



**ECOLE MAROCAINE DES
SCIENCES DE L'INGENIEUR**
Membre de
HONORIS UNITED UNIVERSITIES

RAPPORT



CONTIFY
GESTION DES CONTACTS



CONTIFY

Gestion Des Contacts

Membres

- Mohamed Menfalouti
- Mohamed Ismail Menhal

Plan

1. Introduction
2. Architecture
3. Conteneurisation : Docker
4. CI/CD Jenkins
5. Ngrok
6. SonarQube
7. Application Screenshots
8. Conclusion



Introduction

Aperçu du projet

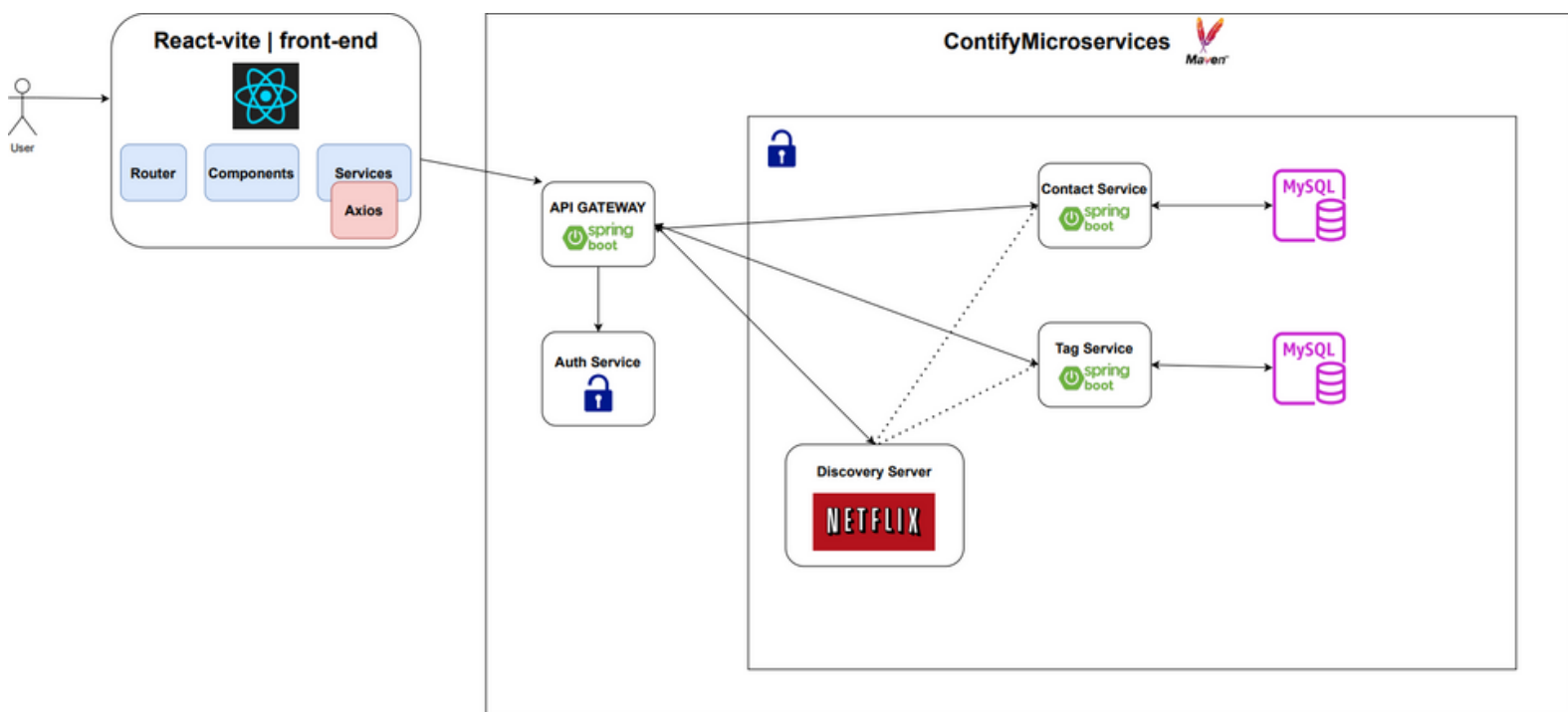
Contify représente une solution complète pour la gestion et l'organisation efficaces de vos contacts. Adoptant une architecture microservices, ce projet vise à offrir une modularité et une évolutivité accrues pour répondre aux besoins dynamiques de la gestion des contacts.

Importance de l'architecture microservices

Nous avons opté pour l'architecture microservices en raison de ses nombreux bénéfices. Grâce à elle, chaque service peut être géré séparément, ce qui améliore la capacité à monter en charge, la robustesse et la facilité de maintenance. De plus, cette architecture favorise une innovation permanente, puisque chaque service peut être conçu, mis en place et amélioré de façon autonome.

Architecture

Le projet est structuré autour de plusieurs services interconnectés, chacun jouant un rôle spécifique et complémentaire dans l'écosystème global de Contify.



Description des services

- **Api Gateway Service** : En tant que point d'entrée, il gère le routage, l'authentification et d'autres aspects transversaux, fournissant une interface unifiée pour les clients externes.
- **Discovery Server** : Essentiel pour la découverte et l'enregistrement des services, il établit un environnement hautement scalable et tolérant aux pannes en facilitant la communication dynamique entre les services.

Description des services

- **Contact Service** : Au cœur de la gestion des contacts, ce service offre une gamme complète de fonctionnalités, de la manipulation des entités à la communication transparente avec la base de données.
- **Tag Service** : Axé sur les opérations liées aux tags, il offre une solution modulaire et dédiée à la gestion efficace des tags.

Description des services

- **Auth Service** : Gérant les aspects cruciaux de l'authentification et de l'autorisation de manière sécurisée, il assure la protection des données sensibles.

Mécanismes de communication

































Eureka, notre mécanisme de choix pour la découverte et l'enregistrement des services, favorise une communication dynamique entre les différentes parties du système, garantissant une extensibilité sans heurts.

OpenFein pour l'affectation des tags aux contacts.

Conteneurisation : Docker

Implémentation et avantages

La conteneurisation avec Docker apporte une valeur ajoutée significative au projet. En encapsulant chaque service dans un conteneur, nous assurons une portabilité, une isolation et une facilité de déploiement accrues. Les conteneurs offrent également un environnement cohérent, éliminant les problèmes potentiels liés aux différences d'environnement entre les étapes de développement et de production.

<input type="checkbox"/>	 contify		Running (7/7)	55 minutes ago	N/A			
<input type="checkbox"/>	 mysql-container	mysql:latest	Running	3306:3306 	1 hour ago	N/A		
<input type="checkbox"/>	 phpmyadmin-1	phpmyadmin/phpmyadmin	Running	8081:80 	1 hour ago	N/A		
<input type="checkbox"/>	 discovery-server-1	contify-discovery-server	Running	8761:8761 	55 minutes ago	N/A		
<input type="checkbox"/>	 api-gateway-1	contify-api-gateway	Running	8888:8888 	55 minutes ago	N/A		
<input type="checkbox"/>	 contact-service-1	contify-contact-service	Running	8088:8088 	55 minutes ago	N/A		
<input type="checkbox"/>	 auth-service-1	contify-auth-service	Running	8087:8087 	55 minutes ago	N/A		
<input type="checkbox"/>	 tag-service-1	contify-tag-service	Running	8089:8089 	55 minutes ago	N/A		

CI/CD avec Jenkins

Processus et configuration

Jenkins, notre outil d'intégration continue, automatise le processus de construction, de test et de déploiement. Cela garantit une livraison rapide et fiable des nouvelles fonctionnalités et corrections, tout en minimisant les erreurs liées au déploiement manuel. La configuration précise de Jenkins assure une harmonie entre les différentes étapes du processus CI/CD.



Stage View

		Checkout	Build	Create Docker Image	Deploy	report in sonarqube
Average stage times: (Average <u>full</u> run time: ~1min 43s)		3s	21s	11s	1min 33s	1min 7s
#56	Jan 19 13:07 2 commits	4s	18s	12s	36s	1min 23s
#54	Jan 19 24:41 1 commit	4s	20s	12s	20s	53s
#51	Jan 19 24:22 No Changes	4s	18s	12s	21s	1min 5s

Pipeline Script

```
pipeline {
  agent any

  stages {
    stage('Checkout') {
      steps {
        script {
          git branch: 'master', url: 'https://github.com/MeedMn/contifyMicroservices.git'
        }
      }
    }

    stage('Build') {
      steps {
        bat 'mvn clean install'
      }
    }

    stage('Create Docker Image') {
      steps {
        bat 'docker-compose build'
      }
    }

    stage('Deploy') {
      steps {
        script {
          bat 'docker-compose up -d'
          sleep 15
        }
      }
    }

    stage('report in sonarqube') {
      steps {
        bat "mvn clean verify sonar:sonar -Dsonar.projectKey=contify -Dsonar.projectName='contify' -Dsonar.token=sqp_2f122f32e6d77efcf6569c20f10148f3069c4848"
      }
    }
  }
}
```

Définition de l'Agent :

agent any : Ce pipeline peut être exécuté sur n'importe quel agent disponible (ouvrier Jenkins).

Stages:

Le pipeline comporte plusieurs stages, chacun représentant une phase dans le processus de construction et de déploiement.

Stage(Checkout) :

Cette étape récupère le code source depuis le référentiel GitHub spécifié. Alors qu'elle va cloner le repo depuis l'URL donnée et récupère la branche 'master'.

Stage(Build) :

Cette étape construit le projet en utilisant Maven.

- **bat 'mvn clean install'** : Exécute une commande batch Windows pour exécuter Maven, nettoyer le projet, puis le construire.

Stage(Create docker image) :

Cette étape crée une image Docker en utilisant le fichier Docker Compose.

- **bat 'docker-compose build'** : pour construire les images Docker.

Stage(Deploy) :

Cette étape déploie l'application en utilisant Docker Compose.

Stage(Report to SonarQube):

Cette étape génère un rapport d'analyse de code en utilisant SonarQube.

Ngrok & WebHooks

Utilisation de Ngrok

L'utilisation de Ngrok assure un fonctionnement en temps réel du système, permettant une expérience utilisateur transparente.

Webhooks / Manage webhook

Settings Recent Deliveries

We'll send a POST request to the URL below with details of any subscribed events. You can also specify which data format you'd like to receive (JSON, x-www-form-urlencoded, etc). More information can be found in [our developer documentation](#).

Payload URL *

`https://12d2-105-158-237-0.ngrok-free.app/github-webhook/`

Content type

`application/json`

Secret

```
Region      Europe (eu)
Latency     65ms
Web Interface http://127.0.0.1:4040
Forwarding  https://12d2-105-158-237-0.ngrok-free.app -> http://localhost:8080

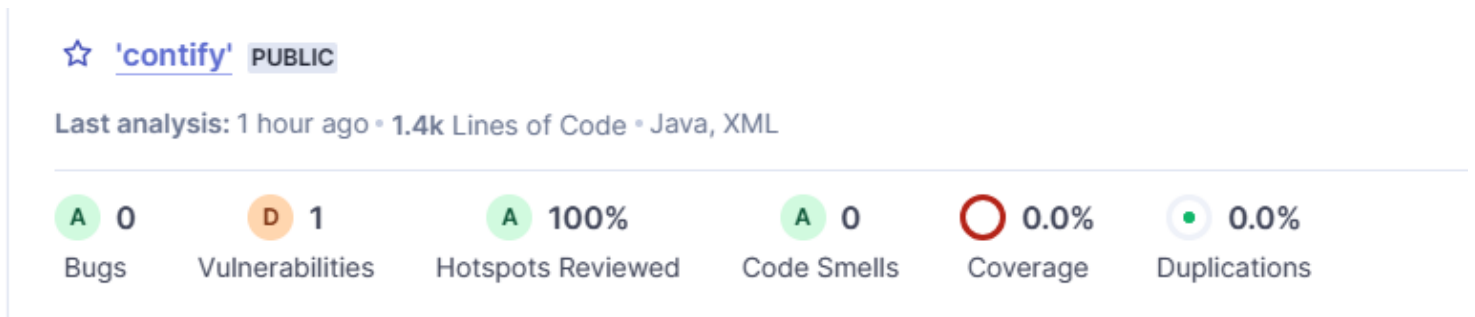
Connections      ttl    opn    rt1    rt5    p50    p90
                  246    1      0.00   0.00   0.31   35.23

HTTP Requests
-----
GET /job/contify/wfapi/runs      200 OK
GET /job/contify/wfapi/runs      200 OK
GET /job/contify/wfapi/runs      200 OK
GET /job/contify/wfapi/runs      200 OK
GET /job/contify/wfapi/runs      200 OK
GET /job/contify/wfapi/runs      200 OK
GET /job/contify/wfapi/runs      200 OK
GET /job/contify/buildHistory/ajax 200 OK
GET /job/contify/wfapi/runs      200 OK
GET /job/contify/wfapi/runs      200 OK
```

SonarQube

Configuration et bénéfices pour la qualité du code

SonarQube, intégré dans notre pipeline, analyse le code à chaque commit. En fournissant des informations détaillées sur la qualité du code, les vulnérabilités et les erreurs potentielles, il renforce notre engagement envers un code propre, sécurisé et facilement maintenable.



Application Screenshots

Landing Page

CONTIFY

SignUp

Login

The Ultimate Contact Management Solution

Sign up now and get started for free!

[Get Started Now !](#)



CONTIFY © 2023 . All rights reserved.

Sign Up Page

CONTIFY

SignUp

Login

CONTIFY

Sign up & Get started!

Full Name

Address

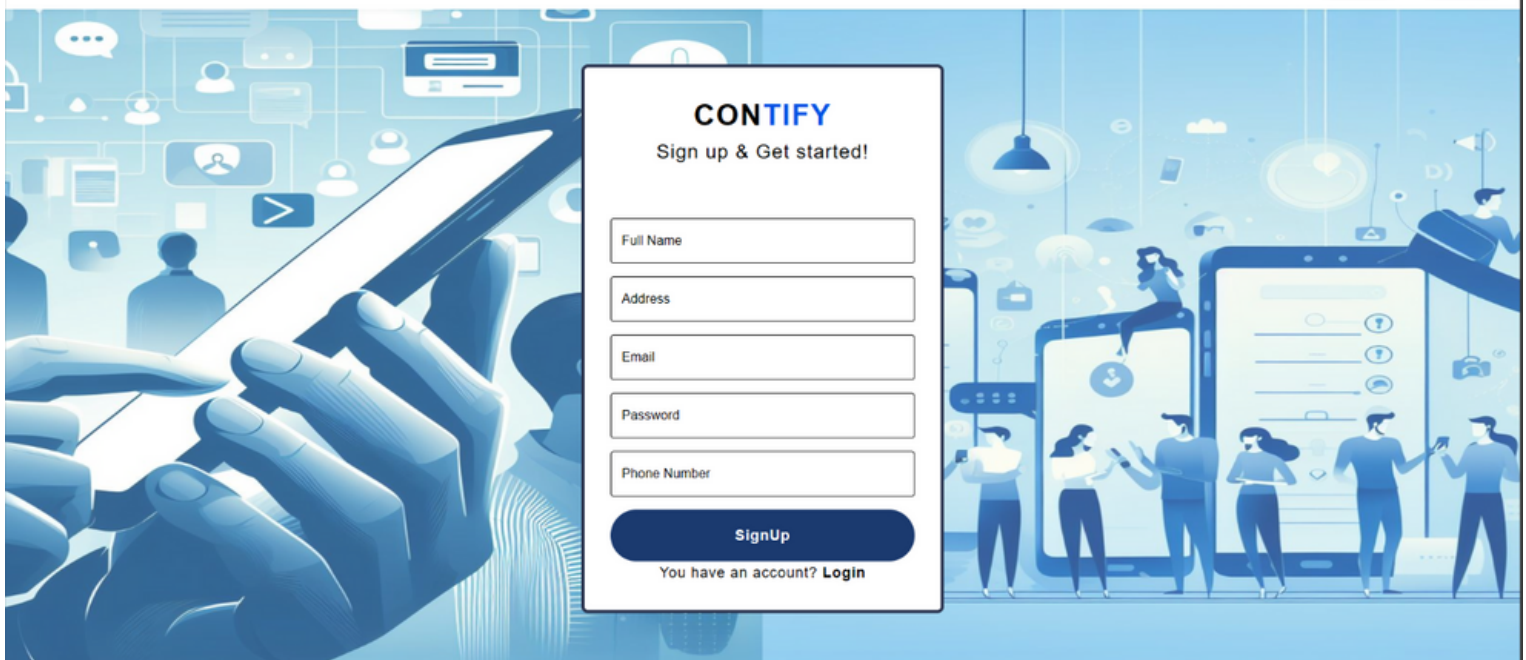
Email

Password

Phone Number

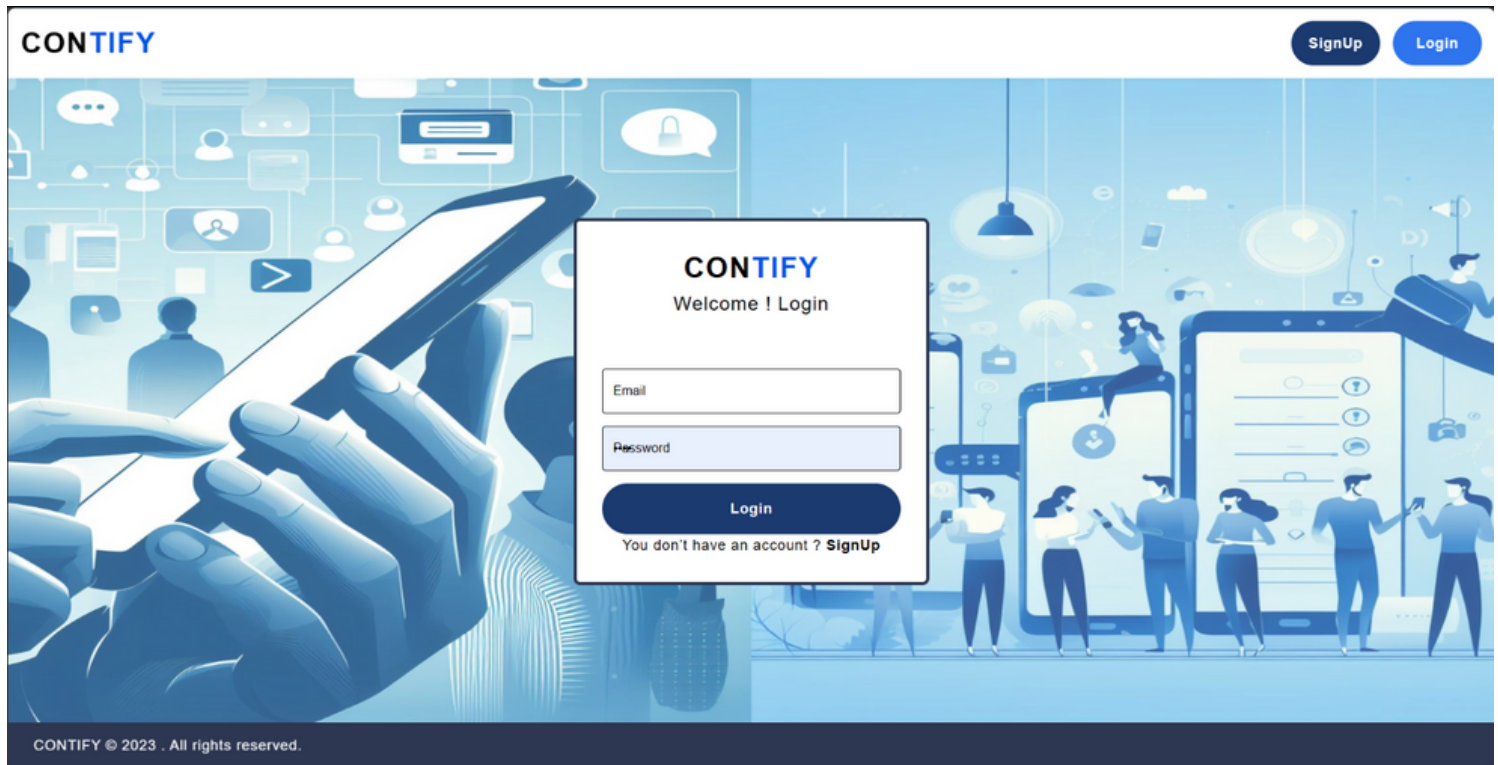
[SignUp](#)

You have an account? [Login](#)

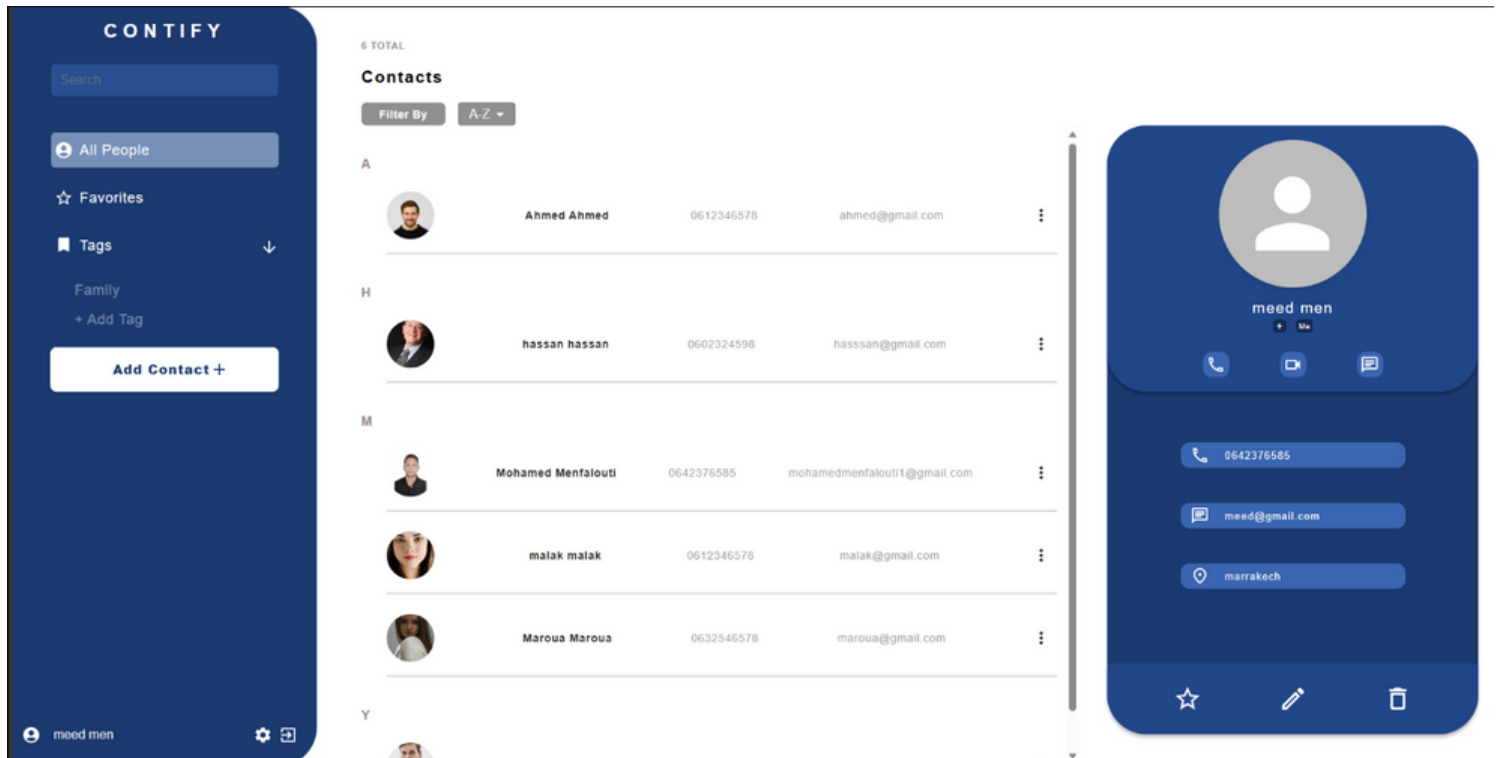


CONTIFY © 2023 . All rights reserved.

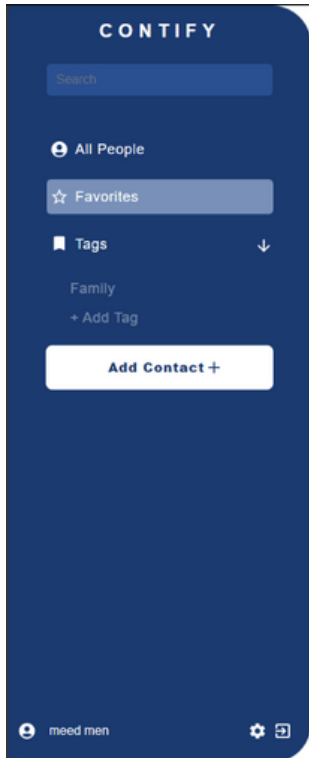
Login Page



All People Page



Favorites Page

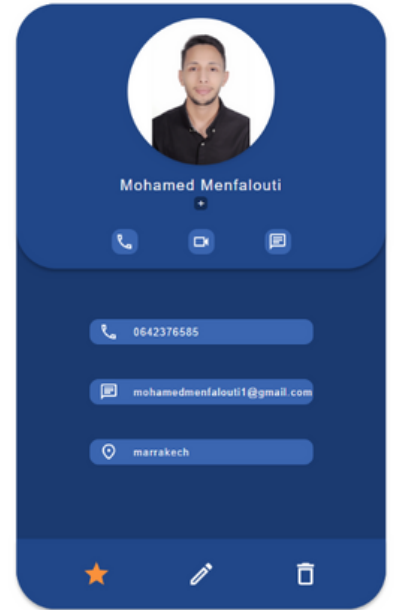
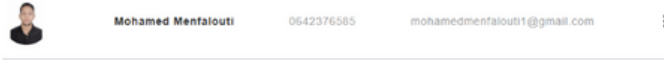


1 TOTAL

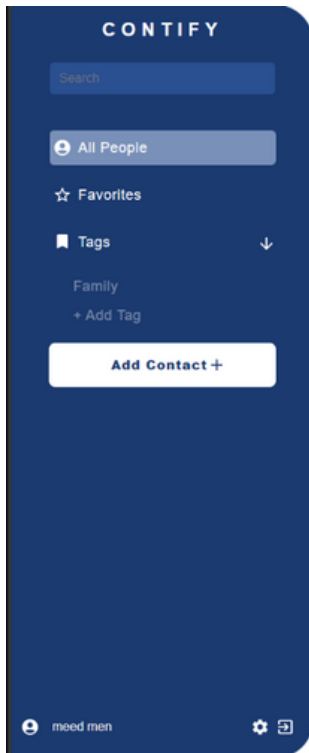
Contacts

Filter By A-Z

M



Tag Page



2 TOTAL

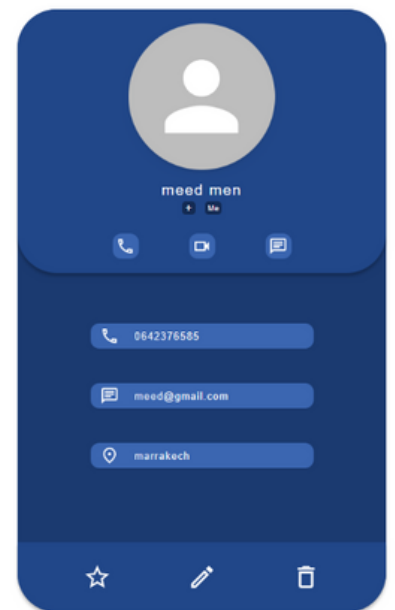
Contacts - Family

Filter By A-Z

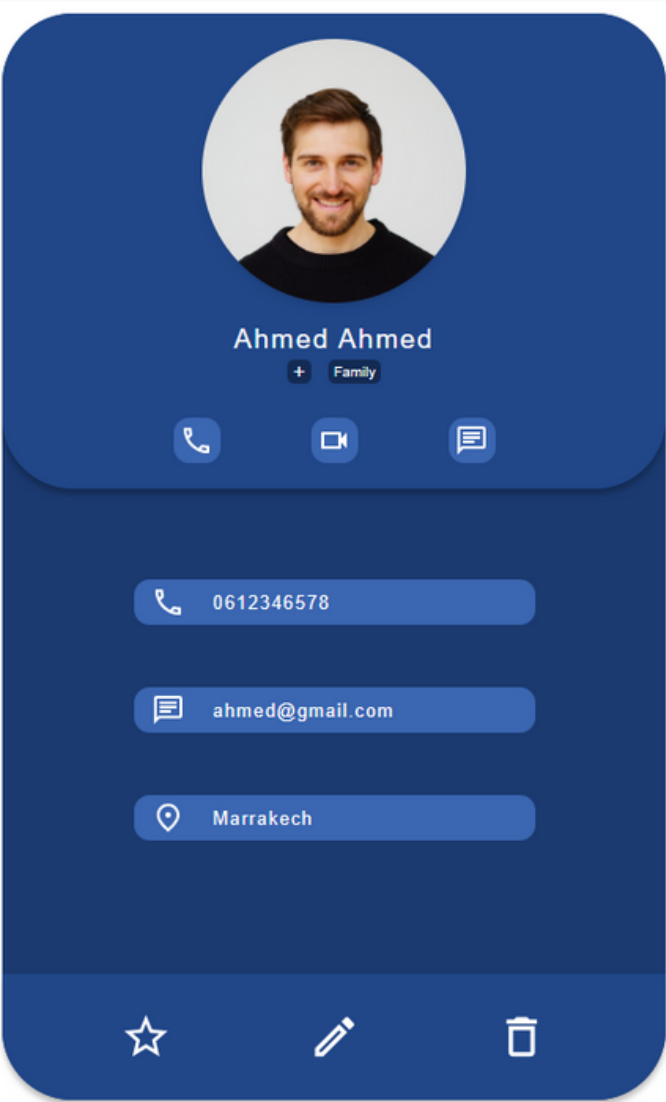
A



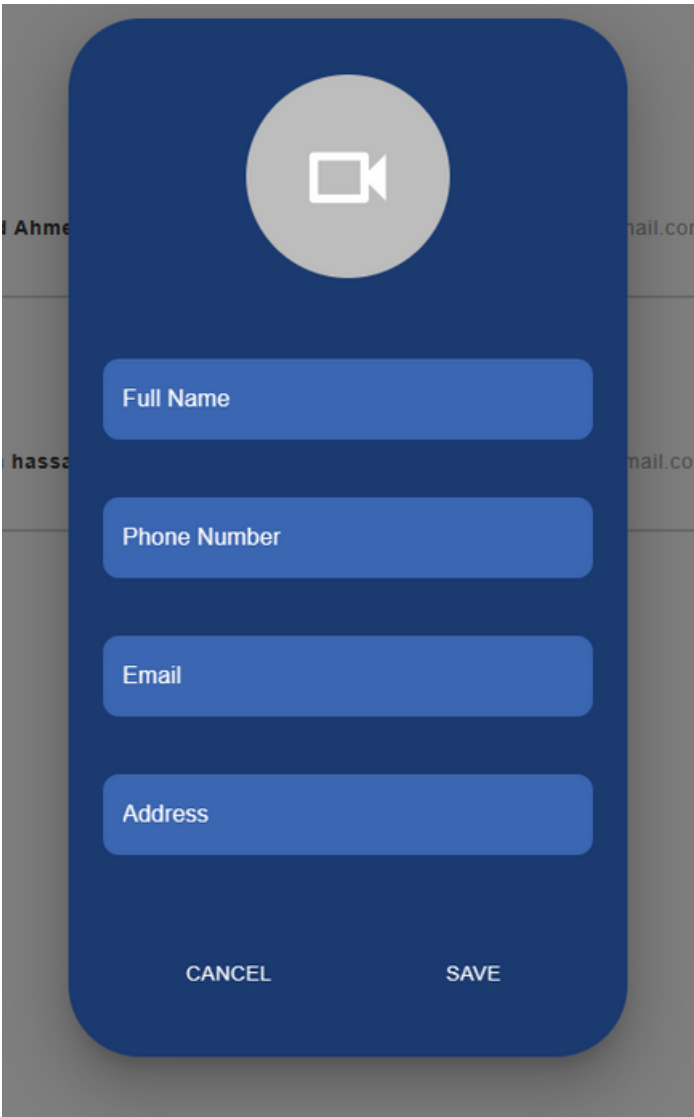
H



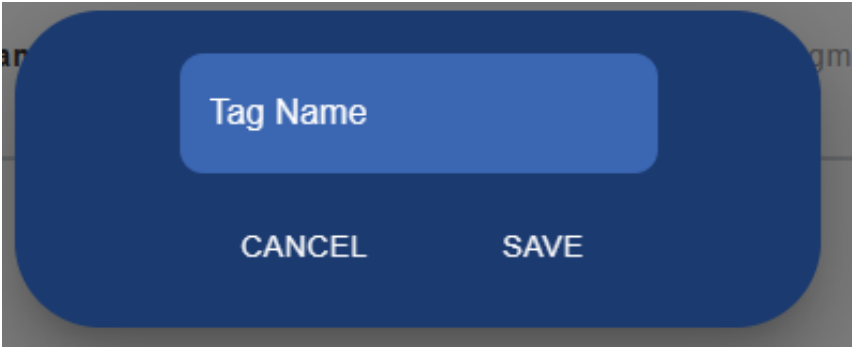
Contact Infos



Add Contact



Add Tag



Conclusion

Résumé des accomplissements

Contify offre une solution complète et moderne pour la gestion de contacts.

L'architecture microservices, la conteneurisation, l'intégration continue et l'analyse de code automatique contribuent à la robustesse et à la qualité du système.

Perspectives futures

Le projet est prêt pour des évolutions futures. Des ajouts de fonctionnalités, des optimisations de performance et l'exploration de nouvelles technologies sont envisageables pour répondre aux besoins changeants de la gestion de contacts.

Github Repos

Front-End :

<https://github.com/Ismailmenhal/contify-frontend>

back-End :

<https://github.com/MeedMn/contifyMicroservices>