CarSet Task Management API

Overview

The CarSet Task Management API is a backend service designed to help the CarSet operations team manage tasks related to luxury car rentals. This API allows users to register, authenticate, and manage tasks efficiently and securely.

Features

- User registration and authentication with JWT
- Secure password storage using bcrypt hashing
- Task management: create, retrieve, update, complete, and delete tasks
- Public and authenticated task creation endpoints

Tech Stack

- Node.js
- Express.js
- MongoDB (with Mongoose)
- JSON Web Tokens (JWT)
- Bcrypt for password hashing

Getting Started

Prerequisites

- Node.js (version 14 or higher)
- MongoDB (local or cloud instance)
- npm (Node package manager)

Installation

1. **Clone the repository:**

   bash

```
git clone https://github.com/Kentswindells45/carset-task-api.git
cd carset-task-api
```

2. Install dependencies:

   bash

```
npm install
```

3. Configure environment variables:

   Create a `.env` file in the root directory and add your MongoDB connection string and JWT secret:

```env
MONGODB_URI=your_mongodb_connection_string
JWT_SECRET=your_jwt_secret
PORT=5000
```

Running the Application

1. **Start the server:**

```bash
npm start
```

2. The server will run on `http://localhost:5000` (or the port specified in your `.env` file).

API Endpoints

Authentication

- POST `/api/register`**: Register a new user

  Request body: `{ "name": "User", "email": "user@example.com", "password": "password123" }`

- POST `/api/login`**: Login and receive a JWT

  Request body: `{ "email": "user@example.com", "password": "password123" }`

Task Management

- POST `/api/tasks`**: Create a new task (requires authentication)

- GET `/api/tasks`**: Retrieve all tasks for the authenticated user (requires authentication)

- PUT `/api/tasks/:id`**: Edit an existing task (requires authentication)

- PATCH `/api/tasks/:id/complete`**: Mark a task as completed (requires authentication)

- DELETE `/api/tasks/:id`**: Delete a task (requires authentication)

- POST `/api/add`**: Add a new task without authentication (for testing/demo)

Usage Examples

Register a User

```bash
curl -X POST http://localhost:5000/api/register \
-d '{"name": "John Doe", "email": "john@example.com", "password": "password123"}' \
-H "Content-Type: application/json"
```

Login

```bash
```

```
curl -X POST http://localhost:5000/api/login \
-d '{"email": "john@example.com", "password": "password123"}' \
-H "Content-Type: application/json"
```

Create a Task (Authenticated)

```bash
curl -X POST http://localhost:5000/api/tasks \
-d '{"title": "Verify vehicle documents", "description": "Check all necessary documents for the vehicle."}' \
-H "Authorization: Bearer your_jwt_token" \
-H "Content-Type: application/json"
```

Add a Task Without Authentication

```bash
curl -X POST http://localhost:5000/api/add \
-d '{"title": "Public Task", "description": "No auth required"}' \
-H "Content-Type: application/json"
```

Testing with Postman

1. **Register** a user using the `/api/register` endpoint.
2. **Login** to receive a JWT token.
3. For all authenticated endpoints, add this header in Postman:
   `Authorization: Bearer <your_jwt_token>`
4. Test task creation, retrieval, update, completion, and deletion as described above.

License

Acknowledgments

- Inspired by the need for efficient task management in the luxury car rental industry.
- Developed by Oppong Kelvin.