Documentation for Chemical Management Web Application

# Table of Contents

1. Project Overview  
2. Design Approach  
3. Technical Choices  
4. JSON Data Structure  
5. Table Sorting Logic  
6. Row Management Features  
7. Editing Functionality  
8. Responsive Design  
9. Progressive Web App (PWA) Setup  
10. File Structure  
11. Instructions to Open and Run the Application  
12. Hosting and Deployment  
13. Future Enhancements  
14. Conclusion

# 1. Project Overview

This web application helps manage a list of chemicals through an easytouse table interface. Users can sort the table by column, add new rows, delete rows, move rows, refresh the data, save changes, and edit individual rows directly in the table.

# 2. Design Approach

I designed the application to focus on simplicity and usability. Bootstrap is used for styling and layout, ensuring a responsive experience across different devices. Plain JavaScript handles the interactivity, such as sorting, row management, and editing.

# 3. Technical Choices

Bootstrap: Chosen for responsive design and ease of use for table and button styling.  
 HTML & CSS: Used for the structure and basic styling.  
 JavaScript: Used to handle sorting, row management, and editing functionalities.  
 PWA Setup: Implemented to allow offline use and a mobile applike experience.

# 4. JSON Data Structure

The JSON data structure for the chemicals looks like this:  
  
```json  
[  
 {  
 "id": 1,  
 "chemical\_name": "Chemical A",  
 "vendor": "Vendor X",  
 "density": "1.2 g/cm³",  
 "viscosity": "10 mPa·s",  
 "packaging": "Drum",  
 "pack\_size": "200 L",  
 "unit": "Kg",  
 "quantity": 50  
 }  
 // More chemicals can be added here  
]  
```

# 5. Table Sorting Logic

The sorting functionality is handled by attaching event listeners to the column headers. When a column is clicked, the data is sorted using JavaScript's `sort()` function, and the table is rerendered. Here's an example of how it's done:  
  
```javascript  
tableData.sort((a, b) => {  
 if (a[column] < b[column]) return ascending ? -1 : 1;  
 if (a[column] > b[column]) return ascending ? 1 : -1;  
 return 0;  
});  
```

# 6. Row Management Features

There are multiple row management functions:  
 Add Row: Adds a new row to the JSON data and updates the table.  
 Move Row Up/Down: Changes the position of rows by swapping their places in the JSON array.  
 Delete Row: Deletes a row from the JSON array and updates the UI.

# 7. Editing Functionality

The app uses inline editing for ease of use. Clicking on a cell allows the user to edit its value, and the changes are saved when the user clicks outside the cell or presses Enter.

# 8. Responsive Design

The layout is responsive thanks to Bootstrap’s grid system and responsive utilities. I’ve also added media queries to ensure everything looks good on different screen sizes.

# 9. Progressive Web App (PWA) Setup

The app is registered as a PWA so that it works offline and feels like a native mobile app. This involves:  
 manifest.json: Defines the app’s name, icons, and start URL.  
 Service Worker: Caches static assets so the app works even without an internet connection.

# 10. File Structure

The project follows a simple structure:  
  
```  
/chemicalmanagement  
 /css  
 styles.css  
 /js  
 app.js  
 index.html  
 manifest.json  
 serviceworker.js  
```

# 11. Instructions to Open and Run the Application

Prerequisites:  
1. Browser: You need a modern web browser like Chrome, Firefox, or Safari.  
2. Internet: Only needed for the initial download or cloning from GitHub.  
3. Code Editor: (Optional) A text editor like Visual Studio Code, in case you want to view or modify the code.  
  
 Steps:  
1. Download or Clone the Repository:  
 Go to the GitHub repository link: `[Insert GitHub link here]`.  
 Click the green "Code" button and either download the ZIP or clone the repo:  
 ```bash  
 git clone https://github.com/username/chemicalmanagement.git  
 ```  
2. Unzip (if necessary):  
 If you downloaded the ZIP, unzip the file to a folder on your computer.  
3. Open the Application in a Browser:  
 Navigate to the project folder.  
 Find the `index.html` file and doubleclick to open it in your default web browser.  
4. Using the Application:  
 The chemical management table should now appear, allowing you to sort, add, delete, and move rows.  
 Click on any cell to edit the information, and save changes by pressing Enter or clicking outside the cell.

# 12. Hosting and Deployment

The application is hosted on GitHub Pages, with all assets stored locally to ensure it works offline and loads faster.  
  
 Live Demo Link: `[Insert hosted link here]`

# 13. Future Enhancements

Adding user authentication to save and load chemical data.  
 Implementing pagination for large datasets.  
 Improving editing functionality with data validation.

# 14. Conclusion

This chemical management web app provides an easytouse platform for managing chemical data. Its design focuses on simplicity, making it a lightweight, fast, and userfriendly solution.