Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра программного обеспечения информационных технологий

Дисциплина: Разработка программного обеспечения для мобильных платформ

ОТЧЕТ

По лабораторной работе №2

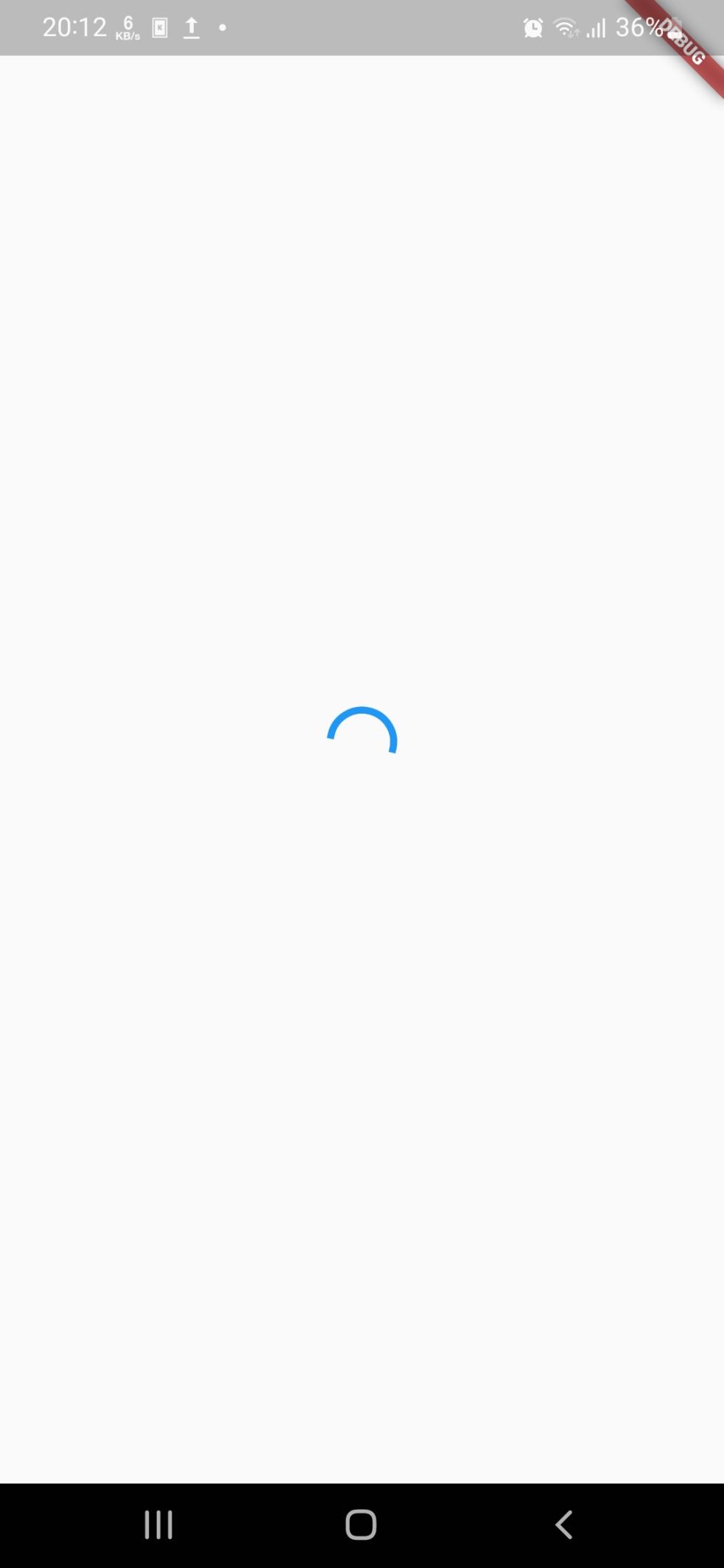
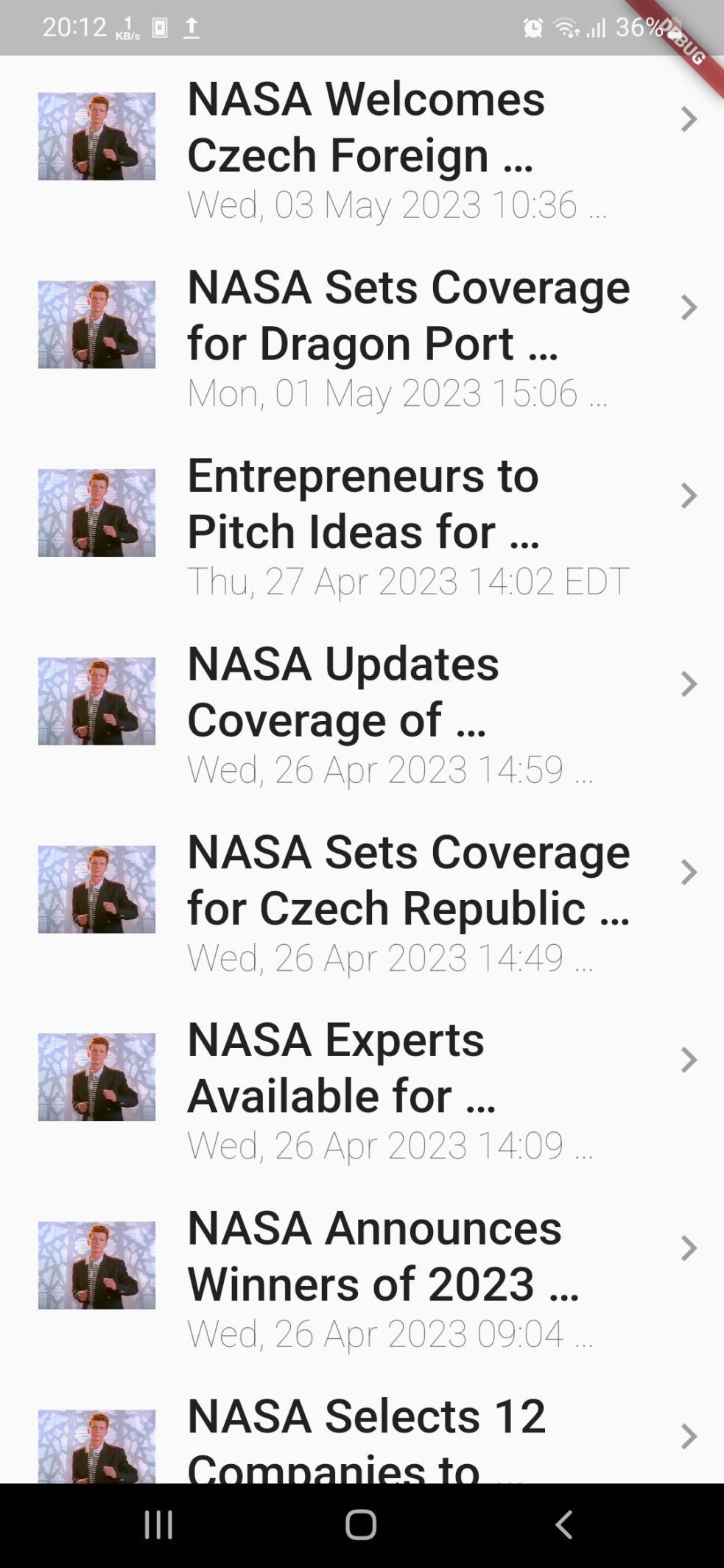
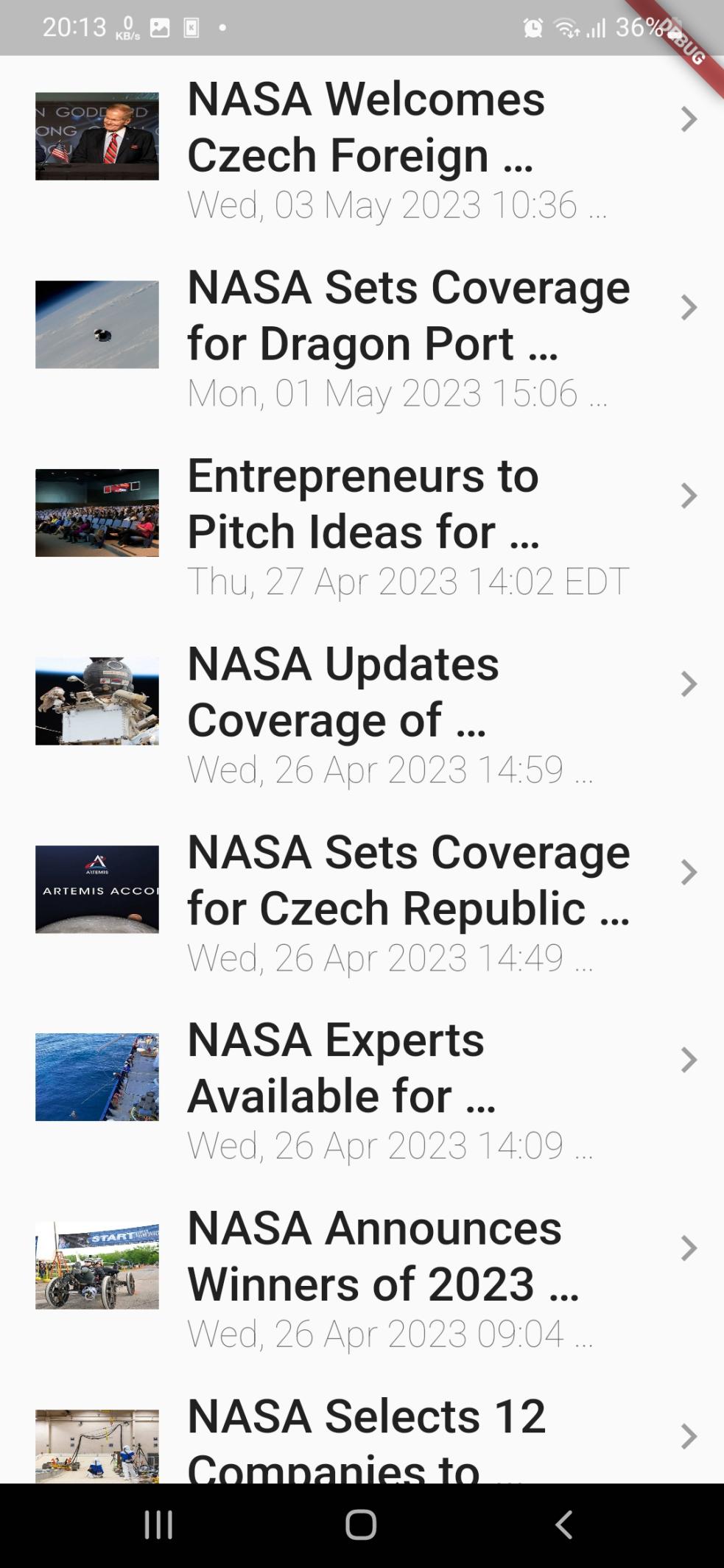
Тема работы: «RSS-reader»

Выполнил:

студент: гр. 051006 Шуляк А.В.

Проверил: Коловайтис Н. А.

Минск 2023



Исходный код:

home.dart:

// ignore\_for\_file: import\_of\_legacy\_library\_into\_null\_safe

import 'dart:developer';

import 'package:cached\_network\_image/cached\_network\_image.dart';

import 'package:flutter/material.dart';

import 'package:dart\_rss/dart\_rss.dart';

import "package:http/http.dart" as http;

import 'package:rss/webview.dart';

class Home extends StatefulWidget {

@override

State<StatefulWidget> createState() => \_Home();

}

class \_Home extends State<Home> {

static const String url = 'https://www.nasa.gov/rss/dyn/breaking\_news.rss';

RssFeed? \_feed;

static const String placeholderImg = 'assets/rickroll.png';

late GlobalKey<RefreshIndicatorState> \_refreshKey;

Future<void> openFeed(String url) async {

log(url);

await Navigator.push(

context, MaterialPageRoute(builder: (context) => WebView(url)));

}

Future<RssFeed> loadFeed() async {

try {

final client = http.Client();

final response = await client.get(Uri.parse(url));

return RssFeed.parse(response.body);

} catch (e) {

//

}

return const RssFeed();

}

load() async {

loadFeed().then((result) {

if (result == const RssFeed()) {

//TODO: show info aput error!

return;

}

updateFeed(result);

});

}

updateFeed(feed) {

setState(() {

\_feed = feed;

});

}

@override

void initState() {

super.initState();

\_refreshKey = GlobalKey<RefreshIndicatorState>();

load();

}

title(title) {

return Text(

title,

style: const TextStyle(fontSize: 18.0, fontWeight: FontWeight.w500),

maxLines: 2,

overflow: TextOverflow.ellipsis,

);

}

subtitle(subTitle) {

return Text(

subTitle,

style: const TextStyle(fontSize: 14.0, fontWeight: FontWeight.w100),

maxLines: 1,

overflow: TextOverflow.ellipsis,

);

}

thumbnail(imageUrl) {

return Padding(

padding: const EdgeInsets.only(left: 15.0),

child: CachedNetworkImage(

placeholder: (context, url) => Image.asset(placeholderImg),

imageUrl: imageUrl,

height: 50,

width: 70,

alignment: Alignment.center,

fit: BoxFit.fill,

),

);

}

rightIcon() {

return const Icon(

Icons.keyboard\_arrow\_right,

color: Colors.grey,

size: 30.0,

);

}

list() {

return ListView.builder(

itemCount: \_feed?.items.length,

itemBuilder: (BuildContext context, int index) {

final item = \_feed!.items[index];

return ListTile(

title: title(item.title),

subtitle: subtitle(item.pubDate),

leading: thumbnail(item.enclosure?.url),

trailing: rightIcon(),

contentPadding: const EdgeInsets.all(5.0),

onTap: () => openFeed(item.link!),

);

},

);

}

isFeedEmpty() {

return null == \_feed?.items;

}

@override

Widget build(BuildContext context) {

return isFeedEmpty()

? const Center(

child: CircularProgressIndicator(),

)

: RefreshIndicator(

key: \_refreshKey,

child: list(),

onRefresh: () async => load(),

);

}

}

main.dart:

import 'package:flutter/material.dart';

import 'package:rss/home.dart';

void main() {

runApp(const MyApp());

}

class MyApp extends StatelessWidget {

const MyApp({super.key});

@override

Widget build(BuildContext context) {

return MaterialApp(

title: 'RSS Reader',

theme: ThemeData(

primarySwatch: Colors.blue,

),

home: Material(child: Home()),

);

}

}

webview.dart:

import 'package:flutter/material.dart';

import 'package:webview\_flutter/webview\_flutter.dart';

class WebView extends StatefulWidget {

final String url;

WebView(this.url);

@override

State<StatefulWidget> createState() => \_WebView(url);

}

class \_WebView extends State<WebView> {

final String url;

late WebViewController controller;

\_WebView(this.url);

@override

void initState() {

controller = WebViewController()

..setJavaScriptMode(JavaScriptMode.unrestricted)

..setNavigationDelegate(NavigationDelegate(

onProgress: (progress) {},

onPageStarted: (url) {},

onPageFinished: (url) {},

onWebResourceError: (error) {},

))

..loadRequest(Uri.parse(url));

super.initState();

}

@override

Widget build(BuildContext context) {

return WebViewWidget(controller: controller);

}

}