

O‘ZBEKISTON RESPUBLIKASI  
OLIY VA O‘RTA MAXSUS TA’LIM VAZIRLIGI

FARG‘ONA DAVLAT UNIVERSITETI

FIZIKA-MATEMATIKA FAKULTETI

Informatika o‘qitish metodikasi yo‘nalishi  
14.06-guruh bitiruvchisi Xo‘jaqulov Ixtiyorjon Xolmatjon  
o‘g‘lining

**“Fizika-Matematika fakulteti Axborot texnologiyalari  
kafedrasi rasmiy web sahifasini less va bootstrap  
texnologiyalarida yaratish”**

mavzusidagi

# **BITIRUV**

# **MALAKAVIY ISHI**

Ilmiy raxbar: Axborot texnologiyalari  
kafedrasi o‘qituvchisi X. Shermatova

Farg‘ona – 2018

Bitiruv malakaviy ish kafedraning 2018 yil \_\_\_\_\_dagi \_\_\_\_\_ -  
sonli yig'ilishida muxokama qilingan va himoyaga tavsiya etilgan.

Kafedra mudiri \_\_\_\_\_ I. Xaydarov

### **Taqrizchilar**

1. FVXTXQTMOI

“Tabiiy va aniq fanlar ta’limi ”

kafedrasi o‘qituvchisi.:

Z. Dexqonova

2. Farg‘ona Davlat Unversiteti

Axborot texnologiyalari kafedrasi

o‘qituvchisi

X.Shermatova

## **Mundarija:**

Kirish .....	4
I BOB. Masalaning qo‘yilishi va yechilish mexanizmi .....	10
1.1. Masalani qo‘yilishi.....	10
1.2. Masalani yechish mexanizmi .....	19
II BOB. Berilganlar bazasini tashkil qilish va uni boshqarish tizimi .....	33
2.1. Berilganlar bazasi tashkil qilish.....	33
2.2. Berilganlar bazasi tashkil qilishdagi asosiy talablar .....	33
2.3. Tarmoqda ma’lumotlar bazalari arxitekturasini .....	39
III BOB. Avtomatlashgan kafedra tizimining ishlash tamoili .....	43
3.1. Berilganlar bazasidagi jadvallar sxemasi .....	43
3.2. Tizimning tavsifi .....	46
3.3. Tizimning qulayligi va avzalligi. ....	52
Xulosa va takliflar .....	54
Fodalanilgan adabiyotlar .....	56
Ilova .....	57

## **Kirish**

*Umrning butun ma`nosi mavhumlikni  
to`xtovsiz zabt etish, tobora va  
hamisha ko`proq bilishga intilishdir.*

**Emil Zolya**

O`zbekiston Respublikasi Oliy Majlisi Qonunchilik palatasining Axborot va kommunikatsiya texnologiyalari masalalari qo`mitasi O`zbekiston Respublikasi Aloqa, axborotlashtirish va kommunikatsiya texnologiyalari davlat qo`mitasi bilan hamkorlikda «Dasturiy mahsulotlar bozorini rivojlantirishga doir qonunchilik bazasini takomillashtirishning ustuvor yo`nalishlari» mavzusida davra suhbatini tashkil etdi. Unda O`zbekiston Respublikasi Oliy Majlisi Qonunchilik palatasining deputatlari, Vazirlar Mahkamasi, manfaatdor vazirlik va idoralar mas'ul xodimlari, dasturiy mahsulotlar ishlab chiqaruvchi kompaniyalar vakillari, axborot texnologilari sohasining mutaxassislari, dasturchilar, ommaviy axborot vositalari va fuqarolik jamiyati institutlarining vakillari ishtirok etdilar. Davra suhbatidan ko`zlangan maqsad-“Axborotlashtirish to`g`risida”gi O`zbekiston Respublikasi Qonunida belgilangan dasturiy mahsulotlar ishlab chiqarishni rag`batlantirish va rivojlantirish bo`yicha normalar ijrosini tahlil qilish, axborot texnologiyalari bozorining muhim qismi bo`lgan dasturiy ta'minot industriyasini davlat tomonidan ko`llab-kuvvatlashning optimal yo`llarini muhokama qilish va mamlakatimiz dasturiy mahsulotlar industriyasini rivojlantirishning qonunchilik asoslarini takomillashtirish yuzasidan aniq taklif va tavsiyalar ishlab chiqishdan iborat edi. O`zbekiston Respublikasi Prezidentining mamlakatimizda yaqin istiqbolda axborotlashtirish jarayonlarini rivojlantirish borasida aniq maqsadli yo`nalishlarni belgilab bergan «Zamonaviy axborot-kommunikatsiya texnologiyalarini yanada keng joriy etish va rivojlantirish chora-tadbirlari to`g`risida»gi 2012 yil 21 martda qabul qilgan qaroridan kelib chiqsak, ushbu tadbirning dolzarbligi yanada ayon bo`ladi. Darhaqiqat, ayni kunda dasturiy mahsulotlar ishlab chiqarish butun

jahonda alohida ahamiyat kasb etayotir. Zotan, dasturiy mahsulotlar axborotkommunikatsiya sohasini rivojlantirishning asosi hisoblanadi, axborot xavfsizligini ta'minlashga ko'maklashadi, milliy iqtisodiyotlar innovatsion rivojlanishining juda muhim sharti bo'lib, globallashtirish sharoitlarida mamlakatning raqobatbardoshligini ta'minlaydi. Davra suhbatida ta'kidlanganidek, mustaqillik yillarida mamlakatimizda dasturiy ta'minot industriyasini rivojlantirishda sifat jihatdan yuksalishni amalga oshirish uchun qulay shart-sharoitlar yaratildi. Sohani rivojlantirishning mustahkam qonunchilik bazasi yaratildi. Xususan, shu davr ichida sohaga tegishli 10 dan ortiq Qonunlar, mamlakat Prezidenti Farmoni va Vazirlar Mahkamasining qator qarorlari qabul qilindi. Keyingi yillarda Parlament tomonidan amaldagi bir qator qonunlarga zamonaviy axborot va kommunikatsiya texnologiyalarini va dasturiy mahsulotlar bozorini yanada rivojlantirishga qaratilgan qo'shimcha va o'zgartishlar kiritildi. "Axborotlashtirish to'g'risida"gi, "Telekommunikatsiyalar to'g'risida"gi, "EHM dasturlari va ma'lumotlar bazasini himoya qilish to'g'risida"gi Qonunlar shular jumlasidandir. Ta'kidlash kerakki, qabul qilingan normativ-xukuriy hujjatlarga muvofiq, dasturiy mahsulot ishlab chiqaruvchi korxonalariga soliq va bojxona sohasida katta imtiyozlar berildi. Chunonchi, bir necha yillar davomida kompyuter va axborot texnologiyalari bo'yicha o'qitish xizmatlarini, shuningdek, dasturiy vositalarni realizatsiya qilishni qo'shimcha qiymat solig'idan ozod etish bo'yicha normalari amal qildi. Bundan tashqari, dasturiy mahsulotlar ishlab chiqarishga ixtisoslashgan korxonalar mikrofirmalar va kichik korxonalar uchun umumiy belgilangan yagona soliq to'lovi stavkasidan kamaytirilgan soliq to'laydilar. Muhokamada ta'kidlanganidek, hozirgi kunda davlat hokimiyati va boshqaruvi organlari dasturiy mahsulotlarning faol iste'molchisi hisoblanadi. Keyingi vaqtda mamlakatimizda vazirliklar, idoralar, xo'jalik birlashmalari va yirik korxonalarining xarajatlar smetasi va biznes rejalarida ishlab chiqarish jarayonlarini avtomatlashtirish, axborot tizimlari va resurslarini yaratish, dasturiy mahsulotlar hamda interfaol davlat xizmatlarini joriy etishga qaratilgan xarajatlarni nazarda tutish bo'yicha tegishli ishlar olib borilmoqda. Joriy yilda mamlakatimizda 5 davlat hokimiyati va boshqaruvi

organlarining idoralararo integratsiyalashgan axborot tizimlarini joriy etish, ma'lumotlar uzatishning idoralararo tarmoqlarini yaratish, «Elektron hukumat» tizimini hamda Internet tarmog'ining milliy segmentini shakllantirish borasidagi loyihalarni ro'yobga chiqarish uchun davlat buyurtmasini shakllantirish va moliyalashtirish ishlari boshlab yuborildi. Tabiiyki, ushbu chora-tadbirlar dasturiy mahsulotlar industriyasining milliy bozori ko'lamini kengaytirishga ijobiy ta'sir ko'rsatib, uning jadal rivojlanishini rag'batlantirmoqda. Natijada mahalliy dasturiy ta'minot ishlab chiquvchilari mahsulotlari hajmi yil sayin kengayib, xorijiy dasturiy mahsulotlar bilan raqobatlashishga qodir yangi dasturiy mahsulotlar ishlanmalari yaratilmoqda. Davra suhbatida axborot texnologiyalari va dasturiy ta'minot bozorini rivojlantirishga qaratilgan qonun ijodkorligi ishlarini ustuvor yo'nalishlari bilan bir qatorda amaldagi qonun hujjatlari ijrosining borishi, dasturiy mahsulotlar ishlab chiqarishning bugungi holati va ularni rivojlantirish istiqbollari ham muhokama qilindi. O'zbekiston Respublikasi Qonunchilik palatasidagi siyosiy partiyalar fraksiyalarining deputatlari o'z partiyalarining ustav qoidalari va saylovoldi dasturlardan kelib chiqqan holda, davra suhbatida muhokama qilingan masalalar yuzasidan o'z fikr-mulohazalarini bildirdilar. Suhbat yakunlariga ko'ra, dasturiy mahsulotlarni yaratish, tarqatish va joriy etish sohasidagi munosabatlarni tartibga solish, milliy dasturiy mahsulotlar industriyasini rivojlantirish va uning eksport salohiyatini oshirish maqsadida qulay iqtisodiy muhitni yaratish uchun qonunchilik bazasini takomillashtirishga qaratilgan bir qator amaliy taklif va tavsiyalar ishlab chiqildi. Bugungi kunda tashkilot qanday bo'lishidan qat'iy nazar ular moliyaviy bo'ladimi, savdo tashkilotlari, sanoat bo'ladimi barchasining faoliyatini berilganlar bazasisiz tasavvur qilib bo'lmaydi. Agar berilganlar bazasi bo'lmaganida ular axborot oqimi ichida qolib ketar edilar

Hozirgi davrda sohani kompyuter, aloqa, texnika-texnologiya, internet tarmog'isiz tasavvur qilib bo'lmaydi. Shuning uchun bu sohalar jahon bozorida o'z o'rnini topishi egallashi uchun texnika olami bo'lmish bu davrda internet tarmo'gida o'z sahifalariga ega bo'lishlari kerak. Ayniqsa bugungi kunda rivojlangan mamlakatlar oliy ta'limida internet saytining o'rnini beqiyos. Bugungi

kunda O‘zbekistonda ham ta’lim muassasalari o‘zlarining internet saytlarini yaratishga harakat qilmoqdalar va yaratmoqdalar.

O‘zbekistonda axborotlashtirishni rivojlantirish, jamiyat taraqqiyotining barcha sohalarida zamonaviy axborot texnologiyalari, texnika va telekommunikatsiya vositalarini ommaviy ravishda joriy etish hamda ulardan foydalanish, fuqarolarning axborotga bo‘lgan talab-ehtiyojlarini yanada to‘liqroq qondirish, jahon axborot resurslaridan bahramand bo‘lishni kengaytirish uchun qulay shart sharoitlarni yaratish aloqa va axborot kommunikatsiyasi sohasini rivojlantirish asosiy vazifalari hisoblanadi.

Shuni aytish zarurki bugungi kunda axborot, axborot resurslari, kompyuter tarmoqlari va turli korporativ tizimlarga nisbatan tahdidlarni oldini olish asosi dolzarb masalalardan biri hisoblanib kelmoqda. Chunki, har qanday korxonaning samarali faoliyat ko‘rsatishida faqatgina moddiy, tabiiy, moliyaviy resurslarning o‘zi yetarli emas. Ulardan qanday qilib samarali, oqilona foydalanish uchun o‘sha sohadagi texnologiyalar haqida yetarlicha axborot bo‘lishi talab etiladi. Mavjud bo‘lgan axborot esa qolgan barcha resurslardan samarali foydalanish uchun keng imkoniyatlar yaratib beradigan yagona resurs hisoblanadi. Axborotlar saqlanayotgan berilganlar bazasidagi ma’lumotlar maxfiy bo‘lishi talab etiladi. Maxfiylik deganda alohida shaxslar yoki tashkilotlarning boshqa shaxslar yoki tashkilotlarga qachon, qancha miqdorda ma’lumotlar berilgani umumiy qilib aytganda qachon va qancha miqdorda ma’lumotlar almashinganligini aniqlash huquqi tushiniladi. Saqlanayotgan axborot undan foydalanilayotgan tashkilot uchun muhim hisoblanadi. U ya’ni axborot o‘g‘irlanmasligi va yo‘qotilmasligi kerak. Shuning uchun ham deyarli barcha xorij mamlakatlarida, balki respublikamizda ham axborot tizimlarning muhofazasiga jiddiy e’tibor berilmoqda.

**Bitiruv malakaviy ishining dolzarbligi:** Birinchi Prezidentimiz I. A. Karimov tashabbusi bilan ishlab chiqilib, Oliy Majlisning IX sessiyasida qabul qilingan “Kadrlar tayyorlash milliy dasturi”, “Ta’lim to‘g‘risidagi qonun”, Vazirlar Mahkamasining umumiy o‘rta ta’lim, akademik litseylar va kasb-hunar kollejarini

tashkil etish haqidagi va boshqa qarorlari shu maqsadlarni ro'yobga chiqarishga qaratilgan.

“Kadrlar tayyorlash milliy dasturi”ni ro'yobga chiqarishning birinchi bosqichida (1997-2001 yillar) – “o'quv-uslubiy majmualarning hamda ta'lim jarayonini didaktik va axborot ta'minotining yangi avlodini ishlab chiqish va joriy etish”; ikkinchi bosqich (2001-2005 yillar)da – “ta'lim muassasalarining moddiy-texnika va axborot bazasini mustahkamlashni davom ettirish, o'quv-tarbiya jarayonini yuqori sifatli o'quv adabiyotlari va ilg'or pedagogik texnologiyalar bilan ta'minlash”; uchinchi bosqichi (2005 va undan keyingi yillar)da – “ta'lim muassasalarining resurs, kadrlar va axborot bazalarini yanada mustahkamlash, o'quv-tarbiya jarayoni yangi o'quv-uslubiy majmualar, ilg'or pedagogik texnologiyalar bilan to'liq ta'minlanishi” dolzarb vazifalar qatorida belgilangan.

Milliy dasturni ro'yobga chiqarishning yuqori sifat ko'rsatkichini ta'minlash, ta'lim mazmunini Davlat ta'lim standartlaridagi talablari amalga oshirildi. Barcha o'quv fanlari bo'yicha Davlat ta'lim standartlarini o'quv jarayonida qo'llab va o'quv yili yakunida o'quvchilar tomonidan o'zlashtirilgan bilimlar shu standartlarga mosligini aniqlash bo'yicha maktab, tuman, shahar, respublika darajasida monitoring ishlari olib borilmoqda. Milliy dasturda ko'zda tutilgandek, zamonaviy axborot texnologiyalari va kompyuterlar tarmoqlari negizida ta'lim jarayonini axborot bilan ta'minlash rivojlanib boradi. O'quv, o'quv-uslubiy, ilmiy, innovatsion, zamonaviy texnologiyalar mukammalashib, o'quv jarayoniga tatbiq etib boriladi.

Hozirgi kunda zamonaviy dastur va texnologiyalarga extiyoj ko'payib bormoqda, shu asosda Farg'ona Davlat Universiteti Fizika-matematika fakulteti Axborot texnologiyalari kafedrasida onlayn tizim ya'ni web sahifa yaratish zarur va bu orqali bir muncha afzalliklariga erishish mumkin. Mazkur bitiruv malakaviy ishi loyihasida **Fizika-Matematika fakulteti Axborot texnologiyalari kafedrasi rasmiy web sahifasini Bootstrap, Less, PHP, Javascript, jQuery va Css texnologiyalari** asosida web sahifa yaratilsa maqsadga muvofiqdir. **Axborot texnologiyalari kafedrasi rasmiy web sahifasini** yaratish jarayonida, yangi

texnologiyalar ushbu web sahifani ish unumdorligini ancha yuqori darajaga ko'taradi. Ushbu web sahifani qulaylik va afzalliklari shundaki, Fakultet va Kafedra miqyosidagi yangiliklar o'zgarishlar e'lon qilib boriladi va bundan tashqari qulayliklarni yaratish maqsadida, kafedra haqida umumiy ma'lumot olish, davomat tizimini onlayn tarzda ish faoliyatini yuritish, kafedra xodimlari va talabalari tomonidan chiqarilgan maqola va tezislarni taqdim etish, kafedra xodimlari haqida umumiy malumotlar olinishi va yana bir qancha foydali vazifalari bajarilishi ko'zda tutilgan.

**Bitiruv malakaviy ishining maqsadi:** Fizika-Matematika fakulteti Axborot texnologiyalari kafedrasini rasmiy web sahifasini Bootstrap, Less, PHP, Javascript, jQuery va Css web texnologiyalari asosida shakllantirish va ushbu web sahifada bir qancha qulaylik va afzalliklarni yaratish, bularga misol qilib, Fakultet va Kafedra miqyosidagi yangiliklar o'zgarishlar e'lon qilib borilish, kafedra haqida umumiy ma'lumotlar olish, davomat tizimini onlayn tarzda ish faoliyatini yuritish, kafedra xodimlari va talabalari tomonidan chiqarilgan maqola va tezislarni taqdim etish, kafedra xodimlari haqida umumiy malumotlar olinishi va yana bir qancha foydali vazifalari ko'rishimiz mumkin.

**Bitiruv malakaviy ishining ob'ekti:** Fizika-Matematika fakulteti Axborot texnologiyalari kafedrasini rasmiy web sahifasini Bootstrap, Less, PHP, Javascript, jQuery va Css web texnologiyalari asosida yaratish jarayoni.

**Bitiruv malakaviy ishining predmeti:** Web sahifasini yaratishda yangi zamonaviy web texnologiyalar Bootstrap, Less, PHP, Javascript, jQuery va Css texnologiyalardan foydalanish.

**Bitiruv malakaviy ishining vazifalari:**

1. Mavzuga doir ma'lumotlarni yig'ish va rejani shakllantirish;
2. Web sahifani yaratishda ish samaradorligini oshirish;
3. Web sahifani yaratish jarayonida bilim va ko'niklar olish;
4. BMIning jihozlab, uni himoyaga tayyor qilish

## **I BOB. Masalaning qo‘yilishi va yechilish mexanizmi**

### **1.1. Masalani qo‘yilishi.**

Ma’lumki, butunjahon tarmog‘ida elektron sahifalar va saytlarning turli shakllari asosidagi web-sahifalar kundan kunga ko‘payib bormoqda. Bu holat O‘zbekistonda, ya’ni Uz domenida ham jadallik bilan rivojlanib bormoqda. Kimdir o‘z shaxsiy sahifasini yaratsa, kompaniya va firmalar korporativ saytlarini, ta’lim muassasalari ta’limga oid saytlarni hamda elektron o‘quv materiallarini yaratmoqda. Yana kimdir o‘z kundaliklarini, ya’ni bloglarini yaratish ustida ish olib bormoqda.

Shu bilan birga, yirik portallarning faoliyati ham rivojlanmoqda. Bularning hammasi Web saytlarni yaratish bilan bog‘liq turli dasturlash tillari va dasturiy vositalari asosida yaratilmoqda. Web-sahifalarni yaratish bilan shug‘ullanuvchi mutaxassislarning ba’zilari HTML tilida, ba’zilari PHP tilida, yana ba’zilari Web saytlarni yaratish va tahrirlash dasturiy vositalari yordamida, ishning mohiyatidan kelib chiqib ba’zi shu sohani mutaxassislari takomillashib borayotgan CMS (Content Management System) deb ataluvchi tizimlar yordamida web-saytlar, sahifalarni yaratmoqda. O‘z-o‘zidan savol tug‘iladi, elektron sahifa yoki saytlarni yaratishda WYSIWYG (What You See Is What You Get) yoki CMS degani o‘zi nima? Bu tizimni nima ekanini tushunish uchun, avvalo, web-sahifalar qanday yaratilishi haqida to‘xtalib o‘tish lozim.

Ma’lumki, Web-texnologiyani (Internet-texnologiya) o‘rganishni Web-dizaynning quyidagi uchta tushunchalari mavjud: Web-sahifa, Web-sayt va Web-server.

Web-sahifa – o‘zining unikal adresiga ega bo‘lgan va maxsus ko‘rish dasturi yordamida (brauzer) ko‘riluvchi xujjatdir. Unga matn, grafika, ovoz, video yoki

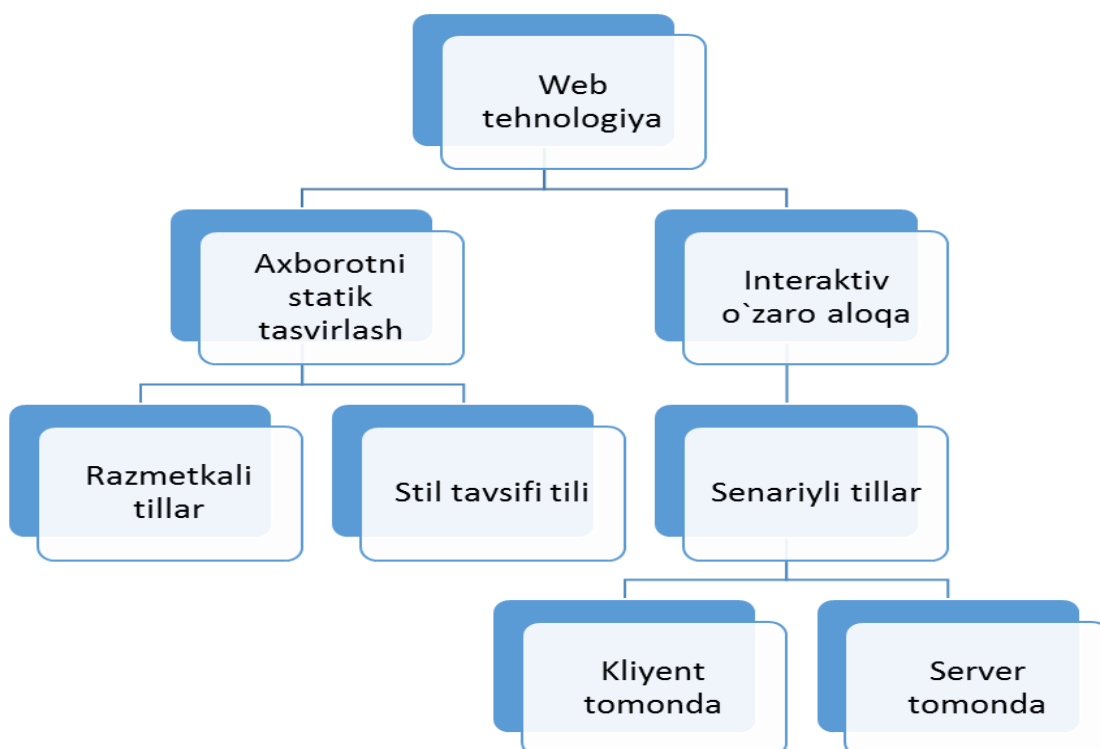
animasiya ma`lumotlar birlashmasi - multimediyali xujjatlar, boshqa hujjatlarga gipermurojaatlar kirishi mumkin.

Web-sayt – bir qancha web-sahifalarning mantiqiy birlashmasi.

Web-server – tarmoqqa ulangan kompyuter yoki undagi dastur hisoblanib, umumiy resurslarni klientga taqdim etish yoki ularni boshqarish vazifalarini bajaradi. Web-serverlar ma`lumotlar bazalari va multimediyali ma`lumotlarni bir biriga moslashtiradi. Web-serverda Web-sahifa va Web-saytlar saqlanadi.

Biz Internet tarmog`idagi Web-sahifalarni ko`rishimiz uchun WWW (World Wide Web) deb ataluvchi servisdan foydalanamiz.

World Wide Web (WWW, Butun dunyo o`rgamchak to`ri) – bu klient-server texnologiyasi asosida tashkil etilgan, keng tarqalgan Internet xizmatidir (quyida keltirilgan).



Shaxsiy elektron hisoblash mashinalarida formatlashtirilgan elektron hujjat WYSIWYG (What You See Is What You Get) «Nimani ko`rayotgan bo`lsang, o`shani olasan» prinsipida ishlaydigan matn tahrirlagichlar yordamida yaratiladi.

Masalan, bularga oddiy MS Word, Lexicon, AmiPro, WordPad va h.k. kabilarni keltirib o‘tish mumkin.

Bunday dasturlar yordamida biz elektron hujjatni xohlagan shriftda, o‘lchamda, chap yoki o‘ng tomondan tekislagan holda yaratishimiz mumkin. Ammo biz ushbu elektron hujjatni internet yordamida e‘lon qila olmaymiz. Sababi, uni o‘qimoqchi bo‘lgan boshqa bir internet mijozining shaxsiy komyuterida biz foydalangan matn tahrirlagich dasturi yoki shriftlar o‘rnatilmagan bo‘lishi yoki ishlash jarayonida turli nosozliklar kelib chiqishi mumkin. Buni oldindan aytib bo‘lmaydi. Undan tashqari ushbu matnni ochishga mo‘ljallangan «oyna»ning o‘lchamlari haqida hech qanday ma‘lumotlarga ega emasmiz. Shuning uchun ham ShEHMLlarda foydalaniladigan matn tahrirlagichlar va ularning «format»lash usullaridan internetda foydalanib bo‘lmaydi.

Bunday noqulayliklarning oldini olish maqsadida yangi HTML (Hyper Text Markup Language) «gipermatnlarni belgilash tili» protokoli, standarti yaratildi. Aslida, gipertekst g‘oyasi 1945 yilda AQSH prezidentining ilmiy maslahatchisi Vanevar Bush tomonidan kiritilgan bo‘lib, u o‘sha vaqtda “Memex” (memeks) elektromexanik informasion tizimini yaratish taklifi bilan chiqqan, lekin uning taklifi inobatga olinmagan edi. 20 yildan so‘ng, ya‘ni 1965 yilda Teodor Nelson “gipertekst” terminini fanga kiritdi va “chiziqsiz” matnlar bilan bog‘liq ba‘zi g‘oyalarni amalga oshirdi. 1968 yilda “sichqoncha” manipulyatori asoschisi D.Enjilbard (Doug Engelbart) tipik gipermatnli interfeysdan iborat (sistema) tizim asosida o‘z ishini namoyish qildi. Bu standart bir qancha maxsus operatorlar majmuasidan iborat bo‘lgan HTML dasturlashtirish tili bo‘lib, uning yordamida elektron hujjatlarni internetda bevosita e‘lon qilish mumkin edi.

1989-yilda Tim Berners-Li tomonidan World Wide Web, ya‘ni Bununjahon o‘rgimchak to‘ri ixtiro qilinganidan boshlab uning har bir foydalanuvchisi oddiy bo‘lsa-da, o‘z web-sahifasini yarata olish imkoniyatiga ega bo‘ldi. Lekin, buning uchun HTML (Hyper Text Markup Language) tilini bilish shart edi. Turli xil IT-

kompaniyalari tomonidan bu ishni osonlashtiruvchi bir qancha Microsoft FrontPage, Adobe Dreamweaver kabi dasturiy paketlar ishlab chiqildi. Bunday dasturlardan ba'zilar sayt yaratishning barcha bosqichlarini (HTML-kodni yozishdan bu kodni serverga yuklashgacha) boshqarishni o'z zimmasiga oldi. Shu bilan birga, ko'pgina hollarda (aynan oddiy sahifalarni yaratishda) foydalanuvchi bu kodni bilishi shart bo'lmagan. Sababi bu dasturlarda web-sahifalarni yaratishning vizual usullaridan foydalanilgan.

Axborot texnologiyalarining tez rivojlanish borishi natijasida HTML-redaktorlarining barcha afzalliklariga qaramay, saytni boshqarish vaqt o'tgani sayin va yangi ma'lumotlar qo'shilishi bilan qiyinlashib boraverdi. Chunki axborot texnologiyalari rivojlanishi bilan web-saytlar ham rivojlanib bordi. Sayt xaritasini yaratishdek oddiy funksiyani qo'shish uchun ham web-yaratuvchilar tomonidan ancha-muncha mehnat talab etilar edi. Axir buning uchun barcha ilovalarni ko'rib chiqish, sahifalar orasidagi bog'liqliklarni tekshirish, menyuni yaratish, forumni qo'shish kabi ishlarni bajarish lozim edi. Bu esa doimiy diqqatni va juda ko'p vaqtni talab etardi.

Endi savol tug'ilishi mumkin: oddiy HTML kodida yozilgan sahifa bilan elektron sahifa va saytlarni yaratish dasturiy vositalari nimasi bilan farq qiladi? Avvalambor, HTML kodi bilan yozilgan sahifalarni tekshirish, tahrirlash uchun juda ko'p buyruqlarni yoddan bilish, ularni formatlarini eslab qolish talab qilinadi, qolaversa har bir element uchun ham dasturlash kodlarini, ko'plab buyruqlarni bilish talab qilinadi.

Oddiy sayt qanday prinsip asosida ishlashini ko'rib chiqamiz. Foydalanuvchi biror-bir, masalan, index.html sahifasiga murojaat qildi deylik. Web-server o'zidagi web-sahifalar katalogini ko'rib chiqadi va agar index.html bo'lsa, foydalanuvchiga uni ko'rsatadi, aks holda esa – 404-chi xabarni (“Fayl topilmadi”) yetkazadi.

Elektron sahifa va saytlarni yaratish dasturiy vositalari yordamida esa, bundan kichik hatoliklarni oldi olinadi. Aytaylik, siz 100 sahifalik (bu unchalik ko'p emas) oddiy sayt yaratdingiz. Ertami-kech siz bu sayt dizaynini, ko'rinishini o'zgartirmoqchi bo'lasiz. U holda esa siz shu 100 sahifani qo'lda o'zgartirib chiqishingizga to'g'ri keladi. Ustiga-ustak, siz avvaliga bu sahifalarni kompyuterindagi lokal serverga yuklab olish, undan so'ng tahrirlab, kerakli o'zgartirishlarni kiritishingiz va oxirida yana serverga yuklash kerak bo'ladi.

Ma'lumki, HTML yaratuvchilar 2 vazifani hal qilishga uringanlar:

- Dizaynerlarga dokument yaratishning oddiy vositasi bo'lgan gipermatnli ma'lumotlar bazasini berish;
- O'sha momentda paydo bo'lgan (mavjud bo'lgan ) gipermatnli ma'lumotlar bazasidan foydalanuvchi interfeys to'g'risidagi tasavvurni tasvirlash (aks ettirish,ko'rsatish) uchun bu vositani yetarlicha qudratli qilib yaratish.

Har qanday gipermatn kitobga o'xshagan bo'ladi va uni turli tarkibiy qismlarga bo'lish mumkin bo'ladi: xususan boblarga, paragraflarga, bo'limlarga, bo'limcha va abzaslarga bo'lsa bo'ladi.

Ana endi dasturiy vositalar yordamida asosiy web sahifalarni yaratishni ko'rib ko'rib chiqaylik.

Web sahifalarni yaratish dasturiy vositalari Web-ishlanmalarni dasturlashsiz tezda yaratish va unda professional ravishda tayyorlangan hujjatlarni nashr etish imkoniyatini beradigan maxsus vosita hisoblanadi. Web sahifalarni yaratish dasturiy vositalari o'ziga xos maxsus interfeysga ega bo'lib, HTML sahifasi bilan ishlash uchun mo'ljallangan instrumentlar, shu jumladan uzoqdagi ko'p marta foydalanish rejimi va WYSIWYG rejimi ham shunday. Dasturiy vositalar tarkibiga web-sahifalar va HTML-sahifalarni interaktiv tadqiq qilishda ishlatiladigan master va shablonlar kiradi.

Elektron saytlarni yaratish dasturiy vositalari odatda, qandaydir saytning asosiy bosh sahifasini anglatadi va bu sahifalarni yaratuvchi dasturlar ham shu nom bilan ataladi. Dasturlar ishlashda juda oddiy, lekin juda ko'plab ishlarni bajarishga imkon beradi.

Bunday dasturiy vositalar sifatida Web Builder, Namo Web Editor, Web Creator, MS Frontpage, A4DeskPro, Easywebsitepro, Antenna Web Design Studio va boshqa ko'plab dasturlarni keltirib o'tish mumkin. Web sahifalarni to'g'ri ko'rish xususiyatlarini ko'rib chiqish, grafika bilan bog'liq muammolarni, saytni har xil multimediali to'ldirish variantlarini, sahifaning mazmunli qismi va uning yuklash tezligi orasidagi balansni topishga imkoniyat yaratib beradi. Web sahifalarni yaratish dasturiy vositalarida qaror va usullar to'plamini ko'rib chiqishdagi kundalik ishlarimiz ma'lumotlardan foydalanib, sanoat saytlarini bajarish mumkin. Natijada yaratiladigan sahifalarni bezashga bog'lash va u bilan distansion rejimda to'g'ri ish tashkil etish mumkin bo'ladi. Elektron tijorat uchun ham elektron qo'llanma va darsliklar yaratish uchun ham uzluksiz axborotli xavfsiz sahifalarni vositalarini tashkil qilish mumkin. Bu ishlarning barchasi dasturlashsiz olib beriladi, maxsus dasturiy vositalar yordamida olib boriladi.

Endi quyidagi elektron sahifa, qo'llanma va darsliklarni yaratishda foydalaniladigan dasturiy vositalarni qisqacha tavsiflab o'taylik:

"Web-Builder CMS" – zamonaviy yuqori funksiyalarga ega bo'lgan, Internet tarmog'ida boshqarish va intranet tarmog'ida loyihalashtirish uchun mo'ljallangan dinamik va statik saytlarni yaratish uchun mo'ljallangan tizim hisoblanadi. CMS bozorida bugungi kunda juda ko'plab dasturiy vositalar yaratilmoqdaki, ular ichida bu dasturiy vositaning imkoniyatlari juda yuqori baholanmoqda. Murojaat qilish uchun rasmiy sayti: [www.WYSIWYGWebBuilder.ru](http://www.WYSIWYGWebBuilder.ru).

Dreamweaver – HTML kodli sahifalarni yaratishda o'ta qulay hisoblangan dasturdir. Albatta o'z kuchi va vazni bilan imkoniyatlari ko'p hisoblangan boshqa dasturlar ham bor. Mana kancha yildirki HTML yaratadigan dasturchilar ikki kurinishda yaratishmokka, biri bloknotli va progressiv-xaqiqiy guruhlar faqat bloknotda yozishadi. Bu bloknotda yozish inson uchun ko'rishga oson bo'lgani

uchun yozishadi. Lekin vaqt degan muammoga duch kelishi aniq. Bu esa, minutda bajariladigan ishni soatda bajarilishiga olib keladi. Adobe Dreamweaver – esa, HTML tilini uncha yaxshi bilmaydiganlar uchun ham qulay hisoblanadi. . Murojaat qilish uchun rasmiy sayti: [www.Dreamweaver.ru](http://www.Dreamweaver.ru)

A4DeskPro – bu dasturiy mahsulot ham internet tarmog‘i uchun web sahifalar va saytlarni yaratish, elektron qo‘llanma va darsliklarni yaratish uchun mo‘ljallangan dasturiy mahsulot, ya`ni vizual redaktorlardan biri hisoblanadi. Dastur 100 dan ortiq o‘zining maxsus shablonlariga ega. Murojaat qilish uchun rasmiy sayti: [www.a4deskpro.ru](http://www.a4deskpro.ru).

Namo Web Editor – o‘zining unikal shablonlari hamda ichki redaktorlariga ega bo‘lgan kichik sahifalarni hamda elektron o‘quv materiallarini yaratishga mo‘ljallangan dasturiy vosita hisoblanadi. Murojaat qilish uchun rasmiy sayti: [www.namowebeditor.ru](http://www.namowebeditor.ru).

FrontPage – kompleks dastur bo‘lib, Web sahifalarni yaratish va tahrirlash uchun ishlatiladi. Front Page express dasturi Internet tarmog‘iga yoki Intranet sahifalari uchun yangi korporativ HTML formatidagi hujjatlarni tez va sifatli hosil qilish, ular bilan aloqalarni tekshirib ko‘rish hamda ular bilan muloqotlarni cheklashda foydalaniladi.

CoffeeCup Visual Site Designer – ushbu redaktorning ishlash prinsipi boshqa sahifa yaratuvchi dasturlar kabi bo‘lib, unda mavjud shablonlar asosida yangi sahifalarni yaratish, filmlarni, rasmlarni, animasiya, turli jadvallarni, kalendar va formalarni qo‘shish imkoniyatlari mavjud.

WebSite X5 –dastur versiyasi 2013 yildan ishlab chiqarilgan bo‘lib, uni tekin va sotuvdagi variantlari mavjud. WebSite X5 dasturida yaratiladigan Web saytni ketma-ketligi aniq ta`minlanishi lozim bo‘ladi. Bu dasturda ham tayyor shablonlardan foydalanish yoki yangi yaratish imkoniyatlari mavjud. WebSite X5 – ko‘p funksiyali dastur bo‘lib, unga turli ob`ektlarni qo‘yish yoki boshqa sahifalarni o‘rnatish imkoniyatlari mavjud. Murojaat qilish uchun rasmiy sayti: <http://htmleditors.ru/List3/WebsiteX5.html>

Xara Web Designer - juda yaxshi lekin o'rganishga biroz qiyinroq bo'lgan dasturiy vosita hisoblanadi. Dastur tayyor shablonlar asosida saytlarni yaratish, yoki alohida yangi saytni shakllantirish imkonini beruvchi maxsus redaktoriga ega. Yaratilgan sayt tayyor bo'lganidan keyin html – ga maxsus o'tkaziladi. Murojaat qilish uchun rasmiy sayti: [http://htmleditors.ru/List3/Xara\\_Web\\_Designer.html](http://htmleditors.ru/List3/Xara_Web_Designer.html)

Site Studio - bu dasturiy vosita asosan o'zida mavjud shablonlar asosida sayt yaratish imkonini beradi. Bu dasturda sayt yaratish katta mehnat talab qilmaydi, chunki bu dastur o'rganishga ham, foydalanishga ham juda oson sayt hisoblanadi. Murojaat qilish uchun rasmiy sayti: <http://htmleditors.ru/List3/SiteStudio.html>

Shunday qilib, yuqorida keltirilgan tavsiyalar asosida yaratiladigan sahifalarni tez va oson ravishda yaratish mumkin. Ammo hech qaysi dasturiy vosita tayyor holdagi saytni taqdim etmaydi. Buning uchun dasturiy vosita imkoniyatlarini bilish, harakat va mehnat talab etiladi. Yaratilayotgan sahifa boshqalarnikiga o'xshamasligi, dizayn jihatdan takrorlanmas va original bo'lishi uchun esa shaxsiy g'oyalar va qobiliyatlarni ishga solish lozim bo'ladi.

Web-texnologiyalari jadal suratda rivojlanishi, hamda tarmoq texnologiyalari rivojlanishi axborot tizimlarini shu texnologiyalardan foydalanib yaratish ommaviylashib ketgan. Qulaylik tomonlaridan biri tizimni biror serverga o'rnatib shu serverni tarmoqqa ulansa, foydalanuvchi ixtiyoriy kompyuter yoki mobil apparatlardan tizimga kirishi mumkin.

Ushbu bitiruv malakaviy ishining asosiy maqsadi: Web texnologiyalari yordamida kafedra faoliyatini avtomatlashtirish axborot tizimini yaratish.

Bitiruv malakaviy ishining asosiy vazifasi kafedra faoliyatini avtomatlashtirish va yengillashtirishdan iborat .

Kafedra tizimini yaratishdagi asosiy vazifalar quyidagilardan iborat:

1. Berilganlar bazasini yaratish;
2. Admin moduli yaratish;
3. Foydalanuvchi moduli yaratish;
4. Tizim dizaynini yaratish.

Web-texnologiya asosida yaratilgan kafedra tizimi xavfsizligini ta'minlash ham asosiy vazifalardan biri hisoblanadi. Xavfsizlikni taminlashda quyidagi masalalar yechiladi:

1. Foydalanuvchilar parolini saqlashda kriptografiyadan foydalaniladi;
2. Talaba va domlalar rollarini yaratish. Shu rollarni tizimdan o'rnini aniqlash;
3. Mahfiy foydalanuvchilarning tizimdagi o'rnini hisobga olish.

Qo'yilgan masalani yechish uchun quyidagilarni o'rganib chiqish talab qilinadi:

1. HTML, CSS, JavaScript, AJAX, jQuery, OOP, PHP;
2. MySQL;
3. Yii framework, Bootstrap;

Yuqorida keltirib o'tilgan fikrlarni mujassamlashtirgan holda **“Axborot texnologiyalari kafedrasining rasmiy Web saytni yaratish”** mavzusidagi bitiruv malakaviy ishini bajarish jarayoni amalga oshirilishi lozim bo'lgan ishlarni ketma-ketligini aniqlab, bajarilishi lozim bo'lgan ishlarni keltirib o'tsak. Bitiruv malakaviy ishini bajarishda biz quyidagi ishlarni amalga oshirishni nazarda tutamiz:

- Web saytlarni yaratish bo'yicha mavjud manbalar tahlili
- Sahifa va saytlarni yaratish va boshqarish tizimlari taraqqiyotining ustivor yo'nalishlari
- Web saytlar, ularning turlari va modellari
- Masalaning o'rganilganlik darajasi va uning qo'yilishi
- Web saytlarni yaratish vositalari va dasturlash tillari
- Dinamik va statik saytlar, ularni yaratish tillari va vositalari
- **Axborot texnologiyalari kafedrasiga** oid manbalarni shakllantirish
- Web sayt yaratish bo'yicha qo'yiladigan talablar
- **Axborot texnologiyalari kafedrasiga** Web saytni yaratish
- Web saytni yaratishning umumiy strukturasi ishlab chiqish
- Bajarilgan ishlar yuzasidan xulosa va takliflarni ishlab chiqish;

- Foydalanilgan adabiyotlar ro'yxatini keltirish hamda yaratilgan elektron sayt va ko'rinishlarini ilova bo'limida keltirib o'tishni amalga oshirish talab qilinadi.

- Yuqoridagi masalaning qo'yilishidagi fikrlardan ko'rinadiki, yangi axborot texnologiyalarini amalda qo'llash orqali bilim olish katta imkoniyatlarni ochib beradi.

## **1.2. Masalani yechish mexanizmi**

Kafedra faoliyatini avtomatlashtirish tizimini yaratishda XAMPP dasturi Windows operatsion tizimida o'rnatildi. Bu kafedra axborot tizimi php tilida yaratildi. Chunki bu tilning tushunarligi va soddaligidir. Yana shu tilda yaratilgan yii frameworkdan foydalanildi, bunga sabab bu frameworkning imkoniyatlari biz yaratgan tizimga mos. Berilganlar bazasini saqlash MySQL tizimida amalga oshirildi va server sifatida Apache server tanlandi.

### **AJAX**

Ajax texnologiyasi terminini Jesse James Garrent 2005-yil 18-fevraldagi "Web sahifalarga yangi yondashuv" nomli maqolasida birinchi bo'lib ishlatgan. Keyinchalik uni amalda qo'llagan. AJAX - Asynchronous Javascript and XML ya'ni asinxron JavaScript va XML\_.

Web texnologiyalari 2 turga bo'linadi.

- Server
- Klient

Klient qismdagi ayrim qulayliklarning ta'minlanishi ajax texnologiyasi bilan bog'liq ya'ni internet tezlik uncha yuqori bo'lmasa ham ajax texnologiyasi yordamida serverdan ma'lumotlarni olish mumkin. Agar php yordamida serverga murojaat qiladigan bo'lsak sahifalar qayta yuklanishi kerak bo'ladi. Bunda internet tezligi yuqori bo'lishi kerak va har bir harakat uchun sahifalar yangilanishini kutish kerak. Ajax texnologiyasi serverga qismga yordamchi qism hisoblanadi, ya'ni foydalanuvchining imkoniyatlarini oshiradi.

Ajax texnologiyasining eng muhim jihatlaridan biri bu sahifalarni qayta yuklamasdan serverdan ma'lumotlarni yuklab oladi.

Ajax ning 2 ta xususiyati bor:

1. Sahifalar qayta yuklanmasidan serverga borib amal bajarib qaytib kela oladi. Bunda ma'lumotlar JSON yoki XML formatida jo'natiladi va qabul qilinadi.
2. DHTML ya'ni o'zgaruvchan sahiflar yaratish, ya'ni javascript bilan ishlash.

### **Yii framework**

Yii framework dunyo bo'ylab ancha keng tarqalgan. Bu framework bepul va qulay uni <http://yiiframework.ru> saytidan yuklab olish mumkin. Shuning uchun ham shu frameworkdan foydalanildi. Bu framework faqat Web serverlar (Denwer, XAMPP ) kabilarda ishlaydi.

*Framework bu*— web sayt qilish jarayonida qiyinchilik tug'diradigan yoki ko'p takrorlanadigan funksiyalar to'plamidir. Umuman olganda dasturchiga yordamchi funksiyalar.

*CMS* — *Content Management System* bu deyrli tayyor sayt.

Umuman olganda PHP dan framework, frameworkdan CMS, CMS dan Website yaratiladi.

MVC (Model View Controller, Model Ko'rinish Nazorat) Bu umumiy tushuncha bo'lib, loyihalash shablonidir.

*Model bu*- ma'lumotlar bazasidagi biror bir jadval bilan ishlaydigan obyekt (class)dir.

*View bu*- HTML kodlarni yozasidan fayl yoki foydalanuvchiga (browserda) ko'rinadigan kod.

*Controller bu*— foydalanuvchi so'roviga mos holda kerakli model bilan ishlab, kerakli viewni foydalanuvchi jo'natadi. Yiiframeworkdan foydalanish uchun yii ning classlaridan voris class olib, shu classlardan foydalanib sayt quriladi.

### **PHP**

1994 yili php tilining yaratuvchisi Rasmus Lerdorf o'zining saytiga mehmonlar kirishini hisoblash uchun Perl dasturlash tilada maxsus qobiq yozib

amalda qoʻllagan. Koʻp oʻtmay qobiqni ishlash unumdorligi juda past va sekinligi aniqlanganidan soʻng, dasturlarni yangidan “C” tilida yozib chiqishga toʻgʻri keladi. Keyin, dastlabki dastur kodlari muallif tarafidan barchaga koʻrish uchun serverga nashr qilingan. Server foydalanuvchilari kodlar bilan qiziqib, uni ishlatish muxlislari ham paydo boʻlgan. Hademay, bu dasturlar alohida loyihaga aylanib, 1995 yilning iyun oyida dasturiy mahsulot PHP(Personal Home Page) nomi bilan birinchi nashri chiqarildi. Imkoniyatlari oddiy va sodda boʻlib: bir necha buyruqlarni tushunadigan kod analizatori(tekshiruvchisi), mehmonlar hisoblagichini, kitobini, chatini yaratish uchun foydali dasturlar toʻplamidan iborat boʻlgan.

1996 yil aprel oyida dasturlar jiddiy qayta ishlanganidan soʻng, PHP/FI (Personal Home Page / Forms Interpreter) nomi bilan mahsulotning ikkinchi nashri paydo boʻldi. Bu mahsulot html-kod ichiga yozilib, html-formalarni qayta ishlab, hozirgi PHP dasturlash tilining tayanch imkoniytlarini ichiga olgan. PHP/FI kod yozilishi Perl tiliga juda oxshagan, lekin soddarok boʻlgan.rasmuslerdorf ZivSuraski EndiGutmans Rasmus Lerdorf, Endi Gutmans va Ziv Suraski 1997 yili PHP/FI 2.0 nashri chiqdi. Oʻsha paytda bu mahsulot bilan dunyo boʻyicha bir necha ming odam foydalanib, taxminan 50 ming domen boʻlib, Internetning 1%-ni tashkil qildi.1997 yilda Endi Gutmans va Ziv Suraski PHP/FI kodini boshqatdan yozib chiqishdi, chunki eski kod ular ishlatayotgan elektron tijorat tizimlari uchun yaroqsiz edi. Eski kodning mualliflaridan yordam olish uchun ular birlashishni taklif etib, PHP3 nomli loyihani PHP/FI -ni rasmiy vorisi deb eʼlon qilishdi. Yangi loyiha uyushgandan keyin PHP/FI loyihasi ishlab chiqarilishi toʻxtatilgan.PHP 3.0 -ning eng kuchli taraflaridan biri uning kengaytirala olinadigan yadrosi(tizimning bosharuv qismi) boʻlib, bundan tashqari, maʼlumot jamgʻarmalar bilan, turli protokollar va interfeyslar bilan birgalikda ishlash keng imkoniyatlari yaratildi. Muvaffaqiyatga erishishga ancha ahamiyatli fakt bu yangi tilni boyligi va obʼektlarga moʻljallangan dasturlashni koʻlla olishi. Yangi loyiha bilan birga nafaqat tilni tashqi, ichki tuzulishi oʻzgardi, balki oʻzini nomi ham. Endi PHP qisqartmasi “PHP: Hypertext Preprocessor”

ma'nosini anglatishi bildirildi.1998 yilning oxirida PHP foydalanuvchilarning soni o'n minglardan oshdi. Yuz mingdan oshiq veb-saytlar bu tilni qo'llashini e'lon qilishdi. Taxminan Internetning 10% serverlarida PHP 3.0 o'rnatilgan edi.1998 yilning iyun oyda PHP3 to'qqiz oy ommaviy tekshiruvidan keyin rasman e'lon qilindi. Shu yilning qishida Endi Gutmans va Ziv Suraski PHP yadrosini qaytadan ishlab chiqarishni boshlashgan. Ularning asosiy vazifasi PHP tizimini unumdorligini ko'tarish va kodning modullarini yaxshilash edi.1999 yilning o'rtalarida birinchi marta taqdim qilingan yangi yadro "Zend Engine" deb nomlangan ("Zend": mualliflar "Zeev" va "Andi" ismlardan tashkil topgan). Uni asosida tuzilgan yangi til PHP4 2000 yilning may oyida rasman chiqarilgan. Unumdorlik yaxshilangandan tashqari, PHP 4.0 muhim yangiliklarga ega bo'lib, sessiyalarni ko'llash, buferli chiqarish, kiritilgan ma'lumotlarni havfsiz qayta ishlash va yana bir necha yangi til tuzuvchilarini paydo bo'lishidan iborat.Hozirgi kunlarda "Zend Engine" qayta yaxshilanib PHP5 tili ishlab chiqarildi. Asosiy o'zgarishlar ob'ektlarga mo'ljallangan dasturlash modelida bo'lib, tilning imkoniyatlari yanada kengaytirdi.Hozirda PHP tilini yuz minglab dasturchilar ishlatib, bir necha million saytlar yaratilib, Internetning 20% domenlarini tashkil etmoqda.

### **MySql**

MySQL Shvetsiyadagi MySQL AB firmasi tomonida yaratilgan bo'lib, 2008 yilning fevralida Sun Microsystems tomonida sotib olingan. Endlikda dastur kodini yaratishgda ushbu firma javobgar. MySQL Server ochiq, ya'ni tekin dasturiy ta'minot bo'lib, General Public License himoyasidadir ya'ni turli himoyalar qo'yilmagan(kalitlar,litsenziya). MySQL AB/Sun firmasi dastur kodiga mualliflik huquqiga egaligi sababli, ushbu dasturiy ta'minotning kommerzial turi ham bor.MySQL nomining kelib chiqishi aniq emas. 1996 yildan beri ko'plab biblioteka va tool (vosita)lar nomi oldida My prefiksi qo'yila boshlangan.

Bu dastur o'zi bilan quyidagi amallarni bajaradi, yani kompyuterlar ichki tizimi bilan aloqa o'rnatish va ma'lumotlarni takrorlash va ma'lumotlarning saqlanish joyini aniqlash.

Berilganlar bazasining alohida bo'limlaridan turli foydalanuvchilar qismlarga ajratib foydalanishlari mumkin, berilganlar bazasining asosiy komponent tizimi bu ma'lumotlar hisoblanadi.

Berilganlar bazasining boshqa bir komponenti, foydalanuvchi, texnik hodim (hardware) va dastur ta'minotchisi (software) xisoblanadi, bu yerda Hardware-ichki qo'shimcha xotiraga ega bo'lgan disk va magnit lentalaridir. Berilganlar bazasining dasturiy ta'minoti foydalanuvchi tizimi va berilganlar bazalari bilan o'zaro aloqani o'rnatish vazifasini bajaradi, shunday dasturiy ta'minot bo'limiga, berilganlar bazalarini boshqarish tizimlari (BBBT) deb yuritiladi.

Foydalanuvchilarning berilganlar bazalari bo'yicha barcha talabnomalarini berilganlar bazalarini boshqarish tizimlari boshqarib turadi, berilganlar bazalaridan foydalanuvchilar ham bir necha turga bo'linadi:

- *Dasturchi foydalanuvchi*
- *Tizimli dasturchi*
- *Administratorlar*

Berilganlar bazalaridagi dasturlarning to'g'ri yozilishiga dasturchi javob beradi, administrator esa berilganlar bazalaridagi barcha ma'lumotlarning saqlanishini va kengayishini qo'llab quvvatlaydi.

Ko'p tarmoqli foydalanuvchilar uchun administrator berilganlar bazasida asosiy jarayon hisoblanadi.

Berilganlar bazasining tizimi, quyidagi jarayonlarga ega bo'lishi kerak:

- a) Bajarish- tizimning ishlashi vaqtida, foydalanuvchi so'roviga foydalanuvchi terminali mos tushishi kerak.
- b) Minimal takrorlanish-ichki tashuvchilardagi axborotlar hajmi, berilganlar bazasidagi saqlanayotgan axborotlarning takrorlanish darajasi bilan minimal holda bo'lishi talab etiladi.

Bundan tashqari har qanday jarayon, berilganlar bazasidan, hech bo'lmaganda bitta amal bajarilishini va undagi axborotning takrorlanishini talab etadi, bu esa boshqa jarayonlarning bajarilishi darajasini oshiradi. Shuning uchun berilganlar bazasini tashkil etishda, axborotlarning ko'p marotaba takrorlanmas-ligiga erishish kerak.

v) To'liqlik- berilganlar bazasi bilan ishlovchilar uchun, yangi axborotlarni kiritish uchun qonunlar mavjud bo'lib shulardan foydalanish talab etiladi. Shuni e'tiborga olib tizimda saqlanadigan axborotlar ichki ma'lumotlar bilan o'zaro aloqaga kirishadi.

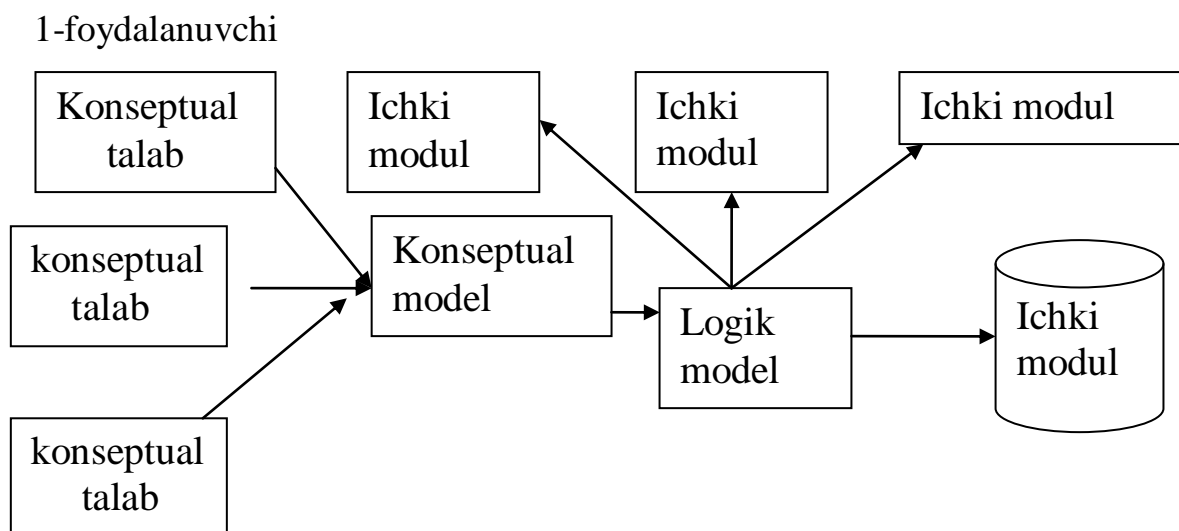
g) Xavfsizlik- Berilganlar bazalari ishonchli ximoya qilingan bo'lishi kerak, berilganlar bazalaridan noqonuniy foydalanish, va undagi ma'lumotlarni o'zgartirishga ruxsat etilmaydi, foydalanuvchi yoki tashkilot axborot olish yoki kiritish uchun maxsus ruhsatnomaga ega bo'lishi kerak.

d) Migrasiya- Ba'zi bir ma'lumotlarga foydalanuvchilar tamoni-dan ba'ziba'zida murojaat amalga oshiriladi. Shuni e'tiborga olib, ma'lumotlarni ichki ma'lumot tashuvchilarga joylashtirish qulay bo'ladi, bu esa doimo ishlatiladigan axborotlarni tez qidi-rib topishga va uni ishlatishga imkon beradi.

Berilganlar bazalarini boshqarish tizimlarida mustaqil ma'lumotlar mavjud bo'lib, ulardan faqat dasturchilar foydalanishlari mumkin. Dasturchi berilganlar bazasidan quyidagilarga javob olishi kerak bo'ladi:

- a) Berilganlar bazasida qanday qilib ma'lumotni format qilish kerak ?
- b) Berilganlar bazalarida ma'lumotlar qayerda joylashtiriladi?
- c) Berilganlar bazalarida ma'lumotlar qanday etib boradi?

So'ralayotgan savollarga tegishli javob berilgandan so'ng foydalanuvchilar dasturining o'zgarishiga olib keladi, barcha ma'lumotlar, yuqoridagi uchta savol korinishida yozilib foydalanuvchining dasturida kod ko'rinishida ishtirok etadi. Bundan tashqari komp-yuterlarning ishlash darajasining o'sishi sababli undagi axborotga kirish usullari va bazida saqlab qo'yilgan axborotlarning o'zgarishiga olib keladi. Berilganlar bazasidagi dasturiy ta'minot qo'yidagicha chizma asosida ish olib boradi:



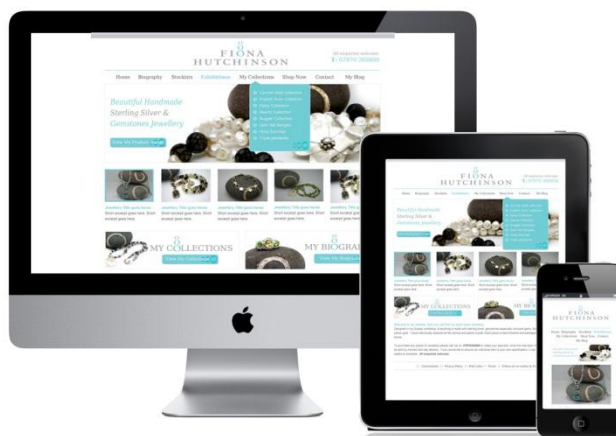
1-rasm . Berilganlar bazalarida mustaqil ma'lumotlarning joylashuvi

Berilganlar bazalarini ishlatish imkoniyatiga ega foydalanuvchilar uchun barcha imkoniyatlar namoyon etilishi bilan bir vaqtda , mustaqil ma'lumotlardan ham foydalanish imkoniyatlari yaratiladi. Boshqacha qilib aytganda (BBBT) berilganlar bazalarini boshqa-rish tizimlari ham shu tarzda quriladi.

Mustaqil ma'lumotlarga erishishning qulay usuli, yuqoridagi rasm-da keltirib o'tilgan. Foydalanuvchilarning har qanday talabi, bitta konseptual modelda jamlangan bo'ladi, shu konseptual model, so'ngra ma'lumotlarning namoyishini amalga oshiradi, qachonki berilganlar bazasini boshqarish tizimlari birgalikda o'rnatilgan bo'lsa.

## Bootstrap

Hozirgi kundagi har bir internet foydalanuvchisi o'zlarining kundalik web-saytlarining mobil versiyasi ham bo'lishini hohlashadi. Buni amaliy jihatdan olib qaraganda bir dizayn har bir qurilma ekraniga mos tushishi muhimdir.



2-rasm. Moslashuvchan arxitektura

**Responsive web design** (2-rasm) – shunday yondoshuvki, bunda dizayn va “razrabotka” foydalanuvchi hatti harakati va muhiti(qurilma) ga ekran o‘lchami, platformasi va orientatsiyasi asosida javob berishidir. Bu amaliyotda moslashuvchan to‘r(grid) va maketlar aralashmasi, rasmlar va CSS media-so‘rovlarning intellektual foydalanishidan iboratdir. Foydalanuvchi laptopdan iPadga o‘tgan zahoti, web-sayt avtomatik moslashuvchanlikka(“razrishiya”, rasm va skriptlar-javascript) o‘tishi kerak. Boshqacha qilib aytganda web-saytda shunday texnologiya bo‘lishi kerakki, sayt avtomatik foydalanuvchi sharoitiga moslanishi kerak. Bu esa bozordagi har bir qurilma uchun har hil dizayn va “razrabotka” jarayonidan qutulish yo‘lini yaratadi. **“Responsive web design” konseptsiyasi** Hozirgi kunda rivojlanib borayotgan “Moslashuvchan arxitektura” sohasida fizik sath qanday qilib soha orqali o‘tayotgan odamlarga moslashishi mumkin degan savol ko‘p berilmoqda. Arxitektorlar devorga o‘rnatilgan robototexnika va cho‘ziluvchan materiallar orqali, odamlar yaqinlashganda egiladigan va kengayadigan devor strukturalarini tajribadan o‘tkazmoqdalar. Yoki harakatni ilg‘ovchi datchiklar xonada odamlar soni ko‘payishi davomida xona temperaturasi va yoritishini nazorat qilish uchun iqlim-boshqarish tizimlari bilan birlashtirilishi mumkin. Ba’zi kompaniyalar allaqachon odam oynaga qanchadir masofada yaqinlashganda avtomatik qorong‘ulashishi yoki yorug‘lashishi mumkin bo‘lgan “aqlli oyna texnologiyasi”ni ishlab chiqishgan. Agar ushbu nazariyalarga web-dizayn

tomonidan qaraganda, webda ham shunga o'xshash bo'ladi. Moslashuvchan arxitektura singari, web-dizaynlar ham avtomatik moslashishi zarur. Bu har bir foydalanuvchi kategoriyasiga sanoqsiz individual yechimlar talab qilmasligi kerak.

Moslashuvchan web-dizaynlar ko'proq fikrlashning abstrakt yo'lini talab qiladi. Shuningdek, ba'zi g'oyalar allaqachon tajribadan o'tkazilgan: "cho'ziluvchan layout(maket)" lar, media so'rovlar va scriptlar(jquery, javascript) yordamida sahifalarni reformat qilish. Lekin "Responsive web-design" faqat ekran o'lchamiga moslashuvchan va avtomatik o'lchami o'zgaruvchan rasmlardan iborat emas. Tizimni yaratishda huddi shu texnologiyani qo'llagan ya'ni bootstrapdan foydalanilgan. Bootstrapning barcha komponentlari bepul. Bootstrapni o'rnatish juda oson bunda bootstrap-theme.css, bootstrap-theme.min.css, bootstrap.css, bootstrap.min.js, bootstrap.js larni bog'lab qo'ysa kifoya va bu bootstrapda foydalanish uchun bootstrapning har bir classi va div lari nima ish qilishi bilan tanishib chiqish kerak. Shundagina ko'zlangan maqsadga erishish mumkin

Bootstrap hozirgi kunda eng ko'p ishlatilayotgan frontend development framework hisoblanadi. Unda HTML, CSS va JavaScript to'liq joylashtirilgan va ushbu framework yordamida o'zingiz istagan sayt ko'rinishini bir necha daqiqalarda, qiyinchiliklarsiz tayyorlab olishingiz mumkin.

Frameworkning rasmiy [getbootstrap.com](http://getbootstrap.com) sayti orqali uni to'liq yuklab olishingiz va hech qanday cheklovlarsiz ishlatishingiz mumkin. Agar tizim bilan ishlashni hali bilmasangiz, Google orqali qidirishda Bootstrapga oid minglab darslarni, maqolalarni topishingiz mumkin. YouTube, Vimeo kabi video saytlarda esa Bootstrap to'g'risida tayyorlangan video darsni juda ko'plab topish mumkin.

Bootstrap ilk «mobile first» framework hisoblanib, unda birinchi e'tibor tayyorlangan sahifalar mobil qurilmalarda qanday ko'rinishiga qaratiladi. O'zining moslashuvchangili sabab Bootstrapda tayyorlangan sahifalar barcha hajmdagi ekranlarda bir xil chiroyli ko'rinadi. Bootstrap haqiqatdan ham front end ustalarining ishini bir necha o'n barobarga osonlashtirdi. Bir necha yil avval

HTML va CSS kodlarni to'liq qo'lda yozib sayt tayyorlayotgan kishilar Bootstrapning qanchalik qulay tizim ekanligini bir necha bor ishlatib ko'rgandanoq bilib olishlari mumkin.

Bootstrap haqida to'liq ma'lumotni tizimning rasmiy [sayti](#)dan olishingiz mumkin. U yerda framework ishlatilishi haqida yetarlicha tushuncha berilgan. Agar u yerda o'rganish Siz uchun qiyinlik qilsa, Bootstrapning eng sodda darslarini [w3schools](#) saytining Bootstrap bo'limidan topishingiz mumkin.

Bootstrap bilan ishlash ko'nikmasini yaxshi o'zlashtirganingizdan so'ng u orqali sahifalar tayyorlab, internetda yaxshigina pul ishlab olishingiz mumkin. To'g'ri, juda ko'p to'lashmaydi, ammo 1-3 kunda bajarsa bo'ladigan har bir loyiha uchun 50-100 dollar atrofida ishlab olishingiz mumkin. Bootstrap tizimini yaxshi o'zlashtirganingizdan so'ng qanday ishlar topishingiz mumkinligini [mana bu sahifadan](#) ko'rib xulosa qilishingiz mumkin.

Bugundan boshlab Bootstrapni o'rganishingizga hech narsa halal bermaydi: uni o'rganish uchun sayt ham, domen ham, hosting ham kerak emas. Shunchaki Bootstrapni kompyuteringizga yuklab oling, darslarda ko'rsatgandek saytlar ko'rinishini tayyorlashga kirishing.

## **Less va Sass**

Less va Sass oldindan ishlov beruvchi freymvorklari CSS asosida yaratilgan bo'lib, dasturchilar uchun CSS bilan ishlashda anchagina qulayliklar va qo'shimcha imkoniyatlarni yaratib beradi. Uning yordamida sayt tuzishga ketadigan vaqtni tejab, nisbatan kamroq kod yozish mumkin bo'ladi. Hatto, bu ikkisi yordamida sayt stillarini yozishga dasturlash tillari kabi yondashish mumkin.

Kamroq (ba'zan LITs sifatida stilize qilingan) - kaskadli jadvallarni (CSS) tuzish va mijoz yoki server tomonida ishlaydigan dinamik dastlabki protsessor uslub sahifasi tili. Alexis Sellier tomonidan ishlab chiqilgan, Sassning ta'siri kamroq va CSS-ga o'xshash blok formatlash sintaksisini moslashtirgan Sassning yangi "SCSS" sintaksisiga ta'sir qilgan. Kamroq - ochiq manba. Uning birinchi

versiyasi Rubyda yozilgan; ammo keyingi versiyalarda Ruby'dan foydalanish bekor qilindi va JavaScript-ni o'zgartirildi. "Less" indentli sintaksisi, ichki CSS-ning tegishli bo'lganligi sababli, ichki kursdagilardan iborat. Kam bo'lmagan mexanizmlar: o'zgaruvchilar, joylashtirish, aralashmalar, operatorlar va funktsiyalar; Less va boshqa CSS prekompyuterlari o'rtasidagi asosiy farqi shundaki, kamcha brauzer tomonidan kamroq .js orqali real vaqtda kompilyatsiya qilish imkonini beradi.

## **Html, CSS, JavaScript**

HTML hujjati mantiqiy tarkibga ega. HTML hujjati sarlavhasida, inson boshidagi kabi, imkoniyatlar taqdim etilgan; ular "sahnadan tashqarida" murakkab amallarni boshqarish uchun ishlaydi. HTML tanasi fizik atributlar sohasi hisoblanadi. HTML tanasi – bu brauzer orqali sahifada ko'rinadigan barcha ma'lumotlar hisoblanadi. Standart deteglar yordamida tanani bo'lishning formal usullari mavjud emas. Ammo, BODY sohasida va mantiqan HTML hujjatining hammasida, buni amalga oshirishning sodda usuli mavjud.

Bu masala maxsus izoh detegi yordamida yechiladi. U hech qanday qoidaga bo'yinsinmasligi bilan HTML ning boshqa deteglaridan ajrailb turadi. Lekin, sahifa haddan tashqari uzun va murakkab bo'lganda, sahifa ichida yo'nalish olishni osonlashtiruvchi kuchli deteg hisoblanadi.

Izohlarni hujjatning ixtiyoriy joyida ishlatish va bir necha satrlarga joylashishi mumkin. Izohlar sizni va boshqa xodimlarni chalg'itmasligi, ya'ni yordam berish lozim. Ulardan faqat zarurat tug'ilgandagina foydalaning.

Hujjat sarlavhasiga quyidagi axborotlar kiradi.

*1. Sahifa nomi.* <TITLE> va unga mos bo'lgan </TITLE> detegi sahifaning nomlanishini aniqlaydi. U HTML-sahifaning tanasida kelmaydi, ya'ni brauzerda ko'rilganda bu ma'lumot asosiy oynada ko'rinmaydi. Bu nom brauzerning sarlavha satrida chiqariladi. <TITLE> detegida sahifaning nomini ko'rsatish lozim. U sodda bo'lishi, ammo sahifaning mazmunini tushunarli bayon etishi lozim. TITLE

elementiga HTML ning boshqa deteglarini qo‘shish mumkin emas, ammo maxsus belgilardan foydalanish mumkin.

2. *Ssenariy.* Sahifada bajariluvchi har qanday ssenariy, masalan, JavaScript tilida yozilgan, hujjat sarlavhasiga biriktiriladi. Bosh (HEAD) vizual emas, balki mantiqiy jarayonlar uchun mo‘ljallangan. Ssenariyning o‘zi ko‘rinmaganligi sababli, uning bajarilish jarayoni ham ko‘rinmaydi. Jarayonnig natijalari sahifaning holatini aniqlaydi.

3. *Stil.* O‘z HTML sahifalariga boshqaruv elementlari va stilarni qo‘shuvchilar uchun sarlavhaga kaskadli sahifalar stillarini kiritish yoki ularni Web-sahifalar bilan bog‘lash imkoniyati mavjud. Bu ma’lumotlar ham HTML hujjati sarlavhasida joylashtiriladi.

4. *Metaaxborot.* META detegi o‘zining ko‘plab “mantiqiy” jarayonlarni bajarish imkonini beruvchi kuchi bilan ajralib turadi. Masalan, hujjat muallifini, kalit so‘zlarni yoki maxsus harakatlarni ko‘rsatish.

Ko‘plab sahifalar ssenariylarni, varaq stillarini yoki metaaxborotlarni talab qilmaydi, ammo barcha sahifalar uchun nom zarur.

HTML hujjati tanasida (BODY deteglar juftligi) ko‘rish uchun taqdim etiladigan barcha axborot joylashadi. Qobiq yaratilgandan so‘ng birinchi masalalardan biri sahifa tanasiga axborot kiritish bo‘ladi. U quyidagilardan iborat bo‘lishi mumkin.

5. *Matn.* Bo‘g‘inning matnli mazmuni tanada joylashadi. Axborotni taqdim etish uchun formatlashdan (matn bo‘g‘inga masofaviy kiruvchilarga o‘qishda qulay bo‘lishi uchun) foydalaniladi.

6. *Tasvirlar.* Bo‘g‘in tematikasini ko‘rsatish uchun grafikadan, matnni to‘ldiruvchi fototasvirlardan yoki navigatsion tugmalardan foydalaniash mumkin. Har qanday holda ham tasvirlar hujjat tanasida joylashgan tasvirlar axborotning muhim qismi hisoblanadi.

7. *Muloqotlar.* Web ning yuragi va qalbi bo‘lgan muloqotlar foydalanuvchilarga bo‘g‘inda harakatlanish imkonini beradi. Muloqotlar har doim sahifa tanasida joylashadi.

*8.Multimedia va maxsus dasrurlar.* Shockwave, Flash, Java apletlari va hattoki biriktirilgan videoni boshqarish uchun HTML-hujjat tasnasida joylashuvhchi kod qo‘llaniladi.

Birinchi bor JavaScript 1995 chi yil o‘rtaga chiqdi. JavaScript 1995 yili ishlatilishi boshladi, ammo 1998 yilga kelib keng qo‘llanishga kirdi. Uning chiqishini asosiy sabablaridan biri bu Client tomonidan bo‘ladigan kiritishlarni tekshirish uchun, yani server tomonida qilinadigan ishlarni bir qismini olib tashlash uchun.Lekin shuncha vaqt o‘tgandan so‘ng javascript web development'ning eng asoslaridan biri bo‘lib qoldi. Microsoft kompanyasi ham o‘zining scripting language yani Vbscriptni chiqardi ammo lekin uni keng qo‘llab bo‘lmas edi, sababi Internet Explorer'dan boshqa hech qaysi Web Browser uni ishlata olmasdi. Keyin esa Microsoft ham Internet Explorer ichiga JavaScript'ni qo‘shdi. Hozirgi kunga kelib hamma Web

Browserlar JavaScriptni ishlata oladi.JavaScript bu “Client side programming” yani kimki sahifangizni ochsa shu odam tomonida ishlaydi. PHP, Perl, CGI, JSP lar kabi serverda ishlamaydi. JavaScriptda DHTML yani Dynamic HTML sahifalarini yozish mumkin. Bu degani foydlanuvchining qandaydir harakatiga qarab ish qilishi, va boshqalardir.JavaScript'ning asosi uch qismdir bular:

1. ECMAScript
2. The Document Object Model (DOM)
3. The Browser Object Model (BOM)

ECMAScript: Bu JavaScriptning asosi, yani programmalshidir, bu hech qanday browser'ga bog‘liq bo‘lmagan, hech qanday server ga bog‘liq bo‘lmagan qismidir. Buning ichida buyruqlar, script yozish usullari, o‘zgaruvchan/o‘zgarmaslar, operatorlar, kalit so‘zlar dir (syntax, types, keywords, operators, statements)

DOM( The Document Object Model): API (Application programming interface) for HTML and XML,yani programmlar yaratish ko‘rinishidir. Agar boshqa tillarda misol uchun Java yoki Delphi, Visual Basic'da programmlarda

ular OOP Object Oriented Programming, ya'ni ularda asos va uning ichida elementlari bor, bu yerda ham huddi shunday.

BOM (Browser Object Model) Bu web browser bilan aloqali ishlardir. Yani bularga statusbar'ning tekstini o'zgartirish, toolbarni o'zgartirish va hkz, faqatgina browser bilan aloqali bo'lib web sahifaning o'ziga aloqali emasdir.

Bular:

1. Yangi browser ochish
2. Browserni yurgizish, katta kichik qilish, yopish
3. Cookie yuklanishi
4. Ekran kattaligi
5. Toolbar, Navigator lar

## **II BOB. Berilganlar bazasini tashkil qilish va uni boshqarish tizimi**

### **2.1. Berilganlar bazasi tashkil qilish**

Berilganlar bazasi-bu o‘zaro bog‘langan va tartiblangan ma’lumotlar majmui bo‘lib, u ko‘rilayotgan ob‘yektlarning xususiyatini, holatini va ob‘yektlar o‘rtasidagi munosabatni ma’lum sohada tavsiflaydi. Darhaqiqat hozirgi kunda inson hayotida berilganlar bazasida kerakli axborotlarni saqlash va undan foydalanish juda muhim rol o‘ynaydi. Sababi: jamiyat taraqqiyotining qaysi jabhasiga nazar solmaylik o‘zimizga kerakli ma’lumotlarni olish uchun, albatta, berilganlar bazasiga murojaat qilishga majbur bo‘lamiz. Demak, berilganlar bazasini tashkil qilish axborot almashuv texnologiyasining eng dolzarb hal qilinadigan muammolaridan biriga aylanib borayotgani davr taqazosi. Shuni qayd qilish lozimki, berilganlar bazasini yaratishda ikkita muhim shartni hisobga olish zarur:

Birinchi, ma’lumotlar turi, ko‘rinishi, ularni qo‘llaydigan dasturlarga bog‘liq bo‘lmasligi lozim, ya’ni MBga yangi ma’lumotlarni kiritganda yoki ma’lumotlar turini o‘zgartirganda, dasturlarni o‘zgartirish talab etilmasligi lozim.

Ikkinchi, berilganlar bazasidagi kerakli ma’lumotni bilish yoki izlash uchun biror dastur tuzishga hojat qolmasin.

Shuning uchun ham MBni tashkil etishda ma’lum qonun va qoidalarga amal qilish lozim.

### **2.2. Berilganlar bazasi tashkil qilishdagi asosiy talablar**

**Oddiyliigi:** Foydalanuvchilar o‘z ixtiyorida qanday ma’lumotlar borligini juda oson tanib va tushinishi mumkin.

**Umumdorligi:** Qaysi ma’lumotlardan foydalanish talab qilinsa, bu ma’lumotlarga qo‘yiladigan talablar shunday tezliklar bilan ta’minlanadi.

Berilganlar bazasi boshqarish tizimida har bir MB modeli quyidagi xususiyatlari bo‘yicha tavsiflanadi:

1. Ma'lumotlar tuzilmalarining turi.
2. Ma'lumotlar ustida bajariladigan amallar.
3. Butunlikning cheklanganligi.

Bu xususiyatlarni e'tiborga olgan holda ma'lumotlar bazasi modellari quyidagi turlarga bo'linadi:

- Daraxtsimon (ierarxik) modellar.
- Tarmoqli (to'rli) modellar.
- Relyatsion modellar.

Yana shu narsani ta'kidlash lozimki, ma'lumotlar bazasi modellarining faqatgina yuqorida qayd qilingan modeli mavjud deyish noto'g'ri. Chunki, bulardan tashqari yana Berilganlar bazasining binar munosabatlar modeli, ER-modellari, semantic model kabi boshqa turlari ham mavjud. Lekin amalda asosan dastlabki ta'kidlangan 3 turli modellar ko'proq qo'llanilib kelinmoqda. Shuning uchun ham ushbu modellar haqida qisqacha to'xtalib o'tish joiz.

Daraxtsimon (ierarxik) modelda ob'yektlar yozuvlar ko'rinishida ifodalanadi.

Ierarxik modelda ikki jadvaldagi elementlar bog'langan bo'lsa, unday ma'lumotlar tarmoqli (to'rli) modelda ifodalangan deyiladi. Tarmoqli modellarda ham ob'yektlar daraxtsimon modellardagi kabi yozuvlar ko'rinishida tasvirlanadi. Ob'yektlarning o'zaro aloqalari yozuvlar o'rtasidagi aloqalar sifatida tavsiflanadi.

Relyatsion modellarda esa ob'yektlar va ularning o'zaro aloqalari ikki o'lchovli jadval ko'rinishida tasvirlanadi. Ma'lumotlarni bunday ko'rinishda tasvirlanishi ob'yektlarning o'zaro aloqalarni yaqqol tasvirlanishiga asos bo'ldi.

Axborot texnologiyalarning rivojlanishi va axborot oqimlarining tobora ortib borishi, ma'lumotlarning tez o'zgarishi kabi holatlar insoniyatni bu ma'lumotlarni o'z vaqtida qayta ishlash choralarini qidirib topishga undaydi. Ma'lumotlarni saqlash, uzatish va qayta ishlash uchun **Berilganlar bazasi (BB)** ni yaratish, so'ngra undan keng foydalanish bugungi kunda dolzarb bo'lib qolmoqda.

**Berilganlar bazasi** – bu o‘zaro bog‘langan va tartiblangan ma’lumotlar majmuasi bo‘lib, u qurilayotgan ob’ektlarning hususiyatini, holatini va ob’ektlar o‘rtasidagi munasabatni ma’lum sohaga tavsiflaydi.

Darhaqiqat, hozirgi kunda inson hoyotida **Berilganlar bazasida** kerakli axborotlarni saqlash va undan oqilona foydalanish juda muhim ro‘l o‘ynaydi. Sababi: jamiyat taraqqiyotining qaysi jabhasiga nazar solmaylik o‘zimizga keraki ma’lumotlarni olish uchun, **Berilganlar bazasiga** murojaat qilishga majbur bo‘lamiz. Demak, **Berilganlar bazasini** tashkil qilish axborot almashuv texnologiyasining eng dolzarb hal qilinadigan sohalaridan biriga aylanib botoyotgani davrning taqozasi hisoblanadi.

Ma’lumki, **Berilganlar bazasi** tushinchasi fanga kelgunaga qadar, ma’lumotlardan turli ko‘rinishda foydalanish juda ko‘plab qiyinchiliklarni keltirib chiqarar edi. Bunda dastur tuzuvchilar ma’lumotlarni shunday tashkil qilar edilarki, u faqat qaralayotgan masala uchungina o‘rinli bo‘lardi. Har bir yangi masalani hal qilishda ma’lumotlar qaytadan tashkil qilinlar va bu hol yaratilgan dasturlardan foydalanishni qiyinlashtirar edi.

Berilganlar bazasini yaratishda ikkita muhim shartni hisobga olmoq zarur:

**Birinchidan**, ma’lumotlarning turi, ko‘rinishi, ularni qo‘llaydigan dasturlarga bog‘liq bo‘lmasligini ta’minlash kerak **Berilganlar bazasiga** yangi ma’lumotlarni kiritganda yoki ma’lumotlar turini o‘zgartirganda, dasturlarni o‘zgartirish talab etilmasligi lozim.

**Ikkinchidan**, berilganlar bazasidagi kerakli ma’lumotni bilish yoki izlash uchun biror dastur tuzishga hojat qolmasligi zarur.

Shuning uchun ham **Berilganlar bazasini** tashkil etishda ma’lum qonun va qoidalarga amal qilish lozim. Bundan buyon axborot so‘zini ma’lumot so‘zidan farqlaymiz, yani axborot so‘zini umumiy tushuncha sifatida qabul qilib, ma’lumot deganda aniq bir belgilangan narsa yoki hodisa sifatlarini nazarda tutamiz.

Bugungi kunda ma’lumotlarni eng ishonchli saqlaydigan vositalardan biri esa hozirgi zamon kompyuterlaridir, kompyuterlarda saqlanadigan berilganlar bazasi –bu maxsus formatga ega bo‘lgan muayyan tuzilmali fayl demakdir.

Kompyuter xotirasida har bir fayl, yozuv deb ataladigan bir hil turdagi qismlardan iborat bo‘ladi. Yozuv-o‘zaro bog‘langan ma’lumotlarning bir qismidir, fayldagi yozuvlar soni, qaralayotgan ma’lumotning o‘lchoviga bog‘liq. Har bir yozuv esa maydon deb ataladigan bo‘laklardan tashkil topadi, maydon ma’lumotlari imkoni boricha qisqa to‘plamdan iborat bo‘lishi lozim.

Har bir maydon o‘zi ifodalaydigan malumotlariga ko‘ra biror nomga ega bo‘ladi. Masalan, biror oliy o‘quv yurtining biron-bir fa’kultetida shlayotgan bitta o‘qituvchi to‘g‘risidagi ma’lumotlar berilgan qo‘yidagi jadvalni ko‘rib o‘taylik.

Familiyasi	Ismi	Tug‘ilgan sanasi	rasmi	Turar joyi	Mutahassisligi
Ochilov	Alisher	2.05.1978	<rasm yoli>	5-kichik daxa	Matematik
Qobulov	Farxod	2.12.1982	<rasm yoli>	I.Sino	Dasturchi
Aminov	Jamshid	3.06.1980	<rasm yoli>	A.Temur	Web dasturchi
Tolipov	Salim	24.05.1974	<rasm yoli>	At.Termiziy	Fizika

Bu misolda 6 ta yozuv bo‘lib, ularning har biri oltita maydondan iborat. Mazkur maydonlarning har biri mos ravishda “Familiyasi”, “Ismi”, “Tug‘ulgan sanasi”, “rasmi”, “Turar joyi”, va “Mutazasissligi” deb nomlangi. Demak, yozuvdagi maydonlar soni yozuvga kiritiladigan ma’lumotlar hajmiga bog‘liq.

Fayldagi yozuv mavzulari birlamchi hisoblanadi, chunki biror yozuvdagi ihtiyoriy ma’lumotlarni boshqa yozuvdagi ma’lumotlar bilan taqqoslab aniqlash mumkin emas. Shuning uchun ham bizga kerakli bo‘ladigan ikkilamchi yozuvlarni esa faqat amaliy dasturlar yordamida olish mumkin bo‘ladi.

Modomiki shunday ekan berilganlar bazasini tashkil qilish, ularga qo‘shimcha ma’lumotlarni kiritish va mavjud ma’lumotlar bazasidan foydalanish uchun mahsus ma’lumot bazalari bilan ishlaydigan dasturlar zarur bo‘ladi.

Bunday dasturlar majmuyi berilganlar bazasini boshqarish tizimi deb yuritiladi. Aniqroq qilib aytganda berilganlar bazasini boshqarish tizimi bu ko‘plab foydalanuvchilar tomonidan berilganlar bazasini yaratish, unga qo‘shimcha

ma'lumotlarni kiritish va ma'lumotlar bazasini birgalikda ishlatish uchun zarur bo'lgan dasturlar majmu kiradi. Berilganlar bazasini boshqarish tizimining tarkibida asosiy komponent – bu ma'lumotlar bo'lsa, boshqa komponenti-foydalanuvchilar, Hardware-texnik va Software dasturiy ta'minoti hisoblanadi. Hardware tashqi qo'shimcha xotiradan(disk magnit lentasi ) iborat bo'lsa, dastur qismi esa berilganlar bazasi bilan foydalanuvchi o'rtasidagi muloqatni tashkil qilishni amalga oshiradi.

Berilganlar bazasining tuzilishi o'rganilyotgan ob'ektning ma'lumotlari ko'rinishi, ma'nosi, tuzilishi va hajmiga bog'liq bo'ladi.

Berilganlar bazalaridan, foydalanuvchilar quyidagi kategoriyalarga bo'lingan holda ish olib boradi:

- foydalanuvchi – dastur tuzuvchi;
- tizimli dastur tuzuvchi;
- berilganlar bazasi administratori;

Bunda dastur tuzgan foydalanuvchi berilganlar bazasini boshqarish tizimi uchun yozgan dasturiga javob beradi, tizimli dastur tuzuvchi esa butun tizimning ishlashi uchun javobgar hisoblanadi. U holda berilganlar bazasi administratori tizimning saqlanish xolatiga va ishonchliligiga javob beradi.

Berilganlar bazalari bilan ishlash vaqtida foydalanuvchilar ko'plab buyruq va operatorlar bilan ishlashlariga to'g'ri keladi, ana shunday buyruqlarning ishlashini qo'yidagicha tahlil etish mumkin.

Berilganlar bazasini boshqarish tizimi quyidagicha tavsiflanadi:

- **Исполнимость**- bajarilishlik, foydalanuvchi so'roviga hozir javoblik bilan muloqotga kirishish.
- **Минимальная повторяемость**- minimal takrorlanishlik

berilganlar bazasidagi ma'lumot iloji boricha kam takrorlanishi lozim, aks holda ma'lumotlarni izlash susayadi.

- **Yaxlitlik** –axborotni berilganlar bazasida saqlash iloji boricha ma'lumotlar orasidagi bog'liqlikni taminlab turadi. **Безопасность**-xavfsizlik berilganlar

bazasiga ruxsat berilmagan kirishdan ishonchli himoya qilishdan iborat. Faqat foydalanuvchi va tashkilotlarga tegishli bo'lgan ma'lumotlarni tashkilot administratorlari kira olish va foydalanish huquqiga egalik qilishi mumkin.

- Миграция- ba'zi bir ma'lumotlar foydalanuvchilar tomonidan tez-tez ishlatilib turiladi, boshqalari esa faqat talab asosida ishlatiladi. Shuni etiborga olib bunday ma'lumotlarni tashqi xotirada joylashtiradi sababi eng ko'p ishlatiladigan ma'lumotlarga murojaat qilish qulay amalga oshirilishi kerak bo'ladi.

Berilganlar bazasini boshqarish tizimida har bir berilganlar baza-lari modeli **qo'yidagi xususiyatlari bo'yicha tavsiflanadi:**

1. Ma'lumotlar tuzilmalarining turi.
2. Ma'lumotlar ustida bajariladigan amallar.
3. Butunlikning cheklanganligi.

Bu xususiyatlarni e'tiborga olgan holda berilganlar bazasi modellari quydagi turlariga bo'linadi:

- Daraxtsimon (ierarxik) modellar
- Tarmoqli (to'rtli) modellar
- Reliyasion modellar

Yana shu narsani ta'kidlash lozimki, berilganlar bazasi modellarining faqatgina yuqorida qayd qilingan modellari mavjud deyish noto'g'iri, chunki bulardan tashqari yana berilganlar bazasining binar munosabatlar modeli, ER- modellar, semantic model kabi boshqa turlari ham mavjud. Lekin amalda asosan dastlabki ta'kidlangan uch turli modellar ko'proq qo'llanilib kelinmoqda. Berilganlar bazalarida modellar o'z navbatida bir necha bo'limlarga bo'linadi.

- Daraxtsimon model. Bu modelda asosan obektlar yozuvlar ko'rinishida ifodalanadi.

- Ierarhik model. Bu modelda ikki turdagi elementlar bog'langan bo'lsa, bunday ma'lumotlar tarmoqli va turli modelda ifodalangan deyiladi. Tarmoqli modellarda ham ob'ektlar daraxtsimon modellardagi yozuvlar ko'rinishida tasvirlanadi, obektlarning o'zaro aloqalari yozuvlar o'rtasidagi aloqalar sifatida tafsiflanadi.
- Relyasion model. Bu modelda ob'ektlar va ularning o'zaro aloqalari ikki o'lchovli jadval ko'rinishida tasvirlanadi. Ma'lumotlarning bunday ko'rinishda tasvirlanishi ob'ektlarining o'zaro aloqalarini yaqqol tasvirlanishiga olib keladi.

### **2.3. Tarmoqda ma'lumotlar bazalari arxitekturasi**

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

#### **Klient/server arxitekturasi**

Sodda xolda klient/server arxitektura amaliy dasturdagi qayta ishlashni ikki yoki undan ko'p mantiqiy qismlarga ajratishga asoslangan. Ma'lumotlar bazasi qandaydir amaliy dastur tomonidan foydalanish uchun yaratilganidir. Soddalashtirib aytish mumkinki ma'lumotlar bazasi klient/server arxitekturasining bir qismini tashkil qiladi. Ma'lumotlar bazasi «server», undan foydalanuvchi har qanday amaliy dastur «klient». Odatda klient va server har xil mashinalarda

joylashgan; ko'p xollarda klient amaliy dasturi ma'lumotlar bazasiga do'stona interfeysdir. Quyidagi grafik shaklda klient/server sodda tizimi tasviri berilgan.



*3-rasm Klient/server arxitekturasi*

Berilganlar bazasi bilan ishlaydigan amaliy dasturlar yaratilganda avvalambor klientni ma'lumotlar bazasi bilan bog'lash imkoniyatiga ega bo'lish kerak. Berilganlar bazalari yaratuvchilar dasturchilardan konkret tilga mo'ljallangan, API yordamida bog'lanish asosiy mexanizmlarni berkitadilar. Ma'lumotlar bazasi bilan ishlovchi dastur yaratganingizda sizni so'rovlaringizni tarmoq orqali ma'lumotlar bazasi serveriga uzatiluvchi TCP/IP paketlariga translyatsiya qiladi. Berilganlar bazasiga murojaat API larining tashqi ko'rinishi har xil va dasturlash tillariga, ko'p xollarda ma'lumotlar bazasining o'ziga bog'liq. MySQL uchun API lar MSSQL bilan o'xshash qilib yaratilgani uchun, biz ko'radigan API lar orasidagi farq minimaldir. Uch bo'g'inli arxitektura Arxitekturani Amaliy dasturlar rivojlanishi bilan takomillashtirish ancha murakkabdir. Bu arxitekturada ob'ektga-yo'naltirilgan dasturlash imkoniyatlaridan foydalanish ham qiyin. Birinchi muammo «nozik klientlar» haqidagi baxslarda o'z aksini topdi. Nozik klientlarga bo'lgan talab, klientga uzatilayotgan ma'lumotlar o'sib boish tendetsiyasidan kelib chiqdi. Bu muammo PowerBuilder va VisualBasic larda ko'rindi. Ular bazadan ma'lumotlarni GUI ga oladi va bu ma'lumotlar ustidagi hamma amallarni GUI da bajaradi. Foydalanuvchi interfeysini baza yadrosiga bog'lab qo'yish foydalanuvchilar soni va ma'lumotlar xajmi oshishi bilan o'zgartirish va masshtablash qiyin bo'lgan dasturlar yaratilishiga olib keladi. Agar sizda foydalanuvchi interfeysi yaratish tajribasi bo'lsa, foydalanuvchi xoxishiga qarab

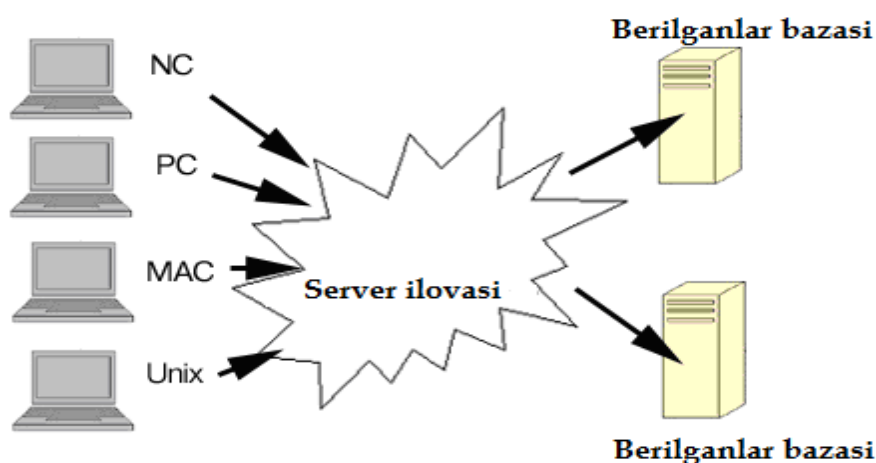
interfeysni qayta ishlab chiqish muammosiga duch kelgansiz. Bunday qayta ishlashni kamaytirish yo‘li GUI uchun faqat bitta vazifa – foydalanuvchi interfeysi vazifasini qoldirish kerak. Foydalanuvchi bunday interfeysi chindan ham nozik klientdir.

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

Nozik klientlar – bugunda yagona yo‘nalish emas. Boshqa yo‘nalish-koddan qayta foydalanish. Xar xil amaliy dasturlar uchun kod biznes logika deb atalgan qayta ishlashga yo‘naltiriladi. Agar biznes lgogika foydalanuvchi interfeysida joylashgan bo‘lsa, koddan qayta foydalanishni ta‘minlash qiyin bo‘ladi. Bu muammoni xal qilish yo‘li Amaliy dasturni ikki qismga emas uch qismga ajratishdir. Bunday arxitektura uch bo‘g‘inli deyiladi. Klientdagi foydalanuvchi interfeys haqida gapirganimizda, mantiqiy farqni nazarda tutamiz. Nozik klient bir turi «O‘ta nozik klient», bo‘lib ko‘pchilik Web-sahifa deb qabul qiladi. Web-sahifa dinamik tarzda Web-serverda yaratilishi mumkin. Bu holda klient ishining ko‘p qismi serverda HTML-sahifalarni dinamik generatsiya qilish shaklida bajariladi. 1- rasmda ko‘rsatilgan ikki bo‘g‘inli arxitekturani 2- rasmda ko‘rsatilgan uch bo‘g‘inli arxitektura bilan solishtiring. Foydalanuvchi interfeysi va berilganlar bazasi orasida qo‘shimcha qatlam joylashtiriladi. Bu yangi qatlam amaliy dasturlar serveri o‘zida biror soha uchun umumiy bo‘ldgan Amaliy dastur ish mantig‘i – biznes mantiqni oladi. Klient o‘rta yarus ob‘ektlarini ko‘rish vositasi, ma‘lumotlar bazasi bo‘lsa shu ob‘ektlar omboriga aylanadi.

Buning eng katta yutug‘i-foydalanuvchi interfeysi bilan berilganlar bazasini ajratilishidir. Endi berilganlar bazasi haqidagi ma‘lumotni GUI ga kiritishga to‘g‘ri kelmaydi. Balkim, ma‘lumotlar bazasi bilan qanday ishlash haqidagi

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

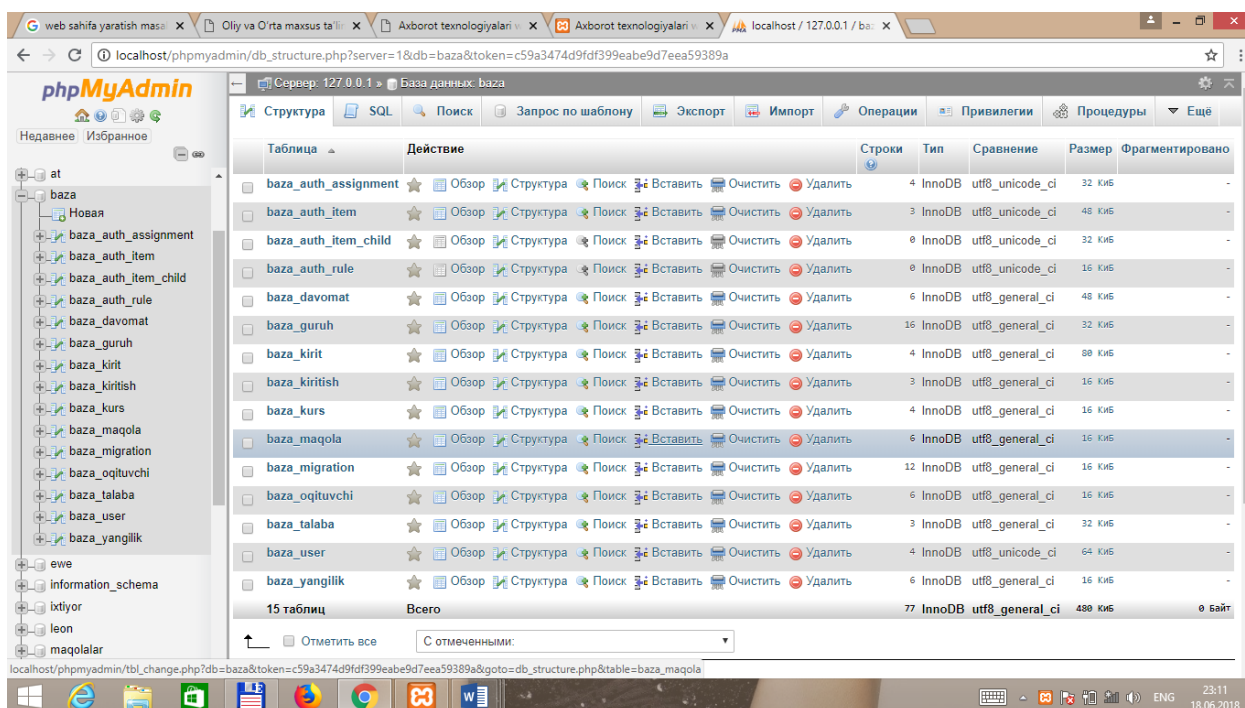


4-rasm Uch bo'g'inli arxitektura

## III BOB. Avtomatlashgan kafedra tizimining ishlash tamoyili

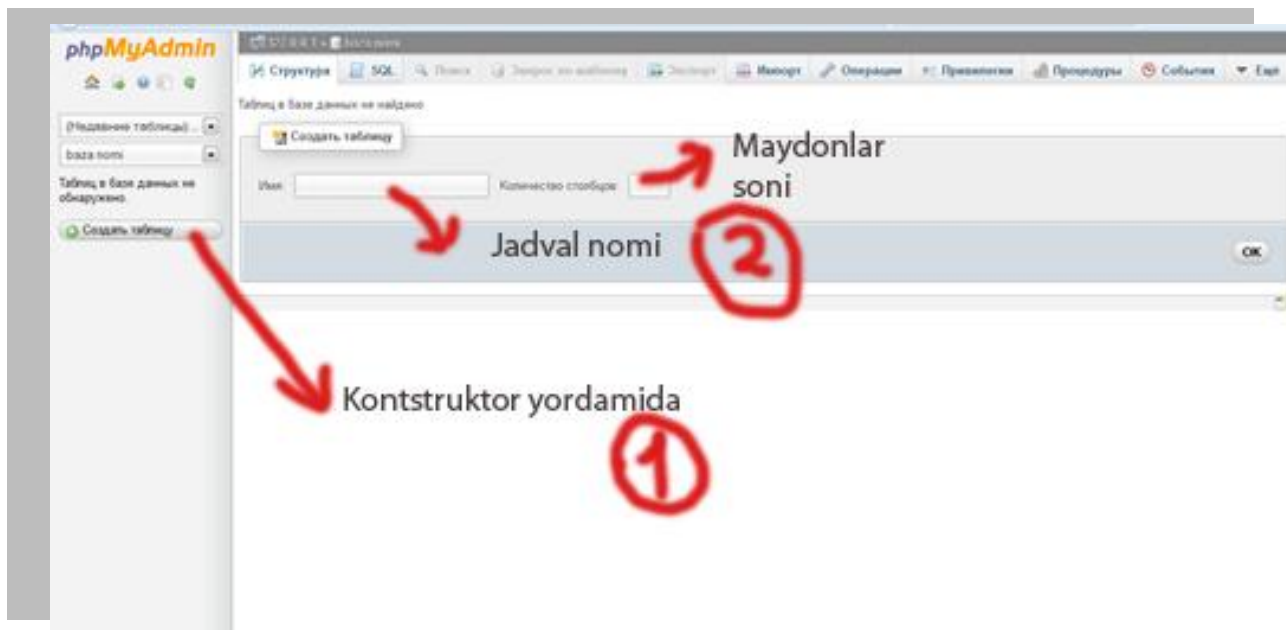
### 3.1. Berilganlar bazasidagi jadvallar sxemasi

Berilganlar bazasini yaratishda MySQL bazasida foydalanildi. MySQLda baza yaratish uchun denwerni o‘rnatib ixtiyoriy brovserni ochib adresga localhost/phpmyadmin yoziladi va “Базы данных” tugmasi bosiladi.” Создать базу данных” degan joyiga bazani nomi yoziladi va “Создать” tugmasi bosiladi (5 - rasm).



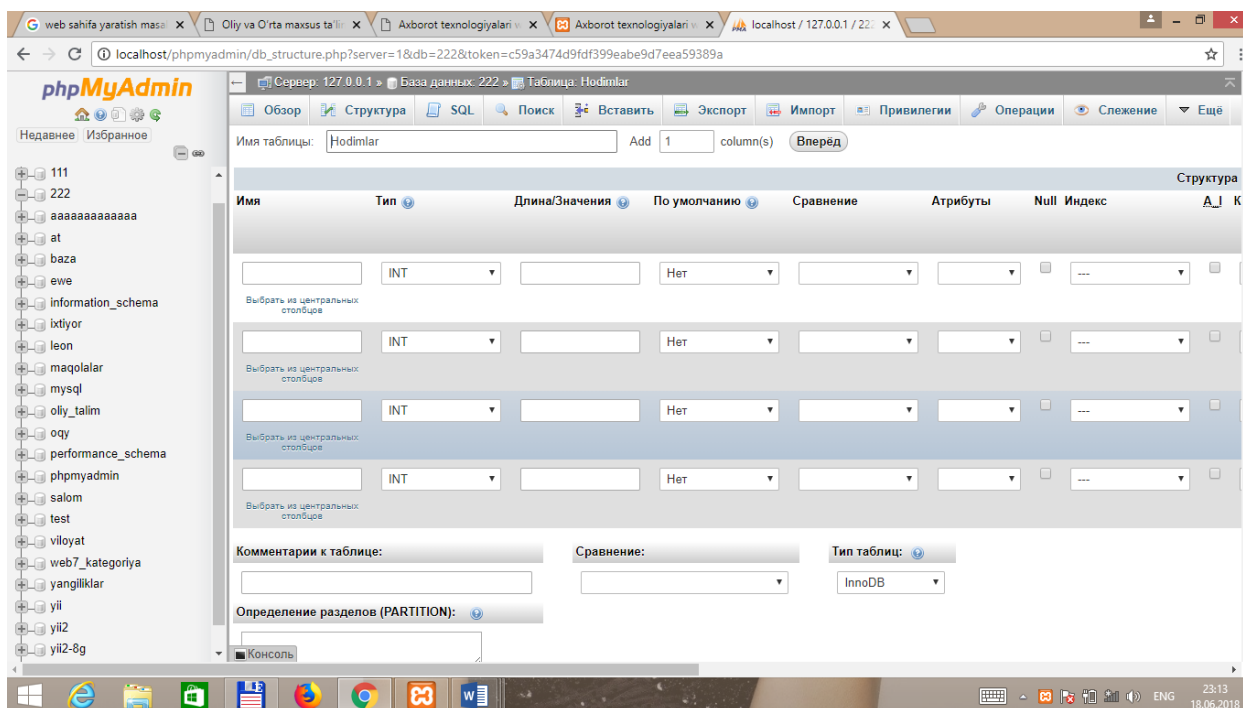
5 - rasm. Yangi baza yasash sahifasi.

Baza yaratilgandan so‘ng jadvallar yaratiladi. Jadval 2-xil (6-rasm) usul bilan yaratiladi:



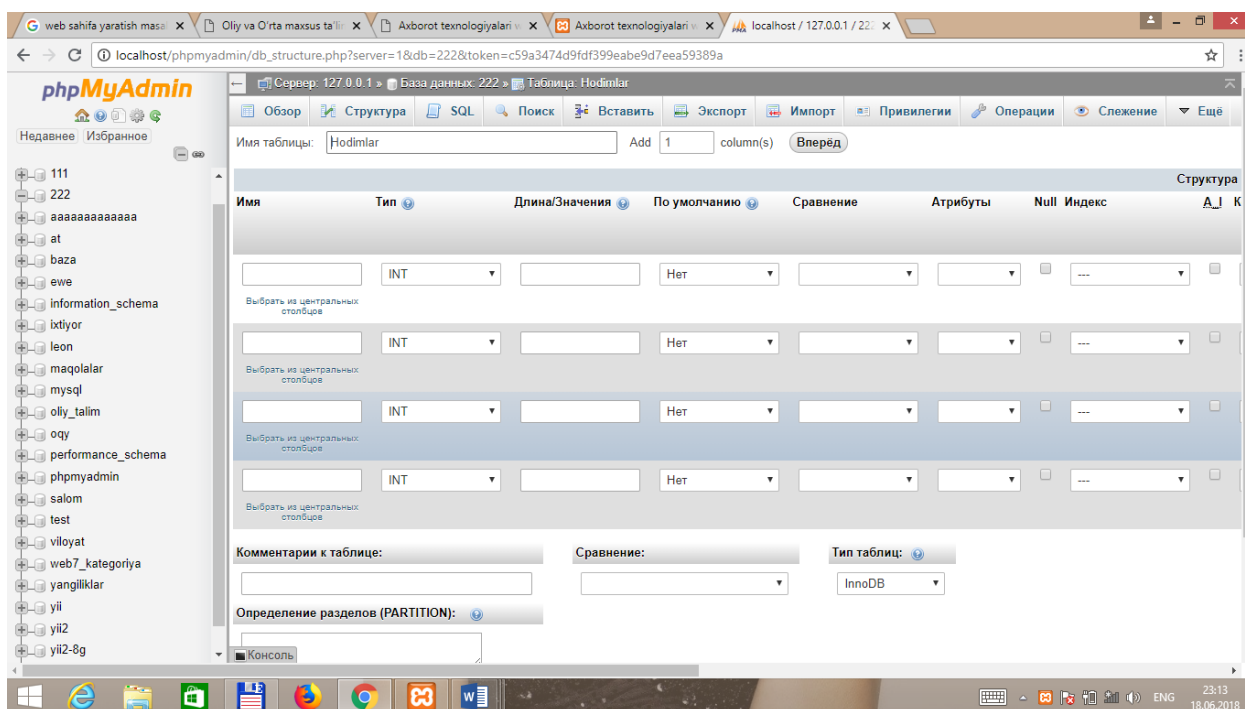
6- rasm. Yangi jadval yaratish sahifasi

1. Konstruktor yodamidan ya'ni bu yerda konstruktor 4 ta maydonli jadvalni avtomatik ravishda yaratib beradi.



7- rasm. Konstruktor yordamida yaratilgan jadval.

2. Bunda jadval nomi va maydonlari soni kiritiladi va shunga mos maydonlar hosil bo'ladi(8-rasm).



8-rasm.Konstruktorsiz yaratilgan jadval.

Yuqorida ko‘rsatilgan qadamlarni qilgan holda quyidagi jadvallarni yaratamiz(9-rasm):

oqituvchi - jadvalda kafedra o‘qituvchilar haqidagi ma’lumotlar saqlanadi.

davomat –talabalarining umumiy davomati haqidagi ma’lumotlar saqlanadi.

guruh – kafedra umumiy guruhulari haqidagi ma’lumotlar saqlanadi.

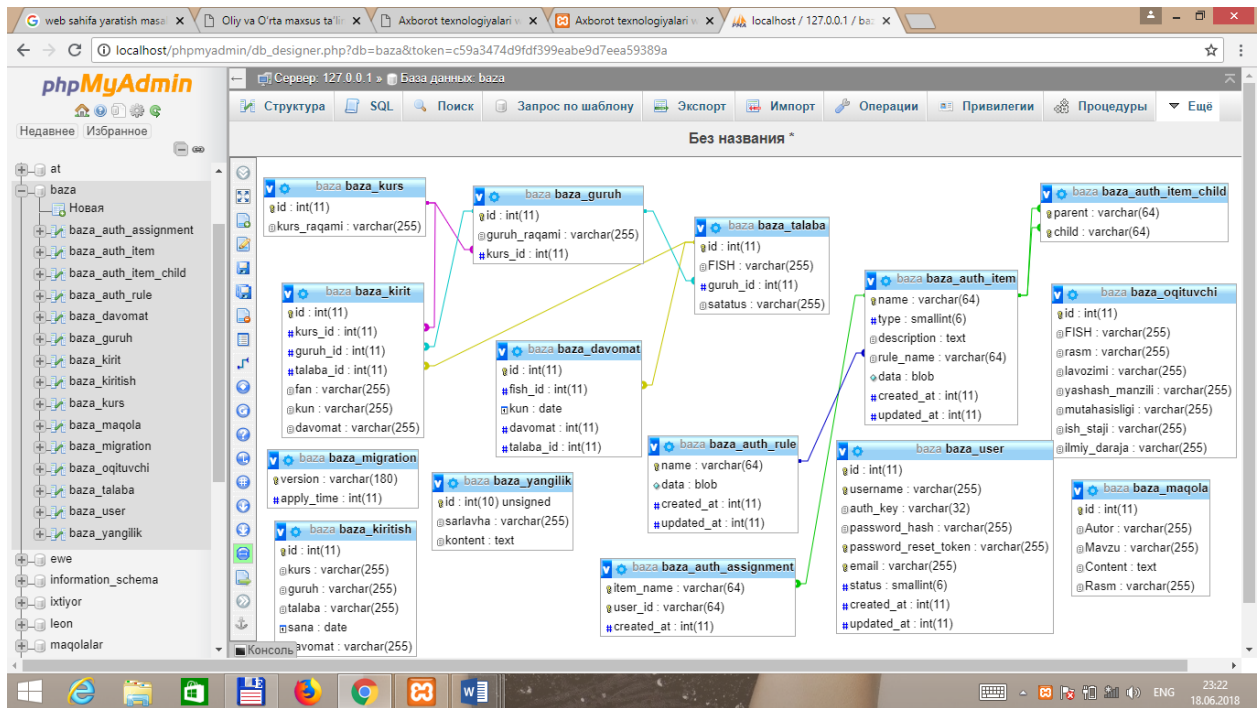
kiritish – kafedra umumiy talabalari haqidagi davomatlarini kiritish uchun.

kurs – kafedra umumiy 1, 2, 3, 4 kurslari ma’lumotlari saqlanadi.

maqola – ilmiy ishlari, Maqolalari va tezislar haqida ma’lumotlar saqlanadi.

talaba – kafedra umumiy talabalari haqidagi ma’lumotlar saqlanadi.

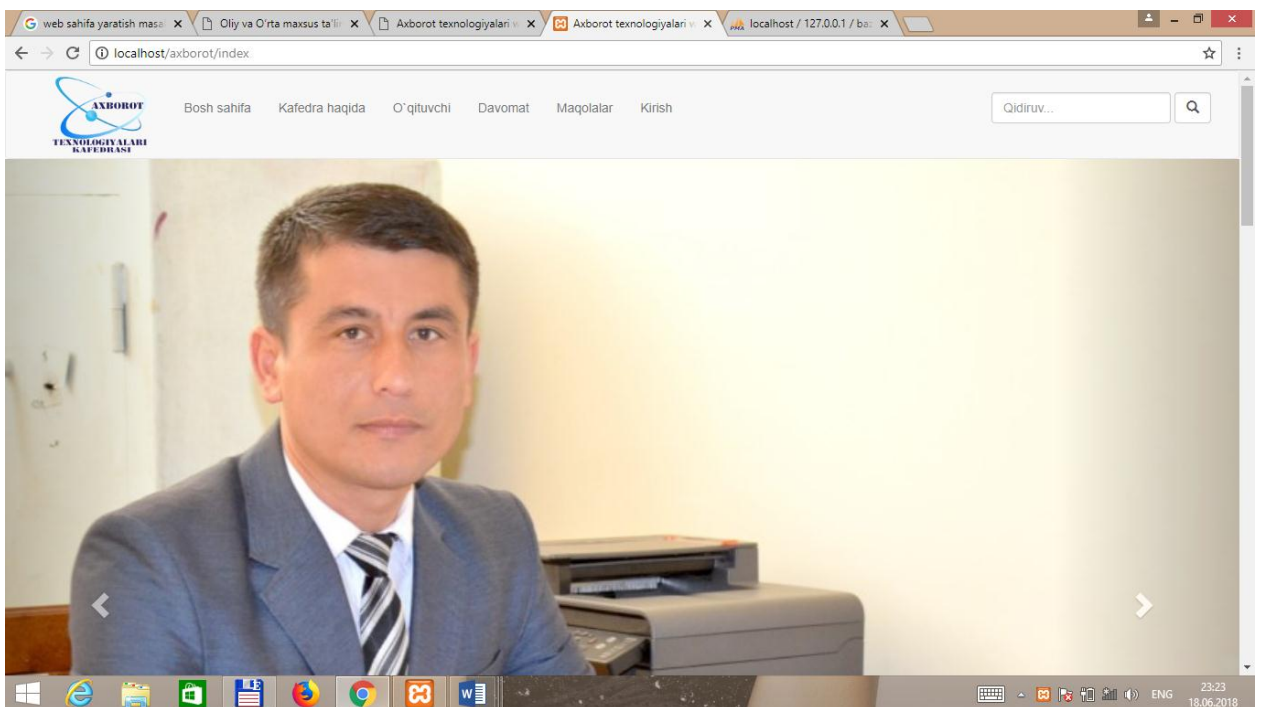
yangilik - kafedra umumiy yangiliklari haqidagi ma’lumotlar saqlanadi.

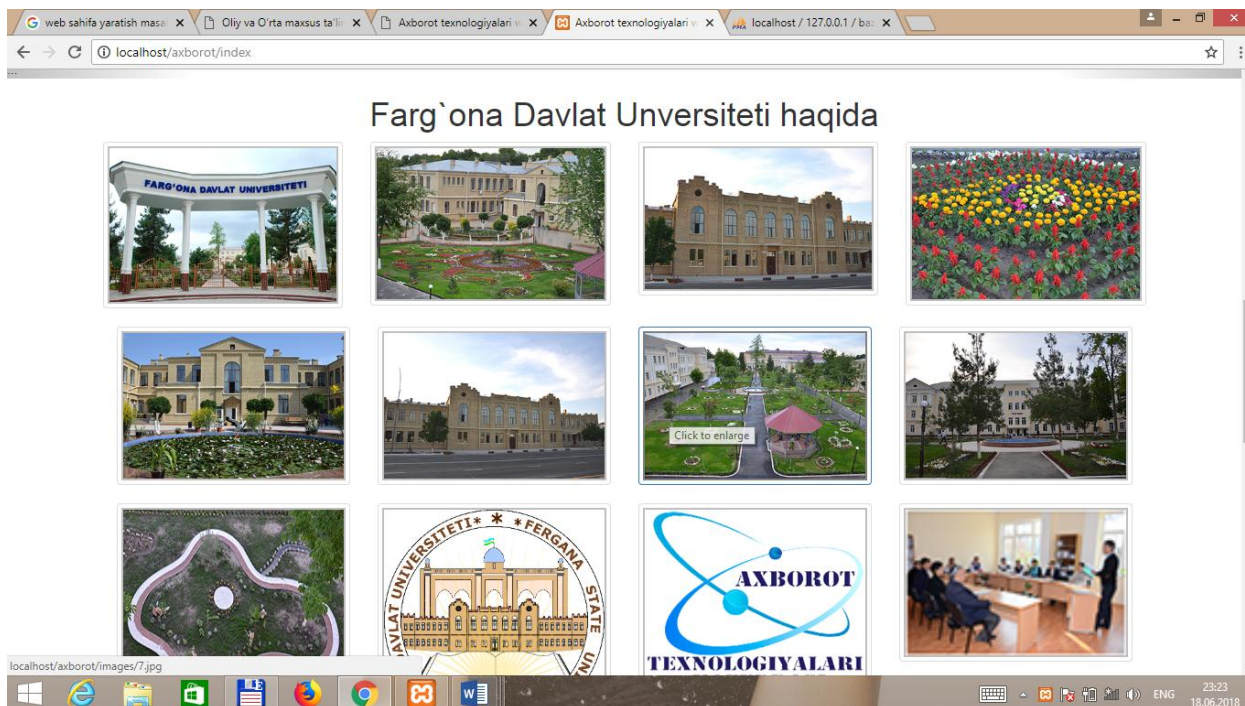


9-rasm. Berilganlar bazasidagi jadvallar sxemasi umumiy ko‘rinishi

### 3.2. Tizimning tavsifi

**3.2.1. Tizimga kirish.** Tizimni (localhostda) ishga tushirish uchun denwer (apache, mysql) dasturi o‘rnatiladi va denwerga start berib, host nomi orqali browserda ishga tushiriladi. Host nomi yozilganda quyidagi ko‘rinishda sahifa ochiladi, bunda saxifaning asosiy qismi ko‘rsatilgan (10-rasm).





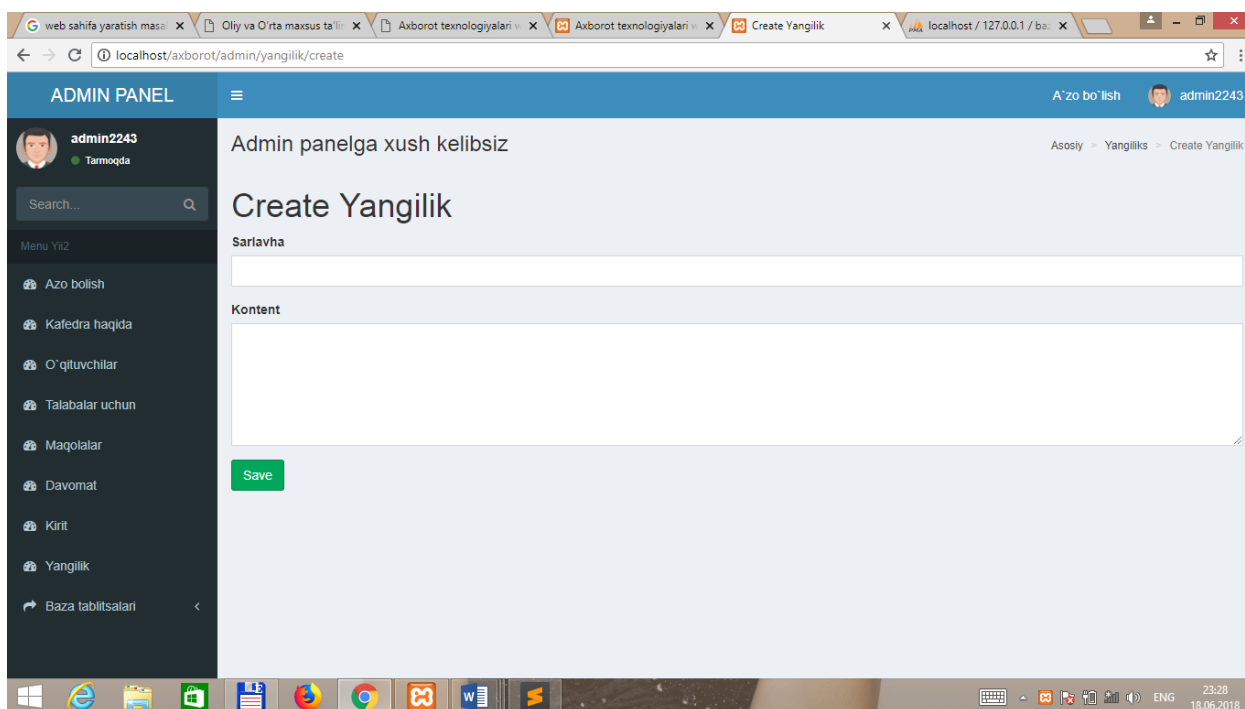
10-rasm. Tizimga kirish sahifasi.

Ikkinchi saxifada esa kafedra faoliyatiga tegishli yangiliklar joy olgan (11-rasm).



11-rasm. Tizimning yangiliklari saxifasi.

Quyidagi saxifada kafedrada chop qilingan maqolalar ro'yxati (12-rasm).



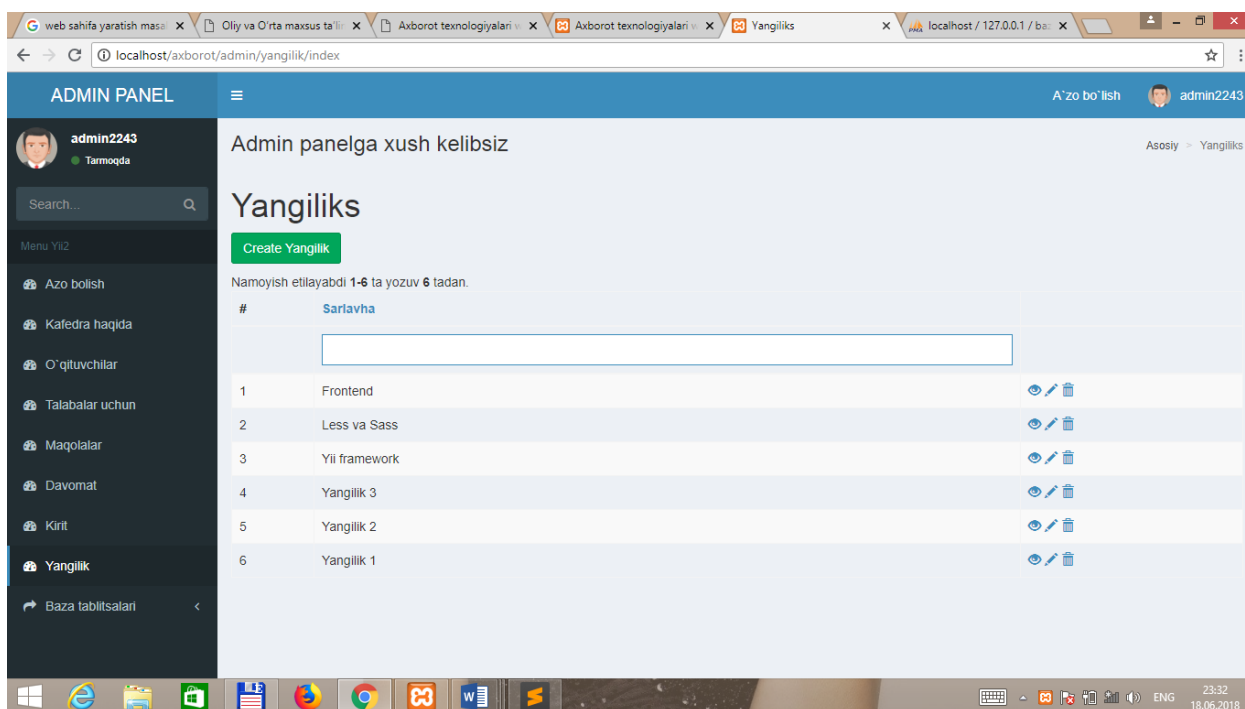
12-rasm. Tizimning maqolalar ro'yxati.

Yuqoridagi saxifada web sahifaning admin paneli ko'rsatilgan bu bo'limda biz maqola kiritamiz sarlavhasi va maqola matni kiritiladi, yuqoridagi maydonlarning tavsifi quyidagicha:

1. sarlavha – bu maydonga maqolaning sarlavhasi yoziladi;
2. kontent – bu maydonga maqolaning matn qismi kiritiladi;

Yuqoridagi maydonlar maqolalar ro'yxatini filtirlash uchun ishlatiladi. Barcha maydonlarni bir vaqtning o'zida to'ldirish shart emas.

Quyidagi saxifada kafedra o'qituvchilar va talabalar ro'yxati keltirilgan (13 - rasm)



13–rasm.Hodimlar ro‘yxati.

Yuqoridagi saxifadagi maydonlarning tavsifi quyidagicha:

Sarlavha ostidagi ya'ni jadvalda joylashgan maydonga qidirilayotgan so'z kiritiladi.

Yuqoridagi maydonlar maqolalar ro'yxatini filtirlash uchun ishlatiladi.

Barcha maydonlarni bir vaqtning o'zida to'ldirish shart bo'ladi.

Tanlash amali bajarilgandan keyin bizga ro'yxat chiqadi. Keyin o'zimizga kerakli ma'lumotlar chiqadi.Ulardan shu shaxs haqida ma'lumotlar olamiz,o'zimizga kerakli ma'lumotlarni chop qilib olishimiz mumkin.

Quyidagi saxifada kafedraning davomatlar ro'yxati keltirilgan  
(14 - rasm)

web sahifa yaratish mas... x Olfiy va O'rita maxsus ta... x Axborot texnologiyalari x Axborot texnologiyalari x Yangilik x localhost / 127.0.0.1 / ba... x

localhost/axborot/davomat/index

AXBOROT  
TEKNOLOGIYALARI  
KAFEDRASI

Bosh sahifa Kafedra haqida O'qituvchi Davomat Maqolalar Kirish

Qidiruv... Q

## Davomatlar jadvali

Davomatlar jadvalini boshqarish bo'limiga o'tish

Namoyish etilayabdi 1-4 ta yozuv 4 tadan.

#	Kurs	Guruh	Talaba	Fan	Kun	davomat
	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
1	1-kurs	106-guruh	Asrayev I	Tizim tarmoqlari	2018-06-17 11:35	yoq
2	1-kurs	105-guruh	Azimov F	Kompyuter grafikasi	2018-06-06 11:35	yoq
3	1-kurs	105-guruh	Azimov A	Kompyuter grafikasi	2018-06-17 11:40	yoq
4	1-kurs	105-guruh	Azimov F	Kompyuter grafikasi	2018-06-18 22:50	yoq

Murojat uchun tel: +99891 676 77 59 Xaydarov Ibrohimjon Tuzuvchi: Xo'jaqulov Ixtiyorjon

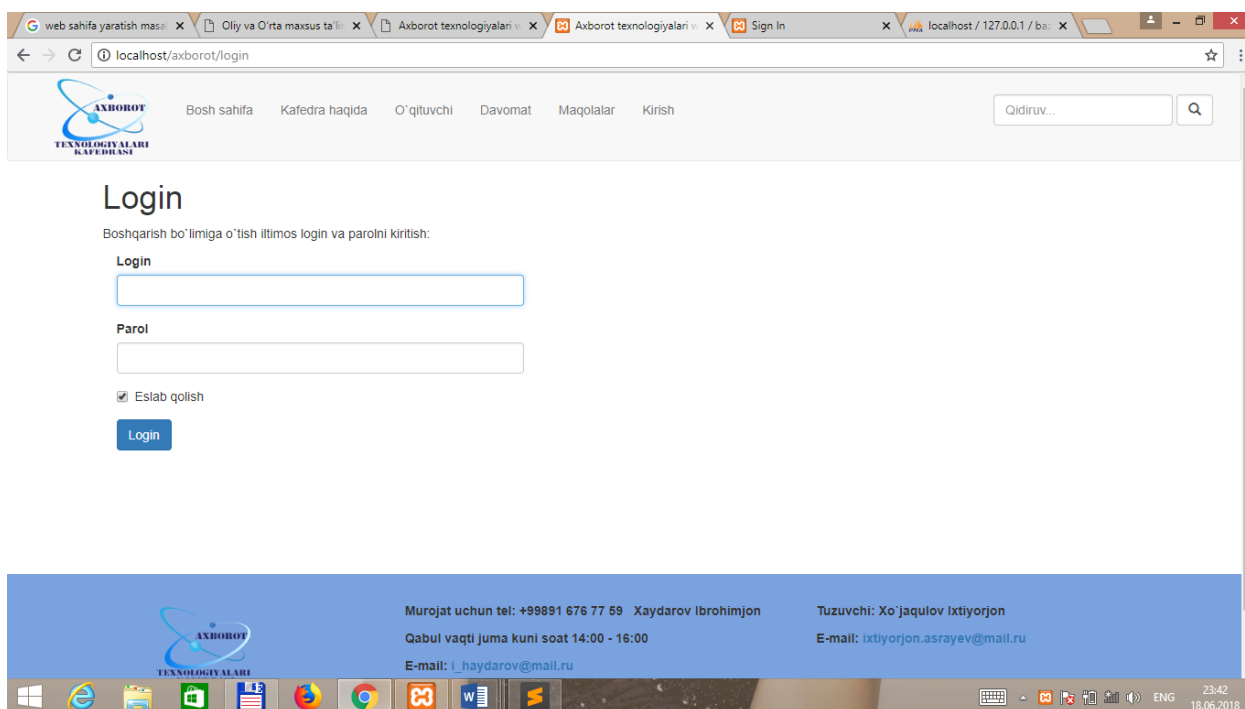
23:36  
18.06.2018

14-rasm. Davomatlar jadvali.

Yuqoridagi maydonlar davomatlar jadvali filtirlash uchun ishlatiladi.

1. Kurs – bu kafedraning umumiy 1, 2, 3, 4 kurslari ro‘yhatini filterlashimiz mumkin;
2. Guruh - bu kafedraning umumiy Guruhlar ro‘yhatini filterlashimiz mumkin;
3. Talaba – talabaning ismi familiyasi bo‘yicha malumotlar ro‘yhatini filterlashimiz mumkin;
4. Fan – bu kafedraning umumiy fanlar ro‘yhatini filterlashimiz mumkin;
5. Kun - bu kiritilgan sanalar ro‘yhatini filterlashimiz mumkin;
6. Davomat – jadval bo‘yicha barcha kelmagan talabalar yoki kelgan talabalarni filterlashimiz mumkin.

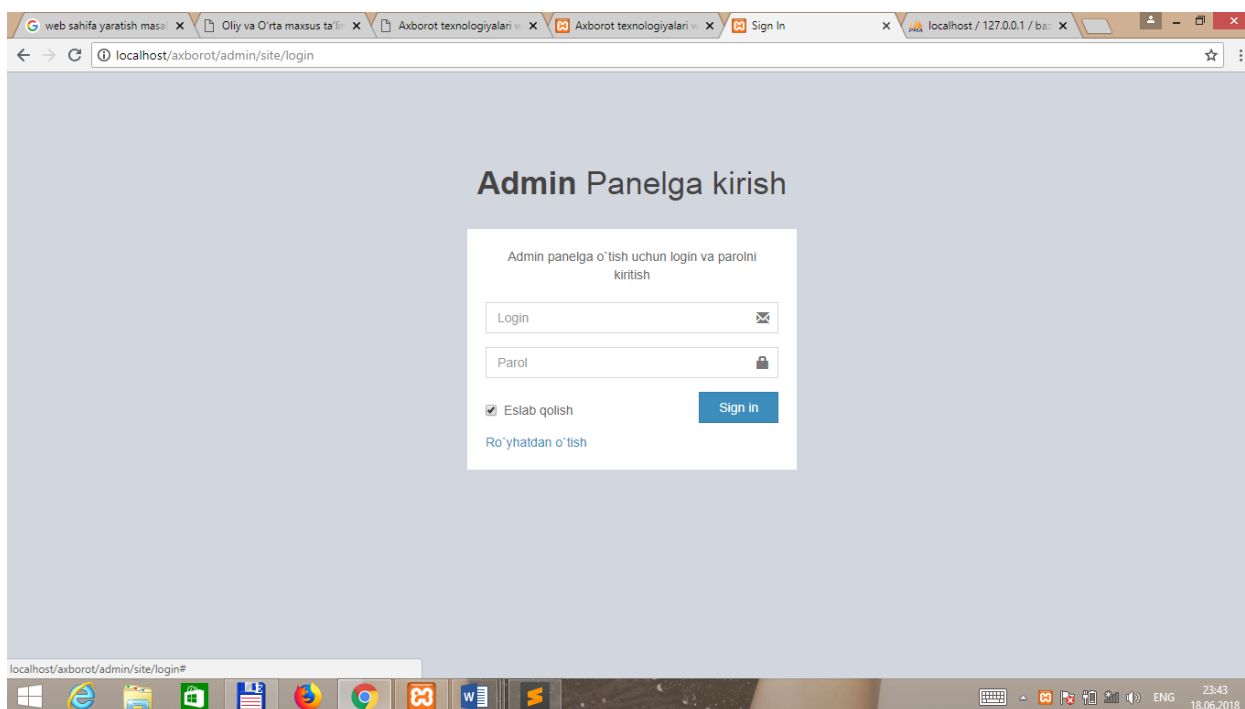
Quyidagi saxifada foydalanuvchining kirish paneli keltirilgan. (15-rasm)



15 – rasm. Kirish paneli

Yuqoridagi maydonlarda foydalanuvchi “login” va “parolni” kiritadi va “Login” tugmasini bosadi va ro‘yxatdan o‘tgandan keyin foydalanuvchi ma’lumotlarini o‘zgartirish imkoniyatiga ega bo‘ladilar.

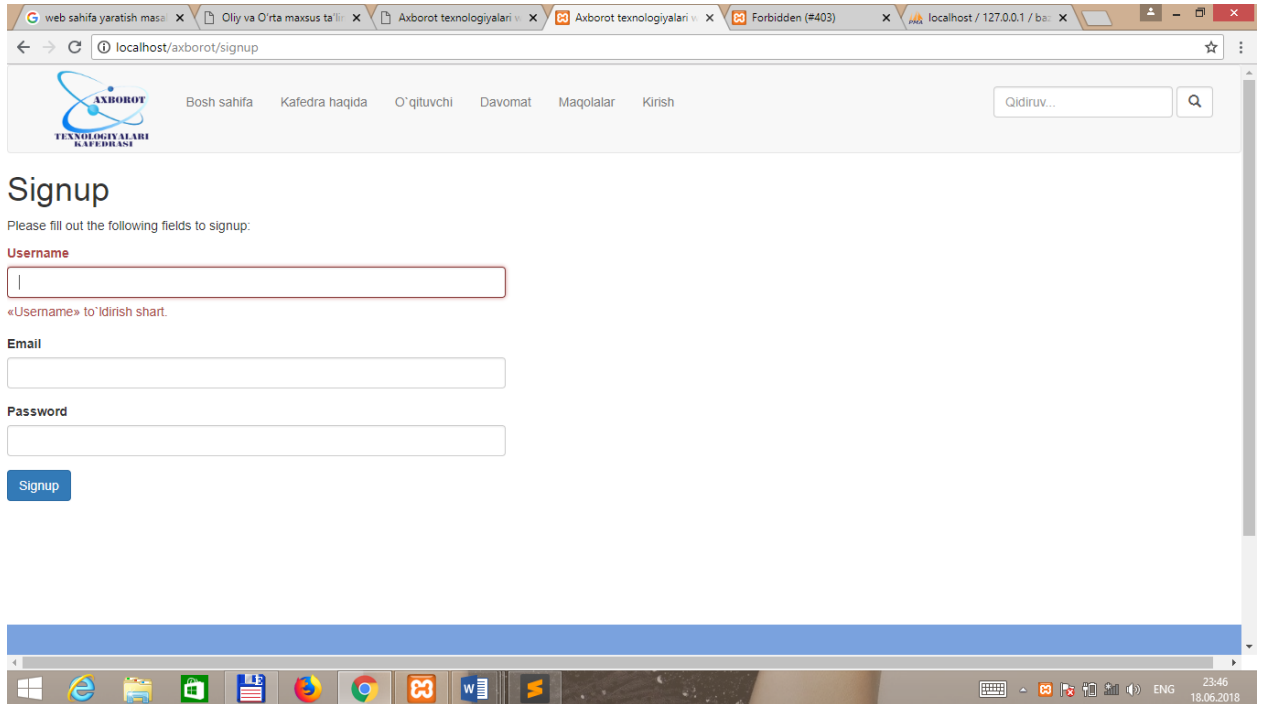
Quyidagi saxifada admin moduli saxifasi keltirilgan. (16-rasm)



## 16-rasm. Admin panel

Yuqoridagi maydonlarda admin o'zining "Login" va "Parol" ini kiritib tizimning ma'lumotlarini o'zgartirib turadi.

Quyidagi saxifada esa ro'yxatdan o'tish paneli keltirilgan. (17-rasm)



## 17-rasm. Ro'yxatdan o'tish

Yuqoridagi maydonlarning tavsifi quyidagicha:

1. Login– bu maydonga foydalanuvchining logini yoziladi;
2. mail - bu maydonga foydalanuvchining E-mail i yoziladi;
3. Password - bu maydonga foydalanuvchining paroli yoziladi;

### 3.3. Tizimning qulayligi va avzalligi.

Yaratilayotgan loyhalar, dasturlar yoki dasturlar paketining qay darajada qulayligi va avzalligi alohida kasb etadi. Ushbu web-texnologiyalar asosida yaratilgan kafedra tizimi ham bir qancha qulayliklarga ega:

-Ixtiyoriy sistema foydalanuvchisi internet tarmog'i orqali sistemada ixtiyoriy joyda ishlay olishi;

- Dastur o'zbek tilda ekanligi;
- Foydalanuvchi ushbu sistemada ma'lumotlarni qidirish jarayonida informatsion sohada taaluqli bilimlarni talab qilmaydi;
- Sahifaning sodda strukturaga ega ekanligi;
- Kafedra haqida umumiy ma'lumotlar berilishi;
- Kafedraning so'ngi yangiliklari haqida ma'lumotlar olish;
- Kafedrada ishlab chiqilayotgan maqolalarni ko'rish;
- Kafedrada talabalar davomatni shakllantirish;
- Kafedrada onlayn jurnal tizimini yo'lga qo'yish;

## **Xulosa va takliflar**

Ushbu bitiruv malakaviy ishini yozish davomida ko'pgina ma'lumotlar bilan tanishdim. Shulardan web-texnologiyalardan foydalanishni, ularni hayotdagi ahamiyatini yoki hayotga taqbiq qilishdagi rolini, berilganlar bazasini boshqarishni, unda turli murakkab so'rovlar yozishni o'rgandim .

Zamonaviy texnologiyalar jadallik bilan rivojlanyotgan kunda, dasturlash tillarining ham yangidan yangi turlari ishlab chiqilmoqda. Shu bilan birga ularning ishlash prinsiplari tobora mukammallashib foydalanuvchi ishlashi uchun qulay imkoniyatlar tug'ilmoqda.

Web texnologiyalari yordamida har bir yaratilgan tizim ayrim sohalarning avtomatlashishiga sabab bo'lmoqda. Avtomatlashgan tizimlar esa inson mehnati kamaytiradi va u bir qancha qulayliklar yaratadi. Har bir sohani avtomatlashtirish esa ish unumdorligini, ishning sifatini oshiradi va eng muhimi vaqtni tejaydi.

Yangi texnologiyalarni o'rganish, ularni ishlatish va ular uchun dasturlar tuzish va hayotga tadbiq qilish ham zamonamizning dolzarb masalalaridan biriga aylandi. Ushbu malakaviy ishini bajara turib web-texnologiyalar yordamida dastur yozish usullarini, yangi web texnologiyalarni (bootstrap, less), ularning yozilishi, rivojlanish tarixini va bir qancha shu dastur bilan bog'liq bo'ladigan hodisalarni o'rganib chiqdim. Dasturlarning xilma xilligi meni hayratga soldi, o'zim uchun yangi bo'lgan dasturlarning turlari bilan tanishib chiqdim.

Ishlash davomida men uchun tanish bo'lgan dastur PHP tilidan va berilganlar bazasini(MySql) boshqarish tizimlaridan foydalangan bo'lsakda undagi malum tushunchalar haqida yanada ko'roq malumot oldik va ular ustida bitiruv malakaviy ishini yaratish davomida ishladik. Menga malum bo'lgan tushunchalar bir qatorda noma'lum ya'ni texnologiyalarni o'rganishda, ularning ma'nosini tushunishda, ularni web sahifa tizimga tadbiq qilishda bir qancha muammolarga duch keldim va ular bartaraf qilishda oldingi bilimlarimni yanada mukammallashtirishga to'g'ri keldi. Bu orqali tajribam va dasturlash sohasidagi bilimlarimni oshirdim. va o'zim

uchun yangi bo'lgan ma'lumotlarni, yangi texnologiyalarni, dasturlashdagi bilimlarimni o'zlashtirdim.

Barchasini xulosa qilib aytadigan bo'lsak bitiruv malakaviy ishini bajarib shunday tushunchaga ega bo'ldimki: bootstrap va less texnologiyalari dasturiy vositalar bilan birgalikda ularning tuzilishi, ularning dasturiy boshqarish usullarini o'rganish, shu o'rganilgan tushunchalarni hayotda keng joriy etishdadur deb o'ylayman.

### **Takliflar**

Kafedra tizimini yangi texnologilardan foydalanib yanada rivojlantirsa bo'ladi. Bunga misollar qilib

-o'qituvchi va talaba shaxsiy kabinetini shakillantirsa bo'ladi, bunda o'qituvchi o'zining kabinetida shaxsiy fayllar, rasmlar, turli hil taqidmotlar va boshqa malumotlarini saqlashi va uni hohlagan payt hohlagan joydan foydalanishni taminlashimiz mumkin, shu kabi talaba ham;

-kafedrada virtual qabuxona tizimini shakillantirishimiz mumkin;

-web sahifada onlayn dars jadvalini tashkil etishimiz va talabaga yanada qulayliklar yaratishimiz mumkin;

va yanada inavatsion g'oyalar asosida kafedra web sahifasini yanda yaxshiroq shakillantirib amaliyotga tadbiq qilishimiz mumkin.

## Fodalanilgan adabiyotlar

1. [http://uzhurriyat.uz/foydali\\_saytlar/akt\\_taraqqiyotga\\_xizmat\\_qiladi.mgr](http://uzhurriyat.uz/foydali_saytlar/akt_taraqqiyotga_xizmat_qiladi.mgr)
2. Веллинг Л., Томсон Л. Разработка Web-приложений с помощью PHP и MySQL,3-е издание, 2008
3. Мориссон М, Бейли Л, Изучаем PHP и MySQL ,2010,Москва
4. А. Мазуркевич Д, Еловой ,PHP настольная книга программиста,2010, Санкт-Петербург
5. <http://yiiframework.com>
6. <http://yiiframework.ru>
7. <http://getbootstrap.com>
8. <http://wrapbootstrap.com>
9. <http://ru.wikipedia.org/wiki/AJAX>
10. Axborot-kommunikatsiya texnologiyalari izohli lug'ati, 2004, UNDP DDI:
11. www.lugat.uz, www.glossaiy.uz
12. O‘zbek internet resurslarining katalogi: www.uz
13. Infocom.uz elektron jurnali: www.infocom.uz
14. ww.ziyonet.uz
15. www.edu.uz
16. www.WYSIWYGWebBuilder.ru

# Ilova

```
<!DOCTYPE html>
<html lang="uz-UZ">
<head>
<meta charset="UTF-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1">
<meta name="csrf-param" content="_csrf-frontend">
<meta name="csrf-token"
content="0cCAcz2gPnnpI7LW150ubRBwsrZjDPBPB3pmKuPURFGBkthAUJByP7IMioClrUt
ZWBOL8RFpgileEh99iOUVBA==">
<title>Axborot texnologiyalari web sahifasi</title>
<link href="/axborot/assets/77fd3cf5/css/bootstrap.css" rel="stylesheet">
<link href="/axborot/css/site.css" rel="stylesheet">
<link
href="https://fonts.googleapis.com/ss?family=Lora:400,700,400italic,700italic&amp;subse
t=latin" rel="stylesheet">
<link href="https://fonts.googleapis.com/css?family=Montserrat:400,700" rel="stylesheet">
<link
href="https://fonts.googleapis.com/css?family=Raleway:100,100i,200,200i,300,300i,400,400i,5
00,500i,600,600i,700,700i,800,800i,900,900i" rel="stylesheet">
<link href="/axborot/css/animate.css/animate.min.css" rel="stylesheet">
<link href="/axborot/css/animate.css/animate.css" rel="stylesheet">
<link href="/axborot/css/highslide/highslide.css" rel="stylesheet">
<style>
#row{
width:100%;
padding-left:0;
margin-left:0;
background:#ccc;
}
</style>
<style>
.navbar{
margin-bottom:0px;
padding-top:15px;
}
#logotip{
margin-right:10px;
margin-left:20px;
height: 90px;
```

```

margin-top:-10px;
padding-bottom:5px;
}
#menu li:hover a{
background:green;
color:white;
border:1px;
border-radius:10px;
transition:1s;
}
#menu li:hover button{
background:green;
color:white;
border:1px;
border-radius:10px;
transition:1s;
}
#menu li:hover{
background:green;
color:white;
border:1px;
border-radius:10px;
transition:1s;
}
#menu li:hover button{
background:green;
color:white;
}
#menu li a:hover{
background:green;
color:white;
}
</style> <style type="text/css">
body{
transition:
}
</style>
</head>
<body>
<nav class="navbar navbar-default navbar-fixed">
<div class="container-fluid">
<!-- Brand and toggle get grouped for better mobile display -->

```

```

<div class="navbar-header">
<button type="button" class="navbar-toggle collapsed" data-toggle="collapse" data-
target="#bs-example-navbar-collapse-1" aria-expanded="false">
<span class="sr-only">Toggle navigation</span>
<span class="icon-bar"></span>
<span class="icon-bar"></span>
<span class="icon-bar"></span>
</button>
<div class="col-sm-12">
<a href="/axborot/index" class="navbar-logo"></a>
</div>
</div>
<!-- Collect the nav links, forms, and other content for toggling -->
<div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1">
<ul id="menu" class="nav navbar-nav">
<li class="" >
<a href="/axborot/index">Bosh sahifa</a></li>
<li class="">
<a href="/axborot/kafedra/index">Kafedra haqida</a>
</li>
<li class="">
<a href="/axborot/oqituvchi/index">O‘qituvchi</a>
</li>
<li>
<a href="/axborot/davomat/index">Davomat</a>
</li>
<li>
<a href="/axborot/maqola/index">Maqolalar</a>
</li>
<li><a id="modalButton" href="/axborot/login">Kirish</a></li>
</ul>

<form action="/axborot/qidiruv" class="navbar-form navbar-right" role="search">
<div class="form-group">
<input type="text" class="form-control" name="q" placeholder="Qidiruv..." required>
</div>
<button type="submit" class="form-control"><span class="glyphicon glyphicon-
search"></span></button>
</form>
</div><!-- /.navbar-collapse -->
</div><!-- /.container-fluid -->
</nav>

```

```

<div id="modal" class="fade modal" role="dialog" tabindex="-1">
<div class="modal-dialog">
<div class="modal-content">
<div class="modal-header">
<button type="button" class="close" data-dismiss="modal" aria-
hidden="true">&times;</button>
</div>
<div class="modal-body">
<div id="modalContent"></div>
</div>
</div>
</div>
</div><div class="site-index">
<div id="carousel-example-generic" class="carousel slide" data-ride="carousel">
<!-- Indicators -->
<ol class="carousel-indicators">
<li data-target="#carousel-example-generic" data-slide-to="0" class="active"></li>
<li data-target="#carousel-example-generic" data-slide-to="1"></li>
<li data-target="#carousel-example-generic" data-slide-to="2"></li>
<li data-target="#carousel-example-generic" data-slide-to="3"></li>
</ol>
<!-- Wrapper for slides -->
<div class="carousel-inner" role="listbox">
<div class="item active">

<div class="carousel-caption">
Axborot texnologiyalari kafedراس
</div>
</div>
<div class="item">

<div class="carousel-caption">
Axborot texnologiyalari kafedراس
</div>
</div>
<div class="item">

<div class="carousel-caption">
Axborot texnologiyalari kafedراس mudiri
</div>
</div>
</div>

```

```

<div class="item">

<div class="carousel-caption">
Axborot texnologiyalari kafedraasi hodimlari
</div>
</div>
...
</div>
<!-- Controls -->
<a class="left carousel-control" href="#carousel-example-generic" role="button" data-
slide="prev">
<span class="glyphicon glyphicon-chevron-left" aria-hidden="true"></span>
<span class="sr-only">Previous</span>
</a>
<a class="right carousel-control" href="#carousel-example-generic" role="button" data-
slide="next">
<span class="glyphicon glyphicon-chevron-right" aria-hidden="true"></span>
<span class="sr-only">Next</span>
</a>
</div>
<!-- dlkfhsjlfj -->
<center><h1>Farg‘ona Davlat Unversiteti haqida</h1></center>
<div class="container-fluid maxsulot">
<div class="container">
<div class="row">
<div class="col-lg-3 col-xs-12 col-md-4 col-sm-6 ">
<div class="highslide-gallery ">
<a href="images/4.jpg" class="highslide thumbnail produkt" onclick="return hs.expand(this)">

</a>
</div>
</div>
<div class="col-lg-3 col-xs-12 col-md-4 col-sm-6 ">
<div class="highslide-gallery ">
<a href="images/8.jpg" class="highslide thumbnail produkt" onclick="return hs.expand(this)">

</a>
</div>
</div>
<div class="col-lg-3 col-xs-12 col-md-4 col-sm-6 ">
<div class="highslide-gallery ">
<a href="images/3.jpg" class="highslide thumbnail produkt" onclick="return hs.expand(this)">

```

```


</a>
</div>
</div>
<div class="col-lg-3 col-xs-12 col-md-4 col-sm-6 ">
<div class="highslide-gallery">
<a href="images/10.jpg" class="highslide thumbnail produkt" onclick="return hs.expand(this)">

</a>
</div>
</div>
</div>
<div class="col-lg-3 col-xs-12 col-md-4 col-sm-6 ">
<div class="highslide-gallery ">
<a href="images/5.jpg" class="highslide thumbnail produkt" onclick="return hs.expand(this)">

</a>
</div>
</div>
<div class="col-lg-3 col-xs-12 col-md-4 col-sm-6 ">
<div class="highslide-gallery">
<a href="images/6.jpg" class="highslide thumbnail produkt" onclick="return hs.expand(this)">

</a>
</div>
</div>
<div class="col-lg-3 col-xs-12 col-md-4 col-sm-6 ">
<div class="highslide-gallery">
<a href="images/7.jpg" class="highslide thumbnail produkt" onclick="return hs.expand(this)">

</a>
</div>
</div>
<div class="col-lg-3 col-xs-12 col-md-4 col-sm-6 ">
<div class="highslide-gallery">
<a href="images/11.jpg" class="highslide thumbnail produkt" onclick="return hs.expand(this)">

</a>
</div>

```

```

</div>
<div class="col-lg-3 col-xs-12 col-md-4 col-sm-6 ">
<div class="highslide-gallery">
<a href="images/9.jpg" class="highslide thumbnail produkt" onclick="return hs.expand(this)">

</a>
</div>
</div>
<div class="col-lg-3 col-xs-12 col-md-4 col-sm-6 ">
<div class="highslide-gallery">
<a href="images/1.png" class="highslide thumbnail produkt" onclick="return hs.expand(this)">

</a>
</div>
</div>
<div class="col-lg-3 col-xs-12 col-md-4 col-sm-6 ">
<div class="highslide-gallery">
<a href="images/14.png" class="highslide thumbnail produkt" onclick="return
hs.expand(this)">

</a>
</div>
</div>
<div class="col-lg-3 col-xs-12 col-md-4 col-sm-6 ">
<div class="highslide-gallery ">
<a href="images/12.jpg" class="highslide thumbnail produkt" onclick="return hs.expand(this)">

</a>
</div>
</div>
</div>
</div>
</div>
</div>
<!-- sa;ldmlsakd -->
<!-- yangilik -->
<div class="container">
<div class="panel panel-primary">
<div class="panel-heading">Yangiliklar Bo‘limi</div>

```





```

var xhr = window.XMLHttpRequest ? new XMLHttpRequest() : new
ActiveXObject('Microsoft.XMLHTTP');
settings = settings || {};
xhr.open(settings.method || 'GET', url, true);
xhr.setRequestHeader('X-Requested-With', 'XMLHttpRequest');
xhr.setRequestHeader('Accept', 'text/html');
xhr.onreadystatechange = function (state) {
if (xhr.readyState === 4) {
if (xhr.status === 200 && settings.success) {
settings.success(xhr);
} else if (xhr.status !== 200 && settings.error) {
settings.error(xhr);
}
}
};
xhr.send(settings.data || "");
},
url,
div,
toolbarEl = findToolbar(),
toolbarAnimatingClass = 'yii-debug-toolbar_animating',
barSelector = '.yii-debug-toolbar__bar',
viewSelector = '.yii-debug-toolbar__view',
blockSelector = '.yii-debug-toolbar__block',
toggleSelector = '.yii-debug-toolbar__toggle',
externalSelector = '.yii-debug-toolbar__external',
CACHE_KEY = 'yii-debug-toolbar',
ACTIVE_STATE = 'active',
animationTime = 300,
activeClass = 'yii-debug-toolbar_active',
iframeActiveClass = 'yii-debug-toolbar_iframe_active',
iframeAnimatingClass = 'yii-debug-toolbar_iframe_animating',
titleClass = 'yii-debug-toolbar__title',
blockClass = 'yii-debug-toolbar__block',
blockActiveClass = 'yii-debug-toolbar__block_active',
requestStack = [];
if (toolbarEl) {
url = toolbarEl.getAttribute('data-url');
ajax(url, {
success: function (xhr) {
div = document.createElement('div');
div.innerHTML = xhr.responseText;

```

```

toolbarEl.parentNode && toolbarEl.parentNode.replaceChild(div, toolbarEl);
showToolbar(findToolbar());
},
error: function (xhr) {
toolbarEl.innerText = xhr.responseText;
}
});
}

function showToolbar(toolbarEl) {
var barEl = toolbarEl.querySelector(barSelector),
viewEl = toolbarEl.querySelector(viewSelector),
toggleEl = toolbarEl.querySelector(toggleSelector),
externalEl = toolbarEl.querySelector(externalSelector),
blockEls = barEl.querySelectorAll(blockSelector),
iframeEl = viewEl.querySelector('iframe'),
iframeHeight = function () {
return (window.innerHeight * 0.7) + 'px';
},
isIframeActive = function () {
return toolbarEl.classList.contains(iframeActiveClass);
},
showIframe = function (href) {
toolbarEl.classList.add(iframeAnimatingClass);
toolbarEl.classList.add(iframeActiveClass);
iframeEl.src = externalEl.href = href;
viewEl.style.height = iframeHeight();
setTimeout(function() {
toolbarEl.classList.remove(iframeAnimatingClass);
}, animationTime);
},
hideIframe = function () {
toolbarEl.classList.add(iframeAnimatingClass);
toolbarEl.classList.remove(iframeActiveClass);
removeActiveBlocksCls();
externalEl.href = '#';
viewEl.style.height = '';
setTimeout(function() {
toolbarEl.classList.remove(iframeAnimatingClass);
}, animationTime);
},
removeActiveBlocksCls = function () {
[].forEach.call(blockEls, function (el) {

```

```

el.classList.remove(blockActiveClass);
});
},
toggleToolbarClass = function (className) {
toolbarEl.classList.add(toolbarAnimatingClass);
if (toolbarEl.classList.contains(className)) {
toolbarEl.classList.remove(className);
} else {
toolbarEl.classList.add(className);
}
setTimeout(function () {
toolbarEl.classList.remove(toolbarAnimatingClass);
}, animationTime);
},
toggleStorageState = function (key, value) {
if (window.localStorage) {
var item = localStorage.getItem(key);
if (item) {
localStorage.removeItem(key);
} else {
localStorage.setItem(key, value);
}
}
},
restoreStorageState = function (key) {
if (window.localStorage) {
return localStorage.getItem(key);
}
},
togglePosition = function () {
if (isIframeActive()) {
hideIframe();
} else {
toggleToolbarClass(activeClass);
toggleStorageState(CACHE_KEY, ACTIVE_STATE);
}
};
toolbarEl.style.display = 'block';
if (restoreStorageState(CACHE_KEY) === ACTIVE_STATE) {
toolbarEl.classList.add(activeClass);
}
window.onresize = function () {

```

```

if (toolbarEl.classList.contains(iframeActiveClass)) {
viewEl.style.height = iframeHeight();
}
};

barEl.onclick = function (e) {
var target = e.target,
block = findAncestor(target, blockClass);
if (block && !block.classList.contains(titleClass)
&& e.which !== 2 && !e.ctrlKey // not mouse wheel and not ctrl+click
) {
while (target !== this) {
if (target.href) {
removeActiveBlocksCls();
block.classList.add(blockActiveClass);
showIframe(target.href);
}
target = target.parentNode;
}
e.preventDefault();
}
};

toggleEl.onclick = togglePosition;
}

function findAncestor(el, cls) {
while ((el = el.parentElement) && !el.classList.contains(cls));
return el;
}

function renderAjaxRequests() {
var requestCounter = document.getElementsByClassName('yii-debug-toolbar__ajax_counter');
if (!requestCounter.length) {
return;
}
var ajaxToolbarPanel = document.querySelector('.yii-debug-toolbar__ajax');
var tbodies = document.getElementsByClassName('yii-debug-toolbar__ajax_requests');
var state = 'ok';
if (tbodies.length) {
var tbody = tbodies[0];
var rows = document.createDocumentFragment();
if (requestStack.length) {
var firstItem = requestStack.length > 20 ? requestStack.length - 20 : 0;
for (var i = firstItem; i < requestStack.length; i++) {

```

```

var request = requestStack[i];
var row = document.createElement('tr');
rows.appendChild(row);

var methodCell = document.createElement('td');
methodCell.innerHTML = request.method;
row.appendChild(methodCell);

var statusCodeCell = document.createElement('td');
var statusCode = document.createElement('span');
if (request.statusCode < 300) {
statusCode.setAttribute('class', 'yii-debug-toolbar__ajax_request_status yii-debug-
toolbar__label_success');
} else if (request.statusCode < 400) {
statusCode.setAttribute('class', 'yii-debug-toolbar__ajax_request_status yii-debug-
toolbar__label_warning');
} else {
statusCode.setAttribute('class', 'yii-debug-toolbar__ajax_request_status yii-debug-
toolbar__label_error');
}
statusCode.textContent = request.statusCode || '-';
statusCodeCell.appendChild(statusCode);
row.appendChild(statusCodeCell);

var pathCell = document.createElement('td');
pathCell.className = 'yii-debug-toolbar__ajax_request_url';
pathCell.innerHTML = request.url;
pathCell.setAttribute('title', request.url);
row.appendChild(pathCell);

var durationCell = document.createElement('td');
durationCell.className = 'yii-debug-toolbar__ajax_request_duration';
if (request.duration) {
durationCell.innerHTML = request.duration + " ms";
} else {
durationCell.innerHTML = '-';
}
row.appendChild(durationCell);
row.appendChild(document.createTextNode(' '));

var profilerCell = document.createElement('td');
if (request.profilerUrl) {
var profilerLink = document.createElement('a');
profilerLink.setAttribute('href', request.profilerUrl);
profilerLink.innerHTML = request.profile;
profilerCell.appendChild(profilerLink);
} else {

```

```

profilerCell.innerHTML = 'n/a';
}
row.appendChild(profilerCell);
if (request.error) {
if (state !== "loading" && i > requestStack.length - 4) {
state = 'error';
}
} else if (request.loading) {
state = 'loading'
}
row.className = 'yii-debug-toolbar__ajax_request';
}
while (tbody.firstChild) {
tbody.removeChild(tbody.firstChild);
}
tbody.appendChild(rows);
}
ajaxToolbarPanel.style.display = 'block';
}
requestCounter[0].innerHTML = requestStack.length;
var className = 'yii-debug-toolbar__label yii-debug-toolbar__ajax_counter';
if (state === 'ok') {
className += ' yii-debug-toolbar__label_success';
} else if (state === 'error') {
className += ' yii-debug-toolbar__label_error';
}
requestCounter[0].className = className;
};

var proxied = XMLHttpRequest.prototype.open;
XMLHttpRequest.prototype.open = function (method, url, async, user, pass) {
var self = this;
/* prevent logging AJAX calls to static and inline files, like templates */
if (url.substr(0, 1) === '/' && !url.match(new RegExp("{{ excluded_ajax_paths }}")) {
var stackElement = {
loading: true,
error: false,
url: url,
method: method,
start: new Date()
};
requestStack.push(stackElement);
this.addEventListener("readystatechange", function () {

```

```

if (self.readyState === 4) {
stackElement.duration = self.getResponseHeader("X-Debug-Duration") || new Date() -
stackElement.start;
stackElement.loading = false;
stackElement.statusCode = self.status;
stackElement.error = self.status < 200 || self.status >= 400;
stackElement.profile = self.getResponseHeader("X-Debug-Tag");
stackElement.profilerUrl = self.getResponseHeader("X-Debug-Link");
renderAjaxRequests();
}
}, false);
renderAjaxRequests();
}
proxied.apply(this, Array.prototype.slice.call(arguments));
};

// catch fetch AJAX requests
if (window.fetch) {
var originalFetch = window.fetch;
window.fetch = function(input, init) {
var method;
var url;
if (typeof input === "string") {
method = (init && init.method) || 'GET';
url = input;
} else if (window.Request && input instanceof Request) {
method = input.method;
url = input.url;
}
var promise = originalFetch(input, init);

/* prevent logging AJAX calls to static and inline files, like templates */
if (url.substr(0, 1) === '/' && !url.match(new RegExp("{{ excluded_ajax_paths }}")) {
var stackElement = {
loading: true,
error: false,
url: url,
method: method,
start: new Date()
};
requestStack.push(stackElement);
promise.then(function(response) {
stackElement.duration = response.headers.get("X-Debug-Duration") || new Date() -
stackElement.start;

```

```

stackElement.loading = false;
stackElement.statusCode = response.status;
stackElement.error = response.status < 200 || response.status >= 400;
stackElement.profile = response.headers.get("X-Debug-Tag");
stackElement.profilerUrl = response.headers.get("X-Debug-Link");
renderAjaxRequests();

return response;
}).catch(function(error) {
stackElement.loading = false;
stackElement.error = true;
renderAjaxRequests();

throw error;
});
renderAjaxRequests();
}

return promise;
};
}

})();</script><script src="site/js/web/assets/jquery/jquery.min.js"></script>
<script src="site/js/dist/js/jquery.js"></script>
<script src="/axborot/assets/f577fc0c/jquery.js"></script>
<script src="/axborot/assets/eb057c2c/yii.js"></script>
<script src="/axborot/assets/77fd3cf5/js/bootstrap.js"></script>
<script src="/axborot/js/theme/js/script.js"></script>
<script src="/axborot/js/highslide/highslide-with-gallery.js"></script>
<script src="/axborot/js/bootstrap/js/bootstrap.min.js"></script>
<script src="/axborot/js/dist/js/bootstrap.min.js"></script>
<script src="/axborot/js1/yana.js"></script>
<script>jQuery(function ($) {

hs.graphicsDir = '/axborot/css/graphics/';
hs.align = 'center';
hs.transitions = ['expand', 'crossfade'];
hs.fadeInOut = true;
hs.dimmingOpacity = 0.8;
hs.outlineType = 'rounded-white';
hs.captionEval = 'this.thumb.alt';
hs.marginBottom = 105; // make room for the thumbstrip and the controls
hs.numberPosition = 'caption';

// Add the slideshow providing the controlbar and the thumbstrip
hs.addSlideshow({
//slideshowGroup: 'group1',

```

```
interval: 5000,  
repeat: false,  
useControls: true,  
overlayOptions: {  
  className: 'text-controls',  
  position: 'bottom center',  
  relativeTo: 'viewport',  
  offsetY: -60  
},  
thumbstrip: {  
  position: 'bottom center',  
  mode: 'horizontal',  
  relativeTo: 'viewport'  
}  
});  
jQuery('#modal').modal({"show":false});  
});</script></body>  
</html>
```