**Front-end UI dashboard-Created by Xu Ye**

**1 Layout**

The web page uses the basic "header-content-footer" layout (Figure 1). In the layout of this website, the main navigation is placed at the top of the page and includes the team logo “Greenlight”, with the first level of navigation from left to right in it. We place the content in a fixed size navigation (1200px). The layout of the whole page is stable which is not affected by the browsing area. The top-down structure is in line with top-down browsing habits and is a classic website navigation pattern. This pattern demonstrates the efficiency of the main working area, while using some vertical space.

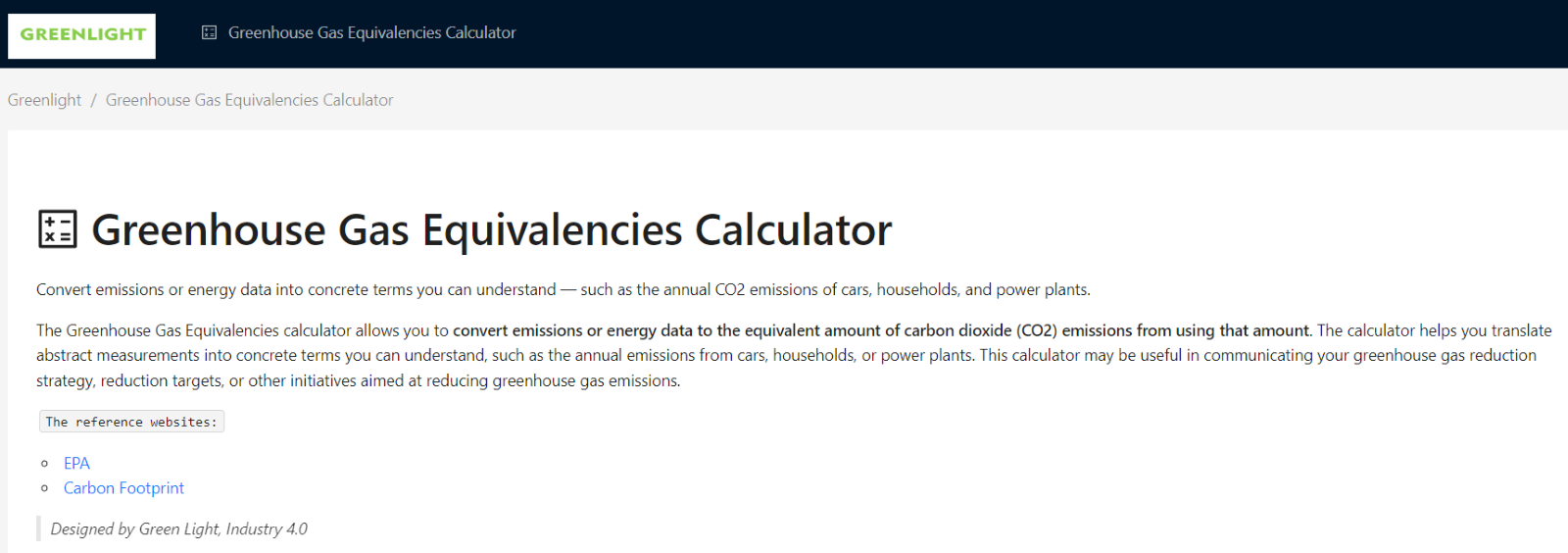


Figure 1 Layout of the website

**2 Typography**

The typography (Figure 2) of the website is relatively clear. The text part of the website shows the titles, paragraphs, and links of the web page. The functional modules are laid out with cards for Buttons, forms, drawers, tables, and charts respectively.

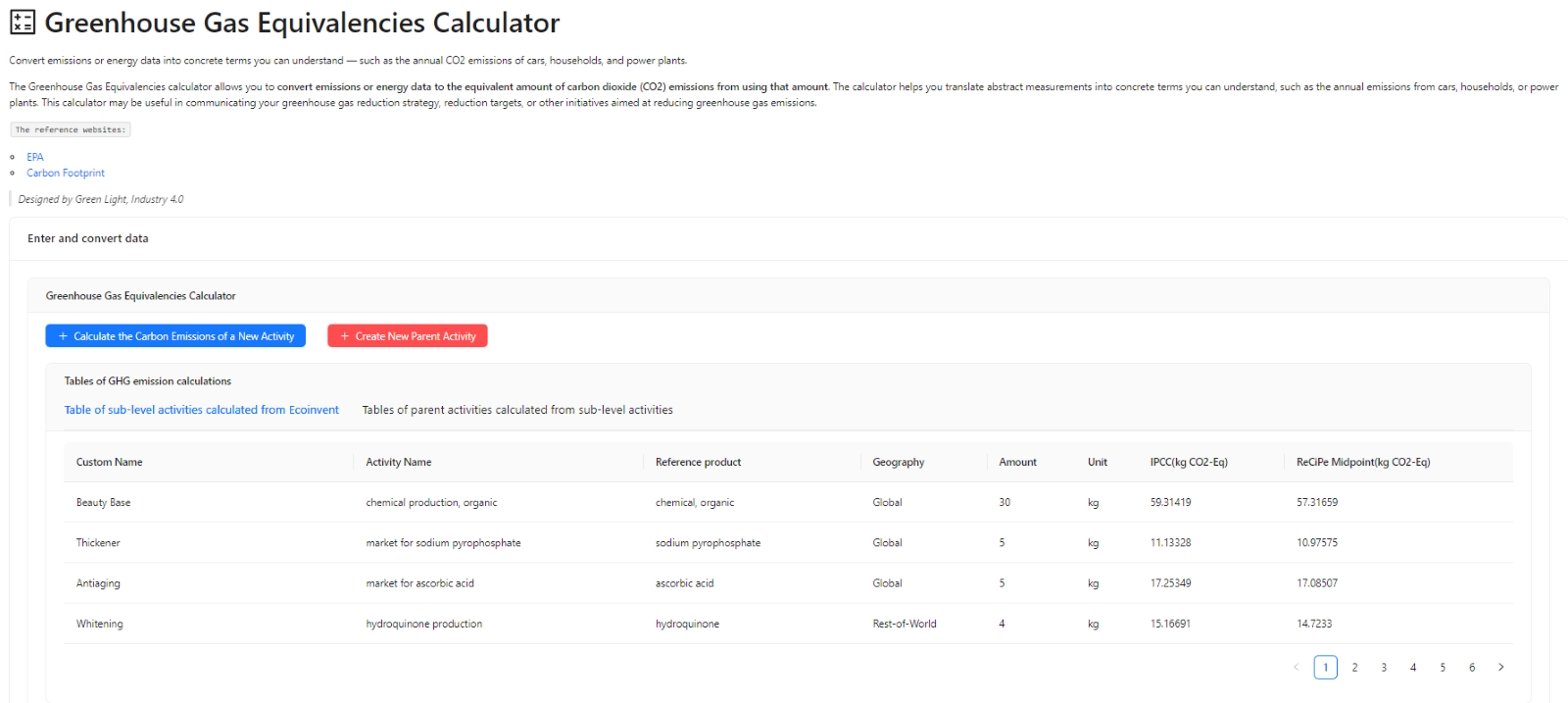


Figure 2 The typography of the website

**3 Functionality**

There are two main functions we implemented: Map1, Sub-level activity carbon emission calculation; Map2, Carbon emissions Life Cycle Impact Assessment of products. The specific functional modules are described in detail below.

***3.1 Map1, Sub-level activity carbon emission calculation***

Map1 acts as a carbon footprint calculator that allows users to calculate the equivalent emissions of carbon dioxide caused by the production of a certain number of products in an activity.

The sub-level activity carbon emission calculator is triggered in the drawer with a form when the button “Calculate the Carbon Emissions of a New Activity” is pressed. The drawer slides in from the edge of the parent form, covering part of the parent form content. When the user clicks “close” button, the page can smoothly return to the original task. When the user clicks the "What are they?" button, a second-level drawer will slide out the introduction of IPCC 2013, GWP 100a and ReCiPe Midpoint (H), GWP 100a.

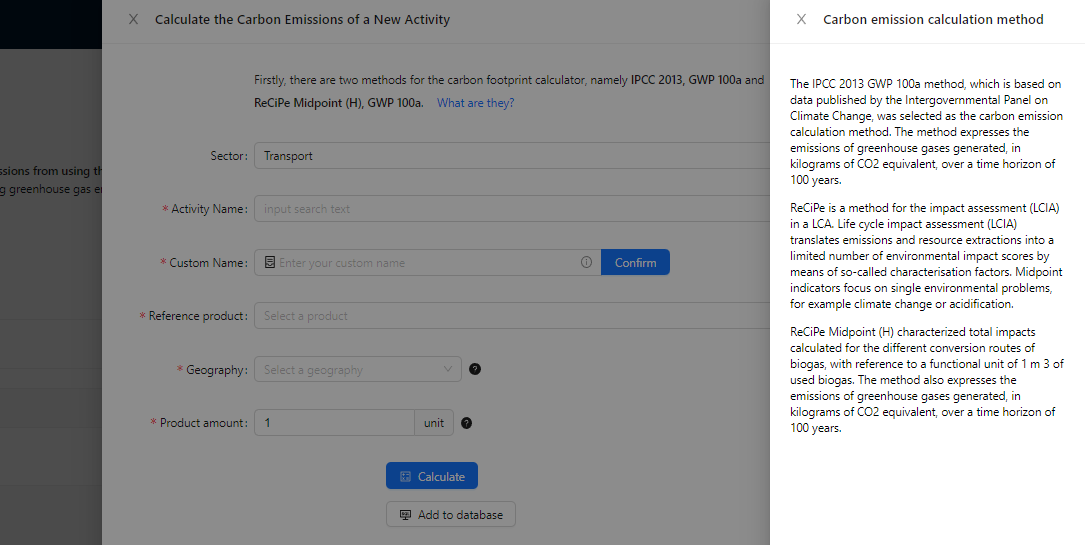


Figure 3 The second-level drawer

For the calculation section (Figure 4), we interact with a form that includes data input, data selection, validation, and styling. The form consists of four selectors, one InputNumber, one Input, two buttons, and five tooltips. The selectors are expanded to perform remote data searches for the corresponding options. Each selector interacts with the URL of its corresponding Fast API via Fetch or Jsonp, which returns all records containing the keyword entered by the user. The URL of the Fast API returns the results returned by our designed carbon emission algorithm and keyword search algorithm and is presented accordingly in the selector options of the front-end. We need the user to select sector, activity name, product, and geographic location.

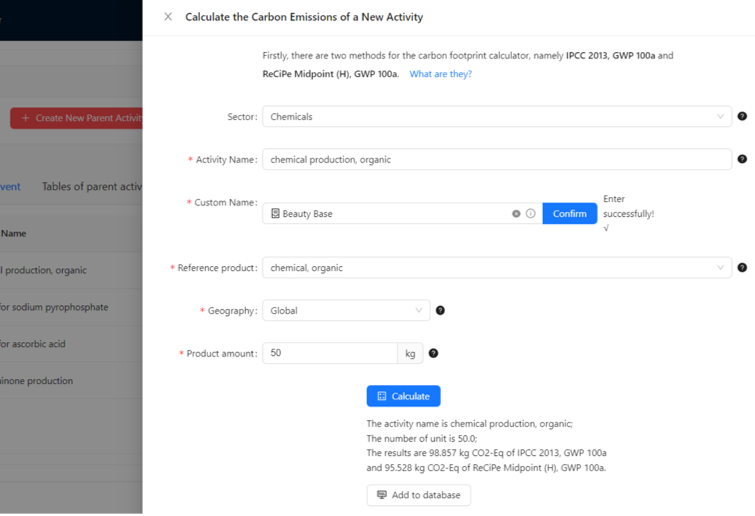


Figure 4 Carbon footprint calculation form

Sector selector covers a range of industrial sectors, including agriculture and animal husbandry, building and construction, chemicals and plastics, energy, forestry and wood, metals, textiles, transport, touristic accommodation, waste treatments and recycling, and water supply, among other. The sector selector rule is unnecessary here, and we can still calculate the carbon footprint by not selecting the sector option. However, all subsequent selectors are required. The activity name selector is a prerequisite for the proper operation of subsequent selectors. We set the activity as a global variable and assign the onChange value to the variable when the user selects an activity. Activities can be ordinary transforming activity, market activity, import activity or market group. Similarly, when the user selects an activity name, the product selector returns all the product names for that activity for the user to choose from. The geography selector returns all the locations for that activity for the user to choose from. InputNumber lets the user enter the number of the corresponding unit of product, which is a floating-point number. As the user enters the value, the unit of the product will be automatically returned in the post tab. By setting the hook function useState, subsequent re-rendering will return the unit of the corresponding product. The input allows the user to customize the name of the activity, so that the activity is materialized in the actual production process.

In order to help users to search for activities more quickly and accurately, sometimes the input word does not match any Ecoinvent activity name. In this case, we will trigger the synonym recommendation algorithm. Our trained synonym recommendation algorithm can help users better find corresponding activities. When there are no matched activities for the keyword entered by the user, our algorithm generates synonyms for the input word and returns activities containing synonyms in the selected sector. Figure 5 shows an example: we want to search "flight", but no activity name contains "flight". The selector will recommend activities containing its synonyms, you can see that the algorithm recommends a lot of activities about "aircraft" for the user to choose.

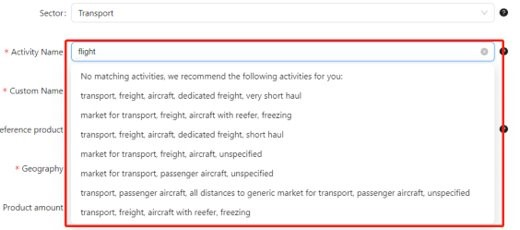


Figure 5 Synonym recommendation activities

Next are two buttons, namely the "Calculate" button and the "Add to database" button. Clicking the "Calculate" button will send a get request through axios for all parameters previously selected or entered by the user to obtain the results of the carbon emission algorithm through the backend. The useState function is used to transform the initial state "" into the calculated result. Clicking the "Add to database" button will send the Activity Name, Reference product, Geography, Amount, Unit and calculated carbon emissions through axios get request. They are stored in a local MySQL database through Fast API and pymysql. When the execution is successful, the web page will feedback the success message and refresh the web page. If the execution fails, the console will respond with an error, and the web page will respond with a failed message. Finally, in order to let the user know the meaning of each component of the form more clearly, so that the user can better choose, there will be a tooltip with an icon behind the end of each component. When the mouse moves in, the tip will be displayed, and when the mouse moves out, it will disappear.

When the calculation is completed, we use a table (Figure 6) to display the results of the calculation, which has eight columns and can display five rows of data on a page. In this project, through the ajax method, it is demonstrated that the data is get and displayed from the back-end database (Figure 7), and the relevant records are selected according to the row number through the select command of pymysql. There are two parameters, page and results, which represent the number of pages and the number of rows per page. Through these two parameters, the corresponding data can be selected to realize the pagination of the table.

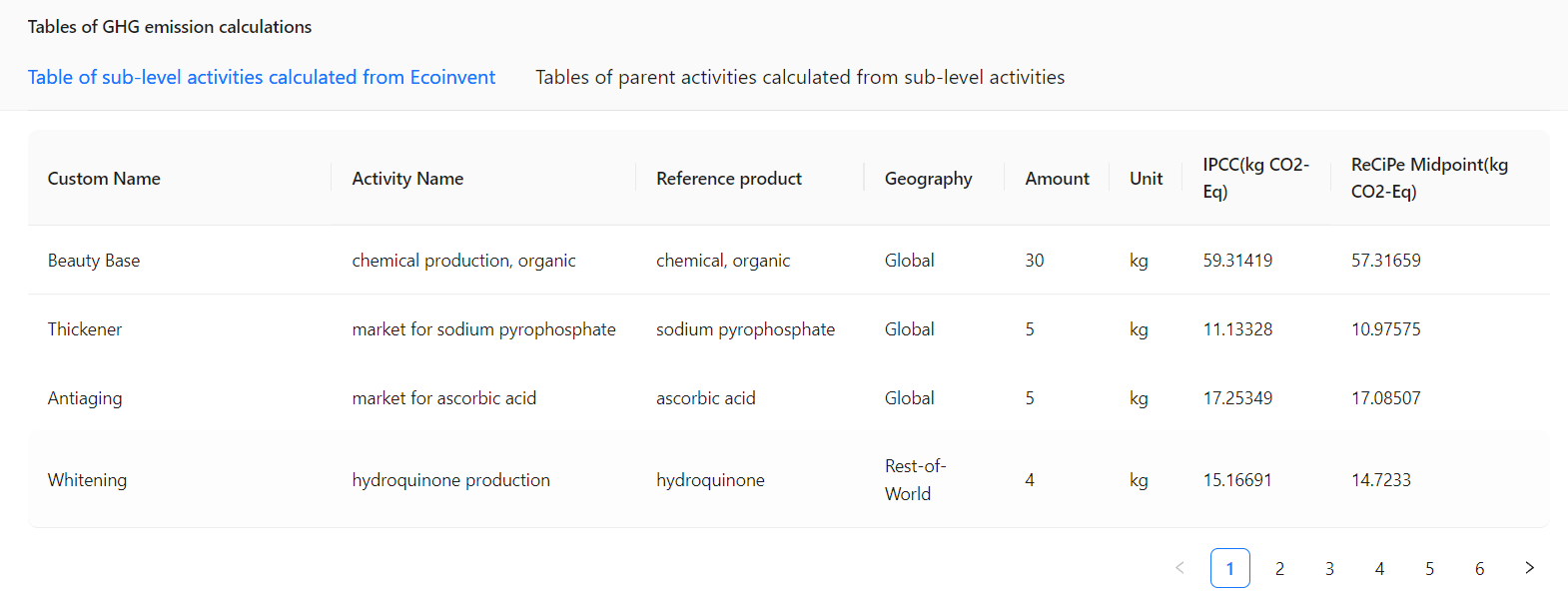


Figure 6 Table of sub-level activities

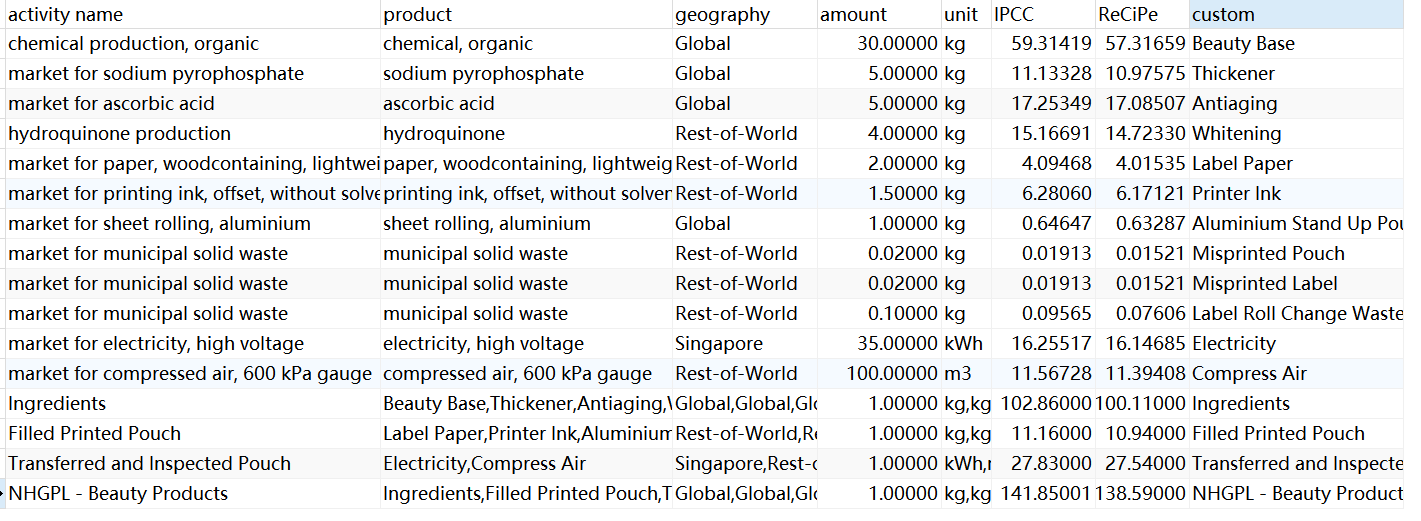


Figure 7 Map1 database

We make dashboards to show the carbon emissions of all the activities. For clarity and cleanliness, we use a card with tabs to host all the charts. This project uses Ant Design Charts to draw graphs, which is a very useful UI charts library. For the carbon emission presentation of different carbon emission calculation methods, we use a grouped bar chart (Figure 8) to draw and set more chart interactions so that users can compare the carbon emission of activities very clearly. We use fetch and hook functions *useEffect* and *useState* for requesting and getting remote data from Fast API. We use the calculation method as a series Field, with the activity name as the horizontal coordinate and the CO2 equivalent as the vertical coordinate. The pie charts (Figure 9) are presented using cards with tabs divided into IPCC 2013, GWP 100a and ReCiPe Midpoint (H), GWP 100a. The pie charts provide a more visual representation of the percentage of carbon emissions between different activities. The pie charts are also set up with graphical interactions that allows the selection of activities or methods, allowing the user to select several specific activities or different method to compare the percentage of carbon emissions. We also use fetch and the hook functions useEffect and useState to request and obtain remote data from the Fast API, with the carbon emissions as the angle field and the activity name as the color field.

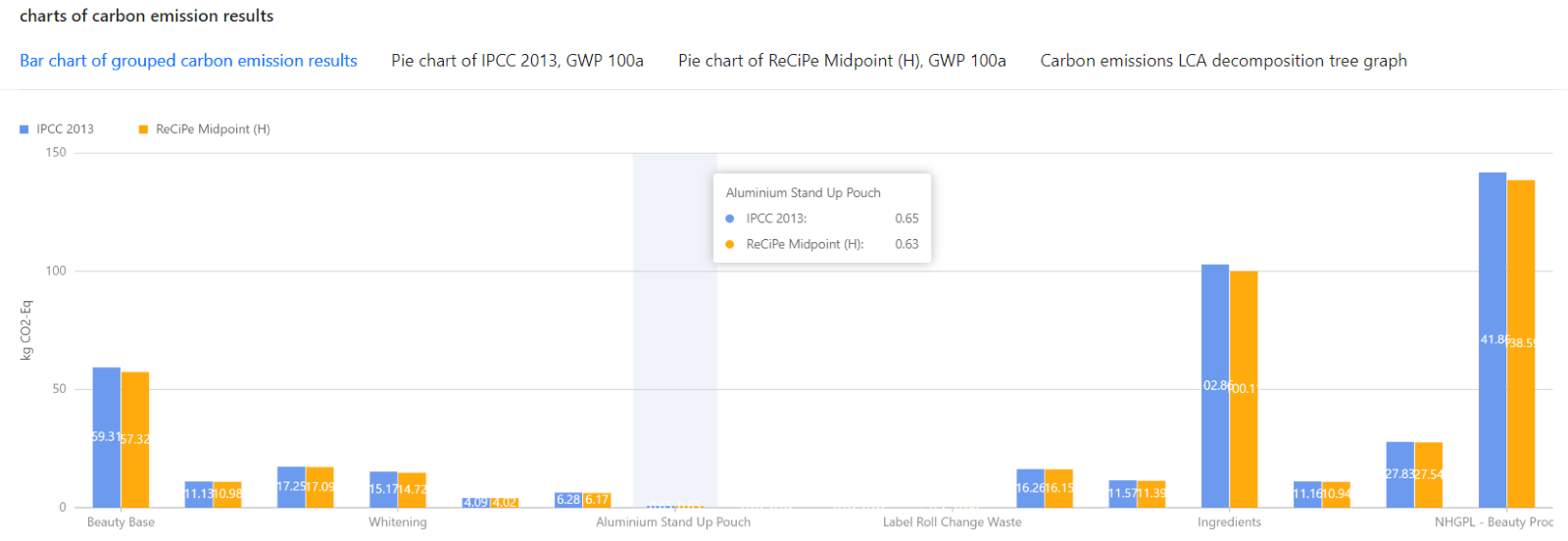
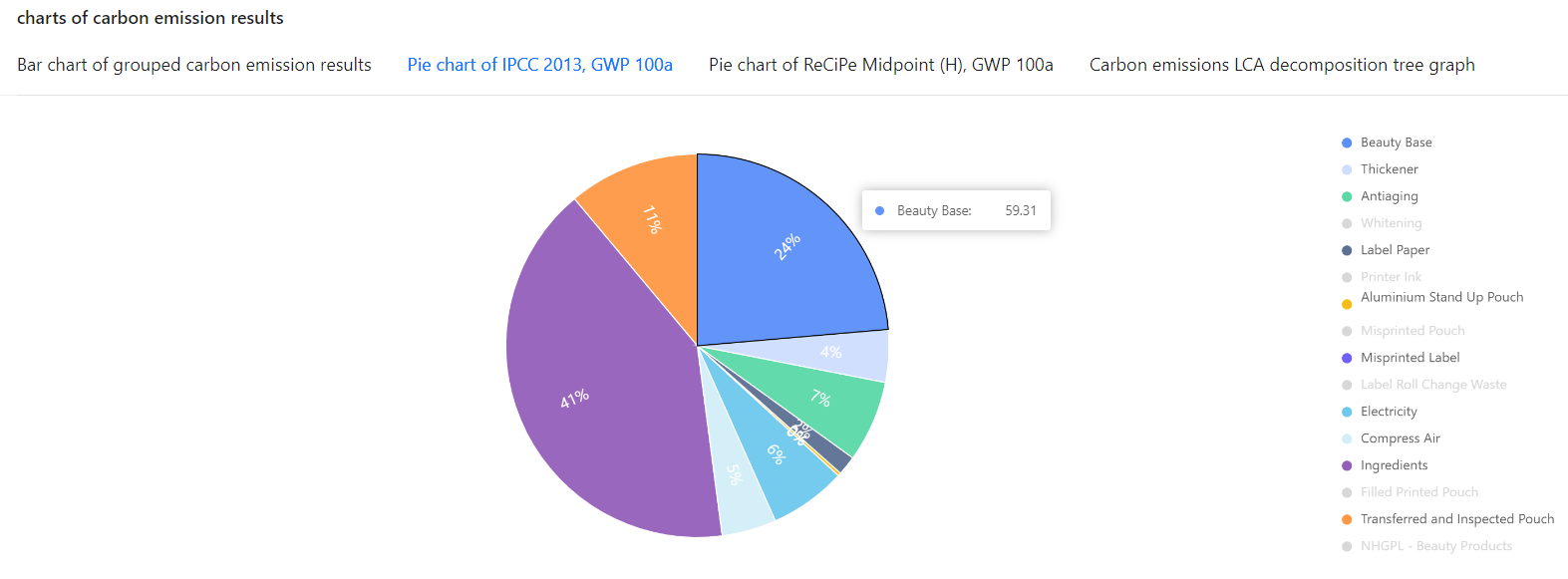


Figure 8 Grouped bar chart with graphical interactions

 Figure 9 Pie charts with graphical interactions

***3.2 Map2, Carbon emissions Life Cycle Impact Assessment of products***

Since the methodology of our project is product LCIA, after completing the network implementation of the basic sub-level activity carbon calculation, considering a product supply chain, each stage may include multiple sub activities, and each sub activity contains multiple activities, our function cannot be limited to calculating the carbon emissions of the lowest level activity only. Suppose that activity A and activity B existing in the database but user’s new activity requires 2A and 1B to produce 1C, how to calculate the carbon emissions for this new activity? This question is addressed at Map2 function.

Map2 function can chain all the activities of the product life cycle, showing the supply, flow, hierarchy, and proportion of the product. Through this function, different roles in the production process can form a complete carbon emission analysis of the product life cycle.

Map2 allows users to combine the results of activities previously filled out by different vendors to form a parent activity. We trigger the Map2 form (Figure 10) in a drawer by clicking on "create new parent activity" button. The form contains two selectors, one InputNumber, two inputs, three buttons, and three Tooltips.  The user names the parent activity in the input and enters the number of parent and child activities in InputNumber. Pressing the buttons following inputs assign the global variables to the input value.  A multiple selector returns the activity names of all child activities, which the user can select from. Users can select multiple sublevel activities at the same time. We use fetch to get the child's activity name. The level selector selects the level of the activity in the product’s life cycle. Pressing the "Create parent activity" button sends all data to the back end, where an algorithm calculates carbon emissions and inserts them into the database.

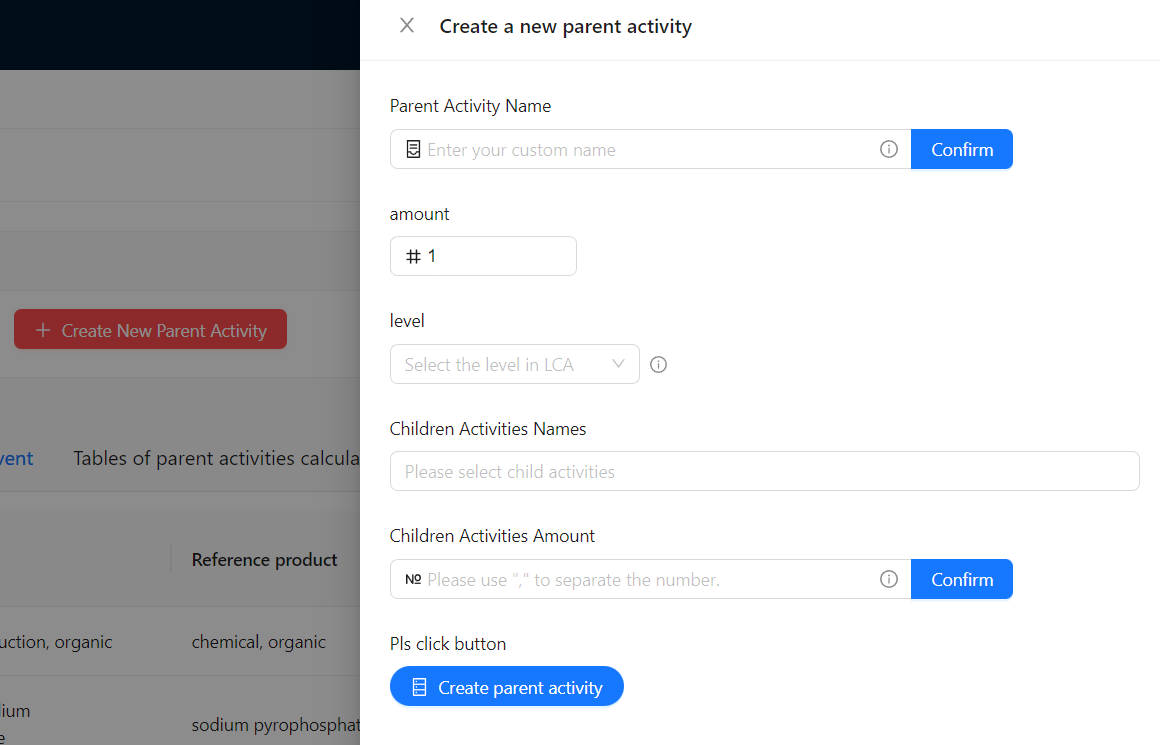
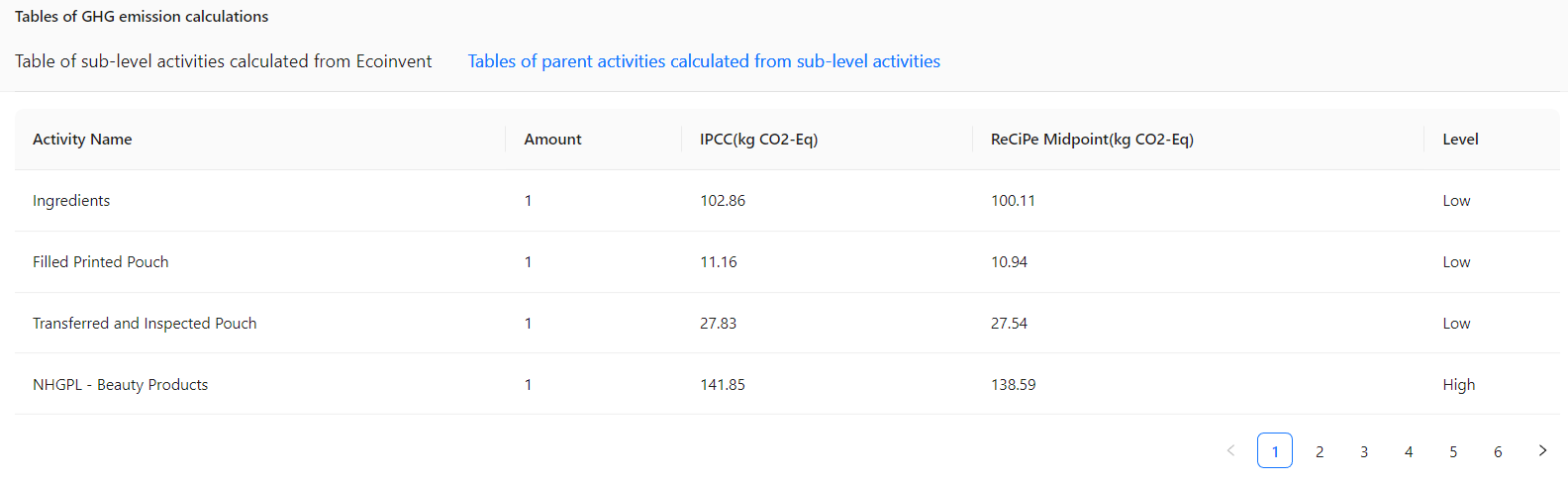


Figure 10 Map2 form

We also display the data in the card of the table and get the data by Fetch. The table (Figure 11) has 5 columns and displays up to 5 records on one page. There are two parameters, page and results, which represent the number of pages and the number of rows per page. Through these two parameters, the corresponding data can be selected to realize the pagination of the table. The database (Figure 12) stores the name, coefficient and level of the parent activity and the name, coefficients, number and proportion of carbon emission of the corresponding sub- activities.

Figure 11 Tables of parent activities calculated from sub-level activities

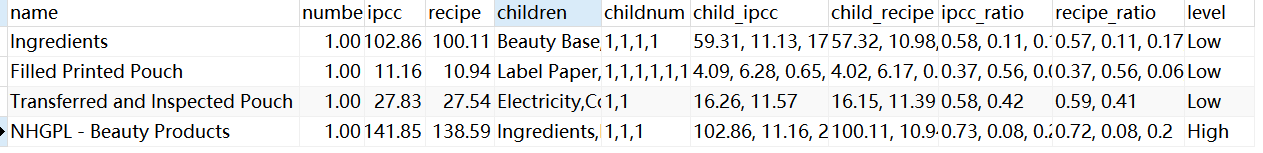
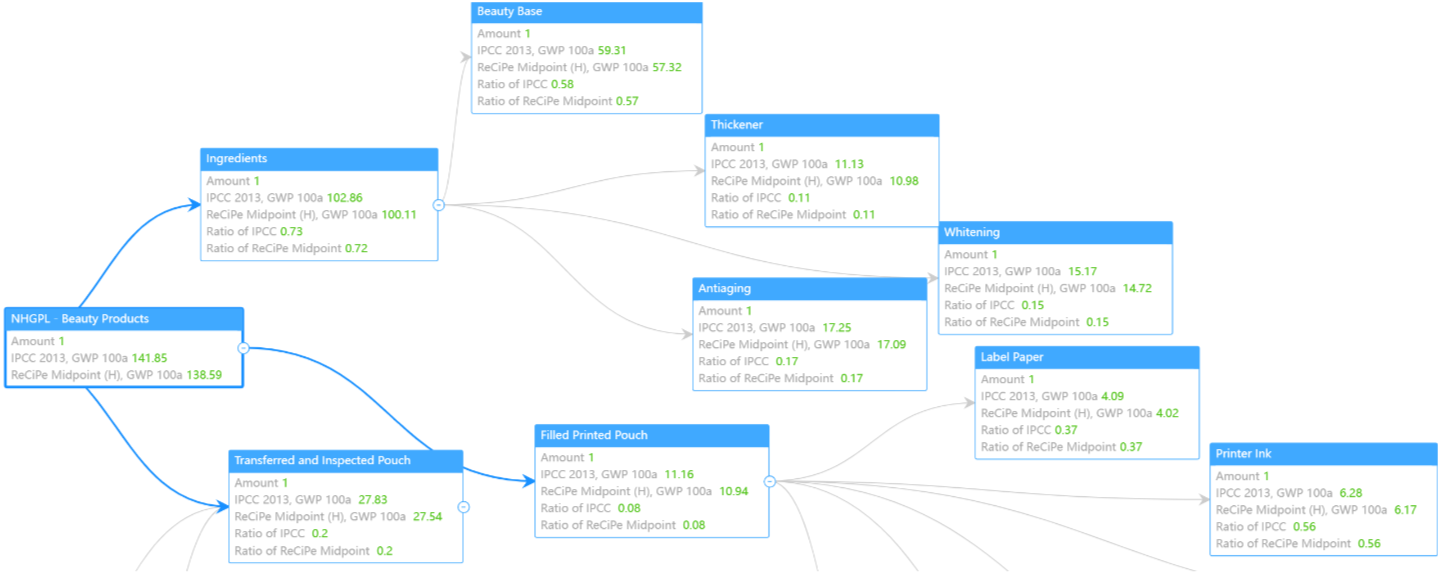


Figure 12 Map2 database

Finally, we will transform the data obtained from the Map2 database into a specific data format to clarify the node relationship through the back-end algorithm. And the dashboard will display them through the decomposition tree graph in the card with tabs. The carbon emissions LCIA decomposition tree graph shows the hierarchy, flow and proportional relationship of carbon emission activities. We can see the carbon emission and proportion of each node, helping users to carry out a more comprehensive assessment of the carbon emission in the life cycle of the product. We use the fetch method to obtain the parent-child level data.

As an example, from Figure 13, our topmost activity is the production line of beauty products, whose carbon emission of one time is about 140 kg CO2 equivalence. LCIA of beauty products comes from 3 activities, namely “Ingredients”, “Filled Printed Pouch”, and “Transferred and Inspected Pouch”. The carbon emissions proportion of these three activities are 72%, 8% and 20% respectively. Therefore, we could regard the “Ingredients” as a hot spot, of which we could focus on the child level activities, such as beauty base, whitening, etc. In the “Ingredients” production activities, the carbon emission of beauty base is about 58 kg CO2 equivalence, accounting for around 58%. Therefore, in the LCIA of beauty products, the carbon emission of the production of beauty base is worthy of attention and optimization.

Figure 13 Carbon emissions LCIA decomposition tree graph

***Created by Xu Ye***

***2023-1-6***