Name: Chen Hsiao Ting
Matriculation Number: A0222182R
Link to GitHub repository: https://github.com/hsiaotingluv/CS3219-OTOT-TaskB (shared to @ruilingk and @wakululuu)
Submission for Task B1 to B4
Note for instructors: feel free to contact me on telegram @hsiaotingluv to clarify anything!

# Instructions on how to

Task B1: Run the API locally, including Postman calls used to demonstrate a working API. Be sure to include the edge cases tested or proof of error-resiliency as per the marking scheme.
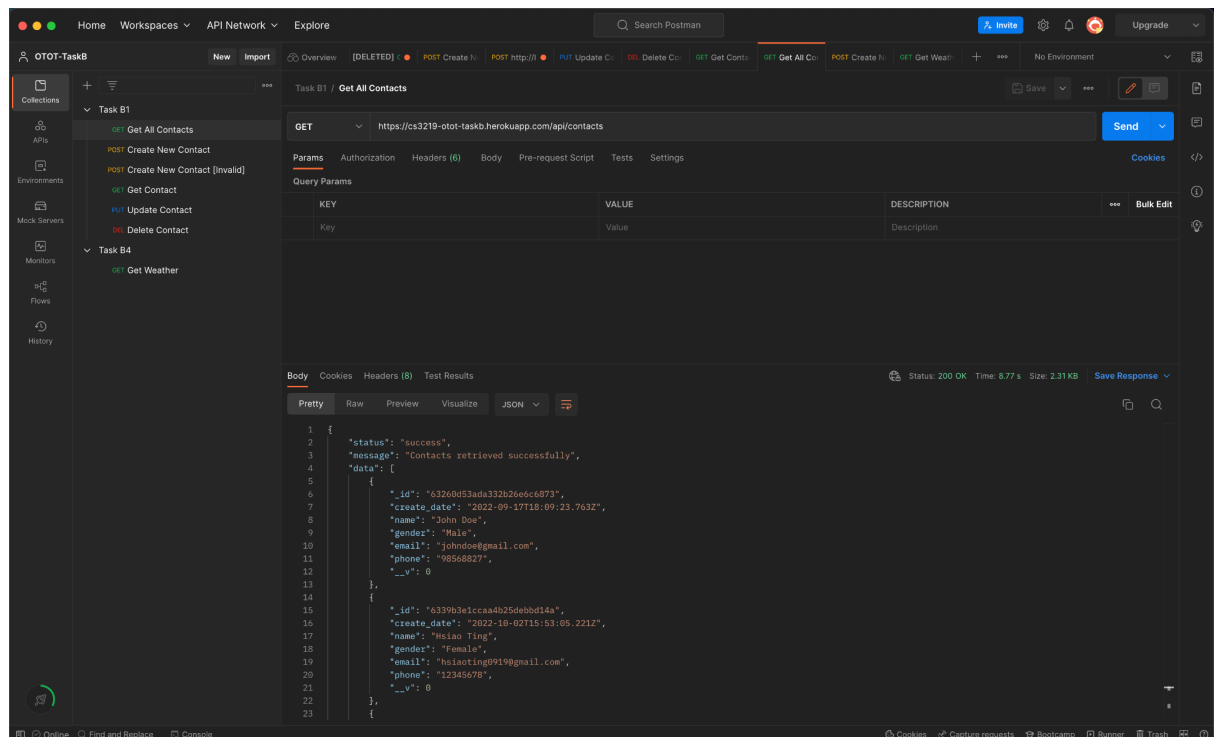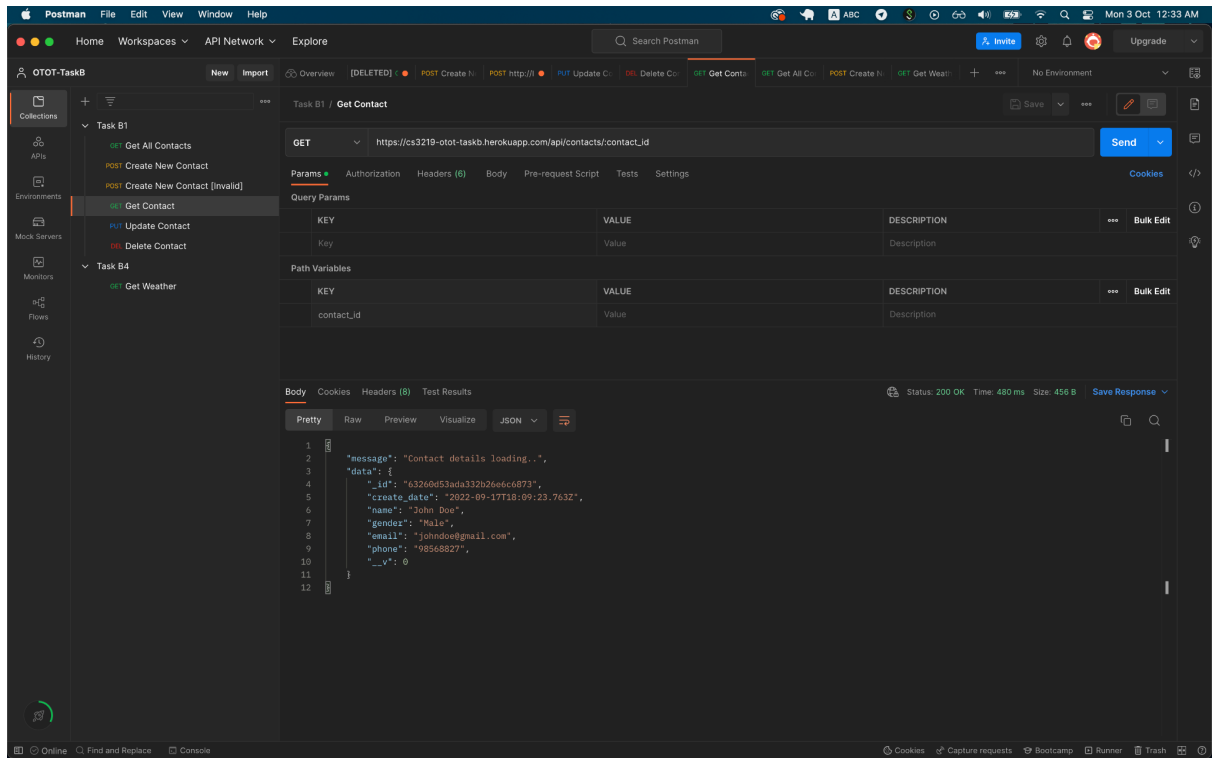
Link to Postman collection

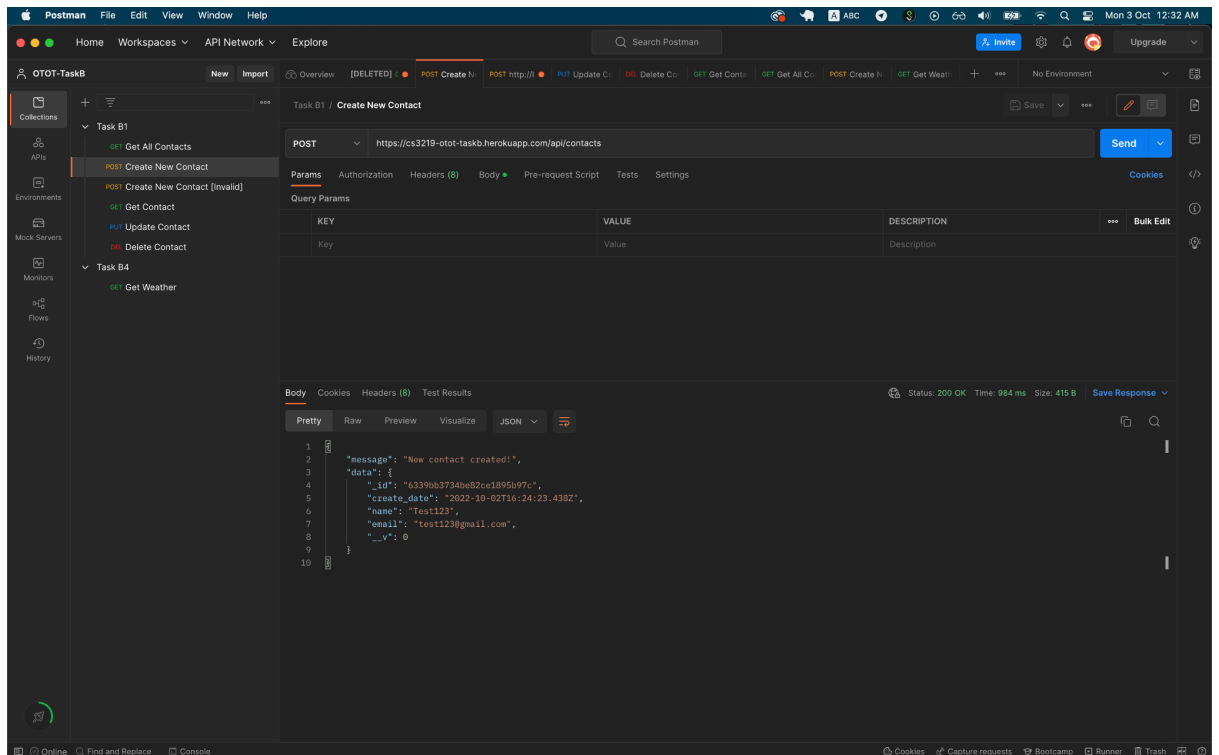https://app.getpostman.com/join-team?invite_code=c56fa438126e7a4915e6b75aa52d172b
or
https://warped-space-936145.postman.co/workspace/c95e921a-8165-43dd-8583-13d5f5e88f1f/collection/12012373-c2dc0a5c-9101-4abf-9961-cbea7d51c10a?action=share&creator=12012373
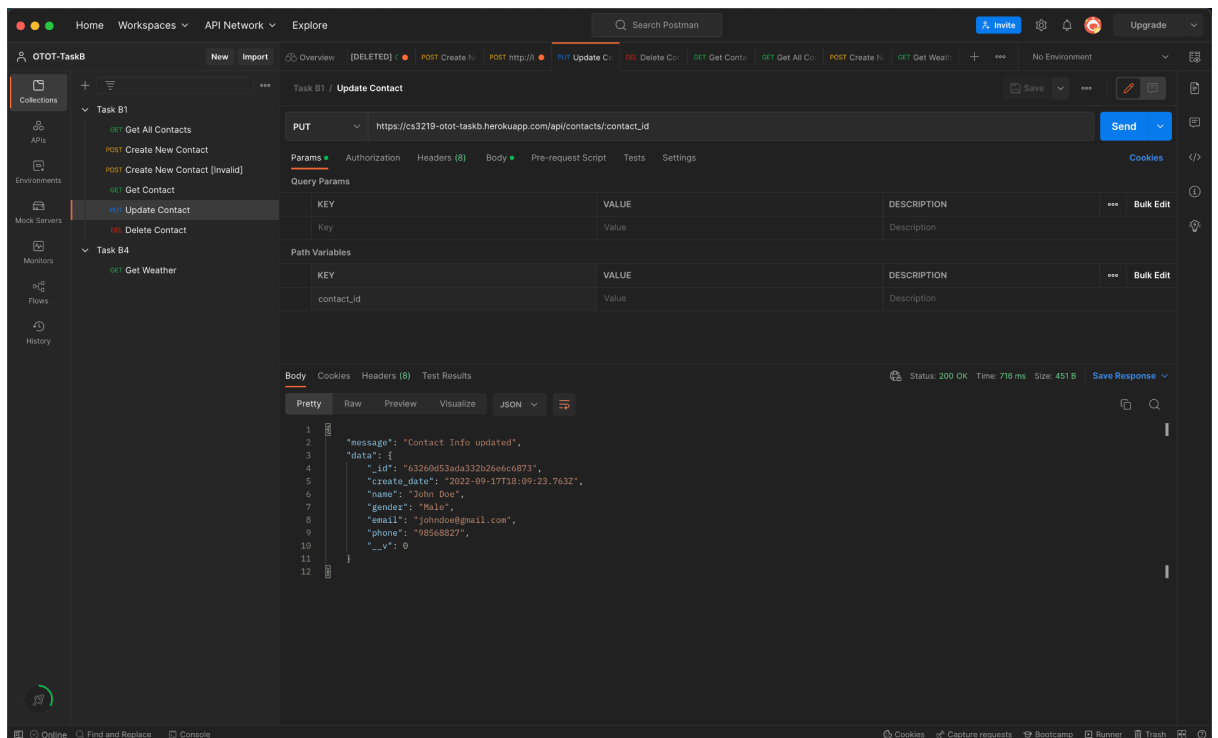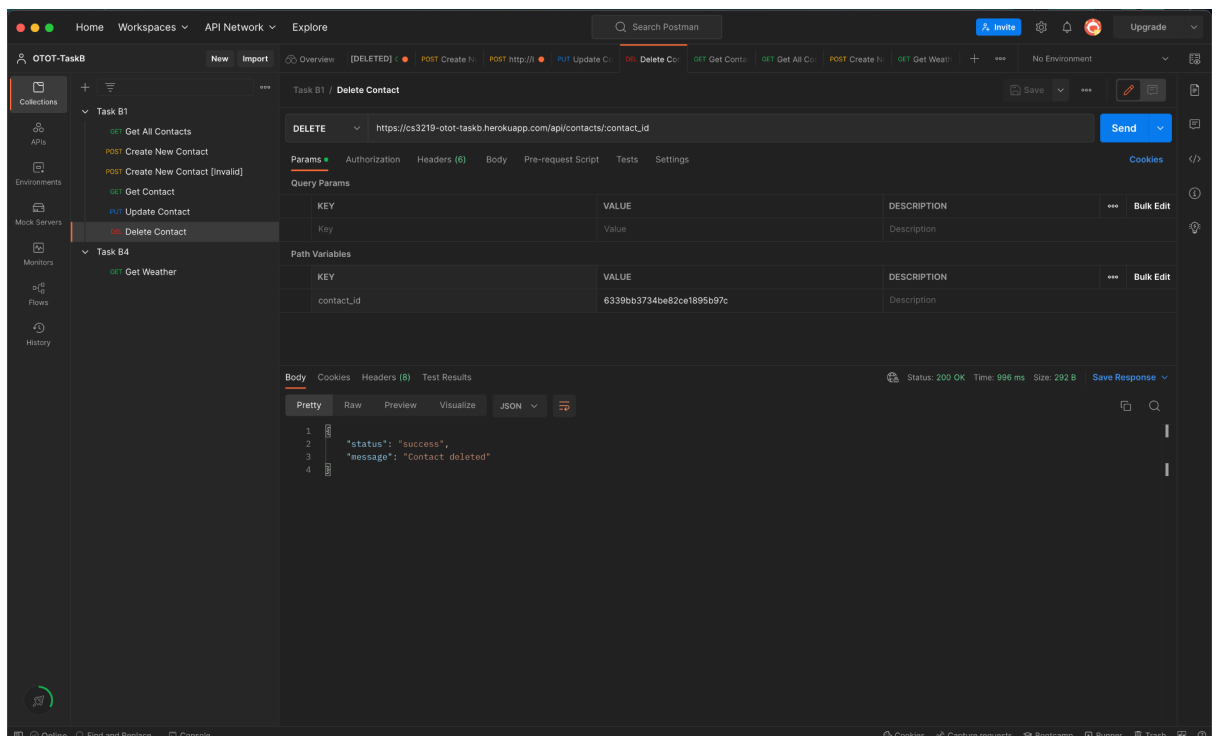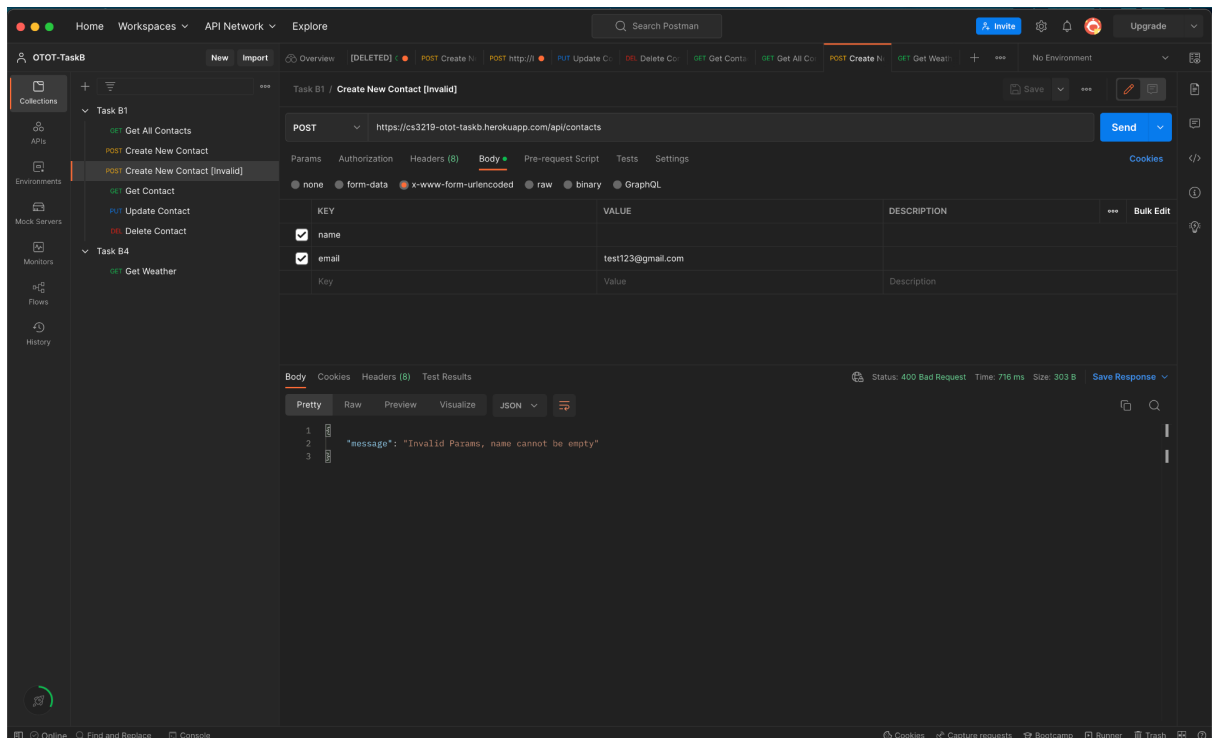
1. GET

2. POST

## 3. PUT



## 4. DELETE



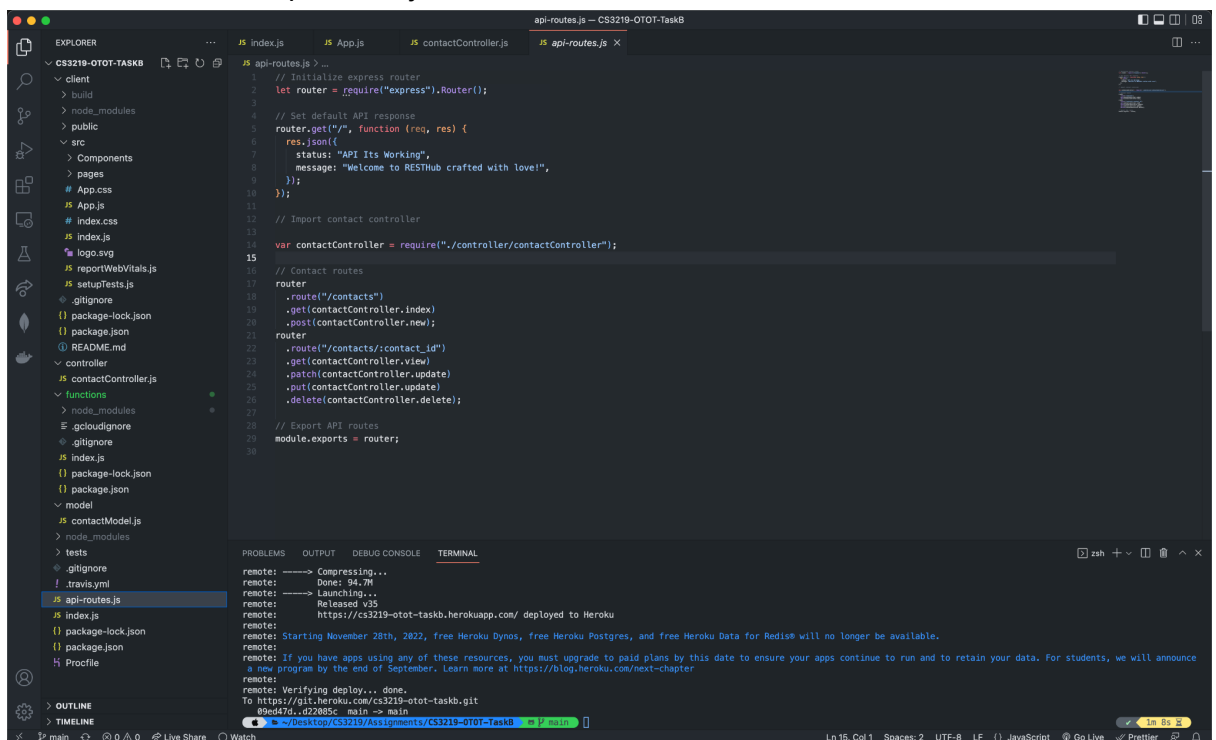5. Invalid data & edge cases tested or proof of error-resiliency

- case 1: [post] contact name cannot be empty, throw an invalid message

- case 2: [put] if the contact name is changed to empty, will update with the previous name instead of throwing an error



# Task B1/B2.2: Access the deployed API

- Define API routes in api-routes.js



- Use API routes in the app in index.js
- To access data from deployed API, use http://{{YOUR ROOT URL}}/api/contacts
  - i.e. https://cs3219-otot-taskb.herokuapp.com/api/contacts

- Continuous Integration whenever commit is pushed to github



hsiaotingluv Remove invalid data for update contact ✓ d22085c 29 minutes ago 🕐 39 commits
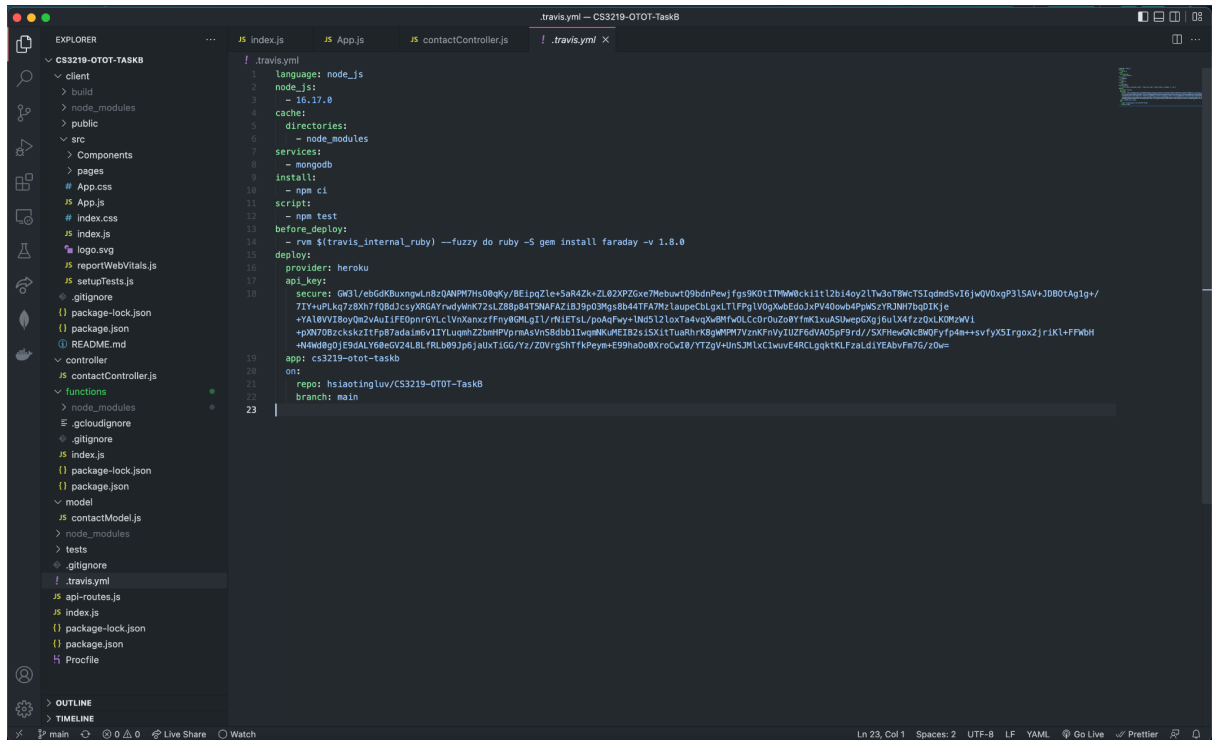
- Continuous Deployment using Travis and Heroku

```
deploy:
  provider: heroku
  api_key:
    secure: GW3l/ebGdKBuxngwLn8zQANPM7HsO0qKy/BEipqZle+5aR4Zk+ZL02XPZGxe7MebuwtQ9bdnPewjfgs9KOtITMWW0cki1tl2bi4oy2lTw3oT8WcTSIqdmdSvI6jwQVOxgP3lSAV+JDBOtAg1g+/
    7IY+uPLkq7z8Xh7fQBdJcsyXRGAYrwdyWnK72sLZ88p84T5NAFAZiBJ9pO3Mgs8b44TFA7MzlaupeCbLgxLTlFPglVOgXwbEdoJxPV4Oowb4PpWSzYRJNH7bqDIKje
    +YAl0VVI8oyQm2vAuIiFEOpnrGYLclVnXanxzfFny0GMLgIl/rNiETsL/poAqFwy+lNd5l2loxTa4vqXwBMfwOLCcOrOuZo0YfmK1xuASUwepGXgj6ulX4fzzQxLKOMzWVi
    +pXN7OBzckskzItFp87adaim6v1IYLuqmhZ2bmHPVprmAsVnS8dbb1IwqmNKuMEIB2siSXitTuaRhrK8gWMPM7VznKFnVyIUZF6dVAO5pF9rd//SXFHewGNcBWQFyfp4m++svfyX5Irgox2jriKl+FFWbH
    +N4Wd0gOjE9dALY60eGV24L8LfRLb09Jp6jaUxTiGG/Yz/ZOVrgShTfkPeym+E99haOo0XroCwI0/YTZgV+UnSJMlxC1wuvE4RCLgqktKLFzaLdiYEAbvFm7G/zOw=
  app: cs3219-otot-taskb
  on:
    repo: hsiaotingluv/CS3219-OTOT-TaskB
    branch: main
```

  ○ run deploy on heroku on travis to automate deployment to heroku
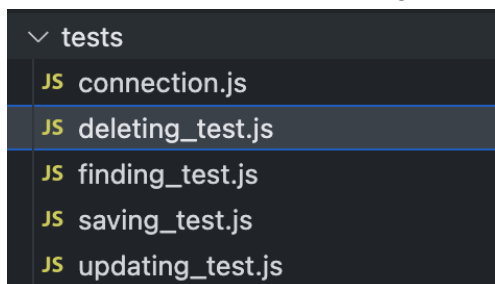
# TaskB2.1: Run tests locally and via Travis/other CI tool

- Create .travis.yml file



- Create separate files for testing

- To run test, run "npm test" in the root directory
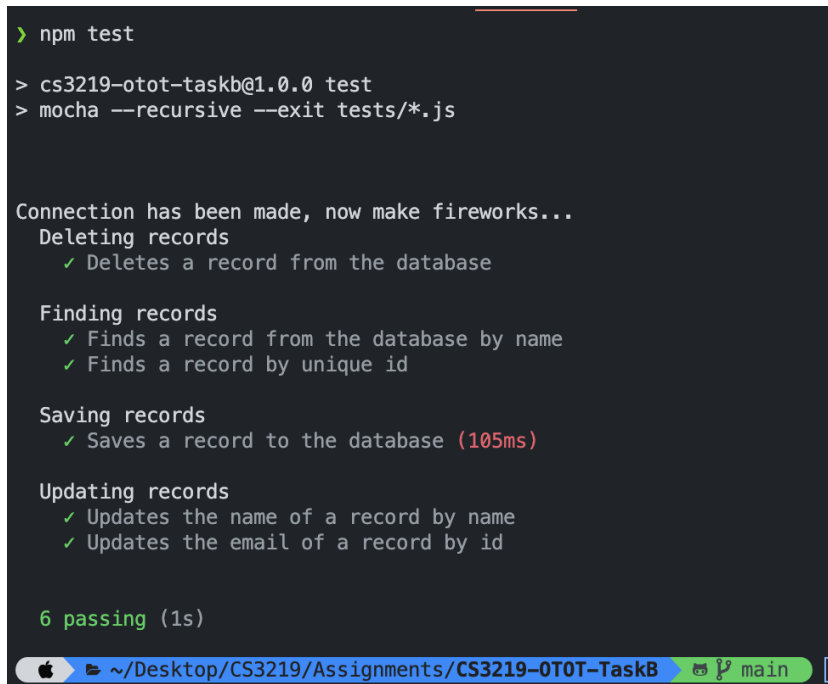


- Continuous Integration whenever commit is pushed to github

# TaskB3: Set up frontend (B3)

- run "npx create-react-app client" to set up frontend



- fetch data from backend api (i.e. fetch("/api/contacts")) and display in frontend through app.js

## Contact List

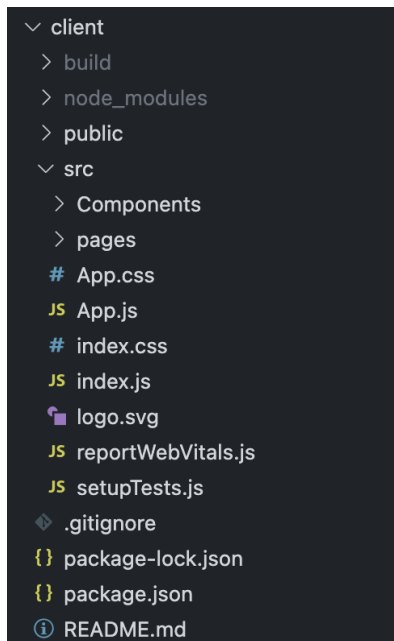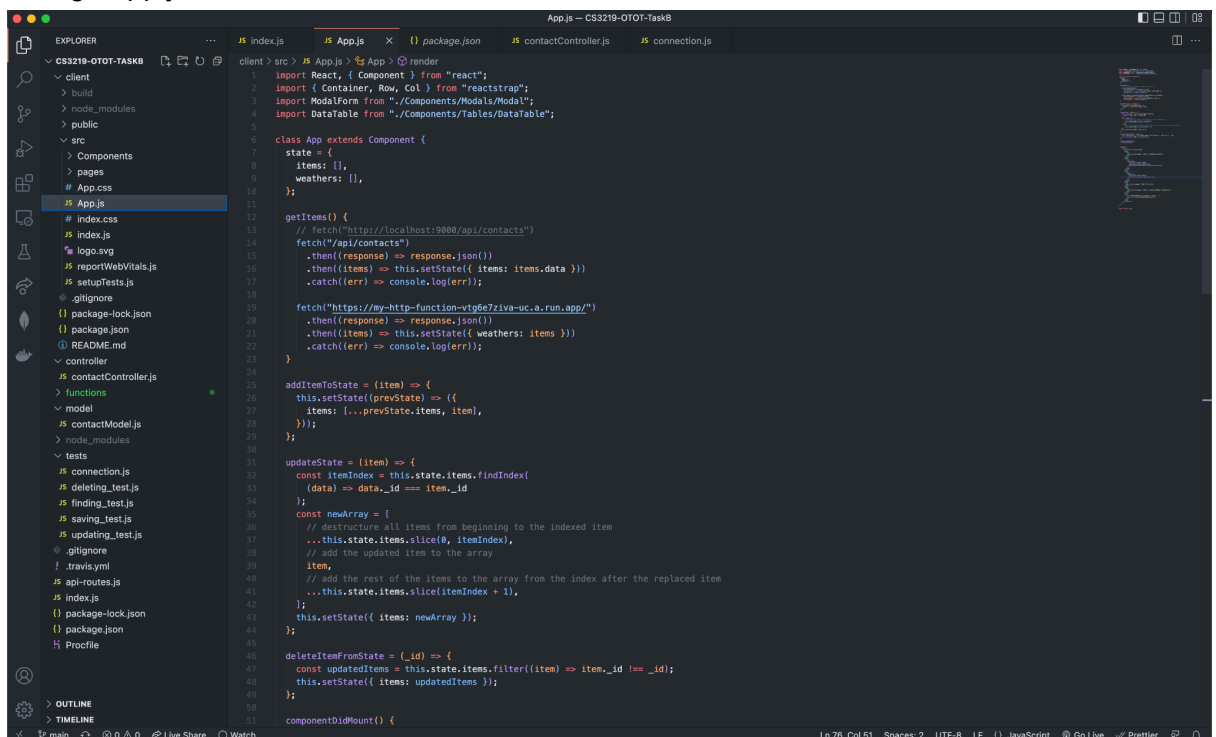| Name | Gender | Email | Phone | | |
|------|--------|-------|-------|---|---|
| John Doe | Male | johndoe@gmail.com | 98568827 | Edit | Del |
| Hsiao Ting | Female | hsiaoting0919@gmail.com | 12345678 | Edit | Del |
| Bob | Male | bob@gmail.com | 99917232 | Edit | Del |
| Gary | Male | gary737@gmail.com | 92994567 | Edit | Del |
| Sandy | Female | sandy@gmail.com | 89173723 | Edit | Del |
| Maria | Female | maria@gmail.com | 75730294 | Edit | Del |
| Sarah | Female | sarah@gmail.com | 92613712 | Edit | Del |
| Josh | Male | josh183@gmail.com | 86889399 | Edit | Del |
| Ian | Male | ian836@gmail.com | 85668399 | Edit | Del |
| Timothy | Male | tim919@gmail.com | 98568827 | Edit | Del |
| Jane | Female | jane88@gmail.com | 74629462 | Edit | Del |
| Test123 | | test123@gmail.com | | Edit | Del |

Add Contact

# TaskB4: A brief explanation of what your serverless function does

- fetch weather forecasts data for the next 4 days from "https://api.data.gov.sg/v1/environment/4-day-weather-forecast"

```
functions > JS index.js > functions.http("weatherGET") callback > then() callback
1   const functions = require("@google-cloud/functions-framework");
2   const axios = require("axios");
3
4   functions.http("weatherGET", (req, res) => {
5     res.set("Access-Control-Allow-Origin", "*");
6     res.set("Access-Control-Allow-Methods", "GET, POST");
7     axios
8       .get("https://api.data.gov.sg/v1/environment/4-day-weather-forecast")
9       .catch((err) => {
10        res.send(err);
11      })
12      .then((result) => {
13        res.send(result.data.items[0].forecasts.map((item) => item.forecast));
14      });
15  });
16
```

- get result from the API, loop through the list of data items and keep only the forecast data

TaskB4: Screenshots of the front-end interaction with your serverless function

# 4-Day Weather Forecasts

1. Afternoon thundery showers
2. Pre-dawn hours and early morning thundery showers
3. Pre-dawn hours and early morning thundery showers
4. Afternoon thundery showers