# Pool & Save

—

Team

Indraneel Ray, Hemanth Teja, Durga Prasad, Vivek Kumar

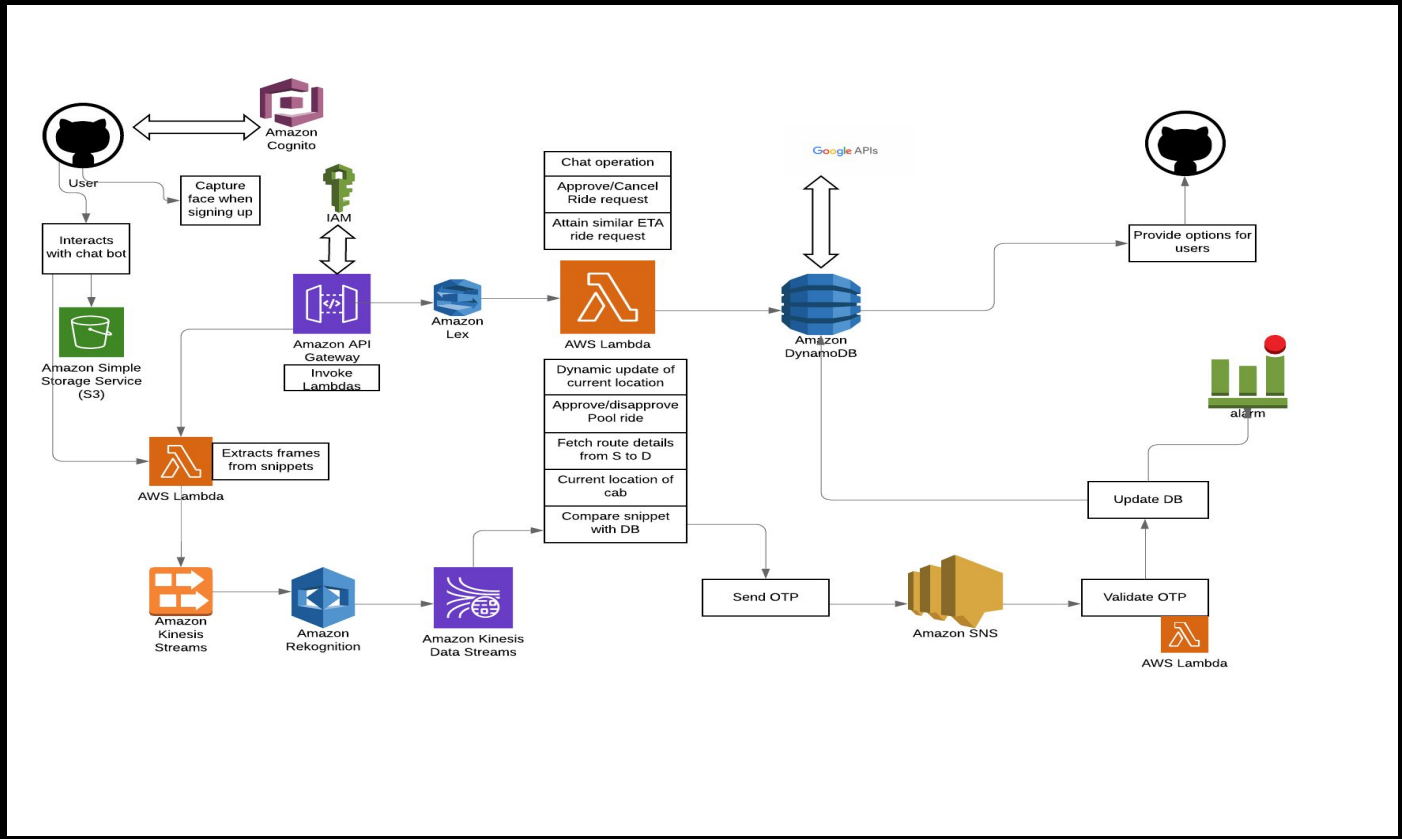
New York University

# Problem Statement

- The goal of this application is to show how the idea of a car pool system can be hosted on a powerful cloud computing platform such as the Amazon Web Services.
- We have architected and constructed a complete web-based application to provide car pooling services.
- Various different AWS services such as- Rekognition, Lex, Api gateway, S3, SQS, SNS, SES, IAM, Cloudwatch, etc are used to host this web based application.

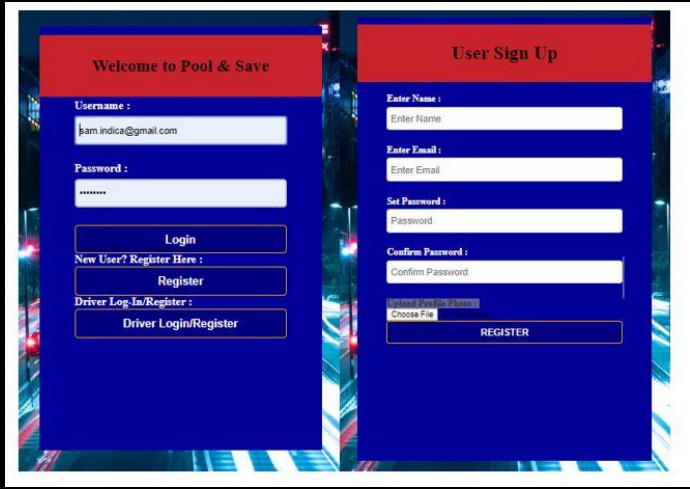
# Architecture of AWS Services Used

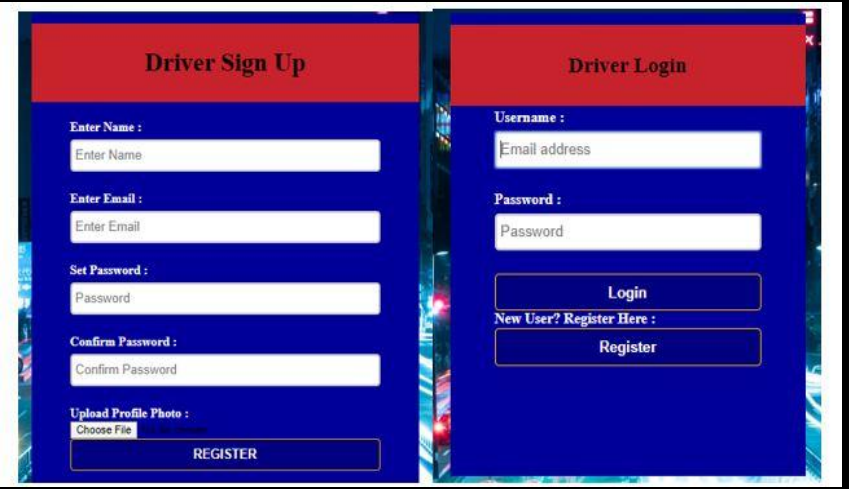
1



# AWS Cognito

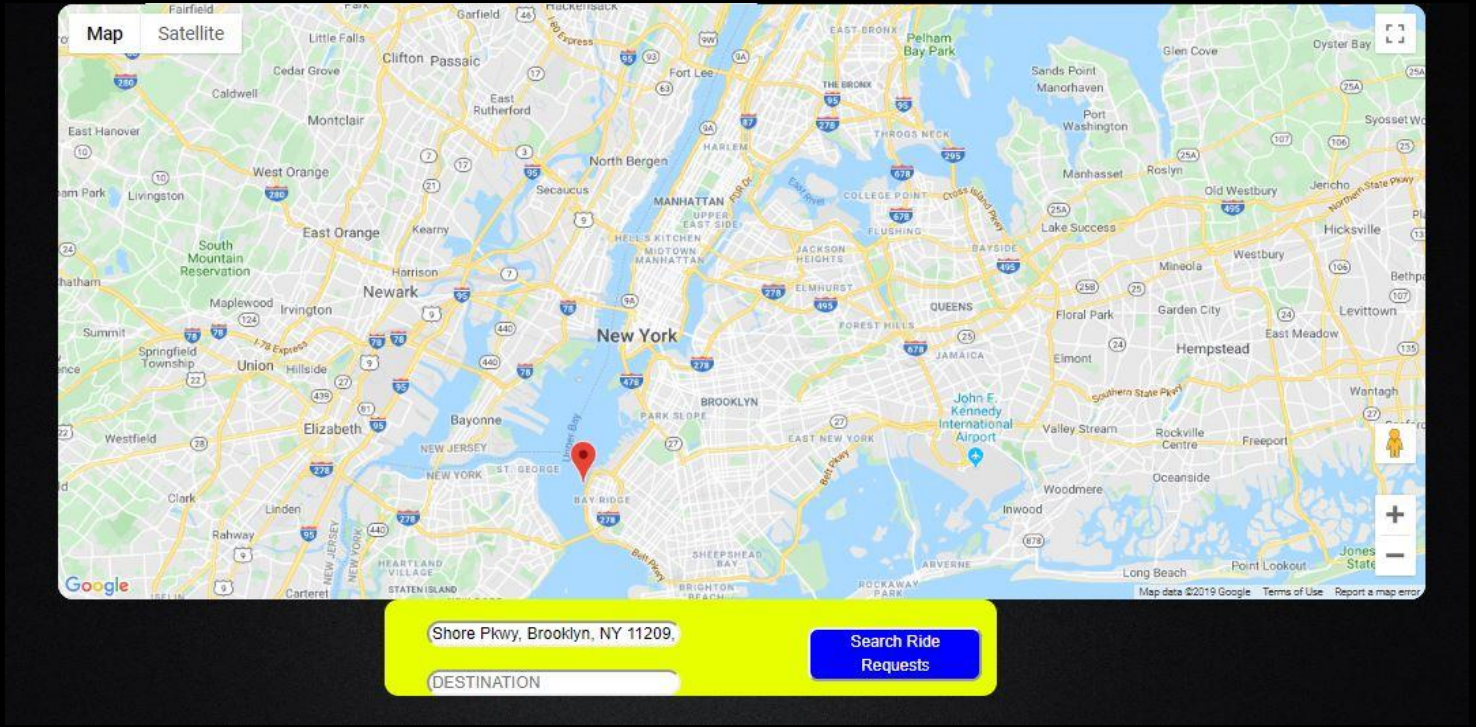
- A userpool consisting of all the riders is hosted on Cognito.
- A driverpool consisting of all the drivers is hosted on Cognito.
- An identity pool is generated for both the pools.



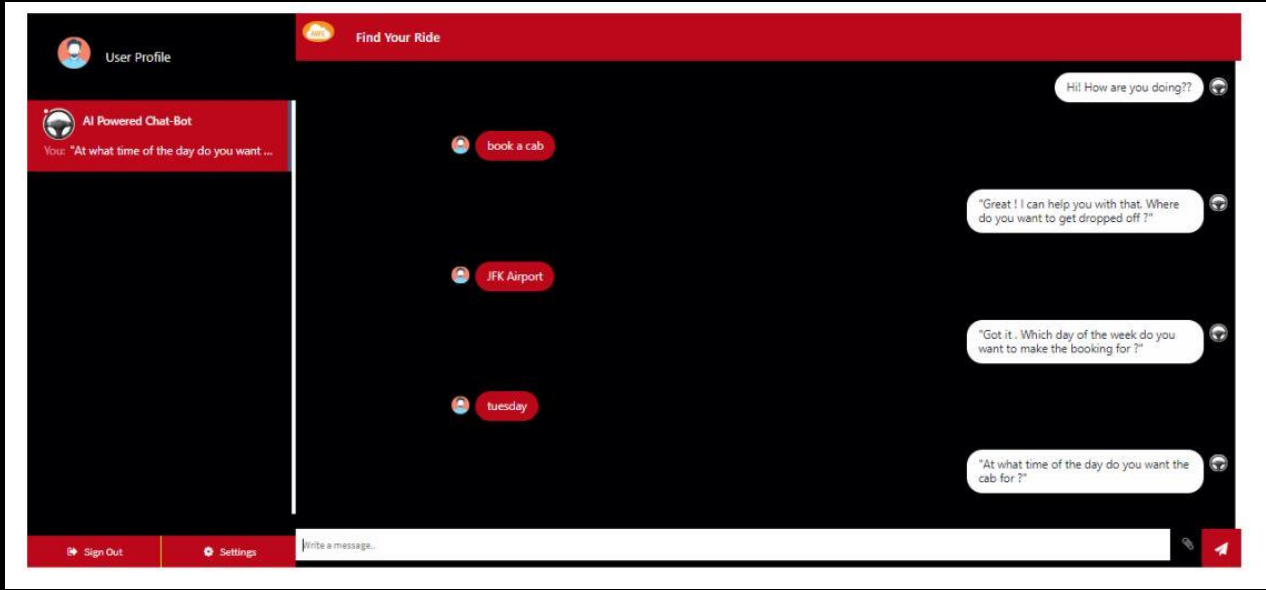
# User and Driver Profiles


# Functionality for driver to update current location



# AWS Lex

- A lex chatbot is hosted to satisfy the user requirements for travel. It queries the information about destination, date and time, number of people who will accompany the user in his ride.



# AWS API Gateway

- Multiple API Gateways are setup to ensure the cohesive integration of multiple AWS services used. API gateway ensures the reliable processing of every API call.  API Gateways that are set up are:
- API gateway to post images from the front-end to S3 bucket. The S3 bucket link of the image is stored in the cognito data.
- API gateway to interact between lex and the front end input data.
- API gateway to extract data of the current users from the users.
- API gateway to confirm the ride by the driver. The API gateway interacts between dynamoDB, SQS and lambda.

# AWS DynamoDB

- Rider - This DB is used to store all the user(rider) details. Details of the rider's current location , destination, date and time of travel, etc are stored in the DB.
- Driver - This DB is used to store data about the driver. Details such as driver location are stored.
- Rides - This DB is used to store all the details of the current and past rides.

# AWS Rekognition

- Rekognition is used to compare the image of the user that was during sign up and the current live image that is extracted by the camera through front-end functionality.

# AWS SQS

- The queue holds the details of the user/ rider requests that are attained every time the user uses lex to make a request. Another queue is used to store the driver data.

# AWS Lambda

- Chat operation with user through Lex
- Approve/Cancel ride request
- Fetching Route details from source to destination
- Fetching current location of the cab
- Comparing the front-end user image with the images uploaded on S3- using Rekognition
- Attaining the user image from the front-end
- Validating user when he registers for the first time using SES

# AWS SES

- AWS SES is used to send confirmation mails of the user and driver emails - that are inputted during the signup process.

# AWS S3

- Front-end for user and driver pages are hosted on S3.
- The pictures of users and drivers are uploaded to buckets.
- Front-end for Lex is on S3
- Front-end for verifying users using Rekognition is also uploaded in S3 bucket

# AWS CloudWatch

- AWS cloudwatch is used to trigger logs for the various functionalities that are deployed. Logs help in debugging the errors and also verify the responses received from various different microservices.

# AWS IAM

- IAM is used to manage access to AWS services and resources securely. Using IAM, we can create and manage AWS users and groups, and use permissions to allow and deny their access to AWS resources