# React Developer – Machine Task

## Objective

Build a visually appealing and fully responsive **dashboard application** using **Next.js**, **Tailwind CSS**, and **Material UI**. This task will assess your design creativity, attention to detail, component structuring, and ability to work with third-party libraries.

---

## Core Requirements

### 1. Authentication UI (Login Page)

- Design a login form with **email** and **password** fields.
- No authentication logic needed – only UI.
- Use **Material UI** components.
- Ensure a **pixel-perfect layout** with **mobile responsiveness**.

---

### 2. Dashboard Layout

Upon dummy login (redirect), the user should land on a dashboard with the following:

#### a. Sidebar Navigation

- Responsive sidebar with icons.
- Include links to:
  Dashboard, Reports, Chat, and Settings.

#### b. Welcome Header

- Display a message like:
  Welcome, John! (use any dummy name).

### c. Stats Cards (Minimum 3)

Examples:

- Total Users
- Active Users
- Total Orders

Requirements:

- Use icons (preferably from Material UI).
- Must be **responsive**, visually **attractive**, and show good **design sense**.

### d. Responsive Data Table

- Fetch user data from:
  https://jsonplaceholder.typicode.com/users
- Display the following columns:
  - Name
  - Email
  - Phone
  - Company Name
- Add:
  - **Client-side search filter**
  - **Client-side pagination**
  - **Column sorting** (bonus)
  - **Transitions/animations** (bonus)

### e. Reports Section (Charts + Analytics)

- Build a **Reports** page with **visual analytics**.
- Use mock or real data to generate:
  - Line Chart or Bar Chart for daily/weekly/monthly trends
  - Pie or Doughnut Chart for distribution metrics (optional)
- Integrate:
  - **Date range filter (From - To)** to dynamically update chart data
  - Display **count/metric progression by date** based on selected range
- You may use any charting library: **Chart.js**, **Recharts**, **Nivo**, etc.

#### f. Chat Module (Basic UI)

- Integrate any open-source chat UI library (e.g., Chat UI kit, CometChat, or your own custom design).
- Dummy users/messages are fine.
- Must have:
  - Message list
  - Input box
  - Send button

---

## Timeline

Please aim to submit the task **by the end of Thursday**.
Let us know if you need additional time in advance.

---

## Evaluation Criteria

- Mobile-first & responsive design
- Visual polish & pixel-perfect layout
- Clean, reusable code structure
- Creative, thoughtful UI/UX decisions
- Effective usage of **Material UI** + **Tailwind CSS**
- Component breakdown and structure
- API integration
- Date filtering & dynamic chart updates
- Bonus features implementation

---

## Bonus Points (Optional)

- Dark mode toggle
- Smooth transitions using **framer-motion**
- Realtime data simulation (setInterval or mock websocket for stats)
- Column sorting in tables

---

## Tech Stack

- **Next.js** (mandatory)
- **Tailwind CSS** (mandatory)
- **Material UI** (mandatory)
- **Open-source chat library** (your choice)

- **Charting library** of your choice (with date filtering support)

---

## Submission Instructions

1. Push your completed project to a **public GitHub repository**.
2. Share the repo link with us.
3. Include a README.md with:
   - Setup instructions
   - npm install and npm run dev usage
   - A list of any bonus features implemented