<!DOCTYPE html>

<html lang="zh-CN">

<head>

<meta charset="utf-8" />

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

<title>能力 × 模式 · 玻璃瓶矩阵 · 拟物书架</title>

<style>

:root{

  --sidebar-w: 300px;

  --bg: #0f172a; --panel: #0b1020; --txt: #e5e7eb; --muted: #9aa4b2; --hover: #172033;

  --accent: #60a5fa; --accent-2: #34d399; --radius: 12px; --shadow: 0 8px 30px rgba(0,0,0,.35);

  --duration: 220ms; --bezier: cubic-bezier(.2,.7,.2,1);

  --bg-grad: radial-gradient(1200px 800px at 20% -10%, #111b33 0%, #0b1020 40%, #090f1b 100%);

  --steel-reflect: linear-gradient(120deg, rgba(255,255,255,.65) 0%, rgba(255,255,255,0) 30%, rgba(255,255,255,.25) 65%, rgba(255,255,255,0) 100%);

  --glass: linear-gradient(180deg, rgba(255,255,255,.22), rgba(255,255,255,.06));

}

\*{ box-sizing:border-box; }

html,body{ height:100%; }

body{

  margin:0; font-family: ui-sans-serif, system-ui, -apple-system, "Segoe UI", Roboto, "PingFang SC","Hiragino Sans GB","Microsoft YaHei", Arial, "Noto Sans", sans-serif;

  color:var(--txt); background: var(--bg-grad); display:grid; grid-template-columns: var(--sidebar-w) 1fr; min-height:100vh;

}

/\* ==== 侧栏（保留原风格与动画） ==== \*/

aside{ position:sticky; top:0; height:100svh; background: linear-gradient(160deg, var(--bg), #0d152a 60%, #0a1326 100%); border-right:1px solid rgba(255,255,255,.06); padding:18px 14px; }

.brand{ display:flex; align-items:center; gap:12px; padding:12px; border-radius:14px; text-decoration:none; color:var(--txt);

  box-shadow: inset 0 0 0 1px rgba(255,255,255,.06);

  background: linear-gradient(140deg, rgba(96,165,250,.25), transparent 40%) border-box,

             linear-gradient(140deg, rgba(52,211,153,.25), transparent 60%) border-box,

             rgba(255,255,255,.02); margin-bottom: 16px; }

.brand i{ display:grid; place-items:center; width:38px; height:38px; border-radius:10px; background: conic-gradient(from 280deg, #60a5fa, #34d399, #a78bfa, #60a5fa); color:#0a1020; font-weight:900; font-style:normal; box-shadow: var(--shadow);}

.brand small{ color:var(--muted); font-size:12px; margin-left:auto }

.nav{ list-style:none; margin:0; padding:4px; }

.section{ margin:8px 0 10px; border-radius: var(--radius); overflow: clip; }

.trigger{ width:100%; display:flex; align-items:center; gap:10px; padding:12px; color:var(--txt); background:transparent; border:0; border-radius: var(--radius); cursor:pointer; text-align:left; font:inherit; }

.trigger:hover{ background: var(--hover); }

.icon{ width:24px; display:grid; place-items:center; opacity:.92 }

.label{ flex:1 1 auto; }

.chev{ width: 16px; height:16px; transform: rotate(-90deg); opacity:.85; transition: transform var(--duration) var(--bezier); }

.section[data-open="true"] .chev{ transform: rotate(0deg); }

.submenu-wrap{ height:0; overflow:hidden; transition: height var(--duration) var(--bezier); }

.submenu{ list-style:none; margin:0; padding: 6px 8px 12px 44px; background: linear-gradient(180deg, rgba(255,255,255,.02), rgba(255,255,255,.01)); border-radius: 0 0 var(--radius) var(--radius); }

.submenu a{ display:block; padding:8px 10px; color:var(--muted); text-decoration:none; border-radius:8px; opacity:0; transform: translateY(-2px); transition: .22s var(--bezier); }

.section[data-open="true"] .submenu a{ opacity:1; transform: translateY(0); }

.submenu a:hover{ background:#152036; color:var(--txt); }

.active{ color: var(--accent) !important; }

/\* ==== 主区 ==== \*/

main{ padding: 26px 30px 80px; }

.page-title{ font-size: 30px; margin: 6px 0 10px; font-weight: 800; }

.lead{ color:var(--muted); line-height:1.7; max-width:70ch; }

/\* ==== 轴样式（胶囊玻璃） ==== \*/

.axis{

  display:grid;

  grid-template-columns: 160px repeat(5, minmax(140px, 1fr));

  gap: 14px;

  align-items:center;

  margin: 14px 0 8px;

}

.axis .pill{

  display:flex; align-items:center; justify-content:center; gap:10px;

  padding:10px 14px; border-radius:999px;

  background: var(--glass); border:1px solid rgba(255,255,255,.18);

  box-shadow: 0 8px 24px rgba(0,0,0,.28), inset 0 1px 0 rgba(255,255,255,.35);

  backdrop-filter: blur(5px);

  text-align:center; font-weight:800;

}

.axis .pill .dot{ width:8px; height:8px; border-radius:50%; background: radial-gradient(circle at 30% 30%, #9ec5ff, #5ea8ff 70%, #2b5fb8); box-shadow: 0 0 10px rgba(96,165,250,.6); }

/\* ==== 玻璃瓶矩阵 ==== \*/

.grid{

  display:grid;

  grid-template-columns: 160px repeat(5, minmax(140px,1fr));

  gap:14px; align-items:stretch;

}

.row-head{

  display:flex; align-items:center; gap:10px; padding:12px 14px;

  border-radius:14px; background: var(--glass);

  border:1px solid rgba(255,255,255,.18);

  box-shadow: 0 8px 24px rgba(0,0,0,.28), inset 0 1px 0 rgba(255,255,255,.35);

  backdrop-filter: blur(5px); font-weight:800;

}

.cell{

  position:relative; border-radius:16px; padding:12px;

  background: linear-gradient(180deg, rgba(255,255,255,.07), rgba(255,255,255,.02));

  border:1px solid rgba(255,255,255,.14); box-shadow: var(--shadow);

}

.cell-inner{

  height: 120px; display:grid; place-items:center; position:relative;

}

.cell .badge{

  position:absolute; top:10px; right:10px; font-size:12px; padding:3px 8px; border-radius:999px;

  color:#0a1020; background: var(--steel-reflect), linear-gradient(180deg, #e6eef7, #c8d4e3 55%, #b0bdcd);

  border:1px solid rgba(40,60,80,.45);

}

.cell .label{

  margin-top:10px; text-align:center; color:var(--muted); font-size:12px;

}

/\* 瓶子（SVG 容器自适应）, 使用 CSS 变量 --fill 控制填充 0~1 \*/

.bottle{ width:80px; height:120px; position:relative; cursor:pointer; }

.bottle svg{ width:100%; height:100%; display:block; filter: drop-shadow(0 6px 14px rgba(0,0,0,.45)); }

.bottle .liquid{

  transition: transform var(--duration) var(--bezier);

  transform: translateY(calc((1 - var(--fill, 0)) \* 120px)); /\* 120px 是瓶身内容区域的估算高度 \*/

}

.bottle:hover .liquid{ transform: translateY(calc((1 - var(--fill, 0)) \* 120px - 4px)); }

.bottle .shine{ opacity:.8; }

.bottle .bubble{ opacity:.35; animation: up 4s linear infinite; }

@keyframes up{

  0%{ transform: translateY(0); opacity:.0;}

  15%{ opacity:.35;}

  100%{ transform: translateY(-60px); opacity:0; }

}

.hidden{ display:none !important; }

/\* ==== 书架（保持原拟物风格；略） ==== \*/

.wall{ max-width:1240px; margin: 18px auto 0; padding-right: 6px; }

.shelf{ position:relative; padding-bottom: 82px; margin-bottom:10px; }

.shelf::after{

  content:""; position:absolute; left:-18px; right:-18px; bottom:0; height: 24px; border-radius:10px;

  background:

    linear-gradient(180deg, rgba(255,255,255,.12) 0%, rgba(255,255,255,0) 40%),

    repeating-linear-gradient(90deg, rgba(110,78,57,.14) 0 6px, rgba(110,78,57,.05) 6px 12px),

    linear-gradient(180deg, #3b2b20, #2c211a 55%, #1b1310 100%);

  box-shadow: 0 24px 40px rgba(0,0,0,.55), inset 0 3px 0 rgba(255,255,255,.08), inset 0 -10px 20px rgba(0,0,0,.45);

}

.shelf-label{

  position:absolute; left:50%; transform: translateX(-50%); bottom: 36px;

  display:inline-flex; align-items:center; gap:10px; padding:8px 14px; border-radius:999px;

  color:#0a1020; font-weight:800; letter-spacing:.6px; text-transform: uppercase;

  background: var(--steel-reflect), linear-gradient(180deg, #e6eef7, #c8d4e3 55%, #b0bdcd);

  box-shadow: 0 8px 18px rgba(0,0,0,.45), inset 0 1px 0 rgba(255,255,255,.9), inset 0 -2px 0 rgba(0,0,0,.15);

  border:1px solid rgba(40,60,80,.45);

}

.shelf-grid{ display:grid; grid-template-columns: repeat(auto-fit, minmax(220px, 1fr)); gap:18px; align-items:end; }

.card-s{ position:relative; border-radius:14px; padding:10px 10px 12px; background: linear-gradient(180deg, rgba(255,255,255,.06), rgba(255,255,255,.02));

         border:1px solid rgba(255,255,255,.08); box-shadow: var(--shadow); }

.cover-s{ width:100%; aspect-ratio: 3/3; border-radius:12px; background:#0f172a; box-shadow: inset 0 0 0 1px rgba(255,255,255,.06); }

.name-s{ display:flex; align-items:center; gap:8px; margin:10px 6px 0; font-weight:800; }

.name-s .dot{ width:7px; height:7px; border-radius:50%; background: radial-gradient(circle at 30% 30%, var(--accent), #3a79ff 70%, #2750a8); box-shadow: 0 0 8px rgba(96,165,250,.6); }

.desc-s{ margin: 4px 6px 0; color: var(--muted); font-size:12px; }

.actions-s{ display:flex; gap:10px; padding:10px 6px 0; }

.btn{ appearance:none; border:1px solid rgba(255,255,255,.12); background: rgba(255,255,255,.03); color:var(--txt); font-size:12px; padding:6px 10px; border-radius:999px; cursor:pointer; text-decoration:none; display:inline-flex; align-items:center; gap:6px; transition: background var(--duration) var(--bezier), border-color var(--duration) var(--bezier), transform 120ms ease; }

.btn:hover{ background: rgba(255,255,255,.07); border-color: rgba(255,255,255,.18); }

.btn.primary{ border-color: rgba(96,165,250,.45); background: rgba(96,165,250,.14); }

/\* ==== 响应式 ==== \*/

@media (max-width: 1200px){

  body{ grid-template-columns:1fr; }

  aside{ position: static; height:auto; border-right:none; border-bottom:1px solid rgba(255,255,255,.06); }

  .axis, .grid{ grid-template-columns: 180px repeat(5, minmax(120px,1fr)); }

}

</style>

</head>

<body>

  <!-- 侧栏：原风格与动画 -->

  <aside>

    <a class="brand" href="#dashboard" aria-label="返回仪表盘">

      <i>Λ</i>

      <div>

        <strong>我的网站</strong><br>

        <span style="font-size:12px; color:var(--muted)">能力 × 模式 · 玻璃瓶矩阵</span>

      </div>

      <small>v4.0</small>

    </a>

    <nav aria-label="侧边导航"><ul class="nav" id="nav-root"></ul></nav>

  </aside>

  <main>

    <section id="view-dashboard" class="route">

      <h1 class="page-title">玻璃瓶矩阵</h1>

      <p class="lead">每个格子是一只玻璃瓶，液面高度 = (x,y) 的值。点击可进入对应培养方案。</p>

      <!-- 顶部列轴 -->

      <div class="axis" id="axis-col"></div>

      <!-- 矩阵主体 -->

      <div class="grid" id="matrix"></div>

    </section>

    <section id="view-dynamic" class="route hidden" aria-live="polite"></section>

  </main>

<script>

  // ✅ 路由规则（你只改这一块）

const ROUTE = {

  // 默认：仍走单页 hash 路由（#mode/<模式>/ability/<能力>）

  modeAbilityTo: (mode, ability) =>

    `#mode/${encodeURIComponent(mode)}/ability/${encodeURIComponent(ability)}`,

  // 可选：为某些 (mode|ability) 单独指定目标（支持外链或本地子页）

  // 键名格式："<模式>|<能力>"

  overrides: {

    // 示例：

    "数据说|数据获取与规整能力": "subpages/data-acq.html",

    "模型说|模型构建与量化分析能力": "https://example.com/guide"

  },

  // 可选：整列或整行的统一跳转（比 overrides 优先级低）

  byMode: {

    "数据说":      "数据说.html",

    "模型说":      "模型说.html",

    "深度报告":    "深度报告.html",

    "比赛/项目":  "比赛项目.html",

    "数据科学项目":"数据科学项目.html"

  },

  byAbility: {

    // "模型解读与决策能力": "subpages/insight.html",

  },

  // 外链打开方式：'\_self'（当前页）或 '\_blank'（新开标签页）

  target: "\_self"

};

(function(){

  /\* =======================

   \*  唯一需要你改的地方：CONFIG

   \*  - abilities: 纵轴标签（x）

   \*  - modes: 横轴标签（y）

   \*  - values: 0~1 的二维数组（行=ability，列=mode）

   \*  修改这里，UI与路由会自动更新。

   \* ======================= \*/

  const CONFIG = {

    abilities: [

      "数据获取与<br>规整能力",

      "模型构建与<br>量化分析能力",

      "模型解读与<br>决策能力",

      "学术与商业场景<br>迁移能力",

      "决策思维与<br>战略视野能力"

    ],

    modes: ["数据说","模型说","深度报告","比赛/项目","数据科学项目"],

    // 示例数值：5×5，0~1 任意小数。可直接改。

    values: [

      [0.8, 1.0, 1.0, 0.7, 0.9],

      [0.8, 1.0, 0.9, 0.7, 0.6],

      [0.1, 1.0, 1.0, 0.5, 0.8],

      [0.1, 0.3, 0.9, 0.9, 1.0],

      [0.0, 0.0, 0.3, 0.9, 0.9]

    ],

    // 阶段用于书架视图（可改名/增删，非必须）

    stagesByMode: {

      "数据说":["数据处理","建模分析","决策应用"],

      "模型说":["数据处理","建模分析","决策应用"],

      "深度报告":["数据处理","建模分析","决策应用"],

      "比赛/项目":["数据处理","建模分析","决策应用"],

      "数据科学项目":["数据处理","建模分析","决策应用"]

    }

  };

  // ====== 侧栏（模式→能力）确保正确显示 ======

  const navRoot = document.getElementById('nav-root');

  (function buildSidebar(){

    navRoot.innerHTML = '';

    // 仪表盘

    const dash = document.createElement('li'); dash.className='section'; dash.dataset.id='dashboard'; dash.setAttribute('data-open','true');

    dash.innerHTML = `

      <button class="trigger" type="button" aria-expanded="true" aria-controls="sm-dashboard">

        <span class="icon">📊</span><span class="label">仪表盘</span>

        <svg class="chev" viewBox="0 0 24 24" aria-hidden="true"><path fill="currentColor" d="M8.12 9.29L12 13.17l3.88-3.88a1 1 0 0 1 1.41 1.42l-4.59 4.59a1 1 0 0 1-1.41 0L6.7 10.7A1 1 0 0 1 8.12 9.3z"/></svg>

      </button>

      <div class="submenu-wrap" id="sm-dashboard" role="region" aria-label="仪表盘 子菜单">

        <ul class="submenu"><li><a href="#dashboard" class="active">总览</a></li></ul>

      </div>`;

    navRoot.appendChild(dash);

    // 折叠动画 + 记忆

    const LS\_KEY = "sidebar-open-sections";

    const load = ()=>{ try{return new Set(JSON.parse(localStorage.getItem(LS\_KEY)||'[]'));}catch(e){return new Set();}};

    const save = set=>{ try{localStorage.setItem(LS\_KEY, JSON.stringify([...set]));}catch(e){} };

    const openSet = load();

    document.querySelectorAll(".section").forEach(section=>{

      const id = section.dataset.id;

      const btn = section.querySelector(".trigger");

      const wrap = section.querySelector(".submenu-wrap");

      const open = section.getAttribute('data-open')==="true" || openSet.has(id);

      setOpen(section,btn,wrap,open,{skip:true});

      btn.addEventListener('click', ()=>toggle(section,btn,wrap));

      btn.addEventListener('keydown', e=>{ if(e.key==="Enter"||e.key===" "){ e.preventDefault(); toggle(section,btn,wrap);} });

      wrap.addEventListener('transitionend', e=>{ if(e.propertyName==="height"&&section.getAttribute('data-open')==="true"){ wrap.style.height="auto"; }});

    });

    function toggle(section,btn,wrap){

      const willOpen = section.getAttribute('data-open')!=="true";

      setOpen(section,btn,wrap,willOpen);

      const id = section.dataset.id;

      if(willOpen) openSet.add(id); else openSet.delete(id);

      save(openSet);

    }

    function setOpen(section,btn,wrap,open,{skip}={}){

      if(skip){ section.setAttribute("data-open", open?"true":"false"); btn.setAttribute("aria-expanded", open?"true":"false"); wrap.style.height=open?"auto":"0px"; return; }

      const h = wrap.getBoundingClientRect().height;

      if(open){

        section.setAttribute("data-open","true"); btn.setAttribute("aria-expanded","true");

        wrap.style.height = h+"px"; void wrap.offsetHeight; wrap.style.height = wrap.scrollHeight+"px";

      }else{

        if(getComputedStyle(wrap).height==="auto"){ wrap.style.height = wrap.scrollHeight+"px"; void wrap.offsetHeight; }

        section.setAttribute("data-open","false"); btn.setAttribute("aria-expanded","false"); wrap.style.height="0px";

      }

    }

  })();

  // ====== 轴（顶部列名 & 左侧行名）======

  const axisCol = document.getElementById('axis-col');

  axisCol.innerHTML = `<div class="pill" style="justify-content:flex-start;padding-left:18px;">能力 \\ 模式</div>` +

    CONFIG.modes.map(m=>`<div class="pill"><span class="dot"></span>${m}</div>`).join('');

  // ====== 矩阵主体 ======

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

  function bottleSVG(fill){

    // fill 0..1 -> CSS var --fill 控制液面平移；用 mask/clip 保证瓶型

    const id = 'clip'+Math.random().toString(36).slice(2,8);

    return `

    <div class="bottle" style="--fill:${Math.max(0,Math.min(1,fill))}">

      <svg viewBox="0 0 120 200" aria-hidden="true">

        <defs>

          <clipPath id="${id}">

            <!-- 瓶身路径（圆角矩形 + 瓶颈） -->

            <path d="

              M45 10

              h30

              a8 8 0 0 1 8 8

              v14

              a10 10 0 0 1 -10 10

              h-26

              a10 10 0 0 1 -10 -10

              v-14

              a8 8 0 0 1 8 -8

              z

              M30 42

              h60

              v120

              a14 14 0 0 1 -14 14

              h-32

              a14 14 0 0 1 -14 -14

              z

            " />

          </clipPath>

        </defs>

        <!-- 背板玻璃 -->

        <rect x="26" y="38" width="68" height="142" rx="16" fill="url(#glassGrad)" opacity="0.55"></rect>

        <!-- 液体（被剪裁为瓶形） -->

        <g clip-path="url(#${id})">

          <g class="liquid">

            <rect x="28" y="60" width="64" height="120" fill="url(#liquidGrad)"/>

            <!-- 液面圆角 -->

            <ellipse cx="60" cy="60" rx="32" ry="6" fill="url(#liquidTop)" />

            <!-- 小气泡 -->

            <circle class="bubble" cx="45" cy="150" r="3" fill="#b2e8ff"/>

            <circle class="bubble" cx="70" cy="160" r="2.5" fill="#b2e8ff" style="animation-delay:.6s"/>

            <circle class="bubble" cx="55" cy="140" r="2" fill="#b2e8ff" style="animation-delay:1.2s"/>

          </g>

        </g>

        <!-- 高光与轮廓 -->

        <path d="M30 42 h60 v120 a14 14 0 0 1 -14 14 h-32 a14 14 0 0 1 -14 -14z"

          fill="none" stroke="rgba(255,255,255,.35)" stroke-width="2"/>

        <path class="shine" d="M36 48 v120 a10 10 0 0 0 10 10" fill="none" stroke="white" stroke-opacity=".25" stroke-width="6"/>

        <rect x="44" y="10" width="32" height="22" rx="8" fill="rgba(255,255,255,.25)" stroke="rgba(255,255,255,.35)" />

        <rect x="40" y="36" width="40" height="10" rx="5" fill="rgba(255,255,255,.15)" />

        <defs>

          <linearGradient id="glassGrad" x1="0" x2="0" y1="0" y2="1">

            <stop offset="0%" stop-color="white" stop-opacity=".18"/>

            <stop offset="100%" stop-color="white" stop-opacity=".05"/>

          </linearGradient>

          <linearGradient id="liquidGrad" x1="0" x2="0" y1="0" y2="1">

            <stop offset="0%" stop-color="#6fb7ff"/>

            <stop offset="100%" stop-color="#32d399"/>

          </linearGradient>

          <linearGradient id="liquidTop" x1="0" x2="0" y1="0" y2="1">

            <stop offset="0%" stop-color="#dff3ff" stop-opacity=".9"/>

            <stop offset="100%" stop-color="#8bddff" stop-opacity=".6"/>

          </linearGradient>

        </defs>

      </svg>

    </div>`;

  }

  // 生成左侧行头 + 玻璃瓶格子

(function buildGrid(){

  matrix.innerHTML = '';

  for (let r = 0; r < CONFIG.abilities.length; r++) {

    // 行头

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

    pill.className = 'row-head';

    pill.innerHTML = `<span class="dot" style="width:8px;height:8px;border-radius:50%;background:radial-gradient(circle at 30% 30%,#9ec5ff,#5ea8ff 70%,#2b5fb8);box-shadow:0 0 10px rgba(96,165,250,.6)"></span>${CONFIG.abilities[r]}`;

    matrix.appendChild(pill);

    // 列（瓶子）

    for (let c = 0; c < CONFIG.modes.length; c++) {

      const value = Number((CONFIG.values[r] || [])[c] || 0);

      const safe  = Math.max(0, Math.min(1, value));

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

      cell.className = 'cell';

      cell.innerHTML = `

        <span class="badge">${safe.toFixed(2)}</span>

        <div class="cell-inner">${bottleSVG(safe)}</div>

        <div class="label">${CONFIG.abilities[r]} × ${CONFIG.modes[c]}</div>

      `;

      // 跳转

      cell.tabIndex = 0;                       // 键盘可聚焦

      const mode = CONFIG.modes[c];

      const ability = CONFIG.abilities[r];

      function go() {

        let url = ROUTE.byMode[mode] || ROUTE.modeAbilityTo(mode, ability);

        if (String(url).startsWith('#')) {

          location.hash = url;                 // 单页 hash 路由

        } else {

          window.open(url, ROUTE.target);      // 外链或独立子页

        }

      }

      // 只保留这两行监听（不要重复再写）

      cell.addEventListener('click', go);

      cell.addEventListener('keydown', (e) => {

        if (e.key === 'Enter' || e.key === ' ') { e.preventDefault(); go(); }

      });

      matrix.appendChild(cell);

    }

  }

})();

  // ====== 简版路由到书架（如果你已有书架页，可替换为你的实现）======

  const stagesByMode = CONFIG.stagesByMode || {};

  function renderView(mode, ability){

    const viewDyn = document.getElementById('view-dynamic');

    const viewDash = document.getElementById('view-dashboard');

    viewDash.classList.add('hidden'); viewDyn.classList.remove('hidden');

    const stages = (stagesByMode[mode]||["数据处理","建模分析","决策应用"]);

    viewDyn.innerHTML = `<h1 class="page-title">${mode} ｜ ${ability}</h1>`;

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

    stages.forEach(stage=>{

      const shelf = document.createElement('section'); shelf.className='shelf';

      shelf.innerHTML = `

        <div class="shelf-label">

          <span style="width:8px;height:8px;border-radius:50%;background:radial-gradient(circle at 30% 30%,#b9c7d8,#7b8b9d 70%,#4b596a);box-shadow:inset 0 0 0 1px rgba(255,255,255,.4)"></span>

          ${stage}

          <span style="width:8px;height:8px;border-radius:50%;background:radial-gradient(circle at 30% 30%,#b9c7d8,#7b8b9d 70%,#4b596a);box-shadow:inset 0 0 0 1px rgba(255,255,255,.4)"></span>

        </div>

        <div class="shelf-grid"></div>`;

      const grid = shelf.querySelector('.shelf-grid');

      // 这里填你的资料卡；为示例先给占位

      const empty = document.createElement('article'); empty.className='card-s';

      empty.innerHTML = '<div class="cover-s"></div><h3 class="name-s"><span class="dot"></span> 敬请期待</h3><p class="desc-s">在 Excel 补充资料后，这里会自动呈现。</p><div class="actions-s"><a class="btn" href="#dashboard">返回</a></div>';

      grid.appendChild(empty);

      wall.appendChild(shelf);

    });

    viewDyn.appendChild(wall);

  }

  function applyRoute(){

    const hash = decodeURIComponent(location.hash || '#dashboard');

    const viewDyn = document.getElementById('view-dynamic');

    const viewDash = document.getElementById('view-dashboard');

    // 侧栏激活态

    document.querySelectorAll('.submenu a').forEach(a=>{

      if(a.getAttribute('href')===hash){ a.classList.add('active'); }

      else{ a.classList.remove('active'); }

    });

    if(hash==='#dashboard' || !hash.startsWith('#mode/')){

      viewDash.classList.remove('hidden'); viewDyn.classList.add('hidden'); return;

    }

    const m = hash.split('/');

    const mode = decodeURIComponent(m[1]||'');

    const ability = decodeURIComponent(m[3]||'');

    renderView(mode, ability);

  }

  window.addEventListener('hashchange', applyRoute);

  applyRoute();

})();

</script>

</body>

</html>