**每頁具體內容安排**

**第一頁：封面**

 **標題**：電子公文系統技術文件

 **作者**：您的姓名

 **日期**：2024年4月27日

 **公司/組織標誌**（如有）

 **設計**：簡潔專業的版面設計，使用公司標準色調。

**第二頁：目錄**

 **自動生成的目錄**：

 引言

 系統架構

 功能模塊詳解

 登入與註冊

 密碼修改與找回

 文件管理

 安全性設計

 部署與運行

 問題排查與常見問題解答

 結論

 附錄

**第三頁至第五頁：引言**

 **系統背景**：

 電子公文系統的開發背景與必要性。

 **系統目的**：

 提供高效、安全的公文管理解決方案。

 **文件範圍**：

 涵蓋系統的設計、功能、部署和安全性等方面。

 **讀者對象**：

 系統開發人員、維護人員和最終用戶。

**第六頁至第八頁：系統架構**

 **總體架構圖**：

 前端、後端、資料庫的交互關係。

 **技術堆棧介紹**：

 前端：HTML、CSS、JavaScript

 後端：Python、Flask 框架

 資料庫：SQLAlchemy

 **資料庫設計概要**：

 用戶表、文件表、附件表等主要數據表結構。

**第九頁至第十二頁：功能模塊詳解**

**登入與註冊**

 **功能描述**：

 用戶帳號創建與驗證。

 **使用流程圖**：

 用戶註冊、登入流程。

 **代碼說明**：

 routes.py 中相關路由和邏輯。

 python:app/routes.py @main.route("/login", methods=["GET", "POST"]) def login(): ...

**密碼修改與找回**

 **功能描述**：

 用戶密碼的修改與重置。

 **安全措施**：

 密碼加密存儲，重置令牌生成與驗證。

 **代碼說明**：

 routes.py 中 change\_password 和 forgot\_password 路由。

 python:app/routes.py @main.route("/change\_password", methods=["GET", "POST"]) @login\_required def change\_password(): ...

**文件管理**

 **功能描述**：

 文件的上傳、查詢、簽核與下載。

 **使用流程和界面截圖**：

 上傳文件流程圖，查詢界面的截圖。

 **代碼實現概要**：

 routes.py 中 upload 路由及相關功能。

 python:app/routes.py @main.route('/upload', methods=['GET', 'POST']) @login\_required def upload(): ...

**第十三頁至第十五頁：安全性設計**

 **密碼加密與驗證機制**：

 使用 bcrypt 或其他加密算法保護用戶密碼。

 **數位簽章與簽名驗證**：

 數位簽章的創建與驗證過程。

 使用 RSA 算法進行簽名與加密。

 **資料加密與解密流程**：

 AES 加密文檔內容，公鑰加密 AES 密鑰。

**第十六頁至第十八頁：部署與運行**

 **系統部署步驟**：

 環境設置（如 Python 版本、依賴包安裝）。

 **依賴項和環境設置**：

 列出所有需要安裝的依賴項及其版本。

 **運行指引和測試方法**：

 如何啟動應用，進行基本測試確認系統運行正常。

**第十九頁至第二十一頁：問題排查與常見問題解答**

 **常見問題及解決方案**：

 登入失敗、文件上傳錯誤等常見問題的排查步驟。

 **錯誤信息的解釋與修復方法**：

 解釋常見錯誤代碼及如何修復。

**第二十二頁：結論**

 **系統優勢總結**：

 高效的文件管理、安全的數據保護等優勢。

 **未來改進方向**：

 增加更多功能模塊，如多語言支持、高級搜索等。

**第二十三頁：附錄**

 **重要代碼片段**：

 關鍵功能的代碼展示，如密碼加密、數位簽章等。

 python:app/util.py def encrypt\_private\_key(private\_key): ...

 **參考文獻和資源鏈接**：

 使用的技術文檔、框架官方文檔鏈接等。

**最終注意事項**

 **保持內容一致性**：確保所有章節風格一致，標題清晰，段落分明。

 **檢查代碼格式**：使用等寬字體顯示代碼，保持代碼的可讀性。

 **圖片質量**：確保所有截圖清晰，圖表標註明確。

 **備份文件**：隨時備份技術文件，防止數據丟失。

 **時間管理**：嚴格按照計劃完成每天的任務，留有足夠時間進行審核和修改。

祝您順利完成技術文件的撰寫！