



NxtJob Assessment

 Due Date	@January 19, 2025 8:00 PM
 Status	In progress

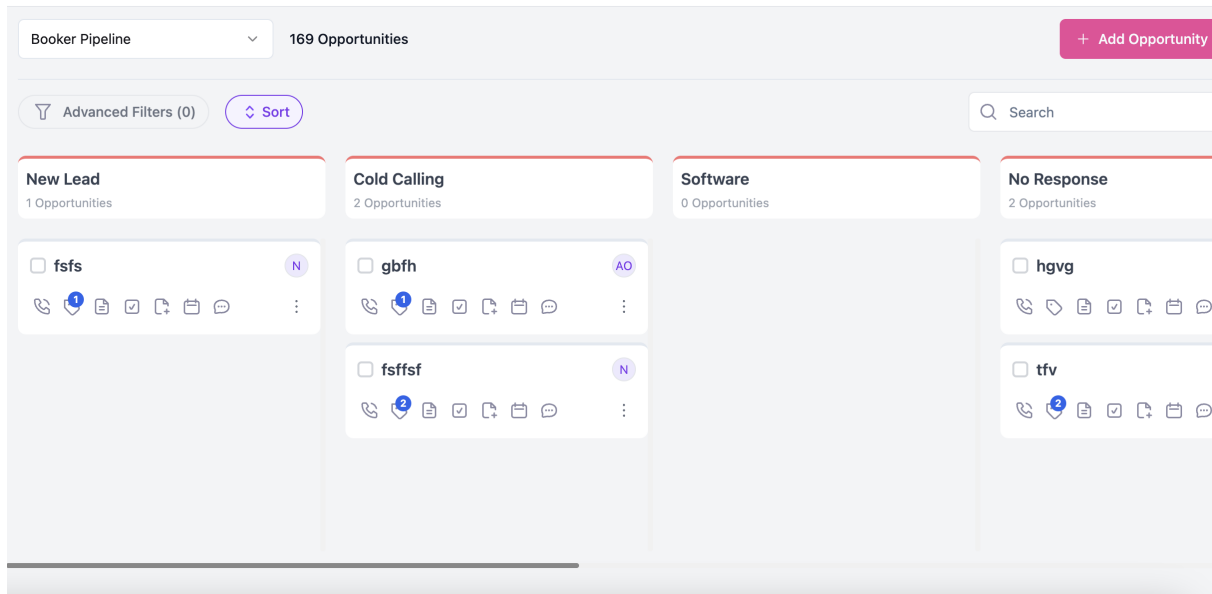
What is it?

We are developing a **Backend System for our ERP Platform**. The goal of this task is for candidates to build a backend that integrates seamlessly with our existing frontend, providing APIs for key operations as inferred from the provided frontend screenshot.

User Story

1. **As a Salesperson**, I want to add a new lead from various sources (e.g., Facebook, LinkedIn, or manual entry) so that I can track potential opportunities effectively.
 2. **As a Salesperson**, I want to move leads between different stages (e.g., New Lead, No Response, In Progress) so that I can update their status as they progress through the pipeline.
 3. **As a Salesperson**, I want to search for leads by name or other attributes so that I can quickly locate specific opportunities.
 4. **As a Salesperson**, I want to filter leads by their source (e.g., Facebook, LinkedIn) or the owner assigned to them so that I can focus on specific subsets of opportunities.
 5. **As a Sales Manager**, I want to change the owner assigned to a lead so that I can reassign responsibilities as needed.
-

Diagram



High-Level Design (HLD)

- **Frontend** interacts with **Cloudflare Workers Backend** via RESTful APIs.
- **Database** (NeonDB) is accessed using **Drizzle ORM** for CRUD operations.

API Requirements

Example: Fetch All Leads

Endpoint:

GET	/leads
-----	--------

Request Parameters:

Key	Description	Type	Notes
page	Current Page	number	1
limit	Results per page	number	20 (default)
source	Lead source	string	Optional (e.g., Facebook, LinkedIn)
owner	Assigned owner	string	Optional

Success Response:

```
{
  "leads": [],
  "count": 0,
  "totalPages": 2
}
```

Additional APIs:

1. Add Lead

- POST /leads
- Request Body: `{ "name": "John Doe", "source": "Facebook", "owner": "1234" }`

2. Update Lead Stage

- PATCH /leads/{leadId}/stage
- Request Body: `{ "stage": "In Progress" }`

3. Search Leads

- GET /leads?query=searchTerm
- Query Parameter: `query` (string)

4. Filter Leads

- GET /leads?source=Facebook&owner=1234
- Query Parameters: `source`, `owner`

5. Change Lead Owner

- PATCH /leads/{leadId}/owner
- Request Body: `{ "owner": "5678" }`

Tasks

Setup and Environment

1. Cloudflare Worker, NeonDb
2. Project Setup

API Development

1.

Testing

1.

Deployment

1.

Deployment Plan

1. Setup Cloudflare Workers:

- Use the `wrangler` CLI to initialize and deploy the project.

2. Database Setup:

- Use Drizzle ORM migration scripts to initialize the schema in NeonDB.

3. Deployment Steps:

- Commit code to a Github.
- Use `wrangler publish` to deploy the Cloudflare Worker.
- Verify deployment logs for errors.

4. Testing Post-Deployment:

- Test APIs using tools like Postman.

5. Setup CI/CD (Optional)

Milestones and Timelines

1. **Database Setup and Schema Definition**

2. **API Development**

3. **Deployment and Testing**

4. **Final Submission:** Deadline is 8 PM today