**CMPE 272 - Project**

University: [San Jose State University (SJSU)](http://www.sjsu.edu/)

Course: Enterprise Software Platform

Professor: Andrew Bond

Team: **Techno Spartan**

|  |  |
| --- | --- |
| Name | Student ID |
| Parvathi Pai | 015293460 |
| Shreya Ghotankar | 015304393 |
| Sania Gonsalves | 015313974 |

|  |  |
| --- | --- |
| Techno Spartan | Version: 2.0 |
| Project Plan | Date: 12/04/2020 |

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 09/23/2020 | 0.1 | Initial Draft | Parvathi Pai |
| 09/27/2020 | 1.0 | Minor edits and published | Parvathi Pai  Shreya Ghotankar  Sania Gonsalves |
| 11/30/2020 | 1.1 | Changes to architecture diagram | Shreya Ghotankar |
| 12/01/2020 | 1.2 | Update Deployment information | Parvathi Pai |
| 12/04/2020 | 2.0 | Peer Review and Final publish document |  |

Table of Contents

[**Introduction** 4](#_Toc57897091)

[**Project Idea** 4](#_Toc57897092)

[**Architecture Diagram** 5](#_Toc57897093)

[**High Level Design** 6](#_Toc57897094)

[**Project Group** 9](#_Toc57897095)

[**Technology and Services Stack** 9](#_Toc57897096)

[**Demo Screenshots** 9](#_Toc57897097)

[**Development** 10](#_Toc57897098)

[**Deployment** 13](#_Toc57897099)

[**Future Work** 14](#_Toc57897100)

[**References** 15](#_Toc57897101)

# **Introduction**

Purpose of this document

The purpose of this document is to provide a detailed project description of the application called “Relief Fund”, which is designed to help people make contribution in the world by making donations to different and diverse charities registered or listed on the application. This document includes details about the application idea, development, deployment, and deliverables.

Scope

This document defines the project plan of the “Relief Fund” application. The overview includes project idea, individual contribution, development process, deployment process and deliverables.

# **Project Idea**

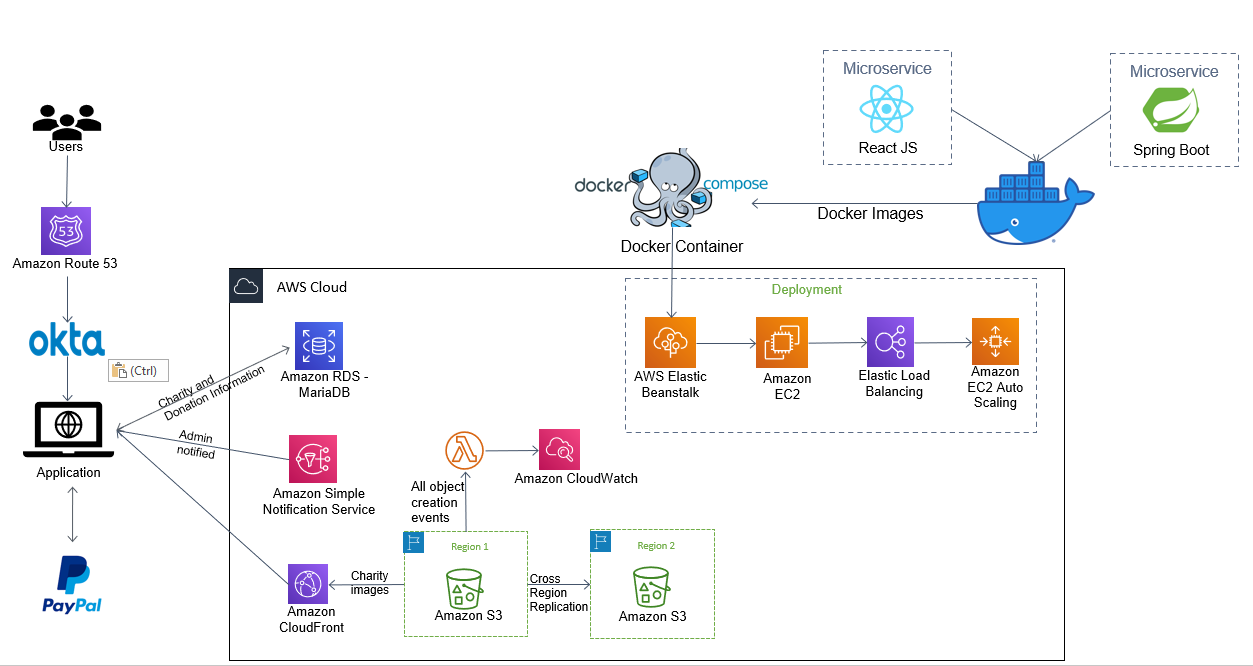
Around 207 natural disasters were recorded in the beginning of 2020 this includes COVID-19, Wildfires, Hurricanes, Storms etc. As we are socially distancing right now due to pandemic, why not try to provide comfort to the needed through a small gesture by donating.

Our team has created a platform for people to contribute to 501(c)(3) approved charitable institutions. This application denotes an organization which helps impacted people and animal rescue/shelter receive funds for restoration. Different charities can register on our application and get approved to receive donations from application users.

Our application is called Relief Fund and it has three features –

* Spot Donation
* Registered Users can donate to already listed and approved charities
* Charity institutions can join our platform

# **Architecture Diagram**



# **High Level Design**

Use Case 1 : Anyone can come and donate to charities anonymously.Diagram

Description automatically generated

Use Case 2 : User wishes to donate through our application platform.Diagram

Description automatically generated

Use Case 3 : New charity organizations can register on Relief Fund by getting approval from application Admin.

Diagram

Description automatically generated

# **Project Group**

|  |  |
| --- | --- |
| **Name** | **Responsibility** |
| Parvathi Pai | Charity Registration, PayPal Integration UI, AWS services setup, Admin Dashboard, Docker Deployment, Documentation |
| Shreya Ghotankar | Authentication and Authorization, Role-based access, UI, CI/CD, AWS services setup, Documentation |
| Sania Gonsalves | Charity Registration, PayPal Integration, Admin Dashboard, UI, AWS services setup |

# **Technology and Services Stack**

* Front-end – React JS, CSS, Javascript
* Spring Boot
* Authentication and Authorization – Okta
* Payment Gateway – PayPal Sandbox
* Database - AWS RDS MariaDB
* Images – AWS S3, Amazon CloudFront
* S3 Events – AWS Lambda, Amazon CloudWatch
* Notifications – Amazon SNS
* Deployment – Docker, Docker Compose, AWS Elastic Beanstalk

# **Demo Screenshots**

Link to the demo - <https://youtu.be/xqQx2U5MEkE>

Link to the website - <http://relieffund.parvathipai.com/>

Home PageGraphical user interface, text, application, email, website

Description automatically generated

Spot DonateGraphical user interface, application, website

Description automatically generated

LoginGraphical user interface

Description automatically generated

RegularGraphical user interface, website

Description automatically generated

Register CharityGraphical user interface

Description automatically generated

Admin

Graphical user interface, application

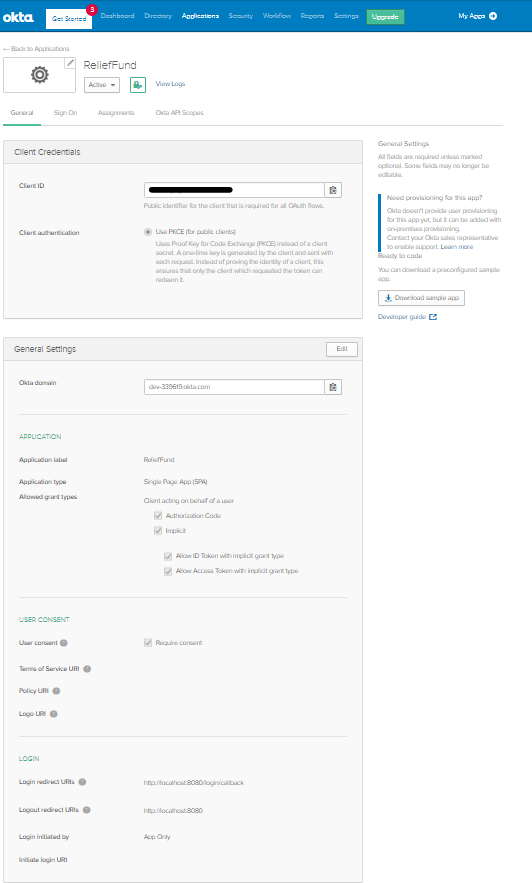
Description automatically generated

# **Development**

GitHub URL: https://github.com/ParvathiRPai/CMPE-272-Final-Project

1. Authentication and Authorization

We have used OKTA for our authentication and authorization feature. User can Login or Sign Up using the Okta Sign-in Widget integrated in our React JS application. We have also implemented role-based access feature. Only select users who are in Admin role will be able to view the Admin dashboard and approve the registering charities.

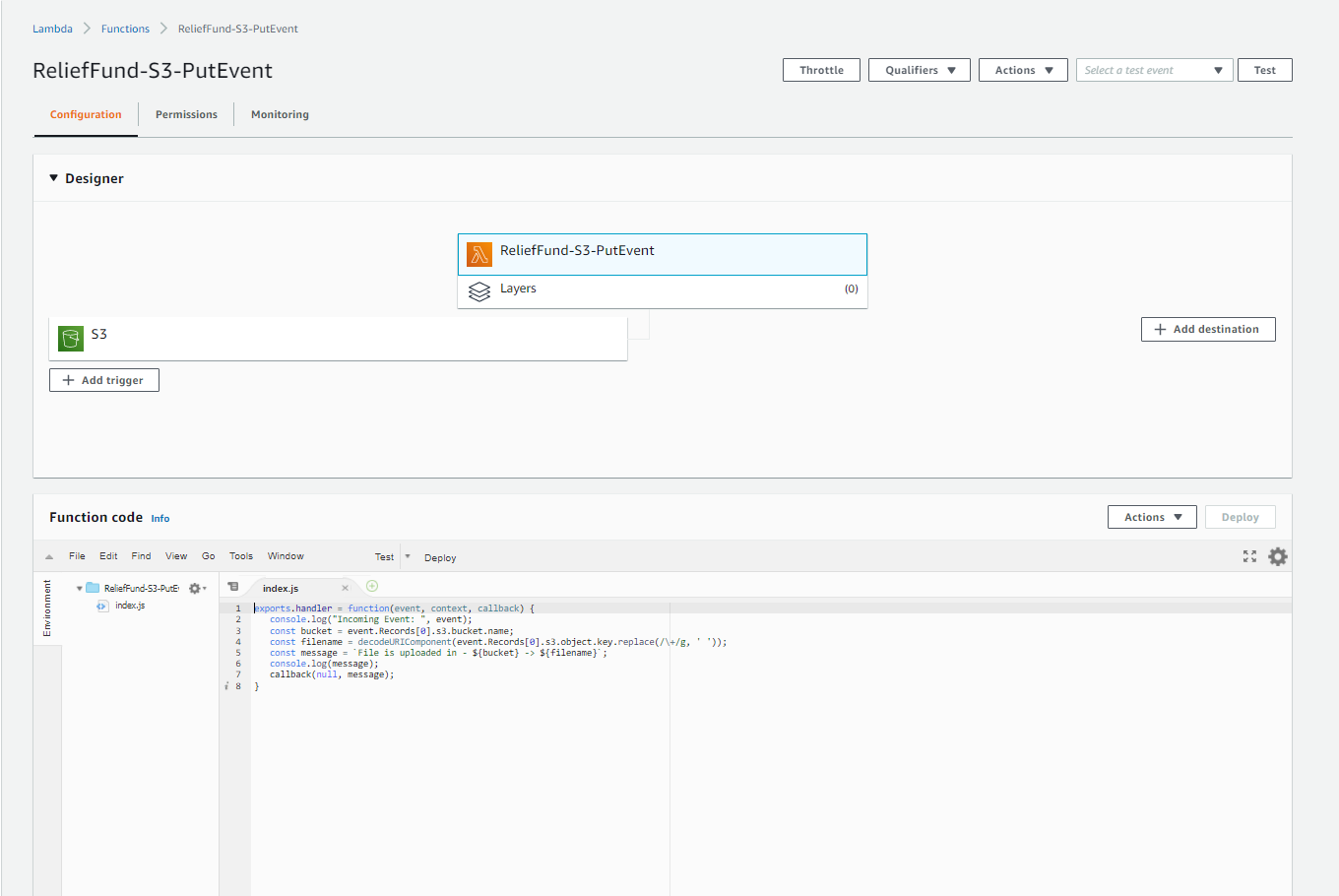


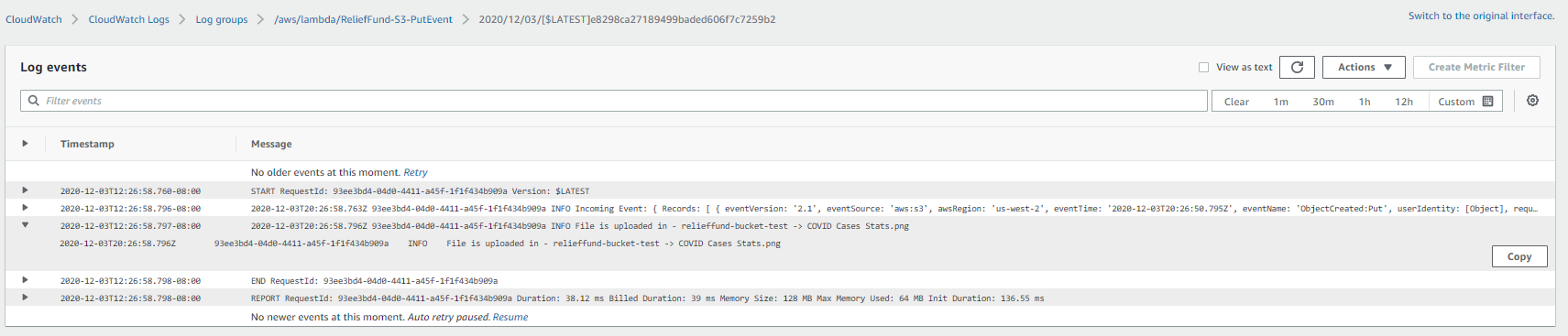
1. Charity Registration

Authenticated User will be able to register their charity by accessing the Register Page and filling out the registration form. As soon as user submits the form the Admin is notified through Amazon SNS email notification. Admin can then login to the application and go to the Admin Dashboard and Approve the charity requesting for addition to our portal.

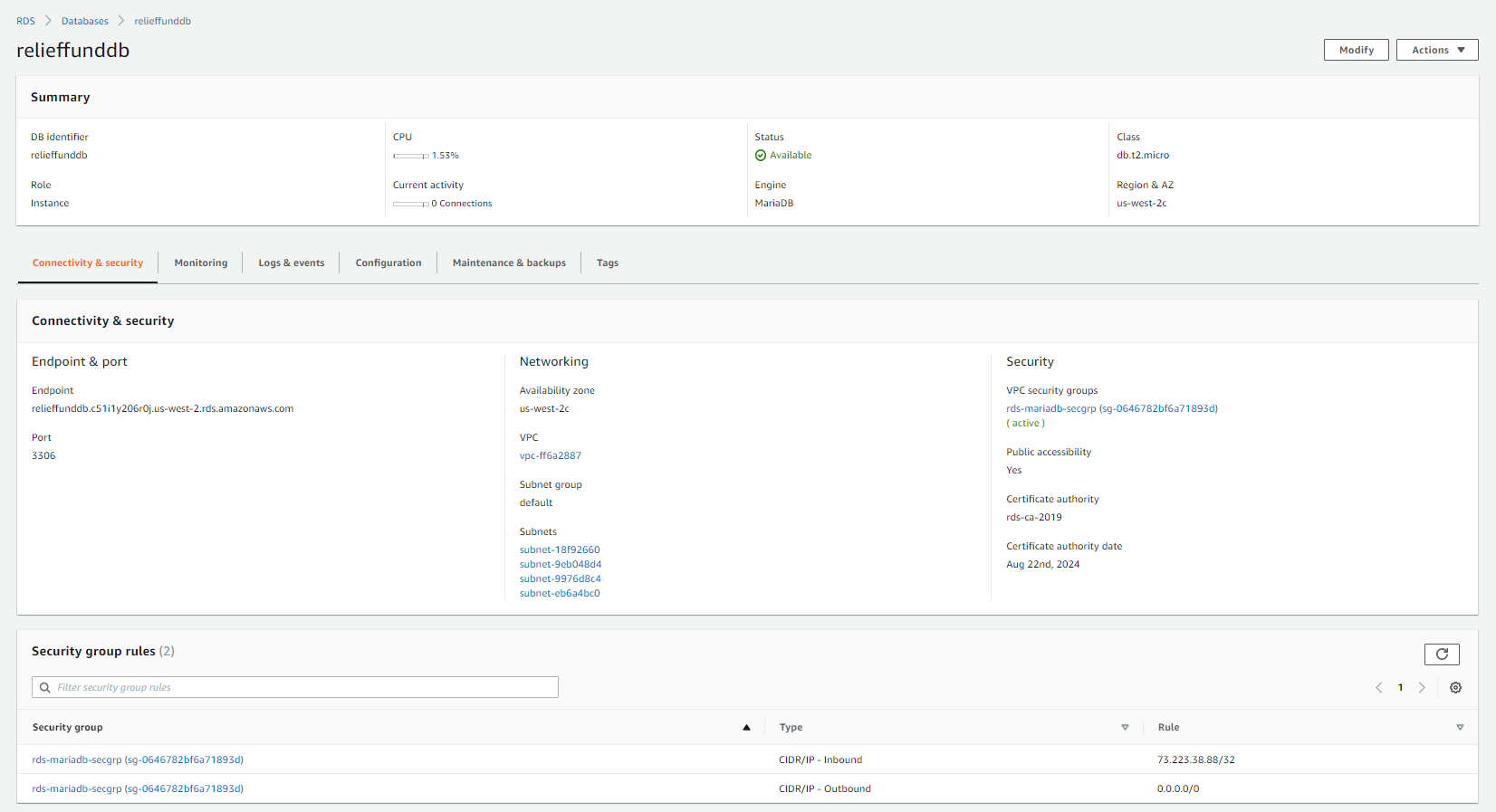
1. As part of Charity Registration, user uploads image for their charity which is stored in Amazon S3 bucket. We have implemented an AWS Lambda Function which gets triggered every time object is created and gets logged in Amazon CloudWatch Logs.

After Admin Approves the charity, the name and image appear in the available charity list.



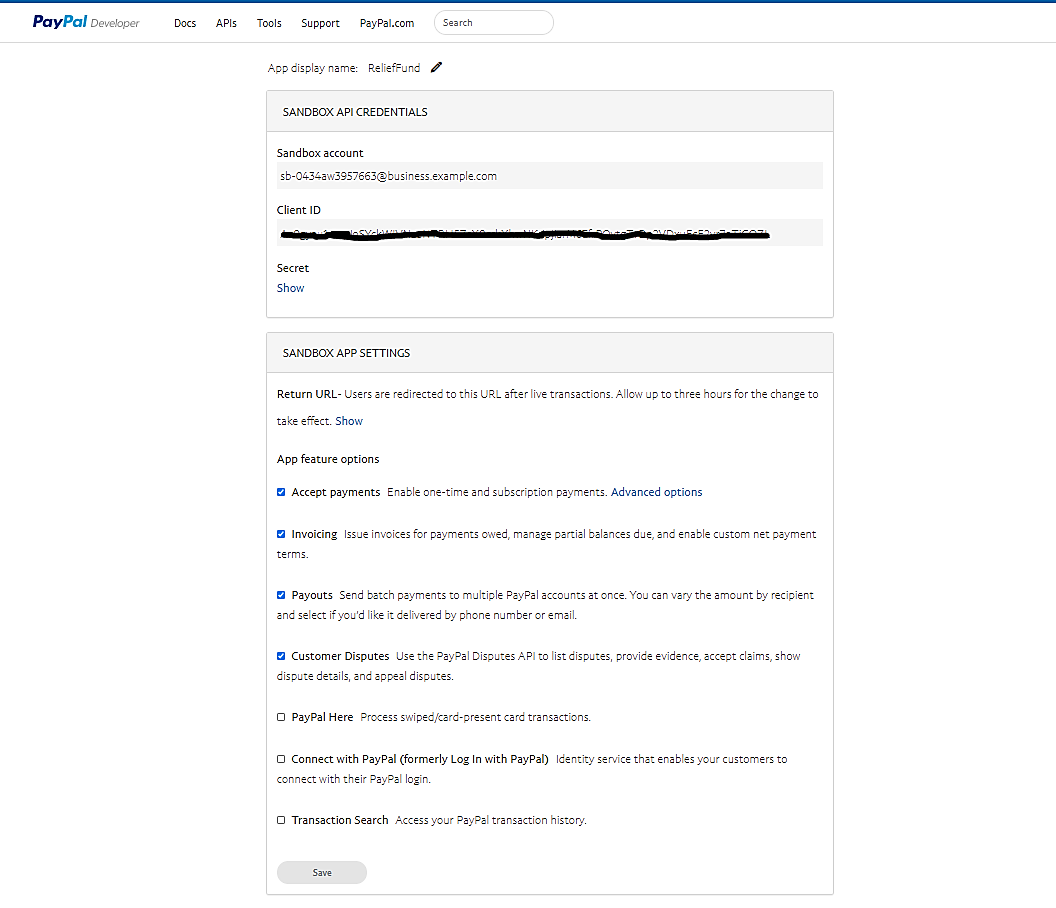


1. We have used AWS RDS – MariaDB for storing our Charity data and donations transactions.



1. PayPal Gateway

We have implemented PayPal Sandbox in our application as a payment feature. Authenticated users can select charities they want to donate to and then using the PayPal donate feature they can make their contributions.



# **Deployment**

1) For deployment we have created two containers for front end and backend service and hosted in elastic bean stalk EC2 instance as shown in the below figure

**Graphical user interface, text, website

Description automatically generated**

2) The containers are created from docker images as shown in the figure

Text

Description automatically generated

3) The containers are isolated and the resources needed for its creation is explained in the docker compose.yml file GitHub link – <https://github.com/ParvathiRPai/CMPE-272-Final-Project/blob/master/docker-compose.yml>

4) The certificate is created using the certificate authority as shown in the figure -http://relieffund.parvathipai.com/

Graphical user interface, text, application

Description automatically generated

# **Future Work**

* Automate Docker Image and Container creation using Amazon Elastic Container Service.
* Social Login through Facebook and/or Twitter.
* Okta SSO to Tableau online server and display monthly donations made through our platform.
* Create S3 bucket lifecycle based on charity registration date-time. If charity is not approved by Admin for 15 days move the image uploaded while registration to Standard IA S3 bucket and after 30 days move to S3 Glacier. After 60 days delete it permanently.

# **References**

* <https://www.downtoearth.org.in/news/climate-change/more-than-200-natural-disasters-across-world-in-1st-half-of-2020-72445#:~:text=At%20least%20207%20natural%20disasters,all%20regions%20except%20the%20Americas>.
* https://www.freelogodesign.org/
* <https://mermaid-js.github.io/mermaid/>
* <https://reactjs.org/>
* <https://spring.io/>
* <https://www.w3schools.com/>
* https://developer.paypal.com/developer/accounts/
* <https://developer.okta.com/code/react/okta_react_sign-in_widget/>
* <https://www.tutorialspoint.com/aws_lambda/aws_lambda_using_lambda_function_with_amazon_s3.htm>