

Tasks

EASY LEVEL PROJECTS (HTML + CSS only)

1. Personal Profile Page

Goal: Create a personal webpage introducing yourself.

Skills You'll Learn:

- Basic HTML structure (`<header>`, `<section>`, `<footer>`)
- Text formatting with `<h1>`, `<p>`, `<a>`, ``
- Adding images with ``
- CSS styling (fonts, padding, colors, spacing)

Project Structure:

```
- index.html
- styles.css
/images
  - your-photo.jpg
```

Key Features:

- Header with your name and profile picture
- About section with a short bio
- List of hobbies or skills
- Contact section with email or social links

Bonus Challenges:

- Make the profile picture circular with `border-radius: 50%`
- Add hover effect on links (`color` change or underline)
- Use Google Fonts

2. Navigation Bar (Horizontal Menu)

Goal: Create a simple and clean horizontal navbar.

Skills You'll Learn:

- Using ``, ``, `<a>` for navigation structure
- Applying Flexbox for horizontal layout
- Styling hover and active states

Project Structure:

```
- index.html  
- styles.css
```

Key Features:

- 4–5 links (Home, About, Services, Contact)
- Horizontal layout with spacing
- Hover effect with underline or background color

Bonus Challenges:

- Make the current link highlighted using a `.active` class
- Add a mobile toggle menu (basic hamburger icon for future enhancement)

INTERMEDIATE LEVEL PROJECTS

1. Responsive Landing Page

Goal: Design a one-page website that works on both mobile and desktop.

Skills You'll Learn:

- Page layout planning
- Responsive design with media queries
- Button styling and hover animations

Project Structure:

```
- index.html
- styles.css
/images
  - hero-bg.jpg
```

Key Features:

- Hero section with background image + CTA
- Features section with icons and text
- About section with profile or team details
- Contact form (non-functional)
- Footer with social links

Bonus Challenges:

- Add smooth scroll behavior with CSS (`scroll-behavior: smooth`)
- Animate the buttons (`transition: all 0.3s`)

2. Image Gallery with Modal Preview

Goal: Build a gallery grid where clicking a thumbnail opens a larger image.

Skills You'll Learn:

- CSS Grid or Flexbox layout
- JavaScript DOM manipulation
- Modal design and behavior

Project Structure:

```
- index.html
- styles.css
- script.js
/images
  - image1.jpg
  - image2.jpg
```

Key Features:

- 6–9 image thumbnails in a grid
- Click opens full-size image in a modal
- Modal includes a close icon/button

Bonus Challenges:

- Add left/right arrows to navigate images
- Add a fade-in animation to modal

ADVANCED LEVEL PROJECTS

1. Validated Contact Form

Goal: Build a form and ensure users fill it out correctly.

Skills You'll Learn:

- HTML5 input types
- JavaScript validation (form events)
- Showing error messages dynamically

Project Structure:

```
- index.html  
- styles.css  
- script.js
```

Key Features:

- Fields: Name, Email, Password, Message
- Required field checks
- Password length check (e.g., min 6 chars)
- Valid email format check using regex

Bonus Challenges:

- Highlight valid/invalid fields using border color
- Disable submit button until all fields are valid

Project Goal:

Build a clean, interactive one-page website for a fictional product like a smartwatch, coffee brand, or travel backpack. The page will include a hero banner, pricing section, contact form, and smooth navigation—all styled with HTML & CSS and enhanced with basic JavaScript for interactivity.

Skills You'll Practice:

- Semantic HTML structure
- CSS Flexbox and Grid layout
- JavaScript DOM manipulation
- Basic form validation
- Smooth scroll and navigation highlighting

1. Header

- Contains a logo and navigation links ([Home](#), [About us](#), [Contact](#))
- Sticky at the top when scrolling
- Smooth scroll to page sections

2. Hero Section

- **Full-screen banner with:**
 - Product name
 - Catchy tagline
 - CTA button (e.g., "Order Now")
- Use a background image or gradient
- Responsive design with centered text

3. About Section

Purpose:

Introduce your product or brand in a simple, clean layout that works on both desktop and mobile.

Structure:

- **Left Side:**
 - A heading (e.g., "About Our Product")
 - A short paragraph describing:
 - What the product is
 - Who it's for
 - Why it's special
- **Right Side:**
 - A product image or illustration

4. Contact Section

- Fields: `Name`, `Email`, `Message`
- **JavaScript validations:**
 - Check if fields are filled
 - Verify valid email pattern
- Show success message after submission (using `alert()` or inline message)

5. Footer

- Includes:
 - Social media icons or text placeholders
 - "Back to top" link
 - Copyright

JavaScript Features:

- Contact form validation:
 - No empty fields
 - Email pattern check (using regex)
- Smooth scroll behavior for nav links
- Optional: Highlight nav links as user scrolls

JavaScript

Easy level

1. Light/Dark Theme Toggle with Persistence

Goal: Allow users to toggle between light and dark mode.

Skills You'll Learn:

- CSS variables and classes
- Toggle button design
- Saving preferences to `localStorage`

Project Structure:

- `index.html`
- `styles.css`
- `script.js`

Key Features:

- A toggle button (or switch)
- `dark-theme` and `light-theme` classes
- Store user selection so it persists on refresh

Bonus Challenges:

- Animate transition between themes
- Make toggle keyboard accessible

Intermideate Level

2. Weather App (Using OpenWeatherMap API)

Goal: Build a real-time weather checker using API integration.

Skills You'll Learn:

- Using `fetch()` to call APIs
- Async/await in JavaScript
- Updating DOM with live data

Project Structure:

```
- index.html
- styles.css
- script.js
```

Key Features:

- Input for city name
- Show weather info: temperature, city, description
- API call to OpenWeatherMap (API key needed)

Bonus Challenges:

- Show icon based on weather condition
- Add background image based on weather type

Advance level

Daily Task Manager

Project Goal:

Build a **daily task manager** that allows users to:

- Add tasks with due dates
 - Organize them into categories (e.g., Work, Personal, Shopping)
 - Mark them as complete
 - Filter by category
 - Persist data using **Local Storage** so nothing is lost on page reload
-

Skills You'll Practice:

- DOM creation & manipulation
 - Event delegation
 - Working with `localStorage`
 - Date handling (`Date()` object)
 - Basic array methods (`filter`, `map`, `forEach`)
 - Conditional rendering
-

Suggested Project Structure:

pgsql

CopyEdit

task-manager/

├── index.html

├── css/

│ └── styles.css

├── js/

│ └── script.js

└── README.md

Core Features:

1. Add Task

- Input field for task name
- Dropdown to choose category
- Date picker for due date
- Add button creates a new task card

2. Task List View

- Show each task as a card:
 - Task name
 - Category tag
 - Due date
 - Checkbox to mark complete
 - Delete icon/button

3. Filter Tasks by Category

- Buttons or dropdown to filter (e.g., Show only "Work" tasks)

4. Data Persistence

- Use `localStorage` to store tasks
- Retrieve and display saved tasks on page load

5. Mark as Completed

- When checked, strike through the task or fade it
- Optionally: move it to a "Completed" section

Recipe Finder App

Project Goal:

Create a **Recipe Finder App** that allows users to search for recipes by keyword (e.g., "pasta"), view a list of matching dishes with thumbnails, and see details like ingredients and instructions when a recipe is selected. You'll use a **public recipe API** or mock data, and manage the entire app with JavaScript.

Skills You'll Practice:

- API fetching (or working with mock JSON data)
 - Dynamic card creation in DOM
 - Event delegation
 - Toggle views and modals
 - Search functionality and input handling
 - State management in JavaScript
-

Suggested Project Structure:

recipe-finder/

├── index.html

├── css/

| └── styles.css

├── js/

| └── script.js

└── README.md

Core Features:

1. Search Recipes

- Input field for keywords
- Fetch data from an API (like TheMealDB) or a local JSON array
- Display recipes as cards with:
 - Dish name
 - Thumbnail image

2. View Recipe Details

- Click on a recipe card to open a modal or expandable card
- Show:
 - Ingredients list
 - Step-by-step instructions
 - Image of the dish
- Add a "Close" button for the detail view

3. Save Favorite Recipes

- Allow users to click a heart icon to save/unsave recipes
- Store favorites in `localStorage`
- Option to filter and view only favorites

Final Instruction

Once you've completed each project, deploy it on Vercel or Netlify to make it live and accessible online.

Link to learn vercel-hosting :

<https://youtu.be/b2bldtSwDhc?si=VvrOvF2pgzKUzVc1>

Sample (HTML & CSS) Project link :

<https://front-end-task-rho-pied.vercel.app/>