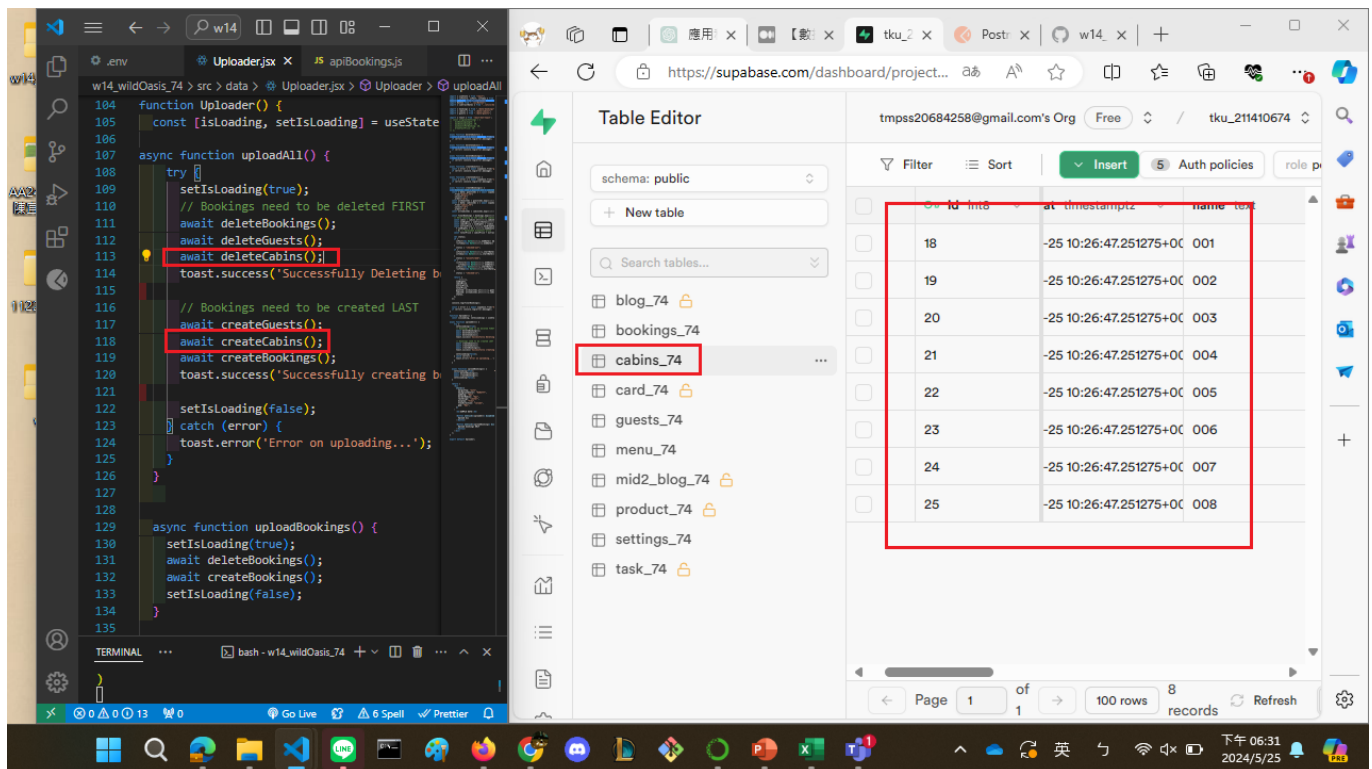
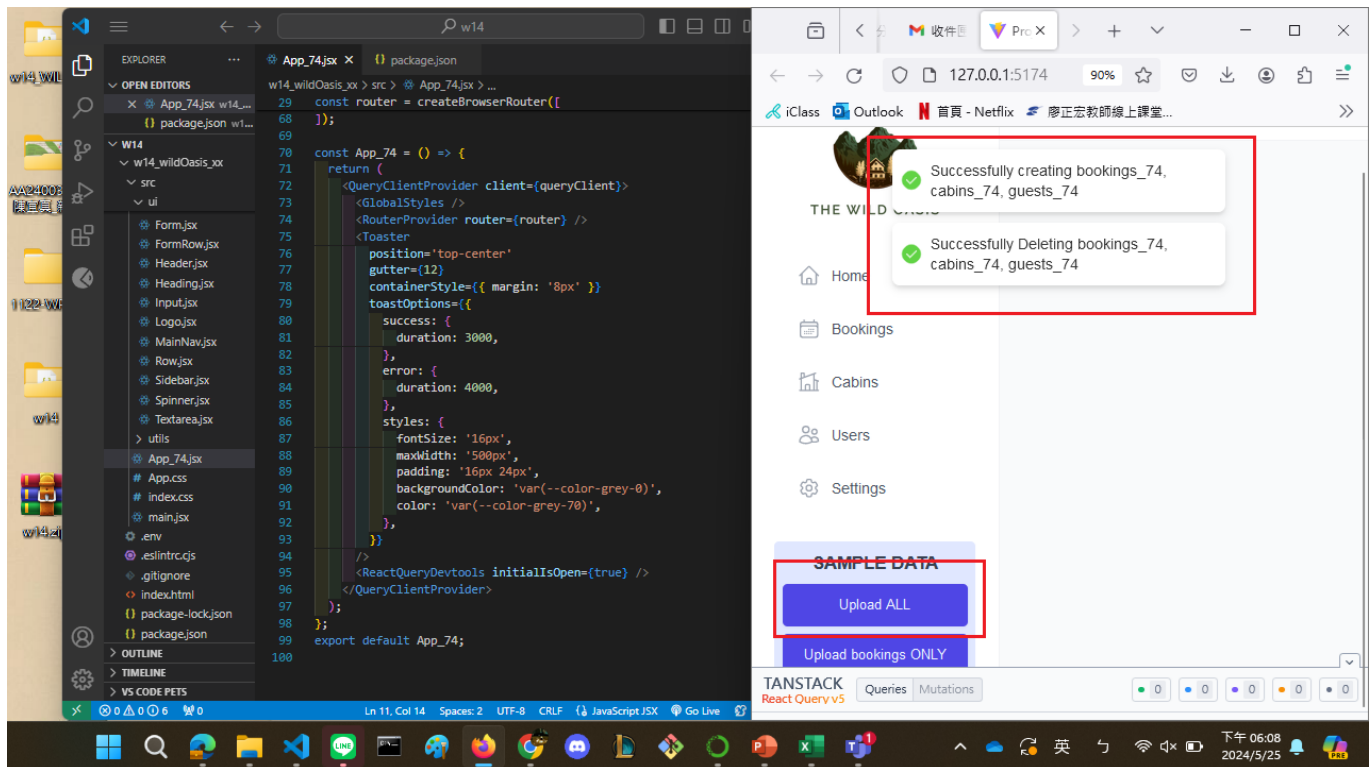


My github URL

W14-P1: Use sql to create schemas of cabins_74, guests_74, bookings_74, settings_74, and use Uploader to upload all data



The screenshot shows a development environment with two main windows. On the left is a code editor displaying the `Upload.jsx` file. The code defines an `Uploader` component with two asynchronous functions: `uploadAll()` and `uploadBookings()`. In `uploadAll()`, `deleteBookings()` and `createBookings()` are called sequentially. In `uploadBookings()`, `createBookings()` is called. The code uses `useState` for `isLoading` and `toast` for notifications. On the right is the Supabase Table Editor for the `public` schema. It shows a list of tables: `blog_74`, `bookings_74`, `cabins_74`, `card_74`, `guests_74`, `menu_74`, `mid2_blog_74`, `product_74`, `settings_74`, and `task_74`. The `bookings_74` table is selected, and its data is displayed in a table with columns `id` and `int4`. The table contains 24 rows of data, with the last row having an `id` of 71 and a value of `round m` in the `int4` column.

The screenshot shows a development environment with four main windows. On the left is the Explorer view showing the project structure. The `data-cabins.js` file is selected. In the center is the `Upload.jsx` file, which is the same as the one in the first screenshot. On the right are two other files: `data-cabins.js` and `data-bookings.js`. The `data-cabins.js` file defines a `cabins` array with two objects, each representing a cabin. The `data-bookings.js` file defines a `bookings` array with two objects, each representing a booking. The `bookings` array is filtered to show only bookings for the cabin with `id: 1`.

W14-P2: Make jonas code work, and show cabins and bookings info

THE WILD OASIS

Home Bookings **Cabins** Users Settings

SAMPLE DATA

Upload ALL

Upload bookings ONLY

CABIN	CAPACITY	PRICE	DISCOUNT
001	Fits up to 2 guests	\$250.00	—
002	Fits up to 2 guests	\$350.00	\$25.00
003	Fits up to 4 guests	\$300.00	—
004	Fits up to 4 guests	\$500.00	\$50.00
005	Fits up to 6 guests	\$350.00	—
006	Fits up to 6 guests	\$800.00	\$100.00
007	Fits up to 8 guests	\$600.00	\$100.00
008	Fits up to 10 guests	\$1,400.00	—

2 ["user"]

1 ["cabins"]

id: "a01bc73e-b060-4d66-a37c-2da879aa2cd0"

aud: "authenticated"

role: "authenticated"

email: "tmpps20684258@gmail.com"

email_confirmed_at: "2024-05-25T12:15:25.402804Z"

phone: ""

confirmed_at: "2024-05-25T12:15:25.402804Z"

last_sign_in_at: "2024-05-25T13:07:25.035899Z"

app_metadata: {}

THE WILD OASIS

Home Bookings **Cabins** Users Settings

SAMPLE DATA

Upload ALL

Upload bookings ONLY

All bookings

All Checked out **Checked in** Unconfirmed

Sort by date (recent first)

CABIN	GUEST	DATES	STATUS	AMOUNT
003	Maria Rodriguez maria@gmail.com	3 days ago → 2 night stay May 23 2024 — May 25 2024	CHECKED IN	\$600.00
004	Khadija Ahmed khadija@gmail.com	5 days ago → 12 night stay May 21 2024 — Jun 02 2024	CHECKED IN	\$6,120.00
008	Jonas Schmedtmann hello@jonas.io	6 days ago → 5 night stay May 20 2024 — May 25 2024	CHECKED IN	\$7,675.00

1 ["bookings", {"field": "status", "value": "checked-in", "direction": "desc", "field": "startDate"}]

2 ["user"]

id: "a01bc73e-b060-4d66-a37c-2da879aa2cd0"

aud: "authenticated"

role: "authenticated"

email: "tmpps20684258@gmail.com"

email_confirmed_at: "2024-05-25T12:15:25.402804Z"

phone: ""

confirmed_at: "2024-05-25T12:15:25.402804Z"

last_sign_in_at: "2024-05-25T13:07:25.035899Z"

app_metadata: {}

user_metadata: {}

identities: {}

created_at: "2024-05-25T12:15:25.397742Z"

updated_at: "2024-05-25T13:25:35.454359Z"

is_anonymous: false

5228170 Chen211410674 Sat May 25 18:56:55 2024 +0800 Use sql to create schemas of cabins_74, guests_74, bookings_74, settings_74, and use Uploader to upload all data完成後兩小題

f639b21 Chen211410674 Sat May 25 18:35:22 2024 +0800 Use sql to create schemas of cabins_74, guests_74, bookings_74, settings_74, and use Uploader to upload all data

The screenshot displays a web browser window with a REST client interface. The browser's address bar shows the URL `127.0.0.1:5173/bookings?status=checked-in`. The REST client interface has a sidebar with a 'bookings' tab selected. The main area shows a GET request to `/bookings?status=checked-in` with a status of 200. The response is a JSON array of booking objects. The first object is highlighted, showing details for a booking by Maria Rodriguez. The REST client interface includes tabs for 'bookings', 'users', and 'cabins', and a 'Data Explorer' on the right showing the JSON structure.

CABIN	GUEST	DATES	STATUS	AMOUNT
003	Maria Rodriguez maria@gmail.com	3 days ago → 2 night stay May 23 2024 — May 25 2024	CHECKED IN	\$600.00
004	Khadija Ahmed khadija@gmail.com	5 days ago → 12 night stay May 21 2024 — Jun 02 2024	CHECKED IN	\$6,120.00
008	Jonas Schmedtmann hello@jonas.io	6 days ago → 5 night stay May 20 2024 — May 25 2024	CHECKED IN	\$7,675.00

W14-P3: Git logs of W14

4 / 5

a66e354 Chen211410674 Sat May 25 21:43:10 2024 +0800 W14-P2: Make jonas code work, and show cabins and bookings info

5228170 Chen211410674 Sat May 25 18:56:55 2024 +0800 Use sql to create schemas of cabins_74, guests_74, bookings_74, settings_74, and use Uploader to upload all data完成後兩小題

f639b21 Chen211410674 Sat May 25 18:35:22 2024 +0800 Use sql to create schemas of cabins_74, guests_74, bookings_74, settings_74, and use Uploader to upload all data