

Создать веб-приложение «**Блог**».

Task1

Заголовок Header – по всей ширине окна. Header: название, картинка. Ширина 100%

Контент Двухколоночная верстка: main-body, right-sidebar. Ширина 80%

Main-body: лента превью статей.

Right-sidebar: меню

Подвал Footer: copyright, back to top.

Copyright: текст. back to top: ссылка/кнопка «телепортирующая» на верх страницы.

Главная состоит из статей.

Статьи: название статьи, дата публикации, текст статьи.

Меню: содержит ссылки на станицы Главная, Гостевая, Анкета.

Гостевая: лента отзывов. Форма для отзывов.

Отзыв: имя автора, дата, текст отзыва.

Форма для отзыва: содержит поля для ввода имени автора и текста отзыва.

Анкета: поля должны состоять из текстовых полей, элементов множественного выбора (**checkbox**), переключателей (**radio**). Для отображения списочных данных написать свой **Helpers** для генерации элементов списка (****, ****). Анкетные данные обрабатываются по запросам **Get** и **Post** одним и тем же методом действия. После прохождения теста пользователь перенаправляется на новую страницу, где отображаются результаты анкетирования.

Task2

1. На основе сущностей предметной области БЛОГ создать классы их описывающие, соблюдая принципы **SOLID**, исключая DI.
2. **Классы и методы** должны иметь отражающую их функциональность названия и должны быть грамотно структурированы в приложении (folders, namespaces).
3. Оформление кода должно соответствовать **C# Code Conventions**.
4. Информацию о предметной области хранить в БД, для доступа использовать **Entity Framework**. В качестве СУБД использовать MS SQL (не Compact).
5. Архитектура приложения должна соответствовать шаблону **MVC**.
6. Код должен содержать **комментарии**.
7. **Уровень доступа к данным** должен быть вынесен в отдельный проект.
8. **В контроллерах** не должно быть создания и прямой работы с контекстом базы данных.

Буква	Означает	Описание
S	<u>Single responsibility principle</u>	<u>Принцип единственной обязанности</u> На каждый <u>класс</u> должна быть возложена одна-единственная обязанность.
O	<u>Open/closed principle</u>	<u>Принцип открытости/закрытости</u> Программные сущности должны быть открыты для расширения, но закрыты для изменения.
L	<u>Liskov substitution principle</u>	<u>Принцип подстановки Барбары Лисков</u> Объекты в программе могут быть заменены их наследниками без изменения свойств программы.
I	<u>Interface segregation principle</u>	<u>Принцип разделения интерфейса</u> Много специализированных интерфейсов лучше, чем один универсальный.
D	<u>Dependency inversion principle</u>	<u>Принцип инверсии зависимостей</u> Зависимости внутри системы строятся на основе абстракций. Модули верхнего уровня не зависят от модулей нижнего уровня. Абстракции не должны зависеть от деталей. Детали должны зависеть от абстракций.