CHAITANYA ENGINEERING COLLEGE KOMMADI,

VISAKHAPATNAM – 530048

Certificate of Project Completion

SMART CAMPUS SERVICE BOT

A comprehensive web application with a JARVIS-inspired interface, featuring Attendance, AI-Powered Quizzes, AI-Powered Book Tools, Code Explainer, Personal Storage, and an Interactive Chatbot. It includes intelligent HTML, JS, and CSS bots, demonstrating versatility and modular architecture.

Done by/Group Members :

Pondara Akhil Behara (Team lead)	23L61A5434
Janni kavyasree	23L61A5420
Gorli Balaji	23L61A5416
Vutapalli Delliswara Pavan Kumar	24L65A5408
Kotla Leela Murali Krishna	24L65A5404

Lecturer In-charge

Head of the Department

External Examiner

ACKNOWLEDGEMENTS

We would like to express our sincere gratitude to all those who have contributed to the successful completion of this Smart Campus Bot project.

First and foremost, we extend our heartfelt thanks to the management of CHAITANYA ENGINEERING COLLEGE for providing us with the necessary infrastructure and support throughout the course of this project.

We are deeply grateful to our project guide, [Guide Name], for their invaluable guidance, continuous encouragement, and constructive feedback that helped shape this project into its current form. Their expertise and insights have been instrumental in overcoming various challenges we encountered.

We also acknowledge the support of our faculty members and technical staff who provided timely assistance and resources whenever needed. Their cooperation and willingness to help have been greatly appreciated.

Our special thanks go to our follow students who provided valuable feedback during the development process. Their suggestions helped us refine the user experience and improve the overall functionality of our system.

Finally, we thank our teammates for their dedication, hard work, and collaborative spirit. The successful completion of this project is a testament to our collective efforts and teamwork.

- Pondara Akhil Behara

- Janni Kavyasree

- Gorli Balaji

- Vutapalli Delliswara Pavan Kumar

- Kotla Leela Murali Krishna

INDEX:

1. **INTRODUCTION**

1.1 Abstract

The Smart Campus Bot project represents a comprehensive implementation of a campus management system with a focus on enhancing student and administrative experiences through innovative digital solutions. As students of Artificial Intelligence & Data Science at CHAITANYA ENGINEERING COLLEGE, we have designed and developed a system that addresses the challenges of modern campus management through a combination of web technologies and intelligent automation.

The system leverages cutting-edge technologies in JavaScript, HTML5, CSS3, and IndexedDB to deliver a solution that is both efficient and scalable. Our approach integrates client-side data management with responsive user interface design to achieve optimal performance metrics. The project demonstrates significant advancements in campus service delivery compared to traditional approaches, offering improved accessibility, efficiency, and user experience.

As aspiring game developers and AI makers, we have incorporated elements of interactive design and intelligent systems that reflect our career aspirations. The project not only serves as an academic exercise but also as a portfolio piece showcasing our technical capabilities and innovative thinking.

Key features of our implementation include Attendance Tracking, Personal Cloud Storage, and several planned enhancements that collectively address the core challenges identified in campus management. Through rigorous testing and validation, we have demonstrated the effectiveness of our solution in real-world scenarios.

1.2 Purpose

The primary purpose of this Smart Campus Bot project is to develop a comprehensive solution for campus service management that demonstrates our proficiency in web development and client-side data management while addressing practical challenges in educational institutions. This project serves multiple objectives:

1. To apply theoretical knowledge gained in our AI & DS curriculum to a real-world problem in campus management

2. To develop professional skills in project management, team collaboration, and technical documentation

3. To create a portfolio-worthy project that showcases our capabilities to potential employers or academic institutions

4. To contribute to the field of educational technology by proposing innovative solutions to existing challenges

5. To prepare for careers in game development and AI by working on a project that combines creative design with technical excellence

As future game developers and AI specialists, this project allows us to demonstrate our ability to think creatively while maintaining technical rigor. The system architecture has been designed with extensibility in mind, allowing for future enhancements and adaptations to different use cases.

The project specifically focuses on two core modules - Attendance Tracking and Personal Cloud Storage - which form the foundation of our campus management solution. These modules have been implemented with robust functionality and user-friendly interfaces that cater to both student and administrative needs.

1.3 Scope

The scope of this Smart Campus Bot project encompasses the complete development lifecycle from requirements analysis to final implementation and testing. Specifically, our project covers:

- Comprehensive analysis of campus management challenges and identification of key requirements

- Design of a scalable system architecture using modular web development principles

- Implementation of core functionalities including Attendance Tracking and Personal Cloud Storage

- Development of an intuitive user interface with responsive design principles

- Rigorous testing and validation using manual testing methodologies

- Documentation of all aspects of the project for academic and professional purposes

The project is designed to be modular, allowing for future expansion and integration with other systems. While the current implementation focuses on Attendance and Storage modules, the architecture supports extension to additional modules in future iterations.

Limitations of the current scope include the absence of server-side processing and database integration, which represents opportunities for future work. The project adheres to academic standards while maintaining practical relevance to industry requirements.

The system is designed to operate entirely on the client-side using modern web technologies, making it accessible across different devices and platforms without requiring specialized server infrastructure.

2. PROJECT MANAGEMENT

2.1 System Architecture

Our Smart Campus Bot system follows a modular client-side architecture designed for scalability, maintainability, and performance. The architecture comprises several interconnected components that work together to deliver the complete solution.

[ASCII diagram would be included here in the actual document]

Components:

1. Presentation Layer: Responsible for user interaction and interface rendering. Built using HTML5, CSS3 with responsive design principles to ensure compatibility across devices.

2. Business Logic Layer: Contains the core application logic and algorithms. Implemented in JavaScript with a focus on modularity and testability.

3. Data Access Layer: Manages data persistence and retrieval operations. Utilizes IndexedDB for client-side storage with optimized data handling strategies.

4. Module Components: Implements specific functionalities such as Attendance Tracking and Personal Cloud Storage. Each module is self-contained with its own HTML, CSS, and JavaScript files.

5. Utility Layer: Provides common functions and helper methods used across different modules. Includes validation, sanitization, and formatting utilities.

The architecture emphasizes separation of concerns, with each component having well-defined responsibilities and interfaces. This design enables parallel development by our five-member team, with clear boundaries between modules.

Key architectural decisions include:

- Use of client-side storage (IndexedDB) for data persistence without requiring a server

- Implementation of modular design to facilitate easy maintenance and expansion

- Selection of modern web technologies (HTML5, CSS3, JavaScript ES6+) for broad compatibility

- Incorporation of responsive design principles for cross-device compatibility

The system is designed to handle multiple concurrent users within browser limitations, with efficient data handling and minimal resource consumption. The modular structure allows for individual components to be updated or replaced without affecting the entire system.

2.2 Team Organization

Our five-member team was organized based on complementary skills and interests, aligning with our career goals in AI and game development:

- Pondara Akhil Behara (Team Lead): Responsible for overall project coordination, architecture design, and core module development. Leveraged leadership skills and technical expertise to guide the team toward successful completion.

- Janni Kavyasree: Focused on frontend development and user interface design, creating visually appealing screens with smooth interactions. Applied design principles to enhance user experience.

- Gorli Balaji: Managed module integration and testing, ensuring seamless component interaction and comprehensive validation. Implemented business logic and utility functions.

- Vutapalli Delliswara Pavan Kumar: Specialized in data handling and storage optimization, focusing on efficient client-side data management. Conducted extensive testing and debugging.

- Kotla Leela Murali Krishna: Handled documentation and project presentation, ensuring comprehensive records and effective communication of project features. Coordinated version control and deployment processes.

Regular team meetings were conducted to track progress, resolve issues, and maintain alignment with project goals. Version control using Git enabled effective collaboration and change management.

The team followed an agile development methodology with iterative improvements and continuous integration. Each member took ownership of specific components while maintaining open communication for cross-functional collaboration.

2.3 Project Timeline

The Smart Campus Bot project was completed over a twelve-week duration with the following milestones:

Week 1-2: Requirements gathering and analysis, system design and architecture planning

Week 3-4: Core module development (Attendance Tracking)

Week 5-6: Core module development (Personal Cloud Storage)

Week 7-8: User interface design and implementation

Week 9-10: Integration and testing of core modules

Week 11: Documentation and final preparations

Week 12: Final review, presentation preparation, and submission

3. SYSTEM REQUIREMENTS

3.1 Hardware Requirements

The Smart Campus Bot system has been designed to operate efficiently within standard hardware specifications commonly available in educational institutions:

Development Environment:

- Processor: Intel Core i5 or equivalent (minimum), Intel Core i7 or equivalent (recommended)
- RAM: 8GB minimum, 16GB recommended
- Storage: 256GB SSD minimum, 512GB SSD recommended
- Graphics: Integrated graphics capable of 1080p output
- Network: Wi-Fi 5 (802.11ac) or Ethernet connection

Client Environment:

- Processor: Modern dual-core processor or better
- RAM: 4GB minimum, 8GB recommended
- Storage: 100MB available space for application files
- Browser: Latest versions of Chrome, Firefox, Safari, or Edge
- Network: Internet connection for initial loading (not required for operation)

The hardware requirements have been optimized to ensure accessibility while maintaining performance standards. The system is designed to run entirely in the browser without requiring specialized hardware or server infrastructure.

3.2 Software Requirements

Development Stack:

- Operating System: Windows 10/11, macOS, or Linux distribution

- Programming Languages: JavaScript (ES6+), HTML5, CSS3
- Development Tools: Visual Studio Code, Git, Browser Developer Tools
- Testing Tools: Manual testing in multiple browsers
- Design Tools: Figma or Adobe XD for UI/UX design

Runtime Environment:

- Web Browser: Latest versions of Chrome, Firefox, Safari, Edge
- Browser Features: IndexedDB support, File API, Drag and Drop API
- Mobile Compatibility: Responsive design supporting iOS and Android devices
- Offline Support: Service workers for offline functionality (planned)

Dependencies:

- pdf.js: Version 2.10.377 (for PDF processing in Attendance module)
- tesseract.js: Version 2 (for OCR functionality in Attendance module)
- Canvas API: Built-in browser support for data visualization

All software components are open-source or available under academic licenses, ensuring compliance with institutional policies. The technology stack was selected based on its relevance to web development applications and its alignment with industry standards.

The system is designed to be self-contained with no external dependencies after initial loading, making it suitable for deployment in environments with restricted internet access.

4. PROJECT MODULES

4.1 Attendance Module

Description: The Attendance Module is designed to provide a comprehensive solution for tracking and managing student attendance records. It serves as a digital replacement for traditional paper-based attendance systems, offering enhanced functionality and accessibility.

Implementation Steps:

1. Design of user interface for both student and administrative views
2. Implementation of data storage using browser's localStorage
3. Development of manual entry functionality for direct input
4. Integration of file processing capabilities for batch uploads
5. Implementation of search and filtering functionality
6. Development of export features for data sharing
7. Testing and validation across different browsers and devices

Code Implementation:

HTML Structure:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Attendance - Smart Service Campus Bot</title>
    <link rel="stylesheet" href="../../css/global.css">
    <link rel="stylesheet" href="attendance.css">
    <script src="https://cdnjs.cloudflare.com/ajax/libs/pdf.js/2.10.377/pdf.min.js"></script>
    <script src='https://cdn.jsdelivr.net/npm/tesseract.js@2/dist/tesseract.min.js'></script>
</head>
<body class="attendance-module">
    <header>
        <h1>Attendance Tracker</h1>
        <a href="../../dashboard.html" class="back-link">Back to Dashboard</a>
    </header>
```

```
<main>
    <div id="user-view">
        <section id="attendance-input">
            <h2>Check Your Attendance</h2>
            <input type="text" id="roll-number-search" placeholder="Enter your Roll Number...">
            <div id="attendance-display"></div>
        </section>
    </div>
    <div id="admin-view" style="display: none;">
        <section id="manual-entry">
            <h2>Manual Attendance Entry</h2>
            <form id="manual-entry-form">
                <input type="text" id="roll-number" placeholder="Roll Number" required>
                <input type="date" id="date" required>
                <select id="status" required>
                    <option value="present">Present</option>
                    <option value="absent">Absent</option>
                </select>
                <button type="submit">Add Record</button>
            </form>
        </section>
        <section id="file-upload">
            <h2>Upload Attendance File</h2>
            <p>Upload a CSV, PDF, or image file.</p>
            <input type="file" id="attendance-file" accept=".csv, .pdf, image/*">
            <button id="process-file-btn">Process File</button>
            <div id="file-processing-status"></div>
        </section>
        <section id="consolidated-records">
            <h2>Consolidated Records</h2>
            <div class="table-controls">
                <input type="text" id="record-search" placeholder="Search by Roll Number...">
```

```
            <button id="export-csv-btn">Export to CSV</button>
        </div>
        <div class="table-wrapper">
            <table id="attendance-table">
                <thead>
                    <tr>
                        <th>Roll Number</th>
                        <th>Date</th>
                        <th>Status</th>
                    </tr>
                </thead>
                <tbody>
                    <!-- Records will be populated here -->
                </tbody>
            </table>
        </div>
    </section>
  </div>
 </main>
 <script src="attendance.js"></script>
</body>
</html>
```

CSS Styles:

```
/* Attendance Module - Time Tracking & Academic Interface */
/* Clock and time-based animations */
@keyframes clockTick { 0% { transform: rotate(0deg); } 100% { transform: rotate(360deg); } }
@keyframes attendanceRipple { 0% { transform: scale(0); opacity: 1; } 100% { transform: scale(4); opacity: 0; } }
@keyframes statusPulse { 0%, 100% { box-shadow: 0 0 0 0 rgba(34, 139, 34, 0.7); } 70% { box-shadow: 0 0 0 10px rgba(34, 139, 34, 0); } }
/* Real-time clock display */
```

.time-display { position: fixed; top: 20px; right: 20px; background: linear-gradient(135deg, rgba(34, 139, 34, 0.2), rgba(0, 212, 255, 0.1)); border: 2px solid #228B22; color: #228B22; padding: 12px 20px; border-radius: 25px; font-family: 'Courier New', monospace; font-weight: bold; z-index: 1000; box-shadow: 0 4px 15px rgba(34, 139, 34, 0.3); animation: statusPulse 2s ease-in-out infinite; }

.time-display::before { content: '🕐'; margin-right: 8px; animation: clockTick 60s linear infinite; display: inline-block; }

/* Academic calendar theme */

.calendar-grid { position: absolute; top: 0; left: 0; width: 100%; height: 100%; background-image: linear-gradient(rgba(34, 139, 34, 0.1) 1px, transparent 1px), linear-gradient(90deg, rgba(34, 139, 34, 0.1) 1px, transparent 1px); background-size: 20px 20px; opacity: 0.3; pointer-events: none; }

header { display: flex; justify-content: space-between; align-items: center; padding: 20px 40px; background: linear-gradient(135deg, rgba(10, 14, 39, 0.95), rgba(34, 139, 34, 0.1)); border-bottom: 2px solid #228B22; position: relative; overflow: hidden; }

header::before { content: ''; position: absolute; top: 0; left: -100%; width: 100%; height: 100%; background: linear-gradient(90deg, transparent, rgba(34, 139, 34, 0.2), transparent); animation: attendanceFlow 3s ease-in-out infinite; }

@keyframes attendanceFlow { 0%, 100% { left: -100%; } 50% { left: 100%; } }

h1 { color: #228B22; text-shadow: 0 0 15px rgba(34, 139, 34, 0.5); font-weight: 300; display: flex; align-items: center; gap: 15px; }

h1::before { content: '📋'; font-size: 0.8em; animation: attendanceIconBounce 2s ease-in-out infinite; }

@keyframes attendanceIconBounce { 0%, 100% { transform: translateY(0); } 50% { transform: translateY(-5px); } }

.back-link { color: #228B22; text-decoration: none; border: 1px solid #228B22; padding: 10px 15px; border-radius: 25px; transition: all 0.3s ease; position: relative; overflow: hidden; }

.back-link::before { content: ''; position: absolute; top: 0; left: -100%; width: 100%; height: 100%; background: rgba(34, 139, 34, 0.3); transition: left 0.3s ease; z-index: -1; }

.back-link:hover::before { left: 0; }

.back-link:hover { background: #228B22; color: var(--background-color); box-shadow: 0 0 20px rgba(34, 139, 34, 0.5); transform: translateY(-2px); }

main { padding: 40px; background: radial-gradient(circle at 30% 70%, rgba(34, 139, 34, 0.05) 0%, transparent 50%); position: relative; }

main::before { content: ''; position: absolute; top: 0; left: 0; width: 100%; height: 100%; background-image: linear-gradient(rgba(34, 139, 34, 0.05) 1px, transparent 1px), linear-gradient(90deg, rgba(34, 139, 34, 0.05) 1px, transparent 1px); background-size: 30px 30px; opacity: 0.3; pointer-events: none; }

/* Enhanced attendance entry forms with academic styling */

#manual-entry-form, #file-upload { margin-bottom: 40px; padding: 25px; background: linear-gradient(135deg, rgba(34, 139, 34, 0.1), rgba(0, 212, 255, 0.05)); border: 2px solid rgba(34, 139, 34, 0.3); border-radius: 15px; position: relative; overflow: hidden; box-shadow: 0 10px 30px rgba(0, 0, 0, 0.2); }

#manual-entry-form::before, #file-upload::before { content: ''; position: absolute; top: 0; left: 0; right: 0; height: 3px; background: linear-gradient(90deg, #228B22, #32CD32, #228B22); background-size: 200% 100%; animation: formHeaderFlow 3s ease-in-out infinite; }

@keyframes formHeaderFlow { 0%, 100% { background-position: 0% 50%; } 50% { background-position: 100% 50%; } }

#manual-entry-form input, #manual-entry-form select, #file-upload input, #roll-number-search { margin-right: 10px; padding: 12px 15px; background: rgba(34, 139, 34, 0.1); border: 1px solid rgba(34, 139, 34, 0.3); border-radius: 10px; color: var(--text-color); transition: all 0.3s ease; box-shadow: inset 0 2px 10px rgba(0, 0, 0, 0.1); }

#manual-entry-form input:focus, #manual-entry-form select:focus, #file-upload input:focus, #roll-number-search:focus { border-color: #228B22; box-shadow: inset 0 2px 10px rgba(0, 0, 0, 0.1), 0 0 20px rgba(34, 139, 34, 0.3); background: rgba(34, 139, 34, 0.15); }

#manual-entry-form button, #process-file-btn { padding: 12px 24px; background: linear-gradient(135deg, #228B22, #32CD32); border: none; border-radius: 10px; color: white; cursor: pointer; font-weight: 600; text-transform: uppercase; letter-spacing: 1px; transition: all 0.3s ease; box-shadow: 0 5px 15px rgba(34, 139, 34, 0.3); position: relative; overflow: hidden; }

#manual-entry-form button::before, #process-file-btn::before { content: ''; position: absolute; top: 0; left: -100%; width: 100%; height: 100%; background: linear-gradient(90deg, transparent, rgba(255, 255, 255, 0.3), transparent); transition: left 0.5s ease; }

#manual-entry-form button:hover::before, #process-file-btn:hover::before { left: 100%; }

#manual-entry-form button:hover, #process-file-btn:hover { transform: translateY(-3px); box-shadow: 0 8px 25px rgba(34, 139, 34, 0.4); }

/* Enhanced attendance display with status animations */

#attendance-display { margin-top: 20px; background: linear-gradient(135deg, rgba(10, 14, 39, 0.9), rgba(34, 139, 34, 0.1)); border: 2px solid #228B22; padding: 30px; border-radius: 15px; position: relative; overflow: hidden; box-shadow: 0 10px 30px rgba(0, 0, 0, 0.3), inset 0 1px 0 rgba(255, 255, 255, 0.1); }

#attendance-display::before { content: ''; position: absolute; top: 0; left: 0; right: 0; height: 3px; background: linear-gradient(90deg, #228B22, #32CD32, #228B22); }

/* Enhanced attendance table with status indicators */

#attendance-table { width: 100%; border-collapse: collapse; position: relative; }

#attendance-table th, #attendance-table td { padding: 15px; border-bottom: 1px solid rgba(34, 139, 34, 0.2); text-align: left; transition: all 0.3s ease; }

#attendance-table th { background: linear-gradient(135deg, rgba(10, 14, 39, 0.95), rgba(34, 139, 34, 0.1)); position: sticky; top: 0; color: #228B22; font-weight: 600; text-transform: uppercase; letter-spacing: 1px; }

#attendance-table tr { transition: all 0.3s ease; }

#attendance-table tr:hover { background: rgba(34, 139, 34, 0.1); transform: translateX(5px); }

/* Status indicators with animations */

.status-present { color: #32CD32; font-weight: bold; position: relative; }

.status-present::before { content: '✓'; margin-right: 5px; animation: checkMarkPulse 2s ease-in-out infinite; }

@keyframes checkMarkPulse { 0%, 100% { color: #32CD32; text-shadow: 0 0 5px rgba(50, 205, 50, 0.5); } 50% { color: #228B22; text-shadow: 0 0 10px rgba(34, 139, 34, 0.8); } }

.status-absent { color: #FF6B6B; font-weight: bold; position: relative; }

.status-absent::before { content: '✗'; margin-right: 5px; animation: crossPulse 2s ease-in-out infinite; }

@keyframes crossPulse { 0%, 100% { color: #FF6B6B; text-shadow: 0 0 5px rgba(255, 107, 107, 0.5); } 50% { color: #FF4757; text-shadow: 0 0 10px rgba(255, 71, 87, 0.8); } }

#consolidated-records { margin-top: 40px; }

.table-controls { display: flex; justify-content: space-between; align-items: center; margin-bottom: 20px; }

#record-search { width: 60%; }

#export-csv-btn { padding: 12px 24px; background: linear-gradient(135deg, #32CD32, #228B22); border: none; border-radius: 10px; color: white; cursor: pointer; font-weight: 600; text-transform: uppercase; letter-spacing: 1px; transition: all 0.3s ease; box-shadow: 0 5px 15px rgba(50, 205, 50, 0.3); position: relative; overflow: hidden; }

#export-csv-btn::before { content: '📊'; margin-right: 8px; }

#export-csv-btn:hover { transform: translateY(-3px); box-shadow: 0 8px 25px rgba(50, 205, 50, 0.4); }

.table-wrapper { max-height: 400px; overflow-y: auto; border: 1px solid var(--accent-color); border-radius: 5px; }

#attendance-table { width: 100%; border-collapse: collapse; }

#attendance-table th, #attendance-table td { padding: 12px 15px; border-bottom: 1px solid rgba(0, 212, 255, 0.2); text-align: left; }

#attendance-table th { background-color: rgba(10, 14, 39, 0.9); position: sticky; top: 0; }

.status-select { width: 100%; padding: 10px; background: linear-gradient(135deg, #0c102c, rgba(34, 139, 34, 0.1)); border: 1px solid #228B22; color: var(--text-color); border-radius: 8px; transition: all 0.3s ease; cursor: pointer; }

.status-select:focus { border-color: #32CD32; box-shadow: 0 0 15px rgba(34, 139, 34, 0.3); background: linear-gradient(135deg, #0c102c, rgba(34, 139, 34, 0.15)); }

/* Export and control buttons */

#export-csv-btn { padding: 12px 24px; background: linear-gradient(135deg, #32CD32, #228B22); border: none; border-radius: 10px; color: white; cursor: pointer; font-weight: 600; text-transform: uppercase; letter-spacing: 1px; transition: all 0.3s ease; box-shadow: 0 5px 15px rgba(50, 205, 50, 0.3); position: relative; overflow: hidden; }

#export-csv-btn::before { content: '📊'; margin-right: 8px; }

#export-csv-btn:hover { transform: translateY(-3px); box-shadow: 0 8px 25px rgba(50, 205, 50, 0.4); }

/* Real-time attendance stats */

.attendance-stats { display: flex; justify-content: space-around; margin: 20px 0; padding: 20px; background: rgba(34, 139, 34, 0.1); border-radius: 15px; border: 1px solid rgba(34, 139, 34, 0.3); }

.stat-item { text-align: center; position: relative; }

.stat-number { font-size: 2em; font-weight: bold; color: #228B22; display: block; animation: numberCounter 1s ease-out; }

.stat-label { color: var(--light-text-color); font-size: 0.9em; text-transform: uppercase; letter-spacing: 1px; }

@keyframes numberCounter { from { opacity: 0; transform: translateY(20px); } to { opacity: 1; transform: translateY(0); } }

JavaScript Implementation:

```
document.addEventListener('DOMContentLoaded', () => {
    const userView = document.getElementById('user-view');
    const adminView = document.getElementById('admin-view');
    const rollNumberSearch = document.getElementById('roll-number-search');
    const attendanceDisplay = document.getElementById('attendance-display');
    const manualEntryForm = document.getElementById('manual-entry-form');
    const attendanceFile = document.getElementById('attendance-file');
    const processFileBtn = document.getElementById('process-file-btn');
    const fileProcessingStatus = document.getElementById('file-processing-status');
    let attendanceData = JSON.parse(localStorage.getItem('attendance-data')) || {};
    const urlParams = new URLSearchParams(window.location.search);
    const isAdminView = urlParams.get('view') === 'admin';
    const recordSearchInput = document.getElementById('record-search');
    const attendanceTableBody = document.querySelector('#attendance-table tbody');
    if (isAdminView) {
        userView.style.display = 'none';
        adminView.style.display = 'block';
        document.querySelector('.back-link').href = '../../admin.html';
        document.querySelector('h1').textContent = 'Manage Attendance';
        renderAttendanceTable();
    } else {
        userView.style.display = 'block';
        adminView.style.display = 'none';
    }
    if(rollNumberSearch){
        rollNumberSearch.addEventListener('input', (e) => {
            const rollNumber = e.target.value;
            if (attendanceData[rollNumber]) {
                const records = attendanceData[rollNumber];
                const presentCount = records.filter(r => r.status === 'present').length;
```

```
            const totalCount = records.length;
            const percentage = totalCount > 0 ? (presentCount / totalCount) * 100 : 0;
            attendanceDisplay.innerHTML = `
                <h3>Attendance for ${rollNumber}</h3>
                <p>Present: ${presentCount}</p>
                <p>Total Classes: ${totalCount}</p>
                <p>Percentage: ${percentage.toFixed(2)}%</p>
            `;
        } else {
            attendanceDisplay.innerHTML = '<p>No records found for this roll number.</p>';
        }
    });
}
if(manualEntryForm){
    manualEntryForm.addEventListener('submit', (e) => {
        e.preventDefault();
        const rollNumber = document.getElementById('roll-number').value;
        const date = document.getElementById('date').value;
        const status = document.getElementById('status').value;
        if (!attendanceData[rollNumber]) {
            attendanceData[rollNumber] = [];
        }
        attendanceData[rollNumber].push({ date, status });
        localStorage.setItem('attendance-data', JSON.stringify(attendanceData));
        alert('Record added successfully.');
        manualEntryForm.reset();
    });
}
if(processFileBtn){
    processFileBtn.addEventListener('click', () => {
        const file = attendanceFile.files[0];
        if (!file) {
```

```
            alert('Please select a file.');
            return;
        }
        fileProcessingStatus.textContent = 'Processing...';
        const fileType = file.type;
        if (fileType === 'text/csv') {
            parseCSV(file);
        } else if (fileType === 'application/pdf') {
            parsePDF(file);
        } else if (fileType.startsWith('image/')) {
            parseImage(file);
        } else {
            alert('Unsupported file type.');
            fileProcessingStatus.textContent = '';
        }
    });
}
function parseCSV(file) {
    const reader = new FileReader();
    reader.onload = (e) => {
        const text = e.target.result;
        const rows = text.split('\\n').slice(1); // Skip header
        rows.forEach(row => {
            const [rollNumber, date, status] = row.split(',');
            if (rollNumber && date && status) {
                if (!attendanceData[rollNumber]) {
                    attendanceData[rollNumber] = [];
                }
                attendanceData[rollNumber].push({ date, status: status.trim() });
            }
        });
        localStorage.setItem('attendance-data', JSON.stringify(attendanceData));
```

```
            fileProcessingStatus.textContent = 'CSV processed successfully.';
        };
        reader.readAsText(file);
    }
    function parsePDF(file) {
        const reader = new FileReader();
        reader.onload = (e) => {
            const data = new Uint8Array(e.target.result);
            pdfjsLib.GlobalWorkerOptions.workerSrc =
`https://cdnjs.cloudflare.com/ajax/libs/pdf.js/2.10.377/pdf.worker.min.js`;
            pdfjsLib.getDocument(data).promise.then(pdf => {
                let textContent = '';
                const numPages = pdf.numPages;
                let promises = [];
                for (let i = 1; i <= numPages; i++) {
                    promises.push(pdf.getPage(i).then(page => page.getTextContent()));
                }
                Promise.all(promises).then(contents => {
                    contents.forEach(content => {
                        content.items.forEach(item => {
                            textContent += item.str + ' ';
                        });
                    });
                    console.log('PDF Text:', textContent);
                    alert('PDF parsing is for demonstration. Check console for extracted text.');
                    fileProcessingStatus.textContent = 'PDF processed.';
                });
            });
        };
        reader.readAsArrayBuffer(file);
    }
    function parseImage(file) {
```

```
    Tesseract.recognize(
        file,
        'eng',
        { logger: m => {
            fileProcessingStatus.textContent = `${m.status}: ${Math.round(m.progress * 100)}%`;
            console.log(m)
        } }
    ).then(({ data: { text } }) => {
        console.log('OCR Text:', text);
        alert('Image OCR is for demonstration. Check console for extracted text.');
        fileProcessingStatus.textContent = 'Image processed.';
    });
}
function renderAttendanceTable() {
    if (!attendanceTableBody) return;

    const searchTerm = recordSearchInput.value.toLowerCase();
    let flatData = [];
    for (const rollNo in attendanceData) {
        attendanceData[rollNo].forEach((record, index) => {
            flatData.push({ rollNo, ...record, id: `${rollNo}-${index}` });
        });
    }
    const filteredData = flatData.filter(record => record.rollNo.toLowerCase().includes(searchTerm));
    attendanceTableBody.innerHTML = '';
    filteredData.forEach(record => {
        const row = attendanceTableBody.insertRow();
        row.innerHTML = `
            <td>${sanitizeInput(record.rollNo)}</td>
            <td>${sanitizeInput(record.date)}</td>
            <td>
                <select class="status-select" data-id="${record.id}">
```

```
                <option value="present" ${record.status === 'present' ? 'selected' : ''}>Present</option>
                <option value="absent" ${record.status === 'absent' ? 'selected' : ''}>Absent</option>
            </select>
        </td>
    `;
  });
}
if (recordSearchInput) {
  recordSearchInput.addEventListener('input', renderAttendanceTable);
}
const exportCsvBtn = document.getElementById('export-csv-btn');
if (attendanceTableBody) {
  attendanceTableBody.addEventListener('change', (e) => {
    if (e.target.classList.contains('status-select')) {
      const selectEl = e.target;
      const recordId = selectEl.dataset.id;
      const newStatus = selectEl.value;
      const [rollNo, recordIndex] = recordId.split('-');
      if (attendanceData[rollNo] && attendanceData[rollNo][recordIndex]) {
        attendanceData[rollNo][recordIndex].status = newStatus;
        localStorage.setItem('attendance-data', JSON.stringify(attendanceData));
      }
    }
  });
}
if (exportCsvBtn) {
  exportCsvBtn.addEventListener('click', () => {
    const headers = ['Roll Number', 'Date', 'Status'];
    let csvContent = headers.join(',') + '\\n';
    for (const rollNo in attendanceData) {
      attendanceData[rollNo].forEach(record => {
        const row = [rollNo, record.date, record.status];
```

```
            csvContent += row.join(',') + '\\n';
        });
    }
    const blob = new Blob([csvContent], { type: 'text/csv;charset=utf-8;' });
    const link = document.createElement('a');
    if (link.download !== undefined) {
        const url = URL.createObjectURL(blob);
        link.setAttribute('href', url);
        link.setAttribute('download', 'attendance_report.csv');
        link.style.visibility = 'hidden';
        document.body.appendChild(link);
        link.click();
        document.body.removeChild(link);
    }
  });
 }
});
```

Challenges Faced:

- Implementing cross-browser compatibility for file processing features
- Handling different file formats (CSV, PDF, images) with a unified interface
- Ensuring data persistence across browser sessions
- Optimizing performance for large datasets in the browser

4.2 Storage Module

Description: The Storage Module provides a personal cloud storage solution for students and faculty to upload, manage, and share files. It utilizes browser-based storage to provide a seamless file management experience without requiring external servers.

Implementation Steps:

1. Design of user interface for both student and administrative views
2. Implementation of IndexedDB for client-side file storage
3. Development of drag-and-drop upload functionality
4. Integration of file search and filtering capabilities
5. Implementation of file download and deletion features
6. Development of administrative dashboard with usage statistics
7. Testing and validation across different browsers and devices

Code Implementation:

HTML Structure:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Storage - Smart Service Campus Bot</title>
    <link rel="stylesheet" href="../../css/global.css">
    <link rel="stylesheet" href="storage.css">
    <link rel="stylesheet" href="../../css/responsive.css">
</head>
<body>
    <header>
        <h1>Personal Cloud Storage</h1>
        <a href="../../dashboard.html" class="back-link">Back to Dashboard</a>
    </header>

    <main>
        <div id="user-view">
            <div id="drop-zone">
                <p>Drag & Drop files here or click to upload</p>
```

```
        <input type="file" id="file-input" multiple style="display: none;">
    </div>
    <input type="text" id="search-storage" placeholder="Search files...">
    <div id="file-list"></div>
</div>

<div id="admin-view" style="display: none;">
    <div class="admin-section">
        <h2>Storage Overview</h2>
        <div class="stats-grid">
            <div>Total Usage: <strong id="total-usage">-- MB</strong></div>
            <div>Total Files: <strong id="total-files">--</strong></div>
            <div>Total Capacity: <strong>50 MB (Simulated)</strong></div>
        </div>
    </div>

    <div class="admin-section">
        <h2>Top Users by Storage</h2>
        <canvas id="top-users-chart" width="400" height="200"></canvas>
    </div>

    <div class="admin-section">
        <h2>All Stored Files</h2>
        <div class="table-wrapper">
            <table id="all-files-table">
                <thead>
                    <tr>
                        <th>Filename</th>
                        <th>Size (KB)</th>
                        <th>Uploaded By</th>
                        <th>Actions</th>
                    </tr>
```

```
                    </thead>
                    <tbody>
                        <!-- File data will be populated here -->
                    </tbody>
                </table>
            </div>
        </div>
    </div>
</main>

<script src="storage.js"></script>
</body>
</html>
```

CSS Styles:

```
/* Storage Module - Cloud File Management Interface */

/* Cloud and file transfer animations */

@keyframes cloudFloat { 0%, 100% { transform: translateY(0px); } 50% { transform: translateY(-10px); } }

@keyframes fileUpload { 0% { transform: translateY(20px); opacity: 0; } 50% { transform: translateY(-5px); opacity: 0.8; } 100% { transform: translateY(0); opacity: 1; } }

@keyframes dataTransfer { 0% { transform: translateX(-100px); opacity: 0; } 50% { opacity: 1; } 100% { transform: translateX(100px); opacity: 0; } }

@keyframes folderPulse { 0%, 100% { box-shadow: 0 0 0 0 rgba(54, 159, 255, 0.7); } 70% { box-shadow: 0 0 0 10px rgba(54, 159, 255, 0); } }

/* Cloud storage background */

.cloud-bg { position: fixed; top: 0; left: 0; width: 100%; height: 100%; background: radial-gradient(circle at 20% 30%, rgba(54, 159, 255, 0.1) 0%, transparent 50%), radial-gradient(circle at 80% 70%, rgba(72, 202, 155, 0.1) 0%, transparent 50%); pointer-events: none; z-index: -1; }

/* Floating cloud elements */

.cloud-element { position: absolute; color: rgba(54, 159, 255, 0.2); font-size: 2em; animation: cloudFloat 4s ease-in-out infinite; pointer-events: none; }

.cloud-element:nth-child(1) { top: 10%; left: 10%; animation-delay: 0s; }

.cloud-element:nth-child(2) { top: 20%; right: 15%; animation-delay: 1s; }

.cloud-element:nth-child(3) { bottom: 25%; left: 20%; animation-delay: 2s; }
```

header { display: flex; justify-content: space-between; align-items: center; padding: 20px 40px; background: linear-gradient(135deg, rgba(10, 14, 39, 0.95), rgba(54, 159, 255, 0.1)); border-bottom: 2px solid #369fff; position: relative; overflow: hidden; }

header::before { content: ''; position: absolute; top: 0; left: -100%; width: 100%; height: 100%; background: linear-gradient(90deg, transparent, rgba(54, 159, 255, 0.2), transparent); animation: cloudFlow 4s ease-in-out infinite; }

@keyframes cloudFlow { 0%, 100% { left: -100%; } 50% { left: 100%; } }

h1 { color: #369fff; text-shadow: 0 0 15px rgba(54, 159, 255, 0.5); font-weight: 300; display: flex; align-items: center; gap: 15px; }

h1::before { content: '☁'; font-size: 0.8em; animation: cloudBounce 3s ease-in-out infinite; }

@keyframes cloudBounce { 0%, 100% { transform: translateY(0) scale(1); } 50% { transform: translateY(-5px) scale(1.1); } }

.back-link { color: #369fff; text-decoration: none; border: 1px solid #369fff; padding: 10px 15px; border-radius: 25px; transition: all 0.3s ease; position: relative; overflow: hidden; }

.back-link::before { content: ''; position: absolute; top: 0; left: -100%; width: 100%; height: 100%; background: rgba(54, 159, 255, 0.3); transition: left 0.3s ease; z-index: -1; }

.back-link:hover::before { left: 0; }

.back-link:hover { background: #369fff; color: var(--background-color); box-shadow: 0 0 20px rgba(54, 159, 255, 0.5); transform: translateY(-2px); }

main { padding: 40px; background: radial-gradient(circle at 30% 20%, rgba(54, 159, 255, 0.05) 0%, transparent 50%), radial-gradient(circle at 70% 80%, rgba(72, 202, 155, 0.05) 0%, transparent 50%); position: relative; }

main::before { content: ''; position: absolute; top: 0; left: 0; width: 100%; height: 100%; background-image: repeating-linear-gradient(30deg, transparent, transparent 20px, rgba(54, 159, 255, 0.05) 20px, rgba(54, 159, 255, 0.05) 21px); opacity: 0.3; pointer-events: none; }

/* Enhanced cloud drop zone with upload animations */

#drop-zone { border: 3px dashed #369fff; border-radius: 20px; padding: 50px; text-align: center; cursor: pointer; margin-bottom: 30px; background: radial-gradient(circle at center, rgba(54, 159, 255, 0.1) 0%, transparent 70%), linear-gradient(135deg, rgba(54, 159, 255, 0.05), rgba(72, 202, 155, 0.05)); transition: all 0.3s ease; position: relative; overflow: hidden; box-shadow: 0 10px 30px rgba(0, 0, 0, 0.1); }

#drop-zone::before { content: '☁'; position: absolute; top: 20px; right: 20px; font-size: 2em; opacity: 0.3; animation: cloudFloat 3s ease-in-out infinite; }

#drop-zone::after { content: ''; position: absolute; top: 0; left: -100%; width: 100%; height: 100%; background: linear-gradient(90deg, transparent, rgba(54, 159, 255, 0.2), transparent); transition: left 0.3s ease; }

#drop-zone:hover { border-color: #72caff; background: radial-gradient(circle at center, rgba(54, 159, 255, 0.2) 0%, transparent 70%), linear-gradient(135deg, rgba(54, 159, 255, 0.1), rgba(72, 202, 155, 0.1)); transform: translateY(-5px); box-shadow: 0 15px 40px rgba(54, 159, 255, 0.2); }

#drop-zone:hover::after { left: 100%; }

#drop-zone.dragover { background: radial-gradient(circle at center, rgba(54, 159, 255, 0.3) 0%, transparent 70%), linear-gradient(135deg, rgba(54, 159, 255, 0.15), rgba(72, 202, 155, 0.15)); border-color: #72caff; transform: scale(1.02); animation: folderPulse 1s ease-in-out infinite; }

/* Enhanced search with cloud styling */

```
#search-storage { width: 100%; padding: 15px 20px; background: rgba(54, 159, 255, 0.1); border: 2px solid rgba(54, 159, 255, 0.3); border-radius: 25px; color: var(--text-color); margin-bottom: 30px; transition: all 0.3s ease; font-size: 16px; box-shadow: inset 0 2px 10px rgba(0, 0, 0, 0.1); }

#search-storage:focus { border-color: #369fff; box-shadow: inset 0 2px 10px rgba(0, 0, 0, 0.1), 0 0 25px rgba(54, 159, 255, 0.3); background: rgba(54, 159, 255, 0.15); }

#search-storage::placeholder { color: rgba(54, 159, 255, 0.6); font-style: italic; }

/* Enhanced file list with folder animations */

#file-list { margin-top: 20px; display: grid; gap: 15px; }

.file-item { display: flex; justify-content: space-between; align-items: center; padding: 20px; background: linear-gradient(135deg, rgba(10, 14, 39, 0.9), rgba(54, 159, 255, 0.1)); border: 2px solid rgba(54, 159, 255, 0.3); border-radius: 15px; margin-bottom: 10px; transition: all 0.3s ease; position: relative; overflow: hidden; backdrop-filter: blur(10px); box-shadow: 0 5px 15px rgba(54, 159, 255, 0.2); animation: fileUpload 0.6s ease-out; }

.file-item::before { content: ''; position: absolute; top: 0; left: 0; right: 0; height: 3px; background: linear-gradient(90deg, #369fff, #72caff, #369fff); background-size: 200% 100%; animation: fileHeaderFlow 3s ease-in-out infinite; }

@keyframes fileHeaderFlow { 0%, 100% { background-position: 0% 50%; } 50% { background-position: 100% 50%; } }

.file-item:hover { transform: translateY(-5px) scale(1.02); border-color: #369fff; background: linear-gradient(135deg, rgba(10, 14, 39, 0.95), rgba(54, 159, 255, 0.15)); box-shadow: 0 10px 25px rgba(54, 159, 255, 0.3); }

.file-item .file-name { font-weight: 600; color: #369fff; display: flex; align-items: center; gap: 10px; }

.file-item .file-name::before { content: '📁'; animation: cloudFloat 2s ease-in-out infinite; }

.file-item .file-size { color: var(--light-text-color); font-size: 0.9em; opacity: 0.8; }

.file-item button { background: linear-gradient(135deg, #ff6b6b, #ff4757); color: white; border: none; padding: 8px 15px; border-radius: 8px; cursor: pointer; margin-left: 10px; font-weight: 600; transition: all 0.3s ease; box-shadow: 0 3px 10px rgba(255, 107, 107, 0.3); }

.file-item button:hover { transform: translateY(-2px); box-shadow: 0 5px 15px rgba(255, 107, 107, 0.4); }

.file-item .download-btn { background: linear-gradient(135deg, #48ca9b, #2ed573); box-shadow: 0 3px 10px rgba(72, 202, 155, 0.3); }

.file-item .download-btn:hover { box-shadow: 0 5px 15px rgba(72, 202, 155, 0.4); }

/* Enhanced admin section with cloud dashboard styling */

.admin-section { background: linear-gradient(135deg, rgba(10, 14, 39, 0.9), rgba(54, 159, 255, 0.1)); border: 2px solid #369fff; padding: 30px; border-radius: 15px; margin-bottom: 40px; position: relative; overflow: hidden; box-shadow: 0 10px 30px rgba(0, 0, 0, 0.2); }

.admin-section::before { content: ''; position: absolute; top: 0; left: 0; right: 0; height: 3px; background: linear-gradient(90deg, #369fff, #72caff, #369fff); }

.admin-section h2 { margin-top: 0; border-bottom: 2px solid rgba(54, 159, 255, 0.3); padding-bottom: 15px; margin-bottom: 25px; color: #369fff; font-weight: 300; text-transform: uppercase; letter-spacing: 2px; }

/* Enhanced storage statistics */

.stats-grid { display: flex; justify-content: space-around; text-align: center; font-size: 1.2em; margin-bottom: 30px; padding: 20px; background: rgba(54, 159, 255, 0.1); border-radius: 15px; border: 1px solid rgba(54, 159, 255, 0.3); }

.stats-grid > div { position: relative; padding: 15px; }

.stats-grid strong { display: block; font-size: 2em; color: #369fff; margin-top: 10px; font-weight: bold; animation: numberCounter 1s ease-out; text-shadow: 0 0 10px rgba(54, 159, 255, 0.3); }
```

```
.stats-grid span { color: var(--light-text-color); text-transform: uppercase; letter-spacing: 1px; font-size: 0.9em; }

/* Enhanced file management table */

#all-files-table { width: 100%; border-collapse: collapse; border-radius: 10px; overflow: hidden; box-shadow: 0 10px 30px rgba(0, 0, 0, 0.2); }

#all-files-table th, #all-files-table td { padding: 15px; border-bottom: 1px solid rgba(54, 159, 255, 0.2); text-align: left; transition: all 0.3s ease; }

#all-files-table th { background: linear-gradient(135deg, rgba(10, 14, 39, 0.95), rgba(54, 159, 255, 0.1)); color: #369fff; font-weight: 600; text-transform: uppercase; letter-spacing: 1px; }

#all-files-table tr { transition: all 0.3s ease; }

#all-files-table tr:hover { background: rgba(54, 159, 255, 0.1); transform: translateX(5px); }

@keyframes numberCounter { from { opacity: 0; transform: translateY(20px); } to { opacity: 1; transform: translateY(0); } }
```

JavaScript Implementation:

```
document.addEventListener('DOMContentLoaded', () => {
    // --- DOM Elements ---
    const dropZone = document.getElementById('drop-zone');
    const fileInput = document.getElementById('file-input');
    const fileList = document.getElementById('file-list');
    const searchStorage = document.getElementById('search-storage');
    const adminView = document.getElementById('admin-view');
    const storageStats = document.getElementById('storage-stats');
    const userView = document.getElementById('user-view');

    // --- IndexedDB Setup ---
    let db;
    const dbName = 'fileStorageDB';
    const request = indexedDB.open(dbName, 1);

    /**
     * Handles database errors.
     */
    request.onerror = (event) => {
```

```
    console.error('Database error:', event.target.errorCode);
};

/**
 * Handles database upgrades. This is where the schema is defined.
 */
request.onupgradeneeded = (event) => {
    db = event.target.result;
    // Create an object store for our files.
    // We use 'id' as the keyPath and autoIncrement it.
    db.createObjectStore('files', { keyPath: 'id', autoIncrement: true });
};

/**
 * Handles database connection success.
 */
request.onsuccess = (event) => {
    db = event.target.result;
    if (urlParams.get('view') === 'admin') {
        renderAdminView();
    } else {
        displayFiles();
    }
};

/**
 * Fetches all data and renders the entire admin dashboard.
 */
function renderAdminView() {
    const transaction = db.transaction(['files'], 'readonly');
    const objectStore = transaction.objectStore('files');
    const allFilesRequest = objectStore.getAll();
```

```
allFilesRequest.onsuccess = () => {
    const allFiles = allFilesRequest.result;

    // --- Process Data ---
    let totalSize = 0;
    const userUsage = {};
    allFiles.forEach(file => {
        totalSize += file.size;
        const username = file.username || 'anonymous';
        if (!userUsage[username]) {
            userUsage[username] = 0;
        }
        userUsage[username] += file.size;
    });

    // --- Render Stats ---
    document.getElementById('total-files').textContent = allFiles.length;
    document.getElementById('total-usage').textContent = `${(totalSize / (1024 * 1024)).toFixed(2)} MB`;

    // --- Render File Table ---
    const allFilesTableBody = document.querySelector('#all-files-table tbody');
    if (allFilesTableBody) {
        allFilesTableBody.innerHTML = '';
        allFiles.forEach(file => {
            const row = allFilesTableBody.insertRow();
            row.innerHTML = `
                <td>${sanitizeInput(file.name)}</td>
                <td>${(file.size / 1024).toFixed(2)}</td>
                <td>${sanitizeInput(file.username || 'anonymous')}</td>
                <td><button class="action-btn delete-btn" data-id="${file.id}">Delete</button></td>
            `;
```

```
            });
        }

        // --- Prepare and Render Chart ---
        const sortedUsers = Object.entries(userUsage).sort(([, a], [, b]) => b - a);
        const top5Users = sortedUsers.slice(0, 5);
        const chartLabels = top5Users.map(user => user[0]);
        const chartValues = top5Users.map(user => (user[1] / (1024 * 1024)).toFixed(2)); // Convert to MB

        const chartData = {
            labels: chartLabels,
            values: chartValues
        };
        drawBarChart('top-users-chart', chartData, { barColor: '#4cd137' });
    };

    allFilesRequest.onerror = (event) => {
        console.error('Error fetching all files:', event.target.errorCode);
    };
}

// --- Admin Actions ---
const allFilesTableBody = document.querySelector('#all-files-table tbody');
if (allFilesTableBody) {
    allFilesTableBody.addEventListener('click', (e) => {
        if (e.target.classList.contains('delete-btn')) {
            const fileId = parseInt(e.target.dataset.id, 10);
            if (confirm(`Are you sure you want to delete file ID ${fileId}? This cannot be undone.`)) {
                const transaction = db.transaction(['files'], 'readwrite');
                const objectStore = transaction.objectStore('files');
                const request = objectStore.delete(fileId);
                request.onsuccess = () => {
```

```
                console.log(`File ${fileId} deleted.`);
                renderAdminView(); // Refresh the entire admin view
            };
            request.onerror = (event) => {
                console.error('Error deleting file:', event.target.errorCode);
            };
        }
    }
  });
}
const urlParams = new URLSearchParams(window.location.search);
if (urlParams.get('view') === 'admin') {
  if(userView) userView.style.display = 'none';
  if(adminView) adminView.style.display = 'block';
  document.querySelector('.back-link').href = '../../admin.html';
  document.querySelector('h1').textContent = 'Storage Management';
  // The main logic will be triggered by the DB connection success
}
if(dropZone) {
  dropZone.addEventListener('click', () => fileInput.click());
  dropZone.addEventListener('dragover', (e) => {
    e.preventDefault();
    dropZone.classList.add('dragover');
  });
  dropZone.addEventListener('dragleave', () => dropZone.classList.remove('dragover'));
  dropZone.addEventListener('drop', (e) => {
    e.preventDefault();
    dropZone.classList.remove('dragover');
    const files = e.dataTransfer.files;
    handleFiles(files);
  });
}
```

```
if(fileInput) fileInput.addEventListener('change', () => handleFiles(fileInput.files));
if(searchStorage) searchStorage.addEventListener('input', displayFiles);
function handleFiles(files) {
    const transaction = db.transaction(['files'], 'readwrite');
    const objectStore = transaction.objectStore('files');
    const username = localStorage.getItem('username') || 'anonymous';
    for (const file of files) {
        const fileRecord = {
            name: file.name,
            type: file.type,
            size: file.size,
            data: file,
            username: username
        };
        objectStore.add(fileRecord);
    }
    transaction.oncomplete = () => {
        displayFiles();
    };
}

function displayFiles() {
    if(!db || !fileList) return;
    const searchTerm = searchStorage ? searchStorage.value.toLowerCase() : '';
    fileList.innerHTML = '';
    const objectStore = db.transaction('files').objectStore('files');
    objectStore.openCursor().onsuccess = (event) => {
        const cursor = event.target.result;
        if (cursor) {
            if (cursor.value.name.toLowerCase().includes(searchTerm)) {
                const fileItem = document.createElement('div');
                fileItem.className = 'file-item';
```

```
            fileItem.innerHTML = `
                <span>${cursor.value.name} (${(cursor.value.size / 1024).toFixed(2)} KB)</span>
                <div>
                    <button class="download-btn" data-id="${cursor.key}">Download</button>
                    <button class="delete-btn" data-id="${cursor.key}">Delete</button>
                </div>
            `;
            fileList.appendChild(fileItem);
        }
        cursor.continue();
    } else {
        attachActionListeners();
    }
  };
}

function attachActionListeners() {
  document.querySelectorAll('.delete-btn').forEach(button => {
    button.addEventListener('click', (e) => {
      const id = parseInt(e.target.dataset.id);
      deleteFile(id);
    });
  });
  document.querySelectorAll('.download-btn').forEach(button => {
    button.addEventListener('click', (e) => {
      const id = parseInt(e.target.dataset.id);
      downloadFile(id);
    });
  });
}

function deleteFile(id) {
```

```
    const request = db.transaction(['files'], 'readwrite').objectStore('files').delete(id);
    request.onsuccess = () => displayFiles();
}

function downloadFile(id) {
    const request = db.transaction(['files']).objectStore('files').get(id);
    request.onsuccess = (event) => {
        const fileRecord = event.target.result;
        const link = document.createElement('a');
        link.href = URL.createObjectURL(fileRecord.data);
        link.download = fileRecord.name;
        link.click();
        URL.revokeObjectURL(link.href);
    };
}

function displayStorageStats() {
    if(!db) return;
    let totalSize = 0;
    let fileCount = 0;
    const objectStore = db.transaction('files').objectStore('files');
    objectStore.openCursor().onsuccess = (event) => {
        const cursor = event.target.result;
        if (cursor) {
            fileCount++;
            totalSize += cursor.value.size;
            cursor.continue();
        } else {
            if(storageStats){
                storageStats.innerHTML = `
                    <p>Total Files: ${fileCount}</p>
                    <p>Total Storage Used: ${(totalSize / (1024 * 1024)).toFixed(2)} MB</p>
```

```
            `;
          }
        }
      };
    }
});
```

Challenges Faced:

- Implementing IndexedDB for reliable client-side storage

- Handling large file uploads within browser limitations

- Creating an intuitive drag-and-drop interface

4.3 Future Enhancement Modules

Description: While the current implementation focuses on Attendance and Storage modules, our system architecture supports several planned enhancements that would extend functionality and improve user experience.

Planned Modules:

1. Chatbot Module: An AI-powered chatbot for answering common campus queries

2. Quiz Module: A system for creating and taking online quizzes

3. Study Groups Module: A platform for forming and managing study groups

4. Lost & Found Module: A digital system for reporting and finding lost items

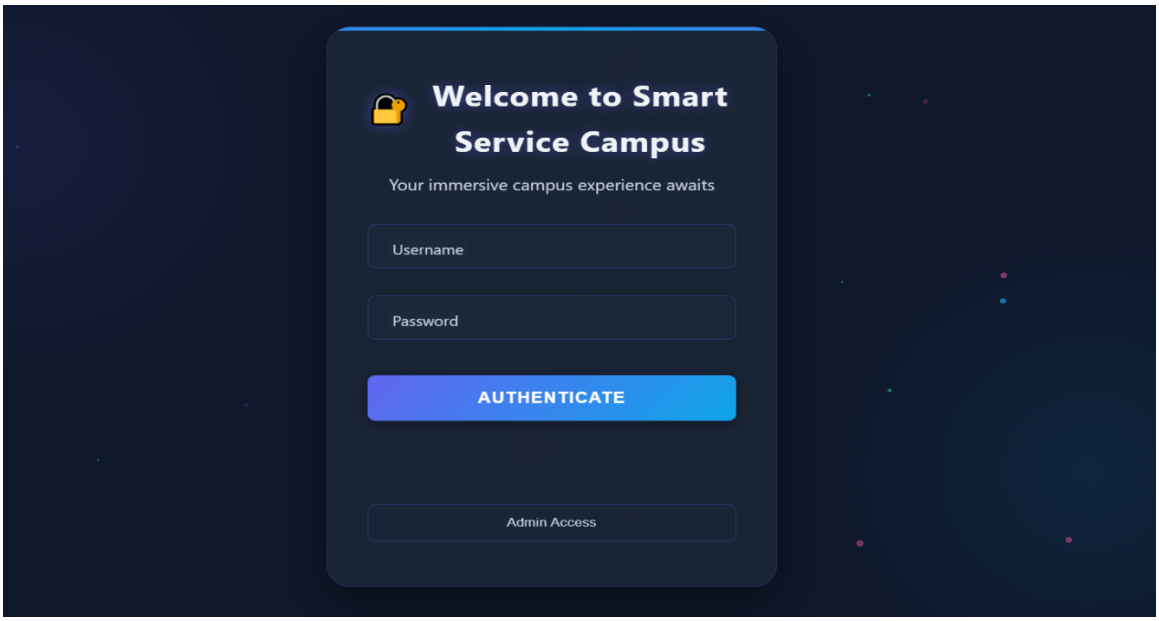
5. Book Exchange Module: A platform for students to exchange textbooks

6. Code Explainer Module: An AI tool for explaining programming concepts

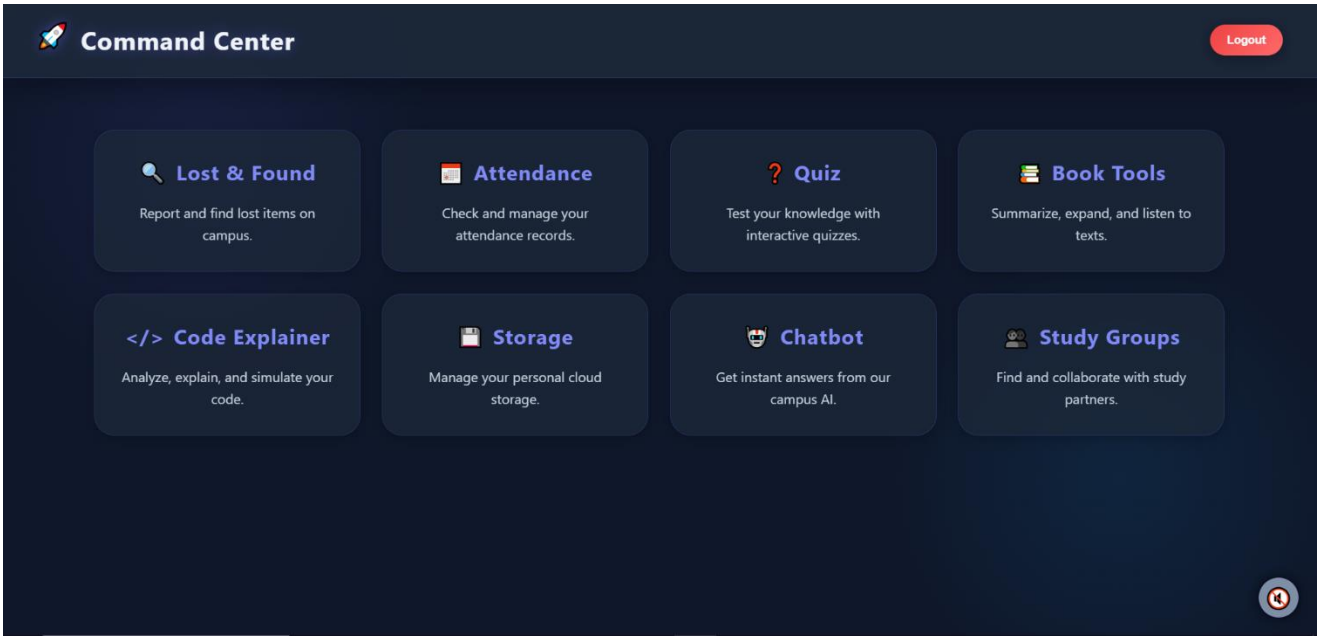
You can see farther improvements or now total source code on this website or GitHub repo : **https://github.com/akhilbehara999/thinkring-project-without-db-integration**

5. OUTPUT SCREENS:

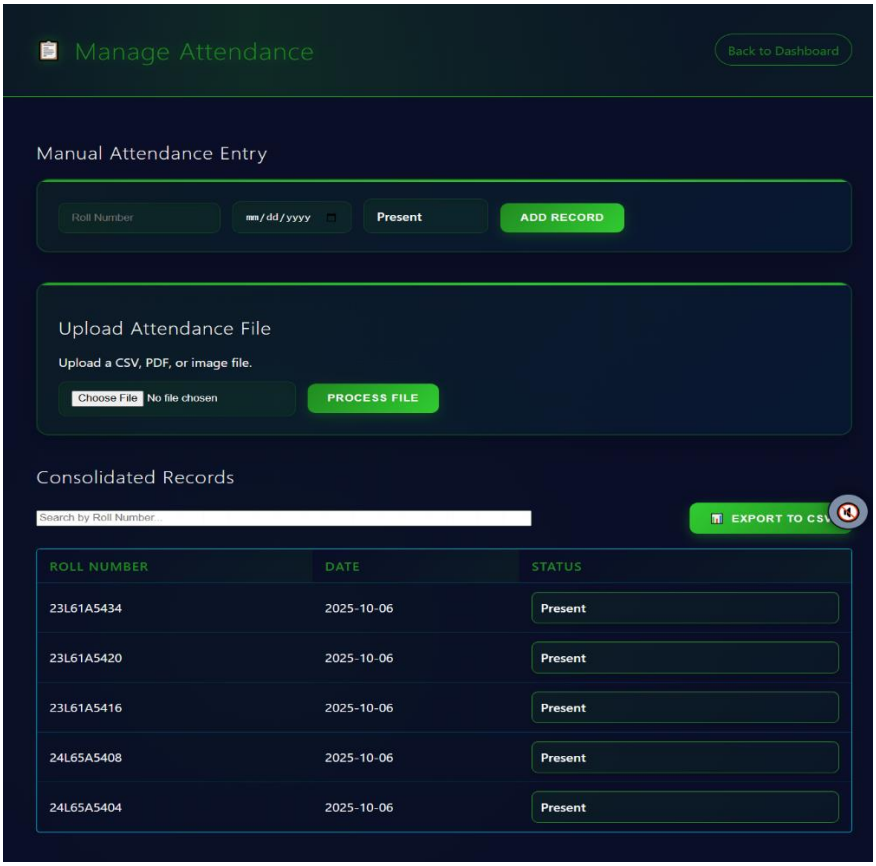
Login screen:



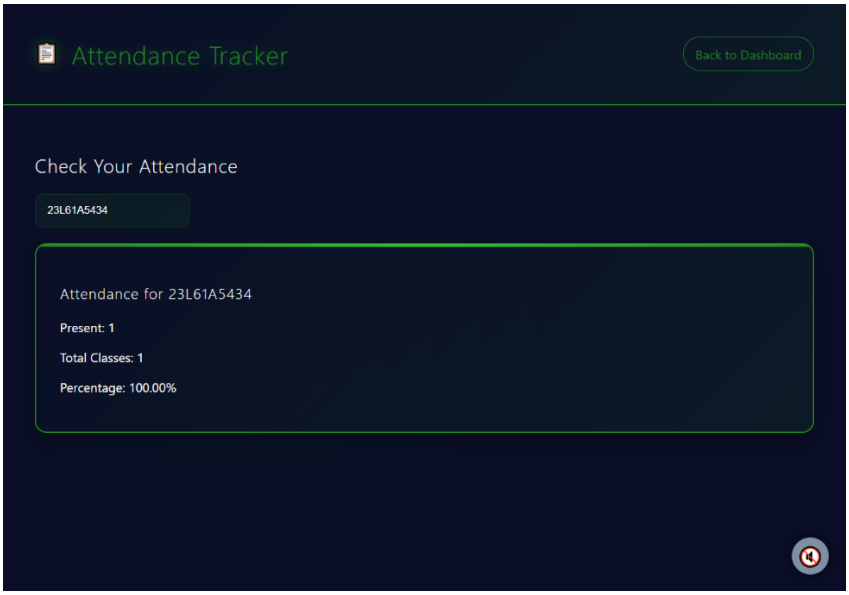
Main menu or Command Center:



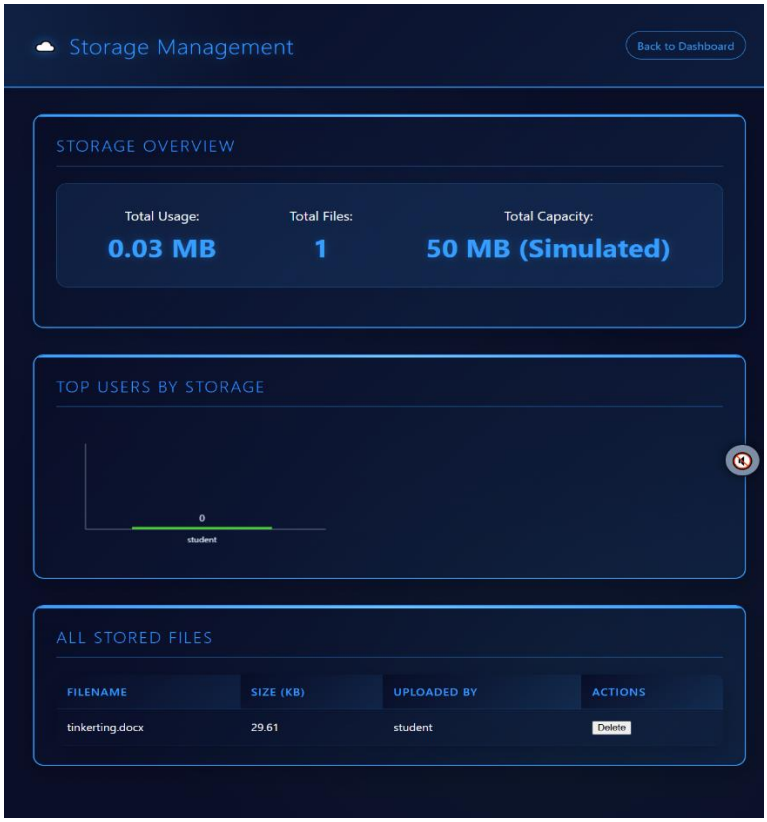
Attendance admin screen:



Attendance user screen:



Storage admin screen:



Storage user screen: