**1. To-Do List Application**

Features**:**

* Task Input: Users can add tasks with a title and set a priority level (Low, Medium, High).
* Mark Tasks as Done: Users can mark tasks as complete/incomplete.
* Delete Tasks: Users can remove tasks from the list.
* Task Persistence: The tasks should persist across browser refreshes using LocalStorage.
* Dark Mode Toggle: Users can switch between dark and light themes.
* Task Filter: Users can filter the list of tasks (e.g., Show All, Active, Completed).
* Task Sorting: Sort tasks by their priority.
* Toggling between light/dark mode

**2.Expense Tracker**

Features**:**

* Add Expense: Users can input an expense with a description, amount, and category.
* Edit Expense: Allow users to update or edit previously added expenses.
* Delete Expense: Users can remove any expense from the list.
* Category Filtering: Display expenses filtered by category.
* Total Calculation: Show the total amount spent across all categories.
* LocalStorage: Save expenses and allow persistence even after the page reloads.

**3.Quiz App**

Features**:**

* Question Navigation: Users can navigate through questions (next, previous).
* Answer Options: Each question has multiple choices, and users select one.
* Timer: Optional countdown timer for each question or for the entire quiz.
* Score Calculation: At the end, the app shows the total score based on correct answers.
* Categories: Users can select quiz categories (e.g., General Knowledge, Science).
* Final Score: After the last question, users see their final score and a message based on their performance.

**4.**Weather App

Features:

* + User Input: Users can type the name of a city and get its weather details.
  + Dynamic Content: Displays temperature, humidity, wind speed, and weather conditions like clear, rainy, cloudy, etc.
  + Location Access: Option to use the user's location to fetch weather data automatically.
  + Error Handling: Show error messages for invalid city names or API failures.
  + LocalStorage: Save recent searches so users can quickly access weather data for previously searched cities.

**5.Digital Clock:**

Features**:**

* Display current time in hours, minutes, and seconds
* Optionally show the time in a 12-hour or 24-hour format
* Display the current date (day, month, year)
* Use setInterval() to update the time every second

**6.Tic-Tac-Toe Game**

Features**:**

* A 3x3 grid where players can place their marks
* Check for a winner (three marks in a row, column, or diagonal)
* Restart the game when a winner is found or it's a draw

**7.Random Joke Generator**

Features**:**

* Fetch random jokes from an API (e.g., JokeAPI)
* Display the joke in a text area or div
* Button to fetch a new joke
* Optionally, add a category filter (e.g., programming jokes, random jokes)
* ex api: https://github.com/15Dkatz/official\_joke\_api?tab=readme-ov-file

**8.Calculator Application**

Features**:**

* Input:
  + The calculator will have buttons for digits (0-9), basic arithmetic operators (+, -, ×, ÷), and other functionalities like C (clear), = (equals), and . (decimal point).
  + The input area (display) should show the current number or expression being typed.
* Operations:
  + Users can input a sequence of numbers and operators, and when they press the = button, the result should be displayed.
  + The calculator should handle basic operations:
    - Addition (+)
    - Subtraction (-)
    - Multiplication (×)
    - Division (÷)
  + Pressing the = button will evaluate the result based on the entered expression.
* Clear Functionality:
  + The C button should clear the display (reset the input).
* Decimal Point:
  + The decimal button (.) should allow users to input floating-point numbers.
* Error Handling:
  + The application should handle errors like division by zero, invalid inputs (e.g., two consecutive operators), and invalid expressions.
* Operator Precedence:
  + The calculator must respect operator precedence, where multiplication and division are evaluated before addition and subtraction.
* Result Display:
  + The result should be shown in the display area when = is clicked.
  + After an operation is completed, the display should show the result, and users can continue entering new numbers or operations.

**9.**Personal Portfolio

Features:

* + Landing Page
  + Project Gallery: Display a list of your personal projects with descriptions and links.
  + Contact Form: Include a form where users can send a message (using front-end validation).
  + Skills Section: List your skills with progress bars or icons.
  + Interactive Elements: Add interactive animations for transitions between sections.
  + **Smooth scrolling** to navigate between sections
  + **Mode :** Dark or Light

**Note : All the Application Should be responsive to ALL the Screens, and application should be modern, use all the css properties you know and feel free to walk through new concepts.**

**Reasons Not Allowed**