Below is the **collection of the Increments** produced at the end of each sprint, reflecting how the product evolved step by step. Each increment meets the **Definition of Done** and represents a **usable** version of the application at that point in time.

**Sprint 1 Increment**

* **Sprint Goal**:  
  *Implement core task management and data persistence, ensuring tasks remain saved after page refresh.*
* **Features Delivered**:
  1. **Basic Task Management**:
     + Users can add new tasks.
     + Users can toggle tasks as complete or incomplete.
     + Users can delete tasks.
  2. **Data Persistence**:
     + Tasks are stored in localStorage (or a simulated JSON file), ensuring they remain after page refresh.
     + On page load, the application retrieves tasks from localStorage.
* **Core Components**:
  1. **index.html**:
     + Structured to display a primary list of tasks.
     + Input field and button for adding new tasks.
  2. **styles.css**:
     + Basic styling for a clean, user-friendly interface.
     + Layout and formatting of task list items, buttons, etc.
  3. **script.js**:
     + JavaScript logic for addTask(), toggleTask(), deleteTask(), renderTasks().
     + saveTasksToLocalStorage() and retrieveTasksFromLocalStorage() for data persistence.
* **Usability**:
  1. The user can create tasks, see them render instantly, and remain in the list even after refreshing or reopening the browser.

**Sprint 2 Increment**

* **Sprint Goal**:  
  *Implement a separate handling of important tasks and introduce an archive for completed tasks.*
* **New Features Delivered**:
  1. **Important Tasks**:
     + A separate “Important Tasks” input and list.
     + Users can add, delete, and toggle important tasks independently from regular tasks.
  2. **Archive / History**:
     + An “Archive Completed” button to move completed tasks into a “History” section.
     + This keeps the main task list clean while preserving records of completed tasks.
  3. **Continued Data Persistence**:
     + Both important tasks and archive data are stored in and retrieved from localStorage.
* **Updated Components**:
  1. **index.html**:
     + Added new sections for “Important Tasks” and an “Archive/History” area.
     + New buttons and lists for these functionalities.
  2. **script.js**:
     + New arrays (e.g., importantTaskList, historyList).
     + Functions like addImportantTask(), toggleImportantTask(), deleteImportantTask(), archiveCompleted(), and renderArchive().
  3. **styles.css**:
     + Adjusted to visually separate the different task sections.
* **Usability**:
  1. Users can distinguish and manage high-priority tasks.
  2. Completed tasks can be archived, reducing clutter in the primary list.
  3. All data remains in localStorage for persistence.

**Sprint 3 Increment**

* **Sprint Goal**:  
  *Polish the UI for a modern, responsive design and implement basic filtering options for easier task navigation.*
* **New Features Delivered**:
  1. **UI/UX Improvements**:
     + More responsive layout via media queries in styles.css.
     + Polished buttons, spacing, and typography for better user experience on various screen sizes.
  2. **Filtering / Sorting**:
     + Filter tasks by status (all, completed, incomplete).
     + Option to sort tasks by creation date or alphabetical order (if implemented).
  3. **Refined Data Handling**:
     + Updated renderTasks() / renderImportantTasks() to handle filtering and sorting logic.
     + Optionally storing creation timestamps for date-based sorting.
* **Updated Components**:
  1. **index.html**:
     + Added UI elements (dropdown or buttons) to filter/sort tasks.
  2. **script.js**:
     + Functions like applyFilter() or similar to choose which tasks to show.
     + Possibly storing dateCreated or other metadata for sorting.
  3. **styles.css**:
     + Enhanced responsiveness (media queries) to ensure a consistent layout on mobile, tablet, or desktop.
* **Usability**:
  1. Users can now view tasks selectively (e.g., only completed tasks).
  2. Sorting options help organize large lists.
  3. The application’s interface looks cleaner and works better across different device sizes.

**Sprint 4 Increment**

* **Sprint Goal**:  
  *Ensure the application is accessible to all users and lay groundwork for potential sync/sharing features.*
* **New Features / Improvements Delivered**:
  1. **Accessibility**:
     + Semantic HTML elements (proper <ul>, <button>, etc.).
     + ARIA tags or roles for key controls.
     + Ensured keyboard navigation works (tab order, focus states visible).
     + Basic screen-reader testing conducted (e.g., VoiceOver, NVDA).
  2. **Initial Share/Sync Planning** (Partial):
     + Conducted requirement gathering, architectural discussion, or partial design for a future sync feature.
     + No fully implemented sharing or backend integration in this increment—just the prep work.
* **Updated Components**:
  1. **index.html**:
     + Reviewed for semantic correctness; added ARIA labels if needed.
  2. **script.js**:
     + May include minor code adjustments for accessibility (focus management, etc.).
  3. **Documentation**:
     + README updated to clarify accessibility improvements and potential future sync architecture.
* **Usability**:
  1. The app is far more friendly to users with screen readers or those relying on keyboard-only navigation.
  2. Plans are in place for future “Share/Sync” functionality, but the app remains primarily single-user with local persistence.

**Summary of Increments**

1. **Sprint 1** gave the application its **core**: adding, toggling, deleting tasks with localStorage support.
2. **Sprint 2** introduced **important tasks** and an **archive** for completed tasks.
3. **Sprint 3** improved **UI/UX** and added **filtering/sorting** capabilities.
4. **Sprint 4** addressed **accessibility** thoroughly and laid groundwork for advanced features (multi-device sync).

At each sprint’s end, the product was **usable and stable** in meeting its acceptance criteria for that stage. This running record of increments lets new teams quickly see how the product grew, what was finalized each sprint, and how to continue building on the existing foundation.