**Functional requirement**

* User can sign up using email and phone(must) = **DONE**
* User can log in = **DONE**
* User can manage his/her profile = **DONE**
* User can add his/her story = **DONE**
* User can search = **DONE**
* User can hide his/her identity = **DONE**
* User can read other people stories = **DONE**
* User can send message to other users = **DONE**
* Admin can delete users = **DONE**
* User can add missing person form = **DONE**
* User can report inappropriate behavior of other user

**To Do**

* Calculate the number of years a person is imprisoned = **DONE**
* Add all possible prisons as Enum
* Show a statical report of number of prisoners in each prison(graphical if possible)= **DONE**
* Add all prisoners(G-15) and from sakitsm as reference
* Show Eritrean constitution in dashboard

**Nonfunctional requirement**

* Secure
* Scalable
* Encrypted

**Technologies to use**

1. **Back end**

* **Spring boot/ java**
* **RDS(MySQl)**

1. **Front end**

* **React native**
* **Bootstrap**

1. **Cloud tech(aws)**

* **Cognito**
* **Lambda**
* **S3**
* **CloudWatch**
* **Route 53**
* **Cache**
* **SNS**
* **SQS**

1. **Repository and CI/CD pipelines**

* **Git**
* **Jenkins**
* **Jira**

1. **Containerization**

* **Docker**
* **Kubernetes**

spring.application.name=Prison-Victims  
spring.datasource.url=jdbc:mysql://database-1.c2hmeuqo6il0.us-east-1.rds.amazonaws.com:3306/Eritrean\_Prison\_Victims\_DB  
spring.datasource.username=admin  
spring.datasource.password=Fvalcon1728$  
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver  
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQLDialect  
spring.jpa.show-sql=false  
spring.jpa.hibernate.ddl-auto=update  
server.port=8181  
server.ssl.enabled=false  
spring.security.oauth2.client.registration.cognito.client-id=1truinrsmh6jhu1e6fqemu6hh3  
spring.security.oauth2.client.registration.cognito.client-secret=5le35qogi70v9c1a0mrokcriiloes2k2n274uuv7lsupgkbh9lu  
spring.security.oauth2.client.registration.cognito.redirect-uri=http://localhost:8181/login/oauth2/code/cognito  
spring.security.oauth2.client.registration.cognito.scope=openid,email,phone,profile  
spring.security.oauth2.client.provider.cognito.issuer-uri=https://cognito-idp.us-east-1.amazonaws.com/us-east-1\_2drt5vFAs  
logging.level.org.springframework.security.oauth2.client=TRACE  
logging.level.org.springframework.security.oauth2.provider=TRACE  
spring.security.oauth2.client.provider.cognito.user-name-attribute=email  
spring.thymeleaf.cache=false  
spring.thymeleaf.mode=HTML  
spring.web.resources.chain.cache=false  
spring.web.resources.cache.period=0  
spring.web.resources.static-locations=classpath:/static/  
spring.kafka.bootstrap-servers=localhost:9092  
spring.kafka.consumer.group-id=admas-chat  
spring.kafka.consumer.value-deserializer=org.springframework.kafka.support.serializer.JsonDeserializer  
spring.kafka.consumer.properties.spring.json.trusted.packages=\*  
spring.kafka.producer.value-serializer=org.springframework.kafka.support.serializer.JsonSerializer  
aws.accessKeyId=${AWS\_ACCESS\_KEY\_ID}  
aws.secretKey=${AWS\_SECRET\_ACCESS\_KEY}  
aws.region=${AWS\_REGION}