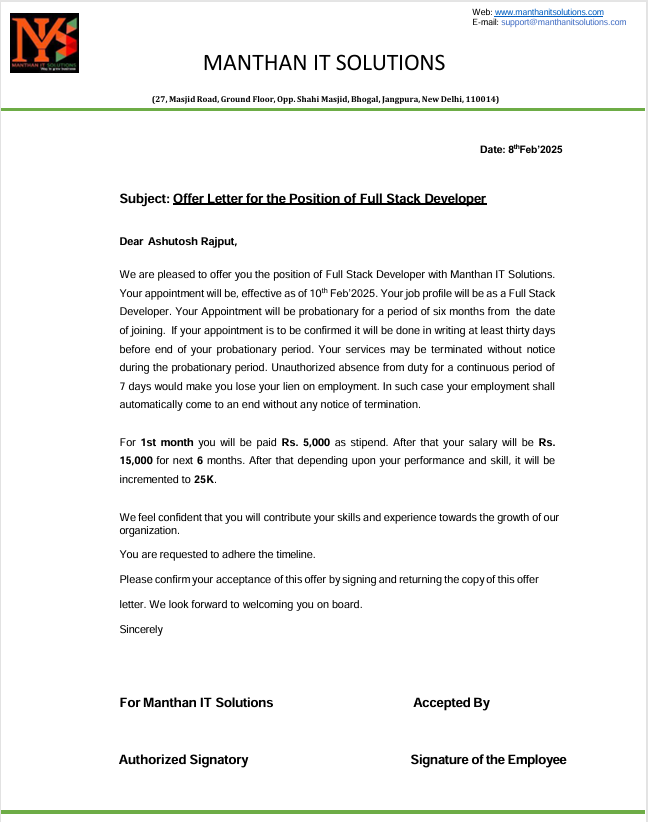
**OFFER LETTER**

****

**STUDENT DECLARATION**

I hereby declare that the work being presented in this report entitled **“Smart Email Assistant”** is an authentic record of my own work carried out under the supervision of **“Ms. Savita Singh”.**

The matter embodied in this report has not been submitted by me for the award of any other degree.

**Date:** **Signature of Student**

**Ashutosh Rajput**

**Dept.: Computer Applications**

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

**Signature of HOD** **Signature of Supervisor Prof.**

**(Dr.) Devendra Kumar** **Ms. Savita Singh**

**HOD-MCA Assistant Professor**

**Dept.: Computer Applications (Sr. Scale)**

**Date:**

**CERTIFICATE**

This is to certify that Project Report entitled **“Smart Email Assistant”** which is submitted by **Ashutosh Rajput** in partial fulfillment of the requirement for the award of degree **Master of Computer Application** in Department of Computer Applications of **Dr. A.P.J. Abdul Kalam Technical University,** is a record of the candidate own work carried out by him under my supervision.

The matter embodied in this Major Project Report is original and has not been submitted for the award of any other degree.

The plagiarism percentage evaluated for the content presented is **Six Percent.**

**Supervisor Signature**

**Ms. Savita Singh**

**Assistant Professor**

**Date: (Sr. Scale)**

**ACKNOWLEDGEMENT**

Introducing the report on the MCA project finished during MCA Final Year fills me with incredible happiness. I owe an exceptional obligation of appreciation to **Ms. Savita Singh, Assistant Professor (Sr. Scale) Department of Computer Applications, ABESEC, Ghaziabad** for her constant support and guidance throughout the course of my work. Her sincerity, thoroughness, and perseverance have been a constant source of inspiration for me. It is just her perceptive endeavors that our undertakings have come around.

I also take the opportunity to recognize the contribution of **Prof. (Dr.) Devendra Kumar Head, Department of Computer Applications, ABESEC, Ghaziabad** for his support and assistance during the development of the project.

I also don't want to miss the opportunity to thank the entire department's faculty members for their helpful support and cooperation during the project's development. Last but not least, I want to thank my friends for their help in getting the project done.

**Signature of Student**

**Ashutosh Rajput**

**2300320140039**

**MCA-IV Semester**

**ABSTRACT**

The Smart Email Assistant is a Spring Boot-based application enhanced with Spring AI to streamline and automate email communication in a dynamic and intelligent environment. Designed to assist users in managing their email interactions more effectively, the system leverages natural language processing and AI-powered suggestions to draft responses, summarize messages, and categorize emails intelligently.

By integrating modern AI capabilities with traditional backend development, the assistant provides real-time content analysis and response generation, significantly reducing the time spent on routine email tasks. It offers a user-friendly interface that ensures seamless interaction while maintaining data security and consistency. The system supports features like smart reply generation, keyword-based filtering, and contextual understanding of incoming messages.

This project aims to improve personal and professional productivity by transforming conventional email management into an intelligent and proactive experience. It serves as a scalable foundation for further integration with enterprise tools, making it a robust solution for smart communication in today’s fast-paced digital ecosystem.

.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sr. No.** | **TABLE OF CONTENT** | | | | **Page No.** |
| **Chapter 1.** | Introduction | | | | 1-4 |
|  | 1.1 | Problem Definition / Statement | | | 1 |
|  | 1.2 | Objective / Project Objective | | | 2 |
|  | 1.3 | Need of Project | | | 2 |
|  | 1.4 | Scope | | | 3 |
| **Chapter 2.** | Literature Review | | | | 5 |
| **Chapter 3.** | Feasibility Study | | | | 6-7 |
|  | 3.1 | Technical | | | 6 |
|  | 3.2 | Operational | | | 7 |
|  | 3.3 | Economic | | | 7 |
| **Chapter 4.** | System Requirements | | | | 8-15 |
|  | 4.1 | Functional Requirements | | | 8 |
|  | 4.2 | Non-Functional Requirements | | | 9 |
|  | 4.3 | Hardware Requirements | | | 11 |
|  | 4.4 | Software Requirements | | | 12 |
|  | 4.5 | Use Cases | | | 14 |
| **Chapter 5.** | System Design | | | | 16-21 |
|  | 5.1 | ER-Diagram | | | 16 |
|  | 5.2 | Data Flow Diagram | | | 17 |
|  | 5.3 | Use Cases | | | 19 |
|  | 5.4 | Sequence Diagram | | | 20 |
|  | 5.5 | Activity Diagram | | | 21 |
| **Chapter 6.** | GUI / Coding | | | | 22-47 |
|  | 6.1 | User Interface Design | | | 22 |
|  | 6.2 | Modules Screenshot | | | 24 |
|  | 6.3 | Coding | | | 29 |
|  |  | 6.3.1 | | Programming Languages and Tools Used | 29 |
|  |  | 6.3.2 | | Code Architecture and Organization | 31 |
|  |  | 6.3.3 | | Key Code Snippets | 35 |
| **Chapter 7.** | Testing (Test Plan/Cases/Result) | | | | 47-50 |
| **Chapter 8.** | Conclusion | | | | 51-55 |
|  | 8.1 | | Project Limitation | | 53 |
|  | 8.2 | | Future Scope | | 54 |
| **Chapter 9.** | References | | | | 56 |
|  | Appendices (Plagiarism Report) | | | | 57 |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **S. No.** | **Figure Description** | **Page No.** |
| **Figure-1** | ER Diagram | 16 |
| **Figure-2** | Data Flow Diagram | 17 |
| **Figure-3** | Use Case Diagram | 19 |
| **Figure-4** | Sequence Diagram | 20 |
| **Figure-5** | Activity Diagram | 21 |
| **Figure-6** | Spring Boot Project (Spring initilzer) | 24 |
| **Figure-7** | Gemini API Generate | 24 |
| **Figure-8** | Postman Check Gemini API | 25 |
| **Figure-9** | POST Request | 25 |
| **Figure-10** | POST Response | 26 |
| **Figure-11** | App.jsx Frontend | 26 |
| **Figure-12** | App..jsx Response | 27 |
| **Figure-13** | Email Writer Extension | 27 |
| **Figure-14** | Extension Integrate With Email | 28 |
| **Figure-15** | Email Output Generate With AI | 28 |
| **Figure-16** | Backend Project Directory | 32 |
| **Figure-17** | Frontend Project Directory | 33 |
| **Figure-18** | Extension Project Directory | 34 |
| **Figure-19** | Plagiarism Report | 57 |

**LIST OF ABBREVIATIONS**

|  |  |
| --- | --- |
| **Service** | **Full Form / Description** |
| **Gemini** | Google's Multimodal Large Language Model (LLM) |
| **Spring AI** | Spring Boot Abstraction Layer for AI/LLM Integration |
| **REST API** | Representational State Transfer Application Programming Interface (used in Spring Boot) |
| **React** | JavaScript Library for Building User Interfaces |
| **MUI** | Material UI – React Component Library for Styling |
| **Chrome Extension** | JavaScript-based browser extension for Gmail interaction |
| **DOM** | Document Object Model (used in Gmail content manipulation) |
| **Fetch API** | Web API for sending HTTP requests from frontend |
| **Spring Boot** | Java-based Framework for RESTful backend services |
| **Maven** | Java Build and Dependency Management Tool |
| **Gmail** | Google Email Service (target UI for the extension) |
| **HTML/CSS/JS** | Web Technologies used in frontend & Chrome extension |
| **Mutation Observer** | JavaScript API for detecting DOM changes in Gmail compose window |