**User Management System**

**Overview**

A Node.js-based user management system with role-based authentication using Express, MongoDB, and JWT. This system allows user registration, authentication, role-based access control, and management of tasks and farms.

**Features**

- User Authentication (Signup, Login, JWT Authentication)

- Role-Based Access Control (FarmAdmin, FarmManager, FarmTechnician, EndUser)

- Task Management (Create, Assign, Update, View Tasks)

- Farm Management (Create, View Farms)

- Secure API Endpoints with JWT Middleware

- MongoDB Database Integration

**Technologies Used**

- Node.js

- Express.js

- MongoDB (Mongoose ODM)

- JWT (JSON Web Token) Authentication

- bcrypt.js for password hashing

- Express Middleware for Authentication & Authorization

**Installation & Setup**

**Prerequisites**

Ensure you have the following installed:

- [Node.js](https://nodejs.org/)

- [MongoDB](<https://www.mongodb.com/try/download/community>)

**Steps to Setup**

1. Clone the repository:

   git clone https://github.com/your-repo/user-management-system.git

   cd user-management-system

2. Install dependencies:

   npm install

3. Configure environment variables:

   Create a `.env` file in the root directory and add the following:

   DATABASE\_URL=mongodb://127.0.0.1:27017/user-management

   JWT\_SECRET=supersecretkey

   PORT=5000

4. Start the server:

   npm start

5. Server will run on `http://localhost:5000`

**API Endpoints**

**Authentication**

| Method | Endpoint          | Description         | Protected |

|--------|------------------|---------------------|-----------|

| POST   | `/api/auth/signup` | Register a new user | No        |

| POST   | `/api/auth/login`  | Authenticate user   | No        |

**Request Body (Signup)**

Json data

{

  "name": "John Doe",

  "email": "johndoe@example.com",

  "password": "securepassword",

  "role": "FarmAdmin"

}

**Request Body (Login)**

Json data

{

  "email": "johndoe@example.com",

  "password": "securepassword"

}

**Farm Management**

| Method | Endpoint     | Description              | Protected |

|--------|-------------|--------------------------|-----------|

| POST   | `/api/farms` | Create a new farm        | Yes (Admin) |

| GET    | `/api/farms` | View all farms          | Yes (Admin, Manager) |

**Request Body (Create Farm)**

Json data

{

  "name": "Green Valley Farm",

  "location": "California, USA"

}

**Task Management**

| Method | Endpoint          | Description         | Protected |

|--------|------------------|---------------------|-----------|

| POST   | `/api/tasks` | Create a new task  | Yes (Manager) |

| PUT    | `/api/tasks/:id` | Update task status | Yes (Technician) |

| GET    | `/api/tasks` | View all tasks | Yes (Admin, Manager) |

**Request Body (Create Task)**

Json data

{

  "title": "Irrigation Check",

  "description": "Ensure all irrigation systems are functioning",

  "assignedTo": "user\_id",

  "deadline": "2024-04-10"

}

**Request Body (Update Task Status)**

json

{

  "status": "Completed"

}

**Middleware**

| Middleware | Description |

|------------|-------------|

| `authMiddleware` | Verifies JWT token |

| `roleMiddleware(roles)` | Restricts access based on user roles |

**Project Structure**

user-management-system/

│── models/

│   ├ user.js

│   ├ task.js

│   ├ farm.js

│── routes/

│   ├ auth.js

│   ├ users.js

│   ├ task.js

│   ├ farm.js

│── middlewares/

│   ├ auth.js

│   ├ roleAuth.js

│── config/

│   ├database.js

│── app.js

│── server.js

│── .env

│── package.json

**Running Tests**

npm test