Тестовое задание для Full-stack разработчика .NET/Angular

# Проблематика

Компании требуется Web-приложение мониторинга и управления электронными счетами клиентов, которое позволит просматривать все уже выставленные счета, а также создавать новые счета. Для реализации этого приложения требуется создать два решения – web-сервис, который предоставляет API для работы со счетами и браузерное web-приложение для предоставления UI управления счетами.

# Функциональные требования

Данное задание включает в себя реализацию web-сервиса, предоставляющего API, и web-приложения, состоящего из двух web-страниц.

## Структура данных

Счет:

* Дата и время создания счета;
* Дата и время изменения счета;
* Номер счета;
* Статус обработки счета (при создании устанавливается в значение «Новый»);
* Сумма счета;
* Способ оплаты счета.

Статусы обработки счета:

* 1 – Новый;
* 2 – Оплачен;
* 3 – Отменен.

Способы оплаты счета:

* 1 – Кредитная карта;
* 2 – Дебетовая карта;
* 3 – Электронный чек.

## Хранение данных

Для хранения данных используется файл CSV Invoices.csv. При открытии первой страницы данные из файла должны быть считаны и отображены в списке счетов. При сохранении нового счета этот счет должен быть добавлен в файл без повреждения уже присутствующих в файле счетов.

## API

API должен предоставлять средующие функции:

1. Получение всех счетов в системе с постраничным разбиением.

Входные данные: принимается фильтрация, порядок сортировка и параметры страницы.

Выходные данные: страница счетов.

1. Получение одного счета по идентификатору.
2. Добавление счета.
3. Изменение счета.

При добавлении и изменении счета должна выполнятся проверка данных: номер должен быть заполнен, сумма должна быть заполнена и быть числом. В случае неуспешной проверки API возвращает ошибку. В случае успешной проверки счет сохраняется.

## Web-приложение

### Первая страница

Первая страница предназначена для удобного просмотра всех существующих в системе счетов.

Элементы на странице:

* Список счетов в табличной форме с заголовками колонок;
* Кнопка создания нового счета.

Колонки списка:

* Создан (дата и время создания счета в формате ДД.ММ.ГГГГ ЧЧ:мм);
* Номер (номер счета);
* Статус (статус обработки счета: Новый, Оплачен, Отменен);
* Сумма (сумма счета)
* Способ оплаты.

По нажатию на заголовок колонки должна осуществляться сортировка списка по значению этой колонки.

По нажатию на кнопку создания нового счета осуществляется переход на вторую страницу.

### Вторая страница

Вторая страница служит для выставления новых счетов.

Элементы на странице:

* Поле ввода номера счета;
* Поле ввода суммы счета;
* Поле ввода способа оплаты;
* Кнопка «Сохранить»;
* Кнопка «Закрыть».

По нажатию на кнопку «Сохранить» выполняется проверка введенных данных: номер должен быть заполнен, сумма должна быть заполнена и быть числом. В случае неуспешной проверки некорректно заполненные поля должны быть подсвечены красным. В случае успешной проверки счет сохраняется в списке счетов, пользователь перенаправляется на первую страницу.

# Нефункциональные требования

1. Решение должно влючать в себя две части – API работы со счетами на .NET Core и UI в виде SPA на Angular 2+.
2. Часть API должна быть оформлена как решение Visual Studio.
3. Часть UI должна быть оформлена как директория с файлами и поддиректориями.
4. Код API должен быть написан на языке C#, код UI - на языке TypeScript; код должен соответствовать лучшим практикам программирования на этих языках (например, рекомендациям Microsoft для .NET).
5. Дополнительными плюсами будет использование дополнительных UI библиотек для создания профессионального внешнего вида приложения (например, Material).

# Заключение

Тестовое задание является упрощенным вариантом реальных задач компании, поэтому при оценке его решения будет учитываться:

* Насколько аккуратно написан код;
* Насколько подходящими являются выбранные наименования объектов;
* Насколько хорошо разделены зоны ответственности объектов;
* Насколько легко будет поддерживать в дальнейшем это решение.