

# Frontend Developer Test Assignment

**Position:** React + Next.js Frontend Developer

**Platform (Dummy):** Artistly.com

**Assignment Type:** Individual | Work-from-anywhere

**Estimated Duration:** ~10-12 hours over 2 days

**Submission Deadline:** 4:00 PM IST, Wednesday, 25th June 2025

---

## 🌟 PROJECT OVERVIEW: Artistly.com (Dummy App for Assignment)

You are assigned to create a 3-4 page **functional and mobile-responsive web demo** of a fictional platform called **Artistly.com**, a Performing Artist Booking Platform.

Artistly.com is designed for **Event Planners** and **Artist Managers** to connect. Event planners can browse through artist profiles, shortlist their preferences, and raise booking/availability requests. Artist Managers can onboard artists, receive booking leads, and manage responses via a dashboard.

The actual backend and database logic is out of scope—for now, use static JSON files or mock APIs. Focus entirely on **frontend engineering quality** using **React + Next.js**, with best practices in routing, responsiveness, UI rendering, and folder structure.

---

## ✅ TASK OBJECTIVE

Build and host a working web app demo using:

- **Next.js (v13+ App Router)**
- **React Functional Components**
- **Tailwind CSS**
- **ShadCN or other open UI components**
- Free UI templates (suggested below)
- Host it for free on **Vercel.com**

You are **not** being evaluated on UI creativity or design polish. Instead, we are testing your **coding structure, responsiveness, code safety, performance optimization, form logic, data handling, and ability to compose a real-world UI using React + Next.js**.

---

## ◆ KEY PAGES TO BUILD (3-4 pages total)

### 1. Homepage (1 Page)

- Overview of the platform
- Dummy header, hero section, CTA to explore artists
- Add 3-4 artist category cards (e.g., Singers, Dancers, Speakers, DJs)
- Include basic navigation to other pages

### 2. Artist Listing Page (1 Page)

- Grid layout for artist cards with filtering controls
- Each card should display: Artist Name, Category, Price Range, Location, "Ask for Quote" CTA
- Filtering options: Category, Location, Price Range
- Dummy data from JSON or API file

#### Evaluation Criteria:

- Filter logic working
- Component reuse (Card, FilterBlock)
- Responsive layout (Grid to List on mobile)

### 3. Artist Onboarding Form (1 Page)

- Multi-section form for artist details:
  - Name, Bio, Category (multi-select dropdown with checkboxes)
  - Languages Spoken (multi-select with checkbox)
  - Fee Range (dropdown)
  - Profile Image Upload (optional)
  - Location (text input)
- Form validation (React Hook Form or Formik + Yup)
- Submits to mock API or local console

#### Evaluation Criteria:

- Use of controlled components
- Form validation & UX
- Dropdown + checkbox combo UI

### 4. Manager Dashboard Page (Optional, 1 Page)

- Simple table to show list of artist submissions
- Each row: Name, Category, City, Fee, Action button
- Static or simulated from mock API/local state

## Evaluation Criteria:

- Conditional rendering
  - Data mapping from source
  - Reusable Table component
- 

## DEPLOYMENT

1. Deploy the project on **Vercel.com** (free tier)
  2. Share:
    - **Live Preview Link** (vercel.app URL)
    - **Credentials of your Vercel account** (temporary access for code review only)
- 

## SUGGESTED UI TEMPLATE SOURCES

Feel free to use **any free UI Kit** to save time. Some great sources:

- <https://tailwindui.com> (samples available)
- <https://ui.shadcn.com/> (our preferred library)
- <https://vercel.com/templates>
- <https://flowbite.com>
- <https://tailblocks.cc>
- <https://www.creative-tim.com>
- <https://webpixels.io/>

**Remember:** Custom styling is optional—focus on **layout logic and interaction**.

---

## TECH STACK EXPECTATION

- Next.js (v13+, App Router preferred)
- React functional components with hooks
- Tailwind CSS (utility classes only, avoid inline styles)
- React Hook Form / Formik + Yup for forms
- HTML best practices for accessibility & SEO
- Working knowledge of state management using `useState` and `useContext`
- Handling side effects with `useEffect`
- Understanding of Next.js data fetching methods: `getStaticProps`, `getServerSideProps`
- Optional: use **dummy GraphQL endpoint** or `fetch()` to simulate data

---

## ✅ WHAT WE ARE LOOKING FOR

Area	Expectations
Code Structure	Neat folder hierarchy, modular component reuse
Responsiveness	Pages should be fully mobile responsive
Forms	Validated inputs, dropdowns with multi-checkbox selections
Listing	Filter logic should be visible & functional
Hosting	Deployed correctly on vercel.com
SEO	Proper head tags, metadata, image alt tags
Comments	Basic inline comments & function headers
React Skills	Demonstrated use of <code>useState</code> , <code>useContext</code> , <code>useEffect</code>
Data Handling	Demonstrated use of <code>getStaticProps</code> , <code>getServerSideProps</code> where applicable

---

## 📢 BONUS (Not Mandatory)

- Simple state management (e.g., `useContext`)
  - Lazy loading / suspense on route changes
  - Framer Motion for smooth page transitions
  - Theme support (light/dark)
- 

## 🕒 SUBMISSION INSTRUCTIONS

- Submit your assignment by **4:00 PM IST on Wednesday, 25th June 2025**
- Send your submission to: **gaurav@eventfulindia.com**
- Email Subject/Title: **[Frontend Dev] Next.js Test Assignment Submission**
- Include:
  - **Vercel Link** (live preview)
  - **Temporary Vercel Credentials** (for code review only)
- You will receive a **definite reply** post-evaluation
- Shortlisted developers will be invited for a **final Zoom Interview**

---

## **FINAL NOTES**

This assignment is designed to evaluate your:

- Frontend development logic and execution
- Understanding of component architecture and responsiveness
- Ability to use React + Next.js to simulate a real-world SaaS frontend

We are not testing you on:

- Graphic design skills
- Backend/database logic

All the best. Show us what you can do 