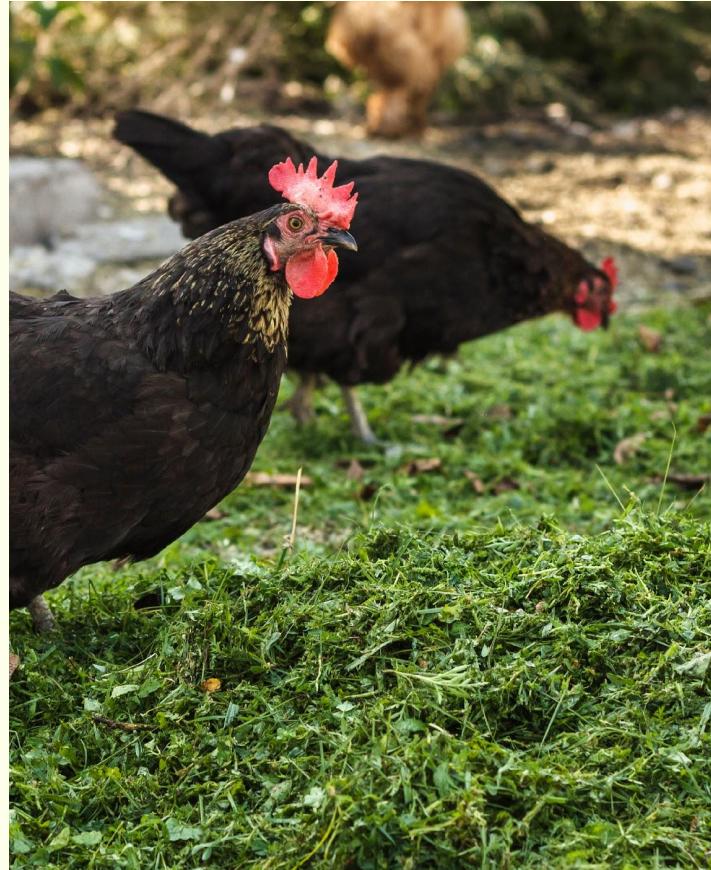


# A Streamlined Livestock Management Application



# Table of contents

## Problem Statement

A look into why I created this application.

## Key Benefits

Key benefits the solution provides and real world use cases.

## Next steps

Planned additions and new features I would like to add

01

02

03

04

05

06

## Solution Overview

Brief recap of what the solution entails.

## Roadmap

Development Cycle of application

## Conclusions

Reflections on the project and what I would do differently.

O1

# Problem Statement

What was the issue?





# Problem Statement

The client, a **local chicken breeder**, currently manages their flock using a **paper-based system**, which is:

- Inefficient
- Error-prone
- Difficult to update.

They want to explore moving to a digital solution that is simple, reliable, and works on **Windows or macOS**.



# Client Requirements

- Develop a **CLI-based Proof of Concept (PoC)** app
- Must:
  - Run on **Windows/macOS**.
  - Use a **list** to store chicken names (Initially).
  - Support **basic operations**: Create, Read, Update and Delete (CRUD).
  - **Clear and print** to the terminal screen
  - Accept **user input** via a **menu system**

This PoC will help the client assess the benefits of switching to a **local database system** in the future.

02

# Solution Overview

How did I resolve the issue?



# Solution Overview



We have developed a **Command Line Interface (CLI)** application as a **Proof of Concept (PoC)** to demonstrate how the client can transition from their paper-based system to a **digital, menu-driven solution**.

## Key Features:

- **Simple CLI Menu:** Easy-to-use terminal interface for non-technical users.
- **Chicken Record Management:** Add, view, remove, and rename chicken names using a basic Python list (**CRUD**).
- **User Interaction:** Accepts real-time input and provides instant feedback.
- **Screen Control:** Clear and print to the screen for a clean and readable experience.

## Designed For:

- **Windows and macOS Compatibility:** Lightweight, portable, and runs in standard terminal environments.
- **Future Scalability:** Easily extendable to include more data fields, local file storage, or a database system.

03

# Key Benefits

Key benefits the solution provides and real world use cases.



# Key Benefits



## Time Savings

Faster record updates



## Data Accuracy

Validations prevent bad entries



## Low Overhead

No server setup,  
minimal footprint



## Scalable

Modular design  
allows easy addition  
of features

# Real-world use cases



## Small-Scale Poultry Farms

**Use Case:** Track individual chickens by name or ID, log ages, breeds, egg production, and health status.

**Benefit:** Replaces paper logs with a simple terminal-based tool, improving organization and reducing human error.

## Backyard or Hobby Breeders

**Use Case:** Maintain a digital flock list to record pairings, hatching dates, and bloodlines.

**Benefit:** Easy access to lineage and breeding cycles without investing in complex farm management software.

## Breeding and Genetic Tracking

**Use Case:** Maintain accurate genetic records and track pairings for selective breeding.

**Benefit:** Supports traceability and smarter breeding choices using historical data stored digitally.

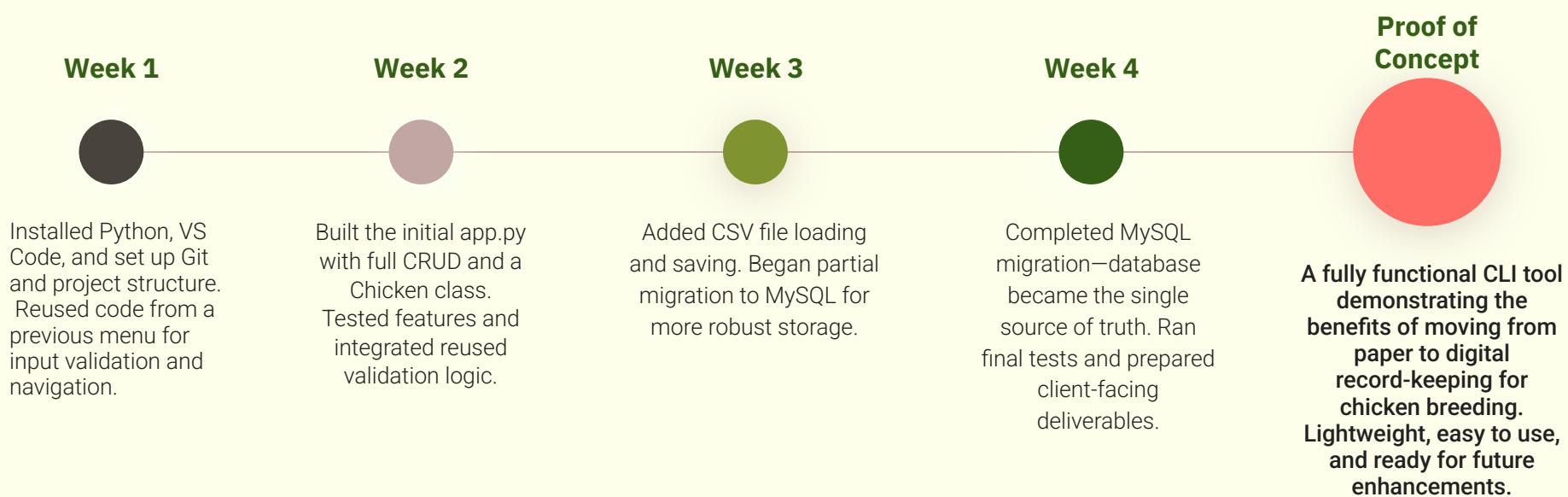
04

# Development Roadmap

Key benefits the solution provides and  
real world use cases.



# Project roadmap



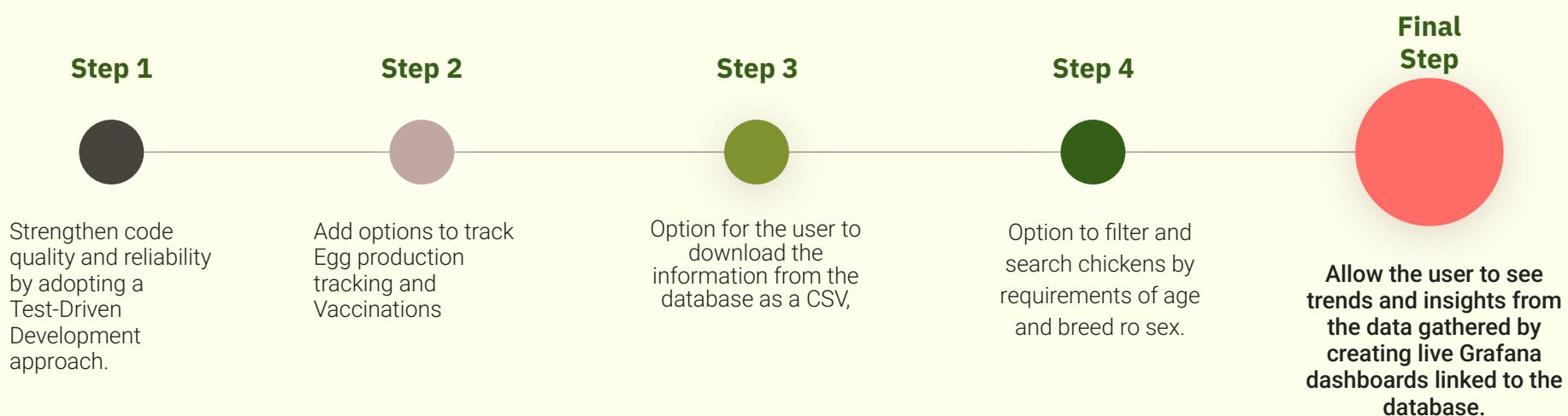
# 05

# Next Steps

Planned additions and new features I  
would like to add



# Next steps



06

# Conclusion

Reflections on the project and what I would do differently.



# Close and retrospective

## What Went Well:

- Delivered a fully functional CLI PoC with database integration.
- Reused and adapted previous code to accelerate development.
- Gained hands-on experience with file and database persistence.

## What I'd Do Differently:

- **Implement Test-Driven Development (TDD)** from the start to catch bugs early and improve code reliability.
- Plan for **data persistence earlier** to avoid late-stage migration challenges.
- Write **more modular and reusable functions** to simplify testing and future scaling.



# Thanks!

Does anyone have any questions?

CREDITS: This presentation template was created by [Slidesgo](#), including icons by [Flaticon](#), infographics & images by [Freepik](#)

