**Partner Management CRUD Application**

Partner management CRUD application is built on Spring-boot and MongoDB (No SQL) technology for client management. An end user can create a new partner or client profile to this system. User can update, delete and look for a user profile. It is built on AngularJS framework on UI as single-paged application.

**Technology stack**

- Java

- Spring-boot with embedded Tomcat server in it

- AngularJS/Javascript

- Maven for Continuous Integration

- MongoDB No SQL

- Docker and Docker Compose

- Junit and Mockito framework for unit and integration testing

- Spring Tool Suite (STS) IDE for development

**Resource URL**

Application Home Page

http://localhost:8080/home

To search for all partners (GET)

http://localhost:8080/partners/

To search for a specific partner by id (GET)

http://localhost:8080/partners/1

To create a new partner (POST)

http://localhost:8080/partners/

**Payload Example:**

{

"id": 1,

"partnerName": "Jesi",

"address": "NY",

"email": "Jesi@nbc.com"

}

To update an existing partner (PUT)

http://localhost:8080/partners/1

**Payload Example:**

{

"id": 1,

"partnerName": "Jesi",

"address": "NY",

"email": "JesiRay@nbc.com"

}

To delete a specific partner by id (DELETE)

http://localhost:8080/partners/2

**Source Code Location**

https://github.com/dinJavaDev/partnerManagement.git

**https://github.com/dinJavaDev/partnerManagement.git**

**Docker Image Location**

This is being pulled by docker compose internally, so not required to pull separately.

docker pull dinjavadev/spring-boot

**Application Deployment**

You need to pull the code from the above mentioned repository. You need to have Docker engine and Docker compose should be running in your local server or with maven.

**Run as a development mode:**

You need to have Java, STS IDE, Maven, Git and MongoDB should be installed and running in your machine.

- Checkout the code

- mvn clean install (from project root directory)

- java -jar partnerManagementApp-1.0.jar

- Please use the home URL (http://localhost:8080/home) to access

**Run as a containerized mode:**

- Checkout the code

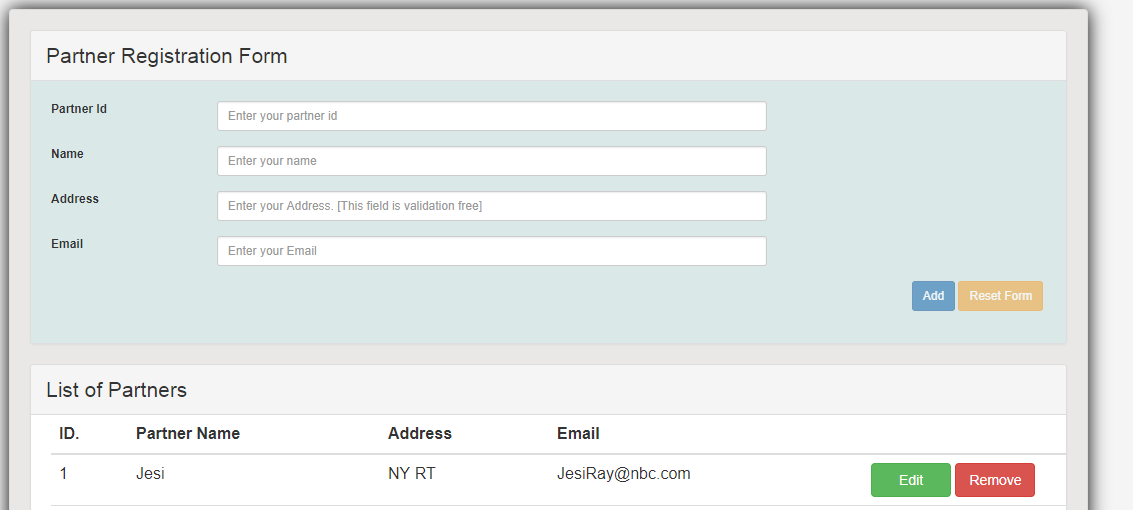
- docker-compose up -d (from project root directory)

- Please use the home URL (http://localhost:8080/home) to access

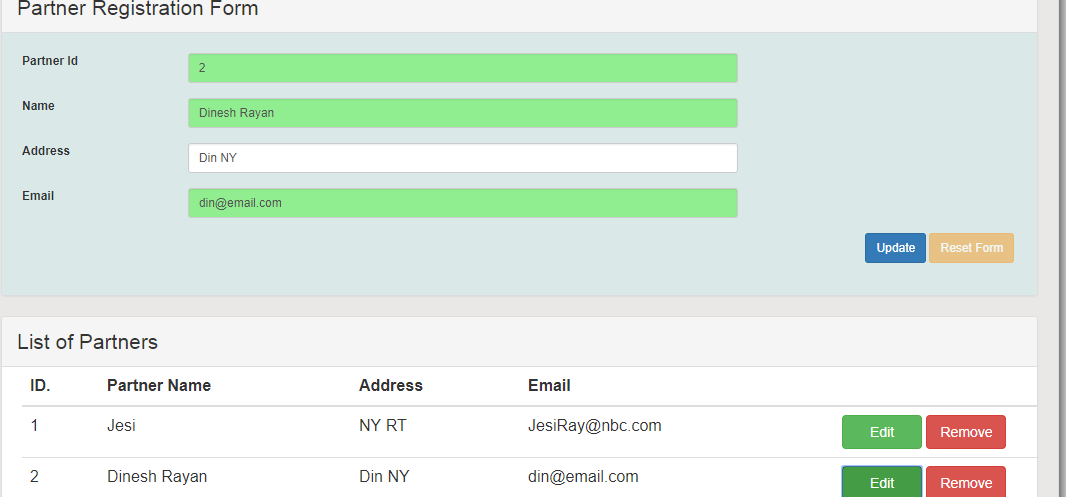
You can also use **Amazon EC2 ECS** for hosting this container there.

**SCREENS**:

Create Option:



Partner Edit/Update:



With MongoDB Data aside:

