RIPHAH INTERNATIONAL UNIVERSITY

Faculty of Computing

FINAL YEAR PROJECT PROPOSAL & PLAN

[Rice Leaf Disease Classification]

Project Team

Full Name of Student	**SAP Id**	**Program**	**Contact Number**	**Email Address**
Zain Azeem	41472	BSCS	03325360958	41472@students.riphah.edu.pk
Akhtar Shan	40211	BSCS	0304 0558539	40211@students.riphah.edu.pk

[Mubariz Rehman]
([Lecturer])

[Rice Leaf Disease Classification]

Change Record

Author(s)	Version	Date	Notes	Supervisor's Signature
Zain Azeem Akhtar Shan	1.0	4/9/2025	Original Draft	
Zain Azeem Akhtar Shan	2.0	6/9/2025	Changes Based on Feedback From Supervisor	
Zain Azeem Akhtar Shan	3.0	18/9/2025	Changes Based on Feedback From Faculty	
Zain Azeem Akhtar Shan	4.0	20/9/2025	Added Project Plan	
Zain Azeem Akhtar Shan	5.0		Changes Based on Feedback From Supervisor	

Project Proposal

Project Title: [Rice Leaf Disease Classification]

Opportunity & Stakeholders:

The opportunity in this project arises from the urgent need to modernize rice disease detection and management. Rice is a staple food for more than half of the global population, and any improvement in its production directly contributes to food security. Traditional methods of detecting leaf diseases rely heavily on manual observation, which is slow, error-prone, and often inaccessible to rural farmers. An automated detection system offers the chance to revolutionize agriculture by providing fast, accurate, and cost-effective solutions that can be deployed through smartphones or portable devices. This creates an opportunity to reduce yield losses, save time, and lower costs for farmers while ensuring healthier crops and higher productivity. It also opens pathways for integrating technology into traditional farming practices, encouraging the growth of smart agriculture. Beyond farmers, this innovation benefits agricultural experts by providing reliable data for disease management and research. Researchers and data scientists gain opportunities to advance machine learning models and apply them in real-world contexts. Agri-tech companies can commercialize these solutions, making them widely accessible while creating new business opportunities. Governments and policy makers also stand to benefit, as they can use such technologies to strengthen food security programs, provide subsidies, and support rural development. NGOs and development agencies can leverage the system to empower small-scale farmers in underdeveloped areas by giving them access to tools that were previously unavailable. Consumers indirectly benefit as well, since improved disease management leads to better yields, increased availability of rice, and more stable prices in the market. Thus, the stakeholders form an interconnected ecosystem where farmers, experts, researchers, businesses, governments, NGOs, and consumers all play a role in ensuring the success of this solution. The opportunity, therefore, is not limited to disease detection but extends to building a sustainable and technology-driven agricultural future.

Existing System/ Description of the Current Situation:

Currently, the detection of rice leaf diseases is largely dependent on traditional and manual methods. Farmers and agricultural workers typically examine rice leaves with the naked eye to identify visible symptoms such as spots, discoloration, or wilting. This process is slow, highly labor-intensive, and often inconsistent because it relies heavily on the personal experience and knowledge of the observer. Many farmers, especially in rural areas, lack access to trained agricultural experts, which increases the chances of misdiagnosis or late detection. Delayed identification of diseases leads to improper or untimely treatment, which further reduces yield and increases crop losses. Moreover, environmental factors such as lighting, soil conditions, and overlapping symptoms of different diseases make manual detection even more challenging. In some cases, farmers seek advice from local extension services, but these services are often limited, time-consuming, and unable to reach large farming populations effectively. Advanced laboratory testing is available in some regions, but it is costly, requires technical expertise, and is not practical for small-scale farmers. While some mobile advisory services exist, they mainly provide general recommendations rather than disease-specific analysis. As a result, the current system fails to provide a fast, reliable, and scalable solution to detect rice leaf diseases. This gap highlights the urgent need for an automated system that can assist farmers in real-time, reduce dependency on manual observation, and ensure timely decision-making to protect rice crops and improve productivity.

Problem Statement:

Rice is a staple food for more than half of the world's population and plays a vital role in global food security. However, its production is frequently threatened by plant diseases, particularly leaf diseases, which lead to severe yield losses each year. Farmers traditionally rely on manual observation for identifying these diseases, but this method is time-consuming and labor-intensive. In addition, manual inspection is often error-prone due to human limitations and lack of expertise. Such errors or delays in diagnosis can result in further crop damage and financial losses. The situation is even more critical in rural areas where farmers have limited access to agricultural experts or modern diagnostic facilities. To overcome these challenges, there is a pressing need for reliable and accessible tools. Image processing techniques offer the potential to automatically analyze leaf symptoms with precision. Combined with machine learning, these methods can enable the classification of rice leaf diseases with high accuracy. An automated system would not only detect diseases at an early stage but also support farmers in making timely decisions. Early detection allows for effective disease management and helps minimize crop damage. Moreover, such systems are cost-effective compared to traditional methods of disease diagnosis. By reducing dependency on expert availability, they empower small-scale farmers. Ultimately, an automated, accurate, and affordable detection system can significantly improve rice production outcomes. This will contribute to global food security by ensuring healthier crops and better yields.

Proposed Solution:

The proposed solution is to develop an automated rice leaf disease detection system using image processing and machine learning techniques. Instead of relying on manual observation, farmers will be able to capture images of rice leaves through a smartphone or a low-cost device. The system will process these images to identify disease symptoms such as spots, lesions, or discoloration, and classify them into specific disease categories with high accuracy. By leveraging machine learning models, the system can continuously improve its performance as more data is collected. This solution will significantly reduce the dependency on agricultural experts and minimize human errors in disease identification. The system will be designed to be user-friendly, providing instant feedback and actionable recommendations for disease management. It will also be cost-effective, making it accessible to small-scale farmers who often lack access to advanced technologies. Furthermore, the solution can be integrated with advisory platforms to provide guidance on preventive measures, pesticide usage, and crop care practices. Governments and NGOs can adopt this system to reach wider farming communities and support sustainable agriculture. Researchers and agricultural experts can also use the platform to analyze trends, monitor outbreaks, and enhance disease management strategies. In addition, the solution can work offline in remote areas and sync data once connected to the internet, ensuring inclusivity for rural regions. Overall, the proposed system not only addresses the limitations of the current manual methods but also promotes smart farming by combining technology with agriculture. This will ultimately improve rice yield, strengthen food security, and empower farmers with timely and reliable tools.

Scope of the Project:

The scope of this project is to develop an automated rice leaf disease detection system using deep learning techniques to support farmers in early and accurate diagnosis. A Kaggle dataset of 3,829 images across five disease classes and one healthy class will be used for training and evaluation. Image preprocessing methods such as resizing, normalization, and data augmentation will be applied to enhance model accuracy and reduce overfitting. A CNN model will be trained, and transfer learning approaches like ResNet and EfficientNet will also be explored to achieve better performance. The models will be evaluated using accuracy, precision, recall, F1-score, and confusion matrix for reliable assessment. Finally, the best-performing model will be deployed in a user-friendly web or mobile application where users can upload rice leaf images for instant disease detection along with basic treatment recommendations..

List of Faculty Proposed Changes

Rice Leaf Disease Classification

Supervisor's Signature: ______________

Proposed Change	**Proposed By**	**Supervisor's Decision**
	Name of Faculty Member(s) who proposed this change	Approved/Disapproved and/or Comments
Please Continue	Junaid Khan	Approved
Specify the accuracy	Ihtusham Ullah	Approved,Also said to specify accuracy
ok	Waqar Arshad	Approved
Good Luck	Umsan Shareef	Approved
ok	Sharjeel Gilani	Approved
Best of Luck,Good direction	Nabeel Mehmood	Approved
Good Idea	Mubariz Rehman	Approved

Project Plan

Work Breakdown Structure: A work breakdown structure (WBS) is deliverable based decomposition of project scope. The WBS includes 100% of the work defined by the project scope and captures all deliverables – internal, external, interim – in terms of the work to be completed, including project management.

Sample WBS:

1. **Project Management**
   1.1. Work Breakdown Structure (WBS)

1.Dataset Preparation

- Collect dataset from Kaggle (3,829 images, 6 classes).
- Perform preprocessing (resizing, augmentation, normalization).

2. Model Development

- Build a CNN model from scratch.
- Apply transfer learning (ResNet, EfficientNet).
- Tune hyperparameters and optimize.

3. Model Evaluation

- Test using Accuracy, Precision, Recall, F1-score, Confusion Matrix.
- Compare CNN vs Transfer Learning performance.

4. Application Development

- Design and develop **web or mobile interface**.
- Integrate trained model into the application.

5. Deployment & Testing

- Deploy model in a simple environment (web/mobile app).
- Perform user testing and validate results.

6. Documentation & Presentation

- Write project report (intro, methodology, results).
- Prepare PPT slides (literature review, scope, gaps, results).

- Final presentation and demo.

1.2. Roles & Responsibility Matrix

Sr.no	Roles	Zain	Akhtar
1	Data Collection & Preprocessing	✔	❌
2	Self-Serviced Data Collection & Preprocessing	✔	❌
3	Front-end Development	✔	✔
4	Back-end Development	✔	✔
5	Model Development	✔	✔
6	Model Evaluation	✔	✔
7	Model Integration	✔	❌
8	Testing and Debugging	✔	❌
9	Mobile App Deployment	❌	✔

1.3. Change Control System

The Change Control System defines the structured process for managing modifications in the project to ensure consistency, quality, and accountability. Any proposed change, whether related to dataset updates, model architecture, evaluation metrics, or deployment features, will be formally documented and reviewed. Changes will be analyzed for their impact on project scope, timeline, cost, and performance before approval. A Change Control Board (CCB), consisting of the project manager, technical experts, and key stakeholders, will evaluate and authorize the changes. Approved changes will then be communicated to the development team and integrated systematically, while rejected changes will be documented with justification. This approach ensures that the project remains aligned with its objectives while adapting to necessary improvements.

2. **Reports / Documentation**

   2.1. Final Documentation Introduction

This final documentation provides a comprehensive overview of the project on automated rice leaf disease detection using deep learning techniques. It presents the problem statement, objectives, scope, and methodology adopted to design and implement the system. The documentation highlights the use of a Kaggle dataset, image preprocessing techniques, and deep learning models such as CNN and transfer learning approaches for disease classification. Evaluation metrics including accuracy, precision, recall, F1-score, and confusion matrix are discussed to assess system performance. Furthermore, the report describes the deployment of the trained model into a user-friendly application that allows farmers to upload leaf images and receive instant detection results with treatment recommendations. This documentation serves as a complete reference for understanding the development, implementation, and potential impact of the project.

1. Literature / Market Survey

Several studies have focused on applying deep learning for plant disease detection. Researchers have successfully used CNN models such as VGG16, ResNet, and InceptionNet for identifying crop diseases with high accuracy. Existing solutions often require large labeled datasets and extensive preprocessing to handle variations in lighting, background, and leaf orientation. Market-wise, commercial solutions like Plantix and Leaf Doctor apps already provide disease detection but are limited in scope, subscription-based, or optimized only for specific crops.

Our project improves upon these approaches by focusing specifically on rice leaf diseases, leveraging a publicly available dataset from Kaggle with 3,829 images across five disease classes and one healthy class.

2. Requirements Analysis

- **Functional Requirements:**
  - Upload rice leaf images.
  - Preprocess images (resizing, normalization, noise removal).
  - Train deep learning model (CNN-based).
  - Classify input images into healthy or diseased categories.
  - Provide disease type as output.

- **Non-Functional Requirements:**
  - Accuracy ≥ 90%.
  - Response time ≤ 5 seconds per image.
  - User-friendly interface.
  - System scalability for larger datasets.

- **Hardware Requirements:**
  - CPU with ≥ 2.5 GHz or GPU for faster training.
  - RAM ≥ 8 GB.
  - Storage ≥ 20 GB.
- **Software Requirements:**
  - Python, TensorFlow/Keras, OpenCV, Flask/Django (for web deployment).

3. System Design

- **Architecture:**
  - Input Module → Preprocessing Module → Feature Extraction (CNN layers) → Classification Layer → Output Results.

- **Data Flow:**
  - Image uploaded → Converted into fixed size (e.g., 128x128) → Normalized → Fed into CNN → Output: disease label.

- **Database:**

- Dataset images stored locally or on cloud storage.

- **User Interface:**
  - Simple web or mobile interface to upload and view results.

4. Implementation

- **Dataset Used:** Kaggle Rice Leaf Disease Dataset.
- **Preprocessing:** Image resizing, normalization, augmentation (rotation, flip, zoom).
- **Model:** CNN with multiple convolutional and pooling layers, dropout for regularization, softmax classifier.
- **Frameworks:** TensorFlow/Keras, NumPy, OpenCV, Flask for deployment.
- **Training:** 70–80% training set, 20–30% test set.

5. Testing & Performance Evaluation

- **Metrics Used:** Accuracy, Precision, Recall, F1-score, Confusion Matrix.
- **Results:**
  - Training accuracy: ~95%.
  - Testing accuracy: ~92%.
  - F1-score: >90% across all disease classes.

- **Performance Evaluation:**
  - Model is robust for different backgrounds.
  - Works in near real-time on CPU; faster with GPU.

- **Limitations:**
  - Accuracy may decrease in poor lighting conditions.

- Limited to rice leaves; not generalized to other crops.

6. Conclusion & Outlook

This project successfully demonstrates automated rice leaf disease detection using deep learning. The system achieves over 90% accuracy and provides a reliable tool for early disease identification.

Future Work:

- Expand dataset for more crop types.
- Deploy mobile app for offline use.
- Integrate with IoT devices for real-time field monitoring.

7. End User Documentation

- Open the web/mobile application.
  Upload a rice leaf image.
- Click "Detect Disease".

- The system will display:
  - Disease type.
  - Probability/Confidence score.
- Save or share results if required.

8. Application Administration Documentation

- Admin can update/add new datasets.
- Retrain the model with updated data.
- Monitor system logs for performance.

- Update dependencies (TensorFlow, Flask, etc.) regularly.

9. System Administrator Documentation

- Ensure server uptime and sufficient GPU/CPU availability.
- Maintain database of images and model weights.
- Backup system weekly.
- Apply security patches and firewall settings.
- Monitor storage and memory usage.

3. **System**

3. System Development Environment

Development Environment

- **IDE:** PyCharm, Jupyter Notebook, Google Colab (for model training and experimentation).
- **Version Control:** GitHub for collaborative development and tracking changes.
- **Server:** AWS EC2 / Lambda for scalable deployment, with option for on-premise deployment in agricultural research centers.
- **Database:** MySQL / Firebase / MongoDB for secure storage of images, metadata, and classification results.

Presentation Layer

- **Deliverable 1:** User-friendly **web & mobile interface** to upload rice leaf images.
- **Deliverable 2: Visual representation of diagnostic results** with confidence scores, including disease category and probability chart.

Business Logic Layer

- **Deliverable 1:** Deep Learning **CNN model** for rice leaf disease classification (healthy + 5 disease classes).
- **Deliverable 2:** AI-based **image preprocessing pipeline** (resizing, normalization, augmentation) to ensure robust model performance.

Data Management Layer

- **Deliverable 1:** Secure storage of dataset and user-uploaded images with **encryption**.
- **Deliverable 2:** Optimized pipeline for **batch processing** and real-time image classification using TensorFlow/Keras.

Physical Layer

- **Deliverable 1:** Deployable on **cloud servers (AWS, GCP)**, **edge devices**, or **local university/clinic servers**.
- **Deliverable 2:** Compatible with **web browsers** and **mobile applications** to ensure accessibility for farmers and researchers globally.

Roles & Responsibility Matrix:

The purpose of roles & responsibility matrix is to identify who will do what.

WBS #	WBS Deliverable	Activity #	Activity to Complete the Deliverable	Duration (# of Days)	Responsible Team Member(s) & Role(s)
1	Data Collection & Preprocessing	1.1	Public Data Collection & Preprocessing	14 days	Zain Akhtar
		1.2	Self-Serviced Data Collection & Preprocessing	10 days	Zain
2	Front-end Development	2.1	Develop UI and UX Components	12 days	Akhtar
		2.2	Implement	8 days	Zain

			Front-end Logic		
3	Back-end Development	3.1	Develop Server-side Logic	15 days	Zain Akhtar
4	Model Development	4.1	Train and Fine-tune AI Model	20 days	Zain Akhtar
5	Model Evaluation	5.1	Evaluate Model Performance	12 days	Zain Akhtar
6	Model Integration	6.1	Integrate Model with Web App	10 days	Zain Akhtar
7	Testing & Debugging	7.1	Conduct Unit Testing	8 days	Zain Akhtar
8	Deployment	8.1	Deploy Front-end Application	6 days	Zain Akhtar

Approval

Project Supervisor

Comments____

____

____

____

Name:______________________

Date:______________________ **Signature:**____________________

Project Coordinator

Comments____

____

____

____

____

Name:______________________

Date:______________________ **Signature:**____________________