структуру каталогов

структуру каталогов

```
microwiki/
├── assets/                    # СИСТЕМНЫЕ АССЕТЫ ЯДРА (Неизменяемая логика)
│   └── js/
│       ├── media-modal.js     # AJAX-загрузка картинок из модального окна в
редакторе
│       └── toolbar.js         # Хоткеи (Ctrl+B/I/U) и асинхронный движок
предпросмотра (Preview)
├── conf/                      # Папка конфигурации движка
│   ├── local.php              # Основные настройки сайта (название, язык)
│   └── users.auth.php         # Плоская база пользователей и хэшей паролей
├── data/                      # Данные (в DokuWiki это папка data/)
│   ├── pages/                 # Текстовые страницы (.txt)
│   ├── media/                 # Каталог картинок и файлов (.png, .jpg,
.pdf)
│   ├── attic/                 # Старые ревизии страниц
│   └── meta/                  # Метаданные и кеш
├── lib/                       # Системные библиотеки и плагины
│   └── tpl/                   # Папка с шаблонами оформления
│       └── default/           # Текущая тема оформления
│           ├── main.php       # Основной каркас
│           ├── header.php     # Верхняя часть и меню
│           ├── footer.php     # Нижняя часть и подпись
│           └── css/
│               └── style.css # ВСЕ стили сайта, монолитного тулбара и
синтаксического кода
├── src/                       # Ядро движка (Core)
│   ├── Action/                # Контроллеры (show, edit, media)
│   ├── Auth.php               # Модуль авторизации, сессий и верификации
паролей
│   ├── CodeHighlighter.php    # Универсальный динамический парсер раскраски и
нумерации кода
│   ├── MediaManager.php       # Файловый менеджер (загрузка, удаление,
просмотр)
│   ├── Parser.php             # Вики-парсер
│   ├── Storage.php            # Работа со страницами и ID (разрешение систем
имен `:`)
│   ├── Template.php           # Диспетчер вывода контента (страницы, формы
правки, медиаменеджер)
│   └── toolbar.php            # HTML-структура кнопок монолитного тулбара и
каркас модалки
└── index.php                  # Единая точка входа
```

local.php

local.php

```php
<?php
/**
 * MicroWiki Configuration File
 * Глобальные настройки движка
 */
if (!defined('WIKI_ROOT')) die();

$conf['title']     = 'Наша Учебная Вики';  // Название вашей базы знаний
$conf['lang']      = 'ru';                 // Язык интерфейса
$conf['template'] = 'default';             // Используемый шаблон

// Настройки безопасности и пользователей
$conf['useacl']    = true;                 // Включить разграничение прав доступа
$conf['superuser'] = 'admin';              // Имя главного администратора
```

main.php

main.php

```php
<?php
if (!defined('WIKI_ROOT')) {
    die('Прямой доступ к файлу запрещен.');
}

// Поочередно собираем страницу из изолированных компонентов
tpl_header();
tpl_content();
tpl_footer();
```

header.php

header.php

```php
<?php if (!defined('WIKI_ROOT')) die(); ?>
<!DOCTYPE html>
<html lang="<?php echo htmlspecialchars($GLOBALS['conf']['lang']); ?>">
```

```
<head>
    <meta charset="UTF-8">
    <title><?php tpl_pagetitle(); ?> - <?php echo
htmlspecialchars($GLOBALS['conf']['title']); ?></title>

  <!-- Абсолютный веб-путь к файлу стилей, чтобы он никогда не терялся -->
    <link rel="stylesheet"
href="./lib/tpl/default/css/style.css?v=<?php echo time(); ?>">
  <!-- Подключаем базовую логику редактора -->
<script src="./assets/js/toolbar.js" defer></script>
<!-- Подключаем изолированную логику модального окна загрузки картинок -->
<script src="./assets/js/media-modal.js" defer></script>

</head>
<body>
<div class="wrapper">
    <header>
        <div class="logo">
            <a href="?id=start">🏠 <?php echo
htmlspecialchars($GLOBALS['conf']['title']); ?></a>
        </div>
        <nav>
            <a href="?id=<?php echo urlencode($GLOBALS['ID']);
?>&do=show">Читать</a>
            <a href="?id=<?php echo urlencode($GLOBALS['ID']);
?>&do=edit">Правка</a>
            <a href="?id=<?php echo urlencode($GLOBALS['ID']);
?>&do=media">Медиафайлы</a>

            <?php if (auth_is_logged()): ?>
                <span class="user-info">👤 <?php $usr = auth_get_user();
echo htmlspecialchars($usr['name']); ?></span>
                <a href="?id=<?php echo urlencode($GLOBALS['ID']);
?>&do=logout" style="color: #ffb3b3;">Выйти</a>
            <?php else: ?>
                <a href="?id=<?php echo urlencode($GLOBALS['ID']);
?>&do=login">Войти</a>
            <?php endif; ?>
        </nav>
    </header>
    <main>
        <div class="breadcrumbs">Вы здесь: <?php tpl_pagetitle();
?></div>
```

footer.php

footer.php

```
<?php if (!defined('WIKI_ROOT')) die(); ?>
    </main>
    <footer>
        <p>&copy; <?php echo date('Y'); ?> <?php echo
htmlspecialchars($GLOBALS['conf']['title']); ?>. Разработано в учебных
целях.</p>
    </footer>
</div>
</body>
</html>
```

style.css

style.css

```
/*
================================================================================
  1. ГЛОБАЛЬНЫЕ СТИЛИ И КАРКАС ШАБЛОНА САЙТА (MicroWiki Framework)
================================================================================ */
body {
    font-family: Arial, sans-serif;
    margin: 0;
    padding: 0;
    color: #333;
    background: #f4f6f9;
}

/* Главная центрирующая обертка сайта */
.wrapper {
    max-width: 1100px;
    margin: 20px auto;
    background: #fff;
    box-shadow: 0 0 15px rgba(0,0,0,0.1);
    min-height: 90vh;
    display: flex;
    flex-direction: column;
    border-radius: 5px;
    overflow: hidden;
}

/* Шапка сайта */
header {
    background: #2b5e8f;
    padding: 15px 25px;
```

```
    display: flex;
    justify-content: space-between;
    align-items: center;
    color: #fff;
}

header .logo a {
    font-size: 22px;
    font-weight: bold;
    color: #fff;
    text-decoration: none;
}

/* Меню навигации в шапке */
nav a {
    margin-left: 15px;
    color: #fff;
    text-decoration: none;
    font-size: 14px;
    opacity: 0.85;
    transition: opacity 0.2s;
    border-bottom: 1px dashed transparent;
    padding-bottom: 2px;
}

nav a:hover {
    opacity: 1;
    border-bottom-style: solid;
    border-bottom-color: #fff;
}

/* Основной блок контента */
main {
    padding: 35px;
    flex-grow: 1;
    background: #fff;
}

/* Подвал сайта */
footer {
    background: #f8f9fa;
    border-top: 1px solid #e9ecef;
    padding: 20px;
    text-align: center;
    font-size: 13px;
    color: #6c757d;
}

/* Хлебные крошки (Вы здесь) */
.breadcrumbs {
    font-size: 13px;
```

```
    color: #6c757d;
    margin-bottom: 25px;
    padding-bottom: 10px;
    border-bottom: 1px solid #eee;
}

/* Плашка информации о залогиненном пользователе */
.user-info {
    margin-left: 15px;
    font-size: 13px;
    padding: 4px 10px;
    background: rgba(0,0,0,0.2);
    border-radius: 3px;
    color: #fff;
    display: inline-block;
}

/* Отрендеренный вики-контент страниц */
.wiki-content h1 { color: #2b5e8f; margin-top: 20px; margin-bottom:
10px; padding-bottom: 5px; border-bottom: 1px solid #eee; }
.wiki-content h2 { color: #333; margin-top: 18px; margin-bottom: 8px; }
.wiki-content h3 { color: #555; margin-top: 15px; margin-bottom: 6px; }

/*
================================================================================
  2. КОМПОНЕНТ РЕДАКТОРА И МОНОЛИТНОГО ТУЛБАРА  DOKUWIKI  С ПИКЕРАМИ
================================================================================ */
.wiki-editor-container {
    margin-top: 15px;
    display: flex;
    flex-direction: column;
}

.doku-toolbar-container {
    background: #fcfcfc;
    border: 1px solid #ccc;
    border-bottom: none;
    padding: 5px 6px;
    border-top-left-radius: 3px;
    border-top-right-radius: 3px;
    display: flex;
    flex-wrap: wrap;
    align-items: center;
    box-sizing: border-box;
    width: 100%;
}
```

```
.doku-toolbar-group {
    display: flex;
    margin-right: 6px;
    border-radius: 2px;
    box-sizing: border-box;
}
.doku-toolbar-group:last-child {
    margin-right: 0;
}

.doku-btn-ico {
    background: #ffffff;
    background: linear-gradient(to bottom, #ffffff 0%, #f3f3f3 100%);
    border: 1px solid #b8b8b8;
    margin-right: -1px;
    padding: 0 8px;
    cursor: pointer;
    font-size: 11px;
    font-family: Arial, sans-serif;
    color: #333;
    display: inline-flex;
    align-items: center;
    justify-content: center;
    height: 24px;
    min-width: 26px;
    position: relative;
    box-sizing: border-box;
    outline: none;
}
.doku-toolbar-group .doku-btn-ico:last-child {
    margin-right: 0;
    border-right: 1px solid #b8b8b8;
}
.doku-btn-ico:hover {
    background: linear-gradient(to bottom, #ffffff 0%, #e2e8f0 100%);
    border-color: #888;
    z-index: 2;
}

.doku-arrow {
    position: absolute;
    right: 1px;
    bottom: 1px;
    width: 0;
    height: 0;
    border-style: solid;
    border-width: 0 0 3px 3px;
    border-color: transparent transparent #555 transparent;
}

/* ------------------------------------------------------------------------
```

```
-----
  НОВЫЕ ГОРИЗОНТАЛЬНЫЕ ВЫПАДАЮЩИЕ ПАНЕЛИ (PICKERS)
    ------------------------------------------------------------------------
----- */
.doku-dropdown {
    position: relative;
    display: inline-block;
    box-sizing: border-box;
}

/* Контейнер подменю: раскрывается горизонтально с тенью */
.doku-dropdown-content {
    display: none;
    position: absolute;
    left: 0;
    top: 25px;
    background-color: #ffffff;
    border: 1px solid #b8b8b8;
    box-shadow: 0px 3px 8px rgba(0,0,0,0.12);
    z-index: 999;
    border-radius: 3px;
    padding: 3px;
}

/* Монолитная флекс-лента внутри подменю */
.doku-sub-toolbar {
    display: flex;
    border-radius: 2px;
    overflow: hidden;
}

/* Кнопка внутри выпадающей ленты */
.doku-sub-btn {
    background: #ffffff;
    background: linear-gradient(to bottom, #ffffff 0%, #f9f9f9 100%);
    border: 1px solid #c5c5c5;
    margin-right: -1px;
    padding: 0 6px;
    cursor: pointer;
    font-size: 11px;
    display: inline-flex;
    align-items: center;
    justify-content: center;
    height: 22px;
    min-width: 24px;
    box-sizing: border-box;
    outline: none;
}
.doku-sub-toolbar .doku-sub-btn:last-child {
```

```
    margin-right: 0;
    border-right: 1px solid #c5c5c5;
}
.doku-sub-btn:hover {
    background: #e6effa;
    border-color: #999;
    z-index: 3;
}

/* КЛАСС КРАСИВЫХ  CSS-ИКОНОК ФАЙЛОВ КОДА */
.file-ico {
    font-size: 7px;
    font-family: 'Arial Black', sans-serif;
    font-weight: bold;
    width: 15px;
    height: 13px;
    line-height: 14px;
    text-align: center;
    border-radius: 1px;
    color: #fff;
    box-shadow: 0 1px 1px rgba(0,0,0,0.1);
}
.css-html { background: #e34c26; } /* Красный HTML */
.css-php  { background: #4f5b93; } /* Синий PHP */
.css-css  { background: #563d7c; } /* Фиолетовый CSS */
.css-js   { background: #f1e05a; color: #333; } /* Желтый JS */
.css-sql  { background: #00758f; } /* Бирюзовый SQL */
.css-bash { background: #4eaa25; } /* Зеленый Bash */

.doku-dropdown-show {
    display: block !important;
}

/* Текстовое поле */
.wiki-textarea {
    width: 100%;
    height: 380px;
    padding: 12px;
    font-family: 'Courier New', monospace;
    font-size: 14px;
    box-sizing: border-box;
    border: 1px solid #ccc;
    border-bottom-left-radius: 3px;
    border-bottom-right-radius: 3px;
    line-height: 1.6;
    resize: vertical;
    outline: none;
}


/*
```

```
/* ================================================================================
   3. КНОПКИ УПРАВЛЕНИЯ СНИЗУ И МЕДИА-МЕНЕДЖЕР
================================================================================ */
.doku-btn {
    background: #fff;
    border: 1px solid #ccc;
    padding: 4px 14px;
    font-size: 13px;
    color: #333;
    border-radius: 3px;
    cursor: pointer;
    margin-right: 5px;
}

.doku-btn:hover {
    background: #eee;
}

/* Сетка отображения Медиа-менеджера */
.media-grid {
    display: grid;
    grid-template-columns: repeat(auto-fill, minmax(180px, 1fr));
    gap: 20px;
}

.media-item {
    border: 1px solid #eee;
    padding: 10px;
    border-radius: 4px;
    text-align: center;
    background: #fff;
    display: flex;
    flex-direction: column;
    justify-content: space-between;
    box-shadow: 0 2px 4px rgba(0,0,0,0.02);
}

/*
================================================================================
   4. СИНТАКСИЧЕСКИЙ БЛОК КОДА  DOKUWIKI  (ОДИНАКОВЫЙ РАЗМЕР ШРИФТА ДЛЯ ЦИФР)
================================================================================ */
.doku-code-block {
    margin: 15px 0;
    position: relative;
```

```
    box-sizing: border-box;
    display: block;
    width: 100%;
}

.doku-code-file-label {
    display: inline-flex;
    align-items: center;
    background: #f7f9fa;
    border: 1px solid #dcdcdc;
    border-bottom: none;
    padding: 4px 14px;
    font-size: 12px;
    font-family: Arial, sans-serif;
    color: #2b5e8f;
    border-top-left-radius: 3px;
    border-top-right-radius: 3px;
    margin-left: 10px;
}

.doku-code-content-wrapper {
    background: #f7f9fa;
    border: 1px solid #dcdcdc;
    border-radius: 3px;
    overflow: hidden;
    width: 100%;
    box-sizing: border-box;
}

.doku-code-content {
    margin: 0;
    padding: 8px 0;
    font-family: 'Consolas', 'Courier New', monospace;
    font-size: 14px;
    overflow-x: auto;
    background: #f7f9fa;
    line-height: 1.4;
}

.doku-code-content > div {
    display: flex;
    padding: 0px 0;
    width: 100%;
    box-sizing: border-box;
}

/* ИСПРАВЛЕНО: Размер шрифта номеров строк увеличен до 14px (один в один с
кодом) */
.code-ln {
    color: #666666;
    width: auto; /* Отменяем жесткий min-width */
```

```
    min-width: 25px; /* Минимальный базис для однозначных чисел */
    text-align: right;
    padding-left: 5px; /* Фиксированный зазор до серой линии */
    padding-right: 5px; /* Фиксированный зазор до серой линии */
    margin-right: 12px;  /* Фиксированный зазор от линии до кода */
    border-right: 1px solid #dcdcdc;
    display: inline-block;
    user-select: none;
    font-size: 14px;
    box-sizing: border-box;
}

.code-txt {
    white-space: pre;
    color: #000000;
    display: inline-block;
}

.code-line-active {
    background-color: #fff9db !important;
    width: 100%;
}
```

index.php

index.php

```
<?php
/**
 * MicroWiki Engine - Главный диспетчер (Точка входа)
 * index.php
 */


declare(strict_types=1);

error_reporting(E_ALL);
ini_set('display_errors', '1');

define('WIKI_ROOT', __DIR__ . '/');

global $conf, $ID, $ACT, $wikiStorage, $mediaManager;
$conf = [];

if (file_exists(WIKI_ROOT . 'conf/local.php')) {
    include WIKI_ROOT . 'conf/local.php';
```

```
}

$conf['title']    = $conf['title'] ?? 'MicroWiki';
$conf['template'] = $conf['template'] ?? 'default';
$conf['lang']     = $conf['lang'] ?? 'ru';

define('WIKI_TPL', WIKI_ROOT . 'lib/tpl/' . $conf['template'] . '/');

require_once WIKI_ROOT . 'src/Auth.php';
require_once WIKI_ROOT . 'src/Storage.php';
require_once WIKI_ROOT . 'src/MediaManager.php';

auth_setup();

$wikiStorage  = new Storage(WIKI_ROOT . 'data');
$mediaManager = new MediaManager(WIKI_ROOT . 'data');

$ID = isset($_GET['id']) ? preg_replace('/[^a-zA-Z0-9_\-:]/', '',
$_GET['id']) : 'start';
$ID = trim($ID, ':');
if (empty($ID)) { $ID = 'start'; }

$ACT = isset($_GET['do']) ? preg_replace('/[^a-z_]/', '', $_GET['do'])
: 'show';

// 1. АСИНХРОННЫЙ API ЭНДПОИНТ ДЛЯ МОДАЛЬНОГО ОКНА КАРТИНОК
if ($ACT === 'api_upload' && $_SERVER['REQUEST_METHOD'] === 'POST') {
    header('Content-Type: application/json');
    if (!auth_is_logged()) { echo json_encode(['success' => false,
'error' => 'Необходима авторизация.']); exit; }
    $currentNS = str_contains($ID, ':') ? substr($ID, 0,
(int)strrpos($ID, ':')) : '';
    if ($mediaManager->upload($currentNS)) {
        $uploadedName = preg_replace('/[^a-zA-Z0-9_\-\.]/', '',
$_FILES['uploadfile']['name']);
        $nsPrefix = $currentNS ? $currentNS . ':' : '';
        $wikiTag = '{{:' . $nsPrefix . $uploadedName . '|}}';
        echo json_encode(['success' => true, 'tag' => $wikiTag]);
        exit;
    } else {
        echo json_encode(['success' => false, 'error' => 'Ошибка
загрузки.']); exit;
    }
}

// 2. ИСПРАВЛЕНО: АСИНХРОННЫЙ API ЭНДПОИНТ ДЛЯ ЖИВОГО ПРЕДПРОСМОТРА (В
самом верху до фильтрации)
if ($ACT === 'api_preview' && $_SERVER['REQUEST_METHOD'] === 'POST') {
    header('Content-Type: text/html; charset=UTF-8');

    $textToParse = isset($_POST['wikitext']) ?
```

```
(string)$_POST['wikitext'] : '';

    require_once WIKI_ROOT . 'src/Parser.php';
    $parser = new Parser();

    // Отдаем только чистый отрендеренный текст и намертво прерываем выполнение
    echo $parser->render($textToParse);
    exit;
}

//
==========================================================================
==

// Обработка авторизации
if ($ACT === 'login' && $_SERVER['REQUEST_METHOD'] === 'POST') {
    $username = isset($_POST['u']) ? (string)$_POST['u'] : '';
    $password = isset($_POST['p']) ? (string)$_POST['p'] : '';
    if (auth_login($username, $password)) {
        header('Location: ?id=' . urlencode($ID));
        exit;
    }
}

// Обработка выхода
if ($ACT === 'logout') {
    auth_logout();
    header('Location: ?id=' . urlencode($ID));
    exit;
}

// Сохранение вики-страницы
if ($ACT === 'save' && $_SERVER['REQUEST_METHOD'] === 'POST') {
    if (!auth_is_logged()) { die('Ошибка доступа.'); }
    $textToSave = $_POST['wikitext'] ?? '';
    if ($wikiStorage->save($ID, $textToSave)) {
        header('Location: ?id=' . urlencode($ID) . '&do=show');
        exit;
    }
}

// Стандартная загрузка медиафайлов (из вкладки "Медиафайлы")
if ($ACT === 'upload' && $_SERVER['REQUEST_METHOD'] === 'POST') {
    if (!auth_is_logged()) { die('Ошибка доступа.'); }
    $currentNS = str_contains($ID, ':') ? substr($ID, 0,
(int)strrpos($ID, ':')) : '';
    if ($mediaManager->upload($currentNS)) {
        header('Location: ?id=' . urlencode($ID) . '&do=media');
        exit;
```

```
    }
}

// Стандартное удаление медиафайлов
if ($ACT === 'delete') {
    if (!auth_is_logged()) { die('Ошибка доступа.'); }
    $currentNS = str_contains($ID, ':') ? substr($ID, 0,
(int)strrpos($ID, ':')) : '';
    $fileToDelete = isset($_GET['file']) ? (string)$_GET['file'] : '';
    if (!empty($fileToDelete)) {
        $mediaManager->delete($currentNS, $fileToDelete);
    }
    header('Location: ?id=' . urlencode($ID) . '&do=media');
    exit;
}

// Массив легальных команд (Добавлено действие api_upload)
$allowed_actions = ['show', 'edit', 'save', 'media', 'upload',
'delete', 'login', 'api_upload', 'api_preview'];
if (!in_array($ACT, $allowed_actions, true)) {
    $ACT = 'show';
}

require_once WIKI_ROOT . 'src/Template.php';

if (file_exists(WIKI_TPL . 'main.php')) {
    include WIKI_TPL . 'main.php';
} else {
    die("Критическая ошибка движка: Главный файл шаблона не найден.");
}
```

template.php

template.php

```
<?php
/**
 * Модуль шаблонизации (Аналог inc/template.php в DokuWiki)
 */

declare(strict_types=1);

if (!defined('WIKI_ROOT')) {
    die('Прямой доступ к файлу запрещен.');
}

/**
 * Выводит заголовок текущей страницы для тега <title>
```

```
 */
function tpl_pagetitle(): void {
    global $ID;
    echo htmlspecialchars(str_replace(':', ' » ', $ID));
}

/**
 * Подключает верхнюю часть  HTML-шаблона (Шапку)
 */
function tpl_header(): void {
    if (file_exists(WIKI_TPL . 'header.php')) {
        include WIKI_TPL . 'header.php';
    }
}

/**
 * Подключает нижнюю часть  HTML-шаблона (Подвал)
 */
function tpl_footer(): void {
    if (file_exists(WIKI_TPL . 'footer.php')) {
        include WIKI_TPL . 'footer.php';
    }
}

/**
 * Системная функция ядра для подключения панели инструментов редактора
 */
function tpl_extensions_toolbar(): void {
    if (file_exists(WIKI_ROOT . 'src/toolbar.php')) {
        include WIKI_ROOT . 'src/toolbar.php';
    }
}

/**
 * Главный диспетчер вывода контента в шаблоне
 */
function tpl_content(): void {
    global $ACT;
    switch ($ACT) {
        case 'login':
            tpl_login_form();
            break;
        case 'edit':
            tpl_edit_form();
            break;
        case 'media':
            tpl_media_manager();
            break;
        case 'show':
```

```
        default:
            tpl_page_body();
            break;
    }
}

/**
 * ВЫВОД СТРАНИЦЫ: Читает файл, рендерит через Parser и выводит живой HTML
 */
function tpl_page_body(): void {
    global $ID, $wikiStorage;

    echo "<h1 style='margin-top:0; color:#2b5e8f;'>" .
htmlspecialchars($ID) . "</h1>";

    if ($wikiStorage->exists($ID)) {
        $rawText = $wikiStorage->read($ID);

        require_once WIKI_ROOT . 'src/Parser.php';
        $parser = new Parser();

        $htmlContent = $parser->render($rawText);

        echo "<div class='wiki-content' style='line-height:1.6; font-
size:15px;'>";
        echo $htmlContent;
        echo "</div>";
    } else {
        echo "<div style='background:#fff9e6; border:1px solid #ffe0b3;
padding:15px; color:#b36b00;'>";
        echo "Страницы с именем <strong>" . htmlspecialchars($ID) .
"</strong> еще не существует.";
        echo "</div>";
        if (auth_is_logged()) {
            echo "<p><a href='?id=" . urlencode($ID) . "&do=edit'
style='display:inline-block; margin-top:15px; padding:8px 15px;
background:#2b5e8f; color:#fff; text-decoration:none; border-
radius:3px; font-weight:bold;'>Создать эту страницу</a></p>";
        } else {
            echo "<p style='font-size:14px; color:#666;'>Войдите в
систему, чтобы иметь возможность создать её.</p>";
        }
    }
}

/**
 * ФОРМА РЕДАКТИРОВАНИЯ И ПРЕДПРОСМОТРА СНИЗУ
 */
function tpl_edit_form(): void {
    global $ID, $wikiStorage;
```

```
    if (!auth_is_logged()) {
        echo "<p style='color:red; font-weight:bold;'>Ошибка: Только
зарегистрированные пользователи могут редактировать страницы.</p>";
        return;
    }

    $currentText = $wikiStorage->read($ID);

    echo "<div style='font-size:13px; color:#444; background:#fbfbfb;
padding:10px; border:1px solid #ddd; margin-bottom:15px;'>Отредактируйте
страницу и нажмите «Сохранить».</div>";
    echo "<h1>Правка страницы: " . htmlspecialchars($ID) . "</h1>";

    echo '<form method="POST" action="?id=' . urlencode($ID) .
'&do=save">';
    echo '<div class="wiki-editor-container">';

    // Системный тулбар кнопок
    tpl_extensions_toolbar();

    // Поле ввода текста
    echo '<textarea id="wikiEditor" name="wikitext" class="wiki-
textarea">' . htmlspecialchars($currentText) . '</textarea>';
    echo '</div>';

    // Блок нижних кнопок управления движком (В стиле DokuWiki)
    echo '<div style="margin-top:15px; background:#f8f9fa;
padding:15px; border:1px solid #ccc; border-radius:4px;">';
    echo '  <div style="margin-bottom:12px;">';
    echo '    <button type="submit" class="doku-btn"
style="background:#2b5e8f; color:#fff; font-
weight:bold;">Сохранить</button>';
    echo '    <button type="button" class="doku-btn"
onclick="runLivePreview()">Просмотр</button>';
    echo '    <a href="?id=' . urlencode($ID) . '" class="doku-btn"
style="text-decoration:none; display:inline-block; text-align:center;
line-height:1.2;">Отменить</a>';
    echo '  </div>';

    echo '  <div>';
    echo '    <label style="display:block; font-size:13px; font-
weight:bold; margin-bottom:4px; color:#333;">Сводка изменений:</label>';
    echo '    <input type="text" name="summary" style="width:100%; max-
width:500px; padding:6px; border:1px solid #ccc; border-radius:3px;
font-size:14px;" placeholder="Что именно вы изменили...">';
    echo '  </div>';
    echo '</div>';
    echo '</form>';
```

```
    // БЛОК ДИНАМИЧЕСКОГО ПРЕДПРОСМОТРА СНИЗУ
    echo '<div id="previewBlock" style="display:none; margin-top:40px;
border-top:2px dashed #bbb; padding-top:20px;">';
    echo '  <h2 style="color:#2b5e8f; margin-top:0;">Просмотр</h2>';
    echo '  <p style="font-size:13px; color:#666; font-style:italic;
margin-bottom:15px;">Здесь показано, как ваш text будет выглядеть.
<strong>Внимание: текст ещё не сохранён!</strong></p>';
    echo '  <div id="previewContent" class="wiki-content"
style="background:#fff; padding:20px; border:1px solid #ddd; border-
radius:4px; min-height:100px; line-height:1.6;"></div>';
    echo '</div>';
}

/**
 * ИНТЕРФЕЙС АВТОРИЗАЦИИ
 */
function tpl_login_form(): void {
    if (auth_is_logged()) {
        echo "<p>Вы уже успешно авторизованы в системе.</p>";
        return;
    }

    if (!empty($GLOBALS['login_error'])) {
        echo "<div style='color: #a00; background: #ffebeb; padding:
15px; border: 1px solid #faaaaa; margin-bottom: 15px; font-
family:monospace; font-size:13px; line-height:1.4;'>";
        echo $GLOBALS['login_error'];
        echo "</div>";
    }

    echo '
    <div style="max-width: 300px; background: #f9f9f9; padding: 25px;
border: 1px solid #ddd; border-radius: 4px;">
        <h3 style="margin-top:0; margin-bottom:20px;">Авторизация</h3>
        <form method="POST">
            <label style="display:block; margin-bottom:5px; font-
weight:bold; font-size:14px;">Логин:</label>
            <input type="text" name="u" required style="width:100%;
margin-bottom:15px; padding:6px; box-sizing: border-box;">

            <label style="display:block; margin-bottom:5px; font-
weight:bold; font-size:14px;">Пароль:</label>
            <input type="password" name="p" required style="width:100%;
margin-bottom:20px; padding:6px; box-sizing: border-box;">

            <button type="submit" style="padding: 7px 20px;
background:#2b5e8f; color:#fff; border:none; border-radius:3px;
cursor:pointer; font-weight:bold;">Войти</button>
        </form>
    </div>';
}
```

```
/**
 * ИНТЕРФЕЙС МЕДИАМЕНЕДЖЕРА
 */
function tpl_media_manager(): void {
    global $ID, $mediaManager;
    $currentNS = str_contains($ID, ':') ? substr($ID, 0,
(int)strrpos($ID, ':')) : '';
    echo "<h1>Медиа-менеджер для пространства имен: " .
htmlspecialchars($currentNS ?: '[корень]') . "</h1>";
    if (auth_is_logged()) {
        echo '
        <div style="background:#f9f9f9; padding:20px; border:1px solid
#ddd; border-radius:4px; margin-bottom:30px;">
            <h3 style="margin-top:0; margin-bottom:15px; font-
size:16px;">Загрузить файл в этот раздел</h3>
            <form method="POST" action="?id=' . urlencode($ID) .
'&do=upload" enctype="multipart/form-data">
                <input type="file" name="uploadfile" required>
                <button type="submit" style="padding:5px 15px;
background:#2b5e8f; color:#fff; border:none; cursor:pointer; font-
weight:bold; border-radius:3px;">Загрузить</button>
            </form>
            <small style="color:#666; display:block; margin-
top:5px;">Разрешены форматы: JPG, PNG, GIF, PDF, TXT</small>
        </div>';
    }
    $files = $mediaManager->getFiles($currentNS);
    if (empty($files)) { echo "<p style='color:#888; font-
style:italic;'>В данном разделе файлов пока нет.</p>"; return; }
    echo '<div style="display:grid; grid-template-columns: repeat(auto-
fill, minmax(180px, 1fr)); gap:20px;">';
    foreach ($files as $file) {
        $webPath = 'data/media/' . ($currentNS ? str_replace(':', '/',
$currentNS) . '/' : '') . $file['name'];
        $sizeKb = round($file['size'] / 1024, 1);
        echo '<div style="border:1px solid #eee; padding:10px; border-
radius:4px; text-align:center; background:#fff; display:flex; flex-
direction:column; justify-content:between; box-shadow: 0 2px 4px
rgba(0,0,0,0.02);">';
        if ($file['is_image']) {
            echo '<div style="height:100px; display:flex; align-
items:center; justify-content:center; margin-bottom:10px;
overflow:hidden; background:#f4f4f4;"><img src="' .
htmlspecialchars($webPath) . '" style="max-width:100%; max-height:100%;
object-fit:contain;"></div>';
        } else {
            echo '<div style="height:100px; display:flex; align-
items:center; justify-content:center; margin-bottom:10px;
```

```
background:#e9ecef; font-size:32px; color:#6c757d; font-weight:bold;
text-transform:uppercase;">' . htmlspecialchars($file['ext']) .
'</div>';
        }
        echo '<div style="font-size:13px; font-weight:bold; word-
break:break-all; text-align:left; flex-grow:1; margin-bottom:8px;">' .
htmlspecialchars($file['name']) . '</div>';
        echo '<div style="font-size:11px; color:#666; text-align:left;
margin-bottom:10px;">Размер:' . $sizeKb . 'КБ</div>';
        if (auth_is_logged()) {
            $delUrl = '?id=' . urlencode($ID) . '&do=delete&file=' .
urlencode($file['name']);
            echo '<a href="' . $delUrl . '" onclick="return
confirm(\'Удалить этот файл?\')" style="color:#c00; font-size:12px; text-
decoration:none; display:block; padding-top:5px; border-top:1px solid
#f4f4f4;">Удалить</a>';
        }
        echo '</div>';
    }
    echo '</div>';
}
```

auth.php

auth.php

```php
<?php
/**
 * Слой аутентификации и работы с сессиями (Аналог inc/auth.php в DokuWiki)
 */

declare(strict_types=1);

if (!defined('WIKI_ROOT')) {
    die('Прямой доступ к файлу запрещен.');
}

/**
 * Инициализация и безопасный старт сессии PHP
 */
function auth_setup(): void {
    ini_set('session.cookie_httponly', '1');
    ini_set('session.use_only_cookies', '1');

    if (session_status() === PHP_SESSION_NONE) {
        session_start();
    }
}
```

```php
/**
 * Проверка учетных данных пользователя по файлу  users.auth.php
 */
function auth_login(string $user, string $pass): bool {
    $auth_file = WIKI_ROOT . 'conf/users.auth.php';

    if (!file_exists($auth_file)) {
        return false;
    }

    $handle = fopen($auth_file, 'r');
    if (!$handle) {
        return false;
    }

    while (($line = fgets($handle)) !== false) {
        $line = str_replace(["\r", "\n"], '', $line);
        $line = trim($line);

        if (str_starts_with($line, '<?php') || str_starts_with($line,
'#') || empty($line)) {
            continue;
        }

        $parts = explode(':', $line);
        if (count($parts) < 5) {
            continue;
        }

        $login  = trim((string)array_shift($parts));
        $hash   = trim((string)array_shift($parts));
        $name   = trim((string)array_shift($parts));
        $email  = trim((string)array_shift($parts));

        if ($login === trim($user)) {
            // Штатная, безопасная проверка пароля по хэшу
            if (password_verify(trim($pass), $hash)) {
                fclose($handle);

                $_SESSION['auth_status'] = 'logged_in';
                $_SESSION['auth_login']  = $login;
                $_SESSION['auth_name']   = $name;
                $_SESSION['auth_email']  = $email;
                return true;
            }
        }
    }
```

```
    fclose($handle);
    return false;
}

/**
 * Выход пользователя из системы (Удаление сессии)
 */
function auth_logout(): void {
    unset($_SESSION['auth_status']);
    unset($_SESSION['auth_login']);
    unset($_SESSION['auth_name']);
    unset($_SESSION['auth_email']);
    session_destroy();
}

/**
 * Проверка, авторизован ли текущий посетитель
 */
function auth_is_logged(): bool {
    return isset($_SESSION['auth_status']) && $_SESSION['auth_status']
=== 'logged_in';
}

/**
 * Получение имени текущего авторизованного пользователя
 */
function auth_get_name(): string {
    return isset($_SESSION['auth_name']) ?
(string)$_SESSION['auth_name'] : 'Гость';
}

/**
 * Возвращает массив данных пользователя для совместимости с шаблоном
header.php
 */
function auth_get_user(): array {
    return [
        'login' => isset($_SESSION['auth_login']) ?
(string)$_SESSION['auth_login'] : '',
        'name'  => auth_get_name(),
        'email' => isset($_SESSION['auth_email']) ?
(string)$_SESSION['auth_email'] : ''
    ];
}
```

users.auth.php

users.auth.php

```
<?php die(); ?>
# MicroWiki Users Auth File
# users.auth.php
# Формат записи: login:password_hash:real_name:email:groups
admin:$2y$10$csfWQtrGXsmJcNQ3aaERWOJ3Am/ljBh/r/gaG8gJM0ajo41FLH0aK:Адми
нистратор:admin@wiki.local:admin,user
editor:$2y$10$7R9rMkGzE8hWd2RjK1l8OuOl/Xp4yq3K9vR8fD7bC0mN5eK6wPy1a:Ред
актор Вася:vasya@wiki.local:user
```

hash.php

hash.php

```
<?php
echo password_hash('admin123', PASSWORD_DEFAULT);
```

storage.php

storage.php

```
<?php
/**
 * Класс управления плоскими файлами страниц (Аналог файлового ядра DokuWiki)
 * storage.php
 */

declare(strict_types=1);

if (!defined('WIKI_ROOT')) {
    die('Прямой доступ к файлу запрещен.');
}

class Storage {
    private string $baseDir;

    /**
   * Конструктор инициализирует базовую директорию для хранения текстов
     */
    public function __construct(string $baseDataPath) {
        // Гарантируем правильные слеши в путях для Windows/Linux систем
        $this->baseDir = rtrim(str_replace('\\', '/', $baseDataPath),
```

```
'/') . '/pages/';
    }

    /**
   * Преобразует Wiki ID (например, "отдел:инструкция") в абсолютный путь к .txt файлу
     */
    public function resolvePath(string $id): string {
        // Безопасность: заменяем любые попытки выйти из папки (файлы вида ../)
        $id = str_replace(['..', '\\'], '', $id);

        // В DokuWiki двоеточие — это разделитель папок. Меняем ":" на "/"
        $relativePath = str_replace(':', '/', $id);

        return $this->baseDir . $relativePath . '.txt';
    }

    /**
   * Проверяет, существует ли физический файл страницы
     */
    public function exists(string $id): bool {
        return file_exists($this->resolvePath($id));
    }

    /**
   * Безопасно читает содержимое вики-страницы
     */
    public function read(string $id): string {
        $filePath = $this->resolvePath($id);
        if (file_exists($filePath)) {
            return file_get_contents($filePath) ?: '';
        }
        return '';
    }

    /**
   * Сохраняет текст страницы, автоматически создавая вложенную структуру папок
     */
    public function save(string $id, string $content): bool {
        $filePath = $this->resolvePath($id);
        $dirPath  = dirname($filePath);

        // Если папки для пространства имен не существуют, рекурсивно создаем их
        if (!is_dir($dirPath)) {
            if (!mkdir($dirPath, 0755, true) && !is_dir($dirPath)) {
                return false; // Не удалось создать папку (проблема прав доступа)
            }
        }

        // Записываем данные в файл атомарно
```

```
        return file_put_contents($filePath, $content) !== false;
    }
}
```

MediaManager.php

MediaManager.php

```php
<?php
/**
 * Класс управления медиафайлами и изображениями (Аналог подсистемы Media в DokuWiki)
 */

declare(strict_types=1);

if (!defined('WIKI_ROOT')) {
    die('Прямой доступ к файлу запрещен.');
}

class MediaManager {
    private string $baseMediaDir;
    private array $allowedTypes;

    public function __construct(string $baseDataPath) {
        $this->baseMediaDir = rtrim(str_replace('\\', '/', $baseDataPath), '/') . '/media/';
        // Белый список разрешенных расширений файлов
        $this->allowedTypes = [
            'jpg', 'jpeg', 'png', 'gif', 'pdf', 'txt'
        ];
    }

    public function resolveDir(string $ns): string {
        $ns = str_replace(['..', '\\'], '', $ns);
        $relativePath = str_replace(':', '/', $ns);
        return rtrim($this->baseMediaDir . $relativePath, '/') . '/';
    }

    public function getFiles(string $ns): array {
        $dir = $this->resolveDir($ns);
        if (!is_dir($dir)) {
            return [];
        }

        $result = [];
```

```
        $files = scandir($dir);

        foreach ($files as $file) {
            if ($file === '.' || $file === '..') {
                continue;
            }
            if (is_file($dir . $file)) {
                $ext = strtolower(pathinfo($file, PATHINFO_EXTENSION));
                $isImage = in_array($ext, ['jpg', 'jpeg', 'png',
'gif'], true);

                $result[] = [
                    'name' => $file,
                    'size' => filesize($dir . $file),
                    'is_image' => $isImage,
                    'ext' => $ext
                ];
            }
        }
        return $result;
    }

    /**
  * Безопасная обработка и загрузка файла из глобального массива  $_FILES
     */
    public function upload(string $ns): bool {
        if (!isset($_FILES['uploadfile'])) {
            return false;
        }

        $error = $_FILES['uploadfile']['error'];
        if ($error !== UPLOAD_ERR_OK) {
            return false;
        }

        $name     = $_FILES['uploadfile']['name'];
        $tmp_name = $_FILES['uploadfile']['tmp_name'];

        // Очищаем имя файла от опасных символов
        $filename = preg_replace('/[^a-zA-Z0-9_\-\.]/', '', $name);
        $ext = strtolower(pathinfo($filename, PATHINFO_EXTENSION));

        // Жесткая проверка: если расширения нет в белом списке — бракуем загрузку
        if (!in_array($ext, $this->allowedTypes, true)) {
            return false;
        }

        $targetDir = $this->resolveDir($ns);
        if (!is_dir($targetDir)) {
            mkdir($targetDir, 0755, true);
        }
```

```
        // Переносим файл из временной папки сервера в каталог data/media/
        return move_uploaded_file($tmp_name, $targetDir . $filename);
    }

    public function delete(string $ns, string $filename): bool {
        $filename = preg_replace('/[^a-zA-Z0-9_\-\.]/', '', $filename);
        $filePath = $this->resolveDir($ns) . $filename;

        if (file_exists($filePath) && is_file($filePath)) {
            return unlink($filePath);
        }
        return false;
    }
}
```

Parser.php

Parser.php

```
<?php
/**
 * Класс трансляции вики-разметки в HTML (Аналог подсистемы парсера DokuWiki)
 */

declare(strict_types=1);

if (!defined('WIKI_ROOT')) {
    die('Прямой доступ к файлу запрещен.');
}

class Parser {
    public function render(string $text): string {
        require_once WIKI_ROOT . 'src/CodeHighlighter.php';
        $highlighter = new CodeHighlighter();

        $codeBlocks = [];
        $blockCount = 0;

        // Шаг 1. Вырезаем блоки <code> в сыром виде ДО какого-либо
экранирования кавычек
        $text = preg_replace_callback('/<code(.*?)>(.*?)<\/code>/s',
function(array $matches) use (&$codeBlocks, &$blockCount,
$highlighter): string {
            $paramLine = isset($matches[1]) ? (string)$matches[1] : '';
            $codeBody  = isset($matches[2]) ? (string)$matches[2] : '';
```

```
            // Обрабатываем блок кода в чистом виде
            $renderedBlock = $highlighter->highlight($paramLine,
$codeBody);

            // Исправлено: Маркер сделан алфавитно-цифровым, чтобы
htmlspecialchars его не портил
            $marker = "DokuCodeBlock" . $blockCount . "X";
            $codeBlocks[$marker] = $renderedBlock;
            $blockCount++;

            return $marker;
        }, $text);

        // Шаг 2. Безопасность: теперь экранируем ОСТАВШИЙСЯ текст страницы от
XSS уязвимостей
        $text = htmlspecialchars($text, ENT_QUOTES, 'UTF-8');

        // Шаг 3. Парсим заголовки DokuWiki
        $text = preg_replace('/======\s*(.*?)\s*======/',
'<h1>$1</h1>', $text);
        $text = preg_replace('/=====\s*(.*?)\s*=====/', '<h2>$1</h2>',
$text);
        $text = preg_replace('/====\s*(.*?)\s*====/', '<h3>$1</h3>',
$text);

        // Шаг 4. Стили форматирования текста
        $text = preg_replace('/\*\*(.*?)\*\*/', '<strong>$1</strong>',
$text);
        $text = preg_replace('/\/\/(.*?)\/\//', '<em>$1</em>', $text);
        $text = preg_replace('/__(.*?)__/', '<u>$1</u>', $text);

        // Шаг 5. Парсим ссылки [[page_id]] или [[page_id|Текст]]
        $text = preg_replace_callback('/\[\[(.*?)\]\]/', function(array
$matches): string {
            $innerContent = isset($matches[1]) ? (string)$matches[1] :
'';
            if (str_contains($innerContent, '|')) {
                $parts = explode('|', $innerContent, 2);
                $pageId = trim((string)array_shift($parts));
                $label  = trim((string)array_shift($parts));
            } else {
                $pageId = trim($innerContent);
                $label  = $pageId;
            }
            $pageId = ltrim($pageId, ':');
            return '<a href="?id=' . urlencode($pageId) . '">' .
htmlspecialchars($label) . '</a>';
        }, $text);

        // Шаг 6. Парсим медиа-теги изображений {{:картинка.png}}
```

```
        $text = preg_replace_callback('/\{\{:(.*?)\}\}/',
function(array $matches): string {
            $innerMedia = isset($matches[1]) ? (string)$matches[1] :
'';
            if (str_contains($innerMedia, '|')) {
                $mediaParts = explode('|', $innerMedia, 2);
                $mediaPath = trim((string)array_shift($mediaParts));
            } else {
                $mediaPath = trim($innerMedia);
            }
            $realPath = str_replace(':', '/', $mediaPath);
            return '<img src="data/media/' .
htmlspecialchars($realPath) . '" style="max-width:100%; height:auto;
margin: 12px 0; display:block; border: 1px solid #ddd; padding: 4px;
border-radius: 4px;" alt="Медиафайл">';
        }, $text);

        // Шаг 7. Делаем автоматические переносы строк для текста страницы
        $text = nl2br($text);

        // Шаг 8. Финал: возвращаем сохраненные блоки кода обратно на свои места в
обход nl2br
        foreach ($codeBlocks as $marker => $renderedHtml) {
            $text = str_replace($marker, $renderedHtml, $text);
        }

        return $text;
    }
}
```

CodeHighlighter.php

CodeHighlighter.php

```
<?php
/**
 * Системный компонент: Честная динамическая подсветка синтаксиса PHP/JS (стиль
DokuWiki)
 */

declare(strict_types=1);

if (!defined('WIKI_ROOT')) {
    die('Прямой доступ к файлу запрещен.');
}
```

```
class CodeHighlighter {
    public function highlight(string $paramLine, string $code): string
{
        $code = htmlspecialchars_decode($code, ENT_QUOTES);
        $code = trim($code, "\r\n");
        $code = trim($code);

        $paramLine = htmlspecialchars_decode($paramLine, ENT_QUOTES);
        $paramLine = trim($paramLine);
        $parts = explode(' ', $paramLine);

        $firstPart = array_shift($parts);
        $lang = $firstPart !== null ?
strtolower(trim((string)$firstPart)) : 'text';

        $filename = '';
        $highlightLine = 0;
        $startLine = 1;

        // ЖЕСТКОЕ ИСПРАВЛЕНО: Используем именованную группу ?P<fname> для
100% изоляции строки от массивов
        if (preg_match('/"(?P<fname>[^"]+)"/', $paramLine,
$fileMatches)) {
            $filename = isset($fileMatches['fname']) ?
(string)$fileMatches['fname'] : '';

            // Вырезаем кавычки с файлом из параметров
            $paramLine = preg_replace('/"[^"]+"/', '', $paramLine);
            $parts = explode(' ', $paramLine);
        }

        foreach ($parts as $part) {
            $part = trim((string)$part);
            if (empty($part)) continue;

            if (str_contains($part, '=')) {
                $kv = explode('=', $part, 2);
                $key = isset($kv[0]) ? trim((string)$kv[0]) : '';
                $val = isset($kv[1]) ? trim((string)$kv[1]) : '';

                if ($key === 'highlight') $highlightLine = (int)$val;
                if ($key === 'start') $startLine = (int)$val;
            } elseif (empty($filename) && $part !== $lang) {
                $filename = $part;
            }
        }

        $lines = preg_split('/\r\n|\r|\n/', $code);
        if ($lines === false) { $lines = [$code]; }

        $firstCodeLineIdx = -1;
```

```
        foreach ($lines as $idx => $line) {
            $t = trim($line);
            if (!empty($t) && !str_starts_with($t, '*') &&
!str_starts_with($t, '/*') && !str_starts_with($t, '*/') && $t !==
'/**') {
                $firstCodeLineIdx = $idx;
                break;
            }
        }

        $htmlLines = '';
        $currentLineNum = $startLine;

        foreach ($lines as $idx => $line) {
            $trimmedLine = trim($line);
            $isCommentStar = str_starts_with($trimmedLine, '*') ||
str_starts_with($trimmedLine, '*/') || $trimmedLine === '/**' ||
($trimmedLine === '' && $idx < $firstCodeLineIdx);

            $colorLine = $this->colorizeLine($line, $lang,
$isCommentStar);
            $highlightClass = ($currentLineNum === $highlightLine) ? '
class="code-line-active"' : '';

            $htmlLines .= '<div' . $highlightClass . '>';
            $htmlLines .= '<span class="code-ln">' . $currentLineNum .
'</span>';
            $htmlLines .= '<span class="code-txt">' . ($colorLine ===
'' ? ' ' : $colorLine) . '</span>';
            $htmlLines .= '</div>';

            $currentLineNum++;
        }

        $html = '<div class="doku-code-block">';
        if (!empty($filename)) {
            $html .= '<div class="doku-code-file-label">';
            $html .= '<img
src="
AAAAGXRFWHRTb2Z0d2FyZQBBZG9iZSBJbWFnZVJlYWR5ccllPAAAAG9JREFUeNksksENwDA
IA02WSboJm6gZun/bZAt6i6pCH8A+YscgInIuOfuS89SIs6vIs8766m6KyvN7S7g7gIio9m
ZpEXfF3Xg6wF1FfXU36gB3w90g7v6vA+DuZvefOuxwFwN3A7O76wV3A6v7bwwAnwEGADZ0K
9776Xm6AAAAAElFTkSuQmCC" style="margin-right:6px; vertical-
align:middle; width:12px; height:12px;">';
            $html .= htmlspecialchars($filename);
            $html .= '</div>';
        }

        $html .= '<div class="doku-code-content-wrapper"><pre
```

```
class="doku-code-content">' . $htmlLines . '</pre></div>';
        $html .= '</div>';

        return $html;
    }

    private function colorizeLine(string $line, string $lang, bool
$isCommentStar): string {
        $line = htmlspecialchars($line, ENT_NOQUOTES, 'UTF-8');

        if ($lang === 'php' || $lang === 'javascript' || $lang ===
'js') {
            if ($isCommentStar) {
                return '<span style="color:#408080; font-
style:italic;">' . $line . '</span>';
            }

            $commentPart = '';
            if (preg_match('/(\/\/|#)(.*)$/', $line, $matches)) {
                $commentPart = isset($matches[0]) ? (string)$matches[0]
: '';
                $line = str_replace($commentPart,
'%%DOKU_COMMENT_PLACEHOLDER%%', $line);
            }

            $line = preg_replace_callback('/(\'.*?\'|".*?")/',
function(array $m): string {
                return '<span style="color:#ff0000;">' . (isset($m[0])
? $m[0] : '') . '</span>';
            }, $line);

            if ($lang === 'php') {
                $line = preg_replace_callback('/(?<!\w)(\$[a-zA-Z_\x7f-
\xff][a-zA-Z0-9_\x7f-\xff]*)/', function(array $m): string {
                    return '<span style="color:#0000cb;">' .
(isset($m[0]) ? $m[0] : '') . '</span>';
                }, $line);

                $controlKeywords = ['if', 'echo', 'return',
'function'];
                foreach ($controlKeywords as $word) {
                    $line = preg_replace_callback('/\b' . $word .
'\b/', function(array $m) use ($word): string {
                        return '<span style="color:#008000;">' . $word
. '</span>';
                    }, $line);
                }

                $funcKeywords = ['defined', 'die'];
                foreach ($funcKeywords as $word) {
                    $line = preg_replace_callback('/\b' . $word .
```

```
'\b/', function(array $m) use ($word): string {
                        return '<span style="color:#0000cb;">' . $word
. '</span>';
                    }, $line);
                }
            }

            // Обычная замена индексов массивов на безопасный вывод в скобках
            $line = preg_replace_callback('/([()\{\}\[\]])/',
function(array $m): string {
                return '<span style="color:#008000;">' . (isset($m[0])
? $m[0] : '') . '</span>';
            }, $line);

            if (!empty($commentPart)) {
                $cleanComment = strip_tags($commentPart);
                $commentHtml = '<span style="color:#408080; font-
style:italic;">' . $cleanComment . '</span>';
                $line = str_replace('%%DOKU_COMMENT_PLACEHOLDER%%',
$commentHtml, $line);
            }
        }

        return $line;
    }
}
```

local.php

local.php

```
```

local.php

local.php

```
```