



# MY ROLE, MY TEAM, MY CITI

## GROUP 1:

HRITIK S  
HARSHAVARDHINI G  
KOUSHALYA S  
NITHYASHREE N  
NIRUPAMA A R



## **GFT (GLOBAL FUNCTIONS TECHNOLOGY)**

- Global Functions Technology (GFT) delivers innovative technology solutions and key global platforms for Citi's Global Functions division.
- These solutions enable Citi to comply with regulatory mandates and empower businesses to achieve their goals, both current and long-term.
- These products and initiatives adhere to Citi's strategic architecture and support the end-to-end integration of multiple global functions within the bank including Finance, Risk assessing and handling and Controls.



## APPLICATION DEVELOPER TEAM - Harshavardhini G

- Responsibility - Part of Technology Development Team and Contribute to Applications Enhancements.
- Function - Work for application called Model Output Automation a.k.a Program 269 (Part of Citi Risk)
- Client - Modelling Users (or) Modelling Team
- Tools and Technologies -
  - Frontend - Ext JS
  - Backend - , Java, Spring, Hibernate, Sql
  - Along with Java Spark and Big data.



## DATA TEAM - NITHYASHREE N

- Our team is currently working on the CITI Insight application.
- The basic function and responsibility of our team is to build reports based on the data collected from various sources and deliver these reports to employees or leaders within the organisation.
- Internal Clients
- Tools and Technologies used: PL/SQL, Java Spark, Java, Shell Scripting, Python



## GRITT - HRITIK SHARMA

- Our team is currently working on GRITT 2.0 tool.
- GRITT stands for global reporting inventory tracking tool.
- My Role : Developer
- Tools and Technologies:-

EBX Dev, Java , Oracle PL/SQL, Microstrategy

# NIRUPAMA A.R



- My team is working on the **CITI INSIGHT** platform.
- It is a consolidated platform linking common reporting, analytics and systems across different management levels like Finance, Risk, Compliance and HR and also score cards for each employee. It ensures to keep up with the Citi standards and policies.
- Technology Stack:
  - **Frontend** - AngularJS, JQuery, Vanilla JS
  - **Backend** - Core JAVA, Spring-MVC, Spring-Batch, Spring AOP, Apache Lucene Library
  - **Integration Tools** - Microstrategy, Tableau
- My role is **Developer**.

## My Citi – GFT- ICRM – GRCT Compliance Risk – S.Koushalya



Our Objective	Milestone
Ensuring appropriate controls in place to avoid Compliance breach at all levels.	Six High street banks <b>Except Citi</b> committed Breach of CMA rules By failing <u>to</u> accurately Inform customers about bank services .  22 July 2022

### Major Initiatives and Projects

- Regulation Inventory Management
- Developed Centralized Platform for Compliance Risk monitoring and assessment

### Responsibilities

- Second Line of Defense – Striving to Mitigate Operational Risk
- Regulatory change management and Impact Assessment
- Mapping of Regulations to Business Units
- Compliance Testing- Instantaneous assessment of compliance control environment.



# INVESTMENT PORTFOLIO

## GROUP 1:

HRITIK S  
HARSHAVARDHINI G  
KOUSHALYA S  
NITHYASHREE N  
NIRUPAMA A R

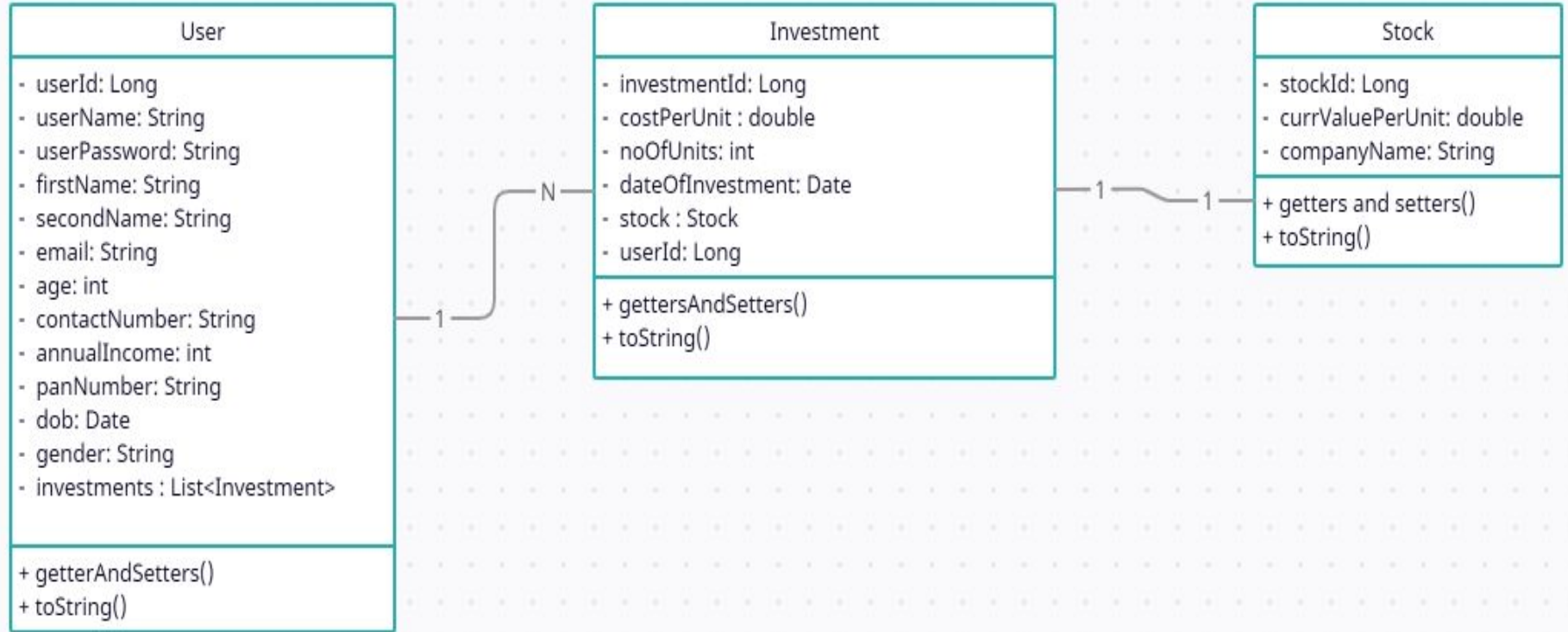




## OBJECTIVE

- The main objective is to build a REST API Spring boot application for investment portfolio.
- This application displays the user information and the various financial investments owned by the user.

# CLASS DIAGRAM



# SCHEMA DIAGRAM

```
User ∨ {
  age                integer($int32)
  annualIncome       number($double)
  contactNumber      string
  dob               string
  email              string
  firstName           string
  gender             string
  investments        ∨ [Investment ∨ {
    costPerUnit       number($double)
    dateOfInvestment  string
    investmentId       integer($int64)
    noOfUnits         integer($int32)
    stock             Stock ∨ {
      companyName     string
      currValuePerUnit number($double)
      stockId         integer($int64)
    }
    }
    userId            integer($int64)
  }]
  lastName            string
  panNumber           string
  password            string
  userId              integer($int64)
  userName            string
}
```

## H2 Database (Sample output):

jdbc:h2:mem:testdb  
+ USERS\_ACC\_ENTITY  
+ INFORMATION\_SCHEMA  
+ Users  
i H2 2.1.214 (2022-06-13)

Run Run Selected Auto complete Clear SQL statement:

SELECT \* FROM USERS\_ ACC\_ENTITY |

SELECT \* FROM USERS\_ACC\_ENTITY USERS\_ACC\_ENTITY;

USERID	AGE	ANNUAL_INCOME	GENDER	USER_NAME	USER_PASSWORD
1	23	1000000	F	HF	HF@123
2	35	2000000	M	Dad	Dad@123
3	25	800000	M	Harry	DF\$00

(3 rows, 3 ms)

localhost:8080\\Users

```
[{"userID":1,"userName":"HF","userPassword":"HF@123","age":23,"annualIncome":1000000,"gender":"F"},  
{"userID":2,"userName":"Dad","userPassword":"Dad@123","age":35,"annualIncome":2000000,"gender":"M"},  
{"userID":3,"userName":"Harry","userPassword":"DF$00","age":25,"annualIncome":800000,"gender":"M"}]
```

- Why H2 db is not recommended?
- Why MySQL?

# SWAGGER API

## user-controller User Controller



**DELETE** /deleteuser deleteUser

**POST** /new\_user addUser

**PUT** /update\_user updateUser

**GET** /user getUser

**GET** /users getUsers

## investment-controller Investment Controller



**DELETE** /deleteinvestment deleteInvestment

**GET** /investment getInvestment

**GET** /investments getInvestments

**POST** /new\_investment addInvestment

**PUT** /update\_investment updateUser

## stock-controller

Stock Controller



**DELETE** /delete\_stock deleteStock

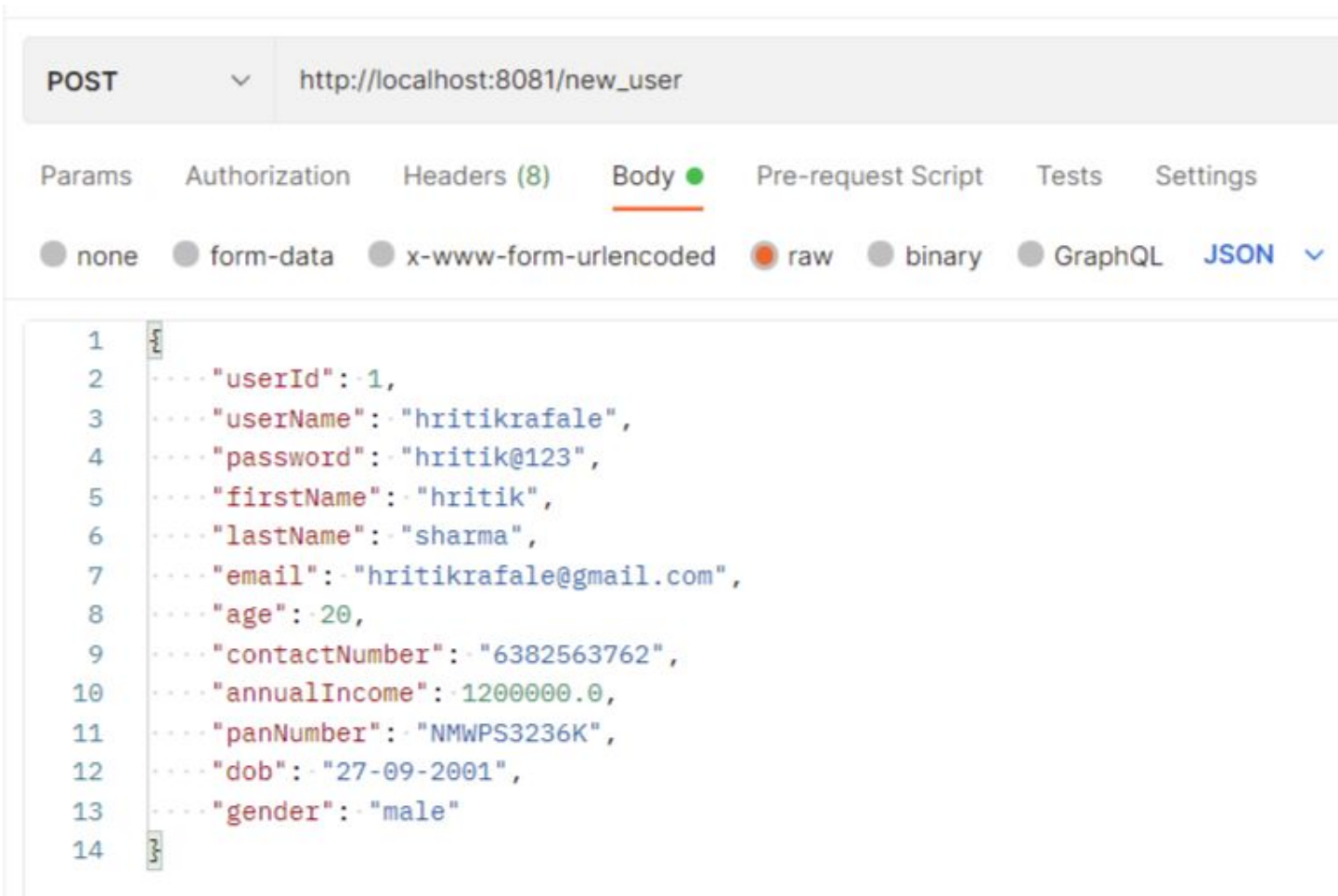
**POST** /new\_stock addStock

**GET** /stock getStock

**GET** /stocks getStocks

**PUT** /update\_stock updateStock

# CRUD OPERATIONS FOR USER ENTITY - ADD USER



The screenshot displays a REST client interface for a POST request. The URL is `http://localhost:8081/new_user`. The 'Body' tab is selected, showing a JSON payload. The payload contains user details: `userId` (1), `userName` (hritikrafafe), `password` (hritik@123), `firstName` (hritik), `lastName` (sharma), `email` (hritikrafafe@gmail.com), `age` (20), `contactNumber` (6382563762), `annualIncome` (1200000.0), `panNumber` (NMWPS3236K), `dob` (27-09-2001), and `gender` (male).

**POST** ⌵ `http://localhost:8081/new_user`

Params Authorization Headers (8) **Body** ● Pre-request Script Tests Settings

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL **JSON** ⌵

```
1 {
2   ... "userId": 1,
3   ... "userName": "hritikrafafe",
4   ... "password": "hritik@123",
5   ... "firstName": "hritik",
6   ... "lastName": "sharma",
7   ... "email": "hritikrafafe@gmail.com",
8   ... "age": 20,
9   ... "contactNumber": "6382563762",
10  ... "annualIncome": 1200000.0,
11  ... "panNumber": "NMWPS3236K",
12  ... "dob": "27-09-2001",
13  ... "gender": "male"
14 }
```

# CRUD OPERATIONS FOR USER ENTITY - GET USER

http://localhost:8081/user?id=1

GET http://localhost:8081/user?id=1

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

1 5

Body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "userId": 1,
3   "userName": "hritikrafale",
4   "password": "hritik@123",
5   "firstName": "hritik",
6   "lastName": "sharma",
7   "email": "hritikrafale@gmail.com",
8   "age": 20,
9   "contactNumber": "6382563762",
10  "annualIncome": 1200000.0,
11  "panNumber": "NMWPS3236K",
12  "dob": "27-09-2001",
13  "gender": "male",
14  "investments": [
15    {
16      "investmentId": 2,
17      "costPerUnit": 1000.0,
18      "noOfUnits": 5,
19      "dateOfInvestment": "10-08-2022",
20      "userId": 1,
21      "stock": {
22        "stockId": 3,
23        "currValuePerUnit": 1000.0,
24        "companyName": "TCS"
25      }
26    }
27  ]
28 }
```



# CRUD OPERATIONS FOR USER ENTITY - DELETE USER

http://localhost:8081/deleteuser?id=1

DELETE



http://localhost:8081/deleteuser?id=1

Params ●

Authorization

Headers (8)

Body ●

Pre-request Script

Tests

Settings

Query Params

	KEY	VALUE
<input checked="" type="checkbox"/>	id	1
	Key	Value

# CRUD OPERATIONS FOR INVESTMENT ENTITY - ADD INVESTMENT

POST

localhost:8080/investments/addInvestment

Send

Params Authorization Headers (9) **Body** Pre-request Script Tests Settings

Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL **JSON**

Beautify

```
1 {
2   "amount": 27000,
3   "noOfUnits": 7,
4   "company": "JP",
5   "type": "stock",
6   "date": "27-05-2020",
7   "currValue": 35000,
8   "userId": 1
9 }
```

**Body** Cookies Headers (5) Test Results



Status: 200 OK

Time: 28 ms

Size: 294 B

Save Response

Pretty

Raw

Preview

Visualize

JSON



```
1 {
2   "investmentId": 3,
3   "amount": 27000.0,
4   "noOfUnits": 7,
5   "company": "JP",
6   "type": "stock",
7   "date": "27-05-2020",
8   "currValue": 35000.0,
9   "userId": 1
10 }
```

# CRUD OPERATIONS FOR INVESTMENT ENTITY - GET INVESTMENT

GET

localhost:8080/investments/investments

Send

Params

Authorization

Headers (9)

Body

Pre-request Script

Tests

Settings

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

1

Body

Cookies

Headers (5)

Test Results

Status: 200 OK

Time: 15 ms

Size: 567 B

Save Response

Pretty

Raw

Preview

Visualize

JSON

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

```
{
  "investmentId": 1,
  "amount": 75000.0,
  "noOfUnits": 5,
  "company": "Fidelity",
  "type": "stock",
  "date": "15-09-2018",
  "currValue": 50000.0,
  "userId": 1
},
{
  "investmentId": 2,
  "amount": 90000.0,
  "noOfUnits": 2,
  "company": "Morgan",
  "type": "bond",
  "date": "11-02-2015",
  "currValue": 97000.0,
  "userId": 1
},
{
  "investmentId": 3,
```

# FINDING INVESTMENT BY ID

GET



localhost:8080/investments/investmentById/6

Send



Params Authorization Headers (9) Body ● Pre-request Script Tests Settings

Cookies

Body Cookies Headers (5) Test Results



Status: 200 OK

Time: 11 ms

Size: 295 B

Save Response



Pretty

Raw

Preview

Visualize

JSON



```
1 {
2   "investmentId": 6,
3   "amount": 57500.0,
4   "noOfUnits": 12,
5   "company": "GS",
6   "type": "stock",
7   "date": "12-06-2016",
8   "currValue": 65000.0,
9   "userId": 1
10 }
```

# FINDING INVESTMENT BY COMPANY

GET



localhost:8080/investments/investmentByCompany/GS

Send



Params

Authorization

Headers (7)

Body

Pre-request Script

Tests

Settings

Cookies

Body Cookies Headers (5) Test Results



Status: 200 OK

Time: 17 ms

Size: 427 B

Save Response



Pretty

Raw

Preview

Visualize

JSON



```
1  [
2    {
3      "investmentId": 4,
4      "amount": 23000.0,
5      "noOfUnits": 5,
6      "company": "GS",
7      "type": "bond",
8      "date": "30-06-2017",
9      "currValue": 25000.0,
10     "userId": 1
11   },
12   {
13     "investmentId": 6,
14     "amount": 57500.0,
15     "noOfUnits": 12,
16     "company": "GS",
17     "type": "stock",
18     "date": "12-06-2016",
19     "currValue": 65000.0,
20     "userId": 1
21   }
22 ]
```

# FINDING INVESTMENT BY TYPE

GET

localhost:8080/investments/investmentByType/stock

Send

Params

Authorization

Headers (7)

Body

Pre-request Script

Tests

Settings

Cookies

Body

Cookies

Headers (5)

Test Results

Status: 200 OK

Time: 11 ms

Size: 565 B

Save Response

Pretty

Raw

Preview

Visualize

JSON

Cookies

Capture requests

Bootcamp

Runner

Trash

# FINDING INVESTMENT BY TYPE

GET

localhost:8080/investments/investmentByType/bond

Send

Params Authorization Headers (7) Body Pre-request Script Tests Settings

Cookies

Body Cookies Headers (5) Test Results



Status: 200 OK

Time: 12 ms

Size: 696 B

Save Response

Pretty

Raw

Preview

Visualize

JSON



```
1 {
2   {
3     "investmentId": 4,
4     "amount": 23000.0,
5     "noOfUnits": 5,
6     "company": "GS",
7     "type": "bond",
8     "date": "30-06-2017",
9     "currValue": 25000.0,
10    "userId": 1
11  },
12  {
13    "investmentId": 5,
14    "amount": 55500.0,
15    "noOfUnits": 10,
16    "company": "Morgan",
17    "type": "bond",
18    "date": "30-06-2016",
19    "currValue": 60000.0,
20    "userId": 1
21  },
22  {
23    "investmentId": 7,
24    "amount": 17200.0,
25    "noOfUnits": 2,
26    "company": "JP",
27    "type": "bond",
```

# CRUD OPERATIONS FOR INVESTMENT ENTITY - UPDATE INVESTMENT

PUT

localhost:8080/investments/updateInvestment

Send

Params

Authorization

Headers (9)

Body

Pre-request Script

Tests

Settings

Cookies

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

Beautify

```
1 {  
2   "investmentId": 2,  
3   "amount": 82000,  
4   "noOfUnits": 1,  
5   "company": "GS"  
6 }
```

Body

Cookies

Headers (5)

Test Results



Status: 200 OK

Time: 28 ms

Size: 293 B

Save Response

Pretty

Raw

Preview

Visualize

JSON



```
1 {  
2   "investmentId": 2,  
3   "amount": 82000.0,  
4   "noOfUnits": 1,  
5   "company": "GS",  
6   "type": "bond",  
7   "date": "11-02-2015",  
8   "currValue": 97000.0,  
9   "userId": 1  
10 }
```



# CRUD OPERATIONS FOR INVESTMENT ENTITY - DELETE INVESTMENT

DELETE



localhost:8080/investments/deleteInvestment/2

Send



Params

Authorization

Headers (9)

Body

Pre-request Script

Tests

Settings

Cookies

☐ none

☐ form-data

☐ x-www-form-urlencoded

☒ raw

☐ binary

☐ GraphQL

JSON



Beautify

```
1 {  
2   ... "investmentId":2,  
3   ... "amount":82000,  
4   ... "noOfUnits":1,  
5   ... "company": "GS"  
6 }
```

Body

Cookies

Headers (5)

Test Results



Status: 200 OK

Time: 32 ms

Size: 183 B

Save Response



Pretty

Raw

Preview

Visualize

Text



```
1 investment removed2
```

# BONUS FEATURE - GET REAL TIME INFORMATION ABOUT STOCKS

http://localhost:8080/api/insights/find?company="TSLA"

GET http://localhost:8080/api/insights/find?company="TSLA"

Params Authorization Headers (6) **Body** Pre-request Script Tests Settings

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL **JSON** ▾

1

**Body** Cookies Headers (5) Test Results

Pretty

Raw

Preview

Visualize

JSON ▾



```
1  {
2    "symbol": "TSLA",
3    "name": "Tesla, Inc.",
4    "currency": "USD",
5    "stockExchange": "NasdaqGS",
6    "quote": {
7      "symbol": "TSLA",
8      "timeZone": "America/New_York",
9      "ask": 0.0,
10     "askSize": 12,
11     "bid": 0.0,
12     "bidSize": 12,
13     "price": 883.07,
14     "lastTradeSize": null,
15     "lastTradeDateStr": null,
16     "lastTradeTimeStr": null,
17     "lastTradeTime": "2022-08-10T20:00:04.000+00:00",
18     "open": 891.2,
19     "previousClose": 850.0,
20     "dayLow": 850.11,
21     "dayHigh": 892.5,
22     "volume": 67057
```

# BONUS FEATURE - GET MARKET SCENARIO OF STOCKS OWNED

GET

http://localhost:8080/api/users/viewAll

Params

Authorization

Headers (6)

Body

Pre-request Script

Tests

Settings

Body

Cookies

Headers (5)

Test Results

Pretty

Raw

Preview

Visualize

JSON

```
2  {
3    "userId": 3,
4    "userName": "koushi",
5    "userPassword": "k",
6    "firstName": "b",
7    "secondName": "i",
8    "email": "ju",
9    "age": 0,
10   "contactNumber": "8",
11   "annualIncome": 0,
12   "panNumber": "33",
13   "dob": null,
14   "gender": null,
15   "investment": [
16     {
17       "investmentId": 1,
18       "amount": 600.0,
19       "noOfUnits": 3,
20       "company": "BABA",
21       "type": "Stock",
22       "date": "12-09-2022",
23       "currValue": 450.0,
24       "userId": 3
25     },
26   ]
27 }
```

http://localhost:8080/api/insights/describe/3

GET

http://localhost:8080/api/insights/describe/3

Params

Authorization

Headers (6)

Body

Pre-request Script

Tests

Settings

Body

Cookies

Headers (5)

Test Results

Pretty

Raw

Preview

Visualize

JSON

```
1  {
2    "BABA": {
3      "symbol": "BABA",
4      "name": "Alibaba Group Holding Limited",
5      "currency": "USD",
6      "stockExchange": "NYSE",
7      "quote": {
8        "symbol": "BABA",
9        "timeZone": "America/New_York",
10       "ask": 0.0,
11       "askSize": 11,
12       "bid": 0.0,
13       "bidSize": 11,
14       "price": 92.43,
15       "lastTradeSize": null,
16       "lastTradeDateStr": null,
17       "lastTradeTimeStr": null,
18       "lastTradeTime": "2022-08-10T20:00:57.000+00:00",
19       "open": 91.17,
20       "previousClose": 91.19,
21       "dayLow": 88.56,
22       "dayHigh": 92.95,
23       "yearLow": 73.28,
24       "yearHigh": 192.98,
25       "priceAvg50": 104.6662,
26     }
27   }
28 }
```

# BONUS FEATURE - GET CURRENT FINANCIAL STATUS OF USER

http://localhost:8080/api/insights/checkRisk/3

Save



GET

http://localhost:8080/api/insights/checkRisk/3

Send

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings

Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL **JSON**

Beautify

```
1 {  
2   ... "userId": 3,  
3   ... "annualIncome": 45000  
4 }
```

body Cookies Headers (5) Test Results



Status: 200 OK

Time: 386 ms

Size: 337 B

Save Response

Pretty

Raw

Preview

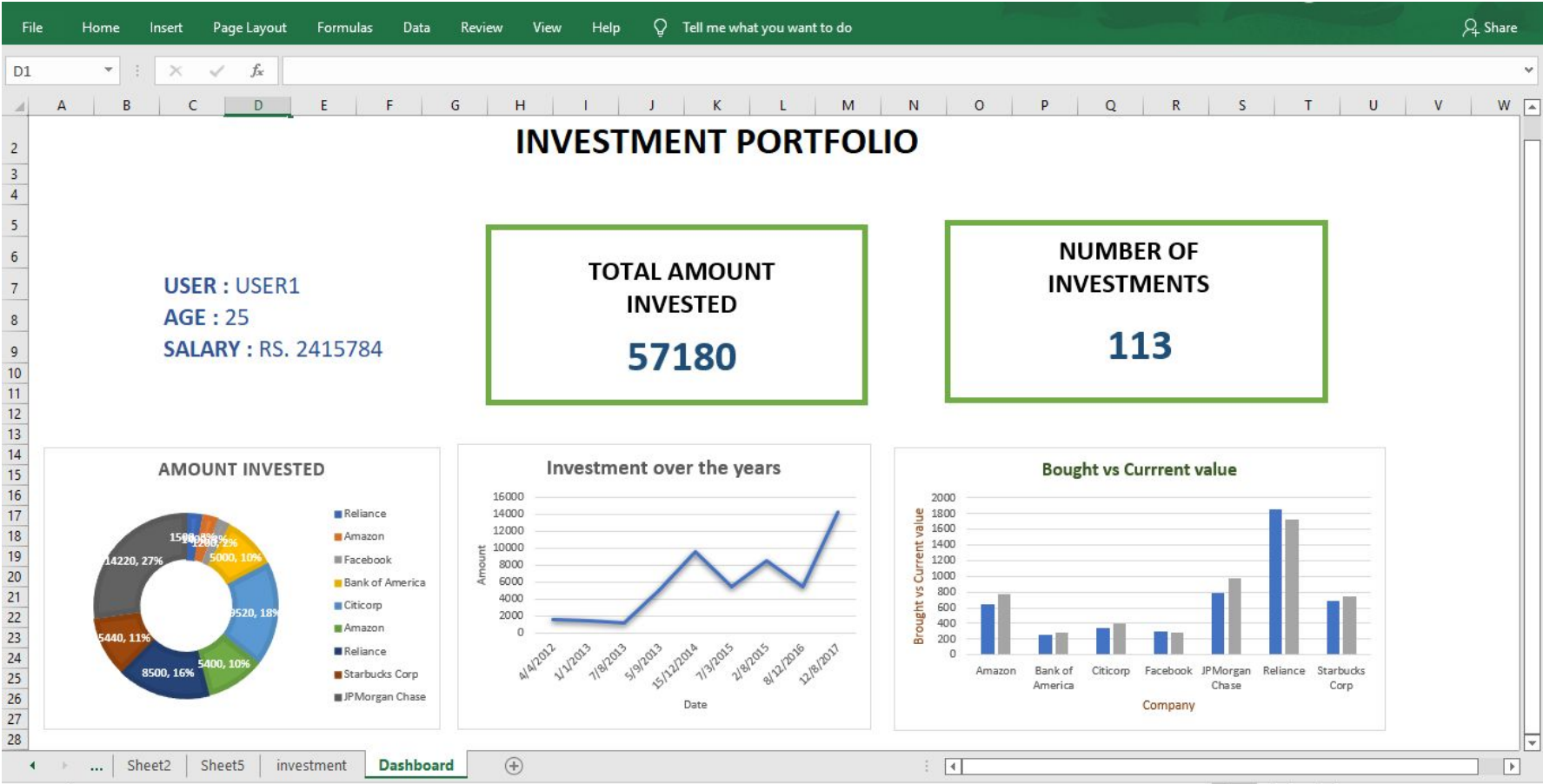
Visualize

Text



```
1 You have invested: 5400 and You can still invest : 1350.0 amount from your annual income and you can expect following returns on selling all  
the stocks in possession: -4425
```

# BONUS FEATURE - USER INVESTMENT PORTFOLIO DASHBOARD





# DOCKERIZE

```
[grads@ip-172-31-0-176 root]$ docker build -t investmentportfolio:0.0.1 .  
Sending build context to Docker daemon 213.2MB  
Step 1/4 : FROM openjdk:11-jre  
----> 362cda5d270e  
Step 2/4 : COPY target/investmentportfolio-0.0.1-SNAPSHOT.jar app.jar  
----> Using cache  
----> b148636a92b3  
Step 3/4 : EXPOSE 8080  
----> Using cache  
----> df1a37d0e895  
Step 4/4 : ENTRYPOINT ["java", "-jar", "/app.jar"]  
----> Using cache  
----> 391354253bdc  
Successfully built 391354253bdc  
Successfully tagged investmentportfolio:0.0.1
```