

COLLEGE CODE : 8203

COLLEGE NAME :AVC COLLEGE OF ENGINEERING

DEPARTMENT :CSE

STUDENT NM ID :4CF0B691E7230C98668406C5A3493455

ROLL NO : 820323104065

DATE : 22-10-25(COMPLETED THE PROJECT)

NAMED AS :PHASE 5 TECHNOLOGY

PROJECT NAME : RESTFUL CONTACT API MANAGEMENT

SUBMITTED BY,

MULLAI. P

MOBILE NO : 7548875472

IBM-NJ-RESTFUL Contact Management API

Phase 5 - Project Demonstration & Documentation

i) Final Demo Walkthrough :

1. Overview :

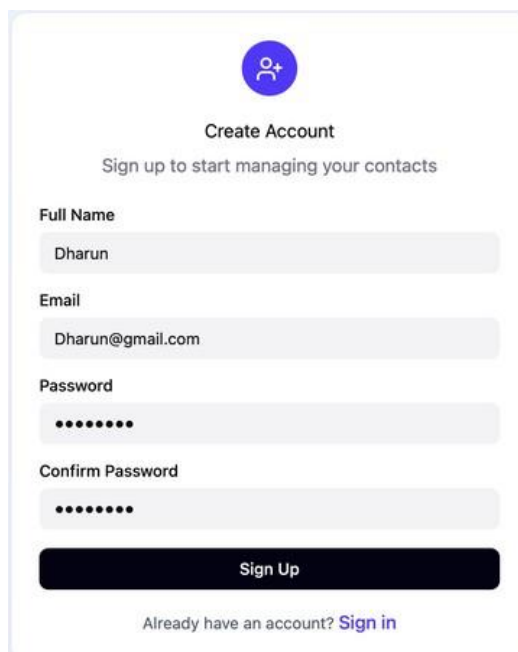
The RESTful Contact Management System is a full-stack web application that allows users to manage their personal and professional contacts efficiently. It demonstrates CRUD (Create, Read, Update, Delete) operations using RESTful API principles.

2. Features Demonstrated :

- ✓ User Registration and Login (JWT Authentication)
- ✓ Add new contact (POST API)
- ✓ Retrieve all contacts (GET API)
- ✓ Update contact details (PUT API)
- ✓ Delete contact (DELETE API)
- ✓ Responsive and user-friendly UI
- ✓ Error handling and validation

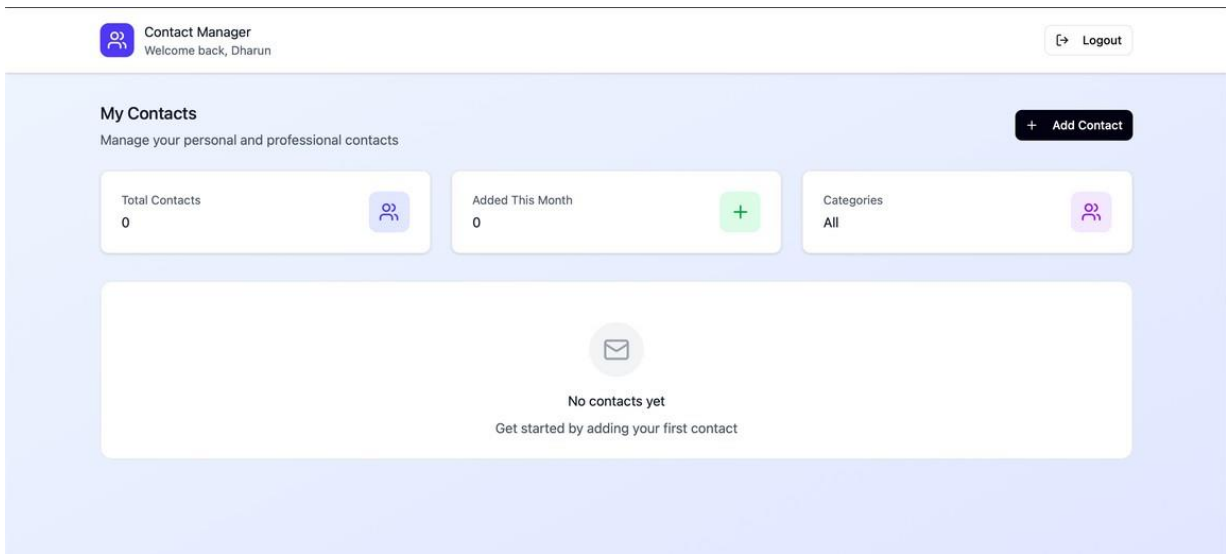
3. Walkthrough Flow :

- User Registration/Login: User signs up or logs in securely.
- Dashboard: Displays all existing contacts from the database.
- Add Contact: Form to add new contact details (name, phone, email, address).
- Edit/Delete Contact: Modify or remove any saved contact.
- Logout: Ends session securely using token invalidation.



The screenshot shows a 'Create Account' form with the following elements:

- A purple circular icon with a white person silhouette and a plus sign.
- The text 'Create Account' in bold.
- The text 'Sign up to start managing your contacts' in a smaller font.
- Four input fields: 'Full Name' (containing 'Dharun'), 'Email' (containing 'Dharun@gmail.com'), 'Password' (masked with dots), and 'Confirm Password' (masked with dots).
- A dark blue 'Sign Up' button.
- A link at the bottom: 'Already have an account? [Sign in](#)'.



Add New Contact

Full Name *

Lisa

Phone Number *

5342672965

Email Address *

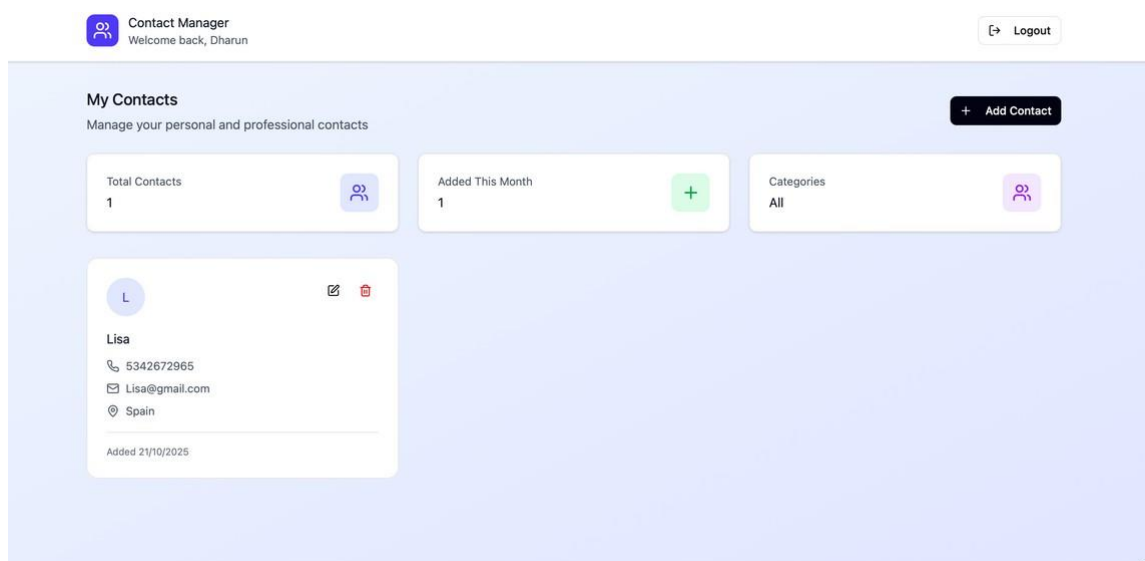
Lisa@gmail.com

Address *

Spain

Cancel

Add Contact



ii) Project Report :

1. Abstract :

The Contact Management System simplifies the process of managing and storing contact details. Built with Node.js and Express.js as backend and MongoDB for storage, it provides RESTful APIs for seamless integration. This project demonstrates REST principles like statelessness, layered architecture, and CRUD operations.

2. Objectives :

- Implement a RESTful API using Node.js and Express.js.
- Perform CRUD operations efficiently.
- Maintain data persistence using MongoDB.
- Ensure secure access using JWT Authentication.
- Develop a user-friendly frontend interface.

3. System Architecture :

User → Frontend (React/HTML) → REST API (Express.js) → Database (MongoDB)

- Each request follows REST conventions using HTTP methods (GET, POST, PUT, DELETE).
- Middleware handles authentication and validation.
- Responses are JSON-based for easy integration.

4. Modules :

1. **Authentication Module** – Handles user login/signup with JWT.
2. **Contact Module** – Manages CRUD operations on contacts.
3. **Validation Module** – Ensures correct input formats.
4. **Database Module** – Connects and queries MongoDB.

5. Tools Used :

Node.js is used as the backend runtime environment, while Express.js serves as the API framework. MongoDB acts as the database with Mongoose for schema management. JWT ensures authentication security. Postman is used for testing APIs, GitHub for version control, and Render or Netlify for deployment.

iii) Screenshots / API Documentation :

1. API Endpoints :

The backend supports several API routes:

- **POST /api/users/register** – registers a new user
- **POST /api/users/login** – logs in the user and returns a JWT token
- **GET /api/contacts** – retrieves all stored contacts
- **POST /api/contacts** – adds a new contact
- **PUT /api/contacts/:id** – updates an existing contact
- **DELETE /api/contacts/:id** – deletes a contact

2. Example Request & Response :

POST /api/contacts

Request Body:

```
{  
  "name": "John Doe",  
  "email": "john@example.com",  
  "phone": "9876543210"  
}
```

Response:

```
{  
  "message": "Contact added successfully",  
  "data": {  
    "_id": "6715c3f4b23a",  
    "name": "John Doe",  
    "email": "john@example.com",  
    "phone": "9876543210"  
  }  
}
```

iv) Challenges and Solutions :

1. User Authentication and Security: Implementing secure login and registration was challenging.

Solution: Used JWT authentication to verify users and protect all private API routes

2. Duplicate Contact Entries: Multiple users were able to add the same contact information.

Solution: Added unique field validation in the Mongoose schema to prevent duplicate records.

3. Database Connection Issues: Connection failures occurred during deployment.

Solution: Configured MongoDB URI through environment variables and added retry logic for stable connections.

4. CORS and API Integration Errors: The frontend was initially blocked from accessing the backend.

Solution: Enabled and configured CORS middleware properly in the Express server to allow cross-origin requests.

5. Error Handling and Validation: Inconsistent error messages and invalid data caused confusion.

Solution: Created centralized error-handling middleware and used validation libraries to ensure clean, accurate input data.

v) GitHub README & Setup Guide :

1. GitHub README :

README : RESTful Contact Management System

A full-stack web application that allows users to manage their personal and professional contacts securely using a RESTful API. It supports user authentication, and CRUD (Create, Read, Update, Delete) operations, and demonstrates clean backend architecture using Node.js, Express.js, and MongoDB.

2. Setup Instructions :

```
# Clone the repository
git clone https://github.com/username/contact-management.git

# Navigate to project directory
cd contact-management

# Install dependencies
npm install

# Setup environment variables
create a .env file and add:
PORT=5000
MONGO_URI=your_mongodb_connection_string
JWT_SECRET=your_jwt_secret

# Start the backend server
npm start

# If frontend is included:
cd client
npm start
```

3. Usage :

- Open <http://localhost:5000> (or deployed URL).
- Register and log in.
- Add, edit, and delete contacts easily.

vi) Final Submission :

- **GitHub Repository:** https://github.com/mullaipari/NM_phase5.git
- **Deployed Link (Render/Netlify):** https://mullaipari.github.io/NM_phase5/