Project Documentation – Offline To-Do List

Project Goal

The goal of this project is to create a fully offline-capable To-Do List web application that runs in the browser. The application enables user registration and login, task creation with multimedia support, drag-and-drop task reordering, and persistent data storage using localStorage. It is built with HTML, CSS, and JavaScript, and uses a service worker to provide offline access.

Implementation Process

1. Designed basic HTML pages for login, registration, and task management.
2. Implemented local authentication using JavaScript and localStorage.
3. Developed the TaskManager class to manage tasks with multimedia support.
4. Added drag-and-drop support for task reordering.
5. Implemented dynamic SVG icons for deletion.
6. Integrated a service worker for offline functionality.
7. Applied responsive and aesthetic styling using CSS.
8. Modularized JavaScript logic across multiple files.

Functional Overview

User Authentication (auth.js)

- Provides local user registration and login functionality.
- User data is stored in localStorage under 'users'.
- Passwords are stored in plain text (for demo/offline use only).
- On successful login, the app redirects to the task list.

Main Application UI (index.html)

- Interface for adding and viewing tasks.
- Tasks support titles, optional notes, image/audio attachments.
- Task actions include complete, delete, and expand.

Header and Footer (partials/)

- header.html and footer.html are loaded dynamically into pages.
- Managed by app.js via fetch API.

Task Management (taskManager.js)

- Handles task creation, deletion, display, and status toggling.
- Integrates multimedia attachments as base64 strings.
- Applies drag-and-drop support and unique task keys per user.

Drag and Drop (dragdrop.js)

- Allows tasks to be reordered by dragging.
- Changes are persisted in localStorage.

Multimedia Handling (media.js)

- Converts selected image/audio files to base64 using FileReader API.
- Used during task creation to store files offline.

SVG Icons (svgHandler.js)

- Generates SVG trash icons for task deletion.

- Provides interaction and callback hooks.

Offline Detection (offline.js)

- Listens for browser 'online' and 'offline' events.
- Displays a notification banner for connectivity changes.

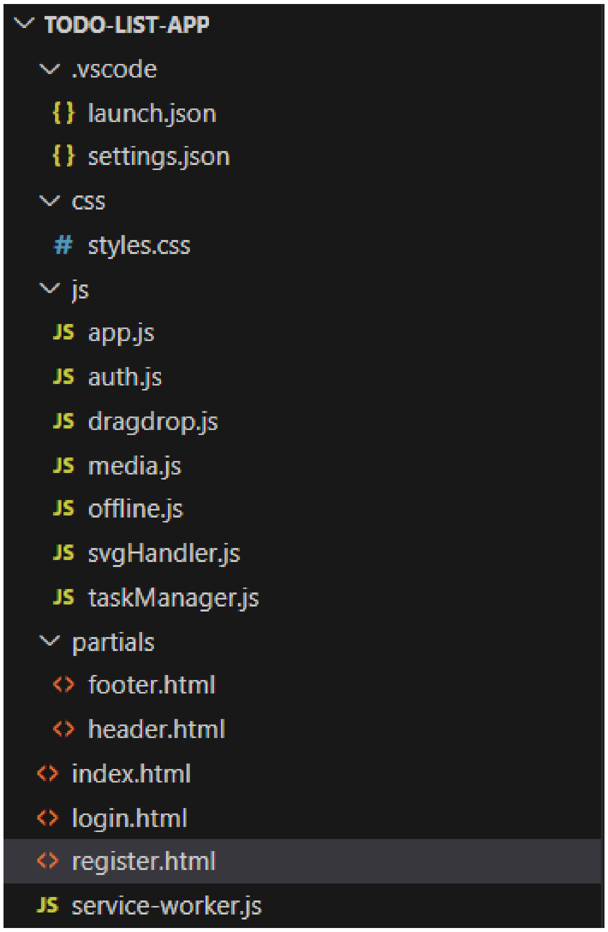
App Initialization (app.js)

- Loads header/footer HTML fragments into the DOM.
- Listens for input focus to auto-scroll and push route hash.
- Initializes the TaskManager class if on main page.

Offline Capability (service-worker.js)

- Caches static files like HTML, CSS, and JS for offline access.
- Removes outdated caches on activation.
- Intercepts fetch requests and serves cached content when offline.

File Structure



```
∨ TODO-LIST-APP
  ∨ .vscode
    {} launch.json
    {} settings.json
  ∨ css
    # styles.css
  ∨ js
    JS app.js
    JS auth.js
    JS dragdrop.js
    JS media.js
    JS offline.js
    JS svgHandler.js
    JS taskManager.js
  ∨ partials
    <> footer.html
    <> header.html
  <> index.html
  <> login.html
  <> register.html
  JS service-worker.js
```

Code Comments

All JavaScript modules are commented to explain their purpose, structure, and behavior. Functions and classes include descriptions of parameters, return values, and side effects. This ensures the code is understandable and maintainable.

How to Use the Application

1. Open the application in your browser by launching 'index.html'.
2. If you are a new user, click 'Register here' on the login page and create an account.
3. Log in using your registered username and password.
4. Once logged in, you will be directed to the main task list interface.
5. To add a task:
    1) Fill in the task title (required).
    2) Optionally add a note, image, or audio file.
    3) Click 'Add Task' to save it.
6. Tasks will appear in a list where you can:
    1) Mark them as complete by clicking the green checkbox.
    2) Expand/collapse to view notes or media by clicking the arrow icon.
    3) Delete tasks using the trash icon.
    4) Reorder tasks by dragging and dropping them into place.
7. You can navigate to 'Add New Task' or 'My Tasks' sections using page routing or by focusing input fields.
8. The app will show your network status and continue working offline using the service worker.
9. All your data is saved in the browser's localStorage and tied to your user account.