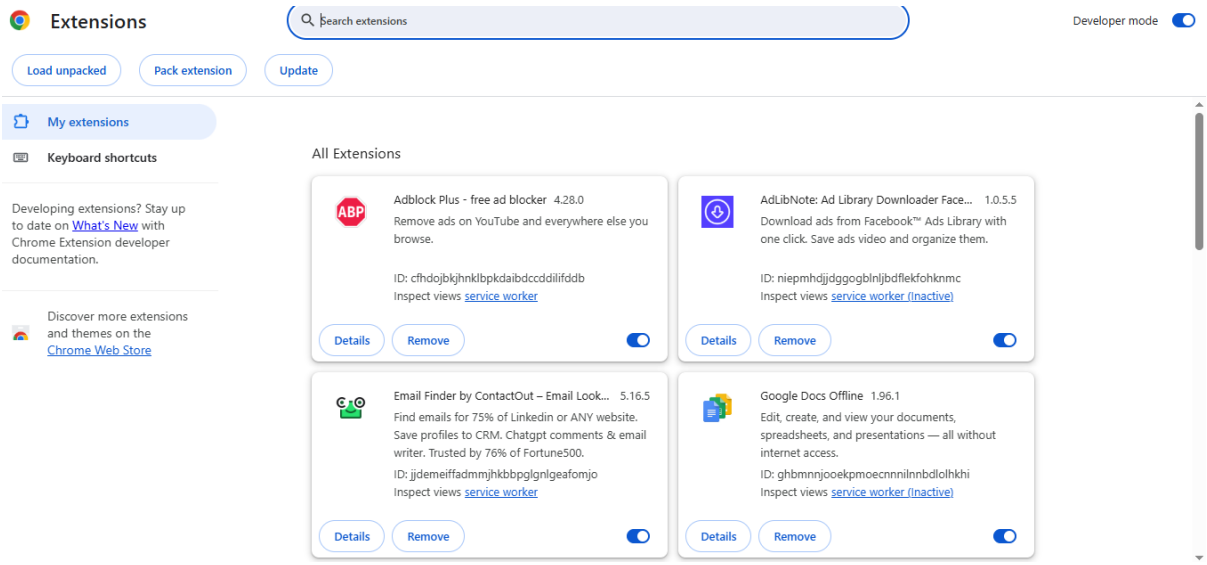
Task 7: Identify and Remove Suspicious Browser Extensions.

1. Check & List Down Installed Extensions



- List of all installed extensions and their IDs.

Chrome / Chromium / Edge (GUI)

1. Open browser → Menu → **More tools → Extensions** (or go to chrome://extensions/ or edge://extensions/).
2. Review each extension card:
   - Name, publisher, short description
   - Permissions shown (e.g., “Read and change all your data on the websites you visit”)
3. Click **Details** for any extension to see:
   - Extension ID
   - Permissions
   - Site access (“On click”, “On specific sites”, “On all sites”)
   - Extension version and “View in Chrome Web Store” link
4. If suspicious: click **Remove** (or toggle off to disable first).
5. Restart browser. Re-check behavior and performance.

Firefox (GUI)

1. Menu → **Add-ons and themes** (or about:addons).
2. Click **Extensions** → inspect each:
   - Name, publisher, permissions, install source
3. Click the three-dots → **Remove** (or **Disable**).
4. Restart browser and re-check.

2. What to look for (suspicious indicators)

- Permissions requesting **“read and change data on all websites”** for a simple extension (e.g., a wallpaper extension asking this).
- Very few installs / no reviews but aggressive permission requests.
- Recent installations you don’t recall, or names that mimic popular extensions but with typos (e.g., AdBlocker Pro vs AdBlock).
- Extensions that change your new tab, inject ads, redirect searches, or cause popups.
- Extensions installed by policy (forced installs) you did not approve.
- Background activity: high CPU, unusual network connections, or processes.
- Unsigned or unpacked extensions in developer mode.

3. Export a list of installed extensions (backup & evidence)

```
# Install jq if you want JSON parsing:
$sudo apt install -y jq   # Debian/Ubuntu
# Path to extensions (Chrome example)
EXT_DIR="$HOME/.config/google-chrome/Default/Extensions"
find "$EXT_DIR" -name manifest.json -print > ~/extensions_manifests.txt
# Show name, version and permissions for each manifest
while read m; do
  echo "---- $m ----"
  jq '{name: .name, version: .version, permissions: .permissions,
host_permissions: .host_permissions, update_url: .update_url}' "$m"
done < ~/extensions_manifests.txt > ~/installed_extensions.json
```

- Save these files into your repo evidence/ folder as installed_extensions.csv / installed_extensions.json.

4. Use browser tools to find which extension is active when a behavior occurs

Chrome Task Manager

- Open Chrome → **Menu → More tools → Task manager** (or press Shift+Esc).
- Sort by CPU or Memory to spot extension processes using resources. The “Task” column often includes extension names.

Firefox Performance / Debugging

- about:performance shows energy/perf impact of add-ons.
- about:debugging#/runtime/this-firefox lets you inspect extension background pages (advanced).

5. Removal steps (safe recommended order)

1. **Disable** suspected extension first (test for change).
2. If behavior persists, **Remove** it via UI:
    - Chrome: chrome://extensions/ → Remove
    - Firefox: about:addons → Remove
3. Restart browser. Confirm extension is gone and traffic/behavior normalized.
4. If extension reappears after removal → check for forced-install policies (next section) and for system-level installers.

6. Post-removal checks — verify persistence & clean up

- Reboot system & reopen browser.
- Re-run the export/list commands in Step 4 to ensure extension no longer present.
- Check OS for related software:
    - Windows: **Control Panel → Programs**, or Get-WmiObject -Class Win32_Product (slow) to find unknown apps.
    - Mac: /Applications and LaunchAgents (~/Library/LaunchAgents, /Library/LaunchAgents).
    - Linux: check ~/.local/share/ and system packages.
- Scan with a reputable anti-malware tool (Malwarebytes / Windows Defender / ClamAV) if you suspect malicious installs.
- Check browser shortcuts for injected args (Windows shortcut target may include --disable-extensions-except or add URLs).

Note:

How malicious extensions can harm users

- **Data exfiltration:** read content of web pages (banking, email) and send to attacker.
- **Credential theft:** inject fake login forms; capture keystrokes.
- **Ad injection / redirect:** inject ads into pages, monetize visitors.
- **Browser fingerprinting / tracking:** long-term tracking across all sites.
- **Cryptomining / performance drain:** run background scripts to mine crypto.

- **Persistence & reinstallation:** use system policies/update URLs to survive removal.