## AWS CloudFormation Deployment

**CloudFormation:**

AWS CloudFormation is an AWS service that uses template files to automate the setup of AWS resources.

Deployment speed, Scaling up, Easy updates, Security are the key features of Cloud Formation.

**Deploying the chatbot using CloudFormation:**

1. Click the LaunchStack for the Region in which you created chatbot

Northern Virginia - [LaunchStack](https://us-east-1.console.aws.amazon.com/cloudformation/home?region=us-east-1%23/stacks/create/review?templateURL=https://s3.amazonaws.com/aws-bigdata-blog/artifacts/aws-lex-web-ui/artifacts/templates/master.yaml&stackName=lex-web-ui&param_BootstrapBucket=aws-bigdata-blog)

Oregon- [LaunchStack](https://us-west-2.console.aws.amazon.com/cloudformation/home?region=us-west-2%23/stacks/create/review?templateURL=https://s3.amazonaws.com/aws-bigdata-blog-replica-us-west-2/artifacts/aws-lex-web-ui/artifacts/templates/master.yaml&stackName=lex-web-ui&param_BootstrapBucket=aws-bigdata-blog-replica-us-west-2)

Ireland- [LaunchStack](https://eu-west-1.console.aws.amazon.com/cloudformation/home?region=eu-west-1%23/stacks/create/review?templateURL=https://s3.amazonaws.com/aws-bigdata-blog-replica-eu-west-1/artifacts/aws-lex-web-ui/artifacts/templates/master.yaml&stackName=lex-web-ui&param_BootstrapBucket=aws-bigdata-blog-replica-eu-west-1)

Sydney- [LaunchStack](https://ap-southeast-2.console.aws.amazon.com/cloudformation/home?region=ap-southeast-2%23/stacks/create/review?templateURL=https://s3.amazonaws.com/aws-bigdata-blog-replica-ap-southeast-2/artifacts/aws-lex-web-ui/artifacts/templates/master.yaml&stackName=lex-web-ui&param_BootstrapBucket=aws-bigdata-blog-replica-ap-southeast-2)

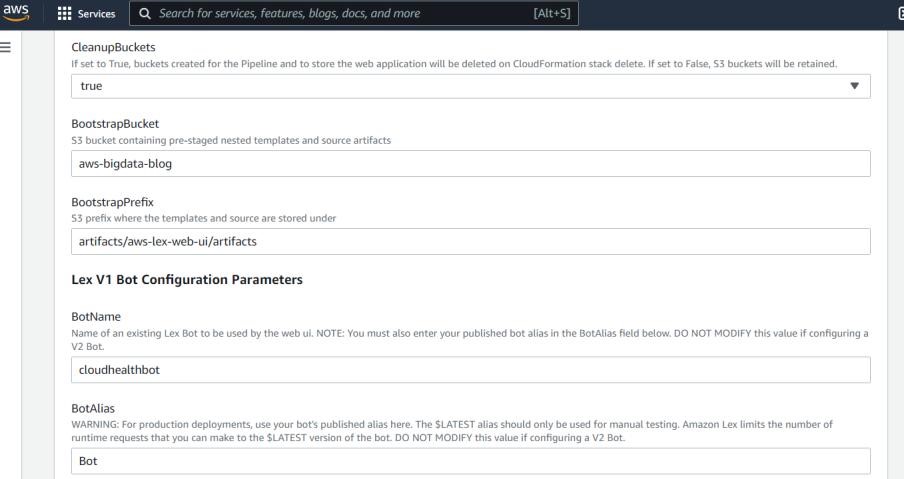
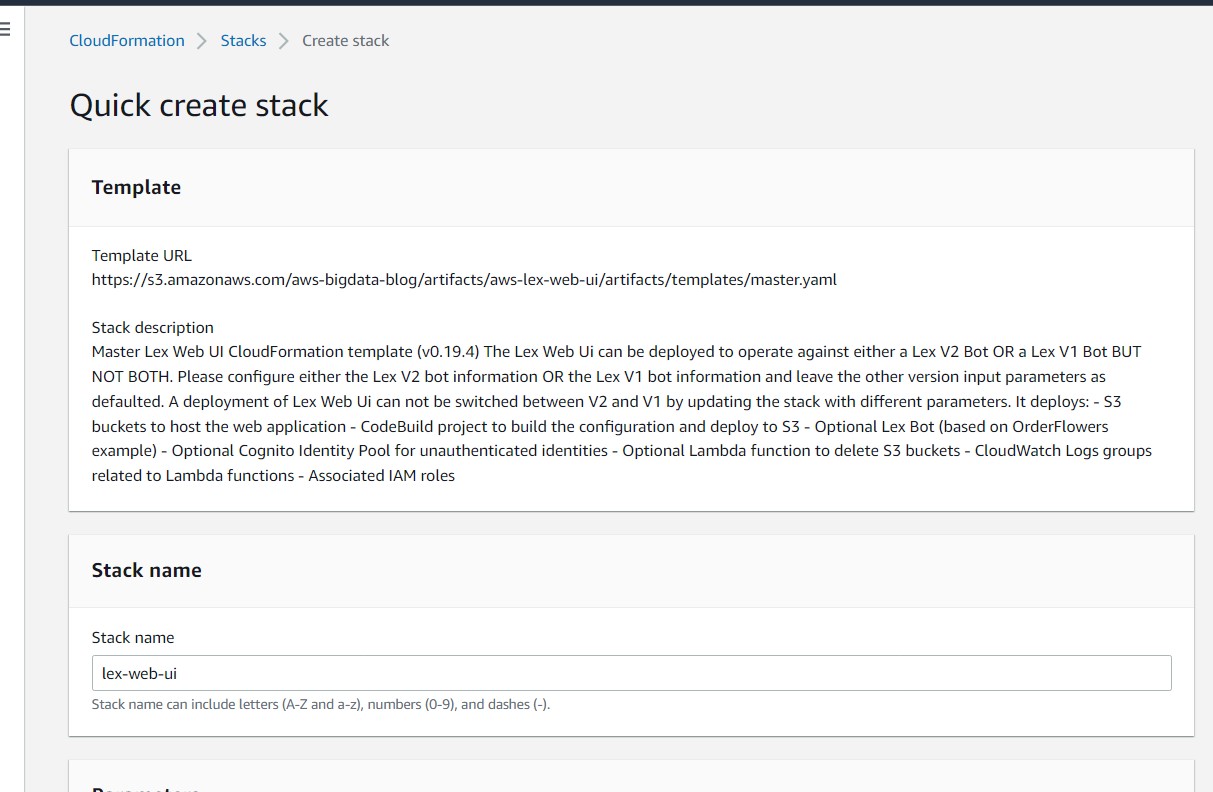
Singapore- [LaunchStack](https://ap-southeast-1.console.aws.amazon.com/cloudformation/home?region=ap-southeast-1%23/stacks/create/review?templateURL=https://s3.amazonaws.com/aws-bigdata-blog-replica-ap-southeast-1a/artifacts/aws-lex-web-ui/artifacts/templates/master.yaml&stackName=lex-web-ui&param_BootstrapBucket=aws-bigdata-blog-replica-ap-southeast-1a)

London[-LaunchStack](https://eu-west-2.console.aws.amazon.com/cloudformation/home?region=eu-west-2%23/stacks/create/review?templateURL=https://s3.amazonaws.com/aws-bigdata-blog-replica-eu-west-2/artifacts/aws-lex-web-ui/artifacts/templates/master.yaml&stackName=lex-web-ui&param_BootstrapBucket=aws-bigdata-blog-replica-eu-west-2)

Tokyo- [LaunchStack](https://ap-northeast-1.console.aws.amazon.com/cloudformation/home?region=ap-northeast-1%23/stacks/create/review?templateURL=https://s3.amazonaws.com/aws-bigdata-blog-replica-ap-northeast-1/artifacts/aws-lex-web-ui/artifacts/templates/master.yaml&stackName=lex-web-ui&param_BootstrapBucket=aws-bigdata-blog-replica-ap-northeast-1)

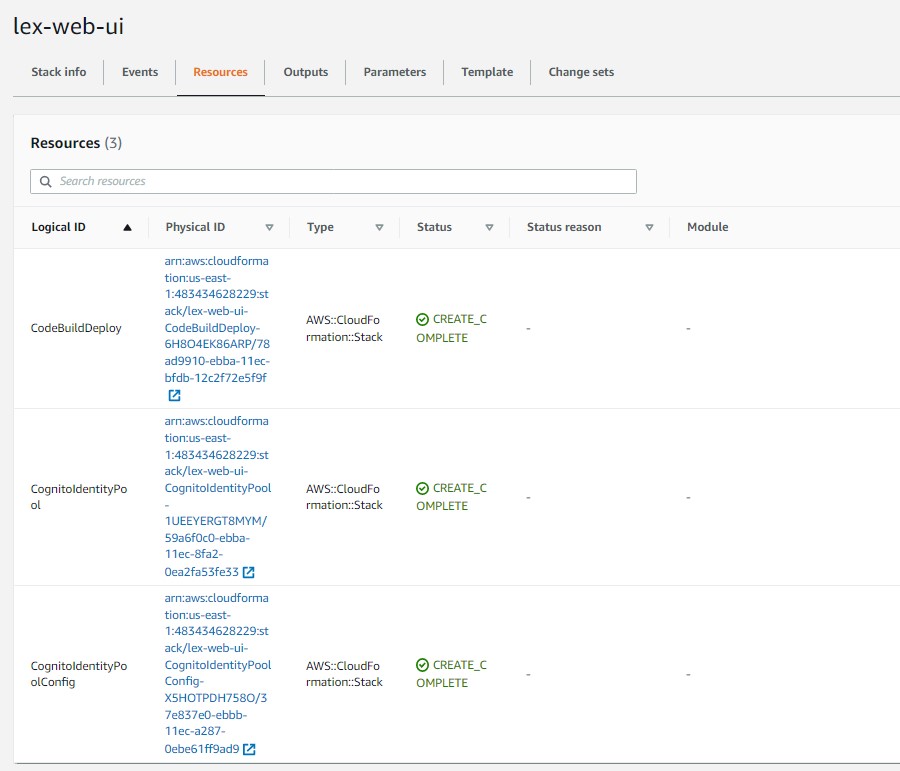
Frankfurt- [LaunchStack](https://eu-central-1.console.aws.amazon.com/cloudformation/home?region=eu-central-1%23/stacks/create/review?templateURL=https://s3.amazonaws.com/aws-bigdata-blog-replica-eu-central-1/artifacts/aws-lex-web-ui/artifacts/templates/master.yaml&stackName=lex-web-ui&param_BootstrapBucket=aws-bigdata-blog-replica-eu-central-1)

1. It directs to a stack creation page.
   * By default a yaml template is created in s3.
   * In the stack creation page give details regarding your bot.
   * Modify all the changes required i.e. Bot initial speech
   * Create a stack



1. Once stack is created it creates resources which generates cognito poll identity..etc.

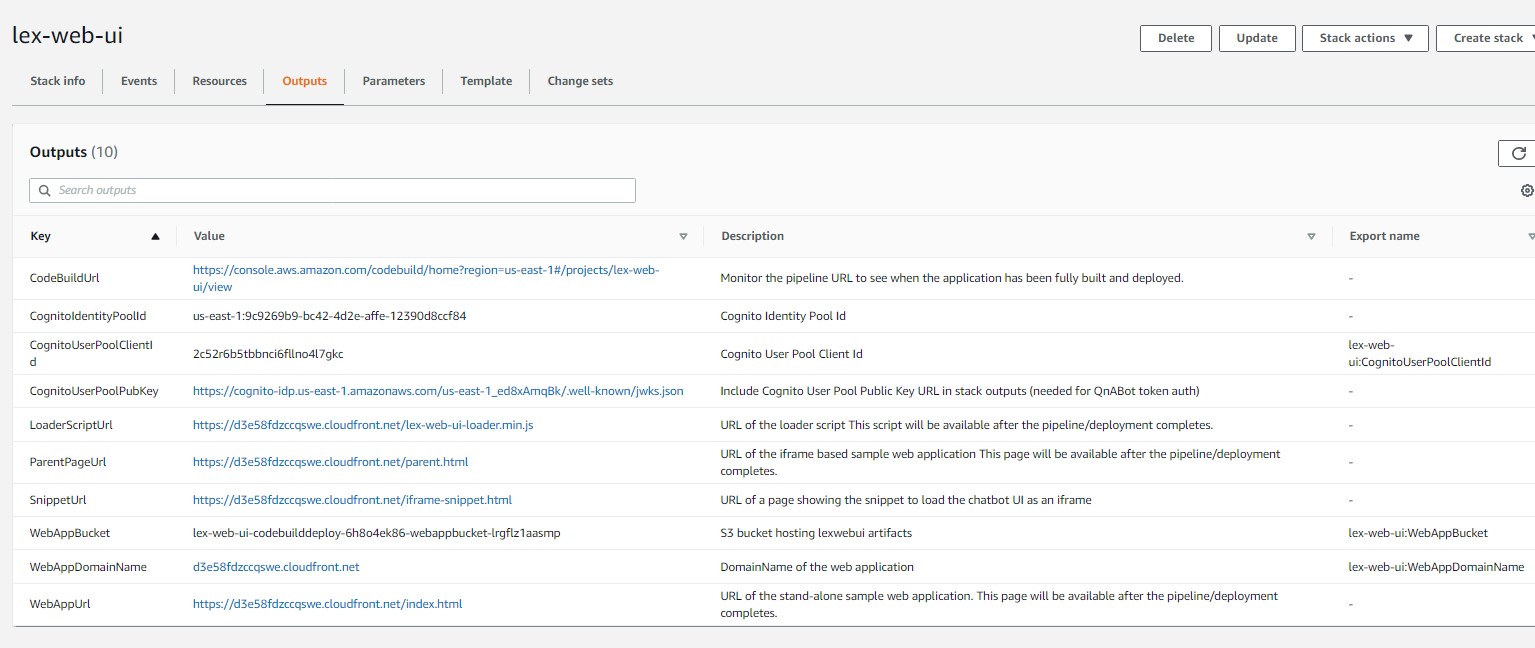
It also creates some nested stacks. Once all the resources were created we can access our chat bot.



Step4: By using WebAppUrl access our deployed chatbot.

[Click Here to access the chatbo](https://d3e58fdzccqswe.cloudfront.net/index.html)

[t](https://d3e58fdzccqswe.cloudfront.net/index.html)



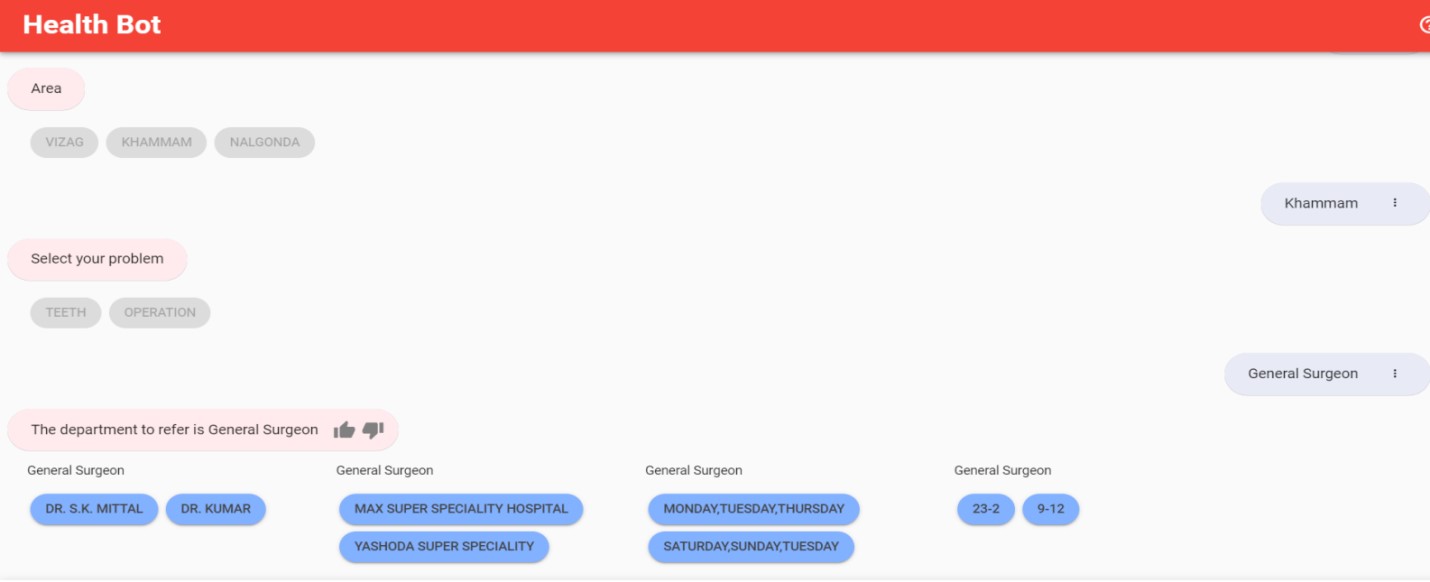
## Sample Outputs and Future Scope

Chat Bot Usage steps:

* User enter his/her name
* User enter his age
* User selects an area according to his/her convenient.
* User selects the type of treatment needed.
* User selects a doctor
* User selects a day on he/she want to book an appointment.
* User selects a time at which doctor is available
* User provides an email id to get confirmation responses
* On successful booking user receives a mail with a booking ID.

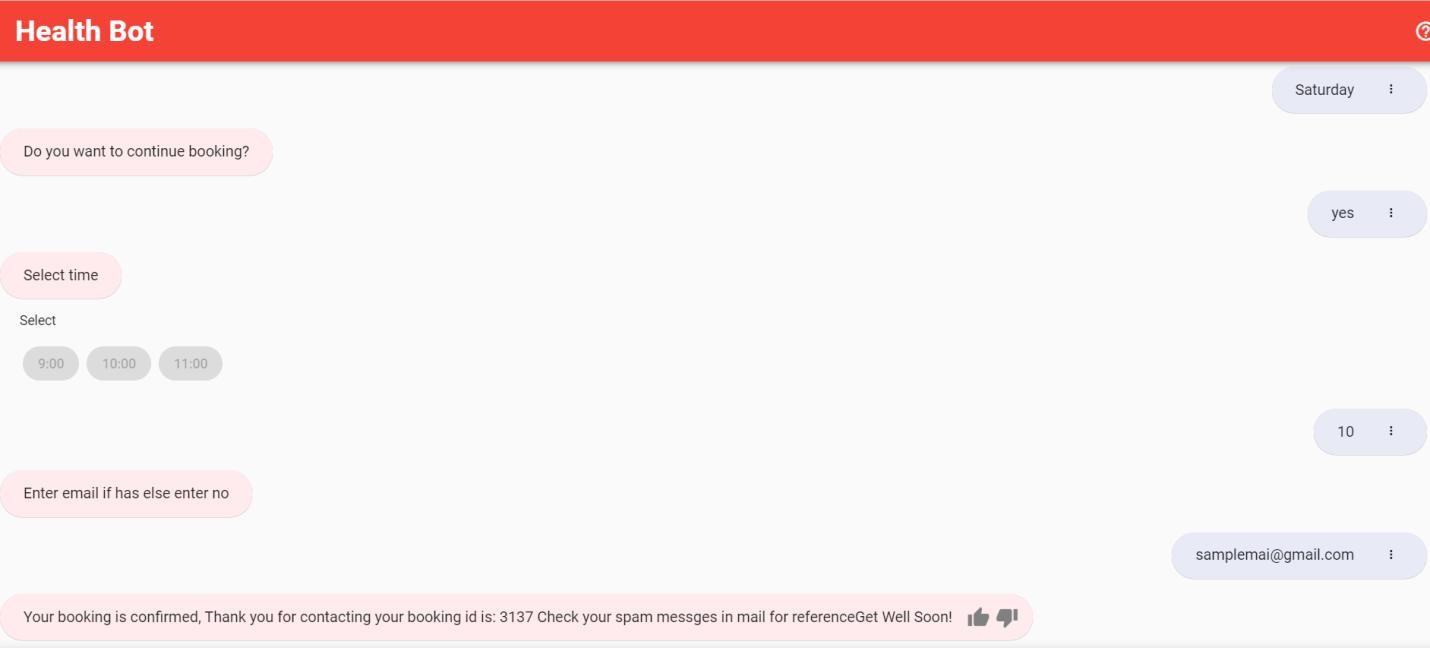
Successful Case:

A user named lex selected area khammam

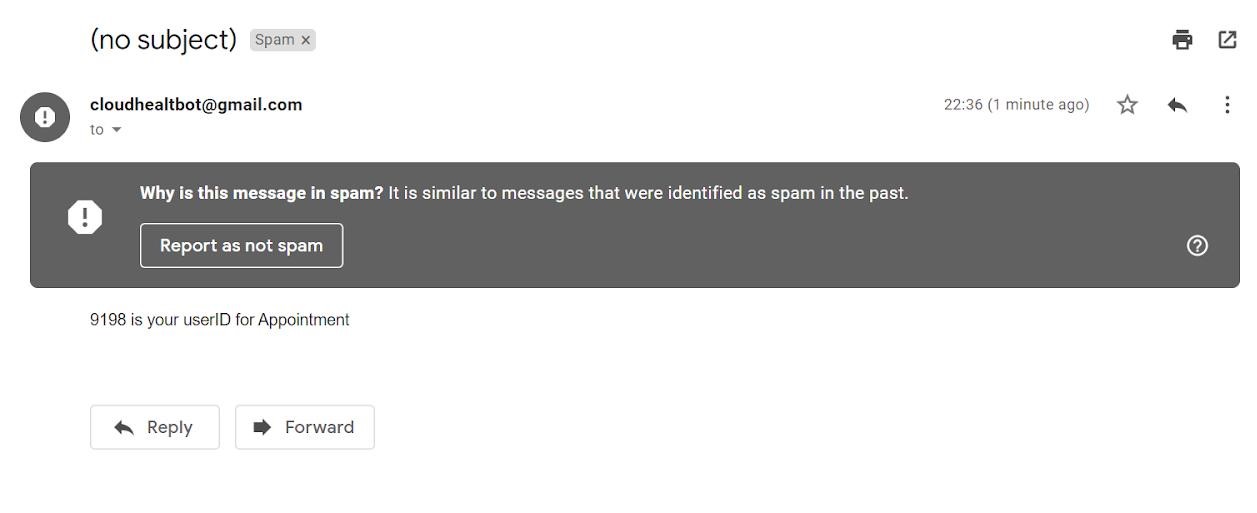


He need surgery and has chosen Dr. Kumar as his doctor.

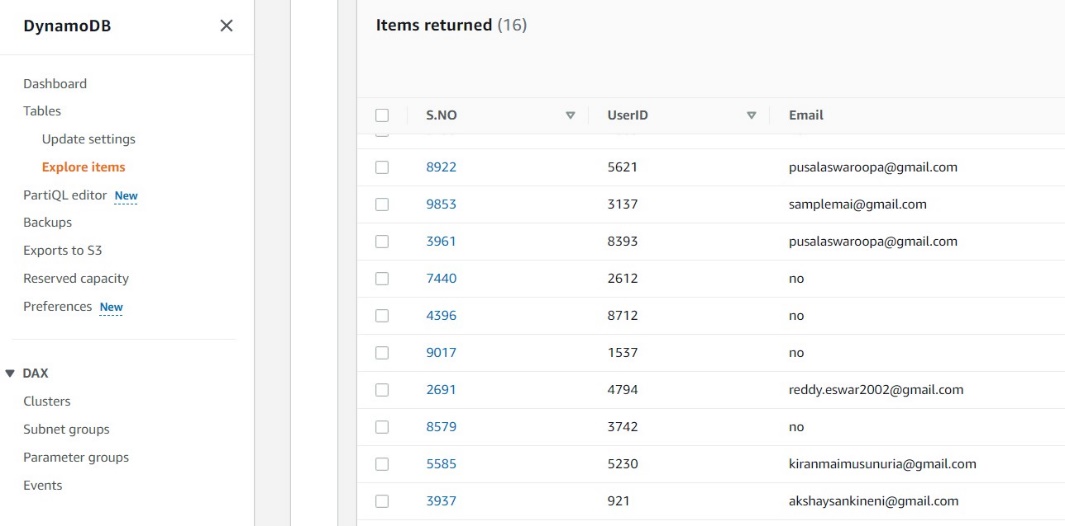
He chose Saturday as the day for his treatment. As doctor is available on that day he proceeded to book an appointment.



As booking is successful appointment id is sendto entered mail id.



All the successful booking information is updated on dynamodb appointment table.

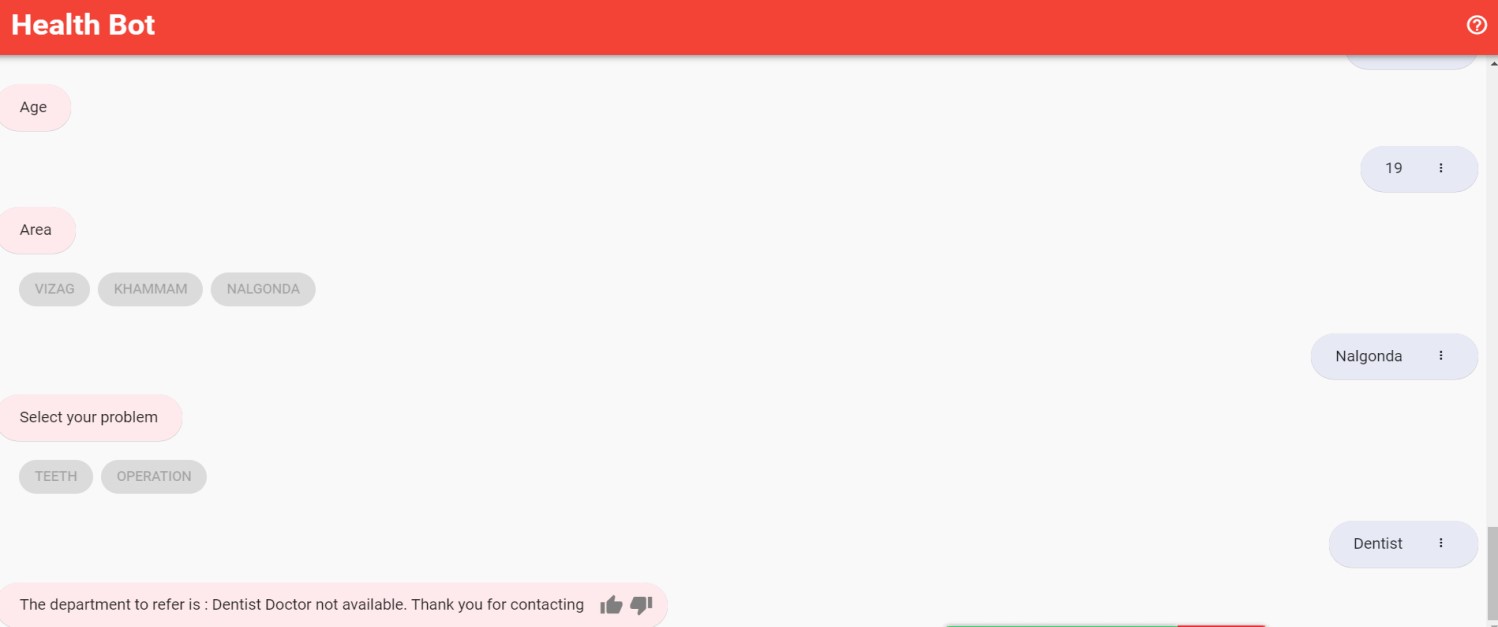


Failure Case:

1. When that specific doctor is not available in the selected area

A user selected nalgonda for his teeth treatment.

As doctor related to that department is not available further can’t proceed with booking. A response is received as Doctor not available.



2. When doctor is not available on the selected day.

A patient need surgery and has chosen Dr. Kumar as his doctor.

Patient chose Thursday as the day for his treatment. As doctor is not available on that day further can’t proceed with booking. A response is received as Doctor not available on that day and a display of all the days doctor available.

