MyCloudPDF Developer Guide

Technical Documentation for Software Engineers

Prepared by: Grok, xAI
Date: August 28, 2025
Version: 1.0

Project Name: MyCloudPDF - Open-Source PDF Management System

1 Introduction

MyCloudPDF is an open-source, self-hosted PDF management system replicating core features of CloudPDF.io. It allows users to upload PDFs, generate secure embeddable viewers, track engagement, and capture leads. This guide provides developers with the technical details needed to set up, maintain, or extend the system.

Tech Stack:

- **Backend:** Core PHP 7+, MySQL (PDO).
- **Frontend:** HTML, CSS, Bootstrap 5, JavaScript, jQuery.
- **Viewer:** PDF.js (Mozilla) for rendering PDFs.

2 System Architecture

The system follows a client-server model:

- **Backend:** PHP handles API endpoints (e.g., `api/upload.php`). MySQL stores user data, documents, tokens, views, and leads.
- **Frontend:** HTML/Bootstrap dashboard (`public/index.html`) with jQuery for AJAX interactions.
- **Viewer:** PDF.js in an iframe (`public/pdfjs/web/viewer.html`) streams PDFs via PHP (`api/get_pdf.php`).
- **File Storage:** PDFs stored in `uploads/` directory.

Directory Structure:

```
mycloudpdf/
|-- includes/
|   |-- config.php          # Database and path config
|-- api/
|   |-- register.php        # User registration
|   |-- login.php           # User login
|   |-- upload.php          # PDF upload
|   |-- documents.php       # List user documents
|   |-- get_pdf.php         # Stream PDF
|   |-- lead.php            # Lead capture
|   |-- check_doc.php       # Document settings
|   |-- track.php           # Track engagement
|-- public/
|   |-- index.html          # Dashboard
|   |-- viewer.js           # Embed viewer script
|   |-- style.css           # Custom styles
|   |-- pdfjs/              # PDF.js distribution
|-- uploads/                # PDF storage (writable)
```

3 Setup Instructions

1. **Install LAMP Stack:** Ensure PHP 7+, MySQL, and Apache are installed.
2. **Create Database:** Set up a MySQL database (e.g., `mycloudpdf`).
3. **Run Schema:** Execute the following SQL to create tables:

```sql
CREATE TABLE users (
    id INT AUTO_INCREMENT PRIMARY KEY,
    username VARCHAR(50) UNIQUE NOT NULL,
    password VARCHAR(255) NOT NULL,
    email VARCHAR(100) UNIQUE NOT NULL,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
CREATE TABLE documents (
    id INT AUTO_INCREMENT PRIMARY KEY,
    user_id INT NOT NULL,
    filename VARCHAR(255) NOT NULL,
    path VARCHAR(255) NOT NULL,
    is_public BOOLEAN DEFAULT TRUE,
    lead_capture BOOLEAN DEFAULT FALSE,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (user_id) REFERENCES users(id)
);
CREATE TABLE tokens (
    id INT AUTO_INCREMENT PRIMARY KEY,
    document_id INT NOT NULL,
    token VARCHAR(255) UNIQUE NOT NULL,
    expires DATETIME DEFAULT NULL,
    allow_download BOOLEAN DEFAULT FALSE,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (document_id) REFERENCES documents(id)
);
CREATE TABLE views (
    id INT AUTO_INCREMENT PRIMARY KEY,
    document_id INT NOT NULL,
    token_id INT DEFAULT NULL,
    ip VARCHAR(45) NOT NULL,
    view_time TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    pages_viewed TEXT DEFAULT NULL,
    FOREIGN KEY (document_id) REFERENCES documents(id),
    FOREIGN KEY (token_id) REFERENCES tokens(id)
);
CREATE TABLE leads (
    id INT AUTO_INCREMENT PRIMARY KEY,
    document_id INT NOT NULL,
    name VARCHAR(100),
    email VARCHAR(100),
    submitted_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (document_id) REFERENCES documents(id)
);
```

4. **Configure:** Edit `includes/config.php` with DB credentials and upload path.

```
<?php
define('DB_HOST', 'localhost');
define('DB_USER', 'root');
define('DB_PASS', '');
define('DB_NAME', 'mycloudpdf');
define('UPLOAD_DIR', __DIR__ . '/../uploads/');
function getDB() {
    $pdo = new PDO("mysql:host=" . DB_HOST . ";dbname=" .
       DB_NAME, DB_USER, DB_PASS);
    $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION
       );
    return $pdo;
}
?>
```

5. **Download PDF.js:** Get the latest stable PDF.js from https://mozilla.github.io/pdf.js/. Extract to `public/pdfjs/`.
6. **Set Permissions:** Ensure `uploads/` is writable by the server.
7. **Deploy:** Place files in web root; Access via `yourdomain/public/index.html`.

4 Module Breakdown

4.1 Authentication Module

- **Purpose:** User registration and login with session management.
- **Files:** `api/register.php`, `api/login.php`, `public/index.html`.
- **Logic:** Register hashes passwords (PHP `password_hash`); Login verifies and sets session.
- **Dependencies:** PHP PDO, jQuery (AJAX).
- **Key Code:** `api/register.php`

```
$stmt = $db->prepare("INSERT␣INTO␣users␣(username,␣password,␣
   email)␣VALUES␣(?,␣?,␣?)");
$hash = password_hash($data['password'], PASSWORD_DEFAULT);
$stmt->execute([$data['username'], $hash, $data['email']]);
```

4.2 Document Upload Module

- **Purpose:** Upload PDFs and store metadata.
- **Files:** `api/upload.php`, `public/index.html`.
- **Logic:** FormData upload; Save file to `uploads/`; Store path in DB.
- **Dependencies:** PHP file handling, MySQL.
- **Key Code:** `api/upload.php`

```
$file = $_FILES['pdf'];
$target = UPLOAD_DIR . basename($file['name']);
move_uploaded_file($file['tmp_name'], $target);
$stmt = $db->prepare("INSERT␣INTO␣documents␣(user_id,␣filename,␣
   path,␣is_public,␣lead_capture)␣VALUES␣(?,␣?,␣?,␣?,␣?)");
$stmt->execute([$_SESSION['user_id'], $file['name'], $target,
   isset($_POST['is_public']), isset($_POST['lead_capture'])]);
```

4.3 Viewer Module

- **Purpose:** Render PDFs securely in an iframe using PDF.js.
- **Files:** `public/viewer.js`, `public/pdfjs/web/viewer.html`, `api/get_pdf.php`.
- **Logic:** Embed code loads iframe; PHP streams PDF with token check.
- **Options:** `documentId`, `darkMode`.
- **Dependencies:** PDF.js.
- **Key Code:** `public/viewer.js`

```
function CloudPDF(config, element) {
    var src = '/pdfjs/web/viewer.html?file=/api/get_pdf.php?
       docid=' + config.documentId;
    if (config.darkMode) src += '#theme=dark';
    var iframe = document.createElement('iframe');
    iframe.src = src;
    iframe.style.width = '100%';
    iframe.style.height = '100%';
    element.appendChild(iframe);
}
```

4.4 Tracking Module

- **Purpose:** Log views and page interactions.
- **Files:** `api/track.php`, `public/pdfjs/web/viewer.html`.
- **Logic:** Hook PDF.js `pagechanging` event; POST to `track.php`.
- **Dependencies:** PDF.js eventBus, Fetch API.
- **Key Code:** `viewer.html` (append to body)

```
PDFViewerApplication.eventBus.on('pagechanging', function(evt) {
    fetch('/api/track.php', {
        method: 'POST',
        headers: {'Content-Type': 'application/json'},
        body: JSON.stringify({docid: docId, pages: [evt.
           pageNumber]})
    });
});
```

4.5 Lead Capture Module

- **Purpose:** Collect user info before PDF access if enabled.
- **Files:** `api/lead.php`, `api/check_doc.php`, `public/pdfjs/web/viewer.html`.
- **Logic:** Check document settings; Show form; Generate token on submit.
- **Dependencies:** Fetch API, MySQL.
- **Key Code:** `api/lead.php`

```
$stmt = $db->prepare("INSERT␣INTO␣leads␣(document_id,␣name,␣
   email)␣VALUES␣(?,␣?,␣?)");
$stmt->execute([$docId, $data['name'], $data['email']]);
$token = bin2hex(random_bytes(16));
$tokenStmt = $db->prepare("INSERT␣INTO␣tokens␣(document_id,␣
   token)␣VALUES␣(?,␣?)");
$tokenStmt->execute([$docId, $token]);
```

4.6 Dashboard Module

- **Purpose:** User interface for managing documents.
- **Files:** `public/index.html`, `api/documents.php`.
- **Logic:** Post-login, list documents; Generate embed codes.
- **Dependencies:** Bootstrap, jQuery.
- **Key Code:** `index.html`

```
$.get('/api/documents.php', function(res) {
    let html = '<ul class="list-group">';
    res.forEach(doc => {
        html += `<li class="list-group-item">
            ${doc.filename} - Embed Code: <code>&lt;div id="
               viewer" style="width:800px;height:500px"&gt;&lt;/
               div&gt;
            &lt;script src="/viewer.js"&gt;&lt;/script&gt;
            &lt;script&gt;CloudPDF({documentId: '${doc.id}',
               darkMode: true}, document.getElementById('viewer
               '));&lt;/script&gt;</code>
        </li>`;
    });
    $('#documents').html(html);
}, 'json');
```

5 API Endpoints

- **POST /api/register.php**: Register user. Input: `username, email, password`. Output: JSON `{message}`.
- **POST /api/login.php**: Login user. Input: `username, password`. Output: JSON `{success, message}`.

- **POST /api/upload.php**: Upload PDF. Input: `pdf` (file), `is_public, lead_capture`. Output: JSON `{message}`.
- **GET /api/documents.php**: List user documents. Output: JSON array of `{id, filename}`.
- **GET /api/get_pdf.php?docid=&token=**: Stream PDF. Requires valid `docid` and optional `token`.
- **GET /api/check_doc.php?docid=**: Check document settings. Output: JSON `{lead_capture}`.
- **POST /api/lead.php**: Submit lead. Input: `docid, name, email`. Output: JSON `{token}`.
- **POST /api/track.php**: Track engagement. Input: `docid, pages`. Output: JSON `{success}`.

6 Security Considerations

- Use HTTPS to encrypt traffic.
- Validate/sanitize all inputs (e.g., file types, SQL injection).
- Implement CSRF tokens for forms (not included in MVP).
- Restrict PDF downloads via PHP streaming; Disable right-click in PDF.js (customize `viewer.html`).
- Add session timeout and secure cookies.

7 Testing Guidelines

- **Unit Tests:** Check each API endpoint with tools like Postman.
- **Integration Tests:** Verify upload → embed → view flow.
- **Security Tests:** Test for SQL injection, XSS, unauthorized access.
- **Tools:** PHPUnit for PHP; Jest for JS (optional).

8 Contribution Guidelines

- **Repository:** Host on GitHub; Use branches for features.
- **Code Style:** Follow PSR-12 for PHP; Prettier for JS/HTML/CSS.
- **Pull Requests:** Include tests; Document changes.
- **Issues:** Report bugs with steps to reproduce.
- **Extensions:** Add features like analytics, token expiry, or CMS plugins.

9 Troubleshooting

- **PDF not loading:** Check `pdfjs/` path; Verify `api/get_pdf.php`.
- **Upload fails:** Ensure `uploads/` is writable; Check PHP upload limits.
- **Database errors:** Verify `config.php` credentials; Run schema.
- **Viewer issues:** Update PDF.js; Debug JS console.