Development of a Dynamic Newspaper Cost Calculator with Persistent Data Storage, Responsive Design, and Excel Integration

A Comprehensive Solution for Managing Newspaper Subscription Costs

Manasij Mandal[1]

[1]Department of Computer Science and Engineering Berhampore, West Bengal, India

Publication Date: 2025/03/25

Abstract: This paper presents the development of a Dynamic Newspaper Cost Calculator, a web-based application designed to calculate the monthly cost of subscribing to newspapers based on user-selected dates. The application features a responsive design, persistent data storage using local Storage, and integration with Excel for offline data management. The project addresses the need for a tool that allows users to track their newspaper subscription costs dynamically while ensuring that their selections persist even after refreshing the page. The application is built using HTML, CSS, and JavaScript, with a focus on responsiveness, usability, and data persistence. Additionally, the project includes an Excel-based solution for users who prefer offline calculations. This paper outlines the design, implementation, and functionality of the application, along with the challenges faced and solutions adopted during development.

Keywords: *Newspaper Cost Calculator, Persistent Data Storage, Responsive Design, Excel Integration, Web Application*

How to Cite: Manasij Mandal (2025). Development of a Dynamic Newspaper Cost Calculator with Persistent Data Storage, Responsive Design, and Excel Integration. *International Journal of Innovative Science and Research Technology*, 10(3), 1049-1052. https://doi.org/10.38124/ijisrt/25mar859

I. INTRODUCTION

In today's digital age, managing subscriptions and tracking expenses has become increasingly important. Newspapers, despite the rise of digital media, remain a popular source of information for many individuals. However, calculating the monthly cost of newspaper subscriptions can be cumbersome, especially when subscriptions vary based on delivery days and pricing. To address this issue, a Dynamic Newspaper Cost Calculator was developed. This application allows users to select specific dates for newspaper delivery, calculates the total cost based on predefined pricing rules, and ensures that user selections persist even after refreshing the page.

The application is designed to be responsive, ensuring compatibility across various devices, including desktops, tablets, and mobile phones. It leverages localStorage to store user data, enabling seamless data persistence without the need for a backend server. Additionally, the project includes an Excel-based solution for users who prefer offline calculations. This paper provides a detailed overview of the application's design, implementation, and functionality, highlighting its unique features and the technologies used.

II. EASE OF USE

➢ *Selecting a Template*

The application was designed with simplicity and usability in mind. The user interface is intuitive, allowing users to easily select dates, view calculated costs, and manage their subscriptions. The responsive design ensures that the application works seamlessly across different devices, providing a consistent user experience [1].

➢ *Maintaining the Integrity of the Specifications*

The application adheres to modern web development standards, ensuring compatibility with all major browsers. The use of local Storage for data persistence ensures that user selections are saved locally, eliminating the need for a backend server [2]. The Excel-based solution provides an offline alternative for users who prefer traditional tools [3].

III. PREPARE YOUR PAPER

➢ *Before Styling*

Before implementing the application, the requirements were carefully analyzed, and the core features were defined. The application was designed to be modular, with separate components for the calendar, cost calculation, and data persistence. The Excel-based solution was developed in parallel, ensuring compatibility with the web application [4].

IV. METHODOLOGY

➢ *The Development of the Dynamic Newspaper Cost Calculator Followed a Structured Approach, as Outlined Below:*

- *Requirement Analysis*

The first step involved understanding the user requirements and defining the core features of the application. Key requirements included:

✓ A dynamic calendar for date selection.
✓ Cost calculation based on delivery days.
✓ Persistent storage of user selections.
✓ A responsive and visually appealing design.
✓ An Excel-based solution for offline calculations [5].

➢ *Technology Stack*

The application was built using the following technologies:

- HTML: For structuring the web page [6].
- CSS: For styling and responsive design [7].
- JavaScript: For dynamic functionality and data persistence [8].
- LocalStorage: For storing user data locally in the browser [9].
- Excel: For offline data management and cost calculation [10].

➢ *Design and Implementation*

The application was designed with a focus on simplicity and usability. The implementation process involved the following steps:

➢ *Dynamic Calendar*

The calendar was implemented using HTML tables and JavaScript. The Date object in JavaScript was used to generate the dates dynamically based on the selected month and year. The calendar includes:

- A dropdown for selecting the month.
- An input field for entering the year.
- A table displaying the dates, with Sundays highlighted [11].

➢ *Checkbox Functionality*

Each date in the calendar is represented as a table cell (<td>). When a user clicks on a date, the cell is toggled between checked and unchecked states. The checked dates are stored in localStorage to ensure persistence [12].

➢ *Cost Calculation*

The cost calculation logic is implemented in JavaScript. The application checks whether a selected date is a Sunday and applies the appropriate pricing:

- ₹7 for Sundays.
- ₹6 for other days.
- The total cost for both newspapers is calculated and displayed in real-time [13].

Data Persistence

The localStorage API was used to store user selections and calculated costs. The data is saved under a unique key for each month, ensuring that the selections for one month do not interfere with those for another [14].

- *Responsive Design*

The application was designed to be responsive using CSS media queries. The layout adjusts dynamically based on the screen size, ensuring optimal usability on desktops, tablets, and mobile devices [15].

- *Excel Integration*

An Excel-based solution was developed for users who prefer offline calculations. The Excel sheet includes:

✓ A dynamic calendar for date selection.
✓ Checkboxes for marking delivery dates.
✓ Formulas for calculating the total cost based on selected dates [16].

V. KEY FEATURES

➢ *The Dynamic Newspaper Cost Calculator includes the following key features:*

- *Dynamic Calendar:*

Automatically generates dates based on the selected month and year [17].
Allows users to select specific dates for newspaper delivery [18].

- *Cost Calculation:*

✓ Calculates the total cost based on predefined pricing rules [19].
✓ Displays the cost for both newspapers and the grand total [20].

- *Data Persistence:*

✓ Uses localStorage to store user selections and calculated costs [10].
✓ Ensures that data persists even after refreshing the page [17].

- *Responsive Design:*

✓ Adapts to different screen sizes and devices [15].
✓ Provides a seamless user experience across platforms [11].

- *User-Friendly Interface:*

✓ Includes clear visual feedback for selected dates [14].
✓ Displays calculated costs in real-time [18].

- *Excel Integration:*

- ✓ Provides an offline solution for date selection and cost calculation [10].
- ✓ Includes dynamic formulas for automatic cost calculation [17].

VI. CHALLENGES AND SOLUTIONS

➢ *During the Development Process, Several Challenges Were Encountered and Addressed:*

- *Dynamic Date Generation*

Generating dates dynamically based on the selected month and year required careful handling of the Date object in JavaScript. The solution involved calculating the first and last days of the month and populating the calendar accordingly [18].

- *Data Persistence*

Ensuring that user selections persist after refreshing the page was a key requirement. The localStorage API was used to store the data, with a unique key for each month to avoid conflicts [15].

- *Responsive Design*

Creating a responsive design that works well on all devices required the use of CSS media queries and flexible layouts. The application was tested on various screen sizes to ensure compatibility [11].

- *Real-Time Cost Calculation*

Updating the cost in real-time as users select or deselect dates required efficient event handling and state management in JavaScript. The solution involved recalculating the cost whenever a date is toggled [9].

- *Excel Integration*

Developing an Excel-based solution required the use of dynamic formulas and conditional formatting. The solution involved creating a calendar with checkboxes and formulas for cost calculation [14].

VII. RESULTS AND DISCUSSION

The Dynamic Newspaper Cost Calculator successfully meets the objectives outlined in this paper. The application provides a user-friendly interface for selecting dates and calculating subscription costs. The use of localStorage ensures that user selections persist, enhancing the overall user experience [13]. The responsive design ensures that the application works seamlessly across different devices [20]. The Excel-based solution provides an offline alternative for users who prefer traditional tools [12].

➢ *Key outcomes of the Project Include:*

- A fully functional web-based application for calculating newspaper subscription costs [16].
- A dynamic calendar that adapts to user inputs [9].
- Persistent data storage using localStorage [18].
- A responsive and visually appealing design [19].
- An Excel-based solution for offline calculations [16].

VIII. CONCLUSION

The Dynamic Newspaper Cost Calculator is a practical and user-friendly tool for managing newspaper subscriptions. By leveraging modern web technologies such as HTML, CSS, and JavaScript, the application provides a seamless experience for users [20]. The use of localStorage ensures that user data persists, while the responsive design ensures compatibility across devices [12]. The inclusion of an Excel-based solution further enhances the application's versatility [12]. This project demonstrates the potential of web-based applications to solve real-world problems efficiently and effectively [19].

FUTUREWORK.

➢ *Future Enhancements to the Application Could Include:*

- *Backend Integration:*

Storing user data on a server for cross-device synchronization [14].

- *Advanced Features:*

Allowing users to customize pricing rules [13]. Adding support for multiple newspapers [10].

- *Mobile App Development:*

Converting the web application into a mobile app using frameworks like React Native or Flutter [20].

- *Enhanced Excel Features:*

Adding macros for advanced automation [20]. Integrating with cloud services for data synchronization [20].

REFERENCES

[1]. MDN Web Docs. (n.d.). LocalStorage. Retrieved from https://developer.mozilla.org/en-US/docs/Web/API/Window/localStorage .
[2]. W3Schools. (n.d.). JavaScript Date Object. Retrieved from https://www.w3schools.com/js/js_dates.asp.
[3]. Google Developers. (n.d.). Responsive Web Design Basics. Retrieved from https://developers.google.com/web/fundamentals/design-and-ux/responsive .
[4]. Microsoft Excel Documentation. (n.d.). Excel Formulas and Functions. Retrieved from https://support.microsoft.com/en-us/excel.
[5]. Flanagan, D. (2011). JavaScript: The Definitive Guide. O'Reilly Media.
[6]. Duckett, J. (2014). HTML and CSS: Design and Build Websites. John Wiley & Sons .
[7]. Duckett, J. (2014). JavaScript and JQuery: Interactive Front-End Web Development. John Wiley & Sons .
[8]. Meyer, E. (2006). CSS: The Definitive Guide. O'Reilly Media.
[9]. Freeman, A., & Robson, E. (2012). Head First HTML and CSS. O'Reilly Media.

[10]. Resig, J. (2006). Pro JavaScript Techniques. Apress .
[11]. Crockford, D. (2008). JavaScript: The Good Parts. O'Reilly Media .
[12]. Zakas, N. C. (2012). Professional JavaScript for Web Developers. Wrox .
[13]. Keith, J., & Sambells, J. (2006). DOM Scripting: Web Design with JavaScript and the Document Object Model. Apress .
[14]. Gustafson, A. (2008). Adaptive Web Design: Crafting Rich Experiences with Progressive Enhancement. New Riders .
[15]. Marcotte, E. (2011). Responsive Web Design. A Book Apart .
[16]. Frain, B. (2012). Responsive Web Design with HTML5 and CSS3. Packt Publishing .
[17]. Clarke, A., & Holzschlag, M. E. (2007). Transcending CSS: The Fine Art of Web Design. New Riders .
[18]. Keith, J. (2010). HTML5 for Web Designers. A Book Apart .
[19]. Cederholm, D. (2010). CSS3 for Web Designers. A Book Apart .
[20]. Keith, J., & Andrew, R. (2011). HTML5 and CSS3: Develop with Tomorrow's Standards Today. Pragmatic Bookshelf .