The Buffalo Academy System

Below is my Proposed Project Budget

|  |  |  |
| --- | --- | --- |
| Module | Sub-modules | Cost Estimate (KSH) |
| Courses |  | 5000 |
|  |  |  |
| Dashboard | User as Supper Admin | 15000 |
| User as Student |
| User as Admin |
| User as Tutor |
| User as Client |
| Blog |  | 5000 |
|  |  |  |
| Projects |  | 5000 |
|  |  |  |
| System Process Workflow | System configurations | 20,000 |
| System Audit Trail |
| System workflow |
| System Notifications |
|  |  |  |
| System Testing | System development | 30,000 |
| System deployment |
|  |  |  |
| Total Project Budget |  | 80,000 |

NOTE: I will hand over the following services for free (not paid for)

# 1. System User Guide/Manual

- Role: This document serves as a comprehensive guide for users on how to navigate and utilize the Buffalo Academy system effectively. It typically includes step-by-step instructions, explanations of features, troubleshooting tips, and best practices for using the system.

- Importance: Providing a user guide/manual enhances user experience by reducing confusion and frustration. It helps users maximize the benefits of the system and encourages adoption and retention.

# 2. System Architecture Document

- Role: The architecture document outlines the high-level structure and components of the Buffalo Academy system. It describes how different modules and components interact with each other, as well as the overall design principles and decisions.

- Importance: Understanding the system's architecture is crucial for developers, architects, and other stakeholders involved in system design, development, and maintenance. It provides a blueprint for building, modifying, and scaling the system while ensuring consistency and maintainability.

# 3. System Requirement Specifications (SRS)

- Role: The SRS document defines the functional and non-functional requirements of the Buffalo Academy system. It specifies what the system should do (functional requirements) and how it should perform (non-functional requirements), including constraints, user expectations, and system behavior.

- Importance: The SRS serves as a contract between stakeholders, including developers, clients, and users. It ensures alignment between expectations and deliverables, guides the development process, and serves as a basis for testing and validation.