FRONTEND INTERN ASSIGNMENT

Final Assignment Document
Deadline: 48 Hours
Tech Stack: React.js, REST API, State Management, Tailwind/Bootstrap, JSON

Objective

Build a responsive and interactive Warehouse Robot Dashboard to demonstrate skills in UI/UX design, component architecture, state management, API integration, and real-time updates. This assignment evaluates both design and development capabilities.

Pages to Build

1. Login / Signup Page

- **Implement a basic form with validation.**
- **Store login state in global state management.**
- **Do not use localStorage for authentication.**

2. Home Dashboard

Display an overview including the following:

- **Total Bots**
- **Active Tasks**
- **Idle Bots**
- **Bots in Error**
- **Pending Tasks**
  Layout and design are flexible.

3. Bot Status Page (Core Requirement)

Display 10 bots as cards.
Each card must show:

- **Battery (%)**
- **Status (idle, busy, charging, error)**
- **Current Task**
- **Speed**
- **Last Updated Time**

Bot data must auto-update every 10 seconds using mock API or random simulated values.

4. Task Allocation Page

Allow the user to create a task with fields (e.g., pickup, drop, priority, comments).
On submit:

- **Add the task to global state.**
- **Display it instantly in the Task Queue page.**

5. Task Queue Page

- **Show all pending tasks from global state.**
- **Automatically remove one task every 3 seconds (simulate task assignment).**
- **Tests timers and queue logic.**

6. Analytics Page

Create charts for metrics of your choice.
In the design document, explain why you chose those metrics and charts.
You may use any React chart library.

7. Map Page (Bonus)

- **User uploads a warehouse layout as an SVG file.**
- **Render the SVG directly.**
- **Show bots as moving circles at simulated coordinates.**
- **No accuracy requirements. Movement can be random.**

Deliverables

1. GitHub/GitLab Repository

Must include:

- **Clean folder structure**
- **Proper commit messages**
- **README containing:**
    - **How to run the project**
    - **Tech stack**
    - **Component architecture**
    - **Data flow explanation**
    - **State management reasoning**
    - **Figma link (if bonus completed)**

2. Design Document (2–3 Pages PDF)

Include the following:

- **System architecture**
- **UI/UX decisions**
- **Data flow**
- **Key assumptions**
- **Trade-offs**
- **State management design**
- **Improvements you would make with more time**

Bonus Requirements

1. **Figma Design (2–3 screens)**
    - **Home Dashboard**
    - **Bot Status Page**
    - **Task Allocation Page**
    - **Focus on clarity, alignment, spacing, and component hierarchy.**
    - **Add Figma link in README.**
2. **Next .js**
3. **Three.js visualization (simple bot movement)**

Submission Format

Submit the following:

1. **GitHub/GitLab Repository Link**
2. **PDF Design Document**

Evaluation & Testing Criteria

1. **Code quality and readability**
2. **Component and folder architecture**
3. **Correct use of state management**
4. **UI/UX clarity and overall visual presentation**
5. **Mobile and desktop responsiveness**
6. **Functional completeness and correctness of features**
7. **Documentation quality (README + Design Document)**
8. **Bonus work and creativity (Figma, Next.js, Three.js, SVG Map)**
