**Python Task for Smart Steel Technologies**

**Task:**

The task is to create two applications,

1. To load ***task\_data.csv*** file into a database.
2. To create a web application and server the information from the database. Also, log the information queried through the web application, in the same database.

**Prerequisites:**

In order to implement and run these application, below software/modules are required.

1. Python 3+
2. Python Modules:
   1. Flask
   2. pandas
   3. mysql-connector-python
   4. base64

**Justifications on development decisions:**

1. ***Web Framework:***

Since we have to create a web application, especially in Python, it requires a web framework to be constructed. For which, I have chosen Flask micro framework, as my choice.

***Reason:*** Flask is a micro web framework of Python. It has following advantages,

* Easy to implement.
* It is simple, flexible.
* It is light weight and shall be constructed and transported easily.

I found Flask would be the optimal for this task and hence chosen Flask.

1. ***Database:***

As this task requires a database to store the task\_data and access it via the web application, I have opted for MySQL database (8.0.16) as a Amazon RDS instance.

***Connectivity:***

The database shall be connected with the below information.

* **Host:** python-db.cxxjqwuxxbfj.eu-west-1.rds.amazonaws.com
* **User:** user\_one
* **Password:** Master#123
* **Database:** python\_test
* **Tables**: TASK\_DATA and LOG\_MASTER

***Why MySQL?***

MySQL is a relational database which is easy to understand and implement at any level of complexity of an application. The task demands basic operations among CRUD, for limited amount of data and hence I found MySQL would be my optimal choice, for database.

***Why RDS instance?***

Since the application is not being handled/accessed at a single place, there is no point of creating a MySQL database, locally. Hence, to access it from anywhere the code is being executed and due to its availability, secure and ease of access, I have opted for Amazon RDS instance for MySQL.

1. ***Base64 for password encryption:***

The password for MySQL database connection has to be hard coded in the web application and the data transfer application. In order to screen the same, I have used base64 module of Python to encrypt and decrypt the same. Even though this implementation is not 100% safe, it safe guards from normal password theft.

1. ***Querying Database from Frontend:***

Since the information from database shall be queried from the web application, I have decided the timestamp to be the criteria to fetch the information.

***Reason:***

After analyzing the information provided in the task\_data.csv, it is evident that the data is of temperature information at a particular time. Hence, it is optimal for the user to provide that particular timestamp [from and to], to retrieve the information. Giving the temperature/duration as input does not make any sense, hence, I opted for timestamp from and to.