Design Overview

A diagram of a computer

Description automatically generated

Frontend (Tkinter GUI):

Display an image of a hen along with jumbled words.

Provide two shaded boxes for nouns and adjectives.

Implement drag-and-drop functionality to move words into the correct boxes.

Provide 6 level buttons, allowing access to a level only after completing the previous one.

Login system to track user progress.

Display score/level completion status.

Backend (SQLite for persistent storage, JSON for easy state management):

Store user progress (which levels are completed).

Store list of words per level with their correct category.

Load game progress and update on completion.

✅ Drag & Drop: Implemented using bind events in Tkinter (to be added).

✅ Persistent Progress: Tracked in SQLite.

✅ Word Data Storage: Stored in JSON (words.json).

✅ Scalability: More words and levels can be added easily.

✅ Modular Design: Separation of frontend and backend logic.

Explanation:

Drag & Drop Setup: Each word button is bound to start\_drag(), on\_drag(), and on\_drop().

Movement Calculation: on\_drag() calculates the new position dynamically.

Drop Detection: on\_drop() checks whether the word is inside the correct box.

Highlighting Correct Drops: Colors change when a word is dropped in the right category.

Key Features in main.py

✅ User Login & Progress Storage

A simple login screen allows users to enter their username.

Progress is fetched from SQLite database via database.py.

✅ Level Selection System

The user cannot access higher levels until the previous one is completed.

Level buttons are dynamically enabled/disabled based on user progress.

✅ Drag & Drop Integration

Words can be dragged and dropped into adjective/noun boxes.

Uses DragDrop from drag\_drop.py for managing movements.

✅ Word Classification Logic

Words are categorized as adjectives or nouns.

If the words are placed correctly, progress is saved.

Incorrect answers prompt a retry message.

✅ User-Friendly UI

Background colors and font styles improve readability.

The game dynamically adjusts based on level selection.

Futurescope

Animations when dragging and dropping? 🎭

Sound effects on correct/incorrect answers? 🎵

Leaderboard to track high scores? 🏆