

Documentation for Contacts Management System

Overview:

This Contacts Management System is a web application for handling user contacts and messages with OTP functionality using Twilio. It allows users to register, login, manage their contacts, and send OTP messages via SMS. And implemented proper error handling for various situations.

Live URL:

- [Contacts Management System](#)
- (or)
- <https://contacts-management-web.vercel.app/>

And github repo for this project : [contacts-otp-repo](#)

Key Features:

1. **User Authentication:** Users can register, login, and access the platform securely using JWT tokens.
2. **Contact Management:** Users can add, update, and view contacts.
3. **Message Sending:** Send OTP messages to contacts using Twilio.
4. **Pagination & Search:** Contacts and messages are paginated and searchable.
5. **Security:** Passwords are hashed using `bcryptjs` and all routes are protected with JWT tokens and middleware.

Technology Stack:

- **Backend:** Node.js with Express.js
- **Database:** MongoDB with Mongoose
- **Frontend:** React.js with React Router
- **Styling:** Tailwind CSS and MUI for styling the UI
- **Authentication:** JWT for secure user sessions
- **Message Service:** Twilio API for sending OTP messages
- **Deployment:** Vercel

Key Libraries Used:

- **Express:** For handling server-side routes and API creation.
- **Mongoose:** For object modeling and database operations.
- **JWT (jsonwebtoken):** To handle secure user authentication and token generation.
- **bcryptjs:** For hashing passwords before storing them in the database.
- **Twilio:** For sending SMS messages (OTP) to contacts.
- **React Router:** For frontend route handling.
- **mongoose-paginate-v2:** To implement pagination for contacts and messages.
- **dotenv:** For managing environment variables securely.

Folder Structure:

Backend:

- **config/db.js:** Handles the connection to MongoDB.
- **routes/authRoutes.js:** Contains routes for user registration and login.
- **routes/contactRoutes.js:** Contains routes for managing contacts (CRUD operations).
- **routes/messageRoutes.js:** Handles OTP message sending and retrieval of sent messages.
- **models:** Defines schemas for **User**, **Contact**, and **Message**.
- **controllers:** Business logic for user registration, contact management, and OTP messaging.

Frontend:

- **components:** Reusable components like **Navbar** and **Sidebar**.
- **pages:** Specific views like **Login**, **Register**, **Contacts**, **SentMessages**, etc.
- **utils:** Contains utility functions for token management and authentication handling.

Good Practices and Decisions:

1. **Separation of Concerns:** The project follows the MVC (Model-View-Controller) pattern to separate data, business logic, and UI.
2. **Environment Variables:** Sensitive credentials such as JWT secrets and Twilio credentials are stored securely in **.env** files.
3. **Error Handling:** Clear error handling at both the API and frontend levels for better debugging.

4. **Security Measures:**
 - **JWT Tokens:** Used for authentication and secure session handling.
 - **bcryptjs:** Passwords are encrypted before storage, ensuring user data protection.
5. **Pagination:** Contacts and messages are paginated to ensure smooth performance, especially with large datasets.
6. **Twilio API Integration:** Robust error handling for cases such as unverified numbers or failed OTP delivery.

Setup & Installation:

1. Clone the repository and navigate into the project directory.

Create a `.env` file with the following variables:

bash

Copy code

```
MONGODB_URI=your_mongodb_connection_string
JWT_SECRET=your_jwt_secret
TWILIO_ACCOUNT_SID=your_twilio_account_sid
TWILIO_AUTH_TOKEN=your_twilio_auth_token
TWILIO_PHONE_NUMBER=your_twilio_phone_number
```

- 2.

Install backend dependencies:

```
npm install
```

- 3.

Run the backend:

```
npm start
```

- 4.

Navigate to the frontend directory and install dependencies:

```
npm install
```

- 5.

Start the frontend:

```
npm run dev
```

6.

Deployment:

- The server and client deployed on **Vercel** for rapid deployment and testing.

Contact Information:

If you encounter any issues or need further clarification, feel free to reach out at:

- **Email:** tejassriavinasha@gmail.com
- **Mobile :** 7386223111