

Отчёт по проделанной работе на семинаре по мобильной разработке (06.05.2022).

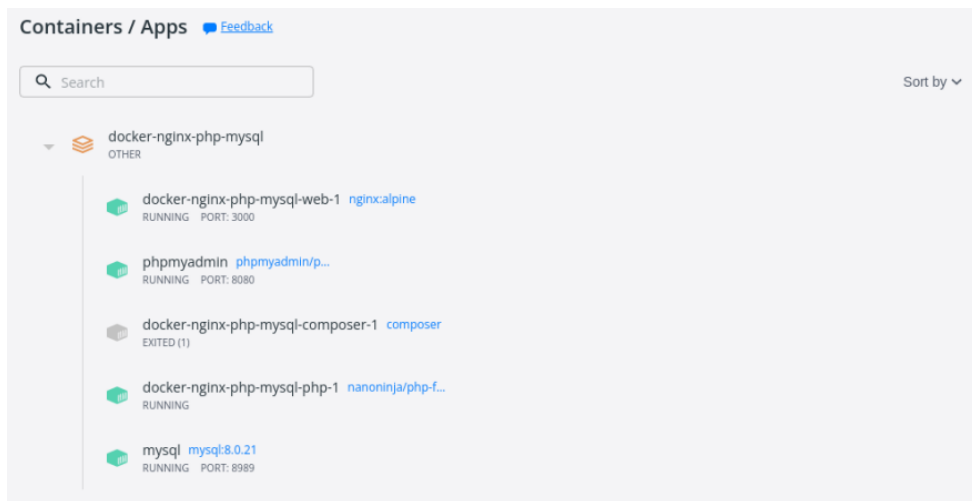
(выполнил Валяев Георгий Анатольевич)

React-native – один из интересных и богатых фреймворков, наследующих большинство свойств фреймворка React (в рамках разработки нативных веб-приложений), разработанный командой Facebook для создания и написания приложений под мобильные устройства с поддержкой ОС Android, iOS и других систем.

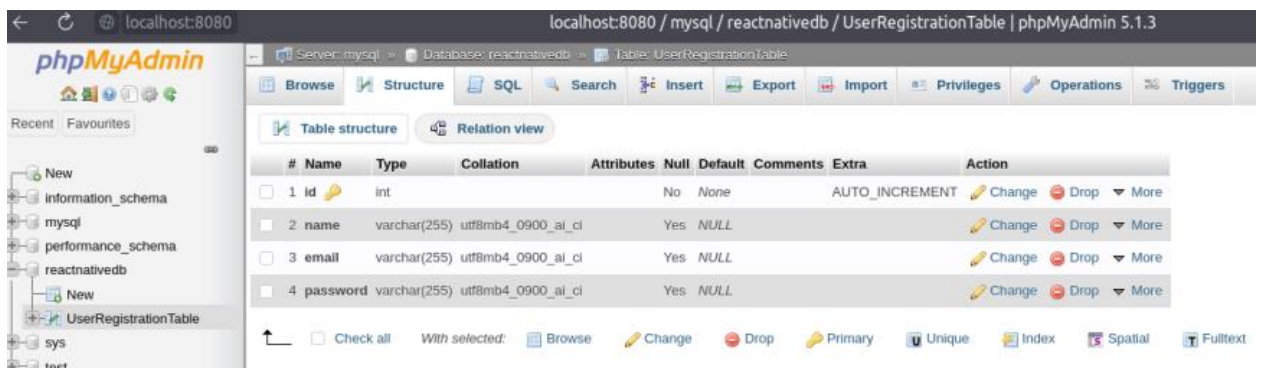
В этой работе мы создадим свою простую форму регистрации с локальной базой данных на сервере с применением контейнерной технологии Docker и рекомендации по работающему функционалу на языке PHP.

Полноценный код всего проекта расположен в моём репозитории GitHub:
https://github.com/Valyaevgeorgiy/Android_projects/tree/main/React-Native/Forms_reg

Чтобы механизм прохождения регистрации функционировал полностью, была задействована контейнерная технология (приложение) Docker с использованием docker-compose, которая состоит из основных 4 контейнеров: mysql, php, nginx, phpmyadmin:

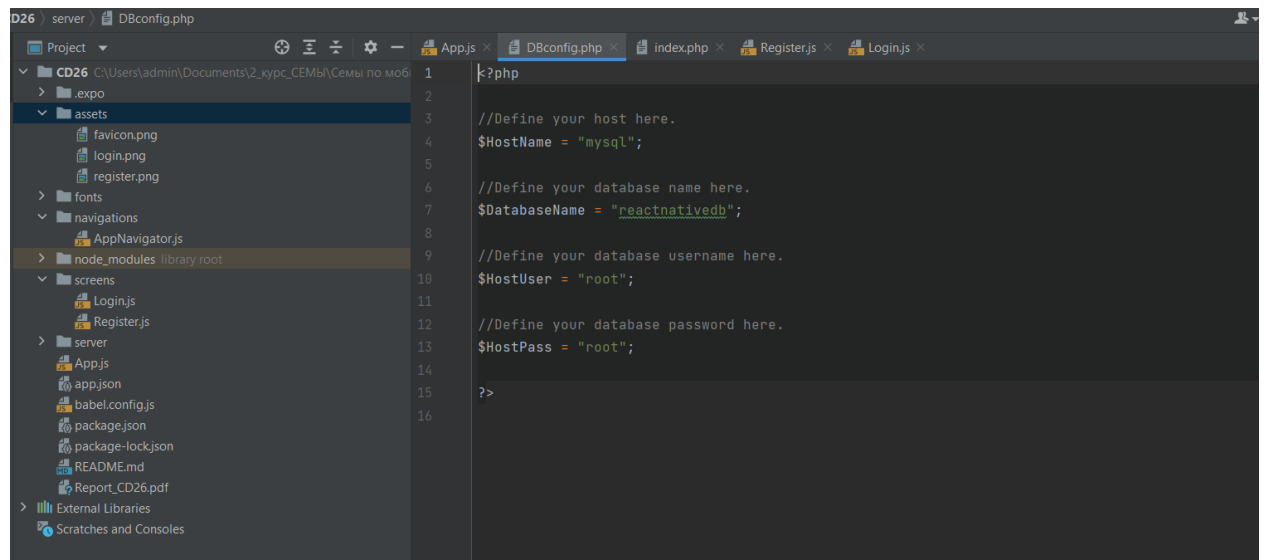


На основе пробного пользования администрирования баз данных была создана простейшая локальная база данных (reactnatedb) введенных имён, паролей и электронных почт пользователей в единой таблице UserRegistrationTable:



По изученным рекомендациям в контейнере nginx через среду разработки WebStorm изменяем исходный файл index.php и дописываем дополнительный к нему файл DBconfig.php с основными конфигурационными именами сервера, базы данных и деталей хоста (HostName, HostUser, HostPass).

Полноценно эти 2 файла выглядят следующим образом:



Процесс входа в сам контейнер через консоль работает следующим способом:

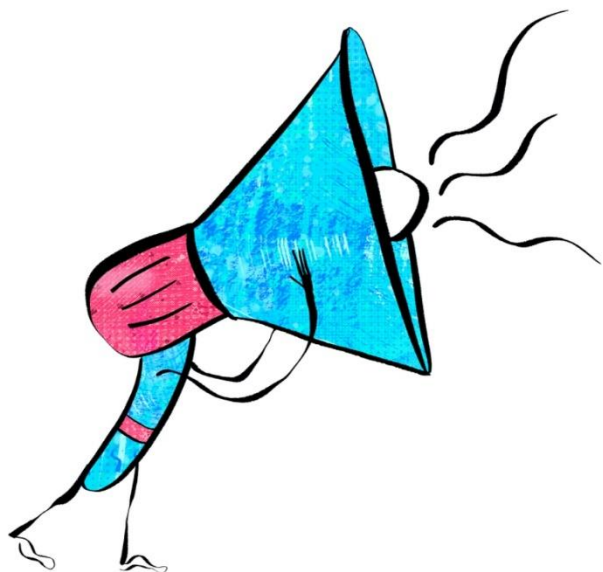
`docker exec -it docker-nginx-php-mysql-web-1`

```
App.js × DBconfig.php × index.php × Register.js × Login.js ×
1 <?php
2
3 // Importing DBConfig.php file.
4 include 'DBConfig.php';
5
6 // Creating connection.
7 $con = mysqli_connect($HostName,$HostUser,$HostPass,$DatabaseName);
8
9 // Getting the received JSON into $json variable.
10 $json = file_get_contents('php://input');
11
12 // decoding the received JSON and store into $obj variable.
13 $obj = json_decode($json,true);
14
15 // Populate User name from JSON $obj array and store into $name.
16 $name = $obj['name'];
17
18 // Populate User email from JSON $obj array and store into $email.
19 $email = $obj['email'];
20
21 // Populate Password from JSON $obj array and store into $password.
22 $password = $obj['password'];
23
24 //Checking Email is already exist or not using SQL query.
25 $CheckSQL = "SELECT * FROM UserRegistrationTable WHERE email='$email'";
26
27 // Executing SQL Query.
28 $check = mysqli_fetch_array(mysqli_query($con,$CheckSQL));
29
30
31 if(isset($check)){
32
```

```
App.js × DBconfig.php × index.php × Register.js × Login.js ×
28 $check = mysqli_fetch_array(mysqli_query($con,$CheckSQL));
29
30
31 if(isset($check)){
32
33     $EmailExistMSG = 'Email Already Exist, Please Try Again !!!';
34
35     // Converting the message into JSON format.
36     $EmailExistJson = json_encode($EmailExistMSG);
37
38     // Echo the message.
39     echo $EmailExistJson ;
40
41 }
42 else{
```

```
Appjs × DBconfig.php × index.php × Register.js × Login.js ×
41 }
42 else{
43
44 // Creating SQL query and insert the record into MySQL database table.
45 $Sql_Query = "insert into UserRegistrationTable (name,email,password) values ('$name','$email','$password')";
46
47
48 if(mysqli_query($con,$Sql_Query)){
49
50 // If the record inserted successfully then show the message.
51 $MSG = 'User Registered Successfully' ;
52
53 // Converting the message into JSON format.
54 $json = json_encode($MSG);
55
56 // Echo the message.
57 echo $json ;
58
59 }
60 else{
61
62 echo 'Try Again';
63
64 }
65 }
66 mysqli_close($con);|
67 ?>
```

Получившийся результат работы приложения выглядит следующим образом:



Вход

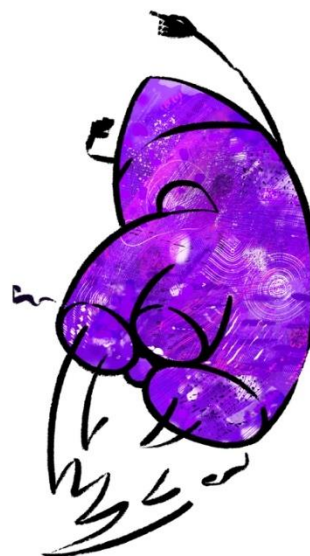
Введите логин и пароль, который создавали ранее при регистрации.

→ Login

🔒 Пароль

Войти

Нет аккаунта? Зарегистрироваться



Регистрация

Введите свои персональные данные для повышения персонализации. Это необходимо только один раз.

👤 Egor

✉ valjr02@mail.ru

🔒

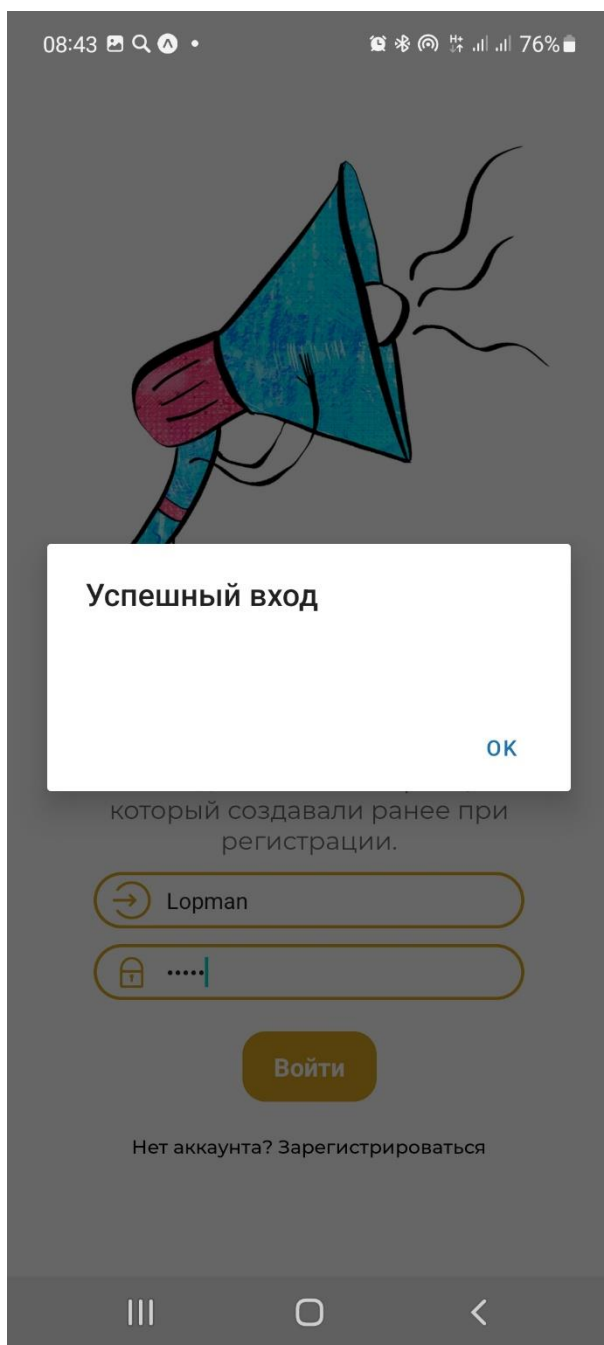
Зарегистрироваться

Уже есть регистрация? Войти

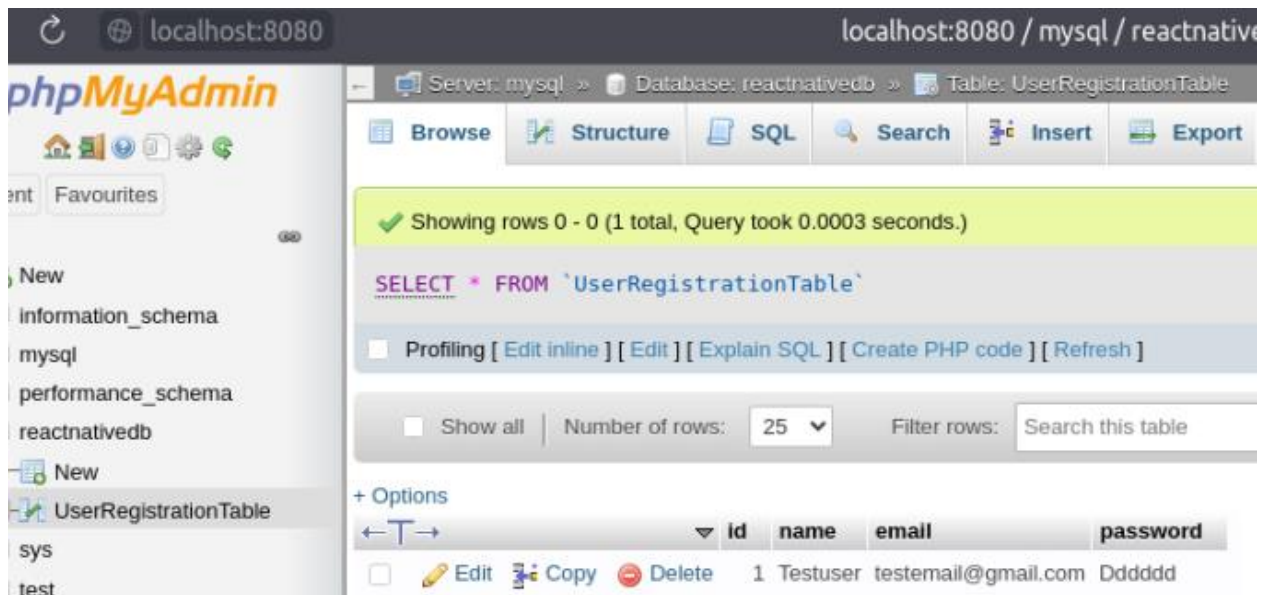


Таким образом реализована подгрузка шрифтов, иконок приложения, картинок и стилизации в плане выравнивания по экрану и цветов.

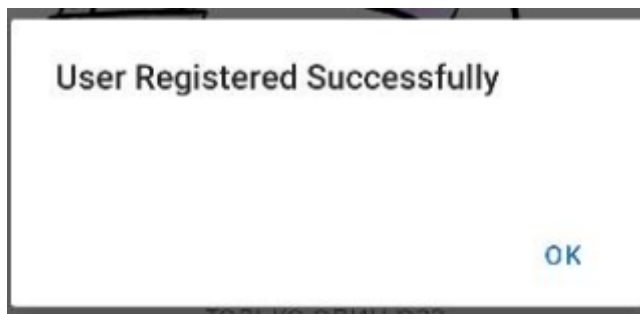
При уже имеющихся данных в базе реализовано окно с пояснением успешного входа в систему:



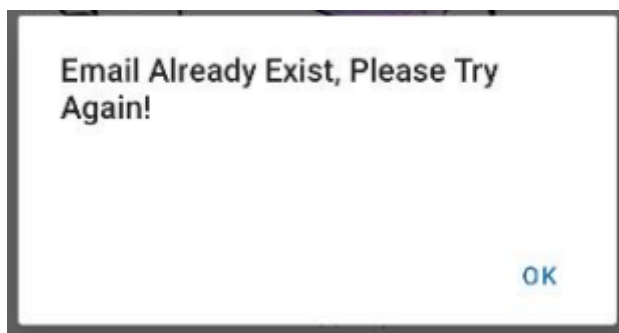
В базу данных попадают данные с формы и добавляется запись в БД с пользователем Testuser. При вводе уникального значения в поле email и нажатии на кнопку «Зарегистрироваться» приложение посылает POST запрос на сервер и выводит в Alert ответ от сервера:



Уже при успешном вводе на странице авторизации зарегистрированных данных пользователем выводится Alert сообщение следующего формата:



В случае, когда пользователь на странице регистрации вводит уже существующие в таблице базы данных значения (например, Email), то сервер возвращает Email Already Exist.



Контрольные вопросы

1. Какие основные свойства для выравнивания объектов на экране существуют?
1. Основными свойствами в стилизации выравнивания объектов являются alignContent, alignItems, alignSelf, justifyContent.