В этой статье мы рассмотрим процесс разработки веб-приложения на микросервисной архитектуре для конгломератного холдинга, используя ряд инструментов и технологий, таких как PostgreSQL, EntityFrameworkCore, Mediatr, FluentValidation, и другие.

Стек технологий:

База данных (БД) PostgreSQL – открытая реляционная СУБД, которая обеспечивает высокую надежность, масштабируемость и производительность.

EntityFrameworkCore – популярный ORM для .NET, который позволяет разработчикам работать с базами данных с помощью объектов .NET.

Unit of Work – шаблон проектирования, который группирует операции, выполняемые с базой данных, в единое целое.

Mediatr – библиотека для упрощения взаимодействия между компонентами системы.

FluentValidation – библиотека для валидации объектов и запросов.

PredicatesBuilder – библиотека для построения предикатов для запросов к БД.

OperationResult – шаблон для возврата результатов операций, соответствующий стандарту RFC7807.

Microsoft Identity – фреймворк для аутентификации и авторизации пользователей.

Vertical Slice Architecture – архитектурный стиль, который структурирует приложение по функциональным возможностям.

Minimal API – новый стиль API в ASP.NET Core, который позволяет создавать более лаконичные и эффективные API.

OpenIddict Auth2.0 – библиотека для реализации OpenID Connect и OAuth 2.0.

Nimble Framework (Microservice Template) – шаблон для создания микросервисов.

Swagger – инструмент для создания документации к API.

AppDefinitions – шаблон для приложений в микросервисной архитектуре.

Domain Events – шаблон проектирования, который используется для обмена сообщениями между различными частями системы.

Domain-Driven Design (DDD) – методология проектирования, которая центируется на домене и использует его язык.

Приложение будет построено на основе микросервисной архитектуры, где каждый сервис будет отвечать за свою часть функциональности. Это позволит масштабировать отдельные части системы независимо друг от друга, улучшит устойчивость и обеспечит быстрое внедрение новых функций.

RabbitMQ используется для обмена сообщениями между сервисами, а Ocelot – для маршрутизации запросов к соответствующим сервисам. Это позволяет создать гибкую и масштабируемую систему, где каждый сервис может быть развернут и масштабирован независимо друг от друга.

Разработанное приложение на микросервисной архитектуре позволит эффективно управлять и отслеживать движения персонала, автоматизировать заполнение документов и предоставлять подробную информацию о работе в холдинге. Стек технологий, используемый в приложении, обеспечивает высокую производительность, масштабируемость и гибкость системы.

С. В. Васильев
(ГГУ имени Ф. Скорины, Гомель)
Науч. рук. **В. Н. Кулинченко**, ст. преподаватель

ПРИМЕНЕНИЕ MICROSOFT ENTITY FRAMEWORK CORE В РАМКАХ РАЗРАБОТКИ WEB API НА БАЗЕ ФРЕЙМВОРКА ASP.NET CORE

Entity Framework Core – это фреймворк, который разработчики программного обеспечения могут использовать для доступа к базам данных (БД). EF Core представляет

собой популярный инструмент для разработки на базе ASP.NET Core, основными преимуществами которого является кроссплатформенность и возможность быстрого написания кода для работы с БД.

EF Core разработан как инструмент объектно-реляционного отображения (ORM). ORM – это технология, которая позволяет работать с базами данных, используя объектно-ориентированный подход.

В основе EF Core лежит ряд паттернов. Один из таких паттернов – это Unit of Work. Он позволяет группировать операции к БД в рамках одной транзакции, обеспечивая целостность данных. Еще одним паттерном, который используется в EF Core, является Repository. Он позволяет абстрагировать доступ к данным от способа их хранения. Repository предоставляет удобный интерфейс для работы с данными, скрывая детали реализации.

При использовании EF Core для разработки базы данных представлены различные подходы. Один из них – Code First, при котором сущности и контекст базы данных создаются на основе классов, а EF Core автоматически генерирует или обновляет схему БД. Другой подход – Database First, когда классы сущностей и контекст базы данных генерируются на основе готовой схемы БД. Также существует подход Model First, при котором модель данных создается в среде визуального проектирования Microsoft Entity Framework Designer, а затем EF Core генерирует классы сущностей и контекст базы данных на основе этой модели.

Таким образом, использование Microsoft Entity Framework Core при разработке Web API на базе ASP.NET Core предоставляет эффективные инструменты для работы с БД. Помимо этого, EF Core обеспечивает гибкость разработки, благодаря наличию различных подходов для работы с БД, а также сокращает время, затраченное на разработку, и повышает производительность приложений.

Н. А. Герасенко
(ГГУ имени Ф. Скорины, Гомель)
Науч. рук. **В. В. Грищенко**, ст. преподаватель

РАЗРАБОТКА МЕТЕОСТАНЦИИ С ОЦЕНКОЙ ВЕРОЯТНОСТИ ИЗМЕНЕНИЯ ПОГОДЫ

В настоящее время метеостанция, созданная для использования в домашних условиях, предоставляет владельцу точную и персонализированную информацию о погоде, что помогает планировать деятельность, принимать решения и создавать комфортабельную среду внутри помещения. Она обеспечивает возможность мониторинга температуры, влажности, атмосферного давления и других параметров, а также предоставляет прогноз погоды на основе собственного анализа данных. Такая метеостанция позволяет эффективно управлять отоплением, кондиционированием воздуха и вентиляцией, а также принимать решения о планировании отдыха и занятиях на открытом воздухе.

Основная часть Микроконтроллер ATmega328 (Arduino nano) – позволяет быстро и легко создавать прототипы электронных устройств и систем. Он предоставляет доступ к различным входным и выходным портам, а также поддерживает подключение к различным датчикам, модулям и компонентам.

Второстепенная часть датчики – mh z19b (датчик углекислого газа), lcd i2c дисплей, rtc модуль (модуль реального времени), bme 280 (датчик температуры, атмосферного давления и влажности), ttp223 (сенсорная кнопка).

Что будет замерять наша метеостанция:

– концентрацию углекислого газа;

– влажность;