

Chat Bot for Students using AWS

FINAL REVIEW REPORT

Submitted by

Manav Kulshrestha (19BCE0718)

Jaival Shah (19BCE0552)

Devansh Saini (19BCE0412)

Prepared For

Virtualization (CSE4011)

– PROJECT COMPONENT

Submitted To

Dr. Satish C J

Associate Professor

School of Computer Science and Engineering



VIT[®]
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

Table of Contents

1. Introduction	3
2. Project SCOPE	3
3. Key Contacts and Stakeholders	3
4. Project Resource Requirements.....	3
4.1 Software Resource Requirements.....	Error! Bookmark not defined.
5. Architectural Design / UI Design – Block diagram/ Screen Shots	4
6. Implementation.....	5

1. INTRODUCTION

The idea of creating a machine that had human thought process made many scientists, philosophers and even sculptors fascinated of humanistic automation. This system is a web application that responds to the student's question. The chatbot that is utilised only requires students to ask questions. There is no required structure for users to follow; students are free to speak in any format they choose. The System responds to the question using built-in artificial intelligence. The responses are pertinent to the user's inquiries. The user only needs to click the button labelled "invalid answer" to inform the administrator that the given response is untrue. Admins can access the portal's invalid answer section by logging in.

2. PROJECT SCOPE

Our chatbot aims at automation of the following processes

- a. Helps the student to be updated about the college activities
- b. User does not have to personally go to the college for enquiry
- c. College related queries are answered mostly without human involvement
- d. Remote access to every type of problem being answered
- e. Offers 24/7 assistance to students
- f. an effective Graphical user interface which implies that as if a real person is talking to the user.

3. KEY CONTACTS AND STAKEHOLDERS

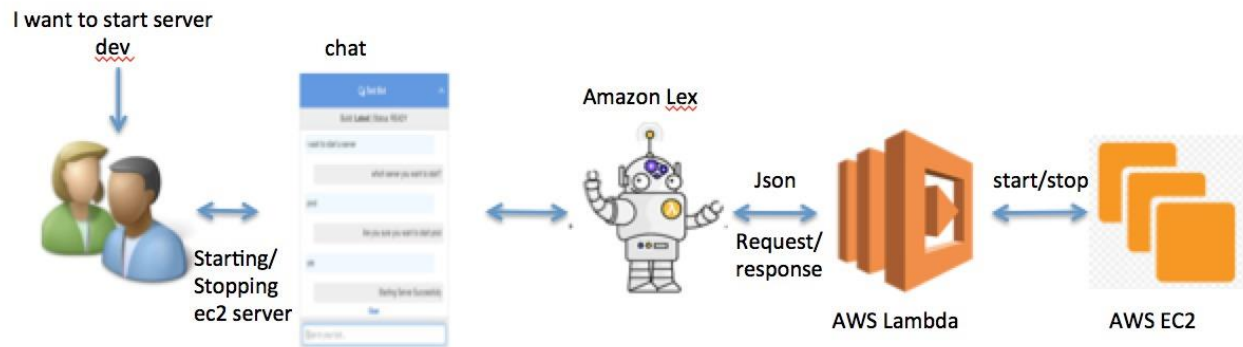
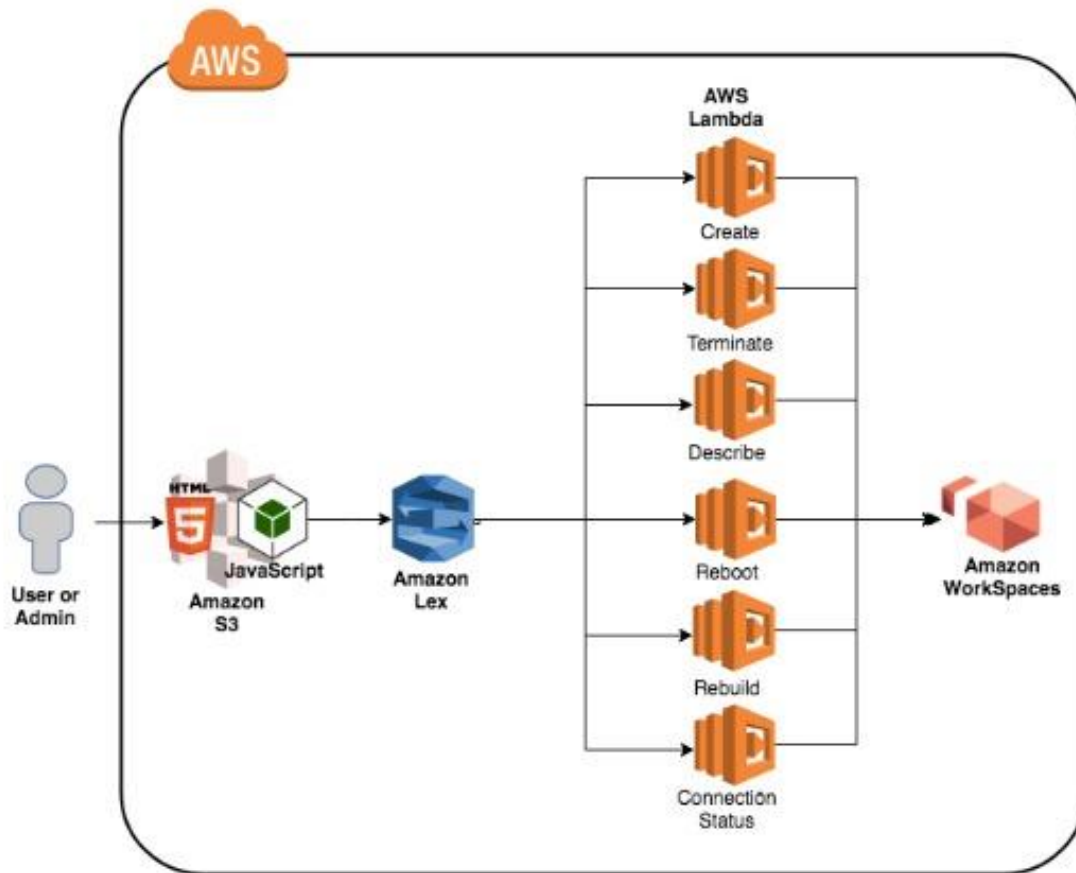
Name	Registration Number	Phone Number
Jaival Shah	19BCE0552	98795 53929
Devansh Saini	19BCE0412	76900 63371
Manav Kulshrestha	19BCE0718	9150474147

4. PROJECT RESOURCE REQUIREMENTS

4.1 Software Resource Requirements

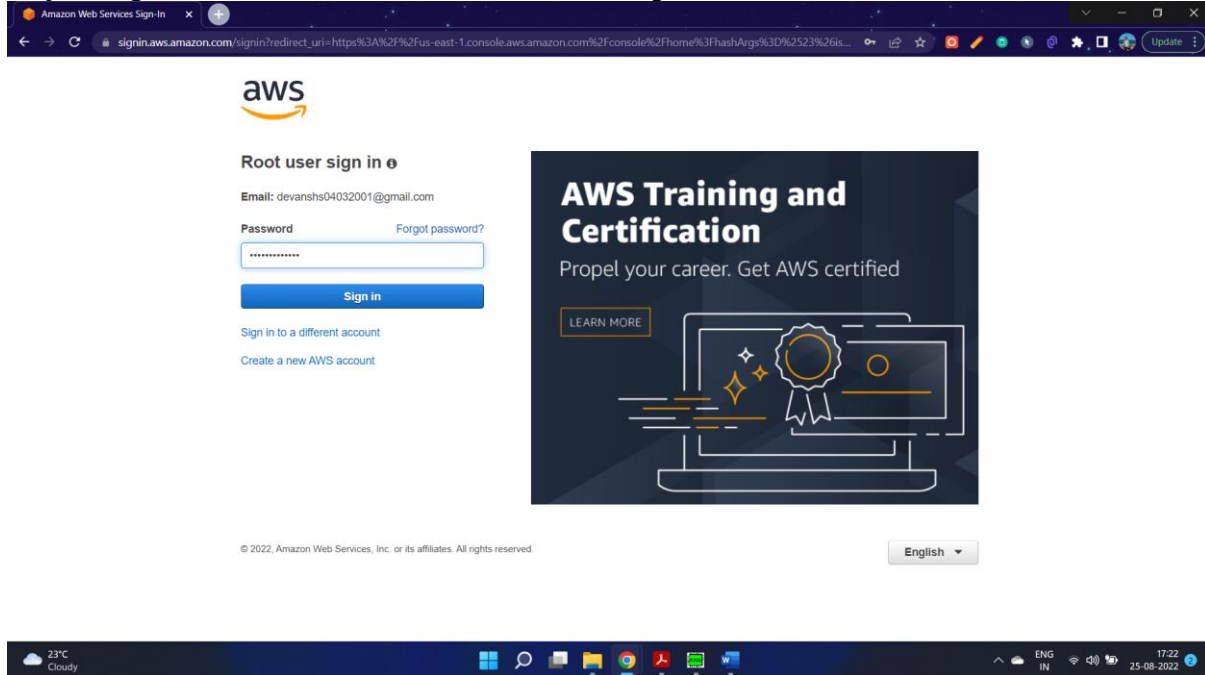
- Windows
- PHP
- CSS

5. ARCHITECTURAL DESIGN / UI DESIGN – BLOCK DIAGRAM/ SCREEN SHOTS

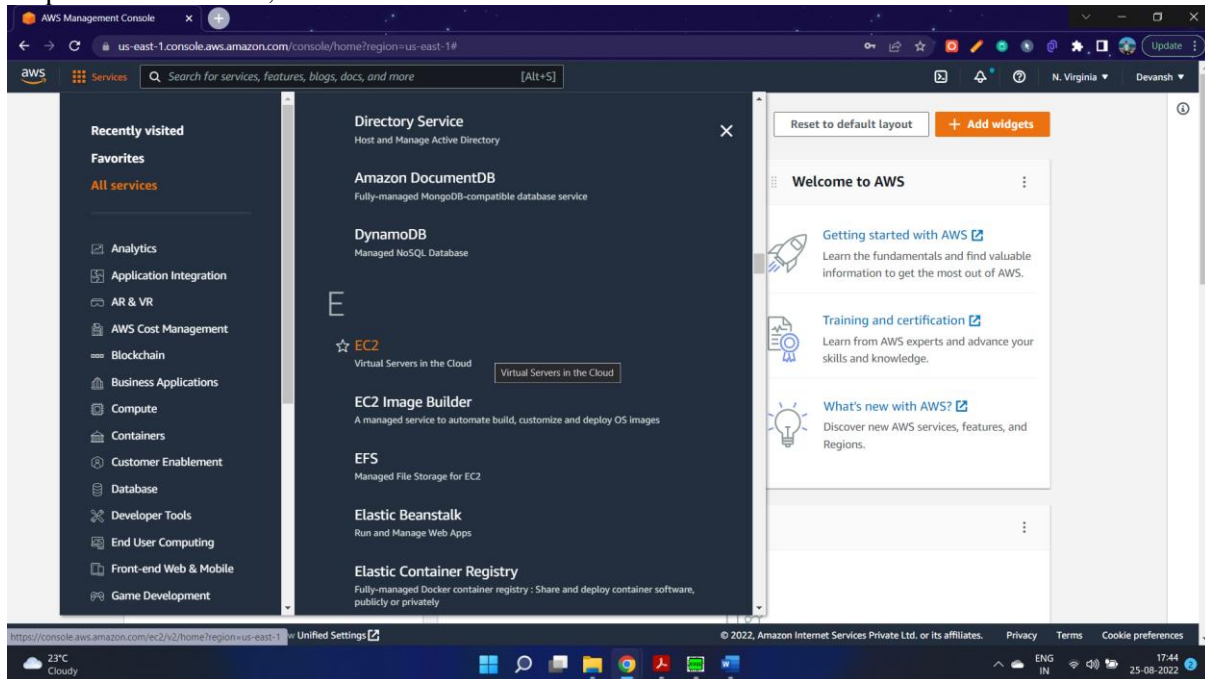


6. IMPLEMENTATION

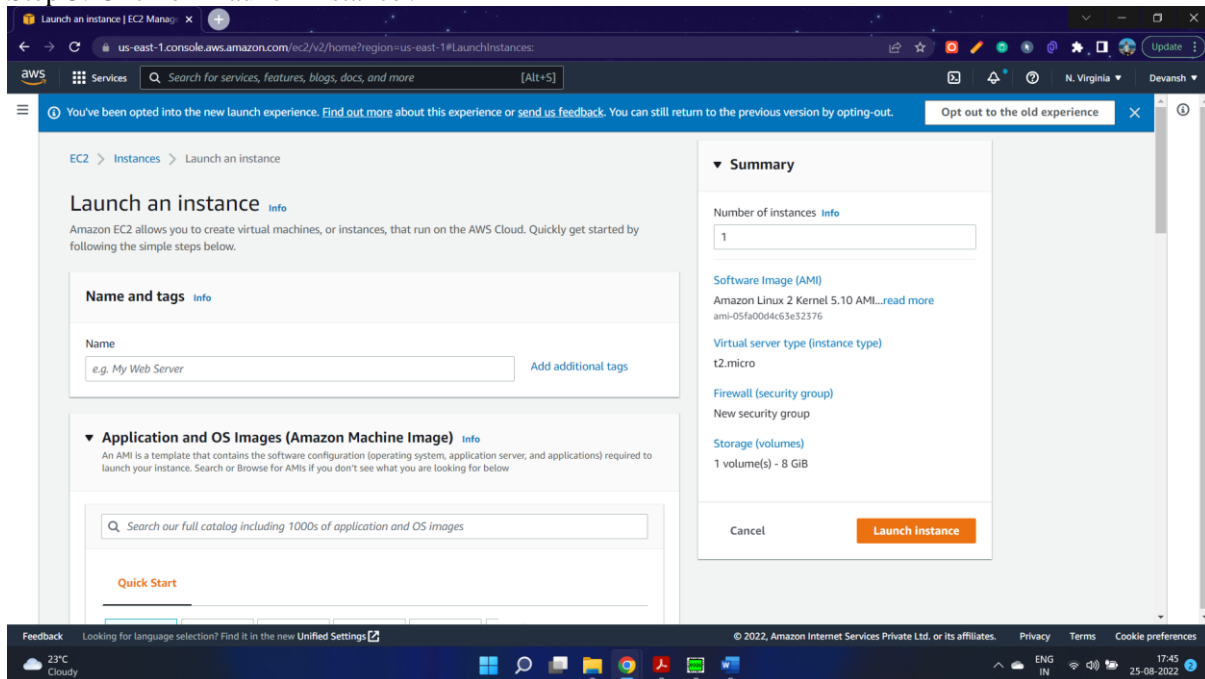
Step 1: Login to AWS Account and then head to Management Console.



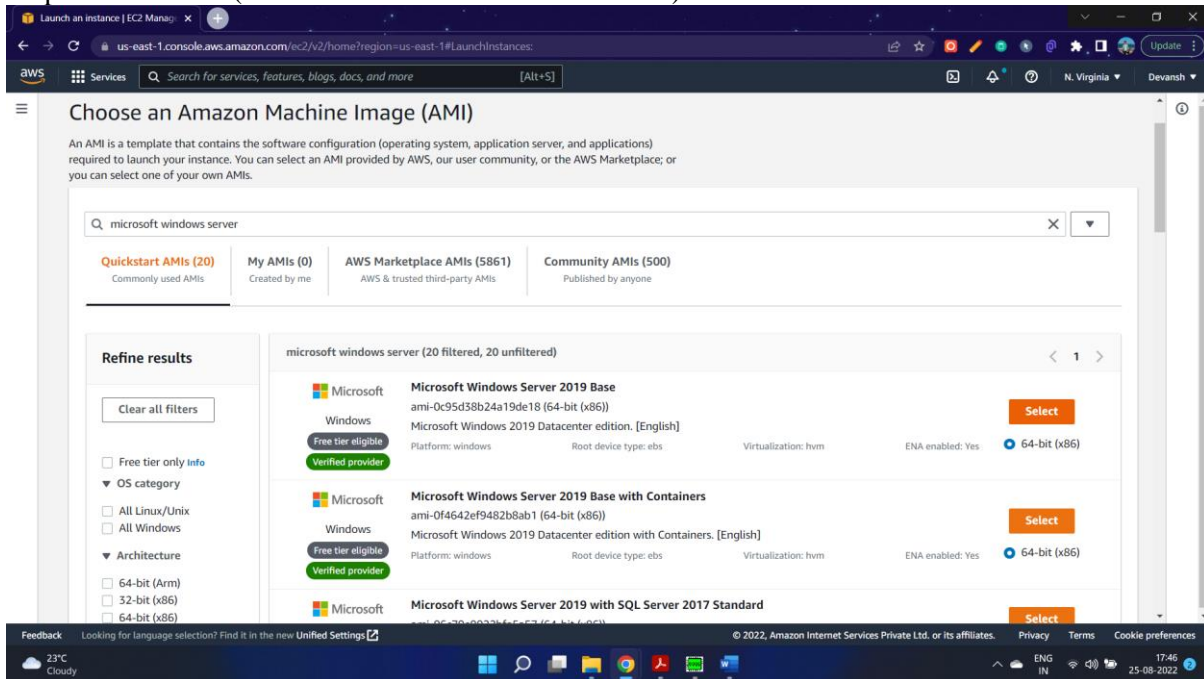
Step 2: From Services, Click on EC2.



Step 3: Click on 'Launch Instance'.

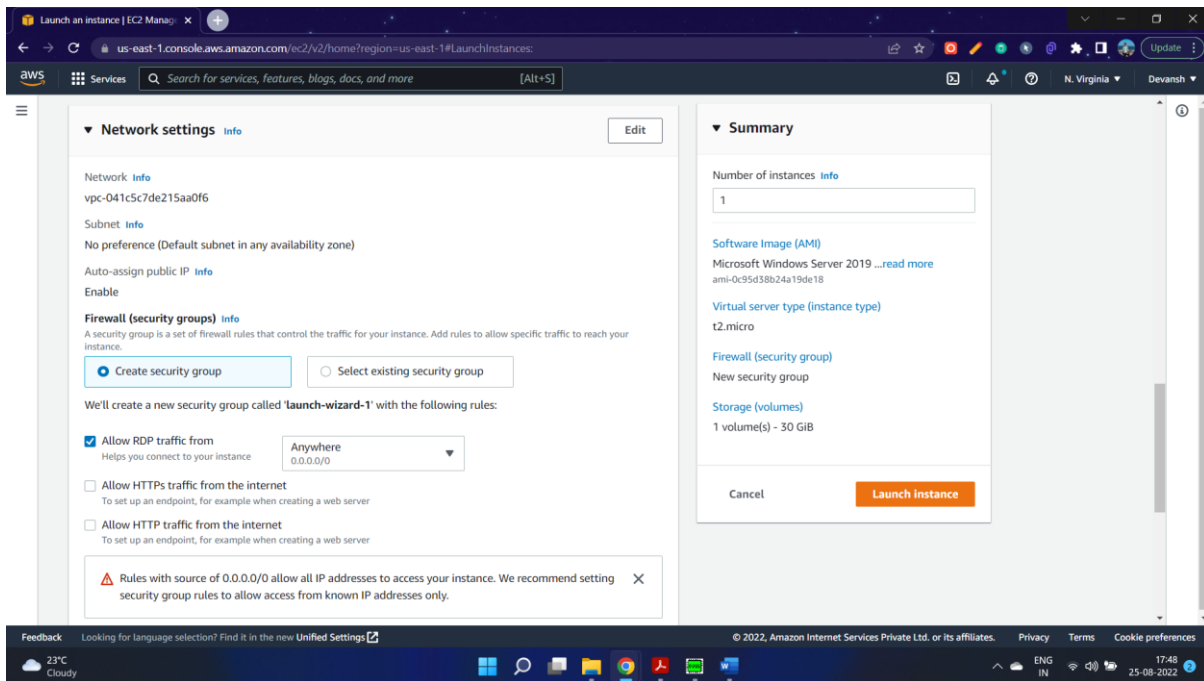


Step 4: Select AMI(microsoft windows server 2019 base).

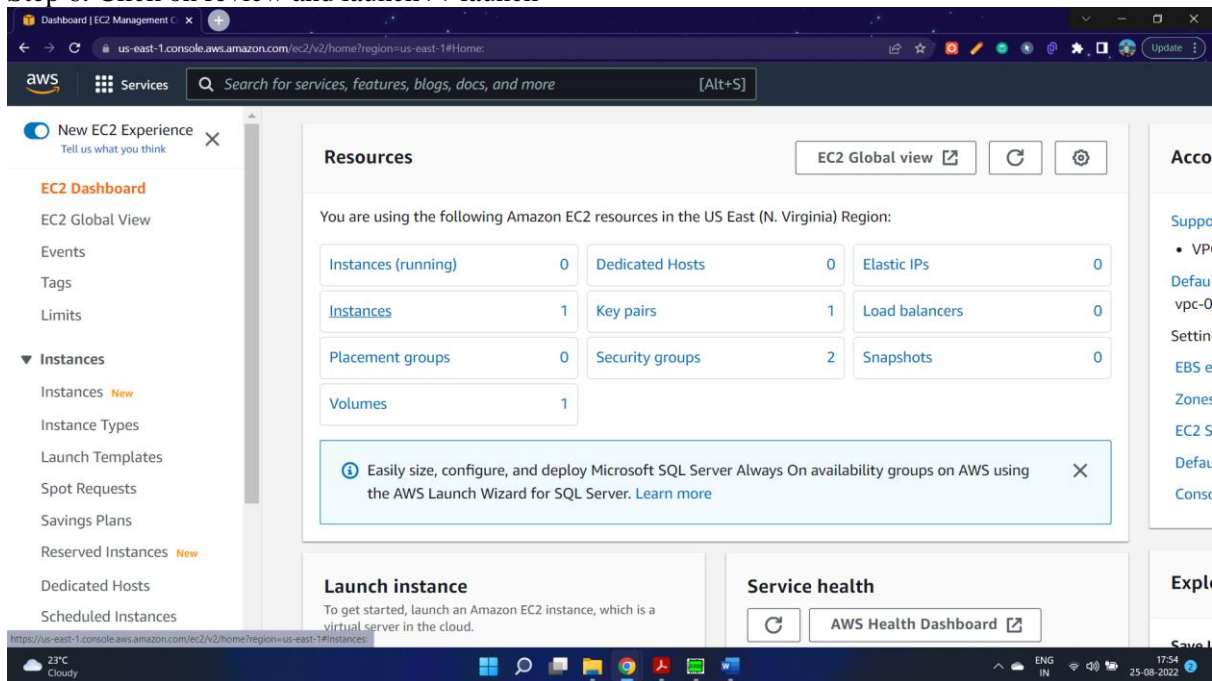


Step 5: In Configure Security group

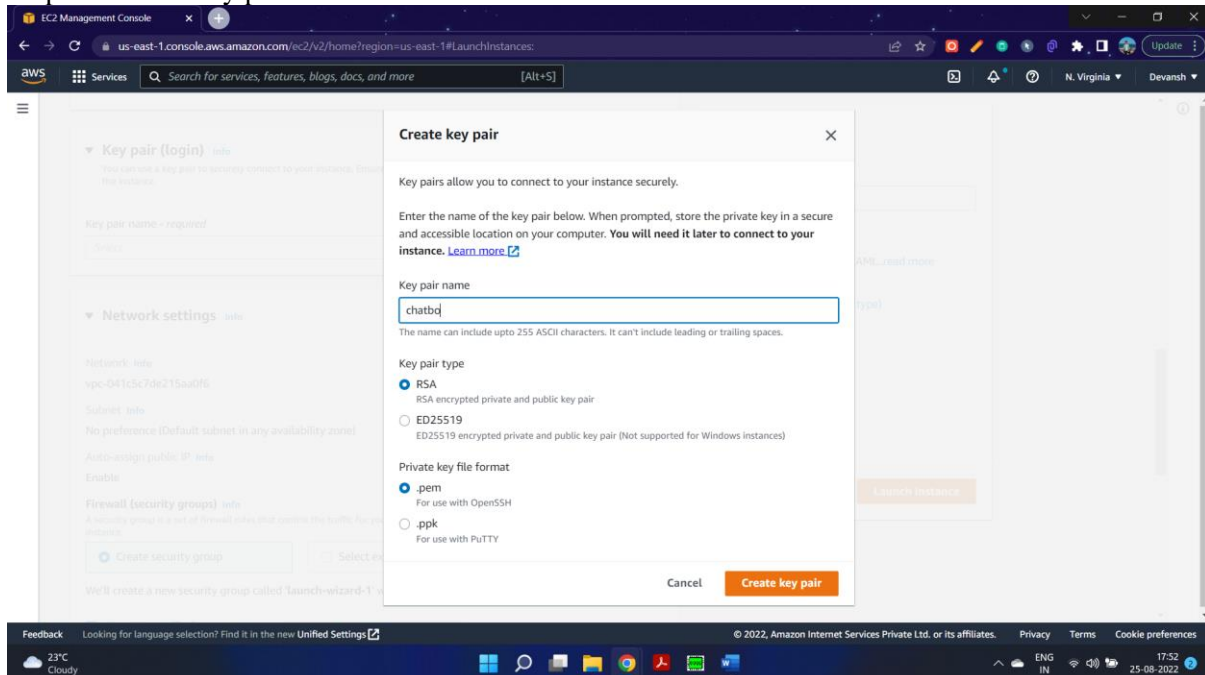
a. Click on create new security group



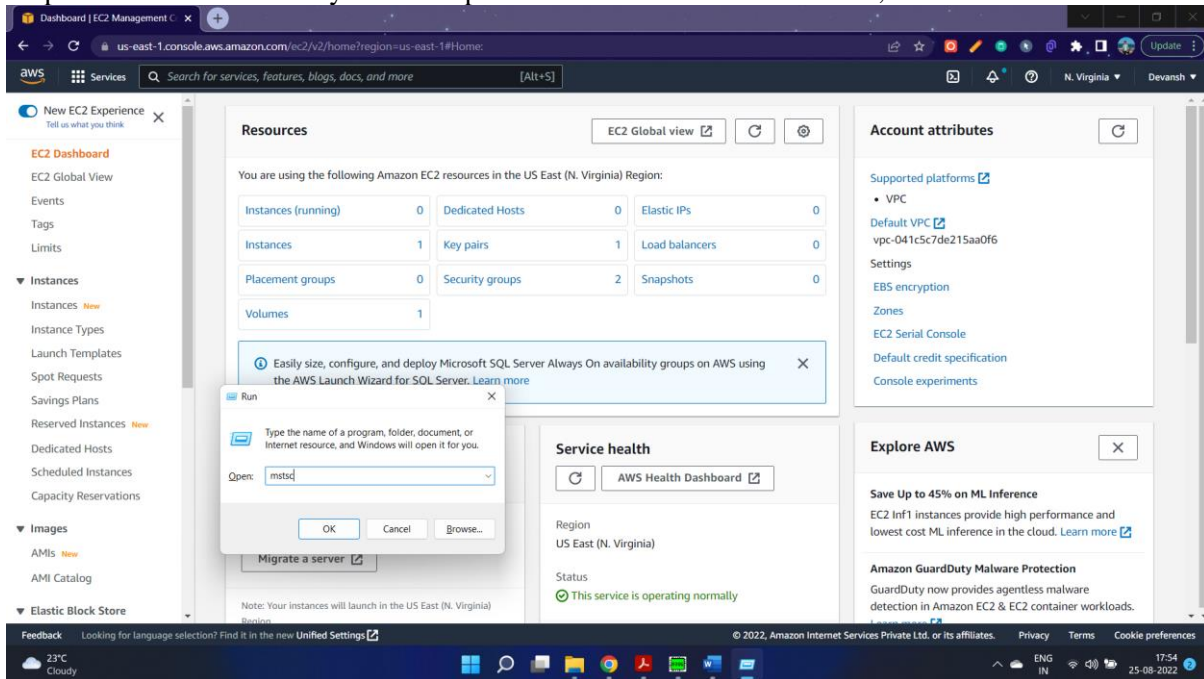
Step 6: Click on review and launch >>launch



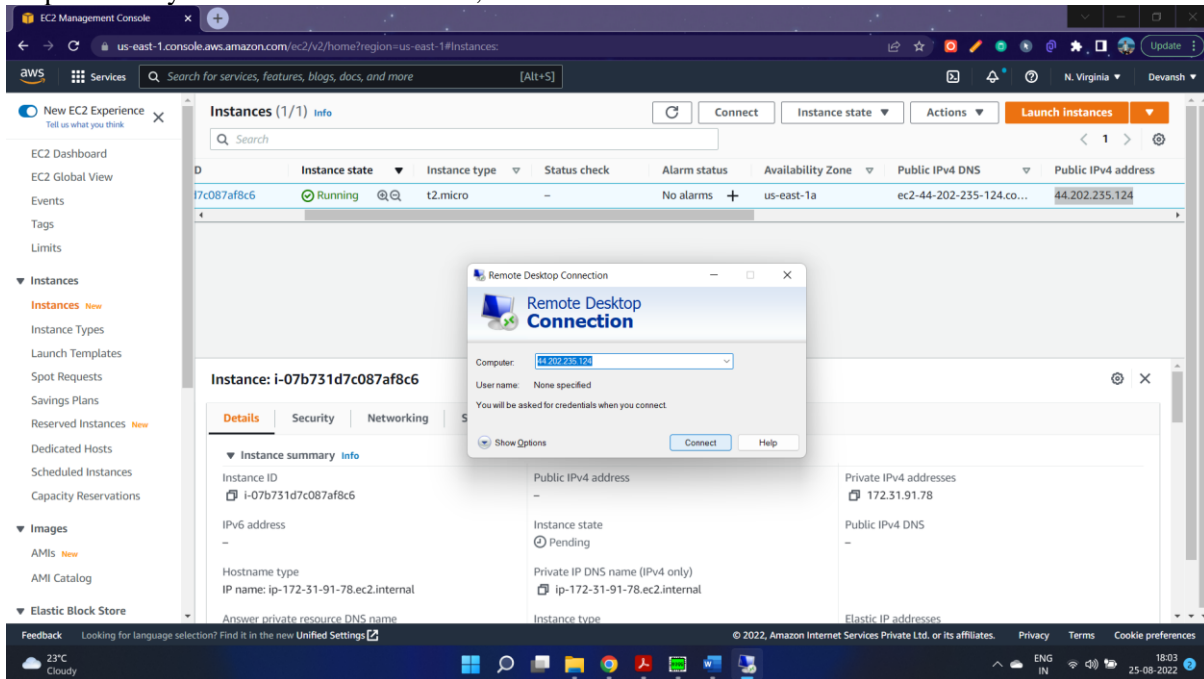
Step 7: create a key pair



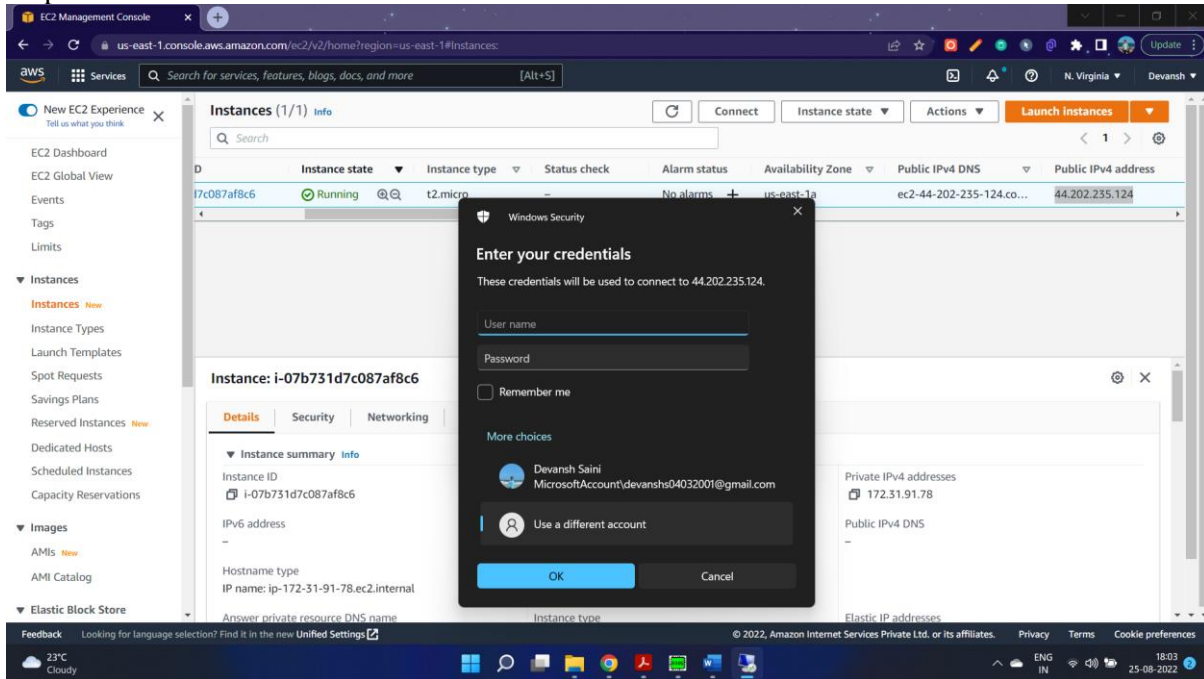
Step 8: Click 'win+R' on keyboard or open Run terminal and write 'mstsc', click ok.



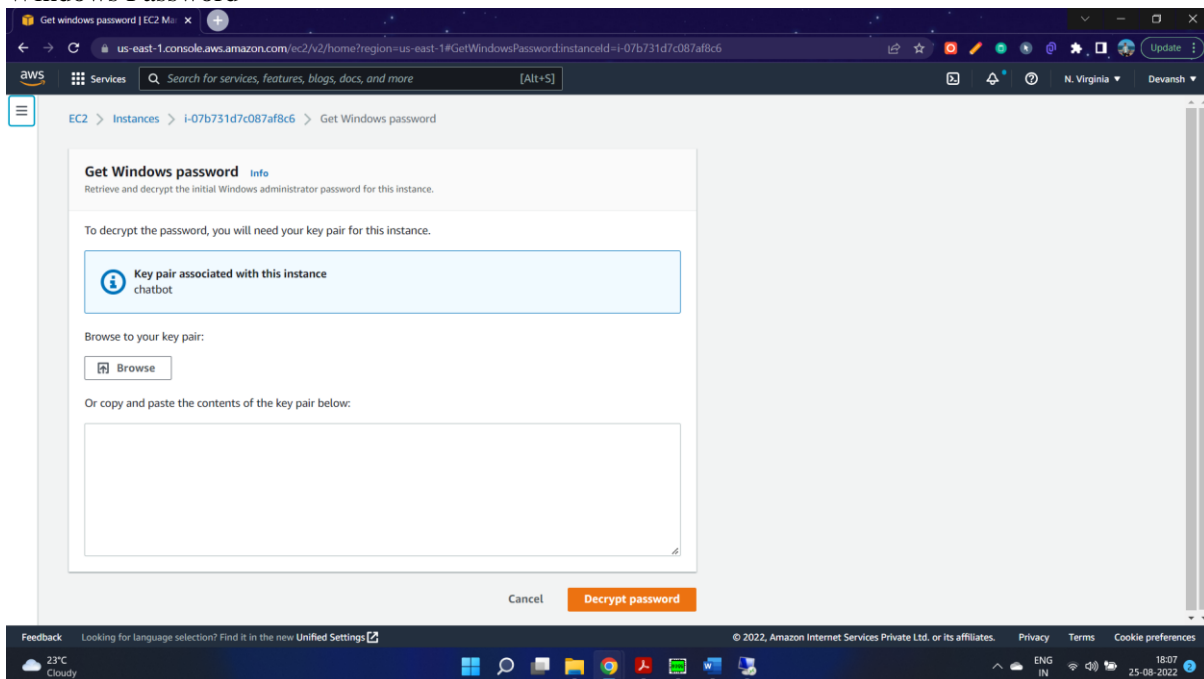
Step 9: Enter your Public IPV4 address ,click connect



Step 10: click on more choices and then use a different account.



Step 11: Now go to the running instance dashboard and select the instance. Click on actions->Security->Get Windows Password



The .pem file:

```
-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEA1+PsbDnT1w5Sn0t9FUI0Fww6bt1jhInHf+fmooV2EtfvT9h
SM5C0Gqs257kbEoyfMnpf7VZ0FTIgwIKPecNVY3Apm2UCHBm0q3s1da5g5bU1Ds
DOGufgAVCJ7f8KAsQD1b8GBXT/c/pwGwQz6C7ZLKr9q3ZFzenVwHfGLBanOmG52
VYTX71X4uc0RUIQ4SReXCV1+dIfaQT0bv22cIPxH06pep1PgVUH8/ZF5d1tZq
3SVC0X1CZVUB01da0r8nMhMFdX10pmf7J0qBfWJ8uQ250Bm7ayPom65YCY
t1bntUu0L81E6Z3UR0UwqBtV5sdFh2V7X010QA0a01BA0d0v1pk1lmfLrpf
8V6Mlv1q9h1h9id7DPTDQdt+4vgx819RffF211o1TF/seiZeh18nJ98F4gbsI
w/CD4Vnc1gEh0m2LGH8pGM1+s07PgfbNSaB4X2+zu0EBUxv/Xf5D1mgw-jdt
b9jkPf0a6tCVT8mzfHdR+wqIRvP013snj1awmtdJni45dvd/Lau+oHKZ1kk4j53s
E+7aB46dmn9HkwIC1DxusmsInHUXv7122atJmh1KhX5nSNK80QY1DK7DFEn0597
K46Wzme/8HAXPK+EOKShpo9vXNc8H71kmgv733D08k81mf7KgrB2BZ81yaovFg
3dxegvKcYEA64EBK8h9y9+1m6Z0Qm0Bh9eZ/lyoxUJH72XgGK10KE0A7JHL
61T1H49W0Q6CE7r+47j0rZASAH7FHL7C5pK0K0K6uCBdZ1h873o0sCwtj
cb/rp1xw550oxD0gt68xbwpgf3eBP850uycurdlaw11A1X1a1zSug8cYEA6q3a
EH6+kSnX4n1bTcmprS/Tnhd1ch2r5JefafBhorV8X2pEwaxHr1dVZAg8n0Q
k/qmzMOCTsbw1A1Jd7nk1XRfT1zRG/2Q212ZfrsFYeg8lyQpfyYvux0039jFpD1
NN15ohN0HYH1Vmuq5CAoRB9+mapYxFmQbwatvMKgYA3H5f0nVHJAFzAKqRQav4
C6sGAPhW7NfeY1TXWjCS+Pg/rpsbZfxbmkB6KTU1Yof123Kw+pZj8pQA01G5KX/
Fg04cG0BumR25pxQ24Z3d0ZPdznvCBKw0r1yzQ7sdudhVrbyrhbXrB5X1D0
9cBktygPp02up02u7j0rBjP0d08+YunPv11n10dG6uHfHfCHf0rG1StG3s+
dbr5Zas0pr50tKEEfp0W0Fz5BUC5ncjPoxE0nuU2Xc0rFFDn0hWkQClWjyF8w
tqhksrq0rF9PF5HARtKt1r3Q1Qu0RXbJfNkq9icnj2r3ctg6gkkt7T8cUw
9H0PA0G8AKn2qWZTSLScyutvgLW1Z00BwGvbx+U5y9f/V0x0V5V0mQ155ZFvSE
+Qs+8E0cXN19Yuh0F8EY0+x+mJ09jtgNRKsvbBUcN1n0C1hR4almzFPT5nqjY0
xN8ndzbEhsZU8gz1eSZsU70w/0BD5CCzUC1Fdr21/9q1r/b3fD0E
-----END RSA PRIVATE KEY-----
```

Step 12: Browse your .pem file and decrypt the password. your default username will be 'Administrator'.

Get Windows password | EC2: x

us-east-1.console.aws.amazon.com/ec2/v2/home?region=us-east-1#GetWindowsPassword:instanceId=i-07b731d7c087af8c6

EC2 > Instances > i-07b731d7c087af8c6 > Get Windows password

Get Windows password info

Retrieve and decrypt the initial Windows administrator password for this instance.

Password change recommended

We recommend that you change your default password. Note: If a default password is changed, it cannot be retrieved using this tool. It is important that you change your password to one that you will remember.

You can use the following information to connect to your Windows instance using Remote Desktop.

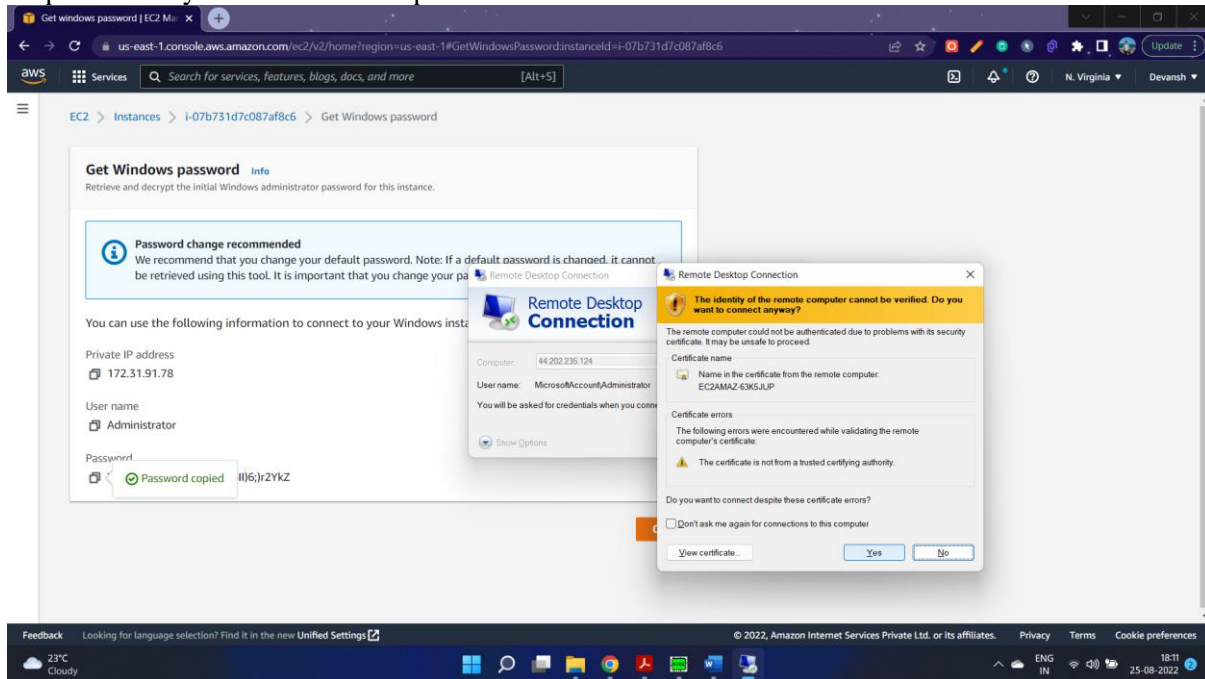
Private IP address
172.31.91.78

User name
Administrator

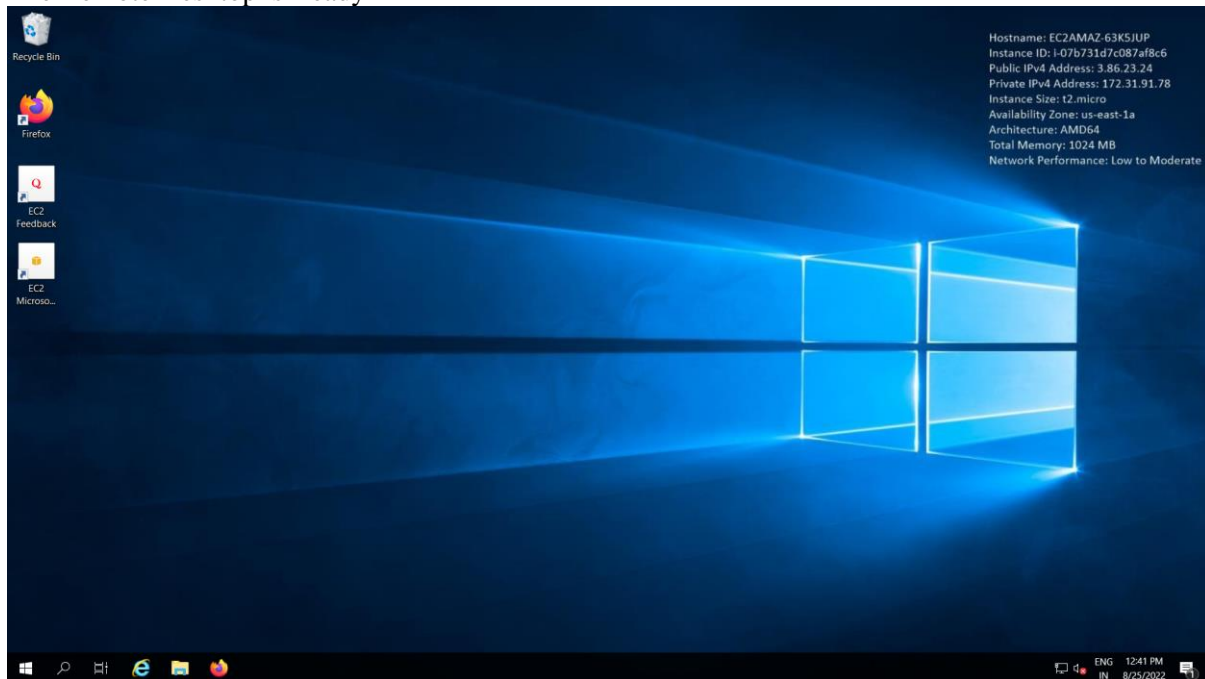
Password
T&Or7dTyLX2Xt=qMIY?O3II)6;)zYkZ

Close

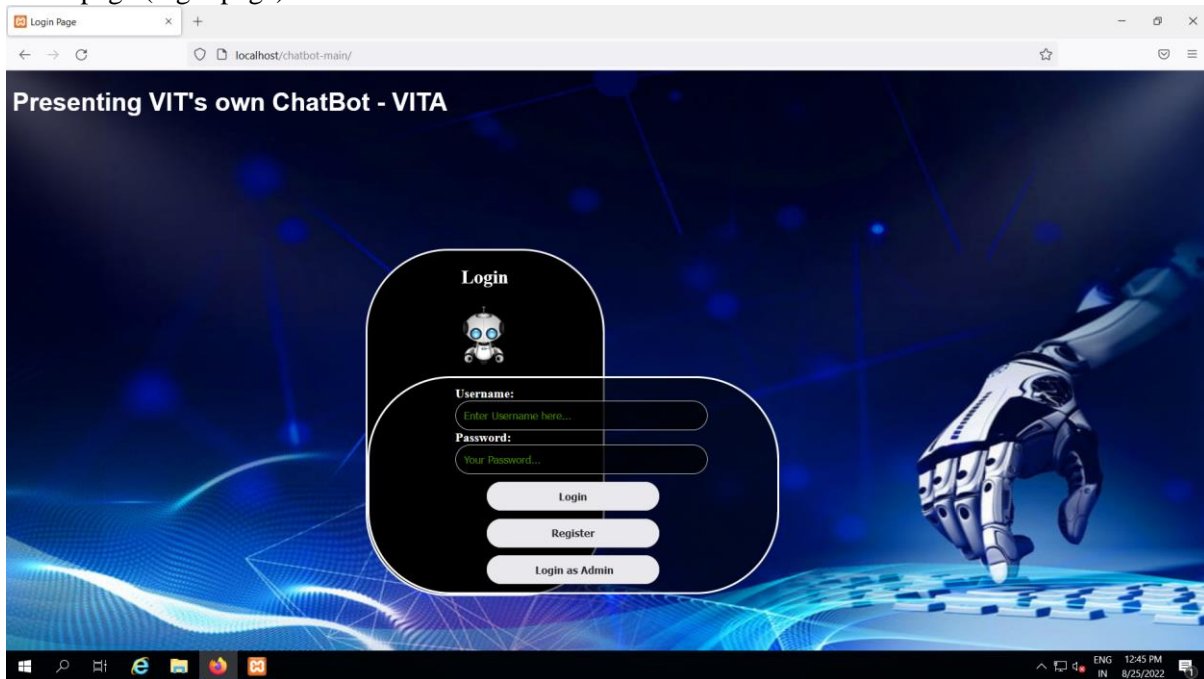
Step 13: Enter your username and password obtained



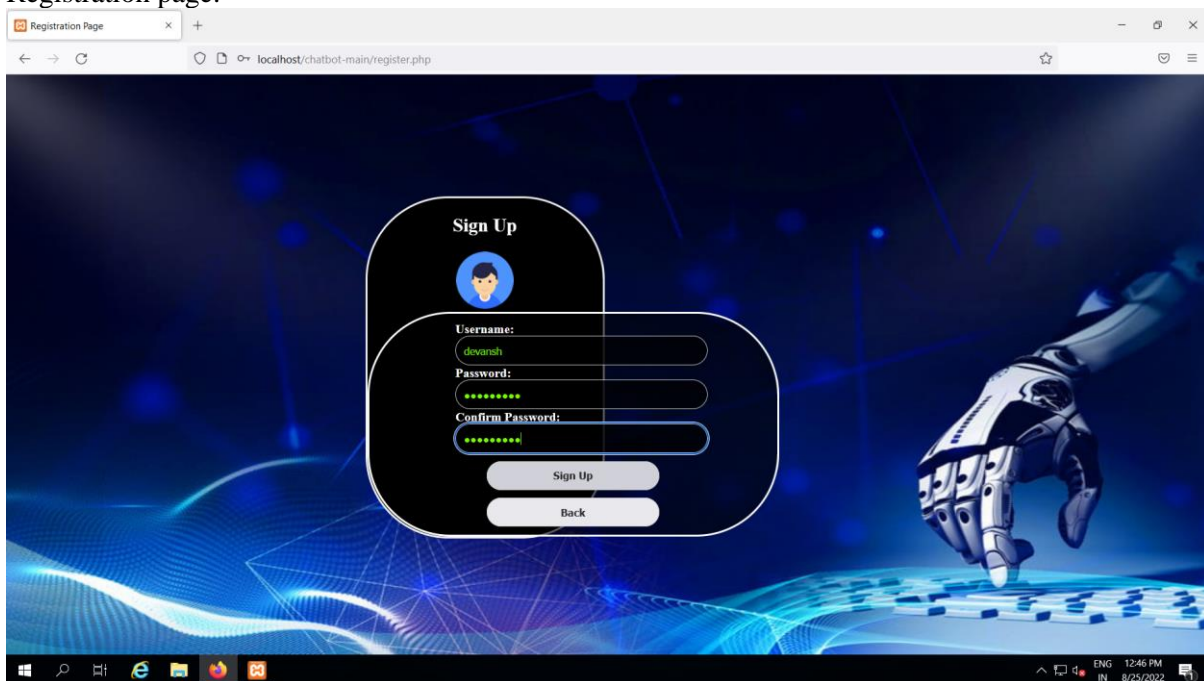
The Remote Desktop is Ready



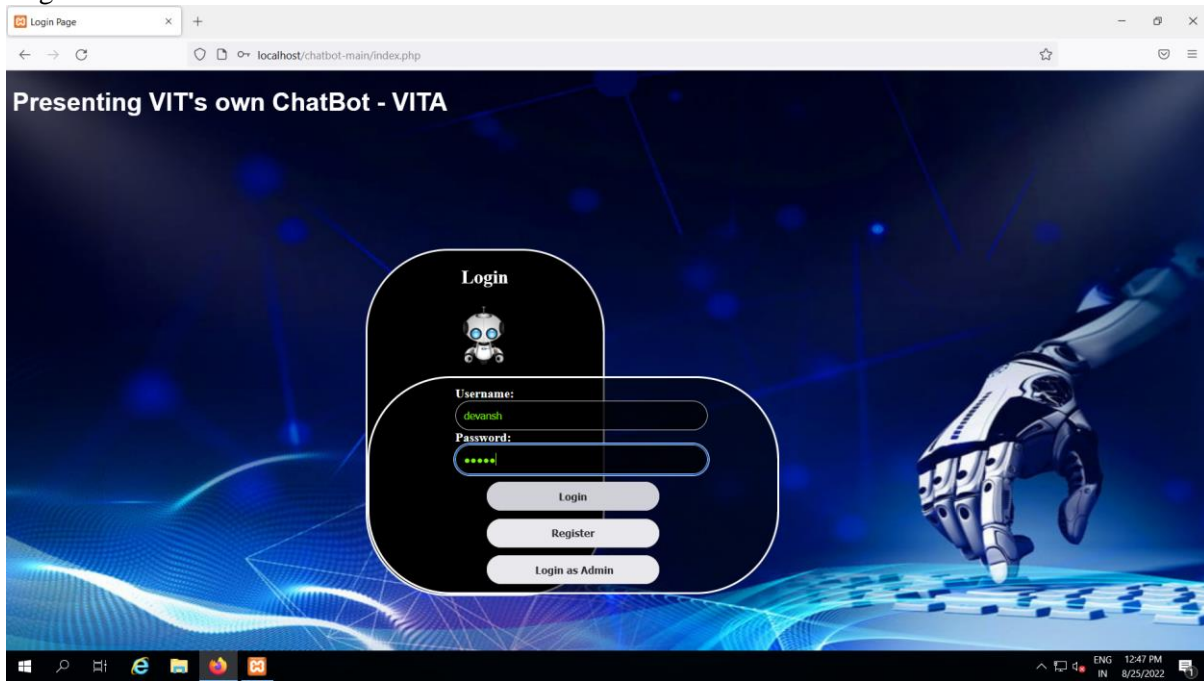
Screenshots of Chatbot for Students
Index page (login page):



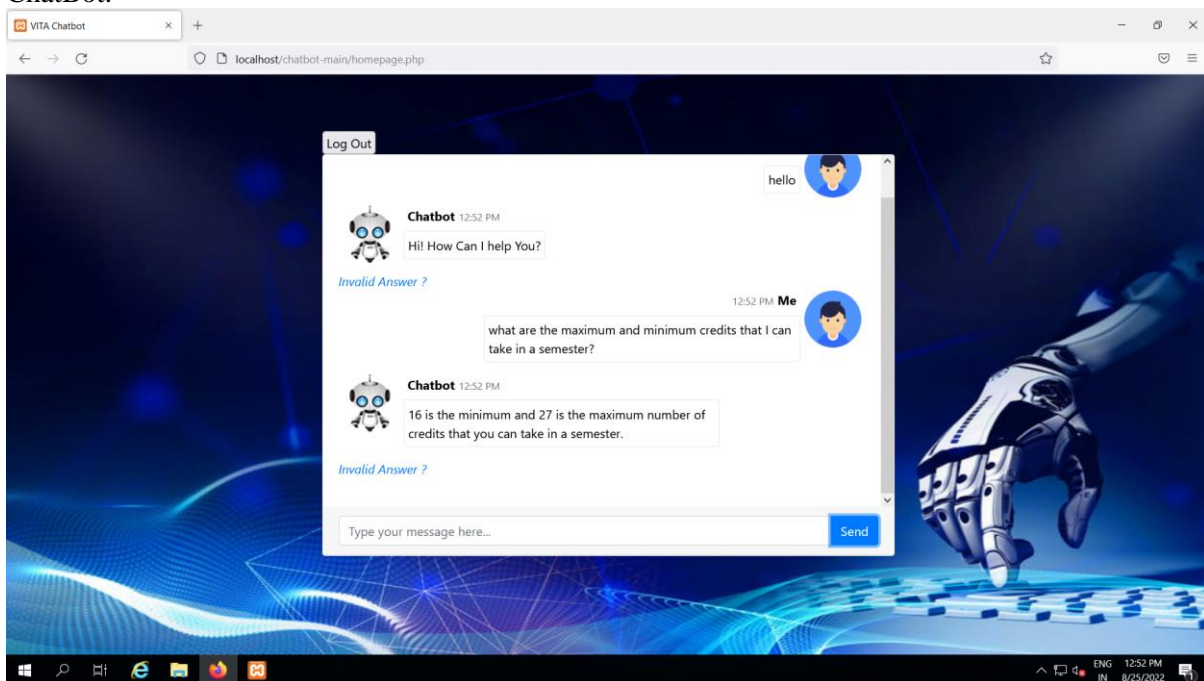
Registration page:



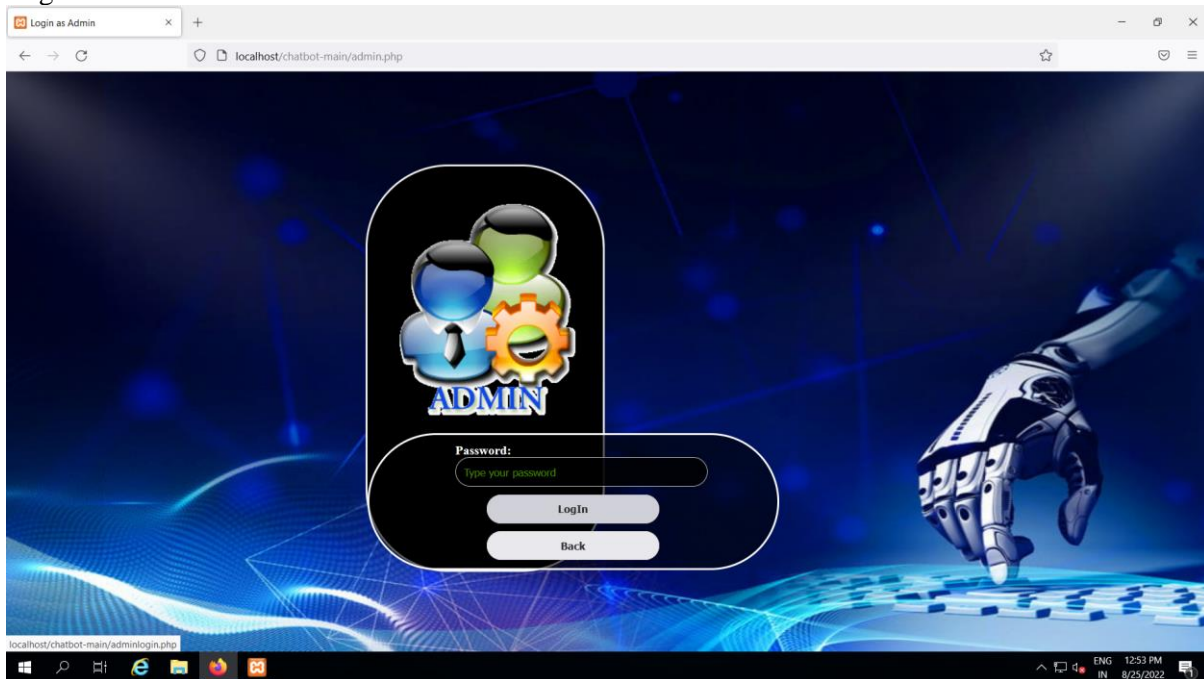
Log in:



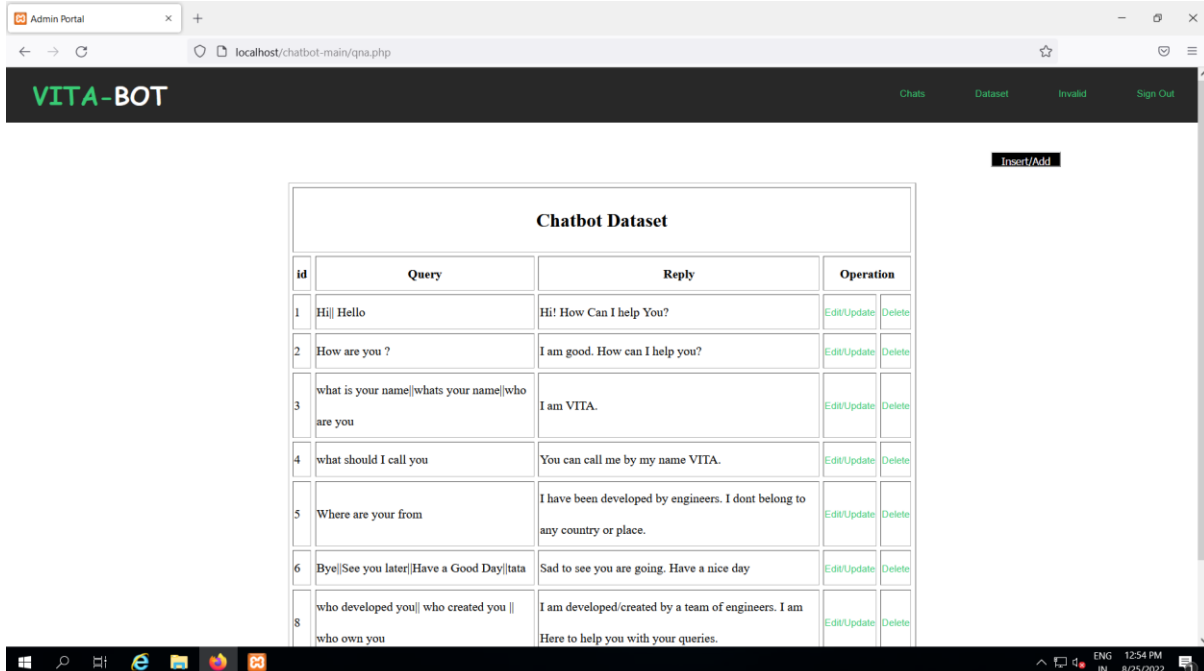
ChatBot:



Login as Admin:



Admin Portal:



Literature Survey:

S.No.	Paper Title	Methodology	Improvement/Conclusion
1	Building a Chatbot with Serverless Computing	Chatbots are emerging as the newest platform used by millions of consumers worldwide due in part to the commoditization of natural language services, which provide developers with many building blocks to create chatbots inexpensively. However, it is still difficult to build and deploy chatbots. Developers need to handle the coordination of the cognitive services to build the chatbot interface, integrate the chatbot with external services, and worry about extensibility, scalability, and maintenance.	We describe our serverless architecture based on function sequences, and how we used these functions to coordinate the cognitive microservices in the Watson Developer Cloud to allow the chatbot to interact with external services. The serverless model improves the extensibility of our chatbot, which currently supports 6 abilities: location based weather reports, jokes, date, reminders, and a simple music tutor.
2	Pregnancy Companion Chatbot Using Alexa and Amazon Web Services	Pregnant ladies often become anxious and get curious about many questions and concerns. Mobile Health services such as chatbots can be used to address these concerns, helping them with authentic healthcare advice; accessible from remote locations irrespective of time and space. The paper describes a pregnancy companion chatbot to help pregnant women during pregnancy. A questionnaire was designed for the expectant mothers and healthcare professionals who treat them, to design the chatbot.	The pregnancy companion chatbot was tested on Amazon Echo dot, a smart speaker device supported by Amazon voice assistant Alexa. Designing a chatbot on top of a custom Alexa skill allows developers to use multiple Amazon Web Services. The design has used AWS Lambda as a triggering function, Simple Notification Service for sending message notifications, Simple Email Service for sending emails and DynamoDB as a database. 'Internet of Things', the concept of connecting smart devices, makes the chatbot solution accessible at any time and from anywhere.
3	Learning System Customer Service Chatbot	Learning System Customer Service Chatbot is a computer application that interacts with users using natural language in a similar way to imitate a technical support representative using voice.	Develop a machine learning voice chatbot system for customer service based on Amazon Lex and Alexa Software Developer Kit (SDK) through extension of an Amazon text based

			example chatbot. The voice chatbot will be based on natural language processing so we can improve human to machine interaction with learning systems customer services.
4	IRJET- COLLEGE ENQUIRY CHATBOT	The chatbots have embedded data that helps them acknowledge the user's question and supply a solution to that. The college enquiry chatbot project is intended exploitation algorithms that interpret user queries and perceive user's message. The college enquiry chatbot project is developed exploitation algorithms that analyze user queries and perceive user message. This technique may be a internet application that gives answers to the student's question	The proposed system was successfully tested to denote its effectiveness and achievability. It basically reduces the paperwork, manpower and time for any individual. In this paper we had developed a system which will interact with the users by means of reducing their time in visiting the college to enquire about the details/information regarding it. The user can chat with the chatbot of any format. The user/student and the Admin are interacted through a chatbot. The questions which are not answered by the chatbot will be updated by the admin.
5	Chatbot for College Website	Chatbot is a program that reenacts a conversation between a user and a machine. The machine has been embedded knowledge to identify the queries asked by the student and other users and making a decision itself to respond to the queries of the student and other users. The user just needs to ask the queries that are answered by the chatbot. These bots will be totally founded on a book-based UI that permits the client to type orders and get message just as content to discourse response. It can be more made sure about when it is coordinated with well-known web services.	The school request chatbots will be fabricated utilizing counterfeit calculations that can undoubtedly comprehend and investigate clients message. The client can pose the inquiries that are any school-related activities through chatbot without genuinely accessible to the school for inquiry. By using Artificial Intelligence, the queries are answered by the system asked by various users. The user simply

			needs to enlist and login to the system. Natural language handling (NLP) is utilized for tokenizing, stemming, and separating the substance of the objection given by the client.
--	--	--	---

Conclusion:

This study is a step further to improve the sustainability of car sharing as an environmental friendly transportation service, thus contributing to sustainability within our society. Moreover, our study provides a valuable method for practitioners. They can generate more data to “stress-test” their system via simulation. This could be helpful to prepare for events (e.g., trade fairs), where a higher vehicle demand is anticipated.

Future Enhancement:

This work presents a research agenda based on interviews with experts in the field in order to guide future work from an industrial viewpoint, as current agenda-setting contributions fail to include this perspective. Also, we derived guidelines based on expert interviews to provide insights for companies seeking to implement chat-bots in customer care. To explore, our expert panel and literature review showed us that there is a broad consensus that intelligent chat-bots bare great potential for creating value for companies in customer care. While we see the existing research as highly important, our literature review also indicates that the industrial perspective should be more strongly included in identifying needed research. In this way, practice-based requirements can provide insights that may not have been captured in scientific literature, and thus advance the IS research discipline.

References:

1. <https://smartech.gatech.edu/handle/1853/59561>
2. <https://ieeexplore.ieee.org/abstract/document/9105762>
3. <https://dl.acm.org/doi/abs/10.1145/3007203.3007217>
4. https://d1wqtxts1xzle7.cloudfront.net/64415574/IRJET-V7I3140-with-cover-page-v2.pdf?Expires=1661625106&Signature=WXamBR2K62oolI-IKengDDFdtrKyPsVfIQQNcyaUN3pgO7dpYEjWjosgtc35STrY4W1Ko0zCz1-iAjlqENksfDG5oTLLC9tX25nhSSzD9plRqaLTQjW19LecY1xB0G9wDKjzLFGD1rWqiEI75uUpikKQofMqITvLORUhsHW8CVifCbFjjtmK-29WobJfGs6kYBQW081f4offphT1A0kuP-ZhyrkXP5QLZHiG7ekxGcM7wxa761FvSHFFP6pJMzxQKT0CvyGq3aJWwpsGLIjP6dJtdm3ltKKxMcID7ttgrOFrzQs1BZ3Yr~1XFLnh9EDuLr2pSfpHKLd5cCiGSBZQ_&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA
5. https://link.springer.com/chapter/10.1007/978-981-15-9293-5_47