



SOFTWARE PROGRAMMING

Instructions

- Complete all tasks below.
- Use HTML, CSS, PHP, SQL and JavaScript for your implementation.
- **TOOLS:** Notepad++, Xampp Server, Internet browser
- **TOPIC:** SOFTWARE DEVELOPMENT METHODOLOGIES

Submission Requirements

- Submit a zip file containing:
 - HTML, CSS, PHP files
 - SQL scripts for database creation
 - Any additional resources (images, etc.)
- Ensure your code is well-commented and follows best practices.

QUESTION

You have been appointed as Software Developer at Kampala Parents' Primary School;

- i. Develop a Student Registration Software that have a Following Features:

TASK 1: User Interface Design (HTML & CSS)

Create the user interface for the Student Registration Software.

1. **Registration Form:**
 - Create an HTML form for student registration. The form should have the following fields:
 - Full Name (text input)
 - Date of Birth (date input)
 - Gender (radio buttons: Male, Female, Other)
 - Class (select dropdown)
 - Guardian's Name (text input)

- Email (email input)
- Phone Number (tel input)
- Address (textarea)
- Use CSS to style the form. Ensure that it is visually appealing and responsive.

TASK 2: Client-Side Validation

Implement client-side validation to ensure that the fields are filled out correctly before submission.

1. Validation Rules:

- Ensure that all required fields are filled.
- Validate that the email format is correct.
- Validate that the phone number contains only numbers.
- Show appropriate error messages for invalid fields.

TASK 3: Server-Side Scripting (PHP)

Write PHP code to handle form submission.

1. Processing Registration:

- Capture the form data submitted by the user.
- Validate the data on the server side to ensure no errors were bypassed.
- Store the valid student data in a MySQL database.

2. Create a Confirmation Page:

- After successful registration, redirect the user to a confirmation page that displays their details.

TASK 4: Database Setup (SQL)

Set up a MySQL database to store student data.

1. Database Design:

- Create a database named `school`.
- Create a table named `students` with the following fields:
 - `id` (INT, AUTO_INCREMENT, PRIMARY KEY)
 - `full_name` (VARCHAR)
 - `date_of_birth` (DATE)
 - `gender` (ENUM: 'Male', 'Female', 'Other')
 - `class` (VARCHAR)
 - `guardian_name` (VARCHAR)
 - `email` (VARCHAR)
 - `phone_number` (VARCHAR)
 - `address` (TEXT)

TASK 5: Data Retrieval (PHP & SQL)

Write PHP code to retrieve and display registered students.

1. Display Registered Students:

- Create a page that retrieves all students from the `students` table.
- Display student information in a table format.
- Include options to edit or delete student records.

TASK 6: Edit and Delete Functionality (PHP)

Implement features to edit and delete student records.

1. Edit Student Functionality:

- Create an "Edit" link next to each student record.
- Create an edit form that pre-fills the data for the selected student.
- Upon submission of the edit form, update the student data in the database.

2. Delete Student Functionality:

- Create a "Delete" link next to each student record.
- On clicking the delete link, prompt the user for confirmation.
- If confirmed, delete the student record from the database and refresh the student list.

END