

Таким образом, созданный робот упрощает работу с БД и имеет ряд преимуществ: робот прост в использовании; с его помощью была упрощена работа с БД; позволяет заполнять БД, созданную на основе MicrosoftSQLServer, даже тем пользователям, у которых нет опыта работы с данной СУБД; при необходимости робот может быть легко усовершенствован, так как он разработан на основе платформы с визуальным конструктором; самостоятельно выполняет расчет таких полей, как надбавка за стаж, налоги и итоговая зарплата.

Процесс создания робота был довольно простым и не занял много времени. Поэтому можно полагать, что автоматизация работы с базами данных является перспективным направлением, а, значит, в ближайшем будущем она будет применяться во многих сферах.

УДК 004

Студ. Д.А Куликович  
Науч. рук. В.С. Кантарович  
(кафедра информационных систем и технологий, БГТУ)

## **ВЕБ-ПРИЛОЖЕНИЕ ДЛЯ СОЗДАНИЯ ОПРОСОВ**

Сейчас, в век информационных технологий, онлайн исследования занимают далеко не последнее место среди других методов сбора эмпирической информации. Зачастую респонденты отказываются от опросов ввиду нехватки времени и веб-приложение поможет сэкономить время опрашиваемого, а также позволит более эффективно обрабатывать полученные данные. Основой любого приложения в том числе и веб, является его архитектура.

Существует несколько основных типов архитектур.

– Многоуровневая архитектура (N-tierarchitecture) – это одна из самых распространенных архитектур. На её основе построено множество крупных фреймворков. Система делится на уровни, каждый из которых взаимодействует лишь с двумя соседними. Поэтому запросы к БД, которая обычно располагается в самом конце цепочки взаимодействия, проходят последовательно сквозь каждый «слой».

– Микроядерная архитектура – этот тип архитектуры состоит из двух компонентов: ядра системы и плагинов. Плагины отвечают за бизнес-логику, а ядро руководит их загрузкой и выгрузкой. Как пример микроядерной архитектуры в книге O'Reilly приводится Eclipse IDE. Это простой редактор, который открывает файлы, дает их править и запускает фоновые процессы. Но с добавлением плагинов (например, компилятора Java) его функциональность расширяется. Микроядерную архитектуру в свое время использовала операционная система Symbian

для мобильных устройств (разработку прекратили в 2012 году). В её микроядре находился планировщик задач, системы управления памятью и драйверы, а файловая система и компоненты, отвечающие за телефонную связь, выступали в роли плагинов.

– Микросервисы – похожи на архитектуру, управляемую событиями, и микроядро. Но используются тогда, когда отдельные задачи приложения можно легко разделить на небольшие функции — независимые сервисы. Эти сервисы могут быть написаны на разных языках программирования, поскольку общаются друг с другом при помощи REST API (например, с использованием JSON или Thrift). В каких пропорциях делить код, решает разработчик, но Сэм Ньюмен (SamNewman), автор книги «Создание микросервисов», рекомендует выделять на микросервис столько строк кода, сколько команда сможет воспроизвести за две недели. По его словам, это позволит избежать излишнего «раздувания» архитектуры. Чаще всего микросервисы запускаются в так называемых контейнерах. Эти контейнеры доступны по сети другим микросервисами и приложениям, а управляет ими всеми система оркестровки: примерами могут быть Kubernetes, DockerSwarm и др.

Для моего проекта была выбрана многоуровневая архитектура, а именно трёхуровневая. Мной были выделены уровни доступа к данным (DataAccessLayer), уровень бизнес-логики (BusinessLogicLayer) и уровень с моим веб-приложением. Из технологий использовались для бэкенда:

– ASP.NETCore – свободно-распространяемый кроссплатформенный фреймворк для создания веб-приложений на платформе .NET с открытым исходным кодом. Данная платформа разрабатывается компанией Майкрософт совместно с сообществом и имеет большую производительность по сравнению с ASP.NET.

– IdentityCore – библиотека которая позволяет пользователям создавать учетные записи, аутентифицироваться, управлять учетными записями или использовать для входа на сайт учетные записи внешних провайдеров, таких как Facebook, Google, Microsoft, Twitter и других.

– EntityFrameworkCore – EntityFramework (EF) Core – это простая, кроссплатформенная и расширяемая версия популярной технологии доступа к данным EntityFramework с открытым исходным кодом.

– EFCore может использоваться в качестве объектно-реляционного модуля сопоставления (O/RM), который позволяет разработчикам .NET работать с базой данных с помощью объектов .NET; устраняет необходимость в большей части кода для доступа к данным, который обычно приходится писать. EFCore поддерживает множество систем баз данных.

Для фронтенда:

– Razorpages - предоставляют технологию, альтернативную системе Model-View-Controller. RazorPages позволяют создавать страницы с кодом Razor, которые могут обрабатывать запросы. В некоторой степени эта функциональности напоминает работу веб-форм, которые представляли страницу с расширением aspx и имели файл логики на C#, связанный с данной страницей. В этом плане RazorPages представляют альтернативу стандартной модели MVC для построения приложения.

– Bootstrap – свободный набор инструментов для создания сайтов и веб-приложений. Включает в себя HTML- и CSS-шаблоны оформления для типографики, веб-форм, кнопок, меток, блоков навигации и прочих компонентов веб-интерфейса, включая JavaScript-расширения.

– JQuery – набор функций JavaScript, фокусирующийся на взаимодействии JavaScript и HTML. Библиотека jQuery помогает легко получить доступ к любому элементу DOM, обращаться к атрибутам и содержимому элементов DOM, манипулировать ими. Также библиотека jQuery предоставляет удобный API для работы с AJAX.

В результате было получено веб-приложение с аутентификацией и авторизацией для проведения опросов, а также с возможностью их создавать и редактировать. В нем присутствует система ролей, которая отделяет админов от обычных юзеров.

## ЛИТЕРАТУРА

1. Стиль многоуровневой архитектуры [Электронный ресурс] / Сайт docs.microsoft.com. – 2022. – Режим доступа: <https://docs.microsoft.com/en-us/azure/architecture/guide/architecture-styles/n-tier>. – Дата доступа: 10.03.2022.

2. Стиль микросервисной архитектуры [Электронный ресурс] / Сайт docs.microsoft.com. – 2022. – Режим доступа: <https://docs.microsoft.com/en-us/azure/architecture/guide/architecture-styles/microservices>. – Дата доступа: 10.03.2022.

3. Кратко о типах программного обеспечения, и какую из них выбрали мы для IaaS-провайдера [Электронный ресурс] / Сайт habr.com. – 2022. – Режим доступа: <https://habr.com/ru/company/1cloud/blog/424911/>. – Дата доступа: 10.03.2022.

4. Аутентификация и авторизация для любого приложения [Электронный ресурс] / Сайт docs.microsoft.com. – 2022. – Режим доступа: <https://docs.microsoft.com/en-us/azure/active-directory/develop/>. – Дата доступа: 10.03.2022.