

УДК 004.9

## **РАЗРАБОТКА МОБИЛЬНОГО ПРИЛОЖЕНИЯ НА ОСНОВЕ АРХИТЕКТУРЫ REST**

Драчев Н.П., студент гр. ПИБ-192, III курс  
Фанасков Н.В., студент гр. ПИБ-191, III курс  
Чашкин Д.А., студент гр. ПИБ-191, III курс  
Ковшов Д.В., студент гр. ПИБ-191, III курс  
Нечитайло Д.В., студент гр. ПИБ-191, III курс

Научный руководитель: Тайлакова А.А.  
Кузбасский государственный технический университет  
имени Т.Ф. Горбачева  
г. Кемерово

В современном мире интернет является неотъемлемой частью нашей жизни. С появлением интернета клиент-серверный подход набирает обороты. В нем есть ряд преимуществ, таких как:

1. Мощный сервер – нужен для того, чтобы уменьшить нагрузку клиента, т.е. вся обработка информации происходит на сервере, а клиент получает результат.

2. Нет дублирования кода – основной код хранится на сервере, клиент отвечает только за «нарисовать красиво» и простые проверки на полях «тут число, тут строка не длиннее 200 символов».

3. Персональные данные в безопасности – простой пользователь не видит лишнего. Информацию из базы данных он получает обработанную.

Этапы разработки проекта:

1. Разработка идеи приложения и интеграции со сторонним API;
2. Разработка сервера архитектуры REST API [1];
3. Разработка web-приложения.

Первым этапом было найти стороннее API к которому обращался бы наш сервер, в свою очередь обрабатывал информацию и записывал ее в базу данных. При этом клиент обращался бы к API нашего сервера. В качестве стороннего API был выбран «Weather API» [4], который позволяет по введённому городу узнать погоду. Вторым этапом стала разработка сервера. Была использована среда разработки PyCharm, язык программирования Python с библиотеками Django [3], REST Framework. Разработаны endpoint:

- авторизация – по логину, паролю и выбранному языку;
- регистрация – по логину, паролю и выбранному языку;
- список погоды
- история – получения списка истории фактов авторизованного пользователя;
- удаление записей истории авторизованного пользователя;

- получение погоды по выбранному городу, почтовому индексу, IP-адресу;

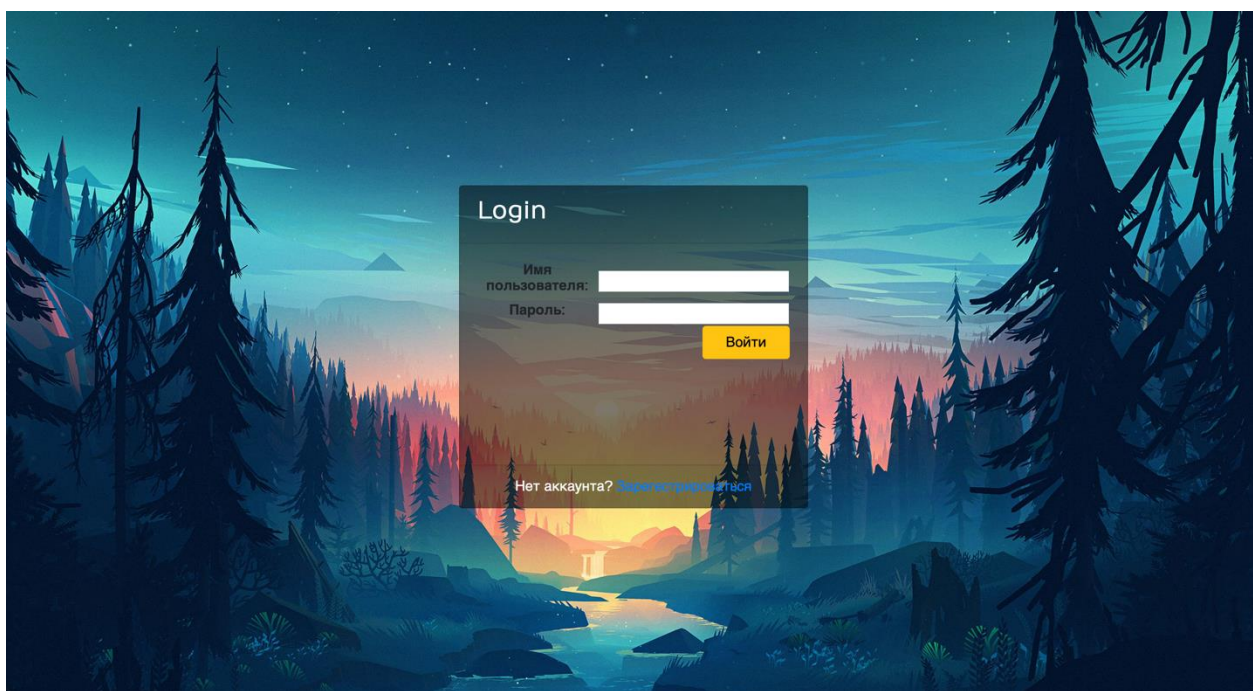
Для третьего этапа разработки web-приложения использована среда разработки rwa [Ошибка! Источник ссылки не найден.]. В ней используется объектно-ориентированный язык программирования C# [2], который в свою очередь имеет низкий порог вхождения и высокую производительность. Так была прописана логика мобильного приложения, позволяющая увидеть конечный результат проекта.

Целью работы является создание мобильного приложения с показом фактов о цифрах с использованием архитектуры REST, которое могло бы конкурировать с себе подобными.

Для достижения цели поставлены следующие задачи:

- 1) Придумать идею приложения
- 2) Разработать сервис, отдающий свое API и обрабатывающий стороннее API
- 3) Разработать web-приложение

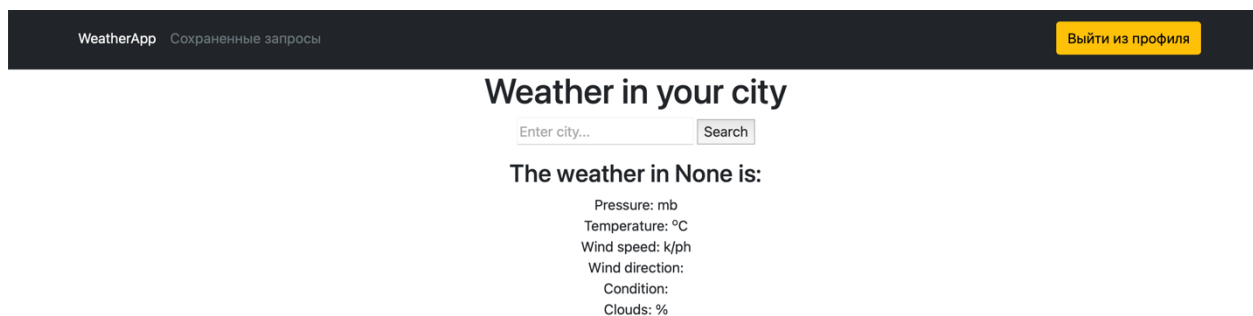
Меню входа/регистрации представляет собой интуитивно понятный пользовательский интерфейс, на котором располагаются такие кнопки управления, как вход/регистрация, выбор языка (рис. 1).



*Рисунок 1 – Меню входа/регистрации*

После входа пользователь видит основное меню (рис. 2).

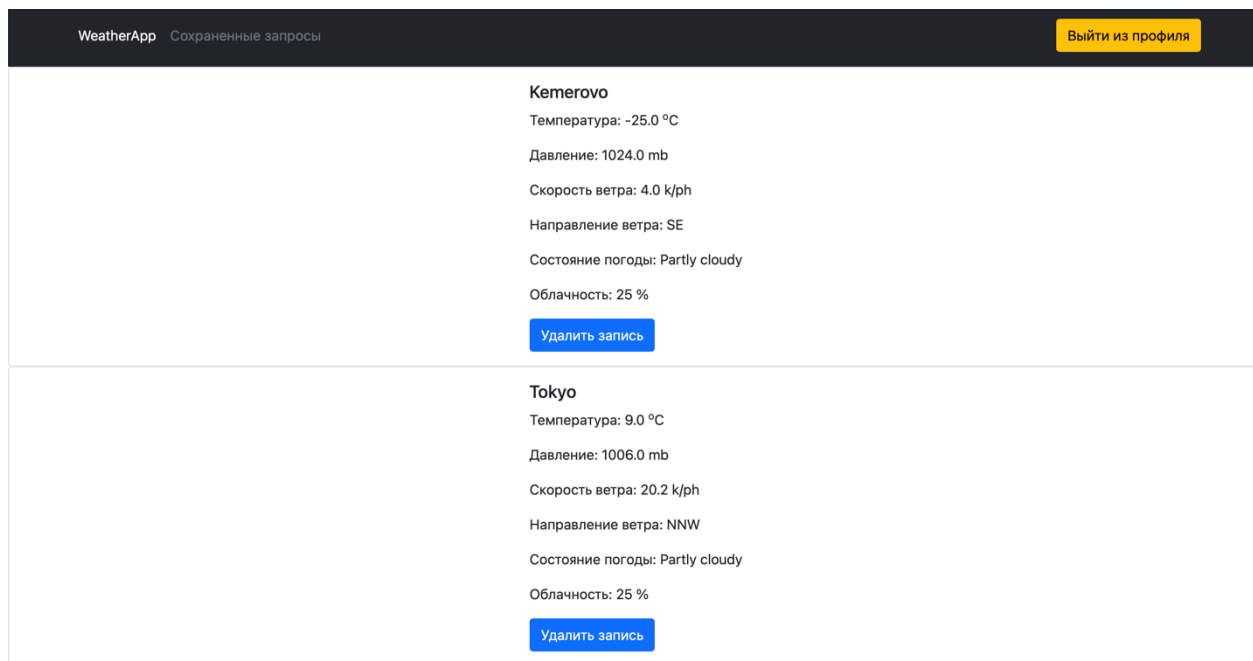
На основной странице представлена поле для ввода города. Для генерации погоды используется введенный город. При нажатии на кнопку «search» программа выдаст вам погоду в настоящий момент времени в выбранном месте.



*Рисунок 2 – основное меню*

При нажатии на кнопку в правом верхнем углу экрана «Выйти из профиля», мы попадем обратно в меню входа/регистрации (рис. 2).

При нажатии на кнопку в левом верхнем углу экрана «Сохраненные запросы» мы перейдем в раздел истории запросов (рис. 3).



*Рисунок 3 – история запросов*

В меню истории запросов представлен архив с погодой на заданные пользователем значения с указанием города, почтового индекса, IP адреса.

В результате выполнения работы разработано web-приложение, использующее архитектуру REST, целью работы которого является показ погоды в выбранном городе.

### Список литературы:

1. Руководство REST API [Электронный ресурс]. – Режим доступа: <https://mcs.mail.ru/blog/vvedenie-v-rest-api>, свободный (дата обращения 29.12.2021).
2. Руководство C# [Электронный ресурс]. – Режим доступа: <https://docs.microsoft.com/ru-ru/dotnet/csharp/programming-guide/>, свободный (дата обращения 29.12.2021).
3. Документация Django [Электронный ресурс]. - Режим доступа: <https://docs.djangoproject.com/en/4.0/>, свободный (дата обращения 29.12.2021).
4. Стороннее API - [RapidApi.com](https://rapidapi.com/), свободный (дата обращения 29.12.2021).