# Лабораторная работа №1. Основы HTML и CSS. Верстка и оформление сайта.

Пререквизиты: Visual Studio 2017 Community Edition версии 15.3 (скачать можно отсюда <https://www.visualstudio.com/downloads/>, при установке нужно отметить модули «ASP.NET and web development», «Azure development» и «.NET Core cross-platform development»)

1. Создать проект ASP.NET Core (с аутентификацией или без).
2. Оформить выбранный доклад [https://1drv.ms/x/s!Ag5pMwumdpilhMoDii4Vxn9Ze\_8sMQ](https://vk.com/away.php?to=https%3A%2F%2F1drv.ms%2Fx%2Fs%21Ag5pMwumdpilhMoDii4Vxn9Ze_8sMQ&cc_key=) в HTML. Разделы доклада можно оформить как в виде блоков одной страницы, так и на нескольких страницах.
3. Каждый доклад предполагает наличие примеров. Примеры можно представить на самом сайте, либо реализовать в одном из сервисов <https://codepen.io/>, <https://jsbin.com/>, <https://jsfiddle.net/>.
4. Если примеры предполагают использование javascript, то javascript-код требуется вынести в отдельные файлы, а в HTML загружать javascript-код с использованием инструкции <script src=’{url-javascript-файла}’></script>.
5. Оформление элементов страницы должно строиться на CSS-классах, чтобы избежать дублирование css-правил на однотипных элементах страницы. Весь css-код требуется вынести в отдельные файлы, а в HTML загружать css-код с использованием инструкции <link rel=’stylesheet’ href=’{url-css-файла}’>.
6. Сайт одновременно должен эстетично выглядеть и быть информативным.

Примеры, как не стоит оформлять сайты:

* <http://www.dokimos.org/ajff/>
* <http://www.spaceistheplace.ca/>
* <http://www.ronoslund.com/>
* <http://www.theworldsworstwebsiteever.com/>

Советы по популярному в настоящее время плоскому-дизайну: <http://tilda.education/articles-flat-design>.

1. В случае использования сторонних примеров на них обязательно нужно ссылаться.

# Лабораторная работа №2. Отправка данных форм на сервер. Контактная форма.

В этой ЛР требуется добавить на сайт из ЛР№1 страницу с контактной формой и реализовать отправку сообщений на email.

1. Создать контроллер ContactController.
2. Добавить в контроллер действие Index. Действие не принимает параметров и должно возвращать представление с контактной формой.
3. Контактная форма должна содержать кнопку отправки сообщения на сервер и три текстовых поля:
   1. email для обратной связи;
   2. тема сообщения;
   3. текст сообщения.
4. По нажатию кнопки отправки данные формы должны отправляться на сервер методом POST в действие Send контроллера ContactController.
5. Форма должна поддерживать валидацию полей. Валидацию требуется реализовать c использованием атрибутов модели <https://docs.microsoft.com/en-us/aspnet/core/mvc/models/validation> с проверкой состояния модели в действии контроллера. Правила валидации:

* email должен быть валидным;
* тема сообщения не должна превышать 50 символов;
* текст сообщения не должен превышать 2000 символов;
* поле email обязательно к заполнению;
* поле ‘текст сообщения’ обязательно к заполнению.

В случае, если модель не валидна, сообщение не должно отправляться, а на странице должны быть отображены ошибки валидации.

1. Если модель валидна, то сообщение должно отправляться на email администратора сайта. Настройки отправки требуется вынести в файл конфигурации, например, в appsettings.json. Типичный набор настроек для отправки email-сообщения:

* Адрес smtp-сервера (например, smtp.gmail.com)
* Порт smtp-сервера (например, 587)
* Логин
* Пароль
* Email получателя

Для отправки email можно использовать любую библиотеку, например: <https://github.com/jstedfast/MailKit>, <https://github.com/lukencode/FluentEmail> или любую иную.