DUN Task Management Application – **Hours Panel Features**

---

0. Scope

A self-contained prototype demonstrating: **click-to-create tasks**, **drag-to-reposition**, **bottom-handle resize** (15-minute increments), **inline rename**, **full modal edit**, and **delete**—with mandatory limit lines and a current-time line. No persistence, no additional panels.

If integrating with existing DUN code, every effort should be made to:

- Use existing architecture, functions, and libraries.
- Avoid rewriting features already implemented elsewhere.
- Follow "Don't Repeat Yourself" (DRY) principles.
- Ensure compatibility with existing data handling methods (e.g., persistence via local storage, Google Sheets, or remote DB, abstracted through common functions).
- Clearly comment all code to aid integration and reuse.

---

1. Product & Technical Setup

- **Languages:** Plain HTML, CSS, vanilla JavaScript (ES6).
- **Libraries:** No libraries unless absolutely required. Use only pre-approved, lightweight JS libraries if necessary. **Do not use InteractJS or switch to new drag libraries**—use the one already working.
- **Hosting:** Static files (e.g., GitHub Pages). No backend/database assumed for this prototype.
- **Debug Flag:** Global `debug` boolean logs create/drag/resize/delete/limit-drag events.

---

2. Core Layout

2.2 Fixed Top Banner

- Full-width dark banner (`#212121` , `position: fixed; top: 0; width: 100%; z-index: 1000`).
- Left: **logo + "DUN"** text.
- Right: **Task quick-add input**, `[Add]` button, and **Filter dropdown**.

2.3 Hours Panel (primary view)

- Vertical timeline column covering **12 AM → 12 AM**.
- **Row grid:** visible hour lines every hour; half-hour dashed; tasks may start/stop at **0, 15, 30, 45** minutes.
- Left-side hour labels show plain numbers (`12` , `1` , `2…`) with **PM hours bolded** for quick distinction.

---

3. Interaction Rules

3.1 Create Task

1. Double Click empty slot ⇒ prompt asks for task name.
2. Creates a 60-minute default block anchored to the nearest 15-min grid.
3. After creation, the viewport does **not** auto-scroll (user scrolls manually).

3.2 Drag & Drop Re-position

- Drag block vertically; snap to nearest 15-min increment.
- Reject drop if overlap; flash red border 400 ms then revert.

3.3 Resize Duration

- Bottom edge resize handle (`cursor: ns-resize`).
- Snap in 15-minute increments; minimum 15 min; can extend to next task or midnight.
- Reject/flash red if overlap, snap back to the first open 15min increment.

3.4 Delete Task

- Trash icon (`×`) top-right of block.
- Confirm via `confirm()`; on accept, remove block.

3.5 Inline Rename & Modal Edit

- **Double-click title text** ⇒ switches to inline editable `<input>`; Enter to save, Esc to cancel.
- **Double-click anywhere else** on block (or click Pencil icon) ⇒ open **Task Edit Modal** (Exact same modal code as is already defined or implemented. If it is not defined yet, then use the following fields:

Task Edit Modal Fields

Field	Notes
Title	Required text field
Start Time	Time picker locked to 15-min increments
Duration	Numeric stepper (15-min steps)
Description	Multiline notes
Priority	Optional flag toggle
Delete Button	Removes task (same confirm)

- Modal saves on blur and on close. Save button also saves and closes.

3.6 Visual Feedback

- Drag/resize highlight = brighter accent outline.
- Ghost placeholder = accent 30 % opacity.
- Hover lightens block fill by 20 %.

---

4. Time Lines (Mandatory)

4.1 Current-Time Line

- Solid accent line (1 px) across full width.
- Small clock icon off right edge.
- Updates 5 minutes;

4.2 Limit Lines

Line	Default Time	Visual	Dragging
STOP	6:00 PM	3×-thick diagonal-striped accent line + octagon icon	Draggable vertically
SLEEP	11:00 PM	3×-thick diagonal-striped accent line + "ZZZ" icon	Draggable vertically

- The value representing the remaining-time before the next limit line (e.g., " `9:30` rem") displayed just below each limit line icon.
- Drag handle, label, time indicator, and limit line all move together; snap to 15-min increments.

---

5. Styling

- Dark theme: `#212121` background, `#dddddd` body text, `#00CEF7` accent.
- Hour lines: solid `#333` ; half-hours: dashed `#2a2a2a` .
- Task blocks: accent fill (`rgba(0,206,247,0.85)`), white text.
- Icons & handles: accent 100 % opacity; fade to 70 % on idle.

---

6. Data (runtime-only)

```
const tasks = [
  { id, title, startIndex, durationSteps /* 15-min units */ }
];
```

Page reload resets data.

---

7. Logging

- With `debug=true`, console logs: `create`, `dragStart`, `dragEnd`, `resize`, `delete`, `limitDrag`.

End of Standalone Hours-View Prototype Requirements