ALBUKHARY INTERNATIONAL UNIVERSITY

**SCHOOL OF COMPUTING AND INFORMATICS**

**MARCH INTAKE**

## CDE2224: WEB DESIGN AND DEVELOPMENT

# Report, System Prototype Documentation and User Manual.


| NAME | STUDENT ID |
|---|---|
| UGYEN TSHERING | AIU22102222 |
| AREYAN MUHEMED ALI HUSSEIN | AIU21102179 |
| ABDULLAHI ADEWOLE ZAKARIYAH | AIU22102360 |
| NIMA YOEZER | AIU22102221 |
| UMAR MUHAMMAD | AIU22102266 |

**Lecturer: Assoc. Prof. Dr Leelavathi Rajamanickam**

# System Prototype Documentation

## 1. Introduction

The purpose of this document is to demonstrate the **working prototype** of the KD Academy School Management System. This system is designed to streamline course enrollment, attendance tracking, and academic performance monitoring for students, teachers, and administrators. The prototype reflects the final implementation of the system, including its user interfaces, workflows, and key functionalities.

## 2. System Overview

The system is a **web-based application** built with:

- **Frontend**: HTML, CSS, JavaScript, Bootstrap (for responsive design).
- **Backend**: PHP and JavaScript.
- **Database**: MySQL for storing user data, courses, attendance, and grades.
- **Deployment**: Runs locally on a laptop using XAMPP

**Key Features Implemented:**

- Role-based user authentication (Student, Teacher, Admin).
- Programme-specific course registration (BCS, BMC, BBA).
- Student group management (Group A/B).
- Attendance tracking with color-coded feedback.
- Academic results and GPA calculation.
- Admin/Teacher dashboard with graphical data.
- User management and account editing.

## 3. User Roles

1. **Students**: Enroll in courses, view attendance (color-coded), check grades, and download course PDFs.
2. **Teachers**: Mark attendance, upload grades, and view course/student statistics.
3. **Administrators**: Manage users, courses, and system settings.

# 4. Functional Features with Screenshots

## 4.1 Login and Registration Page

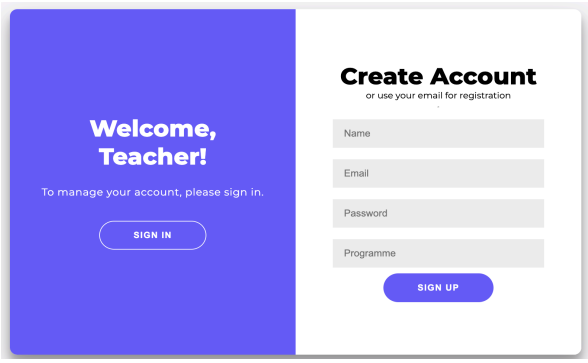



**Figure 1.** Teacher Sign-in Page

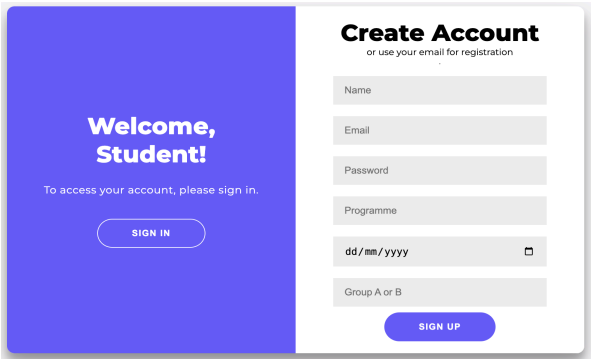



**Figure 2.** Student Sign-in Page

**Description**:

- Students register with **Name, Email, Password, Programme (BCS/BMC/BBA), Date of Birth, and Group (A/B)**.
- Teachers register with **Name, Email, Password, and Programme**.
- Admins are pre-configured in the database.

| username | password |
|---|---|
| admin | HelloWorld |
| admin@gmai | HelloWorld |

## 4.2 Sign in Page

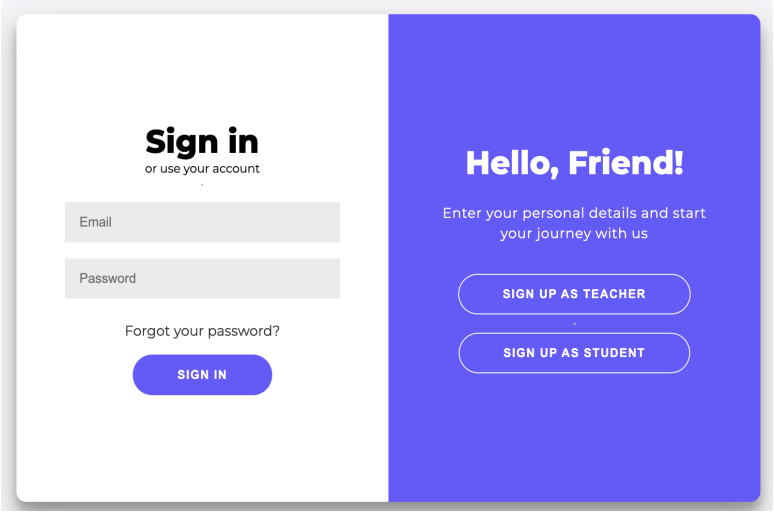



**Figure 3.** Account Log-in Page (For both student and teacher)

**Description**:

- Users log in with email and password.
- The system redirects to role-specific dashboards after authentication.

### 4.2 Student Dashboard

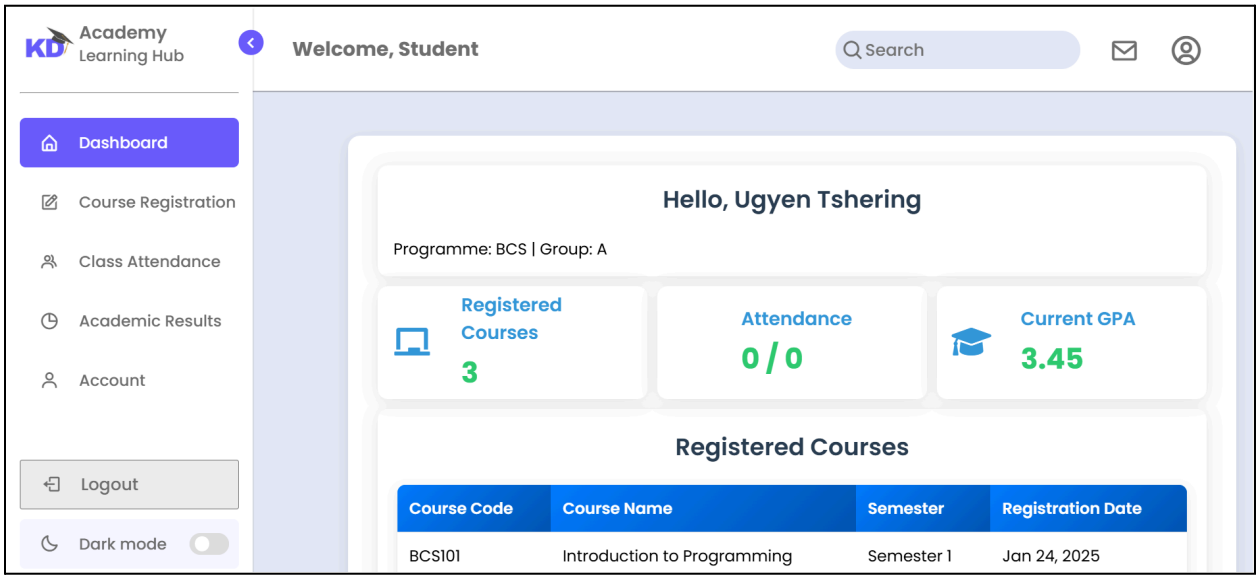



**Figure 4.** Student Dashboard

**Description**:

- Displays name, programme and group they are registered to.
- Also displays the **number of enrolled courses** and **semester GPA**.
- Includes a button to **download registered courses as a PDF**.

### 4.3 Course Registration Page

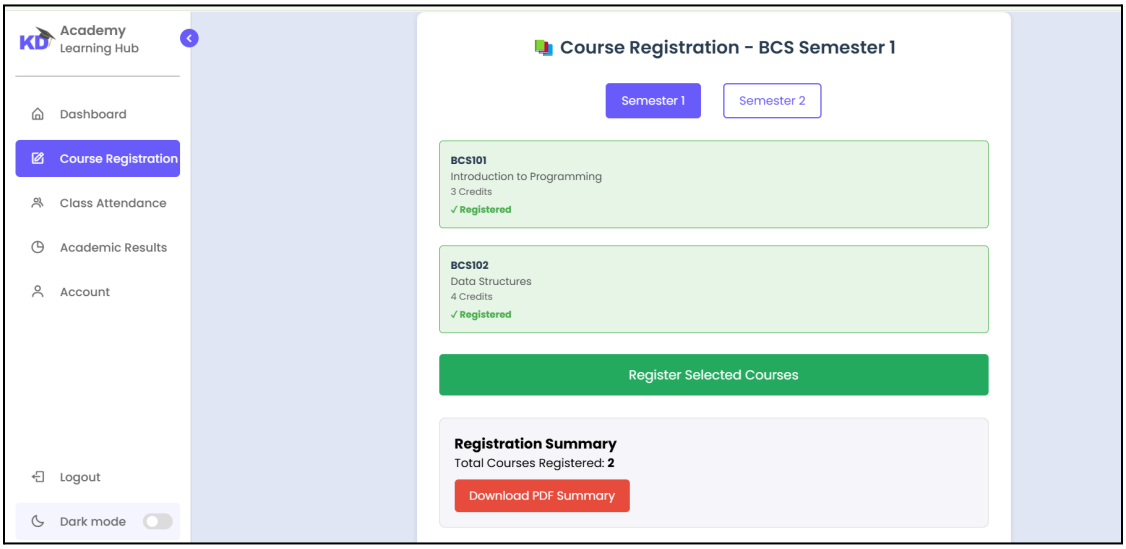


**Figure 5.** Student's Course Registration Page

- Only courses relevant to the student's programme (e.g., BCS) are displayed.
- Courses include **name, code, and credits**.
- Total registered credits are shown at the bottom.

### 4.4 Attendance Tracking (Teacher View)

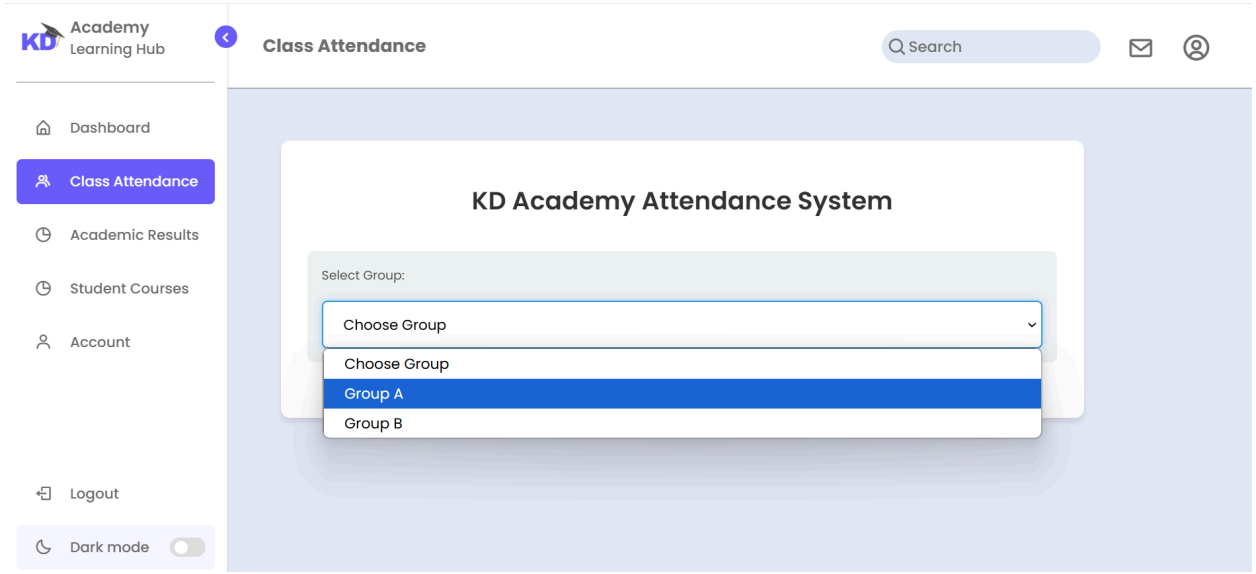



**Figure 6**: Group selection for teacher class attendance

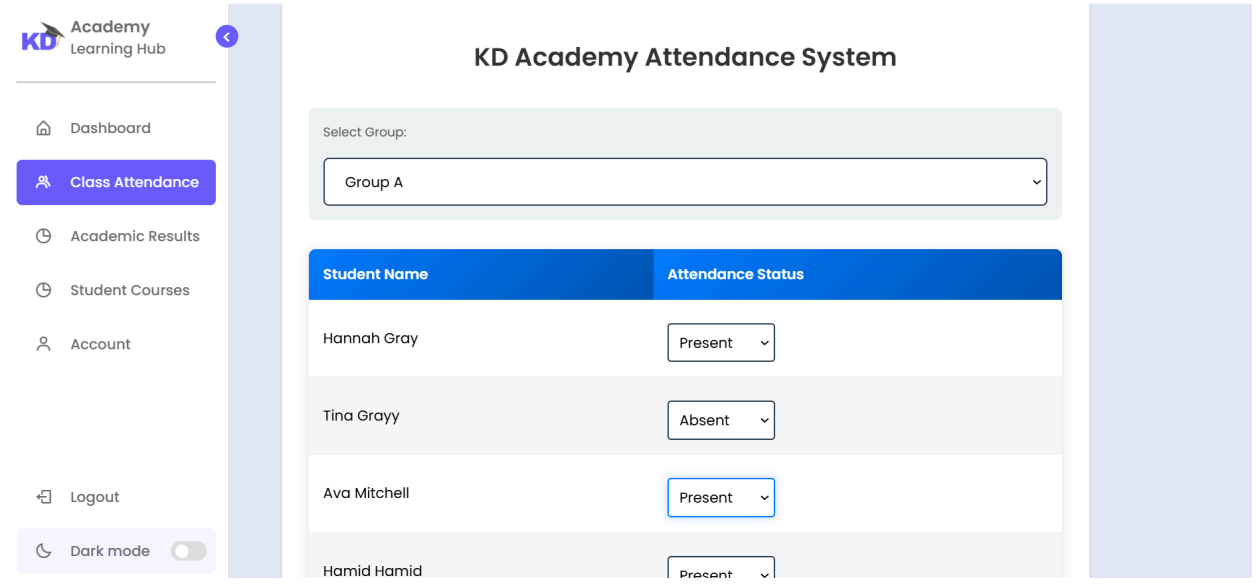



**Figure 7**: Students attendance marking by Teacher

**Description**:

- Teachers filter students by **programme (BCS/BMC/BBA)** and **group (A/B)**.

- Attendance is marked as **Present/Absent** with a dropdown or checkbox.

### 4.5 Academic Results Page

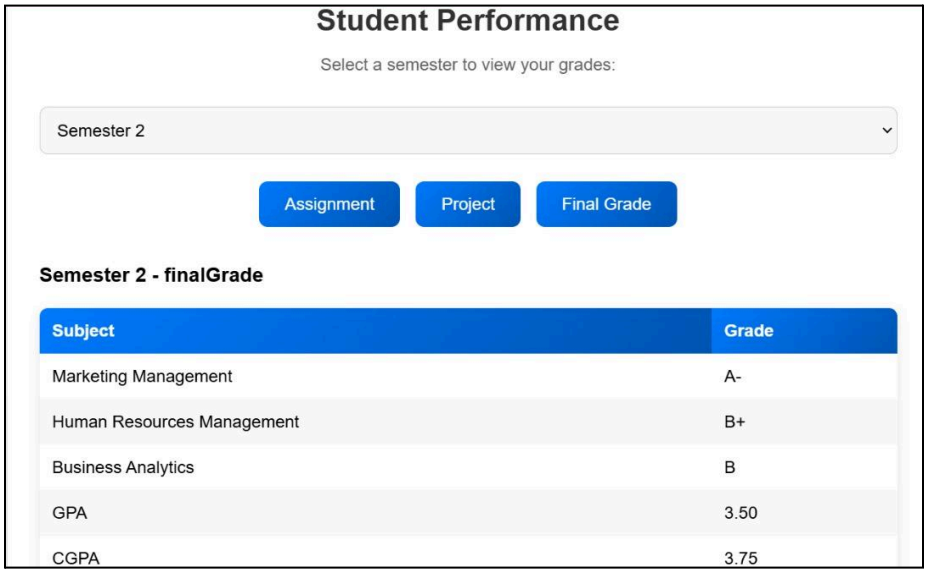



**Figure 8:** Students Academic Results Display

**Description**

- Students select a semester from a dropdown to view **grades, grade points, remarks, and GPA**.
- Results are categorized by exams, projects, and assignments.

### 4.6 Admin/Teacher Dashboard

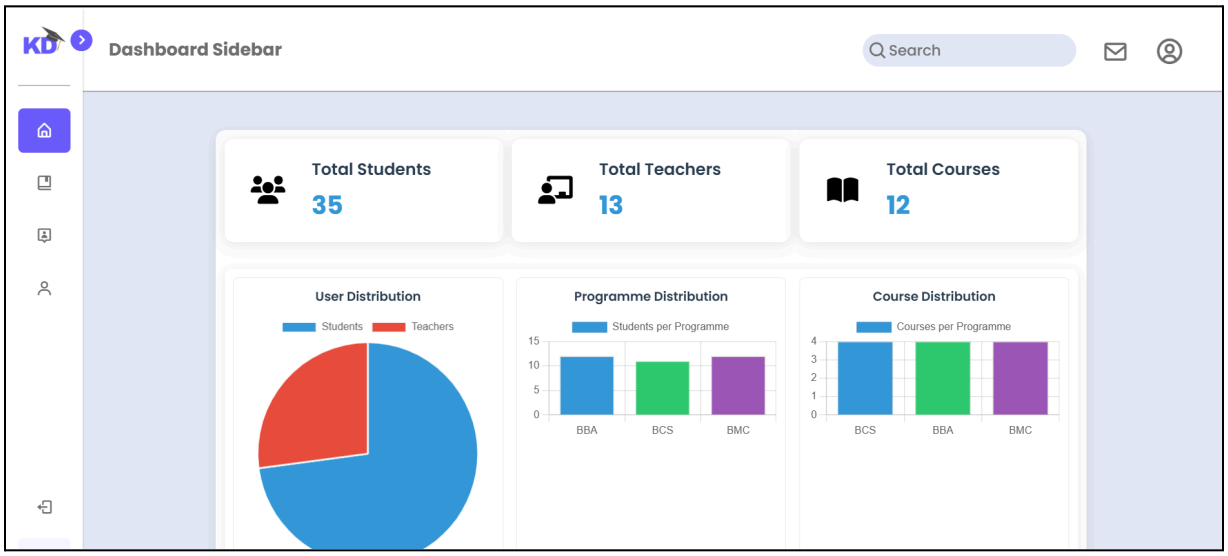



**Figure 9.** Dashboard of Teacher and Admin

**Description**:

- Displays **total students, teachers, and courses** using **pie/bar charts**.

### 4.7 Student Courses (Teacher View)

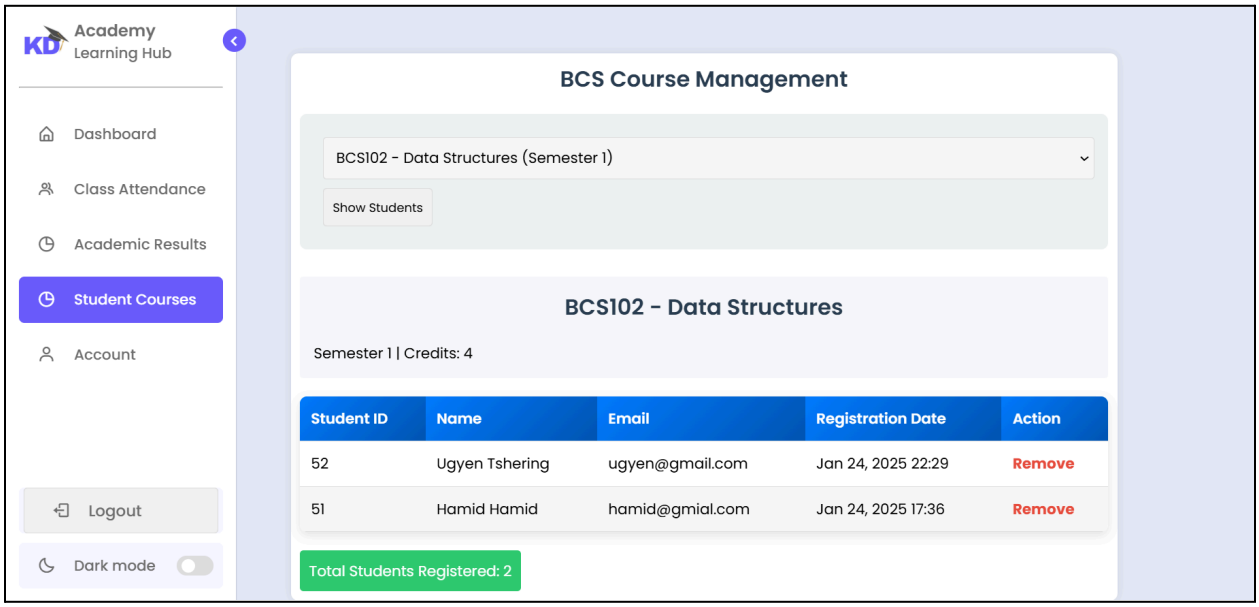



**Figure 10.** Student's management

**Description:**

- Teachers can view/remove students from courses.

### 4.7 User Management (Admin View)

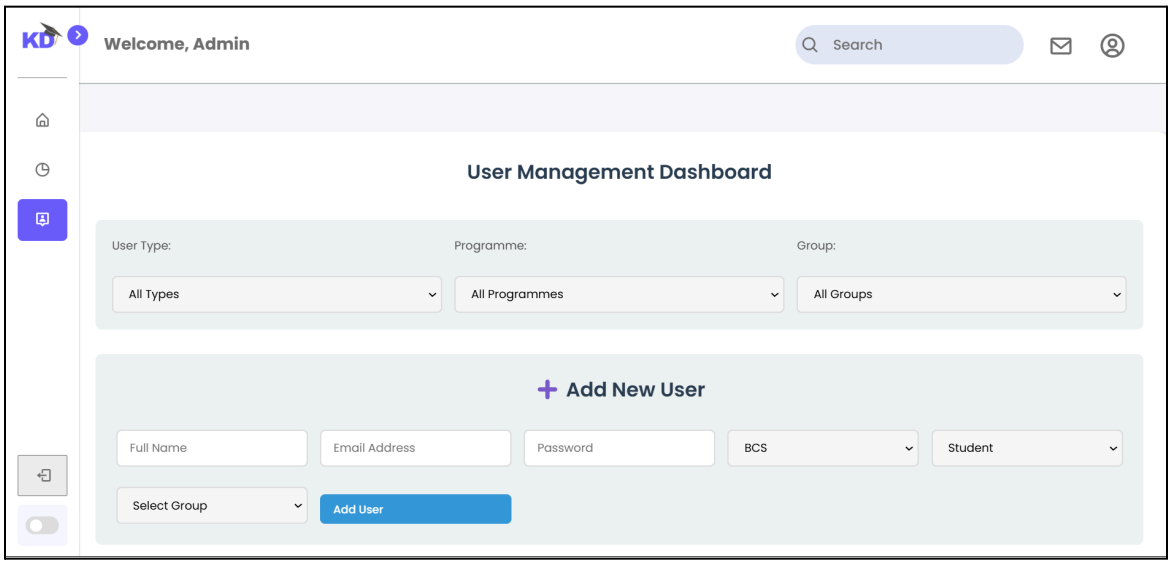



**Figure 11.** Admin's management page for filtering users and adding new user

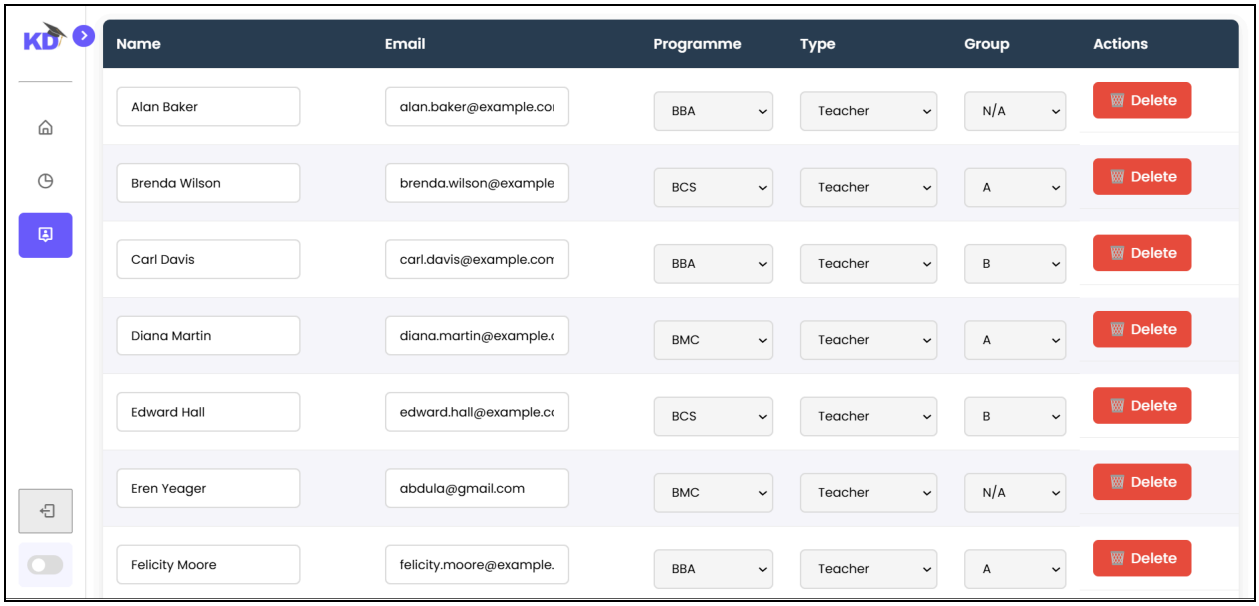



**Figure 12.** Admin's management page for editing users data and removing users

**Description**:

- Admins can **add users** (name, email, password, role, programme, group).
- Users are filtered by **role (student/teacher)**, **programme**, or **group**.
- Admins can edit or delete users and **save all changes** at once.

### 4.8 Account Management Page

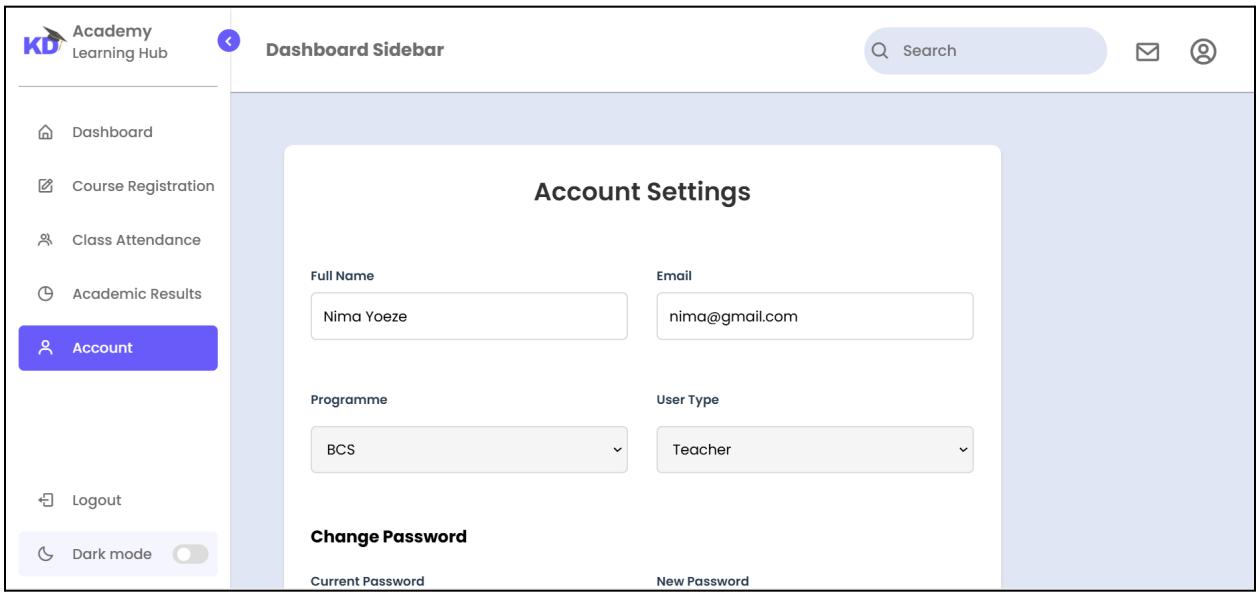



**Figure 13.** Account Setting

**Description**:

- Users can update their **name, email, programme, group, or password**.

## 5. User Interface (UI) and User Experience (UX)

- **Design Principles**:
  - Minimalist layout with Bootstrap for consistency.
  - Color-coded attendance for quick visual feedback.
  - Intuitive navigation menus for students, teachers, and admins.

## 6. Technical Architecture

### 6.1 Database Structure

- **Tables**:
  - **users** (id, name, email, password, programme, group, dob, user_group, user_type).
  - **courses** (id, name, code, credits, programme, semester).
  - **attendance** (student_id, course_id, date, status).
  - **student_grades** (student_id, course_id, semester, grade, quiz_score, project_score, test_score, programme, user_id).
  - **registration** (id, user_id, course_id, semester, registration_date)
  - **semester** (semester_id, semester_name)
  - **admin** (username, password)
  - **subjects** (subject_id, subject_name)

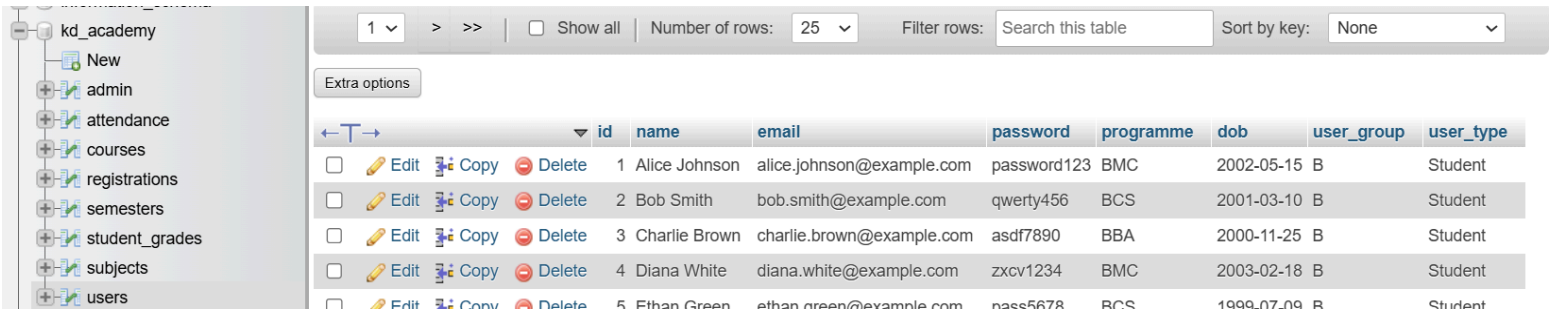



**Figure 14**: MySQL database tables

### 6.2 System Workflow

1. **Registration → Login → Role-specific Dashboard**.
2. **Students**: Register for courses → View attendance/results → Download PDF → Change account info.
3. **Teachers**: Mark attendance → Upload grades → View statistics → Change account info.
4. **Admins**: Manage users/courses → View analytics.

## 7. Testing Summary

- **Test Cases Covered**:
    - User authentication and role-based access.
    - Programme-specific course enrollment.
    - Attendance marking and GPA calculation.
    - PDF generation and data filtering.

## 8. Conclusion

The prototype successfully implements all core functionalities outlined in the SRS. Future enhancements could include email notifications and data export features.

# Report

## 1. Introduction

The KD Academy School Management System is a web-based platform designed to streamline course enrollment, attendance tracking, and academic performance monitoring for KD Academy, a private school in Malaysia with over 1,000 students. Developed over two weeks, this system prioritizes simplicity, security, and usability, aligning with the functional and non-functional requirements outlined in the **Software Requirements Specification (SRS)**. This report documents our development process, design choices, and system outcomes.

## 2. Documentation

### 2.1 Requirements Analysis

#### Alignment with SRS Goals

The system fully addresses the SRS objectives:

- **User Authentication**: Role-based login pages (Figures 1–3) ensure secure access for students, teachers, and administrators.
- **Course Management**: Teachers and admins can add/remove courses (Figures 11–12), while students register via an intuitive interface (Figure 5).
- **Attendance & Performance Tracking**: Teachers mark attendance (Figure 11), and students view real-time percentages (Figure 6) and grades (Figure 7).

*Challenges*: Balancing the 2-week deadline with role-specific dashboards required prioritizing core features over advanced analytics (excluded per SRS 1.2).

#### Non-Functional Requirements

- **Security**: Passwords are hashed using PHP's password_hash() function.
- **Performance**: Optimized SQL queries ensure pages load within 2–3 seconds.
- **Usability**: Bootstrap's grid system enabled responsive design across devices.

### 2.2 Methodology & Technology Selection

**Development Workflow**

- **Agile Methodology**: We adopted Agile for iterative development, with daily standups to track progress.
- **Version Control**: GitHub facilitated team collaboration.
- **Wireframing**: Figma wireframes guided UI development

**Technology Stack**

- **Frontend**: HTML5, CSS, and JavaScript with Bootstrap for responsive layouts.
- **Backend**: PHP for server-side logic and MySQL for structured data storage (e.g., attendance records).
- **Justifications**:
    - **PHP & MySQL**: Chosen for compatibility and rapid prototyping, as outlined in SRS 1.2.
    - **Bootstrap**: Used UI development while ensuring cross-device consistency.

### 3.1. Unit testing

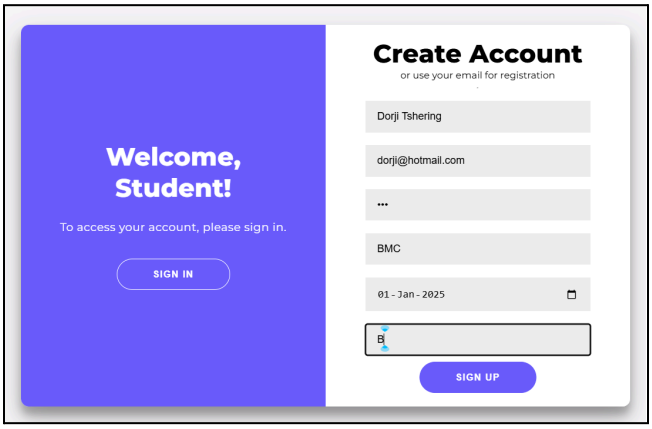
Unit testing is a process where small parts of a program, like functions or methods, are tested individually to ensure they work correctly. Each unit of a web application like user course registration, grade performance page, and attendance system is tested in isolation to ensure it works according to expectations.

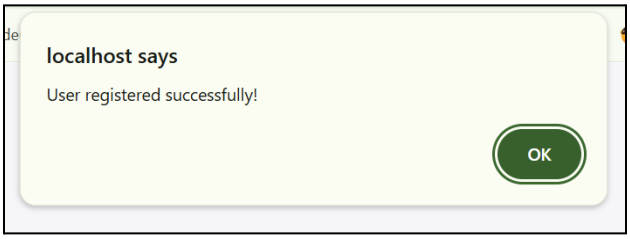
**Tools used:**

- **MySQL:** A relational database management system (RDBMS) that uses SQL for storing, querying, and managing structured data, widely used in web applications.
- **HTML:** The core markup language for structuring web content, defining elements like headings, paragraphs, and forms to create web page layouts.
- **JavaScript:** A versatile scripting language enabling dynamic, interactive web experiences through client-side logic, DOM manipulation, and frameworks like React.
- **PHP:** A server-side scripting language designed for web development, powering dynamic content and database interactions (e.g., with MySQL) in platforms like WordPress.

# Test case Testing

## 1. New User registration





New user has been successfully registered and updated in the sql database.

2. **Course Registration Management:** Tested the course registration functionality, confirming that students can successfully register for courses. Verified that the students can register the course according to their programme and is successfully registered in the database.

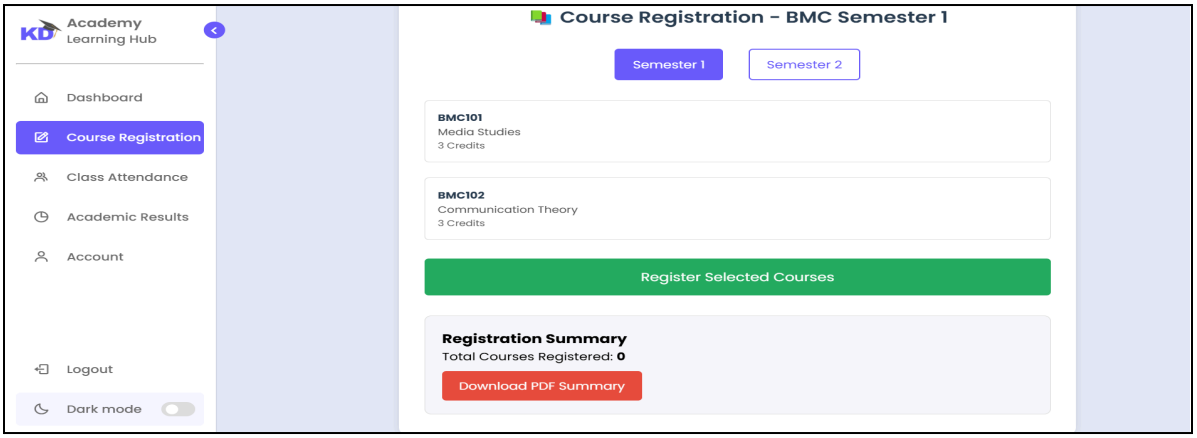



**Figure 15.** Student Registering course

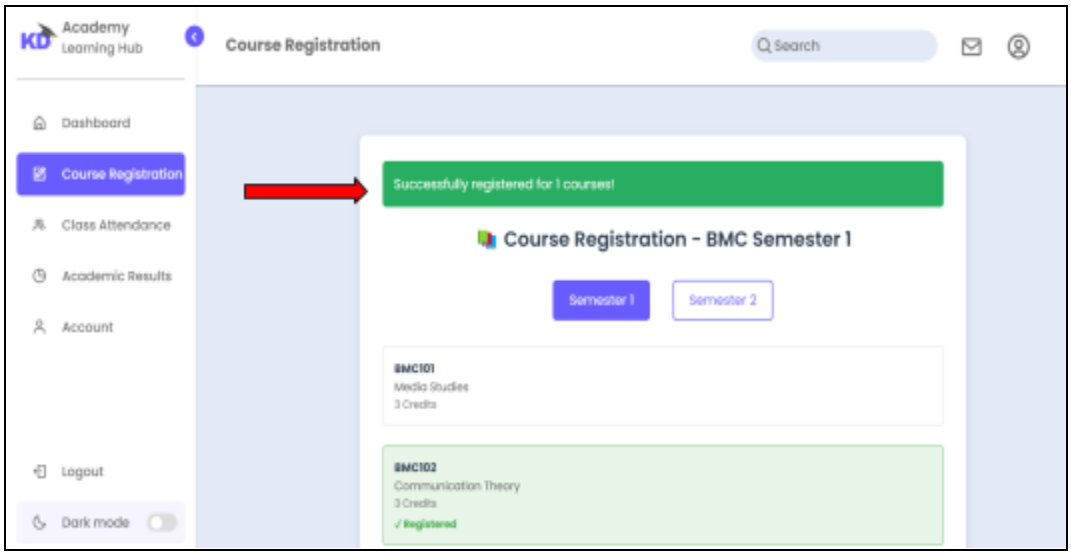



**Figure 16.** Course Registered Successfully

| | id | user_id | course_id | semester | registration_date |
|---|---|---|---|---|---|
| Edit Copy Delete | 8 | 50 | 7 | 2 | 2025-01-24 16:35:39 |
| Edit Copy Delete | 9 | 50 | 8 | 2 | 2025-01-24 16:35:39 |
| Edit Copy Delete | 10 | 50 | 6 | 1 | 2025-01-24 16:42:18 |
| Edit Copy Delete | 12 | 51 | 2 | 1 | 2025-01-24 17:36:26 |
| Edit Copy Delete | 13 | 51 | 1 | 1 | 2025-01-24 19:49:39 |
| Edit Copy Delete | 14 | 52 | 1 | 1 | 2025-01-24 22:29:12 |
| Edit Copy Delete | 15 | 52 | 2 | 1 | 2025-01-24 22:29:12 |
| Edit Copy Delete | | 52 | 4 | 2 | 2025-01-24 22:29:15 |
| Edit Copy Delete | 18 | 53 | 10 | 1 | 2025-01-25 00:16:57 |

**Figure 17.** Student successfully registered in Database

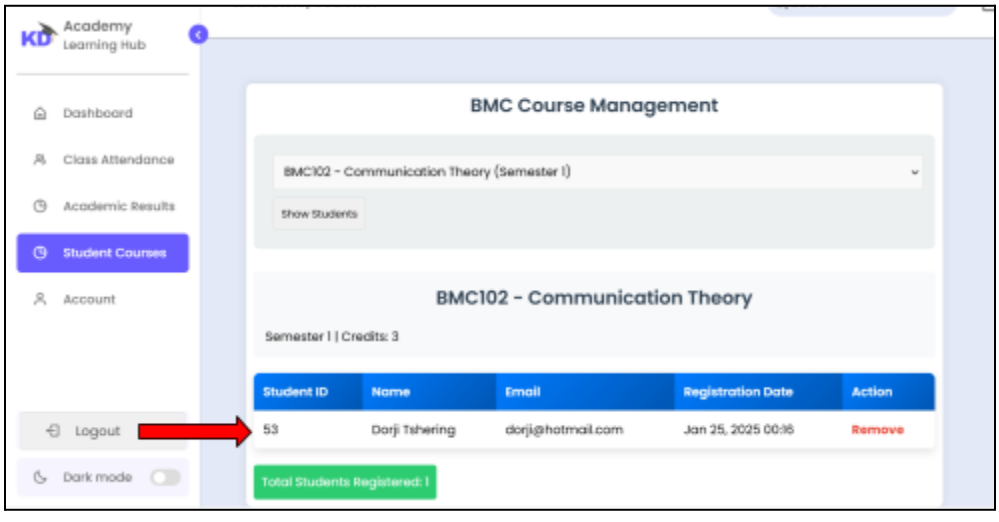



**Figure 18.** Registered students can be viewed by the teachers.

3. **Test login Functionality:** Verified that the users can register and log in using valid credentials(correct email and password )

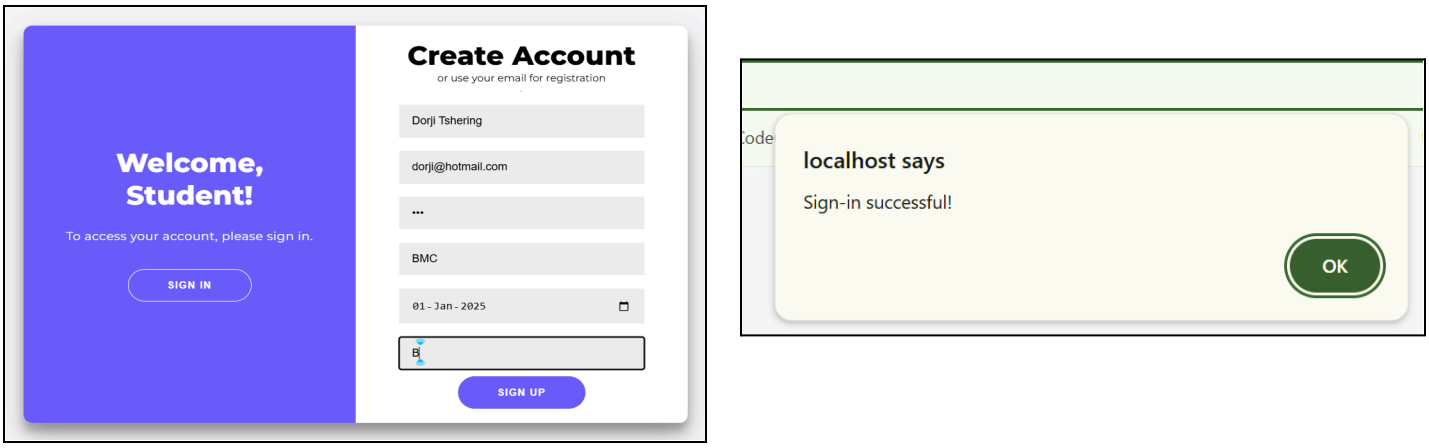
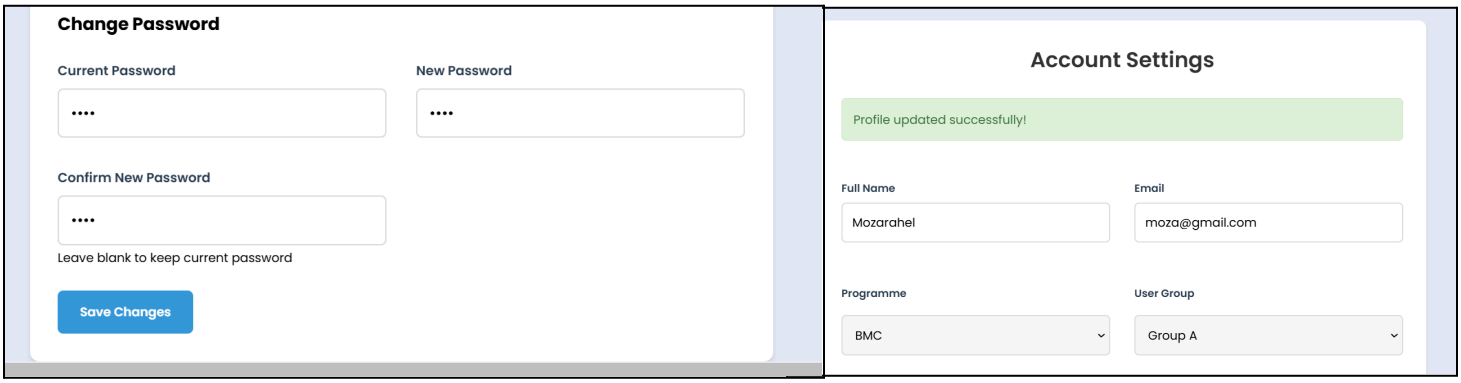
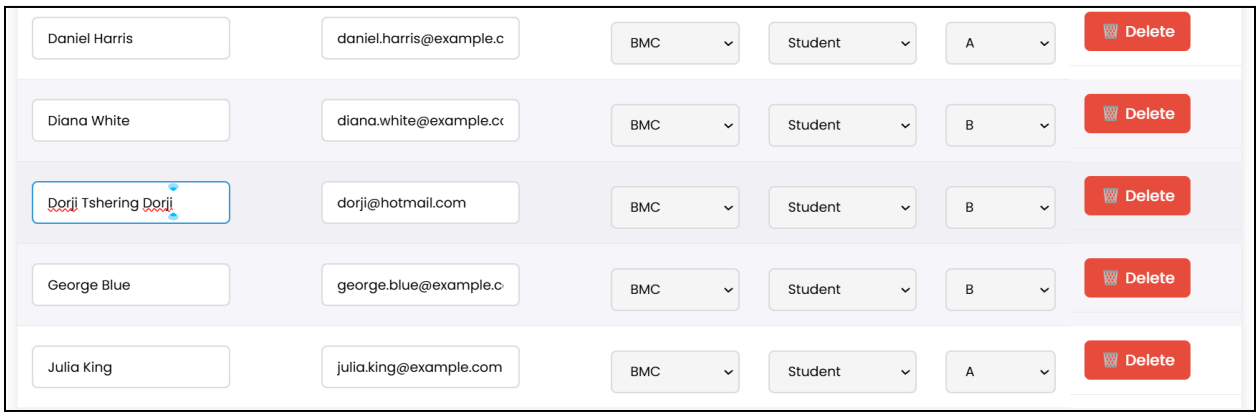


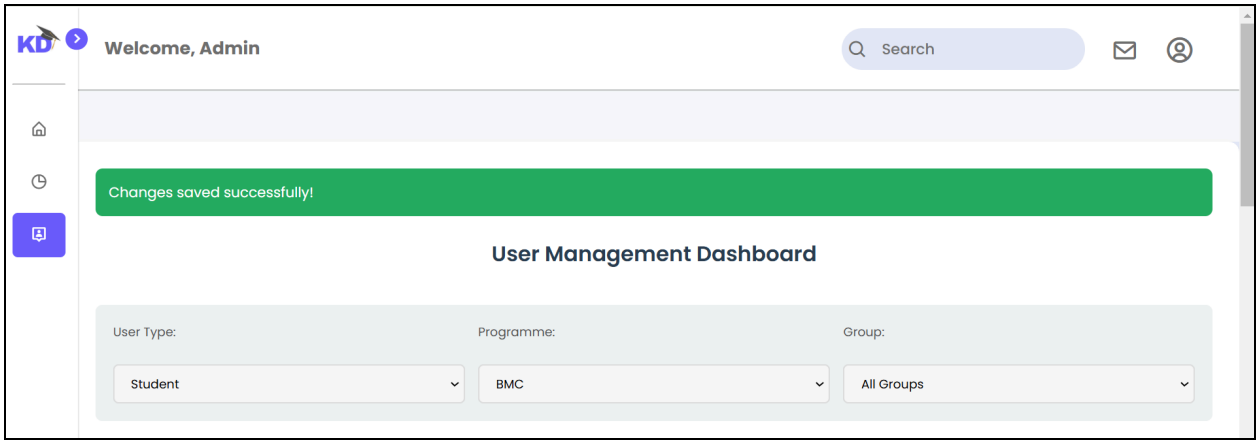

**Figure 19.** Users successfully registered and can log in.



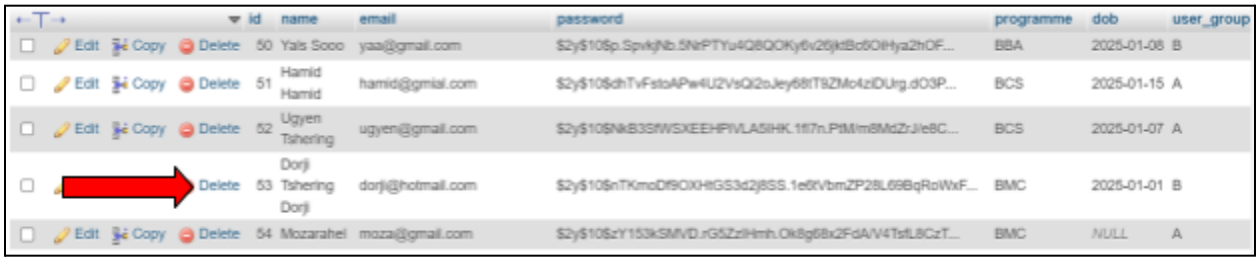

Checking if admin user management system is working or not.




Adding ‘Dorji’ to Dorji Tshering and clicking ‘save’.




Changes made successfully.




Updated in the database as well.

**3.2. User Testing:** User testing focuses on assessing the system from the perspective of actual users to ensure it aligns with their needs and expectations. This process identifies issues related to usability, navigation, and overall user experience.

**Test Cases: Alpha Testing**

1. **User Interface Usability:**

   The KD Academy Web Application's testing process verifies alignment with functional requirements and usability benchmarks. Unit tests ensure the accuracy and stability of isolated components, system testing assesses the integrated platform's seamless operation and efficiency, while user acceptance testing (UAT) guarantees a user-friendly interface tailored to diverse needs. Insights from these evaluations drive iterative refinements, ensuring long-term scalability and heightened user satisfaction.

Video Demonstration video here:
https://drive.google.com/drive/folders/11ZZXXJzrldcjcuagJSjxJKg0UJwq3tSD?usp=shar

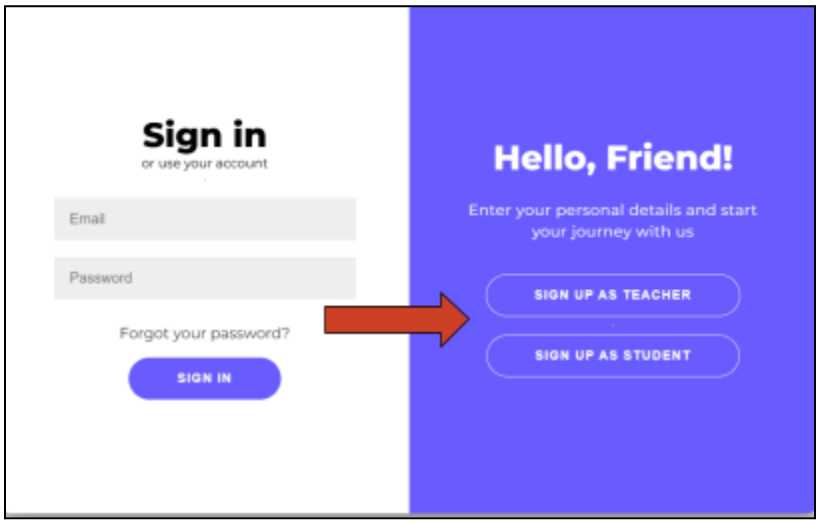
# User Manual

**How to signup and login?**

1. **Sign Up as a Student or Teacher**

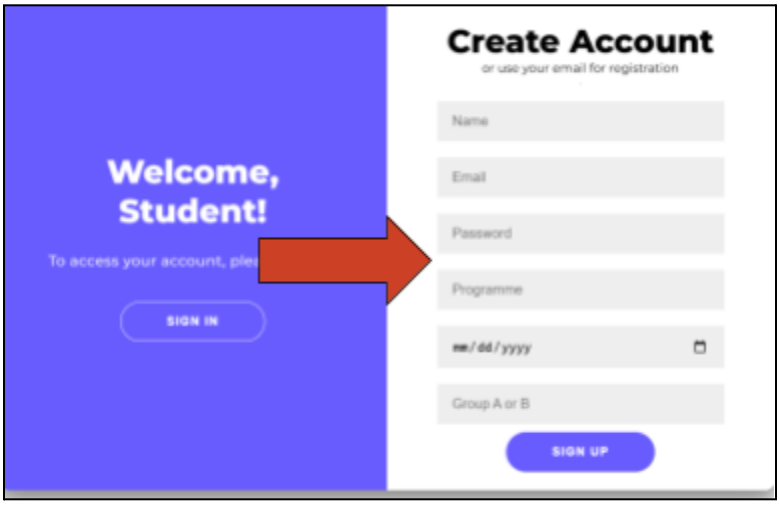
Step 1: Access the Sign-Up Page

- If using the website for the first time, open the system and click on SIGN UP AS STUDENT or SIGN UP AS TEACHER on the welcome screen.



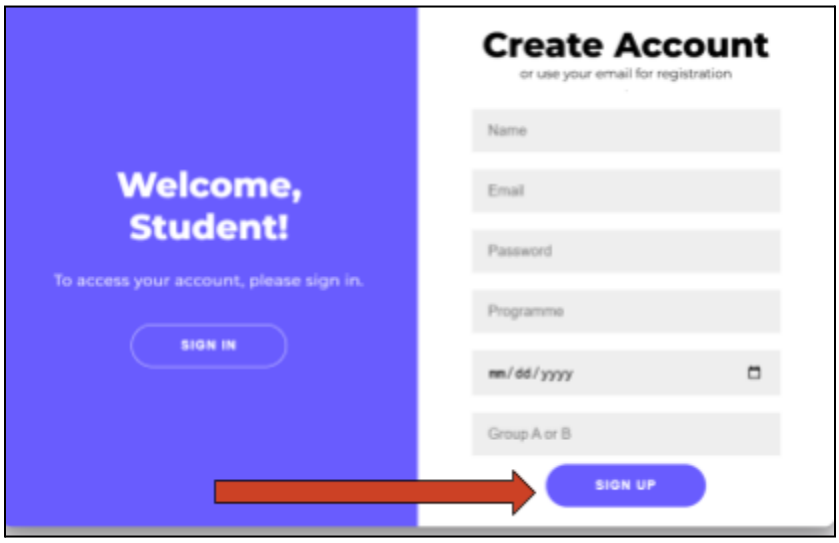

Step 2: Fill in Your Details

- On the Create Account page, enter the following information:
    - Name: Your full name.
    - Email: Your email address.
    - Password: Create a secure password.
    - Programme: Select your programme (e.g., BCS for students).



Step 3: Complete the Sign-Up

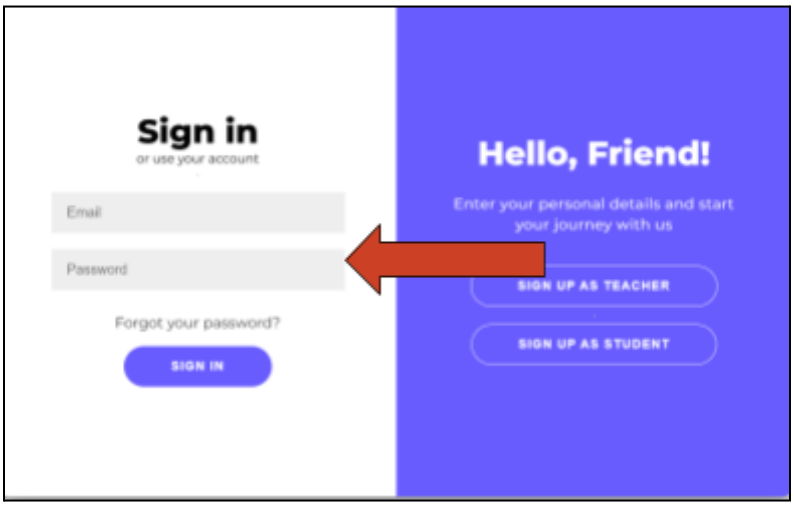
- Click the SIGN UP button to create your account.
- You will receive a confirmation message once your account is successfully created.




2. **Log In to Your Account**

Step 1: Access the Log-In Page

- Open the system and click on Sign In on the welcome screen.




Step 2: Enter Your Credentials

- On the Sign In page, enter the following details:
    - Email: The email address you used during sign-up.
    - Password: Your secure password.

Step 3: Access Your Account

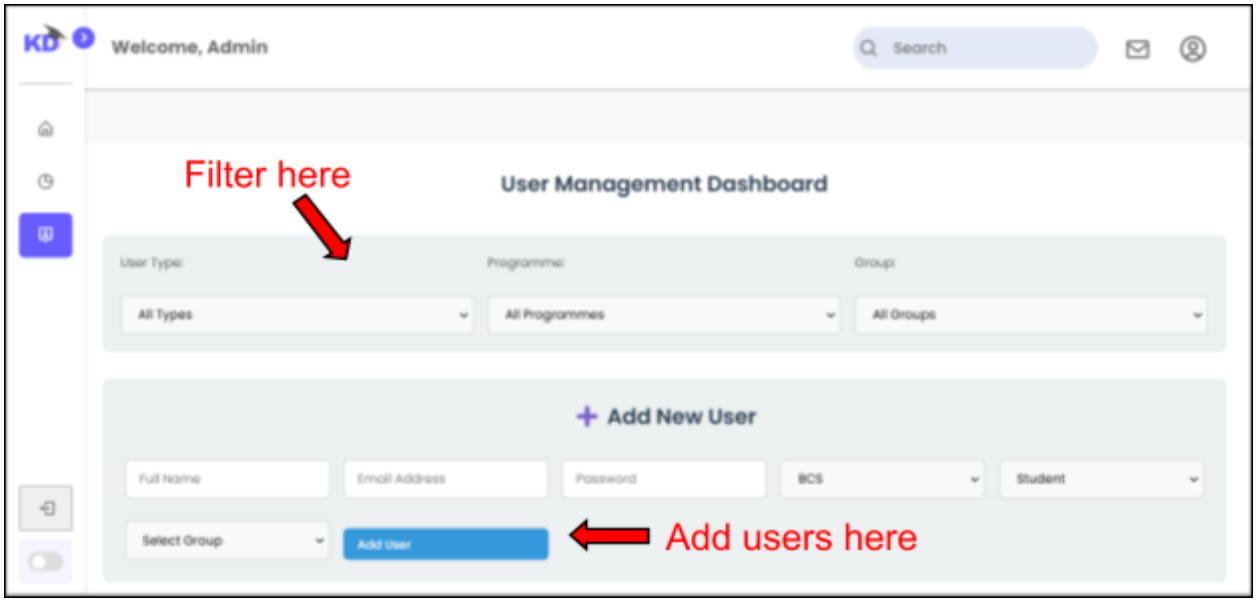
- Click the Sign In button to log in.
- You will be redirected to your Dashboard based on your role (Student or Teacher).

**How to register courses (for students)?**

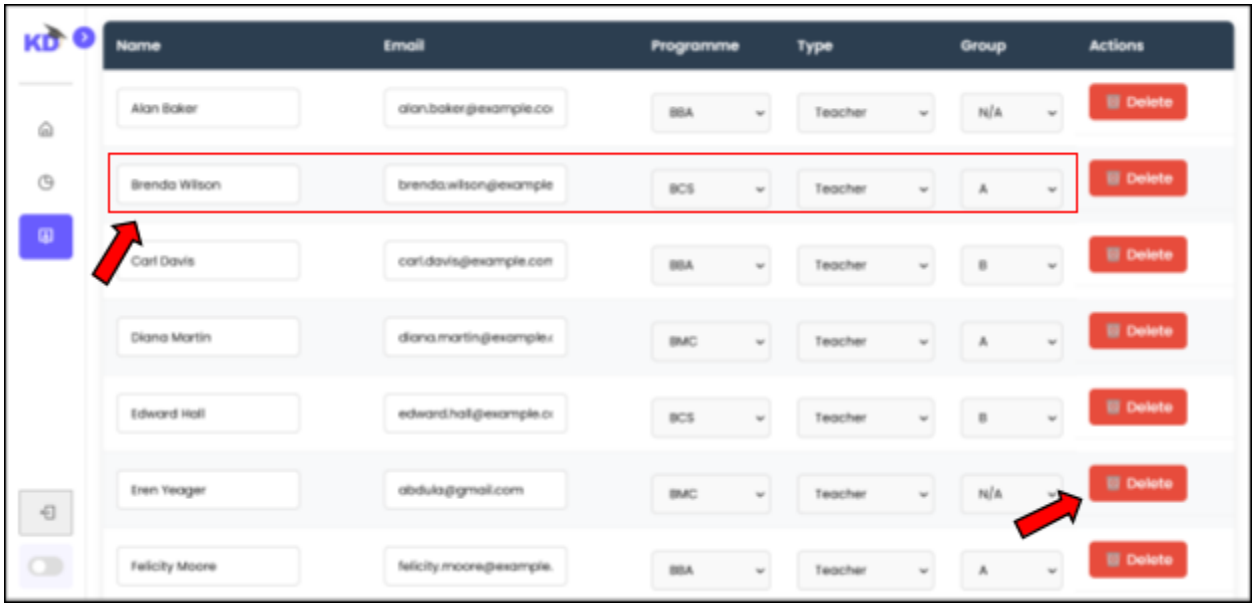
1. Sign in and click **‘course registration**’ tab on the left (sidebar)
2. Click on any courses you want to register and click the **‘Register Selected Courses**’ button.
3. If you want to register for semester 2, click on the **‘semester 2**’ button and register.
4. You can download the pdf of courses you have registered using the **‘Download PDF Summary**’ button.


**How to add or delete users (admin)**

1. Filter the users by selecting **User Type, Programme or Group**.
2. Enter the users’ details and click the **‘Add User**’ button to add new users.
3. Modify the data of existing users by simply clicking on specific fields and clicking the **‘save changes’** button at the end to save the changes in the database.
4. Delete the users by clicking the **‘delete’** button on the extreme right.



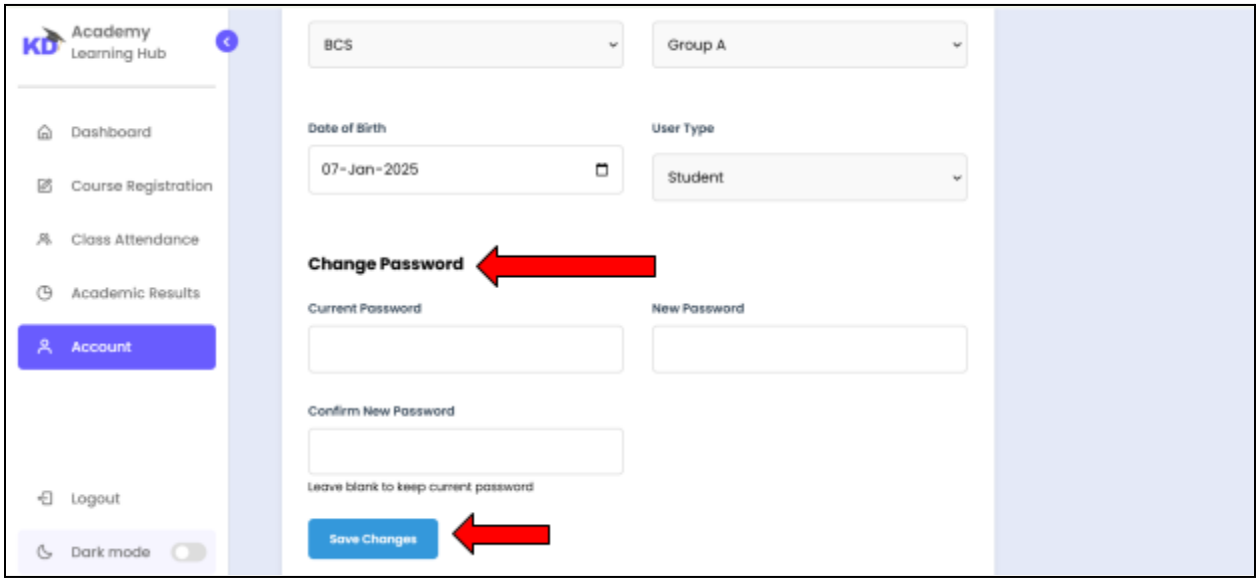

Filter out users based on the user type, whether students or teachers. Or filter using group A or B. Or filter using the Programme, whether BMC or BBA or BCS.




Click on any cells to edit or make changes. Or delete the specific user by clicking the red ‘Delete’ button on the right. Then click ‘save changes’ to update the changes.

## How to change the password

1. Go to **Account** section
2. There under **Change Password**, type in your current password and new password and confirm it.
3. Click the **‘Save Changes**’ button to save it



# Conclusion

The development of the KD Academy School Management System represents a significant step forward in modernizing the academy's academic and administrative processes. By transitioning from manual, time-consuming methods to a centralized, automated platform, the system successfully addresses key needs such as course enrollment, attendance tracking, academic performance monitoring, and user management. The integration of PHP and MySQL for backend development ensured a robust and scalable system, while the use of HTML, CSS, JavaScript, and Bootstrap for the frontend provided a responsive and user-friendly interface. Rigorous unit testing validated the system's reliability, usability, and performance, ensuring it meets the project's objectives and prepares the academy for future growth.

The system's role-based access control, intuitive dashboards, and streamlined workflows have significantly improved efficiency for students, teachers, and administrators. By prioritizing simplicity, security, and usability, the project aligns with the functional and non-functional requirements outlined in the **Software Requirements Specification (SRS)**, delivering a solution that enhances the overall educational experience at KD Academy.

**Recommendations**

To further enhance the system's capabilities and ensure its long-term success, the following recommendations are proposed:

1. **Expand System Integrations**:
    - Integrate third-party tools such as payment gateways for tuition fees and video conferencing platforms to support online learning and financial transactions.
2. **Mobile Application Development**:
    - Develop a mobile app version of the platform to provide students and staff with on-the-go access, ensuring real-time connectivity and convenience.
3. **Advanced Analytics and Reporting**:
    - Implement data analytics tools to provide insights into student performance trends, course popularity, and attendance patterns, enabling data-driven decision-making for administrators.

4. **Enhanced Security Features**:
    - Introduce advanced security measures such as multi-factor authentication and real-time monitoring to safeguard sensitive user data and prevent unauthorized access.
5. **Scalability for Future Growth**:
    - Adopt containerization technologies like Docker and consider a microservices architecture to ensure the system can scale efficiently as the academy expands.
6. **User Training and Documentation**:
    - Provide comprehensive user manuals and training sessions to ensure all stakeholders can effectively utilize the platform's features with minimal support.
7. **Feedback Mechanism**:
    - Introduce a feedback system where students and teachers can provide suggestions for improving the platform, ensuring continuous enhancement based on user needs.
8. **Automated Notifications**:
    - Implement automated email or SMS notifications for important updates such as attendance alerts, grade postings, and course registration deadlines.

By implementing these enhancements, the KD Academy School Management System can continue to evolve as a cutting-edge solution, meeting the academy's current needs while adapting to future challenges in the educational landscape. This will ensure the platform remains a valuable tool for students, teachers, and administrators alike.

# REFERENCES (APA STYLE)


Bootstrap Team. (2023). *Bootstrap documentation*. Retrieved from https://getbootstrap.com/docs/5.3/getting-started/introduction/

Figma. (2023). *Figma: The collaborative interface design tool* [Software]. Retrieved from https://www.figma.com

GitHub. (2023). *GitHub: Where the world builds software* [Software]. Retrieved from https://github.com

MySQL. (2023). *MySQL 8.0 reference manual*. Oracle Corporation. Retrieved from https://dev.mysql.com/doc/refman/8.0/en/

PHP Group. (2023). *PHP: Hypertext Preprocessor*. PHP Manual. Retrieved from https://www.php.net/manual/en/

W3Schools. (2023). *JavaScript tutorial*. Retrieved from https://www.w3schools.com/js/

W3Schools. (2023). *PHP tutorial*. Retrieved from https://www.w3schools.com/php/

World Wide Web Consortium. (2023). *HTML5: A vocabulary and associated APIs for HTML and XHTML*. W3C Recommendation. Retrieved from https://www.w3.org/TR/html52/

Zakas, N. C. (2012). *Maintainable JavaScript*. O'Reilly Media.

Agile Alliance. (2001). *Manifesto for Agile software development*. Retrieved from https://agilemanifesto.org/

GeeksforGeeks. (2023). *Introduction to MySQL*. Retrieved from https://www.geeksforgeeks.org/introduction-to-mysql/