

SteamCMS - Multi-Era Steam Website Content Management System

A comprehensive CMS backend that recreates and manages multiple versions of steampowered.com from 2002-2009, featuring a custom template engine, historical content, and full admin interface.



Features

Multi-Era Theme System

- **8 different Steam eras:** 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009
- **Historical accuracy:** Recreated layouts based on actual archive.org snapshots
- **Evolution showcase:** From table-based layouts (2002-2005) to modern CSS (2006-2009)
- **Dynamic theme switching:** Seamlessly switch between eras

Custom Template Engine

- **Template inheritance:** Extend layouts with blocks
- **Dynamic content:** Variables, loops, conditionals
- **Include system:** Reusable template components
- **Theme-specific rendering:** Automatic theme-based template selection

Content Management

- **News Articles:** Add, edit, remove news for each era with historical context
- **FAQ System:** Categorized FAQ entries with full CRUD operations

- **Navigation Management:** Dynamic navigation links with drag-and-drop reordering
- **Logo Management:** Upload custom logos for each theme era
- **Media Library:** File upload and management system

Historical Data

- **Pre-loaded content:** Historical news articles for each Steam era
- **Era-appropriate navigation:** Theme-specific navigation structures
- **Sample FAQ entries:** Contextual help content for each time period
- **Default configurations:** Ready-to-use theme settings

Admin Interface

- **Modern admin panel:** Responsive design with intuitive navigation
- **Theme-aware management:** Switch between eras while managing content
- **Real-time preview:** View site changes instantly
- **User authentication:** Secure admin access with role management

Installation

Prerequisites

- **PHP 7.4+** with PDO MySQL extension
- **MySQL 5.7+** or MariaDB 10.2+
- **Web server** (Apache/Nginx)
- **Modern web browser**

Quick Installation

1. **Clone or download** the SteamCMS files to your web server

2. **Navigate to** `install.php` in your web browser
3. **Follow the installation wizard:**
 - Configure database connection
 - Create admin user account
 - Install historical data

Manual Installation

1. Database Setup:

```
sql CREATE DATABASE steamcms;
```

2. Configuration:

```
json { "host": "localhost", "username": "your_db_user", "password":  
"your_db_password", "database": "steamcms" }
```

3. Run Installation:

```
php require_once 'code/cms.php';  
  
$cms = new SteamCMS();  
$cms->install(['username' => 'admin', 'password' =>  
'steamcms', 'email' => 'admin@example.com']);
```

Usage

Admin Panel Access

- **URL:** `/admin/`
- **Default Login:** `admin` / `steamcms`
- **Features:** News, FAQ, Navigation, Media, Settings management

Frontend Access

- **URL:** `/index.php?theme=2004` (or any year 2002-2009)
- **Theme Switching:** Change the `theme` parameter to view different eras
- **Direct URLs:** Each theme has its own navigation structure

Theme Examples

- **2002:** `index.php?theme=2002` - Early beta with simple table layout
- **2004:** `index.php?theme=2004` - Half-Life 2 era with promotional design
- **2006:** `index.php?theme=2006` - Modern CSS store layout
- **2009:** `index.php?theme=2009` - Semantic HTML with advanced features



Architecture

Template Engine (`template_engine.php`)

```
$template = new TemplateEngine();
$template->setTheme('2004');
<span class="math-inline" style="display: inline;"><math
xmlns="http://www.w3.org/1998/Math/MathML"
display="inline"><mrow><mi>t</mi><mi>e</mi><mi>m</mi><mi>p</
mi><mi>l</mi><mi>a</mi><mi>t</mi><mi>e</mi><mo>&#x002212;</
mo><mo>&#x0003E;</mo><mi>a</mi><mi>s</mi><mi>s</mi><mi>i</
mi><mi>g</mi><mi>n</mi><msup><mo stretchy="false">&#x00028;</
mo><mi>&#x02032;</mi></msup><mi>n</mi><mi>e</mi><mi>w</
mi><msup><mi>s</mi><mi>&#x02032;</mi></msup><mo>&#x0002C;</mo></
mrow></math></span>news_articles);
echo $template->render('index');
```

Database Layer (`database.php`)

```
$db = new Database();
$db->createSchema();
<span class="math-inline" style="display: inline;"><math
xmlns="http://www.w3.org/1998/Math/MathML"
display="inline"><mrow><mi>n</mi><mi>e</mi><mi>w</mi><mi>s</
mi><mo>&#x0003D;</mo></mrow></math></span>$db->query("SELECT * FROM
news_articles WHERE theme = ?", ['2004']);
```

CMS Core (`cms.php`)

```
$cms = new SteamCMS();
$cms->setTheme('2004');
echo $cms->renderPage('index');
```

Models

- **NewsModel:** News article management
- **FAQModel:** FAQ categories and entries
- **NavigationModel:** Navigation link management
- **MediaModel:** File upload and media management

Directory Structure

```
steamcms/
├── admin/                # Admin panel
│   ├── css/             # Admin styles
│   ├── pages/           # Admin page controllers
│   ├── index.php        # Admin dashboard
│   └── login.php        # Authentication
├── code/                # Core PHP classes
│   ├── cms.php          # Main CMS class
│   ├── database.php     # Database layer
│   └── template_engine.php # Template system
├── themes/              # Theme templates
│   ├── 2002/            # 2002 Steam theme
│   ├── 2003/            # 2003 Steam theme
│   ├── 2004/            # 2004 Steam theme (default)
│   ├── 2005/            # 2005 Steam theme
│   ├── 2006/            # 2006 Steam theme
│   ├── 2007/            # 2007 Steam theme
│   ├── 2008/            # 2008 Steam theme
│   └── 2009/            # 2009 Steam theme
├── uploads/             # User uploaded files
├── index.php            # Frontend entry point
├── install.php          # Installation wizard
└── README.md            # Documentation
```

Theme Development

Creating a New Theme

1. Create theme directory:

```
php $template->createThemeStructure('custom_theme');
```

2. Main layout (themes/custom_theme/layouts/main.tpl):

```
```html
```

```
{block_placeholder "content"}
```

```
```
```

1. Index template (themes/custom_theme/index.tpl):

```
```html
```

```
{block "content"}
```

# Welcome to {\$site\_title}

```
{foreach newsarticlesas article}
```

```
{ $article.title }
```

```
{/foreach}
```

```
{/block}
```

```
```
```

Template Syntax

- **Variables:** {\$variable_name}
- **Loops:** {foreach <math xmlns="http://www.w3.org/1998/Math/MathML" display="inline"><mrow><mi>i</mi><mi>t</mi><mi>e</mi><mi>m</mi><mi>s</mi><mi>a</mi><mi>s</mi></mrow></math>item} ... {/foreach}
- **Conditionals:** {if \$condition} ... {/if}
- **Includes:** {include "partial_name"}
- **Blocks:** {block "block_name"} ... {/block}
- **Extends:** {extends "layout_name"}

Configuration

Database Settings

Configure in `config.json` or pass to constructor:

```
$config = [  
    'host' => 'localhost',  
    'username' => 'steamcms_user',  
    'password' => 'secure_password',  
    'database' => 'steamcms_db'  
];
```

Theme Settings

Manage through admin panel or programmatically:

```
$cms->setSetting('site_title', 'My Steam Site');  
$cms->setSetting('logo_url', '/uploads/custom_logo.png');
```

Historical Content

Pre-loaded News Articles

- **2002:** Steam beta launch, Counter-Strike 1.6
- **2003:** Public beta, Day of Defeat integration
- **2004:** Half-Life 2 release, Source Engine debut
- **2005:** Lost Coast, Steam platform updates
- **2006:** Episode One, Steam Store launch
- **2007:** Orange Box, Portal revolution

- **2008:** Left 4 Dead, 15M users milestone
- **2009:** Left 4 Dead 2, Steam Workshop beta

Navigation Evolution

- **Early years (2002-2005):** Simple text links
- **Store era (2006-2009):** Modern navigation with categories
- **Theme-specific:** Each era has appropriate navigation structure

Development

Adding New Features

1. Database changes:

```
php $db->getConnection()->exec("ALTER TABLE ...");
```

2. Model methods:

```
php class CustomModel { public function getCustomData($theme)
{ return <span class="math-inline" style="display: inline;"><math
xmlns="http://www.w3.org/1998/Math/MathML"
display="inline"><mrow><mi>t</mi><mi>h</mi><mi>i</mi><mi>s</
mi><mo>&#x02212;</mo><mo>&#x0003E;</mo><mi>d</mi><mi>b</
mi><mo>&#x02212;</mo><mo>&#x0003E;</mo><mi>q</mi><mi>u</mi><mi>e</
mi><mi>r</mi><mi>y</mi><mo stretchy="false">&#x00028;</mo><mi>"</
mi><mi>S</mi><mi>E</mi><mi>L</mi><mi>E</mi><mi>C</mi><mi>T</
mi><mo>&#x0002A;</mo><mi>F</mi><mi>R</mi><mi>O</mi><mi>M</
mi><mi>c</mi><mi>u</mi><mi>s</mi><mi>t</mi><mi>o</mi><msub><mi>m</
mi><mi>t</mi></msub><mi>a</mi><mi>b</mi><mi>l</mi><mi>e</mi><mi>W</
mi><mi>H</mi><mi>E</mi><mi>R</mi><mi>E</mi><mi>t</mi><mi>h</
mi><mi>e</mi><mi>m</mi><mi>e</mi><mo>&#x0003D;</mo><mo>&#x0003F;</
mo><mi>"</mi><mo>&#x0002C;</mo><mo stretchy="false">[</mo></mrow></
math></span>theme]); } }
```

3. Admin pages:

Create new file in `admin/pages/` following existing patterns

4. Template functions:

Add to `TemplateEngine` class for new template features

Extending Themes

1. **CSS customization:** Add to `themes/{theme}/css/`

2. **JavaScript:** Add to `themes/{theme}/js/`

3. **Images:** Add to `themes/{theme}/images/`

4. **Layouts:** Create new layouts in `themes/{theme}/layouts/`

Security

Best Practices

- **Input validation:** All user inputs are sanitized
- **SQL injection prevention:** PDO prepared statements
- **XSS protection:** HTML escaping in templates
- **Authentication:** Secure password hashing
- **File uploads:** Type and size validation

Recommended Hardening

1. **Remove** `install.php` after installation
2. **Set strong admin passwords**
3. **Restrict file permissions**
4. **Use HTTPS** in production
5. **Regular backups**



Troubleshooting

Common Issues

Database Connection Failed

- Check MySQL service is running
- Verify database credentials
- Ensure database exists

Template Not Found

- Check theme directory exists
- Verify template file names
- Check file permissions

Media Upload Failed

- Check `uploads/` directory permissions
- Verify PHP upload settings
- Check file size limits

Theme Not Loading

- Verify theme exists in database
- Check theme configuration
- Clear any template cache

Debug Mode

Enable error reporting during development:

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



Performance

Optimization Tips

1. **Enable PHP OPcache**
2. **Use database indexing**
3. **Implement template caching**
4. **Optimize images**
5. **Use CDN for static assets**

Caching

The template engine supports caching:

```
$template = new TemplateEngine('themes', 'cache');
```



Contributing

Development Setup

1. Fork the repository
2. Create feature branch
3. Follow coding standards
4. Add tests for new features
5. Submit pull request

Coding Standards

- **PSR-4** autoloading
- **PSR-12** coding style

- **Comprehensive comments**
- **Security-first approach**

License

This project is open source software licensed under the MIT License.

Credits

Author: MiniMax Agent

Historical Research: Based on Internet Archive snapshots of steampowered.com

Inspiration: Valve Corporation's Steam platform evolution

Support

For support, documentation, or feature requests:

1. **Check the troubleshooting section**
2. **Review the admin panel help**
3. **Consult the code comments**
4. **Check database logs**

SteamCMS - Preserving Steam's digital heritage through interactive content management.