**COMP229 – Web Application Development**

Assignment 2

# Portfolio Application – Node.js, Express REST APIs & MongoDB

Student: Wenping Wang

Student number: 301250155

Instructions :

The Portfolio Application:

1. Using MongoDB database, create:**(25 Marks):**
   1. A database by name Skeleton**.**

SKELETON Database created:

A screenshot of a computer

Description automatically generated

* 1. Create the following collections with their respective property. (5 Marks: Functionality).

1. **contacts**

firstname: string

lastname: string

email: string



1. **users**

name: string

email: string

password: string

created: Date

updated: Date

A screenshot of a computer

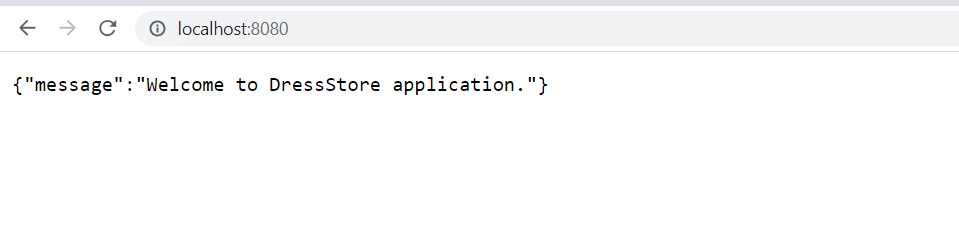
Description automatically generated

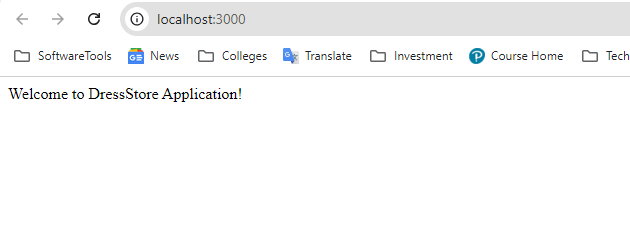
* 1. Obtain your connection string ( url or uri)

mongodb+srv://benjaminwang2088:<db\_password>@cluster0.ulgqy.mongodb.net/?retryWrites=true&w=majority&appName=Cluster0

* 1. Provide the screen snapshot of your MongoDB database showing the above steps from 1a – c.

1. Configure the Backend of your portfolio Application by creating the server.js file in the root folder for your server and a server folder or backend folder in the root folder for all your backend code.
   1. Update the server.js file i.e the server with code to display the message shown in the snapshot below when you run the app and provide a screen snapshot of it running in the browser as follows:





1. Create the web server after creating the Express web server next: **(30 Marks)**
   1. Add the configuration for the MongoDB database.
   2. Create the contact and user model with Mongoose.
   3. Write the controller for the contact and user.
   4. Define the routes for handling all CRUD operations for the contacts and users api.

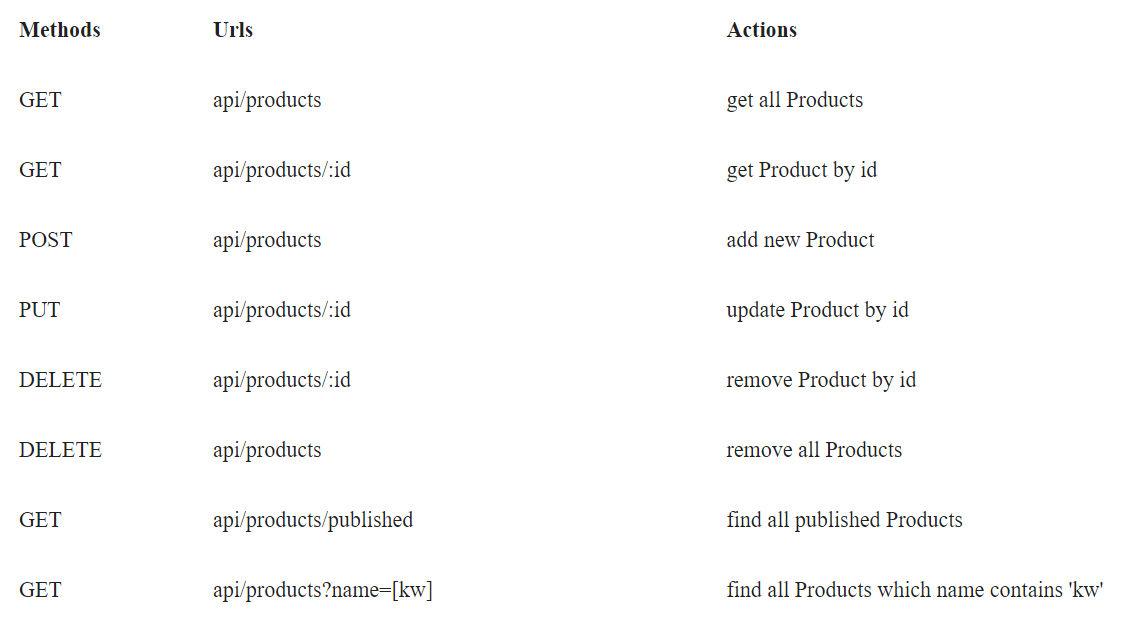
Below is an overview of the REST APIs that will be exported:

**contacts**

|  |  |  |
| --- | --- | --- |
| **Methods** | **Urls** | **Actions** |
| GET | api/contacts | get all contacts |
| GET | api/contacts/:id | get contacts by id |
| POST | api/contacts | add new contact |
| PUT | api/contacts/:id | update contact by id |
| DELETE | api/contacts/:id | remove contact by id |
| DELETE | api/contacts | remove all contacts |

**Users**

|  |  |  |
| --- | --- | --- |
| **Methods** | **Urls** | **Actions** |
| GET | api/users | get all users |
| GET | api/users/:id | get users by id |
| POST | api/users | add new user |
| PUT | api/users/:id | update user by id |
| DELETE | api/users/:id | remove user by id |
| DELETE | api/users | remove all users |

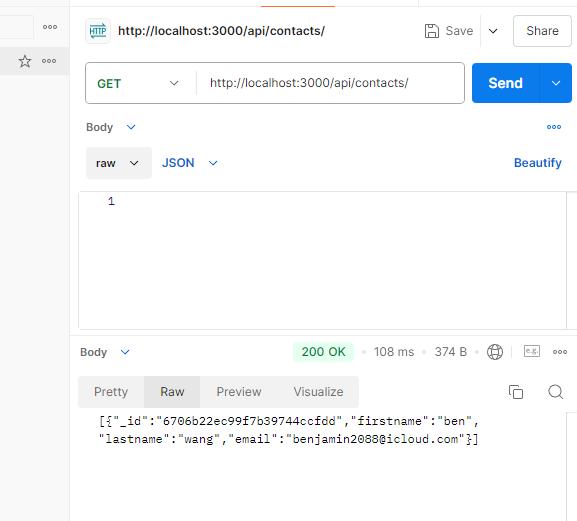


1. a) Test the REST APIs using Postman, Thunder client or any tool you are familiar with. e.t.c.

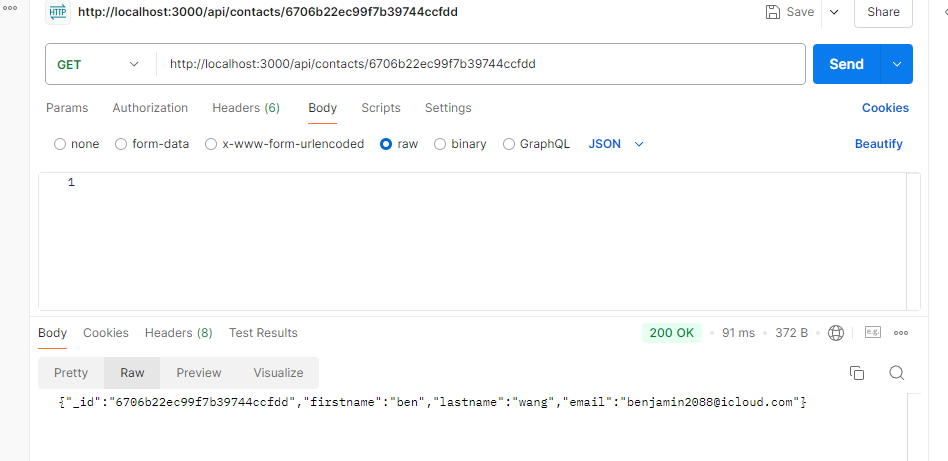
b) Provide the screen snapshot of the test. **(20 Marks)**

Test by postman:

1. <http://localhost:3000/api/contacts/>



1. [http://localhost:3000/api/contacts/:id](http://localhost:3000/api/contacts/:id%20)



1. 

Add one contact to mongo database:

A screenshot of a computer

Description automatically generated

Use get to get the result from mongo

A close up of a text

Description automatically generated

1. Update contact by id:

A screenshot of a computer screen

Description automatically generated

**We can see it updated as follows:**

A screenshot of a computer

Description automatically generated

1. Delete contact by ID:

A screenshot of a computer

Description automatically generated

1. Delete all contacts

A screenshot of a computer

Description automatically generated

**Test user collection**

|  |  |  |
| --- | --- | --- |
| **Methods** | **Urls** | **Actions** |
| GET | api/users | get all users |
| GET | api/users/:id | get users by id |
| POST | api/users | add new user |
| PUT | api/users/:id | update user by id |
| DELETE | api/users/:id | remove user by id |
| DELETE | api/users | remove all users |

1. Get all users

A screenshot of a computer

Description automatically generated

1. Get user by id:

A screenshot of a computer

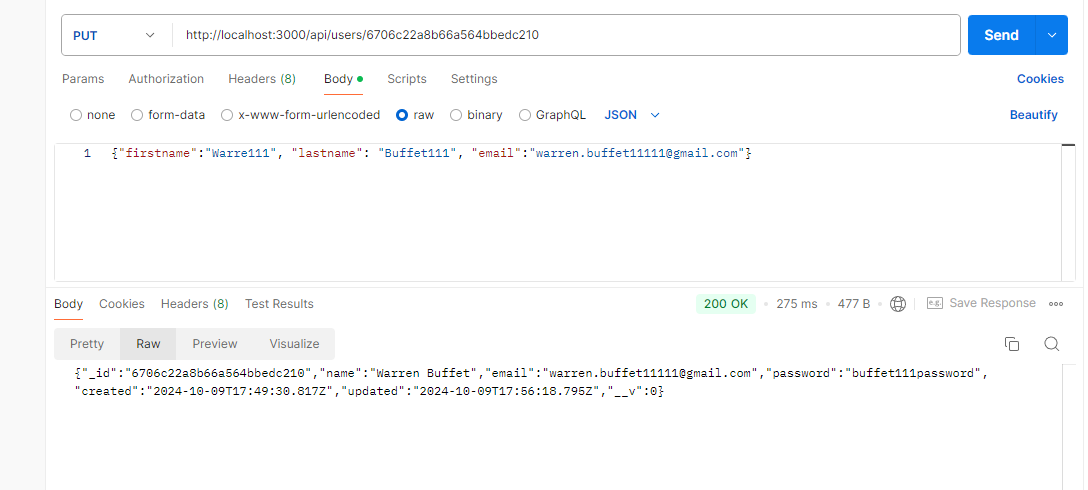
Description automatically generated

1. Add new user

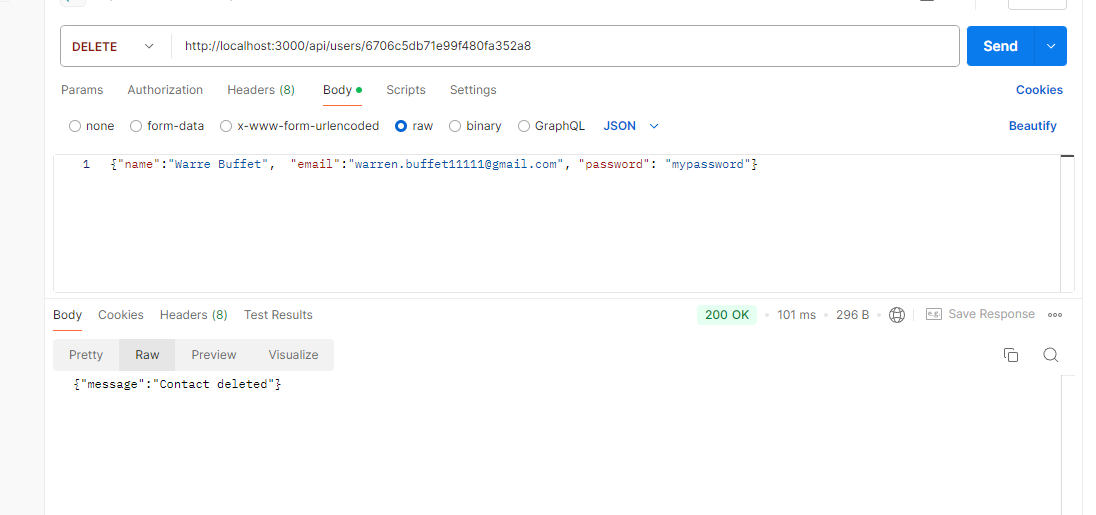
A screenshot of a computer

Description automatically generated

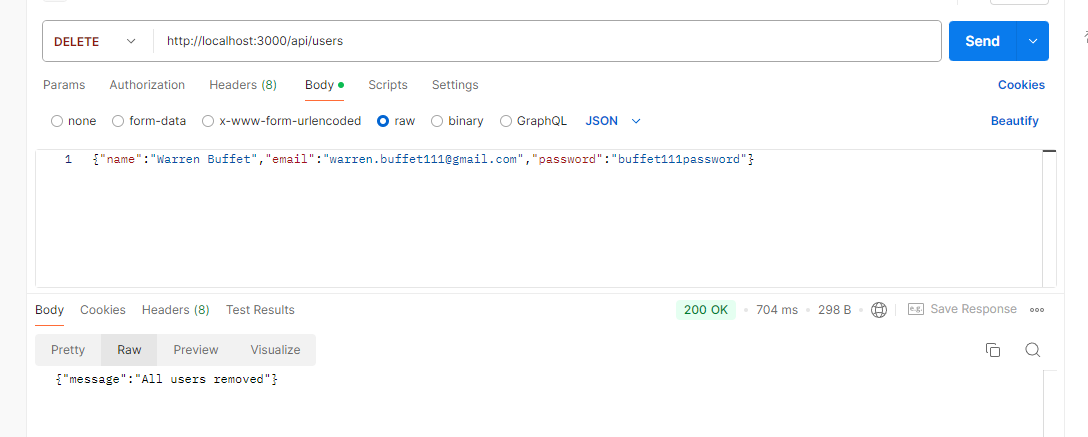
1. Update user by id:



1. Delete user by id:



1. Delete all users



Check the result

**SUBMITTING YOUR WORK**

Your submission should include:

1. A zip archive of your Portfolio Project files
2. A link to GitHub
3. A word doc. Showing the snapshots

This assignment is weighted **10%** of your total mark for this course. Late submissions:20% deducted for each day late.