Tracker Blocker Chrome Extension – Project Report

1 Project Title

Tracker Blocker Chrome Extension for Enhanced Web Privacy

2 Objective & Motivation

In the modern digital ecosystem, personal privacy is constantly under threat.
Most websites use third-party trackers, analytics services, and advertisement scripts to monitor user behavior without explicit consent.
These trackers collect data such as browsing history, IP address, location, and device info to build user profiles for targeted advertising and analytics purposes.

Objective:
To design and implement a powerful yet lightweight Chrome Extension that helps users block trackers and maintain control of their privacy while browsing.
This solution focuses on efficiency, flexibility, and ease of use, leveraging Chrome's Manifest V3 and Declarative Net Request (DNR) API to achieve maximum performance and privacy protection.

3 Technology Stack & APIs Used

Component	Technology / API
Language	Vanilla JavaScript (ES6+)
Browser API	Chrome Declarative Net Request (DNR)
Storage	chrome.storage.local (persistent settings & logs)
UI	HTML + CSS (responsive & modern design)
Manifest Version	Manifest V3 (service workers, better security)
Alarms	chrome.alarms API (for scheduled auto-updates)

Why Declarative Net Request (DNR)?
Declarative Net Request is performant and recommended for Manifest V3. It allows rule-based blocking without intercepting individual network requests, making it far more efficient.

4 Detailed Features

- **Default Tracker Blocking**

Pre-defined list of popular known tracker domains (e.g., doubleclick.net, google-analytics.com).
Blocks requests matching these patterns.

- **Custom Tracker List**

Users can add custom domain patterns or URLs to block. Supports wildcards like *://*.example.com/*

- **Whitelist Management**

Allows users to maintain a list of trusted domains where blocking will not be applied, ensuring compatibility for essential services.

- **Remote Auto-Update**

Periodic (24-hour) remote list fetching system to automatically update and merge a public tracker list (DuckDuckGo, EasyList, or custom URL).
Ensures that the block list stays up-to-date without user intervention.

- **Block or Redirect Mode**

Flexible behavior:

- Block Mode → Stops the request entirely.
- Redirect Mode → Redirects blocked trackers to a blank image placeholder.

- **Live Statistics & Logs**

- Displays the number of trackers blocked in real-time (badge icon).
- Stores recent 200 block logs including timestamp, initiator URL, tab title, and rule ID.
- Clear logs anytime through the UI.

- **Alarm-Based Auto Update System**

Using chrome.alarms, the extension periodically performs remote auto-update in the background without user effort.

5️ Architecture & Design Workflow

1️ Initialization (on install or reload)

- Load default settings and stats from chrome.storage.local.
- Apply stored blocking rules using chrome.declarativeNetRequest.updateDynamicRules.

2️ User Interaction

- Via popup/options page: Users can modify custom trackers, whitelist domains, enable redirect mode, or configure remote auto-update settings.
- All configuration changes trigger re-application of rules dynamically.

3️ Blocking Mechanism

- Rules are defined as per DNR syntax → URL patterns + Excluded initiators (for whitelisted domains).
- Rules are chunked (max 8000 per update) to ensure Chrome's API limits are respected.

4 Remote Update Flow

- On Alarm trigger or setting change, fetch remote list from configured URL.
- Parse list safely (supports JSON and plain text with comments).
- Merge remote list into custom trackers → Update the dynamic rules.

5 Logging System

- Each blocked request is logged:-

```
{
    "ts": 1694096400000,
    "url": "https://doubleclick.net/ads.js",
    "initiator": "https://example.com",
    "tabId": 12,
    "title": "Example Site",
    "ruleIds": [1234]
}
```

6 Challenges Faced & Solutions

Challenge	Solution
Manifest V2 Deprecation	Migrated to Manifest V3 with service worker.
Large Rule Limits	Implemented chunked updates (max 8000 rules per batch).
Remote List Parsing	Built robust parser to handle JSON and plain text with comments.
Performance	Used Declarative Net Request API for efficient blocking instead of webRequest.
Logging Limit	Limited logs to recent 200 records to avoid storage bloat.

7 Testing Methodology

1. Test visiting known tracker domains:
   - https://doubleclick.net
   - https://google-analytics.com
   - https://ads.yahoo.com
2. Add custom tracker patterns via options → Confirm they block as expected.
3. Add trusted domains to whitelist → Confirm no blocking happens on these.
4. Toggle Redirect Mode → Confirm requests are redirected to blank image.
5. Enable Remote Auto-Update → Set a known remote list URL → Check if rules are auto-updated.
6. View Stats → Confirm blocked counts and recent logs are populated correctly.

7. Test badge icon updates per tab dynamically.

⚠️ Important Notes

- Uses Manifest V3 — Works on latest Chrome versions only.
- Fully client-side: no server dependencies.
- Max limit of rules (20,000) enforced by Chrome API.
- Ensure remote lists contain valid domain patterns or wildcard URLs.
- Whitelisted sites are exempt from blocking for safe browsing.

✅ Results Achieved

- Real-time blocking of tracking requests
- Fully functional custom tracker and whitelist management
- Automatic remote tracker list updates working
- Logs show complete details of blocked requests
- Badge icon shows accurate per-tab block count.
- No performance overhead; Chrome remained responsive

📂 Future Scope

- Support rule export/import as JSON
- Add advanced analytics dashboard
- Integrate with public EasyList tracker lists
- Improve UI/UX with animated live block indicator
- Option for scheduled enabling/disabling of blocking