**BACKEND DOCUMENTATION**

For the Livestock Management Software

What is this software for?

This software is for goat herders to manage their finances and livestock of goats using this software. It is built for those who want to manage their finances relating to the livestock they own all at one place with an intuitive UI and a fast response time.

Tools used to build this software:

Python – Used for both frontend and backend.

SQLite – For Database.

**DOCUMENTATION**

What is this documentation for?

To hopefully help anyone reading this code to understand the mess of a code that I have written and for myself in the future if I ever return to update the software.

What happens when the software is started for the first time?

Once the software starts, and the backend script is called upon, a database file is created in the working directory if the database file is not already present and initializes all the tables required.

The tables initialized are: WRITE ABOUT THE COLUMN NAMES AND EXPLAIN THEM TODO

* MasterTable – To hold all the basic information about the goats. This will be queried in several functions to retrieve the information about the selected goat and output the desired result.
* WeightTable – To hold all the weight changes about a particular goat.
* KidsIdTable – To hold all the data about a particular goat.
* LivestockNetworth – To hold all the financial information about the breed of a goat.
* Labour – To hold all the financial information about the laborers employed.
* Feed – To hold all the financial information about the amount spent on feed for the animals.
* HealthExpense – To hold all the information about the amount spent on maintaining the health of the goat(s).
* Misc – To hold all the financial information about Miscellaneous accidental information.

What happens next?

A connection to the database is established (variable name *c*) using python’s inbuilt library – sqlite3. A cursor variable with the name *conn* is also established.

What is the weird *if* statement that checks for the first of April?

That condition runs when it is the beginning of a new financial year. It erases all the data in the tables Labour, Feed, HealthExpense, Misc, LiveStockNetworth and all the dead or sold goats from the MasterTable.

Following that, the LivestockNetworth table is updated using the *updateLivestockNetworth* function.

What is the *updateLivestockNetworth* function?

It takes in one argument named *values*. It selects all the breed and category from the LivestockNetworth table.

According to the category(age + gender) of the goat, the LivestockNetworth table is updated. If a category or breed is not already present, it is created as a new record in the table, else the existing values are updated.

This function is called at the beginning of the financial year(1st of April) to keep all the tables in sync.

***CLASS DATABASE****:*

\_\_init\_\_ - Initializer method

* No purpose for the initializer.
* Input – No args
* Output – None

getGoatRecords – Class method

* Returns all the records present in the MasterTable.
* Input – No args
* Output – List of list of records

getNumberOfRecords – Class method

* Returns the number of records present in the MasterTable.
* Input – No args
* Output – int

getColumnNames – Class method

* Returns the name of the columns present in the MasterTable.
* Input – No args
* Output – List of str

insertGoatRecord – Class method

* If no mother ID is present for the goat, (i.e.) the goat was bought or it’s mother is unknown, it gets inserted as a new record with all the values provided to it.
  + Inserts into the columns – goat\_no, breed, date\_of\_birth, gender, pregnant, weight, v1, v2, v3, v4, v5, v6. The vaccination columns are set to the system’s date and time for ease of calculation for vaccination dates.
* Else
  + In the MasterTable a new record is created and these specific columns are written into – goat\_no, breed, date\_of\_birth, gender, pregnant, weight.
  + In the KidsTable a new record is created and the following columns are written into – mother\_id, kid\_id, gender.
  + The number of kids in the MasterTable are updated for the mother\_id provided.
* The mortality of the new record is updated to *Alive*.
* A separate WeightTable for the goat record inserted is created with the name of the WeightTable being initialised as *(WeightTable + goat\_id)*.
* The *LivestockNetworth* method is called to update the LivestockNetworth table.
* Input – Dictionary containing the values TODO
* Output – Consoles the string “Inserted successfully”