Final Project Report
SI579 – Winter 2025
Zichen Jiang jiangzc@umich.edu

Interactive Game Rating Recorder

Author: Zichen Jiang (GitHub: mtjqmszb)
GitHub Repo: https://github.com/mtjqmszb/579-final-project
Live App: https://mtjqmszb.github.io/579-final-project/

🧾 Project Summary

This project is a simple web app that lets users log and manage their personal game ratings. Each entry includes the game name, rating (0–10), date started, and optional review notes. Users can sort their entries by name, date, or rating. The data is saved using localStorage, so it persists across sessions.

The app is built entirely using **Vanilla JavaScript**, with all DOM updates handled via innerHTML and template literals (no createElement() used), per SI 579 best practices.

✅ What Was Achieved

- Fully functioning game entry form with validation
- Add, delete, and edit game entries
- Sort entries by name, rating, or date
- Data persistence using localStorage
- Clean and readable DOM rendering using template strings
- Deployment via GitHub Pages

🔸 What Was Not Achieved

- No login or user authentication
- No database or server-side persistence
- No advanced styling (minimal layout only)

💬 Challenges

- Debugging sorting logic (rating sort wasn’t working due to a typo)
- Removing leftover restrictions from Problem Set 4 (game names with spaces were initially blocked)
- Remembering to save to localStorage after every change
- Refactoring to remove createElement() and use innerHTML instead
- Building edit functionality in a clean, non-redundant way

📈 Possible Improvements & Scalability

- Add a “search” or filter feature
- Group entries by genre or platform
- Implement user login + cloud storage with Firebase or Supabase
- Export/download saved list as CSV or JSON
- Add visual styling or themes using a CSS framework like Tailwind

---

📂 Where to Look in the Code

This is a Vanilla JS project. All logic is in the following files:

- index.html: sets up the structure of the page and includes the form and game list container
- script.js: contains **all interactivity**, including:
  - Form handling and validation
  - Sorting logic
  - Rendering the game list using template literals
  - Add/Edit/Delete functionality
  - LocalStorage management

No external libraries or build tools are used. The project is self-contained.

📌 Note for Submission:

This report is also available as the README.md in the project GitHub repository.
GitHub repo: https://github.com/mtjqmszb/579-final-project