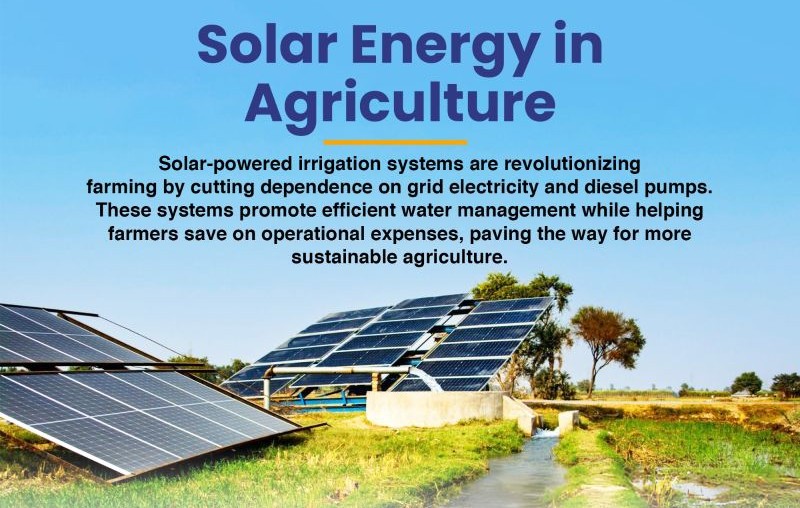
**Introduction**

As the global population increases the agricultural sector is put under pressure to meet the increasing demands for resources. This growth has contributed to the expansion of unsustainable agricultural practices, however, the shift towards green energy has proven to promote resource efficiency.



Afri-Energy Connect will serve as an enterprise software, described by the IIE (2025) as a software used to support the operations of organizations. Its main objective will be to provide a centralised hub where the agricultural sector and green energy providers can acquire the same data, share and allocate resources and engage in collaborative problem-solving.

**Non-Functional Requirements**

**Scalability**

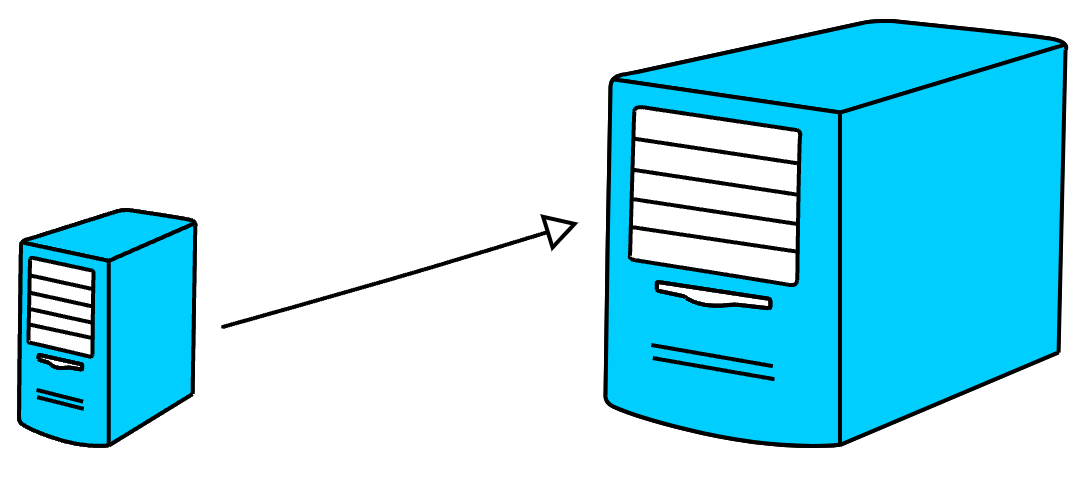
Enterprise software is designed to handle a large user base and large volumes of diverse data. Singh (2025) explains that scalability ensures that this growth and different workloads can be managed.

**Requirements:**

* The system must handle 1,000 concurrent users and scale up to support 10,000 users without performance issues.
* The system must automatically increase storage to handle increasing data persistence.

**Implementation:**

Tools and infrastructures that automatically scale resources based on demand will be implemented.



**Security**

Robust security strategies should be implemented to protect sensitive data which enterprise software often stores.

**Requirements:**

* The system must set role-based access controls so that it is accessed only by authorized users and services.
* The system must perform regular security updates to identify potential vulnerabilities.

**Implementation:**

A risk assessment will be conducted and after its review appropriate measures will be implemented. These measures will include data classification, access controls and security software use.

A diagram of a role-based access control

AI-generated content may be incorrect.

**Usability**

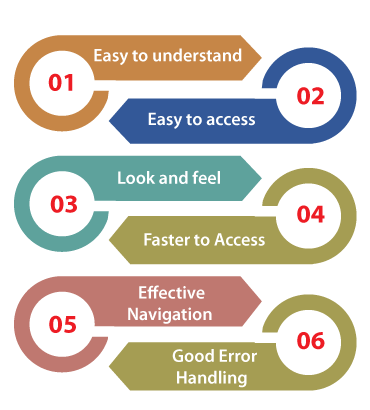
Usability determines how easy a user can use a product to accomplish their goals (The IIE 2025).

**Requirements:**

* The system must have screen layouts and functions that are easy to understand even for users with limited technical skills.
* The system must provide support information to assist different users.

**Implementation:**

A user-centric design approach that prioritizes understanding the needs and abilities of users will be applied. Concise feedback will also be provided to users about their requests and actions.



**Performance**

Performance refers to how well software responds to requests under specific conditions (The IIE 2025).

**Requirements:**

* The system must respond to users’ interactions and requests within 5 seconds.
* The system must automatically collect and process real-time data from sensors and other sources without delays.

**Implementation:**

AI-powered analytics will be integrated to gain insight into the software’s response time to requests and processing of real-time data.

A screenshot of a graph

AI-generated content may be incorrect.

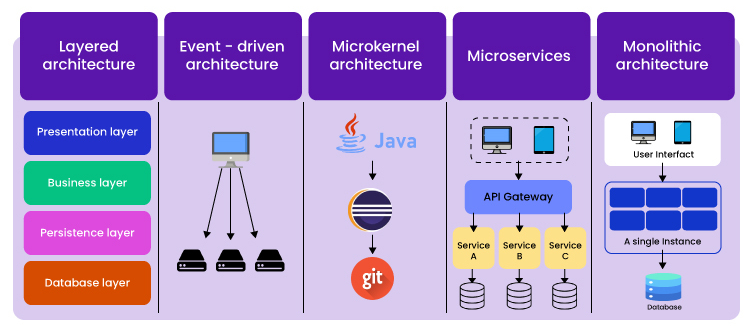
**Design and architecture patterns**

**Importance**

Commonly known problems are prone to arise during software development. Khan (2023) demonstrates how design and architecture patterns offer an approach to solving these problems. Although related, these patterns are different, with design patterns providing proven solutions and architecture patterns providing structured frameworks (The IIE 2025).

These patterns are important because they allow developers to predict and address problems before they exist. This saves development time and effort. They also allow for maintainable and reliable software to be built because most of the solutions offered reduce software complexity making it easier to modify and debug (Khan 2024).

**Integration**

 Steps for pattern integration:

1. Understanding user needs and software requirements to select effective patterns.
2. Selecting appropriate design patterns:

* Structural patterns to define how different users will interact with the software.
* Concurrency patterns to manage multiple tasks concurrently, such as receiving different requests from different users (The IIE 2025).
* Behavioural patterns to find appropriate handlers to be responsible for different requests.

1. Selecting appropriate architecture pattern:

* Event-driven architecture to analyse and process real-time data.
* Microservices architecture to enable the incorporation of third-party services.

**Benefits for Afri-Energy Connect** (Khan 2023)

* Adaptability – The software will adapt to changing needs and requirements.
* Risk Reduction – These patterns provide pre-tested solutions that address common problems. Integrating them will reduce the likelihood of errors in the software.
* Scalability – The software will be able to manage changes such as increased workloads without requiring modification.
* Modularity – These patterns can divide and organize the software into smaller parts simplifying maintenance.
* Collaboration – They provide a common framework that will enable multiple developers to work on the software, resulting in improved productivity.

**References**

[Khan](https://dev.to/sardarmudassaralikhan), S. 2023 Design Patterns and Their Benefits. *[Online]* Available: <https://dev.to/sardarmudassaralikhan/design-patterns-and-their-benefits-2ngk>. [Accessed 05 April 2025].

Singh, A. 2024. System Design: What is Scalability? *[Online]* Available: <https://blog.algomaster.io/p/scalability>. [Accessed 05 April 2025].

The IIE. 2025. Programming 3A [PROG7311 Module Manual]. The Independent Institute of Education: Unpublished.