File Upload and Management Application

Prepared by: Semih Gökmen

Date: 22.05.2025

1. Introduction

This report explains the design, implementation, and technologies used in the project titled “File Upload and Management Application,” requested by DFA Bilişim as part of internship tasks. The project is a web-based platform where users can securely upload, manage files, register, and log in. The application is designed with simplicity, usability, and security as priorities.

2. Project Objectives

- Enable users to manage their files online in a secure and user-friendly environment.
- Provide basic authentication features (registration, login, password reset).
- Support file upload, listing, downloading, and deletion operations.
- Ensure each user can only access their own files.
- Provide a modern and responsive user interface.

3. System Architecture

The application is developed using a simple client-server architecture:

- Frontend: HTML, CSS, JavaScript
- Backend: PHP
- Server: PHP built-in server
- Data Storage: JSON files (users.json, uploads.json)
- File Storage: Local directory (uploads/)

- Development Environment: Visual Studio Code

4. Technologies Used and Reasons

4.1 PHP

- Reason: PHP is a widely supported, easy-to-deploy language suitable for small to medium web applications. Procedural PHP is sufficient for rapid prototyping.
- Usage: For authentication, file upload/download/delete, and data management operations.

4.2 HTML & CSS

- Reason: Standard technologies for structuring and styling web pages. CSS provides responsive and modern design.
- Usage: All user interfaces (login, register, dashboard, file list, etc.) are created using HTML and CSS.

4.3 JavaScript

- Reason: Enables dynamic user interactions such as drag-and-drop file upload, updating file lists without page reload, and table sorting.
- Usage: File selection, AJAX-based file upload and deletion, UI feedback.

4.4 JSON Files

- Reason: Lightweight and easy-to-manage data storage method. Suitable for small-scale applications without a database requirement.
- Usage: users.json stores user information (passwords hashed), uploads.json stores file upload data.

4.5 Font Awesome (CDN)

- Reason: Popular modern icon set; improves user experience and visual simplicity.

- Usage: Icons for file upload, download, delete, logout actions.

4.6 Server: PHP Built-in Server

- Reason: PHP's built-in server allows quick testing and development without configuration, ideal for small projects and local development.
- Usage: Used to run the application during local development and testing without needing an external web server setup.

4.7 Development Environment: Visual Studio Code

- Reason: Lightweight, widely used editor with extensive plugin support and integrated terminal for modern web development.
- Usage: Writing code, managing project files, running PHP server, debugging.

5. Security Measures

- Password Hashing: User passwords are securely stored using PHP's password_hash() function.
- Session Management: Authentication is controlled via PHP sessions. Only logged-in users can access file operations.
- File Validation: Only PDF, JPG, and PNG file types are allowed. File names are sanitized to prevent directory traversal attacks.
- User Isolation: Each user can only view and manage their own files.

6. Application Structure

- dashboard.php – Main panel after login
- login_form.php, register_form.php, forgot_password_form.php – Authentication screens
- file_list.php – Page listing user files with management options
- helpers.php – Helper functions for sessions and JSON handling
- upload_process.php – File upload handling
- delete_file.php – File deletion handling
- login_process.php, register_process.php, reset_password_process.php – Authentication logic
- logout_process.php – Session termination
- uploads/ – Directory storing uploaded files

- users.json, uploads.json – Data storage files

7. User Experience

- Registration/Login: Simple, validated forms with user feedback.
- File Upload: Drag-and-drop support and multiple file selection.
- File Management: Users can only view, download, and delete their own files.
- Password Reset: Secure and user-friendly reset process.

8. Limitations and Future Work

- Scalability: JSON files are not suitable for large data or concurrent users; migrating to a database like MySQL is recommended.
- Security: Additional measures such as CSRF protection, rate limiting, and virus scanning should be implemented for production.
- Additional Features: User roles, file sharing, advanced search and filtering could be added in the future.

9. Conclusion

The File Upload and Management Application provides a secure, user-friendly, and modern way to handle basic file management on the web. The selected technologies balance simplicity and functionality, making the project easy to deploy and maintain for educational or personal use.