



CSCI-5410 SERVERLESS DATA PROCESSING
PROJECT DESIGN DOCUMENT
GROUP-6

GROUP MEMBERS:

Name	B00####
Kavan Patel	B00869224
Navya Jayapal	B00886554
Neelansh Gulati	B00892946
Prit Sorathiya	B00890175
Qiwei Sun	B00780054
Samarth Jariwala	B00899380

Course Instructor: Dr. Saurabh Dey

Table of Contents

1. Revised Cloud Architecture	3
2. Serverless and Server-Oriented Technologies	3
2.1 User Management Module.....	4
2.2 Authentication Module.....	4
2.3 Online Support Module.....	5
2.4 Message Passing Module.....	5
2.5 Machine Learning Module.....	6
2.6 Web Application Building and Hosting.....	6
2.7 Other Essential Modules.....	7
3. RoadMap.....	8
3.1 User Management and Authentication Module.....	8
3.2 Online Support Module.....	10
3.3 Message Passing Module.....	13
3.4 Machine Learning Module.....	14
3.5 Web Application Hosting.....	16
3.6 Other Essential Modules.....	17
3.7 Final RoadMap.....	19
4. Module Distribution	21
5. Meeting logs	21
References	27

1. Revised cloud architecture

The architecture of the Serverless B&B application is depicted in Figure 1. The application will be composed of a frontend developed using the React.js framework and a backend built with Express.js, Node.js, and various Amazon Web Services [1] (AWS) and Google Cloud Platform [2] (GCP) Services.

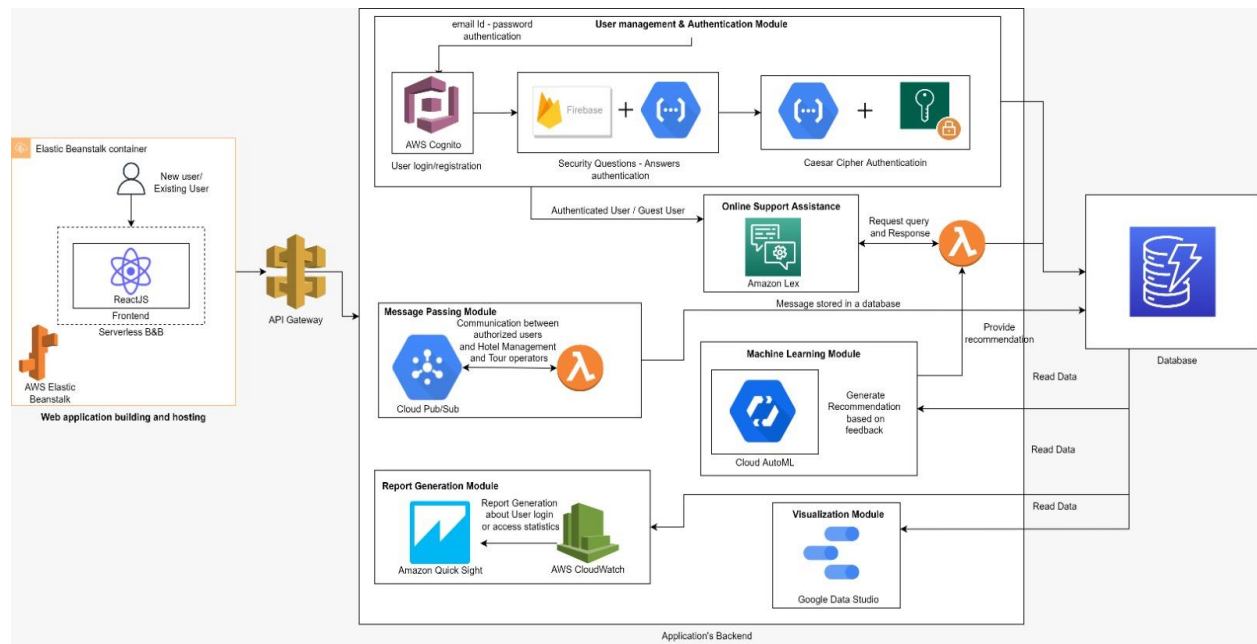


Figure 1: Revised Cloud Architecture

Figure 1 depicts the project's cloud architecture. This project will be a serverless web application, and the front end will be built with the ReactJS framework. We intend to host the application using the Google Cloud Run service. Users can use 3-way MFA to register or authenticate in Serverless B&B. (Multifactor Authentication). We have opted to utilize AWS Cognito for user authentication through ID and password, and GCP Firestore for question/answer verification in order to achieve this feature.

Amazon Lex will be used as virtual support by Serverless B&B. Lex will be able to answer queries from users about room availability, reserving accommodations, ordering meals, and so on. As a result, Amazon Lex will use AWS Lambda in the background to compute query replies. As a result, that lambda function will interact with Amazon DynamoDB and retrieve the necessary information. Furthermore, message forwarding is the application's essential functionality, and we will implement it using Google cloud pub/sub, which functions as a messaging service. All of the message passing, and communication will be saved in AWS DynamoDB.

Serverless B&B will be able to collect customer input and provide analyses based on it. To execute that service, we will utilize Google Cloud AutoML, which accepts data from a

DynamoDB table and delivers prediction or analysis reports automatically by training, deploying, and delivering a machine learning model. Google Data Studio will also be utilized to view the data in real time and produce the Customer booking graph, customer meal orders, and so on. Finally, for report production, we will use Amazon QuickSight, a business intelligence tool. By evaluating the data, it will also create user login or access statistics. Furthermore, API Gateway will be utilized to trigger lambda functions to access various services to construct a bridge between the front-end and back-end.

2. Serverless and Server-Oriented Technologies

2.1 User Management Module

We'll utilize AWS Cognito for user management because it supports security APIs in the User Management Module. It also allows us to create customized email templates for account verification. When a user registers for our program, a customer number is created at random and issued to the registered user. Cognito also synchronizes user data across services. AWS DynamoDB will be used to store and manage user information.

The main reason for using AWS Cognito for user management and AWS DynamoDB is that they both offer a variety of functionalities. Amazon Cognito allows you to manage all of your users in a single location. With the help of AWS Cognito, we can handle user signup, login, forgotten passwords, and other security features such as multi-factor authentication and compromised credentials. You may also construct your own email template [2]. Cognito's capabilities include User Management, Custom email templates, multi-factor authentication, and Identity Pool.

DynamoDB is a NoSQL database accessible on AWS for executing large-scale high-performance applications. Because DynamoDB is a NoSQL database, the data is stored as a key-value pair. It provides simple APIs for accessing the data contained within it [1]. Dynamo DB has built-in security, replication in many locations, and easy backup.

2.2 Authentication Module

AWS Cognito will be used in this module for user ID and password validation. GCP Firestore + CloudFunction will be used for security question and answer authentication, and GCP CloudFunction will be utilized for encryption of sensitive data such as the password and security question response. The Multi-Cloud Model is used in this module. The use of many public cloud providers, known as the multi-cloud approach, is necessary to achieve ultra-high availability.

The main reason we chose AWS Cognito and GCP Firestore + CloudFunction for user data authentication and encryption is because AWS Cognito allows us to create customized email templates for account verification [2]. We can manage user signup, login, and other security

features such as multi-factor authentication and compromised credentials, whereas GCP Firestore, a NoSQL document database, offers a variety of benefits such as easy scalability, built-in live synchronization, security, and data validation. The GCP Cloud function was chosen for authentication because Cloud Functions is a scalable, pay-as-you-go functions as a service (FaaS) platform that allows you to construct and link event-driven applications using simple, single-purpose code. Cloud Functions uses open-source Function as a service framework that runs multiple functions across multiple environments which eventually avoids lock-in.

2.3 Online Support Module

This module focuses on interactions between guests and hotel management, such as when guests want to book rooms, check the availability of rooms, order food from the kitchen, or receive an invoice at check-out. Instead of keeping client invoices in a database, we will be using S3 to store them so they can be conveniently retrieved when needed.

We would keep hotel booking data in DynamoDB, including information about available rooms, booked rooms, kitchen orders, and food menus. Due to DynamoDB's automatic data management, storing of conflicting schema elements, and ease of scaling as needed [7], we will be adopting it. Additionally, we have chosen DynamoDB once more for the user module, so this would be in sync with it.

We'll be leveraging AWS Lex's chatbot capability, which will accept user input for culinary orders and room reservations. Which might include the number of rooms the customer wants to book, the style of accommodation, the length of the stay, as well as offering the consumer food options and guiding them to the kitchen via hotel personnel. Conversational AI solutions for voice and chat are powered by Amazon Lex, an NLP service provided by AWS [8].

2.4 Message Passing Module

Authorized consumers can connect with hotel management and tour operators using this feature. There are many services in our system because we are adopting a server-less design. We must create a message queue to hold the events in order to communicate and interact with one another's services. When only one user sends requests, it is simple. However, it becomes challenging to manage the message queue in parallel or concurrent requests as the number of users rises. To address the peak scenario, we also require scalability. These services should automatically scale up if there are too many messages created. Additionally, conversations between guests, hotel managers, and tour guides are typically not real-time or immediate. The system should not wait for a response from message receiver. Therefore we need to pass messages asynchronously.

We made the decision to employ Google Cloud Pub/Sub Services based on the analyses mentioned above. Services can communicate asynchronously using Pub/Sub with latencies

of around 100 ms, according to [5]. It produces publishers and subscribers, who are message/event creators and consumers, respectively [5]. Events are broadcast asynchronously by the publisher to the subscribers [5]. The publisher can broadcast messages outlining the timing and method of processing an event. Additionally, it will send messages to all of the services that must respond [5].

2.5 Machine Learning Module

The application will have the ability to gather customer comments and develop a comprehensive analysis from it. The application will utilize Google Cloud AutoML, which takes data from a DynamoDB table and generates a prediction or analysis report automatically by training, testing, and deploying a machine learning model. One of the best things about of auto ml is we do not have to manually train the model to produce ml models. They have facilities to train and test the model. The application must only give the input table into auto ml table to get the ml model.

We will be using Auto ML tables. It automatically builds and deploys ML models using structured feedback data given by the endpoints. It automates modelling on a wide range of data types.

There are two types of analysis to be performed. First analysis is to find the similarity of stay duration of customers with which the application can propose a tour package. Second analysis will be to understand which tour package is most popular among customers based on the feedback given by the customers and give an appropriate score which will help the management decide how to improve their services.

The input for the auto ml will be received from the tour service and the data will be processed and converted into csv table which will be consumed by the auto ml service.

2.6 Web Application Building and Hosting

We will be using React and Nodejs to build a front-end application with backend services. The reason to choose ReactJS is that it provides interactive layout of any UI. Furthermore, it enables rapid and high-quality application development that saves time for both client and developer. ReactJS is a perfect combination of JavaScript and HTML tags. The syntax of JavaScript and HTML is always used which simplifies writing the code. Apart from that, it is extremely competent, easy to adopt, makes template design easy and provide good developer tools [10].

For hosting the application, we are going to use the Amazon service AWS Elastic Beanstalk. AWS EB is easy to use service for deployment of application. It automates setup and configuration of various AWS services, and we don't have to manage servers. Moreover, it supports Docker web applications [11].

2.7 Other Essential Modules

Report Generation using AWS Quick sight

For report generation we are going to use AWS Quick sight which is the embedded analytics tool for all application. It allows deeper data insights, and it is server less and easy to manage. Amazon Quick Sight has a server less architecture that scales many users without the need to setup, configure, or manage the servers [12].

Visualization using Google Studio

For visualization – building graphs, charts etc. we are going to use Google’ data studio visualization. It is an online tool that helps visualize data for free. The AI of Google Data Studio combines data from sources, and it automatically analyze those data and enables to make interactive charts, graphs [13].

3. RoadMap

3.1 User Authentication and Management Module

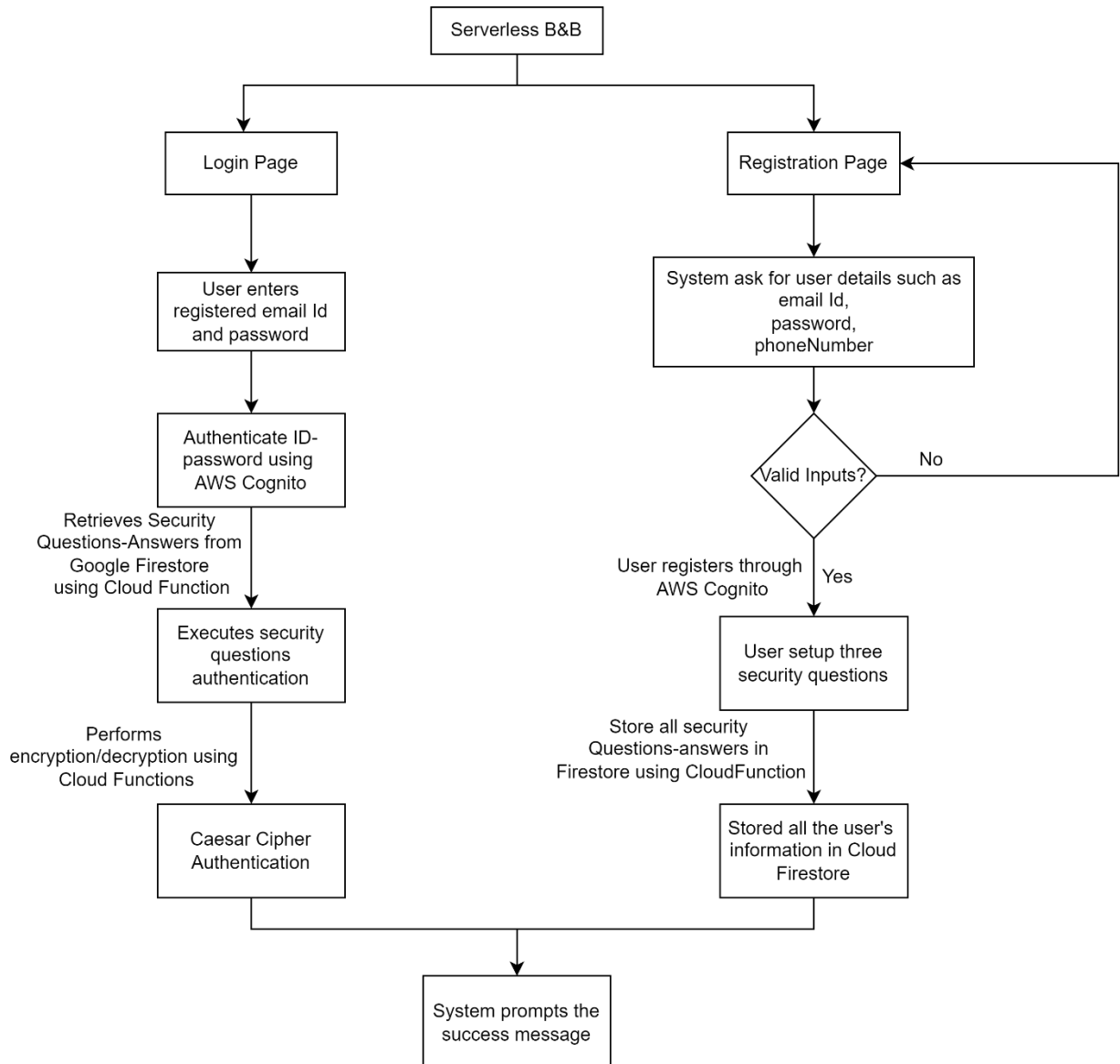


Figure 2: Flowchart of User Authentication and Management Module

It comprises a flowchart for two modules, one for user authentication and one for user administration, as illustrated in the above figure. If a user is not already registered, they must register in the application by entering information such as their email address, password, and other data. If the user does not submit the correct information, the user will be routed to the registration page until the user provides all the necessary and legitimate

information to create an account. If a user submits valid information, AWS Cognito will store all the user's information.

Apart from this, the user must configure security questions and answers, which will be saved in Google Firestore through cloud features. GCP Firestore will store all the user's information. The application will display a success message confirming that registration was successful and will redirect users to the login page as soon as the details are saved in the database.

The login procedure will be divided into three main parts. The user will initially input an email address and a password. The user's email address and password will be validated against data in AWS Cognito. If the user's data is correct, the user will be asked a security question, which will be fetched from Google Firestore using Cloud Function. The user's answer will be authenticated with an answer linked to that specific question in Firestore, and if valid, the user will be asked to convert the given plain text code displayed on the webpage into Caesar cipher, and this Caesar cipher code, along with the plain text code, will be checked at the backend using the Lambda function. If the query is correct, the user will be sent to the home page. If the user's information is invalid at any point, the user will be routed to the login page until the user provides proper authentication data.

3.2 Online Support Module

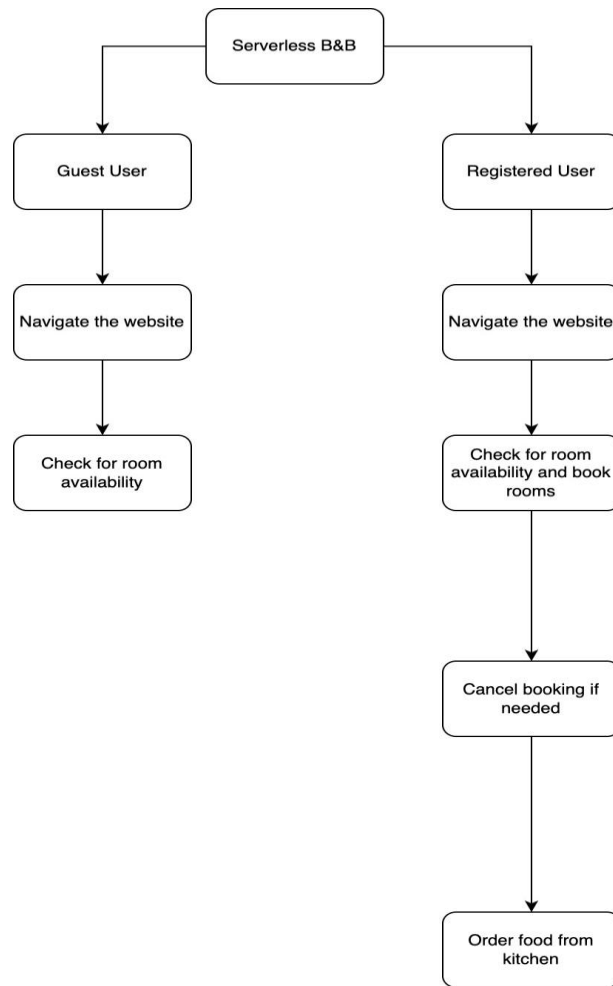


Figure 3: Functionalities for different user having access to online support

Online Support module would be handling requests of registered and guest user, each of which would be provided with different functionality as shown in Figure 3. We will be integrating Amazon Lex with ReactJS and provide users with the above functionalities (room, kitchen and navigation functionalities).

Flow

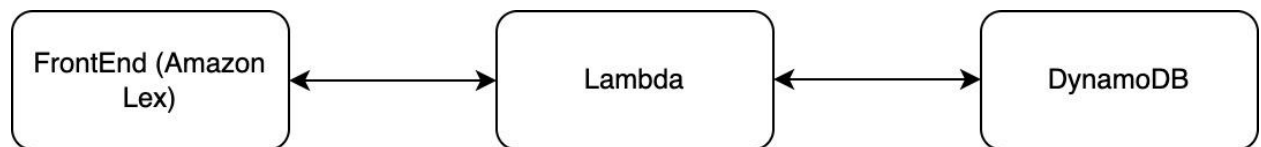


Figure 4: Flow of online support module

Users will communicate with Amazon Lex which will trigger lambda functions and perform CRUD operation on DynamoDB and store details about room booking and kitchen orders (as shown in Figure 4).

Test cases

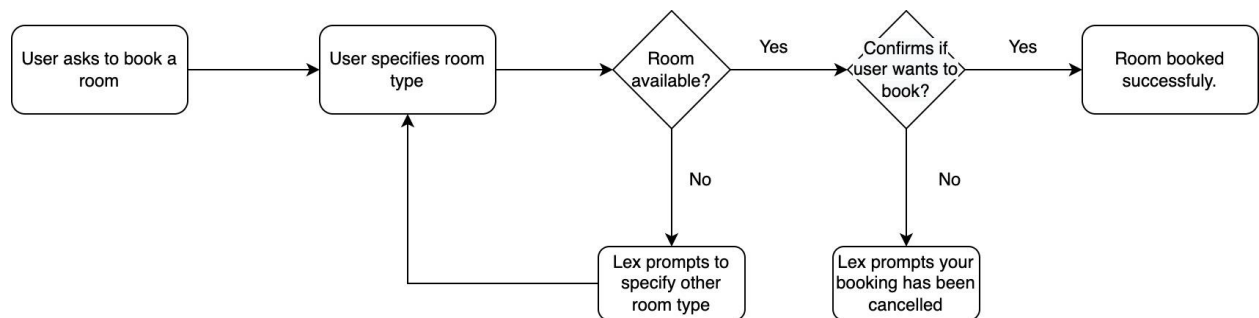


Figure 5: Book room test case for online support module

If users want to book a room via Chatbot, he/she will specify the room type and lambda will check the availability for the room type and prompt user if not available or confirms the booking if available (as shown in Figure 5). Once user has successfully booked the room, a prompt will be displayed to the user confirming the same.

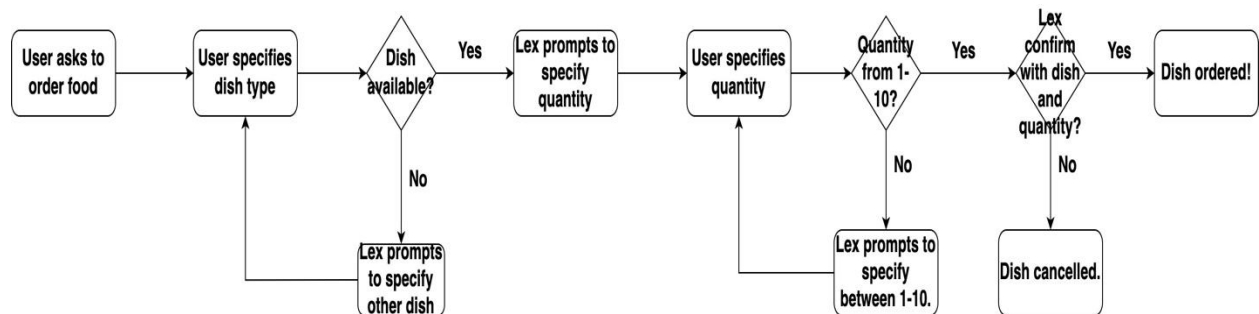


Figure 6: Order food test case for online support module

If users want to order food via Chatbot, he/she will specify the dish type and lambda will check the availability for the particular room type and prompt user if not available and ask for the available dish or confirms the dish if available (as shown in Figure 6). Once user has

successfully ordered the dish, a prompt will be displayed to the user confirming the same, else prompt that the same has been canceled.

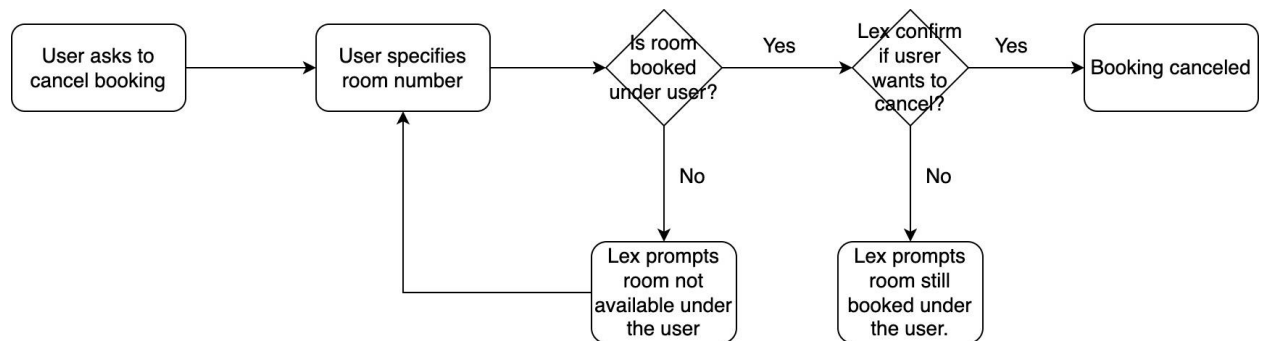


Figure 7: Cancel room test case for online support module

If users want to cancel room booking via Chatbot, he/she will specify the room number and lambda will check if the room is booked under the same person or not. A prompt will be displayed to the user confirming the same, on confirming the particular room booking will be canceled and prompt for the same will be displayed (as shown in Figure 7).

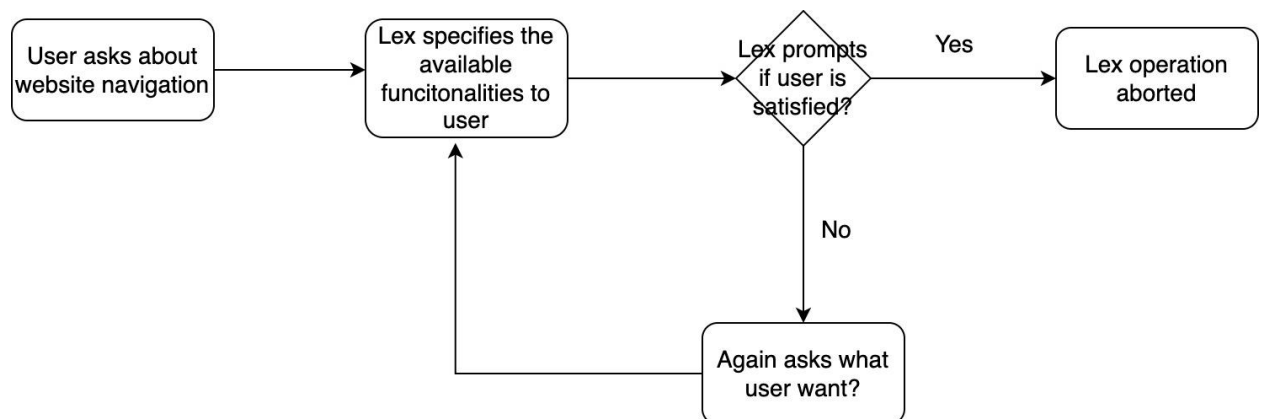


Figure 8: Website Navigation test case for online support module

If users want to navigate through the website, Lex will provide directions/functionalities to the user and a prompt will be displayed to the user confirming the same, on confirming Lex operation will be aborted, if not user will be prompted to specify their needs again (as shown in Figure 8).

3.3 Message Passing Module

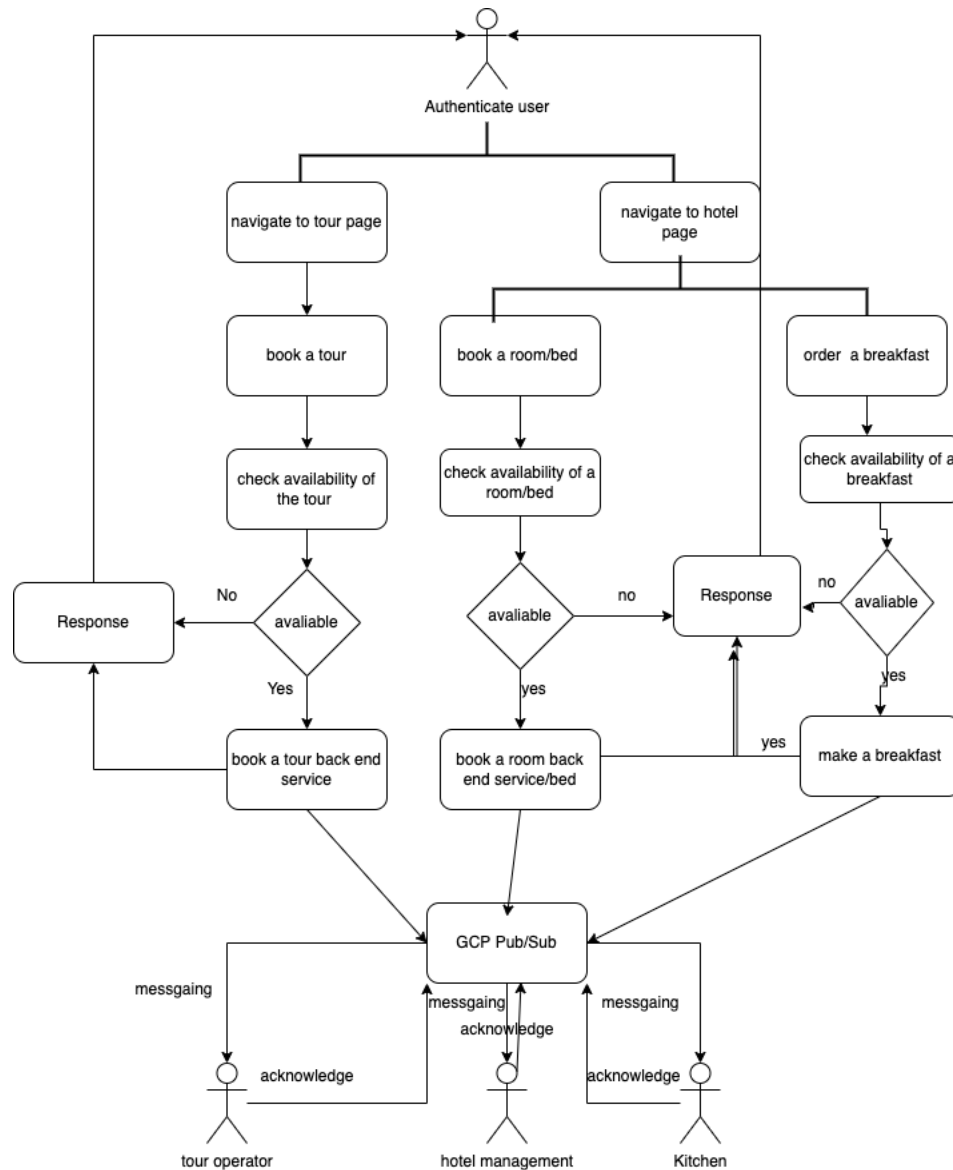


Figure 9: Flow chart of message passing module

This module decided to use GCP pub/sub services to allow the authenticate user to communicate with tour operator and hotel management. Unauthenticated are not allowed to use these services. Authenticate user can book a tour, book a room/bed, and order a breakfast with serverless B&B. If the service book successfully it acknowledged by tour operator, hotel management and kitchen. Figure 9 shows how the message passing module works. It starts from tour page or hotel page. From here use can book a tour or book a room. Then it will automatically check the availability of the tour or room. If the service is available, then the tour or room will book and respond to user while GCP pub/sub will also

send a message to tour operator, hotel management and kitchen. Tour operator, hotel management and kitchen can acknowledge services are created, so they can prepare the service.

3.4 Machine Learning Module

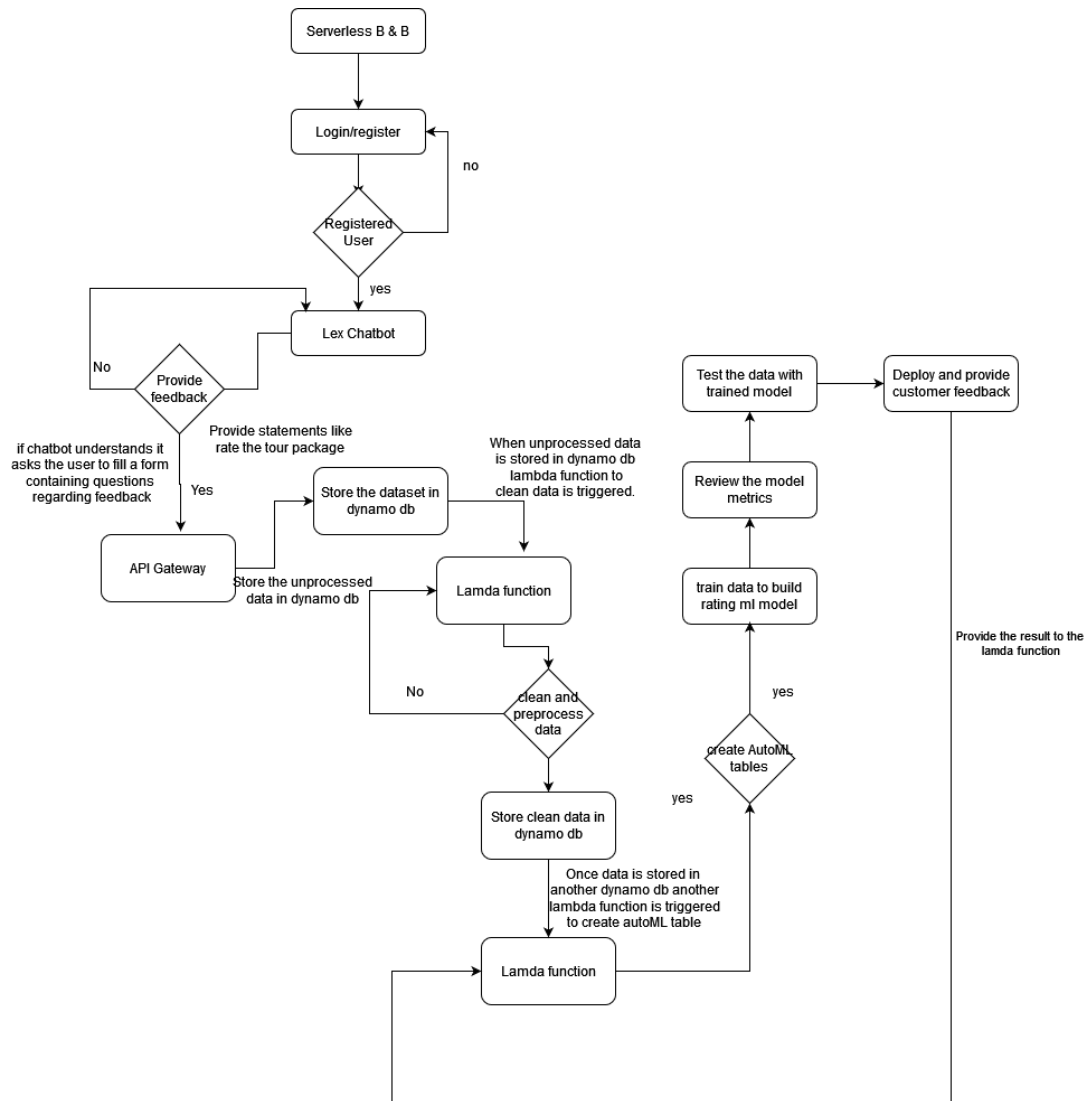


Figure 10: Flow chart of Machine Learning Module

The app will have a facility for providing comments and evaluations on the services. However, to do so, the user must first authenticate. So, after the user authenticates, the Lex may deliver by asking it. Lex analyses the statement and, if it is acceptable, suggests a series of questions related to feedback to the user. Users can respond to those questions, and the responses will be saved in DyanmoDB via the Lambda and API gateway. However, because the output generated by unstructured or unprocessed data is not noticeable or dependable,

these stored data will be unprocessed data that cannot be directly used to build or train the model. As a result, the unprocessed data will be subjected to cleaning and other necessary pre-processing processes, which will be carried out by executing other Lambda functions.

When the data is processed, it is placed in another DynamoDB table and triggers other Lambda functions that oversee building AutoML tables. These tables may then be utilized to train the machine learning model. Following that, evolution metrics for that training will be created, and appropriate adjustments will be performed throughout the testing phase based on the study of those metrics. So, after the final model is completed, it will be deployed and utilized to make suggestions based on the input provided by the consumer.

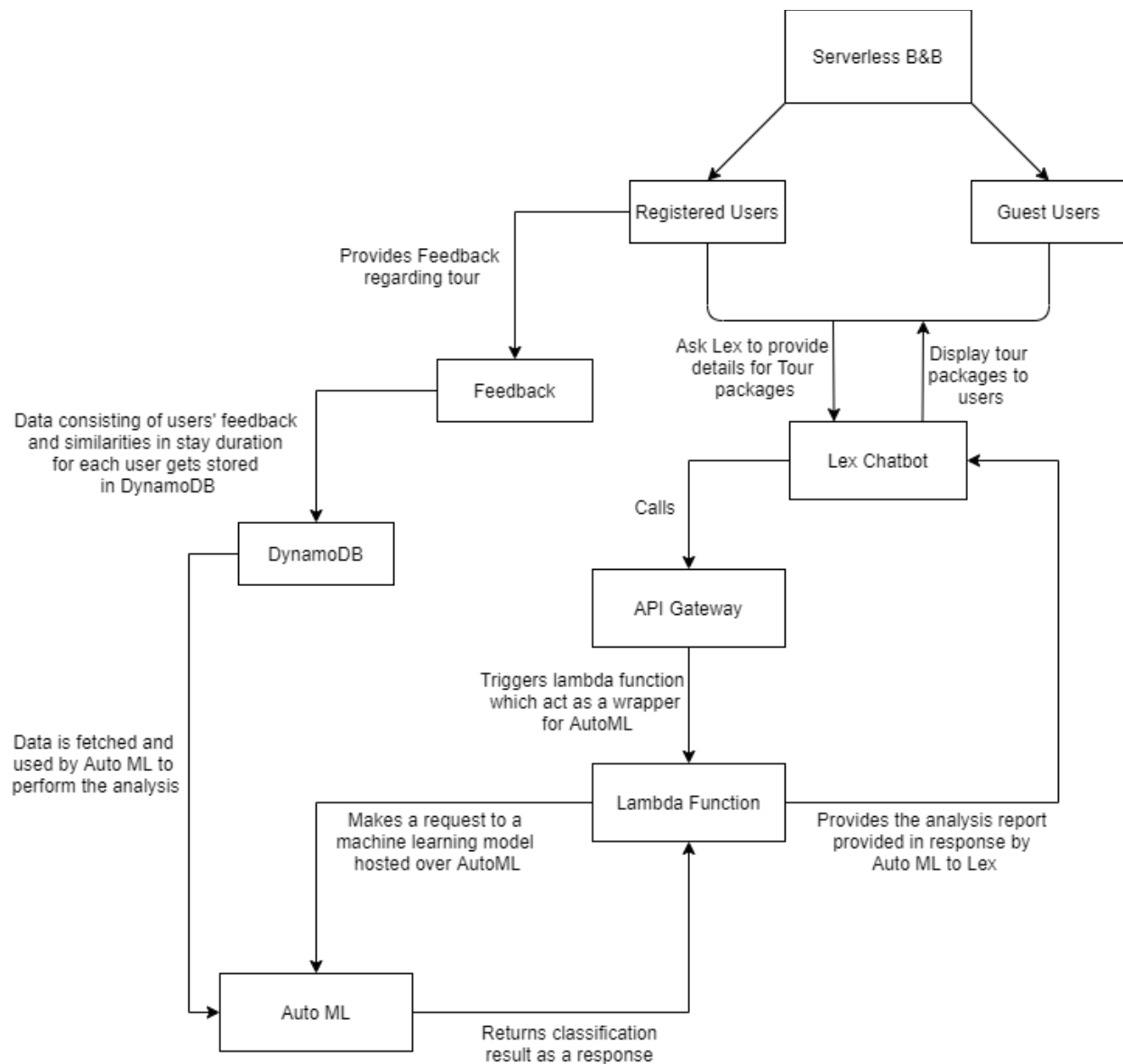


Figure 11: Flowchart of Tour Management Module

As illustrated in the above figure, both guests and registered users will be able to obtain information about the tour package. They will inquire about tour package specifics in Lex. As soon as Lex receives a query from a user, it will invoke a lambda function via API Gateway, which will request the tour data from GCP Auto ML. The request received by Auto ML is then processed, and the classification result produced through analysis is returned to the lambda function. When lambda receives the analytical data from Auto ML, it sends the response to Lex, who then displays it to the user.

The app will make use of Google Cloud AutoML, which takes data from a DynamoDB table and automatically creates a prediction or analysis report by training, testing, and deploying a machine learning model. There are two forms of analysis that must be carried out. The initial analysis is to determine the similarity of client stay durations so that the application may recommend a tour package. The second study will be to determine which tour package is most popular among consumers based on customer input and provide a suitable score to help management decide how to improve their services. The tour service will provide the input for the auto ml, and the data will be processed and turned into a CSV table that will be consumed by the auto ml service.

3.5 Web Application Building and Hosting

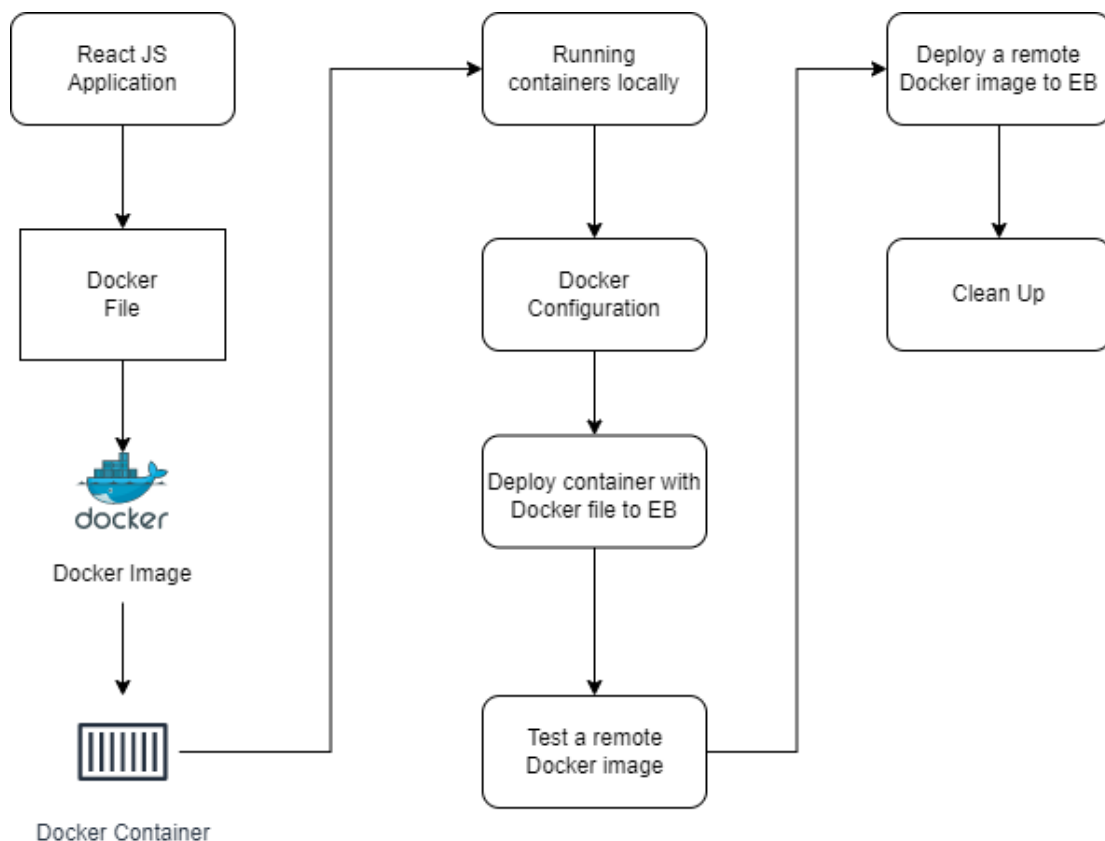


Figure 12: Flow chart of Web Application Hosting

To host the application on Elastic Beanstalk using Docker platform branch, firstly, the application is dockerized into the Docker Container. Then the container needs to be tested that it is running locally and to perform the testing, the Elastic Beanstalk CLI is used to configure the local repository for deployment to Elastic Beanstalk. The docker is then configured in the hosted elastic beanstalk server [11].

The application now is deployed to an Elastic Beanstalk environment. Elastic Beanstalk uses the information in the Docker File to build and run the image. After the deployment, the remote Docker Image is tested and then deployed to Elastic Beanstalk. When finish working with Elastic Beanstalk, the environment is terminated. Elastic Beanstalk terminates all AWS resources associated with your environment, such as Amazon EC2 instances, database instances, load balancers, security groups etc. [11].

3.6 Other Essential Modules

Report Generation using AWS Quicksight

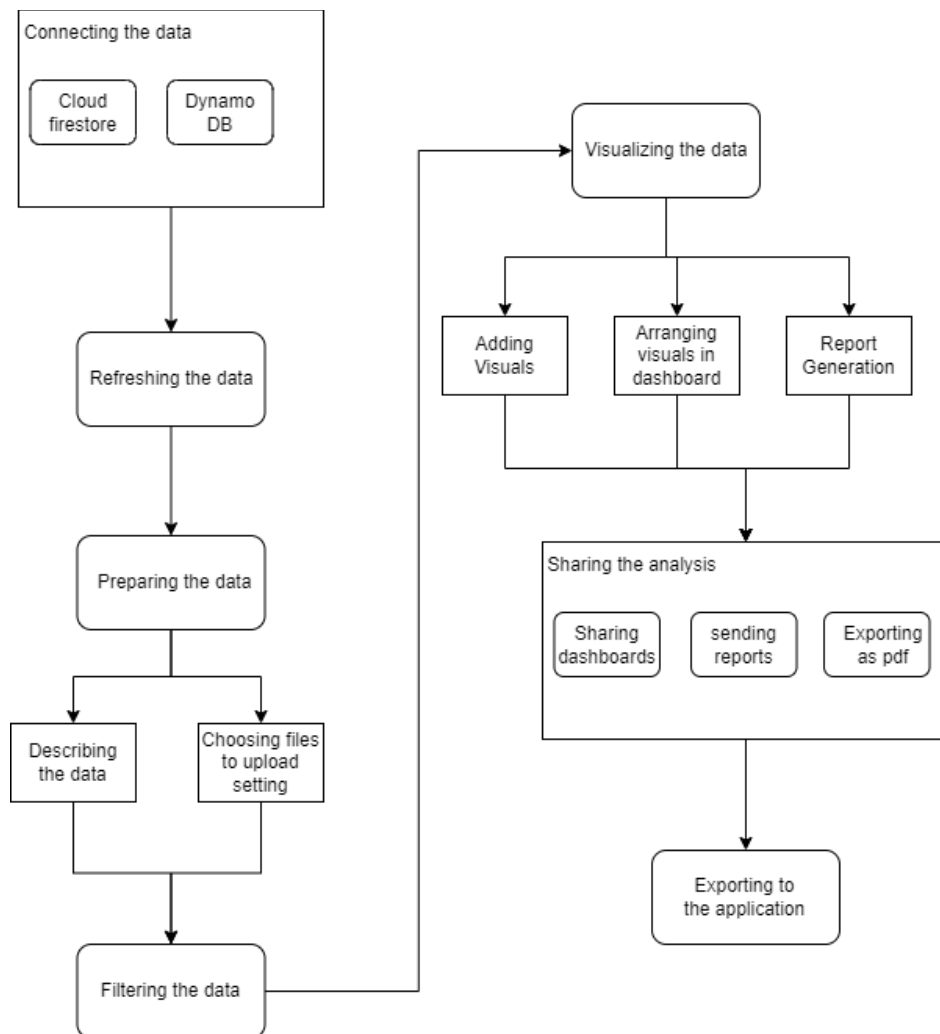


Figure 13: Flow chart for Report Generation

The first step for any visualization is connecting the data. The data on which the report is going to be generated is connected to AWS Quicksight. When refreshing data, Amazon QuickSight handles datasets differently depending on the connection properties and the storage location of the data. Datasets keep track of whatever data preparation is done so that the prepared data in different studies is utilized. Options for data preparation include the ability to apply filters, add calculated fields, change field names, and change the data types of fields [12].

The filtering is done to refine the data in the dataset for analysis. Then the visualization is started, that is the report generation. The report can be exported as pdf. Moreover, the dashboard can be shared as report to the users [12].

Visualization using Google Studio

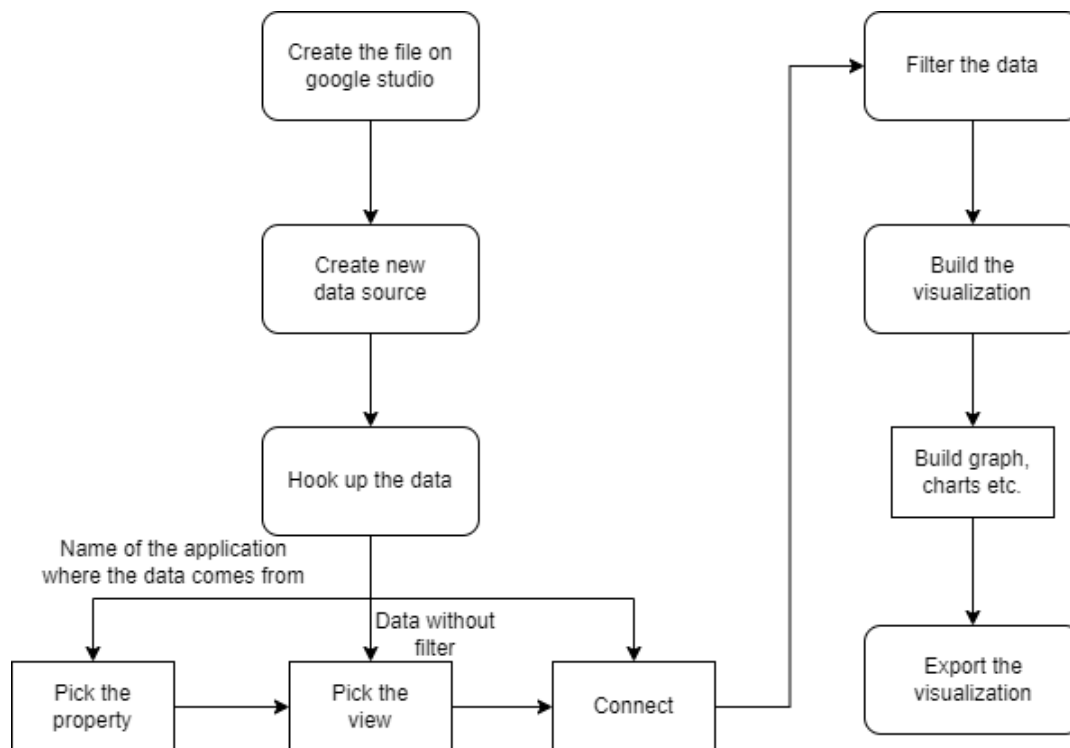


Figure 14: Flow chart for Visualization

For performing visualization in google data studio, the first step should be creating a file which is usually a blank report. After that the data is to be added and for that we must create a new data source. Then after configuring account, property, and view, we can connect to the visualization [14].

Here we can filter, sort, and arrange the data. After that the visualization is build. We can export the visuals through links and as google studio is a free online visualization tool we can share through invitation or share link and can give view and/or edit access [14].

3.7 Final RoadMap

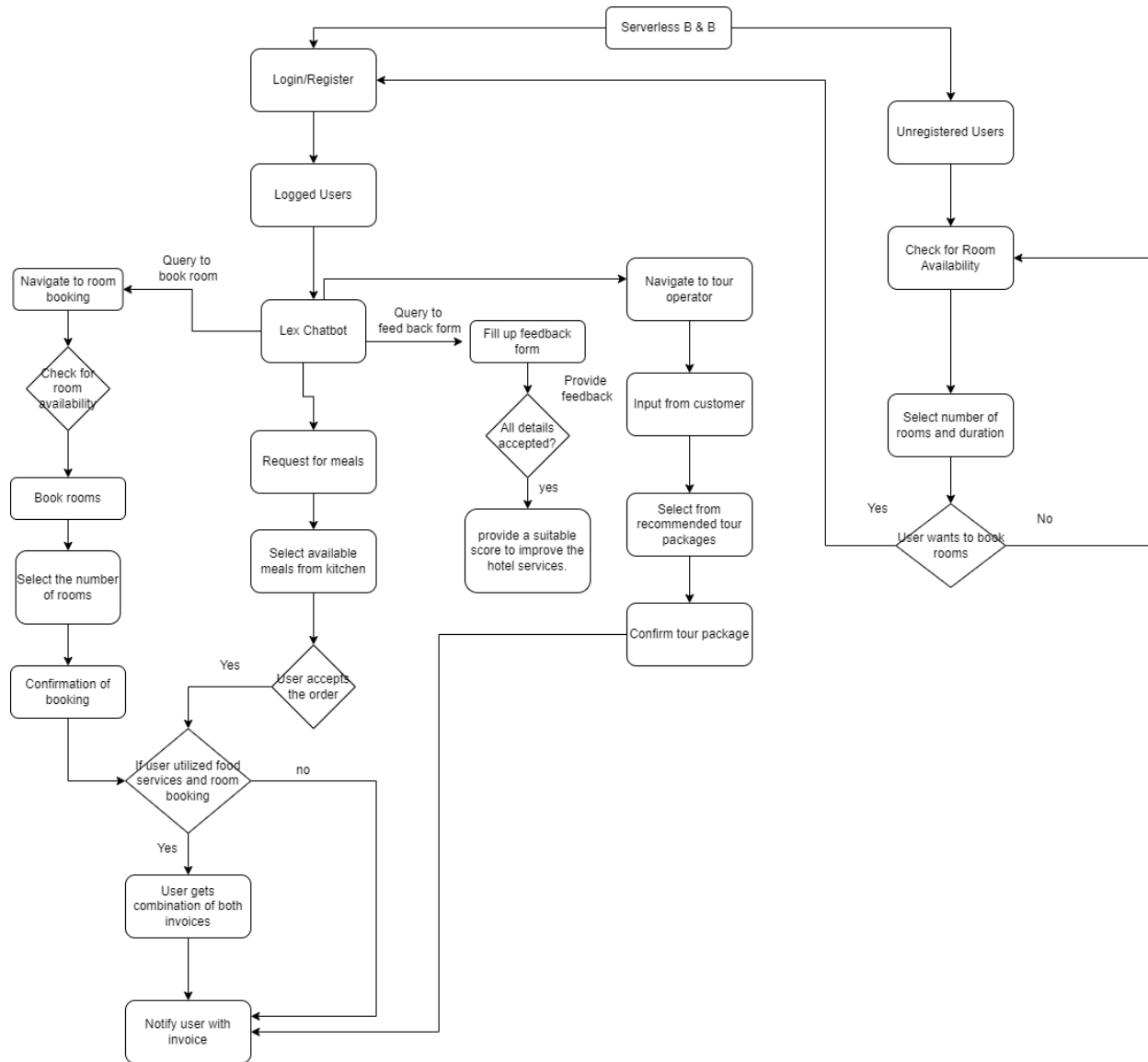


Figure 15: Serverless B&B RoadMap

The application's main page is where users begin. The user has two choices if they have previously registered and authenticated. The user begins by submitting questions to the chatbot, which serves as our online help section. The user may reserve lodging, place meal orders, and customize a travel itinerary to suit his requirements. We also offer feedback forms so that customers may assess and evaluate trip packages and hotel services, which helps the hotel management deliver better services. The user can browse the numerous services offered by the hotel management even if they are not registered. As a guest user, he has access to the room

availability. But if he wants to book the rooms he has to be authorized to do so. So, he has to create an account with the hotel management.

The system's registered customers can reserve available rooms with or without the option for dining services. To book rooms, the user uses the online support module. The user may decide which dates and how many rooms he wishes to reserve before confirming the reservation. This activates the notification system, which sends the confirmation email and details on the room booking.

Once a room is reserved, the customer could pre-order meals and take advantage of the hotel's suggested tour packages, if the user chooses to pre-opt for the meal services. When the food order is verified, an invoice for the meal will be sent to the client. He can inform the chatbot that he wishes to buy food and provide feedback to online support. The customer may choose tour package services offered by the hotel if he wants to book a tour package to get better discounts. He will provide input to the tour operator, which utilizes machine learning to select the most appropriate tour packages based on his input. When the users in the hotel, user orders breakfast from the kitchen. Once the kitchen acknowledges the order and prepares breakfast. The system will notify the user the order has been made and the breakfast has been created. At end of the tour, System will generate invoices for the user. And users can write feedback about the tour service and hotel service.

4. Modules Distribution

Table 1: Modules Distribution for Serverless B&B

Sr. No.	Modules	Group Members
1	User Management and Authentication	Kavan Patel, Prit Sorathiya
2	Online Support	Neelansh Gualti
3	Message Passing	Qiwei Sun
4	Machine Learning	Kavan Patel, Prit Sorathiya, Navya Jayapal
5	Web application Hosting	Samarth Jariwala
6	Other Essential Modules	Samarth Jariwala

5. Meeting logs

Meeting Number: #1

Date of Meeting: May 12, 2022

Time: 10:00pm – 10:17pm

Place: Teams

Agenda: Meeting team members and schedule the team meetings

Discussion: Availability for the team meeting and team strength (experience and technology stacks)

Decision: Meet on Tuesday at 12:30 every week

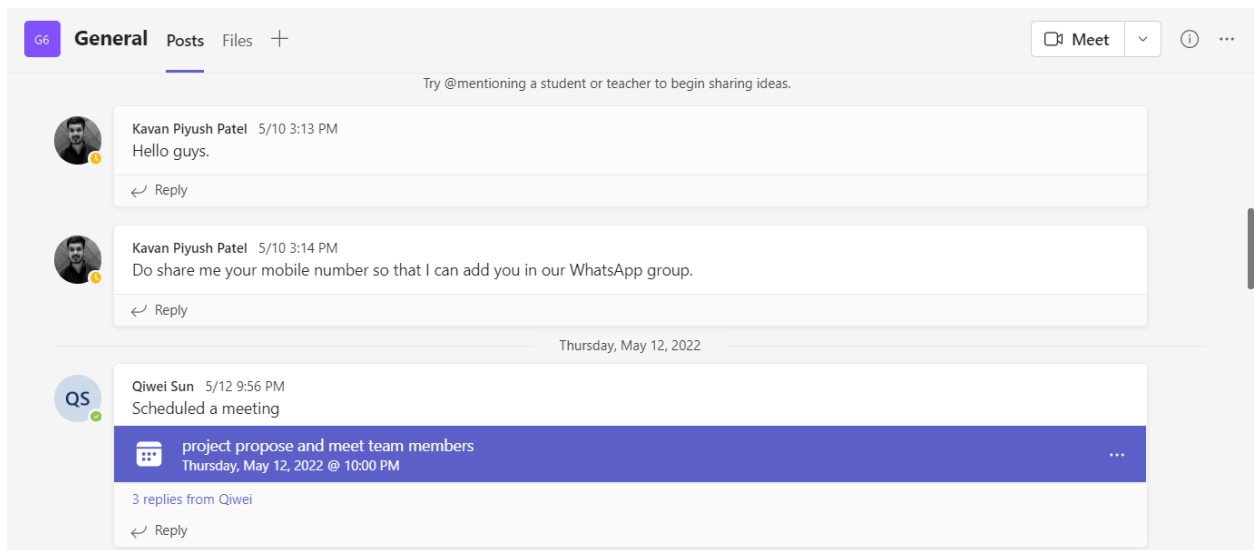


Figure 16: Screenshot of Meeting 1

Meeting Number: #2

Date of Meeting: May 17, 2022

Time: 9:00pm – 9:10pm

Place: Teams

Agenda: Pop-up meeting to discuss project

Discussion: understanding the project then distributing task for next meeting

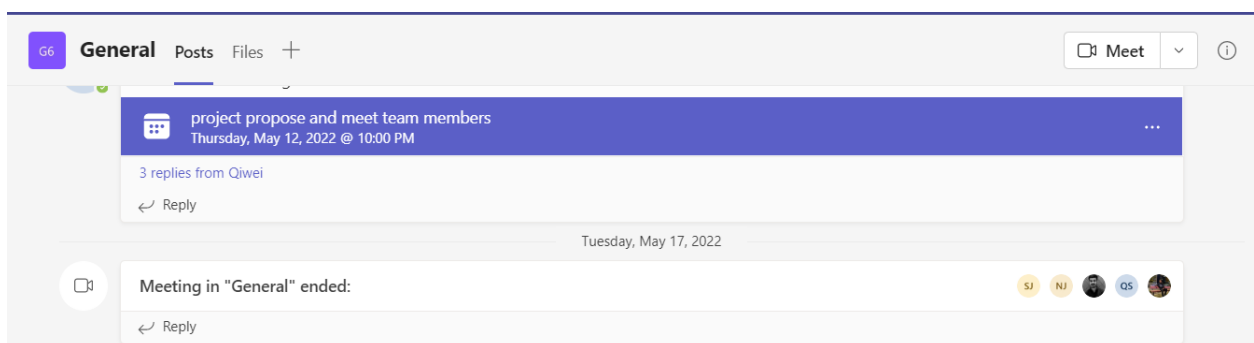


Figure 17: Screenshot of Meeting 2

Meeting Number: #3

Date of Meeting: May 25th, 2022

Time: 5:30pm – 7:30pm

Place: shift lab key 426

Agenda: overview about project architecture and project comments and web Technologies

Meeting Number: #4

Date of Meeting: May 30th, 2022

Time: 5:30pm – 6:40pm

Place: shift lab key room 426

Agenda: make a conceptual level understanding about the project and explore aws and gcp cloud services which we will use in our project.

Meeting Number: #5

Date of Meeting: June 1st, 2022

Time: 10:00pm – 10:10pm

Place: Teams

Agenda: Finalization of conception report.

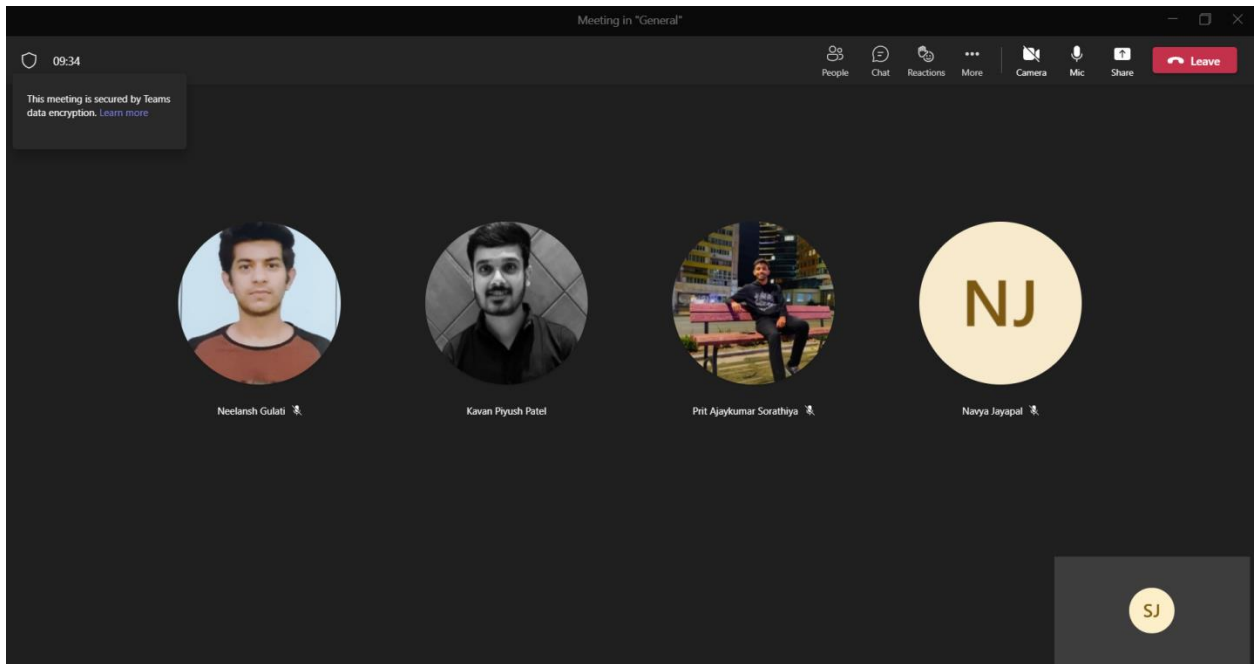


Figure 18: Screenshot of Meeting 5

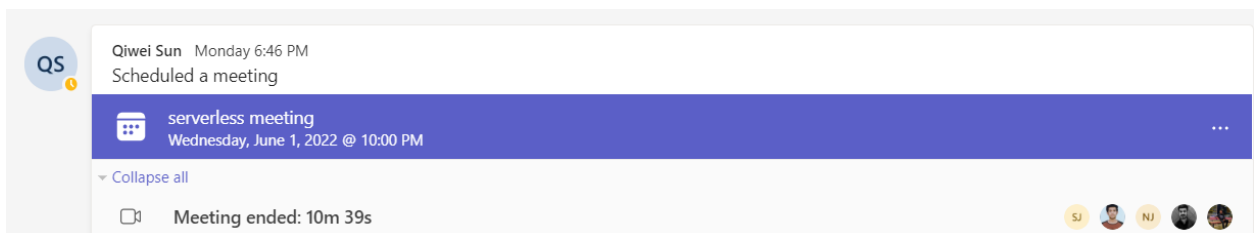


Figure 19: Screenshot of Meeting 5

Meeting Number: #6

Date of Meeting: June 28th, 2022

Time: 7:00pm – 7:30pm

Place: Teams meeting

Agenda: Doubts clarification with TA

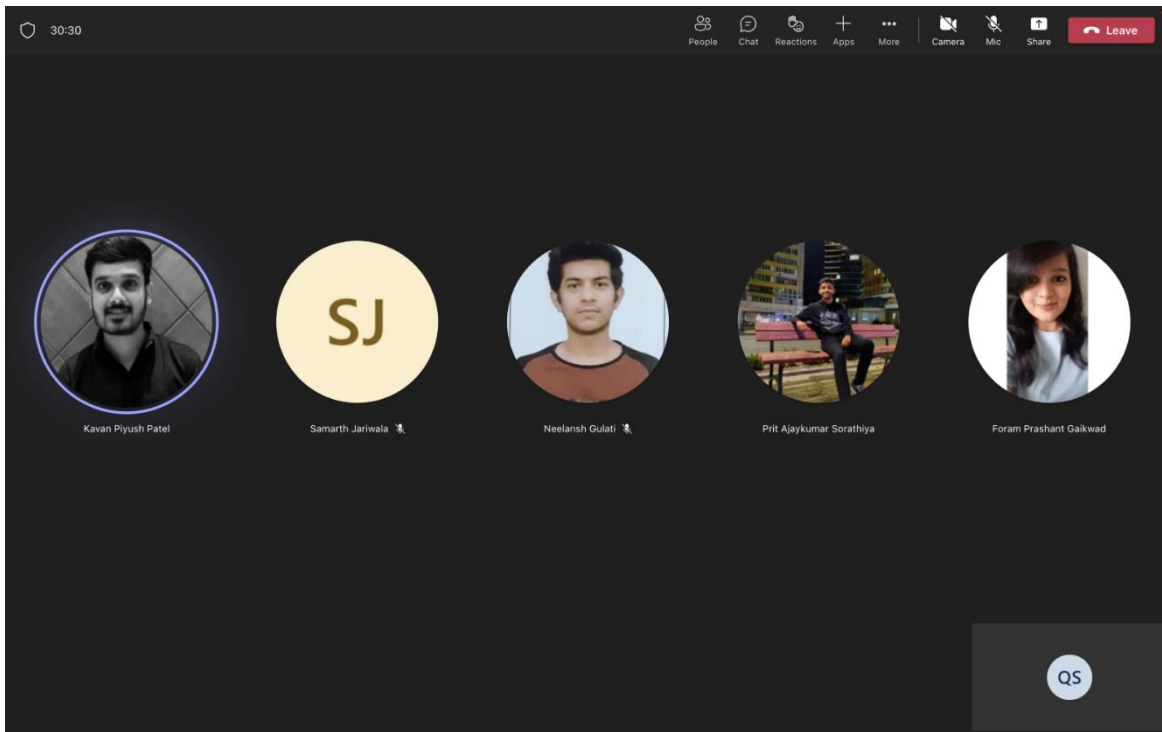


Figure 20: Screenshot of Meeting 6

Meeting Number: #7

Date of Meeting: July 2nd, 2022

Time: 2:00pm – 2:14pm

Place: Teams meeting

Agenda: project report discussion

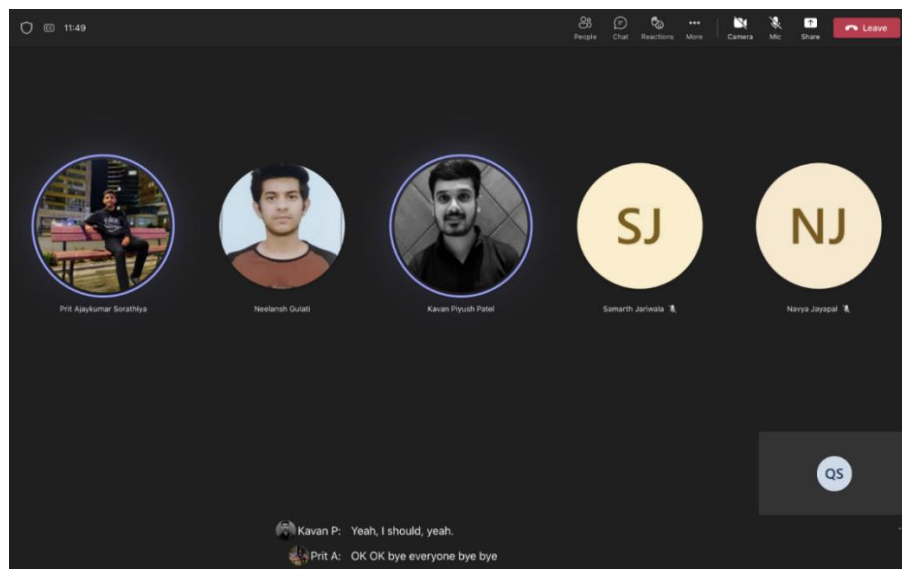


Figure 21: Screenshot of Meeting 7

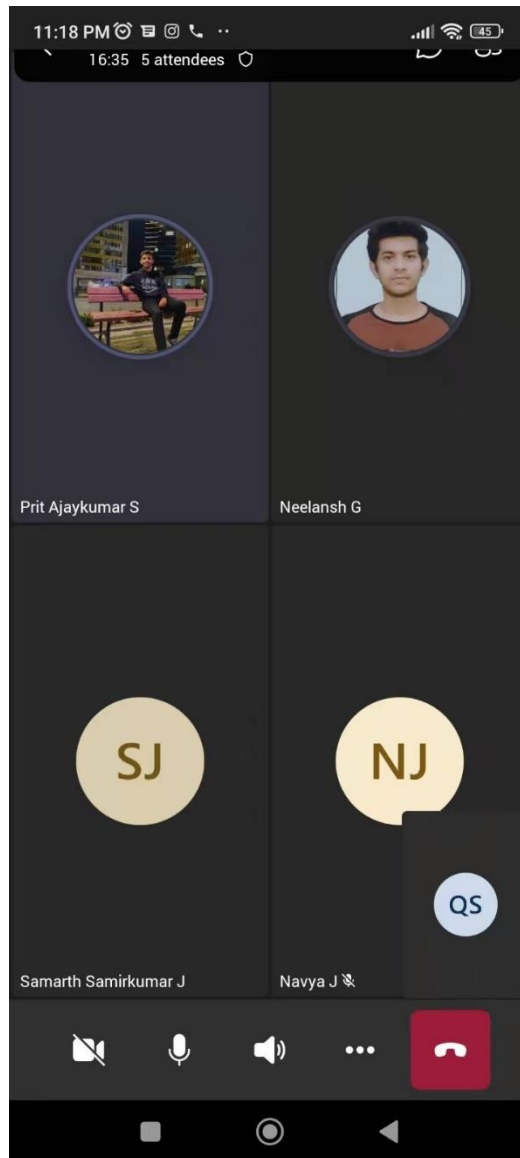
Meeting Number: #8

Date of Meeting: July 2nd, 2022

Time: 11:00pm – 11:18pm

Place: Teams meeting

Agenda: project report discussion

*Figure 22: Screenshot of Meeting 8*

References

- [1] "Fast NoSQL Key-Value Database – Amazon DynamoDB – Amazon Web Services", *Amazon Web Services, Inc.*, 2022. [Online]. Available: <https://aws.amazon.com/dynamodb/>. [Accessed: 30- May- 2022].
- [2] "Amazon Cognito - Simple and Secure User Sign Up & Sign In | Amazon Web Services (AWS)", *Amazon Web Services, Inc.*, 2022. [Online]. Available: <https://aws.amazon.com/cognito/>. [Accessed: 30- May- 2022].
- [3] "Firestore: NoSQL document database | Google Cloud", *Google Cloud*, 2022. [Online]. Available: <https://cloud.google.com/firestore#section-5>. [Accessed: 31- May- 2022].
- [4] "Cloud Functions | Google Cloud", *Google Cloud*, 2022. [Online]. Available: <https://cloud.google.com/functions>. [Accessed: 31- May- 2022].
- [5] "What is Pub/Sub? | Cloud Pub/Sub Documentation | Google Cloud", *Google Cloud*, 2022. [Online]. Available: <https://cloud.google.com/pubsub/docs/overview>. [Accessed: 31- May- 2022].
- [6] blazeclan. *5 key benefits of Amazon S3*. [online] Available at: <https://www.blazeclan.com/blog/5-key-benefits-of-amazon-s3/> [Accessed 2 June 2022].
- [7] *5 Use cases for DynamoDB*. [online] Available at: <https://rockset.com/blog/5-use-cases-for-dynamodb/> [Accessed 2 June 2022].
- [8] McGloin, C., *How to Overcome Intent Limitations on Amazon Lex and other NLP Engines*. [online] ServisBOT AI. Available at: <https://servisbot.com/nlp-intent-limitations/> [Accessed 2 June 2022].
- [9] *What is Amazon Lex?* [online] Available at: <https://docs.aws.amazon.com/lex/latest/dg/what-is.html> [Accessed 2 June 2022].
- [10] Deshpande, A., *7 Advantages of ReactJS for Building Interactive User Interfaces*. [online] Clariontech.com. Available at: <https://www.clariontech.com/blog/7-advantages-of-reactjs-for-building-interactive-user-interfaces> [Accessed 2 June 2022].
- [11] Amazon Web Services, Inc. *AWS Elastic Beanstalk – Deploy Web Applications*. [online] Available at: <https://aws.amazon.com/elasticbeanstalk/> [Accessed 2 June 2022].
- [12] Amazon Web Services, Inc. *Amazon QuickSight Features - Business Intelligence Service - Amazon Web Services*. [online] Available at: <https://aws.amazon.com/quicksight/features/> [Accessed 2 June 2022].

[13] MUO. *The 8 Best Features of Google Data Studio for Data Analysis and Visualization*. [online] Available at: <https://www.makeuseof.com/best-features-google-data-studio-data-analysis-visualization/> [Accessed 2 June 2022].

[14] “How to visualize data with Google Data studio,” *Content Marketing Institute*. [Online]. Available: <https://contentmarketinginstitute.com/2020/03/google-data-studio-visual-content/> [Accessed: 05-Jul-2022].