ANKARA ÜNİVERSİTESİ MÜHENDİSLİK FAKÜLTESİ BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ



BLM 4537 PROJE RAPORU

Security for Everyone Blog-App

Ömer Faruk Eren

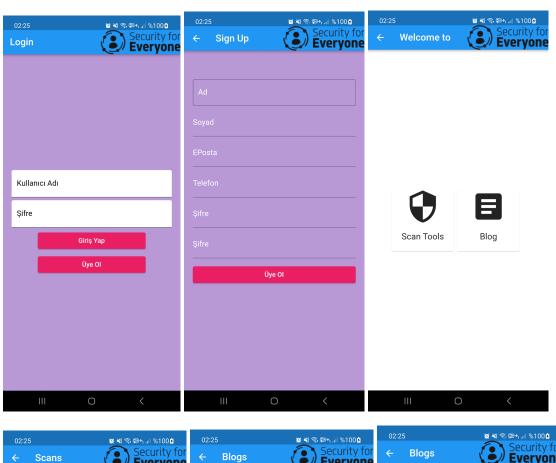
19290242

Enver Bağcı

"Security for Everyone" bloglarının yayınlanması ve sunulan scan türlerinin tanıtılması için geliştirdiğim taslak uygulamadır. Firebase ile üye olunup giriş yapılabilmektedir. Scan ve Blog sayfaları, verilen json dosyalarından çekilip dosya devam ettiği sürece akışına devam ederek ilerlemektedir. Ben 6 blogluk bir json ve 100 scanlik bir json dosyasını siteden indirdim. Siteye atılacak bir fetch sonucu alınan bir json entegre edilirse sitedeki her veriyi alacak şekilde de değiştirilebilir ama bunu yapmak yerine local dosya ile çalıştırdım. Json dosyalarından çekilen html kodunu anlamlandıracak uygun kütüphaneleri ve kodları da kullanarak blog ve scan ekranlarını şekillendirdim. Bu sayede daha gerçekçi görüntüye ulaştım. Json dosyaları ve blogların resimleri images klasöründedir. Uygulama 5 sayfadan oluşmaktadır. Login, Sign in sayfaları dışında admin, scan ve blog sayfaları vardır. Kullanıcı admin sayfasından blog veya scanleri açmayı tercih edebilir.

Ekranlar ve onlara ait kodlardan kısımlar:

Tüm Ekranlar:





The CRLF (\r\n) abbreviation refers to Carriage Return and Line Feed. A CRLF injection attack is a type of injection attack that exploits the combination of a carriage return and line feed characters, which are used to end a line of text in a file or command.



Generic CSRF Vulnerability Scanner

Cross-Site Request Forgery, also known as CSRF is an attack vector that tricks a web browser into sending malicious requests to a web application on behalf of the user that is currently authenticated. These requests can perform any action that the user is authorized to do, such as changing their password, making purchases, or posting comments.



Generic SSRF vulnerability scanner

Server-side request forgery (SSRF), is a vulnerability that allows an attacker to execute unauthorized requests from the perspective of the webserver. SSRF can be used to access sensitive data, such as internal network resources and user data, or to launch attacks on other



The Need for Website Security Scans

You might not think of it as a priority, but website security scans are essential to the health of your website. A full security scan will detect any possible vulnerabilities and identify what needs to be done to protect your site from malicious attacks. In this blog post, we will discuss why it is important to make a full security scan of your website regularly.

The Benefits of Security Scans

Security scans serve an important purpose in protecting your website from hackers and cybercriminals. They can detect any potential vulnerabilities that could expose your site to malicious activity, such as data corruption or data breaches. Regular security scans also help ensure that all security measures are up-to-date and functioning correctly. Additionally, they can help you identify any areas where you need to make further improvements to better protect your site from attack.

Minimize Risk and Protect Your Reputation

By regularly performing security scans, you can minimize the risk of being a victim of a cyberattack. Security breaches can have devastating consequences



scans in order to ensure maximum safety and protection against malicious attacks. These scans should occur at least once every month or two depending on the size and traffic of the website, and should include an examination for suspicious activity or irregularities in user behavior along with changes made within the codebase that could introduce new vulnerabilities into the system. Another and the most recommended method is using a continuous scanner so that any changes made to the site can be monitored and identified quickly. By making sure these steps are taken regularly, you can rest assured that your website remains secure throughout its lifetime online!

Know in 2023

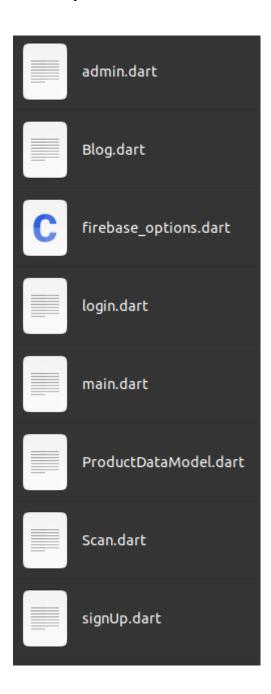
5 Key Cybersecurity Terms You Should

5 Key Cybersecurity Terms You Should Know in 2023

Cybersecurity is a rapidly evolving field that requires professionals to stay on their toes at all times. As technology and the threat landscape change, new cybersecurity terms are being introduced every day. To help keep you up to date with the latest trends, we've compiled a list of five cybersecurity keywords you should know in 2023.

1 - Al-Driven Security Solutions

Tüm dosyaların listesi:



Model dosyası:

```
class ProductDataModel{
                          The file name 'ProductDataMo
 String? mini desc; The variable name 'mini desc'
 String? meta title;
 String? name;
 String? article;
 String? img banner;
 ProductDataModel(
     this.mini desc, The variable name 'mini desc'
     this.meta title,
                        The variable name 'meta title
     this.name,
     this.article,
     this.img banner, The variable name 'img banner
     });
 ProductDataModel.fromJson(Map<String,dynamic> json)
   mini desc =json['mini desc'];
   meta title = json['meta title'];
   name = json['name'];
   article = json['article'];
   img banner = json['img banner'];
```

Json çekme:

```
Future<List<ProductDataModel>> ReadJsonData() async {         The variable name 'Re
          final jsondata = await rootBundle.rootBundle.loadString('images/scan.json');
          final list = json.decode(jsondata) as List<dynamic>;
          return list.map((e) => ProductDataModel.fromJson(e)).toList();
    }
}
```

sign up page kod parçacığı

```
import 'package:flutter/src/widgets/container.dart';
import 'package:flutter application 1/login.dart';
import 'package:cloud firestore/cloud firestore.dart';
class SignUp extends StatefulWidget {
 const SignUp({super.key});
 @override
 State<SignUp> createState() => SignUpState();
class _SignUpState extends State<SignUp> {
 final _sigInFormKey = GlobalKey<FormState>();
final _firestore = FirebaseFirestore.instance;
        firestore = FirebaseFirestore.instance;
 TextEditingController ad = TextEditingController();
 TextEditingController soyad = TextEditingController();
 TextEditingController eposta = TextEditingController();
 TextEditingController telefon = TextEditingController();
 TextEditingController sifre1 = TextEditingController();
 TextEditingController sifre2 = TextEditingController();
 @override
 Widget build(BuildContext context) {
   CollectionReference makeupRef = _firestore.collection('makeup');
     appBar: AppBar(
          title: Text("Sign Up "), Use 'const' with the constructor to imp
          actions: <Widget>[
            Row (
              children: <Widget>[
              Image.asset("images/banner.png"),
      backgroundColor: ■Color.fromARGB(221, 214, 175, 247),
      body: Form(
        child: Padding(
          padding: EdgeInsets.all(20), Use 'const' with the constructor to
```

```
children: <Widget>[
    Image.asset('images/${(items[index].img_banner.toString())}');
```

Blog sayfasının kodlarının tamamı:

```
import 'package:flutter/material.dart';
import 'package:flutter application 1/ProductDataModel.dart';
import 'package:flutter/services.dart' as rootBundle; The prefix 'rootBundle
import 'package:flutter html/flutter html.dart';
class ScanPage extends StatefulWidget {
 const ScanPage({super.key});
 @override
 State<ScanPage> createState() => ScanPageState();
class _ScanPageState extends State<ScanPage> {
 @override
 Widget build(BuildContext context) {
   return Scaffold(
            appBar: AppBar(
         title: Text("Scans "),
         actions: <Widget>[
           Row(
             children: <Widget>[
             Image.asset("images/banner.png"),
             ], // <Widget>[]
         ], // <Widget>[]
       ), // AppBar
       body: FutureBuilder(
     future: ReadJsonData(),
     builder: (context, data) {
       if (data.hasError) {
         return Center(child: Text("${data.error}"));
       } else if (data.hasData) {
         var items = data.data as List<ProductDataModel>;
         return ListView.builder(
             itemBuilder: (context, index) {
                 width: MediaQuery.of(context).size.width,
                 child: Card(
                   child: Column(
                    children: <Widget>[
                            Icons.security,
```

```
size: 80,
                        ...), // Icon
                      SizedBox(
                       height: 10,
                     Text(items[index].name.toString(),
                         style: TextStyle(
                             fontWeight: FontWeight.bold, fontSize: 18)), //
                      SizedBox(
                      height: 10,
                      ), // SizedBox
                     Html(data: items[index].mini desc),
                    ], // <Widget>[]
           }); // ListView.builder
      } else {
     return Center(
         child: CircularProgressIndicator(),
Future<List<ProductDataModel>> ReadJsonData() async { The variable name 'Re
  final jsondata = await rootBundle.rootBundle.loadString('images/scan.json');
  final list = json.decode(jsondata) as List<dynamic>;
  return list.map((e) => ProductDataModel.fromJson(e)).toList();
```