

# VOiC - Virtual Office in the Cloud

Akhil Karri  
karr8512@vandals.uidaho.edu  
University of Idaho  
Moscow, ID, USA

November 2023

## Abstract

The rise of cloud computing has paved the way for innovative solutions in digital document management, particularly for the legal sector. The Virtual Office in Cloud (VOiC) project has been developed from the ground up using the PHP and MySQL framework, presenting a robust platform that operates seamlessly on local systems. Central to VOiC's design is its commitment to security, with SQL injection shields, secure session management, email filtration, and password hashing mechanisms ensuring the safeguarding of user data. The interface of VOiC is a model of modern, minimalistic design, enhancing user experience without compromising functionality. With a two-tiered user privilege system that is scalable for additional levels, VOiC facilitates nuanced access control. The integration of CKEditor provides users with a rich editing environment, accompanied by the capability to store document metadata for extended document management. Search functionality is versatile, offering both keyword-based and attribute-based retrieval to cater to varied search intents. Moreover, the platform delineates user roles, allowing for the creation and editing of documents by general users while reserving document removal privileges for administrators. This paper expounds on the creation and evolution of VOiC, illustrating its potential as a benchmark for legal document management platforms in the cloud computing domain.

## 1 Keywords

Cloud Computing, Document Management System, Legal Document Automation, Web Application Security, User Interface Design, Access Control, PHP and MySQL Development, Rich Text Editing, Data Protection, Search Algorithms, User Authentication, Encryption, Database Design, Software Architecture.

## 2 Introduction

The evolution of digital document management has been markedly accelerated by the intersection of legal requirements and cloud-based technologies. The Virtual Office in Cloud (VOiC) project, developed from the ground up, harnesses this momentum to offer a specialized platform for legal document handling. At its core, VOiC is an innovative web-based application that empowers legal professionals to create, edit, and securely manage their documentation online with unprecedented ease and flexibility. One of VOiC's groundbreaking features is an advanced text editor equipped with auto-complete functionality, designed to facilitate the drafting process by suggesting and auto-filling legal terms and phrases. This intelligent feature not only increases efficiency but also helps in minimizing errors, ensuring the accuracy and consistency of legal documents. In addition to providing cutting-edge editing tools, VOiC takes a proactive stance on security. It is fortified with several layers of protection against common web vulnerabilities, including SQL injection and session hijacking, while offering robust encryption for user credentials. The interface of VOiC marries aesthetics with functionality, delivering a user experience that is both visually appealing and intuitively navigable. This introductory section delineates the VOiC project's vision, challenges surmounted during its development, and the strategic approach to its deployment. It further describes the project's methodology and technological innovation, setting the stage for a comprehensive exploration of the system's architecture, security framework, and feature set.

### 2.1 Project Motivation

VOiC was conceived in response to the legal industry's urgent need for a dynamic document management system that can adapt to the evolving landscape of digital workflows. This need has been particularly highlighted by the transition to remote work arrangements, where traditional document handling poses significant limitations in terms of accessibility and security.

## 2.2 Objective

The primary goal of VOiC is to deliver a cloud-based document management platform that balances functional excellence with operational security. It aspires to redefine the standards of legal documentation processes, making them more streamlined, secure, and user-oriented.

## 2.3 Contributions

The main contributions of this report are as follows:

- Introduction of an auto-complete feature in the text editor to improve the drafting experience for legal documents.
- Creation of a sleek, minimalistic interface that complements the advanced features without overwhelming the users.
- Deployment of comprehensive security measures to preserve data integrity and privacy.
- Provision of a versatile search mechanism, accommodating both fuzzy keyword searches and precise attribute-based queries.
- A showcase of applying PHP and MySQL technologies effectively in a fully functional cloud-based application.

## 3 Related Work

The concept of enhancing office management through virtual systems is a progressive idea that has seen various iterations and improvements over decades. The earliest inklings of an electronic office system trace back to designs proposed in 1985, with subsequent works recognizing the need for collaborative document management systems as early as 1996. The maturation of cloud computing technologies has enabled significant enhancements over these initial proposals, introducing sophisticated features like metadata tagging and document version control, blockchain-based architectures for preventing data silos, and role-based access in web applications. VOiC stands on the shoulders of these advancements by not only integrating metadata tags, stakeholder access protection, and role-based sharing but also by refining the approach to data discovery. The system uses graph-based searching and HTML for document representation, ensuring that its design remains platform-agnostic as befits a cloud application. Moreover, VOiC utilizes a modern front-end library to enhance the cross-device user experience. The foundational features of VOiC, including user search, account creation, and document-sharing capabilities, were initially laid out by Kugler and Carroll. Their work established the system's user permission schema and introduced the 'danger zone' feature for critical account and document management operations. This groundwork allowed subsequent developments to incorporate more sophisticated features into VOiC. The work of Dr. Jamil on semantic query engines offers a road map for feature integration within digital libraries, providing a solid conceptual framework for VOiC's capabilities in graph visualization and interface design. Additionally, research efforts like those documented in PloS ONE on automating meta-tag search results lend insight into potential methodologies VOiC could employ to refine tag search functionality and optimize match retrieval, drawing parallels between patent searches and legal document queries. In a similar vein to PloS ONE's patent verification process, VOiC's G-Search module aims to verify and reference previous judicial cases to inform current legal proceedings. The research's approach to tag comparison using cosine similarity measures presents an intriguing possibility for enhancing VOiC's search algorithm to deliver more relevant results.

## 4 Background

The architecture of the Virtual Office in Cloud (VOiC) is built on a foundation of technologies each serving distinct aspects of the system. From server management to front-end development, VOiC integrates a suite of tools that represents the best practices in modern web application development. The proliferation of cloud computing has revolutionized numerous industries by enabling the delivery of hosted services over the Internet. Cloud services offer vast storage capabilities, on-demand resource scalability, and high availability, which are invaluable features for modern digital solutions. The domain of legal document management is one such area that has benefited from cloud computing, as the need for secure, accessible, and reliable document storage and retrieval has become increasingly prominent.

## 4.1 PHP: Versatile Server-Side Scripting

PHP's role in web development is undeniably significant, with its origins dating back to the early days of dynamic web content. It has been the backbone of countless web applications, providing the flexibility to create custom server-side logic. PHP's syntax and language constructs, which are continually improved with each version release, offer developers an efficient way to write clean and reusable code. For VOiC, PHP facilitates a responsive user experience by handling session management, dynamic content rendering, and database interactions robustly, thanks to its rich set of built-in functions and advanced features for handling various HTTP request and response types.

## 4.2 HTML5 and CSS3: Crafting the Web's Look and Feel

HTML5 extends the capabilities of its predecessors by introducing features that once required external plugins or complex scripting. Its semantic elements like `<article>`, `<section>`, `<nav>`, and `<footer>` allow for more readable and accessible code, an aspect that VOiC capitalizes on to structure its content effectively. CSS3, with its advanced selectors, custom fonts, shadows, gradients, and animation features, enables VOiC to craft a distinctive and engaging user interface without compromising on performance or compatibility across modern browsers.

## 4.3 Bootstrap Framework: A Pillar of Responsive Design

Bootstrap's contribution to responsive web design is transformative, providing a comprehensive set of styling options that adhere to responsive design principles. VOiC's utilization of Bootstrap allows for a mobile-first approach, ensuring the application's compatibility with a broad spectrum of devices and screen sizes. Bootstrap's grid system and responsive utility classes make the layout and spacing adjustments effortless, which is key for VOiC's document editing and presentation features that require adaptability to varying display conditions.

## 4.4 MySQL: The Standard for Database Management

MySQL's proven track record in providing a stable and efficient database solution has made it the go-to choice for managing complex datasets and transactional data. In VOiC, MySQL's role is integral to storing user profiles, document histories, and meta-information that are pivotal for the application's search and retrieval features. The database's support for advanced query operations and transactional consistency ensures that VOiC can manage concurrent user operations, a critical requirement for a cloud-based collaborative environment.

## 4.5 XAMPP Server: Development and Testing Made Easy

The development process for web applications can be cumbersome without the right set of tools. XAMPP provides a full-fledged local server environment that mimics live server behavior, reducing the friction in the development lifecycle. Its package includes MariaDB as a drop-in replacement for MySQL, which VOiC uses during development for database operations testing. By bundling Apache and PHP, XAMPP allows VOiC developers to test new features in a controlled environment, ensuring that updates do not disrupt the application's functionality.

## 4.6 Apache NetBeans: A Comprehensive IDE for Web Development

Apache NetBeans IDE, known for its cross-platform support and comprehensive tooling, empowers VOiC developers with code generators, editors, and debugging tools that are tailored for PHP development. It supports the entire development cycle from writing code to debugging and deploying applications. Its project management capabilities enable VOiC developers to work within a unified environment, simplifying code organization and ensuring consistency across the project's development team.

## 4.7 GitHub: Distributed Version Control and Teamwork

GitHub has revolutionized the way developers collaborate on code. It is not just a repository to store code; it is a platform for tracking changes, discussing improvements, and managing tasks related to software development. For VOiC, GitHub facilitates the adoption of agile development methodologies, allowing multiple streams of development to proceed in parallel. Features like issue tracking, milestones, and GitHub Actions for automated workflows are integral to maintaining VOiC's development pace and quality.

## 4.8 Evolution of Document Management Systems

Document Management Systems (DMS) have undergone significant evolution since their inception. Initial systems were primarily focused on digitizing paper-based workflows to reduce physical storage needs and improve retrieval efficiency. As technology progressed, the emphasis shifted towards collaborative features, allowing multiple users to work on documents simultaneously from disparate locations.

## 4.9 Security Considerations in DMS

Security in document management involves not only safeguarding against unauthorized access but also ensuring that documents are tamper-proof and verifiable. The legal industry's reliance on the integrity and confidentiality of documents necessitates stringent security measures. Over time, DMS security has evolved to include advanced user authentication, access controls, audit trails, and encryption standards, among other features.

## 4.10 Collaborative Features and Accessibility

With the emergence of web technologies, DMS have embraced more interactive and user-friendly interfaces. Collaboration tools have become a staple in these systems, enabling real-time editing, commenting, and version control. Accessibility has also been a key focus, with systems ensuring that documents are accessible from a variety of devices and platforms, reflecting the increasing mobility of users.

## 4.11 The Role of PHP and MySQL in Web Development

The use of PHP and MySQL in developing web applications has been prevalent due to their open-source nature, wide support, and flexibility. PHP's server-side scripting capabilities, when combined with MySQL's efficient data storage and retrieval, make them a powerful duo for creating robust web applications. VOiC leverages these technologies to manage user interactions and document storage efficiently.

## 4.12 Advances in Text Editing Technology

Text editing technology has seen advancements with rich text editors like CKEditor providing users with word processor-like features in web browsers. The auto-complete functionality in text editors, a feature now incorporated into VOiC, exemplifies the intelligent user assistance that modern editors provide to enhance user productivity and experience. In sum, the background of VOiC is rooted in the historical development of DMS and cloud computing, enriched by advances in security, collaboration, and web development practices. This project leverages these advancements to deliver a platform designed to meet the high standards of legal professionals in document management.

## 4.13 Putting it all together

The selection and assembly of these technologies for VOiC's development are guided by principles of modularity, security, and user experience design. Each technology has been chosen not only for its individual merits but also for how seamlessly it integrates with others. The convergence of these technologies provides a platform that is greater than the sum of its parts, ensuring VOiC's position as a versatile and powerful tool for legal professionals.

# 5 Environment Setup

Properly establishing the environment for the Virtual Office in Cloud (VOiC) is crucial for both development and production phases. This section provides a comprehensive guide to configuring the development environment, managing source control, and deploying the application to a live server.

## 5.1 Local Development with XAMPP

XAMPP is a free and open-source cross-platform web server solution that provides an easy-to-install Apache distribution containing MariaDB, PHP, and Perl. VOiC developers use XAMPP to emulate the application's production environment on their local machines.

### Detailed XAMPP Installation:

1. Obtain the latest version of XAMPP suitable for the operating system.
2. Install XAMPP and ensure that all components, including Apache, MySQL, and PHP, are selected.

3. Launch the XAMPP Control Panel to start the Apache and MySQL services, confirming their operational status via the control panel's indicators.
4. Access the local PHPMyAdmin through a web browser to create a new database for VOiC, importing the schema from the provided SQL files.

**Common Troubleshooting:** Developers may encounter port conflicts if other services are using default HTTP ports. Resolving this involves configuring XAMPP to use alternative ports or shutting down conflicting services.

## 5.2 Version Control with Git and GitHub

Version control is handled through Git, with GitHub serving as the central repository for the VOiC project. The setup ensures that all team members can work concurrently on different features without conflict. **Initial Repository Setup:**

```
git clone https://github.com/AKHIL00124/CS360-VOiC.git
git checkout -b development
git push -u origin development
```

This sets up a local Git environment that tracks the remote 'development' branch, allowing changes to be merged into the main project upon peer review.

## 5.3 Setting up the XAMPP Environment

XAMPP is a popular PHP development environment that is easy to install and use. It is a free and open-source cross-platform web server solution stack package that includes the Apache web server, MariaDB (a fork of MySQL), and interpreters for PHP and Perl scripts.

### Installation Steps:

1. Download XAMPP from the official website, selecting the version compatible with your operating system.
2. Open the downloaded file and run the installer. During the installation, you may choose which components to install. It is recommended to install Apache, MySQL, and PHP.
3. Follow the prompts to complete the installation. By default, XAMPP is installed in the C:\xampp directory on Windows or in the /Applications/XAMPP directory on Mac.
4. After installation, launch the XAMPP Control Panel, which provides control over each component of the stack. Start the Apache and MySQL modules.

### Accessing XAMPP's PHPMyAdmin:

- With the Apache and MySQL modules running, open a web browser and go to <http://localhost/phpmyadmin>. This accesses the PHPMyAdmin interface, a web-based tool for managing MySQL databases.

## 5.4 Running the XAMPP Server

1. To start the Apache server, open the XAMPP Control Panel and click on the 'Start' button next to Apache.
2. To test if Apache is running correctly, navigate to <http://localhost> in a web browser. If it is running, you will see the XAMPP dashboard.

## 5.5 Importing a MySQL Database

### Via PHPMyAdmin:

- In PHPMyAdmin, select the 'Import' tab.
- Click on 'Choose File' and navigate to the location of your .sql database file.
- Select the file and click on 'Go' at the bottom of the page to import the database.
- Import the VOiC.sql file in the cloned GitHub folder.

This process will create a new database and populate it with the data from your SQL file. It's a straightforward way to set up your application's database locally and is essential for applications that rely on a database, such as VOiC.

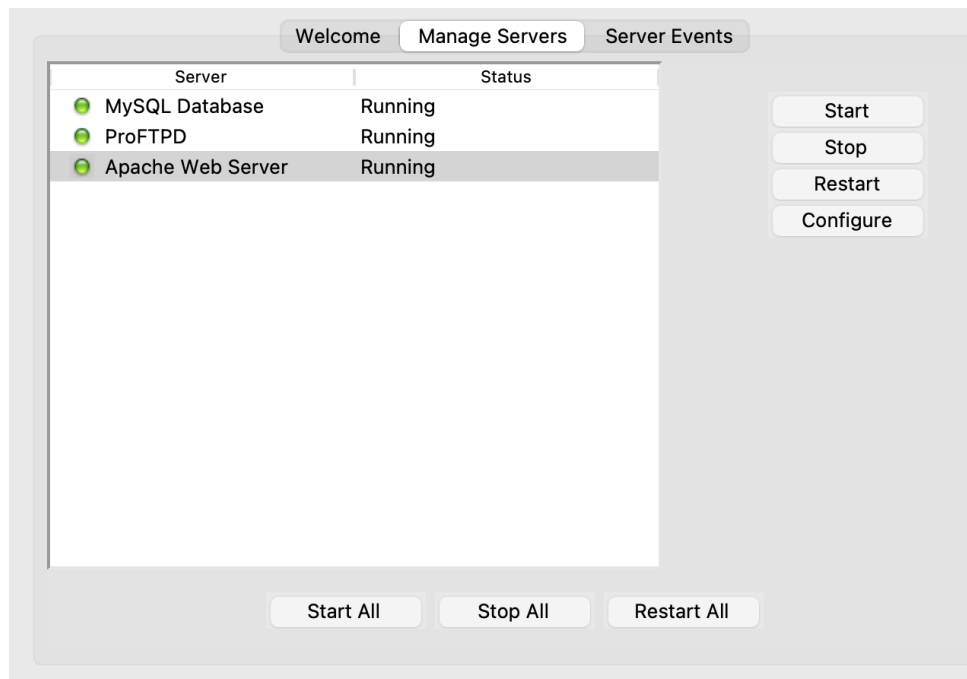


Figure 1: Start the XAMPP Server

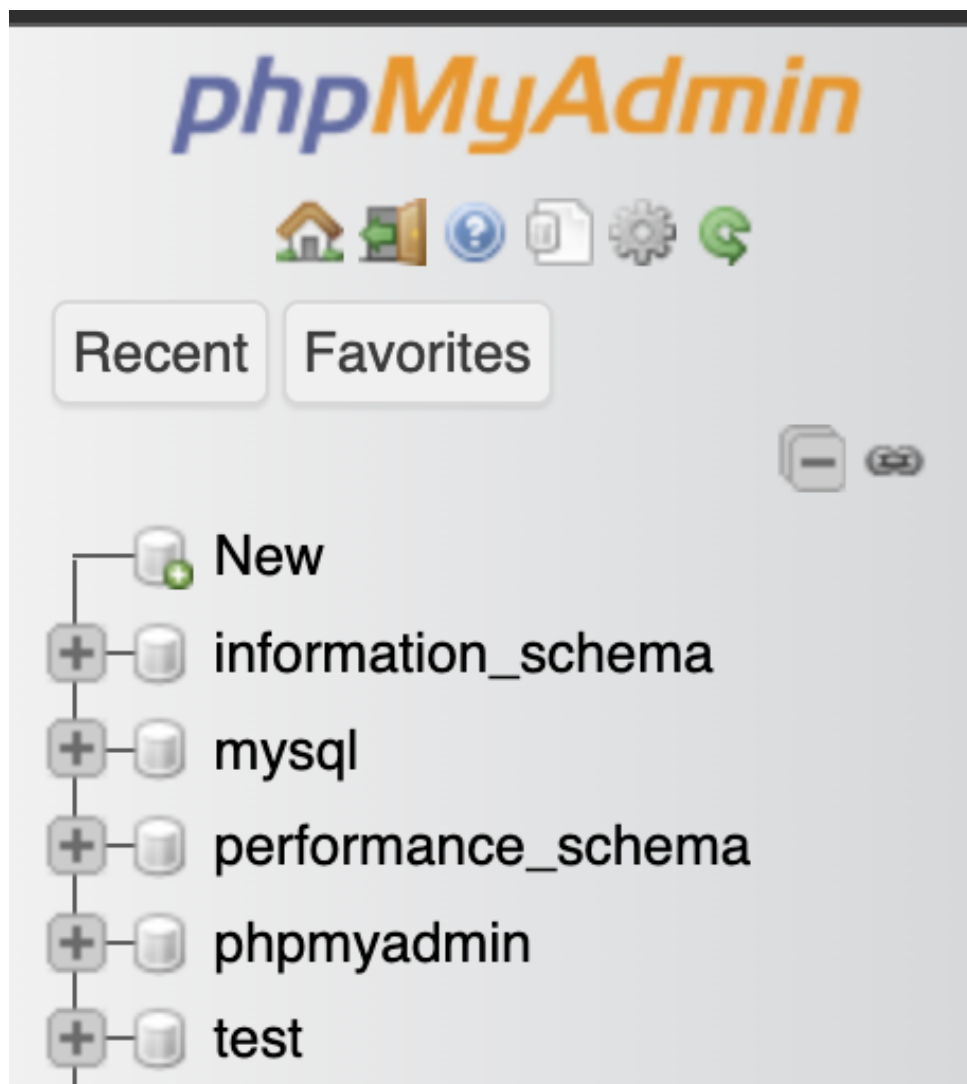


Figure 2: Create New DB

## Databases

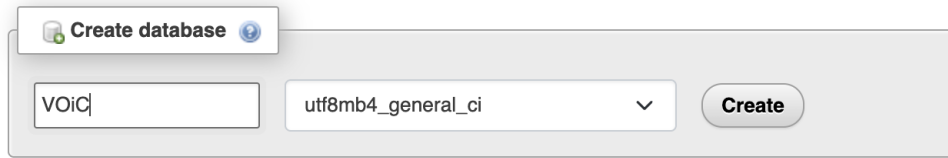


Figure 3: Name the DB 'VOiC'



Figure 4: Import the VOiC.sql from the cloned repo folder

### 5.6 Apache NetBeans IDE Setup for PHP

Apache NetBeans offers a rich set of features tailored to PHP development, including code highlighting, syntax validation, debugging, and version control integration. To leverage NetBeans for the VOiC project, developers need to set up their environment to support PHP and related web technologies effectively. **NetBeans Installation:**

1. Download the latest version of Apache NetBeans IDE that supports PHP from the official website.
2. Run the installer and follow the instructions, making sure to select the PHP development pack.
3. Once installed, launch Apache NetBeans and configure it to recognize the XAMPP PHP interpreter and the MySQL server.

#### Configuring PHP Environment in NetBeans:

- Go to 'Tools' - 'Options' - 'PHP' and set the PHP interpreter path to the PHP executable within the XAMPP directory.
- In the same options panel, configure the 'php' file used by NetBeans to match the one used by XAMPP.

#### Importing the VOiC Project:

1. Open NetBeans and select 'File' - 'New Project' to start the project setup wizard.
2. Copy the cloned file into htdocs of the XAMPP folder
3. Choose 'PHP Application with Existing Sources' and navigate to the directory where the VOiC project is located.
4. Set the 'Run Configuration' to local web server and specify the project URL to match the virtual host configuration set in XAMPP for VOiC.
5. If the project uses a database, configure the 'Database' section in NetBeans to connect to the MySQL server running on XAMPP, using the VOiC database credentials.

**Debugging Configuration:** To configure debugging, set up xdebug in the PHP configuration used by XAMPP and then configure NetBeans to listen for the PHP debugging connections.

- Verify that xdebug is correctly installed and configured in the 'php' file used by XAMPP.
- In NetBeans, go to 'Project Properties' - 'Run Configuration' and set the Debugging options, ensuring that the debugger port matches the one specified for xdebug.
- Use NetBeans' built-in debugging controls to start, stop, and manage breakpoints within the application.

**Version Control Integration:** NetBeans integrates directly with Git for version control. Set up the Git repository within NetBeans by:

1. Opening the 'Team' tab and selecting 'Git' - 'Clone' to clone an existing GitHub repository.

2. Enter the repository URL, user credentials, and select the local folder for the project.
3. Once cloned, the project is ready for development. Commit and push changes directly from NetBeans to the GitHub repository.

These steps provide VOiC developers with a well-structured development environment using Apache NetBeans, which streamlines the coding, testing, and debugging processes integral to robust PHP development.

## 6 Components of VOiC

The Virtual Office in Cloud (VOiC) is a sophisticated web application designed for managing legal documents in a cloud environment. The application is composed of several key components that work in harmony to offer a seamless user experience and robust backend functionality.

### 6.1 Front-end Components

**HTML/CSS/JavaScript:** The front-end user interface is built using HTML5 for structuring the web content, CSS3 for styling, and JavaScript for dynamic interactions. HTML5 provides a semantic structure, making the content accessible and SEO-friendly. CSS3 is used for responsive design, ensuring the application is mobile-responsive. JavaScript, possibly with libraries such as jQuery, handles the client-side scripting for interactive features. **Bootstrap:** Bootstrap framework is used for designing a responsive and mobile-first front-end architecture. It provides a grid system and pre-designed components like modals, dropdowns, and buttons that speed up the development process.

### 6.2 Back-end Components

**PHP:** As the server-side scripting language, PHP handles all the business logic of the application, including user authentication, document processing, and database interactions. **MySQL Database:** MySQL serves as the relational database management system for storing all the application data, including user profiles, document metadata, and activity logs. **Apache Server:** The Apache HTTP Server is utilized to serve the VOiC web application, handle HTTP requests, and manage secure communications.

### 6.3 Development and Deployment Tools

**XAMPP:** XAMPP is an open-source cross-platform web server solution stack package, used here as a local development environment for running the Apache server, MySQL database, and PHP interpreter. **Apache NetBeans:** Apache NetBeans is the integrated development environment (IDE) of choice for the project, offering robust PHP development tools and facilitating efficient coding practices. **Git and GitHub:** Version control is managed via Git, with GitHub serving as the remote repository for code collaboration, feature branching, issue tracking, and release management.

### 6.4 Security Components

**SSL/TLS:** Secure Sockets Layer (SSL) and Transport Layer Security (TLS) are used to encrypt data transmitted over the network, protecting sensitive user data and authentication credentials. **Content Security Policy (CSP):** CSP is implemented to prevent common security vulnerabilities such as Cross-Site Scripting (XSS) and data injection attacks.

### 6.5 Additional Libraries and Plugins

**CKEditor:** CKEditor 5 is integrated as a rich text editor to provide users with word processor-like features for document editing within the web browser. **jQuery:** jQuery may be used to simplify the HTML document traversal, event handling, animation, and Ajax interactions.

Each component is carefully chosen and configured to meet the specific needs of managing legal documents securely and efficiently in the cloud. The following sections will delve into the specific implementation details and the interactions between these components.



```

<?php

if (empty($_POST["name"])) {
    die("Name is required");
}

if ( ! filter_var($_POST["email"], FILTER_VALIDATE_EMAIL)) {
    die("Valid email is required");
}

if (strlen($_POST["password"]) < 8) {
    die("Password must be at least 8 characters");
}

if ( ! preg_match("/[a-z]/i", $_POST["password"])) {
    die("Password must contain at least one letter");
}

if ( ! preg_match("/[0-9]/", $_POST["password"])) {
    die("Password must contain at least one number");
}

if ($_POST["password"] !== $_POST["password_confirmation"]) {
    die("Passwords must match");
}

if (empty($_POST["role"])) {
    die("UserType should be selected");
}

$password_hash = password_hash($_POST["password"], PASSWORD_DEFAULT);

$mysqli = require __DIR__ . "/database.php";

$sql = "INSERT INTO user (name, email, password_hash, role)
VALUES (?, ?, ?, ?)";

$stmt = $mysqli->stmt_init();

if ( ! $stmt->prepare($sql)) {
    die("SQL error: " . $mysqli->error);
}

$stmt->bind_param("sssi",
    $_POST["name"],
    $_POST["email"],
    $password_hash, $_POST["role"]);

if ($stmt->execute()) {

    header("Location: Log-In.php");
    exit;

} else {
    if ($mysqli->errno === 1062) {
        die("email already taken");
    } else {
        die($mysqli->error . " " . $mysqli->errno);
    }
}
}

```

Figure 5: ProcessNewUsers.php

## 6.6 Password Hashing

**Secure Authentication:** VOiC implements secure authentication practices, including password hashing. Passwords are never stored as plain text in the database. Instead, they are hashed using a strong hashing algorithm.

**Hashing Algorithm:** The PHP `password_hash()` function is used, which provides a secure bcrypt hash of the password. This function automatically generates a salt for each password, making every hash unique. When users log in, the `password_verify()` function checks the entered password against the stored hash to authenticate the user.

## 6.7 SQL Injection Prevention

**Prepared Statements:** To prevent SQL injection, one of the most common security vulnerabilities in web applications, VOiC uses prepared statements with parameterized queries. This approach ensures that SQL commands are separated from the data, negating the possibility of malicious SQL code execution.

**Input Validation:** All user inputs are sanitized and validated on both the client and server sides before being used in any database queries. This not only prevents SQL injection but also helps maintain data integrity.

```

<?php
$is_invalid = false;
if ($_SERVER["REQUEST_METHOD"] === "POST") {
    $mysqli = require __DIR__ . "/database.php";
    $sql = sprintf("SELECT * FROM user
        WHERE email = '%s'",
            $mysqli->real_escape_string($_POST["email"]));
    $result = $mysqli->query($sql);
    $user = $result->fetch_assoc();
    if ($user) {
        if (password_verify($_POST["password"], $user["password_hash"])) {
            session_start();
            session_regenerate_id();
            $_SESSION["role"] = $user["role"];
            $_SESSION["user_id"] = $user["id"];
            if ($user["role"] == 2){
                header("Location: admin.php");
                exit;
            }
            if ($user["role"] == 1){
                header("Location: home.php");
                exit;
            }
        }
    }
    $is_invalid = true;
}
?>

```

Figure 6: SQL Injection Protection and Password hashing

## 6.8 Custom CKEditor Auto-Complete Feature

**Enhancing the Editing Experience:** VOiC’s integration of CKEditor is enhanced with a custom auto-complete feature, specifically tailored for legal document editing. This feature provides real-time suggestions and auto-completion for legal terminology and phrases, aiding in the accuracy and consistency of document creation. **Implementation Details:** The auto-complete functionality is achieved through a combination of CKEditor’s plugin system and a predefined lexicon of legal terms. As users type, the editor dynamically matches the input against this lexicon and suggests relevant completions. **Security Considerations:** The custom plugin for CKEditor is developed with security in mind, ensuring that dynamic content generation does not introduce any client-side vulnerabilities such as Cross-Site Scripting (XSS). User inputs that are part of the auto-complete suggestions are handled in a way that prevents any executable code from being inserted into the document. The security infrastructure of VOiC is designed to be robust and resilient, utilizing best practices and current standards in web application security. By prioritizing the protection of user data and system integrity, VOiC establishes a trusted platform for legal professionals.

## 6.9 PHP Processing

PHP is the scripting language at the heart of VOiC, providing the server-side logic necessary for handling user interactions, processing document data, and performing CRUD operations on the database. **Scripts and Execution Flow:** The PHP codebase includes scripts for authentication, document processing, user session management, and various AJAX endpoints that facilitate asynchronous operations. **Schemas and Relationships:** Carefully designed table schemas ensure data normalization, while foreign key constraints maintain the relational integrity between users, documents, and metadata records.

## 6.10 Bootstrap and Front-end Framework

Bootstrap underpins the front-end framework of VOiC, ensuring that the user interface is responsive and accessible across devices. Custom themes and overrides are applied to align with the VOiC branding guidelines. **Responsive Design:** The use of Bootstrap’s grid system ensures that VOiC’s layout is responsive, providing an optimal viewing experience for both mobile and desktop users.

## 6.11 SQL Injection Prevention

**Prepared Statements:** To prevent SQL injection, one of the most common security vulnerabilities in web applications, VOiC uses prepared statements with parameterized queries. This approach ensures that SQL commands are separated from the data, negating the possibility of malicious SQL code execution. **Input Validation:** All user inputs are sanitized and validated on both the client and server sides before being used in any database queries. This not only prevents SQL injection but also helps maintain data integrity.

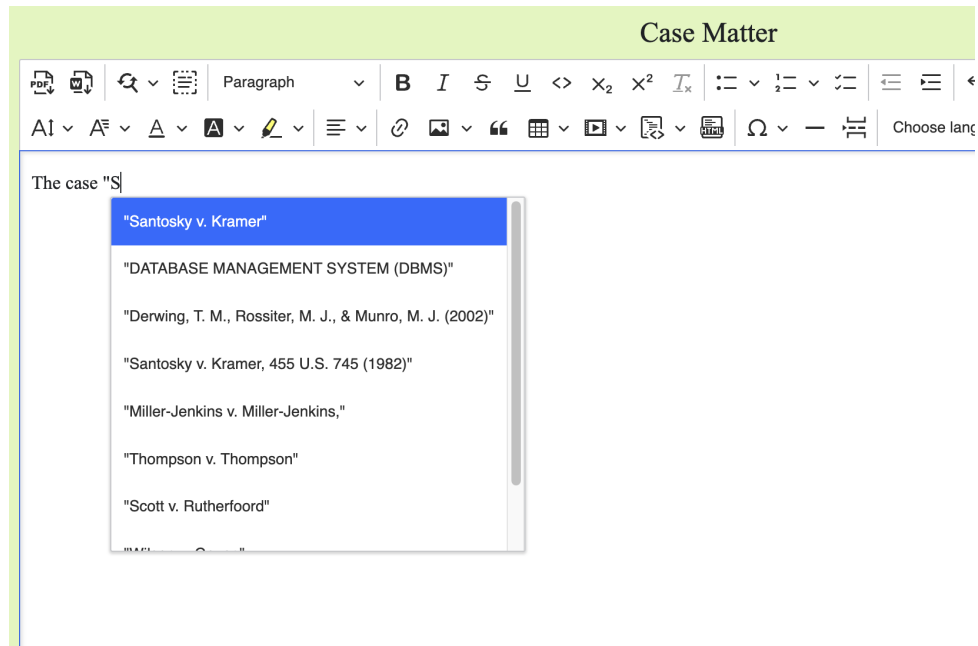


Figure 7: Auto-Complete Feature

## 6.12 Role-Based Access Control in VOiC

VOiC implements a role-based access control (RBAC) system to manage users' interactions with the application based on their roles and privileges. This ensures that users have access to the features and data necessary for their work while maintaining the confidentiality and integrity of the documents. **Role Hierarchy:** VOiC defines a hierarchy of user roles, each with a specific set of access permissions. At the highest level, administrators have full control over the system, including user management and system settings. Regular users, such as legal professionals, have access to document creation, editing, and management functions relevant to their role. **Role Assignment:** Upon user registration, a default role is assigned based on the user type. Administrators can later update user roles as necessary through the admin panel. **Granular Permissions:** Privileges within VOiC are granular, allowing for detailed control over who can view, create, edit, or delete documents. For instance, some users can be given read-only access to certain documents, while others might have permission to edit or share them. **Access Controls:** Access controls are implemented at both the application and database levels. The system checks the user's role and permissions before executing any action, whether it's a request to access a document or a database query.

## 7 Structure of VOiC

The Virtual Office in Cloud (VOiC) is architecturally structured into three primary layers: the front-end presentation layer, the back-end logic layer, and the database layer. Each layer is designed to operate both independently and interdependently, providing a scalable, maintainable, and secure web application.

### 7.1 Front-end Layer

The front-end layer is what users interact with. It includes all the client-side components that provide the user interface and user experience (UI/UX). **Components:**

- **HTML Documents:** The skeletal structure of the web pages.
- **CSS Stylesheets:** The design and aesthetic presentation of the web pages.
- **JavaScript Scripts:** The dynamic and interactive elements of the application.
- **Bootstrap Framework:** Ensures a responsive and consistent layout across various devices and screen sizes.
- **CKEditor:** Provides rich text editing capabilities.

[illegible]

id	name	email	password_hash	role
9	normal	noraml@gmail.com	\$2y\$10\$DgFw1Cu2Zvzn6rrcvMNOItzlaUYIMRQl1gYe1STc...	1
10	admin	admin@gmail.com	\$2y\$10\$YMXNe5T0HocFloXy.0ok0.0A94fgjrbMifHw3JcC3u...	

## 7.2 Back-end Layer

**Components:**

### 7.3 Database Layer

### Components:

## 7.4 Interconnectivity Between Layers

**Data Flow:** Data flows through the system starting from user inputs in the front-end, processed by business logic in the back-end, and stored or retrieved from the database layer.

**Functionality:** Handles data submission for new documents.

- Captures form data related to documents.
- Validates and sanitizes input.
- Inserts document information into the database.
- Redirects to a specified page post-submission.

## process-signup.php

**Functionality:** Manages user registration.

**Features:**

- Validates user input for account creation.
- Hashes passwords for secure storage.
- Checks for existing email to prevent duplicates.
- Inserts new user data into the database and redirects to login.

## read.php

**Functionality:** Displays content of a selected document.

**Features:**

- Retrieves document details by document ID.
- Formats and presents document information.
- Implements user session checks for access control.

## search.php

**Functionality:** Facilitates the search functionality for documents.

**Features:**

- Enables document search based on criteria.
- Displays search results with action options.
- Incorporates user session verification.

## validate-email.php

**Functionality:** Validates email uniqueness during registration.

**Features:**

- Checks if email exists in the database.
- Returns JSON response indicating email availability.

## update-doc.php

**Functionality:** Manages the update process for existing documents.

**Features:**

- Retrieves document data for editing.
- Allows users to modify document details.
- Updates document information in the database.
- Ensures session-based access control.

## update.php

**Functionality:** Processes updates to documents.

**Features:**

- Captures updated data from the form.
- Validates and sanitizes input.
- Executes database update queries.
- Redirects to home with success or failure notification.

## VOiCSearch.php

**Functionality:** Custom search interface for documents.

**Features:**

- Allows complex queries based on multiple document attributes.
- Displays search results with multiple options (read, edit, delete).
- Implements user session checks for access control.

## VOiC.php

**Functionality:** Main interface for advanced document search.

**Features:**

- Provides a detailed search form for documents.
- Integrates with VOiCSearch.php for result processing.
- Includes session-based user authentication.

## Log-In.php

**Functionality:** Handles user authentication.

**Features:**

- Authenticates user credentials.
- Manages user sessions and redirects based on roles.
- Provides user feedback on login failure.

## admin.php

**Purpose:** The `admin.php` file serves as the administrative dashboard for managing the web application. **Features:**

- User Management: Facilitates the viewing, adding, editing, and deletion of user accounts and roles.
- Content Oversight: Allows administrators to manage user-submitted content.
- System Settings: Control global settings of the application.
- Reports and Analytics: Access to user activity and system usage reports.
- Security: Management of security settings and monitoring.

## create.php

**Purpose:** This file is responsible for creating new entries in the system, like documents or records. **Features:**

- Data Entry Form: Provides a form for inputting new entry details.
- Validation: Ensures data integrity and compliance with required formats.
- Database Interaction: Handles insertion of new data into the database.
- Feedback Mechanism: Offers user feedback post submission.
- File Uploads: Manages uploading of related files and media.

## fetch-mentions.php

**Purpose:** Used for retrieving specific mentions or references from the database. **Features:**

- Data Retrieval: Executes queries to search for specific mentions.
- Dynamic Search: Supports user-inputted searches.
- Response Formatting: Structures the search results in a user-friendly format.
- Error Handling: Manages potential errors or data retrieval issues.

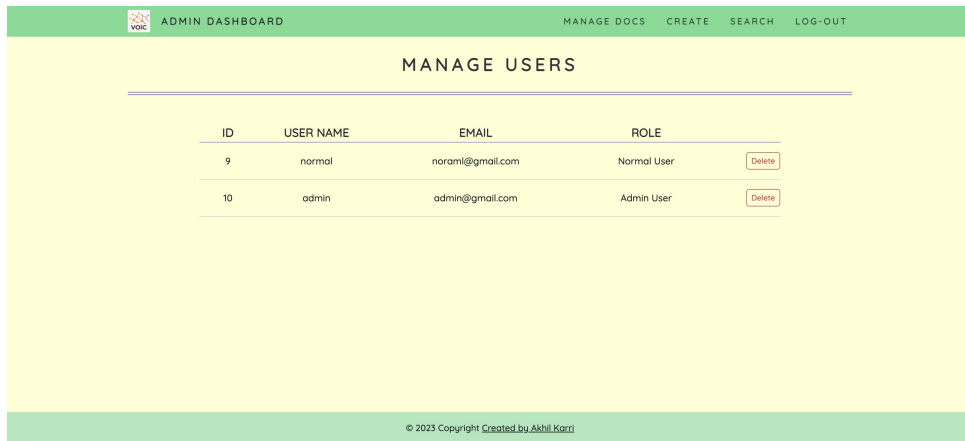


Figure 10: Admin Dashboard for Managing Users

## validate-email.php

**Purpose:** The script is utilized to validate email addresses during user registration or data entry processes.

**Features:**

- Email Verification: Checks if the email address format is valid.
- Database Check: Verifies whether the email already exists in the database.
- JSON Response: Returns the validation result in JSON format for easy handling by client-side scripts.
- Security: Ensures secure handling of email data to prevent SQL injection and other security vulnerabilities.

## 7.5 style.css

**Functionality:** Provides styling for the web application.

**Features:**

- Defines CSS rules for layout and visual elements.
- Ensures consistent look across pages.
- Enhances user experience with responsive design.

## 7.6 VOiC.sql

**Functionality:** Contains the database schema and initial data for the VOiC application.

**Features:**

- Defines the structure for 'documents' and 'user' tables.
- Includes initial data for testing and setup purposes.
- Provides a basis for database operations within the application.

# 8 Admin Dashboard - Manage Users

This interface is part of the VOiC's administrative panel. It lists users with options to manage their accounts, including deletion. The table provides information such as user ID, name, email, and role within the system.

## 9 Admin Dashboard - Documents

This page serves as a repository for documents stored within VOiC, allowing administrators to read, edit, or delete documents. It displays key information such as the title, author, and date of creation.

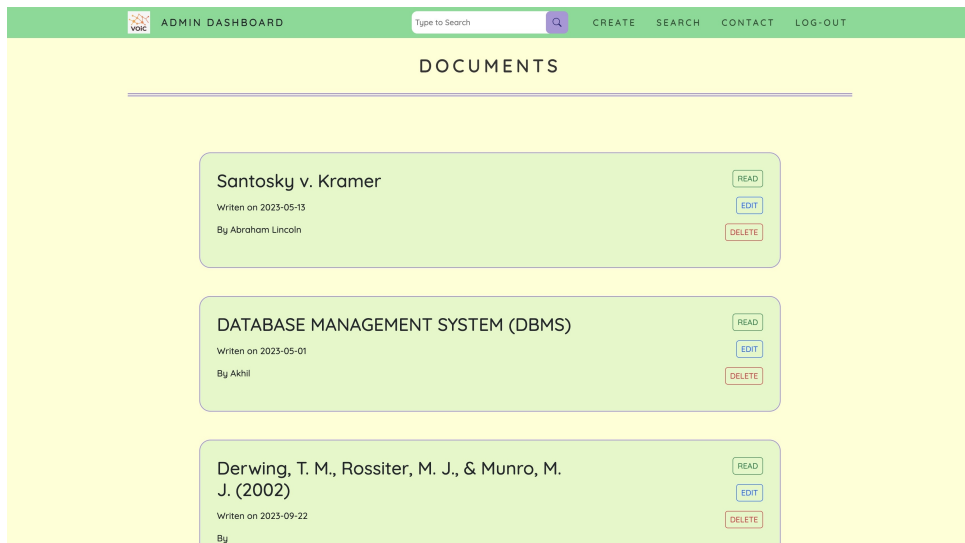


Figure 11: Documents Management in Admin Dashboard

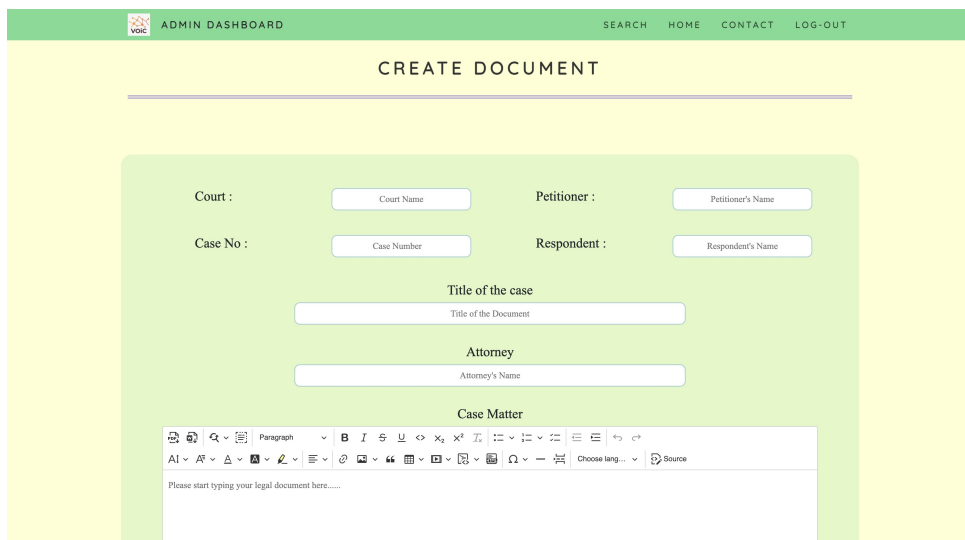


Figure 12: Document Creation Interface



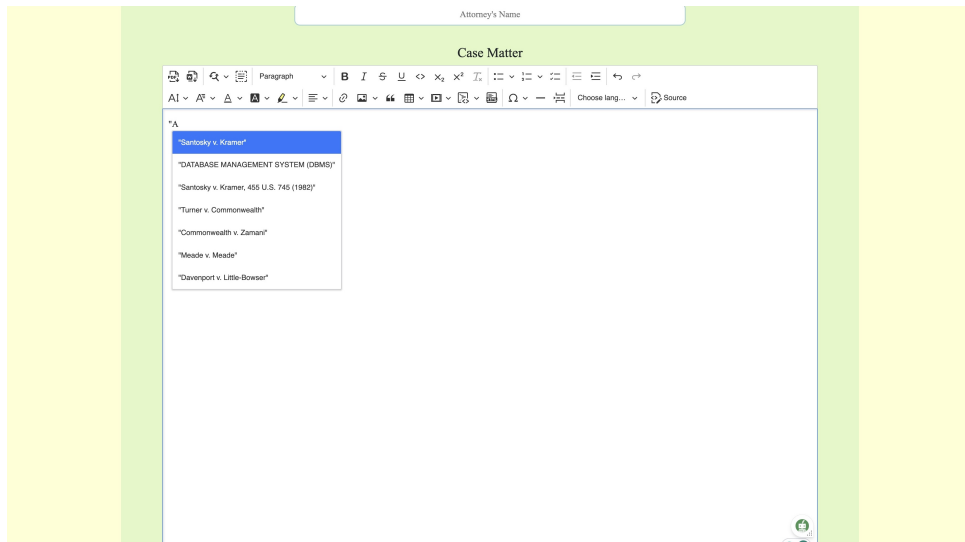


Figure 13: Extended Document Creation Interface with Dropdown Auto-Complete Citations Feature

Figure 14: Advanced Search Functionality

## 10 Admin Dashboard - Create Document

Here we see a form designed for the creation of new legal documents. It features comprehensive fields for case details and a rich text editor for content entry, offering a user-friendly document drafting experience.

## 11 Admin Dashboard - Create Document with Auto Complete Citations

An extension of the document creation interface, showcasing a dropdown menu that likely facilitates the insertion of standard legal clauses or templates into the document.

## 12 Admin Dashboard - VOiC Search

The VOiC search interface provides advanced filtering options for users to locate documents based on various criteria such as date, involved parties, and location.


## 13 Sign-Up Page

The sign-up page is the entry point for new users, offering a straightforward registration form that collects essential user information and promotes the system's capabilities.

VIRTUAL OFFICE IN CLOUD

ABOUTCONTACTLOG-IN

SIGN-UP



VIRTUAL OFFICE IN THE CLOUD  
BEST ONLINE WEBSITE TO STORE LEGAL DOCUMENTS  
VOIC USES SUB-GRAPH ISOMORPHISM WHICH HELPS YOU FIND ANY DOCUMENT EFFORTLESSLY

Full Name :

Email :

Password :

Confirm Password :

User Type :


Sign Up

Figure 15: User Registration Form

VIRTUAL OFFICE IN CLOUD

ABOUTCONTACTSIGN-UP

LOG-IN



Email :

Password :

Log In

Figure 16: User Authentication Interface

## 14 Log-In Page

This login page is designed for user authentication, featuring a minimalistic form that requests the user's email and password.

## 15 Discussion

The development of VOiC (Virtual Office in Cloud) represents a significant step forward in the realm of legal document management systems. Throughout the project, several key insights were gleaned that not only shaped the development of VOiC but also have broader implications for cloud-based document management solutions.

**Integration of Advanced Technologies:** One of the primary achievements of VOiC is the successful integration of various technologies, including PHP, MySQL, Apache, and CKEditor. The seamless interplay between these technologies resulted in a robust, user-friendly platform that can cater to the dynamic needs of legal professionals.

**Security and User Experience:** Balancing rigorous security protocols with a user-friendly interface was a considerable challenge. Implementing features like password hashing, SQL injection prevention, and role-based access control ensured the security of user data without compromising on the ease of use.

## 16 Conclusion

VOiC has been a project of both technical achievement and practical application. Its development journey highlights the importance of a user-centered approach in software design, especially in fields that handle sensitive information like legal documentation.

### 16.1 Achievements:

The project successfully delivered a cloud-based platform that simplifies the complexities inherent in legal document management. Key features such as rich text editing, auto-complete functionalities, and role-based user privileges demonstrate VOiC's commitment to enhancing productivity and security.

### 16.2 Challenges and Resolutions:

Throughout the project, various challenges, including optimizing the user interface, Auto-Complete Citations and ensuring robust security measures, were effectively addressed. These challenges were valuable learning experiences, contributing significantly to the professional growth of the development team and the overall success of VOiC.

### 16.3 Future Directions:

As VOiC evolves, there are several promising avenues for its advancement, reflecting the ongoing need for innovation in legal document management systems. **Subgraph Isomorphism for Advanced Document Search:** A significant enhancement to VOiC's search functionality would be the incorporation of subgraph isomorphism algorithms. This approach can revolutionize how legal professionals search through documents by allowing for more complex, pattern-based queries. Subgraph isomorphism can enable users to find documents that contain specific patterns or structures in their text, making the search process not only more efficient but also more comprehensive. This feature would be particularly useful for identifying documents with similar legal arguments, references, or clauses, enhancing the depth and relevance of search results.

**Machine Learning for Document Categorization:** Integrating machine learning algorithms for smarter document categorization and search could significantly improve the system's efficiency. By analyzing document content and user interaction patterns, the system could automatically categorize and tag documents, facilitating quicker and more accurate retrieval. **Natural Language Processing**

**Enhancements:** Expanding the auto-complete functionality with advanced natural language processing techniques could further streamline the document creation process. This would allow for more intuitive and context-aware suggestions, reducing the time legal professionals spend on drafting and editing documents.

**Blockchain-Based Document Verification:** Exploring blockchain technology for document verification could offer a new layer of security and authenticity. Blockchain integration would ensure the integrity of documents, making VOiC a more robust platform for handling sensitive legal information. These future enhancements aim to solidify VOiC's position as a cutting-edge solution in the legal tech space, continually adapting to the evolving needs of its users and leveraging the latest technological advancements.

## References

1. Hayden Carroll and Austin Kugler. Voic: Virtual office in the cloud. 2001.
2. Dr. Hasan Jamil. A semantic query engine for knowledge-rich legal digital libraries. 20xx.
3. Git Hub, 2022. URL <https://github.com/>.