**Project Report Format**

**INTRODUCTION:**

* 1. **project overview:**

Poultry farming plays a vital role in food security and economic sustainability for both smallholder farmers and large-scale commercial producers. However, the sector faces significant challenges due to infectious diseases that can rapidly spread and devastate flocks. Timely and accurate diagnosis is critical to mitigating losses and maintaining animal welfare. Yet, many farmers—especially in rural and underserved regions—lack access to veterinary support and diagnostic infrastructure.

To address this gap, this project proposes the development of a **Transfer Learning-based machine learning system** designed to classify poultry diseases into four key categories: **Salmonella, New Castle Disease, Coccidiosis**, and **Healthy**. The system will be integrated into a **mobile application**, offering farmers and veterinary practitioners a user-friendly, AI-powered diagnostic tool.

**1.2** **purpose**:

The purpose of this project is to develop a Transfer Learning-based classification system integrated into a mobile application to enable rapid, accurate diagnosis of common poultry diseases—Salmonella, New Castle Disease, and Coccidiosis—alongside identification of healthy birds. This solution aims to empower farmers, veterinarians, and agricultural stakeholders with an accessible tool to improve disease detection, treatment, and overall poultry health management, especially in regions with limited access to veterinary services. By facilitating early intervention and informed decision-making, the system helps reduce mortality rates, minimize economic losses, and enhance the productivity and sustainability of poultry farming operations.

**2. IDEATION PHASE**:

**2.1 PROBLEM STATEMENT:**

Poultry farmers, especially in rural areas, often struggle to diagnose and manage diseases like Salmonella, New Castle Disease, and Coccidiosis due to limited access to veterinary services and diagnostic tools. Traditional methods are often slow, expensive, and require expertise that is not readily available. This leads to delayed treatment, high mortality rates, and significant economic losses. There is a clear need for a fast, affordable, and accessible solution to enable early detection and effective disease management in poultry farming.

**2**.**2 EMPATHY MAP CANVAS:**

* Who?  
  Rural poultry farmer with limited veterinary access.
* Needs:  
  Quick, affordable disease diagnosis and treatment advice.
* Sees:  
  Sick birds, no nearby veterinary help.
* Says/Does:  
  Asks neighbors, tries home remedies.
* Hears:  
  Local advice, word-of-mouth info.
* Thinks/Feels:  
  Pains: Fear of loss, stress, uncertainty.  
  Gains: Confidence, timely action, healthier flock.

**2.3 BRAINSTROMING:**

**Problems:**

* Delayed diagnosis in rural areas
* High poultry deaths, low access to vets

**Solutions:**

* AI-based mobile app using transfer learning
* Simple symptom input + treatment advice
* Offline access for remote users

**3.REQUIREMENT ANALYSIS:**

**3.1 customer journey map:**

**Customer Journey Map – Smallholder Poultry Farmer**

| **Stage** | **Actions** | **Feelings** | | **Challenges** | | **Opportunities** | |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Awareness | Hears about app | Curious, unsure | | Distrust of new tech | | Local demos, simple messaging | |
| Onboarding | Downloads app | Hopeful | | Limited tech skills | | Easy UI, tutorials | |
| Symptom Input | Enters symptoms | | Anxious | | Unsure what symptoms mean | | Guided questions | |
| Diagnosis | Gets results | | Relieved or worried | | Confusing results | | Clear, simple advice | |
| Action | Treats birds | | Motivated | | Cost concerns | | Suggest affordable treatments | |
| Follow-up | Monitors health | | Hopeful | | Fear of relapse | | Reminders and alerts | |

**3.2 solution Requirements:**

1. Easy and simple app interface.
2. Accurate and fast disease diagnosis.
3. Input symptoms and environment data.
4. Clear treatment advice.
5. Works offline.
6. Supports local languages.
7. Keeps health records.
8. Affordable for farmers.
9. Includes basic disease info.

**3.3 Data flow diagram:**

**Entities:**

* User (Farmer/Vet Student/Manager)
* Mobile Application
* ML Disease Classification Model
* Database

Processes:

1. User Inputs Data
   * Symptoms, environmental conditions, (optional) images.
2. Data Validation & Preprocessing
   * App checks data completeness and formats inputs.
3. Disease Classification
   * Transfer learning model analyzes data and predicts disease.
4. Generate Diagnosis & Treatment Advice
   * System outputs diagnosis and recommended treatment.
5. Store Data
   * Saves input data and results to database for history and monitoring.
6. User Receives Results
   * Diagnosis and advice shown on app.

Data Flows:

* User → Inputs symptoms/environment → Mobile App
* Mobile App → Validated data → ML Model
* ML Model → Diagnosis → Mobile App
* Mobile App → Diagnosis + treatment advice → User
* Mobile App ↔ Data stored/retrieved → Database

**3.4 Technology stack:**

**1. Frontend (Mobile Application)**

* **Framework:** Flutter or React Native (cross-platform for Android & iOS)
* **UI Design:** Material Design or custom lightweight UI components
* **Local Storage:** SQLite or Shared Preferences for offline data storage

**2. Backend**

* **Server:** Node.js with Express.js or Python Flask/Django (optional if cloud needed)
* **Database:** Firebase Firestore (cloud-based) or PostgreSQL / MongoDB for structured data
* **API:** RESTful API to connect app and backend

**3. Machine Learning**

* **Model:** Transfer learning with pre-trained CNNs like MobileNet, ResNet
* **Framework:** TensorFlow or PyTorch for training
* **Deployment:** TensorFlow Lite or ONNX Runtime for on-device inference
* **Data Processing:** Python (Pandas, NumPy) for preprocessing training data

**4. Other Tools**

* **Version Control:** Git/GitHub or GitLab
* **CI/CD:** GitHub Actions, Bitrise, or Firebase App Distribution
* **Localization:** i18n libraries for multilingual support
* **Analytics:** Firebase Analytics or Google Analytics for usage data

**4. PROJECT DESIGN:**

**4.1 problem solution pit:**

**Problem:**Poultry farmers, especially in rural areas, face frequent disease outbreaks like Salmonella, New Castle Disease, and Coccidiosis. They often lack quick, reliable diagnostic tools and access to veterinary support, resulting in delayed treatment, high mortality, and economic losses.

**Solution:**We propose a Transfer Learning-based mobile application that allows farmers to input symptoms and environmental data, quickly diagnosing poultry diseases with high accuracy. The app provides immediate treatment recommendations, works offline, supports local languages, and empowers farmers to manage flock health proactively, reducing losses and improving productivity.

**4.2 proposed solution:**

We propose developing a mobile application powered by transfer learning-based machine learning models to classify poultry diseases into four categories: Salmonella, New Castle Disease, Coccidiosis, and Healthy. The app will enable farmers and poultry managers to input observable symptoms, environmental data, and optionally biological samples or images. The system will rapidly analyze the input and deliver an accurate diagnosis along with tailored treatment and management recommendations.

**Key features of the solution include:**

* **User-friendly interface designed for farmers with varying digital literacy.**
* **Offline functionality to ensure usability in remote areas with poor internet connectivity.**
* **Multilingual support to accommodate local languages.**
* **Data logging for monitoring flock health trends over time.**
* **Real-time alerts and notifications for early outbreak detection.**

By empowering users with timely, accessible disease diagnostics, this solution aims to reduce poultry mortality, minimize economic losses, and improve overall farm productivity and health management.

**4.3 solution architecture:**

**1. User Interface (Mobile App)**

* **Platform: Android & iOS (built with Flutter or React Native)**
* **Functions:**
  + User input: symptoms, environmental data, optional images
  + Display diagnosis and treatment recommendations
  + Offline data storage and syncing when online
  + Multilingual support

**2. Data Processing Layer**

* Preprocessing user inputs for model compatibility
* Data validation and formatting

**3. Machine Learning Model**

* Transfer learning-based CNN model (e.g., MobileNet or ResNet)
* Runs inference locally on device using TensorFlow Lite or ONNX Runtime
* Classifies input into Salmonella, New Castle Disease, Coccidiosis, or Healthy

**4. Backend Server (Optional)**

* Stores user data and diagnostic history (Firebase or cloud DB)
* Provides updates to ML model and treatment guidelines
* Handles user authentication and analytics

**5. Database**

* Cloud database (Firestore, MongoDB, or PostgreSQL)
* Stores user profiles, input records, diagnosis results, and app content

**6. Notification System**

* Push notifications for reminders, alerts on outbreaks, or treatment updates

**Data Flow Summary:**

User inputs symptoms → Data preprocessing → ML model (on device) → Diagnosis & treatment generated → Results displayed to user → Data synced to backend for storage and analytics (when online)

**A diagram sketch for this architecture :**

Here’s a simple text-based sketch of the Solution Architecture Diagram that you can visualize or use to create a graphic**:**

**+-------------------+ +------------------+ +-------------------+**

**| | | | | |**

**| User (Farmer) | <---> | Mobile App UI | <---> | Local ML Model |**

**| | | (Flutter/React) | | (TensorFlow Lite) |**

**+-------------------+ +------------------+ +-------------------+**

**|**

**v**

**+--------------------+**

**| Data Preprocessing |**

**| & Validation |**

**+--------------------+**

**|**

**v**

**+-----------------------+**

**| Backend Server (Cloud) |**

**| - User Data Storage |**

**| - Model Updates |**

**| - Authentication |**

**+-----------------------+**

**|**

**v**

**+--------------------+**

**| Database |**

**| (Firestore/Mongo) |**

**+--------------------+**

**|**

**v**

**+--------------------+**

**| Notification System |**

**+--------------------+**

**How it works:**

1. User interacts with the Mobile App UI to input symptoms and get results.
2. The Data Preprocessing module cleans and formats data for the model.
3. The Local ML Model runs inference on the device and returns the diagnosis.
4. Data syncs with the Backend Server when internet is available.
5. The backend stores info in the Database and manages updates.
6. Notification System sends alerts or reminders to users.

# **5.PROJECT PLANNING & SCHEDULING:**

* 1. **Project Planning:**

**Week 1–2:**

* Define goals and gather poultry disease data
* Clean and label datasets

**Week 3–4:**

* Develop and fine-tune transfer learning model
* Evaluate model accuracy

**Week 4–6:**

* Build mobile app UI and integrate ML model
* Set up backend and database

**Week 7:**

* Test app and model with users
* Collect and apply feedback

**Week 8:**

* Deploy app and provide user training
* Plan ongoing support and updates

**6.FUNCTIONAL AND PERFORMANCE TESTING:**

**6.1 performance testing:**

**Performance Testing – Poultry Disease Classification App**

**🔹 1. App Performance Testing**

| **Test Area** | **Goal** | **Expected Outcome** |
| --- | --- | --- |
| App Launch Time | Ensure quick access | App starts within 2–3 seconds |
| Input Response | User inputs (symptoms/data) processed smoothly | Inputs registered without lag |
| Offline Function | App works in areas with poor/no internet | Diagnosis and recommendations still functional |
| Device Load | Run app on low-end devices | No crashes; memory/battery usage within limits |

**🔹 2. ML Model Performance Testing**

| **Test Area** | **Goal** | **Expected Outcome** |
| --- | --- | --- |
| Inference Speed | Disease prediction time after symptom input | < 2 seconds per prediction |
| Model Accuracy | Evaluate performance on test data | > 90% overall accuracy |
| Precision/Recall | For each disease class | Balanced scores (ideally > 85%) |
| Model Size | Suitable for mobile deployment | < 30MB (TensorFlow Lite or ONNX format) |

**🔹 3. Backend/Database Performance (Optional/If Cloud-Connected)**

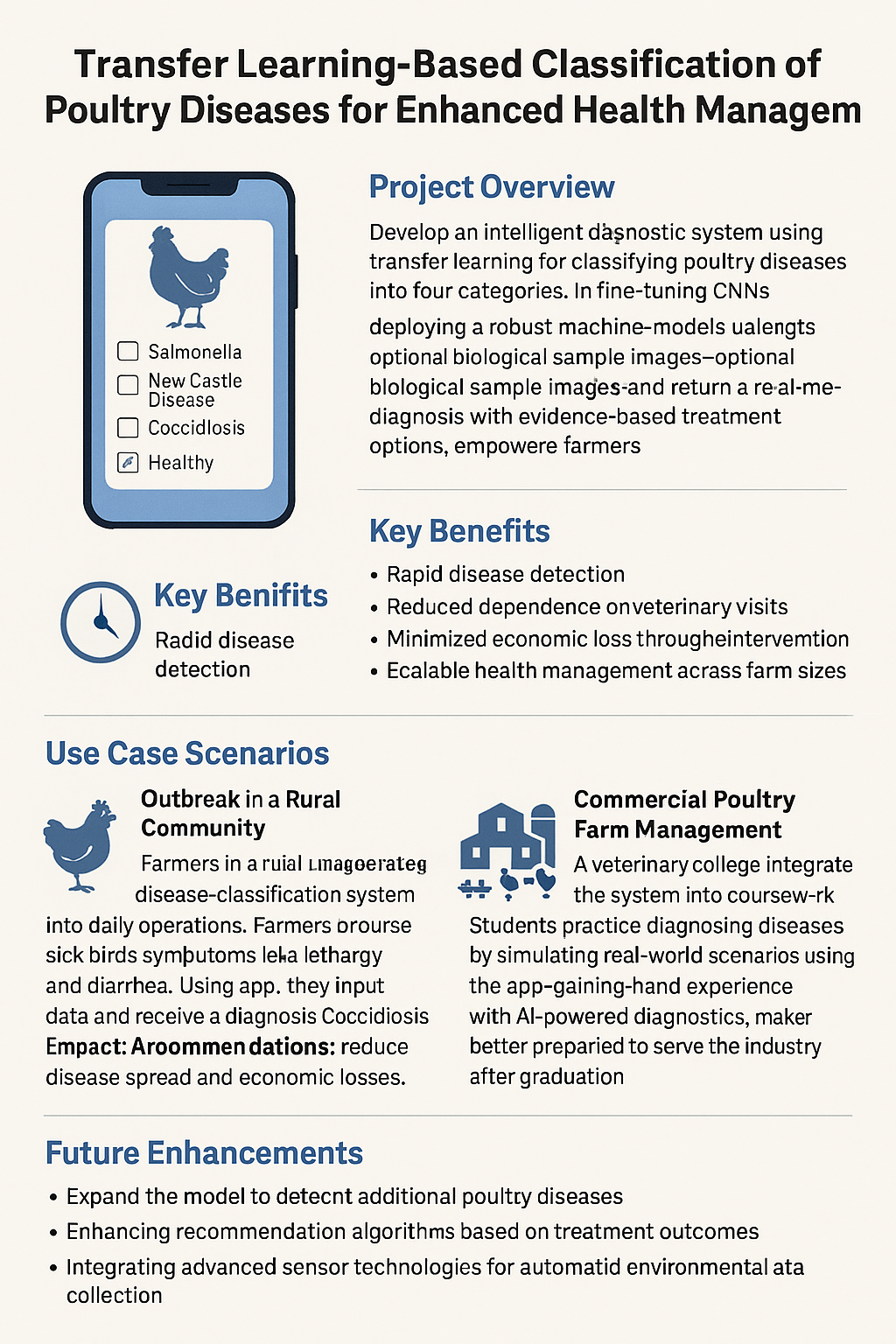
| **Test Area** | **Goal** | **Expected Outcome** |
| --- | --- | --- |
| API Speed | Fast diagnosis upload and sync | < 300ms response time |
| Concurrent Load | Handle multiple users at once | Stable with 100–500 users |
| Data Sync | Offline → Online transition | No data loss or delay in updates |

**✅ Tools You Can Use:**

* Firebase Test Lab – for mobile performance
* TensorBoard / TFLite Benchmark Tool – for ML model profiling
* Postman / JMeter – for backend and API load testing

**7.RESULTS:**

**Result screenshot:**

**8.ADVANTAGES AND DISADVANTAGES:**

**Advantages:**

1. **Fast and Accurate Diagnosis**
   * Leverages pre-trained models to quickly classify diseases with high accuracy, even on limited datasets.
2. **Cost-Effective for Farmers**
   * Reduces the need for frequent veterinary visits, saving time and money.
3. **Accessible via Mobile Application**
   * Can be used remotely by farmers in rural or underserved areas without requiring advanced equipment.
4. **Early Detection and Prevention**
   * Identifies disease symptoms early, enabling timely intervention and limiting outbreaks.
5. **Scalable and Adaptable**
   * Can be expanded to include more diseases or integrated with IoT/sensor data for real-time monitoring.
6. **Educational Utility**
   * Useful as a training tool for veterinary students and professionals.

**Disadvantages:**

1. **Dependence on Quality Data Input**
   * Inaccurate or incomplete symptom input by users can lead to misclassification.
2. **Limited by Initial Disease Categories**
   * Only detects predefined diseases (e.g., Salmonella, New Castle, Coccidiosis); cannot identify unknown or rare conditions without retraining.
3. **Requires Initial Dataset for Training**
   * Needs labeled images/symptoms for transfer learning, which may be hard to obtain in some regions.
4. **Technology Access Barrier**
   * May be less accessible to farmers with limited digital literacy or smartphone access.
5. **Environmental and Biological Variability**
   * Accuracy may be affected by different breeds, climates, or symptom presentation across regions**.**
6. **Not a Substitute for Expert Diagnosis**
   * AI predictions should be used to support, not replace, professional veterinary assessments in complex cases.

**9.CONCLUSION:**

The development of a transfer learning-based poultry disease classification system offers a transformative solution for enhancing poultry health management. By combining advanced machine learning techniques with accessible mobile technology, this system empowers farmers, educators, and veterinarians to detect diseases early, respond rapidly, and improve overall flock productivity.

While there are limitations—such as dependency on accurate data input and technological access—the benefits of cost savings, early intervention, and scalability far outweigh the challenges. With continued refinement and expansion, this system has the potential to revolutionize livestock care, especially in rural and underserved areas, contributing significantly to food security, economic resilience, and animal welfare in the poultry industry.

**10.FUTURE SCOPE:**

The proposed transfer learning-based poultry disease classification system holds significant potential for future development and impact. Key areas for expansion and enhancement include:

**1. Expansion of Disease Categories**

* Integrate more poultry diseases beyond the initial four (e.g., Avian Influenza, Infectious Bronchitis).
* Build a more comprehensive diagnostic tool for broader application across poultry types and regions.

**2. Integration with IoT and Smart Sensors**

* Combine real-time environmental monitoring (e.g., temperature, humidity, ammonia levels) using IoT sensors.
* Enable predictive analysis for disease outbreaks based on environmental and behavioral patterns.

**3. Multilingual and Voice-Based Interfaces**

* Incorporate local language support and voice input features to improve usability for farmers with limited literacy.

**4. Continuous Learning and Model Updates**

* Use data collected from app users to periodically retrain and improve the model, adapting to new disease strains and regional variations**.**

**5. Offline Functionality**

* Implement offline diagnosis capability for areas with poor or no internet connectivity.

**6. Integration with Veterinary Networks**

* Create a referral system within the app for farmers to connect with nearby veterinary professionals when high-risk conditions are detected.

**7. Extension to Other Livestock**

* Adapt the platform for other livestock species such as cattle, goats, and swine, creating a universal animal health management tool.

**8. Research Collaboration and Data Sharing**

* Partner with academic institutions, research centers, and government bodies to validate and refine the model with real-world data.

**11.APPENDIX:**

**GITHUB LINK:**  https://github.com/Manju173214/POULTRY-DISEASE-MANAGEMENT/blob/main/copy\_of\_poultry\_disease\_classification\_model\_compared%20(1).py

**DEMO VEDIO LINK:** **https://youtu.be/bfw0Azv4jCA?feature=shared**