## **Job Assignment: Node.js Developer**

### **Objective:**

Develop a full-featured web application using **Node.js**, **Express.js**, **sequelize**. Include **user authentication**, **role-based access control**, and **task/user management** with all relevant relationships and validations.

## **Requirements**

### **1. Web Application Features**

#### **a. User Authentication**

* Implement secure **user registration and login** using **email and password**.
* Use **JWT (JSON Web Token)** or **Passport.js** with sessions.
* Implement **role-based access control (RBAC)** middleware.
* Roles: Super-admin, Admin, Manager

#### **b. Navigation Menus (API routes or frontend views)**

##### **🔹 Task Section**

**Task Add Menu:**

* Fields:  
  + Task Name (string)
  + Task Description (string)
  + Task Type (multi-select): a-task, b-task, c-task, d-task, e-task
  + Task Created By (store role of the user who created)
  + Task Start Date / End Date (Date fields)
* Endpoint: POST /api/tasks

**Task View Menu:**

* List all tasks with full details
* Endpoint: GET /api/tasks

##### **🔹 User Section**

**User Add Menu:**

* Fields:  
  + Name (string)
  + Image (file upload via Multer)
  + Email (string)
  + Gender (radio: Male, Female, Others)
  + Role (Super-admin, Admin, Manager) – checkboxes
  + Created By (store role of creator)
* Endpoint: POST /api/users

**User List Menu:**

* Endpoint: GET /api/users

#### **c. Roles and Permissions Table**

| Role | add | update | delete | view |
| --- | --- | --- | --- | --- |
| Super-admin |  |  | User,Task | User,Task |
| Admin | User,Task | User,Task | User,Task | User,Task |
| Manager | Task |  | Task | User |

#### 

### **2. Tasks**

#### **a. Setup**

1. Initialize project using express-generator or custom setup.
2. Define **schemas** and **models**:  
   * User
   * Task

#### **b. Authentication & Authorization**

* Use **JWT or Passport.js**
* Protect API routes with middleware that checks:  
  + If the user is authenticated
  + If the user has permission based on role

### **✅ Deliverables**

1. Full **Node.js (Express.js)** app
2. All required models with proper schema design
3. API endpoints for User and Task management
4. Authentication (Login/Register)
5. Role-based access control (RBAC)
6. File upload (Multer for user image)
7. Detailed **README.md** with:  
   * Setup instructions
   * API docs
   * Role descriptions and access rules

🕒 Timeline

2 hours

📧 Contact

Questions?

📩 Email: tech@krishivikas.com