

Roadmap & Improvement Plan

Planning document for **Professional Markdown PDF Exporter**. It records what the project still needs before its first public release and the backlog of improvements that would make it a stronger product.

Effort key: **S** ≈ under an hour · **M** ≈ half a day · **L** ≈ a day or more.

1. Current status (v0.1.0)

The extension is feature-complete against its v1 specification and verified end-to-end:

- Three commands — quick chooser, direct Arabic, direct English.
 - Explorer + editor context menus, Command Palette entries.
 - System browser auto-detection (Linux / macOS / Windows) with manual override.
 - Full pipeline: Markdown → HTML → image resolution → A4 PDF via `puppeteer-core`.
 - Arabic (RTL) and English (LTR) professional print CSS presets.
 - Overwrite protection, progress UI, dedicated Output Channel.
 - Scripts disabled during render (no Markdown-derived JavaScript executes).
 - `compile`, `lint`, and 13 unit tests all pass; both sample PDFs render correctly.
-

2. What we need before the first Marketplace release

These are prerequisites for publishing — mostly outside the codebase.

#	Item	Owner	Effort	Notes
2.1	Upgrade local toolchain to Node 20 LTS	Maintainer	S	<code>vsce 3.x</code> requires Node 20+; <code>npm run package</code> fails on Node 18.
2.2	Replace <code>media/icon.png</code> with a designed icon	Maintainer / designer	S–M	Current icon is a generated placeholder (128×128 document glyph).
2.3	Confirm the <code>i-chehab</code> publisher exists on the Marketplace	Maintainer	S	Create via the VS Code publisher portal if not.
2.4	<code>vsce login i-chehab</code> → <code>npm run package</code> → <code>vsce publish</code>	Maintainer	S	Produces and uploads the <code>.vsix</code> .
2.5	<code>git init</code> + initial commit + push to GitHub	Maintainer	S	Repo URL already set in <code>package.json</code> .
2.6	Manual smoke test inside the Extension Development Host (F5)	Maintainer	S	Verify menus, when clauses, and the three commands in a real VS Code window.
2.7	Verify README images resolve on the Marketplace listing	Maintainer	S	Relative <code>media/*.png</code> paths must resolve against the published repo.

Open decision

- **Context-menu label.** The spec asks for the Command Palette title `Markdown PDF: Export PDF...` (Section 7) *and* a context-menu label `Export Markdown to PDF...` (Section 8). VS Code uses one `title` for both surfaces, so only one can be literal. Current choice favours Section 7. Decide whether to keep `Export PDF...` or switch to `Export Markdown to PDF...` (which makes the palette read `Markdown PDF: Export Markdown to PDF...`).

3. Improvement backlog

Tier 1 — highest impact on output quality

#	Improvement	Why it matters	Effort
3.1	Code syntax highlighting	Code blocks render as plain grey today. <code>markdown-it-highlightjs</code> colourises at render time in Node — fully compatible with the no-JavaScript export policy. Immediate visual upgrade.	M
3.2	Bundle Arabic fonts	Output currently depends on the user's installed fonts; a machine lacking a good Arabic face produces inconsistent PDFs. Embedding Noto Naskh / Noto Kufi (woff2, base64 in the preset CSS) makes output deterministic everywhere. This underpins the core Arabic value proposition.	M
3.3	PDF outline / bookmarks from headings	<code>page.pdf({ tagged: true })</code> turns headings into a navigable PDF outline — essential for long reports, memos, and specs.	S–M
3.4	Footnotes & task lists	<code>markdown-it-footnote</code> and <code>markdown-it-task-lists</code> . Footnotes are core to legal and professional documents; task lists are widely used.	S
3.5	HTML <code><h1></code> title extraction	<code>extractFirstH1</code> only matches Markdown <code>#</code> headings, so documents using <code><h1></code> inside a <code>.cover-page</code> fall back to the filename for the PDF <code><title></code> . Parse the rendered HTML for the first <code><h1></code> as a fallback.	S

Tier 2 — product features

#	Improvement	Why it matters	Effort
3.6	Automatic language detection	Inspect document text to pre-select Arabic vs English, keeping manual override. Removes a step for most users.	M
3.7	Custom output folder / "ask every time"	<code>outputPathResolver</code> is already a switch over <code>OutputMode</code> ; add <code>customFolder</code> , <code>askEveryTime</code> , and a <code>workspace /pdf mode</code> .	M
3.8	HTML preview before export	A webview showing the styled document so users can check layout before producing a PDF.	M–L
3.9	Batch export of a folder	Export every <code>.md</code> in a directory, reusing a single browser instance for speed.	M
3.10	Cancellable export progress	<code>withProgress</code> is currently <code>cancellable: false</code> ; allow cancellation and close the browser cleanly.	S
3.11	PDF metadata	Set document title / author / subject. Requires a post-process step (e.g. <code>pdf-lib</code>) since <code>page.pdf()</code> does not expose metadata.	M
3.12	Configurable page setup	Settings for page size, margins, and a default preset, layered over the CSS <code>@page</code> rule.	M

Tier 3 — repository & engineering infrastructure

#	Improvement	Why it matters	Effort
3.13	GitHub Actions CI	Run <code>lint</code> + <code>compile</code> + <code>test</code> on every push/PR. Expected for a credible public repo.	S
3.14	esbuild bundling	Bundle <code>out/</code> into a single file to shrink the <code>.vsix</code> and speed up activation.	M
3.15	UI localization (l10n)	Translate command titles and messages via <code>package.nls.json</code> / <code>vscode.l10n</code> — a natural fit given the Arabic audience.	M
3.16	VS Code integration tests	<code>@vscode/test-electron</code> to assert command registration and <code>when</code> clauses inside a real host, complementing the pure unit tests.	M
3.17	Contributor docs	<code>CONTRIBUTING.md</code> , issue / PR templates.	S

Considered and intentionally excluded

- **Mermaid / live diagrams** — would require executing JavaScript in the page, which conflicts with the security model. Could only be supported via build-time pre-rendering.
 - **Telemetry / analytics** — deliberately omitted; the extension makes no external network requests.
-

4. Suggested sequence

1. **Ship v0.1.0** — complete the Section 2 checklist and publish.
2. **v0.2.0 — output quality:** items 3.1, 3.2, 3.5 (visible polish, no risk).
3. **v0.3.0 — document structure:** items 3.3, 3.4.
4. **v0.4.0 — workflow features:** items 3.6, 3.7, 3.10.
5. **Ongoing:** infrastructure items 3.13–3.17 folded in alongside feature work.