.1.5-rasim Ma'lumotlar bazalariga ob'ektga-yo'naltirilgan murojaat va ODBC

Agar siz MySQL uchun yaratilgan dasturni boshqa MBBT ga ko'chirmoqchi bo'lsangiz o'z kodingizni shu yadro API sidan foydalanadigan qilib qayta yozishingiz kerak.

Lekin dasturchilar boshqa ma'lumotlar bazasiga ko'chirish muammosidan asosan xalos bo'lganlar. Ularda yagona API, Open DataBase Connectivity API (ODBC), hamma SQL-ma'lumotlar bazalariga unifikatsiya qilingan interfeys mavjud.

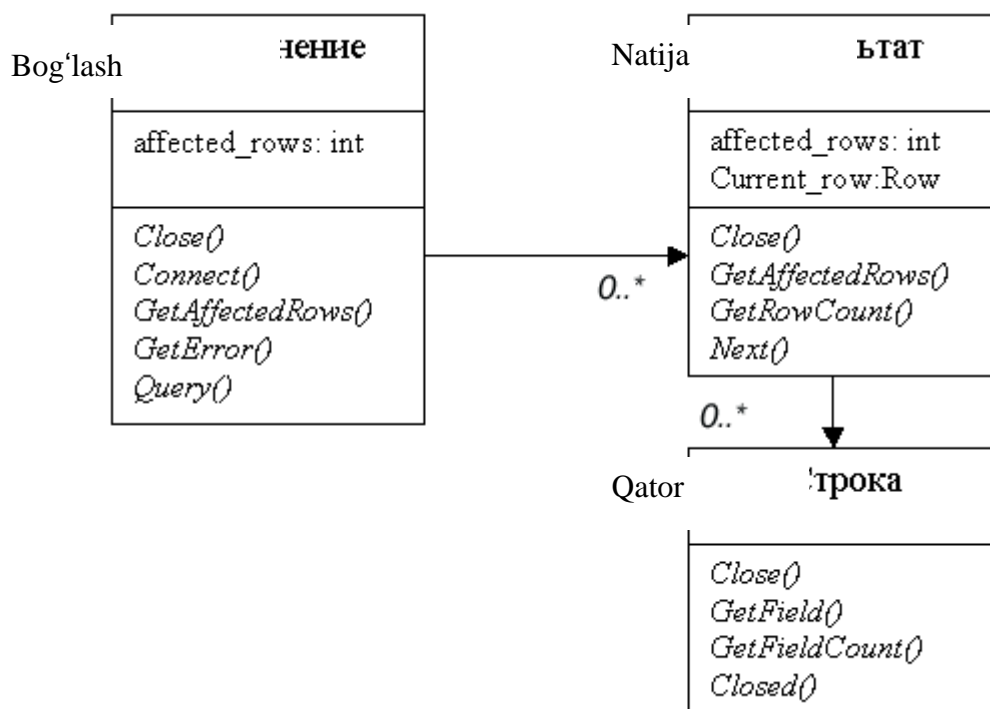
ODBC hamma ma'lumotlar bazalariga yagona interfeys bo'lgani uchun, MySQL va boshqa MBBT lar bilan ishlovchi dasturlar yaratish uchun uni o'rganib chiqish etarli. Agar siz kerakli tarzda ODBC dan foydalansangiz, siz yaratgan dasturlar ixtiyoriy MBBT bilan ishlay oladi.

ODBC haqida tushuncha

Hamma API lar kabi ODBC birgalikda ma'lum funktsiyalar to'plamini ta'minlovchi sinflar va interfeyslar to'plamidir. ODBC xolida bu funktsiyalar ma'lumotlar bazasiga murojaatni ta'minlaydi. ODBC API ni tashkil qiluvchi sinflar va interfeyslar ixtiyoriy turdagi ma'lumotlar murojlat qilishdagi umumiy tushunchalar abstraktsiyasidir.

Masalan, Connection ma'lumotlar bazasi Bilan bog'lanishni tasvirlovchi interfeysdir. Shunga o'xshab ResultSet SQL SELECT komandasi qaytaruvchi natijaviy to'plamni tasvirleydi. Tabiiyki ma'lumotlar bazasiga murojaat konkret detallari uning yaratuvchisiga bog'liq. ODBC bu detallar Bilan ishlamaydi.

ODBC sinflarini ma'lumotlar bazasini dasturlashga ob'ektga-yo'naltirilgan usullar nuqtai nazaridan ko'rib chiqamiz.



2.1.5-rasim Ma'lumotlar bazasiga ob'ektga-yo'naltirilgan murojaat bibliotekasi

MBBT Bilan ishlashni uchta asosiy tushuncha tasvirlaydi: ulanish, natijaviy to'plam va natijaviy to'plam satrlari. 2-5 rasm bu ob'ektlarni UML-diagrammada ko'rsatadi.

UML - bu Yangi Unifikatsiyalangan modellashtirish tili bo'lib, sozdanny Gradi Buch, Ayvar Yakobson va Djeys Rambo (Grady Booch, Ivar Jacobson, James Rumbaugh) tomonidan ob'ektga-yo'naltirilgan loyihalash va taxlilni xujjatlash Yangi standarti sifatida taklif qilingan.

Ma'lumotlar bazasiga ulanish:

Ixtiyoriy muxitda ma'lumotlar bazasiga murojaat ulanishdan boshlanadi. Bizning ob'ektga-yo'naltirilgan bibliotekamizni yaratish Connection ob'ektini yaratishdan boshlanadi. Ob'ekt Connection server Bilan bog'lanishni o'rnatish, zarur ma'lumotlar bazasini tanlash, so'rovlarni uzatish va natija olishni bilishi kerak.

Connection sinfi usullari hamma MBBT lar uchun bir xildir. Lekin sinf ichida, kompilyatsiya qilinayotgan biblioteka uchun xos bo'lgan yopiq a'zolar

berkitilgandir. Bog‘lanish o‘rnatishda ma’lumotlar Bilan bog‘lanishni ta’minlaydigan sinf a’zolari farqli bo‘lib qoladi.

Ma’lumotlar bazasi bilan ulanishni o‘rnatish:

Bu API yordamida yaratiladigan hamma amaliy dasturlarga ma’lumotlar bazasiga ulanish uchun Connection sinfi nusxasini uning konstruktorlaridan biri yordamida yaratish kerak bo‘ladi. U kabi uzilish uchun Amaliy dastur Connection nusxasini o‘chirishi kerak. U to‘g‘ridan to‘g‘ri Close() va Sonnect() usullariga murojaat qilib Connection nusxasini qaytadan ishlatishi mumkin.

Ma’lumotlar bazasidan uzilish:

Connection yana bir mantiqiy funktsiyasi ma’lumotlar bazasi bilan aloqani uzish va dasturdan berkitilgan resurslarni ozod qilishdir. Bu funktsiyani Close () usuli amalga oshiradi.

Ma’lumotlar bazasiga murojaatlarni bajarish:

Bog‘lanishni ochish va yopish odatda ma’lumotlar bazasiga komandalar yuboriladi. Connection sinfi argument sifatida SQL komanda oluvchi Query() usuli yordamida bajaradi. Agar komanda so‘rov bo‘lsa 2-5. rasmda ko‘rsatilgan ob’ekt modelidan Result sinfi nusxasini qaytaradi. Agar komanda ma’lumotlarni Yangilayotgan bo‘lsa, usul NULL qaytaradi va affected_rows qiymatini o‘zgartirilgan satrlar soniga teng qiladi.

Natijaviy to‘plamlar:

Result sinfi natijaviy to‘plam ma’lumotlariga hamda shu natijaviy to‘plam bilan bog‘liq meta ma’lumotlarga murojaatni ta’minlashi kerak. 2-5 rasmda ko‘rsatilgan ob’ektlil modelga asosan bizning Result sinfimiz natijaviy to‘plam satrlarinisikl bo‘yicha o‘qish va undagi satrlar sonini aniqlashni ta’minlaydi.

Natijalar bo‘yicha ko‘chish:

Bizning Result sinfimiz natijaviy to‘plam Bilan qatorma qator ishlaydi. Result sinfi nusxasini Query() usuli yordamida olgandan so‘ng amaliy dastur to navbatdagi Next() usuli 0 qaytarmaguncha, ketma ket Next() va GetCurrentRow() usullarini chaqirishi lozim.

Satrlar:

Natijaviy to'planning aloxida satri bizning ob'ektlı modelimizda Row sinfi bilan tasvirlanadi.

Ma'lumotlarga murojaat massiv indeksi bo'yicha so'rov tomonidan berilgan tartibda amalga oshiriladi. Masalan, `SELECT user_id , password FROM users` so'rov uchun indeks 0 foydalanuvchi nomini va indeks 1 – parolni ko'rsatadi. Bizning C++ API bu indeklashni foydalanuvchi uchun do'stona qiladi. `GetField(1)` Yoki `fields[0]` birinchi maydonni qaytaradi.

2.2. ORACLE ma'lumotlar bazasini boshkarish tizimi va uning avzalliklari

Xozirgi kunda bir qancha tizimlar ishlab chiqilgan ularning xar biri o'zida bir qancha maxsulotlarni o'zida mujassam qilgan misol uchun Oracle 8, Oracle 9, Oracle 10d. maxsulotlar to'plami o'zidan kelib chiqqan xolda shaxsiy SUBD ni o'ziga keltirgan (Oracle Database 10d, Oracle Database 11d),xuddi shunday ma'lumotlarni qayta ishlash va analiz qilish .

Tizim xaqidagi malumotlarni Oracle serverida keltiramiz, Oracle qulay ,umumiy, ochiq texnologik yechimlardan foydalanish taklif qilinadi.

Tayyor yechimlar avtamatik tarzda o'zida butun ma'lumotlar bazasini; servis ko'rsatmalarini, integratsyani platformani, analitika qilish jixozlarini va strukturalashtirilgan malumotlarni boshqaradi. Masshtablashtirilgan biznes- Oracle ilovasi ITdan osongina integratsya qilishi mumkun- bunda tashkilot infrastrukturası IT qo'yilgan investitsyani o'zida saqlab qoladi (yani talofat yetmaydi). SUBD Oracle Database 11d axborot ximoyasi qamrovida ma'lumotlashtirish masalasini avtomatlashtirilganligini va imkoniyatlar olamida eng yaxshi ta'millangan xavsizlik bo'yicha va normativlardan kelib chiqqan xolda o'zida yanada yangilangan xarakteristikalar bilan ta'millangan.

Yanada keng avtomatlashtirish funksyalari o'zini o'zi diagnostika qilish va boshqarish imkonyati paydo bo'ldi. Tizim xarakteristikasi orasidan bo'lingan tablitsyalar va kompressyalarning katta miqtorida foydalanishini boshqarishini inobatga olishimiz mumkin.

2.3. My SQL ma'lumotlar boshqarish tizimidan foydalanishning axamiyati

Quyida qanday qilib klient dasturi mysql ga ulanishni ko'ramiz. Bu dastur yordamida MySQL-serverga ulanish, SQL-so'rovlarni bajarish va shu so'rovlar natijalarini ko'rib chiqish mumkin. Bu qsimni o'rganish uchun kompyuteringizda utilita mysql o'rnatilgan va MySQL serveri bilan boo'langan bo'lishi kerak.

MySQL serveriga mysql dasturi yordamida bog'lanish uchun foydalanuvchi nomini va odatda parol kiritish lozim. Agar server va klient har xil mashinalarda joylashgan bo'lsa, MySQL serveri ishga tushirilgan xost nomini ko'rsatish lozim:

```
shell> mysql -h host -u user -p
```

Shundan so'ng ekranda quyidagi so'rov paydo bo'ladi Enter password:, va sizga o'z parolingizni kiritishingiz kerak bo'ladi. Agar ulanish to'g'ri amalga oshgan bo'lsa, ekranda quyidagi ma'lumot va komanda satri belgisi paydo bo'ladi mysql>:

```
Welcome to the MySQL monitor. Commands end with ; or \g.
```

```
Your MySQL connection id is 459 to server version:
```

```
Type 'help' for help.
```

```
mysql>
```

Quyidagi mysql> belgining paydo bo'lishi mysql dasturi ishga tayyorligini bildiradi.

Serverdan ixtiyoriy paytda QUIT komandasini terib uzilish mumkin:

```
mysql> QUIT
```

```
Be
```

```
Izox
```

Odatda MySQL lokal mashinaga Yangi o'rnatilgan bo'lsa, murojaat parol va xost kiritilmasdan, komana qatorigamysql komandasini kiritish yo'li Bilan amalga oshiriladi.

Serverga ulangandan so'ng komandalar sintaksisini o'rganish uchun bir necha sodda so'rovlar berishingiz mumkin. Hali hech qanday ma'lumotlar bazasi tanlanmagani uchun quyida keltirilgan so'zrovlar umumsiy harakterga ega.

Quyida serverdan versiyasi va vaqtni so'raydigan sodda komandani keltiramiz:

```
mysql> SELECT VERSION(), CURRENT_DATE;
```

MySQL ning bu so'rovga javobi quyidagi jadvaldan iborat:

```
+-----+-----+
| version() | current_date |
+-----+-----+
1 row in set (0.02 sec)
```

Bu so'rovni bajarish misolida MySQL bilan ishlash asosiy xususiyatlarini ko'rish mumkin:

- Serverga yuborilayotgan komanda, odatda SQL-ifodadan iborat bo'lib, ketidan nuqta vergul keladi. Bu qoidadan chekinishlar bor, masalan QUIT komandasidan so'ng nuqta vergul qo'yilmaydi;
- MySQL so'rov natijasini jadval shaklida chiqaradi;
- So'rov natijalaridan iborat jadvalni chiqargandan so'ng, mysql qaytarilgan satrlar soni va so'rov bajarish vaqtini ko'rsatadi. Bu qulay, chunki server unumdorligini va so'rov bajarish effektivligini baxolashga imkln beradi;
- So'rov natijalari va bajarilish vaqtinichi qargandan so'ng, mysql Yangi mysql> satrni chiqaradi, bu esa Yangi komandalar bajarishga tayyorligini ko'rsatadi.

MySQL komandalari registrga bog'liq emas, shuning uchun quyidagi so'rovlar bir xildir:

```
mysql> select version(), current_date;
```

```
mysql> SELECT VERSION(), CURRENT_DATE;
```

```
mysql> Select Version(), Surrent_DATE;
```

MySQL bir satrga bir necha komandalarni joylashtirishga imkon beradi, lekin ular har biri nuqta vergul bilan tugashi kerak. Masalan:

```
mysql> SELECT VERSION(); SELECT NOW();
```

Bunday so'rovga quyidagi natijani olamiz:

```
+-----+
| version() |
+-----+
| 4.0.13-nt |
+-----+
1 row in set (0.00 sec)

+-----+
| NOW()          |
+-----+
| 2004-01-25 16:57:00 |
+-----+
1 row in set (0.03 sec)
```

Lekin hamma komandalarni bir satrga joylash shart emas:

```
mysql> SELECT USER(),
-> CURRENT_DATE;
```

Natija:

```
+-----+-----+
| user()          | current_date |
+-----+-----+
| ODBC@localhost | 2004-01-25   |
+-----+-----+
1 row in set (0.00 sec)
```

E'tibor berinki biz Yangi satrga o'tgandan so'ng, komanda satri belgisi mysql> dan -> ga o'zgardi. Bu bilan mysql tugatilgan so'rov olinmaganligini va so'rov oxirini kutayotganligini bildiradi. Bu belgi juda foydali, chunki ba'zi xatolar

oldini olishga imkon beradi. Agar siz so'rov oxirida nuqia vergul qo'yishni unutgan bo'lsangiz, mysql bu to'g'rida -> belgini chiqarib bildiradi:

```
mysql> select user()
```

->

MySQL dan soda kalkulyator sifatida foydalanish uchun maslan quyidagi so'rovni kiritish kerak:

```
mysql> select cos(pi()/10), (2*5)-5;
```

Ma'lumotlar bazasiga murojaat huquqini berish

MBBT MySQL o'z ma'lumotlar bazalariga murojaat qilish huquqlarini berish uchun maxsus ma'lumotlar bazasidan foydalanadi. Bu huquqlar serverlar va yoki foydalanuvchilar nomlariga asoslangan bo'lishlari va bir Yoki bir necha ma'lumotlar bazalari uchun berilishlari mumkin.

Foydalanuvchilar akkauntlari parollar bilan ta'minlangan bo'lishi mumkin Ma'lumotlar bazasiga murojaat qilinganda parollar shifrlanadi. Shuning uchun uni o'zgalar bilib olib foydalana olmaydi.

Yangi foydalanuvchilar qo'shishga misol:

```
$ mysql mysql
```

```
mysql> INSERT INTO user VALUES ('%','monty',password('something'),
```

```
-> 'Y','Y','Y','Y','Y','Y','Y','Y','Y','Y');
```

```
mysql>          INSERT          INTO          user          (host,user,password)
values('localhost','dummy','');
```

```
mysql>          INSERT          INTO          user          VALUES
('%','admin','','N','N','N','N','N','N','Y','N','Y','N');
```

```
mysql> quit
```

```
$ mysqladmin reload
```

Uchta Yangi foydalanuvchi qo'shilgan:

monty: Superfoydalanuvchi (administrator), mysql bilan ishlash uchun paroldan foydalanishi kerak.

dummy: Individual ma'lumotlar bazasiga 'db ' jadval bo'yicha murojaat qilishi mumkin.

admin: Parol kerak emas, lekin faqat 'mysqladmin reload' and 'mysqladmin processlist' komandalarini bajarishi mumkin. Individual ma'lumotlar bazasiga 'db ' jadval bo'yicha murojaat qilishi mumkin.

Parolga ega foydalanuvchi yaratish uchun password() funktsiyasidan foydalanish kerak. MBBT MySQL shifrlangan parol olishni kutadi.

Foydalanuvchilar jadvalidagi atributlar DB jadvalidagi atributlarni berkitadi. Agar server ko'p ma'lumotlar bazalarini qo'llasa yaxshisi foydalanuvchilar jadvaliga murojaat qilish huquqiga 'ga bo'lmagan foydalanuvchilarni yaratish va ularga db jadvali bo'yicha ma'lumotlar bazasiga murojaat qilish huquqini berish kerak.

Agar siz MIT threads package dan foydalansangiz, shunga e'tibor beringki localhost nom ishlamaydi, chunki MIT threads package socket-ulanishni qo'llamaydi. Bu shuni bildiradiki siz ulanishda har doim hostname (server nomini) aniqlashingiz kerak, bita server bilan ishlasangiz ham.

Murojaat huquqlarini sozlashda quyidagi qoidalarga amal qilish kerak:

- Server nomi va db jadvaldagi maydon nomi SQL tili regulyar ifodalarini o'z ichiga olishi mumkin: % va _. Boshqa maydonlarda ulardan foydalanish mumkin emas.

- Server nomi domenli nom, localhost nomi, IP adres Yoki SQL ifoda bo'lishi mumkin. Bo'sh maydon " server nomi" ixtiyoriy serverni bildiradi.

- Maydon db ma'lumotlar bazasi nomi Yoki SQL ifodadir.

- Bo'sh foydalanuvchi nomi ixtiyoriy foydalanuvchiga ekvivalentdir.

- Bo'sh parol ixtiyoriy parolga ekvivalent. Siz superfoydalanuvchi(super-user) yaratishingiz mumkin, unga

foydalanuvchilar jadvali 'Y' gida hamma huquqlarni oʻrnatish yoʻli bilan. Bu foydalanuvchi DB jadvalidagi qiymatlarga qaramasdan ixtiyoriy oʻzgartishni qilishi mumkin!

— Severlar jadvali tekshiriladi faqat va vaqat db jadvalida "server nomi" maydoni boʻsh boʻlsa.

— Hamma jadvallar host-user-db boʻyicha tartiblanishi mumkin.

Maʼlum foydalanuvchi maʼlum maʼlumotlar bazasiga qanday murojaat qilishini hisoblab chiqish uchun server 3.20.19 versiyasidan boshlab quyidagi ayyorlik kiritilgan. Masalan quyidagicha registratsiyadan oʻtgan Djo nomli foydalanuvchi mavjud boʻlsin:

```
INSERT INTO user VALUES('%external.domain.com',' ','N','N','N','N','N','N','N','N','N','N');
```

Maʼlumotlar bazasi jadvalini tekshirish foydalanuvchi Djo uchun emas " (boʻsh nom) foydalanuvchi uchun bajariladi, hatto Djo maʼlumotlar bazasiga murojaatlar jadvalida(db jadvalda) yozuvga ega boʻlsa ham.

Foydalanuvchi akkauntlarini yaratish MBBT MySQL ningeng murakkab koʻrinuvchi tomonidir. mysqlaccess dasturidan foydalanish murojaatni boshqarishni aniqroq qiladi.

Umuman olganda server nomi maydonlarida SQL ifodalardan foydalanmaslik kerak. Bu sozlashni osonlashtiradi.

Aniqroq " server nomi " hamma maydonlarini '%' ga oʻrnating va serverlar jadvalini tozalang. Agar xato olinmasa, serverlar jadvaliga serverlar nomini kitirib koʻrish mumkin.

Agar 'Access denied' xatoligi olinsa, demak siz mysqld demoni bilan toʻgʻri bogʻlangansiz, lekin foydalanuvchilar jadvalida notoʻgʻri maʼlumotga egasiz.

Parollar qanday ishlaydilar

SHifrlangan parol foydalanuvchilar jadvalida saqlanadi.

— Ulanish oʻrnatilganda server klientga tasodifiy son yuboradi.

— Klient serverdagi maxlumotni olish uchun parolni shifrlaydi va serverdagi olingan tasodifiy son hamda parol asosida Yangi son hisoblaydi. Bu son serverga yuboriladi.

— Server saqlangan shifrlangan parol va tasodifiy son asosida Yangi son hisoblaydi. Agar bu son klient yuborgan songa teng bo'lsa bog'lanish o'rnatiladi.

MySQL ma'lumotlar bazasida ishlatiladigan ma'lumotlar tiplari

Butun sonlar

Ma'lumotlar	tipini	ko'rsatish	umumiy	formasi:
prefiksINT				[UNSIGNED]

Shart bo'lmagan bayroq UNSIGNED iorasiz sonlar (o' ga teng Yoki katta) sonlar saqlash uchun maydon yaratishni bildiradi.

TINYINT	Diapazoni -128 dan 127 gacha sonlarni saqlashi mumkin
SMALLINT	Diapazoni -32 768 dan 32 767 gacha sonlarni saqlashi mumkin
MEDIUMINT	Diapazoni -8 388 608 dan 8 388 607 gacha sonlarni saqlashi mumkin
INT	Diapazoni -2 147 483 648 dan 2 147 483 647 gacha sonlarni saqlashi mumkin
BIGINT	Diapazoni -9 223 372 036 854 775 808 dan 9 223 372 036 854 775 807 gacha sonlarni saqlashi mumkin

Kasr sonlar

MySQL da butun sonlar bir necha turga ajratilganidek, kasr sonlar ham bir necha turga ajratiladi.

Umumiy	xolda	ular	quyidagicha	yoziladi:
TipNomi[(length, decimals)] [UNSIGNED]				

Bu erda – kasr uzatilishda joylashadigan belgi joylari soni (maydon kengligi).

decimals – oʻnli nuqtadan soʻng hisobga olinuvchi raqamlar soni.
UNSIGNED – ishorasiz sonlarni beradi.

FLOAT Aniqligi uncha katta boʻlmagan suzuvchi nuqtali son.

DOUBLE Ikkilik aniqlikka ega boʻlgan suzuvchi nuqtali son.

REAL DOUBLE uchun sinonim.

DECIMAL Satrlar shaklida saqlanuvchi kasr son.

NUMERIC DECIMAL uchun sinonim.

Satrlar

Satrlar simvollar massivlaridan iborat. Odatda SELECT soʻrovi boʻyicha matnli maydonlar boʻyicha izlashda simvollar registri hisobga olinmaydi, Yaʼni "Vasya" va "VASYA" satrlari bir xil hisoblanadi. Agar maʼlumotlar bazasi matni joylashtirish va oʻqishda avtomatik qayta kodlashga sozlangan boʻlsa, bu maydonlar siz koʻrsatgan kodlashda saqlanadi.

Oldiniga *length* dan oshmagan simvollar saqlovchi satrlar tiplari Bilan tanishamiz.,
length prinadlejit diapazonu ot 1dan to255 gacha boʻlgan diapazonda yotadi.

VARCHAR (length) [BINARY]

Bu tipdagi maydonga biror qiymat kiritilganda undan oxirini koʻrsatuvchi simvollar avtomatik ravishda qirqib olinadi. Agar BINARY bayrogʻi koʻrsatilgan boʻlsa, SELECT soʻrovda satr registri hisobga olgan xolda solishtiriladi.

VARCHAR 255 dan ortiq boʻlmagan simvollarni saqlashi mumkin.

TINYTEXT 255 dan ortiq boʻlmagan simvollarni saqlashi mumkin.

TEXT 65 535 dan ortiq boʻlmagan simvollarni saqlashi mumkin.

MEDIUMTEXT 16 777 215 dan ortiq boʻlmagan simvollarni saqlashi mumkin.

LONGTEXT 4 294 967 295 dan ortiq boʻlmagan simvollarni saqlashi mumkin.

Koʻpincha TEXT tpi qoʻllanadi, lekin maʼlumotlar 65 536 simvoldan oshmasligiga

ishonmasangiz, LONGTEXT tipidan foydalaning.

Sana va vaqt

MySQL sana va vaqtni har xil formatlarda saqlash uchun mo'ljallangan maydonlar bir necha tiplarini qo'llaydi.

DATE GGGG-MM-DD formatdagi sana

TIME CHCH:MM:SS formatdagi vaqt

DATETIME GGGG-MM-DD CHCH:MM:SS formatdagi sana va vaqt

timestamp formatdagi sana va vaqt. Lekin maydon qiymatini olishda u timestamp formatida emas, TIMESTAMP GGGGMMDDCHCHMMSS formatda aks etadi, bu esa PHP dan undan foydalanish qiymatini ancha kamaytiradi.

Ma'lumotlar bazasini yaratish MySQL (CREATE DATABASE)

Ma'lumotlar bazasi CREATE DATABASE komandasi yordamida yaratiladi.

Komanda sintaksisi :

CREATE DATABASE database_name

— database_name – Ma'lumotlar bazasiga beriladigan nom.

Keyingi misolda db_test ma'lumotlar bazasini yaratamiz:

CREATE DATABASE db_test

PHP da ma'lumotlar bazasini yaratish:

```
$sql="CREATE DATABASE db_test";
```

```
mysql_query($sql);
```

MySQL ma'lumotlar bazasini o'chirish (DROP DATABASE)

Ma'lumotlar bazasini o'chirish uchun DROP DATABASE komandasidan foydalaniladi.

Sintaksis:

DROP DATABASE database_name

Bu erda

— database_name – o'chirish kerak bo'lgan ma'lumotlar bazasi nomi.

Quyidagi misolda db_test ma'lumotlar bazasi o'chiriladi:

```
DROP DATABASE db_test
```

PHPda ma'lumotlar bazasini o'chirish:

```
$sql="DROP DATABASE db_test";
```

```
mysql_query($sql);
```

```
USE
```

Jadvallar bilan ishlash uchun MySQL ga qaysi baza bilan ishlash haqida ma'lumot berish kerak. Bu USE komandasi yordamida amalga oshiriladi:

```
USE db_name;
```

Bu erda db_name – tanlangan ma'lumotlar bazasi nomi. Yaratilgan db_test bazasini tanlaymiz:

```
mysql> CREATE DATABASE db_test;
```

```
Database changed;
```

MySQL ma'lumotlar bazasida jadval yaratish (CREATE TABLE)

Jadval yaratish CREATE TABLE komandasi orqali amalga oshiriladi.

```
CREATE TABLE table_name(column_name1 type, column_name2 type,...)
```

— table_name – Yangi jadval nomi;

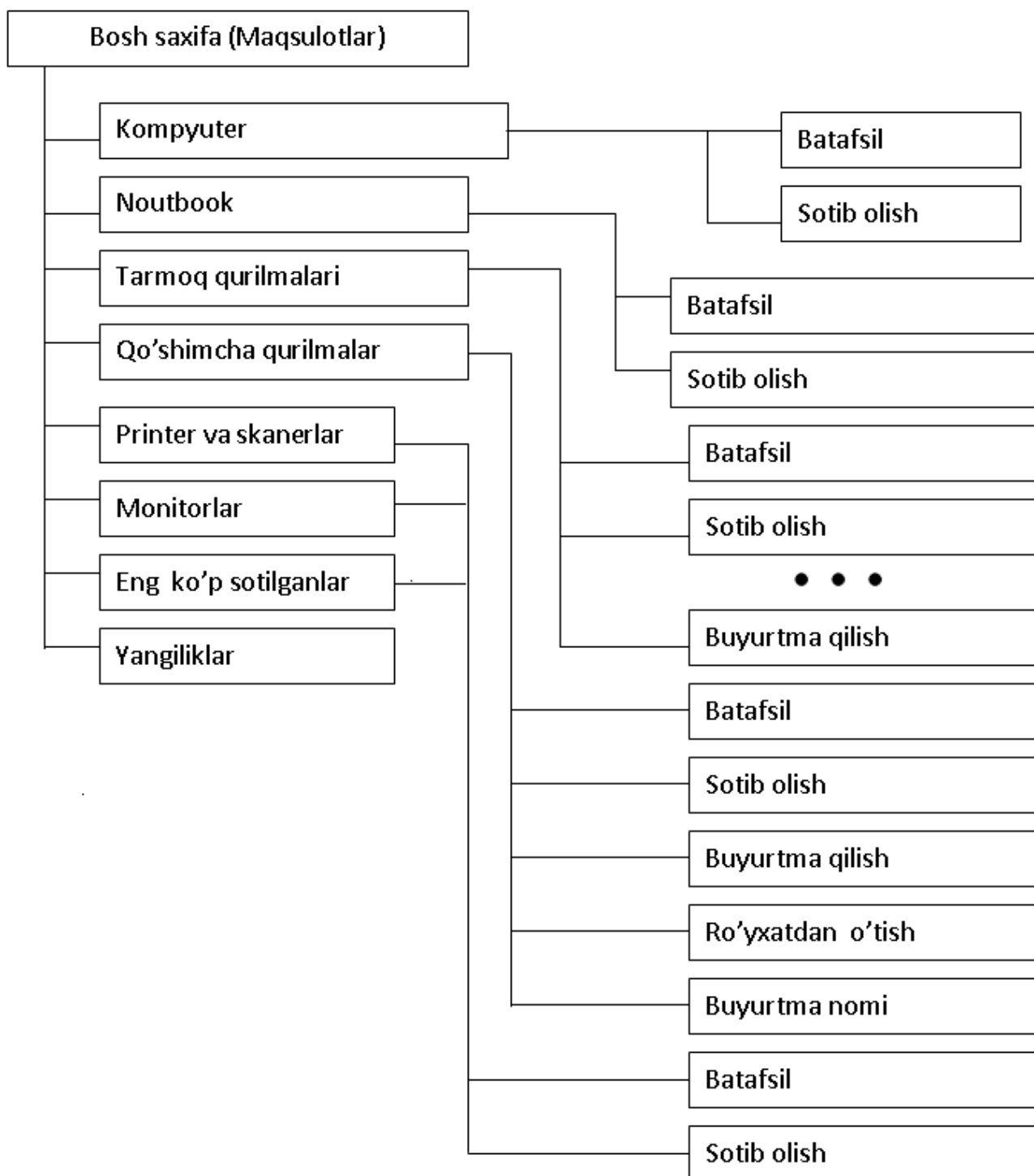
— column_name – yaratilayotgan jadval ustunlari
(maydonlari), nomlari.

— type – ustun tipi.

3. «IT-prise» dasturiy tizimini yaratish

3.1. Tizimning tafsifi

Yaratilayotgan tizimning umumiy strukturasi quydagi ko'rinishda bo'ladi:

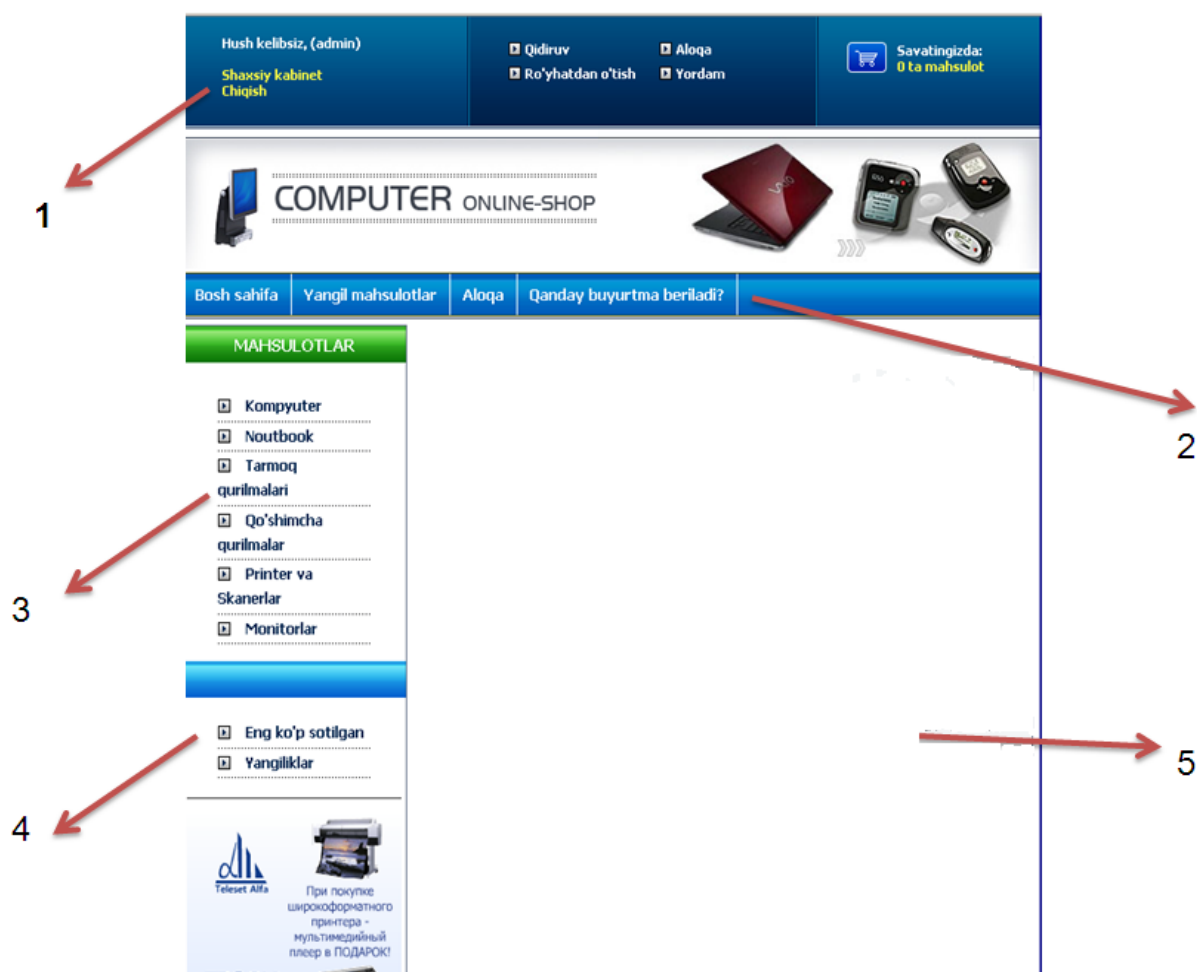


3.3.1-rasm. «IT-prise» dasturiy tizimini yaratish dasturining modeli

Tizimning ish jarayoni bosh sahifani yuklash orqali boshlanadi. Asosiy sahifada kompyuter, noutbook, tarmoq qurilmalari, qo'shimcha qurilmalar, printer va skaner, monitorlar, eng ko'p sotilganlar, yangiliklardan iborat menyu qatorlari mavjud bo'lib bular alixida sahifa ko'rinishida turadi. Bundan tashqari yuqorida sanab o'tilgan bo'limlar o'rtasida navigatsiya mavjut bo'lib bu boshqa bo'limlarga o'tish imkonini beradi.

3.2 Foydalanuvchiga qo'llanma

Yuqorida keltirilgan dastur modeliga asoslanib «IT-prise» dasturiy tizimining quydagicha tashkil qilindi (3.2.1-rasm).



3.2.1-rasm. «IT-prise» dasturiy tizimini asosiy oynasi.

Bu erda:

- 1 – Shaxsiy kabenet, qidruv, ro'yxatdan o'tish, aloqa va mahsulot savatchasi;
- 2 – Bosh sahifa, yangi mahsulotlar, aloqa, qanday mahsulotlarga buyurtma berish ;
- 3 – Kompyuter, noutbook, tarmoq qurilmalari, o'shimcha qurilmalar, printer va skanerlar, manitor;
- 4 – Eng ko'p sotilganlar;
- 5 – Dasturning o'shimch menyu qatori, bu qatorda xar bir asosiy menyu qatorining o'ziga tegishli qo'shimcham menyu qatori mavjud.

Birinchi saxifada kompyuter bo'limi joylashgan bo'lib, Web-saxifaga kirgan foydalanuvchi birinchi bo'lib kompyuter mahsulotlaridan xabardor bo'ladi- (3.2.2-rasm).



3.2.2-rasm. «IT-prise» dasturiy tizimini yaratish dasturining birinchi saxifasi

Ikkinchi sahifada noutbook bo'limi joylashgan bo'lib, bu Web-sahifada xar xil noutbook mahsulotlaridan xabardor bo'ladi va sotib olish mumkin. Shuningdek keying bo'limlarda noutbook (3.2.3-rasmda), tarmoq qurilmalari (3.2.4-rasmda), printer va skanerlar (3.2.5-rasmda), monitorlar (3.2.6-rasmda).


COMPUTER ONLINE-SHOP



[Bosh sahifa](#)
[Yangil mahsulotlar](#)
[Aloqa](#)
[Qanday buyurtma beriladi?](#)

MAHSULOTLAR

Mahsulotlar

▶ Kompyuter

▶ Noutbook

▶ Tarmoq qurilmalari

▶ Qo'shimcha qurilmalar

▶ Printer va Skanerlar

▶ Monitorlar

▶ Eng ko'p sotilgan

▶ Yangiliklar



FUJITSU ESPRIMO V6555

Parametrlari:
 Diagonal 15.4
 Prosesir turi Intel
 Pentium dual-core
 HDD 320 Gb ROM 2
 Gb DVD-RW
 Web-kamera bor

2413805 so'm

[Batafsil](#)
[Sotib olish](#)



MSI WIND U130

Parametrlari:
 Monitor 10" Prosesir
 turi Intel Atom HDD
 160 Gb ROM 2 Gb
 Web-kamera 1.3

1341003 so'm

[Batafsil](#)
[Sotib olish](#)



ASUS EEE PC 1001HA

Parametrlari:
 Monitor 10.1 Turi
 Intel Atom HDD 160
 Gb ROM 1 Gb
 Web-kamera bor

1341003 so'm

[Batafsil](#)
[Sotib olish](#)



SAMSUNG NP-N150-JA02KZ

Parametrlari:
 Monitor 10" Turi
 Intel Atom HDD 160
 Gb ROM 1 Gb
 Web-kamera bor

1341003 so'm

[Batafsil](#)
[Sotib olish](#)



ASUS K42F

Parametrlari:
 Monitor 14 Turi
 Intel Core i3 Qattiq
 disk hajmi 250 Gb
 HDD 2 Gb CD
 DVD-RW
 Web-kamera bor

2745111 so'm

[Batafsil](#)
[Sotib olish](#)



ACER ASPIRE 7736ZG-433G32MN 4 GB

Parametrlari:
 Monitor 17" Turi
 Intel Pentium Dual
 Core HDD 160Gb
 ROM 4 Gb CD
 DVD-RW
 Web-kamera bor

2808217 so'm

[Batafsil](#)
[Sotib olish](#)



Teleset Alfa

При покупке
 бизнес проектора –
 экран
 в ПОДАРОК!



EPSON
 EXCEED YOUR VISION

Copyright © 2010 TUIT. All Rights Reserved

3.2.3-rasm. Noutbook bo'limi

72

COMPUTER ONLINE-SHOP

[Bosh sahifa](#)
[Yangil mahsulotlar](#)
[Aloqa](#)
[Qanday buyurtma beriladi?](#)

MAHSULOTLAR

- ☐ Kompyuter
- ☐ Noutbook
- ☐ Tarmoq qurilmalari
- ☒ Qo'shimcha qurilmalar
- ☐ Printer va Skanerlar
- ☐ Monitorlar

☐ Eng ko'p sotilgan
 ☐ Yangiliklar

КУПИ

ПОДАРОК

Mahsulotlar

<p>ADSL MODEM EDIMAX AR-7025UMA Parametrlari: Turi Tashqi Interfeys USB 75727 so'm</p> <p>Batafsil Sotib olish</p>	<p>ADSL2+MODEM (ROUTER) TENDA D810R Parametrlari: Turi Tashqi Interfeys USB 75060 so'm</p> <p>Batafsil Sotib olish</p>
<p>TP-LINK TM-IP5600 Parametrlari: Turi ichki Interfeys PCI 31275 so'm</p> <p>Batafsil Sotib olish</p>	<p>WI-FI TP-LINK TL-WR1043ND Parametrlari: Wi-fi standart IEEE 802.11b ,IEEE 802.11g ,IEEE 802.11n 228308 so'm</p> <p>Batafsil Sotib olish</p>
<p>WI-FI TP-LINK TL-WR841ND Parametrlari: Wi-fi standart IEEE 802.11b ,IEEE 802.11g ,IEEE 802.11n 172013 so'm</p> <p>Batafsil Sotib olish</p>	<p>WI-FI TARMOQ KARTASI TP-LINK TL-WN851N Parametrlari: Wi-fi standart IEEE 802.11n 103208 so'm</p> <p>Batafsil Sotib olish</p>

3.2.4-rasm. Tarmoq qurilmalari bo'limi

COMPUTER ONLINE-SHOP

[Bosh sahifa](#)
[Yangil mahsulotlar](#)
[Aloqa](#)
[Qanday buyurtma beriladi?](#)

MAHSULOTLAR

- ☐ Kompyuter
- ☐ Noutbook
- ☐ Tarmoq qurilmalari
- ☐ Qo'shimcha qurilmalar
- ☒ Printer va Skanerlar
- ☐ Monitorlar

☐ Eng ko'p sotilgan
 ☐ Yangiliklar

КУПИ

ПОДАРОК

Mahsulotlar

<p>HP COLOR LASER JET 5550 A3 Parametrlari: Turi Lazerniy Rangli Rangli Qog'oz O'lchami A3 Olchami 577x640x704 m 11934179 so'm</p> <p>Batafsil Sotib olish</p>	<p>EPSON STYLUS C91 Parametrlari: Turi Struyyniy Rangli Rangli Qog'oz O'lchami A4 Olchami 435x165x219 mm 267019 so'm</p> <p>Batafsil Sotib olish</p>
<p>EPSON V30 Parametrlari: Olchovi A4 Interfeys USB 2.0 Olchami 430x41x280mm 337770 so'm</p> <p>Batafsil Sotib olish</p>	<p>CANON LIDE 25 USB Parametrlari: O'lchami 256 x 383 x 34 mm 197032 so'm</p> <p>Batafsil Sotib olish</p>
<p>EPSON PERFECTION V700 Parametrlari: Interfeys USB 2.0 FireWire O'lchami 308x152x502 mm 2361263 so'm</p> <p>Batafsil Sotib olish</p>	<p>EPSON STYLUS PHOTO T50 Parametrlari: Turi Struyyniy Rangli Rangli 547313 so'm</p> <p>Batafsil Sotib olish</p>

3.2.5-rasm. Printer va skanerlar bo'limi



3.2.6-rasmda Monitorlar bo'limi

Bundan tashqari tizim orqali mahsulotlarni xarid qilish imkoniyati mavjud. Agarda foydalanuvchining biror bir fakir muloxazalari mavjud bo'lsa quydagi keltirilgan oyna orqali yuborish imkoniyati mavjud (3.2.7-rasm).

COMPUTER ONLINE-SHOP

Bosh sahifa | Yangil mahsulotlar | Aloqa | Qanday buyurtma beriladi?

MAHSULOTLAR

- ☒ Kompyuter
- ☒ Noutbook
- ☒ Tarmoq qurilmalari
- ☒ Qo'shimcha qurilmalar
- ☒ Printer va Skanerlar
- ☒ Monitorlar

☒ Eng ko'p sotilgan

☒ Yangiliklar

Biz bilan aloqa

Kontakt ma'lumotlari:
 Tel: +998 (93) 564-29-09
 e-mail: karimbek_85@mail.ru

Xabar jo'natish:

Ism:

Shahar:

Email:

Xabar:

Jo'natish

Joylashgan manzili:

CH "Teleset Alfa"

EPSON
EXCEED YOUR VISION

3.3.9-rasm. Fkir muloxazalarni jo'natish bo'limi

Asosiy sahifada “Qanday buyurtma berish” bo’limi mavjud bo’lib bu bo’lim tavarlarni harit qilishga mo’ljallangan bo’lib, mahsulot xaqida ma’lumot olish va buyurtma berish imkoniyati mavjuddir (3.2.9-rasm).

Sotib olish uchun ko'rsatma



LCD MONITOR LG 1934SB

Parametrlari:
Turi LCD Monitor
18"

581159 so'm

Batafsil
Sotib olish

1. Buyurtma berish uchun avval, **sotib olish** tugmasini bosing, va siz qildagi oynaga o'tasiz.
2. Bu formadan **Buyurtma** tugmasini tanlang.

No	Nomi	Firmasi	Narxi	O'chirish
1	LCD Monitor LG 1934SB		581159	<input type="checkbox"/>
Umumiy narx			581159	<input type="checkbox"/>

Buyurtma

3. Kerakli ma'lumotlarni to'ldiring, va **Davom etish** tugmasini bosing

Buyurtmani rasmiylashtirish:

F.I.O: *

Telefon: *

E-mail: *

Turar joyi:*

Manzil(Ko'cha, uy, kvartira):*

Mo'ljal:

Karim Abdurahmonov

(+998) 97 788-90-22

karim@mail.ru

Yakkasaroy tumani

Yu.Otaboyev ko'cha,
15-uy, 25-kv.

Musiqachilar uyi

Davom etish >>

Dastur PHP tilida My SQL ma’lumotlarlar bazasini boshqarish tizimidan foydalangan xolda yaratildi. Dastur klent-server texnologiyasi asosida ishlaydi, bunda sahifalar va ma’lumotlar bazasi serverda saqlanadi. Foydalanuvchi brauzer dasturlaridan foydalangan xolda tizimda ishlaydi.

3.3. Dastur ishlash uchun qo'yiladigan talablar

Dastur ishlash uchun quydagi texnik va dasturiy talablar bo'lishi maqsadga muvofiq bo'ladi:

- protsessor 1.7 GGs dan kam bo'lmasligi,
- kamida 256 Mb operativ xotira,

- doimiy xotirada kamida 5 Gbayt bo'sh joy bo'lishi,
- tarmoq kartalarining mavjudligi va tarmoqqa ulangan bo'lishi talab qilinadi.

Dastur tarmoqda ishlashga mo'ljallangan bo'lib, barcha ma'lumotlar tarmoq orkali taqdim etiladi. Shuni hisobga olgan xolda dastur ishlashi uchun dasturiy vositalarga ko'yidagicha talablar qo'yiladi:

- operatsion tizim Windows XP yoki Windows Vista,
- Web-brouzer Internet Explorer ning turi 6.0 dan past bo'lmasligi,
- Serverda Web-server (Apache yoki IIS) o'rnatilgan (bu faqat bitta kompyuterda, ya'ni server vazifasini bajaruvchi kompyuterda) bo'lishi talab qilinadi.

4. Xayot faoliyati hafsizligi

4.1.Vibratsiya(titrash) ishlab chiqarishdagi zararli omil sifatida

Ishlab chiqarish korxonalarda mexanizmlarning harakati natijasida har xil titrashlar vujudga keladi. Titrash ish unimdorligini kamaytiribgina qolmasdan, balki uning sog'ligiga ham ta'sir ko'rsatishi mumkin bu ta'sirini oldi olinmasa, xavfli titrash kasalligiga olib keladi..Shuning uchun xam titrashga karshi kurash muhim ahamiyatga ega.

Titrash umumiy va qisman bo'lishi mumkin. Umumiy titrashda inson organizmining butunlay titrash ta'sirida buladi, kismanda esa inson organizmining ba'zi bir kismilarigina titrash ta'sirga tushadi.Transport vositalarini boshkaruvchilar yuk kutraish kranlarini boshkaruvchilar shtamp bosuvchilar umumiy titrash ta'sirida buladilar.. Kisman tatrash ta'sirida bulish kulda ishlatiladigan elektrov a pnevmatik qurilmalar bilan ishlovchilar tushadi.

Umumiy tirashning 0.7 Gs dan kichik bulgan chastotalari umuman titrash kasalligiga olib kelmaydi,ammo dengiz kasalga uxshash kasallikaolib keladi.Bunda ichki organizmida muvozanatni buzilishi kuzatilinadi.

Inson organizmining hamma kismilarida xar xil chastotadagi titrashlar mavjud. Masalan, odam boshi, bo'yni, yurak qismlariga titrashlar tizimi sifatida karalishi mumkinki,bu uziga yarasha ogirlikka ega bulgan prujina simon vositalar yordamida titrashlar vududga keladi va bu titrashlarni sundirishga xarakat kiluvchi qarshiliklar guruxi xam mavjud.Agar bu tirovchi qismilarga tashqaridan xuddi shu chastotadagi titrashlar ta'sir ko'rsatsa, organizmda rezonans vujudga kelishi mumkinki, bu titrashni bir necha o'n marta ortishga olib keladi. Bu esa o'z navbatida, organizm kismilarida siljishni vujudga keltiradi..

Masalan tik turib ishlaganda bosh,yelka, bo'yin va umurtqa qismilarining titrashi 4-6 Gs ni tashkil kiladi..Utirib ishlaganda boshning yelkaga nisbatan titrashi 25-30 Gs ni kupchilik ichki a'zolarining titrashi 6-9 Gs atrofida buladi.Xuddi shunday titrash chastotasiga tushganda yomon asorat kelib chikadi,ba'zan bu mexanik jarohatlarga olib kelishi mumkin.. Titrashning doimiy

ta'siri esa titrash kasalligiga olib keladi..Bunda titrashning markazi nerv tizimlariga ta'siri natijasida organizmning fiziologik funksiyalari buziladi. Bu xol bosh og'rigi, bosh aylanishi, uykunung yomonlashuvi, mehnat kobiliyatining susayishi, yurak faoliyatining buzilishi bilan ifodalash mumkin.

Qisman titrash kon tomirlardagi koni kuyiqalashtiradi.Bu xolat asosan tananing oxirgi kismlarida bo'lgan qo'l panjalaridan boshlanib, butun kulga utadi va yurakdankelayotgan konning utishini yomonlashtiradi va bu bilan kon ta'minoti susayadi. Shuning bilan birga titrash ta'siri tashki nerv tizimlari ishini yomonlashtiradi, bu esa terining sezish qobilyatini susaytiradi, pay qavatlarining qotib kolishiga olib keladi ,bugimlarda tuz yigiladi va bugimlar harakatini susaytiradi. Bu xolatlar ayniksa sovuk fasllarda kuchayadi. Titrash – mastavkam zich jismlarning mexanik tebranishlari yoki tebranma harakatlar.

F dan 16 Gs gacha mexanik tizimlar joylashuv amplitudasi bilan xarakterlanadi.

$A = (t)$, tezligi $V=f(t)$, jadalligi $a=(t)$.

Tebranish (titrash) tezligining logarifma darajasi

$L = 20 \lg V f(5(10-8,$

deya belgilanadi,

V – titrash tezligining m/sek irtacha kvadrat ahamiyati.

Titrashning aylanma detallar va harakatlanayotgan mexanizmlar, vallar, mashina shkivlari, dostgomlar, pnevmatik asboblarning noto'g'ri balansirovkasi, truba quvurlar bo'yicha suyuqlik va gazlarni transportirovka qilish oqibatida yuzaga keladi.

Titrashlar dinamik yuklar ta'sirida vujudga kelib, mashinalar, qurilmalar va ishlab chiqarish binolari poydevoriga uzatiladi, ular orqali yerga boradi.

Shuning uchun titrashlar va chayqalishlar katta masofaga yetib borib, boshqa binolar va inshootlarda chayqalish uosil qilishi mumkin. Texnologik va boshqa uskunalarni loyihalashtirishda ishlab chiqarish binolarida ish joylarida uskunalar

titrashi tegishli kattalikdan oshib ketmasligi va standartlar me'yorlariga javob berishini inobatga olish zarur.

Ishlab chiqarish titrashlarini me'yorlash muammosi 2 yunalishda hal qilinadi: injener (muxandis) – texnik va sanitariya – gigiyenik. Kuchli va sekin titrashlar ortasidagi chegara f chastotaga bog'lik tarzda hal quyiladigan titrash amplitudalarining me'yoriy kattaligining qiyshiq izgarishi sifatida kiritib chiqish mumkin.

$n = 400 \text{ ob/min}$ da $A = 0,2 \text{ mm}$

$n (2400 \text{ ob/min})$ da $A = 0,05 \text{ mm}$

$n < 300 \text{ ob/min}$ da $A (0,19 \text{ mm})$

Agar $A (0,19 \text{ mm})$ bo'lsa, qoniqarli bo'ldi hisoblanadi.

$A (0,15 \text{ mm})$ – yaxshi bo'ldi;

$A (0,1 \text{ mm})$ – a'lo bo'ldi.

Past chastotali mashinalar uchun TU-60-49 me'yorlariga ko'ra, A poydevorlar tebranishlari amplitudasi $0,2 \text{ mm}$ gacha hal quyiladi; siltov

$S = 2A = 0,4 \text{ mm}$ – hozirgi paytda bu kattalik mashinalar poydevorlarining hisob-kitobi tashkil qilish bahosining asosiy kriteriysi.

Insonga ta'siri xarakteri bo'yicha titrashlar umumiy, maxalliy, uyg'unlashtirilganga bo'linadi. Ko'ndalang, uzunasiga yoki aylanma tebranishlar mavjud.

Umumiy titrashlar tananing biror bir qismiga ta'sirida qon ta'minotining yomonlashuviga olib keladi. Bo'g'inlarning deformatsiyasiga va harakatlanishining sustlashuviga, terining sezishini kamayishiga olib keladi.

Insonning ichki organlarini hal chastota tebranishlariga ega tebranuvchi tizim sifatida qarab chiqish mumkin:

$f = 6 \text{ Gs}$ - insonning butun tanasi uchun;

$f = 8 \text{ Gs}$ – bosh va oshqozon uchun;

$f = 20 - 25 \text{ Gs}$ – boshqa organlar uchun;

$f > 25 \text{ Gs}$ - noxush.

Tashqi tebranishlarning ta'siri rezonans holatni chaqirishi va insonning ichki organlari chayqalishi va shikastlanishiga olib kelishi mumkin.

Titrash nafas organlari, yurak-bu\in va MAS faoliyatini izdan chiqaradi, k\urish-eshitish qobiliyatini sustlashtiradi. Uzoq va tez titrashda titrash kasalligi vujudga kelishi mumkin. Ayniqsa inson uchun bir vaqtning \uzida shovqin, titrash va past \marorat zarar.

Titrash kasb kasalliklar toifasiga kiradigan kasallik bulib ,uni davolash asosan boshlangich davrlarda natija beradi.Kasallikning orkaga kaytishi juda sekin boradi.Agar oldi olinmasa, kishi ishga yaroksiz bulib koladi.Bu kasallikning oldini olishning asosiy vositasi ish joylarida titrash normalarini belgilash dir.Titrash normalari gigiyenik va texnik normalarga bulinadi.

Me'yoriy xujjatlardagi "Mehnat xavfsizlik standartlar tizimi.Titrash,xavfsizlikning umumiy talablari"ga asosan titrashning inson organizmiga ta'siri nuqtai nazaridan yul kuyishi mumkin bulgan mikdori va titrashning gigiyenik tavsifnomasini baxolash usullari belgilangan .

Titrash normalari umumiy va kisman titrashlar asosida ayrim –ayrim xolda baxolanadi .Umumiy titrash normalari akliy mexnat bilan shugullanuvchilar ,titrash bilan boglik sexlar va titrashdan xoli bulgan zonalarda ish bajaradigan ma\ina va mexanizmlar turlari asosida belgilangan.

Ish joylarining titrash noimasi belgilanganda (pol mashinalarning asosiva boshkaruvchilar uchun utirgichlar) titrash tezligining logarifmik darajasi urta geometrik chastotalari 2,4,8,16,32,63Gs gacha belgilanadi. Kisman titrashda esa 16,32,63,125,250,500,1000 Gs gacha norma belgilanadi. Gigiyena normalari 8 soatli ish vakti uchun ishlab chikiladi

Titrashni kamaytirishdagi tadbirlarga mexanizatsiyalash va avtomatlashtirish ishlarini kuzda tutiladi. Titrash ta'sirini butunlay yukatishning birdan bir chorasi- butiun texnologiyani avtomatlashtirish va titrash zonalariga odamlarning kirmasligini ta'minlashdir. Xozirda titrashni kamaytirishni kuiydagi usullaridan foydalaniladi:

- titrashni ajralib chikayotgan manbadan kamaytirish:
- tarkalish yulida kamaytirish:
- maxsus ish sharoitini tashkil etish yuli Bilan titrash ta'sirini kamaytirish:
- shaxsiy muxofaza aslaxalaridan foydalanish:
- soglamlashtirish chora-tadbirlarini belgilash.
- titrashni dinamik sundirish –bunda tizinga titrovchi tayanch orkali ma'lum kuch kuyish natijasida,titrashni fundamentga utmasligini ta'minlash
- vibrodeempirlash usuli titrash energiyasini boshka turdagi energiyaga aylantirish xisobiga amalga oshiriladi:
- mashina elementlari va kurilish konstruksiyalarini uzgartirish yuli Bilan kamaytirish.

Titrashdan saklanish shaxsiy muxofaza choralarida:

Qo'l orkali utuvchi titrashga karshi kulkoplar,titrashdan saklovchi plastina va proklatkalar kiradi.

Titrashlarning sovuk sharoitlarda ta'sir darajasi ortib ketishni xisobga olib, ishchilarni qish vaktida issik kiyim va issik kulkoplar bilan ta'minlash tavsiya etiladi. Titrash kasligini oldini olish uchun maxsu ish rejimi tashkil kilinadi. Masalan, kul ishlatiladigan titrash tarkatuvchi mexanizm bilan ishlovchi ishchi sanitariya normalarini talabiga muvofik umumiy ish smenasining 2/3 kismida oshmagan mikdorda ishlashi mumkin..Ovqatlanish uchun tanaffus 40 minutdan kam bulmasligi va ikki marta chegaralangan tanaffus kilinadi. Bu tanaffuslar ish boshlangandan keyin taxminan 1-2 soat davomida 20 min. va tushlik tanaffusidan 2 soat utgach 30 min. Bulishi kerak.

Umumiy titrash ta'sirda ishlayotgan ishchi maxsus oyok kiyimlar kiyishi kerak. Titrashni o'lchaydigan asboblarga IShV-1 va VIP-2 asboblardan foydalaniladi.

4.2. Favkulodda vaziyatlar

Favkulodda vaziyatlarni oldindan bilish –tabiiy ofatlar, avariylar va halokatlar vaqtida yuz beradigan holatlar va sharoitlarni taxminlab aniqlashga asoslangandir. Bunda, uncha to'liq va aniq bo'lmagan ma'lumotlar asosida FX lar sodir bo'lish ehtimoli bor rayon hamda Favkulodda vaziyatlarning xarakteri va masshtabi baholanib, FX lar oqibatlarini bartaraf etishga qaratilgan ishlarning xarakteri va hajmi taxminan belgilanadi.

Hozirgi vaqtda seysmik rayonlar, sel oqimlari, suv bosimlar sodir bo'ladigan, qor kuchishi va boshqa kuchishlar sodir bo'lish xavfi mavjud bo'lgan joylar aniqlangan.

Shuningdek, katta halokatlarga va avriyalarga olib kelishi mumkin bo'lgan sanoat korxonalari ham belgilangan. bu uzok muddatli oldindan bilish deb tushiniladi. Oldindan bilish vazifasiga FX lar sodir bulish extimoli vaktini aniklash masalasi ham kiradi. Bunday aniqlashning qisqa muddatli oldindan bilish deb tushiniladi. Buning uchun hozirgi vaqtda quyosh aktivligi siklining o'zgarishi to'g'risidagi statistik ma'lumotlardan, yerning sun'iy yo'ldoshi yordamida olingan ma'lumotlardan, hamda meterologik, seysmik, vulqon, sel oqimi va boshqa stansiyalarning ma'lumotlaridan keng foydalaniladi.

Masalan, bo'ronlar, dengiz bo'ronlari, vulqonlar otilishi, sel oqimlarining bo'lish ehtimoli, meterologik Yer yo'ldoshlari yordamida aniqlanadi. Yer qimirlashlarni sodir bo'lish ehtimoli seysmik rayonlarda suv tarkibini kimyoviy taxlil kilish, tuprokning elastiklik, elektrik

va magnit xarakteristikasini o'lchash, quduqlardagi suv sathi o'zgarishini kuzatish, hayvonlar holatini kuzatish orqali aniqlanishi mumkin. Katta

o'rmonlardagi va yer osti torf yog'inlarining yashirin o'choqlari samolyot yoki Yer yo'ldoshi yordamida infraqizil nurlar orqali tasvirga olish asosida aniqlanadi.

FX lar sodir bo'lish asosida yuzaga kelish mumkin bo'lgan holat va sharoitlar matematik usullar asosida baholanadi. Bunda boshlang'ich ma'lumotlar sifatida yashirin xavf joyi, koordinatasi va moddalar hamda energiya zahirasi, aholi soni va joylashish zichligi; qurilishlar xarakteri, himoya inshootlarining soni va turi, ularning hajmi, meteorologik sharoitlar, joyning xarakteri qabul kilinishi mumkin.

FXlar vaqtida qutiladigan shart-sharoitlarni oldindan baholashda FX ning turiga bog'lik holda uning chegarasi, halokatli suv toshqini, yog'in va radiatsion, kimyoviy va bakteriologik zaharlanish o'choqlari, FX lar natijasida yuz berishi ehtimol qilingan o'limlar va material boyliklarni barbod bo'lishi, xalq xo'jalik ob'yektlaridagi zarar miqdori taxminan aniqlanadi.

Oldindan bilish va baholash ma'lumotlari birlashtirilib, taxlil asosida xulosalanadi va FX larda qutqaruv va avariya-tiklash ishlarini olib borish bo'yicha tadbirlar ishlab chiqiladi. FX larning ta'sir darajasini kamaytirish, uning zararli faktorlaridan himoyalanishga qaratilgan tadbirlar ko'p bosqichli sistemadan iborat bo'lib, quyidagilarni o'z ichiga oladi: doimiy o'tkaziladigan tadbirlar uzoq muddatli oldindan bilish ma'lumotlari asosida amalga oshiriladi. Ularga qurilish montaj ishlarini kurilish normalari va qoidalari asosida amalga oshirish; xavf tug'risida aholiga xabar berishning ishonchli sistemasini ishlab chiqish; himoya inshootlarini kurish va aholini ShXV bilan ta'minlash;

radiatsion, bakteriologik va kiyoviy kuzatishni, razvedkani hamda laboratoriya tekshirishlarini tashkil etish; FX lar vaqtidagi harakat koidalari bo'yicha aholini umumiy hamda majburiy o'qitish; sanitar-gigiyenik va proflaktik tadbirlar o'tkazish; AES ni ko'rmaslik, kimyoviy va sellyulozakogoz va shu kabi potensial xavfli ob'yektlarni xavfsiz zonalarda kurish; FX lar okibatlarini bartaraf etish rejalarini ishlab chiqish, uni material va moliyaviy ta'minlashni tavshkil etish va boshka shu kabi tadbirlar kiradi.

FX lar sodir bo'lish ehtimoli aniqlangan vaqtdagi himoya tadbirlari jumlasiga oldindan bilish ma'lumotlarini aniklashtirish buyicha kuzatish va razvedka sistemasini ishlab chikish; axoliga FX lar tugrisida xabar berish sistemasini tayyor xolatga keltirish;

iktisodni va ijtimoiy xayotni davom etishining maxsus koidalarini joriy etish, FX ni e'lon qilish; yuqori xavflilikdagi ob'yektlarni (AES, zaharli va portlashga xavfli ishlabyu chikarish va b.) neytrallashtirish, ularda ishni to'xtashish va ularni qo'shimcha mustaxkamlash yoki demontaj kilish; avariya-kutkaruv xizmatini tayyor holatga keltirish va aholini kisman evakuatsiya kilish kabi tadbirlar kiradi.

Ushbu tadbirlar majmuasidan ma'lumki ayrim tadbirlar uzok muddatli oldindan bilish ma'lumotlari asosida bajarilib, ularni amalga oshirish uchun kup yillar talab etiladi.

Ayrim tadbirlar esa qisqa vaqt ichida tez amalga oshiriladi. Bunday tadbirlar qisqqa muddatli oldindan bilish ma'lumotlari asosida amalga oshiriladi.

Xozirgi vaktida fan-texnika taraqqiyoti, mutaxassislar FX lar sodir bo'lish vaqtini va joyini oldindan yuqori aniqlikda aytib berish imkoniyatiga ega emas.

Favkulodda vaziyatlar vaktida xayot faoliyati xavfsizligini ta'minlashga karatilgan tadbirlarni rejalashtirish, FX lar vaqtida hayot faoliyati xavfsizligini ta'minlashning yetakchi funksiyasi va markaziy zvenosi hisoblanadi. Rejalashtirishda hujjat-reja tuziladi va u quyidagi qismlardan iborat bo'ladi: aniq ko'rsatkichlar (ish turi, tadbirlar); ushbu ishlarni bajarish vaqti; ishlarni bajarish uchun zarur resurslar (turi, soni, miqdori,manbai); ishni bajaruvchi ma'sul shaxs (har bir punkt bo'yicha); ishni bajarilishini nazorat qilish usuli.

Rejaning matn qismi ikki bo'limdan iborat bo'lib, birinchi bo'limida FX lar vaqtidagi shart-sharoitlarni baholash bo'yicha xulosalar, ikkinchi bo'limda esa FX lar xavfidan aholini ximoyalash tadbirlari kursatiladi. Ushbu tadbirlarga asosan quyidagilar kiritilishi mumkin, ya'ni: FX to'g'risida xabar berish tartibi; kuzatish va razvedkani tashkil etish;

qutqaruv va boshqa muhim ishlarni bajarish uchun kuch va vositalarni tayyorlash; FX Lar ta'sirini bartaraf etish yoki susaytirish tadbirlari; odamlar va

material boyliklarni himoyalash tadbirlarini tezkor bajarish usullari; tabiiy ta'minlash, dozimetrik va kimyoviy nazorat; ishlab chiqarishni avariyasiz to'xtatish tartibi; odamlarni himoyalashni tashkil etish, ShXV bilan ta'minlash; evakuatsiya tadbirlarini tashkil etish; ularni boshkarish; har xil sharoitlarda qutqaruv ishlarini tashkil etish tartibi; yuqori tashkilotlarga va FX lar bo'yicha tuzilgan komissiyalarga axborot hamda ma'lumotlar berish tartibi.

Rejaga turli xil zarur lug'ataviy va tushuntiruvchi xarakterdagi materiallar ham ilova qilinadi. Reja real, qisqa mazmunli, lekin tuliq ifoda etilgan, iqtisodiy jihatdan makbul bilishi hamda ob'yektning barcha imkoniyatlarini ifoda etishi zarur.

Rejaning realligi tabiiy va texnogen kurinishidagi FX lar vaqtida haqiqiy ishlab chiqarish sharoitida hayot faoliyat xavfsizligini ta'minlash bo'yicha sistemali turli xil mashg'ulotlar va amaliy mashqlar o'tkazish yo'li bilan tekshiriladi.

Favkulodda xolatlar vaktida korxonalarda turgun ishlashini ta'minlash.

Xalq xo'jalik ob'yektlarining (XXO) turg'unligi ularni FX larning xavfli va zararli faktorlari ta'siriga chidamliligi, ya'ni FX lar sharoitida rejalashtirilgan hajmda va nomenklaturada maxsulot ishlab chikarish, ishchi va xizmatchilar xayot faoliyati xavfsizligini to'g'ri ta'minlash hamda ishlab chikarishga zarar yetgan holatlarda o'z ish qobiliyatini tiklashga moslashishi orqali baholanadi.

FX lar vaktida ob'yektning turg'un ishlashiga tashkiliy, injener-texnik va boshqa tadbirlarni kompleks ravishda amalga oshirish natijasida erishiladi.

Ushbu tadbirlar birinchi navbatda ishchilar va hizmatchilarni himoyalashga karatilgan bo'lishi kerak. Chunki inson resursi hamda XXO larini FX larning xavfli va zararli faktorlaridan himoyalanmasdan turib, ularni turg'un ishlashini ta'minlab bo'lmaydi. Bundan tashqari, ob'yektdagi ishchilar va ob'yekt yaqinida yashovchi aholini hayot faoliyati xavfsizligini ta'minlashda FX larning buzuvchi faktorlari ta'sirida yuzaga keluvchi

ikkilamchi xavfli faktorlar sodir bo'lish xavfining oldini olishga qaratilgan tadbirlar ham muxim rol o'ynaydi. Ikkilamchi xavfli faktorlarga, ichki va tashqi sabablar natijasida vujudga kelishi mumkin.

XXO larining FX lar vaqtida turg'un ishlashini ta'minlashga qaratilgan tadbirlar kompleksi ichidan asosiy ikkita tadbirga, ya'ni, aynan FX larda ishchi va xizmatchilarni hayot faoliyati xavfsizligini ta'minlash muammolariga hamda ikkilamchi xavfli faktorlar hosil bo'lishini bartaraf etishga qaratilgan tadbirlarga to'xtalamiz.

Ishchi-hizmatchilarni himoyalash tadbirlariga- texnologik jarayonlarda portlashga va yonginga xavfli xamda zararli va radiaktiv moddalar ishlatiladigan ish sharoitlarida ish rejimini tashkil etish; zaharlanish uchogini bartaraf etishga qaratilgan ishchilarni aniq bajarish yullari buyicha o'qitish; ob'yektdagi ishchi va hizmatchilar xamda ob'yekt yaqinidagi

aholiga, ob'yektda xosil bulgan xavf tugrisida habar berishning lokal sistemasini tashkillashtirish va uni doimiy tayyor holatda saqlash kabi ishlar kiradi.

FX larning xavfli va zararli faktorlari ta'sirida yuz beradigan yog'inlar, portlashlar, zaharli, radiaktiv moddalarni muxitga tarkalishi ikkilamchi faktorlar jumlasiga kiradi.

Ma'lumki, normal ish sharoitida ob'yektning xavfsiz va avariyasiz ishlashini ta'minlashga qaratilgan kator tadbirlar amalga oshiriladi. Lekin bu faktorlar FX lar vaqtida yetarli darajada bo'lmaydi. Shu sababli, FX larning ikkilamchi faktorlaridan himoyalashga qaratilgan kushimcha tadbirlar ishlab chikish talab etiladi. Bunday tadbirlarga- saqlanadigan

portlashga, yonginga xavfli va zaxarli moddalar zaxirasini minimum darajagacha kamaytirish;

saklash omborlarini xavfsiz joda, mustaxkam kilib, shamol yunalishini, yongin oraliklari va yo'laklarini yonginga karshi suv ta'minotini xisobga olgan xolda kurish; ularni yongin uchiruvchi vositalar, zaxira elektr manbalari, aloqa vositalari, avtomat signalizatsiya kabi vositalar bilan ta'minlash ishlari kiradi.

Favqulodda holatlar oqibatlarini bartaraf etish.

Avariylar, xalokatlar va tabiiy ofatlar oqibatlarini bartaraf etish, mamlakatning avariya-qutqaruv xizmatini doimiy tayyor holatini ta'minlash hamda ishlab chiqarish korxonalarida avariylar va halokatlarni oldini olishga qaratilgan chora-tadbirlarni bajarilishi ustidan nazorat qilish maqsadida O'zbekiston Respublikasida faqulotda holatlar komiteti tuzilgan. FX lar oqibatlarini bartaraf etishga qaratilgan barcha vazifalar bosqichma-bosqich, aniq ketma-ketlik asosida maksimal qisqa muddatlar ichida bajarilishi lozim. Birinchi bosqichda axolini tezkor ximoyalash masalalari, FX lar xavfli faktorlarini tarqalishini cheklash va uning ta'sir darajasini kamaytirish chora-tadbirlari hamda qutqaruv ishlarini amalga oshirish kabi vazifalar amalga oshiriladi.

Aholini tezkor himoyalashning asosiy tadbirlariga xavf to'g'risidagi rejimga rioya qilishni ta'minlash; xavfli zonalardan evakuatsiya qilish; tabiiy proflaktik tadbirlarni amalga oshirish, jarohatlanganlarga tibbiy va boshqa turdagi yordamlar kursatish kabi ishlar kiradi.

FX lar ta'sir doirasini cheklash va uning oqibatlarini susaytirishga karatilgan tadbirlar asosan: avariylarni lokalizatsiyalash, ishlab chiqarish texnologik jarayonlarini to'xtatish yoki o'zgartirish, yong'inni oldini olish yoki uni o'chirish kabi vazifalarni o'z ichiga oladi.

Qutkarish va boshqa turdagi kechiktirib bo'lmaydigan tadbirlar jumlasiga boshqarish organlarini, kuch va vositalarni tayyor holatga keltirish, zararlanish o'chog'ini razvedka qilish va mavjud holatni baholash kabi vazifalar kiradi.

Ikkinchi bosqich vazifalariga FX lar oqibatlarini bartaraf etish bo'yicha qutkaruv hamda boshqa kechiktirib bo'lmaydigan ishlarni amalga oshirish kiradi. Bu ishlar uzluksiz ravishda qutqaruvchilar va bartaraf etuvchilar smenalarini almashtirgan holda xavfsizlik texnikasi va ehtiyot choralariga tulinib amal kilib bajarilishi shart.

Qutqaruv ishlari jaraohatlanganlarni qidirib topish, ularni yonadigan binolar, harobalar, transport vositalari ichidan olib chikish, odamlarni xavfli zonalardan evakuatsiya qilish, jaroxatlanganlarga birinchi yordam ko'rsatish va shu kabi boshqa yordamlarni amalga oshirish ishlarini o'z ichiga oladi.

Kechiktirib bo'lmaydigan ishlar jumlasiga esa yonginni lakalizatsiyalash va uchirish, konstruksiyalarni mustahkamlash, qutqaruv ishlarini amalga oshirish maqsadida kommunal-energetik setlarni, aloqa va yo'llarni tiklash, odamlarga sanitar ishlov berish, dezaktivatsiyalash va degazatsiyalash ishlarini amalga oshirish kabi vazifalar kiradi.

Kutkaruv va kechiktirib bulmaydigan ishlar jumlasiga axolini barcha turdagi vositalar bilan ta'minlash, jumladan, ularni xavfsiz joylarga joylashtirish, oziq-ovkat va suv bilan ta'minlash, tibbiy yordam ko'rsatish hamda material va moliyaviy yordamlar berishni amalga oshirish kabi vazifalar ham kiradi.

Ikkinchi bosqich vazifalariga avariylar, halokatlar va tabiiy ofatlar yuz bergan rayonlardagi aholi faoliyatini ta'minlash masalalari kiradi. Bu maqsadda turar joylarni tiklash yoki vaqtinchalik turar joylar barpo etish, energiya va suv ta'minotini, aloqa liniyalarini, kommunal hizmat ob'yektlarini tiklash, zararlanish o'chog'iga sanitar ishlov berish, aholiga oziq-ovqat mahsulotlari hamda birlamchi ehtiyoj buyumlari bilan yordam

ko'rsatish ishlari amalga oshiriladi. Ushbu bosqich nihoyasida evakuatsiya qilingan aholi o'z joylariga qaytariladi va xalq xo'jalik ob'yektlarining ishlashi tiklanadi. Ayrim FX lar ning sodir bo'lishi oldindan aniklanishi mumkin. Bunday holatlarda amalga oshirilishi lozim bo'lgan barcha ishlar oldindan ishlab chiqilgan reja asosida amalga oshiriladi Birinchi guruhdagi tadbirlar aholini himoyalash maqsadida amalga oshiriladi. Bu tadbirlarga- axoliga xavf tugrisida ma'lumot berish va xabar berish; himoya vositalarining tayyor holga keltirish; boshqarish sistemalari va vositalarining tayyorligini tekshirib kurish; tibbiy proflaktika, sanitar

va epidemiyaga karshi tadbirlarni o'tkazish; evakuatsiyaga tayyorlanish va talab etilgan sharoitlarga xavf taxdid soladigan rayonlarda aholini evakuatsiya kilish kabi vazifalar kiradi.

Ikkinchi guruhdagi tadbirlarga FX larning xavfli va zararli faktorlarini bartaraf etishga karatilgan vazifalar kiradi. Bu tadbirlarga-xalq xo'jaligi ob'yektlari ishini to'xtatish yoki ish rejimini o'zgartirish; energiya, suv, gaz sistemasi ish

rejimini o'zgartirish yoki vaqtincha to'xtatish; mavjud injenerlik inshootlarini mustahkamlash yoki qo'shimcha qurish; yong'inga karshi tadbirlar o'tkazish; xavfli rayonlardan material boyliklar va chorva mollarini olib chiqish; oziq-ovqat, oziqa hom ashyosi va suv manbalarini himoyalash kabi ishlar kiradi.

FX lar sodir bulganligi tugrisida xabar olingach, birinchi navbatda belgilangan ma'lumotlarni to'g'riligi tekshirilib, qo'shimcha axborot va ma'lumotlar olish bo'yicha tadbirlar amalga oshiriladi. Chunki, turli xil FX larning xar xil sharoitlardagi oqibatlari turlicha bulishi mumkin. Shu sababli, dastlab, FX lar ta'sirida yuzaga kelishi mumkin bulgan ikkilamchi, uchlamchi va h.k. xavfli faktorlar aniqlanib, keyingagina kompleks tadbirlar amalga oshiriladi.

Xulosa

Bitiruv malakaviy ishini bajarish davomida quydagi natijalarga erishildi:

1. HTML dashturlash tili va uning imkoniyatlari tahlil qilindi.
2. Dinamik saytlarni yaratish texnologiyasining afzalliklari va kamchiliklari ko'rib chiqildi.
3. Web texnologiyalari imkoniyatlari oshirish vositalaridan foydalanish masalalari ko'rib chiqildi.
4. Ma'lumotlar bazasini boshqarish tizimlari taxlil qilindi
5. Instrumental vosita sifatida dinamik saytlar yaratish imkonini beruvchi PHP web-dasturlash tili va My SQL ma'lumotlar bazasini boshqarish tizimi tanlandi.
6. «Klent-server» texnologiyasi asosida ishlovchi "IT-prise" dasturi yaratilgan.

"IT-prise" dasturi Internet mag'azin bo'lib asosan kopyuter qurilmalarini internet tarmog'i orqali sotish imkonini beradi. Dastur Web-saxifa ko'rinishida yaratilgan. Budan tashqari dastur orqali kompyuter , noutbook va shuningdek tarmoq qurilmalari xaqida ma'lumotlar olish mumkin.

Foydalangan adabiyotlar

1. I.A.Karimov “Jahon moliyaviy-iqtisodiy inqirozi, O’zbekiston sharoitida uni bartaraf etishning yo’llari va choralari” nomli asari. Toshkent. 2009. 56 b.
2. Шафран Э. Создание Web-страниц: самоучитель-СПб:Питер,2001.-320с.
3. Игорь Шапошников HTML4 «БХВ-Петербург» 2002.
4. Максим Кузнецов, Игорь Симдянов, Сергей Голшев. PHP 5. Практика создания Web-саутов (+ CD-ROM). Серия: Pro. Профессиональное программирование.,Издательство: БХВ-Петербург, 2005 г., 948 стр.
5. Дмитриу Котеров, Алексеу Костарев., PHP 5. Наиболее полное руководство Серия: В подлиннике Издательство: БХВ-Петербург, 2005 г., 1120 стр.
6. Анатолиу Мотев Уроки MySQL (+CD-ROM) Серия: Самоучитель Издательство: БХВ-Петербург, 2006 г., 208 стр.
7. Викрам Васвани. Полну справочник по MySQL. MySQL: The Complete Reference. Издательство: Вильямс, 2006 г., 528 стр.
8. Люк Веллинг, Лора Томсон MySQL. Учебное пособие MySQL Tutorial Перевод:Мягкая обложка Издательство: Вильямс, 2005 г., 304 стр.
9. Поль Дюбуа MySQL MySQL Серия: Landmark Другие издания: Тверду переплет Аналоги: Тверду переплет Издательство: Вильямс, 2004 г., 1056 стр.
- 10.Уилтон П. JAVASCRIPT. Основ. Символ-плюс. 2002. 1056 с.
- 11.Кингли-Хью Э., Кингли-Хью К. JAVASCRIPT 1.5: Учебнй курс. Питер. 1-е издание. 2002.
- 12.Бранденбау. JAVASCRIPT. Сборник рецептов для профессионалов. СПб. 2001.
- 13.Рейнбоу В. Компьютерная графика СПб-Питер, 2003. 768 с.
- 14.www.intuit.ru
- 15.www.citforum.ru

Ilova

Dastur kodi:

```
<?
if (!isset($_GET[id]) or $_GET[id]=="" or !is_numeric($_GET[id])) {
header("location:/");
exit();
}
$cid = (int) $_GET[id];
$query = mysql_query("select * from mahsulotlar where catID = $cid");
$text .= "<h3> Mahsulotlar</h3><table><tr>";
while ($row = mysql_fetch_array($query)){
$text .= "<td valign='top'>
        <div class='item'>
            <img src='images/bordtop.gif' alt='' width='274'
height='2' />
            <div>
                <img src='\"$row[image]\"' alt='' width='102'
height='88' class='items' />
                <h1>\".strtoupper($row[name]).\"</h1>
                <span><strong>Parametrlari:</strong>
$row[descript]</span>
                <p><strong>$row[summa] so'm</strong></p>
                <a href='\"?cat=deteils&id=$row[id]\"'><img
src='images/button1.gif' alt='' width='129' height='20' /></a>
                <a href='\"?cat=order&id=$row[id]\"'><img
src='images/button2.gif' alt='' width='127' height='20' /></a>
            </div>
            <img src='images/bordbot.gif' alt='' width='274'
height='2' />
        </div>
    }
}
```

```

</td>";
$i++;
if ($i % 2==0) $text = "<tr>".$text."</tr>";
}

$text .="</tr></table>";
echo $text;

?>
<?
if (isset($_POST['submit'])) {
require_once('./pages/contact/contactengine.php');
}
?>
<h3>Biz bilan aloqa</h3>
<br>
<div id="title-content" style="text-align:left;">Kontakt ma'lumotlari:</div>

<br>
<p>
Tel: +998 (93) 564-29-09</p>
<p>
e-mail: karimbek_85@mail.ru
</p>

<br>

<div id="title-content" style="text-align:left">Xabar jo'natish:</div>

<? if ($success) {

```

```
echo "<br><p style='color:maroon'> Xabaringiz jo'natildi!</p><br>";  
}
```

```
else {
```

```
?>
```

```
<div id="contact-area">
```

```
<form action="" method="post">
```

```
<label for="Name">Ism:</label>
```

```
<input type="text" id="Name" name="Name">
```

```
<label for="City">Shahar:</label>
```

```
<input type="text" id="City" name="City">
```

```
<label for="Email">Email:</label>
```

```
<input type="text" id="Email" name="Email">
```

```
<label for="Message">Xabar:</label><br>
```

```
<textarea id="Message" cols="20" rows="20"
```

```
name="Message"> </textarea>
```

```
<input type="submit" class="submit-button"
```

```
value="Jo'natish" name="submit">
```

```
</form>
```

```
<br>
```

```
</div>
```

```
<? } ?>
```

```
<div id="title-content">Joylashgan manzili:</div>
```

```
<br>
```

```

<div style="text-align:center;">

    <a href="/images/map.png" target="_blank">
        
    </a>
</div>

</div>
<?
if (!isset($_GET[id]) or $_GET[id]=="" or !is_numeric($_GET[id])) {
header("location:");
exit();
}
$id = (int) $_GET[id];
$query = mysql_query("select*from mahsulotlar where id='$id'");
$row = mysql_fetch_array($query);

$text .= "<h3>$row[name]</h3>
<table id='prd-content' width='100%'>
<tr><td width='40%' valign='top'><img width='175' height='125'
src='$row[image]'></td><td valign='top' width='60%'>

<p><strong>Garantiya: $row[garanty]</strong></p>
<p><strong>$row[summa] so'm</strong></p>
</td></tr>

<tr><td><a href='?cat=order&id=$id'><img style='border:0'
src='/images/order.gif' title='Buyurtma berish'></td></tr>
</table>".htmlspecialchars_decode($row[parametr])."";

```

```

echo $text;

?>

<?

session_start();

ob_start();

require_once("./inc/categories.php");

require_once("./inc/functions.php");

header("Content-type: text/html;charset=utf-8");

connect_db();

```

```

$tovar = $_SESSION['tovar'];

if ($tovar!="") $tovar = explode(",",$tovar);

$tovar_count = count($tovar);

?>

```

```

<!DOCTYPE    HTML    PUBLIC    "-//W3C//DTD    HTML    4.01
Transitional//EN"

```

```

"http://www.w3.org/TR/html4/loose.dtd">

```

```

<html>

```

```

<head>

```

```

<title>Home</title>

```

```

<meta http-equiv="Content-Type" content="text/html; charset=Windows-
1251" />

```

```

<link rel="stylesheet" type="text/css" href="style.css" />

```

```

<script src="js/jquery.js" type="text/javascript"></script>

```

```

<script src="js/functions.js" type="text/javascript"></script>

```

```

</head>

```

```

<body>

```

```

<div id="main">

```

```

<div id="header">

```



```

<div class="block1">

<?
if (isset($_SESSION['username'])) {
    echo "<div style=\"padding-top:10px;\"><span>Hush kelibsiz,
($_SESSION[username])<br>                                $_SESSION[firstname]
$_SESSION[lastname]</span>
    <br><a href='?cat=profile'>Shaxsiy kabinet</a>
    <br><a href='?cat=logout'>Chiqish</a></div>";
}
else {
?>

<form name="login" action="?cat=login" method="POST">
<table                                style="padding-top:5px;"><tr><td><label
for="username">Login: </label> </td>
<td><input type="text" name="username" id="username" value=""
size="17"></td>
</tr>
<tr><td><label for="password">Parol: </label></td>
<td><input type="password" name="password" id="password"
value="" size="17">
</td> </tr>
<tr><td></td> <td><input type="submit" value="Kirish"> <a
href="?cat=forgot" title="Parol unutildimi?">Parol unutildimi? </a> </td> </tr>
<input type="hidden" name="redirect_url" value="<?if
(isset($_GET[cat])) echo curPageURL(); ?>">
</table>
</form>

<? } ?>

```

```

</div>
<div class="block2">
    <ul>
        <li><a href="?cat=search">Qidiruv</a></li>
        <li><a href="?cat=contact">Aloqa</a></li>
        <li><a href="?cat=register">Ro'yhatdan o'tish</a></li>
        <li><a href="?cat=help">Yordam</a></li>
    </ul>
</div>
<div class="block3">
    <a href="?cat=order"></a>
    <span>Savatingizda:<br/>
    <strong><?echo $tovar_count?> ta mahsulot</strong></span>
</div>

<div class="banner">
<div class="title">

</div>

</div>
<ul class="menu">

    <!--<li><a href="#"></a></li>
    <li><a href="#"></a></li>

```

```

        <li><a href="#"></a></li>
        <li><a href="#"></a></li>
        <li><a href="#"></a></li>
        <li></li>
    </ul>
    -->
    <link rel="stylesheet" type="text/css" href="./style/top-menu.css"/>
    <ul class="solidblockmenu">
    <li><a href="/">Bosh sahifa</a></li>
    <li><a href="?cat=new-products" class="current">Yangi
mahsulotlar</a></li>
    <li><a href="?cat=contact">Aloqa</a></li>
    <li><a href="?cat=order-help">Qanday buyurtma beriladi?</a></li>
    </ul>
    <br style="clear: left" />

</div>
<div id="content">
    <? left_block();?>
    <div class="right">
    <? select_category();?>
    </div>
</div>

```

```

<div id="footer">
    Copyright &copy; 2010 TUIT. All Rights Reserved
</div>
</div>
</div>
</body>
</html>
<?
if (!isset($_GET[id]) or $_GET[id]=="" or !is_numeric($_GET[id])) {
    //    header("location:");
    //exit();
}
$id = (int) $_GET[id];
$query = mysql_query("select * from mahsulotlar order by Edate desc limit
6");
$text .= "<h3> Yangi mahsulotlar</h3> <table><tr>";
while ($row = mysql_fetch_array($query)){
    $text .= "<td valign='top'>
        <div class=\"item\">
            <img src=\"images/bordtop.gif\" alt=\"\" width=\"274\"
height=\"2\" />
            <div>
                <img src=\"\$row[image]\" alt=\"\" width=\"102\"
height=\"88\" class=\"items\" />
                <h1>".strtoupper($row[name])."</h1>
                <span><strong>Parametrlari:</strong>
$row[descript]</span>
                <p><strong>$row[summa] so'm</strong></p>
                <a      href=\"?cat=deteils&id=$row[id]\"><img
src=\"images/button1.gif\" alt=\"\" width=\"129\" height=\"20\" /></a>

```

```

                                <a          href="\?cat=order&id=$row[id]\ "></a>
                                </div>
                                
                                </div>
                                </td>";
                                $i++;
                                if ($i % 2==0) $text = "<tr>".$text."</tr>";
                                }

                                $text .="</tr></table>";
                                echo $text;

                                ?>

                                <div id="posts">

                                <?

                                $query = mysql_query("select*from posts") or die(mysql_error());
                                while ($row = mysql_fetch_array($query)){
                                //$postid = (int) $row[id];
                                //$post_author = mysql_fetch_array(mysql_query("select name from admin
                                where id='{ $row[post_author]} '"));
                                //$comment_number      =      mysql_fetch_array(mysql_query("SELECT
                                COUNT(*) as num FROM comments where comment_post_ID='{ $postid} '"));

```

```

        //      $comment_number      =      $comment_number[num];      if
($comment_number==0) $comment = "Izohlar yo'q"; else $comment = "Izohlar
(".$comment_number).") ";

        echo                                                    "<h3><a
href='?cat=showpost&postid=$row[id]'">".$row[post_title]."</a></h3>
        <span> $row[post_author]: $row[post_date]</span>
        <div                                                    id='post-
content'">".htmlspecialchars_decode($row[description])."</div>

        ";

    }

    ?>

    <!--
    <div class='post-info'>
        <span    class=\"post-cat\"><a    rel=\"category    tag\"    title=\"Dasturlar
bo'limidagi            barcha            maqolalarni            ko'rish\"
href=\"?category=information\">Ma'lumotlar</a></span>
        <span            class=\"post-comments\"><a            title=\" $row[title]\"
href=\"?category=showpost&postid=$row[id]#comments\">$comment</a></span>
    >

    </div>

    -->

    </div>

    <?

    // Finally, destroy the session.

```

```

if (isset($_GET[id]) and $_GET[id]!="") {
    $id = (int)$_GET[id];

    if (isset($_SESSION['tovar'])) {
        $array = explode(",",$_SESSION['tovar']);
        foreach ($array as $value){
            if ($id == $value ) $check = true; break;    }
            if      (!$check)      $_SESSION['tovar']      =
$_SESSION['tovar'].",".$id;
        }
    else $_SESSION['tovar'] = $id;
    }

    $products = $_SESSION['tovar'];

    if (isset($products) and $products != "")
        $products = explode(",", $products);

    $text .="<form name='form' action='?cat=order-make' method='post'>
    <table
class='sale_basket_basket'><th>No</th><th>Nomi</th><th>Firmasi</th><th>Nar
xi</th><th>O'chirish</th>";

    $i = 1;

    if ($products!=""){

        foreach ($products as $value){
            $row    =    mysql_fetch_array(mysql_query("select    *,SUM(summa)from
mahsulotlar where id=$value group by name"));
            $sumumiy_sum += $row['summa'];

```

```

$text .= "<tr><td>${i}</td><td>${row[name]}</td><td>${row[firma]}</td><td>${r
ow[summa]}</td>
    <td>
    <input type='checkbox' value='${row[id]}' id='paradigm' name='paradigm[]'>
    </td>
    <tr><tr>";
    $i++;
}
}
$text
                                .= "<tr><td>Umumiy
narx</td><td></td><td></td><td>". $umumiy_sum."</td><td><input
type='checkbox' id='paradigm_all'></td></tr>
    </table><br><a
                                href='javascript:document.form.submit()'><img
src='./images/order.gif' title='Buyurtmani rasmiylashtirish></a>
    </form>";

?>
<script type='text/javascript'>
    $(document).ready(function()
    {
        $('#paradigm_all').click(function()
        {
            var checked_status = this.checked;
            $('input[@id=paradigm]').each(function()
            {
                this.checked = checked_status;
            });
        });
    });
});

```



```

</script>
<div id="prd-title">
<h3>Sizning savatingiz:</h3>
<br>
</div>
<? echo $text; ?>
<?
if ($_POST['submit']){
$search = secureSuperGlobalPOST('search');
if ($search!="") {
    $products = search('mahsulotlar','name',$search);
    }
    else {
        $text .= "<div class='message'>Hech qanday ma'lumot
topilmadi!</div>";
    }
}
?>
<div id="prd-title">
<h3>Qidiruv:</h3>
</div>
<form name='searchform' action="" method="POST">
<table class="register">
<tr>
<td>
<select name="product">
<option value="0">--Katagoriya--</option>
<?
$query = mysql_query("select*from categories");

```

```

while ($row = mysql_fetch_array($query)) {
    if ($row[id]==$_POST[product]) $selected = "selected='selected'"; else
    $selected = "";
    echo "<option $selected value=\"\$row[id]\">$row[cat_name]</option>";
    }
?>

```

```

</select> </td>
<td>
<input type="text" name="search" value="<? echo $_POST['search'];?>">
</td>
<td><input      type="submit"      value="Qidiruv"      name="submit"
style="width:70px;"> </td>
</tr>
</table>
</form>
<br>
<?
if (isset($products)){
    $result = false;
    while ($productsrow = mysql_fetch_array($products)){
        $result = true;
        $text .= "<div class=\"item\">
                    <img src=\"images/bordtop.gif\" alt=\"\" width=\"274\"
height=\"2\" />
                    <div>

```

```


<h1>".strtoupper($productsrow[name])."</h1>
<span><strong>Parametrlari:</strong>
$productsrow[descript]</span>
<p><strong>$productsrow[sum]
so'm</strong></p>
<a
href="\"?cat=deteils&id=$productsrow[id]"></a>
<a
href="\"?cat=order&id=$productsrow[id]"></a>
</div>

</div>";
$i++;
if ($i % 2==0) $text = "<tr>".$text."</tr>";
}
if ($result == false) {
    $text .="<div class='message'>Hech qanday ma'lumot
topilmadi!</div>";
}

}

echo $text;
?>

```

```

<div id="posts">

<?
$id= (int) $_GET['postid'];

if($id == 0 or !isset($id)) { header("location:?cat=news");exit();}

$query = mysql_query("select*from posts where id='$id'") or
die(mysql_error());

while ($row = mysql_fetch_array($query)){
// $postid = (int) $row[id];
// $post_author = mysql_fetch_array(mysql_query("select name from admin
where id='{ $row[post_author]}"));
// $comment_number = mysql_fetch_array(mysql_query("SELECT
COUNT(*) as num FROM comments where comment_post_ID='{ $postid}"));
// $comment_number = $comment_number[num]; if
($comment_number==0) $comment = "Izohlar yo'q"; else $comment = "Izohlar
(".$comment_number.") ";

echo "
"
href='?cat=showpost&postid=$row[id]'>".$row[post_title]."</a></h3>
"
<span> $row[post_author]: $row[post_date]</span>
"
<div id='post-
content'>".htmlspecialchars_decode($row[post_content])."</div>

";

}

?>

```

```

<!--
<div class='post-info'>
    <span    class=\"post-cat\"><a    rel=\"category    tag\"    title=\"Dasturlar
bo'limidagi            barcha            maqolalarni            ko'rish\"
href=\"?category=information\">Ma'lumotlar</a></span>
    <span            class=\"post-comments\"><a            title=\"<!--
    $row[title]<!--
    href=\"?category=showpost&postid=$row[id]#comments\">$comment</a></span
>
</div>
-->
</div>

```