

УДК 004.42

Студ. Д. А. Татарович
Науч. рук. ст. преп. А. С. Наркевич
(кафедра программной инженерии, БГТУ)

РАЗРАБОТКА ВЕБ-ПРИЛОЖЕНИЯ «ТАКСИ АГРЕГАТОР»

Цель работы: разработать веб-приложение, предоставляющее широкий спектр функциональных возможностей для водителей такси, клиентов службы такси и администраторов веб-приложения.

Веб-приложение «Такси агрегатор» должно выполнять следующие задачи:

- создание аккаунтов с различными ролями: водителя, клиента и администратора;
- учёт данных о водителях, клиентах и автомобилях;
- предоставление пользователям функционала по созданию и обработке заказов такси;
- защита данных пользователей на всех этапах обработки и хранения;
- учёт поездок и хранение подробных данных для отчетов;
- осуществление поиска по учтённым данным;
- хранение всей информации в базе данных;
- создание и получение статистических отчётов;
- предоставление административного функционала для аккаунтов с ролью администратора.

Данное веб-приложение предназначено для использования как клиентами службы такси, так и водителями такси.

Несмотря на то, что на сегодняшний день в Беларуси услуги заказа такси предоставляет множество приложений, новые приложения продолжают появляться на рынке, а значит разработка такого веб-приложения остаётся актуальной и востребованной.

Веб-приложение должно предоставлять максимально удобный и простой в освоении функционал для каждой роли пользователей, осуществлять безопасную обработку, транспортировку и хранение всех пользовательских данных и выполняться одинаково вне зависимости от устройства и операционной системы.

На сегодняшний день в сфере разработки веб-приложений существует множество технологий, предоставляющих возможность реализовать приложение, полностью соответствующее сформулированным требованиям. Поэтому, на стадии проектирования веб-приложения был проведён сравнительный анализ подходящих техно-

логий. Таким образом был подобран соответствующий целям и задачам проекта набор технологий.

Приложение реализует клиент-серверную модель с внешней базой данных на основе архитектурного шаблона MVC.

В качестве платформы для разработки была выбрана платформа .NET Core, соответственно, был использован веб-фреймворк ASP.NET Core.

.NET Core — это универсальная платформа разработки с открытым кодом, которую поддерживает корпорация Майкрософт и сообщество .NET на сайте GitHub. Она является кроссплатформенной (поддерживает Windows, macOS и Linux) и может использоваться для создания приложений для различных устройств, облачных приложений и Интернета вещей.

ASP.NET Core является кроссплатформенной, высокопроизводительной средой с открытым исходным кодом для создания современных веб-приложений, и предоставляет множество инструментов для разработчика.

Разработку веб-приложения на основе архитектурного шаблона MVC существенно упрощает такой компонент фреймворка ASP.NET Core, как ASP.NET Core MVC.

ASP.NET Core MVC — это упрощенная, эффективно тестируемая платформа с открытым исходным кодом, оптимизированная для использования с ASP.NET Core. К возможностям ASP.NET MVC относятся: маршрутизация, привязка модели, проверка модели, внедрение зависимостей, фильтрация, тестируемость, работа с представлениями.

Для отрисовки представлений ASP.NET Core MVC используют подсистему просмотра Razor. Razor — это компактный, выразительный и гибкий язык разметки шаблонов для определения представлений с помощью встроенного кода C#. Razor используется для динамического создания веб-содержимого на сервере. Код на стороне сервера можно полностью комбинировать с содержимым и кодом на стороне клиента.

Для облегчения работы с данными был использован Entity Framework Core. Entity Framework Core — это простая, кроссплатформенная и расширяемая версия популярной технологии доступа к данным Entity Framework с открытым исходным кодом.

EF Core может использоваться как объектно реляционный модуль сопоставления (ORM), позволяя разработчикам работать с базой данных с помощью объектов .NET и устраняя необходимость в напи-

сании большей части кода, требуемого для доступа к данным. EF Core поддерживает множество СУБД, в том числе postgresql 11.

Структура приложения реализована в соответствии с паттерном MVC: клиент обращается к контроллеру, контроллер возвращает представление, представление взаимодействует с контроллером и моделью, модель работает с базой данных, обновляя данные, и возвращая необходимые данные контроллеру. Таким образом, мы получаем чёткую структуру проекта, обеспечивающую удобную разработку и дальнейшую поддержку приложения.

В качестве СУБД для проекта была выбрана PostgreSQL 11. Эта СУБД предоставляет всё, что необходимо для удобного хранения и работы с данными веб-приложения.

В соответствии с планируемым функционалом была спроектирована структура базы данных, состоящая из девяти таблиц. Структура базы данных соответствует третьей нормальной форме, таким образом, достигается целостность данных, независимость данных и минимальная избыточность.

Составленный список технологий позволяет реализовать весь требуемый функционал, и является удобным как для пользователей веб-приложения, так и для разработчика.

ЛИТЕРАТУРА

1. Документация по .NET [Электронный ресурс] – Режим доступа: <https://docs.microsoft.com/ru-ru/dotnet/index> – Дата доступа 15.04.2019.
2. Проектирование реляционных баз данных с использованием нормализации [Электронный ресурс] / С.Д. Кузнецов. – Режим доступа: http://citforum.ru/database/osbd/glava_23.shtml – Дата доступа: 15.04.2019.
3. Документация к PostgreSQL 11.2 [Электронный ресурс] – Режим доступа: <https://postgrespro.ru/docs/postgresql/11> – Дата доступа: 15.04.2019.