<!doctype html>

<html lang="vi">

<head>

<meta charset="utf-8" />

<meta name="viewport" content="width=device-width,initial-scale=1" />

<title>Flashcards Tiếng Anh - Tiếng Việt</title>

<style>

:root{

--bg:#f6f8fb; --card:#ffffff; --accent:#4f46e5; --muted:#6b7280; --danger:#ef4444;

--glass: rgba(255,255,255,0.65);

}

\*{box-sizing:border-box}

body{font-family:Inter,system-ui,Segoe UI,Roboto,Arial; margin:0; background:linear-gradient(180deg,var(--bg),#eef2ff); color:#111827}

.app{max-width:980px;margin:28px auto;padding:20px}

header{display:flex;align-items:center;gap:16px}

h1{margin:0;font-size:20px}

.top{display:flex;gap:12px;align-items:center}

.container{display:grid;grid-template-columns:380px 1fr;gap:18px;margin-top:18px}

.panel{background:var(--card);border-radius:12px;box-shadow:0 6px 18px rgba(16,24,40,0.06);padding:16px}

/\* form \*/

label{display:block;font-size:13px;color:var(--muted);margin-top:8px}

input[type=text], textarea, select{width:100%;padding:10px;border:1px solid #e6e9f2;border-radius:8px;margin-top:6px;font-size:14px}

textarea{min-height:76px;resize:vertical}

.row{display:flex;gap:8px}

.btn{display:inline-flex;align-items:center;gap:8px;padding:8px 12px;border-radius:10px;border:0;cursor:pointer;font-weight:600}

.btn.primary{background:var(--accent);color:white}

.btn.ghost{background:transparent;border:1px solid #e6e9f2}

.btn.warn{background:var(--danger);color:white}

/\* list \*/

.list{max-height:460px;overflow:auto;margin-top:12px}

.item{display:flex;justify-content:space-between;align-items:flex-start;padding:10px;border-radius:8px;border:1px solid #f1f5f9;margin-bottom:8px}

.item .meta{max-width:220px}

.meta b{display:block}

.meta small{color:var(--muted)}

.controls{display:flex;gap:6px}

.muted{color:var(--muted);font-size:13px}

/\* card \*/

.card-stage{display:flex;flex-direction:column;align-items:center;justify-content:center;min-height:360px}

.flashcard{width:100%;max-width:560px;background:linear-gradient(180deg,#fff, #fbfdff);border-radius:14px;padding:28px;border:1px solid rgba(15,23,42,0.04);box-shadow:0 10px 30px rgba(2,6,23,0.05);text-align:left}

.word{font-size:28px;font-weight:700}

.pron{color:var(--muted);margin-top:6px}

.meaning{margin-top:12px;font-size:16px}

.example{margin-top:12px;padding:12px;border-radius:8px;background:var(--glass);font-size:14px}

.controls-bottom{display:flex;gap:8px;align-items:center;justify-content:center;margin-top:14px}

.small{font-size:13px;padding:6px 8px}

.actions{display:flex;gap:10px;align-items:center}

footer{margin-top:18px;text-align:center;color:var(--muted);font-size:13px}

@media (max-width:880px){

.container{grid-template-columns:1fr;}

.card-stage{order: -1}

}

</style>

</head>

<body>

<div class="app">

<header>

<div class="top">

<svg width="36" height="36" viewBox="0 0 24 24" fill="none" aria-hidden>

<rect width="24" height="24" rx="8" fill="#eef2ff"></rect>

<path d="M7 12h10M7 8h10M7 16h6" stroke="#4f46e5" stroke-width="1.5" stroke-linecap="round" stroke-linejoin="round"/>

</svg>

</div>

<div>

<h1>Flashcards Tiếng Anh — Tiếng Việt</h1>

<div class="muted">Thêm / chỉnh sửa / xoá từ. Lưu trên trình duyệt (Local Storage).</div>

</div>

</header>

<div class="container">

<!-- left panel: add & list -->

<div class="panel">

<h3>Thêm từ mới</h3>

<div id="form">

<label for="word">Từ (English)</label>

<input id="word" type="text" placeholder="e.g. abandon" />

<label for="pron">Phiên âm (tùy chọn)</label>

<input id="pron" type="text" placeholder="e.g. /əˈbændən/" />

<label for="meaning">Nghĩa (Tiếng Việt)</label>

<input id="meaning" type="text" placeholder="e.g. từ bỏ, bỏ rơi" />

<label for="example">Ví dụ (English) — sẽ hiển thị kèm dịch</label>

<textarea id="example" placeholder="e.g. She abandoned her attempt to climb the mountain."></textarea>

<div style="display:flex;gap:8px;margin-top:10px">

<button id="addBtn" class="btn primary">Thêm / Lưu</button>

<button id="clearBtn" class="btn ghost">Xoá form</button>

<button id="importBtn" class="btn ghost">Nhập JSON</button>

<button id="exportBtn" class="btn ghost">Xuất JSON</button>

</div>

</div>

<hr style="margin:12px 0;border:none;border-top:1px solid #f1f5f9">

<div style="display:flex;justify-content:space-between;align-items:center">

<div class="muted">Danh sách từ</div>

<div style="display:flex;gap:8px;align-items:center">

<label class="muted">Sắp xếp</label>

<select id="orderSel">

<option value="default">Thêm gần nhất</option>

<option value="alpha">A → Z</option>

<option value="alpha-desc">Z → A</option>

</select>

</div>

</div>

<div class="list" id="list"></div>

</div>

<!-- right panel: flashcard stage -->

<div class="panel card-stage">

<div class="flashcard" id="flashcard">

<div class="word" id="cardWord">—</div>

<div class="pron" id="cardPron">&nbsp;</div>

<div class="meaning" id="cardMeaning">Chưa có từ nào trong danh sách — hãy thêm từ ở bên trái.</div>

<div class="example" id="cardExample" style="display:none"></div>

</div>

<div class="controls-bottom">

<div class="actions">

<button id="prevBtn" class="btn small">← Trước</button>

<button id="nextBtn" class="btn small">Tiếp →</button>

<button id="shuffleBtn" class="btn small">Ngẫu nhiên</button>

</div>

<div style="width:20px"></div>

<div class="actions">

<button id="playBtn" class="btn small">🔊 Nghe phát âm</button>

<select id="voiceSel" style="padding:6px;border-radius:8px;border:1px solid #e6e9f2">

<option value="default">Giọng mặc định</option>

</select>

</div>

</div>

<div style="margin-top:12px;display:flex;justify-content:center;gap:8px;flex-wrap:wrap">

<button id="editModeBtn" class="btn ghost small">Chỉnh sửa từ hiện tại</button>

<button id="deleteCardBtn" class="btn warn small">Xoá từ hiện tại</button>

<button id="clearAllBtn" class="btn ghost small">Xoá toàn bộ</button>

</div>

</div>

</div>

<footer>Chạy trên Chrome / Edge. Dữ liệu lưu trong Local Storage của trình duyệt.</footer>

</div>

<script>

// KEY for localStorage

const STORAGE\_KEY = 'pv\_flashcards\_v1';

// state

let cards = []; // array of {id, word, pron, meaning, example, created}

let idx = 0; // current index in orderList

let orderList = []; // indexes into cards

// DOM

const wordEl = document.getElementById('word');

const pronEl = document.getElementById('pron');

const meaningEl = document.getElementById('meaning');

const exampleEl = document.getElementById('example');

const addBtn = document.getElementById('addBtn');

const clearBtn = document.getElementById('clearBtn');

const listEl = document.getElementById('list');

const cardWord = document.getElementById('cardWord');

const cardPron = document.getElementById('cardPron');

const cardMeaning = document.getElementById('cardMeaning');

const cardExample = document.getElementById('cardExample');

const prevBtn = document.getElementById('prevBtn');

const nextBtn = document.getElementById('nextBtn');

const shuffleBtn = document.getElementById('shuffleBtn');

const playBtn = document.getElementById('playBtn');

const voiceSel = document.getElementById('voiceSel');

const editModeBtn = document.getElementById('editModeBtn');

const deleteCardBtn = document.getElementById('deleteCardBtn');

const clearAllBtn = document.getElementById('clearAllBtn');

const orderSel = document.getElementById('orderSel');

const importBtn = document.getElementById('importBtn');

const exportBtn = document.getElementById('exportBtn');

// util

const uid = ()=> Date.now().toString(36) + Math.random().toString(36).slice(2,8);

// load voices

function loadVoices(){

const voices = speechSynthesis.getVoices();

// clear except default

voiceSel.innerHTML = '<option value="default">Giọng mặc định</option>';

voices.forEach((v,i)=>{

const opt = document.createElement('option');

opt.value = i;

opt.textContent = `${v.name} — ${v.lang}`;

voiceSel.appendChild(opt);

});

}

window.speechSynthesis.onvoiceschanged = loadVoices;

setTimeout(loadVoices, 250);

// storage

function save(){

localStorage.setItem(STORAGE\_KEY, JSON.stringify(cards));

}

function load(){

try{

const raw = localStorage.getItem(STORAGE\_KEY);

if(raw) cards = JSON.parse(raw) || [];

}catch(e){cards=[]}

}

// render list

function renderList(){

listEl.innerHTML = '';

if(cards.length===0){ listEl.innerHTML = '<div class="muted">Chưa có từ nào</div>'; return; }

const sorted = cards.slice().sort((a,b)=> b.created - a.created);

sorted.forEach(c=>{

const div = document.createElement('div'); div.className='item';

const meta = document.createElement('div'); meta.className='meta';

const b = document.createElement('b'); b.textContent = c.word;

const s = document.createElement('small'); s.textContent = c.meaning;

meta.appendChild(b); meta.appendChild(s);

const actions = document.createElement('div'); actions.className='controls';

const btnPlay = document.createElement('button'); btnPlay.className='btn small'; btnPlay.textContent='🔊'; btnPlay.title='Nghe từ này';

btnPlay.onclick = ()=> speak(c.word);

const btnEdit = document.createElement('button'); btnEdit.className='btn small'; btnEdit.textContent='✏️'; btnEdit.title='Chỉnh sửa';

btnEdit.onclick = ()=> loadToForm(c.id);

const btnDelete = document.createElement('button'); btnDelete.className='btn small'; btnDelete.textContent='🗑️'; btnDelete.title='Xoá';

btnDelete.onclick = ()=>{ if(confirm('Xoá từ này?')){ removeCard(c.id); }};

actions.appendChild(btnPlay); actions.appendChild(btnEdit); actions.appendChild(btnDelete);

div.appendChild(meta); div.appendChild(actions);

listEl.appendChild(div);

});

}

function buildOrder(){

const mode = orderSel.value;

if(mode==='default'){

// newest first

orderList = cards.slice().sort((a,b)=> b.created - a.created).map(c=>c.id);

}else if(mode==='alpha'){

orderList = cards.slice().sort((a,b)=> a.word.localeCompare(b.word)).map(c=>c.id);

}else if(mode==='alpha-desc'){

orderList = cards.slice().sort((a,b)=> b.word.localeCompare(a.word)).map(c=>c.id);

}

if(orderList.length===0) idx = 0; else idx = 0;

}

function showCardByIdx(i){

if(cards.length===0){

cardWord.textContent='—'; cardPron.textContent=''; cardMeaning.textContent='Chưa có từ nào trong danh sách — hãy thêm từ ở bên trái.'; cardExample.style.display='none';

return;

}

if(i<0) i = 0; if(i>=orderList.length) i = orderList.length-1;

idx = i;

const id = orderList[idx];

const c = cards.find(x=>x.id===id);

if(!c) return;

cardWord.textContent = c.word;

cardPron.textContent = c.pron || '';

cardMeaning.textContent = c.meaning || '';

if(c.example){ cardExample.style.display='block'; cardExample.textContent = 'Ví dụ: ' + c.example; } else { cardExample.style.display='none'; }

}

function nextCard(){ if(orderList.length===0) return; idx = (idx+1) % orderList.length; showCardByIdx(idx); }

function prevCard(){ if(orderList.length===0) return; idx = (idx-1 + orderList.length) % orderList.length; showCardByIdx(idx); }

function shuffleOrder(){

// Fisher-Yates shuffle

for(let i=orderList.length-1;i>0;i--){ const j=Math.floor(Math.random()\*(i+1)); [orderList[i],orderList[j]]=[orderList[j],orderList[i]] }

idx = 0; showCardByIdx(idx);

}

// CRUD

function addOrUpdateFromForm(){

const w = wordEl.value.trim(); if(!w){ alert('Nhập từ tiếng Anh.'); wordEl.focus(); return; }

const pr = pronEl.value.trim(); const mn = meaningEl.value.trim(); const ex = exampleEl.value.trim();

const editingId = addBtn.dataset.editId;

if(editingId){

const card = cards.find(c=>c.id===editingId);

if(card){ card.word = w; card.pron = pr; card.meaning = mn; card.example = ex; }

delete addBtn.dataset.editId; addBtn.textContent='Thêm / Lưu';

} else {

cards.push({id:uid(), word:w, pron:pr, meaning:mn, example:ex, created:Date.now()});

}

save(); renderList(); buildOrder(); showCardByIdx(idx);

clearForm();

}

function loadToForm(id){

const c = cards.find(x=>x.id===id); if(!c) return;

wordEl.value = c.word; pronEl.value = c.pron; meaningEl.value = c.meaning; exampleEl.value = c.example;

addBtn.dataset.editId = c.id; addBtn.textContent='Cập nhật'; wordEl.focus();

}

function removeCard(id){

cards = cards.filter(c=>c.id!==id); save(); renderList(); buildOrder(); showCardByIdx(idx);

}

function clearAll(){ if(!confirm('Xoá toàn bộ flashcards?')) return; cards=[]; save(); renderList(); buildOrder(); showCardByIdx(0); }

function clearForm(){ wordEl.value=''; pronEl.value=''; meaningEl.value=''; exampleEl.value=''; delete addBtn.dataset.editId; addBtn.textContent='Thêm / Lưu'; }

// speak using Web Speech API

function speak(text){

if(!('speechSynthesis' in window)){ alert('Trình duyệt không hỗ trợ Web Speech API.'); return; }

const utter = new SpeechSynthesisUtterance(text);

const sel = voiceSel.value;

if(sel !== 'default'){

const voices = speechSynthesis.getVoices();

const v = voices[parseInt(sel)]; if(v) utter.voice = v;

}

utter.lang = 'en-US'; // guide for English pronunciation — user can change voice if available

speechSynthesis.cancel(); speechSynthesis.speak(utter);

}

// export / import JSON

function exportJSON(){

const blob = new Blob([JSON.stringify(cards, null, 2)], {type:'application/json'});

const url = URL.createObjectURL(blob);

const a = document.createElement('a'); a.href = url; a.download = 'flashcards.json'; a.click(); URL.revokeObjectURL(url);

}

function importJSON(){

const inp = document.createElement('input'); inp.type='file'; inp.accept='application/json';

inp.onchange = e=>{

const f = e.target.files[0]; if(!f) return;

const reader = new FileReader();

reader.onload = ev=>{

try{ const arr = JSON.parse(ev.target.result);

if(Array.isArray(arr)){

// simple merge: keep existing then new ones

const mapped = arr.map(x=> ({id: uid(), word: x.word||x.w||x.text||'', pron: x.pron||'', meaning: x.meaning||'', example: x.example||'', created: Date.now()})).filter(x=>x.word);

cards = cards.concat(mapped);

save(); renderList(); buildOrder(); showCardByIdx(idx);

alert('Nhập thành công ' + mapped.length + ' từ.');

} else alert('File không đúng định dạng.');

}catch(err){ alert('Lỗi khi đọc file.'); }

};

reader.readAsText(f);

};

inp.click();

}

// events

addBtn.addEventListener('click', addOrUpdateFromForm);

clearBtn.addEventListener('click', clearForm);

nextBtn.addEventListener('click', nextCard);

prevBtn.addEventListener('click', prevCard);

shuffleBtn.addEventListener('click', ()=>{ shuffleOrder(); });

playBtn.addEventListener('click', ()=>{ speak(cardWord.textContent); });

deleteCardBtn.addEventListener('click', ()=>{ const id = orderList[idx]; if(!id){ alert('Không có từ để xoá.'); return; } if(confirm('Xoá từ đang hiển thị?')){ removeCard(id);} });

editModeBtn.addEventListener('click', ()=>{ const id = orderList[idx]; if(!id){ alert('Không có từ để chỉnh sửa.'); return; } loadToForm(id); });

clearAllBtn.addEventListener('click', clearAll);

orderSel.addEventListener('change', ()=>{ buildOrder(); showCardByIdx(0); });

importBtn.addEventListener('click', importJSON);

exportBtn.addEventListener('click', exportJSON);

// keyboard: left/right

document.addEventListener('keydown', (e)=>{ if(e.key==='ArrowRight') nextCard(); if(e.key==='ArrowLeft') prevCard(); });

// initialize

load(); renderList(); buildOrder(); showCardByIdx(0);

// small helpful guidance for pronouncing: if user didn't enter pron, we still use Web Speech API to read word

// Compatible with Chrome & Edge which support speechSynthesis

</script>

</body>

</html>