Research and Innovation Action
HORIZON-CL6-2021-FARM2FORK-01
TOPIC: HORIZON-CL6-2021-FARM2FORK-01-18

NESTLER
oNe hEalth SusTainabiLity partnership between
EU-AFRICA for food sEcuRity

Deliverable D5.2
Evaluation of NESTLER 1st phase platform

Authors	Dr Rabbi, Kolawole Olalekan
Nature	Report
Dissemination	PUBLIC
Version	2.0
Status	Final
Delivery Date (DoA)	M32
Actual Delivery Date	30/05/2025

Keywords	NESTLER, One Health, sustainability, food security, AI models, predictive analytics, crop farming, livestock management, aquaculture, IoT sensors, data accuracy, real-time monitoring, disease management, resource optimization, insect protein, sustainable agriculture, environmental monitoring, stakeholder engagement, interdisciplinary collaboration, digital tools, remote sensing technologies, big data analytics, food safety, traceability systems, capacity building, training programs, user feedback, performance indicators (KPIs), system uptime, usability.
Abstract	Deliverable D5.2: Evaluation of NESTLER 1st phase platform, focus on the initial evaluation of the NESTLER platform. The first phase evaluates the platform's operational performance, data collection efficiency, and its integration into agricultural systems for enhancing productivity and resilience. Key performance indicators and user feedback are analyzed to assess the platform's functionality, identify areas for improvement, and ensure alignment with project objectives for future development phases.

© Copyright by the NESTLER Consortium
Funded by the European Union under Grant Agreement no.101060762

DISCLAIMER

This document is a deliverable of the NESTLER project funded by the European Union under Grant Agreement no.101060762. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Executive Agency, while neither the European Union nor the granting authority can be held responsible for any use of this content.

This document may contain material, which is the copyright of certain NESTLER consortium parties, and may not be reproduced or copied without permission. All NESTLER consortium parties have agreed to the full publication of this document. The commercial use of any information contained in this document may require a license from the proprietor of that information.

Neither the NESTLER consortium as a whole, nor a certain party of the NESTLER consortium warrant that the information contained in this document is capable of use, nor that use of the information is free from risk and does not accept any liability for loss or damage suffered using this information.

	Participant organisation name	Short	Country
01	SYNELIXIS SOLUTIONS S.A.	SYN	EL
02	CloudEO AG (Terminated)	CEO	DE
03	RINIGARD DOO ZA USLUGE	RINI	HR
04	EBOS TECHNOLOGIES LIMITED	eBOS	CY
05	STICHTING IDH SUSTAINABLE TRADE INITIATIVE	IDH	NL
06	ZANASI ALESSANDRO SRL	Z&P	IT
07	AGRIX TECH SARL	AGRI	CM
08	CONSERVATION THROUGH PUBLIC HEALTH	CTPH	UG
09	THE INTERNATIONAL CENTRE OF INSECT PHYSIOLOGY AND ECOLOGY LBG	ICIPE	KE
10	ETHIOPIAN INSTITUTE OF AGRICULTURAL RESEARCH	EIAR	ET
11	RWANDA AGRICULTURE AND ANIMAL RESOURCES DEVELOPMENT BOARD	RAB	RW
12	INTERNATIONAL INSTITUTE OF TROPICAL AGRICULTURE	IITA	NG
13	MANA BIOSYSTEMS LIMITED	MANA	UK
14	UNIVERSITY COLLEGE LONDON	UCL	UK
15	RINISOFT LTD	RINIS	BG
16	ADAPT IT	ADA	DE

ACKNOWLEDGEMENT

This document is a deliverable of NESTLER project. This project has received funding from the European Union's Horizon Research and innovation programme under grant agreement № 101060762.
Views and opinions expressed are, however, those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Executive Agency, while neither the European Union nor the granting authority can be held responsible for any use that may be made of the information it contains.

Document History

Version	Date	Contributor(s)	Description
V0.1	6 Nov, 2024	Dr I. Rabbi (IITA)	First draft, work assignments
V0.2	12 Nov, 2024	K. Olalekan (IITA), K Pramataris (SYN), G. Athanasiou (SYN), P. Karkazis (SYN)	Updated Version
V0.3	10 Dec, 2024	K. Olalekan (IITA), I. Oikonomidis (SYN), S. Charamousou (SYN), P. Karkazis (SYN)	Chapter 1-2
V0.4	20 Feb. 2024	I. Oikonomidis (SYN), K.Pramataris (SYN) G. Athanasiou (SYN), A. Skias(SYN)	Chapter 3
V0.5	10 Apri. 2024	I. Oikonomidis (SYN), A. Skias(SYN) A. Tomaras (SYN), N. Arvanitis (SYN)	Revision Chapter 3 Chpater 4
V0.6	26 April, 2025	A. Tomaras (SYN), N. Arvanitis (SYN), G. Athanasiou (SYN), A. Skias (SYN)	Technical Validation of the platform
V0.7	10 May, 2025	A. Tomaras (SYN), N. Arvanitis (SYN)	Revision of Chapter 1-4
V0.8	15 May, 2025	K. Olalekan (IITA), Th. Zahariadis (SYN)	Document Review
V0.9	21 May 2025	A. Tomaras (SYN), P. Athanasoulis (SYN)	Revision of Chapters 3-5
V1.0	30 May, 2025	Th. Zahariadis (SYN), N. Arvanitis (SYN)	Final Version

Document Reviewers

Date	Reviewer's name	Affiliation
23/05/2025	Adamou Nchange Koutou	AGRI
30/05/2025	G. Kliafas	ADA
30/05/2025	P. Athanasoulis, A. Tomaras	SYN

Table of Contents

List of Figures

List of Tables

Definitions, Acronyms and Abbreviations

AI	Artificial Intelligence
ARIMA	Auto Regressive Integrated Moving Average
BSF	Black Soldier Fly
BSFL	Black Soldier Fly Larvae
BSFLM	Black Soldier Fly Larvae Meal
CNN	Convolutional Neural Network
DGR	Daily Growth Rates
DMO	Data Management Officer
DMP	Data Management Plan
DSS	Decision Support System
DZARC	Debre Zeit Agricultural Research Centre
EC	European Commission
EPPPA	Ethiopian Poultry Producers and Processors Association
ESA	European Space Agency
EU	European Union
FCF	Fulton's Condition Factor
GAN	Generative Adversarial Network
GBM	Gradient Boosting Machines
GIS	Geographic Information System
GPS	Global Positioning System
IACUC	Institutional Animal Care and Use Committee
IoT	Internet of Things
KALRO	Kenya Agricultural and Livestock Research Organization
KPI	Key Perform Indicator
LDR	Light Dependent Resistor
LMIC	Low- to Middle Income Countries
ML	Machine Learning
NIR	Near Infrared Reflectance
NPK	Nitrogen, phosphorus, and potassium fertiliser
pH	Potential of Hydrogen
SRTM DEM	Shuttle Radar Topography Mission Digital Elevation Model
TL	Total Length
TW	Total Weight
UAV	Unmanned Aerial Vehicles
VSRI	Veterinary Science Research Institute
VU	Virtual User

Executive Summary

NESTLER is a joint project between the EU and African member states designed to promote One-Health sustainable partnership. The project focuses on the critical need to bring together experts in interdisciplinary research to devise solutions that enable technological interventions ensuring the safety and security of the food supply-chain between the EU and Africa. NESTLER project aims to integrate interdisciplinary technological advancements for effective monitoring of the well-being of animals, plants, and humans, adopting a holistic approach. To validate its objectives and platform outcomes, NESTLER conducts pilot demonstration sites across six African member states (Uganda, Ethiopia, Cameroon, Kenya, Rwanda, and Nigeria).

Deliverable D5.2 includes activities from WP5 and especially the tasks T5.2, T5.3, T5.4 and T5.5, focusing on the first phase of the NESTLER platform's evaluation. It includes the operational and user experience assessment coming from the end users including researchers, policy makers and farmers and the technical validation of the platform. This report evaluates the technical efficiency, integration, and effectiveness of the platform in addressing challenges in crop and livestock farming.

The keystones of the deliverable are summarised below:

- Analysis of the assessment methodology
- End-users' assessment across pilot and across different region
- Technical validation of the NESTLER platform

The assessment outcomes will be taken into consideration, and any update will be reported in the deliverable "D5.5 Evaluation of NESTLER 2nd phase platform" (M42).

1. Introduction

This deliverable evaluates the NESTLER platform's performance across all aspects of its pilot implementations. The primary objective is to assess how effectively the platform integrates Internet of Things (IoT) technologies, cloud storage systems, and AI-driven analytics to solve targeted problems in plant, animal, and human health interaction monitoring from the pilots to enhance food security. The NESTLER platform is designed as an integrated platform that combines various technological components to support agricultural practices. The platform employs a modular architecture that allows for scalability and flexibility in deployment.

The following components were adopted in rendering services to users (farmers, researchers, policymakers):

- IoT Framework: Utilizes sensors for real-time monitoring of soil health, crop growth, and livestock conditions.
- Cloud Storage Systems: Provides centralized data storage for easy access and analysis.
- AI-Driven Analysis: Leverages machine learning algorithms to generate predictive insights regarding crop yields, disease outbreaks, weather-related risks and livestock performance.
- Field-Specific Applications: Tailored applications for crop and wildlife monitoring, livestock and aquaculture management.

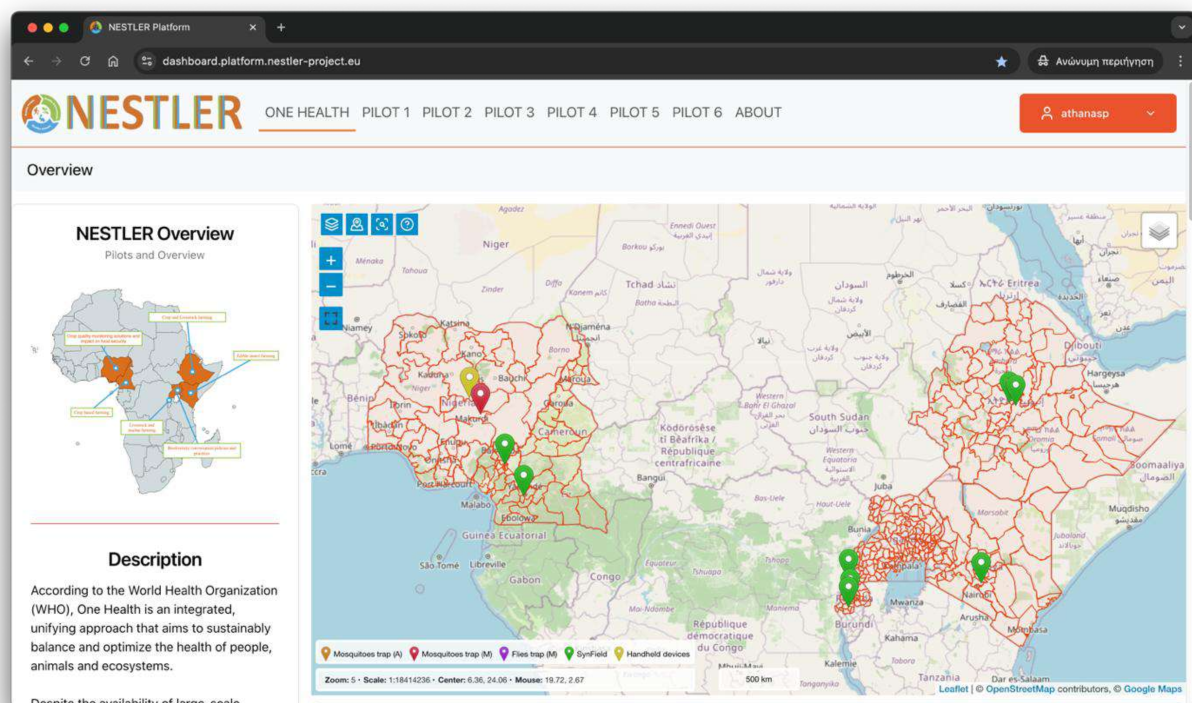

Figure 1: NESTLER Platform Homepage

1.1. Model Development, Integration, and Optimization in the NESTLER Platform

The development of AI models for the NESTLER platform involves several key steps, including data collection, model design, integration with platform architecture, and continuous optimization. This section provides a detailed explanation of how the models used within the platform are developed, integrated, and optimized for real-world agricultural applications, ensuring high performance, scalability, and reliability.

1.1.1. Model Development Process

The development of AI models within the NESTLER platform involves a structured process that ensures models are tailored to specific agricultural needs and optimized for performance. The steps are outlined below:

Data Collection and Preprocessing:

- Data Sources: The NESTLER platform collects data from multiple sources, including IoT sensors, satellite imagery, drones and open-source datasets. The data includes real-time environmental metrics such as air temperature, relative humidity, air quality, soil moisture, precipitation, water quality and crop and pests' imagery.
- Data Cleaning: Raw data is often incomplete or noisy. A preprocessing pipeline is applied to clean, normalize, and standardize the data. This step ensures that the AI models receive high-quality inputs.
- Feature Engineering: Key features, such as historical yield data, weather trends, soil health indicators, and pest prevalence, are engineered to provide the most relevant insights into crop health, disease outbreaks, and yield optimization.

Model Training and Selection:

- Algorithm Selection: The NESTLER platform uses various machine learning models based on the problem being addressed. For example, supervised learning models are used for yield predictions, while unsupervised learning helps in clustering farming practices.
- Training Data: Historical datasets with labelled outputs (e.g., crop yield data or disease incidence reports) are used to train the models. In cases where labelled data are limited, semi-supervised learning techniques are applied.
- Model Tuning: During training, hyperparameters such as learning rates, tree depth, and regularization parameters are fine-tuned using techniques like grid search or random search to maximize accuracy.

Testing and Validation:

- Cross-Validation: The models are tested using k-fold cross-validation to ensure that they generalize well on unseen data.
- Performance Metrics: Key metrics such as mean squared error (MSE) for regression models or precision, recall, and F1-score for classification models are used to evaluate model performance.

- Error Analysis: Any misclassifications or inaccurate predictions are analyzed to identify model weaknesses, which informs further adjustments and retraining.

1.1.2. Model Integration with NESTLER Platform

Once models are developed, they need to be seamlessly integrated into the NESTLER platform for real-time use by stakeholders. This involves architectural planning and coordination between data pipelines, cloud services and graphical user interfaces.

Integration with Data Pipelines:

- Data Ingestion: The platform's AI models are connected to real-time data streams via APIs that continuously feed sensor data, satellite imagery, and other inputs.
- Real-Time Analytics: To provide real-time insights, the models are optimized to process data within milliseconds. For example, models detecting pest outbreaks based on drone images are designed to verify as soon as potential threats are identified.

Cloud-based Deployment:

- Scalability: The AI models are deployed on a (Kubernetes-based) cloud infrastructure to ensure scalability. This allows them to handle increasing workloads efficiently coming from multiple regions and farms, supporting global agricultural stakeholders.
- High Availability and Redundancy: The models can be replicated across different node or/and regions in the cloud infrastructure to ensure high availability and to avoid downtime. Whenever one node becomes unavailable, others take over to ensure uninterrupted service.

User Interface Integration:

- Dashboard Visualization: After processing data, the model outputs are presented on interactive dashboards that provide farmers with actionable insights. For instance, alerts related to disease spread risk are presented through an intuitive, time-enabled color map, making the information easily accessible even to users without technical backgrounds.

1.2. Effectiveness of the Models in the NESTLER Platform

The NESTLER platform utilizes a combination of AI models to address various challenges in agriculture, food security, and environmental monitoring. These models leverage big data gathered from IoT devices, satellite imagery, and environmental sensors to provide real-time analytics and decision support systems (DSS) for stakeholders in the agricultural sector. Below is a detailed breakdown of each type of AI model employed in the platform, highlighting their effectiveness and how they contribute to the platform's goals.

1.2.1. Machine Learning Algorithms

Machine Learning (ML) algorithms play a pivotal role in predicting and analyzing various agricultural metrics, such as crop yields, poultry and livestock health, and pest outbreaks. NESTLER utilizes both supervised and unsupervised Learning algorithms to optimize its predictions.

1.2.1.1. Supervised Learning

NESTLER platform leverages both supervised learning models and foundational pretrained models to enhance agricultural decision-making. Algorithms like Random Forest, Gradient Boosting Machines (GBM), and XGBoost are trained on historical datasets including weather data, soil composition, crop growth patterns and pest infestation trends to predict crop yields and monitor livestock health with high accuracy. Additionally, foundational pretrained deep learning models such as YOLO and MobileNet are fine-tuned for real-time disease and pest recognition in crops, poultry and livestock.

Regarding the development process, the platform ingests large-scale historical and real-time agricultural data, including variables such as rainfall, temperature, soil moisture, crop yield history, satellite imagery, and pest/disease reports. This data is enriched with geospatial and IoT sensor inputs for comprehensive analysis.

In terms of the model training, supervised learning and pretrained Vision models are used. Supervised Learning is applied in historical datasets and trains predictive models (e.g., Random Forest, XGBoost) to forecast crop yields, disease risks, and optimal planting times based on weather and soil conditions. Pretrained Vision (foundational AI) models such as YOLO and MobileNet are fine-tuned for disease and pest detection using image datasets, enabling real-time crop health monitoring.

Afterwards, the trained models are deployed across pilot regions via cloud-based APIs or edge devices (e.g. drones, mobile apps) and the farmers receive actionable insights, such as yield forecasts, disease alerts and irrigation recommendations, optimizing planting schedules and resource allocation.

As a result of above, the platform provides with precise yield forecasts based on data-driven predictions (yield prediction) and predicts disease outbreaks in livestock based on health records and environmental factors.

1.2.1.2. Unsupervised Learning

Unsupervised learning models are effective for clustering and categorizing data, allowing the NESTLER platform to identify trends in farming practices and categorize them based on performance metrics.

During the development process, the platform collects unlabeled datasets, such as sensor data from soil moisture or atmospheric conditions and aggregate them. Moreover, clustering algorithms such as K-Means or Hierarchical Clustering analyze these datasets to find hidden patterns or categorize farming techniques based on their effectiveness. The platform detects anomalies in soil data or crop health that could indicate early signs of stress or disease.

1.2.2. Computer Vision Models

Computer Vision models are employed in the NESTLER platform to automate the detection of crop diseases and pests, and diseases in poultry through image analysis, making it easier for farmers to address issues early. These models analyze images captured by drones and ground-based cameras to identify visible signs of crop diseases, pests, nutrient deficiencies or diseases in poultry and livestock farms.

The steps of the development process consist of the following ones:

- Image Dataset Collection: Large datasets of 13 labelled images showing healthy crops and poultry, diseased crops/poultry and pest infestations are compiled. Also, several publicly available datasets have been utilized.
- Convolutional Neural Networks (CNNs): The models are fine-tuned on the dataset to recognize patterns indicative of diseases or pests.
- Foundational Models (Pretrained): These models are pretrained on vast and diverse datasets to learn general features and representations, which can then be fine-tuned for specific tasks such as disease detection, pest identification, or other farming applications. By leveraging transfer learning, they reduce the need for large, labelled datasets while improving accuracy and efficiency by recognizing complex patterns.
- Real-Time Analysis: The models are deployed in the field, where drones and cameras capture images. The platform uses these images to provide real-time insights about crop, poultry, fish and wildlife health.

By integrating the above models in the platform, the farmers receive insights about diseases such as blight or leaf rust, allowing them to take corrective actions before significant crop loss occurs. Also, these models enable the platform to provide pest monitoring – the early identification of pests enables rapid intervention, preventing the spread of infestations.

1.2.3. Predictive Analytics Models

Predictive Analytics models leverage historical data to forecast outcomes such as crop yields, pest outbreaks, and weather-related risks. These models allow farmers to plan effectively by forecasting agricultural outcomes based on various environmental and historical factors.

The development flow includes the following steps:

- Historical Data Analysis: Predictive models are trained on datasets that include previous crop yields, weather conditions, and pest outbreaks.
- Algorithm Development: Machine learning and statistical techniques are employed for time-series forecasting and predictive modelling in agriculture. Traditional methods like ARIMA (Auto Regressive Integrated Moving Average) and regression analysis are used to forecast trends,

seasonal patterns, and future agricultural events. Additionally, statistical classification models such as Random Forest, XGBoost, and Support Vector Machines enhance predictive accuracy by analyzing complex, high-dimensional datasets. These models help in crop yield prediction, disease risk assessment, and pest outbreak forecasting, enabling data-driven decision-making for farmers and agronomists.

- Validation: The model's predictions are validated by comparing forecasts with actual outcomes from past agricultural seasons.

By leveraging these models, the farmers can adjust their planting schedules according to the expected weather conditions, thereby improving the crop yield. Furthermore, the adoption of these models in the platform helps to optimize the use of water, fertilizers and pesticides, simultaneously reducing both costs and environmental impact.

2. NESTLER AI Models Contribution to Food Security

The integration of AI models within the NESTLER platform has made substantial contributions toward addressing key challenges and enhancing food security across Africa. Through predictive analytics for crop and livestock management, effective disease detection and management strategies, optimized resource use in agriculture, innovative insect protein research, improved supply chain logistics, and enhanced decision-making processes driven by stakeholder feedback.

These technological interventions not only support agricultural productivity but also promote a more resilient food system capable of withstanding future challenges. As a result of these advancements, farmers are better equipped to face challenges related to climate change, resource scarcity, and market fluctuations while promoting sustainable practices that benefit both local communities and the environment. The success of these initiatives underscores the potential of AI technologies in creating resilient food systems capable of meeting future demands while ensuring food security across diverse contexts in Africa.

2.1. Predictive Analytics for Crop and Livestock Management

Predictive analytics forms the backbone of decision-making in modern agriculture. In the NESTLER project, machine learning algorithms were employed to analyze vast datasets from agricultural practices, enabling farmers to make informed decisions regarding crop management and livestock care.

Implementation: AI models were trained on historical data, weather patterns, soil conditions, and crop performance metrics. This comprehensive data analysis allowed the models to forecast potential outcomes based on real-time inputs and historical trends. For instance, farmers could predict crop yields by inputting variables such as planting dates, irrigation levels, and fertilizer application rates.

Benefits:

- Informed Decision-Making: By providing accurate forecasts, farmers could optimize their planting and harvesting schedules, ensuring that crops are cultivated under the best possible conditions.
- Resource Allocation: Predictive analytics enabled better resource allocation by identifying which crops would yield the highest returns based on current conditions.
- Risk Mitigation: The ability to forecast adverse conditions, such as droughts or pest infestations, allowed farmers to take preventive measures, reducing potential losses.

Case study: In Nigeria and Uganda pilot regions, farmers reported a marked improvement in yield predictions due to the implementation of these AI techniques. The ability to anticipate market demands also helped them adjust their production strategies accordingly.

2.2. Disease Detection and Management

The threat of zoonotic diseases poses a significant risk to both human health and agricultural productivity. The NESTLER platform utilized computer vision models to identify pests and diseases in crops through advanced image analysis techniques.

Implementation: Drones equipped with high-resolution cameras captured images of fields, which were then processed using AI algorithms. These algorithms were trained to detect early signs of diseases or pest infestations by analyzing color patterns, texture changes, and other visual indicators.

Benefits:

- Timely Alerts: Farmers received real-time alerts about potential threats to their crops, enabling them to take immediate action.
- Economic Impact Reduction: Early detection minimized the economic impact of disease outbreaks on food production by allowing for targeted interventions rather than broad-spectrum treatments.
- Improved Crop Health: Continuous monitoring facilitated better crop management practices, leading to healthier plants and increased yields.

Case study: In Cameroon, the deployment of computer vision technology led to a reduction in crop losses due to pests and diseases within the first year of implementation. Farmers were able to respond more effectively to threats as they emerged.

2.3. Optimizing Resource Utili

AI-driven models facilitated precision agriculture by analyzing soil health, moisture levels, and nutrient availability. This information allowed farmers to apply fertilizers and to irrigate more efficiently.

Implementation: Using IoT sensors placed throughout fields, data on soil conditions was collected continuously. Machine learning algorithms processed this data to provide actionable insights regarding when and how much water or fertilizer should be applied.

Benefits:

- Waste Reduction: By applying resources only when necessary and in precise amounts, farmers significantly reduced waste.
- Sustainable Practices: The reduction in chemical inputs supports environmental health while ensuring that crops receive the necessary nutrients for optimal growth.
- Cost Savings: Efficient resource use translated into lower operational costs for farmers.

Case study: In Ethiopia's pilot, farmers who adopted precision agriculture techniques reported a decrease in water usage while maintaining crop yields. This not only improved profitability but also contributed positively to local water conservation efforts.

2.4. Insect Protein Research

The NESTLER project focused on researching insect protein as an alternative feed source for livestock and aquaculture. AI models were used to analyze the nutritional value of various insect species and optimize feed recipes.

Implementation: Research teams utilized machine learning algorithms to evaluate the growth rates of different insect species under varying conditions. This data informed decisions about which insects would provide the best nutritional profiles for livestock feed.

Benefits:

- Sustainable Protein Source: Insects require significantly less land and water than traditional livestock feed sources.
- Waste Reduction: Utilizing organic waste as feed for insects contributes to a circular economy model.
- Nutritional Benefits: Insect protein is rich in essential amino acids, making it an excellent supplement for animal diets.

Case study: In Kenya, trials involving black soldier flies demonstrated that livestock fed with insect-based diets showed improved growth rates compared to those fed conventional feeds. This research has paved the way for broader adoption of insect protein in animal husbandry across Africa.

3. NESTLER Platform Assessment from the End-User Perspective

3.1. Methodology

The methodology employed in the NESTLER project is critical to understanding the effectiveness of its interventions aimed at enhancing food security across various African countries. This subsection outlines the data collection processes, feedback mechanisms from pilot sites, and the key performance indicators (KPIs) used to assess the platform's performance.

As a key component of the NESTLER project, data collection facilitated the evaluation of its impact in diverse pilot sites across six African countries:

- Cameroon
- Uganda
- Nigeria
- Rwanda
- Kenya, and
- Ethiopia

The approach involved gathering quantitative and qualitative data and feedback from various stakeholders, including farmers, policymakers, and researchers.

3.1.1. Data Collection Techniques

A combination of four techniques was used to gather data. Especially:

- **Surveys and Questionnaires**: Structured surveys were distributed to the stakeholders involved in each pilot site. These surveys collected information on agricultural practices, resources utilization, and perceptions of AI technologies implemented through the NESTLER platform.
- **Focus Group Discussions**: Facilitated discussions with local communities provided insights into their experiences with the NESTLER platform. These discussions allowed participants to share their challenges and successes in adopting new technologies.
- **Field Observations**: Researchers conducted on-site observations to gather real-time data on farming practices and livestock management. This hands-on approach helped validate survey responses and provided context to the quantitative data collected.
- **Remote Monitoring Technologies**: Synfield and other IoT devices and remote sensing technologies were utilized to collect data on soil health, moisture levels, crop growth, and livestock conditions continuously. This real-time data was crucial for assessing the effectiveness of interventions.

3.1.2. Stakeholder Engagement

Engagement with stakeholders was prioritized throughout the data collection process. Regular interactions with farmers, local government officials, agricultural extension workers, and researchers ensured that diverse perspectives were considered in evaluating the platform's performance.

Pilot site feedback: Feedback from various pilot sites was instrumental in analyzing the platform's performance against real-world challenges faced by farmers and agricultural stakeholders.

Regional focus areas:

- The pilots in Cameroon and Nigeria primarily focused on crop farming practices, where feedback was gathered on how AI tools influenced planting decisions, pest management, and overall crop productivity.
- The pilots in Ethiopia and Rwanda focused on livestock health monitoring and aquaculture outcomes. Feedback highlighted how AI-driven insights improved animal welfare and productivity in aquaculture systems.

Analysis of feedback: The feedback collected was systematically analyzed to identify common issues related to user experience within the NESTLER platform:

- Effectiveness of AI Tools: Users reported varying levels of satisfaction with AI tools for predicting crop yields and managing livestock health.
- Training Needs: Many users expressed a need for additional training resources to maximize their understanding of how to utilize the platform effectively.
- System Usability: Feedback indicated that while many users found the platform intuitive, others faced challenges navigating certain features.

3.1.3. Definition of Key Performance Indicators (KPIs)

To ensure a comprehensive evaluation of the NESTLER platform's performance, a set of Key Performance Indicators (KPIs) was defined. These metrics enabled the measurement of outcomes aligned with the project's core objectives.

Specifically, the KPIs focused on the following areas:

- Overall Satisfaction
- Ease of Navigation
- Data Accuracy
- System Uptime, and
- Usability

3.2. Pilot-level Insights

The NESTLER pilots have implemented various use cases across different regions in Africa, focusing on optimizing agricultural practices and enhancing food security through advanced technologies. The following pilots represents use cases targeted for crop, livestock and human for one health monitoring.

3.2.1. Crop Farming (Cameroon and Nigeria)

In Cameroon and Nigeria, the NESTLER platform aimed to optimize crop farming practices by leveraging real-time soil health data and advanced monitoring technologies. The focus was on enhancing yield management strategies through precise data collection and analysis.

Implementation:
- Real-Time Soil Monitoring: IoT sensors were deployed in fields to monitor soil moisture, nutrient levels, and pH balance. This data was transmitted to farmers via mobile applications, providing them with actionable insights.
- Predictive Analytics: Machine learning algorithms analyzed historical data alongside real-time inputs to forecast crop yields and identify optimal planting times.

Results:
- Improved Yield Management: Farmers reported a significant increase in crop yields due to better-informed decision-making. The ability to monitor soil health in real time allowed for timely interventions.
- Resource Optimization: The platform enabled farmers to apply fertilizers and water more efficiently, reducing waste and lowering costs associated with excessive resource use.

3.2.2. Livestock and Aquaculture (Ethiopia and Rwanda)
In Ethiopia and Rwanda, the focus shifted towards livestock health monitoring and aquaculture management. The integration of IoT technologies aimed to improve disease management practices and enhance overall productivity in livestock farming.

Implementation:
- IoT Sensors for Livestock Monitoring: Sensors were installed on livestock to track health indicators such as temperature, activity levels, and feeding patterns. This data was analyzed to detect early signs of illness.
- Aquaculture Monitoring: Similar sensor technologies were employed in aquaculture settings to monitor water quality parameters like temperature, pH levels, and oxygen content.

Results:
- Enhanced Disease Management: The real-time monitoring capabilities allowed farmers to identify health issues early, leading to timely veterinary interventions that improved livestock survival rates.
- Increased Productivity: Farmers reported improved productivity in both livestock and aquaculture due to better health management practices facilitated by the platform.

3.3. Questionnaire Assessment Report
The first phase evaluation of the NESTLER platform gathered feedback from nominees across three stakeholder groups—Researchers, Policymakers, and Farmers in six pilot countries: Cameroon, Ethiopia, Kenya, Nigeria, Rwanda, and Uganda. The questionnaire structure is available in Annex (see 7 ANNEX I). This report presents quantitative summaries of five key performance indicators (KPIs) as those in subsection 3.1.3.

3.3.1. Stakeholder Role-Based KPI Score

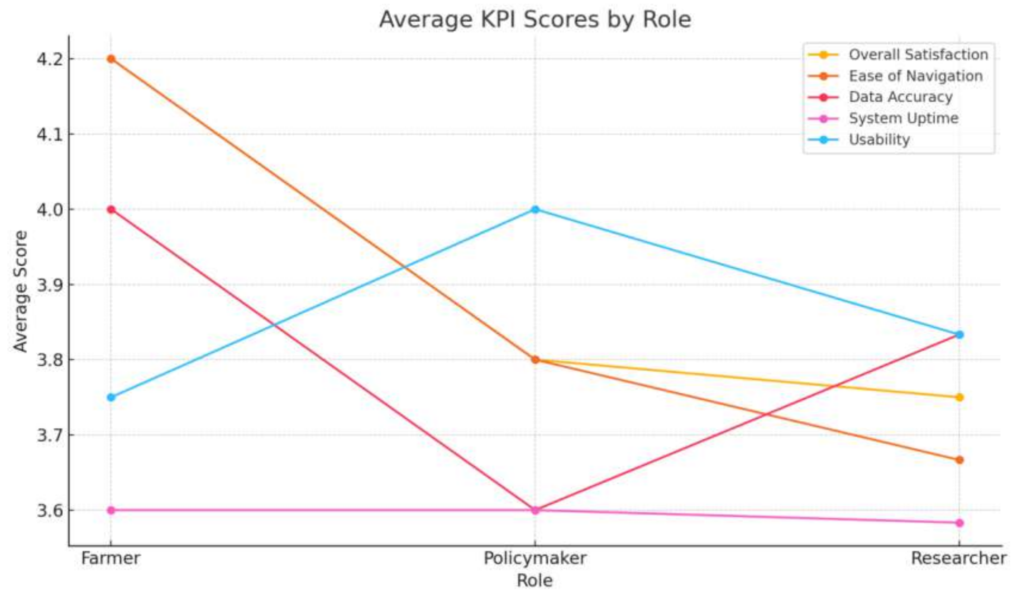

Figure 2: Average KPI Scores by Stakeholder Role (1: Very Low to 5: Very High)

Table 1. KPI score explanation

Score	Meaning
1	Very Low / Poor
2	Low / Fair
3	Moderate
4	High / Good
5	Very High / Excellent

Ease of Navigation received the highest score among farmers, with an average rating of 4.20**.** This suggests that the platform's user interface is particularly intuitive and accessible for this user group, likely reflecting alignment with their operational needs and digital familiarity. The positive feedback in this area underscores the effectiveness of the NESTLER platform's design in supporting day-to-day agricultural tasks.

In contrast, Data Accuracy presents a more varied perception across user groups. While farmers rated it relatively high at 4.00, policymakers provided the lowest score at 3.60. This disparity stem from the differing expectations and use cases between the two groups. Farmers typically rely on operational data for immediate, practical decision-making in the field, where slight inaccuracies may have limited impact. On the other hand, policymakers may require highly reliable, standardized, and validated data to support strategic planning, policy formulation, and large-scale reporting.

Furthermore, System Uptime emerged as the lowest-rated KPI overall, with scores averaging around 3.60 across all roles. This score can be attributed to the NESTLER platform's ongoing development and integration activities in WP3, WP4 during the evaluation period. As new features were being implemented and platform components integrated, occasional interruptions or temporary instabilities may have affected users' perception of platform availability. While such fluctuations are expected during active development phases, they underscore the importance of rigorous testing and stabilization prior to full-scale deployment to ensure consistent system performance and user confidence.

3.3.2. Country-Level Overall Satisfaction

The following table and figure depict the average overall satisfaction per country as those measured in the first assessment of the NESTLER platform.

Table 2 – Average overall satisfaction per country

Country	Average Overall Satisfaction
Cameroon	3.7
Ethiopia	4.2
Kenya	4.0
Nigeria	3.8
Rwanda	3.5
Uganda	3.8

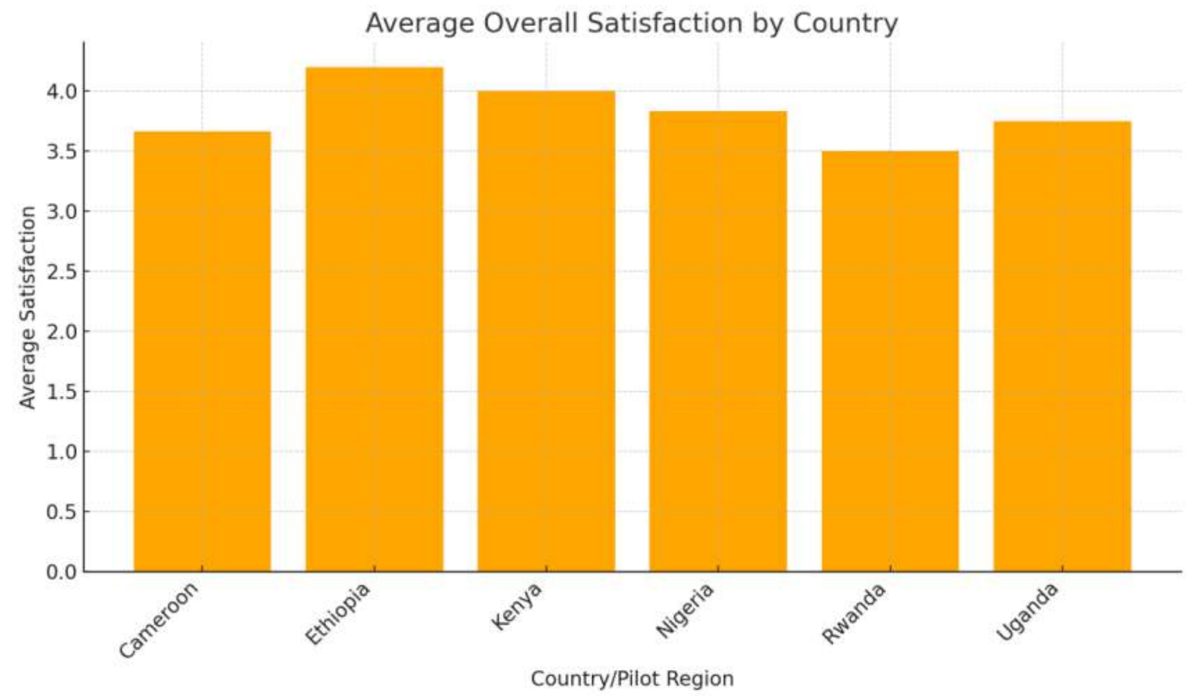

Figure 3: Average Overall Satisfaction by Country (1: Very Low to 5: Very High)

Among the evaluated countries, Ethiopia reports the highest overall satisfaction with a score of 4.2, indicating a strong alignment between user expectations and platform performance in its pilot sites. This suggests that the deployment in Ethiopia has been particularly effective in meeting the needs of its users. Conversely, Rwanda reports the lowest satisfaction score at 3.5, highlighting the need for closer examination of that specific outcome. This outcome may point to challenges related to user experience, training adequacy, or contextual factors unique to the Rwandan pilot.

The above results reflect users' initial perceptions of the platform's key performance indicators within diverse national contexts. By analyzing satisfaction levels across countries, valuable insights can be drawn regarding regional strengths, areas for improvement, and potential variations in user expectations or implementation environments. This comparative view serves as a foundational reference for tracking progress and guiding future enhancements of the platform.

3.3.3. Country-Level KPI Heatmap

The heatmap provides a visual comparison of the average scores per country across all five Key Performance Indicators (KPIs), offering insight into national-level performance trends.

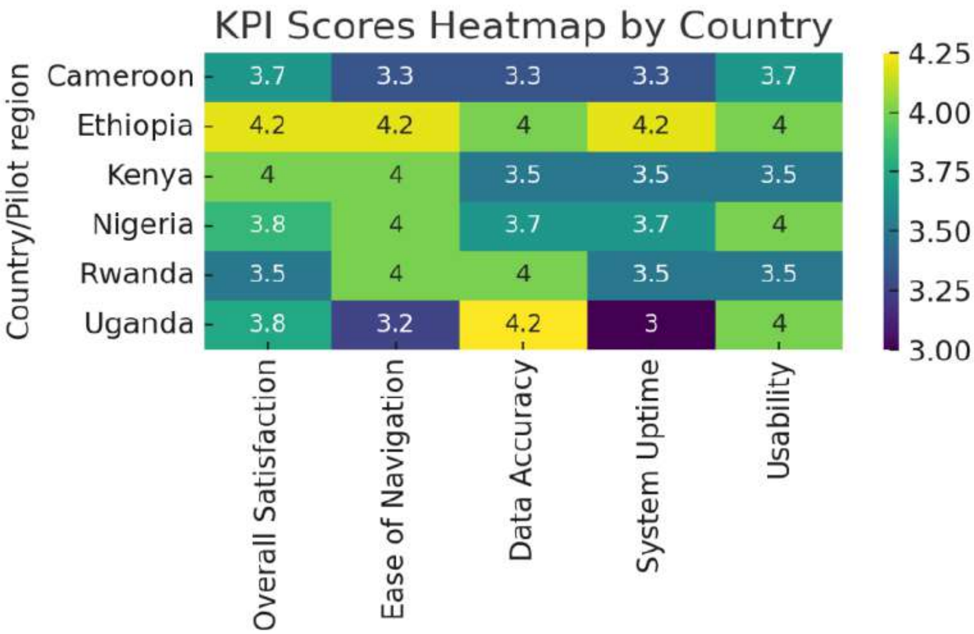

Figure 4: KPI Scores Heatmap by Country (1: Very Low to 5: Very High)

Ethiopia emerges as a consistent high performer, with average scores equal to or exceeding 4.0 on most indicators. This suggests a strong alignment between platform functionality and user expectations in the Ethiopian pilots, potentially reflecting effective deployment, training, and local engagement.

Uganda demonstrates particularly high satisfaction with Data Accuracy, scoring 4.25, which indicates user confidence in the reliability of the information provided by the NESTLER platform. However, its moderate score in System Uptime points to challenges related to platform stability or connectivity, which may hinder consistent access to services.

In contrast, Rwanda records moderate performance in both Ease of Navigation and System Uptime, with scores around 3.5. These results may reflect usability challenges or technical limitations that impact the user experience. The lower performance in these areas suggests a need for targeted improvements in interface design to enhance user satisfaction and engagement in the Rwandan context.

3.3.4. Recommendations

1. Strengthen Technical Stability and Platform Uptime

One of the recurring concerns across all stakeholder groups especially noted in open comments was intermittent platform loading issues and unresponsive sections, particularly within the pilot-specific interfaces. While the homepage generally performed reliably, several users experienced sections going blank or timing out after initial loads.

To mitigate these challenges, it is essential to prioritize services optimization and network resource management. As a first step, load testing was conducted on the NESTLER platform using the Grafana k6[1] tool under a range of operating conditions. A detailed analysis of the platform's API technical validation is provided in Section 4. Moreover, changes over the deployment topology initially planned and then, applied. Each platform service consists of at least 2 replicas (pods) located in different Kubernetes nodes.

For sure, the usage of real-time monitoring dashboards such as Grafana and periodic audits can also pre-empt issues before they affect end-users. Collectively, these steps will enhance system reliability and reduce friction in user experience.

2. Improve Data Accuracy and Reliability—Especially for Policymakers

Policymakers in the evaluation cohort expressed moderate levels of confidence in the platform's data accuracy. Given the role of policymakers in shaping resource allocation, development strategies, and investment in agricultural systems, ensuring high data fidelity is paramount.

This could be achieved by enhancing the data validation pipeline, both at point-of-entry (sensor and user input level) and during backend processing. Adding automated consistency checks, range validations, and cross-referencing routines with historical baselines or third-party datasets (e.g., meteorological databases) will help filter anomalies. Furthermore, integrating a data confidence indicator or error log traceability feature into the platform could provide policymakers with transparency and a better understanding of uncertainty margins in forecasts or data summaries.

3. Implement Country-Specific Improvements Based on Evaluation Feedback

Even though the overall satisfaction scores were positive, certain pilot countries exhibited moderate ratings, often for context-specific reasons. Tailored improvements for each region are essential to maximize platform adoption and relevance.

Rwanda: Respondents from Rwanda flagged usability challenges, particularly around accessing specific pilot content and interpreting output. To address this, a targeted usability audit will be conducted in collaboration with local stakeholders to pinpoint friction points. Outcomes of this assessment could drive on the development of a localized user interface adaptation, possibly with simplified layout, translated tooltips, or offline summaries to accommodate local user preferences.

Uganda: Technical feedback suggests platform responsiveness was limited in Uganda, likely due to bandwidth constraints or regional infrastructure. To mitigate this, caching mechanisms could be explored, allowing users to load data intermittently and still retain access to key summaries. Additionally, light-weight data visualizations and dashboards could improve performance on low-bandwidth connections.

These geographically responsive adjustments will improve stakeholder satisfaction, enable deeper engagement, and ensure the platform performs consistently in heterogeneous digital environments.

[1] https://k6.io/

4. Strengthen Stakeholder Training and Onboarding Materials

Several evaluators, especially first-time users, indicated a need for clearer instructions or demonstrations to guide their interaction with the platform. Even though the platform GUI is generally user-friendly, its multi-functional nature (e.g., dashboards, data layers, predictive tools) requires initial guidance.

To tackle this, a set of structured training resources could be added:

- Quick-start guides with screenshots and short task-specific workflows.
- Material such as short videos (2–3 minutes each) for platform navigation, data entry, and dashboard interpretation.
- Onboarding webinars at key stages, especially when major updates are introduced.

These materials could be developed in both English and, where applicable, regional languages, and should remain accessible from within the platform under a “Help” tab.

5. Establish Continuous Feedback Loops for Ongoing Improvement

The evaluation exercise has been instrumental in identifying platform performance issues and capturing user preferences. However, relying solely on formal evaluation cycles results in significant delays between the emergence of insights and the implementation of corresponding actions.

The evaluation exercise has proven valuable for surfacing platform performance issues and user preferences. However, waiting until formal evaluations are conducted creates long gaps between insight and action. To maintain iterative platform improvement, mechanisms that enable users to provide feedback will be evaluated:

- A “Was this helpful?” button under data visualizations or policy summaries.
- A short 1-minute feedback form that can pop up periodically or after a session.
- A contact support chat or ticket-logging system (even if asynchronous) for reporting bugs or feature requests.

Aggregated feedback from these tools can be analyzed monthly to guide low-effort-high-impact updates. It also gives users a direct channel for input, reinforcing the evolution of the NESTLER platform.

4. NESTLER Platform Technical Validation

In this section, we provide the technical aspects of the validation procedures, encompassing performance testing. By conducting a technical evaluation, we aim to establish the viability and robustness of the use case implementations.

4.1. Performance validation

Performance testing refers to the testing related to verifying the NESTLER platform's performance and monitoring how it behaves under stress. Therefore, we can say that performance testing is concerned with the following metrics:

- **Reliability:** Determine the error rate and how it changes under higher loads.
- **Stability:** Measure this through memory and CPU usage.
- **Response time:** Measure the average response time for requests.
- **Scalability:** Determine how the application behaves under different types of loads.

Performance testing is often linked to non-functional requirements. For example, a rest API might be expected to be reliable and stable, e.g., by being able to handle up to 30,000 requests per minute. This is a non-functional requirement that performance testing helps to validate. The goal of performance testing is not to identify buggy implementations but to find performance bottlenecks. This is important as a single performance bottleneck can have a huge impact on the overall API's performance. Therefore, it is crucial to conduct performance testing to detect such issues. In short, the goal of performance testing is to gather insights into the platform's rest APIs performance and communicate these performance metrics to the stakeholders.

The benefits of performance testing would be the following:

- Measure the stability of the software.
- Assess how your application behaves under a normal load, as this is key information for the client.
- Find performance bottlenecks early in the development life cycle.
- Measuring performance helps you to further improve performance because it helps you tailor configurations for components to make them more streamlined.

Benchmarking is aiming a specific goal for the platform under test. The goal is to see how a certain factor (or factors) holds up under a specific amount of a certain stress. It might not be an excessive amount of stress, just a regular amount or for 2x, 10x a normal load. The aim is for a specific result - to be able to claim, "given this traffic, 95% of requests are completed within *n* milliseconds.".

The validation at this stage focuses on the platform API component, which is considered one of the central components of the NESTLER platform, crucial for the operation of the platform. Besides collecting the already mentioned information, the API is also responsible for serving this information to other platform's components, such as the dashboard.

4.1.1. Tool and methodology

The tool that was utilised for the performance testing was Grafana k6[2], an open-source load testing tool designed for testing the performance, reliability, and scalability of applications, especially APIs and web services. Grafana k6 runs multiple iterations in parallel with virtual users (VUs) to see how the API responds under various conditions and load. The key metrics collected during the tests are depicted in Table 3.

Table 3. List of Grafana k6 employed performance metrics

Metric	Explanation
Total Requests	The total number of HTTP requests sent during the test.
Requests per Second	The average number of requests sent per second during the test.
Status Check Success Rate	The percentage of requests that returned a successful status (200).
Data Received	The total amount of data received from the server during the test.
Data Sent	The total amount of data sent to the server during the test.
Average HTTP Request Duration	The average time taken for each request to complete.
Minimum HTTP Request Duration	The shortest time taken for a request to complete.
Maximum HTTP Request Duration	The longest time taken for a request to complete.
HTTP Request Failed	The percentage of requests that failed during the test.
Average Time Waiting for Response	The average time spent waiting for the server's response.
Average Time Blocking	The average time requests spent being blocked before being sent.
Virtual Users (VUs)	The max number of concurrent virtual users simulated during the test.
Total Iterations	The total number of iterations completed by all virtual users.
Iteration Duration	The average duration of each iteration by a virtual user.

Each case highlights different testing scenarios, showcasing the API's behavior under simulated traffic. The following tables present the performance metrics collected during the load testing conducted using k6. Each case corresponds to a separate API endpoint and summarizes key metrics that provide insight into the NESTLER API performance under load.

[2] https://k6.io/

4.1.2. Performance tests

This paragraph reports the results of performance tests conducted on the most load-intensive endpoints, especially the ones that concern the collected measurements coming from SynField and other IoT devices. It's worth mentioning that we didn't make any changes to the API while testing was in progress.

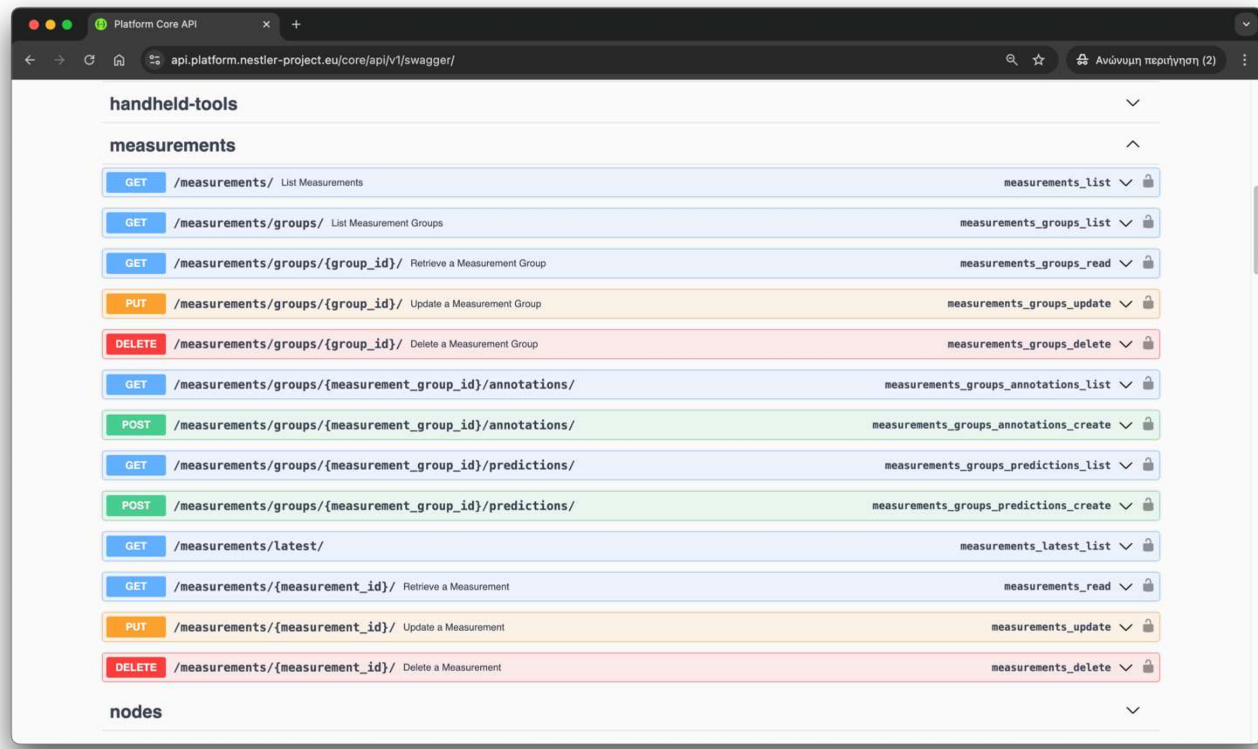

Figure 5: NESTLER Measurements API

4.1.2.1. Measurements Endpoint

The endpoint under testing is the “GET /core/api/v1/measurements/” endpoint, which is considered the most load-intensive endpoint, depending on the number of measurements collected. The tests have been conducted with a variety of VUs and for durations of 1 and 10 seconds. The results are reported in the following paragraphs.

Case 1: Tests with a duration of 1 second

Table 4. Performance results for measurements endpoint for a duration of 1s

Metric	10 VUs	20 VUs	40 VUs	80 VUs	160 VUs
Total Requests	14	27	45	81	160
Requests per Second	1.32	6.11	9.53	12.11	15.25
Status Check Success Rate	100.00%	100.00%	100.00%	100.00%	100.00%
Data Received	423 kB	819 kB	1.4 MB	2.5 MB	5.0 MB
Data Sent	21 kB	42 kB	84 kB	166 kB	334 kB
Average HTTP Request Duration	0.78 s	1.55 s	1.78 s	3.13 s	4.48 s
Minimum HTTP Request Duration	0.24 s	0.44 s	0.60 s	0.79 s	1.15 s
Maximum HTTP Request Duration	1.23 s	3.63 s	4.26 s	5.88 s	9.45 s
HTTP Request Failed	0.00%	0.00%	0.00%	0.00%	0.00%
Average Time Waiting for Response	0.77 s	1.55 s	1.77 s	3.09 s	4.39 s
Average Time Blocking	0.010 s	0.028 s	0.34 ss	0.49 s	1.06 s
Total Iterations	14	27	45	81	160
Iteration Duration	0.81 s	1.58 s	2.12 s	3.62 s	5.55 s

The K6 performance tests were executed for a variable number of VUs, ranging from 10 - 160 for a duration of 1 second, resulting in 14 to 160 HTTP requests. The test for all numbers of VUs achieved a 100% success rate for status checks, indicating that all requests returned the HTTP status code 200 OK.

The average time taken to process each request was 0.78 seconds for 10 VUs, reaching 4.48 seconds for 160 VUs, reflecting the overall responsiveness of the server. For 10 VUs, the shortest request took 0.24 seconds, while the longest took 1.23 seconds, showing some variance in the processing times. For 160 VUs, the same metrics yield results of 1.15 seconds for the shortest request to 9.45 seconds for the longest one.

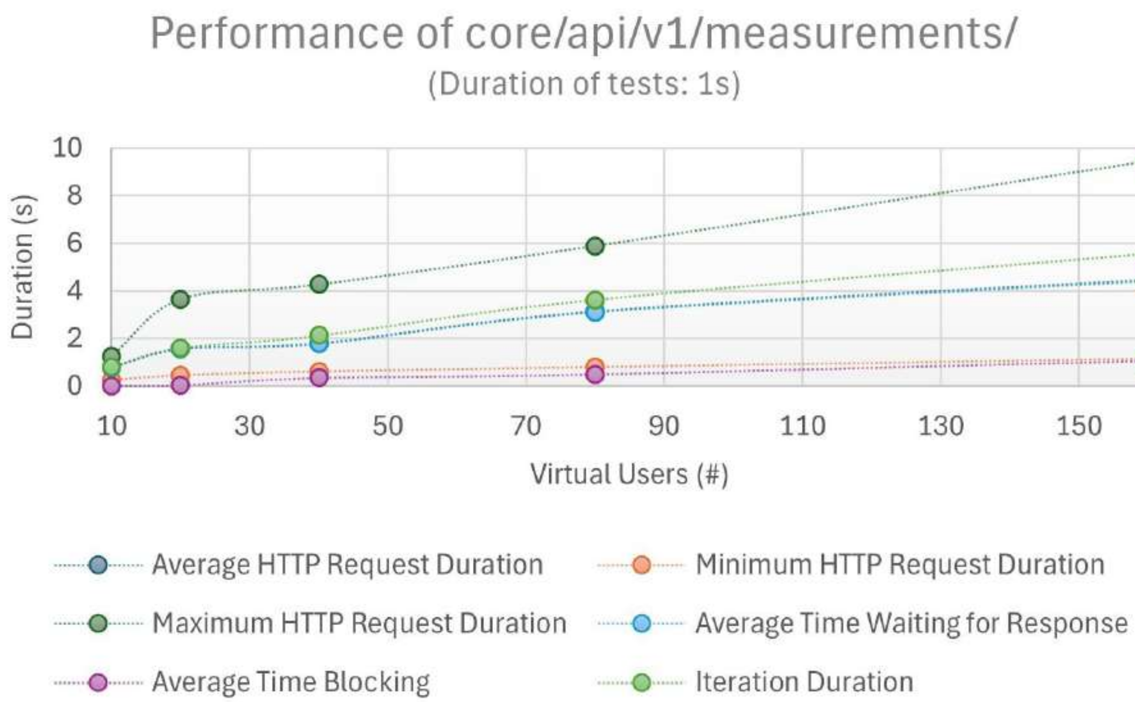

Figure 6. Performance of core/api/v1/measurements endpoint for a duration of 1s

Nestler API handled approximately 1.32 requests per second in the case of 10 VUs, and 15.25 requests per second in the case of 160 VUs, indicating the capacity to process a moderate load. A total of 68 requests were made during the test, reflecting the throughput achieved under the specified load. A total of 423 kB of data was received by the client during the test with 10 VUs, reaching 5 MB in the case of 160 VUs. The client sent from 21 kB to 335 kB of data during the test, reflecting the volume of information exchanged between the client and server.

More information about HTTP Requests' Time Distribution is presented below:

- The average blocked time was 0.11 seconds for 10 VUs (with a maximum of 0.29 seconds), while for 160 VUs the average block time reached 1.06 seconds (with a maximum of 2.68 seconds), indicating occasional delays in processing the requests, which is justified due to the load within 1 second.
- The average connection time was 386.45 µs for 10 VUs and reached 4.21 milliseconds for 160 VUs, reflecting efficient connection establishment.
- The average time spent in TLS handshaking was 8.1 milliseconds for 10 VUs, reaching 1.04 seconds for 160 VUs, relevant for securing HTTPS connections.
- The average time waiting for a response was 770 milliseconds for 10 VUs, reflecting the time taken for the API to process the request and provide the initial data. This indicator reached 4.39 seconds for 160 VUs.
- The average time spent receiving data from the server was 10.94 milliseconds for 10 VUs, reaching 95.6 milliseconds for 160 VUs.

For each scenario, the tests maintained a minimum of 4 active VUs during the execution, with the maximum being the desired number of VUs. The total number of iterations varies on the selected desired number of VUs. Generally, each VU completed one or more requests depending on the configuration.

The test results depict a stable and reliable application performance, achieving a 100% success rate for all requests. An important notice is that the duration of 1 second is really stressing for the platform and was selected to depict the performance of the platform under an intensive load. Data transfer metrics were within acceptable ranges, and the system showed efficient handling of secure HTTPS connections.

Case 2: Tests with a duration of 10 seconds

Table 5. Performance results for measurements endpoint for a duration of 10s

Metric	10 VUs	20 VUs	40 VUs
Total Requests	54	111	133
Requests per Second	4.79	9.32	9.59
Status Check Success Rate	100.00%	100.00%	100.00%
Data Received	1.5 MB	3.2 MB	3.8 MB
Data Sent	29 kB	57 kB	98 kB
Average HTTP Request Duration	0.94 s	0.97 s	2.56 s
Median HTTP Request Duration	0.47 s	0.89 s	2.49 s
Minimum HTTP Request Duration	0.40 s	0.43 s	0.55 s
Maximum HTTP Request Duration	3.20 s	2.26 s	5.65 s
HTTP Request Failed	0.00%	0.00%	0.00%
Average Time Waiting for Response	0.94 s	0.97 s	2.56 s
Average Time Blocking	0.022 s	0.024 s	0.08 s
Total Iterations	54	111	133
Iteration Duration	1.96 s	1.99 s	3.65 s

The K6 performance tests were executed for a variable number of VUs, ranging from 10 to 40 for a duration of 10 seconds, resulting in 54 to 133 HTTP requests. The test for all numbers of VUs achieved a 100% success rate for status checks, indicating that all requests returned the HTTP status code 200 OK.

The average time taken to process each request was 0.94 seconds for 10 VUs, reaching ~2.56 seconds for 40 VUs, reflecting the overall responsiveness of the server. For 10 VUs, the shortest request took 0.4 seconds, while the longest took 3.2 seconds, showing some variance in the processing times. For 40 VUs, the same metrics yield results of 0.55 seconds for the shortest request to 5.65 secs for the longest one.

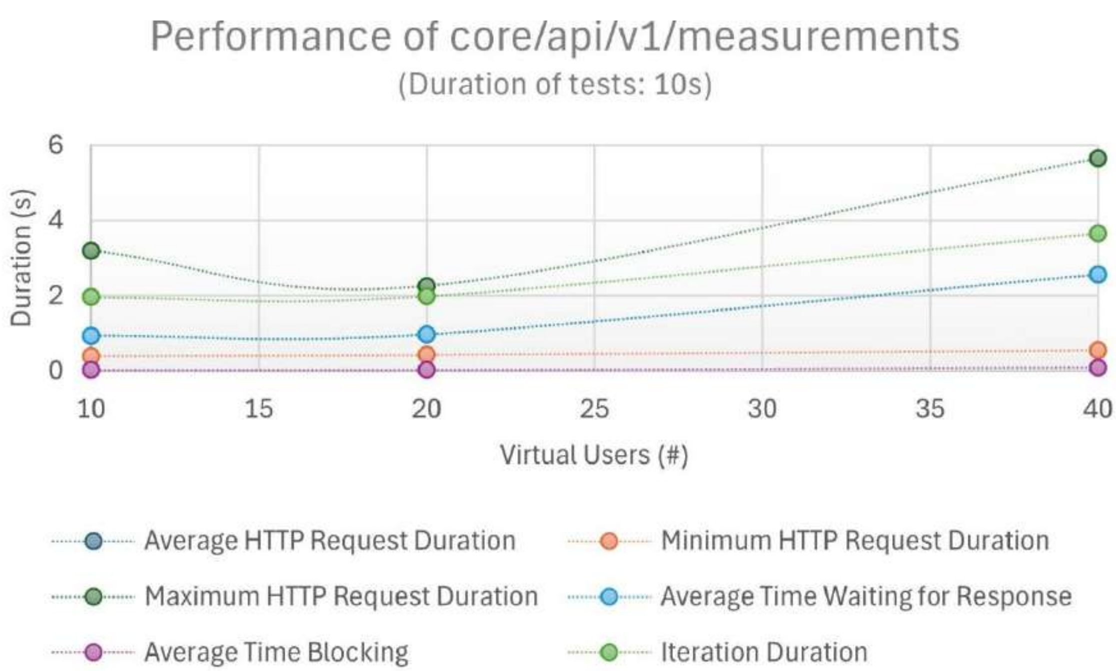

Figure 7. Performance of core/api/v1/measurements endpoint for a duration of 10s

The API handled approximately ~4.79 requests per second in the case of 10 VUs, and 9.59 requests per second in the case of 40 VUs, indicating the capacity to process a moderate load. A total of 54 requests were made during the test for 10 VUs, and 133 requests for 40 VUs, reflecting the throughput achieved under the specified load. A total of 1.5 MB of data was received by the client during the test with 10 VUs, reaching 3.8 MB in the case of 40 VUs. The client sent from 29 kB to 98 kB of data during the test, reflecting the volume of information exchanged between the client and server.

The HTTP Request Time Distribution follows:

- The average blocked time was 0.022 seconds for 10 VUs (with a maximum of 164.62 milliseconds), while for 160 VUs the average block time reached 0.08 seconds (with a maximum of 0.84 seconds), indicating occasional delays in processing the requests.
- The average connection time was 126.93 µs for 10 VUs and reached 350.42 µs for 40 VUs, reflecting efficient connection establishment.
- The average time spent in TLS handshaking was 8.1 msecs for 10 VUs, reaching 1.04 seconds for 160 VUs, relevant for securing HTTPS connections.

- The average time waiting for a response was 8.48 msecs for 10 VUs, reflecting the time taken for the server to process the request and send the initial data. This indicator reached 66.08 msecs for 40 VUs.
- The average time spent receiving data from the server was 839.44 µs for 10 VUs, reaching 3.24 milliseconds for 160 VUs.

For each of the scenarios, the tests maintained a minimum of 4 active VUs during the execution, with the maximum being the desired number of VUs. The total number of iterations varies according to the selected desired number of VUs. Generally, each VU completed one or more requests depending on the configuration. The test results show a stable and reliable application performance, achieving a 100% success rate for all requests. Data transfer metrics were within acceptable ranges, and the system showed efficient handling of secure HTTPS connections.

4.1.2.2. Measurement Groups Endpoint

The endpoint under testing is the "GET /core/api/v1/measurements/groups/" endpoint, which is considered the second most load-intensive endpoint, depending on the number of measurements collected. The tests have been conducted with a variety of VUs and for durations of 1 and 10 seconds. The results are recorded in the following paragraphs.

Case 1: Tests with a duration of 1 second

Table 6. Performance results for measurement groups endpoint for a duration of 1s

Metric	10 VUs	20 VUs	40 VUs	80 VUs	160 VUs
Total Requests	20	30	51	81	160
Requests per Second	12.20	12.53	13.28	14.00	13.86
Status Check Success Rate	100.00%	100.00%	100.00%	100.00%	100.00%
Data Received	1.1 MB	1.6 MB	2.8 MB	4.5 MB	8.9 MB
Data Sent	25 kB	48 kB	93 kB	177 kB	355 kB
Average HTTP Request Duration	0.62 s	1.08 s	1.70 s	2.37 s	5.12 s
Minimum HTTP Request Duration	0.32 s	0.49 s	0.50 s	0.49 s	0.99 s
Maximum HTTP Request Duration	1.10 s	2.29 s	3.64 s	5.09 s	10.42 s
HTTP Request Failed	0.00%	0.00%	0.00%	0.00%	0.00%
Average Time Waiting for Response	0.62 s	1.08 s	1.69 s	2.30 s	4.93 s

Average Time Blocking	0.064 s	0.049 s	0.19 s	0.64 s	1.32 s
Total Iterations	20	30	51	81	160
Iteration Duration	0.68 s	1.13 s	1.88 s	3.02 s	6.45 s

The K6 performance tests were executed for a variable number of VUs, ranging from 10 to 160 for a duration of 1 second, resulting in 20 to 160 HTTP requests. The test for all numbers of VUs achieved a 100% success rate for status checks, indicating that all requests returned the HTTP status code 200 OK.

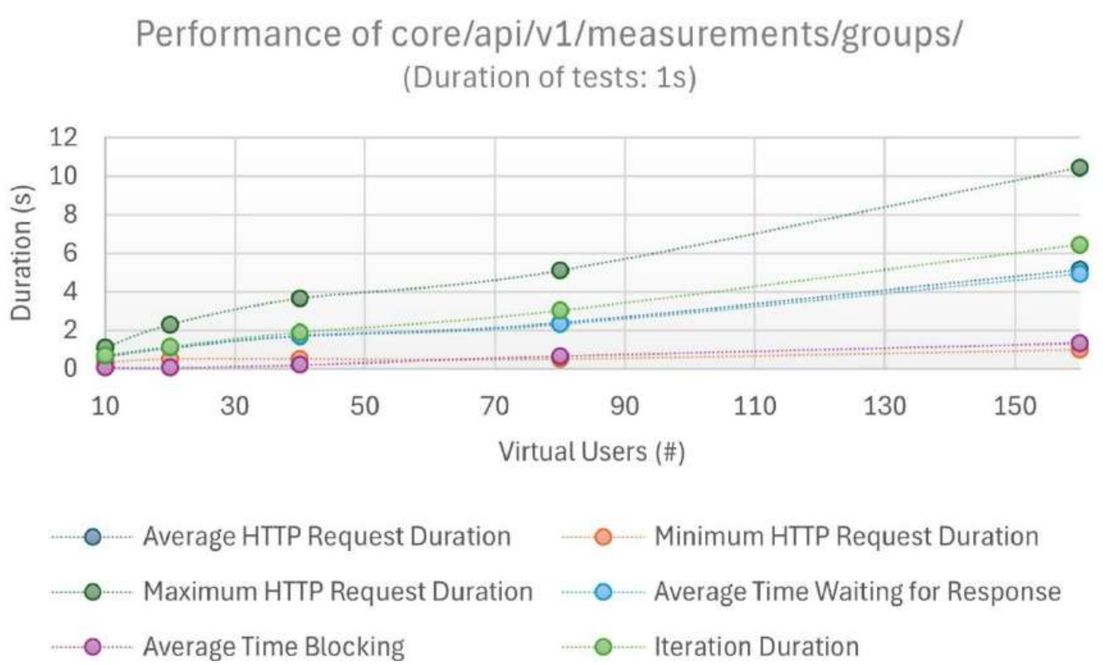

Figure 8. Performance of core/api/v1/measurements/groups endpoint for a duration of 1s

The average time taken to process each request was ~0.62 seconds for 10 VUs, reaching 5.12 seconds for 160 VUs, reflecting the overall responsiveness of the server. For 10 VUs, the shortest request took 0.32 seconds, while the longest took 1.1 seconds, showing some variance in the processing times. For 160 VUs, the same metrics yield results of 0.99 seconds for the shortest request to 10.42 seconds for the longest one.

The API handled approximately 12.2 requests per second in the case of 10 VUs, and 13.86 requests per second in the case of 160 VUs, indicating the capacity to process a moderate load. A total of 20 requests were made during the test with 10 VUs, increasing to 160 for 160 VUs, reflecting the throughput achieved under the specified load. A total of 1.1 MB of data was received by the client during the test with 10 VUs, reaching 8.9 MB in the case of 160 VUs. The client sent from 25 kB to 355 kB of data during the test, reflecting the volume of information exchanged between the client and server.

More information about HTTP Requests' Time Distribution is presented below:

- The average blocked time was 64.02 milliseconds for 10 VUs (with a maximum of 152.67 milliseconds), while for 160 VUs the average block time reached 1.32 seconds (with a maximum of 2.87 seconds), indicating occasional delays in processing the requests, which is justified due to the load within 1 second.
- The average connection time was 386.46 μs for 10 VUs and reached 5.18 milliseconds for 160 VUs, reflecting efficient connection establishment.
- The average time spent in TLS handshaking was 21.81 milliseconds for 10 VUs, reaching 1.3 seconds for 160 VUs, relevant for securing HTTPS connections.
- The average time waiting for a response was 615.53 milliseconds for 10 VUs, reflecting the time taken for the server to process the request and send the initial data. This indicator reached 4.93 seconds for 160 VUs.
- The average time spent receiving data from the server was 1.35 milliseconds for 10 VUs, reaching 193.15 milliseconds for 160 VUs.

For each of the scenarios, the tests maintained a minimum of 4 active VUs during the execution, with the maximum being the desired number of VUs. The total number of iterations varies according to the selected desired number of VUs. Generally, each VU completed one or more requests depending on the configuration.

The test results reflect a stable and reliable application performance, achieving a 100% success rate for all requests. An important notice is that the duration of 1 second is really stressing for the platform API and was selected to depict the performance of the platform under an intensive load. Data transfer metrics were within acceptable ranges, and the system showed efficient handling of secure HTTPS connections.

Case 2: Tests with a duration of 10 seconds

Table 7. Performance results for measurement groups endpoint for a duration of 10s

Metric	10 VUs	20 VUs	40 VUs	80 VUs
Total Requests	70	112	133	191
Requests per Second	6.11	9.75	10.58	12.00
Status Check Success Rate	100.00%	100.00%	100.00%	100.00%
Data Received	3.7 MB	5.9 MB	7.1 MB	10 MB
Data Sent	37 kB	59 kB	112 kB	205 kB
Average HTTP Request Duration	0.57 s	0.80 s	2.35 s	4.31 s

Minimum HTTP Request Duration	0.29 s	0.28 s	0.26 s	0.52 s
Median HTTP Request Duration	0.51 s	0.60 s	1.97 s	4.33 s
Maximum HTTP Request Duration	1.10 s	2.50 s	9.74 s	10.08 s
HTTP Request Failed	0.00%	0.00%	0.00%	0.00%
Average Time Waiting for Response	0.57 s	0.79 s	2.34 s	4.29 s
Average Time Blocking	0.014 s	0.016 s	0.081 s	0.19 s
Total Iterations	70	112	133	191
Iteration Duration	1.58 s	1.79 s	3.43 s	5.50 s

The K6 performance tests were executed for a variable number of VUs ranging from 10 to 80 for a duration of 10 seconds, resulting in 70 to 191 HTTP requests. The test for all numbers of VUs achieved a 100% success rate for status checks, indicating that all requests returned a status of 200. The average time taken to process each request was 0.57 seconds for 10 VUs, reaching 4.31 seconds for 80 VUs, reflecting the overall responsiveness of the server.

For 10 VUs, the shortest request took 0.29 seconds, while the longest took 1.1 seconds, showing some variance in the processing times. For 80 VUs, the same metrics yield results of 0.52 seconds for the shortest request to 10.08 seconds for the longest one.

The API handled approximately ~6.1 requests per second in the case of 10 VUs, and 12 requests per second in the case of 80 VUs, indicating the capacity to process a moderate load. A total of 70 requests were made during the test for 10 VUs, and 191 requests for 80 VUs, reflecting the throughput achieved under the specified load. A total of 3.7 MB of data was received by the client during the test with 10 VUs, reaching 10 MB in the case of 80 VUs. The client sent from 37 kB to 205 kB of data during the test, reflecting the volume of information exchanged between the client and server.

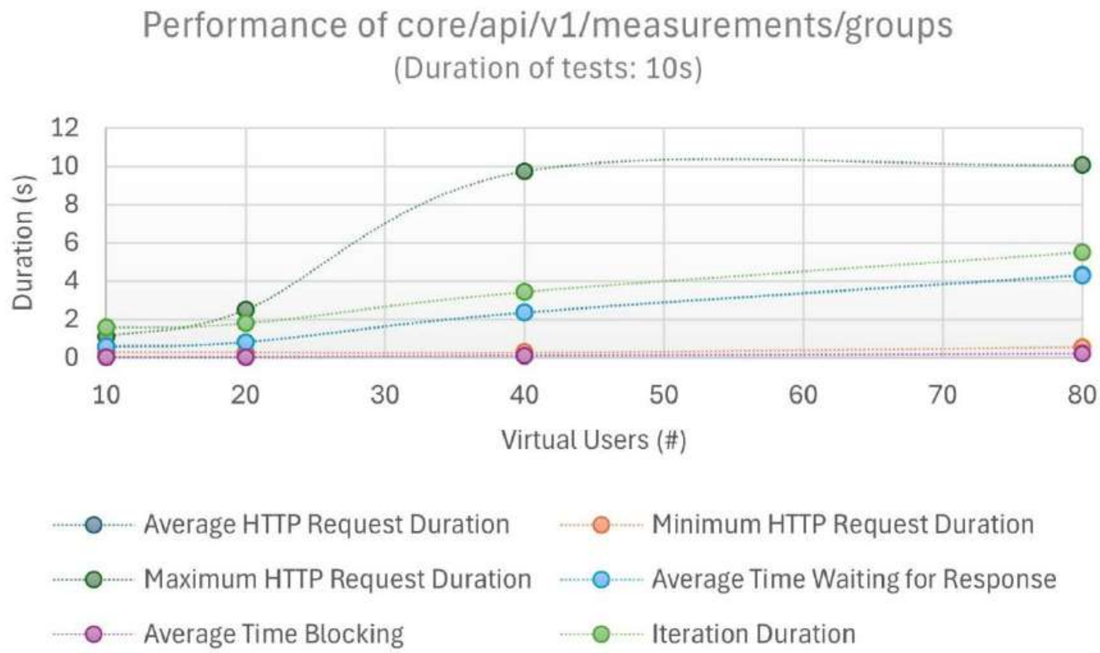

Figure 9. Performance of core/api/v1/measurements/groups endpoint for a duration of 10s

The HTTP Request Time Distribution follows:

- The average blocked time was 14.42 milliseconds for 10 VUs (with a maximum of 214.68 milliseconds), while for 80 VUs the average block time reached 0.08 seconds (with a maximum of 0.84 seconds), indicating occasional delays in processing the requests.
- The average connection time was 90.12 µs for 10 VUs and reached 926.68 µs for 80 VUs, reflecting efficient connection establishment.
- The average time spent in TLS handshaking was 7.74 milliseconds for 10 VUs, reaching 170.64 milliseconds for 80 VUs, relevant for securing HTTPS connections.
- The average time waiting for a response was 566.22 milliseconds for 10 VUs, reflecting the time taken for the server to process the request and send the initial data. This indicator reached 4.29 seconds for 80 VUs.
- The average time spent receiving data from the server was 7.56 milliseconds for 10 VUs, reaching 18.08 milliseconds for 80 VUs.

For each scenario, the tests maintained a minimum of 4 active VUs during the execution, with the maximum being the desired number of VUs. The total number of iterations varies on the selected desired number of VUs. Generally, each VU completed one or more requests depending on the configuration. The test results show a stable and reliable application performance, achieving a 100% success rate for all requests. Data transfer metrics were within acceptable ranges, and the system showed efficient handling of secure HTTPS connections.

4.1.2.3. Measurement by Sensor Endpoint

The endpoint under testing is the “GET /core/api/v1/sensors/{sensor_id}/measurements/**”** endpoint, which is considered the third most load-intensive endpoint, depending on the number of measurements collected. The tests have been conducted with a variety of VUs and for durations of 1 and 10 seconds. The results are recorded in the following paragraphs.

Case 1: Tests with a duration of 1 second

Table 8. Performance results for measurements by sensor endpoint for a duration of 1 second

Metric	10 VUs	20 VUs	40 VUs	80 VUs	160 VUs
Total Requests	10	20	40	81	160
Requests per Second	4.94	8.26	9.33	11.96	15.03
Status Check Success Rate	100.00%	100.00%	100.00%	100.00%	100.00%
Data Received	315 kB	630 kB	1.3 MB	2.5 MB	5.0 MB
Data Sent	21 kB	43 kB	84 kB	169 kB	337 kB
Average HTTP Request Duration	0.57 s	0.84 s	1.59 s	2.86 s	4.13 s
Minimum HTTP Request Duration	0.32s	0.37 s	0.50 s	0.80 s	0.60 s
Maximum HTTP Request Duration	0.88 s	1.36 s	3.14 s	4.80 s	8.38 s
HTTP Request Failed	0.00%	0.00%	0.00%	0.00%	0.00%
Average Time Waiting for Response	0.57 s	0.84 s	1.59 s	2.78 s	3.97 s
Average Time Blocking	0.13 s	0.74 s	0.34 s	0.50 s	1.20 s
Total Iterations	10	20	40	80	160
Iteration Duration	1.69 s	1.91 s	2.94 s	4.37 s	6.34 s

The K6 performance tests were executed for a variable number of VUs, ranging from 10 to 160 for a duration of 1 second, resulting in 10 to 160 HTTP requests. The test for all numbers of VUs achieved a 100% success rate for status checks, indicating that all requests returned the HTTP status code 200 OK.

The average time taken to process each request was 0.57 seconds for 10 VUs, reaching 4.13 seconds for 160 VUs, reflecting the overall responsiveness of the server. For 10 VUs, the shortest request took 0.32 seconds, while the longest took 0.88 seconds, showing some variance in the processing times. For 160 VUs, the same metrics yield results of 0.6 secs for the shortest request to 8.38 secs for the longest one.

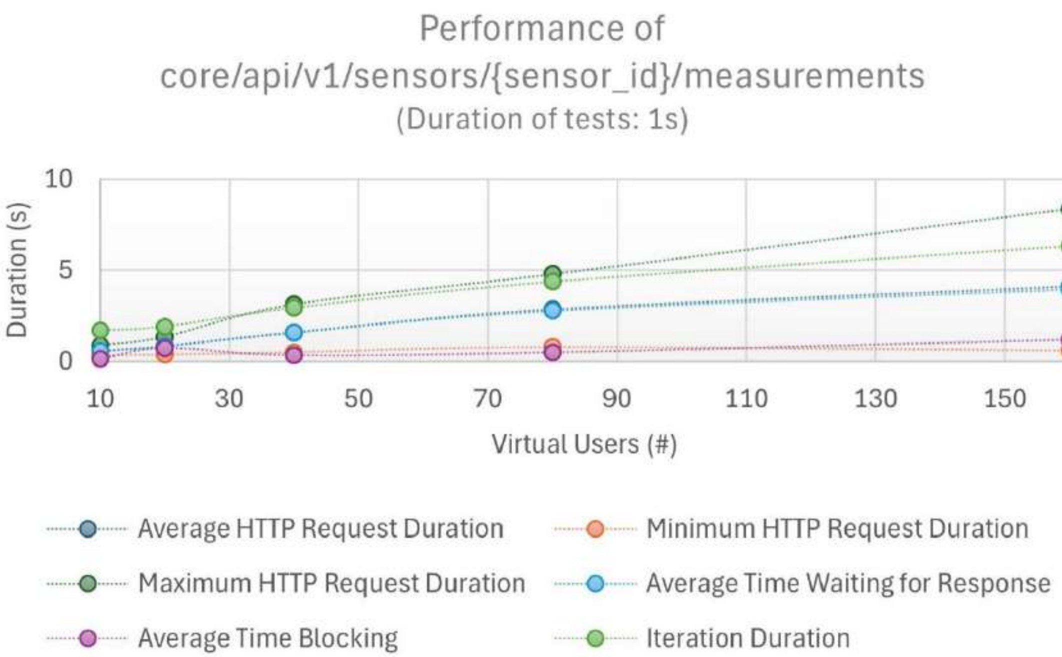

Figure 10. Performance of core/api/v1/sensors/{sensor_id}/measurements endpoint (duration: 1s)

The API handled approximately 4.94 requests per second in the case of 10 VUs, and 15.03 requests per second in the case of 160 VUs, indicating the capacity to process a moderate load. A total of 10 requests were made during the test with 10 VUs, increasing to 160 for 160 VUs, reflecting the throughput achieved under the specified load. A total of 315 kB of data was received by the client during the test with 10 VUs, reaching 5 MB in the case of 160 VUs. The client sent from 21 kB to 337 kB of data during the test, reflecting the volume of information exchanged between the client and server.

More information about HTTP Requests' Time Distribution is presented below:

- The average blocked time was 126.18 milliseconds for 10 VUs (with a maximum of 143.25 milliseconds), while for 160 VUs the average block time reached 1.2 seconds (with a maximum of 2.6 seconds), indicating occasional delays in processing the requests, which is justified due to the load within 1 second.
- The average connection time was 767.74 µs for 10 VUs and reached 5.14 milliseconds for 160 VUs, reflecting efficient connection establishment.
- The average time spent in TLS handshaking was 30.92 milliseconds for 10 VUs, reaching 1.13 seconds for 160 VUs.
- The average time waiting for a response was 570 msecs for 10 VUs, reflecting the time taken for the server to process the request and send the initial data. This indicator reached 3.97secs for 160 VUs.
- The average time spent receiving data from the server was 1.3 milliseconds for 10 VUs, reaching 160.22 milliseconds for 160 VUs.

For each of the scenarios, the tests maintained a minimum of 4 active VUs during the execution, with the maximum being the desired number of VUs. The total number of iterations varies according to the selected desired number of VUs. Generally, each VU completed one or more requests depending on the configuration.

The test results show a stable and reliable application performance, achieving a 100% success rate for all requests. An important notice is that the duration of 1 second is really stressing for the platform and was selected to depict the performance of the platform under an intensive load. Data transfer metrics were within acceptable ranges, and the system showed efficient handling of secure HTTPS connections.

Case 2: Tests with a duration of 10 seconds

Table 9. Performance results for measurements by sensor endpoint for a duration of 10s

Metric	10 VUs	20 VUs	40 VUs	80 VUs
Total Requests	74	144	181	237
Requests per Second	6.54	12.99	14.58	16.17
Status Check Success Rate	100.00%	100.00%	100.00%	100.00%
Data Received	2.1 MB	4.1 MB	5.2 MB	6.9 MB
Data Sent	32 kB	63 kB	109 kB	194 kB
Average HTTP Request Duration	0.41 s	0.44 s	1.38 s	2.99 s
Minimum HTTP Request Duration	0.19 s	0.21 s	0.30 s	0.25 s
Median HTTP Request Duration	0.32 s	0.34 s	1.14 s	2.75 s
Maximum HTTP Request Duration	2.60 s	1.23 s	6.65 s	9.98 s
HTTP Request Failed	0.00%	0.00%	0.00%	0.00%
Average Time Waiting for Response	0.41 s	0.44 s	1.38 s	2.96 s
Average Time Blocking	0.016 s	0.017 s	0.069 s	0.14 s
Total Iterations	74	144	181	237
Iteration Duration	1.42 s	1.45 s	2.45 s	4.13 s

The K6 performance tests were executed for a variable number of VUs, ranging from 10 to 80 for a duration of 10 seconds, resulting in 74 to 237 HTTP requests. The test for all numbers of VUs achieved a 100% success rate for status checks, indicating that all requests returned the HTTP status code 200 OK.

The average time taken to process each request was 0.41 seconds for 10 VUs, reaching 2.99 seconds for 80 VUs, reflecting the overall responsiveness of the server. For 10 VUs, the shortest request took 0.19

seconds, while the longest took 2.6 seconds, showing some variance in the processing times. For 80 VUs, the same metrics yield results of 0.25 seconds for the shortest request to 9.98 secs for the longest one.

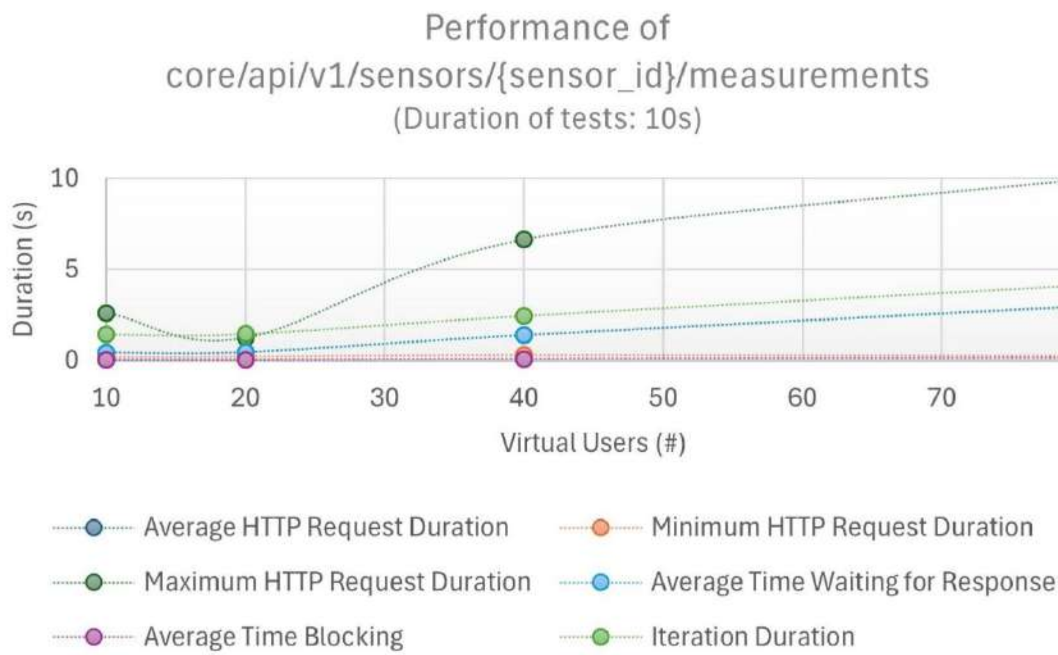

Figure 11. Performance of core/api/v1/sensors/{sensor_id}/measurements endpoint (duration: 10s)

The API handled approximately 6.54 requests per second in the case of 10 VUs, and 16.17 requests per second in the case of 80 VUs, indicating the capacity to process a moderate load. A total of 74 requests were made during the test for 10 VUs, and 237 requests for 80 VUs, reflecting the throughput achieved under the specified load. A total of 2.1 MB of data was received by the client during the test with 10 VUs, reaching 6.9 MB in the case of 80 VUs. The client sent from 32 kB to 194 kB of data during the test, reflecting the volume of information exchanged between the client and server.

The HTTP Request Time Distribution follows:

- The average blocked time was 16.27 milliseconds for 10 VUs (with a maximum of 218.09 milliseconds), while for 80 VUs the average block time reached 0.14 seconds (with a maximum of 0.98 seconds), indicating occasional delays in processing the requests.
- The average connection time was 88.76 µs for 10 VUs and reached 761.23 µs for 80 VUs, reflecting efficient connection establishment.
- The average time spent in TLS handshaking was 10.97 milliseconds for 10 VUs, reaching 128.06 milliseconds for 80 VUs, relevant for securing HTTPS connections.
- The average time waiting for a response was 410.93 milliseconds for 10 VUs, reflecting the time taken for the server to process the request and send the initial data. This indicator reached around 3 seconds for 80 VUs.
- The average time spent receiving data from the server was 1.35 milliseconds for 10 VUs, reaching 22.89 milliseconds for 80 VUs.

For each of the scenarios, the tests maintained a minimum of 4 active VUs during the execution, with the maximum being the desired number of VUs. The total number of iterations varies according to the selected desired number of VUs. Generally, each VU completed one or more requests depending on the configuration. The test results show a stable and reliable application performance, achieving a 100% success rate for all requests. Data transfer metrics were within acceptable ranges, and the system showed efficient handling of secure HTTPS connections.

5. Conclusion

In conclusion, this deliverable has presented a comprehensive evaluation of the NESTLER platform during its first phase of implementation across six African countries. Emphasis was placed on the integration of AI models, IoT technologies, and cloud-based systems to support data-driven agricultural practices and enhance food security outcomes. The platform's performance was assessed in terms of technical functionality, predictive accuracy, and real-time responsiveness in various agricultural contexts.

Moreover, the effectiveness of the platform was demonstrated through multiple pilot use cases focusing on crop farming, livestock health, aquaculture, and insect-based feed innovation. Key strengths identified include accurate data collection, high stakeholder engagement, and improved efficiency in resource use. At the same time, areas such as system stability, training provision, and AI model optimization were highlighted for further improvement. Targeted recommendations were outlined to support future development and ensure broader platform adoption.

These findings will guide the continued evolution of the NESTLER platform, helping to align its technological capabilities with local needs and agricultural realities. Ultimately, the insights gained from this evaluation contribute to the platform's mission to foster a more resilient, inclusive, and sustainable food system across Africa.

6. References

[1] European Commission. (2020). "Food Safety: The Importance of Traceability." Retrieved from European Commission Website. This document outlines the significance of traceability in ensuring food safety standards.

[2] FAO & WHO. (2019). "Framework for Action on Food Security and Nutrition in Protracted Crises." This framework provides guidelines for developing policies that enhance food security and nutrition.

[3] Godfray, H.C.J., et al. (2010). "Food Security: The Challenge of Feeding 9 Billion People." Science, 327(5967), 812-818. This article addresses the challenges facing global food security as the population grows.

[4] Halloran, A., et al. (2016). "Insects as Food: A Global Perspective." Food Security, 8(2), 363-373. This article provides insights into the global perspective on using insects as food and feed.

[5] Intergovernmental Panel on Climate Change (IPCC). (2019). "Climate Change and Land: An IPCC Special Report." This report discusses how climate change impacts agricultural productivity and food security.

[6] Jones, K.E., et al. (2008). "Global Trends in Emerging Infectious Diseases." Nature, 451(7181), 990-993. This study discusses trends in emerging infectious diseases globally, with a focus on zoonotic diseases.

[7] Klerkx, L., & Rose, D.C. (2020). "The Role of Digital Technologies in Sustainable Agriculture." Food Security, 12(2), 393-406. This article examines how digital technologies can support sustainable agricultural practices.

[8] Liakos, K.G., et al. (2018). "Machine Learning in Agriculture: A Review." Sensors, 18(8), 2674. This article provides insights into how machine learning algorithms are transforming agricultural practices.

[9] One Health Sustainability Partnership between EU and Africa for Food Security. (2023). Retrieved from NESTLER_Section1-3_v6_Final.pdf

[10] O'Neill, H., & Kearney, J.F. (2020). "Artificial Intelligence Applications in Livestock Production: A Review." Animal Production Science, 60(8), 1071-1085. This review discusses various AI applications that enhance livestock management practices.

[11] Van Huis, A., et al. (2013). "Edible Insects: Future Prospects for Food and Feed Security." FAO Forestry Paper 171. This report discusses the potential of insects as a sustainable protein source for livestock and aquaculture.

[12] Wolfert, S., et al. (2010). "Big Data in Smart Farming – A Review." Agricultural Systems, 153, 69-80. This paper reviews the role of big data analytics in enhancing smart farming practices.

[13] World Health Organization (WHO). (2020). "Emerging and Re-emerging Infectious Diseases." This report highlights the increasing frequency of infectious disease epidemics in Africa and their impact on public health and productivity.

[14] Zhang, Y., & Wang, C. (2019). "Application of Remote Sensing Technology in Precision Agriculture: A Review." Remote Sensing, 11(12), 1415. This review explores how remote sensing technologies can enhance precision agriculture practices.

[15] Zinsstag, J., et al. (2011). "Integrating Human and Animal Health in a One Health Approach." Preventive Veterinary Medicine, 101(3-4), 208-218. This paper discusses the importance of a One Health approach in addressing zoonotic diseases.

7. ANNEX I: Stakeholders Questionnaire Structure for the NESTLER Platform Assessment

7.1. Introduction

To support the evaluation of the NESTLER platform and gather structured feedback from end users, a dedicated questionnaire was developed. The objective of the questionnaire was to assess various aspects of the user experience, including system usability, functionality, relevance to user needs, and overall satisfaction. The structure of the questionnaire combined mainly quantitative and less qualitative questions, allowing for a comprehensive understanding of user perceptions. The survey was designed to be concise and user-friendly, ensuring high response quality and minimal response fatigue. All responses were collected and used exclusively for the purpose of project evaluation and improvement.

The questionnaire was divided into nine thematic sections, with each section containing one or more related questions.

7.2. Section 1 - Intro

Section 1 requests the participant's email address, followed by the next question.

- **Question 1**: Do you volunteer to participate in the evaluation process?
    - Yes
    - No

7.3. Section 2 – Evaluator information

Section 2 contains four questions related to the evaluator’s personal details, location, and role. The questions are depicted below:

- **Question 2**: Full name (free text)

- **Question 3**: Surname (free text)

- **Question 4**: Country / Pilot region
    - Cameroon
    - Ethiopia
    - Kenya
    - Uganda
    - Rwanda
    - Nigeria

- **Question 5**: Role
    - Researcher
    - Policymaker
    - Farmer

7.4. Section 3 – General platform evaluation

Section 3 pertains to the evaluator's overall impression of the NESTLER platform and includes the following five questions:

- **Question 6**: Overall Satisfaction. How satisfied are you with the NESTLER platform overall? (1: very dissatisfied, 5: very satisfied)
    - 1
    - 2
    - 3
    - 4
    - 5

- **Question 7**: Ease of Navigation & User Interface. Please rate the ease of navigation and clarity of the user interface (1: very difficult, 5: very easy)
    - 1
    - 2
    - 3
    - 4
    - 5

- **Question 8**: Data Accuracy & Reliability. How would you rate the accuracy and reliability of the data provided by the platform? (1: not accurate, 5: very accurate)
    - 1
    - 2
    - 3
    - 4
    - 5

- **Question 9**: System Uptime & Performance. How reliable is the platform in terms of uptime and performance? (1: unreliable, 5: very reliable)
  - 1
  - 2
  - 3
  - 4
  - 5

- **Question 10**: Overall Usability. How easy is it to complete tasks using the platform? (1: very difficult, 5: very easy)
  - 1
  - 2
  - 3
  - 4
  - 5

7.5. Section 4 – Researcher-specific evaluation

Section 4 is dedicated to the evaluation from the researcher's perspective and includes the following questions:

- **Question 11**: Data Integration and Analysis. Rate the effectiveness of the platform in integrating and analyzing diverse data sources (e.g., IoT, satellite data) (1: not effective, 5: very effective)
  - 1
  - 2
  - 3
  - 4
  - 5

- **Question 12**: Advanced Analytics & AI Features. How well do the AI-driven features support predictive modeling and research insights? (1: ineffective, 5: highly effective)
  - 1
  - 2
  - 3
  - 4
  - 5

- **Question 13**: Access to Historical and Real-Time Data. How valuable is the combination of historical data and real-time updates for your research needs? (1: not valuable, 5: extremely valuable)
    - 1
    - 2
    - 3
    - 4
    - 5

7.6. Section 5 – Policymaker-specific evaluation

Section 5 focuses on the evaluation specific to the policymakers' experience and consist of three questions:

- **Question 14**: Policy-Relevant Insights. To what extent does the platform provide insights that are relevant for policy formulation in food security (1: poor relevance, 5: strong relevance)
    - 1
    - 2
    - 3
    - 4
    - 5

- **Question 15**: Alignment with National Priorities. How well does the platform align with and support your country's food security policies? (1: not aligned, 5: perfectly aligned)
    - 1
    - 2
    - 3
    - 4
    - 5

- **Question 16**: Utility of Summarized Dashboards. Rate the usefulness of the platform's dashboards and summary reports for policy briefings. (1: not useful, 5: very useful)
    - 1
    - 2
    - 3
    - 4
    - 5

7.7. Section 6 – Farmer-specific evaluation

This section addresses aspects of the evaluation relevant to farmers and includes the below questions:

- **Question 17**: Actionability of Insights. How useful is the platform in providing actionable insights for crop management and livestock health? (1: not useful, 5: very useful)
    - 1
    - 2
    - 3
    - 4
    - 5

- **Question 18**: Clarity of Visualizations and Recommendations. How clear and understandable are the visualizations and recommendations provided on the platform? (1: not clear, 5: very clear)
    - 1
    - 2
    - 3
    - 4
    - 5

- **Question 19**: Timeliness of Field Data. How would you rate the timeliness of the data (e.g., weather alerts, pest warnings) for your decision-making? (1: very slow, 5: very timely)
    - 1
    - 2
    - 3
    - 4
    - 5

7.8. Section 7 – KPI Assessment and Improvement Suggestions

Section 7 covers the assessment of KPIs and includes suggestions for improvement.

- **Question 20**: Data Accuracy (KPI). How would you rate the accuracy of the data provided by the platform? (1: poor, 5: excellent)
    - 1
    - 2
    - 3
    - 4
    - 5

- **Question 21**: System Uptime (KPI). The reliability of the platform's uptime: (1: unreliable, 5: very reliable)
    - 1
    - 2
    - 3
    - 4
    - 5

- **Question 22**: Usability (KPI). How user-friendly is the platform overall? (1: not usable, 5: highly usable)
    - 1
    - 2
    - 3
    - 4
    - 5

- **Question 23**: AI Model Performance (KPI). How effective are the AI-driven features (predictive analytics, data processing) in delivering accurate insights? (1: ineffective, 5: highly effective)
    - 1
    - 2
    - 3
    - 4
    - 5

- **Question 24**: Provide Overall Suggestions for Improvement (free text)

7.9. Section 8 – Pilot-specific Evaluation

This questionnaire section focuses on the assessment of individual pilot cases, incorporating relevant fields and questions:

- **Question 25**: Country-Specific Challenges and Opportunities - How well does the platform address the unique challenges and opportunities in your country/pilot region? (1: not a lot, 5: very well)
    - 1
    - 2
    - 3
    - 4
    - 5

- **Question 26**: Support for Local Food Security Initiatives - Rate the platform's support for local initiatives aimed at improving food security. (1: not supportive, 5: highly supportive)
    - 1
    - 2
    - 3
    - 4
    - 5

- **Question 27**: Integration with National Policies and Practices - How well is the platform better aligned with your country's policy framework and agricultural practices? (1: not aligned, 5: fully aligned)
    - 1
    - 2
    - 3
    - 4
    - 5

7.10. Section 9 – Final comments

In section 9 participants can provide their additional feedback, if any.

- **Question 28**: Provide additional comments or suggestions (free text)

- **Question 29**: May we contact you for further clarification if needed?
    - Yes
    - No

Upon completion of this section, the participant proceeds to submit their responses, thereby finalizing the questionnaire.