

# Создание собственной CMS

В этой статье мы рассмотрим создание собственной cms с инсталлятором, регистрацией пользователей и шаблонами сайта. При этом это будет общая cms-шаблон, для которой будет возможность создавать темы и плагины. Поэтому нам понадобиться время и все наше усердие. Предполагается, что модулями выступят: сайт блога, видеохостинг, буххостинг и небольшая социальная сеть с возможностью личных сообщений.

А начнем мы с самого простого, с видеохостинга, так как блог- это совсем просто. И что нам понадобиться? Правильно сервер с установленным ПО.

В моем случае:

- [Apache HTTP Server](#)
- база данных [MariaDB](#)

Позже сдаем тоже самое с файловой базой для домашнего использования. Основное рукописание в Notepad++ на:

- [PHP](#)
- [MySQL](#)
- [CSS](#)
- [JavaScript](#)
- [HTML](#)

## Подготовка к работе

### Рабочая станция

Мною будет использоваться Apache/2.4.58 (Unix) на [Synology NAS](#) с PHP: 8.2.21 и [MariaDB 10](#), но можно для простоты использовать локальный сервер, например [Denwer](#)

- Устанавливаем [Putty](#) - клиентскую программу для работы с сетевым протоколом SSH, SFTP и генерации цифровых SSH-ключей, которая является свободным приложением с открытым исходным кодом и распространяется под [Open Source](#) лицензией [MIT](#).
- Устанавливаем редактор [Notepad++](#) - бесплатный текстовой редактор исходного кода. Работая в среде [MS Windows](#), его использование регулируется [GNU General Public License](#).

### Планирование

Рассмотрим примерную структуру нашего приложения (названия директорий произвольное, но проще использовать понятные или традиционные, чтобы не запутаться, например: user - в переводе «пользователь», install - «установка», readme - «прочти меня», user\_login - «пользователь\_авторизация» и т.д.). Ниже приведен предварительной проект директорий нашей будущей CMS.

```
«./ucms/»
  |
  +-- users
```

```
admin
  └── account_admin
      └── admin_setting.php
  └── admin_nav.php
  └── admin.php

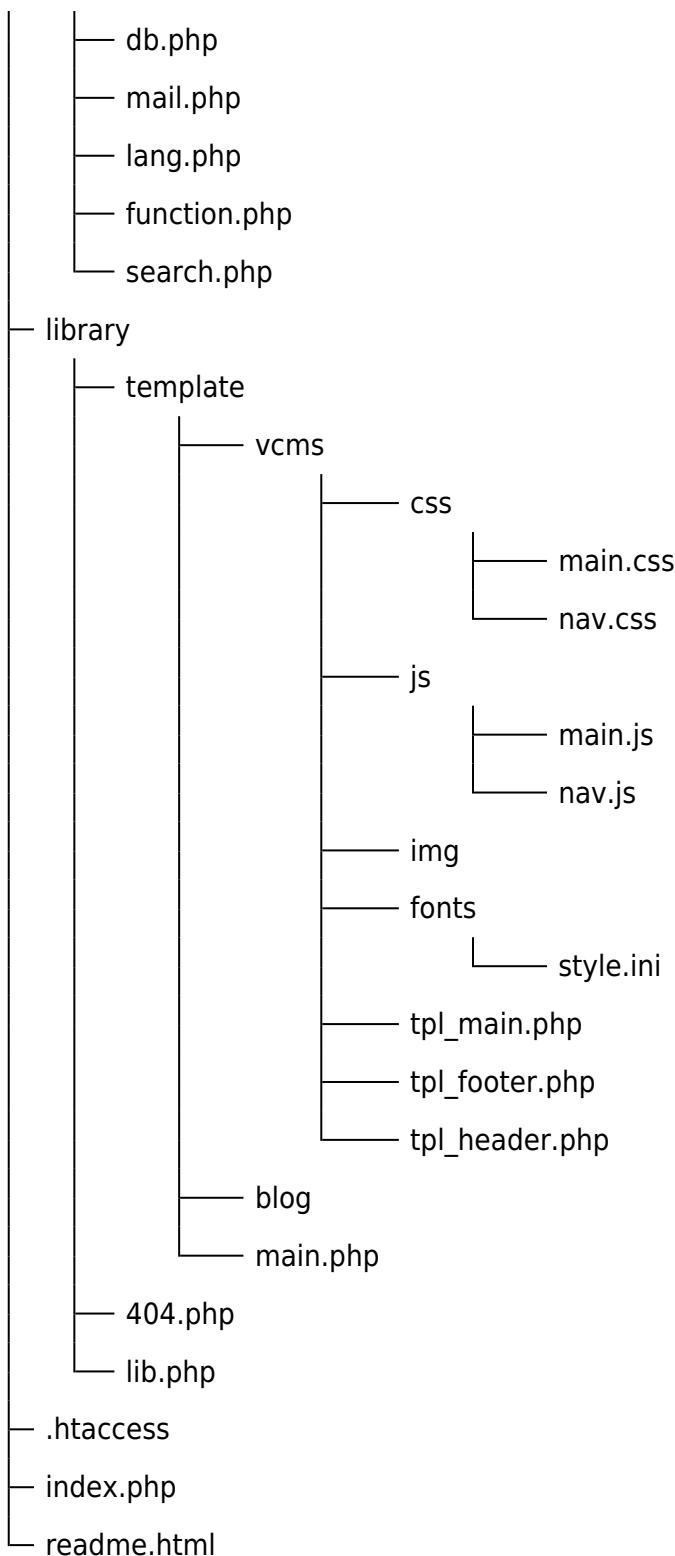
user
  └── account_user
      ├── user_login.php
      ├── user_logout.php
      └── user_signup.php
  └── messages
  └── user_edit.php
  └── user_nav.php
  └── user.php

install
  ├── account_settings.php
  ├── database_options.php
  ├── installation_progress.php
  ├── options.php
  ├── setting.php
  └── install.php

ghost.php
administrator.php

content
  └── data
      ├── books
      ├── images
      ├── media
      ├── music
      └── media
  └── cont.php

includes
  └── config.php
```



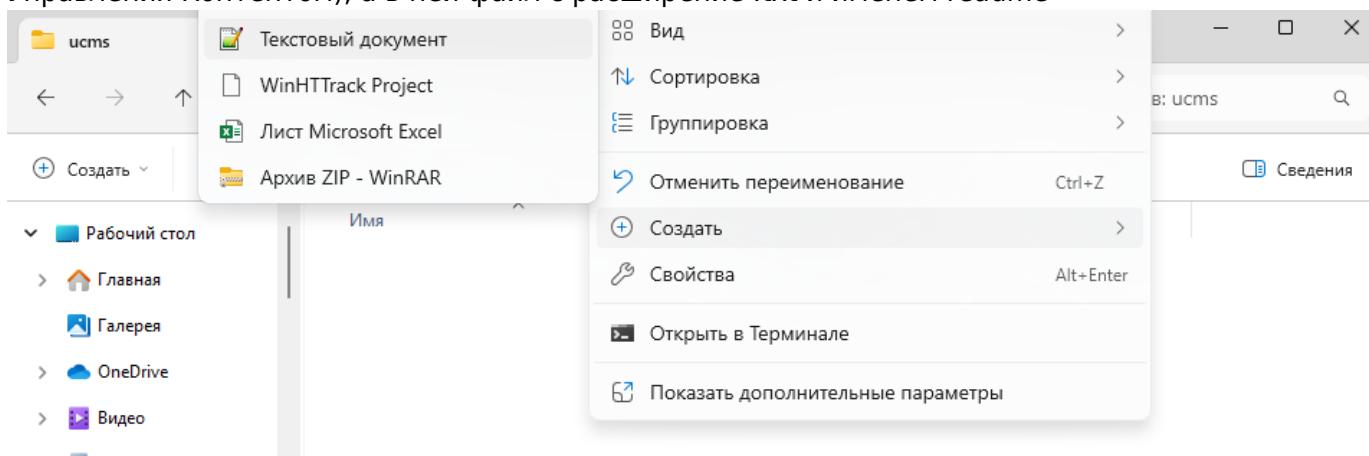
Надеюсь, что из вышеприведенной схемы директорий все понятно? Кратко:

- .htaccess — это локальный конфигурационный файл веб-сервера Apache, который позволяет управлять настройками сайта.
- index.php - это стартовая страница.
- readme.html - руководство, которое дает пользователям подробное описание проекта
- user - главная папка пользователей
- guest - пользователь с правом только просмотра и поиска по сайту контента.
- account\_user - папка аккаунта пользователя (пользователь с правом просмотра, загрузки, редактирования и удаления своего контента, поиска контента, написания комментариев,

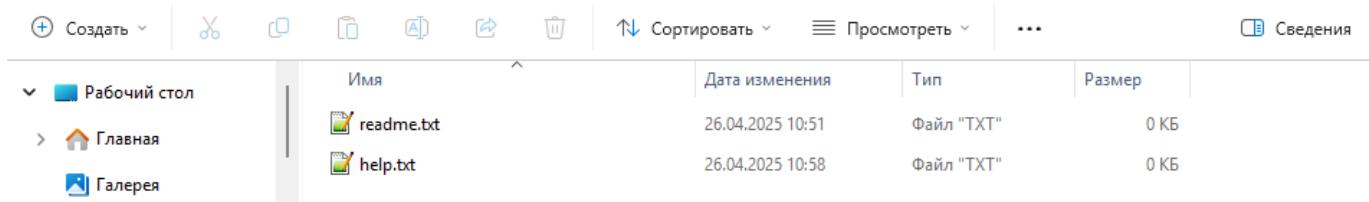
голосования и прочего.

- user\_login.php - файл авторизации пользователя
- user\_logout.php - файл окончания сеанса авторизации пользователя
- user\_signup.php - файл регистрации нового пользователя
- user\_edit.php - редактор постов пользователя
- user\_nav.php - меню пользователя
- account\_admin - папка аккаунта администратора (полные права, включая настройку сайта и регистрацию прав пользователя/удаление пользователей и модерацию сообщений)
  - admin\_setting.php - файл настроек CMS администратором
  - admin\_nav.php - расширенное меню пользователя-администратора
- install - папка установки CMS

и т.д... (см. описание в файле readme.html) с создание которого мы и начнем. На нашем вебсервере создаем директорию с понятным названием, к примеру... и так уже обозначено в нашем проекте «ucms» - «Universal Content management system» (Универсальная Система Управления Контентом), а в ней файл с расширение .txt и именем readme



Этот файл не будет участвовать в работе приложения, в этом файле мы будем писать подробное руководство для пользователя проекта. Да, мы будем много писать, и в основном это будут комментарии. Поэтому создаем еще один файл с именем help.txt



этот файл в конце нашей работы мы удалим, в нем будут записываться используемые части кода, чтобы вы всегда могли посмотреть как и что пишется, чтобы не тратить время на перелистывание страниц в интернете. Так-же там будут наши заметки, т.к. за один день мы не успеем охватить весь процесс.

Пример файла help.txt

### help.txt

```
!Это файл - подсказка по написанию кода в руководстве по написанию UCMS
-----
<!-- Комментарии html -->
// Комментарии в PHP или /* ... */
/* Комментарии CSS */
// Комментарии в JavaScript или /* ... */
```

```
/* Комментарии MySQL */

-----  
<?php  
    // Вставка кода PHP в HTML-файл  
?  
-----  
Подключение файлов JavaScript к HTML  
<script src="js/main.js"></script>  
-----  
Подключение файлов CSS к HTML  
<link rel="stylesheet" href="/style.css">  
/////////////////Важно!///////////////////////////  
в файле login.php переменную password заменили на login_pass  
не забыть !!! произвести замену в reg.php  
/////////////////////////////26.04.2025///////////////////////////  
закончили на подключении файла стилей CSS в index.php в директории  
./ucms/bin/user
```

Казалось бы, этого можно не делать, но поверьте, когда у вас будет много директорий и много файлов, вы начнете путаться, а если вас еще отвлекут от процесса на неделю, то вы вообще все забудите, а так у вас будет все записано и вам не придется начинать все с начала.

← [Оглавление](#) [Создание установщика CMS](#) →

← [Оглавление](#) [Файл readme.txt](#) →

From:  
<https://wwoss.ru/> - **worldwide open-source software**



Permanent link:

[https://wwoss.ru/doku.php?id=software:development:demo:cms:video\\_cms](https://wwoss.ru/doku.php?id=software:development:demo:cms:video_cms)

Last update: **2025/04/26 12:01**