

Erino SDE Internship Assignment — Lead Management System

Objective

You need to build a **basic Lead Management System**.

This system should allow users to **register/login**, manage leads with **CRUD operations**, view leads in a **grid with pagination and filters**, and run fully deployed on the web.

The assignment is designed to test your **end-to-end engineering ability**.

Time limit: 72 hours from when you receive this document.

Evaluation Checklist:

Your assignment will be rejected if **any item below is missing or fails**:

- JWT auth with httpOnly cookies (no localStorage).
- CRUD for leads with correct status codes.
- Server-side pagination and filters should be working.
- Create/Edit/Delete reflect correctly in the UI.
- Unauthorized requests return 401.
- Fully deployed (frontend + backend + DB).

Very Important:

If your project is not hosted and publicly accessible, it will not be evaluated.

Good luck, and ship like a pro 🚀 waiting to meet you in the next round.

Requirements

Tech Stack:

- Frontend: ReactJS
- Backend: Express
- Database: Your choice

1. Authentication

- Users must be able to **register, login, logout, and fetch current user**.
- Authentication must be done using **JWT** stored inside an **httpOnly cookie** (not localStorage).
- Passwords must be hashed (e.g., bcrypt).
- API routes must return correct HTTP codes (**201 Created, 200 OK, 401 Unauthorized**, etc.).
- **Unauthenticated requests** to lead APIs must return **401**.

2. Leads (CRUD)

Each Lead must have the following fields:

- **id**
- **first_name, last_name**
- **email** (unique)
- **phone**
- **company**
- **city**
- **state**
- **source** (enum: **website, facebook_ads, google_ads, referral, events, other**)
- **status** (enum: **new, contacted, qualified, lost, won**)
- **score** (integer 0–100)
- **lead_value** (number)
- **last_activity_at** (datetime, nullable)
- **is_qualified** (boolean, default false)
- **created_at, updated_at**

APIs required:

- `POST /leads` → create lead
- `GET /leads` → list with pagination & filters
- `GET /leads/:id` → fetch single lead
- `PUT /leads/:id` → update lead
- `DELETE /leads/:id` → delete lead

3. Server-Side Pagination

- `GET /leads` must support:
 - `page` (default 1)
 - `limit` (default 20, max 100)

Response format:

```
{
  "data": [/* Leads */],
  "page": 2,
  "limit": 20,
  "total": 146,
  "totalPages": 8
}
```

4. Server-Side Filtering

Filters must be applied **in the backend**. Multiple filters combine with **AND** logic.

Operators required:

- **String fields** (`email`, `company`, `city`) → `equals`, `contains`
- **Enums** (`status`, `source`) → `equals`, `in`
- **Numbers** (`score`, `lead_value`) → `equals`, `gt`, `lt`, `between`
- **Dates** (`created_at`, `last_activity_at`) → `on`, `before`, `after`, `between`
- **Boolean** (`is_qualified`) → `equals`

5. Frontend (ReactJS)

- Must be built with **ReactJS**.
- Must use **AG Grid** or an equivalent grid, table of your choice.
- Pages required:
 - Login / Register
 - Leads List (grid with server-side pagination & filters)
 - Lead Create/Edit form

6. Hosting

- **Frontend** must be deployed on **Vercel (or equivalent)**.
- **Backend** must be deployed on **Render/Railway/Fly/Heroku/Supabase, etc.**

Make sure:

- Both frontend and backend must be publicly accessible.
- We should be able to test end-to-end without local setup.
- **And with a seed data of 100+ leads in a test user account.**
- Test user credentials work on deployed app.
- Leads should be visible immediately after login.

7. Submission

- After successfully completing this assignment, email to **akansha@erino.io**.
 - a. Email **MUST** include all of the below:
 - i. Live app links (frontend + backend)
 - ii. GitHub Project Repo Link
 - iii. Test user credentials which should definitely work on your deployed app.
 - Leads should be visible immediately after login.
 - iv. Reflection (5–10 lines):
 - What was the hardest part in this assignment?
 - What did you learn?
 - What would you have improved if you had 1 more day?
-

