# Contact Management App (MyContacts)

## Project Introduction

</talentlabs>

# Contact Management App (Backend APIs)

## Introduction

Contact management App is the set of backend APIs which allows the user to manage their contacts. As this is a backend express project so to test the APIs we use HTTP client (Thunder client) and do the CRUD operations on the Contacts. The App will also contain the Registration of the Users along with login and current logged in user info apis. Only logged in users can manage the contacts of their own.

## Goals

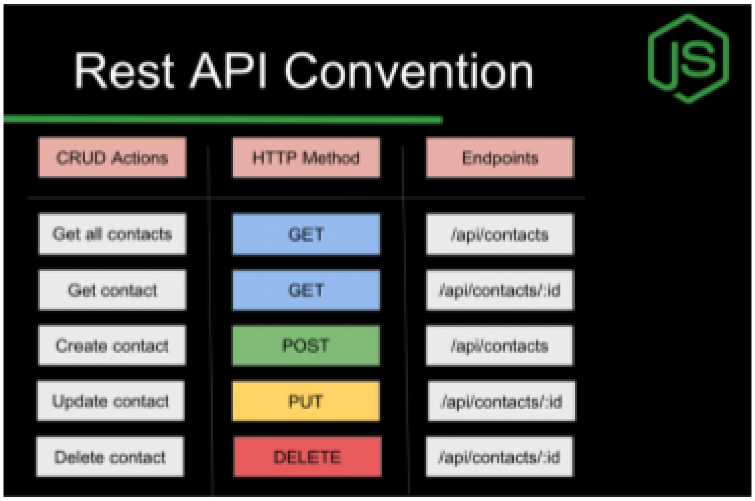
- Users can register for their account.
- Once registration is successful, users can login to their account and manage their contacts.
- Users can create, update, delete, get all contacts or get a single contact which they have created.
- Prevent users from managing contacts of different users.
- Users should have a valid access token to use the private endpoints.
- Proper error messages with status code should be provided to the users in case of any error.
- Users password should not be stored as raw password in the Database.
- Provide users with a signed JWT access token on successful login to access private endpoints.

## Tools & Libraries

- **Node.js :** ( module: http, path, fs )
- **Express :** Use to get robust features to build web application using node.js (router, body parser, request params)
- **Bcrypt :** Used for hashing the user password and compare the hash password during login
- **Dotenv :** Used to access the environment variables via node.js process core module into our files
- **Express-async-handler :** Middleware used to handle exceptions inside of async express routes and passing them to your express error handlers
- **Jsonwebtoken :** Used to add authentication to the apis by signing and verifying the access token.
- **MongoDB :** Used as a NO-SQL database to store our data in document/ collections in the form of JSON objects.
- **Mongoose :** Used to create object model design schema for our entities like (Contacts, Users) and provide access to methods on the entities to communicate with MongoDB.
- **Thunder Client** : Used to test our API from VScode. It is a lightweight Rest API client.
- **Nodemon** : Used to detect the changes in file and restart the server automatically, especially used for development purposes.

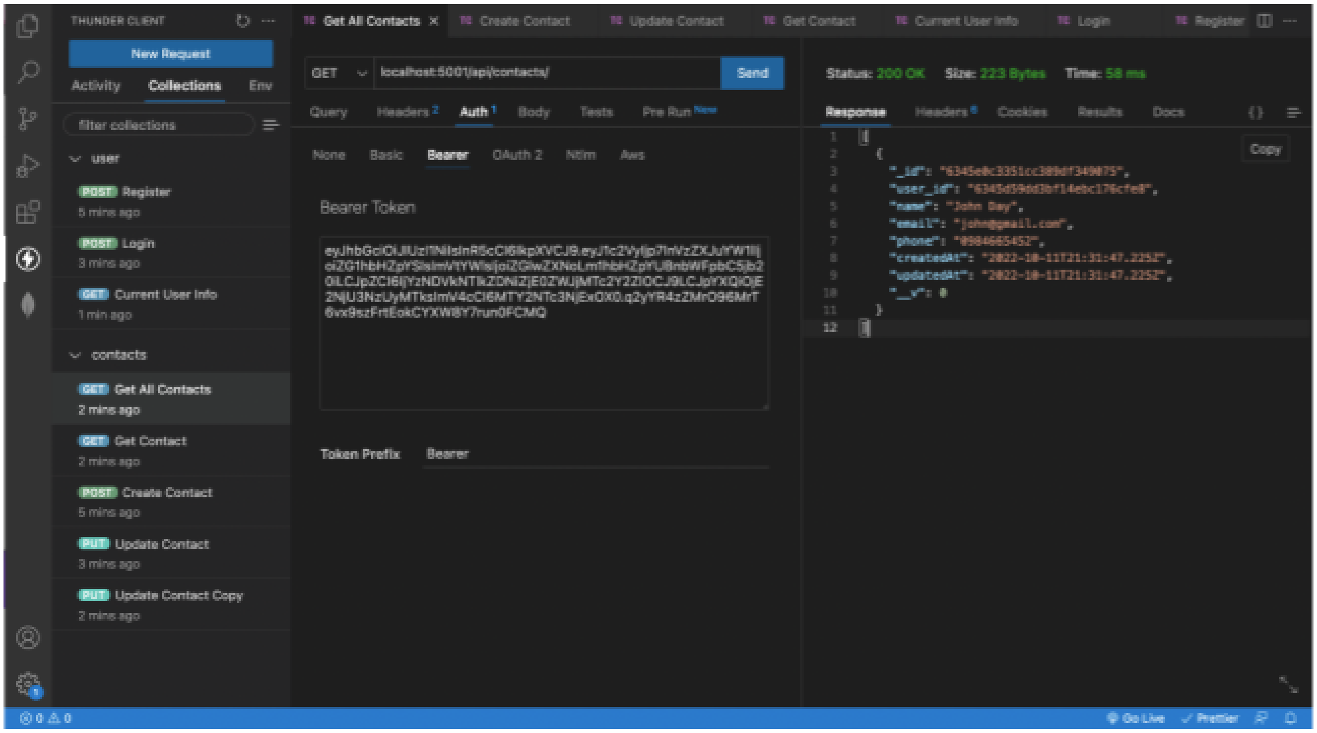
## Resources & Collections

The app will have 2 resources one will be dealing with Contacts and other will be Users. For the resources we will follow RESTful API naming convention



Rest API Convention

| CRUD Actions | HTTP Method | Endpoints |
|---|---|---|
| Get all contacts | GET | /api/contacts |
| Get contact | GET | /api/contacts/:id |
| Create contact | POST | /api/contacts |
| Update contact | PUT | /api/contacts/:id |
| Delete contact | DELETE | /api/contacts/:id |



user
- POST Register  
  5 mins ago
- POST Login  
  3 mins ago
- GET Current User Info  
  1 min ago

contacts
- GET Get All Contacts  
  2 mins ago
- GET Get Contact  
  2 mins ago
- POST Create Contact  
  5 mins ago
- PUT Update Contact  
  3 mins ago
- PUT Update Contact Copy  
  2 mins ago

GET localhost:5001/api/contacts/

Bearer Token

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyIjp7InVzZXJuYW1lIj
oiZG1hbHZpYSIsImVtYWlsIjoiZGIwZXNoLmthbHZpYUBnbWFpbC5jb2
0iLCJpZCI6IjYzNDVkNTkZDNiZjE0ZWJjMTc2Y2ZlOCJ9LCJpYXQiOjE
2NjU3NzUyMTksImV4cCI6MTY2NTc3NjExOX0.q2yYR4zZMrO96MrT
6vx9szFrtEokCYXW8Y7run0FCMQ
```

Token Prefix: Bearer

Status: 200 OK  Size: 223 Bytes  Time: 58 ms

```
[
  {
    "_id": "6345e0c3351cc389df349075",
    "user_id": "6345d59dd3bf14ebc176cfe8",
    "name": "John Dey",
    "email": "john@gmail.com",
    "phone": "0984665452",
    "createdAt": "2022-10-11T21:31:47.225Z",
    "updatedAt": "2022-10-11T21:31:47.225Z",
    "__v": 0
  }
]
```

Please refer to the video for more details.

# Thank you.

# All The Best :)