Frontend Internship Technical Assessment

NexCell Solutions Ltd

Deadline: Tuesday 20th January 2026 at 5:00 PM GMT

Overview

Build a small lead management dashboard using Next.js. This assessment tests your ability to create clean UIs, handle data properly, and structure code like a professional engineer.

We value clarity and correctness over complexity.

Time expectations

- Expected effort: 1.5 to 2 hours
- Deadline: Tuesday 20th January 2026 at 5:00 PM GMT
- AI tools and documentation are allowed and encouraged
- You must understand and be able to explain every line of code you submit

Important: We are not expecting you to spend multiple days on this. The deadline exists to give you flexibility to fit this into your schedule, not to increase workload.

What you will build

A simple lead management web app with the following stack:

- Next.js (App Router)
- React + TypeScript
- TailwindCSS
- shadcn/ui (at least 2-3 components)

Requirements

1. Login page (/login)

Must have:

- Email input field
- Password input field
- "Sign in" button
- On button click, redirect to /dashboard

Notes:

- No real authentication required
- You can use localStorage or session storage to track "logged in" state
- Basic form validation is optional but appreciated

2. Dashboard page (/dashboard)

Must have:

a) Header section

- Page title (e.g., "Lead Dashboard")
- User menu or profile icon (can be static/non-functional)

b) Leads table/list

Display leads with these columns:

- Name
- Email
- Status
- Created date (formatted nicely)

c) Filters

- Search input: filters leads by name OR email
- Status dropdown: filters by status (e.g., New, Contacted, Qualified, Lost)

d) Lead details view

- Clicking a table row opens a modal or side panel
- Shows all lead information including notes field

e) Analytics section

- Show total lead count
- Show count per status (can be simple cards or a basic chart)

3. Data requirements

Choose ONE option:

Option A: Local JSON file (recommended)

- Create data/leads.json with exactly 20 sample leads
- Load and use this data in your components
- Treat it like API data (async loading is good practice)

Option B: Next.js API route

- Create app/api/leads/route.ts
- Return the same lead structure
- Call this endpoint from your frontend

Lead data structure (each lead must include):

```
{
  id: string;
  name: string;
  email: string;
  status: "New" | "Contacted" | "Qualified" | "Lost";
  createdAt: string; // ISO date string
  notes: string;
}
```

UI and code quality expectations

Must have:

- Clean layout with proper spacing
- Responsive design (works on desktop and mobile)
- Empty states (e.g., "No leads found" when filters return nothing)
- Use Tailwind classes (no inline styles)
- At least 2-3 shadcn/ui components
- Proper TypeScript types (no any types)

Nice to have (optional, only if time permits):

- Loading states
- Error handling
- Smooth animations/transitions
- TanStack Query for data fetching
- Dark mode toggle

Required deliverables

You must submit all three of these:

- Code (GitHub repo link OR zip file)
- SUBMISSION.md (using template below)
- SUBMISSION.json (using schema below)

File structure

```
your-project/
├── app/                    # Next.js app directory
│   ├── login/
│   ├── dashboard/
│   └── api/                # (if using Option B)
├── components/             # React components
├── data/
│   └── leads.json          # (if using Option A)
├── lib/                    # Utils, types, etc.
├── SUBMISSION.md           # Required
├── SUBMISSION.json         # Required
├── README.md
└── package.json
```

SUBMISSION.md template

Create a file called **SUBMISSION.md** and fill it out exactly as shown:

````
# Frontend Assessment Submission

## Candidate information
**Name:**
**Email:**
**Time spent:** X minutes

---

## What I delivered
(1-2 sentences describing what you built)

---

## How to run locally
```bash
# List the exact commands here
npm install
npm run dev
```

---

## Technical decisions

**Data approach:**
(Option A or Option B, and why you chose it)

**Key UI decisions:**
-
-
-

**How filtering works:**
(2-3 sentences explaining your search and status filter implementation)

**Shadcn/ui components used:**
-
-

---

## Reflection

**If I had 1 more hour, I would:**


**One thing I kept intentionally simple and why:**


**Biggest risk or known bug:**
````

```
(Be honest - we appreciate self-awareness)

---

## AI usage
**Did you use AI tools?** (Yes/No)
**If yes, which ones and for what?**
(e.g., "ChatGPT for TypeScript types", "GitHub Copilot for boilerplate")
```

SUBMISSION.json schema

Create a file called **SUBMISSION.json** with this exact structure:

```
{
  "candidate": {
    "name": "",
    "email": "",
    "github_username": "",
    "time_spent_minutes": 0
  },
  "stack": {
    "nextjs_version": "",
    "typescript": true,
    "tailwind": true,
    "shadcn_ui": true,
    "data_loading_method": "local-json | next-api"
  },
  "features_completed": {
    "login_page": true,
    "dashboard_layout": true,
    "leads_table": true,
    "search_filter": true,
    "status_filter": true,
    "details_view_modal": true,
    "analytics_section": true,
    "responsive_design": true,
    "empty_states": true
  },
  "optional_features": {
    "loading_states": false,
    "error_handling": false,
    "tanstack_query": false,
    "animations": false,
    "dark_mode": false
  },
  "file_paths": {
    "login_route": "",
    "dashboard_route": "",
    "data_file": "",
    "api_route": ""
  },
  "self_assessment": {
    "confidence_level": "1-10",
```

```
    "biggest_challenge": "",
    "proudest_aspect": ""
  }
}
```

Instructions:

- Set boolean values to true or false based on what you completed
- Leave unused paths as empty strings ""
- Be honest in your self-assessment

Evaluation criteria

Your submission will be evaluated in this order:

1. Completeness (40%) - Does it meet all requirements and run without errors?
2. Code quality (30%) - Clean structure, proper TypeScript, organized components
3. UI/UX (20%) - Clean design, responsive layout, good user experience
4. Communication (10%) - How well you explain your decisions in SUBMISSION.md

We care more about:

- Code that works and is easy to understand
- Honest self-reflection and clear communication
- Making intentional trade-offs and explaining them

We care less about:

- Fancy features beyond requirements
- Perfect pixel-pushing
- Complex state management for simple use cases

Rules and guidelines

Allowed:

- AI tools (ChatGPT, GitHub Copilot, etc.)
- Documentation and Stack Overflow
- shadcn/ui documentation
- Any learning resources

Not allowed:

- Copying entire projects/templates without understanding
- Having someone else write your code
- Submitting code you cannot explain

Important:

- You will be asked to explain your implementation in a follow-up interview
- We can tell when code is copied vs. understood
- It's better to submit something simple that works than something complex that breaks

Submission instructions

How to submit:

5. Push your code to a public GitHub repository (preferred)
   OR
   Create a zip file of your project (exclude node_modules)
6. Ensure SUBMISSION.md and SUBMISSION.json are in the root directory
7. Email your submission to: internships@nexcellsolutions.com
   Subject line: Frontend Assessment - [Your Name]

Email must include:

- GitHub repo link OR zip file attachment
- Your full name
- Your phone number

Deadline: Tuesday 20 January 2026 at 5:00 PM GMT

Late submissions will not be accepted unless you have communicated with us beforehand.

Questions?

If you have questions about the requirements, please email contact@nexcellsolutions.com before Monday 19 January 2026 at 12:00 PM GMT.

We will respond to clarification questions within 24 hours.

Sample data

To help you get started, here's a sample lead object:

```
{
  "id": "lead_001",
  "name": "Sarah Johnson",
  "email": "sarah.johnson@example.com",
  "status": "Qualified",
```

```
  "createdAt": "2025-12-15T10:30:00Z",
  "notes": "Interested in an enterprise plan. Follow up next week."
}
```

You should create 19 more similar objects for your leads.json file.

Good luck!

We're excited to see what you build. Remember:

- Keep it simple
- Make it work
- Explain your thinking

This is your opportunity to show us how you approach problems and write code. We're rooting for you!

NexCell Solutions Ltd
London, United Kingdom
www.nexcellsolutions.com