

Организация онлайн галереи

Общие технические вопросы

Содержание

- **Теория**

- Доменное имя
- Платформа: обычный хостинг, VPS/VDS, облачный хостинг
- Контент: готовые CMS и собственные решения, генераторы статичных сайтов

- **Практика**

- Привязка доменного имени, Web сервер, SSL-сертификаты
- Управление контентом, контроль версий (Git)
- Дизайн и адаптация к различным устройствам
- Онлайн галерея как PWA приложение (iOS, Android)

- **Заключение**

- **Готовый пример**

Теория

Доменное имя

Уникальный адрес сайта в интернете (например, mygallery.com).

- Имя должно быть коротким, запоминающимся и отражать суть галереи (по возможности).
- Проверить доступность домена можно на сервисах регистраторов (Namecheap, Reg.ru).
- Рассмотрите тематические доменные зоны: **.art**, **.gallery**, **.online** (mygallery.art).
- Настройте DNS (Name Servers) для хостинга.

Платформа

- **Виртуальный хостинг (Shared Hosting):**
 - Подходит для небольших проектов.
 - **Преимущества:** низкая стоимость, простота настройки.
 - **Недостатки:** ограниченные ресурсы, низкая гибкость.
- **VPS (Virtual Private Server) / VDS (Virtual Dedicated Server):**
 - Виртуальный сервер с выделенными ресурсами.
 - **Преимущества:** гибкость, root-доступ, возможность установки нужного ПО.
 - **Недостатки:** требует базовых навыков администрирования (GNU/Linux).

Платформа

- **Облачные решения (AWS, Google Cloud, Netlify):**
 - **Преимущества:** масштабируемость, высокая надежность.
 - **Недостатки:** сложность входа/управления, стоимость.
- **Самостоятельный сервер:**
 - Полный контроль, но требует глубоких технических знаний и вложений.
- **Отдельное упоминание: Tilda**

Платформа

Критерии выбора платформы:

- Бюджет.
- Требования к производительности.
- Уровень технической подготовки.

Рекомендация для галереи:

- Для небольших галерей подойдёт VPS.
- При росте проекта можно перейти на облачные решения.
- При нежелании разбираться с VPS можно использовать платформы типа Tilda.

Платформа

На что обратить внимание:

- **Надежность:** высокий uptime (99,9% и выше) и качественная техническая поддержка.
- **Производительность:** достаточные ресурсы (CPU, RAM, дисковое пространство) для вашего проекта.
- **Стоимость:** соотношение цены и предоставляемых услуг, включая **скрытые платежи**.

Контент

Готовые решения:

- *WordPress*: простота использования, множество плагинов
- *Drupal/Joomla*: более гибкие, но сложные в освоении.
- *Shopify/Squarespace*: подходят для галерей с функцией продажи.
- **Плюсы**: простота использования, наличие готовых плагинов и шаблонов.
- **Минусы**: «шаблонные» и «архитектурные» ограничения, нагрузка на сервер, уязвимости, более сложная настройка сервера (не касается сервисов типа Tilda).

Контент

«Самописные» решения:

- **Плюсы:**
 - **Полный контроль:** уникальная функциональность (не функционал!), точно соответствующий задачам проекта.
 - **Гибкость:** легкость внесения изменений и адаптации под новые требования.
- **Минусы:**
 - **Сложность:** необходимы квалифицированные разработчики и тестировщики.
 - **Риски:** ошибки в архитектуре или коде могут привести к проблемам в будущем.

Контент: статические сайты

Быстрые, простые, безопасные, но ограничены в функциональности и относительно сложны в обновлении.

Идеальны для лёгких проектов.

Подходят, если:

- У вас есть **технические навыки** (или доступ к разработчику).
- Вы готовы использовать **генераторы статических сайтов** (Hugo, Jekyll и т.д.).
- Вам **не нужна сложная интерактивность** (фильтры, поиск, комментарии).

Контент: статические сайты

Как это работает:

- Вы добавляете новые картины и информацию о выставке в формате **Markdown** и/или **JSON**.
- Генератор статических сайта (например, Hugo) автоматически создает HTML-страницы на основе шаблонов.
- Обновленный сайт загружается на хостинг (GitHub Pages, Netlify, облако, VPS).

Контент: статические сайты

Преимущества:

- **Высокая скорость загрузки:** статические сайты работают быстрее.
- **Низкая стоимость:** хостинг для статических сайтов дешевле (или даже бесплатный, например, GitHub Pages).
- **Безопасность:** нет серверного кода, что снижает риски взлома.

Недостатки:

- **Ручное обновление:** требуется знание Markdown или помощь разработчика.
- **Ограниченная интерактивность:** сложно реализовать динамические функции (например, фильтры или поиск).

Контент: динамичные сайты

Гибкие, функциональные, легко обновляемые, но требуют больше ресурсов и сложнее в защите.

Лучше для сложных, интерактивных и часто обновляемых проектов.

Подходят, если:

- Вы хотите **легко управлять контентом** без технических навыков.
- Вам **нужна интерактивность** (например, фильтры по жанру, автору, году).

Контент: динамичные сайты

Как это работает:

- Использование **CMS (системы управления контентом)**, например, WordPress, Drupal.
- Новые картины и выставки добавляются через удобный интерфейс CMS.
- Контент хранится в базе данных и динамически загружается на сайт.

Контент: динамичные сайты

Преимущества:

- **Удобство управления:** добавление новых картин и выставок через интерфейс CMS.
- **Интерактивность:** возможность реализации фильтров, поиска, комментариев.
- **Масштабируемость:** подходит для больших проектов с постоянно растущим контентом.

Недостатки:

- **Сложность настройки:** требуется больше технических знаний для настройки CMS (и сервера).
- **Высокая стоимость:** хостинг для динамичных сайтов дороже.
- **Меньшая скорость:** динамичные сайты загружаются медленнее, чем статичные.

Контент

Рекомендация для галереи:

- При редких обновлениях экспозиций подойдут статичные сайты.
- Если вам нужно удобство управления контентом и интерактивность, выбирайте динамичный сайт с CMS.

Следует упомянуть: гибридный подход (JAMstack)

<https://en.wikipedia.org/wiki/JAMstack>

Практика

Привязка доменного имени

DNS-серверы

Бесплатные DNS-серверы Рег.ру

ns1.reg.ru

ns2.reg.ru

 Изменить

Ресурсные записи

 Очистить зону

[+](#) Добавить запись

A @ → 111.111.111.111



A testing → 111.111.111.111



A www → 111.111.111.111



https://ru.wikipedia.org/wiki/Типы_ресурсных_записей_DNS

Web сервер: для собственного VPS

```
sudo dnf install nginx -y  
sudo firewall-cmd --permanent --add-service=http  
sudo firewall-cmd --permanent --add-service=https  
sudo firewall-cmd --reload
```

```
sudo nano /etc/nginx/conf.d/gallery.conf
```

```
sudo systemctl restart nginx
```



```
server {  
    listen 80;  
    server_name mygallery.art www.mygallery.art;  
  
    root /var/www/mygallery.art;  
    index index.html;  
  
    location / {  
        try_files $uri $uri/ =404;  
    }  
}
```

Nginx — это высокопроизводительный веб-сервер. Он используется для обработки HTTP-запросов, обслуживания статических файлов, балансировки нагрузки и многого другого.

SSL сертификаты: для собственного VPS

```
sudo dnf install certbot python3-certbot-nginx -y  
sudo certbot --nginx -d mygallery.art -d www.mygallery.art
```

После завершения **Certbot** автоматически настроит **Nginx** для работы с **HTTPS**.

Сайт будет доступен:

- По **HTTP**: <http://mygallery.art>
- По **HTTPS**: <https://mygallery.art>

Зачем нужен SSL?

- Шифрование данных.
- Повышение доверия пользователей.
- Влияние на **SEO**.

Ссылки на полные инструкции:

<https://wiki.almalinux.org/series/nginx/NginxSeriesA02R92.html>

<https://www.digitalocean.com/community/tutorials/how-to-install-nginx-on-rocky-linux-9>

Управление контентом, контроль версий

Git — это распределенная система управления версиями. Она позволяет отслеживать изменения в файлах и проектах, а также работать над ними в команде.

Что нам интересно в **Git**:

- Сохранение истории изменений (коммиты).
- Откат к предыдущим версиям.

Gitea — это легковесная система для управления репозиториями Git:

- Простота установки и настройки.
- Низкие требования к ресурсам.

Актуально для статичных сайтов!

Управление контентом, контроль версий

Как это работает:

- Внесение изменений в локальный репозиторий (хранилище) сайта у вас на компьютере.
- Отправка изменений на сервер.
- Сервер компилирует страницы и обновляет сайт.
- Каждое изменение сайта хранится на сервере, он позволит «откатиться» к любому предыдущему состоянию.

По такому же принципу работают сервисы типа **Netlify**.

Актуально для статичных сайтов!

Дизайн

- **Основные принципы:**

- Минимализм и **акцент на контенте**.
- Чёткая структура, лёгкая навигация.

- **Адаптивность:**

- Использование CSS-фреймворков (например, Bootstrap).
- Медиа-запросы для адаптации под разные устройства (@media (pointer: coarse) и др.).

- **Тестирование**

- <https://pagespeed.web.dev/> <https://www.bing.com/webmaster/tools/mobile-friendliness>

Пример отчёта

Мобильное устройство

PageSpeed Insights

[Скопировать ссылку](#) [Документация](#)

Анализировать

Мобильные устройства Компьютер

Определяйте фактическую производительность сайта

Нет данных

Выявляйте проблемы с производительностью

100

Производительность

100

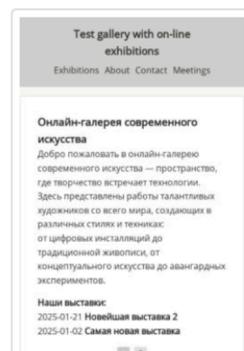
Специальные возможности

100

Рекомендации

100

Поисковая оптимизация



Компьютер

PageSpeed Insights

[Скопировать ссылку](#) [Документация](#)

Анализировать

Мобильные устройства Компьютер

Определяйте фактическую производительность сайта

Нет данных

Выявляйте проблемы с производительностью

100

Производительность

100

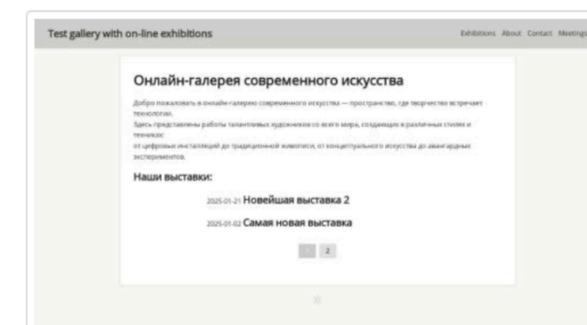
Специальные возможности

100

Рекомендации

100

Поисковая оптимизация



Использование сайта как WebApp

Progressive Web Apps (PWA) работают в браузере, но при этом предоставляют пользователю опыт близкий к нативным приложениям:

- PWA корректно отображаются на любых устройствах: от смартфонов до десктопов.
- Благодаря использованию **Service Workers**, PWA могут работать оффлайн или при слабом интернет-соединении.
- PWA можно добавить на домашний экран устройства (как нативное приложение) **без необходимости загрузки из магазина приложений** (App Store или Google Play).
- PWA загружаются мгновенно благодаря кэшированию ресурсов.
- PWA работают только через **HTTPS**, что обеспечивает безопасность данных пользователей.

Использование сайта как WebApp

Как это работает:

- **Manifest** — это JSON-файл, который содержит метаданные о приложении: название, иконки, цвет темы, стартовый URL.

Эти данные используются для добавления PWA на домашний экран устройства.

- **Service Workers** — это скрипты, которые работают в фоновом режиме и позволяют кэшировать ресурсы (HTML, CSS, JS, изображения) для быстрой загрузки, обеспечивать работу приложения без интернета, перехватывать сетевые запросы и управлять ими.

Использование сайта как WebApp

manifest.json

```
{
  "name": "АРТ Галерея»,
  "short_name": "ArtGallery",
  "icons": [
    {
      "src": "favicon_192.png",
      "sizes": "192x192",
      "type": "image/png"
    }
  ],
  "start_url": "/",
  "display": "standalone",
  "background_color": "#ffffff",
  "theme_color": "#f0f0f0"
}
```

sw.js

```
self.addEventListener('install', event => {
  event.waitUntil(
    caches.open('gallery-cache').then(cache => {
      return cache.addAll(['/ ', '/index.html', '/styles.css']);
    })
  );
});
...

```

Заключение

Заключение

- **Выбор платформы** зависит от бюджета, требований к производительности и уровня технической подготовки. VPS подходит для небольших или средних галерей, облачные решения – для крупных проектов.
- Безопасность является обязательным условием: **настройка SSL-сертификатов** через Let's Encrypt повышает доверие пользователей и улучшает SEO.
- Контент можно реализовать через **готовые CMS** или с помощью **самописных решений**. **Статичные** сайты подходят для редких обновлений, а **динамичные** – для интерактивных проектов.
- **Дизайн** должен быть минималистичным, **с акцентом на контент**, и адаптивным для различных устройств.
- **PWA** позволяет сделать сайт похожим на нативное приложение, что повышает вовлеченность пользователей и улучшает их опыт.

Готовый пример

А как у меня?

«Тестовый полигон»: <https://testing.vkireyeu.art/> (или <https://shelter.vkireyeu.art/>)

- Виртуальный выделенный сервер (VPS).
- Статичный сайт.
- Самописное решение (генератор статических страниц).
- Сайт работает как PWA приложение (может устанавливаться в iOS и Android).
- Сайт адаптирован под разные экраны.
- Автоматическое использование системной темы (светлая/тёмная).
- Gitea для организации обновления и хранения истории сайта.

Как это работает?

Структура рабочей директории генератора статических страниц

— content/	— директория для контента
— css/	— директория для таблиц стилей
— js/	— директория для JavaScript макросов
— shortcodes/	— директория для "шорткодов" («коротких кодов»)
— static/	— директория для различных статичных файлов
— templates/	— директория для шаблонов страниц
— config.yaml	— файл с основными настройками сайта
— generator.py	— код генератора (Python)

Настройки

config.yaml

- site_name: название сайта.
- site_url: адрес сайта.
- template: шаблон для страниц сайта.
- favicon: иконка сайта.

Важные блоки:

- dirs: определяет директории для контента, шаблонов, «шорткодов» и т.д..
- navigation: определяет пункты «меню».
- pagination: количество пунктов на страницу в списке «выставок».

```
site_name: Test gallery with on-line exhibitions
site_url: https://testing.vkireyeu.art
template: base.html
favicon: "favicon_192.png"
dirs:
  content: "content"
  templates: "templates"
  shortcodes: "shortcodes"
  css: "css"
  js: "js"
  static: "static"
  output: "output"
  galleries: "galleries"
markdown:
  breaks: true
  html: True
  enable_table: true
server:
  host: 127.0.0.1
  port: 8000
navigation:
  - name: Exhibitions
    url: "/pages/exhibitions/"
  - name: About
    url: "/pages/about/"
  - name: Contact
    url: "/pages/contacts/"
  - name: Meetings
    url: "/pages/meetings/"
pagination:
  galleries_per_page: 2
```

Контент

Обычные страницы (About, Contact, Meetings):

- Для каждой страницы в директории **content/pages** создаётся поддиректория с файлом `index.md`, например, `content/pages/about/index.md`.
- Файл `index.md` наполняется содержимым в формате **Markdown**.
- Генератор `generator.py` преобразует файл `index.md` в **HTML файл** используя шаблоны из каталога `templates`.
- Путь поддиректория, например, `content/pages/about` сохраняется в выходной директории с сайтом: `output/pages/about`.
- Если в поддиректории сожержатся другие файлы, необходимые для страницы, они также будут скопированы в новую выходную директорию.

Контент

Страницы галерей:

- Для каждой галереи в директории **content/galleries** создаётся поддиректория коротким названием «выставки», например, *content/galleries/my_new_exhibition*
- В новой директории обязательно должны быть:
 - Фотографии картин.
 - Директория **thumbnails** с миниатюрами картин.
 - JSON файл **gallery.json** с описанием выставки и картин.
- Генератор создаёт выходную поддиректорию *output/galleries/my_new_exhibition*, переносит в неё фотографии картин, директорию с их миниатюрами, файл с описанием, формирует файл `index.html` на основе шаблонов.

Контент

Описание «выставки» (gallery.json)

`galleryName` — название выставки, используется в т.ч. как текст ссылки на «выставку» в списке всех «выставок».

`galleryDescription` — описание «выставки», отображается на странице выставки.

`date` — дата «выставки», отображается в списке «выставок» и используется для сортировки списка (самые новые вверху).

```
{
  "galleryName": "Самая новая выставка",
  "galleryDescription": "Описание новой выставки",
  "date": "2025-01-02",
  "images": [
    {
      "src": "image-01.jpg",
      "title": "Название 1",
      "author": "Автор 1",
      "canvasType": "Холст, масло",
      "year": "2024",
      "size": "50 x 70 см",
      "price": "100 000 руб.",
      "contact": "+7 (000) 000-00-00",
      "fullDescription": "Подробное описание картины."
    },
    {
      "src": «image-02.jpg»,
      "title": "Название 2",
      "author": "Автор 2»,
      "canvasType": "Холст, масло",
      "year": "2024",
      "size": "50 x 80 см",
      "price": "80 000 руб.",
      "contact": "+7 (000) 000-00-00",
      "fullDescription": "Описание картины"
    }
  ]
}
```

Контент

Главная страница:

- Формируется из *Markdown* файла ***content/index.md***.
- Список галерей можно отобразить на главной странице используя «шорткод» **[galleries]**.
- Генератор формирует главную страницу с использованием шаблонов из директории `templates`.

```
index.md x
1 # Онлайн-галерея современного искусства
2
3
4 Добро пожаловать в онлайн-галерею современного искусства – пространство, где творчество встречается технологии.
5 Здесь представлены работы талантливых художников со всего мира, создающих в различных стилях и техниках:
6 от цифровых инсталляций до традиционной живописи, от концептуального искусства до авангардных экспериментов.
7
8 ## Наши выставки:
9 [galleries]
10 |
```

Работа генератора

- Генератор создаёт директорию **output/** для выходных файлов, в этой директории будет создана структура сайта.
- Генератор переносит в **output/** директории с таблицами стилей, JavaScript макросами (css, js соответственно) и файлы из директории **static/**.
- Также генератор переносит в выходную директорию контент: обычные страницы и страницы галерей с сохранением структуры поддиректория (см. слайды 35, 36).
- Таким образом, в выходной директории **output/** находится **готовый статический сайт** со всем необходимым набором файлов.
- Генератор можно запустить в интерактивном режиме: `./generator.py --serve`

Обновление сайта

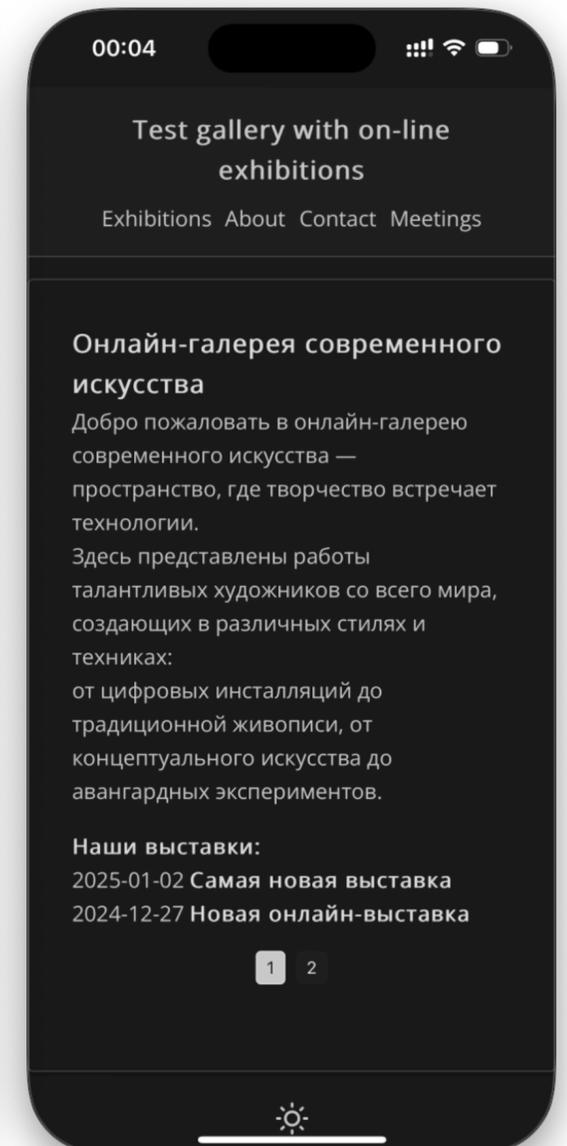
- Создаётся новая «выставка» (см. слайд 36) или другие изменения.
- Запускается генератор `generator.py` в интерактивном режиме для проверки работоспособности сайта.
- Если всё ОК — локальные изменения отправляются в `git`-репозиторий в Gitea:
 - `git add .`
 - `git commit -m "Update"`
 - `git push origin master`
- Gitea на VPS сервере запускает генератор, «компилирует» сайт и копирует в директории, настроенные для сайта в конфигурации `web`-сервера `nginx`.
- Сайт обновлён.

Progressive Web App

Для <https://testing.vkireyeu.art/> в сформированы файлы manifest.json, load-sw.js и sw.js, необходимые для работы «галереи» как PWA приложения для смартфонов на базе iOS или Android.

Во время «установки» приложения оно скачивает сайт на устройство пользователя, и, таким образом, после установки может работать без подключения к Интернет.

Приложение, как и сайт, «подстраивается» под системную тему устройства пользователя, если пользователь явным образом не выбрал «светлую» или «тёмную» тему для приложения.



Спасибо за внимание!