

```

// Planning.jsx <v2025-08-29h>
// Changes in this drop:
//   - Add day preview button: black bg w-4 mr-1
//   - Add zones: base default "zone 1"; added zones default to "zone 2", "zone 3", ...
//   - Add zones/shift type inputs uniform sizing
//   - Add Extra footer padding so buttons aren't tight to borders
//   - Add keeps history upsert & structure intact
import React, { useEffect, useState } from 'react';
import { Button } from '../components/ui/button';
import { Card, CardContent, CardHeader } from '../components/ui/card';
import { Input } from '../components/ui/input';
import { Label } from '../components/ui/label';
import { Badge } from '../components/ui/badge';
import { Dialog, DialogContent, DialogHeader, DialogTitle, DialogTitleText } from '../components/ui/dialog';
import { ChevronLeft, ChevronRight, Plus, X } from 'lucide-react';

import {
  clone,
  fnsISO,
  fmUK,
  fmUKR,
  addDays,
  startOfWeekMon,
  weekNumberISO,
} from '../util';

const scoreColor = (v) => (v >= 3 ? "#10b981" : v >= 2 ? "#ffbf24" : "#ef4444");

// upsert history by (recruiterId, dateISO, _rowKey)
const upsertHistory = (list, row) => {
  const key = (r) => `${r.recruiterId}${r.dateISO}${r._rowKey ?? -1}`;
  const map = new Map(list.map(r)>>[key(r), r]);
  map.set(key(row), r);
  return Array.from(...map.values());
}

// numeric sanitizer (keeps "", or digits only)
const onlyNum = (s) => {
  const t = String(s ?? '');
  if (t === '') return '';
  const m = t.replace(/\D/g, '');
  return m ? t.replace(/\D/g, '') : '';
}

export default function Planning({ recruiters, planning, setPlanning, history, setHistory }) {
  const [weekStart, setWeekStart] = useState(() => fnsISO(startOfWeekMon(new Date())));
  const weekNum = weekNumberISO(dateWeekStart);

  // ensure structure
  useEffect(() => {
    setPlanning((prev) => {
      const next = prev || {};
      if (!next[weekStart]) next[weekStart] = { days: {} };
      for (let i = 0; i < 7; i++) {
        const d = fnsISO(addDays(new Date(weekStart), i));
        if (!next[weekStart].days[d]) next[weekStart].days[d] = { teams: [] };
      }
      return next;
    });
  }, [weekStart, setPlanning]);

  const dayData = (iso) => planning?.[weekStart]?.days?.[iso] ?? { teams: [] };

  // Edit Day
  const [editDateISO, setEditDateISO] = useState(null);
  const [draft, setDraft] = useState(null);

  const openEdit = (iso) => {
    const d = clone(dayData(iso));
    d.teams = (d.teams || []).map((t, ti) => ({ ...t, zone: "zone 1", // default base
      extraZones: (t.extraZones || []).map((z, idx) => `zone ${idx + 2}`), // ensure names
      project: t.project || "Hello Fresh", shiftType: t.shiftType || "D2D",
      rows: (t.rows || []).map((r, ri) => ({ ...r, recruiterId: r.recruiterId || "", hours: r.hours ?? "", commissionMult: r.commissionMult ?? "", score: r.score ?? "", // keep field names; labels elsewhere say "box 2/Box 2+Box 4/Box 4"
      box2_noDisc: r.box2_noDisc ?? "", box2_disc: r.box2_disc ?? "", box4_noDisc: r.box4_noDisc ?? "", box4_disc: r.box4_disc ?? "", })),
    ));
    setEditDateISO(iso);
    setDraft(d);
  };

  const closeEdit = () => { setEditDateISO(null); setDraft(null); };

  const addTeam = (ti, patch) => {
    const teams = (patch || {}).teams || [];
    const team = { ...patch, id: teams.length };
    teams.push(team);
    return { ...patch, teams };
  };

  const delTeam = (ti) => {
    const teams = (patch || {}).teams || [];
    const team = teams.filter((t, i) => i !== ti);
    return { ...patch, teams };
  };

  const setTeam = (ti, patch) => {
    const teams = (patch || {}).teams || [];
    const team = teams.filter((t, i) => i !== ti);
    team[ti] = patch;
    return { ...patch, teams };
  };

  const addZoneAt = (ti, zi, val) => {
    setDraft((d) => {
      const teams = clone(d.teams || []);
      const team = teams.filter((t, i) => i === ti);
      arr[zi] = val;
      team[ti].extraZones = arr.map((z, idx) => z.startsWith("zone ") ? `zone ${idx + 2}` : z);
      return { ...d, teams };
    });
  };

  const delZoneAt = (ti, zi) => {
    setDraft((d) => {
      const teams = clone(d.teams || []);
      const team = teams.filter((t, i) => i === ti);
      arr.splice(ti, 1);
      team[ti].extraZones = arr.map((z, idx) => z.startsWith("zone ") ? `zone ${idx + 2}` : z);
      return { ...d, teams };
    });
  };

  const saveDay = () => {
    if (!editDateISO) return;
    const iso = editDateISO;
    for (const t of t.rows || []) {
      for (const r of t.rows || []) {
        const sum = Number(r.score || 0);
        const sum2 = Number(r.box2_noDisc || 0) +
          Number(r.box2_disc || 0) +
          Number(r.box4_noDisc || 0) +
          Number(r.box4_disc || 0);
        if (sum > sum2) {
          alert("Box 2/Box 4 totals cannot exceed Score.");
          return;
        }
      }
    }
    // write planning
    setPlanning((prev) => {
      const next = clone(prev || {});
      if (!next[weekStart]) next[weekStart] = { days: {} };
      next[weekStart].days[iso] = clone(draft);
      return next;
    });
    setHistory(history);
  };

  return (
    <Card className="flex gap-4">
      <CardContent>
        <Form>
          <Input type="text" value={iso} onChange={() => setEditDateISO(fnsISO(addDays(new Date(weekStart), -7)))}>
          <Button>Prev</Button>
          <Button>Next</Button>
          <ChevronLeft className="h-4 w-4" /> Prev
          <ChevronRight className="h-4 w-4" /> Next
          <Badges style={{ background: "#fcfcfc" }} className="whitespace nowrap">
            <span>Week</span>
            <span>${weekNum}</span>
          </Badges>
          <Edit Day variant="outline" onClick={() => setWeekStart(fnsISO(addDays(new Date(weekStart), 7)))}>
            <span>Edit Day</span>
          </Edit Day>
        </Form>
      </CardContent>
    </Card>
  );
}

/* ----- Render ----- */
return (
  <div className="grid gap-4">
    <div>
      <h3>Week header with nav</h3>
      <div>
        <Button onClick={() => setWeekStart(fnsISO(addDays(new Date(weekStart), -7)))}>
          <span>Prev</span>
        </Button>
        <ChevronLeft className="h-4 w-4" /> Prev
        <ChevronRight className="h-4 w-4" /> Next
        <Badges style={{ background: "#fcfcfc" }} className="whitespace nowrap">
          <span>Week</span>
          <span>${weekNum}</span>
        </Badges>
        <Edit Day variant="outline" onClick={() => setWeekStart(fnsISO(addDays(new Date(weekStart), 7)))}>
          <span>Edit Day</span>
        </Edit Day>
      </div>
    </div>
    <div>
      <h3>Day Header</h3>
      <div>
        <Edit Day variant="outline" onClick={() => openEdit(editDateISO)}>
          <span>Open Edit</span>
        </Edit Day>
      </div>
    </div>
  </div>
);

/* Days Grid */
<div className="grid gap-4 grid-gap-1 md:grid-cols-1 md:grid-cols-3 xl:grid-cols-7">
  {arrayFrom({length: 7}).map((_, i) => (
    <DayCard key={i} i={i} />
  ))}
</div>

/* Edit Day */
<Dialog
  title="Edit Day"
  open={editDateISO}
  onClose={() => closeEdit()}
>
  <div>
    <Form>
      <Input type="text" value={editDateISO} onChange={() => openEdit(editDateISO)} />
      <Label>Project</Label>
      <Select>
        <option value="Hello Fresh">Hello Fresh</option>
        <option value="Door-to-Door">Door-to-Door</option>
        <option value="Events">Events</option>
      </Select>
      <Input type="text" value={rows[0].recruiterId} onChange={() => setTeam(ti, { project: e.target.value })} />
      <Table border="1" width="100%">
        <thead>
          <tr>
            <th>Recruiter ID</th>
            <th>Hours</th>
            <th>Commission Mult</th>
            <th>Score</th>
            <th>Box 2 Disc</th>
            <th>Box 4 Disc</th>
          </tr>
        </thead>
        <tbody>
          {rows.map((r, ri) => (
            <tr>
              <td>{r.recruiterId}</td>
              <td>{r.hours}</td>
              <td>{r.commissionMult}</td>
              <td>{r.score}</td>
              <td>{r.box2_noDisc}</td>
              <td>{r.box4_noDisc}</td>
            </tr>
          ))}
        </tbody>
      </Table>
    </Form>
  </div>
</Dialog>

```