

OpenMark

PDF Viewer with Annotations

Features

- * PDF Visualization - View PDF documents in your browser
- * Sticky Notes - Create virtual post-it notes on documents
- * Text Highlighting - Highlight text with customizable colors
- * Secure Authentication - Multiple auth backends supported
- * Usage Statistics - Track document views and annotations
- * History Tracking - View consultation history
- * Plugin System - Extensible architecture

Quick Start

1. `python -m venv venv`
2. `source venv/bin/activate`
3. `pip install -r requirements.txt`
4. `python run.py`
5. Open `http://localhost:5000`

Default Credentials

Username: admin

Password: admin123

API Endpoints

POST /api/authenticate - User authentication

POST /api/requestDocument - Request PDF access

GET /api/viewDocument - View PDF with annotations

POST /api/saveAnnotations - Save annotations

GET /api/getAnnotations - Get annotations

GET /api/statistics - Usage statistics

GET /api/history - Consultation history

POST /api/logout - Logout

This is a sample PDF for testing OpenMark

Page 2 - Architecture

OpenMark follows a modular architecture with a plugin system for extensibility.

Components:

- Flask Backend: Handles API requests and serves the web interface
- Plugin Manager: Loads and manages authentication, PDF source, and annotations plugins
- PDF Viewer: JavaScript-based viewer using PDF.js library
- Annotations Layer: Overlay for notes and highlights on PDF pages

The application can be deployed using Docker for easy containerization.

Page 3 - Plugin Development

OpenMark supports three types of plugins:

1. Authentication Plugins

- Handle user authentication
- Built-in: Local file-based auth

2. PDF Source Plugins

- Retrieve PDF documents
- Built-in: HTTP/HTTPS and local file

3. Annotations Plugins

- Store and retrieve annotations
- Built-in: Local JSON and MongoDB

Each plugin type has a base class that defines the required interface.

Create custom plugins by extending these base classes.