

Contents

[Introduction 2](#_Toc178362200)

[Operating Environment 2](#_Toc178362201)

[User Roles 2](#_Toc178362202)

[Key Features 2](#_Toc178362203)

[Technical Overview 2](#_Toc178362204)

[Backend 2](#_Toc178362205)

[Frontend 3](#_Toc178362206)

[Hosting 3](#_Toc178362207)

[Data Requirements 3](#_Toc178362208)

[Constraints 3](#_Toc178362209)

[Setup Instructions for Developers 3](#_Toc178362210)

[Prerequisites 3](#_Toc178362211)

[Cloning the Repository 3](#_Toc178362212)

[Backend Setup 4](#_Toc178362213)

[API Integration 4](#_Toc178362214)

[QuickChart Watermark API 4](#_Toc178362215)

[Usage 4](#_Toc178362216)

[Example Integration 4](#_Toc178362217)

[Conclusion 5](#_Toc178362218)

# Introduction

Siyafunda is an online centralized platform designed for educators and learners to facilitate resource sharing and enhance accessibility. The platform enables institutions and educators to register and create an online space where students can access educational resources, tests, assignments, and communication tools.

## Operating Environment

The system is web-based and accessible through any modern web browser on mobile and desktop platforms. It is hosted on Azure at the URL <https://siyafunda.azurewebsites.net/>

## User Roles

* System Admin: Has unrestricted access to all system parts and can manage users and business-related functions.
* System Developer: Maintains system functionality and available modules.
* Moderator: Assigned to manage files that are uploaded and moderate activity on the site.
* Educator: Manages the resources, files, tests, and grading within their allocated section.
* Student: Accesses resources, assignments, and communication tools provided by their educators.

## Key Features

* Account Creation & Secure Sign-in
* File Uploading & Storage
* File Moderation and Reporting
* Document Tagging and Searching
* Analytics to Monitor User Engagement
* Document Rating System
* FAQ Page Management

# Technical Overview

### Backend

* Framework: ASP.NET
* Data Storage: SQL database for user data, file metadata, and analytics.
* Security: JWT for authentication, HTTP-only cookies, and encrypted passwords.
* API: Handles request behaviour and manages complex interactions between the front and backend.

### Frontend

* Framework: React
* Responsive Design: Accessible via desktop and mobile browsers.

### Hosting

The platform is hosted on Microsoft Azure, ensuring scalability and reliability.

### Data Requirements

* File Storage: Documents are stored in a file storage system.
* File Metadata: Subject, grade, keywords, and ratings are stored in an SQL database.
* User Data: Securely stores user roles, login details, and activity logs.

### Constraints

* Compatibility: Primarily a web application.
* Modularity: The system is designed to be flexible and extendable for future expansion.
* Open-Source Libraries: All frameworks and libraries used will be open-source, minimizing costs.
* Security: Standard data security practices will be followed.
* Project Timeline: The project will be completed by October 16, 2024.
* Version Control: Managed with GitHub.

# Setup Instructions for Developers

## Prerequisites

* Visual Studio 2022 or later
* SQL Server (LocalDB or a full SQL Server instance)
* .NET Framework 4.8.1
* Node.js (for the frontend)
* Git (for version control)

## Cloning the Repository

Clone the repository using:

git clone https://github.com/yourusername/Siyafunda.git

## Backend Setup

* Open the SiyafundaApplication.sln (solution file) in Visual Studio 2022.
* Open SQL Server Management Studio and run the necessary SQL scripts to create the database schema (these should be provided in the Scripts folder).
* Build the solution to restore NuGet packages and ensure everything compiles without errors.
* Run the application using the built-in Visual Studio web server (IIS Express).
* Important to note: The system uses .Net framework version 4.8.1, so make sure that is installed locally beforehand.

# API Integration

## External APIs

### QuickChart Watermark API

Siyafunda integrates with the QuickChart Watermark API to apply watermarks to images uploaded by educators. The following parameters are used in the POST request:

* mainImageUrl: URL of the main image to which the watermark will be applied.
* markImageUrl: URL of the watermark image (e.g., institution logo).
* markRatio: The ratio of the watermark size relative to the main image.

### Usage

To apply a watermark during the file upload process, the application makes an HTTP GET request to the QuickChart Watermark API using the image URLs and mark ratio. The resulting watermarked image can then be saved back to the server, replacing the original image.

### Example Integration

In the ASP.NET backend, the watermarking process is handled in the UploadFileAsync method, which manages file uploads and interacts with the QuickChart Watermark API.

For instance, the following code demonstrates how the API is called:

string mainImageUrl = $"/{relativeFilePath}";  
string watermarkUrl = $"/Assets/SiyafundaLogo.png"; //logo path  
watermarkedImageUrl = await AddWatermark(mainImageUrl, watermarkUrl, 0.25);

## Internal APIs

The internal API forms of Siyafunda form the backbone of the core functionality of the system in terms of scope. The main implemented APIs are internal REST APIs that allow users to access server data as well as store their own approved data.

Where REST APIs were essential:

* **Account Creation and Secure Sign-in**: Allows users to create and log into existing accounts that are stored securely server side.
* **File Uploading**: Educators upload data to the server relevant to their module and in the process, the system also collects metadata of the file and stores it in the SQL database. The API also ensures that only properly authenticated users with appropriate levels of access can utilize this.
* **File Moderation**: Pending files are fetched and presented to moderators to either approve or deny depending on the content. This process determines the final public availability of moderated files.
* **Document Searching**: Searches documents relevant to the client based on keywords.
* **FAQ instances**: Frequently asked questions of a module can be answered by educators and are then easily available to students.
* **Announcements**: Similarly to FAQ instances, educators can make important announcements, and the server will make those announcements available to relevant students.
* **Timetable**: Allows students to create their timetables according to their individual schedules and have it always available to them through an API request.
* **Online Quizzes and grade book**: Educators can make custom quizzes and have them automatically marked once students complete them. The results are then almost immediately available to students in a grade book format.

# Conclusion

Our developers are actively working on the project, and we aim to construct the front end of the website fully as part of our next major deliverable. We are committed to ensuring that the platform is user-friendly and meets the needs of both educators and learners. Continuous feedback from users will guide our development process, allowing us to enhance functionality and improve the overall user experience. We appreciate your support and look forward to launching a comprehensive solution that promotes effective resource sharing in the educational community.