# React Assignment: Advanced IoT Device Dashboard

## Project Overview

Build a Single Page Application using React.js for an IoT Device Dashboard. The dashboard will support real-time monitoring of temperature and humidity, threshold alerts, historical graphs, data export, and user authentication. The goal is to simulate a functional and professional IoT interface. Dashboard should be mobile responsive.

## Core Features

1. User Authentication  
- Simple login form  
 - Use: admin@example.com / password123  
- Form validation  
 - Required fields  
 - Basic email format check  
- On successful login, redirect to Dashboard  
- Logout functionality to clear session and return to login  
  
2. Dashboard with Tabs or Routing  
Include the following 3 tabs in the interface:  
- Live Monitor  
- Threshold Settings  
- Historical Data  
  
3. Live Monitor Tab  
- Fetch device list from demo API (demoapi)  
- Display data in cards or a table, including:  
 - Device Name  
 - Temperature (updates every 5 seconds)  
 - Humidity (updates every 5 seconds)  
 - ON/OFF toggle for device status  
 - Visual alerts (e.g., red indicator) if values exceed threshold  
- Show live updating graphs (temperature & humidity)  
 - Use libraries like Recharts or Chart.js  
  
4. Threshold Settings Tab  
- Set Min/Max temperature and humidity limits  
 - Per-device or global  
- Store thresholds in localStorage or mock API  
- Thresholds should be respected in Live Monitor alerts  
  
5. Historical Data Tab  
- Display historical temperature & humidity graph  
 - Use dummy data (timestamp + random values)  
- Filters:  
 - Device selector  
 - Time range (e.g., Last 24h, Last 7 days, 3 Months)  
- CSV export of filtered data  
 - Include: Timestamp, Temperature, Humidity

## Technical Requirements

- React.js (functional components with hooks)  
- Routing or Tab-based UI  
- Local state management: useState, useContext (bonus: Redux)  
- Charting: Recharts / Chart.js / Victory  
- Data simulation using setInterval and random number generator  
- Use of demo api as backend

## Deliverables

- GitHub repo link  
- README with:  
 - Setup instructions  
 - Technology stack used  
- Live hosted link (if available)  
- Sample CSV downloaded

## Time Limit

3 Days