

ГУАП

КАФЕДРА № 41

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ
ПРЕПОДАВАТЕЛЬ

Доцент, к.т.н.

должность, уч. степень, звание

подпись, дата

С. А. Чернышев

инициалы, фамилия

ОТЧЕТ О ПРАКТИЧЕСКОЙ РАБОТЕ №1

«Проектирование GUI кроссплатформенных приложений»

по курсу: «Методы объектно-ориентированного проектирования»

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ гр. №

подпись, дата

инициалы, фамилия

Санкт-Петербург 2024

1. Цель работы

Реализовать верстку приложения на основе разработанного макета.

2. Выполнение работы

2.1. Описание приложения

Приложение по аренде автомобилей. Содержит следующие экраны:

- 1) Экран загрузки приложения;
- 2) Экран авторизации пользователя;
- 3) Экран регистрации пользователя;
- 4) Главный экран, на котором представлен список акций и блоки: с популярными автомобилями, всеми автомобилями;
- 5) Экран с детальной информацией об автомобиле;
- 6) Экран профиля;
- 7) Экран редактирования данных.

2.2. Дизайн

Дизайн приложения представлен по ссылке:

<https://www.figma.com/design/N5XJsNPI3qzT12kt8hUmCK/Flutter-App?node-id=3-46612&t=9niEtllrtkGB1JeI-1>

Ниже, на рисунках 1 – 7 представлены экраны мобильной версии приложения.

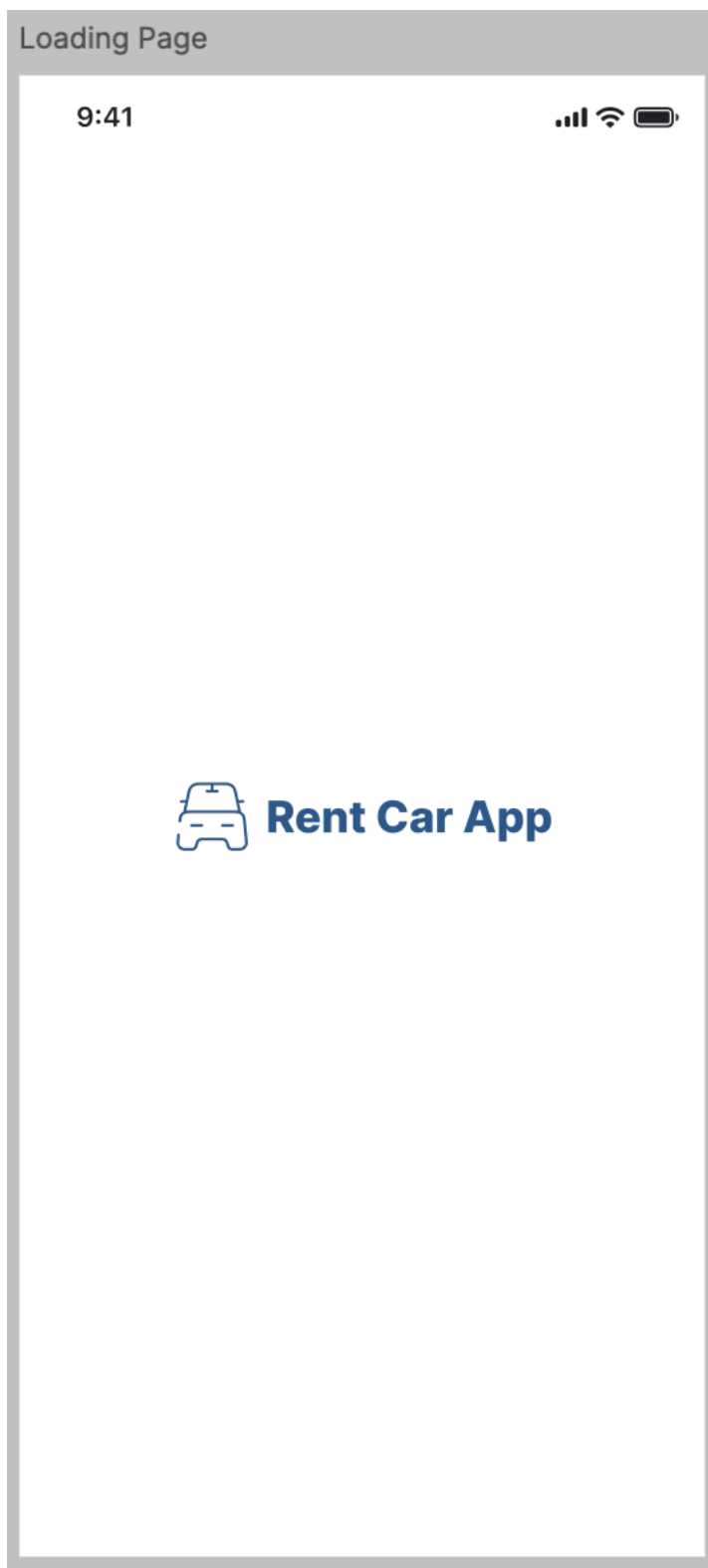


Рисунок 1 – Экран загрузки

9:41



Welcome!
Login to your account,
Or create new one



[forgot password?](#)

Login




or login with




Don't have an account? [Register](#)

Рисунок 2 – Экран авторизации


Register Page


9:41   

 **Register**

Full name

Email adress

Password 

Confirm password 

Register

Already have an account? [Login](#)

Рисунок 3 – Экран регистрации

9:41



Your location ▾

Moscow, Russia



New car in rent - M5 F90



Popular cars

[see all](#)



BMW M5 F90, 2022
★ 4.8 (100+ reviews)

22 000 ₺ / day



Porsche 911 Turbo S, 2022
★ 4.8 (100+ reviews)

40 000 ₺ / day



BM
★

22

All cars



Porsche 911 Turbo S, 2022
★ 4.8 (100+ reviews)

40 000 ₺ / day



BMW M5 F90, 2022
★ 4.8 (100+ reviews)

22 000 ₺ / day



BMW M5 F90, 2022
★ 4.8 (100+ reviews)



Porsche 911 Turbo S, 2022
★



Рисунок 4 – Главный экран



BMW M5 F90, 2022

★ 4.8 (100+ reviews)

Experience the ultimate combination of luxury, power, and precision with the BMW M5 F90. This high-performance sports sedan features a twin-turbocharged 4.4L V8 engine, delivering an exhilarating 600 horsepower for an unforgettable drive. Equipped with BMW's signature xDrive all-wheel-drive system, the M5 F90 ensures both stability and thrilling speed in any road condition.

Inside, enjoy a sophisticated, tech-forward interior with premium leather seating, a state-of-the-art infotainment system, and advanced safety features. Whether you're navigating city streets or taking a weekend getaway, the BMW M5 F90 offers unparalleled comfort and driving dynamics.

Book today and elevate your driving experience with the power and luxury of the M5 F90!

CAR INFO

Engine	4.4L twin-turbo V8
Power	600
Fuel	Gasoline
Color	Dark Blue
Drivetrain	xDrive All-Wheel Drive

BMW M5 F90, 2022

22 000 ₺ / day

Rent Car

Рисунок 5 – Экран с детальной информацией об автомобиле

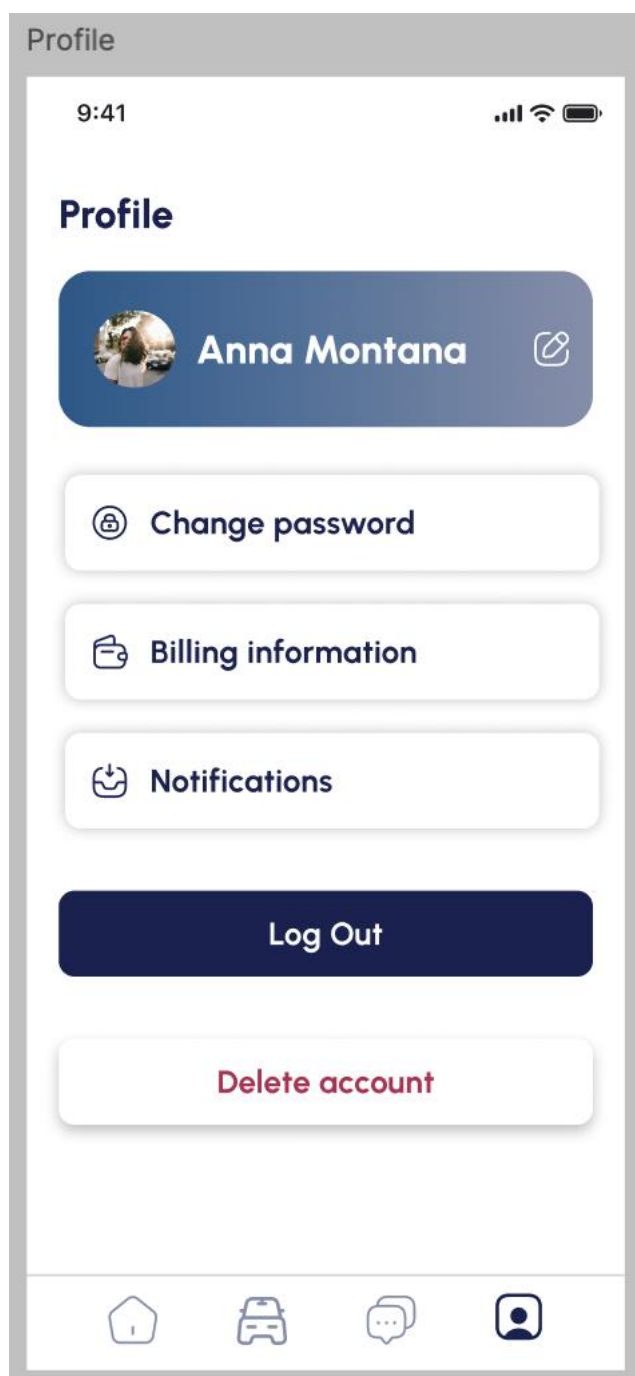


Рисунок 6 – Экран профиля

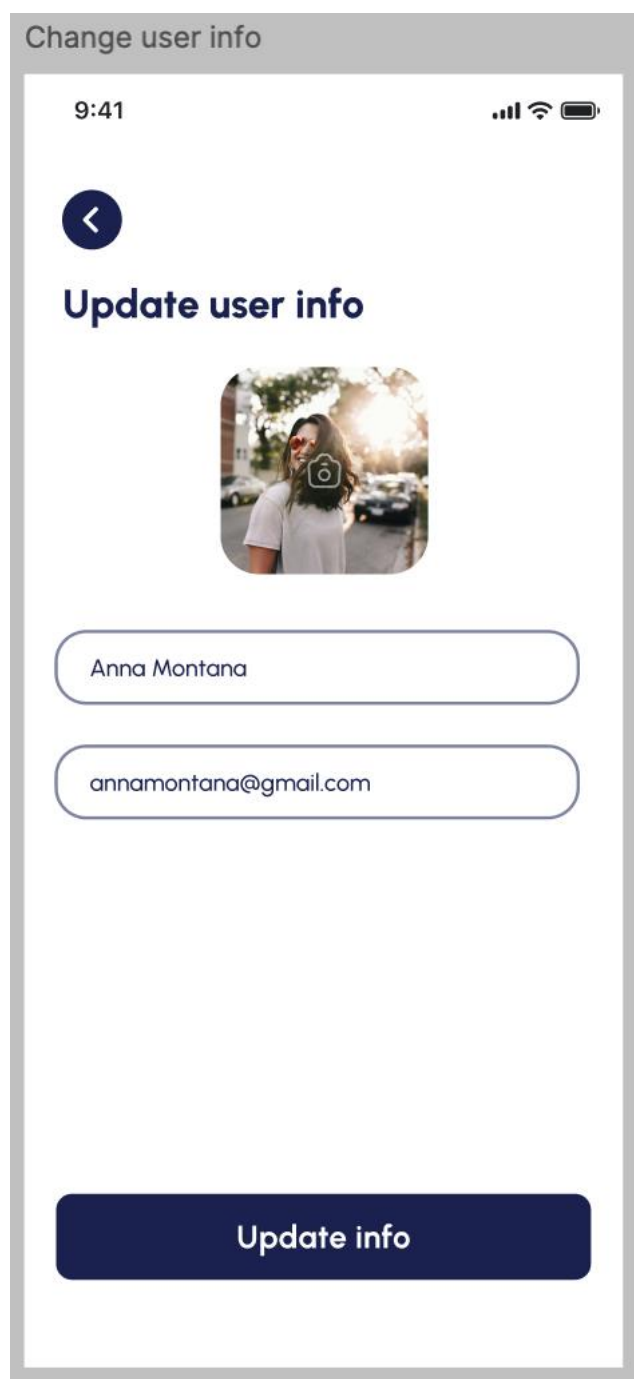


Рисунок 7 – Экран редактирования профиля

На рисунках 8 – 13 представлены экраны десктопной версии приложения.

Login Desktop



Welcome!
Login to your account.
Or create new one

Email address

Password

Login

or login with

Don't have an account? Register

Рисунок 8 – Экран авторизации

Register Desktop

← Register

Full name

Email address

Password

Confirm password

Register

Already have an account? Login

Рисунок 9 – Экран регистрации

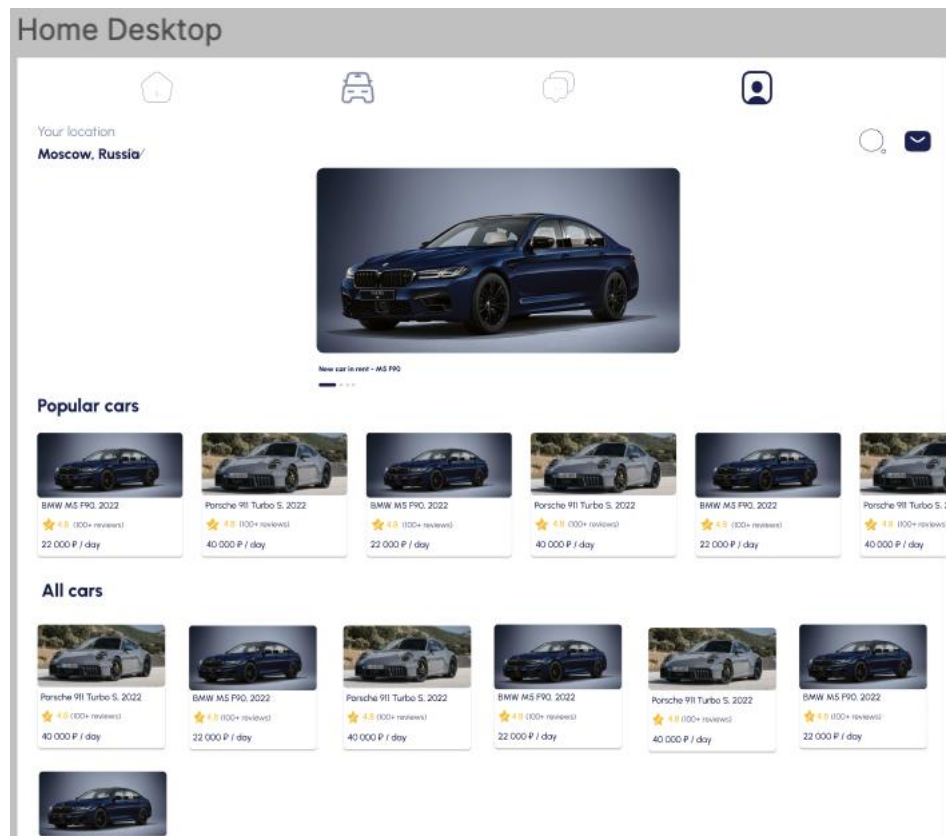


Рисунок 10 – Главный экран



Рисунок 11 – Экран с детальной информацией об автомобиле

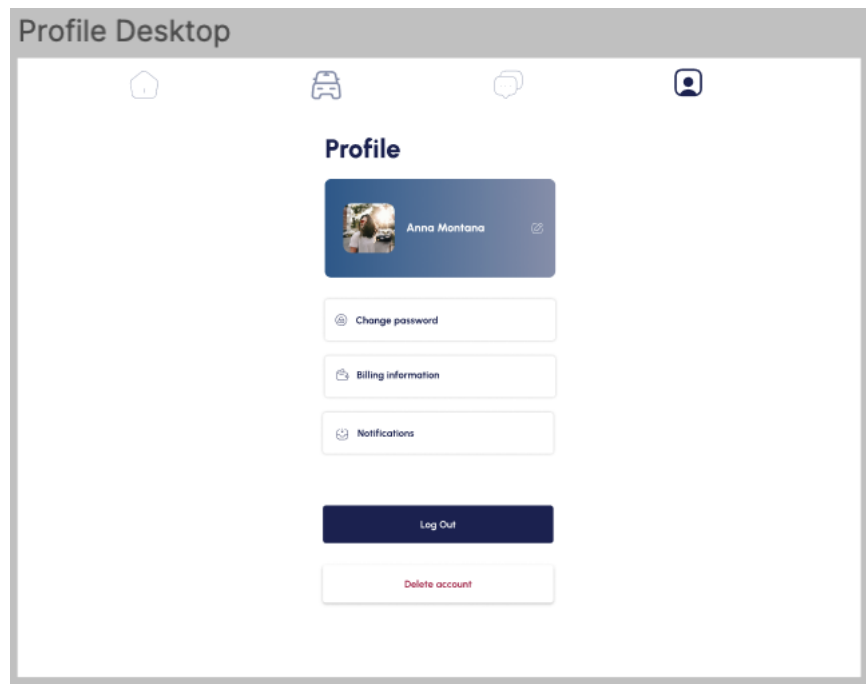


Рисунок 12 – Экран профиля

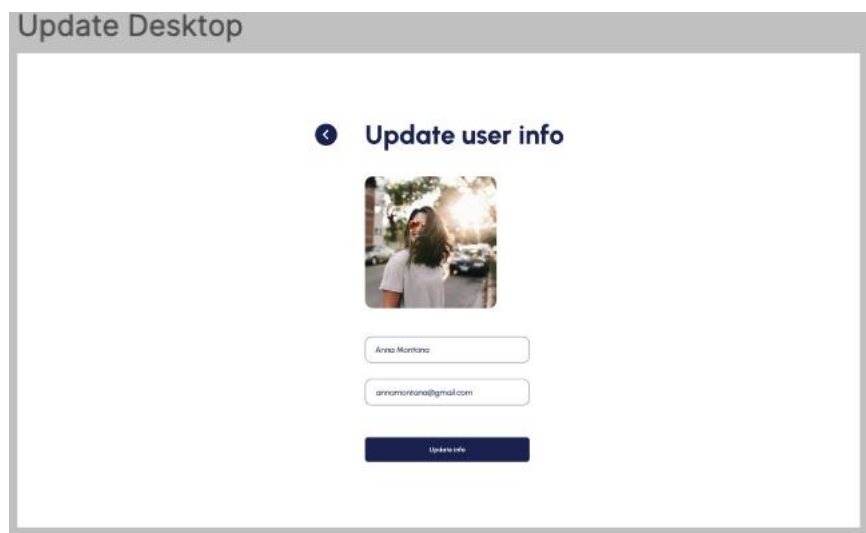


Рисунок 13 – Экран изменения данных пользователя

2.3. Описание этапов верстки

Для разработки приложения на Flutter будет использоваться IDE VSCode и установленные в него библиотеки Dart и Flutter. Создадим проект при помощи команды Flutter: New Project, дадим ему название rent_car_project. В папке lib создадим папку Pages в которой будут храниться наши экраны. В папке Pages создадим следующие файлы с расширением dart: detail_page, edit_user_page, home_page, loading_screen, login_page, profile_page, register_page.

2.3.1. Верстка экрана загрузки

Начнем верстку `loading_screen`, она состоит из иконки приложения и текста с названием приложения «Rent Car App». Для начала в файл импортируем библиотеку Material Design, которая предоставляет основные виджеты для построения пользовательского интерфейса.

Листинг 1 – Импорт библиотеки

```
import 'package:flutter/material.dart';
```

Далее определим виджет `LoadingScreen`. Для этого нам необходимо создать класс `LoadingScreen` как `StatelessWidget`, так как верстка статистическая и не будет изменяться со временем и в нем создадим конструктор класса.

Листинг 2 – Создание класса

```
class LoadingScreen extends StatelessWidget {  
  const LoadingScreen({super.key});  
}
```

Далее внутри класса переопределим метод `build`, где `@override` указывает, что метод `build` переопределяется из родительского класса `StatelessWidget`, а `build` – метод, который возвращает интерфейс виджета. Далее перейдем к созданию структуры экрана, для этого во Flutter есть виджет `Scaffold`, где `body` – основная область содержимого. Содержимое страницы будет помещено в виджет `Container`, у которого в качестве цвета заднего фона будет установлен белый цвет, а в содержимое будет центрировано по вертикали и горизонтали при помощи виджета `Center`. Так как нам надо расположить элементы в горизонтальном формате, то для этого необходимо воспользоваться виджетом `Row`, внутри которого мы центрируем внутренние элементы по главной оси, в данном случае – по горизонтали, при помощи свойства `mainAxisAlignment` и передачи в него параметра `MainAxisAlignment.center`.

Листинг 3 – Создание структуры экрана

```
@override  
Widget build(BuildContext context) {  
  return Scaffold( // Основной каркас экрана  
    body: Container( // Контейнер для фона  
      color: Colors.white, // Белый фон  
      child: Center( // Центрирование содержимого по вертикали и горизонтали
```

```

child: Row( // Горизонтальное расположение элементов
  mainAxisAlignment: MainAxisAlignment.center, // Центрируем элементы по горизонтали
  children: [
    ],
  ),
),
),
);
}

```

Теперь добавим логотип нашего приложения в Row. Для начала создадим папку в корне проекта assets, а в ней папку images. Добавим в images логотип приложения. После этого необходимо перейти в файл pubspec.yaml, найти и раскомментировать поле assets и добавить туда путь до нашего логотипа.

Листинг 4 – Добавление пути в pubspec.yaml

```

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:
  - assets/images/car.png

```

После – возвращаемся в файл с экраном и добавляем логотип при помощи виджета Image.asset(), в свойства которого передаем путь до нашего изображения, размер изображения (width и height), цвет изображения (color), а также установим наложение цвета на изображение (colorBlendMode).

Листинг 5 – Виджет изображения

```

Image.asset(
  'assets/images/car.png', // Путь к изображению
  width: 50, // Ширина изображения
  height: 50, // Высота изображения
  color: const Color(0xFF1B588C), // Синий цвет
  colorBlendMode: BlendMode.srcATop, // Наложение цвета
),

```

Добавим отступ между изображением и текстом (названием приложения) при помощи виджета SizedBox(width: 10).

Листинг 6 – Виджет отступа

```

const SizedBox(width: 10), // Отступ между изображением и текстом

```

Теперь добавим название нашего приложения, для этого воспользуемся виджетом Text, в свойства которого передадим название и стиль текста. Установим стиль текста при помощи TextStyle:

- Размер (fontSize) – 24;
- Толщину (fontWeight) – FontWeight.w900 (очень жирный шрифт);
- Цвет текста (color);

Листинг 7 – Виджет текста

```
const Text(  
  "Rent Car App", // Текст  
  style: TextStyle(  
    fontSize: 24, // Размер шрифта  
    fontWeight: FontWeight.w900, // Жирный шрифт  
    color: Color(0xFF1B588C), // Синий цвет текста  
  ),  
),
```

Листинг 8 - Полный код Loading Screen

```
import 'package:flutter/material.dart'; // Подключение библиотеки Material Design  
  
// Создание виджета LoadingScreen как StatelessWidget  
class LoadingScreen extends StatelessWidget {  
  const LoadingScreen({super.key}); // Конструктор класса  
  
  // Переопределение метода build для построения интерфейса  
  @override  
  Widget build(BuildContext context) {  
    return Scaffold( // Основной каркас экрана  
      body: Container( // Контейнер для фона  
        color: Colors.white, // Белый фон  
        child: Center( // Центрирование содержимого по вертикали и горизонтали  
          child: Row( // Горизонтальное расположение элементов  
            mainAxisAlignment: MainAxisAlignment.center, // Центрируем элементы по горизонтали  
            children: [  
              // Логотип автомобиля  
              Image.asset(  
                'assets/images/car.png', // Путь к изображению  
                width: 50, // Ширина изображения  
                height: 50, // Высота изображения  
                color: const Color(0xFF1B588C), // Синий цвет  
                colorBlendMode: BlendMode.srcATop, // Наложение цвета  
              ),  
              const SizedBox(width: 10), // Отступ между изображением и текстом  
              // Название приложения  
              const Text(  
                "Rent Car App", // Текст  
                style: TextStyle(  
                  fontSize: 24, // Размер шрифта  
                  fontWeight: FontWeight.w900, // Жирный шрифт
```

```
color: Color(0xFF1B588C), // Синий цвет текста
),
),
1,
),
),
),
);
}
}
```

На рисунке 14 представлен результат верстки экрана в симуляторе iPhone.



Рисунок 14 – Верстка экрана Loading Screen в симуляторе iPhone

2.3.2. Верстка экрана авторизации

Создаем файл `login_page.dart`. Добавляем в него необходимые библиотеки, аналогично пункту 2.3.1 – Листинг 9

Листинг 9 – Импорт необходимых библиотек

```
import 'package:flutter/material.dart';
import 'package:flutter/foundation.dart';
```



```
import 'dart:io';
```

Аналогично пункту 2.3.1 создаем класс экрана и его каркас – Листинг 10

Листинг 10 – Создание класса и каркаса

```
class LoginPage extends StatefulWidget {  
  const LoginPage({super.key});  
  
  @override  
  _LoginPageState createState() => _LoginPageState();  
}  
  
class _LoginPageState extends State<LoginPage> {  
  @override  
  void initState() {  
    super.initState();  
  }  
  
  @override  
  Widget build(BuildContext context) {  
    return Scaffold(  
      body: Container(  
        color: Colors.white,  
        child: // Остальной код,  
      ),  
    );  
  }  
}
```

При помощи виджета `Padding` и свойства `EdgeInsets.all` добавим отступы по всем краям в 20px – и полученный виджет в контейнер – Листинг 11.

Листинг 11 – Добавление отступов по всем краям

```
@override  
Widget build(BuildContext context) {  
  return Scaffold(  
    body: Container(  
      color: Colors.white,  
      child: Padding(  
        padding: const EdgeInsets.all(20.0),  
        child: // Остальной код  
      ),  
    ),  
  );  
}
```

В зависимости от устройства контент экрана будет либо помещен в прокручиваемый контейнер или нет. Если устройство не мобильное – то содержимое необходимо поместить в виджет `SingleChildScrollView`, для этого необходимо внутри создания страницы добавить логическую переменную `isMobile`, в которой будет проверяться тип устройства – листинг 12.

Листинг 12 – Создание логической переменной и помещение содержимого в ScrollView

```
@override
Widget build(BuildContext context) {
  bool isMobile = !kIsWeb && (Platform.isAndroid || Platform.isIOS);
  return Scaffold(
    body: Container(
      color: Colors.white,
      child: Padding(
        padding: const EdgeInsets.all(20.0),
        child: isMobile
          ? // Содержимое
            : SingleChildScrollView(
                child: // Содержимое,
              ),
      ),
    ),
  );
}
```

Перейдем теперь к созданию содержимого страницы. Оно у нас состоит из приветственного сообщения, двух текстовых полей, кнопок авторизации, забыли пароль, регистрации и альтернативных способов входа. Так как контент будет один и тот же что на мобильных устройствах, что на десктоп версии, то создадим переменную `content` с типом `Widget`, которая будет представлять собой вертикальный стек - `Column`, в котором все элементы по вертикальной оси будут находиться по центру – `mainAxisAlignment: MainAxisAlignment.center`, а вдоль горизонтальной оси по началу горизонтали – `crossAxisAlignment: CrossAxisAlignment.start` – Листинг 13.

Листинг 13 – создание переменной `content`

```
@override
Widget build(BuildContext context) {

  bool isMobile = !kIsWeb && (Platform.isAndroid || Platform.isIOS);

  Widget content = Column(
    mainAxisAlignment: MainAxisAlignment.center,
    crossAxisAlignment: CrossAxisAlignment.start,
    children: []
  )

  return Scaffold(
    body: Container(
      color: Colors.white,
      child: Padding(
        padding: const EdgeInsets.all(20.0),
```

```

        child: isMobile
          ? content
          : SingleChildScrollView(
              child: content,
            ),
      ),
    ),
  );
}

```

Начнем заполнение вертикального стека элементами. В начале создадим приветственный текст при помощи виджета Text, в который передадим приветственное сообщение, в свойство style передадим стиль для нашего текста при помощи TextStyle(), где установим размер шрифта 24, вес – жирный при помощи FontWeight.bold и цвет в формате hex при помощи Color(0xFF), где после FF будет идти цвет в hex формате. После чего обернем текст в Padding и добавим отступ сверху в 82px и снизу в 62 при помощи EdgeInsets.fromLTRB() – Листинг 14 – Создание приветственного сообщения.

Листинг 14 – Создание приветственного сообщения

```

    Padding(
      padding: EdgeInsets.fromLTRB(0, 82, 0, 62),
      child: Text(
        S.of(context).welcomeLogin,
        style: TextStyle(
          fontSize: 24,
          fontWeight: FontWeight.bold,
          color: Color(0xFF1B588C),
        ),
      ),
    ),
  ),
)

```

Так как текстовое поле будет использоваться не только на экране авторизации, но и экране регистрации, изменения данных о пользователе, то логично вынести его в отдельный компонент, который мы сможем легко использовать в дальнейшем. В папке lib создаем еще одну папку с названием Components и создаем там файл text_field.dart. Импортируем Material Design библиотеку в данный файл и создаем класс CustomTextField как StatelessWidget. Пропишем необходимые параметры, которые будет принимать виджет:

- labelText типа String – название поля, например email;
- isObscure типа bool – скрытое ли данное поле (пароль или нет);

После чего создадим конструктор для нашего класса – Листинг 15

Листинг 15 – Создание конструктора класса

```
const CustomTextField({  
  Key? key,  
  required this.labelText,  
  this.isObscure = false  
}) : super(key: key);
```

Далее создаем каркас класс, аналогично предыдущим экранам при помощи override Widget build(), который будет возвращать виджет TextField. Установим свойство obscureText как isObscure, после чего установим стиль текста внутри TextField при помощи style и TextStyle, в который передадим цвет текста в hex формате при помощи Color(0xFF...), а также установим вес текста при помощи FontWeight.w400. Далее настроим внешний вид нашего текстового поля при помощи свойства decoration. Установим его как InputDecoration, в который передадим:

- labelText: labelText – отобразим название поля, например email;
- labelStyle – зададим стиль текста названия поля при помощи TextStyle с установленными параметрами цвета и размера текста;
- enableBorder и focusBorder – определим стили границ текстового поля в различных состояниях, когда поле неактивно (enableBorder) и активно (focusBorder). Для этого воспользуемся OutlineInputBorder(), где зададим при помощи BorderSide цвет и толщину границ (окантовки) текстового поля, а также установим закругление для нашего текстового поля при помощи BorderRadius.circular().

В листинге 16 представлен полный код кастомного текстового поля.

Листинг 16 – Кастомное текстовое поле.

```
import 'package:flutter/material.dart';  
  
class CustomTextField extends StatelessWidget {  
  final String labelText;  
  final bool isObscure;  
  
  const CustomTextField({  
    Key? key,  
    required this.labelText,  
    this.isObscure = false,  
  }) : super(key: key);
```

```

@override
Widget build(BuildContext context) {

  return TextField(
    obscureText: isObscure,
    style: const TextStyle(
      color: Color(0xFF192252),
      fontWeight: FontWeight.w400
    ),
    decoration: InputDecoration(
      labelText: labelText,
      labelStyle: TextStyle(
        color: const Color(0xFF424F7B).withOpacity(0.7),
        fontSize: 16,
      ),
    ),
    enabledBorder: OutlineInputBorder(
      borderSide: BorderSide(
        color: const Color(0xFF424F7B).withOpacity(0.7),
        width: 2.0,
      ),
    ),
    borderRadius: BorderRadius.circular(20.0),
  ),
    focusedBorder: OutlineInputBorder(
      borderSide: BorderSide(
        color: const Color(0xFF424F7B).withOpacity(0.7),
        width: 2.0,
      ),
    ),
    borderRadius: BorderRadius.circular(20.0),
  ),
);
}
}

```

Теперь вернемся в login_page.dart. Для того, чтобы воспользоваться созданным текстовым полем необходимо там, где происходит добавление библиотек прописать путь до нашего файла с полем – Листинг 17.

Листинг 17 – Добававление кастомного текстового поля в login_page.dart

```

import 'package:flutter/material.dart';
import '../Components/text_field.dart';
import 'package:flutter/foundation.dart';
import 'dart:io';

```

Теперь, после созданного текста с приветственным сообщением создаем текстовое поле – email при помощи CustomTextField(), в который передаем название поля (labelText) – email. После созданного поля добавим отступ в 25px при помощи SizedBox(height:25) и создадим еще одно текстовое, но уже

для пароля, дополнительно к labelText со значением password добавляем еще один параметр isObscure и устанавливаем его в true - Листинг 18.

Листинг 18 – Добавление текстовых полей на экран.

```
CustomTextField(  
  labelText: "Email address"  
)  
,  
const SizedBox(height: 25),  
CustomTextField(  
  labelText: "Password",  
  isObscure: true,  
)
```

После чего необходимо добавить кнопку «Forgot password?». Для этого воспользуемся виджетом Align, так как необходимо выровнять кнопку по правому краю – для этого в параметр alignment установим Alignment.centerRight. В параметр child установим кнопку TextButton(), действие оставим пустым при помощи: () {}, а child установим в виде Текста, настроим его внешний вид при помощи TextStyle. Так как тут используется свой шрифт, то необходимо скачать шрифт, в папку assets добавить папку fonts, после чего в файле pubspec.yaml раскомментировать блок с настройкой своих шрифтов – Листинг 19 и 20 соответственно.

Листинг 19 – Добавление шрифта в pubspec.yaml

```
fonts:  
  - family: Urbanist  
  fonts:  
    - asset: assets/fonts/Urbanist-VariableFont_wght.ttf
```

Листинг 20 – Создание кнопки «Forgot password?»

```
Align(  
  alignment: Alignment.centerRight,  
  child: TextButton(  
    onPressed: () {},  
    child: Text(  
      "Forgot password?",  
      style: TextStyle(  
        color: const Color(0xFF424F7B).withOpacity(0.7),  
        fontFamily: "Urbanist",  
        fontSize: 12,  
        fontWeight: FontWeight.w600,  
      ),  
    ),  
  ),  
)
```

Создадим кнопку входа при помощи `ElevatedButton`. Действие оставим пустым при помощи `() {}`, настроим стиль кнопки при помощи `ElevatedButton.styleFrom()`:

- Установим цвет кнопки в параметр `backgroundColor`;
- Форму кнопки как прямоугольник с закругленными углами при помощи `RoundedRectangleBorder` и радиусом закругления в `8px`;
- Минимальный размер сделаем так, чтобы он занимал все доступное пространство по ширине, а высоту `50px` при помощи `Size(double.infinity, 50)`;
- Установим текст кнопки «Login» и настроим его стиль.

После чего обернем созданную кнопку в `Padding` и добавим отступ сверху в `9px` и снизу в `53px` – листинг 21.

Листинг 21 – Создание кнопки «Войти».

```
Padding(  
  padding: EdgeInsets.fromLTRB(0, 9, 0, 53),  
  child: ElevatedButton(  
    onPressed: () {},  
    style: ElevatedButton.styleFrom(  
      backgroundColor: const Color(0xFF192252),  
      shape: RoundedRectangleBorder(  
        borderRadius: BorderRadius.circular(8),  
      ),  
      minimumSize: const Size(double.infinity, 50),  
    ),  
    child: Text(  
      "Login",  
      style: TextStyle(  
        fontFamily: "Urbanist",  
        fontSize: 25,  
        color: Colors.white,  
        fontWeight: FontWeight.w600,  
      ),  
    ),  
  ),  
),  
)
```

Далее, создадим блок «or login with». Для этого воспользуемся виджетом `Row`, в который поместим разделитель при помощи виджета `Divider()` и установим его цвет. После чего обернем его в виджет `Expanded`, чтобы он мог занять все доступное пространство. После чего создадим при помощи виджета

Text() строку «or login with», настроим стиль текста, а после обернем его в виджет Padding() и установим отступы по горизонтали при помощи EdgeInsets.symmetric(horizontal: 8). Далее добавляем еще один отступ, обернутый в виджет Expanded. Листинг созданного блока представлен в листинге 22.

Листинг 22 – Создание блока «or login with»

```
Row(  
  children: [  
    Expanded(  
      child: Divider(  
        color: const Color(0xFF424F7B).withOpacity(0.5),  
      ),  
    ),  
    Padding(  
      padding: EdgeInsets.symmetric(horizontal: 8.0),  
      child: Text(  
        "or login with",  
        style: TextStyle(  
          color: Color(0xFF424F7B),  
          fontFamily: "Urbanist",  
          fontSize: 20,  
        ),  
      ),  
    ),  
    Expanded(  
      child: Divider(  
        color: const Color(0xFF424F7B).withOpacity(0.5),  
      ),  
    ),  
  ],  
)
```

Теперь добавим отступ в 20px при помощи SizedBox(height: 20). После которого создадим блок с иконками google и ВК. Для этого для начала необходимо добавить иконки в проект – делаем это аналогично пункту 2.3.1, листинг 4 и 5. После чего воспользуемся виджетом Row, установим внутри него выравнивание по центру горизонтальной оси и создадим иконки при помощи IconButton(), в icon передадим виджет Image.asset(путь до нашей иконки), а действие: () {}. Обернем иконку в SizedBox, чтобы их размер был постоянным, а именно 50x50. Между иконками добавим отступ в 20px при помощи sizedBox(width: 20) – Листинг 23.

Листинг 23 – Создание блока с иконками


```

Row(
  mainAxisAlignment: MainAxisAlignment.center,
  children: [
    SizedBox(
      width: 50,
      height: 50,
      child: IconButton(
        icon: Image.asset('assets/images/google_icon.png'),
        onPressed: () {},
      ),
    ),
    const SizedBox(width: 20),
    SizedBox(
      width: 50,
      height: 50,
      child: IconButton(
        icon: Image.asset('assets/images/vk_icon.png'),
        onPressed: () {},
      ),
    ),
  ],
),

```

Далее – для правильного расположения остальных элементов при помощи проверки `isMobile` добавляем либо виджет `Spacer()`, либо отступ в 100px. После которого создаем кнопку «Don't have an account? Register». Для этого воспользуемся виджетом `TextButton()`, где действие установим как `() {}`, а в качестве `child` установим виджет `RichText`, так как «Register» должен быть более выделенным, чем остальной текст в предложении. В `RichText` передаем `TextSpan`, в который в `text` устанавливаем основной текст «Don't have an account?"» и устанавливаем его стиль при помощи `TextStyle`. В `children` передаем массив дочерних `TextSpan`, в нашем случае – Register. Для этого в массив добавляем `TextSpan()`, в который передаем «Register» и настраиваем его стиль при помощи `TextStyle()`. После чего, всю кнопку необходимо обернуть в `Center()`, чтобы выравнивать ее посередине экрана – Листинг 24.

Листинг 24 – Создание текста «Don't have an account? Register»

```

isMobile
? const Spacer()
: const SizedBox(height: 100),
Center(
  child: TextButton(
    onPressed: () {},
    child: RichText(
      text: TextSpan(

```

```
text: "Don't have an account? ",
style: TextStyle(
  fontFamily: "Urbanist",
  color: Color(0xFF424F7B),
  fontSize: 18,
),
children: <TextSpan>[
  TextSpan(
    text: "Register",
    style: TextStyle(
      fontFamily: "Urbanist",
      color: Color(0xFF103F74),
      fontWeight: FontWeight.bold,
    ),
  ),
],
),
),
),
),
),
),
```

В листинге 25 представлен полный код экрана `login_page.dart`

Листинг 25 – Полный код экрана login_page.dart

```
import 'package:flutter/material.dart';
import '../Components/text_field.dart';
import 'package:flutter/foundation.dart';
import 'dart:io';

class LoginPage extends StatefulWidget {
  const LoginPage({super.key});

  @override
  _LoginPageState createState() => _LoginPageState();
}

class _LoginPageState extends State<LoginPage> {

  @override
  void initState() {
    super.initState();
  }

  @override
  Widget build(BuildContext context) {
    bool isMobile = !kIsWeb && (Platform.isAndroid || Platform.isIOS);

    Widget content = Column(
      mainAxisAlignment: MainAxisAlignment.center,
      crossAxisAlignment: CrossAxisAlignment.start,
      children: [
        Padding(
          padding: EdgeInsets.fromLTRB(0, 82, 0, 62),
          child: Text(
            "Welcome!\nLogin to your account,\nOr create new one",

```

```

style: TextStyle(
  fontSize: 24,
  fontWeight: FontWeight.bold,
  color: Color(0xFF1B588C),
),
),
),
CustomTextField(
  labelText: "Email address"
),
const SizedBox(height: 25),
CustomTextField(
  labelText: "Password",
  isObscure: true,
),
Align(
  alignment: Alignment.centerRight,
  child: TextButton(
    onPressed: () {},
    child: Text(
      "Forgot password?",
      style: TextStyle(
        color: const Color(0xFF424F7B).withOpacity(0.7),
        fontFamily: "Urbanist",
        fontSize: 12,
        fontWeight: FontWeight.w600,
      ),
    ),
  ),
),
),
),
),
Padding(
  padding: EdgeInsets.fromLTRB(0, 9, 0, 53),
  child: ElevatedButton(
    onPressed: () {},
    style: ElevatedButton.styleFrom(
      backgroundColor: const Color(0xFF192252),
      shape: RoundedRectangleBorder(
        borderRadius: BorderRadius.circular(8),
      ),
    ),
    minimumSize: const Size(double.infinity, 50),
  ),
  child: Text(
    "Login",
    style: TextStyle(
      fontFamily: "Urbanist",
      fontSize: 25,
      color: Colors.white,
      fontWeight: FontWeight.w600,
    ),
  ),
),
),
),
Row(

```

```

children: [
  Expanded(
    child: Divider(
      color: const Color(0xFF424F7B).withOpacity(0.5),
    ),
  ),
  Padding(
    padding: EdgeInsets.symmetric(horizontal: 8.0),
    child: Text(
      "or login with",
      style: TextStyle(
        color: Color(0xFF424F7B),
        fontFamily: "Urbanist",
        fontSize: 20,
      ),
    ),
  ),
  Expanded(
    child: Divider(
      color: const Color(0xFF424F7B).withOpacity(0.5),
    ),
  ),
],
),
const SizedBox(height: 20),
Row(
  mainAxisAlignment: MainAxisAlignment.center,
  children: [
    SizedBox(
      width: 50,
      height: 50,
      child: IconButton(
        icon: Image.asset('assets/images/google_icon.png'),
        onPressed: () {},
      ),
    ),
    const SizedBox(width: 20),
    SizedBox(
      width: 50,
      height: 50,
      child: IconButton(
        icon: Image.asset('assets/images/vk_icon.png'),
        onPressed: () {},
      ),
    ),
  ],
),
isMobile
? const Spacer()
: const SizedBox(height: 100),
Center(
  child: TextButton(
    onPressed: () {},
  ),
)

```

```

        child: RichText(
          text: TextSpan(
            text: "Don't have an account? ",
            style: TextStyle(
              fontFamily: "Urbanist",
              color: Color(0xFF424F7B),
              fontSize: 18,
            ),
          children: <TextSpan>[
            TextSpan(
              text: "Register",
              style: TextStyle(
                fontFamily: "Urbanist",
                color: Color(0xFF103F74),
                fontWeight: FontWeight.bold,
              ),
            ),
          ],
        ),
      ),
    ),
  ],
);

return Scaffold(
  body: Container(
    color: Colors.white,
    child: Padding(
      padding: const EdgeInsets.all(20.0),
      child: isMobile
        ? content
        : SingleChildScrollView(
            child: content,
          ),
    ),
  ),
);
}

@override
void dispose() {
  super.dispose();
}
}

```

На рисунках 15 и 16 представлены разработанный экран авторизации на мобильном устройстве и в chrome.

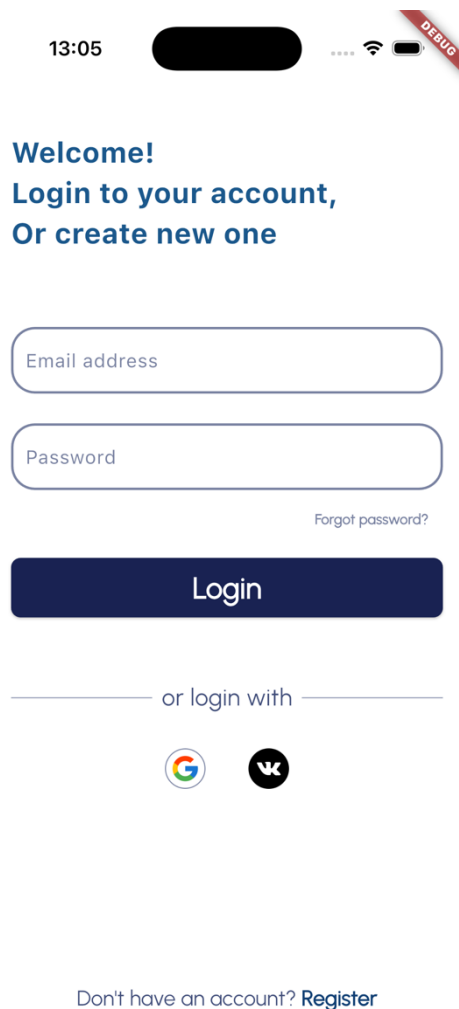


Рисунок 15 – Разработанный экран в симуляторе iOS

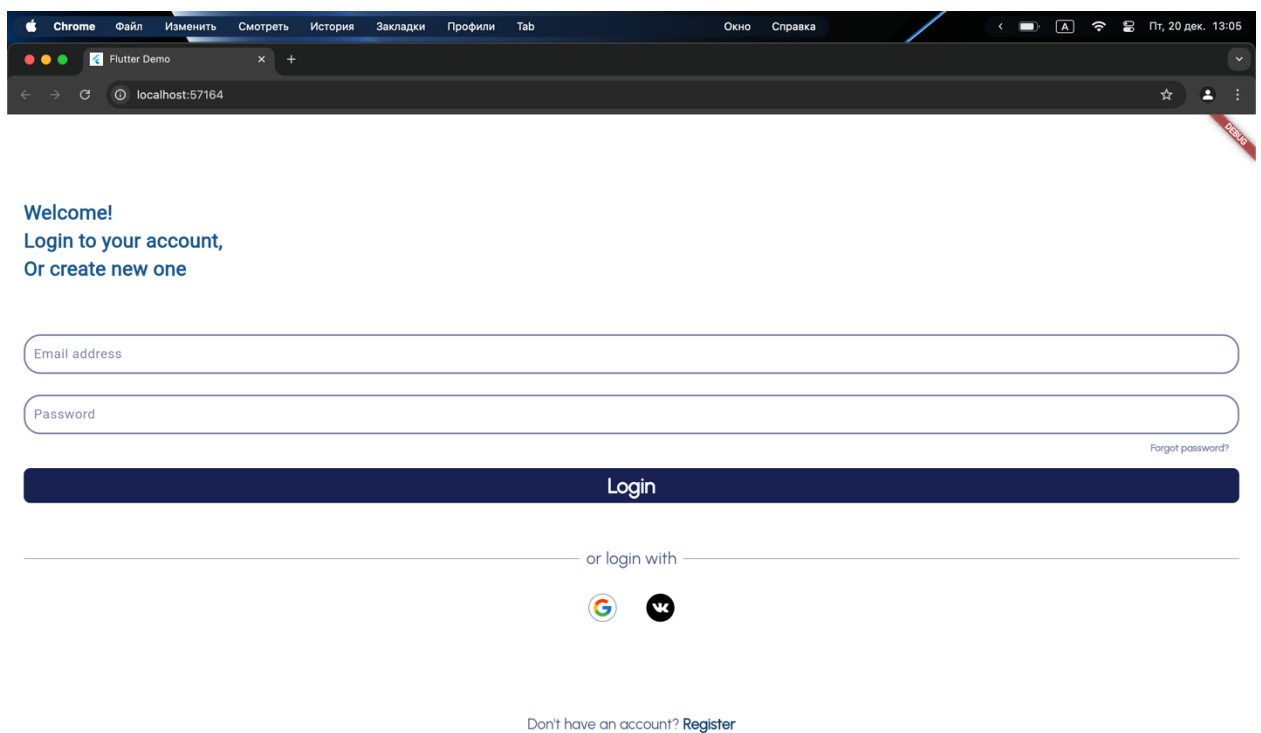


Рисунок 16 – Разработанный экран в chrome

2.3.3. Верстка экрана регистрации

В папке pages создаем файл register_page.dart. Импортируем в него необходимые библиотеки и файлы, создаем класс и настроим каркас экрана аналогично пункту 2.3.2, листинг 13 – листинг 26.

Листинг 26 – базовая настройка файла

```
import 'package:flutter/material.dart';
import '../Components/text_field.dart';
import 'package:flutter/foundation.dart';
import 'dart:io';

class RegisterPage extends StatefulWidget {
  const RegisterPage({super.key});

  @override
  _RegisterPageState createState() => _RegisterPageState();
}

class _RegisterPageState extends State<RegisterPage> {

  @override
  void initState() {
    super.initState();
  }

  @override
  Widget build(BuildContext context) {
    bool isMobile = !kIsWeb && (Platform.isAndroid || Platform.isIOS);

    Widget content = Column(
      crossAxisAlignment: CrossAxisAlignment.start,
      children: [
        ],
    );
    return Scaffold(
      body: Container(
        color: Colors.white,
        child: Padding(
          padding: const EdgeInsets.all(20.0),
          child: isMobile
            ? content
            : SingleChildScrollView(
                child: content,
              ),
        ),
      ),
    );
  }
}
```

Создадим AppBar, для этого в начале Scaffold добавим параметр appBar, зададим ему цвет заднего фона в параметре backgroundColor, зададим заголовок в параметре title при помощи виджета Text(), уберем тень под appBar при помощи установки elevation в 0. Добавим кнопку при помощи IconButton(), и обернем его в SizedBox(размеры) задав нужные размеры – Листинг 27 и 28.

Листинг 27 – Блок appBar

```
appBar: AppBar(  
  backgroundColor: Colors.white,  
  title: Text(  
    "Register",  
    style: TextStyle(  
      fontSize: 30,  
      fontWeight: FontWeight.bold,  
      color: Color(0xFF192252)  
    ),  
  ),  
  elevation: 0,  
  leading: SizedBox(  
    width: 45,  
    height: 45,  
    child: IconButton(  
      icon: Image.asset('assets/images/backIcon.png'),  
      onPressed: () {},  
    ),  
  ),  
),
```

Листинг 28 – Блок appBar в Scaffold

```
return Scaffold(  
  appBar: AppBar(  
    backgroundColor: Colors.white,  
    title: Text(  
      "Register",  
      style: TextStyle(  
        fontSize: 30,  
        fontWeight: FontWeight.bold,  
        color: Color(0xFF192252)  
      ),  
    ),  
    elevation: 0,  
    leading: SizedBox(  
      width: 45,  
      height: 45,  
      child: IconButton(  
        icon: Image.asset('assets/images/backIcon.png'),  
        onPressed: () {},  
      ),  
    ),  
  ),  
),
```



```

body: Container(
  color: Colors.white,
  child: Padding(
    padding: const EdgeInsets.all(20.0),
    child: isMobile
      ? content
      : SingleChildScrollView(
          child: content,
        ),
  ),
),
);
}

```

Перейдем к переменной content. Выравнивание по горизонтальной оси устанавливаем в `CrossAxisAlignment.start`. Установим отступ сверху в 40px при помощи `SizedBox(height: 40)`, после которого добавим 4 текстовых поля:

- Full name – полное имя;
- Email – адрес электронной почты;
- Password – пароль;
- Confirm Password – подтверждение пароля;

между которыми установим отступы в 24px при помощи `SizedBox(height: 24)`. После последнего поля добавим отступ в 43px при помощи `SizedBox(height: 24)` – листинг 29.

Листинг 29 – Добавление в column текстовых полей

```

const SizedBox(height: 40),
CustomTextField(
  labelText: "Full name"
),
const SizedBox(height: 24),
CustomTextField(
  labelText: "Email address"
),
const SizedBox(height: 24),
CustomTextField(
  labelText: "Password",
  obscure: true,
),
const SizedBox(height: 24),
CustomTextField(
  labelText: "Confirm password",
  obscure: true,
),
const SizedBox(height: 43),

```

Добавим кнопку «Register». Для этого воспользуемся виджетов ElevatedButton и повторим тот же цикл действий, что и в пункте 2.3.2, листинг 21 – листинг 30.

Листинг 30 – Создание кнопки «Register»

```
ElevatedButton(  
  onPressed: () {},  
  style: ElevatedButton.styleFrom(  
    backgroundColor: const Color(0xFF192252),  
    shape: RoundedRectangleBorder(  
      borderRadius: BorderRadius.circular(8),  
    ),  
    minimumSize: const Size(double.infinity, 50),  
  ),  
  child: Text(  
    "Register",  
    style: TextStyle(  
      fontFamily: "Urbanist",  
      fontSize: 25,  
      color: Colors.white,  
      fontWeight: FontWeight.w600,  
    ),  
  ),  
)
```

После кнопки, в зависимости от того – какое у нас устройство – добавим либо отступ, либо Spacer(). После которого, повторим цикл действий из пункта 2.3.2, листинг 24 с целью создания кнопки в виде строки «Already have an account? Login» - листинг 31.

Листинг 31 – Создание кнопки в виде строки «Already have an account? Login»

```
isMobile  
? const Spacer()  
: const SizedBox(height: 200),  
Center(  
  child: TextButton(  
    onPressed: () {},  
    child: RichText(  
      text: TextSpan(  
        text: "Already have an account? ",  
        style: TextStyle(  
          fontFamily: "Urbanist",  
          color: Color(0xFF424F7B),  
          fontSize: 18,  
        ),  
      ),  
      children: <TextSpan>[  
        TextSpan(  
          text: "Login",  
          style: TextStyle(  

```

```
fontFamily: "Urbanist",  
color: Color(0xFF103F74),  
fontWeight: FontWeight.bold,  
),  
),  
],  
),  
),  
),  
),
```

В листинге 32 представлен полный код экрана регистрации.

Листинг 32 – экран регистрации

```
import 'package:flutter/material.dart';
import '../Components/text_field.dart';
import 'package:flutter/foundation.dart';
import 'dart:io';

class RegisterPage extends StatefulWidget {
  const RegisterPage({super.key});

  @override
  _RegisterPageState createState() => _RegisterPageState();
}

class _RegisterPageState extends State<RegisterPage> {

  @override
  void initState() {
    super.initState();
  }

  @override
  Widget build(BuildContext context) {
    bool isMobile = !kIsWeb && (Platform.isAndroid || Platform.isIOS);

    Widget content = Column(
      crossAxisAlignment: CrossAxisAlignment.start,
      children: [
        const SizedBox(height: 40),
        CustomTextField(
          labelText: "Full name"
        ),
        const SizedBox(height: 24),
        CustomTextField(
          labelText: "Email address"
        ),
        const SizedBox(height: 24),
        CustomTextField(
          labelText: "Password",
          obscure: true,
        ),
        const SizedBox(height: 24),
      ],
    );
  }
}
```

```
CustomTextField(
  labelText: "Confirm password",
  isObscure: true,
),
const SizedBox(height: 43),
ElevatedButton(
  onPressed: () {},
  style: ElevatedButton.styleFrom(
    backgroundColor: const Color(0xFF192252),
    shape: RoundedRectangleBorder(
      borderRadius: BorderRadius.circular(8),
    ),
    minimumSize: const Size(double.infinity, 50),
  ),
  child: Text(
    "Register",
    style: TextStyle(
      fontFamily: "Urbanist",
      fontSize: 25,
      color: Colors.white,
      fontWeight: FontWeight.w600,
    ),
  ),
),
),
isMobile
? const Spacer()
: const SizedBox(height: 200),
Center(
  child: TextButton(
    onPressed: () {},
    child: RichText(
      text: TextSpan(
        text: "Already have an account? ",
        style: TextStyle(
          fontFamily: "Urbanist",
          color: Color(0xFF424F7B),
          fontSize: 18,
        ),
      ),
      children: <TextSpan>[
        TextSpan(
          text: "Login",
          style: TextStyle(
            fontFamily: "Urbanist",
            color: Color(0xFF103F74),
            fontWeight: FontWeight.bold,
          ),
        ),
      ],
    ),
  ),
],
),
),
),
),
```

```


);


return Scaffold(
  appBar: AppBar(
    backgroundColor: Colors.white,
    title: Text(
      "Register",
      style: TextStyle(
        fontSize: 30,
        fontWeight: FontWeight.bold,
        color: Color(0xFF192252)
      ),
    ),
  ),
  elevation: 0,
  leading: SizedBox(
    width: 45,
    height: 45,
    child: IconButton(
      icon: Image.asset('assets/images/backIcon.png'),
      onPressed: () {},
    ),
  ),
  body: Container(
    color: Colors.white,
    child: Padding(
      padding: const EdgeInsets.all(20.0),
      child: isMobile
        ? content
        : SingleChildScrollView(
            child: content,
          ),
    ),
  ),
);
}

@override
void dispose() {
  super.dispose();
}
}

```

На рисунках 17 и 18 представлены разработанный экран регистрации на мобильном устройстве и в chrome.

13:40 

 **Register**

Full name

Email address


Password

Confirm password

Register


Already have an account? [Login](#)

Рисунок 17 – Разработанный экран в симуляторе iOS

Chrome Файл Изменить Смотреть История Закладки Профили Tab Окно Справка 

Flutter Demo x +

localhost:59059/#/register

 **Register**

Full name

Email address

Password

Confirm password

Register

Already have an account? [Login](#)

Рисунок 18 – Разработанный экран в chrome

2.3.4. Верстка главного экрана

В папке pages создаем файл home_page.dart. Импортируем в него необходимые библиотеки и файлы, создаем класс и настроим каркас экрана – листинг 33.

Листинг 33 – Базовая настройка главного экрана

```
import 'package:flutter/material.dart';
import 'package:smooth_page_indicator/smooth_page_indicator.dart';
import 'package:flutter/foundation.dart';
import 'dart:io';

class HomePage extends StatefulWidget {
  const HomePage({super.key});

  @override
  _HomePageState createState() => _HomePageState();
}

class _HomePageState extends State<HomePage> {

  @override
  void initState() {
    super.initState();
  }

  @override
  Widget build(BuildContext context) {

    bool isMobile = !kIsWeb && (Platform.isAndroid || Platform.isIOS);

    return Scaffold(
      body:
    ),
  );
}
}
```

Создадим appBar, который будет содержать блок с локацией пользователя и иконками сообщения и поиска. Для этого, в поле title помещаем Column, который выравниваем по началу горизонтальной оси. В качестве дочерних виджетов добавляем Row, который будет содержать текст «Your location» и индикатор выпадающего списка, под рядом помещаем текст с локацией пользователя: «Moscow, Russia». Кнопки добавляем в параметр actions, создаем их при помощи IconButton, аналогично предыдущим пунктам. В листинге 34 представлен блок appBar.

Листинг 34 – блок appBar

```
appBar: AppBar(  
  backgroundColor: Colors.white,  
  elevation: 0,  
  title: Column(  
    crossAxisAlignment: CrossAxisAlignment.start,  
    children: [  
      Row(  
        children: [  
          Text(  
            "Your location",  
            style: TextStyle(  
              color: Color(0xFF848FAC),  
              fontSize: 18,  
              fontFamily: "Urbanist",  
            ),  
          ),  
          SizedBox(width: 10),  
          ImageIcon(AssetImage('assets/images/arrow-down.png')),  
        ],  
      ),  
      Text(  
        "Moscow, Russia",  
        style: TextStyle(  
          color: Color(0xFF192252),  
          fontSize: 20,  
          fontWeight: FontWeight.bold,  
        ),  
      ),  
    ],  
  ),  
  actions: [  
    IconButton(  
      icon: const ImageIcon(  
        AssetImage('assets/images/search.png'),  
        color: Color(0xFF192252),  
      ),  
      onPressed: () {},  
    ),  
    IconButton(  
      icon: const ImageIcon(  
        AssetImage('assets/images/sms.png'),  
        color: Color(0xFF192252),  
      ),  
      onPressed: () {},  
    ),  
  ],  
),
```

Основное содержимое экрана поместим в Container, установим белый цвет фона. В качестве дочернего виджета будет SingleChildScrollView с установленным отступом сверху в 10px при помощи Padding,

EdgeInsets.only(top: 10). Дочерний виджет – Column, выравнивание по началу горизонтальной оси.

Начнем создание блока с акциями / новостями. Так как нам нужно создать адаптивную верстку, то воспользуемся виджетом LayoutBuilder, который представляет текущие ограничения для дочерних виджетов через BoxConstraints. Создадим переменную screenWidth в которую занесем ширину экрана при помощи constraints.maxWidth, blockHeight – переменная в которой будет вычисляться высота блока с ограничениями от 200 до 400 px, imageHeight – переменная, в которой будет вычисляться размер изображения путем вычитания 50px из blockHeight, но с ограничениями от 150 до 350px. Возвращать будет SizedBox контейнер, высотой blockHeight, и дочерним виджетом PageView, так как акции должны перелистываться как страницы. В классе _HomePageState создадим переменную _pageController = PageController() – листинг 35.

Листинг 35 – Создание переменной pageController

```
class _HomePageState extends State<HomePage> {  
  final PageController _pageController = PageController();  
  // Остальной код  
}
```

В PageView.builder() в поле controller передадим созданную переменную pageController, зададим количество акций / новостей в поле itemCount, в поле onChanged укажем метод, который будет вызываться при смене страницы, в нашем случае – setState(), чтобы обновить виджет при смене индекса, itemBuilder – функция, которая вернет содержимое каждой страницы, передавая индекс текущей страницы. В качестве возвращаемого у нас будет вертикальный стек Column, содержащий Container высотой imageHeight с закругленными углами при помощи BoxDecoration, изображение - DecorationImage, в который мы передадим AssetImage(путь до нашего изображения), название акции – Text() с ограничением длины строки равное одной строке. Весь блок LayoutBuilder обернем в виджет Padding и добавим горизонтальные отступы по 8px - листинг 36.

Листинг 36 – блок с LayoutBuilder для промоакциями

```

Padding(
  padding: EdgeInsets.symmetric(horizontal: 8.0),
  child: LayoutBuilder(
    builder: (context, constraints) {
      double screenWidth = constraints.maxWidth;
      double blockHeight = (screenWidth / 3).clamp(200, 400);
      double imageHeight = (blockHeight - 50).clamp(150, 350);

      return SizedBox(
        height: blockHeight,
        child: PageView.builder(
          controller: _pageController,
          itemCount: 4,
          onPageChanged: (int index) {
            setState(() {});
          },
          itemBuilder: (context, index) {
            return Column(
              crossAxisAlignment: CrossAxisAlignment.start,
              children: [
                Container(
                  height: imageHeight,
                  decoration: BoxDecoration(
                    borderRadius: BorderRadius.circular(16),
                    image: DecorationImage(
                      image:
                        AssetImage('assets/images/bmw5.jpg'),
                      fit: BoxFit.cover),
                ),
                const SizedBox(height: 8),
                Text(
                  "New car in rent - M5 F90",
                  style: const TextStyle(
                    fontSize: 20,
                    fontWeight: FontWeight.bold,
                    fontFamily: "Urbanist",
                  ),
                  maxLines: 1,
                  overflow: TextOverflow.ellipsis,
                ),
              ],
            );
          },
        ),
      );
    },
  ),
);

```

Добавим индикатор страниц. Для этого нам понадобится `smooth_page_indicator`, чтобы его добавить, в `pubspec.yaml`, где добавляются зависимости добавим строку:

```
smooth_page_indicator: ^1.2.0+3
```

В `home_page.dart` добавим импорт:

```
import 'package:smooth_page_indicator/smooth_page_indicator.dart';
```

Теперь под отступом добавим `SmoothPageIndicator`, в который в качестве `controller` передадим `_pageController`, установим ограничение в 4 объявления, в качестве эффекта используем `ExpandingDotsEffect`. Обернем весь индикатор в `Padding`, чтобы добавить отступы по горизонтали в 8px и вертикали – сверху в 10px, а снизу в 27 – Листинг 37.

Листинг 37 – Добавление `SmoothPageIndicator`

```
Padding(  
  padding: EdgeInsets.fromLTRB(8, 10, 8, 27),  
  child: SmoothPageIndicator(  
    controller: _pageController,  
    count: 4,  
    effect: const ExpandingDotsEffect(  
      expansionFactor: 4.0,  
      activeDotColor: Color(0xFF192252),  
      dotColor: Color(0xFFD9D9D9),  
      dotHeight: 8,  
      dotWidth: 8,  
      spacing: 4.0,  
    ),  
  ),  
),
```

Перейдем к созданию блока с популярными машинами. Он у нас содержит горизонтальный стек с названием блока и прокручиваемым блоком с популярными автомобилями. Для начала создадим горизонтальный стек, для этого воспользуемся `Row`, в который поместим `Text` (“Popular cars”) и `TextButton` с текстом “see all”. Чтобы развести текст «Popular cars» и кнопку «see all» в `mainAxisAlignment` у `Row` установим `MainAxisAlignment.spaceBetween`. Обернем данный блок в `Padding` с горизонтальным отступом в 8px – листинг 38.

Листинг 38 – Горизонтальный стек «Popular cars»

```
Padding(  
  padding: EdgeInsets.symmetric(horizontal: 8.0),  
  child: Row(  
    mainAxisAlignment: MainAxisAlignment.spaceBetween,  
    children: [  
      Text(  
        "Popular cars",  
        style: TextStyle(  
          color: Color(0xFF192252),  
          fontStyle: FontStyle.normal,  
          fontWeight: FontWeight.normal,  
          size: 16.0,  
        ),  
      ),  
      TextButton(  
        onPressed: () {  
          Navigator.push(  
            context, MaterialPageRoute(builder: (context) => PopularCarsPage()),  
          );  
        },  
        child: Text(  
          "see all",  
          style: TextStyle(  
            color: Color(0xFF192252),  
            fontStyle: FontStyle.normal,  
            fontWeight: FontWeight.normal,  
            size: 16.0,  
          ),  
        ),  
      ),  
    ],  
  ),  
),
```

```
        fontSize: 20,  
        fontWeight: FontWeight.bold,  
      ),  
    ),  
    TextButton(  
      onPressed: () {},  
      child: Text(  
        "see all",  
        style: TextStyle(  
          fontSize: 20,  
          color: Color(0xFF192252),  
          fontFamily: "Urbanist",  
        ),  
      ),  
    ),  
  )  
],  
),  
),
```

Теперь создадим горизонтально прокручиваемый блок с карточками автомобилей. Для этого создадим виджет `carCard`. Установим входные параметры:

- `context` — объект `BuildContext`, предоставляющий контекст для построения виджетов. Обычно используется для доступа к теме приложения или навигации.
- `imageUrl` — строка с путем к изображению автомобиля.
- `title` — название автомобиля.
- `price` — цена аренды автомобиля за день.
- `rating` — рейтинг автомобиля (например, "4.5").
- `numOfReviews` — количество отзывов (например, "23").
- `isHorizontalScroll` (по умолчанию `false`) — указывает, используется ли горизонтальный скролл (влияет на отступы карточки).

Создадим переменную `numOfReviewsInt`, в которую будет передаваться строковое значение отзывов, которое будет убирать все нечисловые символы при помощи метода `parse`. Далее – создадим переменную `formattedReviews`, где в зависимости от количества отзывов будем выдавать правильное написание. Сама карточка представляет из себя закругленный белый контейнер с тенью, шириной в `180px` и `219px`, настройкой отступов в зависимости от значения переменной `isHorizontalScroll`. Дочерним виджетом является горизонтальный

стек – Column, выравненный по началу горизонтальной оси и, в качестве дочерних виджетов, содержащий закругленный контейнер высотой 85px с изображением, названием автомобиля, рядом, содержащим иконку звезды и количество отзывов, Spacer(), чтобы переместить цену вниз карточки -

Листинг 39 – Карточка автомобиля.

```
static Widget carCard({
  required BuildContext context,
  required String imageUrl,
  required String title,
  required int price,
  required String rating,
  required String numOfReviews,
  bool isHorizontalScroll = false,
}) {
  int numOfReviewsInt = int.parse(numOfReviews.replaceAll(RegExp(r'\D'), ''));
  String formattedReviews;
  if (numOfReviewsInt >= 100) {
    formattedReviews = "(100+ reviews)";
  } else if (numOfReviewsInt == 1) {
    formattedReviews = "($numOfReviews) review";
  } else {
    formattedReviews = "($numOfReviews) reviews";
  }

  return Container(
    width: 180,
    height: 219,
    margin: isHorizontalScroll
      ? const EdgeInsets.only(right: 16, bottom: 10, left: 8)
      : const EdgeInsets.only(bottom: 8),
    decoration: BoxDecoration(
      color: Colors.white,
      borderRadius: BorderRadius.circular(16),
      boxShadow: [
        BoxShadow(
          color: Colors.black.withOpacity(0.1),
          spreadRadius: 1,
          blurRadius: 5,
          offset: const Offset(0, 3),
        ),
      ],
    ),
    child: Column(
      crossAxisAlignment: CrossAxisAlignment.start,
      children: [
        Container(
          height: 85,
          decoration: BoxDecoration(
            borderRadius: BorderRadius.circular(16),
```

```

        image: DecorationImage(
          image: AssetImage(imageUrl),
          fit: BoxFit.cover,
        ),
      ),
    ),
    Padding(
      padding: const EdgeInsets.all(8.0),
      child: Text(
        title,
        style: const TextStyle(
          fontSize: 15,
          fontWeight: FontWeight.bold,
          color: Color(0xFF192252),
        ),
        overflow: TextOverflow.ellipsis,
        maxLines: 1,
      ),
    ),
    Padding(
      padding: const EdgeInsets.symmetric(horizontal: 8),
      child: Row(
        children: [
          const ImageIcon(
            AssetImage('assets/images/star.png'),
            color: Color(0xFFFFBD16),
          ),
          const SizedBox(width: 3),
          Text(
            rating,
            style: const TextStyle(
              fontSize: 15,
              color: Color(0xFFFFBD16),
            ),
          ),
          const SizedBox(width: 10),
          Text(
            formattedReviews,
            style: const TextStyle(
              fontSize: 12,
              color: Color(0xFF192252),
            ),
          ),
        ],
      ),
    ),
    const Spacer(),
    Padding(
      padding: const EdgeInsets.only(left: 8, bottom: 8, right: 8),
      child: Text(
        '$price ₪ / day',
        style: const TextStyle(
          fontSize: 15,

```

```
color: Color(0xFF192252),
    ),
    ),
    ),
],
),
);
}
```

Теперь, после ряда с текстом «Popular cars» создадим `SizedBox` высотой 220px, как дочерним виджетом будет `ListView.builder`, который позволяет создать горизонтально прокручиваемый блок. Установим ось прокрутки – горизонтальной, кол-во иконок – 5, `shrinkWrap` – true, так как надо, чтобы список занимал столько места, сколько требуется, устанавливаем `physics` как `ClampingScrollPhysics`, чтобы прокрутка ограничивалась краями списка, в `itemBuilder` создаем каждый элемент по индексу. Возвращать будет созданную карточку автомобиля. Обернем список в `Padding`, чтобы добавить отступы по вертикали в 22px – листинг 40.

Листинг 40 – Прокручиваемый блок «Popular cars»

```
Padding(
  padding: const EdgeInsets.symmetric(vertical: 22),
  child: SizedBox(
    height: 220,
    child: ListView.builder(
      scrollDirection: Axis.horizontal,
      itemCount: 5,
      shrinkWrap: true,
      physics: const ClampingScrollPhysics(),
      itemBuilder: (context, index) {
        return carCard(
          context: context,
          imageUrl: 'assets/images/bmw5.jpg',
          title: "BMW M5",
          price: 22000,
          rating: "5",
          numOfReviews: "100",
          isHorizontalScroll: true,
        );
      },
    ),
  ),
),
```

Теперь создадим блок со всеми машинами, он состоит из теста «All cars» и сетки с карточками автомобилей. Для начала создадим `Text("All cars")`, обернем его в `Padding`, чтобы добавить отступ сверху в 8px и снизу в 22px. При

помощи `LayoutBuilder` будем создавать сетку, для этого, внутри него определяем переменную `screenWidth` и задаем максимальную ширину, создаем переменную `crossAxisCount`, которая будет содержать количество элементов в одном ряду сетки: от двух до бесконечности. Вычисляется при помощи целочисленного деления ширины на ширину карточки, равного 180px. Возвращаться будет `GridView.builder` с отступом снизу в 50px, блокировкой дополнительной прокрутки в `physics` при помощи `NeverScrollableScrollPhysics()`, `shrinkWrap` – `true`, установкой делегата `SliverGridDelegateWithFixedCrossAxisCount`, в параметры которого передадим значение количества элементов в одном ряду сетки, отступом по главной оси в 22px, по второстепенной оси в 8px, `childAspectRatio` равному ширину делим на высоту ячейки (180 / 220). Установим у `GridView.builder` количество элементов в сетки равно 8, а `itemBuilder` будет возвращать карточку автомобиля. Обернем `LayoutBuilder` в `Padding`, чтобы добавить горизонтальный отступ в 8px – листинг 41.

Листинг 41 – Блок со всеми автомобилям

```
Padding(  
  padding: EdgeInsets.fromLTRB(8, 0, 0, 22),  
  child: Text(  
    "All cars",  
    style: TextStyle(  
      fontSize: 20,  
      fontWeight: FontWeight.bold,  
      color: Color(0xFF192252),  
    ),  
  ),  
),  
),  
Padding(  
  padding: EdgeInsets.symmetric(horizontal: 8),  
  child: LayoutBuilder(  
    builder: (context, constraints) {  
      double screenWidth = constraints.maxWidth;  
      int crossAxisCount =  
        (screenWidth ~/ 180).clamp(2, double.infinity).toInt();  
  
      return GridView.builder(  
        padding: EdgeInsets.fromLTRB(0, 0, 0, 50),  
        physics: const NeverScrollableScrollPhysics(),  
        shrinkWrap: true,  
        gridDelegate: SliverGridDelegateWithFixedCrossAxisCount(  
          crossAxisCount: crossAxisCount,
```



```

        mainAxisAlignment: 22,
        crossAxisAlignment: 8,
        childAspectRatio: 180 / 220,
      ),
      itemCount: 8,
      itemBuilder: (context, index) {
        return carCard(
          context: context,
          imageUrl: 'assets/images/bmw5.jpg',
          title: "BMW M5",
          price: 22000,
          rating: "5",
          numOfReviews: "100",
          isHorizontalScroll: false);
      },
    );
  },
),
),

```

Листинг 42 – Полный код главного экрана

```

import 'package:flutter/material.dart';
import 'package:smooth_page_indicator/smooth_page_indicator.dart';
import 'package:flutter/foundation.dart';
import 'dart:io';

class HomePage extends StatefulWidget {
  const HomePage({super.key});

  @override
  _HomePageState createState() => _HomePageState();
}

class _HomePageState extends State<HomePage> {
  final PageController _pageController = PageController();

  @override
  void initState() {
    super.initState();
  }

  @override
  void dispose() {
    _pageController.dispose();
    super.dispose();
  }

  @override
  Widget build(BuildContext context) {
    bool isMobile = !kIsWeb && (Platform.isAndroid || Platform.isIOS);

    return Scaffold(
      appBar: AppBar(
        backgroundColor: Colors.white,

```

```

elevation: 0,
title: Column(
  crossAxisAlignment: CrossAxisAlignment.start,
  children: [
    Row(
      children: [
        Text(
          "Your location",
          style: TextStyle(
            color: Color(0xFF848FAC),
            fontSize: 18,
            fontFamily: "Urbanist",
          ),
        ),
        SizedBox(width: 10),
        ImageIcon(AssetImage('assets/images/arrow-down.png')),
      ],
    ),
    Text(
      "Moscow, Russia",
      style: TextStyle(
        color: Color(0xFF192252),
        fontSize: 20,
        fontWeight: FontWeight.bold,
      ),
    ),
  ],
),
actions: [
  IconButton(
    icon: const ImageIcon(
      AssetImage('assets/images/search.png'),
      color: Color(0xFF192252),
    ),
    onPressed: () {},
  ),
  IconButton(
    icon: const ImageIcon(
      AssetImage('assets/images/sms.png'),
      color: Color(0xFF192252),
    ),
    onPressed: () {},
  ),
],
),
body: Container(
  color: Colors.white,
  child: SingleChildScrollView(
    padding: EdgeInsets.only(top: 10),
    child: Column(
      crossAxisAlignment: CrossAxisAlignment.start,
      children: [
        Padding(

```

```
padding: EdgeInsets.symmetric(horizontal: 8.0),
child: LayoutBuilder(
  builder: (context, constraints) {
    double screenWidth = constraints.maxWidth;
    double blockHeight = (screenWidth / 3).clamp(200, 400);
    double imageHeight = (blockHeight - 50).clamp(150, 350);

    return SizedBox(
      height: blockHeight,
      child: PageView.builder(
        controller: _pageController,
        itemCount: 4,
        onPageChanged: (int index) {
          setState(() {});
        },
        itemBuilder: (context, index) {
          return Column(
            crossAxisAlignment: CrossAxisAlignment.start,
            children: [
              Container(
                height: imageHeight,
                decoration: BoxDecoration(
                  borderRadius: BorderRadius.circular(16),
                  image: DecorationImage(
                    image:
                      AssetImage('assets/images/bmwm5.jpg'),
                    fit: BoxFit.cover),
              ),
              const SizedBox(height: 8),
              Text(
                "New car in rent - M5 F90",
                style: const TextStyle(
                  fontSize: 20,
                  fontWeight: FontWeight.bold,
                  fontFamily: "Urbanist",
                ),
                maxLines: 1,
                overflow: TextOverflow.ellipsis,
              ),
            ],
          );
        },
      ),
    );
  },
),
Padding(
  padding: EdgeInsets.fromLTRB(8, 10, 8, 27),
  child: SmoothPageIndicator(
    controller: _pageController,
    count: 4,
```

```

effect: const ExpandingDotsEffect(
  expansionFactor: 4.0,
  activeDotColor: Color(0xFF192252),
  dotColor: Color(0xFFD9D9D9),
  dotHeight: 8,
  dotWidth: 8,
  spacing: 4.0,
),
),
),
Padding(
  padding: EdgeInsets.symmetric(horizontal: 8.0),
  child: Row(
    mainAxisAlignment: MainAxisAlignment.spaceBetween,
    children: [
      Text(
        "Popular cars",
        style: TextStyle(
          fontSize: 20,
          fontWeight: FontWeight.bold,
        ),
      ),
      TextButton(
        onPressed: () {},
        child: Text(
          "see all",
          style: TextStyle(
            fontSize: 20,
            color: Color(0xFF192252),
            fontFamily: "Urbanist",
          ),
        ),
      ),
    ],
  ),
),
Padding(
  padding: const EdgeInsets.symmetric(vertical: 22),
  child: SizedBox(
    height: 220,
    child: ListView.builder(
      scrollDirection: Axis.horizontal,
      itemCount: 5,
      shrinkWrap: true,
      physics: const ClampingScrollPhysics(),
      itemBuilder: (context, index) {
        return carCard(
          context: context,
          imageUrl: 'assets/images/bmwm5.jpg',
          title: "BMW M5",
          price: 22000,
          rating: "5",
          numOfReviews: "100",

```

```

        isHorizontalScroll: true,
      );
    },
  ),
),
Padding(
  padding: EdgeInsets.fromLTRB(8, 0, 0, 22),
  child: Text(
    "All cars",
    style: TextStyle(
      fontSize: 20,
      fontWeight: FontWeight.bold,
      color: Color(0xFF192252),
    ),
  ),
),
Padding(
  padding: EdgeInsets.symmetric(horizontal: 8),
  child: LayoutBuilder(
    builder: (context, constraints) {
      double screenWidth = constraints.maxWidth;
      int crossAxisCount =
        (screenWidth ~/ 180).clamp(2, double.infinity).toInt();

      return GridView.builder(
        padding: EdgeInsets.fromLTRB(0, 0, 0, 50),
        physics: const NeverScrollableScrollPhysics(),
        shrinkWrap: true,
        gridDelegate: SliverGridDelegateWithFixedCrossAxisCount(
          crossAxisCount: crossAxisCount,
          mainAxisSpacing: 22,
          crossAxisSpacing: 8,
          childAspectRatio: 180 / 220,
        ),
        itemCount: 8,
        itemBuilder: (context, index) {
          return carCard(
            context: context,
            imageUrl: 'assets/images/bmw5.jpg',
            title: "BMW M5",
            price: 22000,
            rating: "5",
            numReviews: "100",
            isHorizontalScroll: false);
        },
      );
    },
  ),
),
],
),
),

```

```

    ),
  );
}

static Widget carCard({
  required BuildContext context,
  required String imageUrl,
  required String title,
  required int price,
  required String rating,
  required String numOfReviews,
  bool isHorizontalScroll = false,
}) {
  int numOfReviewsInt = int.parse(numOfReviews.replaceAll(RegExp(r'\D'), ''));
  String formattedReviews;
  if (numOfReviewsInt >= 100) {
    formattedReviews = "(100+ reviews)";
  } else if (numOfReviewsInt == 1) {
    formattedReviews = "($numOfReviews) review";
  } else {
    formattedReviews = "($numOfReviews) reviews";
  }

  return Container(
    width: 180,
    height: 219,
    margin: isHorizontalScroll
      ? const EdgeInsets.only(right: 16, bottom: 10, left: 8)
      : const EdgeInsets.only(bottom: 8),
    decoration: BoxDecoration(
      color: Colors.white,
      borderRadius: BorderRadius.circular(16),
      boxShadow: [
        BoxShadow(
          color: Colors.black.withOpacity(0.1),
          spreadRadius: 1,
          blurRadius: 5,
          offset: const Offset(0, 3),
        ),
      ],
    ),
    child: Column(
      crossAxisAlignment: CrossAxisAlignment.start,
      children: [
        Container(
          height: 85,
          decoration: BoxDecoration(
            borderRadius: BorderRadius.circular(16),
            image: DecorationImage(
              image: AssetImage(imageUrl),
              fit: BoxFit.cover,
            ),
          ),
        ),
      ],
    ),
  );
}

```

```

),
Padding(
  padding: const EdgeInsets.all(8.0),
  child: Text(
    title,
    style: const TextStyle(
      fontSize: 15,
      fontWeight: FontWeight.bold,
      color: Color(0xFF192252),
    ),
    overflow: TextOverflow.ellipsis,
    maxLines: 1,
  ),
),
Padding(
  padding: const EdgeInsets.symmetric(horizontal: 8),
  child: Row(
    children: [
      const ImageIcon(
        AssetImage('assets/images/star.png'),
        color: Color(0xFFFFBD16),
      ),
      const SizedBox(width: 3),
      Text(
        rating,
        style: const TextStyle(
          fontSize: 15,
          color: Color(0xFFFFBD16),
        ),
      ),
      const SizedBox(width: 10),
      Text(
        formattedReviews,
        style: const TextStyle(
          fontSize: 12,
          color: Color(0xFF192252),
        ),
      ),
    ],
  ),
),
const Spacer(),
Padding(
  padding: const EdgeInsets.only(left: 8, bottom: 8, right: 8),
  child: Text(
    '$price P / day',
    style: const TextStyle(
      fontSize: 15,
      color: Color(0xFF192252),
    ),
  ),
),
),
],

```

```
),  
);  
}  
}
```

На рисунках 19 и 21 представлены разработанный главный экран на мобильном устройстве и в chrome.

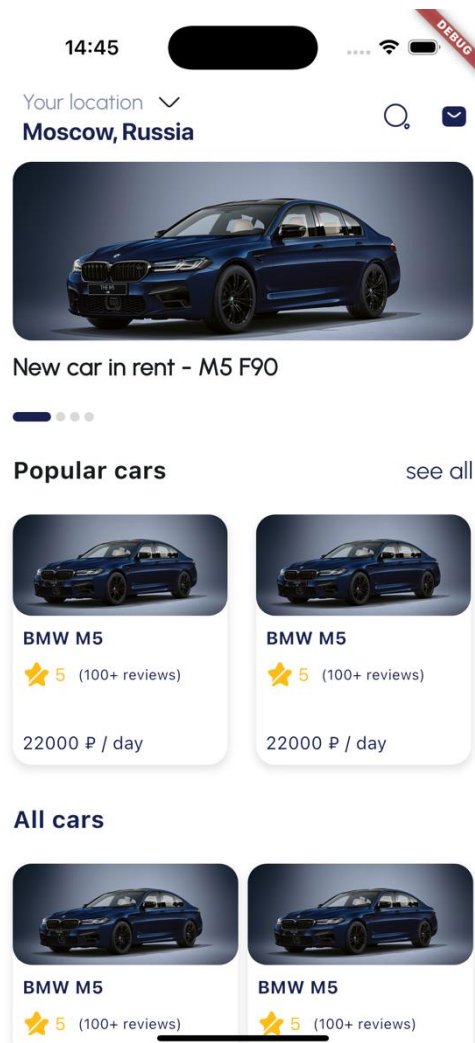


Рисунок 19 – Разработанный экран в симуляторе iOS

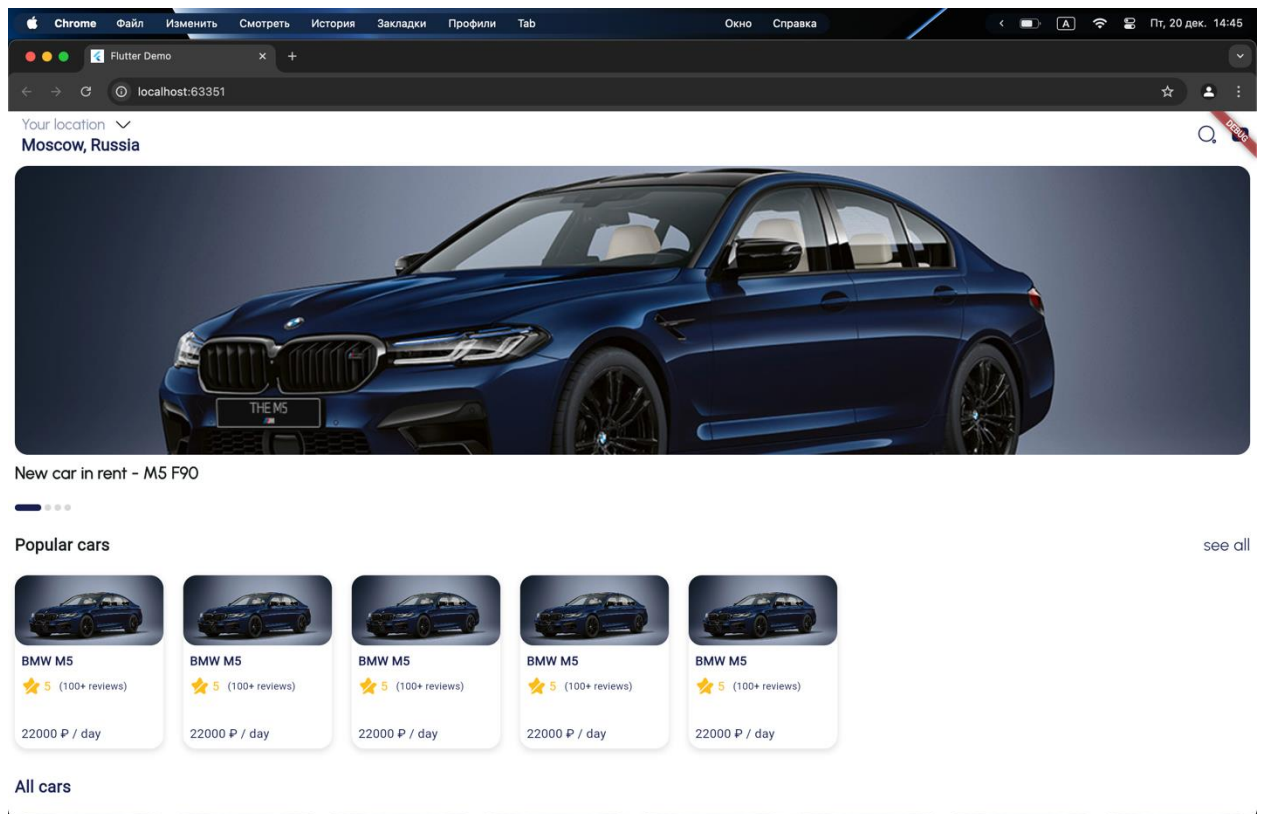


Рисунок 20 – Разработанный экран в chrome (часть 1)

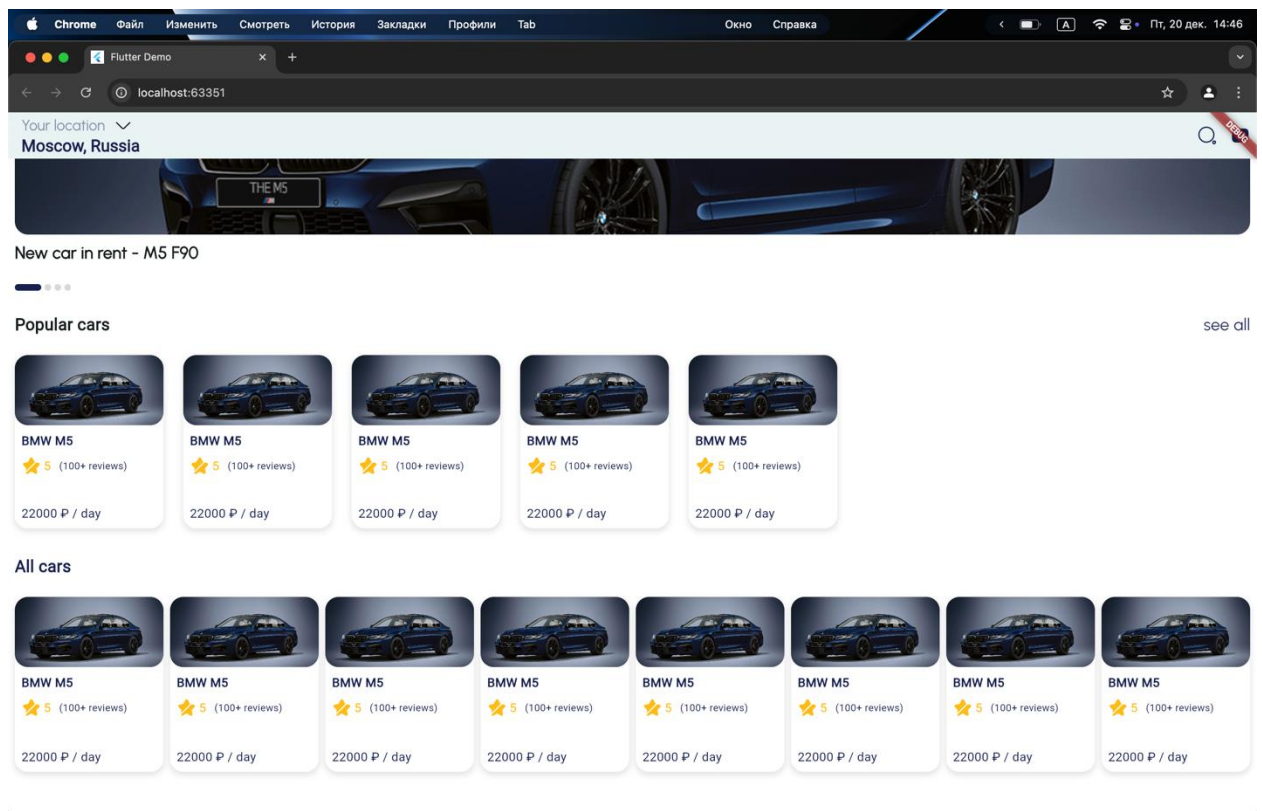


Рисунок 21 – Разработанный экран в chrome (часть 2)

2.3.5. Верстка экрана с детальной информацией об автомобиле

В папке pages создаем файл detail_page.dart. Импортируем в него необходимые библиотеки и файлы, создаем класс и настроим каркас экрана – листинг 43.

Листинг 43 – Базовая настройка detail_page.dart

```
import 'package:flutter/material.dart';
import 'package:smooth_page_indicator/smooth_page_indicator.dart';

class DetailPage extends StatefulWidget {

  const DetailPage({super.key});

  @override
  _DetailPageState createState() => _DetailPageState();
}

class _DetailPageState extends State<DetailPage> {

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body:
    );
  }
}
```

Контент экрана будет помещен в виджет Stack. В качестве дочернего виджета будет использоваться CustomScrollView(), в параметр slivers будут переданы SliverAppBar, SliverToBoxAdapter. А также внизу страницы зафиксирован будет блок с кнопкой аренды автомобиля. Для начала настроим SliverAppBar. Установим его параметры:

- automaticallyImplyLeading: false;
- leading – кнопка назад;
- expandedHeight – динамическая высота;
- flexibleSpace – контент внутри заголовка.

Так как нам надо сделать адаптивную верстку, то необходимо написать функцию, которая рассчитывала бы высоту для изображения – листинг 44.

Листинг 44 – Функция для расчета динамической высоты SliverAppBar

```
double _calculateDynamicHeight(BuildContext context) {
  double screenWidth = MediaQuery.of(context).size.width;

  if (screenWidth < 400) {
    return 200;
  } else if (screenWidth > 800) {
    return 600;
  } else {
    return 200 + (screenWidth - 400) * (400 / 400);
  }
}
```

В `flexibleSpace` помещаем `LayoutBuilder` и возвращаем `FlexibleSpaceBar`, в `title` устанавливаем анимированную прозрачность, `background` устанавливаем `Stack`, который содержит `PageView` из изображений автомобиля и выравненный по центру снизу индикатор страницы, а также полупрозрачный слой, который исчезает при прокрутке. Для `PageView` и индикатора страницы создадим аналогично пункту 2.3.5. `pageController`.

Листинг 45 – `FlexibleSpace`

```
flexibleSpace: LayoutBuilder(
  builder: (BuildContext context, BoxConstraints constraints) {
    var top = constraints.biggest.height;
    return FlexibleSpaceBar(
      title: AnimatedOpacity(
        opacity: top <= kToolbarHeight + 50 ? 1.0 : 0.0,
        duration: const Duration(milliseconds: 300),
      ),
      background: Stack(
        fit: StackFit.expand,
        children: [
          PageView(
            controller: _pageController,
            children: [
              Image.asset(
                'assets/images/bmwm5.jpg',
                fit: BoxFit.cover,
              ),
              Image.asset(
                'assets/images/bmwm5.jpg',
                fit: BoxFit.cover,
              ),
            ],
          ),
          Align(
            alignment: Alignment.bottomCenter,
            child: Padding(
              padding: const EdgeInsets.only(
                bottom: 16),
              child: SmoothPageIndicator(
```

```

        controller:
          _pageController,
        count: 2,
        effect: const ExpandingDotsEffect(
          expansionFactor:
            4.0,
          activeDotColor:
            Color(0xFF192252),
          dotColor: Color(
            0xFFD9D9D9),
          dotHeight: 8,
          dotWidth: 8,
          spacing: 8.0,
        ),
      ),
    ),
  ),
  IgnorePointer(
    child: Container(
      color: Colors.white.withOpacity(
        top <= kToolbarHeight + 50
          ? 1.0
          : 0.0,
      ),
    ),
  ),
],
),
);
},
),

```

Листинг 46 – SilverAppBar

```

SilverAppBar(
  automaticallyImplyLeading: false,
  leading: Padding(
    padding: const EdgeInsets.only(left: 24),
    child: InkWell(
      onTap: () {
        Navigator.pop(context);
      },
      borderRadius: BorderRadius.circular(25),
      child: Container(
        width: 50,
        height: 50,
        decoration: const BoxDecoration(
          shape: BoxShape.circle,
          color: Colors.white,
        ),
        child: const Center(
          child: Icon(
            Icons.chevron_left,
            size: 30,
            color: Color(0xFF192252),

```

```

    ),
    ),
    ),
    ),
),
expandedHeight: _calculateDynamicHeight(context),
floating: false,
pinned: true,
flexibleSpace: LayoutBuilder(
  builder: (BuildContext context, BoxConstraints constraints) {
    var top = constraints.biggest.height;
    return FlexibleSpaceBar(
      title: AnimatedOpacity(
        opacity: top <= kToolbarHeight + 50 ? 1.0 : 0.0,
        duration: const Duration(milliseconds: 300),
      ),
      background: Stack(
        fit: StackFit.expand,
        children: [
          PageView(
            controller: _pageController,
            children: [
              Image.asset(
                'assets/images/bmw5.jpg',
                fit: BoxFit.cover,
              ),
              Image.asset(
                'assets/images/bmw5.jpg',
                fit: BoxFit.cover,
              ),
            ],
          ),
          Align(
            alignment: Alignment.bottomCenter,
            child: Padding(
              padding: const EdgeInsets.only(
                bottom: 16),
              child: SmoothPageIndicator(
                controller:
                  _pageController,
                count: 2,
                effect: const ExpandingDotsEffect(
                  expansionFactor:
                    4.0,
                  activeDotColor:
                    Color(0xFF192252),
                  dotColor: Color(
                    0xFFD9D9D9),
                  dotHeight: 8,
                  dotWidth: 8,
                  spacing: 8.0,
                ),
              ),
            ),
          ),
        ],
      ),
    ),
  ),
),

```

```

    ),
  ),
  IgnorePointer(
    child: Container(
      color: Colors.white.withOpacity(
        top <= kToolbarHeight + 50
          ? 1.0
          : 0.0,
      ),
    ),
  ),
),
],
),
);
},
),
),
),

```

Далее поместим основной контент, для этого воспользуемся SilverToBoxAdapter. В качестве дочернего виджета будет использоваться Container, в котором будет Column, содержащая:

- Название автомобиля;
- Блок с рейтингом и отзывами;
- Блок с описанием;
- Блок с информацией об автомобиле.

Для начала создадим название автомобиля, для этого воспользуемся виджетом Text(), обернем его в Padding и установим отступ в 16px со всех сторон при помощи EdgeInsets.all(16) – листинг 47.

Листинг 47 – Блок с названием автомобиля

```

Padding(
  padding: const EdgeInsets.all(16),
  child: Text(
    "BMW M5 F90, 2022",
    style: const TextStyle(
      fontSize: 25,
      fontWeight: FontWeight.w900,
      color: Color(0xFF192252),
    ),
  ),
),
),
),

```

После создадим блок с рейтингом и отзывами. Для этого воспользуемся виджетом Row, который будет содержать изображение звезды, рейтинг в виде Text() и Text() с количеством отзывов – листинг 48.

Листинг 48 – Блок с рейтингом и отзывами об автомобиле

```
Padding(  
  padding: const EdgeInsets.symmetric(horizontal: 16),  
  child: Row(  
    children: [  
      const ImageIcon(  
        AssetImage('assets/images/star.png'),  
        size: 30,  
        color: Color(0xFFFFBD16),  
      ),  
      Padding(  
        padding: EdgeInsets.fromLTRB(3, 0, 10, 0),  
        child: Text(  
          '5.0',  
          style: const TextStyle(  
            fontSize: 20,  
            color: Color(0xFFFFBD16),  
          ),  
        ),  
      ),  
      Text(  
        "(100+ reviews)",  
        style: const TextStyle(  
          fontSize: 15,  
          color: Color(0xFF192252),  
        ),  
      ),  
    ],  
  ),  
)
```

После – создадим блок с описанием. Для этого воспользуемся виджетом Text() и обернем его в Padding и установим отступы: 16px по горизонтали и 20px по вертикали – листинг 49.

Листинг 49 – Блок с описанием автомобиля

```
Padding(  
  padding: const EdgeInsets.symmetric(  
    horizontal: 16, vertical: 20),  
  child: Text(  
    "Experience the ultimate combination of luxury, power, and precision with the BMW M5 F90. "  
    "This high-performance sports sedan features a twin-turbocharged 4.4L V8 engine, delivering an  
    exhilarating 600 horsepower for an unforgettable drive. "  
    "Equipped with BMW's signature xDrive all-wheel-drive system, the M5 F90 ensures both stability  
    and thrilling speed in any road condition.\n\n"  
    "Inside, enjoy a sophisticated, tech-forward interior with premium leather seating, a state-of-the-art  
    infotainment system, and advanced safety features. "  
    "Whether you're navigating city streets or taking a weekend getaway, the BMW M5 F90 offers  
    unparalleled comfort and driving dynamics.\n\n"  
    "Book today and elevate your driving experience with the power and luxury of the M5 F90!",  
    style: const TextStyle(  
      color: Color(0xFFFFBD16),  
      fontStyle: FontStyle.italic,  
      fontSize: 14,  
    ),  
  ),  
)
```

```

fontWeight: FontWeight.normal,
color:
  Color(0xFF848FAC),
fontSize: 15,
fontFamily:
  'Urbanist',
),
textAlign: TextAlign
.start,
),
),

```

После – создадим блок с информацией об автомобиле. Для этого, при помощи виджета Text() создадим текст «CAR INFO» и обернем его в Padding() и настроим отступы: по горизонтали – 16px, сверху – 5px, снизу – 20px. Так как следующие блоки по дизайну будут повторяться, то создадим static переменную типа Widget – infoRow, которая будет принимать:

- Label – название блока типа String!
- Value - значение типа String.

Возвращать будет Row, с Text() – название блока и Text(), содержащий значение поля – листинг 50.

Листинг 50 – Info Row

```

static Widget infoRow({required String label, required String value}) {
  return Row(
    mainAxisAlignment: MainAxisAlignment.spaceBetween,
    children: [
      Text(
        label,
        style: const TextStyle(
          fontSize: 18,
          color: Color(0xFF848FAC),
          fontWeight: FontWeight.normal,
        ),
      ),
      Text(
        value,
        style: const TextStyle(
          fontSize: 18,
          color: Color(0xFF192252),
          fontWeight: FontWeight.bold,
        ),
      ),
    ],
  );
}

```


Добавим Column и поместим 5 infoRow в него. Между infoRow установим отступ при помощи SizedBox() в размере 20px. Обернем Column в Padding и установим отступы – по горизонтали 16px, а снизу 150px – листинг 51.

Поля в infoRow:

- Engine;
- Power;
- Fuel;
- Color;
- Drivetrain.

Листинг 51 – блок с infoRow

```
Padding(  
  padding: const EdgeInsets.fromLTRB(16, 0, 16, 150),  
  child: Column(  
    children: [  
      infoRow(  
        label: "Engine", value: "4.4L twin-turbo V8"),  
        const SizedBox(height: 20),  
        infoRow(label: "Power", value: "600hp"),  
        const SizedBox(height: 20),  
        infoRow(label: "Fuel", value: "Casoline"),  
        const SizedBox(height: 20),  
        infoRow(label: "Color", value: "Dark Blue"),  
        const SizedBox(height: 20),  
        infoRow(  
          label: 'Drivetrain',  
          value: "xDrive All-Wheel Drive"),  
      ],  
    ),  
  ),  
)
```

Листинг 52 – SilverToBoxAdapter

```
SilverToBoxAdapter(  
  child: Container(  
    color: Colors.white,  
    child: Column(  
      crossAxisAlignment: CrossAxisAlignment.start,  
      children: [  
        Padding(  
          padding: const EdgeInsets.all(16),  
          child: Text(  
            "BMW M5 F90, 2022",  
            style: const TextStyle(  
              fontSize: 25,            ),  
          ),  
        ],  
      ),  
    ),  
  ),  
)
```

```

fontWeight: FontWeight.w900,
color: Color(0xFF192252),
),
),
),
Padding(
padding: const EdgeInsets.symmetric(horizontal: 16),
child: Row(
children: [
const ImageIcon(
AssetImage('assets/images/star.png'),
size: 30,
color: Color(0xFFFFBD16),
),
Padding(
padding: EdgeInsets.fromLTRB(3, 0, 10, 0),
child: Text(
'5.0',
style: const TextStyle(
fontSize: 20,
color: Color(0xFFFFBD16),
),
),
),
Text(
'(100+ reviews)',
style: const TextStyle(
fontSize: 15,
color: Color(0xFF192252),
),
),
),
],
),
),
Padding(
padding: const EdgeInsets.symmetric(
horizontal: 16, vertical: 20),
child: Text(
"Experience the ultimate combination of luxury, power, and precision with the BMW M5 F90. "
"This high-performance sports sedan features a twin-turbocharged 4.4L V8 engine, delivering an
exhilarating 600 horsepower for an unforgettable drive. "
"Equipped with BMW's signature xDrive all-wheel-drive system, the M5 F90 ensures both stability
and thrilling speed in any road condition.\n\n"
"Inside, enjoy a sophisticated, tech-forward interior with premium leather seating, a state-of-the-art
infotainment system, and advanced safety features. "
"Whether you're navigating city streets or taking a weekend getaway, the BMW M5 F90 offers
unparalleled comfort and driving dynamics.\n\n"
"Book today and elevate your driving experience with the power and luxury of the M5 F90!",
style: const TextStyle(
fontWeight: FontWeight.normal,
color: Color(0xFF848FAC),
fontSize: 15,
fontFamily: 'Urbanist',

```



```

        shape: BoxShape.circle,
        color: Colors.white,
      ),
      child: const Center(
        child: Icon(
          Icons.chevron_left,
          size: 30,
          color: Color(0xFF192252),
        ),
      ),
    ),
  ),
  expandedHeight: _calculateDynamicHeight(context),
  floating: false,
  pinned: true,
  flexibleSpace: LayoutBuilder(
    builder: (BuildContext context, BoxConstraints constraints) {
      var top = constraints.biggest.height;
      return FlexibleSpaceBar(
        title: AnimatedOpacity(
          opacity: top <= kToolbarHeight + 50 ? 1.0 : 0.0,
          duration: const Duration(milliseconds: 300),
        ),
        background: Stack(
          fit: StackFit.expand,
          children: [
            PageView(
              controller: _pageController,
              children: [
                Image.asset(
                  'assets/images/bmwm5.jpg',
                  fit: BoxFit.cover,
                ),
                Image.asset(
                  'assets/images/bmwm5.jpg',
                  fit: BoxFit.cover,
                ),
              ],
            ),
          ],
        ),
        Align(
          alignment: Alignment.bottomCenter,
          child: Padding(
            padding: const EdgeInsets.only(bottom: 16),
            child: SmoothPageIndicator(
              controller: _pageController,
              count: 2,
              effect: const ExpandingDotsEffect(
                expansionFactor: 4.0,
                activeDotColor: Color(0xFF192252),
                dotColor: Color(0xFFD9D9D9),
                dotHeight: 8,
                dotWidth: 8,

```

```
        spacing: 8.0,
      ),
    ),
  ),
),
IgnorePointer(
  child: Container(
    color: Colors.white.withOpacity(
      top <= kToolbarHeight + 50 ? 1.0 : 0.0,
    ),
  ),
),
],
),
);
},
),
),
SliverToBoxAdapter(
  child: Container(
    color: Colors.white,
    child: Column(
      crossAxisAlignment: CrossAxisAlignment.start,
      children: [
        Padding(
          padding: const EdgeInsets.all(16),
          child: Text(
            "BMW M5 F90, 2022",
            style: const TextStyle(
              fontSize: 25,
              fontWeight: FontWeight.w900,
              color: Color(0xFF192252),
            ),
          ),
        ),
        Padding(
          padding: const EdgeInsets.symmetric(horizontal: 16),
          child: Row(
            children: [
              const ImageIcon(
                AssetImage('assets/images/star.png'),
                size: 30,
                color: Color(0xFFFFFBD16),
              ),
              Padding(
                padding: EdgeInsets.fromLTRB(3, 0, 10, 0),
                child: Text(
                  '5.0',
                  style: const TextStyle(
                    fontSize: 20,
                    color: Color(0xFFFFFBD16),
                  ),
                ),
              ),
            ],
          ),
        ),
      ],
    ),
  ),
),
```

```

    ),
    Text(
      "(100+ reviews)",
      style: const TextStyle(
        fontSize: 15,
        color: Color(0xFF192252),
      ),
    ),
  ],
),
),
Padding(
  padding: const EdgeInsets.symmetric(
    horizontal: 16, vertical: 20),
  child: Text(
    "Experience the ultimate combination of luxury, power, and precision with the BMW M5 F90. "
    "This high-performance sports sedan features a twin-turbocharged 4.4L V8 engine, delivering an
    exhilarating 600 horsepower for an unforgettable drive. "
    "Equipped with BMW's signature xDrive all-wheel-drive system, the M5 F90 ensures both stability
    and thrilling speed in any road condition.\n\n"
    "Inside, enjoy a sophisticated, tech-forward interior with premium leather seating, a state-of-the-art
    infotainment system, and advanced safety features. "
    "Whether you're navigating city streets or taking a weekend getaway, the BMW M5 F90 offers
    unparalleled comfort and driving dynamics.\n\n"
    "Book today and elevate your driving experience with the power and luxury of the M5 F90!",
    style: const TextStyle(
      fontWeight: FontWeight.normal,
      color: Color(0xFF848FAC),
      fontSize: 15,
      fontFamily: 'Urbanist',
    ),
    textAlign: TextAlign.start,
  ),
),
),
Padding(
  padding: EdgeInsets.fromLTRB(16, 5, 16, 20),
  child: Text(
    "CAR INFO",
    style: TextStyle(
      fontSize: 20,
      fontWeight: FontWeight.bold,
      color: Color(0xFF192252),
    ),
  ),
)),
Padding(
  padding: const EdgeInsets.fromLTRB(16, 0, 16, 150),
  child: Column(
    children: [
      infoRow(
        label: "Engine", value: "4.4L twin-turbo V8"),
      const SizedBox(height: 20),
      infoRow(label: "Power", value: "600hp"),
      const SizedBox(height: 20),
    ],
  ),
),

```

```
infoRow(label: "Fuel", value: "Casoline"),
const SizedBox(height: 20),
infoRow(label: "Color", value: "Dark Blue"),
const SizedBox(height: 20),
infoRow(
  label: 'Drivetrain',
  value: "xDrive All-Wheel Drive"),
],
),
],
),
),
],
),
```

Теперь создадим блок с ценой автомобиля и кнопкой аренды. Для этого создадим контейнер – `Container()`, в который поместим `Row()`, содержащий в себе `Column` с названием автомобиля и ценой его аренды, а также кнопку `ElevatedButton()`. Обернем весь `Container` в `Positioned`, чтобы зафиксировать его снизу экрана – листинг 53.

Листинг 53 – Блок с кнопкой аренды автомобиля

```
Positioned(
  bottom: 0,
  left: 0,
  right: 0,
  child: Container(
    color: Colors.white,
    padding: const EdgeInsets.symmetric(horizontal: 16, vertical: 20),
    child: Padding(
      padding: const EdgeInsets.only(bottom: 16),
      child: Row(
        mainAxisAlignment: MainAxisAlignment.spaceBetween,
        children: [
          Column(
            crossAxisAlignment: CrossAxisAlignment.start,
            children: [
              Text(
                "BMW M5 F90, 2022",
                style: const TextStyle(
                  fontSize: 20,
                  fontWeight: FontWeight.bold,
                  color: Color(0xFF192252),
                ),
              ),
              const SizedBox(height: 8),
              Text(
                '22 000 ₪ / day',
                style: const TextStyle(

```



```

    }
}

@override
void dispose() {
  _pageController.dispose();
  super.dispose();
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    body: Stack(
      children: [
        CustomScrollView(
          slivers: [
            SliverAppBar(
              automaticallyImplyLeading: false,
              leading: Padding(
                padding: const EdgeInsets.only(left: 24),
                child: InkWell(
                  onTap: () {
                    Navigator.pop(context);
                  },
                  borderRadius: BorderRadius.circular(25),
                  child: Container(
                    width: 50,
                    height: 50,
                    decoration: const BoxDecoration(
                      shape: BoxShape.circle,
                      color: Colors.white,
                    ),
                    child: const Center(
                      child: Icon(
                        Icons.chevron_left,
                        size: 30,
                        color: Color(0xFF192252),
                      ),
                    ),
                  ),
                ),
              ),
            ),
          ],
          expandedHeight: _calculateDynamicHeight(context),
          floating: false,
          pinned: true,
          flexibleSpace: LayoutBuilder(
            builder: (BuildContext context, BoxConstraints constraints) {
              var top = constraints.biggest.height;
              return FlexibleSpaceBar(
                title: AnimatedOpacity(
                  opacity: top <= kToolbarHeight + 50 ? 1.0 : 0.0,
                  duration: const Duration(milliseconds: 300),
                ),
              ),
            ),
          ),
        ),
      ],
    ),
  );
}

```

```

background: Stack(
  fit: StackFit.expand,
  children: [
    PageView(
      controller: _pageController,
      children: [
        Image.asset(
          'assets/images/bmw5.jpg',
          fit: BoxFit.cover,
        ),
        Image.asset(
          'assets/images/bmw5.jpg',
          fit: BoxFit.cover,
        ),
      ],
    ),
    Align(
      alignment: Alignment.bottomCenter,
      child: Padding(
        padding: const EdgeInsets.only(bottom: 16),
        child: SmoothPageIndicator(
          controller: _pageController,
          count: 2,
          effect: const ExpandingDotsEffect(
            expansionFactor: 4.0,
            activeDotColor: Color(0xFF192252),
            dotColor: Color(0xFFD9D9D9),
            dotHeight: 8,
            dotWidth: 8,
            spacing: 8.0,
          ),
        ),
      ),
    ),
    IgnorePointer(
      child: Container(
        color: Colors.white.withOpacity(
          top <= kToolbarHeight + 50 ? 1.0 : 0.0,
        ),
      ),
    ),
  ],
),
);
},
),
),
SliverToBoxAdapter(
  child: Container(
    color: Colors.white,
    child: Column(
      crossAxisAlignment: CrossAxisAlignment.start,
      children: [

```

```

Padding(
  padding: const EdgeInsets.all(16),
  child: Text(
    "BMW M5 F90, 2022",
    style: const TextStyle(
      fontSize: 25,
      fontWeight: FontWeight.w900,
      color: Color(0xFF192252),
    ),
  ),
),
Padding(
  padding: const EdgeInsets.symmetric(horizontal: 16),
  child: Row(
    children: [
      const ImageIcon(
        AssetImage('assets/images/star.png'),
        size: 30,
        color: Color(0xFFFFBD16),
      ),
      Padding(
        padding: EdgeInsets.fromLTRB(3, 0, 10, 0),
        child: Text(
          '5.0',
          style: const TextStyle(
            fontSize: 20,
            color: Color(0xFFFFBD16),
          ),
        ),
      ),
      Text(
        "(100+ reviews)",
        style: const TextStyle(
          fontSize: 15,
          color: Color(0xFF192252),
        ),
      ),
    ],
  ),
),
Padding(
  padding: const EdgeInsets.symmetric(
    horizontal: 16, vertical: 20),
  child: Text(
    "Experience the ultimate combination of luxury, power, and precision with the BMW M5 F90. "
    "This high-performance sports sedan features a twin-turbocharged 4.4L V8 engine, delivering an exhilarating 600 horsepower for an unforgettable drive. "
    "Equipped with BMW's signature xDrive all-wheel-drive system, the M5 F90 ensures both stability and thrilling speed in any road condition.\n\n"
    "Inside, enjoy a sophisticated, tech-forward interior with premium leather seating, a state-of-the-art infotainment system, and advanced safety features. "
    "Whether you're navigating city streets or taking a weekend getaway, the BMW M5 F90 offers unparalleled comfort and driving dynamics.\n\n"

```

```

        "Book today and elevate your driving experience with the power and luxury of the M5 F90!",
        style: const TextStyle(
          fontWeight: FontWeight.normal,
          color: Color(0xFF848FAC),
          fontSize: 15,
          fontFamily: 'Urbanist',
        ),
        textAlign: TextAlign.start,
      ),
    ),
    Padding(
      padding: EdgeInsets.fromLTRB(16, 5, 16, 20),
      child: Text(
        "CAR INFO",
        style: TextStyle(
          fontSize: 20,
          fontWeight: FontWeight.bold,
          color: Color(0xFF192252),
        ),
      ),
    ),
    Padding(
      padding: const EdgeInsets.fromLTRB(16, 0, 16, 150),
      child: Column(
        children: [
          infoRow(
            label: "Engine", value: "4.4L twin-turbo V8"),
          const SizedBox(height: 20),
          infoRow(label: "Power", value: "600hp"),
          const SizedBox(height: 20),
          infoRow(label: "Fuel", value: "Casoline"),
          const SizedBox(height: 20),
          infoRow(label: "Color", value: "Dark Blue"),
          const SizedBox(height: 20),
          infoRow(
            label: 'Drivetrain',
            value: "xDrive All-Wheel Drive"),
        ],
      ),
    ),
  ],
),
),
],
),
),
),
),
],
),
Positioned(
  bottom: 0,
  left: 0,
  right: 0,
  child: Container(
    color: Colors.white,
    padding: const EdgeInsets.symmetric(horizontal: 16, vertical: 20),
    child: Padding(

```

```

padding: const EdgeInsets.only(bottom: 16),
child: Row(
  mainAxisAlignment: MainAxisAlignment.spaceBetween,
  children: [
    Column(
      crossAxisAlignment: CrossAxisAlignment.start,
      children: [
        Text(
          "BMW M5 F90, 2022",
          style: const TextStyle(
            fontSize: 20,
            fontWeight: FontWeight.bold,
            color: Color(0xFF192252),
          ),
        ),
        const SizedBox(height: 8),
        Text(
          '22 000 ₺ / day',
          style: const TextStyle(
            fontSize: 15,
            color: Color(0xFF192252),
          ),
        ),
      ],
    ),
    ElevatedButton(
      style: ElevatedButton.styleFrom(
        backgroundColor: const Color(0xFF192252),
        shape: RoundedRectangleBorder(
          borderRadius: BorderRadius.circular(10),
        ),
      ),
      onPressed: () {},
      child: Padding(
        padding:
          EdgeInsets.symmetric(vertical: 20, horizontal: 14),
        child: Text(
          "Rent Car",
          style: TextStyle(fontSize: 12, color: Colors.white),
        ),
      ),
    ),
  ],
),
),
],
);
}

static Widget infoRow({required String label, required String value}) {

```

```
return Row(  
  mainAxisAlignment: MainAxisAlignment.spaceBetween,  
  children: [  
    Text(  
      label,  
      style: const TextStyle(  
        fontSize: 18,  
        color: Color(0xFF848FAC),  
        fontWeight: FontWeight.normal,  
      ),  
    ),  
    Text(  
      value,  
      style: const TextStyle(  
        fontSize: 18,  
        color: Color(0xFF192252),  
        fontWeight: FontWeight.bold,  
      ),  
    ),  
  ],  
);  
}
```

На рисунках 22 и 25 представлены разработанный экран с детальной информацией на мобильном устройстве и в chrome.



BMW M5 F90, 2022

★ 5.0 (100+ reviews)

Experience the ultimate combination of luxury, power, and precision with the BMW M5 F90. This high-performance sports sedan features a twin-turbocharged 4.4L V8 engine, delivering an exhilarating 600 horsepower for an unforgettable drive. Equipped with BMW's signature xDrive all-wheel-drive system, the M5 F90 ensures both stability and thrilling speed in any road condition.

Inside, enjoy a sophisticated, tech-forward interior with premium leather seating, a state-of-the-art infotainment system, and advanced safety features. Whether you're navigating city streets or taking a weekend getaway, the BMW M5 F90 offers unparalleled comfort and driving dynamics.

Book today and elevate your driving experience with the power and luxury of the M5 F90!

BMW M5 F90, 2022

22 000 ₽ / day

Rent Car

Рисунок 22 – Разработанный экран в симуляторе iOS (часть 1)



Рисунок 23 – Разработанный экран в симуляторе iOS (часть 2)

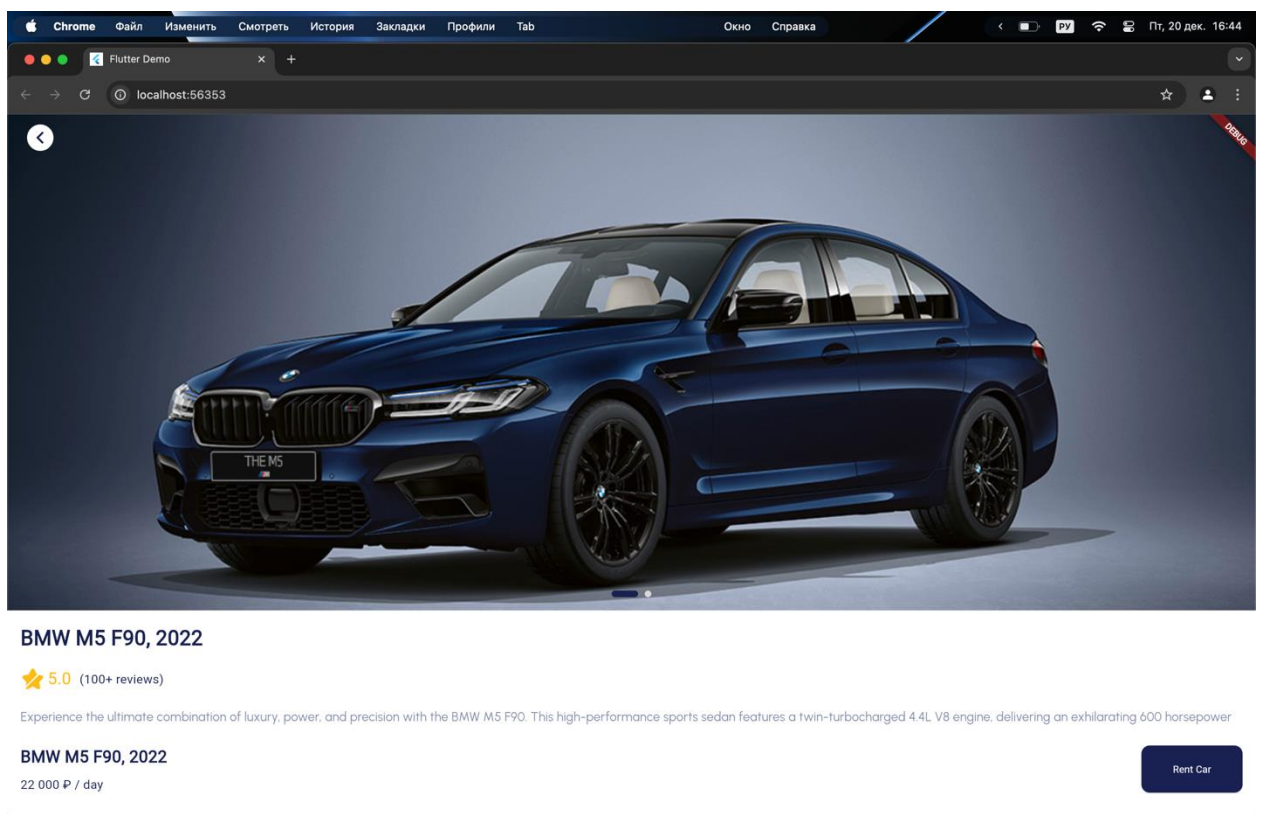


Рисунок 24 – Разработанный экран в chrome (часть 1)

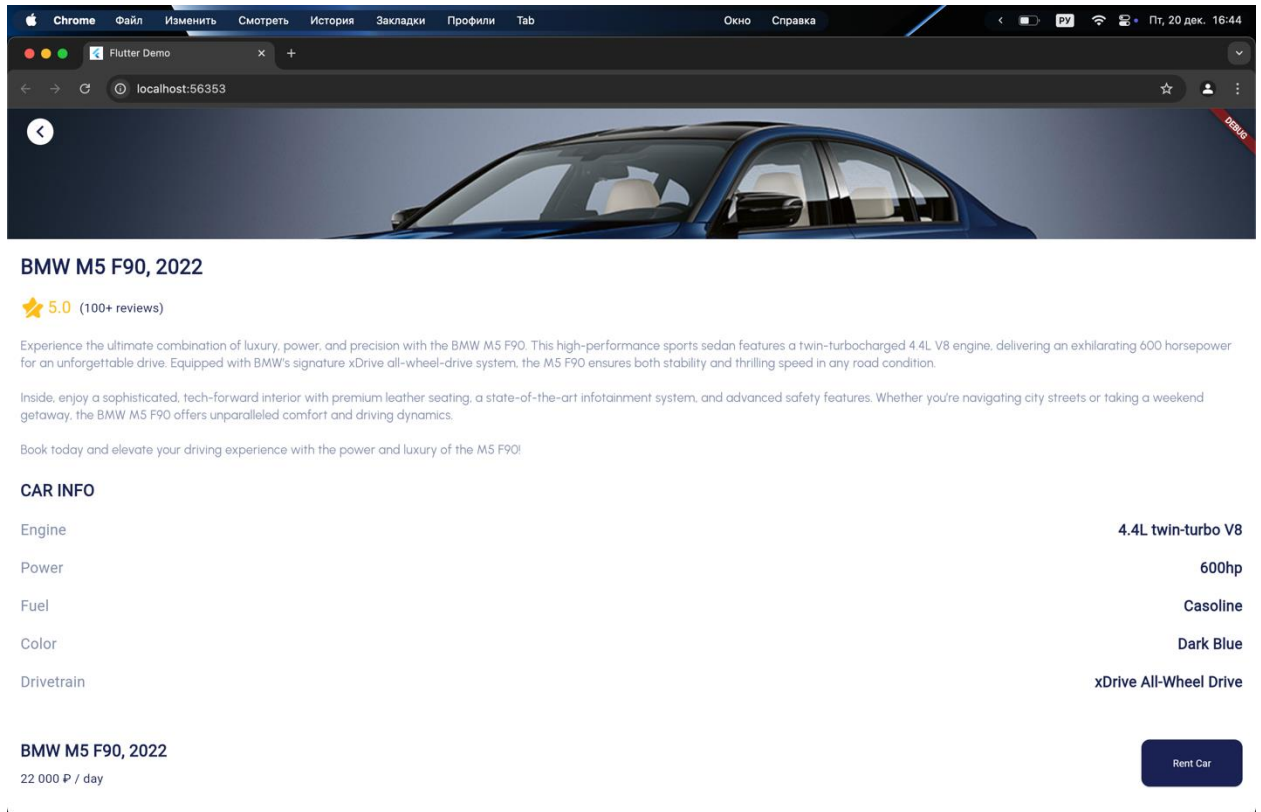


Рисунок 25 – Разработанный экран в chrome (часть 2)

2.3.6. Верстка экрана профиля

В папке `pages` создаем файл `profile_page.dart`. Импортируем в него необходимые библиотеки и файлы, создаем класс и настроим каркас экрана—
листинг 55.

Листинг 55 – Базовая настройка `profile_page.dart`

```
import 'package:flutter/material.dart';
import 'package:flutter/foundation.dart';
import 'dart:io';

class ProfilePage extends StatefulWidget {
  const ProfilePage({super.key});

  @override
  _ProfilePageState createState() => _ProfilePageState();
}

class _ProfilePageState extends State<ProfilePage> {

  @override
  void initState() {
    super.initState();
  }
}
```

```

@override
Widget build(BuildContext context) {
  bool isMobile = !kIsWeb && (Platform.isAndroid || Platform.isIOS);

  Widget content = Column(
    children: []
  );

  return Scaffold(
    appBar: AppBar(
      backgroundColor: Colors.white,
      elevation: 0,
      title: Text(
        "Profile",
        style: TextStyle(
          color: Color(0xFF192252),
          fontSize: 25,
          fontWeight: FontWeight.bold,
        ),
      ),
      centerTitle: false,
    ),
    body: Stack(
      children: [
        Container(
          color: Colors.white,
          child: Padding(
            padding: const EdgeInsets.symmetric(horizontal: 16.0),
            child: isMobile
              ? content
              : SingleChildScrollView(
                  child: content,
                ),
          ),
        ),
      ],
    ),
  );
}

```

Для начала добавим appBar в Scaffold. Он у нас содержит только Text() «Profile» - Листинг 56.

Листинг 56 – Создание appBar

```

appBar: AppBar(
  backgroundColor: Colors.white,
  elevation: 0,
  title: Text(
    "Profile",
    style: TextStyle(
      color: Color(0xFF192252),

```

```

        fontSize: 25,
        fontWeight: FontWeight.bold,
      ),
    ),
    centerTitle: false,
  ),

```

Перейдем к переменной `content`. Она будет содержать в себе `Container` с отступами внутри по всем осям 25px, также он представляет из себя закругленный прямоугольник с закруглением 20px и заливкой линейным градиентом в направлении: слева направо, а также тенью. Внутри контейнера находится блок с информацией о пользователе, который помещен в `Row`. В начале `Row` идет изображение пользователя обернутое в круг радиусом 24, дальше идет полное имя пользователя, после идет `Spacer()`, чтобы расположить иконку редактирования справа – Листинг 57.

Листинг 57 – Контейнер с информацией о пользователе

```

Container(
  padding: const EdgeInsets.all(25.0),
  decoration: BoxDecoration(
    borderRadius: BorderRadius.circular(20),
    gradient: const LinearGradient(
      colors: [Color(0xFF1B588C), Color(0xFF848FAC)],
      begin: Alignment.centerLeft,
      end: Alignment.centerRight,
    ),
    boxShadow: const [
      BoxShadow(
        color: Colors.black26,
        blurRadius: 10,
        offset: Offset(0, 4),
      )
    ],
  ),
  child: Row(
    children: [
      ClipOval(
        child: SizedBox.fromSize(
          size: const Size.fromRadius(24),
          child: Image.asset('assets/images/userPhoto.jpg')),
      ),
      const SizedBox(width: 13),
      Text(
        'Anna Montana',
        style: const TextStyle(
          fontSize: 20,
          fontWeight: FontWeight.bold,
          color: Colors.white,

```

```

    ),
    overflow: TextOverflow.ellipsis,
    maxLines: 1,
  ),
  const Spacer(),
  const ImageIcon(
    AssetImage('assets/images/edit.png'),
    color: Colors.white,
    size: 24,
  ),
],
),
),
),

```

Дальше у нас будут одинаковые кнопки, поэтому создадим отдельный компонент CustomButton в папке components. Кнопка принимает иконку, название и действие. Представляет собой Container с закругленными углами и тенью, и Row() внутри, который содержит иконку и текст – листинг 58.

Листинг 58 – CustomButton

```

import 'package:flutter/material.dart';

class CustomButton extends StatelessWidget {
  final String iconPath;
  final String label;
  final VoidCallback onTap;

  const CustomButton({
    super.key,
    required this.iconPath,
    required this.label,
    required this.onTap,
  });

  @override
  Widget build(BuildContext context) {
    return InkWell(
      onTap: onTap,
      borderRadius: BorderRadius.circular(15),
      child: Container(
        padding: const EdgeInsets.symmetric(vertical: 18, horizontal: 16),
        decoration: BoxDecoration(
          color: Colors.white,
          borderRadius: BorderRadius.circular(10),
          boxShadow: [
            BoxShadow(
              color: Colors.grey.withOpacity(0.2),
              spreadRadius: 2,
              blurRadius: 5,
              offset: const Offset(0, 3),
            ),
          ],
        ),
      ),
    );
  }
}

```

```

    ],
  ),
  child: Row(
    children: [
      Image.asset(
        iconPath,
        width: 24,
        height: 24,
        fit: BoxFit.cover,
      ),
      const SizedBox(width: 13),
      Text(
        label,
        style: const TextStyle(
          fontSize: 20,
          fontWeight: FontWeight.bold,
          color: Color(0xFF192252),
        ),
      ),
    ],
  ),
],
),
);
}
}

```

Добавим в profile_page.dart нашу кастомную кнопку:

```
import '../Components/custom_button.dart';
```

Теперь создадим 3 кнопки:

- Change Password;
- Billing Information;
- Notification.

Для этого произведем вызов CustomButton() и передадим в него путь до нужного изображения, название и действие. Между кнопками добавим отступы в размере 21px – листинг 59.

Листинг 59 – добавление CustomButton

```

const SizedBox(height: 30),
CustomButton(
  iconPath: 'assets/images/lock-circle.png',
  label: "Change password",
  onTap: () {},
),
const SizedBox(height: 21),
CustomButton(
  iconPath: 'assets/images/empty-wallet.png',
  label: "Billing information",
  onTap: () {},

```

```

),
const SizedBox(height: 21),
CustomButton(
  iconPath: 'assets/images/direct-inbox.png',
  label: "Notifications",
  onTap: () {},
),
isMobile ? const Spacer() : const SizedBox(height: 250),

```

Теперь создадим кнопки LogOut и Delete Account. Для этого создадим еще одну кастомную кнопку CustomActionButton. Для этого в components создаем новый файл custom_action_button. Содержит следующие поля:

- Label;
- isPrimary;
- Действие.

Сама кнопка представляет собой контейнер с текстом внутри. Цвет контейнера и текста зависит от значения isPrimary – листинг 60.

Листинг 60 – Custom Action Button

```

import 'package:flutter/material.dart';

class CustomActionButton extends StatelessWidget {
  final String label;
  final bool isPrimary;
  final VoidCallback onTap;

  const CustomActionButton({
    super.key,
    required this.label,
    required this.isPrimary,
    required this.onTap,
  });

  @override
  Widget build(BuildContext context) {
    return InkWell(
      onTap: onTap,
      borderRadius: BorderRadius.circular(15),
      child: Container(
        padding: const EdgeInsets.symmetric(vertical: 16, horizontal: 16),
        decoration: BoxDecoration(
          color: isPrimary ? const Color(0xFF192252) : Colors.white,
          borderRadius: BorderRadius.circular(15),
          boxShadow: [
            BoxShadow(
              color: Colors.grey.withOpacity(0.2),
              spreadRadius: 2,
              blurRadius: 5,

```

```

        offset: const Offset(0, 3),
      ),
    ],
  ),
  child: Center(
    child: Text(
      label,
      style: TextStyle(
        fontSize: 20,
        fontWeight: FontWeight.bold,
        color: isPrimary ? Colors.white : const Color(0xFFB7264F),
      ),
    ),
  ),
),
);
}
}

```

Добавим в profile_page.dart нашу CustomActionButton:

```
import '../Components/custom_action_button.dart';
```

Теперь добавим их в Column – листинг 61

Листинг 61 – Добавление Custom Action Button в profile_page.dart

```

CustomActionButton(
  label: "Log Out",
  isPrimary: true,
  onTap: () async {},
),
const SizedBox(height: 20),
CustomActionButton(
  label: "Delete account",
  isPrimary: false,
  onTap: () async {},
),
isMobile ? const SizedBox(height: 50) : const SizedBox(height: 0)

```

Листинг 62 – Полный код экрана

```

import 'package:flutter/material.dart';
import '../Components/custom_button.dart';
import '../Components/custom_action_button.dart';
import 'package:flutter/foundation.dart';
import 'dart:io';

class ProfilePage extends StatefulWidget {
  const ProfilePage({super.key});

  @override
  _ProfilePageState createState() => _ProfilePageState();
}

class _ProfilePageState extends State<ProfilePage> {

```

```

@override
void initState() {
  super.initState();
}

@override
Widget build(BuildContext context) {
  bool isMobile = !kIsWeb && (Platform.isAndroid || Platform.isIOS);

  Widget content = Column(
    children: [
      Container(
        padding: const EdgeInsets.all(25.0),
        decoration: BoxDecoration(
          borderRadius: BorderRadius.circular(20),
          gradient: const LinearGradient(
            colors: [Color(0xFF1B588C), Color(0xFF848FAC)],
            begin: Alignment.centerLeft,
            end: Alignment.centerRight,
          ),
        ),
        boxShadow: const [
          BoxShadow(
            color: Colors.black26,
            blurRadius: 10,
            offset: Offset(0, 4),
          )
        ],
      ),
      child: Row(
        children: [
          ClipOval(
            child: SizedBox.fromSize(
              size: const Size.fromRadius(24),
              child: Image.asset('assets/images/userPhoto.jpg')),
          ),
          const SizedBox(width: 13),
          Text(
            'Anna Montana',
            style: const TextStyle(
              fontSize: 20,
              fontWeight: FontWeight.bold,
              color: Colors.white,
            ),
            overflow: TextOverflow.ellipsis,
            maxLines: 1,
          ),
          const Spacer(),
          const ImageIcon(
            AssetImage('assets/images/edit.png'),
            color: Colors.white,
            size: 24,
          ),
        ],
      ),
    ],
  );

```



```

    ),
  ),
  const SizedBox(height: 30),
  CustomButton(
    iconPath: 'assets/images/lock-circle.png',
    label: "Change password",
    onTap: () {},
  ),
  const SizedBox(height: 21),
  CustomButton(
    iconPath: 'assets/images/empty-wallet.png',
    label: "Billing information",
    onTap: () {},
  ),
  const SizedBox(height: 21),
  CustomButton(
    iconPath: 'assets/images/direct-inbox.png',
    label: "Notifications",
    onTap: () {},
  ),
  isMobile ? const Spacer() : const SizedBox(height: 250),
  CustomActionButton(
    label: "Log Out",
    isPrimary: true,
    onTap: () async {},
  ),
  const SizedBox(height: 20),
  CustomActionButton(
    label: "Delete account",
    isPrimary: false,
    onTap: () async {},
  ),
  isMobile ? const SizedBox(height: 50) : const SizedBox(height: 0)
],
);

return Scaffold(
  appBar: AppBar(
    backgroundColor: Colors.white,
    elevation: 0,
    title: Text(
      "Profile",
      style: TextStyle(
        color: Color(0xFF192252),
        fontSize: 25,
        fontWeight: FontWeight.bold,
      ),
    ),
  ),
  centerTitle: false,
),
body: Stack(
  children: [
    Container(

```

```

color: Colors.white,
child: Padding(
  padding: const EdgeInsets.symmetric(horizontal: 16.0),
  child: isMobile
    ? content
    : SingleChildScrollView(
        child: content,
      ),
),
),
],
),
);
}
}

```

На рисунках 26 и 27 представлены разработанный экран профиля на мобильном устройстве и в chrome.

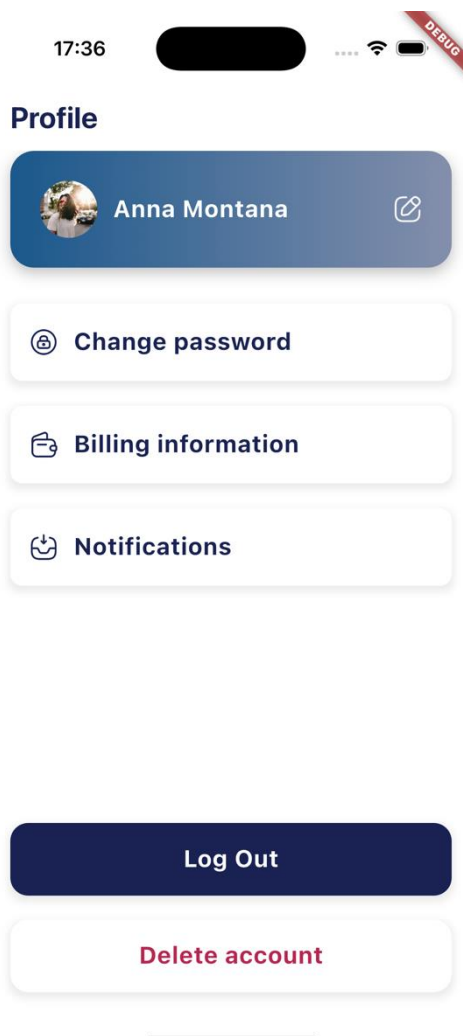


Рисунок 26 – Разработанный экран в симуляторе iOS

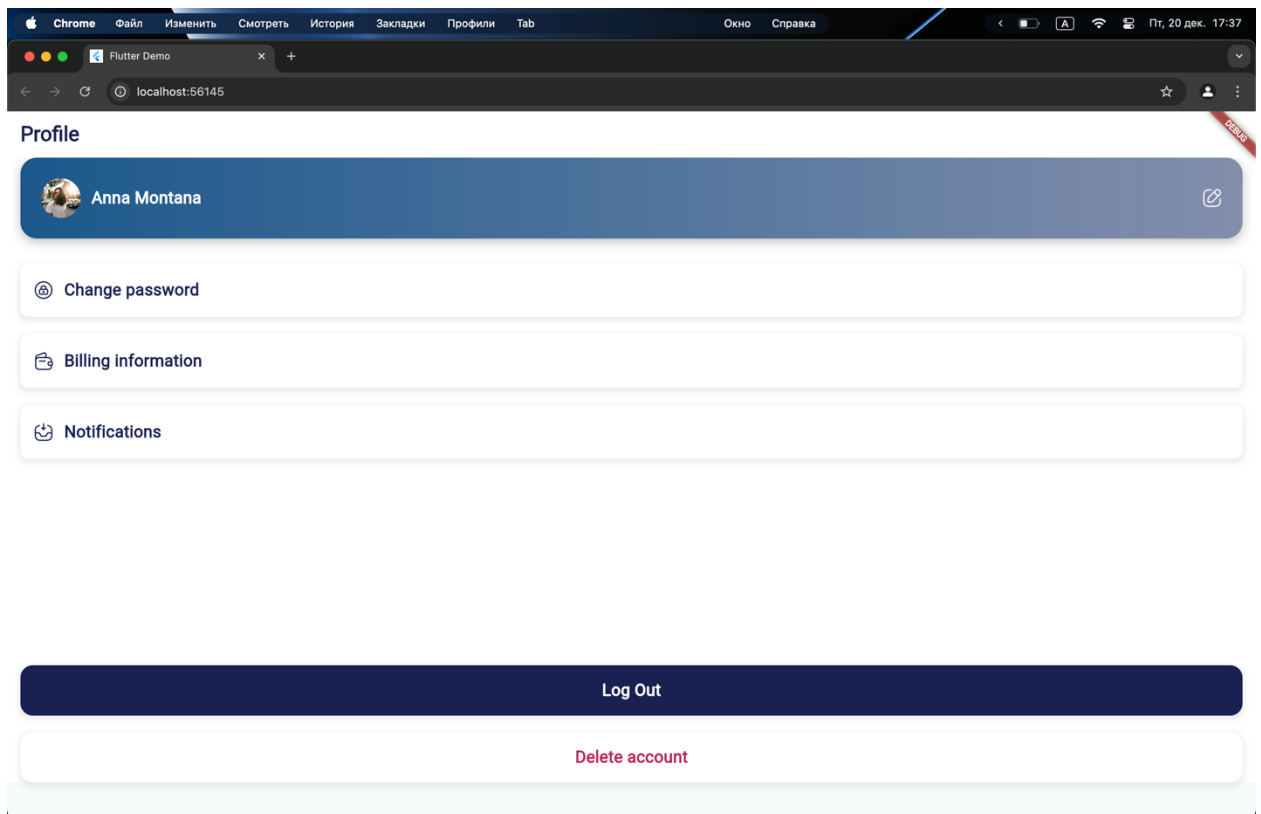


Рисунок 27 – Разработанный экран в chrome

2.3.7. Верстка экрана с изменением данных

В папке `pages` создаем файл `edit_user_page.dart`. Импортируем в него необходимые библиотеки и файлы, создаем класс и настроим каркас экрана—
листинг 63.

Листинг 63 – Базовая настройка экрана

```
import 'dart:io';
import 'package:flutter/material.dart';
import '../Components/text_field.dart';
import '../Components/custom_action_button.dart';
import 'package:flutter/foundation.dart';

class EditUserPage extends StatefulWidget {

  const EditUserPage({super.key});

  @override
  _EditUserPageState createState() => _EditUserPageState();
}

class _EditUserPageState extends State<EditUserPage> {
  @override
  void initState() {
    super.initState();
  }
}
```

```

@override
void dispose() {
  super.dispose();
}

@override
Widget build(BuildContext context) {
  bool isMobile = !kIsWeb && (Platform.isAndroid || Platform.isIOS);

  var content = Padding(
  );

  return Scaffold(
    backgroundColor: Colors.white,
    body: isMobile ? content : SingleChildScrollView(child: content),
  );
}
}

```

Создадим appBar, который будет содержать в себе ряд, выравненный по левой стороне, а внутри кнопка назад – листинг 65.

Листинг 65 – appBar

```

appBar: AppBar(
  backgroundColor: Colors.white,
  elevation: 0,
  automaticallyImplyLeading: false,
  title: Row(
    crossAxisAlignment: CrossAxisAlignment.start,
    children: [
      SizedBox(
        width: 50,
        height: 50,
        child: IconButton(
          icon: Image.asset('assets/images/backIcon.png'),
          onPressed: () {
            },
        ),
      ),
    ],
  ),
),

```

Основное тело состоит из следующих компонентов:

- Text() – «Update User Info»
- Контейнера с фотографией пользователя;
- Двух текстовых полей с изменением «Full name» и « Email address»;
- Кнопки обновления.

Листинг тела представлен в листинге 66.

Листинг 66 – Тело страницы

```
var content = Padding(  
  padding: const EdgeInsets.symmetric(vertical: 8, horizontal: 24),  
  child: Column(  
    children: [  
      Align(  
        alignment: Alignment.centerLeft,  
        child: Text(  
          "Update user info",  
          style: TextStyle(  
            fontWeight: FontWeight.bold,  
            fontSize: 25,  
            color: Color(0xFF192252),  
          ),  
        ),  
      ),  
      const SizedBox(height: 20),  
      Container(  
        width: 130,  
        height: 130,  
        decoration: BoxDecoration(  
          borderRadius: BorderRadius.circular(24),  
          image: const DecorationImage(  
            image: AssetImage(  
              'assets/images/userPhoto.jpg'),  
            fit: BoxFit.cover,  
          ),  
        ),  
      ),  
      child: Center(  
        child: Icon(  
          Icons.camera_alt,  
          color: Colors.white.withOpacity(0.7),  
          size: 50,  
        ),  
      ),  
    ],  
  ),  
  const SizedBox(height: 35),  
  CustomTextField(  
    labelText: "Full name",  
    initialText: "Enter your name",  
  ),  
  const SizedBox(height: 25),  
  CustomTextField(  
    labelText: "Email address",  
    initialText: "Enter your email",  
  ),  
  isMobile ? const Spacer() : SizedBox(height: 220),  
  CustomActionButton(  
    label: "Update info",  
    isPrimary: true,  
    onTap: () {},  
  ),  
  const SizedBox(height: 55),  
);
```

```
],  
,  
);
```

Полный код экрана представлен в листинге 67.

Листинг 67 – Полный код экрана

```
import 'dart:io';  
import 'package:flutter/material.dart';  
import '../Components/text_field.dart';  
import '../Components/custom_action_button.dart';  
import 'package:flutter/foundation.dart';  
  
class EditUserPage extends StatefulWidget {  
  
  const EditUserPage({super.key});  
  
  @override  
  _EditUserPageState createState() => _EditUserPageState();  
}  
  
class _EditUserPageState extends State<EditUserPage> {  
  @override  
  void initState() {  
    super.initState();  
  }  
  
  @override  
  void dispose() {  
    super.dispose();  
  }  
  
  @override  
  Widget build(BuildContext context) {  
    bool isMobile = !kIsWeb && (Platform.isAndroid || Platform.isIOS);  
  
    var content = Padding(  
      padding: const EdgeInsets.symmetric(vertical: 8, horizontal: 24),  
      child: Column(  
        children: [  
          Align(  
            alignment: Alignment.centerLeft,  
            child: Text(  
              "Update user info",  
              style: TextStyle(  
                fontWeight: FontWeight.bold,  
                fontSize: 25,  
                color: Color(0xFF192252),  
              ),  
            ),  
          ),  
          const SizedBox(height: 20),  
          Container(  
            width: 130,
```

```

        height: 130,
        decoration: BoxDecoration(
          borderRadius: BorderRadius.circular(24),
          image: const DecorationImage(
            image: AssetImage(
              'assets/images/userPhoto.jpg'),
            fit: BoxFit.cover,
          ),
        ),
        child: Center(
          child: Icon(
            Icons.camera_alt,
            color: Colors.white.withOpacity(0.7),
            size: 50,
          ),
        ),
      ),
      const SizedBox(height: 35),
      CustomTextField(
        labelText: "Full name",
        initialText: "Enter your name",
      ),
      const SizedBox(height: 25),
      CustomTextField(
        labelText: "Email address",
        initialText: "Enter your email",
      ),
      isMobile ? const Spacer() : SizedBox(height: 220),
      CustomActionButton(
        label: "Update info",
        isPrimary: true,
        onTap: () {},
      ),
      const SizedBox(height: 55),
    ],
  ),
);

return Scaffold(
  appBar: AppBar(
    backgroundColor: Colors.white,
    elevation: 0,
    automaticallyImplyLeading: false,
    title: Row(
      crossAxisAlignment: CrossAxisAlignment.start,
      children: [
        SizedBox(
          width: 50,
          height: 50,
          child: IconButton(
            icon: Image.asset('assets/images/backIcon.png'),
            onPressed: () {

```

```

    ),
  ),
  ],
),
),
backgroundColor: Colors.white,
body: isMobile ? content : SingleChildScrollView(child: content),
);
}
}

```

На рисунках 28 и 29 представлены разработанный экран изменения данных на мобильном устройстве и в chrome.

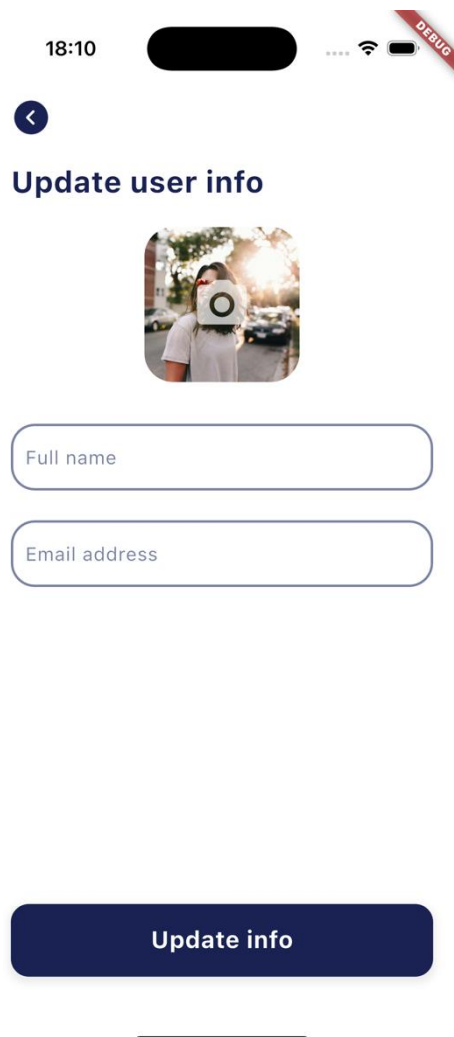


Рисунок 26 – Разработанный экран в симуляторе iOS

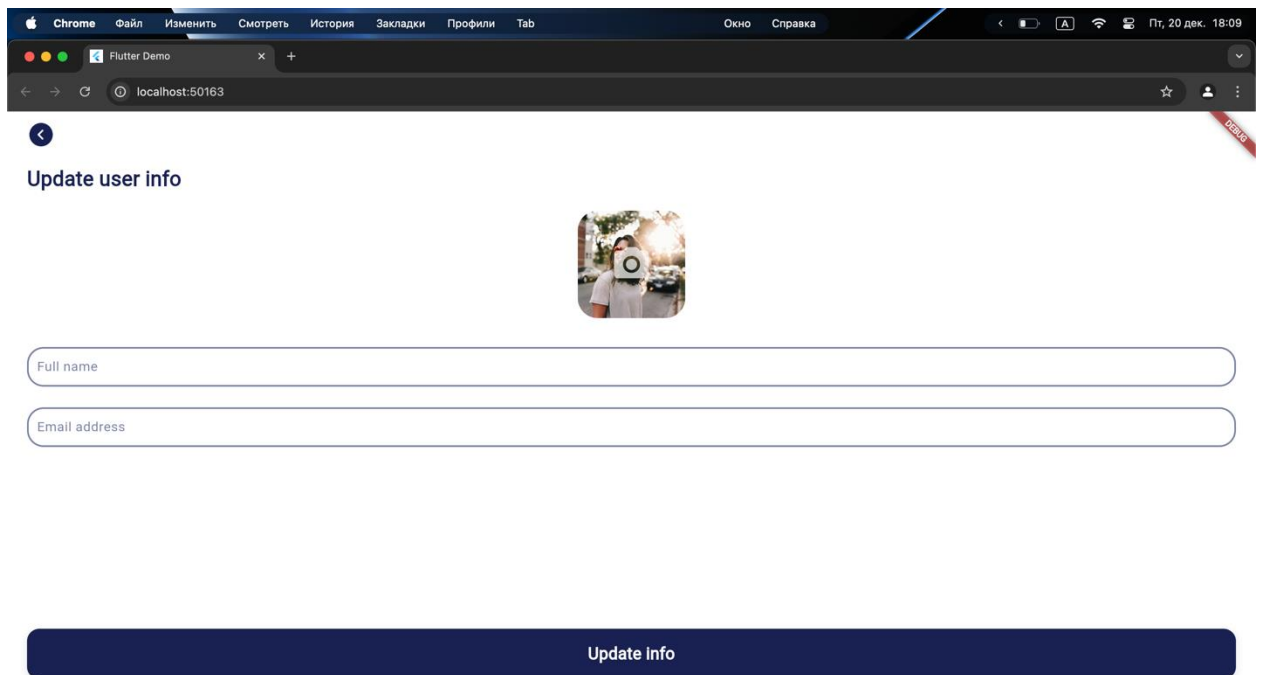


Рисунок 27 – Разработанный экран в chrome

Листинг 68 – Полный код pubspec.yaml

```
name: rent_car_project
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/Core
# FoundationKeys.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: ^3.5.2

# 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:
  flutter:
    sdk: flutter
  smooth_page_indicator: ^1.2.0+3
  cupertino_icons: ^1.0.8
  flutter_launcher_icons: ^0.14.1

dev_dependencies:
  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: ^5.0.0

# 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:
    - assets/images/car.png
    - assets/images/google_icon.png
    - assets/images/vk_icon.png
    - assets/images/backIcon.png
    - assets/images/home.png
    - assets/images/carRent.png
    - assets/images/chat.png
    - assets/images/profile.png
    - assets/images/bmwm5.jpg
    - assets/images/ferrari.webp
    - assets/images/porsche911.webp
    - assets/images/arrow-down.png
    - assets/images/sms.png
    - assets/images/search.png
    - assets/images/star.png
    - assets/images/userPhoto.jpg
    - assets/images/edit.png
    - assets/images/lock-circle.png
    - assets/images/direct-inbox.png
    - assets/images/empty-wallet.png

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

```
# https://flutter.dev/to/resolution-aware-images
# For details regarding adding assets from package dependencies, see
# https://flutter.dev/to/asset-from-package
# 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: Urbanist
    fonts:
      - asset: assets/fonts/Urbanist-VariableFont_wght.ttf
# For details regarding fonts from package dependencies,
# see https://flutter.dev/to/font-from-package
flutter_icons:
  android: true
  ios: true
  image_path: "assets/images/applcon.png" # Укажите путь к вашей иконке
flutter_intl:
  enabled: true
```