



# Smart Bookmark Saver –

A Chrome Extension

# The Problem We Are Facing



## ○ Inflexible Bookmark Organization

Saving bookmarks in browsers is often time-consuming and hard as Chrome bookmarks are limited to a folder structure, which makes saving related links across different topics cumbersome.

## ○ Forgotten Links

Many people forget to properly organize or categorise their saved links, making retrieval impossible later on.

## ○ Feature Gap

Traditional bookmarking lacks the quick access, tagging, and custom features required for modern productivity.

## ○ The Core Need

We desperately need a faster, more personalised way to save and retrieve bookmarks directly while browsing.

# What Exactly Is a Browser Extension?



## Small Software Program

At its core, it is a small software program that enhances the browser's functionality.



## Standard Web Stack

Extensions are typically built using standard web technologies: HTML for structure, CSS for styling, and JavaScript for dynamic functionality.



## Interactive Capabilities

It can interact with web pages, modify browser behaviour, or add new tools.



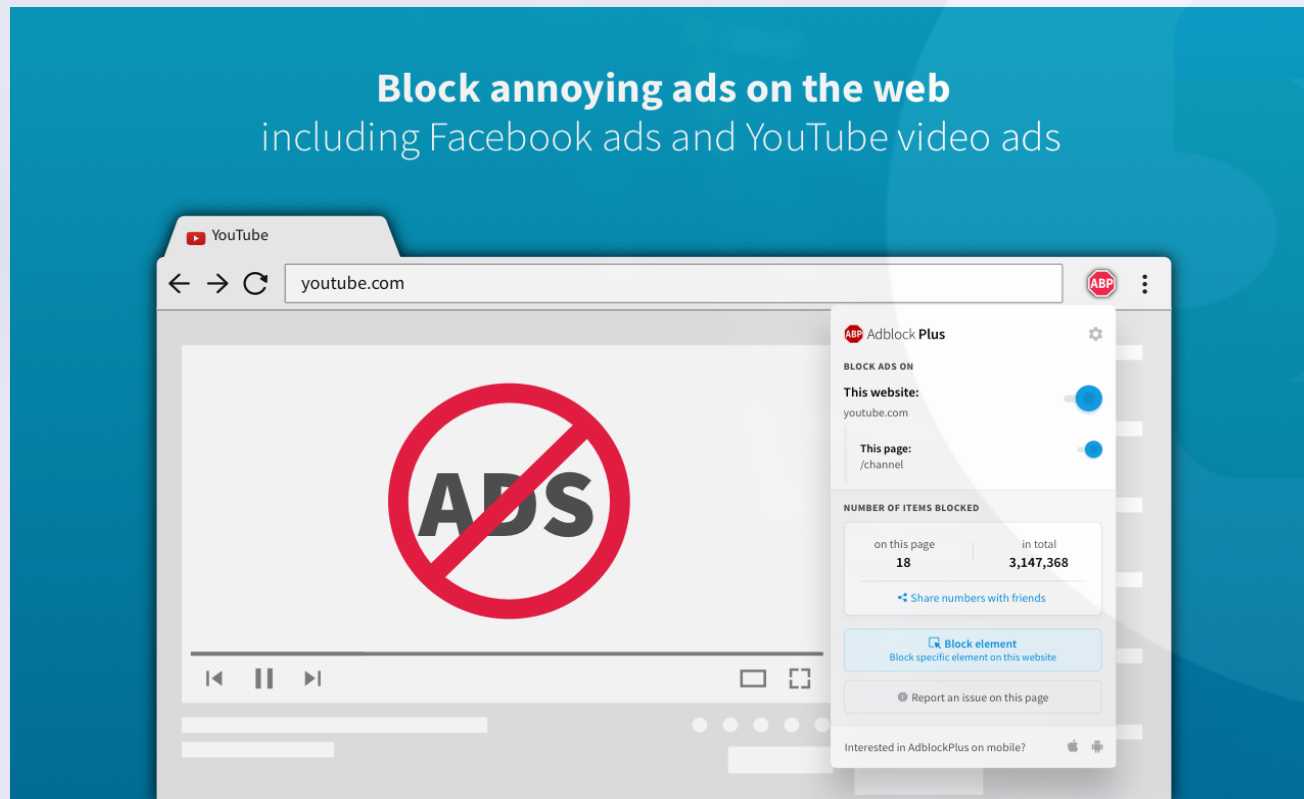
## Seamless Integration

Once installed, it appears as a small icon in the browser toolbar or is integrated discreetly into the browser's context menus.

# Real-World Use Cases of Browser Extensions



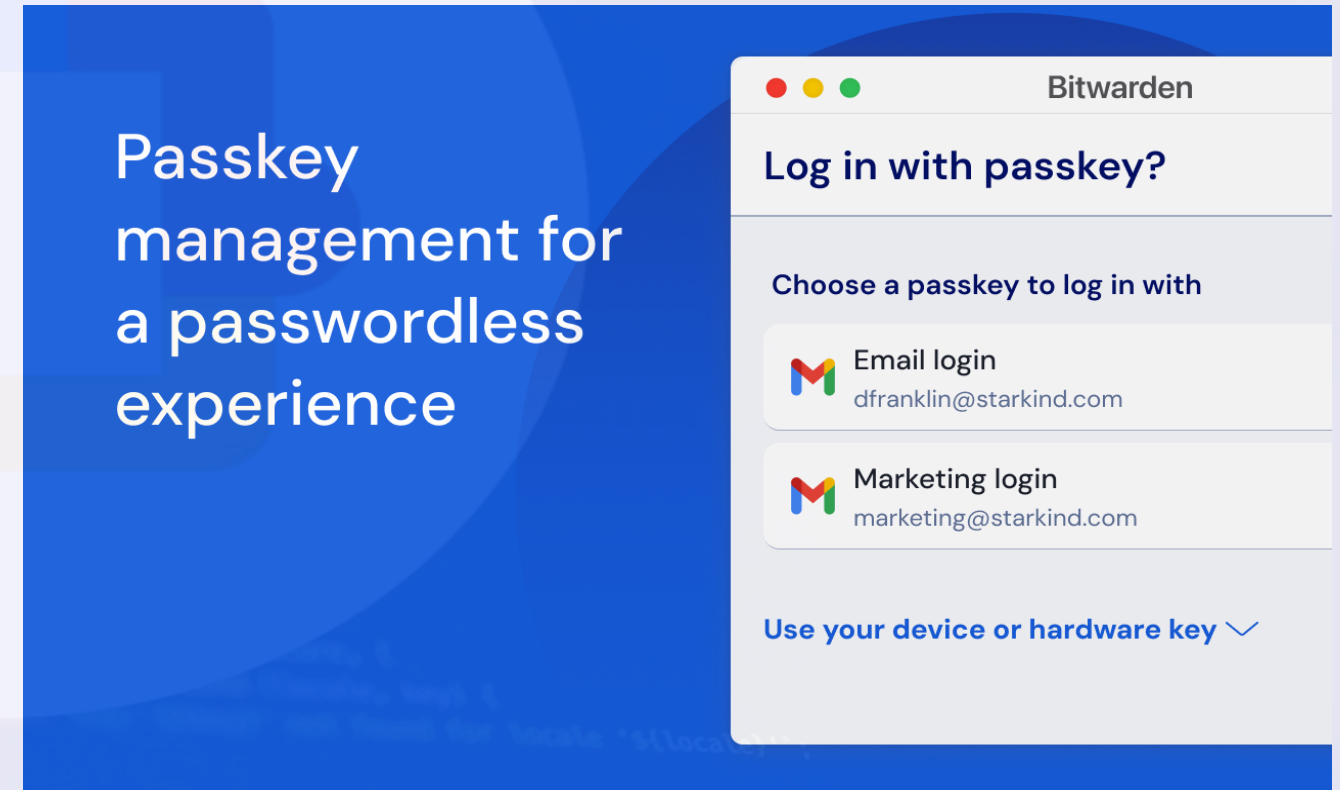
## Ad Blockers



Block unwanted advertisements, resulting in significantly improved browsing speeds and a cleaner user experience.



## Password Managers



Securely store, manage, and auto-fill login credentials across different sites, bolstering personal security.



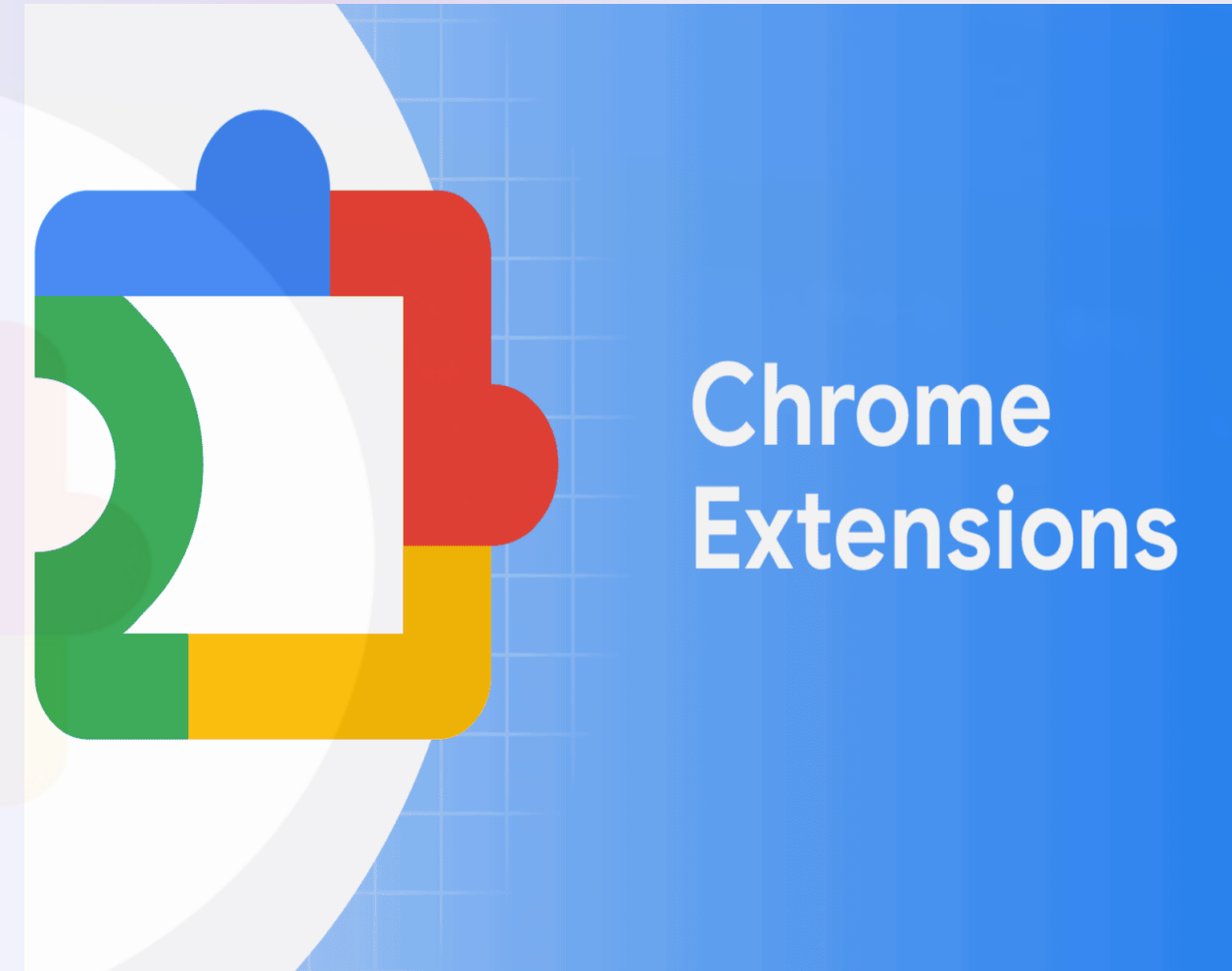
# Why it's useful ? 🧠

## → Browser Extensions:

- Make browsing faster and easier.
- Add helpful tools inside the browser.
- Save time by reducing extra steps or apps.

## → Our Smart Extension

- Keeps all your favorite pages organized in one place.
- Works across categories (shopping, makeup, study, etc.).
- Easy to find, tag, and revisit saved sites.
- Makes online work and shopping more efficient.

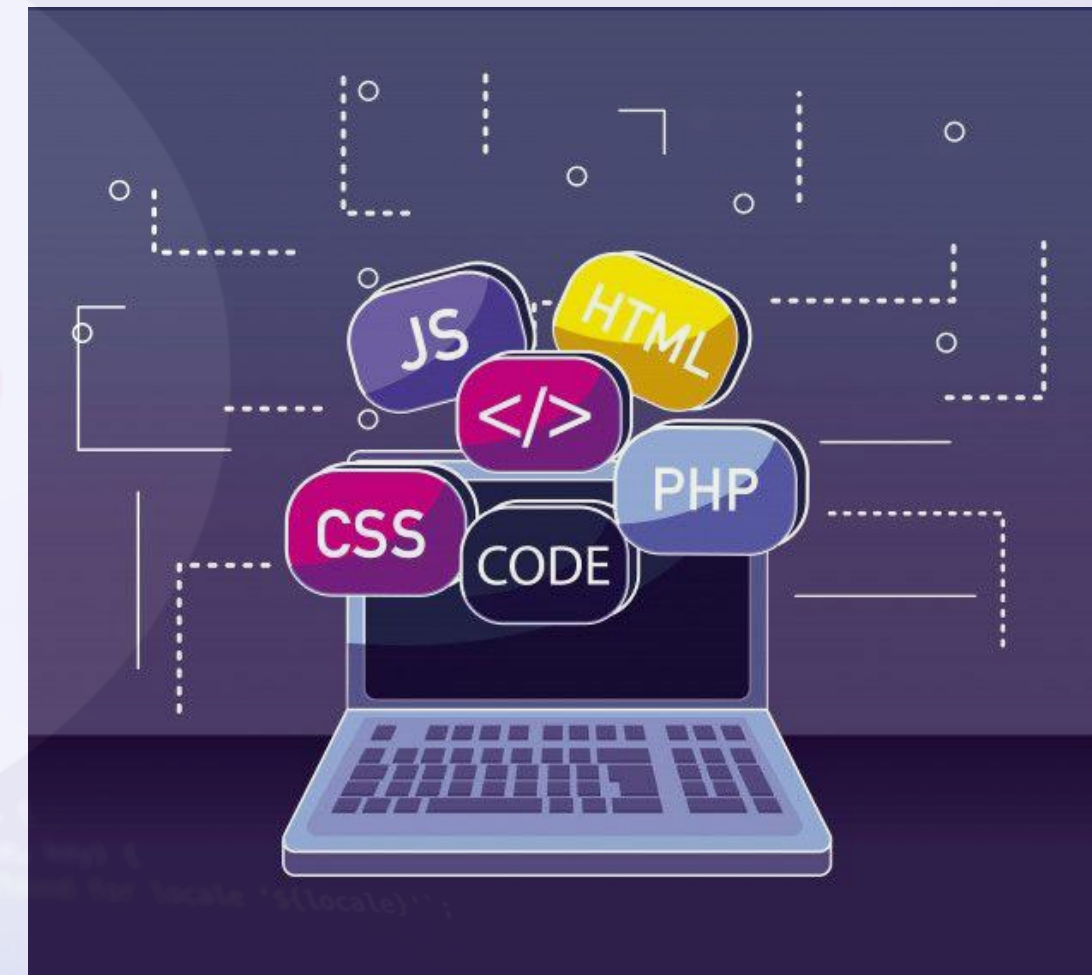


# Traditional Browser Extension vs. Smart Browser Extension :

Feature	Traditional Browser Extensions	Our Smart Extension
Access Speed	Slower	Faster
Simplicity	<b>More Steps</b>  Requires creating and organizing folders.	<b>Fewer Steps</b>  Simply select a tag.
Flexibility	<b>Limited</b>  A bookmark can only be in one folder at a time.	<b>Flexible</b>  A single bookmark can have multiple tags, allowing for dynamic organization.
Storage philosophy	<b>Distributed</b>	<b>Centralized</b>

# Technologies Used

- HTML 5 → Provides the structure and content for the extension's user interface (links the necessary styling (popup.css) and logic (popup.js) files).
- CSS3 → Styles the HTML elements, controlling the layout, colors, and visual appearance of the popup (for styling and layout, specifies dropdown menus and tags)
- JavaScript → Handles the logic, interactivity, and communication with the browser's extension system (implements all the extension's functionality ).
- Chrome Extension APIs → Specific JavaScript methods provided by Chrome for interacting with the browser (extension interacts with the browser's core features).



# Step-By-Step Development Process

## 1. Setup & Structure (Manifest and HTML)

Initial step focuses on defining the extension's identity and basic user interface.

- **Create the Manifest:**
  - Define the extension's metadata.
  - Declare required permissions
  - Set the entry point for the popup UI using the action key pointing to popup.html.
- **Create the HTML Structure:**
  - Layout the basic elements for the popup: title, dropdown for tag selection the Save Button the filter dropdown, and the bookmark list container.
  - Link the CSS and JavaScript files.

## 2. Core Saving Logic (JavaScript and Chrome APIs)

This step implements the primary function of the extension: capturing and saving the current tab's data.

- **Capture Tab Data:** In popup.js, use `chrome.tabs.query` to get the URL and title of the currently active tab when the `saveBtn` is clicked.
- **Implement Extension Local Storage Save:**
  - Use `chrome.storage.local.get()` to retrieve the existing array of bookmarks.
  - Prevent Duplicates
  - Create a new bookmark object with the URL, title, date, and selected tag and save it to array using `chrome.storage.local.set()`
- **Implement Native Chrome Bookmark Save (Optional/Feature):**
  - Use `chrome.bookmarks.search()` to find or create a dedicated "Smart Bookmarks" folder.
  - Use `chrome.bookmarks.create()` to add the tab's URL/title to that folder.



# Step-By-Step Development Process Cont.

## 3. Display and Management (JavaScript Functions)

This step focuses on showing the saved bookmarks and adding interactivity (filtering and deleting).

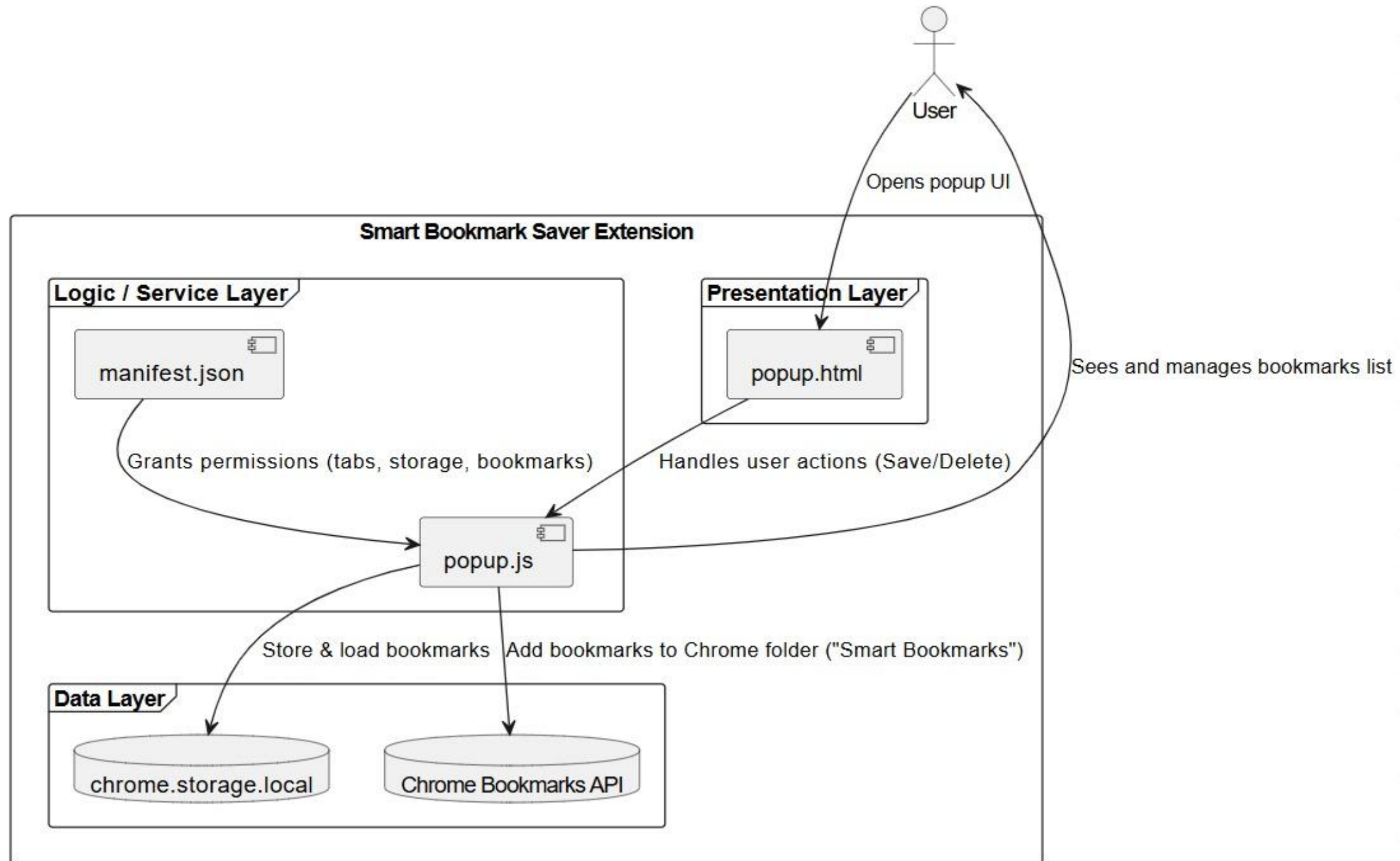
- **Display Function (showBookmarks):**
  - Read the bookmarks array from `chrome.storage.local`.
  - **Handle Filtering:** Check if a `filterTag` is supplied and filter the bookmarks array accordingly.
- **Implement Filtering:** Attach an event listener to the `filterSelect` dropdown to call `showBookmarks()` with the new filter value whenever the selection changes.
- **Implement Deletion:**
  - Use `chrome.storage.local.get()` and `filter()` to create a new array without the item to be deleted.
  - Save the updated array and re-render the list by calling `showBookmarks()`.

## 4. Refinement and Styling (CSS)

The **final** step is to apply styling and polish the user experience.

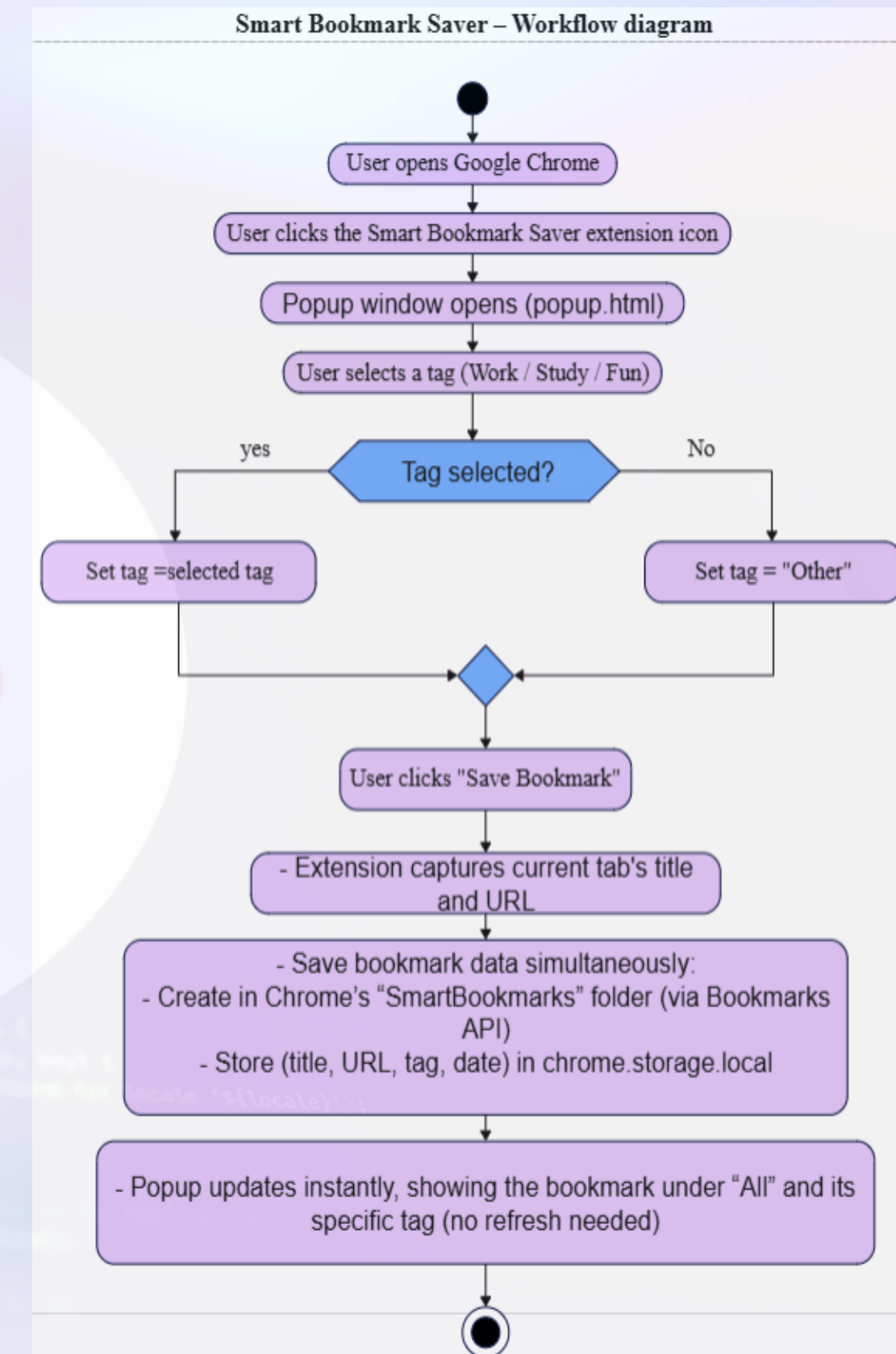
- **General Styling (popup.css):** Apply a consistent font, background, width, and padding to the body.
- **Component Styling:** Style the major elements like the save button, dropdowns, and the list items for readability and visual appeal.
- **Bookmark Card Layout:** Use **Flexbox** on `.bookmark-card` to align the favicon, info, tag, and delete button neatly.
- **Tag Coloring:** Define specific color rules for different tags to enhance categorization.

# Smart Bookmark Saver - Layered Architecture Diagram



# A Three-Step Workflow for Clarity

We have simplified the process of saving and retrieving links into workflow Diagram , seamless steps, putting speed and user experience first.



# Future Enhancements



## Adding Comment or Notes

Allowing users to write short comment or note when saving bookmark. This could be a reminder, a personal thought ,or a reason for saving the link.



## Highlight Key Content

Allowing users to highlight important sentences or sections from the webpage they're saving. These highlights will be stored along with the bookmark.



## AI Auto-Categorization And generate DOC

Use AI to analyze highlights and notes from multiple bookmarks that are related to the same topic, then generate a structured summary or article. The result can be exported as a Word or PDF file.



# Configuration File:

```
{ } manifest.json
1
2  {
3    "manifest_version": 3,
4    "name": "Smart Bookmark Saver",
5    "version": "1.0",
6    "description": "Save and manage bookmarks easily.",
7    "permissions": ["storage", "tabs", "bookmarks"],
8    "action": {
9      "default_popup": "popup.html"
10   }
11 }
12
```

# html code:

```
<body>
  <h2>Smart Bookmarks</h2>

  <!-- Tag selection before saving -->
  <select id="tagSelect">
    <option value="Work"> Work</option>
    <option value="Study"> Study</option>
    <option value="Shopping"> Shopping</option>
    <option value="Entertainment"> Entertainment</option>
    <option value="Other" selected> Other</option>
  </select>

  <button id="saveBtn">Save Current Tab</button>

  <hr />

  <!-- Filter section -->
  <label for="filterSelect"><strong>Filter by Tag:</strong></label>
  <select id="filterSelect">
    <option value="All">All</option>
    <option value="Work">Work</option>
    <option value="Study">Study</option>
    <option value="Shopping">Shopping</option>
    <option value="Entertainment">Entertainment</option>
    <option value="Other">Other</option>
  </select>

  <ul id="bookmarkList"></ul>

  <script src="popup.js"></script>
</body>
</html>
```

# JS Code:

```
// Save current tab to extension + Chrome bookmarks
saveBtn.addEventListener("click", () => {
  const tag = tagSelect.value;

  chrome.tabs.query({ active: true, currentWindow: true }, (tabs) => {
    const tab = tabs[0];
    const newBookmark = {
      url: tab.url,
      title: tab.title,
      date: new Date().toLocaleDateString(),
      tag: tag
    };

    // 1 Save to local storage
    chrome.storage.local.get({ bookmarks: [] }, (data) => {
      const updated = data.bookmarks;
      if (!updated.some(b => b.url === tab.url)) {
        updated.push(newBookmark);
        chrome.storage.local.set({ bookmarks: updated }, showBookmarks);
      }
    });

    // 2 Save to Chrome's native bookmarks
    const folderName = "Smart Bookmarks";

    chrome.bookmarks.search({ title: folderName }, (results) => {
      if (results.length > 0) {
        chrome.bookmarks.create({
          parentId: results[0].id,
          title: tab.title,
          url: tab.url
        });
      } else {
        chrome.bookmarks.create({ title: folderName }, (folder) => {
```

# Team Members:

- Basant Hatem 20226026
- Habiba Bassel 20226034
- Halla Ayman 20226036
- Mariam Sherif 20226100
- Rahaf Yasser 20226138

## Repo Link:

- [https://github.com/HabibaBassel34/Smart\\_Bookmark](https://github.com/HabibaBassel34/Smart_Bookmark)





Thank You!!