International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

ZORO – AI Driven Task Automation

Ms.Narmatha M[1],Jaya Shree Lakshmi S[2]

[1]Assistant Professor, Department of Artificial Intelligence and Machine Learning,Sri Shakthi Institute of Engineering and Technology, Coimbatore – 641062,India

[2]Bachelor of Technology,Department of Artificial Intelligence and Machine Learning,Second Year,Sri Shakthi Institute of Engineering and Technology, Coimbatore – 641062,India

A B S T R A C T

This paper presents the design and implementation of **ZORO – AI Driven Task Automation**, a full-stack web application developed using Python (Flask) that enables hands-free email communication using voice commands. The system is designed for startups and small businesses to enhance communication efficiency by integrating offline speech recognition through the Vosk library, email automation using smtplib, and role-based access control via Flask-Session. ZORO supports multi-recipient handling, CC/BCC, file attachments, and real-time voice-to-text transcription. The system is modular, scalable, and designed to reduce manual effort while increasing accessibility and productivity.

Keywords: Email Automation, Flask, Speech Recognition, Voice Commands, Vosk, Role-based Access, smtplib, SQLite3

1. Main text

Thisplatform was conceptualized to solve the problem of inefficient email communication in small-scale businesses and startup environments. Traditional email drafting is a time-intensive process that limits multitasking and can be restrictive for users with accessibility needs. ZORO streamlines this by introducing a voice-controlled interface that can compose, edit, and send emails using natural language inputs, operating entirely offline for privacy and reliability.

Nomenclature

Symbol	**Definition**
V	Voice input captured via microphone
T	Transcribed text using Vosk engine
E	Email content generated from commands
R	Recipient list including To, CC, BCC

1.1. Structure

ZORO is developed with a clean, modular architecture:

* *Corresponding author.* Tel.: +91-9486186364
E-mail address: shreejayalakshmis@gmail.com

- **Frontend:** HTML, CSS, JavaScript (Vanilla JS)
- **Backend:** Python (Flask Framework)
- **Database:** SQLite3 for user data and session handling
- **Voice Recognition:** Vosk Offline Speech Recognition
- **Email Transmission:** Python's smtplib
- **Authentication:** Flask-Session, with JSON-based role definitions
- **Security:** Role-based access, user session encryption, restricted command execution

The backend exposes RESTful endpoints to perform operations such as voice command processing, email draft creation, sending, and user authentication.

Table 1 - Key API Endpoints in ZORO

Functionality	**Endpoint**	**Method**
User login	/api/login	POST
Process voice command	/api/process_command	POST
Send email	/api/send_email	POST
Retrieve transcription	/api/get_transcription	GET

1.2. Construction of references

The paper is structured into key sections: **Introduction, System Architecture, Implementation, Results, and Conclusion.** Each section elaborates on system modules, backend integration, voice recognition workflow, and overall application performance.

1.3. Section headings

The paper is structured into key sections: **Introduction, System Architecture, Implementation, Results, and Conclusion.** Each section elaborates on system modules, backend integration, voice recognition workflow, and overall application performance.

1.4. General guidelines for the preparation of your text

Variable names such as V (voice), T (transcription), and E (email draft) are consistently used. Command processing handles natural speech patterns using mapped keywords. Email metadata such as subject, recipients, and body are extracted in real-time. All system messages are timestamped in ISO 8601 format.

1.5. File naming and delivery

The system is organized into a Flask application directory:

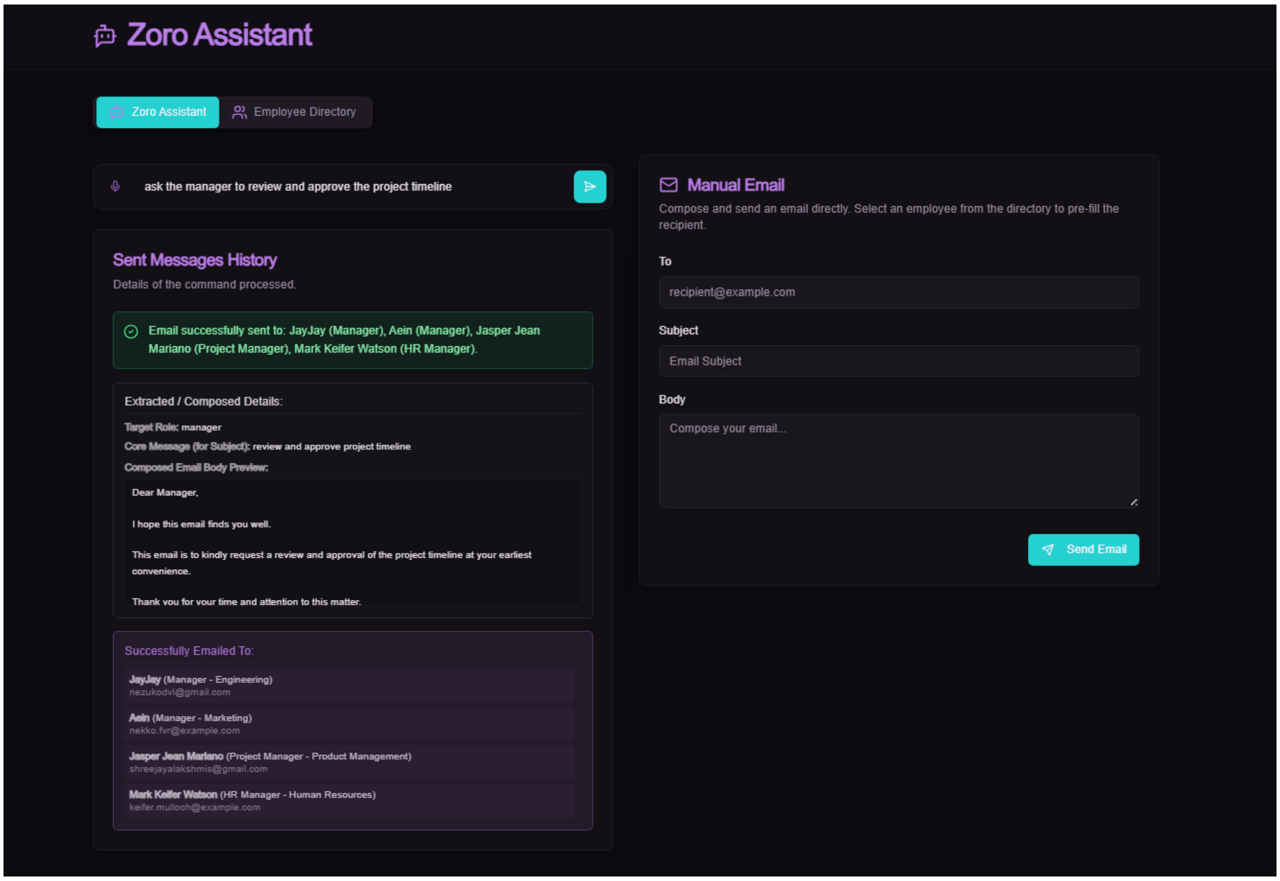
- app.py – Main application
- /static – JavaScript, CSS, media files
- /templates – HTML frontend templates
- /voice – Vosk model integration and utilities
- utils.py – Functions for command parsing, email formatting, etc.

1.6. Footnotes

Security methods including Flask-Session token generation, session lifetime control, and SMTP server environment variable encryption are described in-line. Footnotes are minimized to keep the document accessible and technical.

2. Illustrations

Fig. 1 – Workflow of ZORO from voice command capture to final email delivery.



ask the manager to review and approve the project timeline

Sent Messages History

Details of the command processed.

Email successfully sent to: JayJay (Manager), Aein (Manager), Jasper Jean Mariano (Project Manager), Mark Keifer Watson (HR Manager).

Extracted / Composed Details:

Target Role: manager

Core Message (for Subject): review and approve project timeline

Composed Email Body Preview:

Dear Manager,

I hope this email finds you well.

This email is to kindly request a review and approval of the project timeline at your earliest convenience.

Thank you for your time and attention to this matter.

Successfully Emailed To:

JayJay (Manager - Engineering)
nezukodvl@gmail.com

Aein (Manager - Marketing)
nekko.fvr@example.com

Jasper Jean Mariano (Project Manager - Product Management)
shreejayalakshmis@gmail.com

Mark Keifer Watson (HR Manager - Human Resources)
keifer.mulloch@example.com

Manual Email

Compose and send an email directly. Select an employee from the directory to pre-fill the recipient.

To: recipient@example.com

Subject: Email Subject

Body: Compose your email...

Fig. 1 - (a) first picture;

3. Equations

A sample formula for generating role-based permissions

permission = JSON[User][Role]

server.sendmail(sender_email, recipient_list, message)

T = Vosk_Model(V)

4. Online license transfer

The authors agree to license this work under standard open-access publication agreements. All source code and resources have been originally developed by the author and are available upon request for academic and research purposes.

Acknowledgements

The author extends gratitude to the faculty and mentors of Sri Shakthi Institute of Engineering and Technology for their valuable feedback and support. Special thanks to the testers and early adopters who helped refine ZORO's voice automation capabilities.

Appendix A. An example appendix

A.1. *Leave Submission Example:*

```
{
  "to": ["john@example.com"],
  "cc": ["team@example.com"],
  "subject": "Project Update",
  "body": "The project is on track and will be delivered by Friday."
}
```

References

Van der Geer, J., Hanraads, J. A. J., & Lupton, R. A. (2000). The art of writing a scientific article. *Journal of Science Communication*, 163, 51–59.

Mettam, G. R., & Adams, L. B. (1999). How to prepare an electronic version of your article. In B. S. Jones & R. Z. Smith (Eds.), *Introduction to the electronic age* (pp. 281–304). New York: E-Publishing Inc.

Flask Documentation: https://flask.palletsprojects.com/

Python ReportLab User Guide: https://www.reportlab.com/docs/reportlab-userguide.pdf

OWASP Foundation. (2022). Web Application Security Cheat Sheet.

.