

Mini Indoor Plant Nursery

The screenshot displays three main components of the application:

- Search plants (UI-only):** A sidebar on the left containing a search bar and a list of plants. The list includes "Pilea peperomioides" (selected), "String of Hearts", and "ZZ Plant". Each item has difficulty and rarity status indicators.
- Add New Plant (UI-only):** A form on the right for adding new plants. It includes fields for "Plant name", "Scientific name", "Difficulty" (dropdown), "Light" (dropdown), "Short description", and buttons for "Add Plant" and "Clear".
- Pilea peperomioides Detail View:** A card showing the details of the selected plant. It includes a "View" button, a "Buy (UI-only)" button, the scientific name "Pilea peperomioides", a description ("Coin-shaped leaves, easy to propagate; prefers bright, indirect light."), difficulty and rarity status indicators, and light and water requirements ("Light: Bright", "Water: Moderate").

types.ts

```
export interface Plant {
    id: number;
    name: string;
    scientific?: string; // ? used to denote optional properties
    difficulty: 'Easy' | 'Medium' | 'Hard';
    light: 'Low' | 'Medium' | 'Bright';
    water: 'Low' | 'Moderate' | 'Frequent';
    rarity?: 'Common' | 'Uncommon' | 'Rare';
    description: string;
}
```

main.tsx

```
import { StrictMode } from 'react'
import { createRoot } from 'react-dom/client'
import './index.css'
import App from './App.tsx'
import type { Plant } from './types';
import 'bootstrap/dist/css/bootstrap.min.css';

const plants: Plant[] = [
    {
        id: 1,
        name: "Pilea peperomioides",
        scientific: "Pilea peperomioides",
        description: "Coin-shaped leaves, easy to propagate; prefers bright, indirect light.",
        difficulty: "Easy",
        rarity: "Common",
        light: "Bright",
        water: "Moderate"
    }
]
```

```
difficulty: "Easy",
light: "Bright",
water: "Moderate",
rarity: "Uncommon",
description: "Coin-shaped leaves, easy to propagate; prefers
bright, indirect light."
},
{
  id: 2,
  name: "String of Hearts",
  scientific: "Ceropegia woodii",
  difficulty: "Medium",
  light: "Bright",
  water: "Low",
  rarity: "Rare",
  description: "Trailing succulent-like vine with heart-shaped
leaves."
},
{
  id: 3,
  name: "ZZ Plant",
  scientific: "Zamioculcas zamiifolia",
  difficulty: "Easy",
  light: "Low",
  water: "Low",
  rarity: "Common",
  description: "Extremely hardy plant that tolerates low light and
neglect."
}
];

createRoot(document.getElementById('root')!).render(
  <StrictMode>
    <App plants={plants} />
  </StrictMode>,
)
```

app.tsx

```
import React, { useState } from 'react';
import type { Plant } from './types';
import PlantList from './PlantList';
import PlantCard from './PlantCard';
import { PlantForm } from './PlantForm';

interface AppProps {
  plants: Plant[];
}

const App: React.FC<AppProps> = ({ plants }) => {

  const [selectedId, setSelectedId] = useState<number>(plants[0].id);
  const selectedPlant = plants.find((p) => p.id === selectedId);

  return (
    <div className=".container p-4">

      <h1 className="mb-4">Mini Indoor Plant Nursery</h1>

      <div className="row">

        <div className="col-md-4">

          <div className="mb-3">
            <input
              type="text"
              className="form-control"
              placeholder="Search plants (UI-only)"
            />
          </div>

          <PlantList
            plants={plants}
```

```
        selectedId={selectedId}
        onSelect={setSelectedId}
      />

      <button
        className="btn btn-secondary w-100 mt-3"
        // Optional: You can add logic here later to reset the
selection
        onClick={() => console.log("Reset clicked")}
      >
        Reset Selection (UI-only)
      </button>

    </div>

<div className="col-md-8">

  <div className="mb-4">
    <PlantForm />
  </div>

  {selectedPlant && <PlantCard plant={selectedPlant} />}

  </div>
</div>
</div>
);

};

export default App;
```

plantcard.tsx

```
import React from 'react';
import type { Plant } from './types';

interface PlantCardProps {
    plant: Plant;
}

const PlantCard: React.FC<PlantCardProps> = ({ plant }) => {
    return (
        <div className="card mb-3">
            <div className="card-body">

                <div className="d-flex justify-content-between align-items-start">
                    <div>
                        <h3 className="card-title">{plant.name}</h3>
                        <h6 className="card-subtitle mb-2 text-muted">({plant.scientific})</h6>
                    </div>
                    <div>
                        <button className="btn btn-outline-secondary btn-sm me-2">View</button>
                        <button className="btn btn-outline-danger btn-sm">Buy (UI-only)</button>
                    </div>
                </div>

                <p className="card-text">{plant.description}</p>

                /* Badges for Difficulty and Rarity */
                <div className="mb-3">
                    <span className="badge bg-secondary me-2">{plant.difficulty}</span>

```

```
        <span className="badge bg-warning text-dark">{plant.rarity}</span>
    </div>

    <div className="row">
        <div className="col-sm-6">
            <small className="text-muted">Light</small>
            <div>☀️ {plant.light}</div>
        </div>
        <div className="col-sm-6">
            <small className="text-muted">Water</small>
            <div>💧 {plant.water}</div>
        </div>
    </div>
</div>
);
};

export default PlantCard;
```

plantform.tsx

```
import React from 'react';

export const PlantForm: React.FC = () => {
    return (
        <div className="card p-3 bg-light">
            <h4 className="mb-3">Add New Plant (UI-only)</h4>
            <form>
                {/* Row 1: Text Inputs */}
                <div className="row mb-3">
                    <div className="col">
                        <input type="text" className="form-control" placeholder="Plant name" disabled />
                    </div>
                </div>
            </form>
        </div>
    );
};

export default PlantForm;
```

```
        <div className="col">
            <input type="text" className="form-control"
placeholder="Scientific name" disabled />
        </div>
    </div>

    {/* Row 2: Select Dropdowns */}
    <div className="row mb-3">
        <div className="col">
            <select className="form-select"
defaultValue="" disabled>
                <option value="" disabled>Difficulty</option>
                <option value="Easy">Easy</option>
                <option value="Medium">Medium</option>
                <option value="Hard">Hard</option>
            </select>
        </div>
        <div className="col">
            <select className="form-select"
defaultValue="" disabled>
                <option value="" disabled>Light</option>
                <option value="Low">Low</option>
                <option value="Medium">Medium</option>
                <option value="Bright">Bright</option>
            </select>
        </div>
        <div className="col">
            <select className="form-select"
defaultValue="" disabled>
                <option value="" disabled>Water</option>
                <option value="Low">Low</option>
                <option value="Moderate">Moderate</option>
                <option value="Frequent">Frequent</option>
            </select>
        </div>
    </div>
```

```
<div className="mb-3">
  <textarea
    className="form-control"
    placeholder="Short description"
    rows={2}
    disabled
  />
</div>

<div>
  <button type="button" className="btn btn-primary me-2" disabled>
    Add Plant
  </button>
  <button type="button" className="btn btn-outline-secondary" disabled>
    Clear
  </button>
</div>
</form>
</div>
);
};
```

plantlist.tsx

```
import React from 'react';
import type { Plant } from './types';

interface PlantListProps {
  plants: Plant[];
  selectedId: number;
  onSelect: (id: number) => void;
}
```

```
const PlantList: React.FC<PlantListProps> = ({ plants, selectedId, onSelect }) => {
  return (
    <div className="list-group">
      {plants.map((plant) => (
        <button
          key={plant.id}
          type="button"
          // We add 'active' if this plant is the selected one
          className={`list-group-item list-group-item-action
${plant.id === selectedId ? 'active' : ''}`}
          onClick={() => onSelect(plant.id)}
          style={{ cursor: 'pointer' }} // [cite: 32]
        >
          <div className="d-flex w-100 justify-content-between align-items-center mb-1">
            <h5 className="mb-1">{plant.name}</h5>
            <div>
              {/* Badges for Difficulty and Rarity */}
              <span className={`badge ${plant.id === selectedId ? 'bg-light text-dark' : 'bg-success'} me-1`}>
                {plant.difficulty}
              </span>
              <span className={`badge ${plant.id === selectedId ? 'bg-light text-dark' : 'bg-info text-dark'}`}>
                {plant.rarity}
              </span>
            </div>
            <p className="mb-1 small">{plant.scientific}</p>
          </button>
        )));
    </div>
  );
};

export default PlantList;
```