**Chemical Supplies Management System Documentation**

**1. Introduction**

The Chemical Supplies Management System is a web-based application designed to manage and display information about various chemical supplies. It provides a user-friendly interface for viewing, adding, editing, sorting and managing chemical data.

**2. System Architecture**

The system follows a client-side architecture, utilizing HTML, CSS, and JavaScript to create a dynamic and interactive user interface. The application is structured into three main components:

1. HTML (index.html): Defines the structure of the web page.
2. CSS (style.css): Handles the styling and layout of the application.
3. JavaScript (script.js): Manages the dynamic behavior and data manipulation.

**3. Key Features**

* Display chemical data in a tabular format
* Add new chemical entries
* Edit existing chemical entries
* Delete chemical entries
* Move rows up and down
* Refresh data
* Save data locally and download as JSON
* Sort table by columns

**4. Data Model**

The chemical data is stored in an array of objects, where each object represents a chemical with the following properties:

* id: Unique identifier for the chemical
* chemicalName: Name of the chemical
* vendor: Supplier of the chemical
* density: Density of the chemical (g/m³)
* viscosity: Viscosity of the chemical (m²/s)
* packaging: Type of packaging
* packSize: Size of the package
* unit: Unit of measurement
* quantity: Available quantity

**5. User Interface Design**

The user interface is designed to be clean, intuitive, and responsive. Key UI elements include:

* Main heading with the title "Chemical Supplies"
* Action icons for various operations (add, move, delete, refresh, save)
* Tabular display of chemical data
* Form for adding/editing chemical details

The design uses a color scheme with blue (#007bff) as the primary color, providing a professional and accessible interface.

**6. Functionality Implementation**

**6.1 Table Management**

* fillTable(): Populates the table with chemical data.
* editRow(id): Enables editing of a specific row.
* moveRowDown() / moveRowUp(): Reorders table rows.
* deleteSelectedRow(): Removes the selected row from the table.
* sortTable(columnIndex): Sorts the table based on the clicked column header.

**6.2 Data Operations**

* refreshData(): Simulates data refresh by randomizing certain values.
* saveData(): Saves data to local storage and offers a download option.
* downloadData(): Generates a JSON file of the chemical data for download.

**6.3 Form Handling**

* The form for adding/editing chemicals is toggled using JavaScript.
* Form submission is handled to either add a new chemical or update an existing one.

**7. Responsive Design**

The application is designed to be responsive, adapting to different screen sizes:

* Desktop: Full view with spacious layout
* Tablet: Adjusted padding and font sizes
* Mobile: Stacked layout for header, centered icons, and optimized table view

**8. Performance Considerations**

* Data is managed client-side to reduce server load and improve responsiveness.
* Event delegation is used for table interactions to optimize performance.
* CSS transitions are used for smooth UI interactions.

**9. Future Enhancements**

Potential areas for future development include:

* Backend integration for data persistence
* User authentication and authorization
* Advanced filtering and search capabilities
* Data visualization features (charts, graphs)
* Batch operations for multiple chemicals

**10. Conclusion**

The Chemical Supplies Management System provides a robust, user-friendly interface for managing chemical inventory data. Its modular design and client-side architecture allow for easy maintenance and future expansions.