BANKING-APPLICATI0N

This document describes the complete microservices architecture of the Banking Applica on, including Discovery and Config servers, API Gateway, Auth, Customer, Account, Payment, Audit, Feedback, and No fica on services. It also explains Angular frontend integra on, Ka a messaging, and deployment architecture.

GlobalSystemOverview:

The Banking Applica on is implemented as a microservices ecosystem: - Angular Frontend for user interface - API Gateway as a single-entry point - Discovery Server for service registry - Config Server for centralized configura ons - Domain-driven microservices (Customer, Account, Payment, etc.) - Ka a and Zookeeper for messaging.

Discovery &ConfigServices- Discovery Service (Eureka): Registers and discovers microservices dynamically. - Simplifies API Gateway rou ng by avoiding hard-coded URLs. - Enables load balancing. Config Server: Centralized configura on management for all microservices. - Externalized proper es for different environments (dev, QA, prod). - Integrated with Git for version control.

API Gateway- Acts as a single entry point for client (Angular) requests. - Performs authen ca on, rou ng, and load balancing. - Handles CORS, rate limi ng, and logging.

Auth Service- Provides login, registra on, and JWT token genera on. - Integrates with API Gateway to enforce authoriza on policies. - Uses Spring Security for authen ca on.

Customer Service- Stores and manages customer profiles and KYC informa on. - Supports CRUD opera ons via REST endpoints. - Communicates with AuditService to log all profile updates.

Account Service- Manages account data including balances and account types. - Exposes APIs for fetching and upda ng balances. - Called by PaymentService for fund valida on.

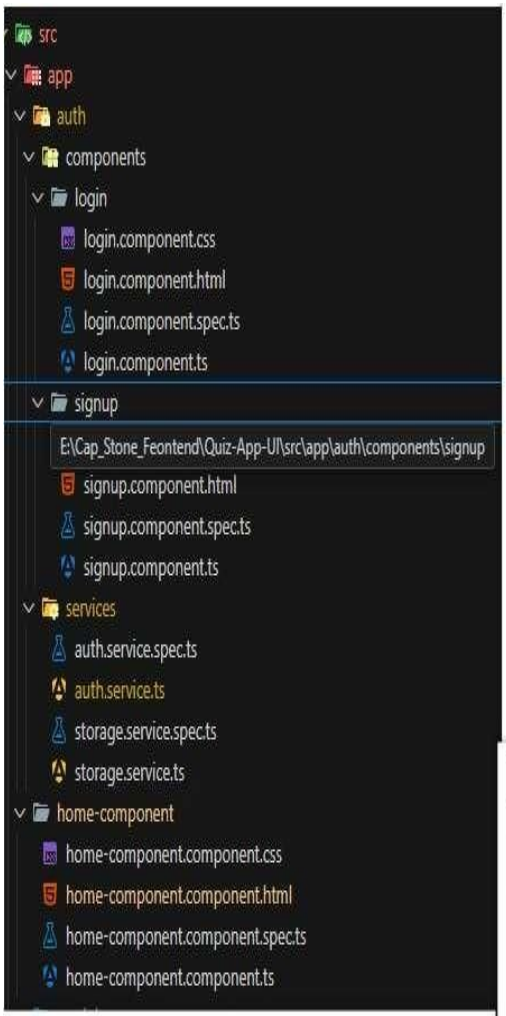
Payment Service- Handles fund transfers and payment workflows. - Communicates with AccountService, AuditService, and No fica onService. - Publishes Ka a events for asynchronous no fica ons.

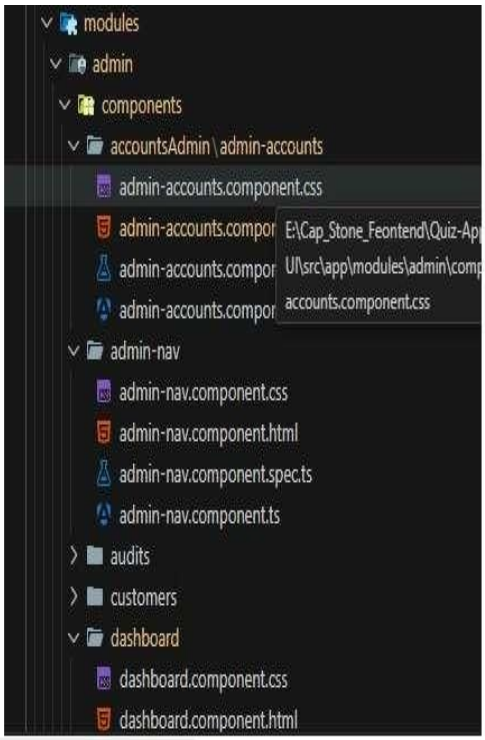
Audit Service- Maintains logs for all cri cal ac ons and transac ons. - Provides APIs for querying audit records. Integrated with PaymentService and CustomerService. Feedback Service- Collects and stores customer feedback. - Communicates with No fica onService to send alerts based on nega ve feedback. No fica on Service- Sends no fica ons via email. - Listens to Ka a events (e.g., payment confirma on)

Angular-Backend Communica on Flow 1. Angular sends authen ca on request via API Gateway. 2. JWT token is issued by AuthService and stored by Angular. 3. Angular makes secured requests to Gateway with token headers. 4. Gateway routes requests to respec ve services (Customer, Payment, etc.). 5. Ka a events no fy No fica onService asynchronously.

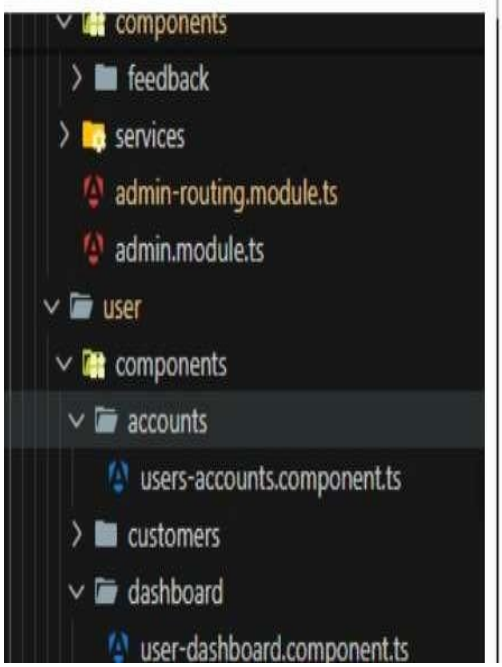
ANGULAR VIEW OUTPUTS

Folder Structure:


src
app
auth
components
login
login.component.css
login.component.html
login.component.spec.ts
login.component.ts
signup
E:\Cap_Stone_Feontend\Quiz-App-UI\src\app\auth\components\signup
signup.component.html
signup.component.spec.ts
signup.component.ts
services
auth.service.spec.ts
auth.service.ts
storage.service.spec.ts
storage.service.ts
home-component
home-component.component.css
home-component.component.html
home-component.component.spec.ts
home-component.component.ts

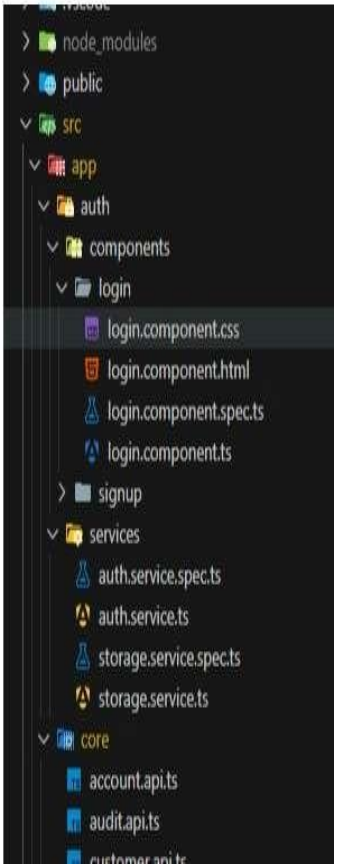

modules
admin
components
accountsAdmin\admin-accounts
admin-accounts.component.css
admin-accounts.compor
admin-accounts.compor
admin-accounts.compor
E:\Cap_Stone_Feontend\Quiz-Ap
UI\src\app\modules\admin\comp
accounts.component.css
admin-nav
admin-nav.component.css
admin-nav.component.html
admin-nav.component.spec.ts
admin-nav.component.ts
audits
customers
dashboard
dashboard.component.css
dashboard.component.html

modules
admin
components
add-questions
add-questions.component.css
add-questions.component.html
add-questions.component.spec.ts
add-questions.component.ts
admin-nav
admin-nav.component.css
admin-nav.component.html
admin-nav.component.spec.ts
admin-nav.component.ts
dashboard
dashboard.component.css
dashboard.component.html
dashboard.component.spec.ts
dashboard.component.ts
services

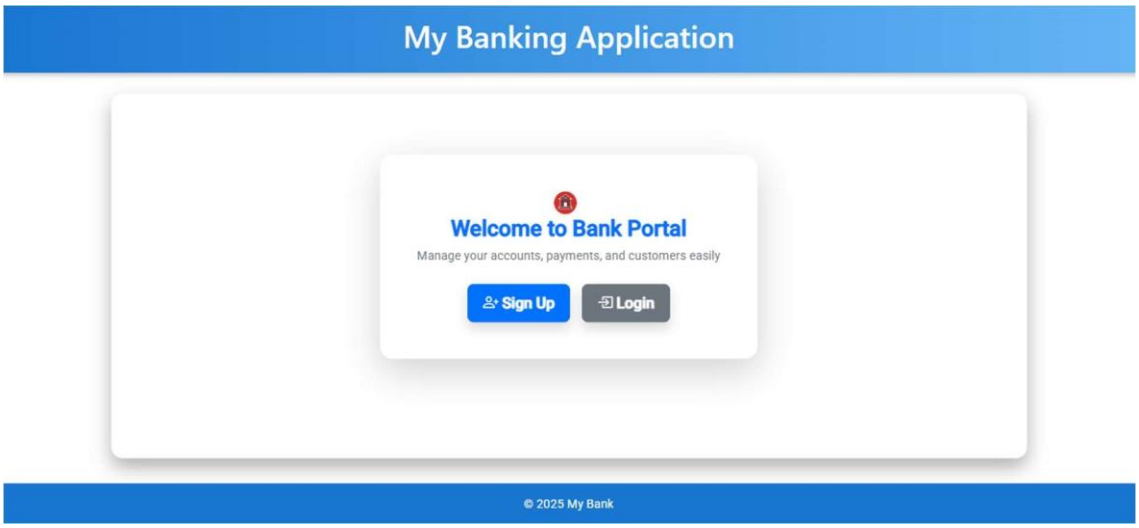

components
feedback
services
admin-routing.module.ts
admin.module.ts
user
components
accounts
users-accounts.component.ts
customers
dashboard
user-dashboard.component.ts

QUIZ-APP-UI
src
app
app.config.server.ts
app.config.ts
app.routes.server.ts
app.routes.ts
material.module.ts
environments
environment.prod.ts
environment.ts
index.html
main.server.ts
main.ts
server.ts
styles.css
.editorconfig
.gitignore
angular.json
package-lock.json
package.json
README.md
tsconfig.app.json
tsconfig.json
tsconfig.spec.json

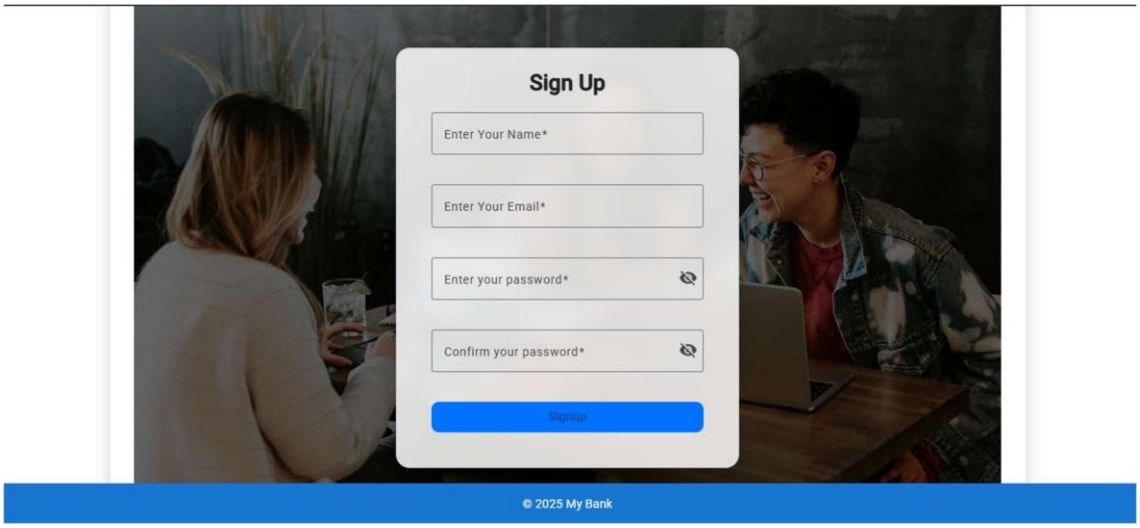
user
componets\dashboard
dashboard.component.css
dashboard.component.html
dashboard.component.spec.ts
dashboard.component.ts
user-routing.module.ts
user.module.ts
app.component.css
app.component.html
app.component.spec.ts
app.component.ts
app.config.server.ts
app.config.ts
app.routes.server.ts
app.routes.ts
material.module.ts
index.html
main.server.ts
main.ts


node_modules
public
src
app
auth
components
login
login.component.css
login.component.html
login.component.spec.ts
login.component.ts
signup
services
auth.service.spec.ts
auth.service.ts
storage.service.spec.ts
storage.service.ts
core
account.api.ts
audit.api.ts

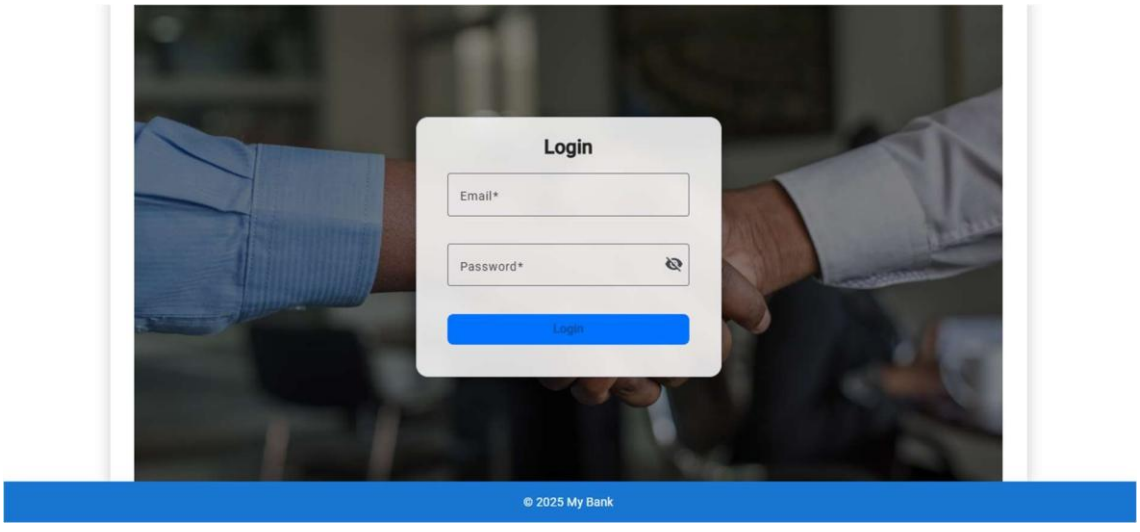
HOME PAGE:



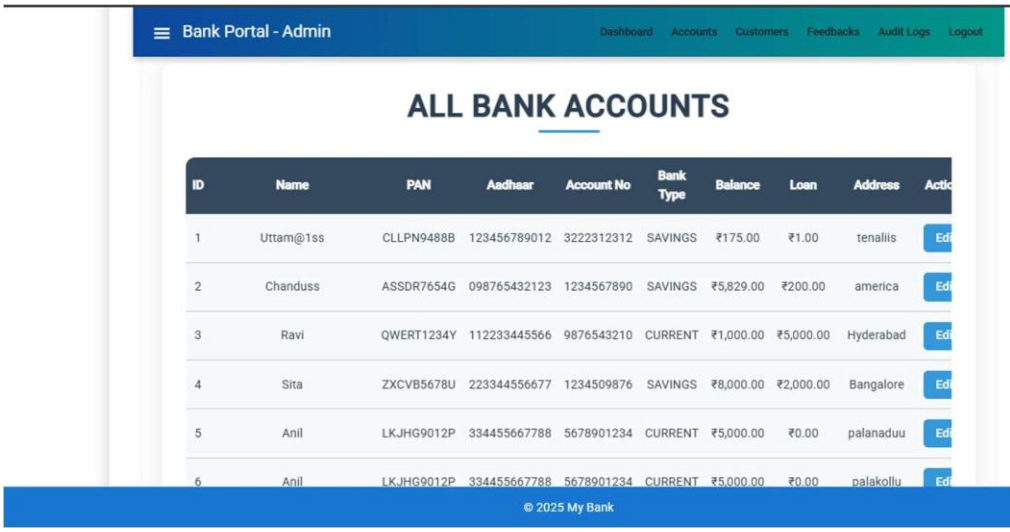
SIGNUP PAGE:

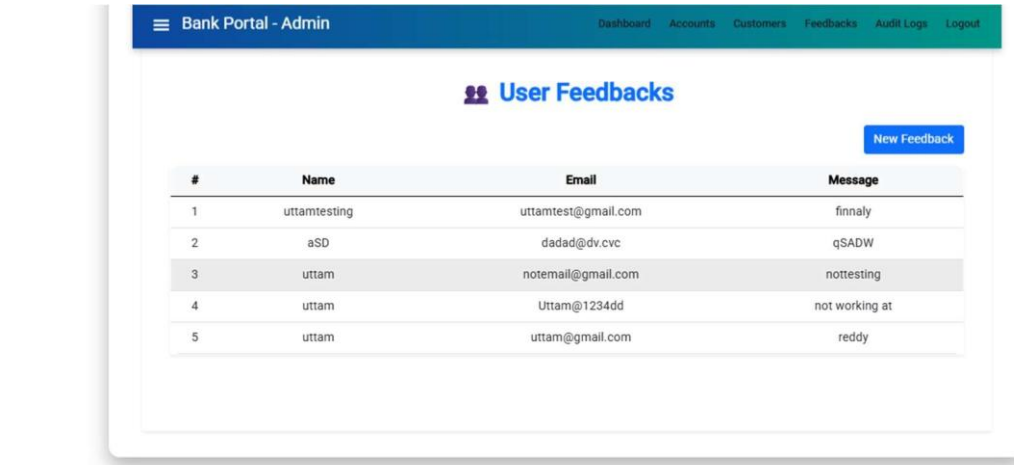


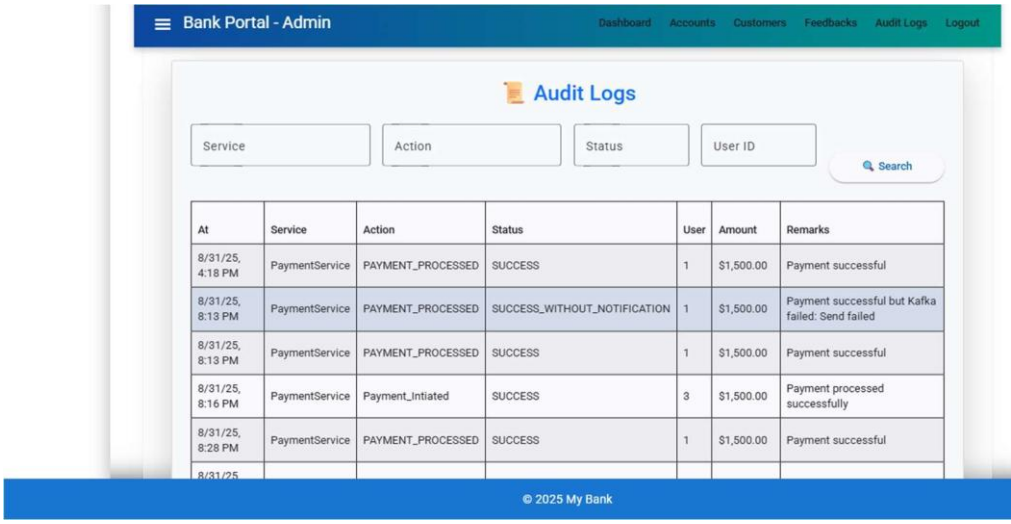
LOGIN PAGE:

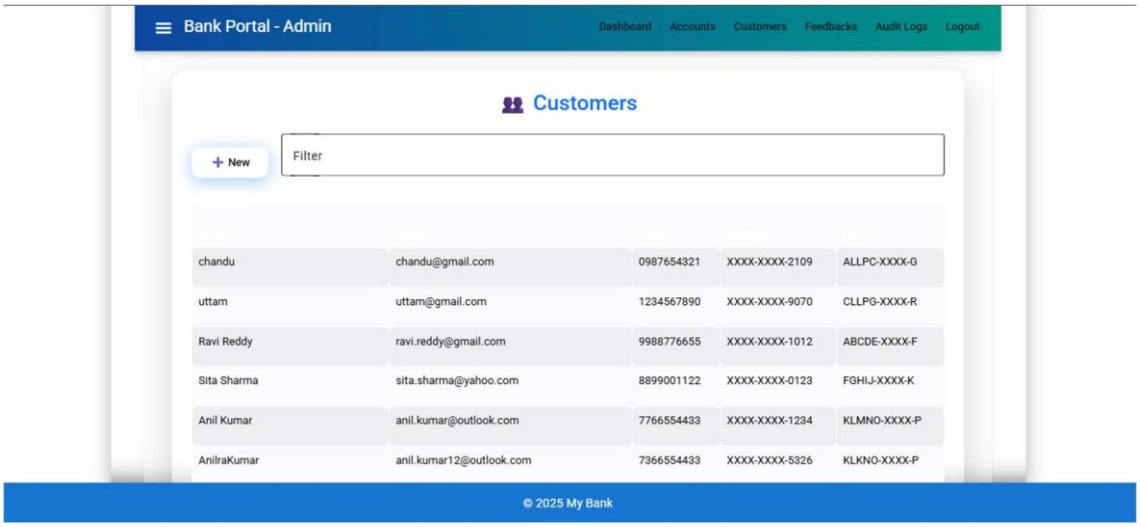


ADMIN COMPONENTS VIEWS:

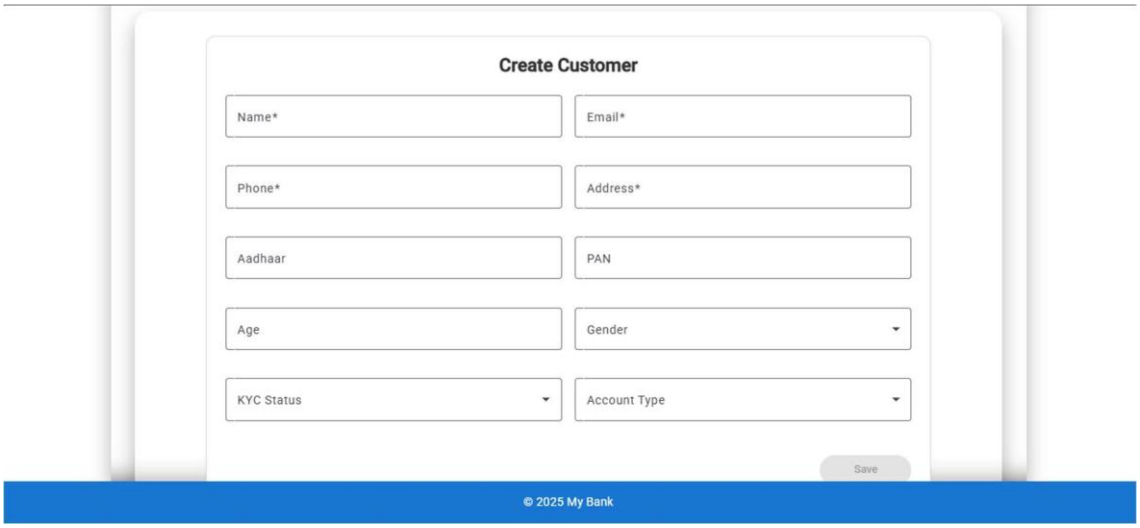

Bank Portal - Admin
Dashboard
Accounts
Customers
Feedbacks
Audit Logs
Logout
ALL BANK ACCOUNTS
ID
Name
PAN
Aadhaar
Account No
Bank Type
Balance
Loan
Address
1 Uttam@1ss CLLPN9488B 123456789012 3222312312 SAVINGS ₹175.00 ₹1.00 tenaliis
2 Chanduss ASSDR7654G 098765432123 1234567890 SAVINGS ₹5,829.00 ₹200.00 america
3 Ravi QWERT1234Y 112233445566 9876543210 CURRENT ₹1,000.00 ₹5,000.00 Hyderabad
4 Sita ZXCVB5678U 223344556677 1234509876 SAVINGS ₹8,000.00 ₹2,000.00 Bangalore
5 Anil LKJHG9012P 334455667788 5678901234 CURRENT ₹5,000.00 ₹0.00 palanaduu
6 Anil LKJHG9012P 334455667788 5678901234 CURRENT ₹5,000.00 ₹0.00 palakollu
© 2025 My Bank

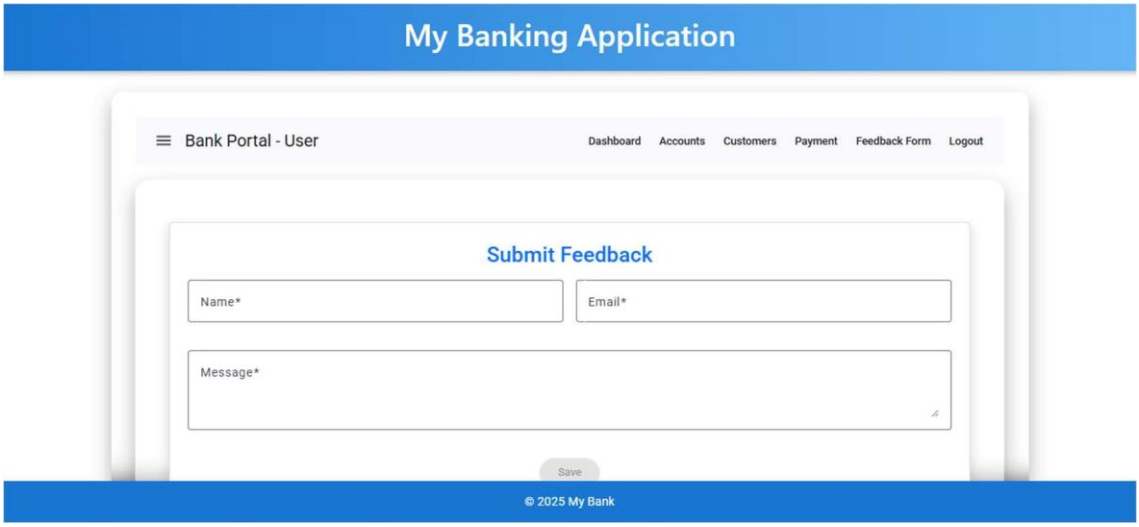

Bank Portal - Admin
Dashboard
Accounts
Customers
Feedbacks
Audit Logs
Logout
User Feedbacks
New Feedback
Name Email Message
1 uttamtesting uttamtest@gmail.com finnaly
2 aSD dadad@dv.cvc qSADW
3 uttam notemail@gmail.com nottesting
4 uttam Uttam@1234dd not working at
5 uttam uttam@gmail.com reddy

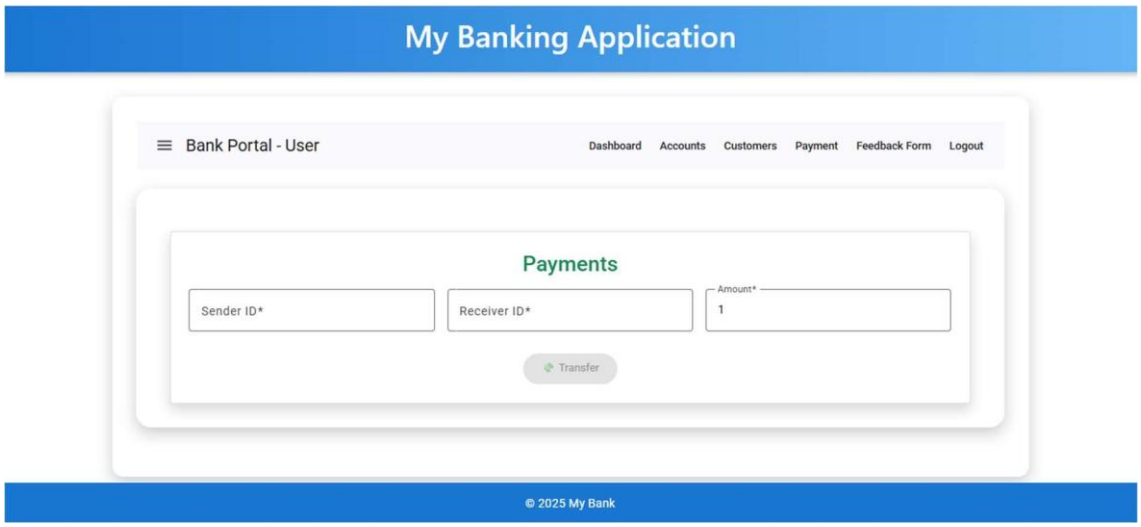

Bank Portal - Admin
Dashboard
Accounts
Customers
Feedbacks
Audit Logs
Logout
Audit Logs
Service
Action
Status
User ID
Search
At Service Action Status User Amount Remarks
8/31/25, 4:18 PM PaymentService PAYMENT_PROCESSED SUCCESS 1 $1,500.00 Payment successful
8/31/25, 8:13 PM PaymentService PAYMENT_PROCESSED SUCCESS_WITHOUT_NOTIFICATION 1 $1,500.00 Payment successful but Kafka failed: Send failed
8/31/25, 8:13 PM PaymentService PAYMENT_PROCESSED SUCCESS 1 $1,500.00 Payment successful
8/31/25, 8:16 PM PaymentService Payment_Intiated SUCCESS 3 $1,500.00 Payment processed successfully
8/31/25, 8:28 PM PaymentService PAYMENT_PROCESSED SUCCESS 1 $1,500.00 Payment successful
© 2025 My Bank



USER COMPONENTS VIEWS:

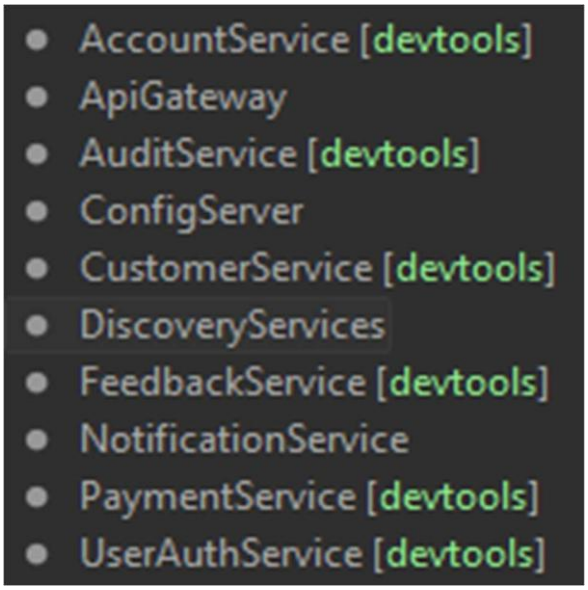






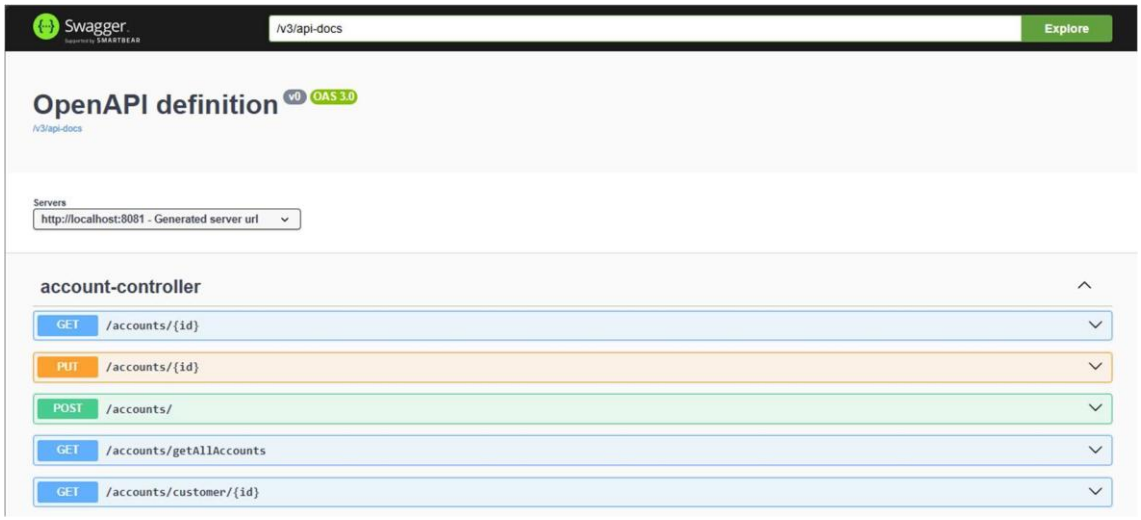
JAVA-SPRING_BOOT_MICROSERVICES:

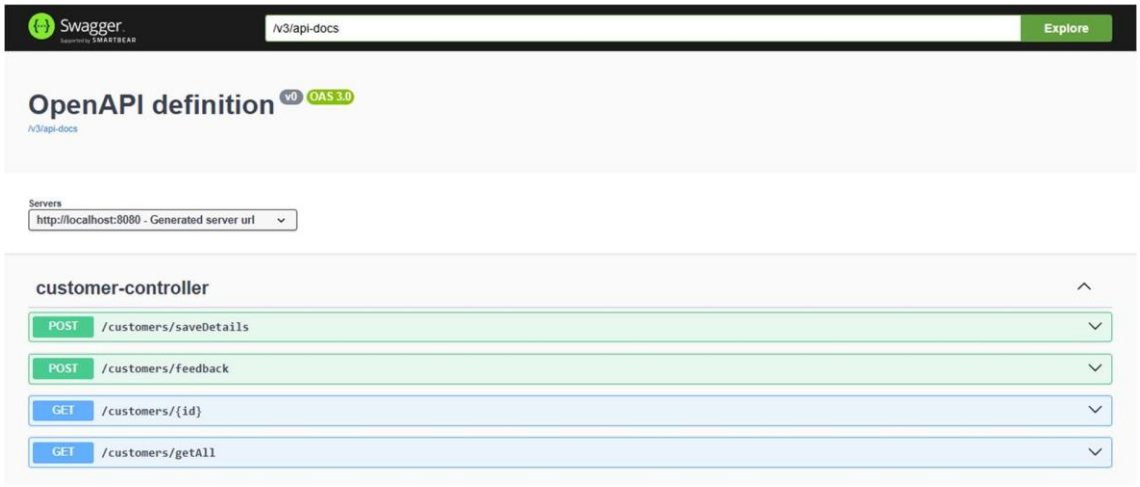
SERVICES:

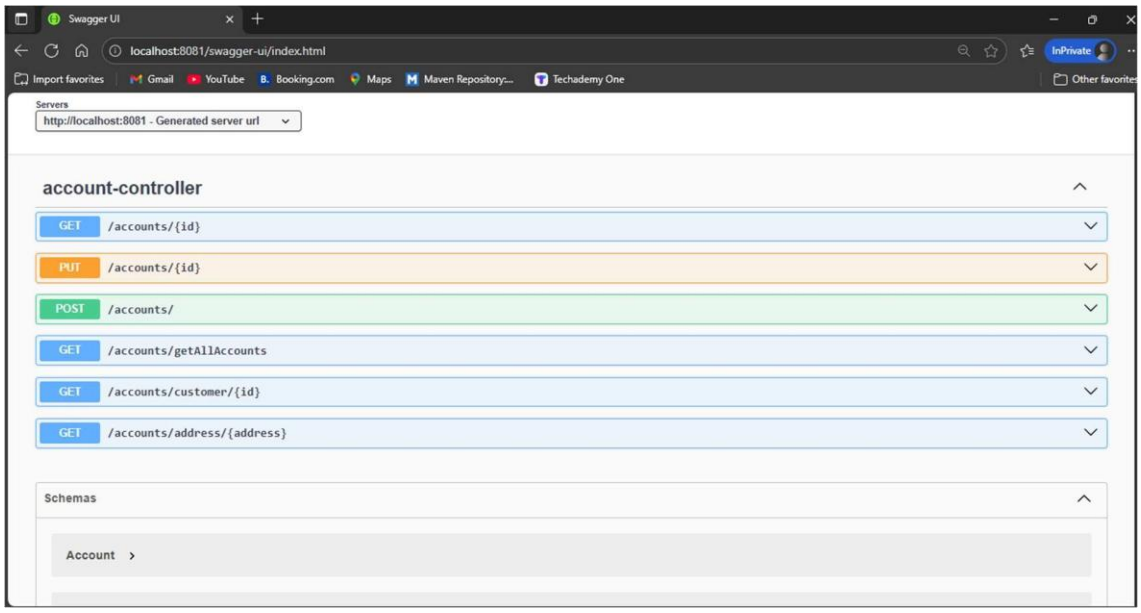


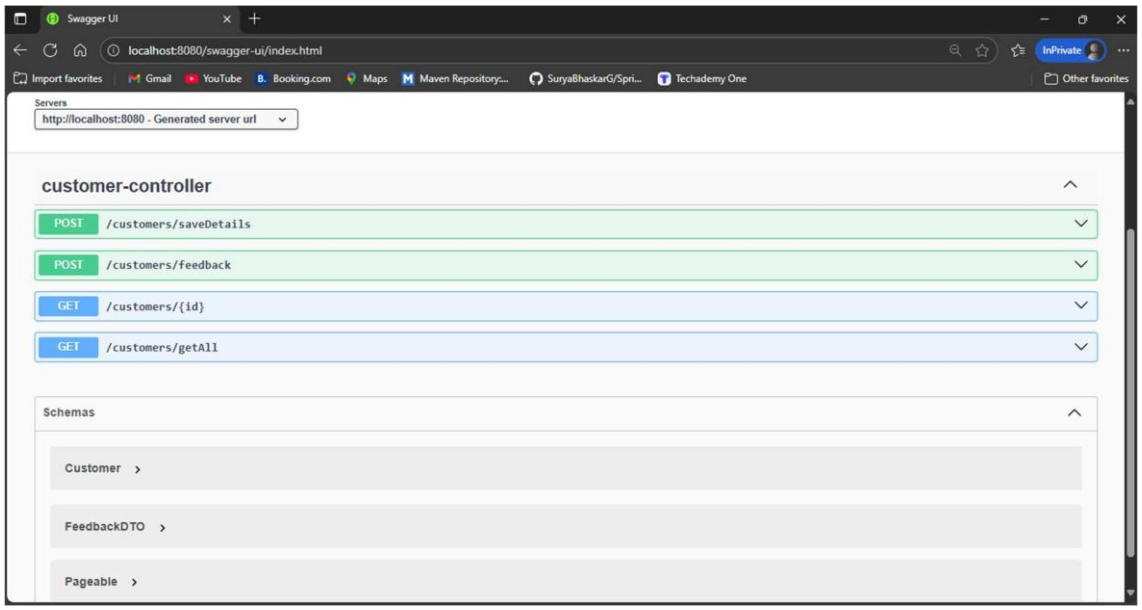
Service	Responsibility	Database
API Gateway	Routes client requests, handles load balancing	N/A
AuthService	User authentication, JWT issuance	AuthDB
AccountService	Account data and balance management	AccountDB
CustomerService	Customer profile and KYC data	CustomerDB
PaymentService	Handles payments, transaction workflow	PaymentDB
FeedbackService	User feedback storage and reporting	FeedbackDB
NotificationService	Async notifications via Kafka	NotificationDB
AuditService	Transaction logs, audit trails	AuditDB

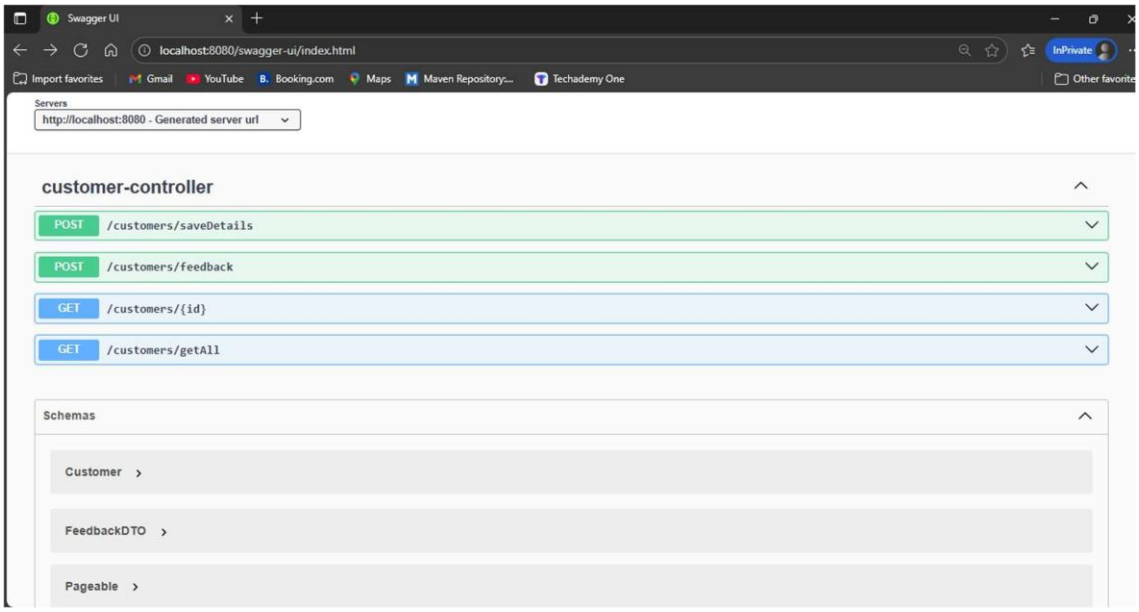
SWAGGER and ZIPKIN and PROMETHEUS and GRAFANA :

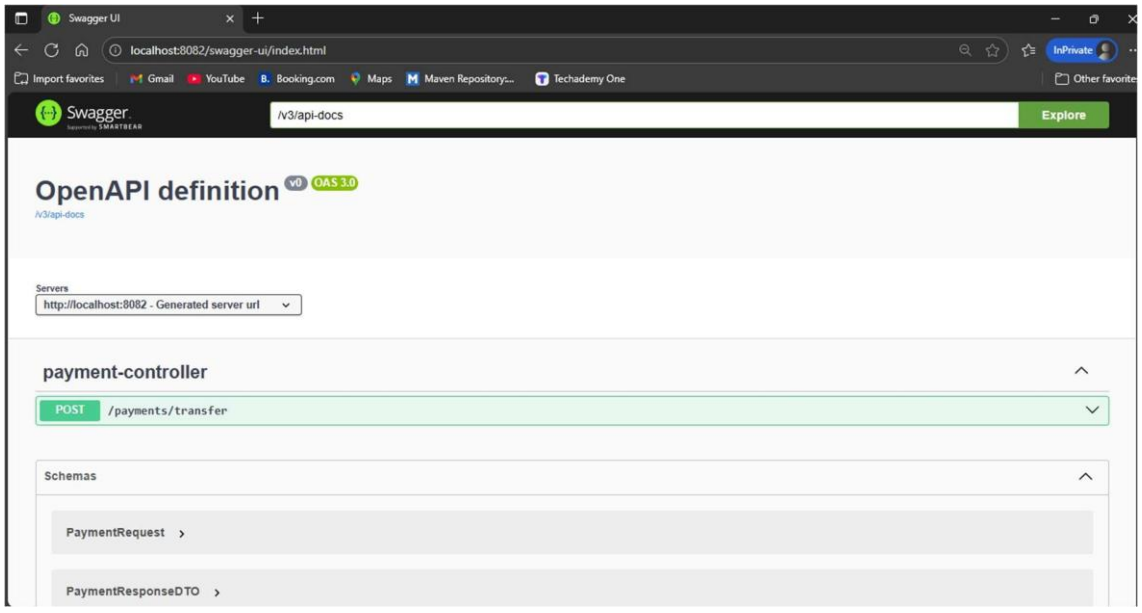

Swagger
/v3/api-docs
Explore
OpenAPI definition
v0
OAS 3.0
/v3/api-docs
Servers
http://localhost:8081 - Generated server url
account-controller
GET /accounts/{id}
PUT /accounts/{id}
POST /accounts/
GET /accounts/getAllAccounts
GET /accounts/customer/{id}

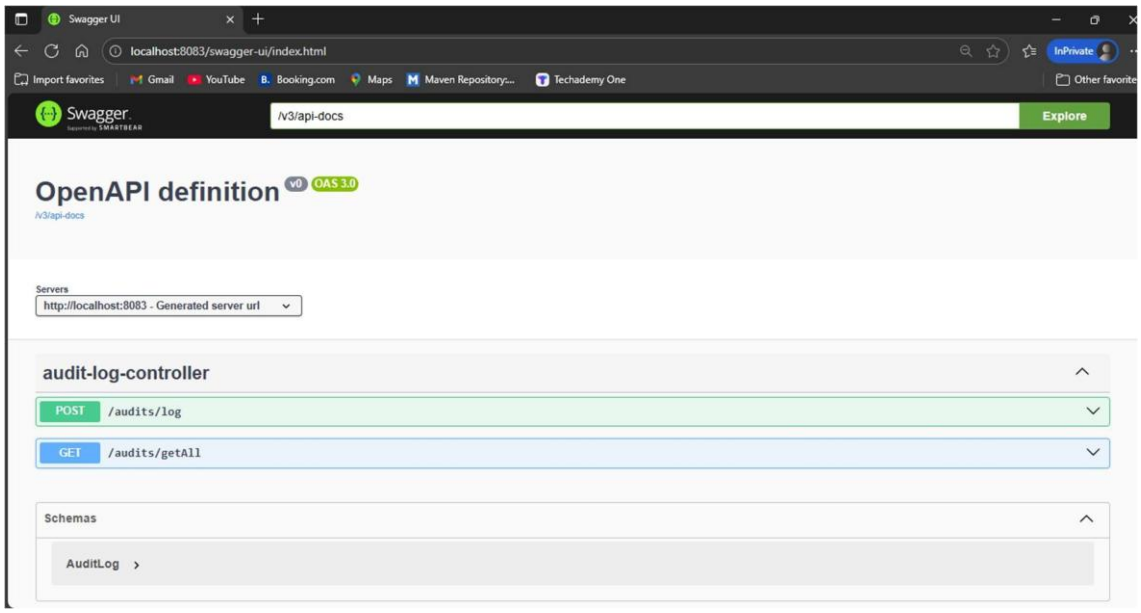

Swagger
/v3/api-docs
Explore
OpenAPI definition
v0
OAS 3.0
/v3/api-docs
Servers
http://localhost:8080 - Generated server url
customer-controller
POST /customers/saveDetails
POST /customers/feedback
GET /customers/{id}
GET /customers/getAll

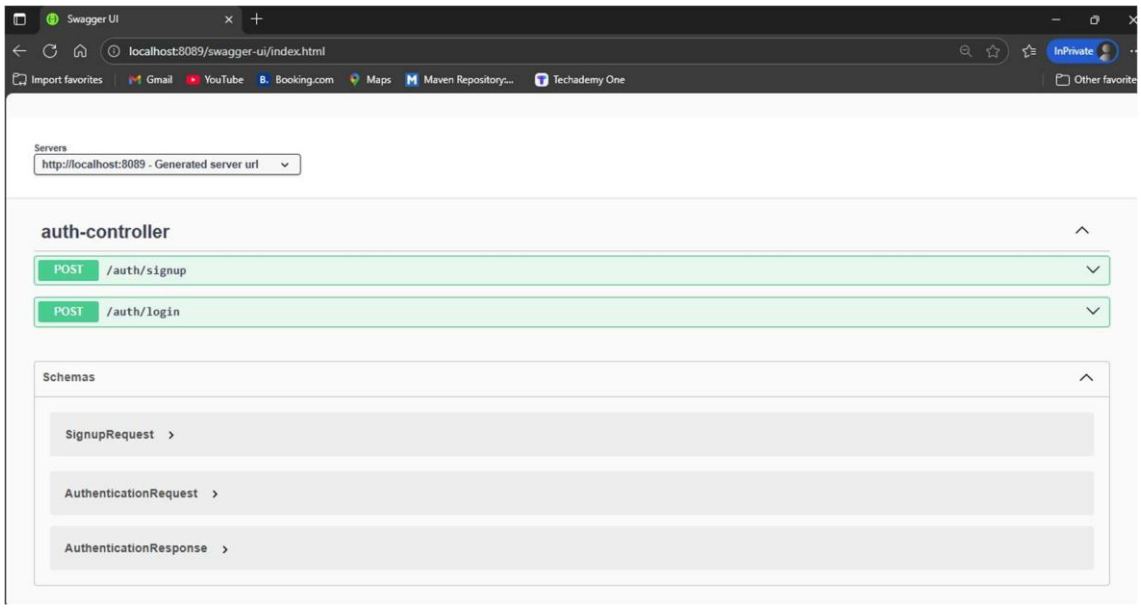

Swagger UI
localhost:8081/swagger-ui/index.html
InPrivate
Import favorites
Gmail
YouTube
Booking.com
Maps
Maven Repository:...
Techademy One
Other favorites
Servers
http://localhost:8081 - Generated server url
account-controller
GET /accounts/{id}
PUT /accounts/{id}
POST /accounts/
GET /accounts/getAllAccounts
GET /accounts/customer/{id}
GET /accounts/address/{address}
Schemas
Account

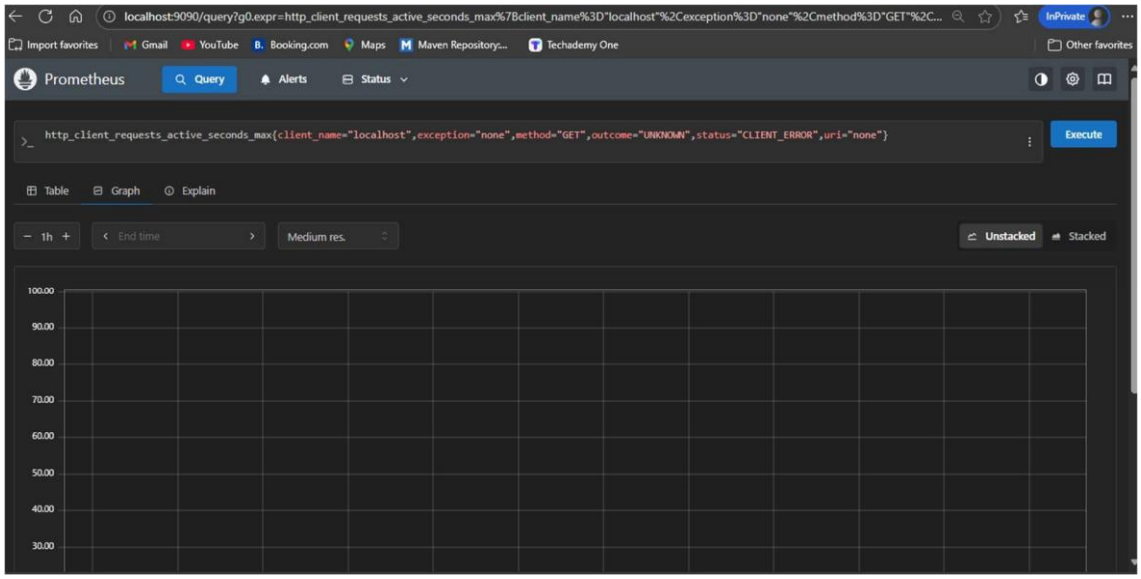

Swagger UI
localhost:8080/swagger-ui/index.html
Servers
http://localhost:8080 - Generated server url
customer-controller
POST /customers/saveDetails
POST /customers/feedback
GET /customers/{id}
GET /customers/getAll
Schemas
Customer
FeedbackDTO
Pageable

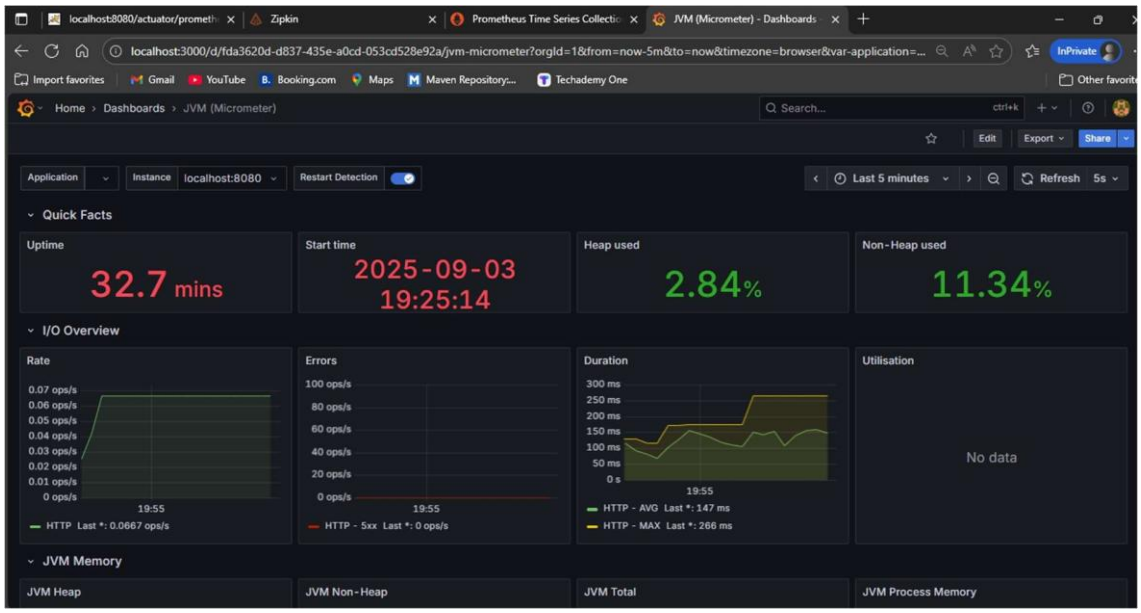

Swagger UI
localhost:8080/swagger-ui/index.html
Servers
http://localhost:8080 - Generated server url
customer-controller
POST /customers/saveDetails
POST /customers/feedback
GET /customers/{id}
GET /customers/getAll
Schemas
Customer
FeedbackDTO
Pageable

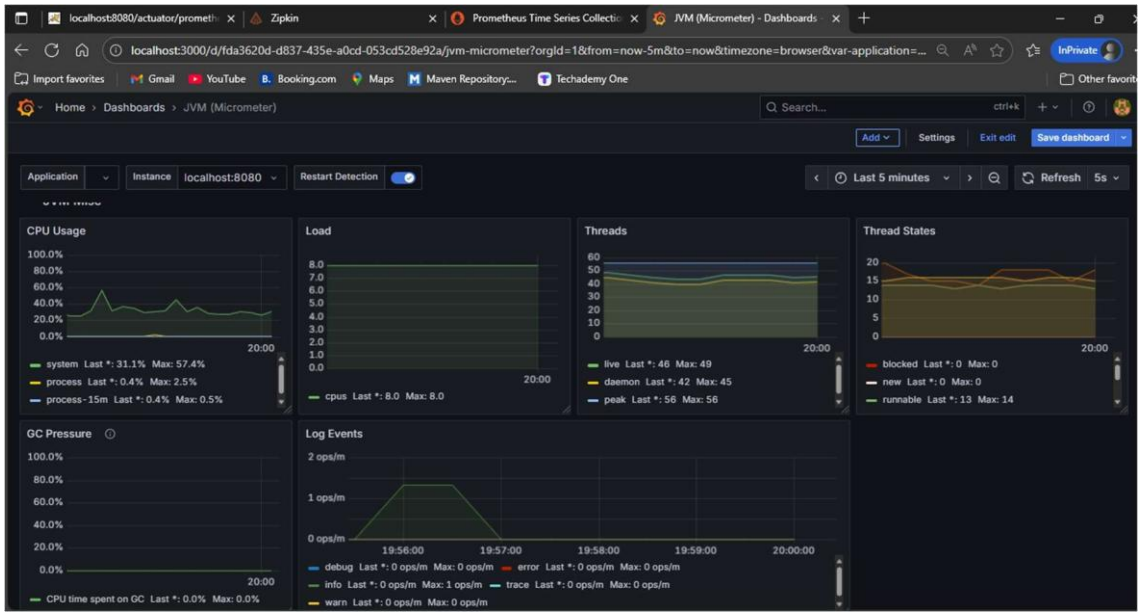

Swagger UI
localhost:8082/swagger-ui/index.html
Swagger
/v3/api-docs
Explore
OpenAPI definition
v0
OAS 3.0
/v3/api-docs
Servers
http://localhost:8082 - Generated server url
payment-controller
POST /payments/transfer
Schemas
PaymentRequest
PaymentResponseDTO

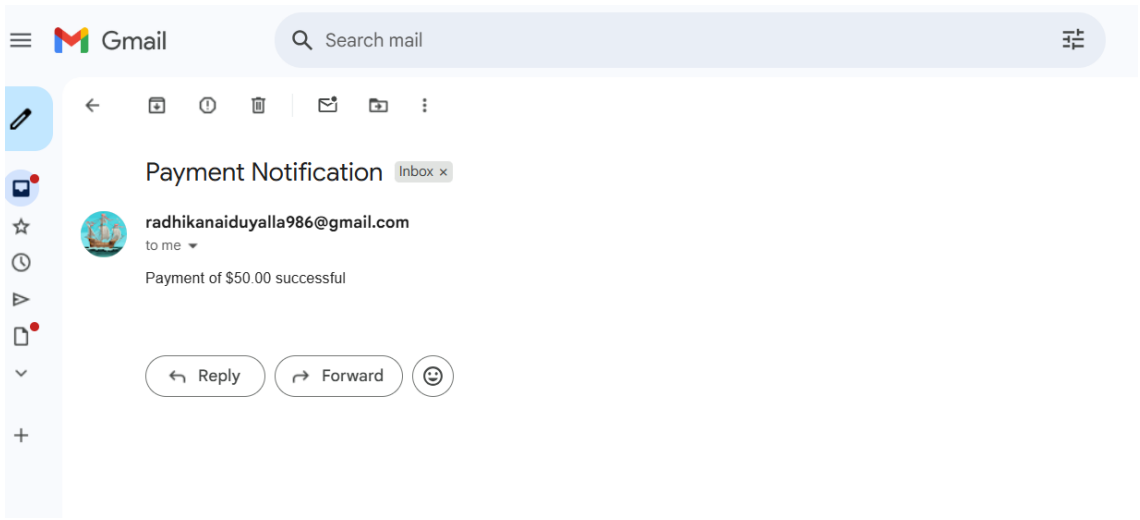

Swagger UI
localhost:8083/swagger-ui/index.html
Import favorites
Gmail
YouTube
Booking.com
Maps
Maven Repository:...
Techademy One
Other favorites
InPrivate
Swagger
/v3/api-docs
Explore
OpenAPI definition
v0
OAS 3.0
/v3/api-docs
Servers
http://localhost:8083 - Generated server url
audit-log-controller
POST /audits/log
GET /audits/getAll
Schemas
AuditLog


Swagger UI
localhost:8089/swagger-ui/index.html
Import favorites
Gmail
YouTube
Booking.com
Maps
Maven Repository:...
Techademy One
Other favorites
InPrivate
Servers
http://localhost:8089 - Generated server url
auth-controller
POST /auth/signup
POST /auth/login
Schemas
SignupRequest
AuthenticationRequest
AuthenticationResponse


localhost:9090/query?g0.expr=http_client_requests_active_seconds_max%7Bclient_name%3D"localhost"%2Cexception%3D"none"%2Cmethod%3D"GET"%2C...
Import favorites
Gmail
YouTube
Booking.com
Maps
Maven Repository:...
Techademy One
Other favorites
InPrivate
Prometheus
Query
Alerts
Status
http_client_requests_active_seconds_max{client_name="localhost",exception="none",method="GET",outcome="UNKNOWN",status="CLIENT_ERROR",uri="none"}
Execute
Table
Graph
Explain
1h
End time
Medium res.
Unstacked
Stacked
100.00
90.00
80.00
70.00
60.00
50.00
40.00
30.00







OVERVIEW:

The Banking Web Application project demonstrates a robust, microservicesbased architecture built with Angular and Spring Boot. Core functionalities, including

customer management, accounts, payments, feedback, audit, and notifications, are seamlessly integrated, with Kafka enabling asynchronous communication between Payment and Notification services and Feign Clients facilitating synchronous inter-service calls. Supported by API Gateway, Config Server, and Discovery Service, the system is secure, scalable, and maintainable. This project highlights modern enterprise-level development practices and provides a solid foundation for real-world banking solutions.

Acknowledgment / Conclusion:

This project has been completed successfully under the guidance of Ramakrishna Sir. I have applied my knowledge of Angular, Spring Boot, and Microservices to develop a scalable and secure Banking Web Application. I hope this project meets the objectives and expectations outlined at the start.

Submitted By:

Radhika yalla

BATCH _II

radhikanaiduyalla986@gmail.com

ANGULAR