CS3219 OTOT Task B

- **Name**: Ryan Aidan
- **Matric. Number**: A0218327E
- **Repo Link**: https://github.com/aidanaden/otot-b

Tech stack used

- Next.JS (frontend + backend API routes via nextjs' serverless functions)
- TailwindCSS (frontend styling)
- Prisma (database ORM)
- MySQL (database used)
- tRPC (used to set up API routes with end-to-end typesafety)

Hosting

- Vercel (provides easy integration with Next.JS + continuous deployment)
- Planetscale (highly scalable serverless MySQL database with generous free plans)

Task B1: Simple Backend

Requirements

1. MySQL (SQL database used)
2. NodeJS >= `v14.19.2`
3. Postman is used to (manually) test CRUD for the API.

Local Deployment

1. Install docker by clicking here

2. Install pnpm by clicking here

3. Copy the `.env-example` file to `.env` and paste the following:

```
# Prisma
DATABASE_URL=mysql://root:root@localhost:3369/mysqldb

# Next Auth
NEXTAUTH_SECRET=secret

# Calendarific
CALENDARIFIC_API_KEY=00e9be3d9730c3d5504e5c4794788bc428cdeea5
```

4. Run the following commands to deploy the app locally

```
# install packages
pnpm install

# start local mysql db, deploy schema, build source files
pnpm build:local
```

```
# run locally
pnpm start
```

Verify successful deployment

Run the following commands to verify successful local deployment

```
# query the /api/activity endpoint
curl http://localhost:3000/api/activity

# expected output
[]
```

Visit `http://localhost:3000` to confirm successful frontend deployment

Testing via Postman

Import the Postman collection via [this link](#).

2 versions of the `API_URL` variable exists, one set to `localhost:3000` (locally deployment) and one set to `https://otot-b-phi.vercel.app` (vercel deployment)
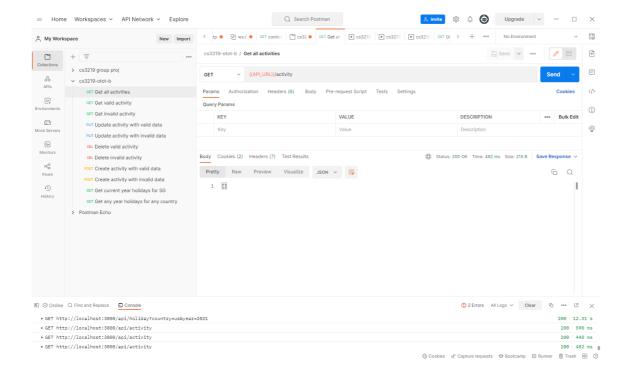
Explanation of HTTP requests:

- `activity` : list all activities in database (will be empty if using local MySQL database)
- `activity/:id` : returns error if ID does not exist
  - `GET` : returns activity with the specified ID
  - `PATCH` / `PUT` : updates specified field(s) in request body given an ID
  - `DELETE` : removes the activity with the given ID from the database
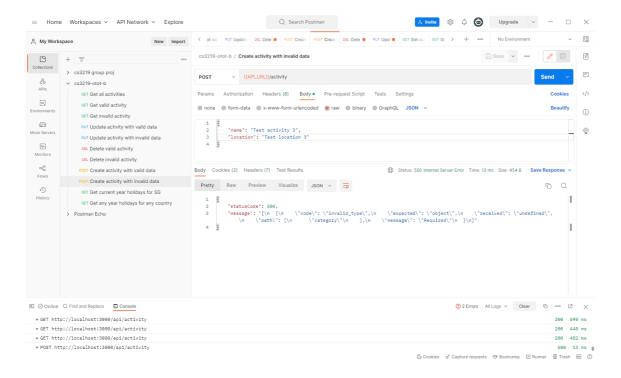
Note:

- `params` and `body` would need to be updated manually, where they are necessary.
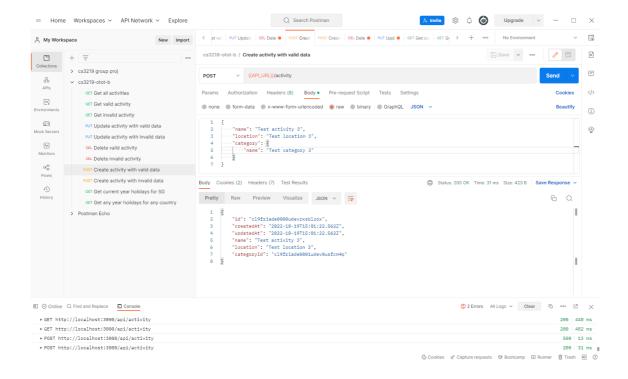
Screenshots
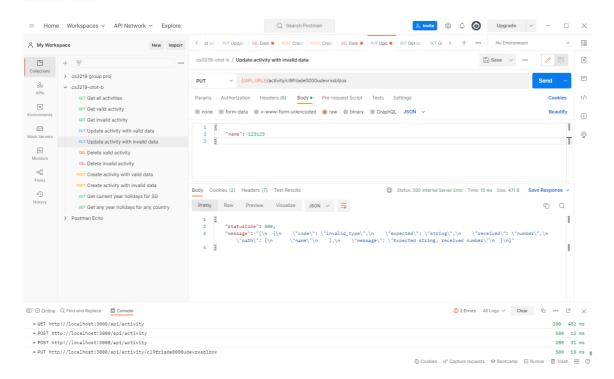
`GET` **: get activities given empty database**

POST : create invalid activity (missing category)
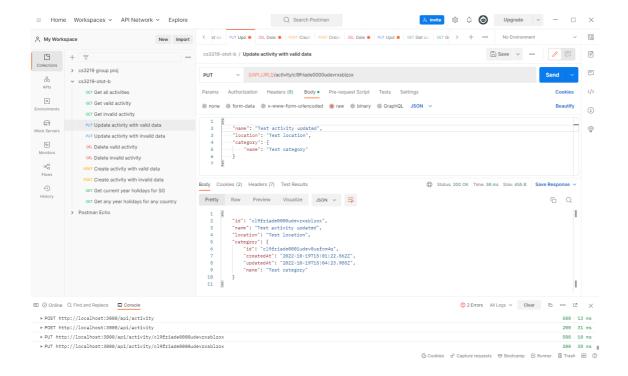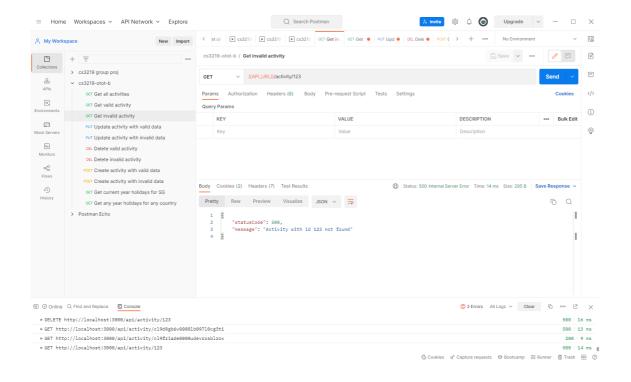

POST : create valid activity

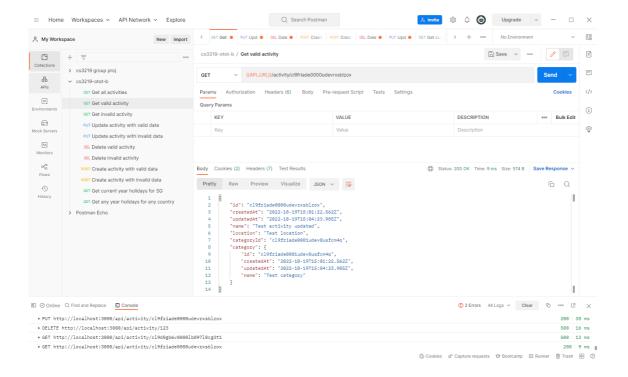**PATCH / PUT : update name of activity with invalid value**

PATCH / PUT : update name of activity with valid value

GET : get invalid activity given invalid id

GET : get valid activity given valid id

DELETE : remove invalid activity with invalid id

DELETE : remove valid activity with valid id

Task B2.1: Testing through Continuous Integration

Jest was used to test the API routes locally as well as via Github Action.

Summary of CI

- [Jest](#) used to test API endpoints
- Local MySQL database used for tests
- Postman collection can be found [here](#) with variables already set
- Github Action yaml file can be viewed [here](#).

Local test output

Testing via **Github Action**

Task B2.2: Deploying through Continuous Deployment

Since vercel is used to host both the frontend + backend, continuous deployment is automatically provided via vercel's github integration

Continuous Deployment Screenshots

Vercel deployment

Deployed via vercel

Task B3: Implement a Frontend

The website is deployed on Vercel and can be accessed through this link https://otot-b-phi.vercel.app/.

The home page automatically loads all of the activities currently stored in the live db hosted in planetscale.

1. To create an "activity", click on the "create activity" button on the top right of the screen.

2. To edit an activity, click on the "edit" button in the top right of the activity.

3. To delete an activity, click on the "delete" button in the top right of the activity.

Screenshot of Activities on frontend

NextDate

New activity +

Activity1 Edit Delete
Category1 Location1

Activity2 Edit Delete
Category2 Location2

Activity3 Edit Delete
Category3 Location3

Activity4 Edit Delete
Category4 Location4

Activity5 Edit Delete
Category5 Location5

Activity6 Edit Delete
Category6 Location6

Activity7 Edit Delete
Category7 Location7

Activity8 Edit Delete
Category8 Location8

Activity9 Edit Delete
Category9 Location9

NextDate

New activity +

Activity1

Edit Delete

Category1 Location1

Activity2

Edit Delete

Category2 Location2

Activity3

Edit Delete

Category3 Location3

Activity4

Edit Delete

Category4 Location4

Activity5

Edit Delete

Category5 Location5

Activity6

Edit Delete

Category6 Location6

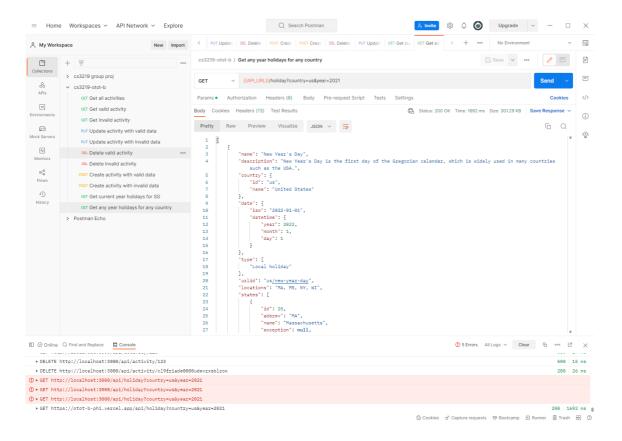# Task B4: Pulling data from Serverless Function to Frontend

The serverless API function will be pulling data from [Calendarific's API](#) to query the holidays in SG for the current year

The serverless function will be run using Next.JS' serverless API routes. When a client queries the API route, the serverless function will query the calendarific API endpoint (given a valid country code and year) and return a list of holidays.

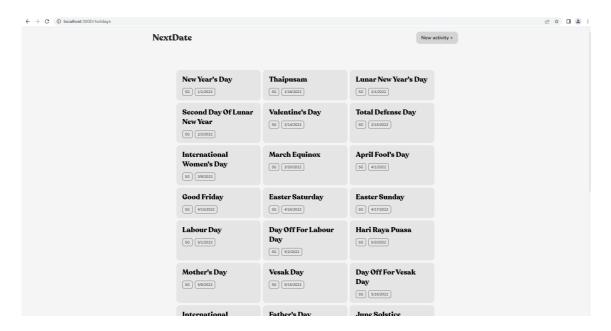To view the holiday data, a page has been created on the frontend to display the holidays returned here https://otot-b-phi.vercel.app/holidays.
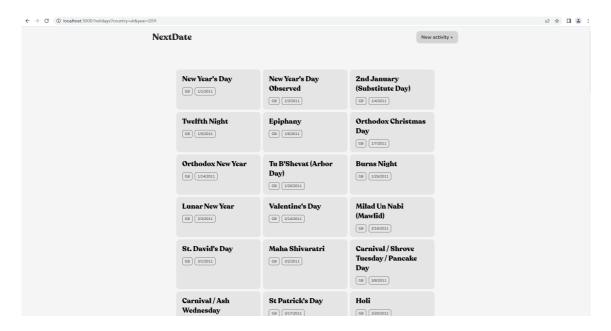
Screenshot of B4

Serverless function API route (via Postman)

View all SG holidays for 2022

View all UK holidays for 2011

NextDate

New activity +

New Year's Day
GB 1/1/2011

New Year's Day Observed
GB 1/3/2011

2nd January (Substitute Day)
GB 1/4/2011

Twelfth Night
GB 1/5/2011

Epiphany
GB 1/6/2011

Orthodox Christmas Day
GB 1/7/2011

Orthodox New Year
GB 1/14/2011

Tu B'Shevat (Arbor Day)
GB 1/20/2011

Burns Night
GB 1/25/2011

Lunar New Year
GB 2/3/2011

Valentine's Day
GB 2/14/2011

Milad Un Nabi (Mawlid)
GB 2/16/2011

St. David's Day
GB 3/1/2011

Maha Shivaratri
GB 3/2/2011

Carnival / Shrove Tuesday / Pancake Day
GB 3/8/2011

Carnival / Ash Wednesday

St Patrick's Day
GB 3/17/2011

Holi
GB 3/20/2011

Potential improvements

In the future, google location API can be integrated to allow location autocomplete when creating activities (instead of having to manually enter the activity location information).