Start of main content

Junior Developer

SPACEAI APP

You applied to this position on October 25, 2025.

View job

Job details

Senior/Junior MERN Stack Developers(1-5 years Experience)

Location: Kochi, Kerala

Job type: Full-time

&nbsp;

Today

Hi AKSHAI, thanks for applying to our Senior/Junior MERN Stack Developers(1-5 years Experience) job. As we review applications, you're welcome to reach out about details in your work or education history you'd like us to consider, such as transferable skills and experience. Thanks, the team at SPACEAI APP

"WELCOME TO SPACEAI"

This is a pre-requisite for the job interview.Any applicants failing to send the assignment within stipulated timeframe are not eligible for an interview call.

Submission of tasks will be evaluated on first come first basis and will be called for an interview.

Good Luck

We look forward to receive the assignments ASAP

Task: Employee Directory Application

Backend Requirements

1. Node.js Server with GraphQL

Set up Apollo Server

Connect to MongoDB database

2. Data Schema

Employee Type: id, name, position, department, salary

Company Department Type: id, name, floor

3. GraphQL Operations

Queries:

▪ getAllEmployees (returns name + position)

▪ getEmployeeDetails(id) (returns all fields)

▪ getEmployeesByDepartment(department)

Mutation: addEmployee(name, position, department, salary)

4. Database

Use MongoDB Node.js driver

Seed initial data (minimum 5 employees across 3 departments)

Frontend Requirements (Next.js)

1. Pages

Home Page (/):

▪ List employees (name + position) in a table

▪ Department filter dropdown

▪ "Add New Employee" button o Employee Detail Page (/employee/[id]):

▪ Show all employee details

▪ Back button to home page

2. Data Handling

Apollo Client for GraphQL operations

Implement loading states

3. UI Components

Form for adding new employees (with validation)

Responsive design using Tailwind CSS

Interactive department filter

Technical Requirements

• Backend

Apollo Server 4

MongoDB driver (no Mongoose)

Proper error handling (e.g., invalid IDs)

• Frontend

Next.js App Router

Apollo Client with cache configuration

Client-side form validation

• Styling

Tailwind CSS components

Mobile-responsive layout

Evaluation Criteria

1. Complete functional implementation

2. Clean component structure

3. Efficient GraphQL queries

4. Proper state management

5. Error handling (frontend + backend)

6. UI consistency

Deliverables

1. Working Node.js GraphQL API

2. Functional Next.js frontend

3. GraphQL schema definition

4. MongoDB connection logic 5. Styled UI components

Key Differences from Original Task:

• Adds mutations (create operation)

• Includes data filtering implementation

• Requires relationship handling (employee-department)

• Implements form validation

• Features more complex UI interactions

Task Duration:-3 Days

This maintains the core technical stack (Node.js, GraphQL, MongoDB, Next.js, Apollo) while introducing new challenges like data relationships, mutations, and client-side filtering - all achievable within a 1-hour timeframe for an intermediate developer.

Submission Format:

Github Repo

Jini Raju

1:09 PM

Write your message