

**Основные требования к контрольной работе «Разработка веб-приложения»  
по дисциплине «САиПИС с РВР» 2017/18**

**Общие требования:**

- Необходимо создать веб-приложение (на локальном хостинге), представляющее собой функционирующее клиент-серверное приложение.
- Минимальное количество веб-страниц – 5.
- Веб-страницы, объединенные общей темой и авторским дизайном создаются на языках HTML, XHTML, JavaScript, XML, библиотек и инструментов JavaScript и содержат возможности (меню, гиперссылки) для быстрого перехода на другие страницы.
- Ко всему сайту для форматирования текста/объектов должна применяться единая каскадная таблица стилей.
- Авторизацию осуществить с помощью Javascript, логин и пароль необходимо сохранять в СУБД.
- Используемые в проекте данные, библиотеки, плагины HTML-, js- и другие файлы, а также ресурсные данные (видео-, звуковые файлы, изображения и т.д.) должны храниться в отдельных папках.
- К работе должен прилагаться отчет о проделанной работе (пояснительная записка примерно 6-10 страниц).
- Оформление пояснительной записки к контрольной работе должно быть в соответствии с требованиями и образовательным стандартом БГУИР (см. на сайте БГУИР).
- Основная часть должна содержать описание выполненных требований (листинг кода) и подкрепляться экранными копиями (скриншотами), должен быть сделан вывод или заключение о проделанной работе.

**1. Вёрстка:**

- Использование блочной вёрстки с использованием 2-3 видов позиционирования.
- Гибкая блочная верстка (flexible), с указанием display: flex в css.
- Допускается использование Bootstrap, flexbox и других инструментов.

**Основные требования:**

**2. Архитектура проекта:**

- Реализация модульной концепции AMD (возможности RequireJS - не менее 4-5 реализованных модуля, массив зависимостей).
- Реализация клиентской и серверной части.
- Предпочтительно использование MVP и других паттернов проектирования.

**3. Серверная сторона**

- Использовать возможности node.js.
- Запуск всего приложения идет через сервер.
- Настройка и организация клиент-серверного обмена производится самостоятельно.
- Предпочтительно использование npm, framework express.

**4. Клиентская сторона, использование библиотек, применение динамического HTML, основные требования:**

- Использование возможностей CSS, HTML5.
- Должна присутствовать форма для заполнения с валидацией введенных данных и сохранением их в СУБД.
- Перехват и обработка событий (документа, клавиатуры, мыши, формы, фокуса,).
- Использовать интерактивное удаление и добавление элементов, в том числе вставку дочерних элементов и элементов-потомков между узлами DOM имеющегося document-содержимого.
- Использовать библиотеку jQuery, Dojo или Prototype (1 по выбору или несколько).
- Применить фильтры и селекторы контента, видимости, форм, атрибутов и др.
- Использование динамических элементов на странице (динамическое меню, интерактивную анимацию, текст или меню с всплывающей подсказкой, динамический выпадающий список).
- Использовать шаблонизатор (Mustache, Jade (Pug)) для организации вывода значений.

**5. Использование Ajax+ jQuery:**

- Загрузка данных в формате json (в файле json - не менее 8 видов различных данных с обязательным наличием массива, например, массива оценок или предметов), а также из html-файла и txt-файла.
- Привязка (bind) 3 любых событий по выбору вида: focus(), .focusin(), .focusout(), .load(), .resize(), .scroll(), .unload(), .click(), .dblclick(), .mousedown(), .mouseup(), .mousemove(), .mouseover(), .mouseout(), .mouseenter(), .mouseleave(), .change(), .select(), .submit(), .keydown(), .keypress(), .keyup(), .error().

**6. Хранилище данных**

Хранение данных (заполнение формы регистрации, логин, пароль и другие данные) осуществить в документо-ориентированной БД MongoDB (Mongoose для MongoDB).

- Дополнительно можно использовать новые технологии, библиотеки и другие технические средства.