

Todo App Assignment - TaskFlow

You are tasked with building a **Todo Application** called "TaskFlow" - a personal productivity companion that helps users manage their tasks across different stages (Todo, Completed, Archived). This is a full-stack frontend project that will test your skills in HTML, CSS, and Vanilla JavaScript.

Project Objectives

By completing this assignment, you will demonstrate your ability to:

- Build responsive web interfaces using HTML/CSS
- Implement dynamic functionality with Vanilla JavaScript
- Work with browser localStorage for data persistence
- Integrate with external APIs
- Handle form validation and user input
- Manage application state and user flow

Design Requirements

We have a reference design available for you to check out 🙌 [LINK](#)

Creative Freedom:

- You have complete creative freedom to design the user interface
- The provided mockups are reference designs - feel free to improve upon them
- Ensure your design is responsive and works on different screen sizes

Must-Have Design Elements:

- Clean, modern interface
- Clear visual distinction between different task stages
- Responsive design that works on desktop and mobile

Technical Requirements

Technology Stack:

- HTML5
- CSS3 (you may use any CSS framework or write custom CSS)
- Vanilla JavaScript (No frameworks like React, Vue, or Angular)

- Browser localStorage for data persistence
- External APIs for dummy data and profile pictures

Application Structure

Your application should consist of two main pages:

Note: This is just a suggested structure. You can change the structure if you want and we would encourage you to do so.

1. Landing Page (**index.html**)

Purpose: Age verification and user registration

Key Features:

- Age verification form (must be >10 years old)
- Name input field
- Date of birth input field
- Form validation
- Data storage in localStorage
- Automatic redirect for returning users

User Flow:

1. User enters their name and date of birth
2. System validates age (must be over 10 years)
3. If valid, store user data in localStorage
4. Redirect to app.html
5. On subsequent visits, if user data exists, automatically redirect to app.html

2. Main Application (**app.html**)

Purpose: Todo management interface

Key Features:

- User profile display with avatar
- Task management across three stages: Todo, Completed, Archived
- Add new tasks functionality
- Move tasks between stages
- Archive tasks (no delete functionality)
- Display last modified timestamps
- Persistent data storage

Detailed Feature Specifications

Landing Page Features

Age Verification System:

- Implement date picker for date of birth
- Calculate age from date of birth
- Show error message if user is 10 years old or younger
- Only allow access if user is over 10 years old

Data Persistence:

- Check for existing data on page load
- Auto-redirect if valid user data exists

Form Validation:

- Required field validation for name and date of birth
- Age validation with user-friendly error messages
- Prevent form submission with invalid data

Main Application Features

Header Section:

- Display user's name from localStorage
- Show profile picture using UI Avatars API:
https://ui-avatars.com/api/?background=0D8ABC&color=fff&name={USER_NAME}
- Replace {USER_NAME} with actual user name from localStorage
- Include sign out functionality that clears localStorage and redirects to index.html

Task Stage Management: Create three distinct stages with dynamic counters:

- **Todo (Active Tasks):** Tasks that need to be completed
- **Completed:** Tasks that have been finished
- **Archived:** Tasks that are no longer active but kept for reference

Task Input System:

- Text input field for new tasks
- Submit button to add tasks
- Tasks are added to currently active/selected stage

- Input validation (prevent empty tasks)
- Clear input field after successful submission

Task Cards: Each task should display:

- Task title/description
- Action buttons based on current stage:
 - **Todo Stage:** "Mark as Completed" and "Archive" buttons
 - **Completed Stage:** "Move to Todo" and "Archive" buttons
 - **Archived Stage:** "Move to Todo" and "Move to Completed" buttons
- Last modified timestamp (format: MM/DD/YYYY HH:MM:SS AM/PM)
- Update timestamp whenever task is moved between stages

Initial Data Loading:

- On first visit to app.html, populate Todo stage with dummy data
- Use DummyJSON API: <https://dummyjson.com/todos>
- Extract todos from API response (ignore "completed" field)
- Store initial data in localStorage for persistence

API Integration Requirements

UI Avatars API:

- Base URL: <https://ui-avatars.com/api/>
- Parameters: [?background=0D8ABC&color=fff&name={USER_NAME}](https://ui-avatars.com/api/?background=0D8ABC&color=fff&name={USER_NAME})

DummyJSON Todos API:

- Endpoint: <https://dummyjson.com/todos>
- Use only the `todo` field from each item
- Ignore the `completed` status from API
- Transform API data to match your task object structure

Evaluation Criteria

Your assignment will be evaluated based on:

Functionality (40%)

- All required features work as specified
- Proper error handling and validation
- Data persistence across browser sessions
- API integration works correctly

Code Quality (25%)

- Clean, well-organized code structure
- Proper use of HTML semantics
- Efficient JavaScript implementation
- Consistent coding style and naming conventions

User Experience (20%)

- Intuitive and user-friendly interface
- Responsive design across devices
- Smooth interactions and transitions
- Clear visual feedback for user actions

Design & Creativity (15%)

- Visual appeal and professional appearance
- Creative improvements to the reference design
- Consistent design language throughout the app

Project Milestones

Milestone 1: Landing Page - Design

- Basic HTML structure
- Responsive layout with CSS
- Name + DOB form UI

Milestone 2: Landing Page - Functionality

- Age validation (>10)
- Form validation
- Save to localStorage
- Redirect to app.html

Milestone 3: Todo App - Basic UI & Functionality

- Layout for Todo app
- Add + display tasks

Milestone 4: Completed & Archive Features

- Display user name + avatar
- Add Completed & Archived sections

- Buttons to move tasks between stages
- Show last modified timestamp

Milestone 5: Final Touches

- Fetch dummy todos from API
- Sign out functionality
- Full responsive design
- Bug fixes + localStorage updates
- Smooth transitions + UI polish

Submission Requirements

Deliverables:

1. Complete source code (HTML, CSS, JavaScript files)
2. Hosted on public github repo.
3. And if possible host it on any free hosting site such as vercel.

Testing Checklist: Before submission, ensure:

- ☐ Age verification works correctly
- ☐ User data persists across browser sessions
- ☐ All task stage transitions work properly
- ☐ API integrations function as expected
- ☐ Responsive design works on mobile devices
- ☐ No console errors or broken functionality

Tips for Success

Development Approach:

1. Start with basic HTML structure for both pages
2. Implement core JavaScript functionality
3. Add styling and responsive design
4. Test thoroughly across different scenarios
5. Add creative enhancements and polish

Bonus Points:

- Implement additional features like task search or filtering
- Add smooth animations and transitions
- Include data export/import functionality

- Implement keyboard shortcuts for common actions
- Add task categories or priority levels

Questions and Support

If you have any questions about the requirements or need clarification on any aspect of the assignment, please don't hesitate to reach out. We're here to help you succeed!

Good luck, and we look forward to seeing your creative solution!