R2 Presentation

SSAD 29

MLIT Consight Solutions

Client And Team

MLIT Solutions

- Srinivas Chindam : CEO, MLIT solutions
- Archana Chindam
- Vinod Sir : Data Analytics Expert

TA Mentor

Venumadhav Kattagoni

Team Names

Atreyee Ghosal

Sai Venkat

Abhishek Nalla

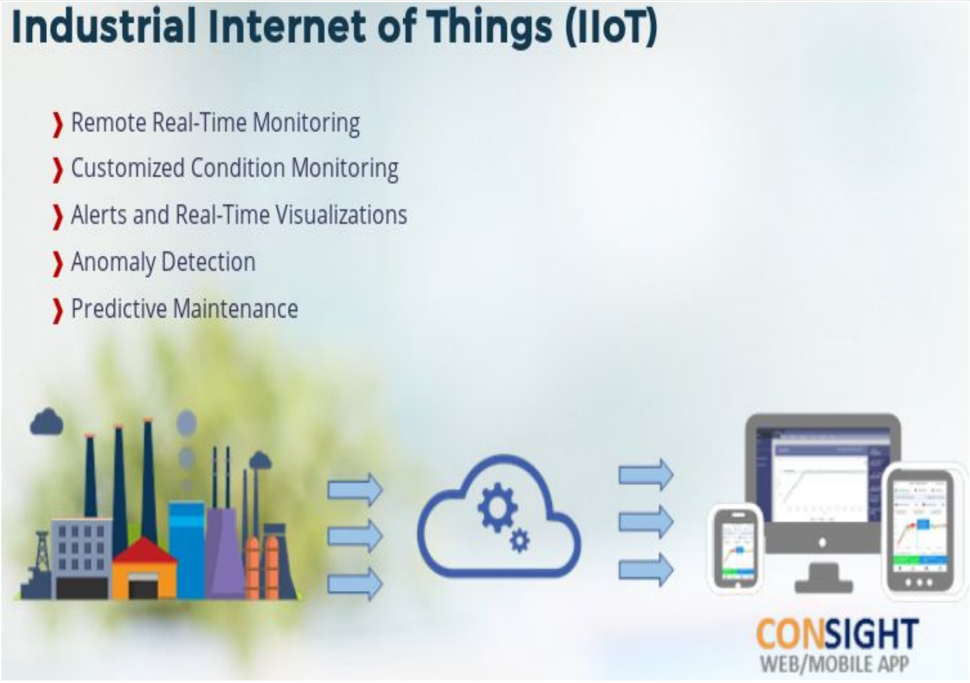
What is Consight?

Consight is an Industrial IoT application for remote condition monitoring of machinery, detecting anomalies and developing prediction based analytics for production efficiency. It is a WebApp, requiring the use of Python, PHP, HTML, CSS, iOS. Specifically, the project requires use of data analytics and UI development techniques.

Profile of Users

- The users are the customers who want the data from various industrial devices to be analysed and stored in representative forms like graphs and tables.
- More specifically, various indicators like temperature and pressure are measured by sensors located on-site which send data that needs to be processed and then analysed for the customers.

Usage Diagram



Industrial Internet of Things (IIoT)

- Remote Real-Time Monitoring
- Customized Condition Monitoring
- Alerts and Real-Time Visualizations
- Anomaly Detection
- Predictive Maintenance

Project Description

Phases

- Build software that can extract data and deliver basic reports on machines monitored. (No data analysis)

- Build software that can analyse data using data analytics techniques and deliver analytic reports

- If previous goals have been met: build software that can provide predictive reports of faults likely to happen for machines

Software Development Life Cycle

- The software development cycle used for this Project is Agile Scrum.
- We are given new requirements at the end of each phase of the Project.
- We are required to complete the given work in a specified amount of time.

Why it’s better than existing framework?

- The current framework used by MLIT is PHP
- Python is a better tool than PHP for processing large amounts of data
- The MLIT database has a large amount of data, that requires a fast language for processing
- Better Visualization by using Js

Attention areas

- Protected Access to Data.
- Fast Data Processing.
- Extracting SQL Data faster.

Softwares

Backend:

1. Python
2. MySQLdb

User Interface:

3. PHP

Visualization:

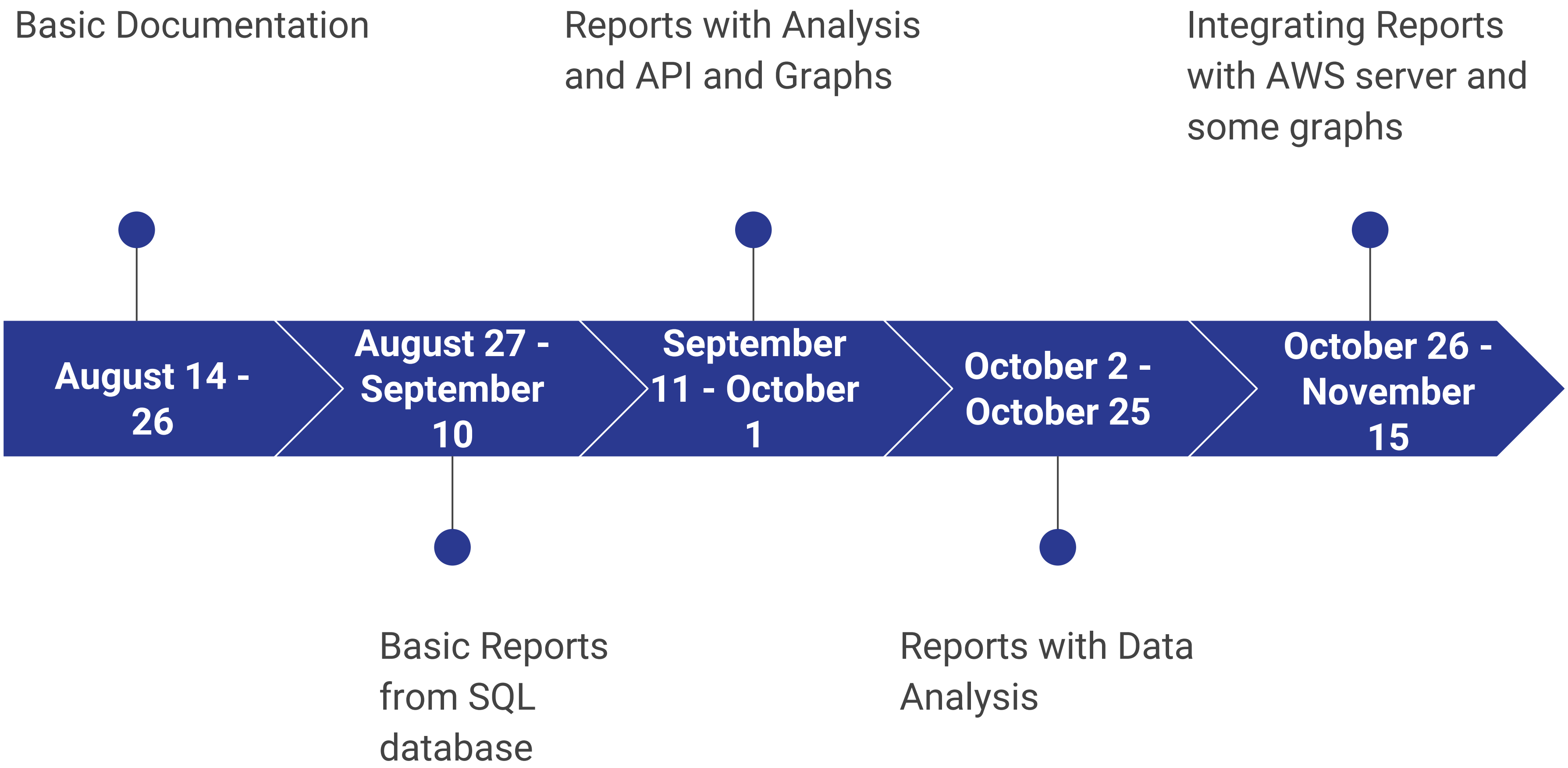
Charts.js

Matplotlib

A bit about softwares

- Python is an open source and an object-oriented language which is easy to implement.
- PHP is an open-source and a basic server language.
- Java script

Project Timeline



August 14 - 26

Basic Documentation

August 27 - September 10

Basic Reports from SQL database

September 11 - October 1

Reports with Analysis and API and Graphs

October 2 - October 25

Reports with Data Analysis

October 26 - November 15

Integrating Reports with AWS server and some graphs

What Has Been Done For R2

- Software that generates reports with some data analysis meeting eight different sets of requirements.
- A sample API to test the reports generated.
- Integration with Consight Development AWS server.
- Generation of graph visualisation.

PHP

- PHP is used for extracting information from cloud and inserting into the database.

- Our objective is to optimise the PHP part of the data transfer.

Reports

Report 1

This gives the data of the latest data entered into a particular machine.

This can be used for validation of set values

Parameters:Firm_id

Output:Json

Sample_Url:

http://13.126.193.159:8080/report/thresholds/1

Report 2

This report outputs anomaly data- the number of times a given sensor crosses the threshold in a cycle, and the value + time every time the sensor crosses threshold.

Parameters:Machine_id,Cycle_id

Output:Json

Sample_Url: http://13.126.193.159:8080/report/threshold_crossed/15/759

PHP

Optimised some code

Report 3

This report output data of Temperature,Humidity,

Pwr_status,Door_status,Fan_status for a particular day and cycle_id

This data is used for visualization of sudden drop of Temperature due door opening or pwr_off

Parameters:Date,Cycle_id

Output:Json

Sample_Url: http://13.126.193.159:8080/report/get/2017-10-20/799

Report 4

This gives the hatch_efficiency, total_eggs, total_chicks for a particular month

This data is used for visualization using pie charts

Parameters: Month

Output : Json

Sample_Url: /13.126.193.159:8080/report/hatch/2017-09http:/

Line Graph

This graph is generated using data from report5

As I have mentioned before it shows whether door is opened or not

From this user can know due to which values of the Temperature are reduced abruptly

Parameters: Date,Cycle_id

This is not yet deployed in client server

Pie Chart

This graph uses the data of report4

It shows the ratio of eggs to chicks

From this User can visualize chicks quantity

Parameters : Month

This is also not yet deployed in server

Report 5

This is used to extract Temperature, Pressure, Humidity, Methanol and time when the records are inserted for a particular cycle_id.

This data can be used for visualization.

Parameters: cycle_id

Output:Json

Sample URL:
http://13.126.193.159:8080/report/cycle/651

Report 6

This report is concerned with the duration of each cycle, more specifically, with the start time , the end time and the running time of the machine during that cycle.

This allows the customer to get knowledge of the start and end dates of each cycle to analyse data accordingly.

Output: Json

Parameters: None

Sample Url: 13.126.193.159:8080/report/get_cc

Report 7

This report is concerned with the data from the sensor. The report specifies the creation date of the sensors. It gives information about the temperatures returned by each sensor in each cycle.

This data is used by customer to check the variations in the temperature of the sensor for each specific cycle.

Output: Json

Parameters: None

Sample Url:
http://13.126.193.159:8080/report/get_mm

Report 8

This report gathers information from the hatcheries and calculates the hatch_efficiency, the broken and dead egg rates. This is ordered by the cycle id.

The customer can look upon the information from the hatcheries during each specific cycle and make room for improvement of hatch_efficiency.

Output: Json

Parameters: Date(Month)

Sample Url:
13.126.193.159:8080/report/hatch_c/2017-09

Report 9

This report is used to get data out of hatcheries database and use this data to calculate hatching efficiency, broken and dead egg efficiency for each machine id.

This data can be used by customers wanting to monitor the efficiency of hatcheries.

Output: Json

Parameters: Date(Month)

Sample Url: http://13.126.193.159:8080/report/hatch_m/2017-09

Plotly Graphs

There are a set of graphs plotted on plotly where whereas parameters like temperature and alert number have been plotted against time.

The customer can have an easy look at the graph and visualize the variations of temperature vs time

Or look upon the alerts vs time graph to obtain information about the alert type like starting of a sensor, halting of a sensor or the sensor being manually switched off.

Plotly Graphs

There are a set of graphs plotted on plotly where whereas parameters like temperature and alert number have been plotted against time.

The customer can have an easy look at the graph and visualize the variations of temperature vs time

Or look upon the alerts vs time graph to obtain information about the alert type like starting of a sensor, halting of a sensor or the sensor being manually switched off.

Histogram

A histogram has been plotted for the hatchery data for each specified month. The values or percentages of hatch efficiency of chicks or the percentage of broken and dead eggs has been plotted with the sql database provided.

Deploying And Troubleshooting the API on Development Server

The API was deployed on Development AWS server. This involved the following steps:

- Installing Python, setting up the virtual environment with all dependencies on AWS Development server.
- The first attempt was an attempt to integrate the API with AWS server using mod_wsgi. THis failed.
- The second attempt was deploying the API on flask server, and troubleshooting the issues involved.
- Finally, the API was deployed using Flask on the AWS server.

Steps For Deployment To Production Server

Preparation of a document detailing the steps needed to deploy the API to production AWS server.

- Data validation - tests to ensure the data delivered in the reports is correct and in the correct format.
- Unit testing - running unit tests on the API
- Installing a virtual environment on production server
- Installing dependencies on production server
- Running the API using Gunicorn (as Flask is not recommended for production deployment) on production server.

All of the above steps should be automated.

Thank Y0u