Цель: Изучить различные библиотеки и фреймворки создания клиентской части веб-приложений.

Задание: В ходе выполнения работы вам необходимо 3 раза с использованием различных технологий создать клиентскую часть веб приложения.

Требование к клиентской части:

1. Целостность (интерфейс должен предназначаться для выполнения какой-либо одной функции);
2. Наличие минимум двух страниц с различными URL.
3. Наличие дизайн-концепции веб интерфейса (нельзя просто прикрутить стандартные кнопочки и успокоиться!);
4. Адаптивный веб-дизайн (правильное отображение на мобильных устройствах);
5. Наличие форм ввода данных. Обработка форм ввода данных.
6. Наличие изображений (картинок);

Особые требования: вам необходимо реализовать одно и то же приложение 3 раза с использованием различных технологий. Получившиеся интерфейсы и поведение системы должно быть одинаковое на различных версиях. При освоении технологии попытайтесь изучить принципы и паттерны построения приложений, которые применяются при её использовании.

Технолоигии: при выполнении задания вы должны выбрать две технологии из следующих трех: Angular, ReactJS, VueJS. Третью технологию вы можете выбрать на свое усмотрение из следующего списка: jQuery, Ember.js, BackboneJS, Sematic-UI, Foundation, Svelte, Preact и другие

Требование к отчету: В отчете необходимо показать всю проделанную работу, а также сравнить  технологии, выделить их плюсы и минусы, сделать вывод под какие задачи подходит каждая из технологий.

Контрольные вопросы:

1. Трехуровневая архитектура веб-приложения. Основные звенья.
2. Язык HTML. История создания. Предназначение. Синтаксис.
3. Язык CSS. История создания. Предназначение. Синтаксис.
4. Препроцессоры CSS. SASS и LESS. Предназначение. Синтаксис.
5. Язык JavaScript. Основные конструкции языка и модель работы.
6. Библиотека React. Принципы создания клиентской части веб-приложения.
7. Фреймворк Vue.js. Принципы создания клиентской части веб-приложения.
8. Платформа Angular. Принципы создания клиентской части веб-приложения.
9. Одностраничные и многостраничные веб-приложения. Отличия. Преимущества и недостатки каждого подхода.
10. Прогрессивные веб-приложения. Определение. Преимущества. Ограничения.