

Banking Application

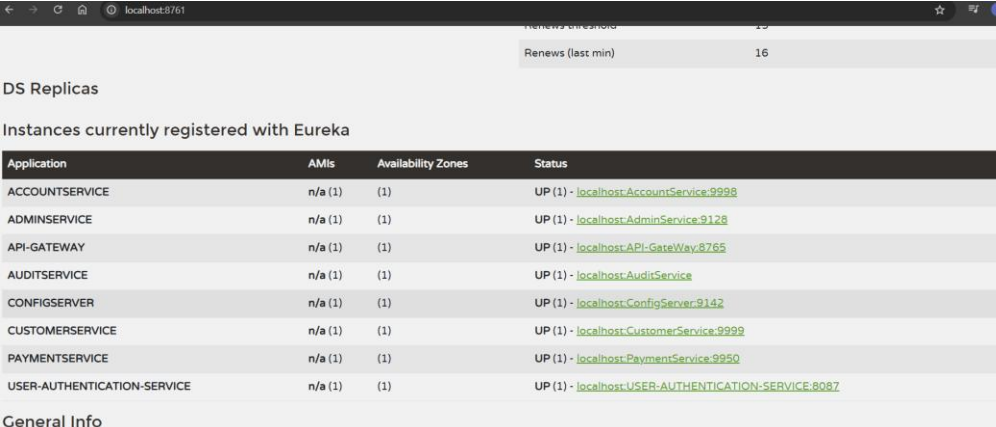
Overview

Services:-

1. DiscoveryServer
2. Config Server
3. API GateWay
4. Authentication Service
5. Customer Service
6. Account Service
7. Payment Service
8. Email Service
9. Audit Service
10. Admin Service

1)Discovery Server:

Purpose: Acts as a registry where all services in the system can register themselves and discover other services.



The screenshot shows the Eureka Discovery Server web interface. At the top, there's a navigation bar with 'Renews' and 'Refresh' buttons. Below that, a table shows 'Renews (last min)' with a value of 16. The main section is titled 'DS Replicas' and 'Instances currently registered with Eureka'. Below this is a table with columns: Application, AMIs, Availability Zones, and Status. The table lists several services: ACCOUNTSERVICE, ADMINSERVICE, API-GATEWAY, AUDITSERVICE, CONFIGSERVER, CUSTOMERSERVICE, PAYMENTSERVICE, and USER-AUTHENTICATION-SERVICE. Each service is shown with its AMIs, Availability Zones, and Status (UP (1) - localhost:ServiceName:Port).

Application	AMIs	Availability Zones	Status
ACCOUNTSERVICE	n/a (1)	(1)	UP (1) - localhost:AccountService:9998
ADMINSERVICE	n/a (1)	(1)	UP (1) - localhost:AdminService:9128
API-GATEWAY	n/a (1)	(1)	UP (1) - localhost:API-GateWay:8765
AUDITSERVICE	n/a (1)	(1)	UP (1) - localhost:AuditService
CONFIGSERVER	n/a (1)	(1)	UP (1) - localhost:ConfigServer:9142
CUSTOMERSERVICE	n/a (1)	(1)	UP (1) - localhost:CustomerService:9999
PAYMENTSERVICE	n/a (1)	(1)	UP (1) - localhost:PaymentService:9950
USER-AUTHENTICATION-SERVICE	n/a (1)	(1)	UP (1) - localhost:USER-AUTHENTICATION-SERVICE:8087

General Info

2)Config Server:

Purpose: Provides a centralized configuration for all microservices.

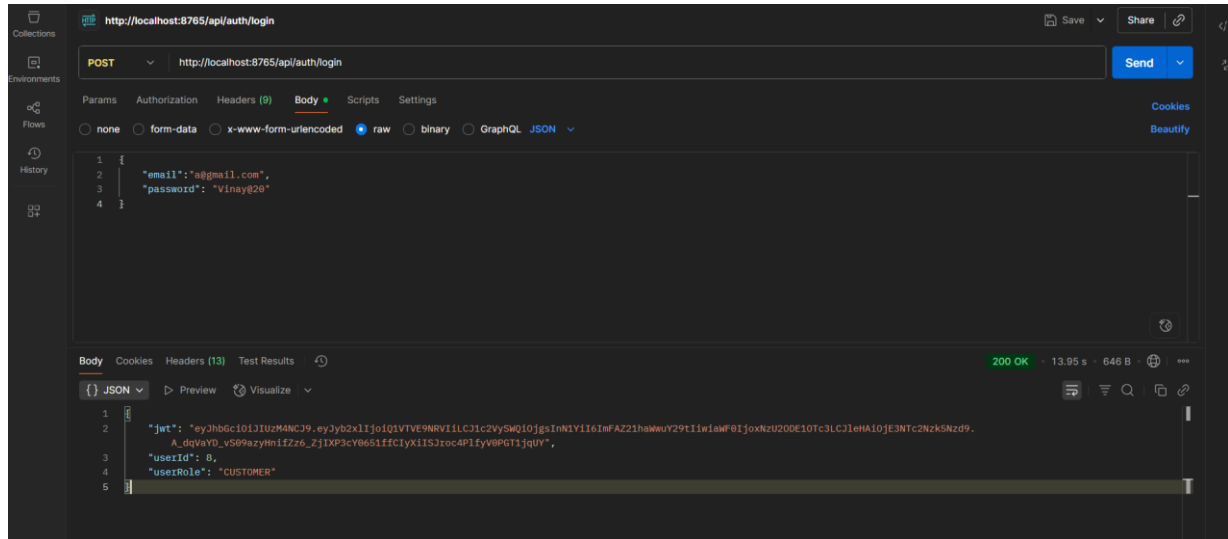
3)API Gateway:

Purpose: Serves as a single entry point for all client requests.

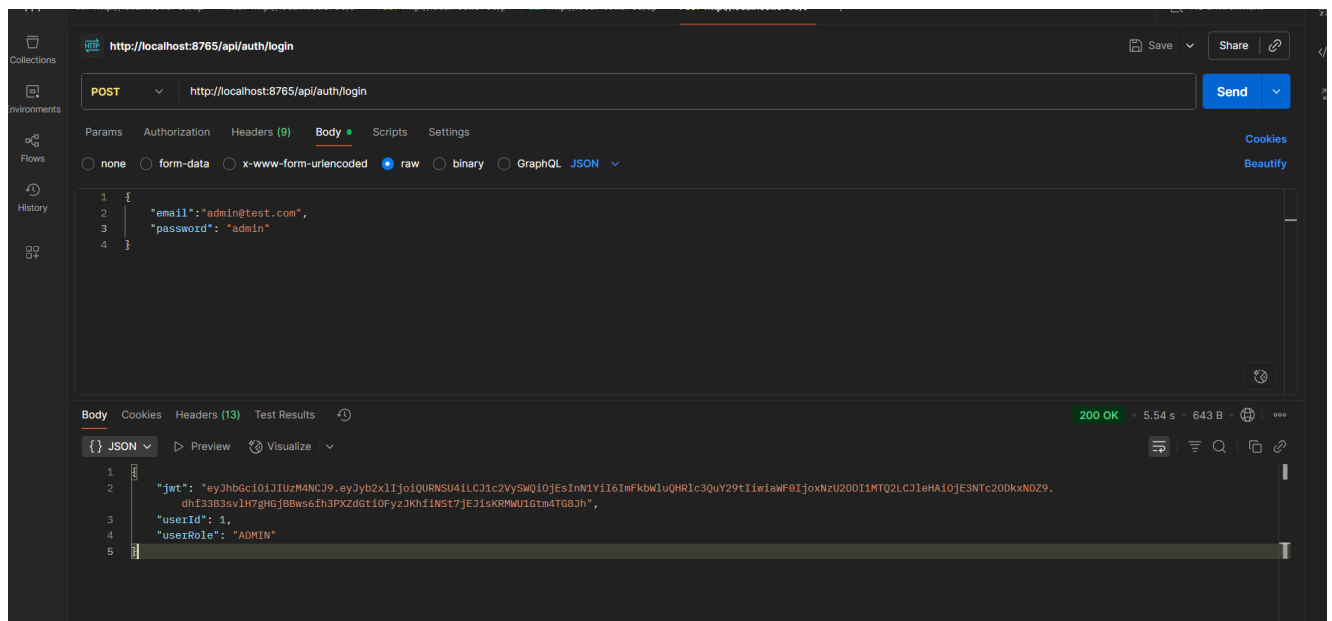
4)Authentication Server:

Purpose: Manages user login, token generation, and authorization.

Customer Login:



Admin Login



5) Customer Service:

Purpose: Adding Customer, Deleting Customer, Updating Customer and Manages all customer-related data such as profiles, personal details, and preferences.

It Uses MySQL Database to Store the Customer Details

Endpoints:-

- 1) /createcustomer
- 2) /updatecustomer/{id}
- 3) /getcustomerbyid/{id}
- 4) /getcustomerbyemail/{email}
- 5) /getcustomerbyname/{name}

6) Account Service:

Purpose: Handles bank account operations, including creation, retrieval, and balance tracking.

It Uses MySQL Database to Store the Bank Account Details and Also Connected through OpenFeign For the Communication with the Customer Service.

Endpoints:-

- 1) /createAccount
- 2) /CheckBalance
- 3) /updatebalance/{accountNumber}/{balance}
- 4) /getaccountbyid/{id}
- 5) /getaccountbyaccountnumber/{accountNumber}

7) Payment Service:

Purpose: Manages transactions, fund transfers

It is Connected Through FeignClient with Customer Service ,Account Service,Audit Service and through kafka with email Service.

It is connected with most of the microservices when the transaction is happened it stores that transaction details in the audit service and at the same time it sends email to both sender and receiver through kafka email service

It contains both internal and external transfer where in within the bank it checks the account number is present or not

Endpoints:-

- 1) /internal
- 2) /external

8) Email Service:

Purpose: Sends email notifications such as transaction alerts

It Sends Email for both Sender and Receiver.

9) Audit Service:

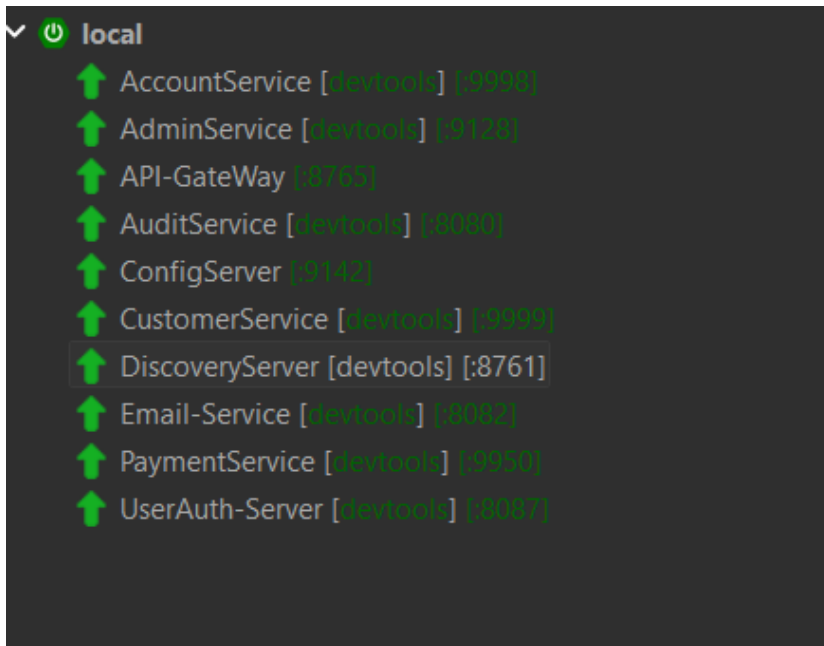
Purpose: Maintains a log of all critical actions.

It saves the all logs of the transactions and it is stored in mongoDB

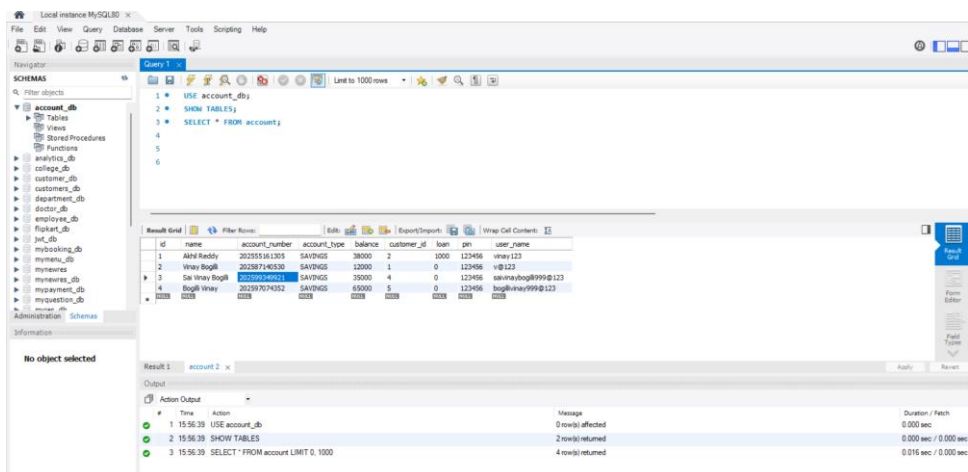
10) Admin Service:

Purpose: Provides an interface for administrators

It is Only for the admin logging



All Services UP



Accounts DB

Zoekopend

```

2025-09-02 15:20:27,794] INFO [LogLoader partition=__consumer_offsets-29, dir=C:\kafka_2.12-3.9.0\kafka-logs] Producer state recovery took 0ms for snapshot
Load and 0ms for segment recovery from offset 0 (kafka.log.UnifiedLog$)
2025-09-02 15:20:27,811] INFO [LogLoader partition=__consumer_offsets-29, dir=C:\kafka_2.12-3.9.0\kafka-logs] Loading producer state till offset 0 with message format version 2 (kafka.log.UnifiedLog$)
2025-09-02 15:20:27,818] INFO [LogLoader partition=__consumer_offsets-29, dir=C:\kafka_2.12-3.9.0\kafka-logs] Reloading from producer snapshot and rebuilding producer state from offset 0 (kafka.log.UnifiedLog$)
2025-09-02 15:20:27,819] INFO [LogLoader partition=__consumer_offsets-29, dir=C:\kafka_2.12-3.9.0\kafka-logs] Producer state recovery took 0ms for snapshot
Load and 0ms for segment recovery from offset 0 (kafka.log.UnifiedLog$)
2025-09-02 15:20:27,826] INFO [Completed load of Log(dir=C:\kafka_2.12-3.9.0\kafka-logs)\__consumer_offsets-29, topicId=mmUwP4MSAEae2iRHzoGLHg, topic=__consumer_offsets, partition=29, highWatermark=0, lastStableOffset=0, logStartOffset=0, logEndOffset=0) with 1 segments, local-log-start-offset 0 and log-end-offset 0 in 47ms (38/57 completed in C:\kafka_2.12-3.9.0\kafka-logs) (kafka.log.LogManager)
2025-09-02 15:20:27,891] INFO [LogLoader partition=__consumer_offsets-3, dir=C:\kafka_2.12-3.9.0\kafka-logs] Recovering unflushed segment 0. 0/1 recovered for __consumer_offsets-3. (kafka.log.LogLoader)
2025-09-02 15:20:27,847] INFO [LogLoader partition=__consumer_offsets-3, dir=C:\kafka_2.12-3.9.0\kafka-logs] Loading producer state till offset 0 with message format version 2 (kafka.log.UnifiedLog$)
2025-09-02 15:20:27,847] INFO [LogLoader partition=__consumer_offsets-3, dir=C:\kafka_2.12-3.9.0\kafka-logs] Reloading from producer snapshot and rebuilding producer state from offset 0 (kafka.log.UnifiedLog$)
2025-09-02 15:20:27,847] INFO [LogLoader partition=__consumer_offsets-3, dir=C:\kafka_2.12-3.9.0\kafka-logs] Producer state recovery took 0ms for snapshot
Load and 0ms for segment recovery from offset 0 (kafka.log.UnifiedLog$)
2025-09-02 15:20:27,861] INFO [LogLoader partition=__consumer_offsets-3, dir=C:\kafka_2.12-3.9.0\kafka-logs] Loading producer state till offset 0 with message format version 2 (kafka.log.UnifiedLog$)
2025-09-02 15:20:27,868] INFO [LogLoader partition=__consumer_offsets-3, dir=C:\kafka_2.12-3.9.0\kafka-logs] Reloading from producer snapshot and rebuilding producer state from offset 0 (kafka.log.UnifiedLog$)
2025-09-02 15:20:27,868] INFO [LogLoader partition=__consumer_offsets-3, dir=C:\kafka_2.12-3.9.0\kafka-logs] Producer state recovery took 0ms for snapshot
Load and 0ms for segment recovery from offset 0 (kafka.log.UnifiedLog$)
2025-09-02 15:20:27,872] INFO [Completed load of Log(dir=C:\kafka_2.12-3.9.0\kafka-logs)\__consumer_offsets-3, topicId=mmUwP4MSAEae2iRHzoGLHg, topic=__consumer_offsets, partition=3, highWatermark=0, lastStableOffset=0, logStartOffset=0, logEndOffset=0) with 1 segments, local-log-start-offset 0 and log-end-offset 0 in 46ms (31/57 completed in C:\kafka_2.12-3.9.0\kafka-logs) (kafka.log.LogManager)
2025-09-02 15:20:27,891] INFO [LogLoader partition=__consumer_offsets-30, dir=C:\kafka_2.12-3.9.0\kafka-logs] Recovering unflushed segment 0. 0/1 recovered for __consumer_offsets-30. (kafka.log.LogLoader)
2025-09-02 15:20:27,891] INFO [LogLoader partition=__consumer_offsets-30, dir=C:\kafka_2.12-3.9.0\kafka-logs] Loading producer state till offset 0 with message format version 2 (kafka.log.UnifiedLog$)
2025-09-02 15:20:27,891] INFO [LogLoader partition=__consumer_offsets-30, dir=C:\kafka_2.12-3.9.0\kafka-logs] Reloading from producer snapshot and rebuilding producer state from offset 0 (kafka.log.UnifiedLog$)
2025-09-02 15:20:27,892] INFO [LogLoader partition=__consumer_offsets-30, dir=C:\kafka_2.12-3.9.0\kafka-logs] Producer state recovery took 0ms for snapshot
Load and 0ms for segment recovery from offset 0 (kafka.log.UnifiedLog$)
2025-09-02 15:20:27,908] INFO [LogLoader partition=__consumer_offsets-30, dir=C:\kafka_2.12-3.9.0\kafka-logs] Loading producer state till offset 0 with message format version 2 (kafka.log.UnifiedLog$)
2025-09-02 15:20:27,909] INFO [LogLoader partition=__consumer_offsets-30, dir=C:\kafka_2.12-3.9.0\kafka-logs] Reloading from producer snapshot and rebuilding producer state from offset 0 (kafka.log.UnifiedLog$)

```

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.26100.5074]
(c) Microsoft Corporation. All rights reserved.

C:\kafka_2.12-3.9.0>.\bin\windows\kafka-topics.bat --create --topic payment-topic --bootstrap-server localhost:9092 --
--partitions 1 --replication-factor 1
Error while executing topic command : Topic 'payment-topic' already exists.
2025-09-02 15:29:04,295] ERROR org.apache.kafka.common.errors.TopicExistsException: Topic 'payment-topic' already exists
(org.apache.kafka.tools.TopicCommand)

C:\kafka_2.12-3.9.0>
```

Topic Creation

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.26100.5074]
(c) Microsoft Corporation. All rights reserved.

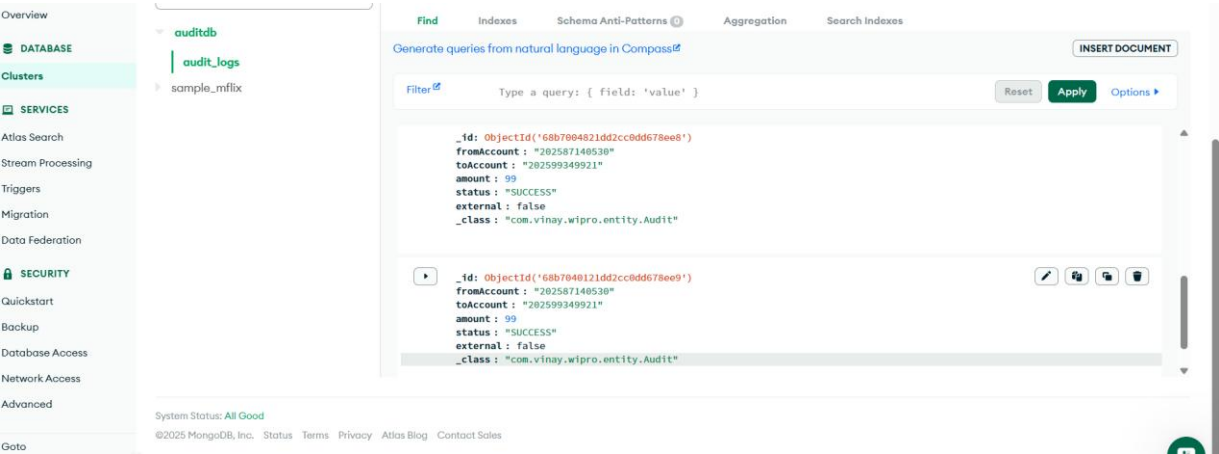
C:\kafka_2.12-3.9.0>. \bin\windows\kafka-topics.bat --list --bootstrap-server localhost:9092
__consumer_offsets
deliveryagent-location
payment-topic
sampledemo
zomato-delivery-agent

C:\kafka_2.12-3.9.0>
```

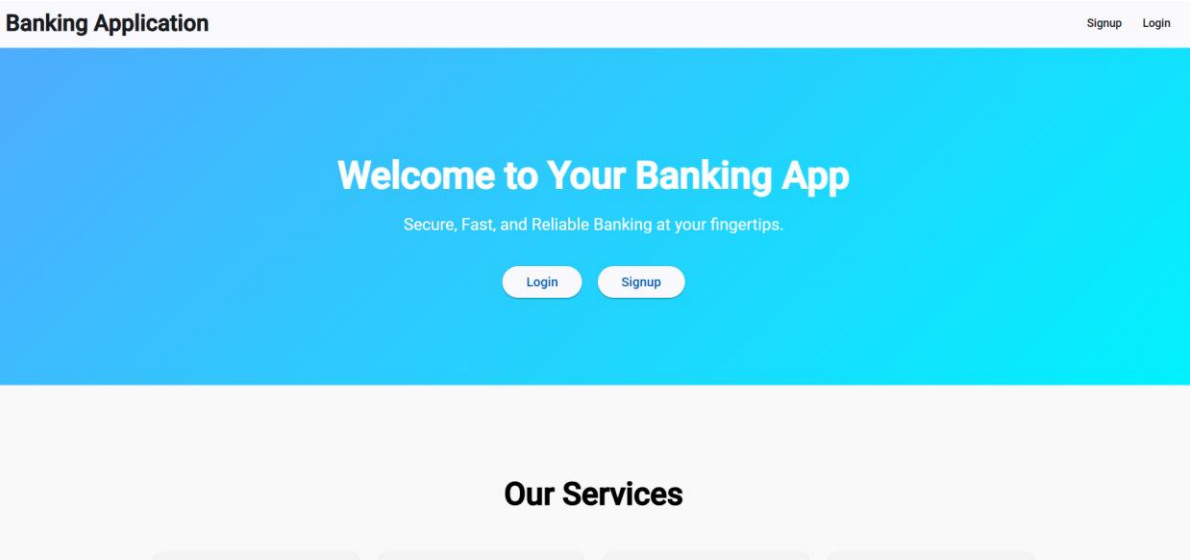
Topic List

```
C:\kafka_2.12-3.9.0>. \bin\windows\kafka-console-consumer.bat --topic payment-topic --from-beginning --bootstrap-server localhost:9092
{"email":"vinaybogilli999@gmail.com","subject":"Transaction Update","body":"SUCCESS","amount":5000}
{"email":"saivinaybogilli9999@gmail.com","subject":"Transaction Update","body":"SUCCESS","amount":5000}
C:\kafka_2.12-3.9.0>. \bin\windows\kafka-console-consumer.bat --topic payment-topic --from-beginning --bootstrap-server localhost:9092
{"email":"vinaybogilli999@gmail.com","subject":"Transaction Update","body":"SUCCESS","amount":5000}
{"email":"saivinaybogilli9999@gmail.com","subject":"Transaction Update","body":"SUCCESS","amount":5000}
C:\kafka_2.12-3.9.0>. \bin\windows\kafka-console-consumer.bat --topic payment-topic --from-beginning --bootstrap-server localhost:9092
{"email":"saivinaybogilli9999@gmail.com","subject":"Transaction Update","body":"SUCCESS","amount":99}
{"email":"vinay@example.com","subject":"Transaction Update","body":"SUCCESS","amount":99}
{"email":"vinaybogilli999@gmail.com","subject":"Transaction Update","body":"SUCCESS","amount":5000}
{"email":"saivinaybogilli9999@gmail.com","subject":"Transaction Update","body":"SUCCESS","amount":5000}
C:\kafka_2.12-3.9.0>. \bin\windows\kafka-console-consumer.bat --topic payment-topic --from-beginning --bootstrap-server localhost:9092
{"email":"saivinaybogilli9999@gmail.com","subject":"Transaction Update","body":"SUCCESS","amount":99}
{"email":"vinay@example.com","subject":"Transaction Update","body":"SUCCESS","amount":99}
2025-09-02 20:45:46,931] WARN [Consumer clientId=console-consumer, groupId=console-consumer-17353] Connection to node 0 (localhost/127.0.0.1:9092) could not be established. Node may not be available. (org.apache.kafka.clients.NetworkClient)
```

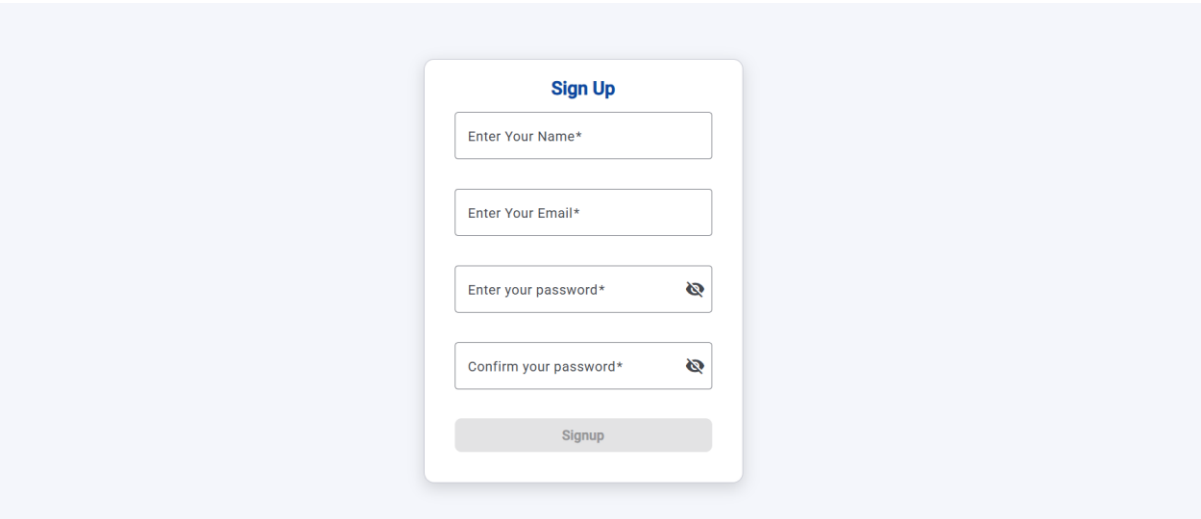
Consumer Running



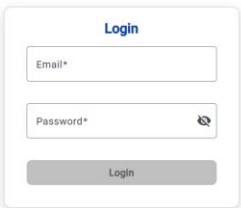
MongoDB



Home Page



Signup Page



Login Page

TrustBank

Add / Update Customer

Create Account

Payment

Customer Banking Dashboard

Add / Update Customer

Name*

Email*

Age*

Account Type*

Phone*

Address*

Aadhar*

PAN*

KYC Status

VERIFIED

Add Customer

UserDashboard

TrustBank

Add / Update Customer

Create Account

Payment

Customer Banking Dashboard

Add / Update Customer

venkatesh

Email*

venkatesh@gmail.com

Age*

22

Account Type*

SAVINGS

Phone*

9494949494

Address*

rajahmundry

Aadhar*

421387749996

PAN*

BSKQA2216H

KYC Status

VERIFIED

Update Customer

Customer added successfully!

Close

Customer Creation

TrustBank

+ Add / Update Customer

Create Account

Payment

Customer Banking Dashboard

Create Account

Full Name*
venkatesh

Username*
venky@123

PIN*

Account Type*
SAVINGS

Create Account

Account created successfully!Close

Account Creation

TrustBank

+ Add / Update Customer

Create Account

Payment

Customer Banking Dashboard

Make a Payment

Receiver Account Number*
202599349921

Amount*
100

Transaction Type
Internal

Send Payment

🔥 Current Balance: ₹500.00

📋 Transaction History

Before Transaction

TrustBank

+ Add / Update Customer

Create Account

Payment

Customer Banking Dashboard

Make a Payment

Receiver Account Number*
Amount*

Transaction Type
Internal

Send Payment

🔥 Current Balance: ₹400.00

📋 Transaction History

Payment successfulClose

Payment Successful

TrustBank Admin Dashboard

Logout

Customers

Accounts

Transactions

All Customers

Search by Name

Search

ID	Name	Email
1	Vinay Bogilli	vinay@example.com
2	Akhil Reddy	akhil@example.com
3	Vinay Bogilli	vinaybogilli999@gmail.com
4	Sai Vinay Bogilli	saivinaybogilli9999@gmail.com
5	Bogilli Vinay	vinaybogilli999@gmail.com
6	vvvvv	v@gmail.com
7	aaaaa	a@gmail.com

Admin Dashboard

All Accounts

Search by Account Number

Search

Account Number	User	Balance	Type
202555161305	vinay123	₹38,001.00	SAVINGS
202587140530	v@123	₹10,845.00	SAVINGS
202599349921	saivinaybogilli999@123	₹21,660.00	SAVINGS
202597074352	bogillivinay999@123	₹80,002.00	SAVINGS
202583483710	aa@123	₹500.00	SAVINGS
202594057383	aa@123	₹500.00	SAVINGS
202572705230	aa@123	₹500.00	SAVINGS
202534976975	aa@123	₹500.00	SAVINGS
202582719732	m@123	₹92.00	SAVINGS

Admin Accounts Dashboard

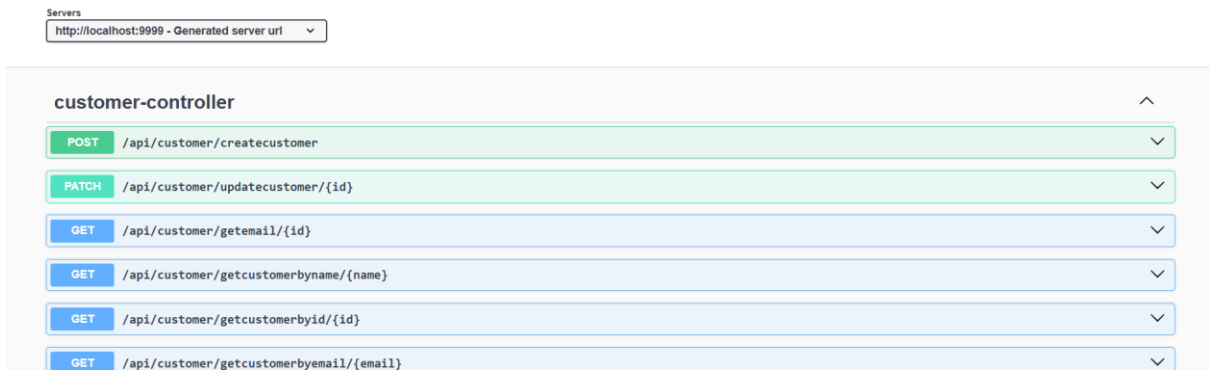
All Transactions

ID	From	To	Amount	Status	IsExternal
	202587140530	202599349921	₹99.00	SUCCESS	false
	202587140530	202599349921	₹99.00	SUCCESS	false
	202587140530	202599349921	₹99.00	SUCCESS	false
	202587140530	202599349921	₹99.00	SUCCESS	false
	202587140530	202599349921	₹99.00	SUCCESS	false
	202582719732	202599349921	₹99.00	SUCCESS	false
	202582719732	202599349921	₹99.00	SUCCESS	false
	202582719732	202599349921	₹99.00	SUCCESS	false
	202582719732	202597074352	₹1.00	SUCCESS	false

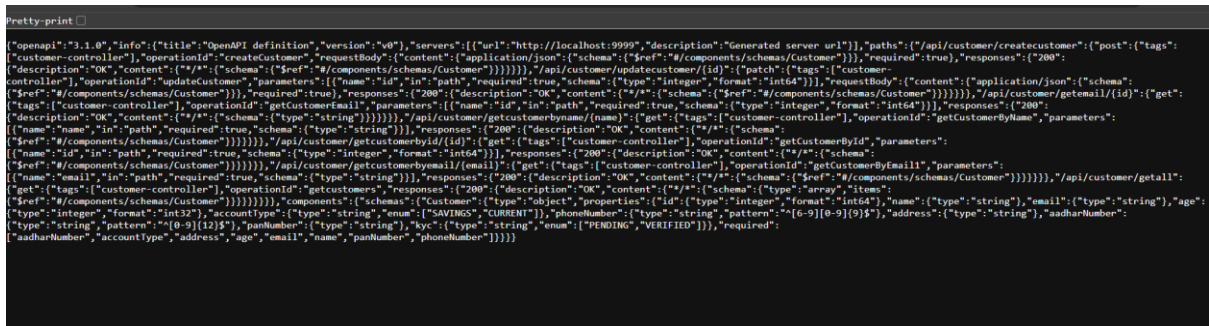
Admin Transactions Dashboard



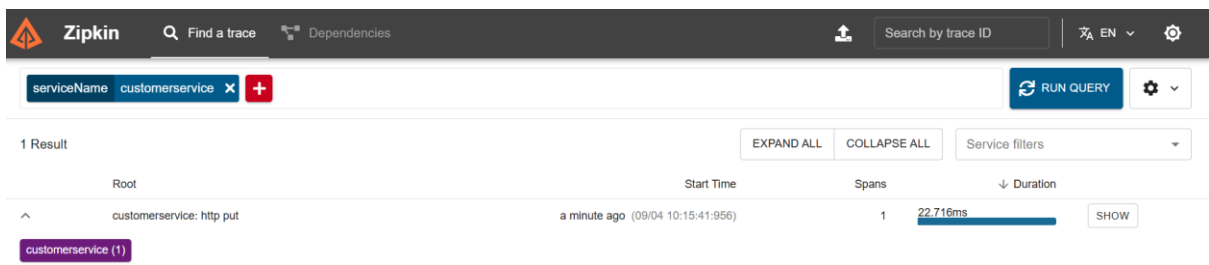
Receiver Email



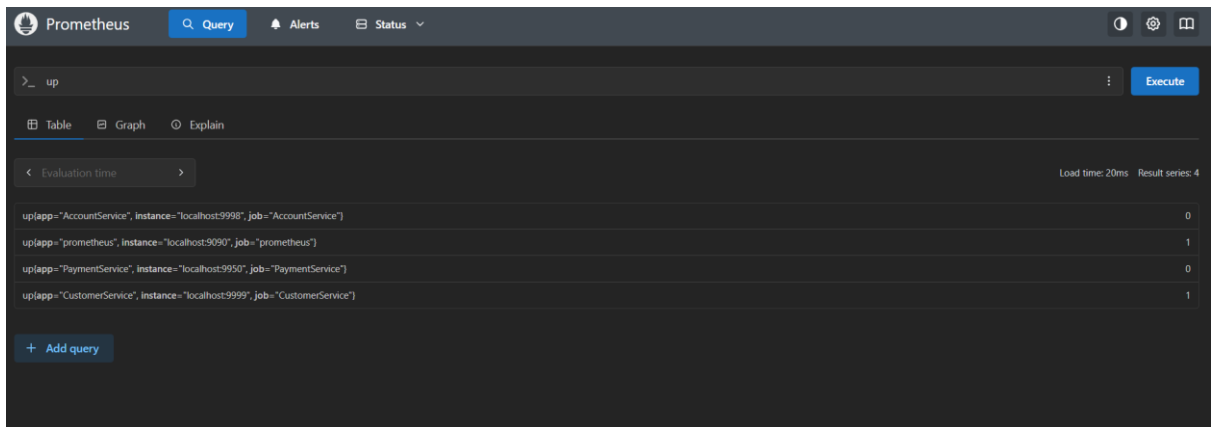
Swagger Ui



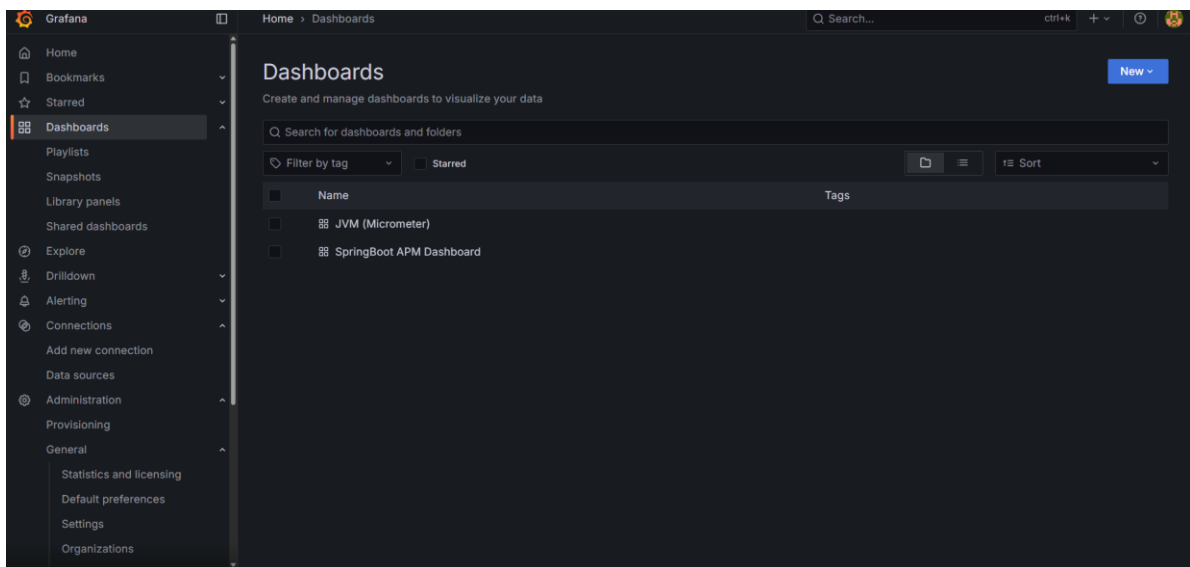
Open Api



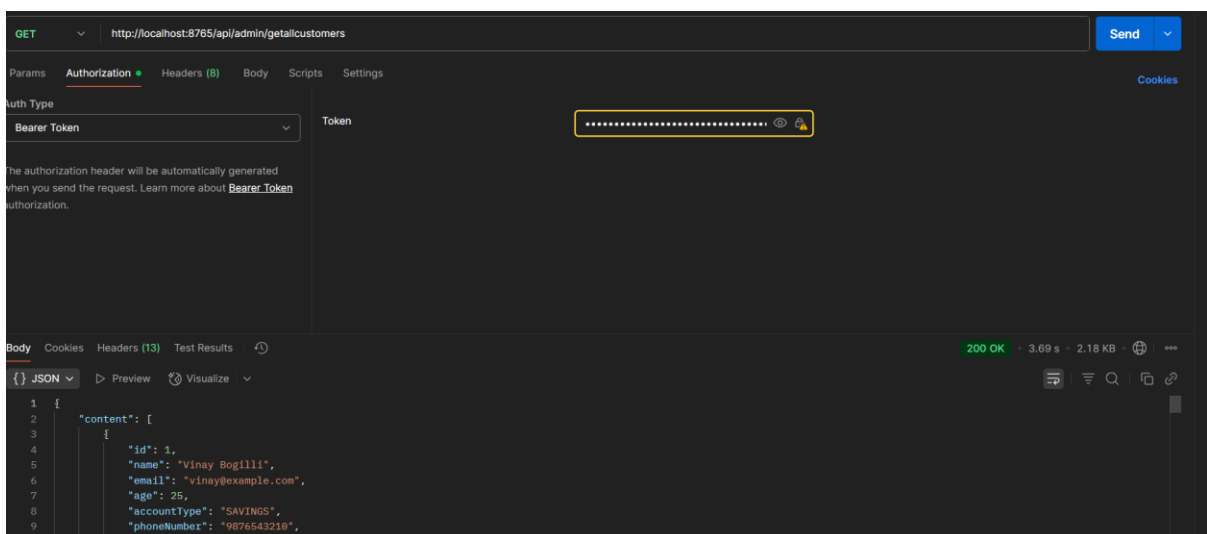
Zipkin



Prometheus



Grafna



Admin Service is opening with Admin token only