Задание 4.

Необходимо разработать frontend для сайта по выбранной теме. На основе технического задания, требований и вёрстки, представленных в заданиях 1 и 3, необходимо создать javascript-скрипты для сайта. Результатом будет являться набор файлов формата JS, созданных в текстовом редакторе, например, Visual Studio Code.

**Обработка элементов формы**

Зачастую бывают такие ситуации, когда от пользователя должны быть получены некоторые данные, необходимые для решения поставленной задачи. Кроме того, существует такое свойство алгоритма, как массовость – если с одними исходными данными был получен верный результат, то и с другими исходными данными, приемлемыми в данной задаче, также должен быть получен верный результат. Для реализации этих целей возможно использование форм HTML-страниц. Для создания форм имеется специальный тег <form>. У этого тега имеются такие атрибуты как action и method. Первый задаёт имя программы, которая будет обрабатывать данные полученные с формы, второй – определяет метод запроса. Отправка данных формы происходит по нажатию кнопки типа submit.

**Метод GET**

При отправке данных с помощью метода GET содержимое формы добавляется к URL после знака вопроса в виде пар имя=значения, объединенных с помощью знака &:

action?name1=value1&name2=value2.

Здесь action – это URL-адрес программы, которая должна обрабатывать форму. Имена name1, name2 имена элементов формы, а value1, value2 – значения этих элементов. В принципе создавать HTML-форму необязательно, достаточно добавить к адресу нужные переменные и их значения, поэтому данный метод имеет существенный недостаток – любой может поделать значения параметров. С другой стороны этот недостаток позволяет достаточно удобно отлаживать скрипты.

**Пример:**

*<HTML>*

*<BODY>*

*<FORM METHOD=get action=1.HTML>*

*name1 <input type=text name="name1"><br><br>*

*name2 <input type=text name="name2"><br><br>*

*name3 <input type=text name="name3"><br><br>*

*<INPUT type=submit value="отправить"><br>*

*</form>*

*</BODY>*

*</HTML>*

**Метод POST**

При использовании этого метода содержимое формы кодируется точно так же, как и для метода GET, но вместо добавления строки к URL содержимое запроса посылается блоком данных как часть операции POST. Этот метод рекомендуется для передачи больших по объему блоков данных.

Информация, введенная пользователем и отправленная серверу с помощью метода POST, подается на стандартный ввод программе. Передать данные методом POST можно только с помощью HTML-формы, поскольку данные передаются в теле запроса, а не в заголовке. При использовании POST пользователь не видит передаваемые серверу данные.

**Реализация обработки форм web-приложения**

В данном задании представляется перечень программных модулей, которые были разработаны для функционирования web-приложения. Приводится название модуля и его характеристика (реализуемые функции).

В данном задании следует привести реализуемые функции обработки данных, которые призваны автоматизировать разрабатываемое web-приложение. При этом можно выделить и детализировать два подмножества функций: реализующих служебные функции (например, проверки пароля) и реализующих основные функции ввода первичной информации, обработки, ведения справочников, ответов на запросы и др.

Выявление состава функций, их иерархии и выбор языка общения (например, языка типа «меню») позволяет разработать структуру сценария диалога, дающего возможность определить состав кадров диалога, содержание каждого кадра и их соподчинённость.

При разработке структуры диалога необходимо предусмотреть возможность работы с входными документами, формирование выходных документов, корректировки вводимых данных, просмотра введённой информации, проект с файлами нормативно-справочной информации, протоколирования действий пользователя, а также помощь на всех этапах работы.

В этом задании следует выбрать способ описания диалога. Как правило, применяется два способа описания диалога. Первый предполагает использование табличной формы описания. Второй использует представление структуры диалога в виде орграфа, вершины которого перенумерованы, а описание его содержания в соответствии с нумерацией вершин, либо в виде экранов, если сообщения относительно просты, либо в виде таблицы.

На основе полученных результатов строится дерево программных модулей, отражающих структурную схему web-приложения, содержащего программные модули различных классов:

* выполняющие служебные функции;
* управляющие модули, предназначенные для загрузки меню;
* модули, связанные с вводом, хранением, обработкой и выдачей информации.

В данном параграфе необходимо для каждого модуля указать идентификатор и выполняемые функции.

Описание программных модулей должно включать блок-схемы алгоритмов работы основных модулей web-приложения и их описание.