

**O'ZBEKISTON RESPUBLIKASI OLIY VA O'RTA MAXSUS
TA'LIM VAZIRLIGI**

BUXORO DAVLAT UNIVERSITETI

Fizika -matematika fakulteti

“Axborot texnologiyalari” kafedrası

Shaxrulloeva Shoxida Abdulloyevna

**“Universitet informatsion tizimida kafedra yig'ilishlari monitoringini
modulini yaratish”**

5130200-“Amaliy matematika va informatika” ta'lim yo'nalishi bo'yicha
bakalavr darajasini olish uchun

BITIRUV MALAKAVIY ISHI

**“Ish ko'rildi va himoya qilishga
ruxsat berildi”**

Kafedra mudiri
_____ dots.T.B.Boltayev
«_____» _____ 2016 y.

Ilmiy rahbar _____ X.U.Xayatov
“_____” _____ 2016 y.
Taqrizchi _____ Nurullayev M.M
“_____” _____ 2016 y.

“Himoya qilishga ruxsat berildi”
Fakultet dekani _____ Sh.M.Mirzayev
“_____” _____ 2016 y.

Buxoro-2016

MUNDARIJA

Kirish.....	2
I. Web sahifalarni yaratish uchun dasturlar	
1.1 HTML tili.....	7
1.2 ASP NET MVC texnologiyasi.....	17
1.3 SQL tili.	32
II. Universitet information tizimida kafedra yig'ilishlari monitoring modulini yaratish	
2.1 Kafedra yig'ilishlari monitoringgi spetsifikatsiyasi.....	47
2.2 Dastur loyihasi.....	48
2.2.1 Ma`lumotlar ER modeli.....	48
2.2.2 Foydalanuvchi grafik interfeysi.....	55
2.3 Asosiy modular va amalga oshirish.....	58
Xotima.....	74
Foydalanilgan adabiyotlar.....	75

KIRISH

Biz o'z istedodli, fidoiy bolalarimiz, farzandlarimizga bilim va kasb cho'qqilarini zabt etishi uchun qanot berishimiz kerak.

I.A.Karimov

Ma'lumki ta'lim orqali xalqning o'ziga xosligi, madaniyati, tarixi anglanadi, mamlakatlarning bir-biriga yaqinlashishi ta'minlanadi. Biz anchadan buyon O'zbekistonda ta'limdagi, jumladan oliy ta'limdagi islohotlarning borishini qiziqish bilan tahlil qilib, boramiz. Oliy ma'lumotli kadrlarni 2 bosqichda tayyorlashga o'tish-bu, xalqaro maydonga chiqish, yuqori zamonaviy talablarga moslashish demakdir.

Biz yashayotgan XXI asr–intellektual boylik hukmronlik qiladigan asr. Kimki bu haqiqatni o'z vaqtida anglab olmasa, intellektual bilim, intellektual boylikka intilish har qaysi millat va xalq uchun kundalik hayot mazmuniga aylanmasa, bunday davlat jahon taraqqiyoti yo'lidan chetda qolib ketishi muqarrar. Shuningdek bu asr intellektual salohiyat, tafakkur va ma'naviyat asri sifatida insoniyat oldida yangi-yangi ufqlar ochish bilan birga, biz ilgari ko'rmagan, duch kelmagan keskin muamolarni ham keltirib chiqarmoqda. Davr bizning oldimizga qo'yayotgan bunday muammolarni yechish, talablarni bilish, ularni chuqur anglash uchun albata zamonaviy bilim zarur.

Ta'limning yangi modeli jamiyatimizning potensial kuchlarini ro'yobga chiqarishda juda katta ahamiyat kasb etadi. Har qaysi insonda muayyan darajada intellektual salohiyat mavjud. Agar shu ichki quvvatning to'liq yuzaga chiqishi uchun zarur bo'lgan barcha shart-sharoit yaratilsa, tafakkur har xil qotib qolgan eski tushuncha va aqidalardan xalos bo'ladi, va har qaysi inson Olloh taola ato etgan noyob qobiliyat va iste'dodini avvalo o'zi uchun, oilasining, millati va

xalqining, davlatining faravonligi, baxt-saodati, manfaati uchun to'liq baxshida etsa, bunday jamiyat shu qadar kuchli taraqqiyotga erishadiki, uning surat va samarasini hatto tasavvur qilish oson emas. Oldimizda turgan buyuk maqsadlarga erishishda jamiyatimizning har bir a'zosi qo'shadigan hissa shu hisobdan beqiyos bo'lishi muqarrar.

Bugun biz tez sur'atlar bilan o'zgarib borayotgan, imkoniyat hozirga qadar boshidan kechirgan davrlardan tubdan farq qiladigan o'ta shiddatli va murakkab bir zamonda yashamoqdamiz. Davlat va siyosat arboblari, faylasuflar va jamiyatshunos olimlar, sharhlovchi va jurnalistlar bu davrni turlicha ta'riflab, har xil nomlar bilan atamoqda. Kimdir uni yuksak texnologiyalar zamoni desa, kimdir tafakkur asri, yana birov yalpi axborotlashuv davri sifatida izohlamogda. Albatta bu fikrlarni barchasiga ham ma'lum ma'noda haqiqat bor. Chunki ularning har biri o'zida bugungi serqirra va rang-barang hayotning qaysidir belgi alomatini aks ettirishi tabiiy.

Ammo ko'pchilikning ongida bu davr globallashuv davri tariqasida taasurolt uyg'otmoqda. Bunday ta'rif meningcha, ko'p tomondan masalaning mohiyatini to'g'ri ifodalaydi. Nega deganda hozirgi paytda yer yuzining qaysi chekkasida qanday bir voqea yuz bermasin odamzot bu haqida dunyoning boshqa chekkasida zudlik bilan xabar topishi hech kimga sir emas.

Globallashuv jarayoni hayotimizga tobora tez va chuqur kirib kelayotganining asosiy omili va sababi xususida gapirganda shuni ob'yektiv tan olish kerak bugungi kunda har qaysi davlatning taraqqiyoti va ravnaqi nafaqat yaqin va uzoq qo'shnilar, balki jahon miqyosida boshqa mintaqa va hududlar bilan shunday chambarchas bog'lanib borayaptiki, biron mamlakatning bu jarayondan chetda turishi ijobiy natijalarga olib kelmasligini tushunib anglash qiyin emas.

Shu ma'noda globallashuv bu avvalo hayot sur'atlarining beqiyos darajada tezlashuvi demakdir. Har bir ijtimoiy hodisaning ijobiy va salbiy tomoni bo'lgani singari, globallashuv jarayoni ham bundan mustasno emas. Hozirgi paytda uning

g'oyat o'tkir va keng qamrovli ta'sirini deyarli barcha sohalarda ko'rish, hisoblash mumkin. Ayniqsa davlatlar va xalqlar o'rtasidagi integratsiya va hamkorlik aloqalarining kuchayishi xorijiy investitsiyalar, kapital va tovarlar ishchi kuchining erkin harakati uchun qulayliklar vujudga kelishi, ko'plab yangi ish o'rinlarining yaratilishi, zamonaviy kommunikatsiya va axborot texnologiyalarining, ilm-fan yutuqlarining tezlik bilan tarqalishi, turli qadriyatlarning umuminsoniy negizda uyg'unlashuvi tabiiyki, bularning barchasiga globallashuv tufayli erishilmoqda.

Binobarin, biz davlatimiz kelajagini o'z qobig'imizga o'ralib qolgan holda emas, balki umumbashariy va demokratik qadriyatlarni chuqur o'zlashtirgan holda tasavvur etamiz.

Oliy majlisning qonun ijodkorligi faoliyatida mamlakatimiz iqtisodiy sohada amalga oshirilayotgan tarkibiy o'zgarishlarni qulay investitsion muhitni shakllantirish, ishlab chiqarishni modernizatsiya qilish va texnik qayta jihozlash bank moliya tizimini rivojlantirishni normativ huquqiy jihatdan ta'minlash masalalari alohida o'rin egallaydi.

Bu davrda xususiy mulkni huquqiy jihatdan himoya qilishni kuchaytirish, mamlakatimizda har tamonlama baquvvat mulk egalari sinfini shakllantirishni ta'minlash harakatini mustahkamlash, iqtisodiyotni yanada liberallashtirishni ta'minlash, tadbirkorlik, avvalo kichik biznes faoliyatini rivojlantirish uchun qulay sharoitlar yaratish keng tarmoqli bozor infratuzilmasini barpo etishni ko'zda tutadigan qonun hujjatlarining butun bir majmuasi qabul qilindi.

Aholi daromadlari va xalq farovonligini uzluksiz oshirib borish, keng miqyosdagi ijtimoiy dasturlarni muvaffaqiyatli amalga oshirish masalalari parlamentning e'tibor markazida bo'ldi.

Iqtisodiy islohatlarni qonuniy jihatdan ta'minlashga yo'naltirilgan bunday va boshqa bir qator tadbirlar 2009-2012-yillarga mo'ljallangan jahon moliyaviy iqtisodiy inqirozning salbiy oqibatlarini imkon qadar kamaytirishga

qaratilgan inqirozga qarshi choralar dasturini samarali amalga oshirishni huquqiy ta'minlash dunyodagi sanoqli davlatlar qatorida O'zbekistonga iqtisodiyotning barqaror o'sish sur'atlarini saqlab qolish va aholining real daromadlarini oshirish imkonini berdi.

Hozirgi bozor iqtisodiyoti sharoitida yashar ekanmiz yoshlarimizning ijtimoiy-iqtisodiy taraqqiyotda faol ishtirokchi bo'lib voyaga yetishlari, o'zining aql zakovati, ilmiy salohiyati, ma'naviyati, madaniyati bilan milliy istiqlol g'oyalarini mustahkamlashda, ozod va obod vatanni qo'riqlashda yonib yashaydigan shaxs qilib voyaga yetkazishimiz kerak.

Bugungi kunda talim-tarbiya sohasidagi islohotlarning yo'nalishlari va prinsiplari O'zbekiston Respublikasining "Ta'lim to'g'risidagi qonuni", "Kadrlar tayyorlash milliy dasturi" va Davlat ta'lim standart talablarida ko'rsatib berilgan, "Kadrlar tayyorlash milliy dasturi" ning talablaridan biri yoshlarni umuminsoniy qadriyatlarga asoslangan shaxsiy va kasbiy fazilatlarni shakllantirish, izlanish, tadbirkorlik, ishbilarmonlik ko'nikmalarini tarkib toptirishdan iboratdir. Bu esa har bir o'qituvchidan yuqori bilim va yuksak madaniyat egasi, tashkilotchi hamda tadbirkor bo'lishi talab qilinadi.

Mavzuning dolzarbligi: Ta'lim sohasida axborot texnologiyalarning imkoniyatlaridan foydalanish. Ish hujjatlarni tayyorlashda talab darajasida rasmiylashtirish va ishonchli saqlashini ta'minlash.

Bitiruv malakaviy ishining maqsadi: ASP NET MVC texnologiyasidan foydalanib kafedra yig'ilishlari monitoringini olib borish qulaylik yaratuvchi web ilova yaratish.

Bitiruv malakaviy ishining vazifalari: ASP NET MVC texnologiyasining imkoniyatlari, afzalliklari hamda web ilovani tayyorlashda foydalaniladigan usullarini ko'rsatib berish.

Bitiruv malakaviy ishining o'rganilganlik darajasi: Bitiruv malakaviy ishini tayyorlash jarayonida kerakli manbalar bir tizimga keltirildi. ASP NET

MVC da Entity Fraemworkning DataBaseFirst usulidan foydalanib ma'lumotlar ombori ustida amallar bajarish haqida ma'lumotga ega bo'ldim. Web ilova yaratish jarayonida MVC, Database first, SQL server kabi texnologiya hamda MBBT lari bilan ishlashni o'rgandim. Web ilova kafedra bayonnomalari strukturasiga muvofiq o'rganilib tayyorlandi.

Bitiruv malakaviy ishining predmeti: HTML, ASP NET MVC texnologiyasi, SQL tili va kafedra yig'ilishlari.

Bitiruv malakaviy ishining obyekti: Kafedra tuzilmasining yig'ilishlarini avtomatlashtirish dasturi.

Bitiruv malakaviy ishining ilmiy farazi: Kafedra yig'ilishlari monitoringi modulini yaratishda ASP NET MVC texnologiyasi va SQL tillarini qo'llash.

Bitiruv malakaviy ishining yangiligi: Ma'lumotlar omborini shakllantirishda Entity Fraemwork hamda ma'lumotlar ustida amallar bajarishda MVC texnologiyasidan foydalanish.

Bitiruv malakaviy ishining amaliy ahamiyati: Kafedra yig'ilishlarini tashkil qiluvchi xodimning ishi osonlashtiriladi, xodimlarga kafedrada bo'ladigan yig'ilishlar qachon, qaysi masalalar ko'rilishi, yig'ilishni qaysi o'qituvchi olib borishi, ko'rilgan masalalar yuzasidan chiqarilgan qarorlar haqida ma'lumotlar berib boriladi.

Bitiruv malakaviy ishining metodologik asoslari: WEB arxitektura va SQL tili.

Bitiruv malakaviy ishining metodlari: WEB serverni dasturlashda ASP NET MVC texnologiyasidan va ma'lumotlar bazasi, ma'lumotlarini qayta ishlashda SQL tilidan foydalanildi.

Bitiruv malakaviy ishining tarkibi va hajmi: Ushbu bitiruv malakaviy ishi 75 betdan iborat bo'lib, kirish, 2 ta bob, xotima, foydalanilgan adabiyotlar ro'yxatidan iborat.

I. Web sahifalarni yaratish uchun dasturlar

1.1 HTML tili

HTML tili buyruqlari teg (tag) deb ataluvchi maxsus elementlar yordamida beriladi, ya'ni uning asosini teglar tashkil etadi. Teglar < > qavs orasida berilib, ular brouzerda ko'rinmaydi, balki qulay ko'rinishga keltirish uchun xizmat qiladi. Odatda, ko'pchilik teglar ikki marta takrorlanib, jufti

bilan beriladi, ya'ni «ochilib-yopiladi». Masalan, < BODY>, < / BODY>.

HTML tili tanasida ajratib ko'rsatish uchun teglar katta harflar bilan yoziladi va < HTML > bilan boshlanib, </ HTML > bilan tugaydi. Bunda katta va kichik harflar farqlanmaydi.

Shuningdek, HTML tili andozasi bo'yicha hujjatga <HEAD> va <BODY> teglarini kiritish tavsiya etiladi. Brauzer HTML hujjatni o'qiganida, ularning borligi hujjat bo'limlarini aniq ko'rsatadi. Biroq ular bo'lmasa ham, brauzer HTML — hujjatni to'g'ri o'qiydi, lekin hujjat bo'limlari bir-biridan ajralib turmaydi.

Shunday qilib, to'g'ri tuzilgan HTML — hujjat quyidagi tuzilishga ega:

< HTML >

< HEAD>

Sarlavhaga oid ma'lumot

< / HEAD>

< BODY>

Hujjatning mazmuni

< / BODY>

</ HTML >

Bunda < HEAD>, < / HEAD> orasida joylashgan sarlavhaga oid ma'lumot qismida, odatda, foydalanuvchiga e'tiborli bo'lmagan, lekin brouzer uchun lozim ma'lumot beriladi.

< BODY>, < / BODY> orasiga esa uning operatorlari to'laligicha ketma-ketlikda joylashtiriladi.

Masalan :

< HTML >

< HEAD>

< TITLE> My 1 web </TITLE> (Bu darchaning sarlavhasi uchun)

< /HEAD>

< BODY>

<p>Mening birinchi veb-sahifam</p> (Bu tahrir darchasi uchun)

< /BODY>

</HTML >

HTML tili asosiy operatorlari va ularning tasnifi: HTML tili operatorlardan tashkil topadi. Ularning asosiylarini ko'rib chiqamiz.

1. — izoh. Shu belgi orasiga joylashtirilgan ixtiyoriy matn izoh deb qaraladi.

2. **A**

<A>... — hujjatga gipermurojaat o'rnatish. Atributlari (qo'shimcha yordamchi operatorlari, bular <A>... orasida ishlatiladi): HREF, NAME, TARGET, _blank, _top, _parent, _self, TITLE, TYPE, CHARSET, HREFLANG.

HREF atributi — gipermurojaatga olib boruvchi URL ta'riflovchisi vazifasini bajaradi. Masalan: O'zMU . Agar yo'nalish ichma-ich joylashgan papkalarda joylashgan bo'lsa, «/» belgisi yordamida ko'rsatiladi. Agar murojaatni shu sahifadagi biror belgiga o'rnatish lozim bo'lsa, «#» belgisidan keyin ko'rsatiladi.

NAME — agar gipermurojaat shu sahifaning o'ziga o'rnatilgan bo'lsa, o'tishni ushbu atribut yordamida ko'rsatish mumkin.

TARGET — gipermurojaat qayerda ochilishi kerakligini ko'rsatadi.

blank — gipermurojaat natijasini yangi darchaga ochib beradi.

top — natijani brouzerdagi barcha darchalarga ochish (maxsus FRAME lar uchun).

parent — natijani brouzerdagi bosh sahifa darchasiga ochish (maxsus FRAME lar uchun).

self — natijani joriy darchada ochish (maxsus FRAME lar uchun).

TITLE — bu hujjat darchasining sarlavhasini ifodalash uchun.

TYPE — gipermurojaat o'rnatilgan hujjatning MIME — turini aniqlaydi.

CHARSET — gipermurojaat o'rnatilgan hujjatning yozuv kodini aniqlaydi.

HREFLANG — gipermurojaat o'rnatilgan hujjatning yozuv tilini aniqlaydi.

Masalan: <a href=first.html#mavzu № 1 type = «text/html» charset = «iso-8859-6» hreflang = «ru» target=»_blank»

3. B

... — matnni to'q-qalin shrift bilan tasvirlaydi.

Masalan: bir, ikki, uch .

4. BASEFONT

<BASEFONT>...</BASEFONT> — hujjatda avvaldan qabul qilingan shriftning o'lchami, turi va rangini ko'rsatish uchun ishlatiladi. Odatda, matnda shrift o'lchamini bir xilda, ta'kidlashsiz ko'rsatish uchun ishlatiladi. Shrift o'lchami 1 dan (eng kichik) 7 gacha (eng katta). Ta'kidlanmasa, avtomatik ravishda 3 o'lchovida yoziladi.

Masalan: <BASEFONT SIZE = «4»>

5. BGSOUND

Sahifa foniga ovozni ulash. *.mid, *.wav, *.au turdagi ovozli fayllarni qabul qiladi.

— **SCR** — orqali ovozli fayl joylashgan joy ko'rsatiladi.

— **LOOP** — fonli ovozning takrorlanish sonini ko'rsatish, «-1» qiymati yoki «INFINITE» ni tanlash orqali cheksiz qilib qo'yish mumkin.

6. BODY

<BODY>...</BODY> — veb-sahifani to'ldiruvchi matn, deskriptorlar va boshqa ma'lumotlarni aniqlaydi, ya'ni asosiy ma'lumotlar ushbu teglar orasiga kiritiladi. Atributlari:

BACKGROUND — fon tasviri yoki rasmning manzilini ko'rsatish.

Masalan: <body background = «rasml.gif» ... >

BGCOLOR — fon rangini ko'rsatish, bunda rang o'mida tasvir yoki rasmlardan ham foydalanish mumkin.

Masalan: <body background= «rasml.gif» bgcolor=»red» ...> .

TEXT — matnning asosiy rangi.

Masalan: <body background=»rasml.gif» bgcolor= «gray» text= «black» ... >

LINK — gipermurojaatli matn rangi.

Masalan: <body background = «rasml.gif» bgcolor = «red» text = «black» link = «blue»... >

VLINK — gipermurojaatli matn tanlangandan keyingi rangi;

Masalan: <body background= «rasml.gif» bgcolor= «red» text=«black» link= «blue» vlink= «olive»... >

ALINK — joriy gipermurojaat rangi, kursor yordamida tanlangan paytda shu rang ko'rinadi.

Masalan: <body background= «rasml.gif» bgcolor= «red» text= «black» link= «blue» vlink= «olive» alink= «orange»... >

7. BR

Satrnı avtomatik bo'lib, keyingi satrga o'tkazish.

8. DD

Matn chap chegara bo'yicha tekislanadi. Ta'riflar ro'yxati (DL) ichida ishlatiladi.

Masalan: <dl>

<dt> Birinchi pog'ona

<dd> Ikkinchi pog'ona</dd>

</dl>

9. DL

<DL>...</DL> — ta'riflar ro'yxatini ko'rsatadi. Ichida <DT> teg orqali aniqlanayotgan termin, <DD> teg bilan esa termin ta'rifı beriladi.

Masalan: <dl>

<dt>termin

<dd> termin ta'rifı <dd>

</dl>

10. EMBED

Sahifadagi bajariluvchi obyektlar uchun zarur bo'lgan vositalar joylashgan joyni ko'rsatish yoki avtomatik bajariladigan qilish. Masalan:

Flash, VRML, QuickTime, Adobe Acrobat va boshqa vositalar orqali.

SCR — orqali obyekt (fayl) ning joylashgan joyi yo'lini ko'rsatish.

11. FONT

... — shrift parametrlarini ko'rsatadi.

Atributlari:

SIZE — matn o'lchamini ko'rsatish.

COLOR — matn rangini ko'rsatish.

FACE — shrift nomini ko'rsatish, shuningdek bir nechta shriftni ham o'rnatish mumkin.

Masalan: qizil rang, o'lchami uch

12. H1

Matn qismi yoki holati.

<H1>...</H1> — birinchi pog'ona sarlavhalari (eng kattasi).

Atributlari:

ALIGN — Tekislashni aniqlash.

Masalan: <h1 align= «center» ... > ... </h1>

FONT — berilgan matn qismi shriftini ko'rsatish.

Masalan: <h1 align = «center» font = «Verdana» ... > ... </h1>

SIZE — shrift o'lchamini ko'rsatish.

Masalan: <h1 align = «center» font = «Verdana» size = «3» >
... </h1>

13. H2

<H2>...</H2> — ikkinchi pog'ona sarlavha. Umuman olti xil sarlavha mavjud. Ularning qolgan to'rttasi <H3>,<H4>,<H5>,<H6> (eng kichigi) bilan belgilanadi.

14. HR

<HR> — gorizontal chiziq (chizg'ich) qo'yadi.

15. I

<I>...</I> — matnni yozma shrift bilan tasvirlaydi.

<i> Bir, ikki, uch </i>.

16. IMG

 — sahifada rasm joylashtirishda ishlatiladi.

Masalan: , bu yerda photol — sizning veb-sahifangizdagi fayl bilan bitta katalogda turgan rasm nomi.

Atributlari:

SCR — rasm joylashgan joyni ko'rsatish.

ALT — rasm ochilguncha yoki ochilmay qolganda tegishli matnli satr kiritish.

ALIGN — tasvirda matnning holatini ko'rsatish:

top — yuqori chegara bo'yicha; bottom — pastki chegara bo'yicha; left — chap chegara bo'yicha; right — o'ng chegara bo'yicha; center — markaz bo'yicha.

WIDTH — rasm kengligi;

HEIGHT — rasm balandligi;

HSPACE — gorizontal bo'yicha tasvirgacha bo'lgan bo'sh maydonni aniqlash;

VSPACE — vertikal bo'yicha tasvirgacha bo'lgan bo'sh maydonni aniqlash.

BORDER — tasvir atrofidagi chiziq o'lchamini ko'rsatish.

NAME — tasvir nomini aniqlaydi.

Masalan:

17. LI

...() — ro'yxatdagi har bir element boshlanishini aniqlaydi (odatda, ... yoki ... ro'yxat teglari orasida ishlatiladi).

VALUE — ro'yxat yozilish tartibini (nomer boshini) aniqlaydi.

TYPE — ro'yxatdagi nomerlash turini ko'rsatadi.

Masalan: <li value = «5»> ...

18. OL

... — to'liq tartiblangan ro'yxatni aniqlaydi.

Atributlari:

TYPE — tartiblangan ro'yxat turini ko'rsatish;

1 — arab raqamlari yordamida tartiblash;

A — katta harflarda tartiblash;

a — kichik harflarda tartiblash;

I — rim raqamlarida tartiblash;

i — kichik rim raqamlarida tartiblash;

START=n — nechadan boshlanishi;

COMPACT — ro'yxatni ixcham ko'rinishda tasvirlash uchun ishlatiladi.

Masalan:

```
<OL TYPE=I START=15>
```

```
<LI> Dasturlash
```

```
<LI> Algoritmlash
```

```
<LI> Loyihalash
```

```
</OL>
```

Natijasi:

XV. Dasturlash

XVI. Algoritmlash

XVII. Loyihalash

19. P

<P>...</P> — abzas (xat boshi)ni aniqlaydi.

Atributlari:

ALIGN — tekislashni aniqlash;

left — chap chegara bo'yicha;

right — o'ng chegara bo'yicha;

center — markaz bo'yicha;

justify — kengligi bo'yicha (eniga);

```
<p align = «Justify»> ... </p>
```

20. PRE

<PRE>...</PRE> — oldindan formatlangan matnni aniqlaydi, matn holatini saqlaydi.

Masalan: `<pre> Probellarni tekshirish </pre>`.

21. SCRIPT

Java Script kodini aniqlaydi.

22. SUB

Matnni quyi indeks kabi ifodalash

Masalan: _{bir, ikki, uch}

23. SUP

Matnni yuqori indeks kabi ifodalash

Masalan: ^{bir, ikki, uch}

24. TABLE

Jadval yaratishda ishlatiladi.

<TABLE> ... </TABLE>

<TABLE ... > ichida atributlari yoziladi.

Atributlari:

BORDER — jadval chegarasi chiziq kengligini ko'rsatish.

Masalan: Ctable border = «2» ... > ... </table>

ALIGN — gorizontal bo'yicha tekislash.

left — chap chegara bo'yicha tekislash;

right — o'ng chegara bo'yicha tekislash;

center -- markaz bo'yicha tekislash.

Masalan: <table border = «2» align = «center»... > ... </table>

WIDTH, HEIGHT — kengligi va balandligi. Odatda, piksel yoki foizlarda beriladi.

Masalan: <table border = «2» align = «center» width = «300» height = «100%»... > ... </table>

CELLSPACING — Qo'shni yacheykalar orasining kengligini ko'rsatish, piksel yoki foizlarda beriladi.

Masalan: Ctable border = «2» align = «center» cellpadding = «5»... > ... </table>

CELLPADDING — yacheykalardagi ma'lumot va chiziq chegarasi orasining kengligini ko'rsatish, piksel yoki foizlarda beriladi.

Masalan: `<table border = «2» align = «center» cellpadding = «5» cellpadding =«10%» ... > ... </table>;`

FRAME — jadvalda qaysi chegaralami ko'rsatish va qaysilarini ko'rinmaydigan qilish;

BGCOLOR — jadval foni rangini ko'rsatish.

Masalan: `<table bgcolor = «green» border = «2» align = «center» ... > ... </table>`

BORDERCOLOR — jadval chegarasi rangi;

BACKGROUND — jadval foni sifatida tasvirdan foydalanish.

25. TD

`<TD>...</TD>` — jadval satrida alohida yacheykani ramkaga oladi.

26. TH

`<TH>...</TH>` — jadval sarlavha yacheykasi uchun ishlatiladi.

27. TITLE

`<TITLE>...</TITLE>` — sarlavhani tashkil etadi. Ushbu teglar orasida brouzerda hujjat nomini ko'rsatish mumkin.

28. TR

`<TR>...</TR>` — jadvalda satming boshi va oxiri.

Jadval satrini aniqlaydi. Odatda, jadvalda `<TH>` yoki `<TD>` teglari bilan ishlatiladi.

29. U

`<U>...</U>` — matnni ostki qismi chizilgan holda tasvirlaydi.

Masalan: `<u> Bir, ikki, uch </u>`.

30. UL

`<UL>...</UL>`

Markerli ro'yxat yaratish. `<LI>` yordamida beriladi.

Atributlari:

TYPE — marker turini ko'rsatish;

disc — bo'yalgan doira;

circle — bo'yalmagan doira;

square — kvadrat;

COMPACT — matn ko'rinishini ixchamlashtirish.

Masalan: `<ul type=square> <li> Bir <li> Ikki<li> Uch <li>`

`To'rt <li> Besh </ul>`

Natijasi:

- Bir
- Ikki
- Uch
- To'rt
- Besh

1.2. ASP NET MVC texnologiyasi.

ASP NET MVC (model – view - controller) patteri (shabloni)dan foydalanib, web- ilovalarni yaratish uchun platformani taqdim etadi.

Yangi platforma ustida 2007 yildan ish boshlangandi, 2009 - yilda birinchi versiyasi mavjud bo'ldi. 2012 yilgacha 4 - versiya ishlab chiqarildi, freymvorkni o'zi esa butun dunyo bo'ylab taniqli bo'lib qoldi.

Yangi platforma asosidagi MVC shablon, uchta komponentning o'zaro aloqasini ko'zda tutadi: kontroller (controller), model (model) va ko'rinish (tavsifi) (view).

Kontroller (controller) – bu komponent ilovaning boshlang'ich sinfini bildiradi, bundan ilovaning ishi boshlanadi. Bu sinf model va ko'rinish o'rtasidagi aloqasini ta'minlaydi. Foydalanuvchi tomonidan kirgizilgan ma'lumotlarni olib, kontroller ichki logikasiga qarab, kerak bo'lsa modelga murojaat qiladi va shunga tegishli ko'rinishni yaratadi.

**Ko`rinish** (view) - bu ilovaning vizual qismi yoki foydalanuvchi interfeysi. Masalan, html – sahifa, qaysisi orqali foydalanuvchi saytga kirib ilova bilan aloqada bo`ladi.

Model (model) – bu sinflar to`plami, qaysilari foydalangan ma`lumotlarning logikasini tavsif etadi.

Kontrollerlar uchun bizlarda **Controllers** papkasi bor. Bu papkaning ustiga sichqonchaning o`ng tugmasini bossak paydo bo`lgan menyuda **Add- >Controller...ni** ajratib olamiz. Yangi kontroller yaratish oynasi paydo bo`ladi. Kontroller nomini **Home Controller** deb qo`yamiz va Addni bosamiz:

Quyidagi kontroller paydo bo`ladi:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;
namespace BookStore.Controllers
{
    public class HomeController : Controller
    {
        // GET: /Home/
        public ActionResult Index()
        {
            return View();
        }
    }
}
```

Kontroller - dasturiy ilovaning asosiy tarkibiy qismi. Bu modelni va foydalanuvchi interfeysini bog`laydi. Kontroller – bu oddiy sinf, bu Controller bazali sinfidan meros qilib olinadi. Hozircha u bitta Index metodiga ega. Bu metod View () – kelajak ifodani qaytaradi. Agar ilovani hozir yuklasak, xato bo`ladi,

chunki bu metod uchun ifodalash aniqlanmagan. Ifodani qo`shishdan oldin va foydalanuvchi interfeysini yaratishdan oldin Index metodini bunday o`zgartiramiz:

```
using System;
    using System.Collections.Generic;
    using System.Linq;
    using System.Web;
    using System.Web.Mvc;
    using BookStore.Models;
    namespace BookStore.Controllers
    {
        public class HomeController : Controller
        { // ma`lumotlar kontekstini yaratamiz
            BookContext db = new BookContext();
            public ActionResult Index()
            {
                //ma`lumot/bazasidan hamma Book ob`yektlarini olamiz
                IEnumerable<Book> books = db.Books;
                // hamma olingan ob`yektlarni Books dinamik xossasiga uzatamiz
                ViewBagga
                    ViewBag.Books = books;
                    // ifodalashni qaytaramiz
                    return View();
            }
        }
    }
```

Diqqatni qaratamiz, moddellar boshqa nomlar doirasida joylashgan, bitta loyihada bo`lsalar ham. Shuning uchun uni ko`chirish (import qilish) kerak. Keyin

ma'lumotlar kontekstini yaratamiz. **Books** xossasini ishlatib, **MBdan** Book ob'yektlari ro'yxatini olamiz.

Keyin **Books** ob'yektini **View Bag** ob'yektida yaratamiz va ro'yxatni unga beramiz. **View Bag** ob'yekti ifodalovchiga uzatiladi. Bu ob'yektga har qanday o'zgaruvchini aniqlashimiz mumkin va unga qiymat berishimiz mumkin. Keyin ifodalash jarayonida uni olib ko'rsatishimiz mumkin. Endi ifodalashni yaratamiz. Kursorni **Index** metodiga qo'yib o'ng tugmachani bosamiz. Menyuda **Add View...** ni tanlaymiz:

Keyin bizlarga ifodalash oynasi ochiladi, qaysisida ifodalashning nomi va boshqa parametrlar tanlanadi. CHunki proektimizda bet – ustasi hali yo'q, **Use a layout or masterpage** dan bayroqchani olib tanlaymiz va **Add** tugmasini bosamiz.

Bundan keyin Visual Studio avtomatik tarzda **Index.cshtml** ifodasini ochadi. Bu ifoda minimal belgilash kodini html tilida saqlaydi. Uni quyidagicha o'zgartiramiz:

```
@{ Layout = null; }
<!DOCTYPE html>
<html>
<head>
    <meta name="viewport" content="width=device-width" />
    <title>Knijniy magazin</title>
</head>
<body>
    <div>
        <h3>Rasprodaja knig</h3>
        <table>
            <tr><td><p>Nazvanie knigi</p></td>
                <td><p>Avtor</p></td>
                <td><p>TSena</p></td><td></td>
            </tr>
        </table>
    </div>
</body>
</html>
```

```

</tr>
@foreach (var b in ViewBag.Books)
{
<tr>
<td><p>@b.Name</p></td>
<td><p>@b.Author</p></td>
<td><p>@b.Price</p></td>
<td><p><a href="/Home/Buy/@b.Id">Kupit'</a></p></td>
</tr>
}
</table> </div></body></html>

```

Bu yerda bizlar jadvalni yaratamiz, bu jadvalda kreditlash dasturlari haqida ma'lumot joylashadi. `@ foreach (var c in ViewBag. Books)` konstruktsiyasi maxsus qiziqishni ko'rsatadi. Bu konstruktsiya Razor ifodalashning harakatchi sintaksisini ishlatadi (ifodalashni yaratganimizda Razorni tanladik). Bilishimiz kerak `@` belgisidan keyin bizlar C# tilida kodni ishlatishimiz mumkin.

TSiklda bizlar ViewBag. Books ob'yektining elementlarini ko'rib chiqamiz. Keyin har bir elementning xossalarini olamiz va ularni jadval kataklariga joylashtiramiz.

Oxiridagi murojaat - `<a href="/Home/Buy/@b.Id">Kupit'</a>` -xaridni chizmaiyalashtirish formasi joylashish manzilini ko'rsatadi.

Model yaratganda shartliliklar.

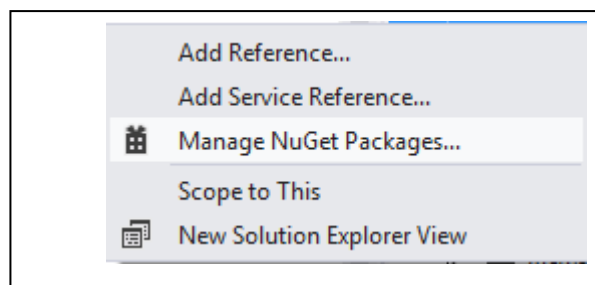
Model yaratsak ba'zi shartliliklar mavjud bo'ladi. Chunki ma'lumotlar bazasi va unga tegishli model dastlabki kalitga ega bo'lishi kerak (Primary Key) shu kalitni ifodalash kerak. Ifodalashning ikkita yo'li bor. Bu yerda bizlar xossaning nomi bilan kalitni ifodalaymiz. Buning uchun model xossani nomi bilan saqlanishi kerak. Model - nomi ID yoki IDning o'zi. Purchase modelida Purchase

ID xossasi ifodalangan, ya'ni bu xossa dastlabki kalit deb bilinadi. Key atributi bilan kalitni ifodalash – bu ikkinchi usul.

Ma'lumotlarning kontekstini yaratish va Entity Framework

Ma'lumotlarga qulayroq ulanish uchun Entity Framework freymvorkni ishlatish tavsiya etiladi. Bu freymvork ma'lumotlar bazasi strukturasidan tashqari, model orqali hamma harakatlarni bajarishga imkonini beradi. Bizlar Empty shabloni bilan loyihani yaratdik, bu shablon bo'sh, shuning uchun loyihaga, MB (ma'lumotlar bazasi) bilan aloqa qilish uchun, Entity Framework fayllarini qo'shamiz. Agar biz boshqa xil shablon ishlatsak, masalan, Basic yoki Internet Application bunda Entity Framework kutubxonasi avtomatik tarzda qo'shilsa edi.

Entity Frameworkni loyihaga qo'shish uchun, sichqonchanning o'ng tugmachasini bosamiz, loyihaaning strukturasini References bo'g'imiga (uning ichida loyihaga ulangan kutubxonalar saqlanadi) va paydo bo'lgan menyuda ManageNuGet Packages...ni tanlaymiz:



1.2.1-chizma

NuGet paketlarni boshqarish oynasida, teppadagi o'ng burchagida **Entity** qidiruv maydonini kiriting va Enterni bosing. Bundan keyin hamma topilgan paketlar, so'rovga qarashli, o'rtadagi ustunda ko'rsatiladi.

Keyin paketni yuklaymiz. Endi Entity Framework ning hamma ustunliklarini, ma'lumotlarni boshqarish uchun ishlatamiz. Yuklanishdan keyin ma'lumotlar kontekstini yaratamiz, bunisi model asosida ma'lumotlarga osonlik bilan ulanish uchun kerak. Models papkasiga Book Context yangi sinf qo'shamiz:

```

using System;
using System.Collections.Generic;
using System.Web;
using System.Data.Entity;
namespace BookStore.Models
{
    public class BookContext : DbContext
    {
        public DbSet<Book> Books { get; set; }
        public DbSet<Purchase> Purchases { get; set; }
    }
}

```

Kontekst yaratish uchun, **DbContext** sinfidan yangi sinfni meros qilib olishimiz kerak.

public DbSet<Book> Books { get; set; } turli xossalar aniq, turli (bu yerda Book) ma'lumotlar to'plamini olishga yordam beradi.

Natijada Entity Framework o'zi ma'lumotlar bazasini yaratadi. Bu yerda bizlar **Code First** yondashish bilan ishlaymiz, bor modellar bo'yicha freym vork ma'lumotlar borasida jadvallar yaratadi. Ulanish satrini ko'rsatamiz. Web.config faylni ochamiz, config Sections seksiyasini topamiz undan keyin connection Strings seksiyasini qo'yamiz:

```

<connectionStrings>
  <add name="BookContext"
        connectionString="DataSource=(LocalDB)\v11.0;
        AttachDbFilename='|DataDirectory|\Bookstore.mdf';
        Integrated Security=True"
        providerName="System.Data.SqlClient" />
</connectionStrings>

```

Bunda bizlar ma'lumotlar bazasiga yo'lni aniqlaymiz, qaysisi keyin yaratiladi./DataDirectory/ ifodasi o'rinbosarni tavsiflaydi, qaysisi ko'rsatadi, ma'lumotlar bazasi App_Data loyihaning papkasida yaratiladi.

Marshrutizatsiya asoslari. HomeController kontrollerlarni chaqirish uchun va unga so'rov uzatish uchun, so'rov satrida uning nomini ko'rsatish kerak – **Home**. Bundan tashqari, kontroller nomidan keyin slesh qo'yib kontrollerlarning harakati yoki metodi ko'rsatiladi, qaysisiga so'rov uzatiladi. Proekt ishini boshlagandan keyin yoki saytga murojaat qilganda, mvc Index harakatini chaqiradi, agar boshqa marshrutni ko'rsatsak, marshrutizatsiya parametrlarida.

[/Home/Buy/@b.Id](#) yo'li ko'rsatadi – bizlar Home Controller kontrollerning Buy metodini ishlatamiz. Metodni yaratishdan oldin ilovani ma'lumotlar bilan to'ldiramiz.

Endi saytimizga stil elementlarini qo'shamiz css, stillar turli va bet - ustasini, qaysisi saytning hamma betlarini bir - xil qilib ko'rsatadi. Buning uchun stil faylini aniqlab olishimiz kerak. Odatda hamma stil fayllari Content papkasida saqlanadi, shuning uchun loyihaga Content papkasini qo'shamiz. Bu ishni bajarish uchun proektning nomiga sichqonchaning o'ng tugmachasini bosamiz va chiqqan menyuda Add (qo'shish) → New Folder (yangi papka) tanlaymiz. Endi yangi papkaga stil faylini qo'shamiz. Uni Site.css deb nomlaymiz. Buni bajarish uchun Content papkaning ustiga sichqonchaning o'ng tugmasini bosamiz va chiqqan menyuda Add (qo'shish) → New Item (yangi element) tanlaymiz. Keyin shablonlar ro'yxatidan Style Sheet shablonini topamiz va yangi faylga Site.css nomini beramiz, endi Site.css faylni yozamiz:

```
body {  
    font-size: 13px;  
    font-family: Verdana, Arial, Helvetica, Sans-Serif;  
    background-color: red;  
    padding-left: 40px;
```

```

    }
    nav{ display: block;
}
    .menu {
        padding-left:10px;
    }
    .menu ul {
        list-style:none;
    }
    .menu li {
        display:inline;
    }
    .menu a:hover {
        color:red;
    }
    table {
        vertical-align:middle;
        text-align:left;
    }
    .header {
        font-weight:bold;
    }
    td {
        padding-right:10px;
    }
    input {
        width: 150px; }

```

Menu sinfi bu yerda saytda navigatsiya (harakatlanish) menyusini bildiradi.

Endi stillarni ishlatsak bo`ldi. Bunda ifodalashlarimiz oddiy web – sahifalar, shuning uchun har qanday html sahifaday head qismida stil fayliga murojaatni qo`shishimiz mumkin:

```
<head>
<meta name="viewport" content="width=device-width" />
    <link type="text/css" rel="stylesheet" href="../../Content/Site.css" />
</head>
```

Agar dasturni ilovada bitta ifoda yoki har – xil ifodalash bo`lsa va har xil stillar va boshqa resurslar ishlatilsa bunday yo`nalish haqli. Ammo bizlarda ikkita ifodalanish, qaysilarida bitta menyu va bir xil stil bo`lishi kerak. Bu yerda bet – ustasini ishlatish juda qulay. Bet – ustasi yagona shablonni boshqa ifodalar uchun beradi.

Demak, bet - ustasini yaratamiz. Birinchi bo`lib Views papkasiga yangi Shared papkasini qo`shamiz. Keyin Shared papkasiga yangi ifodani qo`shamiz - Layout.cshtml deb nomlaymiz:

-Layout.cshtml fayliga quyidagi matnni qo`shamiz:

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" />
    <title>@ViewBag.Title</title>
    <link href="@Url.Content("~/Content/Site.css")" rel="stylesheet"
type="text/css" />
</head><body> <nav>
    <ul class="menu">
    <li>@Html.ActionLink("Glavnaya", "Index", "Home")</li>
    </ul> </nav>
    @RenderBody()
```

</body>

</html>

Bu yerda bizlar boshqa xil usulni ishlatdik. Url.Content metodi ishlatildi, qaysisiga faylni topish yo`lini ko`rsatdik. Bu ikkita usul teng huquqlilikka ega. Keyin bizlar sayt menyusini yaratdik Html. Action link metodlar yordamida.

Bu metod element – murojaatni ishlab chiqaradi va murojaatning nomini kontroller metodini va kontroller nomini qabul qiladi.

Oldingi ko`rinishi ilovada Buy metodi purchase parametrni ishlatdi. Berilgan metod POST – so`rovlariga ishlov beradi va bizlar unga quyidagi formani uzatishimiz mumkin:

```
<form method="post" action="">
  <input type="hidden" value="@ViewBag.BookId" name="BookId" />
  <p>Vvedite svoe imya </p>
  <input type="text" name="Person" />
  <p>Vvedite adres :</p>
  <input type="text" name="Address" />
  <input type="submit" value="Otpравit" />
</form>
```

Bu formadagi name atributi hamma maydonlarda model xossalarining nomini bildiradi, shuning uchun tizim avtomatik tarzda maydonlarning qiymatini tegishli xossalar bilan bog`laydi.

File Result sinfi mijozga fayllarni uzatadi. Ammo bu sinf abstraktli, aslida bizlar merosxo`rlar bilan ishlaymiz:

FileContentResult: fayldan o`qilgan, baytlar massivini mijozga uzatadi.

File Path Result: faylni to`g`ridan to`g`ri serverdan uzatadi.

File StreamResult: bu sinf oqimni yaratadi – System.IO.Stream ob`yekti orqali faylni o`qib chiqadi va faylni mijozga uzatadi.

Ko`rsatilgan uchta usulda File metodi ishlatiladi, qaysisi File Result ob`yektini qaytaradi. Faqat tanlangan usulga qarab, bu metodning tegishli qayta yuklangan versiyasi ishlatiladi.

Fayl tizimidan faylni uzatish uchun (File Path Result ob`yektini ishlatish), File metodida uchta parametrlarni ko`rsatamiz:

Server tomonida faylga yo`l, ichidagi narsalar tipi va kutib olgan tomon uchun fayl nomi (fayl nomi zarur emas).

```
public FileResult GetFile()
{
    // Put' k faylu
    string file_path = Server.MapPath("~/Files/PDFIcon.pdf");
    // Tip fayla - content-type
    string file_type="application/pdf";
    // Imya fayla - neobyazatel'no
    string file_name = "PDFIcon.pdf";
    return File(file_path,file_type,file_name);
}
```

Loyihadagi FilesPDFIcon.pdf fayli bor. **Server.MapPath** metodi loyihaning ichida katalogdan resursga to`liq yo`lini sozlash imkonini beradi. Absolyut yo`llarni ishlatish ham mumkin, fayl tizimining har qanday fayliga murojaat qilib, masalan: *string file-path=C "S:\Book\PDFIcon.pdf"*. Agar bizlar Home/GetFile yo`li bilan murojaat qilsak, bu faylni lokal kompyuterga saqlashni tavsiya qiladi.

Saytni betlarning ko`rinishini bir xil qilish uchun bet – ustasi ishlatiladi. Bet – ustasi aslida – bu o`sha – o`sha ifodalar. Bizlar bet – ustasida ba`zi elementlarni aniqlaymiz, qaysilari saytning hamma betlarida joylashadi. Hamda bizlar to`ldiruvchilarni yoki pleysxolderlarni aniqlashimiz mumkin, ularning mavjudligi boshqa ifodalarni ta`minlaydi.

Birinchi ifodani yaratganda bizlar – Layout.html bet – ustasini ishlatdik:

```

<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8" />
    <title>@ViewBag.Title</title>
    <link href="@Url.Content("~/Content/Site.css")" rel="stylesheet"
type="text/css" />
</head>
<body>
    <nav>
    <ul class="menu">
<li>@Html.ActionLink("Glavnaya", "Index", "Home")</li>
    </ul>
    </nav>
    @RenderBody()
</body>
</html>

```

Ko`rinishda bu oddiy ifoda @ Render Body ( ) metodni chaqirishdan boshqa. Bu chaqiruv – pleyxolder, buning joyiga boshqa ifodalar, qaysilari bu bet – ustasini ishlatadilar, o`zining ichidagi narsalarini qo`yadi. SHunday deb, bizlar Veb – ilovaning ifodalari uchun osonlik bilan bir xil stilni o`rnatishimiz mumkin. Bet – ustasini ifoda uchun ishlatsak, bizlar Layout qismida bet – ustasiga yo`lni ko`rsatishimiz kerak.

Masalan, Index.cshtml ifodasi shu bet – ustasini ishlatib, shunday boshlanadi:

```

@{
    Layout = "~/Views/Shared/_Layout.cshtml";
}

```

Ifodalar html belgilarini ishlatadilar, ichidagi narsalarni ko`rsatish uchun. Lekin, ASP.NET MVC freymvorkda juda kuchli asbob mavjud – bu Html – helperlar, bular html kodini ishlab chiqaradi.

Satrli helperlar.

Satrli helperlar metodlarning oddiy ifodalarga C# tilida o`xshaydi. Satrli xelperlar C# tilida yozilgan metodlarning oddiy ifodalarga o`xshaydi, faqat @ helper tegdan boshlanadi.

Masalan, helper ko`rinishida kitoblar nomlarini ro`yxat qilib chiqarish:

```
@helper BookList(IEnumerable<BookStore.Models.Book> books)
{
    <ul>
        @foreach (BookStore.Models.Book b in books)
        {
            <li>@b.Name</li>
        }
    </ul>
}
```

Bu xelperni ifodaning qaysi bir joyida aniqlashimiz mumkin. Hamda ifodaning har qanday joyida uni ishlatish mumkin, unga *IEnumerable<Book Store. Models Book>* ob`yektni uzatib:

```
<h3>Spisok knig</h3>
@BookList(ViewBag.Books)
<!-- ili esli ispol'zuetsya strogo tipizirovannoe predstavlenie -->
@BookList(Model)
```

Satrli html – xelperlar qulay ishlatish, agar bitta metodni yaratish lozim bo`lsa, qaysisini ifodada bir nechta marta ishlatilishi niyat qilinsa.

Formalarni yaratish uchun standartli html elementlari ishlatiladi.

Masalan:

```

<form method="post" action="/Home/Buy">
    <input type="hidden" value="@ViewBag.BookId" name="BookId" />
    <table>
        <tr><td><p>Vvedite svoe imya </p></td>
        <td><input type="text" name="Person" /> </td></tr>
        <tr><td><p>Vvedite adres :</p></td>
        <td><input type="text" name="Address" /> </td></tr>
        <tr><td><input type="submit" value="Otpavit'" /> </td>
        <td></td></tr>
    </table>
</form>

```

Bu oddiy html – forma, qaysisi tugma bosilgandan keyin, POST so`rovi bilan kiritilgan ma`lumotlarni /Home/Buy manzilga uzatadi. Begin Form/EndForm qurilgan xelper o`sha – o`sha formani yaratadi:

```

@using(Html.BeginForm("Buy", "Home", FormMethod.Post))
{
    <input type="hidden" value="@ViewBag.BookId" name="BookId" />
    <table>
        <tr><td><p>Vvedite svoe imya </p></td>
        <td><input type="text" name="Person" /> </td></tr>
        <tr><td><p>Vvedite adres :</p></td>
        <td><input type="text" name="Address" /> </td></tr>
        <tr><td><input type="submit" value="Otpavit'" /> </td>
        <td></td></tr>
    </table>
}

```

Begin form metodi parametr deb, harakat metodi nomini va kontroller nomini qabul qiladi. Bu xelper <Form> ochilish tegini va </form> yopish tegini

yaratadi. Shuning uchun, ifodani chiqish oqimiga rendering qilganda bizlarda xuddi form tegini ishlatganda html – kodi bo`lib chiqadi.

1.3 SQL tili.

Ma'lumotlar bazasi dunyosi tobora yagona bo`lib bormoqda. Bu jarayon har xil kompyuter muhitlarida faoliyat ko'rsatuvchi axborot tizimlarini hosil qilishda qo'llanuvchi yagona standart til yaratishni talab qildi.

Buyruqlar to'plamini bilgan foydalanuvchilarga standart til, ular shaxsiy kompyuter tarmoq ishchi stansiyasida yoki katta EHM da ishlashidan qat'iy nazar, ma'lumot yaratish, izlash va uzatishga imkon beradi.

SQL (Structured Query Language, odatda <<sikvel>> deyiladi), ma'nosi — Tarkiblangan so'rovlar tili. Bu relyatsion Ma'lumotlar bazalarida ishlashga imkon beradigan tildir. Bu til ifodalarining xususiyati shundan iboratki, ular ma'lumotlarni qayta ishlash protseduralariga emas, natijalariga yo'naltirilgandir.

SQL o'zi ma'lumotlar qayerda joylashgani, indekslar qandayligini va hatto, amallarni qanday izchillikda qo'llasa bo'lishini aniqlaydi, bu tafsilotlarni ma'lumotlar bazasiga so'rovlarda ko'rsatish kerak emas. SQL tili IBM kompaniyasida MBBT DB2 yaratish jarayonida ishlab chiqilgan va keng ko'lamda RISC protsessorli mashinalarda UNIX tizimlar asosida hamda meynfreymlarda, super kompyuterlar asosida qurilgan katta hisoblash tizimlarida qo'llanilgan. Shu bilan birga, u mustaqil bo'lmasdan, PL/SQL va Transact-SQL kabi ichki dasturlash

tillariga inkapsulyatsiya qilinadi. 1986-yilda, ANSI (American National Standart Institute) SQL tilining rasmiy standartini ishlab chiqdi, 1992- yilda bu standart kengaytirildi. Butun til 30 ga yaqin operatorlarga ega bo'lib, ba'zi versiyalarida sal ko'proq, ba'zilarida sal kamroqdir. Har qanday MB har xil obyektlarga, ya'ni jadvallar, protseduralar, funksiyalar, tasavurlar, ketma-ketliklar va hokazolarga ega.

<<Kliyent-Server>> texnologiyasiga ko'ra, foydalanuvchi EHM (Kliyent) lar so'rovlari maxsus Ma'lumotlar serverlarida (Server) qayta ishlanadi, foydalanuvchi

EHM larga faqat so'rovni qayta ishlash natijalari qaytariladi. Tabiiyki, Server bilan muloqot qilish uchun yagona til kerak va bunday til sifatida SQL tanlangan.

Shuning uchun hamma zamonaviy relyatsion MBBT versiyalarida (DB2, Oracle, Ingres, Informix, Sybase, Progress, Rdb) va, hattoki, norelyatsion MBBT versiyalarida (masalan, Adabas) «Klient-Server» texnologiyasi va SQL tilidan foydalaniladi. SQL tilida ma'lumotlarni jadval ko'rinishda tasvirlashga yo'naltirilgan amallar konsepsiyasi ko'p bo'lmagan (30 dan kam) ifodalardan iborat kompakt til yaratishga imkon berdi. Ikki xil: Interaktiv va Joylashtirilgan SQL mavjud. Ko'p hollarda ikkala forma bir xil ishlaydi, lekin ikki xil foydalaniladi. Interaktiv SQL Ma'lumotlar bazasining o'zida faoliyat ko'rsatadi va buyurtmachi foydalanishi uchun chiqish hosil qilishda ishlatiladi. SQLning bu formasida siz buyruq kiritsangiz, u darrov bajariladi va darhol natijani (agar u mavjud bo'lsa) ko'rishingiz mumkin. Joylashtirilgan SQL boshqa tilda yaratilgan dasturga joylashtirilgan SQL buyruqlardan iborat.

SQLning interaktiv va joylashtirilgan formalarida ko'p sonli guruhlar yoki subbo'limlar mavjud. Ular ANSI tomonidan e'tiborga olingan va konseptual darajada foydali, lekin ko'pchilik SQL dasturlar ularni alohida qayta ishlamaydi, shuning uchun ular aslida SQL buyruqlarining funksional kategoriyalaridir.

- DDL (Ma'lumotlarni Ta'riflash Tili) — ANSI da sxemani ta'riflash tili, obyektlarni (jadvallar, indekslar, tasavvurlar va hokazo) yaratuvchi buyruqlardan iborat.

- DSL (Ma'lumotlarni O'zgartirish Tili) — bu jadvallarda qanday qiymatlar saqlanishini istalgan daqiqada aniqlovchi buyruqlar majmuasi.

- DCL (Ma'lumotlarni Boshqarish Tili) foydalanuvchiga ma'lum obyektlar ustida ma'lum ta'sir o'tkazishga ruxsat berish yoki bermaslikni aniqlovchi vositalardan iborat. SQL Standarti ANSI tomonidan aniqlangan va hozir IICO (Standartlashtirish bo'yicha xalqaro tashkilot) tomonidan qabul qilingan. Lekin

kommersial ma'lumotlar bazalari dasturlari ANSI ni ogohlantirmasdan SQL ni kengaytiradilar, ya'ni o'zlari foydali deb hisoblagan har xil xossalarni qo'shadilar.

SQL so'rov tili — ma'lum talablar asosida berilganlar bazasiga murojaat qilib, undan so'rov talablariga javob beradigan natijalarni olish tili. Hozirgi paytda turli berilganlar bazasi va ularning sistemalari yaratilgan. Ammo relyatsion tipdagi barcha berilganlar bazasiga murojaat qilib, ulardan tegishli ma'lumotlarni olish SQL tili ishlab chiqilgan. Bu tilda so'rovlar qanday vositalar yordamida tashkil qilinishiga to'xtalib o'tamiz. Shuni aytish joizki, ba'zi BBTS larda (masalan, Accessda) uni ishlatmasdan tanlashga so'rov yoki maxsus namuna bo'yicha so'rov blanki sifatida murojaat qilib tegishli ma'lumotlar olinadi, bunday so'rovlarni tashkil qilish uchun esa База данных — Создать (Berilganlar bazasi yaratish) buyrug'idan foydalaniladi. So'rov natijasini javob sifatida jadval ko'rinishida olish mumkin.

SQL tili hozirda ixtiyoriy berilganlar bazasiga so'rov berib, undan javob olishni ta'minlovchi andozaviy vosita hisoblanadi. Bu til bilan tanishishni real hayotda o'z o'rnini to'la aks ettira oladigan misollar asosida ko'rib chiqamiz.

Kerakli tushunchalar imkon boricha ko'rilyotgan holatlarni izohlash uchun kiritiladi.

Masalan, 1-jadvalda fabrika xizmatchilari haqidagi ma'lumot keltirilgan.

Name	Dob	Pol	Los	Department	Salary	Telno
Rustam V.	17/01/50	M	22	Admin	2700	1338234
Erkin J.	20/02/55	M	17	Worker	2300	1445754
Barot V.	02/07/75	M	21	Sales	2250	6506133

1.3.1-jadval. Factory.

Ushbu jadval 7 ustun va 5 satrdan iborat. Har bir ustun o'z nomiga ega.

Name — xizmatchilar ismi

Dob — tug'ilgan kun sanasi

Pol —jins, Los — fabrikadagi ish staji (davri)

Department — fabrikadagi xizmatchi ishlayotgan bo'lim

Salary — xodimlarning oylik maoshi

Telno — telefon raqami

Yuqoridagi ustunlar nomi matritsa atributlari nomini olgan. Har bir satr aniq xizmatchi (xodim) haqidagi ma'lumotni o'z ichiga oladi.

Har bir ustun har xil uzunlik va turga ega ekanligini e'tiborga olamiz. Name ustunida fabrika xodimlarining familiyalari qo'llanilmoqda, ya'ni maksimal uzunligi 15 belgiga ega bo'lgan so'zlar. Dob ustuni sana turiga, boshqa uzunlikka va h.k. ga ega. Ustunning turi va uzunligi uning atributlari hisoblanadi. 1 -ustundan ko'rinib turibdiki, bu ta'riflar har xil va shuning uchun ham jadvalni belgilashda ular tegishli tarzda aniqlanishi kerak. Nihoyat, 1-jadvalni boshqa jadvaldan ajrata olish uchun u o'zining nomiga ega bo'lishi kerak. Shu yerda va bundan buyon ham 1-jadvalga <<Factory>> nomini berdik.

Jadvallarni tuzishda ularda 2 ta butunlay bir xil satr bo'lmasligi kerak. Shuni e'tiborga olish kerakki, jadvaldagi hamma ustunlar ham bir ma'noda har bir satrni aniqlayvermaydi.

Masalan, Pol, Los, Salary ustunlari bir-birining nusxasini oluvchi nomlarni o'z ichiga oladi. Bu ular kam funksional nagruzkaga egaligini, ular har bir satrni bir ma'noda aniqlash uchun yaroqli emasligini anglatadi. Bundan tashqari, ulardan bu jadval nima haqida ekanligini aniqlash qiyin.

Keltirilgan jadvalda Name va Telno ustunlari har bir satrni bir ma'noda aniqlash imkonini beradi.

Har bir satrni bir ma'noda aniqlab bera oladigan ustunlar yoki ustunlar guruhi kalitli deb ataladi. Ular 1 -jadvalni tuzishda albatta alohida ajratilishi kerak.

Endi jadvallarning yaratilishi usulini o'rganishga o'tamiz. Jadvalni yaratishda jadval nomini, ustunlar nomini, atributlarining turi va uzunligini berish kerak. SQL satrli, sonli, real vaqt, sana va h.k. o'zgaruvchan turlari kiritilishiga imkon beradi.

Satrli turi — o'zgaruvchilarni tasvirlashga mo'ljallangan belgi va sonlardan tashkil topgan. Bunda birinchi belgi (simvol) albatta harf bo'lishi kerak. Char — uzunligi 254 baytdan oshmaydigan satrli o'zgaruvchini tasvirlashga mo'ljallangan.

Sonli o'zgaruvchilarni tasvirlashda NUMBER kalitli so'z qo'llaniladi. 1.0E — 100 dan 1.0E+100 gacha bo'lgan doirada 22 raqamga ega bo'la oladigan sonlarni tasvirlaydi. Sana va vaqtni tasvirlashda Date standarti qo'llaniladi.

U: — vaqtni aniqlaydi (soat, minut, sekund standartida. Masalan, 18.02.2007);

— sanani ifodalash uchun Yevropa standartidan yoki Amerika standartidan foydalaniladi.

Endi 1-jadvalni yaratish uchun quyidagilarni yozishimiz mumkin:

```
CREATE TABLE factory  
(id NUMBER(5,0) PRIMARY KEY,  
Name CHAR(15) NOT NULL,  
Los NUMBER(2,0),  
Dept CHAR(15),  
Salary NUMBER (7,2),  
Telno CHAR(7)).
```

Biz Name ustunini har bir atribut 15 belgiga ega bo'lgan satrli o'zgaruvchi uzunlik bilan aniqladik. Amaliyotda atribut uzunligi keragidan ko'ra ortiqroq beriladi. Bu shu narsa bilan bog'liqki, biz oldindan satrli o'zgaruvchining qaysi ma'nolari jadvalga kiritilishini bilmaymiz. Shuning uchun ma'lum bir zaxira bo'lgani ma'qul. Bundan tashqari bo'sh o'rinlar zaxirasi ustunlar orasidagi masofaning o'sishiga imkon beradi. 1 -jadval chop etishda dizayn nuqtayi nazaridan ham yanada ko'rkamlashadi. NAME ustuni NOTNULL ni o'z ichiga oladi. Bu esa uning bo'sh satrlarni o'z ichiga ola olmasligini bildiradi. Har bir SQL gapi so'ngida <<nuqtali vergul>> turishi kerak. Berilgan gap bo'yicha EHM factory

jadvalini yaratadi, lekin jadval unga ma'lumotlar kiritilmagani sababli bo'sh bo'ladi.

Standart SQL tilida ma'lumotlarni kiritish INSERT buyrug'i asosida amalga oshiriladi. Bu buyruq bitta satrni kiritishga imkon beradi, keyingi satrlarning kiritilishi INSERT buyrug'ining qaytarilishi yordamida hosil bo'ladi:

```
INSERT INTO factory (Name, Dob, Pol, Los, Dept, Salary, Telno) Values  
(Paul F. 05.07.1962, 'm', 12 'worker', 2250.75, '420027');
```

Barcha satrli o'zgaruvchilar apostroflarga kiritilishi lozim. Agar biz biror o'zgaruvchining ma'nosini bilmasak, kiritishda uning ma'nosini tushirib qoldirish mumkin, EHM uning o'rnini avtomatik ravishda bo'shliq bilan to'ldiradi.

```
Misol: INSERT INTO factory (Name, Dob, Pol, Dept, Salary)
```

```
Values ('Clint E', 12. 12.1970, 'm\ 'tecknic', 2400);
```

Bunday holatda Telno va Los o'rinlari, toki ular tartibli aniqlanmaguncha, EHM yordamida probel va nullar bilan to'ldiriladi, bu jarayon korrekt yaqinlangunicha davom ettiriladi. Alohida qiymatlar tushib qolgan bo'lsa, ularni ham NUll bilan to'ldirish mumkin.

Select buyrug'i

Bu bo'limda foydalanuvchining jadvallar bilan ishlash usullari ko'rib chiqiladi. Select buyrug'i SQL tilining asosiy buyruqlaridan biri hisoblanadi. Bu buyruq barcha amallarni ma'lumotlar qatori bilan ta'minlaydi. Quyida select buyrug'ining asosiy imkoniyatlarini ko'rib chiqamiz.

a) Barcha ma'lumotlarni ko'rish .

```
Select* from tactory;
```

(F) belgisi factory jadvalining barcha ustunlarini tanlash kerakligini bildiradi. Bu natijani Select buyrug'idan keyin barcha ustunlar nomini berish bilan olsa bo'ladi:

```
Select Name, Dob, Pol, Los, Dept, Salary, Telno from factory;
```

Natijada ekranda 1-jadvalning hamma ustunlari paydo bo'ladi.

Ustunlar tartibi Select buyrug'ida belgilangandek bo'ladi:

b) ustunlarni tanlab chaqirish.

Ustunlarni alohida chaqirish uchun Select buyrug'ida ko'rsatish kerak:

Select Name, Dept from factory;

Natijada bir Name va Dept dagi jadvallarni olamiz:

d) Satrlarni tanlab ko'rish . Ma'lumotlar bilan ishlaganda ko'pincha jadvaldan faqat aniq bir satrni ko'rish lozim bo'lgan hollar uchray turadi. Bunday holda Select buyrug'ining umumiy ko'rinishi quyidagicha bo'ladi:

Select — ustunlar nomi;

from — jadvallar nomi;

where — satr tanlash sharti.

e) Solishtirish opera torlarining qo'llanilishi.

Avvalgi misolda satr tanlash shartini tasvirlashda <<tenglik>> belgisi qo'llangan edi.<<Tenglik>> belgisidan tashqari yana solishtirish shartini tanlashda qo'llanilishi mumkin bo'lgan 7 ta eng oddiy solishtirish operatorlari mavjud.

Bular:

!= - teng emas;

> — katta;

< > — teng emas;

! > — berilgandan katta emas;

< — berilgandan kichik;

! < — berilgandan kichik emas;

>= — katta yoki teng;

<= — kichik yoki teng.

Quyidagi misol solishtirish operatorlarining qo'llanish imkoniyatlarini ko'rsatadi.

1-misol: Factory jadvalidan ish staji 10 yildan ortiq bo'lgan ishchilarning ismlarini tanlash.

Select Name, Los from factory

where Los> 10;

Natijada ekranda

NAME	Los
Rustam V.	22
Barot V.	21
Azamat T.	17
Dolly S.	14
Paul F.	12
Joan A.	11

1.3.2-jadval.Factory jadvali.

chiqadi.

2-misol: Factory jadvalidan 12.12.1970 dan keyin tug'ilganlarni tanlash.

Select Name, Dob

from factory

where Dob>12.12.1970;

Ekranda hisobot quyidagicha bo'ladi:

NAME	Dob
Barot V.	02.07.1975
Maria G.	20.11.1972
ViancaA.	14.03.1975
JaneS.	01.02.1979

1.3.3-jadval. Factory jadvali.

Yuqorida tasvirlangan solishtirish operatorlaridan tashqari, quyidagi operatorlarni qo'llash orqali ham qiyoslash mumkin.

1) between... and

2) is null

3) like

4) in

Between operatori aniq bir sohani ajratishga mo'ljallangan.

Misol:

select Name, Salary from factory

where Salary between 2150 and 2350;

Ushbu misol factory jadvalida oylik maoshi 2150—2350 oralig'ida bo'lgan xodimlar ro'yxatini ko'rsatadi.

Name	Salary
Azamat T.	2300
Barot V.	2250
Raul T.	2250
Don T.	2250
Donoron A.	2250

1.3.4-jadval. Factory jadvali.

SQL not between birikmasining qo'llanishiga yo'l qo'yadi.

Misol:

select Name, Salary from factory

where Salary not between 2150 and 2350;

Bu ekranda oylik maoshi 2150—2350 oralig'idan tashqarida bo'lgan xodimlarning ro'yxatini ko'rsatadi.

Is null operatori axborot bo'lmagan satrlarni tanlash imkonini beradi.

Misol: Factory jadvalidan telefoni bo'lmagan ishchilar ro'yxatini tanlash.

Select Name, Telno from factory where Telno. Is null;

Natijada quyidagi jadvalga ega bo'lamiz:

Name	Telno
Jane S.	null
DonT.	null

1.3.5-jadval. Factory jadvali.

Like operatori. Like operatori satrli o'zgaruvchilarni solishtirishni ta'minlaydi. EHM satrli o'zgaruvchilarni shablon bo'yicha solishtiradi. Bu holda satrli o'zgaruvchini solishtirishda shablondan keyin <<%>> belgisi turadi, o'zgaruvchining o'zi qavs ichida yoziladi.

Misol: Factory jadvalidagi Name ustunida kimning nomi <<DO>> dan boshlansa, o'shani ekranga chiqarish.

```
select Name from factory
where Name like, 'DO%';
```

Bu operator satrli kattaliklarni alohida harflari aniq bo'lganda tanlashni ta'minlaydi.

Misol uchun:

```
select Name from factory
where Name like <<AN%>>;
```

Chizish AN dan oldin har qancha miqdor bo'lish imkoni borligini anglatadi.

Bo'limda muhokama qilinadigan eng oxirgi operator In operatori bo'ladi. U tanlov asosida boshqa ma'lumotlarga mos bo'lgan ma'lumotlarni yig'ish imkonini beradi.

Masalan:

```
select Name, Dept
from factory
where Dept in ('admin', 'account');
```

Berilgan buyruq asosida EHM factory jadvalidan administrativ (ma'muriy) bo'limda va buxgalteriyada ishlayotgan xodimlarning ro'yxatini keltiradi.

Name	Dept
Rustam V.	admin
Edwin E.	Account
Dolly S.	Account

1.3.6-jadval. Factory jadvali.

(I — IV) punktlarda tasvirlangan barcha operatorlar NOT operatori bilan ham ishlashi mumkin;

f) shartlarni o'zaro almashtirish .

Hozirgacha biz jadvaldan ma'lumotlarni tanlab olishda faqat bitta shartdan foydalandik. SQL tili bitta iborada ikki yoki undan ortiq shartlarning qo'llanishiga imkon beradi. Bu holatda shartlar bir-biridan AND yoki OR biriktiruvchi kreditlar yordamida ajratiladi. Quyida biz shartlarni almashtirishni qo'llash imkoniyatini ko'rsatadigan ikkita misol keltiramiz.

1-misol. Factory jadvalidan worker bo'limiga tegishli va staji 10 yildan ortiq bo'lgan xodimlarning ro'yxatini tanlash.

```
Select Name, Los, Dept
from factory
where Dept='worker' and Los> 10;
```

Natijada ekranda quyidagilar hosil bo'ladi:

Name	Los	Dept
Erkin J.	17	worker
PauL T.	12	worker

1.3.7-jadval. Factory jadvali.

2-misol. Factory jadvalidan 01.01.1960 dan keyin tug'ilgan, oylik maoshi 2050 dan kam yoki 2450 dan ortiq, telefon raqami bor xodimlarning ro'yxati, tug'ilgan sanasi, oylik maoshi va telefon raqamlarini tanlash.

```
Select Name, Dob, Salary, Telno
```

from factory

where not between 2050 and 2450 and Dob>01.01.1960;

Arifmetik operatsiyalar. SQL faqat tanlashni emas, balki ma'lumotlar bilan arifmetik amallarni bajarishni ham ta'minlaydi.

Masalan:

Select Name, Salary*1.2

from factory

where Dept= 'service';

Natijada ekranda quyidagilar hosil bo'ladi:

Name	Salary* 1.2
Bianca A.	2340
Jane S.	2280

1.3.8-jadval. Factory jadvali.

Agar oxirgi ifodada shart bo'lmasa, EHM barcha xodimlarning oylik maoshini 20% ga oshiradi. Shuni aytib o'tish kerakki, factory jadvalining o'zida Bianca A. va Jane S. ning oylik maoshi o'zgarishsiz qoladi. Jadval ma'lumotlarini o'zgartirish uchun maxsus buyruqlar kerak, ular haqida keyinroq suhbatlashamiz. Ma'lumotlar bazasi bilan ishlaganda shu kundagi sana va vaqtni bilish kerak. Shu maqsadda SQLda quyidagi operatorlar nazarda tutilgan.

SYSTIME — soat, minut, sekund formatida shu kungi vaqtni belgilaydi.

SYS DATE — kun, oy, yil formatida shu kungi sanani belgilaydi.

Misol:

Select Name, Sysdate,systime

from factory

where Salary = 2700 or Salary =3000;

Natijada quyidagiga ega bo'lamiz:

Name	Sysdate	Systime
Rustam B.	14.01.1997.	14:02:27.

1.3.9-jadval. Factory jadvali.

Shuningdek, sysdate va systime operatorlari bilan arifmetik operatsiyalarni bajarish mumkin.

Masalan:

Select Name, Sysdate+31

from factory

where Name like 'Rustam%';

Berilgan buyruq asosida, shu kungi sanaga EHM 31 kunni qo'shib qo'yadi.

Natijada quyidagilar hosil bo'ladi:

Name	Sysdate+31
Rustam B.	14.02.97

1.3.10-jadval. Factory jadvali.

SQL sana va vaqt bilan bog'liq bo'lgan arifmetik operatsiyalardagi literallar kabi, sana va vaqtning har xil konstantalarini qo'llashga imkon beradi. Shu maqsadda SQL ga yil(lar), oy(lar), kun(lar), soat(lar), minut(lar), sekund(lar), mikrosekund(lar), YEAR(S), month(S), Day(S), HOUR(S), MINUTE(S), SECOND(S), MICROSECUND(S) lar kiritiladi.

Quyidagi misol mazkur konstantalarni qo'llash usulini ko'rsatadi.

Select Name, Dob, Dob+5 years

from factory

where Name like 'Dob%';

Display ekranida quyidagilar hosil bo'ladi:

Name	Dob	Dob+5 years
DonT.	11.04.1966	11.04.1971
Donovan A.	14.05.1967	14.05.1972

1.3.11-jadval. Factory jadvali.

1 – bob xulosasi. Malakaviy bitiruv ishimning 1 – bobi “Web sahifani yaratish uchun zaruriy dasturlar” deb nomlangan. Bu bob uch qismdan iborat. 1- qismi “HTML tili” deb nomlangan bo`lib, unda Web sahifalar yaratish uchun HTML tili va undagi HTML uning elementlari, atributlari haqida ma’lumotlar keltirilgan va misollar bilan tushuntirilgan. 2- qismi “ASP NET MVC texnologiyasi” deb nomlangan bo`lib, unda bu texnologiya orqali web sahifalarni yaratish imkoniyatlari yoritilgan. 3 – qismi “SQL tili” deb nomlangan bo`lib, unda Web server, SQL da berilganlar bazasi bilan ishlash imkoniyatlar haqida aytilib o`tilgan.

II.Universitet information tizimida kafedra yig'ilishlari monitoring modulini yaratish.

2.1. Kafedra yig'ilishlari monitoringi spetsifikatsiyasi.

Oliy ta'lim muassasasining muhim bo'g'inlaridan biri bu kafedra hisoblanadi. Kafedra ishini to'g'ri tashkil etishda kafedra yig'ilishlarining ahamiyati katta. Kafedra yig'ilishlari kafedra rejasi asosida shakllantiriladi. Kafedra rejalari kafedra mudiri tomonidan ishlab chiqiladi. Keyin kafedra yig'ilishida yangi o'quv yili uchun rejalar kafedra a'zolari tomonidan tasdiqlanadi. Bu kabi jarayonlarni avtomatlashtirish jarayoni tartib solish imkoniyatini beradi. Shu asnoda bitiruv malakaviy ishda kafedra yig'ilishlarining monitoringini yurituvchi dasturiy ta'minot yaratiladi. Bu dasturiy ta'minotda kafedra rejalari kiritiladi. Har bir yig'ilish rejaloriga masalalar, masalalar qaysi o'qituvchi tomonidan muhokama qilinganligini, shu masala asosida kafedra qanday qarorga kelganligini monitoringini yuritish mumkin. Dasturiy ta'minotda har fakultetda mavjud kafedralarni yig'ilishlar monitoringini olib borish mumkin. Kafedra yig'ilishlar monitoring dasturi kafedra yig'ilish bayonnomalari asosida tahlil qilinib, shakllantirildi. Bayonnomaning tashkil etuvchilari quyidagilar:

Bayonnoma raqami

Qatnashchilar

Kun tartibi

Muhokamaga chiqqan qatnashchilar

Kafedra qarorlari

Demak, yuqoridagi ma'lumotlar inobatga olinib, dasturiy ta'minot uchun ma'lumotlar bazasi yaratildi. Quyidagi axborot texnologiyalar kafedrasida o'tkazilgan yig'ilish bayonnomasi namuna sifatida keltirilgan.

Buxoro davlat universiteti Fizika matematika fakulteti "Amaliy matematika va axborot texnologiyalar" kafedrasining 2013-2014 o'quv yilidagi 5480100 - "Amaliy matematika va informatika" ta'lim yo'nalishi talabalarining bitiruv malakaviy ishi himoyasini o'tkazish bo'yicha Davlat Attestatsiya Komissiyasining yig'ilishi

Bayonnoma № 1

Buxoro shahri 09.06.2014

Qatnashdilar: Yo'ldoshev Sh. S.- BMTI "Informatika va AT" kafedrasidotsenti – azo; Rajabov E. YA.- BDU huzuridagi 1-son AL "Aniq fanlar" kafedrasidotsenti – azo; Bakayev Z.Z.- Turizm KHK o'qituvchisi – azo; Bakayev I.I.- "Turizm" KHK o'qituvchisi – azo; Boltayev T.B.- BDU-"Amaliy matematika va AT" kafedrasimudiri – azo; Jalolov O.I.- BDU - "Amaliy matematika va AT" kafedrasidotsenti – azo.

Kun tartibi:
5480100 - "Amaliy matematika va informatika" ta'lim yo'nalishi talabalarining bitiruv malakaviy ishi himoyasini o'tkazish to'g'risida

Eshitildi:
Yo'ldoshev Sh.S talabalarni Buxoro Davlat universitet rektorining 26.03.2014 yildagi №106-U, 29.05.2014 yildagi 21-FM va 26.09.2013 yildagi №332-U buyruqlari bilan tanishtirdi.

Talaba Karimov Abdurahim Adizovichning "Operations tizimlar va ularning qiyosiy xarakteristikallari" mavzuda MBI yuzasidan chiqishi tinglandi. Talabaga MBI bo'yicha quyidagi savollar berildi:

Savol: Windowsda Linuxdagidek imkoniyat yo'qmi?: Bakayev Z.Z

Javob: yo'q.

Nima uchun Windows 8.1 versiyasi olimladi?: Bakayev Z.Z

Nechta OT tahlil qilingan?: Jalolov O.I

MBI mavzusi bo'yicha elektron ma'lumot tayyorladizmi? : Yo'ldoshev Sh. S

Talaba barcha savollarga qoniqarli javob berdi.

Ilmiy rahbar: Tohirov B.

Taqrizchi: Aslonov Q.

Himoya qilish kuni: 09.06.2014

Yig'ilish qaror qiladi:
Karimov Abdurahim Adizovichning "Operations tizimlar va ularning qiyosiy xarakteristikallari" mavzudagi MBI ishi 66 ball 3 baho bilan baholansin.

2.1.1-rasm.Bayonnoma namunasi.

2.2. Kafedra yig'ilishlar monitoringini dastur loyihasi

Kafedra yig'ilishlar monitoringi dastur loyihasi web texnologiyadan foydalanilib yaratildi.Chunki bu ishda o'qituvchilar xohlagan joyda qachon,qanaqa yig'ilishlar rejalashtirilganini,o'tkazilgan yig'ilishlar to'g'risidagi ma'lumotlarini xohlagan joydan ko'rish imkoniyati bo'ladi va bu esa juda ham qulay.Agar dastur DesckTop yoki Client/Server axrxitekturada yaratilsa dastur ancha noqulay ahvolda bo'lardi.Bu holda o'qituvchi uchun istalgan joydan ishlash imkoniyatini bermaydi va ular uchun bu narsa ancha noqulay bo'ladi shu uchun bu dastur loyihasi uchun web texnologiyani tanladim.

2.2.1. Kafedra yig'ilishlar monitoringini ma'lumotlarining ER modeli.

Kafedra yig'ilishlarining monitoringi dasturi universitet informatsion tizimining bir qismi bo'lganligi tufayli bu dastur ma'lumotlarining ER modeli

universitet informatsion tizimining ma'lumotlar omborining ba'zi jadvallari asoslangan holda tuzildi.

Kafedra yig'ilishlar monitoringi uchun universitet informatsion tizimining ma'lumotlar bazasidan quyidagi jadvallar asosida shakllantirildi:

Jadvallar nomi	Izoh
Fakultet	Universitetdagi fakultetlar jadvali
Kafedra	Har bir fakultet kafedralari jadvali
O'qituvchilar	Universitetdagi barcha o'qituvchilar jadvali

2.2.1-jadval.Mavjud jadvallar.

Kafedra yig'ilishlar monitoringini asosiy jadvallar ro'yxati:

KafedraReja	Kafedra tomonidan rejalashtirilgan yig'ilishlar vaqti
Muhokama	Yig'ilish muhokamasi matni
Qaror	Yig'ilishda qo'yilgan masala bo'yicha qabul qilingan qarorlar
Masala	Yig'ilishda ko'riladigan masalalar
Ustoz	Yig'ilishni qaysi o'qituvchi olib borishi

2.2.2-jadval.Asosiy jadvallar ro'yxati.

Kafedra yig'ilishi monitoringini har bir asosiy jadvallarining tasnifi:

1.KafedraReja:

Ustun nomi	Tipi	Izoh
KrejaId	Int	Kalit
Kreja	date	Kafedra yig'ilishining rejalashtirilgan vaqti
Kamalda	date	Kafedra yig'ilishining amalda o'tkazilgan vaqti
KafedraId	int	Kafedra Id

2.2.3-jadval. KafedraReja jadvali.

2.Muhokama:

Ustun nomi	Tipi	Izoh
MuhokamaId	Int	Kalit
UstozId	Int	Yig'ilish o'tkazuvchi ustoz raqami
MasalaId	Int	Muhokamaga qo'yilgan masala raqami
Mmatn	text	Yig'ilish muhokamasi matni

2.2.4-jadval.Muhokama jadvali.

3.Qaror:

Ustun nomi	Tipi	Izoh
QarorId	Int	Kalit
QarorMatn	nvarchar(450)	Yig'ilish qarori matni
MasalaId	Int	Muhokamaga qo'yilgan masala raqami

2.2.5-jadval.Qaror jadvali.

4.Masala:

Ustun nomi	Tipi	Izoh
MasalaId	Int	Kalit
MasalaMatn	nvarchar(250)	Qo'yilgan masala matni
KrejaId	Int	Qo'yilgan masalaga mos rejalashtirilgan sana nomeri

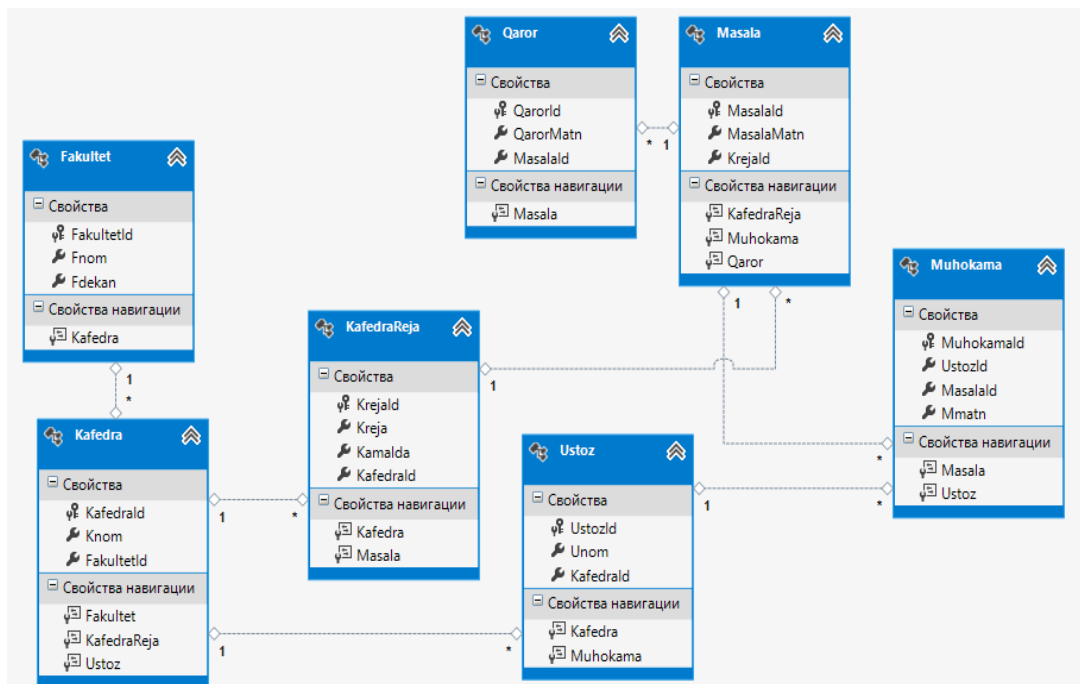
2.2.6-jadval.Masala jadvali.

5.Ustoz:

Ustun nomi	Tipi	Izoh
UstozId	Int	Kalit
Unom	nvarchar(50)	Yig'ilishni olib boruvchi ustoz nomi
KafedraId	Int	Kafedra kalit raqami

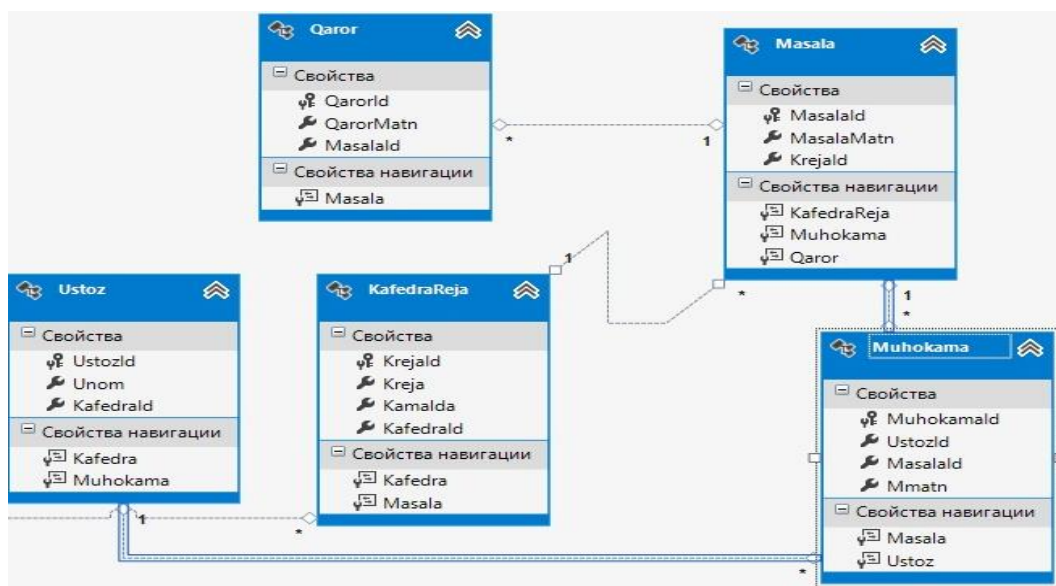
2.2.7-jadval.Ustoz jadvali.

Kafedra yig'ilishlari monitoringini ma'lumotlarini umumiy **ER modeli** yuqoridagi jadvallar asosida yaratildi va uning ko'rinishi quyidagicha:



2.2.1- chizma. Kafedra yig'ilishlari monitoringini ma'lumotlarini umumiy **ER modeli**.

Kafedra yig'ilishlar monitoringini asosiy jadvallarining **ER modeli** quyidagicha:



2.2.2- chizma. Kafedra yig'ilishlar monitoringini asosiy

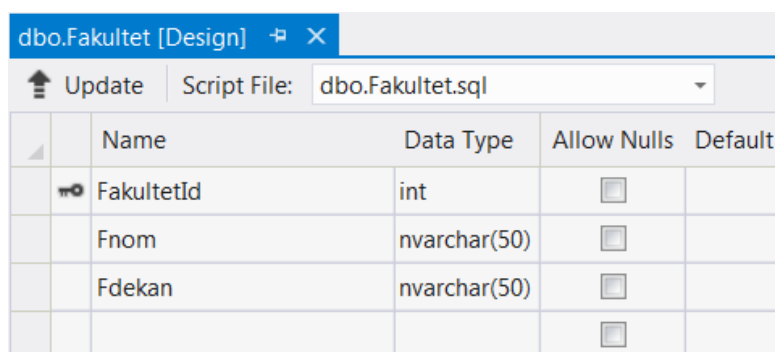
jadvallarining **ER modeli**.

Kafedra yig'ilishlari monitoringini asosiy jadvallarini yaratilishi va ular orasidagi bog'lanishlarni sql tilidagi kodi quyidagicha bo'ladi:

Fakultet jadvali:

```
CREATE TABLE [dbo].[Fakultet] (  
[FakultetId] INT IDENTITY (1, 1) NOT NULL,  
[Fnom] NVARCHAR (50) NOT NULL,  
[Fdekan] NVARCHAR (50) NOT NULL,  
PRIMARY KEY CLUSTERED ([FakultetId] ASC)  
);
```

Fakultet jadvalini konstruktor oynasida tasvirlash:



	Name	Data Type	Allow Nulls	Default
	FakultetId	int	☐	
	Fnom	nvarchar(50)	☐	
	Fdekan	nvarchar(50)	☐	
			☐	

2.2.3 – chizma.Fakultet jadvali.

Kafedra jadvali:

```
CREATE TABLE [dbo].[Kafedra] (  
[KafedraId] INT IDENTITY (1, 1) NOT NULL,  
[Knom] NVARCHAR (50) NOT NULL,  
[FakultetId] INT NOT NULL,  
PRIMARY KEY CLUSTERED ([KafedraId] ASC),  
CONSTRAINT [FK_Kafedra_ToFakultet] FOREIGN KEY  
([FakultetId]) REFERENCES [dbo].[Fakultet] ([FakultetId]) ON  
DELETE CASCADE  
);
```

Kafedra jadvalini konstruktor oynasida tasvirlash:

dbo.Kafedra [Design]*				
Update		Script File: dbo.Kafedra.sql*		
	Name	Data Type	Allow Nulls	Default
	KafedraId	int	☐	
	Knom	nvarchar(50)	☐	
	FakultetId	int	☐	
			☐	

2.2.4 – chizma.Kafedra jadvali.

KafedraReja jadvali:

```
CREATE TABLE [dbo].[KafedraReja] (
    [KrejaId] INT IDENTITY (1, 1) NOT NULL,
    [Kreja] DATE NOT NULL,
    [Kamalda] DATE NOT NULL,
    [KafedraId] INT NOT NULL,
    PRIMARY KEY CLUSTERED ([KrejaId] ASC),
    CONSTRAINT [FK_KafedraReja_ToKafedra] FOREIGN KEY
    ([KafedraId]) REFERENCES [dbo].[Kafedra] ([KafedraId])
);
```

KafedraReja jadvalini konstruktor oynasida tasvirlash:

dbo.KafedraReja [Design]				
Update		Script File: dbo.KafedraReja.sql		
	Name	Data Type	Allow Nulls	Default
	KrejaId	int	☐	
	Kreja	date	☐	
	Kamalda	date	☐	
	KafedraId	int	☐	
			☐	

2.2.5 – chizma. KafedraReja jadvali.

Masala jadvali:

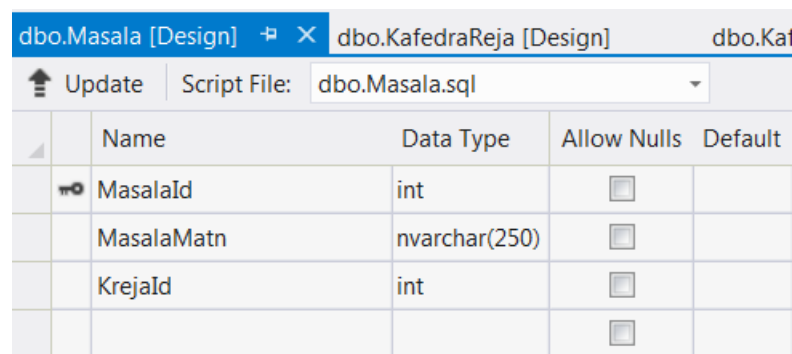
```
CREATE TABLE [dbo].[Masala] (
```

```

[MasalaId] INT IDENTITY (1, 1) NOT NULL,
[MasalaMatn] NVARCHAR (250) NOT NULL,
[KrejaId] INT NOT NULL,
PRIMARY KEY CLUSTERED ([MasalaId] ASC),
CONSTRAINT [FK_Masala_ToKafedrareja] FOREIGN KEY ([KrejaId])
REFERENCES [dbo].[KafedraReja] ([KrejaId]) ON DELETE CASCADE
);

```

Masala jadvalini konstruktor oynasida tasvirlash:



	Name	Data Type	Allow Nulls	Default
	MasalaId	int	☐	
	MasalaMatn	nvarchar(250)	☐	
	KrejaId	int	☐	
			☐	

2.2.6 – chizma.Masala jadvali.

Muhokama jadvali:

```

CREATE TABLE [dbo].[Muhokama] (
[MuhokamaId] INT IDENTITY (1, 1) NOT NULL,
[UstozId] INT NOT NULL,
[MasalaId] INT NOT NULL,
[Mmatn] TEXT NOT NULL,
PRIMARY KEY CLUSTERED ([MuhokamaId] ASC),
CONSTRAINT [FK_Muhokama_ToUstoz] FOREIGN KEY ([UstozId])
REFERENCES [dbo].[Ustoz] ([UstozId]),
CONSTRAINT [FK_Muhokama_ToMasala] FOREIGN KEY ([MasalaId])
REFERENCES [dbo].[Masala] ([MasalaId])
);

```

Muhokama jadvalini konstruktor oynasida tasvirlash:

dbo.Muhokama [Design]				
Update		Script File: dbo.Muhokama.sql		
	Name	Data Type	Allow Nulls	Default
	MuhokamaId	int	☐	
	UstozId	int	☐	
	MasalaId	int	☐	
	Mmatn	text	☐	
			☐	

2.2.7 – chizma.Muhokama jadvali.

Qaror jadvali:

```
CREATE TABLE [dbo].[Qaror] (
    [QarorId] INT IDENTITY (1, 1) NOT NULL,
    [QarorMatn] NVARCHAR (450) NOT NULL,
    [MasalaId] INT NOT NULL,
    PRIMARY KEY CLUSTERED ([QarorId] ASC),
    CONSTRAINT [FK_Qaror_ToMasala] FOREIGN KEY ([MasalaId])
        REFERENCES [dbo].[Masala] ([MasalaId]) ON DELETE CASCADE
);
```

Qaror jadvalini konstruktor oynasida tasvirlash:

dbo.Qaror [Design]				
Update		Script File: dbo.Qaror.sql		
	Name	Data Type	Allow Nulls	Default
	QarorId	int	☐	
	QarorMatn	nvarchar(450)	☐	
	MasalaId	int	☐	
			☐	

2.2.8 –chizma.Qaror jadvali.

Ustoz jadvali:

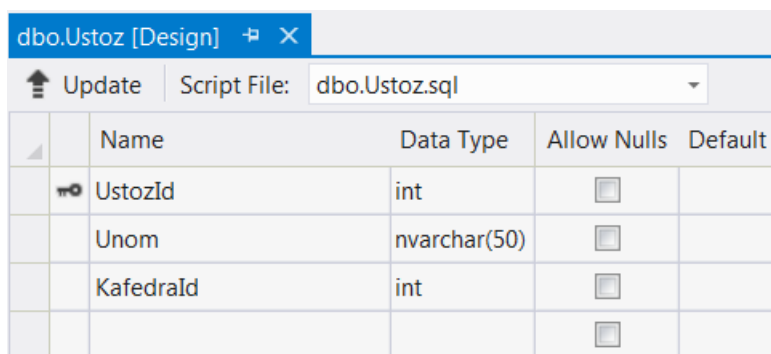
```
CREATE TABLE [dbo].[Ustoz] (
    [UstozId] INT IDENTITY (1, 1) NOT NULL,
    [Unom] NVARCHAR (50) NOT NULL,
```

```

[KafedraId] INT NOT NULL,
PRIMARY KEY CLUSTERED ([UstozId] ASC),
CONSTRAINT [FK_Ustoz_ToKafedra] FOREIGN KEY ([KafedraId])
REFERENCES [dbo].[Kafedra] ([KafedraId]) ON DELETE CASCADE
);

```

Ustoz jadvalini konstruktor oynasida tasvirlash:



	Name	Data Type	Allow Nulls	Default
	UstozId	int	☐	
	Unom	nvarchar(50)	☐	
	KafedraId	int	☐	
			☐	

2.2.9 –chizma.Ustoz jadvali.

2.2.2 Kafedra yig'ilishlari monitoringining foydalanuvchi

grafik interfeysi.

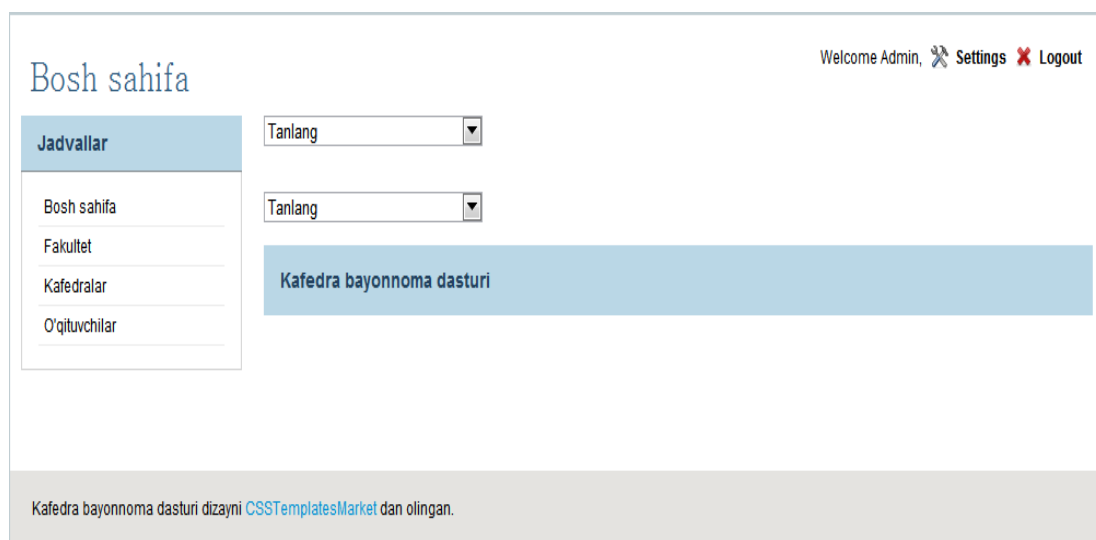
Dastur interfeysida foydalanuvchi dasturni ishga tushirgandan keyin bosh sahifa hosil bo'ladi. Bosh sahifa quyidagi menyulardan tashkil topgan :

Bosh sahifa

Fakultet

Kafedralar

O'qituvchilar



2.2.2.1-chizma.(Kafedra yig'ilishi bayonnomasi dasturi web ilovasi)

Fakultet menyusi tanlansa, universitetda mavjud fakultetlar va fakultet dekanlari nomi chiqadi. Tahrirlash bo'limida agar fakultet nomi o'zgarsa, dekan lavozimiga boshqa kishi tayinlansa o'zgartirish mumkin. Mavjud tahrirlash imkoniyati bilan birga o'chirish imkoniyati ham bor. Oyna pastida qo'shish tugmasi bo'lib yangi fakultet qo'shish mumkin.



2.2.2.2-chizma.Universitetda mavjud fakultetlar ro'yxati oynasi.

Kafedralar menyusida mavjud fakultetlarga tegishli kafedralar ro'yxati va majlislar rejasi chiqadi.

Universitetdagi mavjud fakultetlar ro'yxati					
Kafedra Id	Kafedra nomi	Rejalashtirilgan majlislari	Fakultet nomi	Tahrirlash	O'chirish
6	Amaliy matematika	Reja	Fizika matematika		
7	Fizika	Reja	Fizika matematika		
10	Boshlang'ich ta'lim	Reja	Pedagogika		
11	Chizmachilik	Reja	Pedagogika		
12	Pedagogika Psixologiya	Reja	Pedagogika		
13	Iqtisodiyot	Reja	Ijtimoiy iqtisod		
Yangi fakultet qo'shish uchun quyidagi tugmadan foydalaning!!!					

2.2.2.3-chizma.Kafedralar ro'yxati oynasi.

Reja tanlanganda kafedrada qo'yilgan majlislar ro'yxati chiqadi.

Amaliy matematika kafedrasining majlislar rejasi						
Kafedra Id	Kafedra nomi	Rejada	amalda	Ro'yxat	Tahrirlash	O'chirish
6	Amaliy matematika	01.04.2016	01.04.2016	Masala		
6	Amaliy matematika	02.04.2016	02.06.2016	Masala		
Yangi fakultet qo'shish uchun quyidagi tugmadan foydalaning!!!						

2.2.2.4-chizma.Kafedraning majlislar rejasi oynasi.

Birinchi reja bo'yicha yig'ilish sanasi kiritiladi, so'ngra yig'ilish bo'lib o'tgan amaldagi sana kiritiladi.Bu yerda ham rejadagi va amaldagi sanani,qo'yilgan masalalarni o'zgartirish,o'chirish imkoniyati mavjud. Ro'yxat ustunidagi masalalar tanlanganda ko'riladigan masalalar nomi,majlisning mohokama matni,majlis qarori chiqadi.

Bosh sahifa

Wecome Admin,  Settings  Logout

Jadvallar

Bosh sahifa

Fakultet

Kafedralar

O'qituvchilar

Amaliy matematika kafedrasining majlisida qo'yilgan masalalar

Masala Id	Masala	Kafedra nomi	Reja Id	Muhokama	Majlis qarori	Tahrirlash	O'chirish
24	Iqtidorli talabalar bilan ishlash	Amaliy matematika	5	Muhokama	Qaror		
25	Xo'jalik shartnomalarining borishi	Amaliy matematika	5	Muhokama	Qaror		

Yangi fakultet qo'shish uchun quyidagi tugmadan foydalaning!!!

Qo'shish

2.2.2.5-rasm.Majlisdagi masalalar ro'yxati oynasi.

2.3 Asosiy modullar va amalga oshirish.

Foydalanuvchilarni dasturga kirishini taminlovchi index.html web sahifasining yaratilish kodi:

```
using KafedraBayonnoma.Models;
using System;
using System.Collections;
using System.Collections.Generic;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.Mvc;
namespace KafedraBayonnoma.Controllers
{
    public class HomeController : Controller
    {
        private bayonnomaEntities _db = new bayonnomaEntities();
        //
        // GET: /Home/
        public ActionResult Index()
        {
            ViewBag.FakultetId = new SelectList(_db.Fakultet, "FakultetId", "Fnom");
            return View();
        }
        public JsonResult KafedraList(int Id)
        {
            var kafedra = from s in _db.Kafedra
                           where s.FakultetId == Id
                           select s;

            return Json(new SelectList(kafedra.ToArray(), "KafedraId", "Knom"),
                JsonRequestBehavior.AllowGet);
        }
    }
    #region Fakultet
```

```

public ActionResult fakultet()
{
    DataManagerAction db = new DataManagerAction();
    ViewBag.Model = db.GetFacultet();
    return View();
}

[HttpGet]
public ActionResult FakEdit(int? id)
{
    DataManagerAction db = new DataManagerAction();
    return View(db.GetFaculteId(id));
}

[HttpPost]
public ActionResult FakEdit(Fakultet obj)
{
    if (ModelState.IsValid)
    {
        DataManagerAction db = new DataManagerAction();
        db.SaveFakultet(obj);
        return RedirectToAction("Fakultet");    }
    return View();
}

public ActionResult FakDelete(int id)
{
    DataManagerAction db = new DataManagerAction();
    db.DelFakultet(id);
    return RedirectToAction("Index");    }

[HttpGet]
public ActionResult FakQushish()
{
    return View();    }

```

```

[HttpPost]
public ActionResult FakQushish(Fakultet fakultet)
{
    DataManagerAction db = new DataManagerAction();
    db.CreateFakultet(fakultet);
    return RedirectToAction("Index");    }
#endregion

#region Kafedra
public ActionResult kafedra(int? id)
{
    DataManagerAction db = new DataManagerAction();
    if (id != null)
    {
        ViewBag.Model = db.GetKafedra1(id);
        return View();    }
    ViewBag.Model = db.GetKafedra();
    return View();
}

[HttpGet]
public ActionResult kafEdit(int? id)
{
    DataManagerAction db = new DataManagerAction();
    SelectList fakultetlist = new
SelectList((System.Collections.IEnumerable)(db.GetFacultet()), "FakultetId",
"Fnom");
    ViewBag.fakultetlist = fakultetlist;
    return View(db.GetKafedraId(id));
}

[HttpPost]

```

```

public ActionResult kafEdit(Kafedra obj)
{
    if (ModelState.IsValid)
    {
        DataManagerAction db = new DataManagerAction();
        db.SaveKafedra(obj);
        return RedirectToAction("kafedra");    }
    return View();
}

public ActionResult Delete(int id)
{
    DataManagerAction db = new DataManagerAction();
    db.DelKafedra(id);
    return RedirectToAction("Index");
}

[HttpGet]
public ActionResult KafQushish()
{
    DataManagerAction db = new DataManagerAction();
    SelectList fakultetlist = new
SelectList((System.Collections.IEnumerable)(db.GetFacultet()), "FakultetId",
"Fnom");
    ViewBag.fakultetlist = fakultetlist;
    return View();
}

[HttpPost]
public ActionResult KafQushish(Kafedra kafedra)
{
    DataManagerAction db = new DataManagerAction();
    db.CreateKafedra(kafedra);
    return RedirectToAction("Index");
}

```

```

    }
#endregion
#region KafedraRejalari
public ActionResult kafedrareja(int? id)
{
    DataManagerAction db = new DataManagerAction();
    if (id != null)
    {
        ViewBag.Model = db.GetKafedraReja1(id);
        ViewBag.id1 = id;
        return View();
    }
    ViewBag.Model = db.GetKafedraReja();
    return View(); }
public ActionResult KrejaPartial(int? id)
{
    DataManagerAction db = new DataManagerAction();
    if (id != null)
    {
        ViewBag.Model = db.GetKafedraReja1(id);
        ViewBag.id1 = id;
        var model = _db.Kafedra.Where(p => p.KafedraId ==
id).Select(p=>new {knom = p.Knom}).FirstOrDefault();
        ViewBag.id11 = model.knom;
        return PartialView();
    }
    return RedirectToAction("Index");
} [HttpGet]
public ActionResult KafRejaQushish(int? id)
{
    DataManagerAction db = new DataManagerAction();
    ViewBag.id = id;
    return View();
}

```

```

[HttpPost]
public ActionResult KafRejaQushish(KafedraReja kafedrareja)
{
    DataManagerAction db = new DataManagerAction();
    db.CreateKafedraReja(kafedrareja);
    return RedirectToAction("Index");
}

[HttpGet]
public ActionResult kafRejaEdit(int? id)
{
    DataManagerAction db = new DataManagerAction();
    SelectList kafedra = new
SelectList((System.Collections.IEnumerable)(db.GetKafedra()), "KafedraId",
"Knom");
    ViewBag.kafedralist = kafedra;
    return View(db.GetKafedraRejaId(id));
}

[HttpPost]
public ActionResult kafRejaEdit(KafedraReja kafedrareja)
{
    if (ModelState.IsValid)
    {
        DataManagerAction db = new DataManagerAction();
        db.SaveKafedraReja(kafedrareja);
        return RedirectToAction("KafedraReja");
    }
    return View();
}

public ActionResult kafRejaDelete(int id)
{
    DataManagerAction db = new DataManagerAction();
    db.DelKafedraReja(id);
    return RedirectToAction("Index");
}

```

```

}

#endregion

#region Masalalar

public ActionResult Masala(int? id, string id1, string id2)
{
    //kafedra nomini olish
    ViewBag.kafedra = id1;
    //kafedrarejani id si olish uchun
    ViewBag.KrejaId = id;
    Session["Knom"] = id1;
    Session["Kafedra"] = id2;
    Session["KrejaId"] = id;
    DataManagerAction db = new DataManagerAction();
    if (id != null)
    {
        ViewBag.Model = db.GetMasala1(id);
        return View();
    }
    ViewBag.Model = db.GetMasala();
    return View();
}

[HttpGet]
public ActionResult KafMasalaQushish(int? id)
{
    DataManagerAction db = new DataManagerAction();
    //KrejaId parametr sifatida
    ViewBag.id = id;
    return View();
}

[HttpPost]
public ActionResult KafMasalaQushish(Masala masala)
{
    DataManagerAction db = new DataManagerAction();
    db.CreateMasala(masala);

```

```

        return RedirectToAction("KafedraReja");
    }
    public ActionResult MasalaDelete(int id)
    {
        DataManagerAction db = new DataManagerAction();
        db.DelMasala(id);
        return RedirectToAction("KafedraReja");
    }
    //MasalaEdit
[HttpGet]
public ActionResult MasalaEdit(int? id, int? id1)
{
    DataManagerAction db = new DataManagerAction();
    ViewBag.KrejaId = id1;
    return View(db.GetMasalaId(id));
}
[HttpPost]
public ActionResult MasalaEdit(Masala masala)
{
    if (ModelState.IsValid)
    {
        DataManagerAction db = new DataManagerAction();
        db.SaveMasala(masala);
        return RedirectToAction("KafedraReja");
    }
    return View();
}
}
#endregion

#region Ustoz
public ActionResult Ustoz(int? Kafedra, int? Fakultet)
{
    DataManagerAction db = new DataManagerAction();

```

```

        ViewBag.FakultetId = new
SelectList((System.Collections.IEnumerable)db.GetFakultet(), "FakultetId",
"Fnom");

        ViewBag.KafedraId = new
SelectList((System.Collections.IEnumerable)db.GetKafedra(), "KafedraId",
"Knom");

        if (Kafedra != null)
            return View(Kafedra);
        return View();
    }

    public ActionResult UstozPartial(int? id)
    {
        bayonnomaEntities _db = new bayonnomaEntities();
        var model1 = _db.Ustoz.ToList();
        if (id != null)
        {
            var model = _db.Ustoz.Where(e => e.KafedraId == id);
            return PartialView(model);
        }
        return PartialView(model1);
    }
    [HttpGet]
    public ActionResult UstozQushish()
    {
        DataManagerAction db = new DataManagerAction();
        SelectList Kafedralist = new
SelectList((System.Collections.IEnumerable)(db.GetKafedra()), "KafedraId",
"Knom");

        ViewBag.FakultetId = new SelectList(_db.Fakultet, "FakultetId", "Fnom");
        return View();
    }

```

```

    }
    [HttpPost]
    public ActionResult UstozQushish(Ustoz ustoz)
    {
        if (ModelState.IsValid)
        {
            DataManagerAction db = new DataManagerAction();
            db.CreateUstoz(ustoz);
        }
        return RedirectToAction("Ustoz");
    }
    [HttpGet]
    public ActionResult UztozEdit(int? id)
    {
        DataManagerAction db = new DataManagerAction();
        SelectList kafedralist = new SelectList(db.GetKafedra(), "KafedraId",
"Knom");
        ViewBag.kafedralist = kafedralist;
        SelectList fakultetlist = new SelectList(db.GetFacultet(), "FakultetId",
"Fnom");
        ViewBag.fakultetlist = fakultetlist;
        return View(db.GetUstozId(id));
    }
    [HttpPost]
    public ActionResult UztozEdit(Ustoz ustoz)
    {
        if (ModelState.IsValid)
        {
            DataManagerAction db = new DataManagerAction();
            db.SaveUstoz(ustoz);
            return RedirectToAction("Ustoz");
        }
        return View();
    }
    public ActionResult UstozDelete(int id)
    {
        DataManagerAction db = new DataManagerAction();

```

```

        db.DelUstoz(id);
        return RedirectToAction("Ustoz");
    }
#endregion

#region Muhokama

public ActionResult Muhokama(int? id)
{
    DataManagerAction db = new DataManagerAction();
    Session["MasalaId"] = id;
    if (id != null)
    {
        ViewBag.Model = db.GetMuhokama1(id);
        return View();
    }
    ViewBag.Model = db.GetMuhokama();
    return View();
}

[HttpGet]
public ActionResult MuhokamaQushish(int? id,int? id1)
{
    //ViewBag.MasalaId = id;
    Session["KafedraId"] = id1;
    ViewBag.KafedraId = id1;
    SelectList ustozlist = new SelectList(_db.Ustoz, "UstozId", "Unom");
    ViewBag.ustozlist = ustozlist;
    return View();
} [HttpPost]
public ActionResult MuhokamaQushish(Muhokama muhokama)
{

```

```

        if (ModelState.IsValid)
        {
            DataManagerAction db = new DataManagerAction();
            db.CreateMuhokama(muhokama);
        }
        if (Session["MasalaId"] != null)
        {
            return RedirectToAction("Muhokama/" + Session["MasalaId"]);
            return RedirectToAction("Masala");
        }
        public ActionResult MuhokamaDelete(int id)
        {
            DataManagerAction db = new DataManagerAction();
            db.DelMuhokama(id);
            if (Session["MasalaId"] != null && Session["KafedraId"] != null)
            {
                return RedirectToAction("Muhokama/" + Session["MasalaId"] + "/" +
Session["KafedraId"]);
            }
            return RedirectToAction("Masala");
        }
        [HttpGet]
        public ActionResult MuhokamaEdit(int? id, int? id1)
        {
            DataManagerAction db = new DataManagerAction();
            SelectList Ustozlist = new SelectList(db.GetUstoz(), "UstozId", "Unom");
            ViewBag.Ustozlist = Ustozlist;

            SelectList MasalaList = new SelectList(db.GetMasala(), "MasalaId",
"MasalaMatn",id1);
            ViewBag.MasalaList = MasalaList;
            return View(db.GetMuhokamaId(id));
        }
        [HttpPost]
        public ActionResult MuhokamaEdit(Muhokama muhokama)
        {
            if (ModelState.IsValid)
            {
                DataManagerAction db = new DataManagerAction();
            }
        }
    }

```

```

        db.SaveMuhokama(muhokama);

        return RedirectToAction("Muhokama/" + Session["MasalaId"] + "/" +
Session["KafedraId"]);
    }    return View(); }

#endregion

#region Qaror

public ActionResult QarorPartial(int? id)
{
    IQueryable<Qaror> qar = _db.Qaror.Where(s => s.MasalaId == id);
    ViewBag.Model = qar;
    ViewBag.id1 = id;
    return PartialView();
} [HttpGet]

public ActionResult QarorEdit(int? id)
{
    DataManagerAction db = new DataManagerAction();
    return View(db.GetQarorId(id));
} [HttpPost]

public ActionResult QarorEdit(Qaror qaror)
{
    if (ModelState.IsValid) {
        DataManagerAction db = new DataManagerAction();
        db.SaveQaror(qaror);
        return RedirectToAction("Masala/" + Session["KrejaId"] + "/" +
Session["Knom"] + "/" + Session["Kafedra"]);
        //return RedirectToAction("Kafedra");
        //id=kaf.KrejaId, id1=kaf.Kafedra.Knom,id2=kaf.KafedraId
    }

    return View();
} public ActionResult QarorDelete(int id) {
    DataManagerAction db = new DataManagerAction();

```

```

        db.DelQaror(id);

//return RedirectToAction("Masala");

        return RedirectToAction("Masala/" + Session["KrejaId"] + "/" +
Session["Knom"] + "/" + Session["Kafedra"]);    }

[HttpGet]
public ActionResult QarorQushish(int? id)
{
    ViewBag.id = id;
return View();
} [HttpPost]
public ActionResult QarorQushish(Qaror qaror)
{
    if (ModelState.IsValid)
    {
        DataManagerAction db = new DataManagerAction();
        db.CreateQaror(qaror);
        return RedirectToAction("Masala/" + Session["KrejaId"] + "/" +
Session["Knom"] + "/" + Session["Kafedra"]);
    }
    return View();
} #endregion
}}

```

2-bobning qisqacha xulosasi. Bitiruv malakaviy ishimning ikkinchi bobi yuqorida ta’kidlab o’tilganidek, “Universitet informatsion tizimida kafedra yig’ilishlari monitoringi modulini yaratish” deb nomlanadi. Bu bobning o’zi ham uchga bo’linadi. Birinchi qismi “Kafedra yig’ilishlari monitoringi spetsifikatsiyasi” deb nomlanadi. Bu qismda kafedra yig’ilishlari monitoringi jarayoni mobaynida qanday ishlar qilinishi haqida aytib o’tilgan. Ikkinch qismi “Kafedra yig’ilishlari monitoringini dastur loyihasi” deb nomlanib, u ikkiga bo’lingan. Birinchi - “ER model – kafedra yig’ilishlari monitoringi uchun zarur bo’lgan berilganlar bazasini tashkil etish” deb nomlanib, bunda universitet tizimidagi kafedra yig’ilishlari monitoringi uchun zaruriy jadvallarning diagrammasi ko’rsatilgan. Ikkinchi –

“Foydalanuvchi grafik interfeysi” deb nomlanib, bunda kafedra yig’ilishlari monitoringi berilishini avtomatlashtirish uchun tayyorlangan web sahifalar ko’rinishi ko’rsatilgan va undagi obyektlar vazifasi izohlangan. Bobning uchinchi qismi “Asosiy modullar va amalga oshirish” deb nomlanadi. Bu qismda malaka oshirish monitoringi web saytidan foydalanish uchun asosiy yo’riqnoma ko’rsatilgan.

Xotima

Ushbu bitiruv malakaviy ishining mavzusi “Universitet informatsion tizimida kafedra yig’ilishlari monitoringini modulini yaratish” deb nomlanib, u kirish qismi, ikkita bob, boblar xulosalari, xotima va foydalanilgan adabiyotlardan iborat.

Birinchi bobda asosan “Web texnologiyalari va BBBTlardan foydalanish” haqida atroflicha to’xtalib o’tilgan. Bunda bob uch qismga bo’lingan bo’lib, birinchi qismi “HTML tili”, ikkinchi qismi “ASP NET MVC texnologiyasi”, uchinchi qismi “SQL tili”. Birinchi qismda HTML tili, uning asosiy teglari va bu teglarning asosiy atributlari, forma elementlari imkoniyatlari ko’rsatib o’tilgan. Ikkinchi qismda ASP NET MVC texnologiyasi, undan foydalanish, massivlardan foydalanish, funksiyalar bilan ishlash, ASP NET MVC va SQL Serverni bog’lash imkoniyatlari ko’rsatib o’tilgan. Uchinchi qismda asosan berilganlar bazasi, bog’lanishlar tavsifi va turlari, DML, DDL, DCL imkoniyatlari ko’rsatib o’tilgan.

Ikkinchi bob “Universitet informatsion tizimida kafedra yig’ilishlari monitoringi modulini yaratish” deb nomlanib, bu bobning o’zi ham uchga bo’linadi. Birinchi qismi “Kafedra yig’ilishlari monitoring spetsifikatsiyasi ” deb nomlanadi. Bu qismda Kafedra yig’ilishlari monitoringini jarayoni haqida aytib o’tilgan. Ikkinchi qismi “Kafedra yig’ilishlari monitoring Dastur loyihasi” deb nomlanib, u ham ikkiga bo’lingan. Birinchi - “ER model – kafedra yig’ilishlari monitoringi uchun zarur bo’lgan berilganlar bazasini tashkil etish” deb nomlanib, bunda universitet tizimidagi kafedra yig’ilishlari monitoringini uchun zaruriy jadvallarning diagrammasi ko’rsatilgan. Ikkinchi – “Foydalanuvchi grafik interfeysi” deb nomlanib, bunda kafedra yig’ilishlari monitoringini avtomatlashtirish uchun tayyorlangan web sahifa ko’rinishi ko’rsatilgan va undagi obyektlar vazifasi izohlangan. Uchinchi – “Asosiy modullar va amalga oshirish” deb nomlanib, bunda kafedra yig’ilishlari monitoringini asosiy modullari va

kafedra yig'ilishlari monitoringi web sayti yaratilishining asosiy kodlari keltirilgan.

Foydalanilgan adabiyotlar

1. Karimov.I.A. "Barkamol avlod orzusi", Toshkent 1999yil, 99-102-betlar.
2. O'zbekiston Respublikasining "Ta'lim to'g'risida"gi qonuni – T.: 1997 y.
3. O'zbekiston Respublikasining "Kadrlar tayyorlash to'g'risida"gi qonuni – T 1997 y.
4. Sayfiyev J.F. "SQL tiliga kirish" uslubiy qo'llanma. Buxoro 2005 yil.
5. Гайдар Магдануров." Разработка веб-приложений с использованием ASP.NET MVC Framework".
6. SH. P. Boboqulov, I. I. Bakayev "ASP NET MVC 4" o'quv qo'llanma. BUXORO – 2015.
7. Глушаков С.В., Жакин И. А., Хачиров Т. С. Программирование Web страниц – Харьков: Фолио, 2005. (Учебный курс).
8. Мержевич В.В. HTML и CSS на примерах. – СПб.: БХВ-Петербург, 2005.
9. Муссиано Ч., Кеннеди Б. HTML и XHTML. Подробное руководство, 6 е издание. – Пер. с англ. – СПб: Символ Плюс, 2008.
10. Zakirova T.A., Ibragimov E.U., Masharipov A.K., Musaeva M.A. "Web dasturlash". O'quv qo'llanma – Toshkent 2010 yil.
11. Мальчук Е.В. HTML и CSS. Самоучитель – М.: Издательский дом «Вильямс», 2008.
12. Мейер Э. CSS – каскадные таблицы стилей. Подробное руководство, 3 – е издание. – Пер. с англ. Символ-Плюс, 2008.
13. Хомоненко А. Д., Цыганков В. М., Мальцев М.Г." Базы данных: Учебник для высших учебных заведений "/ КОРОНА –Век, 2009.

Internet manzillar:

14. www.ziynet.uz

