

Test Case Screenshots - TUES 18:30-5

Milestone 2 & 3 Test Case Execution Screenshots

Table of Contents

Test Case Screenshots - TUES 18:30-5	1
Milestone 2 Test Cases (1-10)	2
Test Case ID: 1	2
Test Case ID: 2	4
Test Case ID: 3	5
Test Case ID: 4	6
Test Case ID: 5	7
Test Case ID: 6	8
Test Case ID: 7	9
Test Case ID: 8	10
Test Case ID: 9	10
Test Case ID: 10	11
Milestone 3 Test Cases (11-30)	12
Test Case ID: 11	12
Test Case ID: 12	14
Test Case ID: 13	15
Test Case ID: 14	15
Test Case ID: 15	17
Test Case ID: 16	19
Test Case ID: 17	20
Test Case ID: 18	22
Test Case ID: 19	24
Test Case ID: 20	25
Test Case ID: 21	26
Test Case ID: 22	28
Test Case ID: 23	29
Test Case ID: 24	31
Test Case ID: 25	32
Test Case ID: 26	34
Test Case ID: 27	35
Test Case ID: 28	37
Test Case ID: 29	38
Test Case ID: 30	39

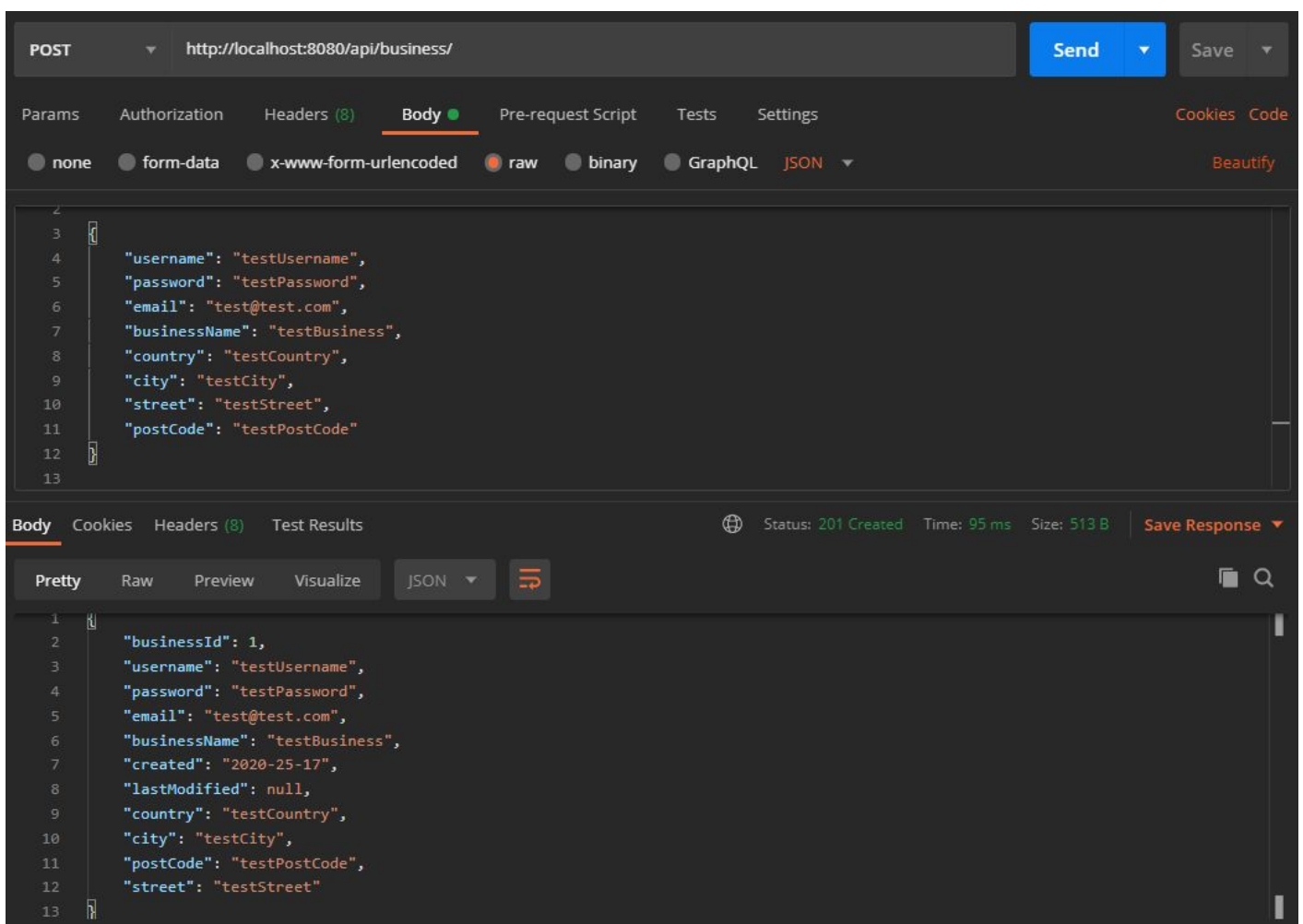
Milestone 2 Test Cases (1-10)

Test Case ID: 1

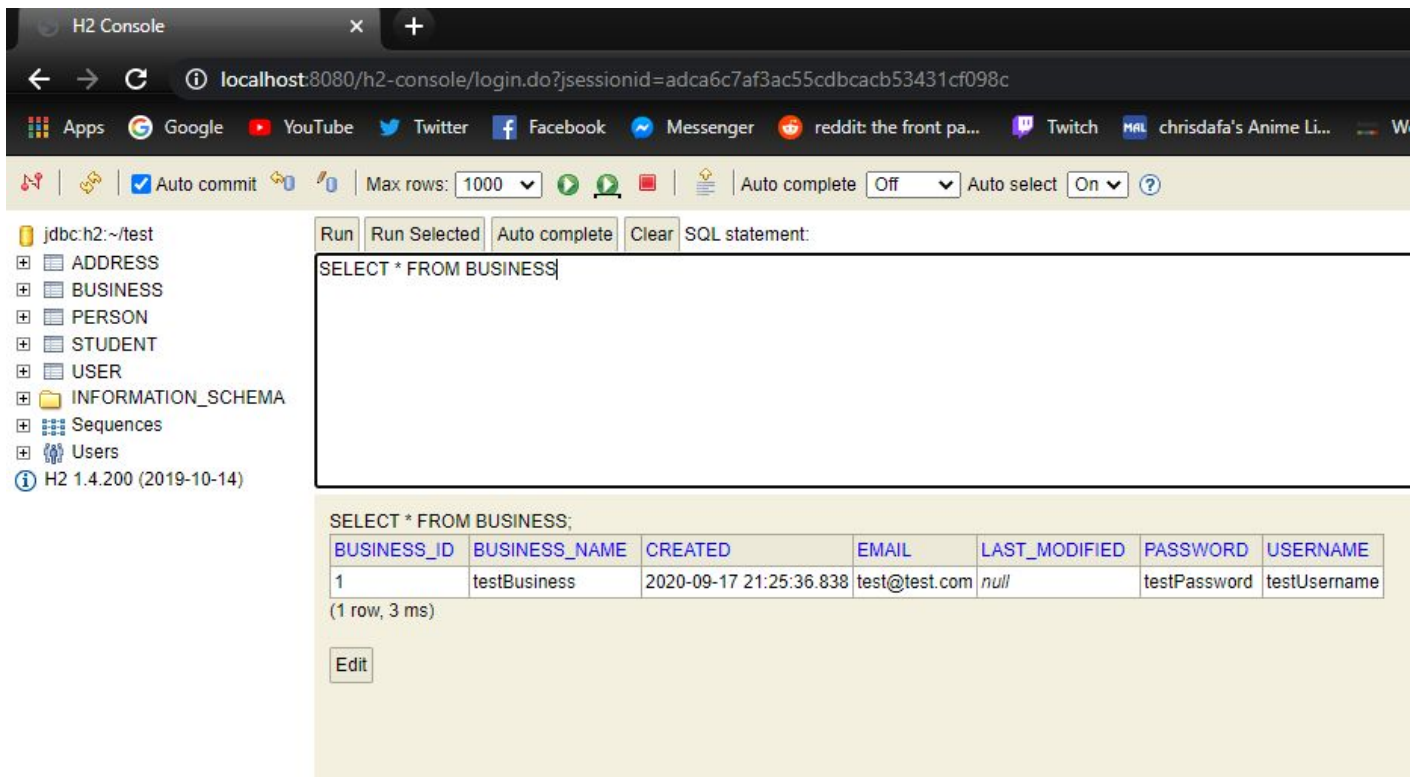
Test Case Description: Submit a post request for business to Backend via Postman

Test Scenario: Check Create Business Functionality

Postman:

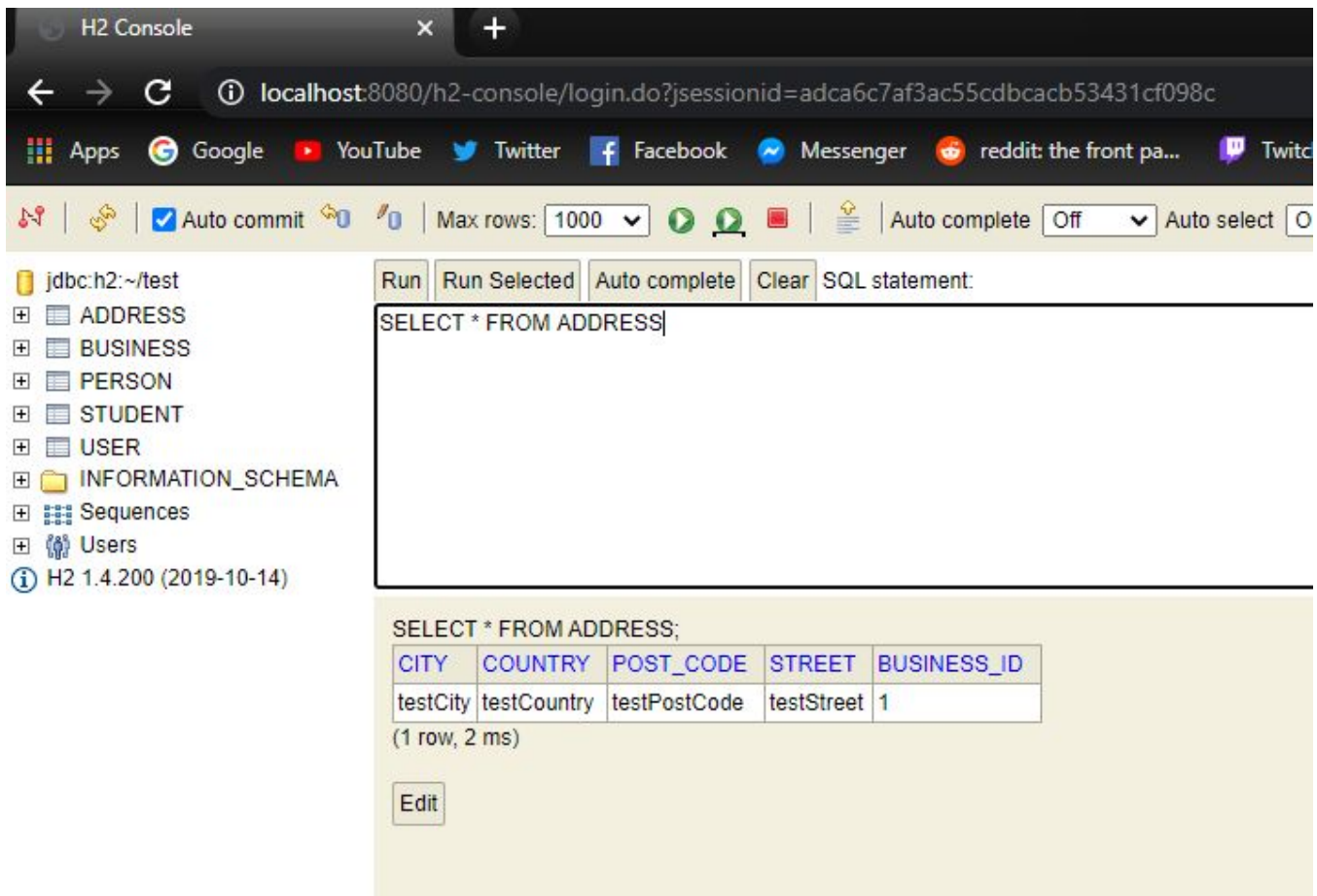


H2 database:



The screenshot shows the H2 Console interface. The left sidebar displays the database schema with tables: ADDRESS, BUSINESS, PERSON, STUDENT, USER, and INFORMATION_SCHEMA. The main area shows the SQL statement `SELECT * FROM BUSINESS;` executed. The results are displayed in a table with 7 columns: BUSINESS_ID, BUSINESS_NAME, CREATED, EMAIL, LAST_MODIFIED, PASSWORD, and USERNAME. The first row contains the values: 1, testBusiness, 2020-09-17 21:25:36.838, test@test.com, null, testPassword, and testUsername. The execution took 3 ms.

BUSINESS_ID	BUSINESS_NAME	CREATED	EMAIL	LAST_MODIFIED	PASSWORD	USERNAME
1	testBusiness	2020-09-17 21:25:36.838	test@test.com	null	testPassword	testUsername



The screenshot shows the H2 Console interface. The left sidebar displays the database schema with tables: ADDRESS, BUSINESS, PERSON, STUDENT, USER, and INFORMATION_SCHEMA. The main area shows the SQL statement `SELECT * FROM ADDRESS;` executed. The results are displayed in a table with 5 columns: CITY, COUNTRY, POST_CODE, STREET, and BUSINESS_ID. The first row contains the values: testCity, testCountry, testPostCode, testStreet, and 1. The execution took 2 ms.

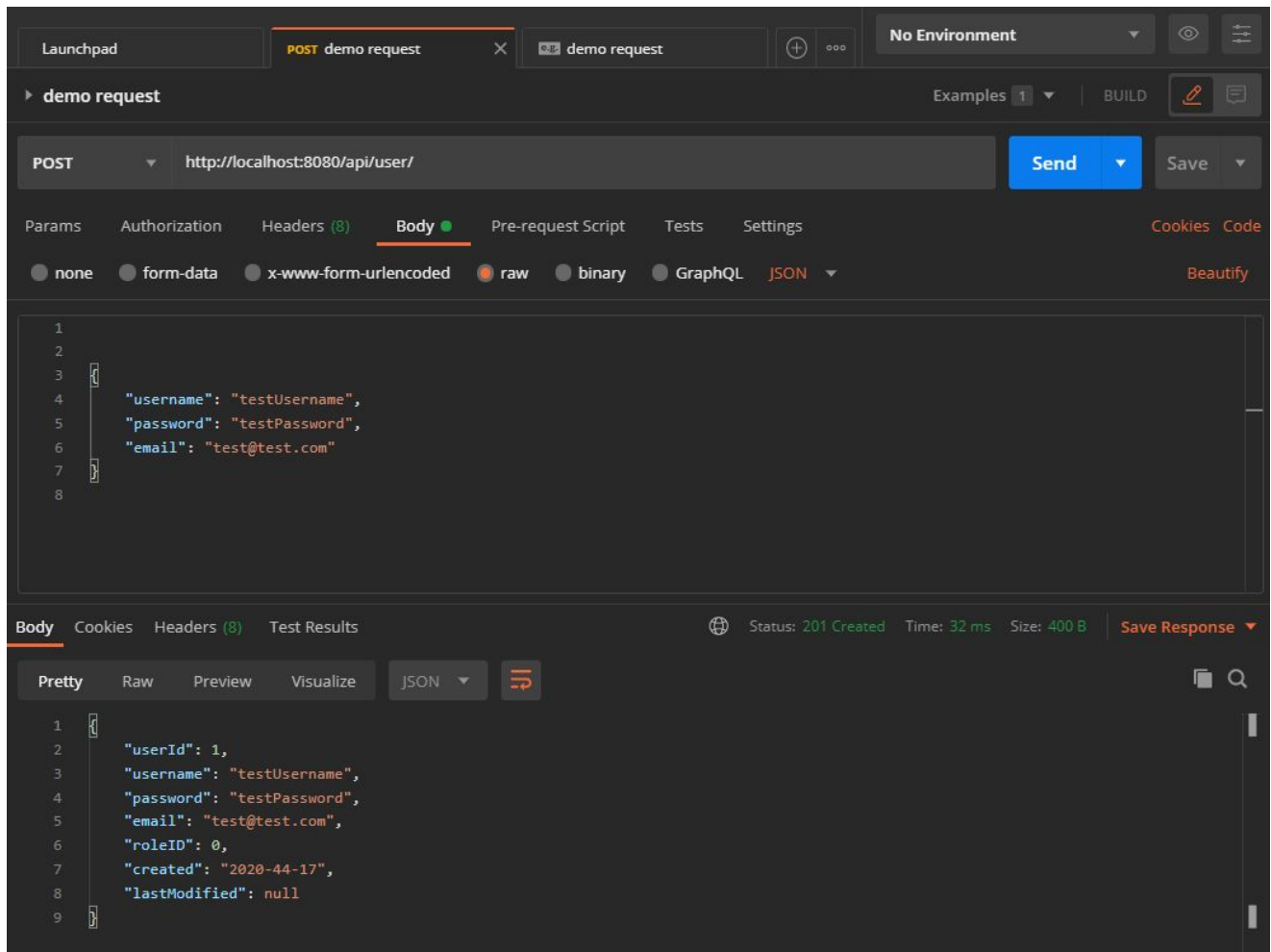
CITY	COUNTRY	POST_CODE	STREET	BUSINESS_ID
testCity	testCountry	testPostCode	testStreet	1

Test Case ID: 2

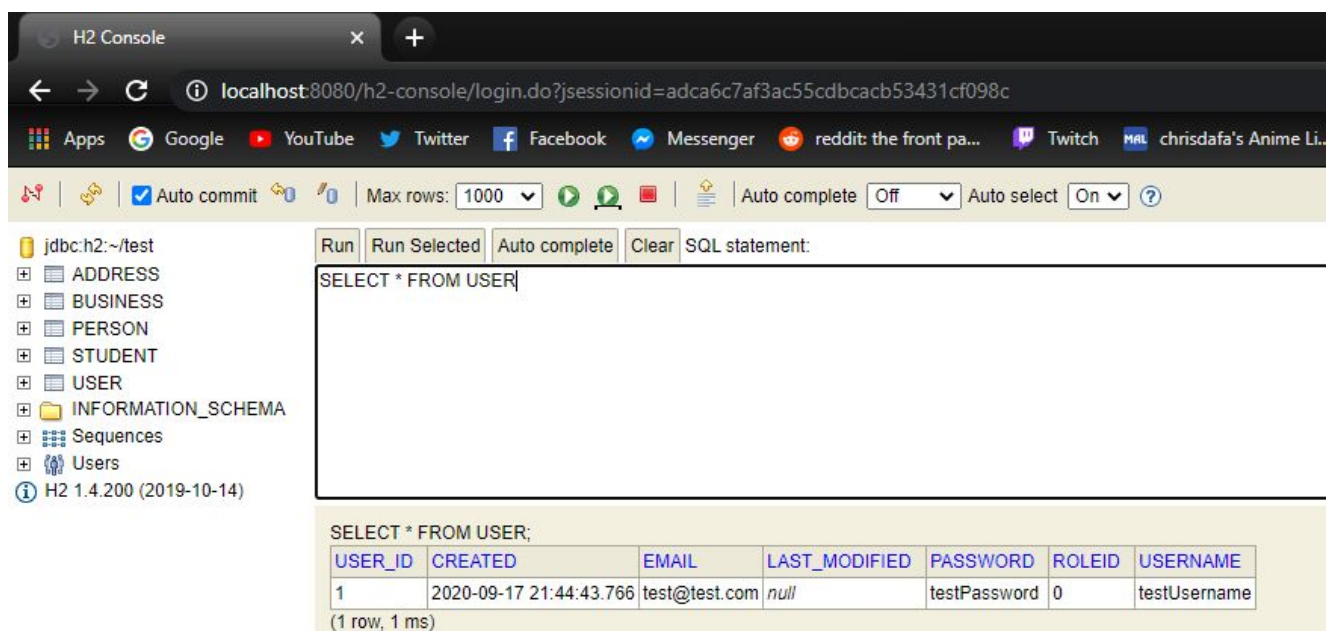
Test Case Description: Submit a post request for user to Backend via Postman

Test Scenario: Check Create User Functionality

Postman:



H2 database:

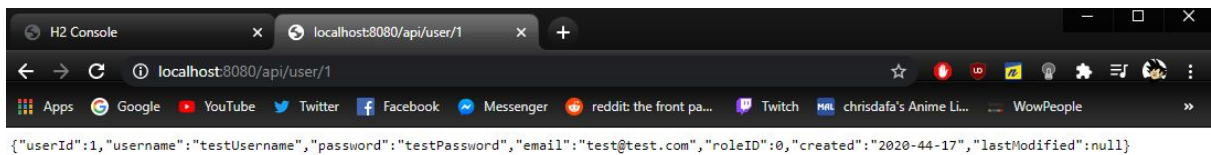


Test Case ID: 3

Test Case Description: send a get request for user to Backend via Postman

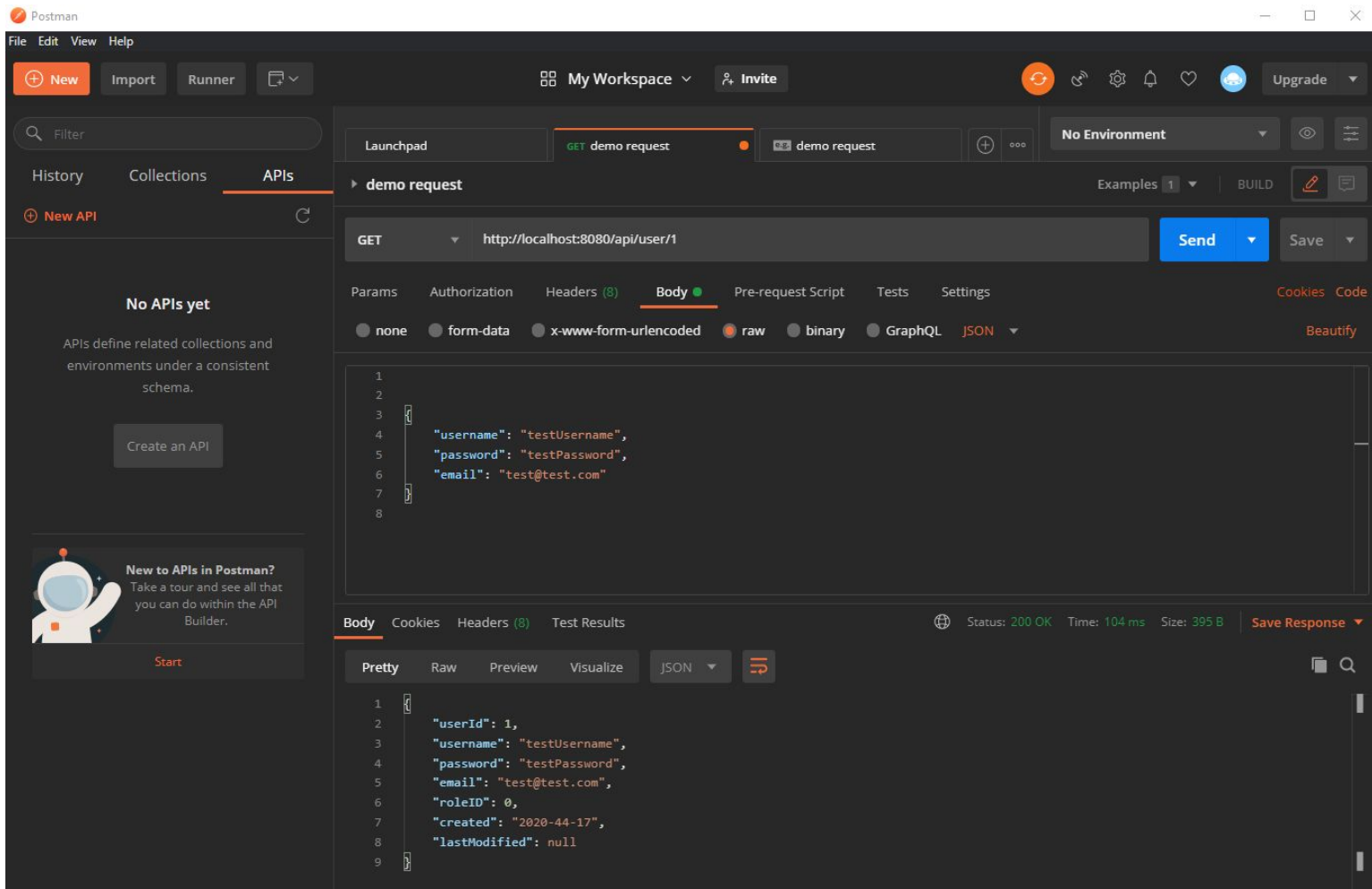
Test Scenario: Check Get User Functionality

User details returned:



```
{ "userId": 1, "username": "testUsername", "password": "testPassword", "email": "test@test.com", "roleID": 0, "created": "2020-44-17", "lastModified": null }
```

Postman:

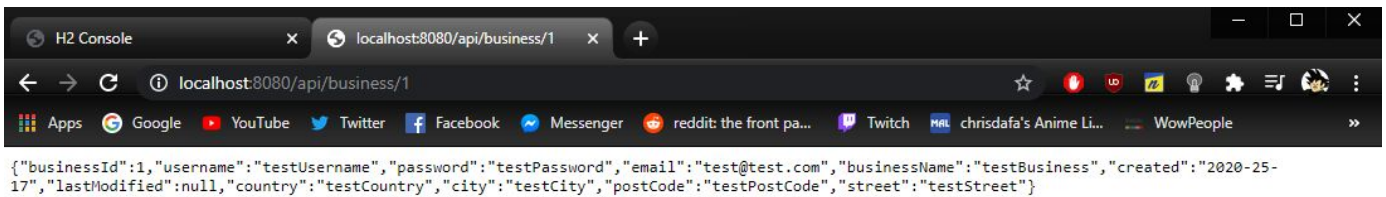


Test Case ID: 4

Test Case Description: send a get request for business to Backend via Postman

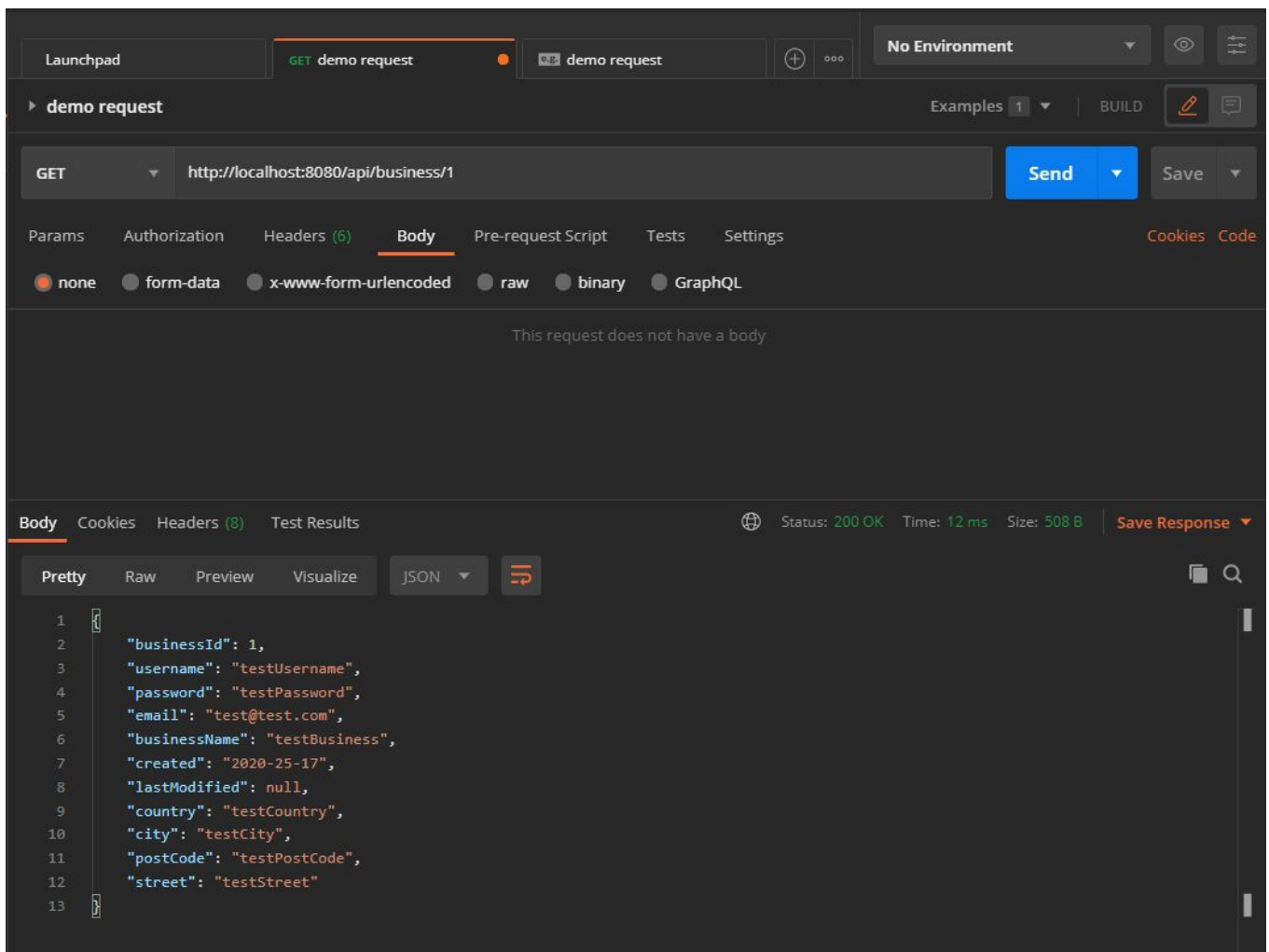
Test Scenario: Check Get Business Functionality

Business details returned:



```
{
  "businessId": 1,
  "username": "testUsername",
  "password": "testPassword",
  "email": "test@test.com",
  "businessName": "testBusiness",
  "created": "2020-25-17",
  "lastModified": null,
  "country": "testCountry",
  "city": "testCity",
  "postCode": "testPostCode",
  "street": "testStreet"
}
```

Postman:

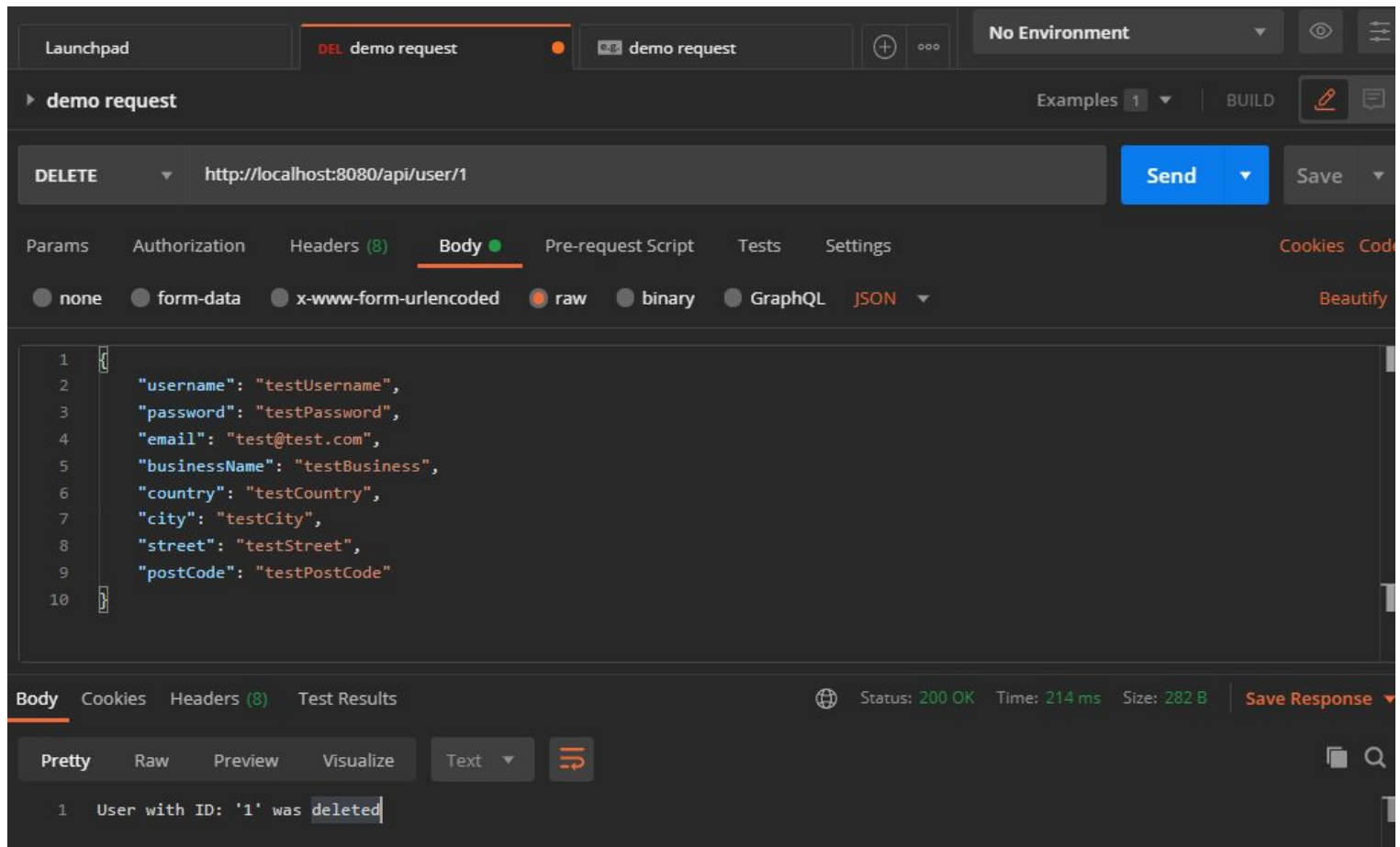


Test Case ID: 5

Test Case Description: send a delete request for user to Backend via Postman

Test Scenario: Check Delete User Functionality

Postman:

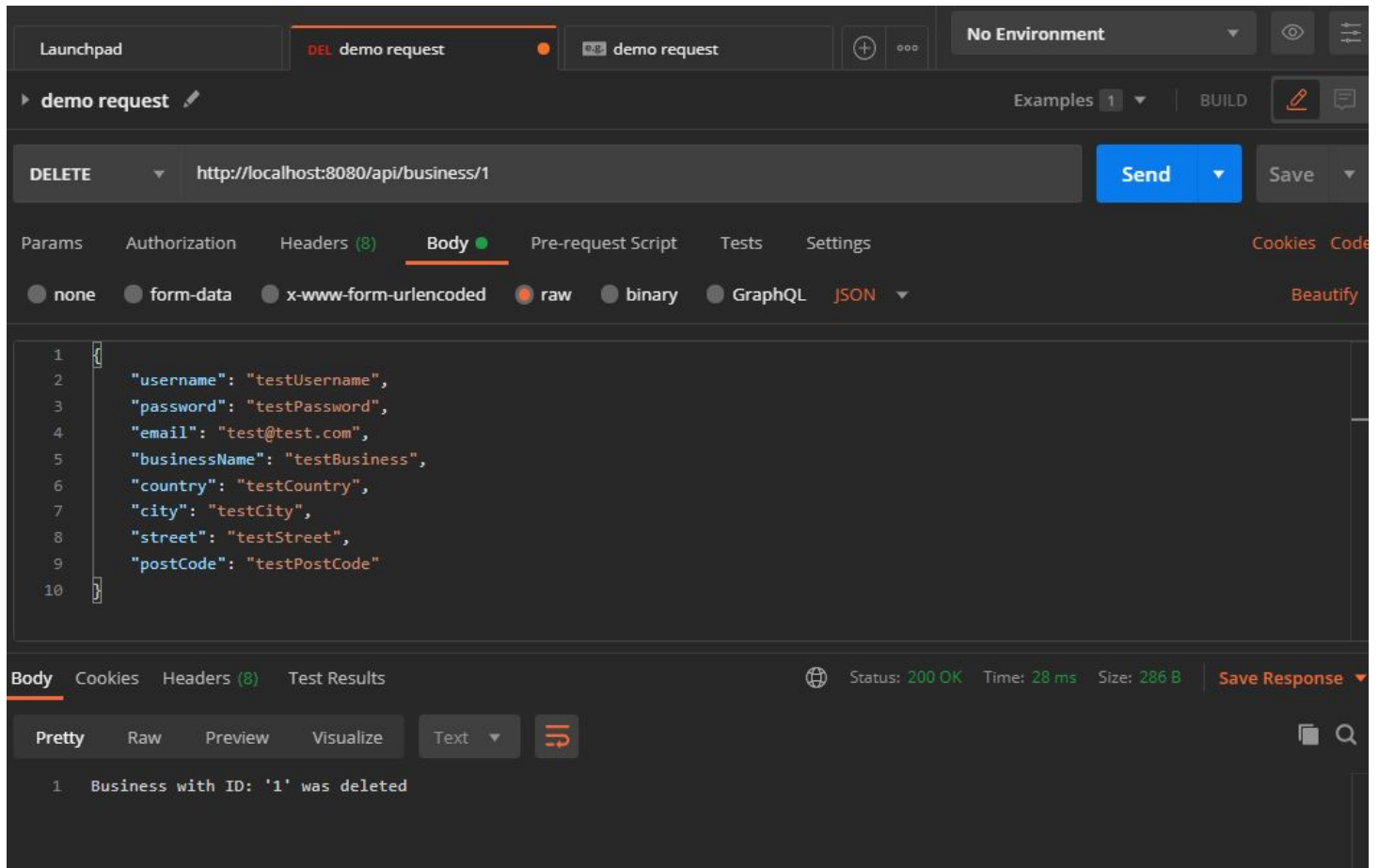


Test Case ID: 6

Test Case Description: send a delete request for business to Backend via Postman

Test Scenario: Check Delete Business Functionality

Postman:

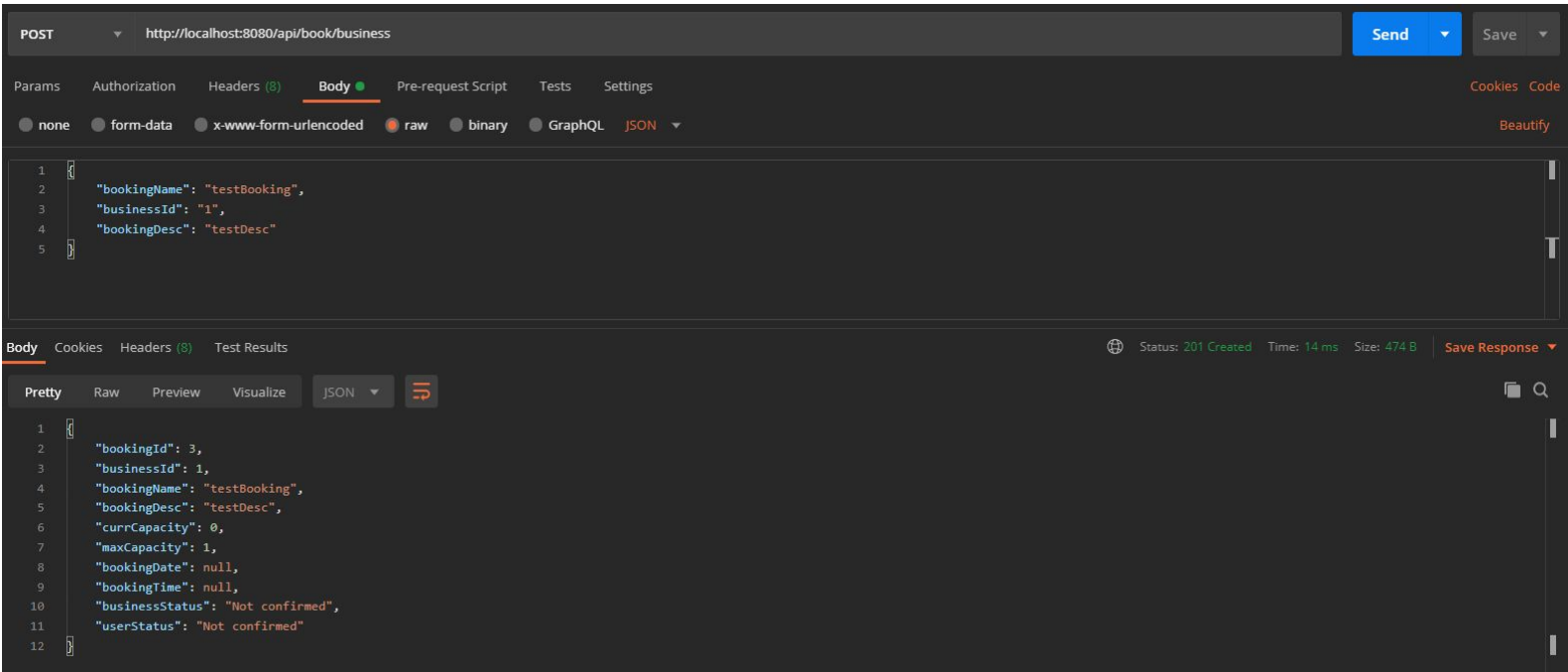


Test Case ID: 7

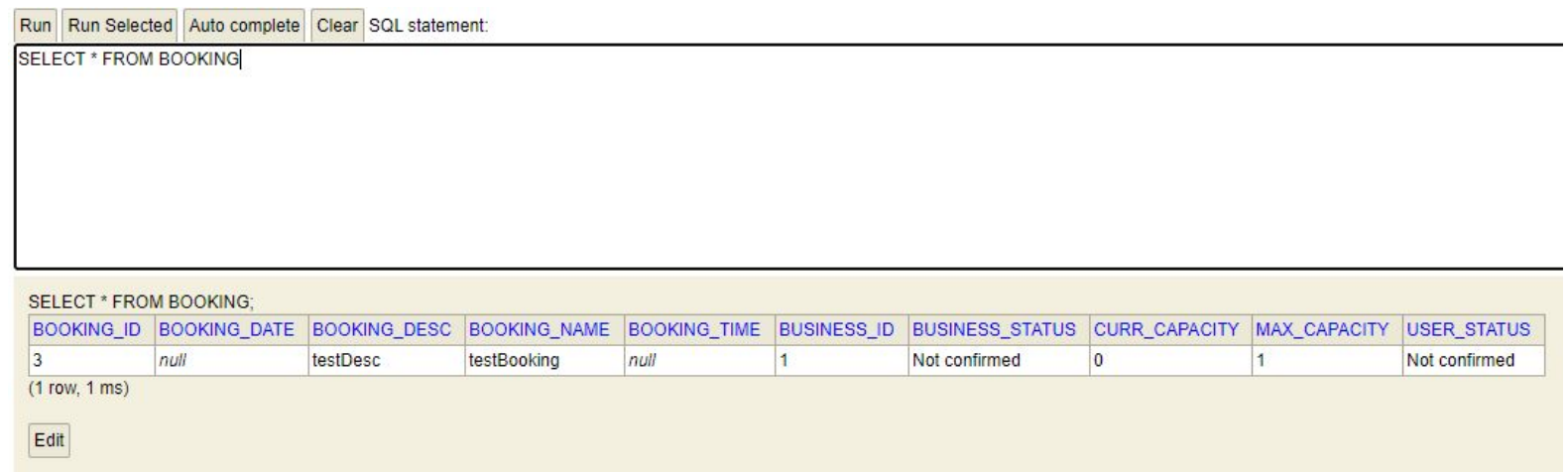
Test Case Description: send a post request for business booking to Backend via Postman

Test Scenario: Check Create Business Booking Functionality

Postman:



Database:

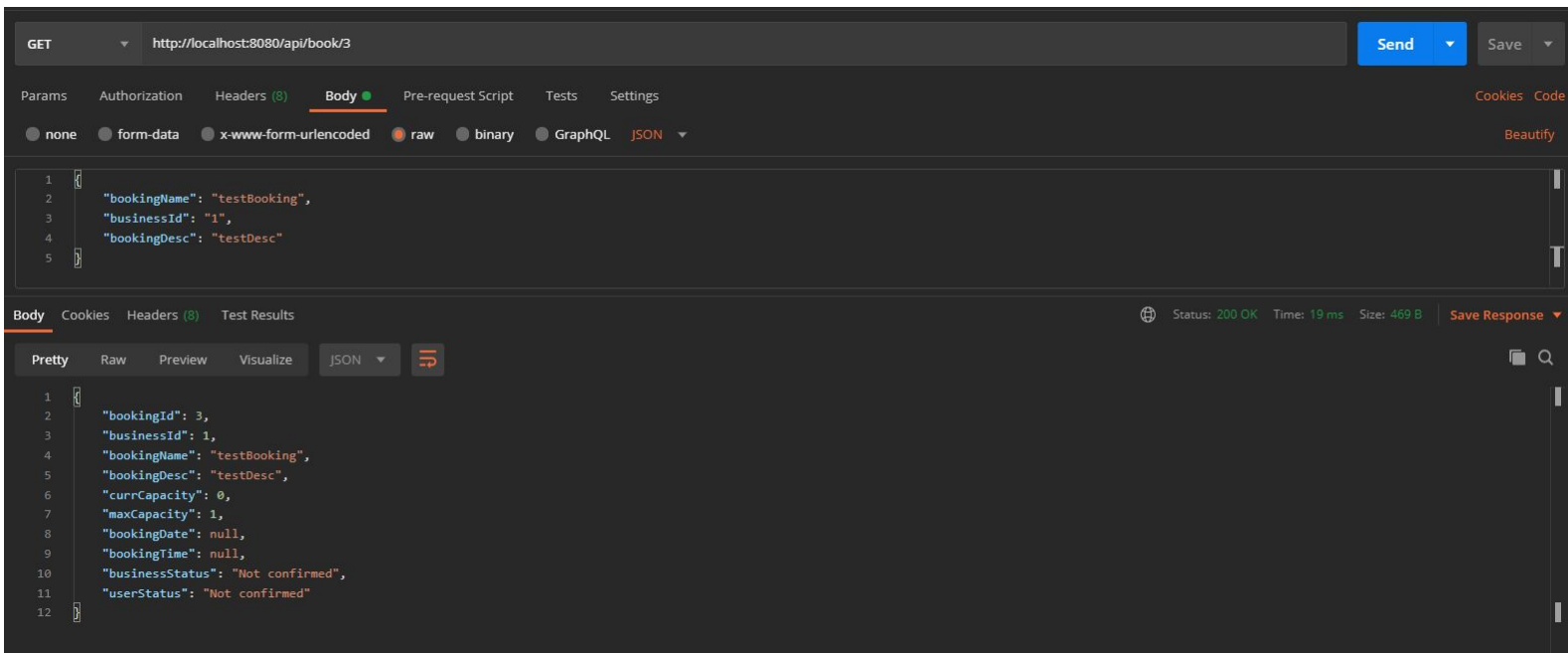


Test Case ID: 8

Test Case Description: send a get request for booking via booking ID to Backend via Postman

Test Scenario: Check Get Booking via Booking ID Functionality

Postman:

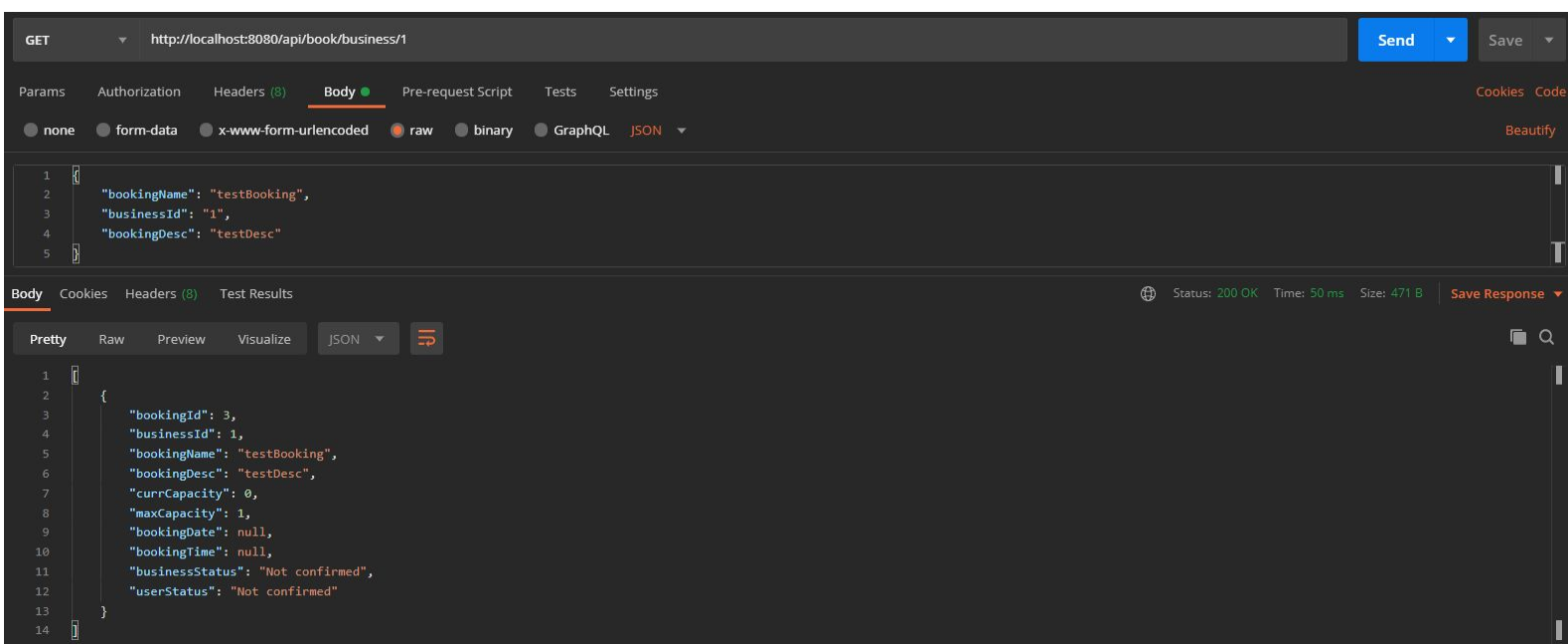


Test Case ID: 9

Test Case Description: send a get request for booking via business ID to Backend via Postman

Test Scenario: Check Get Booking via business ID Functionality

Postman:

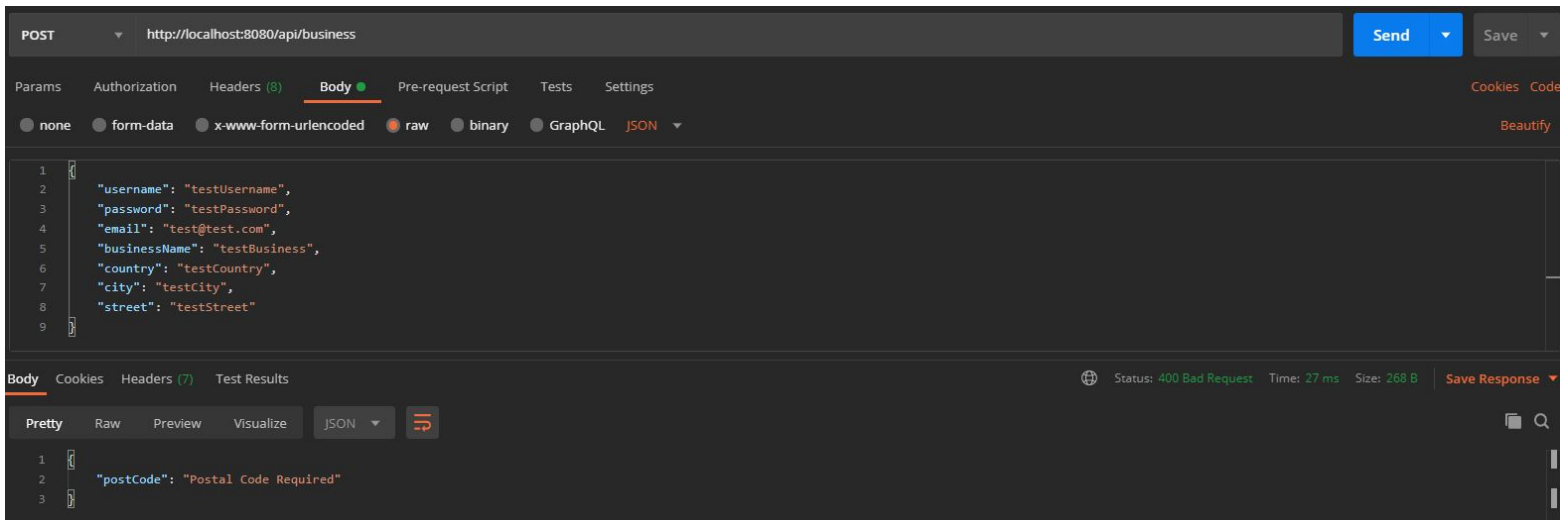


Test Case ID: 10

Test Case Description: Submit a post request with missing postCode for business to Backend via Postman

Test Scenario: Check Create Business with missing postCode Functionality

Postman:

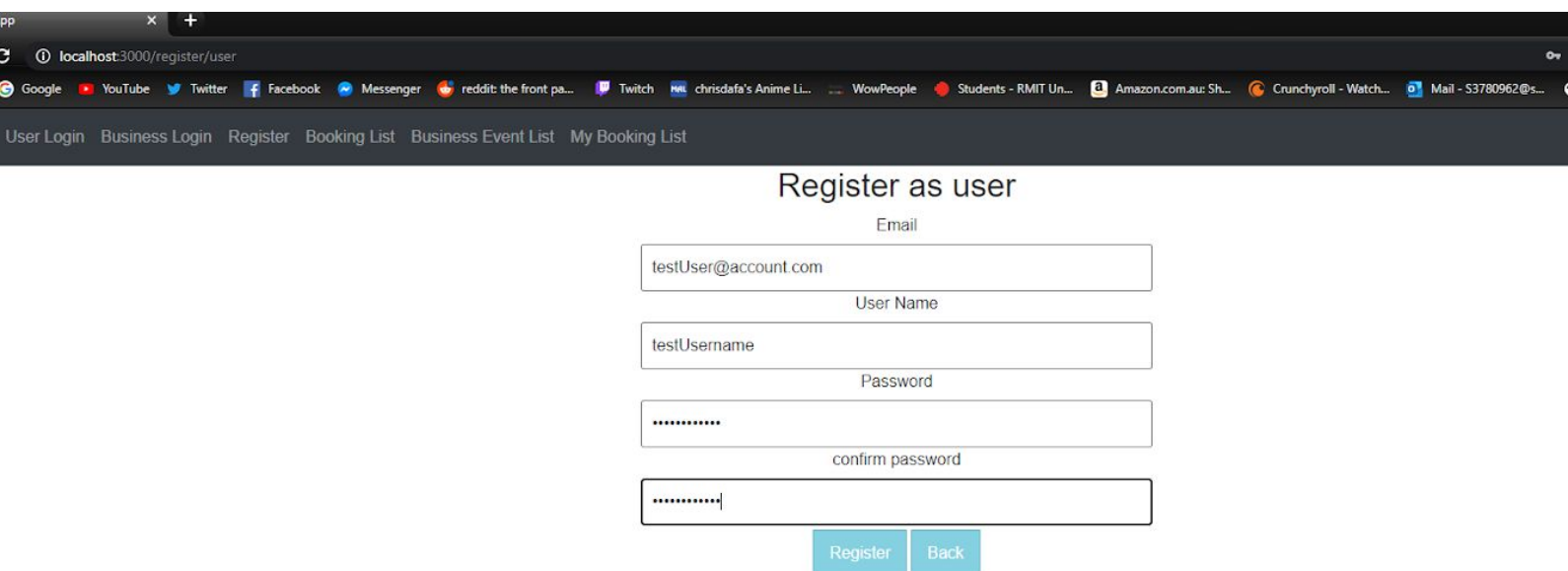
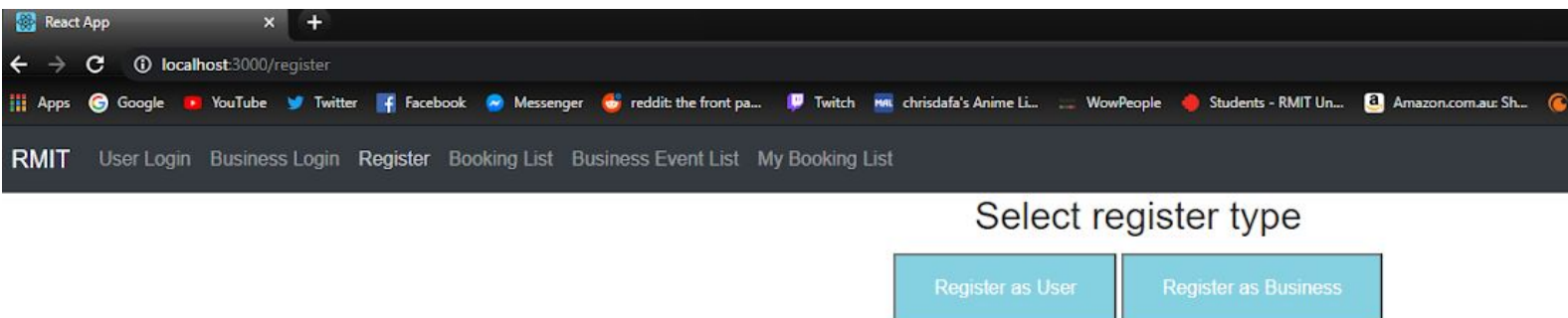
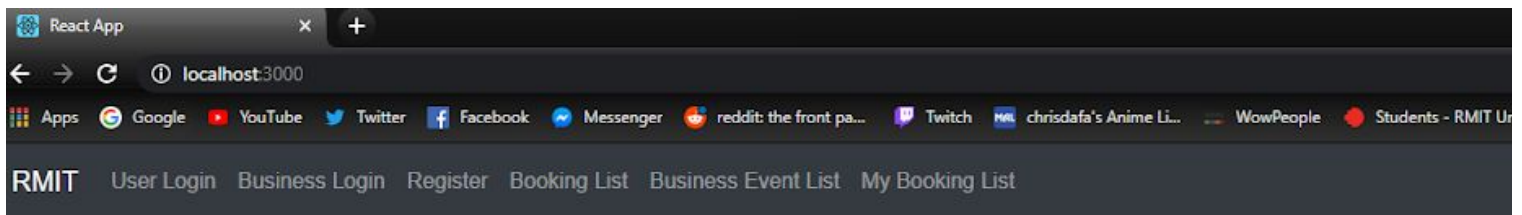


Milestone 3 Test Cases (11-30)

Test Case ID: 11

Test Case Description: Register as a user with valid user details

Test Scenario: Check Registering as a User with valid credentials



Email

testUser@account.com

User Name

testUsername

Password

confirm password

Register Back

Run Run Selected Auto complete Clear SQL statement:

SELECT * FROM CUSTOMER|

SELECT * FROM CUSTOMER;

CUSTOMER_ID	CREATED	EMAIL	LAST_MODIFIED	PASSWORD	USERNAME
1	2020-10-13 20:43:36.745	testUser@account.com	null	testPassword	testUsername

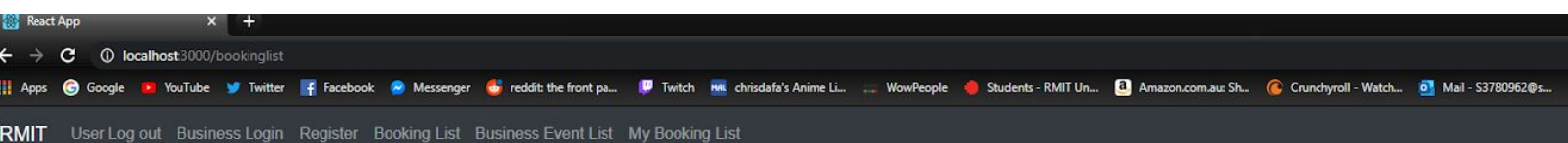
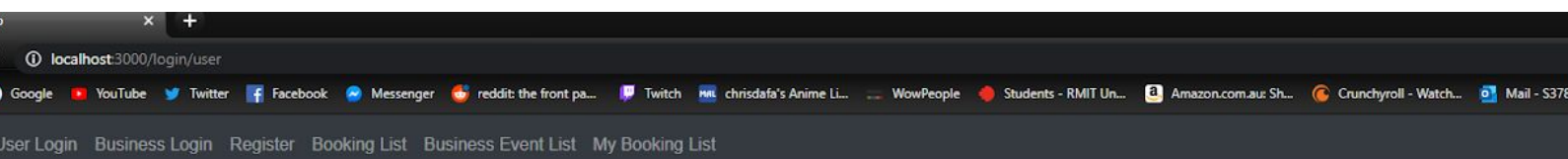
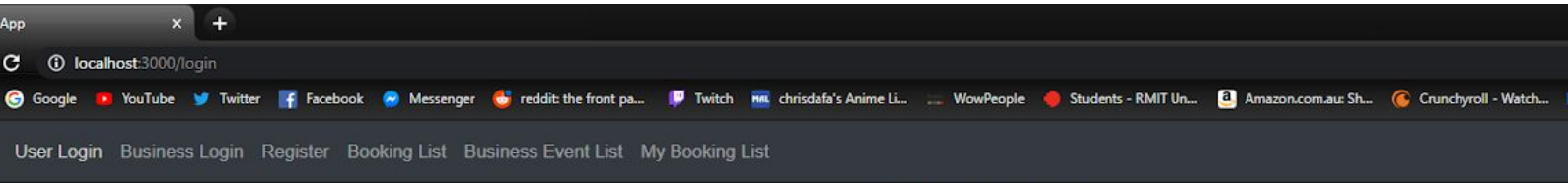
(1 row, 6 ms)

Edit

Test Case ID: 12

Test Case Description: Login as a user, with an existing credentials within database

Test Scenario: Check Logging in as a User



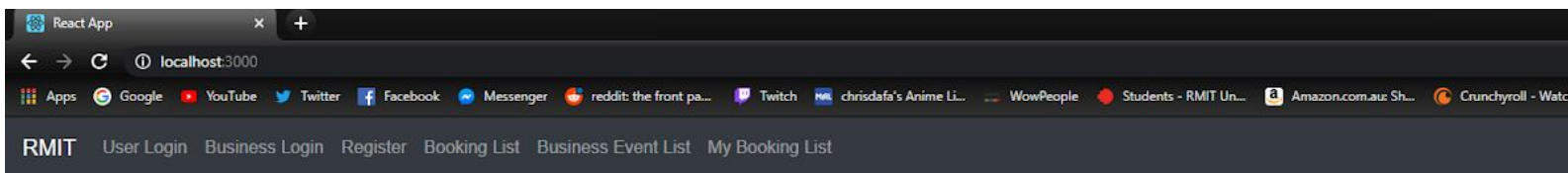
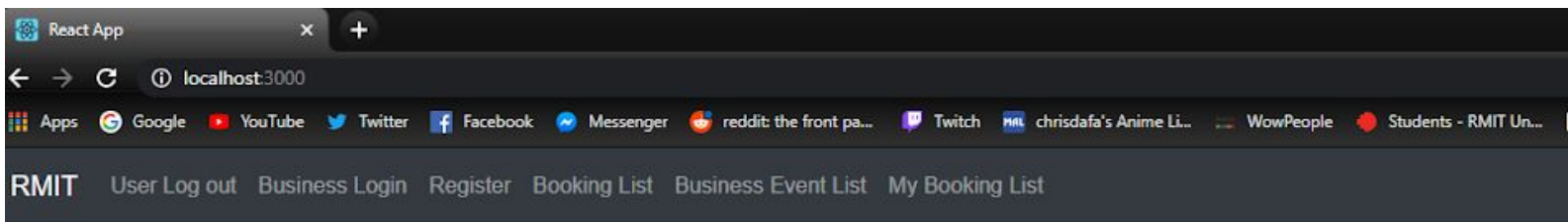
Business list

Welcome testUsername

Test Case ID: 13

Test Case Description: Logging Out as a user, given a user is logged in

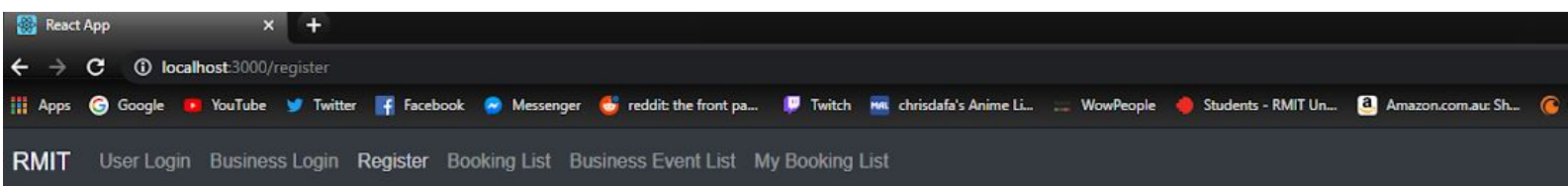
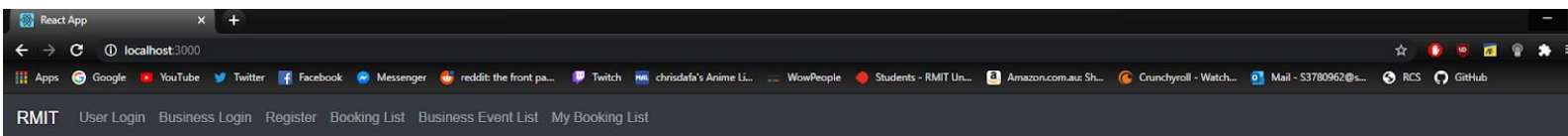
Test Scenario: Check Logging Out as a User



Test Case ID: 14

Test Case Description: Register as a business with valid business credentials

Test Scenario: Check Register as a Business



Select register type

Register as User

Register as Business

Email

test@test.com

business Name

testBusiness

business address

testBusinessAddress

street

testStreet

city

testCity

country

testCountry

postcode

testPostCode

Password

.....

confirm password

.....

Register Back

Run Run Selected Auto complete Clear SQL statement:

SELECT * FROM BUSINESS

SELECT * FROM BUSINESS;

BUSINESS_ID	BUSINESS_NAME	CREATED	EMAIL	LAST_MODIFIED	PASSWORD	USERNAME
1	testBusiness	2020-10-13 21:10:38.266	test@test.com	null	testPassword	testBusinessAddress

(1 row, 1 ms)

Edit

Run
Run Selected
Auto complete
Clear
SQL statement:

SELECT * FROM ADDRESS

SELECT * FROM ADDRESS;

CITY	COUNTRY	POST_CODE	STREET	BUSINESS_ID
testCity	testCountry	testPostCode	testStreet	1

(1 row, 2 ms)
Edit

Test Case ID: 15

Test Case Description: Login as a business with existing credentials within database

Test Scenario: Check Logging in as a Business

React App
localhost:3000/login/business

RMIT
User Login
Business Login
Register
Booking List
Business Event List
My Booking List

Login as business

Email

Password

Login
Back

welcome testBusiness

Event ID	Event Name	Current Capacity	Maximum Capacity	Event Date	Event Time	Selection
----------	------------	------------------	------------------	------------	------------	-----------

Add An Event

Event Name

Enter Event Name

Event Description

Enter Description

Event Date

please enter in a format of DD-MM-YYYY

Event Time

Please enter in a format of HH:MM:SS

Maximum Capacity

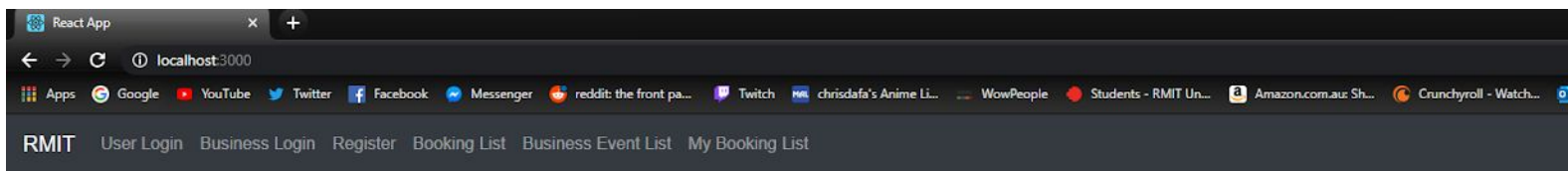
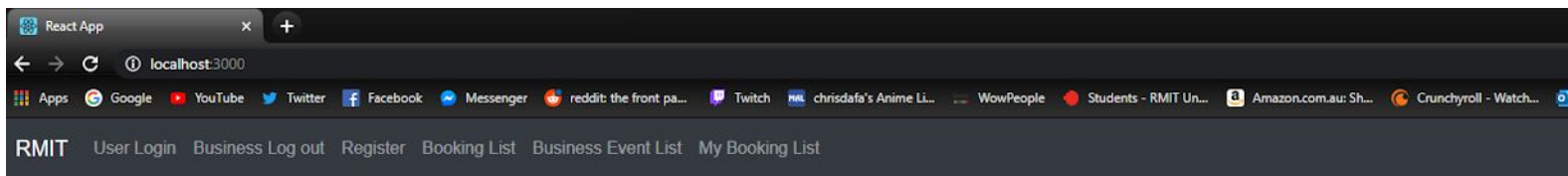
Please enter maximum capacity

Add to event

Test Case ID: 16

Test Case Description: Logging Out as a business, given a business is logged in

Test Scenario: Check Logging Out as a Business



Test Case ID: 17

Test Case Description: Add an Event as a business, given the business is logged in

Test Scenario: Check Adding an Event as a Business

React App

localhost:3000/bookingEvent

RMIT User Login Business Log out Register Booking List Business Event List My Booking List

welcome testBusiness

Event ID	Event Name	Current Capacity	Maximum Capacity	Event Date	Event Time	Selection
----------	------------	------------------	------------------	------------	------------	-----------

Add An Event

Event Name

Event Description

Event Date

Event Time

Maximum Capacity

Add to event

React App

localhost:3000/bookingEvent

RMIT User Login Business Log out Register Booking List Business Event List My Booking List

welcome testBusiness

Event ID	Event Name	Current Capacity	Maximum Capacity	Event Date	Event Time	Selection
1	testEvent	0	10	10-01-2021	04:30:20	<div>Edit</div> <div>Delete</div>

Add An Event

Event Name

Event Description

Event Date

Event Time

Maximum Capacity

Add to event

RunRun SelectedAuto completeClearSQL statement:

SELECT * FROM EVENT|

SELECT * FROM EVENT;

EVENT_ID	BUSINESS_ID	CREATED	CURR_CAPACITY	EVENT_DATE	EVENT_DESC	EVENT_NAME	EVENT_TIME	LAST_MODIFIED	MAX_CAPACITY
1	1	2020-10-13 21:27:51.171	0	2021-01-10 11:00:00	null	testEvent	1970-01-02 02:30:20	null	10

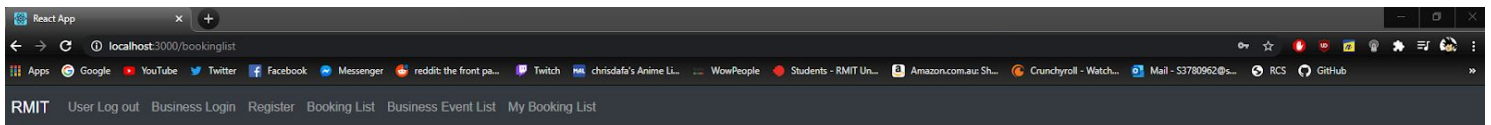
(1 row, 2 ms)

Edit

Test Case ID: 18

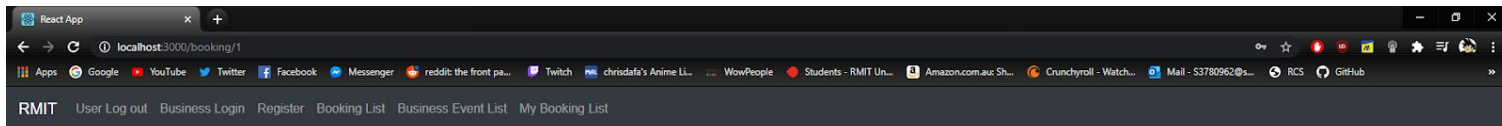
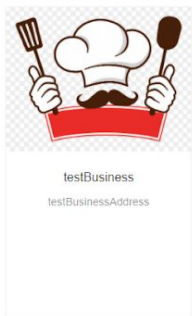
Test Case Description: Create a booking as a user, given the user is logged in, and a business exists with an event

Test Scenario: Check Create a booking as a user with 2 attendees



Business list

Welcome testUsername



welcome testUsername

Select attendance	Event Name	Curr Capacity	Max Capacity	Event Date	Event Time	Selection
1 2 3 4 5	testEvent	0	10	10-01-2021	04:30:20	Reserve

Run Run Selected Auto complete Clear SQL statement:

SELECT * FROM BOOKING|

SELECT * FROM BOOKING;

BOOKING_ID	CREATED	CUSTOMER_ID	EVENT_ID	LAST_MODIFIED
1	2020-10-13 21:38:24.983	1	1	null

(1 row, 2 ms)

Edit

Run Run Selected Auto complete Clear SQL statement:

SELECT * FROM EVENT|

SELECT * FROM EVENT;

EVENT_ID	BUSINESS_ID	CREATED	CURR_CAPACITY	EVENT_DATE	EVENT_DESC	EVENT_NAME	EVENT_TIME	LAST_MODIFIED	MAX_CAPACITY
1	1	2020-10-13 21:27:51.171	1	2021-01-10 11:00:00	null	testEvent	1970-01-02 02:30:20	2020-10-13 21:38:24.984	10

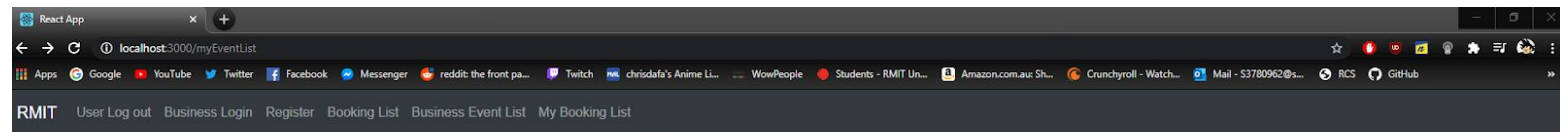
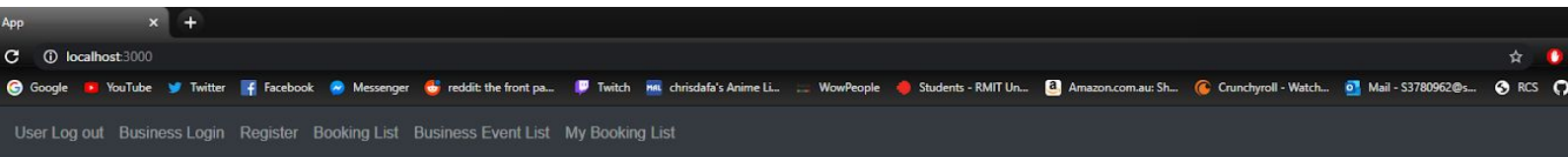
(1 row, 1 ms)

Edit

Test Case ID: 19

Test Case Description: Check a users booking list, given they have booked for an event

Test Scenario: Check Users Booking List

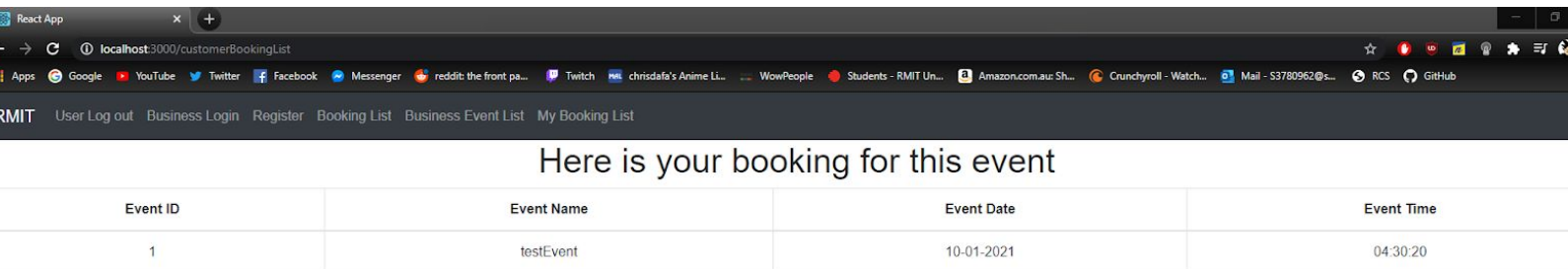
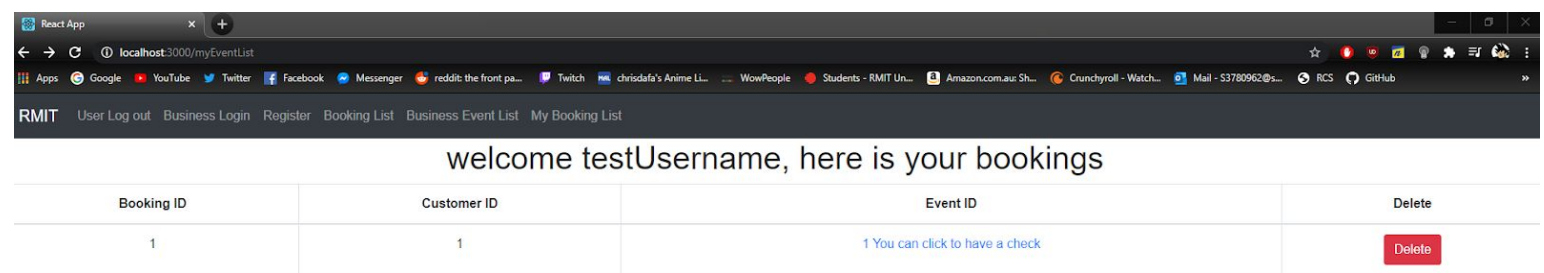
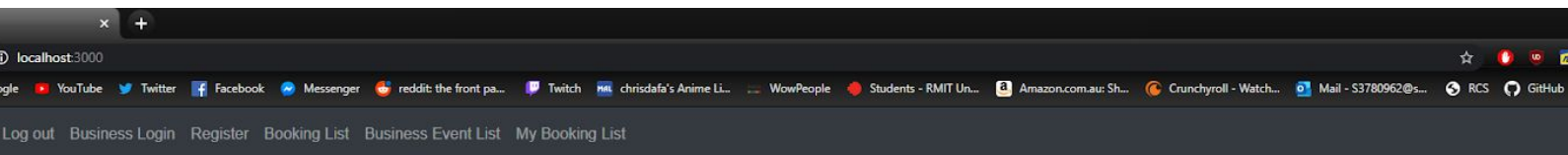


Booking ID	Customer ID	Event ID	Delete
1	1	1 You can click to have a check	Delete

Test Case ID: 20

Test Case Description: Check a users booking event details, given they have a booking

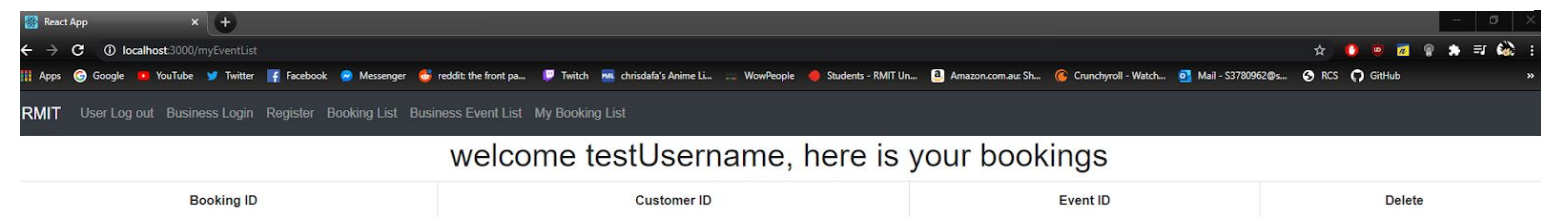
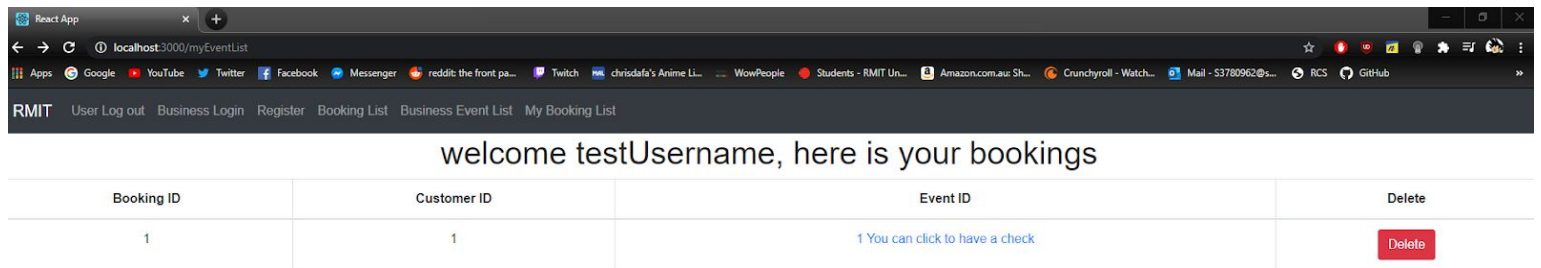
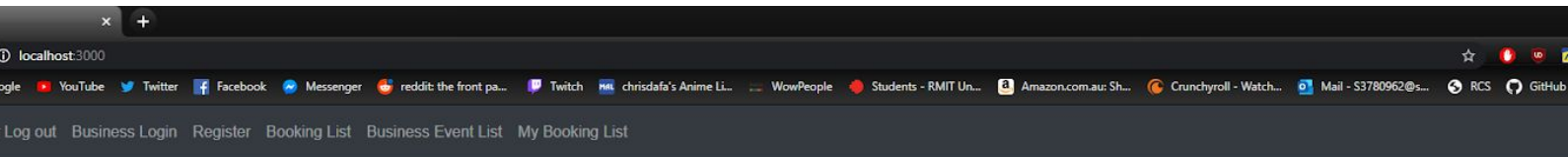
Test Scenario: Check A Users Booking Event Details



Test Case ID: 21

Test Case Description: Check cancellation of a booking, given a user has a booking

Test Scenario: Check cancelling a booking as a user



Run Run Selected Auto complete Clear SQL statement:

SELECT * FROM BOOKING

SELECT * FROM BOOKING;

BOOKING_ID CREATED CUSTOMER_ID EVENT_ID LAST_MODIFIED

(no rows, 1 ms)

Edit

Run Run Selected Auto complete Clear SQL statement:

SELECT * FROM EVENT

SELECT * FROM EVENT;

EVENT_ID	BUSINESS_ID	CREATED	CURR_CAPACITY	EVENT_DATE	EVENT_DESC	EVENT_NAME	EVENT_TIME	LAST_MODIFIED	MAX_CAPACITY
1	1	2020-10-13 21:27:51.171	0	2021-01-10 11:00:00	null	testEvent	1970-01-02 02:30:20	2020-10-13 21:53:28.628	10

(1 row, 1 ms)

Edit

Test Case ID: 22

Test Case Description: Check Admin Dashboard, given there is a business and customer in DB

Test Scenario: Check Admin Dashboard

React App Macky Gee Ft. Stuart Rowe - Af...

localhost:3000/Admindashboard

Apps Google YouTube Twitter Facebook Messenger reddit: the front pa... Twitch chrisdafa's Anime Li... WowPeople Students - RMIT Un... Amazon.com.au: Sh... Crunchyroll - Watch... Mail - S3780962@s... RCS GitHub

RMIT User Login Business Log out Register Booking List Business Event List My Booking List

Welcome to the Admin Dashboard

Business List

Business ID	Business Username	Business Email	Business Name	Created Date	Country	City	Postcode	Street	
1	testBusinessAddress	test@test.com	testBusiness	10:10:38 13-10-2020	testCountry	testCity	testPostCode	testStreet	<div>Edit</div> <div>Delete</div>

Customer List

Customer ID	Customer Username	Customer Email	Created Date	
1	testUsername	testUser@account.com	09:43:36 13-10-2020	<div>Edit</div> <div>Delete</div>

Test Case ID: 23

Test Case Description: Delete business "testBusiness", given they exist, from the Admin Dashboard

Test Scenario: Delete Business from database from Admin Dashboard

React App Macky Gee Pt-Stuart Rowe - Affo

localhost:3000/Admindashboard

Apps Google YouTube Twitter Facebook Messenger reddit the front pa... Twitch chrisdafa's Anime LI... WowPeople Students - RMIT Un... Amazon.com.au: Sh... Crunchyroll - Watch... Mail - S3780962@s... RCS GitHub

RMIT User Login Business Log out Register Booking List Business Event List My Booking List

Welcome to the Admin Dashboard

Business List

Business ID	Business Username	Business Email	Business Name	Created Date	Country	City	Postcode	Street	
1	testBusinessAddress	test@test.com	testBusiness	10.10.38 13-10-2020	testCountry	testCity	testPostCode	testStreet	<div>Edit</div> <div>Delete</div>

Customer List

Customer ID	Customer Username	Customer Email	Created Date	
1	testUsername	testUser@account.com	09.43.36 13-10-2020	<div>Edit</div> <div>Delete</div>

React App Macky Gee Pt-Stuart Rowe - Affo

localhost:3000/Admindashboard

Apps Google YouTube Twitter Facebook Messenger reddit the front pa... Twitch chrisdafa's Anime LI... WowPeople Students - RMIT Un... Amazon.com.au: Sh... Crunchyroll - Watch... Mail - S3780962@s... RCS GitHub

RMIT User Login Business Log out Register Booking List Business Event List My Booking List

Welcome to the Admin Dashboard

Business List

Business ID	Business Username	Business Email	Business Name	Created Date	Country	City	Postcode	Street
-------------	-------------------	----------------	---------------	--------------	---------	------	----------	--------

Customer List

Customer ID	Customer Username	Customer Email	Created Date	
1	testUsername	testUser@account.com	09.43.36 13-10-2020	<div>Edit</div> <div>Delete</div>

RunRun SelectedAuto completeClear

SQL statement:

SELECT * FROM BUSINESS

SELECT * FROM BUSINESS;

BUSINESS_ID	BUSINESS_NAME	CREATED	EMAIL	LAST_MODIFIED	PASSWORD	USERNAME
-------------	---------------	---------	-------	---------------	----------	----------

(no rows, 2 ms)

Edit

RunRun SelectedAuto completeClear

SQL statement:

SELECT * FROM BOOKING

SELECT * FROM BOOKING;

BOOKING_ID	CREATED	CUSTOMER_ID	EVENT_ID	LAST_MODIFIED
------------	---------	-------------	----------	---------------

(no rows, 2 ms)

Edit

Run Run Selected Auto complete Clear SQL statement:

SELECT * FROM EVENT|

SELECT * FROM EVENT;

EVENT_ID	BUSINESS_ID	CREATED	CURR_CAPACITY	EVENT_DATE	EVENT_DESC	EVENT_NAME	EVENT_TIME	LAST_MODIFIED	MAX_CAPACITY
----------	-------------	---------	---------------	------------	------------	------------	------------	---------------	--------------

(no rows, 1 ms)

Edit

Test Case ID: 24

Test Case Description: Change a customer's username in the admin dashboard from "testUsername" to "testUsername1"

Test Scenario: Edit a Customers Username from Admin Dashboard

React App

Mackay Gee Pt. Stuart Rowe - All...

+

localhost:3000/adminDashboard

Apps Google YouTube Twitter Facebook Messenger reddit: the front pa... Twitch chrisdafa's Anime Li... WowPeople Students - RMIT Un... Amazon.com.au Sh... Crunchyroll - Watch... Mail - 53780962@s... fics GitHub

RMIT User Login Business Log out Register Booking List Business Event List My Booking List

Welcome to the Admin Dashboard

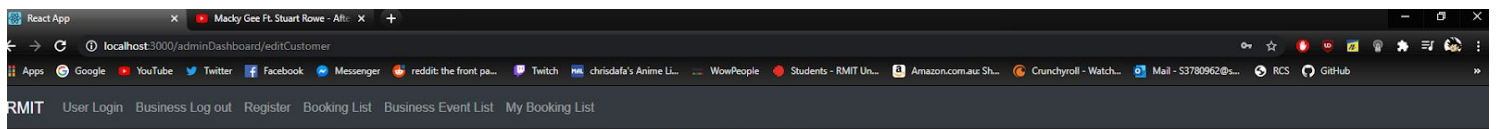
Business List

Business ID	Business Username	Business Email	Business Name	Created Date	Country	City	Postcode	Street
-------------	-------------------	----------------	---------------	--------------	---------	------	----------	--------

Customer List

Customer ID	Customer Username	Customer Email	Created Date
1	testUsername1	testUser@account.com	09:43:36 13-10-2020

EditDelete



Edit existing customer

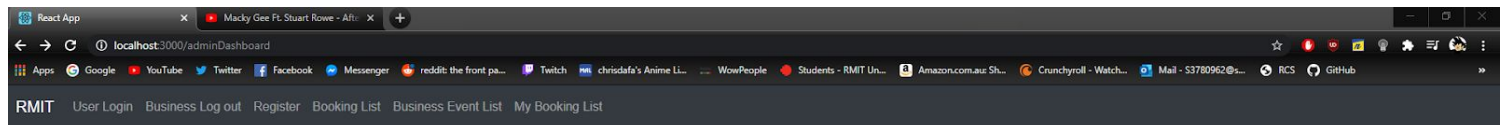
Email

User Name

Password

confirm password

Edit Back



Welcome to the Admin Dashboard

Business List

Business ID	Business Username	Business Email	Business Name	Created Date	Country	City	Postcode	Street
-------------	-------------------	----------------	---------------	--------------	---------	------	----------	--------

Customer List

Customer ID	Customer Username	Customer Email	Created Date	
1	testUsername1	testUser@account.com	09:43:36 13-10-2020	<div>Edit</div> <div>Delete</div>

Test Case ID: 25

Test Case Description: Delete User "testUsername1", given they exist, from the Admin Dashboard

Test Scenario: Delete User from database from Admin Dashboard

Welcome to the Admin Dashboard

Business List

Business ID	Business Username	Business Email	Business Name	Created Date	Country	City	Postcode	Street
-------------	-------------------	----------------	---------------	--------------	---------	------	----------	--------

Customer List

Customer ID	Customer Username	Customer Email	Created Date	
1	testUsername1	testUser@account.com	09:43:36 13-10-2020	<div>Edit</div> <div>Delete</div>

Welcome to the Admin Dashboard

Business List

Business ID	Business Username	Business Email	Business Name	Created Date	Country	City	Postcode	Street
-------------	-------------------	----------------	---------------	--------------	---------	------	----------	--------

Customer List

Customer ID	Customer Username	Customer Email	Created Date
-------------	-------------------	----------------	--------------

Run
Run Selected
Auto complete
Clear
SQL statement:

SELECT * FROM CUSTOMER|

SELECT * FROM CUSTOMER;

CUSTOMER_ID
CREATED
EMAIL
LAST_MODIFIED
PASSWORD
USERNAME

(no rows, 1 ms)

Edit

Test Case ID: 26

Test Case Description: Attempt to create a second event with a previous events time

Test Scenario: Check Restriction for Business Creating two events at the same time

The screenshot shows a web application interface. At the top, there's a navigation bar with links: RMIT, User Login, Business Log out, Register, Booking List, Business Event List, and My Booking List. Below the navigation bar, the text "welcome testBusiness" is displayed. A table lists events with columns: Event ID, Event Name, Current Capacity, Maximum Capacity, Event Date, Event Time, and Selection. The table contains one row with Event ID 1, Event Name testEvent, Current Capacity 0, Maximum Capacity 10, Event Date 10-01-2021, and Event Time 04:30:00. The Selection column has two buttons: Edit (green) and Delete (red). Below the table is a section titled "Add An Event" with a form. The form has fields for Event Name (testEvent2), Event Description (testDescription), Event Date (10-13-2020), Event Time (16:30:00), and Maximum Capacity (10). A blue button labeled "Add to event" is at the bottom of the form.

Event ID	Event Name	Current Capacity	Maximum Capacity	Event Date	Event Time	Selection
1	testEvent	0	10	10-01-2021	04:30:00	<button>Edit</button> <button>Delete</button>

Add An Event

Event Name
testEvent2

Event Description
testDescription

Event Date
10-13-2020

Event Time
16:30:00

Maximum Capacity
10

Add to event

The screenshot shows the same web application interface as the first one. The table and "Add An Event" section are identical, but the form fields in the "Add An Event" section contain placeholder text: "Enter Event Name", "Enter Description", "please enter in a format of DD-MM-YYYY", "Please enter in a format of HH:MM:SS", and "Please enter maximum capacity". The "Add to event" button is still present at the bottom of the form.

Event ID	Event Name	Current Capacity	Maximum Capacity	Event Date	Event Time	Selection
1	testEvent	0	10	10-01-2021	04:30:00	<button>Edit</button> <button>Delete</button>

Add An Event

Event Name
Enter Event Name

Event Description
Enter Description

Event Date
please enter in a format of DD-MM-YYYY

Event Time
Please enter in a format of HH:MM:SS

Maximum Capacity
Please enter maximum capacity

Add to event

Run
Run Selected
Auto complete
Clear
SQL statement:

SELECT * FROM EVENT

SELECT * FROM EVENT;

EVENT_ID	BUSINESS_ID	CREATED	CURR_CAPACITY	EVENT_DATE	EVENT_DESC	EVENT_NAME	EVENT_TIME	LAST_MODIFIED	MAX_CAPACITY
1	1	2020-10-14 16:49:50.624	0	2021-01-10 11:00:00	null	testEvent	1970-01-02 02:30:00	null	10

(1 row, 2 ms)

Edit

Test Case ID: 27

Test Case Description: Attempt to create a second booking as a user for an event the user has already booked

Test Scenario: Check Restriction for User Booking for the same event twice

React App
H2 Console
localhost:3000/myEventList

RMIT
User Log out
Business Login
Register
Booking List
Business Event List
My Booking List

welcome testUsername, here is your bookings

Booking ID	Customer ID	Event ID	Delete
1	1	1 You can click to have a check	Delete

React App H2 Console

localhost:3000/customerBookingList

Apps Google YouTube Twitter Facebook Messenger reddit: the front pa... Twitch chrisdafa's Anime Li... WowPeople Students - RMIT Un... Amazon.com.au Sh... Crunchyroll - Watch... Mail - S3780962@s... RCS GitHub

RMIT User Log out Business Login Register Booking List Business Event List My Booking List

Here is your booking for this event

Event ID	Event Name	Event Date	Event Time
1	testEvent	10-01-2021	04:30:00

React App H2 Console

localhost:3000/booking/1

Apps Google YouTube Twitter Facebook Messenger reddit: the front pa... Twitch chrisdafa's Anime Li... WowPeople Students - RMIT Un... Amazon.com.au Sh... Crunchyroll - Watch... Mail - S3780962@s... RCS GitHub

RMIT User Log out Business Login Register Booking List Business Event List My Booking List

welcome testUsername

Select attendance	Event Name	Curr Capacity	Max Capacity	Event Date	Event Time	Selection
1	testEvent	1	10	10-01-2021	04:30:00	<button>Reserve</button>

Run Run Selected Auto complete Clear SQL statement:

SELECT * FROM BOOKING

SELECT * FROM BOOKING;

BOOKING_ID	CREATED	CUSTOMER_ID	EVENT_ID	LAST_MODIFIED
1	2020-10-14 17:00:16.583	1	1	null

(1 row, 2 ms)

Edit

Run
Run Selected
Auto complete
Clear
SQL statement:

SELECT * FROM EVENT

SELECT * FROM EVENT;

EVENT_ID	BUSINESS_ID	CREATED	CURR_CAPACITY	EVENT_DATE	EVENT_DESC	EVENT_NAME	EVENT_TIME	LAST_MODIFIED	MAX_CAPACITY
1	1	2020-10-14 16:49:50.624	1	2021-01-10 11:00:00	null	testEvent	1970-01-02 02:30:00	2020-10-14 17:00:16.584	10

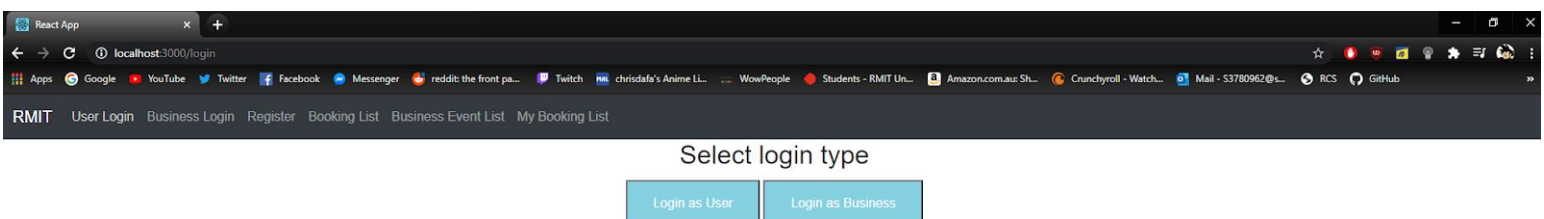
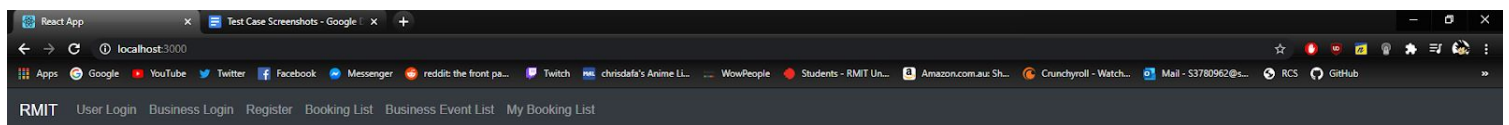
(1 row, 2 ms)

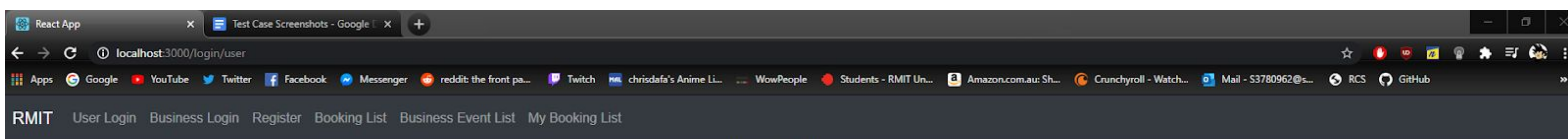
Edit

Test Case ID: 28

Test Case Description: Attempt to login with an invalid email address "NotAnEmail"

Test Scenario: Check Login with invalid email address

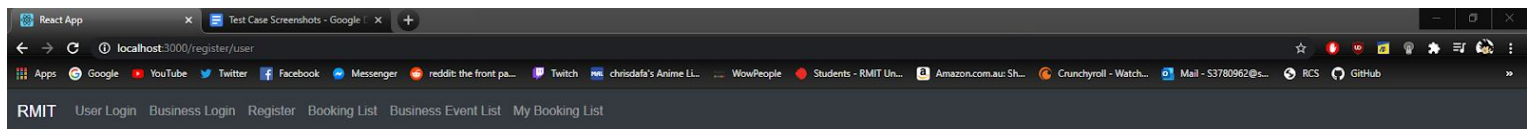
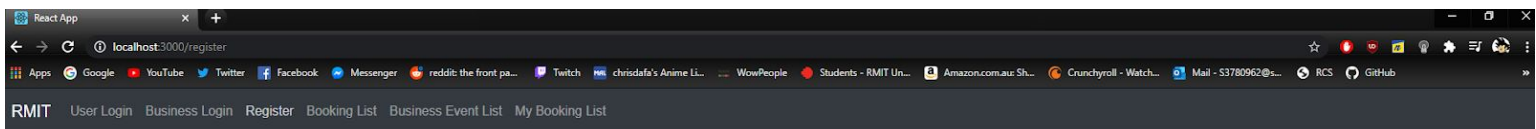




Test Case ID: 29

Test Case Description: Attempt to register with missing email address

Test Scenario: Check Register with missing field



Test Case ID: 30

Test Case Description: Attempt to register with short password "123"

Test Scenario: Check Register with short password

