PRATYUSH RAJ
JUMBOTAIL ASSIGNMENT

INTRODUCTION

- Expense Tracker is an application that is helpful for tracking your daily expenses in easy way.
- Using this application the user could easily store their daily expenses and they could get an idea of how much money they are owing to someone else.

WORKING METHODOLOGY

- First the user is going to create his own expense, in which he will be giving data such as his name, category, expense amount, reason behind the expense and date of the expense.
- Now user can see all his expenses, like where his money was spent and for what purposes.
- Secondly, the user can also add the due amount that he is owing to someone else.In this case the user will be giving the data such as the name of the person to whom he is owing money, the expense id, the owed amount and the date at which he has to fulfill the dues.
- Now the user can know how much money he owes to someone else and by when he should repay it.
- The user can now use this application to clear all his dues,the working algorithm behind this application is ; like for a particular expense there will be due amount(which maybe splitted into various fractions of due amount), based on the value of the due amount ,the user is going to clear the dues(maximum fraction of the due amount will have the highest priority).
- Whenever we are going to hit the API then the highest due amount which is not settled will be cleared, User can hit the same API to settle all his dues one by one.

DATABASE USED

- MySQL

DATABASE AND TABLE STRUCTURE

EXPENSE TABLE

exp_id	BIGINT
reason	VARCHAR
payment_method	VARCHAR
name	VARCHAR
date_of_expense	BIGINT
category	VARCHAR
amount	BIGINT

DUE TABLE

due_id	BIGINT
name	VARCHAR
owe_amount	VARCHAR
repayment_date	BIGINT
status	VARCHAR
expense_exp_id	BIGINT

API CONTRACT

1. addExpense
   method:POST
   Endpoint: “/expense/”

Request Body

```
{
   name
   category
   dateOfExpense
   paymentMethod
   reason
   amount

}
```

API Response(MOCK)

```
{
   "data": {
      "expId": 62,
      "name": "Kaushal",
      "reason": "For Haircut",
      "category": "Salon",
      "dateOfExpense": 1646764452,
      "paymentMethod": "Paytm",
      "amount": 100
   },
   "message": "Expense Created",
   "status": 200
}
```

2. getExpenses

   Method: GET

   EndPoint: “/expense/<name>”

   Where the name is the name of the user by which we will be getting all the expenses of that user.

API Response(Mock)

```
{
   "data": [
      {
         "expId": 55,
         "name": "Kaushal",
         "reason": "For Haircut",
         "category": "Salon",
         "dateOfExpense": 1646764452,
         "paymentMethod": "Paytm",
         "amount": 300
      },
        {
            "expId": 58,
            "name": "Kaushal",
            "reason": "For movie",
            "category": "Mall",
            "dateOfExpense": 1646764459,
            "paymentMethod": "Paytm",
            "amount": 190
        }
    ],
    "message": "Success",
    "status": 200
}
```

3. deleteExpense

   Method: DELETE

   EndPoint: “/expense/<expId>”

   By the expense ID all the expenses and it’s dues will be deleted.

API Response(Mock)

```
{
    "data": null,
    "message": "Success",
    "status": 200
}
```

4. addDue

   Method: POST

   EndPoint: “/due/”

   Response Body

```
     {
    name
    oweAmount
  repaymentDate
  status
    "expense": {
       expId
    }


}
```

API Response

```
{
    "data": {
        "name": "jatin",
        "oweAmount": 10,
        "repaymentDate": 1646734596,
        "status": 0,
        "expense": {
            "expId": 62,
            "name": null,
            "reason": null,
            "category": null,
            "dateOfExpense": 0,
            "paymentMethod": null,
            "amount": 0
        }
    },
    "message": "Success",
    "status": 200
}
```

5. getDueOfExpenses

Method: GET

EndPoint: “/due/expense/<expId>”

By this we will be getting all the dues for a particular expense.

```
  {
    "data": [
        {
            "name": "jatin",
            "oweAmount": 500,
            "repaymentDate": 1646734596,
            "status": 0,
            "expense": {
                "expId": 70,
                "name": "Kaushal",
                "reason": "For Haircut",
                "category": "Salon",
                "dateOfExpense": 1646764452,
                "paymentMethod": "Paytm",
                "amount": 1000
            }
        },
        {
            "name": "vishal",
            "oweAmount": 200,
            "repaymentDate": 1646734596,
            "status": 0,
            "expense": {
                "expId": 70,
                "name": "Kaushal",
                "reason": "For Haircut",
                "category": "Salon",
                "dateOfExpense": 1646764452,
                "paymentMethod": "Paytm",
                "amount": 1000
            }
        },
        {
            "name": "aniket",
            "oweAmount": 300,
            "repaymentDate": 1646734596,
            "status": 0,
            "expense": {
                "expId": 70,
                "name": "Kaushal",
                "reason": "For Haircut",
                "category": "Salon",
                "dateOfExpense": 1646764452,
                "paymentMethod": "Paytm",
                "amount": 1000
            }
        }
    ],
    "message": "Success",
    "status": 200
}
```

6. clearDueByExpenseId

Method: GET

EndPoint: “/due/expense/settleByAmount/<expId>”

This method will help to settle the dues based on the value of the due amount, the due amount cannot be more than the expense amount, check is there to ensure it.

When user hit the API first time response

```
{
    "data": {
        "name": "jatin",
        "oweAmount": 500,
        "repaymentDate": 1646734596,
        "status": 1,
        "expense": {
            "expId": 70,
            "name": "Kaushal",
            "reason": "For Haircut",
            "category": "Salon",
            "dateOfExpense": 1646764452,
            "paymentMethod": "Paytm",
            "amount": 1000
        }
    },
    "message": "Due Clear",
    "status": 200
}
```

When u hit the API second time response

```
{
    "data": {
        "name": "aniket",
        "oweAmount": 300,
        "repaymentDate": 1646734596,
        "status": 1,
        "expense": {
            "expId": 70,
            "name": "Kaushal",
            "reason": "For Haircut",
            "category": "Salon",
            "dateOfExpense": 1646764452,
            "paymentMethod": "Paytm",
            "amount": 1000
        }
    },
    "message": "Due Clear",
    "status": 200
}
```

When u hit the API third time response

```
{
    "data": {
        "name": "vishal",
        "oweAmount": 200,
        "repaymentDate": 1646734596,
        "status": 1,
    "expense": {
      "expId": 70,
      "name": "Kaushal",
      "reason": "For Haircut",
      "category": "Salon",
      "dateOfExpense": 1646764452,
      "paymentMethod": "Paytm",
      "amount": 1000
    }
  },
  "message": "Due Clear",
  "status": 200
}
```

When u hit the API fourth time response, when all dues will be clear, we will be getting this as response.

```
{
  "data": null,
  "message": "All Dues are clear",
  "status": 200
}
```