Industrial Internship Report on

Password Generator

Prepared by

Richard Samuel

| Executive Summary |
|---|
| This report provides details of the Industrial Internship provided by upskill Campus and The IoT Academy in collaboration with Industrial Partner UniConverge Technologies Pvt Ltd (UCT).

This internship was focused on a project/problem statement provided by UCT. We had to finish the project including the report in 6 weeks' time.

My project was Password Generator it has two ways a user can generate easy and hard.

This internship gave me a very good opportunity to get exposure to Industrial problems and design/implement solution for that. It was an overall great experience to have this internship. |

TABLE OF CONTENTS

1   Preface

My six-week Python internship at Upskill Campus was a valuable learning experience. I enhanced my Python skills, tackled real-world challenges in cybersecurity, and delivered a practical tool that contributes to online security.

Throughout the internship, I honed my Python programming skills, particularly in areas related to user interface design. I also gained a deeper understanding of security best practices.

The central project of my internship was creating a password generator. This involved writing Python code to generate secure and random passwords, considering factors like length, complexity, and user preferences.

Opportunity given by USC/UCT: IoT, Cloud Computing, EDGE AI (Application of AI-ML)

How Program was planned
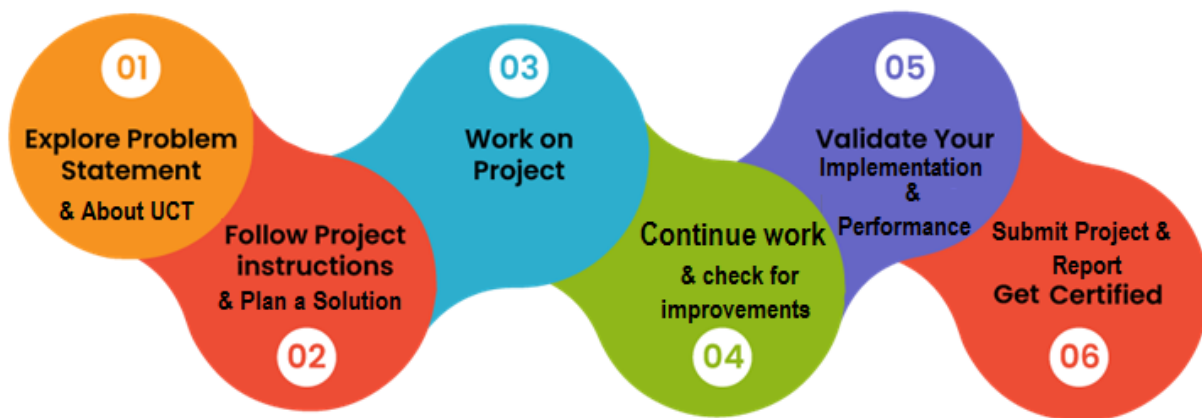
I encountered challenges related to balancing security with usability and optimizing the password generation algorithm.

Thanks to all Pragati, Satyam, Sankar who have helped me in every step.

Overcoming challenges helped me grow as a developer.

2 Introduction

2.1 About UniConverge Technologies Pvt Ltd

A company established in 2013 and working in Digital Transformation domain and providing Industrial solutions with prime focus on sustainability and RoI.

For developing its products and solutions it is leveraging various **Cutting Edge Technologies e.g. Internet of Things (IoT), Cyber Security, Cloud computing (AWS, Azure), Machine Learning, Communication Technologies (4G/5G/LoRaWAN), Java Full Stack, Python, Front end** etc.

i.   UCT IoT Platform (uct Insight)

UCT Insight is an IOT platform designed for quick deployment of IOT applications on the same time providing valuable "insight" for your process/business. It has been built in Java for backend and ReactJS for Front end. It has support for MySQL and various NoSql Databases.

- It enables device connectivity via industry standard IoT protocols - MQTT, CoAP, HTTP, Modbus TCP, OPC UA

- It supports both cloud and on-premises deployments.

It has features to
- Build Your own dashboard
- Analytics and Reporting
- Alert and Notification
- Integration with third party application(Power BI, SAP, ERP)
- Rule Engine
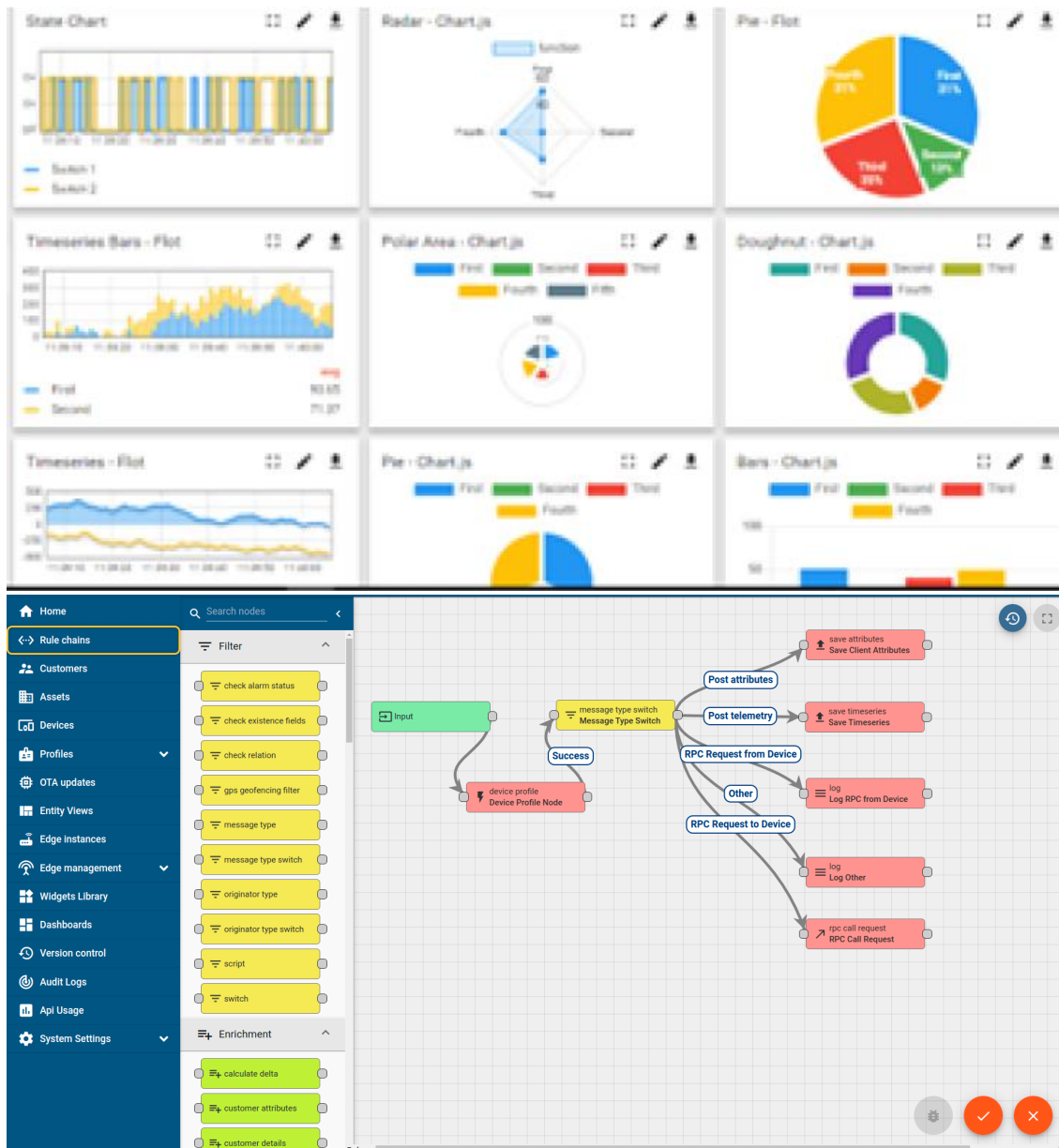
ii.   **Smart Factory Platform (** **FACTORY WATCH** **)**

Factory watch is a platform for smart factory needs.

It provides Users/ Factory

- with a scalable solution for their Production and asset monitoring

- OEE and predictive maintenance solution scaling up to digital twin for your assets.

- to unleased the true potential of the data that their machines are generating and helps to identify the KPIs and also improve them.

- A modular architecture that allows users to choose the service that they what to start and then can scale to more complex solutions as per their demands.

Its unique SaaS model helps users to save time, cost and money.

| Machine	Operator	Work Order ID	Job ID	Job Performance	Job Progress		Output		Rejection	Time (mins)				Job Status	End Customer
					Start Time	End Time	Planned	Actual		Setup	Pred	Downtime	Idle		
CNC_S7_81	Operator 1	WO0405200001	4168	58%	10:30 AM		55	41	0	80	215	0	45	In Progress	i
CNC_S7_81	Operator 1	WO0405200001	4168	58%	10:30 AM		55	41	0	80	215	0	45	In Progress	i
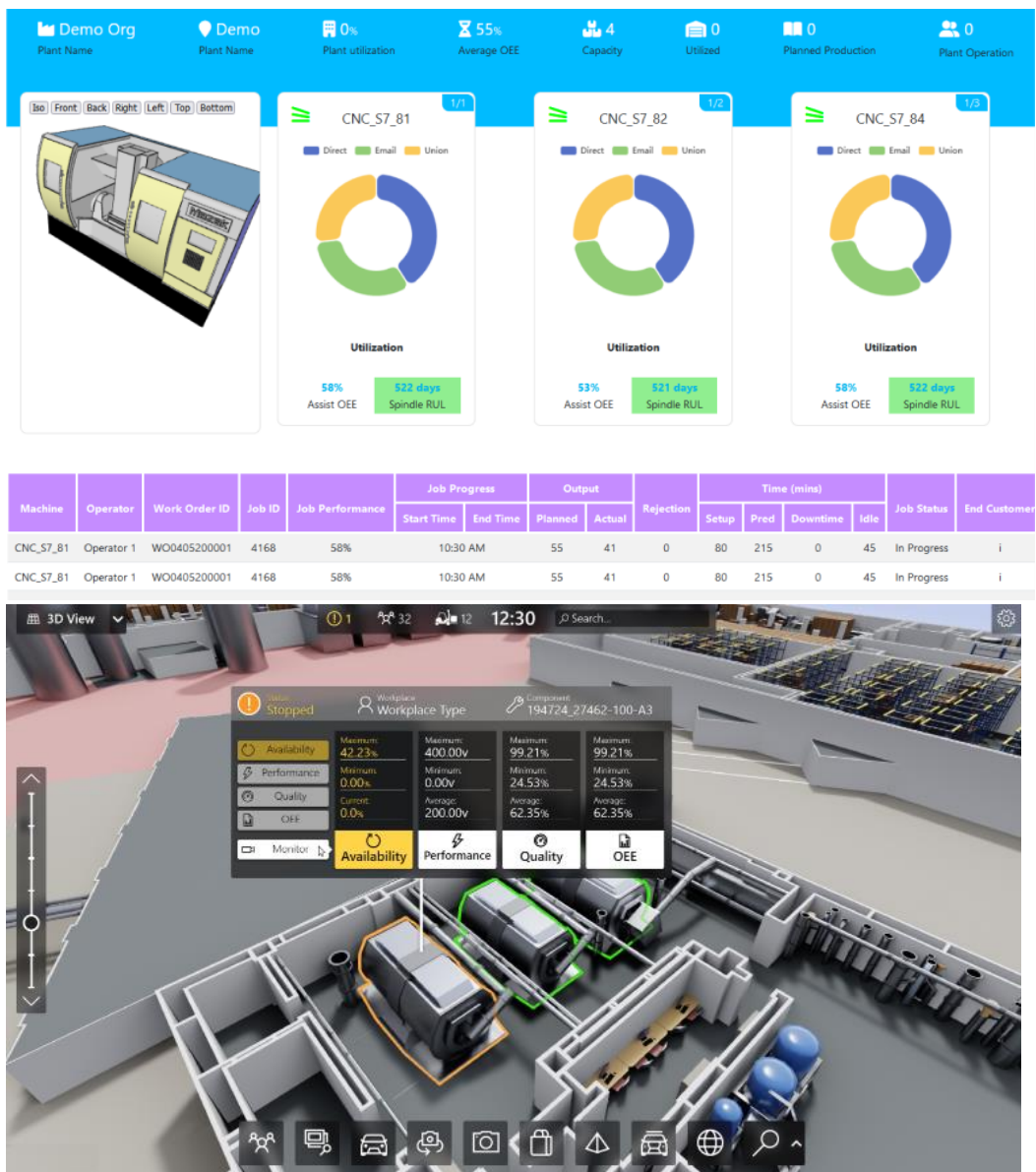
iii. **LoRaWAN** based Solution

UCT is one of the early adopters of LoRAWAN teschnology and providing solution in Agritech, Smart cities, Industrial Monitoring, Smart Street Light, Smart Water/ Gas/ Electricity metering solutions etc.

iv. Predictive Maintenance

UCT is providing Industrial Machine health monitoring and Predictive maintenance solution leveraging Embedded system, Industrial IoT and Machine Learning Technologies by finding Remaining useful life time of various Machines used in production process.
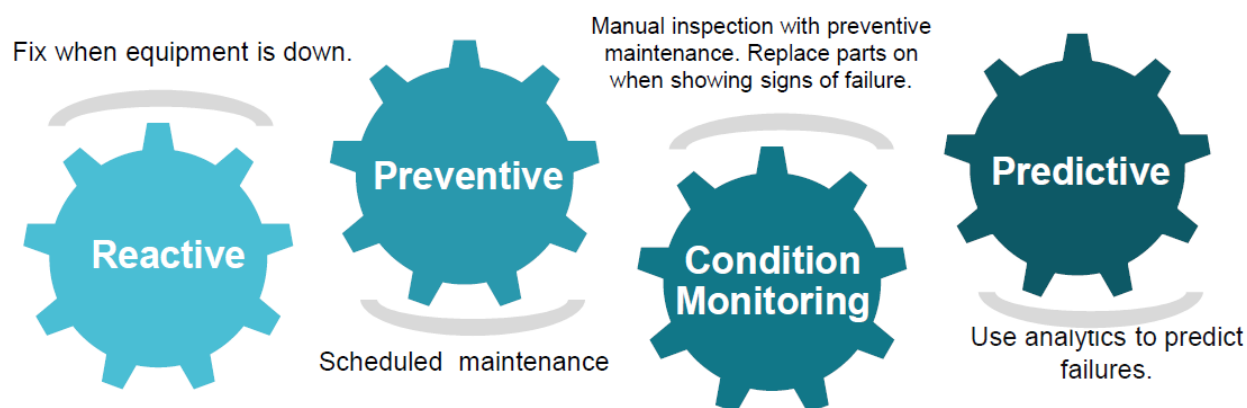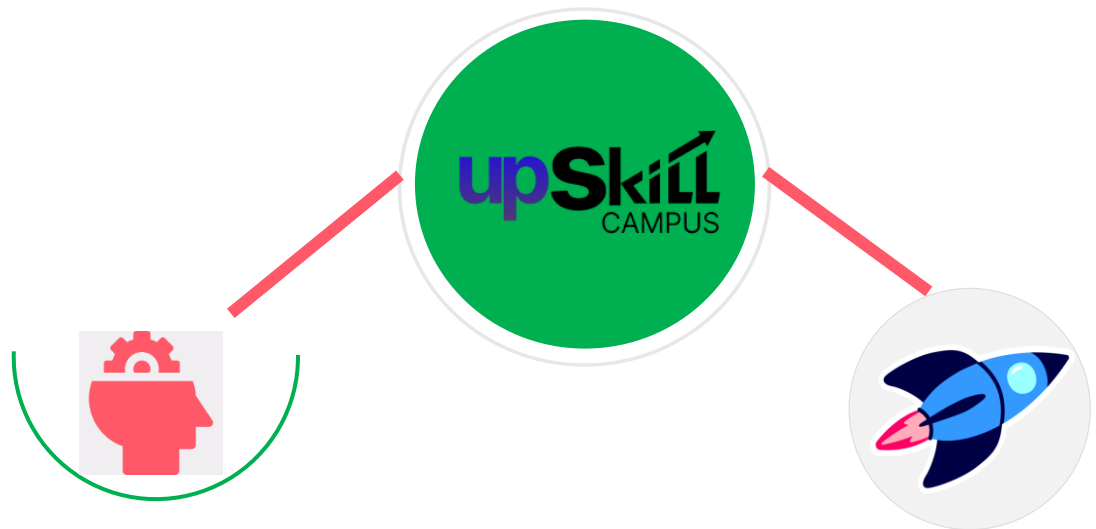

2.2 About upskill Campus (USC)

upskill Campus along with The IoT Academy and in association with Uniconverge technologies has facilitated the smooth execution of the complete internship process.

USC is a career development platform that delivers **personalized executive coaching** in a more affordable, scalable and measurable way.
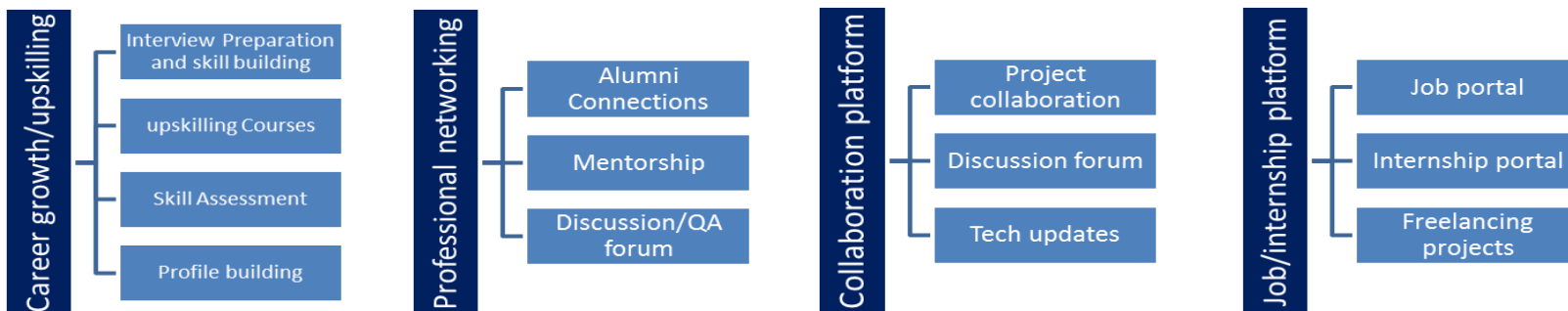
Seeing need of upskilling in self paced manner along-with additional support services e.g. Internship, projects, interaction with Industry experts, Career growth Services

upSkill Campus aiming to upskill 1 million learners in next 5 year

https://www.upskillcampus.com/

Career growth/upskilling
- Interview Preparation and skill building
- upskilling Courses
- Skill Assessment
- Profile building

Professional networking
- Alumni Connections
- Mentorship
- Discussion/QA forum

Collaboration platform
- Project collaboration
- Discussion forum
- Tech updates

Job/internship platform
- Job portal
- Internship portal
- Freelancing projects

2.3 The IoT Academy

The IoT academy is EdTech Division of UCT that is running long executive certification programs in collaboration with EICT Academy, IITK, IITR and IITG in multiple domains.

2.4 Objectives of this Internship program

The objective for this internship program was to

☛ get practical experience of working in the industry.

☛ to solve real world problems.

☛ to have improved job prospects.

☛ to have Improved understanding of our field and its applications.

☛ to have Personal growth like better communication and problem solving.

3 Problem Statement

In the assigned problem statement

To generate a password as per the user choice. This involved writing Python code to generate secure and random passwords, considering factors like length, complexity, and user preferences.

4   Existing and Proposed solution

In the existing Solution of the program the user cannot make use of special character in their password to make it stronger

In my proposed solution I have divided it into two parts easy and hard and make use of list.

----

import random

import os

letters = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z', 'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z']

numbers = ['0', '1', '2', '3', '4', '5', '6', '7', '8', '9']

symbols = ['!', '#', '$', '%', '&', '(', ')', '*', '+']

print("Welcome to the PyPassword Generator!")

nr_letters= int(input("How many letters would you like in your password?\n"))

nr_symbols = int(input(f"How many symbols would you like?\n"))

nr_numbers = int(input(f"How many numbers would you like?\n"))

let=""

for character in range(nr_letters):

   let=let+random.choice(letters)

for character in range(nr_numbers):

   let=let+random.choice(numbers)

for character in range(nr_symbols):

   let=let+random.choice(symbols)

print("Easy Level : ",let)

```python
let1 = []

passw=""


for character in range(nr_letters):

    let1.append(random.choice(letters))

for character in range(nr_numbers):

    let1.append(random.choice(numbers))

for character in range(nr_symbols):

    let1.append(random.choice(symbols))

random.shuffle(let1)

for char in let1:

    passw=passw+char

print("Hard Level : ",passw)

os.system("pause")
```

4.1   Code submission (Github link)

https://github.com/samuelard7/upskill_campus/blob/ce01d518ff41264935898df5f92466d62d7eb59c/PassGenerator.py

4.2   Report submission (Github link) :

https://github.com/samuelard7/upskill_campus/blob/ce01d518ff41264935898df5f92466d62d7eb59c/FinalReport.docx

5 Proposed Design/ Model

The program will ask:

```

How many letters would you like in your password?

```

How many symbols would you like?

```

How many numbers would you like?

```

The objective is to take the inputs from the user to these questions and then generate a random password.

Easy Version (Step 1)

Generate the password in sequence. If the user wants

* 4 letters

* 2 symbols and

* 3 numbers

then the password might look like this:


```

fgdx$*924

```

You can see that all the letters are together. All the symbols are together and all the numbers follow each other as well.

Hard Version (Step 2)

In the advanced version of this project the final password does not follow a pattern. So the example above might look like this:

```

x$d24g*f9

```

And every time you generate a password, the positions of the symbols, numbers, and letters are different.

6 Performance Test

Some of the constraints I encountered related to balancing the security with usability and optimizing the password generation algorithm.

7 My Learning

By the end of the internship, I successfully developed a functional and secure password generator, which can benefit users in creating strong and reliable passwords.

I have learned the importance of soft skills and the versatility of Python language libraries such as Pandas, NumPy and concepts like list comprehension which I quite enjoyed and zip function to form coordinates and learning about nested dictionary and list.

8 Future work scope

While the internship was a fruitful experience, there were certain ideas and areas of potential future work that couldn't be fully explored due to time constraints:

1. **Multi-platform Integration**: Expanding the password generator to work seamlessly on various platforms and devices, including mobile apps and browser extensions.

2. **Password Strength Analysis**: Incorporating a feature that analyzes the strength of existing passwords and provides recommendations for improving security.

3. **Cloud-Based Storage**: Developing a secure cloud-based storage solution for generated passwords, ensuring users can access their passwords from anywhere securely.

4. **User Authentication**: Integrating the password generator into a broader authentication system, enhancing the overall security of user accounts.

5. **User Education Resources**: Developing educational materials and resources on password security, along with tips on using the generator effectively.

6. **Machine Learning Integration**: Exploring the use of machine learning to predict evolving password attack patterns and adjust the generator's parameters accordingly.