**Determining the regulatory requirements.**

Guelph research center.

GROUP 10: nutricheck.

**User Manual**

**1. Aim and Objectives**

The primary aim of this project is to develop a web-based application that calculates PDCAAS (Protein Digestibility-Corrected Amino Acid Score) and IVPDCAAS (In Vitro Protein Digestibility-Corrected Amino Acid Score) claims for protein sources.

The objectives include:

* Categorizing protein sources as "No claim," "Good source," or "Excellent source" based on predefined calculations.
* Displaying the results in a comprehensive table for clarity and ease of understanding.
* Providing users with a dashboard for data visualization to gain deeper insights into protein quality metrics through interactive charts and graphs.
* Supporting both Excel file uploads and manual data input for flexible analysis.

**2. Proposed Methodology and Technologies**

a) Methodology:

* Users input data via Excel uploads or manual entry.
* The system processes the data using Python and calculates PDCAAS and IVPDCAAS values.
* Protein sources are categorized based on these calculations.
* Results are visualized in a dashboard with interactive charts for deeper insights.

b) Technologies Used:

* Frontend: HTML, CSS, JavaScript for UI design, and Chart.js for dynamic data visualizations.
* Backend: Python and Django for server-side processing.
* Database: SQLite for storing processed data.
* Libraries: Pandas for data processing and ReportLab for generating PDF outputs.

**3. Initial Setup and Requirements**

A) System Requirements

Operating System: Windows, macOS, or Linux.

Python: Version 3.8 or above.

IDE: PyCharm (recommended for Django development)

B) Setup Instructions

Verify if Python is Already Installed:

* Open your terminal:
  1. **Windows**: Press Win + R, type cmd, and hit Enter.
  2. Type the following command and press Enter:
     + python --version
  3. **Mac**: Open Terminal from Applications > Utilities.
  4. Type the following command and press Enter:
     + python3 --version

Install Python:

For Windows:

* Go to the official Python website: <https://www.python.org/downloads/>.
* Click the **Download Python** button. Ensure the version matches your system (64-bit is recommended).
* Run the downloaded installer:
  1. During installation, check the box **Add Python to PATH**
  2. Choose "Install Now" or customize the installation directory.
* Once installed, verify the installation by opening Command Prompt and typing:
  1. python --version

For Mac:

* Open your browser and visit <https://www.python.org/downloads/>
* Click the **Download Python** button for macOS.
* Open the .pkg file and follow the installation instructions.
* After installation, verify it by typing in the terminal:
  + python3 –version

Install PyCharm:

Download and install PyCharm from https://www.jetbrains.com/pycharm/.

Set Up Virtual Environment:

* In PyCharm, create a new project.
* Use Virtualenv for isolated environments:
* python -m venv env
* source env/bin/activate # macOS/Linux
* env\Scripts\activate # Windows

Install Dependencies:

* Run the following commands to install required libraries:
* pip install requirements.txt
* Clone the Repository:
* Clone the GitHub repository using:
* git clone [your-repository-link]
* cd [project-folder-name]

Create an admin user:

* Run the following command to create a superuser:
  1. python manage.py createsuperuser
* Follow the prompts:
  1. Enter your desired **username**.
  2. Enter and confirm your **email address**.
  3. Enter and confirm your **password**.
* If the process completes successfully, you will see a message: “Superuser created successfully.”

Run the Server:

* Apply migrations:

python manage.py migrate

* Start the server:

python manage.py runserver

* Access the application at <http://127.0.0.1:8000/>.

Add RACC values at Admin Page:

* Navigate to <http://127.0.0.1:8000/admin> and log in using the superuser credentials you created.
* Click on RACC values user Protein\_label
* Click on Add RACC value button
* Add the specific menu items and its categories:
* For **Legumes**:
  1. Lentil: 35
  2. Tofu: 85
  3. Bean: 35
  4. Pea: 35
  5. Soy: 35
  6. Chickpea: 35
* For **Cereals**:
  1. Hemp: 15
  2. Oats: 40
  3. Wheat: 15
  4. Corn Meal: 30
  5. Rice: 45
  6. Chia Seeds: 15
  7. Tapioca: 15
  8. Flax: 15
  9. Bran: 15
* For **Flours**:
  1. Buckwheat: 30
  2. Potato: 30
  3. Corn: 30
  4. Sorghum: 30
  5. Rye: 30
  6. Almond: 15
  7. Soya: 15
  8. Coconut: 15
  9. Peanut: 15
  10. Barley: 30

**4. Step-by-Step Guide to Operate the Application**

Step 1: Access the Application

Open the application in your browser by navigating to the provided URL.

A screenshot of a computer screen

Description automatically generated

Fig-1: The homepage of the project

Step 2: Upload Data

Navigate to the Upload page.

Upload an Excel file containing sample data.

A screenshot of a computer

Description automatically generated

Fig-2: The Excel Upload page where the user can upload the document

Step 3: Use Manual Input

Alternatively, if you do not wish to upload an Excel file, navigate to the Manual Input section.

Enter the following parameters in the provided fields:

Sample Name: The name of the protein source (e.g., " Buckwheat Flour Control").

Protein Percentage (%): The percentage of protein in the sample (e.g., 12.47).

PDCAAS: The Protein Digestibility-Corrected Amino Acid Score (e.g., 54.27).

IVPDCAAS: The In Vitro Protein Digestibility-Corrected Amino Acid Score (e.g., 55.97).

Click Submit to process the data and generate results.

A screenshot of a computer

Description automatically generated

Fig-4: The feature for manual input of protein data

Step 4: Process the Data

After uploading an Excel file or entering manual input, click Submit to process the data.

The application calculates PDCAAS and IVPDCAAS scores and categorizes the protein sources based on these results.

A screenshot of a computer

Description automatically generatedFig-5: The user uploading the document

Step 5: Results

Results for uploaded excel sheet

The results table provides a detailed breakdown of the processed data, including the calculated claims and labels.

A screenshot of a computer

Description automatically generatedFig-6: The results table where a user can filter results based on PDCAAS and IVPDCAAS labels

A screenshot of a computer

Description automatically generatedFig-7: The four columns which give the labels and claims

Results for Manual Input

A screenshot of a computer

Description automatically generatedFig-8: The results of the manual input data given by user

Step 6: Apply Filters

Filter the processed data based on PDCAAS or IVPDCAAS labels ("No claim," "Good source," or "Excellent source").

The displayed results dynamically update based on the selected filter.

A screenshot of a computer

Description automatically generatedFig-9: The results based on the filtering option

Step 7: View Dashboard Visualizations

Navigate to the Dashboard for interactive data visualizations, including:

Polar Chart: Displays PDCAAS and IVPDCAAS distributions.

A screenshot of a computer

Description automatically generatedFig-10: Polar charts for PDCAAS and IVPDCAAS distribution

Bar Chart: Compares protein percentages across samples.

A screenshot of a graph

Description automatically generatedFig-11: The Bar chart for protein percentages

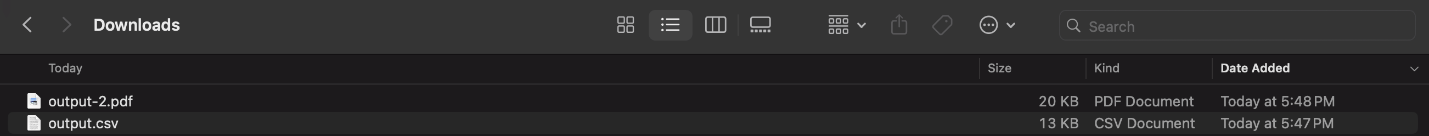
Stacked Bar Chart: Shows amino acid compositions for each sample.

A screenshot of a computer

Description automatically generatedFig-12: The Stacked Bar chart for amino acid compositions

Step 8: Download Results

Use the Download feature to export results in CSV or PDF formats for offline analysis.

Fig-13: The results can be downloaded in CSV or PDF formats

A screenshot of a computer

Description automatically generatedFig-14: The results in PDF format

A screenshot of a computer

Description automatically generatedFig-15: The result in CSV format

A screenshot of a computer

Description automatically generatedFig-16: The labels and claims can be viewed in the downloaded results

5. Best Practices and Lessons Learned

Best Practices:

* Column Splitting for PDFs: Implemented to manage large datasets, ensuring clean formatting and readability.
* Dynamic Filtering: Enabled real-time filtering of data based on user-selected criteria for better usability.
* Error Handling: Designed robust error handling for unsupported file uploads to improve user experience.
* Responsive Visualizations: Ensured that charts adapt seamlessly to different screen sizes and resolutions.

Lessons Learned:

* Clean and structured data is essential for accurate processing and visualization.
* User feedback is valuable in iterating features for better interactivity and ease of use.
* Optimization is key when handling large datasets to maintain performance and responsiveness.

6. GitHub Repository Link

The complete source code and documentation for this project are available at:

[Click here to go to GitHub Repository](https://github.com/Shivamani-A/Nutricheck)