⟳ ChatGPT

WoodLab Configurator – Definitive Implementation Guidelines (v1.1)

This document replaces **WL config guidelines v1.txt**. It removes optional wording and resolves ambiguities. All directives below are **mandatory** unless explicitly marked "OPTIONAL".

---

1. Project Scope & Non-Goals

Aspect	Requirement
Purpose	Deliver an interactive, static mock-up that mirrors the final WoodLab Configurator UI/UX (Three-stage wizard, 3D viewer, sidebar options, summary & export).
Server Code	**None.** All logic runs in the browser.
Data	Use hard-coded placeholder data only. No external APIs.
3D Assets	Placeholder primitive geometry rendered by Three.js.
Out of Scope	Real e-commerce back-end, authentication, payment, production-quality 3D assets.

2. Locked Technology Stack

Layer	Library / Standard	Notes
3D	**Three.js r160**	Import as ES6 module from `./js/lib/three.module.js`.
UI (Desktop & Mobile)	Vanilla JS + **Tailwind v3**	Avoid React/Angular/Vue to keep build pipeline minimal.
PDF export	**jsPDF 2.5.1** + **html2canvas 1.5.1**	Bundle locally.
Icons	**Hero-icons (SVG)**	Place in `/assets/icons`.

No other frameworks may be added without written approval.

3. Canonical File & Folder Layout (fixed)

```
/
├ index.html              # entry point (relative links only)
├ .nojekyll               # ensure GH Pages serves files verbatim
├ css/
│     └ styles.css         # compiled Tailwind output
├ js/
│     ├ main.js            # bootstrap & global state
│     ├ viewer.js          # 3D scene module
│     ├ ui/                # one file per UI segment
│     └ lib/three.module.js  # vendored libs
├ assets/
│     ├ images/
│     ├ icons/
│     └ models/            # simple .glb placeholders if used
└ README.md
```

Do **not** deviate from this tree; automated deployment scripts depend on it.

4. Build & Deployment Pipeline

1. Run `npm i` to fetch **Tailwind** and dev-only tooling.
2. `npm run build` must:
3. compile Tailwind → `/css/styles.css` (JIT, purge unused classes).
4. copy libs into `/js/lib`.
5. **NOT** minify source JS (easier code-review).
6. Commit the `/` tree to `gh-pages` branch.
7. Verify at `https://<user>.github.io/WoodLab/index.html`.

5. Component Specifications

5.1 Stage Bar

- HTML: ordered list inside `<nav id="stage-bar">`.
- Exactly **three** stages: *Model*, *Customize*, *Summary*.
- Activate via `aria-current="step"`.
- Non-completed stages are inert (`pointer-events:none`).

5.2 3D Viewer

- Parent: `<div id="viewer">` flex-expands.
- Initialise once; subsequent stage changes **never** recreate renderer.
- Controls: `OrbitControls` with damping; `reset()` bound to ↻ button.
- On window resize call `viewer.resize()` within a debounced 200 ms handler.

5.3 Sidebar

- `<aside id="sidebar">`, scrollable, `overflow-y:auto`.
- Each stage owns one `<section>`; toggle via `.hidden`.
- Option card markup:

```html
<button class="option-card" data-category="model" data-id="mdl-01" data-
price="100">
  <img ... />
  <span class="label">Model 1</span>
</button>
```

- Exactly **one** selected card per category; enforce via JS.

5.4 Banners & Tooltips

- Use a single reusable component in `/js/ui/banner.js`.
- Banner auto-dismiss after 5 s unless `type === 'error'`.
- Tooltip text lives in `data-tip` attribute; JS positions on hover/focus.

5.5 Help Drawer

- `<aside id="help-drawer">` off-canvas right, width 320 px.
- Open/close via `aria-expanded` toggle; trap focus.

5.6 Summary & Export

- On entering stage 3 call `captureSnapshot()` and populate summary list.
- PDF button runs `exportPDF()` → downloads *WoodLab-Summary.pdf*.
- Provide *Restart* button that resets global state and returns to stage 1.

6. State Management

```js
const state = {
  stage: 1,                 // 1 | 2 | 3
  selections: { model:null, options:{} },
  pricing: { base:0, extras:0, total:0 }
};
```

All UI modules **read-only** observe state. Only `main.js` mutates state and dispatches `Event("statechange")`.

7. Placeholder Assets Policy

- 3D: coloured `BoxGeometry` sized 1 × 1 × 1 m.

- Images: 1 : 1 PNG 256 × 256, gray background, centered label text.
- Mark every placeholder `<img>` with `alt="placeholder"`.

8. Accessibility

- Conform to **WCAG 2.2 AA**.
- Never remove keyboard focus outlines (use `:focus-visible`).
- All dynamic text updates inside `[aria-live="polite"][aria-atomic="true"]` regions.
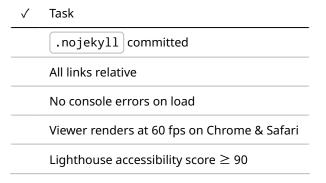- Escape closes help drawer and any modal banners.

9. Responsive Breakpoints (fixed)

Width (px)	Layout
$\geq$ 1024	Two-column (sidebar right 30 %).
768–1023	Sidebar collapsible drawer, default open.
$\leq$ 767	Viewer top, sidebar full-width below. Stage bar sticks bottom.

10. Export Features – Implementation Checklist

- [x] `captureSnapshot()` → PNG $\leq$ 500 kB.
- [x] `exportPDF()` includes snapshot + selected options + total price.
- [x] After successful PDF export, fire `Event('exported')` so analytics can hook.

11. GitHub Pages Compliance Checklist

✓	Task
	`.nojekyll` committed
	All links relative
	No console errors on load
	Viewer renders at 60 fps on Chrome & Safari
	Lighthouse accessibility score $\geq$ 90

12. Definition of Done

A build is **Done** when: 1. All checklist items in §10 & §11 are checked. 2. QA walkthrough completes without functional defects. 3. This document is committed at `/docs/Guidelines_v1.1.md`.

Change Log

- **v1.1 (2025-06-07)** – First unambiguous edition derived from *WL config guidelines v1*.