

IOT Internship Project:

Soldier Health Monitoring System Using Ibm Cloud

under

Smartinterz

Under the Guidelines:

Mrs. Divya Nemuri madam

By

A. Javeed

A. Durga Prasad

A. Keerti Vardhan

A. Chaitanya Sri Krishna

V. Kesava Pavan Sai

Introduction:

Overview:

In this project we are going to build soldiers health monitoring system using IBM cloud services, we created virtual IOT device in the IBM cloud and we used node red services in the IBM cloud to build our project and we also used IBM Watson services to vary the inputs of our device virtually such as pulse rate, body temperature by signing into an account by using the ID of the device. We also decide the location of the soldier by using location sensors, and we will see the result in node red services.

Purpose:

Purpose of this project is to watch Soldier's health. Because, Soldier health monitoring is especially important and essential one. while soldiers are staying in different environment conditions and in various locations, Due to environmental conditions like extreme cold areas, extreme hot areas and due to all this climate changes soldier health may be affected. with the help of this project, we're also able to know the location of the soldier so it is an extremely useful project to track the soldier's location and we're also able to give him the proper medication while he is not feeling healthy, so this device is extremely useful during wars also. So, this project is extremely useful in military, Air Force and Navy applications. Not only useful for soldiers' health monitoring it is also useful for patient's health monitoring. every nation will take care of their soldiers because they are the

treasures to their nation. So, we are glad to do this project called soldier health monitoring using IBM cloud.

Literature survey:

Existing problem:

The main existing Problem is we are unable to know the health condition of a soldier while he is in his duty, mainly soldiers north side of India are doing their jobs in extreme freezing conditions so it is exceedingly difficult to watch the health conditions while they're in duty, and we are also unable to find the location of the soldier while he is sick in the war are in field work. So, to find the location of your soldier and to give proper medication to a soldier is becoming an exceedingly awkward thing to military health care department.



Recently Indonesia soldiers are dived into the sea with the submarine, but due to some problem in the submarine we are unable to find the location of the submarine and the health condition of the soldiers, due to uncertainty of soldiers and

submarine location, finally soldiers were died. to avoid this type of problems in future we are doing this project.

Proposed solution:

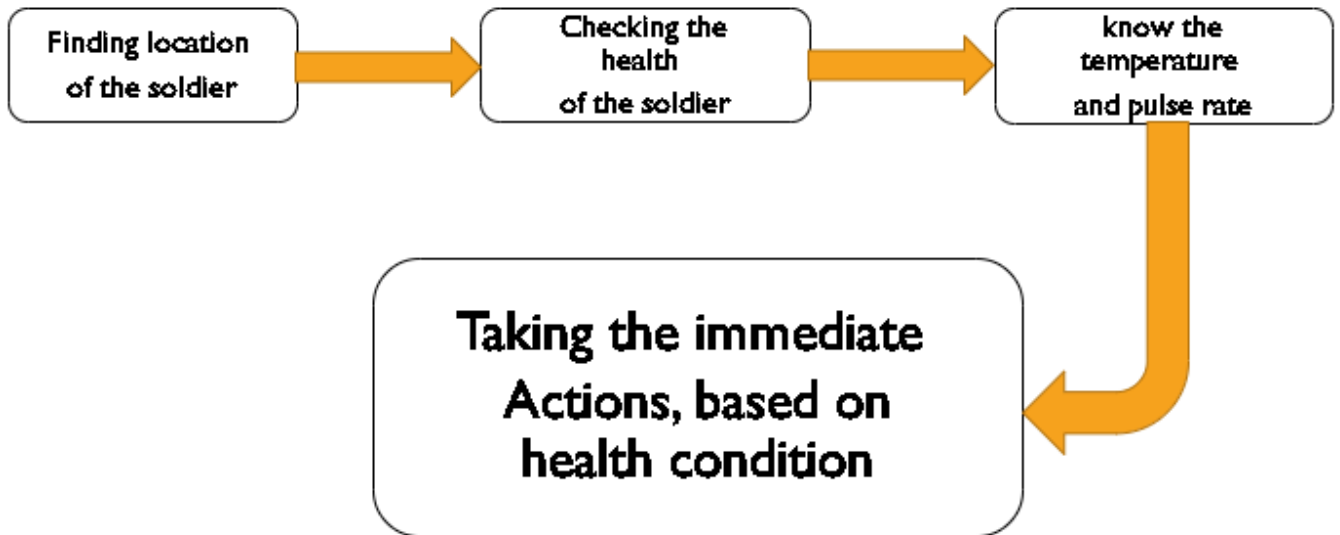
Our proposed solution is the perfect solution to overcome this type of existing problems. Our proposed solution is to find the area of the soldier by using location sensor, we also measure temperature and pulse rate of the soldier regularly, so, when we can know that soldier pulse rate and temperature, we are able to know health of the soldier regularly, then we at once trace his location with the help of location sensing element and we able to find a soldier and we're able to give him perfect medication when soldier health condition is critical.

Our solution is also very help full in the war time, and it is also helpful in the situation like Indonesian's submarine submerge, unknown extreme cold areas, unknown wilderness



Theoretical analysis:

Block Diagram:



Hardware / Software designing

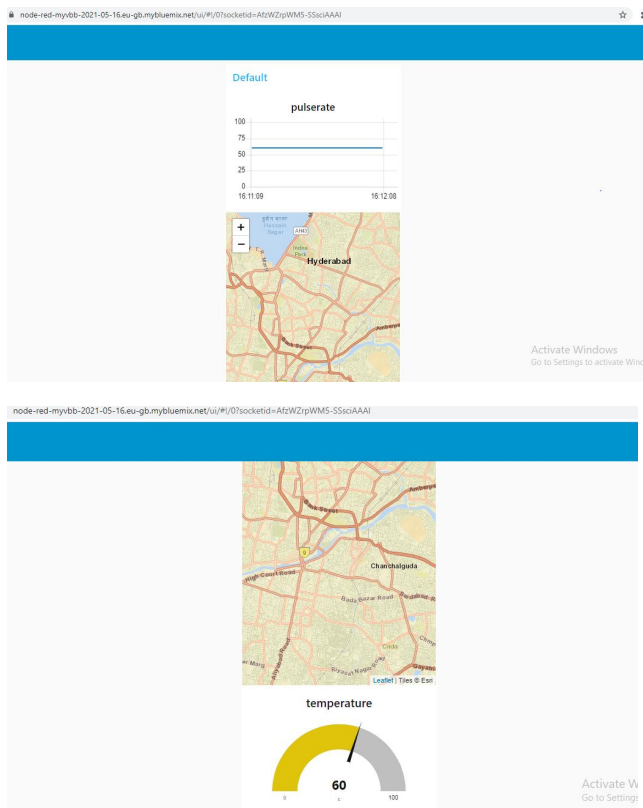
We follow a step-by-step procedure to set up all the interfaces needed for our project and develop the code in python to measure the temperature, pulse rate and location of the soldier. The following software is needed:

- Python Idle (with specified packages installed)
- IBM cloud
- Node Red service
- IBM Watson Service

Experimental Investigations:

In our project firstly we created the virtual IOT device in the IBM cloud services and we also created the Node Red service to connect the required nodes to build our project, then we also get the location link to locate the soldiers position and we added it as a node in the node red services, we runned the python code which is written in the Python Idle to measure pulse rate and temperature of the soldier, and we varied the temperature and pulse rate of the soldier virtually in the IBM Watson Service.

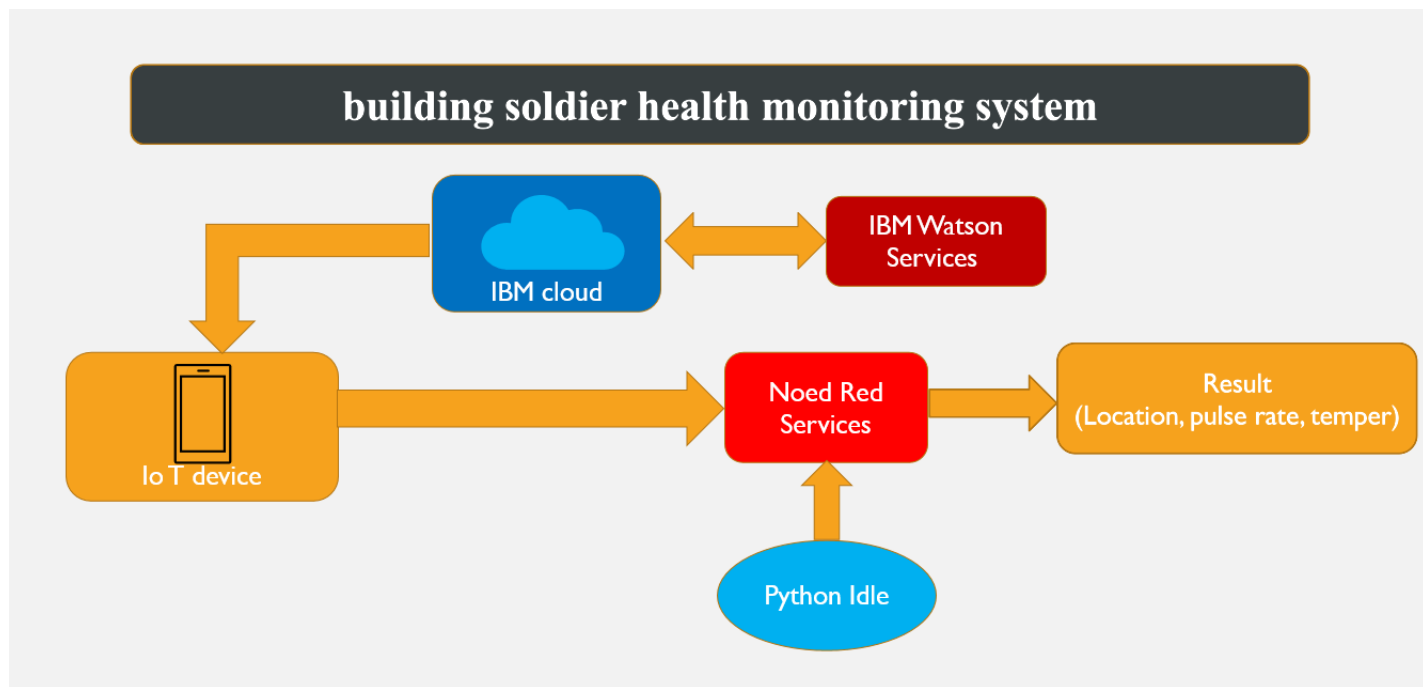
So, finally we can find the location of the soldier and we're also able to see the pulse rate and temperature of the soldier by varying the values in the IBM Watson Service Virtually.



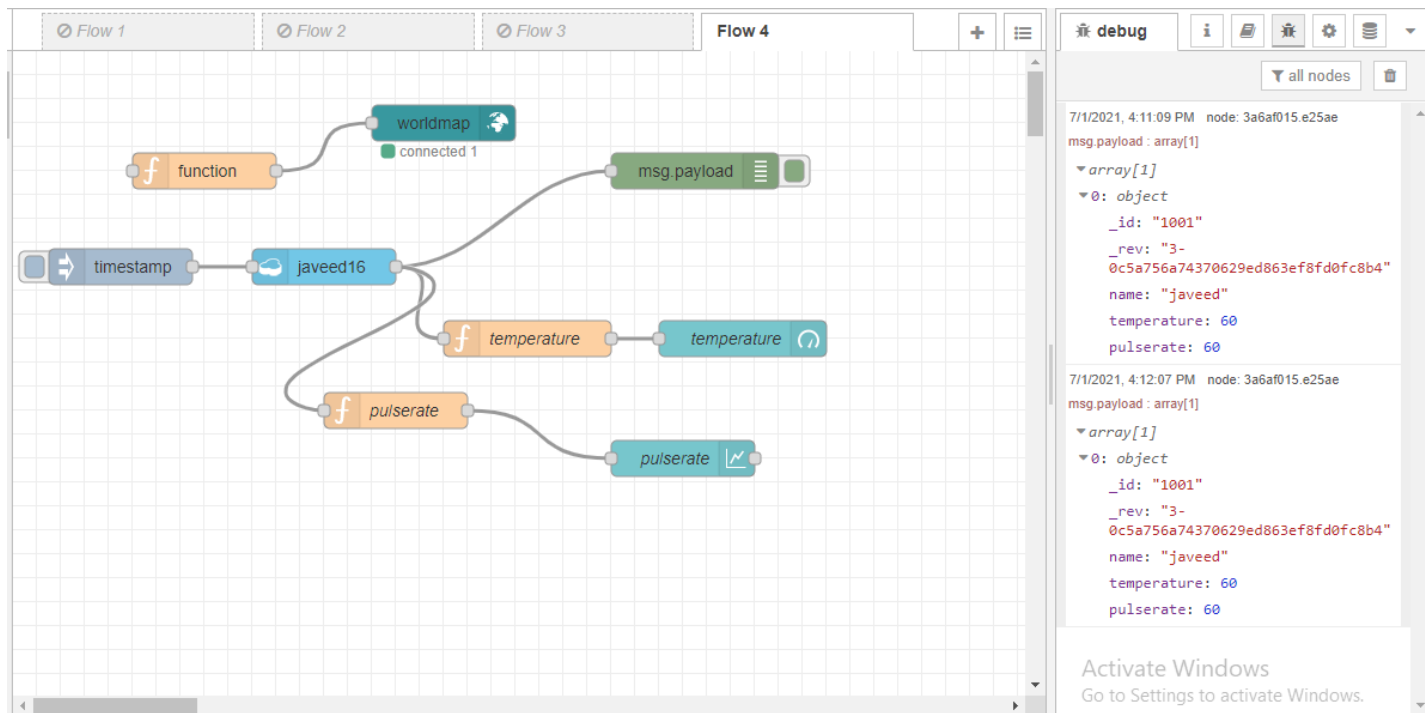
IBM Watson Service

Varying the input results in

Flow Chart:



Result:



Explanation:

The result of our project fulfils our requirements. We have successfully

Executed the designed program and the results have been as expected. Finally, this helps to find the location of the soldier and it also helped to measure the pulse rate and temperature of the soldier. The method proves that it is effective and efficient to watch the health of the soldiers successfully.

Advantages:

1. We can find the location of the soldiers.
2. We can check the health conditions of the soldiers regularly.
3. We can give medication at once while soldiers health is not good
4. We can know the health condition when he is in the war.

Disadvantages:

1. We can't expect same results in hardware development.
2. Our device may get damaged.
3. If our cloud system is hacked, soldiers' life will be in risk.
4. We may also get network disturbance in finding locations of the soldier.

Applications:

1. The main application of this project is to check the health and location of the soldier.
2. We can also use it in navy applications.
3. We can also use this project to check normal patients' health.
4. We can also use this for wildlife officers to check health and location.

Conclusion:

We are able to developed soldier health monitoring system using IBM cloud services, we are able to know soldier pulse rate, temperature and location.

Future Scope:

soldier health monitoring system plays a crucial role in military and navy applications. And this project also had huge scope in patients monitoring.

Bibliography:

Appendix:

```
# IBM Cloudant Legacy authentication
```

```
client = Cloudant("apikey-v2-  
26cvscqymokovdj7rs2u5j0twv2ogcs0x7ewms4tww3a",  
"a161324b37fc53e709f38f4bc1ee5881",
```

```
    url="https://apikey-v2-  
26cvscqymokovdj7rs2u5j0twv2ogcs0x7ewms4tww3a:a161324b  
37fc53e709f38f4bc1ee5881@2a771bbb-85a9-44ca-acb4-  
2b418ff3ab10-bluemix.cloudantnosqldb.appdomain.cloud")
```

```
client.connect()
```

```
database_name = "javeed16"
```

```

my_database = client.create_database(database_name)

if my_database.exists():
    print(f"'{database_name}' successfully created.")
    json_document = {
        "_id": "1001",
        "name": "javeed"
    }
    new_document =
my_database.create_document(json_document)
    if new_document.exists():
        print("Document '{new_document}' successfully created.")

result_collection = Result(my_database.all_docs,
include_docs=True)

# Get the result for matching a key
result = result_collection['1001'] #search by id, if id=1001

print("-----")
print("the data with id =1001 is")
print (result)
print("-----")

```

```
# Iterate over the result collection
```

```
for result in result_collection:
```

```
    print(result)# it will print all the records
```

```
# First retrieve the document
```

```
for document in my_database:
```

```
    my_document = my_database['1001']
```

```
# Update the document content
```

```
# This can be done as you would any other dictionary
```

```
my_document['temperature'] = 60
```

```
my_document['pulserate'] = 60
```

```
my_document['worldmap']
```

```
# You must save the document in order to update it on the  
database
```

```
my_document.save()
```

```
result_collection = Result(my_database.all_docs,  
include_docs=True)
```

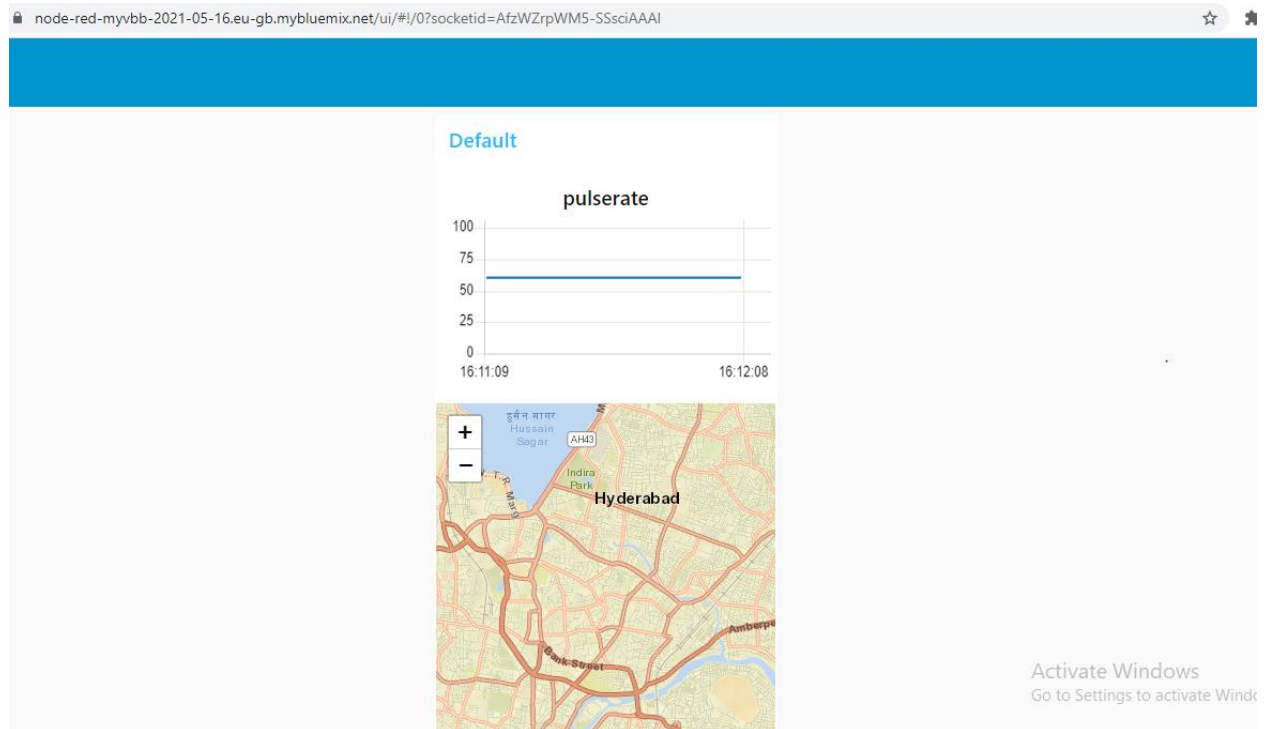
```
# Get the result for matching a key
```

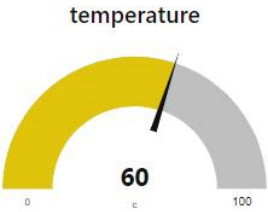
```
result = result_collection['1001']
```

```
# Iterate over the result collection
```

```
print (result)
```

UI output:





Activate W
Go to Settings