Software Development for BUSKARO Registration Management System



by

Hamza Yaseen BCS223031

Syed Adeen BCS223088

Muhammad Ahmed BCS223048

A Project Report submitted to the Department Of Computer Science

in partial fulfillment of the requirements for the course of Introduction To Database

Faculty of Computing

Capital University of Science & Technology, Islamabad

January 2024

Copyright  2024 by CUST Student

All rights reserved. Reproduction in whole or in part in any form requires the prior written permission of Hamza Yaseen, Syed Adeen Sarosh, and Muhammad Ahmed or designated representative.

# 

# Abstract

The focus of this report is on developing a "Software Management System for BUSKARO Management System" tailored specifically for university students. The primary goal is to facilitate students in organizing and managing their daily tasks, schedules, and academic performance. This sophisticated management system will be designed with an intuitive database and user interface using HTML (Hyper Text Markup Language) and CSS (Cascading Style Sheets), ensuring a seamless user experience. The software will store and manage information about students' courses, assignments, exam schedules, and extracurricular activities. Furthermore, it will allow students to track their progress and optimize their study plans. In conclusion, the BUSKARO software management system offers an innovative solution for university students to manage their academic lives more efficiently. The implemented user-friendly system serves as a valuable tool for maintaining organized records of students' activities, promoting improved academic performance, and fostering personal growth. Thus, by creating the BUSKARO software management system, we empower university students to successfully navigate their academic journey.

# 

# Table of Contents

# 

# List of Figures

[*Figure 1 11*](#_bookmark22)

[*Figure 2 12*](#_bookmark24)

# List of Acronyms/Abbreviations

|  |  |
| --- | --- |
| CSS | Cascading Style Sheets |
| PHP | Hypertext Preprocessor |
| SQL | Sequential Query Language |
| HTML | Hyper Text Markup Language |
| ER Model | Entity Relationship Model |
| CMD | Command Prompt |

# Chapter 1

# INTRODUCTION

The advent of technology has transformed numerous sectors by digitalizing various processes. One such aspect involves the handling of institutional transport systems, which have traditionally relied heavily on manual labor, leading to inefficiencies. This report elaborates on the design and implementation of a digital software system – "BUSKARO Transport Management System" – intended to streamline the process of transport registration and execution for university students. The core objective of this report is to provide insight into the project's inspiration, purpose, specifications, application, and execution.

## Overview

The "University Student Transport Management System" aims to provide an effective, efficient, and user-friendly platform to manage transportation processes, from student registration to accessing schedule details. The software seeks to replace outdated manual processes with a sophisticated digital system that effortlessly blends into the university's eco-system. It promises transparency and control to students by providing details such as bus schedules, routes, and driver information, while the management enjoys full access to operate and supervise the system.

## Project Idea

The driving force behind this project is the necessity for a unified and digitally controlled environment that eliminates the limitations of traditional transport systems within universities. It enables students to register for bus services digitally, pay their fees, stay updated with the bus schedules, routes, and driver details. In other words, the intention is to bring all stakeholders – students, drivers, and management – onto a single interconnected platform that synergizes their efforts rather than compartmentalizes them.

# REQUIREMENT SPECIFICATION AND ANALYSIS

Seeing that software development necessitates a clear understanding of system requirements from the start, this segment elucidates the specifications and examines the software requirements necessary for creating the University Student Transport Management System.

## Requirement Specification

The requirement specifications for the design of the transport management system involve the following:

* Robust Database System: An efficient, reliable, and secure database system to store and manage the information of registered students, transport details (buses, routes, schedules), payment details, and drivers’ information.
* User Interface: An intuitive and easy-to-navigate user interface for the students to access details including registration, timetable, drivers' info, and payment of fees. The interface should also facilitate the administrators/managers to review and approve registrations, manage and control buses, routes, schedules, and have access to all relevant information.
* Secure Payment Gateway: A secure and efficient payment processing system to handle fee transactions.
* Security Measures: Implementing appropriate privacy and data protection measures to safeguard sensitive user data.

## Software Requirement Analysis

In terms of software analysis:

* The system should be able to function seamlessly on different platforms such as Windows, Mac OS, and Linux for wider accessibility.
* It should be flexible enough to manage potentially thousands of users concurrently without performance degradation.
* The system should be able to integrate with existing institutional systems/database (like student management systems) for simpler implementation and maintenance.
* From a security perspective, the software should incorporate encryption techniques, firewall protection, and access control systems to ensure the information stored is protected from potential threats and privacy breaches.

By ensuring the above specifications and conduct thorough software requirement analysis, we can ensure that the proposed University Student Transport Management System effectively meets its purpose, providing an efficient and streamlined platform for managing the university's transportation services.

# TOOLS AND TECHNIQUES

## Hardware Tools used

The project is being developed using single hardware tool that is laptop. The BUSKARO registration system software is made only on the laptop. Windows 11 is in the laptop.

## Software, simulation tools used

The web-based software is used in this project. Web-based software allows users to connect with a remote server through a browser.

The whole BUSKARO database system will be opened in the browser and from there the end-user can perform data entry.

The main software used for this project is Xampp, CMD complier and PHP storm.

**CMD**

There are many other compilers for writing queries but CMD complier is used in this project.

All the tables are formed through this compiler. The process of updating or deleting any information can also be done using this compiler.

**XAMPP**

XAMPP is a cross-platform web server that is free and open-source. XAMPP is a short form for Cross-Platform, Apache, MySQL, PHP, and Perl. XAMPP is a popular cross-platform web server that allows programmers to write and test their code on a local webserver. It was created by Apache Friends, and the public can revise or modify its native source code. It includes MariaDB, Apache HTTP Server, and interpreters for PHP and Perl, among other computer languages.

• XAMPP is simply a local host or server.

• This local server runs on personal computer, whether it’s a desktop or a laptop.

• It is used to test clients or websites before publishing them to a remote web server.

• On a local computer, the XAMPP server software provides a suitable environment for testing MYSQL, PHP, Apache, and Perl projects. Because most real-world web server deployments share the same components as XAMPP, moving from a local test server to a live server is straightforward.

The main tools used for this project are HTML and CSS.

**HTML**

The HTML is used for giving a structure to the interface of the BUSKARO registration software. The HTML make-up language designs the web page or software on that web page.

**CSS**

For writing HTML language, PHP storm software is used. All the shape of interface is given through this tool.

CSS is another language which is used for styling the interface of the web page or the software on that web page. For writing CSS language, PHP storm software is also used here. In the project, the HTML and CSS files are merged together.

**Chapter 3**

**Project Design**

* 1. **Proposed Design Methodology**

The design methodology starts with the conceptual model. The conceptual model maps all data into organized form. It adds all the requirements for developing an efficient database. It explains the proper structure of the database before developing it. We have to make tables to store the data so to ensure proper connections between the relations or entities, this model is created. The database which is designed for the BUSKARO registration must contain correct data.

The integrity constraints and checks are also applied here so that the software user or the front-end user must add data correctly.

The proper arrangements are done in this model so that during the database design no database design there should be a proper space to enter every detail of data and no such details should be left for which there is no space to enter.

* 1. **Conceptual diagram of Software**

**Entities:**

Students - This entity represents university students who will use the system. Its attributes might include StudentID, Name, Email, Password, RegisteredBusID, etc.

Buses - This entity represents the buses involved in the university transportation system. Attributes could include BusID, Capacity, RouteID, DriverID, etc.

Drivers - This entity is for the bus drivers. Its attributes may include DriverID, Name, ContactInfo, BusID, etc.

Routes - This represents the different routes taken by the buses. Attributes for this entity could include RouteID, RouteName, StartPoint, EndPoint, BusID, etc.

Manager - This entity represents the university manager who manages the overall system. Attributes may contain ManagerID, Name, Email, Password, etc.

**Relationships:**

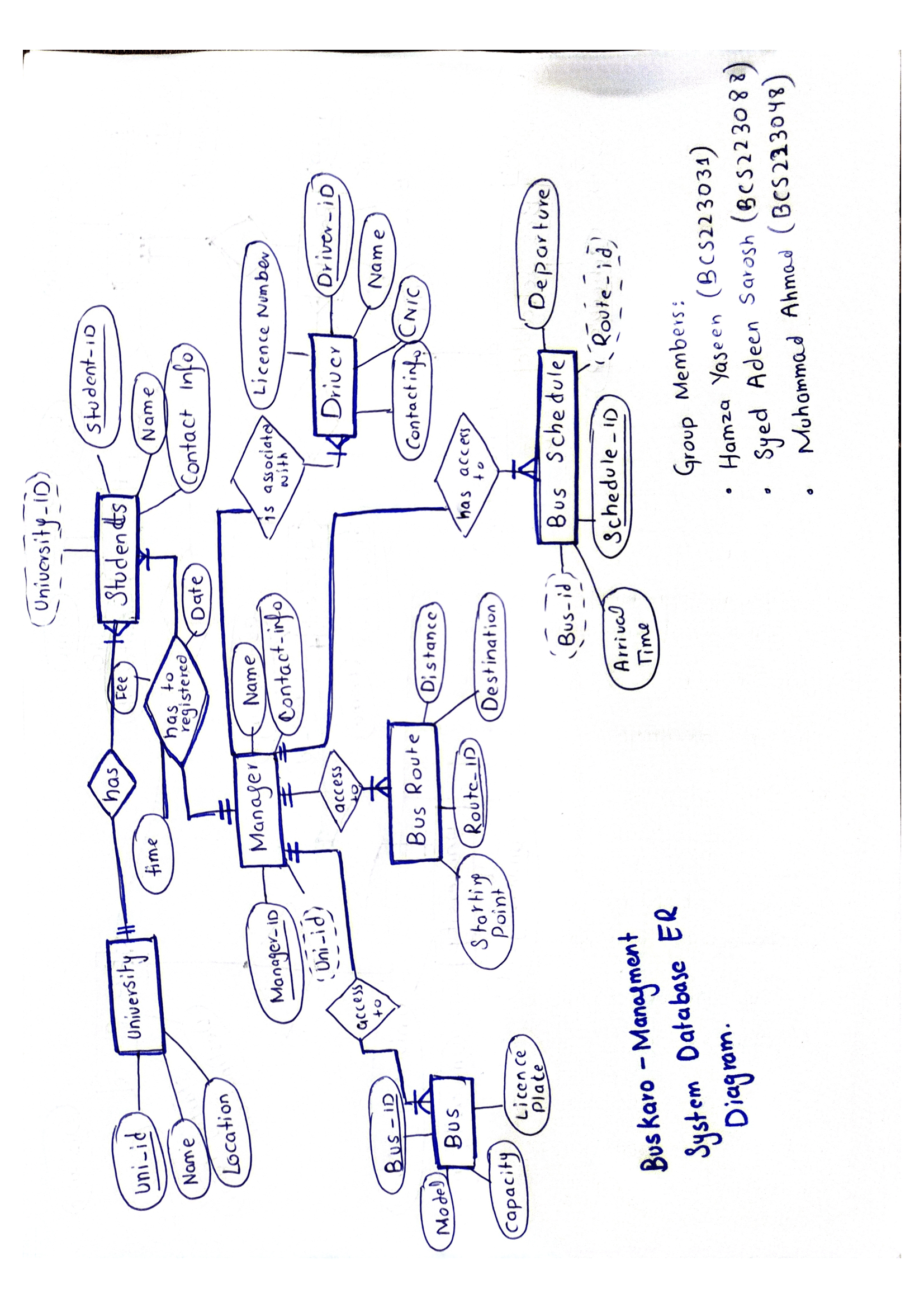
There would be relationships between these entities:

Students and Buses - A many-to-one relationship, as many students can register to one bus.

Buses and Drivers - Typically a one-to-one relationship, assuming each bus has exactly one driver.

Buses and Routes- A one-to-many relationship, as a bus can be assigned to one route, but one route can contain multiple buses.

Manager and other entities - The manager can manage (read, write, update, and delete) all entities. Hence, the relationship should be one-to-many between the Manager and other entities.



**Figure 1**

ER Diagram of Software

* 1. **Logical Design of the software**

When the conceptual model completed, we made the logical design[3] of ER diagram and make relational model.

**TABLES IN THE DATABASE:**

**Creating University Table:**

CREATE TABLE University (

Uni\_ID VARCHAR(255) PRIMARY KEY,

Name VARCHAR(255),

Location VARCHAR(255)

);

**Creating Students Table:**

CREATE TABLE Students (

Student\_ID VARCHAR(255) PRIMARY KEY,

Uni\_ID VARCHAR(255),

Name VARCHAR(255),

Contact\_Info VARCHAR(255),

FOREIGN KEY (Uni\_ID) REFERENCES University(Uni\_ID)

);

**Creating BUS Table:**

CREATE TABLE Bus (

Bus\_ID VARCHAR(255) PRIMARY KEY,

Manager\_ID VARCHAR(255),

Model VARCHAR(255),

Capacity INT,

LicensePlate VARCHAR(255),

FOREIGN KEY (Manager\_ID) REFERENCES Manager(Manager\_ID)

);

**Creating BUSROUTE Table:**

CREATE TABLE BusRoute (

Route\_ID VARCHAR(255) PRIMARY KEY,

Manager\_ID VARCHAR(255),

StartingPoint VARCHAR(255),

Distance INT,

Destination VARCHAR(255),

FOREIGN KEY (Manager\_ID) REFERENCES Manager(Manager\_ID)

);

**Creating BusSchedule Table:**

CREATE TABLE BusSchedule (

Schedule\_ID VARCHAR(255) PRIMARY KEY,

Bus\_ID VARCHAR(255),

Route\_ID VARCHAR(255),

Manager\_ID VARCHAR(255),

DepartureTime VARCHAR(255),

ArrivalTime VARCHAR(255),

FOREIGN KEY (Bus\_ID) REFERENCES Bus(Bus\_ID),

FOREIGN KEY (Route\_ID) REFERENCES BusRoute(Route\_ID),

FOREIGN KEY (Manager\_ID) REFERENCES Manager(Manager\_ID)

);

**Creating Driver Table:**

CREATE TABLE Driver (

Driver\_ID VARCHAR(255) PRIMARY KEY,

Manager\_ID VARCHAR(255),

DName VARCHAR(255),

DCNIC VARCHAR(255),

DContact\_Info VARCHAR(255),

DLicenseno VARCHAR(255),

FOREIGN KEY (Manager\_ID) REFERENCES Manager(Manager\_ID)

);

**Creating Manager Table:**

CREATE TABLE Manager (

Manager\_ID VARCHAR(255) PRIMARY KEY,

Student\_ID VARCHAR(255),

Bus\_ID VARCHAR(255),

Route\_ID VARCHAR(255),

Schedule\_ID VARCHAR(255),

Driver\_ID VARCHAR(255),

MName VARCHAR(255),

MContactInfo VARCHAR(255),

FOREIGN KEY (Student\_ID) REFERENCES Students(Student\_ID)

);

**Creating Registration Table:**

CREATE TABLE Registration (

Registration\_ID VARCHAR(255) PRIMARY KEY,

Student\_ID VARCHAR(255),

Manager\_ID VARCHAR(255),

Time VARCHAR(255),

Fee INT,

Date VARCHAR(255),

FOREIGN KEY (Student\_ID) REFERENCES Students(Student\_ID),

FOREIGN KEY (Manager\_ID) REFERENCES Manager(Manager\_ID)

);

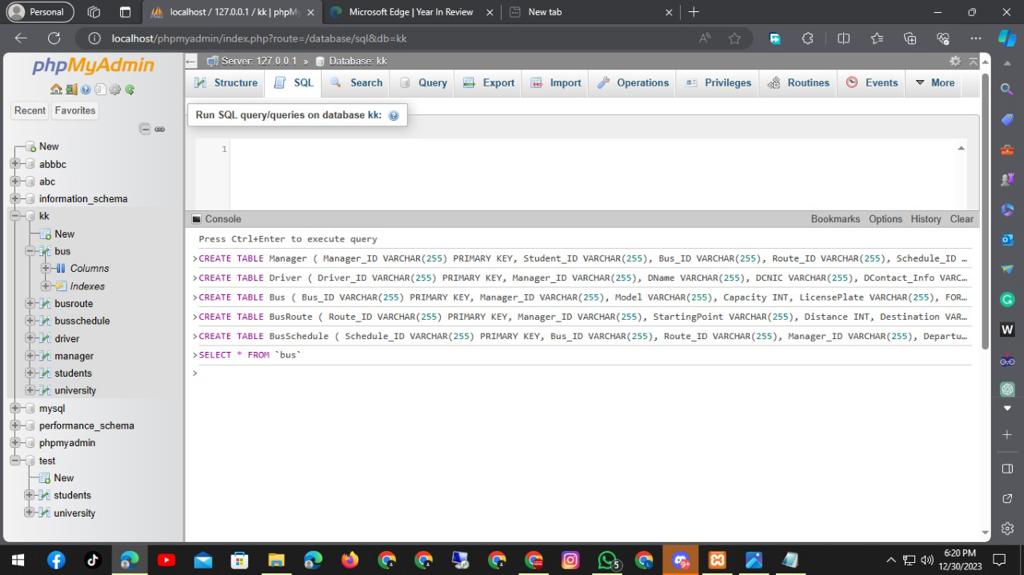
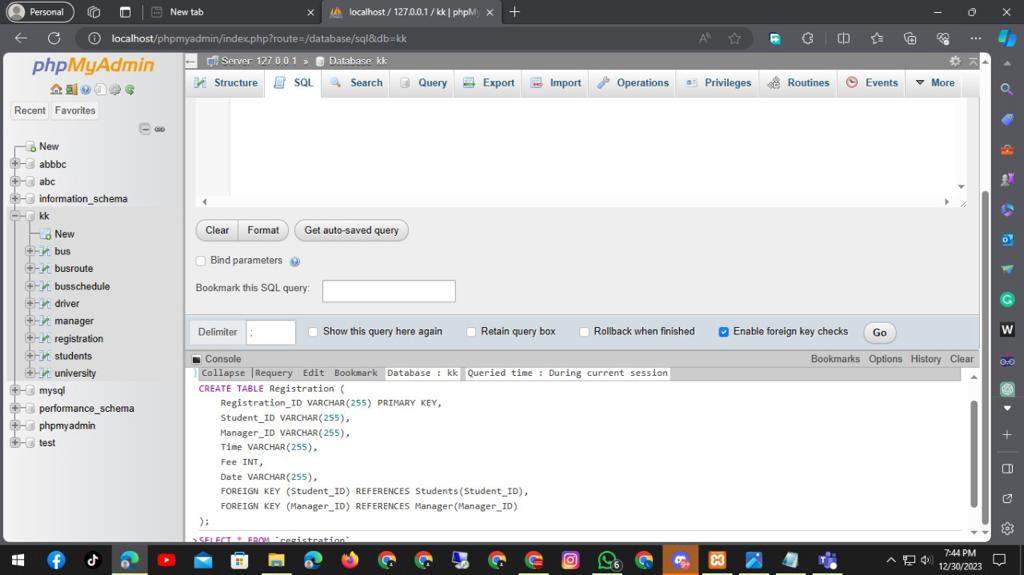
**NORMALIZATION:**

To Avoid the Anomalies We Create the 3rd Table Registration That Stores the Relationship Attributes of Students and Manager that are Time,Fees,Date and Put Registeration ID as a Primary key, Student ID and Manager ID as Foreign Key.

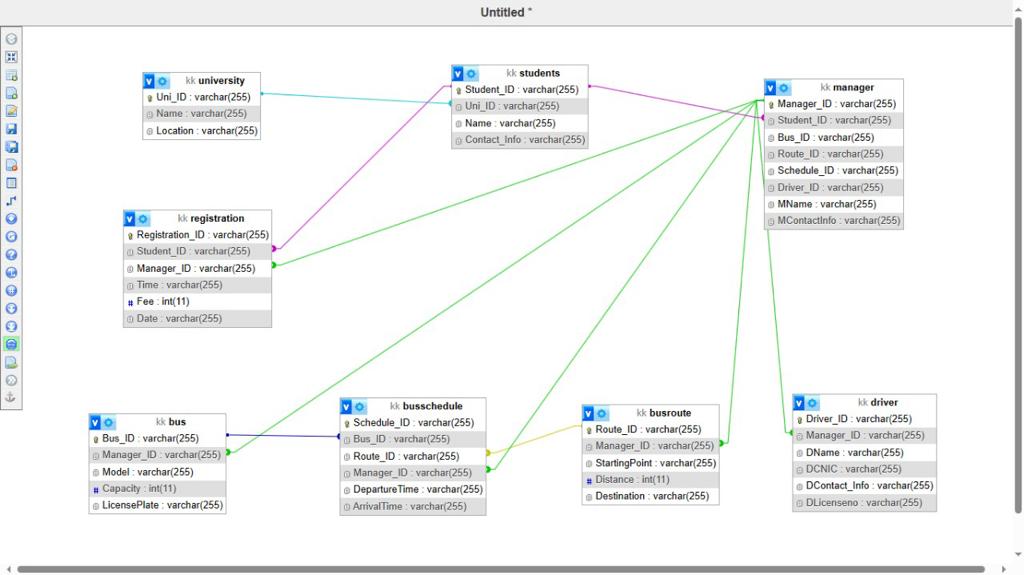
This way We Avoid Anomalies and Our Tables are in 3NF as checked By 1NF,2NF and 3NF conditions.

The Following is The Final Relational Model

**TABLES IN THE PHP ADMIN:**

****

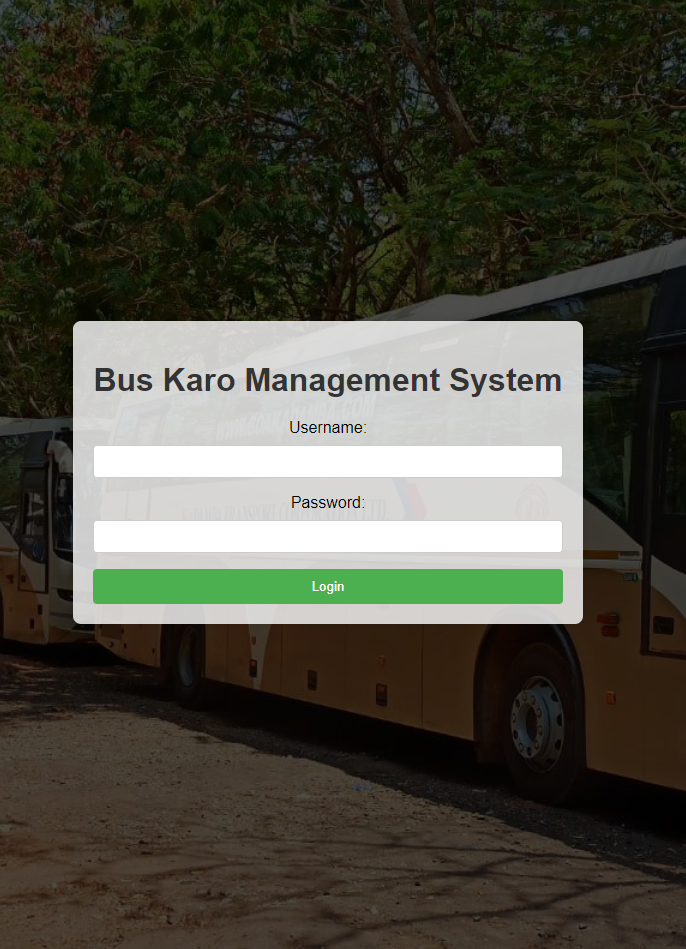
**FINAL RELATIONAL DESIGN**

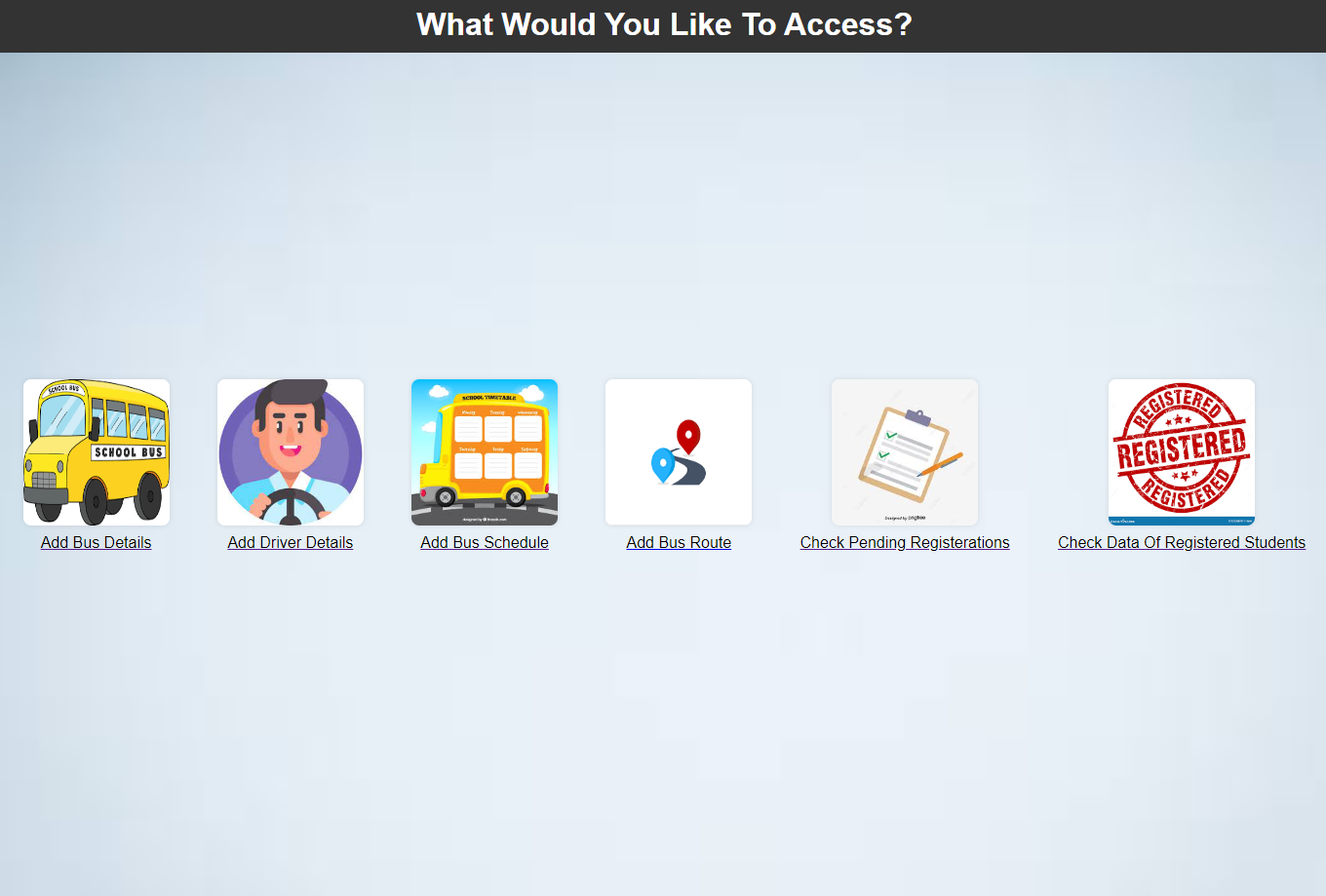
****

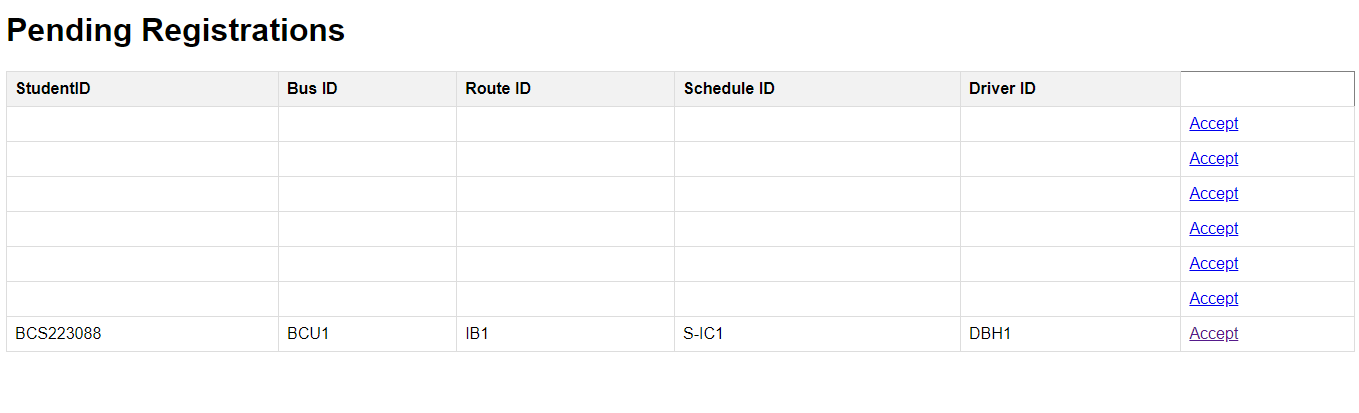
# Chapter 3

# Screenshots of the Frontend

# 







**CONCLUSION AND FUTURE WORK**

Our The project of the University Students Transportation Management System represents a significant milestone in streamlining and optimizing daily transport operations in an academic environment. The system contains features that not only prioritize the users' (students and management) needs and experiences but also improves existing operational workflows, acknowledges efficiency, and embraces digital transformation.

From the student registration to fee payments, bus scheduling, tracking bus routes and driver information, the system aims to handle these complex processes with sophistication and convenience. The project presents a promising alternative to conventional strategies, making the everyday commuting experience seamless for students and making management tasks less daunting for the university authorities.

**Future Work**

Looking into the future, there are a number of enhancements and additions that can be made to further optimize the University Students Transportation Management System:

Real-Time Bus Tracking: The addition of real-time bus tracking functionality can keep students up-to-date with current locations of buses and estimated pick-up times.

Integration with Payment Gateways: Direct integration with secure online payment gateways can be made for smoother and faster fee payments, eliminating the need for physical transactions.

Enhancements in User Interface: The user interface of the software can be constantly updated to keep it in sync with the latest design trends and to provide the best user experience.

Development of Mobile App: In addition to the web application, a mobile app for the system can also be developed which would provide greater accessibility to students and administrators.

Adopting AI for Optimized Routes: Artificial Intelligence can be used to determine the best routes for the buses, ensuring the shortest travel times and increased efficiency.

Integration with other University Systems: The system can be integrated with other university systems such as the student information system or academic scheduling system to share relevant data and improve overall operations.

The above proposed enhancements demonstrate potential directions for work on the project moving forward that will further enhance its users' experience while amplifying efficiency and process economy at the managerial level.

**REFERENCES**

1. A. Hoffer, “The Enhanced E-R Model,” in *Modern database management*, 12th ed. Jeffrey A. Hoffer, V. Ramesh Indiana and Heikki Topi, Ed. Harlow: Pearson, 2016, pp. 150-200.
2. M. Siegert et al. “Introduction to xampp,” Edubca, 2018 [Online] Available: <https://www.educba.com/what-is-xampp/> [Accessed: 29th December 2022].
3. “Overview of logical design,” Oracle, 2020 [Online] Available: <https://docs.oracle.com/cd/A81042_01/DOC/server.816/a76994/logical.htm> [Accessed: 29th December 2022].
4. “SQL Commands,” geeksforgeeks, 2022 [Online] Available: <https://www.geeksforgeeks.org/sql-ddl-dql-dml-dcl-tcl-commands/> [Accessed: 29th December 2022].
5. “HTML Introduction,” W3schools, 2022 [Online] Available: <https://www.w3schools.com/html/html_intro.asp> [Accessed: 30th December 2022].
6. A.oppel, *Databases A Beginner’s Guide,* New York: McGraw-Hill, 2009.
7. “CSS,” Wikipedia, 2022 [Online] Available: <https://en.wikipedia.org/wiki/CSS> [Accessed: 30th December 2022].
8. J. V. Petersen, *Absolute Beginner’s Guide to Databases,* USA: Que Pub, 2009.
9. “Entity relationship Diagram,” Tutorialspoint, 2022 [Online] Available: https:/[/www.tutorialspoint.com/dbms/er\_diagram\_representation.htm](http://www.tutorialspoint.com/dbms/er_diagram_representation.htm) [Accessed: 30th December 2022].
10. J. Murach, R. Harris, *Murach’s PHP and SQL,* 4th ed. USA: Mike Murach, 2022.