**AKADEMİK BİLGİ ERİŞİM AĞI MOBİL**

**BAĞLANTI KODLARI**

**Android seviyesinde build.grdale dosyası**

buildscript {

ext.kotlin\_version = '1.8.10' // Kotlin sürümü

repositories {

google()

mavenCentral()

}

dependencies {

classpath 'com.android.tools.build:gradle:8.3.0' // Android Gradle Plugin sürümü

classpath "org.jetbrains.kotlin:kotlin-gradle-plugin:$kotlin\_version"

// Firebase için gerekli olan Google Services classpath:

classpath 'com.google.gms:google-services:4.3.15' // Firebase hizmetleri

}

}

allprojects {

repositories {

google()

mavenCentral()

}

}

rootProject.buildDir = "../build"

subprojects {

project.buildDir = "${rootProject.buildDir}/${project.name}"

}

subprojects {

project.evaluationDependsOn(":app")

}

tasks.register("clean", Delete) {

delete rootProject.buildDir

}

**Android seviyesinde setting.gradle dosyası**

pluginManagement {

def flutterSdkPath = {

def properties = new Properties()

file("local.properties").withInputStream { properties.load(it) }

def flutterSdkPath = properties.getProperty("flutter.sdk")

assert flutterSdkPath != null, "flutter.sdk not set in local.properties"

return flutterSdkPath

}()

includeBuild("$flutterSdkPath/packages/flutter\_tools/gradle")

repositories {

google()

mavenCentral()

gradlePluginPortal()

}

}

plugins {

id "dev.flutter.flutter-plugin-loader" version "1.0.0"

id "com.android.application" version "8.3.0" apply false // <-- burası güncellendi

id "com.google.gms.google-services" version "4.3.15" apply false

id "org.jetbrains.kotlin.android" version "1.8.10" apply false // <-- burası da güncellendi

}

include ":app"

**Android --> App seviyesinde build.gradle kodları**

plugins {

id "com.android.application"

id "kotlin-android"

id "dev.flutter.flutter-gradle-plugin"

}

def localProperties = new Properties()

def localPropertiesFile = rootProject.file("local.properties")

if (localPropertiesFile.exists()) {

localPropertiesFile.withReader("UTF-8") { reader ->

localProperties.load(reader)

}

}

def flutterVersionCode = localProperties.getProperty("flutter.versionCode")

if (flutterVersionCode == null) {

flutterVersionCode = "1"

}

def flutterVersionName = localProperties.getProperty("flutter.versionName")

if (flutterVersionName == null) {

flutterVersionName = "1.0"

}

android {

namespace "com.ornek.akademikbilgierisimagi"

compileSdkVersion 35

compileOptions {

sourceCompatibility = JavaVersion.VERSION\_1\_8

targetCompatibility = JavaVersion.VERSION\_1\_8

}

kotlinOptions {

jvmTarget = "1.8"

}

defaultConfig {

applicationId "com.ornek.akademikbilgierisimagi"

minSdkVersion 23

targetSdkVersion 35

versionCode 1

versionName "1.0.0"

}

buildTypes {

release {

signingConfig signingConfigs.debug

}

}

}

flutter {

source = "../.."

}

apply plugin: 'com.google.gms.google-services'

**google-services.json dosyası**

{

"project\_info": {

"project\_number": "510498498090",

"firebase\_url": "https://akademikbilgierisimagi-default-rtdb.europe-west1.firebasedatabase.app",

"project\_id": "akademikbilgierisimagi",

"storage\_bucket": "akademikbilgierisimagi.firebasestorage.app"

},

"client": [

{

"client\_info": {

"mobilesdk\_app\_id": "1:510498498090:android:1bb95ae2b3b560b2150553",

"android\_client\_info": {

"package\_name": "com.example.akademikbilgierisimagi"

}

},

"oauth\_client": [

{

"client\_id": "510498498090-f70aun4vcqkmnd5t5jldgeg1eivf0e9u.apps.googleusercontent.com",

"client\_type": 3

}

],

"api\_key": [

{

"current\_key": "AIzaSyDiBq-S020F1vc2w69YrMyOE6w2sKO2idI"

}

],

"services": {

"appinvite\_service": {

"other\_platform\_oauth\_client": [

{

"client\_id": "510498498090-f70aun4vcqkmnd5t5jldgeg1eivf0e9u.apps.googleusercontent.com",

"client\_type": 3

},

{

"client\_id": "510498498090-ulp073ngo4g4sp1emq7vt9u1ogi47rmt.apps.googleusercontent.com",

"client\_type": 2,

"ios\_info": {

"bundle\_id": "com.example.akademikbilgierisimagi"

}

}

]

}

}

},

{

"client\_info": {

"mobilesdk\_app\_id": "1:510498498090:android:c3cbd60214cd37a5150553",

"android\_client\_info": {

"package\_name": "com.ornek.akademikbilgierisimagi"

}

},

"oauth\_client": [

{

"client\_id": "510498498090-f70aun4vcqkmnd5t5jldgeg1eivf0e9u.apps.googleusercontent.com",

"client\_type": 3

}

],

"api\_key": [

{

"current\_key": "AIzaSyDiBq-S020F1vc2w69YrMyOE6w2sKO2idI"

}

],

"services": {

"appinvite\_service": {

"other\_platform\_oauth\_client": [

{

"client\_id": "510498498090-f70aun4vcqkmnd5t5jldgeg1eivf0e9u.apps.googleusercontent.com",

"client\_type": 3

},

{

"client\_id": "510498498090-ulp073ngo4g4sp1emq7vt9u1ogi47rmt.apps.googleusercontent.com",

"client\_type": 2,

"ios\_info": {

"bundle\_id": "com.example.akademikbilgierisimagi"

}

}

]

}

}

}

],

"configuration\_version": "1"

}

**akademikbilgierisimagi-7936dfc97573.json kodları**

{

"type": "service\_account",

"project\_id": "akademikbilgierisimagi",

"private\_key\_id": "ab62f142d28175c84a50c6fe1be7ae6569ab94a6",

"private\_key": "-----BEGIN PRIVATE KEY-----\nMIIEvAIBADANBgkqhkiG9w0BAQEFAASCBKYwggSiAgEAAoIBAQCnibaHdtrNR0zE\nIiJgBf1WmE6hl1V5fVYg/1TieFKccRuviscMtgqBainFsCE9j+45kWszvlEOQUKb\n9gKsP1u7xiS8yQ01CL9zNZ5dLZYjhhoYKQEWc6CVOUKxqDllGQmT7QItktKFsUtk\nzHarqWGHu+pfwY5fQeKuWX23RTeR0RUAlj/3Mi5un/QU8bIyr57ABHQFCFcvp/L+\nCF9tD0qnpZxeVhIcdXsmfmRxPefKVnegFwTwSihGbnPS9g6HEHXyLch+u/rtPE2y\nYD7JPDLJwzjjedsUsBuiiZV0gvZdYlr94ZSocUSIsaIKQ2Cn9WBU3Q1moWVDc3p7\nMQ8kOnuFAgMBAAECggEAOLdMDWvVAm1CtZsAnBUrACerZtS+YNRnVeBaMhidjsxT\nnQeT/t+ZitoPrhbECf0xnld2PRwOgI7jJ70OhpUkTtrzEN3XFL8TYMMkC1MlYn1B\nVDB9Z8ZY9GtgAph6ZZkmPuebNyZfEldEuKjm+5HprHOXEjeKJQE+5drxzhsmhrv+\n7vEHYJ9jiyhWisI9oUJSEa73NRN2gwEPejzwiBeVqzsNohK+8HAlrvO+k1fu8HhG\nw62XpOvrud68QeT8wLowqtTlMFjprFx0RCvQtXuIst3gMNWYdO6yeWLHs+O/Yvcg\nUqOuNvUxrxURKQ+1kHQXt37+IpFEUd2QVKMvnLqZOQKBgQDR5rR+R3sugAPtVIJM\nSiQM46+h7kKsH507tFzRJtZ+mEa+qFkxkmMrUVwZ/HrRy6zHdAtttKgb6PfW64le\nZeR9cWxCUwq+waa4WjBrwHwGWnQNPEVvylofO/nk+UCpRNKSrr7WJR6stq1ZuCLi\nH27T0NNqzhKoTIhmFNFWUh4NYwKBgQDMVTcAalWt9D3lcCObUNVhNoeglkSrSb2n\nfNn6oofu2gelUbv4nmPvm/ov+TjBYziX9tL9OuZIBeyzGO9MwroulcjiuHzZPrDs\nURneDOD2+3/IaURT6Bqu7ZjWsLaNLJw02Iow+Hiv0WKNQI4JV+HF0n5L97am9V+Y\np8gNkA979wKBgFP9bNva3n+M2gVte3XrdOFzeiPhouwc+l2ZnkpSsCGX7Tb9RjN/\n0A7VT3VWgg7Q4SIuXSRY1avVXNazCRSs0PEhIaCoefQ6qOu7qwRy0S35o9Zn96Dz\n53v4E9Z7plQVJe1r3xDxaYTS1CrCc7LnwQEg3nguXWjatvmh+uH/mebhAoGAfb3q\n7PyC4VuuQ2Uw5EFoLjyqVe5jL61fYpR4Z75hPGE710ZnEGYYRbmMyrzpakiu6QVh\nbagzUhsjQEuV53zlqglO6AZz/hSQN7zo2DbAsfyu9BujiqcI1KEuLt/sVxAeyH6A\n/AXL7xZBOX+CaMmGDICua/1yj2fsfjqx53bBsJ0CgYBDO/XvSOaey8eBueDZFvBA\nL+HMW/pkwSwKsXYjbOVZvq0oX1lDIzW/jOd69xoiXm9qb+o3Y7pzX47kHlJVdrH4\nEGbrQSzGNoHY+Sm3WKPDdsOk2jBPPSDkDXC+EDFLpAu28z1oYqH+MMFcue0rmGrr\ndOwCpag3KI56HRioEt8D1g==\n-----END PRIVATE KEY-----\n",

"client\_email": "firebase-adminsdk-fbsvc@akademikbilgierisimagi.iam.gserviceaccount.com",

"client\_id": "105945254914350576748",

"auth\_uri": "https://accounts.google.com/o/oauth2/auth",

"token\_uri": "https://oauth2.googleapis.com/token",

"auth\_provider\_x509\_cert\_url": "https://www.googleapis.com/oauth2/v1/certs",

"client\_x509\_cert\_url": "https://www.googleapis.com/robot/v1/metadata/x509/firebase-adminsdk-fbsvc%40akademikbilgierisimagi.iam.gserviceaccount.com",

"universe\_domain": "googleapis.com"

}

**akademikbilgiersismagi.iml Dosyası**

<?xml version="1.0" encoding="UTF-8"?>

<module type="JAVA\_MODULE" version="4">

<component name="NewModuleRootManager" inherit-compiler-output="true">

<exclude-output />

<content url="file://$MODULE\_DIR$">

<sourceFolder url="file://$MODULE\_DIR$/lib" isTestSource="false" />

<sourceFolder url="file://$MODULE\_DIR$/test" isTestSource="true" />

<excludeFolder url="file://$MODULE\_DIR$/.dart\_tool" />

<excludeFolder url="file://$MODULE\_DIR$/.idea" />

<excludeFolder url="file://$MODULE\_DIR$/build" />

</content>

<orderEntry type="sourceFolder" forTests="false" />

<orderEntry type="library" name="Dart SDK" level="project" />

<orderEntry type="library" name="Flutter Plugins" level="project" />

<orderEntry type="library" name="Dart Packages" level="project" />

</component>

</module>

**pubspec.yaml Dosyası**

name: akademikbilgierisimagi

description: "A new Flutter project."

# The following line prevents the package from being accidentally published to

# pub.dev using `flutter pub publish`. This is preferred for private packages.

publish\_to: 'none' # Remove this line if you wish to publish to pub.dev

# The following defines the version and build number for your application.

# A version number is three numbers separated by dots, like 1.2.43

# followed by an optional build number separated by a +.

# Both the version and the builder number may be overridden in flutter

# build by specifying --build-name and --build-number, respectively.

# In Android, build-name is used as versionName while build-number used as versionCode.

# Read more about Android versioning at https://developer.android.com/studio/publish/versioning

# In iOS, build-name is used as CFBundleShortVersionString while build-number is used as CFBundleVersion.

# Read more about iOS versioning at

# https://developer.apple.com/library/archive/documentation/General/Reference/InfoPlistKeyReference/Articles/CoreFoundationKeys.html

# In Windows, build-name is used as the major, minor, and patch parts

# of the product and file versions while build-number is used as the build suffix.

version: 1.0.0+1

environment:

sdk: '>=3.4.4 <4.0.0'

# Dependencies specify other packages that your package needs in order to work.

# To automatically upgrade your package dependencies to the latest versions

# consider running `flutter pub upgrade --major-versions`. Alternatively,

# dependencies can be manually updated by changing the version numbers below to

# the latest version available on pub.dev. To see which dependencies have newer

# versions available, run `flutter pub outdated`.

dependencies:

firebase\_core: ^3.13.1

firebase\_auth: ^5.5.4

firebase\_database: ^11.3.6

flutter:

sdk: flutter

# The following adds the Cupertino Icons font to your application.

# Use with the CupertinoIcons class for iOS style icons.

cupertino\_icons: ^1.0.6

dev\_dependencies:

firebase\_core: ^3.13.1

firebase\_database: ^11.3.6

flutter\_test:

sdk: flutter

# The "flutter\_lints" package below contains a set of recommended lints to

# encourage good coding practices. The lint set provided by the package is

# activated in the `analysis\_options.yaml` file located at the root of your

# package. See that file for information about deactivating specific lint

# rules and activating additional ones.

flutter\_lints: ^3.0.0

# For information on the generic Dart part of this file, see the

# following page: https://dart.dev/tools/pub/pubspec

# The following section is specific to Flutter packages.

flutter:

# The following line ensures that the Material Icons font is

# included with your application, so that you can use the icons in

# the material Icons class.

uses-material-design: true

# To add assets to your application, add an assets section, like this:

# assets:

# - images/a\_dot\_burr.jpeg

# - images/a\_dot\_ham.jpeg

# An image asset can refer to one or more resolution-specific "variants", see

# https://flutter.dev/assets-and-images/#resolution-aware

# For details regarding adding assets from package dependencies, see

# https://flutter.dev/assets-and-images/#from-packages

# To add custom fonts to your application, add a fonts section here,

# in this "flutter" section. Each entry in this list should have a

# "family" key with the font family name, and a "fonts" key with a

# list giving the asset and other descriptors for the font. For

# example:

# fonts:

# - family: Schyler

# fonts:

# - asset: fonts/Schyler-Regular.ttf

# - asset: fonts/Schyler-Italic.ttf

# style: italic

# - family: Trajan Pro

# fonts:

# - asset: fonts/TrajanPro.ttf

# - asset: fonts/TrajanPro\_Bold.ttf

# weight: 700

#

# For details regarding fonts from package dependencies,

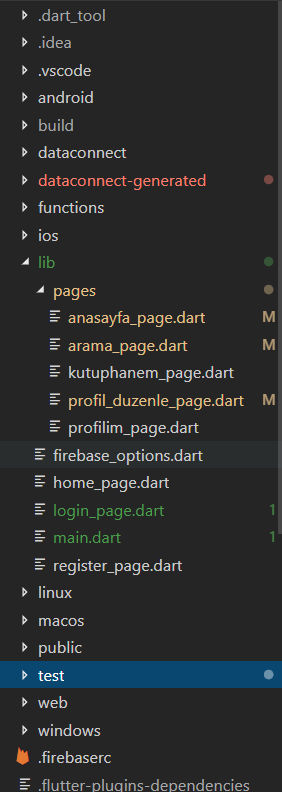
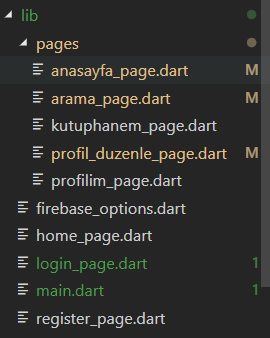
# see https://flutter.dev/custom-fonts/#from-packages

**DOSYA MİMARİ YAPISI AŞAĞIDAKİ GİBİDİR**

**VE GENEL MİMARİ OLARAK CLEAN ARCHİTECTURE MİMARİSİ İLE**

**GELİŞTİRİLMEYE ÇALIŞILDI VE DOSYALAMA SİSTEMİ BU ŞEKİLDE**

**DÜZENLENDİ…**

****

**TASARIM , GERÇEKLEŞTİRME VE RTDB BAĞLANTI KODLARI**

**main.dart kodları**

import 'package:flutter/material.dart';

import 'package:firebase\_core/firebase\_core.dart';

import 'package:firebase\_auth/firebase\_auth.dart';

import 'login\_page.dart';

import 'register\_page.dart'; // bunu oluşturduysan ekle

import 'home\_page.dart'; // giriş sonrası yönlenecek sayfa

import 'firebase\_options.dart'; // Firebase yapılandırma dosyası

void main() async {

WidgetsFlutterBinding.ensureInitialized();

await Firebase.initializeApp(

options: DefaultFirebaseOptions.currentPlatform,

);

runApp(const MyApp());

}

class MyApp extends StatelessWidget {

const MyApp({super.key});

@override

Widget build(BuildContext context) {

return MaterialApp(

title: 'Akademik Bilgi Ağı',

debugShowCheckedModeBanner: false,

theme: ThemeData(primarySwatch: Colors.blue),

home: AuthGate(), // Giriş yaptı mı kontrol eden widget

);

}

}

// Kullanıcı giriş yaptıysa ana sayfa, yapmadıysa giriş ekranı

class AuthGate extends StatelessWidget {

@override

Widget build(BuildContext context) {

return StreamBuilder<User?>(

stream: FirebaseAuth.instance.authStateChanges(),

builder: (context, snapshot) {

if (snapshot.connectionState == ConnectionState.waiting) {

return const Scaffold(

body: Center(child: CircularProgressIndicator()));

} else if (snapshot.hasData) {

return const HomePage(); // Giriş yapmış kullanıcı için ana sayfa

} else {

return const LoginPage(); // Giriş yapmamışsa giriş ekranı

}

},

);

}

}

**login\_page.dart kodları**

import 'package:flutter/material.dart';

import 'package:firebase\_core/firebase\_core.dart';

import 'package:firebase\_auth/firebase\_auth.dart';

import 'login\_page.dart';

import 'register\_page.dart'; // bunu oluşturduysan ekle

import 'home\_page.dart'; // giriş sonrası yönlenecek sayfa

import 'firebase\_options.dart'; // Firebase yapılandırma dosyası

void main() async {

WidgetsFlutterBinding.ensureInitialized();

await Firebase.initializeApp(

options: DefaultFirebaseOptions.currentPlatform,

);

runApp(const MyApp());

}

class MyApp extends StatelessWidget {

const MyApp({super.key});

@override

Widget build(BuildContext context) {

return MaterialApp(

title: 'Akademik Bilgi Ağı',

debugShowCheckedModeBanner: false,

theme: ThemeData(primarySwatch: Colors.blue),

home: AuthGate(), // Giriş yaptı mı kontrol eden widget

);

}

}

// Kullanıcı giriş yaptıysa ana sayfa, yapmadıysa giriş ekranı

class AuthGate extends StatelessWidget {

@override

Widget build(BuildContext context) {

return StreamBuilder<User?>(

stream: FirebaseAuth.instance.authStateChanges(),

builder: (context, snapshot) {

if (snapshot.connectionState == ConnectionState.waiting) {

return const Scaffold(

body: Center(child: CircularProgressIndicator()));

} else if (snapshot.hasData) {

return const HomePage(); // Giriş yapmış kullanıcı için ana sayfa

} else {

return const LoginPage(); // Giriş yapmamışsa giriş ekranı

}

},

);

}

}

**register\_page.dart kodları**

import 'package:flutter/material.dart';

import 'package:firebase\_auth/firebase\_auth.dart';

import 'package:firebase\_database/firebase\_database.dart';

import 'home\_page.dart';

class RegisterPage extends StatefulWidget {

const RegisterPage({Key? key}) : super(key: key);

@override

State<RegisterPage> createState() => \_RegisterPageState();

}

class \_RegisterPageState extends State<RegisterPage> {

final \_formKey = GlobalKey<FormState>();

final nameController = TextEditingController();

final emailController = TextEditingController();

final passwordController = TextEditingController();

final confirmPasswordController = TextEditingController();

String? selectedUniversityId;

String? selectedDepartmentId;

Map<String, String> universityMap = {}; // { "100": "İstanbul Üniversitesi" }

Map<String, String> departmentMap =

{}; // { "400": "Bilgisayar Mühendisliği" }

String? errorText;

@override

void initState() {

super.initState();

fetchUniversities();

}

Future<void> fetchUniversities() async {

try {

final snapshot =

await FirebaseDatabase.instance.ref('universiteler').get();

if (snapshot.exists) {

final data = snapshot.value;

if (data is Map) {

Map<String, String> fetchedUniversities = {};

data.forEach((key, val) {

if (val is Map && val['name'] != null) {

fetchedUniversities[key] = val['name'].toString();

}

});

setState(() {

universityMap = fetchedUniversities;

});

}

}

} catch (e) {

debugPrint('fetchUniversities hatası: $e');

}

}

Future<void> fetchDepartments(String universityId) async {

try {

final snapshot = await FirebaseDatabase.instance

.ref('universiteler/$universityId/bolumler')

.get();

if (snapshot.exists) {

final data = snapshot.value;

if (data is Map) {

Map<String, String> fetchedDepartments = {};

data.forEach((key, val) {

if (val is Map && val['name'] != null) {

fetchedDepartments[key] = val['name'].toString();

}

});

setState(() {

departmentMap = fetchedDepartments;

selectedDepartmentId = null;

});

} else {

setState(() {

departmentMap = {};

selectedDepartmentId = null;

});

}

} else {

setState(() {

departmentMap = {};

selectedDepartmentId = null;

});

}

} catch (e) {

debugPrint('fetchDepartments hatası: $e');

}

}

Future<void> registerUser() async {

setState(() => errorText = null);

if (selectedUniversityId == null || selectedDepartmentId == null) {

setState(() {

errorText = "Lütfen üniversite ve bölüm seçiniz.";

});

return;

}

if (passwordController.text != confirmPasswordController.text) {

setState(() {

errorText = "Şifreler eşleşmiyor.";

});

return;

}

try {

UserCredential userCredential =

await FirebaseAuth.instance.createUserWithEmailAndPassword(

email: emailController.text.trim(),

password: passwordController.text.trim(),

);

final uid = userCredential.user!.uid;

await FirebaseDatabase.instance.ref('kullanicilar/$uid').set({

'isim': nameController.text.trim(),

'email': emailController.text.trim(),

'universite': selectedUniversityId,

'bolum': selectedDepartmentId,

'avatarUrl': '',

'bio': '',

'ayarlar': {

'bildirimlerAcikMi': true,

'dil': 'tr',

'karanlikMod': false,

}

});

if (!mounted) return;

ScaffoldMessenger.of(context).showSnackBar(

const SnackBar(content: Text("Kayıt başarılı")),

);

Navigator.pushReplacement(

context,

MaterialPageRoute(builder: (\_) => const HomePage()),

);

} on FirebaseAuthException catch (e) {

setState(() => errorText = e.message);

}

}

@override

Widget build(BuildContext context) {

return Scaffold(

backgroundColor: Colors.grey[200],

body: Center(

child: SingleChildScrollView(

child: Container(

width: 400,

padding: const EdgeInsets.all(24.0),

decoration: BoxDecoration(

borderRadius: BorderRadius.circular(10),

gradient: const LinearGradient(

colors: [Colors.white, Color(0xFFE0EAFD)],

begin: Alignment.topCenter,

end: Alignment.bottomCenter,

),

),

child: Form(

key: \_formKey,

child: Column(

children: [

const Text('Yeni Hesap Oluştur',

style:

TextStyle(fontSize: 22, fontWeight: FontWeight.bold)),

const SizedBox(height: 20),

TextFormField(

controller: nameController,

decoration: const InputDecoration(labelText: 'Ad Soyad'),

validator: (value) {

if (value == null || value.isEmpty) {

return 'Ad Soyad giriniz';

}

return null;

},

),

const SizedBox(height: 10),

TextFormField(

controller: emailController,

decoration: const InputDecoration(labelText: 'E-posta'),

keyboardType: TextInputType.emailAddress,

validator: (value) {

if (value == null || value.isEmpty) {

return 'E-posta giriniz';

}

if (!RegExp(r'^[^@]+@[^@]+\.[^@]+').hasMatch(value)) {

return 'Geçerli bir e-posta giriniz';

}

return null;

},

),

const SizedBox(height: 10),

DropdownButtonFormField<String>(

isExpanded: true,

decoration: const InputDecoration(labelText: 'Üniversite'),

value: selectedUniversityId,

onChanged: (value) {

setState(() {

selectedUniversityId = value;

departmentMap.clear();

selectedDepartmentId = null;

});

if (value != null) fetchDepartments(value);

},

items: universityMap.entries

.map((e) => DropdownMenuItem(

value: e.key, // üniversiteId

child: Text(e.value), // üniversiteAdı

))

.toList(),

validator: (value) =>

value == null ? 'Üniversite seçiniz' : null,

),

const SizedBox(height: 10),

DropdownButtonFormField<String>(

isExpanded: true,

decoration: const InputDecoration(labelText: 'Bölüm'),

value: selectedDepartmentId,

onChanged: (value) =>

setState(() => selectedDepartmentId = value),

items: departmentMap.entries

.map((e) => DropdownMenuItem(

value: e.key, // bolumId

child: Text(e.value), // bolumAdı

))

.toList(),

validator: (value) =>

value == null ? 'Bölüm seçiniz' : null,

),

const SizedBox(height: 10),

TextFormField(

controller: passwordController,

obscureText: true,

decoration: const InputDecoration(labelText: 'Şifre'),

validator: (value) {

if (value == null || value.isEmpty) {

return 'Şifre giriniz';

}

if (value.length < 6) {

return 'Şifre en az 6 karakter olmalı';

}

return null;

},

),

const SizedBox(height: 10),

TextFormField(

controller: confirmPasswordController,

obscureText: true,

decoration:

const InputDecoration(labelText: 'Şifre Tekrar'),

validator: (value) {

if (value == null || value.isEmpty) {

return 'Şifre tekrar giriniz';

}

if (value != passwordController.text) {

return 'Şifreler eşleşmiyor';

}

return null;

},

),

const SizedBox(height: 10),

if (errorText != null)

Text(errorText!, style: const TextStyle(color: Colors.red)),

const SizedBox(height: 20),

SizedBox(

width: double.infinity,

child: ElevatedButton(

onPressed: () {

if (\_formKey.currentState!.validate()) {

registerUser();

}

},

style: ElevatedButton.styleFrom(

padding: const EdgeInsets.symmetric(vertical: 16),

backgroundColor: Colors.blue[600],

),

child: const Text('Kayıt Ol'),

),

),

const SizedBox(height: 10),

TextButton(

onPressed: () {

Navigator.pop(context); // Geri dön

},

child: const Text("Zaten hesabın var mı? Giriş Yap"),

),

],

),

),

),

),

),

);

}

}

**home\_page.dart kodları**

import 'package:flutter/material.dart';

import 'pages/anasayfa\_page.dart';

import 'pages/profilim\_page.dart';

import 'pages/kutuphanem\_page.dart';

import 'pages/arama\_page.dart';

class HomePage extends StatefulWidget {

const HomePage({super.key});

@override

State<HomePage> createState() => \_HomePageState();

}

class \_HomePageState extends State<HomePage> {

int \_selectedIndex = 0;

final List<Widget> \_pages = [

const AnasayfaPage(),

const ProfilimPage(),

const KutuphanemPage(),

const AramaPage(),

];

void \_onItemTapped(int index) {

setState(() {

\_selectedIndex = index;

});

}

@override

Widget build(BuildContext context) {

return Scaffold(

body: \_pages[\_selectedIndex],

bottomNavigationBar: BottomNavigationBar(

type: BottomNavigationBarType.fixed,

currentIndex: \_selectedIndex,

onTap: \_onItemTapped,

selectedItemColor: Colors.blue,

unselectedItemColor: Colors.grey,

items: const [

BottomNavigationBarItem(

icon: Icon(Icons.home),

label: 'Anasayfa',

),

BottomNavigationBarItem(

icon: Icon(Icons.person),

label: 'Profilim',

),

BottomNavigationBarItem(

icon: Icon(Icons.library\_books),

label: 'Kütüphanem',

),

BottomNavigationBarItem(

icon: Icon(Icons.search),

label: 'Arama',

),

],

),

);

}

}

**firebase\_options.dart kodları**

// File generated by FlutterFire CLI.

// ignore\_for\_file: type=lint

import 'package:firebase\_core/firebase\_core.dart' show FirebaseOptions;

import 'package:flutter/foundation.dart'

show defaultTargetPlatform, kIsWeb, TargetPlatform;

/// Default [FirebaseOptions] for use with your Firebase apps.

///

/// Example:

/// ```dart

/// import 'firebase\_options.dart';

/// // ...

/// await Firebase.initializeApp(

/// options: DefaultFirebaseOptions.currentPlatform,

/// );

/// ```

class DefaultFirebaseOptions {

static FirebaseOptions get currentPlatform {

if (kIsWeb) {

return web;

}

switch (defaultTargetPlatform) {

case TargetPlatform.android:

return android;

case TargetPlatform.iOS:

return ios;

case TargetPlatform.macOS:

return macos;

case TargetPlatform.windows:

return windows;

case TargetPlatform.linux:

throw UnsupportedError(

'DefaultFirebaseOptions have not been configured for linux - '

'you can reconfigure this by running the FlutterFire CLI again.',

);

default:

throw UnsupportedError(

'DefaultFirebaseOptions are not supported for this platform.',

);

}

}

static const FirebaseOptions web = FirebaseOptions(

apiKey: 'AIzaSyBVxbBTWr1z7znsbqTtK-v67uPVdSbvFNI',

appId: '1:510498498090:web:dfeef8efc5ed0e07150553',

messagingSenderId: '510498498090',

projectId: 'akademikbilgierisimagi',

authDomain: 'akademikbilgierisimagi.firebaseapp.com',

databaseURL: 'https://akademikbilgierisimagi-default-rtdb.europe-west1.firebasedatabase.app',

storageBucket: 'akademikbilgierisimagi.firebasestorage.app',

measurementId: 'G-Z0EVJGSE02',

);

static const FirebaseOptions android = FirebaseOptions(

apiKey: 'AIzaSyDiBq-S020F1vc2w69YrMyOE6w2sKO2idI',

appId: '1:510498498090:android:c3cbd60214cd37a5150553',

messagingSenderId: '510498498090',

projectId: 'akademikbilgierisimagi',

databaseURL: 'https://akademikbilgierisimagi-default-rtdb.europe-west1.firebasedatabase.app',

storageBucket: 'akademikbilgierisimagi.firebasestorage.app',

);

static const FirebaseOptions ios = FirebaseOptions(

apiKey: 'AIzaSyD1QWuZMUKSwZKHl2RBiTBPyfuIlS5OkEo',

appId: '1:510498498090:ios:baa74a889d031873150553',

messagingSenderId: '510498498090',

projectId: 'akademikbilgierisimagi',

databaseURL: 'https://akademikbilgierisimagi-default-rtdb.europe-west1.firebasedatabase.app',

storageBucket: 'akademikbilgierisimagi.firebasestorage.app',

iosClientId: '510498498090-ulp073ngo4g4sp1emq7vt9u1ogi47rmt.apps.googleusercontent.com',

iosBundleId: 'com.example.akademikbilgierisimagi',

);

static const FirebaseOptions macos = FirebaseOptions(

apiKey: 'AIzaSyD1QWuZMUKSwZKHl2RBiTBPyfuIlS5OkEo',

appId: '1:510498498090:ios:baa74a889d031873150553',

messagingSenderId: '510498498090',

projectId: 'akademikbilgierisimagi',

databaseURL: 'https://akademikbilgierisimagi-default-rtdb.europe-west1.firebasedatabase.app',

storageBucket: 'akademikbilgierisimagi.firebasestorage.app',

iosClientId: '510498498090-ulp073ngo4g4sp1emq7vt9u1ogi47rmt.apps.googleusercontent.com',

iosBundleId: 'com.example.akademikbilgierisimagi',

);

static const FirebaseOptions windows = FirebaseOptions(

apiKey: 'AIzaSyBVxbBTWr1z7znsbqTtK-v67uPVdSbvFNI',

appId: '1:510498498090:web:d7bc0df7f77dbf05150553',

messagingSenderId: '510498498090',

projectId: 'akademikbilgierisimagi',

authDomain: 'akademikbilgierisimagi.firebaseapp.com',

databaseURL: 'https://akademikbilgierisimagi-default-rtdb.europe-west1.firebasedatabase.app',

storageBucket: 'akademikbilgierisimagi.firebasestorage.app',

measurementId: 'G-VCH8BYZ0SW',

);

}

**Pages 🡪 anasayfa\_page.dart kodları**

import 'package:flutter/material.dart';

import 'package:firebase\_auth/firebase\_auth.dart';

import 'package:firebase\_database/firebase\_database.dart';

class AnasayfaPage extends StatefulWidget {

const AnasayfaPage({super.key});

@override

State<AnasayfaPage> createState() => \_AnasayfaPageState();

}

class \_AnasayfaPageState extends State<AnasayfaPage> {

final user = FirebaseAuth.instance.currentUser;

late DatabaseReference kullaniciRef;

late DatabaseReference takipRef;

List<Map<String, dynamic>> takipEdilenKullanicilar = [];

bool loading = true;

@override

void initState() {

super.initState();

kullaniciRef =

FirebaseDatabase.instance.ref().child('kullanicilar/${user!.uid}');

takipRef = kullaniciRef.child('takip');

\_loadTakipEdilenler();

}

Future<void> \_loadTakipEdilenler() async {

setState(() {

loading = true;

});

final takipSnapshot = await takipRef.get();

if (!takipSnapshot.exists) {

setState(() {

takipEdilenKullanicilar = [];

loading = false;

});

return;

}

final Map<dynamic, dynamic> takipData =

takipSnapshot.value as Map<dynamic, dynamic>;

List<String> takipEdilenIds = takipData.keys.cast<String>().toList();

List<Map<String, dynamic>> loadedKullanicilar = [];

for (String takipId in takipEdilenIds) {

final kullaniciSnapshot =

await FirebaseDatabase.instance.ref('kullanicilar/$takipId').get();

if (kullaniciSnapshot.exists) {

Map<String, dynamic> kullaniciMap =

Map<String, dynamic>.from(kullaniciSnapshot.value as Map);

// Kütüphaneler

final kutuphaneSnapshot = await FirebaseDatabase.instance

.ref('kullanicilar/$takipId/kutuphaneler')

.get();

List<Map<String, dynamic>> kutuphaneler = [];

if (kutuphaneSnapshot.exists) {

Map<dynamic, dynamic> kutuphaneData =

kutuphaneSnapshot.value as Map<dynamic, dynamic>;

kutuphaneData.forEach((key, value) {

Map<String, dynamic> k = Map<String, dynamic>.from(value);

k['id'] = key;

kutuphaneler.add(k);

});

}

// Gönderiler

final gonderiSnapshot = await FirebaseDatabase.instance

.ref('kullanici\_paylasimlari/$takipId')

.get();

List<Map<String, dynamic>> gonderiler = [];

if (gonderiSnapshot.exists) {

Map<dynamic, dynamic> gonderiMap =

gonderiSnapshot.value as Map<dynamic, dynamic>;

for (var postId in gonderiMap.keys) {

final postSnapshot = await FirebaseDatabase.instance

.ref('paylasimlar/$postId')

.get();

if (postSnapshot.exists) {

Map<String, dynamic> post =

Map<String, dynamic>.from(postSnapshot.value as Map);

post['id'] = postId;

gonderiler.add(post);

}

}

}

kullaniciMap['kutuphaneler'] = kutuphaneler;

kullaniciMap['gonderiler'] = gonderiler;

loadedKullanicilar.add(kullaniciMap);

}

}

setState(() {

takipEdilenKullanicilar = loadedKullanicilar;

loading = false;

});

}

Future<void> \_toggleBegeni(String postId, bool begendiMi) async {

final currentUserId = user!.uid;

final begeniRef = FirebaseDatabase.instance.ref('begeni/$postId');

if (begendiMi) {

await begeniRef.child(currentUserId).remove();

} else {

await begeniRef.child(currentUserId).set(true);

}

setState(() {});

}

Widget \_kutuphaneCard(Map<String, dynamic> kutuphane) {

return Card(

margin: const EdgeInsets.symmetric(vertical: 4, horizontal: 8),

child: ListTile(

leading: const Icon(Icons.library\_books, color: Colors.blue),

title: Text(kutuphane['isim'] ?? 'İsimsiz Kütüphane'),

subtitle: Text('Dosya sayısı: ${kutuphane['dosyalar']?.length ?? 0}'),

),

);

}

Widget \_gonderiCard(Map<String, dynamic> gonderi) {

final currentUserId = user!.uid;

final postId = gonderi['id'];

return FutureBuilder<DataSnapshot>(

future: FirebaseDatabase.instance.ref('begeni/$postId').get(),

builder: (context, snapshot) {

bool begendiMi = false;

int begeniSayisi = 0;

if (snapshot.hasData && snapshot.data!.exists) {

Map<dynamic, dynamic> begeniMap =

snapshot.data!.value as Map<dynamic, dynamic>;

begeniSayisi = begeniMap.length;

begendiMi = begeniMap.containsKey(currentUserId);

}

return Card(

margin: const EdgeInsets.symmetric(vertical: 4, horizontal: 8),

child: ListTile(

leading: const Icon(Icons.post\_add, color: Colors.green),

title: Text(gonderi['content'] ?? 'Başlıksız Gönderi'),

subtitle: Column(

crossAxisAlignment: CrossAxisAlignment.start,

children: [

const SizedBox(height: 6),

Row(

children: [

IconButton(

icon: Icon(

begendiMi ? Icons.favorite : Icons.favorite\_border,

color: begendiMi ? Colors.red : Colors.grey,

),

onPressed: () => \_toggleBegeni(postId, begendiMi),

),

Text('$begeniSayisi beğeni'),

const Spacer(),

TextButton.icon(

onPressed: () {

ScaffoldMessenger.of(context).showSnackBar(

const SnackBar(

content: Text("Yorum özelliği yakında")),

);

},

icon: const Icon(Icons.comment, size: 18),

label: const Text("Yorum yap"),

),

],

),

],

),

),

);

},

);

}

@override

Widget build(BuildContext context) {

if (loading) {

return const Center(child: CircularProgressIndicator());

}

if (takipEdilenKullanicilar.isEmpty) {

return const Center(

child: Padding(

padding: EdgeInsets.all(16),

child: Text(

'Henüz kimseyi takip etmiyorsunuz.\nTakip ettiklerinizin gönderi ve kütüphaneleri burada görünecek.',

textAlign: TextAlign.center,

style: TextStyle(fontSize: 16),

),

),

);

}

return RefreshIndicator(

onRefresh: \_loadTakipEdilenler,

child: ListView.builder(

padding: const EdgeInsets.all(8),

itemCount: takipEdilenKullanicilar.length,

itemBuilder: (context, index) {

final kullanici = takipEdilenKullanicilar[index];

final kutuphaneler = (kullanici['kutuphaneler'] as List?) ?? [];

final gonderiler = (kullanici['gonderiler'] as List?) ?? [];

return Card(

elevation: 3,

margin: const EdgeInsets.symmetric(vertical: 8),

shape:

RoundedRectangleBorder(borderRadius: BorderRadius.circular(12)),

child: Padding(

padding: const EdgeInsets.all(12),

child: Column(

crossAxisAlignment: CrossAxisAlignment.start,

children: [

// Kullanıcı adı ve email

Row(

children: [

const Icon(Icons.person, size: 28, color: Colors.blue),

const SizedBox(width: 8),

Expanded(

child: Text(

kullanici['isim'] ??

kullanici['email'] ??

'Bilinmeyen Kullanıcı',

style: const TextStyle(

fontSize: 18, fontWeight: FontWeight.bold),

),

),

],

),

const SizedBox(height: 8),

// Kütüphaneler başlığı ve listesi

if (kutuphaneler.isNotEmpty) ...[

const Text(

'Kütüphaneler:',

style: TextStyle(

fontSize: 16,

fontWeight: FontWeight.w600,

color: Colors.blueAccent),

),

const SizedBox(height: 4),

...kutuphaneler.map((k) => \_kutuphaneCard(k)).toList(),

const SizedBox(height: 8),

],

// Gönderiler başlığı ve listesi

if (gonderiler.isNotEmpty) ...[

const Text(

'Gönderiler:',

style: TextStyle(

fontSize: 16,

fontWeight: FontWeight.w600,

color: Colors.green),

),

const SizedBox(height: 4),

...gonderiler.map((g) => \_gonderiCard(g)).toList(),

],

// Kütüphane ve gönderi yoksa uyarı

if (kutuphaneler.isEmpty && gonderiler.isEmpty) ...[

const Text('Henüz paylaşım yok.'),

],

],

),

),

);

},

),

);

}

}

**Pages 🡪 arama\_page.dart kodları**

import 'package:flutter/material.dart';

import 'package:firebase\_auth/firebase\_auth.dart';

import 'package:firebase\_database/firebase\_database.dart';

class AramaPage extends StatefulWidget {

const AramaPage({super.key});

@override

State<AramaPage> createState() => \_AramaPageState();

}

class \_AramaPageState extends State<AramaPage> {

final TextEditingController \_searchController = TextEditingController();

List<Map<String, dynamic>> bulunanKullanicilar = [];

Map<String, String> takipDurumlari = {};

bool \_isLoading = false;

final FirebaseAuth \_auth = FirebaseAuth.instance;

Future<void> \_aramaYap(String query) async {

if (query.isEmpty) {

setState(() {

bulunanKullanicilar = [];

takipDurumlari = {};

});

return;

}

setState(() {

\_isLoading = true;

});

final dbRef = FirebaseDatabase.instance.ref();

final kullaniciRef = dbRef.child('kullanicilar');

final takipRef = dbRef.child('kullanici\_takipleri');

final currentUserId = \_auth.currentUser?.uid;

try {

final snapshot = await kullaniciRef.get();

if (snapshot.exists) {

final data = Map<String, dynamic>.from(snapshot.value as Map);

final filtrelenmis = data.entries.where((entry) {

final user = Map<String, dynamic>.from(entry.value);

final isim = (user['isim'] ?? '').toString().toLowerCase();

return isim.contains(query.toLowerCase());

}).map((entry) {

final user = Map<String, dynamic>.from(entry.value);

user['id'] = entry.key;

return user;

}).toList();

Map<String, dynamic> takipEdilenler = {};

if (currentUserId != null) {

final takipSnapshot = await takipRef.child(currentUserId).get();

if (takipSnapshot.exists) {

takipEdilenler =

Map<String, dynamic>.from(takipSnapshot.value as Map);

}

}

Map<String, String> durumlar = {};

for (var kullanici in filtrelenmis) {

final id = kullanici['id'];

if (id == currentUserId) {

durumlar[id] = 'kendisi';

} else if (takipEdilenler.containsKey(id)) {

durumlar[id] = 'takipte';

} else {

durumlar[id] = 'takip\_et';

}

}

setState(() {

bulunanKullanicilar = filtrelenmis;

takipDurumlari = durumlar;

});

}

} catch (e) {

debugPrint("Hata: $e");

}

setState(() {

\_isLoading = false;

});

}

Future<void> \_takipEt(String takipEdilecekId) async {

final currentUserId = \_auth.currentUser?.uid;

if (currentUserId == null) return;

final dbRef = FirebaseDatabase.instance.ref();

try {

// 1. takip\_edilenler/{currentUserId}/{hedefUserId}

await dbRef

.child('takip\_edilenler/$currentUserId/$takipEdilecekId')

.set(true);

// 2. kullanici\_takipciler/{hedefUserId}/{currentUserId}

await dbRef

.child('kullanici\_takipciler/$takipEdilecekId/$currentUserId')

.set(true);

setState(() {

takipDurumlari[takipEdilecekId] = 'takipte';

});

} catch (e) {

debugPrint("Takip etme hatası: $e");

}

}

Future<void> \_takipBirak(String takipEdilenId) async {

final currentUserId = \_auth.currentUser?.uid;

if (currentUserId == null) return;

final dbRef = FirebaseDatabase.instance.ref();

try {

// 1. takip\_edilenler/{currentUserId}/{hedefUserId}

await dbRef

.child('takip\_edilenler/$currentUserId/$takipEdilenId')

.remove();

// 2. kullanici\_takipciler/{hedefUserId}/{currentUserId}

await dbRef

.child('kullanici\_takipciler/$takipEdilenId/$currentUserId')

.remove();

setState(() {

takipDurumlari[takipEdilenId] = 'takip\_et';

});

} catch (e) {

debugPrint("Takipten çıkma hatası: $e");

}

}

void \_profilSayfasinaGit(Map<String, dynamic> kullanici) {

Navigator.push(

context,

MaterialPageRoute(

builder: (context) => KullaniciProfilSayfasi(kullanici: kullanici),

),

);

}

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: const Text("Kullanıcı Ara"),

backgroundColor: Colors.deepPurple,

),

body: Padding(

padding: const EdgeInsets.all(12.0),

child: Column(

children: [

TextField(

controller: \_searchController,

decoration: InputDecoration(

hintText: "İsim ile ara...",

prefixIcon: const Icon(Icons.search),

border: OutlineInputBorder(

borderRadius: BorderRadius.circular(14),

),

),

onChanged: \_aramaYap,

),

const SizedBox(height: 12),

if (\_isLoading) const LinearProgressIndicator(minHeight: 3),

if (!\_isLoading)

Expanded(

child: bulunanKullanicilar.isEmpty

? const Center(child: Text("Sonuç bulunamadı."))

: ListView.builder(

itemCount: bulunanKullanicilar.length,

itemBuilder: (context, index) {

final kullanici = bulunanKullanicilar[index];

final kullaniciId = kullanici['id'];

final durum =

takipDurumlari[kullaniciId] ?? 'takip\_et';

return Card(

shape: RoundedRectangleBorder(

borderRadius: BorderRadius.circular(12),

),

child: ListTile(

leading: CircleAvatar(

backgroundImage: kullanici['avatarUrl'] != null

? NetworkImage(kullanici['avatarUrl'])

: null,

child: kullanici['avatarUrl'] == null

? const Icon(Icons.person)

: null,

),

title: Text(kullanici['isim'] ?? ''),

subtitle: Text(kullanici['email'] ?? ''),

trailing: durum == 'kendisi'

? null

: durum == 'takipte'

? TextButton(

onPressed: () =>

\_takipBirak(kullaniciId),

child: const Text('Takibi Bırak'),

)

: TextButton(

onPressed: () =>

\_takipEt(kullaniciId),

child: const Text('Takip Et'),

),

onTap: () => \_profilSayfasinaGit(kullanici),

),

);

},

),

),

],

),

),

);

}

}

class KullaniciProfilSayfasi extends StatelessWidget {

final Map<String, dynamic> kullanici;

const KullaniciProfilSayfasi({super.key, required this.kullanici});

Future<String> \_getUniversiteAdi(String uniId) async {

final ref = FirebaseDatabase.instance.ref('universiteler/$uniId/name');

final snapshot = await ref.get();

return snapshot.exists ? snapshot.value.toString() : 'Bilinmiyor';

}

Future<String> \_getBolumAdi(String uniId, String bolumId) async {

final ref = FirebaseDatabase.instance

.ref('universiteler/$uniId/bolumler/$bolumId/name');

final snapshot = await ref.get();

return snapshot.exists ? snapshot.value.toString() : 'Bilinmiyor';

}

@override

Widget build(BuildContext context) {

final uniId = kullanici['universiteId'] ?? '';

final bolumId = kullanici['bolumId'] ?? '';

return Scaffold(

appBar: AppBar(

title: Text(kullanici['isim'] ?? 'Profil'),

backgroundColor: Colors.deepPurple,

),

body: Padding(

padding: const EdgeInsets.all(16),

child: Column(

children: [

CircleAvatar(

radius: 60,

backgroundImage: kullanici['avatarUrl'] != null

? NetworkImage(kullanici['avatarUrl'])

: null,

child: kullanici['avatarUrl'] == null

? const Icon(Icons.person, size: 60)

: null,

),

const SizedBox(height: 16),

Text(kullanici['isim'] ?? '', style: const TextStyle(fontSize: 22)),

Text(kullanici['email'] ?? '',

style: const TextStyle(color: Colors.grey)),

const SizedBox(height: 8),

FutureBuilder<String>(

future: \_getUniversiteAdi(uniId),

builder: (context, snapshot) => Text(

"Üniversite: ${snapshot.data ?? ''}",

style: const TextStyle(fontWeight: FontWeight.w600),

),

),

FutureBuilder<String>(

future: \_getBolumAdi(uniId, bolumId),

builder: (context, snapshot) => Text(

"Bölüm: ${snapshot.data ?? ''}",

style: const TextStyle(fontWeight: FontWeight.w600),

),

),

],

),

),

);

}

}

**Pages 🡪 kutuphanem\_page.dart kodları**

import 'package:flutter/material.dart';

import 'package:firebase\_auth/firebase\_auth.dart';

import 'package:firebase\_database/firebase\_database.dart';

class KutuphanemPage extends StatefulWidget {

const KutuphanemPage({super.key});

@override

State<KutuphanemPage> createState() => \_KutuphanemPageState();

}

class \_KutuphanemPageState extends State<KutuphanemPage> {

final user = FirebaseAuth.instance.currentUser;

late DatabaseReference kullaniciRef;

late DatabaseReference kutuphaneRef;

List<Map<String, dynamic>> kutuphaneler = [];

@override

void initState() {

super.initState();

kullaniciRef =

FirebaseDatabase.instance.ref().child('kullanicilar/${user!.uid}');

kutuphaneRef = kullaniciRef.child('kutuphaneler');

\_loadKutuphaneler();

}

Future<void> \_loadKutuphaneler() async {

final snapshot = await kutuphaneRef.get();

if (!snapshot.exists) {

setState(() {

kutuphaneler = [];

});

return;

}

final Map<dynamic, dynamic> data = snapshot.value as Map<dynamic, dynamic>;

final List<Map<String, dynamic>> loaded = [];

data.forEach((key, value) {

final kutuphaneMap = Map<String, dynamic>.from(value);

kutuphaneMap['id'] = key;

loaded.add(kutuphaneMap);

});

setState(() {

kutuphaneler = loaded;

});

}

Future<void> \_yeniKutuphaneOlustur() async {

String yeniAdi = '';

await showDialog(

context: context,

builder: (context) => AlertDialog(

title: const Text('Yeni Kütüphane Oluştur'),

content: TextField(

autofocus: true,

decoration: InputDecoration(

hintText: 'Kütüphane adı',

border: OutlineInputBorder(borderRadius: BorderRadius.circular(10)),

contentPadding:

const EdgeInsets.symmetric(horizontal: 12, vertical: 8),

),

onChanged: (value) => yeniAdi = value,

),

actions: [

TextButton(

onPressed: () => Navigator.pop(context),

child: const Text('İptal'),

),

ElevatedButton(

onPressed: () {

if (yeniAdi.trim().isNotEmpty) {

Navigator.pop(context);

}

},

child: const Text('Oluştur'),

),

],

),

);

if (yeniAdi.trim().isEmpty) return;

final yeniKutuphaneRef = kutuphaneRef.push();

await yeniKutuphaneRef.set({

'isim': yeniAdi.trim(),

'dosyalar': {},

});

\_loadKutuphaneler();

}

void \_goToKutuphaneDetay(Map<String, dynamic> kutuphane) {

Navigator.push(

context,

MaterialPageRoute(

builder: (\_) => KutuphaneDetayPage(

kutuphaneId: kutuphane['id'],

kutuphaneAdi: kutuphane['isim'],

),

),

).then((\_) => \_loadKutuphaneler());

}

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: const Text('Kütüphanem'),

backgroundColor: Colors.blue.shade700,

actions: [

IconButton(

icon: const Icon(Icons.add),

tooltip: 'Yeni Kütüphane Oluştur',

onPressed: \_yeniKutuphaneOlustur,

),

],

),

body: kutuphaneler.isEmpty

? Center(

child: Column(

mainAxisSize: MainAxisSize.min,

children: [

Icon(Icons.library\_books,

size: 80, color: Colors.grey.shade400),

const SizedBox(height: 12),

Text(

'Henüz kütüphane yok.',

style: TextStyle(fontSize: 18, color: Colors.grey.shade600),

),

Text(

'+ butonuna tıklayarak oluşturabilirsiniz.',

style: TextStyle(fontSize: 14, color: Colors.grey.shade500),

),

],

),

)

: Padding(

padding: const EdgeInsets.all(12.0),

child: ListView.builder(

itemCount: kutuphaneler.length,

itemBuilder: (context, index) {

final kutuphane = kutuphaneler[index];

return Card(

elevation: 4,

shadowColor: Colors.blue.shade100,

shape: RoundedRectangleBorder(

borderRadius: BorderRadius.circular(14),

),

margin: const EdgeInsets.symmetric(vertical: 8),

child: ListTile(

contentPadding: const EdgeInsets.symmetric(

horizontal: 20, vertical: 12),

title: Text(

kutuphane['isim'],

style: const TextStyle(

fontWeight: FontWeight.w600, fontSize: 18),

),

trailing: const Icon(Icons.arrow\_forward\_ios, size: 20),

onTap: () => \_goToKutuphaneDetay(kutuphane),

),

);

},

),

),

);

}

}

class KutuphaneDetayPage extends StatefulWidget {

final String kutuphaneId;

final String kutuphaneAdi;

const KutuphaneDetayPage({

super.key,

required this.kutuphaneId,

required this.kutuphaneAdi,

});

@override

State<KutuphaneDetayPage> createState() => \_KutuphaneDetayPageState();

}

class \_KutuphaneDetayPageState extends State<KutuphaneDetayPage> {

final user = FirebaseAuth.instance.currentUser;

late DatabaseReference dosyalarRef;

List<Map<String, dynamic>> dosyalar = [];

@override

void initState() {

super.initState();

dosyalarRef = FirebaseDatabase.instance.ref().child(

'kullanicilar/${user!.uid}/kutuphaneler/${widget.kutuphaneId}/dosyalar');

\_loadDosyalar();

}

Future<void> \_loadDosyalar() async {

final snapshot = await dosyalarRef.get();

if (!snapshot.exists) {

setState(() {

dosyalar = [];

});

return;

}

final Map<dynamic, dynamic> data = snapshot.value as Map<dynamic, dynamic>;

final List<Map<String, dynamic>> loaded = [];

data.forEach((key, value) {

final dosyaMap = Map<String, dynamic>.from(value);

dosyaMap['id'] = key;

loaded.add(dosyaMap);

});

setState(() {

dosyalar = loaded;

});

}

Widget \_dosyaIcon(String tur) {

switch (tur.toLowerCase()) {

case 'pdf':

return Icon(Icons.picture\_as\_pdf, color: Colors.red.shade700, size: 36);

case 'ppt':

case 'pptx':

return Icon(Icons.slideshow, color: Colors.orange.shade700, size: 36);

case 'doc':

case 'docx':

return Icon(Icons.description, color: Colors.blue.shade700, size: 36);

case 'xls':

case 'xlsx':

return Icon(Icons.table\_chart, color: Colors.green.shade700, size: 36);

default:

return Icon(Icons.insert\_drive\_file,

size: 36, color: Colors.grey.shade600);

}

}

Future<void> \_dosyaEkle() async {

String dosyaAdi = '';

String dosyaTuru = '';

String dosyaUrl = '';

await showDialog(

context: context,

builder: (context) => AlertDialog(

title: const Text('Dosya Ekle'),

content: SingleChildScrollView(

child: Column(

children: [

TextField(

decoration: InputDecoration(

hintText: 'Dosya Adı',

border: OutlineInputBorder(

borderRadius: BorderRadius.circular(10)),

contentPadding:

const EdgeInsets.symmetric(horizontal: 12, vertical: 8),

),

onChanged: (val) => dosyaAdi = val,

),

const SizedBox(height: 12),

TextField(

decoration: InputDecoration(

hintText: 'Dosya Türü (pdf, ppt, doc, xls vb.)',

border: OutlineInputBorder(

borderRadius: BorderRadius.circular(10)),

contentPadding:

const EdgeInsets.symmetric(horizontal: 12, vertical: 8),

),

onChanged: (val) => dosyaTuru = val,

),

const SizedBox(height: 12),

TextField(

decoration: InputDecoration(

hintText: 'Dosya URL',

border: OutlineInputBorder(

borderRadius: BorderRadius.circular(10)),

contentPadding:

const EdgeInsets.symmetric(horizontal: 12, vertical: 8),

),

onChanged: (val) => dosyaUrl = val,

),

],

),

),

actions: [

TextButton(

onPressed: () => Navigator.pop(context),

child: const Text('İptal'),

),

ElevatedButton(

onPressed: () {

if (dosyaAdi.trim().isNotEmpty &&

dosyaTuru.trim().isNotEmpty &&

dosyaUrl.trim().isNotEmpty) {

Navigator.pop(context);

}

},

child: const Text('Ekle'),

),

],

),

);

if (dosyaAdi.trim().isEmpty ||

dosyaTuru.trim().isEmpty ||

dosyaUrl.trim().isEmpty) return;

final yeniDosyaRef = dosyalarRef.push();

await yeniDosyaRef.set({

'ad': dosyaAdi.trim(),

'tur': dosyaTuru.trim().toLowerCase(),

'url': dosyaUrl.trim(),

});

\_loadDosyalar();

}

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: Text(widget.kutuphaneAdi),

backgroundColor: Colors.blue.shade700,

actions: [

IconButton(

icon: const Icon(Icons.add),

tooltip: 'Dosya Ekle',

onPressed: \_dosyaEkle,

)

],

),

body: dosyalar.isEmpty

? Center(

child: Column(

mainAxisSize: MainAxisSize.min,

children: [

Icon(Icons.insert\_drive\_file,

size: 80, color: Colors.grey.shade400),

const SizedBox(height: 12),

Text(

'Henüz dosya eklenmemiş.',

style: TextStyle(fontSize: 18, color: Colors.grey.shade600),

),

Text(

'+ butonuna tıklayarak dosya ekleyebilirsiniz.',

style: TextStyle(fontSize: 14, color: Colors.grey.shade500),

),

],

),

)

: Padding(

padding: const EdgeInsets.all(12.0),

child: ListView.builder(

itemCount: dosyalar.length,

itemBuilder: (context, index) {

final dosya = dosyalar[index];

return Card(

elevation: 4,

shadowColor: Colors.blue.shade100,

shape: RoundedRectangleBorder(

borderRadius: BorderRadius.circular(14),

),

margin: const EdgeInsets.symmetric(vertical: 8),

child: ListTile(

contentPadding: const EdgeInsets.symmetric(

horizontal: 20, vertical: 12),

leading: \_dosyaIcon(dosya['tur']),

title: Text(

dosya['ad'],

style: const TextStyle(

fontWeight: FontWeight.w600, fontSize: 16),

),

subtitle: Text(dosya['tur'].toUpperCase(),

style: TextStyle(color: Colors.grey.shade600)),

trailing: IconButton(

icon: const Icon(Icons.open\_in\_new, color: Colors.blue),

tooltip: 'Dosyayı Aç',

onPressed: () {

// İstersen dosya url açılabilir.

// launch(dosya['url']) gibi

},

),

),

);

},

),

),

);

}

}

**Pages 🡪 profılım\_page.dart koldarı**

import 'package:flutter/material.dart';

import 'package:firebase\_auth/firebase\_auth.dart';

import 'package:firebase\_database/firebase\_database.dart';

import 'profil\_duzenle\_page.dart';

import '../login\_page.dart';

// Takipçi Listesi Sayfası

class TakipcilerPage extends StatelessWidget {

final String uid;

const TakipcilerPage({required this.uid, super.key});

@override

Widget build(BuildContext context) {

final takipciRef =

FirebaseDatabase.instance.ref('kullanici\_takipciler/$uid');

return Scaffold(

appBar: AppBar(title: const Text("Takipçiler")),

body: StreamBuilder(

stream: takipciRef.onValue,

builder: (context, snapshot) {

if (!snapshot.hasData || snapshot.data!.snapshot.value == null) {

return const Center(child: Text("Takipçi yok"));

}

final data =

Map<String, dynamic>.from(snapshot.data!.snapshot.value as Map);

final takipciUids = data.keys.toList();

return ListView.builder(

itemCount: takipciUids.length,

itemBuilder: (context, index) {

final takipciUid = takipciUids[index];

return ListTile(

title: Text(takipciUid),

);

},

);

},

),

);

}

}

// Takip Edilenler Listesi Sayfası

class TakipEdilenlerPage extends StatelessWidget {

final String uid;

const TakipEdilenlerPage({required this.uid, super.key});

@override

Widget build(BuildContext context) {

final takipEdilenRef =

FirebaseDatabase.instance.ref('kullanici\_takipler/$uid');

return Scaffold(

appBar: AppBar(title: const Text("Takip Edilenler")),

body: StreamBuilder(

stream: takipEdilenRef.onValue,

builder: (context, snapshot) {

if (!snapshot.hasData || snapshot.data!.snapshot.value == null) {

return const Center(child: Text("Takip edilen yok"));

}

final data =

Map<String, dynamic>.from(snapshot.data!.snapshot.value as Map);

final takipEdilenUids = data.keys.toList();

return ListView.builder(

itemCount: takipEdilenUids.length,

itemBuilder: (context, index) {

final takipEdilenUid = takipEdilenUids[index];

return ListTile(

title: Text(takipEdilenUid),

);

},

);

},

),

);

}

}

class ProfilimPage extends StatefulWidget {

const ProfilimPage({super.key});

@override

State<ProfilimPage> createState() => \_ProfilimPageState();

}

class \_ProfilimPageState extends State<ProfilimPage> {

final user = FirebaseAuth.instance.currentUser!;

late DatabaseReference kullaniciRef;

late DatabaseReference universiteRef;

late DatabaseReference paylasimRef;

late DatabaseReference takipciRef;

late DatabaseReference takiplerRef;

Map<String, dynamic>? userData;

String? universiteAdi;

String? bolumAdi;

int gonderiSayisi = 0;

int takipciSayisi = 0;

int takipEdilenSayisi = 0;

bool isFollowing = false;

@override

void initState() {

super.initState();

kullaniciRef =

FirebaseDatabase.instance.ref().child('kullanicilar/${user.uid}');

universiteRef = FirebaseDatabase.instance.ref().child('universiteler');

paylasimRef = FirebaseDatabase.instance

.ref()

.child('kullanici\_paylasimlari/${user.uid}');

takipciRef = FirebaseDatabase.instance

.ref()

.child('kullanici\_takipciler/${user.uid}');

takiplerRef =

FirebaseDatabase.instance.ref().child('kullanici\_takipler/${user.uid}');

\_loadUserData();

\_listenTakipciSayisi();

\_listenTakipEdilenSayisi();

\_checkIfFollowing();

}

Future<void> \_loadUserData() async {

final snapshot = await kullaniciRef.get();

if (!snapshot.exists) return;

final data = Map<String, dynamic>.from(snapshot.value as Map);

final String? uniId = data['universiteId'] ?? data['universite'];

final String? bolumId = data['bolumId'] ?? data['bolum'];

final uniSnapshot = await universiteRef.child(uniId ?? '').get();

final bolumSnapshot =

await universiteRef.child('$uniId/bolumler/$bolumId').get();

final paylasimSnapshot = await paylasimRef.get();

setState(() {

userData = data;

universiteAdi = uniSnapshot.child('name').value as String?;

bolumAdi = bolumSnapshot.child('name').value as String?;

gonderiSayisi = paylasimSnapshot.children.length;

});

}

void \_signOut() async {

await FirebaseAuth.instance.signOut();

if (!mounted) return;

Navigator.pushReplacement(

context,

MaterialPageRoute(builder: (\_) => const LoginPage()),

);

}

void \_sifreSifirla() {

final email = userData?['email'];

if (email != null && email.toString().contains('@')) {

FirebaseAuth.instance.sendPasswordResetEmail(email: email);

ScaffoldMessenger.of(context).showSnackBar(

const SnackBar(content: Text('Şifre sıfırlama e-postası gönderildi.')),

);

}

}

void \_checkIfFollowing() async {

final currentUserUid = FirebaseAuth.instance.currentUser!.uid;

if (currentUserUid == user.uid) {

setState(() {

isFollowing = false;

});

return;

}

final currentUserTakiplerRef = FirebaseDatabase.instance

.ref('kullanici\_takipler/$currentUserUid/${user.uid}');

final snapshot = await currentUserTakiplerRef.get();

setState(() {

isFollowing = snapshot.exists;

});

}

void \_toggleFollow() async {

final currentUserUid = FirebaseAuth.instance.currentUser!.uid;

final takipciRefForUser = FirebaseDatabase.instance

.ref('kullanici\_takipciler/${user.uid}/$currentUserUid');

final takiplerRefForCurrentUser = FirebaseDatabase.instance

.ref('kullanici\_takipler/$currentUserUid/${user.uid}');

if (isFollowing) {

// Takipten çık

await takipciRefForUser.remove();

await takiplerRefForCurrentUser.remove();

} else {

// Takip et

await takipciRefForUser.set(true);

await takiplerRefForCurrentUser.set(true);

}

setState(() {

isFollowing = !isFollowing;

});

}

void \_listenTakipciSayisi() {

final takipciCountRef =

FirebaseDatabase.instance.ref('kullanici\_takipciler/${user.uid}');

takipciCountRef.onValue.listen((event) {

final data = event.snapshot.value;

setState(() {

takipciSayisi = (data != null && data is Map) ? data.length : 0;

});

});

}

void \_listenTakipEdilenSayisi() {

final takipEdilenCountRef =

FirebaseDatabase.instance.ref('kullanici\_takipler/${user.uid}');

takipEdilenCountRef.onValue.listen((event) {

final data = event.snapshot.value;

setState(() {

takipEdilenSayisi = (data != null && data is Map) ? data.length : 0;

});

});

}

@override

Widget build(BuildContext context) {

if (userData == null) {

return const Scaffold(

body: Center(child: CircularProgressIndicator()),

);

}

final currentUserUid = FirebaseAuth.instance.currentUser!.uid;

final isOwnProfile = currentUserUid == user.uid;

return Scaffold(

appBar: AppBar(

title: const Text("Profilim"),

actions: [

IconButton(

icon: const Icon(Icons.logout),

tooltip: "Çıkış Yap",

onPressed: \_signOut,

)

],

),

body: RefreshIndicator(

onRefresh: \_loadUserData,

child: SingleChildScrollView(

physics: const AlwaysScrollableScrollPhysics(),

child: Padding(

padding: const EdgeInsets.all(20.0),

child: Column(

children: [

CircleAvatar(

radius: 60,

backgroundImage: NetworkImage(userData!['avatarUrl'] ?? ''),

),

const SizedBox(height: 16),

Text(

userData!['isim'] ?? '',

style: const TextStyle(

fontSize: 22,

fontWeight: FontWeight.bold,

),

),

const SizedBox(height: 4),

Text(

userData!['email'] ?? '',

style: const TextStyle(color: Colors.grey),

),

const SizedBox(height: 20),

Row(

mainAxisAlignment: MainAxisAlignment.center,

children: [

\_buildStat("Gönderi", gonderiSayisi),

const SizedBox(width: 30),

InkWell(

onTap: () {

Navigator.push(

context,

MaterialPageRoute(

builder: (\_) => TakipcilerPage(uid: user.uid)),

);

},

child: \_buildStat("Takipçi", takipciSayisi),

),

const SizedBox(width: 30),

InkWell(

onTap: () {

Navigator.push(

context,

MaterialPageRoute(

builder: (\_) =>

TakipEdilenlerPage(uid: user.uid)),

);

},

child: \_buildStat("Takip Edilen", takipEdilenSayisi),

),

if (!isOwnProfile) ...[

const SizedBox(width: 30),

ElevatedButton(

onPressed: \_toggleFollow,

child: Text(isFollowing ? "Takipten Çık" : "Takip Et"),

)

]

],

),

const Divider(height: 32, thickness: 1),

ListTile(

leading: const Icon(Icons.school),

title: Text(universiteAdi ?? '-'),

subtitle: const Text("Üniversite"),

),

ListTile(

leading: const Icon(Icons.computer),

title: Text(bolumAdi ?? '-'),

subtitle: const Text("Bölüm"),

),

ListTile(

leading: const Icon(Icons.info\_outline),

title: Text(userData!['bio'] ?? 'Henüz bio eklenmedi.'),

subtitle: const Text("Biyografi"),

),

const SizedBox(height: 20),

ElevatedButton.icon(

onPressed: () {

Navigator.push(

context,

MaterialPageRoute(

builder: (\_) => const ProfilDuzenlePage()),

).then((\_) => \_loadUserData());

},

icon: const Icon(Icons.edit),

label: const Text("Profili Düzenle"),

),

const SizedBox(height: 12),

ElevatedButton(

onPressed: \_sifreSifirla,

child: const Text("Şifre Sıfırla"),

),

],

),

),

),

),

);

}

// Bu fonksiyon stat widget'larını döndürür

Widget \_buildStat(String baslik, int sayi) {

return Column(

children: [

Text(

sayi.toString(),

style: const TextStyle(fontWeight: FontWeight.bold, fontSize: 18),

),

Text(

baslik,

style: const TextStyle(color: Colors.grey),

),

],

);

}

}

**Pages 🡪 profıl\_duzenle\_page.dart kodları**

import 'package:flutter/material.dart';

import 'package:firebase\_auth/firebase\_auth.dart';

import 'package:firebase\_database/firebase\_database.dart';

class ProfilDuzenlePage extends StatefulWidget {

const ProfilDuzenlePage({super.key});

@override

State<ProfilDuzenlePage> createState() => \_ProfilDuzenlePageState();

}

class \_ProfilDuzenlePageState extends State<ProfilDuzenlePage> {

final user = FirebaseAuth.instance.currentUser;

late DatabaseReference kullaniciRef;

final \_isimController = TextEditingController();

final \_bioController = TextEditingController();

final \_avatarUrlController = TextEditingController();

String? email;

@override

void initState() {

super.initState();

kullaniciRef =

FirebaseDatabase.instance.ref().child('kullanicilar/${user!.uid}');

\_loadUserData();

}

Future<void> \_loadUserData() async {

final snapshot = await kullaniciRef.get();

if (!snapshot.exists) return;

final dataRaw = snapshot.value;

if (dataRaw == null) return;

final data = Map<String, dynamic>.from(

(dataRaw as Map).map((key, value) => MapEntry(key.toString(), value)),

);

setState(() {

\_isimController.text = data['isim']?.toString() ?? '';

\_bioController.text = data['bio']?.toString() ?? '';

\_avatarUrlController.text = data['avatarUrl']?.toString() ?? '';

email = data['email']?.toString();

});

}

Future<void> \_saveChanges() async {

await kullaniciRef.update({

'isim': \_isimController.text.trim(),

'bio': \_bioController.text.trim(),

'avatarUrl': \_avatarUrlController.text.trim(),

// Üniversite ve bölüm bilgileri güncellenmiyor

});

if (mounted) {

ScaffoldMessenger.of(context).showSnackBar(

const SnackBar(content: Text("Profil başarıyla güncellendi.")),

);

Navigator.pop(context);

}

}

Widget \_buildAvatarPreview() {

final url = \_avatarUrlController.text.trim();

if (url.isEmpty) return const SizedBox.shrink();

return Padding(

padding: const EdgeInsets.symmetric(vertical: 12),

child: ClipRRect(

borderRadius: BorderRadius.circular(60),

child: Image.network(

url,

width: 120,

height: 120,

fit: BoxFit.cover,

errorBuilder: (context, error, stackTrace) {

return Container(

width: 120,

height: 120,

color: Colors.grey[300],

alignment: Alignment.center,

child:

const Icon(Icons.broken\_image, size: 48, color: Colors.grey),

);

},

loadingBuilder: (context, child, loadingProgress) {

if (loadingProgress == null) return child;

return Container(

width: 120,

height: 120,

alignment: Alignment.center,

child: const CircularProgressIndicator(),

);

},

),

),

);

}

InputDecoration \_inputDecoration(String label) {

return InputDecoration(

labelText: label,

border: OutlineInputBorder(borderRadius: BorderRadius.circular(10)),

focusedBorder: OutlineInputBorder(

borderSide: const BorderSide(color: Colors.blueAccent, width: 2),

borderRadius: BorderRadius.circular(10),

),

contentPadding: const EdgeInsets.symmetric(horizontal: 16, vertical: 14),

);

}

@override

Widget build(BuildContext context) {

final theme = Theme.of(context);

return Scaffold(

appBar: AppBar(

title: const Text("Profili Düzenle"),

centerTitle: true,

backgroundColor: Colors.blueAccent,

),

body: Padding(

padding: const EdgeInsets.symmetric(horizontal: 20, vertical: 20),

child: ListView(

children: [

\_buildAvatarPreview(),

TextField(

controller: \_avatarUrlController,

decoration: \_inputDecoration("Profil Fotoğrafı URL"),

keyboardType: TextInputType.url,

onChanged: (\_) {

setState(() {}); // Fotoğraf önizlemesi güncelle

},

),

const SizedBox(height: 20),

TextField(

controller: \_isimController,

decoration: \_inputDecoration("Adınız"),

textCapitalization: TextCapitalization.words,

),

const SizedBox(height: 20),

TextField(

controller: \_bioController,

decoration: \_inputDecoration("Biyografi"),

maxLines: 4,

),

const SizedBox(height: 25),

if (email != null && email!.isNotEmpty)

Text(

"E-posta: $email",

style: theme.textTheme.bodyMedium

?.copyWith(color: Colors.grey[700]),

),

const SizedBox(height: 35),

SizedBox(

height: 50,

child: ElevatedButton(

style: ElevatedButton.styleFrom(

backgroundColor: Colors.blueAccent,

shape: RoundedRectangleBorder(

borderRadius: BorderRadius.circular(12)),

textStyle: const TextStyle(

fontSize: 18, fontWeight: FontWeight.w600),

),

onPressed: \_saveChanges,

child: const Text("Değişiklikleri Kaydet"),

),

),

],

),

),

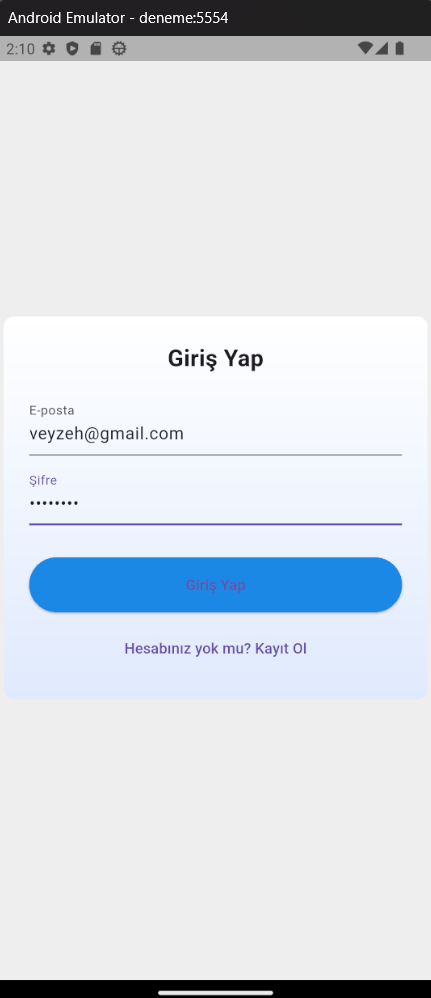
);

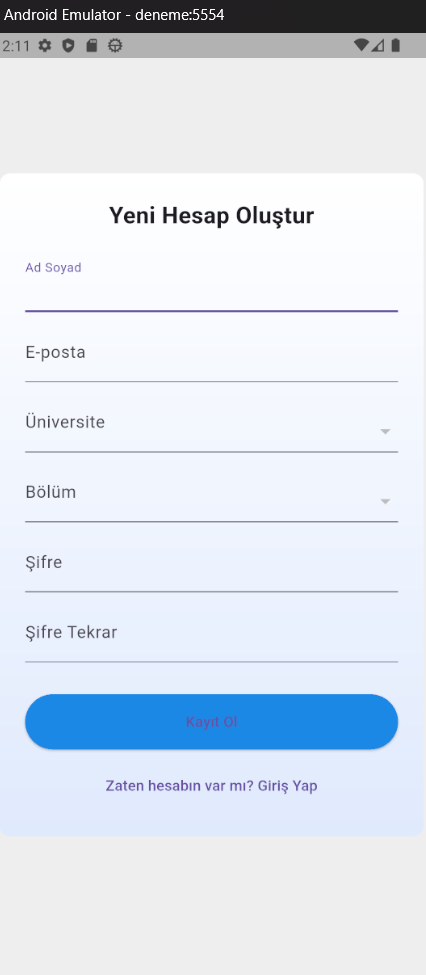
}

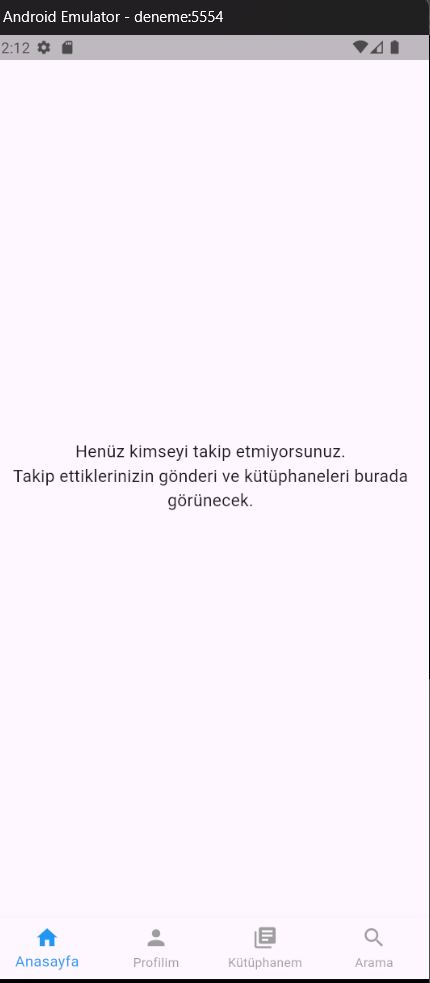
}

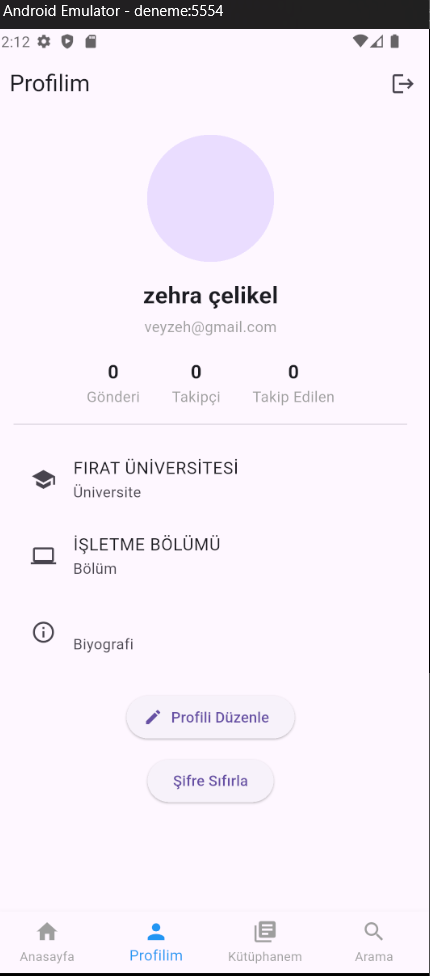
**Android emülatörden uygulama görüntüleri**

**Bazı firebase storgae bağlantıları düzgün bir şekilde gerçekleştirilemedi flutter ve dart yeni bir dil olması dolayısıyla proje sırasından da bi çok kez paket hatası aldım ve bu seviyeye kadar getirebildim relatime detabase bağlantıları tam fakat storage bağlantıları gerçekleştirilemediği için kullanıcı paylaşımları url uzerinden gözükecektir .Aşağıda Uygulamadan birkaç ekran görüntüsü ve örnekler var…**

****

****

****

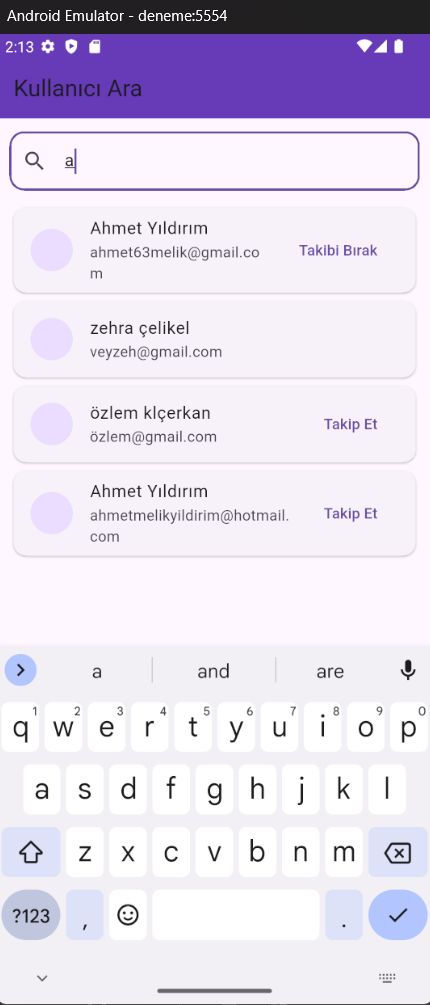
****

****

****

****

****

****

**FİRBASE REALTİME DATABASE**

[**kullanici\_takipleri**](https://console.firebase.google.com/u/0/project/akademikbilgierisimagi/database/akademikbilgierisimagi-default-rtdb/data/~2F)

**arrow\_drop\_down**[**kullanicilar**](https://console.firebase.google.com/u/0/project/akademikbilgierisimagi/database/akademikbilgierisimagi-default-rtdb/data/~2Fkullanicilar)

**arrow\_drop\_down**[**kulüpler**](https://console.firebase.google.com/u/0/project/akademikbilgierisimagi/database/akademikbilgierisimagi-default-rtdb/data/~2F)

**arrow\_drop\_down**[**kutuphane**](https://console.firebase.google.com/u/0/project/akademikbilgierisimagi/database/akademikbilgierisimagi-default-rtdb/data/~2F)

**arrow\_drop\_down**[**mesajlar**](https://console.firebase.google.com/u/0/project/akademikbilgierisimagi/database/akademikbilgierisimagi-default-rtdb/data/~2F)

**arrow\_drop\_down**[**notlar**](https://console.firebase.google.com/u/0/project/akademikbilgierisimagi/database/akademikbilgierisimagi-default-rtdb/data/~2F)

**arrow\_drop\_down**[**ozel\_dokumanlar**](https://console.firebase.google.com/u/0/project/akademikbilgierisimagi/database/akademikbilgierisimagi-default-rtdb/data/~2Fozel_dokumanlar)

**arrow\_drop\_down**[**paylasim\_yorumlari**](https://console.firebase.google.com/u/0/project/akademikbilgierisimagi/database/akademikbilgierisimagi-default-rtdb/data/~2Fpaylasim_yorumlari)

**arrow\_drop\_down**[**paylasimlar**](https://console.firebase.google.com/u/0/project/akademikbilgierisimagi/database/akademikbilgierisimagi-default-rtdb/data/~2F)

**arrow\_drop\_down**[**sikayetler**](https://console.firebase.google.com/u/0/project/akademikbilgierisimagi/database/akademikbilgierisimagi-default-rtdb/data/~2F)

**arrow\_drop\_down**[**sorular**](https://console.firebase.google.com/u/0/project/akademikbilgierisimagi/database/akademikbilgierisimagi-default-rtdb/data/~2F)

**arrow\_drop\_down**[**takip\_edilenler**](https://console.firebase.google.com/u/0/project/akademikbilgierisimagi/database/akademikbilgierisimagi-default-rtdb/data/~2F)

**arrow\_drop\_down**[**takipler**](https://console.firebase.google.com/u/0/project/akademikbilgierisimagi/database/akademikbilgierisimagi-default-rtdb/data/~2F)

**arrow\_drop\_down**[**topluluk\_uyeleri**](https://console.firebase.google.com/u/0/project/akademikbilgierisimagi/database/akademikbilgierisimagi-default-rtdb/data/~2Ftopluluk_uyeleri)

**arrow\_drop\_down**[**topluluklar**](https://console.firebase.google.com/u/0/project/akademikbilgierisimagi/database/akademikbilgierisimagi-default-rtdb/data/~2F)

**arrow\_drop\_down**[**universiteler**](https://console.firebase.google.com/u/0/project/akademikbilgierisimagi/database/akademikbilgierisimagi-default-rtdb/data/~2F)

**arrow\_drop\_down**[**userActivities**](https://console.firebase.google.com/u/0/project/akademikbilgierisimagi/database/akademikbilgierisimagi-default-rtdb/data/~2F)

**Firebase Rules**

**{**

**"rules": {**

**"paylasimlar": {**

**".read": "auth != null",**

**".write": "auth != null"**

**},**

**"kullanici\_paylasimlari": {**

**"$userId": {**

**".read": "auth != null && auth.uid === $userId",**

**".write": "auth != null && auth.uid === $userId"**

**}**

**},**

**"kullanici\_takipleri": {**

**"$userId": {**

**".read": "auth != null",**

**".write": "auth != null"**

**}**

**},**

**"kullanici\_takipciler": {**

**"$userId": {**

**".read": "auth != null",**

**".write": "auth != null"**

**}**

**},**

**"kullanicilar": {**

**".read": "auth != null",**

**"$uid": {**

**".write": "auth != null && auth.uid === $uid"**

**}**

**},**

**"universiteler": {**

**".read": true,**

**".write": "auth != null"**

**},**

**"bolumler": {**

**".read": true,**

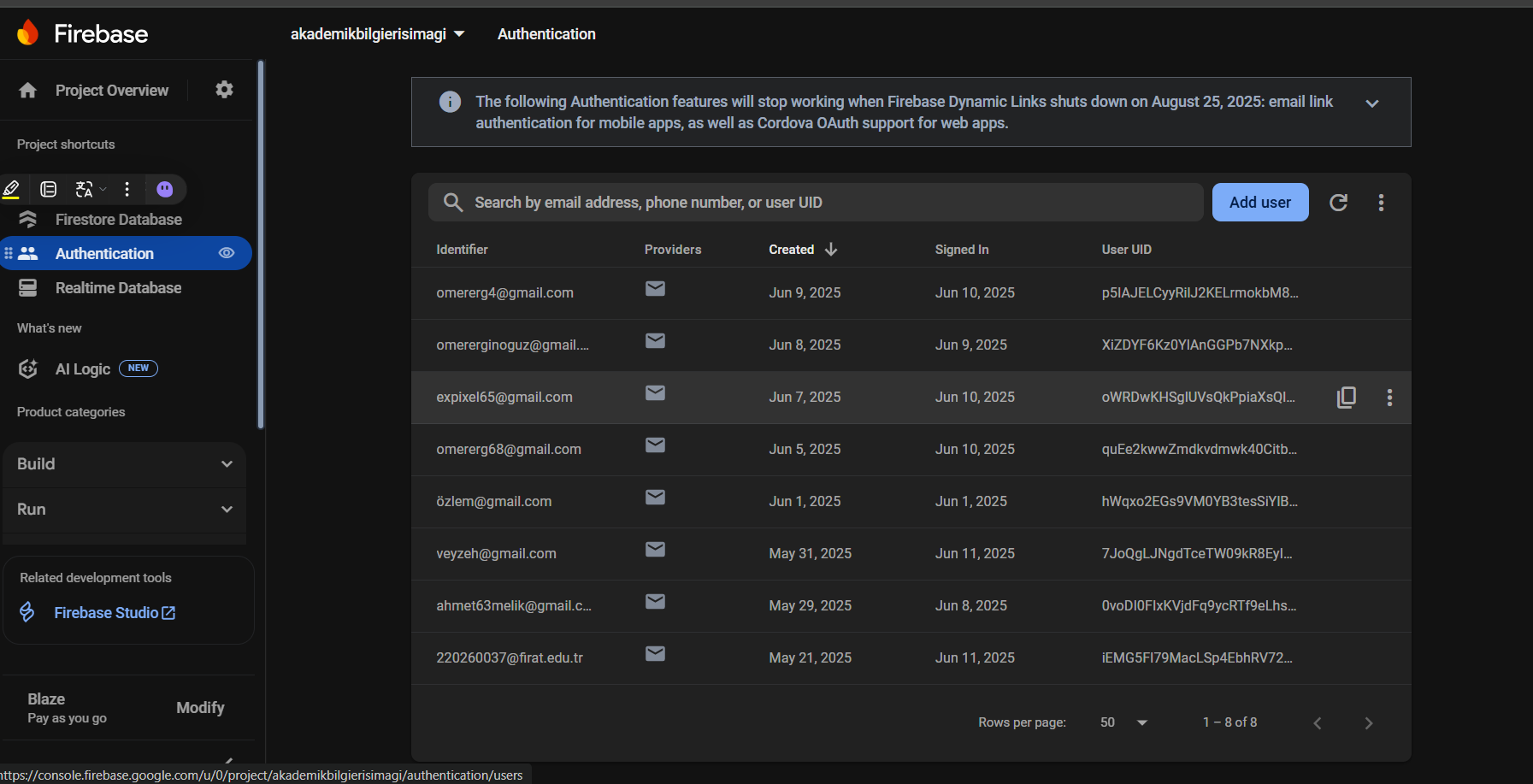
**".write": "auth != null"**

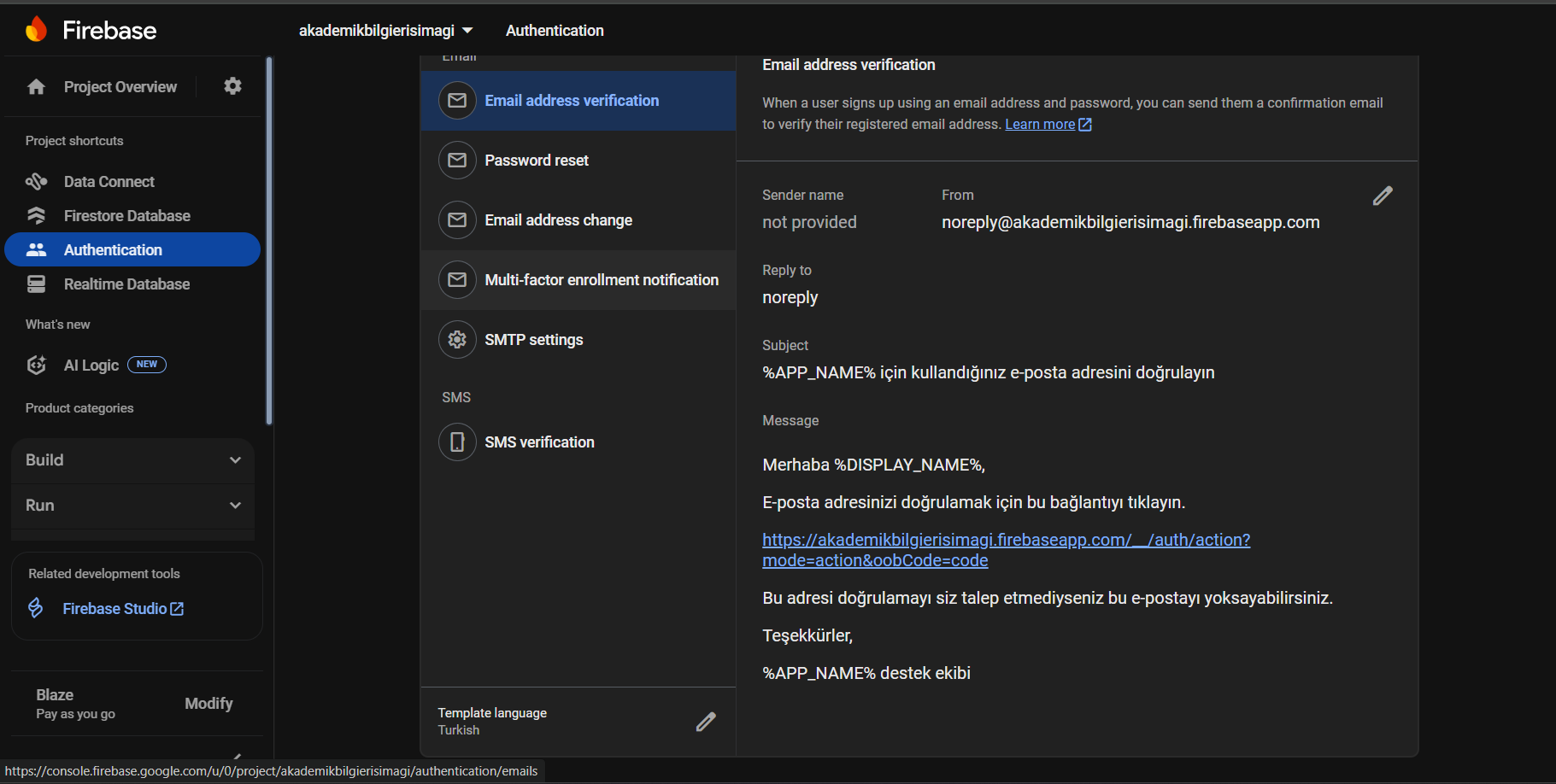
**}**

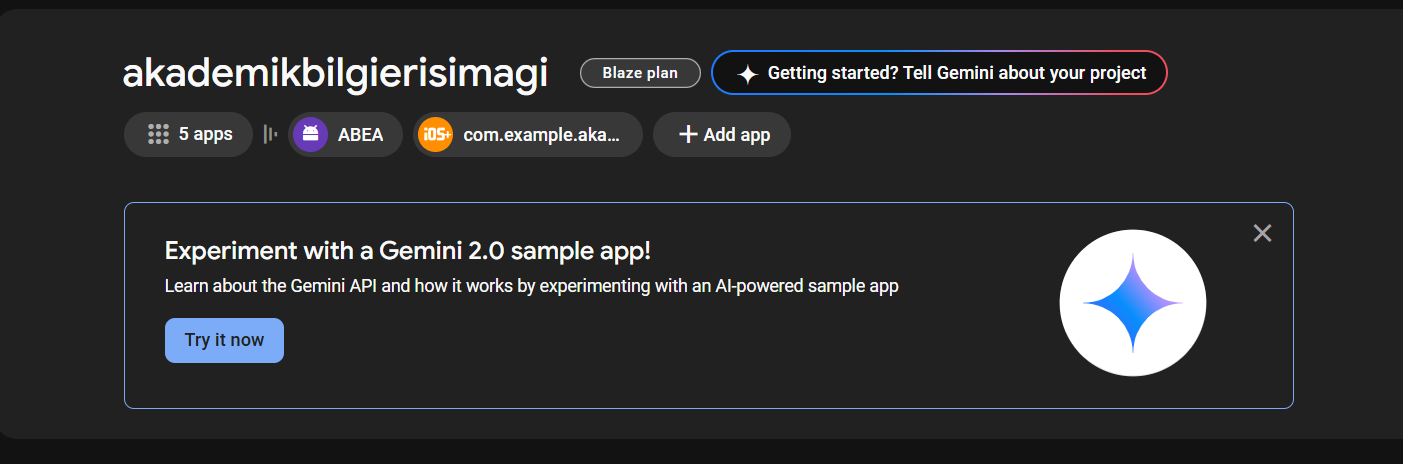
**}**

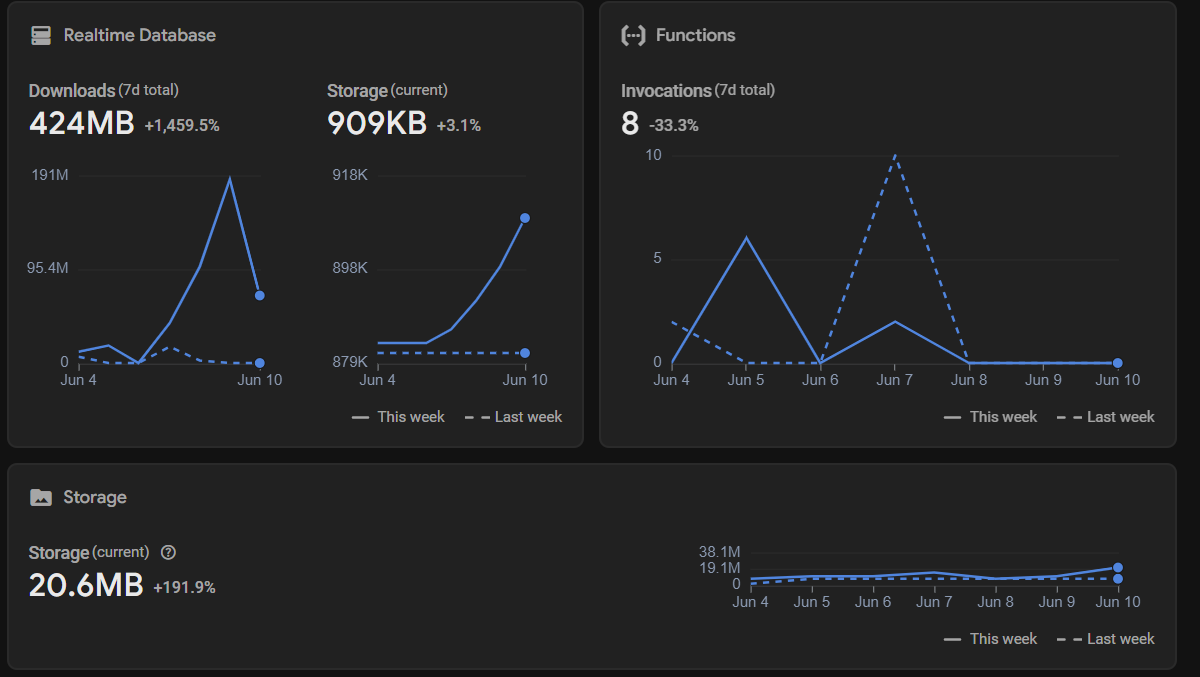
**}**

**Firebas authentication**

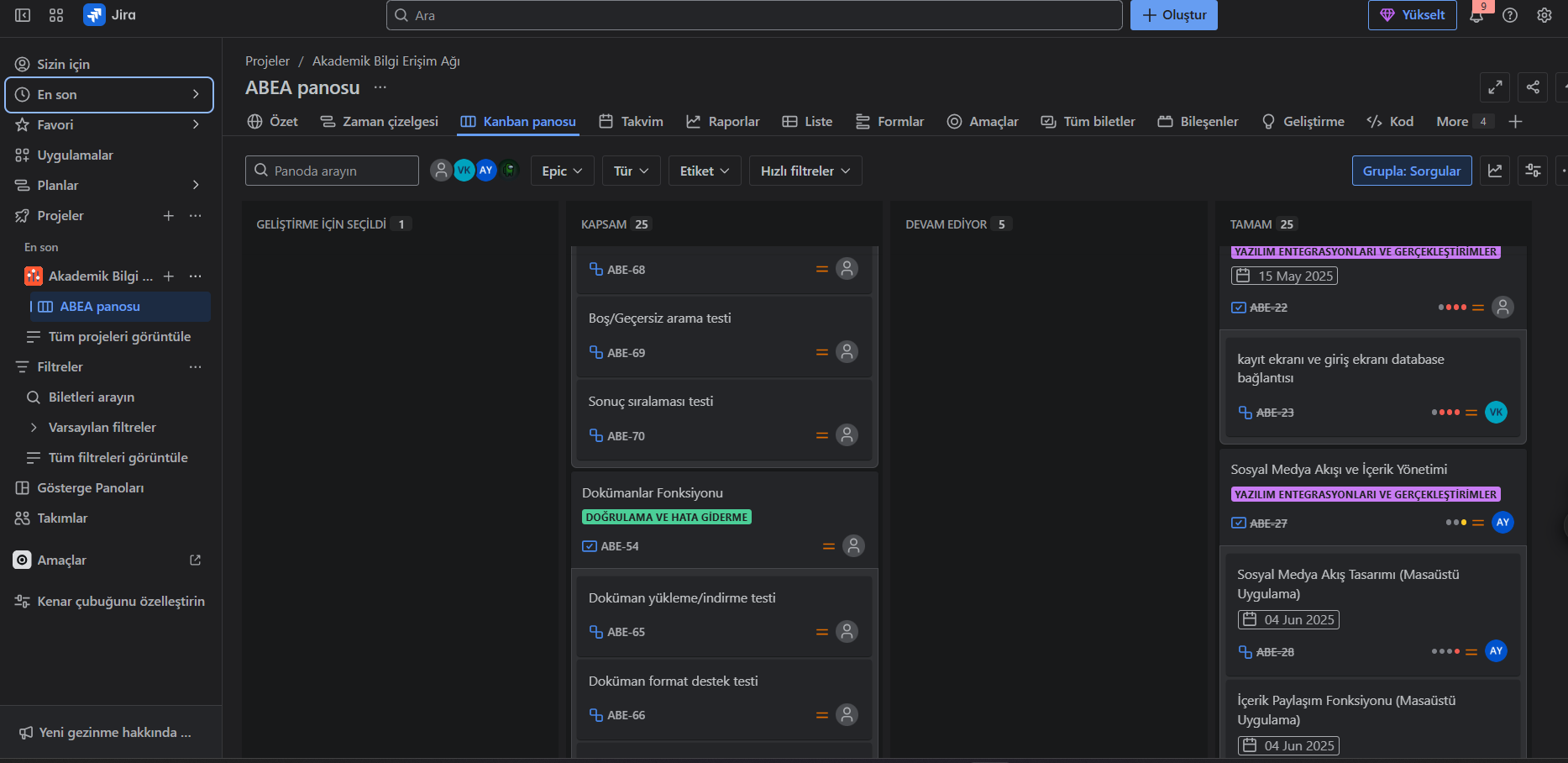
****

****

****

****

**JİRA**

****