Plasma Donor App with Serverless Computing

INTRODUCTION

- Overview

The main objective of this project is to provide the recipient with a donor who is in good form with no health ailments to donate blood of the corresponding blood group. This project provides quick access to donors for an immediate requirement of blood. In case of an emergency/surgery, blood procurement is always a major problem which consumes a lot of time. This helps serve the major time-lapse in which a life can be saved!

- Purpose

Serverless computing is the current trend in software application development. Microservices are a popular new approach for building maintainable, scalable, cloud-based applications. AWS is the perfect platform for hosting micro-services. In this project, we will be building a plasma donor app with AWS services like lambda functions, API gateway, and DynamoDB.

LITERATURE SURVEY

- Existing problem

During the COVID 19 crisis, the requirement of plasma became high and the donor count being low. Saving the donor information and helping the need by notifying the current donors would be a helping hand. In regard to the problem faced, an application is to be built which would take the donor details store it and inform them upon a request.

- Proposed solution

To build a web application that is capable of acting as a medium for recipients and donors of blood. The application must be deployed on Elastic Beanstalk. Create an API Endpoint for the model with the help of API Gateway and AWS Lambda Service. An alert is to be sent using the Simple Notification Service to all the registered users whenever a request for blood is posted.
Recipient: The one who has a requirement of blood can register for the first time and then log in from the next for any requirement of blood. He or She can provide the recipient's details such as the blood group, sex and age, and the minimum time for donation so that the admin as well as the donors can view and act accordingly.
Donor: The one who wants to donate blood can register and login to the site and check for any updates on requirements. If they wish to donate, they can get into contact with the recipient and proceed.

THEORITICAL ANALYSIS

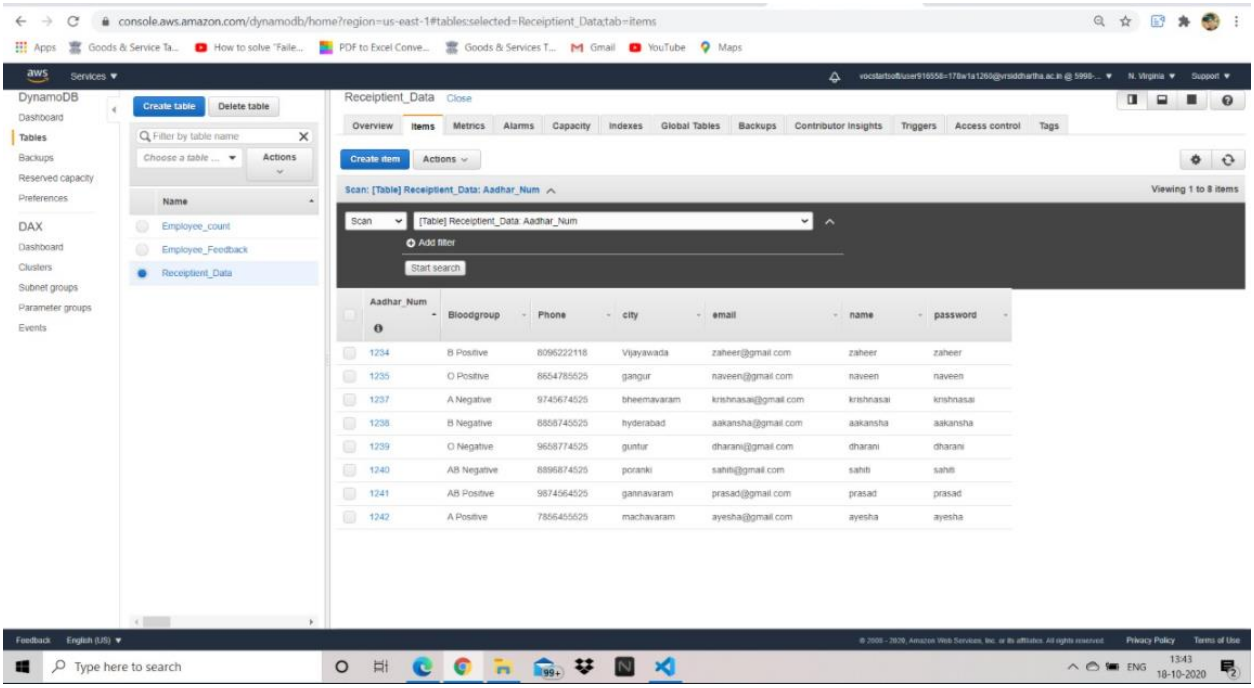
Hardware / Software designing
Block Diagram:

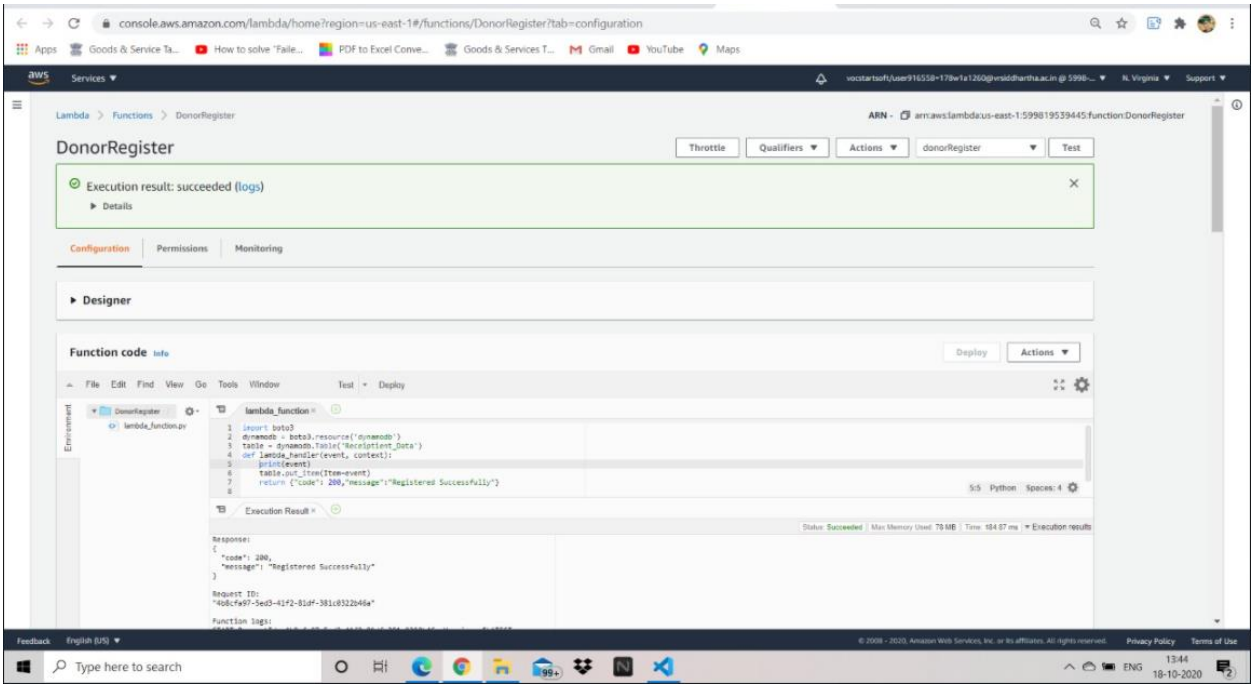
Project Work Flow:

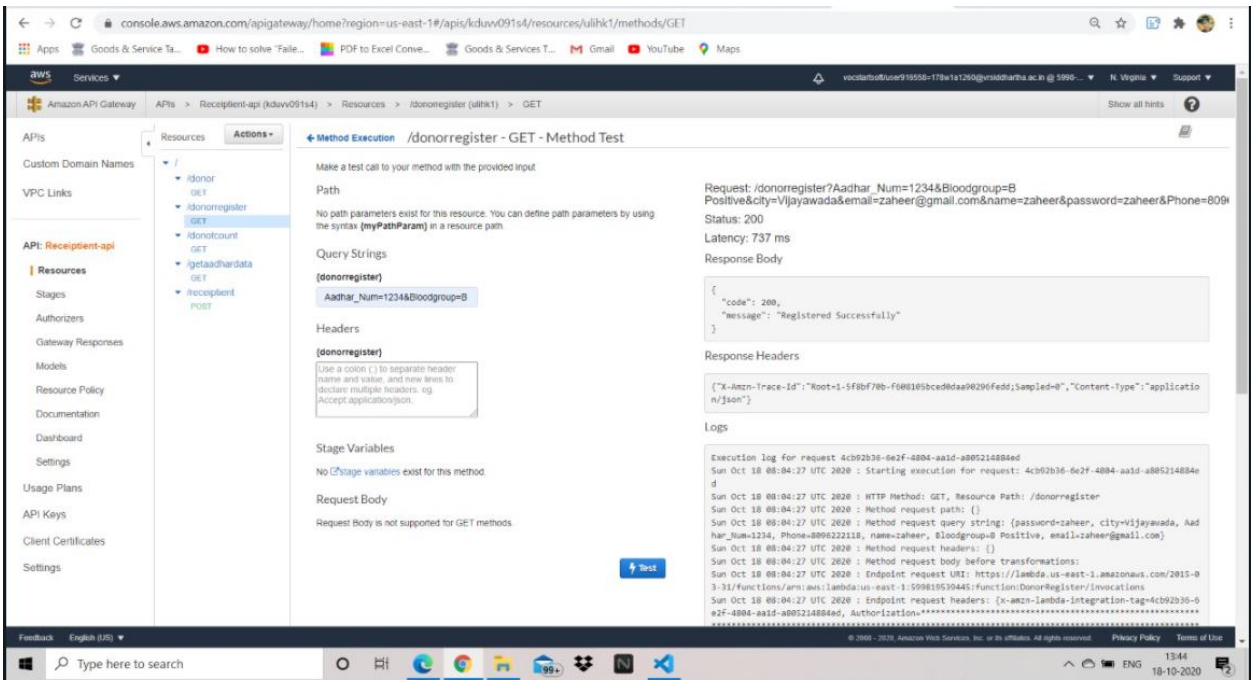
- The user interacts with the application.
- Register by giving the details as a donor.
- The database will have all the details and if a user posts a request then the concerned blood group donors will get notified about it.

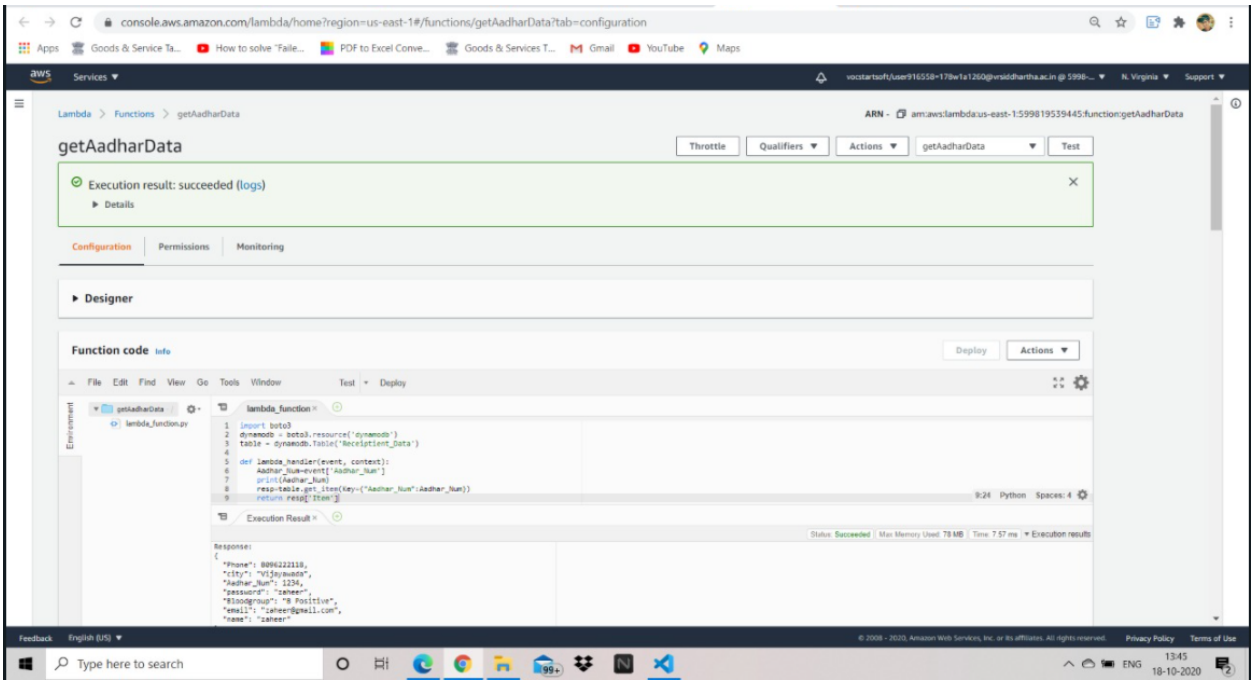
RESULT

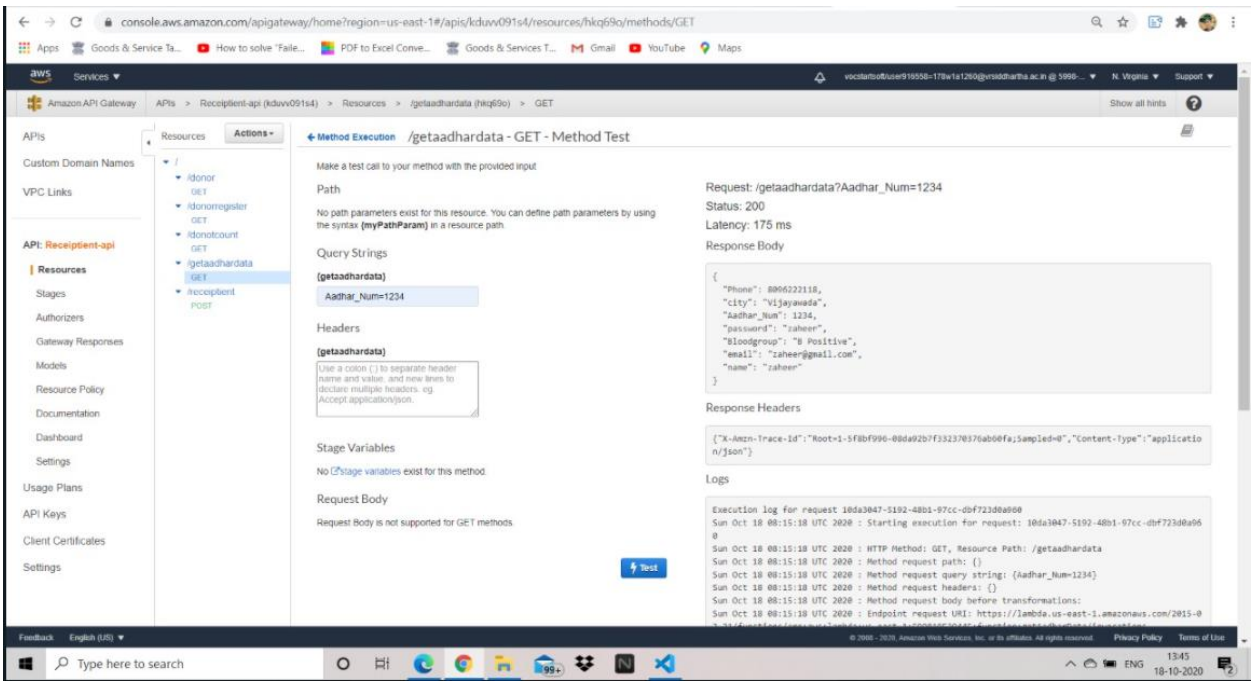
- Screenshots of output

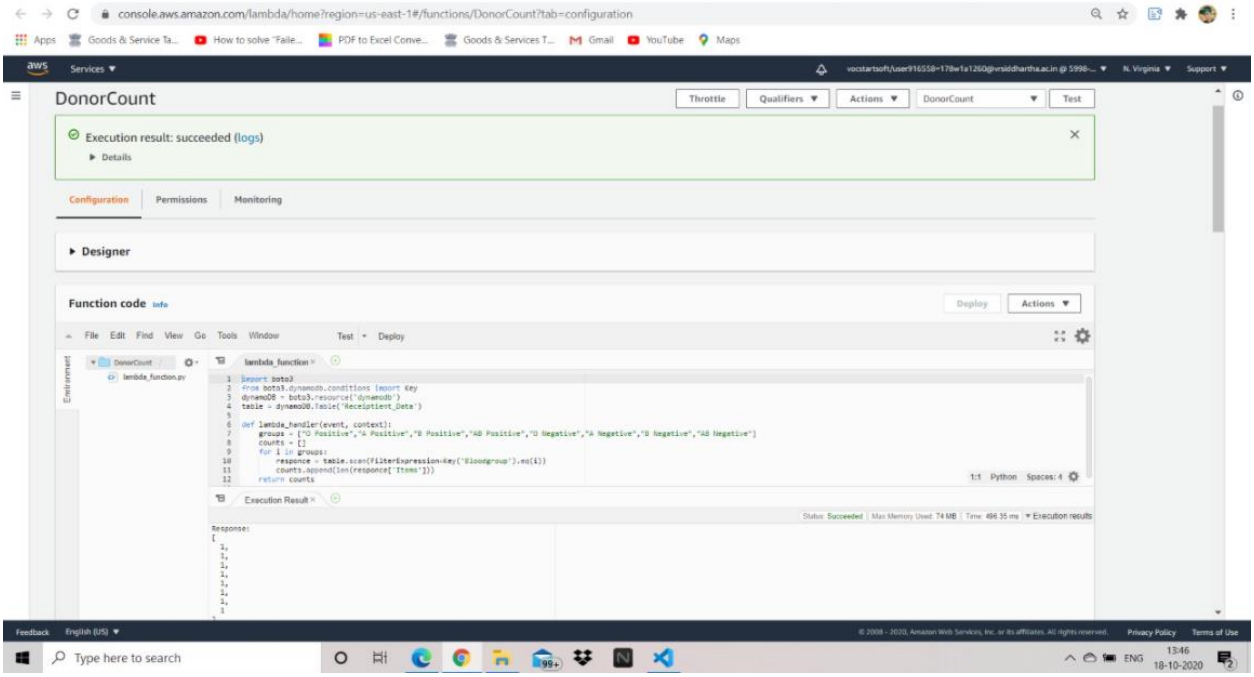


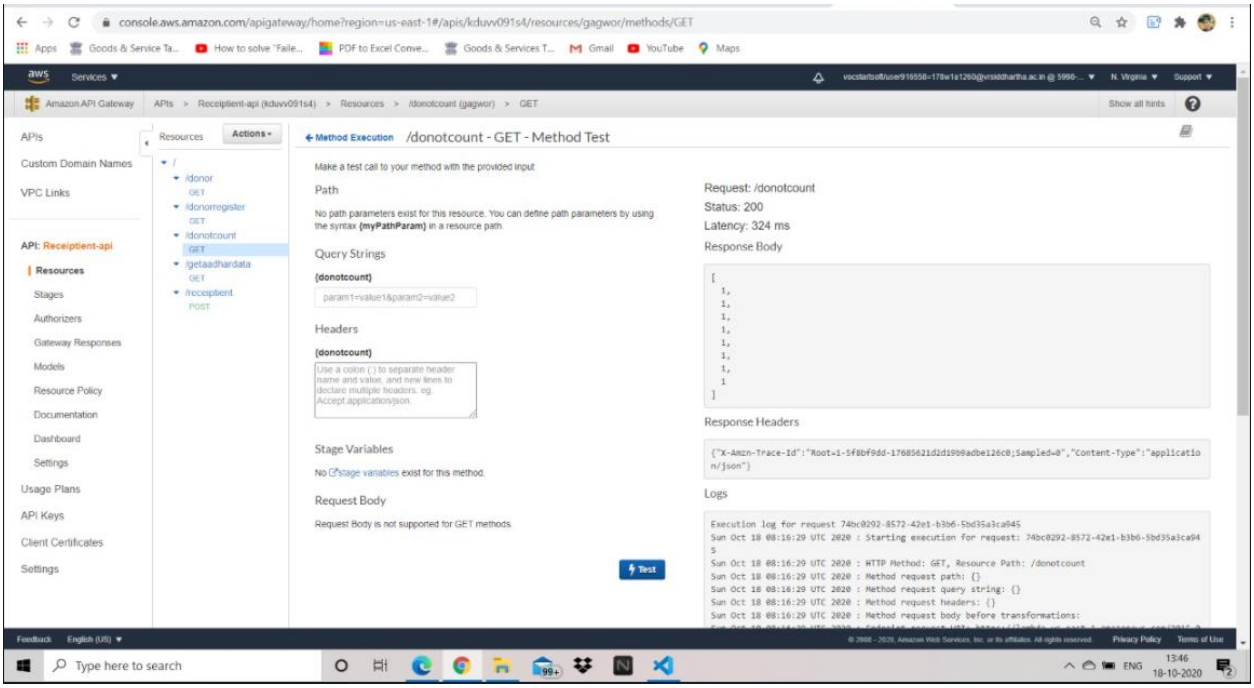


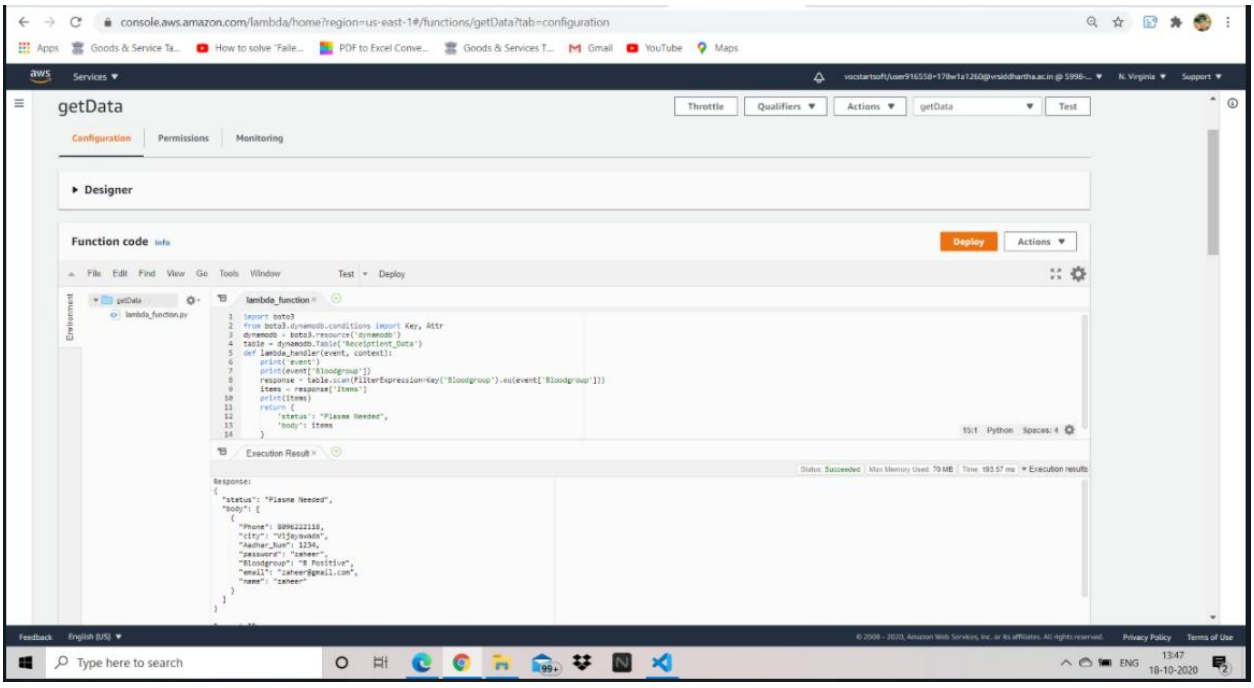

console.aws.amazon.com/apigateway/home?region=us-east-1#/apis/kduvv091s4/resources/ulihk1/methods/GET
Amazon API Gateway
APIs > Receiptient-api (kduvv091s4) > Resources > /donorregister (ulihk1) > GET
Method Execution /donorregister - GET - Method Test
Make a test call to your method with the provided input
Path
Query Strings
(donorregister)
Aadhar_Num=1234&Bloodgroup=B
Headers
(donorregister)
Request Body
Request Body is not supported for GET methods.
Test
Request: /donorregister?Aadhar_Num=1234&Bloodgroup=B
Positive&city=Vijayawada&email=zaheer@gmail.com&name=zaheer&password=zaheer&Phone=809
Status: 200
Latency: 737 ms
Response Body
{
"code": 200,
"message": "Registered Successfully"
}
Response Headers
{"X-Amzn-Trace-Id":"Root=1-5f8bf70b-f608105bced0daa90296fedd;Sampled=0","Content-Type":"applicatio
n/json"}
Logs
Execution log for request 4cb92b36-6e2f-4804-aa1d-a805214884ed
Sun Oct 18 08:04:27 UTC 2020 : Starting execution for request: 4cb92b36-6e2f-4804-aa1d-a805214884e
d
Sun Oct 18 08:04:27 UTC 2020 : HTTP Method: GET, Resource Path: /donorregister
Sun Oct 18 08:04:27 UTC 2020 : Method request path: {}
Sun Oct 18 08:04:27 UTC 2020 : Method request query string: {password=zaheer, city=Vijayawada, Aad
har_Num=1234, Phone=8096222118, name=zaheer, Bloodgroup=B Positive, email=zaheer@gmail.com}
Sun Oct 18 08:04:27 UTC 2020 : Method request headers: {}
Sun Oct 18 08:04:27 UTC 2020 : Method request body before transformations:
Sun Oct 18 08:04:27 UTC 2020 : Endpoint request URI: https://lambda.us-east-1.amazonaws.com/2015-0
3-31/functions/arn:aws:lambda:us-east-1:599819539445:function:DonorRegister/invocations
Sun Oct 18 08:04:27 UTC 2020 : Endpoint request headers: {x-amzn-lambda-integration-tag=4cb92b36-6
e2f-4804-aa1d-a805214884ed, Authorization=**********
Type here to search
ENG
13:44
18-10-2020

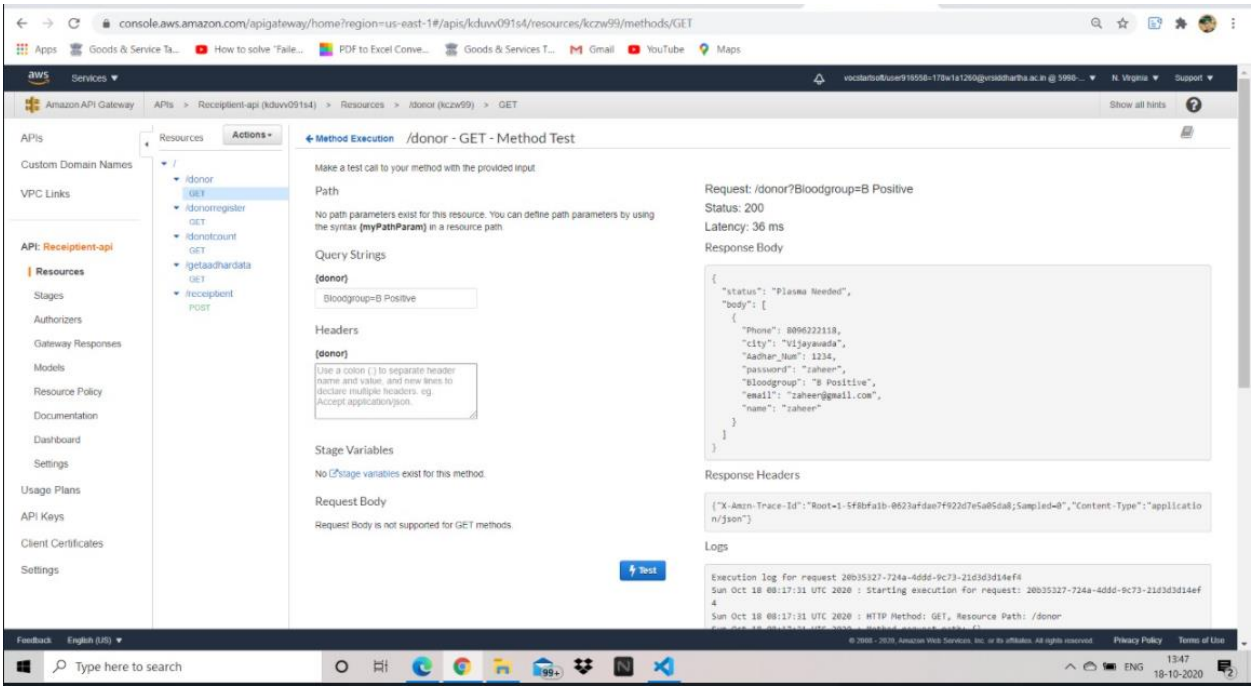

console.aws.amazon.com/lambda/home?region=us-east-1#/functions/getAadharData?tab=configuration
Lambda > Functions > getAadharData
ARN - arn:aws:lambda:us-east-1:599819539445:function:getAadharData
getAadharData
Throttle
Qualifiers
Actions
getAadharData
Test
Execution result: succeeded (logs)
Details
Configuration
Permissions
Monitoring
Designer
Function code Info
Deploy
Actions
File Edit Find View Go Tools Window Test Deploy
lambda_function.py
import boto3
dynamodb = boto3.resource('dynamodb')
table = dynamodb.Table('Receiptient_Data')
def lambda_handler(event, context):
Aadhar_Num=event['Aadhar_Num']
print(Aadhar_Num)
resp=table.get_item(Key={"Aadhar_Num":Aadhar_Num})
return resp['Item']
Execution Result
Response:
{
"Phone": 8096222118,
"city": "Vijayawada",
"Aadhar_Num": 1234,
"password": "zaheer",
"Bloodgroup": "B Positive",
"email": "zaheer@gmail.com",
"name": "zaheer"
Type here to search
ENG
13:45
18-10-2020

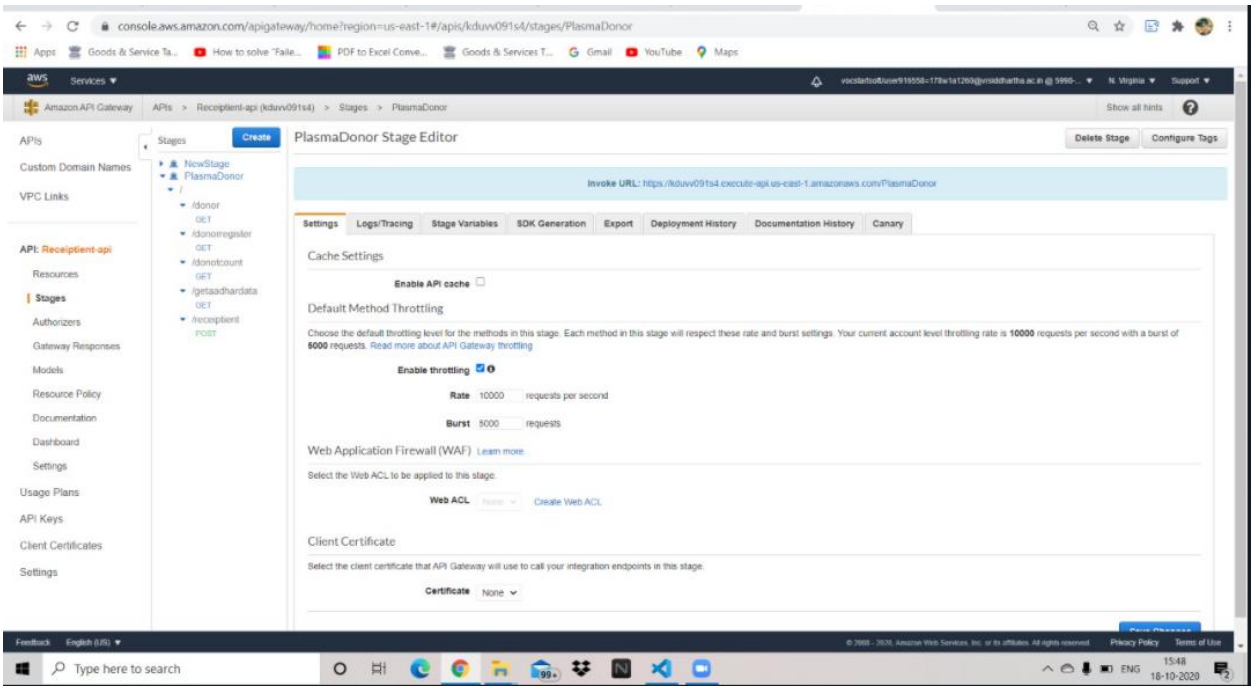

console.aws.amazon.com/apigateway/home?region=us-east-1#/apis/kduvv091s4/resources/hkq69o/methods/GET
Amazon API Gateway
APIs
Custom Domain Names
VPC Links
API: Receiptient-api
Resources
Stages
Authorizers
Gateway Responses
Models
Resource Policy
Documentation
Dashboard
Settings
Usage Plans
API Keys
Client Certificates
Settings
/donor
/donorregister
/donotcount
/getaadhardata
/receiptient
Method Execution
/getaadhardata - GET - Method Test
Make a test call to your method with the provided input
Path
No path parameters exist for this resource. You can define path parameters by using the syntax {myPathParam} in a resource path.
Query Strings
(getaadhardata)
Aadhar_Num=1234
Headers
(getaadhardata)
Stage Variables
No stage variables exist for this method.
Request Body
Request Body is not supported for GET methods.
Test
Request: /getaadhardata?Aadhar_Num=1234
Status: 200
Latency: 175 ms
Response Body
{
"Phone": 8096222118,
"city": "Vijayawada",
"Aadhar_Num": 1234,
"password": "zaheer",
"Bloodgroup": "B Positive",
"email": "zaheer@gmail.com",
"name": "zaheer"
}
Response Headers
{"X-Amzn-Trace-Id":"Root=1-5f8bf996-00da92b7f332370376ab60fa;Sampled=0","Content-Type":"application/json"}
Logs
Execution log for request 10da3047-5192-48b1-97cc-dbf723d0a960
Sun Oct 18 08:15:18 UTC 2020 : Starting execution for request: 10da3047-5192-48b1-97cc-dbf723d0a960
Sun Oct 18 08:15:18 UTC 2020 : HTTP Method: GET, Resource Path: /getaadhardata
Sun Oct 18 08:15:18 UTC 2020 : Method request path: {}
Sun Oct 18 08:15:18 UTC 2020 : Method request query string: {Aadhar_Num=1234}
Sun Oct 18 08:15:18 UTC 2020 : Method request headers: {}
Sun Oct 18 08:15:18 UTC 2020 : Method request body before transformations:
Sun Oct 18 08:15:18 UTC 2020 : Endpoint request URI: https://lambda.us-east-1.amazonaws.com/2015-0
Type here to search
13:45
18-10-2020

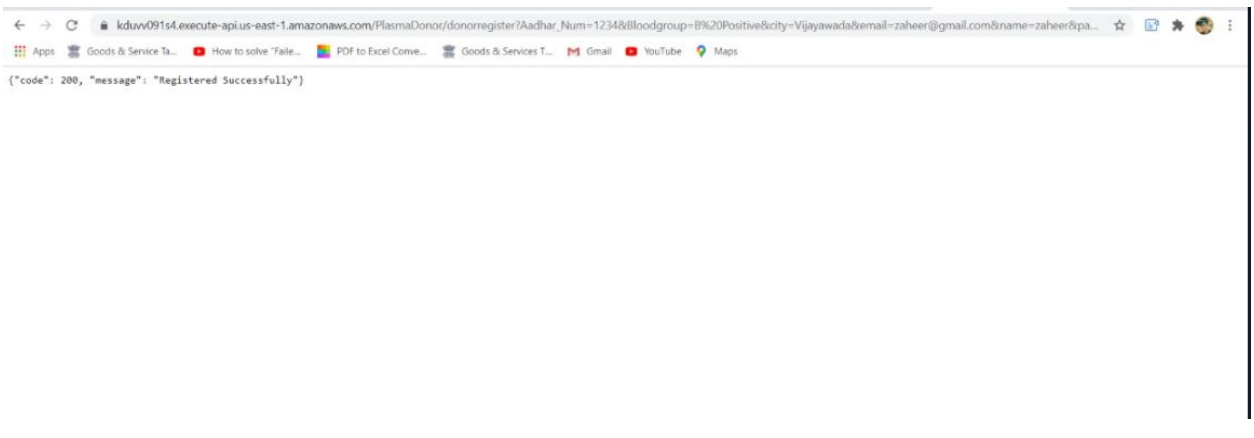

console.aws.amazon.com/lambda/home?region=us-east-1#/functions/DonorCount?tab=configuration
DonorCount
Throttle
Qualifiers
Actions
DonorCount
Test
Execution result: succeeded (logs)
Details
Configuration
Permissions
Monitoring
Designer
Function code
Deploy
Actions
File Edit Find View Go Tools Window Test Deploy
lambda_function.py
import boto3
from boto3.dynamodb.conditions import Key
dynamo08 = boto3.resource('dynamodb')
table = dynamo08.Table('Receiptient_Data')
def lambda_handler(event, context):
groups = ["O Positive","A Positive","B Positive","AB Positive","O Negative","A Negative","B Negative","AB Negative"]
counts = []
for i in groups:
response = table.scan(FilterExpression=Key('Bloodgroup').eq(i))
counts.append(len(response['Items']))
return counts
Execution Result
Response:
[
1,
1,
1,
1,
1,
2,
1,
1
]
Type here to search
13:46
18-10-2020

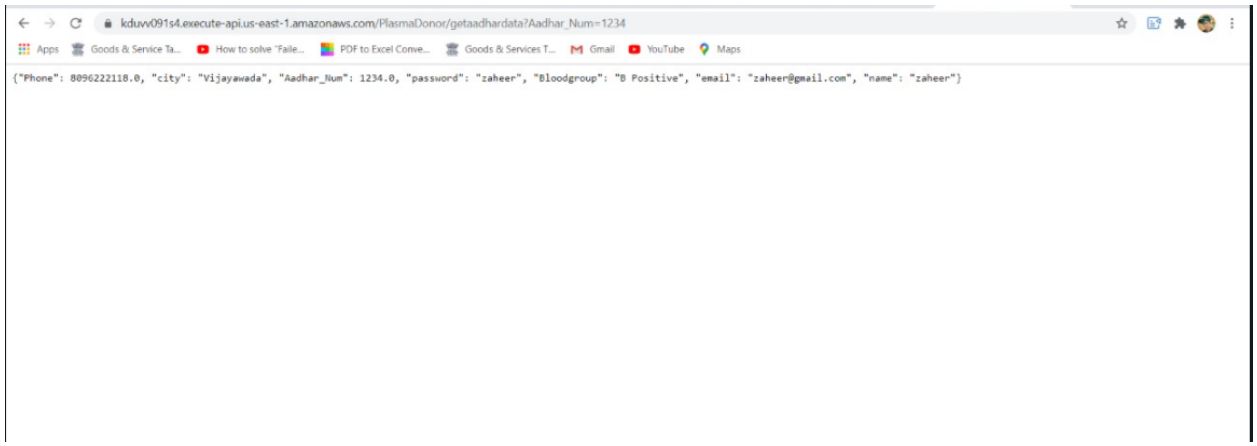

console.aws.amazon.com/apigateway/home?region=us-east-1#/apis/kduvv091s4/resources/gagwor/methods/GET
Amazon API Gateway
APIs > Receiptient-api (kduvv091s4) > Resources > /donotcount (gagwor) > GET
Resources
Actions
/donor
/donorregister
/donotcount
/getaadhardata
/receiptient
Method Execution /donotcount - GET - Method Test
Make a test call to your method with the provided input
Path
No path parameters exist for this resource. You can define path parameters by using the syntax {myPathParam} in a resource path.
Query Strings
(donotcount)
param1=value1&param2=value2
Headers
(donotcount)
Use a colon (:) to separate header name and value, and new lines to declare multiple headers. eg. Accept:application/json
Stage Variables
No stage variables exist for this method.
Request Body
Request Body is not supported for GET methods.
Test
Request: /donotcount
Status: 200
Latency: 324 ms
Response Body
[1, 1, 1, 1, 1, 1, 1, 1]
Response Headers
{"X-Amzn-Trace-Id":"Root=1-5f8bf9dd-17685621d2d1909adbe126c0;Sampled=0","Content-Type":"application/json"}
Logs
Execution log for request 74bc0292-8572-42e1-b3b6-5bd35a3ca945
Sun Oct 18 08:16:29 UTC 2020 : Starting execution for request: 74bc0292-8572-42e1-b3b6-5bd35a3ca945
Sun Oct 18 08:16:29 UTC 2020 : HTTP Method: GET, Resource Path: /donotcount
Sun Oct 18 08:16:29 UTC 2020 : Method request path: {}
Sun Oct 18 08:16:29 UTC 2020 : Method request query string: {}
Sun Oct 18 08:16:29 UTC 2020 : Method request headers: {}
Sun Oct 18 08:16:29 UTC 2020 : Method request body before transformations:

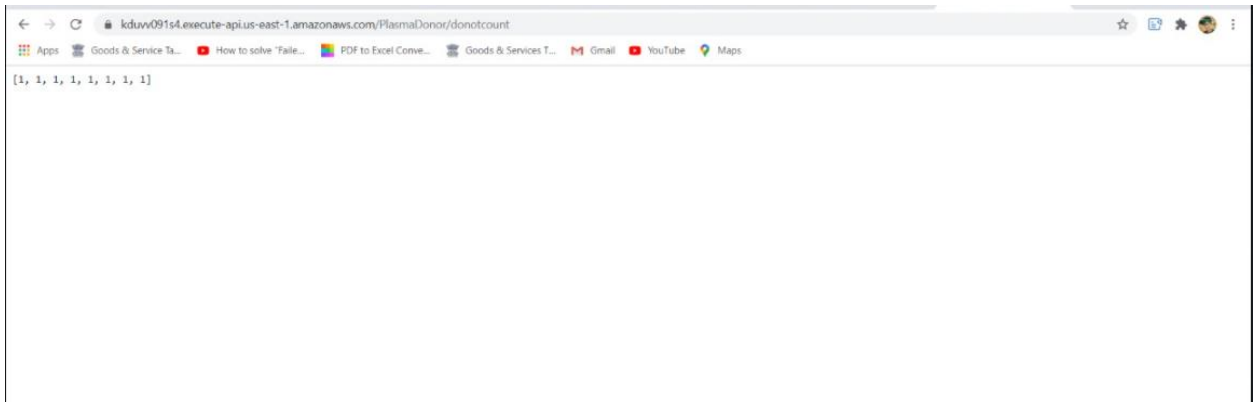

console.aws.amazon.com/lambda/home?region=us-east-1#/functions/getData?tab=configuration
getData
Throttle
Qualifiers
Actions
getData
Test
Configuration
Permissions
Monitoring
Designer
Function code info
Deploy
Actions
lambda_function.py
import boto3
from boto3.dynamodb.conditions import Key, Attr
dynamodb = boto3.resource('dynamodb')
table = dynamodb.Table('Receiptient_Data')
def lambda_handler(event, context):
print('event')
print(event['Bloodgroup'])
response = table.scan(FilterExpression=Key('Bloodgroup').eq(event['Bloodgroup']))
items = response['Items']
print(items)
return {
'status': 'Plasma Needed',
'body': items
}
Execution Result
Response:
{
"status": "Plasma Needed",
"body": [
{
"Phone": 8096222118,
"city": "Vijayawada",
"Aadhar_Num": 1234,
"password": "zaheer",
"Bloodgroup": "B Positive",
"email": "zaheer@gmail.com",
"name": "zaheer"
}
]
}

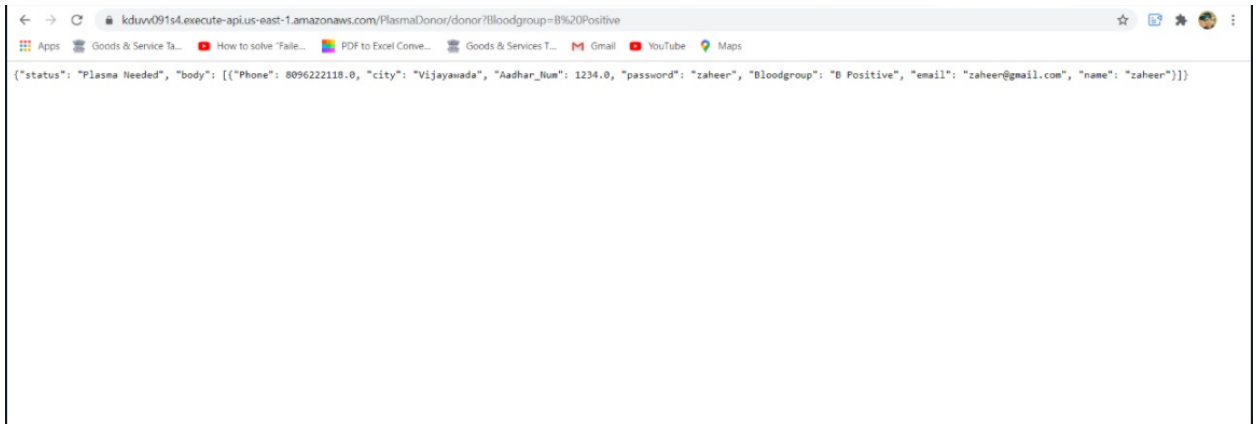

console.aws.amazon.com/apigateway/home?region=us-east-1#/apis/kduvv091s4/resources/kczw99/methods/GET
Amazon API Gateway
APIs > Receiptient-api (kduvv091s4) > Resources > /donor (kczw99) > GET
/donor - GET - Method Test
Make a test call to your method with the provided input
Path
No path parameters exist for this resource. You can define path parameters by using the syntax {myPathParam} in a resource path.
Query Strings
{donor}
Bloodgroup=B Positive
Headers
{donor}
Use a colon (:) to separate header name and value, and new lines to declare multiple headers. eg. Accept:application/json.
Stage Variables
No stage variables exist for this method.
Request Body
Request Body is not supported for GET methods.
Test
Request: /donor?Bloodgroup=B Positive
Status: 200
Latency: 36 ms
Response Body
{
"status": "Plasma Needed",
"body": [
{
"Phone": 8096222118,
"city": "Vijayawada",
"Aadhar_Num": 1234,
"password": "zaheer",
"Bloodgroup": "B Positive",
"email": "zaheer@gmail.com",
"name": "zaheer"
}
]
}
Response Headers
{"X-Amzn-Trace-Id":"Root=1-5f8bfa1b-0623afdae7f922d7e5a05da8;Sampled=0","Content-Type":"application/json"}
Logs
Execution log for request 20b35327-724a-4ddd-9c73-21d3d3d14ef4
Sun Oct 18 08:17:31 UTC 2020 : Starting execution for request: 20b35327-724a-4ddd-9c73-21d3d3d14ef4
Sun Oct 18 08:17:31 UTC 2020 : HTTP Method: GET, Resource Path: /donor
Type here to search
13:47
18-10-2020

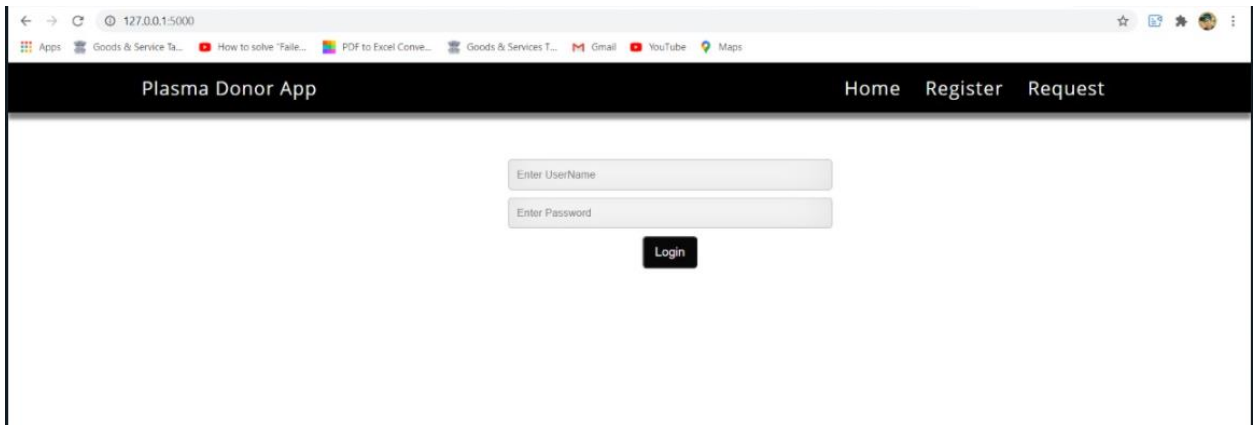

console.aws.amazon.com/apigateway/home?region=us-east-1#/apis/kduvv091s4/stages/PlasmaDonor
Amazon API Gateway
APIs > Receiptient-api (kduvv091s4) > Stages > PlasmaDonor
PlasmaDonor Stage Editor
Delete Stage
Configure Tags
Invoke URL: https://kduvv091s4.execute-api.us-east-1.amazonaws.com/PlasmaDonor
Settings
Logs/Tracing
Stage Variables
SDK Generation
Export
Deployment History
Documentation History
Canary
Cache Settings
Enable API cache
Default Method Throttling
Choose the default throttling level for the methods in this stage. Each method in this stage will respect these rate and burst settings. Your current account level throttling rate is 10000 requests per second with a burst of 5000 requests. Read more about API Gateway throttling
Enable throttling
Rate 10000 requests per second
Burst 5000 requests
Web Application Firewall (WAF) Learn more
Select the Web ACL to be applied to this stage.
Web ACL
Create Web ACL
Client Certificate
Select the client certificate that API Gateway will use to call your integration endpoints in this stage.
Certificate None
Type here to search
15:48
18-10-2020

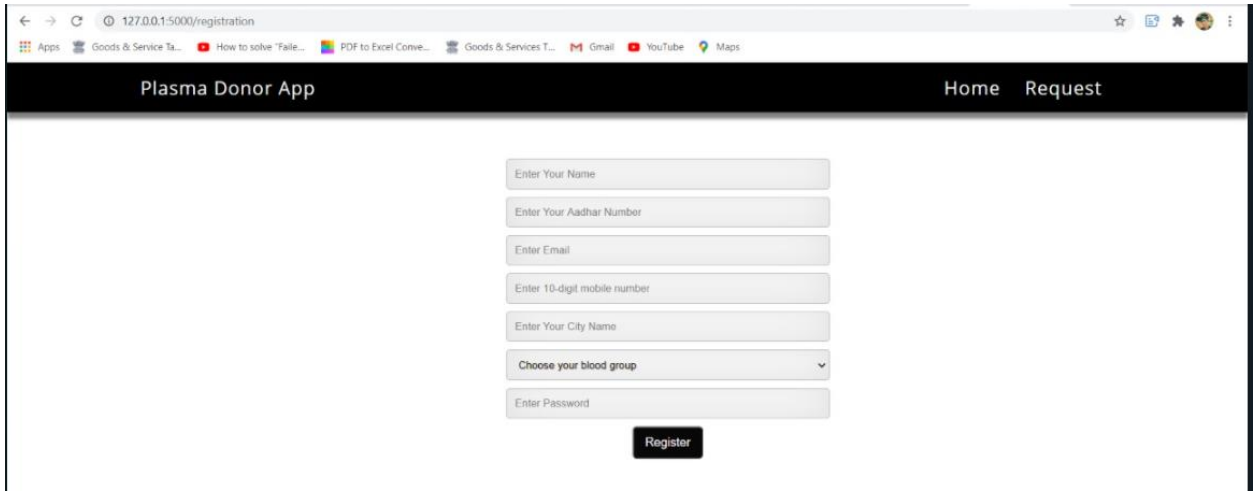

kduvv091s4.execute-api.us-east-1.amazonaws.com/PlasmaDonor/donorregister?Aadhar_Num=1234&Bloodgroup=B%20Positive&city=Vijayawada&email=zaheer@gmail.com&name=zaheer&pa...
{"code": 200, "message": "Registered Successfully"}

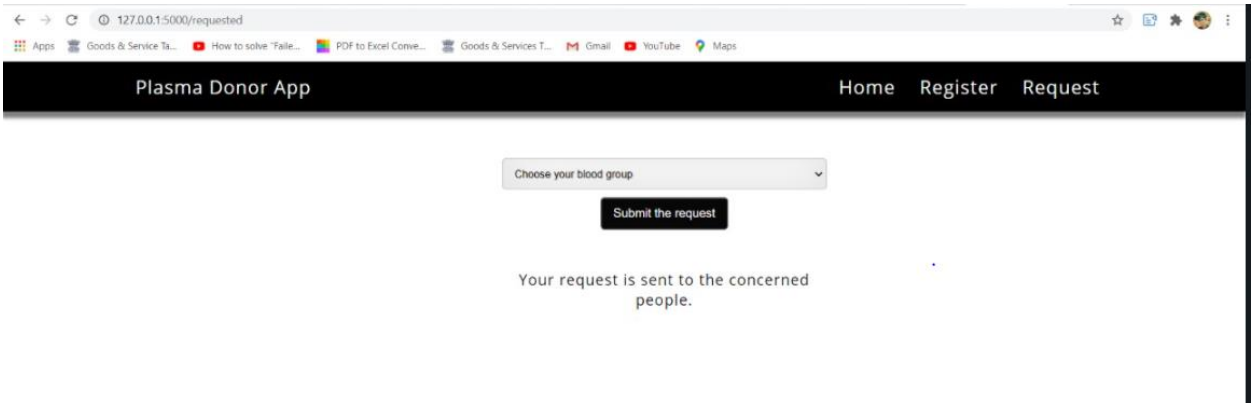

kduvv091s4.execute-api.us-east-1.amazonaws.com/PlasmaDonor/getaadhardata?Aadhar_Num=1234
Apps
Goods & Service Ta...
How to solve "Faile...
PDF to Excel Conve...
Goods & Services T...
Gmail
YouTube
Maps
{"Phone": 8096222118.0, "city": "Vijayawada", "Aadhar_Num": 1234.0, "password": "zaheer", "Bloodgroup": "B Positive", "email": "zaheer@gmail.com", "name": "zaheer"}

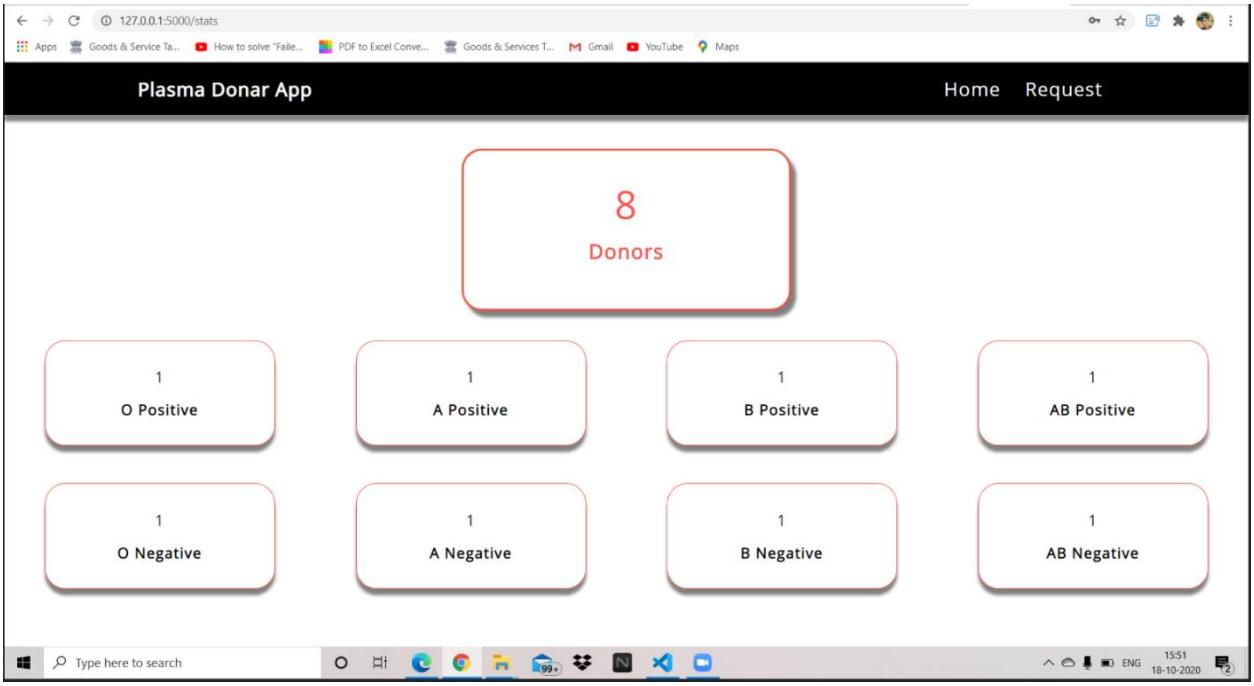

kduvv091s4.execute-api.us-east-1.amazonaws.com/PlasmaDonor/donotcount
Apps
Goods & Service Ta...
How to solve "Faile...
PDF to Excel Conve...
Goods & Services T...
Gmail
YouTube
Maps
[1, 1, 1, 1, 1, 1, 1]


kduvv091s4.execute-api.us-east-1.amazonaws.com/PlasmaDonor/donor?Bloodgroup=B%20Positive
Apps
Goods & Service Ta...
How to solve "Faile...
PDF to Excel Conve...
Goods & Services T...
Gmail
YouTube
Maps
{"status": "Plasma Needed", "body": [{"Phone": 8096222118.0, "city": "Vijayawada", "Aadhar_Num": 1234.0, "password": "zaheer", "Bloodgroup": "B Positive", "email": "zaheer@gmail.com", "name": "zaheer"}]}


127.0.0.1:5000
Apps
Goods & Service Ta...
How to solve "Faile...
PDF to Excel Conve...
Goods & Services T...
Gmail
YouTube
Maps
Plasma Donor App
Home
Register
Request
Enter UserName
Enter Password
Login


127.0.0.1:5000/registration
Apps
Goods & Service Ta...
How to solve "Faile...
PDF to Excel Conve...
Goods & Services T...
Gmail
YouTube
Maps
Plasma Donor App
Home
Request
Enter Your Name
Enter Your Aadhar Number
Enter Email
Enter 10-digit mobile number
Enter Your City Name
Choose your blood group
Enter Password
Register


127.0.0.1:5000/requested
Plasma Donor App
Home
Register
Request
Choose your blood group
Submit the request
Your request is sent to the concerned people.


127.0.0.1:5000/stats
Plasma Donar App
Home
Request
8
Donors
1
O Positive
1
A Positive
1
B Positive
1
AB Positive
1
O Negative
1
A Negative
1
B Negative
1
AB Negative
Type here to search
ENG
15:51
18-10-2020

ADVANTAGES & DISADVANTAGES

It has many advantages; it has a very simple and clean user interface. A large number of Active Donors around can help a large number of needy people free of cost.
There are some disadvantages also, it can be misused for back marketing, and false blood request and registration vulnerabilities are also there.

APPLICATIONS

- Thousands of Active Donors around can help in any public workplaces and societies.
- Thousands of Donors immediately Get a push notification/email/SMS of your Request.
- It can be used in hospitals, big public institutions and workplaces.

CONCLUSION

Plasma & Blood Donation App ,which puts the power to save a live in the palm of your hand. The main purpose of this App is to create & manage a platform for all donors of the world & remove the recent crisis

FUTURE SCOPE

Many more improvements can be made in future, with larger community with integrity and variety , and its area of application shall be highly innovated and implemented.

BIBILOGRAPHY

- smartinternz.com
- AWS Educate resources
- https://www.ncbi.nlm.nih.gov/books/NBK138212/
- Eder A, et al. Selection criteria to protect the blood donor in North America and Europe: past (dogma), present (evidence), and future (hemovigilance). Transfusion Medicine Reviews. 2009;23(3):205–220. [PubMed]
- Moreno J. “Creeping precautionism” and the blood supply. Transfusion. 2003;43:840–842. [PubMed]
- Farrugia A. The mantra of blood safety: time for a new tune? Vox Sanguinis. 2004;86:1–7. [PubMed]