The Kelkar Education Trust's
V G Vaze College of Arts, Science and Commerce
(Autonomous)

EMPLOYEE TRANSPORT MANAGEMENT SYSTEM

A Project Report

Submitted in partial fulfilment of the Requirements for the award of the Degree of

BACHELAOR OF SCIENCE
(INFORMATION TECHNOLOGY)

By

Anuj Pravin Mhatre
3945A034

Under The Esteemed Guidance of
Mrs. Mohini Bhole
Assistant Professor

DEPARTMENT OF INFORMATION TECHNOLOGY
V G VAZE COLLEGE OF ARTS, SCIENCE AND COMMERCE
(AUTONOMOUS)
(Affiliated to University of Mumbai)
Mulund, 400080 MAHARASHTRA
2022-2023

THE KELKAR EDUCATION TRUST'S VINAYAK GANESH VAZE COLLEGE OF ARTS, SCIENCE AND COMMERCE
(Autonomous)

MULUND, MAHARASHTRA, 400081

DEPARTMETN OF INFORMATION TECHNOLOGY

CERTIFICATE

This is to certify that the project entitled,"_____",
is bonafied work of_____bearing Roll No:_____submitted in partial fulfillment of the requirements for the award of degree of BACHELOR OF SCIENCE in INFORMATIONTECHNOLOGY.

Internal Guide **Head Of Department**

External Guide

Date: **College Seal**

THE APPROVAL PROJECT PROPOSAL

(Note:All entries of the proforma of approval should be filled up with appropriate and complete information. Incomplete proforma of approval in any respect will be summarily rejected.)

PNR No :

1. Name of the Student	
2. Title of the Project	
3. Name of the Guide	
4. Is this your first submission?	
Student Name :	
Student Control ID :	
Student Roll No. :	

ACKNOWLEDGEMENT

I am glad to say that, I have satisfactorily reached my intentions to make this documentation. However, it would not have been possible without the kind support and help of many individuals. I would like to extend my sincere thanks to all of them.

I am highly indebted of my guide, **Mrs Mohini Bhole** for her guidance and constant supervision as well as for providing necessary information regarding the project.

I would also like to extend my gratitude towards **Principal (Prof) Dr. Preeta Nilesh** and the Head of the Department, **Mrs. Pournima Bhangale** for providing me with all the facilities that was required.

Then I would like to thank my parents who have helped me with their valuable suggestions and guidance which has been very helpful.

Last but not the least I would like to thank my classmates who have helped me a lot. Directly or indirectly their contribution was indispensable, and will always be remembered.

This opportunity has given me a valuable experience about Software development for which I shall be thankful for the years to come.

~Anuj Pravin Mhatre

DECLARATION

I here by declare that the project entitled, "_____

_____"done at place where the project is done, 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, none 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 (INFORMATIONTECHNOLOGY) to be submitted as final semester project as partof our curriculum.

Name : _____

Signature : _____

EMPLOYEE TRANSPORT MANAGEMENT SYSTEM (ETMS)

TABLE OF CONTENTS

List of figures: -

List of tables: -

SYNOPSIS

❖ Statement about the problem: -

Employee transport management system (ETMS). Enables easy and user interactive access to manage employee transportation for respective company. It fetches the location of employee. And manages cab according to geolocation.

❖ Why this topic: -

Currently employee transport management system is manual. And not proper organized system user can request for hassle free and fast and the employees will get safe and hassle-free rides to office from their home or vice versa.

❖ Objective and scope of system: -

Objective: The system helps the companies for transportation of their employees and they can travel hassle free.

Scope: The project has wide scope, as it is not intended to a particular company. This project is going to develop generic software. Which can be applied by any business, companies more over it provides facility to its users.

❖ Methodology: -

Incremental model

❖ Propose architecture: -

Three tire architecture

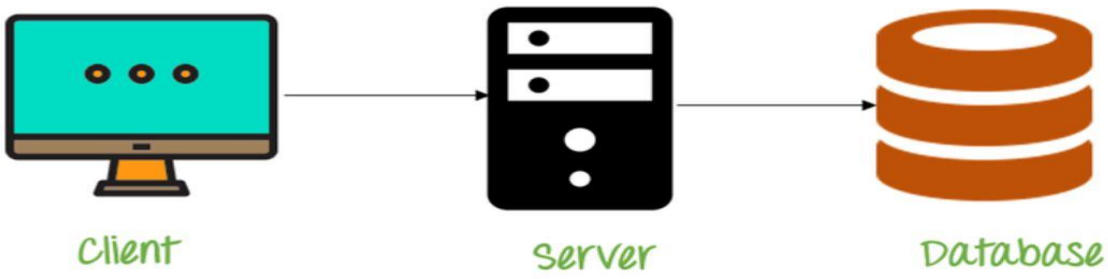


Fig 1.1 Proposed architecture

- **Requirements related to software and hardware:**

Software: Visual studio

Front end: HTML, CSS, JAVASCRIPT

Back end: My SQL

Hardware: O.S-windows 10

RAM-8GB

Processor-i5

- **Platform:** .NET
- **How is your system is contributing to your society?**

With the help of this system the respective company get user friendly system for transportation for their employees. And the employee can travel stress free and do their work properly.

CHAPTER 1:

INTRODUCTION

1.1 Background: -

The Current Employee transport system is very time Consuming requites and lot of manual work. The Booking of cab for employees is done physical the employee has to visit the department for transportation this is very time consuming. for this service Besides this, a particular organization. or company must store data of employee's locations and cab details Manually in files on Paperwork which consumes lot of space. It also results in more consumption of paper. The overall background scenario of the current system is average.

1.2 Objectives: -

1. This system helps users to find cab as per their requirements.

2. This system will also show status of cab for users.

3. Users can also complaint about drivers.

1.3 Purpose, scope, and applicability: -

1.3.1 Purpose: Currently employee transport management system is manual. And not proper organized system user can request for hassle free and fast and the employees will get safe and hassle-free rides to office from their home or vice versa.

1.3.2 Scope: The project has wide scope, as it is not intended to a particular company. This project is going to develop generic software. Which can be applied by any business, companies more over it provides facility to its users. Also, the system is going to provide huge amount of data of cabs and driver.

1.3.3 Applicability: this system will help the various respective companies for transportation of their employees. This system is efficient and easy to store data. This will be useful for employees for transportation and book the cab for them. This will be helpful for various companies.

CHAPTER 2:

SURVEY OF TECHNOLOGIES

❖ React.js: -

React is an open-source, front-end JavaScript library for creating interactive UIs. React is developed and maintained by Facebook and a large community of dedicated developers. React can also be used as a base for a single-page or mobile application. React is based on the MVVM (Model-View-View Model) pattern, which ultimately allows the view and model to communicate directly with each other. This enables React to break down the app into modular, single-purpose components that are more complex for your applications.

❖ Node.js: -

Node.js is an open-source, cross-platform, back-end, JavaScript runtime environment for writing server-side applications using JavaScript. Node.js is usually used for non-blocking, event-driven servers for traditional websites and back-end API services. Node.js is known for being lightweight and efficient and is perfect for data-intensive, real-time applications that run across devices. Popular websites that use Node.js include Netflix, PayPal, Medium, LinkedIn, Uber, and eBay.

❖ Django: -

Django is a high-level, open-source, MVC Python web framework for secure and maintainable websites. The framework is named after the guitarist Django Reinhardt. Django has been gaining popularity for its simplicity, ease of use, pragmatic design, yet fully-featured compared to many other frameworks. Django is also very beginner-friendly and is suited for both frontend and backend. Django can be used for all types of websites, such as social networking, chat applications, interactive pages, or content management. It is also compatible with most major databases. Django also inherits all of Python's benefits, such as great support, productivity boost, and advanced development

speed. Many popular websites are built using Django, such as YouTube, Instagram, Spotify, Drobox, and Pinterest.

❖ ASP.NET: -

ASP.NET is a web application framework developed and marketed by Microsoft to allow programmers to build dynamic web sites. It allows you to use a full featured programming language such as C# or VB.NET to build web applications easily. This tutorial covers all the basic elements of ASP.NET that a beginner would require to get started.

❖ Bootstrap: -

Bootstrap is a free and open-source tool collection for creating responsive websites and web applications. It is the most popular HTML, CSS, and JavaScript framework for developing responsive, mobile-first websites. Nowadays, the websites are perfect for all the browsers and for all sizes of screens. All thanks to Bootstrap developers – Mark Otto and Jacob Thornton of Twitter, though it was later declared to be an open-source project.

❖ Which language is use for frontend and backend?

I am going to use HTML, CSS, JavaScript for frontend development. As the HTML is easy to use to develop webpage by CSS, we can design it properly. With JavaScript we can validate form for backend I am going to use My SQL and C#. It is easy to store and read the database. Also, I am going to use SQL server to connecting database to webpage.

CHAPTER 3:

REQUIREMENTS

AND ANALYSIS

3.1 Problem definition: -

The main problem of the current employee transport system is that it requires lot of Manual Work and all the records needs to be Saved properly and hence takes up lot of storage space. Currently a register is use to maintain all the info of employees and cabs. This task requires a lot of physical work. employee transport system enables easy and user interactive access to book they transportation by this employee can book its transportation easily there is no more physical work to do and no more physical storage

Some key problems faced due to current system: -

- **Physical / Manual work: -**

  The current system is totally based on manual work and this project will reduce the manual work as many operations can be performed through the system and hence the physical involvement will be minimized.

- **Storage space: -**

  The current employee transport system stores data in register and thus registers are maintained for all data month wise which takes up lot of storage space.

- **Delay in booking: -**

  This system creates issues and delay in travelling and time management.

3.2 Requirement specification: -

1. Employee transport system is to be designed, will provide smooth flow of service between the admin and the user.
2. The admin will be management and user will be employee.
3. The admin panel needs to login into system the admin will have many functions they can modify pick-ups and drop-offs, driver, vehicle related.
4. User need to register after successful registration, employee can login using credentials after login they can check the slot and vehicle available according to shifts timings.
5. All these features will provide a hassle-free pick and drop management system for a
   company.

3.2.1Requirement gathering: -

Various techniques used for requirement gathering:

- Survey
- Interview
- User observation
- Document analysis

For my system, I have used interview method to collect information from users. Interviews are the primary ways for information gathering where the admin can have face-to-face interaction with user.

***Questionaries: -**

1.first I have told them about my project what I am going to make.

2.which mode of transport they prefer for travelling to office?

Ans- some of them said train but they also said that it is hectic and most of the times preferred the cab service to reach their office comfortably.

3.is their any proper system for employee transportation?

Ans- there is not a proper system for booking cab for transportation.

4.how, they book their cab in current situation?

Ans-they call their office and book the cab but sometimes they pickup call and sometimes do not they do not have proper communication.

5.what features they like to see in new system?

Ans-they need proper system for booking cab also view availability of cabs and can book and can book according to their shift timings.

6.do you want any user care service like feedback system?

Ans-yes for giving our opinion and ratings.

7.any suggestions or feature for my system?

Ans-you must ensure that cab which you have sending for employees is safe and driver is proper and non-acholic and ensure that cab have alert switch for women safety.

3.2.2 Requirement analysis: -

3.2.2.1Functional requirements: -

1. **Booking**: To book cab.
2. **Status**: To check status of your cab booking and which cab is allotted.
3. **Managing roles**: In managing roles should have respective profile to manage the access (admin, user).

3.2.2.2 Non-functional requirements: -

1. **Availability**: The system must work properly without any failure in it and it should be a stable system.
2. **Security**: All the data regarding users should be secured and should not be disclosed to anyone and not be misused.
3. **Reliability**: The system should be reliable and should work under the stated conditions.
4. **Usability**: System can be used for company for employee transportation.

3.2.2.3 System requirements: -

1. **Employee registration**: -

   Function: For registration of user.
   Description: The employee must need to register to get access to cab booking.
   Input: Name, phone no, employee id, set password.
   Source: User.
   Output: Registration complete.
   Action: After registration account of employee get created.
   Pre-condition: User must provide details like phone no, name, employee id, etc. Also visit website.
   Post-condition: User can login to account with registered username and password.
   Destination: stored in database.

2. **Login**: -

   Function: For login of user.
   Description: After registering the user will have to login in the system by the username/id and the password and after it is verified by the registered id and password the user will be logged in the system.
   Input: Username and password.
   Source: User.
   Output: login successful.
   Action: After login user can able to book cab.
   Pre-condition: Visit website, user must provide username and password.

Post-condition: If details are correct login will successful.

Destination: stored in database.

3. **Booking cab**: -

Function: To book cab.

Description: The employee can able to book cabs available in their location.

Input: Select cab in your location.

Source: User, admin.

Output: Your request will be accepted.

Action: After booking cab user will get response.

Pre-condition: location, searching cab, booking cab.

Post-condition: User can book cab and it will be arrived to their location.

Destination: stored in database.

3.3 Planning and scheduling: -

3.3.1 Activity table: -

Activity name	Start date	End date
Synopsis	15/06/22	18/06/22
Chapter 1: Introduction	20/06/22	25/06/22
1.1 Background		
1.2 Objectives		
1.3 Purpose, Scope, and Applicability		
1.3.1 Purpose		
1.3.2 Scope		
1.3.3 Applicability		
Chapter 2: Survey of technologies		
Chapter 3: Requirements and analysis		
3.1 Problem definition	27/06/22	02/07/22
3.2 Requirements specification		
3.2.1Requirements gathering	04/07/22	09/07/22
3.2.2Requirements analysis	11/07/22	16/07/22
3.3 Planning and scheduling	18/07/22	23/07/22
3.4 Software and Hardware Requirements		

3.5 Conceptual Models		
3.5.1 Entity relationship diagram		
3.5.2 Schema diagram	25/07/22	29/07/22
3.5.3 Data flow diagram	01/08/22	20/08/22
3.5.4 Class diagram	22/08/22	25/08/22
3.5.5 Use case diagram	29/08/22	10/09/22
3.5.6 Sequence diagram	12/09/22	13/09/22
3.5.7 Activity diagram	14/09/22	17/09/22
3.5.8 State diagram		
Chapter 4: System design		
4.1 User interface design	19/09/22	22/09/22
4.2 Test Cases Design		
Re-engeneering	02/11/22	09/12/22
Chapter 5: Implementation And Testing		
5.1 Implementation Approaches		
5.2 Coding Details and Code Efficiency	09/12/22	26/02/23
5.2.1 Coding Details		
5.2.1 Code Efficiency		
5.3 Testing Approach	01/01/23	26/02/23
5.3.1 Unit Testing		
Chapter 6: Results And Discussion		
6.1 Test Reports	02/02/23	26/02/23
6.2 User Documentation		
Chapter 7:Conclusions		
7.1 Conclusion	26/02/23	28/02/23
7.2 Limitations of System		
7.3 Future Scope of project		

Table 3.1 Activity table

3.3.2 Gantt chart: -

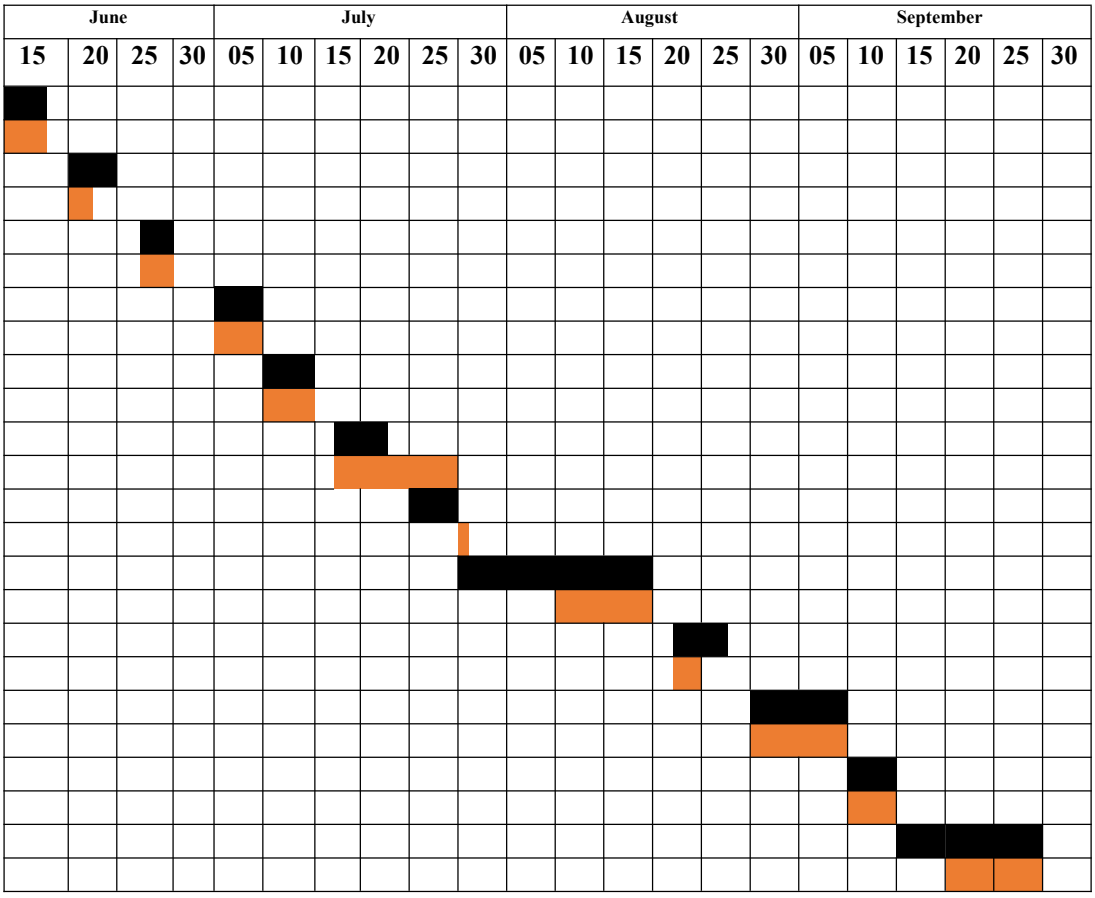


Fig 3.1 Gantt chart 1

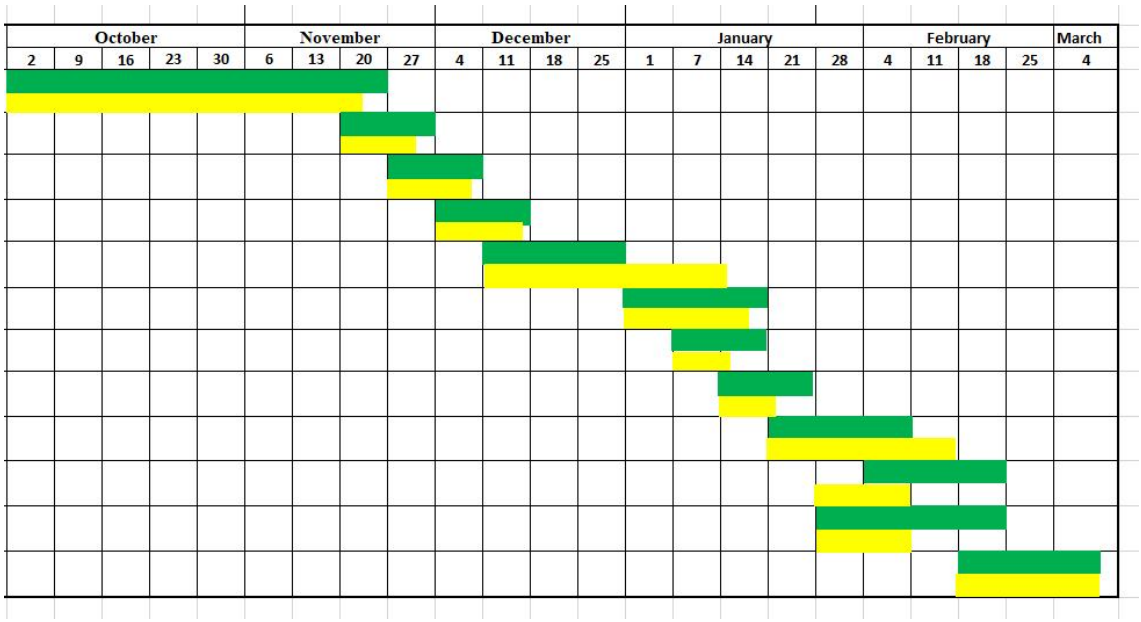


Fig 3.2 Gantt chart 2

3.4 Software and Hardware Requirements: -

3.4.1 Software requirements: -

Visual studio

Front end: - HTML

CSS

JavaScript

ASP.Net

Back end: - My SQL

C#

3.4.2 Hardware requirements: -

O.S: Windows 10, XP, 8

RAM: 8GB,2GB

Processer: i5, i3

3.5 Conceptual module: -

3.5.1 E-R Diagram: -

ER diagram stands for Entity Relationship Diagram that displays the relationship of the entity sets stored in the database. An entity in this context is an object, a component of

data. An entity set is a collection of similar entities. These entities can have attributes that define its properties. In software engineering an ER model is commonly formed to represent things that a business needs to remember in order to perform business processes. Consequently, the ER model becomes an abstract data model that defines a data or information structure which can be implemented in a database, typically a relational database.

Symbols: -

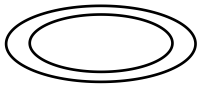
Name	Symbol	Description
Rectangle		Represent an entity
Ellipse		Represents an attribute
Double ellipse		Represents multivalued attribute
Diamond		Represents an relationship
Line		Links attribute to entity set

Table 3.2 Symbols for ER diagram

Reference: Database System and Concepts, A Silberschatz, H Korth, S Sudarshan, McGraw-Hill, Fifth Edition.

Entity sets: -

1. **Admin**
2. **User**
3. **Booking cab**
4. **Cab**
5. **Driver**

- **Admin: -** All the updating of information will be done by the admin.

1.Admin_id: - It is a primary key will be used to identify the admin.

2.Admin name: -name of admin.

3.Password:- To validate the admin at the time of login.

4.Username: -Username is use for login admin

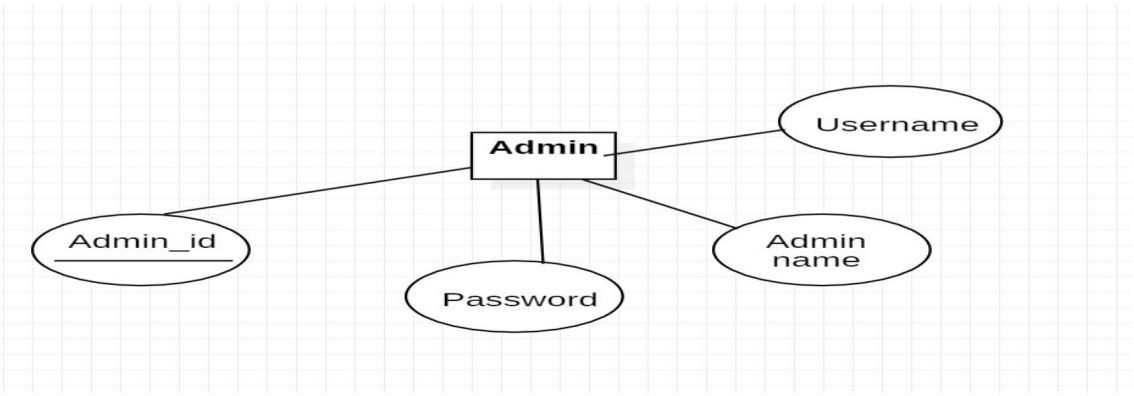


Fig 3.3 Admin entity set

- **User: -** User in my system are employees. Employees needs to register and login on website. Employee needs to provide required details like, name, phone-no, address(location), user_id, password

1.User_id: - user_id is a primary key use to identify employees of company each employee has unique id.

2.Name: -Name is composite attribute further divided into first name and last name.

3.Phone-no: - phone-no is multivalued attribute use to store phone-no.

4.Address: -Is a composite attribute use to fetch the location of user which is further divided into state, city, pin code.

5.Password: -To validate the user at the time of login.

6.Email id:-Email is use to verify and send details.

7.Username:-Username is use for login user.

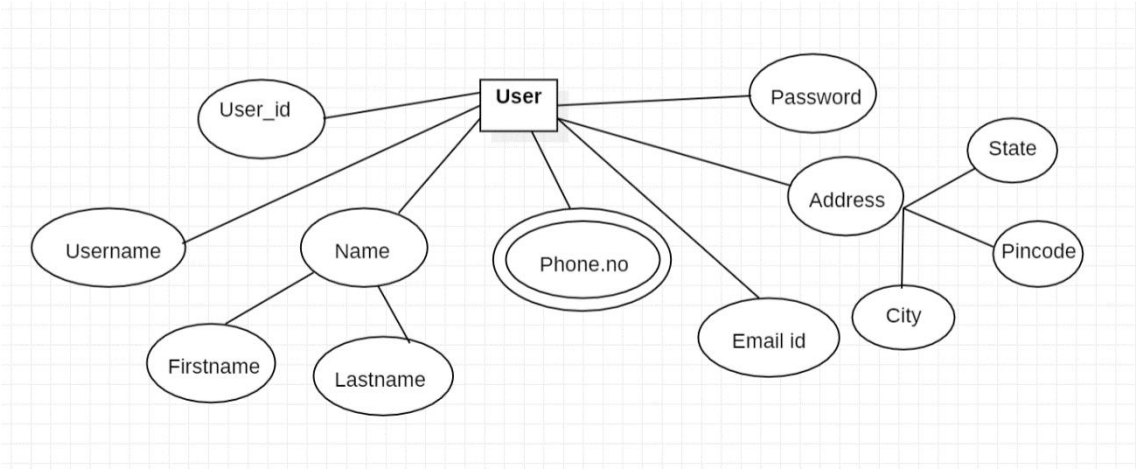


Fig 3.4 User entity set

- **Booking cab: -** Is for booking cab for a user to travel from home to office.

1.Employee name: -Is for who is going to travel from cab.

2.Booking_id: -Booking_id is unique for each cab booked.

3.Location: -To fetch the location of employee for pickup and drop.

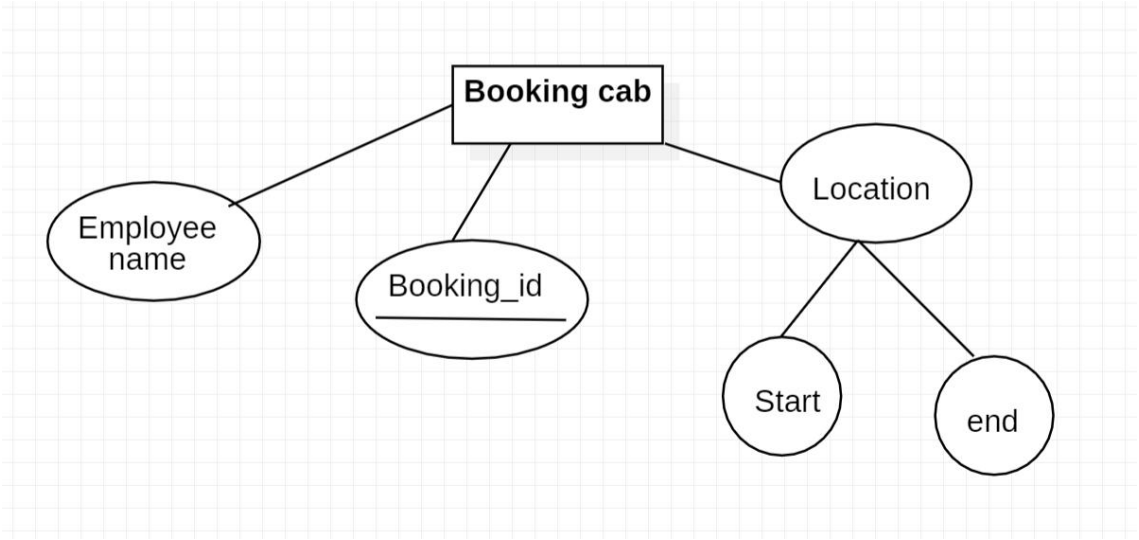


Fig 3.5 Booking cab entity set

- **Cab: -**Is for storing details of cab.

1.Vehicle_no: -It is primary key each vehicle has unique no.

2. Vehicle name: -It is name and model of vehicle.

3. Vehicle permit: -It is to check the vehicle has permit for public transportation.

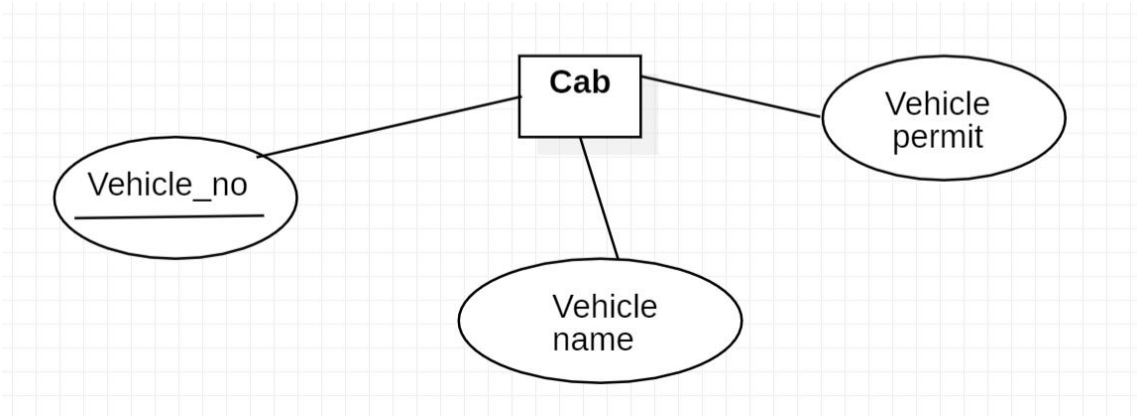


Fig 3.6 Cab entity set

- **Driver: -** Is for storing details of driver.

1.Driver_id: -It is primary key each driver has his unique id.

2.Driver name: -It is composite attribute which is further divide into first name, last name.

3.Driver phone.no: -It is multivalued attribute for contacting drivers.

4.Driver licence: -Is it have authority to drive vehicle.

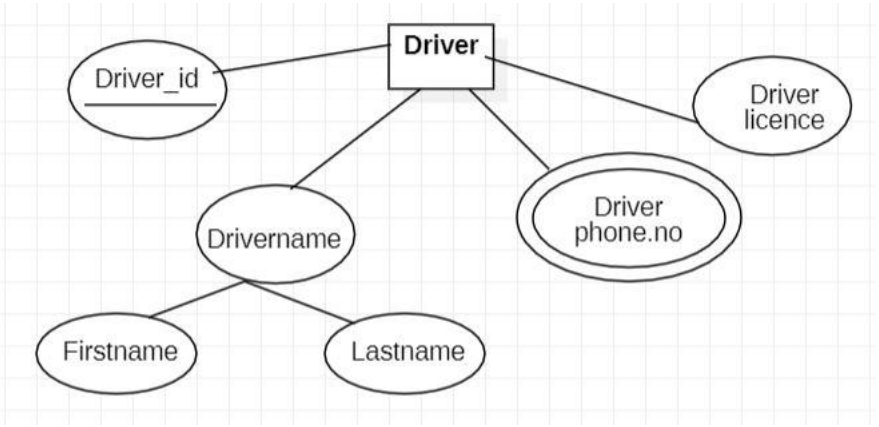


Fig 3.7 Driver entity set

- **Relationship sets:**

1.Admin manages user: -Admin can manage all the users there is many to many relations between them there are 2-3 admins they can handle all the users.

Fig 3.8 Manages relationship set

2.User needs boking cab: - User can book cab as per their location there is one to many relations between them there is 1user but he cabs bookings are many.

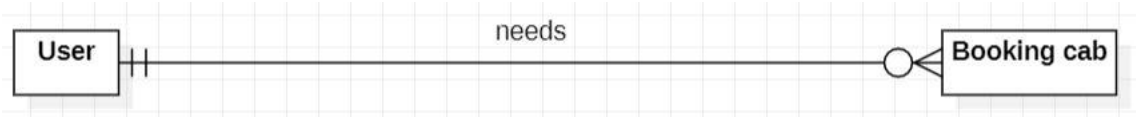


Fig 3.9 Needs relationship set

3.Cab has driver: -Every cab needs a driver there is many to many relations between them there is many cars any driver can drive any car.

Fig 3.10 Has relationship set

❖ **E-R diagram: -**

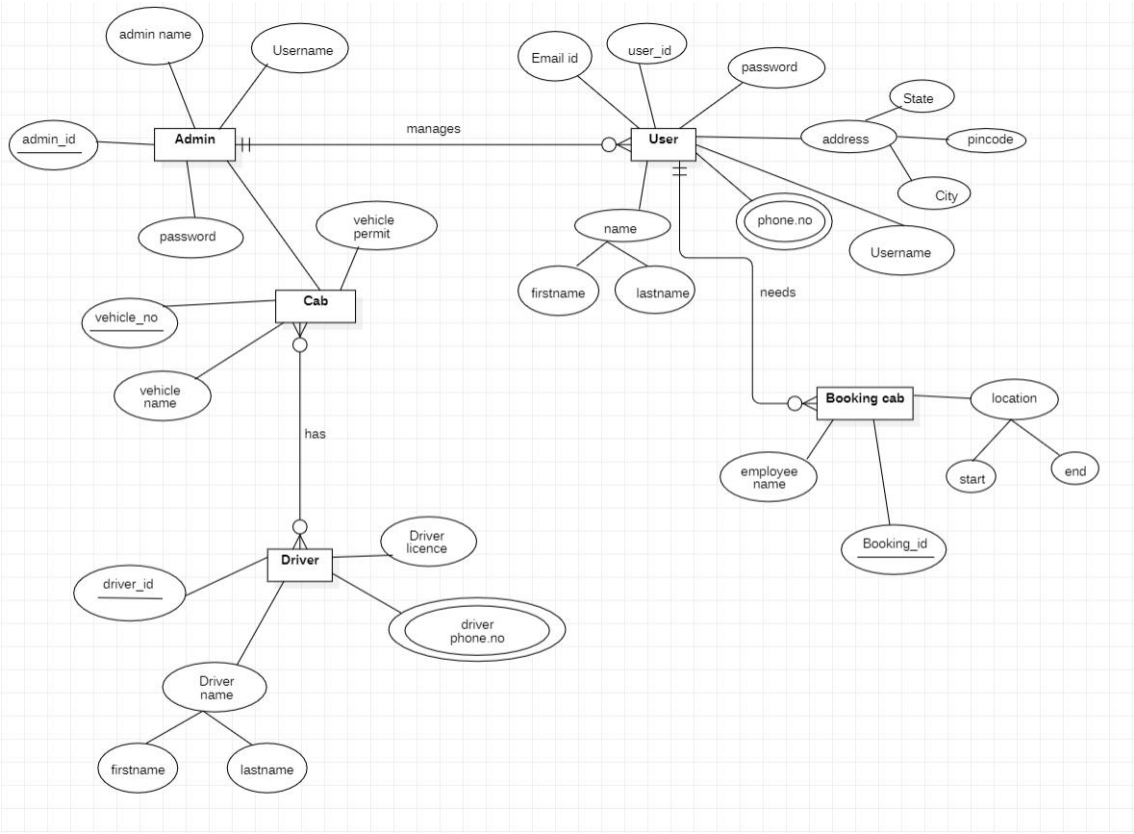
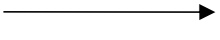


Fig 3.11 E-R diagram

3.5.2 Schema diagram: -

A database schema is the skeleton structure that represents the logical view of the entire database. It defines how the data is organised and how the relations among them are associated. It formulates all the constraints that are to be applied on the data.

Symbols: -

Name	Symbol	Description
Table		A table is a collection of related data held in table format within a database.
Relation		In a relational database system, a one-to-one table relationship links two tables based on a Primary Key column in the child which is also a Foreign Key referencing the Primary Key of the parent table row. Therefore, we can say that the

		child table share the Primary Key with the parent tables.

Table 3.3 Symbols for schema diagram

Reference: Database System and Concepts, A Silberschatz, H Korth, S Sudarshan, McGraw-Hill, Fifth Edition.

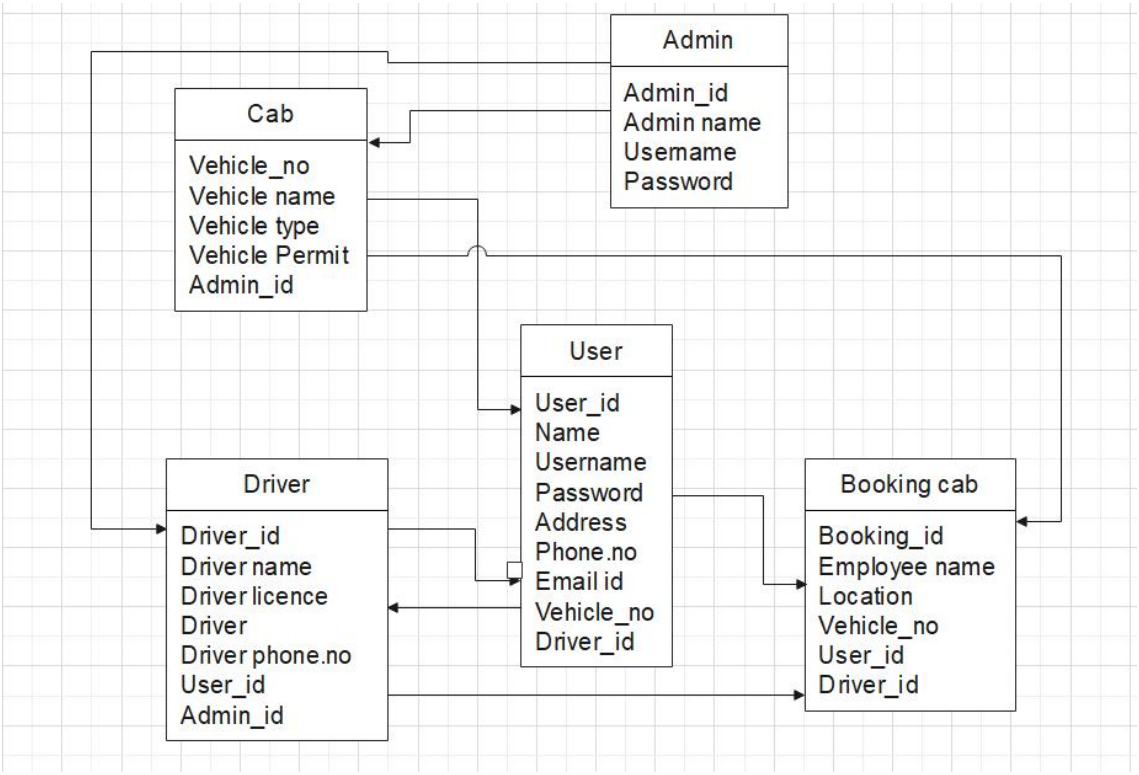


Fig 3.12 Schema diagram

3.5.3 Data flow diagram: -

A Data Flow Diagram (DFD) shows what kinds of data will be input to and output from the system, where the data will come from and go to, and where the data will be stored. It does not show information about the timing of processes, or information about whether processes will operate in sequence or in parallel.

Symbols: -

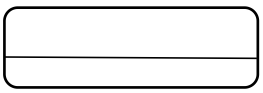
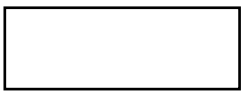
Name	Symbol	Description
Process		A process transforms incoming data flow into outgoing data flow.
External Entity		External entities are objects outside the system, with which the system communicates
Data Flow		Data flows are pipelines through which packets of information flow. Label the arrows with the name of the data that moves through it.
Data Store		Data stores are repositories of data in the system.

Table 3.4 Symbols for DFD

Reference: Software Engineering, edition, Ian Somerville Pearson Education. Ninth Object – Oriented Modelling and Design Michael Blaha, James Rumbaugh Pearson 2011

Level 0 DFD (Context level):

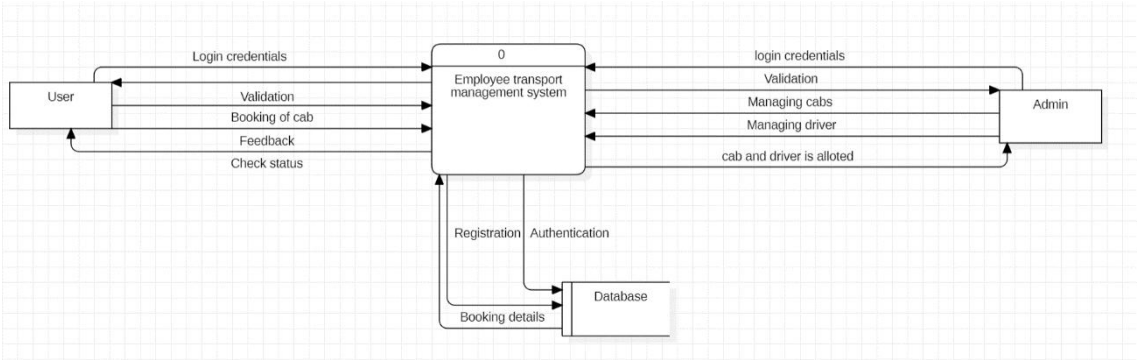


Fig 3.13 Level 0 DFD

Level 1 DFD:

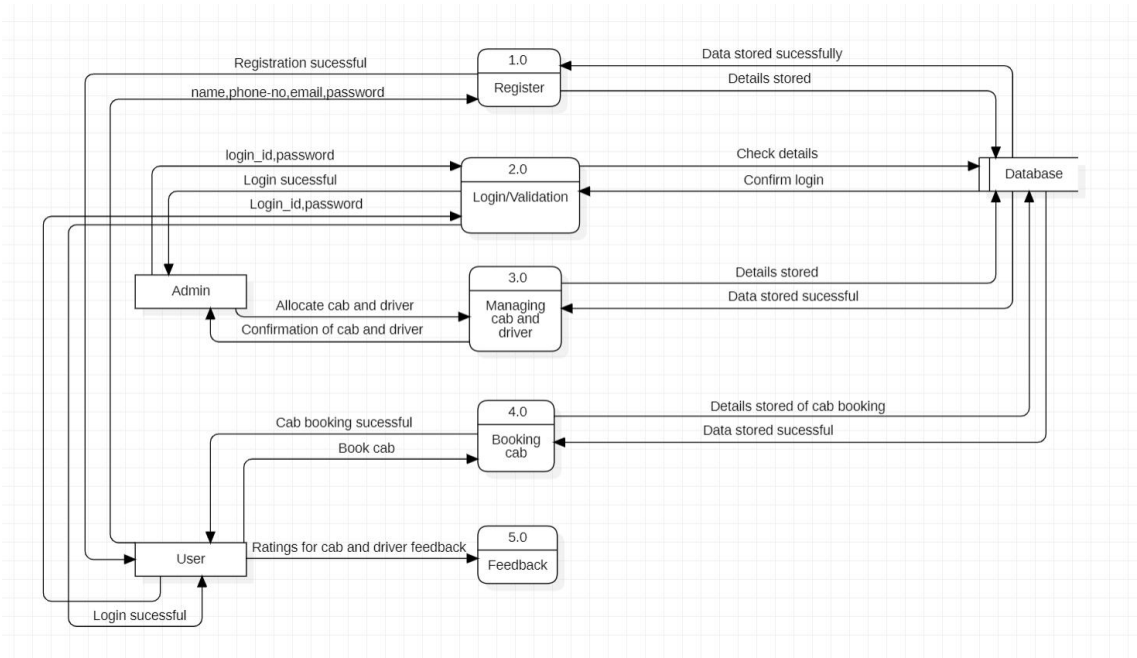


Fig 3.14 Level 1 DFD

Level 2 DFD:

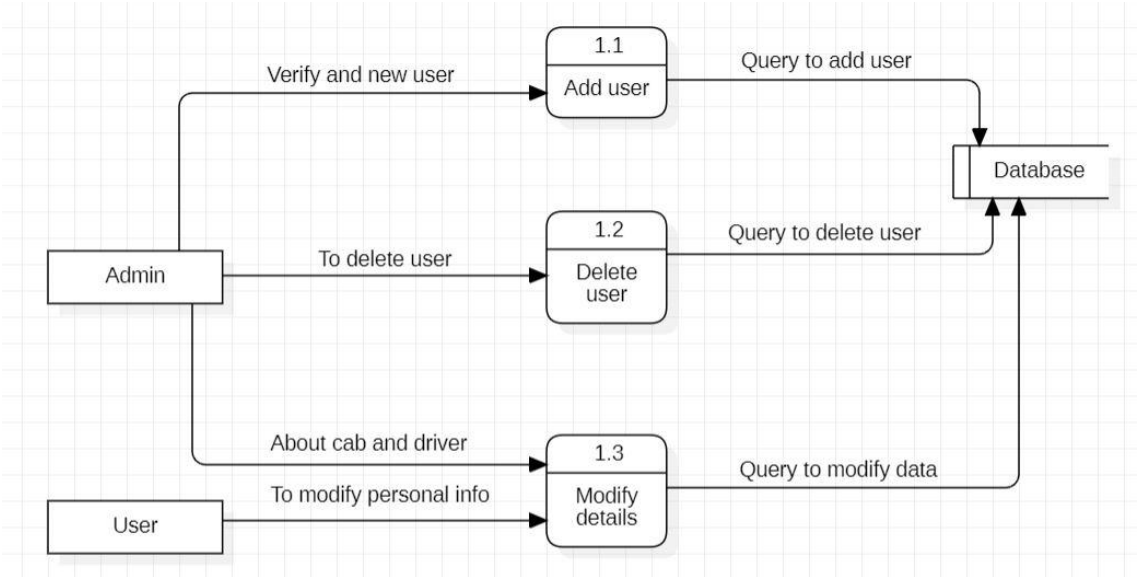


Fig 3.15 Level 2 DFD

3.5.4 Class diagram:

Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application. Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modelling of object-

oriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages.

Symbols: -

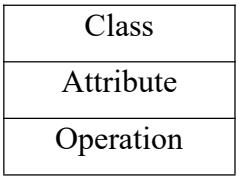
Name	Symbol	Description
Class		Classes and interfaces in UML show architecture and features of the designed system.
Association		Represents the static relationship shared among the objects of two classes.

Table 3.5 Symbols for class diagram

Reference: Software Engineering, edition, Ian Somerville Pearson Education. Ninth Object – Oriented Modelling and Design Michael Blaha, James Rumbaugh Pearson 2011

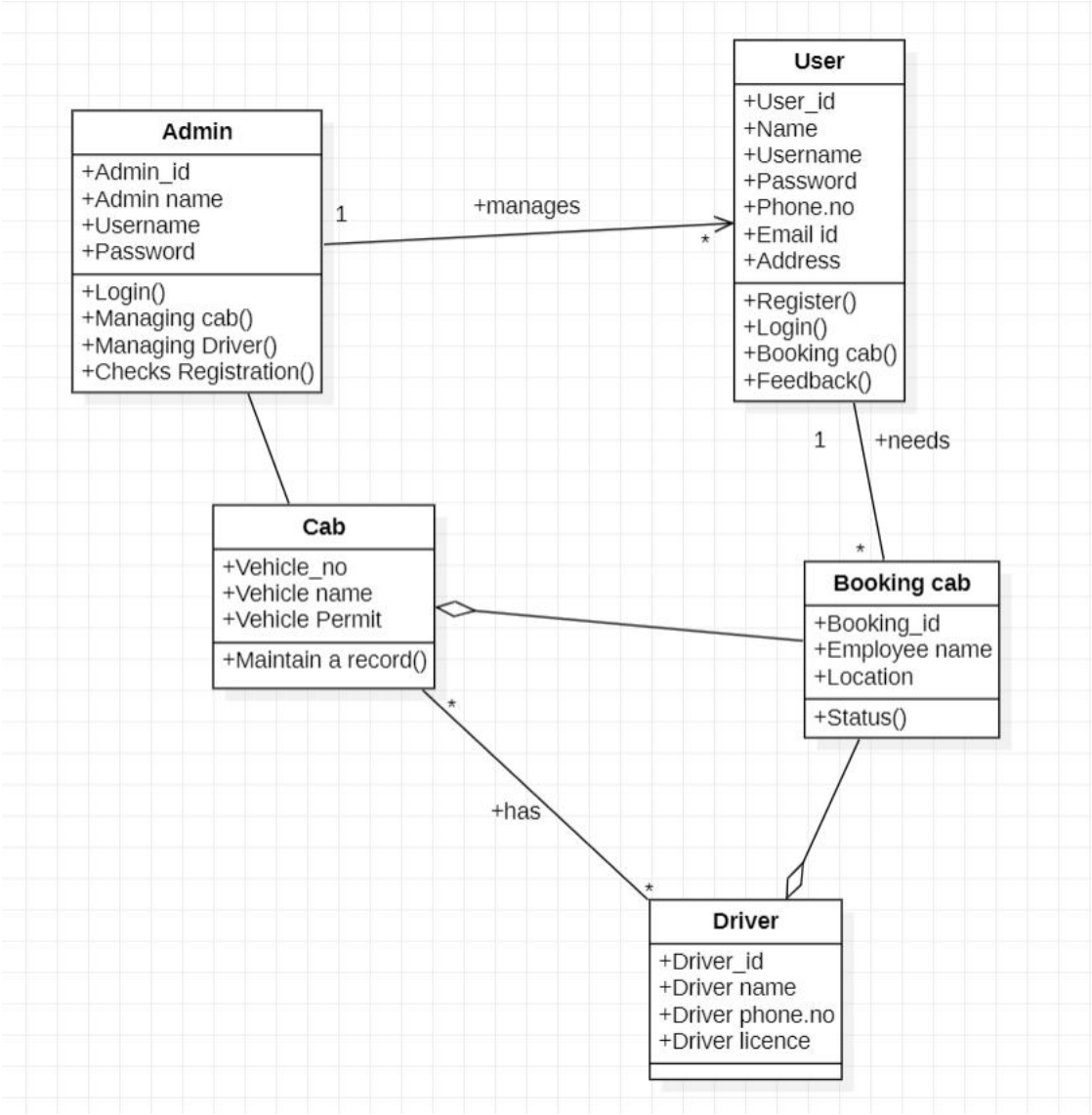


Fig 3.16 Class diagram

3.5.5 Use case diagram: -

A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well. The use cases are represented by either circles or ellipses.

Symbols: -

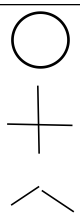
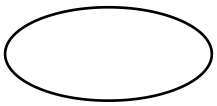
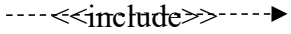
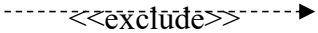
Name	Symbol	Description
Actor		Actor represents a user or another system that will interact with the system you are modelling.
Use case		A use case is an external view of the system that represents some action the user might perform in order to complete a task.
Association		An association is use to show interaction of actors with use cases.
Include		This association states that the base use case is executed with the help of include use case.
Exclude		The extend states that the extend use case will be executed after the execution of base use case but it will not always be executed.

Table 3.6 Symbols for use case diagram

Reference: Software Engineering, edition, Ian Somerville Pearson Education. Ninth Object – Oriented Modelling and Design Michael Blaha, James Rumbaugh Pearson 2011

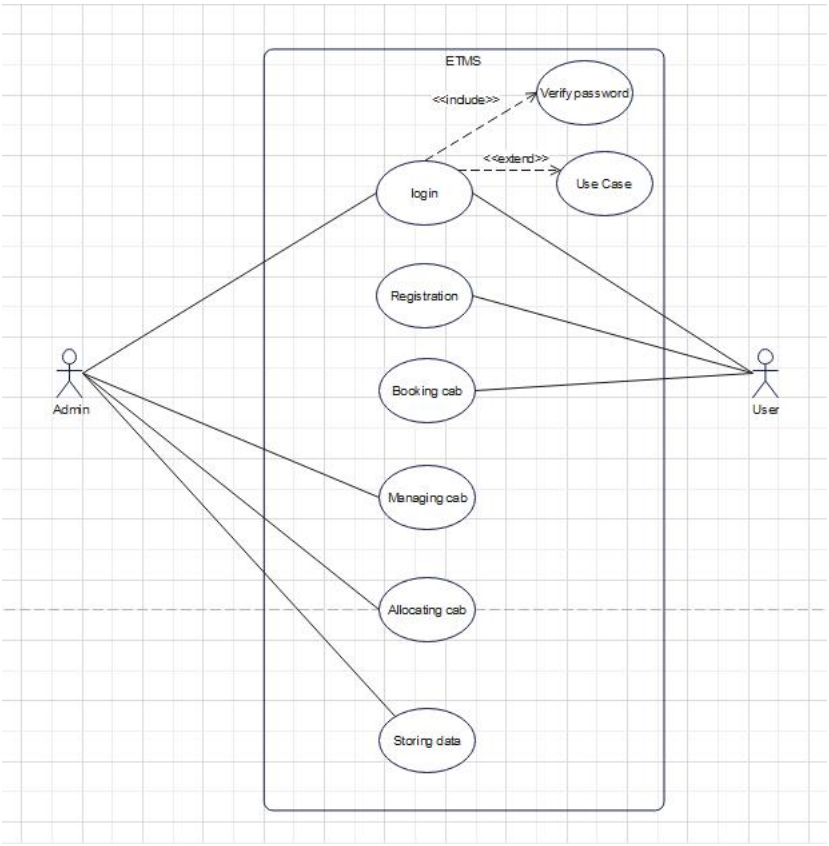


Fig 3.17 Use case diagram

Scenarios: -

1.Usecase: -Login

- Description: -To login in to their respective account.
- Actor: -User, Admin
- Pre-condition: -To provide login credentials.
- Post-condition: -Details are verified and login is successful.

2.Usecase: -Registration

- Description: -To register on the system.
- Actor: -User
- Pre-condition: -To give all the details about himself.
- Post-condition: -The registration will be successful.

3.Usecase: -Booking cab

- Description: -To book the cab.
- Actor: -User
- Pre-condition: -To check availability of cab and put request to book.
- Post-condition: -You will get notification when cab is book.

4.Usecase: -Managing cabs

- Description: -To check availability of cabs.
- Actor: -Admin
- Pre-condition: -To admin check the availability and feed in system and allocation is done for user.
- Post-condition: -Cab managed successfully.

5.Usecase: -Allocating driver

- Description: -To manage the drivers according to the cabs.
- Actor: -Admin
- Pre-condition: -To allocate the driver to particular car.
- Post-condition: -The driver is allocated for cab.

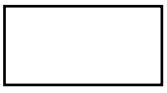
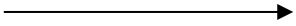
6.Usecase: -Storing data

- Description: -To store all the data in system database.
- Actor: -Admin
- Pre-condition: -To store each and every data related to users and bookings.
- Post-condition: -data stored successfully.

3.5.6 Sequence diagram:

A sequence diagram in a Unified Modelling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams typically are associated with use case realizations in the Logical View of the system under development.

Symbols: -

Name	Symbol	Description
Object		An object that is created, performs actions, and/or is destroyed during the lifeline
Synchronous message		An instantaneous communication between objects that conveys

		information, with the expectation that an action will be initiated as a result.
Activation box		The period during which an object is performing an action.

Table 3.7 Symbols for Sequence diagram

Reference: Software Engineering, edition, Ian Somerville Pearson Education. Ninth Object – Oriented Modelling and Design Michael Blaha, James Rumbaugh Pearson 2011

1.Login: -

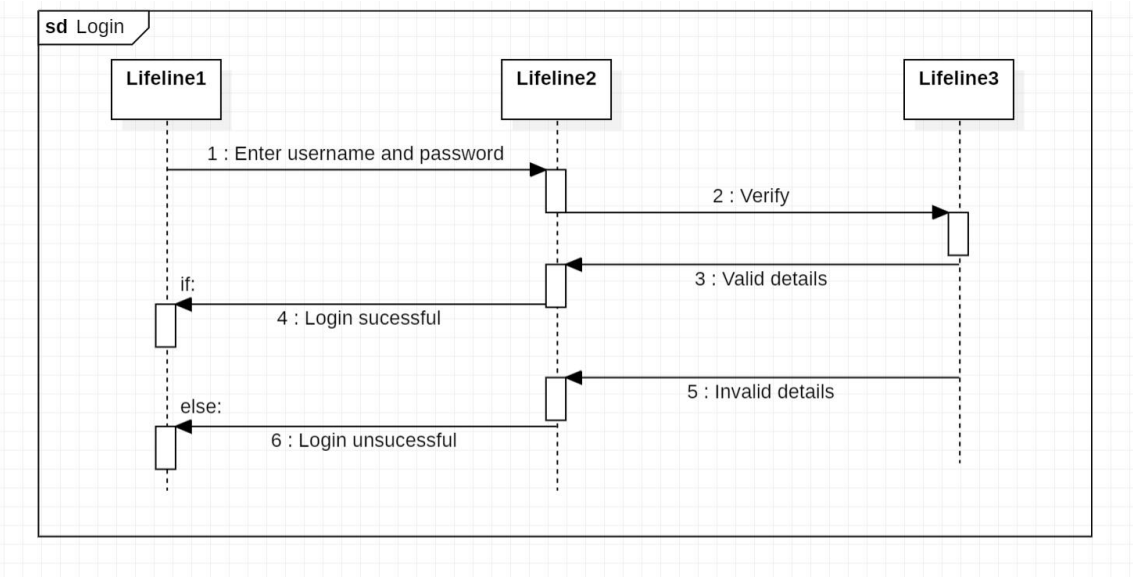


Fig 3.18 Sequence diagram for login

2.Registration: -

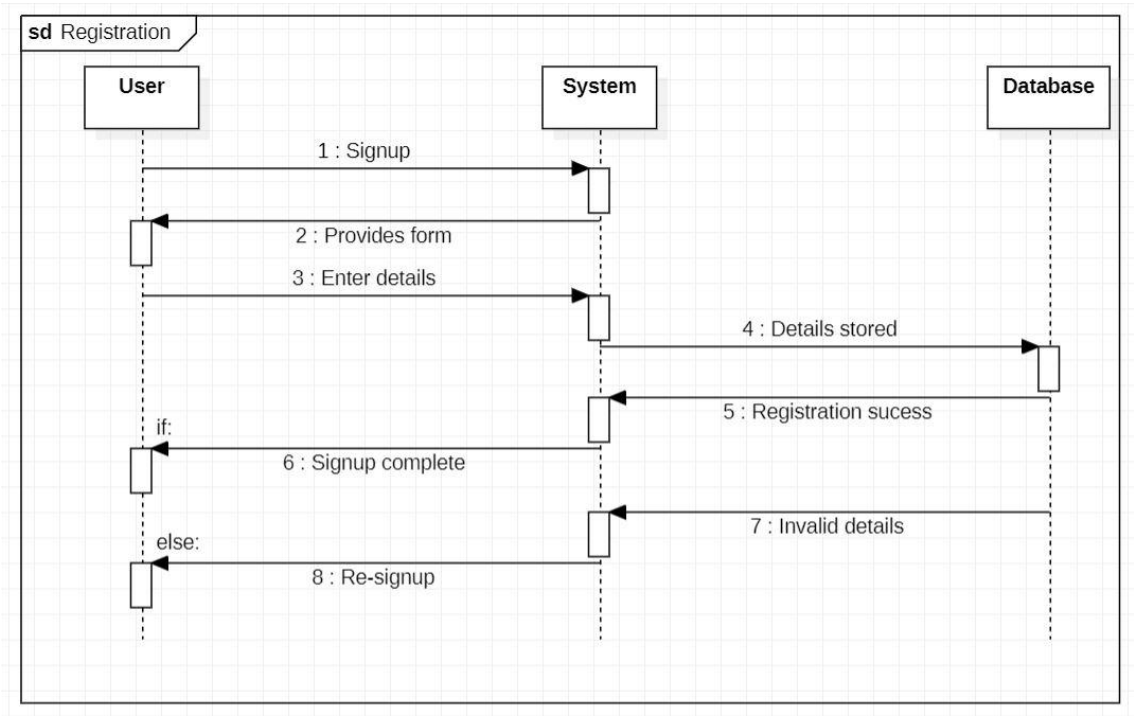


Fig 3.19 Sequence Diagram for registration

3.Booking cab: -

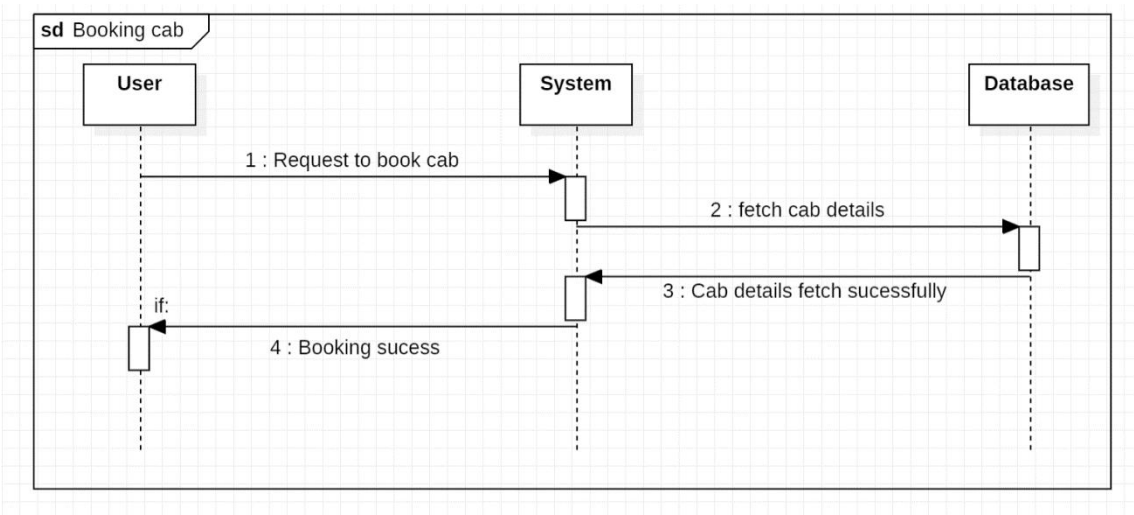


Fig 3.20 Sequence Diagram for booking cab

4.Fetching location: -

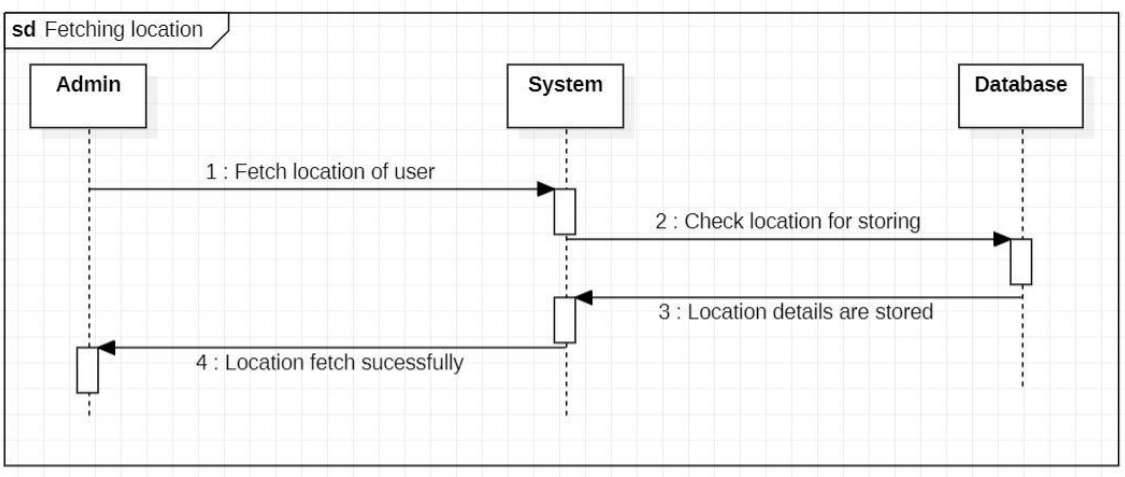


Fig 3.21 Sequence diagram for fetching location

5.Managing cab and driver: -

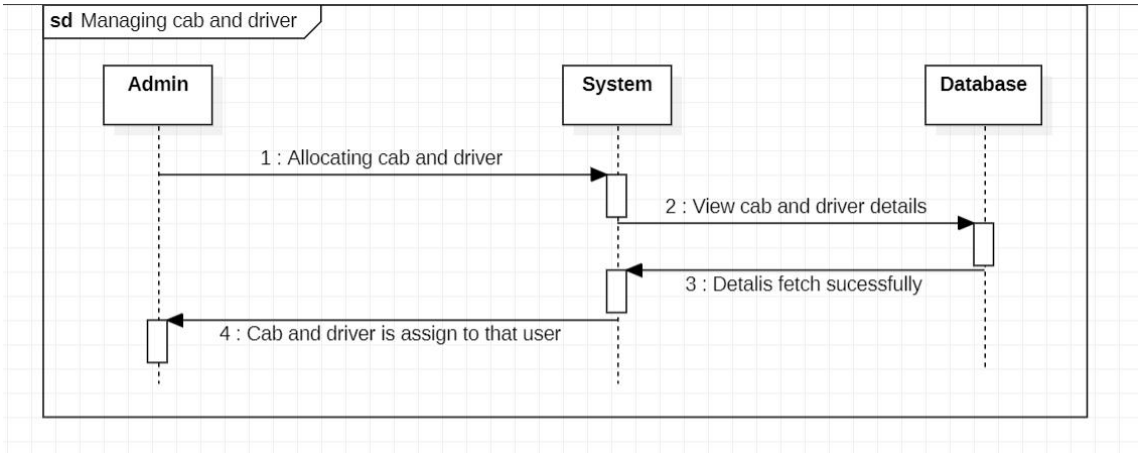


Fig 3.22 Sequence diagram for managing cab and driver

6.Storing data: -

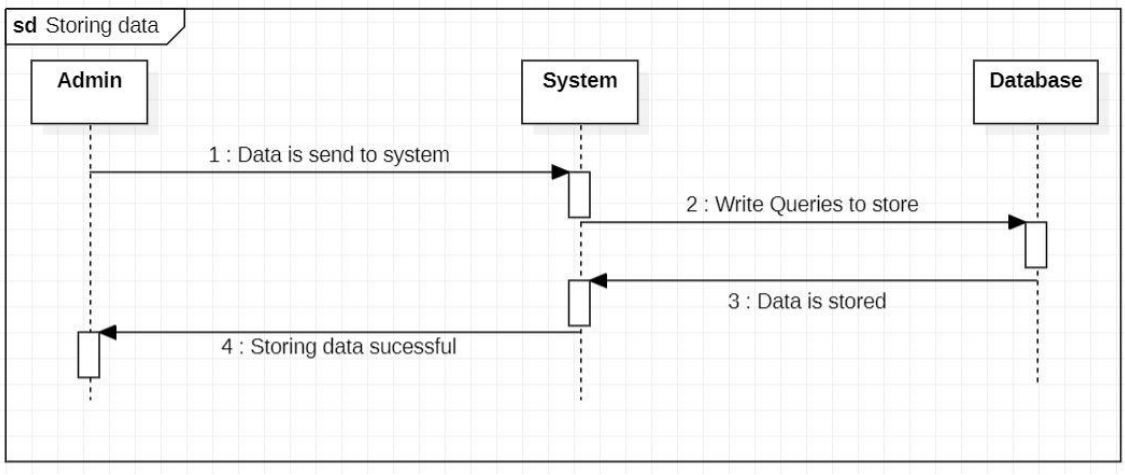


Fig 3.23 sequence diagram for storing data

3.5.7 Activity diagram: -

Activity diagram is another important diagram in UML to describe the dynamic aspects of the system. Activity diagram 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. Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc.

Symbols: -

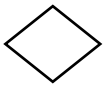
Name	Symbol	Description
Initial state		This shows the starting point or first activity of the flow.
Final state		The end of the Activity diagram, also called as a final activity.
Action		It represents the activity to be performed.
Decision		A logic where a decision is to be made is depicted by a diamond.
Transition		A transition link represents control flow between nodes.

Table 3.8 Symbols for activity diagram

Reference: Software Engineering, edition, Ian Somerville Pearson Education. Ninth Object – Oriented Modelling and Design Michael Blaha, James Rumbaugh Pearson 2011

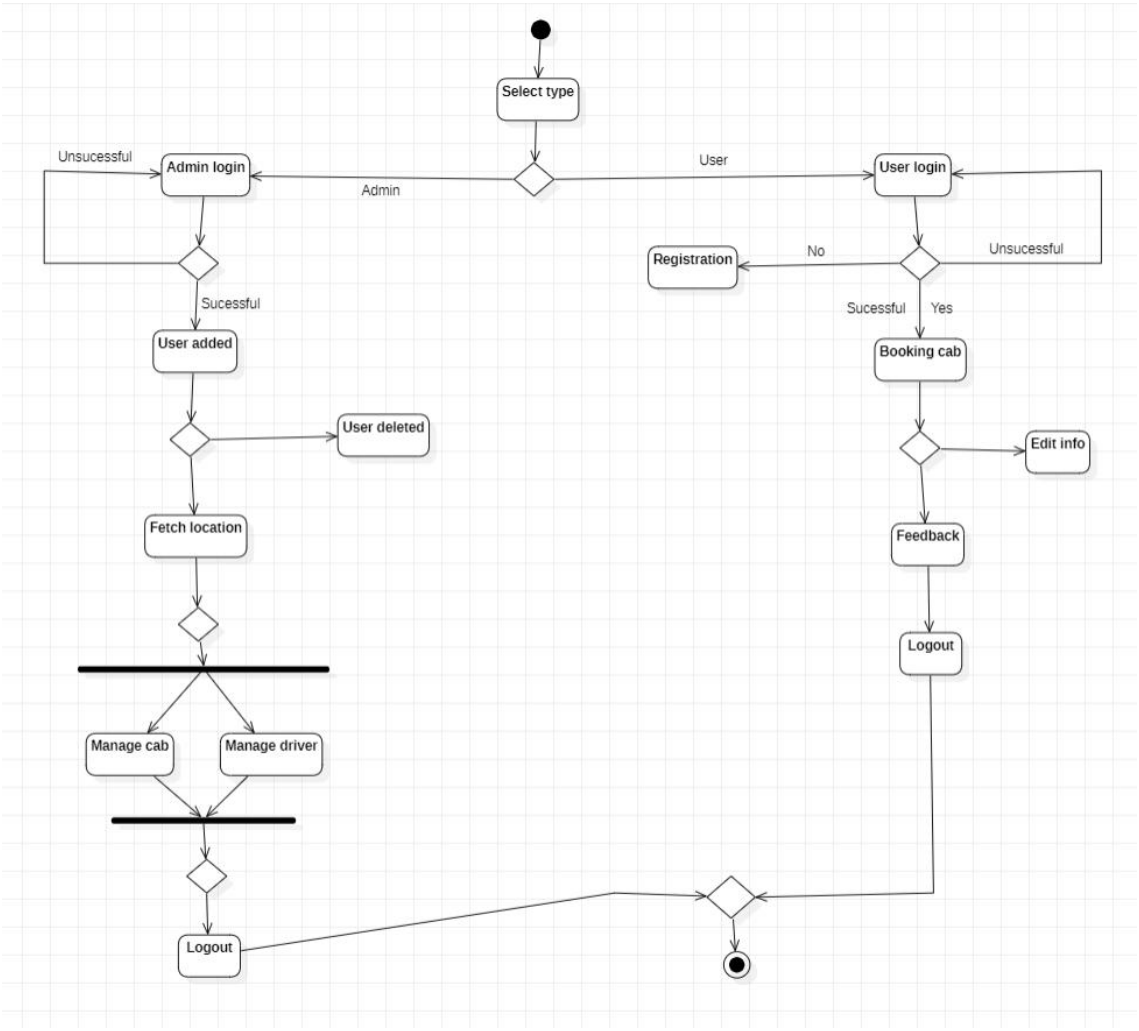


Fig 3.24 Activity diagram

3.5.8 State chart diagram:

A state diagram is used to represent the condition of the system or part of the system at finite instances of time. It's a behavioural diagram and it represents the behaviour using finite state transitions. State diagrams are also referred to as State machines and State-chart Diagrams. These terms are often used interchangeably. So simply, a state diagram is used to model the dynamic behaviour of a class in response to time and changing external stimuli.

Symbols:

Name	Symbol	Description
Initial state		This represents the starting of the state diagram.
Final state		This represents the final state or end of the state

		diagram.
Transition		This represents the change of one state into another state.
State		This represents the state of the activity.

Table 3.9 Symbols for state chart diagram

Reference: Software Engineering, edition, Ian Somerville Pearson Education. Ninth Object – Oriented Modelling and Design Michael Blaha, James Rumbaugh Pearson 2011

1.login: -

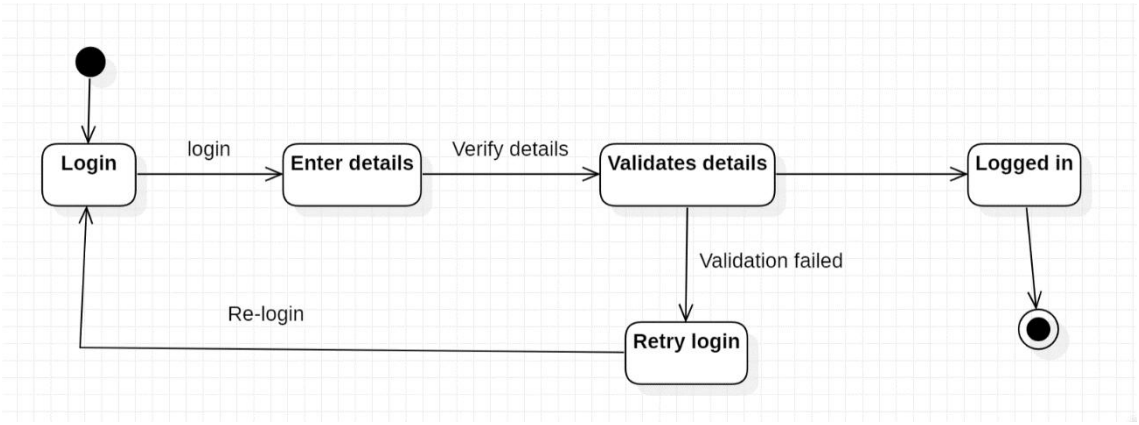


Fig 3.25 State chart diagram for login

2.Registration: -

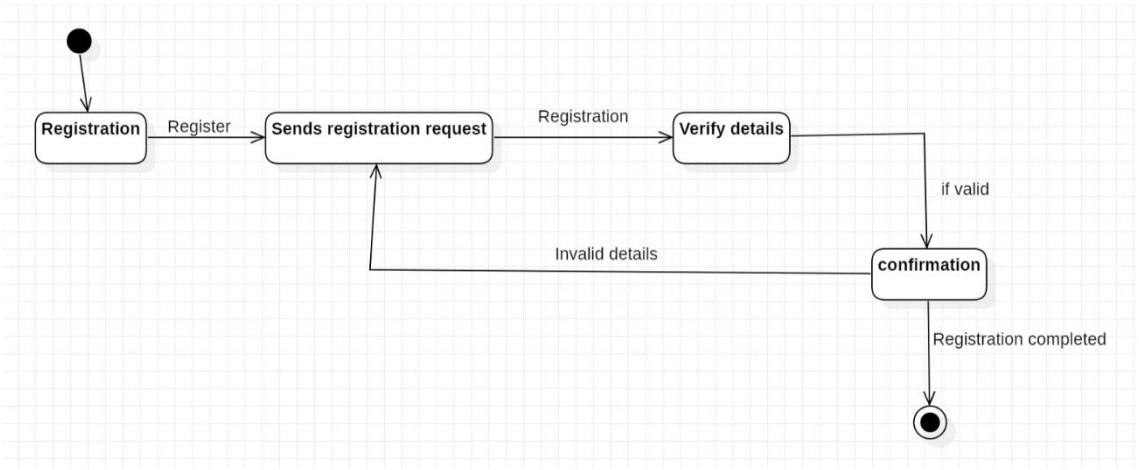


Fig 3.26 State chart diagram for registration

3.Booking cab: -

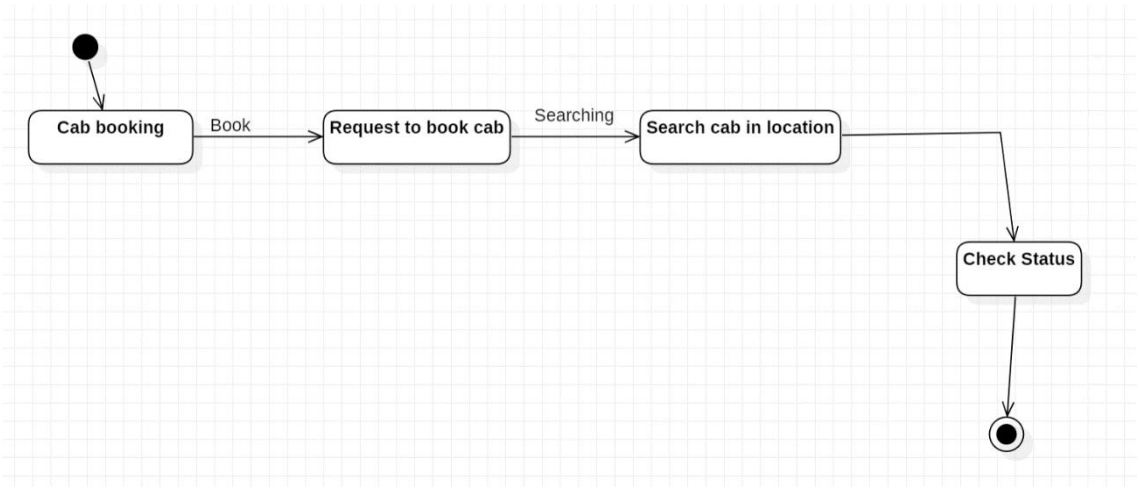


Fig 3.27 state chart diagram for booking cab

CHAPTER 4:

SYSTEM DESIGN

4.1 User interface design: -

User login page: -

Fig 4.1 User login UI

Registration page: -

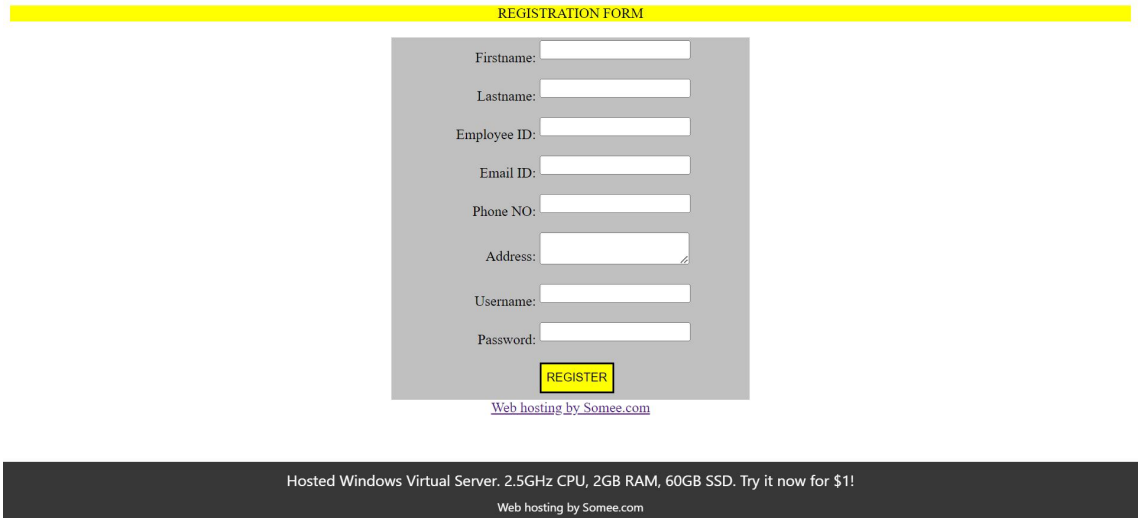


Fig 4.2 Registration UI

Home page: -

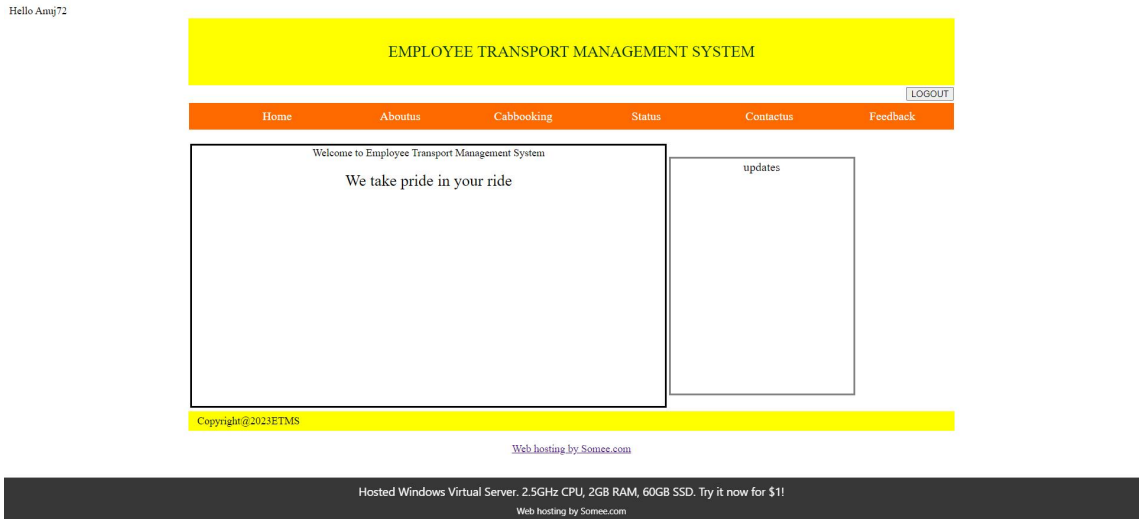


Fig 4.3 Homepage UI

About us page:-

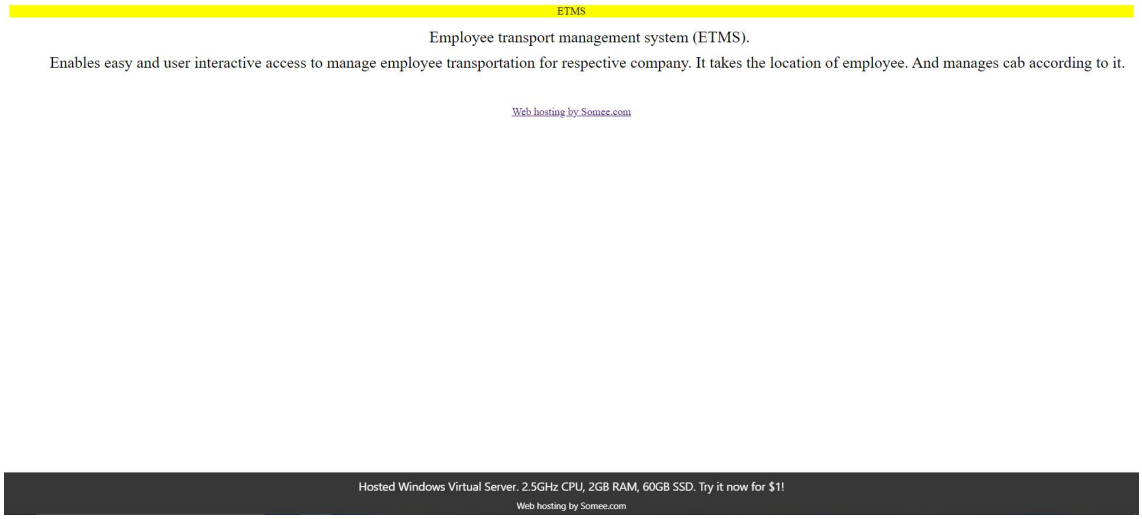


Fig 4.4 About us UI

Cab booking page: -

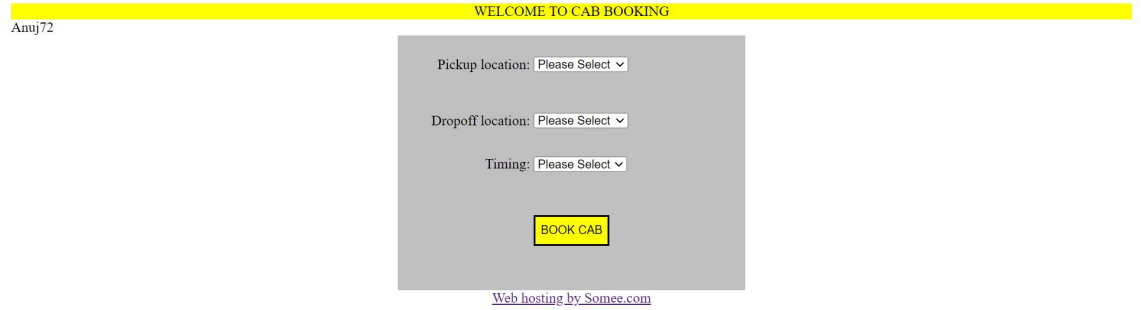


Fig 4.5 Cab booking UI

Status page:-

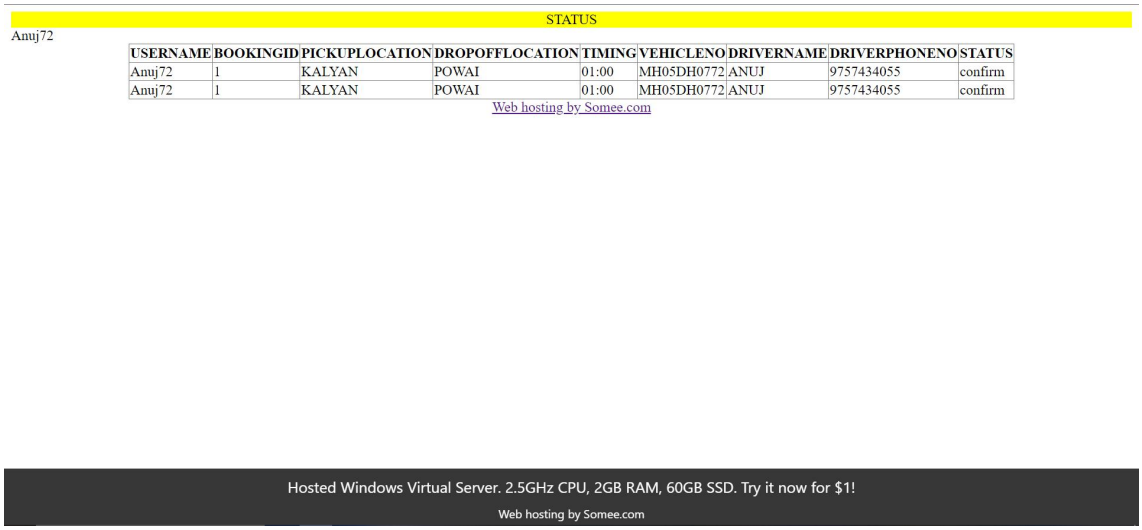


Fig 4.6 Status UI

Contact us page: -

Fig 4.7 Contact us UI

Feedback page: -

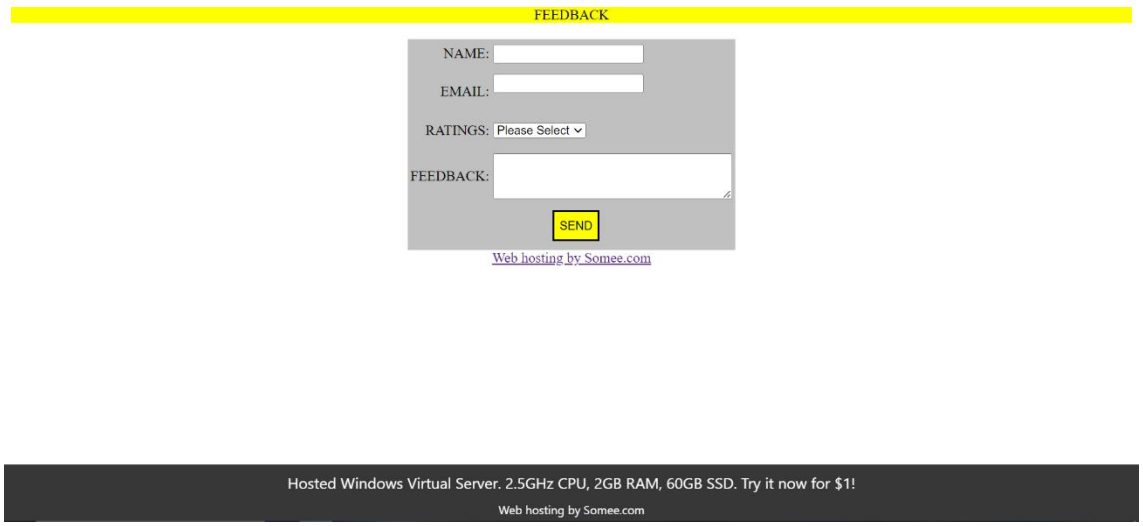


Fig 4.8 Feedback UI

Admin login page: -

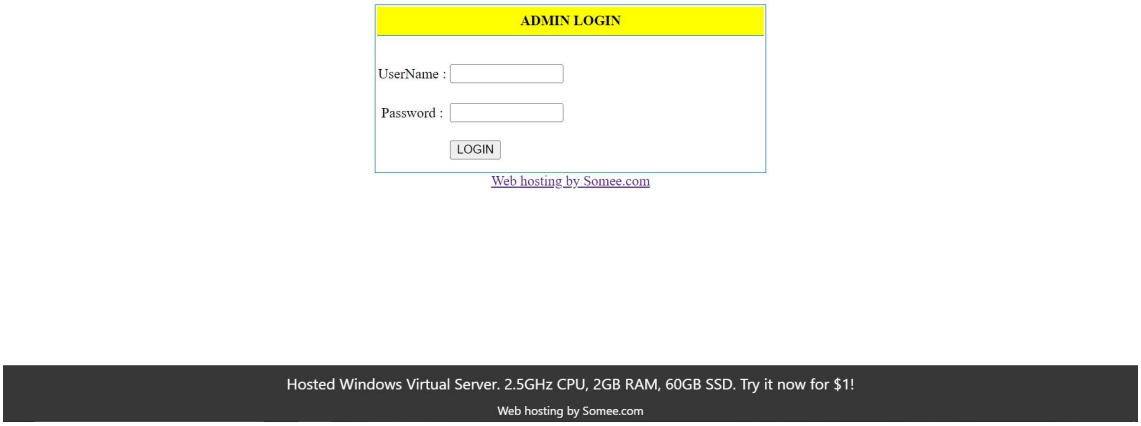


Fig 4.9 Admin login UI

Admin homepage:-

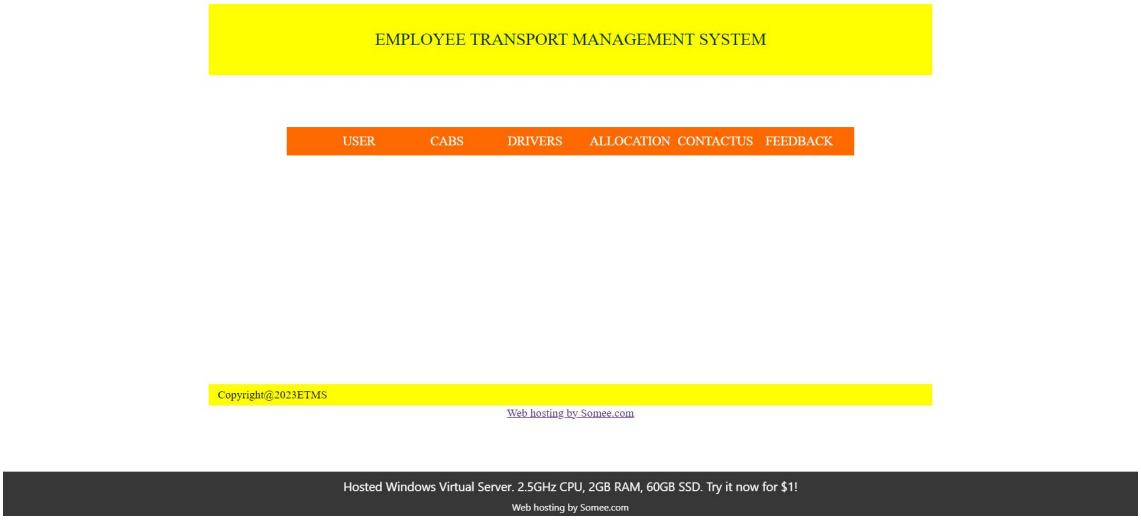


Fig 4.10 Admin homepage UI

Managing users page:-

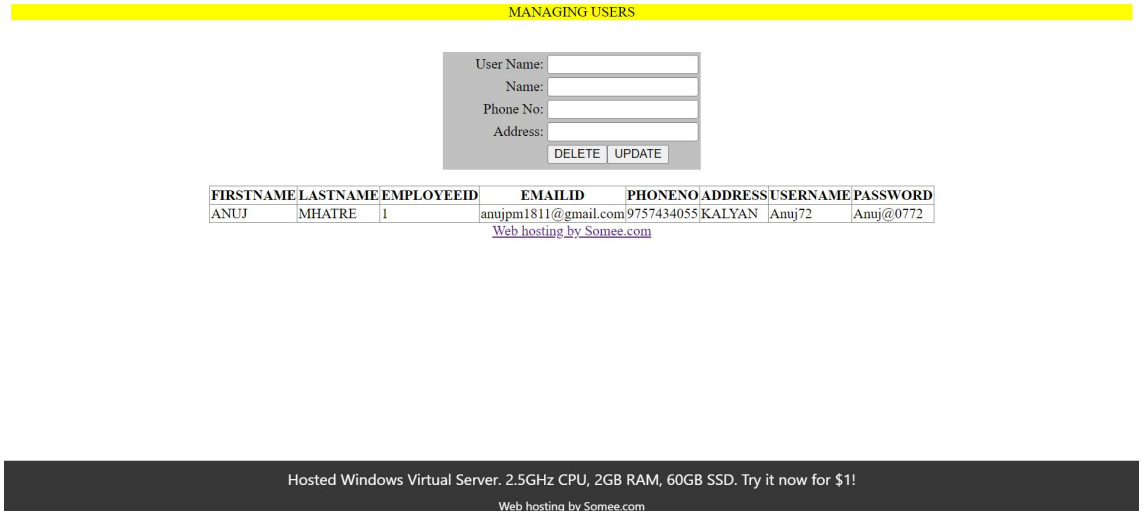


Fig 4.11 Managing users UI

Managing cabs page:-

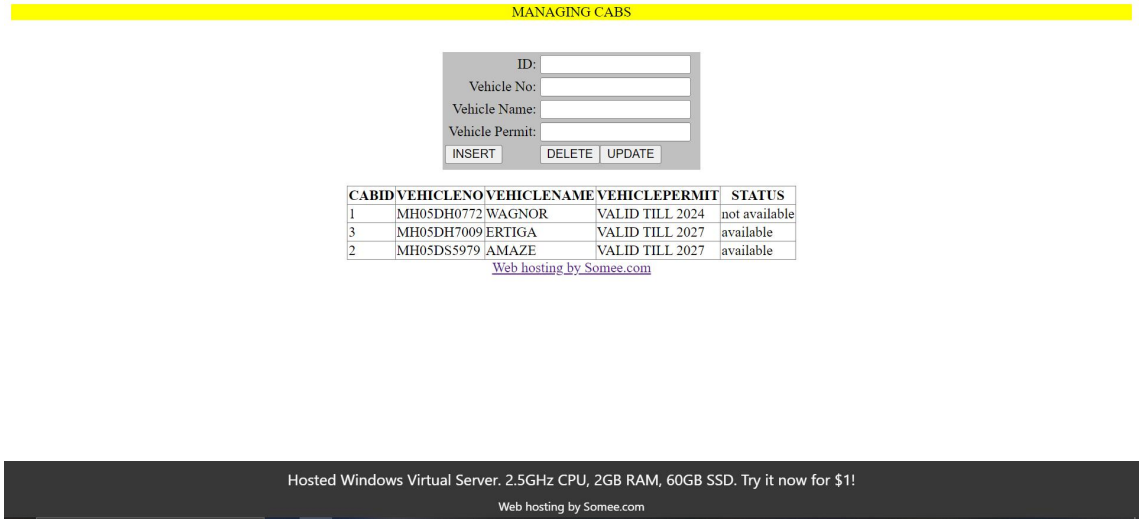


Fig 4.12 Managing cabs UI

Managing drivers page:-

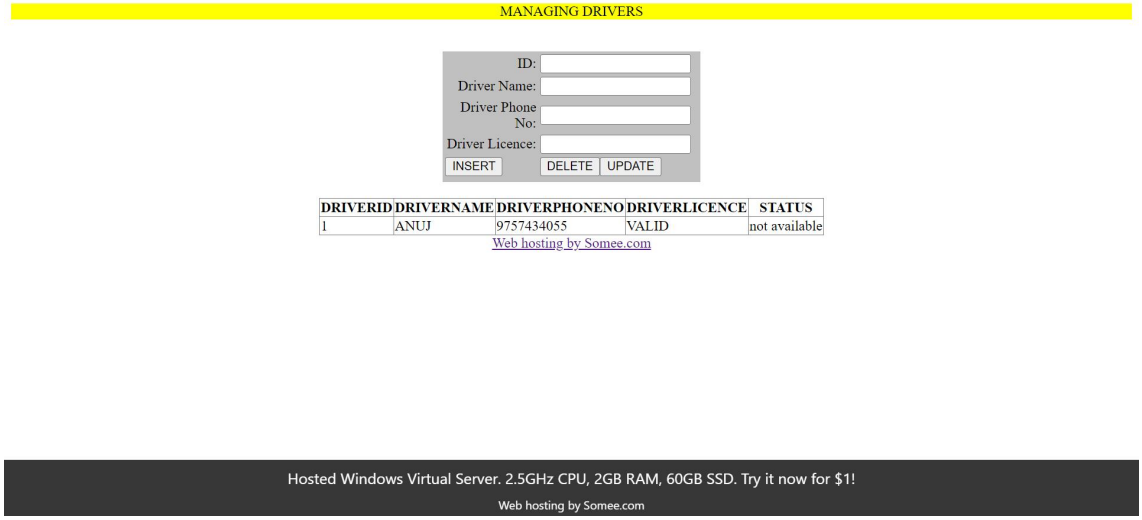


Fig 4.13 Managing drivers UI

Confirm booking page:-

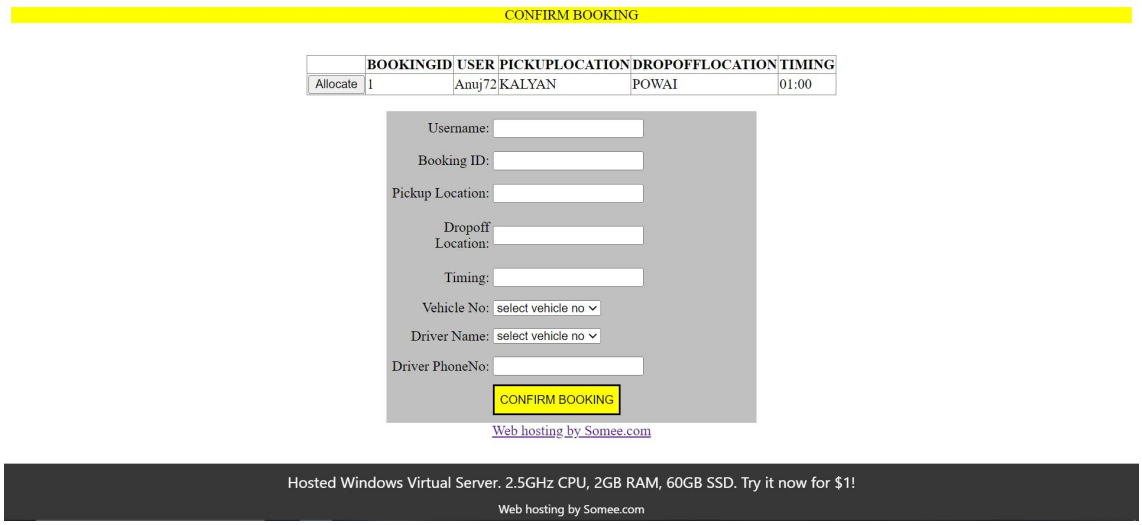
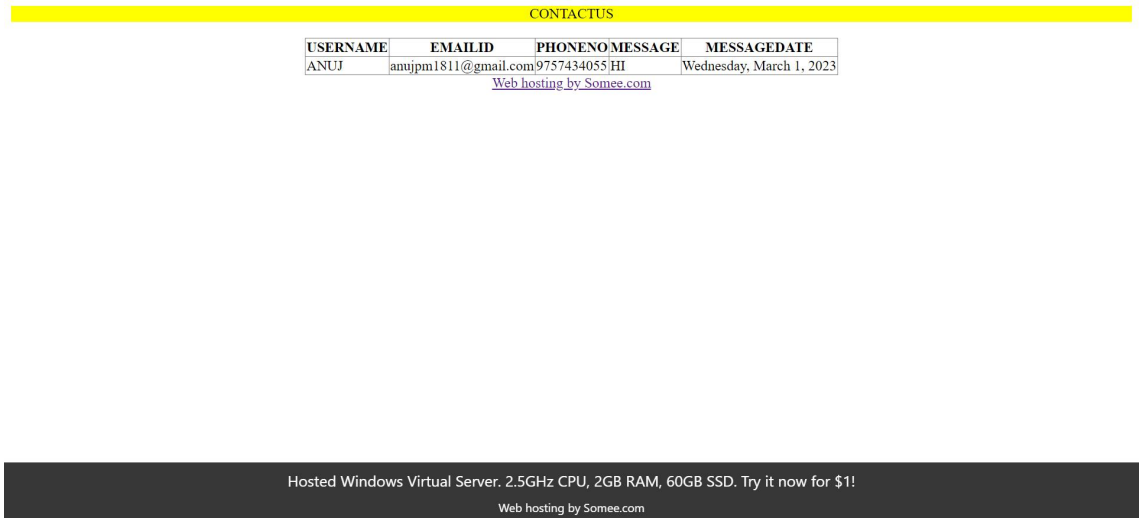


Fig 4.14 Confirm booking UI

Managing contactus page:-


CONTACTUS

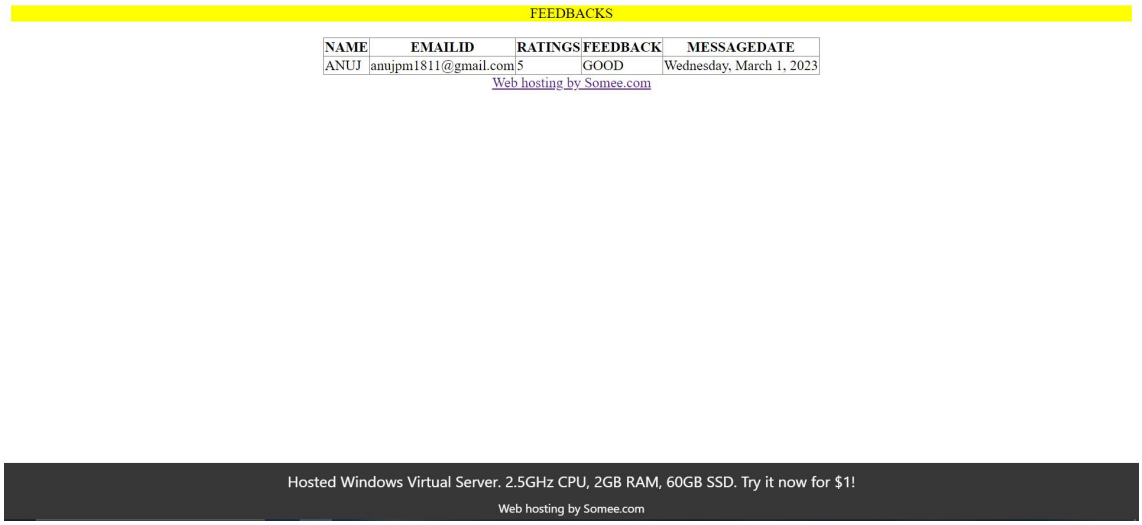
USERNAME	EMAILID	PHONENO	MESSAGE	MESSAGEDATE
ANUJ	anujpm1811@gmail.com	9757434055	HI	Wednesday, March 1, 2023

Web hosting by Somee.com

Hosted Windows Virtual Server. 2.5GHz CPU, 2GB RAM, 60GB SSD. Try it now for $1!
Web hosting by Somee.com

Fig 4.15 Managing contactus UI

Managing feedback page:-


FEEDBACKS

NAME	EMAILID	RATINGS	FEEDBACK	MESSAGEDATE
ANUJ	anujpm1811@gmail.com	5	GOOD	Wednesday, March 1, 2023

Web hosting by Somee.com

Hosted Windows Virtual Server. 2.5GHz CPU, 2GB RAM, 60GB SSD. Try it now for $1!
Web hosting by Somee.com

Fig 4.16 Managing feedback UI

4.2 Test Cases Design: -

Test case no.	Test case name	Expected output	Actual output	Remark
1.	Register: Write details	Registered successfully		
2.	Register: Wrong details	Please enter valid details		
3.	Login: Valid details	Login successful		
4.	Login: If user not registered	User is not registered		

5.	Login: Wrong username	Please enter valid user name		
6.	Login: wrong Password	Password incorrect! please enter again		
7.	Modify Users: To update details	User modified successfully		
8.	Delete Users: User who have left that company	User deleted successfully		
9.	Add cab: New cab	Cab added successfully		
10.	Update cab: To update details like permit	Cab details modified successfully		
11.	Delete cab: Cab which is not in use	Cab deleted successfully		
12.	Add driver: New driver	Driver details added successfully		
13.	Update Driver: Update details	Driver details modified successfully		
14.	Delete driver: Driver which is not working	Driver deleted successfully		
15.	Booking cab: To book cab	Cab booking successfully		
16.	Contact us	Complaint send successfully		
17.	Contact us	Complaint not send successfully		
18.	Feedback	Ratings has sent		

		successfully		
19.	Feedback	Ratings not sent to system		
20.	Confirm booking	Booking confirm		

Table 4.1 Test cases design

CHAPTER 5: IMPLEMENTATI0N AND TESTING

5.1 Implementation Approaches:-

This project was implemented using the Incremental model. Incremental Model is a process of software development where requirements divided into multiple standalone modules of the software development cycle. In this model, each module goes through the requirements, design, implementation and testing phases. Every subsequent release of the module adds function to the previous release. The process continues until the complete system achieved.

The best part about incremental model is that, we can develop a raw system and then upgrade it slowly step by step into required system. If some defects or error arises, then according to that, we can do some changes and modification to the system without any difficulties. Implementation of the project was majorly carried out on Visual Studio 2010. The requirements were analyzed and thus began the implementation of the project with creating proper user interfaces on visual studio. The interfaces were designed and created using Visual Studio. After the user interfaces were created, database connectivity was performed. I connected my system to a SQL Server at the free web hosting site https://somee.com/. The coding part of the project was done in C# language. The project was divided into modules. These modules were created one by one and after completion of each module, unit testing was performed on that module. As soon as the module fulfilled its requirements it was integrated into the main project. After integration, each functionality was checked which can also be said to be as integration testing. After adding all the modules to the main project, the final testing was performed to check whether the system was performing properly or not.

I have used validations wherever it was required. The system is made by considering all the problems into the view and the final project should be fulfilling all the requirements.

5.2 Coding Details and CodeEfficiency:-

Coding is one of the major and important part of project development. The code should always be efficient and as minimum as possible, but we must make sure that the functionalities and reliability of the system are not compromised. User interfaces were designed in Visual Studio and main part of the coding was done using C# language as it is a convenient as well as a user-friendly language. For backend connectivity SQL was used as it is supported by SQL database. The SQL code was used to retrieve and update the data in every area where database was to be accessed

5.2.1 Coding Details:-

Cabbooking.aspx:-

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="cabbooking.aspx.cs" Inherits="CABBOOKING" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head id="Head1" runat="server">
    <title></title>
    <style type="text/css">
        .style1
        {
            width: 31%;
            background-color: #C0C0C0;
            height: 287px;
        }
        .style2
        {
            width: 846px;
            text-align: right;
        }
        .style3
        {
            width: 846px;
            height: 30px;
            text-align: right;
        }
        .style4
        {
            height: 30px;
```

```
        }
    </style>
</head>
<body>
    <form id="form1" runat="server">
    <div>
    <center style="background-color: #FFFF00">WELCOME TO CAB BOOKING</center>
        <asp:Label ID="Label1" runat="server"></asp:Label>
    <br />
    </div>
    <table align="center" class="style1">
        <tr>
            <td class="style2">
                Pickup location:</td>
            <td>
                <asp:DropDownList ID="txtpl" runat="server">
                  <asp:ListItem Value="">Please Select</asp:ListItem>
            <asp:ListItem>KALYAN</asp:ListItem>
            <asp:ListItem>THANE</asp:ListItem>
            <asp:ListItem>DOMBIVLI</asp:ListItem>
            <asp:ListItem>MULUND</asp:ListItem>
            <asp:ListItem>POWAI</asp:ListItem>
            <asp:ListItem>VIKROLI</asp:ListItem>
            <asp:ListItem>GHATKOPAR</asp:ListItem>
            <asp:ListItem>BADLAPUR</asp:ListItem>
            <asp:ListItem>AMBERNATH</asp:ListItem>
                </asp:DropDownList>
            </td>
        </tr>
        <tr>
            <td class="style2">
                Dropoff location:</td>
            <td>
```

```
        <asp:DropDownList ID="txtdl" runat="server">
          <asp:ListItem Value="">Please Select</asp:ListItem>
    <asp:ListItem>KALYAN</asp:ListItem>
    <asp:ListItem>THANE</asp:ListItem>
    <asp:ListItem>DOMBIVLI</asp:ListItem>
    <asp:ListItem>MULUND</asp:ListItem>
    <asp:ListItem>POWAI</asp:ListItem>
    <asp:ListItem>VIKROLI</asp:ListItem>
    <asp:ListItem>GHATKOPAR</asp:ListItem>
    <asp:ListItem>BADLAPUR</asp:ListItem>
    <asp:ListItem>AMBERNATH</asp:ListItem>
        </asp:DropDownList>
    </td>
</tr>
<tr>
    <td class="style3">
        Timing:</td>
    <td class="style4">
     <asp:DropDownList ID="txttime" runat="server">
        <asp:ListItem Value="">Please Select</asp:ListItem>
    <asp:ListItem>01:00</asp:ListItem>
    <asp:ListItem>02:00</asp:ListItem>
    <asp:ListItem>03:00</asp:ListItem>
    <asp:ListItem>04:00</asp:ListItem>
    <asp:ListItem>05:00</asp:ListItem>
    <asp:ListItem>06:00</asp:ListItem>
    <asp:ListItem>07:00</asp:ListItem>
    <asp:ListItem>08:00</asp:ListItem>
    <asp:ListItem>09:00</asp:ListItem>
    <asp:ListItem>10:00</asp:ListItem>
    <asp:ListItem>11:00</asp:ListItem>
    <asp:ListItem>12:00</asp:ListItem>
    <asp:ListItem>13:00</asp:ListItem>
    <asp:ListItem>14:00</asp:ListItem>
```

```
            <asp:ListItem>15:00</asp:ListItem>
            <asp:ListItem>16:00</asp:ListItem>
            <asp:ListItem>17:00</asp:ListItem>
            <asp:ListItem>18:00</asp:ListItem>
            <asp:ListItem>19:00</asp:ListItem>
            <asp:ListItem>20:00</asp:ListItem>
            <asp:ListItem>21:00</asp:ListItem>
            <asp:ListItem>22:00</asp:ListItem>
            <asp:ListItem>23:00</asp:ListItem>
            <asp:ListItem>24:00</asp:ListItem>
               </asp:DropDownList>
            </td>
        </tr>
                <tr>
    <td class="style7"></td>
    <td align="right" class="style8">
        <asp:Button ID="Btnreg" runat="server" Text="BOOK CAB"
onclick="Btnreg_Click"

            style="height: 35px; margin-right: 150px; margin-left: 0px; background-color:
#FFFF00;" /></td>
    </tr>
            <tr>
        <td colspan="2">
            <asp:Label ID="status" runat="server" ForeColor="Red"></asp:Label></td>
        </tr>
    </table>
    </form>
</body>
</html>
```

Cabbooking.aspx.cs:-

```
using System;
```

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

public partial class CABBOOKING : System.Web.UI.Page
{
   protected void Page_Load(object sender, EventArgs e)
   {
      Label1.Text =(string)Session["username"];
   }
   protected void Btnreg_Click(object sender, EventArgs e)
   {
      String ConnectionString =
WebConfigurationManager.ConnectionStrings["dbconnection"].ConnectionString;
      SqlConnection con = new SqlConnection(ConnectionString);
      try
      {
         con.Open();
         SqlCommand cmd = new SqlCommand(@"INSERT INTO dbo.cabbooking
        ([user],pickuplocation,dropofflocation,timing)
    VALUES
        ('" + Session["username"] + "','" + txtpl.Text + "','" + txtdl.Text + "','" +
txttime.Text + "')",con);
         SqlDataAdapter sda = new SqlDataAdapter(cmd);
         DataTable dt = new DataTable();
         cmd.ExecuteNonQuery();
         con.Close();

         status.Text = "cab booking sucessfull";
```

```
        }
        catch (Exception ex)
        {
            Response.Write(ex.Message);
        }
    }
}
```

adminbooking.aspx:-

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="adminbooking.aspx.cs" Inherits="adminbooking" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
    <style type="text/css">
    .style1
    {
        background-color: #FFFF00;
    }
    .style2
    {
        width: 33%;
        background-color: #C0C0C0;
        height: 351px;
    }
    .style3
    {
        width: 414px;
        text-align: right;
    }
    .style6
    {
        width: 631px;
    }
    .style7
    {
        width: 414px;
        background-color: #C0C0C0;
        height: 32px;
    }
    .style8
    {
        width: 32%;
```

```
        background-color: #C0C0C0;
        height: 32px;
    }
</style>
</head>
<body>
    <form id="form1" runat="server">
    <div style="text-align: center; background-color: #FFFF00">
        CONFIRM BOOKING</div>
        <br />
        <br />
        <asp:GridView align=center ID="GridView1" runat="server"
            onselectedindexchanged="GridView1_SelectedIndexChanged">
            <Columns>
                <asp:ButtonField ButtonType="Button" CommandName="Select"
Text="Allocate" />
            </Columns>
        </asp:GridView>
        <br />
        <table align="center" class="style2">
<tr>
        <td class="style3">
            Username:</td>
        <td class="style6">
            <asp:TextBox ID="Txtusr" runat="server"></asp:TextBox>
        </td>
    </tr>
    <tr>
        <td class="style3">
            Booking ID:</td>
        <td class="style6">
            <asp:TextBox ID="Txtbkid" runat="server"></asp:TextBox>
        </td>
    </tr>
        <tr>
        <td class="style3">
            Pickup Location:</td>
        <td class="style6">
        <asp:TextBox ID="Txtpl" runat="server"></asp:TextBox>
        </td>
    </tr>
    <tr>
        <td class="style3">
            Dropoff Location:</td>
        <td class="style6">
            <asp:TextBox ID="Txtdl" runat="server"></asp:TextBox>
        </td>
    </tr>
    <tr>
        <td class="style3">
```

```
            Timing:</td>
        <td class="style6">
            <asp:TextBox ID="Txttime" runat="server"></asp:TextBox>
        </td>
    </tr>
    <tr>
        <td class="style3">
            Vehicle No:</td>
        <td class="style6">
            <asp:DropDownList ID="Txtvn" runat="server" AutoPostBack="True">
            </asp:DropDownList>
        </td>
    </tr>
    <tr>
        <td class="style3">
            Driver Name:</td>
        <td class="style6">
            <asp:DropDownList ID="Txtd" runat="server" AutoPostBack="True"
                onselectedindexchanged="Txtd_SelectedIndexChanged">
            </asp:DropDownList>
        </td>
    </tr>
        <tr>
        <td class="style3">
            Driver PhoneNo:</td>
        <td class="style6">
            <asp:TextBox ID="Txtdpno" runat="server"></asp:TextBox>
        </td>
    </tr>
        <tr>
<td class="style7"></td>
<td align="right" class="style8">
        <asp:Button ID="Btnconf" runat="server" Text="CONFIRM BOOKING"
onclick="Btnconf_Click"
        style="height: 35px; margin-right: 150px; margin-left: 0px; background-color:
#FFFF00;" /></td>
</tr>
    <tr>
<td colspan="2">
    <asp:Label ID="status" runat="server" ForeColor="Red"></asp:Label></td></tr>
    </table>
    </form>
</body>
</html>
```

Adminbooking.aspx.cs:-

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
```

```
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using System.Data.SqlClient;
using System.Web.Configuration;

public partial class adminbooking : System.Web.UI.Page
{
   protected void Page_Load(object sender, EventArgs e)
   {
      ShowData();
      if (!IsPostBack)
      {
         binddropdownlist();
         binddropdownlist1();
      }
      }
   void ShowData()
   {
      String ConnectionString =
WebConfigurationManager.ConnectionStrings["dbconnection"].ConnectionString;
      SqlConnection con = new SqlConnection(ConnectionString);
      try
      {
         SqlCommand cmd = new SqlCommand(@"select * from cabbooking", con);
         con.Open();
         SqlDataReader sdr = cmd.ExecuteReader();
         GridView1.DataSource = sdr;
         GridView1.DataBind();
         con.Close();
      }
      catch (Exception ex)
      {
         Response.Write(ex.Message);
      }
   }
   protected void GridView1_SelectedIndexChanged(object sender, EventArgs e)
   {
      GridViewRow gr = GridView1.SelectedRow;
      Txtbkid.Text = gr.Cells[1].Text;
      Txtusr.Text = gr.Cells[2].Text;
      Txtpl.Text = gr.Cells[3].Text;
      Txtdl.Text = gr.Cells[4].Text;
      Txttime.Text = gr.Cells[5].Text;
   }
   private void binddropdownlist1()
   {
      String ConnectionString =
WebConfigurationManager.ConnectionStrings["dbconnection"].ConnectionString;
      SqlConnection con = new SqlConnection(ConnectionString);
```

```
        string query = "select * from drivers";
        SqlDataAdapter sda = new SqlDataAdapter(query, con);
        DataTable data = new DataTable();
        sda.Fill(data);
        Txtd.DataSource = data;
        Txtd.DataTextField = "drivername";
        Txtd.DataValueField = "drivername";
        Txtd.DataBind();
        ListItem no = new ListItem("select vehicle no", "-1");
        no.Selected = true;
        Txtd.Items.Insert(0, no);

    }

    protected void Txtd_SelectedIndexChanged(object sender, EventArgs e)
    {
        string selectedDriver = Txtd.SelectedItem.Text;
        string connectionString =
WebConfigurationManager.ConnectionStrings["dbconnection"].ConnectionString;
        SqlConnection con = new SqlConnection(connectionString);
        string query = "SELECT driverphoneno FROM drivers WHERE drivername =
@drivername";
        SqlCommand cmd = new SqlCommand(query, con);
        cmd.Parameters.AddWithValue("@drivername", selectedDriver);
        con.Open();
        string driverPhoneNo = (string)cmd.ExecuteScalar();
        con.Close();
        Txtdpno.Text = driverPhoneNo;
    }

    private void binddropdownlist()
    {
        String ConnectionString =
WebConfigurationManager.ConnectionStrings["dbconnection"].ConnectionString;
        SqlConnection con = new SqlConnection(ConnectionString);
        string query = "select * from cabs";
        SqlDataAdapter sda = new SqlDataAdapter(query, con);
        DataTable data = new DataTable();
        sda.Fill(data);
        Txtvn.DataSource = data;
        Txtvn.DataTextField = "vehicleno";
        Txtvn.DataValueField = "vehicleno";
        Txtvn.DataBind();

        ListItem no = new ListItem("select vehicle no", "-1");
        no.Selected = true;
        Txtvn.Items.Insert(0, no);
    }

    protected void Btnconf_Click(object sender, EventArgs e)
```

```
    {

        String ConnectionString =
WebConfigurationManager.ConnectionStrings["dbconnection"].ConnectionString;
        SqlConnection con = new SqlConnection(ConnectionString);
        try
        {
            con.Open();
            SqlCommand cmd = new SqlCommand(@"INSERT INTO
etms.dbo.cabbookinghistory

(username,bookingid,pickuplocation,dropofflocation,timing,vehicleno,drivername,drive
rphoneno,status)
    VALUES
          ('" + Txtusr.Text + "','" + Txtbkid.Text + "','" + Txtpl.Text + "','" + Txtdl.Text +
"','" + Txttime.Text + "','" + Txtvn.Text + "','" + Txtd.Text + "','" + Txtdpno.Text +
"','confirm')", con);
            SqlDataAdapter sda = new SqlDataAdapter(cmd);
            DataTable dt = new DataTable();
            cmd.ExecuteNonQuery();
            con.Close();
            status.Text = "booking confirm";

            SqlCommand cmdv = new SqlCommand(@"UPDATE cabs SET status = 'not
available' WHERE vehicleno = @vehicleno", con);
            cmdv.Parameters.AddWithValue("@vehicleno", Txtvn.Text);
            cmdv.ExecuteNonQuery();

            SqlCommand cmdd = new SqlCommand(@"UPDATE drivers SET status = 'not
available' WHERE drivername = @drivername", con);
            cmdd.Parameters.AddWithValue("@drivername", Txtd.Text);
            cmdd.ExecuteNonQuery();
        }
        catch (Exception ex)
        {
            Response.Write(ex.Message);
        }
    }
}
```

5.2.2 Code Efficiency:-

I have tried to keep the codes as short as possible but functionalities and reliability aren't compromised. Efficiency is an important aspect of the system as the usability by reducing the complexity.

5.3 Testing Approach:-

Software Testing is a process of evaluating the functionality of a software application to find any software bugs. It checks whether the developed software met the specified requirements and identifies any defect in the software in order to produce a qualityproduct. It is basically executing a system in order to identify any gaps, errors, or missing requirements in contrary to the actual requirements.

The various levels of the testing are as follows:
1.Unit Testing
Unit testing is a type of software testing that focuses on individual units or components of a software system. The purpose of unit testing is to validate that each unit of the software works as intended and meets the requirements. Unit testing is typically performed by developers, and it is performed early in the development process before the code is integrated and tested as a whole system.

Unit tests are automated and are run each time the code is changed to ensure that new code does not break existing functionality. Unit tests are designed to validate the smallest possible unit of code, such as a function or a method, and test it in isolation from the rest of the system. This allows developers to quickly identify and fix any issues early in the development process, improving the overall quality of the software and reducing the time required for later testing.

5.3.1 Unit Testing:-

Testing phase starts with designing the test cases for each module. The system is divided into different modules. These modules are further divided into small units. Each module and its units were analyzed and then test cases were formed.

5.4 TestCases:-

Login:-

Test case no.	Test case name	Expected output	Actual output	Remark
1.	UserName :Anuj72 Password:Anuj@0772	User should get redirected to home page after successful login.	User was redirected to home page.	**PASS**
2.	UserName :NULL Password:NULL	User should get an error message Saying. *Please, enter username. *Please, enter password.	User got an error message Saying please. *Please, enter username *Please, enter password.	**PASS**
3.	UserName :abc Password:Abc@123	User should get a message saying. *invalid credentials.	User got a message saying. *invalid credentials.	**PASS**

Table 5.1 Login

Registration:-

Test case no.	Test case name	Expected output	Actual output	Remark
1.	Firstname:ANUJ Lastname:MHATRE Employee ID:1 Email ID:anujpm1811@gmail.com Phone NO:9757434055	User should get redirected to login page after successful	User was redirected to login page.	**PASS**

	Address:KALYAN Username:Anuj72 Password:Anuj@0772	registration.		
2.	Firstname:NULL Lastname:NULL Employee ID:NULL Email ID:NULL Phone NO:NULL Address:NULL Username:NULL Password:NULL	User should get an error message Saying. *mandatory.	User got an error message Saying please. *mandatory.	**PASS**
3.	Firstname:ANUJ Lastname:MHATRE Employee ID:1 Email ID:anujpm1811gmail.com Phone NO:975743405 Address:KALYAN Username:Anuj72 Password:Anuj@0772	Registered successfully and redirected to login page.	User got a system error. *mandatory.	**PASS** I have set pattern for email and maxlength 10 digits for phone no.

Table 5.2 Registration

Cab booking:-

Test case no.	Test case name	Expected output	Actual output	Remark
1.	Pickup location:KALYAN Dropoff location:POWAI Timing:01:00	User should get message Saying. cab booking successful.	User got an message Saying. cab booking successful.	**PASS**

2.	Pickup location:NULL Dropoff location:NULL Timing:NULL	User should get message Saying. pls fill all fields.	User got message Saying. pls fill all fields.	**PASS**
3.	Pickup location:KALYAN Dropoff location:POWAI Timing:01:00	User should get message Saying. cab booking successful.	User got an message Saying. cab booking successful.	**FAIL**

Table 5.3 Cab booking

Contact us:-

Test case no.	Test case name	Expected output	Actual output	Remark
1.	NAME:ANUJ EMAIL:anujpm1811@gmail.com PHONE NO:9757434055 MESSAGE:HI	User should get message Saying. complaint submited successful.	User got an message Saying. complaint submited successful.	**PASS**
2.	NAME:NULL EMAIL:NULL PHONE NO:NULL MESSAGE:NULL	User should get message Saying. pls fill all details.	User got message Saying. pls fill all details.	**PASS**
3.	NAME:ANUJ EMAIL:anujpm1811gmail.com PHONE NO:975743405 MESSAGE:HI	User should get message Saying. pls fill all details.	User got message Saying. pls fill all details.	**PASS** I have set pattern for email and maxlength 10 digits

				for phone no.
4.	NAME:ANUJ EMAIL:anujpm1811@gmail.com PHONE NO:9757434055 MESSAGE:HI	User should get message Saying. complaint submited successful.	User got an message Saying. complaint submited successful.	**FAIL**

Table 5.4 Contact us

Feedback:-

Test case no.	**Test case name**	**Expected output**	**Actual output**	**Remark**
1.	NAME:ANUJ EMAIL:anujpm1811@gmail.com RATINGS:5 FEEDBACK:GOOD	User should get message Saying. feedback submited successful.	User got an message Saying. feedback submited successful.	**PASS**
2.	NAME:NULL EMAIL:NULL RATINGS:NULL FEEDBACK:NULL	User should get message Saying. pls fill all details.	User got message Saying. pls fill all details.	**PASS**
3.	NAME:ANUJ EMAIL:anujpm1811gmail.com RATINGS:5 FEEDBACK:GOOD	User should get message Saying. pls fill all details.	User got message Saying. pls fill all details.	**PASS** I have set pattern for email.

Table 5.5 Feedback

Manage users:-

Test case no.	Test case name	Expected output	Actual output	Remark
1.	User Name:Anuj72 Name:ANUJ Phone No:9757434055 Address:KALYAN Click on delete button	Admin should get message Saying. user data deleted successfully.	Admin got an message Saying. user data deleted sucessfully.	**PASS**
2.	User Name:NULL Name:NULL Phone No:NULL Address:NULL Click on delete button	Admin should get message Saying. pls fill all details.	Admin got an message Saying. pls fill all details.	**PASS**
3.	User Name:Anuj72 Name:ANUJ Phone No:9757434055 Address:THANE Click on update button	User should get message Saying. data updated successfully.	User got message Saying. data updated successfully.	**PASS**
4.	User Name:NULL Name:NULL Phone No:NULL Address:NULL Click on update button	Admin should get message Saying. pls fill all details.	Admin got an message Saying. pls fill all details.	**PASS**

Table 5.6 Manage users

Manage cabs:-

Test case no.	Test case name	Expected output	Actual output	Remark
1.	ID:1 Vehicle No:MH05DH0772 Vehicle Name:WAGNOR Vehicle Permit:VALID TILL 2024 Click on insert button	Admin should get message Saying. data inserted successfully.	Admin got an message Saying. data inserted successfully.	**PASS**
2.	ID:NULL Vehicle No:NULL Vehicle Name:NULL Vehicle Permit:NULL Click on insert button	Admin should get message Saying. pls fill all details.	Admin got an message Saying. pls fill all details.	**PASS**
3.	ID:1 Vehicle No:MH05DH0772 Vehicle Name:ERTIGA Vehicle Permit:VALID TILL 2024 Click in update button	User should get message Saying. data updated successfully.	User got message Saying. data updated successfully.	**PASS**
4.	ID:NULL Vehicle No:NULL Vehicle Name:NULL Vehicle Permit:NULL Click in update button	Admin should get message Saying. pls fill all details.	Admin got an message Saying. pls fill all details.	**PASS**
5.	ID:1 Vehicle No:MH05DH0772	Admin should get	Admin got an message	**PASS**

	Vehicle Name:WAGNOR Vehicle Permit:VALID TILL 2024 Click on delete button	message Saying. cab data deleted successfully.	Saying. cab data deleted sucessfully.	
6.	ID:NULL Vehicle No:NULL Vehicle Name:NULL Vehicle Permit:NULL Click in delete button	Admin should get message Saying. pls fill all details.	Admin got an message Saying. pls fill all details.	**PASS**

Table 5.7 Manage cabs

Managing drivers:-

Test case no.	Test case name	Expected output	Actual output	Remark
1.	ID:1 Driver Name:ANUJ Driver Phone No:9757434055 Driver Licence:VALID Click on insert button	Admin should get message Saying. data inserted successfully.	Admin got an message Saying. data inserted successfully.	**PASS**
2.	ID:NULL Driver Name:NULL Driver Phone No:NULL Driver Licence:NULL Click on insert button	Admin should get message Saying. pls fill all details.	Admin got an message Saying. pls fill all details.	**PASS**
3.	ID:1 Driver Name:APM	User should get message	User got message	**PASS**

	Driver Phone No:9757434055 Driver Licence:VALID Click in update button	Saying. data updated successfully.	Saying. data updated successfully.	
4.	ID:NULL Driver Name:NULL Driver Phone No:NULL Driver Licence:NULL Click in update button	Admin should get message Saying. pls fill all details.	Admin got an message Saying. pls fill all details.	**PASS**
5.	ID:1 Driver Name:ANUJ Driver Phone No:9757434055 Driver Licence:VALID Click on delete button	Admin should get message Saying. driver data deleted successfully.	Admin got an message Saying. driver data deleted sucessfully.	**PASS**
6.	ID:NULL Driver Name:NULL Driver Phone No:NULL Driver Licence:NULL Click in delete button	Admin should get message Saying. pls fill all details.	Admin got an message Saying. pls fill all details.	**PASS**

Table 5.8 Manage drivers

Allocating cab and driver:-

Test case no.	Test case name	Expected output	Actual output	Remark
1.	Username:Anuj72 Booking ID:1 Pickup Location:KALYAN Dropoff Location:POWAI Timing:01:00	Admin should get message Saying. booking	Admin got an message Saying. booking confirm.	**PASS**

	Vehicle No:MH05DH0772 Driver Name:ANUJ Driver PhoneNo:9757434055	confirm.		
2.	Username:NULL Booking ID:NULL Pickup Location:NULL Dropoff Location:NULL Timing:NULL Vehicle No:NULL Driver Name:NULL Driver PhoneNo:NULL	Admin should get message Saying. pls fill all details.	Admin got an message Saying. pls fill all details.	**PASS**
3.	Username:Anuj72 Booking ID:1 Pickup Location:KALYAN Dropoff Location:POWAI Timing:01:00 Vehicle No:NULL Driver Name:NULL Driver PhoneNo:NULL	Admin should get message Saying. pls allocate.	Admin got an message Saying. pls allocate.	**PASS**

Table 5.9 Allocating cab and driver

Logout:-

Test case no.	Test case name	Expected output	Actual output	Remark
1.	Click on logout button	User should get redirect to user login page.	User should got redirect to user login page.	**PASS**

Table 5.10 Allocating cab and driver

CHAPTER 6:

RESULTS AND DISCUSSION

6.1 Test Reports:-

The testing part of project development is a very important phase. The testing phase helps to know whether all the functionalities are being performed the way that they are supposed to be executed.

Unit testing has been performed for this project.

Total test cases -36

Passed test cases-34

Failed test cases-02

The failed test cases were search functionality is not working for cabbooking and contact us is being added without fill details.

Cause- Absence of required validation

Solution- Added required validation and make compulsory for user to fill all details.

6.2 User Documentation:-

1. The first page of website is Login page. User need to login first.

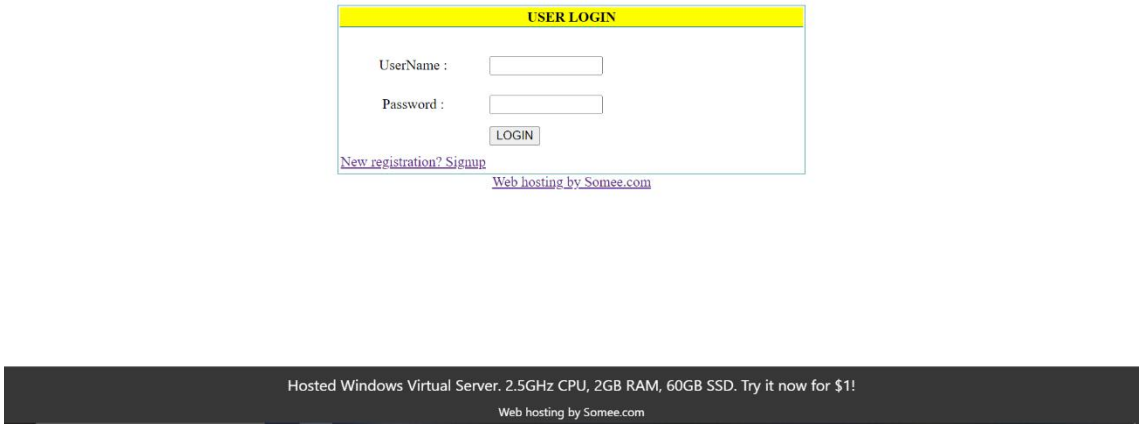


Fig 6.1 User login UI

2. If the user is new to the website, he/she has to register/signup first.

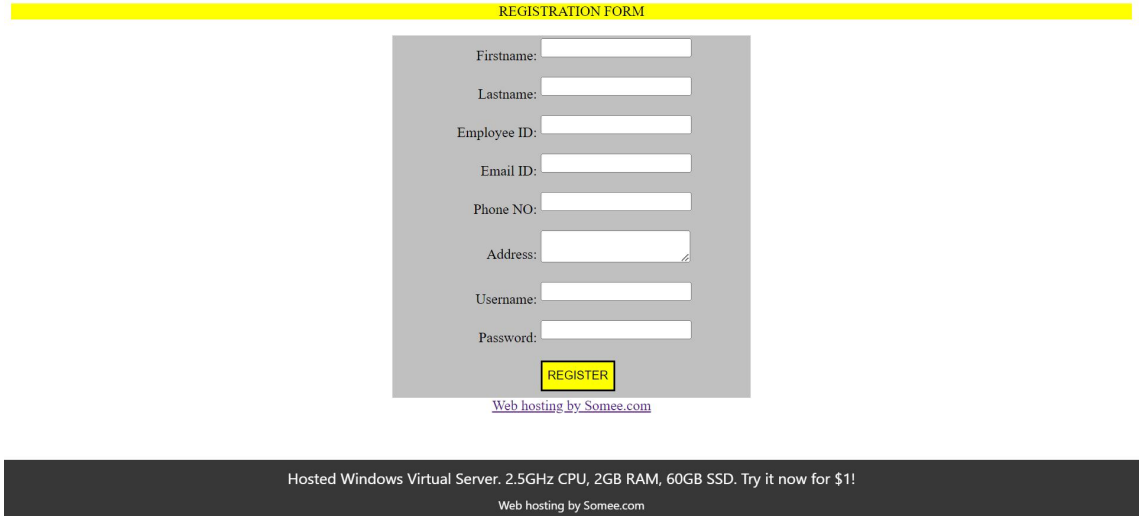


Fig 6.2 Registration UI

3.The users will get to see a homepage like this after logging in to the website.

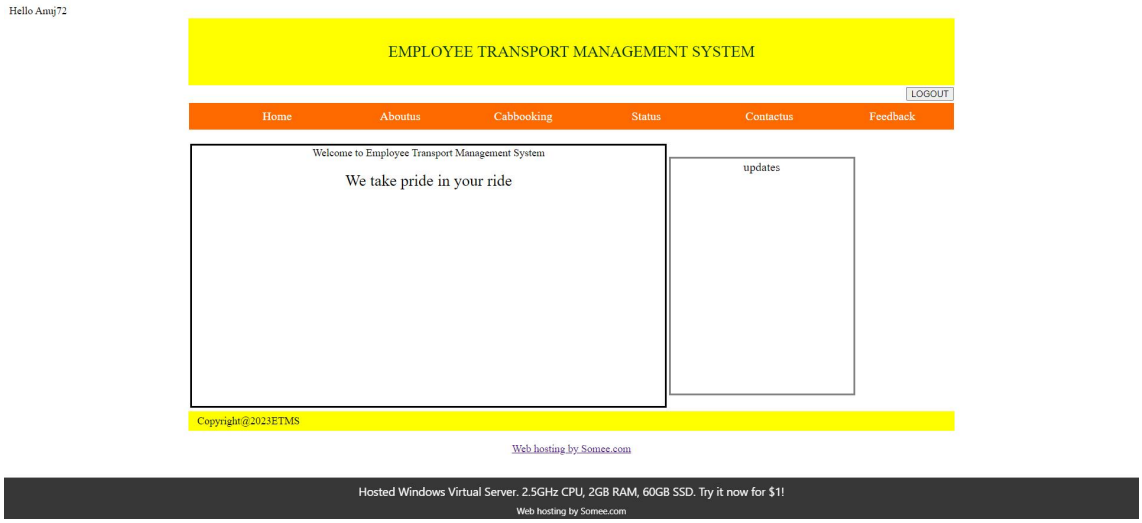


Fig 6.3 Homepage UI

Homepage has 6 tabs in the menu bar Home, Aboutus, Cabbooking, Status, Contactus,Feedback and Logout button,and below there is updates section.

4.About us section has details of the ETMS.

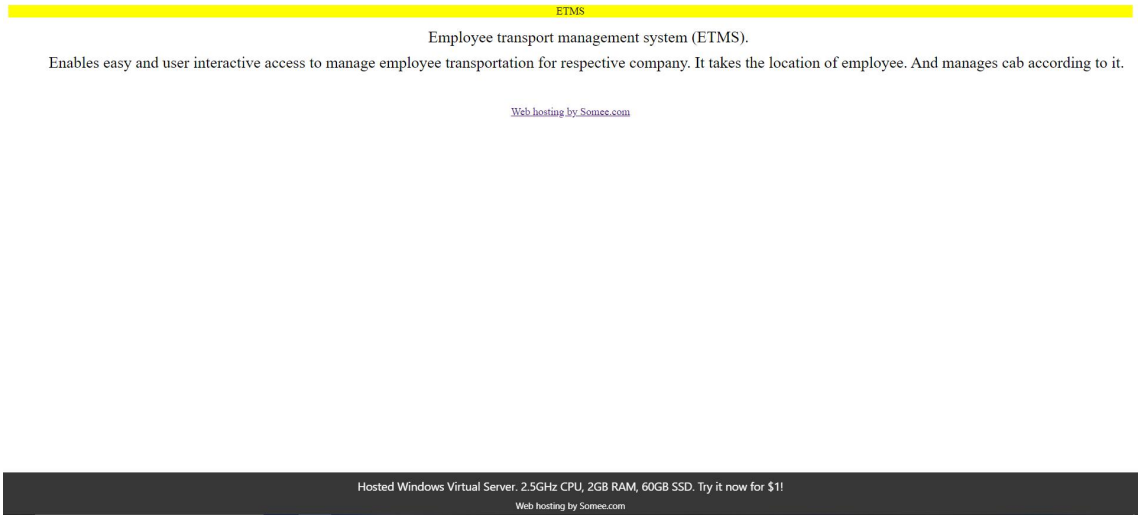
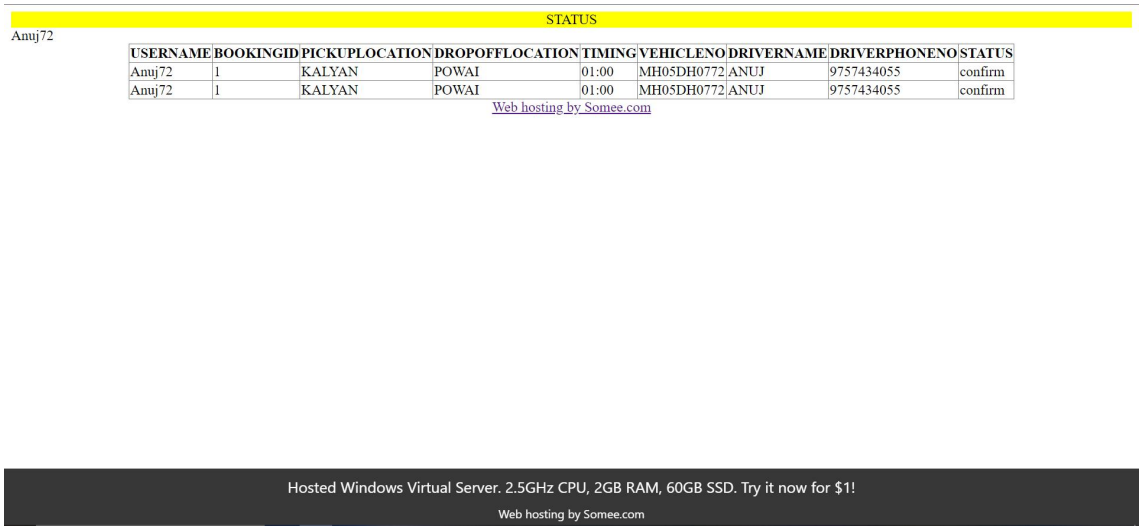


Fig 6.4 About us UI

5.In cabbooking section user cab book cab ny secting location and time given in dropdown list.

Fig 6.5 Cabbooking UI

6.In status section user can see his booking history and booking status.


STATUS

Anuj72

USERNAME	BOOKINGID	PICKUPLOCATION	DROPOFFLOCATION	TIMING	VEHICLENO	DRIVERNAME	DRIVERPHONENO	STATUS
Anuj72	1	KALYAN	POWAI	01:00	MH05DH0772	ANUJ	9757434055	confirm
Anuj72	1	KALYAN	POWAI	01:00	MH05DH0772	ANUJ	9757434055	confirm

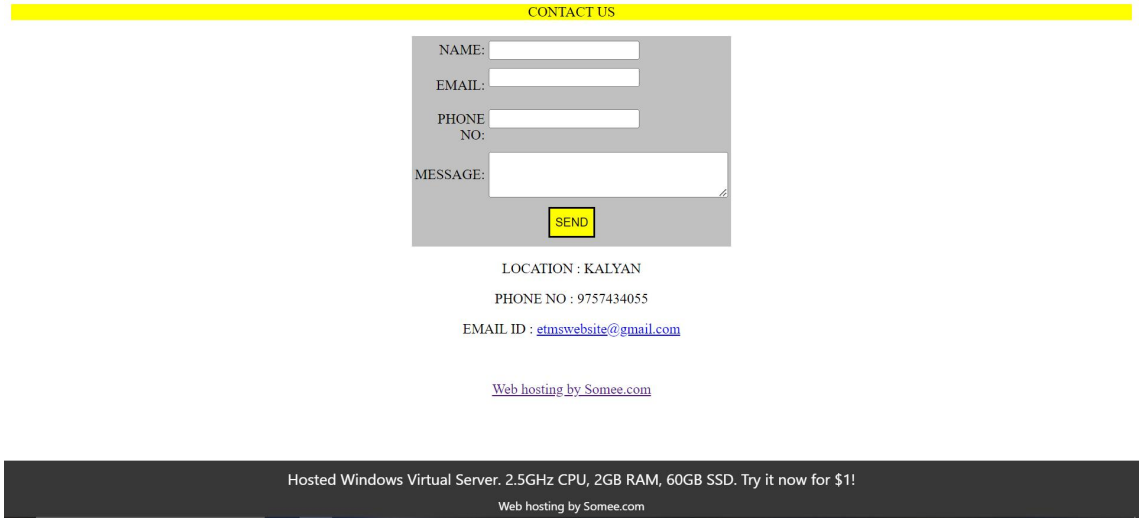
Web hosting by Somee.com

Hosted Windows Virtual Server. 2.5GHz CPU, 2GB RAM, 60GB SSD. Try it now for $1!

Web hosting by Somee.com

Fig 6.6 Status UI

7.Contact us page is where user can write any query for admin after filling their registered username and valid email id.


CONTACT US

NAME:

EMAIL:

PHONE NO:

MESSAGE:

SEND

LOCATION : KALYAN

PHONE NO : 9757434055

EMAIL ID : etmswebsite@gmail.com

Web hosting by Somee.com

Hosted Windows Virtual Server. 2.5GHz CPU, 2GB RAM, 60GB SSD. Try it now for $1!

Web hosting by Somee.com

Fig 6.7 Contact us UI

8. Feedback page is where user can write feedback about system after filling their registered username and valid email id.

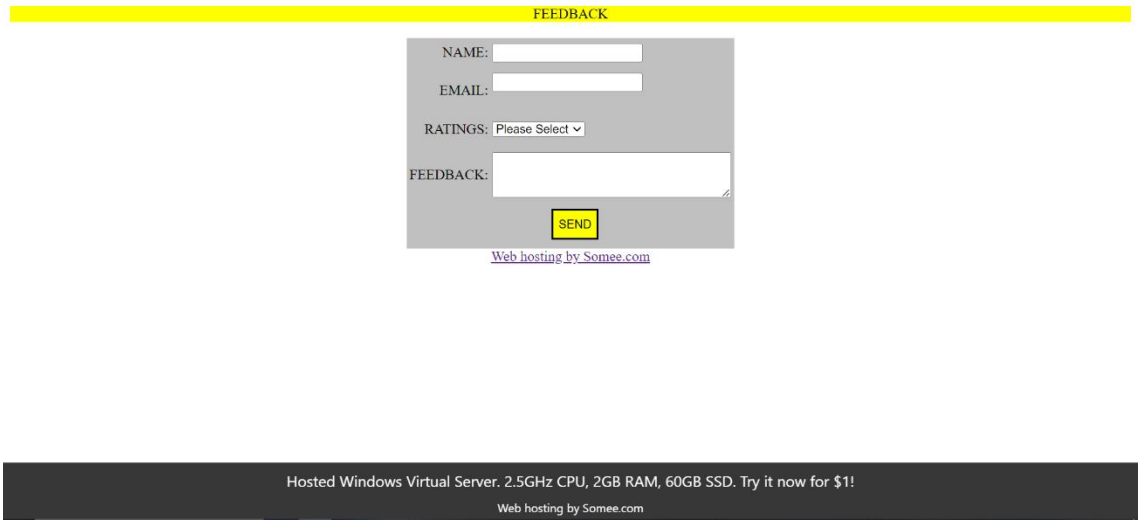


Fig 6.8 Feedback UI

9. This is admin Login page. admin need to login first.

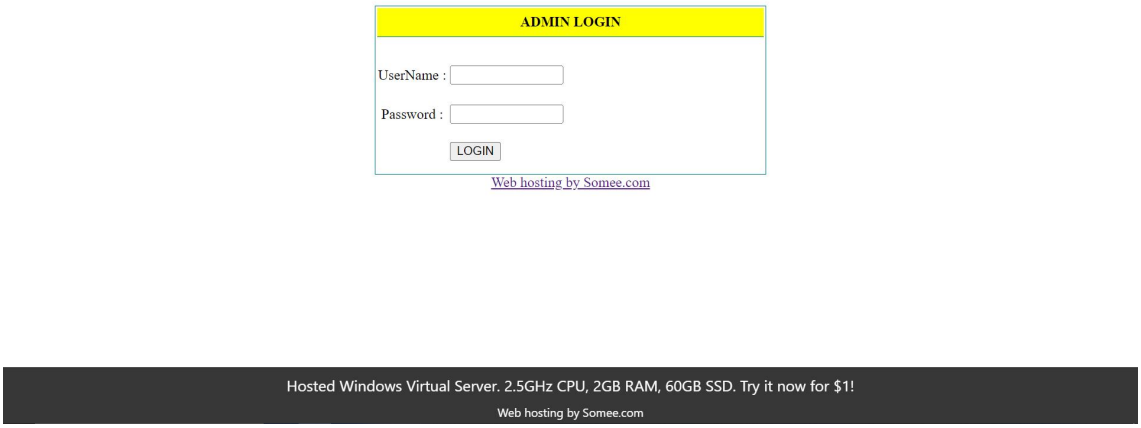


Fig 6.9 Admin login UI

10. The admin will get to see a homepage like this after logging in to the website.

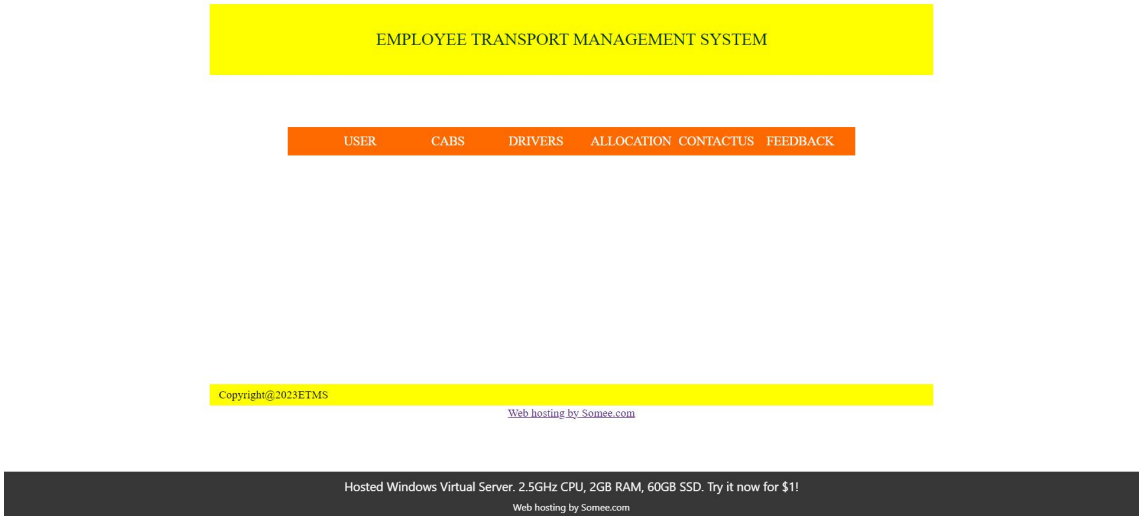


Fig 6.10 Admin homepage UI

Homepage has 6 tabs in the menu bar User, Cabs, Drivers, Allocation, Contactus and Feedback.

11. User page where admin can manage user data.

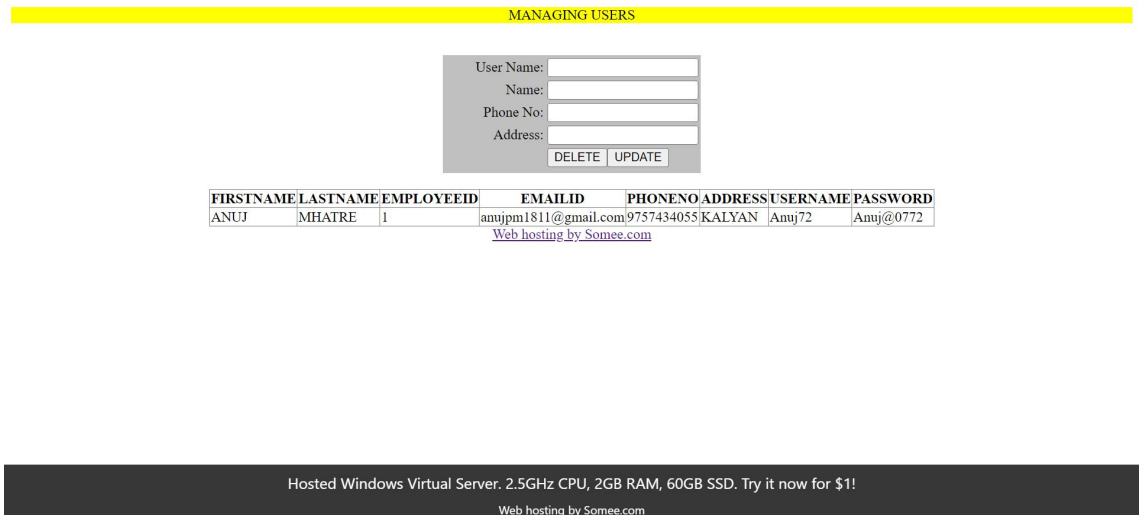


Fig 6.11 Managing users UI

12. Cabs page where admin can manage cab data.

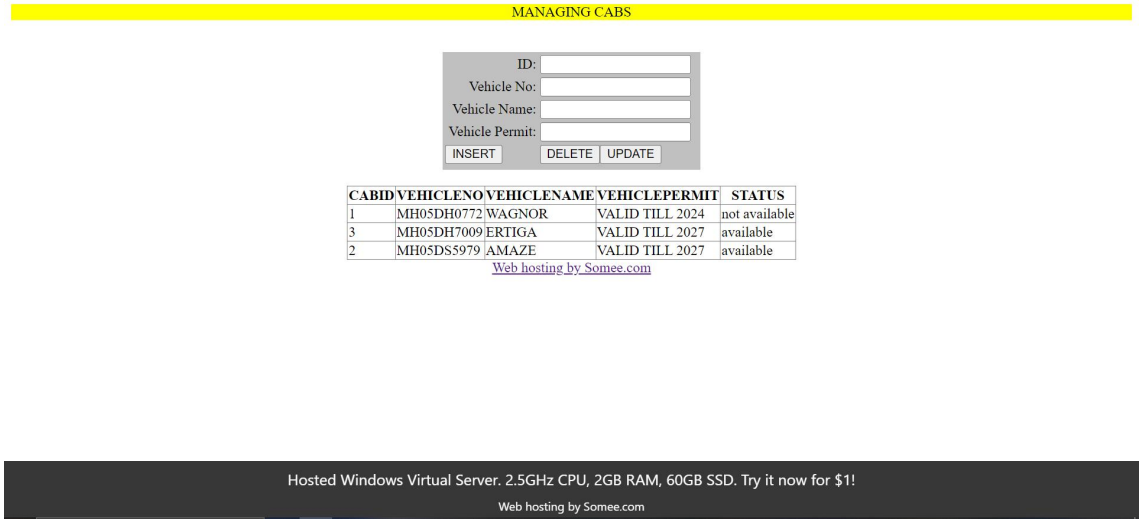


Fig 6.12 Managing cabs UI

13. Drivers page where admin can manage driver data.

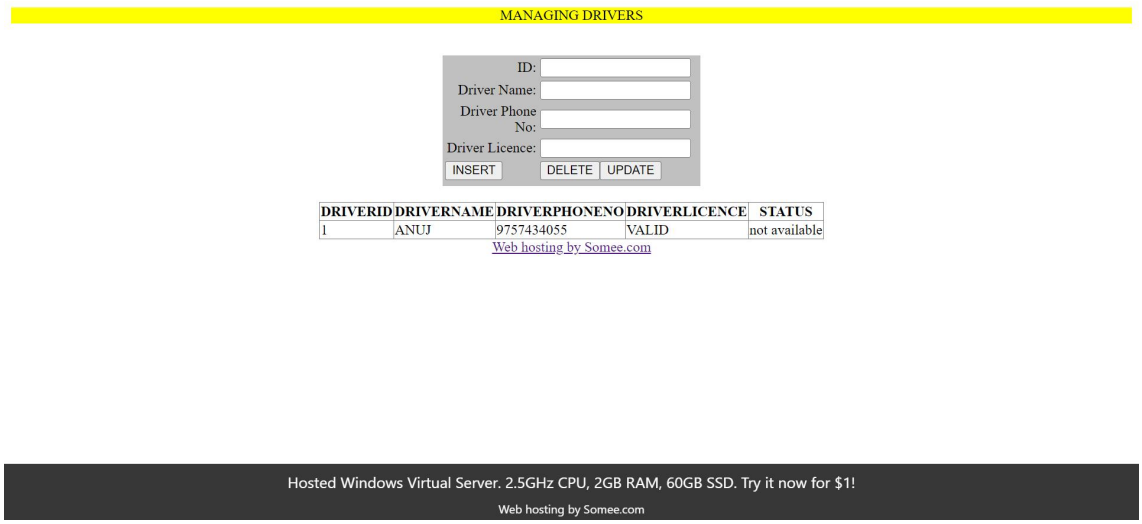


Fig 6.13 Managing drivers UI

14. Allocation page where admin can allocate cab and driver to all cab booking request data.

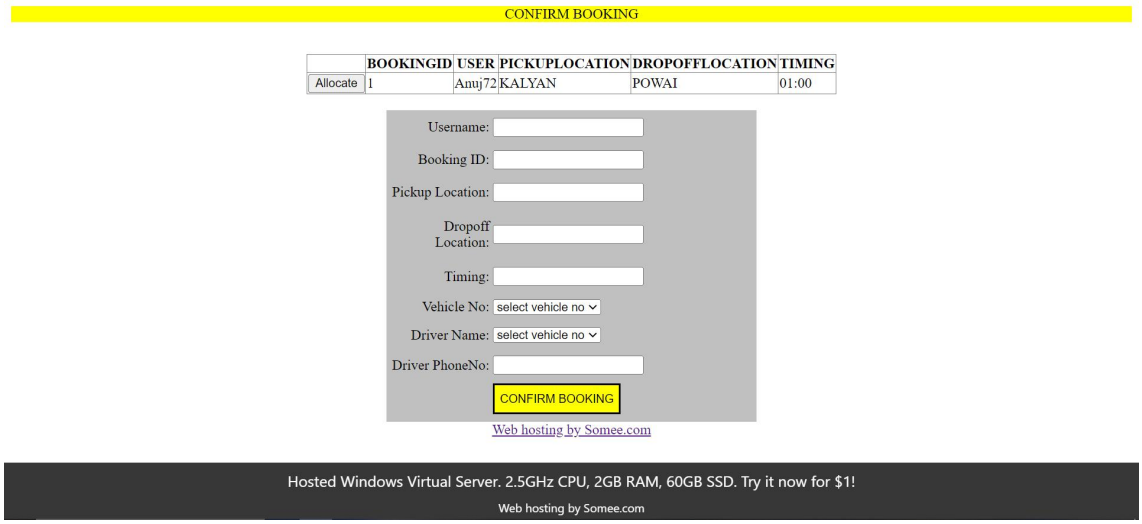


Fig 6.14 Confirm booking UI

15. Contact us page where admin can see complaint requests from users.

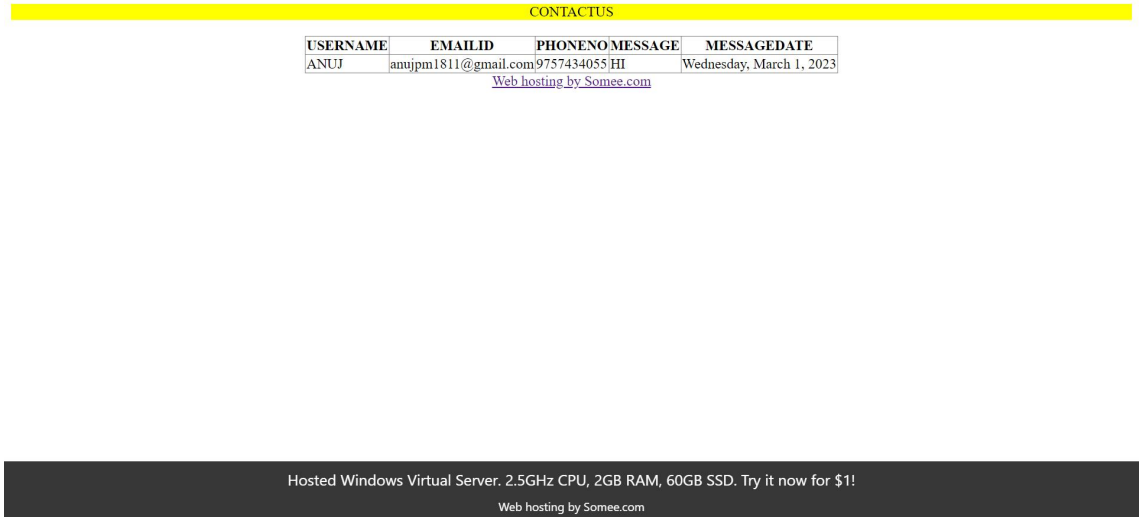


Fig 6.15 Managing contactus UI

16. Feedback page where admin can see feedback from users.

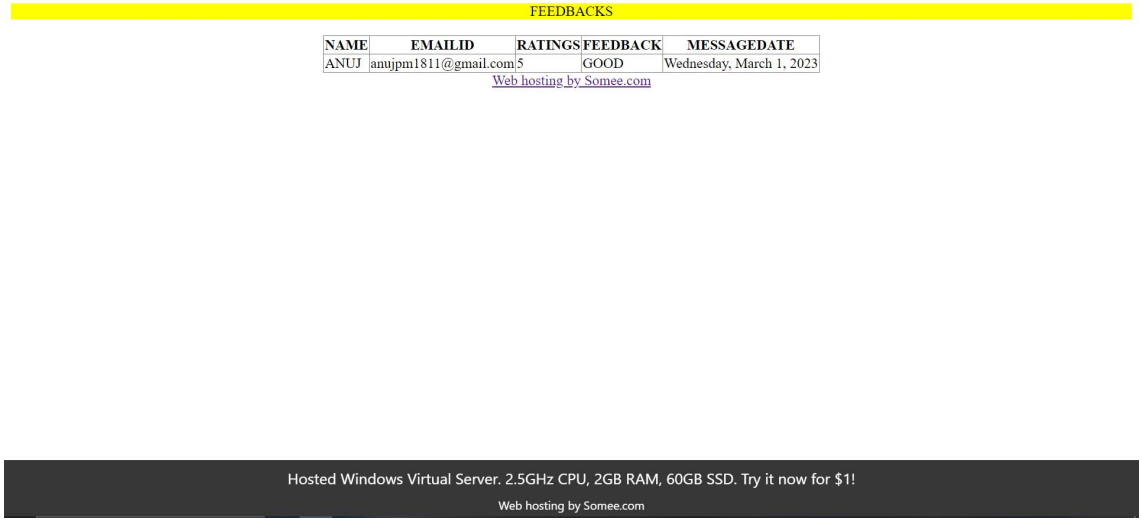


Fig 6.16 Managing feedback UI

CHAPTER 7:

CONCLUSIONS

7.1 Conclusion:-

This project helped me to develop and learn a new aspect of coding. This project was successfully developed using various programming languages like HTML, CSS, C#,Javascript, and MySQL,Asp.net. A lot of various features were developed in this system which is used in real-time environments. It was a wonderful learning experience for me while working on the project. This project took me through the various phases of project development and gave me a real insight into the world of software engineering.

The main idea behind the projrect was to create a system that will make the employee transportation easy. And while developing I learned new things and explored a new way of learning too.

Step-wise planning and of modules and each and every page of this system was done before programming. All the modules of this system/website were tried and tested using various inputs and only then were finalized for the final project. While programming for this project, lots of errors was encountered. After lots of errors and bugs testing and finding their solutions, a final system was developed. Overall, the development of this project was a new learning experience for an IT student.

This project will help all the employee for their hassel free transportation for free with reliability.

7.2Limitations of System:-

1.This Project is only capable to Handle Moderate Traffic as the hosting solution is based on 2.5GHz CPU, 2GB RAM, 60GB SSD.

2. User cannot update their username and password.

3. Admin cannot reply directly from the page to any complaint or feedback request, They have to copy the users email address and paste it in the mail recipients to reply them.

4.User cannot cancle their cabbooking request.

7.3 Future Scope of project:-

1. With little more modification of this system, it can actually be used by the organization.

2. In future user can get text msg of their cabbooking.

3.The project has wide scope, as it is not intended to a particular company. This project is going to develop generic software. Which can be applied by any business, companies more over it provides facility to its users.

BIBLIOGRAPHY: -

Book references: -

- Database System and Concepts, A Silberschatz, H Korth, S Sudarshan, McGraw-Hill, Fifth Edition.
- Software Engineering, edition, Ian Somerville Pearson Education. Ninth
- Object – Oriented Modelling and Design Michael Blaha, James Rumbaugh Pearson 2011

Website references: -

Referred from 25/07/22 to 26/02/23

- https://www.lucidchart.com
- https://www.tutorialspoint.com
- https://www.javatpoint.com
- https://www.geeksforgeeks.org
- https://stackoverflow.com/
- https://www.c-sharpcorner.com/
- https://www.asp.net/
- https://somee.com/