

Welcome to Your Mini Nursery

Manage your plant collection efficiently and effortlessly.

All Difficulties

Select a person

Choose a user...

Pilea peperomioides

Pilea peperomioides

Easy

Uncommon

String of Hearts

Ceropegia woodii

Medium

Rare

ZZ Plant

Zamioculcas zamiifolia

Easy

Common

NewPlantName

NewScientificName

Easy

Common

Reset Filters

Add New Plant

Plant name

Scientific name

Easy

Medium

Moderate

Short description

Add Plant

Clear

Pilea peperomioides

(Pilea peperomioides)

View Details

Remove

Coin-shaped leaves, easy to propagate; prefers bright, indirect light.

Easy

Uncommon

Light

Bright

Water

Moderate

© 2024 Mini Nursery.

Mini Plant Manager

Welcome to Your Mini Nursery

Manage your plant collection efficiently and effortlessly.

All Difficulties

Select a person

Choose a user...

Pilea peperomioides

Pilea peperomioides

Easy

Uncommon

String of Hearts

Ceropegia woodii

Medium

Rare

ZZ Plant

Zamioculcas zamiifolia

Easy

Common

NewPlantName

NewScientificName

Easy

Common

Reset Filters

Add New Plant

Plant name

Scientific name

Easy

Medium

Moderate

Short description

Add Plant

Clear

Pilea peperomioides

(Pilea peperomioides)

View Details

Remove

Coin-shaped leaves, easy to propagate; prefers bright, indirect light.

Easy

Uncommon

Light

Bright

Water

Moderate

© 2024 Mini Nursery.

Banner.tsx

```
import React from "react";

interface BannerProps {
  children: React.ReactNode;
}

const Banner: React.FC<BannerProps> = ({ children }) => {
  return (
    <div className="container-fluid p-5 mb-4 bg-light rounded-3 border">
      <div className="container py-2 text-center">
        {children}
        <p className="lead text-muted">
          Manage your plant collection efficiently and effortlessly.
        </p>
      </div>
    </div>
  );
};

export default Banner;
```

Footer.tsx

```
import React from "react";

const Footer: React.FC = () => {
  return (
    <footer className="text-center py-4 mt-5 border-top text-muted">
      <p>&copy; 2024 Mini Nursery.</p>
    </footer>
  );
};

export default Footer;
```

Header.tsx

```
import React from "react";

const Header: React.FC = () => {
  return (
    <nav className="navbar navbar-light bg-white border-bottom shadow-sm mb-4">
      <div className="container">
        <span className="navbar-brand mb-0 h1">🌿 Mini Plant
Manager</span>
      </div>
    </nav>
  );
};

export default Header;
```

PeopleDropdown.tsx

```
import React, { useEffect, useState } from "react";
import { getUsers, type User } from "../services/UserService";

interface PeopleDropdownProps {
  onSelect: (user: User) => void;
}

const PeopleDropdown: React.FC<PeopleDropdownProps> = ({ onSelect }) => {
  const [users, setUsers] = useState<User[]>([]);
  const [selectedUser, setSelectedUser] = useState<User | null>(null);
  const [loading, setLoading] = useState(true);
  const [error, setError] = useState<string | null>(null);

  useEffect(() => {
```

```
getUsers()
  .then((data) => {
    setUsers(data);
    setLoading(false);
  })
  .catch(() => {
    setError("Failed to load users.");
    setLoading(false);
  });
}, []);

const handleChange = (e: React.ChangeEvent<HTMLSelectElement>) => {
  const user = users.find((u) => u.id === parseInt(e.target.value));
  if (user) {
    setSelectedUser(user);
    onSelect(user);
  }
};

if (loading)
  return <div className="spinner-border text-primary"
role="status"></div>;
if (error) return <div className="alert alert-danger">{error}</div>;

return (
  <div className="mb-4">
    <label className="form-label">Select a person</label>
    <select className="form-select mb-3" onChange={handleChange}>
      <option value="">Choose a user...</option>
      {users.map((user) => (
        <option key={user.id} value={user.id}>
          {user.name} ({user.email})
        </option>
      ))}
    </select>

    {selectedUser && (
      <div className="card bg-light border-0 shadow-sm p-3">
```

```

        <h5>{selectedUser.name}</h5>
        <p className="small mb-1">
            <strong>Email:</strong> {selectedUser.email}
        </p>
        <p className="small mb-1">
            <strong>Phone:</strong> {selectedUser.phone}
        </p>
        <p className="small mb-0">
            <strong>Company:</strong> {selectedUser.company?.name}
        </p>
    </div>
    )}
</div>
);
};

export default PeopleDropdown;

```

PlantCard.tsx

```

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

interface PlantCardProps {
    plant: Plant;
    onDelete: (id: number) => void;
}

const PlantCard: React.FC<PlantCardProps> = ({ plant, onDelete }) => {
    return (
        <div className="card mb-3 shadow-sm">
            <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"
        onClick={() => alert(`Plant: ${plant.name}`)}
    >
        View Details
    </button>
    <button
        className="btn btn-outline-danger btn-sm"
        onClick={() => onDelete(plant.id)}
    >
        Remove
    </button>
</div>
</div>

<p className="card-text mt-3">{plant.description}</p>

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

```

```
        </div>
      </div>
    );
  };

export default PlantCard;
```

PlantForm.tsx

```
import React, { useState } from "react";
import type { Plant } from "../types";

interface PlantFormProps {
  onAdd: (plant: Omit<Plant, "id">) => void;
}

export const PlantForm: React.FC<PlantFormProps> = ({ onAdd }) => {
  const initialState = {
    name: "",
    scientific: "",
    difficulty: "Easy" as const,
    light: "Medium" as const,
    water: "Moderate" as const,
    rarity: "Common" as const,
    description: "",
  };

  const [form, setForm] = useState(initialState);

  const handleSubmit = (e: React.FormEvent) => {
    e.preventDefault();
    if (!form.name) return alert("Plant name is required");
    onAdd(form);
    setForm(initialState);
  };

  return (
```

```

<div className="card p-3 bg-light border-0 shadow-sm mb-4">
  <h4 className="mb-3">Add New Plant</h4>
  <form onSubmit={handleSubmit}>
    <div className="row mb-3">
      <div className="col">
        <input
          type="text"
          className="form-control"
          placeholder="Plant name"
          value={form.name}
          onChange={(e) => setForm({ ...form, name: e.target.value
}}}
        />
      </div>
      <div className="col">
        <input
          type="text"
          className="form-control"
          placeholder="Scientific name"
          value={form.scientific}
          onChange={(e) => setForm({ ...form, scientific:
e.target.value })}
        />
      </div>
    </div>
    <div className="row mb-3">
      <div className="col">
        <select
          className="form-select"
          value={form.difficulty}
          onChange={(e) =>
            setForm({ ...form, difficulty: e.target.value as any
}})
        >
        <option value="Easy">Easy</option>
        <option value="Medium">Medium</option>
        <option value="Hard">Hard</option>

```



```
    </select>
  </div>
  <div className="col">
    <select
      className="form-select"
      value={form.light}
      onChange={(e) =>
        setForm({ ...form, light: e.target.value as any })
      }
    >
      <option value="Low">Low</option>
      <option value="Medium">Medium</option>
      <option value="Bright">Bright</option>
    </select>
  </div>
  <div className="col">
    <select
      className="form-select"
      value={form.water}
      onChange={(e) =>
        setForm({ ...form, water: e.target.value as any })
      }
    >
      <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}
    value={form.description}
    onChange={(e) => setForm({ ...form, description:
e.target.value })}
  />
```

```

    </div>
    <button type="submit" className="btn btn-primary me-2">
      Add Plant
    </button>
    <button
      type="button"
      className="btn btn-outline-secondary"
      onClick={() => setForm(initialState)}
    >
      Clear
    </button>
  </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"

```

```

        className={`list-group-item list-group-item-action
${plant.id === selectedId ? 'active' : ''}`}
        onClick={() => onSelect(plant.id)}
        style={{ cursor: 'pointer' }}
      >
        <div className="d-flex w-100 justify-content-between align-
items-center mb-1">
          <h5 className="mb-1">{plant.name}</h5>
          <div>

            <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>
        </div>
        <p className="mb-1 small">{plant.scientific}</p>
      </button>
    )}}
  </div>
);
};

export default PlantList;

```

App.tsx

```

import React, { useState, useMemo } from "react";
import type { Plant } from "./types";
import PlantList from "./PlantList";
import PlantCard from "./PlantCard";
import { PlantForm } from "./PlantForm";
import Header from "./components/Header";
import Footer from "./components/Footer";

```

```
import Banner from "../components/Banner";
import PeopleDropdown from "../components/PeopleDropdown";

interface AppProps {
  plants: Plant[];
}

const App: React.FC<AppProps> = ({ plants: initialPlants }) => {
  const [plants, setPlants] = useState<Plant[]>(initialPlants);
  const [selectedId, setSelectedId] =
    useState<number>(initialPlants[0]?.id);
  const [search, setSearch] = useState("");
  const [difficultyFilter, setDifficultyFilter] = useState("All");

  const filteredPlants = useMemo(() => {
    return plants.filter(
      (p) =>
        p.name.toLowerCase().includes(search.toLowerCase()) &&
        (difficultyFilter === "All" || p.difficulty ===
difficultyFilter)
    );
  }, [plants, search, difficultyFilter]);

  const selectedPlant = plants.find((p) => p.id === selectedId);

  const handleAdd = (newPlant: Omit<Plant, "id">) => {
    const plantWithId = { ...newPlant, id: Date.now() };
    setPlants([...plants, plantWithId]);
  };

  const handleDelete = (id: number) => {
    setPlants(plants.filter((p) => p.id !== id));
    if (selectedId === id) setSelectedId(plants[0]?.id);
  };

  return (
    <>
      <Header />
```

```
<Banner>
  <h1>Welcome to Your Mini Nursery</h1>
</Banner>
<div className="container px-4">
  <div className="row">
    <div className="col-md-4">
      <input
        type="text"
        className="form-control mb-2"
        placeholder="Search plants..."
        value={search}
        onChange={(e) => setSearch(e.target.value)}
      />
      <select
        className="form-select mb-3"
        value={difficultyFilter}
        onChange={(e) => setDifficultyFilter(e.target.value)}
      >
        <option value="All">All Difficulties</option>
        <option value="Easy">Easy</option>
        <option value="Medium">Medium</option>
        <option value="Hard">Hard</option>
      </select>
      <PeopleDropdown onSelect={(u) => console.log(u)} />
      <PlantList
        plants={filteredPlants}
        selectedId={selectedId}
        onSelect={setSelectedId}
      />
      <button
        className="btn btn-secondary w-100 mt-3"
        onClick={() => {
          setSearch("");
          setDifficultyFilter("All");
        }}
      >
        Reset Filters
      </button>
```

```

        </div>
        <div className="col-md-8">
          <PlantForm onAdd={handleAdd} />
          {selectedPlant && (
            <PlantCard plant={selectedPlant} onDelete={handleDelete}
          />
          )}
        </div>
      </div>
      <Footer />
    </>
  );
};

export default App;

```

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';
import "bootstrap/dist/js/bootstrap.bundle.min";

const plants: Plant[] = [
  {
    id: 1,
    name: "Pilea peperomioides",
    scientific: "Pilea peperomioides",
    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>,
)
```