

УДК 004.624

РАЗРАБОТКА ВЕБ-АРИ И БАЗЫ ДАННЫХ ВЕБ-ПРИЛОЖЕНИЯ
ИНФОРМАЦИОННОЙ СИСТЕМЫ ДЛЯ ПОЧТОВЫХ ОТДЕЛЕНИЙ

П.В. ДЕРЕВЯГА

(Представлено: О.Н. ПЕТРОВИЧ)

В данной статье описывается практическая часть разработки серверной части и базы данных веб-приложения информационной системы для почтовых отделений. Приведен и обоснован выбор технологий для разработки, а также особенности системы.

Введение. Для упрощения жизни практически во всех сферах деятельности стало очень выгодно переходить на автоматизированные информационные системы.

Автоматизированная информационная система (АИС) – это комплекс, который включает компьютерное и коммуникационное оборудование, программное обеспечение, лингвистические средства, информационные ресурсы, предназначенных для сбора, подготовки, хранения, обработки и предоставления информации, а также системный персонал, обеспечивающий поддержку динамической информационной модели некоторой части реального мира для удовлетворения информационных потребностей пользователей и для принятия решений.

Основа любой информационной системы – база данных.

Базы данных уже давно стали основой не только любого солидного приложения, но и любой информационной системы. Они позволяют упорядочить хранение данных о предметной области в одной структуре, предоставляют широкие возможности для взаимодействия с этой информацией. Кроме того, грамотно спроектированная и разработанная база данных – залог успешного приложения. Поэтому сегодня практически в каждой сфере человеческой деятельности необходимы эти самые базы данных.

Не исключение и сфера почтовых услуг – информация о газетах и журналах, подписчиках и подписках, работниках и так далее – все это должно быть сохранено и упорядочено. База данных поможет упростить доступ к информации, организовать её поиск, а также значительно сократить затраты времени и сил на ведение документации и архивов. Кроме того, база данных для почтовых отделений при помощи её возможностей позволит частично автоматизировать некоторые процессы, например, составление квитанций, подсчет финансов, составление накладных и заказов на печать.

Для обычного пользователя такая информационная система позволит просматривать общую информацию об изданиях, отделениях и т.д., для заведующего почтовым отделением – это возможности по учету и анализу ситуации, найму и увольнению сотрудников.

Основная часть. В качестве СУБД для разработки базы данных была выбрана PostgreSQL, которая имеет ряд преимуществ: бесплатное ПО с открытым исходным кодом. Эта СУБД является очень мощной системой; большое количество дополнений - существует очень много дополнений, позволяющих обрабатывать данные для этой СУБД и управлять ими; расширения - существует возможность расширения функционала за счет сохранения своих процедур; объектность – PostgreSQL это не только реляционная СУБД, но также и объектно-ориентированная с поддержкой наследования и много другого.

Реляционная модель представляет собой базу данных в виде множества взаимосвязанных отношений (таблиц), которые используются для хранения информации об объектах, представленных в базе данных.

Исходя из функциональных требований к приложению был определен набор таблиц, описывающий заданную предметную область и позволяющий хранить всю необходимую информацию. Список таблиц представлен в таблице 1.

Таблица 1 – таблицы базы данных приложения

Имя таблицы	Описание
stations	Описывает отделение почтовой связи
employees	Работники почты
districts	Участки, на которые поделен город
subscribers	Подписчики
editions	Печатные издания
subscriptions	Подписки
receipts	Чеки за подписки
waybills	Накладная

Окончание таблицы 2

1	2
typographies	Типографии
print_orders	Заказы на печать для типографий
roles	Роли пользователей
users	Пользователи

Также стоит упомянуть, что обязательная часть любой автоматизированной системы с базой данных являются триггеры. Триггер – это своего рода скрипт, выполняющий SQL-код при наступлении какого-либо события, например, добавление новой записи в таблицу, изменение или удаление записи. Для информационной системы почтовых отделений можно выделить несколько процессов, нуждающихся в подобной автоматизации. Для начала, клиенту необходимо выписать квитанцию после успешного оформления подписки, а также сохранить эту квитанцию в базе данных. Тут как нельзя кстати придется триггер, который при добавлении в базу новой подписки автоматически сформирует квитанцию и добавит в отдельную таблицу. Ещё один похожий процесс – формирование заказов на печать. Заказы одной газеты для одного отделения могут дробиться на несколько штук для разных типографий, поэтому необходим триггер, собирающий количество заказанных газет для каждого почтового отделения. И последний пример – увольнение сотрудника. Если это почтальон – нужно перераспределить участки между оставшимися почтальонами, дабы ни один из участков не остался без обслуживания.

Разработка веб-API данного веб-приложения велась в одном проекте с графическим интерфейсом, такой особенностью обладает выбранная технология разработки – ASP.NET Core 3 MVC.

Платформа ASP.NET Core представляет технологию от компании Microsoft, предназначенную для создания различного рода веб-приложений: от небольших веб-сайтов до крупных веб-порталов и веб-сервисов [1].

Преимущества платформы ASP.NET Core:

- Единое решение для создания пользовательского веб-интерфейса и веб-API.
- Страницы Razor делают создание кодов сценариев для страниц проще и эффективнее.
- Blazor позволяет использовать в браузере язык C# вместе с JavaScript. совместное использование серверной и клиентской логики приложений, написанных с помощью .NET.
- Возможность разработки и запуска в ОС Windows, macOS и Linux.
- Открытый исходный код и ориентация на сообщество.
- Интеграция современных клиентских платформ и рабочих процессов разработки..
- Облачная система конфигурации на основе среды.
- Встроенное введение зависимостей.
- Упрощенный высокопроизводительный модульный конвейер HTTP-запросов [2].

Веб-API при таком подходе будет иметь простую структуру – методы-обработчики будут вызываться при выполнении пользователем GET или POST запросов на соответствующий URL. В самом методе реализуются необходимые действия. Если это действие связано с базой данных, то для взаимодействия с этой самой базой будет использоваться Entity Framework, который весьма удобным образом свяжет API и базу данных.

Заключение. В результате данной разработки были получены такие части веб-приложения с автоматизированной информационной системой как веб-API (или серверная часть) и база данных, служащие для обработки, хранения и доставки пользователям актуальной информации по интересующим их вопросам.

ЛИТЕРАТУРА

1. Сайт о программировании [Электронный ресурс]. – Режим доступа: <https://metanit.com>. – Дата доступа: 24.09.2020.
2. Документация по ASP.NET Core [Электронный ресурс]. – Режим доступа: <https://docs.microsoft.com/ru-ru/ASPNET>. – Дата доступа: 24.09.2020.