

## Materialdatenbank Tool (v1.0.0)

### Documentation

#### Project Title:

Materialdatenbank Tool.

#### Project Goal:

The goal of the project is to develop a simple, interactive, online neighborhood planning tool. The tool is called "*Materialdatenbank Tool*".

The tool's purpose is to provide the user with a holistic evaluation of buildings' variants from an ecological point of view and to enable comparing them. It achieves this by calculating the total values of different environmental effect and energy consumption indicators of a building. The calculation is based on the life cycle assessment of materials used in constructing the building.

This tool supports the user in the decision-making process during the early planning phase. It guides the user to select the best building variant according to their criteria.

#### License:

The tool is developed by *Fraunhofer IESE*.

The material database is provided by *lfaS*.

#### End User:

*Materialdatenbank* tool is a simple, easy to use tool that could be used by:

- Domain specialized users (e.g., architects, civil engineers, construction companies).
- Unspecialized users (i.e., anyone interested in studying the ecological effect of a building).

#### Used Technologies:

The tool is a *single-page React application*. It is bootstrapped with *Create React App* and is built using *TypeScript* and *React Bootstrap* front-end framework components.

*D3.js* library is used for data visualization. While *react-i18next* internationalization framework is used for translation.

## Tool Functionality:

The tool has 4 routes. The following tables (1- 4) explain the main functionalities of each route:

Table (1): Route 1 functionalities

<b>Route 1</b>	
<b>Navigation-bar link:</b> Build	<b>Project module:</b> src/modules/Build.tsx
<b>Purpose (1):</b> the user can create a new building model	
<b>Accessibility (1):</b> the user can use one of the following ways: <ol style="list-style-type: none"><li>1. The user specifies the route in the URL.</li><li>2. The user clicks on the corresponding navigation-bar link.</li></ol>	
<b>Main functionalities (1)</b> <ol style="list-style-type: none"><li>1. The user can add a material to the model being created:<ol style="list-style-type: none"><li>1.1. The user must select the category &gt; subcategory &gt; material.</li><li>1.2. The tool should display the reference unit of the selected material.</li><li>1.3. The user must enter a valid quantity of the material, i.e., quantity is a number &gt; 0.</li><li>1.4. The user must click on the add button.<ol style="list-style-type: none"><li>1.4.1. If the user did not choose a material, the tool should warn the user and the add operation would not be successful.</li><li>1.4.2. If the quantity is not valid or not specified by the user, the tool should warn the user and the add operation would not be successful.</li></ol></li><li>1.5. The tool adds the material with the specified quantity and displays its details in a table.</li></ol></li><li>2. The user can delete a material from a model.</li><li>3. The user can decrease the quantity of a material in a model<ol style="list-style-type: none"><li>3.1. The user should perform step 2.</li><li>3.2. The user should perform step 1.</li></ol></li><li>4. The user can increase the quantity of a material in a model<ol style="list-style-type: none"><li>4.1. The user performs step 1 for the material he/she wants to increase its quantity.</li><li>4.2. In step 1.3, the user must enter the desired increment in the quantity, and the tool should add this quantity to the existing one and reflect the update on the corresponding field in the table.</li></ol></li><li>5. The user can delete all the materials from a model.<ol style="list-style-type: none"><li>5.1. The tool should display a confirmation message.</li></ol></li><li>6. The user can save a model.<ol style="list-style-type: none"><li>6.1. The model must contain materials. If the model is empty, the tool should notify the user, and the model should not be saved.</li><li>6.2. The user must enter a valid model name, i.e., a non-taken name.</li><li>6.3. If the model's name already exists, the tool should notify the user, and the model should not be saved.</li></ol></li><li>7. The tool displays a visual representation, a bubble chart, of the model created by the user.<ol style="list-style-type: none"><li>7.1. Each bubble represents a material. A different color is assigned to each one.</li><li>7.2. The size of the bubble encodes the mass of the material.</li></ol></li></ol>	

<p>7.3. If the reference mass of the material is not available in the material database, i.e., the mass cannot be calculated by the tool, the tool should visualize this material as a black dot in the bubble chart.</p> <p>7.4. The x-axis represents the quantity of the material, according to the material reference unit.</p> <p>7.5. The y-axis represents the value of an environmental effect/energy consumption indicator selected by the user.</p> <p>7.6. Using a dropdown, the user can select one environmental effect/energy consumption indicator to be displayed.</p> <p>7.7. Using a PC ,the user can hover over a material to view its details, e.g., material name, material mass ... etc.</p>
<b>Purpose (2):</b> the user can view/update/rename a model
<b>Accessibility (2):</b> the user clicks on the View/Update/Rename link of a model in Route 4
<b>Main functionalities (2)</b>
<ol style="list-style-type: none"> <li>1. Main functionalities (1).</li> <li>2. The user can select 'cancel' to cancel the operation.</li> <li>3. The tool redirects the user to Route 4 upon successful completion/cancelation of an operation.</li> </ol>

Table (2): Route 2 functionalities

<b>Route 2</b>	
<b>Navigation-bar link:</b> Compare	<b>Project module:</b> src/modules/Compare.tsx
<b>Purpose:</b> the user can compare his/her buildings' models	
<b>Main functionalities</b>	
<ol style="list-style-type: none"> <li>1. The tool displays a list containing the names of user's models.</li> <li>2. The user can select one or more models to compare.</li> <li>3. For each selected model, the tool calculates and displays the total value of each energy consumption and environmental effect indicator.</li> </ol>	

Table (3): Route 3 functionalities

<b>Route 3</b>	
<b>Navigation-bar link:</b> Explore	<b>Project module:</b> src/modules/Explore.tsx
<b>Purpose:</b> the user can navigate through the material database content	
<b>Main functionalities</b>	
<ol style="list-style-type: none"> <li>1. The tool displays the list of materials' categories.</li> <li>2. The user can select a category to view the list of its subcategories.</li> <li>3. The user can select a subcategory to view the list of its materials.</li> <li>4. The user can select a material to view its details.</li> </ol>	

Table (4): Route 4 functionalities

<b>Route 4</b>	
<b>Navigation-bar link:</b> My Modules	<b>Project module:</b> src/modules/WorkSpace.tsx
<b>Purpose:</b> the user can control his/her buildings' models	
<b>Main functionalities</b>	
<ol style="list-style-type: none"> <li>1. The tool displays a list of the user's buildings' models.</li> <li>2. The user can delete a model. <ol style="list-style-type: none"> <li>2.1. The tool should display a confirmation message.</li> </ol> </li> <li>3. The user can view/edit/rename an existing model. <ol style="list-style-type: none"> <li>3.1. The tool redirects the user to Route 1 to perform these operations.</li> <li>3.2. Upon the successful completion/cancelation of an operation, the tool should redirect the user to Route 4.</li> </ol> </li> </ol>	

### Tool Features:

- **Responsiveness:** the tool is built using *React Bootstrap grid system*, which makes it fully responsive. The layout is suitable for PC's, tablets, and smartphones. It works on different screen resolutions and browsers' window sizes both on portrait and landscape modes.
- **Supported Languages:** the tool supports both English and German. The user can select the desired language from the *Language* dropdown in the navigation-bar.

### Material Database:

The tool uses the material database provided by *IfaS*. The database contains many categorized construction materials and their ecological life cycle assessment. It is provided as an excel file; the data is spread on multiple sheets. Table (5) illustrates the columns used from the excel file along with their description. To use/update the database in the *Materialdatenbank tool*, the data should be cleaned, reformatted, and saved as a *JSON* file. To create the *JSON* file, the following steps should be taken:

1. Create a single excel sheet composed of the excel file columns specified in table (5), in column '*Excel*'.
2. Clean the data and make sure that data entries satisfy the corresponding type, specified in table (5), in column '*Data type*'.
3. Rename the columns, using the corresponding names specified in the table (5), in column '*JSON*'.
4. Copy/paste the data from the excel sheet, along with the renamed headers, into *Mr. Data Convertor tool* (<http://shancarter.github.io/mr-data-converter/>).
5. Choose *Output as: "JSON-Properties"* to get the *JSON* format.
6. Save the result from step 5 into a *JSON* file named "*data*" (i.e., *data.json*).
7. In the project directory, place the file in (*src/shared/services*).

Table (5): The mapping between the excel and the JSON formats

Excel	JSON	Data type	Description
english	nameEn	string   null	The English name of the category/subcategory/material
german	nameDe	string   null	The German name of the category/subcategory/material
Nummer	id	number	A unique identifier for the record
Level	level	number	Enum {1,2,3} for the type of the record: 1: Category, 2: subcategory, 3: material
Parent	parentId	number	The identifier of the direct parent record (category/subcategory). 0: no parent (this record is a category)
Bezugseinheit	referenceUnit	string	The unit by which the material is measured (e.g., kg, m,...etc.)
Flächengewicht	basisWeight	number   null	Currently not used
Rohdichte [kg/m <sup>3</sup> ]	grossDensity	number   null	Currently not used
Schichtdicke	layerThickness	number   null	Currently not used
Bezugsmasse [kg]	referenceMass	number   null	The mass of a single unit of the material
alle in kWh PENRT	penrt	number   null	Energy Consumption indicators
alle in kWh PERT	pert	number   null	
alle in kWh PENRT-PENRM	penrtPenrm	number   null	
alle in kWh PERT-PERM	pertPerm	number   null	
alle in kg GWP	gwp	number   null	Environmental Effect indicators
alle in kg AP	Ap	number   null	
alle in kg EP	Ep	number   null	
alle in kg POCP	Pocp	number   null	
alle in kg ODP	odp	number   null	

## Architectural Decisions:

In the current architecture, all the processing is done at the client-side (i.e., the browser). The *data.json* file is sent to the client along with the needed *JS* and *TS* files.

User created buildings' models and the current selected language are stored in local storage at the client-side.

## Project Structure and Technical Details:

*src/Index.tsx* is the entry point of the application and *src/MaterialdatenbankApp.tsx* is the main component in which *React Router* is used.

Components are placed in 2 different directories inside the project directory:

1- **Container components** (*src/modules*):

Components that perform business logic, i.e., fetching the data, managing the state, performing side effects...etc.

1. **Presentational components** (*src/shared/views*):

Components concerned with the look of the UI. These components use the data and the callback functions, that are passed to them as props, to display the elements on the UI.

For styling, *inline styles* are used in each component, i.e., there is no central *CSS* file.

In *src/shared/services/MaterialService.ts*, the data (*data.json*) is parsed and the categories and subcategories are extracted and cached as variables for reuse, to avoid skimming the whole list again and again. The *MaterialService* class contains the functions that could be transferred to the server-side if needed in the future. However, even though there is no processing at the server side in version 1.0.0, asynchronous function calls are used to make it easier to migrate to server-side processing when needed.

Managing the storage and the retrieval of the models from the local storage is done in *src/shared/services/ModelsContext.tsx*.

In *src/i18n*, you will find *en.ts* and *de.ts* for the English and German versions of the static text used in the tool; for translation. For the translation of the categories, subcategories and materials' names, the corresponding names (*nameEn*, *nameDe*) in the *data.json* file are used.

## Project Run:

In the project directory, you can run: *npm start*

This will Run the app in the development mode.

Open <http://localhost:3000> to view it in the browser.