



Atomic Gains: Prompt Assistant

Chrome Extension Technical Documentation



Overview / Introduction

Atomic Gains: Prompt Assistant is a Chrome Extension developed by **Unica Publications** designed to streamline AI-assisted writing and productivity tasks. It functions similarly to **AIPRM**, providing users with a curated library of prompt templates that can be easily applied to ChatGPT and other AI interfaces directly from their browser.

Users can:

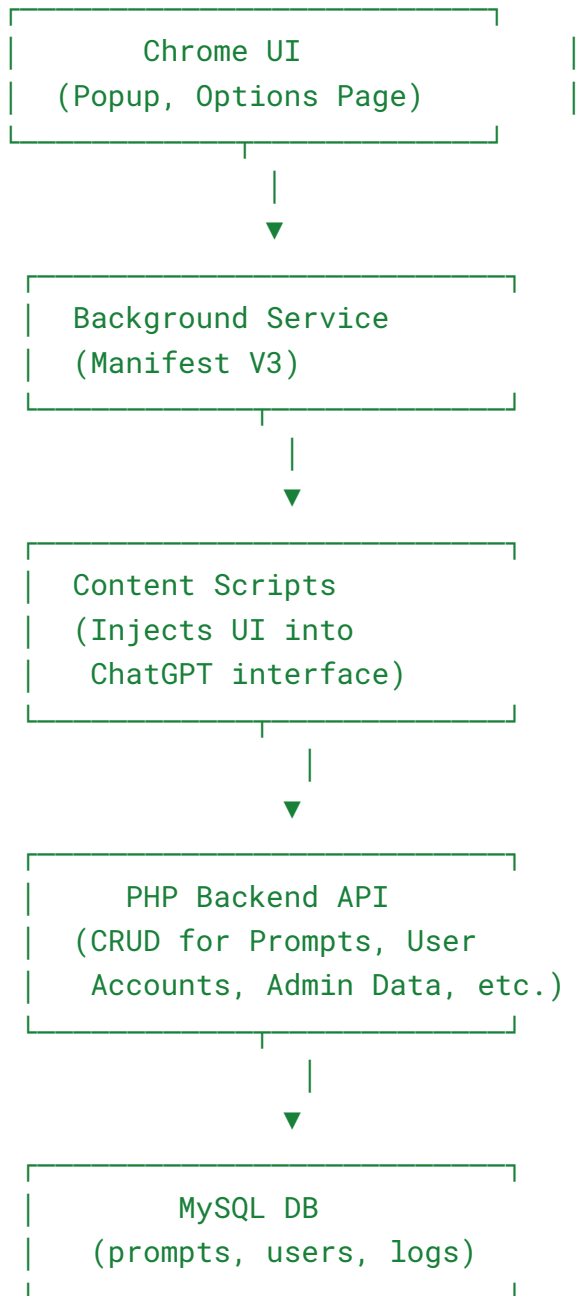
- Browse categorized prompt templates (marketing, SEO, storytelling, etc.)
- Save and favorite prompts
- Automatically inject prompts into supported AI chat windows
- Synchronize templates and favorites through the backend (PHP + MySQL)

This documentation provides a complete technical overview, installation instructions, architecture, and maintenance guide for developers and internal teams.



System Architecture

Overview Diagram



Tech Stack

Layer	Technology
Frontend (Extension)	HTML5, CSS3, JavaScript (ES6+), Manifest V3

UI Library	Vanilla JS + TailwindCSS (optional)
Backend API	PHP (REST API)
Database	MySQL 8.x
Authentication	Token-based (JWT or session token)
Browser Compatibility	Chrome (v120+), Edge (Chromium-based)
Packaging Format	.zip extension bundle



File and Folder Structure

```

digi-prompts-extension/
|
|— againlogofavicon32.png
|— againlogofavicon64.png
|— againlogofavicon128.png
|— againlogofavicon256.png
|— againlogofavicon512.png
|
|— background.js                # Background service worker
(Manifest V3)
|
|— changePassword.html          # Password reset page
|— changePassword.js            # JS logic for password reset
|
|— content_script.js            # Injected script for ChatGPT
integration
|— content_script_claude.js      # Content script for Claude AI
|— content_script_injectable.js  # Shared inject logic for multiple
platforms
|
|— firebase-app.js              # Firebase configuration
|— firebase-firestore.js        # Firestore database operations
|
|— fontsgoogleapis.css          # Google Fonts import
|— gl-loader.png                # Loader/spinner asset
|

```

— inject.js	# Handles DOM injection and prompt
insertion	
— jquery.min.js	# jQuery library
— listing.html	# Prompts listing UI
— listing.js	# JS logic for listing and selecting
prompts	
— logo64x64.png	
— logo128x128.png	
— manifest.json	# Chrome extension configuration
(Manifest V3)	
— popup.html	# Popup interface shown when icon is
clicked	
— popup.js	# Logic for popup actions
— popup.css	# Popup styling
— save.js	# Handles saving new prompts or edits
— select2.min.css	# External CSS for dropdowns
— select2.min.js	# External JS for enhanced selects
— style.css	# General styles used across pages
— thankyou.html	# Confirmation page after signup or save

Manifest Configuration (Manifest V3)

manifest.json

```
{
  "action": {
    "default_title": "Open ChatGPT",
    "default_popup": "popup.html"
```

```
},
"background": {
  "service_worker": "background.js",
  "type": "module"
},
"options_page": "thankyou.html",
"content_scripts": [
  {
    "css": [ "style.css" ],
    "js": [ "content_script.js" ],
    "matches": [ "https://chatgpt.com/*" ],
    "run_at": "document_end"
  },
  {
    "js": [ "firebase-app.js", "firebase-firestore.js",
"content_script_injectable.js" ],
    "matches": [ "https://chatgpt.com/*" ],
    "run_at": "document_end"
  },
  {
    "js": [ "content_script_claude.js" ],
    "matches": [ "https://claude.ai/*" ],
    "run_at": "document_idle"
  }
],
"description": "Save and share fantastic AI prompts.",
"manifest_version": 3,
"name": "Atomic Gains: Prompt Assistant",
"permissions": [ "tabs", "activeTab", "identity", "identity.email",
"storage" ],
"version": "1.1.9",
"web_accessible_resources": [ {
  "matches": [ "https://chatgpt.com/*", "https://claude.ai/*" ],
```

```
    "resources": [ "config.js", "inject.js", "inject2.js",  
"listing.html", "againslogofavicon32.png", "againslogofavicon64.png"  
],  
    "use_dynamic_url": true  
  } ],  
  "icons": {  
    "512": "againslogofavicon512.png",  
    "256": "againslogofavicon256.png",  
    "128": "againslogofavicon128.png",  
    "64": "againslogofavicon64.png",  
    "32": "againslogofavicon32.png"  
  },  
  "host_permissions": [  
    "https://www.gstatic.com/"  
  ]  
}
```

Installation and Setup

1. Clone or Download

```
git clone  
https://github.com/unicapublications/atomic-gains-extension.git  
cd atomic-gains-extension
```

2. Load the Extension

1. Open **Chrome** → navigate to `chrome://extensions/`
 2. Enable **Developer Mode**
 3. Click **Load unpacked**
 4. Select the `atomic-gains-extension` folder
-

Key Modules and Functions

1. popup.js

Handles UI interactions.

2. background.js

Manages long-lived service tasks and storage sync.

3. content.js

Injects UI buttons or indicators into ChatGPT.

4. manifest.json.

The file that will make the Chrome extension working.

Backend (PHP + MySQL)

Database Schema (MySQL)

The backend of **Atomic Gains: Prompt Assistant** is powered by a MySQL database connected via PHP APIs.

It contains **15 tables** that manage users, prompts, activities, groups, shares, and reports.

1. activities

Column	Type	Description
id	bigint (PK, AI)	Unique activity ID
activity	varchar(100)	Description of the activity
topic	varchar(100)	Related topic

2. claudechats

Column	Type	Description
id	bigint (PK, AI)	Record ID

chatid	varchar(500)	Claude chat session ID
hint	varchar(2500)	System or user hint text

3. filteredwords

Column	Type	Description
id	int (PK, AI)	Record ID
createdby	varchar(100)	Email of the creator
word	varchar(100)	Blocked or restricted word

4. groupmembers

Column	Type	Description
id	bigint (PK, AI)	Record ID
groupname	varchar(200)	Associated group name
groupmemberemail	varchar(100)	Email of member

5. groups

Column	Type	Description
id	bigint (PK, AI)	Group ID
groupname	varchar(500)	Group name
createdby	varchar(100)	Email of group creator

6. groupshare

Column	Type	Description
--------	------	-------------

id	bigint (PK, AI)	Record ID
groupid	bigint	Linked group ID
templateid	bigint	Shared template ID

7. individualshare

Column	Type	Description
id	bigint (PK, AI)	Record ID
createdby	varchar(50)	Creator email
templateid	bigint	Template shared
useremail	varchar(50)	Recipient email

8. newslettersubscribers

Column	Type	Description
id	bigint (PK, AI)	Subscriber ID
email	varchar(100)	Subscriber email address

9. promptactions

Column	Type	Description
id	bigint (PK, AI)	Record ID
action	varchar(100)	Action performed (like, copy, etc.)
templateid	bigint	Associated prompt/template
useremail	varchar(100)	User performing the action

10. promptdeletes

Column	Type	Description
id	bigint (PK, AI)	Record ID
datecreated	timestamp	Date of deletion
fromuseremail	varchar(100)	User who deleted
templateid	bigint	Related template ID

11. promptdetailhider

Column	Type	Description
id	int (PK, AI)	Record ID
email	varchar(100)	User email
menuid	bigint	Menu or section ID
prompttoshow	text	Prompts to be displayed

12. promptreports

Column	Type	Description
id	int (PK, AI)	Report ID
datecreated	timestamp	When report was filed
reason	varchar(500)	Reason for report
templateid	bigint	Reported prompt
useremail	varchar(50)	Reporter's email

13. prompts

Column	Type	Description
id	bigint (PK, AI)	Prompt ID

authoremail	varchar(100)	Author's email
authorname	varchar(100)	Author's name
authorurl	varchar(100)	Author's profile URL
category	varchar(100)	Prompt category
community	varchar(100)	Community name
datepublished	timestamp	Publish date
prompt	longtext	Main prompt text
prompthint	varchar(1000)	Hint or context
publicity	varchar(100)	Public/Private visibility
rejects	int	Rejection count
teaser	varchar(1000)	Short summary
title	varchar(2000)	Prompt title
usages	int	Usage count
views	int	View count
votes	int	Vote count
instruction	varchar(1000)	Instructions for usage
instructiontype	varchar(50)	Instruction format
links	varchar(100)	Reference links
type	varchar(100)	Type of prompt (e.g., Claude, ChatGPT)

14. topics

Column	Type	Description
id	bigint (PK, AI)	Topic ID
topic	varchar(100)	Topic name

15. users

Column	Type	Description
id	bigint (PK, AI)	User ID
email	varchar(100)	User email
fullname	varchar(500)	Full name
isadmin	int	1 = admin, 0 = normal user
ispaused	int	1 = paused, 0 = active
password	varchar(100)	Hashed password

Example PHP Endpoint

GET /prompts

```
<?php
require 'config.php';
header('Content-Type: application/json');
$query = $pdo->query("SELECT * FROM prompts");
echo json_encode($query->fetchAll(PDO::FETCH_ASSOC));
?>
```

Deployment Guide

1. Build the Extension

Zip the folder:

```
zip -r atomic-gains.zip atomic-gains-extension/
```

2. Publish to Chrome Web Store

- Go to Chrome Developer Dashboard
- Upload `atomic-gains.zip`
- Set version, description, and screenshots
- Submit for review

3. Deploy Backend

- Upload PHP API to your production server
- Import MySQL schema via phpMyAdmin or command line