

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Jnana Sangama, Belagavi – 590 014



A Database Management System Mini Project Report On

“Bus Management System”

Submitted in Partial Fulfilment of the Requirements for the

V Semester of the Degree of Bachelor of Engineering

In

Computer Science & Engineering

By

SANJEEV KAMATH K - 4MW21CS084

SATHVIK K – 4MW21CS085

SHREESHA – 4MW21CS095

VIGHNESH S SHENOY – 4MW21CS117



Department of Computer Science and Engineering

**SHRI MADHWA VADIRAJA INSTITUTE OF TECHNOLOGY AND
MANAGEMENT**

Vishwothama Nagara, BANTAKAL – 574 115, Udupi District

2023-2024

SHRI MADHWA VADIRAJA INSTITUTE OF TECHNOLOGY AND MANAGEMENT

(A Unit of Shri Sode Vadiraja Mutt Education Trust @, Udupi)

Vishwothama Nagar, BANTAKAL – 574 115, Udupi District, Karnataka, INDIA

Department of Computer Science and Engineering

CERTIFICATE

Certified that the Database Management System Project Work titled ‘**Bus Management System**’ has been carried out by **Mr. Sanjeev Kamath K** (4MW21CS084), **Mr. Sathvik K** (4MW21CS085), **Mr. Shreesha** (4MW21CS095) and **Mr. Vighnesh S Shenoy** (4MW21CS117), who are the bonafide students of Shri Madhwa Vadiraja Institute of Technology and Management, in partial fulfillment for the award of Bachelor of Engineering in Computer Science and Engineering of Visvesvaraya Technological University, Belagavi during the year 2023-24. The Database Management System Mini Project Report has been approved as it satisfies the academic requirements with respect to the project work guidelines prescribed for the said Degree.

Dr. Soumya J Bhat

HOD

Dept. of CSE

Ms. Savitha A Shenoy

Assistant Professor (Sr)

Dept. of CSE

External Viva

**Name of the Examiners:
with Date**

Signature

1.

2.

Acknowledgements

We express our deepest gratitude and respect to **Dr. Soumya J Bhat**, Associate Professor and Head, Department of Computer Science and Engineering, and **Ms. Savitha Shenoy**, Assistant Professor (Sr), Department of Computer Science and Engineering for her encouragement while doing this project work.

We are indebted to Prof. **Dr. Thirumaleshwara Bhat**, Principal, for their advice and suggestions at various stages of the work.

We extend our thanks to the Management of **Shri Madhwa Vadiraja Institute of Technology and Management, Bantakal**, Udupi for providing good laboratory and library facilities. We also remain grateful for the co-operation and help rendered by the teaching and non-teaching staff of the Computer Science and Engineering Department.

Lastly, we take this opportunity to offer our regards to all of those who have supported us directly or indirectly in the successful completion of this project work.

Sanjeev Kamath K (4MW21CS084)

Sathvik K (4MW21CS085)

Shreesha (4MW21CS095)

Vighnesh S Shenoy (4MW21CS117)

ABSTRACT

The Bus Management System is a comprehensive database management system (DBMS) project designed to automate and streamline the operations of bus transportation services. This system serves as a digital platform to facilitate the management of bus services by different user roles, namely administrators, conductors, and drivers, each with distinct access levels and functionalities.

The core objective of the system is to replace manual and error-prone processes with an automated solution that enhances efficiency, reduces operational errors, and improves service delivery. The system is equipped with a secure login mechanism that requires an ID and password, ensuring that only authorized personnel can access their respective dashboards and perform their duties.

For administrators, the system provides a dashboard with capabilities to add conductor and driver credentials, as well as input comprehensive trip details including bus numbers, schedules, and personnel assignments. Conductors have access to a dashboard that allows them to manage passenger details, ticketing, and trip revenue. Drivers can utilize their dashboard to report trip specifics such as fuel consumption and actual travel times.

A key feature of the system is the ability for users to check bus timetables without needing to log in, thus providing essential information to passengers planning their journeys. The system is designed with a focus on user-friendliness, security, and data integrity, ensuring that all stakeholders have a reliable and effective tool for managing bus transportation services.

In summary, the Bus Management System DBMS project aims to deliver a dynamic and automated solution for bus management, enhancing the overall experience for administrators, conductors, drivers, and passengers alike.

Table of Contents

	Page No.
Acknowledgements	iii
Abstract	iv
Table of Contents	v
List of Figures	vi
Chapter 1 Introduction	1-5
1.1 Bus Management System Introduction	1
1.2 Literature Overview	2-3
1.3 Technology Overview	3-4
1.4 Problem Formulation	4-5
Chapter 2 Methodology	6-16
2.1 Database Creation	6
2.2 Table Creation	6 -14
2.3 Schema Diagram	15
2.4 System Requirements	16
Chapter 3 Snapshots	17-25
3.1 Admin Dashboard	17-20
3.2 Conductor Dashboard	21-23
3.3 Driver Dashboard	23-25
Chapter 4 Conclusion	26
4.1 General	26
4.2 Future scope	26
Reference	26

List Of Figures

Figure No.	Description	Page No.
Figure 2.2	Tables Created in the Database	14
Figure 2.3	Schema Diagram of The Database	15
Figure 3.1	Design of Admin Login Page	17
Figure 3.1.1	Design of Admin Dashboard	17
Figure 3.1.2	Add Conductor	18
Figure 3.1.3	Add Driver	18
Figure 3.1.4	Add Trip Details	19
Figure 3.1.5	View Timetable	19
Figure 3.1.6	View Milage Details	20
Figure 3.1.7	View Revenue Details	20
Figure 3.2	Design of Conductor Login Page	21
Figure 3.2.1	Design of Conductor Dashboard	21
Figure 3.2.2	Add Passenger Details	22
Figure 3.2.3	Add Trip Details	22
Figure 3.2.4	Assigned Trips of Conductor	23
Figure 3.3	Design of Driver Login Page	23
Figure 3.3.1	Design of Driver Dashboard Page	24
Figure 3.3.2	Add Trip Details	24
Figure 3.3.3	Assigned Trips of Driver	25

Chapter 1

Introduction

1.1 Introduction To Bus Management System

A Bus Management System (BMS) is a sophisticated software solution designed to automate and streamline the operations and management of bus fleets for public and private transportation services. This system encompasses a wide range of functionalities aimed at improving the efficiency, safety, and reliability of bus operations, thereby enhancing the overall passenger experience and operational productivity. At its core, a BMS serves multiple stakeholders, including fleet managers, drivers, conductors, and passengers, by providing tools for scheduling, ticketing, tracking, and managing buses in real-time. For fleet managers and transportation authorities, it offers comprehensive capabilities for route planning, vehicle maintenance scheduling, and performance monitoring. Drivers and conductors benefit from features such as digital logs, navigation assistance, and communication tools, which help in adhering to schedules and managing passenger interactions more effectively.

1.2 Literature Overview

Bus management systems (BMS) are increasingly crucial for improving public transportation efficiency. Research highlights their potential to address various challenges and enhance the overall experience for both passengers and operators. Here's a breakdown of key themes explored in the literature:

Benefits and Functionalities:

Improved Efficiency: Studies report that BMS can optimize scheduling, dispatching, and fleet management, leading to reduced operational costs and improved resource allocation.

Enhanced Passenger Experience: Real-time tracking, mobile ticketing, and arrival information displays are some functionalities that contribute to increased passenger satisfaction and reduced waiting times.

Data-Driven Decision Making: BMS can generate valuable data on passenger ridership, traffic patterns, and vehicle maintenance. This data empowers authorities to make informed decisions regarding route optimization, resource allocation, and service improvements.

Technological Advancements:

GPS and RFID Integration: These technologies enable real-time bus tracking, allowing passengers to monitor arrival times and improving overall service reliability.

Mobile Applications: Mobile ticketing and journey planning apps offered through BMS provide convenience and flexibility for passengers.

Integration with Intelligent Transportation Systems (ITS): BMS can be integrated with broader ITS infrastructure for better traffic management and route optimization.

Challenges and Considerations:

Cost of Implementation and Maintenance: The initial investment and ongoing maintenance costs of BMS can be a barrier for some transportation authorities, particularly in developing regions

Data Security and Privacy Concerns: Data collected by BMS needs robust security measures to protect passenger privacy.

User Adoption: Encouraging passengers to utilize new technologies like mobile ticketing requires effective communication and user-friendly interfaces.

Overall, the literature suggests that Bus Management Systems offer significant potential for improving public transportation efficiency, passenger experience, and data-driven decision making. However, successful implementation requires careful consideration of costs, data security, and user adoption strategies.

1.3 Technology Overview

For DBMS Mini Project on a BUS MANAGEMENT SYSTEM with the functionalities described, the technology stack involves a combination of front-end and back-end technologies, along with a database management system. Here's a detailed breakdown of the tech stack:

Front-end Technologies:

HTML: Used for structuring the content on the web pages.

CSS: Utilized for styling the HTML content, including layouts, colors, and fonts.

JavaScript: Helps in making the web pages interactive. It can be used for form validations, dynamic updates to the content without reloading the page, and handling user events.

Bootstrap: A front-end framework that simplifies the development of responsive and mobile-first web pages. It provides pre-designed components like forms, buttons, navbars, and modals that can be easily customized and integrated.

Back-end Technology:

PHP: A server-side scripting language used to develop the web application's back-end logic. It handles user authentication, database operations, and server-side validations. PHP scripts are executed on the server, and they generate HTML content that is sent to the client's browser.

Database Management System:

MySQL: An open-source relational database management system used to store and manage the application's data. It stores information about users (admin, conductor, driver), trip details, passenger details, and other relevant data. MySQL supports complex queries, transactions, and concurrency, making it suitable for handling the data management needs of the bus management system.

Development Environment:

XAMPP: A free and open-source cross-platform web server solution stack package developed by Apache Friends. It consists of the Apache HTTP Server, MariaDB database, and interpreters for scripts written in PHP and Perl. XAMPP is used to set up a local development environment that allows you to run the web application on your computer without needing to deploy it to an external web server.

Summary of the Tech Stack:

Front-end: HTML, CSS, JavaScript, Bootstrap

Back-end: PHP

Database: MySQL

Development Environment: XAMPP

These tech stack provides a comprehensive set of tools for developing a dynamic and interactive web application for the bus management system. It supports user authentication, data storage

and retrieval, and the implementation of the specific functionalities required by the admin, conductor, and driver dashboards.

1.4 Problem Formulation

1.4.1 Problem Statement

To develop a reliable and efficient bus management system that addresses the needs of different user types (admin, conductor, driver) and provides functionalities such as user authentication, timetable access, and management of trip details.

1.4.2 Some of the Problems

1. How to develop a secure login module that authenticates different types of users (admin, conductor, driver) and provides them with appropriate access to system functionalities?
2. How to design an admin dashboard that facilitates the easy addition and management of conductors and drivers, as well as the ability to input and update trip details?
3. How to develop a system for conductors and drivers to efficiently manage passenger details, ticketing, and trip information.
4. How to design the Bus Management System to handle high volumes of data and user requests without performance degradation?

1.4.3 Significance of Project

The significance of the DBMS mini project on a bus management system (BMS) lies in its ability to streamline and enhance the operations of bus services by leveraging technology to solve several critical issues faced by the transportation industry. Here are some key points that highlight the significance of this project:

Improved Efficiency and Automation: By automating the ticket booking process and other administrative tasks, the BMS reduces the need for manual intervention, thereby increasing efficiency and reducing the potential for human error

Enhanced User Experience: The system provides a user-friendly interface for admins, conductors, and drivers, improving their ability to manage their responsibilities effectively. For passengers, the ability to check timetables without logging in adds convenience and accessibility

Secure and Reliable User Authentication: With a robust login system, the BMS ensures that

sensitive operations and data are accessible only to authorized personnel, enhancing the security of the system

Real-time Data Access and Management: The BMS allows for real-time updates and management of trip details, passenger information, and ticketing, which is crucial for accurate scheduling and resource allocation

1.4.4 Present Scenario

The current scenario of traditional bus management systems involves several key aspects: ticketing, scheduling, fleet management, etc.

In ticketing, traditional systems often rely on manual processes for selling and managing tickets, which can be time-consuming and prone to errors. Scheduling typically involves creating and managing routes and timetables, which can be complex and inflexible in traditional setups. Fleet management in a traditional context often includes manual record-keeping for vehicle maintenance and driver assignments, which can lead to inefficiencies.

Chapter 2

Methodology

2.1 Database Creation

A Database named dbms_bus_mgm must be created in the xampp server.

The query for this is:

```
CREATE DATABASE db_bus_mgm;
```

2.2 Tables & Attributes

Bus_Details Table

The `Bus_Details` table is a central component of the bus management system, storing information about each bus trip. It includes the following attributes:

- `bus_no`: The unique identifier for the bus.
- `destination`: The final stop or destination of the bus trip.
- `source`: The starting point or origin of the bus trip.
- `tripdate`: The date on which the bus trip is scheduled.
- `trip_number`: The primary key for the table, uniquely identifying each trip.

Login Tables

There are three login tables designed to authenticate different types of users in the system: `Login_Admin`, `Login_Conductor`, and `Login_Driver`. Each table contains the following attributes:

- `password`: The password used for authentication.
- `username`: The username used for login.

These tables ensure that each user type has separate credentials for accessing their respective dashboards and functionalities within the bus management

system.

Passenger Table

The `Passenger` table records details about passengers and their tickets. It includes the following attributes:

- `passenger_destination`: The destination stop for the passenger.
- `passenger_source`: The source stop from where the passenger boards the bus.
- `phone_number`: The contact number of the passenger.
- `ticket_id`: A unique identifier for the passenger's ticket.
- `ticket_price`: The price of the ticket.
- `trip_number`: A foreign key that references the `Bus_Details` table to associate the ticket with a specific bus trip.

Trip_Incharge Table

The `Trip_Incharge` table contains information about the a particular trip. It includes the following attributes:

- `trip_no`: A foreign key that references the `Bus_Details` table, linking to a specific trip.
- `driver_id`: The identifier for the driver in charge of the trip.
- `conductor_id`: The identifier for the conductor in charge of the trip.
- `scheduled_arrival_time`: The planned arrival time at the destination.
- `scheduled_departure_time`: The planned departure time from the source.

Trip_Real_Details Table

The `Trip_Real_Details` table captures the actual details of a bus trip as it occurs. It includes the following attributes:

- ``trip_no``: A foreign key that references the ``Bus_Details`` table, linking to a specific trip.
- ``fuel``: The amount of fuel consumed during the trip.
- ``actual_arrival_time``: The actual time the bus arrived at the destination.
- ``actual_departure_time``: The actual time the bus departed from the source.
- ``km_count``: The total kilometers traveled during the trip.

Trip_Result Table

The ``Trip_Result`` table summarizes the financial results of each bus trip. It includes the following attributes:

- ``total_revenue``: The total revenue generated from ticket sales for the trip.
- ``ticket_sold``: The total number of tickets sold for the trip.
- ``trip_no``: A foreign key that references the ``Bus_Details`` table, linking to the financial results of a specific trip.

Views

The bus management system includes two views that provide aggregated data for analysis and reporting:

- ``Revenueperbus`` view: Displays the ``bus_no``, total ``revenue``, and ``ticket_sold`` for each bus, allowing the admin to view the revenue generated per bus.
- ``Millage`` view: Shows the ``bus_no`` and ``milage``, which is calculated from the fuel consumption and km count, to assess the efficiency of each bus.

Code for Create and Insert into Tables

```
SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
```

```
START TRANSACTION;
```

```
CREATE DATABASE IF NOT EXISTS `db_bus_mgm` DEFAULT
CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci;
```

```
USE `db_bus_mgm`;
```

```
DROP TABLE IF EXISTS `bus_details`;
```

```
CREATE TABLE IF NOT EXISTS `bus_details` (  
  `bus_no` int(11) NOT NULL,  
  `trip_no` int(11) NOT NULL,  
  `Source` varchar(255) NOT NULL,  
  `Destination` varchar(255) NOT NULL,  
  `TripDate` date DEFAULT NULL,  
  PRIMARY KEY (`trip_no`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4  
COLLATE=utf8mb4_general_ci;
```

```
INSERT INTO `bus_details` (`bus_no`, `trip_no`, `Source`, `Destination`,  
`TripDate`) VALUES  
  
(100, 1, 'Udupi', 'Mangalore', '2024-04-20'),  
  
(101, 2, 'Mangalore', 'Kundapura', '2024-04-20'),  
  
(102, 3, 'Kundapura', 'Brahmavara', '2024-04-20');
```

```
DROP TABLE IF EXISTS `login`;
```

```
CREATE TABLE IF NOT EXISTS `login` (  
  `user_name` varchar(255) NOT NULL,  
  `password` varchar(255) NOT NULL,  
  PRIMARY KEY (`user_name`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4  
COLLATE=utf8mb4_general_ci;
```

```
INSERT INTO `login` (`user_name`, `password`) VALUES
```

```
('C001', '123'),
```

```
('C002', '123');
```

```
DROP TABLE IF EXISTS `login_admin`;
```

```
CREATE TABLE IF NOT EXISTS `login_admin` (
```

```
  `user_name` varchar(255) NOT NULL,
```

```
  `password` varchar(255) NOT NULL,
```

```
  PRIMARY KEY (`user_name`)
```

```
)          ENGINE=InnoDB          DEFAULT          CHARSET=utf8mb4  
COLLATE=utf8mb4_general_ci;
```

```
INSERT INTO `login_admin` (`user_name`, `password`) VALUES
```

```
('admin', '123');
```

```
DROP TABLE IF EXISTS `login_driver`;
```

```
CREATE TABLE IF NOT EXISTS `login_driver` (
```

```
  `user_name` varchar(255) NOT NULL,
```

```
  `password` varchar(255) NOT NULL,
```

```
  PRIMARY KEY (`user_name`)
```

```
)          ENGINE=InnoDB          DEFAULT          CHARSET=utf8mb4  
COLLATE=utf8mb4_general_ci;
```

```
INSERT INTO `login_driver` (`user_name`, `password`) VALUES
```

```
('D001', '123'),
```



```
('D002', '123');
```

```
DROP TABLE IF EXISTS `passenger`;
```

```
CREATE TABLE IF NOT EXISTS `passenger` (  
  `phone_no` varchar(255) NOT NULL,  
  `ticket_id` int(11) NOT NULL AUTO_INCREMENT,  
  `ticket_price` int(11) NOT NULL,  
  `Passenger_source` varchar(255) NOT NULL,  
  `trip_no_passenger` int(11) NOT NULL,  
  `Passenger_destination` varchar(255) NOT NULL,  
  PRIMARY KEY (`ticket_id`),  
  KEY `trip_no_passenger` (`trip_no_passenger`)  
) ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT CHARSET=utf8mb4  
COLLATE=utf8mb4_general_ci;
```

```
INSERT INTO `passenger` (`phone_no`, `ticket_id`, `ticket_price`,  
  `Passenger_source`, `trip_no_passenger`, `Passenger_destination`) VALUES  
  
('9999999999', 1, 10, 'Udupi', 1, 'Mangalore'),  
  
('88888888888', 2, 15, 'Mangalore', 2, 'Kundapura'),  
  
('77777777777', 3, 5, 'Kundapura', 3, 'Brahmavara');
```

```
DROP TABLE IF EXISTS `trip_incharge`;
```

```
CREATE TABLE IF NOT EXISTS `trip_incharge` (  
  `trip_no_incharge` int(11) NOT NULL,  
  `Driver_emp_id` varchar(255) NOT NULL,
```

```

`Conductor_emp_id` varchar(255) NOT NULL,

`scheduled_dept_time` time NOT NULL,

`scheduled_arr_time` time NOT NULL,

PRIMARY KEY (`trip_no_incharge`)

)          ENGINE=InnoDB          DEFAULT          CHARSET=utf8mb4
COLLATE=utf8mb4_general_ci;

```

```

INSERT INTO `trip_incharge` (`trip_no_incharge`, `Driver_emp_id`,
`Conductor_emp_id`, `scheduled_dept_time`, `scheduled_arr_time`) VALUES

(1, 'D001', 'C001', '09:00:00', '10:00:00'),

(2, 'D002', 'C002', '10:00:00', '11:00:00'),

(3, 'D001', 'C001', '12:00:00', '13:00:00');

```

```

DROP TABLE IF EXISTS `trip_real_details`;

CREATE TABLE IF NOT EXISTS `trip_real_details` (

`trip_no_real` int(11) NOT NULL,

`fuel` int(11) NOT NULL,

`arrival_time` time NOT NULL,

`departure_time` time NOT NULL,

`km_count` int(11) NOT NULL,

PRIMARY KEY (`trip_no_real`)

)          ENGINE=InnoDB          DEFAULT          CHARSET=utf8mb4
COLLATE=utf8mb4_general_ci;

```

```

INSERT INTO `trip_real_details` (`trip_no_real`, `fuel`, `arrival_time`,
`departure_time`, `km_count`) VALUES

```

(1, 50, '09:30:00', '10:30:00', 40),

(2, 65, '10:30:00', '11:30:00', 60),

(3, 55, '12:30:00', '13:30:00', 40);

DROP TABLE IF EXISTS `trip_result`;

CREATE TABLE IF NOT EXISTS `trip_result` (

 `trip_no_result` int(11) NOT NULL,

 `bus_no_result` int(11) NOT NULL,

 `revenue` int(11) NOT NULL,

 `tickets_sold` int(11) NOT NULL,

 PRIMARY KEY (`trip_no_result`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_general_ci;

INSERT INTO `trip_result` (`trip_no_result`, `bus_no_result`, `revenue`,
`tickets_sold`) VALUES

(1, 0, 100, 10),

(2, 0, 150, 10),

(3, 0, 50, 10);

CREATE VIEW `mileage` AS SELECT `bus_details`.`bus_no` AS `bus_no`,
avg(`trip_real_details`.`km_count` / `trip_real_details`.`fuel`) AS `MILAGE`
FROM (`bus_details` join `trip_real_details` on(`trip_real_details`.`trip_no_real` =
`bus_details`.`trip_no`)) GROUP BY `bus_details`.`bus_no` ;

CREATE VIEW `revenueperbus` AS SELECT `bus_details`.`bus_no` AS
`bus_no`, sum(`trip_result`.`revenue`) AS `revenue`,
sum(`trip_result`.`tickets_sold`) AS `tickets_sold` FROM (`bus_details` join
`trip_result` on(`trip_result`.`trip_no_result` = `bus_details`.`trip_no`)) GROUP
BY `bus_details`.`bus_no` ;

```
ALTER TABLE `passenger` ADD CONSTRAINT `passenger_ibfk_1` FOREIGN
KEY (`trip_no_passenger`) REFERENCES `bus_details` (`trip_no`) ON DELETE
CASCADE ON UPDATE CASCADE;
```

```
ALTER TABLE `trip_incharge` ADD CONSTRAINT `trip_incharge_ibfk_1`
FOREIGN KEY (`trip_no_incharge`) REFERENCES `bus_details` (`trip_no`) ON
DELETE CASCADE ON UPDATE CASCADE;
```

```
ALTER TABLE `trip_real_details` ADD CONSTRAINT
`trip_real_details_ibfk_1` FOREIGN KEY (`trip_no_real`) REFERENCES
`bus_details` (`trip_no`) ON DELETE CASCADE ON UPDATE CASCADE;
```

```
ALTER TABLE `trip_result` ADD CONSTRAINT `trip_result_ibfk_1` FOREIGN
KEY (`trip_no_result`) REFERENCES `bus_details` (`trip_no`) ON DELETE
CASCADE ON UPDATE CASCADE;
```

```
COMMIT;
```

Table	Action	Rows	Type	Collation	Size	Overhead
<input type="checkbox"/> bus_details	★ Browse Structure Search Insert Empty Drop	3	InnoDB	utf8mb4_general_ci	16.0 KiB	-
<input type="checkbox"/> login	★ Browse Structure Search Insert Empty Drop	2	InnoDB	utf8mb4_general_ci	16.0 KiB	-
<input type="checkbox"/> login_admin	★ Browse Structure Search Insert Empty Drop	1	InnoDB	utf8mb4_general_ci	16.0 KiB	-
<input type="checkbox"/> login_driver	★ Browse Structure Search Insert Empty Drop	2	InnoDB	utf8mb4_general_ci	16.0 KiB	-
<input type="checkbox"/> mileage	★ Browse Structure Search Insert Edit Drop	~0	View	---	-	-
<input type="checkbox"/> passenger	★ Browse Structure Search Insert Empty Drop	3	InnoDB	utf8mb4_general_ci	32.0 KiB	-
<input type="checkbox"/> revenueperbus	★ Browse Structure Search Insert Edit Drop	~0	View	---	-	-
<input type="checkbox"/> trip_incharge	★ Browse Structure Search Insert Empty Drop	3	InnoDB	utf8mb4_general_ci	16.0 KiB	-
<input type="checkbox"/> trip_real_details	★ Browse Structure Search Insert Empty Drop	3	InnoDB	utf8mb4_general_ci	16.0 KiB	-
<input type="checkbox"/> trip_result	★ Browse Structure Search Insert Empty Drop	3	InnoDB	utf8mb4_general_ci	16.0 KiB	-
10 tables	Sum	~20	InnoDB	utf8mb4_general_ci	144.0 KiB	0 B

Figure: 2.1 Tables & Views Created in the Database “dbms_mgm_sys”

2.3 SCHEMA DIAGRAM

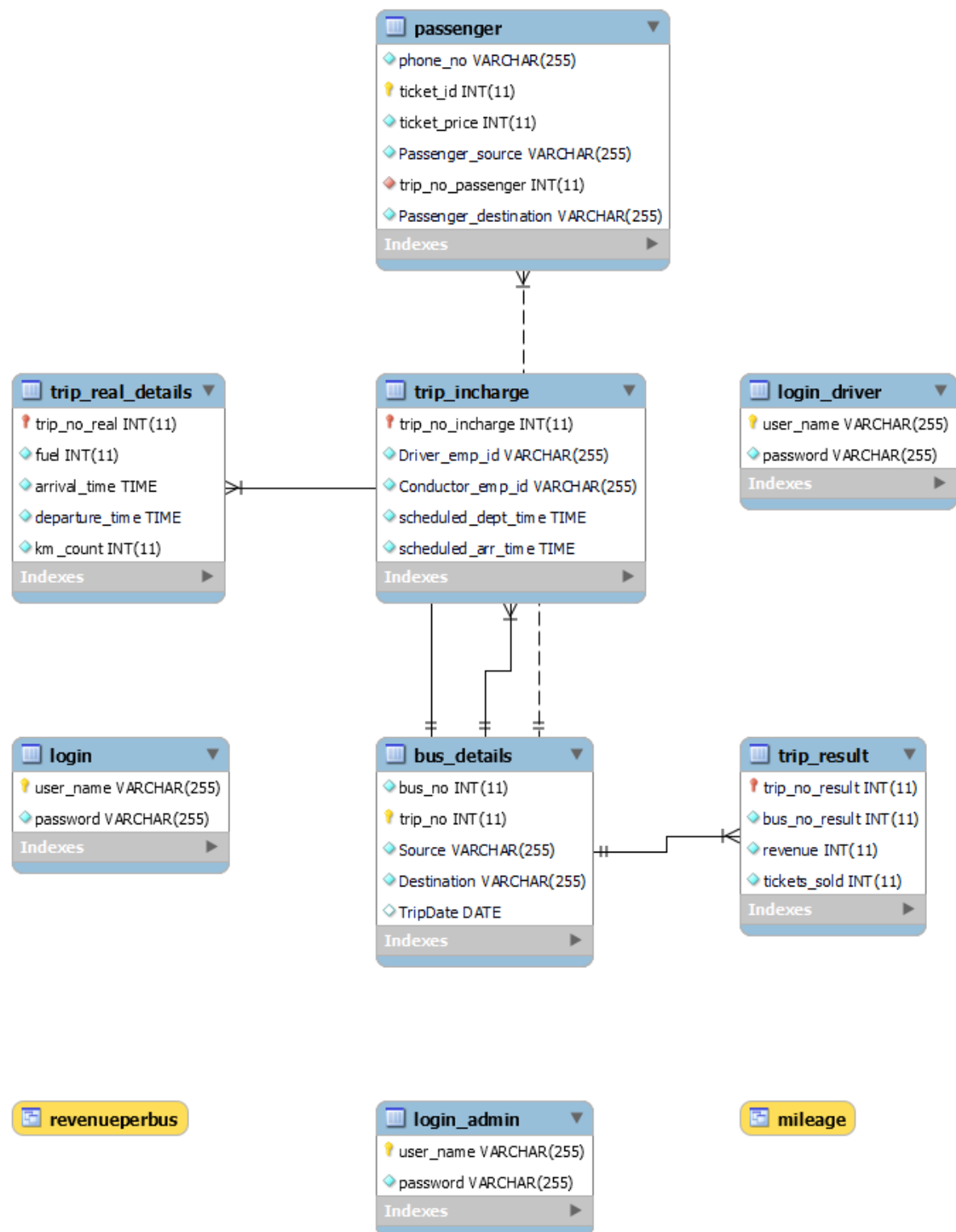


Figure 2.4: Schema Diagram

2.4 System Requirement

System Requirements:

1. Operating System: Windows, Linux, or macOS.
2. Web Server: Apache, Nginx, or any other compatible web server.
3. PHP Version: PHP 7.x or later.
4. MySQL Version: MySQL 5.7 or later.
5. Browser Compatibility: Support for modern web browsers (Chrome, Firefox, Safari, Edge).

Software Requirements:

1. Text Editor or IDE: Visual Studio Code, Sublime Text, PHPStorm, etc.
2. MySQL Database Management Tool: phpMyAdmin, MySQL Workbench, or any other MySQL management tool.
3. Web Browser: Chrome, Firefox, Safari, Edge, etc.

Specifications:

1. User Authentication: Implement user authentication system (login, registration, logout).
2. Database Schema: Design the database schema including tables, relationships, and constraints.
3. User Interface (UI): Design intuitive user interfaces for data entry, viewing, and manipulation.
4. Functionality:
 - CRUD Operations: Allow users to Create, Read, Update, and Delete records.
 - Validation: Validate user inputs to ensure data integrity.
5. Security:
 - Session Management: Manage user sessions securely to prevent unauthorized access.
 - Password Encryption: Store user passwords securely using hashing algorithms.
6. Error Handling: Implement robust error handling mechanisms to handle unexpected scenarios gracefully.
7. Documentation:
 - Provide comprehensive documentation including installation instructions, usage guidelines, and database schema explanation.
 - Document code for better understanding and future maintenance.

Chapter 3

Snapshots

3.1 Admin

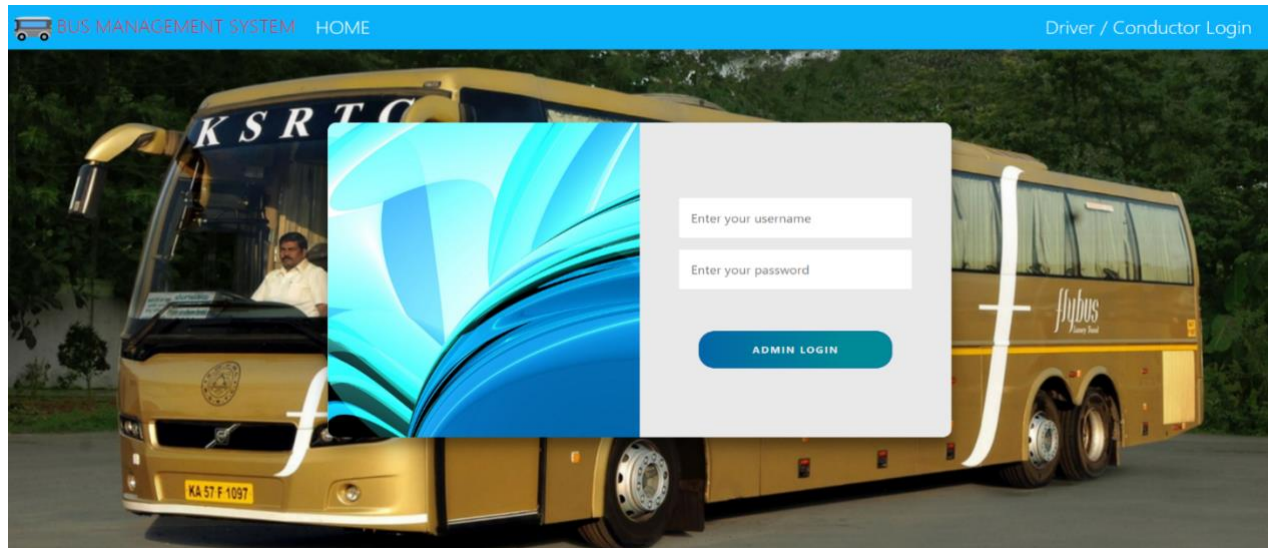


Figure 3.1: Admin Login Page

3.1.1 Admin Dashboard

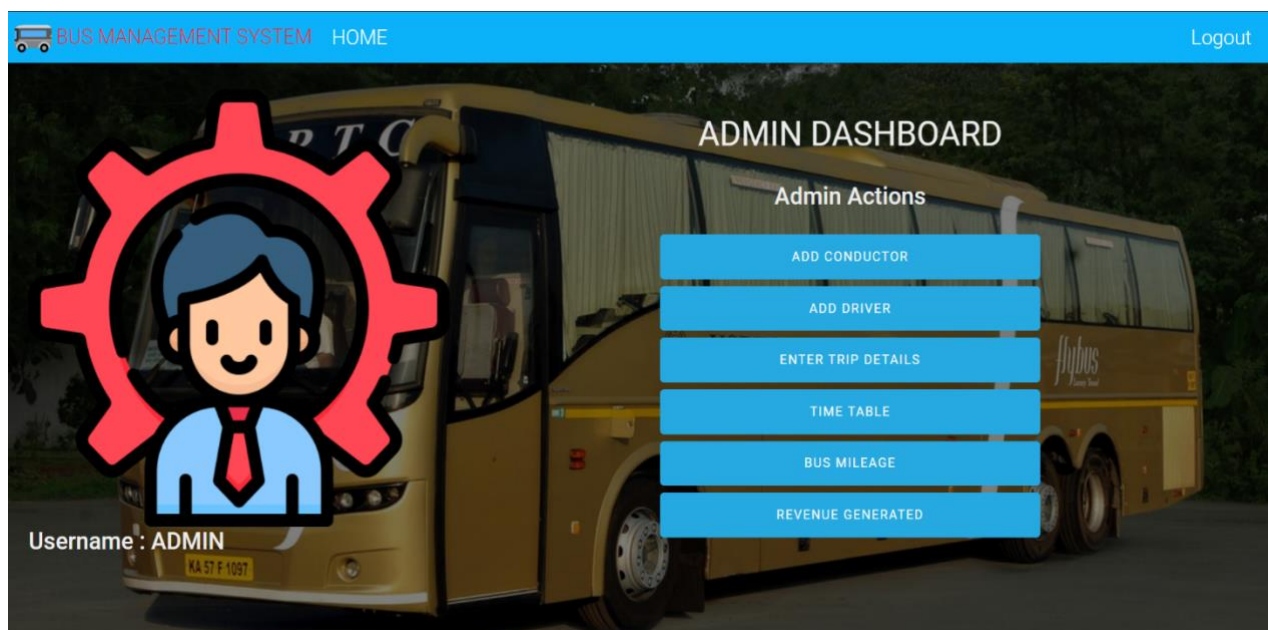


Figure 3.1.1: Admin Dashboard Page

3.1.2 Admin Dashboard Functionalities

The screenshot shows the 'Add Conductor Credentials' form overlaid on a background image of a yellow KSRTC bus. The form is a white rounded rectangle with a blue 'Submit' button. It contains two input fields for 'Conductor ID' and 'Password', each with a label and a small icon. The background image shows a yellow KSRTC bus with a driver visible through the windshield. The bus has 'KSRTC' on the front and 'Hybus' on the side. The license plate is KA 57 F 1097. The top navigation bar is blue with a bus icon, 'BUS MANAGEMENT SYSTEM', 'HOME', 'Go Back', and 'Logout'.

Add Conductor Credentials

Conductor ID :

Conductor ID

Password :

Password

Submit

Figure 3.1.2: Add Conductor

The screenshot shows the 'Add Driver Credentials' form overlaid on a background image of a yellow KSRTC bus. The form is a white rounded rectangle with a blue 'Submit' button. It contains two input fields for 'Driver ID' and 'Password', each with a label and a small icon. The background image shows a yellow KSRTC bus with a driver visible through the windshield. The bus has 'KSRTC' on the front and 'Hybus' on the side. The license plate is KA 57 F 1097. The top navigation bar is blue with a bus icon, 'BUS MANAGEMENT SYSTEM', 'HOME', 'Go Back', and 'Logout'.

Add Driver Credentials

Driver ID :


Driver ID

Password :

Password

Submit

Figure 3.1.3: Add Driver


BUS MANAGEMENT SYSTEM
[HOME](#)
[Go Back](#)
[Logout](#)

Enter Trip Details

Bus Number:

Date:

Source:

Destination:

Scheduled Departure:


Scheduled Arrival:

Driver ID:

Conductor ID:


Submit

Figure 3.1.4: Add Trip Details


BUS MANAGEMENT SYSTEM
[HOME](#)
[Go Back](#)
[Logout](#)

#	Trip Number	Bus Number	Source	Destination	Trip Date	Driver ID	Conductor ID	Departure Time	Arrival Time
1	1	100	Udupi	Mangalore	2024-04-20	D001	C001	09:00:00	10:00:00
2	2	101	Mangalore	Kundapura	2024-04-20	D002	C002	10:00:00	11:00:00
3	3	102	Kundapura	Brahmavara	2024-04-20	D001	C001	12:00:00	13:00:00

Figure 3.1.5: View Timetable


BUS MANAGEMENT SYSTEM
HOME
Go Back
Logout


Search by Bus Number:

Bus Number

SUBMIT

S.N	Bus Number	MILAGE
1	100	5.00000000
2	101	5.29410000
3	102	4.44440000

Figure 3.1.6: View Milage Per BUS


BUS MANAGEMENT SYSTEM
HOME
Go Back
Logout

Search by Bus Number:

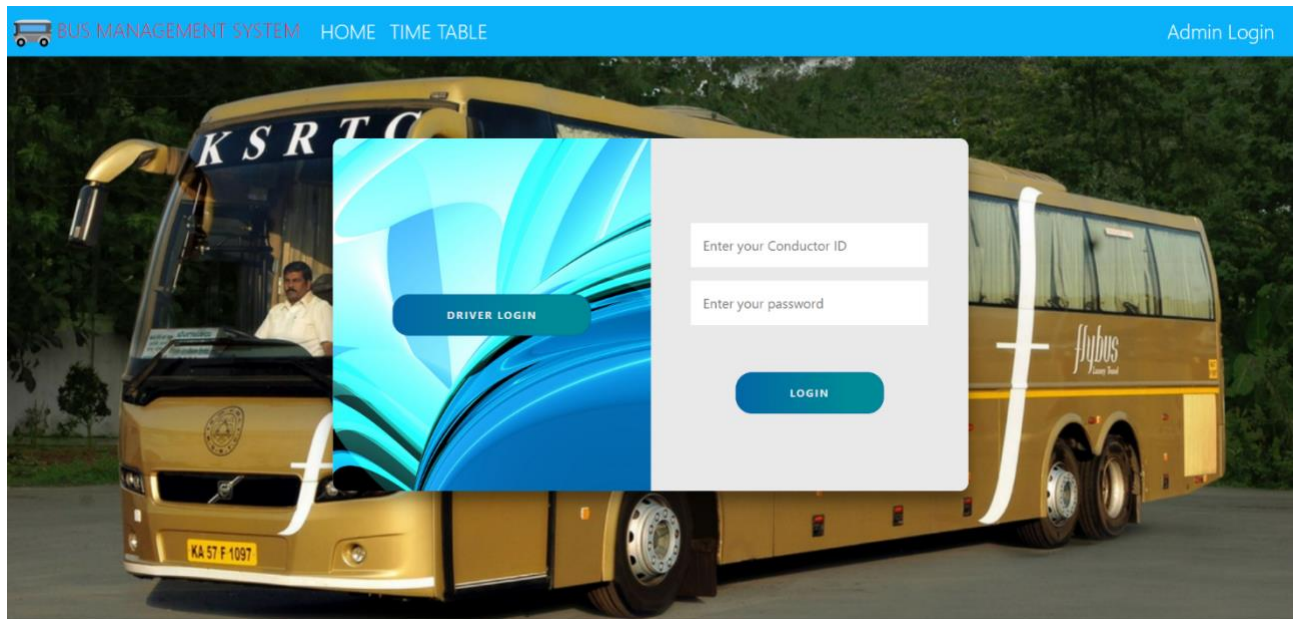
Bus Number

SUBMIT

#	Bus Number	Revenue Generated	Tickets Sold
1	100	100	10
2	101	150	10
3	102	50	10

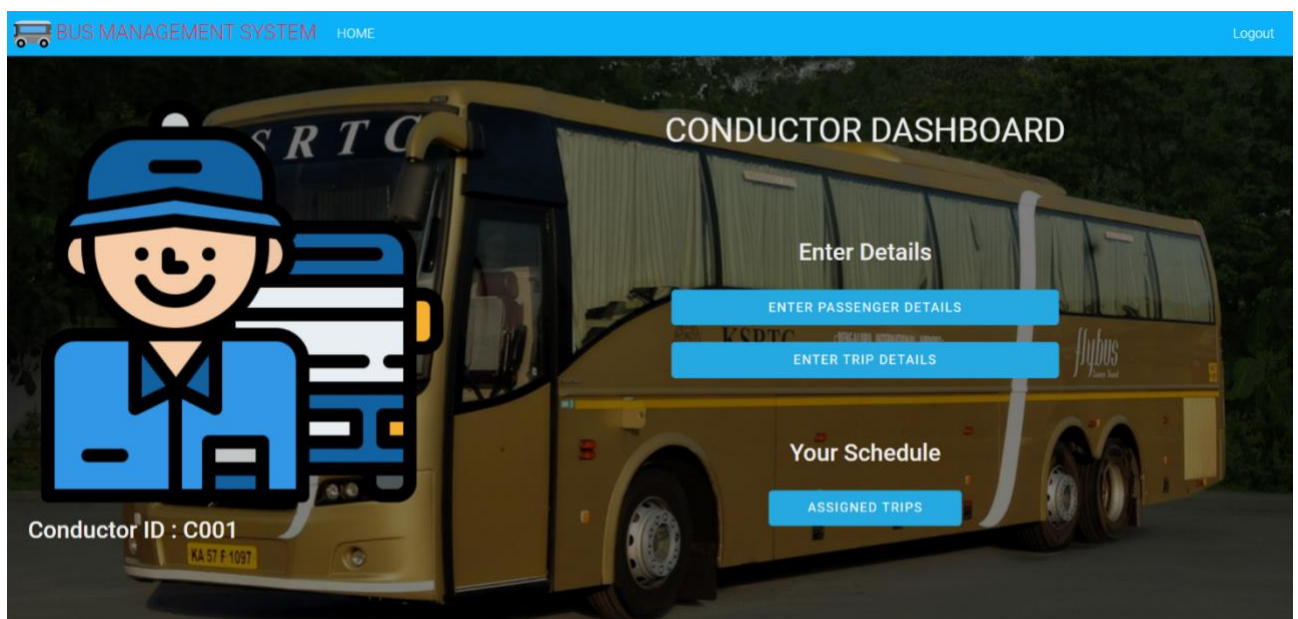
3.1.7: View Revenue Per BUS

3.2 Conductor



3.2: Conductor Login Page

3.2.1 Conductor Dashboard



3.2.1: Design of Conductor Dashboard


3.2.2 Conductor Dashboard Functionalities


The screenshot shows a web application interface for a bus management system. At the top, there is a blue header bar with a bus icon, the text "BUS MANAGEMENT SYSTEM", a "HOME" link, and "Go Back" and "Logout" links. The background of the page features a large image of a yellow KSRTC bus. Overlaid on this is a white modal form titled "Enter Passenger Details". The form contains the following fields: "Trip Number:" with a dropdown menu showing "1"; "Phone Number" with a text input field; "Select Source Bus stop:" with a text input field; "Select Destination Bus stop:" with a text input field; and "Ticket Price:" with a text input field. A blue "Submit" button is located at the bottom of the form.

Figure 3.2.2: Add Passenger

The screenshot shows the same web application interface as Figure 3.2.2, but with a different modal form titled "Enter Trip Details" overlaid. The form contains the following fields: "Trip Number:" with a dropdown menu showing "1"; "Total Revenue Generated:" with a text input field showing "0"; and "Total Tickets Sold:" with a text input field showing "Total Tickets Sold". A blue "Submit" button is located at the bottom of the form.

3.2.3: Add Trip Details



BUS MANAGEMENT SYSTEM
HOME
Go Back
Logout

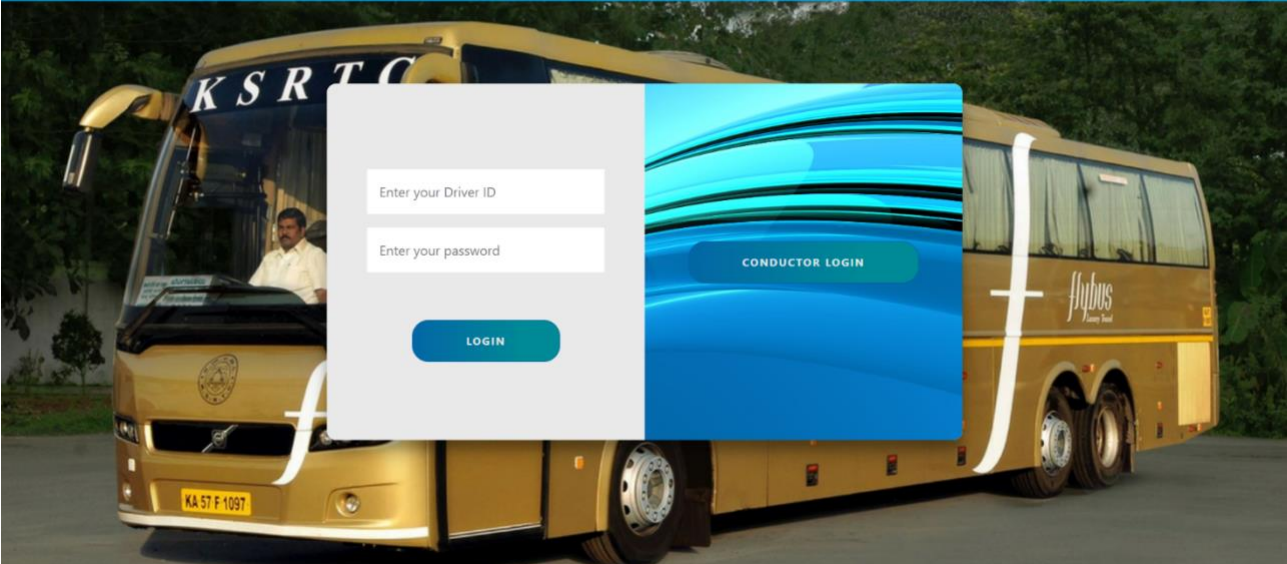



#	Trip Number	Bus Number	Source	Destination	Trip Date	Driver ID	Departure Time	Arrival Time
1	1	100	Udupi	Mangalore	2024-04-20	D001	09:00:00	10:00:00
2	3	102	Kundapura	Brahmavara	2024-04-20	D001	12:00:00	13:00:00

3.2.4: View Assigned Trips

3.3 Driver


BUS MANAGEMENT SYSTEM
HOME
TIME TABLE
Admin Login

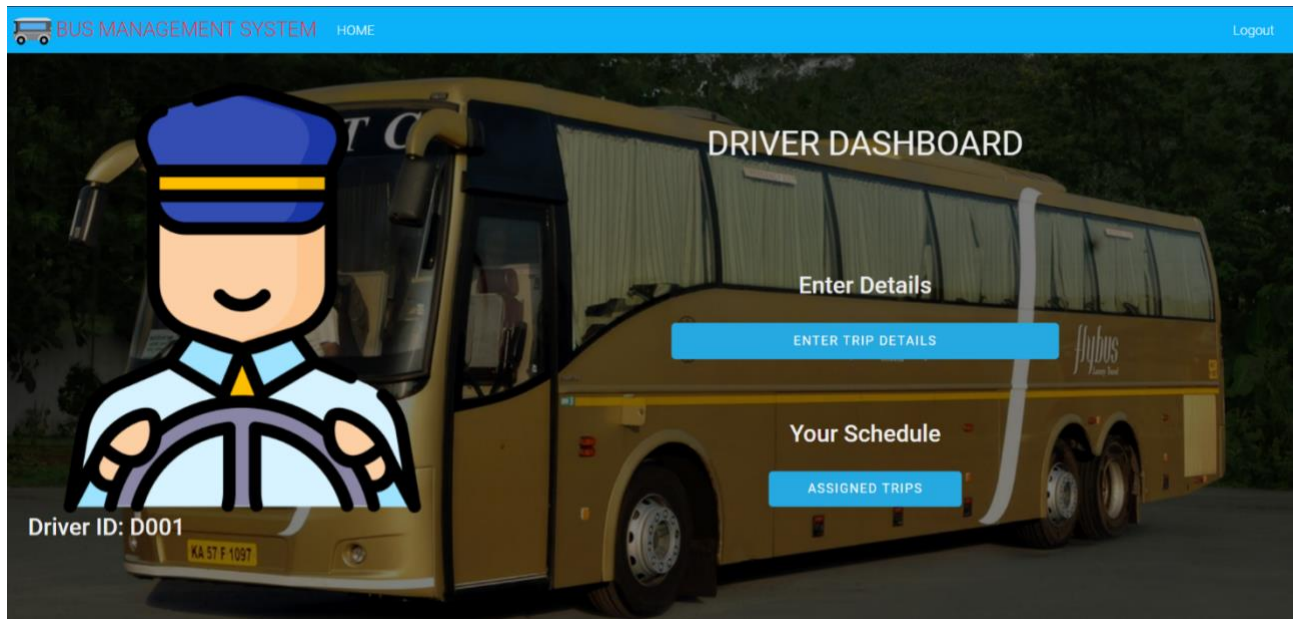




CONDUCTOR LOGIN

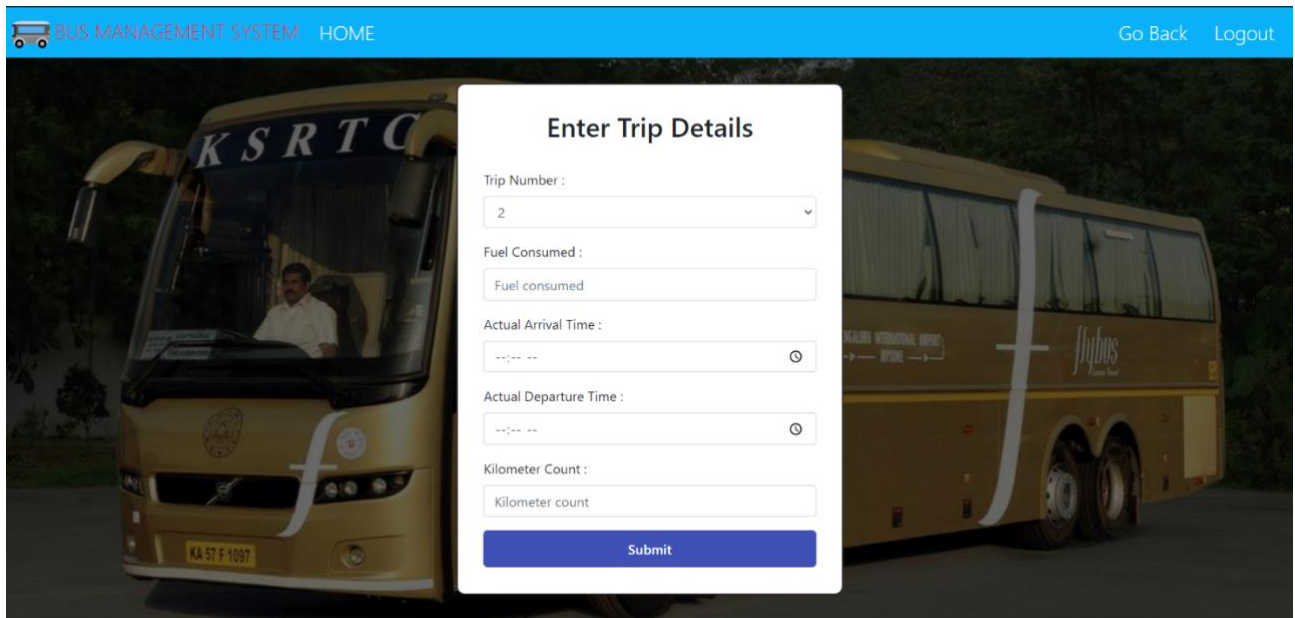
3.3: Driver Login Page

3.3.1 Driver Dashboard




3.3.1: Driver Dashboard Page

3.3.2: Driver Dashboard Functionalities

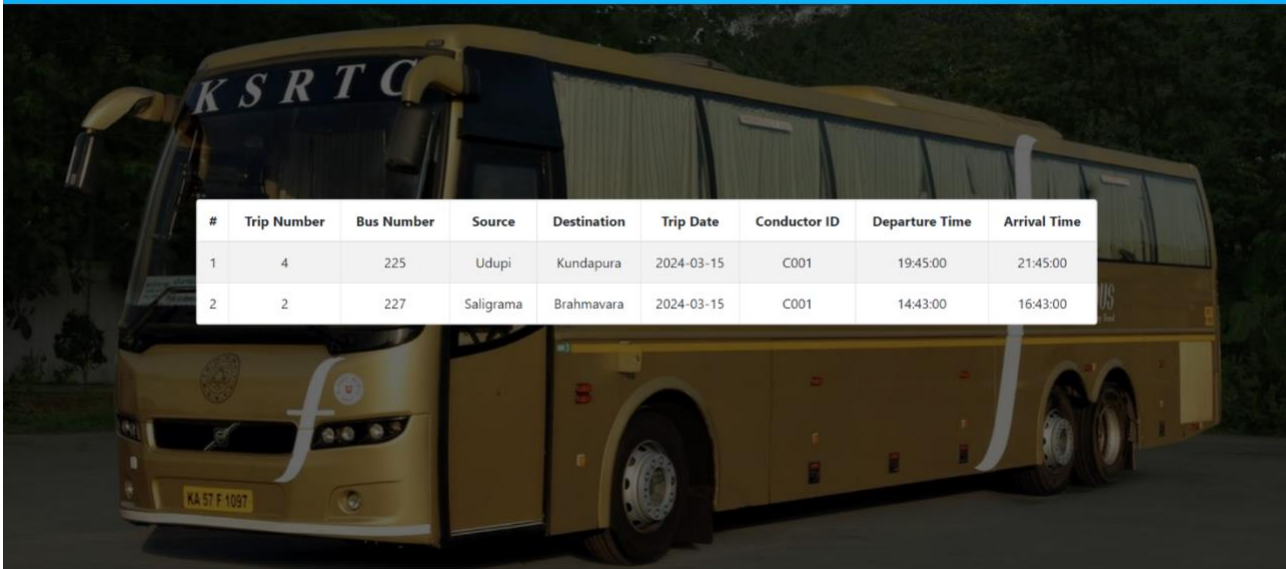


3.3.2: Add Trip Details

 BUS MANAGEMENT SYSTEM

HOME

Go BackLogout



#	Trip Number	Bus Number	Source	Destination	Trip Date	Conductor ID	Departure Time	Arrival Time
1	4	225	Udupi	Kundapura	2024-03-15	C001	19:45:00	21:45:00
2	2	227	Saligrama	Brahmavara	2024-03-15	C001	14:43:00	16:43:00

3.3.3: View Assigned Trip

Chapter 4

Conclusion

4.1 General

The bus management system described provides a comprehensive solution for managing various aspects of bus operations, including trip scheduling, user authentication, and passenger ticketing. The system's database is well-structured, with dedicated tables for bus details, user logins, passenger information, trip assignments, and actual trip data. The inclusion of views for revenue per bus and mileage offers valuable insights for administrative purposes. The system effectively segregates functionalities across different user roles admin, conductor, and driver ensuring that each has access to the necessary tools and information to perform their duties efficiently. The use of foreign keys to link related data ensures data integrity and provides a relational structure that facilitates complex queries and reports. The system's design allows for secure login processes for different user types, and the dashboard functionalities provide a user-friendly interface for managing and reporting on the various aspects of the bus service

4.2 Future Scope

Future enhancements for the bus management system could include the development of a mobile app, GPS tracking for buses, an online payment gateway, data-driven route optimization, predictive maintenance analytics, a customer feedback feature, advanced reporting tools, and scalable system architecture to support growth.

Reference

- [1] PHP Manual: <https://www.php.net/manual/en/ref.mysql.php>
- [2] MySQL 8.0 Reference Manual. (2024): <https://dev.mysql.com/doc/refman/8.0/en/>