Frontend Developer Task: Home Inventory UI (Homebox API)

Goal

Build a small frontend that matches the provided **Figma/wireframes** and integrates with a running **Homebox-compatible API** (already deployed by us).

What we provide

- **Figma / wireframes** (pixel-perfect target)
- **API Base URL:** `http://4.213.57.100:3100/`
- **Swagger UI:** `http://4.213.57.100:3100/swagger/index.html`
    - Use Swagger's **"Try it out"** to discover request/response shapes and even copy curl snippets.
- **Figma Link:** https://www.figma.com/design/2wOGr4GxmJHwkEgdCMD6WA/Inventory-Management?node-id=0-1&t=gGnwohAw4huioeeU-1

Critical rule: no credentials provided

You must create your own test account by calling the API's **registration endpoint** (no signup page required).

Self-registration instructions (required)

1. Open Swagger UI
2. Find the **User Register** endpoint under the **User** section
3. Use **Try it out** to register a new user (Or you can use it in Postman too, if it is comfortable for you there)
4. Use those credentials to log in to your frontend

Pages to implement (3–4 pages)

Page 1 — Login (required)

- Username/email + password fields
- Submit → call login endpoint (`/api/v1/users/login`)
- Store token, redirect to Inventory
- Handle loading + error states
- (Optional) small helper text like: “No account? Register via API (see Swagger).”

Page 2 — Inventory (Items List) (required)

- Fetch and list items (use Swagger to confirm the endpoint + query params)
- Search box (server-side if supported; otherwise client-side)
- Row click → Item Details
- Loading skeleton / empty state / error + retry

Page 3 — Item Details (required)

- Fetch item details by ID
- Show fields according to Figma (name, location, quantity, labels, notes, etc.)
- Optional: Edit flow (drawer/modal) if included in design

Page 4 — Locations (optional but recommended)

- Fetch locations and render tree/list view
- Selecting a location shows a details panel and/or items within it (based on Figma)
- Optional: create/edit location if in design

Technical requirements (React mandatory)

Required

- **React** (React 18+)
- SPA routing with **React Router** *or* a React meta-framework:
    - Allowed: **Vite + React**, **Next.js (React)**
    - Not allowed: Vue / Angular / Svelte / other non-React frameworks
- A small **API client layer** (e.g., `src/api/`) with:
    - `baseUrl` from env (e.g., `VITE_API_BASE_URL` or `NEXT_PUBLIC_API_BASE_URL`)
    - automatic `Authorization` header attachment after login
- Auth:
    - login page uses the API login endpoint
    - token stored safely (ok: memory + localStorage) and reused for API calls
- Responsive UI (desktop + mobile)
- Basic accessibility: labels, focus states, keyboard navigation, semantic HTML

Recommended (optional, but good signals)

- **TypeScript**
- React Query / TanStack Query or SWR for caching and loading states
- Simple component structure (pages/components/hooks/api)
- Error boundaries / centralized error handling

Deliverables

- Git repo (or zip)
- `README.md` with:
    - how to run
    - env vars (e.g. `API_BASE_URL`)
    - how you handled auth
    - any tradeoffs / next steps

- **Test user credentials** you created via API:
    - Put them in `SUBMISSION.md` **and** paste into the submission form fields (below)

Submission Guidelines

- You can submit and edit the form anytime within **9:00 AM, 12 January, Monday**
- Make sure you have given the following usernames access to your repository
  iftekharzeeon
  akib1689
  Erfan8048
- Since you can submit/edit the form multiple times within the mentioned time, if any required field has to be empty now, keep it *N/A*
- Please make sure you have submit the form and given access to the repository within the mentioned time. Otherwise you may not be considered for evaluation.
- You will need to create a recording video to showcase your work & upload it to youtube as unlisted video/ google drive. Just record the screen while you show us what you have done. Better to wrap the recording within 3-4 minutes, no need of anything extra or formal, just to the point. You can speak in Bangla for the recording.
- Submission Link : https://forms.gle/Dtb31bL1jZdwBqeT8

Evaluation Metrics

- UI accuracy vs Figma
- Correct API integration + auth flow
- UX states (loading/empty/error)
- Code quality, structure, readability
- Responsiveness + accessibility basics