A PROJECT REPORT ON

"Farmers Disease Diagnostic/Reporting Portal– Mobile Portal AI-Based"

SUBMITTED TO THE SAVITRIBAI PHULE PUNE UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE

COMPUTER ENGINEERING

SUBMITTED BY

STUDENT NAME	SEAT No.
KUNAL BHALERAO	B401170130
VISHAL LAHANE	B401170159
YASH ZOLEKAR	B401170195
SHAIKH SAQLIN	B401170186

UNDER THE GUIDANCE OF

Dr.M.T.Jagatap

DEPARTMENT OF COMPUTER ENGINEERING

SMES Sanghavi College of Engineering

Mhasrul, Warvandi Road, Nashik-422202, Maharashtra, India,

SAVITRIBAI PHULE PUNE UNIVERSITY

2024-25

Shree Mahavir
Education Society's

Sanghavi
College of Engineering
Nashik

This is to certify that the project report entitles

"Farmers Disease Diagnostic/Reporting Portal– Mobile Portal AI-Based"

Submitted by

STUDENT NAME	Seat No
KUNAL BHALERAO	B401170130
VISHAL LAHANE	B401170159
YASH ZOLEKAR	B401170195
SHAIKH SAQLAIN	B401170186

Is a bonafide student of this institute and the work has been carried out by him/her under the supervision of **Dr.M.T.Jagtap and it is approved for the partial fulfillment of the requirement of Savitribai Phule Pune University, for the award of the degree of Computer Engineering.**

Dr.M.T.Jagtap	**Dr. M. T. Jagtap**	**Prof. P. Biswas**
Internal Guide	**H.O.D**	**Principal**

External Examiner

ACKNOWLEDGEMENT

We would like to express our special thanks to our Project guide and source of inspiration Dr.M.T.Jagtap for his valuable guidance to make our project towards perfection. We are also extremely grateful to our HOD Dr.M.T.Jagtap and our Project Cocoordinator Dr.M.T.Jagatap for their various suggestions and all staff member of Computer department for their constant encouragement and kind help during our project for providing all facility and help for smooth progress of our project work. We would also like to express sincere gratitude towards our Principal Prof.P.Biswas for being supportive and always encouraging. In particular, we would like to thank our project guide Dr.B.S.Shirole for her kind co-operation and continuous interest in our project work. "We always pray to god to shower his blessing on us without nothing is possible to be done." Last but not least, the backbone of our success and confidence lies solely on blessing of my parent and my best friends. Thanking You.

Student Names

Kunal Bhalerao
Vishal Lahane
Yash Zolekar
Shaikh Saqlin

ABSTRACT

The rapid growth of the agricultural and livestock sectors demands efficient and intelligent solutions to combat plant and animal diseases, which significantly impact productivity, food security, and farmers' livelihoods. This project presents the "Farmers Disease Diagnostic and Reporting Portal – AI-Based Mobile Platform", a smart, user-centric system designed to assist farmers, veterinarians, and agricultural professionals in the timely diagnosis and management of plant and animal diseases. By integrating advanced machine learning algorithms, image processing, and data analytics, the system enables users to capture images or enter symptoms for accurate disease prediction. The portal provides comprehensive diagnostic support, including disease names, descriptions, suggested treatments, and precautionary measures. Additionally, it offers real-time geo-tagged reporting and nearby veterinary service suggestions using Google Maps. With an intuitive interface and support for local languages, the platform ensures accessibility and usability, especially for rural communities. The system also supports offline usage, feedback mechanisms, and data-driven decision-making, contributing to more sustainable agricultural practices. Ultimately, this mobile-based AI platform aims to enhance disease monitoring, reduce economic losses, and empower users through intelligent, responsive, and inclusive technology.

.

Keywords : symptoms matching, image classification, object detection, model integration.

Contents

List of Figures

List of Tables

Chapter 1

INTRODUCTION

1.1 Introduction

The increasing prevalence of plant and animal diseases poses a significant threat to global food security, biodiversity, and economic stability. Timely detection and accurate diagnosis are crucial for risks to human health. By creating a centralized portal for disease detection and management, we aim to address several key issues: Problem Statement Farmers and veterinarians face challenges in identifying plant and animal diseases in their early stages, leading to delayed intervention and the spread of infections. The scarcity of affordable and user-friendly diagnostic tools contributes to poor disease management. Developing a portal that includes AI-driven diagnostic tools, image-based detection, and symptoms analysis for early disease identification could enable users to address issues promptly and minimize damage. Managing these diseases effectively, minimizing losses, and ensuring the health and sustainability of agricultural and natural ecosystems. The Plant and Animal Diseases Portal aims to serve as a centralized platform for the monitoring, diagnosis, and management of diseases affecting various plant and animal species. Moreover, this portal fosters collaboration among farmers, veterinarians, researchers, and policymakers, providing them with a shared platform to share insights, report outbreaks, and access up-to-date information on disease management practices. Ultimately, this initiative aims to reduce the spread of diseases, protect agricultural resources, and support sustainable farming practices. This report outlines the portal's design, core functionalities, and the potential benefits it offers to stakeholders in agricultural and environmental sectors. The motivation behind developing an Animal and Plant Disease Portal stems from the significant impact that diseases in plants and animals have on agriculture, economy, and public health. Diseases in plants can lead to severe crop losses, af-

fecting food security and farmers' livelihoods similarly, diseases in animals, especially livestock, can impact food production, animal welfare, and even pose risks to human health. By creating a centralized portal for disease detection and management, we aim to address several key issues.

1.2 Motivation

Farmers Disease Diagnostic/Reporting Portal – Mobile Portal AI-Based is more than just a documentation task; it is an opportunity to showcase your innovation and technical expertise. This project has the potential to revolutionize agriculture by empowering farmers with an AI-driven tool for disease diagnosis and reporting. By integrating artificial intelligence with mobile technology, you are not only improving efficiency but also contributing to the well-being of farmers and the sustainability of their crops and livestock. Your report will serve as a comprehensive record of your work, demonstrating your proficiency in Flutter, AI model deployment, backend integration.

1.3 Problem Definition

Farmers and veterinarians face challenges in identifying plant and animal diseases in their early stages, leading to delayed intervention and the spread of infections. The scarcity of affordable and user-friendly diagnostic tools contributes to poor disease management. Developing a portal that includes AI-driven diagnostic tools, image based detection, and symptoms analysis for early disease identification could enable users to address issues promptly and minimize damage.

1.4 Need of Farmers Disease Diagnostic/Reporting Portal – Mobile Portal AI-Based

The need for the Farmers Disease Diagnostic/Reporting Portal – Mobile Portal AI-Based arises from the growing challenges faced by farmers in identifying and managing plant and animal diseases effectively. Traditional disease diagnosis methods often rely on expert consultation, which can be time-consuming, costly, and inaccessible for farmers in remote areas. Delayed diagnosis can lead to severe

crop losses, reduced agricultural productivity, and financial hardships. With climate change and the emergence of new plant and animal diseases, there is an urgent demand for a fast, accurate, and accessible disease detection system.

This AI-based mobile portal addresses these challenges by providing instant disease diagnosis, real-time reporting, and data-driven insights. By leveraging AI and mobile technology, farmers can simply upload images of affected crops or animals and receive immediate diagnostic results, along with recommended treatments.

Chapter 2

LITERATURE SURVEY

2.1 Literature Summery

In this chapter we will see the various studies and research conducted in order to identify the current scenarios and trends in digital Data Privacy and also the attempts of introducing mobile devices in education.

[1] **G. Shrestha, Deepsikha, M. Das and N. Dey, "Plant Disease Detection Using CNN," *IEEE Access*, vol. 2020, pp. 109-113, doi: 10.1109/ASPCON49795.2020.9276722 .**"Plant Disease Detection Using CNN", the agricultural output of the nation is impacted by pest-infected plants and crops. To find and detect diseases, farmers or experts typically monitor the plants closely. But this process is often expensive, time-consuming, and inaccurate. Finding a spot on the leaves of the infected plant is one way to identify plant diseases. This work aims to develop a disease recognition model that is backed by the classification of leaf images. We are using image processing with a convolution neural network (CNN) to identify plant illnesses. A type of artificial neural network designed especially to process pixel input and utilised in image identification is called a convolutional neural network (CNN)

[2] **M. H. Saleem, J. Potgieter and K. H. Arif, "Plant disease detection and classification by deep learning", Plants, vol. 8, no. 11, pp. 468, 2019.** "Plant Disease Detection and Classification by Deep Learning," Plants. Traditional leaf and crop disease detection is labor-intensive, requiring expert manpower and time. AI and deep learning offer faster, more accurate solutions, reducing labor and economic impact. This review analyses deep learning techniques for plant disease detection, summarizing methods, datasets, diseases,

plant types, performance metrics, and results. Challenges and future research directions are also discussed.

[3] **K.A Sharada, Najma Taj, Rida Sameer and Rukhsha Khan, "Ruzaina Zareen", Detection of Lumpy Skin Disease in cattle using IOT and Deep Learning Techniques International Journal of Advanced, vol. 3, no. 2,2023, pp. 64-78.** "Detection of Lumpy Skin Disease in Cattle Using IoT and Deep Learning Techniques", A virus belonging to the Capripoxvirus genus, which is a member of the Poxviridae family, causes Lumpy Skin Disease (LSD), a contagious illness in cattle. It is thought that the virus is transferred by biological vectors such as mosquitoes, flies, ticks, and direct contact. Numerous economic costs, including those in fertility, milk production, trade tariffs, and in certain cases, livestock animal mortality, are attributed to this virus. Additionally, the ocular secretions and pus of infected calves were found to contain the LSD virus. Because of the Web of Things and smart devices, industries including agriculture and livestock are undergoing significant change. We find it very difficult to imagine living in a world without the Internet of Things, where most objects are connected to one another.

[4] **MR Srivalli, NK Vishnu and V. Kanchana, "Teat and Udder Disease Detection on Cattle using Machine Learning", 2022 International Conference on Signal and Information Processing (IConSIP), pp. 1-5, 2022 Aug 26.** "Teat and Udder Disease Detection Cattle Using Machine Learning", in animals that produce a lot of milk, udder and teat illnesses are prevalent. As a result, producers are experiencing increased challenges with milk output and quality. Research on automatic cow disease is crucial for agriculture's large-scale population monitoring, and the disease should be automatically identified as soon as signs show up on the udder or teat. An automated system is required to identify cow teat and udder problems, determine how they affect milk production, and determine the best medications to treat them. This work aims to address the issue of identifying and preventing illnesses in various cow breeds. A number of models are examined in order to identify the best deep learning architecture.

[5] **P. Kumar, Madhu, J. Kumar and C. Sathish, "Health experts for**

pets using mobile apps", 2017 International Conference on Algorithms Methodology Models and Applications in Emerging Technologies (ICAMMAET), pp. 1-2, Feb. 2017. "Health Experts for Pets Using Mobile Apps", Our lives are greatly impacted by mobile technologies, and as these technologies improve, end users can enable new kinds of healthcare systems using rule-based expert systems. Particularly, the availability of more affordable smartphones and a more user-friendly graphical user interface (GUI) based on the Android OS opens up new opportunities for continuously monitoring the health state of pets, including dogs and cats, as well as any harmful ingestions or swallowed objects. The relevant features that are typically offered to users by the suggested applications are extensive and crucial to clinical practice and health management. Pets can be protected against disease attacks by using the pet mobile app. In an emergency, we can use this app to schedule an online appointment with a pet professional.

[6] **Ramesh, S., Hebbar, R., Niveditha M. Pooja R., Bhat N., P., Shashank N., Vinod P.V., 2018. "Plant Disease Detection Using Machine Learning", International Conference on Design Innovations for 3CsCompute Communicate Control (ICDI3C). pp. 41-45** "Crop diseases are a noteworthy risk to sustenance security, however their quick distinguishing proof stays troublesome in numerous parts of the world because of the non attendance of the important foundation. Emergence of accurate techniques in the field of leaf-based image classification has shown impressive results. This paper makes use of Random Forest in identifying between healthy and diseased leaf from the data sets created. Our proposed paper includes various phases of implementation namely dataset creation, feature extraction, training the classifier and classification. The created datasets of diseased and healthy leaves are collectively trained under Random Forest to classify the diseased and healthy images. For extracting features of an image we use Histogram of an Oriented Gradient (HOG). Overall, using machine learning to train the large data sets available publicly gives us a clear way to detect the disease present in plants in a colossal scale".

[7] **Ashar, D., Kanojia, A., Parihar, R., Kudoo, S., 2021. Livestock Disease Prediction System. IVA-Tech International Journal for Research and Innovation, 97 1, 1–3** "Livestock are farm animals who are raised to generate profit. They are used for the commodities such as meat, eggs, milk, fur, leather and wool. Livestock animals usually distribute in remote areas, with relatively poor condition of disease diagnosis. Generally, it is difficult to carry out disease diagnosis rapidly and accurately.Livestock diseases often pose a risk to public health and even affects the economy at large extent as we are quite dependent on the essential commodities we procure from the livestock. It is necessary to detect the disease outcome in the livestock to take the precautionary measures in order to avoid spread amongst them. So, there is a need for a system which can help in predicting the diseases among livestock on the basis of symptoms and suggest the precautionary measures to be taken with respect to the disease predicted. Our proposed system will predict the livestock (Cow, Sheep and Goat) disease using SVC (Support Vector Classifier) multi-class classification algorithm based on the symptoms and also provide the precautionary measures on the basis of disease predicted. There are some diseases which can prove to be fatal. So, our system will also alert the livestock owner if the predicted disease may cause a sudden death.

Chapter 3

SOFTWARE REQUIREMENTS SPECIFICATION

3.1 Assumptions and Dependencies

- The system assumes users have internet connectivity to access online services (disease prediction, Google Maps).
- It depends on trained machine learning models (CNN for plant diseases, SVM for animal diseases).
- The mobile application assumes access to the device's camera and storage.
- Google Maps API and Firebase/SQL database services are assumed to be functional.

3.2 Functional Requirements

System Feature 1 – Plant Disease Detection

- The user captures or uploads a plant image.
- The system preprocesses the image.
- The CNN model predicts the disease.
- Backend returns the disease name, description, and treatment.
- User receives the result with an option to download a report.

System Feature 2 – Animal Disease Detection

- The user selects the animal type and enters 2–7 symptoms.
- The SVM model processes symptoms and predicts the disease.
- Results with treatment suggestions are displayed and downloadable.

System Feature 3 – Contact and Feedback

- Users can submit feedback or queries via the Contact Page.
- Form data includes name, mobile number, email, and message.
- Admin receives and reviews feedback.

System Feature 4 – Veterinary Services

- Users can access a list of nearby veterinary services via map integration.
- Google Maps API fetches nearest hospitals based on current location.

System Feature 5 – User Authentication

- New users can register using username, email, and password.
- Registered users can login and logout securely.

3.3 External Interface Requirements

3.3.1 User Interfaces

Mobile UI with Home, Scan, Animal, Services, and Contact pages.

3.3.2 Hardware Interfaces

- Mobile device camera for capturing images.
- Storage access for image upload/download.

3.3.3 Software Interfaces

- TensorFlow or PyTorch for CNN model.
- Scikit-learn for SVM model.
- Google Maps API.
- Backend API (Flask/Express).
- MySQL.

3.3.4 Communication Interfaces

RESTful API calls between frontend (Flutter) and backend services (Flask/Express).

3.4 Nonfunctional Requirements

3.4.1 Performance Requirements

- Image prediction response time ¡ 3 seconds.
- Symptom-based prediction response time ¡ 2 seconds.
- System uptime: 99% availability.

3.4.2 Safety Requirements

Exception handling for invalid inputs, model errors, and no internet.

3.4.3 Security Requirements

- Password encryption.
- Secure API endpoints.

3.4.4 Software Quality Attributes

- **Usability:** Clean, responsive UI.
- **Maintainability:** Modular code structure.

- **Portability:** Compatible with Android and iOS.
- **Scalability:** Cloud-ready architecture.

3.5 System Requirements

Firstly, if privacy policies and relevant user preferences are not adequately embedded in PDC solutions (e.g., if the options provided to the users are not flexible and/or editable enough), the users might give away more personal data than they would otherwise; A final aspect of PDCs relates to the very relationship between the individuals and their personal data. In order to assume control, users may need to be actively in- volved in the management of their data. Therefore, we must account for the possibility that too much granularity in information management (for example, in access control settings) may, in the end, overburden the users and alienate them from carrying out adequate management of their data. Therefore, striking a balance between granularity and usability may be of utmost importance in that respect.

3.5.1 Database Requirements

Figure 3.1: MySQL Database

MySQL is a fast, easy-to-use RDBMS being used for many small and big businesses. MySQL is developed, marketed and supported by MySQL AB, which is a Swedish company. MySQL is becoming so popular because of many good reasons

- MySQL is released under an open-source license. So you have nothing to pay to use it.
- MySQL is a very powerful program in its own right. It handles a large subset of the functionality of the most expensive and powerful database packages.
- MySQL uses a standard form of the well-known SQL data language.
- MySQL works on many operating systems and with many languages including PHP, PERL, C, C++, JAVA, etc.
- MySQL works very quickly and works well even with large data sets.
- MySQL is very friendly to PHP, the most appreciated language for web development.
- MySQL supports large databases, up to 50 million rows or more in a table. The default file size limit for a table is 4GB, but you can increase this (if your operating system can handle it) to a theoretical limit of 8 million terabytes (TB).
- MySQL is customizable. The open-source GPL license allows programmers to modify the MySQL software to fit their own specific environments.

3.5.2 Software Requirements(Platform Choice)

- **Python**

Figure 3.2: Python software programming language

Python is a multi-paradigm programming language. Object-oriented programming and structured programming are fully supported, and many of their features support functional programming and aspect-oriented programming (including metaprogramming and metaobjects).
Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse.

3.5.3 Technologies and Tools Used

– **Dart**: Dart is a client-optimized programming language developed by Google, specifically designed for fast and efficient UI development. It is the backbone of Flutter applications, enabling reactive and declarative

programming styles. Dart is known for its high performance and ease of use in building mobile, web, and desktop apps.

- **Flutter**: Flutter is an open-source UI software development kit (SDK) developed by Google. It allows developers to build natively compiled applications for mobile, web, and desktop from a single codebase. With its built-in widgets and hot reload feature, Flutter speeds up development and delivers a rich, customizable UI experience.
- **Node.js**: Node.js is a runtime environment that allows developers to execute JavaScript code server-side. It is used in this project to handle backend processes, APIs, and database operations efficiently. Its asynchronous, non-blocking architecture makes it ideal for real-time applications and scalable systems.
- **Android Studio**: Android Studio is the official Integrated Development Environment (IDE) for Android development, provided by Google. It includes powerful features such as a visual layout editor, emulator, and tools for profiling and debugging apps. Android Studio is essential for testing Flutter apps on various Android devices and configurations.
- **Visual Studio Code (VS Code)**: Visual Studio Code is a lightweight yet powerful code editor developed by Microsoft. It supports a wide range of programming languages and is highly extensible through plugins. For this project, it is primarily used to write Dart and Node.js code, with extensions for Flutter, Git integration, and real-time debugging.

list of software requirement are as follow:

1. Operating System : Windows 9/10/11
2. Programming Language : Python
3. Software Version : Python 4.5
4. Tools : Anaconda/pycharm
5. Front End : Python

3.5.4 Hardware Requirements

1. Processor - Pentium IV/Intel I3 core
2. Speed - 1.1 GHZ
3. RAM - 512 MB(min)
4. Hard disk - 20 GB
5. Keyboard - Standard Keyboard
6. Mouse - Two Or Three Button Mouse
7. Monitor - LED Monitor
8. SD card

Chapter 4

SYSTEM DESIGN

4.1 System Architecture

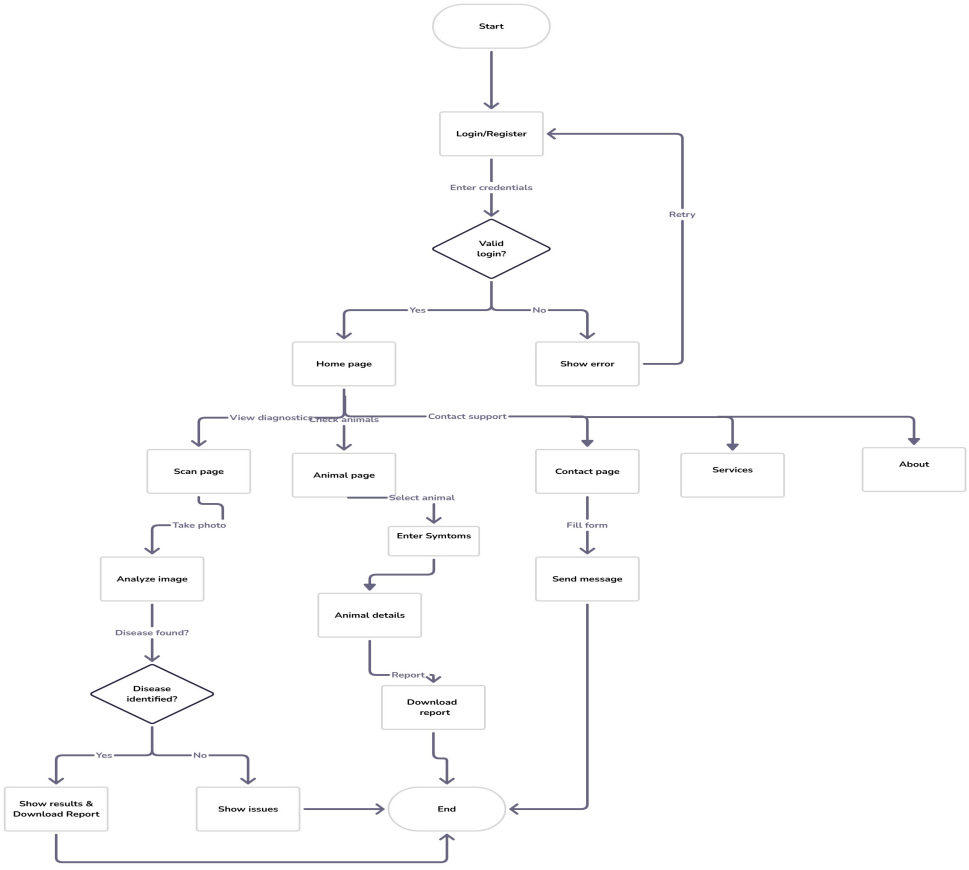


Figure 4.1: System Architecture Diagram

The flowchart begins with the user opening the app and logging in or registering. Upon successful login, the user reaches the home page, from where they can access different features. They can either scan a plant image to detect diseases using AI, or go to the animal page to enter symptoms manually. If a disease is identified, a report is generated and can be downloaded. Additionally, users can contact support, explore services, or learn more through the about page.

4.2 Mathematical Model

1. Mathematical Model Representation

Let the system be represented as a 7-tuple:

$$S = \{I, O, P, F, D, R, S'\}$$

Where:

- I = Set of inputs
- O = Set of outputs
- P = Processes/Functions performed by the system
- F = Set of functions or transformations
- D = Dataset used for training and prediction
- R = Set of rules/conditions
- S' = Final state/output of the system

2. Detailed Description

Inputs (I):

$I = \{\text{Image of plant/animal (Img)}, \text{Symptoms (Sym)}, \text{Location (Loc)}, \text{User credentials (Cred)}\}$

Outputs (O):

$O = \{\text{Predicted disease (Pd)}, \text{Disease description (Dd)}, \text{Treatment suggestions (Ts)}, \text{Nearby vet inf}$

Processes (P):

$$P = \{ \begin{array}{ll} P1: & \text{Image preprocessing} \\ P2: & \text{Symptom analysis} \\ P3: & \text{Disease prediction using AI models} \\ P4: & \text{Report generation} \\ P5: & \text{Contact/help support} \\ P6: & \text{Map integration} \end{array} \}$$

Functions/Transformations (F):

$$F = \{ \begin{array}{ll} f1: & f(\text{Img}) \rightarrow \text{Preprocessed Image} \\ f2: & f(\text{Preprocessed Img}) \rightarrow \text{Feature vector} \\ f3: & f(\text{Feature vector + Sym}) \rightarrow \text{Disease prediction (CNN/SVM)} \\ f4: & f(\text{Prediction}) \rightarrow \text{Description + Treatment + Precaution} \\ f5: & f(\text{Loc}) \rightarrow \text{Nearby hospitals (Google Maps API)} \end{array}$$

Dataset (D):

$$D = \{\text{Labeled plant and animal disease images, text-based symptom records}\}$$

Rules/Conditions (R):

$$R = \{ \begin{array}{ll} r1: & \text{User must be authenticated} \\ r2: & \text{Minimum 2 symptoms required for prediction} \\ r3: & \text{Valid image format: .jpg/.png} \\ r4: & \text{Internet connectivity OR Offline cache available} \end{array} \}$$

Final State (S'):

$$S' = \{\text{Disease identified \& result shown}\}$$

4.3 Data Flow Diagram

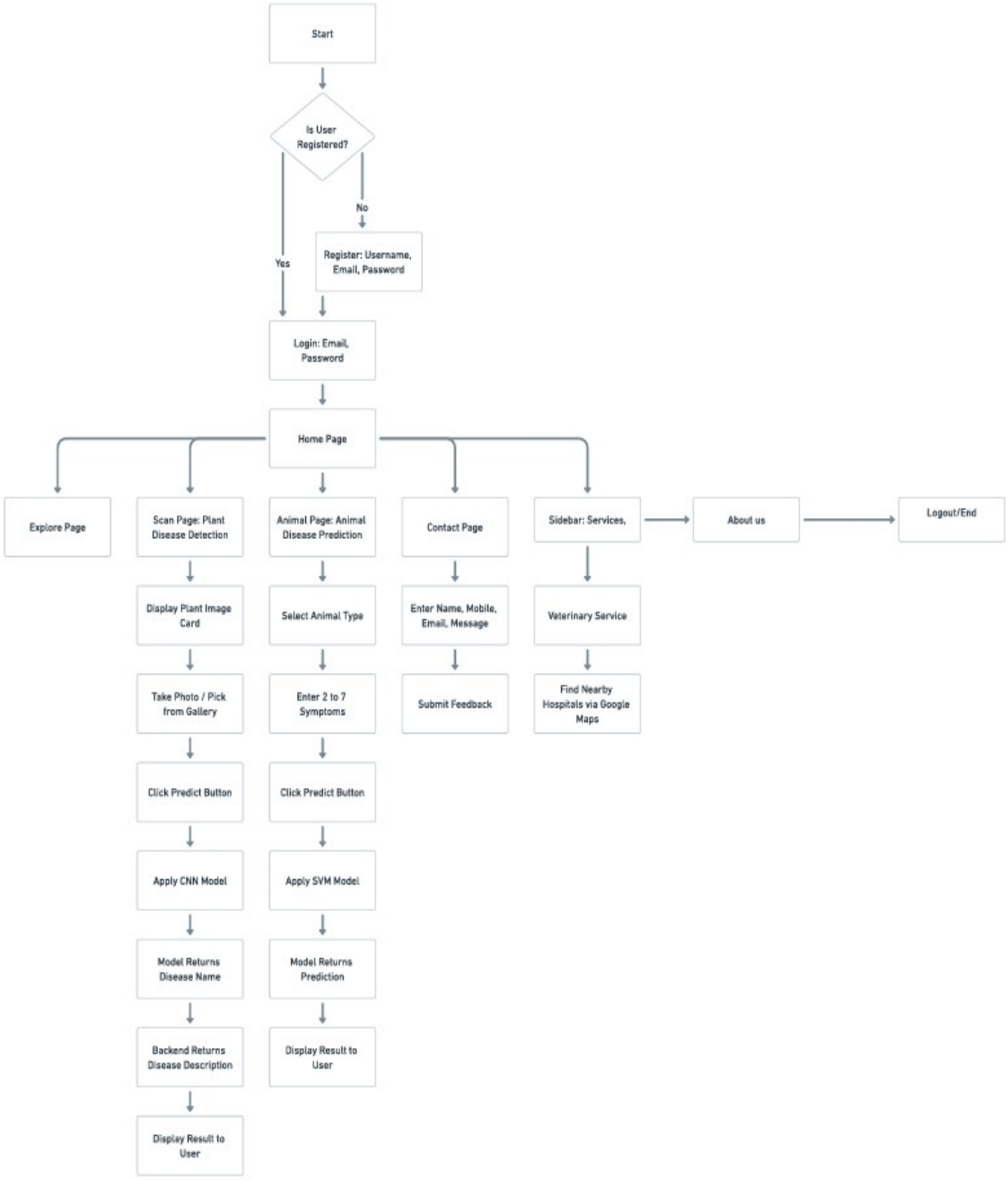


Figure 4.2: Data Flow Diagram

The figure illustrates the complete user flow of a mobile-based AI portal for plant and animal disease detection, from registration and login to disease prediction, feedback submission, and service access.

Chapter 5

PROJECT PLAN

The Farmers Disease Diagnostic/Reporting Portal is an AI-powered mobile application aimed at helping farmers detect and report plant and animal diseases. The application leverages machine learning, cloud computing, and mobile technology to provide real-time diagnostic insights, recommendations, and disease reporting mechanisms.

5.1 Project Estimate

5.1.1 Reconciled Estimates

The project estimate is based on hours allocated to each task across development, testing, AI integration, and project management, along with associated hourly costs.

- **Total Estimated Hours:** 405 hours
- **Average Hourly Rate:** 0/hour
- **Total Cost Estimate: ,100**

Breakdown by Major Components:

- AI Model Development & Integration: ,000
- Mobile App Development (Flutter): ,000
- Backend & Database: ,200
- UI/UX Design: 00
- Testing & QA: 00
- Project Management: ,000

5.1.2 Project Resources

Resource Role	Responsibility	Duration	Hours/Week
Project Manager	Planning, coordination, stakeholder updates	10 weeks	10 hrs
ML Engineer	Model training, Flask API integration	2 weeks	25 hrs
Flutter Developer	Mobile UI development, API integration	5 weeks	30 hrs
Backend Developer	Node.js APIs, MySQL integration	3 weeks	20 hrs
QA Tester	Functional, integration, UAT testing	2 weeks	15 hrs
UI/UX Designer	Wireframes, UI prototyping	1 week	15 hrs

Table 5.1: Project Resources

5.2 Risk Management

5.2.1 Risk Identification

- R1: Poor internet connectivity in rural deployment areas
- R2: Low image quality from farmer mobile devices
- R3: Inadequate dataset for disease detection
- R4: Delay in integration between frontend and backend
- R5: Data privacy and security issues
- R6: Multilingual support inconsistencies

5.2.2 Risk Analysis

Risk ID	Likelihood	Impact	Severity	Description
R1	High	High	Critical	Farmers may not access key features due to poor internet
R2	Medium	High	Major	Poor image quality may lead to incorrect predictions
R3	Medium	High	Major	Limited datasets may reduce model accuracy
R4	Medium	Medium	Moderate	Backend delays may block complete app functionality
R5	Low	High	Major	User data may be compromised without security measures
R6	Medium	Medium	Moderate	Language barriers may confuse users

Table 5.2: Risk Analysis

5.2.3 Overview of Risk Mitigation, Monitoring, Management

Risk ID	Mitigation Strategy	Monitoring & Management
R1	Implement offline mode and retry queue	Log upload failures and notify users upon reconnection
R2	Add image quality checks with guidance	Review failed predictions and improve user prompts
R3	Expand dataset with diverse samples	Retrain model quarterly with updated datasets
R4	Parallel development with mock APIs	Weekly integration sync meetings
R5	Encrypt data, secure APIs, use HTTPS	Conduct regular security audits

5.3 Project Schedule

5.3.1 Project Task Set

The following task set defines the primary activities and deliverables throughout the project lifecycle:

- **Requirement Analysis:** Gather functional and non-functional requirements.
- **UI/UX Design:** Develop wireframes, mockups, and app flow.
- **AI Model Development:** Train, evaluate, and deploy the plant disease detection model.
- **Backend Development:** Create APIs, database schema, and server-side logic.
- **Mobile App Development:** Implement frontend using Flutter and integrate APIs.
- **Testing & QA:** Perform unit, integration, and user acceptance testing.
- **Deployment:** Launch the application and monitor the environment.

5.3.2 Task Network

The task dependencies are illustrated below:

- Requirement Analysis → UI/UX Design
- UI/UX Design → Mobile App Development
- AI Model Development → Backend API Integration
- Backend Development → API Integration
- API Integration + Frontend → Testing & QA
- Testing → Deployment

This structure ensures minimal bottlenecks and allows parallel execution where feasible.

5.3.3 Timeline Chart

Task	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10
Requirement Analysis	X									
UI/UX Design		X								
AI Model Development		X	X							
Backend Development			X	X						
Flutter App Development				X	X	X				
Integration					X	X				
Testing & QA						X	X			
Deployment								X	X	

5.4 Team Organization

5.4.1 Team Structure

The team is organized in a collaborative, cross-functional structure as follows:

- **Project Manager:** Oversees planning, execution, and delivery.
- **ML Engineer:** Develops and integrates the plant disease detection model.
- **Flutter Developer:** Builds mobile application and integrates APIs.
- **Backend Developer:** Implements database, API, and server-side logic.
- **UI/UX Designer:** Designs intuitive and accessible app interfaces.
- **QA Tester:** Validates functionality, performance, and reliability.

5.4.2 Management Reporting and Communication

The project uses agile-based communication with structured weekly sprints. Key reporting and communication channels include:

- **Daily Stand-ups:** Quick updates on progress and blockers.
- **Weekly Review Meetings:** Review deliverables and timelines.

- **Sprint Retrospectives:** Analyze progress and plan improvements.
- **Collaboration Tools:** Slack (chat), GitHub (code), Trello (tasks), Google Drive (docs).
- **Status Reports:** Sent every Friday by Project guide to stakeholders.

Chapter 6

PROJECT IMPLEMENTATION

6.1 Overview of Project Modules

The AI-based Farmers Disease Diagnostic and Reporting Portal consists of several modules:

- **User Module:** Enables farmers to register, log in, and interact with the mobile app.
- **Image Upload & Capture Module:** Allows users to capture or upload plant leaf images.
- **AI-Based Disease Detection Module:** Utilizes machine learning models (SVM and CNN) to classify and detect diseases.
- **Disease Information Module:** Provides disease details such as symptoms, treatments, and precautions from the backend database.
- **Report Generation Module:** Creates a diagnostic summary report for users.
- **Admin Panel Module:** Admin interface for managing datasets and monitoring the system.

6.2 Tools and Technologies Used

Tool / Technology	Purpose
Flutter (Dart)	Mobile application development
Flask (Python)	Backend API for AI inference
TensorFlow / Keras	Deep learning framework used for CNN model
Scikit-learn	Library used for SVM implementation
MySQL	Relational database for storing disease information
Firebase	User authentication and real-time database (optional)
Visual Studio Code	Code editor used during development
Postman	Testing and debugging APIs
Git & GitHub	Version control and collaboration

Table 6.1: Tools and Technologies Used

6.3 Algorithm Details

6.3.1 SVM Algorithm

Support Vector Machine (SVM) is a powerful machine learning algorithm used for:

- **Classification** (e.g., "Is the plant diseased or not?")
- **Regression** (less common for SVM, used for predicting values)

In the context of disease detection, it helps classify data into categories like:

- Healthy
- Diseased – Bacterial Spot
- Diseased – Early Blight
- And so on...

How SVM Works (Step-by-Step)

Imagine we want to separate data points into two categories: Healthy vs. Diseased. Each data point has features (like leaf color, temperature, size of spots).

1. Plot The Data

SVM plots these points in a space (2D, 3D, or even higher dimensions).

2.Find the Best Boundary

SVM draws a line (in 2D) or a plane (in 3D) called a *hyperplane* that separates the categories.
But not just any line it finds the best one that has the **maximum margin** between both classes.

3.Support Vectors

The data points that are **closest to the hyperplane** are called *support vectors*.
They are the most critical points that help define the boundary.

4.Classify New Data

When new data comes in (like a new leaf image), SVM checks which side of the hyperplane it falls on and **predicts the class**.

6.3.2 CNN Algorithm

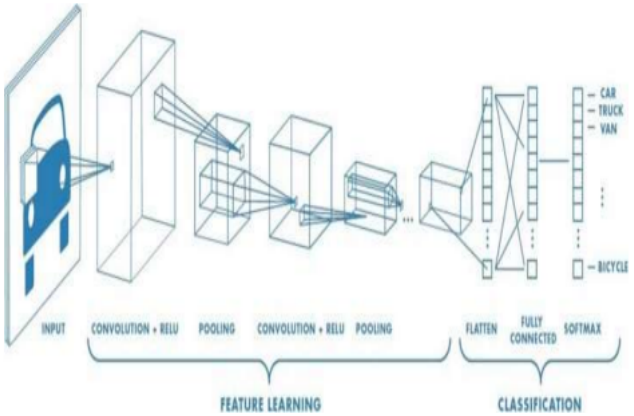


Figure 6.1: CNN Algorithm

purpose

The primary purpose of the **Convolutional Neural Network (CNN)** model in the context of disease detection is to classify images of plants or animals into categories such as *"Healthy"* or *"Diseased"*. This is achieved by training the model on labeled images, where each image represents a specific condition. Once trained, the CNN can automatically identify visual symptoms like discoloration, spots, or lesions—patterns commonly associated with diseases. This approach significantly enhances early diagnosis, helping farmers take timely action to prevent disease spread and minimize crop or livestock losses.

Key Components of the CNN Model

A CNN model is designed to work specifically with image data. It is inspired by the way human vision processes images, allowing the model to detect important features and patterns. The CNN architecture includes several essential layers:

1. Convolutional Layers ()

These layers apply filters (kernels) to the input image, scanning it piece by piece. Each filter is responsible for detecting a specific feature, such as edges, textures, or color variations. The result is a **feature map** that highlights where these patterns appear in the image. Multiple convolutional layers help the model learn from simple to complex patterns (e.g., from spots to whole disease regions).

2. Pooling Layers ()

Pooling layers reduce the size of the feature maps created by convolutional layers. The most common type is **max pooling**, which takes the highest value in a small window (usually 2x2). This helps:

- Reduce computational complexity,
- Extract dominant features,
- Prevent overfitting by adding a bit of translational invariance.

Pooling acts like compressing the image while keeping the most important information intact.

3. Flatten Layer ()

After multiple convolution and pooling operations, the resulting data is still in matrix (2D) form. The **flatten layer** reshapes this data into a 1D vector. This is necessary to pass it into the **dense (fully connected) layers** that follow.

4. Dense (Fully Connected) Layers ()

Dense layers work like a traditional neural network. Each neuron is connected to every neuron in the previous layer. These layers learn complex combinations of the features detected earlier and help in making the final classification decision. Activation functions like **ReLU** are commonly used to introduce non-linearity and improve learning.

5. Output Layer ()

The output layer produces the final prediction:

- For **binary classification** (e.g., Healthy vs Diseased), a **sigmoid** activation function is used, giving a value between 0 and 1.
- For **multi-class classification** (e.g., different disease types), a **softmax** activation function is used, producing probabilities for each class.

The class with the highest probability becomes the predicted label for the input image.

Why CNN is Effective for Disease Detection ()

- **Automation:** Reduces the need for manual inspection by experts.
- **Accuracy:** Detects subtle patterns that may not be visible to the human eye.
- **Scalability:** Can be used with large datasets and deployed in real-time applications (like mobile apps).
- **Adaptability:** Works with both plant and animal diseases if trained with diverse and labeled data.

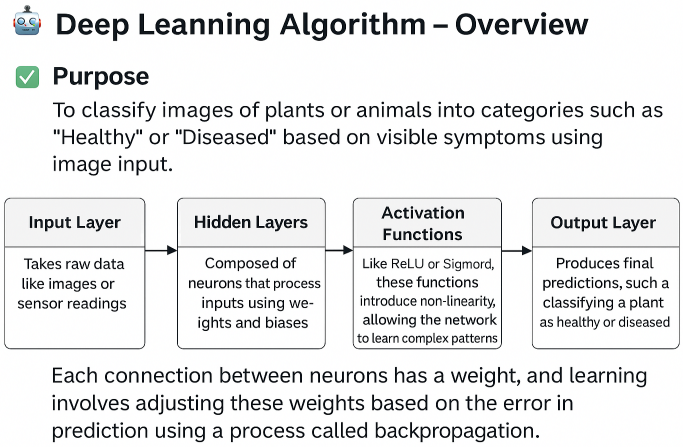


Figure 6.2: Deep Learning Algorithm

6.3.3 Deep Learning Algorithm – Overview

What is Deep Learning?

Deep Learning is a subset of **machine learning** that uses multi-layered artificial neural networks to model and understand complex patterns in data. These networks are called "deep" because they contain multiple hidden layers between the input and output layers. Unlike traditional algorithms, deep learning automatically extracts features from raw data, making it highly effective for tasks like *image recognition*, *speech processing*, and *natural language understanding*.

How Deep Learning Works

At its core, a deep learning algorithm consists of:

- **Input Layer:** Takes raw data like images or sensor readings.
- **Hidden Layers:** Composed of neurons that process inputs using weights and biases.
- **Activation Functions:** Like ReLU or Sigmoid, these functions introduce non-linearity, allowing the network to learn complex patterns.
- **Output Layer:** Produces final predictions, such as classifying a plant as healthy or diseased.

Each connection between neurons has a **weight**, and learning involves adjusting these weights based on the error in prediction using a process called backpropagation.

Common Deep Learning Architectures

- **CNN (Convolutional Neural Network):** Best for image data like plant and animal disease detection.
- **RNN (Recurrent Neural Network):** Great for sequence data like time-series or text.
- **DNN (Deep Neural Network):** Fully connected layers for general classification or regression.
- **GANs (Generative Adversarial Networks):** Used for data generation and augmentation.

Training Process

1. Feed input data (like plant images) into the network.
2. Compute output and compare it with the actual label.
3. Calculate the error (loss) using a loss function.
4. Use backpropagation to adjust weights.
5. Repeat for multiple epochs until accuracy improves.

Advantages in Disease Detection

- **High Accuracy:** Learns from complex patterns not visible to the human eye.
- **Real-Time Processing:** Can be integrated into mobile apps and devices.
- **Automation:** No need for manual feature extraction.
- **Scalability:** Performs well with large datasets.

Chapter 7

SOFTWARE TESTING

7.1 Type of Testing

- **Unit Testing**: Testing individual components like image upload, disease detection.
- **Integration Testing**: Verifying interaction between mobile frontend and backend APIs.
- **Functional Testing**: Ensuring app features work as expected from user perspective.
- **System Testing**: Testing the entire system on real or virtual devices.
- **Performance Testing**: Checking API response time, model latency, and mobile UI responsiveness.
- **Usability Testing**: Evaluating user interface and experience, ensuring accessibility and ease of use.

7.2 AI Inference Test Cases

Test Case ID	Description	Input	Expected Output
AI-01	Detect disease from healthy leaf image	Healthy tomato leaf	"Healthy"
AI-02	Detect disease from infected leaf	Tomato leaf with blight	"Tomato Early Blight"
AI-03	Detect multiple crops	Pepper leaf	"Pepper Bell Bacterial Spot"
AI-04	Handle poor image quality	Blurry or low-res image	"Error: Unclear Image" or Retry prompt
AI-05	Handle unsupported crops	Corn leaf	"Crop not supported"

Table 7.1: AI Interface Test Case

7.3 Mobile Portal Functional Test Cases

Test Case ID	Description	Action	Expected Result
M-01	Upload and scan image	Select image from camera or gallery	Inference and disease info shown
M-02	Offline image scan	Try scanning without internet	Prompt: "No internet connection"
M-03	View detailed report	Tap on result	Show disease, description, treatment, medicine
M-04	Submit report to authority	Tap "Report Issue"	Report successfully sent
M-05	Switch language	Select a different language	UI switches to selected language

Table 7.2: Mobile Portal Functional Test Cases

7.4 Sample Test Results

Test Case ID	Result	Notes
AI-01	Pass	Correctly detected as “Healthy”
AI-02	Pass	Correctly identified as “Tomato Early Blight”
AI-04	Warning	Prompted retry due to unclear image
M-03	Pass	Past reports displayed successfully with timestamps
M-06	Fail	Language not updated in report screen

Table 7.3: Sample Test Results

Chapter 8

RESULTS AND EVALUATION

8.1 Result Screnshot

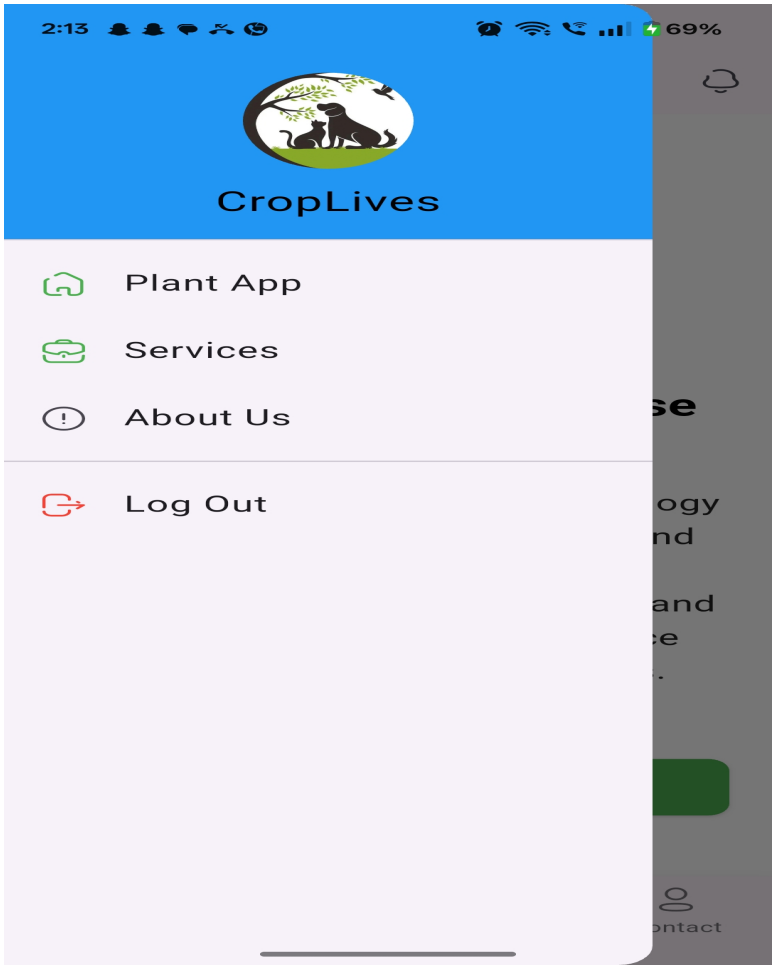


Figure 8.1: Home Page

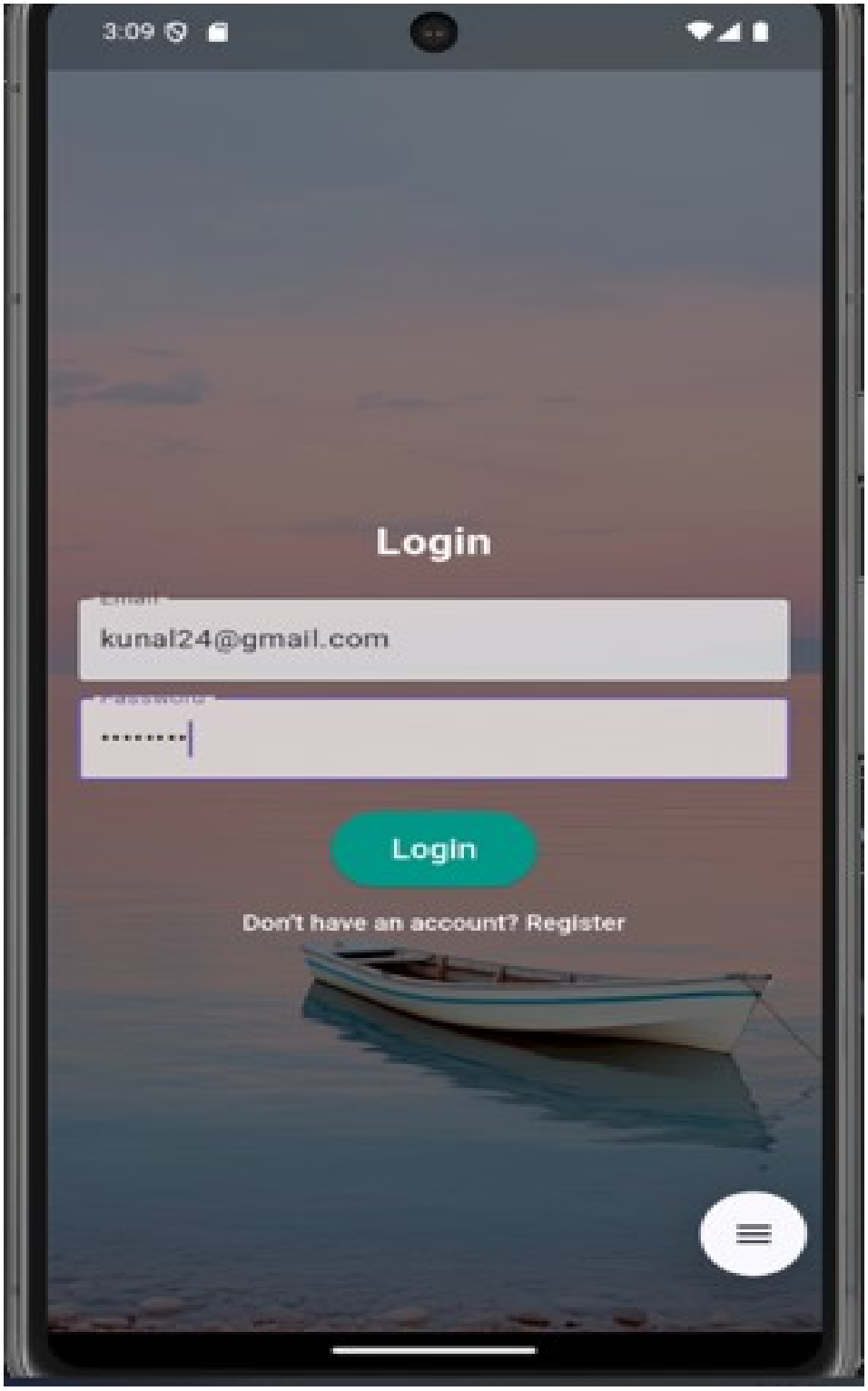


Figure 8.2: login Page

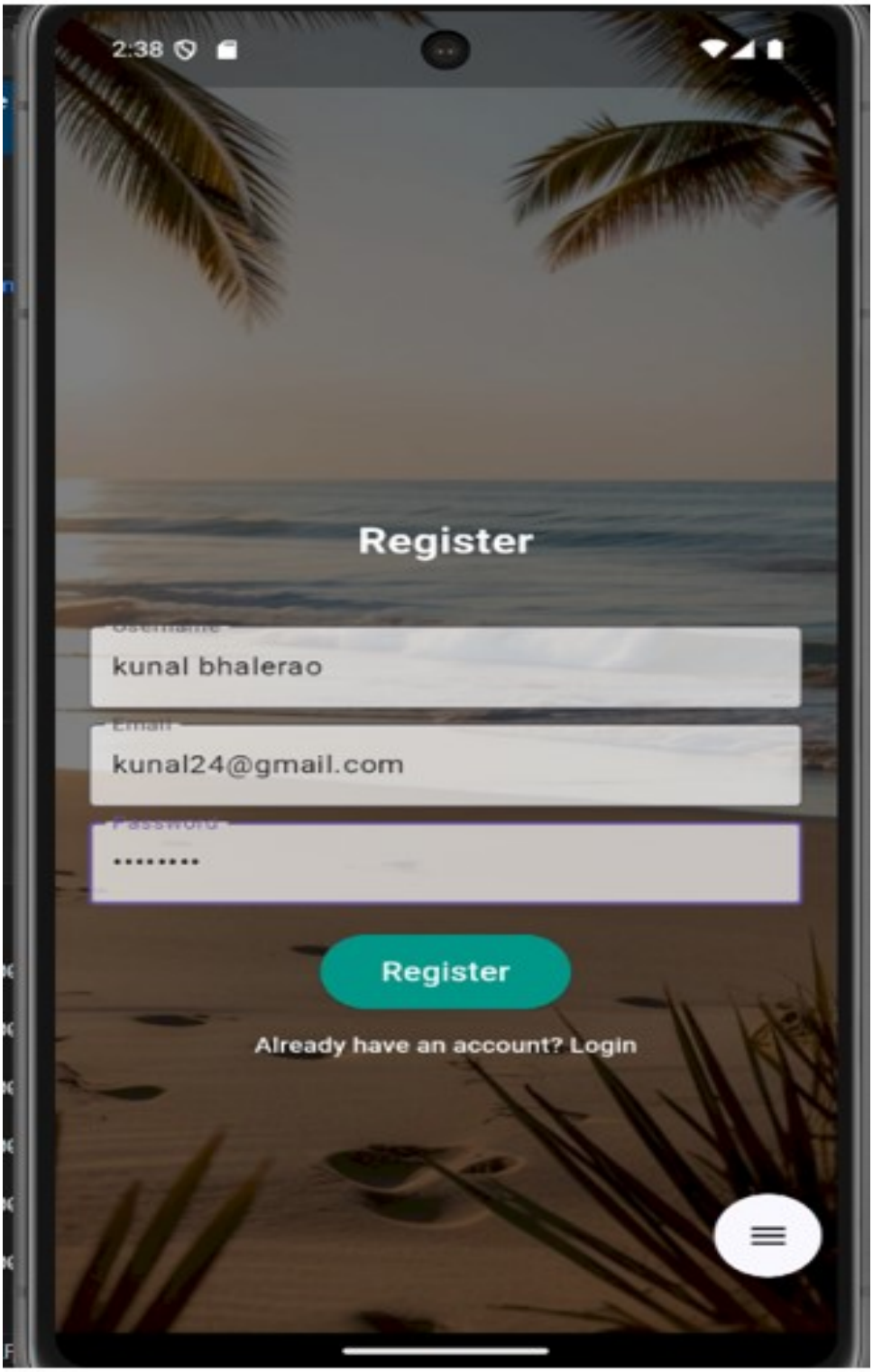


Figure 8.3: register Page

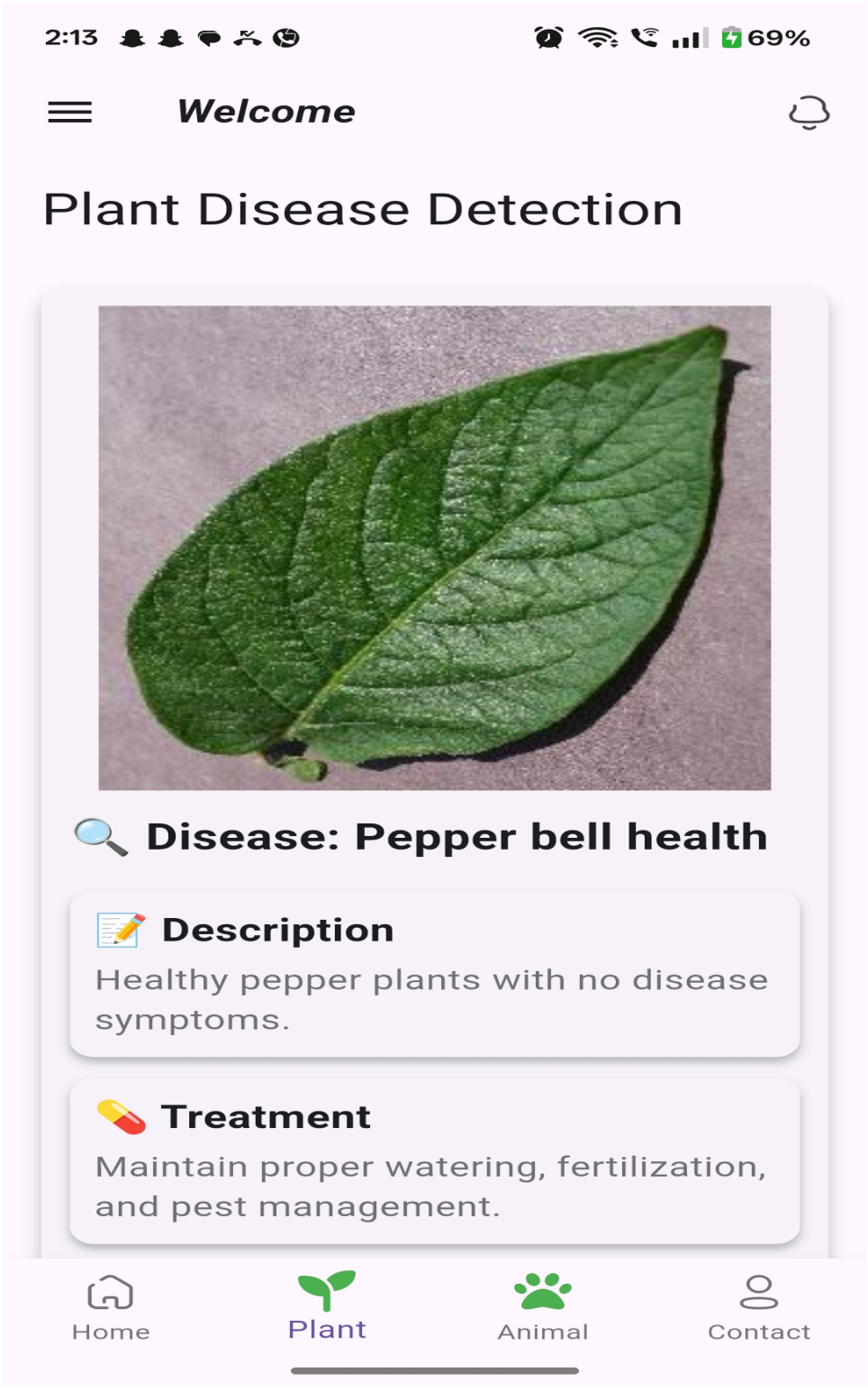


Figure 8.4: plant Disease Predictor Application Page

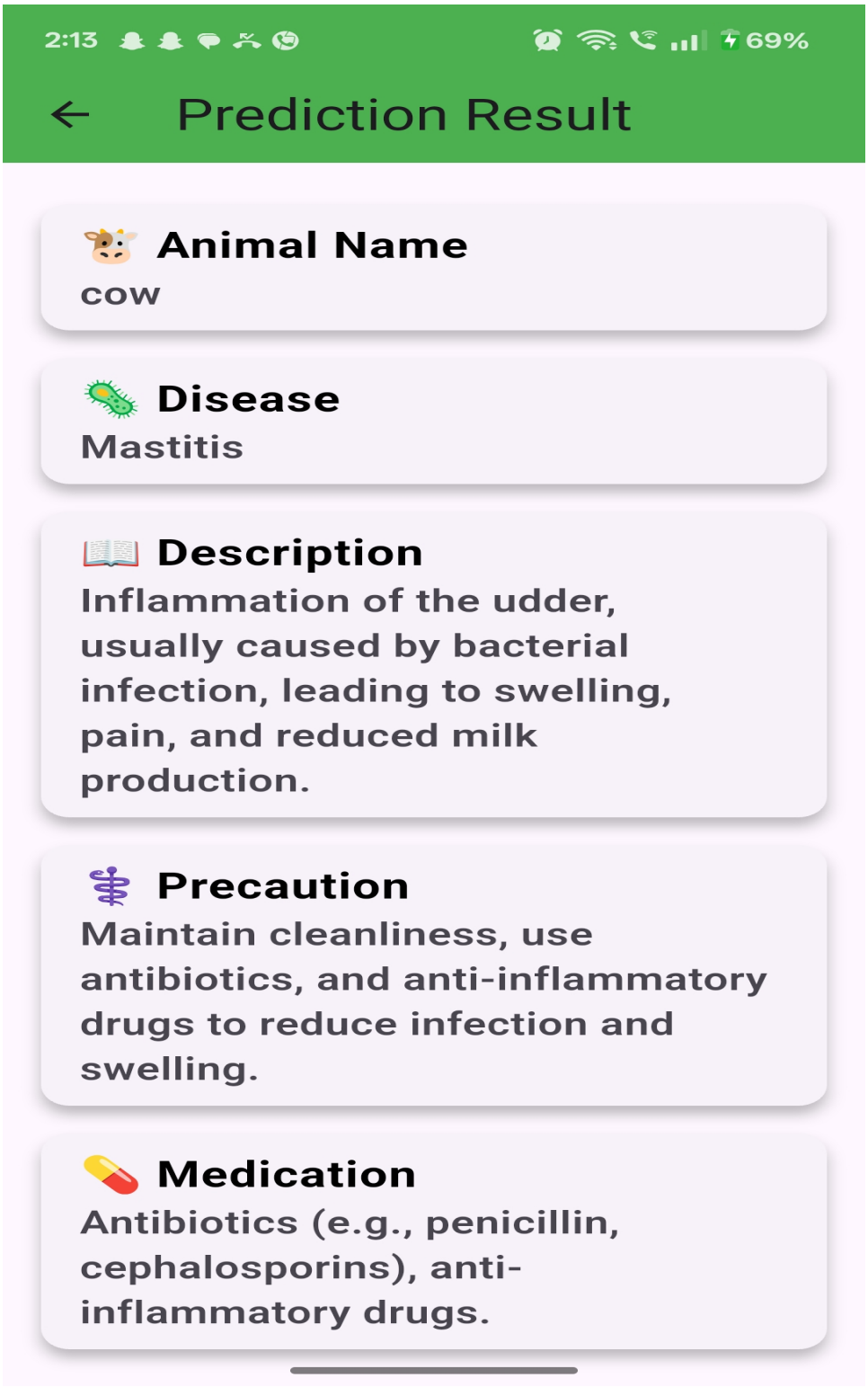


Figure 8.5: Animal Disease Predictor Application page

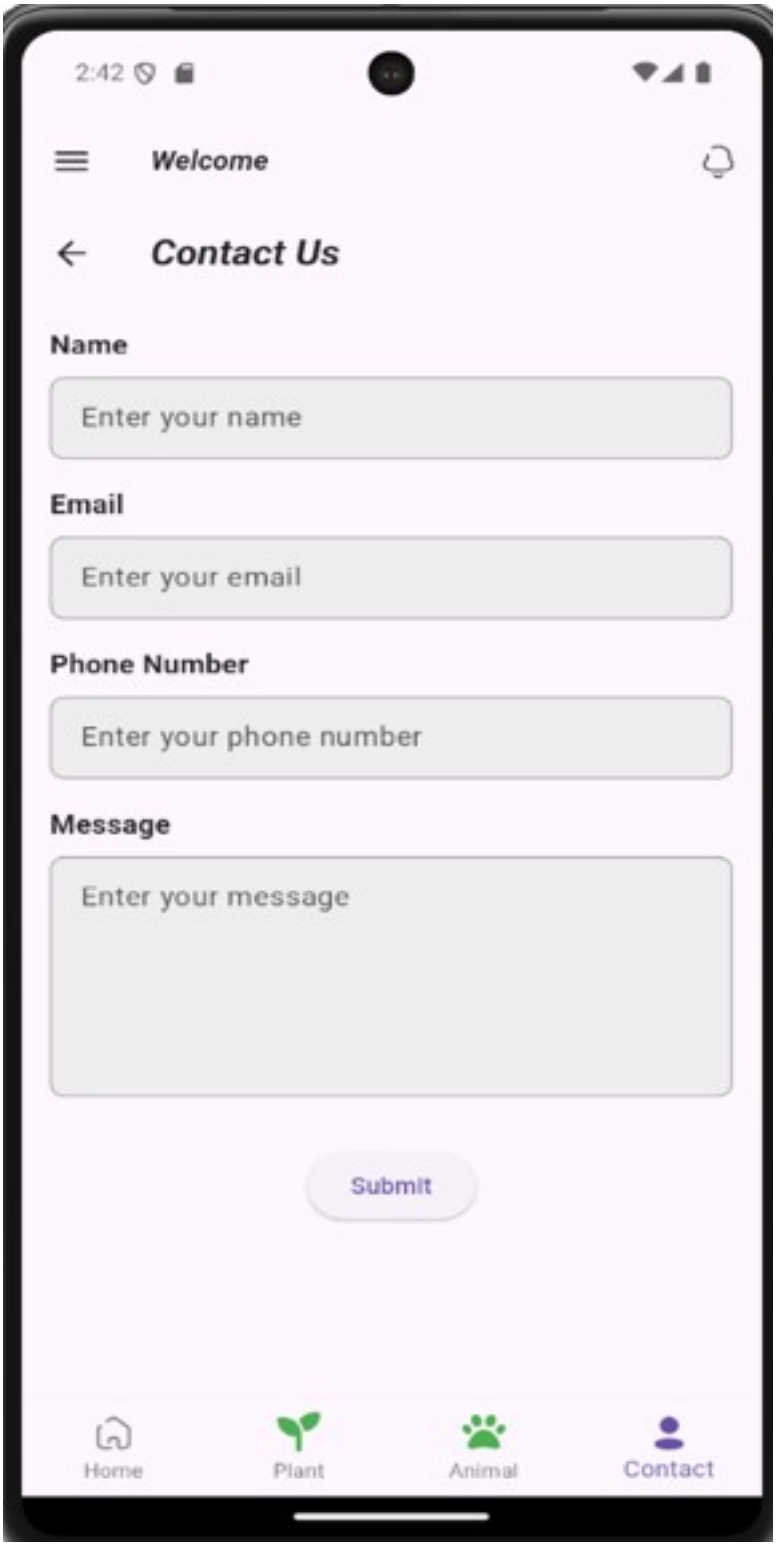


Figure 8.6: Contact Us page

Chapter 9

CONCLUSION

9.1 Conclusion

The Farmers Disease Diagnostic/Reporting Portal mobile app offers a user-friendly and accessible platform for identifying and reporting plant and animal diseases. By integrating deep learning—specifically Convolutional Neural Networks (CNNs)—the app provides accurate disease predictions from images, empowering farmers with timely information to protect their crops and livestock.

This solution enhances agricultural productivity by reducing the delay between disease detection and treatment. It also fosters communication between farmers and experts through support and contact features. The app's structured navigation ensures ease of use, even for users with limited technical experience.

Overall, the project demonstrates how modern technology can be leveraged to solve traditional farming challenges, improving food security and farmer livelihoods through innovation and accessibility. .

9.2 Advantages

- **Early Disease Detection**
  The app helps farmers identify symptoms of plant or animal diseases at an early stage using image analysis, allowing faster response and treatment.

- **High Accuracy with CNN**
  By leveraging Convolutional Neural Networks, the app provides accurate classification based on visual symptoms, improving diagnosis reliability compared

to manual observation.

- **User-Friendly Interface**
  Simple and intuitive navigation makes it accessible to users with minimal technical background, ensuring widespread usability.

- **Real-Time Results**
  Farmers receive instant feedback on uploaded images, making it practical for real-time decision-making in the field.

- **Support and Connectivity**
  Integrated contact and support options enable farmers to reach out to agricultural experts, fostering better communication and guidance.

- **Cost-Effective Solution**
  Reduces the need for physical consultations and lab testing, making disease detection more affordable and accessible.

- **Scalability**
  The system can be extended to support multiple crops, animals, and disease types, adapting to different farming needs and regions.

- **Educational Resource**
  The app also serves as an informational hub, offering insights about various diseases, symptoms, and preventive measures.

9.3 Limitations

- **Dependence on Image Quality**
  The accuracy of disease detection heavily depends on the clarity, lighting, and angle of the image uploaded by the user. Poor image quality may lead to incorrect predictions.

- **Limited Dataset**
  The model's performance is restricted by the size and diversity of the training dataset. Rare diseases or conditions with fewer examples may not be classified accurately.

- **Connectivity Requirements**
  The app may require an internet connection for real-time predictions or accessing updated services, which could be a challenge in remote farming areas.

- **Language and Literacy Barriers**
  If the app supports only a single language or lacks voice assistance, it may be difficult for some farmers to use effectively.

- **Hardware Constraints**
  Older or low-end mobile devices may struggle to process image data efficiently or run deep learning models, leading to performance issues.

- **Generalization Issues**
  The model might not generalize well to new disease strains or environmental conditions not represented in the training data.

- **No Physical Testing**
  Visual diagnosis can miss internal symptoms or diseases that do not show clear external signs, limiting the app's scope compared to lab-based methods.

- **Privacy Concerns**
  Storing images and user data may raise privacy and data security issues if not handled properly.

REFERENCES

1. G. Shrestha, Deepsikha, M. Das and N. Dey, "Plant Disease Detection Using CNN," *IEEE Access*, vol. 2020, pp. 109-113, doi: 2020.9276722 .

2. K.A Sharada, Najma Taj, Rida Sameer and Rukhsha Khan, "Ruzaina Zareen", Detection of Lumpy Skin Disease in cattle using IOT and Deep Learning Techniques International Journal of Advanced, vol. 3, no. 2,2023, pp. 64-78.

3. MR Srivalli, NK Vishnu and V. Kanchana, "Teat and Udder Disease Detection on Cattle using Machine Learning", 2022 International Conference on Signal and Information Processing (IConSIP), pp. 1-5, 2022 Aug 26.

4. K. Ferentinos, "Deep learning models for plant disease detection and diagnosis", Computers and Electronics in Agriculture, vol. 145, pp. 311-318, 2018.

5. M. H. Saleem, J. Potgieter and K. H. Arif, "Plant disease detection and classification by deep learning", Plants, vol. 8, no. 11, pp. 468, 2019.

6. P. Kumar, Madhu, J. Kumar and C. Sathish, "Health experts for pets using mobile apps", 2017 International Conference on Algorithms Methodology Models and Applications in Emerging Technologies (ICAMMAET), pp. 1-2, Feb. 2017.

7. S. Neethirajan, "Recent advances in wearable sensors for animal health management", Sens. Bio-Sens. Res., vol. 12, pp. 15-29, Feb. 2017.

8. Liu, J., Wang, X., 2021. "Plant diseases and pests detection based on deep learning: a review", Plant Methods 17:22. pp. 1 18.

9. Ramesh, S., Hebbar, R., Niveditha M. Pooja R., Bhat N., P., Shashank N., Vinod P.V., 2018. "Plant Disease Detection Using Machine Learning", International Conference on Design Innovations for 3CsCompute Communicate Control (ICDI3C). pp. 41-45

10. Ashar, D., Kanojia, A., Parihar, R., Kudoo, S., 2021. Livestock Disease Prediction System. IVA-Tech International Journal for Research and Innovation, 97 1, 1–3.

Appendix A: Problem Statement Feasibility Assessment

Feasibility Assessment

To assess the feasibility of the proposed system, we examined the logical structure, system modules, user interaction flow, and integration with external APIs. The system is feasible due to its reliance on well-established technologies such as convolutional neural networks (CNN), support vector machines (SVM), and cloud-based APIs like Google Maps. The combination of offline functionality and real-time analysis ensures accessibility and responsiveness.

1. Satisfiability Analysis and Computational Classification

We analyzed the system through the lens of modern algebra and computational theory to determine its computational complexity:

- **Satisfiability:** The core decision problems (e.g., symptom matching, diagnosis validation) are modeled using logical constraints. These constraints must be satisfiable for valid predictions and results.
- **Class P:** Tasks like image uploading, displaying results, and form submissions are solvable in deterministic polynomial time.
- **NP:** Disease prediction involves pattern recognition and similarity matching, where the solution is verifiable in polynomial time.
- **NP-Complete:** When extending the system to include optimal resource allocation (e.g., assigning veterinarians, scheduling visits under constraints), the problem maps to classical NP-Complete problems like Job Scheduling, Set Cover, and 0/1 Knapsack.
- **NP-Hard:** In scenarios involving large-scale interdepartmental coordination, tender routing, or epidemic forecasting, the solution becomes computationally intensive and aligns with NP-Hard problems like the Travelling Salesman Problem.

2. NP-Complete Classification Justification

The core computational problems addressed by our system—such as image-based disease prediction, optimal task scheduling, and constrained resource allocation—demonstrate characteristics commonly associated with NP-Complete problems. These problems often involve selecting the best configuration among numerous possibilities, subject to strict constraints like time, resources, and accuracy. For instance, assigning veterinary or agricultural experts to cases based on urgency, location, and specialization closely resembles the Job Scheduling Problem, a classical NP-Complete problem.

These problems are classified as NP-Complete due to several core reasons. First, any valid solution (such as a diagnosis or route plan) can be verified in polynomial time, even though finding the solution might not be feasible within polynomial time for all instances. Second, these problems can be reduced to other well-known NP-Complete problems, establishing their theoretical equivalence. Finally, no deterministic algorithm is currently known that can solve all NP-Complete problems efficiently in polynomial time, making them computationally intensive, particularly as input size scales.

Thus, the system spans multiple computational classes, with most real-time operations being in P/NP, while extended modules approach NP-Complete and NP-Hard classifications. This justifies the use of approximation and heuristic-based algorithms for complex tasks.

3. Why NP-Complete Classification Justification

The classification of a computational problem as **NP-Complete** is grounded in its logical complexity and the nature of constraints it involves. In the context of our Disease Diagnostic and Reporting Portal, several components exhibit characteristics that align closely with well-known NP-Complete problems. Below is a detailed justification:

Complexity of the Tasks

- **Task Allocation and Scheduling:** Assigning tasks such as allocating doctors or processing multiple diagnoses with overlapping resources is analogous to the *Job Scheduling* or *Graph Coloring Problem*—both of which are NP-Complete.

- **Tender Assignment with Constraints:** Selecting optimal bids within fixed budgets and deadlines is structurally similar to the *0/1 Knapsack Problem* or *Set Cover Problem*, both recognized as NP-Complete.
- **Routing and Coordination:** Routing specialists or disease samples across interconnected regions under time constraints can be modeled after the *Travelling Salesman Problem (TSP)*, another NP-Complete problem.

Appendix B: Details of Paper Publication

Paper Title

Farmers Disease Diagnostic/Reporting Portal– Mobile Portal AI-Based

Authors

Dr.B.S.Shirole, Yash Zolekar, Kunal Bhalerao, Vishal Lahane, Shaikh Saqlin Raza Alauddin

Name of Conference/Journal

JETIR Journal of Emerging Technologies and Innovative Research

Reviewer Comments

- Reviewer 1: "The methodology is strong, but more experiments are suggested."
- Reviewer 2: "Paper is well-written; minor grammatical edits needed."

paper

© 2025 JETIR March 202 www.jetir.org (ISSN-2349-5162)

JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue

JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)

An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Farmers Disease Diagnostic/Reporting Portal– Mobile Portal AI-Based

Dr.B.S.Shirole[1], Yash Zolekar[2], Kunal Bhalerao[3], Vishal Lahane[4], Shaikh Saqlin Raza Alauddin[5] Department of computer engineering, SMES
Sanghavi College Of Engineering, Nashik,
Savitribai Phule Pune University (SPPU)

Abstract: The Plant and Animal Diseases Portal provides a comprehensive platform for monitoring, diagnosing, and managing various diseases affecting plants and animals. This portal integrates machine learning and data analytics to identify disease patterns, track outbreaks, and offer diagnostic support to agricultural and veterinary professionals. Users can access a database of symptoms, images, and recommended treatments, aiding in early detection and containment. This system aims to support farmers, researchers, and policymakers in protecting biodiversity, and enhancing the health of agricultural resources through real-time information and disease management tools.

Keywords: symptoms matching, image classification, object detection, model integration

I. Introduction

The increasing prevalence of plant and animal diseases poses a significant threat to global food security, biodiversity, and economic stability. Timely detection and accurate diagnosis are crucial for risks to human health. By creating a centralized portal for disease detection and management, we aim to address several key issues: Problem Statement Farmers and veterinarians face challenges in identifying plant and animal diseases in their early stages, leading to delayed intervention and the spread of infections. The scarcity of affordable and user-friendly diagnostic tools contributes to poor disease management. Developing a portal that includes AI-driven diagnostic tools, image-based detection, and symptoms analysis for early disease identification could enable users to address issues promptly and minimize damage. Managing these diseases effectively, minimizing losses, and ensuring the health and sustainability of agricultural and natural ecosystems. The Plant and Animal Diseases Portal aims to serve as a centralized platform for the monitoring, diagnosis, and management of diseases affecting various plant and animal species. Moreover, this portal fosters collaboration among farmers, veterinarians, researchers, and policymakers, providing them with a shared platform to share insights, report outbreaks, and access up-to-date information on disease management practices. Ultimately, this initiative aims to reduce the spread of diseases, protect agricultural resources, and support sustainable farming practices. This report outlines the portal's design, core functionalities, and the potential benefits it offers to stakeholders in agricultural and environmental sectors. The motivation behind developing an Animal and Plant Disease Portal stems from the significant impact that diseases in plants and animals have on agriculture, economy, and public health. Diseases in plants can lead to severe crop losses, affecting food security and farmers' livelihoods similarly, diseases in animals, especially livestock, can impact food production, animal welfare, and even pose risks to human health. By creating a centralized portal for disease detection and management, we aim to address several key issues.

Problem Statement: Farmers and veterinarians face challenges in identifying plant and animal diseases in their early stages, leading to delayed intervention and the spread of infections. The scarcity of affordable and user-friendly diagnostic tools contributes to poor disease management. Developing a portal that includes AI-driven diagnostic tools, image based detection, and symptoms analysis for early disease identification could enable users to address issues promptly and minimize damage.

© 2025 JETIR March 2025, Volume 12, Issue 3 www.jetir.org (ISSN-2349-5162)

Objectives:

- Flutter Mobile App (Dart): User Interface for viewing disease information. Disease detection interface (upload photos, select plant/animal type, etc.).
- Python Backend: API services for disease detection and information retrieval. ML model integration for detecting diseases from images (optional). Data storage and reporting system.
- Database: Stores disease information, reports, and user-contributed data. .

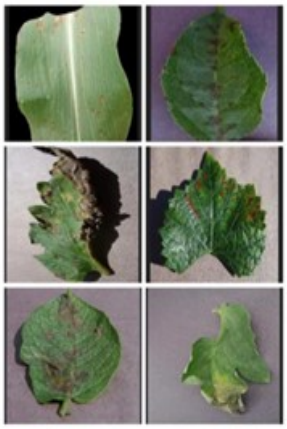


Fig. 1. Dataset Images

II. Literature Survey

A. Machine Learning-Based Approaches

ML models analyse handcrafted features from plant leaf images. Popular algorithms include Support Vector Machines (SVM), Convolutional neural network (CNN), deep learning

B. Deep Learning Approaches

Convolutional Neural Networks (CNNs) can automatically extract hierarchical features from images. Models like ResNet, VGG16, InceptionV3, and MobileNet are widely used for plant and animal disease classification.

C. Mobile Applications for Disease Detection

Several mobile apps use image processing and AI to detect plant diseases. Mobile-based animal disease detection is emerging with AI-powered veterinary tools.

1) G. Shrestha, Deepsikha, M. Das, and N. Dey proposed in 2020 "Plant Disease Detection Using CNN", the agricultural output of the nation is impacted by pest-infected plants and crops. To find and detect diseases, farmers or experts typically monitor the plants closely. But this process is often expensive, time-consuming, and inaccurate. Finding a spot on the leaves of the infected plant is one way to identify plant diseases. This work aims to develop a disease recognition model that is backed by the classification of leaf images. We are using image processing with a convolution neural network (CNN) to identify plant illnesses. A type of artificial neural network designed especially to process pixel input and utilised in image identification is called a convolutional neural network (CNN)
2) Dr. Sharada K.A, Najma Taj, Rida Sameer, Rukhsha Khan, Ruzaina Zareen proposed in 2024 "Detection of Lumpy Skin Disease in Cattle Using IoT and Deep Learning Techniques", A virus belonging to the Capripoxvirus genus, which is a member of the Poxviridae family, causes Lumpy Skin Disease (LSD), a contagious illness in cattle. It is thought that the virus is transferred by biological vectors such as mosquitoes, flies, ticks, and direct contact. Numerous economic costs, including those in fertility, milk production, trade tariffs, and in certain cases, livestock animal mortality, are attributed to this virus. Additionally, the ocular

© 2025 JETIR March 2025, Volume 12, Issue 3 www.jetir.org (ISSN-2349-5162)

secretions and pus of infected calves were found to contain the LSD virus. Because of the Web of Things and smart devices, industries including agriculture and livestock are undergoing significant change. We find it very difficult to imagine living in a world without the Internet of Things, where most objects are connected to one another.

3) M.R. Srivalli, N.K. Vishnu, and V. Kanchana, proposed in 2022"Teat and Udder Disease Detection Cattle Using Machine Learning", in animals that produce a lot of milk, udder and teat illnesses are prevalent. As a result, producers are experiencing increased challenges with milk output and quality. Research on automatic cow disease is crucial for agriculture's large-scale population monitoring, and the disease should be automatically identified as soon as signs show up on the udder or teat. An automated system is required to identify cow teat and udder problems, determine how they affect milk production, and determine the best medications to treat them. This work aims to address the issue of identifying and preventing illnesses in various cow breeds. A number of models are examined in order to identify the best deep learning architecture.
4) K. K. Ferentinos, proposed in 2018"Deep Learning Models for Plant Disease Detection and Diagnosis", Computers and Electronics in Agriculture. Employing basic leaf photos of both healthy and ill plants, convolutional neural network models were created in this article employing deep learning techniques to detect and diagnose plant diseases. An open database including 87,848 photos of 25 distinct plants in 58 different classes of

[Plant, disease] combinations—including healthy plants—was used to train the models.

5) M.H. Saleem, J. Potgieter, and K.H. Arif, proposed in 2019 "Plant Disease Detection and Classification by Deep Learning," Plants Traditional leaf and crop disease detection is labor-intensive, requiring expert manpower and time. AI and deep learning offer faster, more accurate solutions, reducing labor and economic impact. This review analyses deep learning techniques for plant disease detection, summarizing methods, datasets, diseases, plant types, performance metrics, and results. Challenges and future research directions are also discussed.
6) P. Kumar, Madhu, J. Kumar, and C. Sathish, proposed in 2017 "Health Experts for Pets Using Mobile Apps", Our lives are greatly impacted by mobile technologies, and as these technologies improve, end users can enable new kinds of healthcare systems using rule-based expert systems. Particularly, the availability of more affordable smartphones and a more user-friendly graphical user interface (GUI) based on the Android OS opens up new opportunities for continuously monitoring the health state of pets, including dogs and cats, as well as any harmful ingestions or swallowed objects. The relevant features that are typically offered to users by the suggested applications are extensive and crucial to clinical practice and health management. Pets can be protected against disease attacks by using the pet mobile app. In an emergency, we can use this app to schedule an online appointment with a pet professional.
7) S. Neethirajan, proposed in 2017 "Recent Advances in Wearable Sensors for Animal Health Management" Biosensors for animal health management are rapidly gaining global recognition. Various sensors are at different commercialization stages, with some human health technologies being adapted for animals. These innovations are now being explored for livestock development and welfare. Precision livestock farming incorporates advanced technologies like microfluidics, sound analysers, image detection, sweat and salivary sensing, and serodiagnosis. However, integrating these sensors into an efficient real-time monitoring system is essential. This review discusses wearable technologies, nano biosensors, and molecular diagnostics for detecting infectious cattle diseases, comparing their advantages and drawbacks to guide future advancements in animal health and welfare.

III. Proposed System Overview

(A) For Animal Step 1: Select Animal

The user selects an animal type (e.g., Cow, Dog, Cat, Goat, etc.). The system filters the relevant disease database for the chosen animal. Why? Different animals have different diseases, so selecting an animal helps the system focus on relevant diseases.

Step 2: Enter Symptoms

The user manually enters the observed symptoms (e.g., fever, loss of appetite, coughing, weakness). The system converts symptoms into numerical features for processing by the SVM model. Example Feature Encoding: Fever: 1 Weakness: 1 Loss of Appetite: 0 Coughing: 1

© 2025 JETIR March 2025, Volume 12, Issue 3 www.jetir.org (ISSN-2349-5162)

Step 3: Predict Button (Triggering the SVM Model)

The user clicks the Predict button. The system sends the symptom feature vector to the SVM model for disease classification. Why? This triggers the machine learning model to analyse the input and predict the disease.

Step 4: Apply SVM Model (Feature Extraction Classification)

A. How SVM Works in This Step?

Training Phase (Before Deployment): The SVM model is trained on a dataset of animal symptoms and diseases. Each disease is treated as a class (e.g.,"Foot and Mouth Disease," "Rabies," "Influenza"). The model learns the patterns that distinguish one disease from another. The algorithm finds an optimal hyperplane that separates different diseases in high-dimensional space.

Prediction Phase (After User Input):

- The system maps the symptom vector to the trained SVM model.

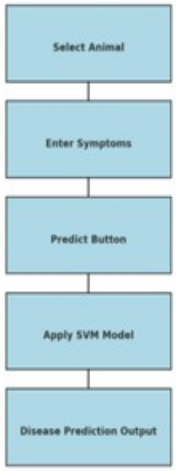


Fig. 2. System Flow Diagram for Animal Disease Detection App

- The decision boundary (hyperplane) determines which disease category the input belongs to.
- If the data is not linearly separable, an SVM Kernel Trick (e.g., RBF Kernel) is used to transform it into a higher dimension for better separation.

Example Hyperplane Classification in SVM: Suppose the symptoms match *Foot and Mouth Disease*, and the SVM classifier assigns it with a 92% confidence score.

step5: Disease Prediction Output

The system displays the predicted disease and a confidence score (how certain the model is about the classification). The user can take further action based on the result (e.g., consult a veterinarian).

Example Output:

Animal: sheep

Predicted Disease: Foot Root

Description: Bacterial infection of the foot, swelling precaution: Maintain clean and dry living, treat hooves

Medication: Antibiotics (e.g. Penicillin), foot care and proper sanitation

© 2025 JETIR March 2025, Volume 12, Issue 3 www.jetir.org (ISSN-2349-5162)

(B) For Plant

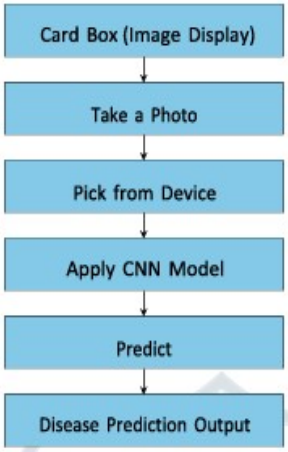


Fig. 3. System Flow Diagram for Plant Disease Detection App

Step 1: Card Box (Image Display):

This is the UI component where the user can view the selected image. It serves as a preview before running the model.

- It ensures the user selects the correct image before processing.

Step 2: Take a Photo

- The user can capture an image of the affected plant.
- This image is directly sent for processing.

Step 3: Pick from Device

- The user can select an existing image from their gallery or storage.
- This allows using previously captured images. Why?
- Flexibility in image selection ensures better usability.

Step 4: Apply CNN Model How CNN Works in This Step?

- The image is pre-processed (resized, normalized).
- It is fed into a pre-trained CNN model (e.g., ResNet, VGG16, or MobileNet).
- The CNN extracts important features (textures, shapes, colours) through multiple layers:
  - Convolutional Layers → Extract features using filters.
  - Pooling Layers → Reduce dimensions while retaining key information. – Fully Connected Layers → Classify extracted features into diseases.

Example Feature Extraction:

- A CNN learns to recognize disease-specific patterns, e.g., brown spots on leaves for fungal infections.

*Step 5: Predict

- The model classifies the image into a specific disease category.
- It assigns a confidence score to the prediction. Example Output from CNN:
- Healthy (95% confidence)
- Leaf Blight (87% confidence)
- Powdery Mildew (90% confidence)

Step 6: Disease Prediction Output

- The system displays the predicted disease name, diseases, Description, Precaution and Medication

© 2025 JETIR March 2025, Volume 12, Issue 3 www.jetir.org (ISSN-2349-5162)

- The user can take action, such as consulting an expert. Example Final Output:

Predicted Disease: Powdery Mildew
Confidence Score: 90%

IV. IMPLEMENTATION AND RESULT

(A) Register Page

Fig. 4. Register Screen of the Mobile App

The image shows a Register screen of a mobile app with a beach-themed background. It includes:

- Input Fields: Username, Email, and Password.
- Register Button: A green button for user registration.
- Login Option: A link for existing users to log in.
- Floating Action Button (FAB): Likely for additional options or navigation.

(B) Login Page

The image shows a Login screen of a mobile app with a calm lake-themed background.
It includes:

- Input Fields: Email and Password.
- Login Button: A green button for user authentication.
- Register Option: A link for new users to sign up.
- Floating Action Button (FAB): Likely for additional options or navigation.

© 2025 JETIR March 2025, Volume 12, Issue 3 www.jetir.org (ISSN-2349-5162)

Fig. 5. Login Screen of the Mobile App

The design is clean and nature-themed, focusing on AI-driven plant and animal health monitoring.

(C) Home Page

Fig. 6. Home Page

The image shows a mobile app interface for Plant and Animal Disease Detection. It features:

- Top Bar: Menu icon,"Welcome" text, and refresh button.
- Main Content: Logo, app title, and a brief AI-powered disease detection description.
- Bottom Navigation: Four tabs – Home (active), Plant, Animal, and Contact.

The design is clean and nature-themed, focusing on AI-driven plant and animal health monitoring.

© 2025 JETIR March 2025, Volume 12, Issue 3 www.jetir.org (ISSN-2349-5162)

(D) Plant disease Detection Result Page

Fig. 7. Detected Disease: Pepper Bell Bacterial Spot

Class Index: 0

Confidence: 90.72% article graphic

(E) Animal disease Detection Result Page

This document describes a mobile application designed for predicting animal diseases based on user-inputted symptoms.

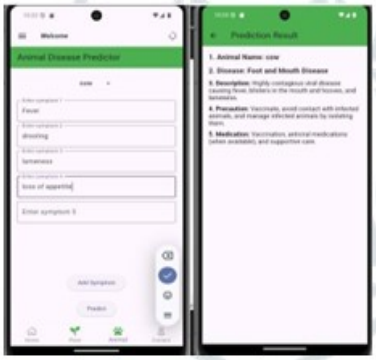


Fig. 8. Animal Disease Predictor Application Screens

On the left screen, the user selects an animal (e.g., cow) from a dropdown menu and enters symptoms in the provided text fields. In the example shown, the user has entered:

- Fever
- Drooling
- Lameness
- Loss of appetite

After entering symptoms, the user clicks the Predict button to get a diagnosis.

On the right screen, the application displays the predicted disease and relevant information. For the given symptoms, the app identifies the disease as the result includes:

1) Animal Name: Cow
2) Disease: Foot and Mouth Disease
3) Description: Highly contagious viral disease causing fever, blisters in the mouth and hooves, and lameness.
4) Precaution: Vaccinate, avoid contact with infected animals, and manage infected animals by isolating them.
5) Medication: Vaccination, antiviral medications (when available), and supportive care. (E) Contact Us Page

© 2025 JETIR March 2025, Volume 12, Issue 3 www.jetir.org (ISSN-2349-5162)

Fig. 9. Contact Us Page

V. Challenges and Future Directions

Despite the significant advancements in assistive technologies, several challenges remain to be addressed for these devices to become more effective and accessible.

- Improving ML Algorithms: Suggest future improvements, such as incorporating more advanced AI models or multi-modal data.
- Expansion to More Diseases and Regions: Discuss plans for expanding the portal to cover more diseases and geographic areas.
- Integration with Government Policies: Propose how this portal could integrate with public health and agriculture departments to manage disease outbreaks.
- Integration with AI and Machine Learning: There is potential to enhance these technologies through AI-driven personalization, allowing devices to learn and adapt to a user's specific navigation patterns and behaviours over time.
- Cost and Accessibility: Advanced assistive devices often come with high costs due to sophisticated software requirements, limiting their availability to users who need them.

VI. Conclusion

The development of a mobile portal for plant and animal disease detection using Flutter, Dart, and Python presents a significant advancement in agricultural and veterinary diagnostics. By integrating machine learning models with a cross platform framework, we have created an application that is accessible, efficient, and user-friendly. This solution addresses the critical need for early disease detection, helping farmers, veterinarians, and hobbyists protect plant and animal health and reduce economic losses.

References

[1] G. Shrestha, Deepsikha, M. Das and N. Dey, "Plant Disease Detection Using CNN," *IEEE Access*, vol. 2020, pp. 109-113, doi: 10.1109/ASPCON49795.2020.9276722 .

[2] K.A Sharada, Najma Taj, Rida Sameer and Rukhsha Khan, "Ruzaina Zareen", Detection of Lumpy Skin Disease in cattle using IOT and Deep Learning Techniques International Journal of Advanced, vol. 3, no. 2,2023, pp. 64-78.

© 2025 JETIR March 2025, Volume 12, Issue 3 www.jetir.org (ISSN-2349-5162)

[3] MR Srivalli, NK Vishnu and V. Kanchana, "Teat and Udder Disease Detection on Cattle using Machine Learning", 2022 International Conference on Signal and Information Processing (IConSIP), pp. 1-5, 2022 Aug 26.

[4] K. Ferentinos, "Deep learning models for plant disease detection and diagnosis", Computers and Electronics in Agriculture, vol. 145, pp. 311-318, 2018.

[5] M. H. Saleem, J. Potgieter and K. H. Arif, "Plant disease detection and classification by deep learning", Plants, vol. 8, no. 11, pp. 468, 2019.

[6] P. Kumar, Madhu, J. Kumar and C. Sathish, "Health experts for pets using mobile apps", 2017 International Conference on Algorithms Methodology Models and Applications in Emerging Technologies (ICAMMAET), pp. 1-2, Feb. 2017.

[7] S. Neethirajan, "Recent advances in wearable sensors for animal health management", Sens. Bio-Sens. Res., vol. 12, pp. 15-29, Feb. 2017.

[8] Liu, J., Wang, X., 2021. "Plant diseases and pests detection based on deep learning: a review", Plant Methods 17:22. pp. 1 18.

[9] Ramesh, S., Hebbar, R., Niveditha M. Pooja R., Bhat N., P., Shashank N., Vinod P.V., 2018. "Plant Disease Detection Using Machine Learning", International Conference on Design Innovations for 3CsCompute Communicate Control (ICDI3C). pp. 41-45

[10] Ashar, D., Kanojia, A., Parihar, R., Kudoo, S., 2021. Livestock Disease Prediction System. IVA-Tech International Journal for Research and Innovation, 97 1, 1–3.

Certificate

Journal of Emerging Technologies and Innovative Research
An International Open Access Journal Peer-reviewed, Refereed Journal
www.jetir.org | editor@jetir.org An International Scholarly Indexed Journal

Certificate of Publication

The Board of
Journal of Emerging Technologies and Innovative Research (ISSN : 2349-5162)
Is hereby awarding this certificate to
Yash Zolekar
In recognition of the publication of the paper entitled
Farmers Disease Diagnostic/Reporting Portal– Mobile Portal AI-Based
Published In JETIR (www.jetir.org) ISSN UGC Approved (Journal No: 63975) & 7.95 Impact Factor
Published in Volume 12 Issue 3 , March-2025 | Date of Publication: 2025-03-15

EDITOR EDITOR IN CHIEF
JETIR2503267 Research Paper Weblink http://www.jetir.org/view?paper=JETIR2503267 Registration ID : 556486
An International Scholarly Open Access Journal, Peer-Reviewed, Refereed Journal Impact Factor Calculate by Google Scholar and Semantic Scholar | AI-Powered Research Tool, Multidisciplinary, Monthly, Multilanguage Journal Indexing in All Major Database & Metadata, Citation Generator

Journal of Emerging Technologies and Innovative Research
An International Open Access Journal Peer-reviewed, Refereed Journal
www.jetir.org | editor@jetir.org An International Scholarly Indexed Journal

Certificate of Publication

The Board of
Journal of Emerging Technologies and Innovative Research (ISSN : 2349-5162)
Is hereby awarding this certificate to
Vishal Iahane
In recognition of the publication of the paper entitled
Farmers Disease Diagnostic/Reporting Portal– Mobile Portal AI-Based
Published In JETIR (www.jetir.org) ISSN UGC Approved (Journal No: 63975) & 7.95 Impact Factor
Published in Volume 12 Issue 3 , March-2025 | Date of Publication: 2025-03-15

EDITOR EDITOR IN CHIEF
JETIR2503267 Research Paper Weblink http://www.jetir.org/view?paper=JETIR2503267 Registration ID : 556486
An International Scholarly Open Access Journal, Peer-Reviewed, Refereed Journal Impact Factor Calculate by Google Scholar and Semantic Scholar | AI-Powered Research Tool, Multidisciplinary, Monthly, Multilanguage Journal Indexing in All Major Database & Metadata, Citation Generator

Journal of Emerging Technologies and Innovative Research
An International Open Access Journal Peer-reviewed, Refereed Journal
www.jetir.org | editor@jetir.org An International Scholarly Indexed Journal

Certificate of Publication

The Board of
Journal of Emerging Technologies and Innovative Research (ISSN : 2349-5162)
Is hereby awarding this certificate to
Shaikh Saqlin Raza Alauddin
In recognition of the publication of the paper entitled
Farmers Disease Diagnostic/Reporting Portalâ€" Mobile Portal AI-Based
Published In JETIR (www.jetir.org) ISSN UGC Approved (Journal No: 63975) & 7.95 Impact Factor
Published in Volume 12 Issue 3 , March-2025 | Date of Publication: 2025-03-15

EDITOR EDITOR IN CHIEF

JETIR2503267 Research Paper Weblink http://www.jetir.org/view?paper=JETIR2503267 Registration ID : 556486

An International Scholarly Open Access Journal, Peer-Reviewed, Refereed Journal Impact Factor Calculate by Google Scholar and Semantic Scholar | AI-Powered Research Tool, Multidisciplinary, Monthly, Multilanguage Journal Indexing in All Major Database & Metadata, Citation Generator

Journal of Emerging Technologies and Innovative Research
An International Open Access Journal Peer-reviewed, Refereed Journal
www.jetir.org | editor@jetir.org An International Scholarly Indexed Journal

Certificate of Publication

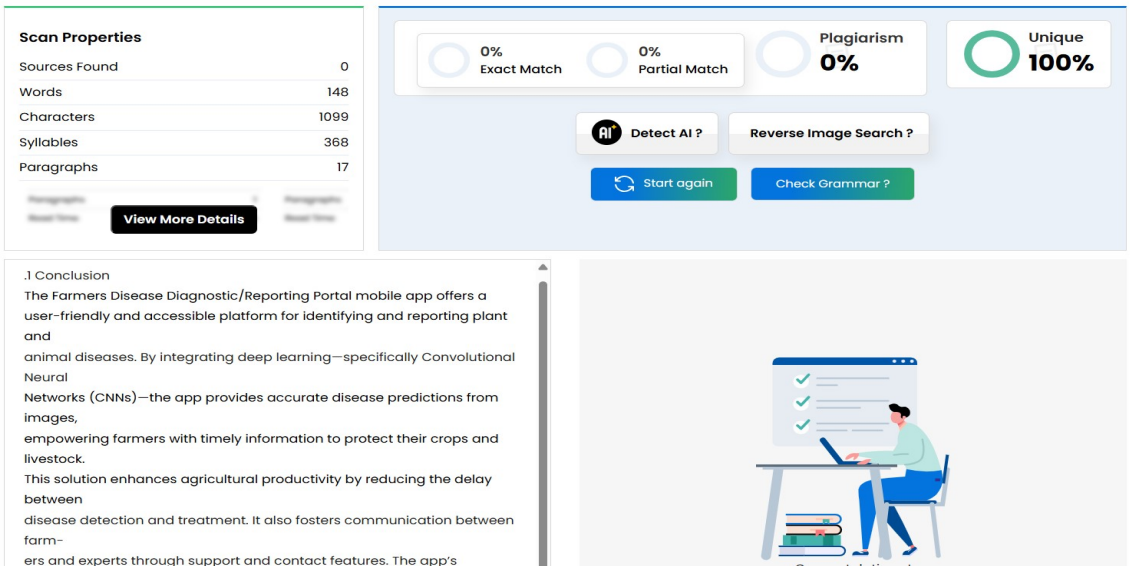
The Board of
Journal of Emerging Technologies and Innovative Research (ISSN : 2349-5162)
Is hereby awarding this certificate to
Kunal Bhalerao
In recognition of the publication of the paper entitled
Farmers Disease Diagnostic/Reporting Portal– Mobile Portal AI-Based
Published In JETIR (www.jetir.org) ISSN UGC Approved (Journal No: 63975) & 7.95 Impact Factor
Published in Volume 12 Issue 3 , March-2025 | Date of Publication: 2025-03-15

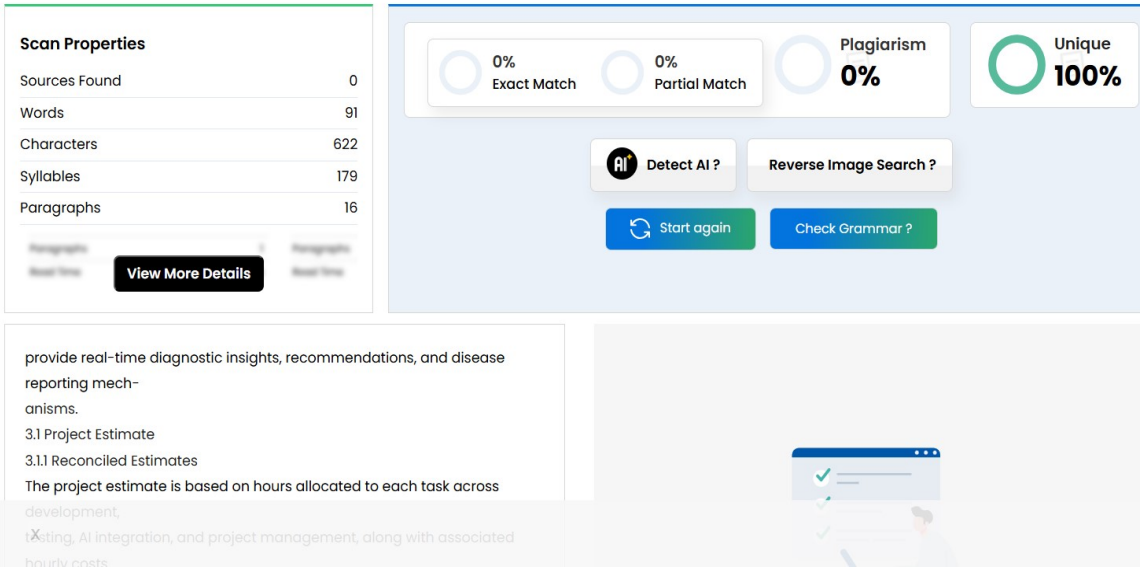
EDITOR EDITOR IN CHIEF

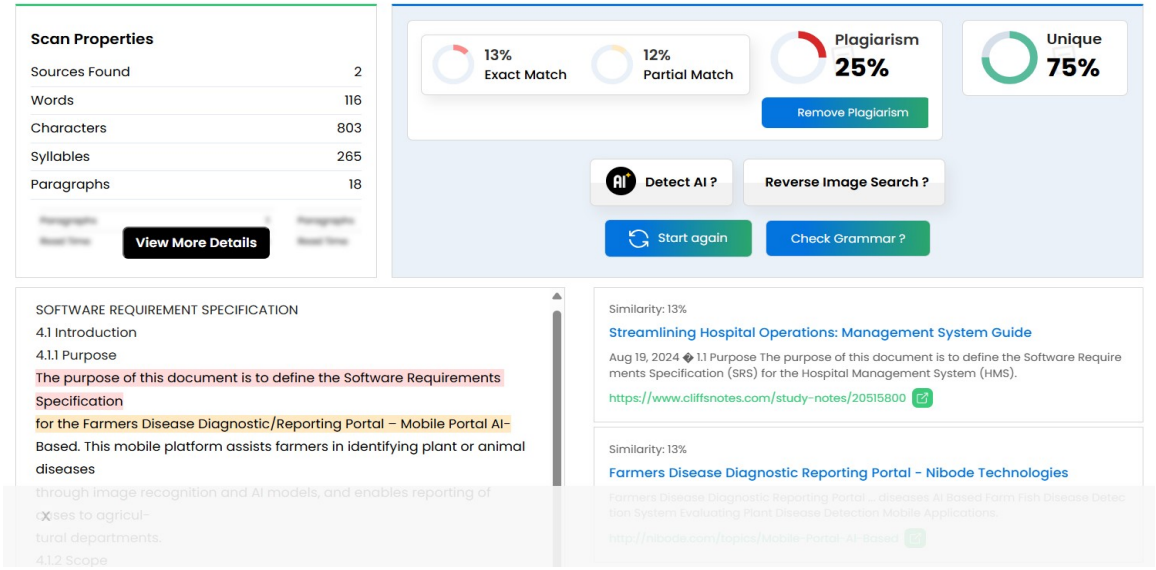
JETIR2503267 Research Paper Weblink http://www.jetir.org/view?paper=JETIR2503267 Registration ID : 556486

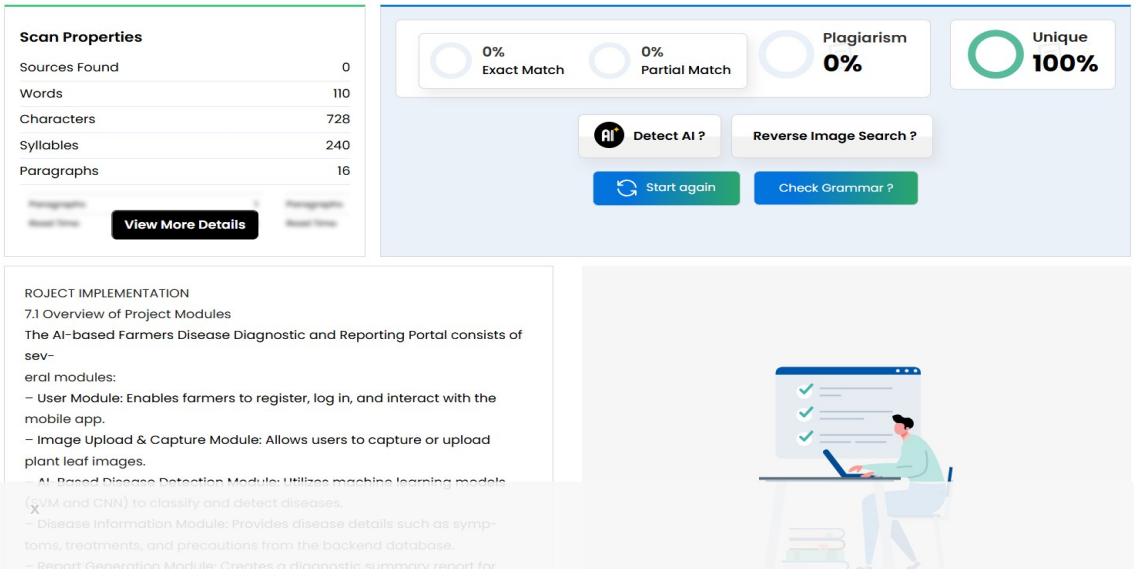
An International Scholarly Open Access Journal, Peer-Reviewed, Refereed Journal Impact Factor Calculate by Google Scholar and Semantic Scholar | AI-Powered Research Tool, Multidisciplinary, Monthly, Multilanguage Journal Indexing in All Major Database & Metadata, Citation Generator

Appendix C: Plagiarism Report






Scan Properties
Sources Found 2
Words 116
Characters 803
Syllables 265
Paragraphs 18
View More Details
13% Exact Match
12% Partial Match
Plagiarism 25%
Unique 75%
Remove Plagiarism
Detect AI ?
Reverse Image Search ?
Start again
Check Grammar ?
SOFTWARE REQUIREMENT SPECIFICATION
4.1 Introduction
4.1.1 Purpose
The purpose of this document is to define the Software Requirements Specification
for the Farmers Disease Diagnostic/Reporting Portal – Mobile Portal AI-Based. This mobile platform assists farmers in identifying plant or animal diseases
through image recognition and AI models, and enables reporting of cases to agricultural departments.
4.1.2 Scope
Similarity: 13%
Streamlining Hospital Operations: Management System Guide
Aug 19, 2024 1.1 Purpose The purpose of this document is to define the Software Requirements Specification (SRS) for the Hospital Management System (HMS).
https://www.cliffsnotes.com/study-notes/20515800
Similarity: 13%
Farmers Disease Diagnostic Reporting Portal - Nibode Technologies


Scan Properties
Sources Found 0
Words 110
Characters 728
Syllables 240
Paragraphs 16
View More Details
0% Exact Match
0% Partial Match
Plagiarism 0%
Unique 100%
Detect AI ?
Reverse Image Search ?
Start again
Check Grammar ?
ROJECT IMPLEMENTATION
7.1 Overview of Project Modules
The AI-based Farmers Disease Diagnostic and Reporting Portal consists of several modules:
– User Module: Enables farmers to register, log in, and interact with the mobile app.
– Image Upload & Capture Module: Allows users to capture or upload plant leaf images.
– AI-Based Disease Detection Module: Utilizes machine learning models (SVM and CNN) to classify and detect diseases.
– Disease Information Module: Provides disease details such as symptoms, treatments, and precautions from the backend database.
– Report Generation Module: Creates a diagnostic summary report for