ITM SKILLS UNIVERSITY

School of Future Tech

Case Study Report

Title : Code Snippet Library – Web-Based Code Management System

(Group Number 7)

Group Members:

Parth Pathari - 150096725125
Ashutosh Mishra - 150096725129
Atharv Ghadge - 150096725131
Piyush Owalekar - 150096725114

Index

---

1. Introduction to the Case Study

Modern developers frequently reuse code snippets such as utility functions, components, and API logic. However, managing these snippets using text files, notes, or scattered documents becomes inefficient and error-prone over time.

This case study focuses on the design and implementation of a **Code Snippet Library**, a lightweight web-based application that allows users to **store, search, manage, and reuse code snippets** efficiently. The system is built using **HTML, JavaScript, and Browser Local Storage**, making it fast, offline-capable, and easy to use.

The application provides essential features such as snippet creation, tagging, searching with debounce optimization, copying code to clipboard, and persistent storage.

---

2. Problem Statement

Background

Developers often struggle to organize reusable code snippets across different programming languages. Existing solutions may require authentication, cloud storage, or complex setups, which are not ideal for beginners or offline usage.

Abstract

This case study presents a **client-side Code Snippet Library** that enables users to store and manage code snippets directly in the browser using **Local Storage**. The system supports multiple programming languages, tagging, debounced search functionality, and clipboard copying. It demonstrates practical usage of **DOM manipulation, event handling, data persistence, and optimization techniques** in JavaScript.

---

3. Case Study Design

The Code Snippet Library is designed as a **single-page web application** with the following components:

1. User Interface

- Input fields for title, language, description, code, and tags
- Buttons for add, save, cancel, copy, delete, and clear all
- Dynamic rendering of stored snippets

2. Data Management

- Snippets stored as JavaScript objects
- Persistent storage using `localStorage`

3. Search & Filtering

- Real-time search using keywords
- Debounce technique to improve performance

4. Security & Safety

- HTML escaping to prevent script injection

---

4. Methods & Technologies Applied

Technologies Used

- **HTML5** – Structure of the application
- **JavaScript (ES6)** – Logic, interactivity, and data handling
- **Local Storage API** – Persistent client-side storage

Key Methods Implemented

1. **CRUD Operations**
    - Create: Add new code snippets
    - Read: Display stored snippets
    - Update: Re-render snippets dynamically
    - Delete: Remove individual or all snippets
2. **Debounce Algorithm**
    - Reduces unnecessary function calls during search input
3. **Clipboard API**
    - Enables one-click copy of code snippets
4. **Data Sanitization**
    - Escaping user input to prevent XSS attacks

---

5. Case Study Implementation Details and Snapshots

Implementation Details

- Snippets are stored as objects with properties:
  `id, title, language, description, code, tags`
- Data is saved and retrieved using `localStorage`
- Search functionality checks title, description, language, code, and tags
- Default snippets are loaded on first use

Core Functional Modules

- `render()` – Displays snippets dynamically
- `save()` – Saves new snippets
- `search()` – Filters snippets
- `copy()` – Copies code to clipboard
- `del()` – Deletes a snippet
- `clear()` – Clears all stored snippets

---

6. Results and Conclusion

Results

- The application successfully stores and retrieves code snippets
- Search functionality is fast and efficient due to debouncing
- Snippets persist even after browser refresh
- User-friendly interface suitable for beginners

Conclusion

This case study demonstrates how a **simple yet powerful code management system** can be built using only frontend technologies. The Code Snippet Library improves productivity by organizing reusable code efficiently without external dependencies. It also strengthens understanding of **JavaScript fundamentals, browser storage, and UI rendering**, making it an excellent real-world learning project.

---

7. References

- MDN Web Docs – JavaScript & Local Storage
- MDN Web Docs – Clipboard API
- HTML5 Specification
- JavaScript ES6 Documentation
- Web Development Best Practices