

**DC 101 Project: CSS Student
Information System**

DOCUMENTATION PROJECT REPORT

Prepared by Zeus Sevilla

KLL BSCS 2C

Final Project for DC 101

January 4, 2026



Edit with WPS Office

INTRODUCTION

The CCS Student Information System (SIS) is a web-based application designed to manage and organize student records for the College of Computer Studies (CCS). The system aims to provide an efficient way to store, retrieve, update, and manage student-related data such as personal information, department, year and section, and enrolled courses.

The primary purpose of the system is to replace manual or spreadsheet-based record keeping with a centralized digital platform. This improves data accuracy, accessibility, and consistency while reducing redundancy and human error. The system also allows administrators to manage departments, sections, and course subjects in an organized manner.



Edit with WPS Office

Web Interface: Key Pages and Functionalities

1. Student Management Page

Features:

- Displays a table of all students
- Add, edit, and delete student records
- Search students by name or student ID
- Sort student data in ascending or descending order
- Dropdown selections for Department (ACT / BSCS) and Year & Section

2. Department Management Page

Features:

- Displays CCS departments (ACT and BSCS)
- Filters students based on department
- Manages Year & Section entries
- Add and delete sections dynamically
- Organized viewing of students per department

3. Course Management Page

Features:

- Displays list of courses in a table format
- Add, edit, and delete course records
- Stores course ID, name, description, and credit units
- Provides a structured way to manage academic subjects

3. Setting Management Page

Features:

- Controls Dark/Light Theme



- Manage Accounts and Password
- Logout and Clear LocalData

3. Login and Home Page

Features:

- Displays Login ID and Password
- Displays Main Page for Student, Department, Course and Setting Redirectories

Database Design: ER Diagram, Table Description, and Relationships

ER Diagram

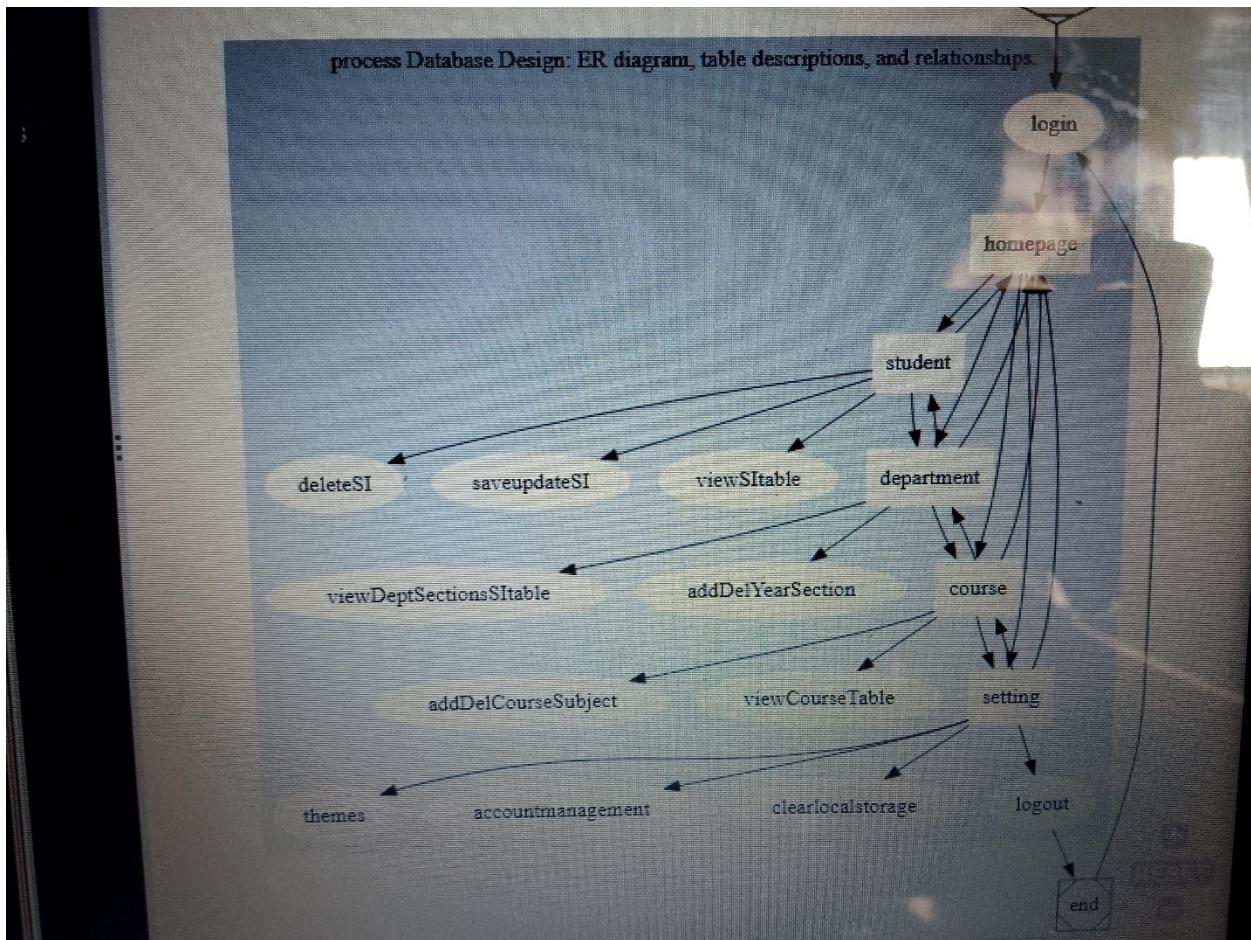


Table Descriptions

1. Student Management Page

Field Name	Data Type	Description
student_id	VARCHAR	Unique Student Identifier
last_name	VARCHAR	Student lastname
first_name	VARCHAR	Student name
middle_initial	CHAR	Student middle initials
department	VARCHAR	BSCS or ACT
year_section	VARCHAR	Year and Section
gender	VARCHAR	Gender
birth_date	DATE	Date of Birth
age	INT	Student Age
email	VARCHAR	Email address
contact	VARCHAR	Contact number
address	TEXT	Home address

2. Department Table

Field name	Data Type	Descriptions
department_id	VARCHAR	Department code (BSCS or ACT)
Department_na	VARCHAR	Department name



me		
----	--	--

3. Year and Section Table

Field Name	Data Type	Description
section_id	INT	Unique section id
Department_id	VARCHAR	Linked department
Year_section	VARCHAR	Ex: 1A, 2C, 3B

4. Course Table

Field Name	Data Type	Description
course_id	VARCHAR	Course code
Course_name	VARCHAR	Course name
description	TEXT	Course Description
data_units	INT	Number of unit

Relationships

- A Department can have many Students
- A Department can have many Year & Sections
- A Student belongs to one Department and one Year & Section



- A Course is independent and can be associated with departments if extended in the future



Challenges and Learning:

1. Challenges Encountered

One of the main challenges encountered was managing data relationships between students, departments, and year sections while ensuring data consistency. Another challenge was implementing search and sorting functionality efficiently using JavaScript, compiling and sorting the students html information data in each department and year & section in department html and using SQL. Designing a clean and user-friendly interface while maintaining functionality also required careful planning.

2. Learned Outcomes

Through this project, significant learning was gained in:

- Designing relational databases and ER diagrams
- Implementing CRUD (Create, Read, Update, Delete) operations
- Using HTML, CSS, and JavaScript for dynamic web applications
- Managing data persistence using localStorage
- Improving UI/UX design principles
- Understanding how real-world information systems are structured

This project enhanced problem-solving skills and provided practical experience in developing a functional web-based information system.

3. Conclusion

The CCS Student Information System successfully demonstrates how a web-based application can efficiently manage academic records. It serves as a strong foundation for future improvements such as database integration, authentication, and role-based access control. The project highlights the importance of proper system design, data organization, and user-centered development.





