

**O‘ZBEKISTON RESPUBLIKASI A‘LOQA,
AXBOROTLASHTIRISH VA TELEKOMMUNIKATSIYA
TEXNOLOGIYAALARI DAVLAT QO‘MITASI**

**TOSHKENT AXBOROT TEXNOLOGIYALAR UNIVERSITETI
NUKUS FILIALI**

Axborot texnologiyalari kafedrası

**Kompyuter injiniring fakultetining “Informatika va axborot
texnologiyalari” yo‘nalishining 4-kurs talabasi Matnazarov Usmanjonning**

BITIRUV MALAKAVIY ISHI

**Mavzu: “Abonentlarga xizmat ko‘rsatish markazi” axborot tizimini
yaratish. (Internet xizmati)**

Nukus tumani PAT

Ilmiy rahbar:_____ boshlig‘i Musakulova K.

Ilmiy maslahatchi:_____ ass. Madreymova Z.

Kafedra boshlig‘i:_____ t.f.n.Arzimbetov T.

NUKUS – 2014

MUNDARIJA

KIRISH.....	3
I. BOB. AXBOROTLAR TIZIMLARINI ISHLAB CHIQISHNING BOSQICHMA – BOSQICHLIGI.....	5
1.1. Axborotlar tizimlarini ishlab chiqishning bosqichma – bosqichligi....	5
1.2. Axborotlar tizimlarini ishlab chiqishning rivojlanishi.....	8
II. BOB. WEB – SAYT YARATISH UCHUN KERAK BO‘LGAN DASTURIY TILLAR HAQIDA.....	11
2.1. HTML tili.....	11
2.1.1. HTML hujjatdagi matnni formatlash.....	11
2.1.2. HTML hujjatga grafik ob’ektni joylashtirish.....	19
2.2. JavaScript dasturlash tili.....	25
2.2.1. HTML-sahifada JavaScript ni joylashtirish.....	25
2.2.2. Funksiya va usullar. JavaScript buyruqlari.....	28
2.3. PHP dasturlash tili.....	33
2.3.1. PHP Web dasturlash tili haqida ma’lumot.....	33
2.3.2. PHP – dasturida komentariyalar (izohlar).....	38
2.3.3. PHP4 da shartli jarayonlarni ifodalash.....	39
2.3.4. Funktsiyalarga dinamik murojat qilish.....	43
2.4. MYSQL ma’lumotlar ombori haqida.....	47
III. BOB. UCELL KOMPANIYASI DILLERINING ISHLASH MEXANIZMIDA Joomla DASTURINING QO‘LLANILISHI.....	53
3.1. Joomla dasturida web – saytni yaratilish jarayoni.....	53
3.2. Abonentlar uchun ajratilgan sahifa.....	57
3.3. Korporatsiyalar uchun.....	60
3.4. Ucell haqida ma’lumot.....	63
XULOSA.....	65
FOYDALANILGAN ADABIYOTLAR.....	66
ILOVALAR.....	67

KIRISH.

Mavzuning dolzarbligi. Abonentlarga xizmat markazlarini(internet xizmati) ishlab chiqish mabaynida abonentlarga yaxshi va juda samaralari xizmat ko'rsatish hozirgi kunda juda dolzarb masalalardan biri bo'lib, bu yo'nalishda ko'pgina ishlar amalga oshirilmoqda. Bizning hayotimizda abonentlarga xizmat ko'rsatish sifati jihatdan qaraladigan aniqlikga ko'proq e'tibor qaratiladiganini hisobga olsak, abonentlarga xizmat ko'rsatish biz uchun qanchalik ahamiyatga ega ekanligini tushinib yetamiz. "Abonentlarga xizmat ko'rsatish markazi" axborot tizimini yaratish. (Internet xizmati) esa hozirgi kunning muhim muammolari sirasiga kiradi.

Oxirgi 10 yillik ichida abonentlarga xizmat ko'rsatish bo'yicha internet xizmatlariga ehtiyoj juda kuchayib bordi. Zamonaviy axborot texnologiyalar juda kuchayib bormoqda, abonentlar o'z ma'lumotlar va o'zining hisob raqamlari va hokozolarini bilishi uchun kompaniya binolariga bormasdan turib internet orqali bolib olish uchun harakat qilishyapdi. Abonentlar o'zlariga kerak ma'lumotlarni olish uchun ko'proq soddalik va o'zlari uchun qulaylikni hisobga olishadi. Abonentlarga xizmat ko'rsatish bo'yicha internet orqali web-sahifalardan foydalanish orqali abonentlarga xizmat ko'rsatish rivojlangan mamlakatlarda keng qo'llanilmoqda. Bunga ayniqsa ulardagi bank tizimlarining ochiq tizimlarga o'tganligi juda kata qulaylik keltirmoqda. Lekin bizda bu tizimlar juda mukammal darajada emas, bu tizimlarni mukammal darajada takomillashtirish muhim vazifalardan biridir.

Prezidentimiz I.Karimov XXI asrning ma'naviy-ma'rifat asri, ilm-fan madaniyat va axborotlar asri deb ta'riflar ekan, bu bilan hozirgi davrning muhim belgisi va xususiyatini ham belgilab berganini payqab olish qiyin emas. Buyuk maksadlarimizga, ezgu niyatlarimizga erishishimiz, jamiyatimizning yangilanishi, xayotimiz tarakkiyoti va istikboli, amalga oshirilayotgan isloxotlarimiz va rejalarimizning samarali takdiri, avvalambor, davr talablariga javob beradigan

yukori malakali, ongli, tafakkurga ega bulgan mutaxassis kadrlar tayyorlash muammosi bilan bogliq

Hozirgi kunda bank tizimlari, ulardagi pul o'tkazish va uni qabul qilish va shunga o'xshagan ko'pgina masalalar muammo bo'lib qolmoqda.

Ishning maqsadi. Ushbu bitiruv malakaviy ishda "Abonentlarga xizmat ko'rsatish markazi" axborot tizimining internet xizmati bo'yicha Ucell kompaniyasi dillerlariga xizmat ko'rsatish web-saytini yaratishni ko'rib chiqamiz, va bu sayt internetga joylanishi ham mumkin.

Ushbu bitiruv malakaviy ishda quyidagilar bilan tanishib o'tamiz:

Birinchi BOB da axborot tizimlarini ishlab chiqishning bosqichma – bosqichligini ko'rib chiqamiz. Bunda abonentlarga xizmat ko'rsatish markazi bo'yicha kerak bo'ladiga modellashtirish va loixalashtirish bo'yicha hozirgi zamon texnologiyalarini o'rganib chiqamiz.

Ikkinchi BOB da bo'lsa web-saytni yaratish uchun zarur bo'ladigan eng asosiy tillarni ko'rib o'tamiz. Unda asosan php, javascript, phpmysql, html kabi tillar va ular orqali yaratiladigan web-sahifalardan iborat.

Uchinchi BOB da esa Ucell kompaniyasi dillerlari uchun yaratiladigan web-sayt uchun joomla dasturining qo'llanilishi keltirilgan bo'lib unda uning o'rnatilishi, webserverni hosil bo'lishi, unda abonentlar uchun zarur bo'ladigan ma'lumotlar haqida qisqacha berib o'tilgan.

Xulosa qismida bitiruv malakaviy ishida olingan asosiy natijalar hulosalangan.

Adabiyotlar qismida bitiruv malakaviy ishini bajarishda foydalanilgan 14 asosiy adabiyotlar ro'yxati keltirilgan. Ilovada yaratilgan Ucell.uz saytining HTML, PHP, Javascript, PHPMySQL lardagi yozilgan matnlari ilova qilingan.

I. AXBOROTLAR TIZIMLARNI ISHLAB CHIQISHNING BOSQICHMA – BOSQICHLIGI.

1.1. Axborotlar tizimlarni ishlab chiqishning bosqichma – bosqichligi

Axborotlar tizimlarini loyixalashtirishning boshlang'ich materiali bo'lib boshqarish ob'ekti tahlilining natijalari xizmat qiladi, ular eng avvalo ma'lumotlarni qayta ishlash tizimi yordamida hal etuvchi boshqarish tizimi funksiyalari va vazifalarini aniqlashga xizmat qiladi.

Funksional tuzilmadan kelib chiqib, funksional tizimlarni amalga oshirish uchun qo'llaniladigan matematik model va algoritmlar aniqlanadi yoki ishlab chiqiladi. Bundan keyin zarur axborot tarkibi, uni tashkil etish usullari aniqlanadi, zarur dasturiy va texnik ta'minot tanlovi yoki ishlab chiqilishi amalga oshiriladi, xodimlarning tashkiliy va funksional majburiyatlarni aniqlashtiriladi. Huquqiy ta'minot AT ishlashi sharoitida boshqarishning tashkiliy tuzilmasi elementlari o'zaro aloqasi qoidalarini mustahkamlaydi.

Axborot texnologiyalarining tadrijiy yo'li uyushgan biznesning yangi modellari rivojlanishi bilan yaqindan shunday bog'liqlik, bu jarayonlar ko'pincha yagona bir butunlik sifatida qabul qilinadi. Kompaniyalarning AT samaradorligini oshirishga intilishi ancha takomillashgan apparat va dasturiy vositalarning paydo bo'lishini rag'batlantiradi, ular o'z navbatida, foydalanuvchilarni ATni bundan keyingi yangilashga undaydi.

ATni ishlab chiqishdan oldin, korxona mutahasislari bilan birgalikda, ushbu proektni ishlab chiqib tadbiq qilish maqsadga muvofiqmi yoki yo'qmi degan savolga javob berishlari kerak. Aks holda ATni korxona faoliyatida qo'llash natijasida kutilgan samaraga erisholmaslik mumkin. Bundan tashqari, mavjud muammoni avtomatlashtirish usulida echishda bozorda taklif qilinayotgan AT ko'llash mumkin yoki mumkin emasligini xam ko'rib chiqish maqsadga muvofiqdir. Bu xolda ATni ishlab chiqish va amaliyotga tatbiq qilish orasidagi vakt bir necha marotaba qisqarishi mumkin. Tizimli tadqiqot asosida qurilayotgan muammoga tegishli bo'lgan barcha masalalar majmuasi tizimlar nazariyasiga

asoslangan holda qabul qilinib, ularning orasidagi mavjud aloqalarni ko'rsatish kerakdir. Ushbu AT ni ishlab chiqish bosqichi muammoni har tomonlama ko'rib chiqadi va tizimni loyihalash uchun zamin tayyorlaydi.

Loyihalash bosqichida, muammoga tegishli barcha kirish (birlamchi), oraliq (agar mavjud bo'lsa) va chiqish hujjatlarining o'zaro bog'liqligi va ularning formalari kompyuterga kiritishda osonlashtirish holiga keltiriladi. Hujjatlardagi ortiqcha ma'lumotlar chiqarib tashlanadi.

Tasdiqlangan loyihani zamonaviy dasturlash tillari asosida kompyuterning ichki tiliga o'tkaziladi. Bu jarayon judayam murakkab bo'lib, yuqori saviyali dasturlovchilarni jalb qilishni talab etadi.

Hozirgi paytda iqtisodiy tizim infratuzilmasining muhim qismini savdo kompaniyalari tashkil etadi, ularning bosh vazifasi iste'molchilar buyurtma beradigan tovarlarni savdo kompaniya bo'linmalari, savdo tashkilotlari kompaniya bo'linmalari va vakolatxonalari orqali iste'molchiga o'z vaqtida yetkazib berish va samarali sotishdir.

Savdo kompaniyalarida logistika muhim ro'l o'ynaydi. Agar unga tegishli bo'lgan muammolar doirasini jamlikda ko'rib chiqilsa, unda turli tuman oqimlarni boshqarish masalalari ular uchun umumiy bo'ladi. Bozor munosabatlari sharoitida moddiy texnika bazasini takomillashtirish me'yorlari o'zining ilgarigi ahamiyatini yo'qotdi. Har bir xo'jalik sub'ekti aniq bir vaziyatni mustaqil baxolaydi va qarorlar qabul qiladi. Dunyo tajribalarining guvohlik berishicha, kimki logistika sohasida uquvli bo'lsa va uning uslublarini yaxshi egallasa, raqobatli kurashda peshqadamlikni qo'lga kiritadi. Bu o'rinda logistikaning namunaviy biznes-jarayonlari, tashkiliy-shtat tuzilmasi, xizmat ko'rsatish ishi texnologiyasi ko'rib chiqiladi.

Logistika xizmati kechadigan biznes jarayonlar bayoniga bevosita o'tishdan avval, savdo kompaniyalarida qo'llaniluvchi biznes-tizimlar va biznes-jarayonlar ta'riflarini beramiz. Biznes-tizim tugal maqsadi – mahsulotni sotish bo'lgan ko'plab biznes-jarayonlar bilan bog'lik holda namoyon bo'ladi. Mahsulot deganda tovarlar, xizmatlar va axborot tushuniladi.

Biznes-jarayonning tugal maqsadi - mahsulotni sotish bilan bog'liq bo'lgan ichki o'zaro to'be funksional harakatning gorizontal pog'onasini o'zida namoyon qiladi. Odatda biznes-jarayonlarning quyidagi kategoriyalari farqlanadi:

a) biznes – tizimga daromad keltiruvchi mahsulotni sotish jarayonlari(gorizontal);

b) mahsulotni sotish gorizontal biznes – jarayonlarini realizatsiya qilishda daromad olish bilan samarali rejalashtirish va boshqarishni ta'minlovchi rejalashtirish va boshqarish jarayonlari(vertikal);

v) harakatlarni bevosita nuqtasiga mahsulotni yetkazib berish va joylashtirishni ta'minlovchi zahirali jarayonlar(vertikal);

d) mavjud texnologiyalarni o'zlashtirish uchun zarur bo'lgan yordamchi jarayonlar-o'zgartirishlar jarayonlari(vertikal).

Logistika xizmati biznes-jarayonlari doirasida quyidagi asosiy vazifalar amalga oshiriladi:

tovarlar turkumini olish va junatish muddatlarini kelishish,
transport agentlik bo'yicha axborotni tahlil qilish,
tovarlar yetkazib berishning qulay yo'llarini belgilash,
kompaniya omborida tovarlarni qabul qilish va saqlash,
bo'lim va mayda do'konchalar buyurtmalarini qayd etish,
tovarni kompaniya omboriga qaytishini nazorat qilish.

Savdo kompaniyasining korporativ axborot tizimi(KAT) foydalanuvchilarning turli tipiga qarab lokal va uzoqlashtirilgan ish joylarini birlashtiruvchi kengaygan tarmoqni o'zida namoyon etadi. KAT asosiy biznes-markazlariga aloqador bo'lgan tashkilotlarning axborot tizimlari bilan a'loqani ta'minlaydi.KATning asosiy elementlari quyidagilar:

markaziy ofisni lokal hisoblash tarmog'i(LHT);
vakolatxonalar LHTsi;
kompaniya savdo bo'linmalarining terminallari;
savdo agentliklaridagi portativ agentliklar;
omborxona zahiralarini rejalashtirish va qulaylashtirish.

Logistika xizmatining avtomatlashtirilgan ish joylari (AIJ) tarkibiga quyidagilar kiradi:

- AIJ logistika bo'yicha direktor o'rinbosari;
- AIJ mijozlar xizmati boshlig'i;
- AIJ mijozlar xizmati boshlig'ining yordamchisi.

Logistika xizmatining AIJsi quyidagi jarayonlarni avtomatlashtirish uchun mo'ljallangan:

- transport oqimlarini rejalashtirish;
 - tovarlarni ta'minotchilar, savdo korxonalari, kompaniya vakolatxonalari va savdo bo'limlaridan olib kelishni rejalashtirish;
 - omborxona operatsiyalarini hisob-kitob qilish;
 - transport korxonalari bilan shartnoma qilish;
 - tashuvchilar faoliyatini tahlil qilish va tashish harajatlarini belgilash;
- omborxonalar inventarizatsiyasi aktlarini hisobga olish.

AT ma'lumotlar bazasining asosiy tuzilmaviy elementlari quyidagilar: mahsulot, shartnoma, ta'minotchi, iste'molchi, omborxona, buyurtma, mahsulotni qaytarish, harajat qog'ozi, miqdor, akt, tavsifnoma, tashuvchi.

Aytib o'tilganidek, zamonaviy sharoitda logistika ko'pgina kompaniyalar uchun muhim ro'l o'ynaydi, shu bois biznes jarayonlarini tashkil etish va qayta tashkil etishsiz, shuningdek logistika xizmatining axborot tizimini tashkil qilmay turib, kompaniyada samarali ish olib borish mumkin emas.

1.2. Axborotlar tizimlarini ishlab chiqishning rivojlanishi.

ATlarini bir necha avlodlari farqlanadi: 1-avlod (1960-70)markaziy EHM negizida "bir korxona - bir qayta ishlash markazi" tamoili bo'yicha qurilgan funksional vazifalarni bajarishning standart muhiti sifatida esa IBM-MVS firmasining operatsiyaviy tizimi xizmat qilgan.

2-avlod (1970-80 y.)ATning markazlashtirilmagan davrga o'tishdagi ilk qadamlari, bu jarayonda foydalanuvchilar DEC VAXtipidagi kichik kompyuterlardan foydalangan holda ofis va kompaniya bo'limlariga axborot texnologiyalarini joylashtira boshlagan. Markazlashmagan ma'lumotlar negizi va

amaliy dasturlar to'plami asosida axborot poydevori bo'lgan ma'lumotlar qayta ishlash tizimlarini tashkil etishning ikki va uch darajali modeli yangi avlod ATlarida o'ta yangilik bo'ldi.

3-avlod(1980-90 y)bu yillar shaxsiy kompyuterga ommaviy o'tish davri bo'ldi. Uyushgan biznes mantiqiy tarqoq ish joylarini yagona ATga birlashishni talab etdi. Biroq tez orada bir rangli (darajali) tarmoqlarda ierarxialilik belgilari - avvalo ajratilgan fayl-serverlar, nashr, serverlar va telekommunikatsiya serverlari ko'rina boshladi. Uchinchi avlod ATlarni rivojlantirishda sof (bir rangli) taqsimlovchi qayta ishlash g'oyasi "klient-server"ierarxiya modeliga o'rnini bo'shatib berdi.

4-avlod dunyoga kela boshladi, biroq xozirdanok tushunarliki, yuqori darajada AT zaxiralarini markazlashgan xolda qayta ishlash va yagona boshqarish quyi darajada taqsimlovchi qayta ishlash bilan uyg'unlashadigan zamonaviy xususiyatlari avvalgi avlodlar tizimlarida sinovdan o'tgan qarorlar sintezi bilan belgilanadi. AT 4-avlodi uch model: katta,o'rta va kichik yo'lidan boradi deb karalmoqda. Modellar ma'lumotlari mantiqiga ko'ra, AT tuzilmasida bir yoki bir necha axborot tugunlari jamlanmasi(ATJ) mavjud bo'lishi lozim, ulardan xar biri oxirgi foydalanuvchilar ishini samarali qo'llab quvvatlash uchun mo'ljallangan apparat va dasturiy vositalarni birlashtiradi. Oxirgi foydalanuvchilar lokal tarmoqlar muhitida ishlaydi, ularning individual ilovalari va ma'lumotlari mijozlar stantsiyalari darajasida eng ko'p lokallashadi. Tugunlar jamlanmasi zahiralarining harakatga kelishi faqat nisbatan kamdan-kam hollarda, masalan, uyushgan ma'lumotlar negiziga murojat qilishda yoki fayllardan zahiraviy olishda ro'y beradi. Shunday qilib, yangi element-tugunli jamlanma bilan to'ldirilgan taqsimlovchi qayta ishlashning erkin ifodalangan modelini ko'rib turibmiz. ATning bunday tashkil qilinishini ba'zan AT 3-avlodi markazlashmagan tarmoqlariga qarama-qarshi o'laroq markazlashgan tarmoq deb ataladi. Markazlashgan tarmoqli AT modeli foydalanish harajatlarini iqtisod qilishdan tashqari yana ikki afzallikka ega:

- “klient-server” texnologiyasini samarali amalga oshirish imkoniyati;
- tugunli jamlanmaga jo bo‘lgan apparat va dasturiy vositalar uyg‘unligidagi keng spektr xisobiga foydalanuvchilar talabiga moslashuvchanlik.

Markaziy hisoblash tizimi sifatida meyn-freym, UNIX server ishchi stantsiyalar klasterlari yoki super kompyuterlardan foydalanish mumkin. Biroq yagona server atrofida tizimning jamlanishi har doim ham eng yaxshi qaror bo‘lavermaydi. Birinchidan, serverga ulangan mijozlar soni keskin chegaralangan. Mijozlar sonining oshib borishi tizim reaksiyasining sekinlashuviga olib keladi. Ikkinchidan, zamonaviy ATlardan bugalterlik hisobi dasturi turidagi an’anaviy biznes ilovalardan boshlab, to umuman korxonani boshqarish vazifalarigacha bo‘lgan ko‘plab xilma-xil rejalardagi vazifalarni bajarish talab etiladi.

II. BOB. WEB – SAYT YARATISH UCHUN KERAK BO‘LGAN DASTURIY TILLAR HAQIDA.

Bu bobda web sayt yaratish uchun kerak bo‘ladigan dasturiy vasitalar bilan tanishib chiqish va o‘rganish bo‘yicha ma’lumotlar keltirilgan.

2.1. HTML tili.

2.1.1. HTML hujjatdagi matnni formatlash.

Shrift, uning o‘lchami va rangini o‘zgartiruvchi teglar

Matnni formatlash kompyuter ekranida hujjatni ko‘rinishini o‘zgartirishning eng qulay usullari hisoblanadi. Matnni formatlashning barcha teglarini ikki sinfga ajratish mumkin: jismoniy va mantiqiy formatlash teglari. Ko‘pincha sirtidan har xil bo‘lgan teglar qo‘llanganda bir xil natija berishi mumkin. juft tegi shrift parametrlarini boshqarish imkoniyatini beradi. U albatta quyidagi uch atributlardan birini o‘z tarkibiga olishi kerak: COLOR, FACE, SIZE. Masalan, . Umumiy holda juft tegining yozish sintaksisi quyidagi ko‘rinishga ega:

FACE atributi shriftlarning to‘la komplektini beradi. Bu atributning qiymati va teglarining orasiga joylashtirilgan matnni ko‘rish uchun rejalashtirilgan shrift nomi. Maslan, . Ushbu matn Arial shriftidan foydalanib yozilgan .

Matnni ekranga chiqarish uchun zarur bo‘lgan shriftning o‘lchamini SIZE atributi ko‘rsatadi. Bu atributning qiymati (parametri) absolyut yoki nisbiy miqdorda berilishi mumkin. Parametr sifatida absolyut miqdor shrift balandligini punktlarda ko‘rsatuvchi, 1 dan 6 gacha bo‘lgan oraliqdagi butun son nazarda tutiladi. Nisbiy miqdor esa, plus yoki minus ishorali butun son bo‘lib, browser foydalanayotgan odatdagi shriftga nisbatan o‘lcham oshirilishi yoki kamaytirilishi zarur bo‘lgan punktlar soni. Masalan, . Ushbu matn, odatdagi hujjat matniga shriftning kattaligiga nisbatan, ikki punktga kichik shrift bilan yozilgan .

Shrift rangini o‘zgartirish uchun COLOR atributi ishlatiladi. Bu atribut qiymatining o‘rniga tanlanayotgan rangga mos raqamli kodni yoki bu rangni

belgilovchi simbolik nomini qo'yishimiz mumkin. Shuni ta'kidlab o'tish zarurki, juft tegidan foydalanganda uning atributlaridan ayrimlarini tashlab yuborish mumkin.

Shriftning yozilishini o'zgartiruvchi teglar

Odatda, odam boshqalar bilan suhbatlashganda u yoki bu so'zga urg'u beradi. Shunday qilib, biz nimaga alohida e'tibor berish kerakligini suhbatdoshimizga ko'rsatamiz. Yozma nutqda esa bunday imkoniyatdan holi. Ammo, biz so'zlarni ajratib yozishning turli usullaridan foydalanishimiz mumkin.

Matnning zarur qismini qalin (Bold) shrift bilan ajratib ko'rsatish mumkin. Buning uchun va teglaridan foydalanish kerak. Masalan: Ushbu matn qalin shrift bilan yozilgan . Ayrim so'zlarni tagiga chizish (Underline) mumkin. Bunday hollarda <U> va </U> teglari qo'llaniladi. Masalan: <U> Ushbu matn so'zlarining tagiga chizilgan </U>. Ko'p hollarda ajratib ko'rsatish uchun kursiv (Italie) shrifti ishlatiladi. Buning uchun <i> va </i> teglari xizmat qiladi. Masalan: <i> Ushbu matn kursiv (qiya bosma harf) bilan yozilgan </i>. Bu teglarni bir-birining ichiga yozish mumkin. Masalan, matnni qalin kursiv shrift bilan ajratib ko'rsatish quyidagicha amalga oshiriladi: <i> Ushbu matn qalin kursiv bilan yozilgan </i>. Ammo, HTML tilining yaratuvchilari bu teglardan ehtiyotkorlik bilan foydalanishni maslahat beradi. Chunki, hamma brouzerlardagi interpretatorlar ularni to'g'ri tushuna olmaydi. Shuning uchun ham ularga o'xshash bo'lgan va "mantiqiy ajratib ko'rsatish farmoyishlari" deb ataluvchi boshqa farmoyishlardan foydalanishni maslahat beradi. Ular bir xil funksiya bajaradi, ammo yozish sintaksisi biroz boshqacharoqdir.

 matn kursiv shrift bilan yozilgan .

 matn qalin shrift bilan yozilgan .

Matnni ajratib ko'rsatishning yuqorida ko'rib o'tilgan teglardan tashqari, yana birqancha matn unsirlari mavjud bo'lib, ular turli usullar bilan foydalanuvchining diqqatini u yoki bu so'zlarga qaratish imkoniyatini beradi. Ularning orasidan quyidagilarni ta'kidlab o'tish zarur. hujjatning biror qismidagi matnni ustiga chizib chiqish uchun <STRIKE> juft tegi ishlatiladi. Bu tegning yozilish sintaksisi quyidagicha:

<STRIKE> Ushbu matnning ustiga chizilgan </STRIKE>. Matnni qatorning quyi qismiga kichik harflar bilan yozish imkoniyatini <SUB> juft tegi beradi. <SUB> tegining yozilish sintaksisi quyidagi ko‘rinishga ega:

Matn1 _{Matn2}

Ushbu qator brouzerning oynasida quyidagi ko‘rinishga ega bo‘ladi:
Matn1Matn2

Matnni qatorning yuqori qismiga kichik harflar bilan yozish uchun <SUP> tegidan foydalaniladi. Bu tegning yozilish sintaksisi quyidagi ko‘rinishga ega:

Matn1 ^{Matn2}

Bu qator kodlarini qayta ishlash natijasida quyidagiga ega bo‘lamiz:

Matn1 Matn2

Matnning o‘zgarmas (fiksirlangan) shrift bilan yozilgan qismini ekranga chiqarish <TT> tegi yordamida amalga oshiriladi. Matnning fiksirlangan shriftlar bilan yozilgan qismi teletayp usulda yozilgan matn (yoki qisqacha talaetayp matni) deb atadi. Bu shrift asosan dastur matnini (listigini), ekran formalari va foydalanuvchining farmoyishlarini. Bu tegning yozilish sintaksisi quyidagi ko‘rinishga ega:

<TT> Ushbu matn “teletayp” shrifti bilan yozilgan </TT>. Matnning bir qismini katta o‘lchamli harflar bilan ekranga chiqarish imkoniyatini <BIG> tegi beradi. Bu tegdan foydalanish natijasida normal shriftlarning o‘lchami bir punktga ortadi. <BIG> tegini yozilish sintaksisi quyidagicha:

<BIG> Ushbu matn odatdagi (normal) shriftdan bir punktga katta shrift bilan yozilgan </BIG>.

Matnning bir qismini kichik harflar bilan, ya’ni normal shriftdan bir punktga kichik bo‘lgan shriftlar bilan ekranga chiqarishni <SMALL> tegi amalga oshiradi. Bu tegni yozilish sintaksisi quyidagi ko‘rinishga ega:

<SMALL> Ushbu matn odatdagi (normal) shriftdan bir punktga kichik harflar bilan yozilgan </SMALL>.

Shuni yodda tutish kerakki, matnni ajratib yozishda juda ehtiyot bo‘lish kerak. Chunki, qalin, qiya (kursiv) yoki tagiga chizilgan shriftlar bilan yozilgan matn bilan ishlash ancha qiyin, uni o‘qish ko‘zni tez charchatadi va diqqatni tez-tez

bo‘lib turadi. Shuning uchun agar so‘zlarni ajratib yozishni iloji boricha kamroq ishlatish, ya’ni faqat juda zarur bo‘lgan hollardagina ishlatish tavsiya etiladi. Shriftning yozilishini o‘zgartiruvchi teglarining ro‘yxati 1 – jadvalda keltirilgan. Axborot turini boshqarish teglarining ro‘yxati 1 – jadvalda keltirilgan.

1 – jadval. Tasvirlash shaklini boshqarish teglari.

Teg	Qiymat
<I>...</I>	Kursiv (Italic)
...	Qalin (Bold)
<TT>...</TT>	Teletayp
<U>...</U>	Ostiga chizilgan
<S>...</S>	Ustidan chizilgan
<BIG>...</BIG>	Katta o‘lchamli shrift
<SMALL>...</SMALL>	Kichik o‘lchamli shrift
_{...}	«Satr osti» belgilar
^{...}	«Satr usti»belgilar

2-jadval. Axborot turini boshqarish teglari.

Teg	Qiymat
<ABBR>...</ABBR>	Abbreviatura
<ACRONYM>...</ACRONYM>	Akronim
...	Tipografik qalinlashtirish, ajratish
<CITE>...</CITE>	Sitatalar
...	Kuchaqtirish
<CODE>...</CODE>	Kodni tasvirlash(masalan,"dastur kodi")
<SAMP>...</SAMP>	Belgilar ketma-ketligi (namuna)
<KBD>...</KBD>	Klaviaturadan kiritish

<VAR>...</VAR>	O'zgaruvchi
<DFN>...</DFN>	Ta'rif
<Q>...</Q>	Qavsga olingan matn

Masalan:

qizil matn joriysidan bir o'lchamga kichikroq, shrift nomi "Times New Roman"

Matnning negiz o'lchami <BASEFONT size="o'lcham"> tegida ko'rsatiladi.

Simvollarni jismoniy formatlashning boshqa teglari 2-jadvalda keltirilgan, mantiqiysi –2-jadvalda.

<BLINK> – matnni o'chib-yondiradi. U faqat Netscape Navigator brauzerida ishlaydi. Juda ehtiyotkorlik bilan foydalanish kerak. Format:

<BLINK> o'chib-yonuvchi matn </BLINK>

<MARQUEE> – matnni harakatlantiradi (yuqoridan pastga yoki u yoqdan – bu yoqqa). Format:

<MARQUEE behavior=alternate| scroll| slide direction=down| left| right| up > matn </MARQUEE>

behavior atributi skrolling (harakatlanish) ko'rinishini aniqlaydi: alternate – chapdan o'ngga tebranish; scroll – matnni direction orqali ko'rsatilgan yo'nalishda harakatlantirish, ekran chetiga yetgan matn qarama-qarshi tomondan chiqib keladi; slide –scroll ga o'xshash, lekin matn faqat bir marta harakatlanadi va to'xtaydi. direction atributi skrolling yo'nalishini aniqlaydi: down – pastga; left – o'ngdan chapga (indalmaganda); right – chapdan o'ngga; up – yuqoriga.

Matnni tekislash teglari.

Matnlarni tekislashning ayrim usullarini 1-bo'limda ko'rib chiqqan edik. Bu erda yuqorida ko'rib o'tilgan abzats tegining atributlarisiz matnning ma'lum bir qismini tekislash farmoyishlarini ko'rib chiqamiz. HTML tili brouzer oynasidagi hujjatning bir necha ob'ektini ma'lum bir o'rinlarga joylashtirish uchun mo'ljallangan teglarga ega. Web-sahifasidagi ob'ektlarning ma'lum bir qismi

ekranning chap tomoniga nisbatan tekislash uchun <LEFT> va </LEFT> teglaridan foydalanish mumkin. Bu teglarning yozilishi quyidagi ko‘rinishga ega:

<LEFT> </LEFT> Bu HTML kodlarida yozilgan matnning bir qismi <LEFT> va </LEFT> teglarining orasidagi hamma ob’ektlar ya’ni matn, jadval va grafiklar ekranning chap tomoniga nisbatan tekislanadi.

HTML-hujjatdagi matn, jadval va grafik ob’ektlarning ma’lum bir qismini ekranning markaziga nisbatan tekislash uchun <CENTER> va </CENTER> teglaridan foydalanish mumkin. Bu teglar quyidagi ko‘rinishda yoziladi:

<CENTER>

Bu HTML kodlari yordamida tekislangan matnning bir qismi

</CENTER>

Ushbu teglarning <CENTER> va </CENTER> orasiga hamma ob’ektlar ekranning markaziga nisbatan tekislanadi.

Agar HTML-hujjatdagi ob’ektlarning qandaydir bir qismini ekranning o‘ng tomoniga nisbatan tekislash zarurati tug‘ilsa, u holda <RIGHT> va </RIGHT> teglaridan foydalanish mumkin. Bu teglarning yozilishi ham yuqoridan tekislash teglariga o‘xshash.

Endi ko‘rib chiqilgan teglardan foydalanib matnni tekislashga doir bir misol ko‘ramiz.

<LEFT> HTML-hujjatdagi boshlovchi va tamomlovchi LEFT juft teglari orasiga yozilgan matnni ushbu qismi ekranning chap tomoniga nisbatan tekislanadi.(Brouzer oynasida bu matn chap tomonga nisbatan tekislanadi.)

</LEFT>

<RIGHT>

matnning ushbu qismi esa RIGHT juft teglarining orasiga joylashgan bo‘lib, u ekranni o‘ng tomoniga nisbatan tekislanadi. (Bu matn esa brouzer oynasida o‘ng tomonga nisbatan tekislangan.) Bunday tekislangan matnni o‘qish ancha qiyin, ammo u o‘ziga diqqatni o‘ziga tez jalb qiladi. Shuning uchun sarlavhalarni va epigraflarni yozishda ular ko‘p qo‘llaniladi.

</RIGHT>

<CENTER>

Bu matn CENTER juft teglari orasiga joylashgan, shuning uchun u ekranni markaziga nisbatan tekislanadi. (Bu teglar qisqa sarlavhalarni formatlash uchun ishlatiladi, ammo markazga nisbatan tekislangan katta matnni o‘qish juda qiyin.

</CENTER>

Ko‘p hollarda tartiblangan axborot ro‘yxatlar ko‘rinishida taqdim etish juda qulay bo‘ladi. Bunga biror ishni bajarish uchun ko‘rsatish, ob‘ektlar yoki predmetlarni sanab o‘tish misol bo‘ladi.

Ro‘yxatlar

Ro‘yxatlar yaratish uchun quyidagi teglar mavjud:

 – belgilangan (markirovanno‘y);

 – tartiblangan (numerovanno‘y);

 – ro‘yxat elementlarini sanash;

<LH> – ro‘yxat nomi;

<DL> – ta’riflar ro‘yxati;

Web-sahifadagi ro‘yxatlar turli ko‘rinishga ega bo‘lib, ular quyidagilardan iborat: belgilangan ro‘yxatlar, nomerlangan ro‘yxatlar, ta’riflar ro‘yxatlari va ichma-ich joylashgan ro‘yxatlar.

Belgilangan (markerlangan) ro‘yxat.

Belgilangan (markerlangan) ro‘yxat yaratish tegi ushbu formatga ega:

<UL type = disc|circle|square title=”suzib chiquvchi yordam”> ...

type atributida belgilash markeri turi ko‘rsatiladi: disc – bo‘yalgan doira (boshqa qiymat ko‘rsatilmasa); circle – aylana; square – kvadrat.

Ro‘yxatning har bir elementi tegidan boshlanadi, uni yopmasa ham bo‘ladi. Format:

<LI type = "disc|circle|square">....

Belgilangan (markerlangan) ro‘yxat (Unordered List) quyidagi ko‘rinishga ega:

«YaNGI AVLOD» O‘QUV METODIK MARKAZI QUYIDAGI KURSLARGA QABUL E’LON QILADI:

- Kompyuter savodxonligi (Windows, Word, Excel)
- Kompyuter grafikasi (PhotoShop, Coreldrew)

- Chet tili(ingliz, nemis, frantsuz)
- Bank ishi
- Ish yurutuvchi
- Tasviriy san'at

Bunga o'xshash belgilangan ro'yxatlarni tashkil etish uchun HTML tilida juft tegidan foydalanish mumkin. Ro'yxatning har bir elementi oldin ma'lum bir belgi (nullet) turishi kerak. Bu yordamida amalga oshiriladi. Buning uchun ro'yxatning har bir elementini quyidagicha yozish kerak:

 Ro'yxatning bir elementi .

Sarlavha yoki ro'yxatni belgilash uchun tilning LH elementidan foydalaniladi.

Sodda misol keltiraylik:

Ro'yxat nomi

- 1 - qiymat
- 2 - qiymat
-
- N - qiymat.

HTML tilida bunday belgilangan ro'yxatni buyruqi yordamida quyidagicha ko'rsatish mumkin:

```
<UL type="disc">
<LH> Ro'yxat nomi </LH>
<LI> 1 - qiymat </LI>
<LI> 2 - qiymat </LI>
....
<LI> n - qiymat </LI>
</UL>
```

2.1.2. HTML hujjatga grafik ob'ektni joylashtirish

Tarmoq texnologiyasining grafik formatlari

Ko'p hollarda grafik illyustratsiyalar Web-hujjatlarning ajralmas qismi bo'lib xizmat qiladi. Hozirgi paytda Web-sahifada grafik elementlarni tasvirlash uchun asosan ikki xil formatdan foydalaniladi. Ular GIF va JPEG formatlaridir.

Hozirgi paytda Internetdagi grafik ob'ektlarning 90 foizi GIF va JPEG formatlarida saqlanadi. Web-saxifalarini kompyuter ekranida ko'rsatish (aks ettirish) uchun mo'ljallangan grafik brauzerlarning hammasi bu formatdagi fayllar boshqa fayllardan ajrata oladi va ularni ekranga aks ettiradi. Bu formatlar uchun tasvir sifatini yo'qotish hisobiga, u saqlayotgan fayl o'lchamini kamaytirish algoritmlari asos qilib olingan.

GIF formati 1978 yilda ikki isroillik olim –Yakob Zif (Jacob Ziv) va Abraham Lempel (Abraham Lempel) ma'lumotlarni yo'qotmasdan axborotlar hajmini qisqartirishning printsiplari yangi algoritmini ishlab chiqdilar, va unga LZ78 deb nom berdilar. Bu algoritmning tuzilish printsiplari haqidagi axborot hamma uchun ochiq edi. Buning natijasida amerikalik dasturlovchi Terri Uelch (Terry Welch) uni mukammallashtirdi va yangi algoritm nomiga o'z familiyasini bosh harfini qo'shib, uni LZW nomi bilan patentladi. U ham Yakob Zif hamda Abraham Lempel kabi o'z algoritmini hamma hoxlovchilar uchun foydalanishga taqdim etdi.

Bunday "hoxlovchilardan" biri CompuServe Inc. kompaniyasining xodimi Bob Berri (Bob Berry) bo'lib, u 1987 yilda LZW asosida printsiplari yangi grafik formatini yaratdi. Bu grafik format GIF (Graphic Interchange Format) ma'lumotlarni qisqartirish uchun LZW algoritmidan foydalandi va bu jarayon ayrim adabiyotlarda kompressiya deb ataldi. Bu paytlarda Terri Uelch tomonidan tashkil qilingan Unisys kompaniyasi (LZW algoritmining avtorlik huquqi unga tegishli bo'lgani uchun) faqat kompyuter uchun apparat ta'minotini ishlab chiquvchilar GIF standartidan foydalanganliklari uchun ma'lum bir miqdorda haq berar edi. Masalan, modamlarni ishlab chiqaruvchilardan ushbu standartdan foydalanganlari uchun. Dasturiy ta'minotni ishlab chiqaruvchilardan "kommission to'lovlar" olinmas edi.

Ammo 1994 yilning qishda Unisys kompaniyasi moliyaviy muammoga duch keldi va LZW standartidan foydalanish tijorat asosida amalga oshirilishini e'lon qildi, natijada foydalanuvchilar haq to'lashga majbur bo'ldi. Bu avtomatik ravishda GIF formatini dunyoda birinchi "pullik" grafik formatiga aylantirdi va Internet foydalanuvchilari orasida norozilik to'lovini paydo bo'lishiga sabab bo'ldi.

Chunki, bu davrda Web-sahifalarining juda katta qismiga GIF elemenlari tadbqiqilingan edi. Shunga qaramasdan GIF grafik formati hozirgi kunda ham Internet tarmog'ida juda keng foydalaniladigan formatlardan biri. Bundan tashqari bu formatdagi grafik ob'ektini Web-sahifaga joylashtirgan odam hech qanday to'lov to'lashga majbur emas, chunki yuqorida keltirilgan moliyaviy talab, birinchi navbatda GIF bilan ishlaydigan dasturiy ta'minotni ishlab chiqaruvchilarga tegishlidir. Bu standartning kelajak taqdirini belgilovchi holat hozirda hal qilingan.

LZW algoritmining imkoniyatlari tufayli GIF standarti yakuniy grafik fayl hajmini boshlang'ich tasvirga nisbatan sezilarli darajada qisqartirish imkoni hosil bo'ladi. Bu juda o'xshash ranglarni birga qo'shib yuborish usuli bilan amlaga oshiriladi. Masalan, agar rasmning tarkibida bir necha o'xshash (misol uchun havo rang, och havo rang, to'q havo rang) iborat parchalar mavjud bo'lsa, ular bir xil, ya'ni havo rang rangi bilan kodlashtiriladi. Tasvir haqidagi axborot GIF standartining faylda satr(qator) bo'ylab yoziladi, ya'ni bir piksel balandligidagi satrlarni tasniflovchi massivdan iborat. Ushbu format fiksirlangan, boshqacha aytganda indekslashtirilgan palitraga tayanadi. Ammo bu palitradagi ranglarning soni 256 dan oshmaydi.

Shuni ta'kidlab o'tish kerakki, HTML hujjatlarda GIF standartidan faqat chizilgan grafik ob'ektlarni, ya'ni diagrammalar, chegaralovchi chiziqlar, tugmachalar va sahifani bezovchi boshqa elementlarni kompyuter ekarnida aks ettirish uchun ishlatiladi. Boshqa hollarda, masalan, orfoghshrafiyalar yoki juda katta miqdordagi xilma-xil ranglardan tashkil topgan tasvirni Web-sahifaga joylashtirish uchun JPEG formatidan foydalaniladi.

JPEG formati. Bu formatni taklif qilgan tadqiqotchilar guruhi (Joint Photographic Experts Group) sharafiga JPEG nomi qo'yilgan va "Djey-peg" deb o'qiladi. Ushbu grafik standartni tasvir sifatini yo'qotmasdan uning hajmini qisqartirish uchun mo'ljallangan JPEG algoritmi asosida yaratildi. U bir-biriga aynan o'xshash bo'lmagan elementlarni xuddi LZW algoritmi kabi kodlashtiradi, ammo peksillar orasidagi interval boshqacha kodlashtiriladi. JPEG format bilan berilgan fayldagi tasvirni qisqartirishning soddalashtirilgan mexanizmi quyidagi ko'rinishga ega. Qisqartirishning birinchi pog'onasi tasvirni LAB rangli obrazga

aylantirishdan iborat bo'lib, u rasmni 3 mustaqil kanalga yoyib beradi. Ulardan biri (Ligntness) ranglar intensivligini saqlash uchun, qolgan ikkisi (A va B) esa bevosita rang haqidagi axborotni saqlash uchun ajratiladi. Bunda ranglar haqidagi ma'lumot uzluksiz spektr printsipli asosida tashqi shkala ko'rinishida saqlanadi. qisqartirishning ikkinchi pog'onasi quyidagicha amalga oshiriladi: birinchi pog'onada hosil qilingan rang modelidan rang haqidagi axborotning taxminan to'rtadan uch qismi olib tashlanadi, so'ng obraz 8x8 nuqtalar o'lchamidagi qismlarga bo'linadi va sonli ma'lumotlar massiviga aylantiriladi. Har bir blokning sarlavhasi qismning asosiy rangini tasvirlaydi axborotning qolgan qismini esa kam farq qiluvchi ranglar tasniflaydi. Qisqartirishning uchinchi pog'onasida esa ikkinchi darajali ranglarni tasniflovchi axborotning malum bir qismini ma'lumotlar massividan olib tashlaydi. Bunda olib tashlanadigan ma'lumotlarning miqdori natijaviy tasvirni foydalanuvchi tanlagan sifatga bog'liq. Oxirgi pog'onada tayyor bo'lgan fayl Xavfman algoritmgiga muvofiq qisqartiriladi. JPEG faylni dekompressiya qilish teskari tartibda amalga oshiriladi.

Grafik ob'ektni Web-sahifasiga joylashtirish

Web-sahifada tasvirlardan quyidagicha foydalanish mumkin:

- ish yuritish sahifalarida kompaniya logotipi(belgisi) sifatida;
- reklama e'lonlari grafikasi sifatida;
- turli tasvir ko'rinishida;
- diagramma va grafiklar sifatida;
- badiiy shriftlar sifatida;
- sahifa muallifi imzosi sifatida;
- grafik satrdan gorizontal bo'luvchi sifatida;
- grafik markerlardan chiroyli belgilab chiqilgan ro'yxatlar yaratishda.

Brauzerlar tasvirlarning ushbu GIF, JPEG, PNG va BMP formatlari bilan ishlaydilar. Lekin BMP formati katta joy egallashi sababli ulardan foydalanmaslik maqsadga muvofiq bo'ladi. Web da eng ko'p ishlatiladigan grafik formatlar GIF va JPEG formatlaridir. Fotografiya sifatini JPEG formati ta'minlaydi. GIF formati ikki rangli (oq-qora) yoki rangli fotografik tasvirlar uchun juda ham mos emas.

Uning yordamida ko‘proq kam rang ishlatilgan kontur tasvirlar saqlanadi. U yana shaffoflik va animatsiya uchun qulay.

HTML hujjatga grafik ob’ektni joylashtirish tegidan foydalanib amalga oshiriladi. Bu tegning yozilishi quyidagi ko‘rinishga ega:

```
<IMG SRC="URL" ALIGN="qiymat" WIDTH="qiymat"
HEIGHT="qiymat" ALT="matn">
```

Bu erda SRC atributining qiymati siz Web-sahifaga joylashtirmoqchi bo‘lgan grafik ob’ektni adresi. Agar siz sahifaga joylashtirmoqchi bo‘lgan rasm boshqa serverda bo‘lsa, u holda bu atributning qiymati to‘liq URL ko‘rinishida (masalan, <http://www.tdiuserver.uz/myimages/picturename.gif>) beriladi. Agar u lokal serverda ya’ni Web-sahifa va unga qo‘yilayotgan rasm bir serverda joylashgan bo‘lsa, u holda SRC atributining qiymati qisqartirilgan URL ko‘rinishida (masalan, "... /images/picturename.gif") beriladi.

Bu tegning ALIGN atributi yordami bilan rasmni HTML- hujjat yoki jadval katagidagi ma’lum bir pozitsiyaga joylashtirish amalga oshiriladi. Ushbu atributning traditsion qiymatlari (CENTER, LEFT, RIGHT) bilan bir qatorda, uning boshqa qiymatlari ham tegi bilan birga ishlatilishi mumkin. Atributning bu qiymatlari grafik ob’ektni ma’lum bir satrga joylashtirganda uni matn bilan o‘zaro ta’sir rejimini ko‘rsatish uchun xizmat qiladi. Bu holda u quyidagi qiymatlarni qabul qiladi:

- agar ALIGN="TOP" berilgan bo‘lsa, u holda grafik ob’ektning yuqori cheti jadval katagining yuqori chegarasi bo‘yicha yoki matn satrining yuqori chegarasi bo‘yicha tekislanadi.
- agar ALIGN="BOTTOM" berilgan bo‘lsa, u holda grafik ob’ektning quyi cheti jadval katagining quyi chegarasi bo‘yicha yoki matn satrining quyi chegarasi bo‘yicha tekislanadi.
- agar ALIGN="MIDDLE" berilgan bo‘lsa, u holda grafik ob’ektning markazi asosiy satrining o‘rtasi bo‘yicha yoki jadval katagining o‘rtasi bo‘yicha tekislanadi.
- agar ALIGN="BASELING" berilgan bo‘lsa, u holda grafik ob’ekt shartli “asosiy chiziq” bo‘yicha tekislanadi. Agar HTML hujjatga bir necha rasmni

yoki bir rasmning bir necha parchalarini jadvalning turli kataklariga joylashtirish zarur bo'lgan hollarda bu parametrdan foydalanish maslahat beriladi.

Grafik ob'ektni Web-sahifaga joylashtirganda u o'z o'lchamlarini o'zgartirmaydi. Ko'p hollarda rasmni HTML hujjatga qo'shganda uni masishtabini o'zgartirish zaruriyati paydo bo'ladi. Buning uchun HEIGHT va WIDTH atributlaridan foydalanish mumkin. Bu atributlar HTML hujjatga qo'shilayotgan grafik tasvirning balandligi (HEIGHT) va kengligini (WIDTH) ko'rsatadi. Bu atributlarning qiymati piksellar soni bilan beriladi.

Rasmlar bilan bezatilgan Web-sahifalarni yaratayotganda bu tasvirlarni ko'rish vositasiga hamma ham ega emasligini hisobga olish kerak. Ko'p hollarda foydalanuvchilar hujjatni qabul qilish vaqtini kamaytirish maqsadida brouzerning grafik rejimini uzib qo'yadilar. Har ikkala holda ham, agar rasmda nima tasvirlanganini ko'rib bo'lmasa, unda nima tasvirlanganini bilish uchun uni o'qish imkoniyatini berish kerak. Ana shu maqsadda alternativ matndan foydalaniladi. Alternativ matn – bu tasvirning qisqacha tasnifidan iborat bo'lib, undan quyidagi hollarda rasmning o'rnida ko'rish uchun foydalaniladi: 1) agar brouzer grafik ob'ektni ekranda aks ettirish imkoniyatiga ega bo'lmasa; 2) agar brouzer grafik ob'ektlar bilan ishlash rejimi uzib qo'yilgan bo'lsa. Alternativ matnni ko'rish uchun sichkoncha kursorini bir necha sekund tasvirning ustida qimirlatmasdan turish zarur. tegida alternativ matn ALT atributining qiymati ko'rinishda beriladi.

Yuqorida ko'rib o'tilganlardan tashqari tegida quyidagi atributlar ishlatiladi:

vspace – tasvir va matn orasidagi piksellar bilan ifodalanuvchi vertikal masofani beradi. Tasvir yonlariga matn yozish imkoni uchun zarur;

hspace – tasvir va matn orasidagi piksellar bilan ifodalanuvchi gorizontal masofani beradi.;

border - tasvirning piksellardagi ramkasi(chegara chiziqlari). Uning ham ko'rsatilishi shart emas;

align – tasvirni satrning qaysi qismiga chiqarish:

- left – chapga. o'ng tarafidan yozish mumkin;

- right - o'ngga. Chap tarafidan yozish mumkin;
- top va texttop – tasvirning yuqori chegarasini matn satri yuqori chiziqiga tekislash;
- middle – satr negiz chiziqi o'rtasini tasvir markaziga moslab chiqarish;
- absmiddle - joriy satr o'rtasini tasvir markaziga moslab chiqarish;
- bottom va baseline – joriy satr negiz chiziqini tasvirning ostki chiziqiga moslab chiqarish;
- absbottom - joriy satr ostki chiziqini tasvirning ostki chiziqiga moslab chiqarish;

name – joriy xujjat uchun tasvirning unikal ixtiyoriy (raqam va lotin harflaridan, probelsiz)nomini aniqlaydi. Agar siz tasvirga, masalan JavaScript-stsenariylardan, murojaat etmoqchi bo'lsangiz nomni ko'rsatish shart;

lowsrc – SRC parametrida ko'rsatilganiga nisbatan pastroq sifatli(va hajmli!) alternativ tasvirli faylning (URL) adresini beradi. Bu parametr bilan ishlay oladigan brauzerlar oldin LOWSRC tasvirni chiqaradilar, keyin uni SRC ga almashtiradilar;

usemap – tasvirga MAP elementida ko'rsatilgan navigatsion xarita (image map) ni qo'llaydi;

ismap – tasvirni server ishlatadigan navigatsion xarita (image map) sifatida aniqlaydi. Uni faqat tasvir gipermurojat bo'lganidagina ishlatgan ma'qul. Sichqon tugmasi kursor tasvir ustidaligida bosilsa, shu nuqtaning x,y koordinatlari serverga jo'natiladi. Shu koordinatlarga qarab server u yoki bu gipermurojaatli xujjatni ochib beradi. Bu parametr – bayroqcha, qiymat talab qilmaydi. Lekin server uchun navigatsion xaritalarni imkon boricha yaratmaslik maqsadga muvofiq;

dynsrc – video fayllarni ko'rish imkonini beradi. Barcha brauzerlarda ishlaydi. Masalan:

```
<img dynsrc=my1.avi src=my1.jpg width=100 height=100>
```

Agar bu misol Internet Explorer yordamida ishlatilsa 100x100 darchada avi fayl ko'rsatiladi. Agar mabodo brauzer dynsrc= parametri bilan ishlamasa ekranga jpg tasvir chiqadi.

Web-masterlarning oltin qonuni – doimo height va width o‘lchamlarini aniq ko‘rsatib, tasvir yuklanishidan oldinroq brauzerda darcha ajratishdir. Aks xolda xujjat har bir tasvir uchun yangidan chizilaveradi, past tezlikli mashinalar va modemlar uchun bu noxush xolatdir.

Bir vaqtning o‘zida barcha parametrlar birgalikda ishlatilishi mumkin, masalan:

```

```

Bu misolda rasm chapga chiqariladi, o‘ngidan 30 piksel narida matn yoziladi va yuqoridan 5 piksel masofa qoladi. Agar unga kursorni keltirsangiz "mening rasmim " degan yozuv chiqadi.

2.2. JavaScript dasturlash tili.

2.2.1. HTML-sahifada JavaScript ni joylashtirish.

JavaScript skriptining kodi bevosita HTML- sahifada joylashtiriladi. Bu qanday amalga oshirilishini biz quyidagi misolda ko‘rib chiqamiz:

```
<html>
```

```
<body>
```

```
<br>
```

Bu oddiy HTML-xujjat.

```
<br>
```

```
<script language=«JavaScript">
```

```
document.write("Bu esa JavaScript!")
```

```
</script>
```

```
<br>
```

Ushbu joydan yana HTML boshlanadi.

```
</body>
```

```
</html>
```

Birinchi qarashda yuqoridagi misol oddiy HTML faylni yodga soladi. Bu erda bitta yangilik bu quyidagi konstruktsiya:

```
<script language=«JavaScript">
```

```
document.write("Bu esa JavaScript!")  
</script>
```

Bu xaqiqatdan xam JavaScript kodi. Bu skript qanday ishlashini ko‘rishni hoxlasangiz, bu misolni oddiy HTML fayl kabi saqlang va brauzerga yuklang. Natijada siz 3 qatordan iborat matnni ko‘rasiz:

Bu oddiy HTML-hujjat

Bu esa JavaScript!

Ushbu joydan yana HTML boshlanadi.

JavaScript dan foydalanish uchun HTML tilini bilish yetarli deb hisoblaysizmi?

Buni tan olish kerakki, ushbu skript unchalik xam foydali emasdir, chunki ushbu ishni faqat HTML dan foydalanib ham yozish mumkin edi. Bu misolda faqat `<script>` tegi haqida tushuncha berish maqsad qilingan xolos. `<script>` va `</script>` teglari orasidagi barcha ma’lumotlar JavaScript tilidagi kod sifatida interpretatsiya qilinadi. Misolda siz `document.write()` instruktsiyasini ishlatilishini ko‘rdingiz; ushbu instruktsiya JavaScript tilida dasturlashdagi eng muhim buyruqlardan biri hisoblanadi. `document.write()` buyrug‘i joriy hujjatga biror-bir ma’lumot yozish kerak bo‘lib qolganda ishlatiladi. Masalan bizning misolda HTML hujjatdagi JavaScript da yozilgan kichik dasturcha quyidagi yozuvni ekranga chikaradi: “Bu esa JavaScript!”

JavaScript ni “tushunmaydigan” brauzerlar

Agar brauzer JavaScript skriptlarini qabul kilmasa bizning sahifa qanday ko‘rinishga ega bo‘ladi? JavaScript ni qabul qilmaydigan brauzerlar `<script>` tegini ham qabul qilmaydi. Brauzer tegni inobatga olmaydi va ushbu tegdan so‘ng yozilgan barcha kodni oddiy matn singari namoyon etadi. Boshqacha qilib aytganda, bizning misolda keltirilgan JavaScript kodi oddiy matn ko‘rinishida HTML hujjatning o‘rtasida paydo bo‘lib qoladi. Bu esa bizning rejalarimizga kirmaydi. Bunday vaziyatlarda brauzerlarning oldingi versiyalaridan skriptni boshlang‘ich (isxodnik) kodini olish uchun biz HTML tilidagi `<! - - >` izox tegini ishlatamiz. Natijada bizning kodni yangi varianti quyidagi ko‘rinishda bo‘ladi:

```
<html>
```

```
<body>
```

```
<br>
```

Bu oddiy HTML-hujjat.

```
<br>
```

```
<script language=«JavaScript»>
```

```
<!--kodni eski brauzerlardan berkitadi
```

```
document.write("Bu esa JavaScript!")
```

```
// -->
```

```
</script>
```

```
<br>
```

Ushbu joydan yana HTML boshlanadi..

```
</body>
```

```
</html>
```

Bu holda JavaScript ni qabul qilmaydigan brauzerlar quyidagini chop etadi:

Bu oddiy HTML-hujjat

Ushbu joydan yana HTML boshlanadi.

HTML ning izox tegisiz JavaScript ni qabul qilmaydigan brauzerlar quyidagini chop etadi:

Bu oddiy HTML-hujjat

```
document.write("Bu esa JavaScript!")
```

Ushbu joydan yana HTML boshlanadi.

Quyidagiga e'tibor bering: JavaScript ning boshlang'ich kodni (isxodnik) butunlay berkita olmaysiz. Biz amalga oshirayotgan ishlar skript kodini ko'chirishni oldini olish maqsadida qilmoqdamiz lekin foydalanuvchi kodni brauzer menyusining 'View document source' punktidan ko'rishingiz mumkin. Sizning boshlang'ich kodingizni berkitishni boshqa usullari mavjud emas.

JavaScript da ob'ektlar ierarxiyasi.

JavaScript tilida barcha web-saxifadagi barcha elementlar ierarxik tuzilishda joylashtiriladi. Har bir element ob'ekt sifatida namoyon bo'ladi. Va har bir shunday ob'ekt ma'lum xususiyat va uslub (metod)larga ega bo'ladi. O'z

navbatida JavaScript tili sizga web-sahifadagi ob'ektlarni oson boshqarish imkonini beradi, lekin shu bilan birga HTML dagi ob'ektlar ierarxiyasini tushunib olish juda muhimdir. Bu qanday amalga oshirilishini siz quyidagi misoldan ko'rib olishingiz mumkin. Oddiy HTML sahifani ko'rib o'tamiz:

```
<html>
<head> </head>
<body bgcolor=#ffffff>
<center>
<img src=«home.gif" name=«pic1" width=200 height=100>
</center>
<p>
<form name=«myForm">
Name:
<input type=«text" name=«name" value=«"><br>
e-Mail:
<input type=«text" name=«email" value=«"><br><br>
<input      type=«button"      value=«Push      me"      name=«myButton"
onClick=«alert('Yo')">
</form>
<p>
<center>
<img src=«ruler.gif" name=«pic4" width=300 height=15>
<p>
<a          href=«http://rummelplatz.uni-mannheim.deg'~skochg'">My
homepage</a>
</center>
</body>
</html>
```

2.2.2. Funksiya va usullar. JavaScript buyruqlari.

Hodisa va hodisaga ishlov beruvchi (obrabotchik) lar JavaScript dasturlash tilinig eng muhim qismlaridan biridir. Hodisalar , asosan foydalanuvchi tomonidan

bajarilgan amallar bilan bog‘liqdir. Masalan, agar u biror-bir tugmani bosganda "Click" hodisasi ro‘y beradi. Agar sichqoncha ko‘rsatkichi biror-bir ishorat ustidan o‘tsa MouseOver hodisasi ro‘y beradi. Hodisalarning bir necha xillari mavjud. Biz o‘zimizning JavaScript dasturimizni ushbu hodisalardan bir nechtasiga javob berishini ta‘minlashimiz mumkin. Bu amallarni hodisalarga ishlov beruvchi maxsus dasturlar yordamida bajarishimiz mumkin. Shunday qilib, masalan biror-bir tugma bosilganda oyna ochilishi yaratilishi mumkin. Bu esa oyna ochilishi Click hodisasiga javobini bildiradi. Bu holdagi hodisaga ishlov beruvchi dastur onClick deb nomlanadi. Va u kompyuterga ushbu hodisa sodir bo‘lganda nima qilishni bildiradi. Quyida onClick hodisasiga ishlov beruvchi dastur ishini ko‘rsatuvchi kodning misoli keltirilgan:

```
<form>  
<input type="button" value="Click me" onClick="alert('Yo')">  
</form>
```

Ushbu misol bir necha yangi xususiyatlarga ega- ularni navbatma- navbat ko‘rib o‘tamiz. Bu erda biz qandaydir tugma joylashgan forma yaratganimizni ko‘rishingiz mumkin. Birinchi yangilik bu <input> tegidagi onClick="alert('Yo')" yozuvdir. Yuqorida aytib o‘tilganidek ushbu atribut tugmani bosilganda nima sodir bo‘lishini belgilaydi. Shunday qilib Click hodisasi mavjud bo‘lsa kompyuter alert('Yo') ni bajarishi kerak. Bu esa JavaScript tilidagi kodni bildiradi (E’tibor bering, bu misolda biz <script> tegidan ham foydalanmadik). alert() funksiyasi sizga paydo bo‘luvchi oynalar yaratish imkonini beradi. Ushbu funksiyadan foydalanilganda qavs ichida qandaydir qatorni berishingiz lozim bo‘ladi. Bizning holatda bu 'Yo'. Bu esa paydo bo‘luvchi oynadagi matndir. Shunday qilib, foydalanuvchi tugmani bosganda bizning skript 'Yo' matnidan iborat oyna yaratadi.

Yuqoridagi misolni yana bir xususiyati tushunmovchilikni keltirib chiqarishi mumkin: document.write () buyrug‘ida biz ikkitalik qo‘shtirnoq (“ ”) ishlatgan edik, alert() konstruktsiyasida esa faqat bittalik (‘ ’) qo‘shtirnoq ishlatildi. Nimaga? Ko‘p xollarda siz qo‘shtirnoqni ikki ko‘rinishidan ham foydalanishingiz mumkin. Lekin so‘nggi misolda biz onClick="alert('Yo')" deb yozdik, ya’ni biz qo‘shtirnoqni ikki xilidan foydalandik. Agar biz onClick="alert("Yo")" deb

yozganimizda kompyuter bizning skriptda adashib ketishi mumkin edi, chunki u holda onClick hodisaga ishlov beruvchi funksiya konstruktsiyaning qaysi qismiga tegishli qaysi biriga tegishli emasligi noma'lum bo'lib qoladi. Shuning uchun siz bu holda ikki xildagi qo'shtirnoqdan foydalanishingiz zarurdir. Siz qo'shtirnoqni kaysi turini birinchi, qaysinisini ikkinchi bo'lib ishlatishingiz unchalik katta ahamiyatga emas. Ya'ni siz onClick='alert("Yo")' deb ham yozishingiz mumkin

Funksiyalar

JavaScript tilida yozilgan ko'p dasturlarda biz funksiyalarda foydalanamiz. Shuning uchun tilning ushbu elementi haqida batafsilrok to'xtalamiz. Ko'p hollarda funksiyalar bir necha buyruqlarni bog'lovchi vosita sifatida ishlatiladi. Keling misol uchun, biror-bir matnni uch marotaba chop etadigan skriptni yozaylik. Avval eng oddiy yondashuvni ko'rib o'tamiz:

```
<html>
<script language=«JavaScript»>
<!-- hide
document.write("Mening saxifamga xush kelibsiz!<br>");
document.write("Bu JavaScript!<br>");

document.write("Mening sahifamga xush kelibsiz!<br>");
document.write("Bu JavaScript!<br>");

document.write("Mening sahifamga xush kelibsiz!<br>");
document.write("Bu JavaScript!<br>");
// -->
</script>
</html>
```

Bunday skript quyidagi matnni uch marotaba yozadi:

Mening sahifamga xush kelibsiz!

Bu JavaScript!

Agar skriptning boshlang'ich kodiga qaralsa, matn uch marta chop etilishi uchun kodning ma'lum bir qismi uch marta qaytarilganligi ko'rinadi. Bu esa effektiv emasdir. Ushbu masalani biz undan ham yaxshiroq hal etishimiz mumkin. Buni quyidagicha amalga oshiramiz

```
<html>
<script language=«JavaScript»>
<!-- hide
function myFunction() {
    document.write("Mening sahifamga xush kelibsiz!<br>");
    document.write("Bu JavaScript!<br>");
}
myFunction();
myFunction();
myFunction();
// -->
</script>
</html>
```

Ushbu skriptda qandaydir bir funksiyani kiritdik, bu funksiya quyidagi qatorlardan iborat:

```
function myFunction() {
    document.write("Mening sahifamga xush kelibsiz!<br>");
    document.write("Bu JavaScript!<br>");
}
```

Skriptning barcha buyruqlari figurali qavslar ichida joylashtiriladi; {}-qavslari myFunction() funksiyasiga tegishli bo'lib, ikki document.write() buyruqi birlashtirilgan va myFunction() funksiyasi chaqirtirilganda bajariladi. Haqiqatdan ham bizning misolda ushbu funksiya uch marotaba chaqirtirilgan. Misolda ko'rinadiki, funksiyaning o'ziga ta'rif berilgandan so'ng myFunction() qatorini uch marotaba yozildi. Ya'ni uch marotaba murojaat qilindi.

O'z navbatida, ushbu funksiyadagi buyruqlar uch marotaba bajarilganligi ko'rinib turibdi. Bu misol oddiy misol bo'lganligi uchun sizda JavaScript da

funksiyalar nimaga shunchalik zarur degan savol paydo bo‘lishi mumkin. Siz quyida ularni foydasini bilib olasiz. Funksiyani chaqirtirilganda ma’lumotlarni uzatish imkoniyati bizning skriptlarga haqiqiy egiluvchanlikni beradi.

Shuningdek funksiyalar hodisalarga ishlov beruvchilar bilan birgalikda ishlatilishi mumkin. Misolni ko‘rib o‘tamiz:

```
<html>
<head>
<script language=«JavaScript»>
<!-- hide
function calculation() {
    var xq 12;
    var yq 5;
    var resultq x Q y;
    alert(result);
}
// -->
</script>
</head>
<body>
<form>
<input type=«button» value=«Calculate» onClick=«calculation()»>
</form>
</body>
</html>
```

Bu misolda tugma bosilganda calculation() funksiyasi chaqirtiriladi. Ko‘rinib turibdiki, ushbu funksiya x.y va result o‘zgaruvchilardan foydalanib qandaydir hisob-kitob ishlarini olib bormoqda. O‘zgaruvchini var kalit so‘z yordamida berishimiz mumkin. O‘zgaruvchilar turli xildagi kattaliklarni saqlash uchun ishlatilishi mumkin, masalan, son, qator, matn va boshqa. Shunday qilib skriptning var result= x Q y; qatori brauzerga result o‘zgaruvchini yaratish va unga arifmetik operatsiya x + y (ya’ni 5 + 12)ni natijasini joylashtirishni buyuradi.

Shundan so‘ng result o‘zgaruvchisiga 17 soni joylashtiriladi. Bizning holda alert(result) buyrug‘i alert(17) bajargan ishni bajaradi. Boshqacha qilib aytganda biz 17 son yozilgan oynani yaratdik

2.3. PHP dasturlash tili.

2.3.1. PHP Web dasturlash tili haqida ma’lumot.

PHP –o‘z nomini yetarlicha tanitib ulgurgan, dasturlash tili hisoblanadi. Gap shundaki, boshlanishda bu uncha qiyin bo‘lmagan shaxsiy WEB –sahifalarini yaratish uchun mo‘ljallangan oddiy makroslar to‘plamidan iborat bo‘lgan bo‘lib, PHP-personal home page (shaxsiy uy sahifasi) so‘zlarining qisqartmasidan iborat.

Tez orada u foydalanuvchilar nazariga tusha boshladi, hamda tezlik bilan takomillashib, ommalashib bordi. 1997 yildan bu til ustida programistlar guruhi ish olib boradi.

Mexnatlarning samarasida esa PHP3 keyingi versiya yaratildi. Bu PHP ning takomillashgan va zamonaviy versiya bo‘lib, unda matnlarining qayta ishlashning yangi usullari yaratildi va bu usullar Zib Zuraski va endi Gutians (Zeev, Surasky, Andi Ceutmans) ismli progromistlar tomonidan yaratildi. Shuningdek tilning sintaksisida biroz o‘zgarishlar kiritilib, yangi funksiyalar qo‘shildi. Yangi vertsiya shu davrda sarver uchun dasturlash tillarining eng zo‘ri hisoblanib, juda ham tez ommalashib ketdi.

MySQL ma’lumotlar bazasi va Apache serveri bilan ishlash uchun PHP ning imkoniyatlari yanada kengayib bordi. Apache serveri hozirgi kunda dunyodagi eng keng tarqalgan Web -server hisoblanadi va PHP tili Apache serveri uchun modul ko‘rinishida qo‘llanilishi mumkin. MySQL- bu zamonaviy ma’lumotlar bazasi bo‘lib pulsiz (tekin) tarqatiladi, shuning uchun ham PHP ning barcha funksiyalari shu bazaga bog‘langan. Tan olish lozimki Apache, MySQL va PHP larning o‘zaro bir-biri bilan bog‘liq ravishda ishlashi o‘rtadagi raqobatga barham beradi.

Bu esa PHP boshqa MBSI bilan ishlamaydi degani emas. Bu texnologiya juda MBSI va Web serverlar bilan ishlash imkoniyatiga ega.

WEB sahifalarni va tarmoqni yaratish yo‘llari o‘zgarishi bilan PHP ham takomillasha bordi.1990 -yil o‘rtalariga kelib katta tarmoqlarda ham HTML da

yozilgan yuzlab statik sahifalar ishlatilar edi. Hozir esa jarayon o'zgarib bormoqda. WEB sahifalarini yaratuvchilar ma'lumotlar bazasi bilan ishlovchi WEB sahifalarni yaratish imkoniga ega bo'lib, foydalanuvchilarni qayta ishlash imkoniga ega bo'lgan WEB sahifalarni yaratmoqdalar.

Ma'lumotlarni saqlash va ma'lumotlarga murojat qilish uchun ma'lumotlar bazasidan foydalanish yanada aktuallashib, mobil telefonlar, raqamli televideniya va xokazolar. Turli xil qurilmalarda ma'lumotlarni uzatishda sifatni o'sishiga erishishmoqda.

Bu fikrlar asosida aytish mumkinki, kelgusida PHP tili yanada takomillashib o'zining o'ta yuqori darajadagi dasturlash tili ekanligini namoyon qiladi.

PHP turli xil sistemalarda ishlay oladi. U sistema Windows, Unix ning ko'plab versiyalari, shuningdek Linux va hatto Macintosh bo'lishi mumkin. PHP ko'plab tarmoq serverlarida, xususan Apache, Microsoft Internet Information Server, Web Site Pro, Iplanet Web Server va Microsoft Personal Web Server – larda ishlashi mumkin. Agar o'zimiz tuzgan dasturlarimizni Windows tizimida tekshirishni xohlasak oxirgi sanab o'tilgan serverdan foydalanishimiz mumkin, hatto Apache serveri Windows sistemasi boshqaruvida ishlasa ham.

PHP interpretatori yordamida dasturni alohida mustaqil ko'rinishda kompilyasiya qilish mumkin. U holda dasturni mustaqil ishga tushirish mumkin. PHP tilini yaratishda ma'lumotlar bazasi bilan bog'lanish talablarini alohida e'tiborga olingan. Ko'plab ma'lumotlar bazalarini PHP da o'qish mumkin. Masalan bularga Adabas D, InternetBase, Golid, dBase, mSQL, Sybase, Empress, MySQL, Velosic, FilePro, Oracle, Unixdbm, Informix va hokazolarni keltirishimiz mumkin. Shuningdek, PHP ODBC standartini ham o'qiy oladi. Ushbu qo'llanmada esa biz Linux, Apahe va MySQL serverlar asosida fikr yuritamiz. Bu uchta dasturlar majmualari keng foydalanish mumkin bo'lgan

PHP4 ni <http://www.php.net> tarmog'idan olish mumkin. Buning uchun esa, alohida kredit kartochka talab kilinmaydi.

PHP ni web-tarmog'i progromistlar uchun ajoyib manba hisoblanadi. <http://www.php.net/manual> adresida boshqa progromistlarning izohlari, fikrlash va

tankidiy fikrlari keltirilgan ma'lumotlarni va qo'llanmalarni olish mumkin. Bu qo'llanmalarni turli xil formatlarda olish mumkin.

PHP.ini fayli. PHP ni o'rnatib bo'lgandan so'ng, uning nastroykasini PHP.ini fayli yordamida o'zgartirish mumkin. Agar kompyuterga UNIX operatsion sistemasi o'rnatilgan bo'lsa, bu fayl /usr/local/lib katologida, agar Windows operatsion sistemasi o'rnatilgan bo'lsa, Windows katologida joylashgan bo'ladi.

Agar dasturlash tili omma uchun (tekin) bo'lsa, albatta yordamni Internet dan yetarlicha topish mumkin. PHP tili bo'yicha siz bilan fikr almasha oluvchi yana bir yaxshi manba bo'lib, u PHP bilimlar bazasi, yani knowledge Base deb nomlanadi, va uni <http://www.fagts.com/knowledge-base/index.phtml> deb nomlanadi.

Birinchi dastur.

Biz yuqorida PHP ni o'rnatish va o'rganish bo'yicha boshlang'ich ma'lumotlarni oldik. Endi esa, PHP da birinchi dasturimizni yaratishga harakat qilamiz. PHP da dastur tuzish uchun HTML dagi kabi WEB-sahifa yaratishni bilishimiz lozim.

Ushbu dasturda esa, quyidagilarni o'rganishga harakat qilamiz:

PHP-programmalarini yaratish, setverga nusxasini ko'chirish va ishga tushirish PHP komandalari va HTML matnini bitta hujjatga bog'lash, Dastur matnini tushintirish uchun izoh keltirish.

PHP-dasturni tuzish uchun ixtiyoriy matn redaktorini ishga tushiramiz. PHP – dasturi HTML- hujjatlari kabi oddiy matndan tashkil topadi. Shuning uchun dasturni ixtiyoriy matn redaktorida, agar unix bo'lsa, VI-yoki Emacs da ham yozish mumkin. Quyidagi dastur matnini kiritaylik va uni first.php deb saqlaylik:

```
< ? php
    print "hello web";
?>
```

ushbu dastur kengaytmasi albattda . PHP bo'lishi shart. Agar dastur server emas, balki kletn kompyuterida tuzilgan bo'lsa, dastur faylini serverga kiritish uchun FTP servisdan foydalanishga to'g'ri keladi. Agar dasturda xatoliklar mavjud bo'lmasa, u holda natijani brauzer oynasida ko'rish mumkin.

Agar dasturda yoki uning kengaytmasida xatolik mavjud bo'lsa, brauzer oynasida dastur matnining o'zi namoyon bo'ladi.

PHP-programmani yozishda interpretatorga oddiy HTML-tekstdan farqini belgilovchi komandani berishimiz lozim. Aks holda komandalar HTML-tekst sifatida qabul qilinib brauzer uzatiladi. Quyidagi jadvalda PHP-komandalarini berishning 4-xil usuli keltirilgan.

Teg turlari	Ochuvchi teg	Yopuvchi teg
Standart	< ? php	?>
Qisqacha	< ?	?>
ASP	< %	%>
Dasturiy	< script language="php">	</script

Jadvalda keltirilgan standart va dasturiy teglar PHP ning ixtiyoriy konfiguratsiyasida ishlaydi. Qisqa va ASP teglar ishlashi uchun php.ini faylida aniq keltirilgan bo'lishi lozim. Qisqa teglar foydalanilishi uchun php.ini faylida quyidagi direktiva yozilgan bo'lishi lozim:

Short_open_tagkon;

Standart holda bu derektiva yoqiq bo'ladi, shuning uchun phph.ini faylini fakat off qilish uchungina tahrirlashimiz mumkin. ASP tegida foydalanish uchun esa, quyidagi direktiva yoqilgan bo'lishi lozim:

Asp_tagskon;

Php.ini faylini tahrir qilib, ixtiyoriy direktivadan foydalanish mumkin. Quyidagi barcha teglar uchun yozilgan dastur matnini ko'raylik:

```
<?
print "hello web";
?>
<? php
print "hello web";
?>
```

```

<%
print "hello web";
%>

<script language="php">
print "hello web";
</script>

```

print () funksiyasi

Bu funksiya ma'lumotlarni ekranga chop qilish uchun foydalaniladi. Ekranga beriladigan ma'lumotlar () ichiga beriladi. Ekranga beriladigan har bir ma'lumot " " ichiga berilishi lozim. Shuningdek Print dan keyin () qo'yish ham, qo'ymaslik ham mumkin.

HTML va PHP ning birgalikda ishlatilishi. Quyida keltirib o'tiladigan dastur faqat PHP komandalaridan tashkil topgan. Lekin ochuvchi va yopuvchi teglar ko'shish orqali HTML- tekst ko'shib HTML va PHP ning aralash hujjatning yaratish mumkin:

```

<html>
<head>
    <title> php va html hujjati </title>
</head>
<body>
    <b>
        <?php>
            print "hello world!";
        ?>
    </b>
</body>
</html>

```

Agar dasturni saqlab (.PHP) uni brauzer orqali ishga tushirsak, ekranda (jirno'y) hello world! hosil bo'ladi.

2.3.2. PHP – dasturida komentariyalar (izohlar).

Ma'lumki, dasturni tuzish jarayonida hamma narsa tushunarli va oddiy dasturni ham tushunib olish qiyin bo'lib qoladi. Shuning uchun ham, dastur yozish jarayonida albatta izoh keltirib borish zarur, chunki sizni dasturingizdan foydalanuvchiga biroz osonroq bo'ladi.

Komentariya (izoh)- bu interpretorda qayta ishlanmaydigan, dasturdagi oddiy matn hisoblanadi. Izoh dasturni tushinish uchun yengillik yaratish maqsadida foydalaniladi.

O'zgarmlar. O'zgaruvchilardan foydalanish ma'lumotlarni saqlash va qayta ishlashning eng ajoyib usullaridan biri hisoblanadi. Biz bu o'zgaruvchilar qiymatini, hattoki tipini hoxlagan vaqtimizda o'zgartira olishimiz mumkin. Lekin dastur yaratish jarayonida shunday holat bo'lishini istab qolamizki, bunda bizni bergan qiymatimiz yoki ma'lumotimiz dasturni yaratish va foydalanish jarayonida, kerak bo'lsa tasodifan o'zgarib ketishdan holi bo'lsin, ya'ni o'zgarmlar. Buning uchun o'zgarmlardan foydalanishimiz zarur. RNR4 da o'zgarmlar bilan ishlovchi define() funksiyasi manashu maqsadda ishlatiladi. Bu funksiya yordamida yaratilgan o'zgarmlar (konstanta) dasturni yaratish va foydalanish jarayonida o'zgartirilmaydi. O'zgarmlarni tashkil etish uchun define() funksiyasida o'zgarmlar nomi va uning qiymati beriladi.

Define ("nomi",42)

O'zgarmlar qabul qiladigan ma'lumot qiymat va simvol bo'lishi mumkin. O'zgarmlar nomi esa faqat katta harflar bilan beriladi. O'zgarmlardan foydalanishda faqat katta harflar bilan \$ belgisini oldiga qo'ymasdan yoziladi.

Quyidagi o'zgarmlardan foydalanish dasturda qanday ishlatilishiga misol sifatida keltirib o'tamiz:

```
<html>
<head>
<title>O'zgarmlarni tashkil etish</title>
</head>
<body>
<?php
```

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

Agar dasturga ahamiyat bersak, «Welcome» yozuvidan so'ng konkatensatsiya (.) belgisidan foydalandik.

PHPda yana bir imkoniyat borki, u foydalanuvchiga avtomatik ravishda bir nechta o'zgarma tashkil etib beradi. Masalan, `_FILE_` o'zgarma faqat foydalanilayotgan aktiv fayl nomini o'zida saqlaydi. `_LINE_` ishlatilayotgan fayldagi aktiv qatorni aniqlab beradi va h.k.

Ma'lumki, jarayonlar chiziqli, takrorlanuvchi va tarmoklanuvchi bo'lishi mumkin. Hozirgacha o'rgangan va ko'rgan misollarimiz faqat chiziqli jarayonga tegishli edi. Bu mavzuning asosiy maqsadi esa, PHP4 tilida tarmoqlanuvchi va takrorlanuvchi jarayonlar uchun qanday funktsiya, protsedura va instruktsiyalardan foydalanilishi bilan tanishtirishdan iborat. Iz asosan quyidagilarni o'rganamiz:

If instruktsiyasi yordamida dasturdagi ba'zi bir shartlarni qanday tekshirilishi va bajarilishi;

Shart bajarilmagan qism uchun dasturning alternativ bo'limini qanday yaratish;

Switch yordamida ba'zi bir ifodalarni qiymatini e'tiborga olgan holda dastur qismini qanday yaratish ;

While yordamida tsiklik jarayonni tashkil etish;

For yordamida tsiklik jarayonlarni tashkil etish;

Tsiklik jarayonlarni tuxtatish qanday bajariladi;

Bir necha tsiklik jarayonlarni birgalikda qanday tashkil etish.

2.3.3. PHP4 da shartli jarayonlarni ifodalash.

Juda ko'p dasturlar ma'lum bir shartlar sababli o'zining holatini o'zgartirib turadi. Ma'lum bir aniq shartlar asosida dasturning takomillashuvchi yaratilayotgan Web sahifalarning dinamikligini yanada mustahkamlaydi. Boshqa

dasturlash tillari kabi PHP4 tili ham shartli instruktsiyalardan foydalanadi. Bu instruktsiya PHP4 da if yordamida amalga oshiriladi.

IF operatoridan foydalanish. IF instruktsiyasidan foydalanilayotganda hisoblash jarayoni uchun shart qavslar ichida tekshiriladi. Agar shart bajarilsa (true bo'lsa) blok ichidagi ifoda bajariladi, aks holda blok ichidagi ifoda to'liq tashlab yuboriladi. IF instruktsiyasining umumiy ko'rinishdagi sxemasini quyidagicha keltirish mumkin:

IF (shart)

```
{  
// bu erdagi ifoda agar berilayotgan shart rost bo'lsa bajariladi  
}
```

Keltirib o'tilgan umumiy sxemani yanada aniqroq tushunib olish uchun quyidagi misolni keltirib o'taylik.

```
<html>  
<head>  
<title>IF yordamida shartli ifodalarni qo'llash </title>  
</head>  
<body>  
<?php  
$mod="sad";  
if ($mod="happy")  
{  
print "Mening kayfiyatim juda ham yaxshi!"  
}  
?>  
</body>  
</html>
```

Operator yordamida \$mood O'zgaruvchi qiymatini "happy" qatori bilan solishtiramiz. Agar shart bajarilsa, blok ichidagi yozuv ekranga beriladi, aks holda berilmaydi. Keltirib o'tilgan misolda blok ichidagi instruktsiya ko'p emas, shuning uchun bu blokni yanada ixchamroq ko'rinishda quyidagicha keltirish mumkin:


```
if ($mod="happy")  
print "Mening kayfiyatim juda ham yaxshi!";
```

IF instruktsiyasining else bloki. IF instruktsiyasini keltirish jarayonida beriladigan shart bajarilmasa alternativ blok keltirishga majbur bo‘lib qolinadi. Bunning uchun esa, if shartdan keyingi blokdan so‘ng else blokini ham berish lozim. Umumiy holda bu holatni quyidagicha keltirish mumkin:

```
If (shart)  
{  
//bu blokda ifodalar shartni tekshirish natijasi rost bo‘lganda bajariladi  
}  
else  
{  
// bu blokda ifodalar shartni tekshirish natijasi rost bo‘lganda bajariladi  
}
```

Keltirib o‘tilgan umumiy sxemani yanada mustahkamlash uchun quyidagi misolni ko‘rib o‘taylik:

```
<html>  
<head>  
<title>IF yordamida shartli ifodalarni qo‘llash va else blokini tekshirish  
</title>  
</head>  
<body>  
<?php  
$mod="sad";  
if ($mod="happy")  
{  
print "Mening kayfiyatim juda ham yaxshi!"  
}  
else  
{
```

```
print "Menda $mood";  
}  
?>  
</body>  
</html>
```

\$mood O'zgaruvchisi bu misolda sad qatorini qabul qiladi, bu esa "happy" qatori bilan bir xil emas, shuning uchun ham shart bajarilmaydi. Bu esa, birinchi blokni bajarilmasligini bildirib, keyingi blokda keltirib o'tilgan ifodani ekranga berilishi bildiradi.

IF instruktsiyada keltiriladigan else bloki murakkab masalalarni yechish imkoniyatini beradi. PHP4 tilida esa, yanada murakkabrok bo'lgan shartlarni tekshirish imkoniyatlari mavjud bo'lib bu shartlarni biz keyingi mavzularda keltirib o'tamiz.

Funksiya. Funksiya – bu yaxshi ishlangan dasturning eng asosiy komponenti hisoblanadi. Ular yaratilayotgan dasturning tushunarligini juda ham yaxshi bo'lishiga va foydalanishga qulay bo'lishiga katta yordam beradi. Ixtiyoriy yaratilayotgan dastur uning yordamisiz biror katta natijaga erishgan emas. Aytmoqchimizki, funksiyalar dasturlash bilan shug'ullanishni boshlaganlar va davom ettirayotgan uchun juda ham zarur va qulay hisoblanadi. Funksiyalar va ularning yordamida ko'p ishlatiluvchi bir xil tipdagi o'zgaruvchilar ustida ko'p martalik hisoblashlarni qanday qilib ixchamlashtirish mumkinligini o'rganamiz.

Funksiyalarni qanday yaratish va ularga murojaat qilish, funksiya parametrlariga qiymatlarni qanday uzatish va qaytarilgan qiymatlarni qabul qilib olish, matnli o'zgaruvchida uning nomini saqlagan holda qanday qilib funktsiyaga dinamik murojaat qilish , global o'zgaruvchilarga funktsiyadan turib qanday murojaat qilish, «Eslash» funksiyalarini qurish, ko'rsatma bo'yicha funksiya argumentini uzatish kabi narsalarni o'rganamiz.

Funksiya nima? Funksiyani shunday qurol sifatida tasavvur qilish mumkinki, biz unga boshlang'ich ma'lumotlarni ya'ni hom ashyolarni berib tayyor mahsulot olishimiz mumkin. Funksiya bizdan boshlang'ich bir yoki bir necha ma'lumotlar, qiymatlar oladi hamda o'zida ularni biz qo'ygan maqsadimizga qarab

qiymatlarni natija sifatida qaytaradi. Shuning uchun ham funksiya maqsadga bog‘liq ravishda ixtiyoriy vazifani bajaruvchi instrument hisoblanadi.

Masalan, oddiy misol sifatida uyda pechene pishirishni olib qaraylik. Agar bir-ikki marta pishirsak uni qo‘lda tayyorlashimiz mumkin. Lekin, agar bozor uchun har kuni tayyorlaydigan bo‘lsak, u holda unga mos ravishda avtomat qidirib qolamiz. Ushbu misol asosida shunday savol tug‘iladi. «Bu funksiya bizni bajarmoqchi bo‘lgan ishimizdagi takrorlanishlar uchun qanday ahamiyatga ega?». Bu esa jarayonni qonuniy va ko‘p marta takrorlanishi bilan bog‘liq.

U holda funkسيyani dasturimizda murojaat qilish mumkin bo‘lgan alohida nomdagi dastur sifatida tasavvur qilishimiz mumkin. Bu dasturiy funksiya faqat murojaat qilingandagina ishlaydi. Funksiya qaytargan qiymatalarni biz natija sifatida chop qilishimiz mumkin yoki dastur davomida foydalanishimiz mumkin.

Yangi termin Funksiya – bu instruksiya va operatorlardan iborat qism dastur bo‘lib, u asosiy dasturlarda zarur vaqtlarda foydalaniladi. Funksiyalar tayyor qurilgan yoki dasturchi tomonidan qurilgan bo‘lishi mumkin. Ular ba’zi bir boshlang‘ich ma’lumotlarni o‘zida foydalanish uchun so‘rashi va natijaviy qiymatni qaytarishi mumkin.

Funksiyalarga murojaat. Funksiya ikki xil bo‘lishi mumkin. Bular tilning kompilyatorida mavjud funksiya hamda dasturchi tomonidan yaratiladigan funksiya. Birinchi ko‘rinishdagi funksiya haqidagi deyarli ma’lumotga egamiz. Chunki yuqorida keltirib o‘tgan print (“Hello, Web”); funkiyamiz kompilyatorga tegishli. Lekin print() funksiyasi oddiy funksiya hisoblanmaydi, chunki bu funksiya o‘zining argumenti uchun qavslarni qo‘yish shartini qattiq talab qilmaydi. U holda print (“Hello,Web”) va print “Hello,Web” funksiya ekvivalent konstruktsiyalar hisoblanadi. Lekin faqat shu funksiya bunday ko‘rinishga ega bo‘la oladi. qolgan barcha funksiya qavslar qo‘yilishini hatto argument talab qilmaganda ham talab qiladi.

Funksiya albatta o‘zining nomiga ega bo‘lishi lozim. Yuqorida keltirib o‘tgan funkiyamiz nomi esa print va uning argumenti qavs ichida beriladi. Funksiyada bir yoki bir necha argumentlarning bo‘lishi mumkin. Agar ko‘p bo‘lsa

ular bir-biridan vergal(,) bilan ajratib yoziladi. Ularni umumiy holda quyidagicha keltirishimiz mumkin:

Some-function (\$argument-1, \$argument-2); ixtiyoriy funksiya albatta qiymat qaytaradi. Bunga sabab esa, funksiya bajargan ishning muvaffaqiyatli yakunini aniqlash bo‘ladi.

Masalan, abs() funksiyasi sonning absolyut qiymatini qaytaradi. Buning uchun u minfiy son qabul qilib, musbat qiymat qaytaradi. Bu funksiyaga esa, quyidagi dastur misol sifatida keltirishimiz mumkin.

Misol-6.1

```
<html>
<head>
<title> abs() funksiyasini chaqirish</title>
</head>
<body>
<?php
    $num=321;
    $newnum=abs($num);
    print $newnum; // natija «321»

?>
</body>
</html>
```

Bu misolda \$ pit o‘zgaruvchisiga – 321 qiymatni o‘zlashtiramiz. Keyin esa, abs() funksiya argumentiga uzatib, shu funksiya olgan natija \$ new pitni o‘zgaruvchiga uzatiladi. Bu o‘zgaruvchidagi natijani print() orqali chop qilib olamiz. Aslida esa, dasturni yanada ixchamroq bo‘lishi uchun \$newnum o‘zgaruvchisidan foydalanmay print(abs(-321)) berishimiz ham mumkin.

Yangi termin Funksiya argumenti - bu siz funksiyaga murojaat qilganimizdagi beradigan qiymatlaringiz bo‘lib, ular qavs ichida beriladi. Funksiyani yaratishda argument nomi sharli ravishda beriladi, so‘ngra esa, bu argumentdan funksiya o‘zagida foydalanish mumkin.

Funksiyalarni yaratish. Funksiyalarni yaratish uchun function kalit soʻzidan foydalaniladi, hamda uning umumiy sxemasi quyidagicha boʻladi:

```
Function some_func (#argument_1, #argument_2);  
{  
    //funksiya oʻzagi  
}
```

Funksiya nomi function soʻzidan keyin beriladi va qavslar ichida argumentlar keltiriladi. Agar yaratilayotgan funksiya hech qanday argument talab qilmasa ham qavslar qoʻyilishi shart.

Keltirib oʻtgan fikrlarimizni yanada mustahkamlash uchun quyidagi misolni keltirib oʻtaylik:

Misol-6.2

```
<html>  
<head>  
<title> funksiyaning aniqlash</title>  
</head>  
<body>  
<?php  
function bighello()  
{  
    print "<h1>HELLO</h1>";  
}  
bighello()  
?>  
</body>  
</html>
```

Bu dastur brauzer oynasiga faqat «HELLO» yozuvini chiqarib beradi. Dasturda esa, funksiya nomini bighello() deb nomladik, hamda bu funksiya hech qanday argument talab qilmaydi. Bu misolda keltirib oʻtilgan funksiya barcha talablarga toʻgʻri keladi.

Endi esa, argument talab qiluvchi quyidagi misolni keltirib o‘taylik:

Misol-6.3

```
<html>
<head>
<title> Argument talab qiluvchi funksiyaga misol</title>
</head>
<body>
<?php
function printBR($txt);
    {
        print (“txt”<br>g‘n);
    }
printBR(“Bu qator”);
printBR(“Bu keyingi qator”);
printBR(“Bu bir keyingi qator”);
?>
</body>
</html>
```

PrintBR() funksiyasi argument kiritilishini talab qiladi. Argument tipi esa, matnli bo‘ladi va uning nomi \$txt deb atalgan. Endi biz bu funksiya yordamida ixtiyoriy yozuvni brauzerga chop qilishimiz mumkin.

Qiymat qaytaruvchi funksiyalar yaratish. Funksiya qiymat qaytarishi yoki return operatori yordamida ob’ekt qaytarishi mumkin. Bu operator funksiya ishini tugatadi va qaytarilishi lozim bo‘lgan qiymatni chaqirilgan bosh dasturga qaytaradi.

Bu fikrlarni mustahkamlash maqsadida 2 sonning yig‘indisini hisoblashga doir quyidagi misolni keltirib o‘taylik.

Misol-6.4. Qiymat qaytaruvchi funksiya

```
<html>
<head>
<title> Qiymat qaytaruvchi funksiya </title>
```

```

</head>
<body>
<?php
function addNums ($firstnum, $secondnum);
{
    resultq$firstnum+#secondnum);
}
print addNums(3,5); // natija 8 chiqadi
?>
</body>
</html>

```

Bu dasturning bajarilishi natijasida brauzerda 8 chop qilinadi. Chunki add nums() funksiyasining argumentlari 3 va 5 qiymatlarni qabul qiladi. Avvalroq aytib o'tganimizdek, dasturni yanada ixchamroq ko'rinishga keltirishimiz mumkin. U holda funksiyaning o'zagi quyidagi ko'rinishga ega bo'ladi.

```

{
    return ($firstnum+#secondnum);
}

```

return operatori qiymat, ob'ekt qaytarishi yoki hech rasa qaytarmasligi ham mumkin. Return operatori yordamida qiymat qaytarishning bir necha usullari mavjud bo'lib ular:

```

return4;                                -o'zgarmas;
return($a/$b);                          -ifoda yoki
return(function($argument))             -bo'lishi mumkin.

```

2.4. MySQL ma'lumotlar ombori haqida

Ma'lumotlar Bazasi va WWW

Ma'lumotlar bazasi o'ta tez rivojlangan hamda MySQL va mSQL o'zini ko'rsatgan soha, Internet uchun dasturlar yaratishdir. Internet uchun murakkab va ishonchli dasturlarga ehtiyoj oshgan sari ma'lumotlar bazasiga ehtiyoj ham oshib

bormoqda. Server ma'lumotlar bazasi Internetda ko'p funksiyalarni qo'llashi mumkin. Har qanday veb- sahifa ma'lumotlar bazasi tomonidan boshqarilishi mumkin.

Misol tariqasida o'z kattalogini WWW da e'lon qilmoqchi bo'lgan va Internet orqali buyurtmalar qabul qilmoqchi bo'lgan kattalog bo'yicha sotuvchini ko'raylik.

Agar kattalogni HTML-faylar shaklida e'lon qilinsa Yangi tovar qo'shilganda yoki narx o'zgarganda kimdir kattalogni tahrirlashi lozim bo'ladi Agar buning o'rniga kattalog ma'lumotlarini relyatsion ma'lumotlar bazasida saqlansa kattalogdagi o'zgarishlarni ma'lumotlar bazasidagi tovar yoki narx haqidagi ma'lumotlarni o'zgartirish yo'li Bilan real vaqt masshtabida e'lon qilish imkoniyati tug'iladi.

Bundan tashqari kattalogni mavjud buyurtmalarni qayta ishlash elektron tizimlari Bilan integratsiya qilish imkoniyati tug'iladi. Shunday qilib bunday veb-saytni boshqarish uchun ma'lumotlar bazasidan foydalanish sotuvchiga ham oluvchiga ham qulayliklar tug'diradi.

Shu tarzda veb- sahifa ma'lumotlar bazasi bilan bog'lanadi. Ma'lumotlar bazasi sizni veb-serveringizda yoki sizni serveringiz ma'lumot almashishi mumkin bo'lgan boshqa mashinada joylashgan bo'lishi mumkin.(yaxshi MBBT bunday vazifalarni taqsimlashni oson tashkil qila oladi). Siz o'zingizning veb-sahifangizga forma joylashtirasiz va foydalanuvchi uzatish kerak bo'lgan so'rov yoki ma'lumotni shu formaga kiritadi. Formani serverga uzatgandan so'ng, server siz yozgan dasturni ishga tushiradi va bu dastur foydalanuvchi uzatgan ma'lumotlarni ajratib oladi. Bu dasturlar ko'pincha CGI-stsenariylar yoki Java da server dasturlari shaklida yaratiladi, lekin dasturni HTML-sahifaga to'g'ridan to'g'ri joylashtirish ham mumkin.

Endi sizni dasturingiz foydalanuvchiga qanday ma'lumotlar kerak va u ma'lumotlar bazasiga nima kiritmoqchiligini biladi. Dastur ma'lumotlarni tanlash yoki o'zgartirish uchun SQL komanda yaratadi, ma'lumotlar bazasi bo'lsa qolganini bajaradi. Ma'lumotlar bazasidan olingan natijalarni sizni dasturingiz Yangi HTML-sahifa shakliga keltirib qaytadan foydalanuvchiga yuboradi.

mSQL tarixi

To 1994 yilgacha SQL qo‘llovchi RMBBT sotib olish uchun ancha ko‘p pul ketkazishga to‘g‘ri kelar edi. Bozorda Oracle, Sybase va Informix hokimlik qilar edi. Bu ma‘lumotlar bazasini boshqarish tizimlari murakkab bog‘lanishlarga ega bo‘lgan katta hajmdagi ma‘lumotlarni qayta ishlash uchun mo‘ljallangan edi. Ular katta quvvatga va imkoniyatlarga ega bo‘lib, katta hisoblash resurslarini talab qilar edilar va narxi qimmat edilar. U paytda \$2000 ga 200-MHz Pentium li server sotib olish mumkin emas edi. Bunday MBBT uchun kerakli resurslar o‘ng minglab dollar turar edi.

Katta korporatsiyalar va yirik universitetlar uchun bunday serverlar komplektlari va MBBT lar uchun bir necha million dollar sarflash muammo tug‘dirmas edi.

Kichik tashkilotlar va xususiy foydalanuvchilar ojiz kichik Amaliy dasturlardan foydalanishga majbur edilar. Bir nechta arzon klient/ server arxitekturali MBBT lar o‘sha paytda mavjud edi, lekin ularning hech biri so‘rovlar tili sifatida SQL dan foydalanmas edi. Eng ko‘zga ko‘ringanlaridan biri Ingres kommertsial ma‘lumotlar bazasi bilan bitta ajdodga ega bo‘lgan Postgres edi. Lekin baxtga qarshi kommertsial analoglari kabi resurslarni talab qilardi va SQL dan so‘rovlar tili sifatida foydalanish imkoniyatini bermas edi. O‘sha paytda Postgresda QUEL tilining ko‘rinishi bo‘lgan PostQUEL tilidan foydalanardi.

Devid Xyuz

David Xyuz (David Hughes) (yana Bamby sifatida ma‘lum) Avstraliyada Bond Universitetida yozgan dissertatsiya bir qismi monitoring tizimini yaratish va tizimlar guruxini bir yoki bir necha joydan boshqarishga bag‘ishlangan edi. Loyiha Minerva Network Management System deb nomlangan edi. Minerva asosiy elementi tarmoqdagi hamma kompyuterlar haqidagi ma‘lumotlarni saqlovchi ma‘lumotlar bazasi edi. Universitet talabasi bo‘lgani va katta kommertsial ma‘lumotlar bazalari ishlaydigan serverlarga murojaat qilish imkoniyati yo‘q

bo'lgani uchun, Xyuz uning talablariga Postgres – javob beradi degan qarorga keldi.

Uning hamkasblari SQL tilidan Minerva uchun standart so'rovlar tili sifatida foydalanishni taklif qilishdi. SQL ga asoslangan xolda Minerva dunyoning SQL ni qo'llovchi MBBT mavjud ixtiyoriy nuqtasida qo'llanishi mumkin edi. Boshqacha qilib aytganda Postgres foydalanuvchilari bilan chegaralab qo'ygan PostQUEL ga nisbatan SQL Minerva uchun kengroq foydalanuvchilar bilan ishlashga imkon berar edi. Oxiri kelib hatto Postgres ham bugun SQL no qo'llaydi.

Bir tomondan SQL standartidan foydalanish istagi va boshqa tomondan SQLni qo'llovchi ma'lumotlar bazasiga murojaat qilish imkoniyati yo'qligi, Xyuzni qiyin ahvolga solib qo'ydi. Agar Minervada SQLga asoslangan so'rovlar tilidan foydalanilsa, mos ishlash mexanizmiga ega MBBT topib bo'lmaydi. Qimmat RMBBT sotib olish imkoniyatiga ega bo'lmagan Xyuz masalaning ajoyib echimini topdi: SQL so'rovlarni PostQUEL so'rovlariga translyatsiya qiluvchi dastur yaratish. Bu dastur Minervaga uzatilgan SQL so'rovlarni ilib olishi, PostQUELga aylantirishi va natijani Postgresga uzatishi kerak edi. Xyuz shunday dastur yaratdi va uni miniSQL yoki mSQL deb atadi.

PostQUEL translyatoridan RMBBT ga

Bir necha davr mobaynida bu konfiguratsiya Xyuzni qanoatlantirar edi. Minerva uchun agar SQLni tushunsa qanday MBBT dan foydalanishning farqi yo'q edi va u Postgres SQLni tushunadi deb hisoblar edi, chunki o'rtada PostQUELga translyatsiya qiluvchi mSQL joylashgan edi. Baxtga qarshi Minerva o'sishi bilan uning ishi qiyinlashib bordi. Aniq bo'ldiki na Postgres na boshqa katta RMBBT Minerva uchun kerak bo'lgan chekli resurslar asosida kam sonli imkoniyatlarni qo'llay olmas edi. Masalan, Minerva uchun bir vaqtning o'zida bir necha ma'lumotlar bazasiga ulanish talab qilinar edi. Buning uchun Postgres bir vaqtning o'zida ma'lumotlar bazasi serveri bir necha nusxasini ishga tushirishni talab qilar edi. Bundan tashqari bir necha potentsial loyiha qatnashchilari unda

qatnasha olmas edilar, chunki Postgres ularning tizimlarini qo'llamas edi, ular bo'lsa SQLga asoslangan qimmat RMBBT sotib olishga imkonlari yo'q edi.

Bu muammolarga duch kelgandan so'ng Postgresga munosabatini o'zgartirdi. O'zining kattaligi va murakkabligi bilan Minerva talablaridan yuqori edi. Minerva tomonidan generatsiya qilinadigan so'rovlar asosan INSERT, DELETE va SELECT sodda operatorlaridan iborat edi. Postgres da mavjud va unumdorlikni kamaytiruvchi qolgan hamma imkoniyatlar Minerva uchun kerak emas edi.

Xyuzda SQLga translyatsiyani amalga oshiruvchi mSQL mavjud edi. Unga talablariga javob beruvchi ma'lumotlar bazasi serverini yaratish uchun ma'lumotlar ombori va ma'lumotlarni ajratib olish imkoniyatini qo'shish qolgan ediyu Bu evolyutsiya bugungi kunda mavjud mSQL ga olib keldi.

MySQL tarixi

MySQL ni faqat mSQL kamchiliklariga javob sifatida qarash noto'g'ridir. Uning ixtirochisi Maykl Videnius (yana Monty sifatida ma'lum) shved kompaniyasi TsX xodimi ma'lumotlar bazasi bilan 1979 yildan beri ishlaydi. Yaqin paytgacha Videnius TsX da faqat dasturchi edi. 1979 yilda firma ichida foydalanish uchun UNIREG nomli ma'lumotlar bazasini boshqarish vositasini yaratdi. 1979 yildan so'ng UNIREG bir necha tillarda yozildi va katta ma'lumotlar bazalarini qo'llash uchun kengaytirildi.

Bitta dastur bajarilayotgan har bir jarayon bu dastur nusxasi deyiladi, chunki xuddi o'zgaruvchi nusxasi kabi xotiradan joy oladi.

1994 yilda TsX WWW uchun Amaliy dasturlar yarata boshladi va bu loyihani qo'llashda UNIREG dan foydalandi. Baxtga qarshi, UNIREG katta harajatlar talab qilgani uchun, undan veb-sahifalarni dinamik generatsiya qilish uchun muvaffaqiyatli foydalanib bo'lmadi. Va TsX shundan so'ng SQL va mSQL ga murojaat qildi Lekin o'sha paytda mSQL faqat 1.x relizlari shaklida mavjud edi. Yuqorida aytganimizdek mSQL 1.x versiyalari hech qanday indekslarni qo'llamas edi va shuning uchun UNIREG dan unumdorligi past edi.

Videnius mSQL avtorlari s Xyuz bilan bog'landi va mSQL ni UNIREG dagi V+ ISAM qayta ishlovchisiga ulash fikri Bilan qiziqirtirmoqchi bo'ldi. Lekin Xyuz shu

paytga keldib mSQL 2 yaratish yo'lida anchaga ketgan va indekslar Bilan ishlash vositalarini yaratgan edi. TsX o'z talablariga ko'proq mos keluvchi ma'lumotlar bazalari serverini yaratishga qaror qildi.

TsX xodimlari Yangidan velosiped ixtiro qilib o'tirmadilar. Ular UNIREG ni asos qilib oldilar va soni oshib borayotgan o'zga dasturchilar utilitalaridan foydalandilar. O'z tizimlari uchun boshida mSQL uchun yaratilgan API bilan deyarli ustma-ust tushuvchi API yaratdilar. Natijada Yangi kengroq imkoniyatga ega TsX ma'lumotlar bazasiga o'tmoqchi bo'lgan mSQL foydalanuvchisi o'z kodiga juda kam o'zgartirish kiritishi talab qilinardi. Shu bilan birga Yangi ma'lumotlar bazasi kodi to'la original edi.

1995 yil may oyiga kelib TsX kompaniya ichki talablarini qanoatlantiruvchi ma'lumotlar bazasi, - MySQL 1.0 ga ega edi. Firma biznes-partneri Detron HB dan David Aksmark (David Axmark) TsX ga o'z serverini Internetda ko'rsatishni taklif qildi. Serverni Internetda ko'rsatishdan maqsad - birinchi bo'lib Aladdin Peter Deych (Aladdin Peter Deutsch) qo'llagan biznes modelldan foydalanishdir. Natijada MySQLni mSQL ga nisbatan «tekinroq» qiluvchi o'ta moslashuvchan avtorlik huquqlari olindi.

Nomiga kelganda Videnius bu haqida shunday deydi: «Xozirgacha noma'lum MySQL nomi qaerdan kelib chiqqani. TsX da asosiy kattalog, hamda bibliotekalar va utilitalar ko'p qismi bir necha o'n yildan beri «mu» prefiksiga ega. Shu bilan birga mening qizim (bir necha yilga kichik) ismi ham May (My). Shuning uchun haligacha sir, bu ikki manbaning qaysi biri MySQL nomini bergan».

MySQL ni Internetda e'lon qilingandan beri u ko'pgina UNIX-tizimlarga, Win32 i OS/2 ga ko'chirildi. TsX fikricha, MySQL ni 500 000 ga yaqin serverlar ishlatadi.

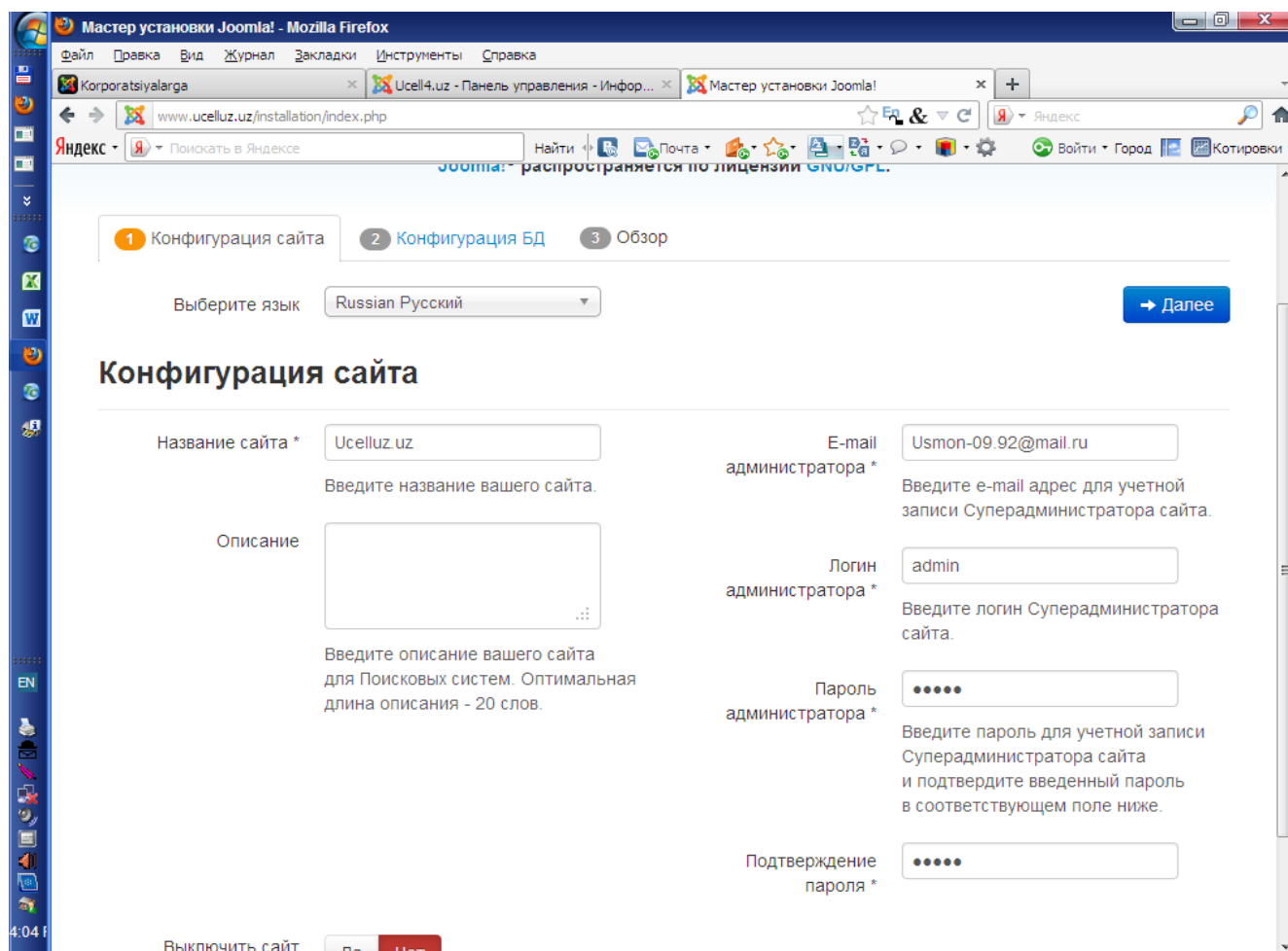
III. BOB. Ucell KOMPANIYASI DILLERINING ISHLASH MEXANIZMIDA Joomla DASTURINING QO‘LLANILISHI.

3.1. Joomla dasturida web – saytni yaratilish jarayoni

Bu bobda Ucell kompaniyasi dillerining ishlash mexanizmda Joomla dasturining qo‘llanilishini ko‘rib chiqamiz. Buning uchun avvalo Joomla dasturi orqali abonentlarga xizmat ko‘rsatish tizimining internet xizmatini yaratib olishimiz kerak. Ya’ni Ucell.uz saytini yaratamiz.

Buning uchun avvalo denwer dasturini o‘rnatib olamiz va Joomla dasturini o‘rnatishni boshlaymiz. Joomla dasturining o‘rnatilishining boshlanishi quyidagidek bo‘ladi:

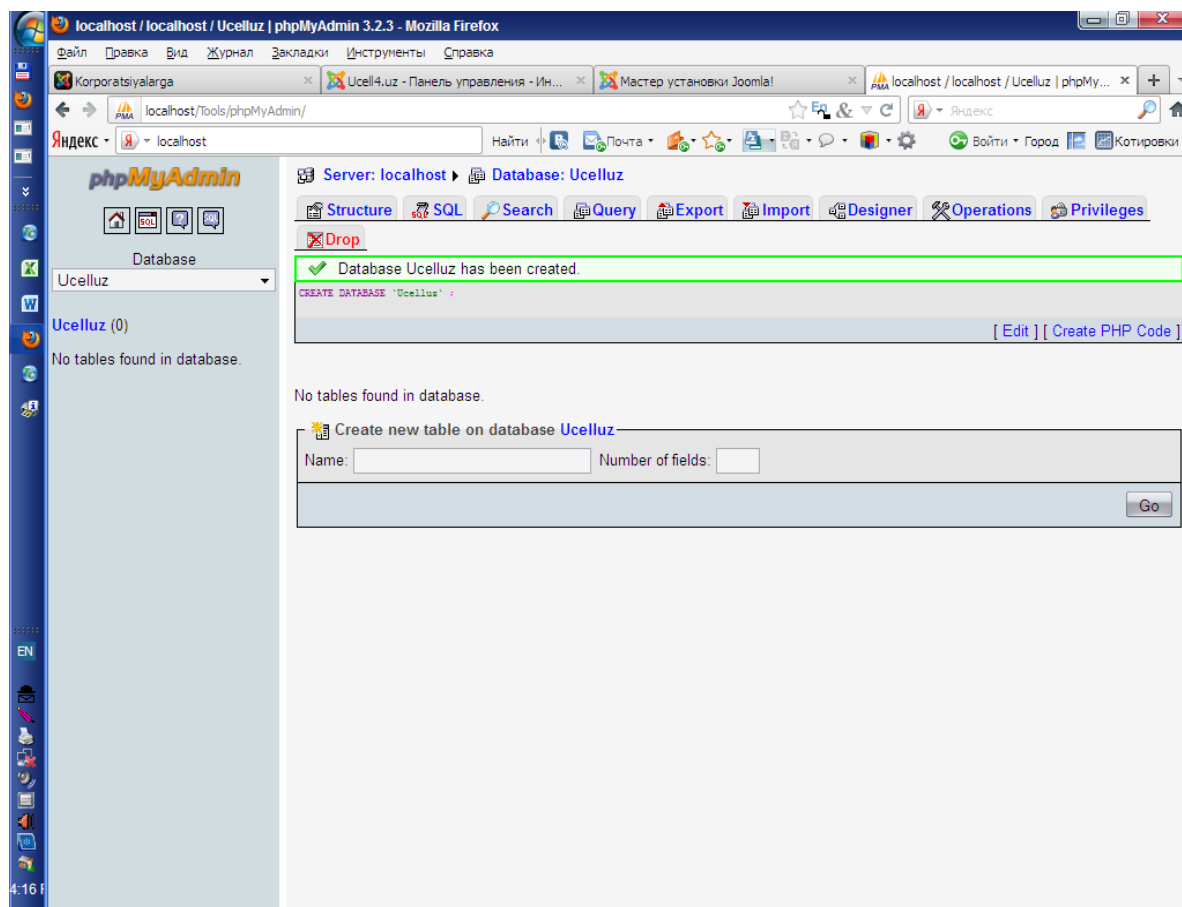
c:\WebServers\home\ adresiga kiramiz va u joyga Ucell.uz papkasini yaratamiz va uning ichiga www papkasini joylaymiz, keyin uning ichiga kirib Joomla dasturining barcha fayllarini ko‘chirib o‘tqazamiz. Endi birinchi bosqichni boshlaymiz:



3.1.1-rasm. Joomla dasturida web-sayt yaratishning 1-bosqichi.

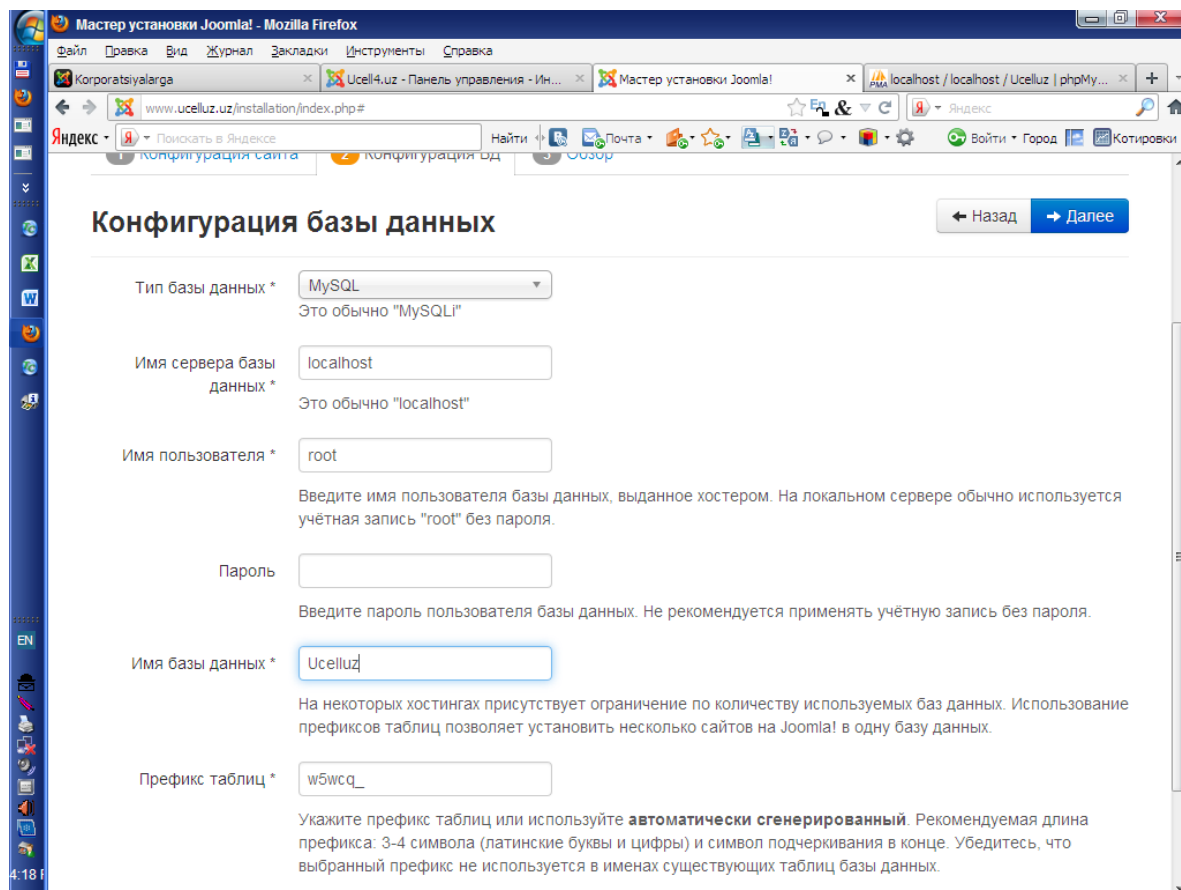
Endi далее tugmasini bosib ikkinchi bosqichga o'tamiz:

Bu bosqichni amalga oshirish uchun oldin phpmyadmin orqali database ochib olamiz:



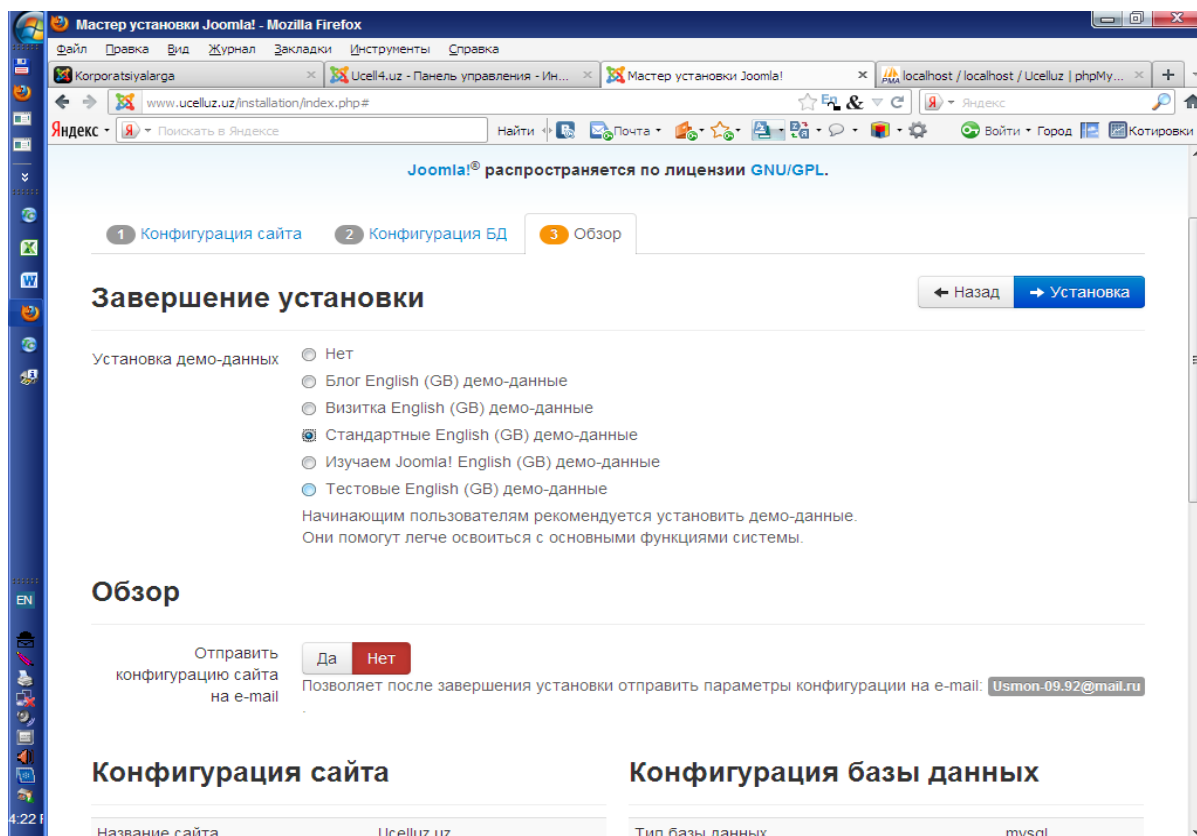
3.1.2-rasm. Phpmyadmin orqali sayt uchun baza yaratib olish.

Va ikkinchi bosqichni davom ettiramiz:



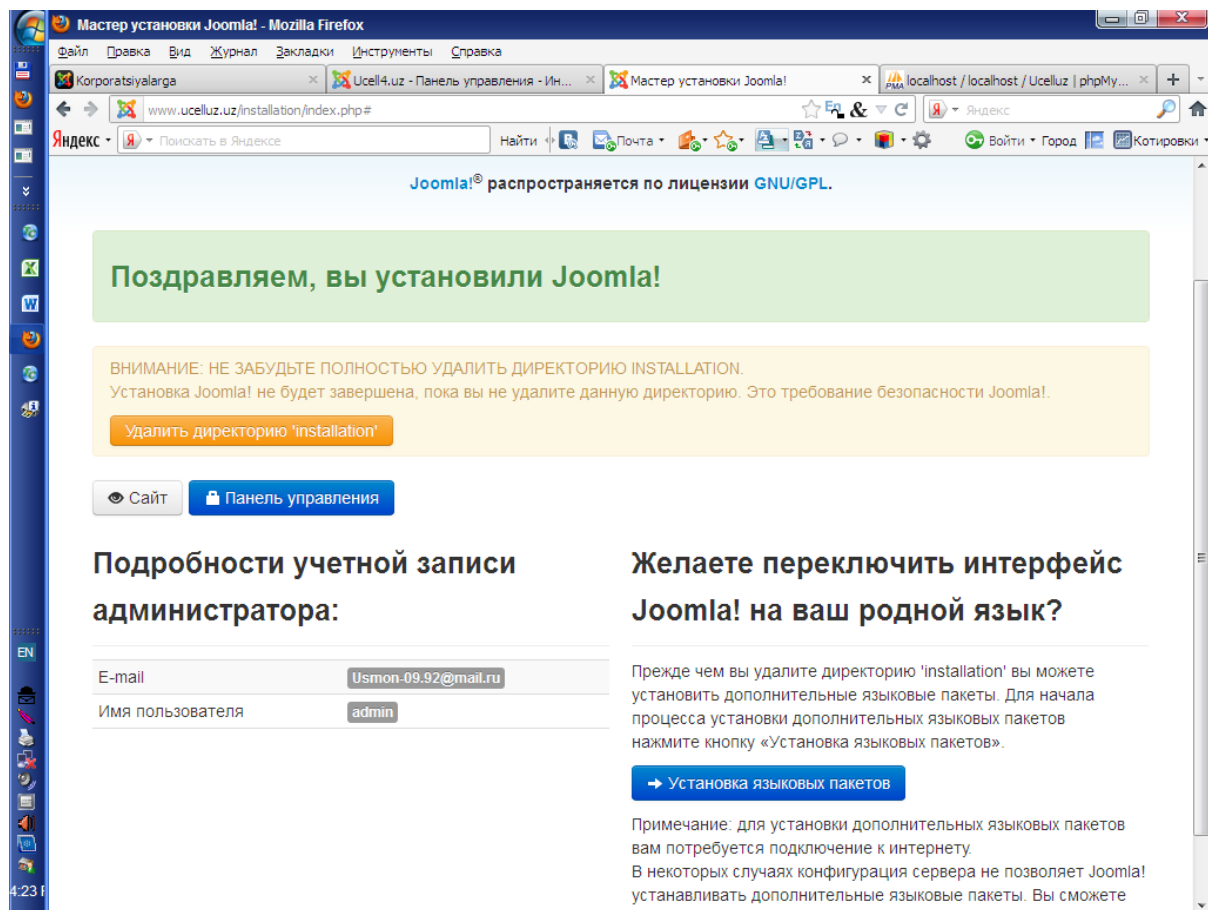
3.1.3-rasm. Joomla dasturida saytni phpmyadmin orqali yaratilgan bazaga joylash.

Va далее ni bosib uchinchi bosqichga o'tamiz:



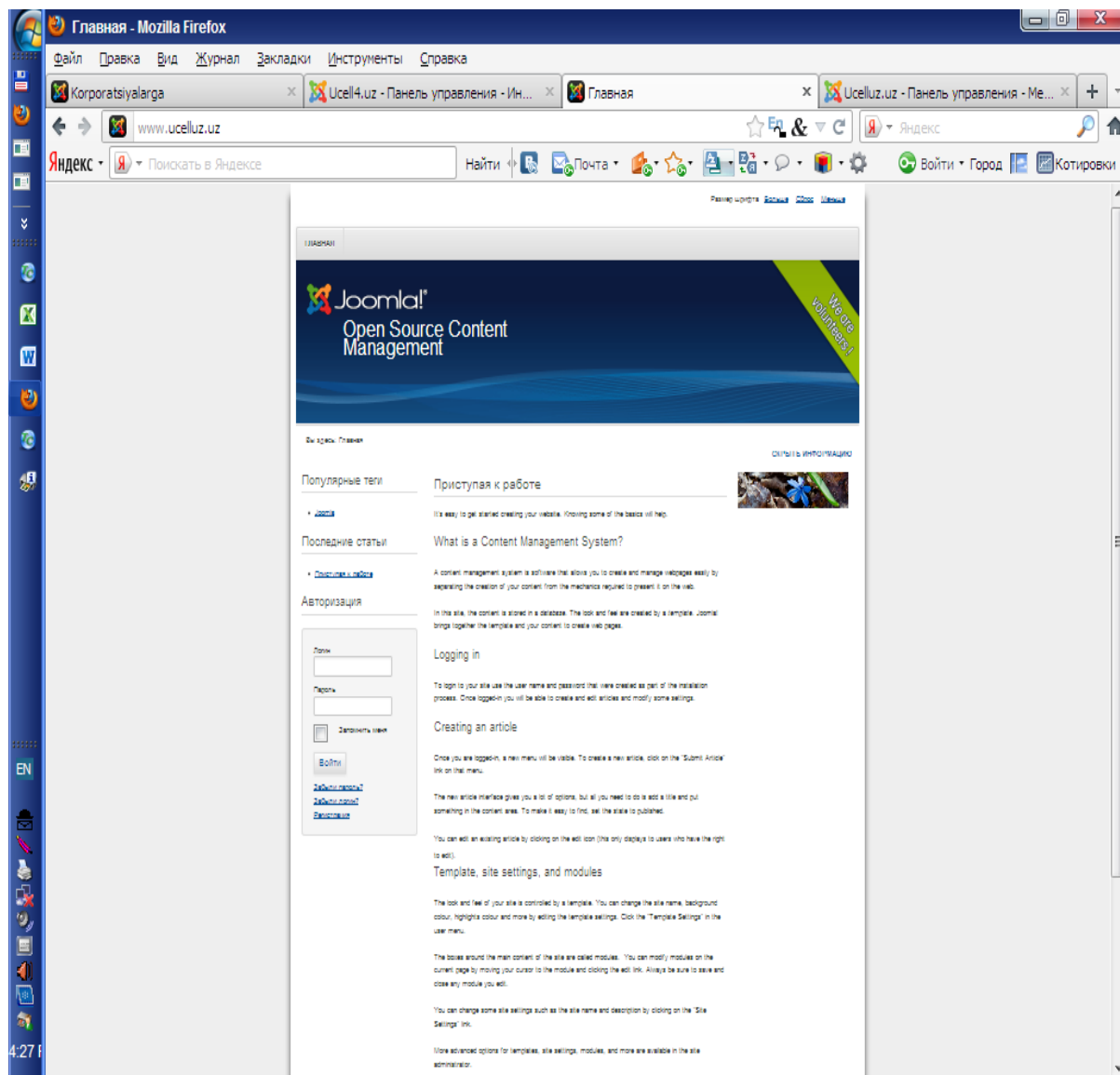
3.1.4-rasm. Joomla dasturining 3-bosqichi yuklash(установка)ni yakunlash.

Va установка tugmasini bosib quyidagi ko'rinishga o'tamiz:



3.1.5-rasm. Installation papkasi o‘chiriladi va joomla dasturi o‘rnatilishi tugaydi.

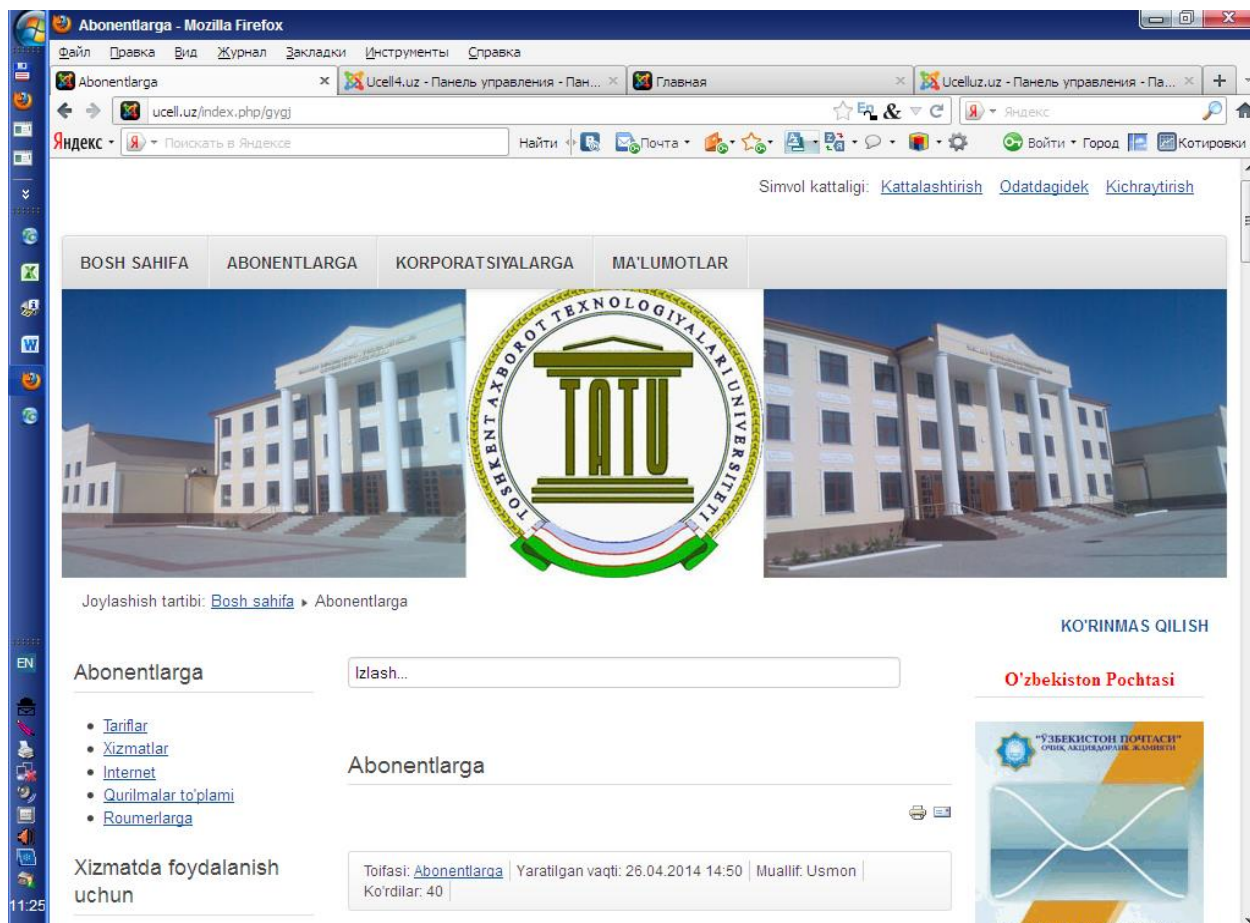
Va bu joydagi buyruqni bajarish orqali quyidagi saytga ega bo‘lamiz va uni istagan ko‘rinishda o‘zgartirishimiz mumkun:



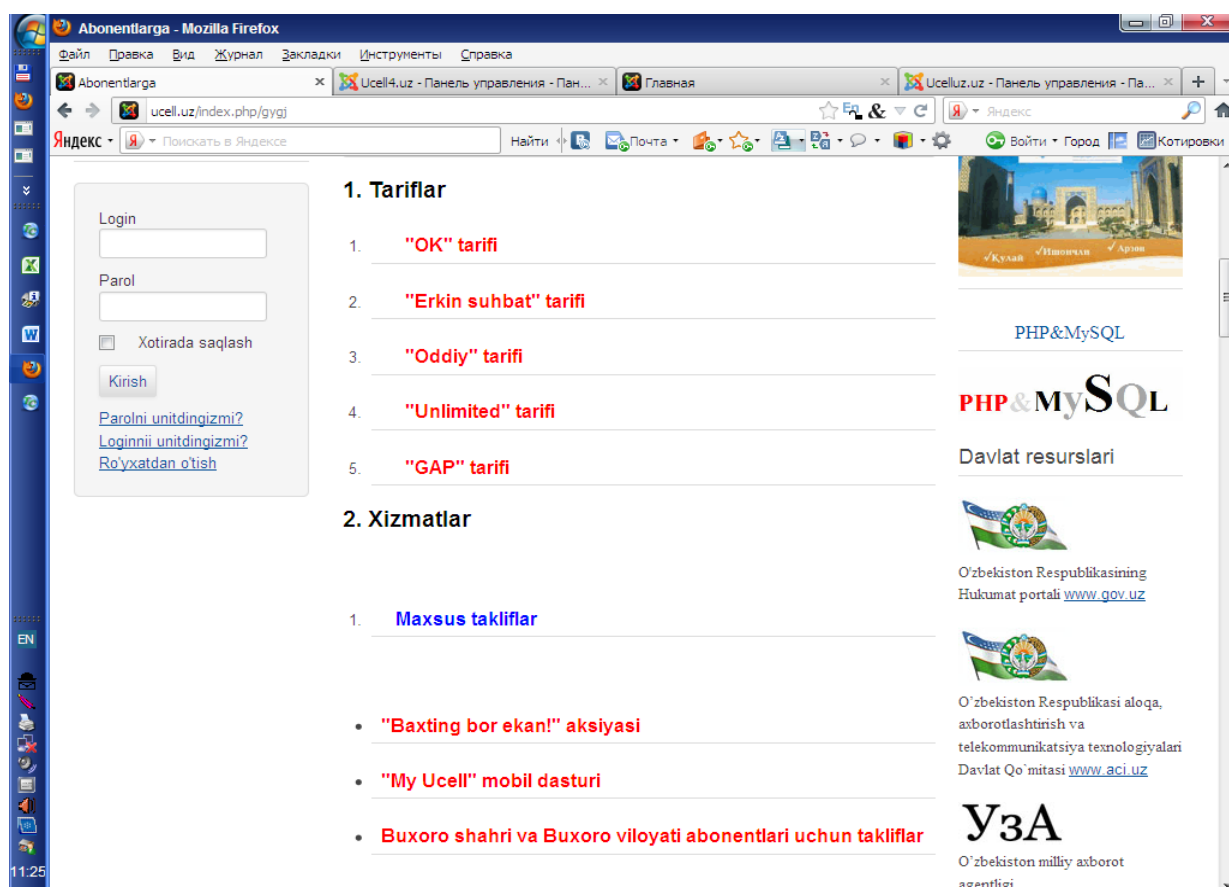
3.1.6-rasm. Yaratilgan yangi sahifa ko‘rinishi.

3.2. Abonentlar uchun ajratilgan sahifa.

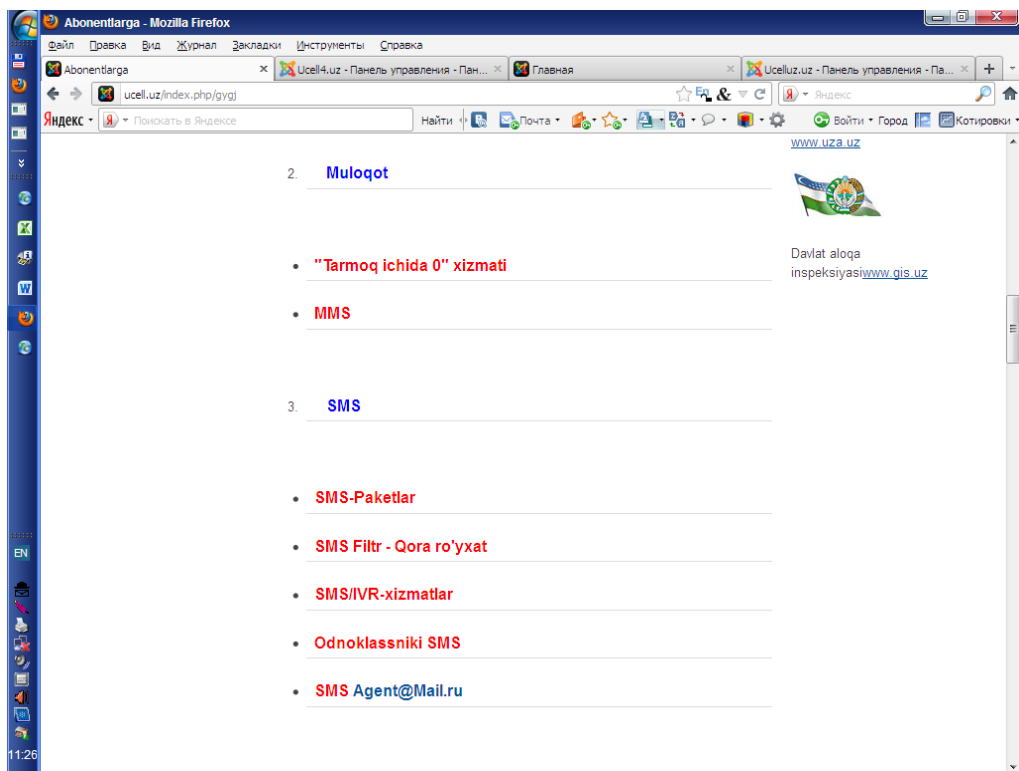
Bu bo‘limda biz abonentlarga xizmat ko‘rsatishdagi web – tizimning abonentlar sahifasini ko‘rishimiz mumkin:



1



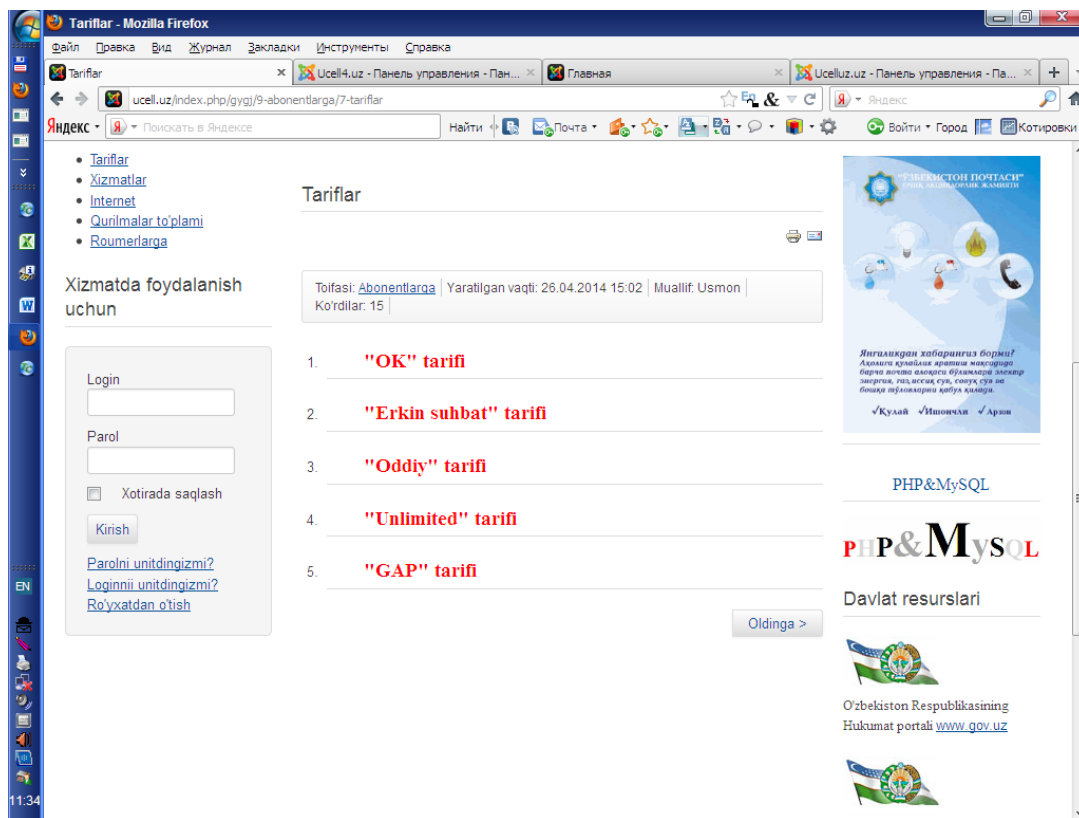
2



3

3.2.1-rasm. 1, 2, 3-larda Abonentlar uchun ajratilgan sahifa ko'rsatilgan.

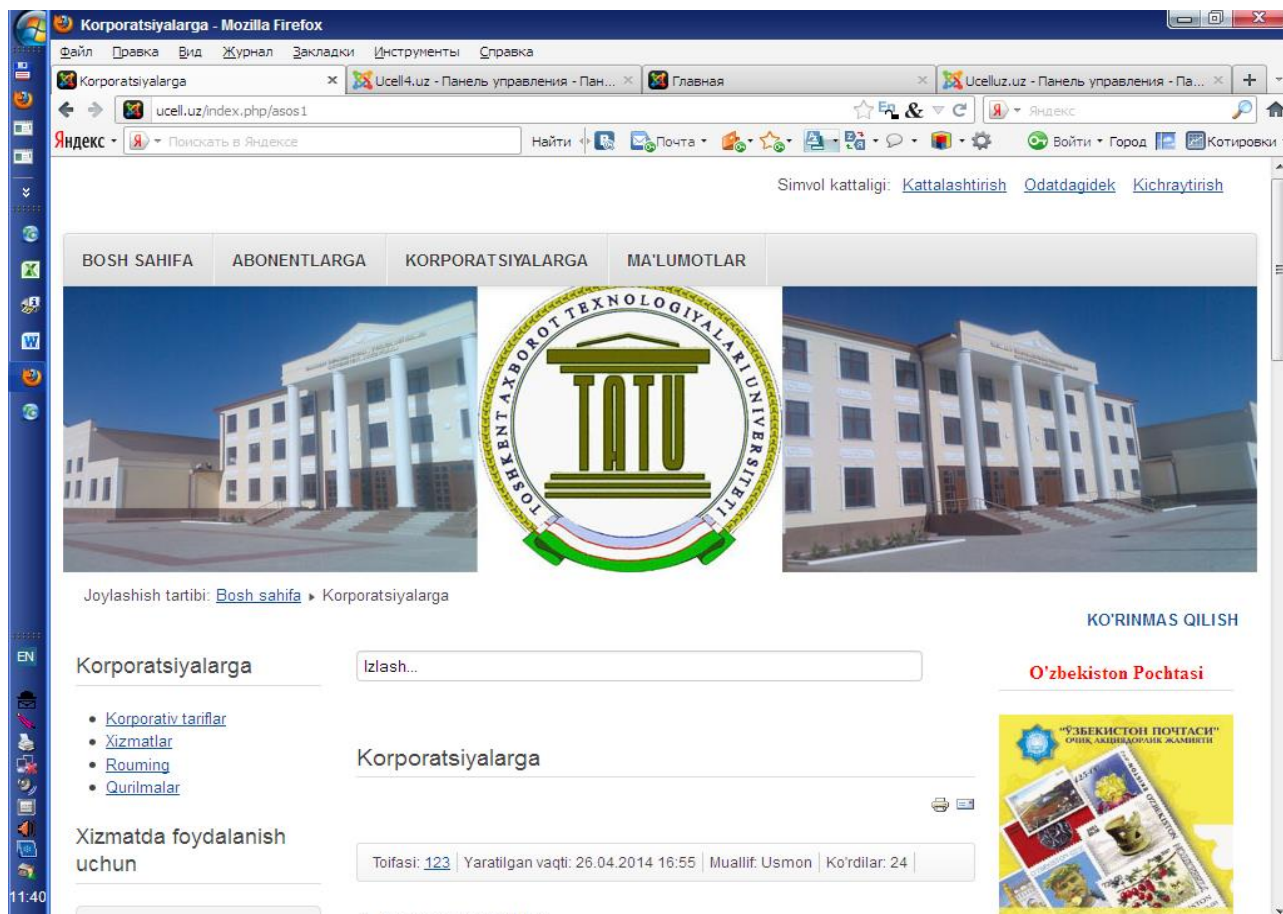
Bu sahifada korinib turibdiki uning ichida "Tariflar", "Xizmatlar", "Internet", "Qurilmalar to'plami", "Roumerlarga" kabi takliflar mavjud. Shu bois ularni bir sahifada ko'rib abonent adashib ketishi mumkin. Buning oldini olish uchun unda har bir taklifga alohida kirish imkoniyati mavjud. Tariflarni alohida quyidagi ko'rinishda ko'rish mumkin:



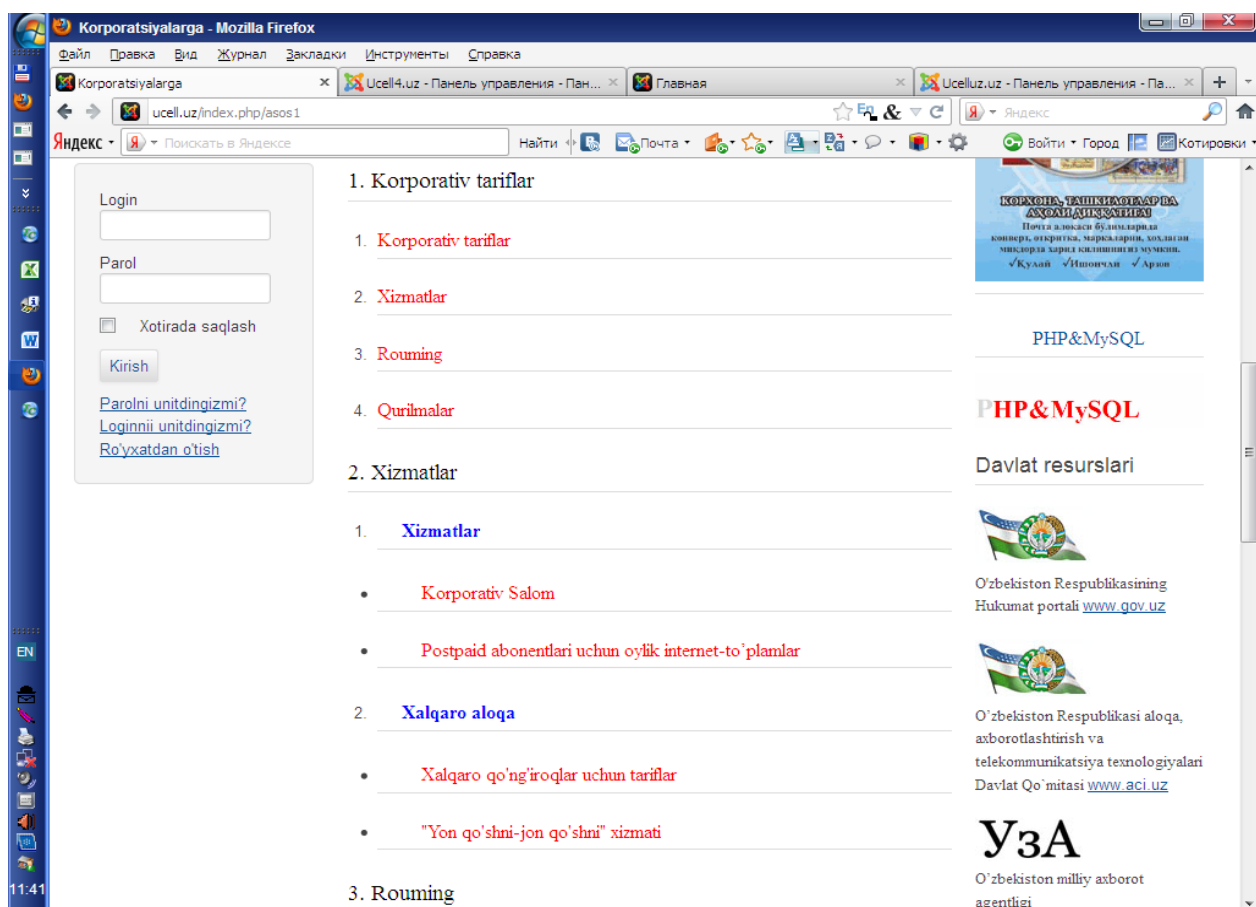
3.2.2-rasm. Abonentlar bo'limidagi tariflarning alohida ko'rinishi.

3.3. Korporatsiyalar uchun.

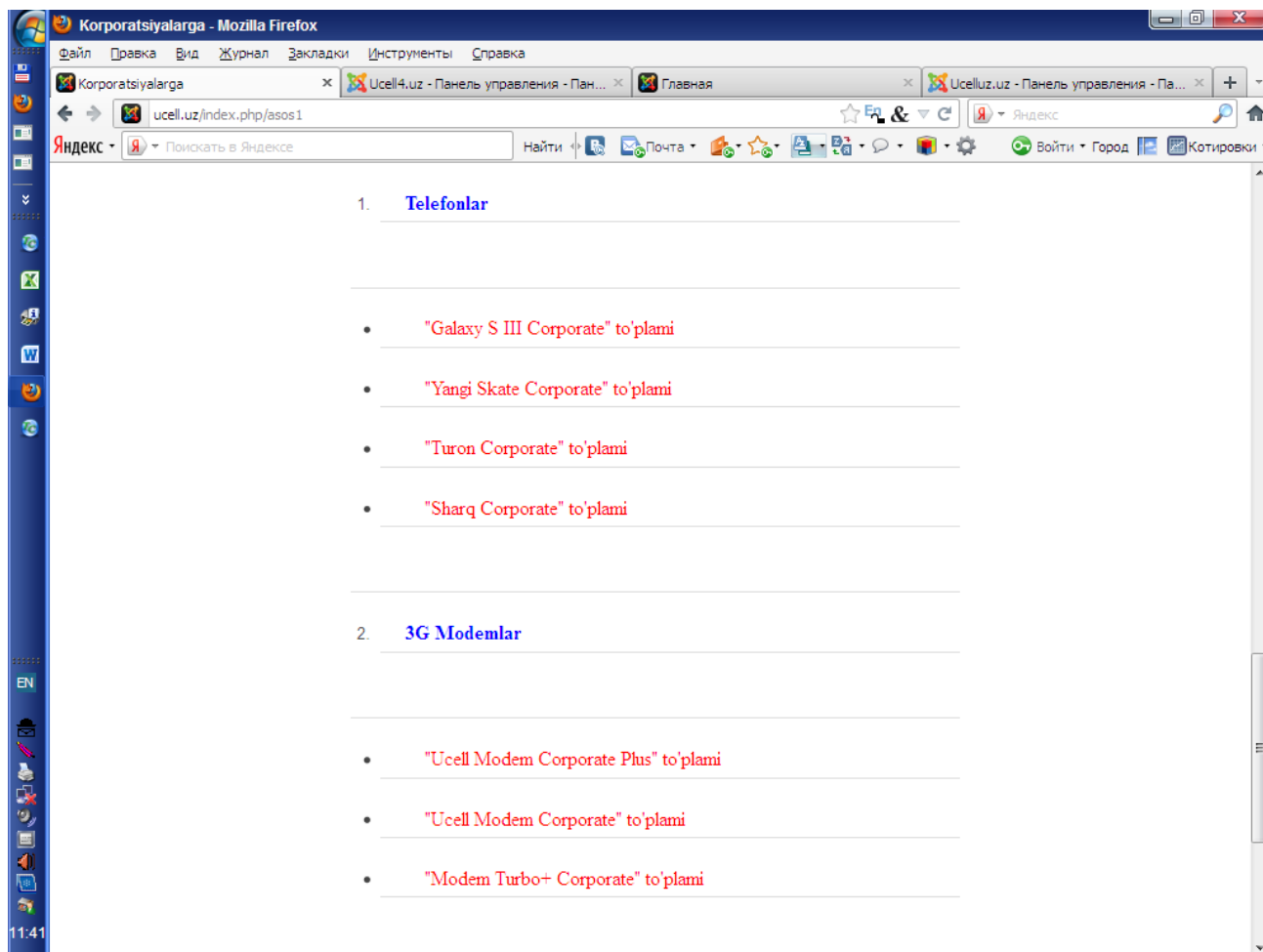
Bu bo'limda ham abonentlar bo'limi kabi usuldan foydalanilgan bo'lib quyida keltirilgan:



1



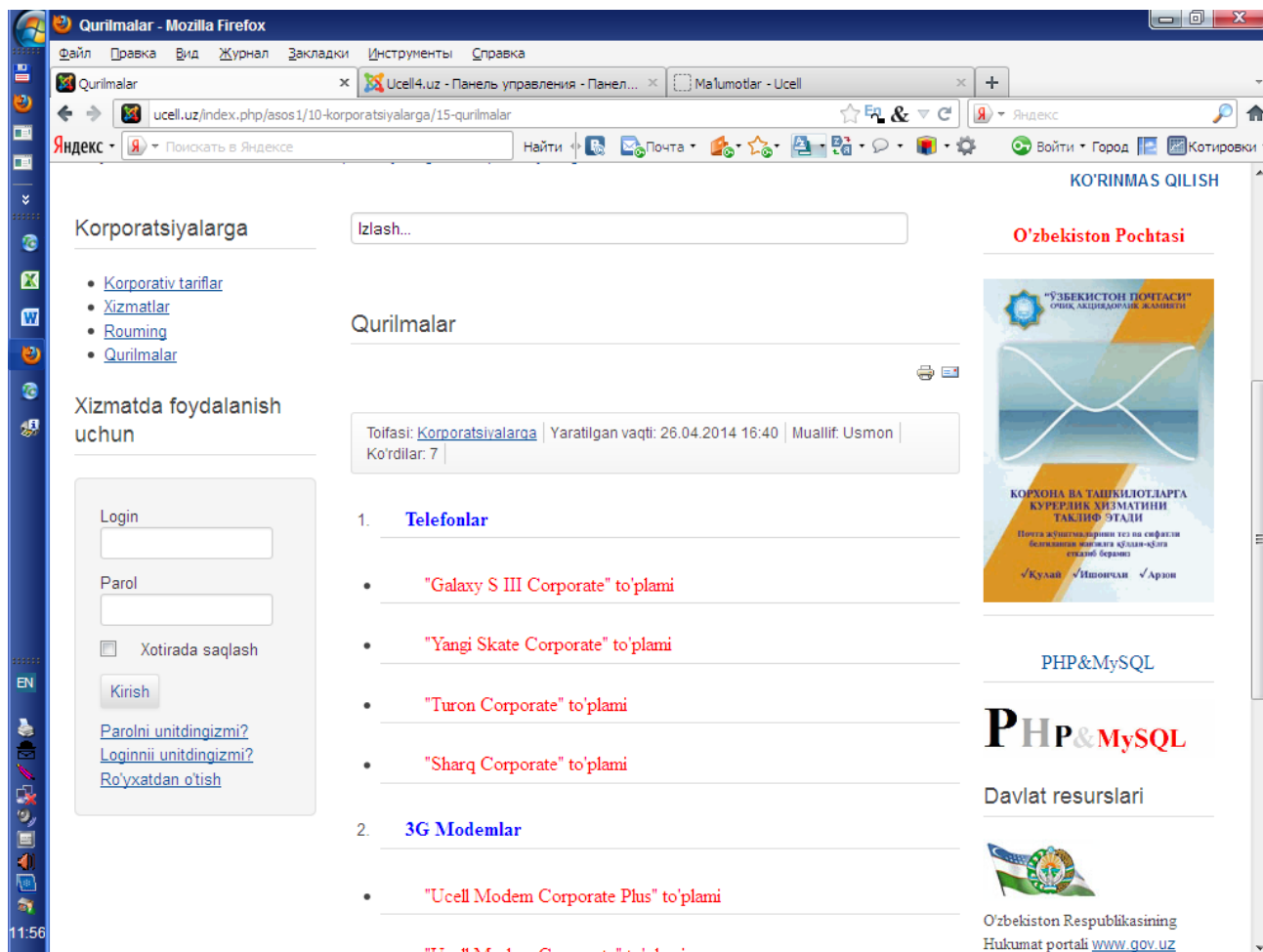
2



3

3.3.1-rasm. 1,2,3-larda korporatsiyalarga ajratilgan sahifaning to'liq ko'rinishi.

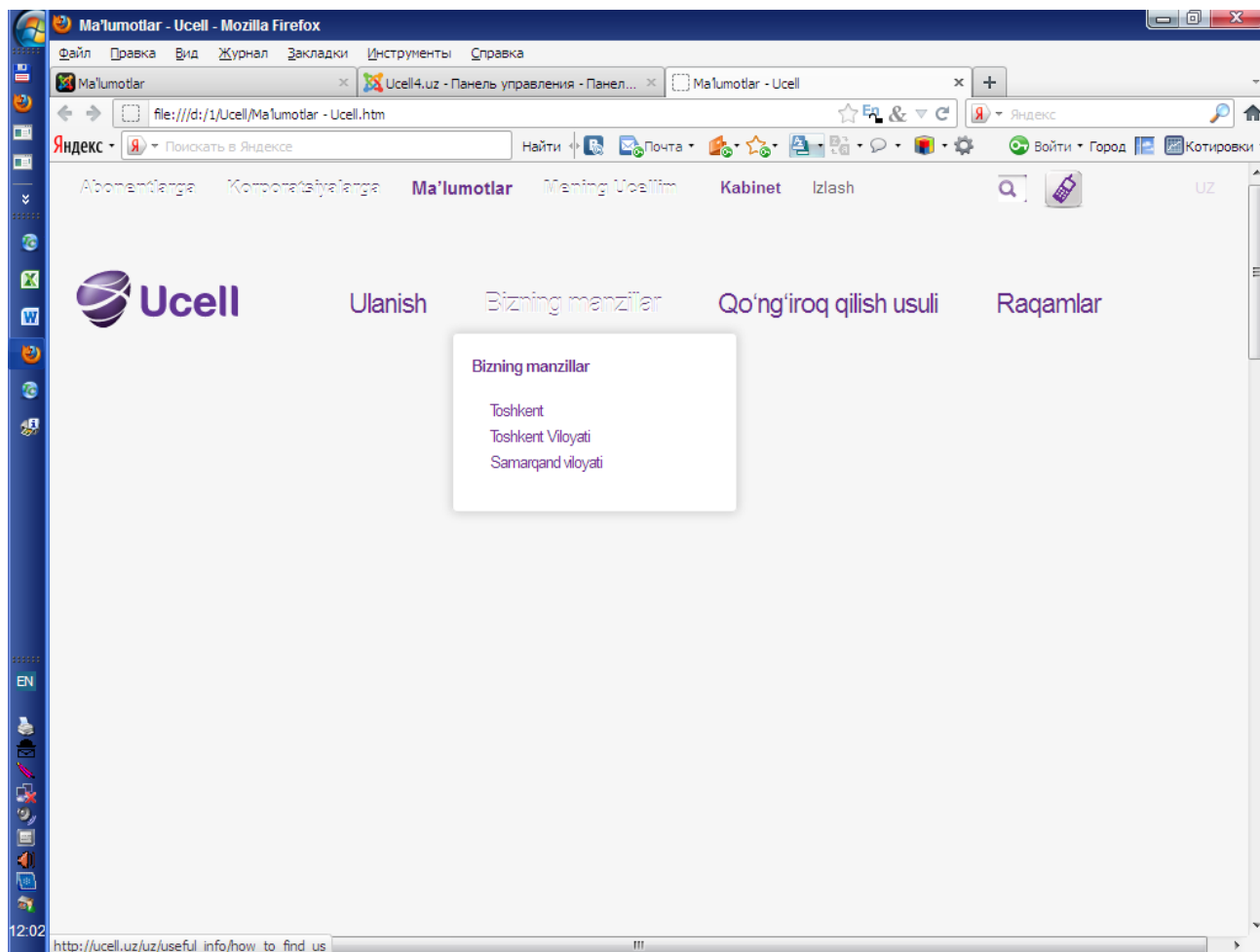
Bunda ham misol qilib oxirgasi bo'lgan qurilmalar taklifini ko'rsatish mumkin:



3.3.2-rasm. Korporatsiyalar bo'limidagi qurilmalarning alohida sahifasi.

3.4. Ucell haqida ma'lumot.

Bu bo'lim tog'ridan - to'g'ri Ucell.uz saytiga ulangan bo'lib unda quyidagi ko'rinish hosil boladi:



3.4.1-rasm. Ucell haqida ma'lumotlar olish sahifasi. Bu bo'limda to'g'ridan-to'g'ri internetdan Ucell kompaniyasining haqiqiy saytiga ulanib ketadi.

XULOSA.

Hozirgi zamonda dasturlarning ishlab chiqilishida uning ishonchliligiga juda kata e'tibor qaratilmoqda. Ayniqsa abonentlarga xizmat ko'rsatish uchun mo'ljallangan dasturlarga. Buning asosiy sabablaridan biri bu hozirgi zamon yoshlarining bilim jihatdan ancha ustunligi va zamon talablarining kuchayishi bo'lib hisoblanadi. Dastur ishlab chiqishda eng kata e'tibor uning foydalanuvchilar uchun qay darajada e'tibor qozonishi hisoblanadi. Chunki ozgina bir xatolik kata mojarolarga yoki bo'lmasa kata sarf xarajatlarga olib kelishi mumkin. Shuning uchun ham har bir ishlanmalar uchun uning to'g'ri va aniq ishlashi eng dolzarb masala hisoblanadi.

Ko'rib chiqilayotga bitiruv malakaviy ishimda "Abonentlarga xizmat ko'rsatish markazi" axborot tizimini yaratish (Internet xizmati) keltirilgan.

Men shu bakalavrlik bitiruv malakaviy ishimda abonentlarga xizmat ko'rsatish uchun zaruriy shart – sharoitlar va uning qanchalik zarurligini tushundim va uning internet orqali har bir abonent o'z sevgan kompaniyasi saytidan foydalanish imkoniyatiga ega bo'lgan saytni ishlab chiqdim. Men bu bitiruv malakaviy ishim uchun yaratilgan saytni eng zamonaviy osonlashtirilgan Joomla dasturidan foydalangan holda ishlab chiqdim. Va bu malakaviy ishimni har bir shu web – dasturlashni o'rganishni xohlovchilar uchun undan foydalanish imkoniyatuga ega qilib, web – dasturlash sohasidagi deyarli barcha tillar haqida boshlang'ich va yuqori darajali ma'lumotlarni berib o'tdim.

Xulosa o'rnida shuni aytishim mumkinki, ushbu yo'nalishdagi ilmiy ishlarni rivojlantirish va ular asosida abonentlarga xizmat ko'rsatish markazlarining takomillashishi bilan bog'liq bo'lgan turli masalalarni hal etishda qatnashish yoki ularga ko'maklashish qo'limdan kelsa, men o'zimni baxtli hisoblayman. Zero, mamlakatimiz fan-texnikasini rivojlantirish va farovon turmush tarzini yaratishga ko'maklashish, Respublikamizning yoshlari qatori mening ham sharaflı burchimdir.

FOYDALANILGAN ADABIYOTLAR.

1. Yuldashev U.Yu., Rahmatullaeva. Sh.K. Internet asoslari: O'quv qo'llanma –Toshkent.2001 Y
2. S.S.G'ulomovning umumiy tahriri ostida. Axborotlar tizimlari va texnologiyalari. T."Shark". 2000 y.
3. R.A.Abdullaev, V.I.Abdukarimov, N.A.Ibrogimov Iqtisodiy axborotni ishlashning avtomatlashgan tizimlari. Toshkent. "Ukituvchi", 1993 y
4. А. Гончарев. HTML. Самоучител. – «Питер», Санкт-Петербург, 2001 г.
5. PHP: Настольная книга программиста. Александр Мазуркевич, Дмитрий Еловой. Москва ООО «Новое здание» 2004.
6. Internetda dasturlash. Muommoli ma'ruzalar matni. O.Jakbarov, T.Jo'raev. Namangan – 2004.
7. SQL va ma'lumotlar ba'zasining keying dasturlash. Sh.Nazirov, R.Qobulov. Toshkent - 2006.
8. Professional PHP programming. Jesus Castagnetto, Harish Rawat, Sascha Schumann, Chris Scollo, Deepak Veliath. © 1999 Wrox Press.
9. G'ulomov S.S. va boshqalar. "Axborot texnologiyalari va tizimlari". Toshkent, O'zbekiston, 2000 y.
10. Работа в интернет. Ташкент, 2001.
11. S.S.G'ulomov, A.T.Shermammedov, B.A.Begalov Iqtisodiy informatika. "Uzbekiston" nashriyoti, "Hilol nuri" shu'ba korxonasi. 1999 y
12. Викрам Васвани. Полный справочник по MySQL. MySQL: The Complete Reference. Издательство: Вилямс, 2006 г., 528 стр.
13. Люк Веллинг, Лора Томсон MySQL. Учебное пособие MySQL Тьюриал Перевод:Мягкая обложка Издательство: Вилямс, 2005 г., 304 стр.
14. Пол Дюбуа MySQL MySQL Серия: Ландмарк Другие издания: Твердый переплет Аналоги: Твердый переплет Издательство: Вилямс, 2004 г., 1056 стр.
15. www.intuit.ru
16. www.ZiyoNet.uz
17. <http://ad.cctpu.edu.ru>
18. www.Joomla.ru

ILOVA

Index.

```
<?php
/**
 * @package Joomla.Site
 * @subpackage Templates.beez3
 * @copyright Copyright (C) 2005 - 2014 Open Source Matters, Inc. All rights
reserved.
 * @license GNU General Public License version 2 or later; see LICENSE.txt
 */

// No direct access.
defined('_JEXEC') or die;

JLoader::import('joomla.filesystem.file');

// Check modules
$showRightColumn= ($this->countModules('position-3') or $this-
>countModules('position-6') or $this->countModules('position-8'));
$showbottom = ($this->countModules('position-9') or $this-
>countModules('position-10') or $this->countModules('position-11'));
$showleft = ($this->countModules('position-4') or $this-
>countModules('position-7') or $this->countModules('position-5'));

if ($showRightColumn == 0 and $showleft == 0)
{
    $showno = 0;
}

JHtml::_('behavior.framework', true);

// Get params
$color = $this->params->get('templatecolor');
$logo = $this->params->get('logo');
$navposition = $this->params->get('navposition');
$headerImage = $this->params->get('headerImage');
$app = JFactory::getApplication();
$doc = JFactory::getDocument();
$templateparams = $app->getTemplate(true)->params;
$config = JFactory::getConfig();

$bootstrap = explode(',', $templateparams->get('bootstrap'));
$jinput = JFactory::getApplication()->input;
$option = $jinput->get('option', '', 'cmd');

if (in_array($option, $bootstrap))
{
    // Load optional rtl Bootstrap css and Bootstrap bugfixes
    JHtml::_('bootstrap.loadCss', true, $this->direction);
```

```

    }

    $doc->addStyleSheet(JUri::base() . 'templates/system/css/system.css');
    $doc->addStyleSheet(JUri::base() . 'templates/' . $this->template .
'/css/position.css', $type = 'text/css', $media = 'screen,projection');
    $doc->addStyleSheet(JUri::base() . 'templates/' . $this->template . '/css/layout.css',
$type = 'text/css', $media = 'screen,projection');
    $doc->addStyleSheet(JUri::base() . 'templates/' . $this->template . '/css/print.css',
$type = 'text/css', $media = 'print');
    $doc->addStyleSheet(JUri::base() . 'templates/' . $this->template .
'/css/general.css', $type = 'text/css', $media = 'screen,projection');
    $doc->addStyleSheet(JUri::base() . 'templates/' . $this->template . '/css/' .
htmlspecialchars($color) . '.css', $type = 'text/css', $media = 'screen,projection');

    if ($this->direction == 'rtl')
    {
        $doc->addStyleSheet($this->baseurl . '/templates/' . $this->template .
'/css/template_rtl.css');
        if (file_exists(JPATH_SITE . '/templates/' . $this->template . '/css/' . $color
. '_rtl.css'))
        {
            $doc->addStyleSheet($this->baseurl . '/templates/' . $this->template .
'/css/' . htmlspecialchars($color) . '_rtl.css');
        }
    }

    JHtml::_('bootstrap.framework');
    $doc->addScript($this->baseurl . 'templates/' . $this->template .
'/javascript/md_stylechanger.js', 'text/javascript');
    $doc->addScript($this->baseurl . 'templates/' . $this->template .
'/javascript/hide.js', 'text/javascript');
    $doc->addScript($this->baseurl . 'templates/' . $this->template .
'/javascript/respond.src.js', 'text/javascript');
    $doc->addScript($this->baseurl . 'templates/' . $this->template .
'/javascript/template.js', 'text/javascript');

    ?>

    <!DOCTYPE html>
    <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="<?php echo $this-
>language; ?>" lang="<?php echo $this->language; ?>" dir="<?php echo $this-
>direction; ?>" >
        <head>
            <?php require __DIR__ . '/jsstrings.php';?>

            <meta name="viewport" content="width=device-width, initial-
scale=1.0, maximum-scale=3.0, user-scalable=yes"/>
            <meta name="HandheldFriendly" content="true" />
            <meta name="apple-mobile-web-app-capable" content="YES" />

```

```

<jdoc:include type="head" />

<!--[if IE 7]>
<link href="<?php echo $this->baseurl ?>/templates/<?php echo
$this->template; ?>/css/ie7only.css" rel="stylesheet" type="text/css" />
<![endif]-->
</head>
<body id="shadow">
<?php if ($color == 'image'):?>
<style type="text/css">
.logoheader {
background:url('<?php echo $this->baseurl . '/' .
htmlspecialchars($headerImage); ?>') no-repeat right;
}
body {
background: <?php echo $templateparams-
>get('backgroundcolor'); ?>;
}
</style>
<?php endif; ?>

<div id="all">
<div id="back">
<header id="header">
<div class="logoheader">
<h1 id="logo">
<?php if ($logo) : ?>
get('sitetitle'));?>" />
<?php endif;?>
<?php if (!$logo AND $templateparams-
>get('sitetitle')) : ?>
<?php
echo
htmlspecialchars($templateparams->get('sitetitle'));?>
<?php elseif (!$logo AND $config-
>get('sitename')) : ?>
<?php
echo
htmlspecialchars($config->get('sitename'));?>
<?php endif; ?>
<span class="header1">
<?php
echo
htmlspecialchars($templateparams->get('sitedescription'));?>
</span></h1>
</div><!-- end logoheader -->
<ul class="skiplinks">
<li><a href="#main" class="u2"><?php
echo JText::_('TPL_BEEZ3_SKIP_TO_CONTENT'); ?></a></li>

```

```

                                <li><a href="#nav" class="u2"><?php
echo JText::_("TPL_BEEZ3_JUMP_TO_NAV"); ?></a></li>
                                <?php if ($showRightColumn) : ?>
                                    <li><a href="#right"
class="u2"><?php echo JText::_("TPL_BEEZ3_JUMP_TO_INFO"); ?></a></li>
                                <?php endif; ?>
                                </ul>
                                <h2 class="unseen"><?php echo
JText::_("TPL_BEEZ3_NAV_VIEW_SEARCH"); ?></h2>
                                <h3 class="unseen"><?php echo
JText::_("TPL_BEEZ3_NAVIGATION"); ?></h3>
                                <jdoc:include type="modules" name="position-
1" />
                                <div id="line">
                                    <div id="fontsize"></div>
                                    <h3 class="unseen"><?php echo
JText::_("TPL_BEEZ3_SEARCH"); ?></h3>
                                    <jdoc:include type="modules"
name="position-0" />
                                </div> <!-- end line -->
                                </header><!-- end header -->
                                <div id="<?php echo $showRightColumn ?
'contentarea2' : 'contentarea'; ?>">
                                    <div id="breadcrumbs">
                                        <jdoc:include type="modules"
name="position-2" />
                                    </div>
                                    <?php if ($navposition == 'left' and $showleft) :
?>
                                        <nav class="left1 <?php if
($showRightColumn == null){ echo 'leftbigger';} ?>" id="nav">
                                            <jdoc:include type="modules"
name="position-7" style="beezDivision" headerLevel="3" />
                                            <jdoc:include type="modules"
name="position-4" style="beezHide" headerLevel="3" state="0 " />
                                            <jdoc:include type="modules"
name="position-5" style="beezTabs" headerLevel="2" id="3" />
                                        </nav><!-- end navi -->
                                        <?php endif; ?>
                                    <div id="<?php echo $showRightColumn ?
'wrapper' : 'wrapper2'; ?>" <?php if (isset($showno)){echo
'class="shownocolumns";}?>>
                                        <div id="main">
                                            <?php if ($this-
>countModules('position-12')) : ?>
                                                <div id="top">

```

```

type="modules" name="position-12" />
</div>
<?php endif; ?>

<jdoc:include type="message" />
<jdoc:include type="component"

/>

</div><!-- end main -->
</div><!-- end wrapper -->

<?php if ($showRightColumn) : ?>
    <div id="close">
        <a href="#" onclick="auf('right')">
            <span id="bild">
                <?php echo
JText::_('TPL_BEEZ3_TEXTRIGHTCLOSE'); ?>
            </span>
        </a>
    </div>

    <aside id="right">
        <h2 class="unseen"><?php echo
JText::_('TPL_BEEZ3_ADDITIONAL_INFORMATION'); ?></h2>
        <jdoc:include type="modules"
name="position-6" style="beezDivision" headerLevel="3" />
        <jdoc:include type="modules"
name="position-8" style="beezDivision" headerLevel="3" />
        <jdoc:include type="modules"
name="position-3" style="beezDivision" headerLevel="3" />
    </aside><!-- end right -->
<?php endif; ?>

<?php if ($navposition == 'center' and
$showleft) : ?>
    <nav class="left <?php if
($showRightColumn == null) { echo 'leftbigger'; } ?>" id="nav" >

        <jdoc:include type="modules"
name="position-7" style="beezDivision" headerLevel="3" />
        <jdoc:include type="modules"
name="position-4" style="beezHide" headerLevel="3" state="0 " />
        <jdoc:include type="modules"
name="position-5" style="beezTabs" headerLevel="2" id="3" />

    </nav><!-- end navi -->
<?php endif; ?>

```



```

        <div class="wrap"></div>
    </div> <!-- end contentarea -->
</div><!-- back -->
</div><!-- all -->

<div id="footer-outer">
    <?php if ($showbottom) : ?>
        <div id="footer-inner" >

            <div id="bottom">
                <div class="box box1"> <jdoc:include
type="modules" name="position-9" style="beezDivision" headerlevel="3" /></div>
                <div class="box box2"> <jdoc:include
type="modules" name="position-10" style="beezDivision" headerlevel="3" /></div>
                <div class="box box3"> <jdoc:include
type="modules" name="position-11" style="beezDivision" headerlevel="3" /></div>
            </div>

        </div>
    <?php endif; ?>

    <div id="footer-sub">
        <footer id="footer">
            <jdoc:include type="modules" name="position-
14" />
        </footer><!-- end footer -->
    </div>
</div>
<jdoc:include type="modules" name="debug" />
</body>
</html>

```