PROFORMA FOR THE APPROVAL PROJECT

PRN No.: 2018016400907983 & 2018016400907832 Roll No: ______21 & 36______

1. Name of the Student:

   Siddhesh Sharad Mhatre & Gururaj Jagajeevan Poojary ______________________________

2. Title of the Project:

   Toll Booth Management System ____

3. Name of the Guide:

   Mrs. Vanita N Valakati ____

4. Teaching experience of the Guide: _____________5 years_________________________

5. Is this your first submission? Yes ● No □

Signature of the Student Signature of the Guide

Date: 07/04/2021_________ Date: ____________________

Signature of the Coordinator

Date: ____________________

TOLL BOOTH MANAGEMENT SYSTEM

A Project Report submitted in partial fulfilment

Requirements for the award of the Degree of

BACHELOR OF SCIENCE (INFORMATION TECHNOLOGY)

By

Siddhesh Sharad Mhatre

Roll No. 21

Under the esteemed guidance of

Mrs. Vanita N Valakati

Lecturer

DEPARTMENT OF INFORMATION TECHNOLOGY

SIES COLLEGE OF COMMERCE & ECONOMICS.

(Affiliated to University of Mumbai)

Mumbai, 400022

MAHARASHTRA

2020-21

TOLL BOOTH MANAGEMENT SYSTEM

A Project Report submitted in partial fulfilment

Requirements for the award of the Degree of

BACHELOR OF SCIENCE (INFORMATION TECHNOLOGY)

By

Gururaj Jagajeevan Poojary

Roll No. 36

Under the esteemed guidance of

Mrs. Vanita N Valakati

Lecturer

DEPARTMENT OF INFORMATION TECHNOLOGY

SIES COLLEGE OF COMMERCE & ECONOMICS.

(Affiliated to University of Mumbai)

Mumbai, 400022

MAHARASHTRA

2020-21

SIES COLLEGE OF COMMERCE & ECONOMICS.

(Affiliated to University of Mumbai)

MUMBAI – MAHARASHTRA – 400022

DEPARTMENT OF INFORMATION TECHNOLOGY

CERTIFICATE

This is to certify that the project entitled, **"Toll Booth Management System"**, is bonafied work of **Siddhesh Sharad Mhatre** bearing Seat. No: **21** submitted in partial fulfillment of the requirements for the award of degree of BACHELOR OF SCIENCE in INFORMATION TECHNOLOGY from University of Mumbai.

Internal Guide **Coordinator**

External Examiner

Date: 07/04/2021 **College Seal**

SIES COLLEGE OF COMMERCE & ECONOMICS.

(Affiliated to University of Mumbai)

MUMBAI – MAHARASHTRA – 400022

DEPARTMENT OF INFORMATION TECHNOLOGY

CERTIFICATE

This is to certify that the project entitled, **"Toll Booth Management System"**, is bonafied work of **Gururaj Jagajeevan Poojary** bearing Seat. No: **36** submitted in partial fulfillment of the requirements for the award of degree of BACHELOR OF SCIENCE in INFORMATION TECHNOLOGY from University of Mumbai.

Internal Guide **Coordinator**

External Examiner

Date: 07/04/2021 **College Seal**

Abstract

Managing multiple toll booths is a very complicated task. The system here is a smart card based Toll booth system that is monitored using IOT. The Internet server maintains all the data of user accounts and also their balance. All vehicle owners would possess an RFID (Radio Frequency Identification) based card that stores their RFID number. The system at toll booths will monitor the cards scanned when a car arrives at the toll booth. The system then connects to the online server to check if the card is valid and if valid what is the balance. If user balance is sufficient, the toll amount is deducted online and web system sends signal back to the card scanner system that the user has been billed. On receiving this signal the system operates a motor to open the toll gate for that car. The system is controlled by a microcontroller to achieve this purpose. Also system allows to store data of all the vehicles passed at particular time intervals for later reference and surveillance. This system thus automates the entire toll booth billing and monitoring process with ease using RFID plus IOT based system.

ACKNOWLEDGEMENT

In the present world of competition there is a race of existence in which those are having will to come forward succeed. Project is like a bridge between theoretical and practical working. With this willing I created this particular project.

I would like to express my profound gratitude and deep regard to my project guide, Mrs. Vanita N Valakati, for her exemplary guidance, valuable feedback and constant encouragement throughout the duration of the project. Her valuable suggestions were of immense help throughout my project work. Working under her was an extremely knowledgeable experience for me.

At last but not the least I take this opportunity to thank all those who have been directly or indirectly related to my project.

DECLARATION

I hereby declare that the project entitled, **“Toll Booth Management System”** done at **SIES COLLEGE OF COMMERCE & ECONOMICS**, has not been in any case duplicated to submit to any other university for the award of any degree. To the best of my knowledge other than me, no one has submitted to any other university.

The project is done in partial fulfillment of the requirements for the award of degree of **BACHELOR OF SCIENCE (INFORMATION TECHNOLOGY)** to be submitted as final semester project as part of our curriculum.

Siddhesh Sharad Mhatre
Name and Signature of the Student

DECLARATION

I hereby declare that the project entitled, **“Toll Booth Management System”** done at **SIES COLLEGE OF COMMERCE & ECONOMICS**, has not been in any case duplicated to submit to any other university for the award of any degree. To the best of my knowledge other than me, no one has submitted to any other university.

The project is done in partial fulfillment of the requirements for the award of degree of **BACHELOR OF SCIENCE (INFORMATION TECHNOLOGY)** to be submitted as final semester project as part of our curriculum.

Gururaj Jagajeevan Poojary

Name and Signature of the Student

TABLE OF CONTENTS

List of Figures

Chapter 1
Introduction

1.1 Background

With the growth in the number of vehicles the need for expansive roads catering to thousands of vehicles moving across India has become inevitable. However, considering the present situation the current toll system has several drawbacks. Due to the limited number of toll booths and slow collection process, the average waiting time per vehicle is 10 minutes. This results in losses worth thousands of crores of Rupees in terms of fuel wastage. This long wait time often results in drivers getting irritated resulting in verbal spats and physical fights among people and the toll attendants. Several such incidents have been reported in the press with some of these fights even resulting in the death of the toll plaza attendants. In addition, there are numerous cases of toll plaza accidents which happen due to the sudden lane changing by drivers for faster clearance. The major reason behind this is that, the security at the tolls is insufficient and it is beyond the traffic polices control to manage the vast number of vehicles. We keep hearing of many such incidents at toll plazas which mostly occur due to negligence either on the people's side or due to lack of control from the government agencies including the police. In case of events, where lives are lost, such losses are a life shattering experience.

As is well known, in such a scenario, the general public is a little hesitant in taking responsibilities of any such incident. Hence the government has to come up with an effective plan which bridges the gap between the toll management and the public expectation of the service that they experience. Introduction of an elective toll plaza operation plan by the government, its strict implementation and monitoring which would result in a more efficient and a more responsive and efficient system could be a good option for easing the challenges associated with the existing tolling process.

The main advantages of web application development as well as popular development features covering integration with different technologies are considered. Integration and possibilities of application of cloud based web applications in real scenarios with different embedded Internet of Things (IOT) devices are considered and described. The design and implementation of a cloud based web application supporting vehicle toll payment system using IOT device is presented and described. The development framework as well as featured and popular technologies used to realize a vehicle toll payment by IOT device are described. The concept of vehicle toll payment over an online payment system is also described. Processing, monitoring and control in the cloud based web application of such payments using IOT devices are described and presented. Also system allows to store data of all the vehicles passed at particular time intervals for later reference and surveillance. This system thus automates the entire toll booth collection and monitoring process with ease using RFID plus IOT based system.

1.2 Objectives

The main objective of this system is to ease the challenges associated with the existing tolling process. Managing multiple toll booths is a very complicated task. The system here is a smart card based toll booth system that is monitored using IOT. The Internet server maintains all the data of user accounts and also their balance. The system at toll booths will monitor the cards scanned when a car arrives at the toll booth. If user balance is sufficient, the toll amount is deducted online and web system sends signal back to the card scanner system that the user has been billed. On receiving this signal the system operates a motor to open the toll gate for that car. Also system allows to store data of all the vehicles passed at particular time intervals for later reference and surveillance.

1.3 Purpose and Scope

1.3.1 Purpose

Now we are living in new generation and facing a problem is not a big issue, but for getting a solution of that problem is a challenge of a person. The particular topic is chosen to manage all these above situations and to facilitate the vehicles owners, so many toll booth manager systems are available in market but that one's are so much costly as well as they are not so much efficient, reliable, precise and friendly to use. Here we have proposed a system that is called IOT based toll booth manager system which is so much efficient, reliable and friendly to use. IOT based toll booth manager system is such type of a system that provides the automatic toll collection process with the help of internet resources and RFID (radio frequency identification) technology. With the help of cloud based web application we can store data of all the vehicles passed at particular time intervals which can be helped for later security and surveillance.

1.3.2 Scope

The existing Toll Booth runs in a very slow and inconvenient manner. Due to the limited number of toll booths and slow collection process, the average waiting time per vehicle is very high. In addition, there are numerous cases of toll plaza accidents which happen due to the inconvenience caused because of the existing system. So to overcome this situation, we can make use of smart IOT Based Toll booth Management System. The scope of this system is to make the drivers and toll employee life with ease. This proposed system systematically manages the intake and departure of all the vehicles and stores data which can helpful for reference and security. This system thus automates the entire toll booth collection and monitoring process with ease using RFID plus IOT based system.

1.3.3 Applicability

This proposed project is applicable in day to day to basis for every driver. As it provides many aspect to the driver which is useful such as no long queue and also easy and fast money transaction. This project is also beneficial and applicable for Toll Booth as it stores all the vehicles data such as arrival and departure time and also vehicle number in the database. This same project with a slight modification can be used in many sectors rather than being just used by drivers and Toll booth Management.

1.4 Achievements

- The knowledge achieved after the completion of this project:
- Learned the Complete concept of how to create a website.
- Acquired the knowledge of how to build an App in Android OS.
- Gained knowledge about how existing Toll Booth is managed and implemented. Also got to learn how the data of the vehicles are stored in traditional existing system.
- Also gained knowledge about how an automated system named IOT Based Toll Booth Management System can be implemented and can solve the drawbacks of the existing system.
- Got to learn more about networking as we had to connect the android app and the web application together. Obtained more knowledge about system and their interactions.

1.5 Organization Report

The concept of the project IOT Based Toll Booth Management System have an altogether different impact in the world of these technologies. This proposed system can easily solve the problem of long queues and provide secure way of online transaction. This helps the driver i.e. user to have an enjoyable trip. This system ensures complete safety from both ends i.e. Toll Booth and Driver.

This project is a web application creating the required environment. The Web application serves a Database which will be accessible to the Toll Booth Manager which consists of all the vehicle data i.e. arrival time, departure time, vehicle number, etc. for further analysis and control. The data from the RFID such as transaction time, vehicle number, etc. will get automatically updated. RFID Card is used by the User i.e. Driver to pay the Toll Booth Bill. If user balance is sufficient, the toll amount is deducted online and web system sends signal back to the card scanner system that the user has been billed. When the amount is deducted it also calls the motor to open the gate for vehicle to pass through it. If the user doesn’t have insufficient balance the user can recharge the card using our simplest website.

The User side of the Web Application is mostly used to recharge and check the balance available in Wallet. This Web Application performs very fast and quick payment method. This Web Application is suited for reliable and provides user friendly interaction.

Chapter 2

Survey of Technologies

Traditionally, web applications are built by using HTML only. In our project, we are developing web applications by using many more technologies. Web pages are designed by using HTML (Hypertext Mark Language) and also CSS (Cascading Style Sheet). HTML is a standard markup language used to design webpages. It is widely used as it is supported by all browsers and there is no need to purchase any software to support. CSS have the great consistency in design and it has more formatting options. We are also using ASP.NET which is an open source, server-side web application framework created by Microsoft. ASP.NET allows developers to create web applications, web services, and dynamic content-driven websites.

The main work of web applications is the functionality of the web pages. For the functionality of web pages we are using C# language. It is a general-purpose language designed for developing apps on the Microsoft platform and requires the .NET framework on Windows to work. C# is often thought of as a hybrid that takes the best of C and C++ to create a truly modernized language. Although the .NET framework supports several other coding languages, C# has quickly become one of the most popular. C# can be used to create almost anything but is particularly strong at building Windows desktop applications. C# can also be used to develop web applications and has become increasingly popular for mobile development too.

As a back-end tool SQL Server (SSMS) is used which is the database platform. SQL Server is the popular online database because it is very easy to use and it is interfaced very well with ASP.NET C#. SQL Server is very secured, fast and simple, and it is heavily supported on different platforms. ASP.NET C# and SQL Server is the excellent use for developing any web application.

Chapter 3
System Analysis

3.1 Problem Definition

Now we are living in new generation and facing a problem is not a big issue, but for getting a solution of that problem is a challenge of a person. With the growth in the number of vehicles the need for expansive roads catering to thousands of vehicles moving across India has become inevitable. However, considering the present situation the current toll system has several drawbacks. Due to the limited number of toll booths and slow collection process, the average waiting time per vehicle is 10 minutes. This results in losses worth thousands of crores of Rupees in terms of fuel wastage. This long wait time often results in drivers getting irritated resulting in verbal spats and physical fights among people and the toll attendants. Several such incidents have been reported in the press with some of these fights even resulting in the death of the toll plaza attendants. In addition, there are numerous cases of toll plaza accidents which happen due to the sudden lane changing by drivers for faster clearance. The major reason behind this is that, the security at the tolls is insufficient and it is beyond the traffic polices control to manage the vast number of vehicles. We keep hearing of many such incidents at toll plazas which mostly occur due to negligence either on the people's side or due to lack of control from the government agencies including the police. In case of events, where lives are lost, such losses are a life shattering experience.

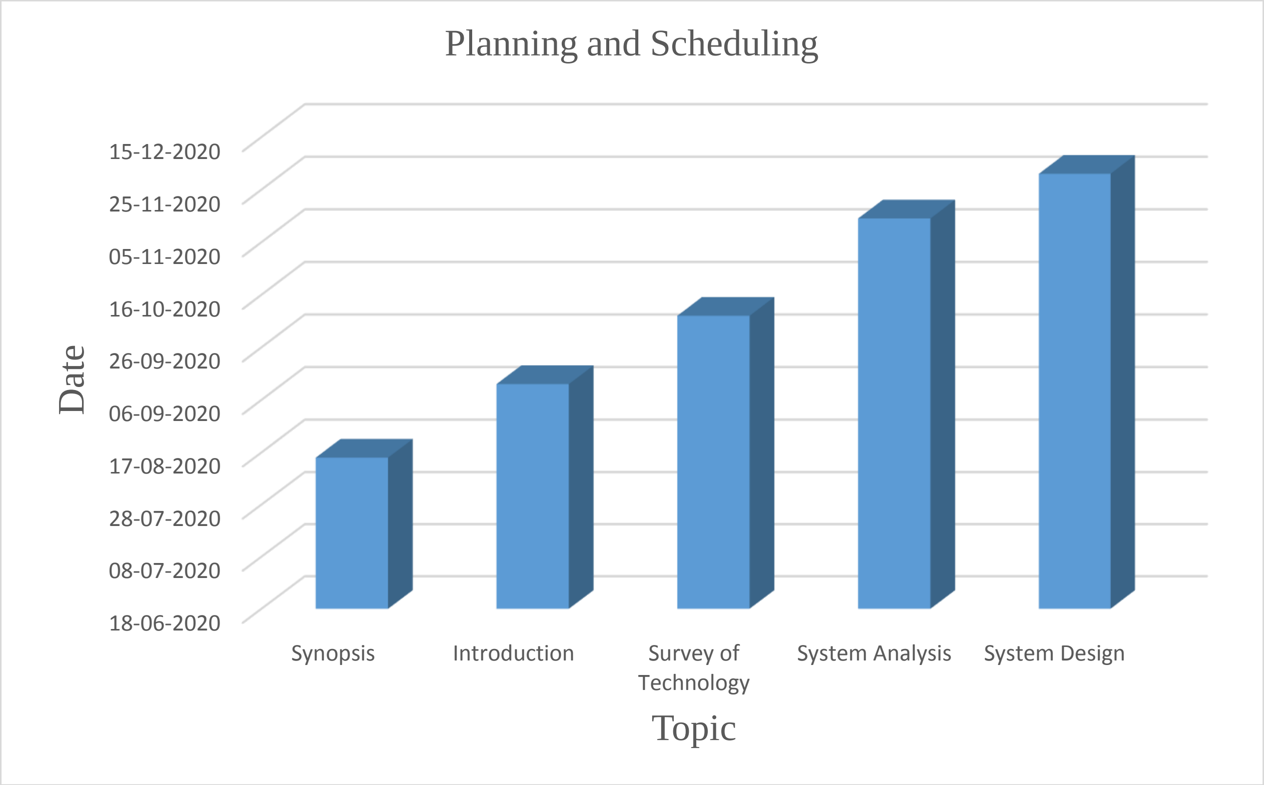
3.2 Existing System

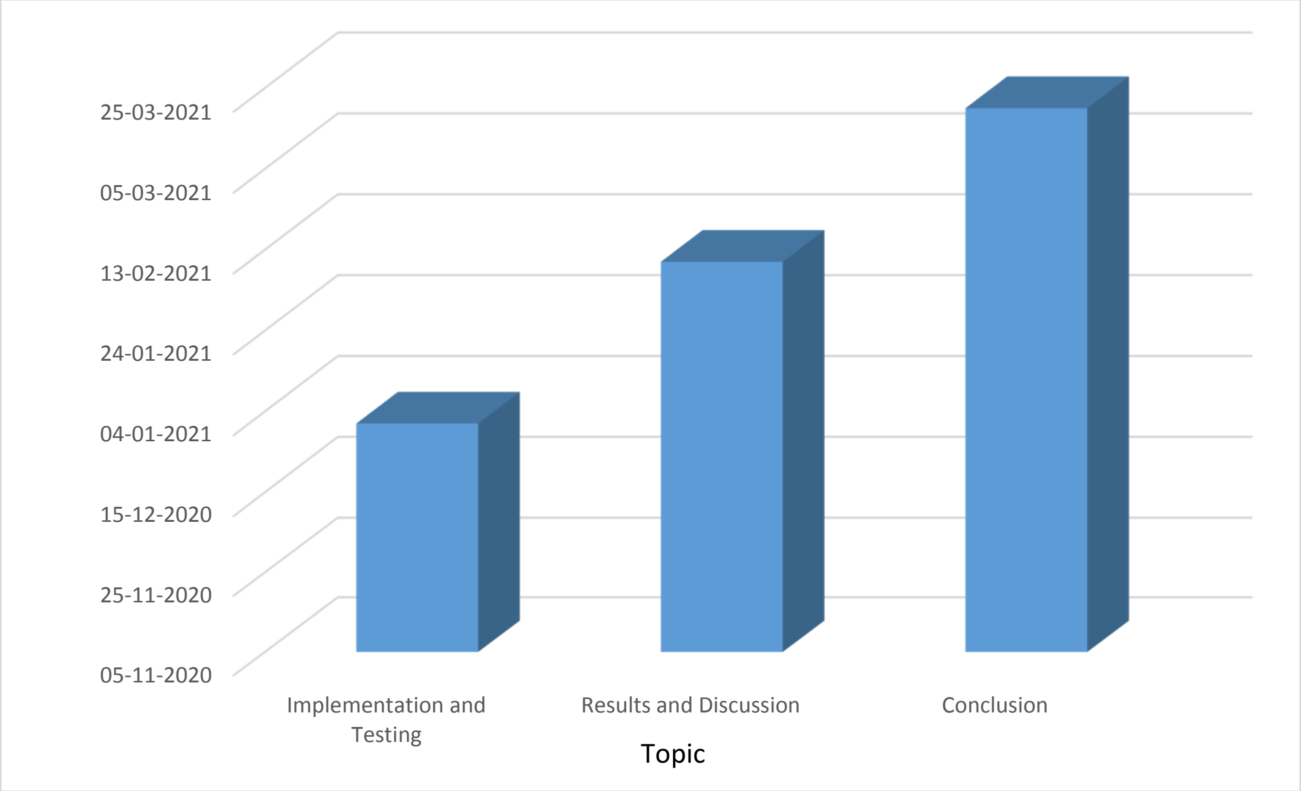
There are many challenges associated with the existing toll booth system. Also, managing multiple toll booth is a very complicated task. The current toll system has several drawbacks. As there are limited number of toll booths and slow collection process, the average waiting time per vehicle is very high. This results in losses worth thousands of crores of Rupees in terms of fuel wastage. This even result in time wastage which is very important aspect. This long wait time often results in drivers getting irritated resulting in verbal spats and physical fights among people and the toll attendants. The major reason behind this is that, the security at the tolls is insufficient and it is beyond the traffic polices control to manage the vast number of vehicles. We keep hearing of many such incidents at toll plazas which mostly occur due to negligence either on the people's side or due to lack of control from the government agencies including the police.

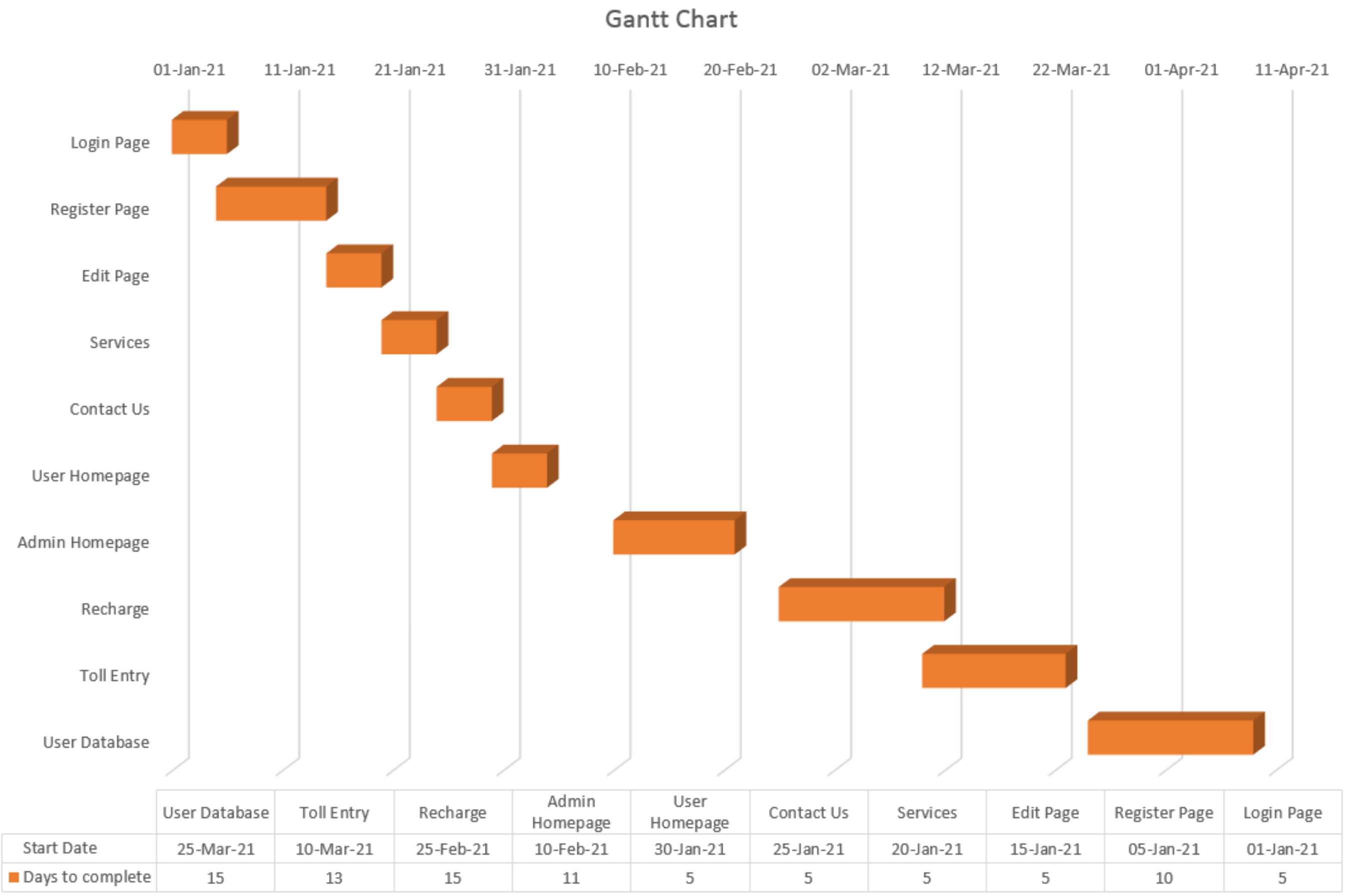
Also, the existing toll booth system only works when there are enough employees to supervise each part of the Toll Booth. Additionally, it requires workers to continuously check the surveillance as well the vehicles arrived at the toll gate. That is why, this existing system is supposed to be time consuming, inconvenient and dependent that means mistake by any employee can result into greater loss. One more important drawback of the existing system is that the vehicle data i.e. arrival time and departure time could not managed properly which can be deal breaker in terms of later surveillance.

3.3 Planning and Scheduling

1. **Gantt chart:**

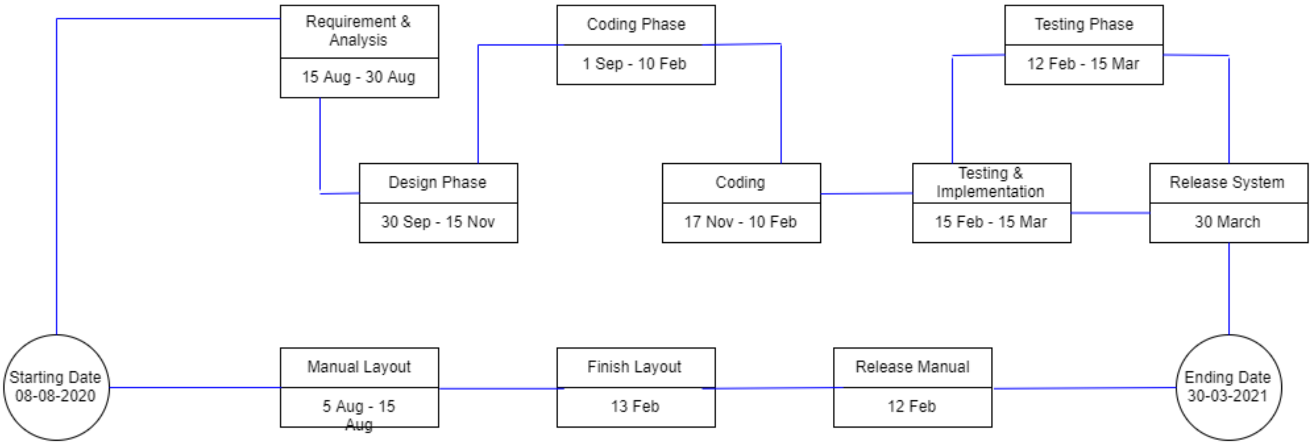






	User Database	Toll Entry	Recharge	Admin Homepage	User Homepage	Contact Us	Services	Edit Page	Register Page	Login Page
Start Date	25-Mar-21	10-Mar-21	25-Feb-21	10-Feb-21	30-Jan-21	25-Jan-21	20-Jan-21	15-Jan-21	05-Jan-21	01-Jan-21
■ Days to complete	15	13	15	11	5	5	5	5	10	5

2. **Pert chart:**



3.4 Proposed System

The proposed System is named as IOT based Toll Booth Management System. This proposed system is fully functional and optimized system. This proposed a system that is called IOT based toll booth manager system which is so much efficient, reliable and friendly to use. This proposed system is a smart card based toll booth system that is monitored using IOT. This smart card can only be used by the owner of the vehicle which can be beneficial as a security perspective. IOT based Management System is an automated system. This proposed system runs without any individual (employee). This system store all the data of the vehicles passed at particular time intervals which can be helped for later security and surveillance. User can access the Toll Booth just by swiping the card. If user balance is sufficient, the toll amount is deducted online and web system sends signal back to the card scanner system that the user has been billed. If the user needs to add money user can do it easily and conveniently using website. The Internet server maintains all the data of user accounts and also their balance.

IOT Based Toll Booth Management System ensures fast, quick and complete safety in transaction. This proposed system does make the users life with ease. Long vehicle queues will be cut off which will result in enjoyable journey and will keep users calm throughout the journey.

3.5 Requirement Analysis

As you can find in today’s age there is a lot of traffic at the Toll Booth stations. Existing Toll Booth results into inconvenience to users i.e. drivers. The earlier system results into fuel wastage as well as time wastage which is supposed to be an important aspect. Sometimes this results in drivers getting irritated resulting in verbal spats and physical fights among people and the toll attendants. To get rid of this current scenario our project named IOT Based Toll Booth Management System came into existence. As this system is an automated and provides fully functional and consistent system this project gives a clear solution to the problem faced by the existing Toll Booth system.

3.6 Requirement Specification

3.6.1 Hardware Specification

Windows:

- 300 MHz or higher processor
- 1 GB of RAM (512 MB minimum)
- 200 MB of available hard disk space

Other Hardware equipment’s:

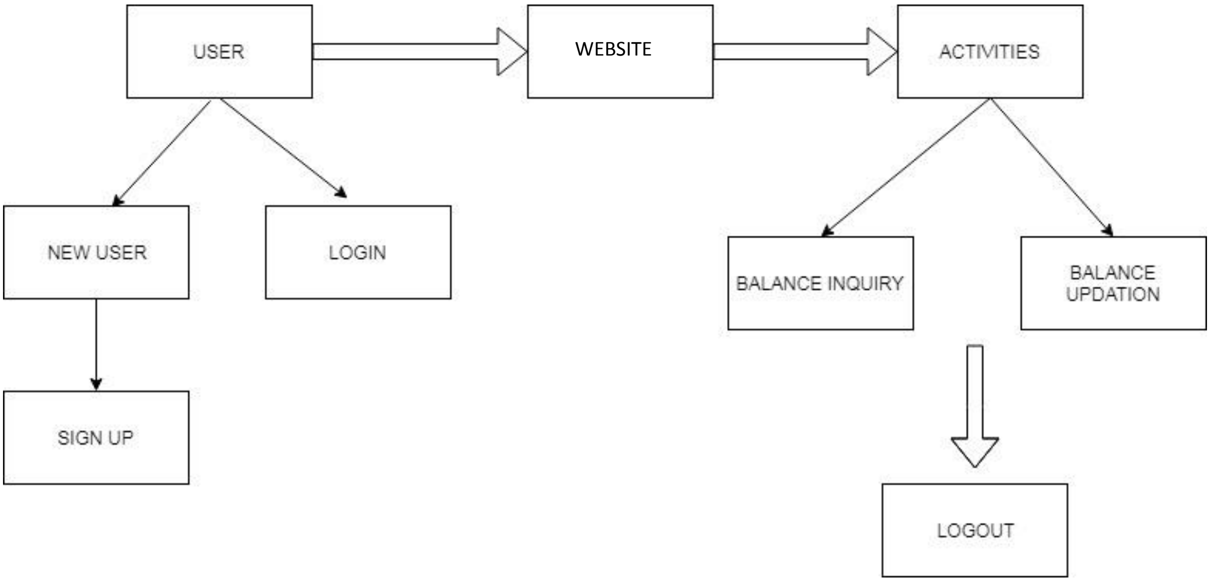
- Arduino Board
- RFID Tag
- RFID Reader
- Motors
- Microcontroller
- LCD Display

3.6.2 Software Specification

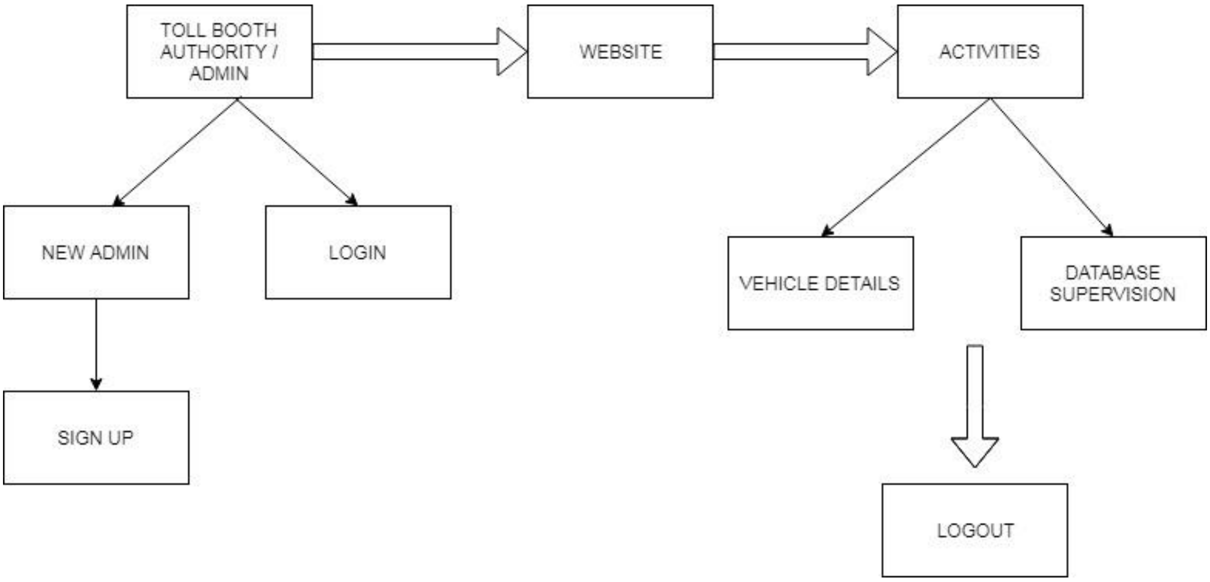
- Arduino Compiler
- SQL Server
- Asp.net
- Html
- C#

3.7 Conceptual Design

1. **User**



2. **Toll Booth Authority / Admin**



3.8 Justification of Selection of Technology

C# is widely used for developing desktop applications, web applications and web services. It is used in creating applications of Microsoft at a large scale. C# is very efficient in managing the system. All the garbage is automatically collected in C#. There is no problem of memory leak in C# because of its high memory backup. Cost of maintenance is less and is safer to run as compared to other languages. C# code is compiled to an intermediate language (Common (.Net) Intermediate Language) which is a standard language, independently irrespective of the target operating system and architecture. That is why we have chosen C# to develop the web page. We are also using ASP.NET along with C# as it is an open source, server-side web application framework which can be used to create web applications, web services, and dynamic content-driven websites.

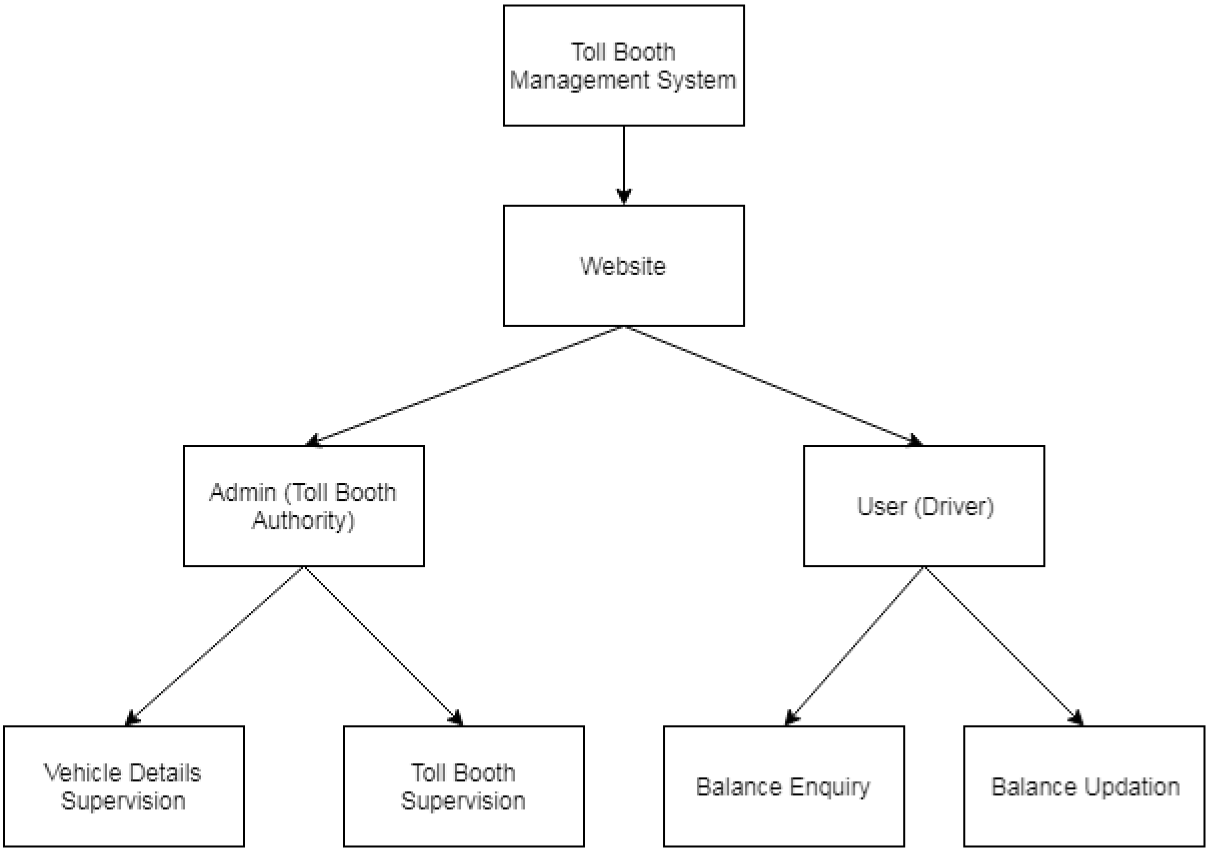
HTML (the Hypertext Markup Language) and CSS (Cascading Style Sheets) are two of the core technologies for building Web pages. HTML is the language for describing the structure of Web pages. CSS is the language for describing the presentation of Web pages, including colors, layout, and fonts.

Structured Query Language (SQL) is the standard and most widely used programming language for relational databases. It is used to create, manage and organize data in all sorts of systems in which various data relationships exist. The application is used for a wide range of purposes, including data warehousing, e-commerce, and logging applications. The most common use for SQL Server however, is for the purpose of a web database. It can be used to store anything from a single record of information to an entire inventory of available products for an online store.

Chapter 4

System Design

4.1 Module Design


Toll Booth Management System
Website
Admin (Toll Booth Authority)
User (Driver)
Vehicle Details Supervision
Toll Booth Supervision
Balance Enquiry
Balance Updation

4.2 Data Dictionary

1. USER:

User_FullName	varchar(50)	Full Name of the User
User_EmailId	varchar(50)	Uniquely identifies this table Email ID of the User
User_Gender	varchar(50)	Gender of the User
User_Address	nvarchar(MAX)	Address of the User
User_Dob	varchar(50)	Date of Birth of the User
User_Phoneno	int	Phone number of the User
User_Vehicleno1	varchar(50)	Vehicle Number 1 of the User
User_Vehicleno2	varchar(50)	Vehicle Number 2 of the User
User_Password	varchar(50)	Email ID of the User
Account_Balance	int	Account Balance of the User
Arrival_Time	DateTime	Arrival Time of the User

2. AUTHORITY / ADMIN

Admin_FullName	varchar(50)	Full name of the Admin
Admin_EmailID	varchar(50)	Uniquely identifies this table Email ID of the Admin
Admin_Gender	varchar(50)	Gender of the Admin
Admin_Dob	varchar(50)	Date of Birth of the Admin
Admin_Phoneno	varchar(50)	Phone number of the Admin
Admin_Password	varchar(50)	Password of the Admin

3. TOLL BOOTH WEBSITE

User_FullName	varchar(50)	Full Name of the User
User_Password	varchar(50)	Password of the User
User_EmailID	varchar(50)	Email ID of the User
User_Phoneno	varchar(50)	Phone number of the User
Vehicle_arrivetime	DateTime	Vehicle arrival time

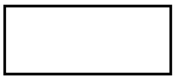
Account_balance	int	Account Balance in RFID Card
Vehicle_no	varchar(50)	Vehicle Number of the User

4.3 Entity Relationship (ER) Diagram

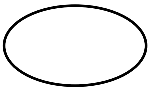
An entity relationship diagram (ERD) is a data modeling technique that graphically illustrates an information system’s entities and the relationships between those entities. An ERD is a conceptual and representational model of data used to represent the entity framework infrastructure. It is used as a high-level logical data model, which is useful in developing a conceptual design for databases.

Elements used in ERD:

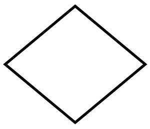
Entity



Attribute

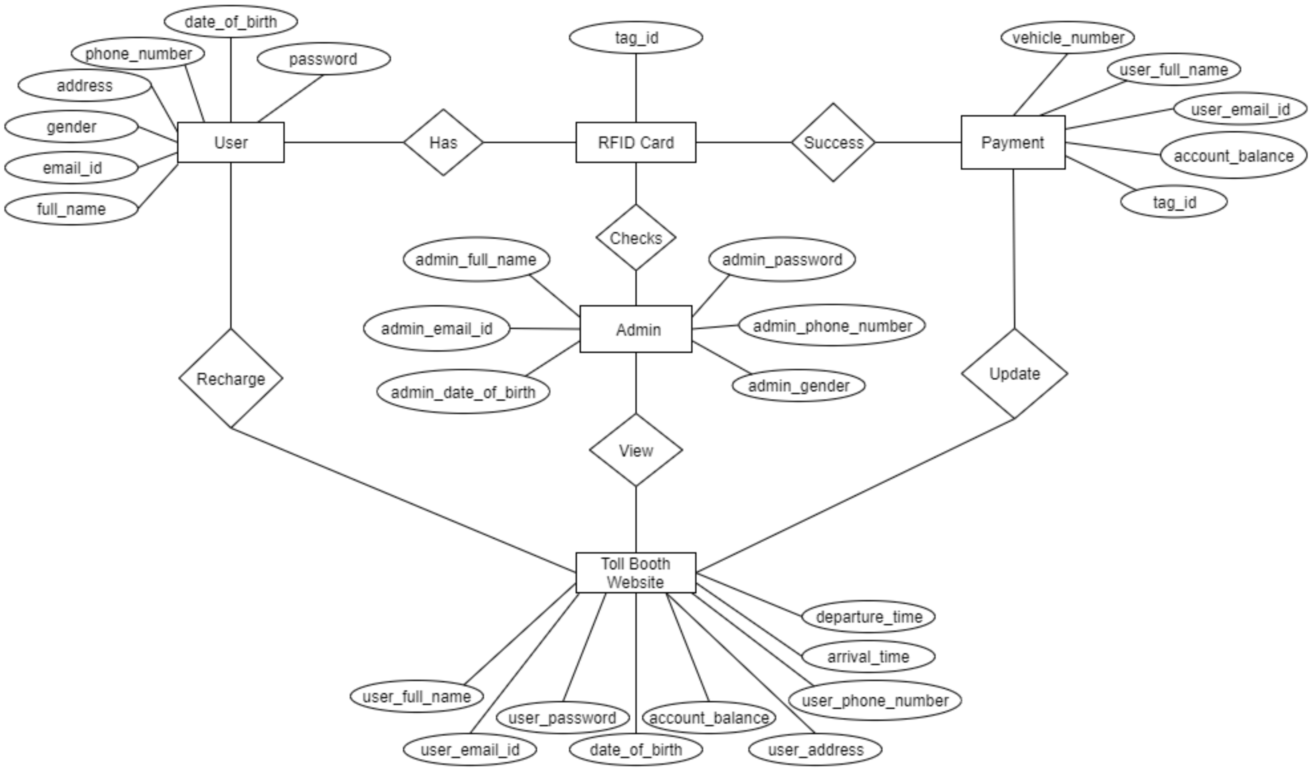


Relationships



One-To-One

Diagram:

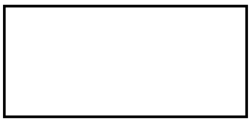


4.4 Data Flow Diagram (DFD)

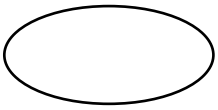
A data flow model is diagrammatic representation of the flow and exchange of information within a system. A DFD is often used as a preliminary step to create an overview of the system which can later be elaborated. DFD can also be used for the visualization of data processing.

Elements of Data Flow diagram:

External Entity



Process



Data Flow

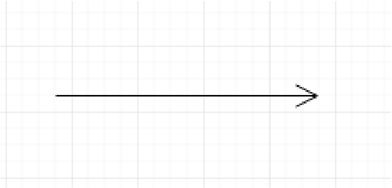
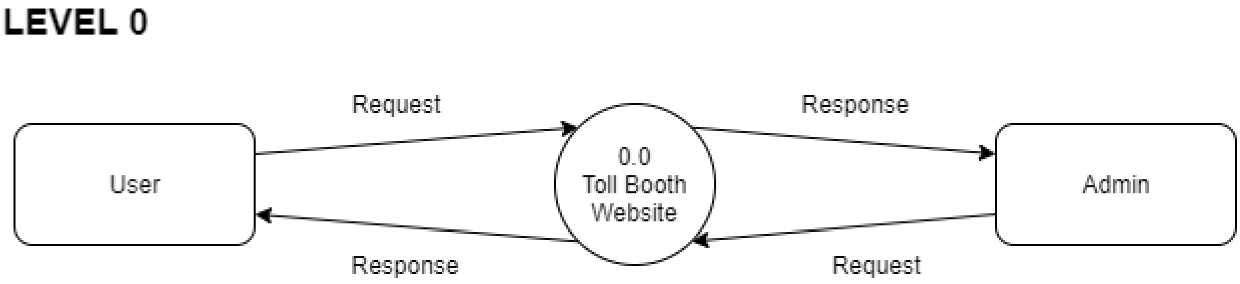
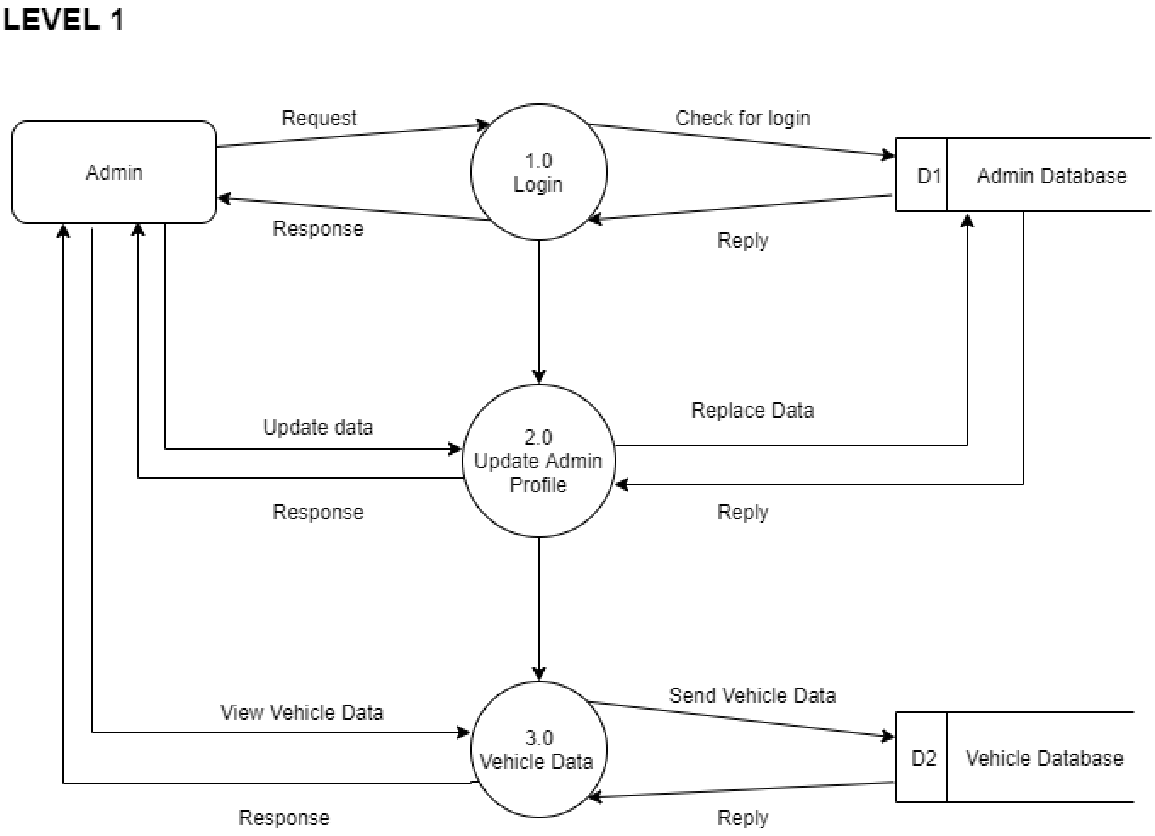
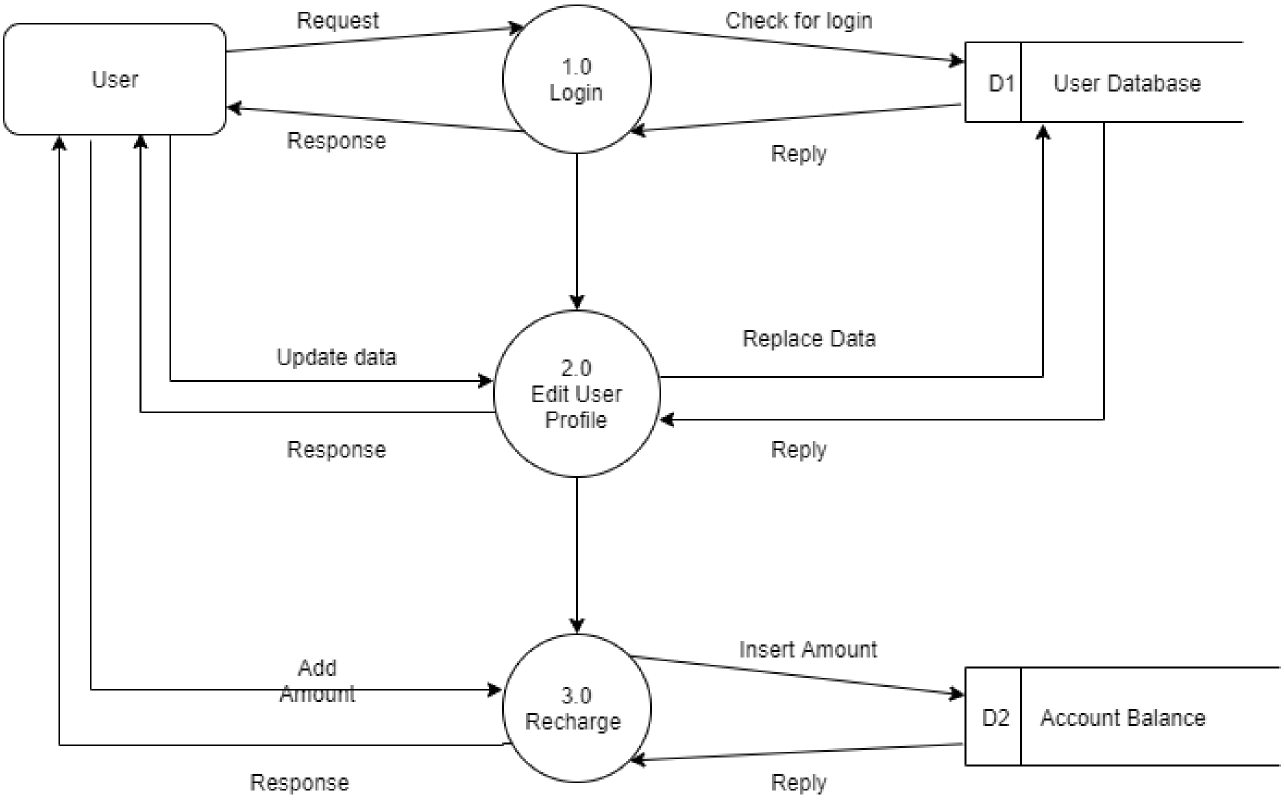


Diagram:





LEVEL 2



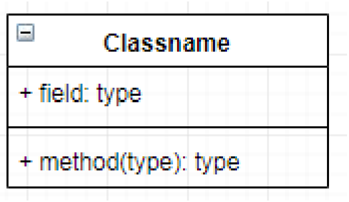
4.5 UML Diagram

4.5.1 Class Diagram

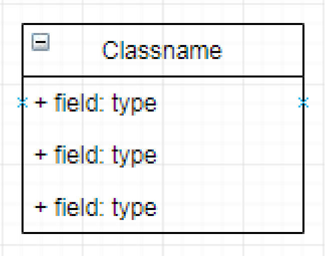
A class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system’s classes, their attributes, operations (or methods) and the relationships among objects.

Elements of the class diagram:

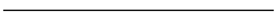
Class, Attribute and Function



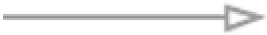
Class and Attribute



Association



Generalization



Relationship

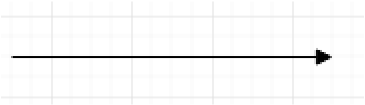
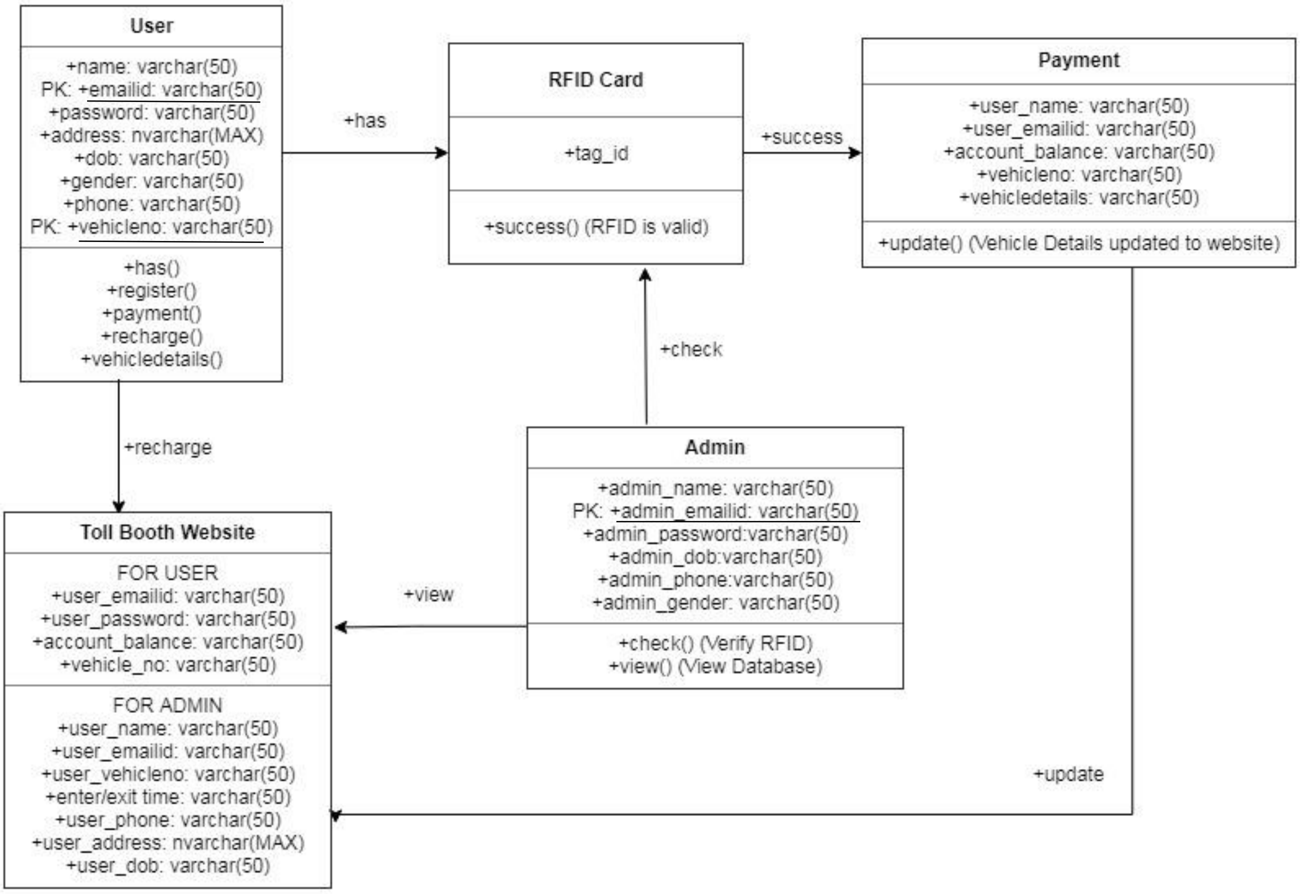


Diagram:

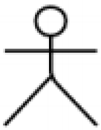


4.5.2 Use Case Diagram

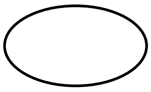
A use case diagram is a graphic depiction of the interactions among the elements of a system. It is used to describe a set of actions (use cases) that some system can perform in collaboration with one or more external users of the system (actors). Each use case provides some observable and valuable result to the actors or other stakeholders of the system. Use case diagrams are twofold: (i)Behavior diagram because they describe behavior of the system (ii)Structure diagrams - as a special case of class diagrams where classifiers are restricted to be either actors or use cases related to each other with associations.

Elements of the Use Case diagram:

Actor



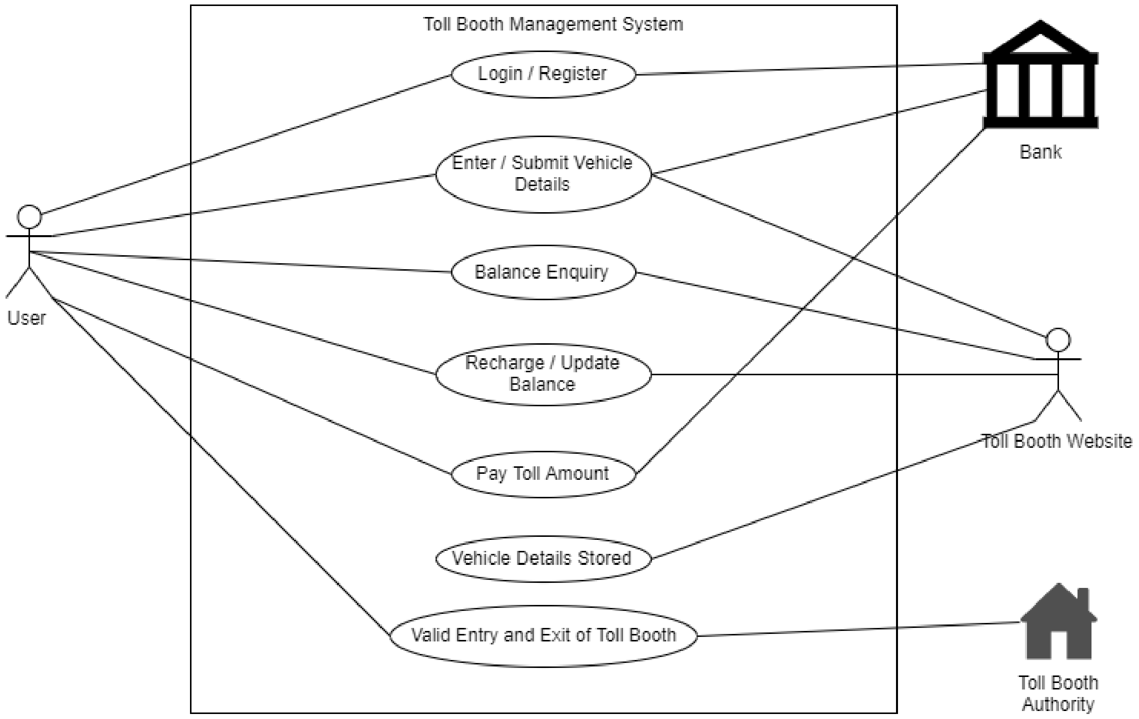
Use Case



Association

System Boundary

Diagram:

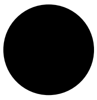

Toll Booth Management System
Login / Register
Enter / Submit Vehicle Details
Balance Enquiry
Recharge / Update Balance
Pay Toll Amount
Vehicle Details Stored
Valid Entry and Exit of Toll Booth
User
Bank
Toll Booth Website
Toll Booth Authority

4.5.3 Activity Diagram

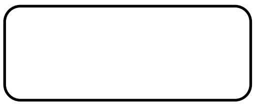
Activity diagram in UML is to describe the dynamic aspects of the system. It is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent.

Elements used in Activity diagram:

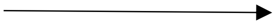
Start Point/Initial Point



Activity



Action Flow



Decision

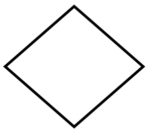
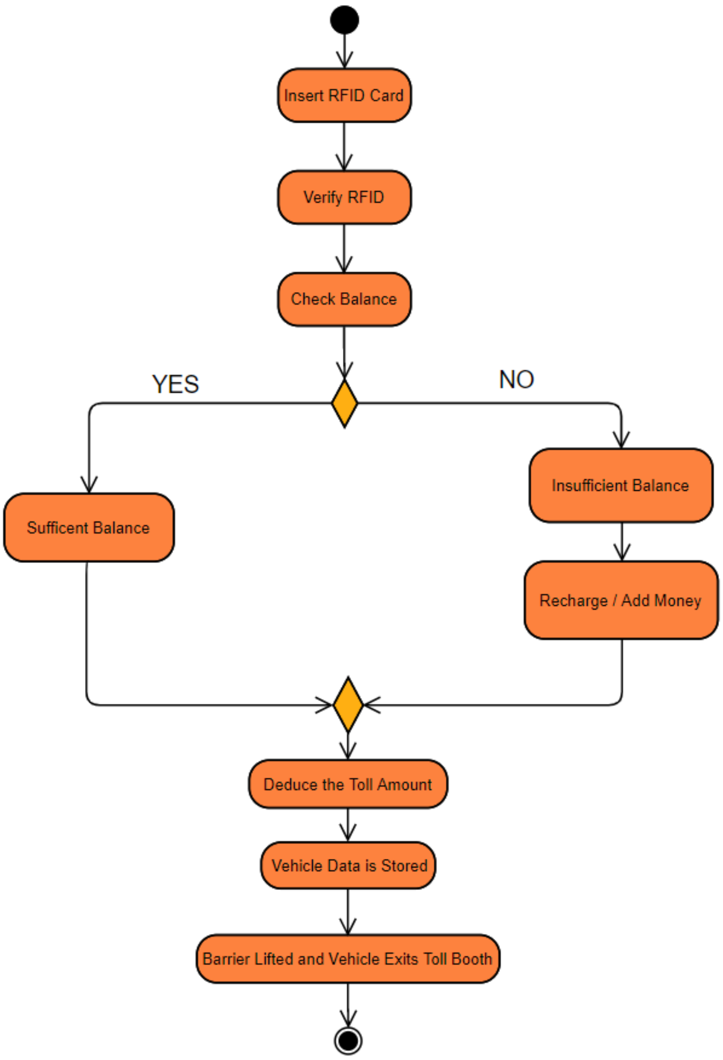


Diagram:

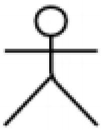


4.5.4 Sequence Diagram

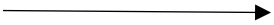
A sequence diagram is a type of interaction diagram because it describes how and in what order, a group of objects works together. Sequence diagrams are time focus and they show the order of the interaction visually by using the vertical axis of the diagram to represent time, what messages are sent and when. These are sometimes known as event diagrams or event scenarios.

Elements of Sequence Diagram:

Actor



Message



Lifeline

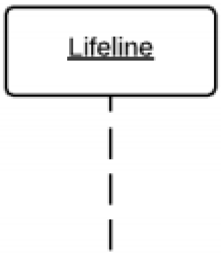
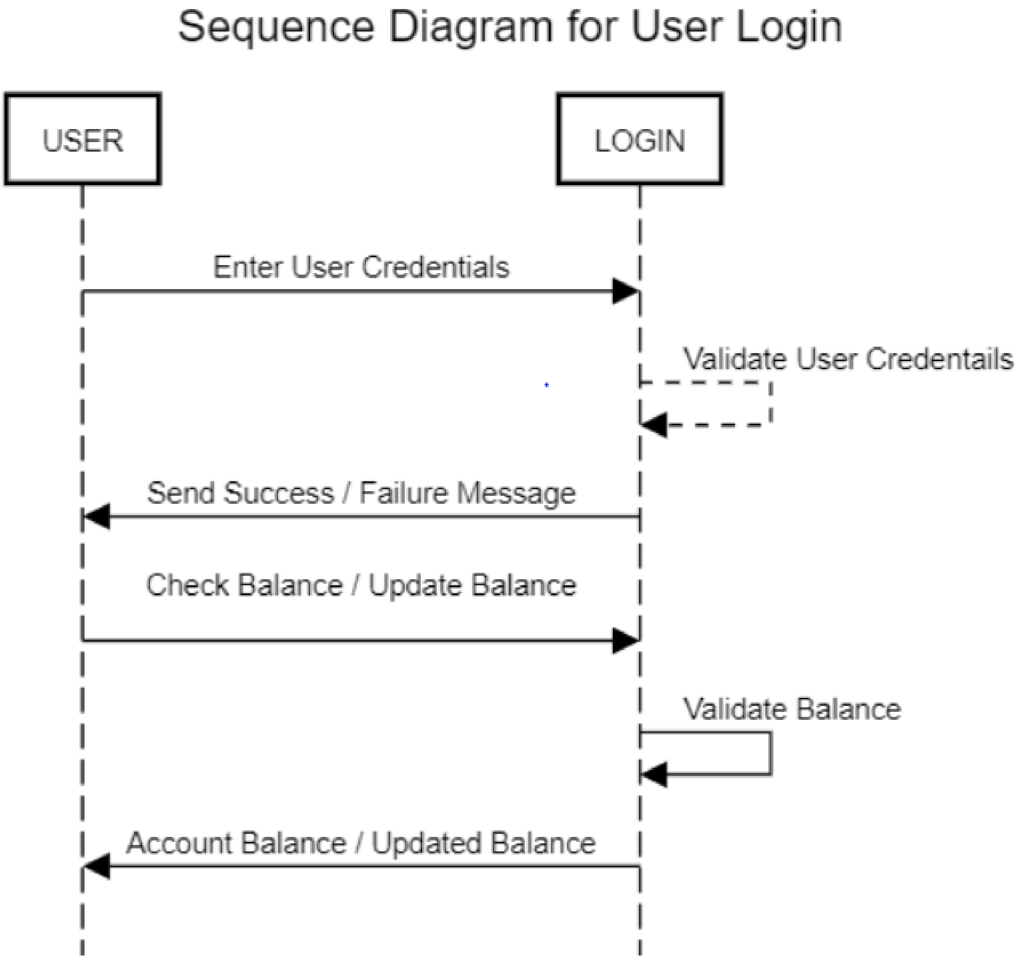
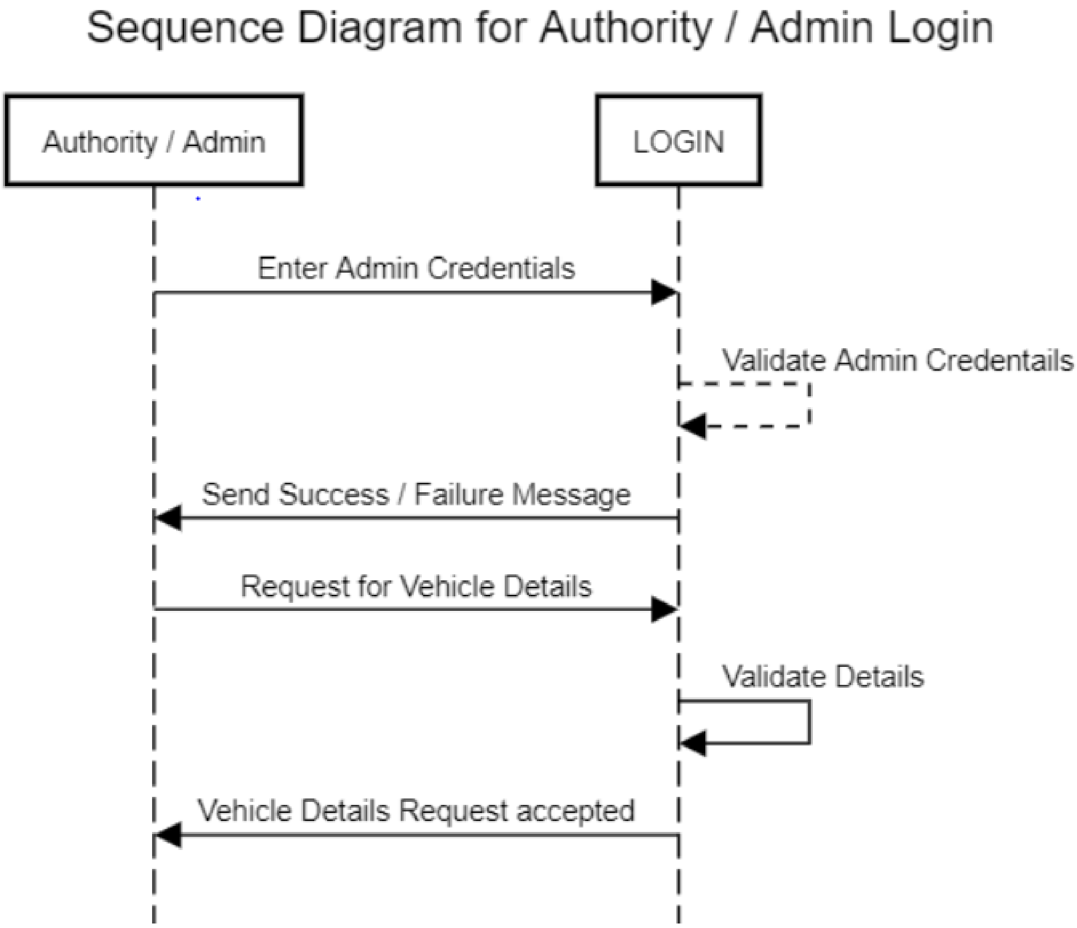


Diagram:

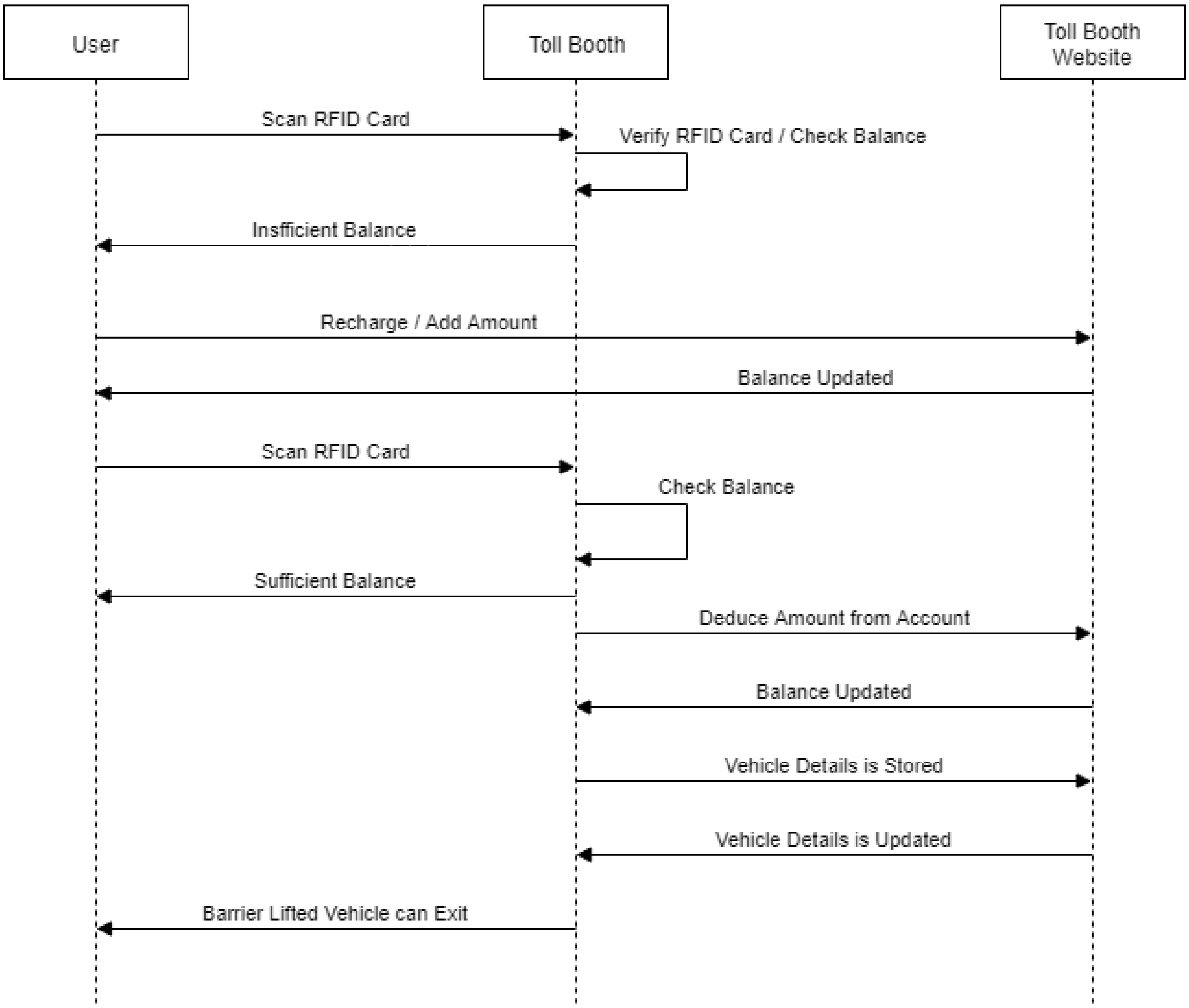
1. Sequence Diagram for User Login


Sequence Diagram for User Login
USER
LOGIN
Enter User Credentials
Validate User Credentails
Send Success / Failure Message
Check Balance / Update Balance
Validate Balance
Account Balance / Updated Balance

2. Sequence Diagram for Admin Login


Sequence Diagram for Authority / Admin Login
Authority / Admin
LOGIN
Enter Admin Credentials
Validate Admin Credentails
Send Success / Failure Message
Request for Vehicle Details
Validate Details
Vehicle Details Request accepted

3. Sequence Diagram for Toll Booth Management System



4.6 Test case Design

1. **Admin**

a. **Registration Page**

TEST CASE No.	CONDITION	EXPECTED OUTPUT	ACTUAL OUTPUT
1	If any of the field is empty	ERROR!!! Enter all the details!	ERROR!!! Enter all the details!
2	If Email-id already exists	ERROR!!! abcd@gmail.com already exists!	ERROR!!! abcd@gmail.com already exits!
3	If both the passwords do not match	ERROR!!! Both the passwords should match!	ERROR!!! Both the passwords should match!!!
4	If Date of Birth has not entered in correct format	ERROR!!! Invalid Date of Birth format! (DD-MM-YYYY)	ERROR!!! Invalid Date of Birth format! (DD-MM-YYYY)
5	If 10 digit mobile number is not entered	ERROR!!! Invalid mobile number!	ERROR!!! Invalid mobile number!
6	If all the details are correct	Registration Successful!	Registration Successful!

b. Login Page

TEST CASE No.	CONDITION	EXPECTED OUTPUT	ACTUAL OUTPUT
1	If any of the field is empty	ERROR!!! Enter Email-Id & password!!!	ERROR!!! Enter Email-Id & password!!!
2	If password is not entered	ERROR!!! Enter password	ERROR!!! Enter password
3	If Email-Id & password do not match	ERROR!!! Your Email-Id or Password is invalid!	ERROR!!! Your Email-Id or Password is invalid!
4	If all the details are correct	Login Successful!!! Home page	Login Successful!!! Home page

c. Update Profile

TEST CASE No.	CONDITION	EXPECTED OUTPUT	ACTUAL OUTPUT
1	If any of the field is empty	ERROR!!! Enter all the details	ERROR!!! Enter all the details
2	If username already exists	ERROR!!! abcd@gmail.com is available	ERROR!!! abcd@gmail.com is available
3	If 10 digit number is not entered	ERROR!!! Invalid mobile number	ERROR!!! Invalid mobile number
4	If Date of Birth has not entered in correct format	ERROR!!! Invalid Date of Birth format! (DD-MM-YYYY)	ERROR!!! Invalid Date of Birth format! (DD-MM-YYYY)
4	If none of the above error occurs	Update successfully!	Update successfully!

2. **User**

a. **Registration Page**

TEST CASE No.	CONDITION	EXPECTED OUTPUT	ACTUAL OUTPUT
1	If any of the field is empty	ERROR!!! Enter all the details!	ERROR!!! Enter all the details!
2	If Email-id already exists	ERROR!!! abcd@gmail.com already exists!	ERROR!!! abcd@gmail.com already exits!
3	If both the passwords do not match	ERROR!!! Both the passwords should match!	ERROR!!! Both the passwords should match!!!
4	If Date of Birth has not entered in correct format	ERROR!!! Invalid Date of Birth format! (DD-MM-YYYY)	ERROR!!! Invalid Date of Birth format! (DD-MM-YYYY)
5	If 10 digit mobile number is not entered correctly	ERROR!!! Invalid Phone number!	ERROR!!! Invalid Phone number!
6	If Vehicle Number has not entered in correct format	ERROR!!! Invalid Vehicle Number! (e.g.MH01CX9318)	ERROR!!! Invalid Vehicle Number! (e.g.MH01CX9318)
7	If Account Balance Field is Empty	ERROR!!! Account Balance Field cannot be empty	ERROR!!! Account Balance Field cannot be empty
8	If all the details are correct	Registration Successful!	Registration Successful!

b. Login form

TEST CASE No.	CONDITION	EXPECTED OUTPUT	ACTUAL OUTPUT
1	If any of the field is empty	ERROR!!! Enter Email-Id & password!!!	ERROR!!! Enter Email-Id & password!!!
2	If password is not entered	ERROR!!! Enter password	ERROR!!! Enter password
3	If Email-Id & password do not match	ERROR!!! Your Email-Id or Password is invalid!	ERROR!!! Your Email-Id or Password is invalid!
4	If all the details are correct	Login Successful!!! Home page	Login Successful!!! Home page

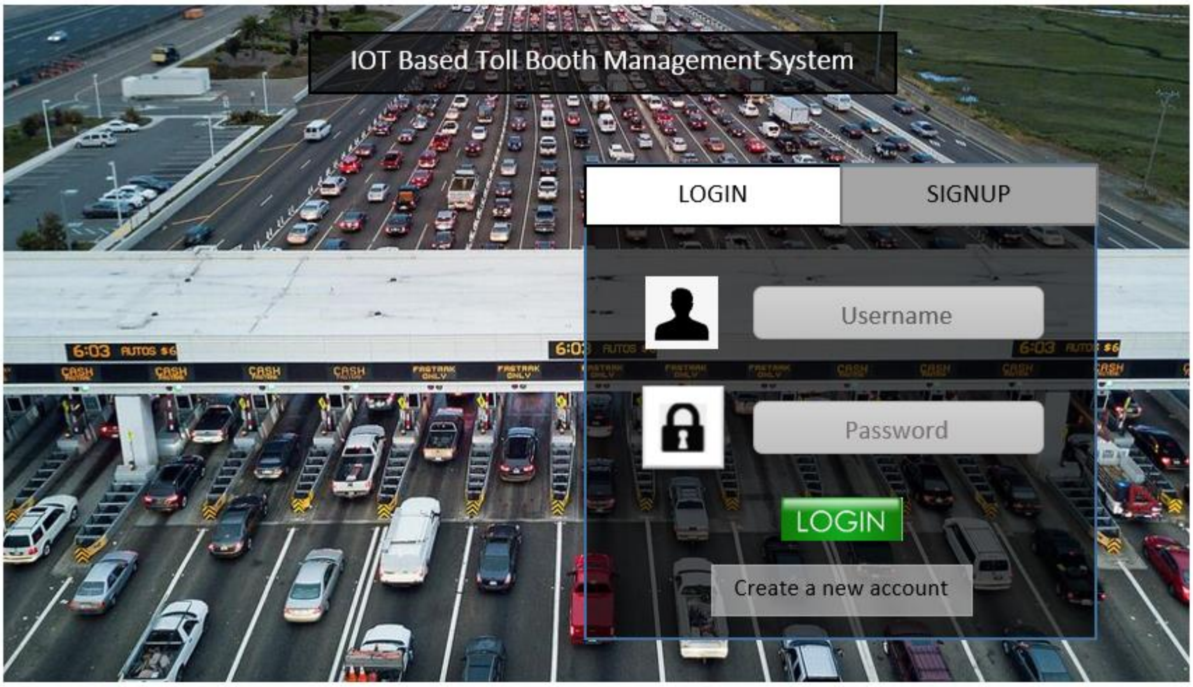
c. Update Profile

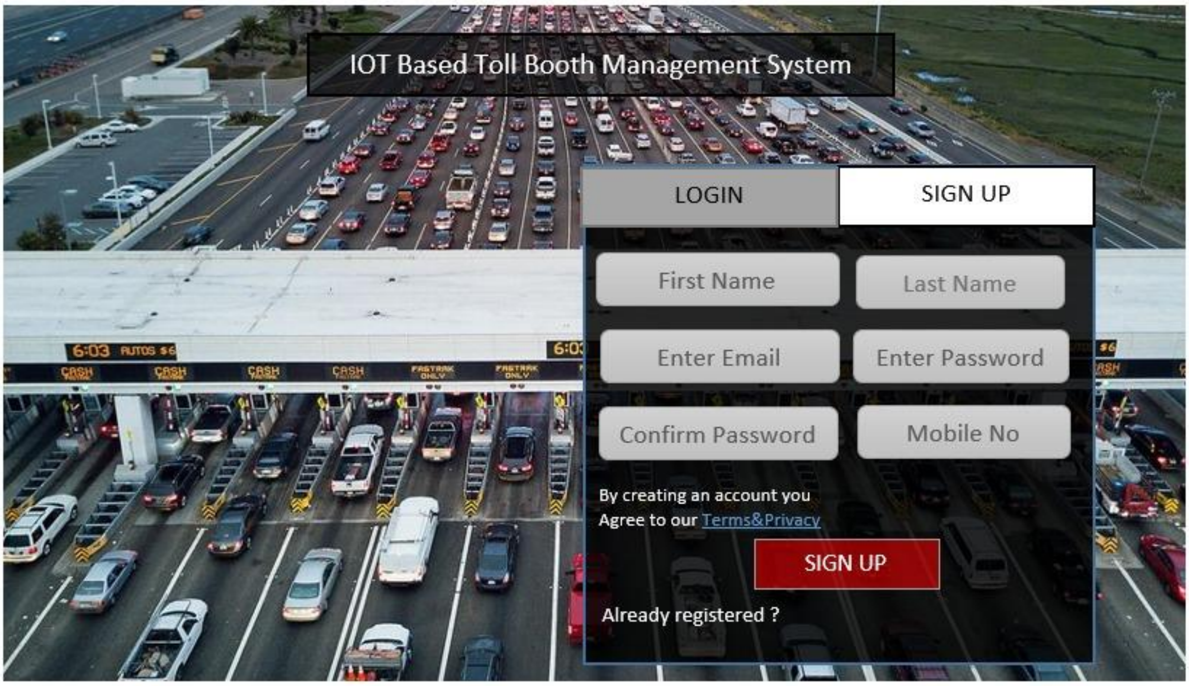
TEST CASE No.	CONDITION	EXPECTED OUTPUT	ACTUAL OUTPUT
1	If any of the field is empty	ERROR!!! Enter all the details!!!	ERROR!!! Enter all the details!!!
2	If date format is not entered properly	ERROR!!! Invalid date format!!! (DD-MM-YYYY)	ERROR!!! Invalid date format!!! (DD-MM-YYYY)
3	If 10 digit mobile number is not entered correctly	ERROR!!! Invalid Phone number!	ERROR!!! Invalid Phone number!
3	If both the passwords do not match	ERROR!!! Invalid password	ERROR!!! Invalid password
4	If vehicle number or format is not entered	ERROR!!! Invalid vehicle	ERROR!!! Invalid vehicle

	properly	number or format!!!	number or format!!!
5	If all the details are entered properly	Updated Successfully!!!	Updated Successfully!!!

4.7 User Interface

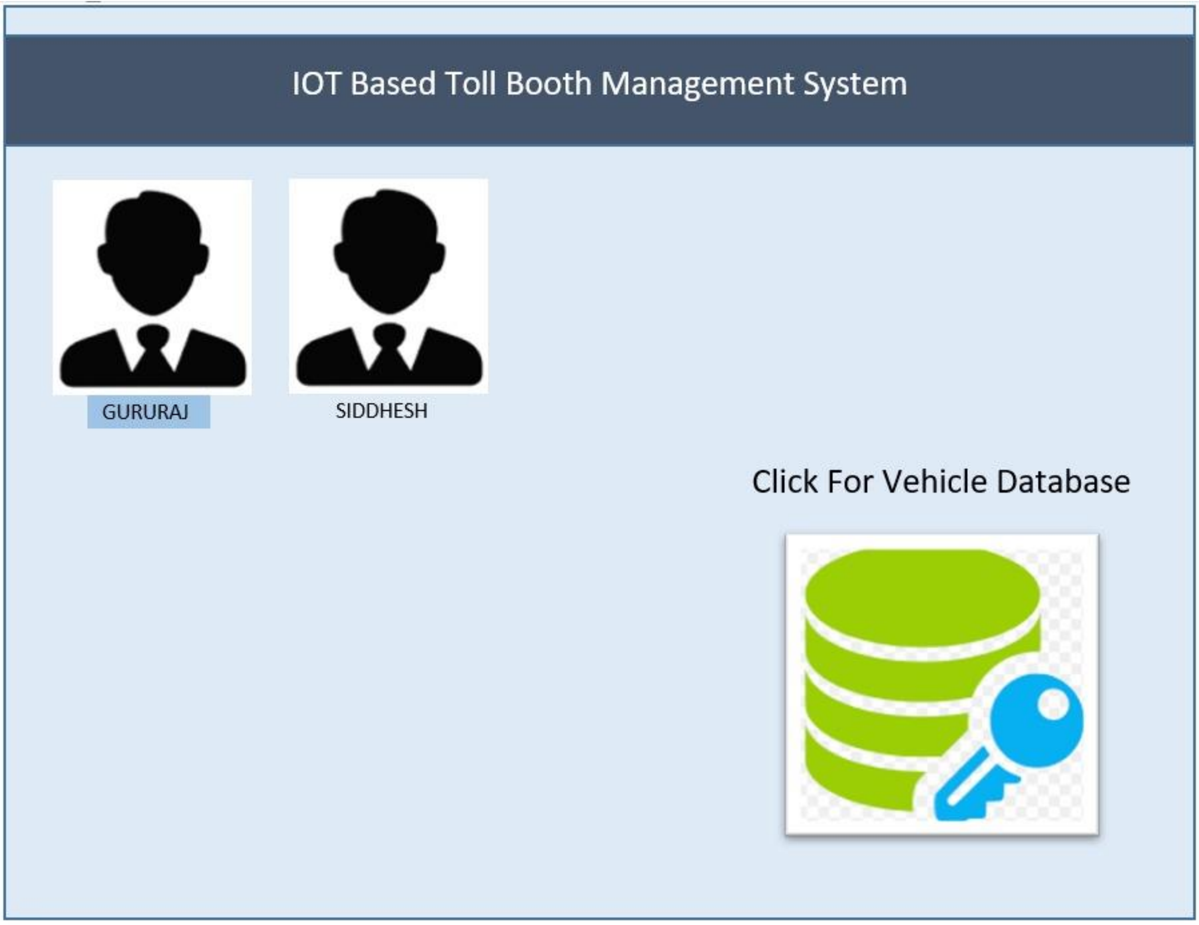
1. Website





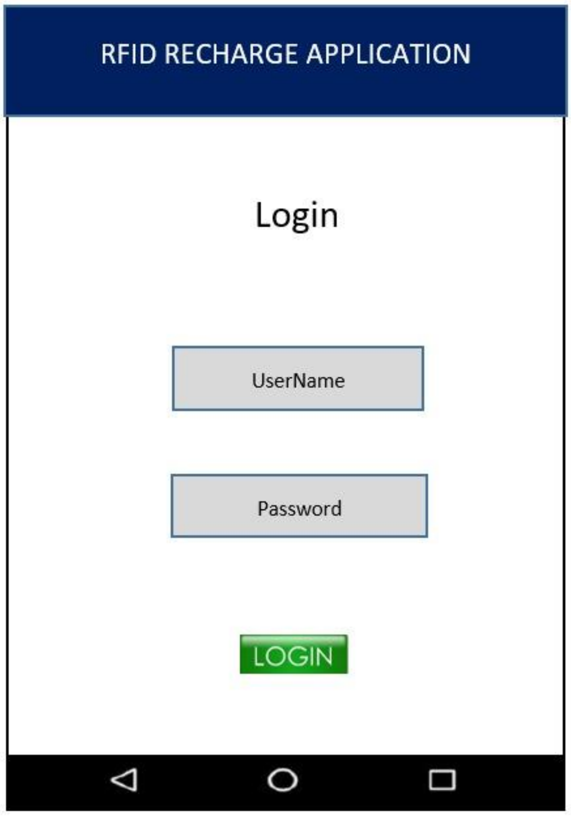
IOT Based Toll Booth Management System

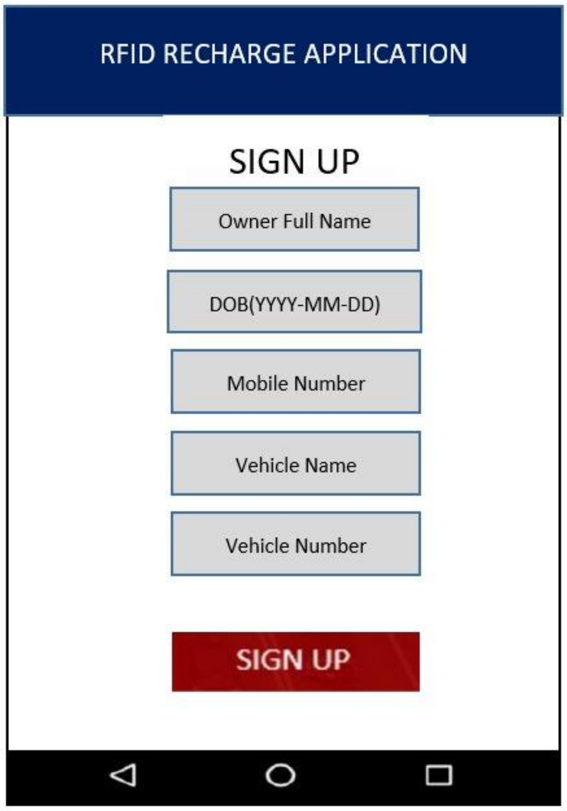
By creating an account you Agree to our Terms&Privacy

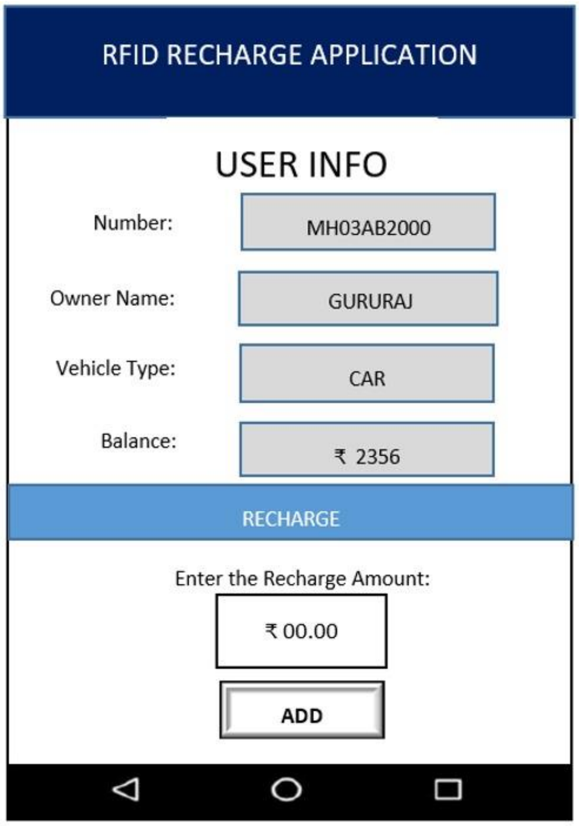


IOT Based Toll Booth Management System

2. Android


RFID RECHARGE APPLICATION
Login
UserName
Password
LOGIN


RFID RECHARGE APPLICATION
SIGN UP
Owner Full Name
DOB(YYYY-MM-DD)
Mobile Number
Vehicle Name
Vehicle Number
SIGN UP



USER INFO

Number: MH03AB2000

Owner Name: GURURAJ

Vehicle Type: CAR

Balance: ₹ 2356

Enter the Recharge Amount: ₹ 00.00

4.8 Security Issues

This project basically provides an easier, convenient and more secured way to the users i.e. drivers. Users don't need to wait at the Toll Booth for long time. Additionally it stores all the details of the vehicles such as vehicle type, vehicle arrival and departure time, etc. which can result into high security and surveillance. So basically here we are providing the user's confidential details to the Toll booth Authority. So there may occur some secure issues.

The main medium through which the Authority is being able to watch Vehicles details is through internet, if there is no internet connectivity in the Authority's device, it would be a failure. Also user requires Web Application to recharge/ update its card balance for successful transaction and it works through internet, so if there is no internet connectivity in the user's device, it would be a failure too. Also, if the user lost his/hers RFID card it will result into failure. That is, the user will not be able to access/ scan at the Toll Station and also there would be a chance of card misuse.

CHAPTER 5
IMPLEMENTATION AND TESTING

5.1 Implementation Approaches

The project IOT Based Toll Booth Management System is a huge and vast project. As it includes two different sides (User and Admin) working together. This project is a web application. So here we had to develop same modules in two different sides, so it was quite challenging to perform the codes.

We started with our project way before. As these concept of building codes were too new to us. We took almost one month to complete with just login and register page. As it was new to work in a combination of front end and back end with database connectivity. It was always that we wouldn’t complete our work before date. We simply knew that our project is vast and tried our level best to complete it before time. In initial stages of the project we had to learn a lot, we had to study many things before hand to start with the project. Once we were done with our research and study we divided the project into modules and started working on it according to the schedule.

5.2 Coding Details and Code Efficiency

USER

1. LOGIN PAGE (DESIGN):

```
<%@ Page Title="" Language="C#" MasterPageFile="~/MasterPage.master"
AutoEventWireup="true" CodeFile="LoginPage.aspx.cs" Inherits="LoginPage" %>

<asp:Content ID="Content1" ContentPlaceHolderID="head" Runat="Server">
    <style type="text/css">
        .auto-style3 {
            width: 100%;
        }
        .auto-style7 {
            text-align: center;
            height: 59px;
        }
        .auto-style14 {
            height: 56px;
        }
        .auto-style15 {
            width: 431px;
            text-align: center;
            height: 60px;
            font-weight: bold;
        }
        .auto-style16 {
            width: 383px;
            text-align: center;
            height: 60px;
        }
        .auto-style17 {
            height: 60px;
            width: 459px;
        }
        .auto-style18 {
            height: 60px;
        }
    </style>
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" Runat="Server">
    <table class="auto-style3" style="border-style: solid; background-color: #99CCFF; width: 1100px;">
        <tr>
            <td colspan="3" class="auto-style7"><strong style="font-size: x-large">USER LOGIN</strong></td>
        </tr>
        <tr>
            <td class="auto-style15">Email-Id:</td>
            <td class="auto-style16" style="color: #FF0000">
                *
                <asp:TextBox ID="TextBoxEmailtxt" runat="server" Height="30px" Width="250px" placeholder="Enter Registered Email-ID"></asp:TextBox>
            </td>
            <td class="auto-style17">
                <asp:RequiredFieldValidator ID="RequiredFieldValidator1" runat="server" ErrorMessage="* Email-Id field cannot be empty" ForeColor="Red" ControlToValidate="TextBoxEmailtxt"></asp:RequiredFieldValidator>
                <br />
                <asp:RegularExpressionValidator ID="RegularExpressionValidator1" runat="server" ErrorMessage="* Enter valid Email-Id" ForeColor="Red" ControlToValidate="TextBoxEmailtxt" ValidationExpression="\w+([-+.']\w+)*@\w+([-.]\w+)*\.\w+([-.]\w+)*"></asp:RegularExpressionValidator>
            </td>
        </tr>
        <tr>
            <td class="auto-style15">Password:</td>
            <td class="auto-style16" style="color: #FF0000">
```

```
                *
                <asp:TextBox ID="TextBoxPasswordtxt" runat="server" Height="30px" Width="250px"
TextMode="Password" placeholder="Enter Password"></asp:TextBox>
            </td>
            <td class="auto-style17">
                <asp:RequiredFieldValidator ID="RequiredFieldValidator2" runat="server"
ErrorMessage="* Password field cannot be empty" ForeColor="Red"
ControlToValidate="TextBoxPasswordtxt"></asp:RequiredFieldValidator>
            </td>
        </tr>
        <tr>
            <td colspan="3" align="center" class="auto-style14">
                <asp:Button ID="Button1" runat="server" Text="Login" Height="30px" Width="150px"
Font-Bold="True" OnClick="Button1_Click" />
            </td>
        </tr>
        <tr>
            <td colspan="3" align="center" class="auto-style18">
                <asp:HyperLink ID="HyperLink1" runat="server"
NavigateUrl="~/RegisterPage.aspx">Does not have an account, REGISTER Now!</asp:HyperLink>
            </td>
        </tr>
        <tr>
            <td colspan="3" align="center" class="auto-style2">
                <asp:Label ID="Label1" runat="server" ForeColor="Red" Height="30px"></asp:Label>
            </td>
        </tr>
    </table>
</asp:Content>
```

(CODE):

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data.SqlClient;

public partial class LoginPage : System.Web.UI.Page
{
    SqlConnection con = new SqlConnection("Data Source=LAPTOP-9NN9NDBK;Initial
Catalog=Project;Integrated Security=True");

    protected void Page_Load(object sender, EventArgs e)
    {

    }
    protected void Button1_Click(object sender, EventArgs e)
    {
        string check = "select count(*) from Userreg where Email = '" + TextBoxEmailtxt.Text + "' and
Password = '" + TextBoxPasswordtxt.Text + "' ";
        SqlCommand com = new SqlCommand(check, con);
        con.Open();
        int temp = Convert.ToInt32(com.ExecuteScalar().ToString());
        con.Close();
        if (temp == 1)
        {
            Session["Email"] = TextBoxEmailtxt.Text.Trim();
            Response.Redirect("UserHomeFinal.aspx");
        }
        else
        {
            Label1.Text = "Your Email-Id or Password is Invalid";
        }
    }
}
```

2. REGISTER PAGE (DESIGN):

```
<%@ Page Title="" Language="C#" MasterPageFile="~/MasterPage.master"
AutoEventWireup="true" CodeFile="DefautRegister.aspx.cs" Inherits="DefautRegister" %>

<asp:Content ID="Content1" ContentPlaceHolderID="head" Runat="Server">

   <style type="text/css">
      .auto-style7 {
         width: 100%;
         height: 569px;
      }
      .auto-style8 {
      }
      .auto-style9 {
      }
      .auto-style10 {
         text-align: center;
         font-size: x-large;
      }
      .auto-style11 {
         width: 306px;
         font-weight: bold;
         font-size: medium;
      }
      .auto-style12 {
         width: 312px;
         text-align: center;
      }
      .auto-style13 {
         width: 306px;
         font-size: medium;
      }
      .auto-style14 {
            width: 306px;
            font-size: medium;
            height: 35px;
        }
        .auto-style15 {
            width: 312px;
            text-align: center;
            height: 35px;
        }
        .auto-style16 {
            height: 35px;
        }
        .auto-style17 {
            height: 36px;
        }
    </style>

</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" Runat="Server">
    <table class="auto-style7" style="border-style: groove; border-width: thin; background-color:
#66FFFF">
        <tr>
            <td class="auto-style10" colspan="3"><strong>USER REGISTRATION</strong></td>
        </tr>
        <tr>
            <td class="auto-style13"><strong>Full Name:</strong></td>
            <td class="auto-style12" style="color: #FF0000">
                *
                <asp:TextBox ID="Name" runat="server" placeholder="Full Name" Width="220px"
Height="25px"></asp:TextBox>
            </td>
            <td>
                <asp:RequiredFieldValidator ID="RequiredFieldValidator1" runat="server"
ControlToValidate="Name" ErrorMessage="* Full Name Field cannot be Empty"
```

```
ForeColor="Red"></asp:RequiredFieldValidator>
      </td>
    </tr>
    <tr>
      <td class="auto-style13">
        <strong>Email-Id:</strong></td>
      <td class="auto-style12" style="color: #FF0000">
        *
        <asp:TextBox ID="Email" runat="server" placeholder="Email-Id" Width="220px"
Height="25px"></asp:TextBox>
      </td>
      <td>
        <asp:RequiredFieldValidator ID="RequiredFieldValidator2" runat="server"
ControlToValidate="Email" ErrorMessage="* Email-Id Field cannot be Empty"
ForeColor="Red"></asp:RequiredFieldValidator>
        <br />
        <asp:RegularExpressionValidator ID="RegularExpressionValidator1" runat="server"
ControlToValidate="Email" ErrorMessage="* Enter Valid Email-Id" ForeColor="Red"
ValidationExpression="\w+([-+.']\w+)*@\w+([-.]\w+)*\.\w+([-
.]\w+)*"></asp:RegularExpressionValidator>
      </td>
    </tr>
    <tr>
      <td class="auto-style13">
        <strong>Gender:</strong></td>
      <td class="auto-style12" style="color: #FF0000">
        *
        <asp:DropDownList ID="DropDownList1" runat="server" Height="25px">
          <asp:ListItem>Select Gender</asp:ListItem>
          <asp:ListItem>Male</asp:ListItem>
          <asp:ListItem>Female</asp:ListItem>
          <asp:ListItem>Other</asp:ListItem>
        </asp:DropDownList>
      </td>
```

```
            <td>
                <asp:RequiredFieldValidator ID="RequiredFieldValidator3" runat="server" ControlToValidate="DropDownList1" ErrorMessage="* Gender Field cannot be Empty" ForeColor="Red"></asp:RequiredFieldValidator>
            </td>
        </tr>
        <tr>
            <td class="auto-style14">
                <strong>Address:</strong></td>
            <td class="auto-style15" style="color: #FF0000">
                *
                <asp:TextBox ID="Address" runat="server" placeholder="Address" Width="220px" Height="25px"></asp:TextBox>
            </td>
            <td class="auto-style16">
                <asp:RequiredFieldValidator ID="RequiredFieldValidator4" runat="server" ControlToValidate="Address" ErrorMessage="* Address Field cannot be Empty" ForeColor="Red"></asp:RequiredFieldValidator>
            </td>
        </tr>
        <tr>
            <td class="auto-style13">
                <strong>Date of Birth:</strong></td>
            <td class="auto-style12" style="color: #FF0000">
                *
                <asp:TextBox ID="Dob" runat="server" placeholder="DD-MM-YYYY" MaxLength="10" Width="220px" Height="25px"></asp:TextBox>
            </td>
            <td>
                <asp:RequiredFieldValidator ID="RequiredFieldValidator5" runat="server" ControlToValidate="Dob" ErrorMessage="* Date of Birth Field cannot be Empty" ForeColor="Red"></asp:RequiredFieldValidator>
            </td>
        </tr>
```

```
    <tr>
        <td class="auto-style11">
            Phone Number:</td>
        <td class="auto-style12" style="color: #FF0000">
            *
            <asp:TextBox ID="Phone" runat="server" placeholder="Phone Number" MaxLength="10"
TextMode="Phone" Width="220px" Height="25px"></asp:TextBox>
        </td>
        <td>
            <asp:RequiredFieldValidator ID="RequiredFieldValidator6" runat="server"
ControlToValidate="Phone" ErrorMessage="* Phone Number Field cannot be Empty"
ForeColor="Red"></asp:RequiredFieldValidator>
        </td>
    </tr>
    <tr>
        <td class="auto-style11">
            Vehicle Number (Primary):</td>
        <td class="auto-style12" style="color: #FF0000">
            *
            <asp:TextBox ID="VP" runat="server" placeholder="Eg: AB00CD0000" MaxLength="10"
Width="220px" Height="25px"></asp:TextBox>
        </td>
        <td>
            <asp:RequiredFieldValidator ID="RequiredFieldValidator7" runat="server"
ControlToValidate="VP" ErrorMessage="* Vehicle Number (Primary) Field cannot be empty"
ForeColor="Red"></asp:RequiredFieldValidator>
        </td>
    </tr>
    <tr>
        <td class="auto-style11">
            Vehicle Number (Optional):</td>
        <td class="auto-style12" style="color: #FF0000">
            *
            <asp:TextBox ID="Vo" runat="server" placeholder="Eg: AB00CD0000" MaxLength="10"
```

```
Width="220px" Height="25px"></asp:TextBox>
      </td>
      <td>
        <asp:RequiredFieldValidator ID="RequiredFieldValidator8" runat="server"
ControlToValidate="Vo" ErrorMessage="* Vehicle Number (Optional) Field cannot be Empty"
ForeColor="Red"></asp:RequiredFieldValidator>
      </td>
    </tr>
    <tr>
      <td class="auto-style11">
        Password:</td>
      <td class="auto-style12" style="color: #FF0000">
        *
        <asp:TextBox ID="Password" runat="server" placeholder="Password"
TextMode="Password" Width="220px" Height="25px"></asp:TextBox>
      </td>
      <td>
        <asp:RequiredFieldValidator ID="RequiredFieldValidator9" runat="server"
ControlToValidate="Password" ErrorMessage="* Password Field cannot be Empty"
ForeColor="Red"></asp:RequiredFieldValidator>
      </td>
    </tr>
    <tr>
      <td class="auto-style11">
        Confirm Password:</td>
      <td class="auto-style12" style="color: #FF0000">
        *
        <asp:TextBox ID="Cp" runat="server" placeholder="Confirm Password"
TextMode="Password" Width="220px" Height="25px"></asp:TextBox>
      </td>
      <td>
        <asp:RequiredFieldValidator ID="RequiredFieldValidator10" runat="server"
ControlToValidate="Cp" ErrorMessage="* Confirm Password Field cannot be Empty"
ForeColor="Red"></asp:RequiredFieldValidator>
```

```
        <br />
        <asp:CompareValidator ID="CompareValidator1" runat="server" ControlToCompare="Password" ControlToValidate="Cp" ErrorMessage="* Password and Confirm Password must be same" ForeColor="Red"></asp:CompareValidator>
      </td>
    </tr>
    <tr>
      <td class="auto-style11">
        Account Balance:</td>
      <td class="auto-style12" style="color: #FF0000">
        *
        <asp:TextBox ID="Balance" runat="server" TextMode="Number" placeholder="Account Balance" MaxLength="5" Width="220px" Height="25px"></asp:TextBox>
      </td>
      <td>
        <asp:RequiredFieldValidator ID="RequiredFieldValidator11" runat="server" ControlToValidate="Balance" ErrorMessage="* Account Balance Field cannot be Empty. Minimum Balance should be Rs. 500/-" ForeColor="Red"></asp:RequiredFieldValidator>
      </td>
    </tr>
    <tr>
      <td class="auto-style9" colspan="3" align="center">
        <asp:Button ID="Button1" runat="server" OnClick="Button1_Click" Text="Register" Height="38px" style="font-weight: 700" Width="126px" />
      </td>
    </tr>
    <tr>
      <td class="auto-style17" colspan="3" align="center">
        <asp:Label ID="Label1" runat="server" style="font-weight: 700">Status</asp:Label>
      </td>
    </tr>
    <tr>
      <td class="auto-style8" colspan="3" align="center">
        <asp:HyperLink ID="HyperLink1" runat="server" style="text-align: center"
```

```
NavigateUrl="~/LoginPage.aspx">Already have an Account, LOGIN Now!!</asp:HyperLink>
            </td>
        </tr>
    </table>
</asp:Content>
```

(CODE):

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data.SqlClient;

public partial class DefautRegister : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        if (IsPostBack)
        {
            SqlConnection con = new SqlConnection("Data Source=LAPTOP-9NN9NDBK;Initial Catalog=Project;Integrated Security=True");
            con.Open();
            string checkuser = "select count(*) from Userreg where Email = '" + Email.Text + "' ";
            SqlCommand cmd = new SqlCommand(checkuser, con);
            int temp = Convert.ToInt32(cmd.ExecuteScalar().ToString());
            if (temp == 1)
            {
                Label1.Text = ("User Already Exists with the current Email-Id. Please use different Email-ID.");
            }
            con.Close();
        }
    }
    protected void Button1_Click(object sender, EventArgs e)
    {
        try
        {
            SqlConnection con = new SqlConnection("Data Source=LAPTOP-9NN9NDBK;Initial
Catalog=Project;Integrated Security=True");
            con.Open();
            SqlCommand cmd = new SqlCommand("insert into Userreg" +
"(Name,Email,Address,Date,Phone,Vehicle1,Vehicle2,Password,Balance,Gender) values
(@Name,@Email,@Address,@Date,@Phone,@Vehicle1,@Vehicle2,@Password,@Balance,@Gen
der)", con);
            cmd.Parameters.AddWithValue("@Name", Name.Text);
            cmd.Parameters.AddWithValue("@Email", Email.Text);
            cmd.Parameters.AddWithValue("@Address", Address.Text);
            cmd.Parameters.AddWithValue("@Date", Dob.Text);
            cmd.Parameters.AddWithValue("@Phone", Phone.Text);
            cmd.Parameters.AddWithValue("@Vehicle1", VP.Text);
            cmd.Parameters.AddWithValue("@Vehicle2", Vo.Text);
            cmd.Parameters.AddWithValue("@Password", Password.Text);
            cmd.Parameters.AddWithValue("@Balance", Balance.Text);
            cmd.Parameters.AddWithValue("@Gender", DropDownList1.SelectedItem.Value);
            cmd.ExecuteNonQuery();
            con.Close();
            Response.Redirect("LoginPage.aspx");
        }
        catch (Exception ex)
        {
            Label1.Text = "User Already Exists with the current Email-Id. Please use different Email-ID.";
        }
    }
}
```

3. EDIT PROFILE:

```
<asp:DetailsView ID="DetailsView1" runat="server" AutoGenerateRows="False" DataKeyNames="Email" DataSourceID="SqlDataSource1" Height="400px" Width="700px" BackColor="#E9F5F3" align="center" BorderColor="Black" BorderStyle="Solid">
        <EditRowStyle BackColor="#E7F6F3" BorderColor="Black" />
        <EmptyDataRowStyle BackColor="#E6F6F3" />
        <Fields>
          <asp:TemplateField HeaderText=" Full Name" SortExpression="Name">
            <EditItemTemplate>
              <asp:TextBox ID="TextBox1" runat="server" Text='<%# Bind("Name") %>'></asp:TextBox>
              <asp:RequiredFieldValidator ID="RequiredFieldValidator1" runat="server" ControlToValidate="TextBox1" ErrorMessage="* Full Name field cannot be empty" ForeColor="Red"></asp:RequiredFieldValidator>
            </EditItemTemplate>
            <InsertItemTemplate>
              <asp:TextBox ID="TextBox1" runat="server" Text='<%# Bind("Name") %>'></asp:TextBox>
            </InsertItemTemplate>
            <ItemTemplate>
              <asp:Label ID="Label1" runat="server" Text='<%# Bind("Name") %>'></asp:Label>
            </ItemTemplate>
          </asp:TemplateField>
          <asp:TemplateField HeaderText="Email-Id" SortExpression="Email">
            <EditItemTemplate>
              <asp:Label ID="Label1" runat="server" Text='<%# Eval("Email") %>'></asp:Label>
            </EditItemTemplate>
            <InsertItemTemplate>
              <asp:TextBox ID="TextBox2" runat="server" Text='<%# Bind("Email") %>'></asp:TextBox>
            </InsertItemTemplate>
            <ItemTemplate>
              <asp:Label ID="Label2" runat="server" Text='<%# Bind("Email") %>'></asp:Label>
            </ItemTemplate>
          </asp:TemplateField>
          <asp:TemplateField HeaderText="Gender" SortExpression="Gender">
            <EditItemTemplate>
              <asp:TextBox ID="TextBox2" runat="server" Text='<%# Bind("Gender")
  %>'></asp:TextBox>
              <asp:RequiredFieldValidator ID="RequiredFieldValidator2" runat="server"
  ControlToValidate="TextBox2" ErrorMessage="* Gender field cannot be empty"
  ForeColor="Red"></asp:RequiredFieldValidator>
            </EditItemTemplate>
            <InsertItemTemplate>
              <asp:TextBox ID="TextBox3" runat="server" Text='<%# Bind("Gender")
  %>'></asp:TextBox>
            </InsertItemTemplate>
            <ItemTemplate>
              <asp:Label ID="Label3" runat="server" Text='<%# Bind("Gender")
  %>'></asp:Label>
            </ItemTemplate>
          </asp:TemplateField>
          <asp:TemplateField HeaderText="Address" SortExpression="Address">
            <EditItemTemplate>
              <asp:TextBox ID="TextBox3" runat="server" Text='<%# Bind("Address")
  %>'></asp:TextBox>
              <asp:RequiredFieldValidator ID="RequiredFieldValidator3" runat="server"
  ControlToValidate="TextBox3" ErrorMessage="* Address field cannot be empty"
  ForeColor="Red"></asp:RequiredFieldValidator>
            </EditItemTemplate>
            <InsertItemTemplate>
              <asp:TextBox ID="TextBox4" runat="server" Text='<%# Bind("Address")
  %>'></asp:TextBox>
            </InsertItemTemplate>
            <ItemTemplate>
              <asp:Label ID="Label4" runat="server" Text='<%# Bind("Address")
  %>'></asp:Label>
            </ItemTemplate>
          </asp:TemplateField>
          <asp:TemplateField HeaderText="Date of Birth" SortExpression="Date">
            <EditItemTemplate>
              <asp:TextBox ID="TextBox4" runat="server" Text='<%# Bind("Date")
  %>'></asp:TextBox>
              <asp:RequiredFieldValidator ID="RequiredFieldValidator4" runat="server"
  ControlToValidate="TextBox4" ErrorMessage="* Date of Birth field cannot be empty"
  ForeColor="Red"></asp:RequiredFieldValidator>
            </EditItemTemplate>
            <InsertItemTemplate>
              <asp:TextBox ID="TextBox5" runat="server" Text='<%# Bind("Date")
  %>'></asp:TextBox>
            </InsertItemTemplate>
            <ItemTemplate>
              <asp:Label ID="Label5" runat="server" Text='<%# Bind("Date") %>'></asp:Label>
            </ItemTemplate>
          </asp:TemplateField>
          <asp:TemplateField HeaderText="Phone Number" SortExpression="Phone">
            <EditItemTemplate>
              <asp:TextBox ID="TextBox5" runat="server" Text='<%# Bind("Phone") %>'
  MaxLength="10" TextMode="Phone"></asp:TextBox>
              <asp:RequiredFieldValidator ID="RequiredFieldValidator5" runat="server"
  ControlToValidate="TextBox5" ErrorMessage="* Phone Number field cannot be empty"
  ForeColor="Red"></asp:RequiredFieldValidator>
            </EditItemTemplate>
            <InsertItemTemplate>
              <asp:TextBox ID="TextBox6" runat="server" Text='<%# Bind("Phone")
  %>'></asp:TextBox>
            </InsertItemTemplate>
            <ItemTemplate>
              <asp:Label ID="Label6" runat="server" Text='<%# Bind("Phone") %>'></asp:Label>
            </ItemTemplate>
          </asp:TemplateField>
          <asp:TemplateField HeaderText="Vehicle (Primary)" SortExpression="Vehicle1">
```

```
          <EditItemTemplate>
            <asp:TextBox ID="TextBox6" runat="server" Text='<%# Bind("Vehicle1") %>'
MaxLength="10"></asp:TextBox>
            <asp:RequiredFieldValidator ID="RequiredFieldValidator6" runat="server"
ControlToValidate="TextBox6" ErrorMessage="* Vehicle (Primary) field cannot be empty"
ForeColor="Red"></asp:RequiredFieldValidator>
          </EditItemTemplate>
          <InsertItemTemplate>
            <asp:TextBox ID="TextBox7" runat="server" Text='<%# Bind("Vehicle1")
%>'></asp:TextBox>
          </InsertItemTemplate>
          <ItemTemplate>
            <asp:Label ID="Label7" runat="server" Text='<%# Bind("Vehicle1")
%>'></asp:Label>
          </ItemTemplate>
        </asp:TemplateField>
        <asp:TemplateField HeaderText="Vehicle (Optional)" SortExpression="Vehicle2">
          <EditItemTemplate>
            <asp:TextBox ID="TextBox7" runat="server" Text='<%# Bind("Vehicle2") %>'
MaxLength="10"></asp:TextBox>
            <asp:RequiredFieldValidator ID="RequiredFieldValidator7" runat="server"
ControlToValidate="TextBox7" ErrorMessage="* Vehicle (Optional) Field cannot be empty"
ForeColor="Red"></asp:RequiredFieldValidator>
          </EditItemTemplate>
          <InsertItemTemplate>
            <asp:TextBox ID="TextBox8" runat="server" Text='<%# Bind("Vehicle2")
%>'></asp:TextBox>
          </InsertItemTemplate>
          <ItemTemplate>
            <asp:Label ID="Label8" runat="server" Text='<%# Bind("Vehicle2")
%>'></asp:Label>
          </ItemTemplate>
        </asp:TemplateField>
        <asp:TemplateField HeaderText="Password" SortExpression="Password">
```

```
          <EditItemTemplate>
            <asp:TextBox ID="TextBox8" runat="server" Text='<%# Bind("Password") %>'></asp:TextBox>
            <asp:RequiredFieldValidator ID="RequiredFieldValidator8" runat="server" ControlToValidate="TextBox8" ErrorMessage="* Password field cannot be empty" ForeColor="Red"></asp:RequiredFieldValidator>
          </EditItemTemplate>
          <InsertItemTemplate>
            <asp:TextBox ID="TextBox9" runat="server" Text='<%# Bind("Password") %>'></asp:TextBox>
          </InsertItemTemplate>
          <ItemTemplate>
            <asp:Label ID="Label9" runat="server" Text='<%# Bind("Password") %>'></asp:Label>
          </ItemTemplate>
        </asp:TemplateField>
        <asp:CommandField ShowEditButton="True" />
      </Fields>
      <InsertRowStyle BackColor="#E5F5F4" />
    </asp:DetailsView>
    <br />
    <asp:SqlDataSource ID="SqlDataSource1" runat="server" ConnectionString="<%$ ConnectionStrings:ProjectConnectionString %>" DeleteCommand="DELETE FROM [Userreg] WHERE [Email] = @Email" InsertCommand="INSERT INTO [Userreg] ([Name], [Email], [Gender], [Address], [Date], [Phone], [Vehicle1], [Vehicle2], [Password]) VALUES (@Name, @Email, @Gender, @Address, @Date, @Phone, @Vehicle1, @Vehicle2, @Password)" SelectCommand="SELECT * FROM [Userreg] WHERE ([Email] = @Email)" UpdateCommand="UPDATE [Userreg] SET [Name] = @Name, [Gender] = @Gender, [Address] = @Address, [Date] = @Date, [Phone] = @Phone, [Vehicle1] = @Vehicle1, [Vehicle2] = @Vehicle2, [Password] = @Password WHERE [Email] = @Email">
      <DeleteParameters>
        <asp:Parameter Name="Email" Type="String" />
      </DeleteParameters>
      <InsertParameters>
```

```
        <asp:Parameter Name="Name" Type="String" />
        <asp:Parameter Name="Email" Type="String" />
        <asp:Parameter Name="Gender" Type="String" />
        <asp:Parameter Name="Address" Type="String" />
        <asp:Parameter Name="Date" Type="String" />
        <asp:Parameter Name="Phone" Type="String" />
        <asp:Parameter Name="Vehicle1" Type="String" />
        <asp:Parameter Name="Vehicle2" Type="String" />
        <asp:Parameter Name="Password" Type="String" />
    </InsertParameters>
    <SelectParameters>
        <asp:SessionParameter Name="Email" SessionField="Email" Type="String" />
    </SelectParameters>
    <UpdateParameters>
        <asp:Parameter Name="Name" Type="String" />
        <asp:Parameter Name="Gender" Type="String" />
        <asp:Parameter Name="Address" Type="String" />
        <asp:Parameter Name="Date" Type="String" />
        <asp:Parameter Name="Phone" Type="String" />
        <asp:Parameter Name="Vehicle1" Type="String" />
        <asp:Parameter Name="Vehicle2" Type="String" />
        <asp:Parameter Name="Password" Type="String" />
        <asp:Parameter Name="Email" Type="String" />
    </UpdateParameters>
</asp:SqlDataSource>
```

5.3 Testing Approach

5.3.1 Unit Testing

Unit testing is a software development process in which smallest unit of programs are individually and independently tested to determine whether any issues occur. The main benefit of Unit Testing are, it reduces defects in the early stage and also improves the design The objective of Unit Testing is to verify the functional correctness of the each unit. A unit may be an individual function or procedure. It is important to test every units/modules of the application to build the project correctly.

In this project, we have tested each unit separately like Registration, Home page, Logout, etc. These parts are tested individually and independently. While testing each of these modules, it became easy to correct the mistakes and we have traced the errors easily. By doing this, we have learnt how to write a code while implementing it.

5.3.2 Integrated Testing

Integration Testing is a level of software testing where individual units/modules are combined and tested as a group. The purpose of the Integration testing is to verify the functional, procedure and reliability between the modules that are combined. Once all the units have been tested, integration testing is performed.

Integration Testing was performed while combining two or more modules and testing them. Like for example, Login part, Login page is integrated with registration part. As Login is incomplete without Registration, so it is combined and tested at once. And also, as our project is a mixture if two different activities which we have integrated into single website. For example, when the User registers himself, his/her data will get stored into the database and Admin can access the user details when he/she does the successful transaction i.e. pays the Toll Bill.

When the user updates the balance the RFID card. The amount in the card should get updated too. Also whenever the User does successful transaction the amount should be deducted. So, like this integration testing is performed many times while developing this project. It became easy to trace the errors and also it became easy to check the functionality and reliability of the project.

5.3.3 Beta Testing

Beta Testing is considered as last stage of testing in the development process. It is also known as User Acceptance testing or End user testing where the product is tested by the real users in a real environment. It shows the usability, compatibility, functionality and reliability of the product.

It also discovers the issues and flaws from the user’s perspective that you would not like to have in the final product. The whole experience of the users are forwarded to the developers who can make final changes as per the users point before releasing the final software.

For this project, we ourselves tried to register multiple users and admin to check whether the data is getting stored successfully. So, we have done Beta testing in one of our college device for testing its functionality and for checking whether it is meeting its requirement or not. Also, we were trying to edit the user's and admin's details to check whether the change is observed. Additionally we also checked whether the balance updating takes place or not. It was working quite perfectly. Also when the user does successful transaction the details of the Vehicle along with the Driver gets stored into the database. So, the Admin was able to check all the details which would result into adequate security and surveillance of the Toll Booth.

The Driver (User) was happy on this product. There were also some flaws in this project. That is, if there is no internet connection in the user’s device, then it is not possible to recharge the Wallet (RFID). And also we cannot store the details in the database without internet connection.

5.3.4 System Testing

System Testing is a testing where an integrated application/system is tested as a whole. It is also known as Black box Testing. Here, all the integrated modules of the system is combined completely and then tested. The purpose of the system testing is to evaluate the compliance of entire system with respect to specified requirements. It helps in approving the functional, technical and non-functional requirements of the client/business

5.4 Modifications and Improvements

Every product or project always needs some or other improvement in every aging time, so do this project needs. Here in our project in the Toll Booth Manager (Admin) only have the ability to see the database which consist of Vehicle details. He/ She cannot block the vehicle at that same instance of time. But we can add more features like Blocking and complaint regarding that vehicle can be filed at the nearest police station. And we can also add feature like Loan facility for the Driver (User) as in case if the user doesn’t have enough balance in the RFID he/ she will be provided loan amount which he/she will have to pay within 7 days.

And here in Toll Booth Management System we have included two modules that is Admin which will do Toll Booth Supervision and User who will recharge the Toll amount. But we can add more features like Call Supervision, Message Supervision, also we can let the driver inform the Toll Station if they encountered any accidents in their journey, and we can also include a section to report any issue to Police or Hospital if the Driver encounter any. These all new features can improve the product in different level.

5.5 Test Cases

1. Admin

a. Registration Page

TEST CASE No.	CONDITION	ACTUAL OUTPUT
1	Full name: Email-Id: Gender: Date of Birth: Phone Number: Password: Confirm Password:	ERROR!!! Enter all the details!
2	Full name: abc Email-Id: abc@gmail Gender: Male Date of Birth: 08-10-2000 Phone Number: 1234567890 Password: *** Confirm Password: ***	ERROR!!! Enter Valid Email-Id!
3	Full name: abc Email-Id: abcd@gmail.com Gender: Male Date of Birth: 08-10-2000 Phone Number: 1234567890	ERROR!!! Email-Id already exists!

	Password: *** Confirm Password: ***	
4	Full name: abc Email-Id: abc@gmail.com Gender: Male Date of Birth: 08102000 Phone Number: 1234567890 Password: *** Confirm Password: ***	ERROR!!! Invalid Date of Birth Format! (DD-MM-YYYY)
5	Full name: abc Email-Id: abc@gmail.com Gender: Male Date of Birth: 08-10-2000 Phone Number: 12345678 Password: *** Confirm Password: ***	ERROR!!! Invalid Phone Number!
6	Full name: abc Email-Id: abc@gmail.com Gender: Male Date of Birth: 08-10-2000 Phone Number: 1234567890 Password: **** Confirm Password: ***	ERROR!!! Password and Confirm Password must be same!

7	Full name: abc Email-Id: abc@gmail.com Gender: Male Date of Birth: 08-10-2000 Phone Number: 1234567890 Password: *** Confirm Password: ***	Register Successful!

b. Login Page

TEST CASE No.	CONDITION	ACTUAL OUTPUT
1	Email-Id : Password :	Enter Email-Id & Password!!!
2	Username : abc@gmail.com Password :	Password field cannot be empty!!!
3	Username : abc@gmail.com Password : *****	Your Email-Id or Password is Invalid!!!
4	Username : abc@gmail.com Password : ******	Login Successful!!!

c. Update Profile

TEST CASE No.	CONDITION	ACTUAL OUTPUT
1	Full name: Email-Id: Gender: Date of Birth: Phone Number: Password: Confirm Password:	ERROR!!! Enter all the details!
2	Full name: abc Email-Id: abc@gmail.com Gender: Male Date of Birth: 08102000 Phone Number: 1234567890 Password: *** Confirm Password: ***	ERROR!!! Invalid Date of Birth Format! (DD-MM-YYYY)
3	Full name: abc Email-Id: abc@gmail.com Gender: Male Date of Birth: 08-10-2000 Phone Number: 12345678 Password: *** Confirm Password: ***	ERROR!!! Invalid Phone Number!
4	Full name: abc Email-Id: abc@gmail.com Gender: Male Date of Birth: 08-10-2000 Phone Number: 1234567890 Password: ****	ERROR!!! Password and Confirm Password

	Confirm Password: ***	must be same!
5	Full name: abc Email-Id: abc@gmail.com Gender: Male Date of Birth: 08-10-2000 Phone Number: 1234567890 Password: *** Confirm Password: ***	Update Successful!

2. User

a. Registration Page

TEST CASE NO.	CONDITION	ACTUAL OUTPUT
1	Full Name: Email-Id: Gender: Address: Date of Birth: Phone Number: Vehicle Details(Primary: Vehicle Details(Optional): Password: Confirm Password: Account Balance:	ERROR!!! Enter all thedetails!
2	Full Name: abc Email-Id: xyz@gmail Gender: Male	ERROR!!! Invalid Email-Id!

	Address: Wadala Date of Birth: 08-06-2000 Phone Number: 1234567890 Vehicle Details (Primary): MH03CY8333 Vehicle Details (Optional): MH01TY1111 Password: ***** Confirm Password: ***** Account Balance: 2000	
3	Full Name: abc Email-Id: xyz@gmail.com Gender: Male Address: Wadala Date of Birth: 08-06-2000 Phone Number: 1234567890 Vehicle Details (Primary): MH03CY8333 Vehicle Details (Optional): MH01TY1111 Password: ***** Confirm Password: ***** Account Balance: 2000	ERROR!!! Email-Id already exist!
4	Full Name: abc Email-Id: xyz@gmail.com Gender: Male Address: Wadala Date of Birth: 08062000 Phone Number: 1234567890 Vehicle Details (Primary): MH03CY8333 Vehicle Details (Optional): MH01TY1111 Password: ***** Confirm Password: ***** Account Balance: 2000	ERROR!!! Invalid Date of Birth Format! (DD-MM-YYYY)
5	Full Name: abc Email-Id: xyz@gmail.com Gender: Male	ERROR!!! Invalid Vehicle Details (Primary)!

	Address: Wadala Date of Birth: 08-06-2000 Phone Number: 1234567890 Vehicle Details (Primary): MH03678333 Vehicle Details (Optional): MH01TY1111 Password: ***** Confirm Password: ***** Account Balance: 2000	(e.g.MH01CX9318)
6	Full Name: abc Email-Id: xyz@gmail.com Gender: Male Address: Wadala Date of Birth: 08-06-2000 Phone Number: 1234567890 Vehicle Details (Primary): MH03CY8333 Vehicle Details (Optional): MH01TY1111 Password: ***** Confirm Password: *** Account Balance: 2000	ERROR!!! Password and Confirm Password must be same!
7	Full Name: abc Email-Id: xyz@gmail.com Gender: Male Address: Wadala Date of Birth: 08-06-2000 Phone Number: 12345678 Vehicle Details (Primary): MH03CY8333 Vehicle Details (Optional): MH01TY1111 Password: ***** Confirm Password: ***** Account Balance: 2000	ERROR!!! Invalid Phone Number!!!
8	Full Name: abc Email-Id: xyz@gmail.com Gender: Male	ERROR!!! Account Balance Field cannot be

	Address: Wadala Date of Birth: 08-06-2000 Phone Number: 1234567890 Vehicle Details (Primary): MH03CY8333 Vehicle Details (Optional): MH01TY1111 Password: ***** Confirm Password: ***** Account Balance:	empty!
9	Full Name: abc Email-Id: xyz@gmail.com Gender: Male Address: Wadala Date of Birth: 08-06-2000 Phone Number: 1234567890 Vehicle Details (Primary): MH03CY8333 Vehicle Details (Optional): MH01TY1111 Password: ***** Confirm Password: ***** Account Balance:2000	Registration Successful!

b. Login Page

TEST CASE NO.	CONDITION	ACTUAL OUTPUT
1	Email-Id: Password:	ERROR!!! Enter Email-Id & Password!
2	Username: abc@gmail.com Password:	ERROR!!! Enter Password!
3	Username: abc@gmail.com Password: *****	ERROR!!! Your Email-

		Id or Password is Invalid!
4	Username: abc@gmail.com Password: ******	Login Successful!

c. Update Profile

TEST CASE NO.	CONDITION	ACTUAL OUTPUT
1	Full Name: Email-Id: Gender: Address: Date of Birth: Phone Number: Vehicle Details(Primary: Vehicle Details(Optional): Password: Confirm Password:	ERROR!!! Enter all the details!
2	Full Name: abc Email-Id: xyz@gmail.com Gender: Male Address: Wadala Date of Birth: 08062000 Phone Number: 1234567890 Vehicle Details (Primary): MH03CY8333 Vehicle Details (Optional): MH01TY1111 Password: ***** Confirm Password: *****	ERROR!!! Invalid Date of Birth Format! (DD-MM-YYYY)
3	Full Name: abc	ERROR!!!

	Email-Id: xyz@gmail.com Gender: Male Address: Wadala Date of Birth: 08-06-2000 Phone Number: 1234567890 Vehicle Details (Primary): MH03678333 Vehicle Details (Optional): MH01TY1111 Password: ***** Confirm Password: *****	Invalid Vehicle Details (Primary)!
4	Full Name: abc Email-Id: xyz@gmail.com Gender: Male Address: Wadala Date of Birth: 08-06-2000 Phone Number: 1234567890 Vehicle Details (Primary): MH03CY8333 Vehicle Details (Optional): MH01TY1111 Password: ***** Confirm Password: ***	ERROR!!! Password and Confirm Password must be same!
5	Full Name: abc Email-Id: xyz@gmail.com Gender: Male Address: Wadala Date of Birth: 08-06-2000 Phone Number: 12345678 Vehicle Details (Primary): MH03CY8333 Vehicle Details (Optional): MH01TY1111 Password: ***** Confirm Password: *****	ERROR!!! Invalid Phone Number!
6	Full Name: abc Email-Id: xyz@gmail.com Gender: Male Address: Wadala	Update Successful!

	Date of Birth: 08-06-2000 Phone Number: 1234567890 Vehicle Details (Primary): MH03CY8333 Vehicle Details (Optional): MH01TY1111 Password: ***** Confirm Password: *****	

Chapter 6

RESULTS AND DISCUSSION

6.1 Test Reports

Project Name	Toll Booth Management System
Product	http://localhost:64320/Index.aspx
Project Description	In current scenario, Due to the limited number of toll booths and slow collection process, the average waiting time per vehicle is very high. So to overcome this situation, we can make use of the proposed project. This project is to make the drivers and toll employee life with ease. This proposed system systematically manages arrival time of all the vehicles and stores data which can helpful for surveillance and security. This system thus automates the entire toll booth collection and monitoring process with ease using RFID plus IOT based system.
Project Type	Website (Embedded System)
Duration	8th August 2020 to 30th March 2021
Total no of Test Cases Executed	35
Passed Test Cases	27
Failed Test Cases	8

6.2 User Documentation

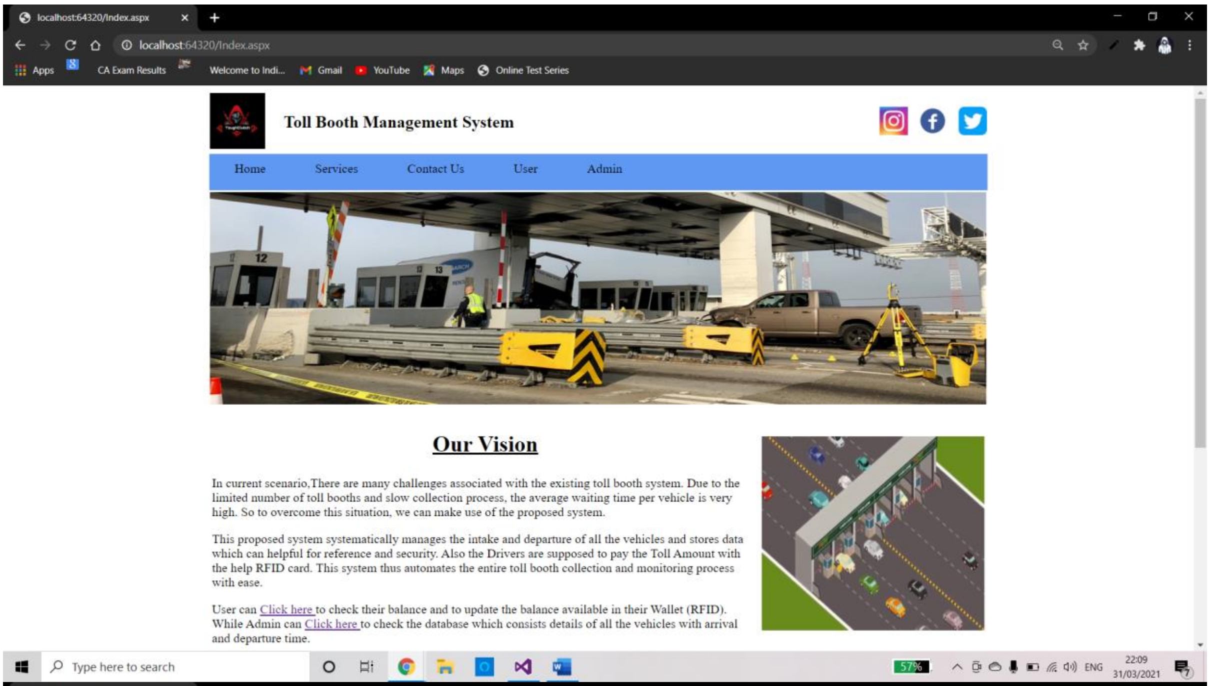
This project/product “IOT Based Toll Booth Management System” is a new concept coming in the markets now a days. The main concept here is to provide a safe and quick journey for drivers to operate by his/her website. And at the same time Admin can have access to Vehicle database which can result into security and surveillance of Toll Booth.

This product comes up with two main concept of Toll Booth Management, one is recharging/updating the wallet (RFID) and the other one is Toll Booth supervision as the vehicle details gets stored in the database. Which would be very useful for surveillance and security. So, basically this project is a web application. It’s a combination of two different activities working together resulting a great product.

Flow of the product:

1. First the User and Admin have to register themselves in the Web Application with the link given below:
2. After registration the User and Admin can login in the website.
3. After Successful Login, The User can use the Website to add/ recharge the Wallet i.e. RFID.
4. User (Driver) uses the RFID to successfully pay the Toll Amount.
5. Admin can use the Website to check the Vehicle details such as Vehicle Owner Name, Vehicle Owner Phone Number, Vehicle Owner Address and Vehicle Number, etc. along with the arrival and departure time.
6. This Website also serves various activities such as Contact Us which can be helpful for both if User and Admin needs any personal assistance.
7. The Website will also include the latest and updated Toll Amount of that particular Toll Booth.
8. The User and Admin can logout after he/she has done his/her purpose.

1. This is Homepage. This consists Our Vision with respect to our Project i.e. Toll Booth Management System



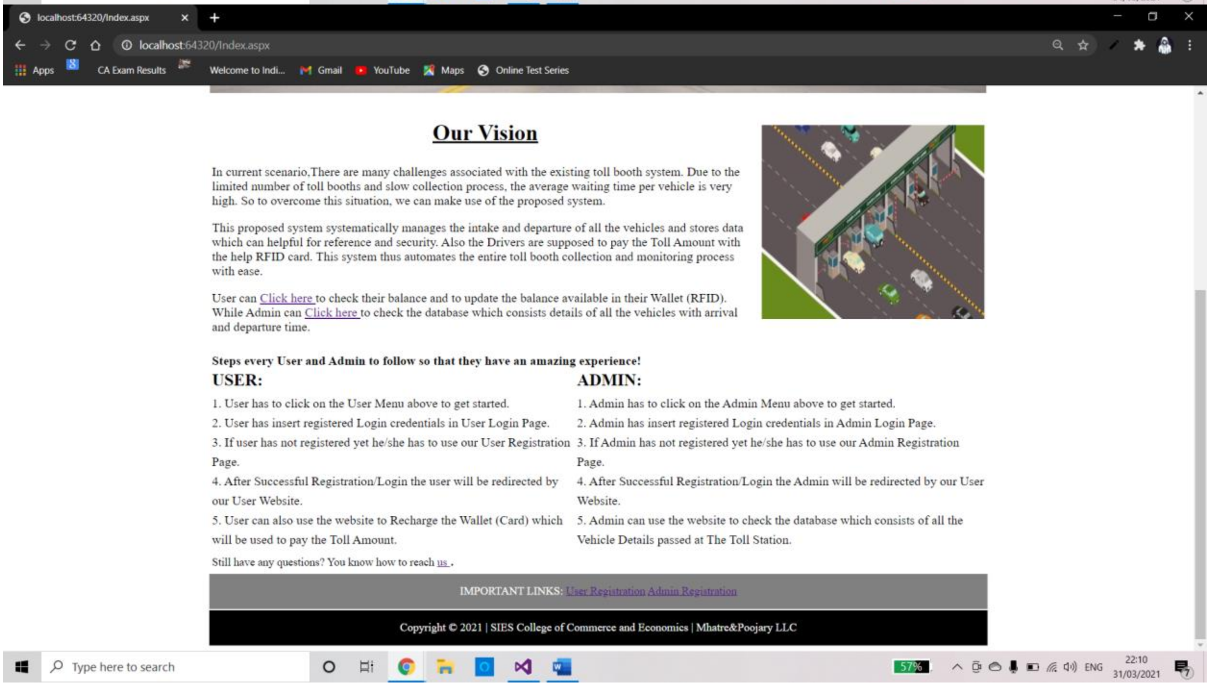
Toll Booth Management System

Our Vision

In current scenario,There are many challenges associated with the existing toll booth system. Due to the limited number of toll booths and slow collection process, the average waiting time per vehicle is very high. So to overcome this situation, we can make use of the proposed system.

This proposed system systematically manages the intake and departure of all the vehicles and stores data which can helpful for reference and security. Also the Drivers are supposed to pay the Toll Amount with the help RFID card. This system thus automates the entire toll booth collection and monitoring process with ease.

User can Click here to check their balance and to update the balance available in their Wallet (RFID). While Admin can Click here to check the database which consists details of all the vehicles with arrival and departure time.



Our Vision

In current scenario,There are many challenges associated with the existing toll booth system. Due to the limited number of toll booths and slow collection process, the average waiting time per vehicle is very high. So to overcome this situation, we can make use of the proposed system.

This proposed system systematically manages the intake and departure of all the vehicles and stores data which can helpful for reference and security. Also the Drivers are supposed to pay the Toll Amount with the help RFID card. This system thus automates the entire toll booth collection and monitoring process with ease.

User can Click here to check their balance and to update the balance available in their Wallet (RFID). While Admin can Click here to check the database which consists details of all the vehicles with arrival and departure time.

Steps every User and Admin to follow so that they have an amazing experience!

USER:

1. User has to click on the User Menu above to get started.
2. User has insert registered Login credentials in User Login Page.
3. If user has not registered yet he/she has to use our User Registration Page.
4. After Successful Registration/Login the user will be redirected by our User Website.
5. User can also use the website to Recharge the Wallet (Card) which will be used to pay the Toll Amount.

ADMIN:

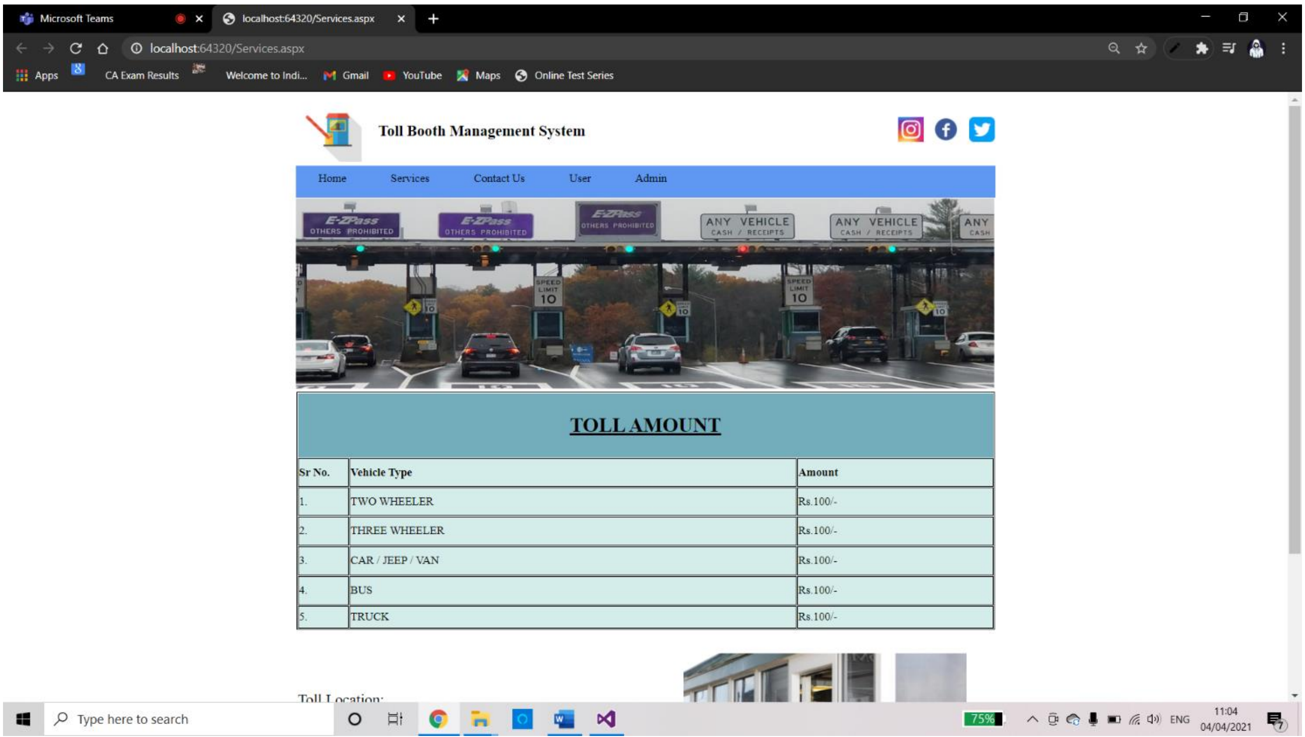
1. Admin has to click on the Admin Menu above to get started.
2. Admin has insert registered Login credentials in Admin Login Page.
3. If Admin has not registered yet he/she has to use our Admin Registration Page.
4. After Successful Registration/Login the Admin will be redirected by our User Website.
5. Admin can use the website to check the database which consists of all the Vehicle Details passed at The Toll Station.

Still have any questions? You know how to reach us .

IMPORTANT LINKS: User Registration Admin Registration

Copyright © 2021 | SIES College of Commerce and Economics | Mhatre&Poojary LLC

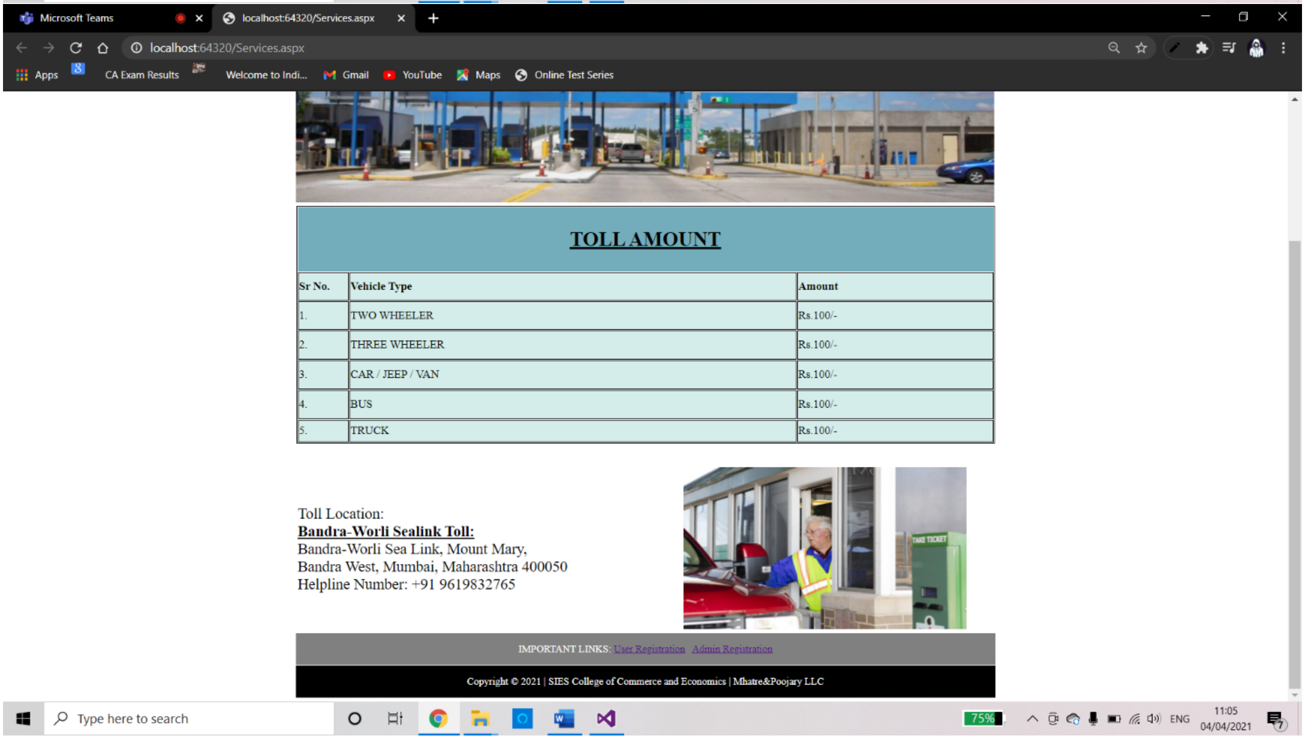
2. Services. This Page consists of Toll Amount which will be helpful for the User for referring purpose.



TOLL AMOUNT

Sr No.	Vehicle Type	Amount
1.	TWO WHEELER	Rs.100/-
2.	THREE WHEELER	Rs.100/-
3.	CAR / JEEP / VAN	Rs.100/-
4.	BUS	Rs.100/-
5.	TRUCK	Rs.100/-

Toll Location:



TOLL AMOUNT

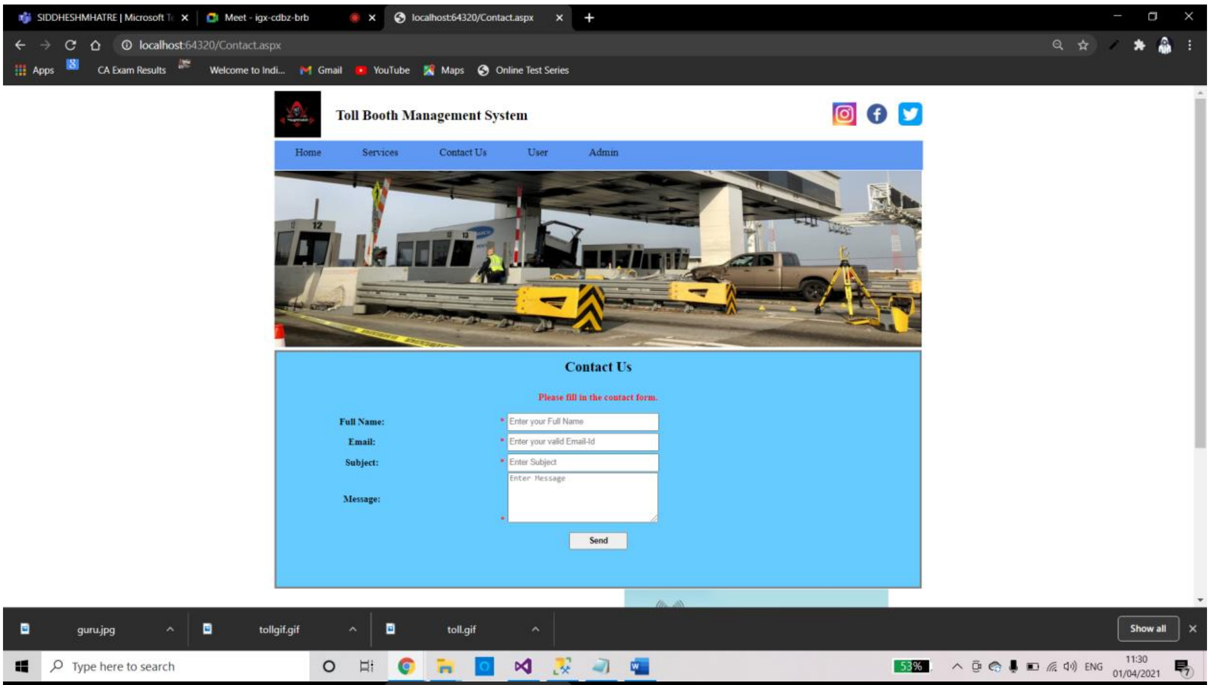
Sr No.	Vehicle Type	Amount
1.	TWO WHEELER	Rs.100/-
2.	THREE WHEELER	Rs.100/-
3.	CAR / JEEP / VAN	Rs.100/-
4.	BUS	Rs.100/-
5.	TRUCK	Rs.100/-

Toll Location:
Bandra-Worli Sealink Toll:
Bandra-Worli Sea Link, Mount Mary,
Bandra West, Mumbai, Maharashtra 400050
Helpline Number: +91 9619832765

IMPORTANT LINKS: User Registration Admin Registration

Copyright © 2021 | SIES College of Commerce and Economics | Mhatre&Poojary LLC

3. Contact Us. This page is made for both User and Admin. They can simply elaborate their issue and the details will be stored in the database which can be helpful for future development.



Toll Booth Management System

Contact Us

Please fill in the contact form.

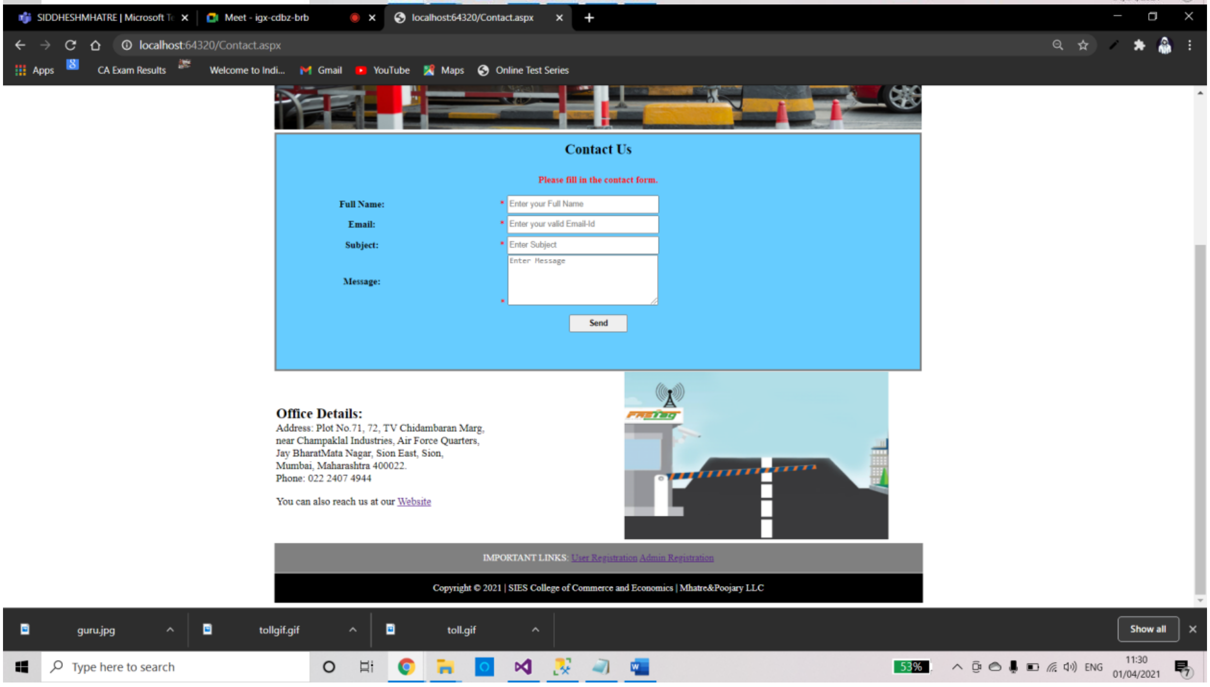
Full Name: * Enter your Full Name

Email: * Enter your valid Email-Id

Subject: * Enter Subject

Message: * Enter Message

Send



Contact Us

Please fill in the contact form.

Full Name: * Enter your Full Name

Email: * Enter your valid Email-Id

Subject: * Enter Subject

Message: * Enter Message

Send

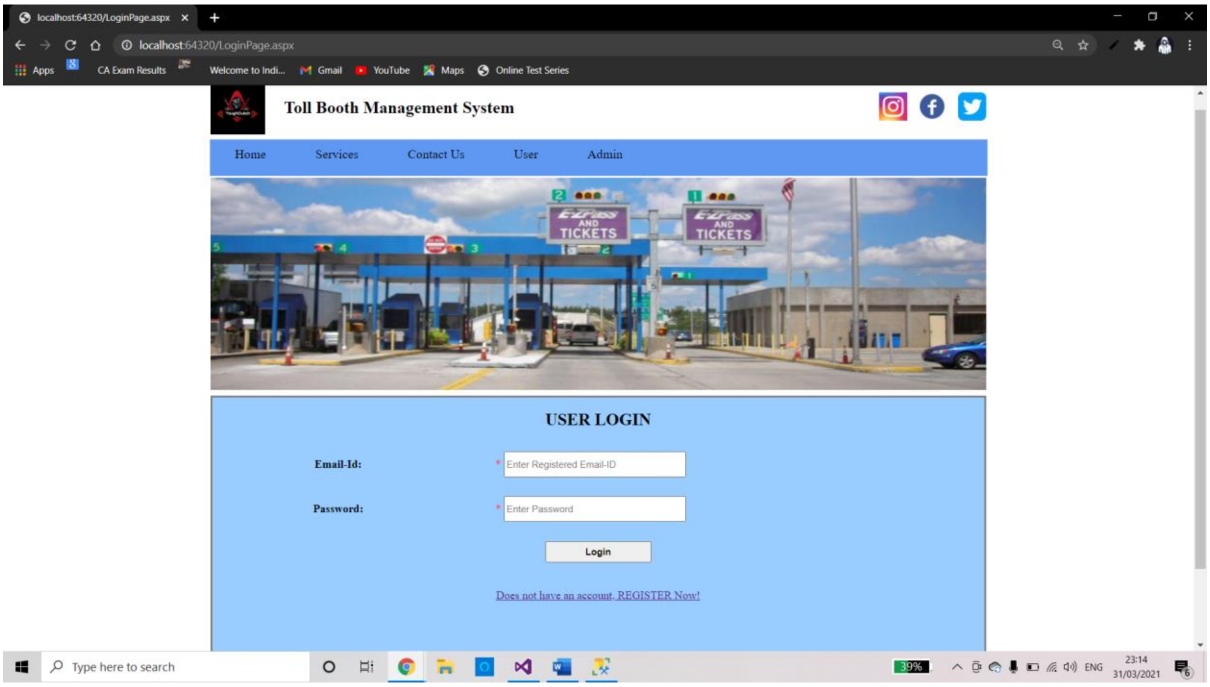
Office Details:
Address: Plot No.71, 72, TV Chidambaran Marg,
near Champaklal Industries, Air Force Quarters,
Jay BharatMata Nagar, Sion East, Sion,
Mumbai, Maharashtra 400022.
Phone: 022 2407 4944

You can also reach us at our Website

IMPORTANT LINKS: User Registration Admin Registration

Copyright © 2021 | SIES College of Commerce and Economics | Mhatre&Poojary LLC

4. User Login.



Toll Booth Management System

USER LOGIN

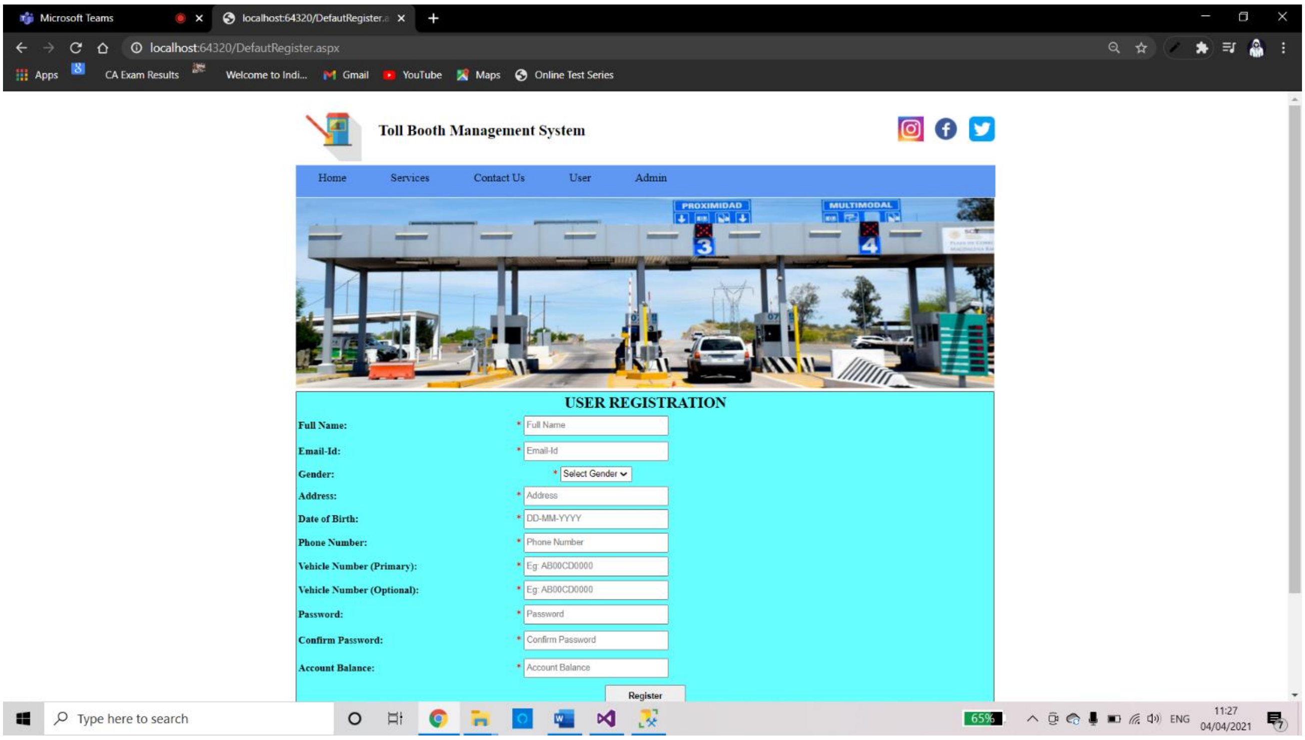
Email-Id: * Enter Registered Email-ID

Password: * Enter Password

Login

Does not have an account, REGISTER Now!

5. User Registration.



Toll Booth Management System

USER REGISTRATION

Full Name: * Full Name

Email-Id: * Email-Id

Gender: * Select Gender

Address: * Address

Date of Birth: * DD-MM-YYYY

Phone Number: * Phone Number

Vehicle Number (Primary): * Eg: AB00CD0000

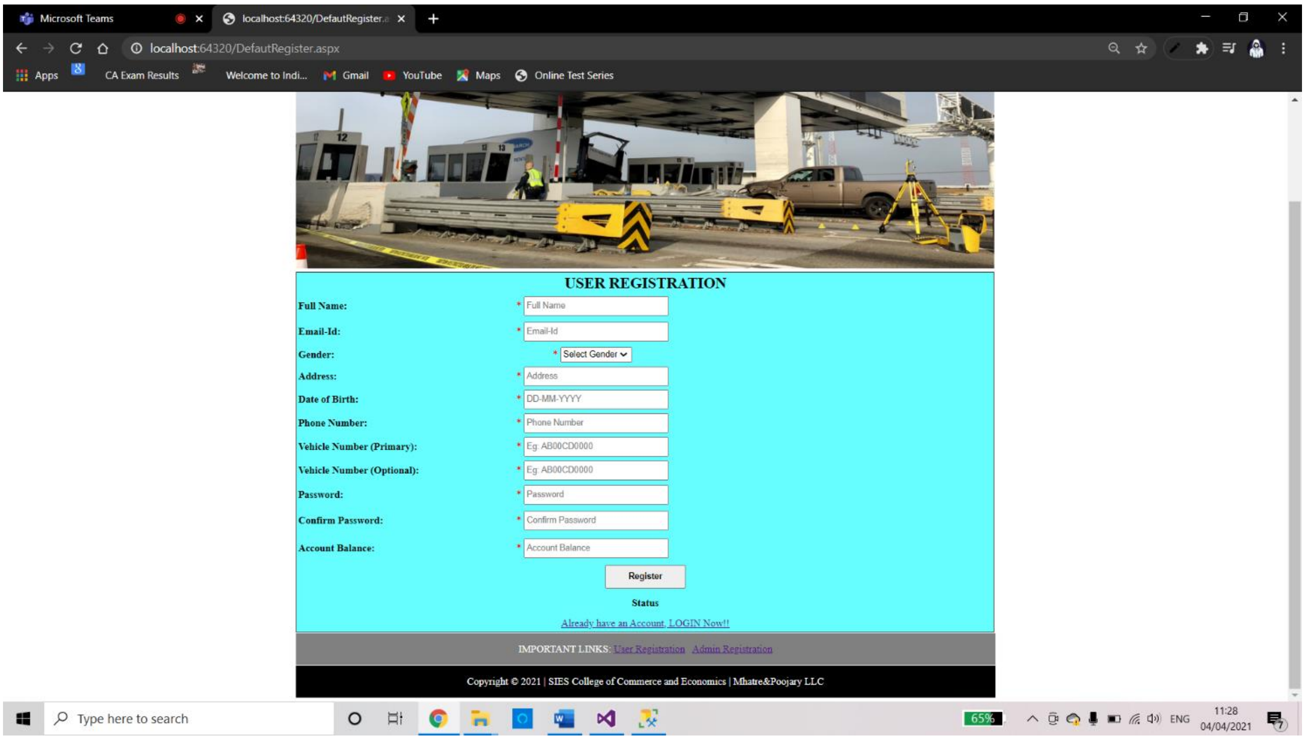
Vehicle Number (Optional): * Eg: AB00CD0000

Password: * Password

Confirm Password: * Confirm Password

Account Balance: * Account Balance

Register



USER REGISTRATION

Full Name:	* Full Name
Email-Id:	* Email-Id
Gender:	* Select Gender
Address:	* Address
Date of Birth:	* DD-MM-YYYY
Phone Number:	* Phone Number
Vehicle Number (Primary):	* Eg. AB00CD0000
Vehicle Number (Optional):	* Eg. AB00CD0000
Password:	* Password
Confirm Password:	* Confirm Password
Account Balance:	* Account Balance

Register

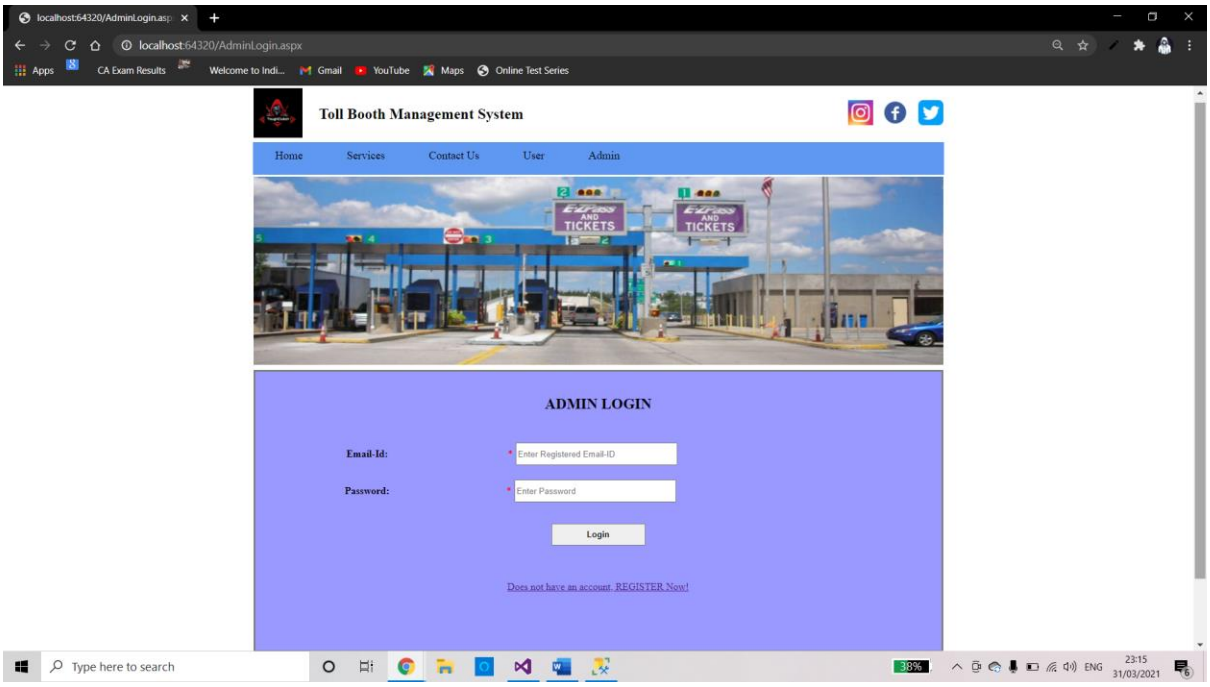
Status

Already have an Account, LOGIN Now!!

IMPORTANT LINKS: User Registration Admin Registration

Copyright © 2021 | SIES College of Commerce and Economics | Mhatre&Poojary LLC

6. Admin Login



Toll Booth Management System

Home Services Contact Us User Admin

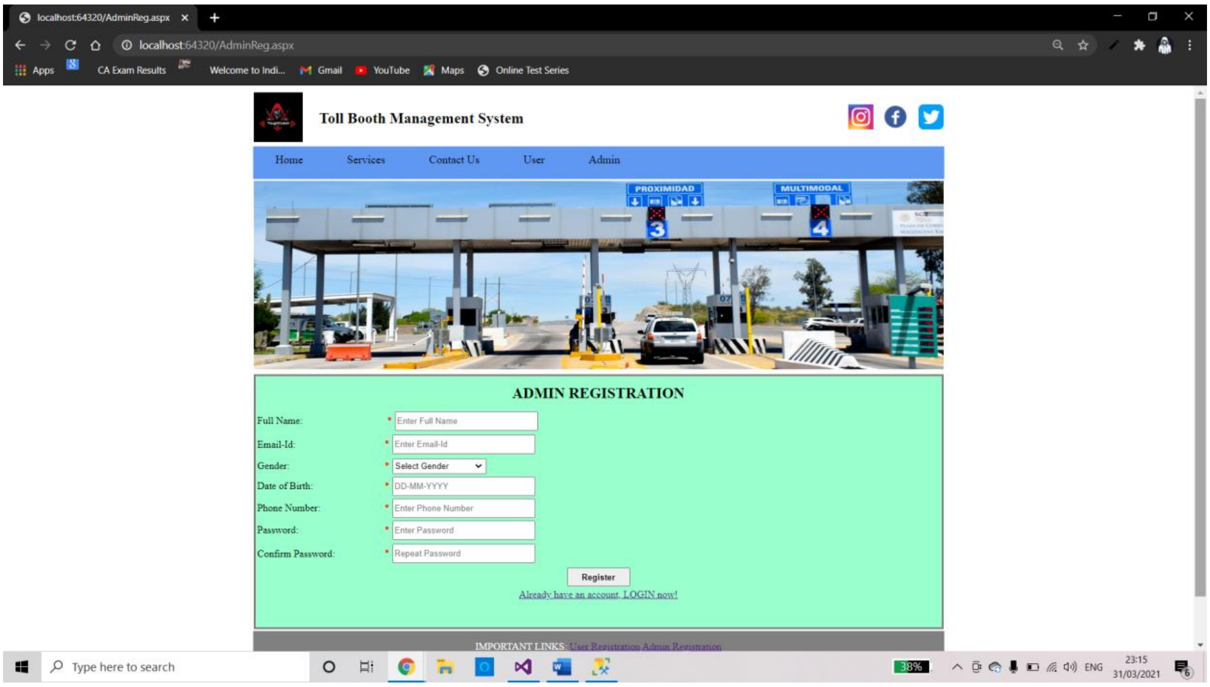
ADMIN LOGIN

Email-Id:	* Enter Registered Email-ID
Password:	* Enter Password

Login

Does not have an account, REGISTER Now!

7. Admin Registration



ADMIN REGISTRATION

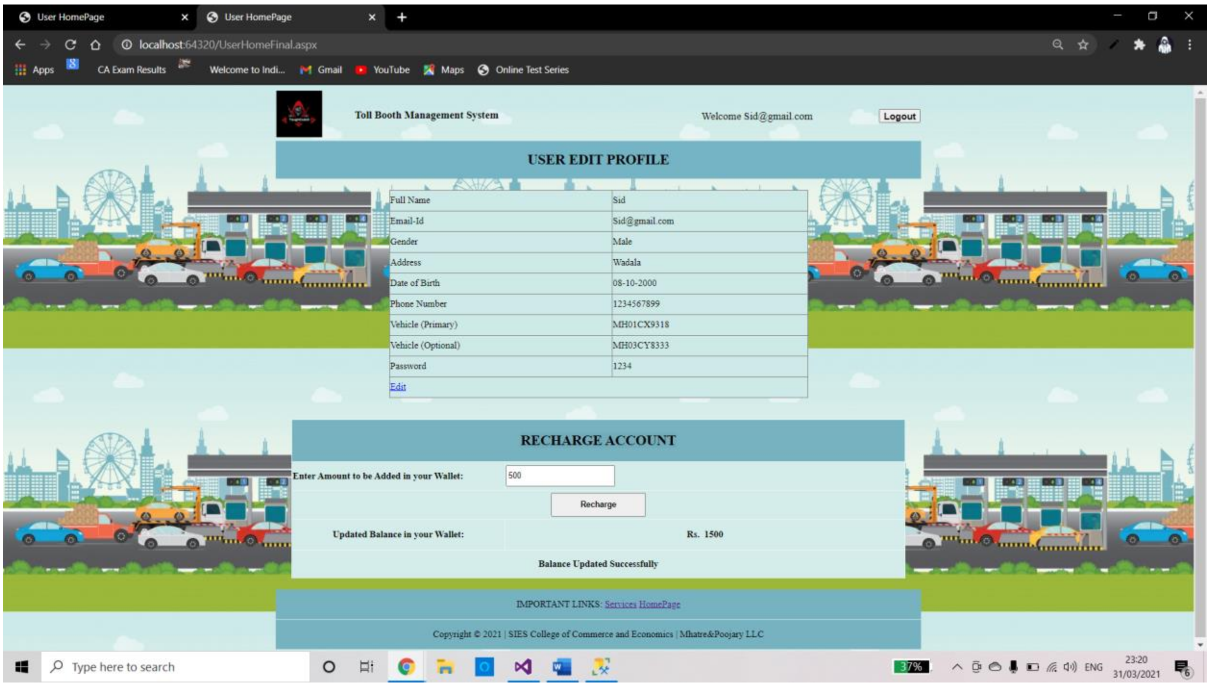
Full Name:	* Enter Full Name
Email-Id:	* Enter Email-Id
Gender:	* Select Gender
Date of Birth:	* DD-MM-YYYY
Phone Number:	* Enter Phone Number
Password:	* Enter Password
Confirm Password:	* Repeat Password

Register

Already have an account. LOGIN now!

IMPORTANT LINKS: User Registration Admin Registration

8. User Homepage. This page consists of User Edit Profile which will be used to Edit User Credentials and also Wallet Recharge Facility. User can simply add the Amount and the Amount will be updated in the Database.



USER EDIT PROFILE

Full Name	Sid
Email-Id	Sid@gmail.com
Gender	Male
Address	Wadala
Date of Birth	08-10-2000
Phone Number	1234567899
Vehicle (Primary)	MH01CX9318
Vehicle (Optional)	MH03CY8333
Password	1234
Edit	

RECHARGE ACCOUNT

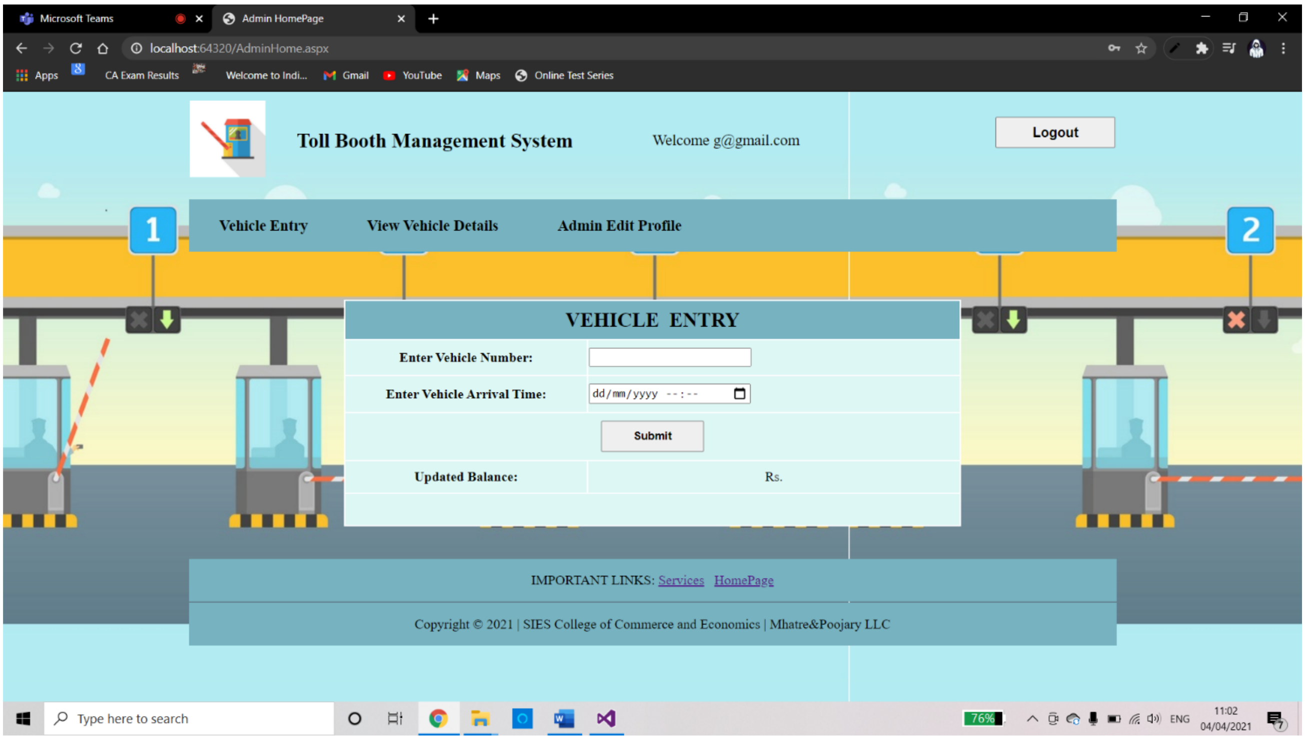
Enter Amount to be Added in your Wallet:	500
	Recharge
Updated Balance in your Wallet:	Rs. 1500

Balance Updated Successfully

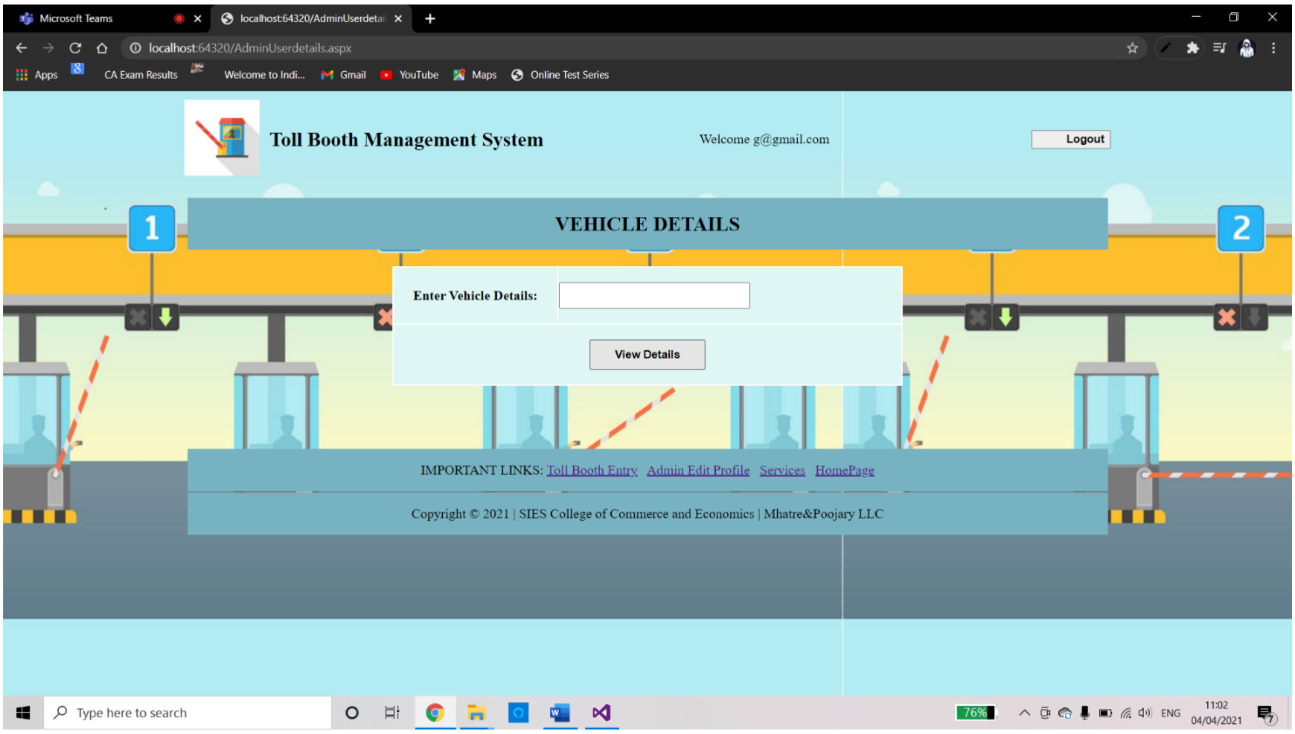
IMPORTANT LINKS: Services HomePage

Copyright © 2021 | SIES College of Commerce and Economics | Mhatre&Poojary LLC

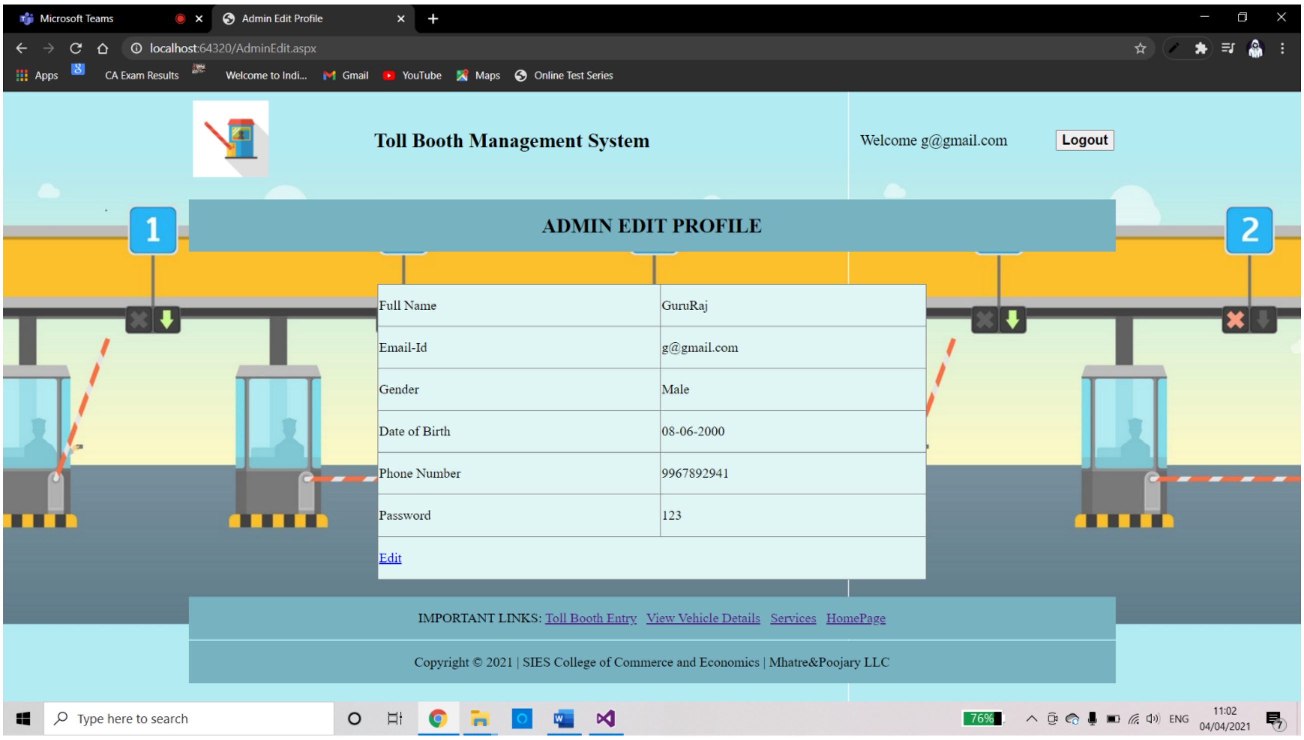
9. Admin Homepage. This page consists of Toll Entry which will used to note the Entry of the all Vehicles at the Toll Booth. After entering valid details, User Balance will be deducted and Vehicle Entry time get noted. This page also show the User Details along with Entry time which will be shown once the Admin enters the valid Vehicle Number.



10. View Vehicle Details. This page show the User Details along with Entry time which will be shown once the Admin enters the valid Vehicle Number.


Microsoft Teams
localhost:64320/AdminUserdeta
localhost:64320/AdminUserdetails.aspx
Apps
CA Exam Results
Welcome to Indi...
Gmail
YouTube
Maps
Online Test Series
Toll Booth Management System
Welcome g@gmail.com
Logout
VEHICLE DETAILS
Enter Vehicle Details:
View Details
IMPORTANT LINKS: Toll Booth Entry Admin Edit Profile Services HomePage
Copyright © 2021 | SIES College of Commerce and Economics | Mhatre&Poojary LLC
Type here to search
76%
ENG
11:02
04/04/2021

11. Admin Edit Profile. This page is used to edit the Admin Credentials.



Toll Booth Management System

Welcome g@gmail.com Logout

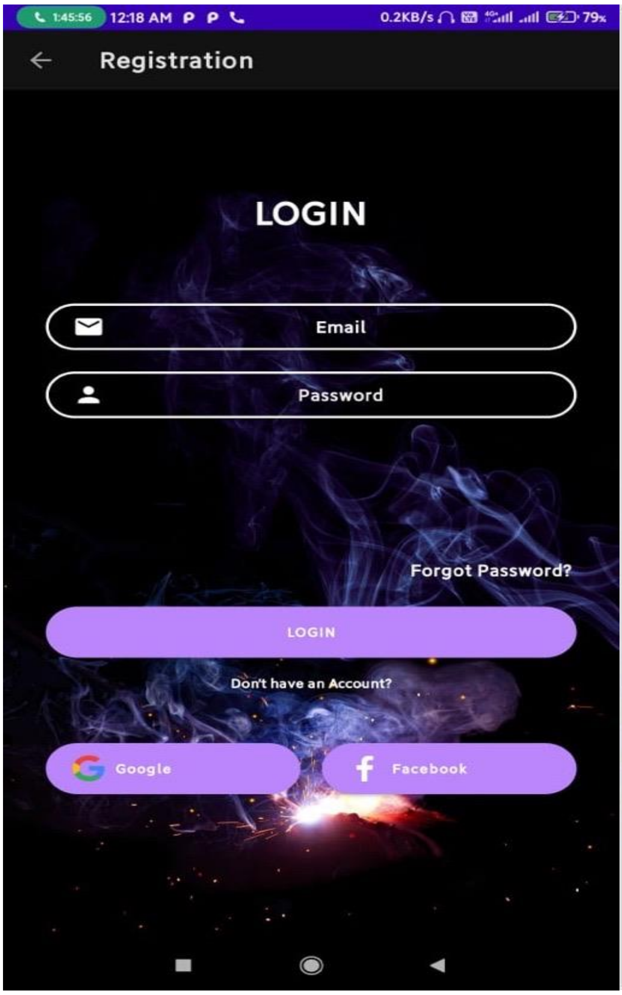
ADMIN EDIT PROFILE

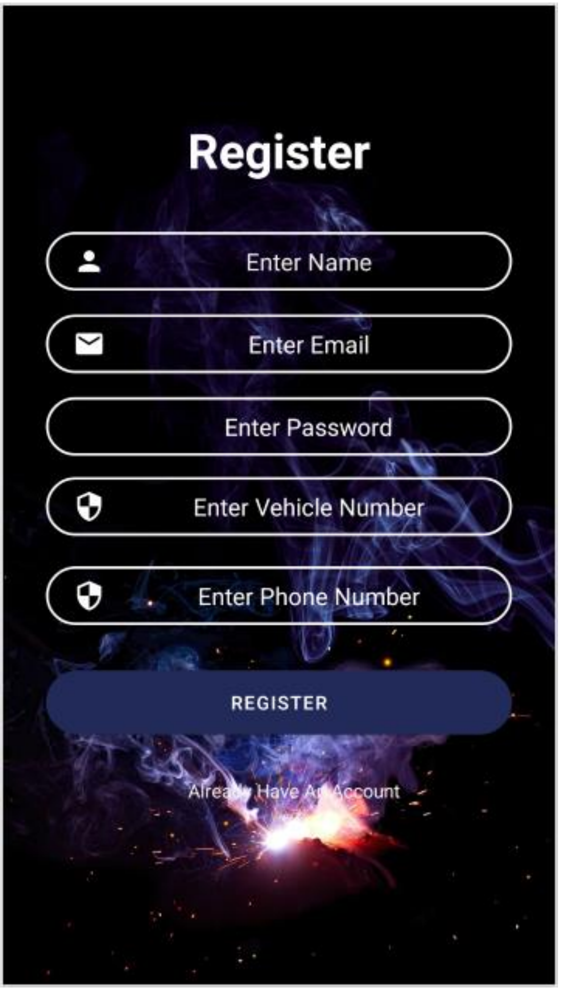
Full Name	GuruRaj
Email-Id	g@gmail.com
Gender	Male
Date of Birth	08-06-2000
Phone Number	9967892941
Password	123
Edit	

IMPORTANT LINKS: Toll Booth Entry View Vehicle Details Services HomePage

Copyright © 2021 | SIES College of Commerce and Economics | Mhatre&Poojary LLC

Android:


1:45:56
12:18 AM
0.2KB/s
79%
Registration
LOGIN
Email
Password
Forgot Password?
LOGIN
Don't have an Account?
Google
Facebook


Register
Enter Name
Enter Email
Enter Password
Enter Vehicle Number
Enter Phone Number
REGISTER
Already Have An Account

Chapter 7
Conclusion

7.1 Conclusion

It was really an amazing and wonderful experience for us to work on this project. In the beginning we found the project a bit vast and difficult, but we have learnt a lot by selecting this topic as our project. It was really a learning experience by working on this project. Working on this project made us go through different phases of project development life cycle. This project gave us a good idea about how the professional world of software developers actually work.

As it is Web Based Project we got to learn about the Web platform in this project itself. Connectivity between Web and Database was all quite challenging. Due to this project we got to learn various new concepts which will obviously help us in the Web development in the future.

7.1.1 Significance of the System

The main objective of this system is to ease the challenges associated with the existing tolling process. Managing multiple toll booths is a very complicated task. The system here is a smart card based toll booth system that is monitored using IOT. The Internet server maintains all the data of user accounts and also their balance. The system at toll booths will monitor the cards scanned when a car arrives at the toll booth. If user balance is sufficient, the toll amount is deducted online and web system sends signal back to the card scanner system that the user has been billed. Also system allows to store data of all the vehicles passed at particular time intervals for later reference and surveillance. This system also consists of latest and updated Toll Amount of that particular Toll Booth.

7.2 Limitations of the System

Every product/project has its own limitations. There can’t be a product without limitations aseveryone knows. So, the same way even our project has its limitations. They are follows.

1. Internet is compulsory in the system for working of this project.
2. If the user lost his/hers RFID card it will result into failure.

7.3 Future Scope of the Project

Every product or project always needs some or other improvement in every aging time, so do this project needs. Here in our project in the Toll Booth Manager (Admin) only have the ability to see the database which consist of Vehicle details. Admin cannot block the User at that same instance of time.

We can add more features like Call Supervision, Message Supervision, also we can let the driver inform the Toll Station if they encountered any accidents in their journey, and we can also include a section to report any issue to Police or Hospital if the Driver encounter any. These all new features can improve the product in different level.

These are the various area where this project can be expanded. Expanding this project will lead to good scope to Drivers and Toll Booth Authority. As Toll Booth is providing a safe and quick environment to their Drivers. And the Driver can have a cheerful journey.

References

https://stackoverflow.com/

https://www.w3schools.com/

https://www.youtube.com/

https://www.google.com/