Low Level Design

Dashboard

User Workflow

1. Login

- **Action:** User fill username and password
  - Click Login
  - API returns a bearer token to access all the API endpoints

2. Case Creation

- **Action**: User initiates creating a new case.
- **Location**: Case can have many tags.
- **Input Form**:
  - **Case Name**: e.g., "Country A Coffee Production Comparison".
  - **Case Description**: Description of a case.
  - ~~**Date**: Current date or a selected date.~~
  - **Private**: Checkbox (True/False)
  - **Year in Case Timeline**: Drop-down selection.
  - **Country**: Specify the country of focus.
  - **Currency**: Drop-down selection.
  - **Commodity**: Drop-down to select a commodity (predefined from master data).
  - **Area Size Unit**: Drop-down (e.g., hectare, m2, count, acre).
  - **Volume Measurement Unit**: Drop-down (e.g, KG, Ton, etc).
  - ~~**Cost of Production Measurement Unit**: Drop-down choices (Per-area/Per-yield/Per-tree).~~
  - ~~**Reporting Period**: Drop-down choices (Per-season/Per-Year), then show previous data if it's available.~~
  - ~~**Segmentation**: Toggle option (On/Off).~~
  - **Multiple Commodities**: Toggle option (On/Off)
  - ~~**Upload Logo**: File upload feature.~~
- **Action**: User clicks "Next".

3. Multiple Commodities Selection (Displayed if "Multiple Commodities" is toggled ON)

- **Secondary Income Source**:

    - Drop-down to select source (predefined by system).
    - Checkbox: "Data on income drivers available". If Checked will enable below fields:
        - Drop-downs for area unit
        - Volume measurement unit
        - ~~Cost of production measurement.~~
- **Tertiary Income Source**:
    - Same options as the secondary source.(Optional)
- **Action**: User clicks "Next".

4. Segment Creation & Comparison

- **Segment Creation**: User begins creating segments (up to a maximum of 5).
- **Target Fields:** Income Target Input
    - In this section, user can set Income Target based on region or set manually.
    - User can set Income Target manually by turn on the toggle "Set income target yourself?".
        - This will hide the per Region Income Target fields,
        - Then show up the input field for manual Income Target
    - When set Income Target based on a region,
        - User can change the number of adult and children in household.
        - After that system will calculated the Income Target automatically.
- **Segment Fields**:
    - Each segment allows input for both current and feasible levels of eg.:
        - **Land/Area**:
            - **Expand Option**: If users click "Expand" beside the "Area" field:
                - They can input details such as:
                    - Total land owned.
                    - ~~Land allocated to the primary commodity.~~
                    - ~~Land allocated to other purposes.~~
        - **Volume**:
            - **Expand Option**: If users click "Expand" beside the "Yield" field:
                - They can input details such as:
                    - Total Yield per weight/area unit.
                    - Total Loss per weight/area unit.
        - **Price**:
            - **Expand Option**: If users click "Expand" beside the "Price" field:
                - They can input details such as:
                    - Farmgate price per currency/weight unit.
                    - Price premium per currency/weight unit.
        - **Cost of Production**:
            - **Expand Option**: If users click "Expand" beside the "Cost of Production" field:
                - They can input details related to various cost components per currency/area unit.
        - **Diversified Income**:

    - **Expand Option**: If users click "Expand" beside the "Diversified Income" field:
        - They can input details related to income from other sources.
- **Household Income Calculation**: After filling all fields, the system calculates and displays the household income.
- **Action**: User proceeds to the next page. (**After mandatory input filled:** Target, Household Size if living income)

5. Data Output Dashboard

- **Tabs**:
    - **Income Overview**: Displays a summarized view of the income metrics.
        - **Charts**:
            - **Current and Feasible**:
                - **Stack Bar** chart:
                    - Current & Feasible value of Primary Commodity.
                    - Current & Feasible value of Diversified Income.
                    - Current & Feasible value of Cost of Production with negative Y-Axis value.
                - **Y-Axis:** Income (currency).
                - **X-Axis:** Segments.
            - **Income Gap**:
                - **Stack Bar** chart:
                    - Current & Feasible value of Household Income.
                    - Current & Feasible value of Income Gap.
                - **Diamond** mark of Income Target for each segment.
                - **Y-Axis**: Income (currency).
                - **X-Axis**: Segments.
            - **Biggest Impact on Income**:
                - **Segment Tab** selection.
                - **Bar** chart:
                    - Change of income driver value in %.
                    - Percentage change in income with driver at feasible level, while all other drivers stay the same.
                    - Percentage change in income when this driver moves from current to feasible level while all other drivers at feasible level.
                - **Y-Axis**: Change (%).
                - **X-Axis**: Drivers of selected Segment.
            - **Breakdown of Drivers**:
                - **Segment Tab** selection.
                - **Dropdown** of Drivers (visual will generated based on selected driver)
                    - ***question with null parent id*** (Income Driver)
                - **Stack Bar** chart:
                    - Each selected dropdown will generate different stack.

        - Stack shows break down of selected income driver / answer of child value (sub-components).
      - **Y-Axis**: Each driver unit size
      - **X-Axis**: Segment
    - **Monetary Impact to Income**:
      - **Segment Tab** selection.
      - **Waterfal** chart:
        - illustrates how adjustments in income drivers influence the transition from the current income level to a feasible income level.
      - **Y-Axis**: Income (currency)
      - **X-Axis**: Drivers of selected segment
    - **Commodity Income Comparison**:
      - **Segment Tab** selection.
      - **Bar** chart (Current Revenue and Feasible Revenue)
      - **Y-Axis**: Income (currency)
      - **X-Axis**: Commodities include Diversified Income based on selected segment
- **Sensitivity Analysis**: Allows users to understand how different combinations of income drivers affect income levels.
  - **Input**:
    - **Select Segment:** drop-down of the segments in case
    - ~~**Select Commodity:** drop-down for The Focus commodity / Secondary / Tertiary commodity (eliminate commodity where it's not break down)~~
    - **Breakdown of Drivers** table: This table will show the breakdown of drivers for selected segment
    - **Select Binning Driver:** drop-down for ***question with null parent id*** (Income Driver)
      - **Number of Bins:** Input Number
    - **Select X-Axis Driver:** drop-down for ***question with null parent id*** (Income Driver)
      - **Min:** Input Number
      - **Max:** Input Number
    - **Select Y-Axis Driver:** drop-down for ***question with null parent id*** (Income Driver)
      - **Min:** Input Number
      - **Max:** Input Number
  - **Adjust Income Target** feature
    - **Select Option** to choose whether you would like to express the changes in current values using percentages or absolute values.
    - **%Change** Input: when choose percentage.
    - **Adjusted Target** Input: when choose absolute.
  - **Charts**:
    - **Zone between current and feasible:** Line with intersection bar zone
    - **Income Heat-map:** Heat-map Cartesian
- **Scenario Modeling**: Enables users to visualize potential scenarios and their impact.

    - **Input:**
        - **Choose Approach**: Select dropdown (percentage, absolute)
        - **Scenario Title**: Input Text
        - **Scenario Description**: Input Textarea
        - **Segment Tabs:** tab section of the segments in case
        - ~~**Select Commodity:** drop-down for The Focus commodity / Secondary / Tertiary name~~
        - **Input for Each grand parent questions** (***question with null parent id / Income Drivers***), e.g:
            - Diversified Income
            - Volume / Yield
            - Price
            - Cost of Production
            - Land Area
    - **Charts**:
        - **Scenario Chart:**
            - **Stack Bar** Chart:
                - Current total household income.
                - Additional income when income drivers are changed.
                - Gap between current and additional income
            - **Y-Axis:** Income (currency)
            - **X-Axis:** Each Segments
        - **Income Gap across Scenario Chart**
            - **Select Scenario** dropdown: select scenario - segment across scenario
            - Same chart as scenario chart but user can select to show a chart from different scenario
    - **Scenario Outcomes**
        - Visualize as a table.
        - User can compare specific outcomes per segment to understand in which scenario the farmers in that segment reach the income target, how this is established, and how it compares to the current scenario.
    - **New Scenario Button:** To create new scenario, maximum 3 scenarios.
    - **Export**: each chart has export/download button.

6. Admin Page

- **Access:** Admin Page can be accessed via button on top headers, this button only show if the role of logged in user is admin
- **Tabs**:
    - **Users**:
        - Display list of user and basic information and role in table format along with edit and delete buttons
        - Pagination 10 users per-page
        - Button to add new user
    - **Income Benchmark**:

        - Display list of Income Drivers in table format along with edit and delete
        - Pagination 10 data per-page
        - Button to add new Income Drivers
    - **References**:
        - Display list of References in table format along with edit and delete
        - Pagination 10 data per-page
        - Button to add new References
- **Actions:**
    - **Add a New User**
        - Admin Click Add new from the user list
        - **Fields:**
            - Full Name
            - Email
            - Role: Admin, User
            - Access: List Access of Case Tags / Individual Cases
        - After Successfully added:
            - The User will received an email to create a password
            - The Invited User click the button to create a new password in the email
            - New tab to the page opened: User create a new password with 2 verification

Roles and User Access

Roles

Super Admin

- Database Role Enum: **super_admin**
- Privileges:
    - Able to view **all the users** in the platform
    - Able to view and edit all of **the existing cases**
    - Can create new **case** and add a User as **Viewer** or **Editor** of the case, but in most cases most likely not the case
    - Can invite new **Business Unit Administrator** or modify existing user as a **Business Unit Administrator**
    - Can invite new **Super Admin** or modify existing user as **Super Admin**
    - Can add or modify **Tags**
    - Can do everything the Business Unit Administrators can do
- Workflow:
    - **Step 1: Log In**
        - Super Admin logs into the admin panel using their credentials.
    - **Step 2: Navigate to User Management**
        - Upon logging in, the Super Admin is presented with a dashboard.
        - From the dashboard, the Super Admin should navigate to the "User Management" section or a similar area where user-related actions can be

performed.
    - **Step 3: Select "Add User"**
        - In the User Management section, there should be an option to "Add User" or a similar action.
        - Click on "Add User" to initiate the process of adding a new user.
    - **Step 4: Enter User Details**
        - A form or interface appears where the Super Admin can enter the new user's details. These details may include:
            - Username
            - Email address
            - User's role (e.g., Business Unit Administrator, Business Unit Editor, Business Unit Viewer, etc.)
            - Business Unit (if applicable)
    - **Step 5: Set Permissions**
        - Depending on the system's design, the Super Admin may have the option to set specific permissions for the new user, such as granting access to certain cases or functionalities.
    - **Step 6: Save User Details**
        - After entering all the necessary information, the Super Admin clicks on the "Save" or "Create User" button to add the new user to the system.
    - **Step 7: Confirmation**
        - The system should provide a confirmation message indicating that the new user has been successfully added.
    - **Step 8: Notify User (Optional)**
        - This action triggers an email notification to the new user, inviting them to set their password and activate their account.

Business Unit Administrator

- Database Role Enum: **admin**
- Privileges
    - Can have multiple **Business Unit**
    - Able to view **all the users** with **the same Business Unit**
    - Able to view and edit all of **the existing cases** in **the same Business Unit** even if it's **private cases**
    - Can create new **case** and add a User as **Viewer** or **Editor** of the case
    - Can invite new **User** or approve new registered user with **the same Business Unit**
    - Can modify role of the User in the same **Business Unit** as case **Editor** or case **Viewer**
    - Can add user to an existing case as editor if:
        - The user current role is not **Editor**
        - The user is not the owner of the case
- Workflow
    - **Step 1: User Registration**
        - A new user registers for an account on the platform, providing their details, including username, email address, and Business Unit during the registration

process.
    - **Step 2: Review Pending Users**
        - After registration, the user's account is created with a "Pending" status, and their details are added to a list of pending users.
    - **Step 3: Business Unit Manager Logs In**
        - The Business Unit Manager logs into the admin panel using their credentials.
    - **Step 4: Navigate to User Approval**
        - From the dashboard or admin panel, the Business Unit Manager navigates to the "User Approval" or "Pending Users" section.
    - **Step 5: View Pending Users**
        - In the "Pending Users" section, the Business Unit Manager can see a list of users who have registered with the same Business Unit and are awaiting approval.
    - **Step 6: Review User Details and Modify Role**
        - The Business Unit Manager selects the user they wish to approve and reviews their registration details. They also have the option to modify the user's role by selecting either "Editor" or "Viewer" from a drop-down menu.
        - If the role is Editor, show radio button:
            - All case in **The Business Unit of the User**
            - Create / Edit case
        - The Business Unit Manager can also change the Business Unit of registered user *only* with their Business Unit
    - **Step 7: Approve User**
        - If the Business Unit Manager is satisfied with the user's details and eligibility, they click on an "Approve" or "Activate" button associated with the user's profile.
    - **Step 8: User Notification**
        - Upon approval, the system sends an email notification to the newly approved user, informing them that their account has been activated and is now accessible.
    - **Step 9: User Marked as Active**
        - Back in the admin panel, the system automatically updates the user's status from "Pending" to "Active" in the user list, indicating that they have been approved and can now access the platform.

Regular User (Users with Business Unit)

- Database Role Enum: **user**
- Privileges
    - They can view all cases , except private
    - Can create new cases
    - They can always edit / view cases if they explicitly added to case (**User Case Access Table**)
- Workflow:
    - **Registration / Invitation:**

        - Regular User sign up with business unit. Then BU admin get notified to approve the user.
        - Business Unit add the regular user and assigned business unit (if they have multiple business unit), notifying the relevant manager(s) via email
    - **View**:
        - All cases except private cases
        - Any private cases if explicitly added to the case as viewer
        - Only case owner can explicitly add as viewer to their cases
    - **Edit**:
        - For cases where they are owner of the case
        - Any cases if explicitly added as editor to the case
        - Only case owner can explicitly add as editor to their cases

External User

- Database Role Enum: **user**
- Privileges
    - Able to see specified case
- Workflow:
    - **Step 1: Registration:**
        - Sign up to Register Page.
        - **Super Admin** review the request.
    - **Step 3: Access to Case:**
        - **Case owner** adds the external user as a viewer or editor to the specified case.
            - Suggestion when they start typing a user (full text search: name / email)
            - The invited user gets a notification

Possibilities

Example Scenario:

- Case 1:
    - Case Name**: Rice in Indonesia**
    - Tag: **Rice**
    - Business Unit Name: **Bali Rice**
- Case 2:
    - Case Name: **Coffee in Vietnam**
    - Tag: **Coffee**
    - Business Unit Name: **Nes-Kopi**
- Case 3:
    - Case Name: **Rice in Bangladesh**
    - Tag: **Rice**
    - Business Unit Name: **Bangla Rice Trade**

Glossary:

- WRP (Within the realm of possibility): something may not be true in the reality, but it could potentially happen within the system or context
- EPC (Edit permission in case)

User Role	All Cases	BU	BU Role	Tag	Description
Super Admin	true	-	null	-	
Admin	true	BR	admin	-	• Rice in Indonesia: ◦ edit ◦ assign • Coffee in Vietnam: ◦ view ◦ edit (EPC, WRP) • Rice in Bangladesh: view

Admin	false	BR	admin	rice	• Rice in Indonesia:
					  ◦ edit
					  ◦ assign
					• Coffee in Vietnam:
					  ◦ noaccess
					  ◦ edit (EPC, WRP)
					• Rice in Bangladesh:
					  ◦ view
					  ◦ edit (EPC, WRP)

Admin	false	BR	admin	coffee	• Rice in Indonesia:
					  ◦ edit
					  ◦ assign
					• Coffee in Vietnam:
					  ◦ view
					  ◦ edit (EPC, WRP)
					• Rice in Bangladesh:
					  ◦ noaccess
					  ◦ edit (EPC, WRP)

Admin	false	BR	admin	-	• Rice in Indonesia: ◦ edit ◦ assign • Coffee in Vietnam: ◦ no access ◦ edit (EPC, WRP) • Rice in Bangladesh: ◦ no access ◦ edit (EPC, W

Data Layer

Data Input

- **Cases**: Store case-related data like name, date, country, selected focus, currency, units, etc.
- **Master Data**: Predefined drop-down values for year, country, commodities (focus), currency, area size unit, volume measurement unit, cost of production measurement unit, etc.
- **Segments**: Store segment data like area, yield, price, cost of production, and diversified income values for both current and feasible levels.

Data Models

- **Case Model**: This will encapsulate all the attributes related to a case including tags, user access, logo, etc.
- **Segment Model**: This will hold data for each segment within a case.

Logic Layer

Case Creation

- Logic to create, update, and delete cases.
- Logic for case permission.
- Logic to validate case attributes, especially drop-down values against master data.
- Logic to handle logo uploads.

Segment Logic

- Logic to create up to 5 segments for a case.
- Logic to calculate household income based on the provided segment data.

Visualization Logic

- Logic to store input changes
- Logic for e-charts which can dynamically changed by input fields

Dashboard Logic

- Logic to generate data points for the Income Overview, Sensitivity Analysis, and Scenario Modeling tabs based on case and segment data.

Presentation Layer

Case Creation Page

- Form to input case details.
- Drop-downs populated from master data.
- Toggle switches for options like segmentation, living income study, multiple commodities, etc.
- Logo upload functionality.
- Add New Question
  - Only applies in Level 2
  - Always has Total (ID: 1 from master data) as Parent
  - Linked to a Specific Case

Multiple Commodities Page

- Dynamic forms based on whether the user selects secondary and tertiary income sources.
- Each form includes related drop-downs and check-boxes.

Comparison Page

- Display side-by-side segments (up to 5).
- Input fields for current and feasible levels of income drivers for each segment.
- Display calculated household income.

Data Output Dashboard

- Three tabs: Income Overview, Sensitivity Analysis, and Scenario Modeling.
- Display data visualizations and metrics relevant to each tab.

Master Data

1. Commodity Categories

List of Commodity Categories to define the list of questions.

2. Commodities (Focus)

List of commodities that users can select from when specifying the focus of their case. Each commodity might have associated attributes like average yield, common diseases, growth cycle, etc.

3. Countries

List of countries that users can select for their cases. This can be further detailed with regions or cities if needed for more granularity.

4. Currencies

List of currencies that users can select for monetary values in their cases. This might also include current exchange rates if conversions are needed.

5. Area Size Units

Units for measuring land/area such as hectares, square meters, acres, etc.

6. Volume Measurement Units

Units for quantifying production, e.g., kilograms, liters, bushels, etc.

7. Cost of Production Measurement Units

Standardized units for cost measurements like Per-area, Per-yield, Per-tree, etc.

8. Reporting Periods

Standard periods like Per-season, Per-Year, etc., which users can select for their cases.

9. Secondary & Tertiary Income Sources

A list of possible secondary and tertiary income sources, such as livestock, other commodities, etc. This could also have attributes related to average income or typical costs associated with each source.

10. Years for Case Timeline

A list of years that users can select from for their case timeline.

11. Questions (Expandable Inputs)

List of Question can be found here: **Income driver breakdown and units**

> Note that each question could have default calculation logic that can be loaded into the default_value of the Question table to facilitate automatic computations on the front-end.

12. Living Income Benchmark

List of Living Income Benchmark

Database Schema

User Table

Column Name	Data Type	Description	Constraints
id	INT	Unique identifier.	Primary Key, Auto-Increment
organization_id	INT	Identifier for Organisation	Foreign Key (References organisation_id)
email	VARCHAR	Email of user	Not Null, Unique
full name	VARCHAR	Full name of user	Not Null
password	TEXT	User Password	Nullable
role	ENUM	Whether user is admin or business unit admin, etc.	Not Null, ENUM (super admin, admin, user)
all_cases	BOOLEAN	Whether all cases or only create	Not Null, Default False
is_active	BOOLEAN	Whether user is active	Not Null, Default False
invitation_id	VARCHAR	Invitation ID for new user	Nullable, Default (Random String)
created_at	DATETIME	Timestamp when the user was last created.	Not Null, Default Current Timestamp
updated_at	DATETIME	Timestamp when the user was last updated.	Nullable

Business Unit

Column Name	Data Type	Description	Constraints
id	INT	Unique identifier.	Primary Key, Auto-Increment
name	VARCHAR	Name of Business Unit	Not Null, Unique

User Business Unit

Column Name	Data Type	Description	Constraints
id	INT	Unique identifier.	Primary Key, Auto-Increment
user_id	INT	Identifier for the user.	Foreign Key (References user_id)
business_unit_id	INT	Identifier for the business unit	Foreign Key (References business_unit_id)

Column Name	Data Type	Description	Constraints
role	ENUM	Whether admin or member	Not null, ENUM (admin, member)

Note:

- unique between **user_id** and **business_unit_id**

User Case Access

Column Name	Data Type	Description	Constraints
id	INT	Unique identifier.	Primary Key, Auto-Increment
user_id	INT	Identifier for the user.	Foreign Key (References user_id)
case_id	INT	Identifier for the case	Foreign Key (References case_id)
permission	ENUM	Whether edit or view	Not null, ENUM (edit,view)

- **Unique Constraint:** user_id, case_id
- **Notes:**
  - Many to Many (Users & Cases)

Commodity Category Table

Column Name	Data Type	Description	Constraints
id	INT	Unique identifier for each commodity.	Primary Key, Auto-Increment
commodity_category_name	VARCHAR	Name of the commodity (e.g., "Land Commodity, Aquaculture, Livestock").	Not Null, Unique

- **Notes:**
  - Commodity will initially seeded via CLI

Commodity Table

Column Name	Data Type	Description	Constraints
id	INT	Unique identifier for each commodity.	Primary Key, Auto-Increment
commodity_category_id	INT	Identifier for the commodity category.	Foreign Key (References commodity_category_id)
commodity_name	VARCHAR	Name of the commodity (e.g., "Coffee").	Not Null, Unique

- **Notes:**
  - Commodity will initially seeded via CLI

Tag Table

Column Name	Data Type	Description	Constraints
id	INT	Unique identifier for each tags.	Primary Key, Auto-Increment
name	VARCHAR	Name of the tag.	Not Null, Unique
created_at	DATETIME	Timestamp when the tag was created.	Not Null, Default Current Timestamp
updated_at	DATETIME	Timestamp when the tag was last updated.	Nullable
created_by	VARCHAR	User who created the tag. This can be an email or user ID, based on your user management system.	Not Null
description	TEXT	Brief description or notes about the tag (Optional).	Nullable

- **Notes:**
  - The **created_by** field is based on the assumption that you may have user management or authentication in place. If not, it can be omitted.
  - Timestamp fields like **created_at** and **updated_at** help in tracking the creation and modification of tag.

Case Table

Column Name	Data Type	Description	Constraints
id	INT	Unique identifier for each case.	Primary Key, Auto-Increment
case_name	VARCHAR	Name of the case (e.g., "Country A Coffee Production Comparison").	Not Null
date	DATE	Date associated with the case.	Not Null
year	INT	Year in the case timeline (from the dropdown).	Not Null
country	INT	Country of focus for the case.	Foreign Key (References Country Table)
focus_commodity_id	INT	Identifier for the commodity (focus). This could reference a master data table for commodities.	Foreign Key (References Commodity Master Table)
currency	VARCHAR	Chosen currency for the case.	Not Null
area_size_unit	VARCHAR	Unit for area size (e.g., hectare, m^2, are).	Not Null

Column Name	Data Type	Description	Constraints
volume_measurement_unit	VARCHAR	Unit for volume measurement.	Not Null
cost_of_production_unit	VARCHAR	Unit for cost of production measurement (Per-area/Per-yield/Per-tree).	Not Null
reporting_period	VARCHAR	Reporting period (Per-season/Per-Year).	Not Null
segmentation	BOOLEAN	Indicates if segmentation is on/off.	Not Null, Default False
living_income_study	ENUM	Better Income or Living Income	Nullable
multiple_commodities	BOOLEAN	Indicates if multiple commodities option is on/off.	Not Null, Default False
logo	VARCHAR	Stores the uploaded logo for the case.	Nullable
private	BOOLEAN	Indicates if the case is private or public	Not Null, Default False
created_at	DATETIME	Timestamp when the case was created.	Not Null, Default Current Timestamp
updated_at	DATETIME	Timestamp when the case was last updated.	Nullable
created_by	VARCHAR	User who created the case. This can be an email or user ID, based on your user management.	Not Null

- **Notes:**
  - The **focus_commodity_id** assumes that there is a separate master data table for commodities.
  - When the case is marked as **private**, user cannot see this case unless:
    - The project owner (**created_by**) invite other user to the case
    - The user role is **business unit admin, business unit editor** or **super admin**
  - The **logo** column for the file path or URL.
  - In delete user workflow API, we need to check whether they have any cases or no
    - **SHOULD NOT delete cascade**
  - Case owner can be changed by **BU Admin / Super Admin**

Case Tags Table

Column Name	Data Type	Description	Constraint
id	INT	Unique Identifier for each Diversified Commodity	Primary Key, Auto-Increment
case_id	INT	Identifier for the case.	Foreign Key (References Case Table)
tag_id	INT	Identifier for the tag.	Foreign Key (References Tag Table)

- **Unique Constraint:** case_id + tag_id
- **Notes:**
  - Many to many: **cases <> tags**

User Tags Table

Column Name	Data Type	Description	Constraint
id	INT	Unique Identifier for each Diversified Commodity	Primary Key, Auto-Increment
user_id	INT	Identifier for the user.	Foreign Key (References User Table)
tag_id	INT	Identifier for the tag.	Foreign Key (References Tag Table)

- **Unique Constraint:** user_id + tag_id
- **Notes:**
  - Many to many: **users <> tags**
- **NOT REQUIRED**

Case Commodity Table

Column Name	Data Type	Description	Constraint
id	INT	Unique Identifier for each Case Commodity	Primary Key, Auto-Increment
case_id	INT	Identifier for the case to which the case commodity belongs.	Foreign Key (References Case Table)
commodity_id	INT	Identifier for the commodity. This could reference a master data table for commodities.	Foreign Key (References Commodity Table)
focus_commodity	BOOLEAN	Whether it's focus commodity	True, Default: False
breakdown	BOOLEAN	To define the question list	Not Null

- **Unique Constraint:** case_id, commodity_id
- **Notes:**
  - IF breakdown is FALSE, then show only question who has null parent and it's children, so grand children will not show.
  - When commodity_id is not equal to focus_commodity_id in the case table, this means the commodity is a diversified income

Segment Table

Column Name	Data Type	Description	Constraints
id	INT	Unique identifier for each segment.	Primary Key, Auto-Increment
case_id	INT	Identifier for the case to which the segment belongs.	Foreign Key (References Case Table)
segment_name	VARCHAR	Name or label of the segment (e.g., "City A").	Not Null
target	DECIMAL		Nullable
house_hold_size	DECIMAL	Average Household size: for calculating living income benchmark	Nullable

- **Unique Constraint:** case_id, segment_name

Question Table

Column Name	Data Type	Description	Constraints
id	INT	Unique identifier for each question.	Primary Key, Auto-Increment
parent_id	INT	Identifier for the parent question. NULL for top-level questions.	Foreign Key (References question_id), Nullable
code	VARCHAR	Code for the question	Nullable
question_text	VARCHAR	Text of the question or breakdown label.	Not Null
descriptions	TEXT	Description text of the question	Nullable
default_value	TEXT	JavaScript string used for front-end calculations.	Nullable
case_commodity_id	INT	Foreign Key	Nullable
created_by	INT	Identifier for the user to which the question belongs.	Foreign Key (References to user_id), Nullable

- **Notes:**
  - The **default_value** column is designed to store JavaScript strings that dictate the front-end calculation logic for each question. This approach provides dynamic flexibility, enabling the system to compute varied calculations based on user inputs for child questions. By embedding the logic directly within the database, the application can adapt to different calculation requirements without necessitating hard-coded logic or frequent code-base updates. However, it's important to note that

while the system facilitates automated calculations through these scripts, users retain the option to manually input values for parent questions.
    - This manual input capability ensures that if users choose not to provide detailed breakdown values, they can still directly specify the desired value for the overarching question.

> Case commodity Id only used for custom added specific question

> If the default value is Null, then the children 1 level below the question should be added up as SUM.

> Parent ID Null is the Income Drivers. Purpose: Selectively fetch values of Income Drivers for the visualization

Commodity Category Question

Column Name	Data Type	Description	Constraints
id	INT	Unique identifier for commodity question.	Primary Key, Auto-Increment
commodity_category_id	INT	Identifier for the commodity category.	Foreign Key (References to commodity_category_id)
question_id	INT	Identifier for the question_id.	Foreign Key (References to question_id)

- **Unique Constraint:** commodity_category_id, question_id
- **Notes:**
    - Commodity category Question define the list of question that will show in the case based on the selected commodity. Commodity -> Commodity Category -> Question.
    - Commodity Category will initially seeded via CLI

Visualization Table

Column Name	Data Type	Description	Constraints
id	INT	Unique identifier for each answer.	Primary Key, Auto-Increment
case_id	INT	Identifier for the case to which the answer belongs.	Foreign Key (References Case Table)

Column Name	Data Type	Description	Constraints
segment_id	INT	Identifier for the segment to which the answer belongs.	Foreign Key (References Segment Table)
tab	ENUM	Enum for which tab this configuration belongs to	Enum (sensitivity_analysis,scenario_modeling)
config	JSONB	Identifier for the segment to which the answer belongs.	JSONB

- **Unique Constraint:** segment_id, tab

Segment Answer Table

Column Name	Data Type	Description	Constraints
id	INT	Unique identifier for each answer.	Primary Key, Auto-Increment
case_commodity_id	INT	Unique Identifier for each commodity case commodity	Foreign Key (References Case Case Commodity Table)
segment_id	INT	Identifier for the segment to which the answer belongs.	Foreign Key (References Segment Table)
question_id	INT	Identifier for the question being answered.	Foreign Key (References Question Table)
current_value	DECIMAL	The response or current value given to the question.	Not Null
feasible_value	DECIMAL	The response or feasible value given to the question.	Nullable, Default: Current

- **Unique Constraint:** case_commodity_id, segment_id, question_id
- **Notes:**
  - If the feasible value is not defined, then store current value to feasible value.

Living Income Benchmark Table

Column Name	Data Type	Description	Constraints
id	INT	Unique identifier.	Primary Key, Auto-Increment
country_id	INT	Identifier for the country.	Foreign Key (References country_id)

Column Name	Data Type	Description	Constraints
currency_id	INT	Identifier for the currency.	Foreign Key (References currency_id)
year	INT	Year of the living income the benchmark.	Not Null
value	DECIMAL	The benchmark value.	Not Null

- **Unique Constraint:** country_id, segment_id, question_id
- **Notes:**
  - Living Income Benchmark will initially seeded via CLI

Reference Data Table

Column Name	Data Type	Description	Constraints
id	INT	Unique identifier.	Primary Key, Auto-Increment
country_id	INT	Identifier for the country.	Foreign Key (References country_id)
commodity_id	INT	Identifier for the commodity.	Foreign Key (References commodity_id)
type	ENUM	Baseline / Segment Average	Enum
year	INT	Year of the living income the benchmark.	Nullable
farm_size	DECIMAL	The Farm Land Size in Hectare	Nullable
farmer_production	DECIMAL	Farmer Production in KG	Nullable
farmgate_price	DECIMAL	Farm Gate Price USD / KG	Nullable
farmer_expenses	DECIMAL	Cost of Production in USD	Nullable
diversified_income	DECIMAL	Secondary or Tertiary Income	Nullable

- **Notes:**
  - Reference Data Table will initially seeded via CLI
  - Initial Data is here: **PROVIDE LINK TO DRIVE**

API Endpoints

Endpoint Prefix: **/api**

1. Users Endpoint

1.1. User Login

- Endpoint: **/user/login**
- Method: **POST**
- Headers:
  - accept: application/json
  - Content-Type: application/x-www-form-urlencoded
- Request Body:
  - grant_type: "password"
  - username: string
  - password: password
  - scope: "openid email"
  - client_id: client_id
  - client_secret: client_secret
- Authorization: None

Example Response


```
{
  "access_token":
"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJlbWFpbCI6ImRlZGVuQGFrdm8ub3JnIiwiZXhwIjoxNjk2MzcwMTI0
fQ.4KR4ELzTpsvlBG3fJkA_MLJS9RMMVoY8X-eTeN9WXwk",
  "token_type": "bearer",
  "user": {
    "id": 1,
    "fullname": "Deden Bangkit",
    "email": "deden@akvo.org",
    "active": true,
    "role": "super admin, admin, editor, viewer, user",
    "business_unit_detail": [{
      "id":1,
      "name": "Coffee Producers",
      "role" "admin / member"
    }],
    "organisation_detail": {
      "id": 1,
      "name": "Akvo"
    }
    "tags_count": 2,
    "cases_count": 3
```

```
  }
}
```

1.2. Logged-in User Info

- Endpoint: **/user/me**
- Method: GET
- Headers:
- Request Body: None
- Query Parameter: None
- Authorization: Bearer Token - Any Access

Example Response

```
{
  "id": 1,
  "fullname": "Deden Bangkit",
  "email": "deden@akvo.org",
  "active": true,
  "role": "super admin / admin / editor / viewer / user",
  "business_unit_detail": [{
      "id":1,
      "name": "Coffee Producers",
      "role": "admin / member"
  }],
  "organisation_detail": {
    "id": 1,
    "name": "Akvo"
  },
  "tags_count": 2,
  "cases_count": 3
}
```

1.3. List of Users

- Endpoint: **/user**
- Method: **GET**
- Headers:
  - accept: application/json
  - Content-Type: application/x-www-form-urlencoded
- Request Body: None
- Query Parameter:
  - page: integer

    - limit: integer
    - search: string
    - organisation: organisation_id
- Authorization: Bearer Token - Admin Access

Example Response


```
{
  "current": 1,
  "data": [
    {
      "id": 0,
      "organisation": 0,
      "email": "string",
      "fullname": "string",
      "active": true,
      "role": "super admin / admin / editor / viewer / user",
      "tags_count": 2,
      "cases_count": 3
    }
  ],
  "total": 1,
  "total_page": 1
}
```


1.4. User Detail

- Endpoint: **/user/<user_id>**
- Method: **GET**
- Path Variable: **user_id**
- Headers:
    - accept: application/json
- Request Body: None
- Query Parameter: None
- Authorization: Bearer Token - Admin Access

Example Response


```
{
  "id": 1,
  "fullname": "Deden Bangkit",
  "email": "deden@akvo.org",
```

```
  "active": true,
  "role": "super admin / admin / editor / viewer / user",
  "organisation_detail": {
    "id": 1,
    "name": "Akvo"
  },
  "tag_ids": [1],
  "cases_ids": [1],
  "business_unit_detail": [{
    "id":1,
    "name": "Coffee Producers",
    "role" "admin / member"
  }],
}
```

1.4. User Registration

- Endpoint: **/user**
- Method: **POST**
- Headers:
  - accept: application/json
  - Content-Type: application/x-www-form-urlencoded
- Request Body:
  - fullname: string
  - organisation: organisation_id
  - business_units: business_unit_id
- Query Parameter: None
- Authorization: None

1.6. Register Password

- Endpoint: **/user/invitation/<invitation_id>**
- Method: **POST**
- Path Variable: **invitation_id**
- Headers:
  - accept: application/json
  - Content-Type: application/x-www-form-urlencoded
- Request Body:
  - password: password
- Query Parameter: None
- Authorization: None

1.5. Edit User

- Endpoint: **/user/<user_id>**
- Path Variable: **user_id**
- Method: **PUT**
- Headers:
  - accept: application/json
- Request Body:
  - fullname: string
  - password: string
  - organisation: organisation_id
  - business_unit: business_unit_id
  - role: enum (super admin, admin, viewer, editor, user)
  - cases: array of object
    - case_id
    - permission: enum (edit, view)
  - tags: array of object
    - tag_id
    - permission: enum (edit, view)
- Query Parameter: None
- Authorization:
  - Bearer Token - Admin Access: All Request
  - Bearer Token - User Access: All except case_id, tags_id, active

1.6. Delete User

- Endpoint: **/user/<user_id>**
- Path Variable: **user_id**
- Method: **DELETE**
- Headers:
  - accept: application/json
- Request Body: None
- Query Parameter: None
- Authorization:
  - Bearer Token - Admin Access

2. Tags Endpoints

2.1. List of Tags

- Endpoint: **/tag**
- Method: **GET**
- Headers:
  - accept: application/json
- Request Body: None
- Query Parameter:
  - page: integer

    - limit: integer
    - search: string
- Authorization: Bearer Token

Example Response

```
{
  "current": 1,
  "data": [
    {
      "id": 1,
      "name": "Tag Name",
      "description": "Tag Description",
      "cases_count": 5
    },
    ...
  ],
  "total": 10,
  "total_page": 1
}
```

2.2. Create Tag

- Endpoint: **/tag**
- Method: **POST**
- Headers:
    - accept: application/json
    - Content-Type: application/json
- Request Body:
    - name: string
    - description: string
    - cases: list of case_id
- Authorization: Bearer Token - Admin Access

2.3. Tag Detail

- Endpoint: **/tag/<tag_id>**
- Path Variable: **tag_id**
- Method: **GET**
- Headers:
    - accept: application/json
- Request Body: None
- Authorization: Bearer Token

Example Response

```
{
  "id": 1,
  "name": "Tag Name",
  "description": "Tag Description",
  "cases_count": 5
}
```

2.4. Edit Tag

- Endpoint: **/tag/<tag_id>**
- Path Variable: **tag_id**
- Method: **PUT**
- Headers:
  - accept: application/json
  - Content-Type: application/json
- Request Body:
  - name: string (nullable)
  - description: string (nullable)
  - cases: list of case_id
- Authorization: Bearer Token - Admin Access

3. Case Endpoints

3.1. List Cases

- Endpoint: **/case**
- Method: **GET**
- Headers:
  - accept: application/json
- Request Body: None
- Query Parameter:
  - page: integer
  - limit: integer
  - search: string
  - tags: single / multiple tags_id
  - focus_commodity: single / multiple focus_commodity_id
- Authorization: Bearer Token - Any

Example Response

```
{
  "current": 1,
```

```
  "data": [
    {
      "id": 1,
      "case_name": "Country A Coffee Production Comparison",
      "country": 1,
      "focus_commodity_id": 1,
      "diversified_commodities_count": 2,
      "created_at": "2023-01-01T00:00:00",
      "created_by": "admin@example.com"
    },
    ...
  ],
  "total": 10,
  "total_page": 1
}
```


3.2. Case Detail

- Endpoint: **/case/<case_id>**
- Path Variable: **case_id**
- Method: **GET**
- Headers:
  - accept: application/json
- Request Body: None
- Authorization: Bearer Token - Admin Access

Example Response


```
{
  "id": 1,
  "case_name": "Country A Coffee Production Comparison",
  "date": "2023-10-03",
  "year": 2023,
  "country": 1,
  "focus_commodity_id": 1,
  "currency": "USD",
  "area_size_unit": "hectare",
  "volume_measurement_unit": "liters",
  "cost_of_production_unit": "Per-area",
  "reporting_period": "Per-season",
  "segmentation": false,
  "living_income_study": "Better Income",
```

```
  "multiple_commodities": false,
  "created_at": "2023-01-01T00:00:00",
  "updated_at": null,
  "created_by": "admin@example.com",
  "segments": [{
    "id": 1,
    "name": "Coffee in Country A",
    "target": 300,000.00,
    "house_hold_size": 400.00
  },{
    "id": 1,
    "name": "Coffee in Country B",
    "target": 300,000.00,
    "house_hold_size": 400.00
  }],
  "case_commodities": [{
    "id": 1,
    "commodity_id": 1,
    "breakdown": true
  },{
    "id": 2,
    "commodity_id": 2,
    "breakdown": true
  }]
}
```


Notes:

- In above example, we have 2 **case_commodities** one has same **commodity_id** as **focus_commodity_id** and the other **case_commodities** has different **commodity_id** with the **focus_commodity_id** which consider as diversified income.

3.3. Add Case

- Endpoint: **/case**
- Method: **POST**
- Headers:
  - accept: application/json
  - Content-Type: application/json
- Request Body:
  - case_name: string
  - date: date
  - year: integer

    - country: integer
    - focus_commodity_id: commodity_id
    - currency: string
    - area_size_unit: string
    - volume_measurement_unit: string
    - cost_of_production_unit: string
    - reporting_period: string
    - segmentation: boolean
    - living_income_study: string (nullable)
    - multiple_commodities: boolean
    - created_by: string
    - other_commodities: array of objects or null
        - commodity_id: commodity_id
        - breakdown: boolean
- Authorization: Bearer Token - Admin Access

Notes:

- If other commodities is null, we still store a row in **case_commodities** table which stores **commodity_id** from the **focus_commodity_id** and **breakdown** by default is **true.** So if we have 1 **other_commodities**, then we store 2 **case_commodities**

3.4. Edit Case

- Endpoint: **/case/<case_id>**
- Path Variable: **case_id**
- Method: **PUT**
- Headers:
    - accept: application/json
    - Content-Type: application/json
- Request Body:
    - case_name: string
    - date: date
    - year: integer
    - country: integer
    - focus_commodity_id: commodity_id
    - currency: string
    - area_size_unit: string
    - volume_measurement_unit: string
    - cost_of_production_unit: string
    - reporting_period: string
    - segmentation: boolean
    - living_income_study: string (nullable)
    - multiple_commodities: boolean
    - other_commodities: array of objects
        - commodity_id: commodity_id

    - breakdown: boolean
- Authorization: Bearer Token - Admin Access

Notes:

- What if we remove the **other_commodities** from case which has existing question value?

4. Segment Endpoint

4.1. Add Segments (single / multiple)

- Endpoint: **/segment**
- Method: **POST**
- Headers:
    - accept: application/json
    - Content-Type: application/json
- Request Body:
    - segments: array of objects
        - case_id: integer
        - segment_name: string
        - target: decimal (nullable)
        - house_hold_size: decimal (nullable)
- Authorization: Bearer Token

5. Question Endpoint

5.1. Get Questions By Commodity ID

- Endpoint: **/questions**
- Path Variable: **commodity_id**
- Method: GET
- Headers:
    - accept: application/json
- Request Body:
    - question_groups: list of objects
        - commodity_id: integer
        - breakdown: boolean
- Authorization: Bearer Token

Example Response

```
[{
  "commodity_id": 1,
  "commodity_name": "Rice",
```

```
  "focus_commodity": true,
  "questions": [{
    "id": 1,
    "parent_id": null,
    "code": "Q1",
    "question_text": "Net Income per day",
    "descriptions": null,
    "default_value": "function() { return #Q2 * #Q3 / 30; }",
    "created_by": 1
  },{
    "id": 2,
    "parent_id": null,
    "code": "Q2",
    "question_text": "Income from Commodity / Month",
    "descriptions": null,
    "default_value": null,
    "created_by": 1
  },{
    "id": 3,
    "parent_id": null,
    "code": "Q3",
    "question_text": "Cost of Production / Month",
    "descriptions": null,
    "default_value": null,
    "created_by": 1
  }]
},{
  "commodity_id": 2,
  "commodity_name": "Chicken"
  "focus_commodity": false,
  "questions": [{
    "id": 1,
    "parent_id": null,
    "code": "Q1",
    "question_text": "Net Income per day",
    "descriptions": null,
    "default_value": "function() { return #Q2 * #Q3 / 30; }",
    "created_by": 1
  }]
}]
```

Note:

- List Question is defined by commodity_id and breakdown parameter
- The list of Question Group is **focus_commodity_id + list_of_commodity_ids_in_diversified_income**

6. Segment Value Endpoint

6.1. Add / Update Segment Value

- Endpoint: **/segment-answer/<segment_id>**
- Method: **POST**
- Headers:
  - accept: application/json
  - Content-Type: application/json
- Request Body:
  - value: array of objects
    - case_commodity_id: case_commodity_id
    - segment_id: segment_id
    - question_id: question_id
    - current_value: float
    - feasible_value: float
- Authorization: Bearer Token

7. Visualization Endpoint

7.1. Get Visualization Config

- Endpoint: **/segment-answer/<case_id>**
- Method: **GET**
- Headers:
  - accept: application/json
  - Content-Type: application/json
- Request Body: None
- Authorization: Bearer Token

Example Response

7.2. Add / Update Visualization Config

- Endpoint: **/segment-answer/<segment_id>**
- Method: **POST**

- Headers:
  - accept: application/json
  - Content-Type: application/json
- Request Body:
  - tab: Enum tab
  - config: objects (any key)
- Authorization: Bearer Token

Error Handling & Validation

- Validate all form inputs to ensure they meet the expected criteria.
- Provide feedback for invalid inputs or missing information.
- Handle potential calculation errors during segment income computations.

Additional Considerations

- Given that there's an upload logo functionality, we need to consider storage solutions for media files.
- Since there's many tags for cases, there should be a way to navigate, search, and manage these cases.
- For the dashboard tabs, we'd need more specific requirements to detail out the metrics and visualizations that would be displayed on each tab.
- Since there's a lot of master data, we should also provide a CLI that operate the data seeder in one shot instead CLI seeder for individual master data

---

Revision #80
Created 2023-09-19 05:35:03 UTC by Deden Bangkit
Updated 2024-07-02 04:11:58 UTC by Galih Pratama