Проект

**Weather for kids**

***Геннадий Фурдуй***

Фак. Номер 1601321027

Софтуерно инженерство III курс

г. Пловдив

2018г.

**Целта на проекта:** изграждане на приложениетоза показ на времето в място където се намира ползовател.

**Основни функционалности**:

* Определяне на текуща геолокация на ползовател
* Зареждане на прогноз на времето
* Отобразяване на прогноз на времето за днес и текуща седмица
* Отобразяване на подробен прогноз на времето за избраният ден

**Системни изисквания**:

* устройство което работи на ОС Android или iOS
* GPS сензор

Проект е реализиран на език за програмиране Dart с испозването програмна рамка Flutter.

Исползван бесплатен API за времето https://openweathermap.org/api

**Дизайн на проекта**

Приложение се состои от три екраните: Splash screen, List screen, Details screen

В приложение исползван шрифт “Five Years”

Икони за времето са **нарисувани от деца** на украински център на творчеството г.Запорожие специално **за този приложение**.



1. **Splash screen**

В момента на зареждане на прогноз отразява икона «Облак». На край на зареждане отразява икона «Слънце»

*LandingPage.dart*

@override

Widget build(BuildContext context) {

return Container(

decoration: BoxDecoration(

image: DecorationImage(

image: AssetImage("assets/background.png"), fit: BoxFit.cover)),

child: Column(

mainAxisAlignment: MainAxisAlignment.center,

children: <Widget>[

Icon(isLoading ? Icons.wb\_cloudy : Icons.wb\_sunny, size: 80.0,),

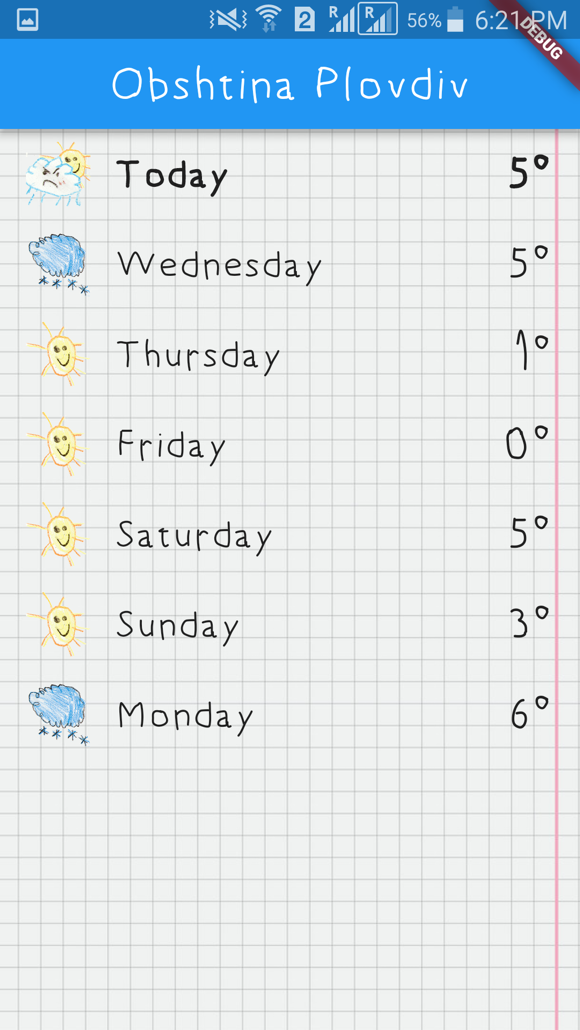
Text("Weather", style: TextStyle(fontSize: 50.0),),

],

),

);

}

1. **List screen**

Показва се след зареждане на прогноз

В хедър се показва текущо място определено с помоща на геолокация GPS

Всеки един елемент от списка показва икона за времето, ден от седмица или «Today», средна температура

*DaysForecastPage.dart*

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: new AppBar(

title: Text(\_forecast.city, style: TextStyle(fontSize: 30.0),),

automaticallyImplyLeading: false,

centerTitle: true,

),

body: Container(

decoration: BoxDecoration(

image: DecorationImage(

image: AssetImage("assets/background.png"), fit: BoxFit.cover)),

child: ListView.builder(

itemCount: \_forecast.list.length,

itemBuilder: (context, int index) {

return ForecastListItem(\_forecast.list[index]);

},

),

),

);

}

*ForecatListItem.dart*

import 'package:intl/intl.dart';

import 'package:flutter/material.dart';

import '../models/OneDatyForecast.dart';

import '../pages/OneDayForecastPage.dart';

final todayDate = new DateFormat("d").format(new DateTime.now());

class ForecastListItem extends StatelessWidget {

final OneDayForecast \_item;

ForecastListItem(this.\_item);

@override

Widget build(BuildContext context) {

var eDateStr = "";

if (todayDate == new DateFormat('d').format(\_item.date)) {

eDateStr = "Today";

} else {

eDateStr = new DateFormat("EEEE").format(\_item.date);

}

return InkWell(

onTap: () => Navigator.of(context)

.push(new MaterialPageRoute(builder: (BuildContext context) {

return new OneDayForecastPage(\_item);

})),

child: ListTile(

leading: Material(

color: Colors.transparent,

child: Image.asset(

'assets/${\_item.icon.substring(0, 2)}.png',

width: 40.0,

height: 40.0,

)),

title: Text(

eDateStr,

style: TextStyle(

fontWeight:

eDateStr == 'Today' ? FontWeight.bold : FontWeight.normal,

fontSize: 25.0),

),

trailing: Text(

\_item.temp.toStringAsFixed(0) + "\u00B0",

style: TextStyle(

fontWeight:

eDateStr == 'Today' ? FontWeight.bold : FontWeight.normal,

fontSize: 30.0),

),

),

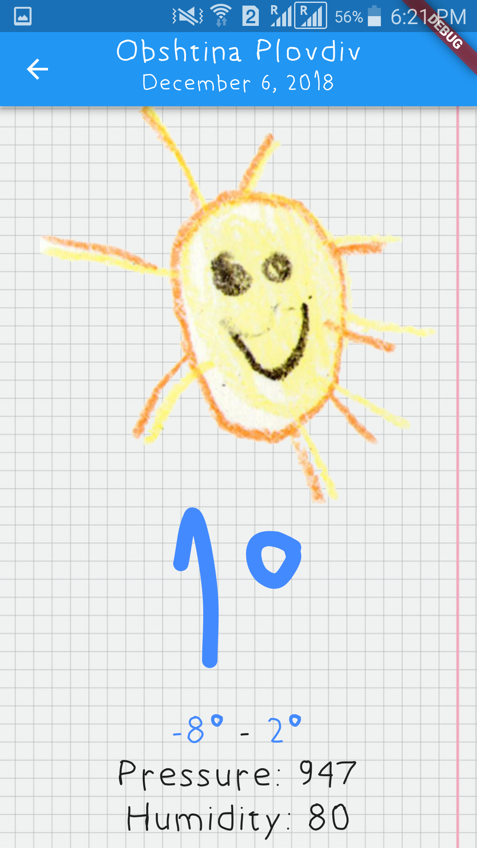
);

}

}

1. **Details screen**

Показва се след избрания дена от списка

*OneDayForecastPage.dart*

return Scaffold(

appBar: AppBar(

title: Column(

children: <Widget>[

Text(

weather.name,

style: TextStyle(fontSize: 25.0),

),

Text(

new DateFormat('yMMMMd').format(weather.date),

style: TextStyle(fontSize: 18.0),

),

],

),

centerTitle: true,

),

body: new Container(

decoration: BoxDecoration(

image: DecorationImage(

image: AssetImage("assets/background.png"), fit: BoxFit.cover)),

child: Column(

children: <Widget>[

Image.asset(

'assets/${weather.icon.substring(0, 2)}.png',

height: 300.0,

width: 300.0,

fit: BoxFit.fill,

),

Row(

mainAxisAlignment: MainAxisAlignment.center,

children: <Widget>[

Text(

weather.temp.toStringAsFixed(0),

style: TextStyle(

fontWeight: FontWeight.bold,

fontSize: mainTempSize,

color: weather.temp < 18.0 ? coldColor : warmColor

),

),

Text(

"\u00B0",

style: TextStyle(

fontWeight: FontWeight.bold,

fontSize: 140.0,

color: weather.temp < 18.0 ? coldColor : warmColor

),

),

],

),

Row(

mainAxisAlignment: MainAxisAlignment.center,

children: <Widget>[

Text(

weather.minTemp.toStringAsFixed(0),

style: TextStyle(fontSize: regularTextSize, color: weather.minTemp < 18.0 ? coldColor : warmColor),

),

Text(

"\u00B0",

style: TextStyle(

fontWeight: FontWeight.bold,

fontSize: regularTextSize,

color: weather.minTemp < 18.0 ? coldColor : warmColor

),

),

Text(

" - ",

style: TextStyle(fontSize: regularTextSize,),

),

Text(

weather.maxTemp.toStringAsFixed(0),

style: TextStyle(fontSize: regularTextSize, color: weather.maxTemp < 18.0 ? coldColor : warmColor),

),

Text(

"\u00B0",

style: TextStyle(

fontWeight: FontWeight.bold,

fontSize: regularTextSize,

color: weather.maxTemp < 18.0 ? coldColor : warmColor

),

),

],

),

Row(

mainAxisAlignment: MainAxisAlignment.center,

children: <Widget>[

Text(

"Pressure: ",

style: TextStyle(fontSize: regularTextSize,),

),

Text(

weather.pressure.toStringAsFixed(0),

style: TextStyle(fontSize: regularTextSize,),

),

],

),

Row(

mainAxisAlignment: MainAxisAlignment.center,

children: <Widget>[

Text(

"Humidity: ",

style: TextStyle(fontSize: regularTextSize,),

),

Text(

weather.humidity.toStringAsFixed(0),

style: TextStyle(fontSize: regularTextSize,),

),

],

),

],

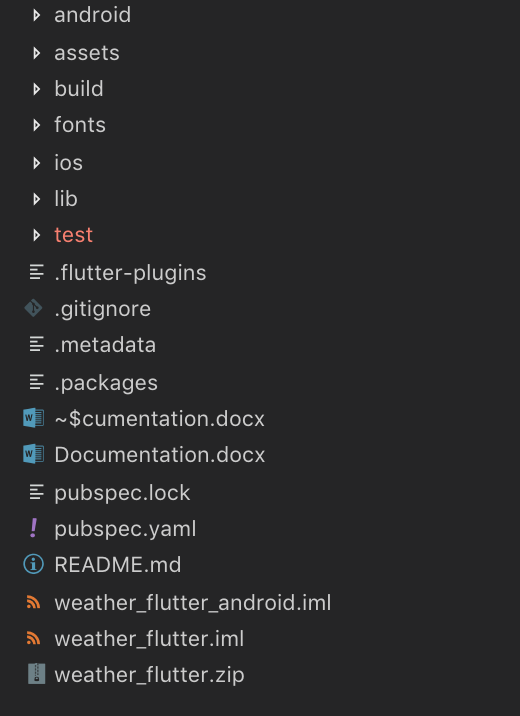
),

),

);

**Фунционалноста на проекта**

Файлова структура Flutter проекта

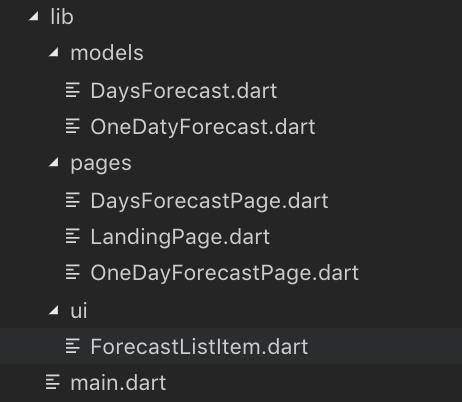


**android/ios** - код генериран от Flutter за съответните платформи

**assets** – ресурсите (.png файли икони на времето)

**fonts** – шрифтите (.ttf)

**lib** – файли на проекта



*main.dart* Главния файл на проекта

import 'package:flutter/material.dart';

import './pages/LandingPage.dart';

void main() {

runApp(MaterialApp(

theme: ThemeData(fontFamily: 'FiveYears'),

home: new LandingPage()

));

}

За работа с API ползваме следващи модели на данните

*OneDayForecast.dart*

class OneDayForecast {

final DateTime date;

final String name;

final double temp;

final String main;

final String icon;

final double minTemp;

final double maxTemp;

final double pressure;

final double humidity;

OneDayForecast({

this.date,

this.name,

this.temp,

this.main,

this.icon,

this.minTemp,

this.maxTemp,

this.pressure,

this.humidity,

});

}

*DaysForecast.dart*  освен модели за прогноз на седмица, дефинира фабрика за преобразуване JSON формат на данни към вътрешни модели

import './OneDatyForecast.dart';

class DaysForecast {

final List list;

final String city;

DaysForecast({this.list, this.city});

factory DaysForecast.fromJson(Map<String, dynamic> json) {

List list = new List();

String city = json['city']['name'];

for (dynamic e in json['list']) {

OneDayForecast w = new OneDayForecast(

date: new DateTime.fromMillisecondsSinceEpoch(e['dt'] \* 1000, isUtc: false),

name: json['city']['name'],

temp: (e['temp']['day'].toDouble() - 273.15),

main: e['weather'][0]['main'],

icon: e['weather'][0]['icon'],

minTemp: e['temp']['min'].toDouble() - 273.15,

maxTemp: e['temp']['max'].toDouble() - 273.15,

humidity: e['humidity'].toDouble(),

pressure: e['pressure'].toDouble()

);

list.add(w);

}

return DaysForecast(

list: list,

city: city,

);

}

}

*LandingPage.dart* Зарежда прогноз на времето

loadData() async {

setState(() {

isLoading = true;

});

Map<String, double> location;

try {

location = await \_location.getLocation();

error = null;

} on PlatformException catch (e) {

if (e.code == 'PERMISSION\_DENIED') {

error = 'Permission denied';

} else if (e.code == 'PERMISSION\_DENIED\_NEVER\_ASK') {

error = 'Permission denied - please ask user';

location = null;

}

}

if (location != null) {

// final lat = 40.730610;

// final lon = -73.935242;

final lat = location['latitude'];

final lon = location['longitude'];

final url =

'https://api.openweathermap.org/data/2.5/forecast/daily?APPID=$OWM\_ID&lat=${lat.toString()}&lon=${lon.toString()}&cnt=7';

final forecastResponse = await http.get(url);

if (forecastResponse.statusCode == 200) {

setState(() {

forecast =

new DaysForecast.fromJson(jsonDecode(forecastResponse.body));

isLoading = false;

});

Navigator.of(context).pushReplacement(

new MaterialPageRoute(builder: (BuildContext context) {

return new FiveDaysForecastPage(forecast);

}));

}

}

}