M.A.K SOLUTIONS • COMMUNITY DROP

FREE AI APP SECURITY PROMPT PACK

5 practical prompts I use as a release-check workflow for AI-built websites, apps and Chrome extensions.

BUILD FAST. CHECK HARD.

THEN LAUNCH.

Complimentary prompt pack • Save it • Reuse it • Share it

CONNECT WITH M.A.K SOLUTIONS

@mak_solutions.in

DMs are open · Quick replies

Instagram @mak_solutions.in

WHY I MADE THIS

AI can build fast. Security does not happen automatically.

This is the release-check layer I use around AI-assisted builds. You can copy the prompts into your coding AI after the project is built.

1. **BUILD**
   Finish the website, app or extension first.
2. **AUDIT**
   Run the relevant prompts from this pack.
3. **FIX + RE-TEST**
   Make fixes, then verify the original product still works.

THE RULE

A prompt is not proof of security. It becomes useful only when the AI can inspect the real source code, explain what it found, make the fix, and report what it could not test.

PROMPT 01 / 05

Secret Leak Prevention

Use this before sharing source code, publishing a repository or deploying a build.

COPY / PASTE THIS PROMPT

Inspect the entire project for exposed secrets before deployment.

Find API keys, passwords, tokens, database URLs, private credentials and signing secrets in source, config, comments and logs.

Move private secrets to environment variables or the platform's secret store. Never expose server-only secrets in browser code.

Check .env handling. Ensure real .env files are excluded where appropriate and create an .env.example containing placeholders only.

Check frontend-exposed environment variables and confirm they contain public-safe values only.

Check logs, API responses and error handlers for accidental credential leakage.

If a secret appears to have been committed or shared previously, do NOT print it. Flag it for immediate rotation.

REPORT: list each issue, file/location, severity, fix made, and anything that still requires manual action.

WHAT THIS CHECK IS TRYING TO STOP

A private key or credential accidentally travelling inside a ZIP, browser bundle, Git history, log or public deployment.

PROMPT 02 / 05

Personal Data Flow Audit

Use this when your build handles customer details, logins, forms, records or uploaded files.

COPY / PASTE THIS PROMPT

Audit how personal and sensitive data moves through this project.

Map every collection point: name, phone, email, address, password, payment-related data, device data, uploaded files and customer records.

For each data type show: COLLECTED -> STORED -> TRANSMITTED -> THIRD PARTY -> DELETED.

Inspect console logs, server logs and error handlers. Remove or redact unnecessary personal data.

Passwords must never be stored in plaintext. Verify the authentication system uses an appropriate password-hashing mechanism where passwords are managed by this project.

Review cookies, localStorage, IndexedDB and other browser storage. Flag sensitive data that should not be exposed to page JavaScript.

Review third-party SDKs/APIs and identify exactly what user data is sent to each service. Remove unnecessary fields.

Review API responses so they return only the fields required by the client.

REPORT: data map, risks found, fixes made, and anything that could not be verified.

Best for: billing tools, lead systems, customer apps, login systems and any project that stores real user information.

PROMPT 03 / 05

Pre-Deploy Production Audit

Run this when the build looks finished and is about to go live.

COPY / PASTE THIS PROMPT

Treat this project as a production release candidate. Inspect it and fix confirmed launch blockers without redesigning the product.

Verify required environment variables are referenced correctly. Critical missing values should fail safely with a clear server-side error.

Find and remove unsafe debug code, hardcoded test credentials, test-only endpoints and accidental backdoors.

Check client-facing errors. Do not expose stack traces, file paths, database details, secrets or internal infrastructure.

Review security headers appropriate to this stack, including content-type protection, framing protection, transport security and a practical Content Security Policy.

Review CORS. Do not allow every origin unless the API is intentionally public.

Review rate limiting for login, signup, password reset, OTP, activation and other abuse-prone endpoints.

Review production database/network configuration and confirm authentication and encrypted transport are used where applicable.

REPORT: PASS / FAIL for each check, exact fixes made, and items not testable in this environment.

Do not accept 'looks secure' as a result. Ask for evidence: file names, configuration inspected and tests actually run.

PROMPT 04 / 05

Auth, Payments & Critical Logic

Use this for projects with accounts, roles, paid access, activation codes or server-side business rules.

EDIT THE BRACKET, THEN COPY / PASTE

This project has [DESCRIBE AUTH / PAYMENTS / ACTIVATION / COMPLEX SERVER LOGIC]. Audit the critical paths.

AUTHORIZATION: inspect every protected route/action. Check whether changing a user ID, order ID, record ID or URL can expose another user's data. Ownership checks must occur on the trusted/server side.

SESSIONS/AUTH: inspect token/session expiry, logout behaviour, password-reset flows and role checks. Do not rely only on hidden UI.

PAYMENTS: never trust price, quantity, discount, plan or payment status supplied by the browser. Recalculate/verify critical values server-side.

WEBHOOKS: where payment providers are used, verify webhook authenticity/signatures before granting paid access.

INPUTS: inspect forms, URL parameters, API fields and database operations for injection and unsafe rendering.

UPLOADS: validate file type and size on the trusted side and prevent executable/untrusted content from becoming active code.

REPORT EACH ISSUE: vulnerability, affected location, realistic impact, exact fix, and verification performed.

This is the prompt I would prioritize for paid products and anything where one user must never access another user's records.

PROMPT 05 / 05

Attacker's Perspective Review

Final pass: stop thinking like the builder and look for ways the product can be abused.

COPY / PASTE THIS PROMPT

Review this project from an attacker's perspective. Do not perform destructive actions or attack third-party systems. Inspect/test only this supplied project and its safe local/test environment.

Check ID manipulation: can one user request another user's records by changing IDs?

Check login bypass: do protected endpoints reject missing, expired and malformed authentication?

Check privilege escalation: can a normal user reach admin actions by URL guessing, request modification or client-side role changes?

Check feature abuse: repeated signup, login, OTP, uploads, API calls, activation attempts, promo/referral logic or resource exhaustion.

Check content/input injection: unsafe HTML/JavaScript rendering, unsafe queries, filenames and URL parameters.

Check internal exposure: .env, .git, debug pages, internal docs, verbose health endpoints, secrets in errors.

Check business logic: negative/zero values, duplicate requests, reused activation codes, repeated trials, discount stacking or bypassing paid access.

PRIORITIZE: data theft -> unauthorized access -> secret exposure -> payment/logic abuse -> resource abuse.

REPORT: what was tested, what was only inspected, what was fixed, and what remains unverified.

BONUS

M.A.K Master Release Command

Use this short command after the five checks to force a clean final verification.

COMPLIMENTARY MASTER PROMPT

You are auditing an EXISTING build. Do not redesign it or remove working features just to make checks pass.

Re-test the complete product after all security fixes: loading, navigation, forms, save/edit/delete, downloads/uploads, login/logout, storage, mobile layout, extension behaviour and deployment configuration where applicable.

Classify every conclusion as one of: VERIFIED / TESTED / INSPECTED ONLY / NOT TESTABLE HERE.

Never claim the project is 'fully secure'. Never fabricate a test result.

Give me a non-technical release report with: Critical Issues, Important Issues, Minor Issues, Fixed, Manual Action Required, Files Changed, Tests Performed, Known Limitations, and Final Deployment Readiness.

If a security fix broke an existing feature, repair the regression before calling the release complete.

MY SIMPLE WORKFLOW

BUILD -> PROMPT 1 -> PROMPT 2 -> PROMPT 3 -> PROMPT 4 (when relevant) -> PROMPT 5 -> FIX -> MASTER RELEASE COMMAND -> FINAL ZIP

REAL M.A.K SOLUTIONS WORK

5 EXTENSIONS.
5 DIFFERENT WORKFLOWS.

Built around real workflow problems - not generic demo concepts.
Connect directly for pricing.

Connect directly for details.

1 M.A.K InstaVault

INSTAGRAM ARCHIVE

- Posts, carousel media & Reels
- Captions, links & media archive
- Organised ZIP export
- Stop + Export workflow

DM FOR DETAILS

2 M.A.K Site Inspector

WEBSITE QA

- Discovers site pages & removes duplicates
- Desktop / mobile / tablet screenshots
- Basic QA + progress / retry
- Report + ZIP package

DM FOR DETAILS

3 M.A.K WebVault

WEBSITE ARCHIVE

- Archives public text, images & documents
- Links, SEO information & assets
- Page / site capture
- Desktop / mobile / tablet inspection

DM FOR DETAILS

4 M.A.K LeadVault

BUSINESS LEAD RESEARCH

- Checks public business websites
- Looks at home + likely contact pages
- Extracts published business contact details
- Excel-friendly CSV export

DM FOR DETAILS

5 M.A.K StoreVault

COMMERCE / STORE WORKFLOW

- Currently under development
- Part of the M.A.K Solutions extension line
- Built as a dedicated browser-based workflow tool

COMING SOON • DM FOR DETAILS

WANT ONE OF THESE - OR YOUR OWN CUSTOM EXTENSION?

DM us directly for details.

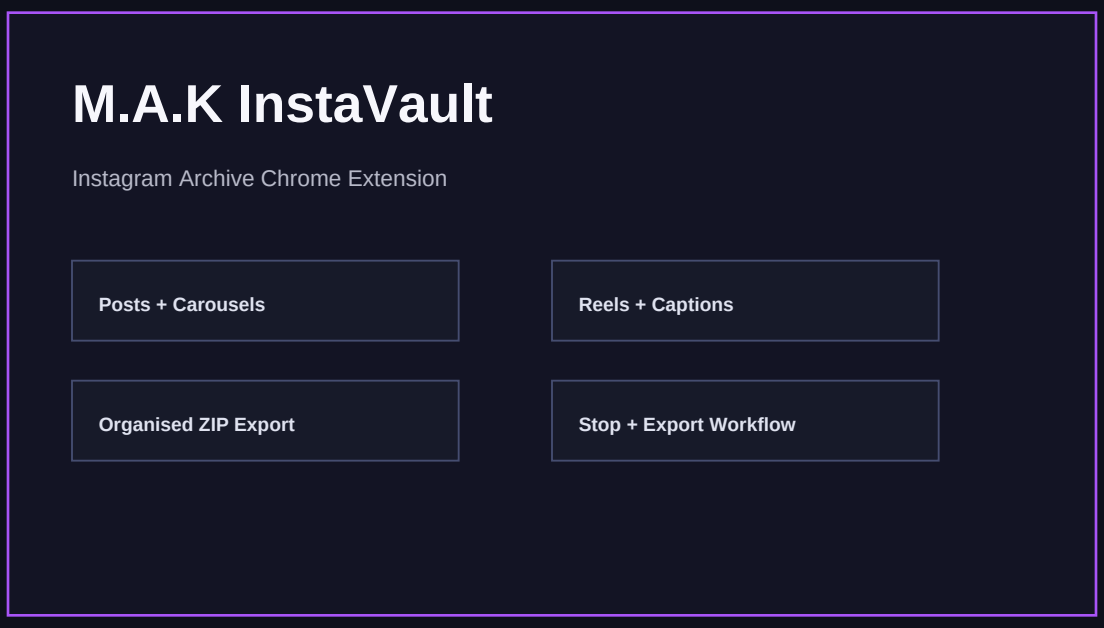
Tell me the workflow you want to automate.

Instagram @mak_solutions.in

REAL M.A.K WORK

M.A.K InstaVault

Chrome extension built around a real Instagram archive workflow.



BUILT AROUND THE WORKFLOW.

A practical extension example from the M.A.K Solutions tool line.

Connect by Instagram DM for product details and availability.

CONNECT WITH M.A.K SOLUTIONS

@mak_solutions.in

DMs are open · Quick replies

KEEP THIS PDF

BUILD.
CHECK.
SHIP BETTER.

FREE FROM M.A.K SOLUTIONS

Use the five prompts on your own AI-built project. Adapt the bracketed parts to your app. Re-run the attacker review whenever you add a major feature.

IMPORTANT

These prompts improve review discipline; they are not a substitute for professional security testing for high-risk systems, sensitive data at scale or significant financial transactions.

CONNECT WITH M.A.K SOLUTIONS

CONNECT WITH M.A.K SOLUTIONS

@mak_solutions.in

DMs are open · Quick replies

Instagram @mak_solutions.in