

## **Task 1: Real-Time Transaction Monitoring and Fraud**

### **Detection**

Objective:

Develop a real-time transaction monitoring API to evaluate and flag transactions based on specific fraud detection rules.

### **ENDPOINTS->**

<u>EndpointExposed</u>	<u>Type</u>	<u>ROLE ACCESS</u>	<u>Endpoint Secured</u>	<u>Rate Limiting Applied</u>
http://localhost:8080/transactions/process	POST	user,admin	Yes	Yes
http://localhost:8080/admin/flagged-transactions	GET	admin	Yes	Yes
http://localhost:8080/admin/action_on_transaction	POST	ADMIN	Yes	Yes

### **Credential->**

Username->admin

Password->admin123

Username->user

Password->user123

### **JSON\_INPUT\_DATA\_PROCESSING->**

http://localhost:8080/transactions/process

**POST** http://localhost:8080/transactions/process Send

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

Type Basic Auth

The authorization header will be automatically generated when you send the request.  
[Learn more about authorization](#)

Username

Password

Store your secrets with end-to-end encryption locally using Vault. Store in Vault

body Cookies Headers (11) Test Results Status: 200 OK Time: 71 ms Size: 3.35 KB Save Response

**POST** http://localhost:8080/transactions/process Send

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

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

```
1 [
2   {
3     "transactionId": "TXN463988445",
4     "accountNumber": "6083103750",
5     "amount": 55525,
6     "transactionTime": "2024-12-16T04:55:10.220804",
7     "ipAddress": "93.252.47.153",
8     "location": {
9       "country": "India",
10      "city": "Chennai"
11    },
12    "transactionType": "Transfer",
13    "remarks": "ATM withdrawal"
14  },
15  {
16    "transactionId": "TXN933888353",
17    "accountNumber": "4285042878",
18    "amount": 33420,
19    "transactionTime": "2024-12-16T11:06:10.220804",
20    "ipAddress": "170.75.227.35",
21    "location": {
22      "country": "India",
23      "city": "Delhi"
24    }
25  }
26 ]
```

JSON->

```
[
{
"transactionId": "TXN463988445",
"accountNumber": "6083103750",
"amount": 55525,
"transactionTime": "2024-12-16T04:55:10.220804",
"ipAddress": "93.252.47.153",
"location": {
"country": "India",
"city": "Chennai"
},
"transactionType": "Transfer",
"remarks": "ATM withdrawal"
},
{
"transactionId": "TXN933888353",
"accountNumber": "4285042878",
"amount": 33420,
"transactionTime": "2024-12-16T11:06:10.220804",
"ipAddress": "170.75.227.35",
"location": {
"country": "India",
"city": "Delhi"
}
}
}]
```

HEADER->

Note->have disbaled csrf token

The image shows a development environment with two main parts. On the left, an IDE (IntelliJ IDEA) displays a project structure for 'transaction\_fraud\_monitor' and a Java file 'SecurityConfig.java'. The code in 'SecurityConfig.java' is as follows:

```
1 package com.project.transaction_fraud_monitor.filter;
2
3
4
5 import org.springframework.context.annotation.Bean;
6
7 @Configuration
8 @EnableWebSecurity
9 public class SecurityConfig {
10
11     @Bean
12     public SecurityFilterChain securityFilterChain(HttpSecurity http) throws Exception {
13         http.csrf(csrf -> csrf.disable());
14         .authorizeHttpRequests(auth -> auth
15             .requestMatchers("/admin/**").hasRole("ADMIN")
16             .requestMatchers("/transactions/**").hasAnyRole("ADMIN", "USER").requestMatchers("/**").hasRole("ADMIN")
17         );
18         .httpBasic(Customizer.withDefaults()); // Enable HTTP Basic Authentication
19         .formLogin(Customizer.withDefaults()); // Enable form-based login
20         .sessionManagement(session -> session
21             .sessionCreationPolicy(SessionCreationPolicy.STATELESS)
22         );
23     }
24
25     return http.build();
26 }
```

On the right, the Postman client interface is shown. The URL is 'http://localhost:8080/transactions/process' and the method is 'POST'. The 'Headers' tab is selected, showing a list of headers:

Key	Value
Authorization	Basic dXNlcj1c2VyMTIz
Postman-Token	<calculated when request is sent>
Content-Type	application/json
Content-Length	<calculated when request is sent>
Host	<calculated when request is sent>
User-Agent	PostmanRuntime/7.43.0
Accept	*/*
Accept-Encoding	gzip, deflate, br
Connection	keep-alive

JSON\_FOR\_POST->

http://localhost:8080/admin/action\_on\_tansaction

Header and Basic Auth can be Same

```
[
  {
    "blockedAccounts": [
      "account1",
      "account2",
      "account3"
    ],
    "action": "add"
  }
]
```

```
1  [  
2    {  
3      "blockedAccounts": [  
4        "account1",  
5        "account2",  
6        "account3"  
7      ],  
8      "action": "add"  
9    }  
10 ]
```

## RESPONSES->

### Expected Endpoints:

#### 1. POST /transactions/process

- **Input:** Transaction JSON payload.

- **Output:**

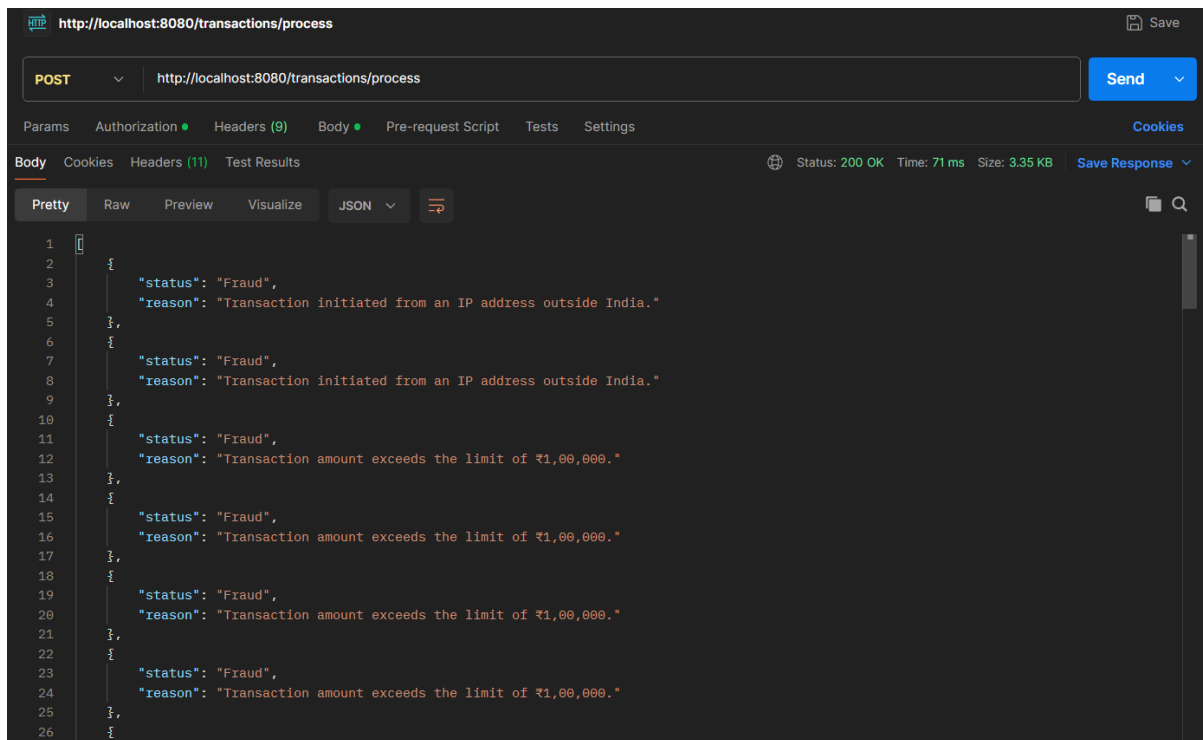
```
{  
  "status": "Fraud",  
  "reason": "Transaction amount exceeds the limit of ₹1,00,000."  
}
```

Or:

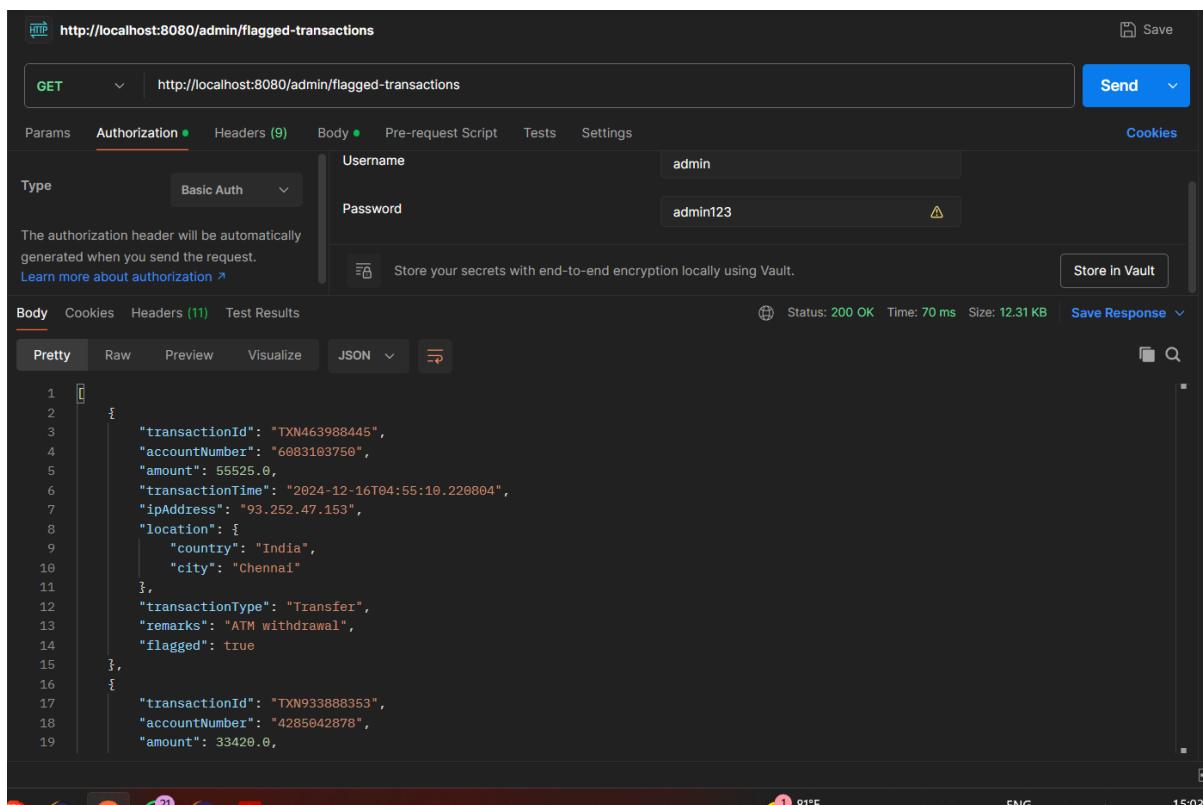
```
{  
  "status": "Success",  
  "message": "Transaction is valid."  
}
```

Note->Tailored Responses specifically to match the Response pattern mentioned in Assessment PDF shared containing the TEST question.

**<http://localhost:8080/transactions/process>**



**http://localhost:8080/admin/flagged-transactions->GET**



**http://localhost:8080/admin/ action\_on\_tansaction-->POST**

TO ADD REMOVE BLACKLIST ACCOUNT NUMBER BY ADMIN

http://localhost:8080/admin/action\_on\_tansaction

POST

Send

ParamsAuthorizationHeaders (9)BodyPre-request ScriptTestsSettingsCookies

TypeBasic Auth

The authorization header will be automatically generated when you send the request.  
[Learn more about authorization](#)

Usernameadmin

Passwordadmin123

POST

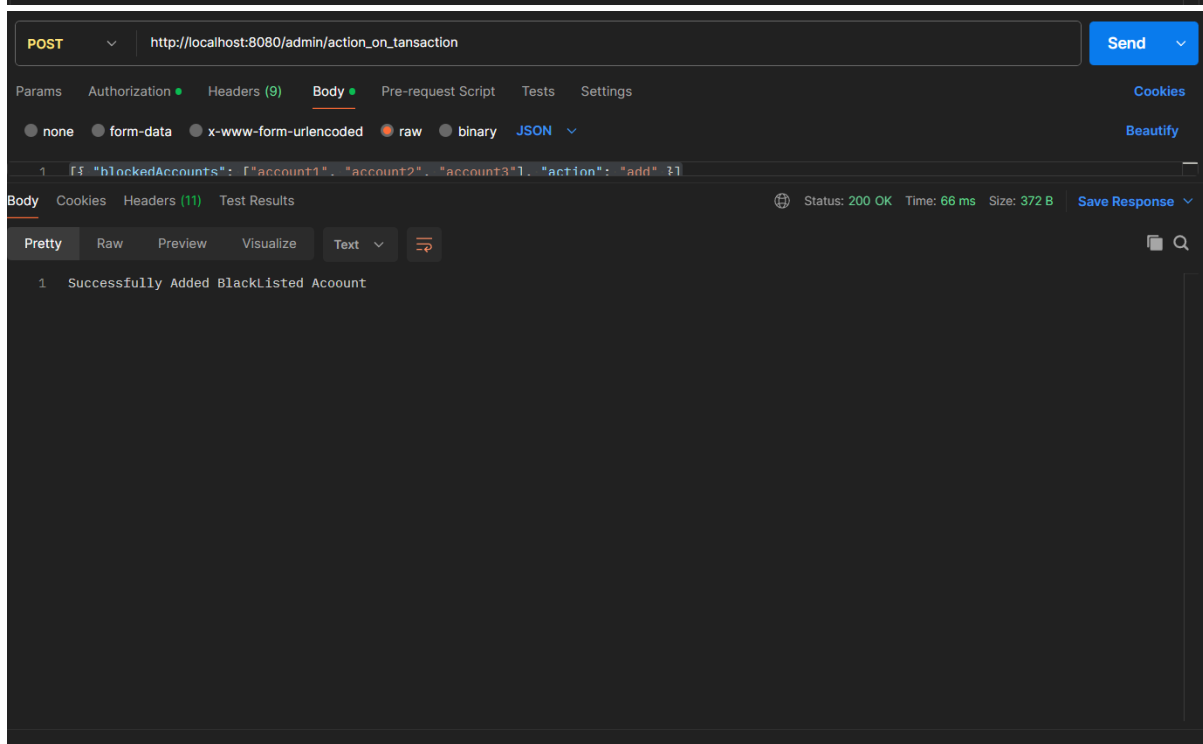
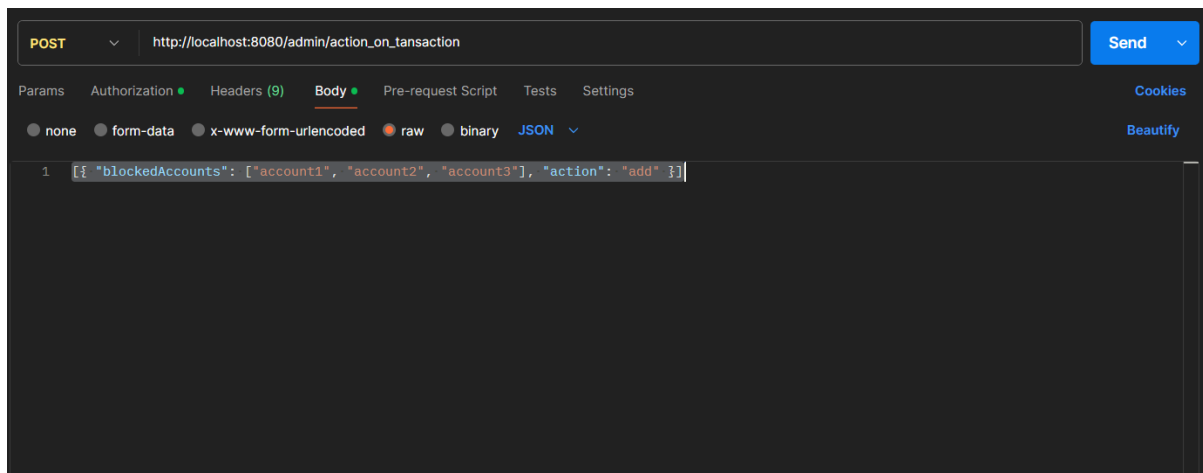
Send

ParamsAuthorizationHeaders (9)BodyPre-request ScriptTestsSettingsCookies

Headers

Hide auto-generated headers

	Key	Value	Bulk Edit
<input checked="" type="checkbox"/>	Authorization ⓘ	Basic YWRtaW46YWRtaW4xMjM=	
<input checked="" type="checkbox"/>	Postman-Token ⓘ	<calculated when request is sent>	
<input checked="" type="checkbox"/>	Content-Type ⓘ	application/json	
<input checked="" type="checkbox"/>	Content-Length ⓘ	<calculated when request is sent>	
<input checked="" type="checkbox"/>	Host ⓘ	<calculated when request is sent>	
<input checked="" type="checkbox"/>	User-Agent ⓘ	PostmanRuntime/7.43.0	
<input checked="" type="checkbox"/>	Accept ⓘ	*/*	
<input checked="" type="checkbox"/>	Accept-Encoding ⓘ	gzip, deflate, br	
<input checked="" type="checkbox"/>	Connection ⓘ	keep-alive	
	Key	Value	

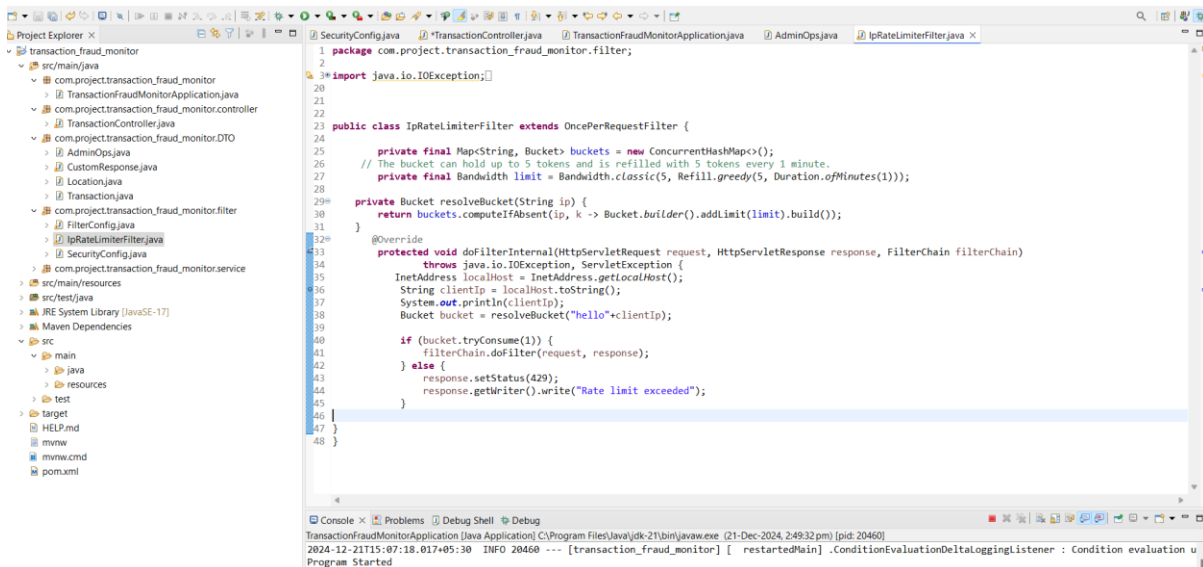
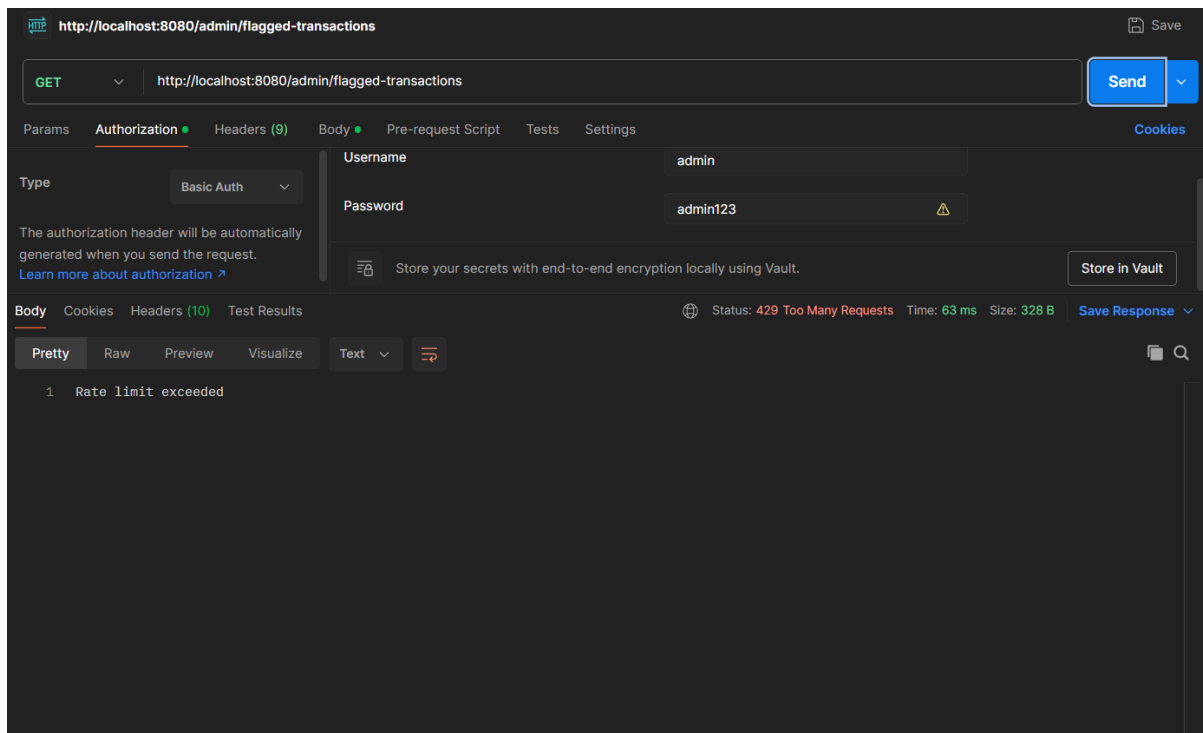


## BONUS FEATURES->

Rate\_Limiting\_usiing\_bucket\_4j->

The bucket can hold up to 5 tokens and is refilled with 5 tokens every 1 minute.





CODE\_BASE->

1.Code is attached as zip folder with file name ->transaction\_fraud\_controller

2.Code has been shared on git repo

Full response->

http://localhost:8080/transactions/process

[

```
{
  "status": "Fraud",
  "reason": "Transaction initiated from an IP address outside India."
},
{
  "status": "Fraud",
  "reason": "Transaction initiated from an IP address outside India."
},
{
  "status": "Fraud",
  "reason": "Transaction amount exceeds the limit of ₹1,00,000."
},
{
  "status": "Fraud",
  "reason": "Transaction amount exceeds the limit of ₹1,00,000."
},
{
  "status": "Fraud",
  "reason": "Transaction amount exceeds the limit of ₹1,00,000."
},
{
  "status": "Fraud",
  "reason": "Transaction amount exceeds the limit of ₹1,00,000."
}
```

```
{
  "status": "Success",
  "message": "Transaction is valid."
},
{
  "status": "Fraud",
  "reason": "Transaction amount exceeds the limit of ₹1,00,000."
},
{
  "status": "Fraud",
  "reason": "Transaction amount exceeds the limit of ₹1,00,000."
},
{
  "status": "Fraud",
  "reason": "Transaction amount exceeds the limit of ₹1,00,000."
},
{
  "status": "Success",
  "message": "Transaction is valid."
},
{
  "status": "Success",
  "message": "Transaction is valid."
},
{
  "status": "Success",
  "message": "Transaction is valid."
}
```

```
{
  "status": "Fraud",
  "reason": "Transaction amount exceeds the limit of ₹1,00,000."
},
{
  "status": "Success",
  "message": "Transaction is valid."
},
{
  "status": "Success",
  "message": "Transaction is valid."
},
{
  "status": "Success",
  "message": "Transaction is valid."
},
{
  "status": "Success",
  "message": "Transaction is valid."
},
{
  "status": "Success",
  "message": "Transaction is valid."
},
{
  "status": "Fraud",
  "reason": "Transaction amount exceeds the limit of ₹1,00,000."
},
```

```
{
  "status": "Success",
  "message": "Transaction is valid."
},
{
  "status": "Success",
  "message": "Transaction is valid."
},
{
  "status": "Fraud",
  "reason": "Transaction amount exceeds the limit of ₹1,00,000."
},
{
  "status": "Success",
  "message": "Transaction is valid."
},
{
  "status": "Success",
  "message": "Transaction is valid."
},
{
  "status": "Success",
  "message": "Transaction is valid."
},
{
  "status": "Success",
  "message": "Transaction is valid."
},
```

```
{
  "status": "Success",
  "message": "Transaction is valid."
},
{
  "status": "Fraud",
  "reason": "Transaction amount exceeds the limit of ₹1,00,000."
},
{
  "status": "Fraud",
  "reason": "Transaction amount exceeds the limit of ₹1,00,000."
},
{
  "status": "Fraud",
  "reason": "Transaction amount exceeds the limit of ₹1,00,000."
},
{
  "status": "Success",
  "message": "Transaction is valid."
},
{
  "status": "Success",
  "message": "Transaction is valid."
},
{
  "status": "Fraud",
  "reason": "Transaction amount exceeds the limit of ₹1,00,000."
},
```

```
{
  "status": "Fraud",
  "reason": "Transaction amount exceeds the limit of ₹1,00,000."
},
{
  "status": "Fraud",
  "reason": "Transaction amount exceeds the limit of ₹1,00,000."
},
{
  "status": "Success",
  "message": "Transaction is valid."
},
{
  "status": "Success",
  "message": "Transaction is valid."
},
{
  "status": "Fraud",
  "reason": "Transaction amount exceeds the limit of ₹1,00,000."
},
{
  "status": "Success",
  "message": "Transaction is valid."
},
{
  "status": "Fraud",
  "reason": "Transaction amount exceeds the limit of ₹1,00,000."
},
```

```
{
  "status": "Success",
  "message": "Transaction is valid."
},
{
  "status": "Success",
  "message": "Transaction is valid."
},
{
  "status": "Fraud",
  "reason": "Transaction amount exceeds the limit of ₹1,00,000."
}
]
```