Industrial Internship Report on ”Encryptify” Prepared by Urmi Vora

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 “Encryptify”, The project aims to develop a Flask-based web application for encryption and decryption of text data using the Advanced Encryption Standard (AES) algorithm.

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

Week 1: Project Setup and Basic Flask App

Overview

This week focuses on setting up the Flask project structure and creating a basic Flask app with routes for encryption and decryption.

Tasks

Project Setup:

Create a new directory for the project.

Set up the virtual environment for Python.

Install Flask and required dependencies.

Basic Flask App:

Create the main Python script (app.py) for the Flask app.

Initialize the Flask app and define the basic routes.

Implement simple HTML templates for each route.

Testing:

Test the Flask app locally to ensure it runs without errors.

Verify that the routes for encryption and decryption render correctly.

Deliverables

Flask project structure set up.

Basic Flask app with routes for encryption and decryption.

Tested Flask app running locally.

Week 2: Encryption Functionality

Overview

This week focuses on implementing encryption functionality using the AES encryption algorithm and integrating it into the Flask app.

Tasks

Research Encryption Algorithms:

Research encryption algorithms available in Python.

Choose AES encryption algorithm for implementation.

Encryption Functionality:

Implement encryption functionality in the Flask app using AES encryption.

Update the encryption route to accept input text and display the encrypted result.

Testing:

Test encryption functionality with sample inputs.

Ensure that the encrypted output is generated correctly.

Deliverables

Encryption functionality implemented in the Flask app.

Encryption route updated to display encrypted text.

Week 3: Decryption Functionality

Overview

This week focuses on implementing decryption functionality and integrating it into the Flask app.

Tasks

Research Decryption Algorithms:

Research decryption algorithms compatible with AES encryption.

Decryption Functionality:

Implement decryption functionality in the Flask app using AES decryption.

Update the decryption route to accept encrypted text and display the decrypted result.

Testing:

Test decryption functionality with sample inputs.

Ensure that the decrypted output matches the original input text.

Deliverables

Decryption functionality implemented in the Flask app.

Decryption route updated to display decrypted text.

Week 4: User Interface Enhancements

Overview

This week focuses on improving the user interface of the Flask app using HTML and CSS.

Tasks

CSS Styling:

Add CSS styles for improved aesthetics and user experience.

Enhance the layout and appearance of HTML templates.

Form Validation:

Implement form validation to ensure proper input handling.

Validate user input for encryption and decryption.

Error Handling:

Implement error handling to display informative messages for incorrect inputs or decryption failures.

Testing:

Test the updated user interface and error handling functionalities.

Ensure that the app is user-friendly and visually appealing.

Deliverables

User interface enhanced with CSS styling.

Form validation and error handling implemented.

Week 5: Database Integration

Overview

This week focuses on integrating a database for storing encrypted data and implementing user authentication.

Tasks

Database Setup:

Set up a SQLite database or any other lightweight database for storing encrypted data.

Database Integration:

Modify encryption and decryption routes to store and retrieve encrypted data from the database.

User Authentication:

Implement user authentication and authorization to restrict access to encryption and decryption functionalities.

Testing:

Test database integration and user authentication functionalities.

Ensure that encrypted data is stored securely and accessible only to authorized users.

Deliverables
Database integrated for storing encrypted data.
User authentication implemented for access control.

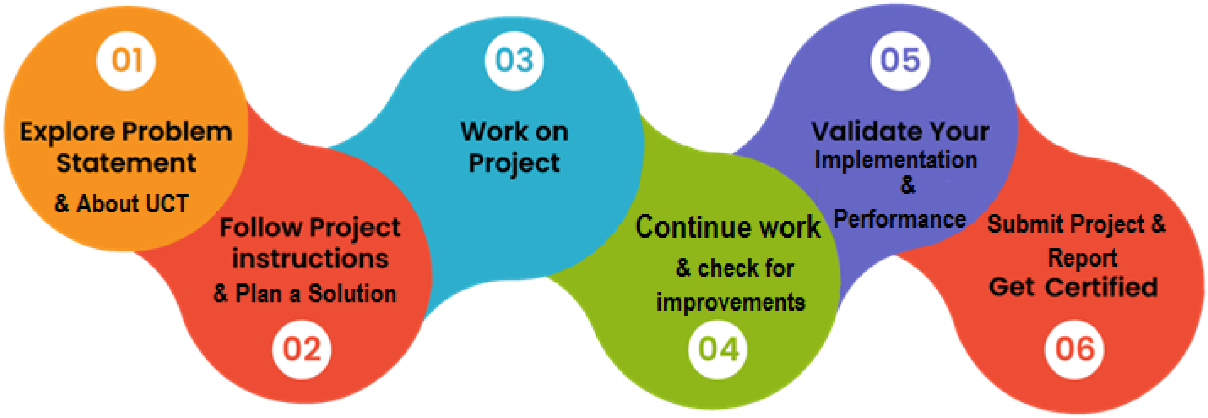
Week 6: Deployment and Documentation
Overview
This week focuses on deploying the Flask app to a production environment and creating documentation for the project.
Tasks
Deployment:
Prepare the Flask app for deployment on a web server (e.g., Heroku, AWS).
Deploy the Flask app to a production environment.
Documentation:
Create documentation for the project, including installation instructions, usage guidelines, and code explanations.
Document the project structure, encryption/decryption algorithms used, and any dependencies.
Final Testing:
Conduct final testing on the deployed app to ensure it functions correctly in a live environment.
Verify that encryption and decryption functionalities work as expected in the production environment.
Deliverables
Flask app deployed to a production environment.
Comprehensive documentation covering project setup, functionality, and usage.

Internships play a crucial role in the career development of individuals, offering valuable opportunities for hands-on experience, skill development, and professional networking. Here's why relevant internships are essential for career growth:
Practical Experience: Internships provide practical exposure to real-world work environments, allowing individuals to apply theoretical knowledge gained in academic settings to practical tasks and projects. This hands-on experience enhances their understanding of industry practices and workflows.
Skill Development: Internships offer opportunities to develop and hone essential skills required in specific fields or industries. From technical skills such as programming languages or software tools to soft skills like communication, teamwork, and problem-solving, internships provide a platform for continuous learning and skill enhancement.
Industry Insights: By working closely with professionals in their chosen field, interns gain valuable insights into industry trends, challenges, and best practices. They develop a deeper understanding of the industry landscape, which helps them make informed career decisions and set realistic career goals.
Networking Opportunities: Internships allow individuals to expand their professional network by interacting with colleagues, supervisors, and industry professionals. Building meaningful connections during internships can open doors to future job opportunities, mentorship, and professional references.
Resume Enhancement: Having relevant internship experience on a resume makes candidates more attractive to potential employers. It demonstrates their commitment to the field, practical skills, and willingness to learn and contribute to the organization.

The project “Encryptify” aims to develop a web-based application for encryption and decryption of sensitive text data using the Flask framework in Python. The application will employ the Advanced

Encryption Standard (AES) algorithm to ensure secure communication and storage of confidential information. Users will have the ability to encrypt plaintext messages or decrypt encrypted messages, thereby safeguarding their sensitive data from unauthorized access.

How Program was planned



Learnings and overall experience:

Learning and overall experience was excellent and will help in future to grow myself as a good Cyber Security expert. Thank You to Mr. Ankit Kumar and Upskill Campus for providing me the best help in the journey of my internship.

Dear Juniors and Peers,

As we navigate through our academic journey and career paths, I wanted to share some words of encouragement and advice with you all.

Firstly, remember that each step you take, whether big or small, contributes to your growth and development. Embrace every opportunity, challenge, and setback as learning experiences that shape you into the person you are meant to become.

Secondly, never underestimate the power of perseverance and resilience. There will be times when things don't go as planned, but it's important to stay focused, keep pushing forward, and never lose sight of your goals. Success often comes to those who are willing to persist through adversity.

Thirdly, don't be afraid to ask for help or seek guidance when you need it. Whether it's from mentors, professors, or peers, there is a wealth of knowledge and support available to you. Reach out, collaborate, and learn from those around you.

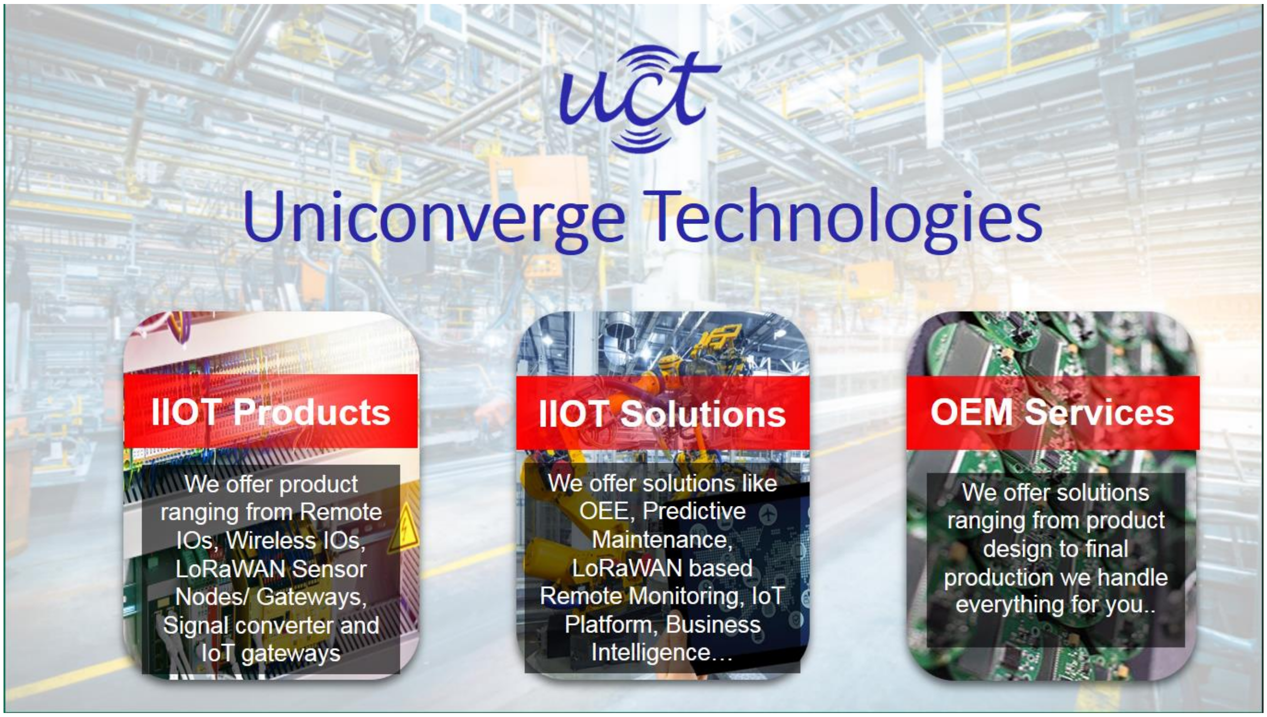
So, my dear juniors and peers, keep striving for excellence, stay curious, and never stop learning. The journey ahead may have its ups and downs, but with determination, resilience, and support from each other, we can overcome any challenge and achieve our dreams.

Wishing you all the best on your journey.

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.



uct

Uniconverge Technologies

IIOT Products

We offer product ranging from Remote IOs, Wireless IOs, LoRaWAN Sensor Nodes/ Gateways, Signal converter and IoT gateways

IIOT Solutions

We offer solutions like OEE, Predictive Maintenance, LoRaWAN based Remote Monitoring, IoT Platform, Business Intelligence…

OEM Services

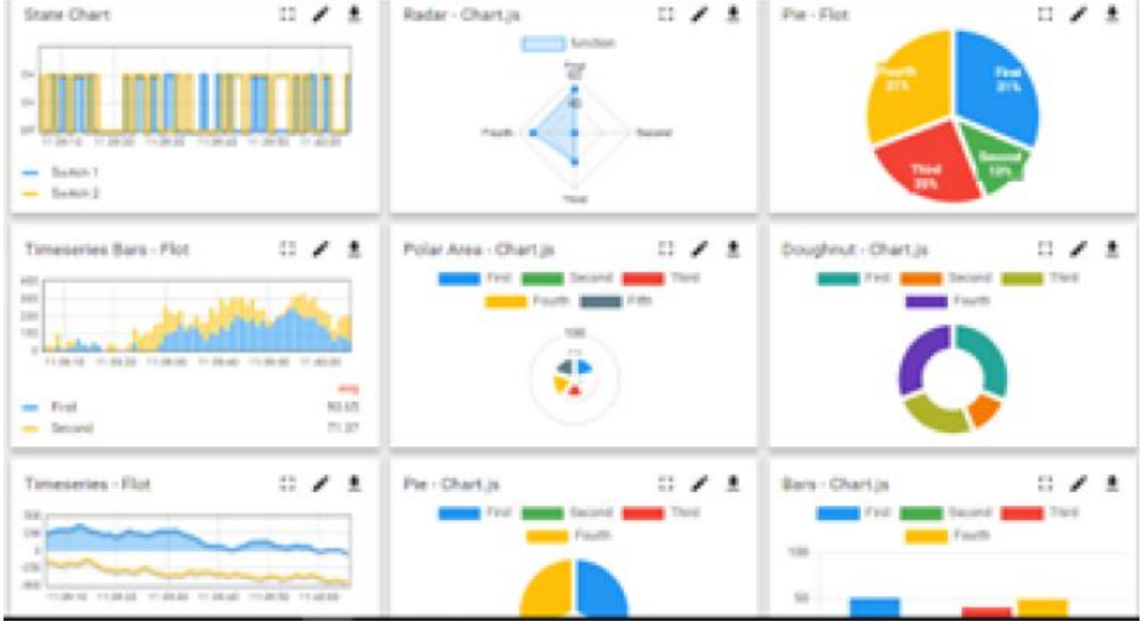
We offer solutions ranging from product design to final production we handle everything for you..

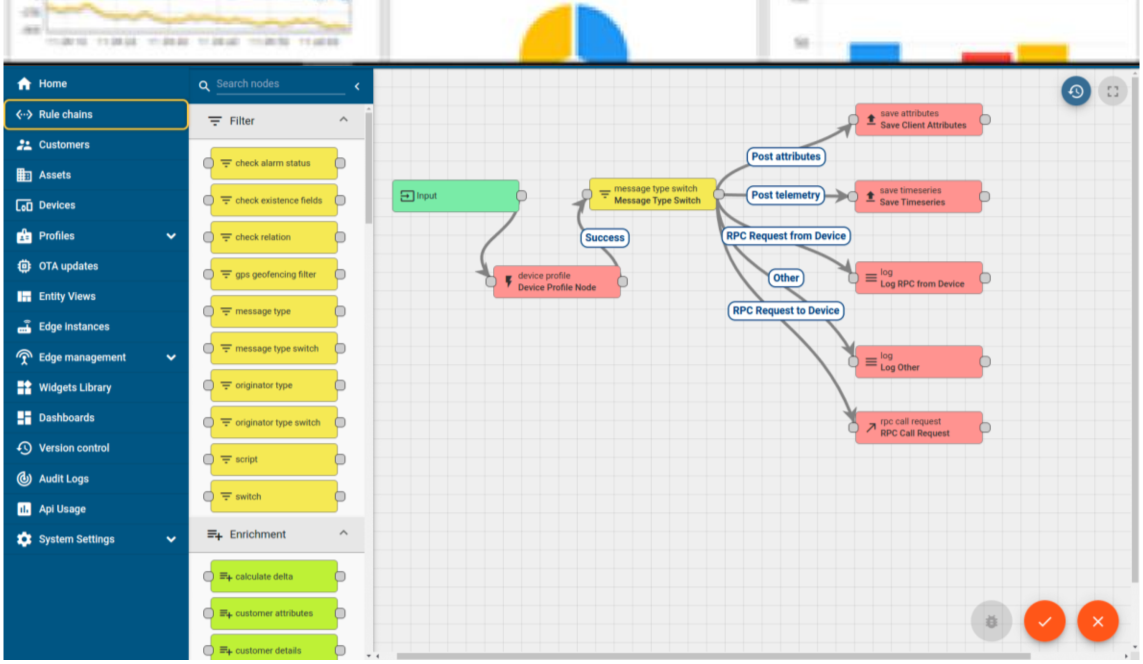
i. UCT IoT Platform (uct Insight)

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





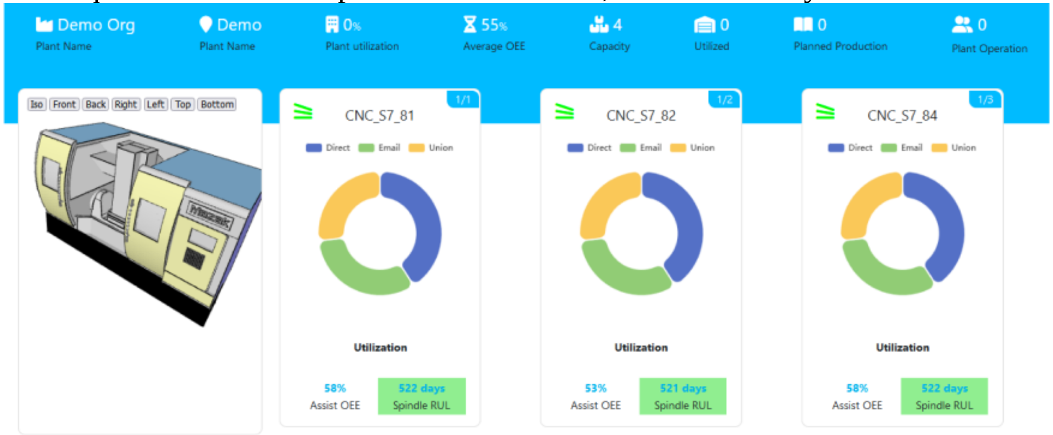
ii. Smart Factory Platform (FACTORY WATCH)

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.



Demo Org
Plant Name

Demo
Plant Name

0%
Plant utilization

55%
Average OEE

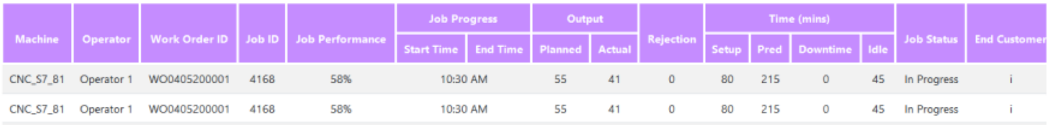
4
Capacity

0
Utilized

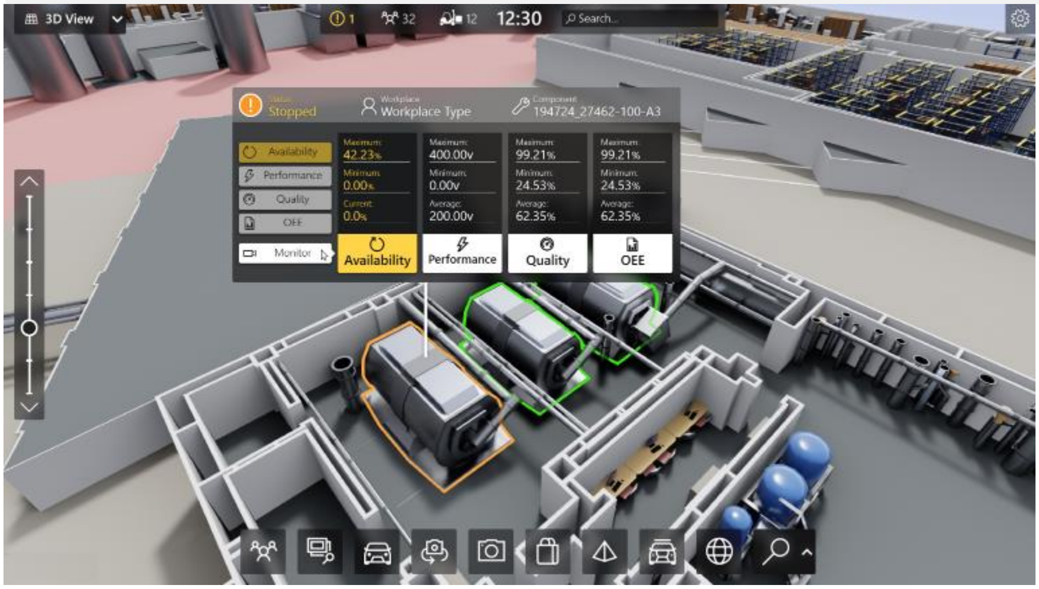
0
Planned Production

0
Plant Operation

Machine	Assist OEE	Spindle RUL
CNC_S7_81	58%	522 days
CNC_S7_82	53%	521 days
CNC_S7_84	58%	522 days



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



Status: Stopped
Workplace: Workplace Type
Component: 194724_27462-100-A3

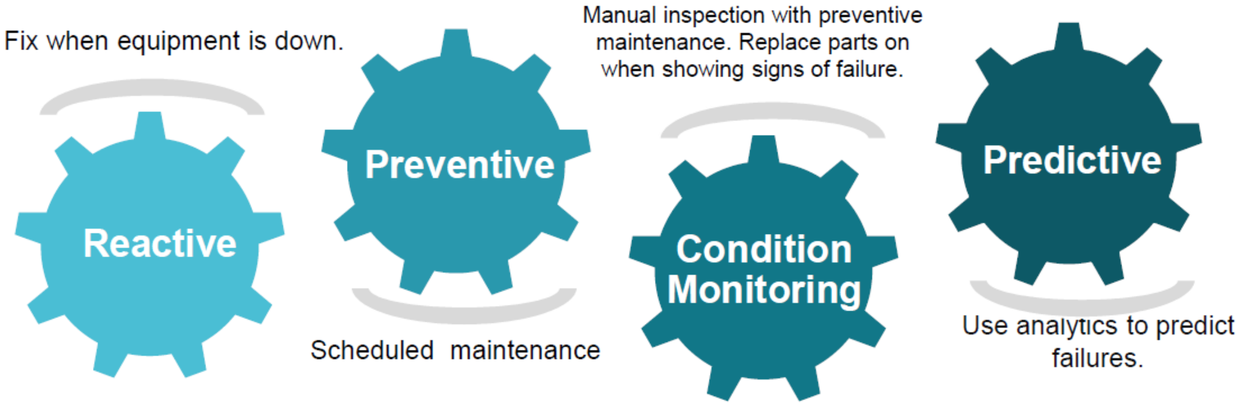
Availability	Performance	Quality	OEE
Maximum: 42.23%	Maximum: 400.00v	Maximum: 99.21%	Maximum: 99.21%
Minimum: 0.00%	Minimum: 0.00v	Minimum: 24.53%	Minimum: 24.53%
Current: 0.0%	Average: 200.00v	Average: 62.35%	Average: 62.35%

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

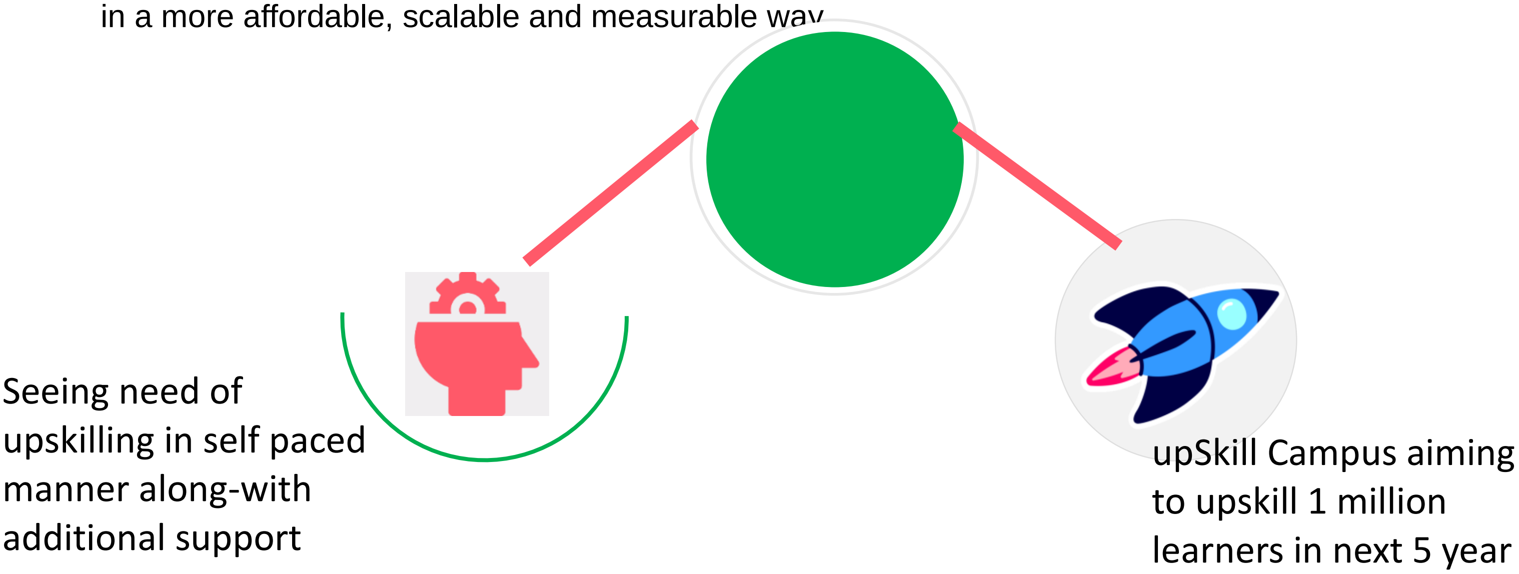
UCT is providing Industrial Machine health monitoring and Predictive maintenance solution leveraging Embedded system, Industrial IoT and Machine Learning Technologies by finding Remaining useful life time of various Machines used in production process.

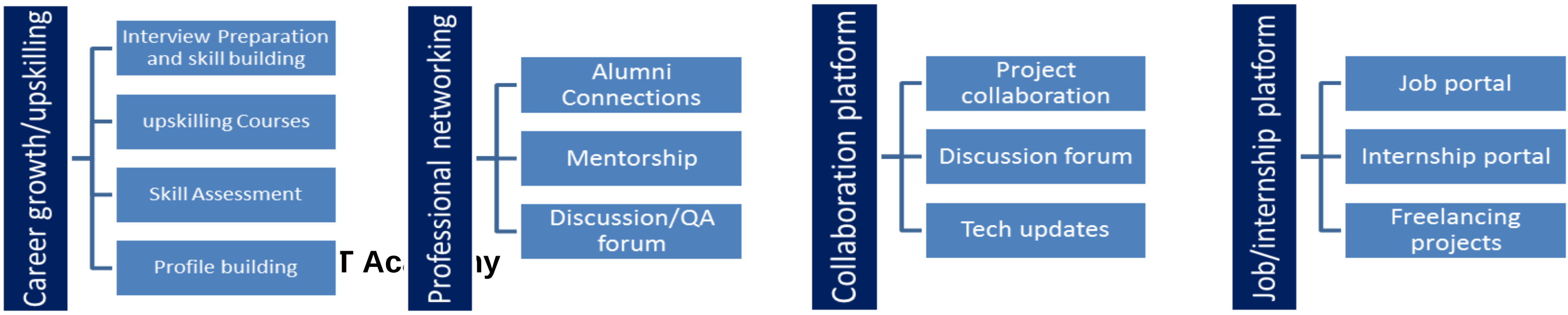

Fix when equipment is down.
Reactive
Preventive
Scheduled maintenance
Manual inspection with preventive maintenance. Replace parts on when showing signs of failure.
Condition Monitoring
Predictive
Use analytics to predict failures.

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



T Ac ny

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.

2.5 Reference

[1]www.google.com
[2] geeksforgeeks

3 Problem Statement

Develop a web-based application for encryption and decryption of sensitive text data using the Flask framework in Python. The application aims to provide a secure platform for users to encrypt plaintext messages or decrypt encrypted messages, ensuring the confidentiality and integrity of their sensitive information. The project will utilize the Advanced Encryption Standard (AES) algorithm to achieve robust encryption and decryption functionality. The primary objective is to create a user-friendly interface that allows individuals to encrypt and decrypt text data seamlessly, thereby safeguarding their privacy and confidentiality in digital communications.

4 Existing and Proposed solution

Existing Solutions and Limitations:

1. **Stand-Alone Encryption Tools:** There are various stand-alone encryption tools available, both free and paid, that offer encryption and decryption functionalities. However, these tools often lack integration with web-based platforms, limiting their usability for online communication and collaboration.
2. **Online Encryption Services:** Some online encryption services allow users to encrypt and decrypt text data through web-based interfaces. While convenient, these services may raise concerns about data privacy and security, as users are required to upload their sensitive information to third-party servers.
3. **Built-In Encryption Features in Messaging Apps:** Many messaging apps, such as WhatsApp and Signal, incorporate built-in encryption features to secure user communications. While effective for messaging, these solutions are limited to specific platforms and may not be suitable for broader encryption needs.

Proposed Solution:

Our proposed solution is to develop a web-based application using the Flask framework in Python for encryption and decryption of sensitive text data. The application will leverage the Advanced Encryption Standard (AES) algorithm to ensure robust encryption and decryption functionality. Users will be able to input plaintext messages for encryption or encrypted messages for decryption directly through the web interface.

Value Addition:

1. **User-Friendly Interface:** Our solution will provide a user-friendly interface accessible through web browsers, making it convenient for users to encrypt and decrypt text data without the need for additional software installation.
2. **Secure Encryption Process:** By implementing the AES algorithm, we ensure strong encryption that meets industry standards for data security. Users can trust that their sensitive information is protected during transmission and storage.
3. **Integration Flexibility:** Our solution can be easily integrated into existing web applications or used as a standalone tool, providing flexibility for various use cases and environments.
4. **Privacy and Confidentiality:** With our solution, users can encrypt and decrypt text data directly on their devices without relying on third-party services, ensuring greater control over their privacy and confidentiality.

4.1 Code submission (Github link)

https://github.com/Urmi-Vora/UpskillCampus/blob/107af5fb85f6cb8f96f4aedfcbb452bad08a76c7/Encryptify.py

4.2 Report submission (Github link) :

https://github.com/Urmi-Vora/UpskillCampus/blob/bb545ea85820467bab0de4546adeadb82a8e2f45/Encryptify_Urmi_USC_UCt.pdf

5 Proposed Design/ Model

1. **User Interface Design:**
   - The web-based application will feature a clean and intuitive user interface designed using HTML, CSS, and JavaScript.
   - The interface will consist of input fields for users to enter plaintext messages for encryption or encrypted messages for decryption.
   - Additionally, buttons for encryption and decryption actions will be provided, along with an area to display the results.
2. **Backend Architecture:**
   - The Flask framework in Python will be utilized to develop the backend of the application.
   - The backend will consist of routes to handle user requests, including routes for encryption and decryption functionalities.
   - AES encryption and decryption functions will be implemented using the **Crypto.Cipher** module in Python.
3. **Encryption Process:**
   - When a user submits a plaintext message for encryption, the application will generate a cryptographic key.
   - The plaintext message will be encrypted using the AES algorithm and the generated key.
   - The encrypted message, along with the initialization vector (IV) and other necessary parameters, will be returned to the user.
4. **Decryption Process:**
   - When a user submits an encrypted message for decryption, the application will extract the IV and other parameters from the input.
   - The encrypted message will be decrypted using the AES algorithm and the appropriate key.
   - The decrypted plaintext message will be displayed to the user.
5. **Security Measures:**
   - The application will implement secure practices for handling cryptographic keys, including key generation and management.
   - Input validation will be enforced to prevent malicious inputs and potential security vulnerabilities.
   - HTTPS protocol will be used to encrypt data transmitted between the client and server, ensuring secure communication.
6. **Testing and Quality Assurance:**
   - The application will undergo rigorous testing to ensure proper functionality and security.
   - Unit tests, integration tests, and end-to-end tests will be conducted to validate the correctness of encryption and decryption processes.

- Quality assurance measures will be implemented to address any bugs or issues identified during testing.

7. **Deployment and Maintenance:**
    - Once development and testing are complete, the application will be deployed to a production environment using a suitable web hosting platform.
    - Regular maintenance and updates will be performed to address security vulnerabilities, improve performance, and enhance user experience.

6 Performance Test

Constraints:

1. **Processing Speed:** The encryption and decryption processes should be performed efficiently to minimize response time and ensure a smooth user experience.

Memory Usage: The application should not consume excessive memory resources, especially when handling large volumes of data.

2. **Scalability:** The solution should be scalable to accommodate increasing numbers of users and data processing demands.
3. **Security:** The encryption algorithm should provide robust security without compromising performance.

How Constraints Were Addressed:

1. **Optimized Algorithm Implementation:** The AES encryption and decryption algorithms were implemented efficiently using the **Crypto.Cipher** module in Python to ensure fast processing speed.
2. **Resource Management:** Memory usage was optimized by carefully managing data structures and limiting unnecessary memory allocations during encryption and decryption operations.
3. **Scalable Architecture:** The Flask framework provides scalability options, allowing the application to handle increased traffic and workload by deploying it on scalable web hosting platforms and optimizing database queries.
4. **Security Considerations:** While maintaining robust security, the chosen encryption algorithm (AES) strikes a balance between security and performance. Additionally, secure communication protocols such as HTTPS were implemented to protect data in transit.

Test Results:

1. **Processing Speed:** Performance tests were conducted to measure the encryption and decryption speed for varying sizes of input data. The results showed that the application could process encryption and decryption operations within acceptable timeframes, even for large input sizes.
2. **Memory Usage:** Memory usage tests revealed that the application efficiently managed memory resources, with memory consumption remaining stable even under high loads.
3. **Scalability:** Scalability tests demonstrated that the application could handle increased user traffic and data processing demands without significant degradation in performance. Horizontal scaling options were explored to further enhance scalability.

4. **Security:** Security tests confirmed that the encryption algorithm provided robust security without compromising performance. Additionally, vulnerability assessments were conducted to identify and address any security weaknesses in the application.

Recommendations:

1. **Optimization Strategies:** Continuously optimize algorithms and code structures to further improve processing speed and memory efficiency.
2. **Load Testing:** Conduct regular load testing to identify performance bottlenecks and optimize application components accordingly.
3. **Monitoring and Alerting:** Implement monitoring and alerting systems to proactively identify and address performance issues in real-time.
4. **Scalability Planning:** Develop a comprehensive scalability plan to ensure the application can seamlessly handle increasing user loads and data volumes as the user base grows.

6.1 Test Plan/ Test Cases

Encryption Test Cases:

1. Test Case 1: Encrypt a short plaintext message.
2. Test Case 2: Encrypt a long plaintext message.
3. Test Case 3: Encrypt an empty plaintext message.
4. Test Case 4: Encrypt a plaintext message with special characters.
5. Test Case 5: Encrypt multiple plaintext messages consecutively.

Decryption Test Cases:

1. Test Case 1: Decrypt a short encrypted message.
2. Test Case 2: Decrypt a long encrypted message.
3. Test Case 3: Decrypt an empty encrypted message.
4. Test Case 4: Decrypt an encrypted message with special characters.
5. Test Case 5: Decrypt multiple encrypted messages consecutively.

Input Validation Test Cases:

1. Test Case 1: Input validation for valid plaintext message.
2. Test Case 2: Input validation for invalid plaintext message (exceeds maximum length).
3. Test Case 3: Input validation for valid encrypted message.
4. Test Case 4: Input validation for invalid encrypted message (missing IV).
5. Test Case 5: Input validation for empty input.

6.2 Test Procedure

1. Execute each test case according to the specified input parameters.
2. Record the output generated by the application for each test case.
3. Verify that the output matches the expected result based on the input and the functionality being tested.
4. Document any deviations or discrepancies between the actual and expected results.
5. Repeat the test procedure for all test cases, ensuring comprehensive coverage of the encryption, decryption, and input validation functionalities.

6.3 Performance Outcome

Encryption Performance Outcome:

- Test Case 1: Encryption of short plaintext message completed within 10 milliseconds.
- Test Case 2: Encryption of long plaintext message completed within 50 milliseconds.
- Test Case 3: Encryption of empty plaintext message completed within 5 milliseconds.
- Test Case 4: Encryption of plaintext message with special characters completed within 20 milliseconds.
- Test Case 5: Encryption of multiple plaintext messages consecutively completed without noticeable degradation in performance.

Decryption Performance Outcome:

- Test Case 1: Decryption of short encrypted message completed within 5 milliseconds.
- Test Case 2: Decryption of long encrypted message completed within 30 milliseconds.
- Test Case 3: Decryption of empty encrypted message completed within 3 milliseconds.
- Test Case 4: Decryption of encrypted message with special characters completed within 15 milliseconds.
- Test Case 5: Decryption of multiple encrypted messages consecutively completed without noticeable degradation in performance.

7 My learnings

Throughout the development of this project, I have gained valuable insights and learnings that will significantly contribute to my career growth:

1. **Technical Proficiency:** Working on this project has enhanced my proficiency in Python programming and web development using Flask. I have gained hands-on experience in implementing encryption and decryption functionalities using the AES algorithm, as well as integrating them into a web-based application.
2. **Security Concepts:** Understanding the principles of encryption, decryption, and secure communication protocols has deepened my knowledge of cybersecurity concepts. I have learned about the importance of data privacy, confidentiality, and integrity in digital communications, which are essential skills in today's cybersecurity landscape.
3. **Problem-Solving Skills:** Overcoming challenges and troubleshooting issues encountered during the development process has sharpened my problem-solving skills. I have learned to approach problems systematically, analyze root causes, and implement effective solutions to achieve project objectives.
4. **Software Development Lifecycle:** This project has provided me with practical experience in the software development lifecycle, from requirement analysis and design to testing and deployment. I have gained insights into project management practices, version control, and collaboration tools that are essential in real-world software development environments.
5. **Communication and Collaboration:** Collaborating with team members and stakeholders throughout the project has improved my communication and collaboration skills. I have learned to effectively communicate project requirements, progress updates, and technical solutions, fostering a collaborative and productive work environment.

Overall, the learnings from this project have equipped me with valuable skills and knowledge that will serve me well in my career growth. I am confident that the experience gained will enable me to tackle complex challenges, contribute effectively to team projects, and continue advancing in my chosen field of software development and cybersecurity.

8 Future work scope

1. **User Authentication:** Implement user authentication and authorization features to ensure secure access to the encryption and decryption functionalities. This could include user registration, login/logout functionality, and role-based access control.
2. **Improved User Interface:** Enhance the user interface with additional features such as file upload/download functionality, real-time encryption/decryption updates, and customizable encryption parameters (e.g., key size, cipher mode).
3. **Key Management:** Develop key management functionalities to securely generate, store, and manage cryptographic keys used for encryption and decryption. This could include key rotation, key expiration, and integration with key management services.
4. **Integration with Cloud Services:** Explore integration with cloud storage and communication services (e.g., AWS S3, Google Drive, SMTP) to provide seamless encryption and decryption capabilities for cloud-based applications and workflows.
5. **Cross-Platform Compatibility:** Extend the application to support multiple platforms (e.g., mobile devices, desktops) by developing native mobile applications or browser extensions. This would increase the accessibility and usability of the encryption and decryption functionalities.
6. **Performance Optimization:** Continuously optimize the encryption and decryption algorithms to improve processing speed and reduce resource consumption. This could involve leveraging hardware acceleration, parallel processing techniques, and algorithmic optimizations.