

**ПРОЕКТИРОВАНИЕ ИНТЕРФЕЙСА ВЕБ-ПРИЛОЖЕНИЯ ИНФОРМАЦИОННОЙ СИСТЕМЫ
ДЛЯ ПОЧТОВЫХ ОТДЕЛЕНИЙ****П.В. ДЕРЕВЯГА****(Представлено: О.Н. ПЕТРОВИЧ)**

В данной статье описано создание графического интерфейса для типичного веб-приложения с автоматизированной информационной системой. Данный интерфейс подразумевает как простое чтение информации, так и более сложные взаимодействия вроде администраторской части приложения, где можно будет напрямую добавлять и/или править записи в базе данных. В статье приведены и обоснованы выбранные технологии для создания интерфейса.

Введение. На сегодняшний день веб-приложения – это, пожалуй, самый распространенный вид приложений. Практически каждая уважающая себя контора, учреждение или даже просто музыкальная группа имеют свой сайт. Не остаются в стороне и государственные структуры, одной из которых является почта. Для грамотного функционирования почтовой системы крайне важен учёт, а также информирование клиентов.

С такими задачами прекрасно может справиться веб-приложение с автоматизированной информационной системой. А одним из важнейших свойств такого приложения будет графический интерфейс. Простота, понятность, функциональность и удобство должны помочь как сотрудникам почты, так и обычным пользователям.

Основная часть. В качестве технологии для разработки веб-приложения была выбрана технология ASP.NET Core в связке с MVC. Такой подход позволяет разрабатывать совместно в одном проекте как графический интерфейс, так и серверную часть приложения.

Так как программа, работающая с базой данных, сделана в виде веб-приложения, все её формы выполнены в виде веб-страниц. Для разметки страниц использован специальный формат веб-страниц в ASP.NET Core – страницы Razor (cshtml). Такие страницы позволяют использовать стандартный html-код одновременно со специальными атрибутами ASP и вставками кода на C#.

Razor – это интеллектуальный обработчик программного кода динамических веб-страниц ASP.NET. Имеет простой, интуитивно понятный, синтаксис встраивания программного кода в веб-страницы. Razor – механизм визуализации (view engine) поддерживаемый .NET Framework, .NET Core в рамках ASP.NET и специально предназначен для создания веб-приложений. [1]

Внешний вид страниц выполнен в стремлении к максимальной понятности интерфейса для пользователей, а также его привлекательности. Все страницы реализованы при помощи специальных классов и шаблонов фреймворка Bootstrap, который значительно облегчает разработку интерфейса веб-приложений при этом позволяя создавать по-настоящему красивые и стильные страницы.

Bootstrap (также известен как Twitter 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. [2]

Bootstrap является, по сути, ярким лидером среди фреймворков для создания графического интерфейса сайтов. С его помощью можно быстро создать стильный и привлекательный сайт, всего лишь применив готовые шаблоны.

Тип приложения подразумевает под собой наличие как клиентской части, так администраторской, предназначенной для управления контентом. При этом если графический интерфейс клиентской части призван в приятной глазу форме представлять пользователю данные и позволять производить некоторые манипуляции (вход/выход в аккаунт, оформление подписки, просмотр личной статистики), то администратору предоставлен полнофункциональный инструмент для работы с данными. Администратор получил интерфейс, благодаря которому он сможет максимально эффективно добавлять информацию на сайт (в базу данных), изменять и удалять её.

Для комфортного восприятия информации пользователями отображение данных из базы оформлено в виде таблиц. При этом таблицы занимают практически всю ширину страницы, дабы читать записи было максимально легко.

Для администратора приложения существуют специальные страницы, попасть на которые можно только пройдя авторизацию. Интерфейс страниц полностью повторяет обычную часть приложения, за исключением добавления кнопок для добавления, редактирования и удаления записей базы данных. Кроме того, в соответствие каждой из таблиц были реализованы специальные страницы с формами, заполнив которые администратор сможет добавлять или изменять записи. Кроме того, добавление новых записей

происходит на отдельных страницах, представляющих собой форму с полями для заполнения и кнопку подтверждения заполненных данных.

На рисунках 1 и 2 представлены примеры получившегося графического интерфейса, а точнее страница авторизации и управления списком изданий для администратора.

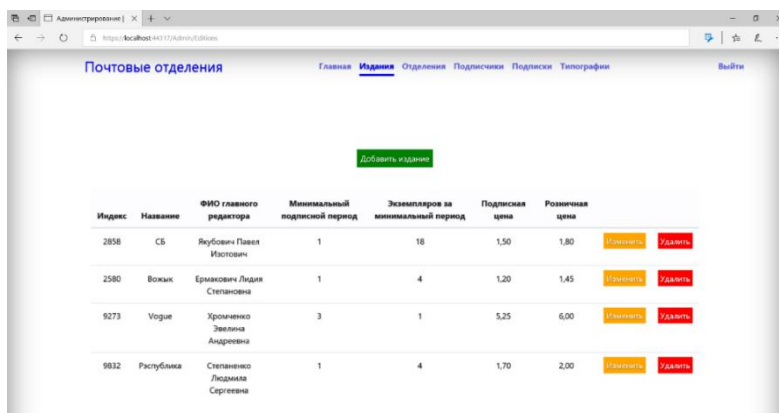


Рисунок 1. – Страница управления списком изданий

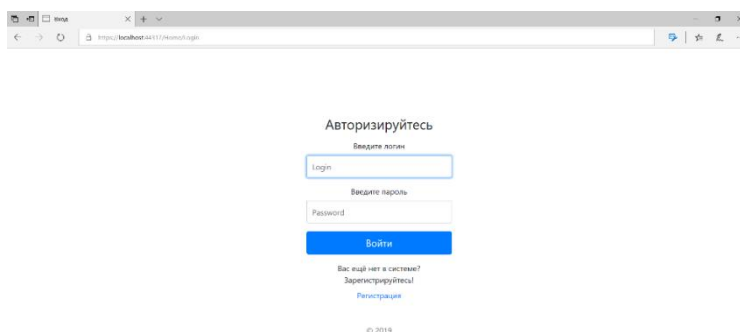


Рисунок 2. – Страница авторизации

Заключение. В результате был разработан графический пользовательский интерфейс для веб-приложения информационной системы почтовых отделений, отвечающий поставленным требованиям как по функциональности, так и по понятности, простоте и привлекательности. Также данный интерфейс был протестирован на предмет некорректной или нестабильной работы и проблем выявлено не было.

ЛИТЕРАТУРА

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