

**O‘ZBEKISTON RESPUBLIKASI  
OLIY VA O‘RTA MAXSUS TA‘LIM VAZIRLIGI  
BUXORO DAVLAT UNIVERSITETI  
MAGISTRATURA BO‘LIMI**

**Qo‘lyozma huquqida  
UDK-**

**IBRAGIMOV SAMANDAR ISKANDAR O‘G‘LI**

**“KATEGORIYALASHTIRILGAN SO‘ZLARNING MA‘NOLARINI  
QO‘LLAB-QUVVATLOVCHI ELEKTRON LUG‘AT”**

**5A130202 – Amaliy matematika va axborot texnologiyalari**

**Magistr  
akademik darajasini olish uchun yozilgan  
DISSERTATSIYA**

**Ilmiy rahbar:**

**f-m.f.n.dots. T.B. Boltayev**

**Buxoro – 2018**

## **Mundarija**

Kirish	<b>4</b>
I bob. Soʻz tahlili borasida umumiy tushunchalar.	<b>5</b>
1.1. Kompyuter lingvistikasiga kirish.	<b>5</b>
1.2. Morfologik lugʻatlar.	<b>7</b>
II bob. JSP texnologiyasi.	<b>12</b>
2.1 JSP ni qoʻllab quvvatlovchi.	<b>12</b>
2.2 JSP texnologiyasi.	<b>14</b>
2.3 MySQL MBBTi va SQL tili.	<b>23</b>
2.4 JDBC.	<b>30</b>
2.5 Servletlar.	<b>35</b>
III bob. Elektron lugʻatni yaratish.	<b>39</b>
3.1 Dastur spetsifikatsiyasi	<b>39</b>
3.2 Dastur loyihasi	<b>40</b>
3.2.1 ER model	<b>41</b>
3.2.2 Dastur interfeysi	<b>44</b>
3.3 Amalga oshirish	<b>48</b>
Xulosa	<b>71</b>
Foydalanilgan adabiyotlar	<b>72</b>

## **Kirish**

Bugun kompyuter texnologiyalari dunyoda jadallik bilan rivojlanayotgan sohalardan biriga aylanib bormoqda. Kompyuterlar asta-sekin insonlar hayotiga kirib bormoqda. Ya'ni, faol ish yurituvchini internet global tarmog'isiz, intellektual ishchini esa ish joyida kompyutersiz tasavvur qilish qiyin. Kompyuter texnologiyalari inson hayot tarzida, kichik va katta korxonalarda, kompaniyalarda, o`quv dargohlarida, davlat muassasalarida behad ahamiyat kasb etmoqda.

Til ichki va tashqi manba asosida boyib boradi. Ya'ni chetdan so'z o'zlashadi yoki til ichki imkoniyati asosida – so'z yasalishi bilan yangi ma'noni bildiruvchi birlik paydo bo'ladi. Shundan so'z yasalishi – tilda bor bo'lgan material va imkoniyat asosida yangi so'z hosil qilishni, lug'at tarkibining boyishini nazarda tutadi. Ana shu so'z yasalishi bilan bog'liq lug'at tilimiz rivojini o'rganish va o'rgatishda muhim manba bo'lib xizmat qiladi.

Hozirgacha o'zbek tilida so'z yasalishi haqida talay ishlar qilinganiga qaramay, bunday lug'at umuman tuzilmagan.

## **I-bob. So'z tahlili borasida umumiy tushunchalar**

### **1.1 Kompyuter lingvistikasiga kirish.**

Kompyuter bu-sonli hisob kitoblarni yechuvchi matematik mashina sifatidagi fikr keng tarqalgan .

Aslida Kompyuterlar- til mashinalaridir.Ma'lum bir ma'noga ega bo'lgan lingvistik belgilar ustidan manipulyatsiya qilish ularning kuchli tomoni hisoblanadi.Tabiiy til informatikada markaziy o'rinni egallaydi. Bu sohadagi eng avvalgi ishlar harbiy shifrlarni avtomatik deshifratsiya qilish bilan bog'liq. 50-yillarda bir tildan boshqa tilga tarjima qilish avtomatik tarjimonlarni yaratishga katta e'tibor berildi. O'z oldiga qo'ygan asosiy maqsadga erishmagan bo'lsa ham, bu ishlar EHM bilan tabiiy tilda muloqot qilish muammosini hal qilishni jiddiy ravishda oldinga surdi. Hozirgi kunda tabiiy tillar vositasida kompyuter bilan aloqa qilish maqsadida ulkan loyihalar ishlanmoqda.

Ba'zi dasturlar muvaffaqiyat bilan ishlayapti. Ko'pchilik kompyuterlar qisman bo'lsa ham matnni qayta ishlashda ishlatiladi. Bir vaqtning o'zida kompyuterni tabiiy tilda "gapirish"ga urinishlar katta qiyinchiliklarga duch keldi. Tizimlarning eg yaxshilari ortacha bolaning lingvistik qobiliyatiga mos keladigan ayanchli ahvolni namoyish etadi.

EHM ning beshinchi avlodi va sun'iy intellekt sohasidagi ishlar tabiiy tilni tadbiq qilishda yangi bir turtki bo'ldi.Intelektli tizimlar bilimlar bilan ish ko'radi. Kompyuter tizimlarining kognitiv tomonini bilimlar hosil qiladi. Jiddiy bog'liq O'zaro bog'liqlikning quyidagi qat'iy ketma-ketligi mavjud:kompyuter-dastur-bilim-til, qaysiki, bir uchida kompyuter tursa boshqasida til.

EHM paydo bo'lgandan boshlab inson va EHM orasida semantik to'siq yuzaga keldiki, hisoblash texnikasi vositalarining rivojlanishi bilan bu to'siq mutaxasislarning hujumiga uchrayapti. Bu to'siqni pasaytirish bir tomondan, mashinaga mo'ljallangan koddan boshlab, algoritmik tillar, mantiqiy shakl almashtirish tillari orqali mantiqiy-semantik va semantik tillar tomon yo'lidan

borayapti. So'ngi ikki til sun'iy intellekt ideyalariga asoslanadi va tushunish jarayoni modellariga kiradi. Boshqa tomondan, to'siqni pasaytirish yo'lida tabiiy tillarni qayta ishlash yangi usullarini axtarish yo'lidan borayapti. Mazkur ishda masala ikkinchi yondashuv nuqtai nazaridan qaraladi. Intellectli tizimlar tayyorlovchilarning oldida turgan asosiy maqsadlardan biri, tabiiy tilning konseptual tashkil etuvchisini kompyuterga oldindan berilgan maqsad yo'lida qayta ishlash maqsadida kiritishdir.

Tilni odatdagidek filologik o'rganish intellectli tizim tayyorlovchilari uchun kam narsa taklif qildi. Teskari ta'sir esa sermahsul bo'ldi, oldin lingvistlarning e'tiboridan chetda qolib kelgan va tilni o'rganishning yangi yondashuvlarini talab qiladigan tilning aspektlarini ochdi. Oxirgi yillarda nazariy lingvistik modellarni EHMda amalgam oshirish bo'yicha kompyuterli lingvistika deb nomlangan yangi ilmiy yo'nalish paydo bo'ldi. Kompyuterli lingvistikaning odatdagi tabiiy tillarni qayta ishlash metodlaridan farqi shundaki, odatdagi metodlarda barcha lingvistika o'rganadigan narsalar butunligicha modelashtilishiga asosiy diqqat qaratiladi, kompyuterli lingvistikada esa asosiy diqqat tilni tushunish jarayonining qismlarga bo'lishga va berilgan modellarning nazariy lingvistik korrektiligi va adekvatligiga qaratiladi.

Kompyuterli lingvistika sun'iy iontellectning markaziy muammosi – bilimning kompyuterli tasviri bilan uzviy bog'liq.

Kompyuterli lingvistikaning asosiy masalasi – mantiqiy – lingvistik modellarni qurish va mos algoritim va dsturlarni yaratish.

## 1.2 Morfologik lug'atlar.

Lug'at- bu, alfavit tartibida joylashgan so'zlar majmuasi bo'lib,so'zning izohi yoki boshqa tilga tarjimasidan iborat.Lug'atlar turlicha vazifada bo'ladi.Morfologik (lingvistik) lug'at-bu shunday lugatki, so'zning ma'nosi va qo'llanilishi izohi va uning gramatik kategoriyasini taqdim qiladi.

Gramatik kategoriya (gr.Kategoria-mulohaza qilish, ta'riflash) –bu barcha bir turdagi gramatik ma'nolar:padejlar kategoriyasida padejlar birikmasi ma'nosi,zamon kategoriyasida alohida zamon formalari birikmasi ma'nosi va boshqalar.

Gramatik kategoriya gramatik mazmunga tegishli, ya'ni gramatik mazmunning qismi.

Xususan so'zning yaxlit leksik-gramatik razryadi (sinfi) ,umumiy semantik birlik va morfologik-sintaktik belgilar, -fe'l kategoriyasi va ravish kategoriyasidan iborat.

Matn-ma'no-matn (qisqacha MMM) nazariyasi quyidagi uchta postulatga tayanadi.

Postulat 1.Tabiiy tildagi cheksiz ma'nolar to'plami va cheksiz matnlar to'plami orasida bir qiymatli moslik bor.

Postulat 1 ni quyidagi quyidagi ko'rinishda ifodalash mumkin:

$$\{Ma'no_i\} \Leftarrow Til \Rightarrow \{Matn_j\} | 0 < i, j < \infty$$

Ma'no nutqning ichki tomonidan tashkil topgan; bu-nutq jarayonida ma'lumotni ruhiy o'tkazish ko'rinishidir. Matn nutqning tashqi tomonidan tashkil topgan;bu- nutq jarayonida eshituvchilar o'rtasidagi ma'lumotni yetkazish,akustik tebranishlar to'plami fizik ko'rinishidir.

Postular 1 uchun quyidagi uchta asosiy izohni beramiz.

Birinchidan , ma’no atamasini bu yerda qisqacha izohlash kerak.

Ma’no invariant til perifrazalaridir. Ya’ni tilda so’zning shu so’z mazmunini ifodalovchi boshqa so’zlarda ifodalanishidir.

Bizning oldimizga matnni qayta ishlash va tarjima qilishni avtomatlashtirish masalasi qo’yilgan. Bu masalani yechimiga ikkita yo’ldan boriladi: deduktiv va induktiv. Semantikani formal ta’rifini tashkil etishdagi birinchi “matn-ma’no-matn” modeli, ikkinchi tarjima qilishda “matn-matn” modeli.

“Matn-ma’no –matn” prinsipida yuqori sifatli mashina tarjimasini avtomatlashtirishda til semantikasidan foydalaniladi.

Ikkinchidan, matn atamasini ham alohida izohlab o’tamiz. Bu yerda nutq hech matn bilan aloqada emas, mantiqiy uyushqoqlikda emas. Matn atamasi ixtiyoriy tilning fizik tomoni, tashqi ko’rinishini ifodalaydi. Matn o’zak, so’z formasi, so’z birikmasi, ibora, abzaslardan iborat.

Uchinchidan, ma’no va matn real borliqda muvofiq kelsa ham, lingvistika nazariyasida ularni va ularning psixik va fizik uyg’unliklarini ko’rib chiqish mumkin emas. MMM faqat uning namoyishi, formal tildagi yozuvi, lingvistikadagi tadqiq qilinganligi bilan shug’ullanadi. Ma’no taqdimoti Semantik taqdimot (SemT), matn taqdimoti fonetik taqdimot (FonT) deb ataladi. Biz kelgusida barcha barcha til taqdimotlariga o’xshagan yoki mulohaza taqdimotlari haqida gaplashamiz. MMMda birinchi darajali muhim til taqdimoti hisobga olinib, (1) ni quyidagi (2) ko’rinishida ifodalanadi.

$$\{SemT_i\} \Leftarrow Til \Rightarrow \{FonT_j\} | 0 < i, j < \infty \quad (2)$$

Postulat 1 keyinchalik manashu ko’rinishda ishlatiladi.

Postulat 2. Mavjud tilning funksional modeli mantiqiy tuzilishi (2) ga muvofiq ko'rinishda amalga oshirilgan bo'lishi lozim.

Tilning matn va ma'nolari uning bevosita so'zlashuvchilarini o'z ichiga oladi, ammo qoida bo'yicha ularning aloqalari umuman bunday emas. Tabiiy til funksional modelini quramiz. Bizda bu Model Ma'no-Matn.

!Funksional model ko'pchilik kitoblarda unga sinonim bo'lgan kibernetik model so'zi orqali ham yoziladi.

O'z tabiatiga ko'ra Model Ma'no-Matn to'la dinamik bo'ladi: u barcha matnlarni ma'noga o'tishini ta'minlaydi, agar bu ma'no ifodalangan bo'lsa (Yoki barcha ma'nolarni matnga o'tadi, agar matn ifodalangan bo'lsa). Shuning uchun Model Ma'no-Matn ikkita qism modelga bo'linadi.

1) Sistema sof til qoidalari, shu til uchun L statistic moslikda berilgan

$$\{SemT_i\} \Leftrightarrow \{FonT_j\}$$

2) Sistem dinamik (prosedurali) qoidalari, ma'no va matn orasidagi o'tish jarayoni amalgam oshirilishi beriladi

$$\{SemT_i\} \Leftrightarrow \{FonT_j\}$$

Ma'no-matn modeli kuzatadigan yoki o'zgartiradigan sistemalardan tashkil topmagan. U ko'pgina qoidalarni keltirib chiqarishi (=hisoblab chiqishi) shart emas, va faqat tilning to'g'ri jumlasini yoki til mazmunini boshqa mazmunda paydo qiladi. Ma'no-matn modeli sof ekvativ (yoki transduktiv) sistema: U SemT va FonT larda bog'liq bo'lishi lozim va (teskarisi) so'zlashuvchilarning tushinishi sezgisi bilan mos kelishi kerak. Biz allaqachon SemT va FonT orasidagi ko'p qiymatli moslikni ko'rdik.



Va bu hali hammasi emas: Fraza (3) 50 000 000 dan ortiq perifrazni o'z ichiga oladi. Bir qancha MMM nuqtai nazar bilan tabiiy til sinonim boyligi asosiy qiziqishni namoyon qiladi. Boshqa tomondan aynan o'sha matn bir qancha ma'nolari mos kelishi mumkin. Quyidagi (4) frazadagidek.

Favqulotta murakkab moslik  $\{SemT_i\} \Leftrightarrow \{FonT_j\}$  postulatga bizni olib keladi, bu moslik bir qancha oddiy komponentlarga bo'linishiga imkon beradi.

Postulat 3.  $\{SemT_i\} \Leftrightarrow \{FonT_j\}$  moslik yozish uchun quyidagi ikki darajani orasiga qo'yish lozim: Sintaktik taqdimot (SinT) va Morfologik taqdimot (MorfT).

Endi biz (2)ni quyidagi ko'rinishda yozamiz:

Semantika   sintaksis        (Morfologiya+fonologiya)

$$\{SemT_i\} \Leftrightarrow \{SinT_k\} \Leftrightarrow \{MorfT_j\} \Leftrightarrow \{FonT_n\} \quad (5)$$

Biz bu yerda taklif qilingan qo'shimcha darajalarni asoslay olmaymiz. Ularni faqat gramatik odatlar bo'yicha qatnashganligini etiborga olamiz. Chunki barcha tillarning qonuniy strukturasi ikki yoki undan ortiq mustaqil avtonom maydonni namoyon qiladi: ibora va so'z qismlari mos ravishda sintaksis va morfologiyani qamrab oladi. Keyin til taqdimoti darajalari, semantic darajadan tashqari, har ikkala darajasini ham qismlarga bo'lamiz: CHUQUR (CH-) (глубинный) ma'no bilan munosabatga kirishadi, SAYOZ (S) (поверхностный) forma bilan munosabatga kirishadi. Barcha til tasvirlari sistemasi quyidagi ko'rinishda bo'ladi:

( Semantika) (chuqur sintaksis)

(sayoz sintaksis)

(chuqur morfologiya) (sayoz

morfologiya) (Morfologiya+fonologiya)

$$\{SemT_i\} \Leftrightarrow \{ChSinT_{k1}\} \Leftrightarrow \{SSinT_{k2}\} \Leftrightarrow \{ChMorfT_{j1}\} \Leftrightarrow \{SMorfT_{j2}\} \\ \Leftrightarrow \{ChFonT_n\} \Leftrightarrow \{SFonT_n\}$$

(6)

## II-bob. JSP texnologiyasi

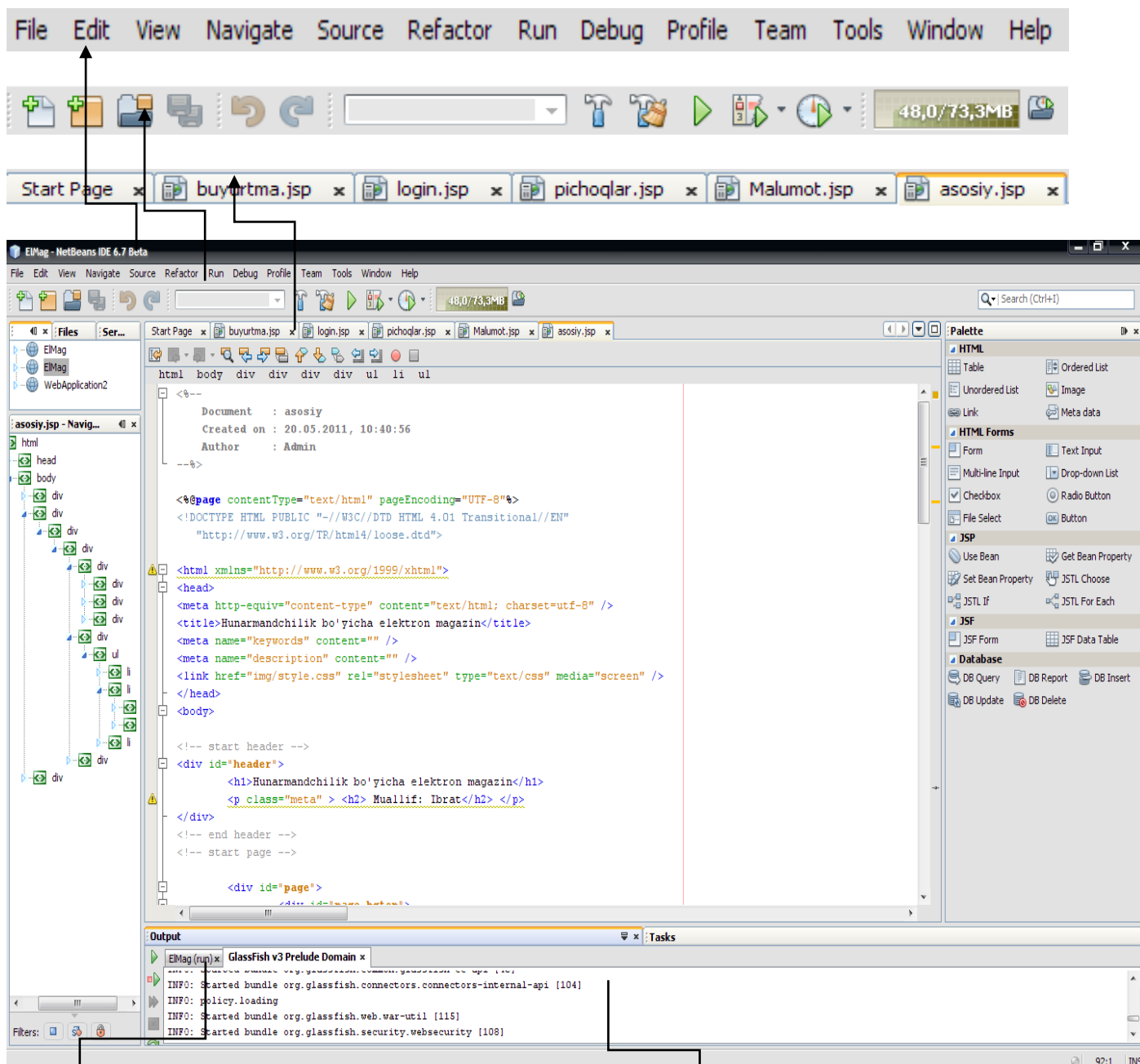
### 2.1 JSP ni qo'llab quvvatlovchi muhitlar

JSP texnologiyasi bilan ishlash uchun bir nechta muhitlardan foydalanishimiz mumkin.

Masalan, “Eclipse”, “IntellegIdea”, “Net-Beans” va hokazo.

Biz Web English ilovamizni yaratishda Net-Beans muhitidan foydalandik.

Net-Beansning bir nechta versiyalari mavjud, lekin hammasida ham **Wep Application** holati mavjud emas. **Net-Beans 6.5, Net-Beans 6.7, Net-Beans 8.0.2 Beta** larda mavjud. Bu muhitni o'rnatish jarayonida bizdan Administrator yoki Opitniy Polzavatel bo'lib kirish kerak. Yana albatta JDK (JAVA)ning biror bir muhiti o'rnatilgan bo'lishi kerak. Keyingi betda **Net-Beans 6.7 Beta** ning sayt hosil qilinayotgan holati interfeysi ko'rsatilgan.



2.1- rasm

1.Serverlar ishga tushayotgan jarayon.  
jarayon.

2.Fayl komplyatsiyaga berilgan

## **2.2 JSP texnologiyasi**

JSP texnologiyasi statik HTML matn bialn servlet tomonidan generatsiya qilinadigan kod (<% va %> ichidagi deskriptorlar)ni umumlashtirish imkoniyatini beradi.

Web dan murojaat qilinadigan JSP tayyorlash uchun JSP ni qo'llab quvvatlaydigan Web serverning JSP uchun mo'ljallangan katalogiga jsp kengaytmali fayl yaratiladi. Serverda servlet sinflari va Java kompilyatorga murojaat qilish imkoniyatli muhit mavjud bo'lishi shart.

Odatda JSP ga birinch bor murojlat qilganda JSP avtomatik ravishda serveletga shakl almashtiriladi va statik HTML kod esa bu servletning chiqish oqimiga beriladi.

HTML ifodalardan tashqari JSP hujjatiga senariy elementlari, direktivalar va amallar kiradi. Senariy elementlari servlet tarkibiga qo'yiladigan Java-kod yasash imkoniyatini, direktivalar – servlet shajarasini boshqarish imkoniyatini, amallar esa servlet hatti harakatini aniqlash imkoniatini beradilar.

Dasturchi ishini osonlashtirish uchun unga oldindan aniqlangan bir necha o'zgaruvchilar beriladi.

### **JSP afzalliklari**

#### **JSP ning ASP va ColdFusion dan afzalligi:**

Active Server Pages(ASP) dinamik fragmentlarni yasash uchun VBScript tili ishlatiladi, JSP da Java. Murakkab, masalan qayta ishlatiluvchi komponentlarni VBScriptda yasash qiyin. Undan tashqari Java-kod platformaga bog'liq emas. ASP esa faqat IIS uchun.

#### **JSP ning PHP dan afzalligi:**

PHP- HTML hujjatga joylashtiriladigan senariylar yozish tili. JSP da tarmoq bog'lanishini, MB ni va tarqalma ob'ektlarni qo'llab quvvatlovchi kuchli API

vositalari mavjud. bu imkoniyatlar PHP da murakkab amalga oshiriladi. Server va uning instrumental vositalarini ishlab chiqaruvchilar JSP ni qo'llab quvvatlaydilar.

### **JSP senariy elementlari**

JSP senariy elementlari JSP asosida generatsiya qilinadigan servletga kod fragmentini joylashtirish imkoniyatini beradi .

Senariy elementlari 3 kategoriyaga bo'linadilar:

`<%=ifoda %>` til konstruksiyasi orqali beriladigan **ifoda**. Ifoda hisoblanib, uning qiymati servletning chiqish berilganlari tarkibiga qo'shiladi.

`<% kod %>` shaklidagi **skriptlet**. Bu kod servletnins service metodi tomonidan chaqiriladigan `_jspService` metodi tarkibiga qo'shiladi.

`<%! kod %>` formatida beriladigan **deklaratsiya**. Deklaratsiyalar servlet sinfi tarkibiga, yangi metod va maydonla qo'shadi.

Ko'p hollarda JSP asosiy qismini shablon matni deb ataluvchi statik HTML matn tashkil etadi. Shablon matni servlet tomonidan o'zgarishsiz mijozga uzatiladi va uni yasash uchun  $\forall$  Web sahifa yasash vositasidan foydalanish mumkin.

Agar `<% sivilollar shablonga kiritilishi kerak bo'lsa u "<%>"` shaklida kiritilishi kerak. Agar JSP izohlari HTML hujjati tarkibiga kirishi kerak bo'lmasa u `<%-- JSP izihi --%>` shaklida bo'lishi kerak. `<!-- HTML izohi -->` izoh HTML hujjatga ga kiradi

#### **Ifodalar va oldindan aniqlangan o'zgaruvchilar**

`<%=ifoda %>` ifoda Java ifodasi bo'lib, uning qiymati hisoblanadi va hujjat tarkibiga kiritiladi. Ifodaning qiymati mijozdan so'rov qabul qilingandan keyin hisoblanadi va unda so'rovda ishtirok etgan informatsiya ishlatilishi mumkin, masalan, Current time: `<%= new java.util.Date() %>` asosida `_jspService` metodiga quyidagi kod joylashtiriladi:

```
out.println(Current time:);
```

```
out.println(new java.util.Date());
```

Ifodaning qiymatini hisoblashda oldindan aniqlangan o'zgaruvchilar ishlatilishi mumkin. Asosiy o'zgaruvchilar:

```
request-      HttpServletRequest ob'ekti
```

response-    HttpServletResponse ob'ekti  
session-     so'rov bilan bog'langan HttpSession ob'ekti  
out-          mijizga berilganlarni uzatish PrintWriter ob'ekti

misol:

Your hostname: <%= request.getRemoteHost() %>

Ifoda uchun XML sintaksisi.

Ba'zi bir serverlar ifoda uchun quyidagi XML formatidagi yozuvni ham qo'llab quvvatlaydilar:

<jsp:expression> Java ifoda </jsp:expression>

Agar oddiy ifodadan ko'ra murakkabroq hisob amalga oshirilishi kerak bo'lsa skriptlet ishlatiladi .

HTML elementidan farqli XML elementi simvollar katta-kichikligiga sezgir. Shuning uchun jsp:expression kichik harflar bilan yozilishi kerak.

JSP hujjati HTML resurslari qayerda saqlansa, shu joyda saqlanishi mumkin va biror maxsus katalog yoki CLASSPATH muhit o'zgaruvchisiga ko'rsatkich yasash kerak emas.

### Skriptletlar

Skriptlet yordamida servletning service metodi tomonidan chaqiriladigan `_jspService` metodiga  $\forall$  java-kod joylashtiriladi. Skriptlet `<% java-kod %>` formatda yoziladi. Skriptletlarga ham yuqorida ko'rsatilgan oldindan aniqlangan o'zgaruvchilar ishlatilishi mumkin. Masalan:

```
<%  
String queryData = request.getQueryString();  
out.println("Attached GET data: " + queryData);  
%>
```

Xuddi shu natijani quyidagiha ham olish mumkin:

Attached GET data: <%= request.getQueryString() %>

Ammo ko'p hollarda skriptletlar murakkab qayta ishlashni amalga oshiradiki, ifoda orqali bunday ishni amalga oshirib bo'lmaydi.

Ma'lum shartlar asosida JSP – fayl fragmentlarini joylashtirish

Skriptlet yordamida generatsiya qilinadigan hujjatga biror bir shart bajarilishi bilan bog'liq holda HTML yoki JSP standart konstruksiyalarini joylashtirish mumkin. Bu imkoniyat statik HTML kod va skriptlet kodi `_jspService` da ular JSP hujjatda qanday ketmaketlikda kelga bo'lsalar shunday ketmaketlikda joylashishlaridan kelib chiqadi. Ya'ni skriptlet hamavaqt tugatilgan ifodani tashkil etishi shart emas. Agar blok ochiq qolsa, statik HTML matn va boshqa JSP konstruksiyalari shu blokning qismini tashkil etasilar, masalan:

```
<% if (Math.random() < 0.5) { %>
```

```
    Have a <B> nice </B> day!
```

```
<% } else { %>
```

```
    Have a <B> lousy </B> day!
```

```
<% } %>
```

Bu `_jspService` da quyidagi ko'rinishga aylantiriladi>:

```
if (Math.random() < 0.5) {
```

```
    out.println("Have a <B> nice </B> day!");
```

```
} else {
```

```
    out.println("Have a <B> lousy </B> day!");
```

```
}
```

### **Page direktivasi**

Page direktivasi quyidagi atributlarni ishlatishi mumkin: `import`, `contentType`, `isThreadSafe`, `session`, `buffer`, `autoflash`, `extends`, `info`, `errorPage`, `isErrorPage` va `language`.

`import`: Servletda import qilinadigan paketlarni ko'rsatadi. Agar import qilinadigan paketlar oshkor o'rsatilmagan bo'lsa servlet quyidagi paketlarni jimlikda import qiladi: `java.lang.*`, `javax.servlet.*`, `javax.servlet.jsp.*`, `javax.servlet.http.*` va konkret server bilan bog'liq ba'zi bir paketlar.

```
<%@ page import = "package.Class" %>
```

```
<%@ page import = "package.Class1, ..., package.ClassN" %>
```



Agar paket to'liq import qilinishi kerak bo'lsa "Class" o'rniga "\*" ishlatiladi. page direktivasi hujjatda bir necha marotaba ishlatilsa bo'ladigan yagona direktiva.

page direktivasining contentType atributi

Mijozga uzatiladigan hujjat MIME tipini ko'rsatuvchi ContentType sarlavha maydoniga qiymat berish uchun ishlatiladi.

U quyidagi formatda beriladi:

```
<%@ page contentType = "MIME-type" %>
```

```
<%@ page contentType = "MIME-type; charset = Character-Set" %>
```

MIME-tip jimlikda servletlardan farqli (qaysiki, "text/plain" o'rnatiladi) JSP da "text/html" o'rnatildi.

Quyidagi msolda mijozga html formatida emas, balki excel formatidagi berilganlar uzatiladi va shuning uchun MIME-tip oshkor ko'rsatilishi kerak

page direktivasining isThreadSafe atributi

Bu atribut JSP asosida yasalgan servlet SingleThreadModel interfeysini amalga oshirishi kerak yoki yo'qligini aniqlaydi.

Atribut quyidagi formatlarda beriladi:

```
<%@ page isThreadSafe = "true" %> <%-- jimlikda --%>
```

```
<%@ page isThreadSafe = "false" %>
```

Odatda servletlarda har bir so'rov alohida oqimda bajariladi. Bunda servlet tayyorlovchisi tomonidan oqimlararo umumiy bo'lgan berilganlarni qayta ishlash uchun choralar ko'rilgan deb hisoblanadi. Bunday choralarga masalan, oqimlar tomonidan baravar murojaatda buziladigan berilganlar uchun murojaatlarni sinxronizatsiyalash kiradi.

Ba'zi hollarda (masalan murojaatlar sonini aniqlashda) bunga hojat yo'q. Ba'zi hollarda (masalan foydalanuvchi identifikatorini aniqlash) esa bu zarur.

Masalan quyidagi fragment foydalanuvchi uchun idNum ni noto'g'ri aiqlaydi. idNumni o'qish va uni yozish orasida boshqa potok ishlab, uni o'zgartirib qoyishi mumkin:

### **Page direktivasining isThreadSafe atributi**

```
<%! private int idNum=0; %>
<%
String userId = "userId " + idNum;
out.println("Your ID is " + idNum + ".");
idNum = idNum + 1;%>
```

Bu fragment korrekt bajarilishi uchun uni synchronized blokiga joylashtirish kerak:

```
synchronized(someObject){...}
```

Bu holda, agar potok blok bilan ishni boshlagan bo'lsa, toki u bu blok bilan ashni tugatmaguncha boshqa hech bir potok blok bilan ishlay olmaydi. Bu ish JSP da quyidagicha amalga oshiriladi:

page direktivasining isThreadSafe atributi

```
<%! private int idNum=0; %>
<%
synchronized(this){
    String userId = "userId " + idNum;
    out.println("Your ID is " + idNum + ".");
    idNum = idNum + 1;
}
%>
```

Agar servlet SingleThreadModel interfeysini amalga oshirsa tizim servlet ekzemplarini faqat bitta so'rov uchun yasashini kafolatlaydi. Bu uchun yoki so'rovlar novbati yoki ekzemplarlar rastasi(pool) tashkil etiladi.

```
<%@ page isThreadSafe = "false" %>
```

yordamida servlet bitta potok uchun ishlatilishi ko'rsatiladi.

page direktivasining session atributi

Hujjat HTTP seansni qo'llab quvatlashi ko'rsatiladi va quyidagi formatda beriladi:

```
<%@ page session = "true" %> <%-- jimlkda --%>
```

```
<%@ page session = "false" %>
```

True qiymat oldindan aniqlangan HttpSession tipidagi sesion o'zgaruvchisini joriy seans ob'ekti bilan (agar mavjud bo'lsa) bog'lash kerakligini ko'rsatadi. Agar ob'ekt mavjud bo'lmasa, u yasaladi va o'zgaruvchi bilan bog'lanadi. false qiymat seans ob'ekti va session o'zgaruvchisi ishlatilmasligini ko'rsatadi.

Seansni qo'llab quvvatlash hujjat bilan emas, balki foydalanuvchi bilan bog'liq.

Agar foydalanuvchi murojlat qilgan hujatdan bu imkoniyat yo'qotilsa, u foydalanuvchi murojaat qiladigan barcha hujjatlardan yo'qotilishi kerak.

Page direktivasining buffer atributi

Oldindan aniqlangan out o'zgaruvchisi bilan bog'langan ob'ekt ishlatadigan bufer o'chamini aniqlaydi. Bu odatda PrintWriter sinfining sinfosti o'lgan JspWriter sinfining ekzemplyari.

Atribut quyidagi formatda beriladi:

```
<%@ page buffer = "o'lchamkb" %> <%-- jimlkda --%>
```

```
<%@ page buffer = "none" %>
```

Servlet tomonidan ishlatiladigan buffer o'lchami ko'rsatilgan qiymatdan kam bo'lmasligi kerak,

masalan, `<%@ page buffer = "32kb" %>` hujjat buferlanishi va toki bufer (32kb) to'lmasa yoki hujjat to'la buferga kiritilmaguncha mijozga uzatilmasligini ko'rsatadi .

### **Page direktivasining boshqa atributlari**

**autoflush:** chiarish buferi to'lganda avtomatik uzatish yoki favqulodda hodisa yuz berishi.

```
<%@ page autoflash = "true" %> <%-- jimlkda --%>
```

`<%@ page autoflash = "false" %>`

Agar `buffer = "none"` ko'rsatilmagan bo'lsa `false` qiymat berish mumkin emas.

**extends:** servlet sinfustisini ko'rsatadi.

`<%@ page extends = "package.Class" %>`

**info:** servletning `getServletInfo` metodi chaqirilganda uzatiladigan satrni aniqlaydi.

`<%@ page info = "matn" %>`

**errorPage:** joriy hujjat tomonidan qayta ishlanmaydigan favqulodda hodisani qayta ishlaydigan (Throwable sinfining  $\forall$  sinfostisi orqali aniqlanadigan) JSP ni ko'rsatadi.

`<%@ page errorPage = "nisbiy URL" %>`

`page` direktivasining boshqa atributlari `isErrorPage`: joriy hujjat boshqa JSP hujjat tomonidan favqulodda hodisani qayta ishlash uchun ishlatilishi mumkin yoki mumkin emasligini aniqlaydi.

`<% @ page isErrorPage = "true" %>`

`<%@ page isErrorPage = "false" %> <%-- jimlkda --%>`

**language:** shu joydan boshlab qo'llaniladigan dasturlash tili.

`<%@ page language = "cobol" %>`

Direktivalarning XML ko'rinishi: ba'zi bir serverlar direktivalarning XML ko'rinishda ham qabul qiladi.

`<jsp:directive.direktiva_tipi atribut = "qiymat" />`

masalan:

`<%@ page import = "java.util.*" %>`

### **Quyidagi XML formatda berilishi mumkin:**

`<jsp:directive.page import = "java.util.*" />`

Apletlar va fayllarni (tashqi kodni) JSP hujjati tarkibiga qo'shish

Tashqi kodni JSP hujjati tarkibiga quyidagi vositalar orqali amalga oshiriladi:

`include` direktivasi: JSP kodni hujjat tarkibiga u servletga aylantirgungacha joylashtiradi.

jsp:include amali: JSP hujatga boshqa hujjatni qayta ishlash natijasini so'rovni qayta ishlash vaqtida qo'shadi.

jsp:plugin amali: JavaPlugIn dan foydalanuvchi appletni JSP hujjatga qo'shadi  
include direktivasi

<%@ include file = "nisbiy URL" %>

### **Fayl JSP**

Qo'shiluvchi fayl asosiy hujjatga ta'sir etuvchi JSP kod o'lishi mumkin, masalan, snippeted.jsp fayl quyidagi kodni saqlasin:

<%! int accessCount 0; %>

U holda bu faylni quyidagicha qo'shish mumkin:

<%@ include file = "snippeted.jsp" %> <%-- accessCount ni aniqlash --%>

<%= accessCount++ %> <%-- accessCountni Ishlatish --%>

Agar qo'shiladigan fayl o'zgarsa, uni ishlatadigan barcha JSP hujjatlar yangilanishi shart. JSP 1.1 da jsp\_precompile so'rov parametrini ishlatish mumkin, qaysiki JSP ni qayta servletga generatsiya qilishni buyuradi, ammo JSP 1.0 da bu imkoniyat yo'q. Shuning uchun JSP asosida servletni qayta generatsiya qilishning ishonchli yo'li JSP hujjat modifikatsiya sanasini o'zgartirish.

include uchun XML format: <jsp:directive.include file = "..."/> />

### **So'rov vaqtida fayl qo'shish**

jsp:include amali so'rov vaqtida JSP-fayl natijasini asosiy hujjatga qo'shish

imkoniyatini beradi. JSP-fayl o'zgarganda o'zgarish sanasini hujjatda o'zgartirishning keragi yo'q, lekin so'rov vaqtida servlet allaqachon generatsiya qilingan va uni o'zgartirish imkoniyati bo'lmaydi.

Qo'shiladigan fayl JSP-kod bo'lishiga qaramasdan, fayl tarkibini tashkil qilishda ishlatiladigan resurslar JSP hujjat bo'lishi mumkin. Qo'shiladigan resurs URLini server odatdagidek interpretatsiya qiladi va u servlet yoki JSP-hujjatga ko'rsatishi mumkin. Servlet tomonidan fayni qo'shish uchun ishlatiladigan RequestDispatcher sinfining method metodi ana shunday ishlaydi.

<jsp:include file = "nisbiy\_URL" flush="true"/> />

## 2.3 SQL haqida

**MySQL** — bu eng mashhur va juda koʻp foydalaniladigan maʼlumotlar bazasini boshqarish tizimi(MBBT) hisoblanadi. Bu tizim juda katta maʼlumotlar bilan ishlash uchun yaratilmagan, aksincha biroz kichik hajmdagi bazalar bilan katta tezlikda ishlash uchun yaratilgan. Uning asosiy ishlash doirasi, saytlar hisoblanadi. Hozirgi kunda juda koʻp sayt va bloklarning maʼlumotlari aynan shu MBBT saqlanadi.

Xoʻsh, saytlarning nimalari bazada saqlanishi mumkin? Dastlab, maʼlumotlar saqlash uchun dasturchilar fayllardan foydalanishgan, yaʼni fayl ochilib kerakli maʼlumotlar u yerga saqlanib, kerakli paytda chaqirib ishlatilgan. Keyinchalik fayllardan voz kechilib (noqulayliklar yuzaga kelgan, yaʼni faylni ochish, oʻqish, yopish,.. koʻp vaqtni olib qoʻygan, u yerdan qidirish, xullas juda koʻp), maʼlumotlar bazasiga oʻtilgan. Bazada saytdagi maqolalar, sayt foydalanuvchilari haqidagi maʼlumotlar, sayt kontentlari, qoldirilgan kommentariyalar, savol-javoblar, hisoblagich natijalari va shunga oʻxshash juda koʻp maʼlumotlar saqlanadi. MySQL shunday maʼlumotlarni oʻzida saqlaydi.

**MySQL** — juda katta tezlikda ishlovchi va qulay hisoblanadi. Bu tizimda ishlash juda sodda va uni oʻrganish qiyinchilik tugʻdirmaydi.

MySQL tizimi tcx kompaniyasi tomonidan, maʼlumotlarni tez qayta ishlash uchun korxona miqyosida ishlatishga yaratilgan. Keyinchalik ommalashib, saytlarning asosiy bazasi sifatida yoyildi.

Soʻrovlar **SQL** tili orqali amalga oshiriladi. Bu MBBT **relyatsion maʼlumotlar baza** hisoblanadi. Bu degani baza jadvallar, jadvallar esa ustunlardan tashkil topgandir.

MySQL MBBTi 2 xil turdagi litsenziyaga ega. Birinchisi tekin, yaʼni MySQLni koʻchirib olish va ishlatish hech qanday harajat talab qilmaydi va **GPL(GNU Public Licenseb, GNU)** litsenziyasiga asoslanadi. Ikkinchi turi, GPL shartiga koʻra, agar siz MySQL kodlarini biror dasturingizda ishlatsangiz, bu

dasturingiz ham GPL(tekin) bo`lishi kerak. Bu esa dasturchiga to`g`ri kelmaydi. Shuning uchun, bu dasturingizni pullik qilishingiz uchun MySQL pullik litsenziyasini sotib olishingiz kerak.

MySQL logotipi delfin hisoblanadi. Bu delfinni ismi «**Sakila**»dir. Bu logotipni OpenSource tuzuvchilaridan biri **Ambrose Twebaze** ga tegishlidir.

MySQL juda ko`p operatsion tizimlar bilan ishlay oladi. Bularni yozadigan bo`lsam: **AIX, BSDi, FreeBSD, HP-UX, Linux, Mac OS X, NetBSD, OpenBSD, OS/2 Warp, SGI IRIX, Solaris, SunOS, UnixWare, Windows 95, Windows 98, Windows NT, Windows 2000, Windows XP, Windows Server 2000, Windows Vista, Windows 7,...**

MySQL shved korxonasi **MySQL AB** ga tegishli bo`lgan, keyinchalik **SUN** firmasi MySQL ni o`ziniki qilib olgan. Bir necha yil oldin esa, gigant korxonalardan biri **Oracle**, SUN firmasini sotib oldi va hozirda MySQL Oracle firmasi mahsuloti sifatida chiqib kelmoqda. Bu bir hisobda Microsoft SQL Server ga katta raqobatdosh ekanligini anglatadi.

MySQL bir necha serverlarning bir qismi hisoblanadi. Misol uchun, **WAMP, AppServ, LAMP, Denwer,...** Kliyentlar MySQL serveriga ma`lum bir kutubxonalar orqali ulanadi. MySQL ga quyidagi dasturlash tillari ulanib ishlashi mumkin: **Delphi, C, C++, Java, Perl, Php, Python, Ruby** va boshqalar.

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

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

SQL (Structured Query Language, odatda «sikvel» deyiladi), ma`nosi — Tarkiblangan so'rovlar tili. Bu relyatsion Ma`lumotlar bazalarida ishlashga imkon

beradigan tildir. Bu til ifodalarining xususiyati shundan iboratki, ular ma'lumotlarni qayta ishlash protseduralariga emas, natijalariga yo'naltirilgandir.

SQL o'zi ma'lumotlar qayerda joylashgani, indekslar qandayligini va, hatto, amallarni qanday izchillikda qollasa bo'lishini aniqlaydi; bu tafsilotlarni ma'lumotlar bazasiga so'rovlarda ko'rsatish kerak emas. SQL tili IBM kompaniyasida MBBT DB2 yaratish jarayonida ishlab chiqilgan va keng ko'lamda RISC protsessorli mashinalarda UNIX tizimlar asosida hamda meynfreymlarda, super kompyuterlar asosida qurilgan katta hisoblash tizimlarida qo'llanilgan. Shu bilan birga, u mustaqil bo'lmasdan, PL/SQL va Transact-SQL kabi ichki dasturlash tillariga inkapsulyatsiya qilinadi. 1986-yilda, ANSI (American National Standart Institute) SQL tilining rasmiy standartini ishlab chiqdi, 1992- yilda bu standart kengaytirildi. Butun til 30 ga yaqin operatorlarga ega bolib, ba'zi versiyalarida sal ko'proq, ba'zilarida sal kamroqdir. Har qanday MB har xil obyektlarga, ya'ni jadvallar, protseduralar, funksiyalar, tasavurlar, ketma-ketliklar va hokazolarga ega.

<<Kliyent-Server>> texnologiyasiga ko'ra, foydalanuvchi EHM (Kliyent) lar so'rovlari maxsus Ma'lumotlar serverlarida (Server) qayta ishlanadi, foydalanuvchi EHM larga faqat so'rovni qayta ishlash natijalari qaytariladi. Tabiiyki, Server bilan muloqot qilish uchun yagona til kerak va bunday til sifatida SQL tanlangan.

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

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

-DCL (Ma'lumotlarni Boshqarish Tili) foydalanuvchiga ma'lum obyektlar ustida malum ta'sir o'tkazishga ruxsat berish yoki bermaslikni aniqlovchi



vositalardan iborat. SQL Standarti ANSI tomonidan aniqlangan va hozir ICO (Standartlashtirish bo'yicha xalqaro tashkilot) tomonidan qabul qilingan. Lekin kommersial ma'lumotlar bazalari dasturlari ANSI ni ogohlantirmasdan SQL ni kengaytiradilar, ya'ni o'zlari foydali deb hisoblagan har xil xossalarni qo'shadilar.

SQL so'rov tili — muhim talablar asosida berilganlar bazasiga murojaat qilib, undan so'rov talablariga javob beradigan natijalarni olish tili. Hozirgi paytda turli berilganlar bazasi va ularning sistemalari yaratilgan. Ammo relyatsion tipdagi barcha berilganlar bazasiga murojaat qilib, ulardan tegishli ma'lumotlarni olish SQL tili ishlab chiqilgan. Bu tilda so'rovlar qanday vositalar yordamida tashkil qilinishiga qisqacha to'xtalib o'tamiz.

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

Jadvallarni tuzishda ularda 2 ta butunlay bir xil satr bo'lmashligi kerak. Shuni e'tiborga olish kerakki, jadvaldagi hamma ustunlar ham bir ma'noda har bir satrni aniqlayvermaydi. Bu ular kam funksional nagruzkaga egaligini, ular har bir satrni bir ma'noda aniqlash uchun yaroqli emasligini anglatadi. Bundan tashqari, ulardan bu jadval nima haqida ekanligini aniqlash qiyin.

Har bir satrni bir ma'noda aniqlab bera oladigan ustunlar yoki ustunlar guruhi kalitli deb ataladi.

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

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

Sonli o`zgaruvchilarni tasvirlashda int kalitli so`z qo`llaniladi.

1.0E —100 dan 1.0E+100 gacha bo`lgan doirada 22 raqamga ega bo`la oladigan sonlarni tasvirlaydi. Sana va vaqtni tasvirlashda Date standarti qo`llaniladi.

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

— sanani ifodalash uchun Yevropa standartidan yoki Amerika standartidan foydalaniladi.

Endi 1-jadvalni yaratish uchun quyidagilarni yozishimiz mumkin:

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

Biz Name ustunini har bir atribut 15 belgiga ega bo`lgan satrli o`zgaruvchi uzunlik bilan aniqladik. Amaliyotda atribut uzunligi keragidan ko`ra ortiqroq beriladi. Bu shu narsa bilan bog`liqki, biz oldindan satrli o`zgaruvchining qaysi ma`nolari jadvalga kiritilishini bilmaymiz. Shuning uchun ma`lum bir zaxira bo`lgani ma`qul.

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

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

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

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

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

**Select buyrug`i** – SQL tilining asosiy buyruqlaridan biri hisoblanadi. Bu buyruq barcha amallarni ma'lumotlar qatori bilan ta'minlaydi. Quyida select buyrug`ining asosiy imkoniyatlarini ko`rib chiqamiz.

*a ) Barcha ma'lumotlarni ko'rish .*

Select \* from factory;

(\*) belgisi factory jadvalining barcha ustunlarini tanlash kerakligini bildiradi.

Bu natijani Select buyrug`idan keyin barcha ustunlar nomini berish bilan olsa bo`ladi:

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

Natijada ekranda 1-jadvalning hamma ustunlari paydo bo'ladi. Ustunlar tartibi Select buyrug`ida belgilangandek bo`ladi:

*b ) ustunlarni tanlab chaqirish.*

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

Select Name, Dept from factory;

Natijada bir Name va Dept dagi jadvallarni olamiz:

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

Select — ustunlar nomi;

from — jadvallar nomi;

where — satr tanlash sharti.

Misol: 1-jadvaldan barcha xodimlar, ishchilarni tanlash uchun quyidagini kiritish kerak:

Select name, dept

from factory

Where Dept='dmik'; buyrug`ini berish lozim.

Natijada ekranda quyidagilar hosil bo`ladi:

*e ) Solishtirish operatorlarining qo`llanilishi.*

Avvalgi misolda satr tanlash shartini tasvirlashda <<tenglik>> belgisi qo`llangan edi.

<<Tenglik>> belgisidan tashqari yana solishtirish shartini tanlashda qo`llanilishi mumkin bo`lgan 7 ta eng oddiy solishtirish operatorlari mavjud. Bular:

!= — teng emas;

> — katta;

< > — teng emas;

! > — berilgandan katta emas;

< — berilgandan kichik;

! < — berilgandan kichik emas;

>= — katta yoki teng;

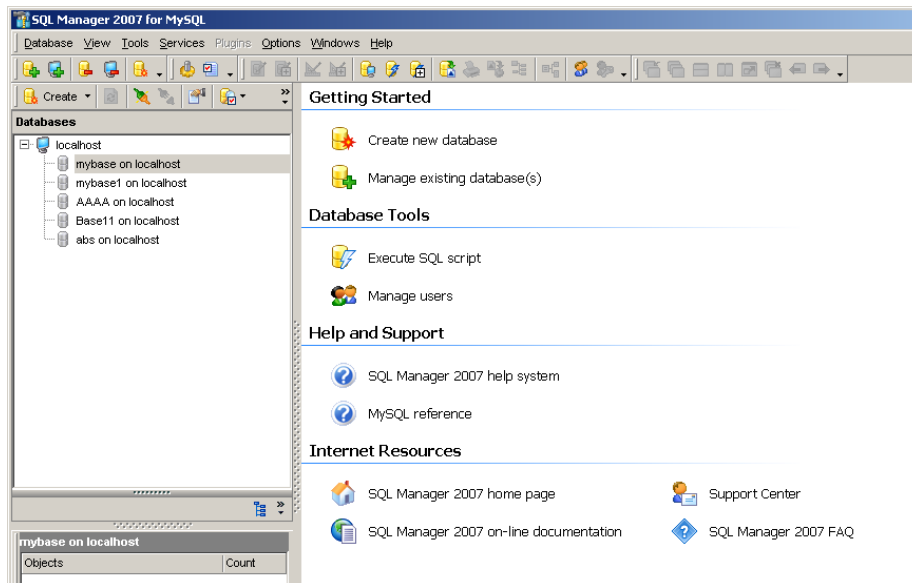
<= — kichik yoki teng.

## 2.4 JDBC

Ushbu bo'limda Java dasturlash tilida MySQL, SQL tillarida yaratilgan ma'lumotlar bazini bo'g'lashni ko'rib chiqamiz. Buning uchun birinchi MySQL da ma'lumotlar bazasi qanday qilib yaratilishini ko'rib chiqamiz.

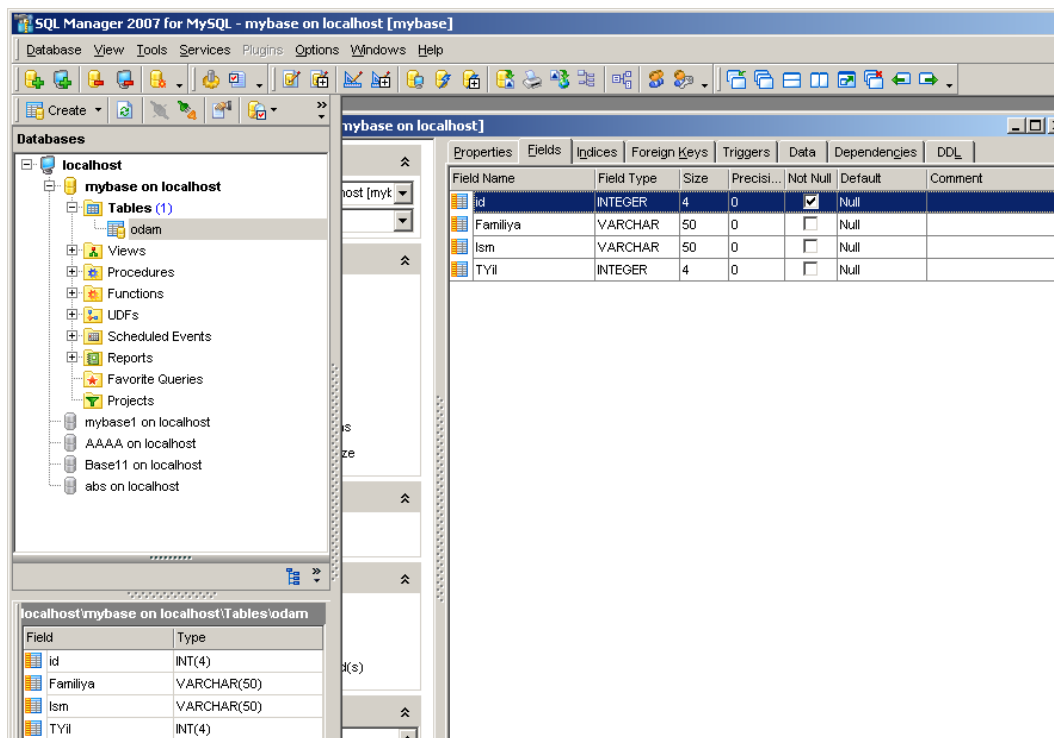
SQL Manager for MySQL dasturini ishga tushuring.

Dastur quyidagi ko'rinishda namoyon bo'ladi.

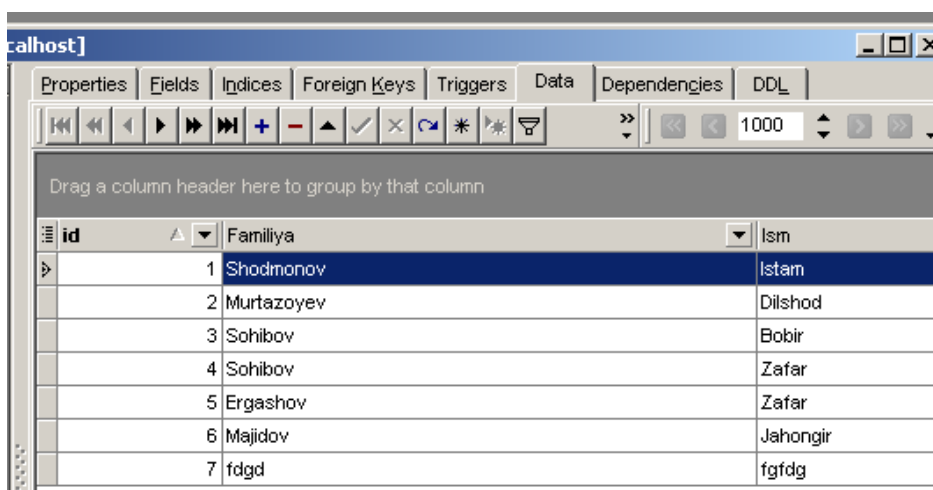


Bu yerda Create new database – orqali yangi ma'lumotlar bazasi yaratiladi.

Ma'lumotlar bazasi yaratilgandan keyin jadvallar yaratiladi.



Masalan Odam degan jadvalni yaratamiz.



Xuddi shi tartibda xoxlagancha ma'lumotlar bazasini va jadvallarni yaratish mumkin.

Endi MySQLda yaratilgan ma'lumotlar bazasini Java dasturlash tilida bog'lashni qarab chiqamiz.

import java.sql.\*; - paketni import qilamiz.

String JDBC\_DRIVER = "com.mysql.jdbc.Driver";

Drayver ko'rsatiladi.

```
String DB_URL = "jdbc:mysql://localhost/mybase";
```

Baza yo'li ko'rsatiladi.

```
String USER = "root";
```

Foydalanuvchi nomi ko'rsatiladi.

```
String PASS = "";
```

Foydalanuvchi paroli ko'rsatiladi.

Bazani bog'lash

```
Connection conn = null;
```

Drayverni bog'lash

```
Class.forName("com.mysql.jdbc.Driver");
```

Bazani bog'lash

```
conn = DriverManager.getConnection(DB_URL,USER,PASS);
```

```
stmt = conn.createStatement();
```

```
String sql;
```

```
sql = "SELECT id, Familiya, ism, tyil FROM Odam";
```

So'rov asosida bazadan ma'lumotlarni olish

```
ResultSet rs = stmt.executeQuery(sql);
```

Malumotlarni o'qish va ekranga chiqalish

```
while(rs.next()){
```

```
    int id = rs.getInt("id");
```

```
    String first = rs.getString("familiya");
```

```
    String first2 = rs.getString("ism");
```

```

String first3 = rs.getString("tyil");

System.out.print(id+"    ");

System.out.print(first+"    ");

System.out.print(first2+"    ");

System.out.print(first3+"    ");

System.out.println("");

}

```

Misol. Mybase ma'lumotlar bazasidagi Odam jadvalidan ma'lumotlarni consol rejimda chiqarish.

Baza ko'rinishi.

id	Familiya	Ism
1	Shodmonov	Istam
2	Murtazoyev	Dilshod
3	Sohibov	Bobir
4	Sohibov	Zafar
5	Ergashov	Zafar
6	Mallidov	Ishnondir

Dasturi.

```

import java.sql.*;

public class FirstExample {

    static final String JDBC_DRIVER = "com.mysql.jdbc.Driver";

    static final String DB_URL = "jdbc:mysql://localhost/mybase";

    static final String USER = "root"; static final String PASS = "";

    public static void main(String[] args) {

        Connection conn = null;

        Statement stmt = null; try{

```



```

Class.forName("com.mysql.jdbc.Driver");

conn = DriverManager.getConnection(DB_URL,USER,PASS);

stmt = conn.createStatement();

String sql;

sql = "SELECT id, Familiya, ism, tyil FROM Odam";

ResultSet rs = stmt.executeQuery(sql);

System.out.print("kod:   Familiyasi:   Ismi:   Tyili\n") ;

while(rs.next()){

    int id = rs.getInt("id");

    String first = rs.getString("familiya");

    String first2 = rs.getString("ism");

    String first3 = rs.getString("tyil");

    System.out.print(id+"   ");

    System.out.print(first+"   ");

    System.out.print(first2+"   ");

    System.out.print(first3+"   ");

    System.out.println("");    }

rs.close();    stmt.close();    conn.close();

} catch(SQLException se){    se.printStackTrace();

} catch(Exception e){    e.printStackTrace(); }

System.out.println("Goodbye!");} }

```

```

D:\Xayatov_X\java_Istam\JDBC1>FirstExample.bat
D:\Xayatov_X\java_Istam\JDBC1>javac FirstExample.java
D:\Xayatov_X\java_Istam\JDBC1>java -cp .;mysql-connector-java-5.0.3-bin.jar FirstExample
1 Familiyasi:      Ismi:      Tyili
2 1 Shodmonov      Istam      1989
3 2 Murtazoyev     Dilshod   1999
4 3 Sohibov        Bobir     1999
5 4 Sohibov        Zafar     1986
6 5 Ergashov       Zafar     1990
7 6 Majidov        Jahongir  1988
8 Goodbye!
D:\Xayatov_X\java_Istam\JDBC1>

```

Dastur natijasi.

## 2.5 Servlet

### Servlet haqida umumiy tushuncha

1. Servlet CGI(Common Gateway Interface) Umumiy shlyuz interfeysiga alternative vosita.
2. Servletlar Web serverda bajariladilar va oraliq server vazifasini bajaradilar
3. Servletlar Web brauzerlardan va boshqa HTTP mijozlardan so'rov qabul qiladilar va ularni bajaradilar.

### Servlet bajaradigan ishlar

- Foydalanuvchi uzatgan berilganlarni o'qish
- HTTP-so'rov bilan uzatilgan boshqa informatsiyani o'qish
- Natijani generatsiya qilish
- Qayta ishlash natijalarini formatlash
- HTTP-javob parametrlarini o'rnatish
- Hujjatni mijozga uzatish

### doGet va doPost parametrlari

Har ikkala metod HttpServletRequest va HttpServletResponse sinflariga tegishli ikkita parametrga ega.

HttpServletRequest ob'ekti metodlari yordamida so'rov haqidagi turli ma'lumotlar (sarlovha maydonlarining tarkibi, forma orqali uzatilgan berilganlar, mijoz tuguni adresi va h.k.) olinadi.

HttpServletResponse ob'ekti metodlari orqali mijoz so'roviga javob qaytariladi.

Javobda holat kodi (200, 404, ...), sarlovha maydolari (Content-Type, Set-Cookie,

...). Bundan tashqari mijozga hujjat uzatuvchi PrintWriter ob'ektini yasash mumkin.

Oddiy servletlarda mijozga javob quyidagicha tashkil etiladi:

...

```
PrintWriter out = response.getWriter();
```

...

```
Try{
```

```
...
```

```
out.println("ixtiyoriy matn");
```

```
...
```

```
}[catch ...]
```

```
finally {
```

```
...
```

```
out.close();
```

```
...
```

```
}
```

...

Servlet yasash vaqtida init metodi chaqiriladi, qaysiki servleti initsializatsiya qiladi va sozlaydi. Shundan keyin mijozning har bir murojaatiga nisbatan yangi oqim yasaladi va unda servletning service metodi chaqiriladi. Agar bir vaqtning o'zida bir necha mijoz murojaat qilga, mos ravishda bir necha oqim yasaladi. Agar servlet SungleServletModel interfeysini amalga oshirsa, u holda faqat bitta oqim ishlaydi. Service metodi ichida doGet, doPost va boshqa doXXX metodlari chaqiriladi.

Agar server servlet ishini tugatishni xohlasa u servletning destroy metodini chaqiradi.

### **HTML kod generatsiya qiluvchi servlet.**

Servlet HTML kod generatsiya qilishi uchun quyidagi ishlarni ajaradi:

Brouzerga uzatiladigan ma'lumot HTML kodi formatida tashkil etilganligini xabar beradi

Println metodi uchun mo'ljallangan satrlarda HTML deskriptorlari va mos qiymatlarni yasaydi.

1-ishni bajarish uchun servlet HTTP javobda Content-Type sarlovha maydonini yasaydi (response.setContentType("text/html");).

HTTP-javob yasash tartibi

HTTP-javob holat satri, bir yoki bir necha sarlovha maydonlari, 1-ta bo'sh satr va javob tarkibi(hujjat)dan iborat.

Sarlovha maydonlari javob hujjatdan oldin o'rnatilishi kerak. Sarlovha maydonlari o'rnatilish tartibining farqi yo'q. Holat satri sarlovha maydonlaridan keyin ham o'rnatilishi mumkin.

Hujjatni buferlashtirib uzatish mumkin. Bufer o'lchamini response.getBufferSize() orqali o'qish va response.setBufferSiza metodi orqali o'rnatish umkin.

### **Servlet hayot sikli**

Servlet yasash vaqtida init metodi chaqiriladi, qaysiki servleti initsializatsiya qiladi va sozlaydi. Shundan keyin mijozning har bir murojaatiga nisbatan yangi oqim yasaladi va unda servletning service metodi chaqiriladi. Agar bir vaqtning o'zida bir necha mijoz murojaat qilga, mos ravishda bir necha oqim yasaladi. Agar servlet SungleServletModel interfeysini amalga oshirsa, u holda faqat bitta oqim ishlaydi. Service metodi ichida doGet, doPost va boshqa doXXX metodlari chaqiriladi.

Agar server servlet ishini tugatishni xohlasa u servletning destroy metodini chaqiradi.

### **Servlet va JSP tayyorlash uchun hujjatlar :**

<http://java.sun.com/products/jsp/download.html>

<http://java.sun.com/products/servlet/2.2/javadoc>

<http://java.sun.com/j2ee/j2sdkee/techdocs/api>

Fayllar odatda Web serverning “lib” katalogida joylashdilar. Lekin har xil Web serverlarda bu katalog har xil bo'lishi mumkin: Servlet sinflari servlet.jar da, JSP sinflari esa jsp.jar, jspengine.jar va jasper.jar fayllarida. CLASSPATH da bu kataloglarga ko'rsatkich yasash ishni qulaylashtiradi.

HTTP protokolidagi so'rovlar.

**GET** so'rovi juda ko'p uchraydigan http so'rov. Brauzer quyidagi hollarda bu so'rov bilan serverga murojaat qiladi:

Foydalanuvchi adres satrida URL kiritsa;

Web sahifada foydalanuvchi gipermatnli murojaatni aktivlashtirsa;

<FORM> deskriptorida **METHOD** atributining GET qiymati ko'rsatilgan forma foydalanuvchi tomonidan aktivlashtirilganda.

<FORM> deskriptorida METHOD=“POST” ko'rsatilgan forma foydalanuvchi tomonidan aktivlashtirilganda servlet POST so'rovini qayta ishlaydi.

### **Xulosa**

Hozirda web ilova yaratadigan texnologiyalar juda ko'p. Masalan JSP, PHP, ASP, ASP.NET va hokazo.

Biz yuqorida JSP texnologiyasi bo'yicha web ilova yaratish uchun kerak bo'lgan barcha texnologiya, instrument va tillar bilan tanishib chiqdik. Bular orqali xohlagan web ilovani amalga oshirish mumkin. Buning uchun ularning imkoniyatlari bilan yaqindan tanishib chiqish kerak.

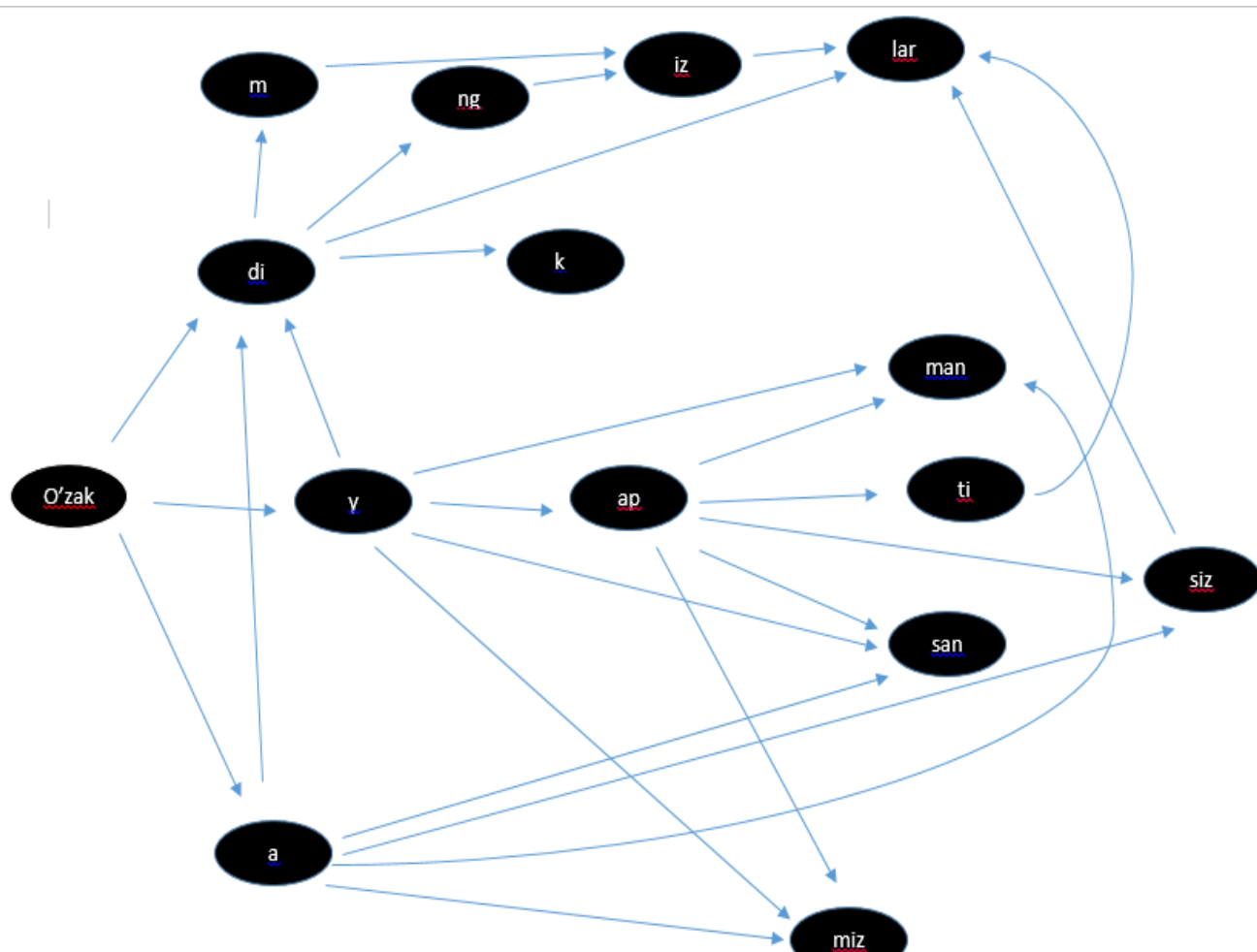
### **III-bob. Elektron lug'atni yaratish**

#### **3.1 Dastur spetsifikatsiyasi**

- Nomi – Kategoriyalashtirilgan so'zlarning ma'nolarini qo'llab-quvvatlovchi elektron lug'at.
- Qisqacha tavsif – So'zni leksik va semantik jihatdan tahlil qilib beruvchi web ilova.
- Maqsad – So'zlarni leksik va semantik jihatdan tahlil qilish va o'zbek tilida so'z tuzilishi haqida tushunchaga ega bo'lish.
- **Asosiy ssenariy:**
  1. Foydalanuvchilar ro'yxatdan o'tadi.
  2. Tizim lug'atlarini ko'rish imkoniyati bo'ladi.
  3. O'zi lug'atni boyitishi mumkin.
  4. So'zni leksik va semantik jihatdan tahlil qiladi.

### 3.2 Dastur loyihasi

Ushbu dasturni yaratishda Web Arxitekturadan foydalanildi. Har bir foydalanuvchi xohlagan joydan turib mobil telefoni yoki kompyuter orqali saytdan foydalanishi mumkin. Bunda dastur ikkita soda sahifadan tuzilgan. Sahifalarning birida foydalanuvchilar fe'l so'z turkumidagi so'zlarning leksik-semantik tahlilini ko'rishlari mumkin. Buning uchun matn kiritiladigan darchaga so'zni kiritish kifoya. Ikkinchi sahifada esa foydalanuvchilar ushbu dastur bazasini to'ldirishda bevosita ishtirok etishlari mumkin. Quyida fe'l so'z turkumining leksik tahlili grafik holda keltiriladi.



### 3.2.1 ER modeli

Berilganlarni modelini yaratishda wordsemantic (data base) ning quyidagi jadvallari hosil qilindi:

#### Asosiy -jadval. wordsemantic asosiy jadvallari.

Jadval nomi	Izoh
ozak	Fe'lining o'zagi jadvali
qushimcha	Fe'l qo'shimchalari jadvali

wordsemantic har bir asosiy jadvallarining tasnifi

1. **ozak** – Fe'l so'z turkumidagi so'zlarning o'zagi saqlanadi:

#### 1-jadval. Ozak jadvali.

Ustun	Tip	Null	Jimlikda	Izoh
<i>id</i>	int(11)	HeT		Kalit
verb	varchar(45)	HeT	NULL	Logini
semantic	varchar(45)	HeT	NULL	Parole

2. **qushimcha** – fe'l so'z turkumida ishlatiladigan qo'shimchalar jadvali:

#### 2-jadval. words jadvali.

Ustun	Tip	Null	Jimlikda	Izoh
<i>id</i>	int(11)	HeT		Kalit
addition	varchar(150)	HeT	NULL	Qo'shimchalar
comment	varchar(150)	HeT	NULL	So'zlarga izoh



ozak
id INT(11)
verb VARCHAR(100)
semantic VARCHAR(200)
Indexes

qushimcha
id INT(11)
addition VARCHAR(100)
comment VARCHAR(100)
Indexes

“wordsemantic”(database)ning asosiy jadvallari yaratilishi va ular orasidagi bog`lanishlarni sql tilidagi kodi quydagicha bo`ladi:

--ro`yxatdan o`tgan foydalanuvchilar jadvali

```
CREATE TABLE users (
    id int(11) NOT NULL AUTO_INCREMENT,
    UserName varchar(45) DEFAULT NULL,
    Password varchar(45) DEFAULT NULL,
    Name varchar(45) DEFAULT NULL,
    SurName varchar(45) DEFAULT NULL,
    Email varchar(45) DEFAULT NULL,
    Address varchar(45) DEFAULT NULL,
    learn int(11) DEFAULT 0,
    PRIMARY KEY (id) );
```

--fe'lining o'zak jadvali

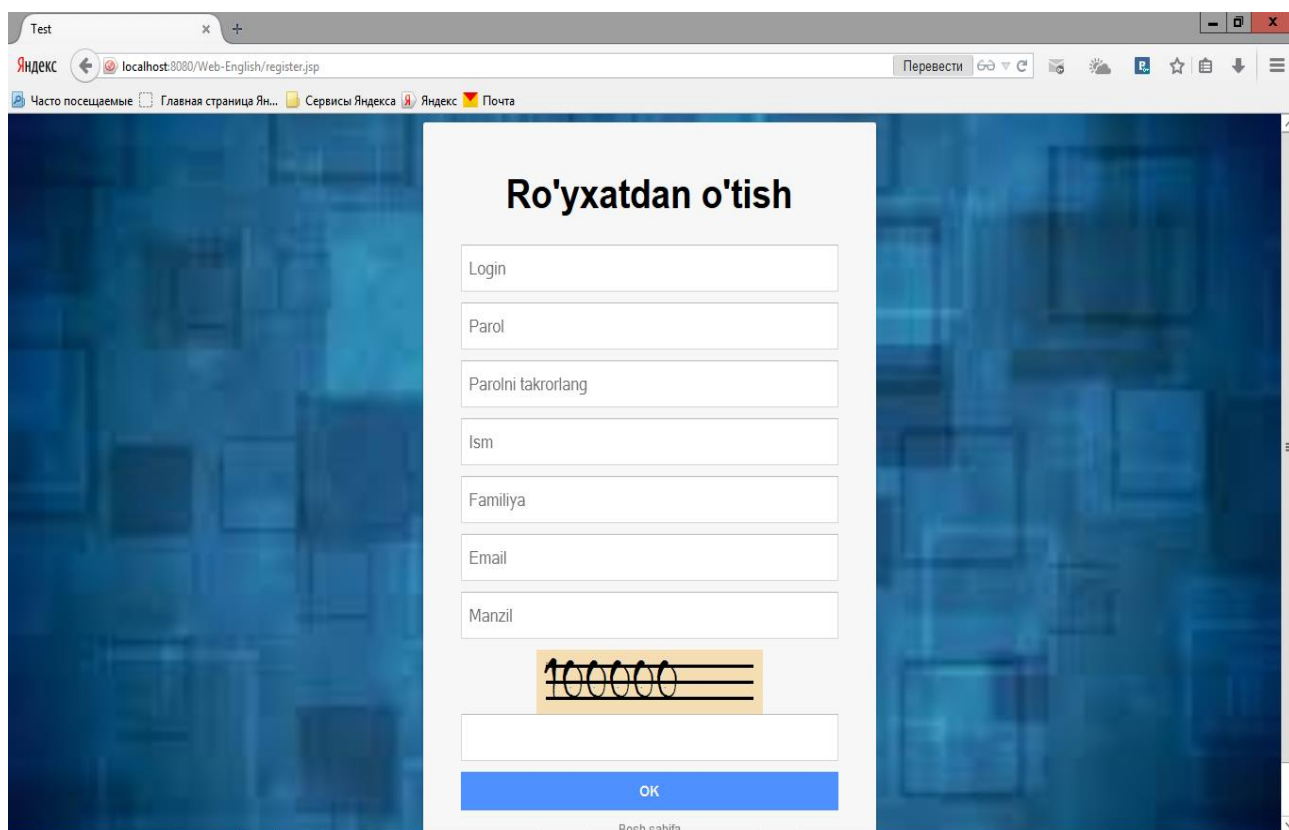
```
CREATE TABLE ozak (  
    id int(11) NOT NULL AUTO_INCREMENT,  
    verb varchar(150) DEFAULT NULL,  
    semantic varchar(150) DEFAULT NULL,  
    PRIMARY KEY (id) );
```

--fe'lining qo'shimchalar jadvali

```
CREATE TABLE qushimcha (  
    id int(11) NOT NULL AUTO_INCREMENT,  
    addition varchar(150) DEFAULT NULL,  
    comment varchar(150) DEFAULT NULL,  
    PRIMARY KEY (id) );
```

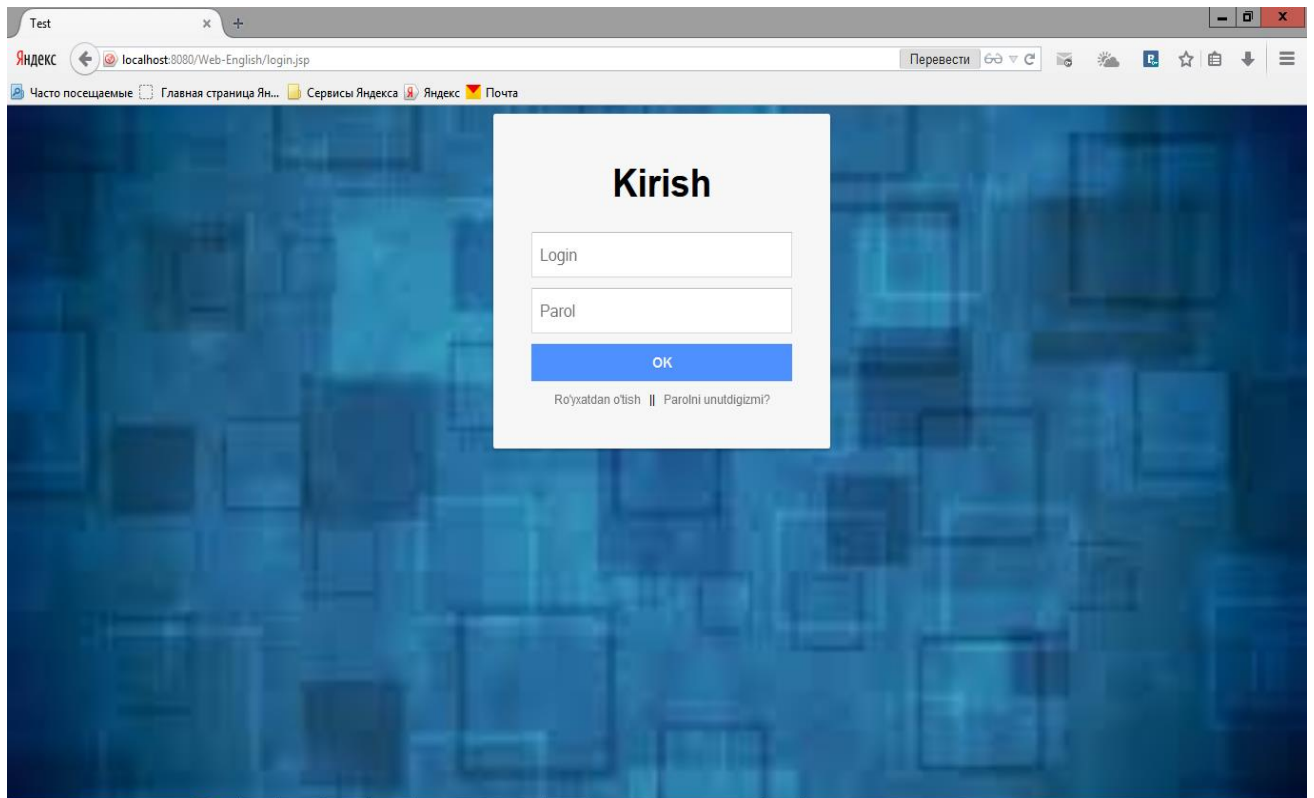
### 3.2.2 Dastur interfeysi

Quyida dasturning interfeysi bilan tanishib chiqamiz. 1-rasmda foydalanuvchilar ro'yxatdan o'tadigan oyna keltirilgan. Foydalanuvchilar keltirilgan darchalarni to'g'ri va to'liq to'ldirib tizimdan ro'yxatdan o'tishlari mumkin.



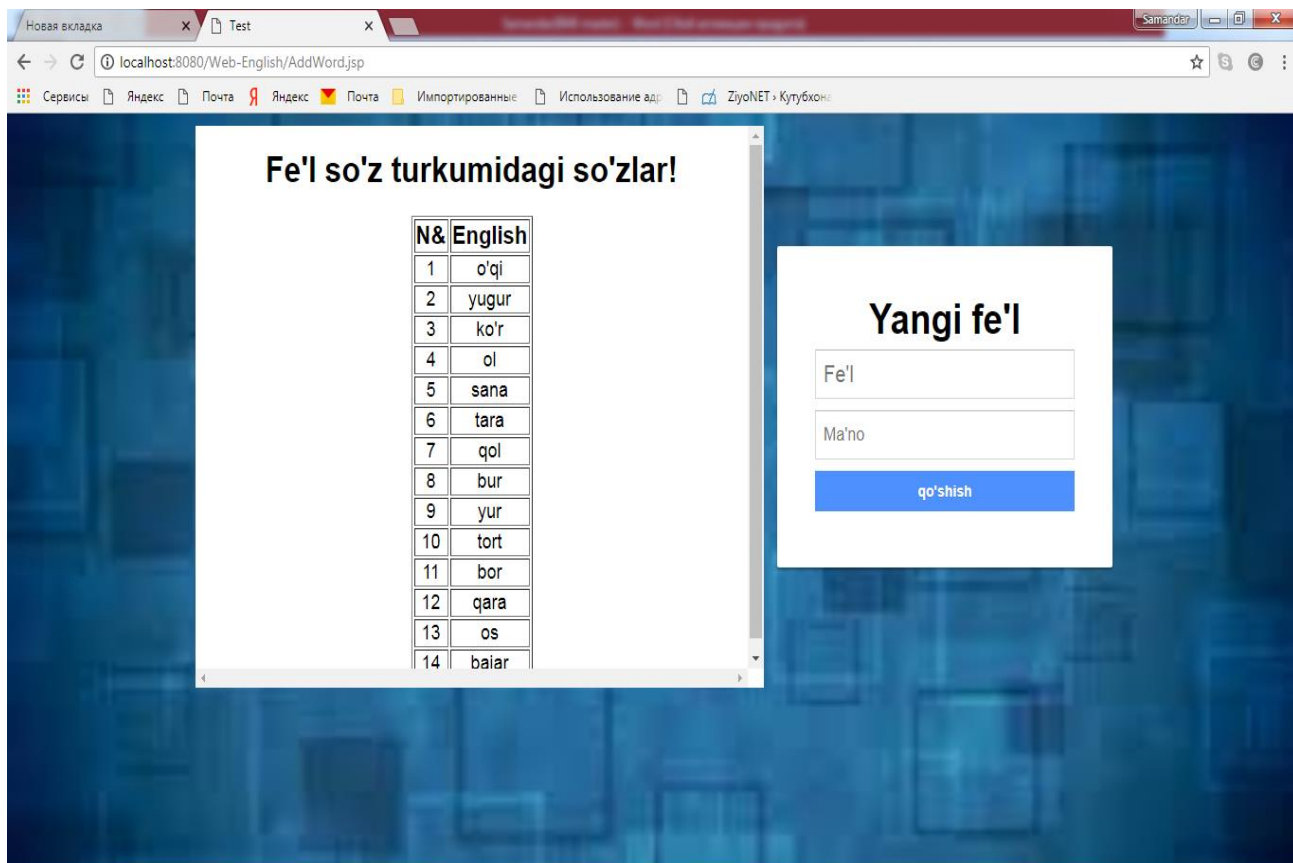
**1-rasm Ro'yxatdan o'tish sahifasi.**

2-rasmda esa tizimga kirish sahifasi ko'rsatilgan. Login va parol to'g'ri kiritilsa, foydalanuvchi tizimga kira oladi, aks holda yana shu kirish sahifasi yuklanadi.



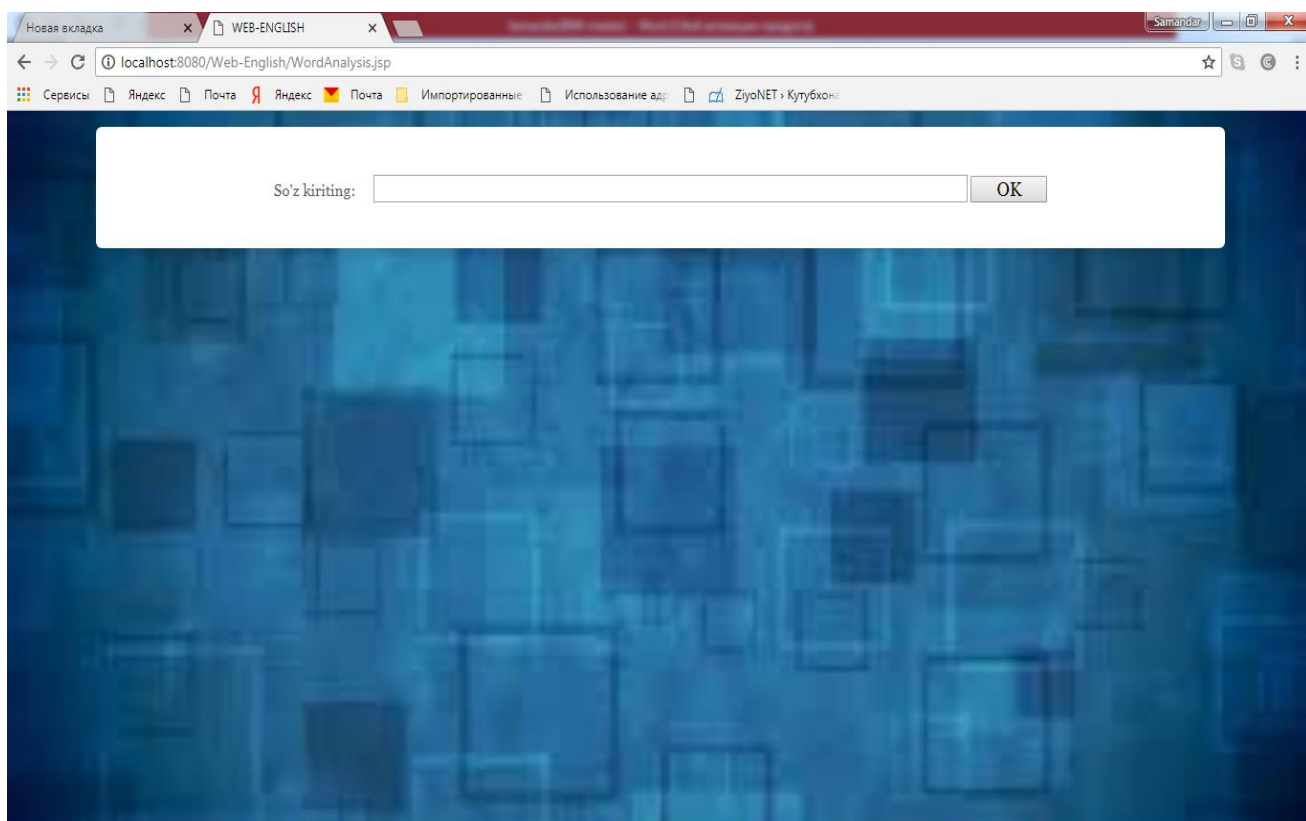
**2-rasm Tizimga kirish sahifasi.**

3-rasmda tizimga yangi so'z kiritish va bazada mavjud fe'llarning ro'yxatini ko'rish imkoniyati mavjud.

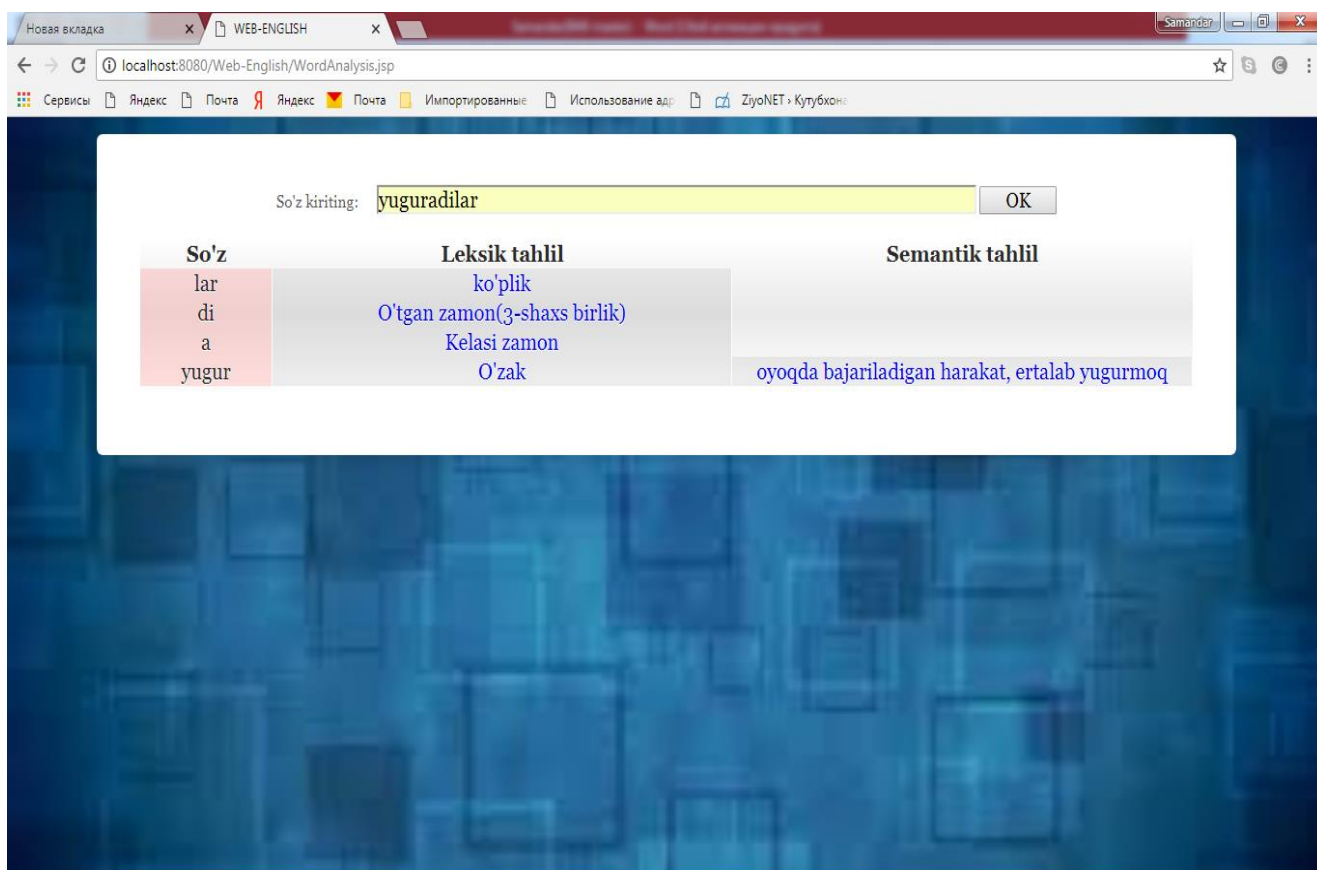


### 3-rasm So'z qo'shish sahifasi.

4-rasm va 5-rasmlar asosiy sahifa bo'lib, bunda so'zning leksik va semantik tahlilini ko'rish mumkin. Bunda foydalanuvchi yuqoridagi darchaga fe'l so'z turkumidagi so'zni kiritadi va OK tugmasini bosib pastda hosil bo'ladigan jadvalda natijani ko'rishi mumkin. Jadvalda fe'l turkumidagi so'z dastlab leksik jihatdan tahlil qilinadi va uning o'zak va qo'shimchalari alohida qatorlarda ajratilib izohlari bilan keltiriladi. Keyingi ustunda esa uning ma'nosi keltiriladi.



**4-rasm Asosiy sahifa.**



**5-rasm Asosiy sahifa.**

### 3.3 Amalga oshirish

```
<% @page contentType="text/html" pageEncoding="UTF-8"%>
<% @page import="java.sql.ResultSet"%>
<% @page import="java.sql.PreparedStatement"%>
<% @page import="java.sql.DriverManager"%>
<% @page import="java.sql.Connection"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta name="description" content="" />
<meta name="keywords" content="" />
<title>WEB-ENGLISH</title>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<link rel="stylesheet" type="text/css" href="style.css" />
<link rel="stylesheet" type="text/css" href="jadval.css" />
<script type="text/javascript" src="lib/js/jquery-1.6.2.js"></script>
<script type="text/javascript" src="lib/jquery-1.7.1.min.js"></script>
<script type="text/javascript" src="lib/jquery.dropotron-1.0.js"></script>
<script type="text/javascript" src="lib/jquery.slidertron-1.1.js"></script>
<script type="text/javascript">
    $(function() {
        $('#menu > ul').dropotron({
            mode: 'fade',
            globalOffsetY: 11,
            offsetY: -15
        });
    });
</script>
<script type="text/javascript">
```

```

$(document).ready(function(){
    $('#btnok').click(function(){
        var word = $('#word').val();
        $.ajax({
            type: 'POST',
            data: {word: word},
            url: 'AjaxAnalysis',
            success: function (result) {
                $('#result1').html(result);
            }
        });
    });
});
</script>
</head>
<body>
<div id="wrapper">
    <div id="page">
        <center>
            <form method="post">
                <label>So'z kiriting: </label>
                &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<input style="font-size: 20px;"
id="word" type="text" size="60px"></input>
                <input type="button" style="font-size: 18px; width: 80px"
value="OK" id="btnok"></input>
                <span id="result1"></span>
            </form>
        </center>
        <br/>
    </div>

```



```

    </div>
</body>
</html>

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package controller;

import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.Statement;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import static model.DictionaryManagedBean.conn;
import graph.AdjacencyMapGraph;
import graph.Edge;
import graph.Vertex;

```

```

@WebServlet(name = "AjaxAnalysis", urlPatterns = {"/AjaxAnalysis"})
public class AjaxAnalysis extends HttpServlet {
    public static void cyclicRoad(AdjacencyMapGraph g, Vertex<String>
c,Vertex<String> fc,String s) {
        Iterable<Edge<Integer>> rds = g.outgoingEdges(c);
        for(Edge<Integer> r: rds){
            Vertex<String> ov = g.opposite(c,r);
            if(ov==fc)
                System.out.println(s);
            else cyclicRoad(g,c,fc,s);
        } }

    protected void processRequest(HttpServletRequest request,
HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        try (PrintWriter out = response.getWriter()) {
            out.println("<!DOCTYPE html>");
            out.println("<html>");
            out.println("<head>");
            out.println("<title>Servlet AjaxController</title>");
            out.println("</head>");
            out.println("<body>");
            out.println("<h1>Servlet AjaxController at " + request.getContextPath()
+ "</h1>");
            out.println("</body>");
            out.println("</html>");
        }
    }
}
@Override

```

```

protected void doGet(HttpServletRequest request, HttpServletResponse
response)
    throws ServletException, IOException {
    processRequest(request, response);
}
@Override
protected void doPost(HttpServletRequest request, HttpServletResponse
response)
    throws ServletException, IOException {
    response.setContentType("text/plain");
    String word = request.getParameter("word");
    String part="",adds="",base=word;
    boolean check=false;
    PrintWriter out = response.getWriter();
    try{
        out.print("<table  style='font-size: 20px; width: 100%;word-wrap: break-
word;\n" +
" margin: 0 auto;\n" +
" border-collapse: separate;\n" +
" border-spacing: 0;\n" +
" border: 0;\n" +
" text-shadow: 0 1px 0 #fff;\n" +
" color: #2a2a2a;\n" +
" background: #fafafa;\n" +
" background-image: -moz-linear-gradient(top, #fff, #eaeaea, #fff); /* Firefox
3.6 */\n" +
" background-image: -webkit-gradient(linear,center bottom,center
top,from(#fff),color-stop(0.5, #eaeaea),to(#fff));\n" +
" margin-top: 20px;\n" +
" margin-bottom: 20px;>");
    }

```

```

out.print("<th align='center'>");
out.print("<tr align='center'>");
out.print("<td align='center'>");
out.print("<strong>So'z</strong>");
out.print("</td>");
out.print("<td align='center'>");
out.print("<strong>Leksik tahlil</strong>");
out.print("</td>");
out.print("<td align='center'>");
out.print("<strong>Semantik tahlil</strong>");
out.print("</td>");
out.print("</tr>");
out.print("</th>");

```

```

Class.forName("com.mysql.jdbc.Driver");

```

```

Connection

```

```

con=DriverManager.getConnection("jdbc:mysql://localhost:3306/wordsemantic
","root","root");

```

```

Statement stmt = con.createStatement();

```

```

ResultSet rss;

```

```

rss=stmt.executeQuery("SELECT verb,semantic FROM ozak where verb
like '"+base+"'");

```

```

while(rss.next())

```

```

{

```

```

out.print("<tr>");

```

```

out.print("<td align='center' style='width: 60px;\n" +

```

```

" background: #efefef;\n" +

```

```

" background: rgba(300,144,144,0.3);\n" +

```

```

" border-right: 1px solid white; '>");

```

```

        out.print(rss.getString(1));
        out.print("</td>");
        out.print("<td align='center' style='color: blue;width: 350px;\n" +
" background: #efefef;\n" +
" background: rgba(144,144,144,0.15);\n" +
" border-right: 1px solid white; '>");
        out.print("O'zak");
        out.print("</td>");

        out.print("<td align='center' style='color: blue;width: 350px;\n" +
" background: #efefef;\n" +
" background: rgba(144,144,144,0.15);\n" +
" border-right: 1px solid white; '>");
        out.print(rss.getString("semantic"));
        out.print("</td>");

        out.print("</tr>");

    }

    do
    {
        rss = stmt.executeQuery("SELECT verb FROM ozak where verb like
""+base+"%");
        if(rss.next())check=true; else {
            base=base.substring(0, base.length()-1);}
    }
    while(!check);

```

```

adds=word.substring(base.length());

for(;true;){
    check=false; part="";
    do
    {
        rss = stmt.executeQuery("SELECT addition FROM qushimcha where
addition like '"+adds+"'");
        if(rss.next())check=true; else {
            part+=adds.substring(0, 1);
            adds=adds.substring(1);
        }
    }
    while(!check);
    rss = stmt.executeQuery("SELECT addition,comment FROM qushimcha where
addition like '"+adds+"'");
    while(rss.next())
    {

        out.print("<tr>");
        out.print("<td align='center' style='width: 100px;\n" +
" background: #efefef;\n" +
" background: rgba(300,144,144,0.3);\n" +
" border-right: 1px solid white; '>");
        out.print(rss.getString(1));
        out.print("</td>");

        out.print("<td align='center' style='color: blue;width: 350px;\n" +
" background: #efefef;\n" +
" background: rgba(144,144,144,0.15);\n" +

```

```

" border-right: 1px solid white; '>");
    out.print(rss.getString(2));
    out.print("</td>");

out.print("</tr>");

    }
    if(part.isEmpty()){break;} else
    {adds=part; continue;}
    }

    rss = stmt.executeQuery("SELECT verb, semantic FROM ozak where verb like
    '"+base+"%");
    while(rss.next())
    {

        //1-qator so`z tarjimasi
        out.print("<tr>");
        out.print("<td align='center' style='width: 60px;\n" +
        " background: #efefef;\n" +
        " background: rgba(300,144,144,0.3);\n" +
        " border-right: 1px solid white; '>");
        out.print(rss.getString(1));
        out.print("</td>");
        out.print("<td align='center' style='color: blue;width: 350px;\n" +
        " background: #efefef;\n" +
        " background: rgba(144,144,144,0.15);\n" +
        " border-right: 1px solid white; '>");
        out.print("O'zak");
        out.print("</td>");

```

```

        out.print("<td align='center' style='color: blue;width: 350px;\n" +
        " background: #efefef;\n" +
        " background: rgba(144,144,144,0.15);\n" +
        " border-right: 1px solid white; '>");
        out.print(rss.getString("semantic"));
        out.print("</td>");

    out.print("</tr>");

}

//out.print("</div>");
}
catch(Exception ex)
{

}

out.print("</table>");
}

@Override
public String getServletInfo() {
    return "Short description";
}

}

<% @page contentType="text/html" pageEncoding="UTF-8"%>
<% @page import="java.sql.*"%>

```



```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title>Test</title>
<meta name="keywords" content="" />
<meta name="description" content="" />
  <link rel='stylesheet' href='http://codepen.io/assets/libs/fullpage/jquery-
ui.css'/>
    <link rel="stylesheet" href="style2.css" media="screen" type="text/css"
/>
</head>
<body>
  <div id="wrapper">
    <div id="page">

      <%

                                try{

                                Class.forName("com.mysql.jdbc.Driver");

                                Connection con

                                =DriverManager.getConnection("jdbc:mysql://localhost:3306/wordsemantic
?" + "user=root&password=root");

                                PreparedStatement ps=

                                con.prepareStatement("select id,verb from ozak");

                                ResultSet

                                rs=ps.executeQuery();%>
          <center>      <table>
                        <tr> <td>

```

```

<div id="" style="background-color: white; overflow: scroll; width: 600px;
height: 500px;" >
    <center> <h1>Fe'l so'z turkumidagi so'zlar!</h1> </center>
<center>
<table border="1px" style="background-color: white;">
<thead style="font-size: 22px;">
    <tr align="center">
        <th>N&</th>
        <th>English</th>
    </tr>
</thead>
    <tbody style="font-size: 18px;">
<% int nmr=0; while(rs.next()){ %>
<tr align="center">
<td><%= ++nmr%></td>
<td><%= rs.getString("verb")%></td>
</tr>
<% } %>
</tbody>
</table>

                                </center>

                                <% }
                                catch(Exception
e){e.printStackTrace();
                                }
                                %>
</div></td>
    <td>

```

```

        <div class="login-card">
                                                    <h1 style="font-
weight: bold">Yangi fe'l</h1>
        <form method="post" style="border: #2C2B2B; "
action="AddVerbProccess.jsp" name="form1">
            <input style="font-size: 20px;" type="text" name="verb"
placeholder="Fe'l"/>
            <input type="text" name="semantic" placeholder="Ma'no"/>
            <input type="submit" name="login" class="login login-submit"
value="qo'shish"/>
        </form>
    </div>

    <script
src='http://codepen.io/assets/libs/fullpage/jquery_and_jqueryui.js'></script>

        </td> </tr>
    </table></center>

</div></div>
</body>
</html>

```

```
package model;
```

```
import java.sql.*;
```

```
public class Database {
    public static Connection con=null;
```

```
    public static Connection getConnection() {
```

```

        try {
            Class.forName("com.mysql.jdbc.Driver");
            con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/wordsemantic",
"root","root");
            return con;
        } catch (Exception ex) {
            System.out.println("Database.getConnection() Error -->" +
ex.getMessage());
            return null;
        }
    }

    public static void close(Connection con) {
        try {
            con.close();
        } catch (Exception ex) {
        }
    }
}

```

```

<% @page import="java.util.Random"%>
<% @page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title>Test</title>
<meta name="keywords" content="" />

```

```

<meta name="description" content="" />

<link rel='stylesheet' href='http://codepen.io/assets/libs/fullpage/jquery-
ui.css'/>

<link rel="stylesheet" href="style2.css" media="screen" type="text/css"
/>

<script type="text/javascript">
    document.ondradstart=noselect;
    document.onselectstart=noselect;
    document.oncontextmenu=noselect;
    function noselect(){return false;}
    function Xato()
    {
var empt = document.forms["form1"]["username"].value;
if (empt === "")
{
    alert("Login kiritilmagan!");
return false;
}
var empt = document.forms["form1"]["userpass"].value;
if (empt === "")
{
    alert("Parol kiritilmagan!");
return false;
}
var empt = document.forms["form1"]["userpass2"].value;
if (empt === "")
{
    alert("Takroriy parol kiritilmagan!");
return false;
}

```

```

var empt = document.forms["form1"]["name"].value;
if (empt === "")
{
    alert("Ism kiritilmagan!");
    return false;
}
var empt = document.forms["form1"]["surname"].value;
if (empt === "")
{
    alert("Familiya kiritilmagan!");
    return false;
}
var empt = document.forms["form1"]["email"].value;
if (empt === "")
{
    alert("Email kiritilmagan!");
    return false;
}
var empt = document.forms["form1"]["address"].value;
if (empt === "")
{
    alert("Manzil kiritilmagan!");
    return false;
}
}

function req()
{
    var email = document.forms["form1"]["email"].value;
    var validemail = /^[a-zA-Z0-9._-]+@[a-zA-Z0-9.-]+.[a-zA-Z]{2,4}$/;
    if(!(validemail.test(email))) {

```

```

alert("Pochta noto'g'ri kiritilgan!");
form1.email.focus;
return false;
}}
</script>

<%
StringBuffer sb=new StringBuffer();
for(int i=1;i<=5;i++)
{
    sb.append((char)(int)(Math.random()*79+23+7));
}
String cap=new String(sb);
%>
</head>
<body>
    <%session.invalidate(); %>

    <%if(request.getAttribute("regerr")!=null){
                                out.print("<font
style='color:red'>"+request.getAttribute("regerr")+"</font>");
    } %>

    <center>                <div class="login-card"
style="width: 400px">
        <h1 style="font-weight: bold">Ro'yxatdan o'tish</h1><br />
        <form name="form1" action="registerprocess.jsp">
            <input type="text" name="username" placeholder="Login" />
            <input type="password" name="userpass" placeholder="Parol"/>

```

```

        <input type="password" name="userpass2" placeholder="Parolni
takrorlang"/>
        <input type="text" name="name" placeholder="Ism"/>
        <input type="text" name="surname" placeholder="Familiya"/>
        <input type="text" name="email" placeholder="Email"
onchange="req()"/>
        <input type="text" name="address" placeholder="Manzil"/>
        <div style="background-color: aqua"><h2><s><i><font
face="casteller"><%=cap%></font></i></s></h2></div>
        <input type="text" name="cap1" value="" />
        <input type="hidden" name="cap2" value='<%=cap%>'
readonly="readonly"/>
        <input type="submit" name="register" class="login login-submit"
value="OK" onclick="validation()"/>
    </form>

    <div class="login-help">
        <a href="index.jsp" style="color: #2C2B2B">Bosh sahifa</a>
    </div>

</div> </center>

<script
src='http://codepen.io/assets/libs/fullpage/jquery_and_jqueryui.js'></script>
</body>
</html>

```

```

<% @ page language="java" import="java.util.*" pageEncoding="ISO-
8859-1"%>
<% @page import="java.sql.*"%>
<%try{
int maxid=0, maxidresult=0;
String username=request.getParameter("username");

```



```

String userpass=request.getParameter("userpass");
String userpass2 = request.getParameter("userpass2");
String name=request.getParameter("name");
String surname=request.getParameter("surname");
String email=request.getParameter("email");
String address=request.getParameter("address");
String c1 = request.getParameter("cap1");
String c2 = request.getParameter("cap2");
if(username.isEmpty()||userpass.isEmpty()||userpass2.isEmpty()||name.isEm
pty()||surname.isEmpty()||email.isEmpty()||address.isEmpty())
{
request.setAttribute("regerr","<div style='color: yellow'
align='center'>Ma'lumotlarni to'liq kiriting</div>");
%>
<jsp:forward page="register.jsp"></jsp:forward>
<%
}
if(c1.isEmpty())
{
request.setAttribute("regerr","<div style='color: yellow; font-size: 24px'
align='center'>Xavfsizlik kodini kiriting</div>");
%>
<jsp:forward page="register.jsp"></jsp:forward>
<%
}
if(c1.equals(c2)==false)
{
request.setAttribute("regerr","<div style='color: yellow; font-size: 24px'
align='center'>Xavfsizlik kodi noto'g'ri kiritilgan!</div>");

```

```

%>
<jsp:forward page="register.jsp"></jsp:forward>
<%
}
if(!userpass.equals(userpass2))
{
request.setAttribute("regerr","<div style='color: yellow; font-size: 24px'
align='center'>Parol va takroriy parol bir xil emas!</div>");
%>
<jsp:forward page="register.jsp"></jsp:forward>
<%
}
Class.forName("com.mysql.jdbc.Driver");
Connection
con=DriverManager.getConnection("jdbc:mysql://localhost:3306/lugat?" +
"user=root&password=root");
PreparedStatement ps=con.prepareStatement("select UserName,Email from
users where UserName = '"+username+"' or Email='"+email+"'");
//ps.setString(1,n);
ResultSet rs=ps.executeQuery();
if(rs.next()){
if(username.equals(rs.getString(1))|| email.equals(rs.getString(2))){
request.setAttribute("regerr","<div style='color: yellow; font-size: 24px'
align='center'>Kechirasiz! Login yoki Email ro'yxatdan o'tgan!</div>");
%>

<jsp:forward page="register.jsp"></jsp:forward>
<%
}

```

```

else{

    if(email.equals(rs.getString(2))){
        request.setAttribute("regerr","<div style='color: yellow; font-size: 24px'
        align='center'>Kechirasiz! Login yoki Email ro'yxatdan o'tgan!</div>");
        %>

        <jsp:forward page="register.jsp"></jsp:forward>
        <%
        }

        }
        }
    else{
        Statement stmt = con.createStatement();
        ResultSet rss = stmt.executeQuery("SELECT max(id) FROM results");
        if (rss.next()){
            maxidresult=rss.getInt(1);
        }

        maxidresult++;
        ResultSet rss2 = stmt.executeQuery("SELECT max(id) FROM users");
        if (rss2.next()){
            maxid=rss2.getInt(1);
        }

        maxid++;

        PreparedStatement ps2=con.prepareStatement("INSERT INTO results
        VALUES (?, ?, ?, ?)");

```

```

ps2.setInt(1,maxidresult);
ps2.setString(2,username);
ps2.setInt(3,0);
ps2.setInt(4,0);
int s2=ps2.executeUpdate();
PreparedStatement ps1=con.prepareStatement("INSERT INTO users
VALUES (?, ?, ?, ?, ?, ?, ?, ?)");
ps1.setInt(1,maxid);
ps1.setString(2,username);
ps1.setString(3,userpass);
ps1.setString(4,name);
ps1.setString(5,surname);
ps1.setString(6,email);
ps1.setString(7,address);
ps1.setInt(8,0);
int s=ps1.executeUpdate();
if(s>0&& s2>0){
    request.setAttribute("reg","<div style='color: yellow; font-size: 24px'
align='center'>Siz ro'yxatdan o'tdingiz, marhamat kirishingiz
mumkin</div>");
    %>
    <jsp:forward page="login.jsp"></jsp:forward>
    <%

}
else{
    request.setAttribute("regerr","<div style='color: yellow; font-size: 24px'
align='center'>Kechirasiz! Ba'zi xatoliklar bor. Ma'lumotlarni aniq
kiriting!</div>");
    %>

```

```

<jsp:forward page="register.jsp"></jsp:forward>
<%
}
}
con.close();
}catch(SQLException e2){
e2.printStackTrace();
request.setAttribute("regerr","<div style='color: yellow; font-size: 24px'
align='center'>Kechirasiz! Ba'zi xatoliklar bor. Ma'lumotlarni aniq
kiriting!</div>");
%>

<jsp:forward page="register.jsp"></jsp:forward>
<%

}

%>

```

## **Xotima**

Bitiruv malakaviy ishini bajarish davomida quydagi natijalarga erishildi:

1. HTML dasturlash tili va uning imkoniyatlari tahlil qilindi.
2. Dinamik saytlarni yaratish texnologiyasining afzalliklari va kamchiliklari ko‘rib chiqildi.
3. Web texnologiyalari imkoniyatlari oshirish vositalaridan foydalanish masalalari ko‘rib chiqildi.
4. Ma‘lumotlar bazasini boshqarish tizimlari taxlil qilindi
5. Instrumental vosita sifatida dinamik saytlar yaratish imkonini beruvchi JSP web-dasturlash tili va My SQL ma‘lumotlar bazasini boshqarish tizimi tanlandi.
6. Foydalanuvchilar JSP dasturlash tili haqida o‘zbekcha ma‘lumot olishlari mumkin bo‘ldi.
7. MySQL ma'lumotlar bazasi bilan JSP dasturlash tili o‘rtasida aloqa o‘rnatildi.
8. O‘rganuvchilar fe‘lning leksik-semantik tahlilini biladigan tizim tashkil qilindi.

### **Foydalanilgan manbalar va adabiyotlar ro`yxati**

1. Karimov I.A. “Yuksak ma’naviyat yengilmas kuch” – Toshkent “Ma’naviyat” nashriyoti. 2008 –yil. 176 bet.
2. Sayfiyev J.F. “SQL tiliga kirish” uslubiy qo’llanma. Buxoro 2005 yil. 76 bet.
3. Хомоненко А. Д., Цыганков В. М., Мальцев М.Г. Базы данных: Учебник для высших учебных заведений / Под ред. Проф. А.Д. Хомоненко. -6-е изд. Доп.–СПб.: КОРОНА –Век, 2009. -736 bet.
4. Ицик Бен –Ган Microsoft SQL Server 2008. Основы T-SQL: Пер. с англ.– СПб.: БХВ –Петербург 2009. -432 bet.
5. Zakirova T.A., Ibragimov E.U., Masharipov A.K., Musaeva M.A. “Web dasturlash” . O’quv qo’llanma – Toshkent 2010 yil.184 bet.
6. Филиппов С.А. “Основы современного веб-программирования” Учебное пособие.М.: НИЯУ МИФИ. 2011 160-с.
7. Лабберс, Питер, Олберс, Брайан, Салим, Френк. HTML5 для профессионалов: мощные инструменты для разработки современных веб – приложений.: Пер. с англ. – М. : ООО «И.Д. Вильямс», 2011. – 272 с.: ил. – Парал. тит. англ.
8. Мальчук Е.В. HTML и CSS. Самоучитель – М.: Издательский дом «Вильямс», 2008. – 416 с.: ил.
9. Мейер Э. CSS – каскадные таблицы стилей. Подробное руководство, 3 – е издание. – Пер. с англ. ~ СПб: Символ-Плюс, 2008. – 576 с., ил.
10. Мержевич В.В. HTML и CSS на примерах. – СПб.: БХВ-Петербург, 2005. – 448 с.: ил.
11. Муссиано Ч., Кеннеди Б. HTML и XHTML. Подробное руководство,6 е издание. – Пер. с англ. – СПб: Символ Плюс, 2008. – 752 с., ил
12. Гудман, Дэнни, Моррисон, Майкл. JavaScript. Библия пользователя, 5-е издание.: Пер с англ. – М.: ООО «И.Д. Вильямс», 2006. – 1184 с.: ил. – Парал. тит . англ.

13. Глушаков С.В., Жакин И. А., Хачиров Т. С. Программирование Web страниц – Харьков: Фолио, 2005. – 390 с. – (Учебный курс).
14. Java Server Pages. Библиотека профессионала. : Пер, с англ. - М. : Издательский дом "Вильямс", 2002. - 448 с. : ил. - Парал. тит. англ.
15. JavaServer Pages, 3rd Edition 3rd Edition by Hans Bergsten.
16. Formal Language Theory for NLP; Department of Computer Science  
University of Haifa 31905 Haifa, Israel

### **Internet manzillari**

17. <http://lessons.itportal.uz/>
18. <http://www.tutorialspoint.com/>
19. <http://www.javatpoint.com/>
20. <http://java-course.ru/>
21. <http://www.oracle.com/>
22. <http://htmlbook.ru/>