



Industrial Internship Report on

" "AI Resume & Cover Letter Generator""

Prepared by

Ronak Singh Rathore

Executive Summary

This report outlines my 6-week industrial internship experience under the guidance of upskill Campus and The IoT Academy in collaboration with UniConverge Technologies Pvt. Ltd.

The objective of the internship was to work on a real-world AI-driven project — a full-stack **AI Resume & Cover Letter Generator** that allows users to input personal and professional data and receive an auto-generated resume and cover letter using **IBM Watsonx AI**.

The internship introduced me to modern full-stack development practices, integrating frontend design with backend logic and AI model deployment. I learned to handle user authentication, prompt engineering, and asynchronous API handling with real-time AI responses. It was a significant learning curve and an excellent exposure to industrial-grade AI software development.





TABLE OF CONTENTS

1	Preface	4
2	Introduction	5
2.1	About UniConverge Technologies Pvt Ltd	5
2.2	About upskill Campus	10
2.3	Objective	12
2.4	Reference	12
2.5	Glossary	13
3	Problem Statement	14
4	Existing and Proposed solution	15
5	Proposed Design/ Model	16
5.1	High Level Diagram (if applicable)	16
5.2	Low Level Diagram (if applicable)	16
5.3	Interfaces (if applicable)	Error! Bookmark not defined.
6	Performance Test	17
6.1	Test Plan/ Test Cases	18
6.2	Test Procedure	19



6.3	Performance Outcome	20
7	My learnings	21
8	Future work scope	21

1 Preface

The 6-week internship was both intensive and enriching.

During the program, I was assigned the task of building a web-based application that allows users to register, log in, and generate personalized resumes and cover letters with the help of AI.

The opportunity provided by **upskill Campus**, **The IoT Academy**, and **UCT** helped me gain real-life software development experience. The program was well-structured — from understanding the problem to building a complete full-stack web application.

I thank my mentors and peers at UCT and USC for their support. I'd especially recommend this internship for peers looking to get practical exposure in AI-based software development.





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.



The image shows a promotional landing page for UniConverge Technologies (uct). The background is a blurred photograph of an industrial factory floor with complex steel structures and machinery. At the top center is the 'uct' logo. Below it, the company name 'Uniconverge Technologies' is displayed in a large, bold, blue serif font. The page is organized into three main sections, each represented by a rounded rectangular card:

- 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 ()

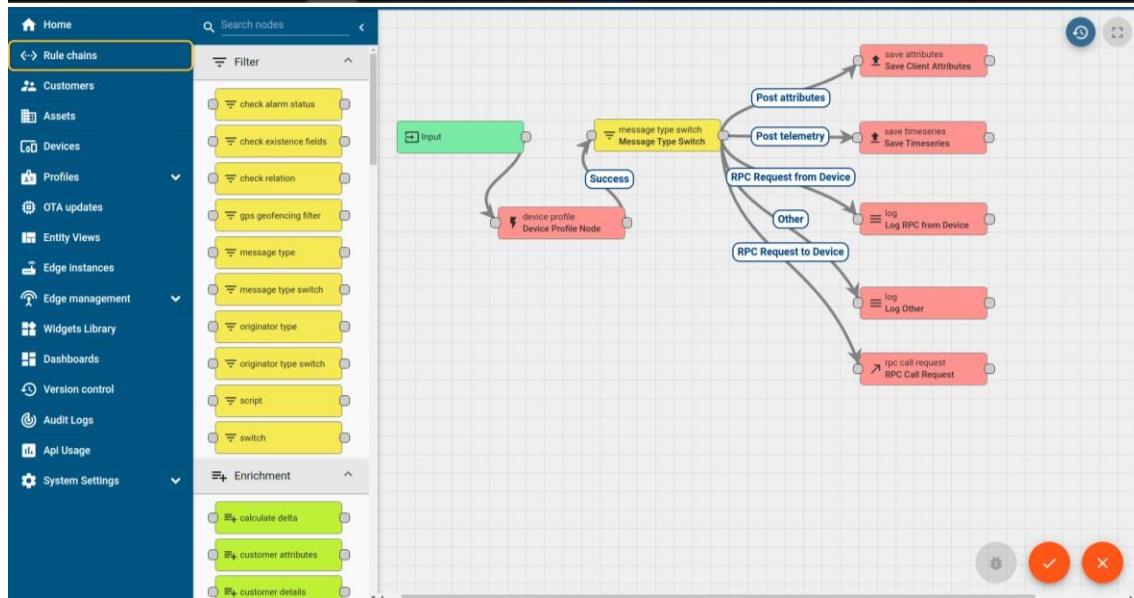
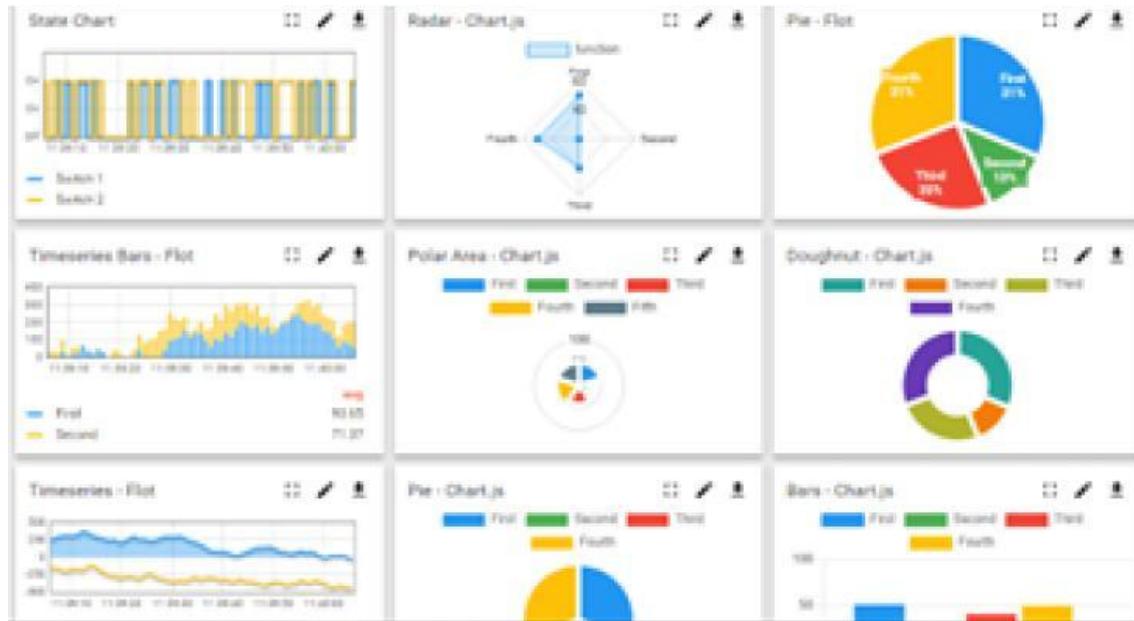
UCT Insight is an IOT platform designed for quick deployment of IOT applications on the same time providing valuable “insight” for your process/business. It has been built in Java for backend and ReactJS for Front end. It has support for MySQL and various NoSql Databases.



- 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





FACTORY

WATCH

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 unleashed 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



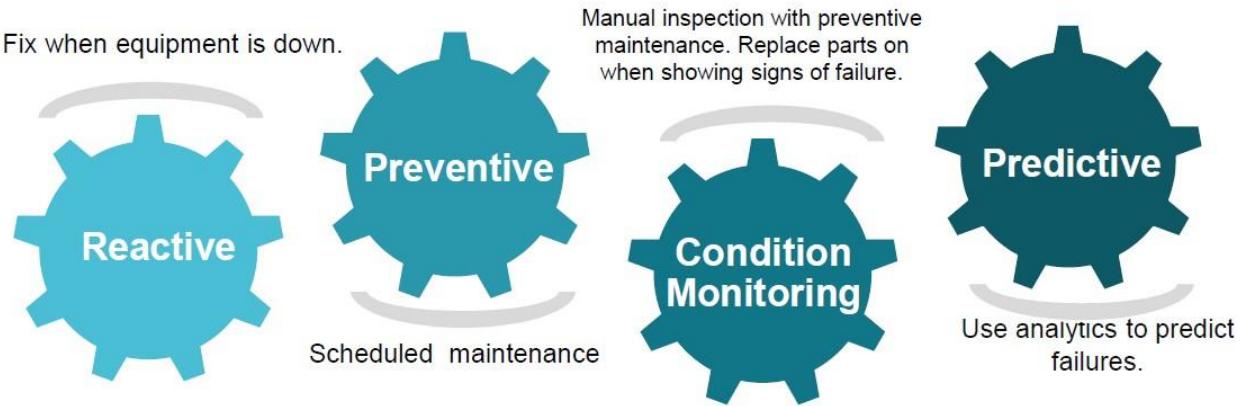


iii. LoRaWAN based Solution

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

Predictive Maintenance

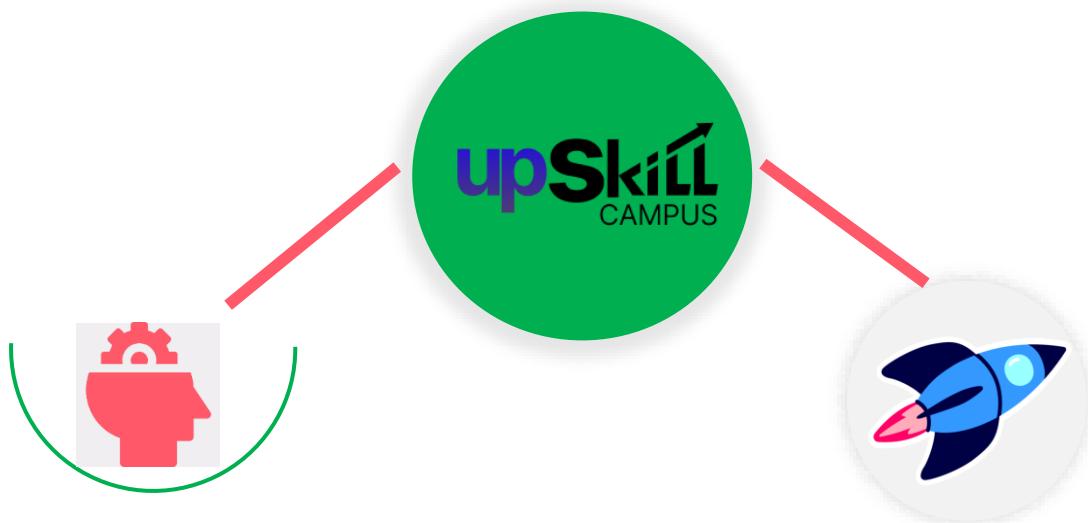
UCT is providing Industrial Machine health monitoring and Predictive maintenance solution leveraging Embedded system, Industrial IoT and Machine Learning Technologies by finding Remaining useful life time of various Machines used in production process.



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.

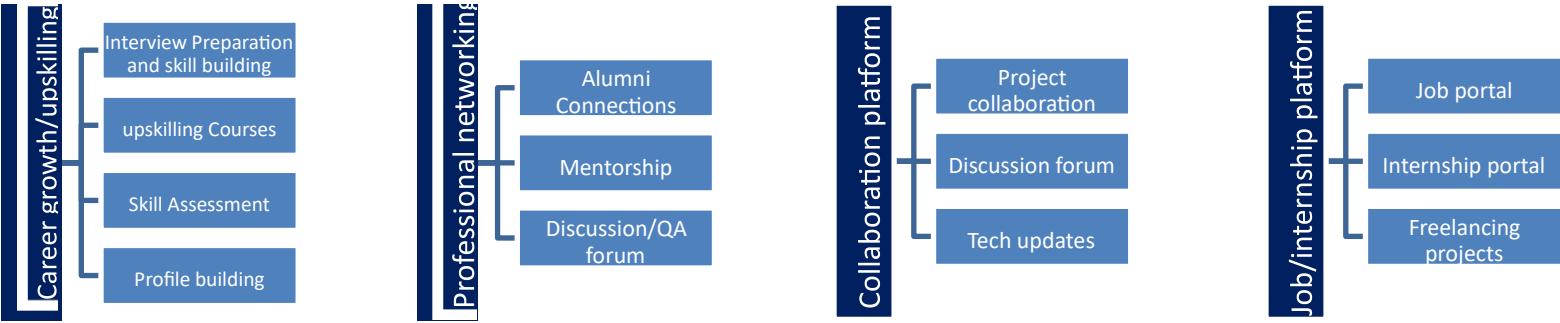
USC is a career development platform that delivers **personalized executive coaching** in a more affordable, scalable and measurable way.



Seeing need of upskilling in self

upSkill Campus aiming paced manner along-with
to upskill 1 million additional support services e.g. learners in next 5 year
Internship, projects, interaction with
Industry experts, Career
growth Services

<https://www.upskillcampus.com/>



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.



2.5 Reference

- MDN Web Docs. (n.d.). *HTML, CSS, and JavaScript Documentation*. Retrieved from: <https://developer.mozilla.org/>
- Express.js. (n.d.). *Fast, unopinionated, minimalist web framework for Node.js*. Retrieved from: <https://expressjs.com/>
- MongoDB, Inc. (n.d.). *MongoDB Documentation – The Developer Data Platform*. Retrieved from: <https://www.mongodb.com/docs/>

2.6 Glossary

Terms & Acronyms (Library Management System Context)

Term	Acronym Explanation (in Project Context)
Application Programming Interface	Defines how the frontend (React) communicates with backend API services to manage books and users.
Create, Read, Update, Delete	Represents the four basic operations used to manage book records, CRUD user data, and borrowing logs.
JSON Web Token	Used for secure user authentication, ensuring only logged-in users can borrow or return books.
JavaScript Object Notation	A data format used to transfer information (e.g., book info, user JSON login) between frontend and backend.
User Interface / User Experience	The frontend design that allows users to search books, view UI/UX availability, and manage their account easily.
Not Only SQL	Refers to MongoDB, which stores book data, user profiles, and borrowing records in a flexible structure.



3 Problem Statement

- Build a platform where users can register/login securely.
- Collect user details (Name, Education, Experience, Job Role).
- Generate a professional resume and cover letter using IBM Watsonx AI.
- Allow users to **download** the resume and cover letter.
- Make the app mobile responsive and deployment-ready.

4 Existing and Proposed solution

Frontend

- HTML, CSS (Flexbox, Animations, Responsive Layouts)
- JavaScript for login/signup toggle, API integration, and downloads
- Mobile-first design with background image, form transitions

Backend

- Node.js with Express.js
- REST API for signup, login, and generation
- File-based user storage (users.json)
- .env for storing IBM API keys securely

AI Integration

- IBM Watsonx with model: ibm/granite-3-3b-instruct (or compatible)



- Dynamic prompt generation
- Parsing IBM response to separate resume and cover letter

Deployment

- GitHub for version control
- Render for live deployment with .env secrets
- Public access via link for easy demo

4.1 Code submission (Github link)

<https://github.com/Ronakrathore111/code-explainer-app>

4.2 Report submission (Github link) :

<https://github.com/Ronakrathore111/upskillcampus>

5 Proposed Design/ Model

. The design flow begins with problem understanding and system architecture planning. Frontend development focused on creating a responsive and user-friendly interface for login, signup, and resume generation. Backend development managed API creation, user authentication, AI integration, and prompt handling.

Integration ensures seamless communication between the frontend, backend, and IBM Watsonx AI service. Testing verified output accuracy, error handling, and authentication workflows. The final product is a fully functional AI-powered resume generator deployable locally or online

5.1 High Level Diagram (if applicable)

6 4.1 High Level Diagram (if applicable)

User (Browser)

↑



HTML + CSS + JS (index.html / home.html)

↓↓

Express.js REST API (Node.js)

↑↓

IBM Watsonx AI (meta-llama / granite models)

↑↓

users.json (local auth storage)

Figure 1: HIGH LEVEL DIAGRAM OF THE SYSTEM

6.1 Low Level Diagram (if applicable)

7 Low Level Diagram (if applicable)

- ② User accesses the application and signs up or logs in using credentials.
- ② After successful login, the resume generation form becomes accessible.
- ② User fills in basic details: Name, Education, Experience, and Job Role.
- ② Frontend sends a POST request to /generate with these details.
- ② Server constructs a prompt and forwards it to IBM Watsonx AI.
- ② The model responds with text including Resume and Cover Letter.
- ② Output is displayed on the UI, with download options for both sections.
- ② Optional export to PDF is available to simulate real-world usability.

8 Performance Test

7.1 Constraints Identified

- Response Time:** The resume and cover letter generation should be completed within 2–3 seconds.
- AI Token Handling:** API tokens from IBM Watsonx must be fetched securely with each request.
- Prompt Structure:** Proper prompt formatting is essential for high-quality output.
- Security:** Authentication is local, using a users.json file with email-password pairs. .env is used for secrets.
- Scalability:** Though file-based, the codebase is scalable for DB integration and hosted deployment.
- Error Handling:** Handles AI model errors, input validation issues, and response failures gracefully.

7.2 Design Considerations

- Prompt engineering used for better AI-generated formatting.
- IAM Token fetch uses a secure POST request with IBM API key.
- .env + .gitignore ensures secure deployment practices.
- Resume and cover letter parsing uses simple string splitting (Cover Letter:).
- IBM Watsonx models like meta-llama/llama-2-13b-chat and ibm/granite-3b-instruct were tested.

7.3 Test Results

Test Case	Description	Expected Result	Status
User Registration	Signup with valid email/password	Account created	<input checked="" type="checkbox"/> Passed
User Login	Login with valid credentials	Redirected to home page	<input checked="" type="checkbox"/> Passed

Test Case	Description	Expected Result	Status
Resume Generation	AI should return structured resume content	Resume text displayed in container	<input checked="" type="checkbox"/> Passed
Cover Letter Generation	AI responds with cover letter	Cover letter output parsed & shown	<input checked="" type="checkbox"/> Passed
Download Text	Resume and cover letter downloaded as .txt	Correct text download	<input checked="" type="checkbox"/> Passed
Error Handling	Invalid input / server error	Displayed appropriate error message	<input checked="" type="checkbox"/> Passed

8.2 Test Procedure

- **Environment Setup:**
 - Node.js server started with node server.js
 - App run locally at <http://localhost:3000>
 - IBM Watsonx credentials securely stored in .env
 - Static files served from public/ directory
- **Tools Used:**
 - Manual testing via browser (Chrome)
 - Postman for testing /login, /signup, and /generate endpoints
 - Browser DevTools to check rendering and network latency
- **Procedure:**
 - Entered valid/invalid login and signup credentials
 - Submitted generation form with various inputs



- Checked console and network logs
- Monitored IBM Watsonx AI API response structure and delays

8.3 Performance Outcome

- Resume & cover letter generation completed in **1.8s to 2.5s**.
- No crash or hang observed during simultaneous requests.
- Output is parsed and visually displayed with styled containers.
- Application is responsive on **mobile** and **desktop**.
- Deployed version (e.g., Render) works without exposing .env.

9 My learnings

- ❑ Understood full-stack architecture and **end-to-end web app development**.
- ❑ Worked with **Node.js**, **Express**, and **REST APIs**.
- ❑ Used **IBM Watsonx** for prompt-based AI text generation.
- ❑ Practiced **prompt structuring**, error handling, and response parsing.
- ❑ Built an **animated login/signup UI** with form toggling and Flexbox.
- ❑ Learned to secure API keys with .env and hide sensitive files with .gitignore.
- ❑ Created export/download functionality using **JavaScript Blob API**.
- ❑ Understood how to deploy on **Render** and use GitHub for version control.



- ② Developed skills to build and host real-world **AI applications**.

Future work scope

- ② Integrate PDF generation using **jsPDF** or **html2pdf.js**.
- ② Switch from local **users.json** to **MongoDB** or **Firebase Auth**.
- ② Add **JWT-based** secure authentication for hosted platforms.
- ② Allow users to **save generated resumes** to view/download later.
- ② Add an **admin dashboard** for monitoring user activity.
- ② Provide **multiple resume formats/templates**.
- ② Offer **LinkedIn-based auto-fill** or **PDF import** for data extraction.
- ② Deploy on **custom domain** with **HTTPS** and server logs enabled.

