

Internship Task: Job Listing Website with Job Cards and Filters

Task Name:

Job Listing Website with Job Cards and Filters

Duration:

10 Days

Task Description:

The task is to design and develop a functional and responsive Job Listing website. The website should display job openings as cards and allow users to filter them based on various criteria such as job title, location, category, and experience level. This project will help students build hands-on experience in creating dynamic and user-friendly web applications.

Objectives:

- Build a responsive Job Listing website with a clean and modern design.
 - Implement job cards to display job details.
 - Add filter functionality to sort or search for jobs based on user input.
 - Focus on UI/UX design principles for a user-friendly experience.
 - Use GitHub for version control to track progress.
-

Features to Implement:

1. Job Cards:

- Each job listing should be displayed as a card.
- Include job details like title, company name, location, salary (optional), and a brief description.
- Add a "View More" button to expand additional details in a modal or new section.

2. Filter Options:

- Allow users to filter jobs based on:
 - **Job Title:** Search by specific keywords.

- **Location:** Filter by city or country.
 - **Category:** Choose job categories (e.g., IT, Marketing, Design).
 - **Experience Level:** Filter by fresher, mid-level, or senior roles.
 - Include a clear/reset filters option.
 - 3. **Interactive Features:**
 - A search bar for quick keyword-based searches.
 - Pagination for job listings if there are many results.
 - 4. **Responsive Design:**
 - Ensure the website adapts well to different devices (mobile, tablet, desktop).
 - 5. **Additional Enhancements (Optional):**
 - Add a "Save Job" button to simulate saving a job for later.
 - Use sample JSON data or an API (e.g., mock job API) for dynamic content.
-

Technology Stack:

- **HTML5**
 - **CSS3** (including frameworks like Bootstrap or Tailwind CSS for responsiveness)
 - **JavaScript** (for filter functionality and interactivity)
 - **Optional Enhancements:**
 - Use a library like React.js for better state management and dynamic updates.
 - Integrate with a job API (e.g., Mockaroo or JSONPlaceholder) for real-time data.
-

Interactive Components:

- Dynamic filters that instantly update results without reloading the page.
 - Modal or expandable sections for viewing detailed job descriptions.
 - Smooth animations for transitions like hover effects and modal pop-ups.
-

Version Control:

- Students must create a repository on GitHub.
- Commit changes regularly (with meaningful commit messages).

- Push the final project code to the GitHub repository.
-

Presentation:

- Deploy the project using platforms like GitHub Pages, Netlify, or Vercel.
 - Prepare a brief presentation/demo (5-10 minutes) explaining:
 - Features and functionality of the website.
 - Technologies used in development.
 - Challenges faced and solutions applied.
-

Deliverables:

1. A fully functional Job Listing website with job cards and filters.
 2. Source code pushed to the GitHub repository.
 3. Link to the deployed webpage.
 4. Documentation (README file) containing:
 - Project Overview.
 - Instructions to run the project.
 - Technologies used.
-

Task Evaluation Criteria:

1. **Design Quality:** UI/UX, responsiveness, and overall appearance. **(25%)**
 2. **Functionality:** Filters, job cards, and interactive features. **(25%)**
 3. **Code Quality:** Clean, readable, and well-structured code. **(20%)**
 4. **GitHub Usage:** Proper commits and repository structure. **(15%)**
 5. **Presentation/Demo:** Clear explanation of work and project deployment. **(15%)**
-

Tools and Resources:

- **Code Editors:** VS Code, Sublime Text, etc.
- **Version Control:** Git and GitHub
- **CSS Frameworks:** Bootstrap or Tailwind CSS

- **Deployment Platforms:** GitHub Pages, Netlify, Vercel
 - **Design References:** Figma, Canva, or Dribbble for UI inspiration
-

Deadline:

The task must be submitted within **10 days** from the date of assignment.