VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"Jnana Sangama", Belagavi-590018

Database Management System

Mini Project Report

On

"TOLL MANAGEMENT SYSTEM"

Submitted in partial fulfillment of the requirement of Semester Database Management System Laboratory

Submitted by,

KARTHIK N HEBBAR 1DT20IS045
M RONITH 1DT20IS051

Under the guidance of

Mrs. Spandana S G
Assistant Professor
Dept. of ISE
DSATM,
Bangalore.

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

DAYANANDA SAGAR ACADEMY OF TECHNOLOGY AND MANAGEMENT

(Affiliated to Visvesvaraya Technological University, Belagavi & Approved by AICTE, New Delhi) All Engineering Branches Accredited by NBA, New Delhi
Opp. Art of Living, Udayapura, Kanakapura Road, Bangalore-560082
2022-2023

DAYANANDA SAGAR ACADEMY OF TECHNOLOGY AND MANAGEMENT

(Affiliated to Visvesvaraya Technological University, Belagavi & Approved by AICTE, New Delhi)
All Engineering Branches Accredited by NBA, New Delhi
Opp. Art of Living, Udayapura, Kanakapura Road, Bangalore- 560082

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

Certificate

This is to certify that the mini-project work entitled **"TOLL MANAGEMENT SYSTEM"** is carried out by **KARTHIK N HEBBAR (1DT20IS045)** and **M RONITH (1DT20IS051)** in partial fulfillment for the requirement of V Semester Database Management System Laboratory(18CSL58) in **Information Science and Engineering of the Visvesvaraya Technological University, Belagavi** during the year 2022-2023. It is certified that all the corrections/suggestions indicated for the given internal assessment have been incorporated in the report. This report has been approved as it satisfies the academic requirements with respect to the mini-project work.

Signature of the Guide

Mrs. Spandana S G
Assistant Professor
Dept. of ISE
DSATM, Bangalore.

Signature of the HOD

Dr. Nandini Prasad K S
Dean - Foreign Affairs & HOD
Dept. of ISE
DSATM, Bangalore.

External Viva

Name of the Examiners	Signature with Date
1.	
2.	

ACKNOWLEDGEMENT

We express our profound gratitude to Dr. M Ravishankar, Principal, DSATM, Bangalore, for providing the necessary facilities and an ambient environment to work.

We are grateful to Dr. Nandini, Head of Department, Information Science and Engineering, DSATM, Bangalore, for valuable suggestions and advice throughout our work period.

We would like to express our deepest gratitude and sincere thanks to our guide Mrs. Spandana S G Assistant Professor, Department of Information Science and Engineering, DSATM, Bangalore, for teaching the course, keen interest and encouragement, whose guidance made the work more efficient.

We would like to thank all the staff members of the Department of Information Science and Engineering for their support and encouragement during the course of this project.

Definitely most, we would like to thank our parents, all my family members and friends, without whose help and encouragement this creative assessment would have been impossible.

ABSTRACT

As the name specifies "TOLL MANAGEMENT SYSTEM" is a computerized system that aims to automate and streamline the process of collecting tolls at toll plazas. The system uses a database to store information about vehicles, toll rates, and transactions. The main features of the TollEase project are Reporting and analytics: Provides real-time and historical information about toll transactions, including the number of vehicles that passed through the toll plaza, revenue generated, and peak traffic hours.

The TollEase system can be implemented in web-based applications, and can be integrated with existing toll plaza infrastructure. It uses MySQL and phpMyAdmin to store and manage data, which can be queried and analyzed to gain insights and make informed decisions about toll plaza operations. This system can serve as a useful tool for toll plaza operators, allowing them to improve efficiency, increase revenue, and provide a better experience for toll users.

We made use of PHP for the backend purpose and have built the frontend using HTML, CSS and JAVASCRIPT. Through our website toll plaza can do billing and all the transaction details are stored in MySQL database and can be seen in phpMyAdmin Additionally, commuters can see the traffic on the highways, so that they can plan their trip with help to traffic on the last day.

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
i	ACKNOWLEDGEMENT	i
ii	ABSTRACT	ii
iii	TABLE OF CONTENT	iii
iv	LIST OF FIGURES	iv
1	INTRODUCTION	1
2	REQUIRED ANALYSIS	2
3	DESIGN	3 - 4
4	IMPLEMENTATION	5 - 7
5	SNAPSHOTS	8 – 13
6	CONCLUSION AND FUTURE ENHANCEMENT	14
7	REFERENCE	15

LIST OF FIGURES

FIGURE NO.	FIGURE DESCRIPTION	PAGE NO.
1	Entity Relationship Diagram	3
2	Schema Diagram	4
3	connect.php Code	8
4	php code to fetch the vehicle_type	8
5	php code to fetch the entry_point and exit_point	9
6	Google charts code	10
7	Billing Table	11
8	Billing Page	11
9	Index Page with Hourly Graph	12
10	Index Page with One Day Graph	12
11	Index Page with One Month Graph	13

CHAPTER 1

INTRODUCTION

Our website "TollEase" is a web-based application that allows users to view and analyze historical traffic data on the highways. The system uses data collected from toll plazas to provide historical information about traffic patterns, including the number of vehicles that passed through the toll plaza, revenue generated, and peak traffic hours. The system also provides a graphical representation of the data, such as line charts, to help users better understand the traffic patterns.

Toll plazas are the places where the vehicles are required to pay a fee for using the highways. The process of toll payment is usually time-consuming and can cause inconvenience to the travelers. TollEase has been designed to overcome these difficulties. The project allows the toll collectors to easily manage the toll payments and also provides real-time information about the traffic on the highways.

One of the key features of the TollEase project is its ability to provide information about the traffic on the highways. The system collects data about the number of vehicles that pass through the toll plaza in a given time period. This data is then processed and presented in the form of a line graph. The graph provides the information about the total number of vehicles that passed through the toll plaza in a particular time period. This information can be used to make decisions about the maintenance of the highways and also to manage the traffic.

The project also has a billing module which helps in generating the bills for the toll payment. The system keeps a record of all the transactions that take place at the toll plaza.

The main features of the project include:

- *Data collection and storage:* The system collects data from toll plazas, and stores it in a MySQL database for easy retrieval and analysis.
- *Traffic analysis*: The system provides historical information about traffic patterns, including the number of vehicles that passed through the toll plaza, revenue generated, and peak traffic hours.
- *Graphical representation of data:* The system provides a graphical representation of the data, such as line charts, to help users better understand the traffic patterns.
- *User-friendly interface:* The system has a user-friendly interface that allows users to easily view and analyze the data.

This system can be used by highway authorities and operators, traffic engineers, and researchers to understand traffic patterns and make informed decisions about highway operations and maintenance. Additionally, it can also be used by businesses, such as logistics companies and fleet managers, to plan their routes more efficiently and reduce costs.

CHAPTER 2

REQUIREMENTS ANALYSIS

The requirement analysis specifies the requirements needed to develop a project. In this phase, we collect the requirements needed for designing the project. The requirements collected are then analyzed and carried to the next phase

2.1 FRONTEND

For frontend we have used:

- HTML
- CSS
- JavaScript

2.2 BACKEND

For backend we have used:

- MySQL
- PHP
- phpMyAdmin
- XAMPP

CHAPTER 3

DESIGN

3.1 ER DIAGRAM

An Entity – Relationship model (ER model) describes inter-related things of interest in a specific domain of knowledge. An ER model is composed of entity types (which classify the things of interest) and specifies relationships that can exist between instances of those entity types.

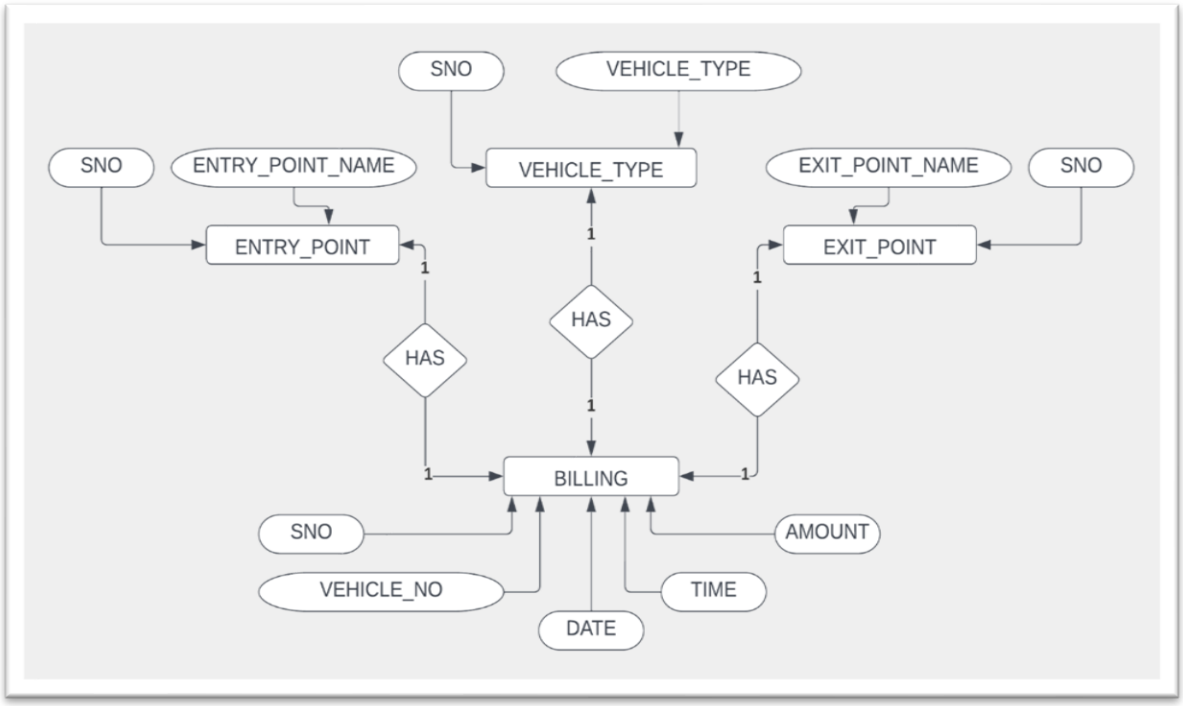

Figure 1. Entity Relationship Diagram

The ER diagram for the "TollEase" toll management system represents the relationships between four entities: billing, vehicle_type, entry_point, and exit_point.

The "billing" entity represents the transactions of vehicles passing through the toll plaza, with attributes such as serial number (sno), type of vehicle (vehicle_type), license plate number (vehicle_no), date, time, entry point (entry_point), exit point (exit_point), and the amount charged for the toll. The "vehicle_type" entity lists the different types of vehicles, with a serial number (sno) and the type name (vehicle_type). The "entry_point" entity contains the names and serial numbers of the entry points at the toll plaza (sno, entry_point_name). The "exit_point" entity holds the names and serial numbers of the exit points at the toll plaza (sno, exit_point_name).

This ER diagram captures the relationships between these entities and the data they hold, which is important for the functioning of the toll management system and accurate billing.

3.2 SCHEMA DIAGRAM

The term "schema" refers to the organization of data as a blueprint of how the database is Constructed (divided into database tables in the case of relational databases).

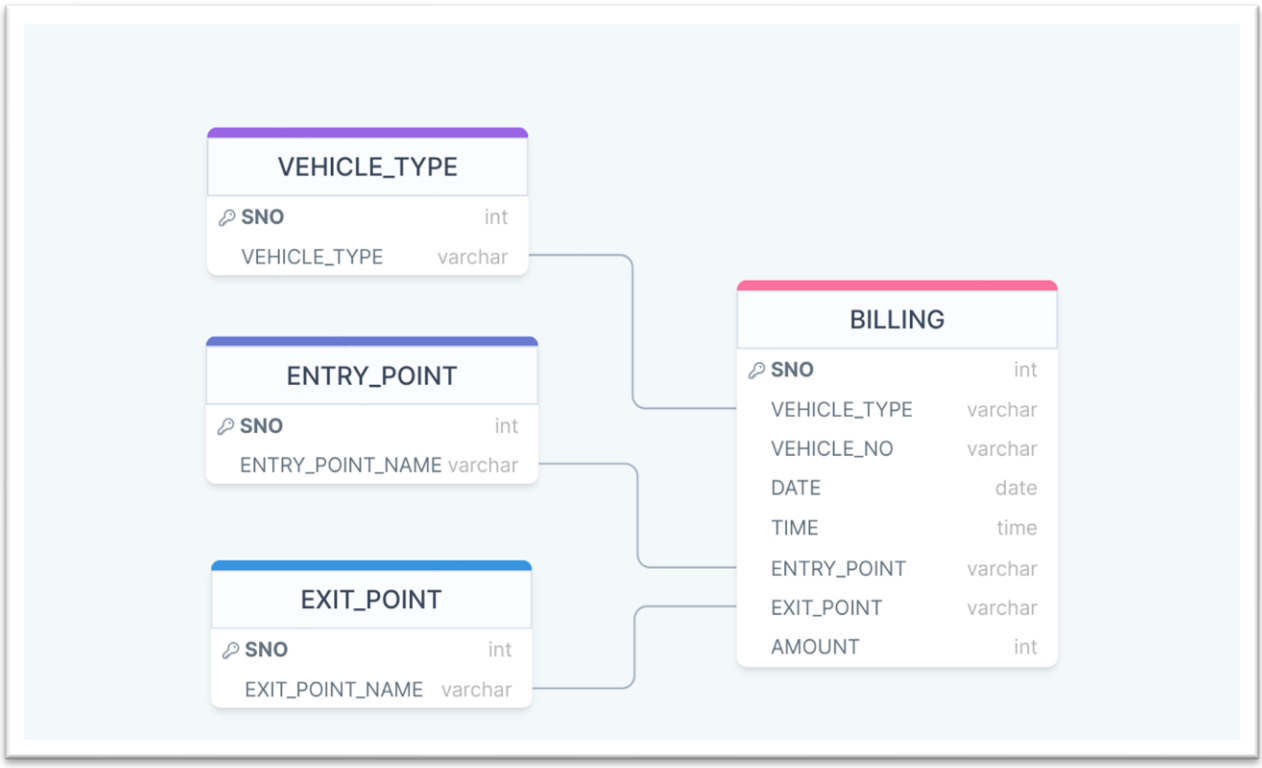

Figure 2. Schema Diagram

The schema diagram for the table in the Toll Management System project, TollEase, consists of four entities: billing, vehicle_type, entry_point, and exit_point.

The billing entity has seven attributes: sno, vehicle_type, vehicle_no, date, time, entry_point, exit_point, and amount. The sno attribute is the serial number of each record, the vehicle_type attribute identifies the type of vehicle, the vehicle_no attribute is the license plate number of the vehicle, the date attribute is the date the vehicle passed through the toll plaza, the time attribute is the time the vehicle passed through the toll plaza, the entry_point attribute is the name of the entry point, the exit_point attribute is the name of the exit point, and the amount attribute is the amount charged for the toll. The vehicle_type entity has two attributes: sno and vehicle_type. The sno attribute is the serial number of each record, and the vehicle_type attribute identifies the type of vehicle. The entry_point entity has two attributes: sno and entry_point_name. The sno attribute is the serial number of each record, and the entry_point_name attribute is the name of the entry point. The exit_point entity has two attributes: sno and exit_point_name. The sno attribute is the serial number of each record, and the exit_point_name attribute is the name of the exit point.

CHAPTER 4

IMPLEMENTATION

4.1 FRONTEND TOOLS

For frontend we have used HTML, CSS and JavaScript.

4.1.1 HTML

Hyper Text Markup Language is the most basic building block of the Web. It defines the meaning and structure of web content. Other technologies besides HTML are generally used to describe a web page's appearance/presentation (CSS) or functionality/behavior (JavaScript). "Hypertext" refers to links that connect web pages to one another, either within a single website or between websites. Links are a fundamental aspect of the Web. By uploading content to the Internet and linking it to pages created by other people, you become an active participant in the World Wide Web. HTML uses "markup" to annotate text, images, and other content for display in a Web browser. HTML markup includes special "elements" such as <head>, <title> An HTML element is set off from other text in a document by "tags", which consist of the element name surrounded by "<" and ">". The name of an element inside a tag is case insensitive. That is, it can be written in uppercase, lowercase, or a mixture. For example, the <title> tag can be written as <Title>, <TITLE>, or in any other way.

4.1.2 CSS

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language such as HTML.CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript. CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts. This separation can improve content accessibility; provide more flexibility and control in the specification of presentation characteristics; enable multiple web pages to share formatting by specifying the relevant CSS in a separate. CSS file, which reduces complexity and repetition in the structural content; and enables the CSS file to be cached to improve the page load speed between the pages that share the file and its formatting.

Separation of formatting and content also makes it possible to present the same markup page in different styles for different rendering methods, such as on-screen, in print, by voice (via speech-based browser or screen reader), and on Braille- based tactile devices. CSS also has rules for alternate formatting if the content is accessed on a mobile device.

The name cascading comes from the specified priority scheme to determine which style rule applies if more than one rule matches a particular element. This cascading priority scheme is predictable.

The CSS specifications are maintained by the World Wide Web Consortium (W3C). Internet media type (MIME type) text/CSS is registered for use with CSS by RFC 2318 (March 1998). The W3C operates a free CSS validation service for CSS documents. In addition to HTML, other markup languages support the use of CSS including XHTML, plain XML, SVG, and XUL.

4.1.3 JAVASCRIPT

Often abbreviated JS, is a programming language that is one of the core technologies of the World Wide Web, alongside HTML and CSS. Over 97% of websites use JavaScript on the client side for web page behavior, often incorporating third- party libraries.

All major web browsers have a dedicated JavaScript engine to execute the code on users' devices. JavaScript is a high-level, often just-in-time compiled language that conforms to the ECMAScript standard. It has dynamic typing, prototype-based object-orientation, and first-class functions. It is multi-paradigm, supporting event- driven, functional, and imperative. programming styles. It has application programming interfaces (APIs) for working with text, dates, regular expressions, standard data structures, and the Document Object Model (DOM).

The ECMAScript standard does not include any input/output (I/O), such as networking, storage, or graphics facilities. In practice, the web browser or other runtime system provides JavaScript APIs for I/O. JavaScript engines were originally used only in web browsers, but are now core components of some servers and a variety of applications like the most important Node.js. Although Java and JavaScript are similar in name, syntax, and respective standard libraries, the two languages are distinct and differ greatly in design.

4.2 BACKEND TOOLS

For backend we have used MySQL as database, PHP to fetch data from database, phpMyAdmin to store data in the local host.

4.2.1 MySQL

MySQL is a popular open-source relational database management system (RDBMS) that is widely used for web-based applications and data storage. It is developed, distributed, and supported by Oracle Corporation. MySQL is known for its reliability, robustness, and ease of use.

MySQL is based on the relational model and uses SQL (Structured Query Language) for querying and manipulating data. It supports a wide range of data types, including strings, numbers, dates, and binary data, and it can handle large amounts of data. MySQL also supports various storage engines, such as InnoDB, MyISAM, and Memory, which allow for different levels of data durability and performance. One of the main advantages of MySQL is its ability to handle high-concurrency and large amounts of data with high performance and low latency. It is used by many popular websites and applications, such as Facebook, WordPress, and Wikipedia.

MySQL can be used on various platforms, including Windows, Linux, and macOS. It can be installed as a standalone server or as a part of a web stack, such as LAMP (Linux, Apache, MySQL, and PHP) or WAMP (Windows, Apache, MySQL, and PHP).

4.2.2 PHP

PHP (Hypertext Preprocessor) is a popular open-source programming language that is widely used for web development and server-side scripting. It is designed to create dynamic web pages and

applications, and it is often used in combination with other technologies, such as HTML, CSS, and JavaScript.

PHP is a server-side language, which means that it runs on the server and generates HTML or other output to be sent to the client's web browser. It can be used to create web pages with dynamic content, handle forms and user input, connect to databases, and perform other tasks that can't be done with HTML and JavaScript alone. One of the main advantages of PHP is its ease of use and flexibility. It has a simple syntax and can be easily integrated with other technologies, such as databases and web servers. PHP also has a large and active community of users and developers, and it is well-documented with a wide range of tutorials, guides, and forums available online. PHP is supported by many popular web servers, such as Apache and Nginx, and it can run on various platforms, including Windows, Linux, and macOS. It is often used in combination with other technologies, such as MySQL, to create a LAMP or WAMP stack.

Popular examples of websites and applications that use PHP include WordPress, Facebook, and Wikipedia. PHP is also used in a wide variety of frameworks, such as Laravel, CodeIgniter, and Symphony, that help developers to build web applications more efficiently.

4.2.3 phpMyAdmin

phpMyAdmin is a free and open-source web-based tool that allows users to manage MySQL databases and servers. It is written in PHP and provides a user-friendly interface for creating, modifying, and querying databases. It is widely used by developers, database administrators, and other users who need to manage MySQL databases. phpMyAdmin provides a variety of features that make managing databases easy, such as:

- A visual interface for creating, modifying, and querying tables and fields
- A SQL editor for executing custom SQL queries
- A user management system for creating and managing MySQL users and their permissions
- Support for importing and exporting data in various formats, such as CSV, SQL, and XML
- Support for managing multiple servers and databases

phpMyAdmin can be installed on various platforms, including Windows, Linux, and macOS, and it can be easily integrated with web servers, such as Apache and Nginx. It can be accessed via a web browser, making it easy to use for users who are not familiar with command-line tools. It is well-documented and has a large and active community of users and developers. It is often used in conjunction with other technologies, such as PHP and MySQL, to create a LAMP or WAMP stack.

4.2.4 XAMPP

XAMPP using the Apache is webserver on the Localhost on the same IP. For connectivity, we use XAMPP app so that we can get the localhost connection to the web browser. In XAMPP, we start the Apache and MYSQL modules.

As a web server, Apache is responsible for accepting directory (HTTP) requests from the internet users and sending them their desired information in the form of files and web pages. Most of the web’s software and code is designed to work along with Apache’s features. MySQL is a relationship database management system based on SQL – Structured Query Language.

CHAPTER 5

SNAPSHOTS

The below screenshots are code of the database of TOLL MANAGEMENT SYSTEM

connect.php

templates > connect.php

```php
<?php
    $vehicle_type = $_POST['vehicle_type'];
    $vehicle_no = $_POST['vehicle_no'];
    $date = $_POST['date'];
    $time = $_POST['time'];
    $entry_point = $_POST['entry_point'];
    $exit_point = $_POST['exit_point'];
    $amount = $_POST['amount'];

    $conn = new mysqli('localhost','root','','tollease');
    if($conn->connect_error){
        die('Connection Failed : '.$conn->connect_error);
    }else{
        $stmt = $conn->prepare("insert into billing(vehicle_type,vehicle_no,date,time,entry_point,exit_point,amount)
        values(?,?,?,?,?,?,?)");
        $stmt->bind_param("ssssssi",$vehicle_type,$vehicle_no,$date,$time,$entry_point,$exit_point,$amount);
        $stmt->execute();
        echo "<a href=/templates/billing_page.php>Billing Successfull</a>";
        $stmt->close();
        $conn->close();
    }
?>
```

Figure 3. connect.php Code

Figure 3:

The above code is to connect the billing page to the database. The values given in the billing are then stored in the database. In this code we have to specify the database name and the attributes of the table.

```php
<select name="vehicle_type" id="vehicle_type" class="io2" value="Select The Vehicle Type" required>
    <option value="entry_point">Select Entry Point</option>
    <?php
    while ($r = mysqli_fetch_array($s)) {
    ?>
        <option><?php echo $r['vehicle_type']; ?></option>
    <?php
    }
    ?>
</select>
```

Figure 4. php code to fetch the vehicle_type present in the vehicle_type table

```
<select name="entry_point" id="entry_point" class="io2">
    <option value="entry_point">Select Entry Point</option>
    <?php
    while ($enp = mysqli_fetch_array($en)) {
    ?>
        <option><?php echo $enp['entry_point_name']; ?></option>
    <?php
    }
    ?>
</select>
<br><br>
<label for="">Exit Point: </label>
<br>
<select name="exit_point" id="exit_point" class="io2">
    <option value="exit_point">Select Exit Point</option>
    <?php
    while ($ext = mysqli_fetch_array($ex)) {
    ?>
        <option><?php echo $ext['exit_point_name']; ?></option>
    <?php
    }
    ?>
</select>
```

Figure 5. php code to fetch the entry_point_name and exit_point_name present in the entry_point and exit_point tables respectively.

Figure 4 & Figure 5:

The above contains code to fetch the data from the database tables VEHICLE_TYPE, ENTRY_POINT and EXIT_POINT and display in the Drop-down-box in the billing page.

```
<script type="text/javascript" src="https://www.gstatic.com/charts/loader.js"></script>
<script type="text/javascript">
  google.charts.load('current', {
    'packages': ['corechart']
  });
  google.charts.setOnLoadCallback(drawChart);

  function drawChart() {
    var data = google.visualization.arrayToDataTable([
      ['date', 'total_vehicle'],
      <?php
      $sql2 = "SELECT time, total_vehicle FROM vehicle_in_a_day";
      $fire2 = mysqli_query($connn, $sql2);
      while ($result = mysqli_fetch_assoc($fire2)) {
        echo "['" . $result['time'] . "'," . $result['total_vehicle'] . "],";
      }
      ?>

    ]);

    var options = {
      title: 'Number Of Vehicles In A Day',
      curveType: 'none',
      legend: {
        position: 'bottom'
      }
    };

    var chart = new google.visualization.LineChart(document.getElementById('curve_chart2'));
    chart.draw(data, options);
  }
</script>
```

Figure 6. Google charts template code to display the graph

Figure 6:

The above code fetches the data from the database table. Google chart is used to display the data in graphical representation. The data for the graph is fetched from the view created in our database TollEase. Php code is used to access the data from the table.

In the above code also contains line to define the graph and its characteristics like title, curve type and position of the graph.

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1	sno	int(11)			No	*None*		AUTO_INCREMENT	Change Drop More
☐	2	Vehicle_type	varchar(30)	utf8mb4_general_ci		No	*None*			Change Drop More
☐	3	Vehicle_no	varchar(10)	utf8mb4_general_ci		No	*None*			Change Drop More
☐	4	date	date			No	*None*			Change Drop More
☐	5	time	time			No	*None*			Change Drop More
☐	6	entry_point	varchar(30)	utf8mb4_general_ci		No	*None*			Change Drop More
☐	7	exit_point	varchar(30)	utf8mb4_general_ci		No	*None*			Change Drop More
☐	8	amount	int(3)			No	*None*			Change Drop More

Figure 7. Billing Table

Figure 7: The above snapshot shows the database structure of the billing table used in the project where it is showing the attributes used and their data types.

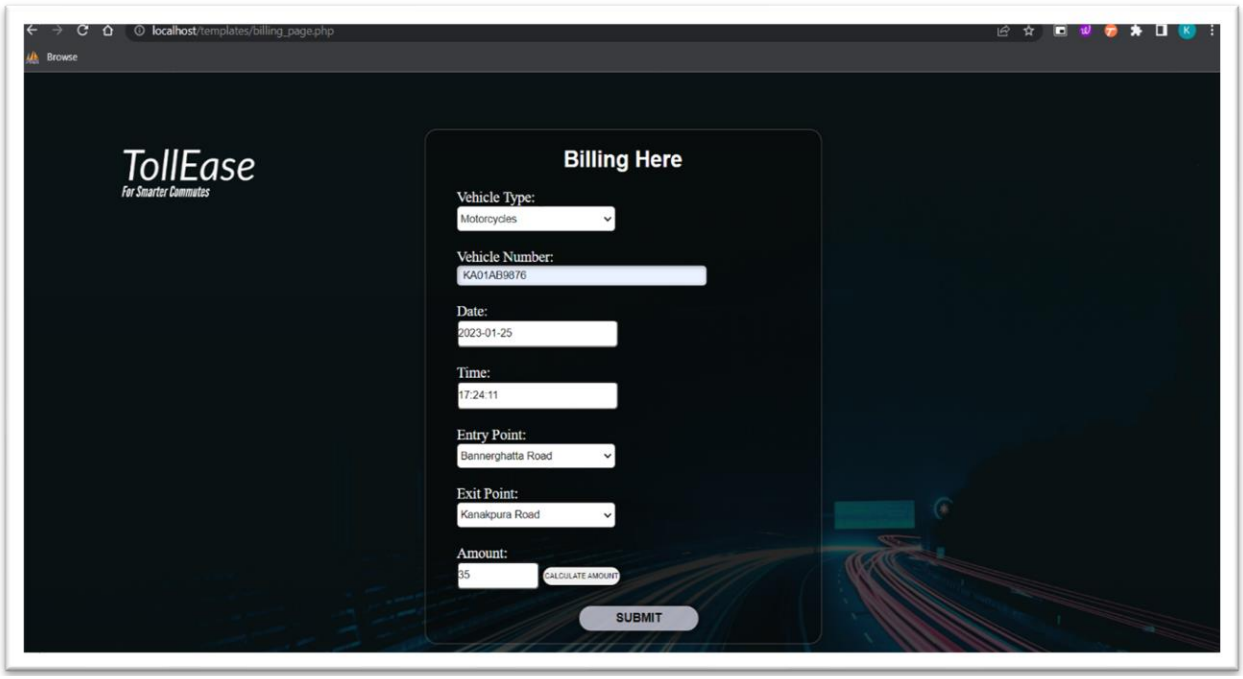

Figure 8. Billing Page

Figure 8: The above screenshot shows the Billing page of the project which can be accessed only by the admins where all the tickets are generated for the vehicles entering the Toll and this page uses php, JavaScript and other frontend tools.

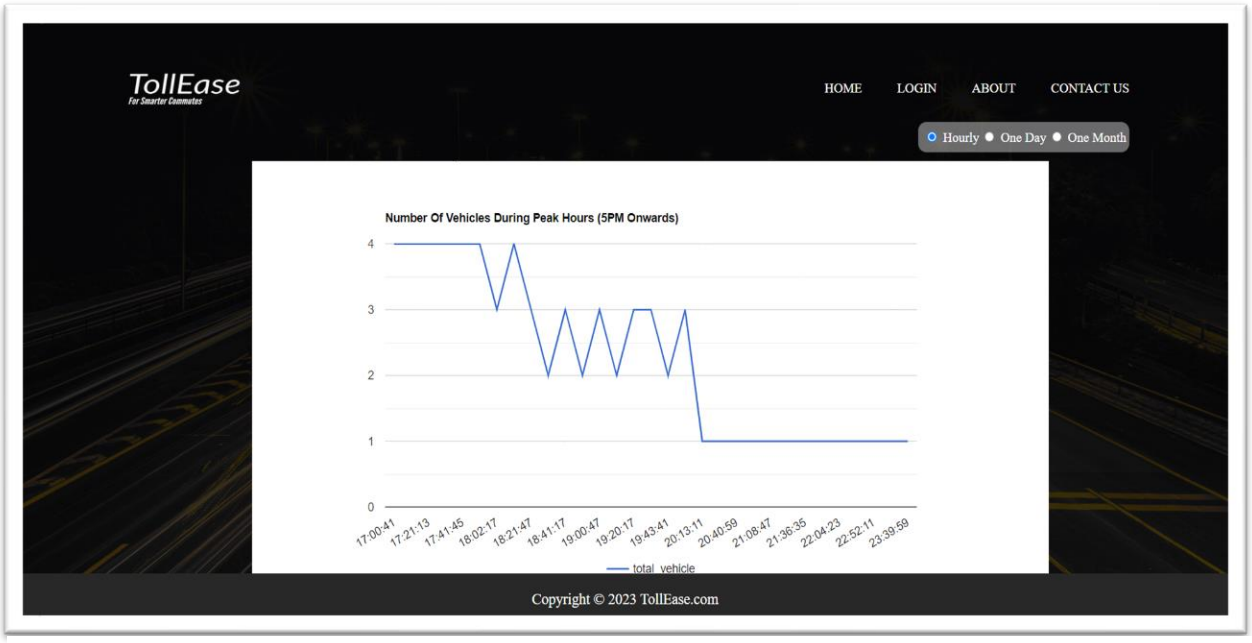

Figure 9. Index Page with Hourly Graph

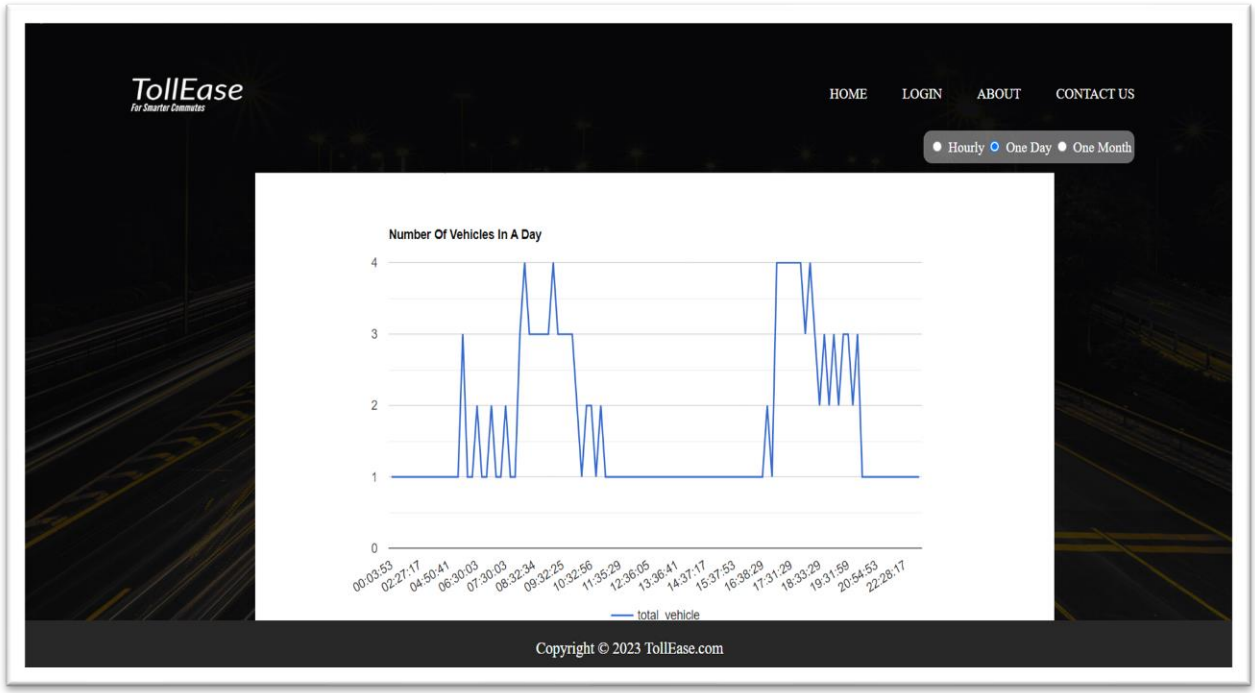

Figure 10. Index Page with One Day Graph

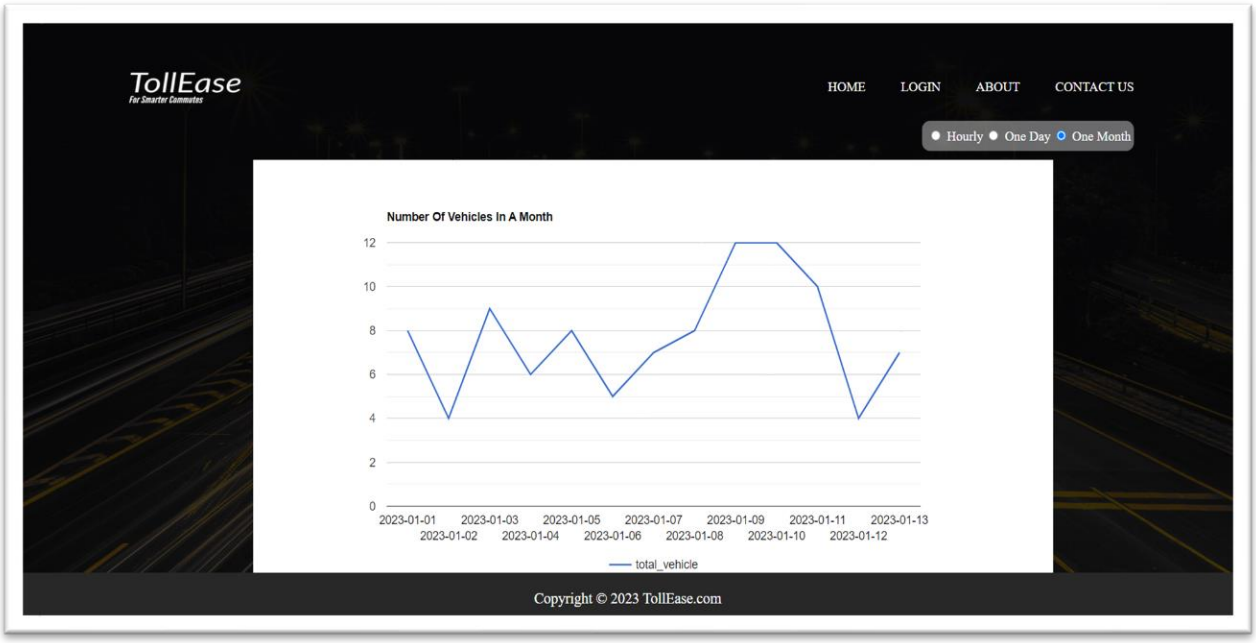

Figure 11. Index Page with One Month Graph

Figure 9, 10 & 11: The below screenshots show the graphical representation about the traffic updates to all the people who open the front page of our website where it shows about the number of vehicles using Toll Road. It can be shown in 3 different time intervals:

- Hourly
- One Day
- One Month

CHAPTER 6

CONCLUSION AND FUTURE ENHANCEMENTS

To conclude the description about the project: The project is developed using MySQL, PHP, phpMyAdmin, HTML, CSS, JS and other dependencies on the requirement specification of the user and the analysis of the existing system, with flexibility for future enhancement.

The expanded functionality of today's software requires an appropriate approach towards software development. The application is developed and designed for people to analyze the traffic and toll plaza to ease their work.

In future enhancements, we plan to add other features like:

- Adding a Backup feature so that we can retrieve our data if the server goes down or even if we have to transport from one domain to another.
- Further we plan to store the database online so we can access it easily. By online we mean to invest in web server and Domain names and getting a .com URL for the Website
- With the help of Machine-Learning we can predict the traffic by using the existing data.
- Generating the printable bill after every transaction.

CHAPTER 7

REFERENCE

1. https://www.geeksforgeeks.org/dbms/
2. https://stackoverflow.com/
3. https://getbootstrap.com/
4. https://developers.google.com/chart
5. https://www.w3schools.com/php/
6. https://youtu.be/2HVKizgcfjo