

# Front-End Web Development Tasks

## Note:

Participants must complete at least 3 tasks for the 2-week internship and 4 tasks for the 1-month internship — from any level.

## Level 1

### Task 1: Build a Collapsible Sidebar

#### Description:

Create a sidebar section that can open and close when the user clicks a button.

The sidebar should include:

- A placeholder for a logo at the top
- A list of 4–5 links with icons
- A button to toggle the sidebar open or closed

Focus on clean layout and smooth open/close animation.

#### Tools & Libraries:

[HTML](#) [CSS](#) [JavaScript](#)

#### Covered Topics

[HTML](#) | [CSS](#) | [JavaScript](#) | [Component Layout](#) | [Animation](#) | [Toggle Logic](#)

#### Bonus:

Make the sidebar responsive on mobile screens.

### Task 2: Create a Contact Form

#### Description:

- Build a contact form that looks good on all screen sizes.  
The form should include: Full Name, Email Address, Subject, Message, and Submit button.
- Use a clean layout with spacing between inputs and a visible button. Make sure it works well on phones and desktops.

#### Tools & Libraries:

[HTML](#) [CSS](#) [JavaScript](#)

#### Covered Topics

[HTML Forms](#) | [CSS Styling](#) | [Responsive Design](#) | [Layout](#)

#### Bonus:

Add basic validation using JavaScript (e.g., required fields, valid email format).

# Front-End Web Development Tasks

## Level 2

### Task 3: Build a One-Page Website for a Task App

#### Description:

Create a full web page for a fictional app called TaskFlow – a tool that helps people organize their tasks.

The page must include:

- A top section with the app name, a short message, and a button
- A section that shows 3 features of the app (each with an icon and description)
- A section with 2–3 reviews or quotes from happy users
- A section with 3 pricing options (for example: Free, Pro, Team)
- A footer with contact links and social media icons

The design should be clean and modern. Make sure it looks good on phones and large screens.

#### Tools & Libraries:

[HTML](#) [CSS or Tailwind](#) [JavaScript](#)

#### Covered Topics

[React](#) | [JSX](#) | [Responsive Design](#) | [Layout Design](#) | [Component Composition](#)

#### Bonus:

Add a small animation when each section appears while scrolling down the page.

### Task 4: Create a Personal Blog Homepage

#### Description:

- Build a homepage for a personal blog showing blog posts in a card layout.
- Add a simple header with the blog title
- Add a section that displays blog post cards (title, image, short description, and date)
- Add a category filter to switch between types of posts (e.g., Tech, Travel, Food)
- Limit the number of visible posts to a specific number per page.

#### Tools & Libraries:

[HTML](#) [CSS or Tailwind](#) [JavaScript](#)

#### Covered Topics

[React](#) | [Props & State](#) | [Responsive Grid](#) | [Filtering](#) | [UI Layout](#) | [Conditional Rendering](#)

#### Bonus:

Add a search input to filter posts by title keyword.

# Front-End Web Development Tasks

## Level 2

### Task 5: Build a Responsive Landing Page for a Tech Product

#### Description:

- Design and implement a marketing landing page for a fictional tech product or SaaS tool.
- Include sections like Hero, Features, Testimonials, Pricing, and a Call-to-Action.
- The page should adapt well on mobile, tablet, and desktop.
- Use icons, cards, and clean grid structure.

#### Tools & Libraries:

HTML CSS JavaScript Flexbox/Grid Bootstrap or Tailwind (optional)

#### Covered Topics

Responsive design | layout structuring | CTA effectiveness | UX writing

#### Bonus:

Integrate a theme toggle (light/dark mode) using JavaScript.

# Front-End Web Development Tasks

## Level 3

### Task 6: Multi-Page Dashboard for a Fictional Freelance Client

#### Description:

- Create a 2–3 page admin dashboard interface for a fictional freelance client.  
The dashboard should include the following pages:
  - Overview Page: Show summary cards (e.g., Total Projects, Earnings, Tasks Due), a recent activity list, and basic stats.
  - Projects Page: Display a list of client projects with project name, status, and deadline in a table or card layout.
  - Profile Settings Page (Optional): Allow the user to view/edit profile info (name, email, password placeholder fields).
- Additional Requirements:
  - Use a sidebar and top header that stay consistent across all pages.
  - Apply routing to switch between pages without refreshing.
  - Use a chart library to display simple visuals (e.g., monthly earnings bar chart or pie chart of task types).
  - The layout should be responsive and user-friendly on both desktop and mobile screens.

#### Tools & Libraries:

ReactJS | React Router | Chart.js, or Recharts | Tailwindcss

#### Covered Topics

React | React Router | Multi-Page Layout | Reusable Components | Charts | Responsive Design | Conditional Rendering

#### Bonus:

Add a notification dropdown in the top header that shows a list of the 3 most recent user activities, using mock data.

# Front-End Web Development Tasks

## Level 3

### Task 7: Build a Real-Time Weather Dashboard Using an API

Description:

- Create a dashboard that fetches and displays real-time weather data for multiple cities.
- Display details like temperature, weather icons, and 3-day forecasts.
- Include a search functionality and loading states.
- Emphasize clean and minimal UI.

Tools & Libraries:

HTML   CSS   JavaScript   OpenWeatherMap API   Axios or Fetch

Covered Topics

API integration | dynamic rendering | asynchronous operations | DOM manipulation

Bonus:

Add user's current location support and auto-fetch weather data for it.

# Front-End Web Development Tasks

## Industry-Level

### Task 8: Build a Job Application Tracker Web App

#### Description:

- Develop a full, responsive web application that allows users to track their job applications.  
The app should include the following pages:
  - Dashboard: Display a list of saved job applications. Each item should show the company name, job title, application status, and applied date.
  - Add Job Page: A form for entering a new job application with fields like company name, job title, status (select from: Applied, Interviewing, Offer, Rejected), application date, and notes.
  - Job Details Page: When a job from the dashboard is clicked, show full details including notes and the ability to edit or delete the entry.
- Additional Requirements:
  - Use React Router for navigation between pages.
  - Use global state (Context API or Redux) to manage the list of applications.
  - The layout should be optimized for mobile and desktop views.
  - Keep the app frontend-only. Store job data temporarily in localStorage.

#### Tools & Libraries:

ReactJs | React Router | Context API or Redux | Tailwindcss

#### Covered Topics

React | React Router | Form Handling | State Management | Responsive Design | LocalStorage

#### Bonus:

Allow users to export their list of job applications to a downloadable .json file, and also provide an option to import from a .json file to restore saved data.