

Предисловие

Форма выполнения:

1. Выполняется в бригаде, состоящей из 1 – 4 человек
2. В качестве веб-сайта выбирается одна из 2 лабораторных работ.
3. Архитектура может быть любая – SPA (популярные нынче фреймворки: React, Vue, Angular), MPA, PWA.
4. Сразу сдаются все лабораторные, которые здесь представлены.
5. В отчёте вёрстка, скрины того, что получилось + исходники.

Лабораторная работа 3

HTML разметка

1. Разметить собственный проект.
2. Каждая страница должна быть размещена в отдельном файле.
3. Документ должен проходить проверку на валидность: <https://validator.w3.org/nu/>.
4. Обязательно должны присутствовать следующие элементы: форма обратной связи, header и footer, будущий слайдер.
5. Ключевые критерии:
 - Должны быть правильно выделены header и footer, которые будут одинаковыми для каждой страницы.
 - У всех изображений в теге **Ошибка! Не указано имя файла.** Прописан размер.
 - Использовано минимально возможное количество HTML-элементов (нет лишних элементов).
 - Не должно быть разницы в коде одинаковых элементов (например, в карточках питомцев, товарах в каталоге и т.д.)

Если архитектура приложение SPA или PWA

Так как разметки по большей части нет в привычном смысле, необходимо будет заготовить компоненты, которые будут «встраиваться» в страницу.

Логику работы программы можно не описывать, а просто сделать константы для значений, заполнить списки вручную или имитировать AJAX-запросы.

Языки можно использовать любой (JS, TS, вроде ещё Microsoft что-то хотели сделать на основе C# и Razor, есть ещё, конечно, PHP и Python), фреймворки тоже.

CSS подключать не нужно на данном этапе, но вы можете прописывать классы для элементов по желанию.

Теория: Учебник: <https://html5book.ru/html-html5/> Интерактивные курсы: <https://htmlacademy.ru/courses/basic-html-css>

Лабораторная работа 4

Сетка и изображения

1. Вырезать из макета изображения в формате, соответствующем назначению изображения. Изображения хранятся в отдельной папке в проекте
2. Подключить шрифты, использованные на макете
3. Выполнить раскладку блоков на страницах, используя flex и grid
4. Ключевые критерии:
 - 1) Расположение и размеры блоков совпадают с макетом (+/- 5 px)
 - 2) Для стилизации не использованы #id
 - 3) Нет вложенности селекторов больше двух.
 - 4) Для блока, у которого есть фоновое изображение, прописан фоновый цвет, который соответствует преобладающему цвету изображения (пока изображение не загружено, страница выглядит похоже на макет).
 - 5) Вёрстка проходит тест на переполнение контентом.
 - 6) Вёрстка идентично отображается в последних версиях браузеров Chrome, Opera, Firefox, Safari, Edge (по желанию). Изображения только вырезать, подключать их будем в дальнейшем
 - 7) Адаптивность дизайн под мобильные устройства.

Теория: Форматы изображений и принципы использования:
<https://htmlacademy.ru/blog/boost/graphics/image-formats>

Интерактивные курсы: <https://htmlacademy.ru/courses/basic-html-css>
Документация по css: <https://developer.mozilla.org/ru/docs/Web/CSS/>

Лабораторная работа 5

Декоративные элементы

В предыдущих трёх лабораторных работах вы уже создали разметку, сетку и подготовили изображения. Теперь необходимо оформить все декоративные элементы на этих страницах. Должны быть выполнены следующие требования:

1. Все размеры, отступы и расположение элементов выполнены по макету;
2. подключены все шрифты, указанные в макете и техническом задании;
3. размеры и толщина текста соответствуют размерам в макете и техническом задании;
4. подключены все изображения, размеры и расположение по макету;
5. свёрстаны состояния элементов. По желанию можно использовать препроцессоры (LESS, SASS) При работе с изображениями стоит руководствоваться следующим правилом:

- если изображение является контентным, окружённым поясняющим текстом или являющимся самостоятельным смысловым блоком, то такое изображение должно быть добавлено с помощью тега `img` с необходимым и заполненным атрибутом `alt`;

- если же изображение является преимущественно оформительским и его потеря не приведёт к потере смысла, то такое изображение должно быть добавлено с помощью свойства `background-image`.

За `inline` стиль можно получить по голове и отправится переделывать.

Теория: Документация по CSS:
<https://developer.mozilla.org/ru/docs/Web/CSS/>