Работа №1

Разработка статической веб-страницы

Цель работы

Получить базовые навыки создания статических веб-страниц с использованием языка разметки HTML и каскадных таблиц стилей (CSS).

Задача

По заданному вариантом макету разработать HTML-страницу. Страница должна корректно отображаться в последних версиях браузеров Google Chrome, Safari, Mozilla Firefox. Необходимо использовать Flexbox.

Запрещается использование сторонних библиотек и фреймворков, необходимо использовать «чистые» HTML и CSS.

Результаты работы необходимо опубликовать в собственном репозитории на GitHub.

Варианты заданий приводятся ниже. Также можете предложить собственные варианты (по согласованию с преподавателем).

Полезные ссылки

1. До начала работы и после её окончания следует проверить код согласно рекомендациям: https://habr.com/ru/company/htmlacademy/blog/566244/
2. Статья по Flexbox (перевод): https://vk.cc/9DbrpB
3. Оригинал статьи по Flexbox: https://css-tricks.com/snippets/css/a-guide-to-flexbox/
4. Гайд по адаптивной верстке: https://html5book.ru/adaptivnaya-vyorstka-sayta/
5. Гайд по использованию Git: https://www.atlassian.com/ru/git/tutorials/what-is-version-control

Работа №2

Адаптивная верстка и media-запросы

Необходимо доработать страницу, полученную в результате выполнения работы №1 таким образом, чтобы она адаптировалась к экранам различной ширины. В качестве ориентиров используйте как минимум экран мобильного телефона.

Варианты заданий

Далее приводятся варианты заданий. Часть из них доступная по ссылке.

Варианты №№1-4: https://www.figma.com/file/5EGZWDsnJEomjvuHDuG6TJ/Untitled

Часть – ниже.

После выбора варианта задания необходимо заполнить форму, указав выбранный вариант:

https://forms.gle/sLCJJpWKqc5tzyN27

№5 Социальная сеть

Макет: https://vk.cc/c6oxlo

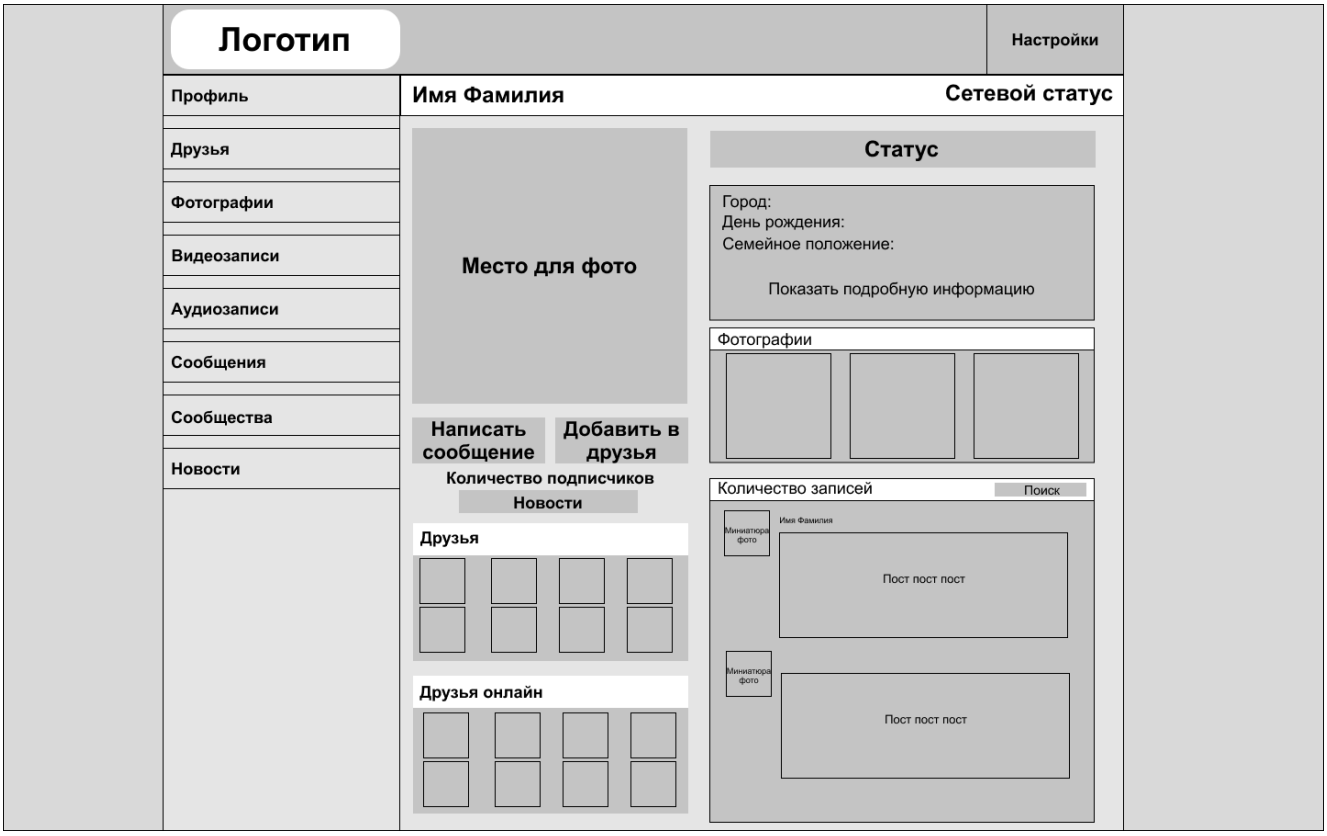


Рисунок 1. Макет главной страницы варианта "Социальная сеть".

№6 Блог

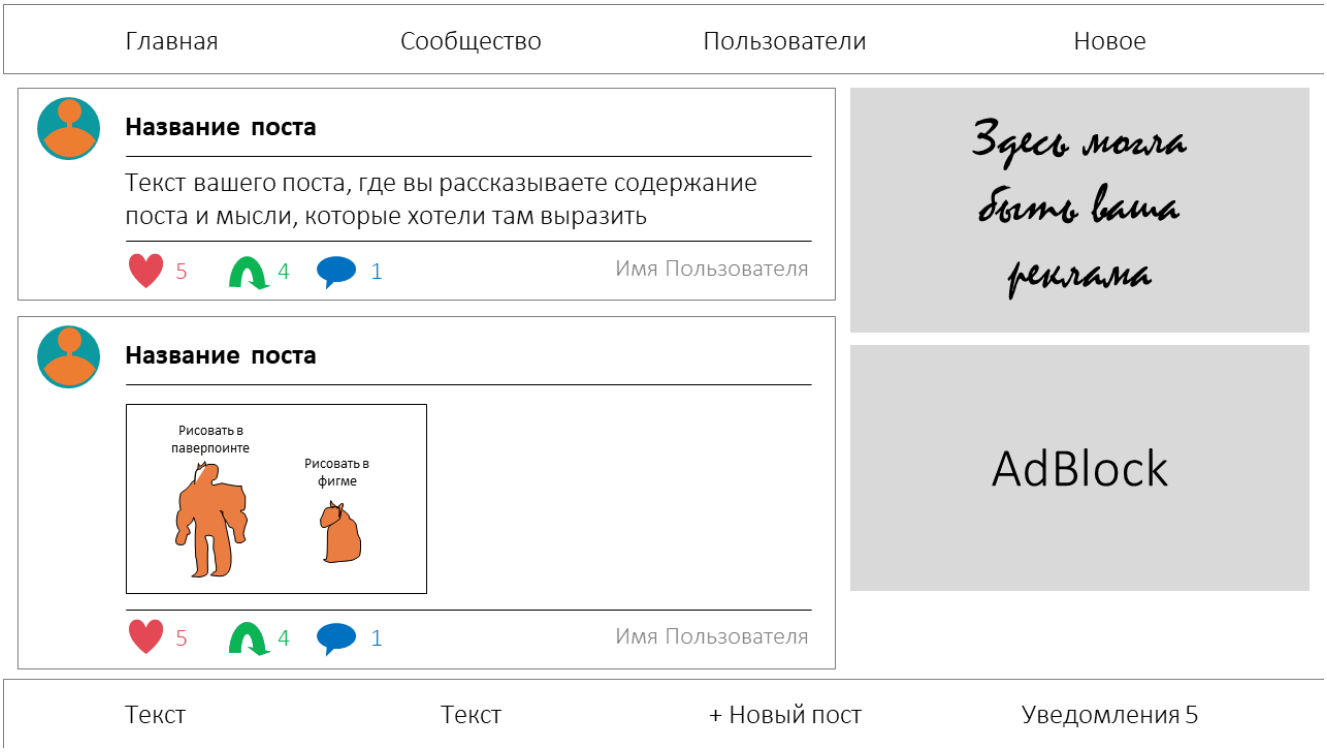


Рисунок 2. Макет главной страницы варианта "Блог".

№7 Планировщик задач

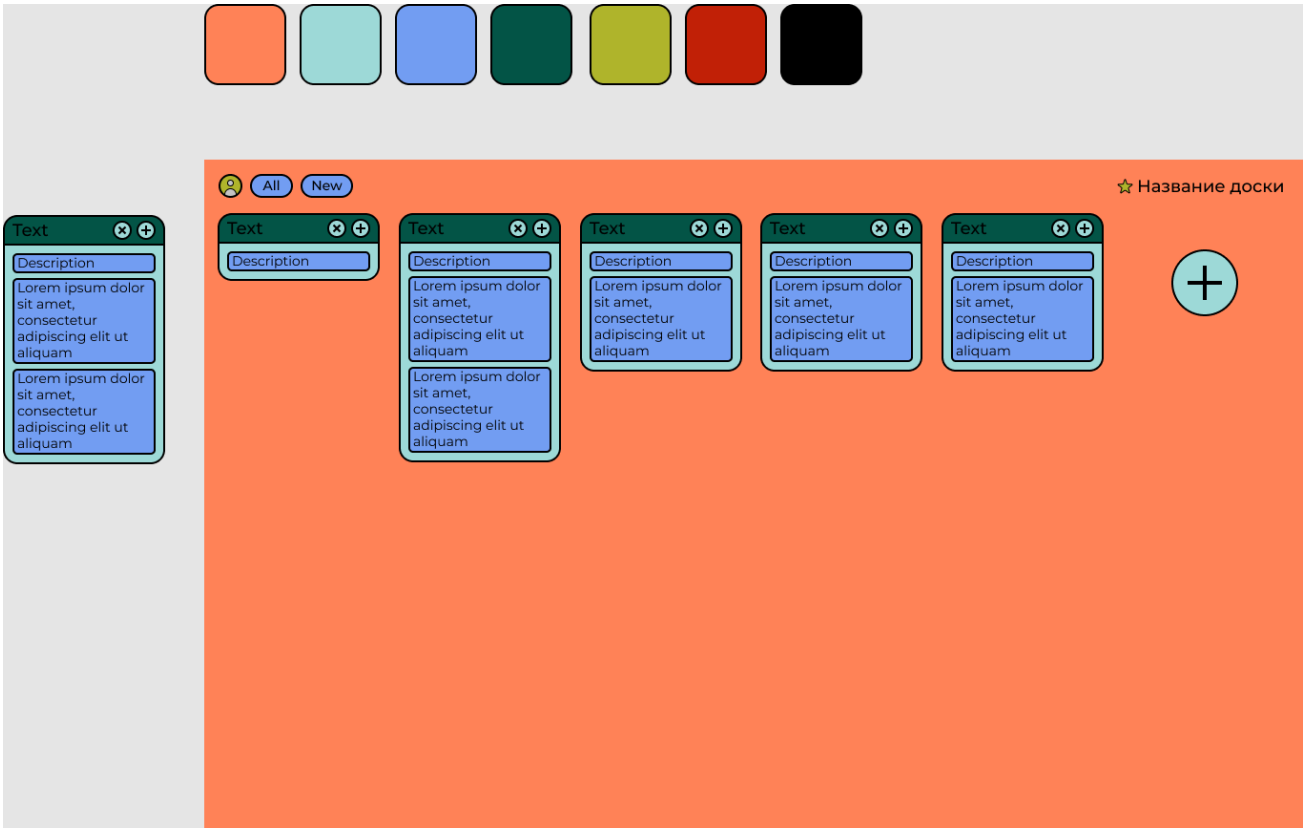


Рисунок 3. Макет главной страницы варианта "Планировщик задач".

№8 Интернет-магазин

Версия 1

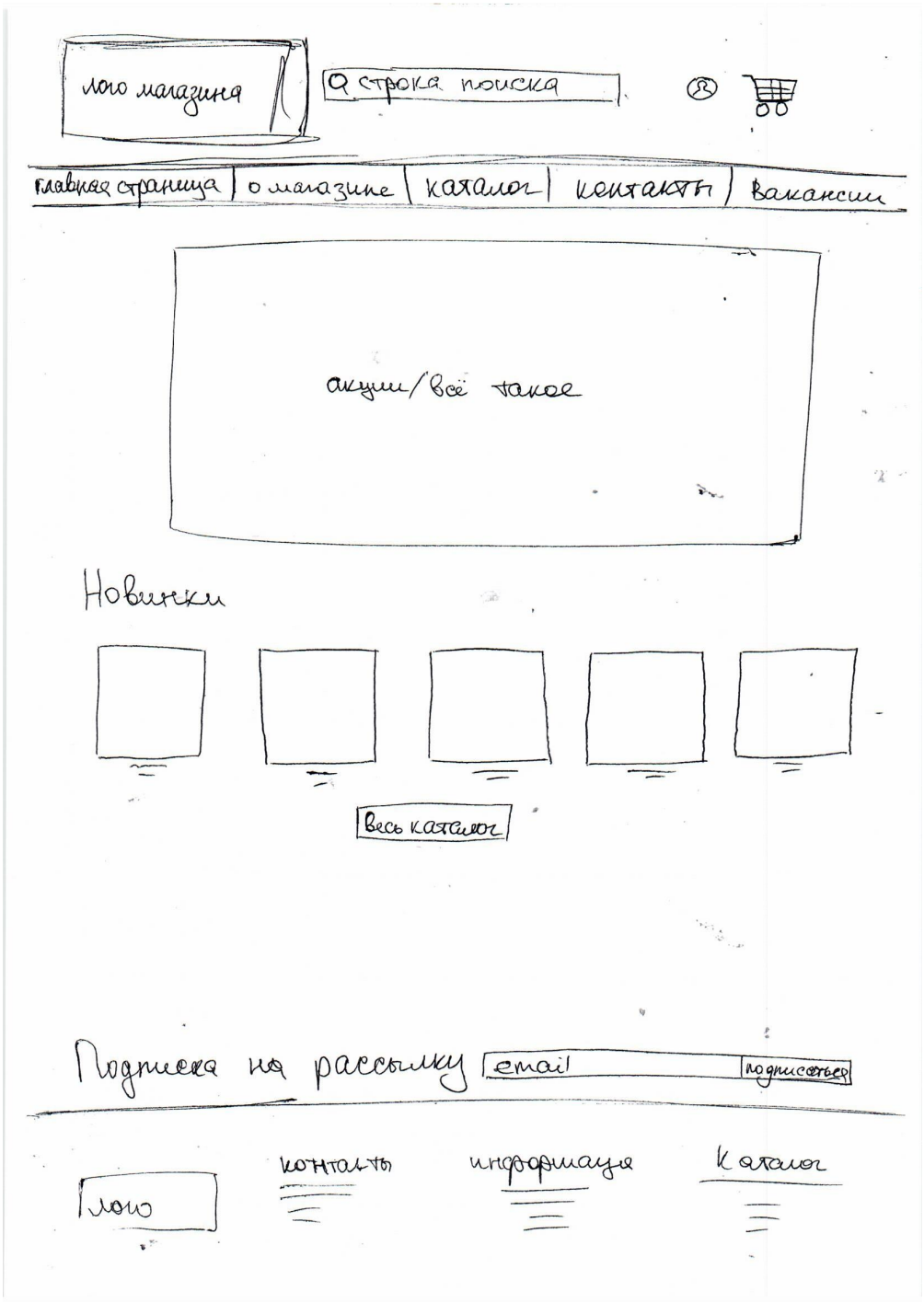


Рисунок 4. Макет главной страницы варианта "Интернет-магазин", версия 1.

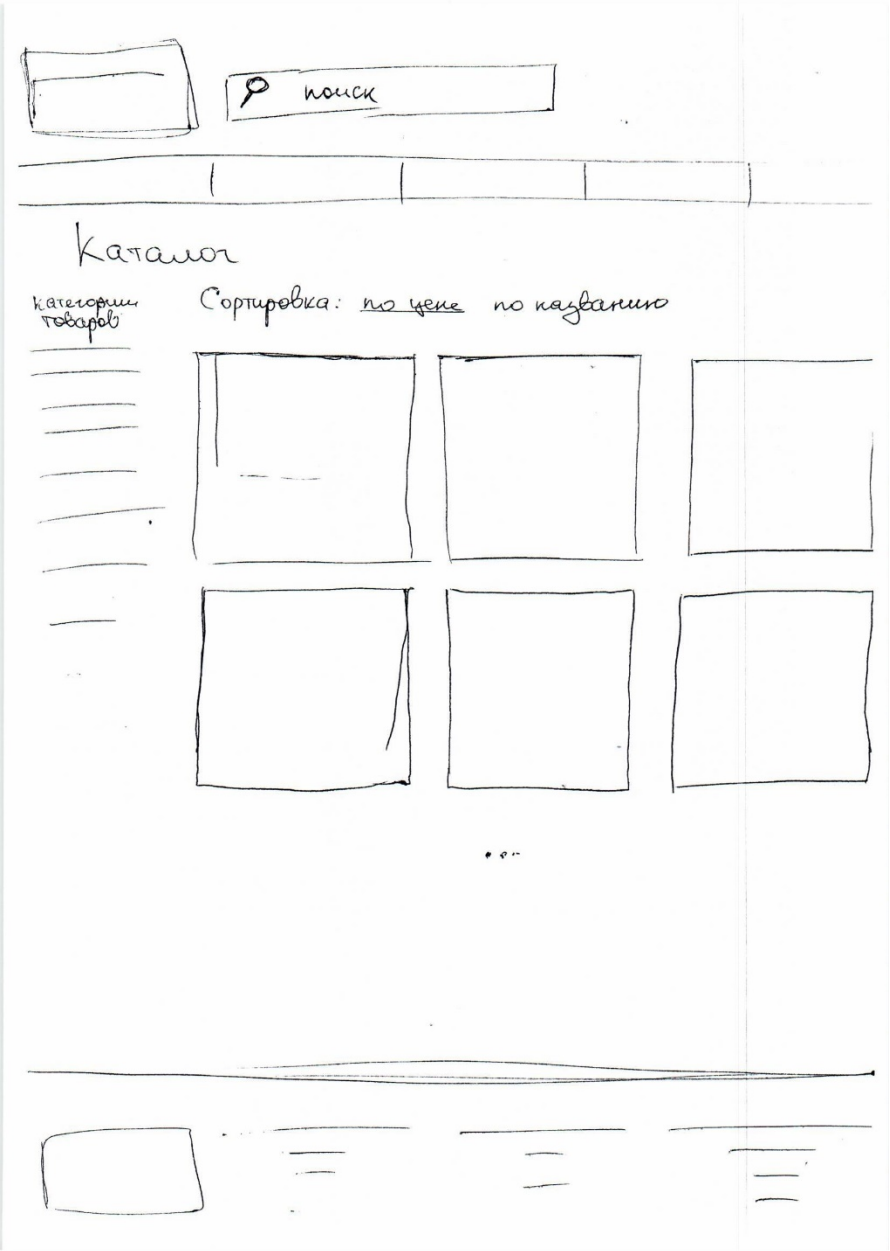


Рисунок 5. Макет страницы каталога варианта "Интернет-магазин", версия 1.

Версия 2

Ссылка на макет: https://www.figma.com/file/pfbj3cL5AapKnKXznbpweZ/Untitled?node-id=0%3A1

Работа №3

Знакомство с языком JavaScript

Цель

Познакомиться с синтаксисом и семантикой языка JavaScript, понятием Document Object Model, Browser Object Model.

Задача

Дано: HTML-страница, на которой есть кнопки «Создать таблицу», «Добавить строку», «Удалить строку №» и текстовое поле для ввода чисел. Далее описано поведение, связанное с нажатием на каждую из них. По-умолчанию все кнопки, кроме «Создать таблицу» заблокированы (используйте атрибут disabled). Таблица должна содержать не менее двух столбцов с произвольным содержимым, однако первый столбец обязательно содержит номер строки.

Создать таблицу

На страницу добавляется элемент <table>. Для простоты доступа к таблице сразу же задайте ей атрибут id. При повторном нажатии на кнопку в случае, если таблица уже существует, необходимо показать модальное окно с сообщением об ошибке (alert).

Добавить строку

Происходит добавление новой строки в конец таблицы.

Удалить строку

Выполняется удаление строки с указанным номером. Номер указывается в текстовом поле, расположенном рядом с этой кнопкой, при этом должна выполняться валидация значений по следующим признакам: во-первых, это должно быть число, а во-вторых, строка с соответствующим номером должна существовать.

Полезные ссылки

1. Учебник по JS: https://learn.javascript.ru/
2. События браузера: https://learn.javascript.ru/introduction-browser-events
3. https://www.section.io/engineering-education/building-a-calculator-a-javascript-project-for-beginners/

Работа №4

Разработка клиент-серверного приложения

Цель

Изучить основы разработки серверных компонентов приложений, обработки HTTP-запросов.

Задача

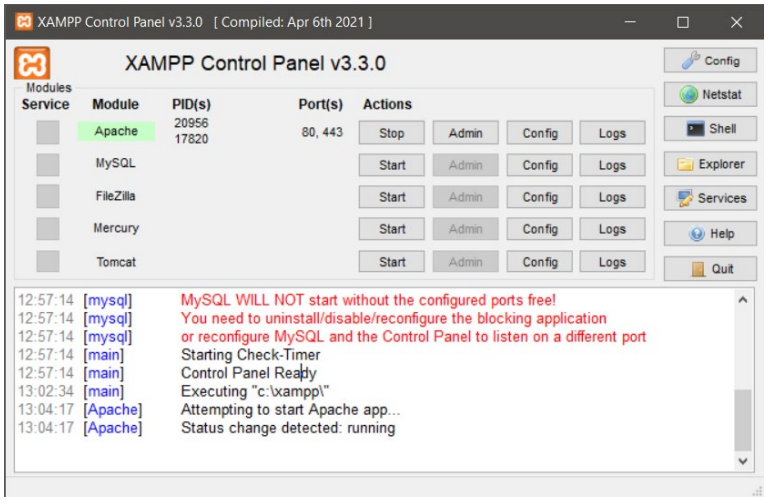
Основываясь на результатах выполнения лабораторной работы №2, разработать северную часть веб-приложения с использованием языка PHP. Серверная часть должна возвращать пользователю динамически формируемую веб-страницу, контент которой зависит от полученного запроса.

Шаг 1. Настройка веб-сервера

Для начала необходимо настроить инфраструктуру, которая позволит динамически генерировать веб-страницы в ответ на HTTP-запросы пользователя. Существует множество веб-серверов, некоторые распространенные, используемые в индустрии – Apache, Nginx, IIS. Развертывание для учебных целей этих веб-серверов возможно, но может потребовать дополнительного обучения, не связанного с данным курсом, поэтому предлагается использовать XAMPP. Это пакет приложений, включающий веб-сервер (Apache), интерпретатор языка PHP, свободную реализацию СУБД MySQL – MariaDB, и другие средства. Его легко установить по ссылкам:

- для Windows: скачать
- для OS X: скачать.

Для установки XAMPP просто следуйте шагам мастера-установщика. После установки XAMPP можно проверить, что всё работает. Для этого достаточно нажать кнопку Start напротив Apache, после чего открыть в браузере ссылку http://localhost/.



Для удобства можно создать каталог, в котором будет размещаться код данной работы. Для этого можно нажать на кнопку Explorer панели управления XAMPP. Откроется каталог, в котором необходимо найти подкаталог htdocs, в котором размещаются файлы, доступные по протоколу HTTP (т.е. данный каталог «обслуживается» веб-сервером). В нем можно создать каталог web, и размещать собственные php-скрипты, html-файлы и другие ресурсы.

Шаг 2. Создание простого веб-приложения

Необходимо создать набор страниц, которые позволят выполнять операции CRUD для набора данных, соответствующего варианту задания. То есть, эти страницы позволят создавать, редактировать и

удалять элементы данных. Например, для варианта «Социальная сеть» это будут персональные страницы, для варианта «Интернет-магазин» это товары. Все данные на данном этапе должны храниться в XML-документе. Структуру документа можно задавать произвольную, главное – обеспечить хранение заданных вариантом данных.

В результате должен быть получен следующий набор страниц:

list.php – страница, на которой отображается список всех элементов данных в виде гиперссылок, ведущих на страницу **index.php** с соответствующим параметром id. Страница должна содержать также ссылку на страницу **create.php**.

index.php – страница, на которой отображаются все данные согласно варианту. Параметр GET-запроса: id, являющийся числом. Страница должна содержать ссылку на страницы **list.php**, **update.php** и **delete.php**.

create.php – страница, содержащая форму для добавления новых элементов данных согласно варианту. Для формы должен быть создан обработчик POST-запроса. Обработчик должен предусматривать валидацию данных по критериям, заданным вариантом. В случае некорректного ввода данных серверный скрипт должен генерировать информативные и понятные пользователю сообщениях об ошибках. После успешного добавления элемента данных должно выполняться перенаправление на созданную страницу (**index.php?id=<новый id>**, где <новый id> - целочисленный идентификатор).

Также на странице должны отображаться гиперссылки на страницы **update.php** и **delete.php** с заданным идентификатором.

update.php – страница, содержащая форму для редактирования элемента, id которого передан в качестве параметра. Функциональность и поведение аналогичны странице **create.php** за тем лишь исключением, что при загрузке в форме отображаются актуальные значения из XML.

delete.php – страница, содержащая форму для удаления элемента. Содержит кнопку, по нажатию на которую на сервер отправляется POST-запрос, выполняющий удаление данных по заданному в GET-запросе id, а также ссылку «Назад», которая ведет на страницу **index.php** с соответствующим id.

Варианты

Социальная сеть

Идентификатор пользователя.

Редактируются анкетные данные пользователя (ФИО, дата рождения, город).

Блог

Идентификатор записи (поста).

Редактируется заголовок и текст записи, устанавливается дата редактирования (создания).

Планировщик задач

Идентификатор задачи.

Редактируется текст задачи, дедлайн, связанный цвет отображения карточки.

Интернет-магазин

Идентификатор товара.

Редактируется название товара, цена, описание.

Работа №5

Использование базы данных MySQL.

Цель

Научиться использовать реляционную базу данных в составе веб-приложения.

Задача

Создать базу данных, содержащую таблицу, хранящую элементы данных согласно варианту (аналогичные предыдущей работе). Рекомендуется использовать MySQL в составе XAMPP, а также phpMyAdmin для управления базой данных.

Веб-приложение должно реализовывать те же самые CRUD-операции, однако в качестве источника будет выступать созданная база данных.

Не требуется использовать средства объектно-реляционного отображения (ORM), достаточно написать запросы в виде текста на языке SQL.

Работа №5

Асинхронные HTTP-запросы

Цель

Научиться использовать реляционную базу данных в составе веб-приложения.

Задача

Изучить средства асинхронного взаимодействия клиентской и серверной частей веб-приложений. Для этого необходимо реализовать простейший WebAPI, который будет реализовывать CRUD-функциональность, аналогичную созданной в предыдущих работах. Отличие составляет формат обмена сообщениями – запросы и ответы должны быть в формате JSON, а для выполнения запросов необходимо использовать XMLHttpRequest.