

**Functional Requirements Report**

**Project Title: Shule Finder – A Web-Based School Discovery Platform**

**Prepared by: Abdulrazzaq**

**Course:** MID-TERM PROJECT

Contents

[1](#_Toc205446576)

[**Functional Requirements Report** 1](#_Toc205446577)

[**Project Title: Shule Finder – A Web-Based School Discovery Platform** 1](#_Toc205446578)

[**Prepared by: Abdulrazzaq** 1](#_Toc205446579)

[1. Introduction 2](#_Toc205446580)

[2. Project Overview 2](#_Toc205446581)

[3. Objectives of the System 2](#_Toc205446582)

[4. Scope of the Functional Requirements 2](#_Toc205446583)

[**5**. Functional Requirements 2](#_Toc205446584)

[**5.1 User Registration and Authentication** 2](#_Toc205446585)

[**5.2 School Search and Filtering** 3](#_Toc205446586)

[**5.3 School Profiles and Listings** 3](#_Toc205446587)

[**5.4 School Submission by Administrators** 3](#_Toc205446588)

[**5.5 Parent Reviews and Ratings** 3](#_Toc205446589)

[**5.6 Comparison Tool** 3](#_Toc205446590)

[**5.7 Favorites and Saved Searches** 4](#_Toc205446591)

[**5.8** Admin Panel 4](#_Toc205446592)

[**5.9** Contact and Support 4](#_Toc205446593)

[**5.10 Responsive Web Interface** 4](#_Toc205446594)

[6. System Integration Requirements 4](#_Toc205446595)

[**8. Conclusion** 5](#_Toc205446596)

# 

# **1. Introduction**

The purpose of this document is to define the **functional requirements** of the Shule Finder platform. This report outlines the primary features, operations, and behaviors of the system that are critical to satisfying user needs. Functional requirements define what the system should do, and they form the foundation for system development and validation.

# **2. System Overview**

Shule Finder is a web-based application designed to assist parents and guardians in Kenya to easily find, evaluate, and compare educational institutions for their children. The system incorporates features for school listing, review management, search filtering, school profile creation, and data visualization.

# **3. Functional Requirements**

Each functional requirement is presented as a numbered item and includes a description, priority, and related actors or components.

## **3.1 User Registration and Authentication**

* **Description:** Users must be able to register accounts, log in securely, and manage profiles.
* **Features:**
  + Account registration (email and password)
  + Login/logout mechanism
  + JWT-based session handling
* **Priority:** High
* **Actors:** Parents, Admins

## **3.2 School Search and Filtering**

* **Description:** Users can search for schools based on various filters such as location, curriculum, tuition range, and rating.
* **Features:**
  + Keyword search
  + Filter by county/sub-county
  + Filter by curriculum (e.g., CBC, 8-4-4, IGCSE)
  + Sort by tuition cost, reviews, or performance
* **Priority:** High
* **Actors:** Parents, Guardians

## **3.3 School Profile Management (Admin/Owner Role)**

* **Description:** School administrators can register and update their institution’s profile.
* **Features:**
  + Add/edit school name, address, contact info
  + Upload school logo and images
  + Define curriculum, performance metrics, tuition fees
* **Priority:** Medium
* **Actors:** School Admins

## **3.4 Review and Rating System**

* **Description:** Parents and guardians can leave reviews and ratings based on their experiences.
* **Features:**
  + Star-based rating
  + Text review submission
  + View aggregated review scores per school
* **Priority:** High
* **Actors:** Parents, Guardians

## **3.5 Data Visualization (Optional MVP Feature)**

* **Description:** Display graphical comparisons (e.g., bar charts for school ratings or tuition).
* **Features:**
  + Charts based on location or performance
  + Interactive comparison tools
* **Priority:** Low (Future enhancement)
* **Actors:** All users

### ****3****.6 Backend and API Services

* **Description:** Backend APIs to handle all business logic and data interaction.
* **Features:**
  + RESTful APIs for CRUD operations (users, schools, reviews)
  + MongoDB integration for data persistence
* **Priority:** High
* **Actors:** System, Frontend

### ****3****.7 Admin Dashboard

* **Description:** An internal interface for managing system data, approving schools, and moderating reviews.
* **Features:**
  + Admin login
  + Approve/reject school applications
  + Monitor platform activity
* **Priority:** Medium
* **Actors:** Platform Admin

## **3.8 Contact and Support Mechanism**

* **Description:** Users should be able to report issues or contact system support.
* **Features:**
  + Contact form or chatbot
  + FAQ or help page
* **Priority:** Low
* **Actors:** All users

## **3.9 Mobile Responsiveness**

* **Description:** The platform must work seamlessly across mobile and desktop browsers.
* **Features:**
  + Responsive UI components
  + Adaptive layout rendering
* **Priority:** High
* **Actors:** All users

# **4. Dependencies**

* **Frontend:** ReactJS, Axios, Tailwind CSS
* **Backend:** Node.js, Express.js, MongoDB
* **Authentication:** JWT
* **Deployment:** GitHub, Render, or Vercel (optional)

# **5. Assumptions**

* Users will have basic internet literacy.
* Schools will provide accurate data.
* A valid email address is used for user verification.

# **6. Conclusion**

The functional requirements detailed above represent the essential behaviors and services that the Shule Finder platform must offer to meet user expectations. These features will be used as a basis for system implementation, quality assurance, and future updates.