



## Project Selection Guidelines

You are required to choose any one of the projects from the list provided below. These projects are for reference purposes only. If you have another project idea that can be built using JavaScript, you are free to pursue it. However, make sure your project includes a substantial list of functionalities so that you can maximize your learning experience.

Each project has a specified list of functionalities that should be implemented, along with the benefits you'll gain from working on the project. Please review these carefully and ensure that your chosen project contains all the features outlined.

For projects requiring **APIs (like Weather App, Movie Search App, or Recipe Finder)**, API documentation links have been provided at the end of this document. If you select a project that involves an API, refer to these links for detailed usage information.

Now, go ahead and choose any one of the projects from the list below:

---

### 1. Todo List Application

#### Functionalities:

- Add, edit, and delete tasks
- Mark tasks as completed
- Filter tasks (all, completed, pending)
- Persistent data using localStorage
- Due date for tasks
- Categories/tags for tasks
- Task priority (high, medium, low)

#### Benefits:

- Understanding DOM manipulation
- CRUD operations
- Working with localStorage for data persistence
- Basic date handling and formatting
- Array and object methods (filter, map, etc.)

**Approach:**

- Start by creating a basic HTML structure with input fields and buttons.
  - Use JavaScript to handle task creation, deletion, and updating.
  - Implement filtering functionality using array methods.
  - Integrate localStorage to save and retrieve tasks.
- 

## 2. Quiz App

**Functionalities:**

- Multiple-choice questions with multiple categories
- Timed quizzes
- Track and display scores
- Store high scores in localStorage
- Progress bar for quiz completion
- Randomize questions for each quiz attempt

**Benefits:**

- Handling user input and validating answers
- Timer functionality
- Data management with objects and arrays
- Conditional rendering and updating DOM elements

**Approach:**

- Create a set of quiz questions and answers in an array or fetch from an API.
  - Build functions to handle quiz progression and scoring.
  - Implement a timer and progress bar.
  - Store and display high scores using localStorage.
- 

## 3. Recipe Finder

**Functionalities:**

- Search recipes by ingredient or dish name



- Fetch recipe data from an API (e.g., Edamam, Spoonacular)
- Display recipe details including ingredients, instructions, and images
- Favorite recipes functionality with localStorage
- Filter recipes by dietary preferences (vegetarian, vegan, etc.)
- Pagination for search results

**Benefits:**

- Deepening knowledge of working with APIs
- Managing complex state with arrays and objects
- Enhancing user experience with search and filter options
- Using localStorage for persistent data

**Approach:**

- Get an API key from a recipe service.
  - Build functions to search and display recipes.
  - Create a UI for displaying recipe details.
  - Implement favorites functionality and filtering options.
- 

## 4. Expense Tracker

**Functionalities:**

- Add, edit, and delete expenses
- Categorize expenses (e.g., food, travel, utilities)
- Track income and expenses
- Visual representation of expenses (charts/graphs)
- Monthly/weekly summaries
- Persistent data using localStorage

**Benefits:**

- Practice with CRUD operations
- Visual data representation using libraries like Chart.js
- Data manipulation and calculations
- Persistent state management with localStorage

**Approach:**



- Create a form to input income and expenses.
  - Use JavaScript to handle adding, editing, and deleting expenses.
  - Integrate Chart.js to visualize expenses.
  - Store and retrieve data from localStorage.
- 

## 5. Movie Search App

### Functionalities:

- Search for movies using an API (e.g., OMDB, TMDb)
- Display movie details (title, genre, rating, description, poster)
- Implement pagination for search results
- Add movies to a favorites list
- Rate and review movies
- Persistent data using localStorage

### Benefits:

- Working with APIs and handling JSON data
- Enhancing UI with dynamic data
- Managing state with arrays and objects
- Using localStorage for data persistence

### Approach:

- Get an API key from a movie database service.
  - Create functions to fetch and display movie data.
  - Implement search and pagination functionality.
  - Build a favorites list and review system using localStorage.
- 

## 6. Notes App

### Functionalities:

- Add, edit, and delete notes
- Categorize notes with tags



- Search notes by title or content
- Markdown support for formatting notes
- Persistent data using localStorage
- Color-coded notes

**Benefits:**

- Practicing CRUD operations
- Enhancing user experience with search and tags
- Working with localStorage for data persistence
- Learning Markdown for note formatting

**Approach:**

- Create a form to input notes.
  - Use JavaScript to handle adding, editing, and deleting notes.
  - Implement search functionality and tag filtering.
  - Store and retrieve notes using localStorage.
- 

## 7. Countdown Timer Project

**Functionalities:**

- Set a countdown to a specific date and time
- Display remaining days, hours, minutes, and seconds
- Update the display in real-time
- Notify the user when the countdown reaches zero
- Save the countdown state using localStorage

**Benefits:**

- Learn date and time calculations
- Practice DOM manipulation
- Use timing functions (setInterval)
- Handle user inputs and events
- Understand data persistence with localStorage

**Approach:**



- Add event listeners for setting the countdown.
  - Write a function to calculate and display remaining time.
  - Use setInterval for real-time updates.
  - Implement a notification for when the countdown ends.
  - Save and retrieve countdown state using localStorage.
- 

## API Documentation

For projects that require APIs (e.g., Weather App, Movie Search App, Recipe Finder), visit the respective API documentation pages for better understanding and detailed information about endpoints:

- OpenWeatherMap API: [OpenWeatherMap API Documentation](#)
- OMDB API: [OMDB API Documentation](#)
- TMDb API: [TMDb API Documentation](#)
- Edamam Recipe API: [Edamam API Documentation](#)
- Spoonacular API: [Spoonacular API Documentation](#)