dell

[Company name]  [Company address]

project :web (university system)

Professor : Dr omar shaalsh

Teaching Assistant : Eng Aya Abd Elhameed

Team member: kareem gaber adbalrady

Reg num: 221003403

**Introduction**

I maked project about university system by html , css langages

This project is about system develop by web programming

This project a contain 6 codes for 6 pages page1 is about sign in

In this page asks the user about his name and password and

there option if new user sign up , page 2 is about sign up

In this page ask new user about his name , email, password ,

Page 3 is about dashboard in this page there 4 options to

choose user manage course, manage student, manage

enrollement ,sign out; page 4 is about manage course in this

page ask user about course name , department course ;

page 5 is about manage student in this page ask user about his

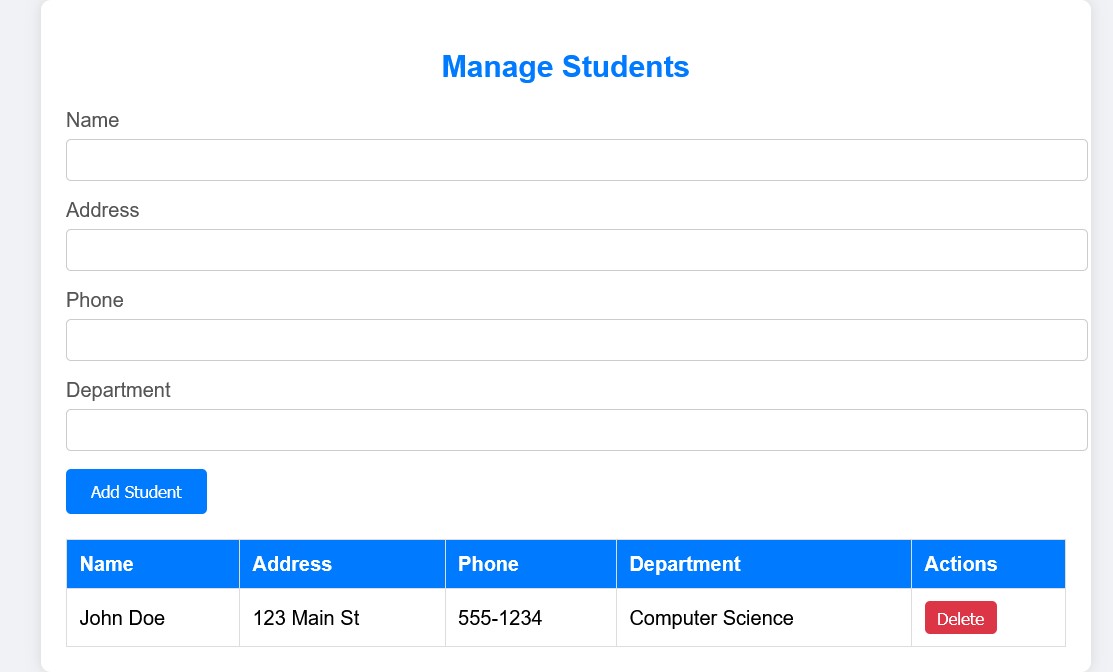
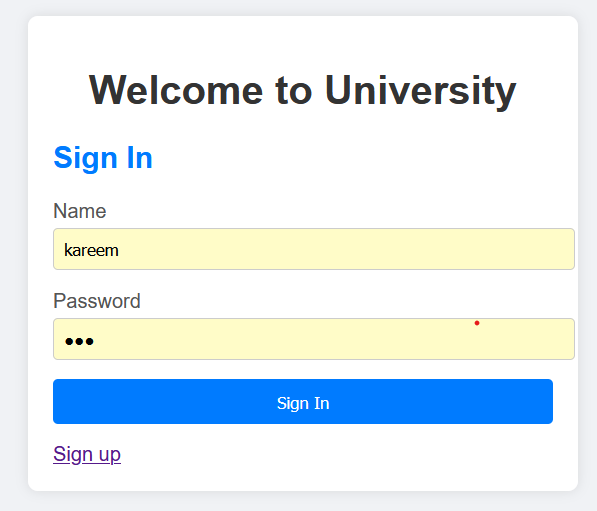
name , address ,phone number ,department; page 6 is about

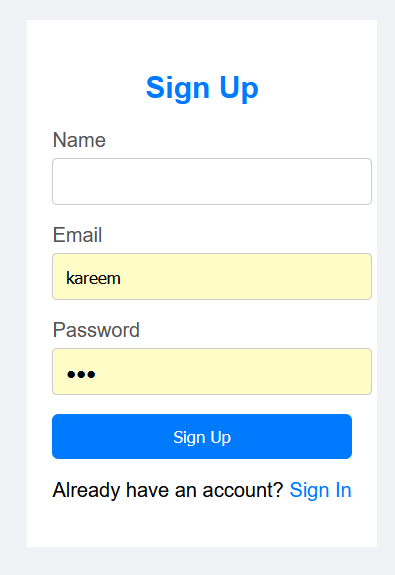
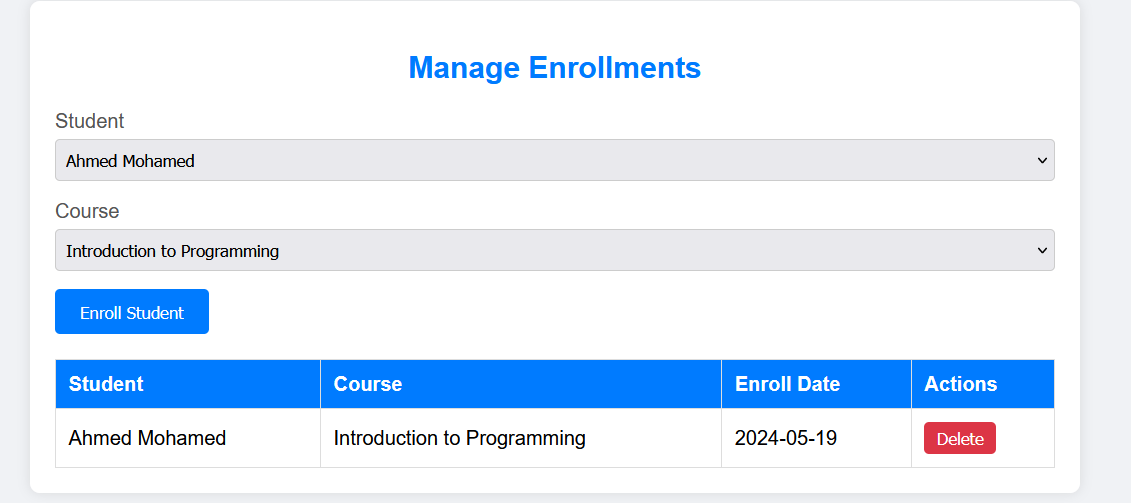
manage enrollement ask user about his name and course ,

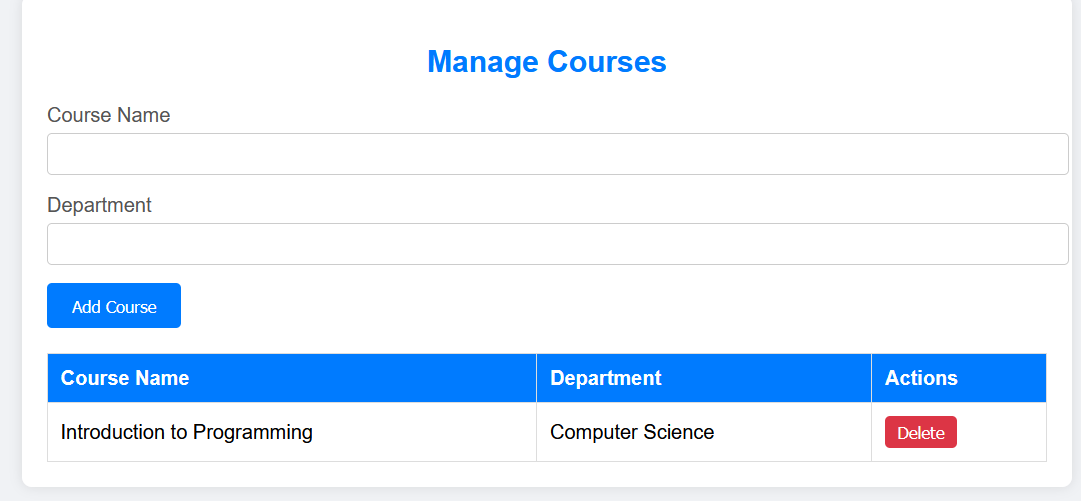
page 7 about sign out is about display for user you sign out and

button ask user if he want return to sign in page .

page 1:





****

**Objectives**

The main goal of the project is to facilitate application to universities for students and the university, and this method is much better and easier than traditional methods, and also easier for the university than traditional methods in many things.

**Sequence diagram**

**User University System Database**

**| | |**

**| Login request | |**

**| | |**

**| | Verify credentials |**

**| | |**

**| | |**

**| | Authentication |**

**| | |**

**| Authentication | |**

**| | |**

**| | |**

**| Redirect to | |**

**| Dashboard | |**

**| | |**

**| | |**

**Managing students**

**Admin User University System Database**

**| | |**

**| Add student | |**

**| | |**

**| | Validate input |**

**| | |**

**| | |**

**| | Insert into DB |**

**| | |**

**| | |**

**| | Confirmation |**

**| | |**

**| Confirmation | |**

**| | |**

**| | |**

**Functional Requirements:**

**User :**

- Users should be able to log in with their username and password.

- The system should verify the credentials and grant access based on the user's role (admin or student).

**Dashboard:**

- After successful login, users should be directed to a dashboard.

- The dashboard should provide links to manage students, courses, and enrollments.

- There should be a sign-out button for users to log out securely.

**Manage Students:**

- Admin users should be able to add new students with their details.

- Admin users should have the ability to delete student records.

- A table listing all students should be displayed with their information.

**Manage Courses:**

- Admin users should be able to add new courses with their details.

- Admin users should have the ability to delete course records.

- A table listing all courses should be displayed with their information.

**Manage Enrollments:**

- Admin users should be able to enroll students in courses.

- A table listing all enrollments should be displayed, showing which student is enrolled in which course along with the enrollment date.

**Student Profile:**

- Users should be able to view the details of a single student, including their enrolled courses.

- Users should have the ability to edit student information.

**Non-Functional Requirements:**

**Security:**

- User authentication and data transmission should be encrypted to ensure security.

- Access to certain functionalities, such as deleting student records, should be restricted to admin users only.

**Usability:**

- The user interface should be intuitive and user-friendly to facilitate easy navigation.

- Forms for adding new students/courses/enrollments should include validation to prevent incorrect data entry.

**Performance:**

- The system should be responsive and capable of handling multiple simultaneous user requests without significant delays.

- Database queries should be optimized for efficient data retrieval and storage.

**Reliability:**

- The system should be reliable, with minimal downtime and robust error handling mechanisms in place.

- Data integrity should be maintained, ensuring that student, course, and enrollment records remain accurate and consistent.

**Scalability:**

- The system should be designed to accommodate future growth, with the ability to scale up resources as needed to support additional users and data.

**Format**

**University System HTML Format Documentation**

**DOCTYPE Declaration:**

- `<!DOCTYPE html>`: Specifies the HTML5 document type.

**HTML Structure:**

- `<html lang="en">`: Defines the root element of the HTML document with the language set to English.

**Head Section:**

- `<head>`: Contains meta-information about the HTML document.

- Meta Tags: Defines character encoding (`charset`) and viewport settings (`viewport`).

**Title:**

**- `<title>`: Sets the title of the HTML document displayed in the browser tab.**

**Style Section:**

- `<style>`: Defines internal CSS styles for the document.

- Body Styling: Sets font family, background color, and flexbox properties for layout.

- Container Styling: Defines styles for the main content container, including width, background, padding, border radius, and box shadow.

- Heading Styling: Specifies styles for headings, including text alignment and color.

- Form Group Styling: Sets margin and padding for form groups and labels.

- Input Styling: Defines styles for input fields, including width, padding, border, and border radius.

- Button Styling: Specifies styles for buttons, including width, padding, background color, border, border radius, and cursor.

- Link Styling: Defines styles for links, including color and text decoration.

**Body Section:**

- `<body>`: Contains the visible content of the HTML document.

- Container Div: Wraps the content in a container with specified styles for layout and appearance.

- Heading: Displays a centered heading for the sign-in or sign-up page.

- Form: Includes input fields for user authentication (sign-in) or registration (sign-up), with appropriate labels and placeholders.

- Submit Button: Allows users to submit the form data.

- Link to Other Page: Provides a link to navigate to the sign-in page if the user is already registered (sign-up page) or to the sign-up page if the user doesn't have an account (sign-in page).