

Рисунок 1 – Страница входа в приложение

После того как пользователь совершит вход в приложение ему открывается доступ к списку сотрудников, вид страницы отображен на рисунке 2.

Employee (Total: 39)	Projects	Position	Start	End	Time Left
Abraham, Jay	-	Scrum Master	-	-	-
Alden, Tille	Self-ful-reliance	BA	03/29/2019	04/02/2019	03:36
Banner, Robert	Blood and Iron	Manager	03/06/2019	-	-
Banner, Bruce	-	Manager	-	-	-
Bloodstone, Elsa	-	BA	-	-	-
Braddock, Brian	-	.Net Developer	-	-	-
Brook, Esdy	Rise of the Panther	.Net Developer	03/22/2018	03/22/2020	03:36:36
Carony, Sean	-	BA	-	-	-
Carlin, Lee	-	Manager	-	-	-

Рисунок 2 – Страница отображения списка сотрудников

По результату выполненной работы, было разработано веб-приложение, которое позволяет производить распределение человеческих ресурсов внутри компании.

## ЛИТЕРАТУРА

1. Официальная документация Nodejs – ode.js Everywhere: [Электронный ресурс]. – 2019. – Режим доступа: <https://nodejs.org>. – Дата доступа: 20.03.2019.
2. Официальная документация React – React Documentation: [Электронный ресурс]. – 2019. – Режим доступа: <https://reactjs.org>. – Дата доступа: 20.03.2019.

УДК 004.042

Студ. А. А. Ермакович

Науч. рук. канд. техн. наук Н. А. Жилияк  
(кафедра информационных технологий БГТУ)

## РАЗРАБОТКА ВЕБ-РЕСУРСА «ОТОБРАЖЕНИЕ АНАЛИТИЧЕСКИХ ДАННЫХ ФАРМАЦЕВТИЧЕСКИХ КОМПАНИЙ»

На сегодняшний день современная фармацевтическая промышленность из обычного производства лекарств для лечения болезней превратилась в большие числа, а именно: прибыль, убытки, спрос. По

этой причине каждая компания, которая занимается производством и продажей лекарственных средств, хотела бы видеть актуальную информацию о продажах своих продуктов по территориальным и временным критериям. По этой причине появились компании, которые выполняют анализ по продаваемости лекарств, количеству заболевших той или иной болезнью, на различных территориях земного шара. Одной из таких компаний является «IQVIA». Конечно каждая фармацевтическая компания, которая обращается к «IQVIA», хочет не только слышать различные цифры о продажах и так далее, а хочет видеть данную информацию в виде красивых графиков и иметь возможность в любой момент их просмотреть самостоятельно. В связи с этим было решено разработать приложение, которое выводит аналитические данные в виде ключевых показателей эффективности.

Перед началом разработки были рассмотрены различные подходы к построению архитектуры приложения. Из многочисленных вариантов был выбран подход разделения приложения на многочисленные уровни и модули, а именно: уровень данных, уровень доступа к данным, уровень пользовательского интерфейса. Данный подход был выбран по причине того, что основное его преимущество заключается в возможности простого внесения изменений в отдельные части приложения, а также упрощения расширения функционала с помощью добавления новых компонентов, не изменяя старые.

Приложение состоит из двух частей: серверной и клиентской.

Серверная часть приложения разработана в виде *RESTful API* на платформе *ASP.NET Core 2.2* [1]. Данная технология предназначена для разработки коммерческих веб-приложений. *ASP.NET Core* получила свою популярность из-за возможности запуска приложения, написанного на данной технологии, в различных операционных системах, таких как: *Windows, Mac OS X, Linux*, т.е. является кросс-платформенной средой разработки. Для работы с базой данных был использован фреймворк *Entity Framework Core* от компании *Microsoft*. Выбор данного фреймворка был обусловлен тем, что он реализует уже многочисленные паттерны разработки и позволяет работать с базой данных, как с простым набором объектов без явной необходимости писать *SQL*-запросы, т.е. позволяет работать с данными на высоком уровне абстракции.

Клиентская часть приложения разрабатывалась как *Single Page Application*, что позволяет не обновлять страницы при переходе между элементами меню, что ускоряет работу приложения, так как обмен данными заключается только в получении информации от *API*-сервера. Для реализации был использован фреймворк *Angular* вер-

сии 5 от компании *Google* [2]. Для написания исходного кода был использован язык программирования *TypeScript*, так как он является строго типизированным и упрощает разработку приложения. Данный фреймворк представляет собой высокоуровневую надстройку для *Javascript*.

При запуске приложения пользователь попадает на страницу с графиками. На данной странице пользователь может посмотреть интересующую его информацию по выбранным критериям, пользовательский интерфейс данной страницы изображен на рисунке 1. При работе с приложением пользователь может изменить критерий по которому будет отображаться информация, а именно: продукт, регион и временной период. Данные, которые отображаются пользователю, берутся из баз данных, список баз данных указывается при разворачивании приложения.

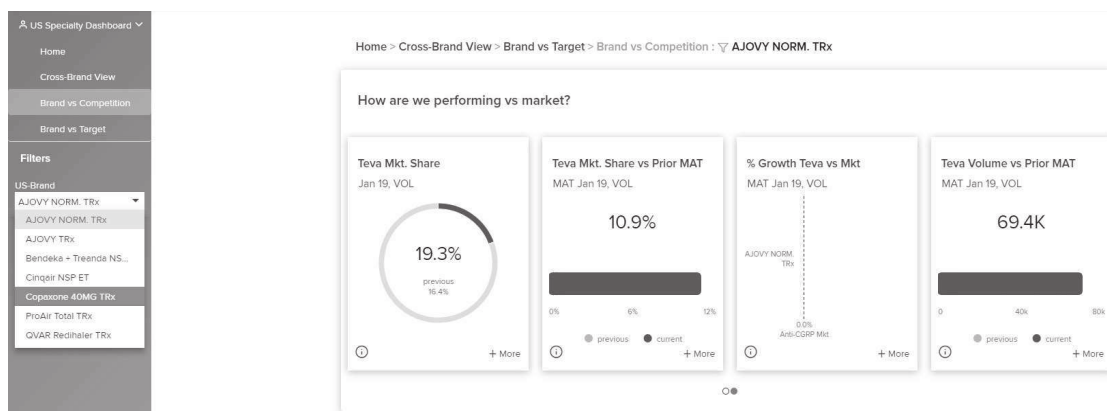


Рисунок 1 – Страница отображения графиков

Результатом разработки стало веб-приложение, которое состоит из двух частей: клиентской, написанной с помощью фреймворка *Angular*, и серверной на платформе *ASP.NET Core* и базы данных *MS SQL Server*.

Во время разработки веб-приложения использовались популярные среды разработки *Visual Studio Community 2017*, *Visual Studio Code* и *Microsoft SQL Server Management Studio*, а также *NPM* и *Angular CLI*.

## ЛИТЕРАТУРА

1. Официальная документация Microsoft - Introduction to ASP.NET Core: [Электронный ресурс]. – 2019. – Режим доступа: <https://docs.microsoft.com/enus/aspnet/core>. – Дата доступа: 20.03.2019.
2. Официальная документация Angular – Angular Documentation: [Электронный ресурс]. – 2019. – Режим доступа: <https://angular.io/docs>. – Дата доступа: 20.03.2019.