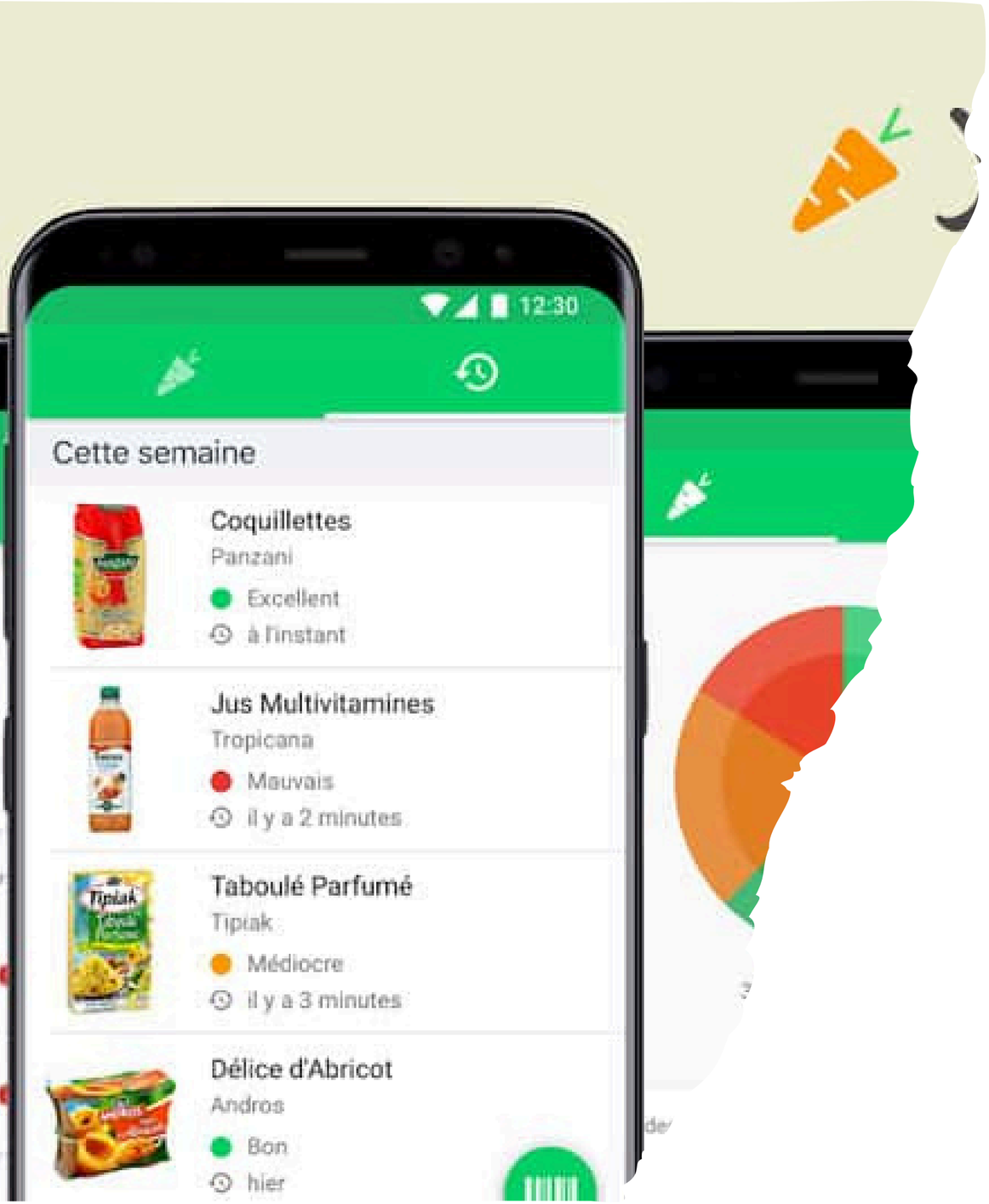
Ironhack

Project presentation

Carbon Foodprint

Valentin Perraud

CARBON
FOODPRINT



Business Case

Everyone needs to go buy groceries. What if you could make an impact while doing it ?

Consumers want to shop sustainably, but:

- No clear info on a product's CO_2 footprint
- No visibility into supply chain emissions
- No tool to track their impact over time

An interactive app (inspired by Yuka) that:

- Lets users build a shopping cart
- Shows CO_2 breakdown per product and stage
- Tracks carbon impact over time
- Encourages low-emission choices

Contents

01/
Data source

02/
Journey

03/
Demonstration

04/
Conclusions

01/
Data sources

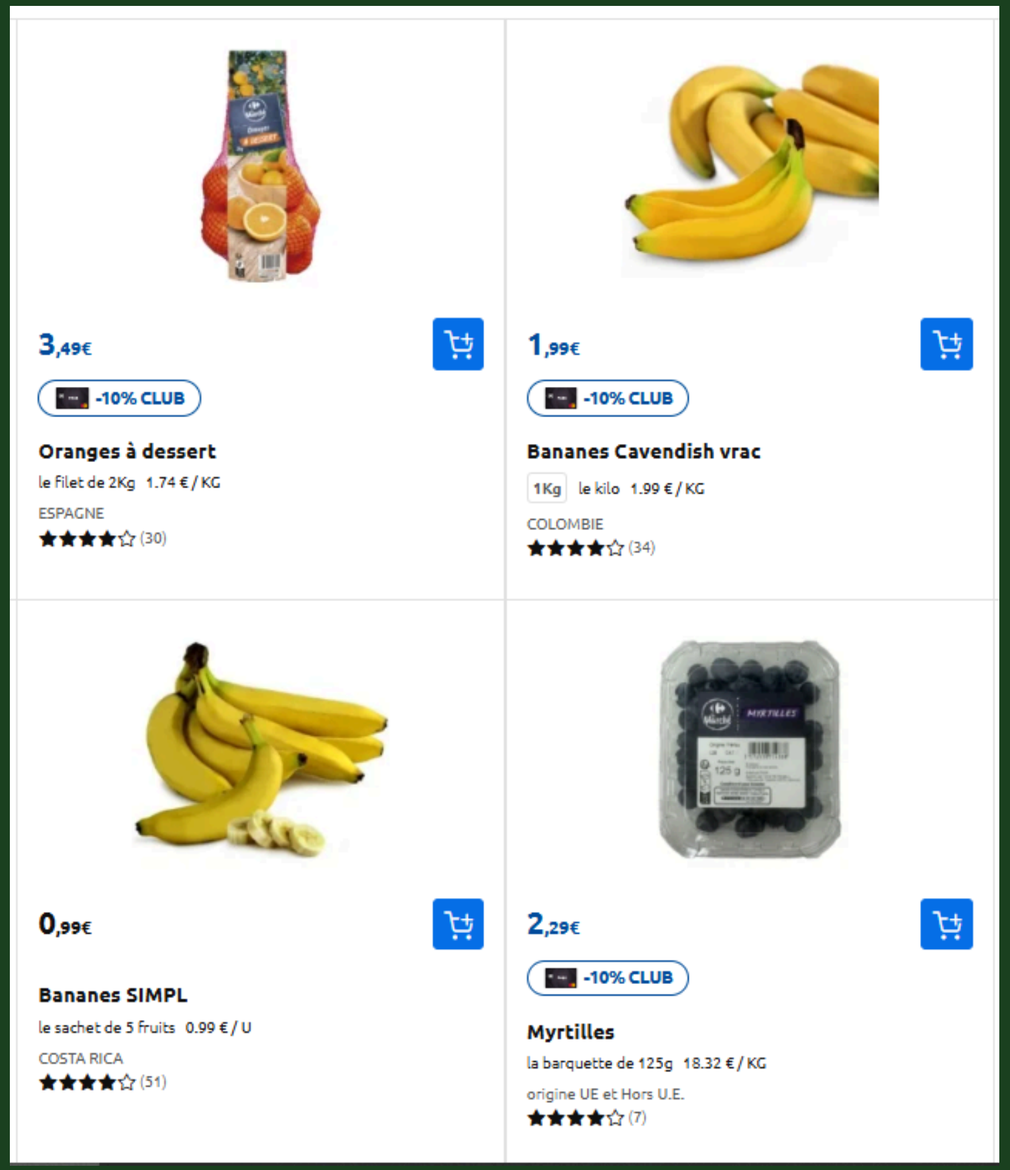
Carrefour (CSV):

- Network calls to get ~17000 products (VPN...)
  - Product_name
  - product_category
  - Price
  - Price/kg

Impact CO2 by ADEME (JSON with 150 entry):

- Footprint/kg
- Supply chain division:
  - Farming
  - Processing
  - Transport
  - Packaging
  - Retail
  - Consumption

01/
Data sources




```
{
    "name": "Poulet",
    "slug": "poulet",
    "ecv": 4.56043,
    "footprint": 4.56043,
    "footprintDetail": [
        {
            "id": 30,
            "value": 3
        },
        {
            "id": 31,
            "value": 0.83
        },
        {
            "id": 32,
            "value": 0.378
        },
        {
            "id": 33,
            "value": 0.305
        },
        {
            "id": 34,
            "value": 0.0406
        },
        {
            "id": 35,
            "value": 0.00683
        }
    ]
},
```

02/
Journey – Data Transformation

- Matching carrefour extract with ADEME table
  - Handle specific name : cacao → chocolat
  - *re.search(keyword, word, re.IGNORECASE) or ratio(keyword, word) > 0.85*
  - Remove null values

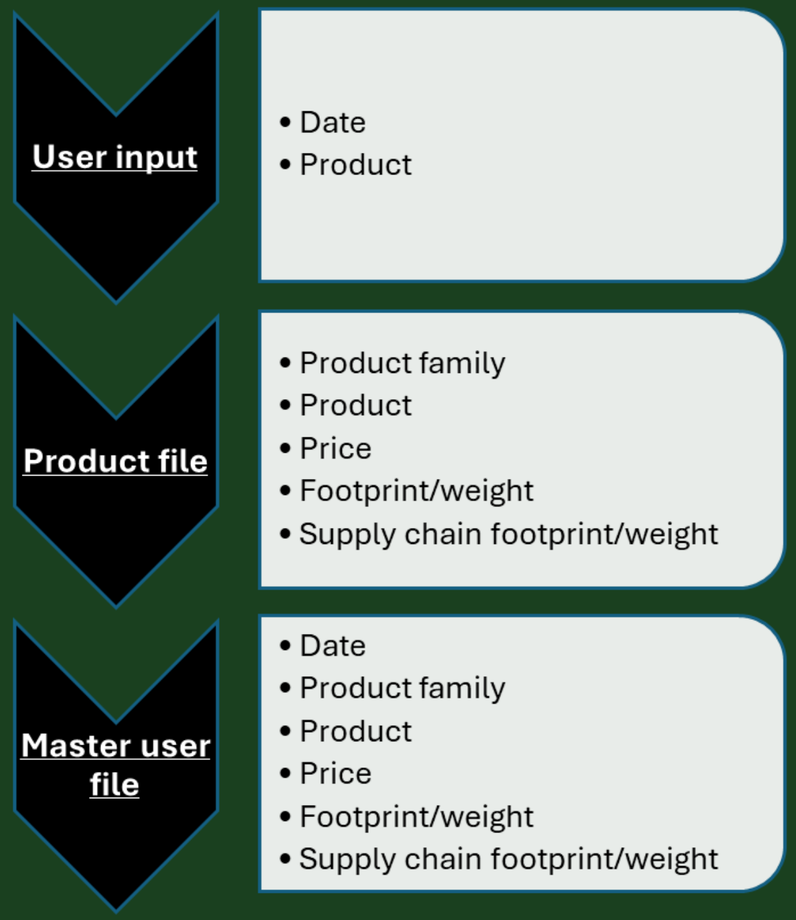
- Estimate product weight using the formula:
  - Weight = Price ÷ Price per kg
- Calculate carbon footprint by applying the estimated weight to footprint values per kg.
  - Apply it to supply chain division

- Use a random function to create data for the master_file
- Put all the tables in SQL

02/
Journey – User Input

- **Chat GPT API:**
  - Upload a photo of your receipt.
  - ChatGPT extracts product titles and formats them into structured data.
  - Prompt :
    - *This is a scanned receipt. Extract only the list of products under the "Description" section. If there are values on the QTE x P U column, repeat the line above in "Description". Ignore prices, totals, and other metadata. Return the product names in a JSON format*
  - Products are matched against the database using fuzzy logic.
  - Matched products are added directly to the master_data table in SQL.
- **Cart :**
  - User selects products from a dropdown menu (based on product table).
  - Selected items are added to a session-based cart.
  - "Submit Cart to your profile"
    - selected items are stored in master_data in SQL.

Workflow:



```
of daily groceries

per week with supply chain percentage

 total_co2_kg) * 100, 2), '%') AS farming_percentage,
g / total_co2_kg) * 100, 2), '%') AS processing_percentage,
 / total_co2_kg) * 100, 2), '%') AS packaging_percentage,
 / total_co2_kg) * 100, 2), '%') AS transport_percentage,
total_co2_kg) * 100, 2), '%') AS retail_percentage,
on / total_co2_kg) * 100, 2), '%') AS consumption_percenta

te, '%d/%m/%Y')) AS year,
te, '%d/%m/%Y'), 1) AS week,
2) AS total_co2_kg,
rming) AS farming,
ocessing) AS processing,
ckaging) AS packaging,
ansport) AS transport,
tail) AS retail,
nsumption) AS consumption
```

03/ Application

```
of daily groceries

per week with supply chain percentage

 total_co2_kg) * 100, 2), '%') AS farming_percentage,
g / total_co2_kg) * 100, 2), '%') AS processing_percentage,
 / total_co2_kg) * 100, 2), '%') AS packaging_percentage,
 / total_co2_kg) * 100, 2), '%') AS transport_percentage,
total_co2_kg) * 100, 2), '%') AS retail_percentage,
on / total_co2_kg) * 100, 2), '%') AS consumption_percenta

te, '%d/%m/%Y')) AS year,
te, '%d/%m/%Y'), 1) AS week,
2) AS total_co2_kg,
rming) AS farming,
ocessing) AS processing,
ckaging) AS packaging,
ansport) AS transport,
tail) AS retail,
nsumption) AS consumption
```

03/ Application

Summary:

- Backend in SQL, app in Streamlit and diagrams in plotly
- Filters:
  - Week/Month/Year
  - Specific period seletion
- User input:
  - ChatGPT API
  - Cart
- Dashboard:
  - Period specific diagrams
  - Trend diagrams
- Advices on what actions to take:
  - Category with the highest impact

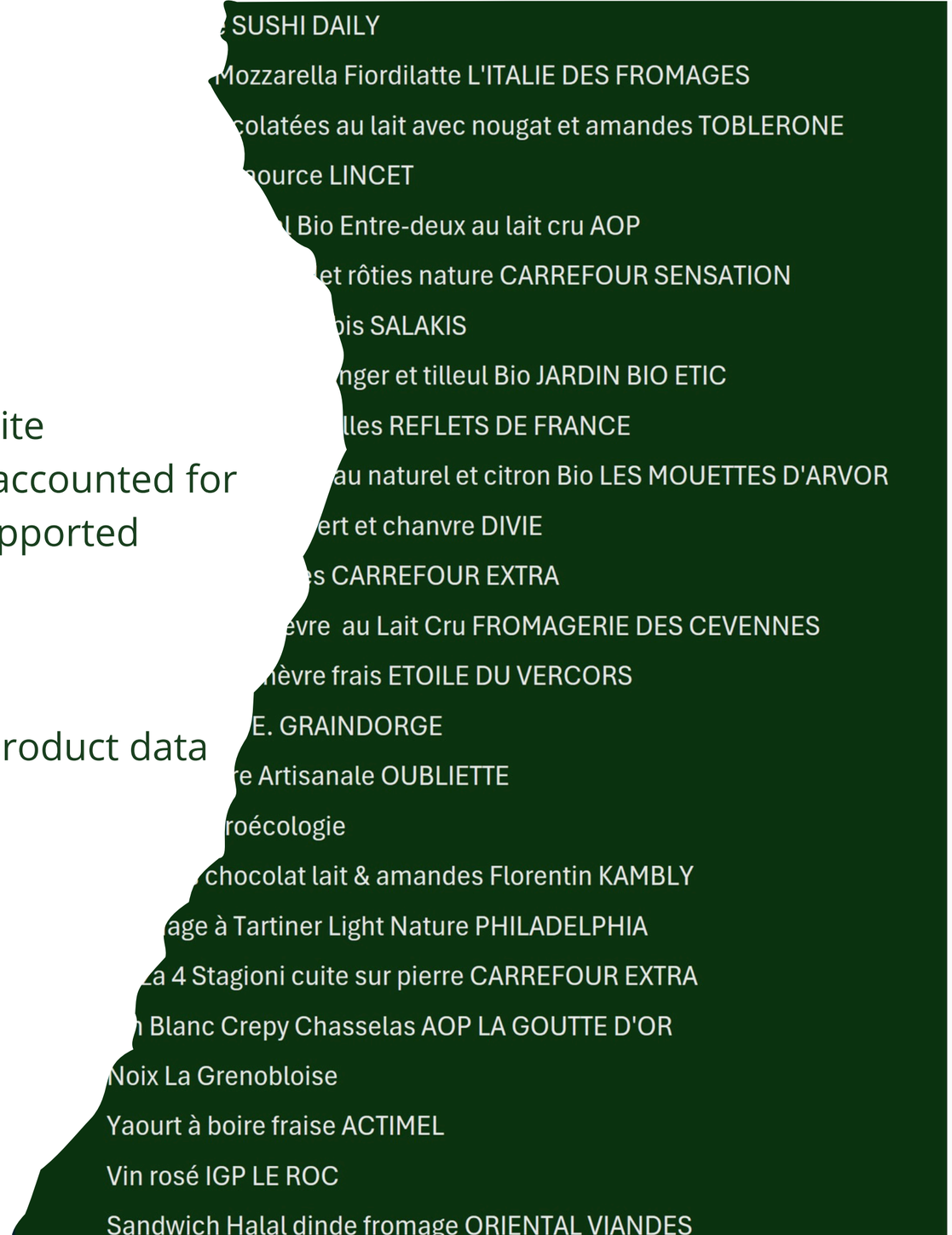
04/
Conclusion

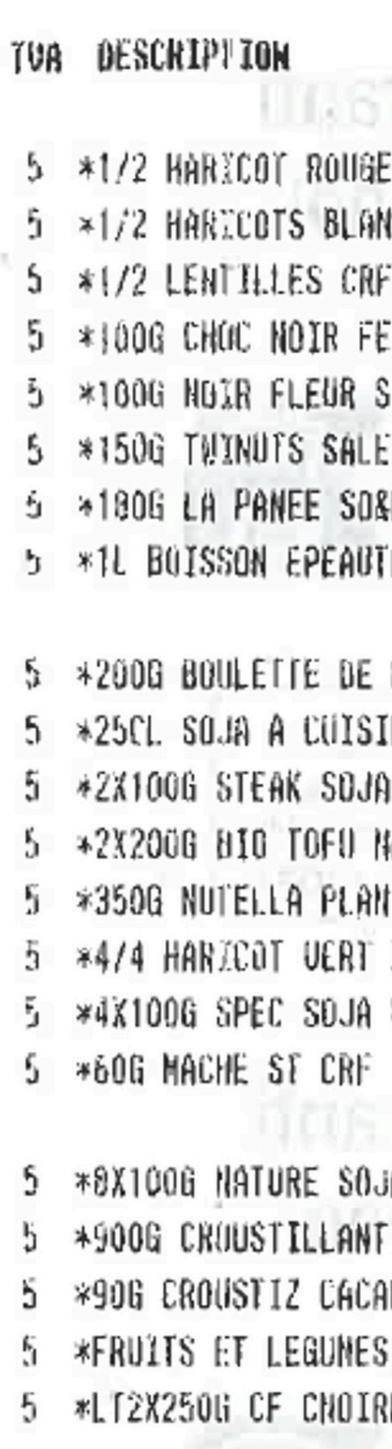
Limitations:

- Hardware difficulties
- Receipt text format differs from website
- Manual quantities (e.g. produce) not accounted for
- Multi-ingredient products not fully supported

Improvements:

- Use ingredient % for better accuracy
- Train fuzzy logic to match receipt ↔ product data
- Add a scanning option
- Add user accounts
- Publish tool online





04/ Conclusions

- Educates consumers
  - Dairy and meat have a higher impact
  - Supply chain impact :
    - Farming >> Transport
- Reduces food-related CO_2 emissions
- Enables carbon tracking per purchase

Join me in transforming grocery shopping into a climate solution.

Let's make every receipt a step toward a greener future 🌎

And don't forget:

It's our daily choices that will shape our tomorrow. Together, we can make an impact

Ironhack

CARBON FOODPRINT

Thank you for your attention and good luck in your life

Special thanks to Jacquie, Eevee and Token

Valentin Perraud