Moksha Kachhia

Overview

Each **user**, identified by their *User UID*, has the following sections:
- name: contains their first and last name, separately
- country: the country they live in
- annual_answers: the answers to their annual survey
- monthly_answers: saved by year-month to track electricity, water & gas CO2E
- daily_answers: saved by year-month-day to track their daily survey answers

Outside of the user, there are X **concrete data sets**:
1. car_type_factor: The car to CO2E kg/km factor for each type of car
2. car_distance: The number of km used for the calculation based on the range
3. public_transit_footprint: the CO2E kg produced by each combination of PT
4. short_flight_footprint: the CO2E kg produced by each number of short flights
5. long_flight_footprint: the CO2E kg produced by each number of long flights
6. avg_annual_emisions: The average per-capita annual carbon footprint for 231 countries

Using realtime DB:

If you don't add concrete values to a key-value pair, the key will not show up in the live db.

Eg. if you add a key named "annual_answers" with an empty hashmap as it's value, the "annual_answers" key will not show up in the live DB.

Concept of a **Unique Key**

Some sections have a unique carbon footprint for each possible answer permutation, this answer is stored in the respective concrete data section of the db.

In the app, we must do create and use a unique key to retrieve the permutation-specific CO2E data.

Example on the next page of why this helps and how to use it <3

For the annual survey, to calculate the housing footprint, we have 6 fields:

1. structure_type (detached, semi-detached, townhouse, condo_apartment, other),
2. size (0-1000 sqft, 1000-2000 sqft., 2000+)
3. household_members (1, 2-3, 4-5, 5+)
4. heating_source (natural gas, electricity, oil, propane, wood, other)
5. bill (0-50, 50-100, 100-150, 150-200, 200+)
6. water_source (if water_source != heating_source add 233 to the housing footprint)

So, we need to store these fields for each user when entering their form data to make their uniqueKey as such in Java:

```
//Store the user answers
String structureType = "detached";
String size = "1000-2000";
String householdMembers = "4-5";
String heatingSource = "natural_gas";
String billing = "100-150";

// Combine responses into a single key
String uniqueKey = structureType + "|" + size + "|" + householdMembers + "|" +
heatingSource + "|" + billing;

// Save the key in Firebase
DatabaseReference databaseRef = FirebaseDatabase.getInstance().getReference();
databaseRef.child("carbon_footprint_by_response").child(uniqueKey).setValue(carbonF
ootprintValue);
```

Tracking Annual Answers

Transportation

<u>User Schema:</u>

```
"transportation":{
        "uses_car": false,
        "car_type": "none",
        "distance": "0",
        "public_transit_use": "occasionally",
        "hours_on_public_transit": "2-3",
        "num_short_flights": "0",
        "num_long_flights": "1",
        "transportation_co2e": 1634
        }
```

<u>Related Concrete Data:</u>

```
"car_type_factors":{
  "none":0,
  "gasoline":0.24,
  "diesel":0.27,
  "hybrid":0.16,
  "electric":0.05,
  "idk":0.20
},

"car_distance":{
  "0":0,
  "5,000":5000,
  "5,000–10,000":10000,
  "10,000–15,000":15000,
  "15,000-20,000":20000,
  "20,000–25,000":25000,
  "25,000+":35000,
},

"short_flight_footprint":{
  "0": 0,
  "1-2": 225,
  "3-5": 600,
  "6-10": 1200,
  "10+": 1800 }
```

```
"public_transit_footprint":{
  "never|0": 0,
  "occasionally|1": 246,
  "occasionally|2-3": 819,
  "occasionally|4-5": 1638,
  "occasionally|6-10":3071,
  "occasionally|10+": 4095,
  "frequently|1": 573,
  "frequently|2-3": 1911,
  "frequently|4-5": 3822,
  "frequently|6-10":7166,
  "frequently|10+": 9555,
  "always|1": 573,
  "always|2-3": 1911,
  "always|4-5": 3822,
  "always|6-10":7166,
  "always|10+": 9555
},

"long_flight_footprint":{
  "0": 0,
  "1-2": 825,
  "3-5": 2200,
  "6-10": 4400,
  "10+": 6600 }
```

Personal Vehicle:

Does not have car:
- uses_car = false
- car_type = "none"
- distance = "0"
- The only no-car value should always be 0

Has car:
- uses_car = true
- Enter the car type for car_type
- Enter the distance for the distance

  * IF THE USER ANSWERS I DONT KNOW CAR TYPE IS "idk"

  Example: User drives a hybrid car 15,000–20,000 km:
  "uses_car": true,
  "car_type": "hybrid",
  "distance": "15,000-20,000",

Public Transit: *MUST CREATE A UNIQUE KEY IN FORM* "<use>|<hours>" *eg:* "always|10+"

Does not use public transit:
- public_transit_use = "never"
- hours_on_public_transit = "0"

Uses public transit:
- Enter the frequency type for public_transit_use
- Enter the duration for hours_on_public_transit

Short Flights and Long Flights:
- Enter the number of times as a STRING under num_short_flights and num_long_flights

Total CO2E

For car, find the associated car_type_factors and car_distance numerical values and multiply them

For public transit, look up it's unique key under the public_transit_footprint

For the flights, look up the flight duration string under short_flight_footprint

Add these three values up and put them into the user's transportation_co2e.

Food

User schema:

```
"food": {
        "diet_type":"vegan",
        "beef_frequency":"never",
        "pork_frequency":"never",
        "chicken_frequency":"never",
        "fish_frequency":"never",
        "food_waste_frequency":"rarely",
        "food_co2e":523.4
    }
```

Associated Concrete data:

```
"diet_type_emisions":{
   "vegan": 500,
   "vegetarian": 1000,
   "pescatarian":1500,
   "meat_based":0
},

"beef_emisions":{
   "daily": 2500,
   "frequently": 1900,
   "occasionally": 1300,
   "never": 0
},
"pork_emisions":{
   "daily": 1450,
   "frequently": 860,
   "occasionally": 450,
   "never": 0
},
```

```
"chicken_emisions":{
   "daily": 950,
   "frequently": 600,
   "occasionally": 200,
   "never": 0
},
"fish_emisions":{
   "daily": 800,
   "frequently": 500,
   "occasionally": 150,
   "never": 0
},

"food_waste_emisions":{
   "never": 0,
   "rarely": 23.4,
   "occasionally": 70.2,
   "frequently": 140.4
},
```

Diet_type: enter their diet type (meat_based for the non vegan/veg/pesc)

X_frequency: the frequency of X (beef, pork, chicken, seafood)

food_waste_frequency: how frequently they waste food

Total CO2E

Add up the three categories' respective emissions values do not require any unique keys

Housing

User schema:

    "housing": {
            "structure_type": "detached",
            "size": "1000-2000",
            "household_members": "4-5",
            "heating_source":
 "natural_gas",
            "bill": "100-150",
            "water_source":"natural_gas",
            "housing_co2e":0
        },

Associated Concrete data:
"housing_emisions":{
   ""
 },

- structure_type (detached, semi-detached, townhouse, condo_apartment, other),
- size (0-1000 sqft, 1000-2000 sqft., 2000+ sqft.)
- household_members (1, 2-3, 4-5, 5+)
- heating_source (natural gas, electricity, oil, propane, wood, other)
- bill (0-50, 50-100, 100-150, 150-200, 200 +)
- water_source (if water_source != heating_source, AND water source is electric/solar reduce total by 233, if they are not equal and water source is natural gas, oil, pro

*A Unique Key is REQUIRED in the form
<structure_type> | <size>|<household_members>|<heatingSource>|bill

Total CO2E

Find the associated one for the unique key in the housing_emissions concrete data AND **add 233** if the **source of heating and water heating are not the same**

store it under housing_co2e

Consumption

User schema:
"consumption": {
    "clothing_frequency": "rarely",
    "eco_friendly_frequency": "occasionally",
    "device_frequency": "4+",
    "recycling_frequency": "always",
    "consumption_co2e": 841
  }

Associated Concrete data:

"clothing_emissions":{
  "monthly": 360,
  "quarterly": 120,
  "annually": 100,
  "rarely": 5
},

"clothing_eco_friendly_adjustment":{
  "no": 0,
  "occasionally": 0.3,
 "regularly": 0.50

},

 "clothing_recylcing_adjustment":{
  "never": 1,
  "occasionally": 0.15,
  "frequently":0.30,
  "regularly": 0.50
},

"device_emissions":{
 "none": 0,
 "1": 300,
 "2": 600,
 "3": 900,
 "4+":1200
},

"device_recylcing_adjustment":{
 "never": 1,
 "occasionally": 0.1,
 "frequently":0.2,
 "regularly": 0.3
},

Adjustments:

To get the total clothing emissions, take the clothing emissions and subtract clothing emissions * the eco-friendly adjustment and subtract the value * the recycling adjustment and same for device and recycling

Eg: user buys clothes rarely, occasionally buys secondhand/eco friendly, buys more than 4 devices a year and always recycles then:

Clothing Co2e (starting = 5)

  - starting * eco-friendly reduction
  = 5 * 0.3 = 1.5

  - starting * recycling reduction
  = 5 * 0.5   = 2.5

  = 5 - 1.5 - 2.5 = **1.0**   ← total clothing

Device Co2e (starting = 1200)

  - starting * recycling reduction
  = 1200 * 0.3 = 360

  = 1200 - 360 = **840**   ← total device

Total CO2E

Sum the clothing and device co2e

Tracking Daily Answers

Monthly Bills

<u>User schema:</u>
```
"monthly_answers": {
     "2024-10": {
       "electricty_bill":100,
       "water_bill": 200,
       "gas_bill": 300,
       "utilities_co2e":0
     }
}
```

Store by date: form "<year>-<month>"
Store the different bills and the total co2e added from those 3 bills

For now, no concrete data sets will be added since we have no idea how they want us to calculate the co2e for the bills nor have they told us how it should affect anything

Transportation Activities

User Schema:
"2024-10-27": {
        "Transportation":{
          "driving_distance":0,
          "public_transit_type":"bus",
          "hours_on_public_transit":1,
          "hours_bike_walk":2,
          "num_short_flights": 0,
          "num_long_flights":0,
          "transportation_co2e":0.67
          },

Car travel CO2E:
        1. Grab the ratio related to their vehicle type and multiply the distance traveled in KILOMETERS eg. drives an electric vehicle 10 km today then the CO2E for it is 0.05*10 = 0.5

Public Transportation CO2E:
  1. create a UNIQUE KEY based on their annual public transit frequency and the range their inputted amount falls into. Eg. they said they annually use public transit occasionally and today they logged 3 hrs of transit, then their special key should be "occasionally|2-3"
  2. Divide the associated CO2E by 365, for this eg, the CO2E should be 819/365 = 2.24

Biking/walking:
        ???? just enter what they answer for now Idk how they want us to use this data T^T

Flights:
ideally give them these options for the number of flights: "0" , "1-2", "3-5" ,"6-10" ,"10+
Sum the associated long and or short flights, exactly how it is done in the annual calculation for CO2E

Total CO2E

Sum these four fields to enter into the "transportation_co2e" field

Food Consumption Activities

user schema:

"2024-10-27": {
        "meal1": {
        "type_of_meal": "plant-based",
        "servings": 2,
        "mealC02E": 4.16   },

        "meal2": {
        "type_of_meal": pork,
        "servings": 1,
        "mealCO2E": 2.64
        }
 },

associated concrete data:

"serving_emission":{
   "plant_based": 2.08,
   "fish": 2.64,
   "chicken": 2.22,
   "pork": 4.03,
   "beef": 6.95
 },

For now, we have the option for multiple meals since it doesn't specify if you need multiple or only 1

Total CO2E

The calculation of CO2E is directly the sum of the meals, where you multiply the servings by the serving emission

Shopping Activities

user schema:

```
"consumption":{
        "clothing": {
          "num_clothes": 50,
          "clothing_co2e": 50
        },
        "devices": {
          "device1": {
            "type_of_device": "plant-based",
            "quantity":2,
            "device_C02E": 0
            },

          "device2": {
            "type_of_meal": "fish",
            "quantity":1,
            "device_CO2E": 0
          }
        },
        "other": {
          "other1": {
            "category": "sofa",
            "quantity":2,
            "other_C02E": 6
          },

          "other2": {
            "category": "fridge",
            "quantity":1,
            "other_CO2E": 10
          }
        }
      }
    },
```

Total CO2E

??? don't know what is going on for this one ngl <3 but for now we have these values:

Clothing: 5 kg per piece bought so total clothing = 5 x # pieces bought
for now, all devices = 300 x # electronics , all other = literally make it up idk T^T