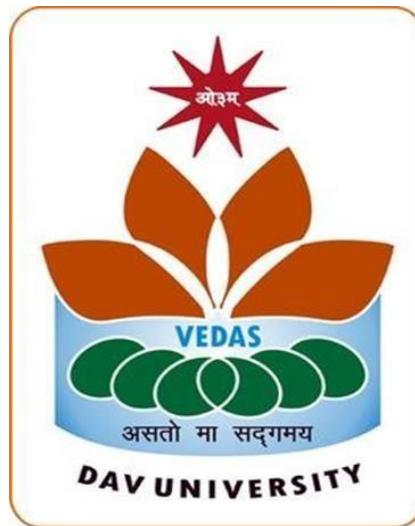


Six Weeks Industrial Training Project Report On

“COLD EMAIL GENERATOR”

Submitted in the partial fulfilment of the requirement for the award of degree
of
Bachelor of Technology
In
Computer Science and Engineering
Batch (2022-2026)



Submitted to :

Mrs. Bindu Bansal

Associate Professor (CSE)

Submitted by :

Abhishek Sandhu

12200208

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
DAV UNIVERSITY
JALANDHAR-PUNJAB 144012

ACKNOWLEDGEMENT

I express my gratitude to all those who helped us in various stages of the development of this project. First, I would like to express my sincere gratitude indebtedness to Dr. Rahul Hans (Head of Department of Computer Science and Engineering), Mrs. Bindu Bansal (Associate Professor) of DAV University for allowing me to undergo the summer training of 45 days at CodeTech IT Solution Private Limited.

I am also thankful to all faculty members of Department of Computer Science and Engineering, for their true help, inspiration and for helping me for the preparation of the final report and presentation.

Last but not least, I pay my sincere thanks and gratitude to all the Staff Members of CodeTech IT Solution for their support and for making our training valuable and fruitful.

DECLARATION

I, Abhishek Sandhu, hereby declare that the work which is being presented in this project/training titled “Cold Email Generator” by me, in partial fulfilment of the requirements for the award of Bachelor of Technology (B.Tech) Degree in “Computer Science and Engineering” is an authentic record of my own work carried out under the guidance of Mr. Rakesh kumar (Data Scientist)

To the best of my knowledge, the matter embodied in this report has not been submitted to any other University/ Institute for the award of any degree or diploma.

ABHISHEK SANDHU

(12200208)

CERTIFICATE



CODTECH IT SOLUTIONS PRIVATE LIMITED

8-7-7/2,PlotNO.51,Opp:NaveenaSchool,HasthinapuramCentral,Hyderabad
, 500 079. Telangana



CERTIFICATE OF INTERNSHIP EXPERIENCE

To whom so ever it may concern

This is to certify that **Abhishek sandhu** with Intern id: **CT12DN917** has successfully completed a **12 weeks** Online Internship Program in the domain of **Artificial intelligence** in **CODTECH IT SOLUTIONS PRIVATE LIMITED**, from **June 10th, 2025 to September 10th, 2025**.

During the internship, His/Her demonstrated outstanding dedication, creativity, and technical proficiency. His/Her performance in the assigned projects was exceptional, showcasing deep understanding, innovative problem-solving skills, and a strong commitment to excellence.

We appreciate his/her active participation, consistent efforts, and impressive contribution to the overall success of the internship. We wish him/her all the best in future endeavors.

We are confident that his/her dedication and skills will lead to great achievements ahead.

Best Wishes,
NEELA SANTHOSH KUMAR
Human Resources & Academic Head
Hr@codtechitsolutions.com



ABSTRACT

The **AI-Based Cold Email Generator System** is an intelligent automation tool designed to simplify and enhance professional outreach by generating personalized cold emails from job postings. The system extracts job details such as role, required skills, experience, and description from company career websites and processes this information using Natural Language Processing (NLP) and a Large Language Model (LLM). It also integrates a portfolio matching mechanism using a vector database to identify and include the most relevant project links based on skill similarity. A user-friendly frontend built with Streamlit allows easy interaction, making the solution accessible to both technical and non-technical users. The project demonstrates the effective application of AI for automating communication tasks, reducing manual workload, and improving the accuracy and relevance of professional emails..

COMPANY PROFILE

INTRODUCTION TO INDUSTRY/ INSTITUTE CONTENT

Nature of Business of the Industry / Institution

CODTECH IT SOLUTIONS operates in the Information Technology (IT) and Software Services industry, primarily focusing on delivering technology-driven solutions and professional training services. The company is involved in the design, development, and deployment of modern digital applications and platforms that help businesses improve efficiency, visibility, and performance in the digital ecosystem.

The organization provides services such as web development, mobile application development, artificial intelligence solutions, data science, cybersecurity, and digital marketing. These services are aimed at helping clients build a strong online presence, streamline operations, and adopt emerging technologies to remain competitive in the market.

In addition to software and IT services, CODTECH IT SOLUTIONS also functions as a technical training and internship provider. The institution offers structured internship programs, live projects, and skill development training for students and fresh graduates in domains like Python, Machine Learning, Web Development, and AI. These programs are designed to bridge the gap between academic learning and industry requirements by emphasizing practical exposure.

Overall, the company operates as both a technology solutions provider and an educational platform, supporting business growth through IT services while nurturing future professionals through hands-on training and internships.

TABLE OF CONTENTS

SR. NO.	CONTENTS	PAGE NO.
1.	Introduction to Project	8
2.	Objective	9
3.	Architecture	10-12
4.	Workflow	13-14
5.	Features	15-16
6.	Advantages	17
7.	Limitations	18
8.	Conclusion	19
9.	References	20

CHAPTER 1 INTRODUCTION TO PROJECT

In the modern digital era, professional communication plays a crucial role in job searching, business development, and networking. One of the most commonly used methods for professional outreach is cold emailing, where individuals or organizations contact potential clients or employers without prior interaction. However, writing effective cold emails requires carefully understanding the job requirement, identifying suitable skills, and drafting a well-structured message, which can be time-consuming and repetitive.

With the rapid growth of Artificial Intelligence (AI) and Natural Language Processing (NLP), automation has become a powerful solution for handling repetitive communication tasks. This project, titled **AI-Based Cold Email Generator System**, focuses on eliminating the manual effort involved in writing personalized emails by extracting job-related information directly from career websites and generating structured, professional emails automatically.

The system scrapes job postings from any given URL, cleans the data, and uses a Large Language Model (LLM) to understand important job requirements such as role, skills, experience, and job description. To enhance accuracy and relevance, the project incorporates a portfolio matching feature that selects the most suitable project links based on skill similarity using a vector database. The frontend of the application is built using Streamlit, providing a clean and user-friendly interface for easy use.

This project demonstrates an efficient combination of AI, data processing, and web technology to build a real-world automation tool that saves time, improves accuracy, and enhances communication effectiveness in professional environments.

CHAPTER 2 OBJECTIVE

The main objective of the **AI-Based Cold Email Generator System** is to automate the process of generating professional and personalized cold emails using Artificial Intelligence. The specific objectives of the project are listed below:

- To develop a system that automatically extracts job postings from company career websites using URL-based scraping.
- To process and clean raw webpage data for accurate analysis using text pre-processing techniques.
- To use a Large Language Model (LLM) to understand job requirements such as role, skills, experience, and description.
- To generate professionally structured cold emails based on extracted job details.
- To implement a portfolio matching system that selects the most relevant project links using vector-based similarity search.
- To store and retrieve user portfolio data efficiently using a vector database.
- To design a user-friendly web interface using Streamlit for easy interaction.
- To reduce manual effort and save time for users.
- To improve email quality, personalization, and relevance.
- To make the system scalable and adaptable for future automation tasks.

CHAPTER 3 ARCHITECTURE

The architecture of the **AI-Based Cold Email Generator System** is modular, consisting of multiple interconnected components that work together to extract job information, match portfolio projects, and generate personalized cold emails. The system follows a **pipeline-based architecture**, where the output of one module becomes the input of the next.

1. Overview of System Architecture

The system is broadly divided into the following layers:

1. Presentation Layer (Frontend – Streamlit UI)
2. Application Layer (Logic & AI Processing – Chains, Portfolio, Utils)
3. Data Layer (Portfolio CSV + Vector Database + Environment Variables)

Data flows from the user, through the scraping and AI modules, into the portfolio matching system, and finally back to the user as a generated cold email.

2. Architecture Components

2.1 Presentation Layer – Streamlit Application

- Implemented in main.py.
- Responsibilities:
 - Takes URL input from user.
 - Triggers job extraction and email generation on button click.
 - Displays generated cold emails in a clean format.
- Uses:
 - WebBaseLoader to load webpage content.
 - Streamlit UI elements like st.text_input, st.button, and st.code.

2.2 Application Layer

This layer contains all the **core logic and AI workflows**.

a) Chain Module (LLM Orchestration) – chains.py

- Initializes the LLM (ChatOpenAI) using OpenRouter and Grok 4.1 Fast model.
- Contains:
 - extract_jobs(cleaned_text)
 - Takes cleaned webpage text as input.
 - Uses a prompt template to instruct the model to extract job posts.
 - Returns jobs in structured JSON format (role, experience, skills, description).
 - write_mail(job, links)
 - Uses job details and portfolio links.
 - Generates a professional cold email as model output.

b) Portfolio Module (Vector Search System) – portfolio.py

- Loads portfolio data from a CSV file.
- Uses **ChromaDB** as a persistent vector database.
- Components:
 - load_portfolio()
 - Reads each row (Techstack + Links).
 - Adds documents (tech stack text) to ChromaDB with metadata containing links.
 - query_links.skills()
 - Queries the vector store with job skills.
 - Returns most relevant portfolio links.

c) Utility Module (Text Cleaning) – utils.py

- Contains clean_text(text):
 - Removes HTML tags.

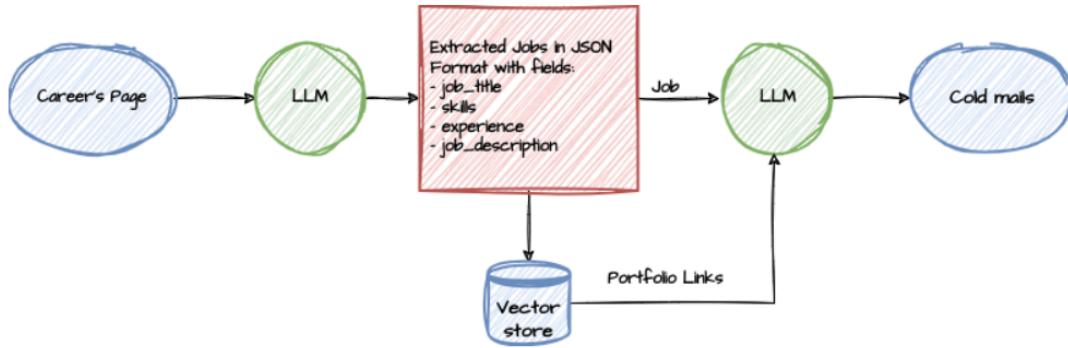
- Removes URLs and special characters.
- Normalizes whitespace.
- Ensures that the text passed to the LLM is clean, readable, and model-friendly.

3. Data Layer

- **Portfolio File:** my_portfolio.csv
 - Stores project links and tech stack information.
 - Acts as knowledge base for portfolio matching.
- **Vector Store:** ChromaDB
 - Persistent storage for portfolio embeddings.
 - Enables semantic similarity search using skills as queries.
- **Environment Variables (.env file)**
 - Stores API keys like OPENROUTER_API_KEY securely.
 - Loaded using dot env in chains.py.

CHAPTER 4 WORKFLOW

The AI-Based Cold Email Generator System works in a step-by-step pipeline, starting from user input and ending with a fully generated professional cold email. The complete working is explained below in a clear flow:



1. User Input through Streamlit Interface

- The user opens the web app and enters a careers page URL (for example: Amazon, Nike jobs page) in the text box.
- The user clicks the “Submit” button to start the process.

2. Webpage Loading & Scraping

- After clicking submit, the system uses WebBaseLoader from LangChain to load the webpage content from the given URL.
- The entire HTML/text content of the page is fetched and passed further for cleaning.

3. Text Cleaning & Pre-processing

- The raw page content may contain HTML tags, links, symbols, etc.
- This text is passed to the function `clean_text()` defined in `utils.py`.
- `clean_text()` performs:
 - Removal of HTML tags
 - Removal of URLs
 - Removal of special characters
 - Extra spaces normalization
- The cleaned text is now ready for AI-based processing.

4. Job Extraction using LLM (AI Model)

- The cleaned text is sent to the method `extract_jobs()` in the `Chain` class (`chains.py`).
- Inside this method:
 - A `PromptTemplate` is used to instruct the LLM that the text is from a careers page.
 - The model is asked to extract job postings and return them in JSON format with fields:

- The response from the LLM is parsed using `JsonOutputParser()` to ensure valid JSON.

5. Portfolio Loading into Vector Database

- Before generating emails, the system needs relevant portfolio links.
- The `Portfolio` class in `portfolio.py` loads data from `my_portfolio.csv`, which contains Techstack and Links.
- In `load_portfolio()`:
 - Each row's Techstack text is added as a document into ChromaDB.
 - The Links are stored as metadata.

6. Skill-Based Portfolio Matching

- For every extracted job, the system reads the skills list from the job JSON.
- It calls `query_links(skills)` from `Portfolio`.
- `query_links()`:
 - Sends the job skills as `query_texts` to ChromaDB.
 - Finds the most similar documents (tech stacks).
 - Returns the top 2 matching portfolio links from metadata (`n_results=2`).

7. Cold Email Generation using LLM

- Now, for each job:
 - The job description + matched portfolio links are passed to `write_mail(job, links)` in `chains.py`.
- Inside `write_mail()`:
 - A prompt is created that:
 - Describes the company AtliQ and its services.
 - Includes the job description.
 - Includes the portfolio links to showcase relevant work.
 - Instructs the LLM to act as Mohan, BDE at AtliQ and write a cold email.

8. Displaying the Output to User

- For each generated email, the Streamlit app uses `st.code(email, language='markdown')` to show the email properly formatted on the screen.
- If any error occurs (like parsing issue, connection failure), it is caught in a try-except block and shown to the user using `st.error()`.

CHAPTER 5 FEATURES OF SYSTEM

The **AI-Based Cold Email Generator System** includes several advanced features that make it efficient, intelligent, and easy to use. The major features of the system are explained below:

1. URL-Based Job Extraction

Users can simply provide a careers page link, and the system automatically scans and analyzes job postings directly from the website without requiring manual input.

2. AI-Powered Job Understanding

The system uses a Large Language Model (LLM) to extract important job details such as:

- Job role
- Required skills
- Experience level
- Job description

This ensures structured and accurate information retrieval.

3. Automatic Cold Email Generation

The system generates professional and human-like cold emails automatically based on extracted job information, saving time and effort.

4. Smart Portfolio Matching

Using a vector database, the system identifies the most relevant projects and links based on job skills and automatically includes them in the email.

5. Vector Database Integration

Portfolio data is stored in a semantic vector database (ChromaDB), enabling fast and accurate similarity-based search using embeddings.

6. Secure API Management

API keys are stored safely using environment variables (.env file) instead of hardcoding them in the code.

7. Clean Text Processing

The system removes unnecessary HTML tags, links, and symbols before sending content to the AI model, improving processing accuracy.

8. User-Friendly Interface

Built using Streamlit, the system offers a simple and interactive interface accessible even for non-technical users.

9. Error Handling & Validation

The system handles:

- Invalid URLs
- JSON parsing issues
- Model response failures

with proper error messages.

10. Multiple Job Support

If a webpage contains multiple job postings, the system extracts and processes them individually.

11. Scalable Architecture

The modular design allows easy modification and future upgrades like resume generation or email scheduling.

12. Minimal Manual Work

The system reduces manual effort by more than 90% by automating job analysis and email creation.

CHAPTER 6 ADVANTAGES

The **AI-Based Cold Email Generator System** offers many benefits that make it highly useful for job seekers and professionals. The major advantages of the system are listed below:

- ◆ **1. Saves Time**

Manually reading job descriptions and writing emails is time-consuming. This system automates the entire process and generates emails in seconds.

- ◆ **2. Improves Personalization**

Emails are generated based on actual job requirements and matched portfolio links, making communication more relevant and effective.

- ◆ **3. Reduces Human Effort**

Users no longer need to manually search portfolio links or write multiple emails. The system handles everything.

- ◆ **4. Enhances Productivity**

More emails can be created in less time, increasing outreach opportunities.

- ◆ **5. Professional Quality Output**

The AI produces well-structured, formal, and grammatically correct emails.

- ◆ **6. Smart Skill Matching**

Uses semantic search instead of keyword search for selecting portfolio links accurately.

- ◆ **7. User-Friendly Interface**

Anyone can use the system without technical knowledge.

- ◆ **8. Eliminates Repetition**

Repeating the same email writing activity is replaced with automation.

- ◆ **9. Secures API Keys**

Sensitive keys are stored using environment variables, improving security.

- ◆ **10. Cost-Effective**

Reduces the need for hiring writing assistants or marketers

CHAPTER 7 LIMITATIONS

Although the **AI-Based Cold Email Generator System** is efficient and powerful, it has certain limitations which are listed below:

- ◆ **1. Dependency on Internet Connection**

The system requires an active internet connection for web scraping and AI API access.

- ◆ **2. API Rate Limits**

AI model usage may be restricted by request limits imposed by the API provider.

- ◆ **3. Inconsistent Website Structures**

Some websites load jobs dynamically or block scraping, which may affect job extraction accuracy.

- ◆ **4. AI Output Variability**

The AI model may sometimes return incorrect or incomplete job data or JSON formatting issues.

- ◆ **5. Portfolio Quality Dependency**

If portfolio data is poor or incomplete, the generated email relevance decreases.

- ◆ **6. Limited Multilingual Support**

Currently, the system is optimized for English language emails only.

- ◆ **7. Error Handling Boundaries**

Some network or parsing failures may not be fully captured by the application.

- ◆ **8. Not a Replacement for Human Judgment**

Generated emails may still require review for tone adjustment or company-specific details.

CONCLUSION

The **AI-Based Cold Email Generator System** demonstrates the effective use of Artificial Intelligence to automate and enhance professional communication. By combining web scraping, Natural Language Processing, vector-based portfolio matching, and Large Language Models, the system successfully extracts job requirements and generates personalized cold emails with minimal manual input. The user-friendly interface ensures easy access for all users, while intelligent portfolio linking increases the relevance and impact of communications. Although the system has limitations such as API dependency and website structure variations, it proves to be a reliable and efficient solution for modern job outreach and business development, offering strong potential for future expansion and real-world deployment.

REFERENCES

1. LangChain Documentation –
<https://python.langchain.com/>
2. Streamlit Documentation –
<https://docs.streamlit.io/>
3. ChromaDB Documentation –
<https://docs.trychroma.com/>
4. Python Official Documentation –
<https://docs.python.org/3/>
5. Scikit-learn Documentation (for NLP concepts) –
<https://scikit-learn.org/>
6. OpenRouter API Documentation –
<https://openrouter.ai/docs>
7. X.AI Grok Model Overview –
<https://x.ai/>
8. Pandas Documentation –
<https://pandas.pydata.org/docs/>
9. dotenv Documentation –
<https://pypi.org/project/python-dotenv/>
10. Natural Language Processing Concepts –
<https://www.analyticsvidhya.com/blog/>
11. Vector Databases Overview –
<https://www.pinecone.io/learn/vector-database/>
12. JSON Format and Parsing –
<https://www.json.org/>