**Описание задания:**

Задание заключается в создании клиент-серверного приложения для платформы Windows и ASP.NET.

При попадании в приложение пользователь должен ввести данные для входа. После этого пользователь попадает на карту (например, Москвы) где отображается набор маркеров. При клике на маркер открывается всплывающее окно с текстом, хранящимся на стороне сервера. Заголовок всплывающего окна должен формироваться случайным образом в HEX формате при каждом открытии.

**Цель задания:**

Оценить навык архитектурного планирования приложения, работу с REST, скорость вхождения в новую область знаний (картография), навыки написания кода на определённом стеке технологий, навыки работы с PostgreSQL и сопутствующими расширениями, умение писать сетевые запросы в клиентской части приложения, навыки поиска информации в открытых источниках;

**Исходные данные:**

Таблицу картографических данных в базе данных можно заполнить следующим перечнем (разделитель — точка с запятой) данных.

Идентификатор записи; Текстовое содержимое; Долгота; Широта

1;Случайный текст;37.40570068359375;55.93074077571185

2;Очень случайный текст;37.38372802734375;55.697711785689854

3;Случайный ли?;37.518310546875;55.79819333412856

4;Коллизия;37.81768798828125;55.63109707296326

5;Литания против страха;37.79846191406249;55.8475661532989

6;Адресная привязка точки;37.65838623046874;55.654347641744195

7;Помидор;37.6611328125;55.79510545223691

**Требования к реализации:**

1) При написании кода использовать язык программирования C# и платформу ASP.NET для Back-end части приложения и JavaScript/HTML5/CSS3 для Front-end части;

2) Проект должен быть упакован и структурирован;

3) Работа с картой должна быть организована через Mapbox, карты Google или Yandex карты;

4) Пароли пользователей не должны храниться в явном виде;

5) При создании механизма авторизации необходимо использовать технологию JWT (JSON Web Token);

6) Коммуникация между клиентской и серверной частью должна организовываться по методологии REST API;

7) В качестве базы данных использоваться PostgreSQL;

**Пожелания к реализации и допущения:**

1) Описание исходного кода (без избытка);

2) Желательно отказаться от 3rd-party библиотек. Если есть причины их использования, то следует описать какую техническую задачу они решают и почему являются лучшим вариантом для решения пославленной технической задачи;

3) Автоматическое формирование документации к REST API в спецификации Swagger будет плюсом;

4) Таблицу пользователей можно задать заранее любым удобным способом;

5) Приложение подразумевается монолитным, разбивать на микросервисы необходимости нет;

6) Наличие скрипта миграций — плюс;

7) Допускается использования как Raw SQL, так и ORM подхода;

8) Дизайн интерфейса и его элементов — произвольный, wireframe или полноценный макет НЕ требуется; навыки вёрстки в данном задании не оцениваются;

9) Заимствование — это хорошо, пока не превращается в плагиат.