

# React Developer – Timer & Step Generator for Digital Watch Simulator

## Project Description

We are developing a simulator for a digital LCD watch system. The goal is to implement a **Timer** and **Sensor (step + state generator)** in React.js, with a simple dashboard to visualize their outputs.

The dashboard will include controls (start/stop, speed, archetype, duration), visualizations (steps/hour and steps/day graphs), and a log panel for interrupts. Code should be **modular, well-documented, and ready for integration** with other modules (calculation, controller, watch face) later.

---

## Milestone 1 – Dashboard Scaffold

Build a functional dashboard layout with Control, Visualization, and Log panels.

### Details:

- Implement Control panel with:
  - Start/Stop button
  - Speed slider (x1–x1000, log scale)
  - Start date selector (direct + calendar input)
  - Duration selector (1–1000 days)
  - Archetype selector (JSON presets)
  - Progress bar (simulation progress)
  - Button to export generated data to JSON
- Visualization panel:
  - Graph 1 = steps per hour (7 days, padded if needed) with bar + smoothed line (7h Hann window)
  - Graph 2 = steps per day (365 days, padded if needed) with bar plot

- Log window (scrollable): shows interrupts (NEW\_SECOND, NEW\_MINUTE, NEW\_STEP, NEW\_STATE) with timestamps.

**Acceptance Criteria:**

- All controls are interactive and update UI state.
  - Graphs render correctly with placeholder/random data.
  - Log updates in real time.
- 

## **Milestone 2 – Timer Module**

**Abstract:** Implement a time engine with adjustable playback speed.

**Details:**

- Timer emits interrupts: NEW\_SECOND and NEW\_MINUTE (flags only).
- Speed adjustable via slider (x1–x1000).
- Provide API for querying current simulated time: year, month, day, hour, minute, second, millisecond, day of week.

**Acceptance Criteria:**

- Slider changes interrupt rate dynamically.
  - Query API returns consistent simulated time.
  - Demo run of 1 simulated day completes without desync.
- 

## **Milestone 3 – Step Generator (30%)**

**Abstract:** Implement step generator producing NEW\_STEP events with archetype presets.

**Details:**

- Generate steps according to selected archetype JSON.
- Support walking (120 steps/min) and jogging (180 steps/min).

- Implement 4 archetypes: office worker, flexible worker, shift worker, athlete.
- Steps distributed according to day/night schedules defined in archetypes.

**Acceptance Criteria:**

- Changing archetype updates visible daily step patterns in graphs.
  - Step counts scale correctly with simulation speed.
  - No missing or duplicate step events at high playback speeds.
- 

## **Milestone 4 – State Generator (20%)**

**Abstract:** Add simulated user/device states alongside steps.

**Details:**

- Sensor emits NEW\_STATE events:
  - **SLEEP** → no steps, user inactive
  - **BACKGROUND** → steps emitted, user inactive
  - **IDLE** → no steps, user actively interacts with device
  - **ACTIVE** → steps emitted, user actively interacts with device
- States depend on archetype and time of day.

**Acceptance Criteria:**

- Log panel shows state changes in sync with steps and time.
  - State transitions follow archetype schedules (e.g., SLEEP at night, ACTIVE in commuting periods).
- 

## **Requirements**

- Strong React.js skills.

- Experience with charting libraries (Recharts preferred).
  - Familiarity with event-driven simulations is a plus.
  - Code must be modular, documented, and structured for integration with future modules.
- 

## Scope of Work

### 1. Dashboard (React.js)

- **Control panel** with:
  - Start/Stop button
  - Simulation speed slider (x1–x1000)
  - Start date selector (direct and calendar input)
  - Duration selector (number of full days 1–1000)
  - Person archetype selector (reads from JSON presets)
  - Progress bar (simulation completion)
  - Buttons to save generated data to JSON files
- **Visualization panel** with:
  - Graph 1: steps per hour (7 days padded if needed) → bar plot + smoothed line (7-hour Hann window)
  - Graph 2: steps per day (365 days padded if needed) → bar plot
- **Log window** (scrollable) showing sequence of interrupts: NEW\_SECOND, NEW\_MINUTE, NEW\_STEP, NEW\_STATE with timestamps.

### 2. Timer module

- Adapts to speed rate from Control panel.
- Emits NEW\_SECOND and NEW\_MINUTE interrupts (flags only).

- Provides API to query current simulated time (year, month, day, hour, minute, second, ms, day of week).

### 3. **Sensor (Step Generator) module**

- Generates NEW\_STEP interrupts according to selected archetype.
  - Supports different cadences (e.g., 120 steps/min walking, 180 jogging).
  - Archetypes: office worker, flexible worker, shift worker, athlete — provide different parameter presets for random step counts generation
  - Generates NEW\_STATE interrupts:
    - **SLEEP** → no steps, user inactive
    - **BACKGROUND** → steps emitted, user inactive
    - **IDLE** → no steps, user interacts with device
    - **ACTIVE** → steps emitted, user interacts with device
- 

## **Deliverables**

- **Source Code** (React.js, modular, in GitHub repo):
    - Timer.js (adjustable speed)
    - Sensor.js (step + state generator with archetypes)
    - Dashboard React components
  - **Documentation:**
    - API between Timer, Sensor, and downstream modules
    - Setup and run instructions
- 

## **Milestones & Acceptance Criteria**

### **Milestone 1 – Dashboard (30%)**

- Functional dashboard with 3 panels (Control, Visualization, Log).
- Control panel fully interactive.
- Visualization panel: steps/hour & steps/day graphs (Recharts or similar).
- Log window shows and scrolls incoming interrupt events.

✓ Acceptance: Run demo, adjust controls, see updates reflected in graphs & log.

### **Milestone 2 – Timer (20%)**

- Timer emits NEW\_SECOND and NEW\_MINUTE.
- Adjustable speed (x1–x1000).
- Query API for current simulated time.

✓ Acceptance: Change slider → interrupt frequency updates correctly.

### **Milestone 3 – Step Generator (30%)**

- Step generator produces NEW\_STEP events parameterized by archetype JSON.
- Walking/jogging cadences supported.
- Archetypes affect day/night distribution of steps.

✓ Acceptance: Switching archetype changes visible step patterns in graphs.

### **Milestone 4 – State Generator (20%)**

- Sensor emits NEW\_STATE events: SLEEP, BACKGROUND, IDLE, ACTIVE.
- States switch realistically according to archetype/daytime.

✓ Acceptance: Log panel shows state changes alongside steps/time interrupts.

---

## **Requirements**

- Strong React.js skills.

- Experience with charting libraries (Recharts preferred).
- Familiarity with simulation/event-driven systems is a plus.
- Code must be modular and documented for handoff to other developers.