

# # User Management and File Upload System Specification (V1)

## ## 1. Introduction

This document outlines the specifications for a User Management and File Upload System. The system allows users to register, log in, upload files, view uploaded files, and manage their uploads.

## ## 2. User Management

### ### 2.1 Registration

- Users can register with an email and password.
- Passwords must be at least 8 characters long.
- Email addresses must be unique in the system.
- User information is stored in a CSV file (`userprofile.csv`).

### ### 2.2 Login

- Users can log in using their registered email and password.
- The system verifies credentials against the information in `userprofile.csv`.
- Successful login creates a session for the user.

### ### 2.3 User Roles

- Basic user role implemented.
- Admin role functionality prepared but not fully implemented.

## ## 3. File Upload System

### ### 3.1 File Upload

- Logged-in users can upload multiple files (up to 10) at once.
- Files are stored in an 'uploads' directory on the server.
- File information (filename, original name, upload date, uploader) is stored in memory.

### ### 3.2 View Uploaded Files

- Users can view a list of all uploaded files.
- The list displays: original file name, uploader's email, and upload date.
- Files are sorted by upload date, with the most recent first.

### ### 3.3 Delete Files

- Users can delete their own uploaded files.
- Deletion removes the file from both the server and the file list.

### ### 3.4 Duplicate File Handling

- When a user uploads a file with the same name as a previously uploaded file, they are prompted to:
  1. Keep both files
  2. Replace the old file with the new one
  3. Keep the old file and discard the new one

### ### 3.5 Export File Information

- Users can export information about all uploaded files to a CSV file named "resumeupload.csv".
- The CSV includes: file ID, filename, original name, upload date, and uploader's email.

## ## 4. Technical Implementation

### ### 4.1 Backend

- Node.js with Express.js framework
- File storage using local file system
- User data stored in CSV format
- Session management for user authentication

### ### 4.2 Frontend

- EJS (Embedded JavaScript) for server-side rendering
- Basic CSS for styling

### ### 4.3 Data Storage

- User profiles: CSV file (`userprofile.csv`)
- Uploaded files: Local directory ('uploads')
- File information: In-memory JavaScript array (not persistent across server restarts)

## ## 5. Security Considerations

- Passwords are hashed before storage
- Session-based authentication
- Basic input validation implemented

## ## 6. Limitations and Future Improvements

- Implement persistent storage for file information (e.g., database)
- Enhance admin functionality
- Implement more robust error handling and user feedback
- Add user profile management features
- Implement file type restrictions and size limits for uploads
- Add pagination for the file list view
- Implement search and filter functionality for uploaded files

## ## 7. Conclusion

This specification outlines the current implementation of the User Management and File Upload System. It provides a foundation for user authentication, file uploading, and basic file management, with room for future enhancements and improvements.