|  |  |
| --- | --- |
|  | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ Информатики и систем управления

КАФЕДРА Теоретической информатики и компьютерных технологий

**Летучка «31.10.24»**

по курсу «Разработка мобильных приложений».

«Разработка клиент-серверного приложения для работы с MySQL»

Выполнил:

студент группы ИУ9-71Б

Митрошкин Алексей

Проверил:

Посевин Д.П.

Москва, 2024

1. **Цель**

Разработать клиентского приложение для записи и чтения данных из СУБД MySQL.

1. **Задание**

* Разработка клиента на flutter.

1. **Реализация**

Исходный код сервера приведён в Листинге 1.

Листинг 1

|  |
| --- |
| import 'package:flutter/material.dart'; import 'package:mysql1/mysql1.dart';  class MySqlScreen extends StatefulWidget {  const MySqlScreen({Key? key}) : super(key: key);   @override  \_UserScreenState createState() => \_UserScreenState(); }  class \_UserScreenState extends State<MySqlScreen> {  final \_nameController = TextEditingController();  final \_phoneController = TextEditingController();  final \_emailController = TextEditingController();  final \_testController = TextEditingController();  List<Map<String, dynamic>> users = [];   Future<void> saveUser() async {  final conn = await MySqlConnection.*connect*(ConnectionSettings(  host: 'students.yss.su',  port: 3306,  user: 'iu9mobile',  db: 'iu9mobile',  password: 'bmstubmstu123'));   try {  await conn.query(  'INSERT INTO Mitroshkin (name, phone, email, test) VALUES (?, ?, ?, ?)',  [\_nameController.text, \_phoneController.text, \_emailController.text, \_testController.text],  );  print('User saved');  } catch (e) {  print('Error: $e');  } finally {  await conn.close();  }  }   Future<void> fetchUsers() async {  final conn = await MySqlConnection.*connect*(ConnectionSettings(  host: 'students.yss.su',  port: 3306,  user: 'iu9mobile',  db: 'iu9mobile',  password: 'bmstubmstu123'));   try {  var results = await conn.query('SELECT name, phone, email, test FROM Mitroshkin');  users = results  .map((row) => {'name': row[0], 'phone': row[1], 'email': row[2], 'test': row[3]})  .toList();  setState(() {});  } catch (e) {  print('Error: $e');  } finally {  await conn.close();  }  }   @override  Widget build(BuildContext context) {  return Scaffold(  appBar: AppBar(  title: const Text('Добавление пользователя'),  ),  body: Padding(  padding: const EdgeInsets.all(16.0),  child: Column(  children: <Widget>[  TextFormField(  controller: \_nameController,  decoration: const InputDecoration(labelText: 'Имя'),  ),  TextFormField(  controller: \_phoneController,  decoration: const InputDecoration(labelText: 'Номер телефона'),  keyboardType: TextInputType.*phone*,  ),  TextFormField(  controller: \_emailController,  decoration: const InputDecoration(labelText: 'Почта'),  keyboardType: TextInputType.*emailAddress*,  ),  TextFormField(  controller: \_testController,  decoration: const InputDecoration(labelText: 'Test'),  keyboardType: TextInputType.*text*,  ),  const SizedBox(height: 20),  ElevatedButton(  onPressed: saveUser,  child: const Text('Сохранить'),  ),  const SizedBox(height: 20),  ElevatedButton(  onPressed: fetchUsers,  child: const Text('Получить все записи'),  ),  const SizedBox(height: 20),  users.isNotEmpty  ? Expanded(  child: SingleChildScrollView(  scrollDirection: Axis.horizontal,  child: DataTable(  columns: const [  DataColumn(label: Text('Имя')),  DataColumn(label: Text('Телефон')),  DataColumn(label: Text('Почта')),  DataColumn(label: Text('Test'))  ],  rows: users  .map((user) => DataRow(cells: [  DataCell(Text(user['name'])),  DataCell(Text(user['phone'])),  DataCell(Text(user['email'])),  DataCell(Text(user['test']))  ]))  .toList(),  ),  ),  )  : const Text('Нет данных для отображения'),  ],  ),  ),  );  } } |

В листиге 2 приведён исходный код клиентского приложения.

Листинг 2

На рисунках 1-2 приведёны результаты работы программы

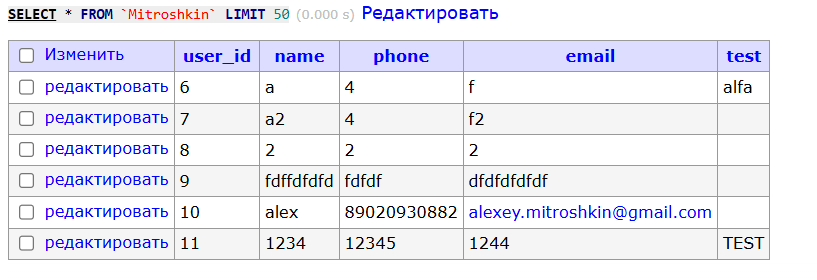


Рисунок 1

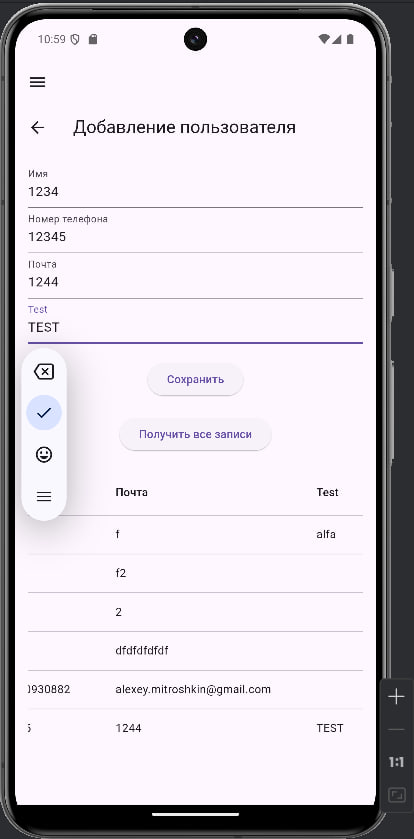


Рисунок 2