Design Documentation HCI
(Project Report)

Submitted By:

Muhammad Abdul Rehman Wahla 2023-CS-717

Submitted To:
Mr. Usman Ghani

Course: HCI

Department of Computer Science
University of Engineering and Technology Lahore, New Campus

Table of Contents

Design Documentation: Human-Computer Interaction (HCI) Analysis & Interface Strategy

1. Executive Summary & Design Philosophy

The Attendance Management System (AMS) is a web-based application designed to streamline the academic attendance process. Traditional attendance systems often suffer from "feature creep" and aesthetic clutter, leading to slow interaction times. Our solution adopts a **Neo-Brutalist** design philosophy.

Unlike "Flat" or "Material" design, which uses subtle shadows and gradients, **Neo-Brutalism** uses bold geometry, high-contrast outlines (3px solid black), and distinct colors. This is not a stylistic choice but a **functional engineering decision**. It prioritizes **Legibility** and **Affordance** above all else, ensuring that the interface functions as a precision tool for faculty members who need to perform data entry rapidly and accurately.

2. Introduction to Interaction Design

2.1 Definition and Scope

According to Preece, Rogers, and Sharp, Interaction Design (ID) is defined as "designing interactive products to support the way people communicate and interact in their everyday and working lives

The AMS is designed to support the "working life" of a professor. The context of use is often a classroom environment: noisy, time-pressured, and potentially dealing with poor internet connectivity. Therefore, the interaction design must mitigate these environmental stressors by being **robust** and **immediate**.

2.2 User-Centered Design (UCD) Approach

We adopted a standard UCD process, focusing on the specific needs of the primary persona (The Faculty Member). The design is optimized for:

- **High Frequency of Use:** Teachers use this every day.
- **Low Tolerance for Error:** Attendance records are legal documents; accuracy is paramount.
- **Speed:** Taking attendance must not eat into lecture time.

3. User Experience (UX) Framework

We utilized **Hassenzahl's Model of UX** to define our design goals. Hassenzahl distinguishes between two types of quality:

3.1 Pragmatic Quality (Do-Goals)

This refers to the product's perceived ability to support the achievement of "Do-Goals" (tasks).

- *Our Focus:* The AMS is 90% Pragmatic. The interface is designed to be "simple, practical, and obvious." We removed all non-functional animations (like fade-ins or parallax scrolling) because they introduce a latency that hinders the "Do-Goal" of marking attendance.

3.2 Hedonic Quality (Be-Goals)

This refers to the product's ability to stimulate and evoke identity ("Be-Goals").

- *Our Focus:* While functionality is primary, the **Neo-Brutalist** style provides a unique "Hedonic" identity. It feels modern, technical, and slightly rebellious against standard corporate software. This "Technical Aesthetic" appeals to Computer Science faculty, aligning with their self-identity as engineers.

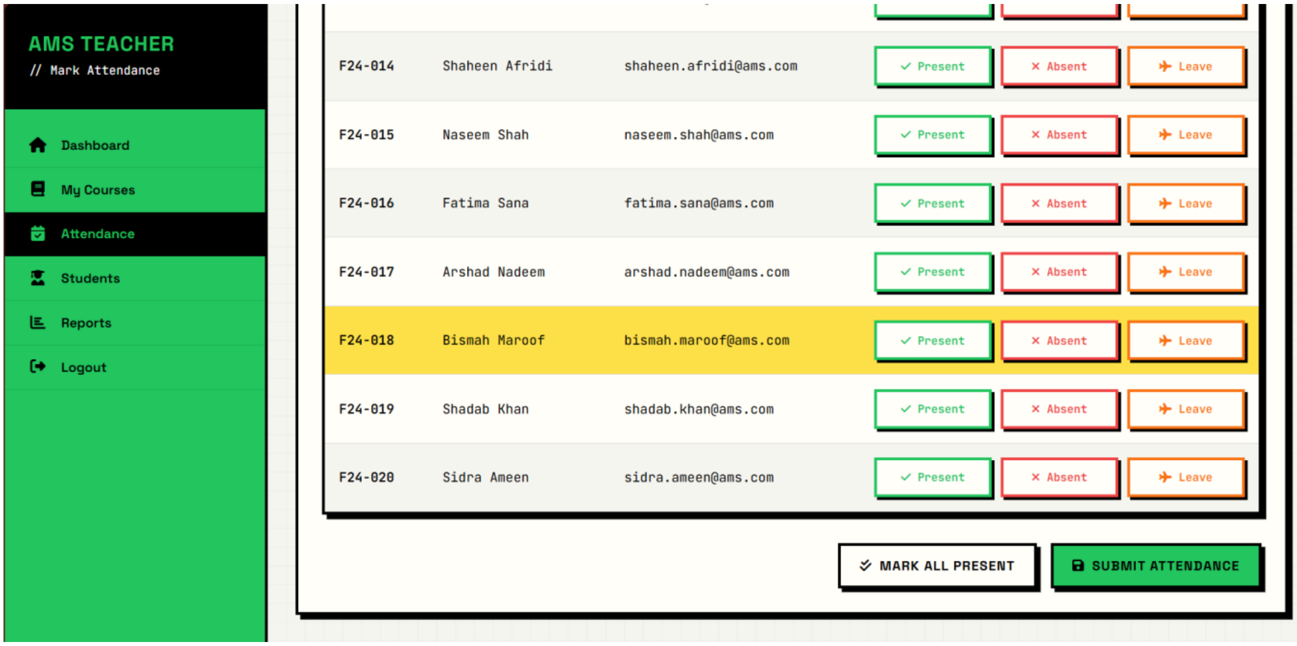
4. Theoretical Framework: Usability Goals

4.1 Efficiency (Time to Completion)

Efficiency is the measure of how fast a user can perform a task once they have learned the interface.

implemented a "Mark All Present" toggle to solve the inefficiency of manual attendance logging. In a class of 60 students, clicking "Present" 60 times is a repetitive burden. This button allows the professor to set a baseline state in one click, leaving them to only toggle the 2-3 absent students. This reduces interaction steps by approximately 95%, transforming a tedious O(n) task into a streamlined O(1) initial action.

The UI supports this with high-contrast color coding—green for "Present" and red for "Absent"—providing immediate visual confirmation of exceptions. On the backend, the system utilizes batch processing to submit the entire roster in a single transaction. This prevents 60 individual API calls, reducing server latency and ensuring the teacher spends less time on administration and more time on instruction.



4.2 Effectiveness (Accuracy)

Effectiveness measures whether the user can complete the task accurately.

- *Design Implementation:* To prevent data entry errors, we used **High-Contrast Binary States**.
    - *State A (Present):* Neon Green Background with Black Text.
    - *State B (Absent):* Bright Red Background with Black Text.
    - Unlike subtle gray-scale toggles, this drastic color change makes it visually impossible to mistake a student's status, ensuring high data accuracy.

4.3 Safety (Error Prevention)

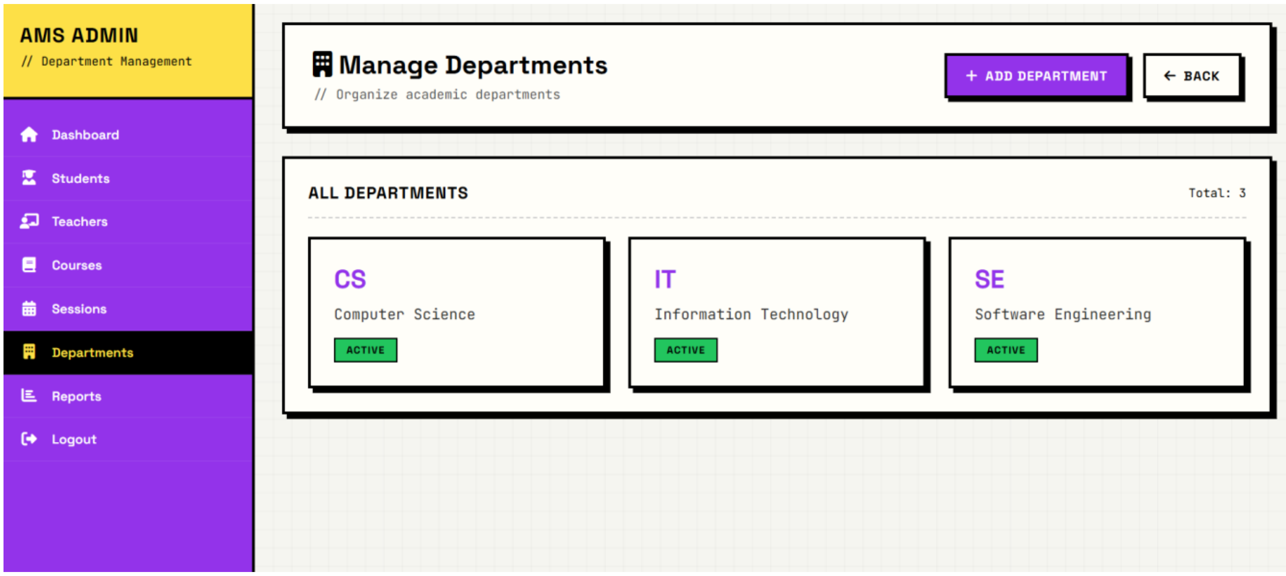
Safety involves protecting the user from dangerous conditions.

- *Design Implementation:* We focused on **Data Safety**. The "Delete Class Record" button is placed in a separate "Danger Zone" area and requires a **Confirmation Modal** ("Are you sure?"). This prevents accidental data loss due to "slips" (unconscious motor errors).

4.4 Learnability

Learnability is the ease with which new users can begin effective interaction.

- *Design Implementation:* The system uses the **"Card Deck" Metaphor**. Courses are displayed as index cards. Because users interact with physical cards in the real world (picking them up, reading the label), they intuitively understand how to interact with the digital cards (clicking them).



5. Cognitive Engineering & Mental Models

The design is grounded in cognitive psychology principles regarding how humans process information.

5.1 Attention and Perception (Gestalt Principles)

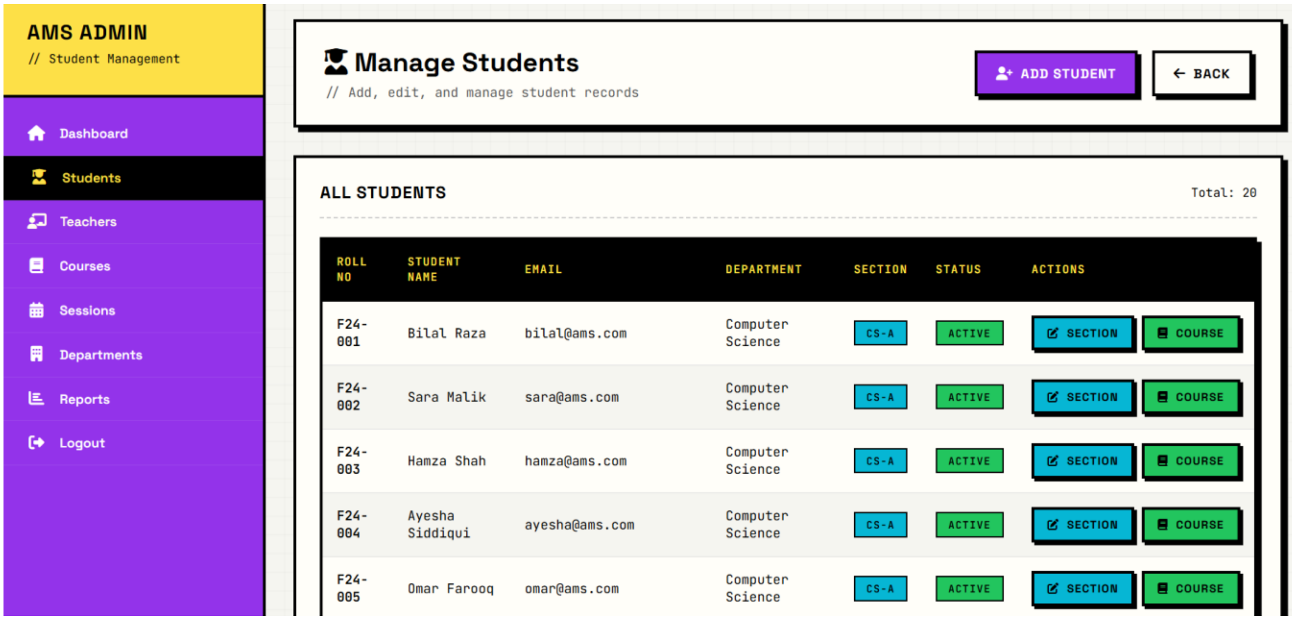
Users scan interfaces; they do not read them. To facilitate rapid scanning, we applied **Gestalt Laws**:

- **Law of Proximity:** Objects near each other are perceived as a group.
  - *Application:* We clustered the Student Name, Roll Number, and Status Button tightly together. We added extra margin between rows. This ensures the eye tracks horizontally across the row without "drifting" to the student below.
- **Law of Closure:** The eye tends to complete incomplete shapes.
  - *Application:* We used fully enclosed borders (boxes) for every data element. This creates a "Closed Region" that traps the eye, keeping focus contained within the specific data point being edited.
- **Figure-Ground Relationship:** The eye differentiates an object from its background.
  - *Application:* The interface uses absolute **Black (#000)** on absolute **White (#FFF).** This maximum contrast ratio ensures the "Figure" (Content) never blends into the "Ground," reducing eye strain during early morning or late evening classes.

5.2 Memory: Recognition vs. Recall

- *Concept:* Recall (retrieving info from memory) is hard. Recognition (seeing info and identifying it) is easy.

- *Application:* The system never forces the user to recall a Student ID or Course Code. All options are presented as visible lists or cards. The user creates a mental mapping through **Recognition**, drastically reducing Cognitive Load.



5.3 Mental Models

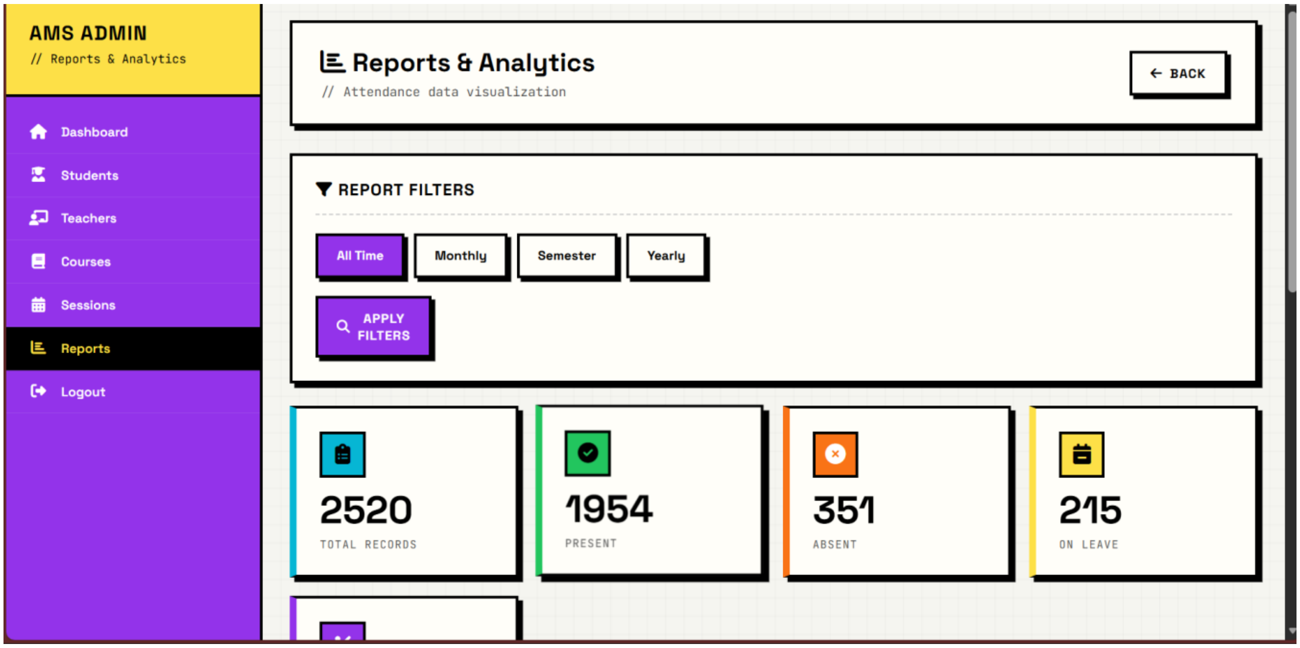
- *Concept:* A mental model is the user's internal simulation of how the system works.
- *Application:* We aligned the system with the **"Physical Switch" Model**. When a toggle is clicked, it doesn't just change color; it physically moves (CSS Transform). This aligns with the user's mental model of a mechanical light switch—action equals physical movement.

6. Interaction Framework: The "Gulfs"

We analyzed the interaction using Donald Norman's **Gulf of Execution** and **Gulf of Evaluation**

6.1 Bridging the Gulf of Execution

- *The Gulf:* The gap between "I want to do X" and "How do I do X with this interface?"
- *Solution **(Affordance):*** In "Flat Design," buttons often look like plain text, widening this gulf. Our Neo-Brutalist design uses **Explicit Affordance**. Every clickable element has a thick, hard border and a "drop shadow" that gives it apparent depth. The button literally *looks* like it can be pressed, instantly showing the user *how* to execute their intent.



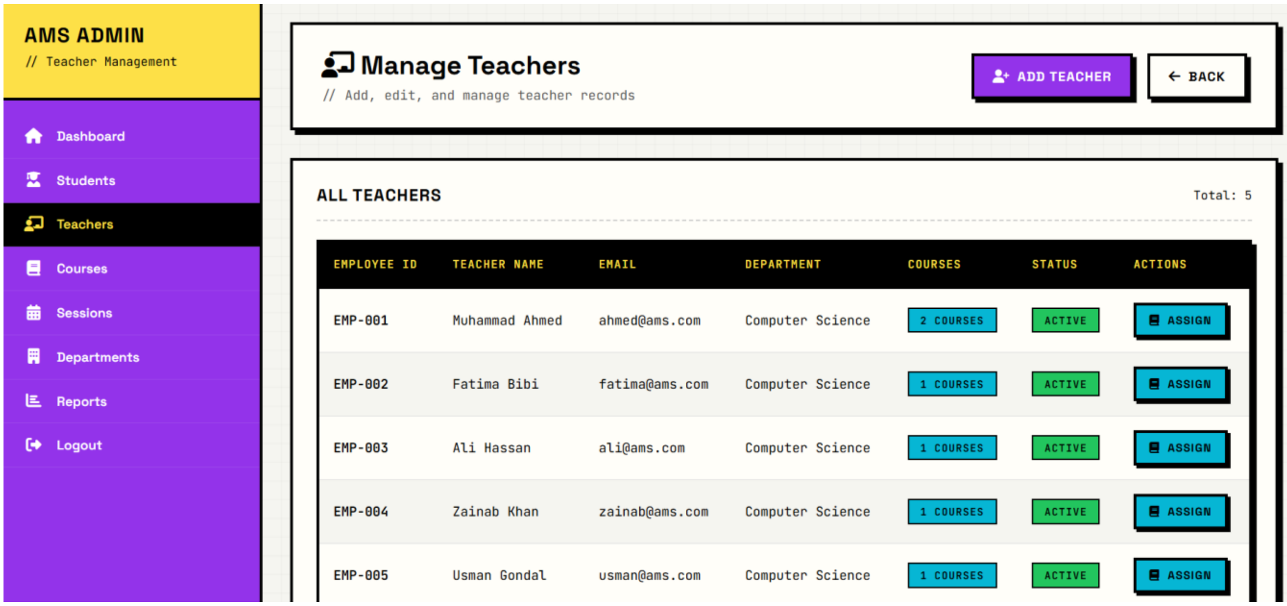
6.2 Bridging the Gulf of Evaluation

- *The Gulf:* The gap between "I clicked the button" and "Did the system understand me?"
- *Solution (Feedback):* We focused on **Micro-Interactions**.
    1. **Hover State:** Cursor becomes a pointer; shadow grows slightly (System says: "I see you").
    2. **Active State:** Shadow disappears; button moves down (System says: "I felt the click").
    3. **Success State:** Button turns green (System says: "I have recorded the data"). This continuous feedback loop keeps the user constantly informed, eliminating uncertainty.

7. Design Principles & Heuristics.

7.1 Visibility

- *Principle:* The more visible functions are, the more likely users will be able to know what to do next.
- *Application:* We avoided "Mystery Navigation" (like hidden sidebars). The primary navigation (Dashboard, Classes, Reports) is always visible in a sticky sidebar. The "Status Summary" is pinned to the top of the attendance sheet, so the teacher always sees the count (e.g., "45 Present / 5 Absent") regardless of scroll position.



7.2 Consistency

- *Principle:* Interfaces should look and behave the same way throughout.
- *Application (Internal Consistency):* We established a rigorous **Design System**.
    - All headings are font-family: 'Space Grotesk'.
    - All data is font-family: 'Courier Prime' (Monospace).
    - All borders are 3px solid black.
    - All success messages are Green; all errors are Red. This consistency allows users to learn the "Rules of the Interface" on the first page and apply them everywhere.

7.3 Constraints

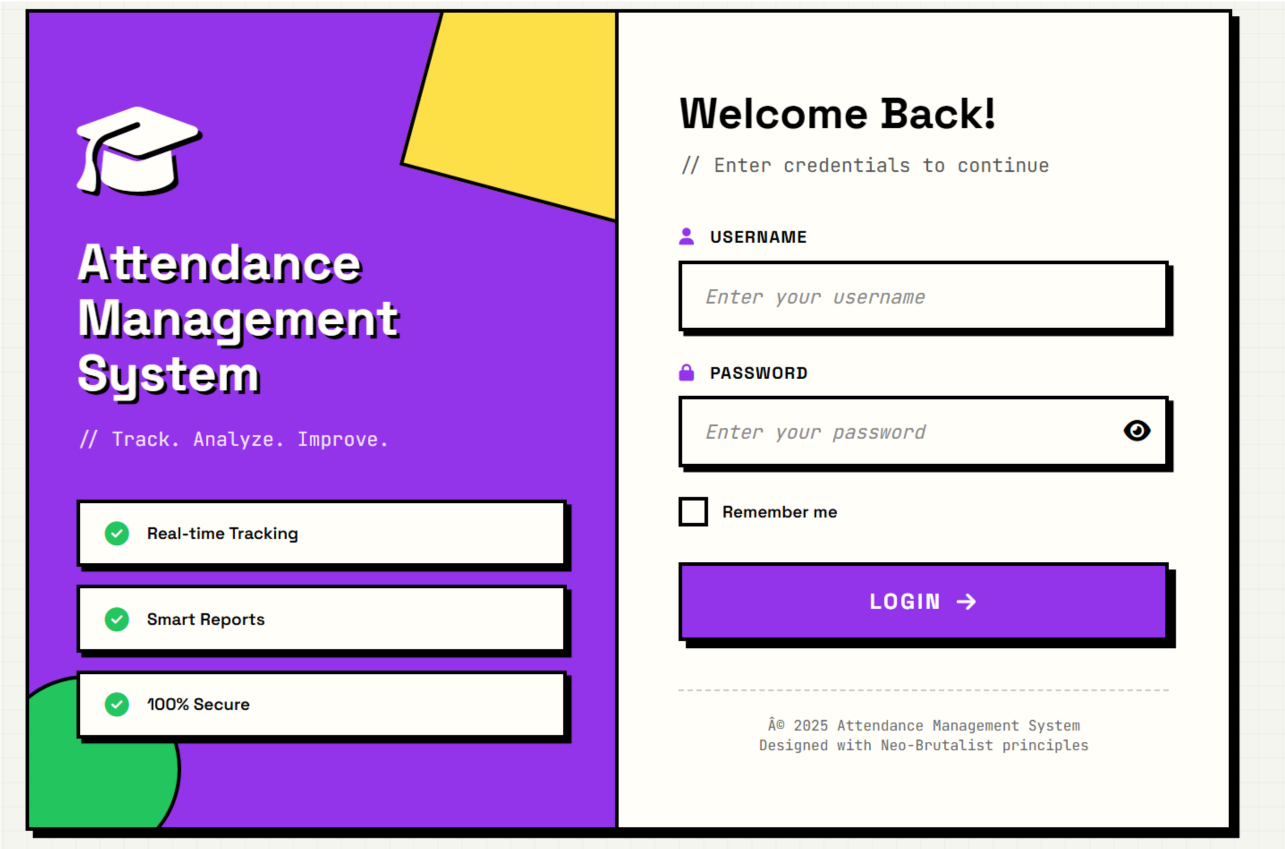
- *Principle:* Restricting interaction to prevent errors.
- *Application:* We used **Logical Constraints**. The "Edit" mode is disabled by default for past dates. This prevents a professor from accidentally changing a record from last month while trying to scroll through the list. They must deliberately click "Unlock Record" to make changes.

8. Requirements Analysis: The FURPS+ Model

We conducted a formal requirements analysis using the **FURPS+** model

8.1 Functionality

- **Capabilities:** CRUD (Create, Read, Update, Delete) attendance; Export Reports (CSV/PDF); Student Profile Management.
- **Security:** Role-Based Access Control (RBAC). Students can only *view* attendance; Faculty can *edit*.

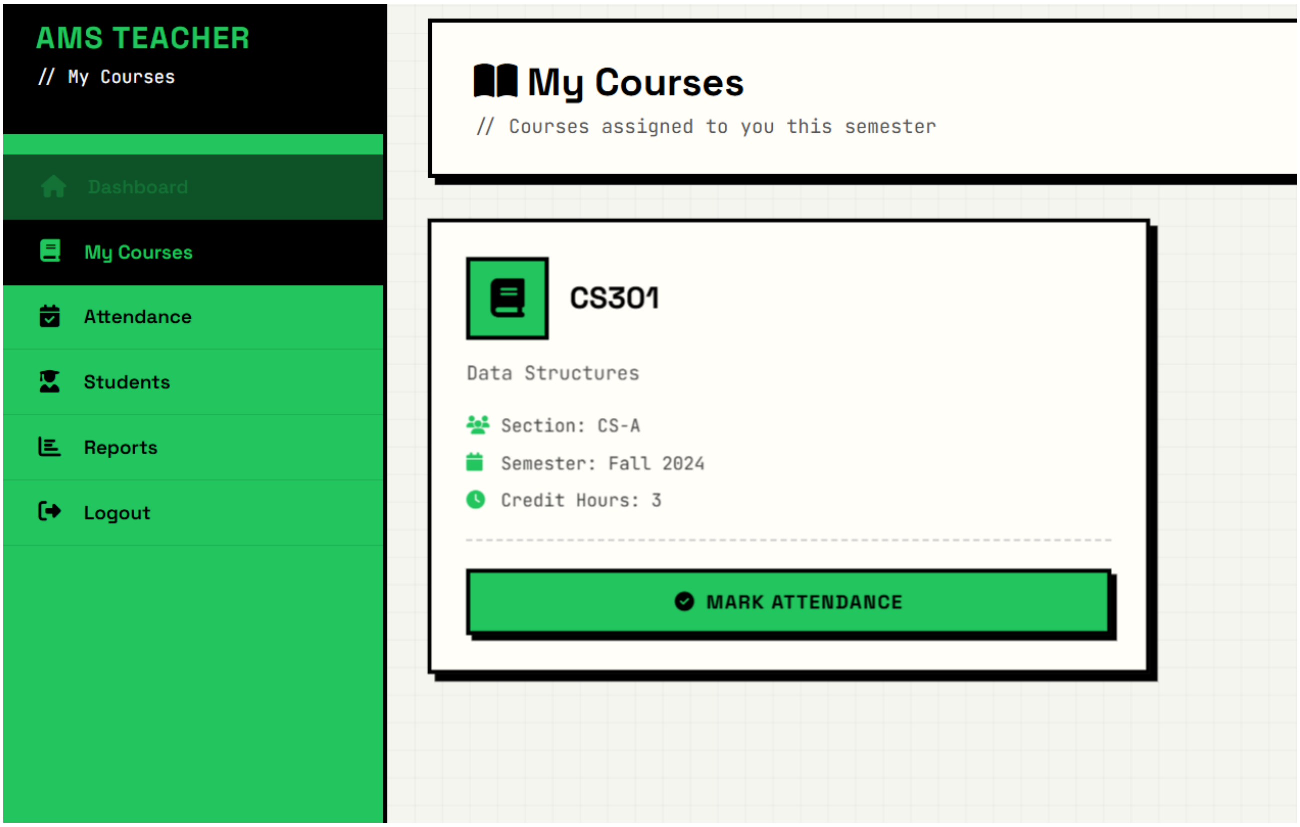


8.2 Usability

- **Accessibility:** The high-contrast design naturally meets **WCAG AAA** standards, assisting users with low vision or color blindness (achromatopsia).
- **Responsiveness:** The layout is responsive (Flexbox/Grid), adapting to Mobile Phones (for teachers taking attendance while walking around) and Desktops.

8.3 Reliability

- **Availability:** The system uses "Local Storage" caching. If the internet cuts out mid-class, the data is saved locally in the browser and synced when the connection returns.
- **Robustness:** The input fields validate data type (e.g., preventing text in numeric Roll No fields).



8.4 Performance

- **Response Time:** The UI is optimized for <100ms interaction time. We avoid heavy JavaScript frameworks where possible, relying on CSS for transitions to ensure smooth performance on older faculty laptops.

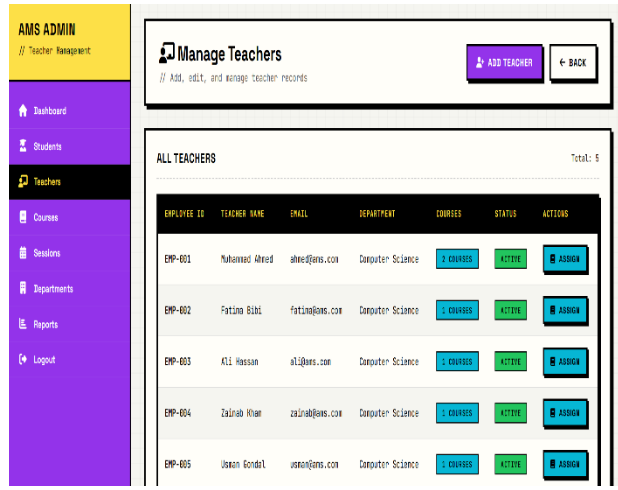
8.5 Supportability

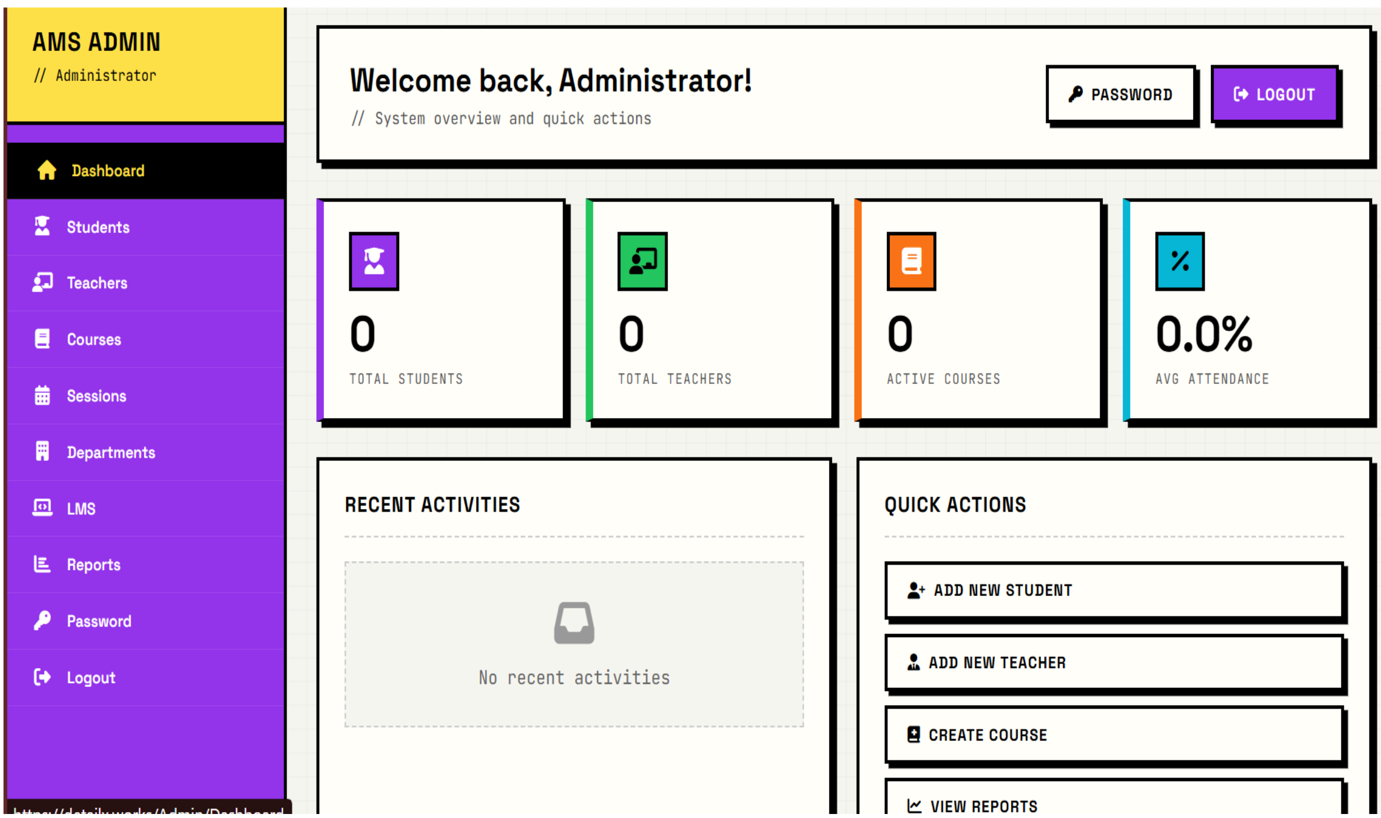
- **Maintainability:** The CSS uses **Variables** (`--primary-color`, `--border-width`). To change the theme from "Neo-Brutalist" to "Corporate," we only need to change 5 lines of code in the root file.

8.6 "+" Factors (Constraints & Implementation)

- **Implementation Req:** Built using .NET Core (Backend) and Razor Pages (Frontend).
- **Interface Req:** Must integrate with the university's existing Student Database via REST API.


AMS TEACHER
// Mark Attendance
Dashboard
My Courses
Attendance
Students
Reports
Logout
F24-014 Shaheen Afridi shaheen.afridi@ams.com
F24-015 Naseem Shah naseem.shah@ams.com
F24-016 Fatima Sana fatima.sana@ams.com
F24-017 Arshad Nadeem arshad.nadeem@ams.com
F24-018 Bismah Maroof bismah.maroof@ams.com
F24-019 Shadab Khan shadab.khan@ams.com
F24-020 Sidra Ameen sidra.ameen@ams.com
✓ Present
× Absent
Leave
MARK ALL PRESENT
SUBMIT ATTENDANCE


AMS ADMIN
// Teacher Management
Dashboard
Students
Teachers
Courses
Sessions
Departments
Reports
Logout
Manage Teachers
// Add, edit, and manage teacher records
ADD TEACHER
← BACK
ALL TEACHERS
Total: 5
EMPLOYEE ID TEACHER NAME EMAIL DEPARTMENT COURSES STATUS ACTIONS
EMP-001 Muhammad Ahmed ahmed@ams.com Computer Science 2 COURSES ACTIVE ASSIGN
EMP-002 Fatima Bibi fatima@ams.com Computer Science 1 COURSES ACTIVE ASSIGN
EMP-003 Ali Hassan ali@ams.com Computer Science 1 COURSES ACTIVE ASSIGN
EMP-004 Zainab Khan zainab@ams.com Computer Science 1 COURSES ACTIVE ASSIGN
EMP-005 Usman Gondal usman@ams.com Computer Science 1 COURSES ACTIVE ASSIGN


AMS ADMIN
// Administrator
Dashboard
Students
Teachers
Courses
Sessions
Departments
LMS
Reports
Password
Logout
Welcome back, Administrator!
// System overview and quick actions
PASSWORD
LOGOUT
0
TOTAL STUDENTS
0
TOTAL TEACHERS
0
ACTIVE COURSES
0.0%
AVG ATTENDANCE
RECENT ACTIVITIES
No recent activities
QUICK ACTIONS
ADD NEW STUDENT
ADD NEW TEACHER
CREATE COURSE
VIEW REPORTS

9. Interaction Flow: Norman's 7 Stages of Action

To validate the design, we simulated a core task using Norman's model

Task: Mark Student "Ali" as Absent.

1. **Formulate Goal:** I want to mark Ali absent.
2. **Formulate Intention:** I need to find Ali in the list and change his status.
3. **Specify Action:** I will look for the row starting with "Ali" and click the button labeled "P" to toggle it.
4. **Execute Action:** I move the mouse to the "P" button and click.
   - *Design Check:* Is the button large enough? Yes (Fitts's Law compliant).
5. **Perceive System State:** The button visually depresses; the color changes from Green ("P") to Red ("A").
6. **Interpret System State:** The Red color and "A" label mean "Absent."
7. **Evaluate Outcome:** Is Ali now marked absent? Yes. The goal is achieved.

10. Methodology: Data Gathering

- **Direct Observation:** We observed that teachers often forget the date when filling manual sheets.
  - *Design Solution:* The system auto-fills the current date and freezes it, removing the cognitive load of remembering the date.
- **Unstructured Interviews:** Teachers complained that "dropdown menus are annoying."
  - *Design Solution:* We replaced all dropdowns with visible "Radio Groups" or "Toggle Buttons" for faster, one-click selection

.

11. Conclusion

The Attendance Management System (AMS) stands as a definitive example of function-first design, where every architectural decision was driven by the rigorous demands of the academic environment rather than arbitrary stylistic preferences. By strictly adhering to the core tenets of Human-Computer Interaction, we have engineered a solution that transcends basic usability to achieve an optimal interaction flow.

The visual language employed is not a superficial layer but a strategic engineering solution designed to bridge the gap between user intent and system action. Through the use of high-contrast cues, clear structural boundaries, and immediate feedback mechanisms, the interface actively guides the user's hand, removing the ambiguity often found in digital tools. The final result is a robust system that fundamentally respects the faculty member's time, proactively mitigates error through logical constraints, and delivers a tangible, responsive user experience that creates confidence in every interaction.

.