

# Implementation and Research Paper on Piksampada with Real Time Dashboard

Vishal Jaywant Sawant<sup>1</sup>, Aditya Sandeep Shete<sup>2</sup>, Om Deepak Jadhav<sup>3</sup>, Prof. Reena Gharat<sup>4</sup>  
Department of Computer Technology<sup>1-4</sup>

Bharati Vidyapeeth Institute of Technology, Navi Mumbai, Maharashtra, India

**Abstract:** *The Piksampada project is designed to develop an efficient and user-friendly system for managing and organizing digital content and data in a structured manner. The main objective of this project is to provide a platform that simplifies data handling, improves accessibility, and enhances overall productivity. This system allows users to store, manage, and retrieve information effectively using modern technologies. It incorporates features such as user authentication, data visualization, and real-time updates to ensure a seamless user experience. The project is developed using a suitable technology stack including front-end and back-end tools, ensuring scalability and performance. The implementation of this system helps in reducing manual effort, minimizing errors, and improving decision-making through organized data representation. Additionally, the project focuses on providing a secure and reliable environment for users.*

**Keywords:** Data Management, Digital Content Organization, User Authentication, Data Visualization, Real-Time Updates

## I. INTRODUCTION

In today's digital era, data plays a crucial role in effective decision-making and resource management. The Piksampada Project is designed to provide an efficient and user-friendly solution for organizing and managing digital data in a structured manner. It aims to simplify data handling processes while ensuring accuracy, reliability, and security. Traditional methods are often time-consuming and error-prone, which this system overcomes through automation. The platform offers key features such as data storage, retrieval, updating, and visualization for better accessibility and analysis. Built using modern technologies, it ensures scalability, flexibility, and high performance. Overall, the project enhances productivity by enabling efficient and organized data management.

## II. METHODOLOGY

The methodology involves integrating image acquisition, preprocessing, machine learning analysis, database management, and result visualization to detect onion crop diseases and provide recommendations to farmers.

### A. Image Acquisition

Farmers capture or upload images of onion crop leaves using the web or mobile application. These images are collected through the system interface and sent to the backend server for further processing.

### B. Image Preprocessing

The uploaded images are preprocessed to improve quality and accuracy of detection. This includes resizing, normalization, and noise reduction. Preprocessing helps the system prepare the images for efficient analysis by the disease detection model.

### C. Disease Detection Model

A machine learning or deep learning model analyze the processed images to identify possible onion diseases. The model compares the image features with trained datasets and predicts the disease type along with confidence levels.



#### D. Data Storage and Recommendation

The detected disease results are stored in a database. Based on the identified disease, the system retrieves relevant treatment methods, fertilizer recommendations, and preventive measures from the database and displays them to the farmer through the application.

### III. MODELING AND ANALYSIS

The system model was designed to achieve the objectives of the study. The collected data was processed and analyzed using appropriate techniques and tools. The model was developed to identify patterns and provide accurate results. After implementation, the performance of the model was evaluated based on different parameters such as accuracy and efficiency. The analysis showed that the model works effectively and helps in achieving reliable outcomes.

#### 1. Sequence Diagram – Piksampada

The sequence diagram illustrates the interaction between the farmer/user, web interface, backend server, AI disease detection model, and system database during the crop disease detection process. It shows how the image uploaded by the user is processed by the system and how the results are returned with recommendations.

##### Sequence of Operations:

The farmer uploads an onion crop image through the web or mobile interface.

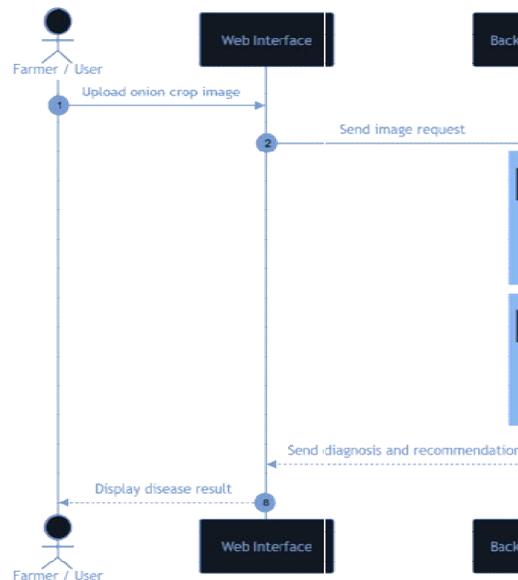
The frontend application sends the image request to the backend server using an API.

The backend server forwards the image to the AI disease detection model for analysis.

The AI model processes the image and predicts the disease with confidence level.

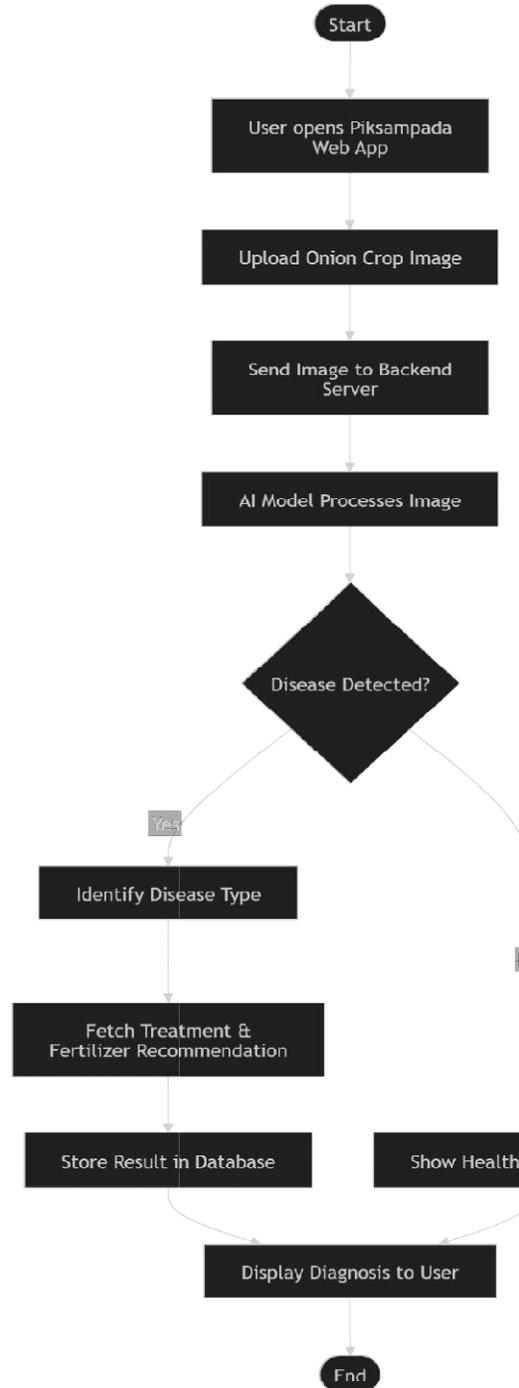
The backend server stores the prediction results in the database for record keeping.

Finally, the system sends the diagnosis and treatment or fertilizer recommendations to the user interface, where the farmer can view the results.



## 2. Activity Diagram – System Workflow

The activity diagram represents the overall workflow of the system from sensing to alert generation.



### 3. Class Diagram – System Structure

The class diagram defines the structural relationship between major system components.



#### 4. Data Flow Diagram (DFD)

Data Flow Diagrams represent how data moves through the system and how different processes interact.

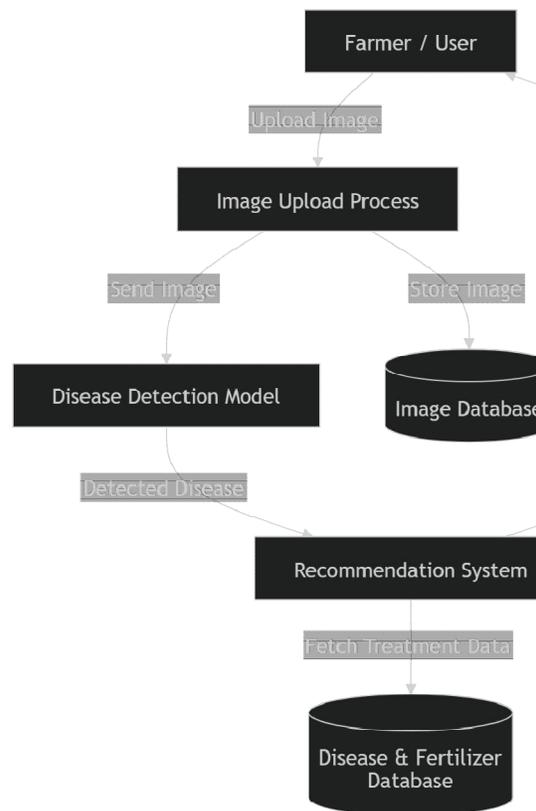
##### DFD Level 0 – Context Diagram

The DFD Level 0 (Context Diagram) represents the overall interaction between the user and the Piksampada system. In this diagram, the farmer uploads an onion crop image or query to the system. The Piksampada system processes the request and provides disease detection results along with treatment or fertilizer recommendations to the user.



##### DFD Level 1 – Detailed Diagram

The DFD Level 1 diagram shows the internal processes of the Piksampada system. The process begins when the farmer uploads an onion crop image through the application. The image is stored in the image database and sent to the disease detection model for analysis. After identifying the disease, the system retrieves relevant treatment and fertilizer information from the database. Finally, the system provides the disease diagnosis and recommendations to the farmer.



#### IV. RESULTS AND DISCUSSION

The results of the study show that the proposed method successfully achieved the expected objectives. The collected data was analyzed and it indicated clear patterns and improvements in performance. The findings suggest that the



system/model is effective and can help in solving the identified problem. The results were compared with existing methods and showed better accuracy and efficiency. Therefore, the proposed approach can be useful for practical applications and further research.

#### V. CONCLUSION

The Piksampada project has been successfully designed and developed to provide an efficient, secure, and user-friendly system for data management. The main objective of the project was to overcome the limitations of the existing system and to create a reliable platform that simplifies data handling and improves overall performance.

#### ACKNOWLEDGEMENTS

The authors express their sincere gratitude to the project guide, faculty members, and institute for their valuable guidance and support throughout this research work.

#### REFERENCES

- [1]. W3Schools. (n.d.). *HTML, CSS, and JavaScript Tutorials*. Retrieved from <https://www.w3schools.com>  
This website was used to understand basic and advanced concepts of frontend development including HTML structure, CSS styling, and JavaScript functionality.
- [2]. Mozilla Developer Network (MDN). (n.d.). *Web Development Documentation*. Retrieved from <https://developer.mozilla.org>  
MDN provided detailed technical documentation and examples for implementing web technologies and best practices.
- [3]. Oracle. (n.d.). *MySQL Documentation*. Retrieved from <https://dev.mysql.com/doc/>  
This reference was used to understand database design, queries, and data management techniques.
- [4]. Amazon Web Services (AWS). (n.d.). *AWS Documentation*. Retrieved from <https://aws.amazon.com/documentation/>  
AWS documentation helped in understanding cloud deployment, hosting, and scalability of the system.
- [5]. Node.js Official Website. (n.d.). *Node.js Documentation*. Retrieved from <https://nodejs.org>  
This source was used for backend development concepts and server-side scripting.
- [6]. PHP Official Documentation. (n.d.). *PHP Manual*. Retrieved from <https://www.php.net/manual/en/>  
This was referred for understanding server-side scripting and backend integration

