Inter IIT Tech Meet 13.0
Development Team Selection Task

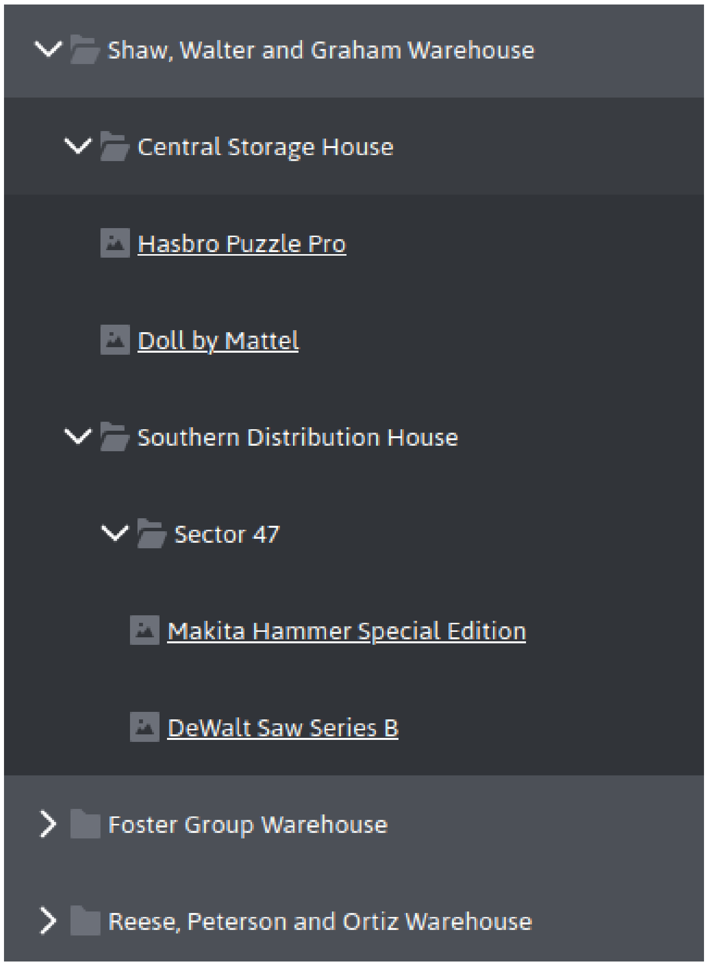
Godowns (or warehouses) require efficiently managing stored items such as toys, electronics, and tools. It's crucial to visualize the hierarchy of locations and the items within them to keep inventory organized and manageable. This challenge involves building a **Tree View Application** that represents godowns, their locations, sub-locations, and stored items, **and display details of selected item once clicked**

The tree structure will include locations, sub-locations, and items. Locations represent various sections of the godown, sub-locations are nested sections, and items represent the actual products stored in these sections.

Frontend

➔ **Tree Structure:** Display the hierarchy of godown locations, sub-locations, and items as a sidebar; the main section contains the details of the selected item.
  - **Locations/Sub-Locations:** Represent the different sections and areas within a godown, each of which may have sub-locations for better organization.
  - **Items:** These are the products stored in each location (e.g., toys, tools, electronics).
  - Allow users to expand/collapse the tree to view locations and items.

Example:


Shaw, Walter and Graham Warehouse
Central Storage House
Hasbro Puzzle Pro
Doll by Mattel
Southern Distribution House
Sector 47
Makita Hammer Special Edition
DeWalt Saw Series B
Foster Group Warehouse
Reese, Peterson and Ortiz Warehouse

➔ **Product Display:** When selecting a particular item, display all the details for the selected item on the main page. Creatively design this page with a user-friendly UI.

➔ **Basic Auth Page:** Create a login page, which can have either some predetermined credentials or a condition on the email domain to login to the web page.

➔ **Data Structure:**
- **Locations:** Represent sections and subsections of the godown.
    - **Godowns:**
        - **Attributes**: id, name, parent_id = null
    - **Sub-Godowns:** Nested within Godowns.
        - **Attributes**: id, parent_id, name
- **Items:** Represent products stored within these sections.
    - Nested under Sub-Godowns
    - **Attributes:** item_id, name, quantity, category, status, godown_id, price, brand, attributes (open to your interpretation), image_url.
- Use a simple JSON structure, similar to the one provided below:
- **Locations Example:** Go to this Link for all godowns: godowns.json

```
[
   {
       "id": "d72518e97c3f4a68979153f2b8e9308e",
       "name": "Torres, Rowland and Peters Warehouse",
       "parent_godown": null
   },
   {
       "id": "a6565c19ccbb4bb8a2a04130a14988db",
       "name": "Western Center House",
       "parent_godown": "d72518e97c3f4a68979153f2b8e9308e"
   },
   {
       "id": "4ce59062eadd4d4ca5e105f30a9f7256",
       "name": "Sector 60",
       "parent_godown": "a6565c19ccbb4bb8a2a04130a14988db"
   },
   {
       "id": "e06d51d013ab47d791d90f0ea097cc66",
       "name": "Sector 77",
       "parent_godown": "a6565c19ccbb4bb8a2a04130a14988db"
   },
   {
       "id": "7579aa5649484332ab86c0b52f2b3222",
       "name": "Western Stockpile House",
       "parent_godown": "d72518e97c3f4a68979153f2b8e9308e"
   }
]
```

◆ **Items Example**: Download all items from items.json

```
[
    {
        "item_id": "9b7b6b4a543c4bb090de37870a49d70b",
        "name": "Black & Decker Screwdriver Series B",
        "quantity": 339,
        "category": "Tools",
        "price": 448.43,
        "status": "in_stock",
        "godown_id": "b3f0e83bb8ee4e308d759c95e2c3507d",
        "brand": "Black & Decker",
        "attributes": {
            "type": "Hand Tool",
            "material": "Plastic",
            "warranty_years": 1
        },
        "image_url": "https://m.media-amazon.com/images/I/41-T3GBGYUL.jpg"
    },
    {
        "item_id": "663a9d18f1894f6e874f7cedd135e248",
        "name": "Samsung Smartphone 62",
        "quantity": 111,
        "category": "Electronics",
        "price": 102.4,
        "status": "in_stock",
        "godown_id": "f37e12cc6bbf437aba6672628f54efa5",
        "brand": "Samsung",
        "attributes": {
            "wattage": 56,
            "voltage": 220,
            "color": "AntiqueWhite"
        },
        "image_url":
"https://fdn2.gsmarena.com/vv/bigpic/samsung-galaxy-f62.jpg"
    }
]
```

Backend (Optional)

- ➔ **API Endpoints:**
    - ◆ Use your preferred backend stack to expose simple REST endpoints according to the data given to you and document them.
- ➔ **Database:** Use databases (JSON, SQLite, PostgreSQL, or MongoDB) to store locations and items. The collections/tables should be:
    - ◆ **Locations**: Store godown locations and their hierarchy.
    - ◆ **Items**: Store individual items and their corresponding locations.

Containerization and Deployment (Must Try)

- ➔ **Dockerization**:
    - ◆ Dockerize the entire system for easier deployments.
    - ◆ You can optionally set up automatic deployment, depending on experience and time constraints.
- ➔ **Deployment:** Deploy the application (e.g., on Vercel, railway, Netlify, or AWS). Preferably deploy from scratch using a VM from Azure / DigitalOcean (You get free credits on your student ID; look at the resources for the GitHub Student Developer Pack).

Bonus Features (Optional)

If you complete the MVP and want to demonstrate additional skills, you can implement these bonus features:

- ➔ **Frontend Bonus Features**
    - ◆ **Search and Filters**:
        - Add a **search feature** to allow users to search for specific locations or items.
        - Add a filter to display only items of a certain type (e.g., Toys, Electronics).
    - ◆ **Drag and Drop:**
        - Allow items to be dragged and dropped between different godowns.
- ➔ **Backend Bonus Features**
    - ◆ **Authentication**:
        - Implement a simple **login system** (e.g., using JWT or Sessions) to protect certain routes.
    - ◆ **Advanced Search and Filtering**:
        - Support backend filtering and searching through query parameters (e.g., filter items based on stock levels, type, etc).
        - AI/ML Integration: You can add any AI or ML features that you find suitable for completing the features.

General Tips

- ➔ **Choice of Tech Stack:** You can choose the tech stack and frameworks for both the front-end and back-end.
- ➔ **Frontend Focus:**
    - ◆ Use modern frontend frameworks like **React, Vue, Angular, etc.**
    - ◆ **Animations and transitions** are important for a smooth user experience.
- ➔ **Backend Simplicity**: Keep the backend as simple as possible. Focus more on front-end development and deployment.
- ➔ **Database**: Use **JSON**, **SQLite, PostgreSQL**, or **MongoDB** as the database.
    - ◆ For **SQLite**, provide an SQL dump.
    - ◆ For **MongoDB**, provide a `**.env**` file with connection details or hardcode the URL.

- ➔ **Use of Resources**: You can use any online resources to complete the task.
- ➔ **Q&A**: Be prepared to answer questions about your project features and implementation choices.
- ➔ For any queries send an email to dev.interiittech13.0@gmail.com or submit the form

Submission Guidelines

- ➔ **GitHub Repository**: Submit the project via a GitHub repository link. Make sure to make the repository public.
- ➔ **README File**:
  - ◆ The README file should explain your ideas, thought process, and the steps you took to implement the project.
  - ◆ Include clear instructions on how to set up the project and run it locally, run it using docker, along with the **deployment** links.
- ➔ **Video Demonstration**: Upload a simple demonstration video of the project to your google drive and insert the link in the GitHub readme to showcase its features. Make sure to adjust the permissions of sharing as everyone with a link can view the document.
- ➔ **Incomplete Submissions**: Partial, but functional submissions are accepted.
- ➔ **Deadline:** The deadline for the task submission is on **12/10/2024.**
- ➔ **Judgment Criteria:**
  - ◆ Responsive Design
  - ◆ UI / UX
  - ◆ Deployment
  - ◆ Code Quality

Resources

- ➔ Free domains for your live deployments: freedomain.one, duckdns.org
- ➔ Custom data generation for more extensive testing application: Faker.js
- ➔ Dockerization: Docker Tutorial, Docker Curriculum
- ➔ GitHub Student Developer Pack: https://education.github.com/pack
- ➔ Deploy from scratch: Deploying a web-application using Nginx server and Reverse Proxy 🌻 | by Rajani Ekunde | Medium, Web-Application Deployment Using Nginx Web Server