--- player\_data.json ---

[  
 { "rank": 1, "name": "Huy Nam", "score": 9999 },  
 { "rank": 2, "name": "thùy vân", "score": 1400 },  
 { "rank": 3, "name": "David minh nhat", "score": 1300 },  
 { "rank": 4, "name": "David", "score": 1200 },  
 { "rank": 5, "name": "Eva", "score": 1100 }  
]

--- web\_notification.json ---

[  
 {  
 "id": 1,  
 "title": "Cập nhật hệ thống",  
 "message": "Chúng tôi vừa cập nhật các tính năng mới cho trải nghiệm tốt hơn.",  
 "date": "2025-05-26T15:30:00Z"  
 },  
 {  
 "id": 2,  
 "title": "Bảo trì định kỳ",  
 "message": "Hệ thống sẽ bảo trì vào lúc 02:00 AM ngày mai.",  
 "date": "2025-05-27T01:00:00Z"  
 },  
 {  
 "id": 3,  
 "title": "Thông báo!Game sắp có bản cập nhật lớn",  
 "message": "Phiên bản 0.0.0.0.0.0.1",  
 "date": "2025-05-29T12:30:00Z"  
 }  
]

--- App.css ---

/\* ==========================================================================  
 Global Settings: Reset & Base Styles  
 ========================================================================== \*/  
\* {  
 box-sizing: border-box;  
}  
  
body {  
 margin: 0;  
 font-family: Arial, Helvetica, sans-serif;  
}  
  
/\* ==========================================================================  
 Header Section  
 ========================================================================== \*/  
.header {  
 background-color: #f1f1f1;  
 padding: 20px;  
 text-align: center;  
}  
  
/\* ==========================================================================  
 Top Navigation Bar  
 ========================================================================== \*/  
.topnav {  
 overflow: hidden;  
 background-color: #333;  
}  
  
.topnav a {  
 float: left;  
 display: block;  
 color: #f2f2f2;  
 text-align: center;  
 padding: 14px 16px;  
 text-decoration: none;  
}  
  
.topnav a:hover {  
 background-color: #ddd;  
 color: black;  
}  
  
  
/\* ==========================================================================  
 Layout: Row and Columns  
 ========================================================================== \*/  
/\* Container for columns \*/  
.row {  
 margin: 0;  
 padding: 10px;  
}  
  
/\* Base column styles \*/  
.column {  
 float: left;  
 padding: 10px;  
}  
  
/\* Left Column (Control Panel) \*/  
.column.left {  
 width: 10%;  
 background-color: #ccc;  
}  
  
/\* Middle Column (Main Scene) \*/  
.column.middle {  
 width: 65%;  
 background-color: #f1f1f1;  
}  
  
/\* Right Column (Leaderboard) \*/  
.column.right {  
 width: 25%;  
 background-color: #ddd;  
}  
  
/\* Clear floats \*/  
.row::after {  
 content: "";  
 display: table;  
 clear: both;  
}  
  
/\* ==========================================================================  
 Footer Section  
 ========================================================================== \*/  
footer {  
 background-color: #777;  
 padding: 10px;  
 text-align: center;  
 color: white;  
}  
  
/\* ==========================================================================  
 Leaderboard Table Styling  
 ========================================================================== \*/  
table {  
 width: 100%;  
 border-collapse: collapse;  
}  
  
th, td {  
 text-align: center;  
 padding: 8px;  
 border: 1px solid #ddd;  
}  
  
th {  
 background-color: #f2f2f2;  
}  
  
  
/\* ==========================================================================  
 Fixed Footage Banner (Always at the Bottom)  
 ========================================================================== \*/  
.footage {  
 position: fixed;  
 bottom: 0;  
 left: 0;  
 right: 0;  
 background-color: #444;  
 color: #fff;  
 padding: 10px;  
 text-align: center;  
 z-index: 1000;  
}  
  
  
/\* ==========================================================================  
 Responsive Styles  
 ========================================================================== \*/  
@media screen and (max-width: 600px) {  
 /\* Stack columns on small screens \*/  
 .column.side, .column.middle {  
 width: 100%;  
 }  
   
 /\* Adjust fixed footage banner on small screens \*/  
 .footage {  
 font-size: 0.9em;  
 padding: 8px;  
 }  
}  
  
/\* ==========================================================================  
 Style cho ảnh "404 Not Found" trong main game scene  
 ========================================================================== \*/  
.game-scene {  
 text-align: center;  
 margin: 20px auto;  
}  
  
.not-found-image {  
 max-width: 100%;  
 height: auto;  
 border: 1px solid #ddd;  
}  
  
/\* ==========================================================================  
 Sidebar.css  
 ========================================================================== \*/  
.sidebar {  
 background-color: inherit; /\* Kế thừa màu nền từ cột cha \*/  
 padding: 10px;  
 /\* Bỏ width, float vì Sidebar sẽ nằm trong .column.left \*/  
}  
  
.sidebar h2 {  
 margin-top: 0;  
}  
  
/\* Các style khác cho nút riêng \*/  
.sidebar ul {  
 list-style: none;  
 padding: 0;  
 margin: 0;  
}  
  
.sidebar ul li {  
 margin-bottom: 10px;  
}  
  
.sidebar button {  
 cursor: pointer;  
 background-color: #fff;  
 border: 1px solid #999;  
 padding: 8px 12px;  
 width: 100%;  
 text-align: left;  
 font-size: 1em;  
 transition: background-color 0.3s, color 0.3s, transform 0.2s;  
 border-radius: 4px;  
}  
  
.sidebar button:hover {  
 background-color: #ddd;  
 color: #333;  
 transform: scale(1.02);  
}

--- App.jsx ---

// App.jsx  
import React from "react";  
import "./App.css";  
import Footage from "./Footage";  
import Sidebar from "./Sidebar"; // Đây là Control Panel cho bên trái  
import Leaderboard from "./Leaderboard"; // Đây là Leaderboard cho bên phải  
import TopNav from "./TopNav"; // Top navigation  
  
function App() {  
 return (  
 <div className="App">  
 <header>  
 <h1>Webapp giúp dự đoán dân số theo khu vực với nhiều tiêu chí</h1>  
 </header>  
  
 {/\* Top Navigation Bar \*/}  
 <TopNav />  
  
 <div className="row">  
 {/\* Cột bên trái (Control Panel) \*/}  
 <div className="column left">  
 <Sidebar />  
 </div>  
  
 {/\* Cột giữa (Main Scene) \*/}  
 <div className="column middle">  
 <div className="main-scene">  
 <img  
 src="https://bizmac.com/Images/Editor/images/404-not-found-la-gi.jpg"  
 alt="404 Not Found"  
 className="not-found-image"  
 />  
 </div>  
 </div>  
  
 {/\* Cột bên phải (Leaderboard) \*/}  
 <div className="column right">  
 <Leaderboard />  
 </div>  
 </div>  
   
 {/\* Footage Component nằm dưới cùng \*/}  
 <Footage />  
 </div>  
 );  
}  
  
export default App;

--- Footage.jsx ---

import React from "react";  
import "./App.css";  
  
const Footage = () => {  
 return (  
 <div className="footage">  
 <p>Webapp này là sản phẩm môn học Khoa học dữ liệu</p>  
 <p>© 2025 Population Insight-Website dự đoán dân số | made by HaHuyNam</p>  
 </div>  
 );  
};  
  
export default Footage;

--- Leaderboard.jsx ---

// Ví dụ component Leaderboard.jsx  
import React, { useEffect, useState } from "react";  
  
const Leaderboard = () => {  
 const [players, setPlayers] = useState([]);  
  
 useEffect(() => {  
 fetch("http://localhost:5003/players")  
 .then((res) => res.json())  
 .then((data) => {  
 setPlayers(data);  
 })  
 .catch((error) => console.error("Error fetching players:", error));  
 }, []);  
  
 return (  
 <div>  
 <h2>Leaderboard</h2>  
 <table>  
 <thead>  
 <tr>  
 <th>Rank</th>  
 <th>Name</th>  
 <th>Score</th>  
 </tr>  
 </thead>  
 <tbody>  
 {players.map((player) => (  
 <tr key={player.rank}>  
 <td>{player.rank}</td>  
 <td>{player.name}</td>  
 <td>{player.score}</td>  
 </tr>  
 ))}  
 </tbody>  
 </table>  
 </div>  
 );  
};  
  
export default Leaderboard;

--- main.jsx ---

import { StrictMode } from 'react'  
import { createRoot } from 'react-dom/client'  
import App from './App.jsx'  
createRoot(document.getElementById('root')).render(  
 <StrictMode>  
 <App />  
 </StrictMode>,  
)

--- Sidebar.jsx ---

import React from "react";  
import "./App.css";  
  
const Sidebar = () => {  
 // Hàm xử lý sự kiện khi nhấn nút, bạn có thể thay thế bằng logic của riêng mình.  
 const handleClick = (menuItem) => {  
 console.log(`Bạn vừa nhấn: ${menuItem}`);  
 // Thêm logic điều hướng hay xử lý ở đây  
 };  
  
 return (  
 <div className="column side sidebar">  
 <h2>Control Panel - Menu</h2>  
 <ul>  
 <li>  
 <button type="button" onClick={() => handleClick("chuc\_nang\_1")}>  
 Chức năng 1  
 </button>  
 </li>  
 <li>  
 <button type="button" onClick={() => handleClick("chuc\_nang\_1")}>  
 Chức năng 2  
 </button>  
 </li>  
 </ul>  
 </div>  
 );  
};  
  
export default Sidebar;

--- TopNav.jsx ---

// TopNav.jsx  
import React, { useState, useEffect } from "react";  
import "./App.css";  
  
function TopNav() {  
 const [modalVisible, setModalVisible] = useState(false);  
 const [notifications, setNotifications] = useState([]);  
 const [hasNewNotifications, setHasNewNotifications] = useState(false);  
  
 // Khi modal được mở, fetch thông báo từ API  
 useEffect(() => {  
 if (modalVisible) {  
 fetch("http://localhost:5003/notifications")  
 .then((res) => res.json())  
 .then((data) => {  
 setNotifications(data);  
 // Nếu có thông báo thì đánh dấu có thông báo mới  
 setHasNewNotifications(data && data.length > 0);  
 })  
 .catch((error) =>  
 console.error("Error fetching notifications:", error)  
 );  
 }  
 }, [modalVisible]);  
  
 // Hàm chuyển đổi trạng thái hiển thị modal;  
 // nếu mở modal, đánh dấu các thông báo là đã xem (badge ẩn đi)  
 const toggleModal = () => {  
 if (!modalVisible) {  
 setHasNewNotifications(false);  
 }  
 setModalVisible(!modalVisible);  
 };  
  
 // Hàm xóa thông báo khỏi state khi nhấn nút X  
 const handleDeleteNotification = (id) => {  
 setNotifications((prevNotifications) => {  
 const filtered = prevNotifications.filter(  
 (notification) => notification.id !== id  
 );  
 // Nếu đã xóa hết, đánh dấu không còn thông báo mới  
 if (filtered.length === 0) {  
 setHasNewNotifications(false);  
 }  
 return filtered;  
 });  
 };  
  
 return (  
 <>  
 <div className="topnav d-flex align-items-center position-relative">  
 <a href="#dang-nhap">Đăng nhập</a>  
 <a href="#dang-ky">Đăng ký</a>  
 <a href="#nap-tien">Nạp tiền</a>  
 {/\* Nút hiển thị biểu tượng chuông \*/}  
 <button  
 onClick={toggleModal}  
 className="btn btn-link ms-auto position-relative"  
 style={{ border: "none", background: "none" }}  
 >  
 <i className="fa fa-bell" style={{ fontSize: "24px" }}></i>  
 {/\* Nếu có thông báo mới, hiển thị badge đỏ \*/}  
 {hasNewNotifications && (  
 <span  
 className="position-absolute top-0 start-100 translate-middle p-1 bg-danger border border-light rounded-circle"  
 style={{ width: "10px", height: "10px" }}  
 >  
 <span className="visually-hidden">New alerts</span>  
 </span>  
 )}  
 </button>  
 </div>  
  
 {/\* Modal hiển thị danh sách thông báo \*/}  
 {modalVisible && (  
 <>  
 <div  
 className="modal show fade"  
 style={{ display: "block" }}  
 tabIndex="-1"  
 >  
 <div className="modal-dialog">  
 <div className="modal-content">  
 <div className="modal-header">  
 <h5 className="modal-title">Thông báo hệ thống</h5>  
 <button  
 type="button"  
 className="btn-close"  
 aria-label="Close"  
 onClick={toggleModal}  
 ></button>  
 </div>  
 <div className="modal-body">  
 {notifications.length > 0 ? (  
 <ul className="list-unstyled">  
 {notifications.map((note) => (  
 <li  
 key={note.id}  
 className="d-flex justify-content-between align-items-center mb-2"  
 >  
 <div>  
 <strong>{note.title}</strong>: {note.message}  
 <br />  
 <small>  
 {new Date(note.date).toLocaleString("vi-VN")}  
 </small>  
 </div>  
 <button  
 className="btn btn-sm btn-danger ms-2"  
 onClick={() => handleDeleteNotification(note.id)}  
 >  
 X  
 </button>  
 </li>  
 ))}  
 </ul>  
 ) : (  
 <p>Không có thông báo mới.</p>  
 )}  
 </div>  
 <div className="modal-footer">  
 <button  
 type="button"  
 className="btn btn-secondary"  
 onClick={toggleModal}  
 >  
 Đóng  
 </button>  
 </div>  
 </div>  
 </div>  
 </div>  
 {/\* Overlay của modal \*/}  
 <div className="modal-backdrop fade show" onClick={toggleModal}></div>  
 </>  
 )}  
 </>  
 );  
}  
  
export default TopNav;

--- .env ---

PORT=5003  
NODE\_ENV=development

--- eslint.config.js ---

import js from '@eslint/js'  
import globals from 'globals'  
import reactHooks from 'eslint-plugin-react-hooks'  
import reactRefresh from 'eslint-plugin-react-refresh'  
  
export default [  
 { ignores: ['dist'] },  
 {  
 files: ['\*\*/\*.{js,jsx}'],  
 languageOptions: {  
 ecmaVersion: 2020,  
 globals: globals.browser,  
 parserOptions: {  
 ecmaVersion: 'latest',  
 ecmaFeatures: { jsx: true },  
 sourceType: 'module',  
 },  
 },  
 plugins: {  
 'react-hooks': reactHooks,  
 'react-refresh': reactRefresh,  
 },  
 rules: {  
 ...js.configs.recommended.rules,  
 ...reactHooks.configs.recommended.rules,  
 'no-unused-vars': ['error', { varsIgnorePattern: '^[A-Z\_]' }],  
 'react-refresh/only-export-components': [  
 'warn',  
 { allowConstantExport: true },  
 ],  
 },  
 },  
]

--- index.html ---

<!doctype html>  
<html lang="vi">  
  
<head>  
 <meta charset="UTF-8" />  
 <link rel="icon" type="image/png"  
 href="https://dcassetcdn.com/design\_img/3696967/549092/22052230/ezvehp4nj9a8ksb82rw6ek0jn9\_image.png" />  
 <meta name="viewport" content="width=device-width, initial-scale=1.0" />  
 <title>Population Insight-Website dự đoán dân số</title>  
  
 <!-- FontAwesome -->  
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/all.min.css"  
 integrity="sha512-iecdLmaskl7CVkqkXNQ/ZH/XLlvWZOJyj7Yy7tcenmpD1ypASozpmT/E0iPtmFIB46ZmdtAc9eNBvH0H/ZpiBw=="  
 crossorigin="anonymous" referrerpolicy="no-referrer" />  
  
 <!-- Bootstrap 5 CSS -->  
 <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet"  
 crossorigin="anonymous" />  
  
 <!-- jQuery (Bootstrap 5 không yêu cầu jQuery nhưng nếu bạn cần dùng jQuery thì thêm như sau) -->  
 <script src="https://code.jquery.com/jquery-3.6.0.min.js" crossorigin="anonymous">  
 </script>  
</head>  
  
<body>  
 <div id="root"></div>  
  
 <!-- Bootstrap 5 Bundle JS (bao gồm Popper) -->  
 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js" crossorigin="anonymous">  
 </script>  
  
 <!-- Entry point của ứng dụng -->  
 <script type="module" src="/src/main.jsx"></script>  
</body>  
  
</html>

--- server.js ---

// server.js (theo cú pháp ES module)  
  
import dotenv from "dotenv";  
dotenv.config();  
  
import express from "express";  
import fs from "fs";  
import path from "path";  
import cors from "cors";  
import { fileURLToPath } from "url";  
  
// Tính \_\_dirname cho ES Module  
const \_\_filename = fileURLToPath(import.meta.url);  
const \_\_dirname = path.dirname(\_\_filename);  
  
const app = express();  
const port = process.env.PORT || 5003;  
  
app.use(cors());  
  
// Đường dẫn file player\_data.json  
const playerDataPath = path.join(\_\_dirname, "src", "assets", "player\_data.json");  
  
// Định nghĩa endpoint GET /players  
app.get("/players", (req, res) => {  
 fs.readFile(playerDataPath, "utf8", (err, data) => {  
 if (err) {  
 console.error("Error reading file:", err);  
 return res.status(500).json({ error: "Error reading file" });  
 }  
 try {  
 const players = JSON.parse(data);  
 res.json(players);  
 } catch (parseErr) {  
 console.error("Error parsing JSON:", parseErr);  
 res.status(500).json({ error: "Error parsing JSON" });  
 }  
 });  
});  
  
// Đường dẫn file web\_notification.json  
const notificationDataPath = path.join(\_\_dirname, "src", "assets", "web\_notification.json");  
  
// Định nghĩa endpoint GET /notifications  
app.get("/notifications", (req, res) => {  
 fs.readFile(notificationDataPath, "utf8", (err, data) => {  
 if (err) {  
 console.error("Error reading notifications file:", err);  
 return res.status(500).json({ error: "Error reading file" });  
 }  
 try {  
 const notifications = JSON.parse(data);  
 res.json(notifications);  
 } catch (parseErr) {  
 console.error("Error parsing notifications JSON:", parseErr);  
 res.status(500).json({ error: "Error parsing JSON" });  
 }  
 });  
});  
  
app.listen(port, () => {  
 console.log(`Express server is running on http://localhost:${port}`);  
});

--- vite.config.js ---

import { defineConfig } from 'vite'  
import react from '@vitejs/plugin-react'  
export default defineConfig({  
 server: {  
 // Cấu hình cổng 5002  
 port: 5002,  
 // Cho phép CORS  
 cors: true,   
 },  
 // Plugin react giúp chuyển đổi mã React khi chạy trên Vite  
 plugins: [react()],  
})