Cloud Computing

Assignment 3

FORUMSBOOK

Name:

Minjin Chou

Chatchapat Dechathaweewat

Student number:

S3641315

S3679216

Date of report:

09/06/2021

Table of Contents

Signed Contribution sheet

Student Name: Minjin Chou	**Student Name:** Chatchapat Dechathaweewat
Student ID: s3641315	**Student ID:** s3679216
Contributions: 1. User authentication with Cognito 2. Email notification 3. SMS notification 4. Elastic Beanstalk deployment 5. CSS Styling with Bootstrap 6. DocumentDB 7. Writing report	**Contributions:** 1. Architecture diagram 2. API Gateway 3. Create AWS Lambda functions for DynamoDB integration 4. Amazon S3 5. Writing report 6. Preparing presentation slides
Contribution percentage: 50%	**Contribution percentage:** 50%
By signing below, I certify all information is true and correct to the best of my knowledge. Signature: Minjin Chou Date: 12/06/2021	**By signing below, I certify all information is true and correct to the best of my knowledge.** Signature: Chatchapat Dechathaweewat Date: 12/06/2021

Links

Deployed project:

Github repository: https://github.com/kuntakinte777/Cloud-A3-Project

Summary

The purpose of our web application is to create a social media platform that allows users to freely express and share their daily moments and thoughts throughout the day. We hope to create a social platform where a user could just post and express their ideas freely in a judgemental free environment. We have implemented multiple cloud services from aws that further convenient us in developing our application. Our application allows users to engage, post as well as modify their own posted media.

Introduction

For this application, we have used AWS which provides us APIS, basic abstract technical infrastructure and basic distributed computing tools [1]. They provide users a range of resources, databases as well as services that web developers can utilise to deploy, manage and sustain their web applications in a single setting. Our project we have heavily utilised cloud computing services provided by aws. We have utilised aws elastic beanstalk for our cloud deployment and have also made use of serverless services such as lambda, api gateway and dynamoDB, the list of services we have used are stated below [2].

The motivation behind our idea is to essentially create a healthy social media platform fueled with positivity for our users. Social media platforms in today's era such as facebook and twitter allows users to express, post and share moments of their lives freely. However, due to the vast amount of freedom granted by these platforms, it also promotes online toxicity that is very destructive towards a user's mental health. Over the years, we have witnessed user's who are victim of cyberbullying from these platforms commited suicide or went into deep stages of depression which takes a huge toll on the users lives whether it being physically or mentally or both. Taking this into account, we have been motivated to create a social media platform which only promotes positivity for our users.

Our application allows users to freely express their feelings, thoughts, activities and moments whenever they want but without the toxicity that is provided in mainstream social media services. Very much like twitter, our application allows users to sign up for an account, and will be prompted for verification upon doing so utilising aws cognito. A user then will be able to log in with the verified account and be directed to the main page. On our main page, users are able to post their thoughts and moments while being able to view other user's shared media. However, to reduce toxicity, we have created a like function where users are able to like other user's posts if they find it interesting or generally like their content, comments are disabled to prevent hate speech or cyberbullying. If a user essentially dislikes a content posted by other users, they could only keep it to themselves, therefore each user is only receiving likes based on the contents that they posted. Each time a user likes another user's content, an email (SES) will be sent to the related user informing them their post has been liked. This promotes positivity as throughout the day, a user who constantly receives like notification will feel better about themselves. Besides that, a user is able to modify their password, and each password changed, a user will receive a text message (SNS) informing them their password has been changed.

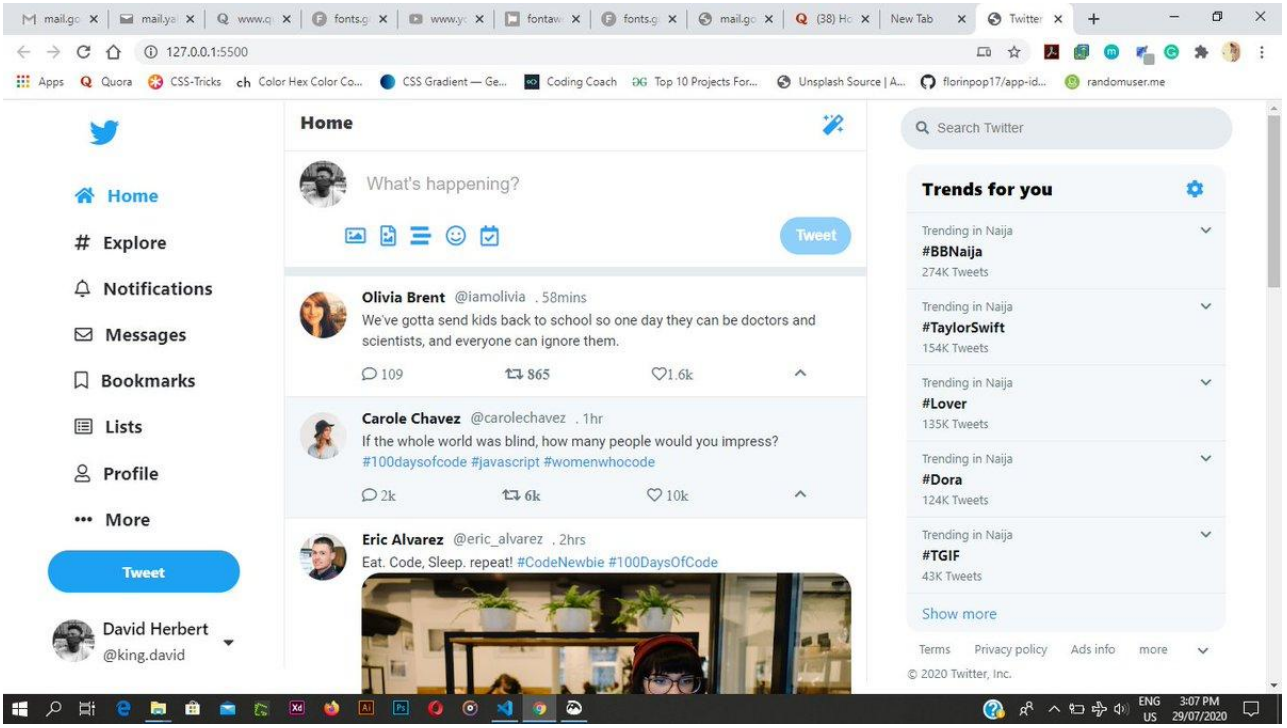
This app is required as social media plays an important role in the daily life of individuals in today's era. Majority of our world today has gone 'online' and taking this into account, users need a space where they can interact with other users from different parts of the world. It can be used in real-life as a socializing application, as we humans are social creatures [3] and this provides a space where we are able to socialise with one another virtually.

Our application has been created with specific users in mind, many introverted and shy individuals are often afraid and scared to voice their opinions. The advantages of our application is, by creating a positive and toxic free environment, this provides a safe space for these individuals where they can voice and give their opinion without worrying about receiving hate speech from other users.

Related application

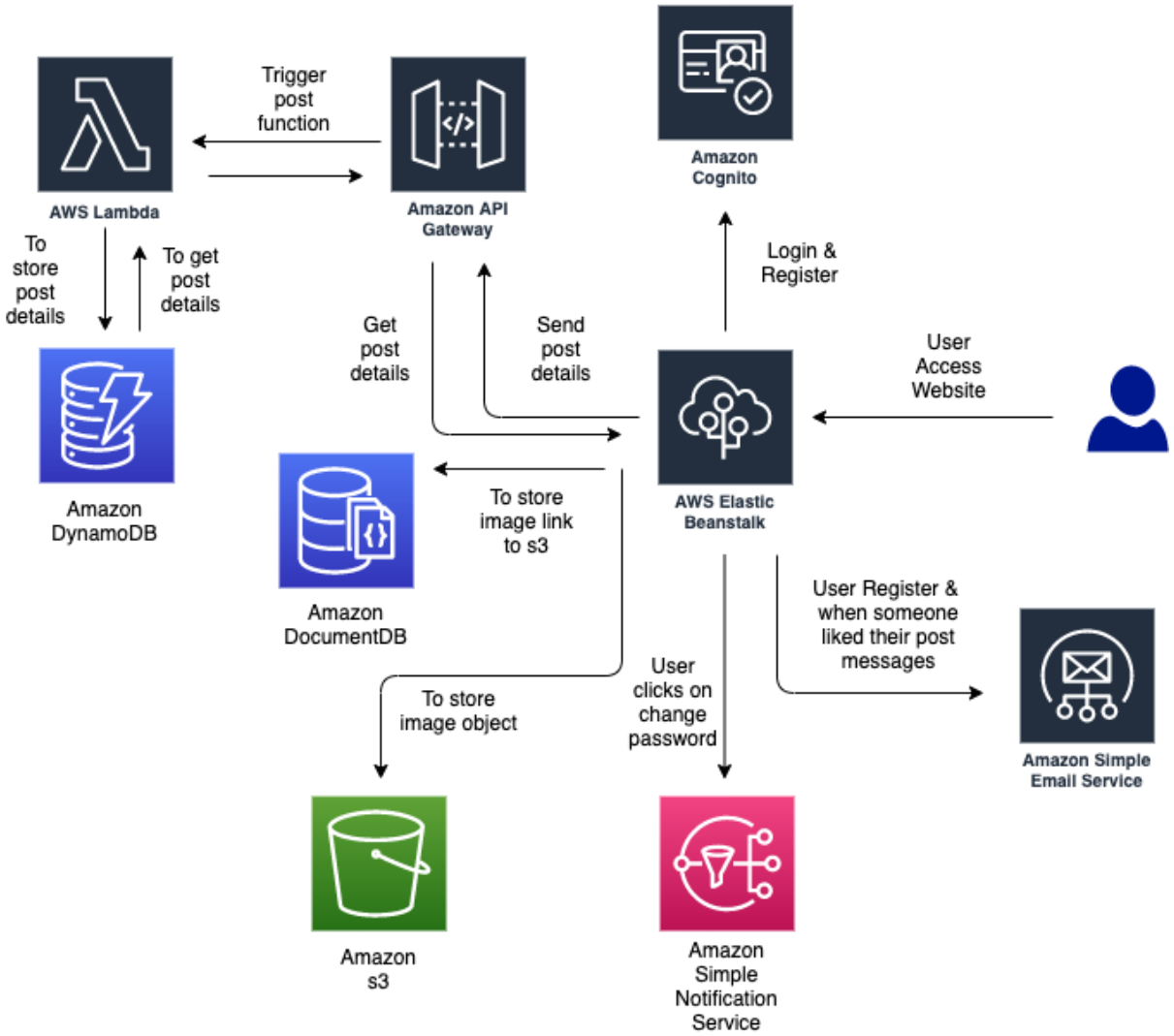
Our application is similar to a specific social media application, Twitter.

Twitter



Twitter is a social media application that allows users to send and receive short posts and messages called tweets [4]. Each tweet is limited to only 140 characters long. Twitter is a majorly popular social platform that is often used by politicians, celebrities, influencers and academic pages to engage with their fans as well as keeps their audience posted about regular updates. Twitter is also used by the general public to follow and stay up-to-date with the latest news and information. It is also used by individuals to promote, share, give feedback to others and express their opinions about a certain matter. Twitter also has a retweet function which allows users to retweet other people's opinions and tweets to be shared as their own, which further promotes engagement and reaches a higher number of audience.

System Architecture



Our application is developed using python flask and is rendered using html and css. Final and testing versions of our application are deployed to elastic beanstalk using elastic beanstalk command line interface. Based on the diagram above, this shows how each of our services communicates with one another.

Elastic Beanstalk

We have used elastic beanstalk for our deployment to the cloud. It allows us to upload our code of our application with environment configurations and will automatically provision and deploy appropriate resources required in AWS to operate our application. Elastic beanstalk auto includes other services such as ec2, auto scaling, elastic load balancing and aws health monitoring service cloudwatch [5].

Cognito

This is a simple user identity and data synchronization service offered by aws [6]. It allows us to authenticate each of our new users and it allows us to add authentication methods based on our needs. It is aws backend solution to handle identity management.

S3

This is an Amazon Web Services S3 which we could call a Simple Storage Service, this is a highly-scalable and secured service in the cloud. The data is stored as objects in buckets, and is able to store an unlimited amount of data. This service is used in our application to store the image object.

DynamoDB

DynamoDb is hosted NoSQL database offered by Amazon Web Services database system. This service is designed for massive scalability with their automatic partitioning model as data volumes grow. We used this service in our application to store all the forum-post messages along with the username of the user who posted the message.

Simple Email Service (SES)

This service allows web owners or developers to send an email to their user. In order for users to use this feature, users must register their email address before receiving a notification email when someone likes their posts as an email must be registered into a sandbox and be verified before use.

Simple Notification Service (SNS)

SNS is a service offered by aws. It is a fully managed service for messaging for application to person communication [7]. Using the mobile number submitted by the users, when a user changes their password, a message will be sent to their related mobile number informing them regarding the performed action.

Lambda Functions

Lambda function is a serverless service that allows us to run code based on triggered events and automatically provides the required computation resources to execute the code. Our lambda function is triggered when a user creates a new post. The post will trigger the lambda function to store the newly created post into dynamoDB. We have two lambda functions, another is used to retrieve posts from the dynamoDB to be displayed in the main page.

Api Gateway

The aws api gateway is a fully managed service that allows us to create and publish secure api for our application [8]. We have created a RESTful APIS that enables us to establish a two way communication of our application. We have used two api gateways, one is for post request and the other is for get requests which in turn will trigger our lambda functions above.

DocumentDB

This is a database service provided by aws to fully- manage JSON format data at scale. It is an enterprise ready document database [9]. We have used it to store each of our images links separately from dynamoDB to prevent mixture of data.

Developer Manual

This part explains a step-by-step guide to create a ForumBook application on a local machine. Before proceeding, ensure you have installed Python 3 (not version 2) AWS SDK, AWS CLI, and Docker.

Configure AWS

The following file is required to be used with AWS CLI. Firstly go to the ~/.aws file and modify the credentials and config file as of below.

~/.aws/credentials

```
[default]
aws_access_key_id = AKIAUNWVEV4RSI4D4PN5
aws_secret_access_key = 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
```

~/.aws/config

```
[default]
region=us-east-1
```

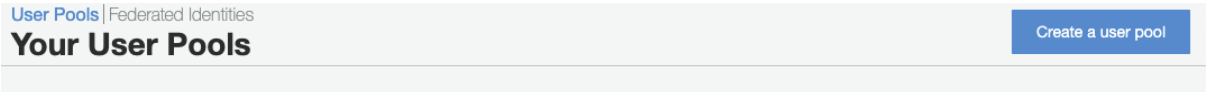
Gave IAM roles permission to work with my teammate

User name	Groups	Access key age	Password age
kuntakinte777	assignment2 and lambda-a3	47 days	None
north_j	None	8 days	None
north23	None	None	10 days

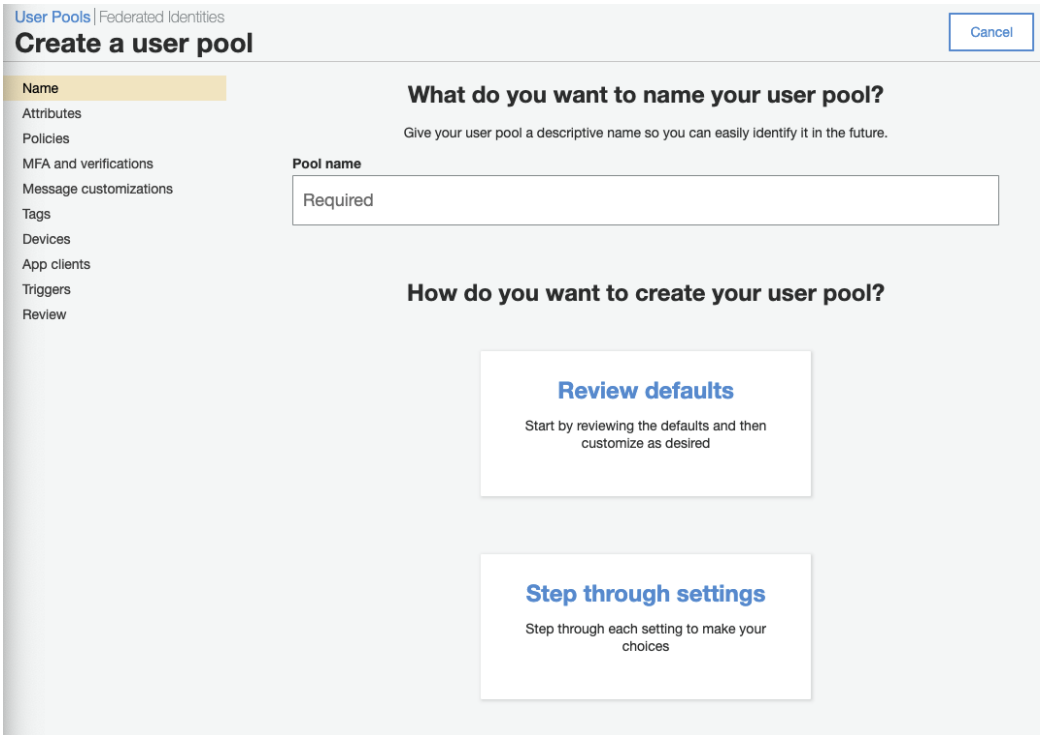
Created roles to be used with lambda functions and other aws services

Role name	Trusted entities	Role ARN	Last activity
aws-elasticbeanstalk-ec2-role	**AWS service:** ec2	arn:aws:iam::304315936547:role/aws-elasticbeanstalk-ec2…	Today
aws-elasticbeanstalk-servic…	**AWS service:** elasticbeanstalk	arn:aws:iam::304315936547:role/aws-elasticbeanstalk-ser…	Today
AWSServiceRoleForAPIGat…	**AWS service:** ops.apigateway (Service-Linke…	arn:aws:iam::304315936547:role/aws-service-role/ops.api…	None
AWSServiceRoleForAutoSc…	**AWS service:** autoscaling (Service-Linked role)	arn:aws:iam::304315936547:role/aws-service-role/autoscal…	Today
AWSServiceRoleForElastic…	**AWS service:** elasticloadbalancing (Service-…	arn:aws:iam::304315936547:role/aws-service-role/elasticlo…	Yesterday
AWSServiceRoleForOrganl…	**AWS service:** organizations (Service-Linked r…	arn:aws:iam::304315936547:role/aws-service-role/organiz…	None
AWSServiceRoleForRDS	**AWS service:** rds (Service-Linked role)	arn:aws:iam::304315936547:role/aws-service-role/rds.ama…	Yesterday
AWSServiceRoleForSupport	**AWS service:** support (Service-Linked role)	arn:aws:iam::304315936547:role/aws-service-role/support.…	None
AWSServiceRoleForTruste…	**AWS service:** trustedadvisor (Service-Linked …	arn:aws:iam::304315936547:role/aws-service-role/trusteda…	None
Forumbook-SMS-Role	**AWS service:** cognito-idp	arn:aws:iam::304315936547:role/service-role/Forumbook-…	None
getPostFunction-role-s6tq9…	**AWS service:** lambda	arn:aws:iam::304315936547:role/service-role/getPostFunc…	None
lambda-api-gateway-role	**AWS service:** lambda	arn:aws:iam::304315936547:role/lambda-api-gateway-role	Today
postfunction-role-ex2hj7ds	**AWS service:** lambda	arn:aws:iam::304315936547:role/service-role/postfunction-…	None
signupfunction-role-1ko2lp14	**AWS service:** lambda	arn:aws:iam::304315936547:role/service-role/signupfuncti…	5 days

Creating User Pool



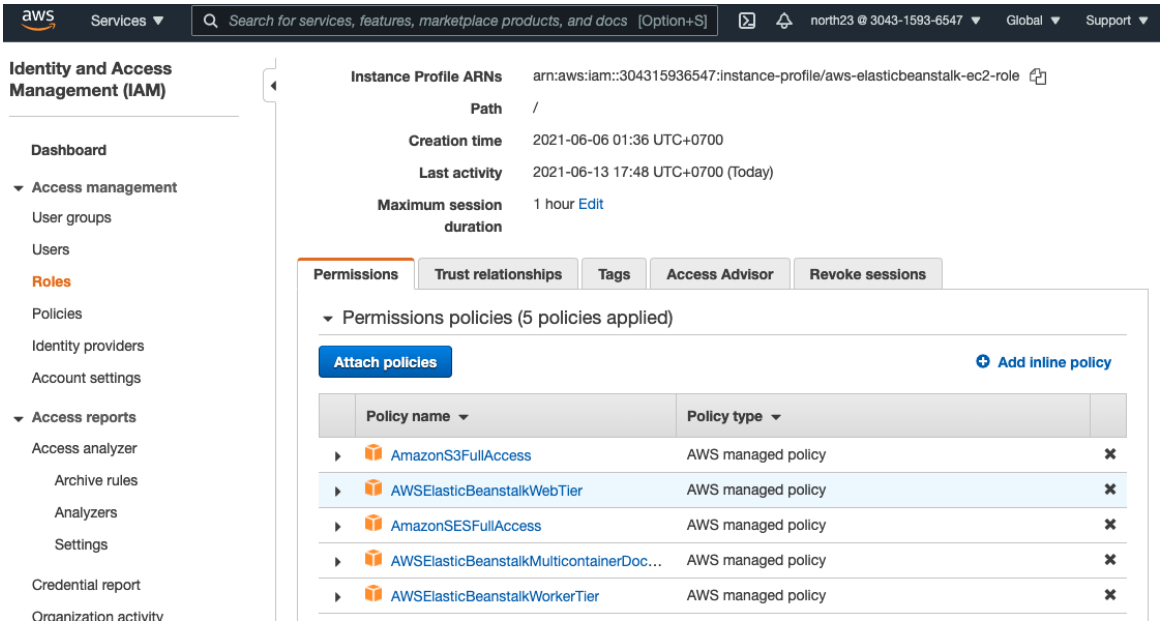
By clicking “Create a user pool” it will bring you to this page, you can specify a pool name that you desire and click on review defaults and click create pool.



Update .env file in the project root directory (create a new one if not exists) according to the user pool configuration object:

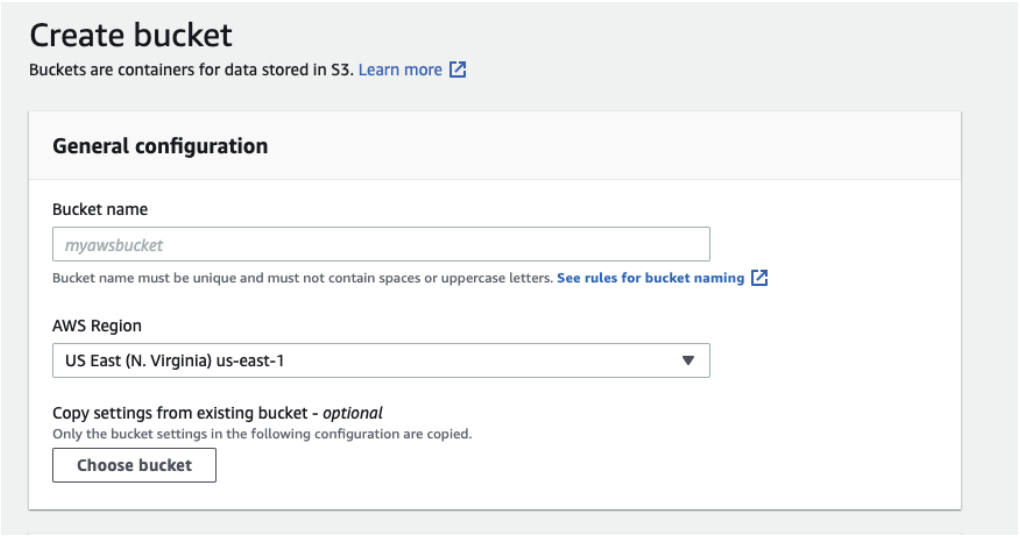
```
USER_POOL_ID=us-east-1_Gd3Licbad
CLIENT_ID=1jvdcfk3iut0f258vvc8pd8p2l
USER_POOL_NAME=User
```

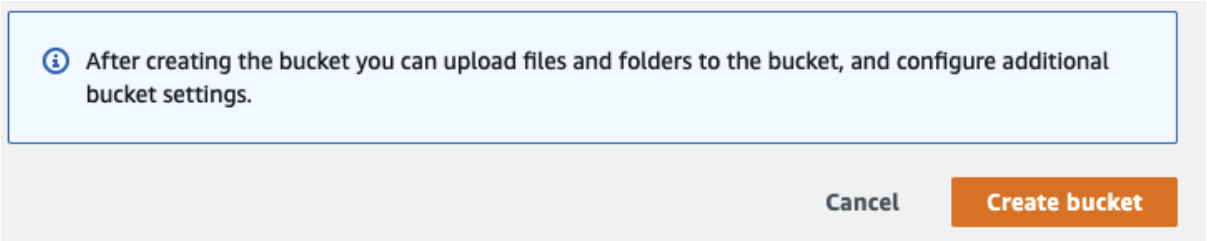
Giving elastic beanstalk access



Creating S3 Buckets

In our application we create S3 buckets through aws console as shown below by clicking “Create Bucket” you can specify the name you want to use for this bucket by filling the bucket name and choosing the region that you want.





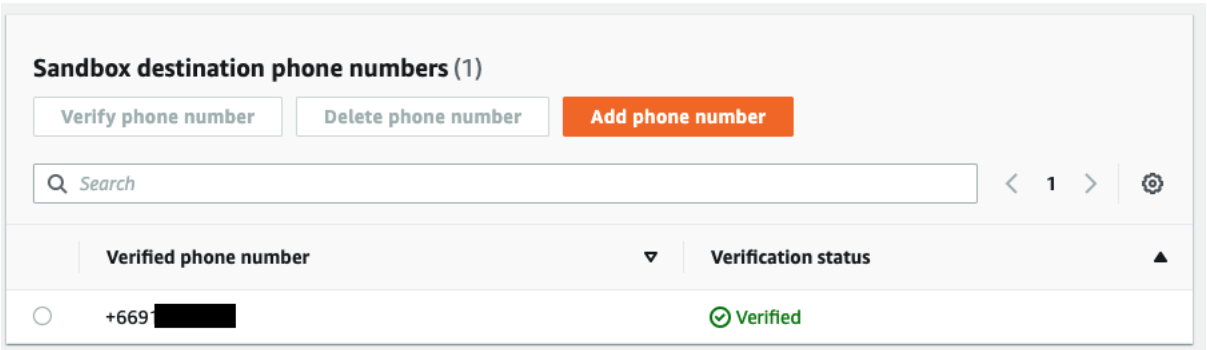
To update the bucket name, just change BUCKET_NAME on the file called util.py and replace “assignment03bucket” with your desired name.

```
REGION_NAME = "us-east-1"
BUCKET_NAME = 'assignment03bucket'
TEMP_PATH = '/tmp/'

DB_POST_COLLECTION = 'posts'
```

```
    # Upload the file
    try:
        response = self.s3.upload_file(file_path, BUCKET_NAME, object_name)
        self.logger.debug('S3 response')
        self.logger.debug(response)
        return object_name
    except ClientError as e:
        self.logger.error(e)
        raise e
```

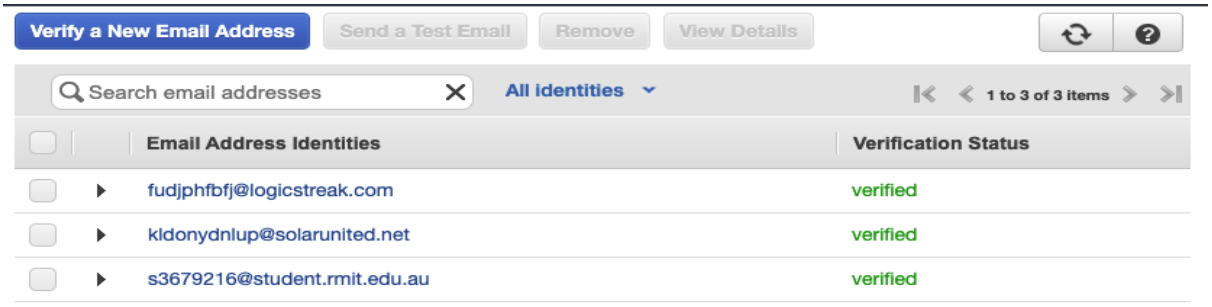
Configure SNS

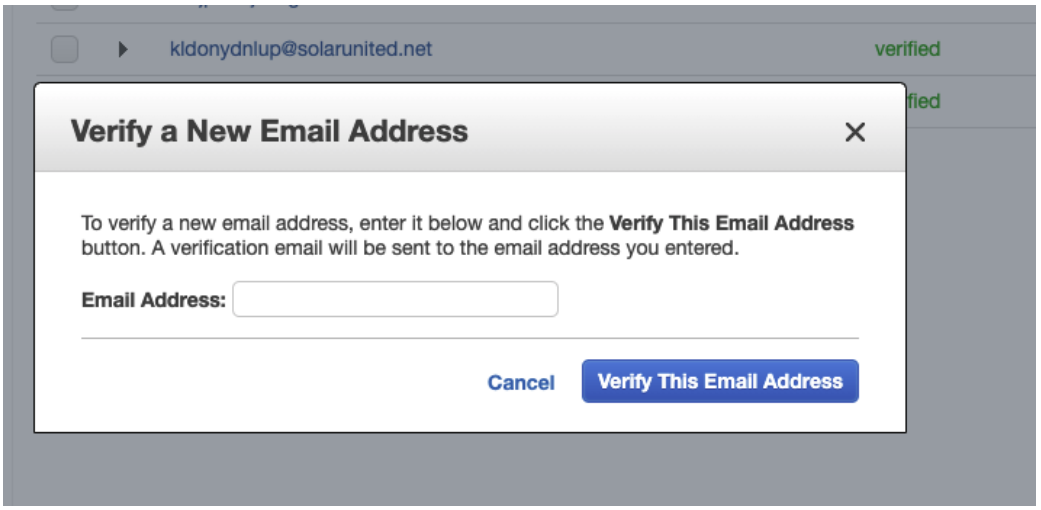
To create an SMS service, navigate to Simple Notification Service and click Text messaging (SMS) on the left menu. Since our accounts are placed in the Amazon SES sandbox, phone number must be verified first in order to send text messages as shown below:



Configure SES

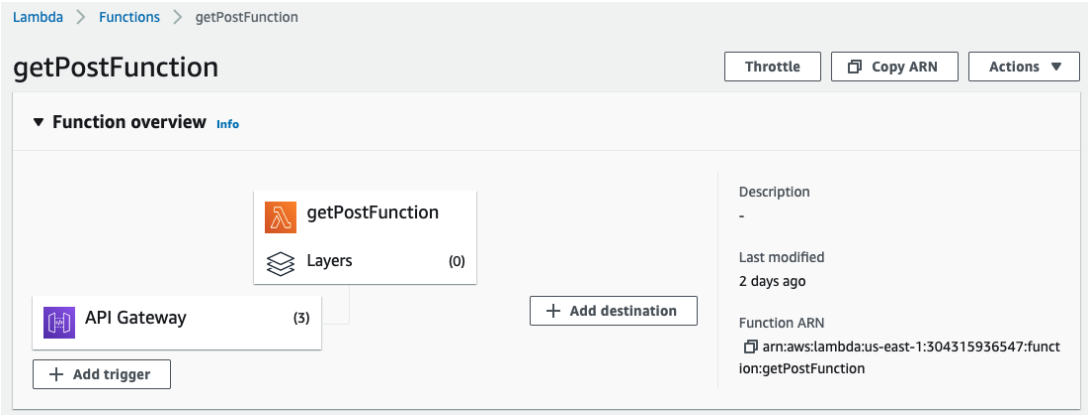
To help prevent fraud and abuse, the new Amazon SES accounts are placed in the Amazon SES sandbox so we cannot send an email to an unverified email. To use SES user must “Verify a New Email Address” then filled in the email and click “Verify This Email Address”

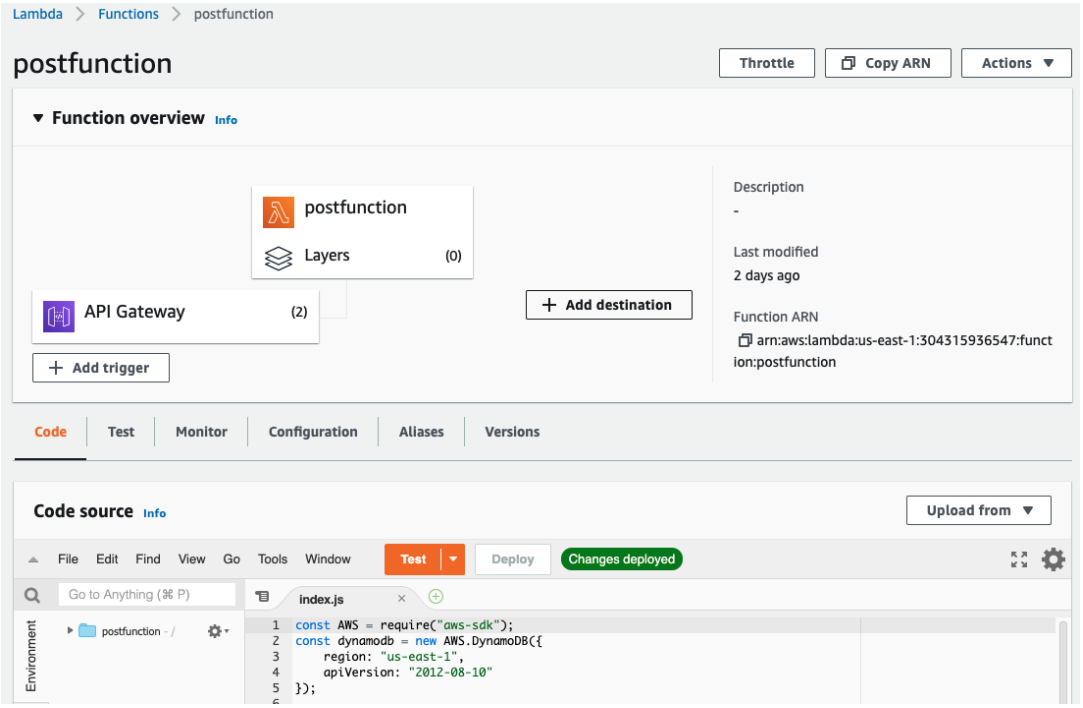




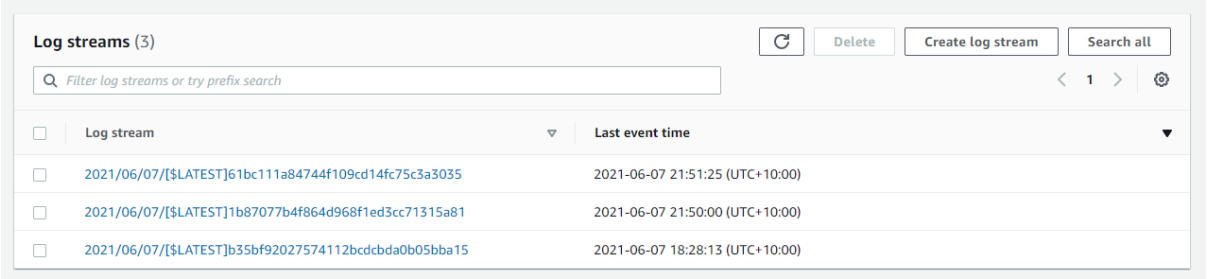
Creating Lambda Functions

We have two lambda functions, one for getting a post detail and another one for creating a new post. These two lambda functions have direct access to DynamoDB.





Using cloudwatch to monitor logs of errors/activities



Development

Go to project root directory and start MongoDB Docker container:

```
docker-compose up -d
```

To run the website on local machine (localhost), execute the following commands:

```
# Execute this command only once to create Python virtual environment
python3 -m venv env
# Install all dependencies
pip3 install -r requirements.txt
# Use virtual environment
source venv/bin/activate
# Start application
python3 main.py
```

The complete environment variables needed for local development:

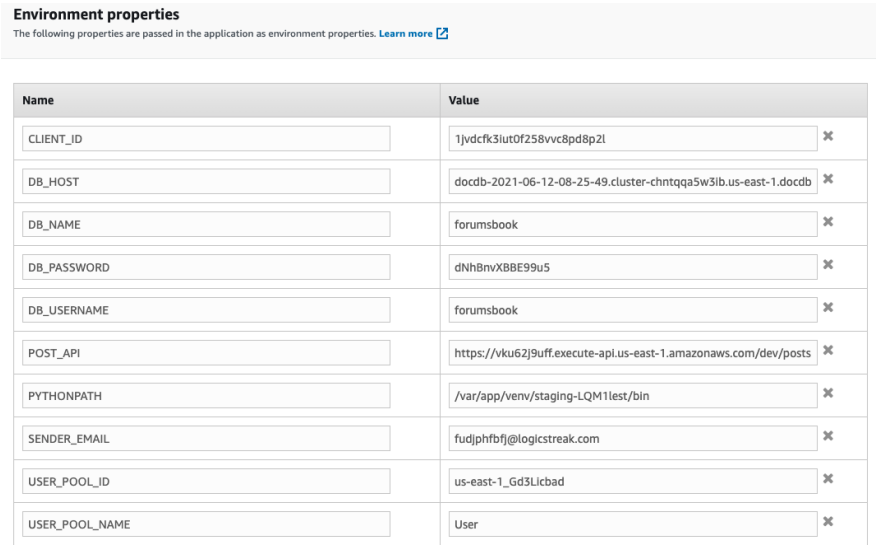
```
DEBUG=true

POST_API=https://vku62j9uff.execute-api.us-east-1.amazonaws.com/dev/post
s
REGION_NAME=us-east-1

# Cognito
USER_POOL_ID=us-east-1_Gd3Licbad
CLIENT_ID=1jvdcfk3iut0f258vvc8pd8p2l
USER_POOL_NAME=User

# Database
DB_HOST=127.0.0.1
DB_NAME=admin
DB_USERNAME=root
DB_PASSWORD=example

# Email
SENDER_EMAIL=fudjphfbfj@logicstreak.com
```

Deployment

To deploy the project to Elastic Beanstalk run:

```
eb deploy
```

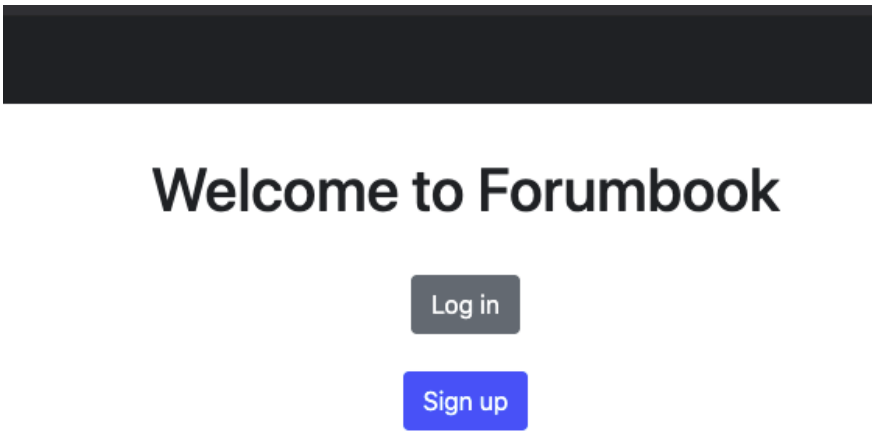
Update environment variables in Elastic Beanstalk environment by navigating to **Elastic Beanstalk > Environments > Forumsbook-env-1 > Configuration:**



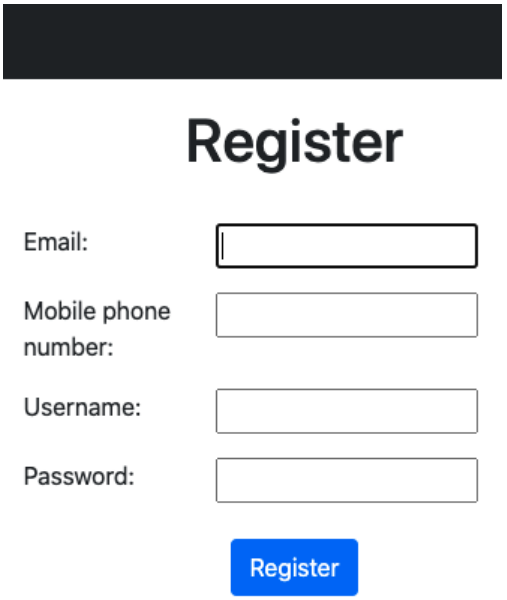
User Manual

Login & Register Page

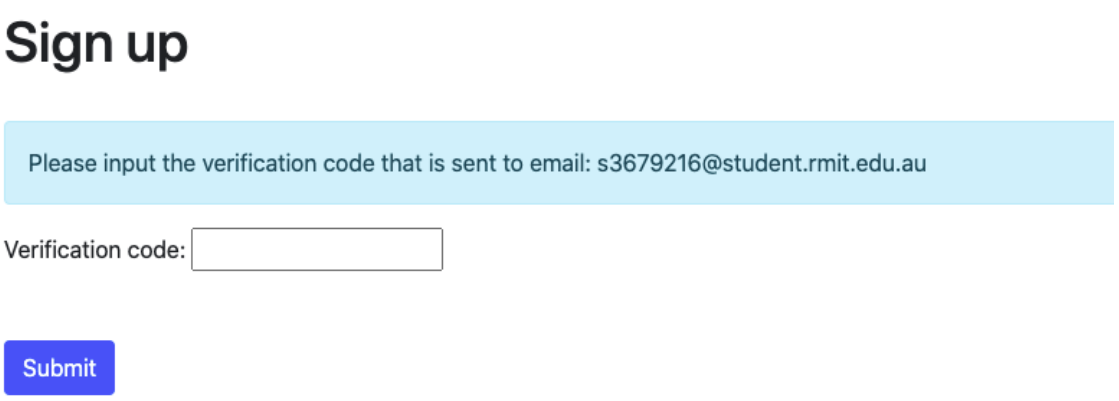
Before logging in, a user is required to sign up for a new account.



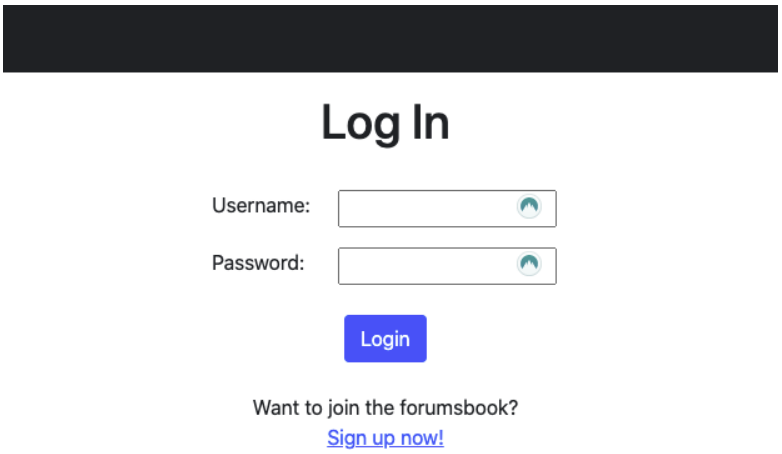
Fill in all the forms and click the Register button, make sure email has not been registered.



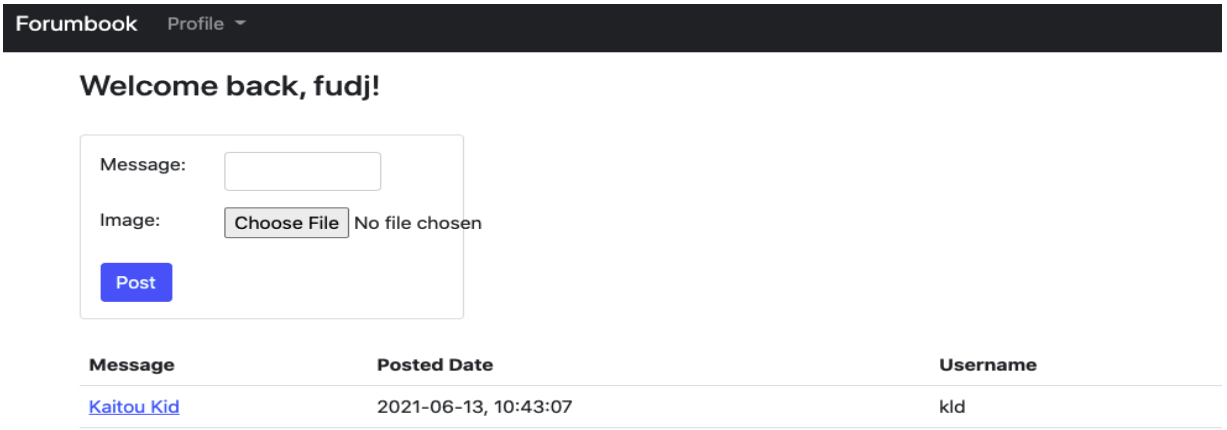
Please enter a verification code that has been sent to your email address.



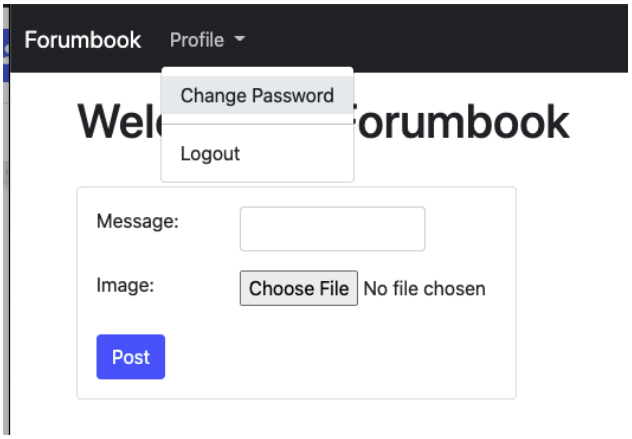
After signing up fill in the form to login



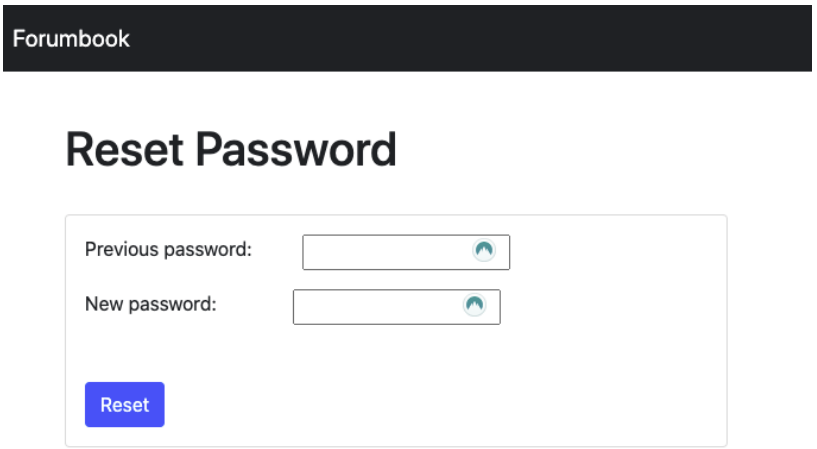
This is the main page where users can post messages, search messages and view messages.

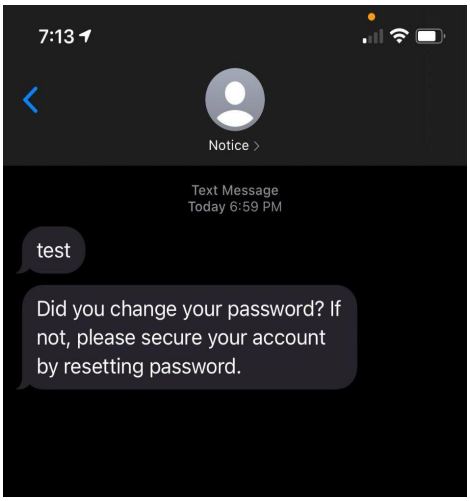


User can click change password to navigate to reset password page

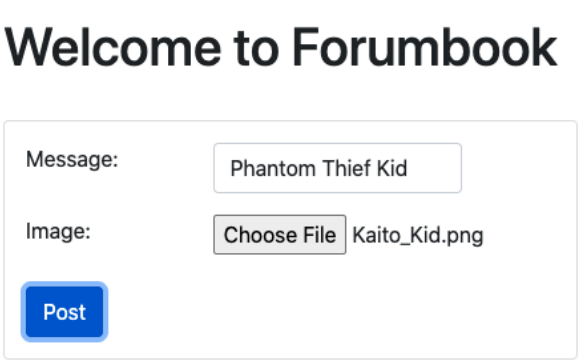


By filling a previous password and the new password, the user will received a sms message informing them that the password has been changed.

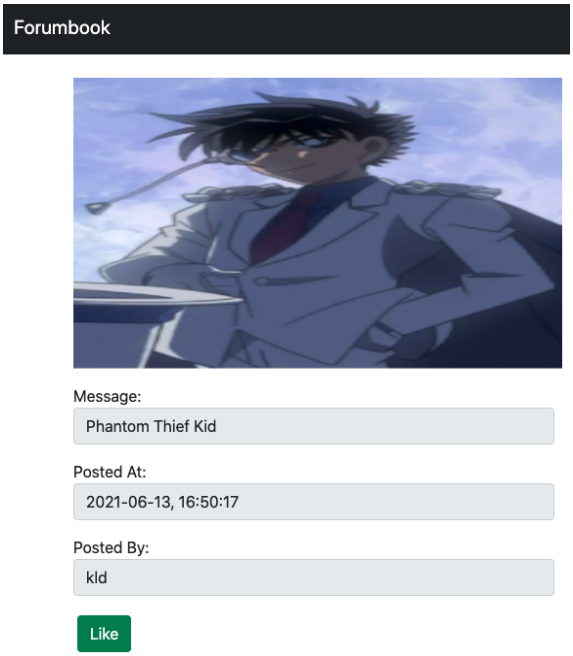


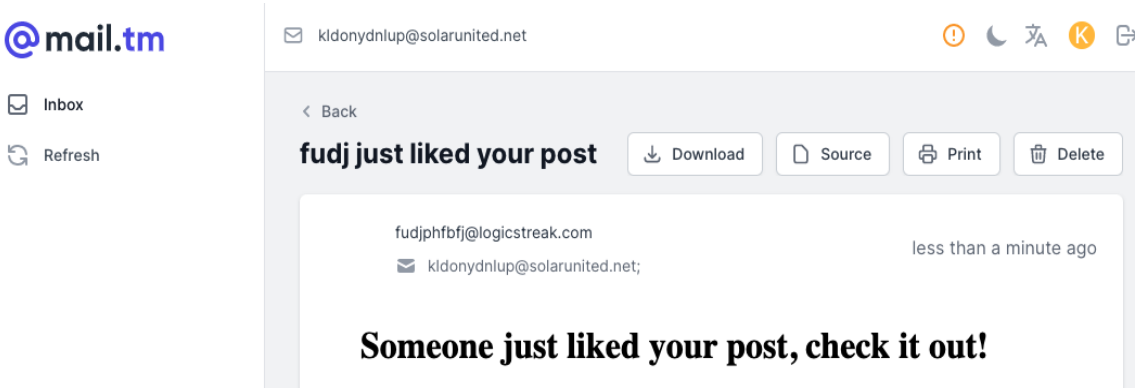


Posting a message with a picture



View message Page
Users are able to view and like other users' posts and like their message, an email will be sent to that user once their post has been liked.





References

Articles used in report:

[1] https://en.wikipedia.org/wiki/Amazon_Web_Services
[2] https://aws.amazon.com/serverless/
[3] https://www.nature.com/articles/s41562-018-0389-1
[4] https://esrc.ukri.org/research/impact-toolkit/social-media/twitter/what-is-twitter/
[5] https://cloudacademy.com/course/deployment-orchestration-aws-elastic-beanstalk/aws-elastic-beanstalk/#:~:text=AWS%20Elastic%20Beanstalk%20is%20a,make%20the%20web%20application%20operational.
[6] https://aws.amazon.com/about-aws/whats-new/2014/07/10/introducing-amazon-cognito/#:~:text=Amazon%20Cognito%20is%20a%20simple,users%20across%20their%20mobile%20devices.&text=Amazon%20Cognito%20is%20available%20to%20all%20AWS%20customers.
[7] https://aws.amazon.com/sns/?whats-new-cards.sort-by=item.additionalFields.postDateTime&whats-new-cards.sort-order=desc
[8] https://aws.amazon.com/api-gateway/
[9] https://aws.amazon.com/documentdb/?trk=ps_a134p000006BoZjAAK&trkCampaign=pac_q1_0320_paidsearch_documentdb_au_DocumentDB_br_prosp_en&sc_channel=ps&sc_campaign=pac_q1_2020_paidsearch_documentdb_au&sc_outcome=PaaS_Digital_Marketing&sc_geo=APAC&sc_country=AU&sc_publisher=Google&sc_category=Database&sc_detail=documentdb&sc_content=DocumentDB_exact&sc_matchtype=e&sc_segment=481116460280&sc_medium=PAC-PaaS-P|PS-GO|Brand|All|PA|Database|DocumentDB|AU|EN|Text&s_kwcid=AL!4422!3!481116460280!e!!g!!documentdb&ef_id=CjwKCAjw2ZaGBhBoEiwA8pfP_uDOipFQYhJsQTRIwWqTMVSkEgpUSslwGEv1gO0OnY58p2ABz3SrCBoCSAoQAvD_BwE:G:s&s_kwcid=AL!4422!3!481116460280!e!!g!!documentdb

Documentations used to develop the application:

Welcome to Flask — Flask Documentation (2.0.x) (palletsprojects.com)
pycognito · PyPI
python-dotenv · PyPI
Boto3 documentation — Boto3 Docs 1.17.93 documentation (amazonaws.com)
amazon s3 - python AWS boto3 create presigned url for file upload - Stack Overflow
PyMongo 3.11.4 Documentation — PyMongo 3.11.4 documentation
Welcome to PyJWT — PyJWT 2.1.0 documentation
Python strftime() - datetime to string (programiz.com)
Moving out of the Amazon SES sandbox - Amazon Simple Email Service Classic
formatting - How can I get code syntax highlighting in Google Docs? - Web Applications Stack Exchange
python - secret key not set in flask session, using the Flask-Session extension - Stack Overflow