**Ration Balancing**



**System Documentation and Final Report**

**Team Name: Ignite**

**Prepared By:**

**Satyavrath Injamuri S533983**

**Venkata Sai Karthik Reddy Bijjam S533488**

**Lakshmi Seshu Kalvakuri S533707**

**Hyndavi Musipatla S533626**

**Sai Varun Reddy Anugu S534089**

**Sanjana Thummala S533577**

**Table of Contents**

|  |  |
| --- | --- |
| 1. **Introduction** |  |
| 1. Problem Statement | 2 |
| 1. Proposed Solution | 2 |
| 1. **Requirement Artifacts** |  |
| 1. Requirements Gathering | 3 |
| 1. Use Cases | 3 |
| 1. Specification Requirements | 5 |
| 1. **Analysis and Design Artifacts** |  |
| 1. Software styles/Architectures | 6 |
| 1. Analysis Model (Database Diagram) | 7 |
| 1. Design Model of the System | 7 |
| 1. Data Model | 8 |
| 1. **Implementation and Testing Artifacts** |  |
| 1. Testing Methodologies and Evaluations | 9 |
| 1. Technologies and Tools Used | 9 |
| 1. **Deployment Artifacts for Users and Clients** |  |
| 1. User’s Guide for users and clients | 12 |
| 1. Installation Guide | 18 |
| 1. Execution Guide | 18 |
| 1. Project URL | 19 |
| 1. **Project Management Artifacts** |  |
| 1. Conclusion | 19 |
| 1. Pros and Cons | 19 |
| 1. Summary of requirements completed | 19 |
| 1. Final Conclusion | 20 |

1. **Introduction**
2. **Problem Statement:**

Agriculture department of Northwest Missouri state university have a problem in getting the nutrient values required in right quantity of ingredients to feed the cattle Where they are already using a solver from excel sheet in which it was difficult for the user to enter all the ingredients and the quantity of each ingredient.

1. **Proposed Solution:**

Our proposed solution is to develop a web application for calculating balanced ration for cattle. This web-based application will help the end user to keep a track of total amount of ingredients feed to the cattle per day along with the nutrition’s consumed. With this application the user can avoid the cumbersome calculations and get the required output with simple steps.

**II. Requirement Artifacts**

**1. Requirements gathering:**

**MoSCoW Rules**

**Must have:**

* Must have the ingredients list and be able to select the ingredients.
* Must be able to select the cattle weight.
* Must be able to set the range(min-max) for nutrients and must be able to edit the nutrients.
* Must be able to take the quantity of ingredients from the user as input.
* Must be able to calculate the ration for the cattle considering weights.
* Calculation of how much each of all the ingredients make up to give between nutrition range.

**Should have:**

* Should have the ability to show the selected ingredients and can select all and deselect all the ingredients at a click.
* Should have a search bar for selecting an ingredient by name.
* Should have the ability to take the nutrient of user choice.

**Could have:**

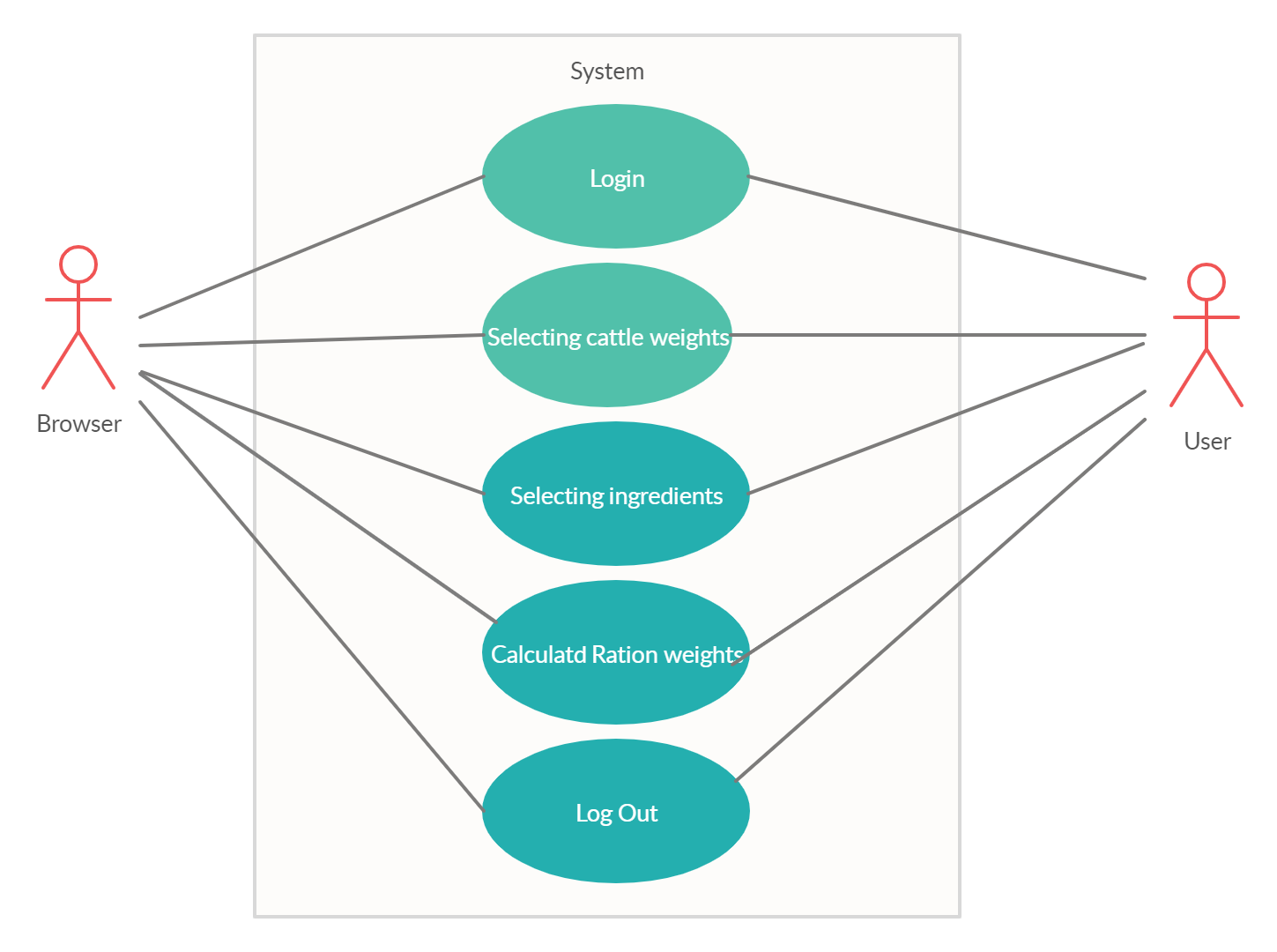
* Could have the ability to set daily weight gain for the cattle growth.
* To display nutrient content for each ingredient.
* Could calculate ingredient quantity in lbs. based on %crude protein required for cattle.
* Could intake the dry matter as the percentage of the body weight.

**Would or won’t have:**

* Would have ability to select ingredients by the system based on cost and availability of stock.
* Would have the ability to store the history of ingredients feed to the cattle.
* Would have the ability to calculate the ration based upon the number of the cattle.

**2. Use Cases:**

### **Use Case Diagram**



**Use Case Scenarios**

1. **Login**

* Primary Actor: Admin/User
* Scope: In Ration Balancing Web Application
* Level: Strategic-level
* Users Story: Admin/User of the Ration Balancing Web Application will use their valid username and password to login to the application. On successful login, the home page of the respective user role i.e. either admin or user is loaded. If the provided username and password are invalid, the system does not log into the application and displays an appropriate error message.

1. **Selecting Cattle Weights/ Cattle Weight Page**

* Primary Actor: User
* Scope: In Ration Balancing Web Application
* Level: User-goal level
* Users : User(s) can log into the system and select the provided weights of the cattle and weight gain of it which are mentioned pounds.
* With the selected cattle weights the min and max values will be generated automatically or can also manually edit the nutrition values by directly entering or adjusting with scroll bar.

1. **Selecting Ingredients/ Feed Stuffs Page**

* Primary Actor: User
* Scope: In Ration Balancing Web Application
* Level: User-goal level
* User: First the page shows the selected cattle weight and selected weight gain.
* Then select the ingredients by alphabetical order
* The selected ingredients will be displayed on the side of the page where user can enter the cost and quantity of the ingredient. Here cost is a mandatory field.
* There is also an option to delete the ingredient.

1. **Calculated Ration Weights/ Ration Weights**

* Primary Actor: User
* Scope: In Ration Balancing Web Application
* Level: User-goal level
* Users : In this page the calculated ration weights will be generated using simplex algorithm written.
* Here, the ration weights are checked if they are in the given range by showing the colors Blue and Red. Where, Blue indicates above the maximum total nutrient range and Red indicates below the minimum total nutrient range.
* In this page, the second table indicates the total AsFed(lbs), DryMatter(lbs), DryMatter Ration(%) of the selected ingredients.

1. **Logout**

* Primary Actor: User
* Scope: In Ration Balancing Web Application
* Users Story: On Click it redirects back to Login Page

**3. Specification requirements**

1. **Platforms:**

* Windows OS
* Linux OS
* Mac OS

1. **Technologies:**

* Node.js
* Express
* Bootstrap

1. **Hosting Strategy:**

* Heroku

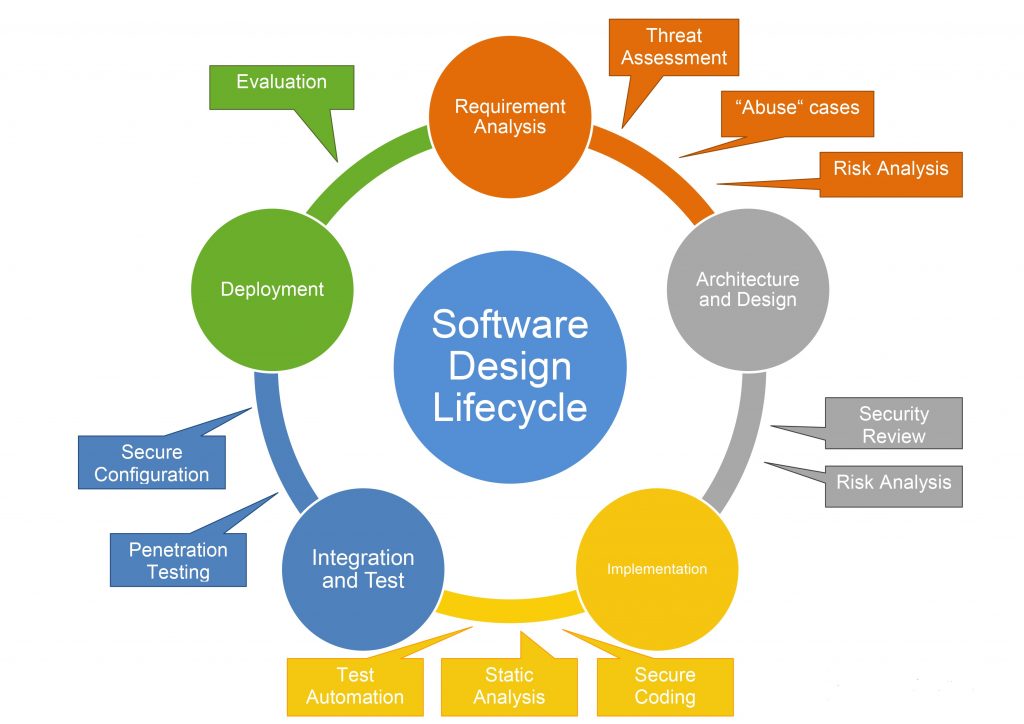
**III. Analysis & Design Artifact**

1. **Software styles/architectures:**

Software architecture is the process of converting the software characteristics like flexibility, reusability, security, scalability into a structured solution that meets the technical and the business architecture.

This definition leads us to ask about the characteristics of a software that can affect a software architecture design. There is long list of characteristics which mainly represents the business or operational requirements, in addition to the technical requirements.

Software characteristics describe the requirements and expectations of a software in technical levels. Documenting software architecture makes easier communication between team and client, able to make decisions early about design and allows reuse of design components between the project.



1. **Analysis model:**

Analysis model is a structural model that is created from the physical model. It is used for analyzing the structural behavior and load bearing, and for design. We are using the flow-based analysis model which is used for transferring the functions, and the Scenario based represents from user point of view.

In the next section report you will find the model diagrams. To store the data, we are using the relational database. To represent the data model, we are using the ER model. ER diagram is shown below.

1. **Design model of the system:**

In implementation and testing the design model is used as essential input. It is an object model describing the achievement of the use cases. It also serves as abstraction for implementation model and its source code. The system software is used to document design. The design model is a composite artifact encompassing all design classes, packages, collaborations, subsystem and relationship between them. Design model primarily sets the architecture. With the use case model and the implementation model the design model continuously kept consistent. A software architect is the one who is responsible for integrity of the design. The functionality described in the use case model, the design model is correct. Including the logical, process, and deployment views the architecture in the design models fulfills its purpose. These views are collected in a separate artifact.

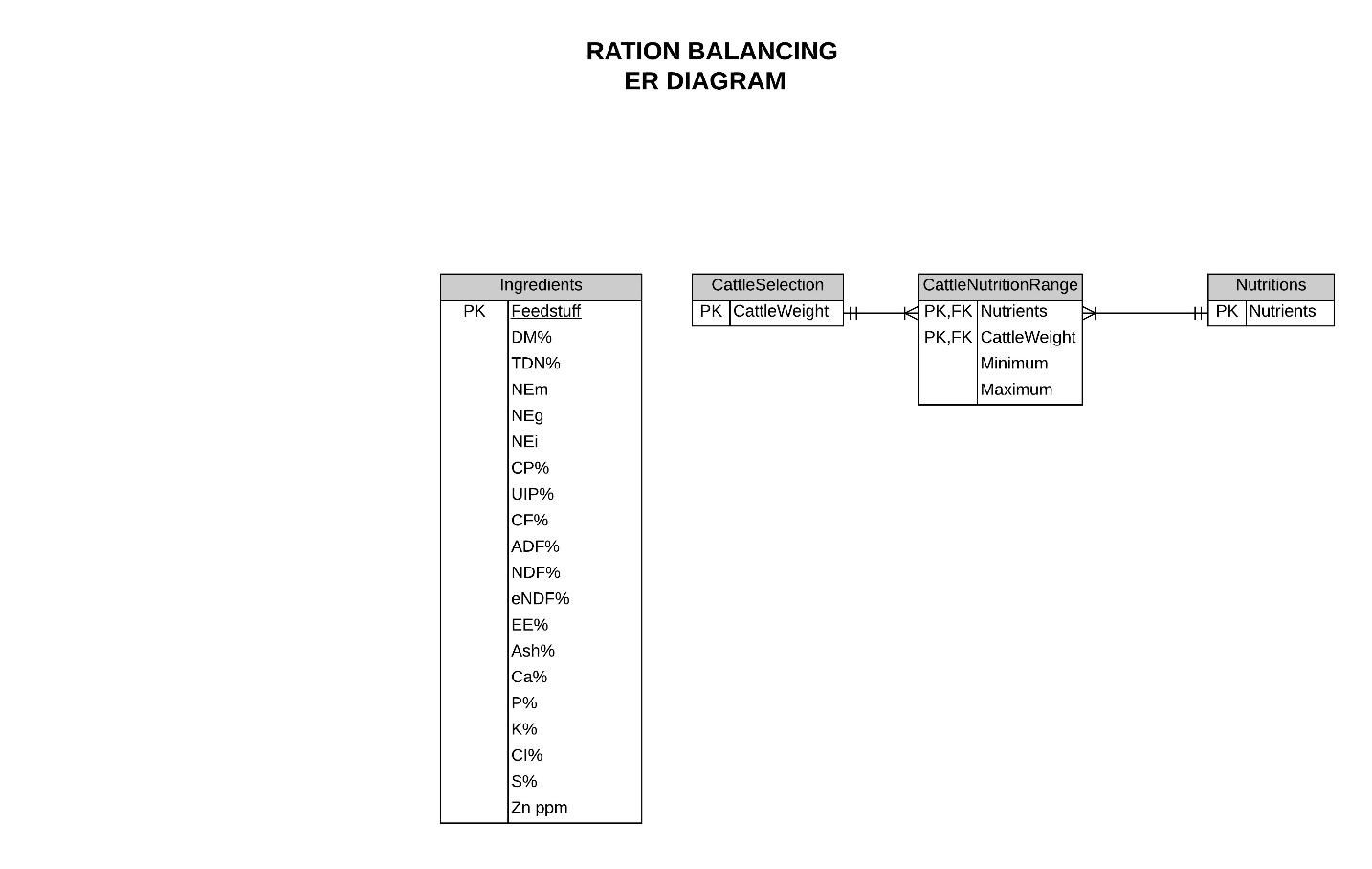
1. **Data model:**

Organizing the elements of the data and standardizes and standardize how they relate to one another is called data model. It defines how data is processed and how the data is connected and stored inside the system. The process of documenting the complex design as an easily understood diagram using the texts and symbols. It is a process documenting a complex design into an easy understood diagram, using symbols and text to represent the way the data flows.

The efficient use of data, as a blueprint for the applications. The data models have been built.

During the analysis and design the phases of the project to ensure requirements data models have been built for a full application are fully understood. They often use the simple data often use the multiple models to view the same data and ensure that all processes, entities, relationships and data flows have been identified. By gathering the requirements from clients, they initiate new projects.

**E-R Diagram:**



**IV. Implementation and Testing Artifacts**

**1. Testing methodologies and evaluations:**

**Unit Testing:**

Unit testing is the first level of testing and is often performed by the developers. It is the process of ensuring individual components of a piece of software at the code level are functional and work as they were designed. Developers in a test-driven environment will typically write and run the tests prior to the software or feature being passed over to the test team. Unit testing can be conducted manually but automating the process will speed up delivery cycles and expand test coverage. Unit testing will also make debugging easier because finding issues earlier means they take less time to fix than if they were discovered later in the testing process.

**Unit Test Cases:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Case ID** | **Module** | **Test Case** | **Steps required** | **Results** |
| **1)** | Login | Login (Incorrect Credentials) | 1.) Open application URL  2.) Click on login button  3.) Provide incorrect username or password  4.) Click on Login | Returned to Login page with error message `Invalid credentials`. |
| Login (Valid details) | 1.) Open application URL  2.) Click on login button  3.) Provide valid username or password  4.) Click on Login | Goes to the cattle weight page. |
| **2)** | Cattle weight Page | Cattle weight | 1.) After login navigates to the cattle weight page.  2.) User can select the weight  3.) User can deselect the weight | The color of the selected weight changes. |
| Cattle weight gain. | 1.) After login navigates to the cattle weight page.  2.) User can select the weight gain from dropdown.  3.) The selected weight gain is displayed at the top. | The selected weight gain displays at the top of the dropdown. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | Min and Max Value. | 1.) After login navigates to the cattle weight page.  2.) User can select the weight gain from dropdown.  3.) The selected weight gain is displayed at the top. | The user should be able to see the default values, if edited the edited values should get updated. |
| **3)** | Feedstuff Page | Selecting Ingredients | 1.) After cattle weight the page navigates to feedstuff page.  2.) User can select the feedstuffs according to the alphabets.  3.) The selected feedstuff displays on the right side.  4.) If user doesn’t want the feedstuff, user can delete them. | The selected feedstuff displayed on the Right. |
| Quantity and Cost | 1.) After cattle weight the page navigates to feedstuff page.  2.) User can enter the cost and the quantity  3.) The quantity is optional whereas the cost is mandatory field.  4.) Only numerical and decimals are accepted. | The data entered by the user should be displayed and can be edited if needed. |
| **4)** | Ration weight Page | Nutrients | 1.) After feedstuff page  the page navigates to Ration Weight Page.  2.) The nutrients are displayed.  3.) The red color is the one with less than the range.  4.) The blue color is the one with more than the range.  5.) The ash color is the one within the range. | The appropriate colors should be displayed according to the nutrient range. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | Total ration | 1.) After feedstuff page  the page navigates to Ration Weight Page.  2.) The nutrients and total ration are displayed.  3.) They are calculated based on the input given by the user | Showing the results of individual ingredient ration and also the total ration |
| Dry Matter | 1.) After feedstuff page  the page navigates to Ration Weight Page.  2.) The nutrients and total ration are displayed along with Dry Matter.  3.) They are calculated based on the input given by the user. |  |

**Integration Testing:**

After each unit is thoroughly tested, it is integrated with other units to create modules or components that are designed to perform specific tasks or activities. These are then tested as group through integration testing to ensure whole segments of an application behave as expected. These tests are often framed by user scenarios, such as logging into an application or opening files. Integrated tests can be conducted by either developers or independent testers and are usually comprised of a combination of automated functional and manual tests.

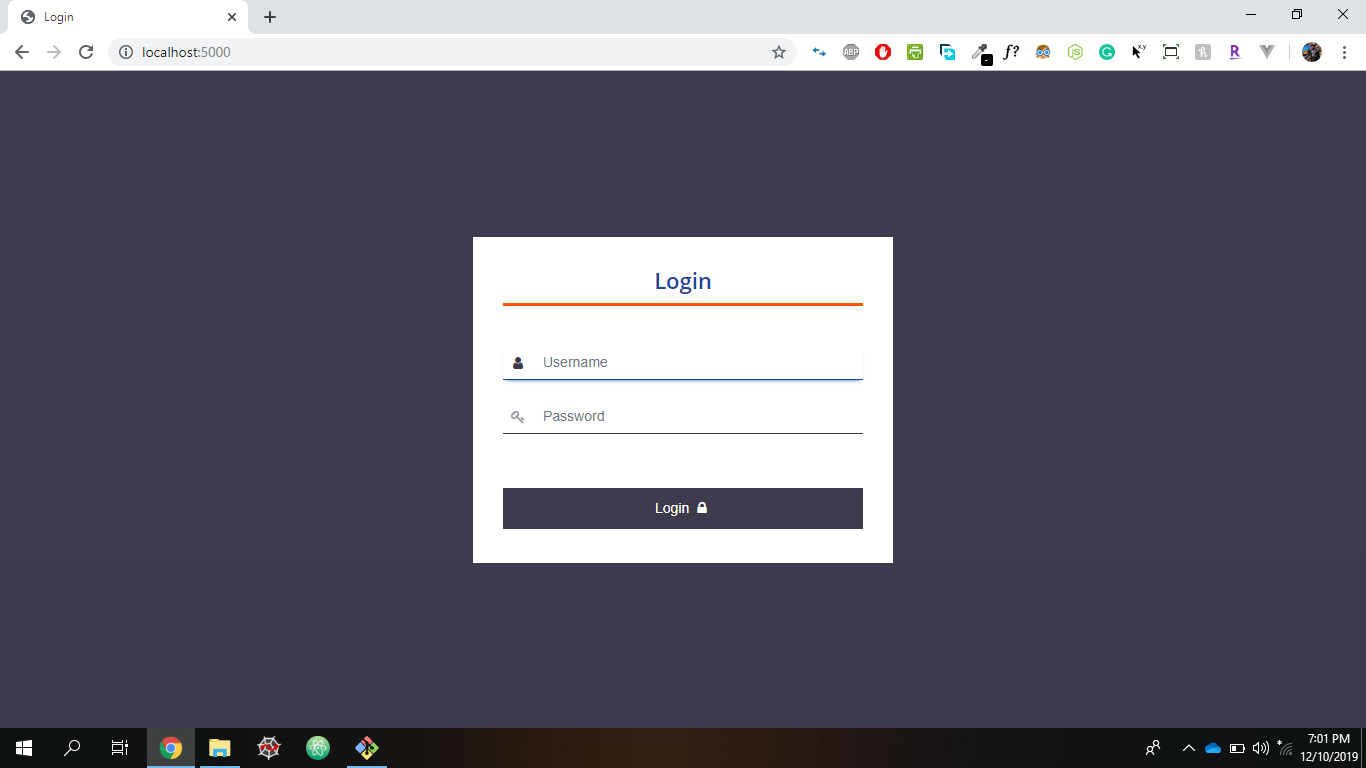
**V. Deployment Artifacts for Users and Clients**

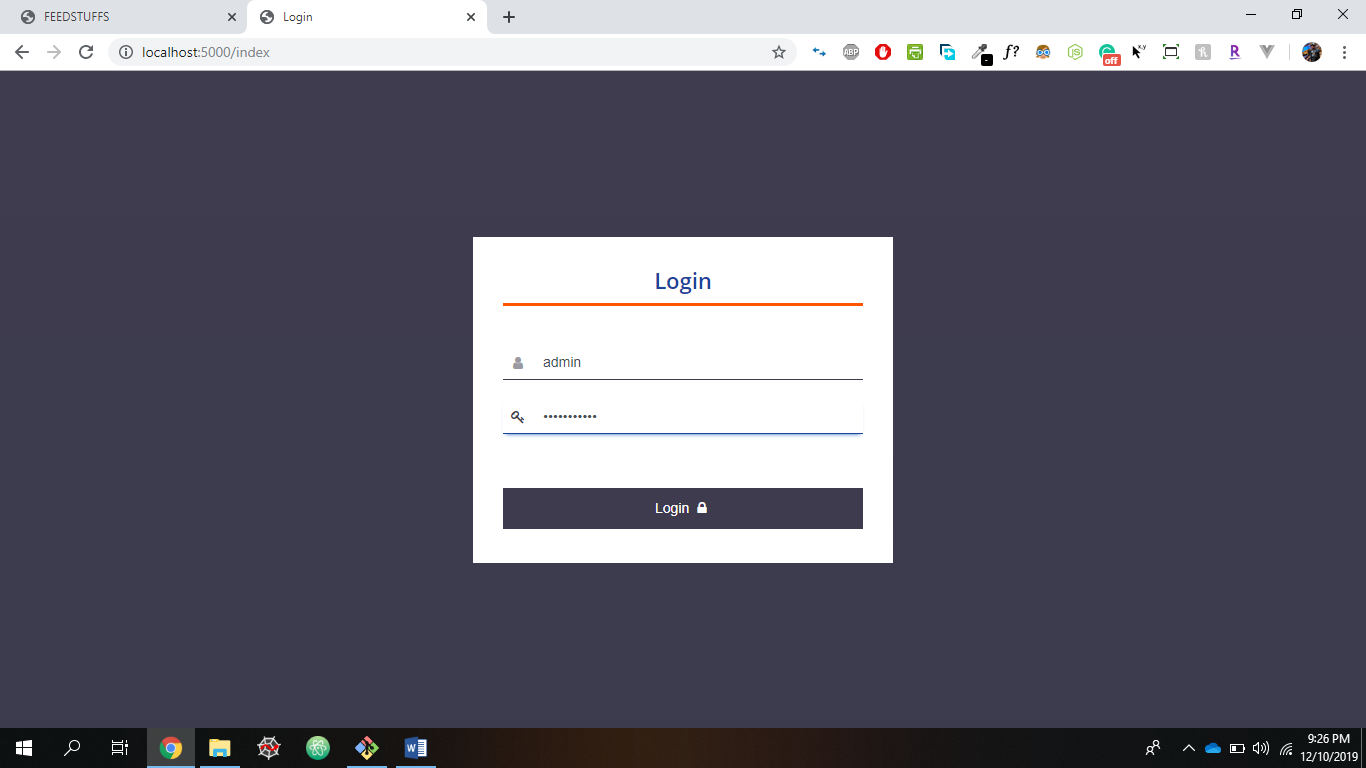
* + - 1. **User’s guide for Users and Clients**

The SSCRA reporting application consists of 3 key functionalities which are managing users, uploading data files and generating reports.

Please note: Initially an admin username and password will be shared with client.

1. **Login Page**

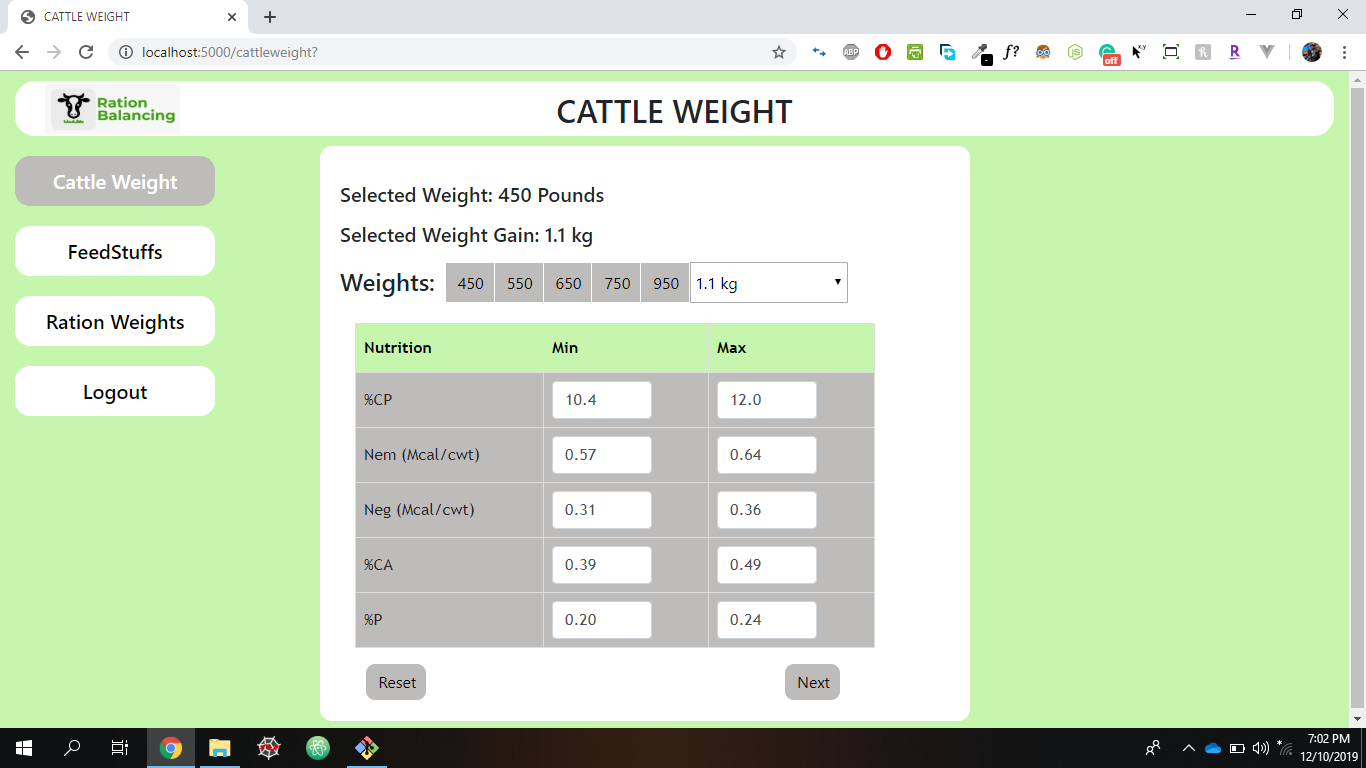




Provide the correct Username and password for Login.

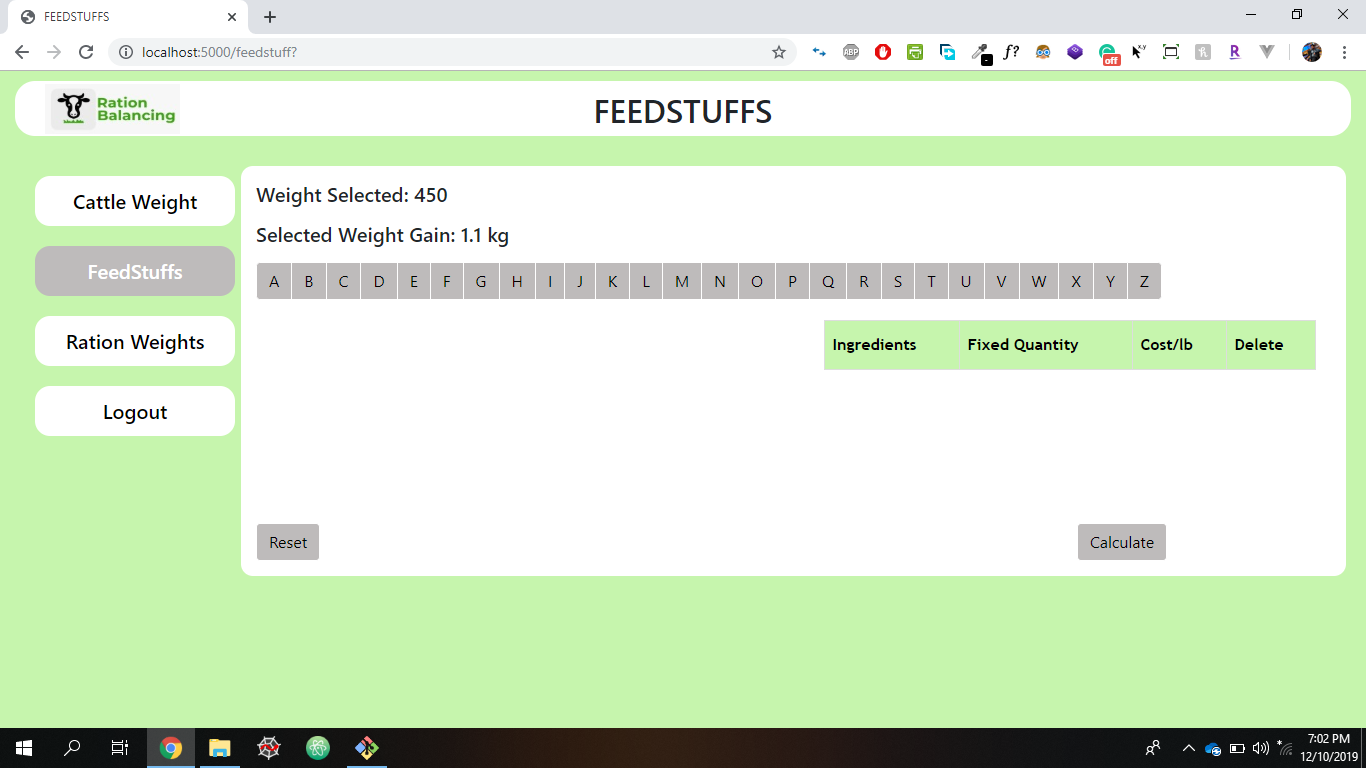
1. **Cattle Weight Page**





The User need to select the weights in pounds and weight gai n. User can manually enter the min max values of the nutrition’s. Reset button is to reset the values which are entered.

1. **FeedStuffs Page**

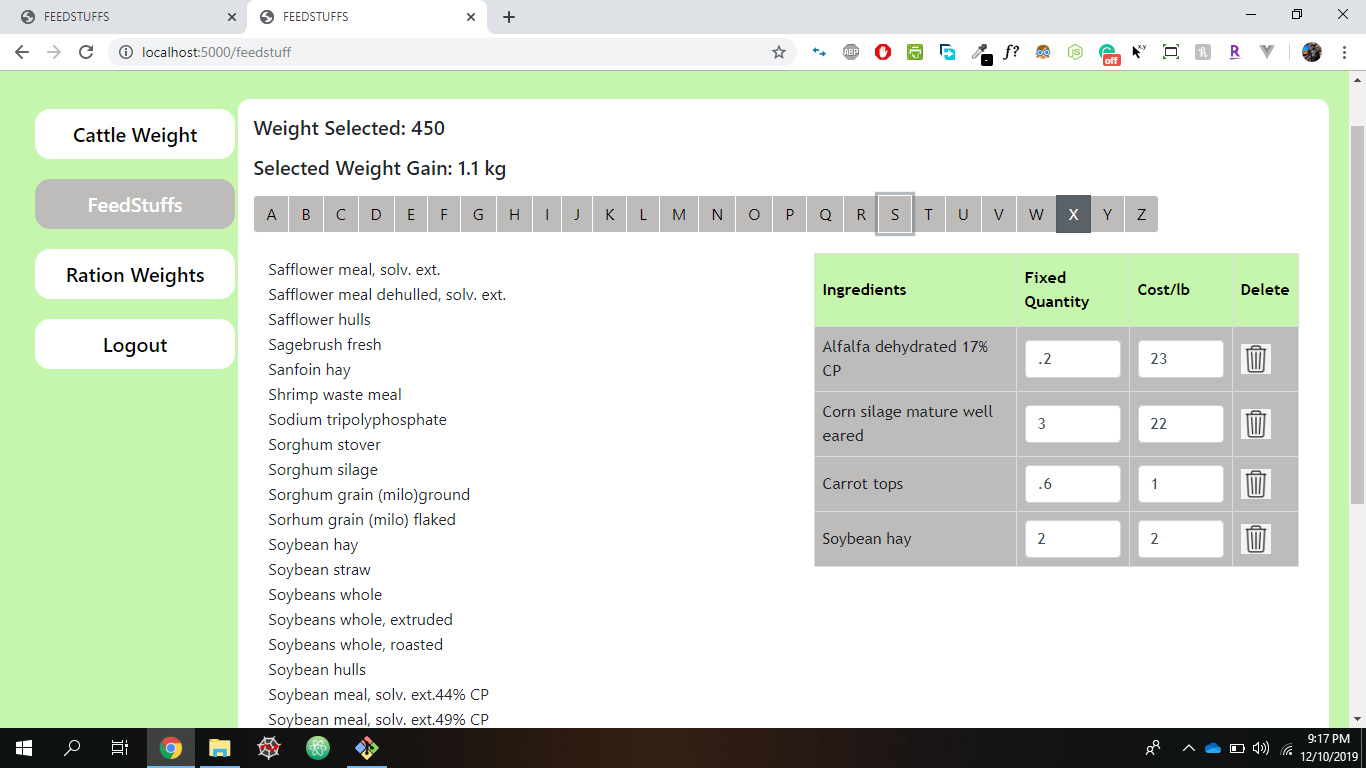


After selected Weights and Weight Gain are displayed in this Feedstuff page.

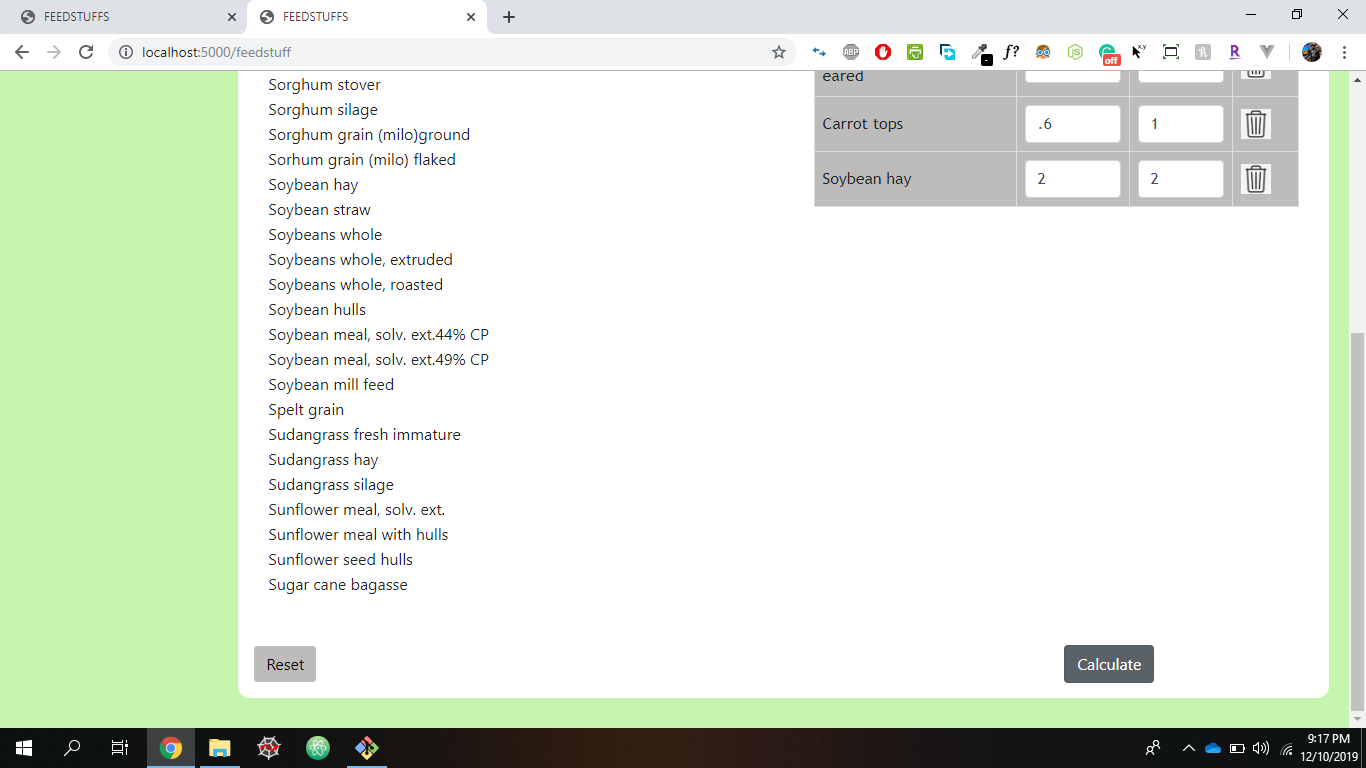


The Ingredients are displayed in alphabetical order. We can select ingredients and the selected ones are displayed on the right side along with fixed quantity and cost.



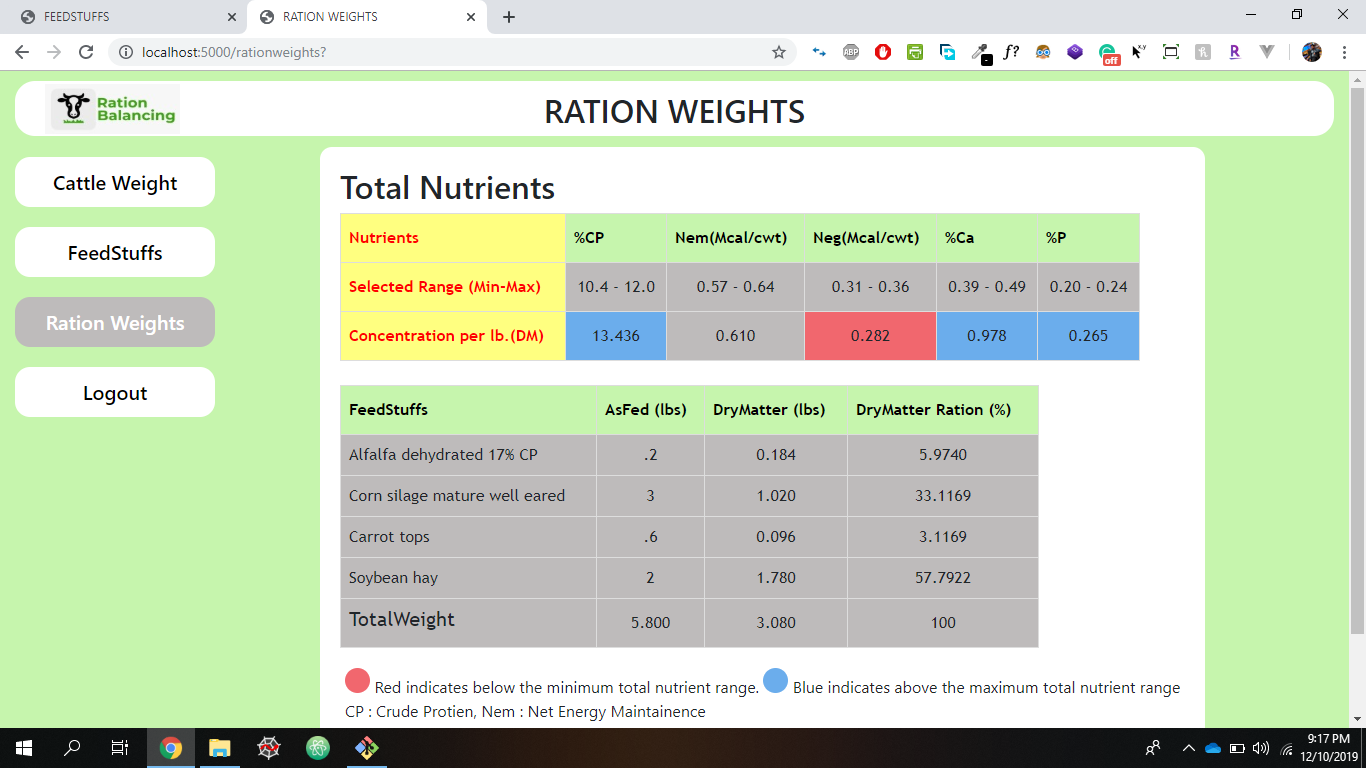


We can also delete any ingredient with delete button. Fixed Quantities and Cost values are to entered according to the user requirements. Here the Cost is the Mandatory field.



Once, the values are entered then user need to press the calculate button to calculate the range and routes to the next Ration Weights Page.

1. **Ration Weights Page**

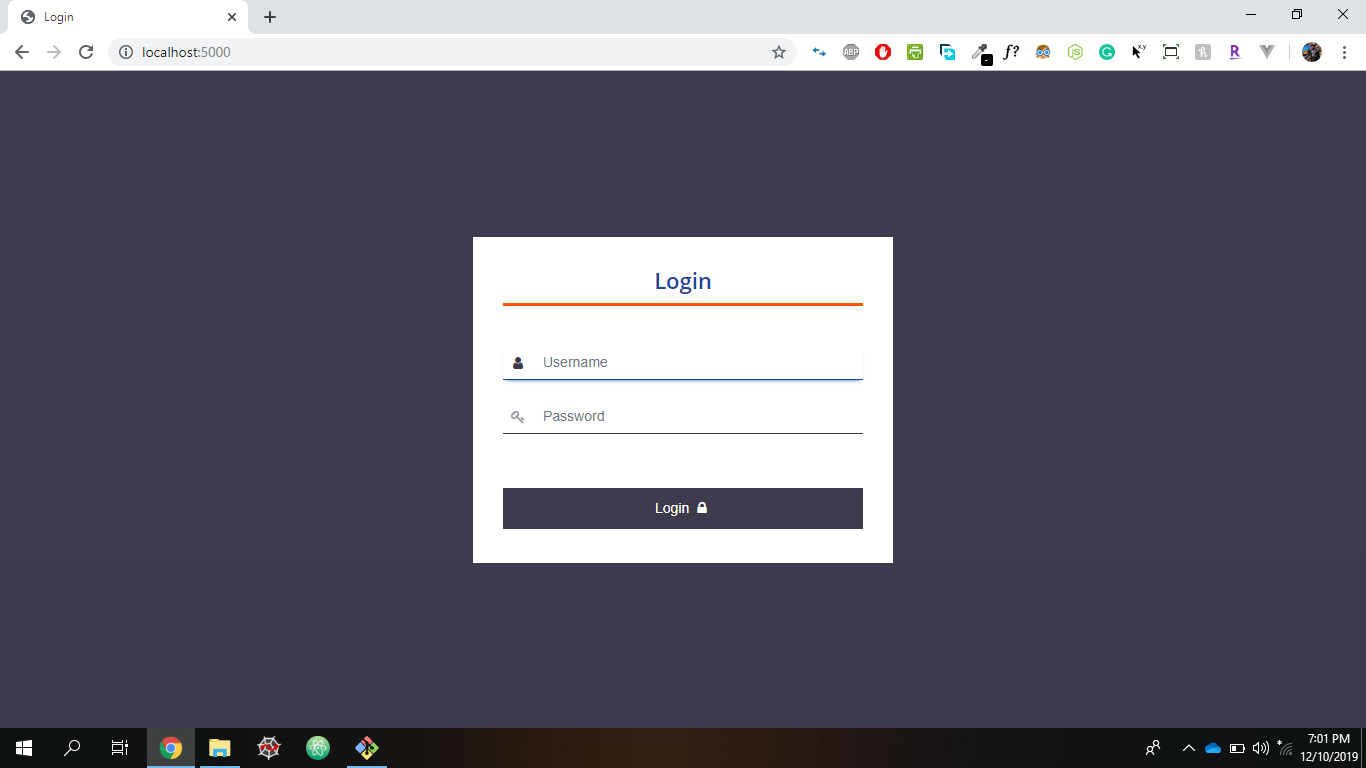


The above page shows that the Total nutrients of selected range (Min-Max) from Cattle Weight Page here the Concentration per lb.(DM) are calculated and displayed based on the selected range through color representation Blue- Above the Maximum range, Red – Below the Minimum range.

It also shows the calculated DryMatter(lbs.) and DryMatter Ration (%) from the selected feedstuffs along with Total weight.

1. **Logout Page**

Once we logout we render to login page.



1. **Installation guide**

**Pre-requisites**

Setup the deployment environment to have latest version of **Node.js** and **Git** & **Tortoise Git** version controlling tool installed.

**Step 1**

This application is developed in Node.js environment for both front-end and back-end.

**Step 2**

Application source code is available in the following GitHub repository. Clone or download the repository on to a Windows OS /Linux OS/Mac OS.

Link: <https://github.com/satyavrath/rationbalancing>

Open the cloned repository location in command prompt or terminal and run `npm install` to get all the node modules installed locally and `npm start` to run the program .

1. **Execution guide**

**Step 1**

**Can directly run the application from given heroku link.**

<http://rationbalancing.herokuapp.com/>

**Step 2**

Open your source location for the project in terminal or command prompt and run `npm install` to get all the node modules installed locally and `npm start` to run the program .

1. **Project URL**

Local application run URL: <http://localhost:5000/>

Project Repository URL: <https://github.com/satyavrath/rationbalancing>

**VI. Project Management Artifacts**

**1. Conclusion**

We can conclude that the application is running smoothly. Security is maintained by providing login for the application. User can enter the details of the cattle like cattle weight , weight gain, nutrition values can be automatically generated by selecting the weights also user can enter the min, max values manually according to the cattle needs. Then ingredients can be selected in a alphabetical order where entering cost of the ingredient is a mandatory field. An option to delete the ingredient is also provided. On clicking calculate button provided it routes to next tab which shows two tables one with nutrient values of the selected ingredients presenting in a color format (blue or red) showing if it’s in between the min-max range.

**2. Pros and Cons**

**Pros:**

* User can login using proper credentials
* Unauthorized credentials are not permitted access into the application
* User can either manually enter the information or can use the generated data.
* Easily select the ingredients alphabetically.
* Simple understandability of min-max range of nutrition through color representation.
* Application works efficiently according to user’s requirements

**Cons:**

* Weight gain must be checked only after selecting the weights.
* Not verifying the length of data being stored in the database
* Uploading a document different from the given extensions .xlsx will lead to error

**3. Summary of Requirements Completed**

1. Login Page
2. Update Cattle Weight
3. Update Weight gain
4. Update Nutrition Min-Max Values
5. Searching Ingredients
6. Update Cost and Fixed Quantity
7. Generating calculated total weights
8. Generating Min-Max nutrient range
9. Logout option

**5. Final Conclusion**

We, as a team, have combined our efforts to develop this application. We would like to finally conclude by acknowledging our mentors, Dr. Richard Scott Bell and Dr. Michael Rogers for their effort and patience in helping us and guiding us through the project and development process. We would also like to thank our Client, Marcus McGee Assistant Professor of Animal Science from the school of Agricultural Department, for giving us the opportunity to provide him with an application as such.