

NETFLIX Projesi

Aybüke Özlem ULUTAŞ-Batuhan AKYOL

Bilgisayar Mühendisliği

Kocaeli Üniversitesi

180202110-180202140

Özet–Projemizde birçok film ve dizinin izlenebildiği Netflix platformunu kendi veritabanlarımızı oluşturarak bir masaüstü uygulaması yapmaktık.

AnahtarKelimeler: Veritabanı,SQL,Jframe,try-catch

I. GİRİŞ

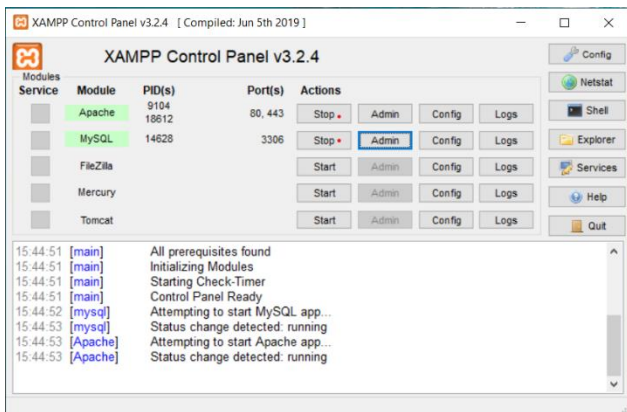
Bize verilen dosyalardaki bilgileri alıp kendimize göre veri tabanları oluşturduktan sonra birçok filmin ve dizinin izlenebileceği gerekli bilgilerin arka plandaki veritabanlarına eklenip , bilgilerin yenilendiği bir netflix uygulaması oluşturduk.

II. YÖNTEM

Programımızı java programlama dili ile geliştirdik.Ek olarak SQL veritabanı kullandık. Kodumuzda sql sorguları için bazı özel fonsiyonlar kullandık(execute(),executeupdate(),executequery()),result set prepared statement) , if-else koşul durumları ,while döngüleri ,try-catch yapısı, public static terimlerini JLabel,Jtable,Jframe,combobox ve bazı özel fonksiyonları kullanarak programımızı geliştirdik.

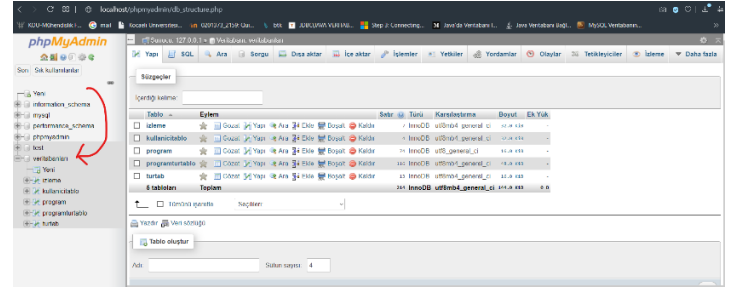
III. DENEYSEL SONUÇLAR

İlk başta kullanıcı XAMPP uygulamasını bilgisayarına yüklemelidir.Daha sonra admin kısmına tıklamalıdır.



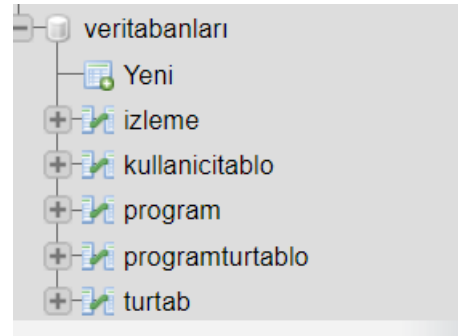
Şekil.1 Kullanıcının yüklemesi gereken XAMPP uygulaması.

Şuan resimde görülen stop kısımları uygulama açıldığında start olarak açılır.Ve kullanıcı starta basmadan uygulama veritabanlarına erişemediği için kullanılamaz.



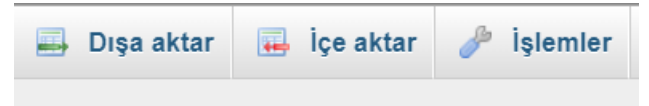
Şekil.2 Kullanıcın karşısına çıkan PhpMyAdmin sayfası.

Kullanıcı Xampp uygulamasında admin sayfasına tıkladığında karşısında böyle bir sayfa açılacaktır.Bu sayfada yapması gereken veritabanları adlı bir database oluşturmaktır.



Şekil.3 Kullanıcının oluşturması gereken veritabanları adlı database

Kullanıcı sıkıştırılmış rar dosyasını içindeki veritabanlar___.sql uzantılı dosyayı aşağıdaki içe aktar yazısına tıklayıp bu dosyayı seçip içe aktarmalıdır.



Şekil.4 Kullanıcı sıkıştırılmış rar dosyasının içindeki veritabanlar___.sql uzantılı dosyayı açma sayfası.



Şekil.5 Kullanıcının uygulamayı kurması için gerekli .exe uzantılı uygulama.

Kullanıcı yukarıda söylenen her adımı yaptıktan sonra bu .exe uzantılı dosyayı kurunca bizim oluşturduğumuz NETFLIX projesi masaüstü uygulaması olarak yüklenir.



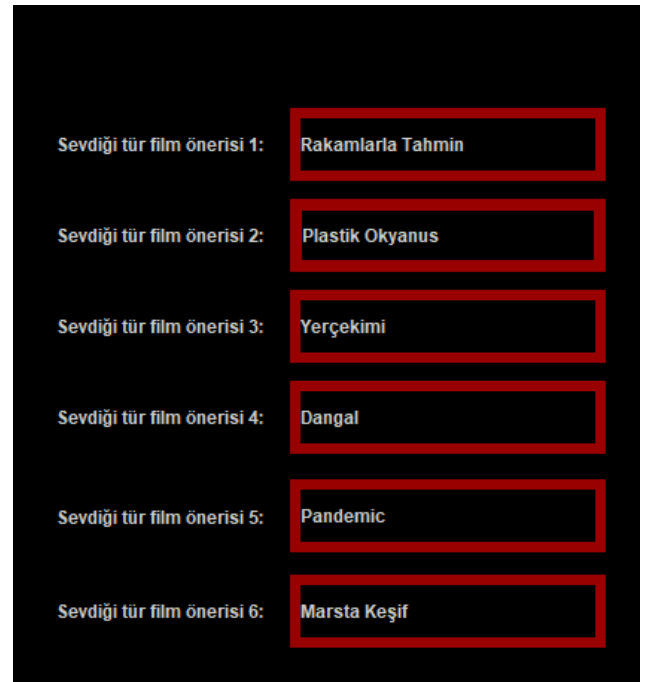
Şekil.6 Kullanıcı uygulamayı açtıktan sonra karşısına çıkan giriş penceresi.

Şuan resimde görülen açılış sayfasına kullanıcı email ve şifre ile giriş yapacaktır.Eğer kullanıcı kayıtlı değilse "Create New Account" yazısına tıklayarak kayıt olmalıdır.



Şekil.7 Kullanıcın kayıt olmak için Create New Account yazısına tıkladıktan sonra açılan kayıt sayfası.

Kullanıcı kayıt olurken yukarıda verilen İsim,Şifre,Email,DoğumTarihi(YYYY/AA/GG) ve uygulamanın ona en yüksek iki puanlı film/dizi önermesi için sevdiği 3 film/dizi türünü seçmelidir.En son Kayıt Ol butonuna bastıktan sonra kayıt olmuş olur.Ve giriş sayfasına yönlendirilip giriş yapması beklenir.Giriş yaptıktan sonra izleme ekranı karşısına çıkar.



Şekil.8 Kullanıcının karşısına sevdiği türlerin çıktığı kısım

IV. YALANCI KOD

Veritabanlarına bağlanma kodu:

```
public class DbConn {

    private String kullanıcı_adi="root";
    private String parola="";
    private String db ="veritabanları";
    private String host ="localhost";
    private String port="3306";
    private Connection con=null;
    public static Connection ConnectDb() {

        try {

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

            Connection conn =
            DriverManager.getConnection("jdbc:mysql://localhost:33
            06/veritabanları","root", "");

            return conn;
        }
        catch (Exception ex) {
            JOptionPane.showMessageDialog(null, ex);
        }
        return null;
    }
}
```

Giriş kısmında değerlerin alındığı kod parçacığı:

```
String gor = "select * from kullanicitablo ";
String sifre =
String.valueOf(jPasswordField1.getPassword());

boolean a = false;
girilenmail = ad.getText();
try (Connection conn = connect();
    Statement st = conn.createStatement();
    ResultSet rs = st.executeQuery(gor)) {
    while (rs.next()) {
        if
(rs.getString("kullaniciemail").equals(ad.getText()) &&
rs.getString("sifre").equals(sifre)) {
            a = true;
            izlemepenceresi ab = new izlemepenceresi();
            ab.setVisible(true);
        }
    }
}

"KULLANICI KAYITLI DEĞİL!!");
}

if (a == false) {
}
} catch (Exception ex) {
    System.err.println("Veri çekme başarısız!");
}
}
```

Kullanıcı kayıt eden kod parçacığı:

```
String sifre =
String.valueOf(kullaniciisifre.getPassword());

String ekle2 = "INSERT INTO kullanicitablo
(kullaniciadi,kullaniciemail,sifre,dogumtarihi,sevdigi1,sev
digi2,sevdigi3) VALUES (?, ?, ?, ?, ?, ?)";

try (Connection conn = connect(); PreparedStatement
pstmt = conn.prepareStatement(ekle2);) {

    pstmt.setString(1, kullaniciisim.getText());
    pstmt.setString(2, kullaniciemail.getText());
    pstmt.setString(3, sifre);
    pstmt.setString(4, kullaniciidg.getText());
    pstmt.setString(5, secilenler[0]);
    pstmt.setString(6, secilenler[1]);
    pstmt.setString(7, secilenler[2]);
    pstmt.executeUpdate();
} catch (Exception ex) {
}

if (sayac == 3) {
    this.dispose();
}
}
```

Kullanıcın seçtiği türlere göre en yüksek iki puana sahip film/dizi yazdıran kod parçacığı:

```
String tur = "SELECT * FROM program WHERE
program.progid IN(SELECT programturtablo.progid
FROM programturtablo WHERE programturtablo.turid
IN( SELECT turtab.turid FROM turtab WHERE
turtab.turad LIKE (SELECT kullanicitablo.sevdigi1
FROM kullanicitablo WHERE
kullanicitablo.kullaniciemail='" + Login_Form.girilenmail
+ "'))order by puanlama desc LIMIT 0,2";

try (Connection conn = connect(); Statement st =
conn.createStatement(); ResultSet rs =
st.executeQuery(tur);) {
    rs.next();
    String ad = rs.getString("progadi");
    jLabel1.setText(ad);
    rs.next();
    ad = rs.getString("progadi");
    jLabel2.setText(ad);
} catch (Exception ex) {
}

String tura = "SELECT * FROM program WHERE
program.progid IN(SELECT programturtablo.progid
FROM programturtablo WHERE programturtablo.turid
IN( SELECT turtab.turid FROM turtab WHERE
turtab.turad LIKE (SELECT kullanicitablo.sevdigi2
FROM kullanicitablo WHERE
```

```
kullanicitablo.kullaniciemail="" + Login_Form.girilenmail
+ ""))order by puanlama desc LIMIT 0,2";
```

```
try (Connection conn = connect(); Statement sta =
conn.createStatement(); ResultSet rsa =
sta.executeQuery(tura);) {
    rsa.next();
    String ada = rsa.getString("progadi");
    jLabel3.setText(ada);
    rsa.next();
    ada = rsa.getString("progadi");
    jLabel4.setText(ada);
} catch (Exception ex) {
}
```

```
String turb = "SELECT * FROM program WHERE
program.progid IN(SELECT programturtablo.progid
FROM programturtablo WHERE programturtablo.turid
IN( SELECT turtab.turid FROM turtab WHERE
turtab.turad LIKE (SELECT kullanicitablo.sevdigi3
FROM kullanicitablo WHERE
kullanicitablo.kullaniciemail="" + Login_Form.girilenmail
+ ""))order by puanlama desc LIMIT 0,2";
```

```
try (Connection conn = connect(); Statement stb =
conn.createStatement(); ResultSet rsb =
stb.executeQuery(turb);) {
    rsb.next();
    String adb = rsb.getString("progadi");
    jLabel5.setText(adb);
    rsb.next();
    adb = rsb.getString("progadi");
    jLabel6.setText(adb);
} catch (Exception ex) {
}
String tural = tip.getText();
}
```

Yazılan türe göre filmleri listeleyen kod parçasığı:

```
public void proglis(String turtip) {
    try {
        String query1 = "SELECT * FROM program
WHERE tipi LIKE '%" + turtip + "%'";
        Statement st = c.createStatement();
        ResultSet rs = st.executeQuery(query1);
        Object[] row = new Object[7];
        DefaultTableModel model = (DefaultTableModel)
jTable1.getModel();
        while (rs.next()) {
            row[0] = rs.getString(1);
            row[1] = rs.getString(2);
            row[2] = rs.getString(3);
            row[3] = rs.getString(4);
            row[4] = rs.getString(5);
            row[5] = rs.getString(6);
            row[6] = rs.getString(7);
            model.addRow(row);
        }
    }
```

```
} catch (Exception e) {
}
}
```

Yazılan isme göre filmleri listeleyen kod parçasığı: :

```
public void proglis2(String turtip) {    try {
    String query1 = "SELECT * FROM program
WHERE progadi LIKE '%" + turtip + "%'";
    Statement st = c.createStatement();
    ResultSet rs = st.executeQuery(query1);
    Object[] row = new Object[7];
    DefaultTableModel model = (DefaultTableModel)
jTable1.getModel();
    while (rs.next()) {

        row[0] = rs.getString(1);
        row[1] = rs.getString(2);
        row[2] = rs.getString(3);
        row[3] = rs.getString(4);
        row[4] = rs.getString(5);
        row[5] = rs.getString(6);
        row[6] = rs.getString(7);
        model.addRow(row);
    }
} catch (Exception e) {
} }
```

Bilgisayar saatinin alındığı kod parçasığı:

```
String izleme="SELECT * FROM izleme";
try (Connection conn = connect();
    Statement st = conn.createStatement();
    ResultSet rs = st.executeQuery(izleme)) {
    while (rs.next()) {

        if(rs.getString("progid").equals(izlemepenceresi.idd))
        {
            gecensani=rs.getInt("izlemesn");
            gecendk=rs.getInt("izlemedk");
            saat=rs.getInt("izlemesaat");
        }
    }
} catch (Exception ex) {
    System.err.println("Veri çekme başarısız!");
}

t=new Timer(1000, (ActionEvent e) -> {
    if(gecensani > 59)
    {
        gecendk++;
        gecensani /=60;
        gecensani--;
        if(gecendk>59)
        {
            saat++;
            gecendk/=60;
            gecendk--;
        }
    }
}
```

```

    }

zaman.setText(String.valueOf(saat)+"."+String.valueOf(gecendk)+":"+String.valueOf(gecensani));
    gecensani++;
    });
    t.start();

```

İzleme databasenin update edildiği kod parçası:

```

boolean count = true;
try {

    String abc = "SELECT * FROM izleme";
    Connection conn = connect();
    Statement st = conn.createStatement();
    ResultSet rs = st.executeQuery(abc);

    while (rs.next()) {

        if
(rs.getString("kullaniciemail").equals(Login_Form.girilen
mail) && cont == 0) {

            if
(rs.getString("progid").equals(izlemepenceresi.idd)) {
                //0 System.out.println("<kjdnkisvd");
                count = false;
                String ekle32 = "UPDATE izleme SET
izlemedk=(?) WHERE kullaniciemail=(?)";
                String s1 = "UPDATE izleme SET
progid=(?) WHERE kullaniciemail=(?)";
                String s2 = "UPDATE izleme SET
izlemesn=(?) WHERE kullaniciemail=(?)";
                String s3 = "UPDATE izleme SET
izlemesaat=(?) WHERE kullaniciemail=(?)";
                String s4 = "UPDATE izleme SET
izlemebolum=(?) WHERE kullaniciemail=(?)";
                String s5 = "UPDATE izleme SET
izlemetarih=(?) WHERE kullaniciemail=(?)";
                String s6 = "UPDATE izleme SET
izlemepuan=(?) WHERE kullaniciemail=(?)";

                try (Connection connna = connect());
                PreparedStatement pstmt2 =
connna.prepareStatement(ekle32); PreparedStatement
pstmt3 = connna.prepareStatement(s1);
                PreparedStatement pstmt4 =
connna.prepareStatement(s2); PreparedStatement pstmt5
= connna.prepareStatement(s3); PreparedStatement
pstmt6 = connna.prepareStatement(s4);
                PreparedStatement pstmt7 =
connna.prepareStatement(s5); PreparedStatement pstmt8
= connna.prepareStatement(s6);) {
                    pstmt2.setInt(1, progizleme.gecendk);
                    pstmt2.setString(2,
Login_Form.girilenmail);
                    pstmt3.setString(1, izlemepenceresi.idd);
                    pstmt3.setString(2,
Login_Form.girilenmail);
                    pstmt4.setInt(1, progizleme.gecensani);

```

```

                    pstmt4.setString(2,
Login_Form.girilenmail);
                    pstmt5.setInt(1, progizleme.saat);
                    pstmt5.setString(2,
Login_Form.girilenmail);
                    pstmt6.setInt(1, progizleme.bolum);
                    pstmt6.setString(2,
Login_Form.girilenmail);
                    pstmt7.setString(1,
progizleme.sistemTarihiniGetir("yyyy/MM/dd"));
                    pstmt7.setString(2,
Login_Form.girilenmail);
                    pstmt8.setFloat(1, veripuan);
                    pstmt8.setString(2,
Login_Form.girilenmail);

                    pstmt2.executeUpdate();
                    pstmt3.executeUpdate();
                    pstmt4.executeUpdate();
                    pstmt5.executeUpdate();
                    pstmt6.executeUpdate();
                    pstmt7.executeUpdate();
                    pstmt8.executeUpdate();
                } catch (Exception ex) {
                }

                cont++;
            } else if (cont == 1) {
                String ekle32 = "INSERT INTO izleme
(izlemetarih,izlemedk,izlemesn,izlemesaat,izlemebolum,iz
lemepuan,progid,kullaniciemail) VALUES
(?,?,?,?,?,?,?,?)";

                try (Connection connnb = connect());
                PreparedStatement pstmt =
connnb.prepareStatement(ekle32);) {
                    pstmt.setString(1,
progizleme.sistemTarihiniGetir("yyyy/MM/dd"));
                    pstmt.setInt(2, progizleme.gecendk);
                    pstmt.setInt(3, progizleme.gecensani);
                    pstmt.setInt(4, progizleme.saat);
                    pstmt.setInt(5, progizleme.bolum);
                    pstmt.setFloat(6, veripuan);
                    pstmt.setString(7, izlemepenceresi.idd);
                    pstmt.setString(8,
Login_Form.girilenmail);

                    pstmt.executeUpdate();
                } catch (Exception ex) {
                }

            }

        }

        if (count == true) {
            String ekle3 = "INSERT INTO izleme
(izlemetarih,izlemedk,izlemesn,izlemesaat,izlemebolum,iz
lemepuan,progid,kullaniciemail) VALUES
(?,?,?,?,?,?,?,?)";

            try (Connection connnv = connect());
            PreparedStatement pstmt15 =
connnv.prepareStatement(ekle3);) {

```

```

        pstmt15.setString(1,
progizleme.sistemTarihiniGetir("yyyy/MM/dd"));
        pstmt15.setInt(2, progizleme.gecendk);
        pstmt15.setInt(3, progizleme.gecensani);
        pstmt15.setInt(4, progizleme.saat);
        pstmt15.setInt(5, progizleme.bolum);
        pstmt15.setFloat(6, veripuan);
        pstmt15.setString(7, izlemepenceresi.idd);
        pstmt15.setString(8,
Login_Form.girilenmail);

        pstmt15.executeUpdate();

```

```

    } catch (Exception ex) {
    }
} catch (Exception ex) {
}
Oylama sayının ve puanların değiştirildiği kod parçacığı:

```

```

int index = jComboBox1.getSelectedIndex();
veripuan = Integer.valueOf(index) + 1;
System.out.println(veripuan);

float yenipuan;
int yenip;
String as = "SELECT * FROM program";

try {
    Connection conn = connect();
    Statement st = conn.createStatement();
    ResultSet rs = st.executeQuery(as);
    while (rs.next()) {

        if
(rs.getString("progid").equals(izlemepenceresi.idd)) {
            yenip = rs.getInt("oylamasayisi") + 1;
            yenipuan = ((rs.getInt("oylamasayisi") *
rs.getFloat("puanlama")) + veripuan) / yenip;
            String ekle30 = "UPDATE program SET
puanlama=(?) WHERE progadi= (?)";
            String ekle33 = "UPDATE program SET
oylamasayisi=(?) WHERE progadi= (?)";
            System.out.println(ekle30);
            try (Connection connnvz = connect();
PreparedStatement pstmt =
connnvz.prepareStatement(ekle30); PreparedStatement
pstmt1 = connnvz.prepareStatement(ekle33);) {
                pstmt.setFloat(1, yenipuan);
                pstmt.setString(2, izlemepenceresi.adii);
                pstmt1.setInt(1, yenip);
                pstmt1.setString(2, izlemepenceresi.adii);
                pstmt.executeUpdate();
            } catch (Exception ex) {

        } catch (Exception ex) {
    }
}

```

V. SONUÇ

Bu projeyi geliştirirken veri tabanı ile ilgili işlemleri javada kullanmayı öğrendik. Bu proje ile veritabanı yönetim sistemlerinin yapısını anladık. Arayüz tasarlamayı daha iyi öğrendik. Bir java projesini nasıl Masaüstü uygulamasına çevirileceğini öğrendik. Bu projede en zorlandığımız yer verileri update etme kısmıdır.

VI. VARLIK-İLİŞKİ DİYAGRAMLARI:

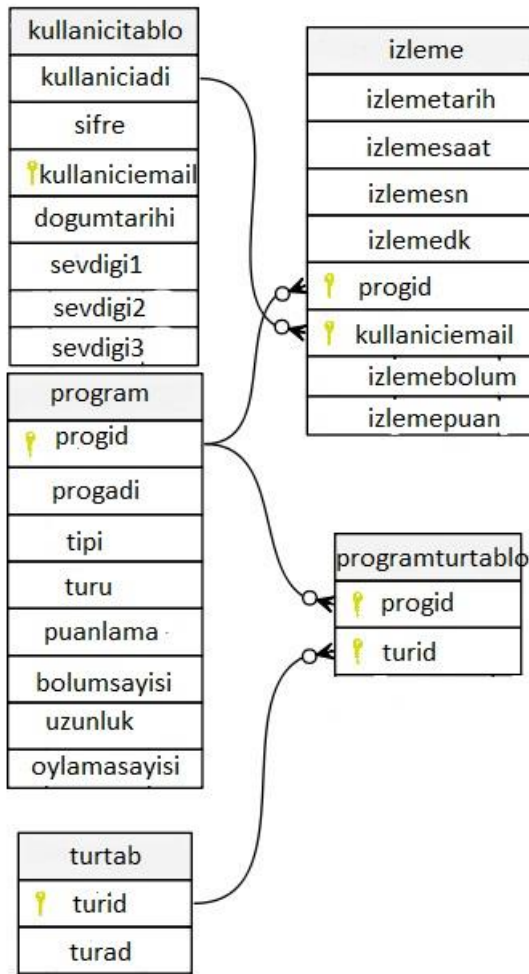
Veritabanı: kullanicitablo
kullaniciadi : varchar(50)
sifre : varchar(50)
kullaniciemail : varchar(50)
dogumtarihi : date
sevdigi1 : varchar(50)
sevdigi2 : varchar(50)
sevdigi3 : varchar(50)

Veritabanı: program
progid : int(11)
progadi : varchar(50)
tipi : varchar(67)
turu : varchar(50)
puanlama : float
bolumsayisi : int(50)
uzunluk : time
oylamasayisi : int(11)

Veritabanı: turtab
tunid : int(11)
turad : varchar(50)

Veritabanı: programturtablo
progid : int(11)
tunid : int(11)

Veritabanı: izleme
kullaniciemail : varchar(50)
progid : int(11)
izlemetarih : date
izlemesaat : int(11)
izlemesni : int(11)
izlemelik : int(11)
izlemebolum : int(11)
izlemepuan : int(11)



<https://teknikakil.com/veritabani/sql-server/sql-server-update-ile-kayit-guncelleme/>

Java JDBC. (2019, October 3). Yusuf SEZER.

<https://www.yusufsezer.com.tr/java-jdbc/>

Java Veritabanı Bağlantısı (JDBC). (n.d.). Java Yaz... – Sadece Java.

https://javayaz.com/?page_id=114

KABAKCI, F. (n.d.). Jdbc(Java veritabanı bağlanabilirliği). Fatih KABAKCI.

[https://fatihkabakci.com/article-JDBC\(JAVA-VERITABANI-BAGLANABILIRLIGI\)](https://fatihkabakci.com/article-JDBC(JAVA-VERITABANI-BAGLANABILIRLIGI))

VII. KAYNAKÇA

C#, JAVA,PHP, programming ,Source code: Java and MySQL - Insert update delete and display data on JTable. (n.d.). C#, JAVA,PHP, Programming ,Source Code.

<https://1bestcsharp.blogspot.com/2016/01/java-and-mysql-insert-update-delete-display.html>

David-Engel. (n.d.). Step 3: Connecting to SQL using Java - SQL server. Technical documentation, API, and code examples | Microsoft Docs.

<https://docs.microsoft.com/en-us/sql/connect/jdbc/step-3-proof-of-concept-connecting-to-sql-using-java?view=sql-server-ver15>

HÜSEM, S. Ş. (2020, February 7). SQL server – UPDATE ile Kayıt Güncelleme. TEKNİK AKIL.