# Declaration of Academic Integrity

Academic Integrity is a central tenet of Singapore Polytechnic. The polytechnic rules state that "Cheating in examinations and other assessed work is a very serious offence. This includes copying and using plagiarised material. Any student who cheats, attempts to cheat or breaches any rules for examinations and tests will face disciplinary action. The student is liable to be expelled."

Check **only one** of the two options below:

I affirm that the work I submit is my own, produced without help from any AI tool(s) and/or other source(s).

I affirm that the work I submit has been produced with the use of AI tool(s) and/or other source(s) which I have acknowledged fully in the [following section](#_Acknowledgement_of_use).

By signing this form, I declare that the above affirmation made is true, and that I have read and understood the rules stated in Students Handbook on “[Plagiarism](https://www.sp.edu.sg/sp/student-services/osc-overview/student-handbook/intellectual-property-copyright-and-plagiarism)” and “[Breach of Examination/Assessment Rules](https://www.sp.edu.sg/sp/student-services/osc-overview/student-handbook/conduct-in-examinations-breach-of-exam-rules)”.

|  |  |  |
| --- | --- | --- |
| Name:  Jerome Loke | Student ID:  2510707 | Class:  DAAA/FT/1A/01 |
| Signature: | Date:  11/8/2025 | |

## Acknowledgement of use of AI tool(s) and/or other source(s) (where applicable)

### AI tool(s)

|  |  |
| --- | --- |
| **Name of AI tool** | Claude |
| **Input prompt** | teach me step by step how to implement localstorage to prevent saved articles from bein glost on page refresh |
| **Date generated** | 31 July 2025 |
| **Output generated** | I'll teach you how to implement localStorage to persist your saved articles across page refreshes. Here's a step-by-step guide:  localStorage Implementation for Saved Articles  Code  /\* ============================================================================ STEP-BY-STEP LOCALSTORAGE IMPLEMENTATION FOR SAVED ARTICLES ============================================================================ \*/ /\* ---------------- STE  Now let me explain the key concepts and implementation steps:  **Step 1: Understanding localStorage**  localStorage is a web browser feature that allows you to store data persistently on the user's device. Unlike regular variables, localStorage data survives page refreshes and browser restarts.  **Key characteristics:**   * Stores data as strings only * Data persists until manually cleared * Has ~5-10MB storage limit per domain * Synchronous API (blocking)   **Step 2: Helper Functions**  The three helper functions I created handle the basic localStorage operations with proper error handling:   * saveToLocalStorage() - Converts data to JSON and stores it * loadFromLocalStorage() - Retrieves and parses stored data * removeFromLocalStorage() - Removes stored data   **Step 3: Smart Data Storage Strategy**  Instead of storing entire article objects, we store only the IDs of saved articles. This approach:   * Reduces storage space * Prevents data inconsistencies * Makes updates more efficient   **Step 4: Initialization Process**  The initializeFromLocalStorage() function:   1. Loads saved article IDs from localStorage 2. Updates the saved property in your allArticles array 3. Rebuilds the savedArticles array 4. Optionally restores page and filter state   **Step 5: Integration Points**  You need to call saveToLocalStorage() whenever saved articles change:   * In saveArticle() - when toggling save status * In removeSavedArticle() - when removing individual articles * In clearAllSaved() - when clearing all saved articles   **Step 6: Page Load Integration**  Add the DOMContentLoaded event listener to initialize everything when the page loads.  **Quick Implementation Checklist:**   1. ✅ Add the helper functions to your code 2. ✅ Add the initialization function 3. ✅ Update your existing save/remove functions 4. ✅ Add the DOMContentLoaded event listener 5. ✅ Test by saving articles and refreshing the page   The beauty of this implementation is that it's backward-compatible - if no data exists in localStorage, it falls back to your current behavior. Once implemented, your users' saved articles will persist across sessions! |
| **Impact on submission** | This helped me prevent the user’s bookmarked articles from being lost on page refresh. |

### Other source(s)

|  |  |
| --- | --- |
| **Source** | **Impact on submission** |
| < *Link to online source* > | < *Briefly explain which part of your submitted work the online source was applied* > |
| < *Author, Book / Article / Journal, Year, Pages* > | < *Briefly explain which part of your submitted work the Book / Article / Journal was applied* > |
| < *Student / Person name* > | < *Briefly explain which part of your submitted work the help of another student / person was applied* > |
| < *Others* > | < *Briefly explain which part of your submitted work the source was applied* > |