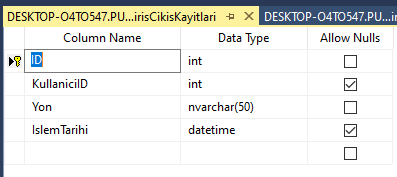
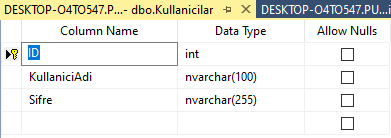
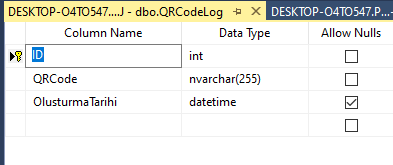
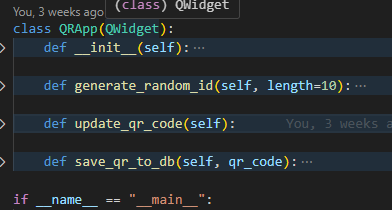
  
  
Mssql Diyagram  
  
Kullnicilar.ID GirisCikisKayitlari.KullaniciID Foreign key ile birbirine bağlanmıştır.



  
  
  
  
  
  
  
2\_QR\_Viewer.py  
  
  
  
3 saniyede bir random değer oluşturacak başına gerekli url’yi ekleyecek ve ekrana qr kodu olarak güncelleyecek ve güncel olmayan 30 saniye öncesinden eklenmiş qr kodlarını sistemden silerek çalışan python kodu

def \_\_init\_\_(self):

        super().\_\_init\_\_()

        self.setWindowTitle("QR Kod Gösterici")

        self.showFullScreen()  # Tam ekran aç

        layout = QVBoxLayout()

        # QR Kod Görseli

        self.qr\_label = QLabel(self)

        self.qr\_label.setAlignment(Qt.AlignCenter)

        self.qr\_label.setScaledContents(True)  # Görseli pencereye sığdır

        layout.addWidget(self.qr\_label)

        self.setLayout(layout)

        # Timer: QR kodu her 3 saniyede bir güncelle

        self.timer = QTimer(self)

        self.timer.timeout.connect(self.update\_qr\_code)

        self.timer.start(3000)

        # İlk QR kodunu oluştur

        self.update\_qr\_code()

Görselin tam ekran olması ve oluşturulan qrkodun ilgili ekrana sığması için init fonksiyonun altı bu şekilde yapılandırılmıştır.  
Qrkod’u 3 saniyede bir güncellenmesi için timer burada tanımlanmıştır.

    def generate\_random\_id(self, length=10):

        """Rastgele ID oluştur"""

        return ''.join(random.choices(string.ascii\_letters + string.digits, k=length))

Random id’yi istenilen format bu fonksiyon ile belirlenmiştir.

    def update\_qr\_code(self):

        """QR kodu güncelle ve veritabanına kaydet"""

        random\_id = self.generate\_random\_id()

        #page\_url = f"http://example.com/user?id={random\_id}"  # Yeni URL

        page\_url = f"http://192.168.1.104/Home/Index?QRData={random\_id}"  # Yeni URL

        # QR Kod Oluşturma

        qr = pyqrcode.create(page\_url)

        qr.png("qrcode.png", scale=30)

        # QR Kodu Güncelle

        pixmap = QPixmap("qrcode.png")

        self.qr\_label.setPixmap(pixmap)

        # QR kodunu veritabanına kaydet

        self.save\_qr\_to\_db(random\_id)

istenilen qr kodu üretecek ve başına ilgili url için ip, path ve parametre ismini ekleyecektir.  
pyqrcode kütüphanes, kullanılarak oluşturulması ve ekrana aktarılması sağlanacaktır.

 def save\_qr\_to\_db(self, qr\_code):

        """MS SQL Veritabanına QR Kodunu Kaydet"""

        try:

            conn = pyodbc.connect(DB\_CONNECTION)

            cursor = conn.cursor()

            cursor.execute("""

                INSERT INTO QRCodeLog (QRCode)

                VALUES (?)

            """, (qr\_code))  # KullaniciID = 1 (Example)

            conn.commit()

            conn.close()

            print(f"QR Code saved: {qr\_code}")

        except Exception as e:

            print(f"Database Error: {e}")

        try:

            conn = pyodbc.connect(DB\_CONNECTION)

            cursor = conn.cursor()

            cursor.execute("DELETE FROM QRCodeLog WHERE DATEDIFF(SECOND, OlusturmaTarihi, GETDATE()) > 120;")

            conn.commit()

            conn.close()

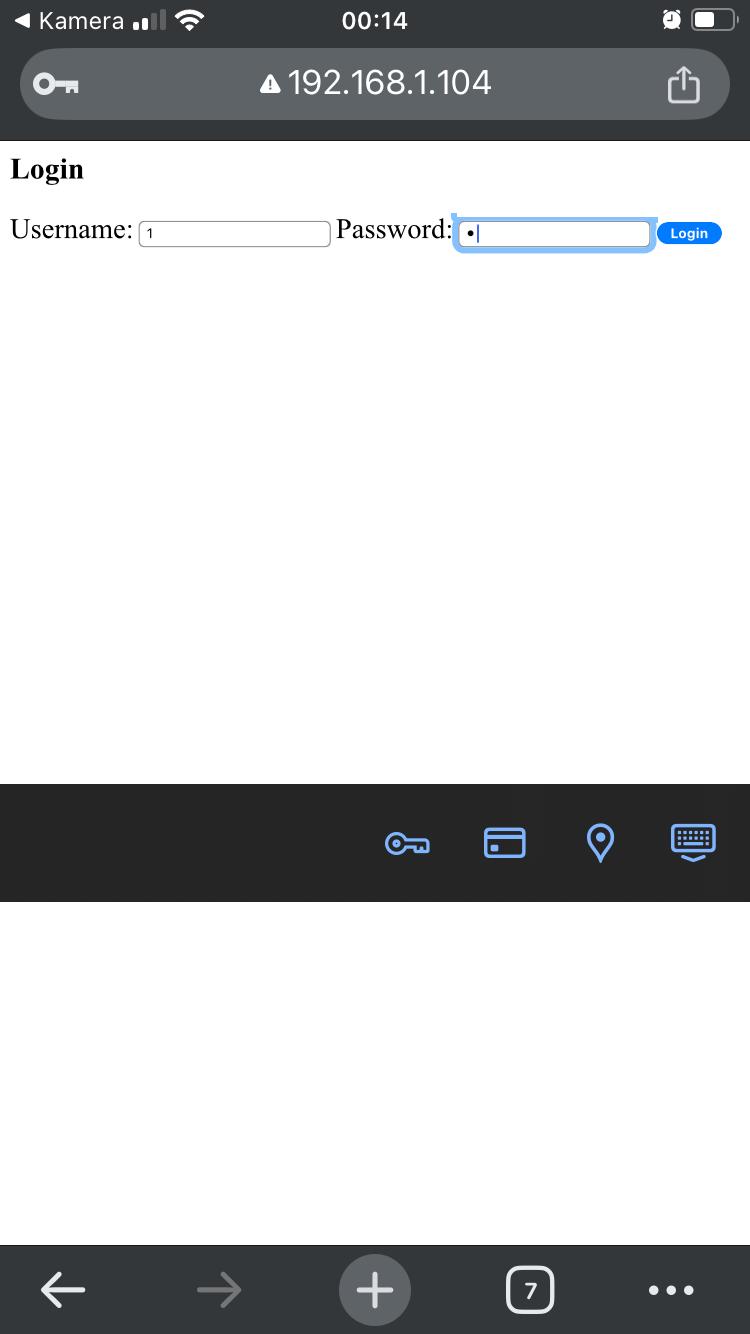
            print("Old QR codes deleted.")

        except Exception as e:

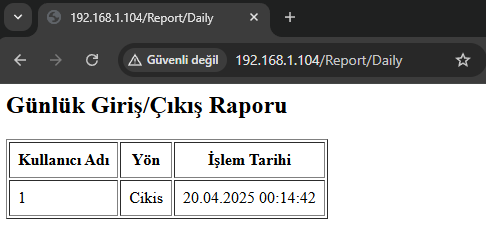
            print(f"Error deleting old QR codes: {e}")

üretilen qr kod dataabasede saklanılarak kullanıcıların işlem süresince son 120 saniyede üretilmiş kodla işlem yapmasına izin verilecek şekle getirilmiştir.



Çalıştırıldığında tam ekran olarak açılır ve qr kodu karşılığı ilgili resim için : <http://192.168.1.104/Home/Index?QRData=yirp48O1e9>  
  
  
  
  
kullanıcı ilk kod okutmasında id ve şifre girmesi istenmektedir cookie’ler ile birdaha istenmemesi için gerekli ayarlamalar yapılmıştır.



Kullanıcı giriş yaptıysa çıkış yapması , yoksa giriş yapması ilk işlemi ise giriş yapması olarak tanımlanmıştır.   
  
  
Raporlama için güvenlik kontrolü olmadan verilen link ile işlem yapılacaktır.

Login.cshtml

@{

    ViewData["Title"] = "Login";

}

<h2>Login</h2>

@if (ViewBag.Error != null)

{

    <p style="color:red;">@ViewBag.Error</p>

}

<form method="post">

    <label>Username:</label>

    <input type="text" name="username" required />

    <label>Password:</label>

    <input type="password" name="password" required />

    <button type="submit">Login</button>

</form>

Arayüz sayfası olarak sadece kullanıcı adı,şifre ve giriş butonundan oluşan sade bir sayfa

AuthController.cs

using Microsoft.AspNetCore.Mvc;

using Microsoft.AspNetCore.Http;

using System;

using System.Data.SqlClient;

using Microsoft.Data.SqlClient;

namespace QRScannerWeb.Controllers

{

    public class AuthController : Controller

    {

        private readonly string \_connectionString = "Server=localhost;Database=PUANTAJ;Trusted\_Connection=True;TrustServerCertificate=True;";

        public IActionResult Login()

        {

            return View();

        }

        [HttpPost]

        public IActionResult Login(string username, string password)

        {

            if (IsValidUser(username, password))

            {

                // Store username in a cookie (valid for 7 days)

                Response.Cookies.Append("UserName", username, new CookieOptions

                {

                    Expires = DateTime.Now.AddDays(7)

                });

                // Retrieve QRData from cookies (if available)

                string qrData = Request.Cookies["QRData"];

                if (!string.IsNullOrEmpty(qrData))

                {

                    return RedirectToAction("Index", "Home", new { qrData });

                }

                return RedirectToAction("Index", "Home");

            }

            ViewBag.Error = "Invalid username or password.";

            return View();

        }

        private bool IsValidUser(string username, string password)

        {

            using (SqlConnection conn = new SqlConnection(\_connectionString))

            {

                conn.Open();

                string query = "SELECT COUNT(\*) FROM Kullanicilar WHERE KullaniciAdi = @username AND Sifre = @password";

                SqlCommand cmd = new SqlCommand(query, conn);

                cmd.Parameters.AddWithValue("@username", username);

                cmd.Parameters.AddWithValue("@password", password);

                int userCount = (int)cmd.ExecuteScalar();

                return userCount > 0;

            }

        }

        public IActionResult Logout()

        {

            Response.Cookies.Delete("UserName"); // Remove username cookie

            Response.Cookies.Delete("QRData");   // Remove QRData cookie

            return RedirectToAction("Login");

        }

    }

}

Kullanıcı adı ve şifresini ilgili database üzerinden kontrol eder, doğru bulunması halinde cookilere kayıt ederek saklanması hedeflenmektedir. Cookilerde bulunması halinde otomatik giriş yapılarak sonraki sayfaya geçilmesi için gerekli düzenlemeler yapılmıştır. Çıkış ypaılması durumunda ise cookiler temizlenerek sayfa yenilenmektedir.

Index.cshtml

@{

    ViewData["Title"] = "Home";

}

<h2>Welcome, @ViewBag.UserName</h2>

@if (!string.IsNullOrEmpty(ViewBag.QRData))

{

    <p>Scanned QR Code: <strong>@ViewBag.QRData</strong></p>

}

@if (!string.IsNullOrEmpty(ViewBag.Message))

{

    <p style="color: green;"><strong>@ViewBag.Message</strong></p>

}

<a href="/Auth/Logout">Logout</a>

qr kod okutulduğundaki ilk karşılama sayfası için olan arayüz giriş kodunu kullanıcı adını ve giriş veya çıkış yapıldığını bildiren mesajı gösterdiğimiz sayfanın arayüz kodu bu şekildedir.

HomeController.cs

using Microsoft.AspNetCore.Mvc;

using Microsoft.Data.SqlClient;

using System;

using System.Data;

namespace QRScannerWeb.Controllers

{

    public class HomeController : Controller

    {

        private readonly string \_connectionString = "Server=localhost;Database=PUANTAJ;Trusted\_Connection=True;TrustServerCertificate=True;";

        public IActionResult Index(string qrData)

        {

            string username = Request.Cookies["UserName"];

            if (string.IsNullOrEmpty(username))

            {

                if (!string.IsNullOrEmpty(qrData))

                {

                    Response.Cookies.Append("QRData", qrData, new CookieOptions

                    {

                        Expires = DateTime.Now.AddMinutes(10)

                    });

                }

                return RedirectToAction("Login", "Auth");

            }

            if (string.IsNullOrEmpty(qrData))

            {

                qrData = Request.Cookies["QRData"];

            }

            string message = "QR Code not found.";

            if (!string.IsNullOrEmpty(qrData) && !IsValidQRCode(qrData))

            {

                ViewBag.Message = "Invalid or expired QR code.";

                qrData = null; // Reset invalid QR data

                return View();

            }

            if (!string.IsNullOrEmpty(qrData))

            {

                message = ProcessQRScan(qrData);

            }

            ViewBag.UserName = username;

            ViewBag.QRData = qrData;

            ViewBag.Message = message;

            return View();

        }

        private string ProcessQRScan(string qrData)

        {

            string username = Request.Cookies["UserName"];

            if (string.IsNullOrEmpty(username))

            {

                if (!string.IsNullOrEmpty(qrData))

                {

                    Response.Cookies.Append("QRData", qrData, new CookieOptions

                    {

                        Expires = DateTime.Now.AddMinutes(10)

                    });

                }

                RedirectToAction("Login", "Auth");

            }

            using (SqlConnection conn = new SqlConnection(\_connectionString))

            {

                conn.Open();

                string userIdQuery = "SELECT ID FROM Kullanicilar WHERE KullaniciAdi = @QRCode";

                int userId = -1;

                using (SqlCommand cmd = new SqlCommand(userIdQuery, conn))

                {

                    cmd.Parameters.AddWithValue("@QRCode", username);

                    var result = cmd.ExecuteScalar();

                    if (result != null)

                    {

                        userId = Convert.ToInt32(result);

                    }

                }

                if (userId == -1)

                {

                    return "User not found.";

                }

                string lastRecordQuery = "SELECT [Yon] FROM GirisCikisKayitlari WHERE [KullaniciID] = @UserID ORDER BY [IslemTarihi] DESC";

                string lastDirection = "";

                using (SqlCommand cmd = new SqlCommand(lastRecordQuery, conn))

                {

                    cmd.Parameters.AddWithValue("@UserID", userId);

                    var result = cmd.ExecuteScalar();

                    if (result != null)

                    {

                        lastDirection = result.ToString();

                    }

                }

                string newDirection = (lastDirection == "Giris") ? "Cikis" : "Giris";

                string insertQuery = "INSERT INTO GirisCikisKayitlari (KullaniciID, Yon, IslemTarihi) VALUES (@UserID, @Direction, GETDATE())";

                using (SqlCommand cmd = new SqlCommand(insertQuery, conn))

                {

                    cmd.Parameters.AddWithValue("@UserID", userId);

                    cmd.Parameters.AddWithValue("@Direction", newDirection);

                    cmd.ExecuteNonQuery();

                }

                return $"User {userId} has been marked as {newDirection}.";

            }

        }

        private bool IsValidQRCode(string qrData)

        {

            using (SqlConnection conn = new SqlConnection(\_connectionString))

            {

                conn.Open();

                string query = "SELECT COUNT(\*) FROM QRCodeLog WHERE QRCode = @qrData -- AND OlusturmaTarihi > DATEADD(SECOND, -120, GETDATE())";

                SqlCommand cmd = new SqlCommand(query, conn);

                cmd.Parameters.AddWithValue("@qrData", qrData);

                int qrCount = (int)cmd.ExecuteScalar();

                return qrCount > 0;

            }

        }

    }

}

İlgili kullanıcının giriş veya çıkış işlemini kaydeden ve onaylayan ana kod sayfası burasıdır. QRkodun db’de olup olmadığını kontrol ederek işleme başlar.kullanıcı adı üzerinden db üzerine son giriş çıkış kaydını kontrol ederek yön belirler ve bu kaydı db’ye kaydeder.

Daily.cshtml

@model List<QRScannerWeb.Controllers.ReportController.DailyEntry>

@{

    ViewData["Title"] = "Günlük Giriş/Çıkış Raporu";

}

<h2>@ViewData["Title"]</h2>

<table border="1" cellpadding="8">

    <thead>

        <tr>

            <th>Kullanıcı Adı</th>

            <th>Yön</th>

            <th>İşlem Tarihi</th>

        </tr>

    </thead>

    <tbody>

        @if (Model.Count > 0)

        {

            foreach (var entry in Model)

            {

                <tr>

                    <td>@entry.AdSoyad</td>

                    <td>@entry.Yon</td>

                    <td>@entry.IslemTarihi.ToString("dd.MM.yyyy HH:mm:ss")</td>

                </tr>

            }

        }

        else

        {

            <tr><td colspan="3">Bugün için kayıt bulunamadı.</td></tr>

        }

    </tbody>

</table>

Tablo olarak günlük o güne ait giriş çıkış kayıtlarını listelemesi için sade bir tasarım

ReportController.cs

using Microsoft.AspNetCore.Mvc;

using Microsoft.Data.SqlClient;

using System;

using System.Collections.Generic;

using System.Data;

namespace QRScannerWeb.Controllers

{

    public class ReportController : Controller

    {

        private readonly string \_connectionString = "Server=localhost;Database=PUANTAJ;Trusted\_Connection=True;TrustServerCertificate=True;";

        public IActionResult Daily()

        {

            List<DailyEntry> entries = new List<DailyEntry>();

            using (SqlConnection conn = new SqlConnection(\_connectionString))

            {

                conn.Open();

                string sql = @"

SELECT k.KullaniciAdi, g.Yon, g.IslemTarihi

FROM GirisCikisKayitlari g

INNER JOIN Kullanicilar k ON k.ID = g.KullaniciID

WHERE CAST(g.IslemTarihi AS DATE) = CAST(GETDATE() AS DATE)

ORDER BY g.IslemTarihi DESC

";

                using (SqlCommand cmd = new SqlCommand(sql, conn))

                using (SqlDataReader reader = cmd.ExecuteReader())

                {

                    while (reader.Read())

                    {

                        entries.Add(new DailyEntry

                        {

                            AdSoyad = reader.GetString(0),

                            Yon = reader.GetString(1),

                            IslemTarihi = reader.GetDateTime(2)

                        });

                    }

                }

            }

            return View(entries);

        }

        public class DailyEntry

        {

            public string AdSoyad { get; set; }

            public string Yon { get; set; }

            public DateTime IslemTarihi { get; set; }

        }

    }

}

İlgili dbye o güne ait sleect atarak tabloyu doldurması sağlanır.