Raffle Ticketing Service

Problem statement: Design raffle ticketing service
Github repo: https://github.com/hitesh2023/raffle-ticketing-service
Postman collection: https://www.getpostman.com/collections/21f9d43a1dc8212cfb90

Features implemented

- User can signup / login to the application, two types of user roles are implemented
  - **Admin:**
    - They can create an event
    - They are not allowed to participate in an event
    - They can see for a specific event which all users are registered
    - They can also check the winner of any specific event
    - They can also see all the winners of all the events that got completed in last 7 days
  - **Normal User:**
    - They can see all the ongoing / running events
    - They can also see all the available non-booked raffle tickets under an event
    - They can also register for any ongoing event with any non-booked raffle ticket number
- Normal User can register himself in any of the ongoing events with any non-booked raffle ticket number
- Once a user registers himself under an event, he is not allowed to register again under the same event, but he can register in other different events simultaneously
- Each event's winner is evaluated by a background JOB that uses some randomized algorithm to select a unique number out of the ticket numbers available, this will be triggered when the current time is in between event's start time & event end time
- Admins can also see for a specific event who was the winner
- Admins can also see all the winners of all the events that completed in last 7 days
- Normal user will get unauthenticated error if he wants to participate in an event but he is not registered in our application
- Normal uses will get unauthorized error if he tries to access the Admin functionality
- Once a user has registered himself to an event with a ticket number, this ticket will be marked as booked & hence no other user can use the same ticket

API Documentation

Registration

User's can call *signup* API to register themself in our application, for Admin's signup they can provide `"admin": true`

```
curl --location --request POST 'http://localhost:3000/signup' \
--header 'Content-Type: application/json' \
--data-raw '{
    "user": {
        "email": "admin.sarangal@gmail.com",
        "password": "password",
        "admin": true
    }
}'
```

Create New Event (Only Admins have access)

Admins can create an event by providing an event name with the start & end time, they can provide the reward item also along with the lucky ticket numbers
Validations in place:
- *event_name* should be unique
- *event_start_time* should not be greater than *event_end_time* & *event_end_time* should not be in the past
- *ticket_numbers* should be unique

```
curl --location --request POST
'http://localhost:3000/create_event?user_id=6' \
--header 'Content-Type: application/json' \
--data-raw '{
    "event_name": "Amazon Sale - 2!",
    "event_start_time": "2021-04-12 11:34:17",
    "event_end_time": "2021-05-13 11:34:17",
    "reward_item": "Washing Machine",
    "raffle_tickets_attributes": [
        {
            "ticket_number": 52
        },
        {
            "ticket_number": 52324
        }
    ]
}'
```

List all the on-going / running events (All user roles can access this)
Users can see all the running events along with their reward items, so they can participate
accordingly

```
curl --location --request GET 'http://localhost:3000/ongoing_events'
Response:
{
    "status": 200,
    "events": [
        {
            "id": 5,
            "event_name": "Big Billion Day",
            "event_start_time": "2021-04-12T11:34:17.000Z",
            "event_end_time": "2021-05-13T11:34:17.000Z",
            "reward_item": "Iphone 12"
        },
        {
            "id": 4,
            "event_name": "Amazon Sale - 4!",
            "event_start_time": "2021-04-12T11:34:17.000Z",
            "event_end_time": "2021-05-13T11:34:17.000Z",
            "reward_item": "Washing Machine"
        }
    ]
}
```

List all the non-booked available tickets under an event (All user roles can access this)
User can see all the non-booked available raffle tickets under an event, so he can participate
using any of the available ticket numbers

```
curl --location --request GET 'http://localhost:3000/events/5/get_tickets'
```

Response:
```
{
    "status": 200,
    "tickets": [
        {
            "ticket_number": 52
        },
        {
            "ticket_number": 52324
        }
    ]
}
```

Register for an event (Only normal user can be registered)

Normal user can come & register himself for an event with any of the available non-booked raffle tickets, once he registers himself that specific ticket status will be marked as booked

```
curl --location --request POST
'http://localhost:3000/events/3/register?user_id=5' \
--header 'Content-Type: application/json' \
--data-raw '{
    "ticket_number": 3221
}'
```

List all the registered users under an event (Only Admins have access)
Admins can see all the registered users along with their chosen ticket number, here user_id = 6 is Admin

```
curl --location --request GET
'http://localhost:3000/events/3/get_registered_users?user_id=6'
```

Response:
```
{
    "status": 200,
    "users": [
        {
            "id": 1,
            "email": "hitesh.sarangal@gmail.com",
            "is_admin": false,
            "raffle_ticket": {
                "ticket_number": 52
            }
        },
        {
            "id": 4,
            "email": "sonu.sarangal@gmail.com",
            "is_admin": false,
            "raffle_ticket": {
                "ticket_number": 21
            }
        },
    ]
}
```

Get winner for an event (Only Admins have access)

Admins can see the winner for a specific event, if & only if the event is ended

```
curl --location --request GET
'http://localhost:3000/events/3/get_winner?user_id=6'
Response:
{
    "status": 200,
    "event": {
        "event_name": "Amazon Sale - 2!",
        "reward_item": "Washing Machine",
        "lucky_ticket_number": 21
    },
    "winner": {
        "email": "amit.sarangal@gmail.com"
    }
}
```

Get winners for all the events completed in last 7 days (Only Admins have access)

Admins can see all the winners for all the events that got completed in last 7 days

```
curl --location --request GET
'http://localhost:3000/events/get_all_winners?user_id=6' \
--data-raw ''
Response:
{
    "status": 200,
    "events": [
        {
            "event_name": "Amazon Sale - 2!",
            "reward_item": "Washing Machine",
            "lucky_ticket_number": 21,
            "winner_email": "amit.sarangal@gmail.com"
        },
        {
            "event_name": "Amazon Sale 22!",
            "reward_item": "Washing Machine",
            "lucky_ticket_number": 52,
            "winner_email": "monika.sarangal@gmail.com"
        }
    ]
}
```

ER Diagram

user_events

+id
+user_id
+event_id

events

+id
+event_name
+reward_item
+lucky_ticket_number
+event_start_time
+event_end_time

users

+id
+email
+password
+is_admin

raffle_tickets

+id
+ticket_number
+event_id
+user_id