Implementation Notes

**Date: 2025-03-04**

# 1. Email Verification System Implementation

## A. register\_email.html Implementation

**HTML:**- Complete HTML structure with Bootstrap integration  
- Form components:  
 • Email input field with validation  
 • Submit button for verification email  
 • Error message display section  
 • Login link for existing users  
- Styling:  
 • Custom error message styling  
 • Responsive form container  
 • Bootstrap classes for layout

## B. email\_sent.html Implementation

**HTML:**- Confirmation page structure:  
 • Success icon using Font Awesome  
 • Verification email sent message  
 • Email check prompt  
 • Spam folder check alert  
- Features:  
 • Resend verification option  
 • Link back to registration  
 • Responsive design with Bootstrap

## C. register.html Updates

**HTML:**- Title update to "Register - Complete Profile"  
- New components:  
 • Hidden email input for verified address  
 • Verified email display section  
 • Updated form submission flow  
- Visual feedback:  
 • Success indicators for verified email  
 • Error message handling  
 • Updated button text

# 2. Survey System Implementation

## A. Database Schema

**SQL:**CREATE TABLE user\_surveys (  
 id INTEGER PRIMARY KEY AUTOINCREMENT,  
 user\_id INTEGER NOT NULL,  
 experience\_rating TEXT NOT NULL,  
 is\_helpful BOOLEAN NOT NULL,  
 created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 FOREIGN KEY (user\_id) REFERENCES users (id)  
);

## B. Survey Modal Implementation

**HTML:**<div class="modal fade" id="surveyModal">  
 - Experience Rating Section:  
 • Positive/Neutral/Negative options  
 • Radio button implementation  
 • Required field validation  
 - Helpfulness Section:  
 • Yes/No options  
 • Required field validation  
 - Submit button with event handler

## C. Survey JavaScript Implementation

**JavaScript:**- State Management:  
 • hasSubmittedSurvey flag  
 • isLoggingOut flag  
  
- Core Functions:  
 • showSurveyModal()  
 • submitSurvey()  
 • Event handlers for:  
 - Logout button  
 - Page unload  
 - Form submission  
  
- API Integration:  
 • POST request to /submit\_survey  
 • FormData handling  
 • Response processing  
 • Error handling

# 3. PDF Document Handling Improvements

## A. Enhanced ChatService Class

**Python:**class ChatService:  
 def get\_document\_context(self, message: str) -> str:  
 # Page Detection  
 page\_match = re.search(r'page\s\*(\d+)', message.lower())  
 page\_number = int(page\_match.group(1)) if page\_match else None  
   
 # Context Retrieval  
 if self.vector\_store.\_vectorstore:  
 relevant\_docs = self.vector\_store.search\_similar(message, k=3)  
   
 # Document Processing  
 for doc in relevant\_docs:  
 page = doc.metadata.get('page', 1)  
 if page\_number and page != page\_number:  
 continue  
 context\_parts.append(f"Page {page}:\n{doc.page\_content.strip()}")

## B. System Prompt Enhancements

**Python:**base\_prompt = (  
 "You are a helpful assistant for the current conversation only. "  
 "When referring to document content, always mention the page number and filename. "  
 "If asked about specific pages, focus on content from those pages. "  
 "If no relevant content is found for a specific page, clearly state that. "  
 "Format your responses in a clear, organized manner. "  
 "If any one asks about the data say the data is crawled from the internet "  
 "if user not ask about the data then no need to say the data is crawled from the internet"  
)

# 4. Integration Testing Points

A. Email Verification Flow:  
 - Email submission → Verification email → Registration completion  
  
B. Survey Trigger Points:  
 - Logout button click  
 - Browser/tab close  
 - Form submission  
  
C. PDF Document Queries:  
 - General content search  
 - Page-specific queries  
 - Error cases

# 5. Security Considerations

A. Email Verification:  
 - Token-based verification  
 - Expiration handling  
 - XSS prevention in templates  
  
B. Survey Data:  
 - User authentication required  
 - Input validation  
 - CSRF protection  
  
C. Document Handling:  
 - Access control  
 - Input sanitization  
 - Error message security