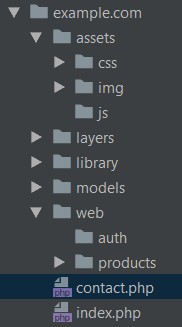
**Учебный сайт на php**

1. Найти Open Server на компьютере, если его нет, то скачать и установить.
2. Создать папку сайта в каталоге domains (например, example.com). Название папки, станет названием сайта.
3. Запустить Open Server, либо перезапустить, если он уже запущен.
4. Создать файл index.php в корне сайта. Нажать на зеленый флажок Open Sever. Открыть Мои сайты и выбрать созданную вами папку/сайт. В случае успеха на экране отобразиться пустая страница.
5. Сверстать каркас главной странички сайта вашего дипломного проекта, используя технологию flexbox и материалы статей:

* <https://html5.by/blog/flexbox/>
* <https://html5book.ru/css3-flexbox/>
* <http://dimox.name/press_footer_bottom_with_css/> - как с помощью CSS прижать footer к низу окна браузера

1. Создать структуру папок:



1. Файл со стилями разместить по адресу /assets/css/style.css и проверить правильность подключения файла в index.php
2. Вынести код шапки и код подвала сайта в отдельные файлы. Разместить файлы в каталоге layers. Присвоить названия файлам header.php и footer.php соответственно.
3. Создать файл core.php в каталоге library. Подключите этот файл в начале вашей страницы index.php
4. Создать функцию отладки dump($var). В качестве параметра она принимает значение некоторой переменной $var, а затем выводит ее в функции var\_dump($var) между тегами <pre>. Используйте ее при необходимости отладки ваших скриптов.
5. В header.php в теге <title> разместите вывод переменной $title.
6. Создать функцию layers($type, $title = ‘’). Первым параметром функция принимает строковый тип (либо header либо footer), а вторым параметром значение переменной $title – название страницы сайта. Используя include $\_SERVER[‘DOCUMENT\_ROOT’] обеспечьте подключение шапки и футера к страницам вашего сайта. Пример: layers(‘header’, ‘Главная страница’) и layers(‘footer’)
7. Создайте функцию url($path). Функция должна принимать относительный адрес от корня сайта, а возвращать абсолютный адрес.
8. Создайте страницу contact.php в корне вашего сайта (или любую другую страницу). Вставьте в нее код из файла index.php и замените title в функции layers на «Страница контактов». Убедитесь в корректной работе страницы. Допишите /contact.php к урл главной страницы вашего сайта в адресной строке браузера. <http://example.com/contact.php> либо если указан порт, то <http://example.com:82/contact.php>
9. В каталоге web создайте папки products и auth. В каталоге products создайте файл index.php и скопируйте туда код из contact.php. Измените title и убедитесь в корректной работе страницы в браузере. <http://example.com/web/products/> либо если указан порт, то <http://example.com:82/web/products/>
10. В шапке сайта создайте меню навигации из 3 гиперссылок на ранее созданные страницы (Главная страница, Страница контактов, Страница товаров). Убедитесь в правильной работе ссылок.
11. Создать файл Products.php и объявить в нем класс Products.
12. Создать метод getProducts(). В нем создайте массив объектов класса stdClass. Каждый объект – это отдельный товар со свойствами (id, name, model). Ключи массива должны начинаться с 1. Массив товаров должен находиться в публичном свойстве products. Метод должен возвращать этот массив через оператор return. (return $this->products)
13. В файле core.php создайте функцию getModel($model). В качестве параметра она должна принимать название модели. Далее она должна найти файл модели в папке models и загрузить его через оператор include и глобальный массив $\_SERVER. Затем создать экземпляр модели new $model() и вывести его через оператор return
14. В файле /web/products/index.php после подключения файла core.php пропишете $products = getModel(“Products”)->getProducts(); Используя функцию dump($products); убедитесь, что вы получаете массив товаров из метода getProducts().
15. Используя цикл foreach($products as $product) выведите содержимое массива товаров в теге <table> в файле /web/products/index.php
16. Добавьте стили к тегу <table> в фале со стилями /assets/css/style.css
17. Создайте файл /web/products/show.php и скопируйте туда содержимое файла /contact.php и измените содержимое title на «Страница товара»
18. Добавьте в класс Products.php метод getProduct(). В этом методе запустите метод $this->getProducts(). Затем проверьте глобальный массив $\_GET на наличие ключа id. Если он есть, то проверьте наличие этого ключа в массиве $this->products. Если в массиве с товарами есть товар с таким ключом, то мы выводим элемент массива через return $this->products[$\_GET[‘id’]]; Во всех остальных случаях нужно сформировать сообщение для пользователя, что страница не найдена. Используйте функцию dump() для отладки метода.
19. В файле /web/products/show.php запустите $product = getModel(‘Products’)->getProduct(); Затем выведите dump($product);
20. Убедитесь, что метод работает корректно. Для этого в браузере введите адрес <http://example.com/web/products/show.php?id=1>, либо <http://example.com:82/web/products/show.php?id=1>, если есть порт. Попробуйте поставить параметр id=2. Убедитесь, что метод возвращает товар с id = 2. Попробуйте запустить страницу без параметра id или введите id несуществующего товара. В этих случаях пользователю должно показываться сообщение о том, что страница не найдена
21. В файле /web/products/index.php добавьте ссылку на соответствующий товар в цикле. Т.е. при нажатии на id товара пользователь должен переходить на страницу с нужным ему товаром.