Week 1 Report — CI/CD Setup for Whitefly Detection System

Overall Objective

To establish a reliable version control and continuous integration (CI) environment for the Whitefly Detection System, enabling automated building and testing of the project upon every code update.

Repository Setup

The project source code was initially developed locally and later linked to a new GitHub repository for collaborative version control and CI/CD integration.

- **Repository URL:** https://github.com/emer-tusiime/CICD-Whitefly
- **Project Type:** Full-stack web application (React Frontend + Python Backend)
- **Version Control:** Git (pushed from local to remote)

Project Overview

Our project, **White Fly Detection System**, aims to detect and classify white flies on crop leaves using a trained **YOLOv8 object detection model**. The system integrates a **React frontend** for user interaction and a **Django backend** for model inference and data management. The CI/CD pipeline is being implemented to ensure automated testing, building, and deployment of both components.

Objectives

- Select and set up a full-stack web application (React + Django).
- Initialize version control using Git and GitHub.
- Establish task management via ClickUp.

Summary of Activities

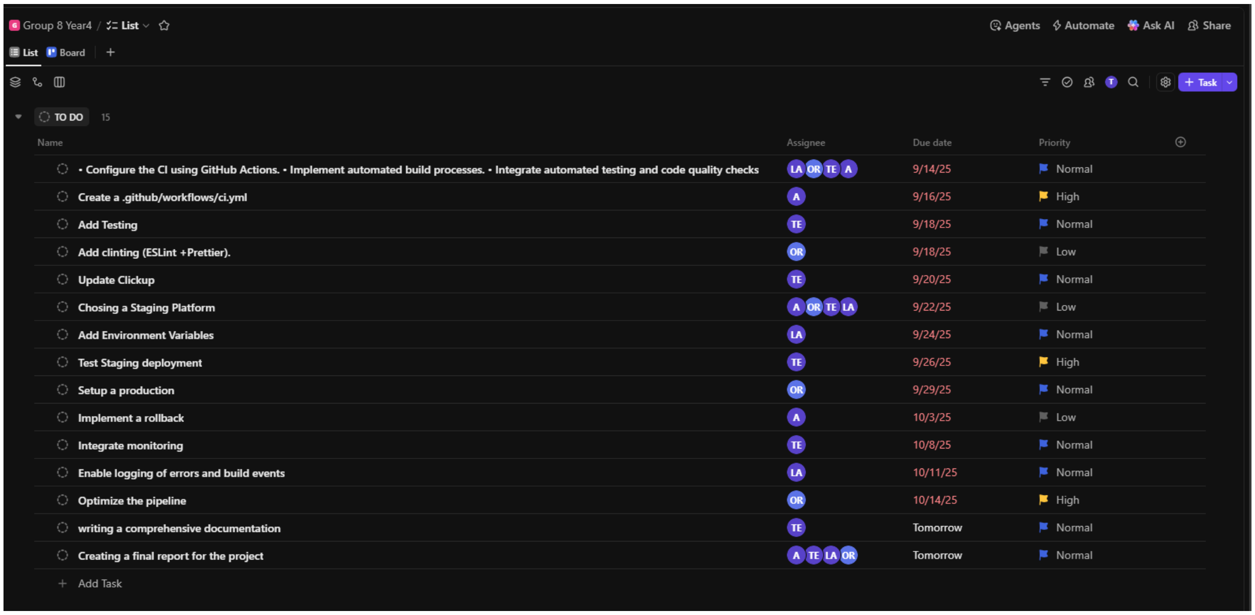
1. **Application Selection and Setup**
   The team decided to implement a White Fly Detection system. The backend (Django) exposes REST endpoints for image submission and inference using a pre-trained YOLOv5 model. The frontend (React) enables users to upload images and view detection results.
2. **Version Control Configuration**
   - A GitHub repository named `WhiteFlyDetection_Group_BSE24-X` was created.
   - The repository was initialized with a `README.md`, `.gitignore`, and folder structure (`frontend/`, `backend/`, `model/`).
   - Team members were added as collaborators for version control contributions.
3. **ClickUp Setup**
   - A ClickUp workspace was created for the group.

- Tasks for Week 1 (repository setup, project initialization, team role assignments) were created and assigned to:
  - **Tusiime Emmanuel** – Repository setup & ClickUp management
  - **Aine Levi (LA)** – Git configuration frontend initialization
  - **Ssentongo Henry** – Backend initialization (Django)
  - **Otim Ronald (OR)** –Documentation support and testing setup planning

Deliverables

- GitHub repository with base project structure
- `README.md` with project description
- ClickUp workspace configured with task assignments

Screenshots



Name	Assignee	Due date	Priority
• Configure the CI using GitHub Actions. • Implement automated build processes. • Integrate automated testing and code quality checks	LA OR TE A	9/14/25	Normal
Create a .github/workflows/ci.yml	A	9/16/25	High
Add Testing	TE	9/18/25	Normal
Add clinting (ESLint +Prettier).	OR	9/18/25	Low
Update Clickup	TE	9/20/25	Normal
Chosing a Staging Platform	A OR TE LA	9/22/25	Low
Add Environment Variables	LA	9/24/25	Normal
Test Staging deployment	TE	9/26/25	High
Setup a production	OR	9/29/25	Normal
Implement a rollback	A	10/3/25	Low
Integrate monitoring	TE	10/8/25	Normal
Enable logging of errors and build events	LA	10/11/25	Normal
Optimize the pipeline	OR	10/14/25	High
writing a comprehensive documentation	TE	Tomorrow	Normal
Creating a final report for the project	A TE LA OR	Tomorrow	Normal

+ Add Task

Figure 1 Shows Clickup Todo list and tasks assigned to each member

Challenges & Future Plans

Initial challenges included aligning environments across team members and managing Django dependencies on different systems. Plans for Week 2 include integrating GitHub Actions for automated builds and testing.