import 'package:flutter/material.dart';

void main() {

runApp(MyApp());

}

class MyApp extends StatelessWidget {

@override

Widget build(BuildContext context) {

return MaterialApp(

title: 'HollyStory',

theme: ThemeData(

primarySwatch: Colors.pink,

),

home: ActorPhotosPage(),

);

}

}

class ActorPhotosPage extends StatelessWidget {

final List<ActorInfo> actors = [

ActorInfo(name: 'Angelina Jolie', image: '../imagens/angelina.PNG'),

ActorInfo(name: 'Brad Pitt', image: '../imagens/brad.PNG'),

ActorInfo(name: 'Leonardo DiCaprio', image: '../imagens/leo.PNG'),

ActorInfo(name: 'Jennifer Aniston', image: '../imagens/jenni.PNG'),

];

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: Text('Atores e Atrizes'),

),

body: GridView.builder(

gridDelegate: SliverGridDelegateWithFixedCrossAxisCount(

crossAxisCount: 2,

),

itemCount: actors.length,

itemBuilder: (context, index) {

return GestureDetector(

onTap: () {

Navigator.of(context).push(

PageRouteBuilder(

pageBuilder: (context, animation1, animation2) {

return AgeCalculatorPage(

actor: actors[index],

);

},

transitionsBuilder: (context, animation1, animation2, child) {

var begin = Offset(1.0, 0.0);

var end = Offset.zero;

var curve = Curves.easeInOut;

var tween = Tween(begin: begin, end: end).chain(CurveTween(curve: curve));

var offsetAnimation = animation1.drive(tween);

return SlideTransition(

position: offsetAnimation,

child: child,

);

},

transitionDuration: Duration(milliseconds: 500),

),

);

},

child: Card(

child: Column(

mainAxisAlignment: MainAxisAlignment.center,

children: [

Hero(

tag: 'actorImage${actors[index].name}',

child: Image.asset(

actors[index].image,

width: 100,

height: 100,

fit: BoxFit.cover,

),

),

SizedBox(height: 8),

Text(actors[index].name),

],

),

),

);

},

),

);

}

}

class ActorInfo {

final String name;

final String image;

ActorInfo({required this.name, required this.image});

}

class AgeCalculatorPage extends StatefulWidget {

final ActorInfo actor;

AgeCalculatorPage({required this.actor});

@override

\_AgeCalculatorPageState createState() => \_AgeCalculatorPageState();

}

class \_AgeCalculatorPageState extends State<AgeCalculatorPage> {

final \_formKey = GlobalKey<FormState>();

int? \_userAge;

int? \_actorAge;

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: Text('Calculadora de Idade'),

),

body: Padding(

padding: const EdgeInsets.all(16.0),

child: Form(

key: \_formKey,

child: Column(

crossAxisAlignment: CrossAxisAlignment.stretch,

mainAxisAlignment: MainAxisAlignment.center,

children: <Widget>[

Hero(

tag: 'actorImage${widget.actor.name}',

child: Image.asset(

widget.actor.image,

width: 150,

height: 150,

fit: BoxFit.cover,

),

),

SizedBox(height: 16),

Text(

'Qual é a sua idade e a idade de ${widget.actor.name}?',

style: TextStyle(fontSize: 18),

textAlign: TextAlign.center,

),

SizedBox(height: 16),

TextFormField(

keyboardType: TextInputType.number,

decoration: InputDecoration(labelText: 'Sua idade'),

validator: (value) {

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

return 'Insira sua idade';

}

return null;

},

onSaved: (value) {

\_userAge = int.tryParse(value!) ?? 0;

},

),

TextFormField(

keyboardType: TextInputType.number,

decoration: InputDecoration(labelText: 'Idade de ${widget.actor.name}'),

validator: (value) {

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

return 'Insira a idade de ${widget.actor.name}';

}

return null;

},

onSaved: (value) {

\_actorAge = int.tryParse(value!) ?? 0;

},

),

SizedBox(height: 12),

ElevatedButton(

onPressed: () {

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

\_formKey.currentState!.save();

Navigator.of(context).push(

MaterialPageRoute(

builder: (context) => AgeCalculatorResultPage(

userAge: \_userAge!,

actorAge: \_actorAge!,

actor: widget.actor,

),

),

);

}

},

child: Text('Calcular Diferença de Idade'),

),

],

),

),

),

);

}

}

class AgeCalculatorResultPage extends StatelessWidget {

final int userAge;

final int actorAge;

final ActorInfo actor;

AgeCalculatorResultPage({required this.userAge, required this.actorAge, required this.actor});

@override

Widget build(BuildContext context) {

int ageDifference = (userAge - actorAge).abs();

return Scaffold(

appBar: AppBar(

title: Text('Resultado da Calculadora de Idade'),

),

body: Center(

child: Column(

mainAxisAlignment: MainAxisAlignment.center,

children: <Widget>[

Text(

'Sua idade: $userAge anos',

style: TextStyle(fontSize: 18),

),

Text(

'Idade de ${actor.name}: $actorAge anos',

style: TextStyle(fontSize: 18),

),

SizedBox(height: 12),

Text(

'Diferença de idade: $ageDifference anos',

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

),

],

),

),

);

}

}