Banking Microservices Application

Capstone Project

Name: Usha Mora
Email: ushamora03@gmail.com

Table of Contents

Abstract

This project, **Banking Microservices Application**, is a capstone project that demonstrates the power of **microservices architecture** in delivering scalable, secure, and modular solutions. It simulates real-world banking functions like account management, customer services, payments, and authentication, all built as independent services. By using microservices, the system achieves better **fault isolation, independent deployment, and scalability**. The frontend, developed in Angular 17, interacts with backend REST APIs, ensuring a seamless and modern user experience. This architecture is designed to align with real-world banking system requirements, including **security, modularity, and extensibility**.

Technologies Used

Frontend: Angular 17, TypeScript, Angular Material, Bootstrap

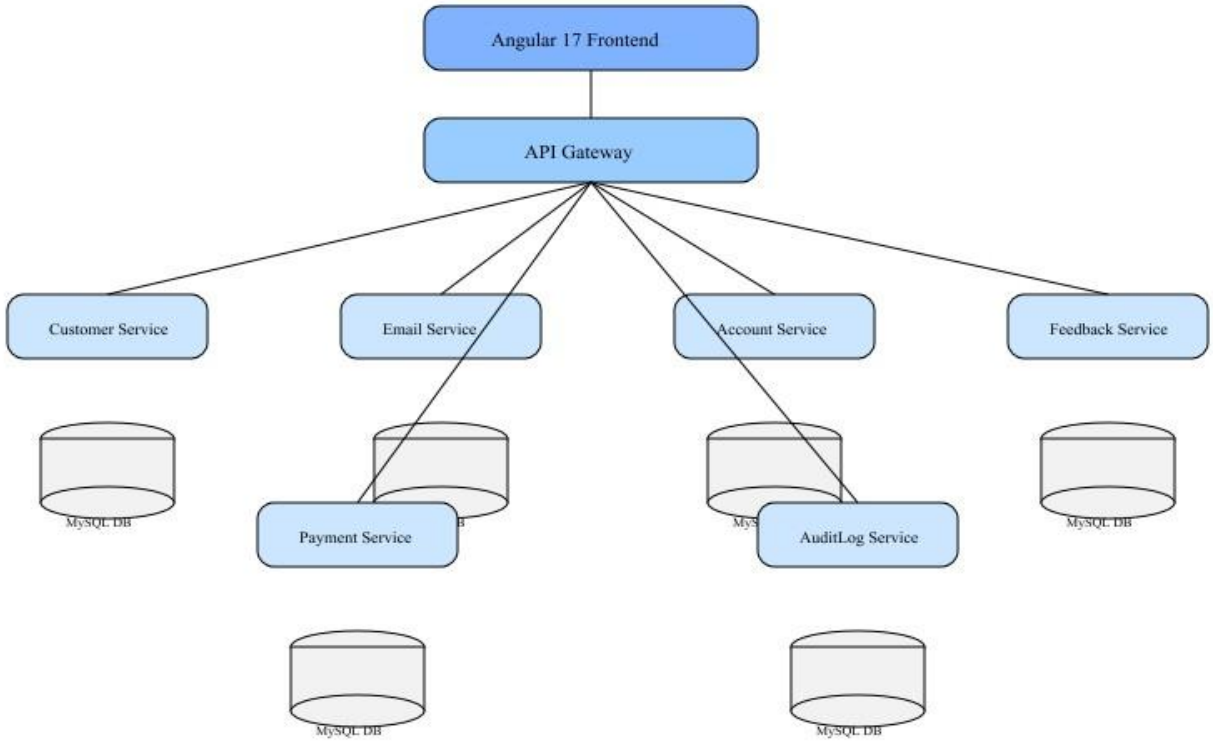
Backend: Spring Boot, Spring Cloud (Eureka, Gateway), Spring Security with JWT, REST APIs

Database: MySQL (per service)

Tools & Build: Maven, npm, Postman

Version Control: Git/GitHub

Architecture Diagram



Process Flow

1. User accesses the Angular frontend.
2. Requests are routed through API Gateway.
3. Gateway forwards requests to respective microservices (Accounts, Customers, Payments, Authentication).
4. Services interact with their own MySQL databases.
5. Responses are aggregated and returned to the frontend.
6. Authentication is managed via Spring Security and JWT.

Execution Steps

Backend:

1. Navigate to the backend folder.
2. Run: `mvn clean install` 3. Start services: `mvn spring-boot:run`

Frontend:

1. Navigate to the frontend folder.
2. Run: `npm install`
3. Start server: `ng serve --open`

Future Enhancements

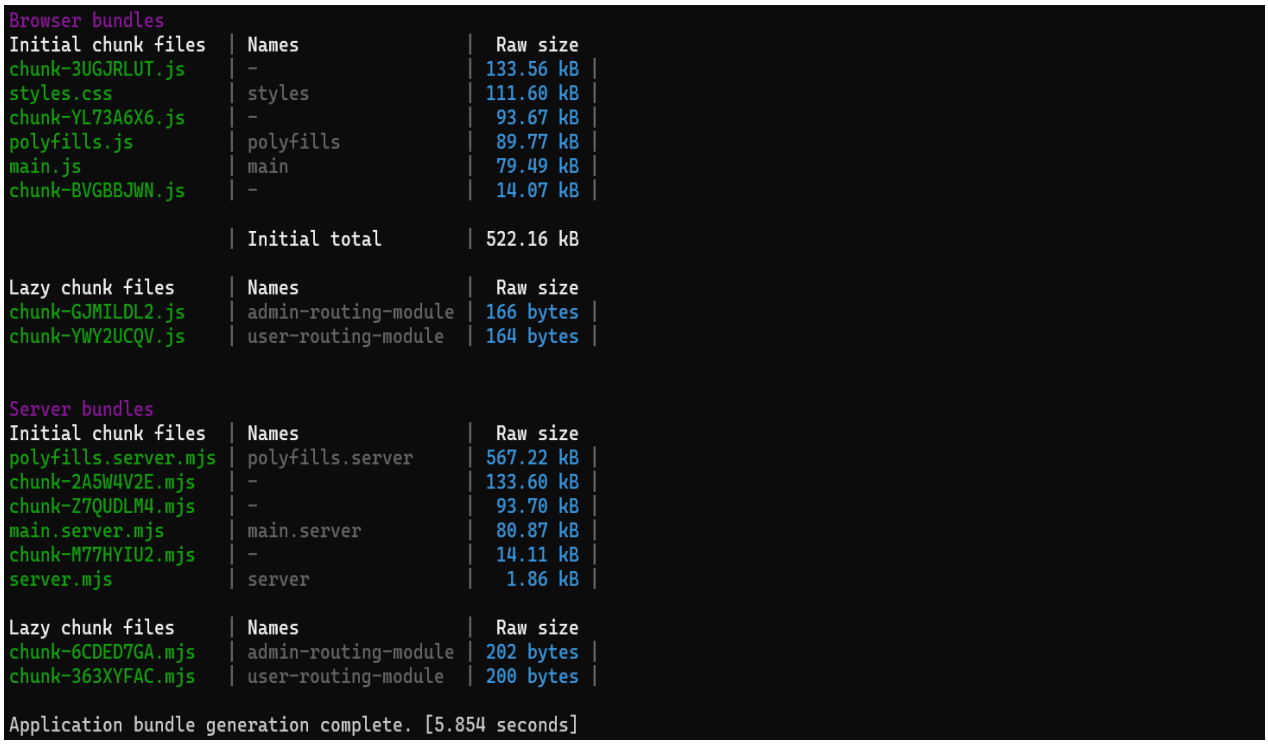
1. **Loan Service**: Extend functionality to support loan management (applications, approvals, repayments).
2. **Reporting & Analytics**: Add dashboards and analytical tools for customer and transaction insights.
3. **Cloud Deployment**: Deploy services using Docker & Kubernetes on AWS/Azure for scalability.
4. **Mobile App Integration**: Provide native/hybrid mobile apps for better customer accessibility.
5. **Advanced Security**: Implement Multi-Factor Authentication (MFA) and OAuth2 support.

Conclusion

The Banking Microservices Application demonstrates a modern approach to building enterprise-grade systems. By leveraging microservices architecture, the solution achieves modularity, scalability, and robust security. The use of Angular 17 ensures a responsive and user-friendly interface, while Spring Boot and Spring Cloud enable reliable backend services. This project provides a foundation that can be extended with additional features like **loans, reporting, and cloud deployment** for future enhancements.

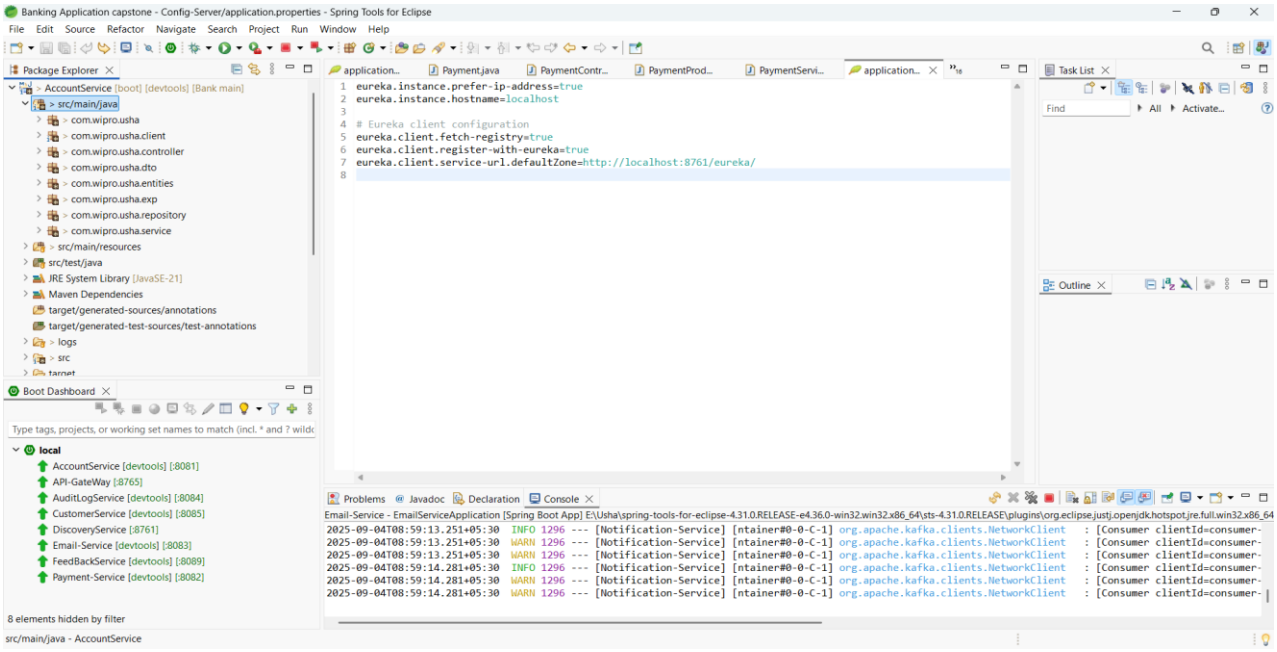
Output Screenshots

1. Front-end Execution

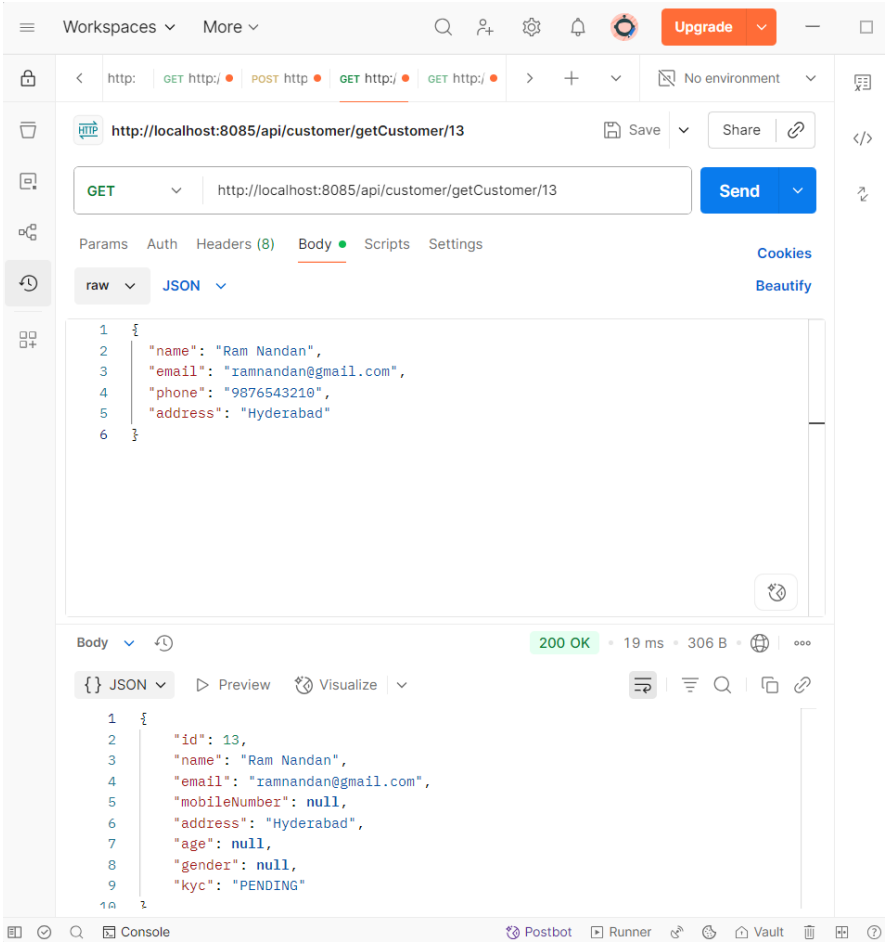


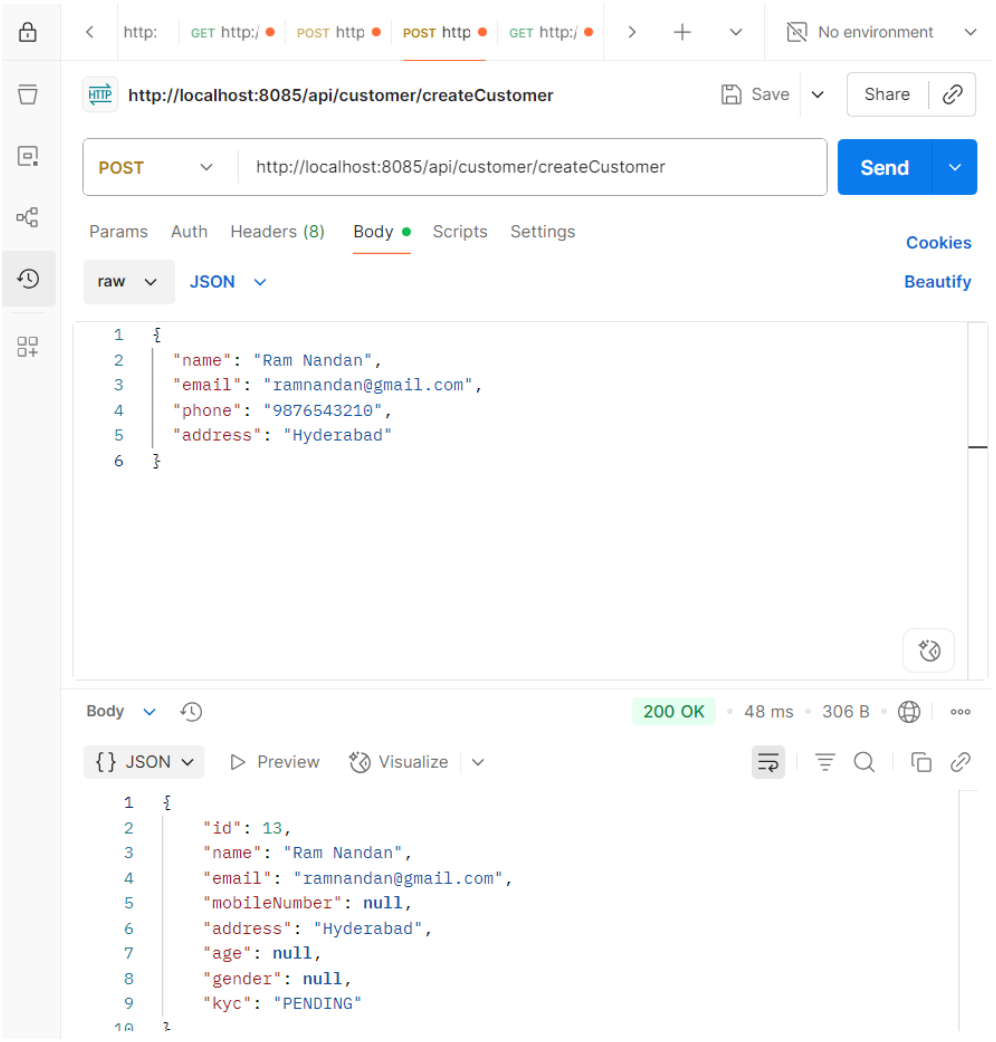
```
Browser bundles
Initial chunk files   | Names                |  Raw size
chunk-3UGJRLUT.js     | -                    | 133.56 kB |
styles.css            | styles               | 111.60 kB |
chunk-YL73A6X6.js     | -                    |  93.67 kB |
polyfills.js          | polyfills            |  89.77 kB |
main.js               | main                 |  79.49 kB |
chunk-BVGBBJWN.js     | -                    |  14.07 kB |

                      | Initial total        | 522.16 kB

Lazy chunk files      | Names                |  Raw size
chunk-GJMILDL2.js     | admin-routing-module | 166 bytes |
chunk-YWY2UCQV.js     | user-routing-module  | 164 bytes |


Server bundles
Initial chunk files   | Names                |  Raw size
polyfills.server.mjs  | polyfills.server     | 567.22 kB |
chunk-2A5W4V2E.mjs    | -                    | 133.60 kB |
chunk-Z7QUDLM4.mjs    | -                    |  93.70 kB |
main.server.mjs       | main.server          |  80.87 kB |
chunk-M77HYIU2.mjs    | -                    |  14.11 kB |
server.mjs            | server               |   1.86 kB |

Lazy chunk files      | Names                |  Raw size
chunk-6CDED7GA.mjs    | admin-routing-module | 202 bytes |
chunk-363XYFAC.mjs    | user-routing-module  | 200 bytes |

Application bundle generation complete. [5.854 seconds]
```

2. Backend Execution

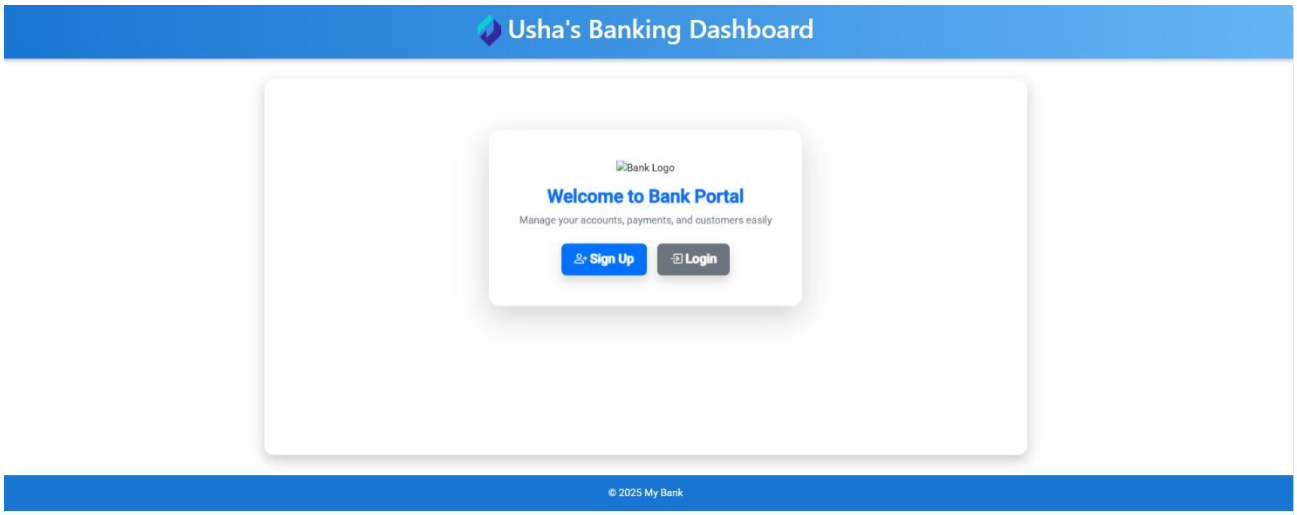


3. POSTMAN

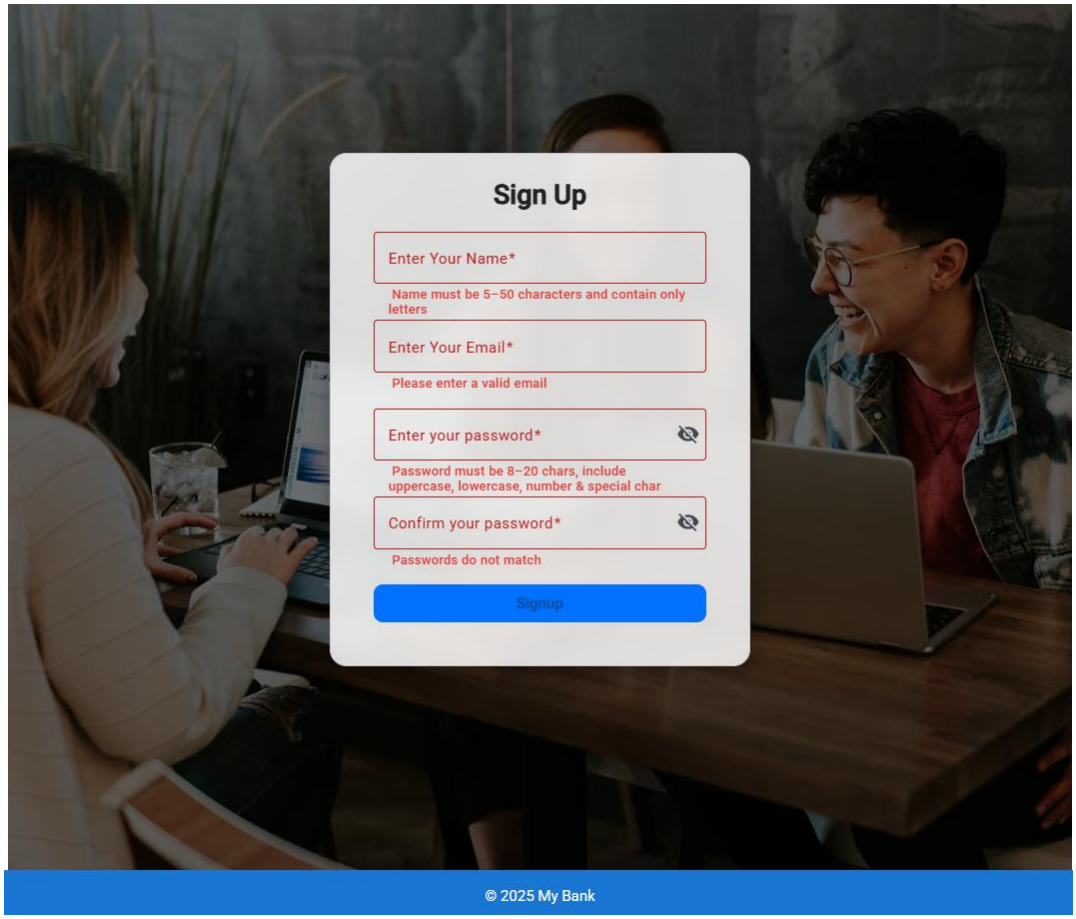

Workspaces
More
Upgrade
No environment
http://localhost:8085/api/customer/getCustomer/13
Save
Share
GET
http://localhost:8085/api/customer/getCustomer/13
Send
Params
Auth
Headers (8)
Body
Scripts
Settings
Cookies
raw
JSON
Beautify
{
"name": "Ram Nandan",
"email": "ramnandan@gmail.com",
"phone": "9876543210",
"address": "Hyderabad"
}
Body
200 OK
19 ms
306 B
JSON
Preview
Visualize
{
"id": 13,
"name": "Ram Nandan",
"email": "ramnandan@gmail.com",
"mobileNumber": null,
"address": "Hyderabad",
"age": null,
"gender": null,
"kyc": "PENDING"
}
Console
Postbot
Runner
Vault


http:
GET http:/
POST http
POST http
GET http:/
No environment
http://localhost:8085/api/customer/createCustomer
Save
Share
POST
http://localhost:8085/api/customer/createCustomer
Send
Params
Auth
Headers (8)
Body
Scripts
Settings
Cookies
raw
JSON
Beautify
{
"name": "Ram Nandan",
"email": "ramnandan@gmail.com",
"phone": "9876543210",
"address": "Hyderabad"
}
Body
200 OK
48 ms
306 B
{} JSON
Preview
Visualize
{
"id": 13,
"name": "Ram Nandan",
"email": "ramnandan@gmail.com",
"mobileNumber": null,
"address": "Hyderabad",
"age": null,
"gender": null,
"kyc": "PENDING"
}

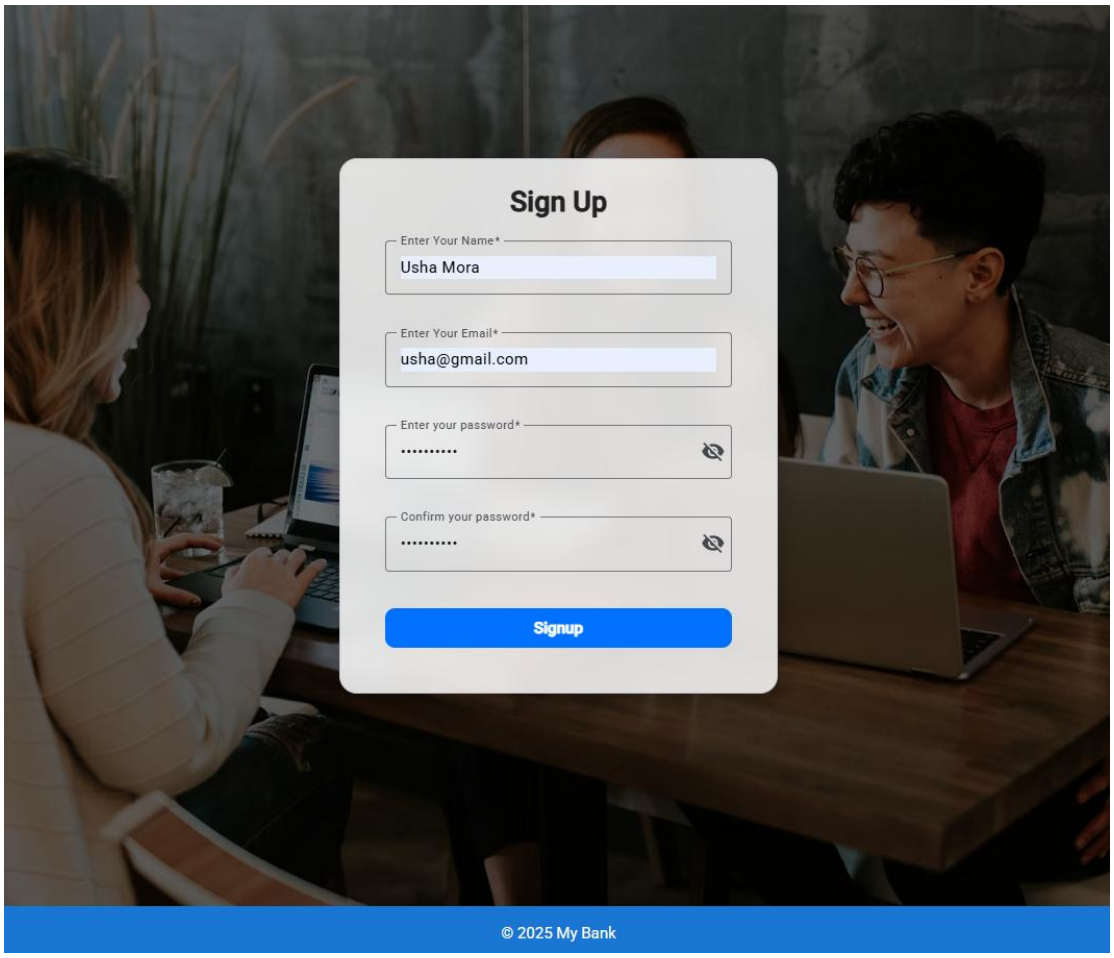
4. Homepage


Usha's Banking Dashboard
Bank Logo
Welcome to Bank Portal
Manage your accounts, payments, and customers easily
Sign Up
Login
© 2025 My Bank

5. Signup page validation



6. Signup Page



7. Login Page