CLOUD APP DEVELOPMENT

SKILL & JOB RECOMMENDER

Project Report Format

PROJECT REPORT

1.INTRODUCTION

Having lots of skills but wondering which job will best suit you?Don't need to worry! We have come up with a skill recommender solution through which the fresher or the skilled person can login and find the jobs by using the search option or they can directly interact with the chatbot and get their dream job.

1.1 PROJECT OVERVIEW

There has been a sudden boom in the technical industry and an increase in the number of good startups. Keeping track of various appropriate job openings into industry names has become increasingly trouble some.This leads to deadlines and hence important opportunities being missed.

Through this research paper, the aim is to automate this process to eliminate this problem. To achieve this, IBM cloud services like db2, Watson assistant , cluster, kubernetes have been used. A hybrid system of Content-Based Filtering and Collaborative Filtering is implemented to recommend these jobs. The intention is to aggregate and recommend appropriate jobs to job seekers, especially in the engineering domain. The entire process of accessing numerous company websites hoping to find a relevant job opening listed on their career portals is simplified.

The proposed recommendation system is tested on an array of test cases with a fully functioning user interface in the form of a web application. It has shown satisfactory results, outperforming the existing systems. It thus testifies to the agenda of quality over quantity

1.2 PURPOSE

With an increasing number of cash-rich, stable, and promising technical companies/startups on the webwhich are in much demand right now, many candidates want to apply and work for these companies. They tend to miss out on these postings because there is an ocean of existing systems that list millions of jobs which are generally not relevant a tall to the users. There is an abundance of choices and not much streamlining. On the basis of the actual skills or interests of an individual, job seekers often find themselves unable to find the appropriate employment for themselves.

This system,therefore,approaches the idea from a data point of view, emphasizing more on the quality of the data than the quantity.

2.LITERATURESURVEY

2.1EXISTINGPROBLEM

Existingsystemis not very efficient,it does not benefit the user in maxim way,so the proposed system usesibmcloudserviceslikedb2,Watsonvirtual assistant , cluster , kubernetes and docker for containerization of the application.

2.2 REFERENCES

ShahaTAl-OtaibiandMouradYkhlef.“Asurveyofjobrecommendersystems”. In: International Journal of the Physical Sciences 7.29 (2012), pp. 5127–5142. issn: 19921950. doi: 10.5897/IJPS12. 482

1.N Deniz,A Noyan,and OG Ertosun.“LinkingPerson jobFittoJobStress:The Mediating Effect of Perceived Person

organization Fit”. In: Procedia - Social and Behavioral Sciences
207 (2015), pp. 369– 376.

2. M Diaby, E Viennet, and T Launay. “Toward the next generation of recruitmenttools:Anonlinesocialnetwork-basedjobrecommendersystem”. In: Proc. of the 2013 IEEE/ACM Int. Conf. on Advances in Social Networks Analysis and Mining, ASONAM 2013 (2013), pp. 821–828. doi: 10. 1145/2492517.2500266.

3. M Diaby and E Viennet. “Taxonomy-based job recommender systems on FacebookandLinkedInprofiles”.In:Proc.ofInt.Conf.onResearchChallenges in Information Science (2014), pp. 1–6. issn: 21511357. doi: 10.1109/RCIS.2014.6861048.

4. MKusneretal.“Fromwordembeddingstodocumentdistances”.In:Proc.of the 32nd Int. Conf. on Machine Learning, ICML’15. 2015, pp. 957–966.

5. T Mikolov et al. “Distributed Representations of Words and Phrases and TheirCompositionality”.In:Proc.ofthe26thInt.Conf.onNeuralInformation ProcessingSystems-Volume2.NIPS’1
LakeTahoe,Nevada,2013,pp.3111– 3119. url:
http://dl.acm.org/citation.cfm?id=2999792. 2999959.

6. TMikolovetal.“Efficientestimationofwordrepresentationsinvector space”. In: arXiv preprint arXiv:1301.3781 (2013).

7. GSaltonandCBuckley.“Term-weightingapproachesinautomatictext retrieval”.In:InformationProcessingandManagement24.5(1988),pp. 513–523.issn:0306-4573.doi:https://doi.org/10.1016/0306-4573(88)90021-0. url:http://www.sciencedirect.com/science/article/pii/030645738890021

2.3 PROBLEM STATEMENT DEFINITION

"Can an efficient recommender system be mode led for the Jobseekers which issue of cold start?".

In current situation recruitment s done manually for lakhs of students in which many talented students may lose their opportunities due to different reasons since It is done manually,and company also need the highly talented people from the mass group for their growth. So we have build a cloud application to do this process in a efficient manner.

3.IDEATION AND PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS

What is an empathy map canvas?

An empathy map canvas is a more in-depth version of the original empathy map, which helps identify and describe the user’s needs and pain points. And this is valuable information for improving the user experience.

Teams rely on user insights to map out what is important to their target audience, what influences them, and how they present themselves. This information is then used to create personas that help teams visualize users and empathize with them as individuals, rather than just as a vague marketing demographic or account number.

Who uses an empathy map canvas?

1. Agile teams in a variety of departments use empathy map canvases to better understand how to meet their customers’ needs.

2. Design teams use them to help understand the various reasons why a user might interact with the product so they can design a user
friendly experience.

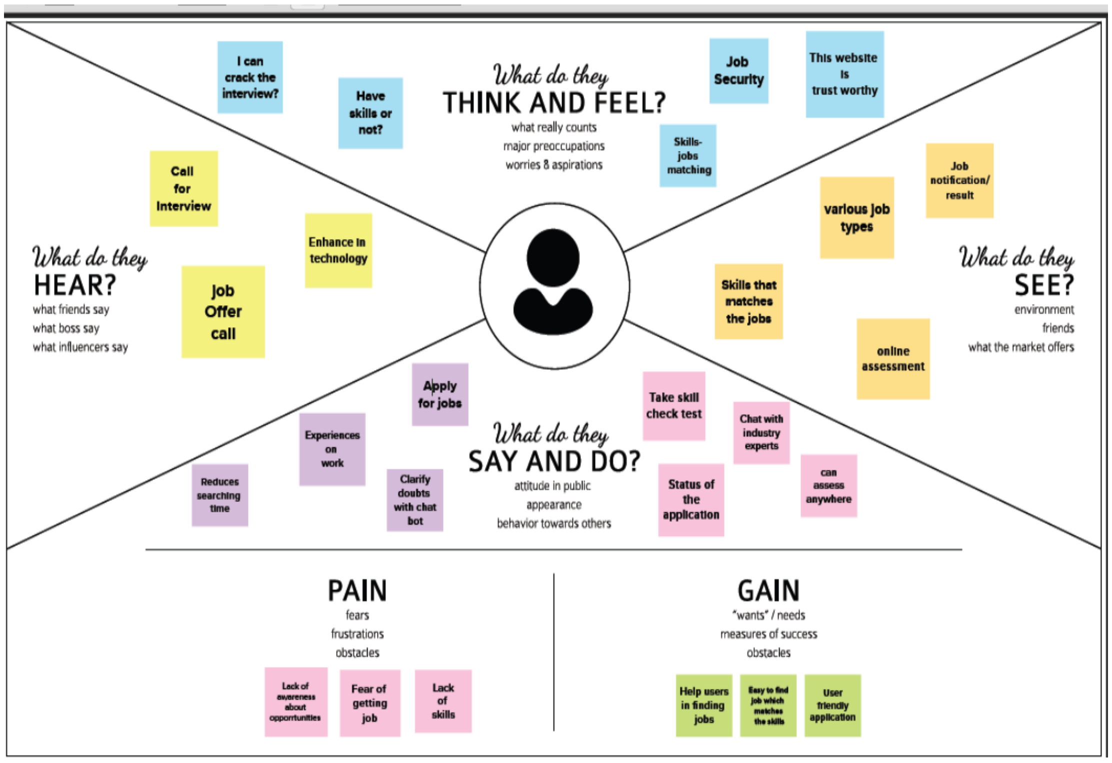
3. Sales teams use them to learn who customers are at an individual level so they can help them invest in a product that suits their needs, rather than leading with a sales pitch that might be off-putting or not appropriately tailored to customers.

Why use an empathy map canvas?

An empathy map canvas helps brands provide a better experience for users by helping teams understand the perspectives and mindset of their customers. Using a template to create an empathy map canvas reduces the preparation time and standardizes the process so you create empathy map canvases of similar quality.

An empathy map is a collaborative visualization used to
articulate what we
knowaboutaparticulartypeofuser.Itexternalizesknowledgeaboutus
ersin order to

1. Create a shared understanding of user needs, and
2. Aid in decision making



3.2 IDEATION AND BRAINSTROMING

The whole creative process of coming up with and communicating new ideas.

What is Ideation?

Ideation essentially refers to the whole creative process of coming up with and communicating new ideas. Ideation is innovative thinking, typically aimed at solving a problem or providing a more efficient means of doing or accomplishing something. It encompasses thinking up new ideas, developing existing ideas, and figuring out means or methods for putting new ideas into practice.

In the business world, ideation is associated with things such as inventing and/or developing new products or services or creating new means or methods of production or delivery of products or services. Amazon’s “Prime” two-day delivery service is an example of ideation being used to address the question of how to serve consumers more efficiently.

Companies such as Apple (NASDAQ: AAPL) are known for encouraging constant innovation by allowing employees to spend a significant percentage of their time working on personal ideas or projects they may be related to developing new products or services.

Ideation is frequently part of what is known as the “design process,” which is the process of developing a plan for producing a new product or creating a new operating system. It may also include detailing or mapping out precisely how a new system or process will be implemented.

Summary

• Ideation refers to the whole creative process of coming up with and communicating new ideas.

• It can take many different forms, from coming up with a totally new idea to combining multiple existing ideas to create a new process or organizational system.

• Ideation is similar to a practice known as brainstorming.

How Ideation Works

Ideation may present itself in any one of a wide variety of ways and arenas. The book “Ideation: The Birth and Death of Ideas,” written by Douglas Graham and Thomas Bachmann, lists several different forms that ideation may take, including the following:

• **Solving Problems** – Ideation is often specifically aimed at problem-solving. For example, production managers at a company may be charged with coming up with ideas on how to reduce production costs.

• **Derivative Ideation** – Derivative ideation refers to building on an existing idea, such as developing complementary products or accessories to sell along with a company's main product.

• **Innovation** – An example of innovation ideation is the process of a pharmaceutical company developing new medicines. Such a type of ideation often involves doing extensive research and experimentation as part of the ideation process.

• **Development of a "Revolutionary Idea"** – Ideation sometimes ends up creating a totally new line of thought or set of ideas, such as the development of a new philosophy.

• **Serendipitous Ideation** – Serendipitous ideation refers to situations where someone just happens to come up with a new idea even though they weren't consciously trying to do so.

• **Combination Ideation** – Ideation often includes combining multiple ideas to create a new process or way of doing something.

Ideation and Brainstorming

Ideation is often closely related to the practice of brainstorming, a specific technique that is utilized to generate new ideas. A principal difference between ideation and brainstorming is that ideation is commonly more thought of as being an individual pursuit, while brainstorming is almost always a group activity. Brainstorming is usually conducted by getting a group of people together to come up with either general new ideas or ideas for solving a specific problem or dealing with a specific situation.

For example, a major corporation that recently learned it is the object of a major lawsuit may want to gather together top executives for a brainstorming session on how to publicly respond to the lawsuit being filed.

Participants in a brainstorming session are encouraged to freely toss out whatever ideas may occur to them. The thinking is that by generating a large number of ideas, the brainstorming group is likely to come up with a suitable solution for whatever issue they are addressing.

The lines between ideation and brainstorming have become a bit more blurred with the development of several brainstorming software programs, such as Brightidea and Ideawake. These software programs are designed to encourage employees of companies to generate new ideas for improving the companies' operations and, ultimately, bottom-line profitability.

The programs often combine the processes of ideation and brainstorming in that individual employees can use them, but companies may simulate brainstorming sessions by having several employees all utilize the software to generate new ideas intended to address a specific purpose.

More Resources

CFI is the official provider of the global Commercial Banking & Credit Analyst (CBCA)™ certification program, designed to help anyone become a world-class financial analyst. To keep advancing your career, the additional resources below will be useful:

- Communication
- Emotional Intelligence
- Hard Skills
- Products and Service

STEP 1:

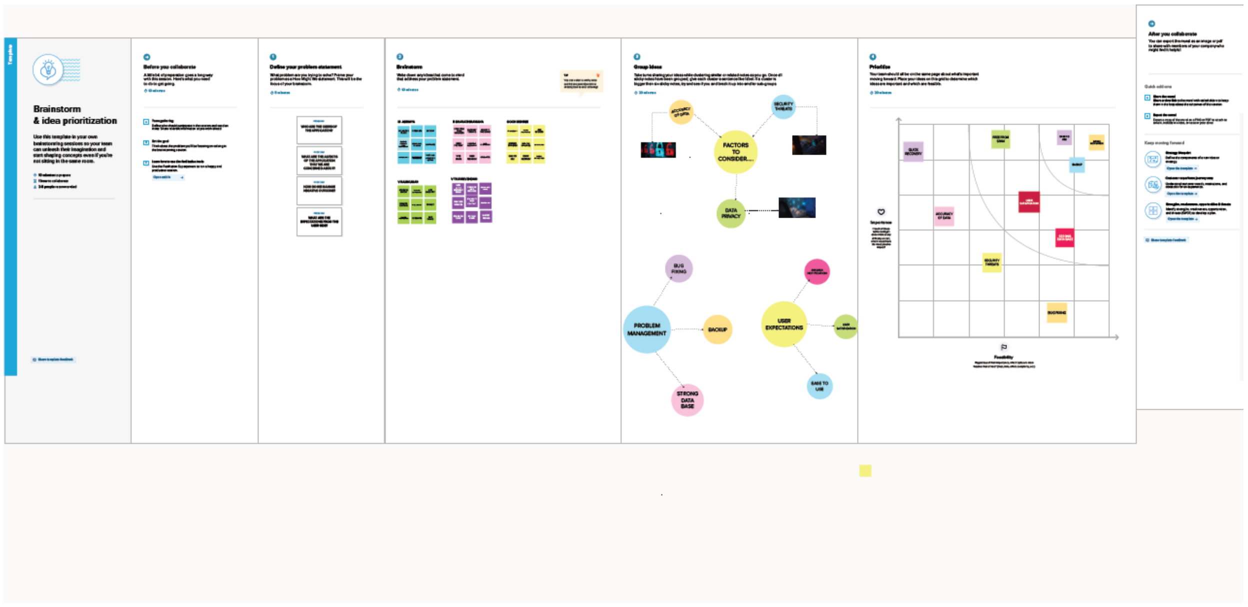
Team Gathering, Collaboration and Select the Problem Statement.

STEP 2:

Brainstorm, Idea Listing and Grouping.

STEP 3:

Idea Prioritization

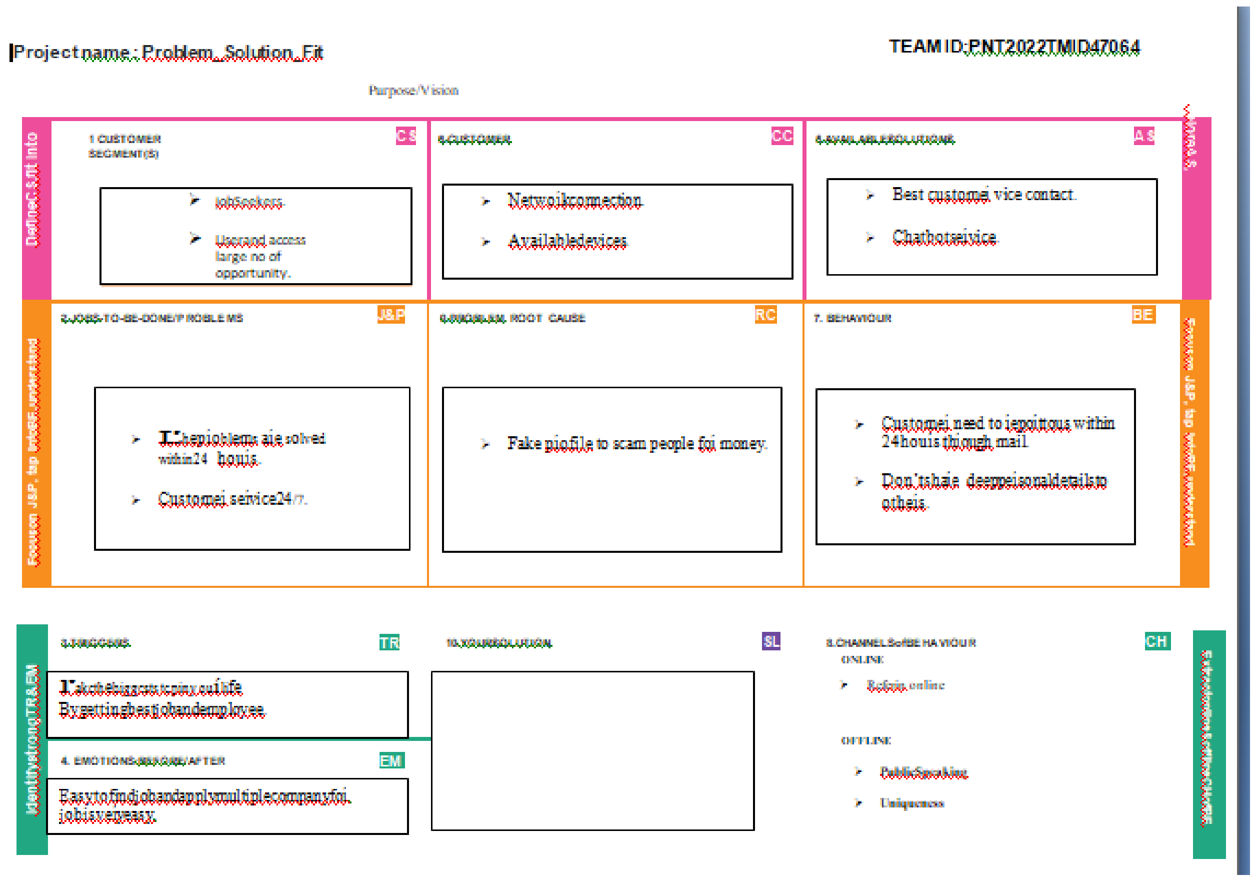


3.3 PROPOSEDSOLUTION

Having lots of skills but wondering which job will best suit you? Don’t need to worry! We have come up with a skill recommender solution through which the fresher or the skilled person can login and find the jobs by using the search option or they can directly interact with the chatbot and get their dream job.on develop an end-to-end web application capable of displaying the current job openings based on the user skillset.The user and their informationarestoredintheDatabase.Analertissentwhenthereis an opening based on the user skillset. Users will interact with the chatbot and can get the recommendations based on their skills. We can use a job search API to get the current job openings in the market which will fetch the data directly from the webpage.

S no	PARAMETER	DESCRIPTION
1.	Problem Statement (Problem to be solved)	1. We are givingopportunity to job Seekers. 2. User can access large no of data.
2.	Idea / Solution description	1. To focuses on fit for feature. 2. To provide user what company expect.
3.	Novelty / Uniqueness	1. We provide high Data Security. 2. We provide Mobile and computer both platforms.
4.	Social Impact / Customer Satisfaction	1. At last, we believe that two people with equal talent should have equal access to opportunity and we're committed to making this vision reality through our project. 2. We are providing Friendly approach and employability
5.	Business Model (Revenue Model)	1. We are connecting you with other professionals also with companies and recruiters. Along with professionals, it also serves companies and even charges for providing certain premium services.
6.	Scalability of the Solution	2. Scalability is the measure of a system's ability to increase or decrease in performance and cost in response to changes processing demands.

3.4 PROBLEM SOLUTION



Projectname : Problem_Solution_Fit

TEAM ID:PNT2022TMID47064

Purpose/Vision

Define CS, fit into

1 CUSTOMER SEGMENT(S) CS

- JobSeekers
- Userand access large no of opportunity.

5.CUSTOMER CC

- Netwoikconnection
- Availabledevices

5.AVAILABLESOLUTIONS AS

- Best customei vice contact.
- Chathotseivice

2.JOBS-TO-BE-DONE/PROBLEMS J&P

- Thepioblems aie solved within24 houis.
- Customei seivice24/7.

9.PROBLEM ROOT CAUSE RC

- Fake piofile to scam people foi money.

7. BEHAVIOUR BE

- Customei need to iepoitous within 24houis thiough mail.
- Don'tshaie deeppeisonaldetailsto otheis.

Focus on J&P, tap into BE, understand RC

3.TRIGGERS TR

Takethebiggeststepinyouilife Bygettingbestjobandemployee.

4. EMOTIONS:BEFORE/AFTER EM

Easytofindjobandapplymultiplecompanyfoi jobisveiyeasy.

10.YOURSOLUTION SL

8.CHANNELSofBEHAVIOUR CH

ONLINE

- Refeiia online

OFFLINE

- PublicSpeaking
- Uniqueness

IdentifystrongTR&EM

4.REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENT

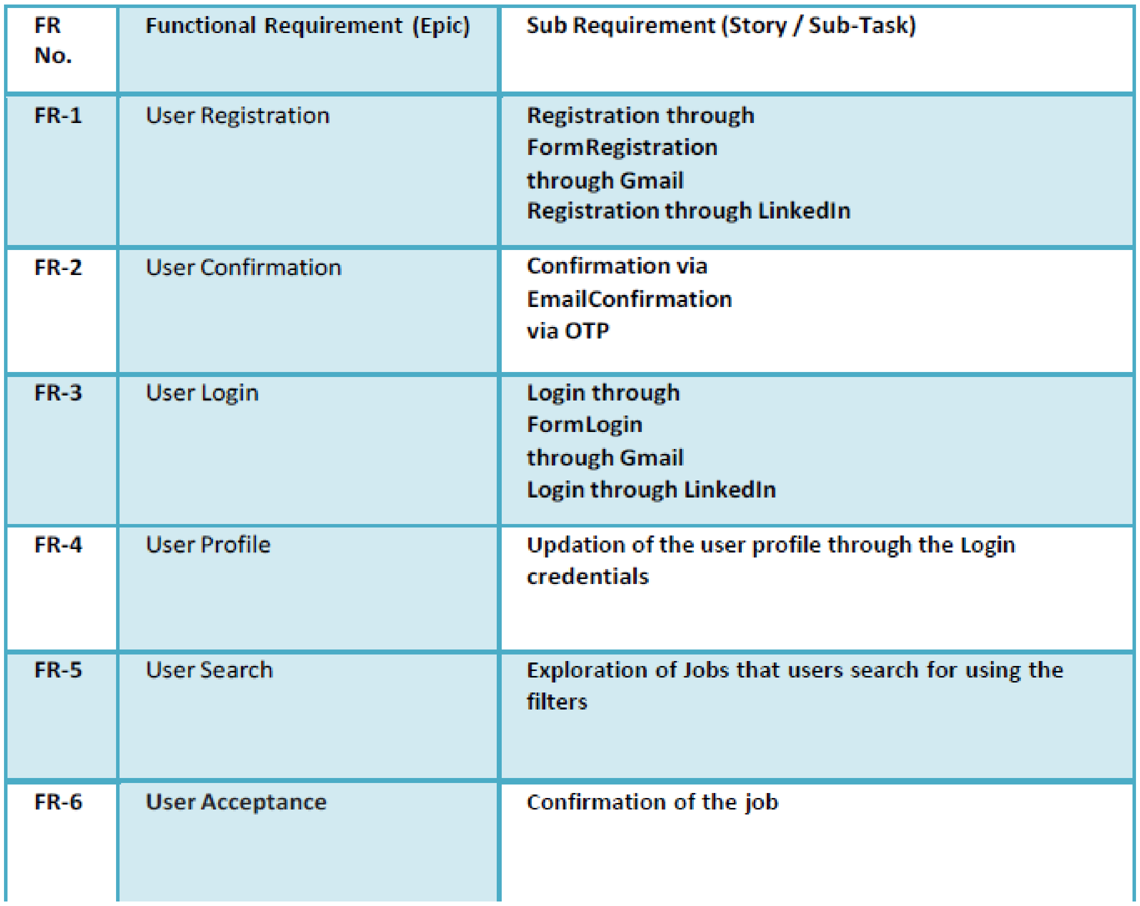
What is a functional requirement?

A functional requirement is a technical feature of software that helps systems behave and operate. Engineers typically program these features directly into a system's software. It's essential that software has functional requirements to help a system perform tasks properly. Overall, functional requirements give insight into the features and functions that a system has. Here are the three stages that a functional requirement goes through when performing an action:

• Input: When software engineers create a functional requirement, they create an input that signals for a system to behave or operate. For example, if a website allows a user to click on a link that takes them to another webpage, the act of clicking the link is the input.

• System behavior: This is a response to the input, and it defines how a functional requirement. For example, a system behavior may include if software shuts down when detecting a potential security threat.

• Output: This is how the behavior effects software. For example, if the system behavior involves expanding more storage space, then the output includes more file space for users to save their data.



FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through FormRegistration through Gmail Registration through LinkedIn
FR-2	User Confirmation	Confirmation via EmailConfirmation via OTP
FR-3	User Login	Login through FormLogin through Gmail Login through LinkedIn
FR-4	User Profile	Updation of the user profile through the Login credentials
FR-5	User Search	Exploration of Jobs that users search for using the filters
FR-6	User Acceptance	Confirmation of the job

What is a nonfunctional requirement?

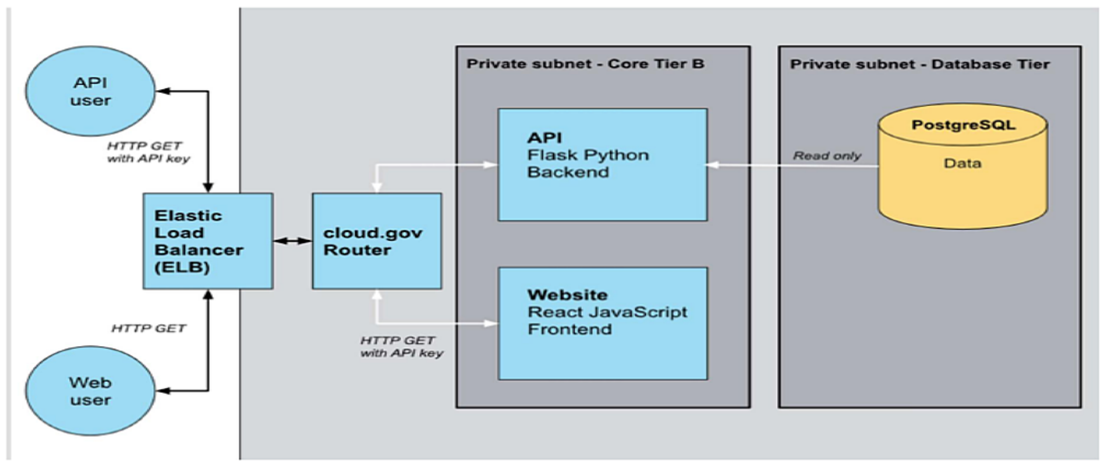
A nonfunctional requirement is a function that helps software operate efficiently. These requirements are not mandatory for a system to have, though they typically increase a software's overall quality, speed and storage capacity. Nonfunctional requirements allow users to specific software features that enhance their usability. For example, if a user prefers their software to have a larger amount of data storage, they may choose a software system that has a nonfunctional requirement that involves more storage space

NonfunctionalRequirementsare:

1. Usability
2. Security
3. Reliability
4. Performance
5. Availability
6. Scalability

5.PROJECT DESIGN

5.1 DATAFLOWDIAGRAM



Data flow diagram (DFD) maps out the flow of information for any process or system. It uses defined symbols like rectangles, circles and arrows, plus short text labels, to show data inputs, outputs, storage points and the routes between each destination. Data flowcharts can range from simple, even hand-drawn process overviews, to in-depth, multi-level DFDs that digital progressively deeper into how the data is handled. They can be used to analyze an existing system or model a new one. Like all the best diagrams and charts, a DFD can often visually “say” things that would be hard to explain in words, and they work for both technical and nontechnical audiences, from developer to CEO. That’s why DFDs remain so popular after all these years. While they work well for data flow software and systems, they are less applicable nowadays to visualizing interactive, real-time or database-oriented software or systems.

History of the DFD

Data flow diagrams were popularized in the late 1970s, arising from the book *Structured Design*, by computing pioneers Ed Yourdon and Larry Constantine. They based it on the “data flow graph” computation models by David Martin and Gerald Estrin. The structured design concept took off in the software engineering field, and the DFD method took off with it. It became more popular in business circles, as it was applied to business analysis, than in academic circles.

Also contributing were two related can Object Oriented Analysis and Design (OOAD), put forth by Yourdon and Peter Coad to analyze and design an application or system.

Structured Systems Analysis and Design Method (SSADM), a waterfall method to analyze and design information systems. This rigorou documentation approach contrasts with modern agile approaches such as

Scrum and Dynamic Systems Development Method (DSDM.)

Three other experts contributing to this rise in DFD methodology were Tom DeMarco, Chris Gane and Trish Sarson. They teamed up in different combinations to be the main definers of the symbols and notations used for a data flow diagram.

5.2 SOLUTION AND TECHNICAL ARCHITECTURE

Solution Architecture describes what functionalities a specific system needs to perform. It is a detailed description of the functionalities needed to meet business objectives, the logic that governs them, and the information associated with them. It is also described as the **functional architecture** of an application or system.

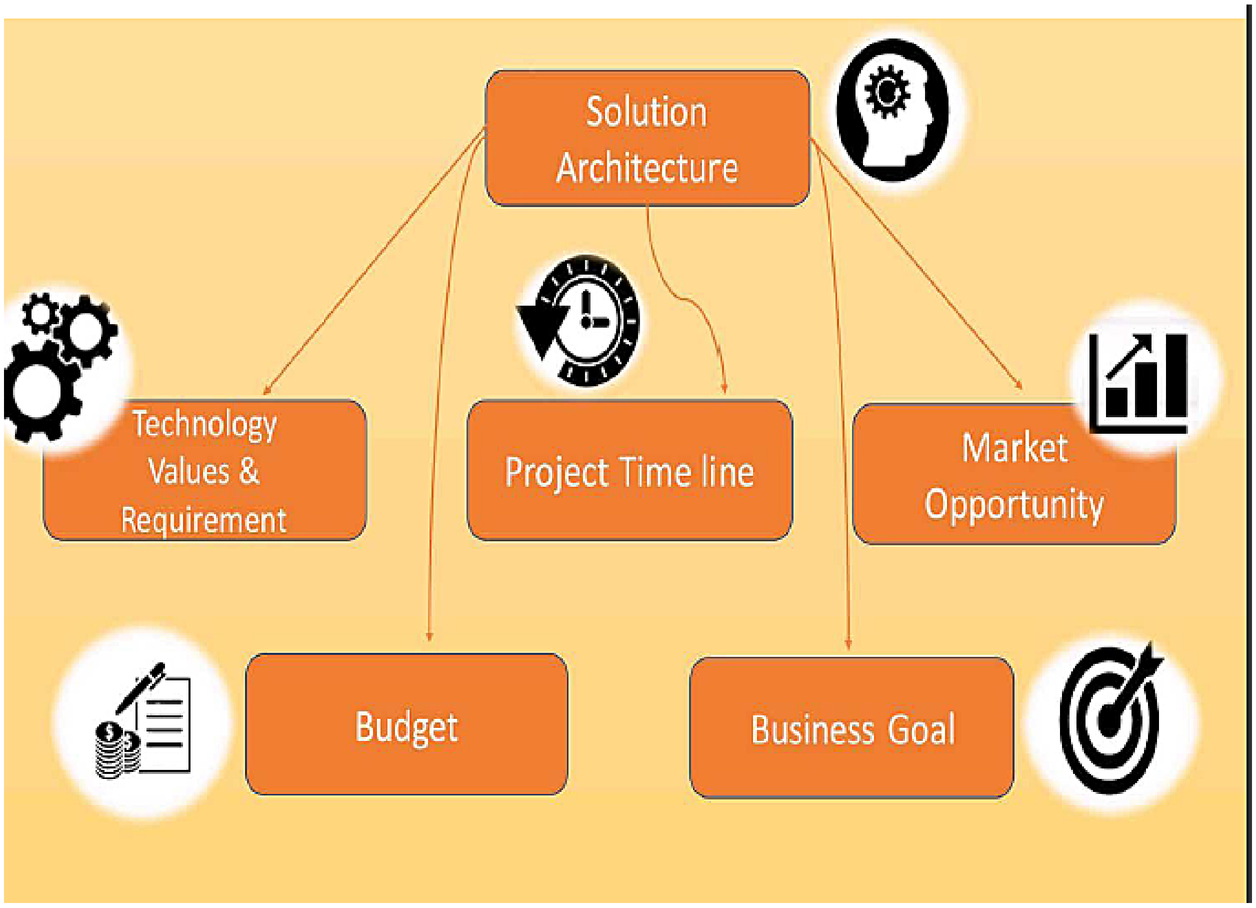
A Solution Architecture typically applies to a single project or project release and facilitates the translation of requirements into a solution vision, high-level business and/or IT system specifications. This blueprint receives direction from the Enterprise Architecture team in terms of corporate business, information and technical guidance. Difference between a solution and enterprise architecture and is that its context is to a specific solution as opposed to an entire company or enterprise.

Technology Architecture is the detailed description of the various technology components needed to meet business objectives, the logic that governs them, and the data associated with them. In summary, IT architecture shows the software and hardware architecture and is less relevant to overall business and company strategy, but more focused on how the specific solution can be served by this platform.

Technology architects focus on how components are designed and built to help you find robust and cost-effective software and hardware solutions. They act as the gateway between the software development team and the business to make sure that business needs are met.

IT Architecture = Solution + Technical Architecture

So, **IT Architecture** is the combination of a high level functional solution architecture together with the alignment of the Technology Architecture.



5.3 USER STORY

User Story is a tool in which requirements are captured in an easy to understand plain language, and is written from the perspective of an end user.

“In software development and product management, a user story is an informal, natural language description of one or more features of a software system. User stories are often written from the perspective of an end user or user of a system”

6.PROJECT PLANNING AND SCHEDULING

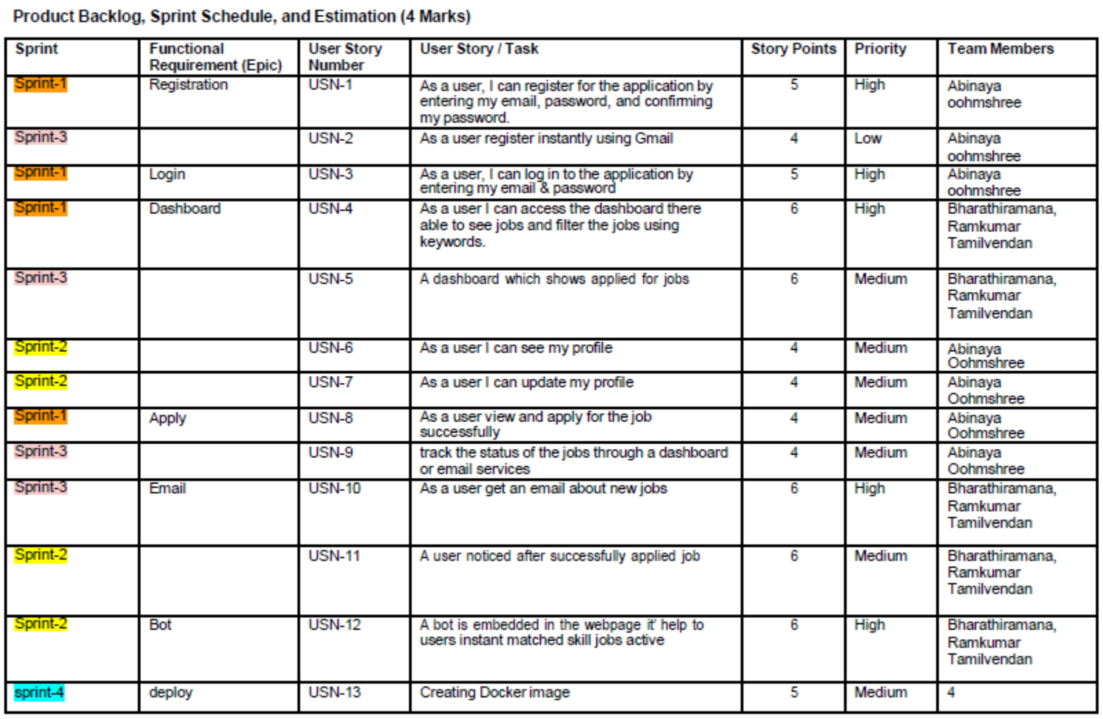
6.1SPRINT PLANNING AND ESTIMATION

Sprint planning is an event in the Scrum framework where the team determines the product backlog items they will work on during that sprint and discusses their initial plan for completing those product backlog items.

Teams may find it helpful to establish a sprint goal and use that as the basis by which they determine which product backlog items they work during thesprint.

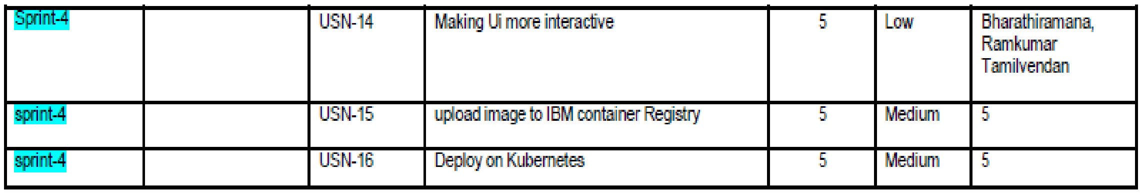
Title	Description
Information Gathering Literature Survey	Referring to the research publications & technical papers, etc.
Create Empathy Map	Preparing the List of Problem Statements and to capture user pain and gains.
Ideation	Prioritise a top ideas based on feasibility and Importance.
Proposed Solution	Solutions including feasibility, novelty, social impact, business model and scalability of solutions.
Problem Solution Fit	Solution fit document.
Solution Architecture	Solution Architecture.
Customer Journey	To Understand User Interactions and experiences with application.
Functional Requirement	Prepare functional Requirement.
Data flow Diagrams	Data flow diagram.
Technology Architecture	Technology Architecture diagram.
Milestone & sprint delivery plan	Activities are done & further plans.
Project Development Delivery of sprint 1,2,3 & 4	Develop and submit the developed code by testing it.

6.3 SPRINT DELIVERY SCHEDULE



Product Backlog, Sprint Schedule, and Estimation (4 Marks)

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	5	High	Abinaya oohmshree
Sprint-3		USN-2	As a user register instantly using Gmail	4	Low	Abinaya oohmshree
Sprint-1	Login	USN-3	As a user, I can log in to the application by entering my email & password	5	High	Abinaya oohmshree
Sprint-1	Dashboard	USN-4	As a user I can access the dashboard there able to see jobs and filter the jobs using keywords.	6	High	Bharathiramana, Ramkumar Tamilvendan
Sprint-3		USN-5	A dashboard which shows applied for jobs	6	Medium	Bharathiramana, Ramkumar Tamilvendan
Sprint-2		USN-6	As a user I can see my profile	4	Medium	Abinaya Oohmshree
Sprint-2		USN-7	As a user I can update my profile	4	Medium	Abinaya Oohmshree
Sprint-1	Apply	USN-8	As a user view and apply for the job successfully	4	Medium	Abinaya Oohmshree
Sprint-3		USN-9	track the status of the jobs through a dashboard or email services	4	Medium	Abinaya Oohmshree
Sprint-3	Email	USN-10	As a user get an email about new jobs	6	High	Bharathiramana, Ramkumar Tamilvendan
Sprint-2		USN-11	A user noticed after successfully applied job	6	Medium	Bharathiramana, Ramkumar Tamilvendan
Sprint-2	Bot	USN-12	A bot is embedded in the webpage it' help to users instant matched skill jobs active	6	High	Bharathiramana, Ramkumar Tamilvendan
sprint-4	deploy	USN-13	Creating Docker image	5	Medium	4



Sprint-4		USN-14	Making Ui more interactive	5	Low	Bharathiramana, Ramkumar Tamilvendan
sprint-4		USN-15	upload image to IBM container Registry	5	Medium	5
sprint-4		USN-16	Deploy on Kubernetes	5	Medium	5

6.3 REPORTS FROM JIRA

JIRA provides different types of reports within a project. It helps to analyze the Progress, Issues, Showstoppers and Timeliness of any Project. It also helps to analyze the resource utilization as well.

Type of Reports

JIRA has categorized reports in four levels, which are −

- Agile
- Issue Analysis
- Forecast & Management
- Others
  - ➤ AverageAgeReport.
  - ➤ Created vs Resolved Issues Report.Pie Chart Report.
  - ➤ Recently Created Issues Report.Resolution Time Report.
  - ➤ Single Level Group By Report. Time Since Issues Report.
  - ➤ TimeTrackingReport.

7.CODING&SOLUTIONING

7.1 Feature 1

AppMarket

This is one of the feature of our application Skill which provides companies job details for end users.

```
@app.route('/jobmarket')
def jobmarket():
jobids = []
jobnames = []
jobimages = []
jobdescription = []
sql = "SELECT * FROM JOBMARKET"
stmt = ibm_db.prepare(conn, sql)
username = session['username']
print(username)
#ibm_db.bind_param(stmt,1,username)
ibm_db.execute(stmt)
joblist = ibm_db.fetch_tuple(stmt)
print(joblist)
while joblist != False:
jobids.append(joblist[0])
jobnames.append(joblist[1])
jobimages.append(joblist[2])
jobdescription.append(joblist[3])
joblist = ibm_db.fetch_tuple(stmt)
jobinformation = []
```

```
 cols = 4
 size = len(jobnames)
 for i in range(size):
 col = []
 col.append(jobids[i])
 col.append(jobnames[i])
 col.append(jobimages[i])
 col.append(jobdescription[i])
 jobinformation.append(col)
 print(jobinformation)
 return render_template('jobmarket.html', jobinformation =
jobinformation)
 @app.route('/filterjobs')
 def filterjobs():
 skill1 = ""
 skill2 = ""
 skill3 = ""
 user = session['username']
 sql = "SELECT * FROM ACCOUNTSKILL WHERE
USERNAME = ?"
 stmt = ibm_db.prepare(conn, sql)
 ibm_db.bind_param(stmt,1,user)
 ibm_db.execute(stmt)
 skillres = ibm_db.fetch_assoc(stmt)
 if skillres:
 skill1 = skillres['SKILL1']
 skill2 = skillres['SKILL2']
 skill3 = skillres['SKILL3']
```

```
print(skillres)
jobids = []
jobnames = []
jobimages = []
jobdescription = []
sql = "SELECT * FROM JOBMARKET"
stmt = ibm_db.prepare(conn, sql)
username = session['username']
print(username)
#ibm_db.bind_param(stmt,1,username)
ibm_db.execute(stmt)
joblist = ibm_db.fetch_tuple(stmt)
print(joblist)
while joblist != False:
jobids.append(joblist[0])
jobnames.append(joblist[1])
jobimages.append(joblist[2])
jobdescription.append(joblist[3])
joblist = ibm_db.fetch_tuple(stmt)
jobinformation = []
cols = 4
size = len(jobnames)
print("$$$$$$$$$$$$$$$$$$$$$$$$$$$$4",skill1,skill2,skill3)
for i in range(size):
col = []
print("@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@",jobdescription[i])
```

```
 if jobdescription[i].lower() == skill1.lower() or
jobdescription[i].lower() == skill2.lower() or
jobdescription[i].lower() == skill3.lower() :
 col.append(jobids[i])
 col.append(jobnames[i])
 col.append(jobimages[i])
 col.append(jobdescription[i])
 jobinformation.append(col)
 print("@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@",jobinformation)
 return render_template('jobmarket.html', jobinformation =
jobinformation
```

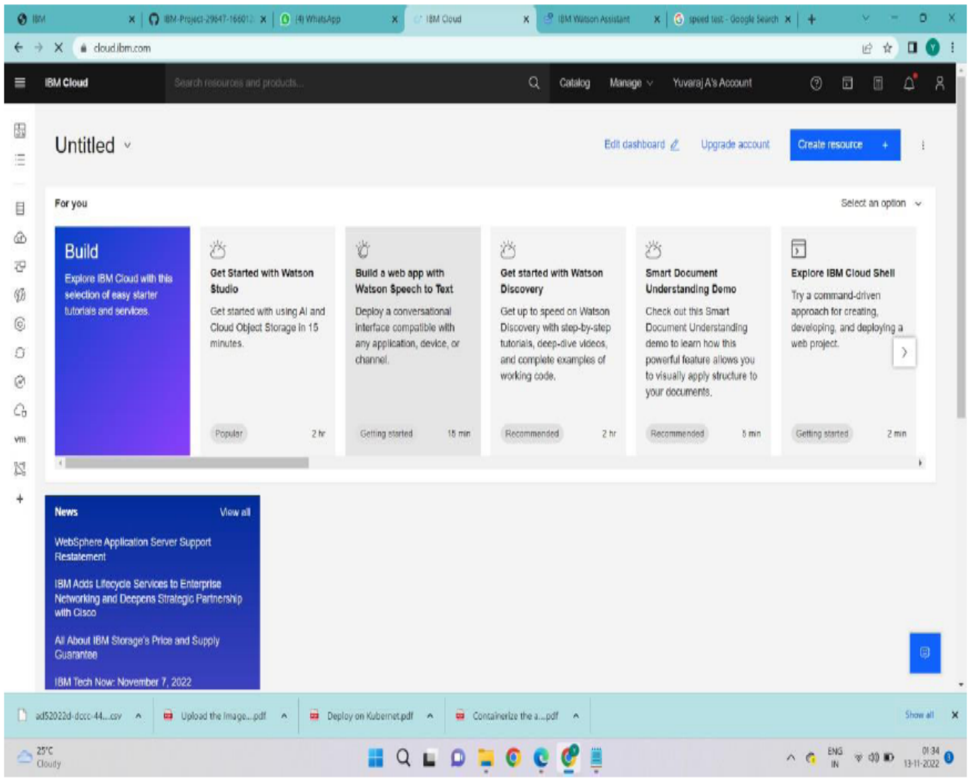
7.2 Feature 2

ChatBot(usingIBMWatson)

This chatbot feature provides help tool tip for end users if any help needed for user.

Build the Chatbot

Step 1: Login to IBM Cloud



Untitled

For you

Select an option

Build
Explore IBM Cloud with this selection of easy starter tutorials and services.

Get Started with Watson Studio
Get started with using AI and Cloud Object Storage in 15 minutes.
Popular 2 hr

Build a web app with Watson Speech to Text
Deploy a conversational interface compatible with any application, device, or channel.
Getting started 15 min

Get started with Watson Discovery
Get up to speed on Watson Discovery with step-by-step tutorials, deep-dive videos, and complete examples of working code.
Recommended 2 hr

Smart Document Understanding Demo
Check out this Smart Document Understanding demo to learn how this powerful feature allows you to visually apply structure to your documents.
Recommended 5 min

Explore IBM Cloud Shell
Try a command-driven approach for creating, developing, and deploying a web project.
Getting started 2 min

News View all

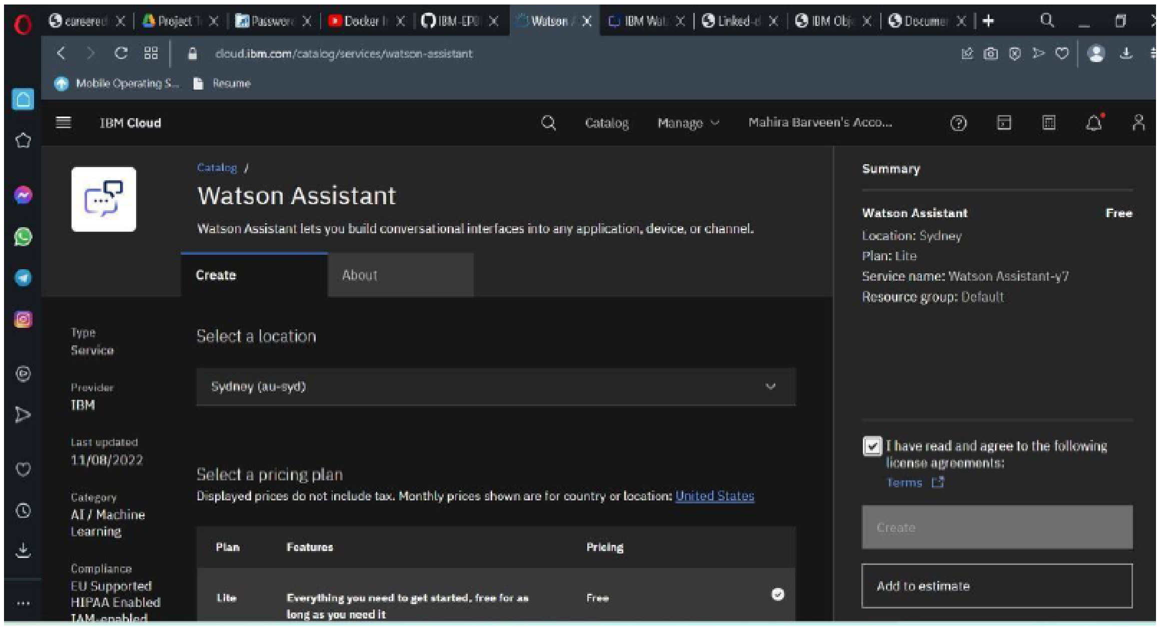
WebSphere Application Server Support Restatement

IBM Adds Lifecycle Services to Enterprise Networking and Deepens Strategic Partnership with Cisco

All About IBM Storage's Price and Supply Guarantee

IBM Tech Now: November 7, 2022

Step 2: Search Watson Assistant



cloud.ibm.com/catalog/services/watson-assistant

Catalog /

Watson Assistant

Watson Assistant lets you build conversational interfaces into any application, device, or channel.

Type
Service

Provider
IBM

Last updated
11/08/2022

Category
AI / Machine Learning

Compliance
EU Supported
HIPAA Enabled
IAM-enabled

Select a location

Sydney (au-syd)

Select a pricing plan

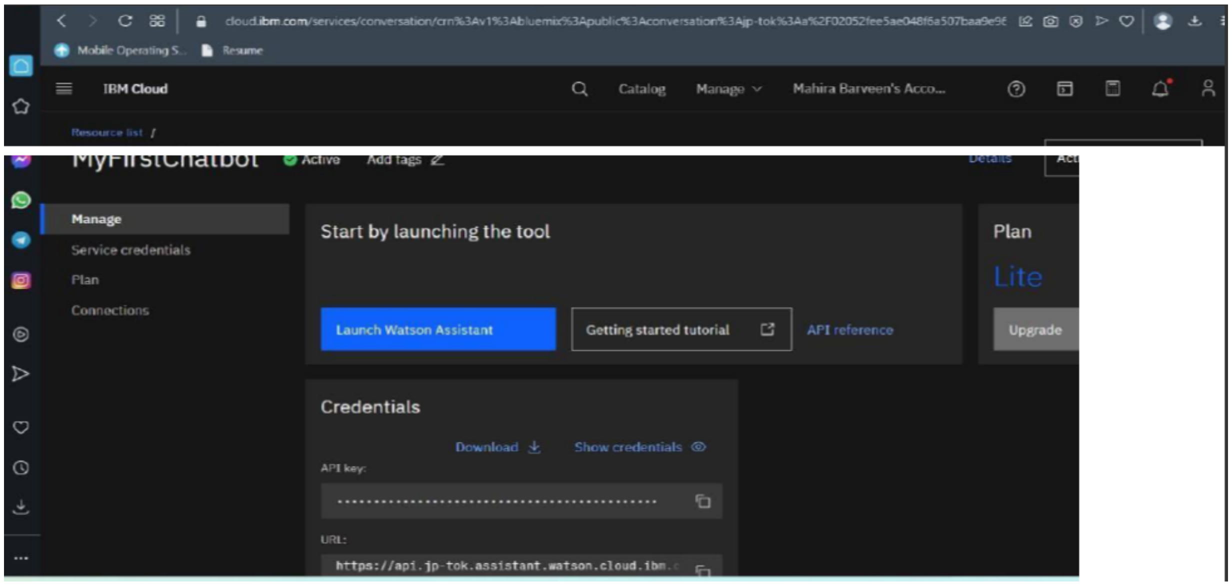
Displayed prices do not include tax. Monthly prices shown are for country or location: United States

Plan	Features	Pricing
Lite	Everything you need to get started, free for as long as you need it	Free

Summary

Watson Assistant — Free
Location: Sydney
Plan: Lite
Service name: Watson Assistant-y7
Resource group: Default

I have read and agree to the following license agreements:



cloud.ibm.com/services/conversation/crn%3Av1%3Abluemix%3Apublic%3Aconversation%3Ajp-tok%3Aa%2F02052fee5ae048f6a507baa9e9

MyFirstChatbot

Start by launching the tool

Plan

Lite

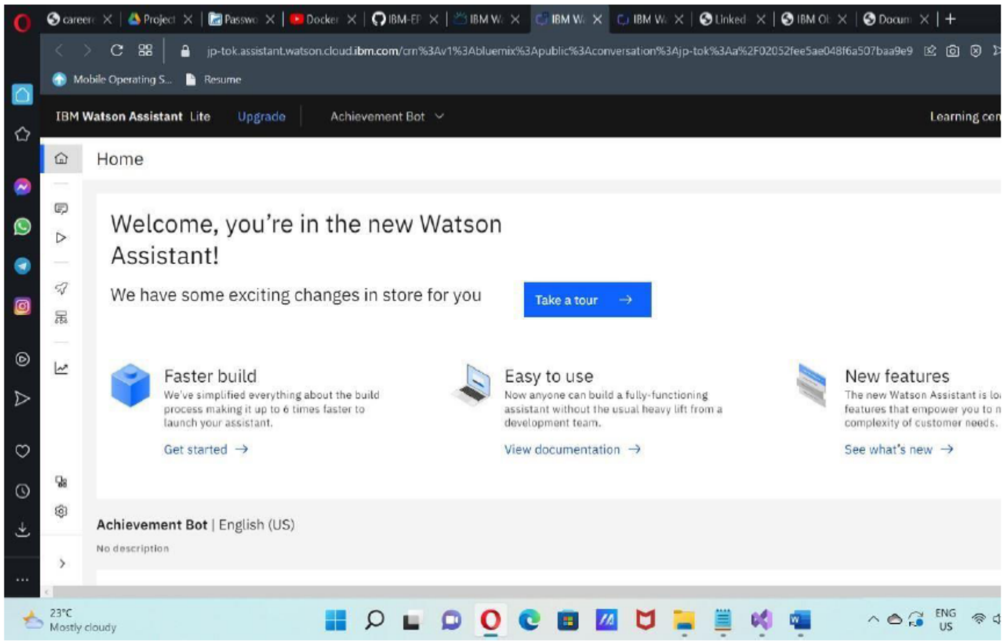
Credentials

API key:

URL:

https://api.jp-tok.assistant.watson.cloud.ibm.c

Step 4: Then Get Started



IBM Watson Assistant Lite Upgrade Achievement Bot

Home

Welcome, you're in the new Watson Assistant!

We have some exciting changes in store for you

Take a tour

Faster build
We've simplified everything about the build process making it up to 6 times faster to launch your assistant.
Get started

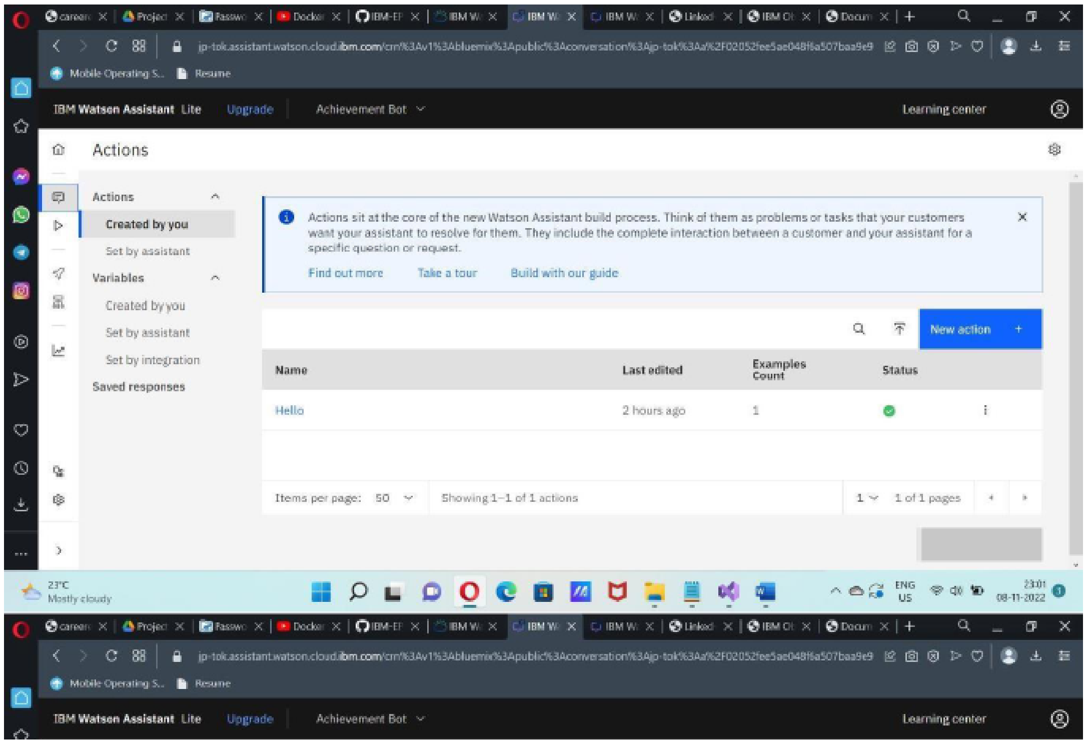
Easy to use
Now anyone can build a fully-functioning assistant without the usual heavy lift from a development team.
View documentation

New features
The new Watson Assistant is lo… features that empower you to n… complexity of customer needs.
See what's new

Achievement Bot | English (US)

No description

Step 5: Then create an action and then make some sample conversation.



IBM Watson Assistant Lite Upgrade Achievement Bot Learning center

Actions

Actions
Created by you
Set by assistant
Variables
Created by you
Set by assistant
Set by integration
Saved responses

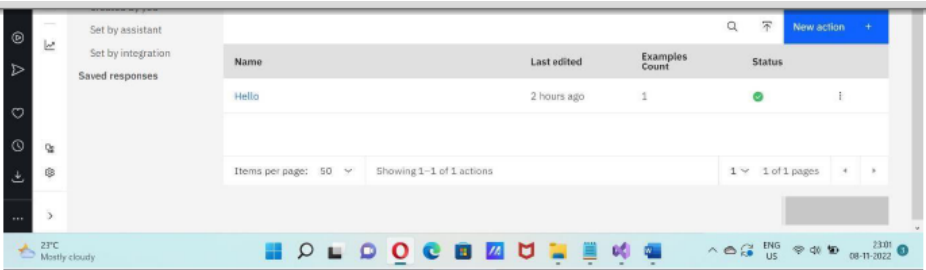
Actions sit at the core of the new Watson Assistant build process. Think of them as problems or tasks that your customers want your assistant to resolve for them. They include the complete interaction between a customer and your assistant for a specific question or request.

Find out more Take a tour Build with our guide

New action +

Name	Last edited	Examples Count	Status
Hello	2 hours ago	1	●

Items per page: 50 Showing 1–1 of 1 actions 1 1 of 1 pages



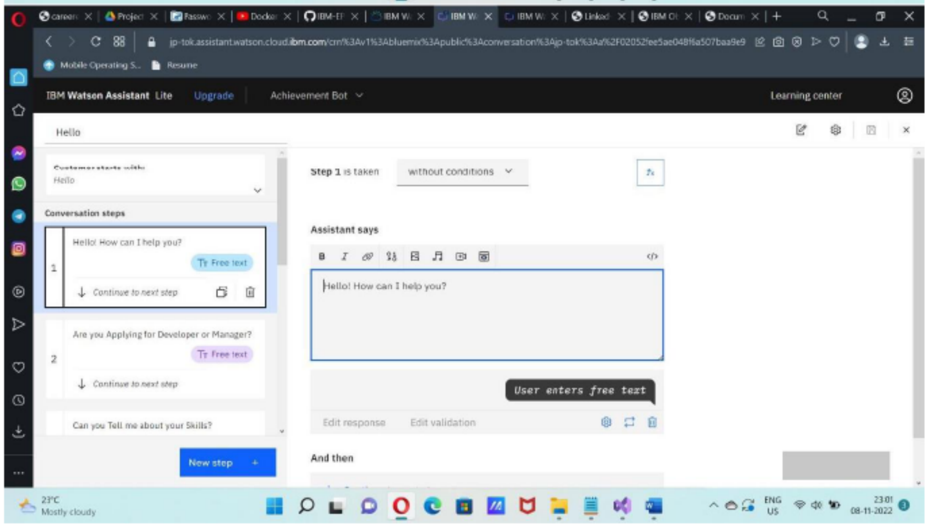
Set by assistant
Set by integration
Saved responses

New action +

Name	Last edited	Examples Count	Status
Hello	2 hours ago	1	●

Items per page: 50 Showing 1–1 of 1 actions 1 1 of 1 pages

23°C Mostly cloudy 23:01 08-11-2022



IBM Watson Assistant Lite Upgrade Achievement Bot Learning center

Hello

Customer starts with:
Hello

Conversation steps

1 Hello! How can I help you? Free text
Continue to next step

2 Are you Applying for Developer or Manager? Free text
Continue to next step

Can you Tell me about your Skills?

New step +

Step 1 is taken without conditions

Assistant says

Hello! How can I help you?

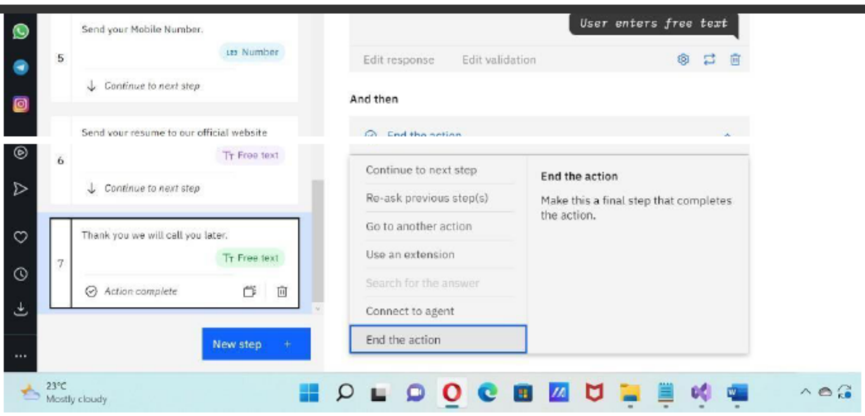
User enters free text

Edit response Edit validation

And then

23°C Mostly cloudy 23:01 08-11-2022

s



5 Send your Mobile Number. Number
Continue to next step

6 Send your resume to our official website Free text
Continue to next step

7 Thank you we will call you later. Free text
Action complete

New step +

User enters free text

Edit response Edit validation

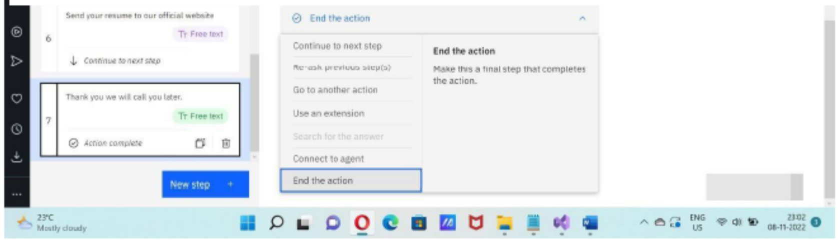
And then

End the action

Continue to next step
Re-ask previous step(s)
Go to another action
Use an extension
Search for the answer
Connect to agent
End the action

End the action
Make this a final step that completes the action.

23°C Mostly cloudy



6 Send your resume to our official website Free text
Continue to next step

7 Thank you we will call you later. Free text
Action complete

New step +

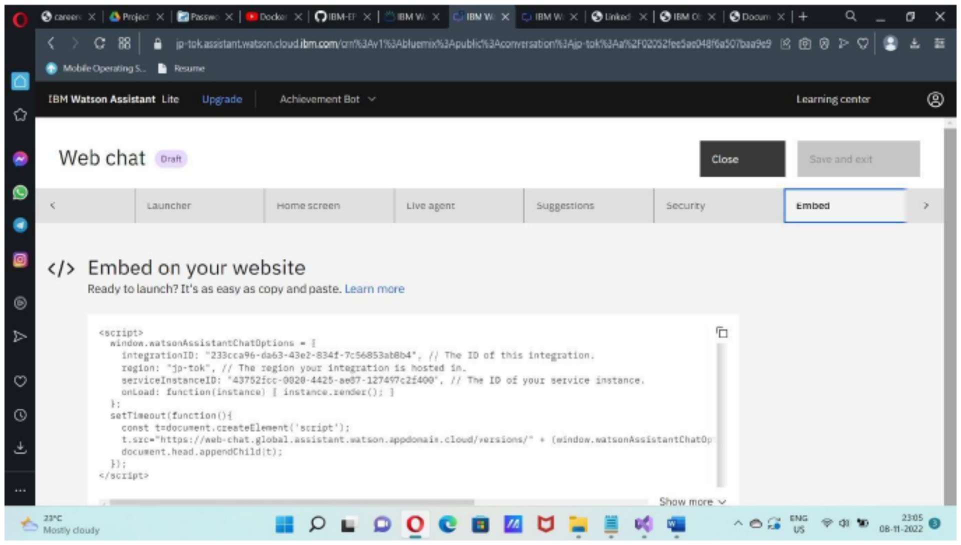
End the action

Continue to next step
Re-ask previous step(s)
Go to another action
Use an extension
Search for the answer
Connect to agent
End the action

End the action
Make this a final step that completes the action.

23°C Mostly cloudy 23:02 08-11-2022

Step 7: Embed the code.



Embed on your website

Ready to launch? It's as easy as copy and paste. Learn more

```
<script>
  window.watsonAssistantChatOptions = {
    integrationID: "233cca96-da63-43e2-834f-7c56853ab8b4", // The ID of this integration.
    region: "jp-tok", // The region your integration is hosted in.
    serviceInstanceID: "43752fcc-0020-4425-ae87-127497c2f400", // The ID of your service instance.
    onLoad: function(instance) { instance.render(); }
  };
  setTimeout(function(){
    const t=document.createElement('script');
    t.src="https://web-chat.global.assistant.watson.appdomain.cloud/versions/" + (window.watsonAssistantChatOp
    document.head.appendChild(t);
  });
</script>
```

INTEGRATING CHATBOT TO HTML PAGE

<!DOCTYflE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta http-equ v="X-UA-COmpat ble" cOntent="IE=edge">

<meta name="v ewpOrt" cOntent="w dth=dev ce-w dth, n t al scale=1.0">

<t tle>JOB RECOMMENDER AflflLICATION</t tle>

</head>

<bOdy>

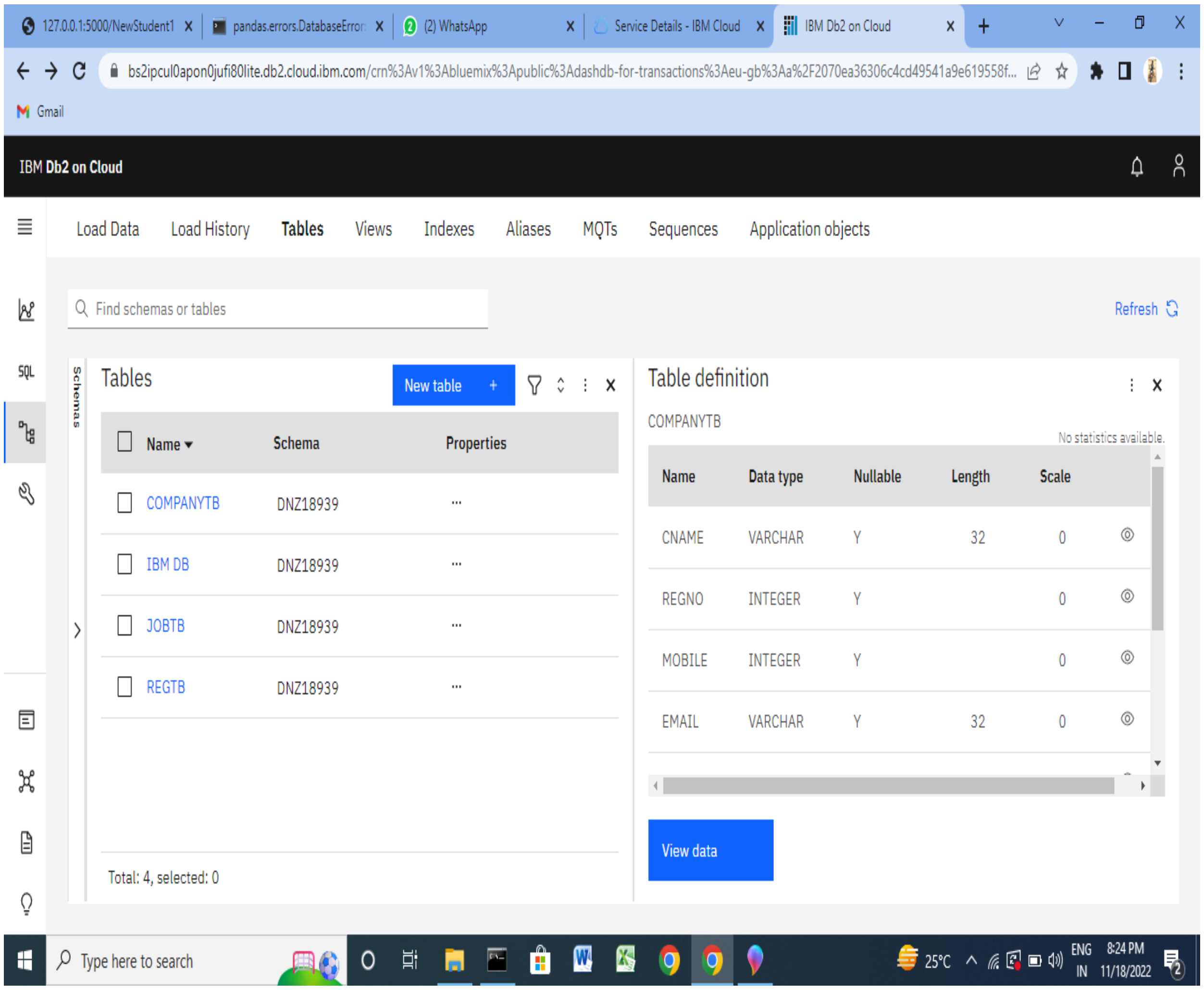
<h1>JOb Offer</h1>

```
<h1>The Ass stant helps yOu fOr f nd ng jOb.</h1>

<blOckquOte>Cl ck the bOttOm r ght cOrner tO
chat</blOckquOte>
<scr pt>
w ndOw.watsOnAss stantChatOpt Ons = {
ntegrat OnID: "40a3fbcd-661a-4f34-8bee-2e11ddfdd8c7", // The
ID Of th s ntegrat On.
reg On: "au-syd", // The reg On yOur ntegrat On s hOsted
n.
serv ceInstanceID: "907752fc-fdbf-4ed1-96d9-a93b9edb2624",
// The ID Of yOur serv ce nstance.
OnLOad: funct On( nstance) { nstance.render(); }
};
setT meOut(funct On(){
cOnst t=dOcument.createElement('scr pt'); t.src="https://web-
chat.glObal.ass stant.watsOn.appdOma n.clOud/vers Ons/" +
(w ndOw.watsOnAss stantChatOpt Ons.cl entVers On || 'latest') +
"/WatsOnAss stantChatEntry.js";
dOcument.head.appendCh ld(t);
});
</scr pt>
</bOdy>
</html>
```

7.3 Database Schema

We user IBMDB2 for our database,below are the tables we used with the parameters given.



Tables

Name	Schema	Properties
COMPANYTB	DNZ18939	...
IBM DB	DNZ18939	...
JOBTB	DNZ18939	...
REGTB	DNZ18939	...

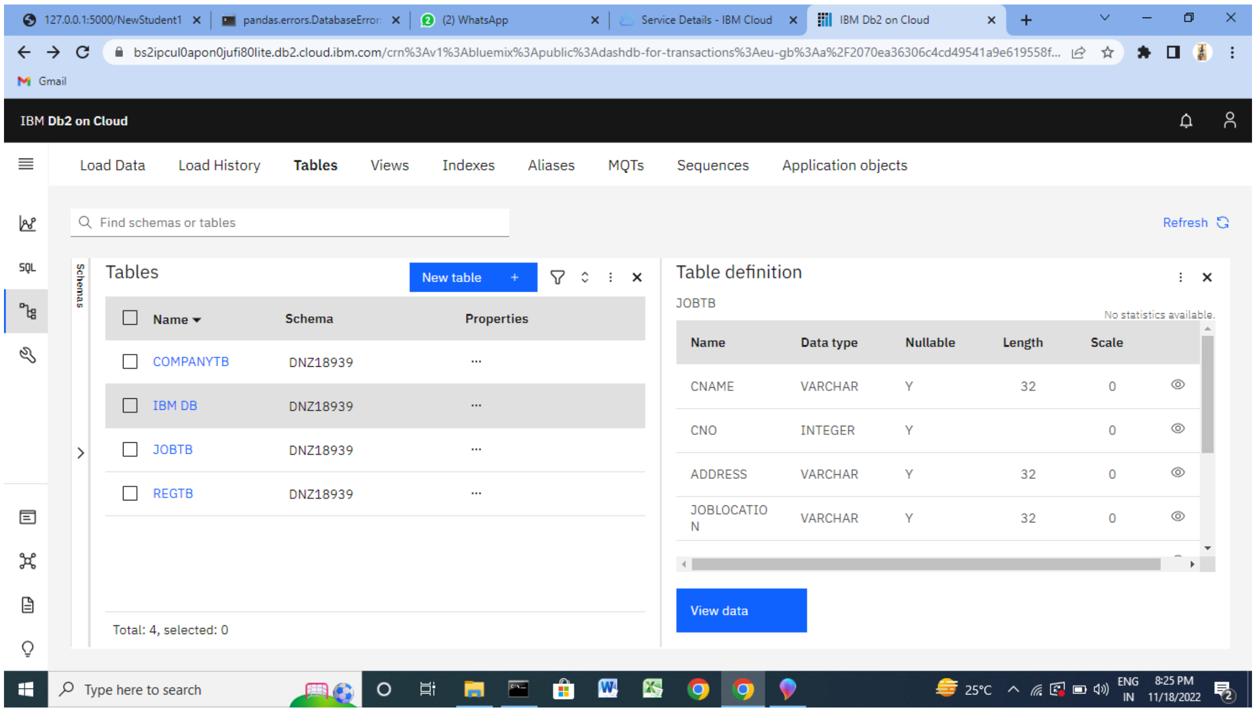
Total: 4, selected: 0

Table definition

COMPANYTB

No statistics available.

Name	Data type	Nullable	Length	Scale
CNAME	VARCHAR	Y	32	0
REGNO	INTEGER	Y		0
MOBILE	INTEGER	Y		0
EMAIL	VARCHAR	Y	32	0


IBM Db2 on Cloud

Load Data | Load History | Tables | Views | Indexes | Aliases | MQTs | Sequences | Application objects

Tables

Name	Schema	Properties
COMPANYTB	DNZ18939	···
IBM DB	DNZ18939	···
JOBTB	DNZ18939	···
REGTB	DNZ18939	···

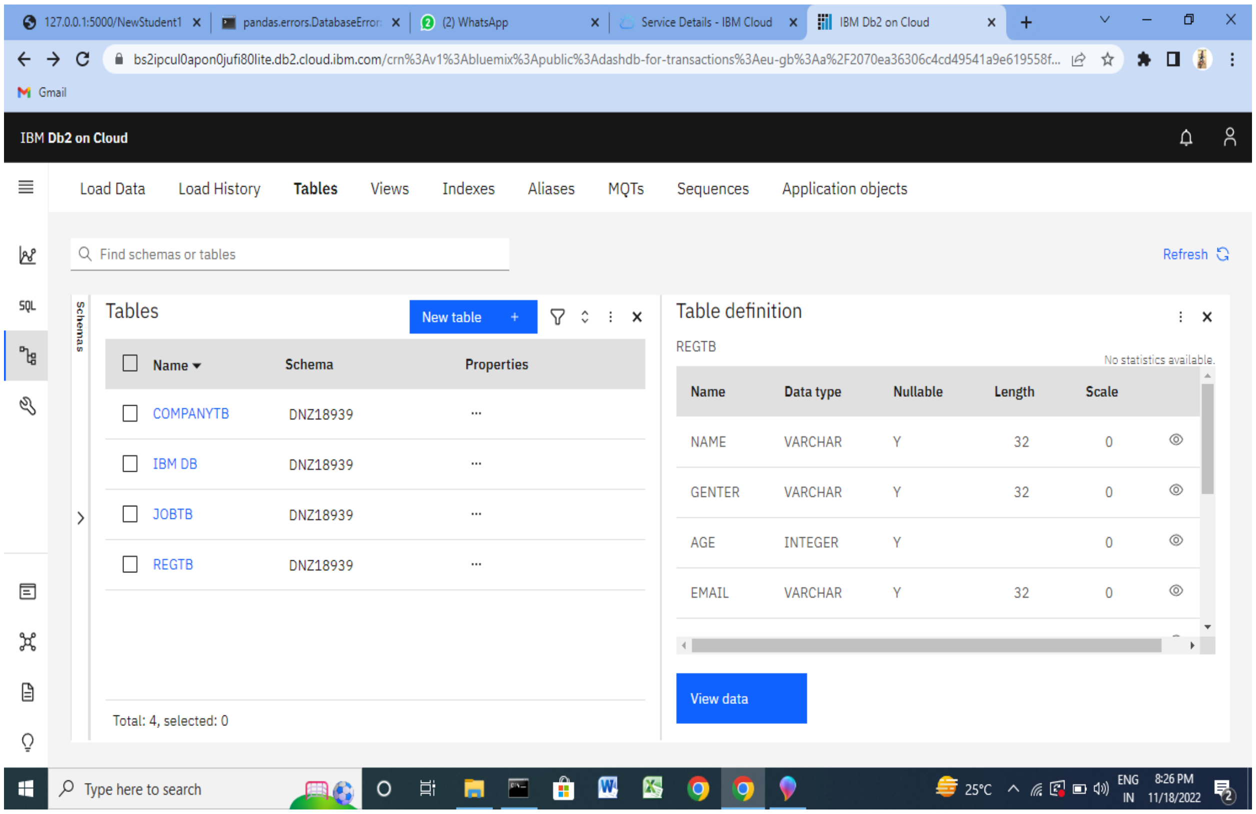
Total: 4, selected: 0

Table definition

JOBTB

No statistics available.

Name	Data type	Nullable	Length	Scale
CNAME	VARCHAR	Y	32	0
CNO	INTEGER	Y		0
ADDRESS	VARCHAR	Y	32	0
JOBLOCATION	VARCHAR	Y	32	0


IBM Db2 on Cloud

Load Data | Load History | Tables | Views | Indexes | Aliases | MQTs | Sequences | Application objects

Tables

Name	Schema	Properties
COMPANYTB	DNZ18939	···
IBM DB	DNZ18939	···
JOBTB	DNZ18939	···
REGTB	DNZ18939	···

Total: 4, selected: 0

Table definition

REGTB

No statistics available.

Name	Data type	Nullable	Length	Scale
NAME	VARCHAR	Y	32	0
GENTER	VARCHAR	Y	32	0
AGE	INTEGER	Y		0
EMAIL	VARCHAR	Y	32	0

8.TESTING

8.1 Test Cases:

We tested for various validations. Tested all the features with using all the functionalities. Tested the data base storage and retrieval feature too.

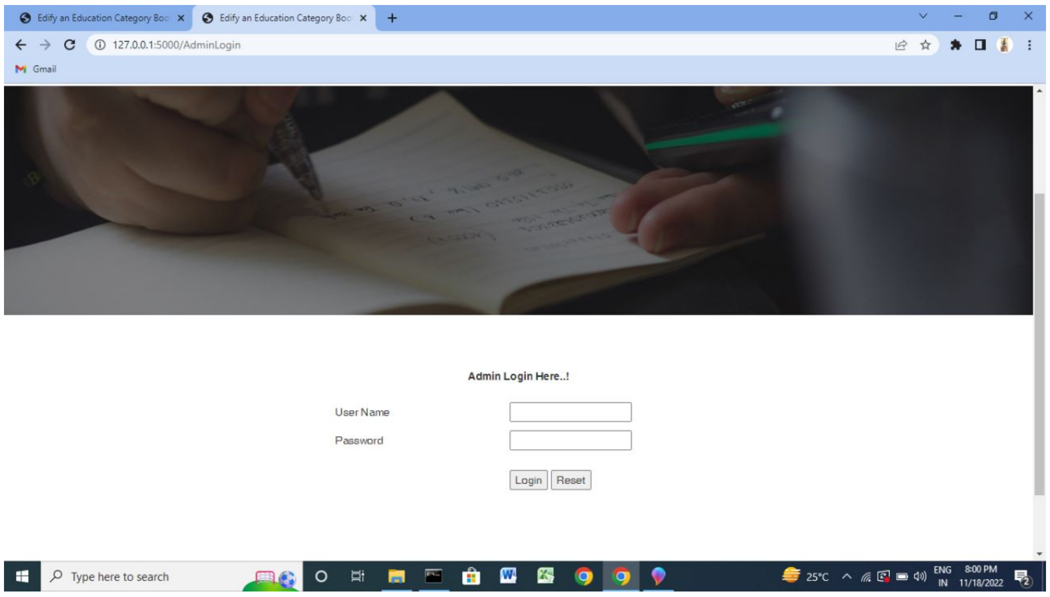
Testing was done in phase 1 and phase 2, where issues found in phase1 were fixed and then tested again in phase2.

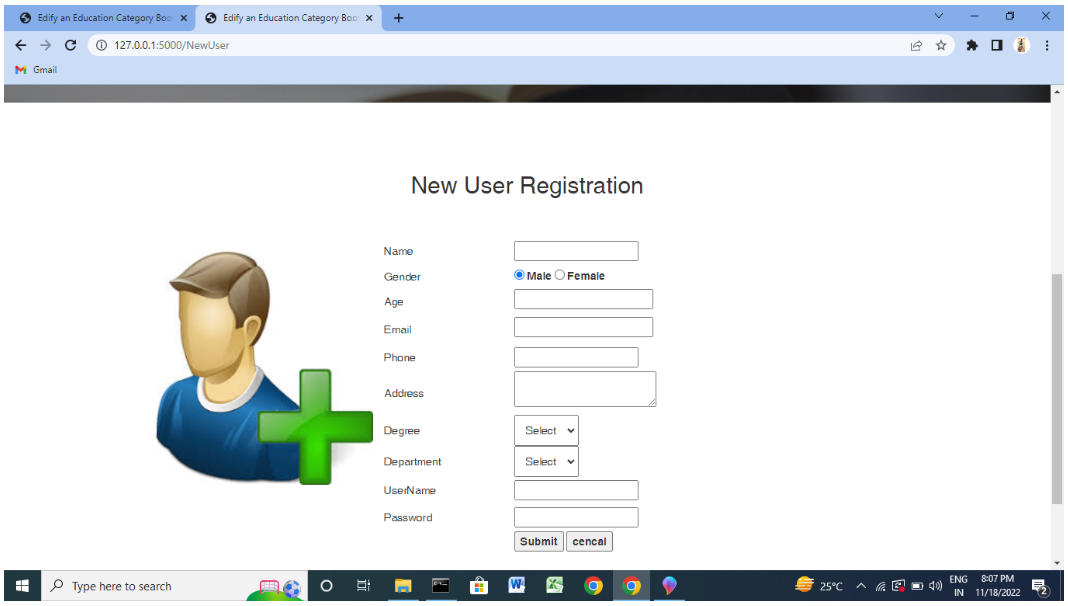
8.2 User Acceptance Testing:

Real world testing was also done, by giving to remote users and asking them to use the application. Their difficulties were fixed and tested again until all the issues were fixed.

9.RESULTS

9.1 PERFORMANCE METRICS

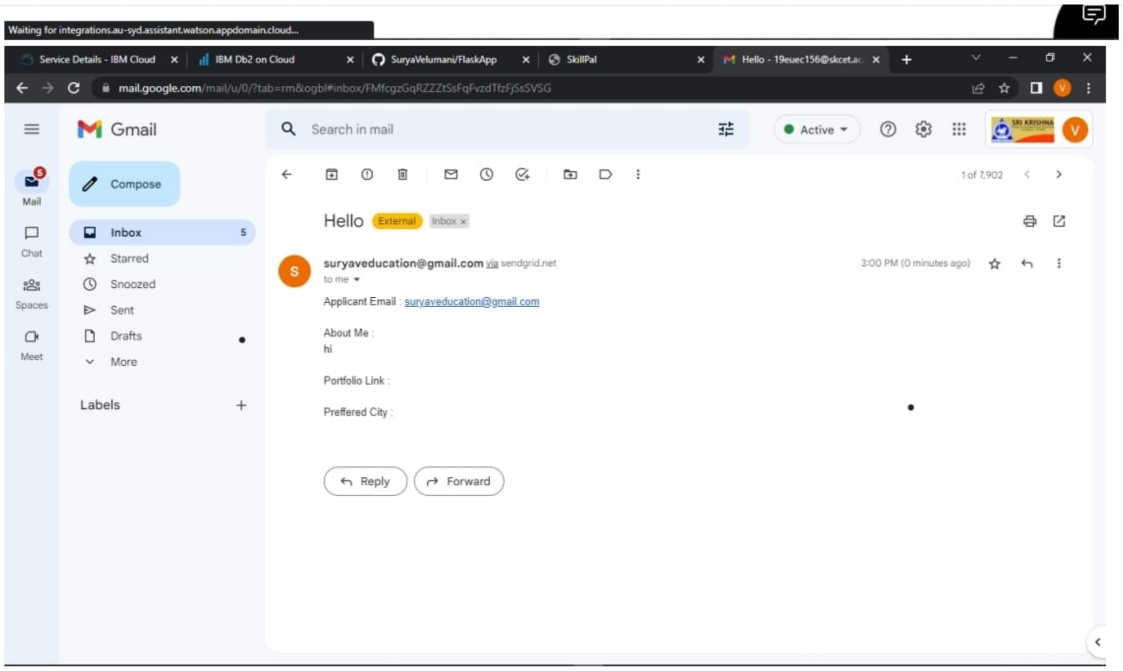




New User Registration

Name	
Gender	[✓ Male] Female
Age	
Email	
Phone	
Address	
Degree	Select
Department	Select
UserName	
Password	

Submit cencal



Hello External Inbox

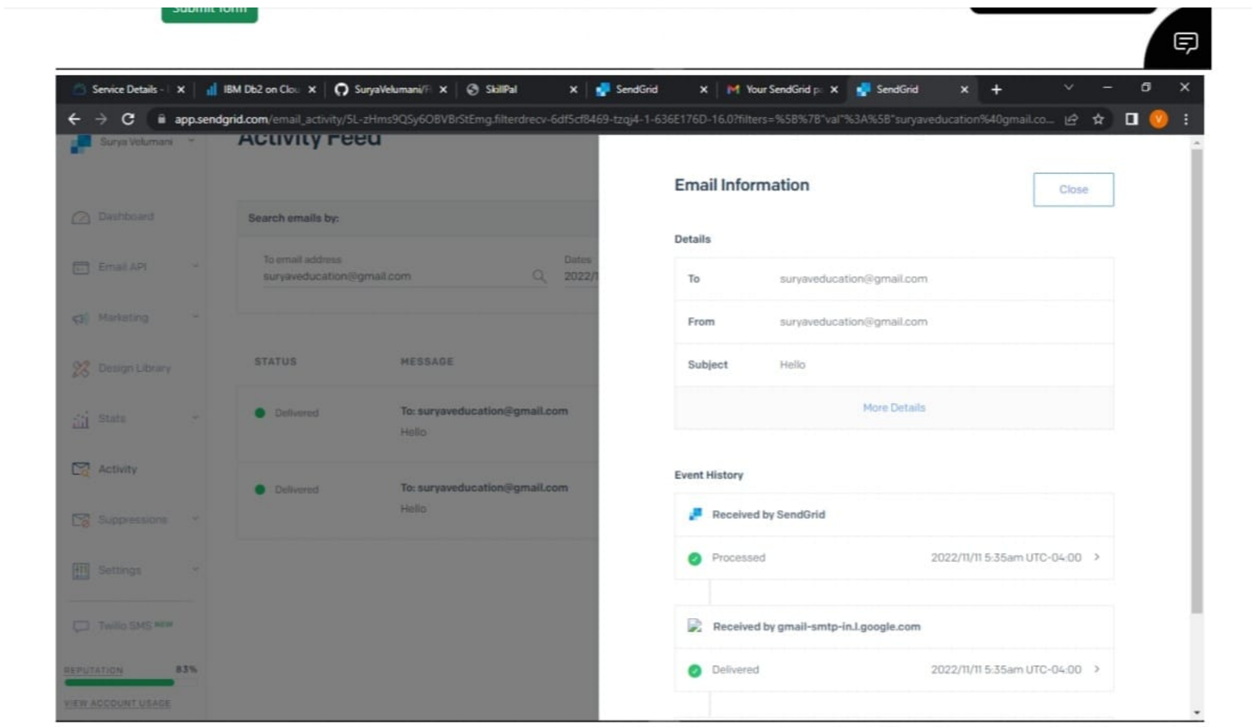
suryaveducation@gmail.com via sendgrid.net
to me
3:00 PM (0 minutes ago)

Applicant Email : suryaveducation@gmail.com

About Me :
hi

Portfolio Link :

Preffered City :



app.sendgrid.com/email_activity/5L-zHms9QSy6OBVBrStEmg.filterdrecv-6df5cf8469-tzqj4-1-636E176D-16.0?filters=%5B%7B"val"%3A%5B"suryaveducation%40gmail.co...

Activity Feed

Search emails by:

To email address: suryaveducation@gmail.com

Dates: 2022/1

STATUS	MESSAGE
● Delivered	To: suryaveducation@gmail.com Hello
● Delivered	To: suryaveducation@gmail.com Hello

Email Information

Details

To	suryaveducation@gmail.com
From	suryaveducation@gmail.com
Subject	Hello

Event History

Received by SendGrid
Processed 2022/11/11 5:35am UTC-04:00

Received by gmail-smtp-in.l.google.com
Delivered 2022/11/11 5:35am UTC-04:00

10. ADVANTAGE AND DISADVANTAGE

ADVANTAGE :

- ◎It helps candidates to search the job which perfectly suites them and make them aware of all the job openings.
- ◎ It help recruiters of the company to choose the right candidates for their organisations with appropriate skills.
- ◎Since it is cloud application , it does require any installation of softwares and is portable.

DISADVANTAGE:

- ◎ It is costly.
- ◎ Uninterrupted internet connection is required for smooth functioning of application.

11. CONCLUSION

we have used ibm cloud services like db2, cloud registry , kubernetes , Watson assistant to create this application , which will be very usefull for candidates who are searching for job and as well as for the company to select the right candidate for their organization.

12. FUTURE SCOPE

Future directions of our work will focus on performing a more exhaustive evaluation considering a greater amount of methods and data as well as a comprehensive evaluation of the impact of each professional skill of a job seeker on the received job recommendation. We can use machine learning technicques to recommend data in a efficient way.

13.APPENDIX

Source Code:

```
from flask import Flask, render_template, flash, request, session
from flask import render_template, redirect, url_for, request

import json
from json2html import *

import ibm_db
import pandas
import ibm_db_dbi
from sqlalchemy import create_engine

engine = create_engine('sqlite://',
                       echo = False)

dsn_hostname = "55fbc997-9266-4331-afd3-
888b05e734c0.bs2io90l08kqb1od8lcg.databases.appdomain.clo
ud"
dsn_uid = "dnz18939"
dsn_pwd = "1WTmeeAdIRpoG24v"

dsn_driver = "{IBM DB2 ODBC DRIVER}"
dsn_database = "BLUDB"
dsn_port = "31929"
dsn_protocol = "TCPIP"
dsn_security = "SSL"
```

```
dsn = (
    "DRIVER={0};"
    "DATABASE={1};"
    "HOSTNAME={2};"
    "PORT={3};"
    "PROTOCOL={4};"
    "UID={5};"
    "PWD={6};"
    "SECURITY={7};").format(dsn_driver, dsn_database,
dsn_hostname, dsn_port, dsn_protocol, dsn_uid,
dsn_pwd,dsn_security)


try:
    conn = ibm_db.connect(dsn, "", "")
    print ("Connected to database: ", dsn_database, "as user: ",
dsn_uid, "on host: ", dsn_hostname)

except:
    print ("Unable to connect: ", ibm_db.conn_errormsg() )

app = Flask(__name__)
app.config['DEBUG']
app.config['SECRET_KEY'] = '7d441f27d441f27567d441f2b6176a'
```

```
@app.route("/")
def homepage():

    return render_template('index.html')




@app.route("/Home")
def Home():
    return render_template('index.html')

@app.route("/AdminLogin")
def AdminLogin():
    return render_template('AdminLogin.html')

@app.route("/NewUser")
def NewUser():
    return render_template('NewUser.html')

@app.route("/NewCompany")
def NewCompany():
    return render_template('NewCompany.html')
```

```
@app.route("/UserLogin")
def StudentLogin():
   return render_template('UserLogin.html')

@app.route("/CompanyLogin")
def CompanyLogin():
   return render_template('CompanyLogin.html')

@app.route("/Search")
def Search():
   return render_template('Search.html')



@app.route("/AdminHome")
def AdminHome():

   conn = ibm_db.connect(dsn, "", "")
   pd_conn = ibm_db_dbi.Connection(conn)
   selectQuery = "SELECT * from regtb "
   dataframe = pandas.read_sql(selectQuery, pd_conn)

   dataframe.to_sql('Employee_Data', con=engine,
if_exists='append')
   data = engine.execute("SELECT * FROM
Employee_Data").fetchall()
   return render_template('AdminHome.html', data=data)
```

```
@app.route("/ACompanyInfo")
def ACompanyInfo():

    conn = ibm_db.connect(dsn, "", "")
    pd_conn = ibm_db_dbi.Connection(conn)
    selectQuery = "SELECT * from companytb "
    dataframe = pandas.read_sql(selectQuery, pd_conn)

    dataframe.to_sql('Employee_Data', con=engine,
if_exists='append')
    data = engine.execute("SELECT * FROM
Employee_Data").fetchall()

    return render_template('ACompanyInfo.html', data=data)


@app.route("/AjobInfo")
def AjobInfo():

    conn = ibm_db.connect(dsn, "", "")
    pd_conn = ibm_db_dbi.Connection(conn)
    selectQuery = "SELECT * from jobtb "
    dataframe = pandas.read_sql(selectQuery, pd_conn)
    dataframe.to_sql('Employee_Data', con=engine,
if_exists='append')
    data = engine.execute("SELECT * FROM
Employee_Data").fetchall()

    return render_template('AjobInfo.html', data=data)

@app.route("/SCompanyInfo")
def SCompanyInfo():

    conn = ibm_db.connect(dsn, "", "")
    pd_conn = ibm_db_dbi.Connection(conn)
    selectQuery = "SELECT * from jobtb "
    dataframe = pandas.read_sql(selectQuery, pd_conn)

    dataframe.to_sql('Employee_Data', con=engine,
if_exists='append')
    data = engine.execute("SELECT * FROM
Employee_Data").fetchall()

    return render_template('SCompanyInfo.html', data=data)

@app.route("/CompanyHome")
def CompanyHome():
    return render_template('CompanyHome.html')

@app.route("/UserHome")
```

```
def UserHome():

    uname=  session['uname']

    conn = ibm_db.connect(dsn, "", "")
    pd_conn = ibm_db_dbi.Connection(conn)
    selectQuery = "SELECT * FROM regtb where Username='"+
uname +"' "
    dataframe = pandas.read_sql(selectQuery, pd_conn)
    dataframe.to_sql('Employee_Data', con=engine,
if_exists='append')
    data = engine.execute("SELECT * FROM
Employee_Data").fetchall()

    return render_template('UserHome.html', data=data)



@app.route("/CJobInfo")
def CJobInfo():

    cname=  session['cname']

    conn = ibm_db.connect(dsn, "", "")
    pd_conn = ibm_db_dbi.Connection(conn)
    selectQuery = "SELECT * FROM jobtb where Cname='"+ cname
+"' "
```

```
    dataframe = pandas.read_sql(selectQuery, pd_conn)
    dataframe.to_sql('Employee_Data', con=engine,
if_exists='append')
    data = engine.execute("SELECT * FROM
Employee_Data").fetchall()


    return render_template('CJobInfo.html', data=data)

@app.route("/adminlogin", methods=['GET', 'POST'])
def adminlogin():
    error = None
    if request.method == 'POST':
       if request.form['uname'] == 'admin' or
request.form['password'] == 'admin':



           conn = ibm_db.connect(dsn, "", "")
           pd_conn = ibm_db_dbi.Connection(conn)
           selectQuery = "SELECT * FROM regtb  "
           dataframe = pandas.read_sql(selectQuery, pd_conn)
           dataframe.to_sql('Employee_Data', con=engine,
if_exists='append')
           data = engine.execute("SELECT * FROM
Employee_Data").fetchall()
```

```
            return render_template('AdminHome.html', data=data)

        else:
         return render_template('index.html', error=error)


@app.route("/userlogin", methods=['GET', 'POST'])
def userlogin():
    error = None
    if request.method == 'POST':

        username = request.form['uname']
        password = request.form['password']

        conn = ibm_db.connect(dsn, "", "")

        pd_conn = ibm_db_dbi.Connection(conn)

        selectQuery = "SELECT * from regtb where UserName='" +
username + "' and password='" + password + "'"
        dataframe = pandas.read_sql(selectQuery, pd_conn)

        if dataframe.empty:
            data1 = 'Username or Password is wrong'
            return render_template('goback.html', data=data1)
        else:
            print("Login")
```

```
          selectQuery = "SELECT * from regtb where UserName='" +
username + "' and password='" + password + "'"
          dataframe = pandas.read_sql(selectQuery, pd_conn)

          dataframe.to_sql('Employee_Data',
                    con=engine,
                    if_exists='append')

          # run a sql query
          print(engine.execute("SELECT * FROM
Employee_Data").fetchall())

          return render_template('UserHome.html',
data=engine.execute("SELECT * FROM
Employee_Data").fetchall())


@app.route("/companylogin", methods=['GET', 'POST'])
def companylogin():
    error = None
    if request.method == 'POST':

        uname = request.form['uname']
        password = request.form['password']
        session['cname'] = uname

        conn = ibm_db.connect(dsn, "", "")
        pd_conn = ibm_db_dbi.Connection(conn)

        selectQuery = "SELECT * from companytb where 
UserName='" + uname + "' and password='" + password + "'"
        dataframe = pandas.read_sql(selectQuery, pd_conn)

        if dataframe.empty:
            data1 = 'Username or Password is wrong'
            return render_template('goback.html', data=data1)
        else:
            print("Login")
            selectQuery = "SELECT * from companytb where 
UserName='" + uname + "' and password='" + password + "'"
            dataframe = pandas.read_sql(selectQuery, pd_conn)

            dataframe.to_sql('Employee_Data',
                             con=engine,
                             if_exists='append')

            # run a sql query
            print(engine.execute("SELECT * FROM 
Employee_Data").fetchall())

            return render_template('CompanyHome.html', 
data=engine.execute("SELECT * FROM 
Employee_Data").fetchall())
```

```
@app.route("/NewStudent1", methods=['GET', 'POST'])
def NewStudent1():
    if request.method == 'POST':

        name = request.form['name']
        gender = request.form['gender']
        Age = request.form['Age']
        email = request.form['email']
        pnumber = request.form['pnumber']
        address = request.form['address']
        Degree = request.form['Degree']
        depat = request.form['depat']
        uname = request.form['uname']
        passw = request.form['passw']

    conn = ibm_db.connect(dsn, "", "")

    insertQuery = "insert into regtb values('" + name + "','" + gender +
"','" + Age + "','" + email + "','" + pnumber + "','" + address + "','" +
Degree + "','" + depat + "','" + uname + "','" + passw + "')"
    insert_table = ibm_db.exec_immediate(conn, insertQuery)

    sendmsg(email, "Successfully registered this website")

    data1 = 'Record Saved!'
    return render_template('goback.html', data=data1)
```

```
@app.route("/newcompany", methods=['GET', 'POST'])
def newcompany():
    if request.method == 'POST':

        cname = request.form['cname']
        regno = request.form['regno']
        mobile = request.form['mobile']

        email = request.form['email']
        Website = request.form['Website']
        address = request.form['address']
        uname = request.form['uname']
        passw = request.form['passw']

        conn = ibm_db.connect(dsn, "", "")

        insertQuery = "insert into companytb
values('"+cname+"','"+regno+"','"+mobile+"','"+email+"','"+Website+"','"
+address+"','"+uname +"','"+passw+"')"
        insert_table = ibm_db.exec_immediate(conn, insertQuery)

        data1 = 'Record Saved!'
        return render_template('goback.html', data=data1)
```

```
@app.route("/newjob", methods=['GET', 'POST'])
def newjob():
    if request.method == 'POST':
        cnn = session['cname']
        cname = request.form['cname']
        cno = request.form['cno']
        Address = request.form['Address']
        JobLocation = request.form['JobLocation']
        Vacancy = request.form['Vacancy']
        Job = request.form['Job']
        Department = request.form['depat']
        website = request.form['website']

        conn = ibm_db.connect(dsn, "", "")

        insertQuery = "insert into jobtb values('" + cname + "','" + cno +
"','" + Address + "','" + JobLocation + "','" + Vacancy + "','" + Job + "','" +
Department + "','" + website + "','"+cnn+"')"
        insert_table = ibm_db.exec_immediate(conn, insertQuery)

        conn = ibm_db.connect(dsn, "", "")
        pd_conn = ibm_db_dbi.Connection(conn)
        selectQuery1 = "SELECT  *  FROM  regtb where
```

```
Department='" + Department + "'"
        dataframe = pandas.read_sql(selectQuery1, pd_conn)

        dataframe.to_sql('regtb', con=engine, if_exists='append')
        data1 = engine.execute("SELECT * FROM regtb").fetchall()

        for item1 in data1:
            Mobile = item1[5]
            Email = item1[4]
            sendmsg(Email,"Jop Title"+Job + " More Info Visit
Website")




    data = 'Record Saved!'
    return render_template("goback.html", data=data)




@app.route("/jobsearch", methods=['GET', 'POST'])
def jobsearch():
    if request.method == 'POST':
       jobname = request.form['name']
```

```
        url = "https://linkedin-jobs-search.p.rapidapi.com/"

        payload = {
            "search_terms": jobname,
            "location": "india",
            "page": "1"
        }
        headers = {
            "content-type": "application/json",
            "X-RapidAPI-Key":
"dbc572baf7msh1cbca677f83bd5dp1b070djsna1bfebcbab06",
            "X-RapidAPI-Host": "linkedin-jobs-search.p.rapidapi.com"
        }

        response = requests.request("POST", url, json=payload,
headers=headers)

        print(response.text)

        infoFromJson = json.loads(response.text)
        print(json2html.convert(json=infoFromJson))

        nutrients = {}

        data = json.loads(response.text)
        concepts = data['foods'][0]['foodNutrients']
        arr = ["Sugars", "Energy", "Vitamin A", "Vitamin D", "Vitamin B",
"Vitamin C", "Protein", "Fiber", "Iron",
            "Magnesium",
            "Phosphorus", "Cholestrol", "Carbohydrate", "Total lipid
(fat)", "Sodium", "Calcium", ]
      for x in concepts:
         if x['nutrientName'].split(',')[0] in arr:
            if (x['nutrientName'].split(',')[0] == "Total lipid (fat)"):
               nutrients['Fat'] = str(x['value']) + " " + x['unitName']
            else:
               nutrients[x['nutrientName'].split(',')[0]] = str(x['value']) +
" " + x['unitName']

      return render_template('display.html', x=jobname,
data=nutrients, account=session['username'])
```

```
#send grid

# https://github.com/sendgrid/sendgrid-pythonimport osfrom
sendgrid import SendGridAPIClientfrom sendgrid.helpers.mail
import Mail message = Mail(
from_email='from_email@example.com',
to_emails='to@example.com', subject='Sending with Twilio
SendGrid is Fun', html_content='<strong>and easy to do anywhere,
even with Python</strong>')try: sg =
SendGridAPIClient(os.environ.get('SENDGRID_API_KEY'))
response = sg.send(message) print(response.status_code)
print(response.body) print(response.headers)except Exception as
e: print(e.message)

def sendmsg(Mailid,message):
    import smtplib
    from email.mime.multipart import MIMEMultipart
    from email.mime.text import MIMEText
    from email.mime.base import MIMEBase
    from email import encoders

    fromaddr = "sampletest685@gmail.com"
    toaddr = Mailid

    # instance of MIMEMultipart
    msg = MIMEMultipart()

    # storing the senders email address
    msg['From'] = fromaddr

    # storing the receivers email address
    msg['To'] = toaddr
    # storing the subject
    msg['Subject'] = "Alert"

    # string to store the body of the mail
    body = message

    # attach the body with the msg instance
    msg.attach(MIMEText(body, 'plain'))

    # creates SMTP session
    s = smtplib.SMTP('smtp.gmail.com', 587)

    # start TLS for security
    s.starttls()

    # Authentication
    s.login(fromaddr, "hneucvnontsuwgpj")

    # Converts the Multipart msg into a string
    text = msg.as_string()

    # sending the mail
    s.sendmail(fromaddr, toaddr, text)

    # terminating the session

if __name__ == '__main__':
```

```
app.run(host='0.0.0.0', debug='TRUE')
```