class **ProfilResmiDB** {

FirebaseAuth **firebaseAuth** = FirebaseAuth.instance;

FirebaseStorage **firebaseStorage** = FirebaseStorage.instance;

Future<String?> **resimYukle**(File file) async **{**

var yol = firebaseStorage

.ref()

.child('profilResmi')

.child(firebaseAuth.currentUser!.uid)

.child('profilResmi.png');

await yol.putFile(file);

String url = await yol.getDownloadURL();

return url;

**}**

Future<String?> **resimcek()** async **{**

var yol = firebaseStorage

.ref()

.child('profilResmi')

.child(firebaseAuth.currentUser!.uid)

.child('profilResmi.png');

try {

String url = await yol.getDownloadURL();

return url;

} catch (e) {

return null;

}

**}**

Future<String?> **idyeGoreResimCek**(String userID) async **{**

var yol = firebaseStorage

.ref()

.child('profilResmi')

.child(userID)

.child('profilResmi.png');

try {

String url = await yol.getDownloadURL();

return url;

} catch (e) {

return null;

}

**}**

class **PostDB** {

FirebaseAuth **firebaseAuth** = FirebaseAuth.instance;

FirebaseStorage **firebase** = FirebaseStorage.instance;

final FirebaseFirestore **db** = FirebaseFirestore.instance**;**

Future<String?> muzikSave(

File **file**,

String **userID**,

String **muzikPath**,

String **klasorId**)

async **{**

try {

Reference **storageReference** = FirebaseStorage.instance

.ref()

.child('’muzikPath’')

.child(firebaseAuth.currentUser!.uid)

.child(klasorId)

.child(muzikPath);

UploadTask uploadTask = **storageReference**.putFile(file);

await uploadTask.whenComplete(() => null);

String **downloadUrl** = await **storageReference**.getDownloadURL();

return downloadUrl;

} catch (e) {

return null;

}

}

Future<void> **postYorumEkleme**

**(** String **postID**, YorumModel **yorumModel)** async {

final **documentReference** = FirebaseFirestore.instance.collection(**'tmpostlar'**).doc(**postID**);

await **documentReference**.update({

'’yorumlar’':FieldValue.arrayUnion([**yorumModel**.toMap()]),

});

}

////////

Future<List<**YorumModel**>?> **yorumlariGetir (**String **postID)** async {

List<YorumModel> **yorumLisem** = [];

var **documentReference** =

await FirebaseFirestore.instance

.collection('tmpostlar')

.doc(postID)

.get();

var data = **documentReference**.data();

if (data != null) {

var yorumList = data['yorumlar'];

for (var element in yorumList) {

YorumModel yorumModel = YorumModel.fromMap(element);

yorumLisem.add(yorumModel);

}

return yorumLisem; } else { return null; }

Future<void> **begenArttirAzalt (**PostLikeModel postLikeModel, bool begeniBool**)** async {

var **poslartRef** = db

.collection('postLike')

.doc(postLikeModel.postID)

.collection('bb')

.doc(postLikeModel.userID);

var **postRef** =

db.collection('tmpostlar')

.doc(postLikeModel.postID);

//

if (begeniBool == true) **{**

await **postRef**.update({'begeniSayisi': **FieldValue.increment(1)**});

await **poslartRef**.set(postLikeModel.toMap());

**}** else **{**

await **poslartRef**.set(postLikeModel.toMap());

await **postRef**.update({'begeniSayisi': **FieldValue.increment(-1)**});

}

}

class PostModel {

.

.

**.buralarda diğer paremetreler var.Yer kaplamasın diye sildim.**

.

List<YorumModel>? yorumlar;

Map<String, dynamic> toMap() {

return {

"yorumlar": **yorumlar** != null

**?** **yorumlar**!.map((e) => e.toMap()).toList()

**: [ ]**

};

PostModel.fromMap(Map<String, dynamic> map) yorumlar = map["yorumlar"],}

***Sayfanın en üstünde***

List<YorumModel> **allYorumlar** = [];

***initState içerisinde çağrılacak metot kodları***

Future<void> **yorumlariGetir**() async {

var **dbProviider** = Provider.of<PostDB>(context, listen: false);

var **gelenList** = await **dbProviider**.yorumlariGetir(widget.postID);

// ignore: unrelated\_type\_equality\_checks

if (gelenList!.isNotEmpty) {

**allYorumlar** = **gelenList**;

}

if (this.mounted) {

setState(() {

});

}