**CONFIDENTIAL**

**CONFIDENTIAL**

**CONFIDENTIAL**

**CONFIDENTIAL**

**SOCC ONLY**

**CONFIDENTIAL**

**CONFIDENTIAL**

**CONFIDENTIAL**

**CONFIDENTIAL**

**Web API SPECS:**

Background Information:

You are a backend developer for Steam Games, your task is to design endpoints for the Company and **KEEP IN MIND** about the different errors that might occur.

1) Getting all games within their category from database sorted alphabetically by name

ENDPOINT: **GET /games/:category**

REQUEST BODY SCHEMA: **N/A**

SUCCESS RESPONSE:

CODE: **200 OK**

CONTENT: When category is set to Action

[

    {

        "name": "Fornite",

        "price": 49.99,

        "description": "Fortnite is an online video game developed by Epic Games and released in 2017.",

        "category": "Action"

    },

    {

        "name": "Overwatch",

        "price": 79.99,

        "description": "Overwatch 2 is a free-to-play, team-based action game set in the optimistic future, where every match is the ultimate 5v5 battlefield brawl.",

        "category": "Action"

    },

    {

        "name": "Valorant",

        "price": 10.99,

        "description": "Valorant is a free-to-play first-person tactical hero shooter developed and published by Riot Games, for Microsoft Windows.",

        "category": "Action"

    }

]

ERROR RESPONSE(S):

CONDITION: INVALID CATEGORY

CODE: **422 (Unprocessable Entity)**

CONTENT:

{

    "Message": "Invalid category"

}

CONDITION: ANY ERROR

CODE: **500 (Internal Server Error)**

CONTENT:

{

    "Message": "Some error occurred"

}

2) Update price of games [Sales]

ENDPOINT: **PUT /games/price/**

REQUEST BODY SCHEMA: gameIDs, price

SUCCESS RESPONSE:

CODE: **200 OK**

CONTENT:

{

gameIDs: 1,8,9

price: 29.99,39.99,49.99

}

{

    "message": "Success"

}

ERROR RESPONSE(S):

CONDITION: If one or all game ids does not exist in game table, do not update any of the game id

CODE: **422 (Unprocessable Entity)**

CONTENT:

{

    "message": "No such game(s) exists"

}

CONDITION: ANY ERROR

CODE: **500 (Internal Server Error)**

CONTENT:

{

    "message": "Some error occurred"

}

3) Giving recommendation [Creating STEAM game bundle]

20% discount off of total price and sort by price descendingly and limit to 3 games

ENDPOINT: **GET /recommendation/:userid**

REQUEST BODY SCHEMA: N/A

SUCCESS RESPONSE:

CODE: **200 OK**

CONTENT: When userid is set to 1

{

    "Recommended": [

        {

            "game": "Overwatch",

            "category": "Action",

            "price": 79.99

        },

        {

            "game": "Fornite",

            "category": "Action",

            "price": 49.99

        },

        {

            "game": "Raid: Shadow Legends",

            "category": "Gacha",

            "price": 29.99

        }

    ],

    "Total Price": 127.98

}

ERROR RESPONSE(S):

CONDITION: No RECOMMENDATION

CODE: **500 (Internal Server Error)**

CONTENT:

{

    "MESSAGE": "No recommendation found"

}

CONDITION: ANY ERROR

CODE: **500 (Internal Server Error)**

CONTENT:

{

    "MESSAGE": "Some error occurred"

}

4) Inserting new interest – Ensure that interests are not duplicated [HARD]

ENDPOINT: **POST /interest/:userid**

REQUEST BODY SCHEMA: categories

SUCCESS RESPONSE:

CODE: **201 CREATED**

CONTENT: When userid is 2

{

categories: “Action,Sandbox”

}

{

    "message": "Success"

}

Table

Description automatically generated

ERROR RESPONSE(S):

CONDITION: Invalid userid

CODE: **500 (Internal Server Error)**

CONTENT:

{

    "message": "Invalid userid"

}

CONDITION: Invalid category

CODE: **500 (Internal Server Error)**

CONTENT:

{

    "message": "No such category(s) exists"

}

CONDITION: ANY ERROR

CODE: **500 (Internal Server Error)**

CONTENT:

{

    "message": "Some error occurred"

}

**You are required to take one screenshot for each web api test showing the test data and results of POSTMAN based on the scenarios below. Correctness will be validated with your source code. Remember to attach the screenshots files in a word document file with your name, admin number and class in the document(screenshots.doc) and attach it together with the submission zip file.**

5) Using Question 1 end point, set category as **Puzzle**. Take a screenshot of the output.

6) Using Question 2 end point, use the following body.

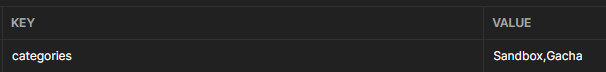
Graphical user interface, application

Description automatically generated

Take a screenshot of the output.

7) Using Question 3 end point, set userid as **2**. Take a screenshot of the output.

8) Using Question 4 end point, set userid as **2** and use the following body.



Take a screenshot of the output and SQL table.

**CONFIDENTIAL**

**CONFIDENTIAL**

**CONFIDENTIAL**

**CONFIDENTIAL**

**SOCC ONLY**

**CONFIDENTIAL**

**CONFIDENTIAL**

**CONFIDENTIAL**

**CONFIDENTIAL**