for a hackathon.. I am trying to build a local mail server, like in a company.. it will be a local server.. where all the mail ids and mails will be hosted.. and we will also build a password manager.. which they can use for their employeed, passwords will saved locally on their system and server, company server

ChatGPT said:

Great idea for a hackathon! You're building an on-premises local mail server combined with a local password manager, which can be a great tool for small businesses or internal enterprise use. Here's how you can structure your project.



💡 Project Name Idea

IntraMail – A local, secure mail server and credential manager for internal company use.

Project Architecture

Overview

- Local-only email server (not connected to external SMTP services).
- A web interface for employees to:
 - Send/receive emails.
 - Store/manage passwords.
- Admin interface to manage user accounts and permissions.

Components

1. Mail Server

You can use or build a lightweight mail server:

- Option 1: Build your own basic SMTP/IMAP server
 - Use Python (smtpd / aiosmtpd + imaplib) or Node.js

- Store mails in local DB (e.g., SQLite or file system)
- Option 2: Use open-source lightweight mail server
 - o MailHog (for dev/local)
 - o <u>Docker mailserver</u> (advanced, full-featured)

For the hackathon, building a **simplified custom mail server** might impress more than using a full stack.

™ Features:

- Send mail (internal only)
- Inbox view
- Store mails in JSON or SQLite
- No spam, no forwarding just internal communication

2. Password Manager

A secure vault for employee credentials (e.g., logins to company systems).

Features:

- Store passwords encrypted (AES or bcrypt)
- Group passwords by department/user
- Master password for admin
- Option to backup/restore vault
- Local-only access (no cloud sync)

Proch stack:

• Backend: Python Flask / Node.js / Go

- Frontend: React / Svelte / simple HTML+JS
- Encryption: use cryptography lib in Python or crypto in Node.js

Stack Suggestion

Component	Tech Suggestion
Backend	Python Flask or Node.js
Frontend	React.js / plain HTML+JS

Database SQLite (lightweight, local)

Mail Protocol Custom SMTP / JSON-based

Encryption Python cryptography or Node

crypto

Auth JWT or session-based

Deployment Run locally on LAN, or Docker

Possible Features to Demo

- 1. Send/receive internal mail
- 2. View inbox with timestamps
- 3. Save and retrieve passwords securely
- 4. Admin adds/removes users
- 5. Everything works locally (LAN demo)

Security Tips

• Encrypt passwords before saving to DB

- Never store raw passwords
- Use bcrypt or scrypt for user auth passwords
- Secure sessions (local cookie or token-based)
- Don't hardcode credentials use .env files

UI Ideas

- Left panel: Inbox, Sent, Compose
- Right panel: View email or compose
- Password manager: List of credentials, Add/Edit/Delete
- Admin panel (toggle switch or login page)

※ Bonus Points (if time allows)

- Email search/filter
- Password generator
- Export/import vault
- Theming (light/dark mode)
- Logs (who sent what/when)