

SOFTWARE DEVELOPMENT LIFE CYCLE (SDLC)

My Project: Student Course Registration System

1. The Planning Phase

The planning phase involves identifying the problem and proposing a solution. Many students face challenges during course registration, such as manual record keeping, errors in course selection, and lack of immediate feedback. To address this, the Student Course Registration System was planned as a simple web-based application that allows students to enter their details and register for courses electronically.

The goal of my project is to:

- Simplify the course registration process
- Reduce errors associated with manual registration
- Provide a basic system that can later be expanded

At this stage, the scope of the project was limited to a frontend-only solution using HTML, internal CSS, and JavaScript to ensure simplicity and clarity.

2. Requirements Analysis Phase

In this phase, the functional and non-functional requirements of the system were identified.

Functional Requirements

- The system should allow students to enter their name.
- The system should allow students to enter their matriculation number.
- The system should allow students to select a course from a predefined list.
- The system should display registered students and their selected courses in a table.
- The system should prevent empty submissions.

Non-Functional Requirements

- The system should be easy to use and understand.
- The interface should be visually clear and well-structured.
- The system should respond immediately to user actions.
- The system should run on any modern web browser.

3. System Design Phase

The design phase defines how the system will look and operate.

System Architecture

The system follows a simple single-page architecture consisting of:

- HTML for structure
- Internal CSS for styling
- JavaScript for functionality

All styling rules are written inside the <style> tag within the HTML file, ensuring internal styling as required.

Design Components

- Form Section: For student data entry
- Course Selection Section: Dropdown menu for courses
- Registration Table: Displays registered students

Naming Consistency

The following names were used consistently in both design and implementation:

1. Student
2. Course
3. Registration

This ensures clarity and alignment between the system design and actual code implementation.

4. Implementation Phase

In this phase, the actual system was developed based on the design.

- HTML was used to create the form elements, table, and page layout.
- Internal CSS was used to style the form, buttons, and table for better user experience.
- JavaScript was used to handle form submission, validate inputs, and dynamically add student records to the registration table.

The system logic ensures that when a student submits the form:

1. The input values are collected.
2. A new row is created in the registration table.
3. The student's data is displayed instantly.
4. The form is reset for the next registration.

5. Testing Phase

1. Testing was conducted to ensure the system works as intended.
2. Testing Activities

3. Tested form submission with valid inputs.
4. Tested submission with empty fields to ensure validation works.
5. Verified that registered courses appear correctly in the table.
6. Checked compatibility across different browsers.

Any errors identified during testing were corrected to improve system reliability and usability.

6. Deployment Phase

After successful testing, the system was deployed.

- The HTML file can be opened directly in any web browser.
- The project files can be uploaded to GitHub for version control and submission.
- Deployment requires no additional server or database since it is a frontend-only system.

This makes the system lightweight and easy to demonstrate.

7. Maintenance Phase

The maintenance phase focuses on future improvements and fixes.

Possible maintenance activities include:

- Adding data storage using local storage or a database.
- Enhancing validation rules.
- Improving the user interface.
- Extending the system to include backend technologies such as PHP or Flask.

Regular updates ensure the system remains functional and relevant.

Conclusion

The Student Course Registration System successfully follows all stages of the Software Development Life Cycle. From planning to maintenance, each phase was carefully implemented using HTML with internal styling and JavaScript. The project demonstrates a clear understanding of SDLC principles and provides a solid foundation for future enhancement.