МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФГБОУ ВО "Пензенский государственный университет архитектуры и строительства"

Инженерно-строительный институт

Кафедра информационно-вычислительных систем

КУРСОВОЙ ПРОЕКТ

по дисциплине «Информационные системы "Клиент-Сервер"»

Тема: «Разработка главной страницы магазина книг»

**Выполнил**: Куприянко Глеб Михайлович

студент 2 курса очной формы обучения направления подготовки 09.03.02 Информационные системы и технологии

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись студента)

**Научный руководитель**:

Гвоздева Ирина Геннадьевна

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись руководителя)

Пенза

2018

Содержание

[Задание 3](#_Toc516326099)

[Реализация 4](#_Toc516326100)

[Локальный сервер – denwer 4](#_Toc516326101)

[PSD макет страницы 6](#_Toc516326102)

[Раздел header страницы – меню 9](#_Toc516326103)

[Баннер страницы 12](#_Toc516326104)

[Раздел-статистика страницы 14](#_Toc516326105)

[Раздел-превью книг со скидками 15](#_Toc516326106)

[Раздел-мотивация для чтения 24](#_Toc516326107)

[Форма обратной связи 26](#_Toc516326108)

[Footer ( подвал ) страницы 33](#_Toc516326109)

[Результат 35](#_Toc516326110)

[Вывод 36](#_Toc516326111)

[Список использованной литературы 36](#_Toc516326112)

# Задание

Создать главную страницу магазина книг.

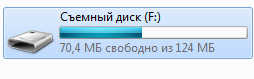
Добавить:

1. Раздел превью книг со скидками ( связать с базой данных );
2. Форма обратной связи.

# Реализация

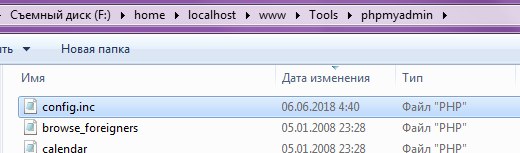
## Локальный сервер – denwer

Denwer я решил для удобства установить на свободную флешку:



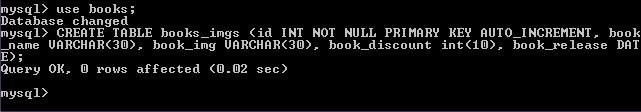
Для нормальной связи с базой данных мне пришлось отредактироваь файл

config.inc.php:



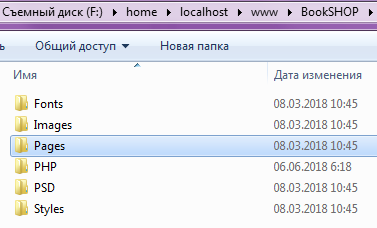


Саму базу данных создаю в MySql:

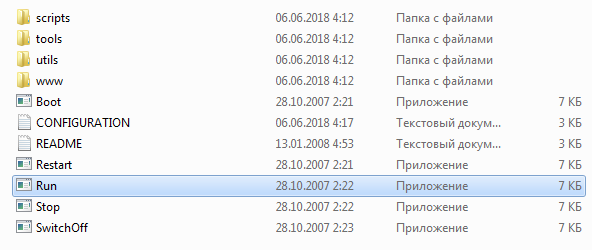


Создадим таблицу. Она пригодится позже, когда я создам блок «Превью» со скидками.

Создаю папку для сайта с необходимой мне структурой:



И так, у меня теперь есть возможность заходить в на localhost ( 127.0.0.1 ), а так же в phpMyAdmin для работы с базами данных. Для этого нужно всего лишь запустить denwer:

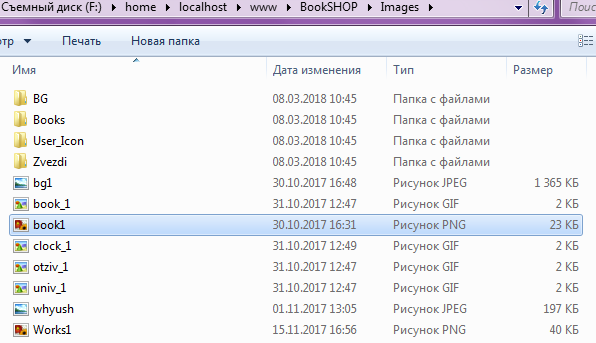


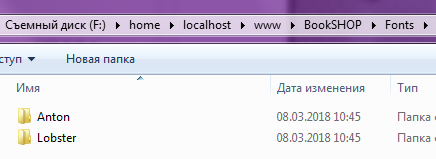
Чтобы перейти в определённый раздел сайта, необходимо лишь указать путь к нему относительно папки localhost ( в моём случае ).

## PSD макет страницы

Чтобы не тратить огромное количество времени на вёртску и дизайн, лучше начать с PSP-макета.

Заодно можно скачать и подключить все необходимые шрифты и картинки:

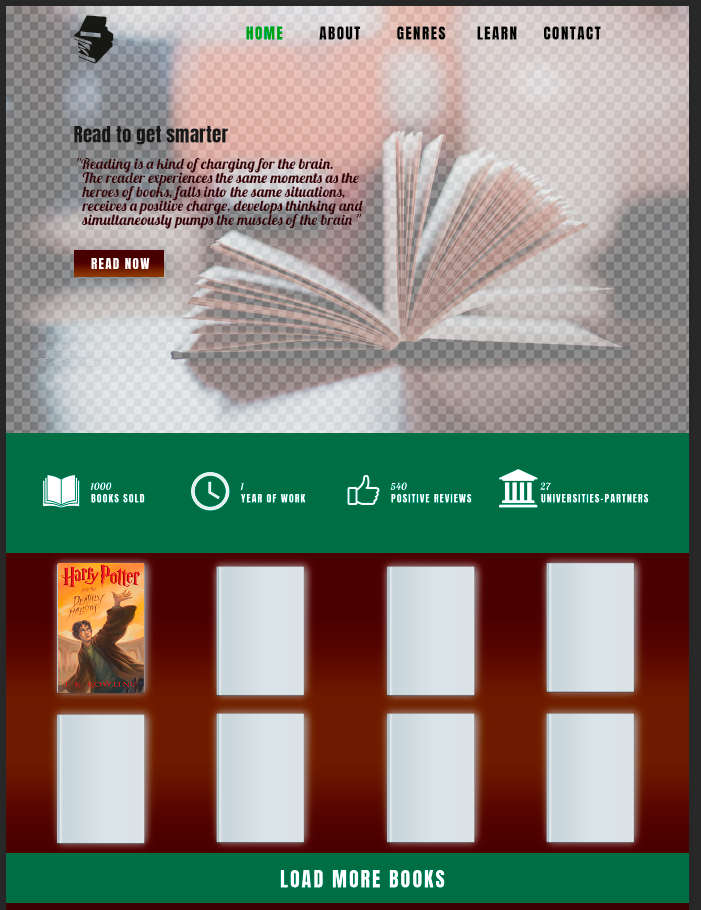




И хоть в итоге я и поменял подключение на прямые ссылки, но никогда не повредит держать при себе локальные версии шрифтов, на всякий случай.



У меня получилось нечто подобное:





Задний фон в некоторых блоках сделан с помощью градиентов, точно такие же я позже применю и в CSS.

Так же, шаблон для наложения обложек книг был сделан отдельно:



Подгрузив его в основной PSD-файл, я применил к нему внешнее свечение.

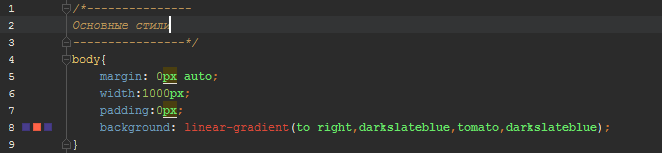
## Раздел header страницы – меню

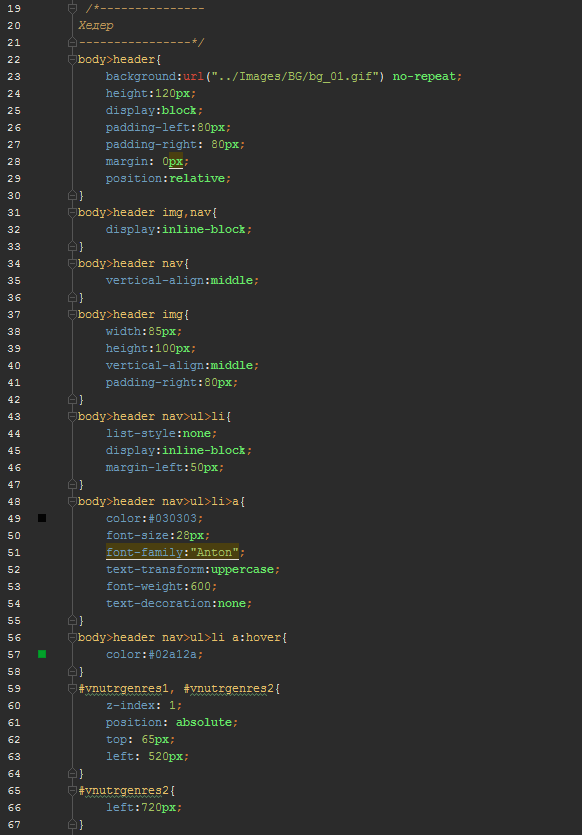
Внутри раздела header я делаю 2 раздела:

1. Визитная картинка – книжки
2. Навигационное меню

К пунктам списка-меню сразу привязываю ссылки на будущие разделы страницы, задаю базовые стили страницы и привожу header к виду PSD-макета:



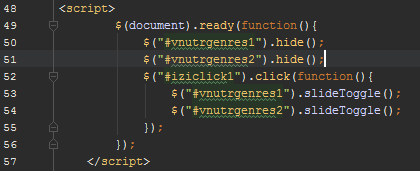




Картинку – задний фон аккуратно нарезаю на кусочки в Photoshop и верхнюю часть подключаю к header с помощью css.

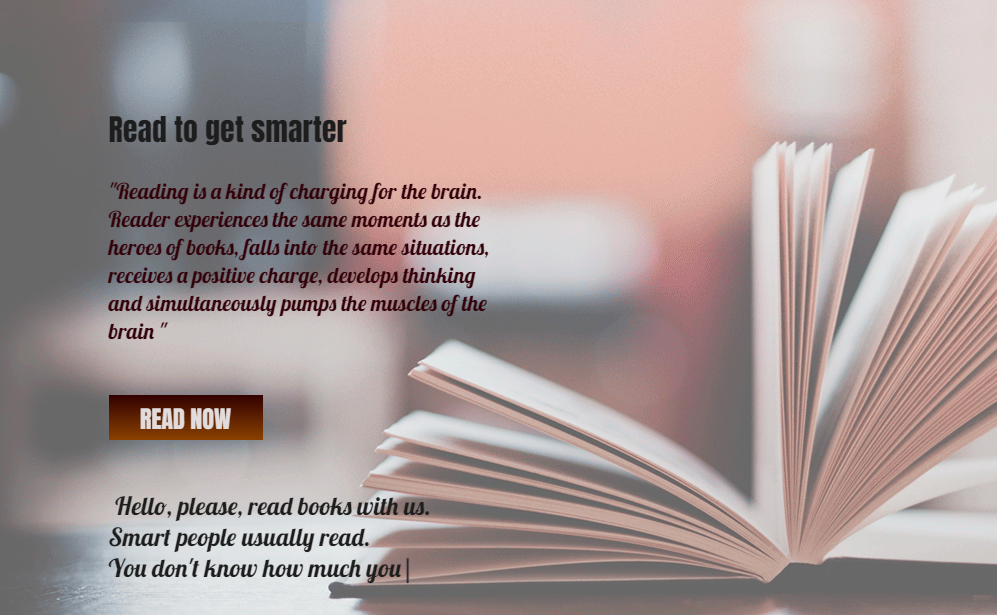
У пункта меню «Жанры» создаю меню второго уровня, состоящее из 2 списков, выводить его буду на баннер с помощью jQuery.



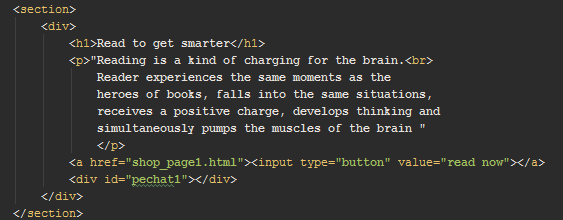


## Баннер страницы

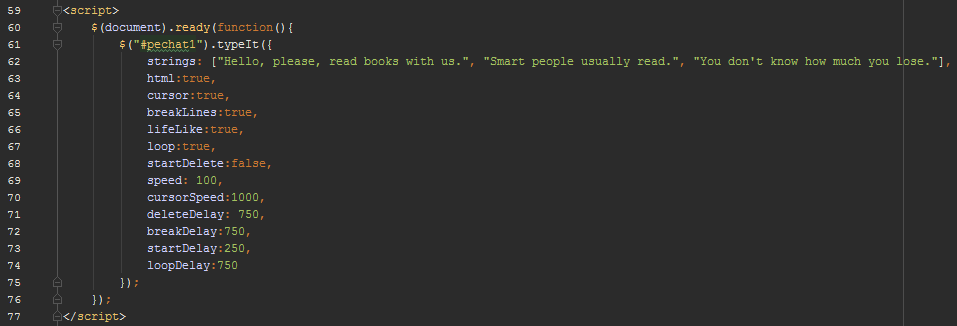
Баннер может оказаться хорошим способом привлечь внимание пользователя, поэтому есть смысл с первых секунд знакомства с сайтом предложить ему почитать:



На деле баннер – это небольшая часть html-кода, состоящая из заголовка, параграфа, кнопки-ссылки и контейнера для функции jQuery – печатающегося текста.



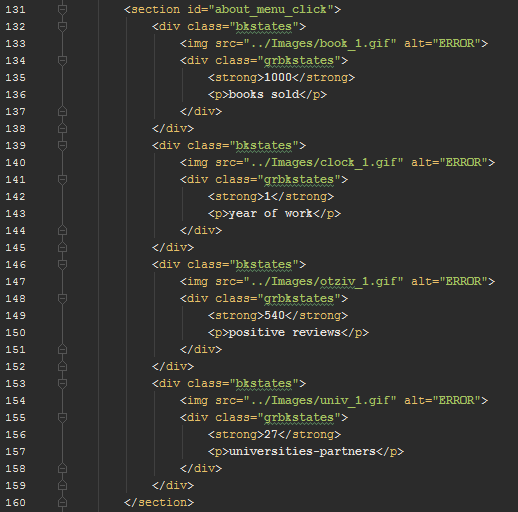
Никаких особых стилей баннер не содержит. Текст оформлен с помощью подключённых шрифтов и изменения цвета, секция расположена с левой стороны баннера, задний фон которого является продолжением header, что придаёт ему элегантности.



## Раздел-статистика страницы

Далее идёт небольшой раздел со статистикой магазина, потенциально вывод значений можно связать с базой данных.



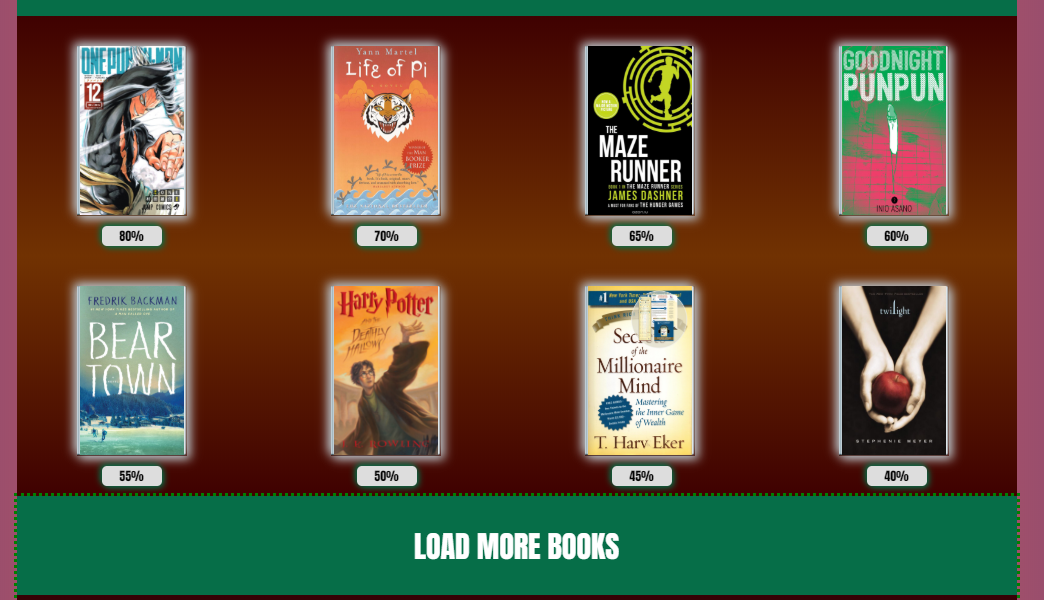


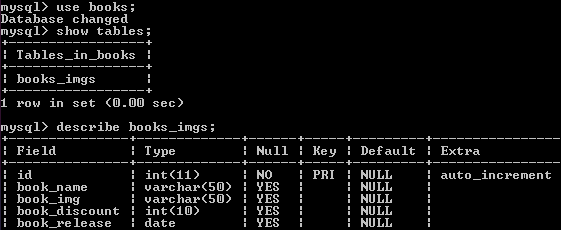
С помощью CSS-стилей блоки расположены на одинаковом расстоянии друг от друга.



## Раздел-превью книг со скидками

Переходим к более динамическому элементу страницы – к секции с превью скидок:



Ранее я уже создал базу и данных таблицу в MySql:

Пора разобраться, что тут к чему.

Я создал таблицу с 6 столбцами:

1. id – идентификатор записи ( позже изменю название ) ( уникальное поле, является ключом для связи с другими таблицами );
2. book\_name – название книги ( тип текст );
3. book\_img – ссылка на обложку книги ( тип текст );
4. book\_discount – величина скидки в процентах ( тип число );
5. book\_release – дата издания книги ( или первой главы ) ( тип дата );

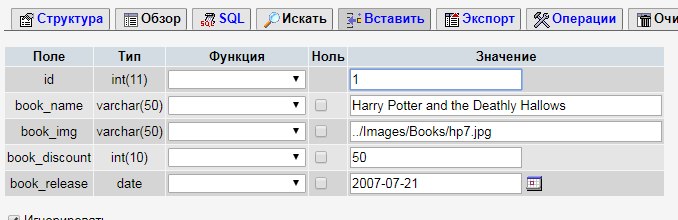
Теперь можно перейти в phpMyAdmin и далее редактировать таблицу уже там:



Редактировать строки и столбцы таблицы можно прямо здесь. Например:

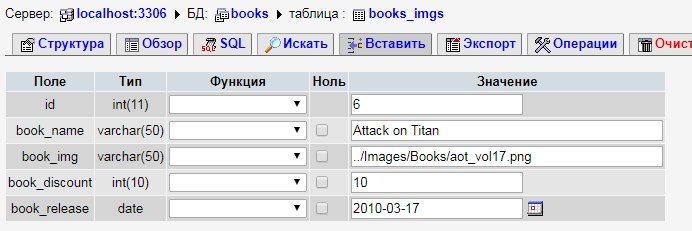


Таким образом, я могу начать добавлять записи о книгах:

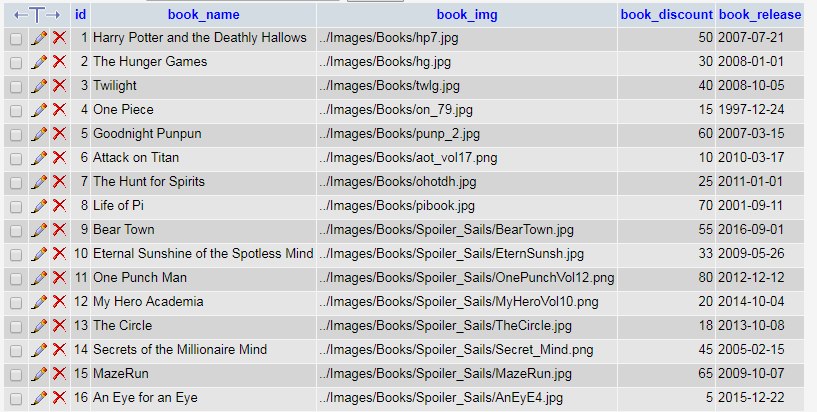


Так как запросы будут задействованы в index.php, то путь к картинке будет именно от index.php.

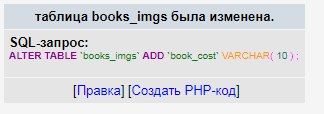
Продолжаю добавлять записи:

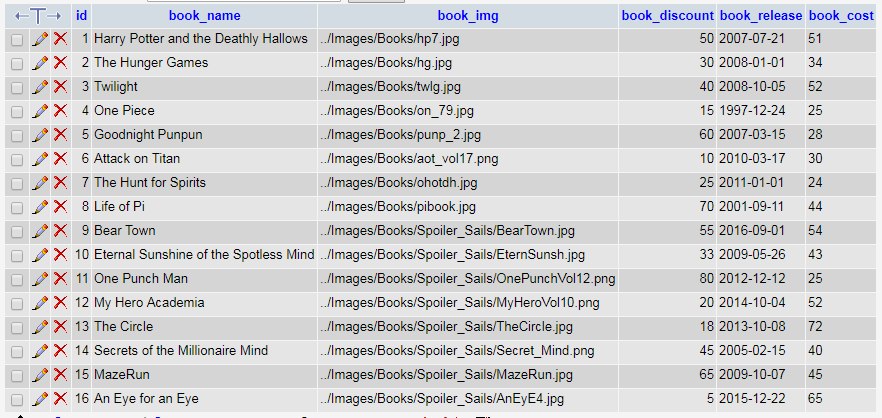


После заполнения таблица выглядит вот так:

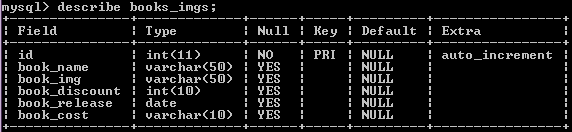


Добавим ещё один столбец – стоимость книги:



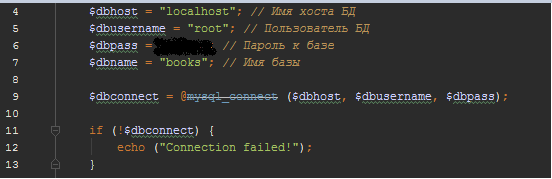


Теперь структура таблицы в MySql выглядит так:



Пора выводить данные из базы данных на страницу.

Создаю подключение напрямую в index.php.



Для подключения к существующей БД используется функция mysql\_connect.

Она передаёт параметры:

1. Имя хоста БД ( по умолчанию localhost );
2. Имя пользователя БД ( по умолчанию root );
3. Пароль к БД ( по умолчанию отключен, но в моём MySql он есть );
4. Имя базы данных ( в моём случае – books ).

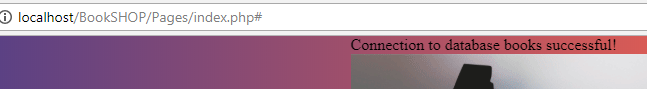
Далее использую функцию как переменную $dbconnect, добавляя небольшую проверку. Если после 9 строки переменная $dbconnect не будет определена, то самым очевидным выводом будет неудачное подключние к БД. В таком случае выводится предупреждение.

Следующий шаг для получения данных состоит в выборе нужной базы, хранящейся на сервере denwer-а. Для этого используется функция mysql\_select\_db.



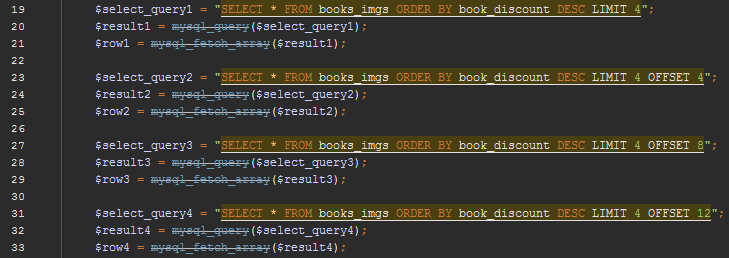
Отслеживать возникновение ошибки буду с помощью функции die, которая прерывает работу программы и выводит предупреждение о нарушении её работы.

Так же при удачном соединении выводится сообщение об успехе.



Пора сформировать запрос на вывод.

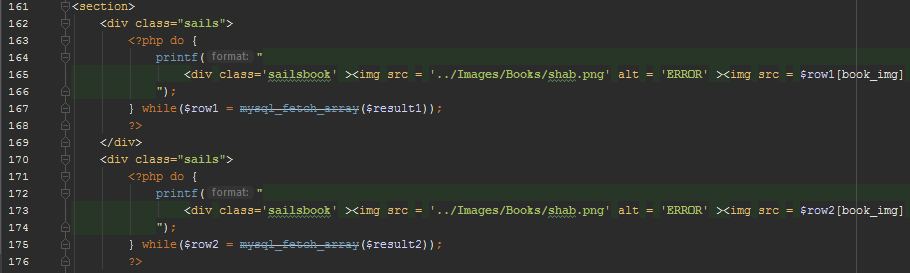
При формировании запроса важно указать таблицу, с которой будем работать, а так же поля этой таблицы. Вывод буду производить по столбцу «book\_discount» - численный процент скидки на книгу. Так же добавлю «DESC» - обратный порядок вывода ( от большего к меньшему ). Таким образом, на основную страницу будут попадать только книги с наиболее ощутимой скидкой. С помощью функции mysql\_query запрос будет обработан. Массив полученных данных разбивается на строки с помощью «mysql\_fetch\_array». Обрабатываем запрос с помощью функции «mysql\_query», после чего разбиваем массив полученных данных на строки при помощи «mysql\_fetch\_array».

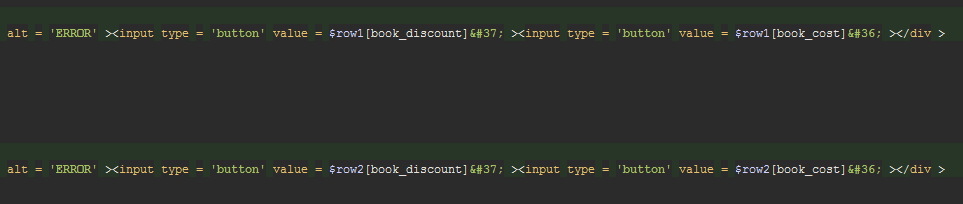


Создаю 4 таких запроса, результат каждого из которых заносится в собственную переменную, начинающуюся на «row».

Добавляю “LIMIT 4” в каждый запрос, чтобы выводить каждую из 4 переменных на в одну строку. Первые 2 переменные будут отвечать за видимую часть обложек, оставшиеся – за спойлер ( «LOAD MORE BOOKS» ). С каждой последующей переменной начальная строка переносится на 4 вперёд.

Основная секция скидок:





Для вывода использую функцию функция «printf». Достаточно создать один блок, все остальное сделает цикл php – do-while.

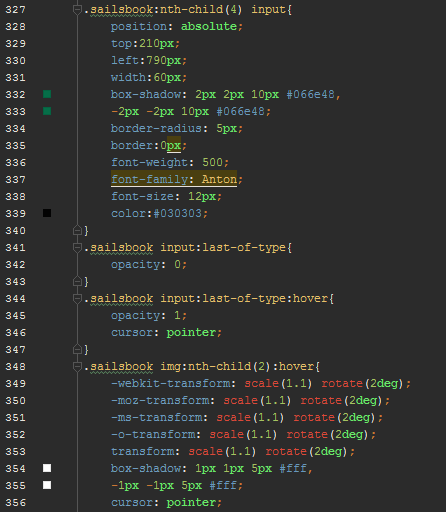
До тех пор, пока выводимые данные подпадают под переменную, указанную в while, блоки дублируются с переходом на следующую строку.

Таким образом, происходит вывод из БД:

1. Ссылка на картинку-обложку;
2. Численное значение процента скидки;
3. Стоимость книги.

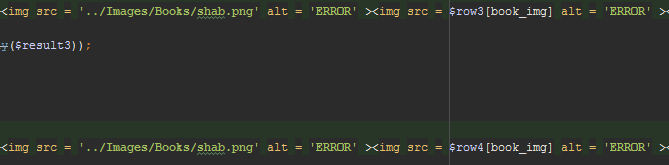
CSS-стили получились внушительных размеров, поэтому продемонстрирую только фрагмент кода:





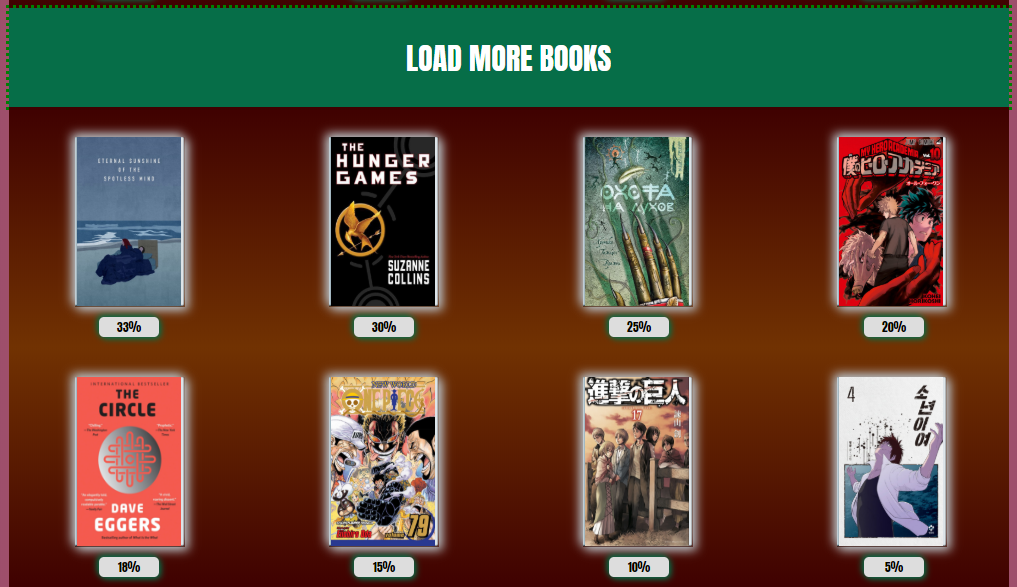
С помощью свойства opacity и селектора hover при наведении на input - процент скидки он перекрывается скрытым до этого момента input – стоимостью книги.

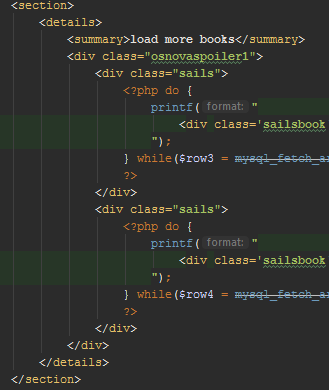
С помощью схожего принципа наложения я создаю достаточно приятные глазу пресеты для обложек:









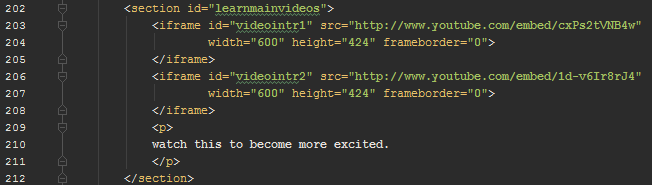
Таким образом, книги распределились по убыванию величины скидки.Вторая часть состоит из тех же самых элементов, заключённых в стилизованный спойлер: 

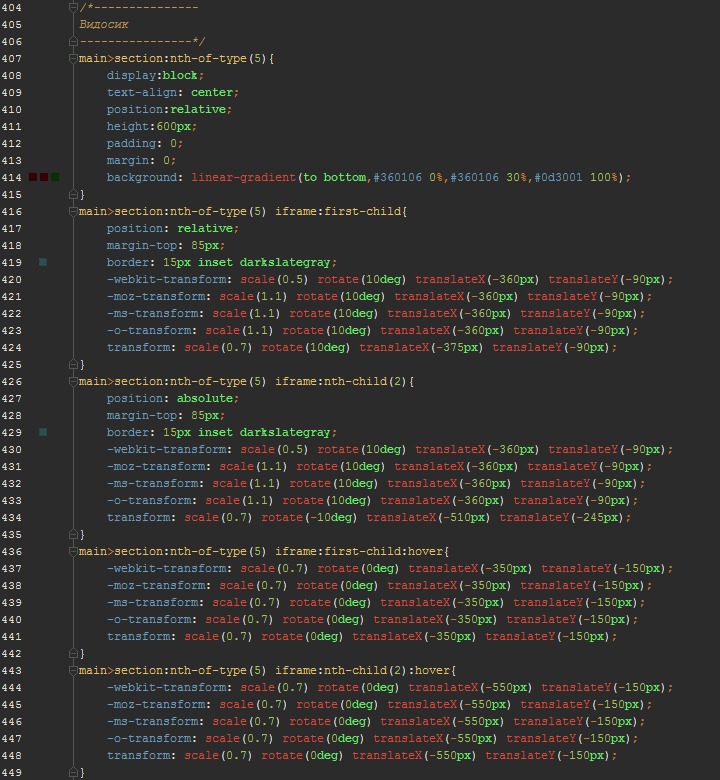


Не использованные строки таблицы пригодились бы при создании страницы, посвящённой отдельно выбранной книге.

## Раздел-мотивация для чтения

Как правило, все мы бываем ленивы для занятия даже тем, что нам кажется интересным. Чтобы помочь пользователям преодолеть свою лень, стоит добавить красочные короткие ролики о пользе чтения с вызывающей записью:

Html-код максимально прост. 2 iframe-а и параграф. Пора оформить их, как подобает:



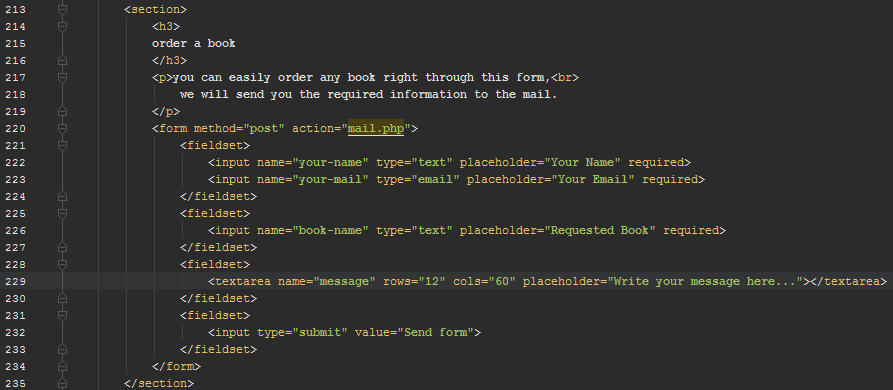
С помощью градиента создаётся задний фон, видео ограждаются «объёмной» рамочкой и поворачиваются по вертикальной оси. При наведении возвращаются в первоначальное положение.

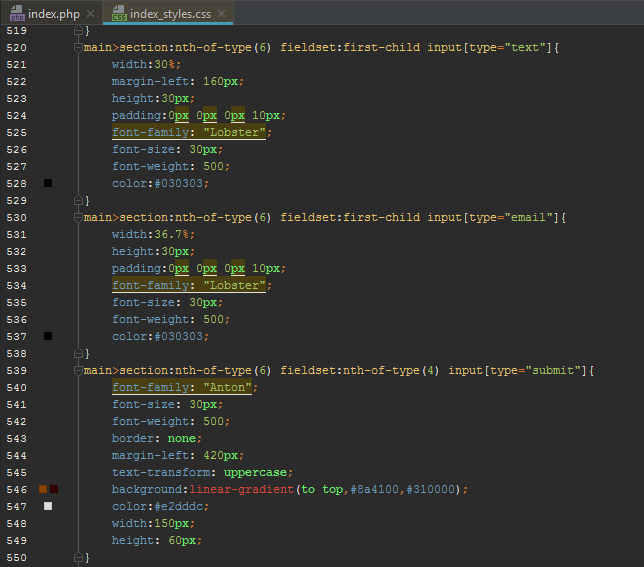




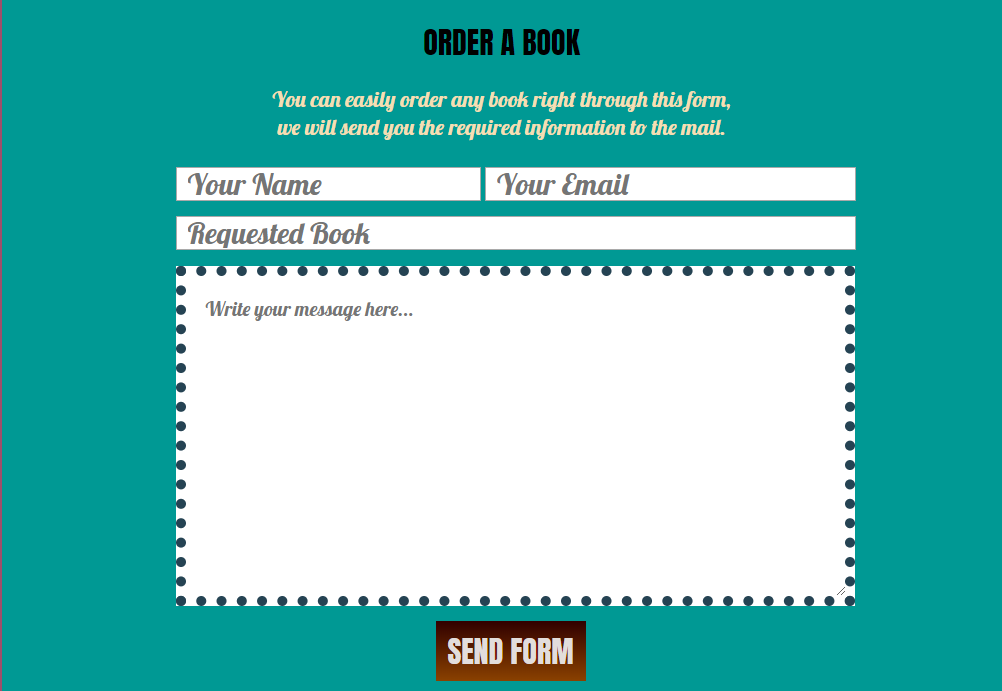
## Форма обратной связи

Для начала напишем html-код нашей формы и оформим её.





Placeholder создаются с помощью тега “placeholder“, оформляются с помощью css:



Для создания формы необходим тег form. У него есть 2 обязательных атрибута:

1. method – метод отправки запроса на сервер;
2. action – путь к файлу-обработчику формы.

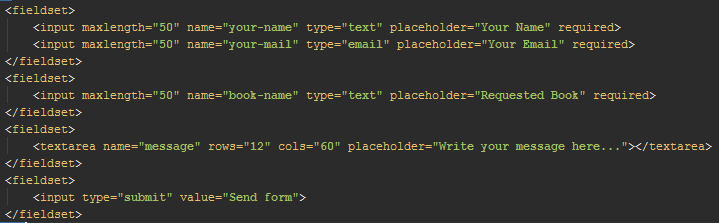
Именно обработчик формы будет отпарвлять введённое пользователем сообщение на почту. Изменяю путь к файлу:



Внутри форм располагаются fildset. Это нечто наподобие разделов формы, они неплохо помогают в оформлении формы стилями. Именно в эти поля пользователь вводит необходимую информацию.

С помощью атрибута name заданы имена конкретных полей. Именно по именам php работает с необходимой ему информацией.

Ограничу с помощью атрибута maxlenght кол-во допустимых для ввода символов в каждом инпуте:



Тег textarea так же предназначен для ввода пользователем информации, однако он может быть многострочным. Rows определяет размер поля в строках, cols – в символах. Однако можно так же указывать ширину-высоту с помощью css.

Поле-кнопка с тимпом «submit» отвечает за отправку формы.

Пора сделать форму работоспособной.

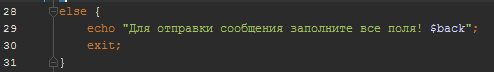
В месте, указанном в теге «form», создаю файл «mail.php».



Эта строка – ссылка на предыдущую страницу, сам адрес создаётся с помощью функции java script на случай изменения названия страницы с формой.

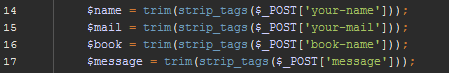
Далее – проверка на наполненность полей:

Если все указанные поля заполнены, то внутренняя часть скрипта начнёт обрабатываться. В ином случае пользователь увидит такое сообщение:



Я сделал проверку для всех полей формы, кроме сообщения, так как даже в самой форме сделал возможным оставление её пустой, это – дополнительный комментарий.

Чтобы обезопасить сообщение от вредоносного кода, очищаем вводимые пользователем данные от тегов и лишних пробелов:



После чистки тегов добавляем отправку сообщения:





Именно mail формирует отправку сообщения. Чтобы сообщение не было похоже на древний манускрипт, кодировку самого сообщения и страницы с ним стоит сделать одинаковой.

Аргумент «to» - почта, на котороую отправляется сообщение, «subject» - тема сообщения. Далее формируется текст сообщения. При этом используются заданные ранее значения переменных.

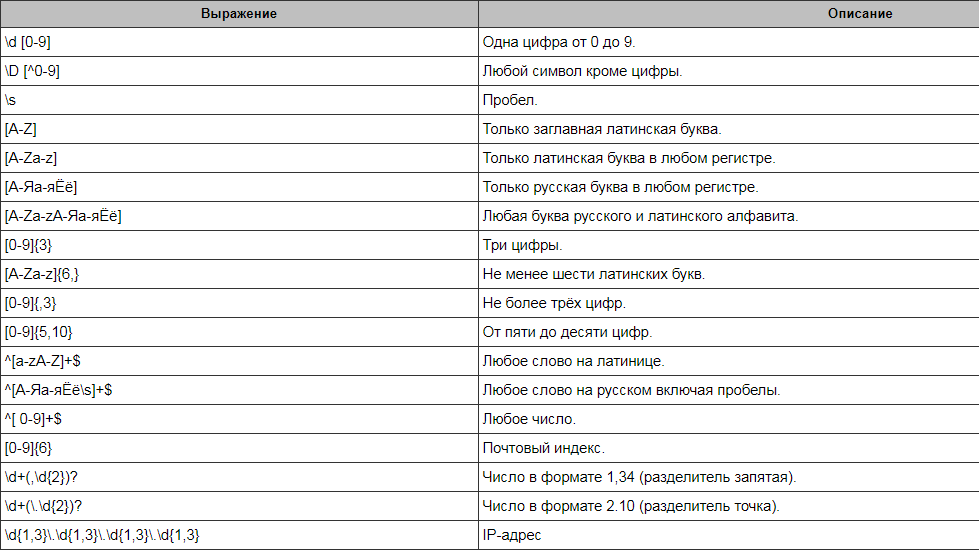
При условии успешного отправления сообщения выводится:

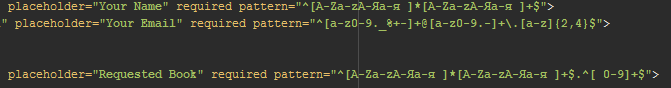


Теперь стоит добавить более подробную проверку вводимых пользователем данных, чтобы он мог узнать, что именно сделал неправильно. Сделаю это с помощью java script, так как это будет более удобно для пользователя.

В html-5 так же есть атрибут, отвечающий за проверку вводимых данных – pattern.

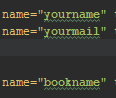
Вот его общий синтаксис:



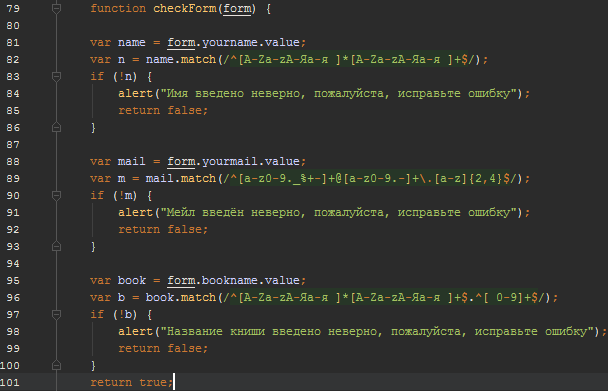


Однако стоит подстраховаться.

Для удобства уберу у всех имён пробелы:



В html.php файле создаю очередной script:



Чтобы при нажатии на кнопку формы происходил вызов функции, стоит привязать к ней onSubmit:



Далее, записывая в переменные выбранные по имени атрибута input, совершаем проверку на правильность введённых данных. Добавляется сообщение об ошибке.

Проверка работоспособности формы:

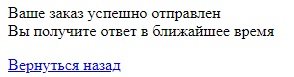


При отправке формы произошла ошибка с кодировкой, поэтому я всё же сменил кодировку на «windows-1251»:

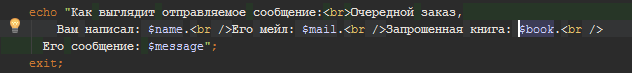


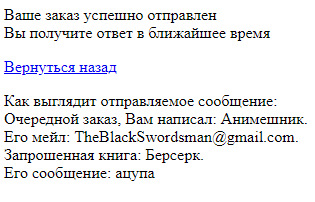


Сообщение отправлено успешно:



Можно еще сделать подобный вывод:





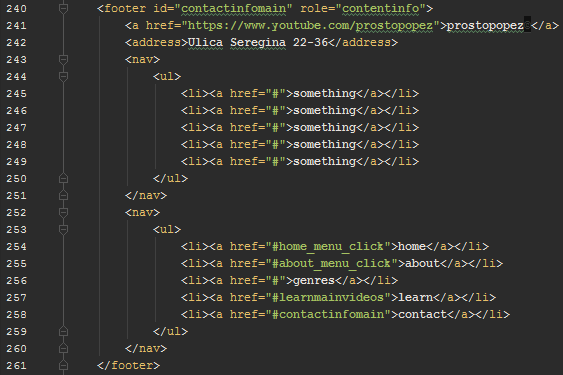
Однако такая конструкция будет срабатывать исключительно на полноценном хостинге.

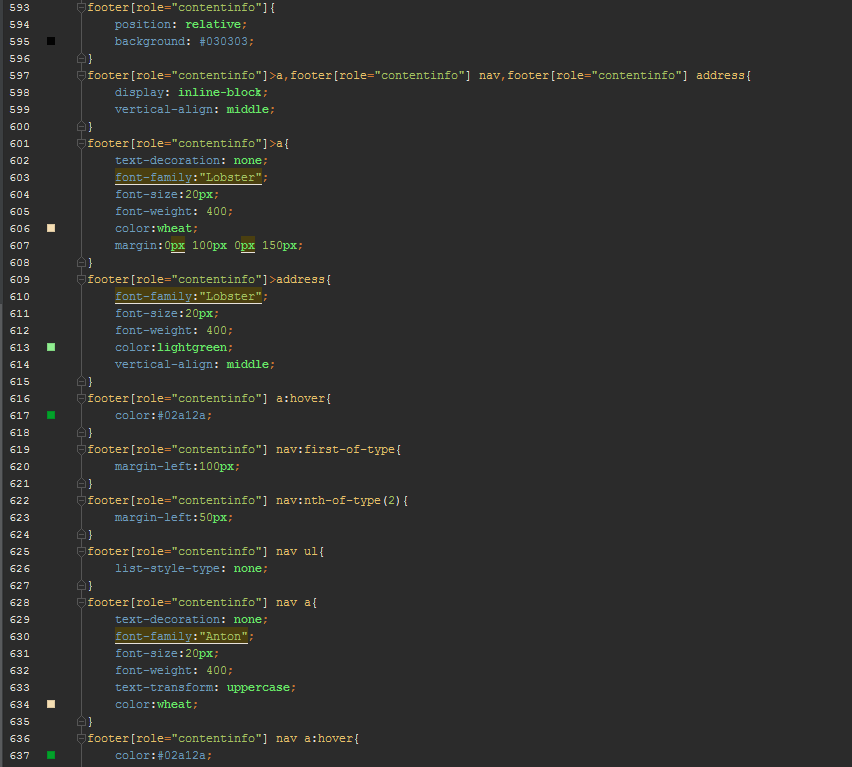
По необходимости локальный сервер можно связать с sendmail.

Основным средством рассылки почты в Internet является программа sendmail. Она обеспечивает работу модульной системы рассылки, которая предназначена для получения и отправки корреспонденции, а также управления программами подготовки и просмотра почтовых сообщений. Sendmail позволяет организовать почтовую службу локальной сети и обмениваться почтой с другими серверами почтовых служб через специальные шлюзы. Sendmail может быть сконфигурирована для работы с различными почтовыми протоколами.

## Footer ( подвал ) страницы

Footer очень похож на header. С той лишь разницей, что оформление footer будет отличаться, а так же там стоит указывать автора и контактные данные/адрес компании.







# Результат



# Вывод

Я создал с нуля главную страницу для сайта-магазина книг.

На странице присутствуют:

1. Header-меню;
2. Баннер с подключенным плагином для анимированной печати текста;
3. Блок со статистикой магазина;
4. Блок книг с активной скидкой, заполняющий с помощью php из базы данных локального сервера;
5. Блок с мотивационными видео;
6. Форма для связи, работающая с помощью отдельного php-файла;
7. Footer.

# Список использованной литературы

1. <https://toster.ru/>
2. <http://htmlbook.ru/>
3. <https://vaden-pro.ru/blog/php/forma-obratnoy-svyazi-php>
4. <https://jquery.com/>