

Contact Form Application

Create a **Contact Form Application** that allows users to submit their contact details through an HTML form. The backend should be developed using **Express**, **Mongoose**, and **MongoDB**, with functionality to send an email notification to a predefined admin email address upon form submission. Admins can retrieve all contact messages using **Postman**, secured by a **secret key** sent through the request headers.

API Requirements

1. POST `/api/contact`

Purpose:

This endpoint allows users to submit their contact details (name, email, mobile number, and message) via a contact form. The data is validated and stored in the MongoDB database. After successful form submission, an email is sent to a pre-configured admin email address with the details of the submitted message.

Request Format:

- **Method:** `POST`
- **Endpoint:** `/api/contact`
- **Content-Type:** `application/json`

Request Body (example):

```
{
  "name": "John Doe",
  "email": "johndoe@example.com",
  "mobile": "1234567890",
  "message": "I'm interested in your services."
}
```

Response:

- **201 Created:** If the submission is successful, the contact form is saved in the database, and an email is sent to the admin.

```
{
  "message": "Contact form submitted successfully."
}
```

- **400 Bad Request:** If there is a validation error (e.g., missing fields or invalid format).

```
{
  "error": "Invalid form data."
}
```

- **500 Internal Server Error:** If the server encounters an issue while saving the data or sending the email.

```
{
  "error": "Something went wrong. Please try again."
}
```

Key Features:

- **Data Validation:** Validates name, email format, mobile number, and message.
 - **Email Notification:** Sends a notification to the admin email (pre-configured in environment variables).
 - **MongoDB Storage:** Saves the submitted form data in the MongoDB database.
-

2. GET `/api/contacts`

Purpose:

This endpoint allows the admin to retrieve all the submitted contact form messages. The request must include a **secret key** in the **Authorization** header for authentication. The secret key is stored securely in the backend environment variables (`process.env.SECRET_KEY`). Only valid requests with the correct secret key will have access to the data.

Request Format:

- **Method:** GET
- **Endpoint:** `/api/contacts`
- **Headers:**
 - **Authorization:** `secret-key` (where `secret-key` is the value stored in `process.env.SECRET_KEY`)

Response:

- **200 OK:** If the secret key is valid and the request is successful, the admin will receive a list of all the contact form submissions.

```
[
  {
    "name": "John Doe",
    "email": "johndoe@example.com",
    "mobile": "1234567890",
    "message": "I'm interested in your services.",
    "submittedAt": "2024-09-27T10:00:00Z"
  },
  {
    "name": "Jane Smith",
    "email": "janesmith@example.com",
```

```
"mobile": "0987654321",
"message": "Looking for more information on your offerings.",
"submittedAt": "2024-09-27T11:00:00Z"
}
]
```

- **403 Forbidden:** If the secret key is missing or incorrect.

```
{
  "error": "Access denied. Invalid secret key."
}
```

- **500 Internal Server Error:** If something goes wrong while fetching the data from the database.

```
{
  "error": "Unable to retrieve contact messages."
}
```

Key Features:

- **Admin Authentication:** Requires a valid secret key in the **Authorization** header.
- **Database Access:** Fetches all contact form submissions from MongoDB.
- **Security:** Access to this route is restricted to authorized admin users by verifying the secret key stored in environment variables.

Email Notification Template

When a user successfully submits a contact form, an email notification is sent to the admin. This email contains the details of the user's submission.

Email Template:

- **To:** Admin's Email Address (configured in **process.env.ADMIN_EMAIL**)
- **Subject:** New Contact Form Submission
- **Body:**

```
Hello Admin,
```

```
A new contact form has been submitted with the following details:
```

```
- **Name**: [User's Name]
- **Email**: [User's Email]
- **Mobile**: [User's Mobile Number]
- **Message**: [User's Message]
```

Submitted At: [Date and Time of Submission]

Best Regards,
Contact Form Application

Example:

- **To:** `admin@example.com`
- **Subject:** New Contact Form Submission
- **Body:**

Hello Admin,

A new contact form has been submitted with the following details:

- ****Name**:** John Doe
- ****Email**:** johndoe@example.com
- ****Mobile**:** 1234567890
- ****Message**:** I'm interested in your services.

Submitted At: 2024-09-27 10:00:00 UTC

Best Regards,
Contact Form Application

Email Sending Process:

1. Once the user submits the contact form via the `/api/contact` POST request, the backend processes the data.
2. After the data is successfully saved in the database, an email is sent to the admin's pre-configured email (`process.env.ADMIN_EMAIL`) using **Nodemailer** (or a similar email service).
3. The email contains all the details of the submitted form (name, email, mobile number, and message), along with a timestamp.