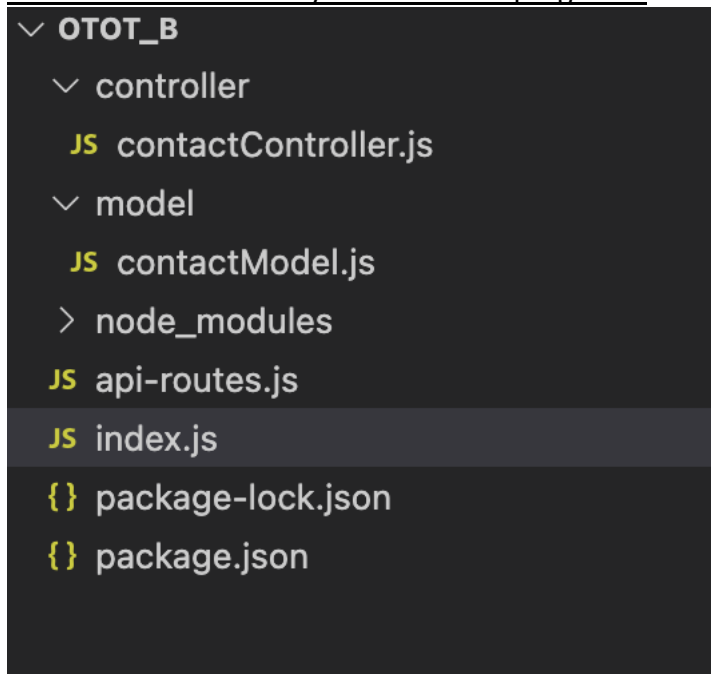


Name: Justin Yip Jeng Teng Matric No.: A0222764E

Github link: <https://github.com/justinyjt/OTOT-B>

B1:

Screenshots of directory structure and programs:



Directory structure



index.js

```

1 // Initialize express router
2 let router = require('express').Router();
3 // Set default API response
4 router.get('/', function (req, res) {
5   res.json({
6     status: 'API Its Working',
7     message: 'Welcome to OTOT_B1 crafted with love!',
8   });
9 });
10 // Import contact controller
11 var contactController = require('./controller/contactController');
12 // Contact routes
13 router.route('/contacts')
14   .get(contactController.index)
15   .post(contactController.new);
16 router.route('/contacts/:contact_id')
17   .get(contactController.view)
18   .patch(contactController.update)
19   .put(contactController.update)
20   .delete(contactController.delete);
21 // Export API routes
22 module.exports = router;

```

api-routes.js

```

1 var mongoose = require('mongoose');
2 // Setup schema
3 var contactSchema = mongoose.Schema({
4   name: {
5     type: String,
6     required: true
7   },
8   email: {
9     type: String,
10    required: true
11  },
12  gender: String,
13  phone: String,
14  create_date: {
15    type: Date,
16    default: Date.now
17  }
18 });
19 // Export Contact model
20 var Contact = module.exports = mongoose.model('contact', contactSchema);
21 module.exports.get = function (callback, limit) {
22   Contact.find(callback).limit(limit);
23 }

```

contactModel.js

```

1 # pull the Node.js Docker image
2 FROM node:alpine
3
4 # create the directory inside the container
5 WORKDIR /usr/app
6
7 # copy the package.json files from local machine to the workdir in container
8 COPY package*.json ./
9
10 # run npm install in our local machine
11 RUN npm install
12
13 # copy the generated modules and all other files to the container
14 COPY . .
15
16 EXPOSE 8080
17
18 # the command that starts our app
19 CMD ["node", "index.js"]

```

Dockerfile

controller > JS contactController.js > update > update > Contact.findById() callback

```
1 // Import contact model
2 Contact = require('../model/contactModel');
3 // Handle index actions
4 exports.index = function (req, res) {
5     Contact.get(function (err, contacts) {
6         if (err) {
7             res.json({
8                 status: "error",
9                 message: err,
10            });
11        }
12        res.json({
13            status: "success",
14            message: "Contacts retrieved successfully",
15            data: contacts
16        });
17    });
18 };
19 // Handle create contact actions
20 exports.new = function (req, res) {
21     var contact = new Contact();
22     contact.name = req.body.name ? req.body.name : contact.name;
23     contact.gender = req.body.gender;
24     contact.email = req.body.email;
25     contact.phone = req.body.phone;
26     // save the contact and check for errors
27     contact.save(function (err) {
28         // if (err)
29         //     res.json(err);
30     res.json({
31         message: 'New contact created!',
32         data: contact
33     });
34 });
35 };
36 // Handle view contact info
37 exports.view = function (req, res) {
38     Contact.findById(req.params.contact_id, function (err, contact) {
39         if (err)
40             res.send(err);
41         res.json({
42             message: 'Contact details loading..',
43             data: contact
44         });
45     });
46 };
47 // Handle update contact info
48 exports.update = function (req, res) {
49     console.log(req.params.contact_id);
50     Contact.findById(req.params.contact_id, function (err, contact) {
51         if (err)
52             res.send(err);
53         contact.name = req.body.name ? req.body.name : contact.name;
54         contact.gender = req.body.gender;
55         contact.email = req.body.email;
56         contact.phone = req.body.phone;
57         // save the contact and check for errors
58         contact.save(function (err) {
59             if (err)
60                 res.json(err);
61             res.json({
62                 message: 'Contact Info updated',
63                 data: contact
64             });
65         });
66     });
67 };
68 // Handle delete contact
69 exports.delete = function (req, res) {
70     Contact.remove({
71         _id: req.params.contact_id
72     }, function (err, contact) {
73         if (err)
74             res.send(err);
75     res.json({
76         status: "success",
77         message: 'Contact deleted'
78     });
79 });
80 };
```

contactController.js

Successful API calls on localhost via postman:

POST ⌵ http://localhost:8080/api/contacts Send ⌵

Params Authorization Headers (8) **Body** ● Pre-request Script Tests Settings Cookies

● none ● form-data ● x-www-form-urlencoded ● raw ● binary ● GraphQL

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/>	name	justin			
<input checked="" type="checkbox"/>	email	justin@gmail.com			
<input checked="" type="checkbox"/>	gender	male			
<input checked="" type="checkbox"/>	phone	99999999			
	Key	Value	Description		

Body Cookies Headers (7) Test Results 200 OK 51 ms 439 B Save Response ⌵

Pretty Raw Preview Visualize JSON ⌵ ⌵

```
1  {
2    "message": "New contact created!",
3    "data": {
4      "_id": "631e97090f1db21142b2947b",
5      "create_date": "2022-09-12T02:18:49.077Z",
6      "name": "justin",
7      "gender": "male",
8      "email": "justin@gmail.com",
9      "phone": "99999999",
10     "__v": 0
11   }
}
```

POST request to insert contact

GET ⌵ http://localhost:8080/api/contacts Send ⌵

Params Authorization Headers (6) **Body** Pre-request Script Tests Settings Cookies

Query Params

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
	Key	Value	Description		

Body Cookies Headers (7) Test Results 200 OK 22 ms 471 B Save Response ⌵

Pretty Raw Preview Visualize JSON ⌵ ⌵

```
1  {
2    "status": "success",
3    "message": "Contacts retrieved successfully",
4    "data": [
5      {
6        "_id": "631e97090f1db21142b2947b",
7        "create_date": "2022-09-12T02:18:49.077Z",
8        "name": "justin",
9        "gender": "male",
10       "email": "justin@gmail.com",
11       "phone": "99999999",
12       "__v": 0
13     }
14   ]
15 }
```

GET request to retrieve contacts

PUT ⌵ http://localhost:8080/api/contacts/631e97090f1db21142b2947b Send ⌵

Params Authorization Headers (8) **Body** ● Pre-request Script Tests Settings Cookies </>

● none ● form-data ● x-www-form-urlencoded ● raw ● binary ● GraphQL i 💡

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/>	name	john			
<input checked="" type="checkbox"/>	email	john@gmail.com			
<input checked="" type="checkbox"/>	gender	female			
<input checked="" type="checkbox"/>	phone	999			
	Key	Value	Description		

Body Cookies Headers (7) Test Results 🌐 200 OK 19 ms 432 B Save Response ⌵

Pretty Raw Preview Visualize JSON ⌵ 📄 🔍

```
1  {
2    "message": "Contact Info updated",
3    "data": {
4      "_id": "631e97090f1db21142b2947b",
5      "create_date": "2022-09-12T02:18:49.077Z",
6      "name": "john",
7      "gender": "female",
8      "email": "john@gmail.com",
9      "phone": "999",
10     "__v": 0
11   }
}
```

PUT call to update contact info

DELETE ⌵ http://localhost:8080/api/contacts/631e97090f1db21142b2947b Send ⌵

Params Authorization Headers (6) **Body** ● Pre-request Script Tests Settings Cookies

Query Params

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
	Key	Value	Description		

Body Cookies Headers (7) Test Results 🌐 200 OK 13 ms 283 B Save Response ⌵

Pretty Raw Preview Visualize JSON ⌵ 📄 🔍

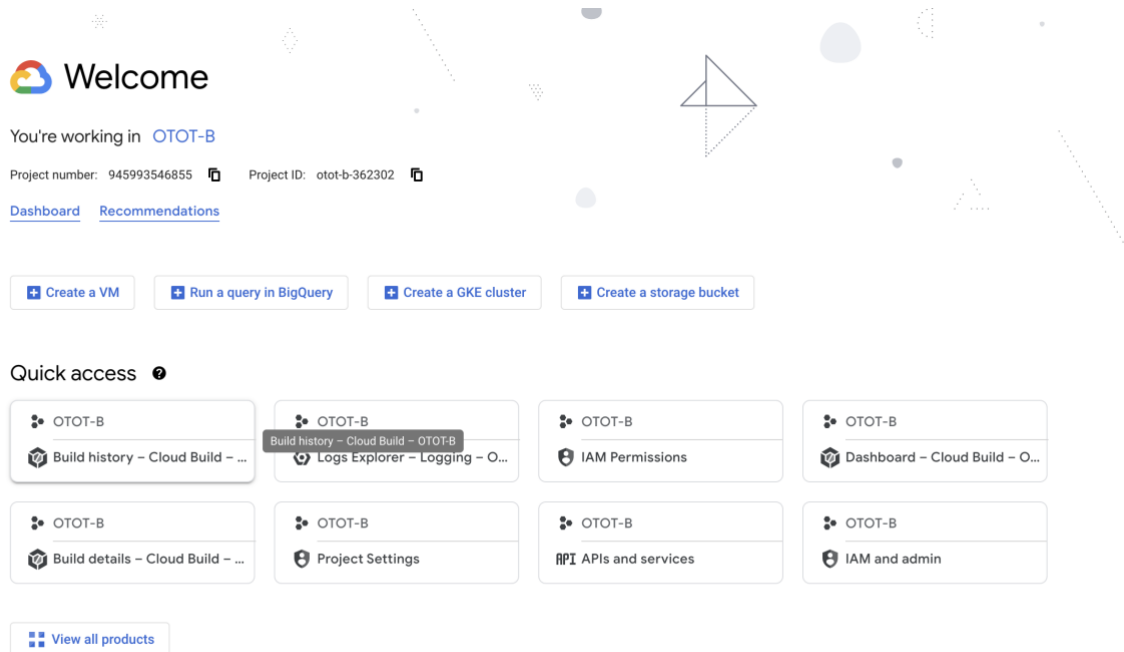
```
1  {
2    "status": "success",
3    "message": "Contact deleted"
4  }
```

DELETE call to delete contact from database

Deployment to GCP:

1. Run the following commands:

- ``gcloud projects list`` to get project name
- ``gcloud set project <project name>``
- ``gcloud config set run/region asia-southeast1``
- ``gcloud run deploy <service-name> --source .``



```
Running OTOT_B on port 8080
^C
(base) justinyip@Justins-MacBook-Pro OTOT_B % gcloud projects list
PROJECT_ID      NAME      PROJECT_NUMBER
otot-b-362302   OTOT-B    945993546855
starry-journal-362302 My First Project 677943302709
(base) justinyip@Justins-MacBook-Pro OTOT_B % gcloud config set project otot-b-362302
Updated property [core/project].
(base) justinyip@Justins-MacBook-Pro OTOT_B % gcloud run deploy otot-b --source .
This command is equivalent to running `gcloud builds submit --tag [IMAGE] .` and `gcloud run deploy otot-b --image [IMAGE]`

Building using Dockerfile and deploying container to Cloud Run service [otot-b] in project [otot-b-362302] region [asia-southeast1]
✓ Building and deploying... Done.
  ✓ Uploading sources...
  ✓ Building Container... Logs are available at [https://console.cloud.google.com/cloud-build/builds/6706e663-b2ff-4b75-99ad-caf16a35d351?project=945993546855].
  ✓ Creating Revision...
  ✓ Routing traffic...
Done.
Service [otot-b] revision [otot-b-00009-mah] has been deployed and is serving 100 percent of traffic.
Service URL: https://otot-b-bkqcejrzeq-as.a.run.app
```

Successful API calls on deployment via postman:

The image shows a Postman interface for a POST request to the endpoint `https://otot-b-bkqcejrzej-as.a.run.app/api/contacts`. The request is configured with the following body data:

KEY	VALUE	DESCRIPTION
name	justin	
email	justin@gmail.com	
gender	cat	
phone	000000000	
Key	Value	Description

The response is a 200 OK status with a response time of 101 ms and a body size of 651 B. The response body is displayed in JSON format:

```
1 {
2   "message": "New contact created!",
3   "data": {
4     "_id": "631eb9871080fff14d997a1d",
5     "create_date": "2022-09-12T04:45:59.016Z",
6     "name": "justin",
7     "gender": "cat",
8     "email": "justin@gmail.com",
9     "phone": "000000000",
10    "__v": 0
11  }
```

POST

GET

https://otot-b-bkqcejrzej-as.a.run.app/api/contacts

Send

Params

Authorization

Headers (6)

Body

Pre-request Script

Tests

Settings

Cookies

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

This request does not have a body

Body

Cookies

Headers (8)

Test Results

200 OK

35 ms

988 B

Save Response

Pretty

Raw

Preview

Visualize

JSON

```
2  "status": "success",
3  "message": "Contacts retrieved successfully",
4  "data": [
5    {
6      "_id": "631eb8a6b81c69e26ab1db62",
7      "create_date": "2022-09-12T04:42:14.628Z",
8      "name": "justin",
9      "gender": "male",
10     "email": "justin@gmail.com",
11     "phone": "99999999",
12     "__v": 0
13   },
14   {
15     "_id": "631eb962ceaa249a90c215c6",
16     "create_date": "2022-09-12T04:45:22.507Z",
17     "name": "justin",
18     "gender": "cat",
19     "email": "justin@gmail.com",
20     "__v": 0
21   },
22 ]
```

GET

PUT

https://otot-b-bkqcejrzej-as.a.run.app/api/contacts/631eb8a6b81c69e26ab1db62

Send

ParamsAuthorizationHeaders (8)BodyPre-request ScriptTestsSettingsCookies

noneform-datax-www-form-urlencodedrawbinaryGraphQL

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/>	name	john			
<input checked="" type="checkbox"/>	email	asf			
	Key	Value	Description		

BodyCookiesHeaders (8)Test Results200 OK60 ms601 BSave Response

PrettyRawPreviewVisualizeJSON

```
1
2  "message": "Contact Info updated",
3  "data": {
4    "_id": "631eb8a6b81c69e26ab1db62",
5    "create_date": "2022-09-12T04:42:14.628Z",
6    "name": "john",
7    "email": "asf",
8    "__v": 0
9  }
10
```

PUT

GCP DEPLOYMENT / New Request

Save

DELETE

https://otot-b-bkqcejrzej-as.a.run.app/api/contacts/631eb8a6b81c69e26ab1db62

Send

ParamsAuthorizationHeaders (6)BodyPre-request ScriptTestsSettingsCookies

noneform-datax-www-form-urlencodedrawbinaryGraphQL

This request does not have a body

BodyCookiesHeaders (6)Test Results200 OK60 ms495 BSave Response

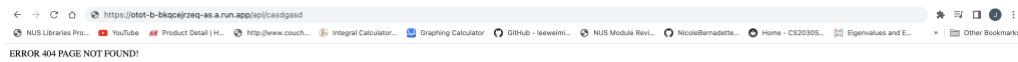
PrettyRawPreviewVisualizeJSON

```
1
2  "status": "success",
3  "message": "Contact deleted"
4
```

DELETE

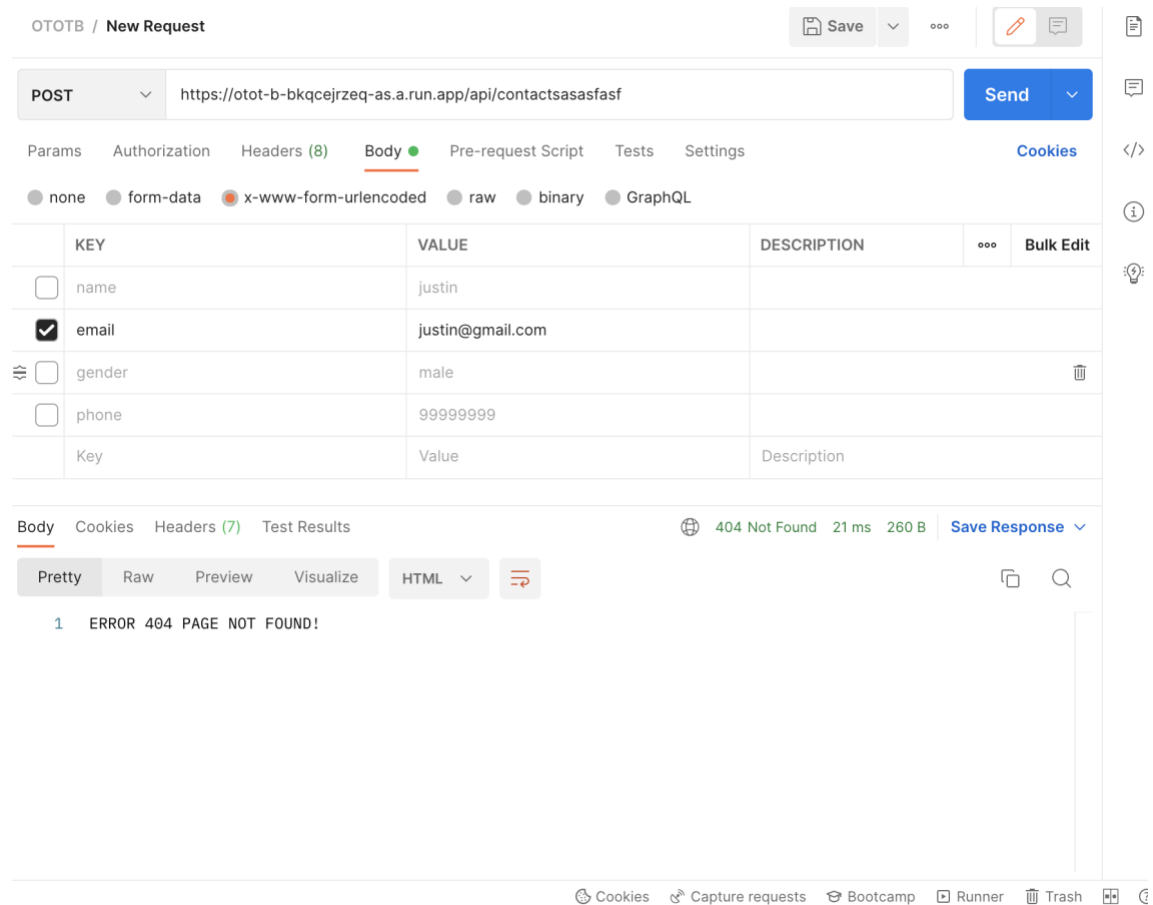
Common edge cases and error-resiliency:

1.



Return error 404 when user tries to access (GET) an invalid url.

2.



Return error 404 when user tries to POST to invalid url

3.

The screenshot displays a REST client interface for a POST request to the endpoint `https://otot-b-bkqcejrzej-as.a.run.app/api/contacts`. The request is configured with the `x-www-form-urlencoded` body type. The body contains the following data:

KEY	VALUE	DESCRIPTION
<input type="checkbox"/> name	justin	
<input checked="" type="checkbox"/> email	justin@gmail.com	
<input type="checkbox"/> gender	male	
<input type="checkbox"/> phone	99999999	
Key	Value	Description

The response section shows a `401 Unauthorized` status with a response time of `5 ms` and a size of `281 B`. The response body, displayed in JSON format, contains the following message:

```
{  "message": "Missing name or email!"}
```

Return error message when user tries to insert contact without required name/email fields.