<!DOCTYPE html>

<html lang="vi">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Cẩm Nang Ôn Thi THPT Quốc Gia</title>

    <script src="https://cdn.tailwindcss.com"></script>

    <link rel="preconnect" href="https://fonts.googleapis.com">

    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>

    <link href="https://fonts.googleapis.com/css2?family=Inter:wght@400;500;600;700&display=swap" rel="stylesheet">

    <script src="https://www.gstatic.com/firebasejs/11.6.1/firebase-app-compat.js"></script>

    <script src="https://www.gstatic.com/firebasejs/11.6.1/firebase-auth-compat.js"></script>

    <style>

        body {

            font-family: 'Inter', sans-serif;

            background-color: #f1f5f9;

        }

        .subject-btn.active {

            background-color: #2563eb;

            color: white;

            box-shadow: 0 4px 6px -1px rgb(0 0 0 / 0.1), 0 2px 4px -2px rgb(0 0 0 / 0.1);

        }

        .grade-btn.active {

            background-color: #4f46e5;

            color: white;

        }

        .content-panel {

            display: none;

        }

        .content-panel.active {

            display: grid;

        }

        .task-item input:checked + label {

            text-decoration: line-through;

            color: #6b7280;

        }

    </style>

</head>

<body class="antialiased text-slate-800">

    <div class="container mx-auto px-4 py-8 md:py-12">

        <header class="text-center mb-8">

            <h1 class="text-3xl md:text-4xl font-bold text-slate-900">Cẩm Nang Ôn Thi THPT Quốc Gia</h1>

            <p class="mt-2 text-slate-600 max-w-2xl mx-auto">Hệ thống kiến thức trọng tâm các môn Toán, Vật lí, Hóa học. Chọn môn và khối lớp để bắt đầu ôn tập.</p>

        </header>

        <main>

            <div id="controls" class="mb-8 p-4 bg-white rounded-xl shadow-md">

                <div class="mb-4">

                    <h2 class="text-lg font-semibold mb-3 text-slate-700">Chọn môn học:</h2>

                    <div id="subject-selector" class="flex flex-wrap gap-3">

                        <button data-subject="toan" class="subject-btn flex-1 md:flex-none justify-center text-center px-6 py-3 font-semibold rounded-lg bg-slate-200 hover:bg-blue-500 hover:text-white transition-all duration-300">

                            <span class="text-2xl mr-2">🧮</span> Toán học

                        </button>

                        <button data-subject="ly" class="subject-btn flex-1 md:flex-none justify-center text-center px-6 py-3 font-semibold rounded-lg bg-slate-200 hover:bg-blue-500 hover:text-white transition-all duration-300">

                            <span class="text-2xl mr-2">⚛️</span> Vật lí

                        </button>

                        <button data-subject="hoa" class="subject-btn flex-1 md:flex-none justify-center text-center px-6 py-3 font-semibold rounded-lg bg-slate-200 hover:bg-blue-500 hover:text-white transition-all duration-300">

                            <span class="text-2xl mr-2">🧪</span> Hóa học

                        </button>

                    </div>

                </div>

                <div>

                    <h2 class="text-lg font-semibold mb-3 text-slate-700">Chọn khối lớp:</h2>

                    <div id="grade-selector" class="flex flex-wrap gap-3">

                        <button data-grade="10" class="grade-btn flex-1 md:flex-none justify-center text-center px-5 py-2 font-medium rounded-md bg-white border border-slate-300 hover:bg-indigo-500 hover:text-white transition-all duration-200">Lớp 10</button>

                        <button data-grade="11" class="grade-btn flex-1 md:flex-none justify-center text-center px-5 py-2 font-medium rounded-md bg-white border border-slate-300 hover:bg-indigo-500 hover:text-white transition-all duration-200">Lớp 11</button>

                        <button data-grade="12" class="grade-btn flex-1 md:flex-none justify-center text-center px-5 py-2 font-medium rounded-md bg-white border border-slate-300 hover:bg-indigo-500 hover:text-white transition-all duration-200">Lớp 12</button>

                    </div>

                </div>

            </div>

            <div id="content-container">

            </div>

            <div id="generated-content-container" class="mt-8 p-6 bg-white rounded-xl shadow-md hidden">

                <div class="flex items-center justify-between mb-4">

                    <h2 id="generated-content-title" class="text-2xl font-bold text-slate-900"></h2>

                    <button id="close-generated-content" class="text-slate-500 hover:text-slate-900 transition-colors">

                        <span class="text-2xl font-bold">&times;</span>

                    </button>

                </div>

                <div id="loading-indicator" class="text-center text-slate-500 hidden">

                    <div class="animate-spin inline-block w-8 h-8 border-4 border-t-transparent border-blue-500 rounded-full"></div>

                    <p class="mt-2">Đang tạo nội dung...</p>

                </div>

                <div id="generated-content-body" class="prose max-w-none"></div>

            </div>

        </main>

    </div>

    <script>

        const curriculumData = {

            toan: {

                10: {

                    "Đại số và Thống kê": [

                        "Chương 1: Mệnh đề toán học. Tập hợp",

                        "Chương 2: Bất phương trình và hệ bất phương trình bậc nhất hai ẩn",

                        "Chương 3: Hàm số và đồ thị",

                        "Chương 6: Một số yếu tố thống kê và xác suất"

                    ],

                    "Hình học và Tổ hợp": [

                        "Chương 4: Hệ thức lượng trong tam giác. Vectơ",

                        "Chương 5: Đại số tổ hợp",

                        "Chương 7: Phương pháp tọa độ trong mặt phẳng"

                    ],

                    "Chuyên đề": [

                        "Hệ phương trình bậc nhất ba ẩn",

                        "Phương pháp quy nạp toán học. Nhị thức Newton",

                        "Ba đường conic và ứng dụng"

                    ]

                },

                11: {

                    "Đại số và Giải tích": [

                        "Chương 1: Hàm số lượng giác và phương trình lượng giác",

                        "Chương 2: Dãy số. Cấp số cộng và cấp số nhân",

                        "Chương 3: Giới hạn hàm số liên tục",

                        "Chương 6: Hàm số mũ và hàm số logarit",

                        "Chương 7: Đạo hàm"

                    ],

                    "Hình học không gian": [

                        "Chương 4: Đường thẳng và mặt phẳng trong không gian .quan hệ song song",

                        "Chương 8: Quan hệ vuông góc trong không gian. Phép chiếu vuông góc"

                    ],

                    "Xác suất và Thống kê": [

                        "Chương 5: Một số yếu tố thống kê và xác suất"

                    ],

                    "Chuyên đề": [

                        "Phép biến hình phẳng",

                        "Làm quen với một vài yếu tố của lí thuyết đô thị",

                        "Một số yếu tố vẽ kĩ thuật"

                    ]

                },

                12: {

                    "Giải tích": [

                        "Chương 1: Ứng dụng đạo hàm để khảo sát hàm số và vẽ đồ thị của hàm số",

                        "Chương 4: Nguyên hàm tích phân"

                    ],

                    "Hình học": [

                        "Chương 2: Tọa độ của vecto trong không gian",

                        "Chương 5: Phương trình mặt phẳng, đường thẳng, mặt cầu trong không gian"

                    ],

                    "Thống kê và Xác suất": [

                        "Chương 3: Các số đặc trưng cho mức độ phân tán cho mẫu số liệu ghép nhóm",

                        "Chương 6: Một số yếu tố xác suất"

                    ],

                    "Chuyên đề": [

                        "Biến ngẫu nhiên rời rạc. Các số đặc trưng của biến ngẫu nhiên rời rạc",

                        "Ứng dụng toán học để giải quyết một số bài toán tối ưu",

                        "Ứng dụng toán học trong một số vấn đề liên quan đến tài chính"

                    ]

                }

            },

            ly: {

                10: {

                    "Động học và Động lực học": [

                        "Chương 1: Mở đầu",

                        "Chương 2: Mô tả chuyển động",

                        "Chương 3: Chuyển động biến đổi",

                        "Chương 4: Ba định luật Newton, một số lực trong thực tiễn"

                    ],

                    "Cân bằng và Bảo toàn": [

                        "Chương 5: Moment lực, điều kiện cân bằng của vật rắn",

                        "Chương 6: Năng lượng",

                        "Chương 7: Động lượng"

                    ],

                    "Các loại chuyển động và Biến dạng": [

                        "Chương 8: Chuyển động tròn đều",

                        "Chương 9: Biến dạng của vật rắn"

                    ],

                    "Chuyên đề": [

                        "Vật lí trong một số nghành nghề",

                        "Trái đất và bầu trời",

                        "Vật lí giáo dục về bảo vệ bầu trời"

                    ]

                },

                11: {

                    "Cơ học và Dao động": [

                        "Chương 1: Mô tả dao động",

                        "Chương 2: Sóng"

                    ],

                    "Điện học và Từ trường": [

                        "Chương 3: Điện trường",

                        "Chương 4: Dòng điện không đổi"

                    ],

                    "Chuyên đề": [

                        "Trường hấp dẫn",

                        "Truyền thông tin bằng sóng vô tuyến",

                        "Mở đầu về điện tử học"

                    ]

                },

                12: {

                    "Vật lý nhiệt": [

                        "Chương 1: Vật lí nhiệt",

                        "Chương 2: Khí lí tưởng"

                    ],

                    "Từ trường và Hạt nhân": [

                        "Chương 3: Từ trường",

                        "Chương 4: Vật lí hạt nhân"

                    ],

                    "Chuyên đề": [

                        "Dòng điện xoay chiều",

                        "Một số ứng dụng vật lí trong chuẩn đoán y học",

                        "Vật lí lượng tử"

                    ]

                }

            },

            hoa: {

                10: {

                    "Cấu tạo và Liên kết": [

                        "Phần mở đầu",

                        "Chương 1: Cấu tạo nguyên tử",

                        "Chương 2: Bảng tuần hoàn các nguyên tố hóa học",

                        "Chương 3: Liên kết hóa học"

                    ],

                    "Phản ứng và Tốc độ": [

                        "Chương 4: Oxi hóa-khử",

                        "Chương 5: Năng lượng hóa học",

                        "Chương 6: Tốc độ phản ứng hóa học"

                    ],

                    "Nhóm halogen": [

                        "Chương 7: Nguyên tố nhóm VIIA-halogen"

                    ],

                    "Chuyên đề": [

                        "Cơ sở khoa học",

                        "Hóa học trong việc phòng chống cháy nổ",

                        "Thực hành hóa học và công nghệ thông tin"

                    ]

                },

                11: {

                    "Hóa học Vô cơ": [

                        "Chương 1: Cân bằng hóa học",

                        "Chương 2: Nitrogen-Sulfur"

                    ],

                    "Hóa học Hữu cơ": [

                        "Chương 3: Đại cương về hóa học hữu cơ",

                        "Chương 4: Hydrocarbon",

                        "Chương 5: Dẫn xuất halogen-alcohol-phenol",

                        "Chương 6: Hợp chất carbonyl-carboxylic acid"

                    ],

                    "Chuyên đề": [

                        "Phân bón",

                        "Trải nghiệm, thực hành hóa học hữu cơ",

                        "Dầu mỏ và chế biến dầu mỏ"

                    ]

                },

                12: {

                    "Hóa học Hữu cơ": [

                        "Chương 1: Este - Lipit. Xà phòng và chất giặt rửa",

                        "Chương 2: Cacbohiđrat",

                        "Chương 3: Hợp chất chứa nitơ",

                        "Chương 4: Polyme"

                    ],

                    "Kim loại và Điện hóa": [

                        "Chương 5: Pin điện và điện phân",

                        "Chương 6: Đại cương về kim loại",

                        "Chương 7: Nguyên tố nhóm IA và nhóm IIA",

                        "Chương 8: Sơ lược về dãy kim loại chuyển tiếp thứ nhất và phức chất"

                    ],

                    "Chuyên đề": [

                        "Cơ chế phản ứng trong hóa hữu cơ",

                        "Trải nghiệm thực hành hóa học vô cơ",

                        "Một số vấn đề cơ bản về phức chất"

                    ]

                }

            }

        };

        const subjectSelector = document.getElementById('subject-selector');

        const gradeSelector = document.getElementById('grade-selector');

        const contentContainer = document.getElementById('content-container');

        const generatedContentContainer = document.getElementById('generated-content-container');

        const generatedContentTitle = document.getElementById('generated-content-title');

        const generatedContentBody = document.getElementById('generated-content-body');

        const loadingIndicator = document.getElementById('loading-indicator');

        const closeButton = document.getElementById('close-generated-content');

        let currentSubject = 'ly';

        let currentGrade = '12';

        const firebaseConfig = typeof \_\_firebase\_config !== 'undefined' ? JSON.parse(\_\_firebase\_config) : {};

        let app, auth;

        if (Object.keys(firebaseConfig).length > 0) {

            app = firebase.initializeApp(firebaseConfig);

            auth = firebase.auth();

            const initialAuthToken = typeof \_\_initial\_auth\_token !== 'undefined' ? \_\_initial\_auth\_token : null;

            if (initialAuthToken) {

                auth.signInWithCustomToken(initialAuthToken);

            } else {

                auth.signInAnonymously();

            }

        }

        async function geminiCall(prompt, isJson = false) {

            let chatHistory = [];

            chatHistory.push({ role: "user", parts: [{ text: prompt }] });

            let payload = { contents: chatHistory };

            if (isJson) {

                payload.generationConfig = {

                    responseMimeType: "application/json",

                    responseSchema: {

                        "type": "array",

                        "items": {

                            "type": "object",

                            "properties": {

                                "question": { "type": "string" },

                                "solution": { "type": "string" }

                            },

                            "propertyOrdering": ["question", "solution"]

                        }

                    }

                };

            }

            const apiKey = "";

            const apiUrl = `https://generativelanguage.googleapis.com/v1beta/models/gemini-2.5-flash-preview-05-20:generateContent?key=${apiKey}`;

            let response;

            let retryCount = 0;

            const maxRetries = 5;

            const delay = (ms) => new Promise(res => setTimeout(res, ms));

            while (retryCount < maxRetries) {

                try {

                    response = await fetch(apiUrl, {

                        method: 'POST',

                        headers: { 'Content-Type': 'application/json' },

                        body: JSON.stringify(payload)

                    });

                    if (response.status === 429) {

                        retryCount++;

                        await delay(Math.pow(2, retryCount) \* 1000);

                        continue;

                    }

                    if (!response.ok) throw new Error(`HTTP error! status: ${response.status}`);

                    break;

                } catch (error) {

                    retryCount++;

                    await delay(Math.pow(2, retryCount) \* 1000);

                }

            }

            if (!response || !response.ok) {

                throw new Error("Failed to fetch from API after multiple retries.");

            }

            const result = await response.json();

            if (result.candidates && result.candidates.length > 0 &&

                result.candidates[0].content && result.candidates[0].content.parts &&

                result.candidates[0].content.parts.length > 0) {

                const text = result.candidates[0].content.parts[0].text;

                if (isJson) {

                    return JSON.parse(text);

                }

                return text;

            } else {

                return "Xin lỗi, không thể tạo nội dung vào lúc này. Vui lòng thử lại sau.";

            }

        }

        async function getSummary(topic) {

            generatedContentContainer.classList.remove('hidden');

            generatedContentTitle.textContent = `Tóm tắt kiến thức: ${topic}`;

            generatedContentBody.innerHTML = '';

            loadingIndicator.classList.remove('hidden');

            const prompt = `Tóm tắt kiến thức trọng tâm của chủ đề "${topic}" trong môn ${currentSubject === 'toan' ? 'Toán học' : currentSubject === 'ly' ? 'Vật lí' : 'Hóa học'} khối ${currentGrade} một cách ngắn gọn, súc tích và dễ hiểu.`;

            try {

                const summary = await geminiCall(prompt);

                generatedContentBody.innerHTML = `<p>${summary}</p>`;

            } catch (error) {

                generatedContentBody.innerHTML = `<p class="text-red-500">Đã xảy ra lỗi khi tạo tóm tắt. Vui lòng thử lại.</p>`;

            } finally {

                loadingIndicator.classList.add('hidden');

            }

        }

        async function getPracticeProblems(topic) {

            generatedContentContainer.classList.remove('hidden');

            generatedContentTitle.textContent = `Bài tập thực hành: ${topic}`;

            generatedContentBody.innerHTML = '';

            loadingIndicator.classList.remove('hidden');

            const prompt = `Tạo 3 bài tập tự luận có đáp án chi tiết về chủ đề "${topic}" trong môn ${currentSubject === 'toan' ? 'Toán học' : currentSubject === 'ly' ? 'Vật lí' : 'Hóa học'} khối ${currentGrade}. Định dạng kết quả là một mảng JSON, mỗi phần tử là một đối tượng có thuộc tính "question" và "solution".`;

            try {

                const problems = await geminiCall(prompt, true);

                if (Array.isArray(problems) && problems.length > 0) {

                    const problemsHtml = problems.map((p, index) => `

                        <div class="mb-6 border-b border-slate-200 pb-4 last:border-b-0 last:pb-0">

                            <h4 class="font-semibold text-slate-800">Câu hỏi ${index + 1}:</h4>

                            <p class="mt-1 mb-2 text-slate-700 whitespace-pre-wrap">${p.question}</p>

                            <details class="group">

                                <summary class="cursor-pointer font-semibold text-indigo-600 hover:text-indigo-800 transition-colors">

                                    <span class="group-open:hidden">Xem đáp án</span>

                                    <span class="group-open:inline-block">Ẩn đáp án</span>

                                </summary>

                                <div class="mt-2 p-3 bg-slate-50 rounded-lg whitespace-pre-wrap">

                                    ${p.solution}

                                </div>

                            </details>

                        </div>

                    `).join('');

                    generatedContentBody.innerHTML = problemsHtml;

                } else {

                    generatedContentBody.innerHTML = `<p>Xin lỗi, không thể tạo bài tập vào lúc này. Vui lòng thử lại sau.</p>`;

                }

            } catch (error) {

                generatedContentBody.innerHTML = `<p class="text-red-500">Đã xảy ra lỗi khi tạo bài tập. Vui lòng thử lại.</p>`;

            } finally {

                loadingIndicator.classList.add('hidden');

            }

        }

        function renderContent() {

            contentContainer.innerHTML = '';

            const data = curriculumData[currentSubject]?.[currentGrade];

            if (!data) {

                contentContainer.innerHTML = `<p class="text-center text-slate-500">Chưa có dữ liệu cho mục này.</p>`;

                return;

            }

            const panel = document.createElement('div');

            panel.className = 'content-panel active grid grid-cols-1 md:grid-cols-2 lg:grid-cols-3 gap-6';

            Object.entries(data).forEach(([category, topics]) => {

                const isSpecialTopic = category === "Chuyên đề";

                const card = document.createElement('div');

                card.className = `rounded-lg shadow p-6 flex flex-col ${isSpecialTopic ? 'bg-indigo-50 border-2 border-indigo-400' : 'bg-white'}`;

                const topicList = topics.map((topic, index) => {

                    const uniqueId = `${currentSubject}-${currentGrade}-${category}-${index}`;

                    return `

                        <li class="task-item py-2 flex items-center justify-between group">

                            <div class="flex items-center">

                                <input type="checkbox" id="${uniqueId}" class="h-4 w-4 rounded border-gray-300 text-indigo-600 focus:ring-indigo-500 mr-3">

                                <label for="${uniqueId}" class="text-slate-700">${topic}</label>

                            </div>

                            <div class="flex space-x-2 opacity-0 group-hover:opacity-100 transition-opacity">

                                <button data-action="summary" data-topic="${topic}" class="text-sm px-3 py-1 bg-blue-100 text-blue-800 rounded-full font-semibold hover:bg-blue-200 transition-colors">✨ Tóm tắt</button>

                                <button data-action="practice" data-topic="${topic}" class="text-sm px-3 py-1 bg-indigo-100 text-indigo-800 rounded-full font-semibold hover:bg-indigo-200 transition-colors">✨ Bài tập</button>

                            </div>

                        </li>

                    `;

                }).join('');

                card.innerHTML = `

                    <h3 class="text-xl font-bold text-indigo-700 mb-4">${category}</h3>

                    <ul class="space-y-1 flex-grow">${topicList}</ul>

                `;

                panel.appendChild(card);

            });

            contentContainer.appendChild(panel);

        }

        function updateActiveButtons() {

            document.querySelectorAll('.subject-btn').forEach(btn => {

                btn.classList.toggle('active', btn.dataset.subject === currentSubject);

            });

            document.querySelectorAll('.grade-btn').forEach(btn => {

                btn.classList.toggle('active', btn.dataset.grade === currentGrade);

            });

        }

        subjectSelector.addEventListener('click', (e) => {

            const subjectBtn = e.target.closest('.subject-btn');

            if (subjectBtn) {

                currentSubject = subjectBtn.dataset.subject;

                updateActiveButtons();

                renderContent();

            }

        });

        gradeSelector.addEventListener('click', (e) => {

            const gradeBtn = e.target.closest('.grade-btn');

            if (gradeBtn) {

                currentGrade = gradeBtn.dataset.grade;

                updateActiveButtons();

                renderContent();

            }

        });

        contentContainer.addEventListener('click', (e) => {

            const actionBtn = e.target.closest('[data-action]');

            if (actionBtn) {

                const action = actionBtn.dataset.action;

                const topic = actionBtn.dataset.topic;

                if (action === 'summary') {

                    getSummary(topic);

                } else if (action === 'practice') {

                    getPracticeProblems(topic);

                }

            }

        });

        closeButton.addEventListener('click', () => {

            generatedContentContainer.classList.add('hidden');

        });

        window.addEventListener('DOMContentLoaded', () => {

            updateActiveButtons();

            renderContent();

        });

    </script>

</body>

</html>