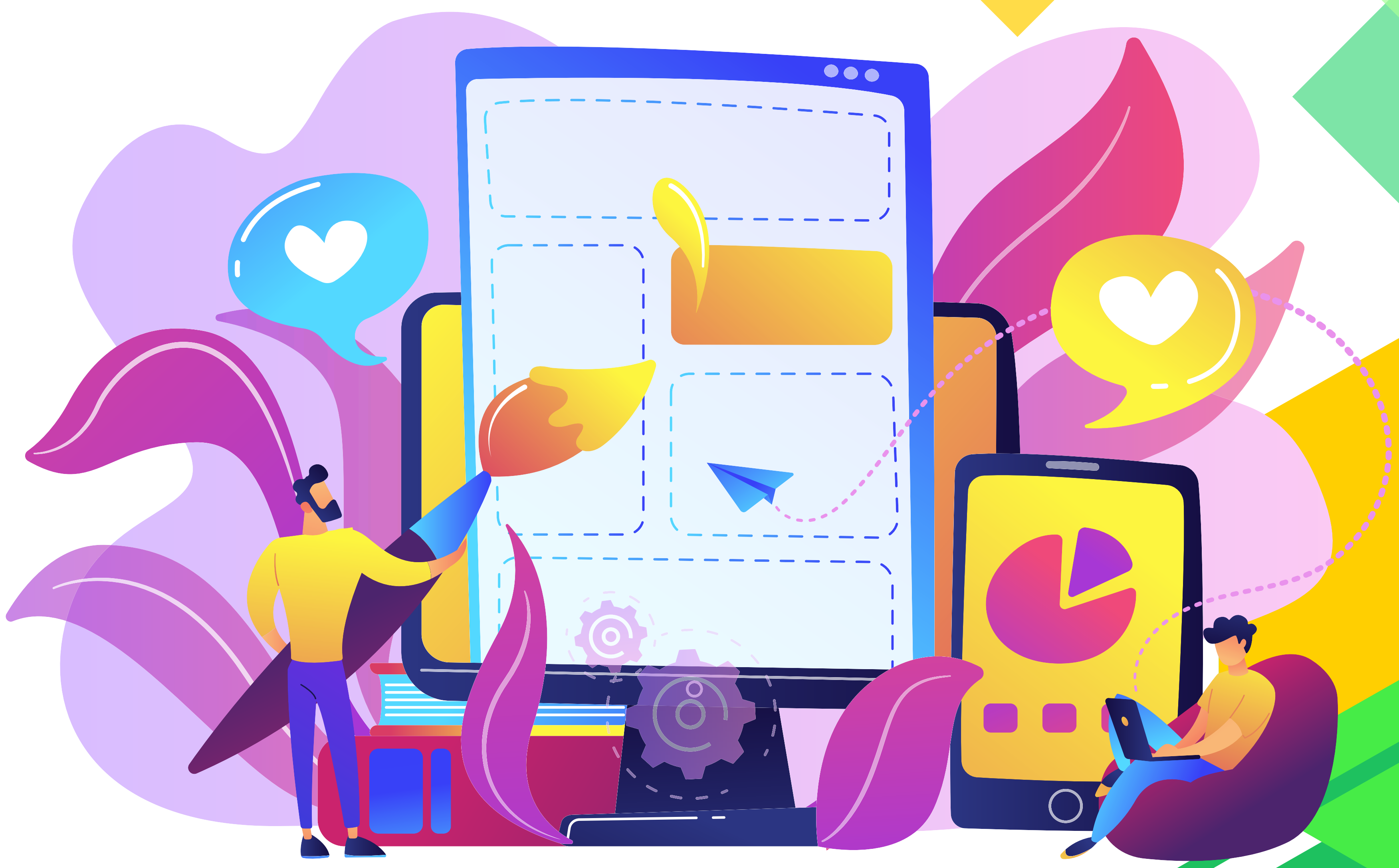
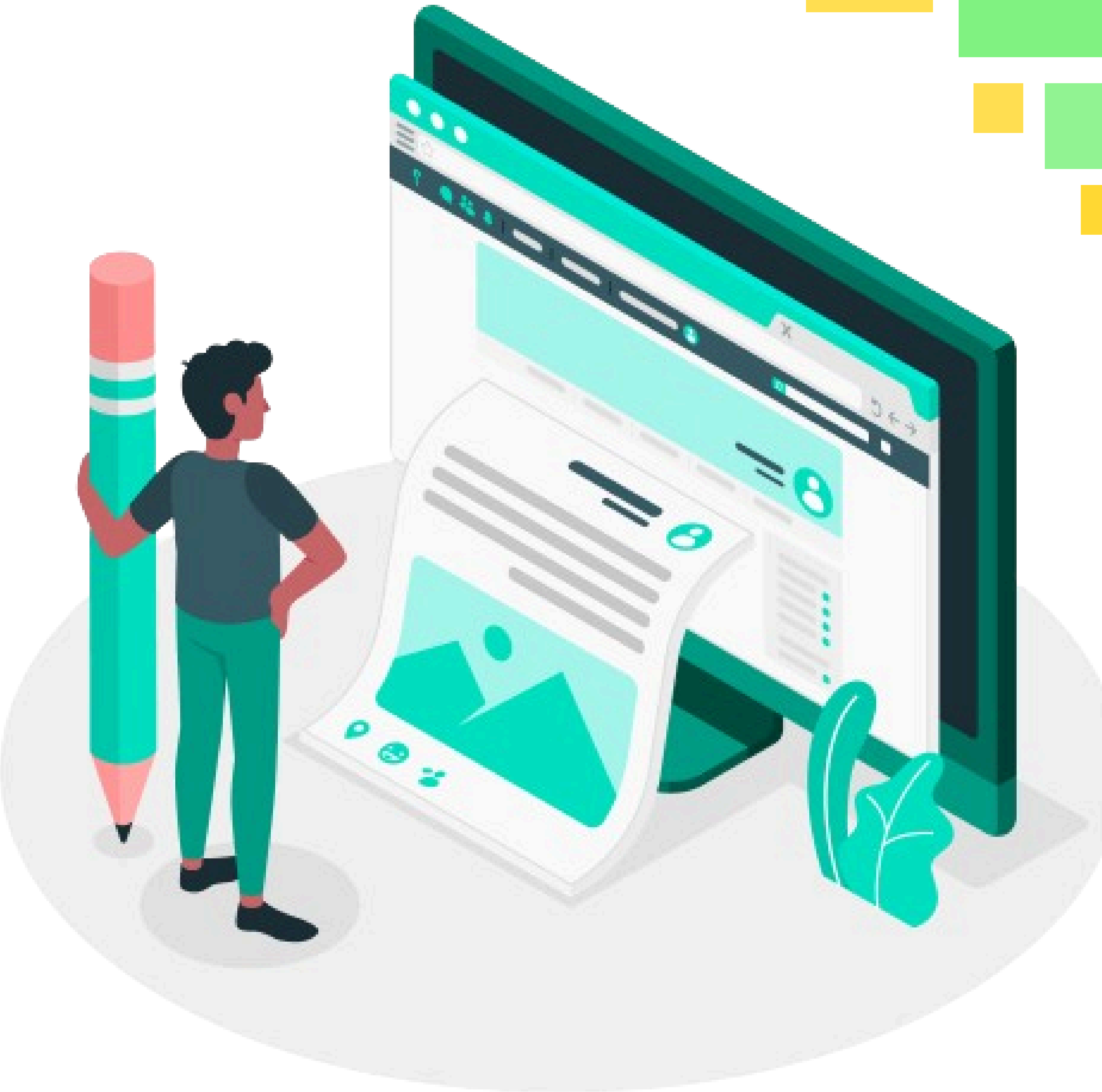
Codveda

Front-End Development



Content

1. **About Us**
2. **Instructions**
3. **Submission**
4. **Task List**



About Us

Welcome to Codveda Technology, where innovation meets excellence. Founded with a vision to empower businesses through cutting-edge IT solutions, we specialize in delivering tailored services that drive success in the digital era.

At Codveda, we offer a diverse range of services, including web development, app development, digital marketing, SEO optimization, AI/ML automation, and data analysis.

Our team of skilled professionals is committed to helping businesses unlock their full potential by providing innovative, scalable, and reliable solutions.

INSTRUCTIONS

- Update your LinkedIn profile with your achievements, including the offer letter and completion certificate. Mention and tag @Codveda in your posts.
- Use hashtags like #CodvedaJourney, #CodvedaExperience, and #FutureWithCodveda to showcase your progress and experiences.
- Share your project completion updates on LinkedIn, accompanied by a video explanation and the GitHub project repository link.
- You will be provided with four tasks. Select and complete any three tasks within your domain to fulfill the internship requirements.
- Submit your completed tasks via the Codveda submission form. Ensure all tasks are submitted within the allocated 15-day period.

SUBMISSION

- Create a professional video showcasing your internship projects and achievements.
- Host the video on LinkedIn to provide proof of your work and establish credibility among your peers. Consider tagging Codveda Technology in your posts to ensure they are notified of your work using hashtags like #CodvedaAchievements and #CodvedaProjects.
- A SUBMISSION FORM will be shared later. Till then, please continue your tasks and maintain a separate file for each level.
- When posting the video on LinkedIn, include engaging content that highlights your contributions and skills. Tailor the post to your specific internship domain to maximize impact and visibility.

Level 1 (Basic)

Task 1: Responsive Landing Page:

- Develop a responsive landing page using HTML, CSS, and basic JavaScript. The page should include essential sections like a navigation bar, hero section, features, and a footer. It must be fully responsive, adjusting seamlessly to different screen sizes (mobile, tablet, and desktop).

Objectives:

- Utilize semantic HTML tags for better accessibility and SEO.
- Implement CSS Flexbox or Grid for layout design.
- Ensure mobile-first responsiveness using media queries.
- Add smooth scrolling and basic animations for a modern look.

Level 1 (Basic)

Task 2: Interactive Form

- Create an interactive form that validates user input using JavaScript. The form should include fields like name, email, phone number, and password. Display real-time error messages for invalid inputs and confirmation upon successful submission.

Objectives:

- Validate form inputs (e.g., required fields, email format, password strength).
- Provide dynamic error messages without page reload.
- Implement input focus and blur events for better UX.
- Style the form for a clean and professional appearance.

Level 1
(Basic)

Task 3: Basic DOM Manipulation

- Build a counter application that demonstrates basic DOM manipulation using vanilla JavaScript. The counter should have increment, decrement, and reset buttons with a display showing the current count.

Objectives:

- Use JavaScript to dynamically update the DOM.
- Implement event listeners for button clicks.
- Ensure the counter doesn’t go below zero.
- Style the app for a minimalistic design.

Level 2 (Intermediate)

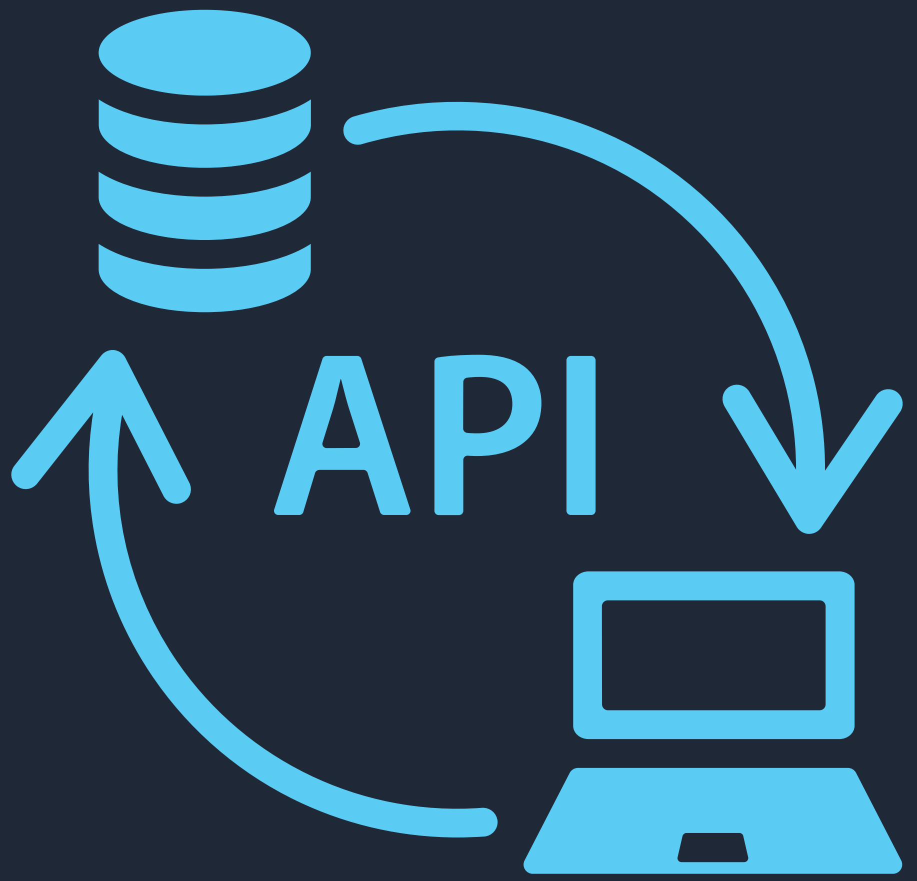
Task 1: Single Page Application (SPA)

- Create a simple Single Page Application using React or Vue.js. The application should have at least three pages (e.g., Home, About, and Contact) with smooth navigation without full page reloads.

Objectives:

- Use React Router or Vue Router for navigation.
- Maintain state across pages using state management (e.g., React Context or Vuex).
- Ensure fast navigation and smooth transitions.
- Deploy the SPA on a platform like Netlify or Vercel.

Level 2 (Intermediate)



Task 2: REST API Integration

- Integrate a public REST API (e.g., GitHub, OpenWeather) to fetch and display data dynamically. Implement search or filter functionality to enhance user interactivity.

Objectives:

- Use Axios or Fetch API to retrieve data.
- Display dynamic content based on user input.
- Handle loading states and error messages gracefully.
- Optimize performance with debounced search queries.

Level 2 (Intermediate)

Task 3: CSS Frameworks

- Build a user interface using a popular CSS framework like Bootstrap or TailwindCSS. Customize the default theme to align with a brand’s style guide, ensuring a cohesive design language.

Objectives:

- Utilize pre-built components for faster development.
- Customize colors, typography, and spacing.
- Ensure consistent design across all UI elements.
- Optimize for responsive design and accessibility.

Level 3 (Advanced)

Task 1: Custom Component Library

- Description: Build a fully functional web application using Django that includes user authentication (login, registration, and password reset). The application can be a blog, task manager, or e-commerce site.

Objectives:

- Build reusable and customizable components.
- Document using Storybook or similar tools.
- Ensure high accessibility standards (ARIA attributes).
- Publish as an NPM package for reusability.

Level 3 (Advanced)

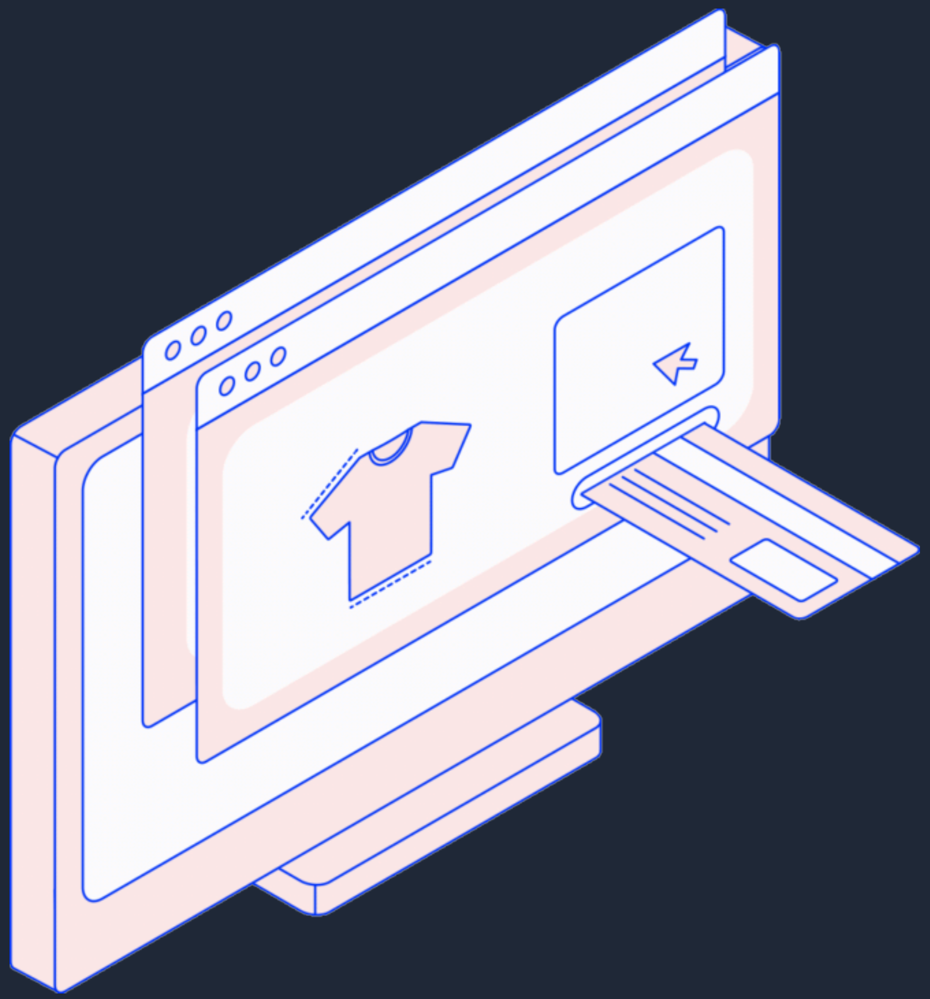
Task 2: Performance Optimization

- Optimize the performance of an existing project. Use tools like Google Lighthouse to identify performance bottlenecks and improve metrics like load time and interactivity.

Objectives:

- Optimize images using lazy loading and modern formats (e.g., WebP).
- Minify and bundle JavaScript/CSS files.
- Implement caching strategies for faster loading.
- Reduce render-blocking resources for better performance.

Level 3 (Advanced)



Task 3: Advanced Animations

- Create advanced animations using libraries like GSAP or Framer Motion. Implement interactive elements triggered by user actions (e.g., scroll-based animations or hover effects).

Objectives:

- Implement smooth and performant animations.
- Utilize timeline animations for complex sequences.
- Ensure animations are accessible and do not hinder usability.
- Optimize for minimal impact on performance.

How to Contact Us?

For additional information, kindly get in touch with our team.

in @codveda

✉ support@codveda.com

🌐 www.codveda.com