

ПРОЕКТ

Обязательная часть проекта, выполняется самостоятельно или в команде 2 человека:

Необходимо сверстать адаптивно (mobile-first) и реализовать на JavaScript веб-страницу, макеты которой находятся в архиве с заданием. Допускается использование других дизайн-макетов аналогичной сложности и с аналогичными функциями (выпадающее меню, форма, слайдер), в этом случае необходимо предоставить исходные дизайн-макеты на защите проекта.

В файле page.png находится общий макет со всеми блоками страницы. В файле mobile.png находится макет страницы на смартфоне. В файле menu.png находится изображение выпадающего по наведению мышки меню. В файле mobile-menu.png находится изображение меню на мобильном устройстве.

В файле fonts.txt находятся названия используемых на странице шрифтов. Шрифты необходимо найти и подключить самостоятельно.

В шапке страницы отображается фоновое видео из файла video.mp4.

Форма внизу страницы должна отправляться с помощью JavaScript без перезагрузки страницы на один из сервисов для сохранения форм из предыдущих заданий.

Дополнительное задание повышенной сложности, выполняется обязательно в команде 2-3 человека (необходимо показать коммиты в git):

Реализовать с помощью одного из популярных реактивных JS-фреймворков (React, Angular или Vue.js на выбор) отображение во всплывающем диалоговом окне контактной формы при нажатии на кнопку «связь с нами». Внешний вид и поля формы аналогичны форме внизу страницы в дизайн-макете.

Реализовать анимацию отображения и сокрытия формы с помощью RequestAnimationFrame. Желательно учитывать положение нажатой кнопки на экране для иллюстрации плавного изменения состояния приложения с помощью анимации в ответ на действия пользователя.

Использовать LocalStorage для сохранения введенных ранее значений и History API для работы с URL текущей страницы аналогично предыдущему заданию. Предполагается использование компонента роутинга для фреймворка.

Для отправки запроса на сервер использовать fetch. Во время выполнения запроса на сервер блокировать кнопку отправки и отображать анимацию на кнопке или форме. В случае ошибки при отправке запроса отобразить сообщение об ошибке с просьбой отправить форму еще раз.

Для меню сайта и меню на смартфонах реализовать анимацию плавного отображения и сокрытия с помощью CSS-переходов (CSS-transition), CSS-анимации или JavaScript. Допускается использование библиотек для реализации анимации.

Желательно использование Redux, MobX, Vuex или другой аналогичной библиотеки для управления состоянием. Возможно использование специальных библиотек и компонентов

работы с формами, таких как, например, Formik, а также библиотек компонентов, таких как Materialize, Vuetify и т. д. Допускается использовать любые возможности и дополнительные модули выбранного фреймворка.

СДАЧА РАБОТЫ И ОЦЕНИВАНИЕ

Страница или сайт публикуется в интернете в виде статического веб-сайта, в Moodle на проверку отправляется ссылка на сайт и список членов команды от каждого участника команды. Каждый член команды получает полный балл за проект согласно критериям ниже. В случае использования фреймворков публикуется версия сборки страницы для разработчика (с поддержкой sourcemaps для просмотра и отладки исходного кода JavaScript). После оценивания работы ее необходимо снять с публикации в интернете.

Работа проверяется в актуальной версии браузера Chrome или Firefox.

Время выполнения обязательной части проекта до 10 часов, включая самостоятельную работу.

За обязательную часть проекта, выставляется **до 100 баллов**:

- 20 реализация всех блоков страницы;
- 20 реализация мобильной версии страницы;
- 10 вставка видео в шапке сайта;
- 10 реализация выпадающего меню;
- 10 реализация мобильной версии выпадающего меню;
- 10 корректное подключение и отображение шрифтов;
- до 10 баллов за соблюдение стандартов кодирования HTML и JavaScript,
- до 10 баллов за отсутствие Error при валидации HTML-разметки по <https://validator.w3.org/> и отсутствие Error и Warning при валидации JavaScript с помощью <https://www.jshint.com/>.

За неточное соответствие макетам могут сниматься баллы.

Допускается использование библиотек JavaScript, например плагина Slick JQuery.

Обязательную часть необходимо выполнить **до зачетной недели**.

За дополнительное задание повышенной сложности дополнительно выставляется **до 260 баллов**:

- 100 корректная реализация отображения, валидации и отправки формы с помощью реактивного фреймворка;
- 40 использование библиотеки управления состоянием (Redux, MobX или аналог) для реализации блокировки кнопки и анимации при выполнении запроса, корректной обработки ошибки при выполнении запроса, анимации отображения формы;
- 40 использование библиотеки управления роутингом для реализации работы с URL и корректной обработки кнопки «Назад» с применением History API;
- 20 отправка данных формы на сервис форм из предыдущего задания с помощью fetch, библиотеки axios или аналогичной основанной на промисах библиотеки;
- 20 анимация открытия и закрытия попапа контактной формы с помощью RequestAnimationFrame;
- 10 анимация открытия и закрытия меню в шапке сайта с помощью CSS-переходов;
- 10 анимация открытия и закрытия меню на смартфонах с помощью JQuery, JQuery Animate или иной библиотеки JavaScript;
- 10 блокировка кнопки и анимация при выполнении запроса;
- 10 корректная обработка ошибки при выполнении запроса;

Дополнительное задание необходимо выполнить **до консультации на экзамене в вашей группе или до конца сессии (если у вас нет экзамена)**.

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

Ознакомьтесь материалом по JavaScript в разделе «РАЗРАБОТКА КЛИЕНТСКИХ ВЕБ-ПРИЛОЖЕНИЙ» учебного пособия "Веб-разработка и веб-сервисы" и содержанием лекции №6-9.

В каталоге `img` находятся уже нарезанные картинки, которые можно использовать в верстке.

Для измерения значений необходимых отступов и размеров элементов на макетах в пикселах можно использовать графический редактор, например GIMP, либо использовать сервис `figma.com` и исходный проект дизайн-макета, ссылка на который есть в файле `figma.txt`. В случае отличий отображения элементов в проекте `figma` от макетов в задании приоритет имеют макеты в задании.

Дополнительное задание повышенной сложности:

Ознакомьтесь с содержанием лекций №10-13. Используйте официальную документацию выбранного вами фреймворка.

Примеры анимации с использованием `RequestAnimationFrame` и CSS-переходов находятся в файлах `raf.html` и `css-transition.html`. Дополнительный материал по CSS-transition: <https://html5book.ru/css3-transition/>

Дополнительный материал по использованию `Fetch`:
https://developer.mozilla.org/ru/docs/Web/API/Fetch_API/Using_Fetch