**AN APPLICATION FOR LOAD SHARING IN TRUCKS**

**A project report submitted in the partial fulfillment of the requirements for the**

**Award of the Degree of**

**MASTER OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**

Submitted By

**Student Name**

**Reg.No:XXXXXXXX**

Under the esteemed guidance of

**Guide Name.**

**Associate Professor**

College Logo

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**College Name**

**2014-2017**

**College Name**

**CERTIFICATE**

This is to certify that this project work entitled “ **Project Name”** is the bonafide work carried out by **Student Name ,Reg.No:XXXXXX** submitted in Partial fulfillment of the requirement for the Award of Degree of xxxx of Technology in Computer Science and Engineering, during the acadamic year 2014-2017.

The results submitted in this project have been verified and are found to be satisfactory. The results embodied in this thesis have not been submitted to any other university for the award of the any other degree/diploma.

**Signature of Project Supervisor Signature of Head of the Department**

**Signature of External Examiner**

**ACKNOWLEDGEMENT**

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of people who made it possible, whose constant guidance and encouragement crowned the efforts with success. It is a pleasant aspect that I have now the opportunity to express my gratitude for all of them.

The first person I would like to thank is my project guide **XXXXX, Associate Professor**, Department of Computer Science and engineering, had given continuous critical suggestions and extension of proper working atmosphere, abiding interest has finally evolved into this research work.

It is indeed with a great sense of pleasure and immense sense of guidance that I acknowledge the help and I am highly indebted to our principal **XXXXX** provided to accomplish this work.

I would like to express my sincere thanks to **XXXXXX** Department of Computer Science and Engineering.

I owe special thanks to Departmental committee for the successful completion of this project. I am grateful to all staff members, Department of Computer Science and Engineering, xxxxxx for their valuable suggestions and encouragement.

**Student Name**

**(Roll No)**

The paradigm of the world has shifted to mobile applications. The services that you need are at your fingertips because of the mobile applications which provide online access to the services you have. Several mobile applications like uber freight, ola are present to facilitate the freight service providers. The existing freight management system depends on the model of separating the users on the basis of fleet provider, normal user and one with the large number of vehicles called as freight provider. For a user to login to the system the user can easily enter his desired destination and send his or her belongings to that destination with the online reservation platform provided by the mobile application. Though the existing system is good but it has one limitation. The limitation of all the existing system is that they don’t monitor the load in the vehicle. In order to overcome this limitation, we propose a system which will monitor the load present in the vehicle in real time. We propose a cross platform mobile application which will work on all the available platform. The main aim of the proposed system is to monitor the load present in the truck. The proposed system will use the web applicaton in the load bearing part of the truck and the total volume as well as the available space in the truck will be continuously monitored and notified to the user who wants to send the load. We will also take into consideration the density of the material so that if the user wants to add a new load the existing load won’t be affected. The path of the truck will also be monitored from origin to destination. The proposed method will develop an application for continuous monitoring of the load in the truck. The main application of the system is use in freight management system for monitoring the load of the vehicle and notifying the users whether the space is available for the vehicle to carry their loads..

.

1. INTRODUCTION 1
   1. [PROBLEM DEFINITION **2**](#_TOC_250018)
   2. [PROJECT PURPOSE **2**](#_TOC_250017)
   3. [PROJECT FEATURES **2**](#_TOC_250016)
   4. [MODULES DESCRIPTION **3**](#_TOC_250015)
2. LITERATURE SURVEY
   1. [SEMANTIC SEARCH **4**](#_TOC_250014)
   2. [EXISTING SYSTEM **5**](#_TOC_250013)
   3. [PROPOSED SYSTEM **5**](#_TOC_250012)
   4. [SOFTWARE DESCRIPTION **6**](#_TOC_250011)
3. REQUIREMENT ANALYSIS
   1. [FUNCTIONAL REQUIREMENTS **23**](#_TOC_250010)
   2. [NON FUNCTIONAL REQUIREMENTS **23**](#_TOC_250009)
   3. [HARDWARE REQUIREMENTS **25**](#_TOC_250008)
   4. [SOFTWARE REQUIREMENTS **25**](#_TOC_250007)
4. DESIGN
   1. [DESIGN GOALS **26**](#_TOC_250006)
   2. [SYSTEM ARCHITECTURE **26**](#_TOC_250005)
   3. [ACTIVITY DIAGRAM **27**](#_TOC_250004)
   4. [CLASS DIAGRAM **28**](#_TOC_250003)
   5. [SEQUENCE DIAGRAM **29**](#_TOC_250002)

[4.6 USE CASE DIAGRAM **30**](#_TOC_250001)

1. IMPLEMENTATION
   1. [MODULE DESCRIPTION **31**](#_TOC_250000)
2. **TESTING**
   1. UNIT TESTING **33**
   2. INTEGRATION TESTING **33**
   3. VALIDATION TESTING **34**
   4. SYSTEM TESTING **35**

#### SNAPSHOT 38

#### CONCLUSION AND FUTURE ENHANCEMENT

* 1. CONCLUSION **46**
  2. FUTURE ENHANCEMENT **46**

#### BIBLIOGRAPHY 47

# LIST OF FIGURES

* 1. Working of a compiler 7
  2. Java compiler 7
  3. Program running on a java platform 8
  4. Java2 SDK 10
  5. Tomcat Web server 21
  6. System Architecture 26
  7. Activity Diagram 27
  8. Class Diagram 28
  9. Sequence Diagram 29
  10. Use Case Diagram 30

6.1 The testing process 34

v

**CHAPTER 1**

# INTRODUCTION

The popular one is through the development of application whether web or mobile or both. Sharing of load is done on the basis of load bearing capacity of the vehicle. A threshold is fixed taking into account the load bearing capacity of the vehicle. If the load exceeds the threshold the load is either discarded or charged extra. Various mobile applications are present in the day to day life to facilitate the users from booking the vehicles online and sharing their load. The mobile applications that are present can be categorized into three types, viz native application, hybrid application and web application. Native application can run in only one specific platform whereas hybrid application is a cross platform application that can run in all the platforms available. Web applications are the responsive version of the website that works on any mobile device. There are various platforms for mobile application development. Amongst them the most popular one is flutter. Flutter is a cross platform application developed by google which helps in the development of mobile application easily and readily. Flutter is an open-source environment developed by google to make various mobile applications such as IOS and Android. It is used to develop applications for web and google itself. A single codebase is used to develop mobile, web and desktop applications. The features of flutter which makes it popular are as follows:  Fast development because of various built-in customizable widgets.  Flutter has flexible and dynamic UI.  Flutter gives the native performance. The development of application using flutter requires flutter SDK along with the editor like android studio and other according to the preference of the developer. The proposed method uses android studio as editor. An android studio is a development environment for Google’s android operating system. It can be used in all the operating systems. With the help of android studio one can edit, analyze, emulate and preview the android application before release. There are various features of android studio which makes is a popular choice amongst all the other editors. Some of the features of android studio are as follows:  Gradle based build support.  ProGuard integration and app signing capabilities.  It has template-based wizards to design the android components.  Dag and drop of UI components makes it easy to use.  Integrated support for google cloud platform..

## PROBLEM DEFINITION

Though the existing system is good but it has one limitation. The limitation of all the existing system is that they don’t monitor the load in the vehicle..

## PROJECT PURPOSE

In order to overcome this limitation, we propose a system which will monitor the load present in the vehicle in real time. We propose a cross platform mobile application which will work on all the available platform. The main aim of the proposed system is to monitor the load present in the truck.

## PROJECT FEATURES

The proposed system will use the ultrasonic sensor in the load bearing part of the truck and the total volume as well as the available space in the truck will be continuously monitored and notified to the user who wants to send the load. We will also take into consideration the density of the material so that if the user wants to add a new load the existing load won’t be affected. The path of the truck will also be monitored from origin to destination.

## MODULES DESCRIPTION

### Admin Module

Using this module admin will login to application and view registered users, active logistic user and view logistic users and check booked details.

### User Module.

Using this module user will register with application with personal and logistic details and search for source and destination details and book slot with kgs and get cost for given kgs and send request to logistic person for confirmation and check bookings.

### Logistic Module.

Using this module logistic person will register with personal details and upload source and destination details with four sub routes and add total kgs and cost for kg for transport. Logistic module can confirm booking or user and view bookings.

**CHAPTER 2**

# LITERATURE SURVEY

Lots of ideas have been proposed for the development of application to share the load in vehicles. Literature review gives the limitations as well as merits of the existing system. It is arranged as per the following. Mohammad et. al (2018) [1] have proposed an appointment system for trucks which is useful for both the container terminal operators and drayage operators. The system focuses on time management by distributing the truck’s arrival time evenly throughout the day as well as giving appointment time to the trucks to avoid heavy traffic and collision. It helps in reducing the impact of both the operators. Xinjie et. al (2019) [2] have proposed a quotation booking process by integrating global tank container operation. It emphasis on policy making and controlling inventory. Also, it focuses on integrating job acceptance/rejection decisions in the quotation booking process. The system uses Heuristic algorithm along with adjusted genetic algorithm and various mathematical expressions. Chenhao et. al (2018) [3] have proposed a system to reduce thcongestion in lighterage terminal by developing a coordinated schedule. Real world terminal is studied and model is developed to determine the coordinated arrival schedule of trucks. Advanced bio-objective simulation optimization is used and the system is delivered using mobile application. Dong (2016) [4] have proposed a booking control system to solve the revenue management problem for a single station car rental system. Decomposition approach and approximate dynamic program are used to analyze the system. Determinant of mobile taxi booking application service’s continuance by Goi Sai Wang (2018) [5] have proposed a system that predicts the intention of the users to continue the use of mobile application by using technology continuance theory. The data are analyzed by using least square technique. Teoder (2018) [6] have analyzed the different aspects of car sharing system and introduced a taxonomy for the same. The differences between the operational level and economic aspects along with the business model and customer’s perspective is studied. Vehicle booking system by S. Vidhya (2016) [7] developed an android application for reservation of available space in the parking lots. The system consists of a microcontroller and GSM module that costs less as compared to the other existing systems. Kangija (2018) [8] have proposed vehicle scheduling methodology to solve online vehicle dispatch problem. An optimization framework is developed and new approach is presented to solve the gap between the big problem size and limited computing time. Zhigang (2019) [9] have proposed consortium blockchain system-based charging guide technique. Based on consortium blockchain a taxi charging guide architecture is developed and fault tolerance algorithm is used to solve the issue of trust amongst multiple charging station as well as the charging information disconnection. The system minimizes the charging cost of the taxis. Quantum based particle swarm optimization algorithm is used for analyzing the system.

## EXISTING SYSTEM

• It helps in reducing the impact of both the operators. Xinjie et. al (2019) [2] have proposed a quotation booking process by integrating global tank container operation. It emphasis on policy making and controlling inventory. Also, it focuses on integrating job acceptance/rejection decisions in the quotation booking process. The system uses Heuristic algorithm along with adjusted genetic algorithm and various mathematical expressions..

## PROPOSED SYSTEM

In proposed system we are developing and website which will provide solution for users to book logistic services through online by selecting source and destination address and location for source to destination. Logistics can register with application and view bookings where admin can manage services.

## SOFTWARE DESCRIPTION

#### Java Technology

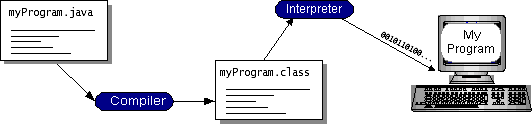
Java technology is both a programming language and a platform.

#### The Java Programming Language

The Java programming language is a high-level language that can be characterized by all of the following buzzwords:

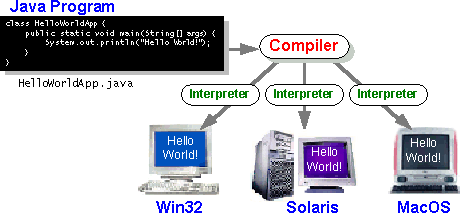
* + - Simple
    - Architecture neutral
    - Object oriented
    - Portable
    - Distributed
    - High performance
    - Interpreted
    - Multithreaded
    - Robust
    - Dynamic
    - Secure

With most programming languages, you either compile or interpret a program so that you can run it on your computer. The Java programming language is unusual in that a program is both compiled and interpreted. With the compiler, first you translate a program into an intermediate language called *Java byte codes* —the platform-independent codes interpreted by the interpreter on the Java platform. The interpreter parses and runs each Java byte code instruction on the computer. Compilation happens just once; interpretation occurs each time the program is executed. The following figure illustrates how this works.



#### FIG 2.1 Working of a compiler

You can think of Java byte codes as the machine code instructions for the *Java Virtual Machine* (Java VM). Every Java interpreter, whether it’s a development tool or a Web browser that can run applets, is an implementation of the Java VM. Java byte codes help make ―write once, run anywhere‖ possible. You can compile your program into byte codes on any platform that has a Java compiler. The byte codes can then be run on any implementation of the Java VM. That means that as long as a computer has a Java VM, the same program written in the Java programming language can run on Windows 2000, a Solaris workstation, or on an iMac.



#### FIG 2.2 Java Compiler

**The Java Platform**

A *platform* is the hardware or software environment in which a program runs. We’ve already mentioned some of the most popular platforms like Windows 2000, Linux, Solaris, and MacOS. Most platforms can be described as a combination of the operating system and hardware. The Java platform differs from most other platforms in that it’s a software-only platform that runs on top of other hardware-based platforms.

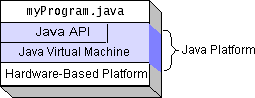
The Java platform has two components:

 The *Java Virtual Machine* (Java VM)

 The *Java Application Programming Interface* (Java API)

You’ve already been introduced to the Java VM. It’s the base for the Java platform and is ported onto various hardware-based platforms.

The Java API is a large collection of ready-made software components that provide many useful capabilities, such as graphical user interface (GUI) widgets. The Java API is grouped into libraries of related classes and interfaces; these libraries are known as *packages*. The next section, What Can Java Technology Do? Highlights what functionality some of the packages in the Java API provide.

The following figure depicts a program that’s running on the Java platform. As the figure shows, the Java API and the virtual machine insulate the program from the hardware.

#### FIG2.3. program running on a java platform

Native code is code that after you compile it, the compiled code runs on a specific hardware platform. As a platform-independent environment, the Java platform can be a bit slower than native code. However, smart compilers, well-tuned interpreters, and just-in-time byte code compilers can bring performance close to that of native code without threatening portability.

#### What Can Java Technology Do?

The most common types of programs written in the Java programming language are *applets* and *applications*. If you’ve surfed the Web, you’re probably already familiar with applets. An applet is a program that adheres to certain conventions that allow it to run within a Java-enabled browser.

However, the Java programming language is not just for writing cute, entertaining applets for the Web. The general-purpose, high-level Java programming language is also a

powerful software platform. Using the generous API, you can write many types of programs.

An application is a standalone program that runs directly on the Java platform. A special kind of application known as a *server* serves and supports clients on a network. Examples of servers are Web servers, proxy servers, mail servers, and print servers. Another specialized program is a *servlet*. A servlet can almost be thought of as an applet that runs on the server side. Java Servlets are a popular choice for building interactive web applications, replacing the use of CGI scripts. Servlets are similar to applets in that they are runtime extensions of applications. Instead of working in browsers, though, servlets run within Java Web servers, configuring or tailoring the server.

How does the API support all these kinds of programs? It does so with packages of software components that provides a wide range of functionality. Every full implementation of the Java platform gives you the following features:

 **The essentials**: Objects, strings, threads, numbers, input and output, data structures, system properties, date and time, and so on.

 **Applets**: The set of conventions used by applets.

 **Networking**: URLs, TCP (Transmission Control Protocol), UDP (User Data gram Protocol) sockets, and IP (Internet Protocol) addresses.

 **Internationalization**: Help for writing programs that can be localized for users worldwide. Programs can automatically adapt to specific locales and be displayed in the appropriate language.

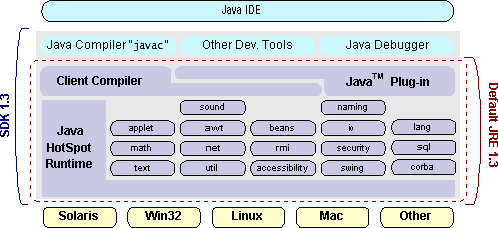
 **Security**: Both low level and high level, including electronic signatures, public and private key management, access control, and certificates.

 **Software components**: Known as JavaBeansTM, can plug into existing component architectures.

 **Object serialization**: Allows lightweight persistence and communication via Remote Method Invocation (RMI).

 **Java Database Connectivity (JDBCTM)**: Provides uniform access to a wide range of relational databases.

The Java platform also has APIs for 2D and 3D graphics, accessibility, servers, collaboration, telephony, speech, animation, and more. The following figure depicts what is included in the Java 2 SDK.



#### FIG 2.4 Java2 SDK

**How Will Java Technology Change My Life?**

We can’t promise you fame, fortune, or even a job if you learn the Java programming language. Still, it is likely to make your programs better and requires less effort than other languages. We believe that Java technology will help you do the following:

 **Get started quickly**: Although the Java programming language is a powerful object-oriented language, it’s easy to learn, especially for programmers already familiar with C or C++.

 **Write less code**: Comparisons of program metrics (class counts, method counts, and so on) suggest that a program written in the Java programming language can be four times smaller than the same program in C++.

 **Write better code**: The Java programming language encourages good coding practices, and its garbage collection helps you avoid memory leaks. Its object orientation, its JavaBeans component architecture, and its wide- ranging, easily extendible API let you reuse other people’s tested code and introduce fewer bugs.

 **Develop programs more quickly**: Your development time may be as much as twice as fast versus writing the same program in C++. Why? You write fewer lines of code and it is a simpler programming language than C++.

#### ODBC

 **Avoid platform dependencies with 100% Pure Java**: You can keep your program portable by avoiding the use of libraries written in other languages. The 100% Pure JavaTM Product Certification Program has a repository of historical process manuals, white papers, brochures, and similar materials online.

 **Write once, run anywhere**: Because 100% Pure Java programs are compiled into machine-independent byte codes, they run consistently on any Java platform.

 **Distribute software more easily**: You can upgrade applets easily from a central server. Applets take advantage of the feature of allowing new classes to be loaded ―on the fly,‖ without recompiling the entire program.

Microsoft Open Database Connectivity (ODBC) is a standard programming

interface for application developers and database systems providers. Before ODBC became a *de facto* standard for Windows programs to interface with database systems, programmers had to use proprietary languages for each database they wanted to connect to. Now, ODBC has made the choice of the database system almost irrelevant from a coding perspective, which is as it should be. Application developers have much more important things to worry about than the syntax that is needed to port their program from one database to another when business needs suddenly change.

Through the ODBC Administrator in Control Panel, you can specify the particular database that is associated with a data source that an ODBC application program is written to use. Think of an ODBC data source as a door with a name on it. Each door will lead you to a particular database. For example, the data source named Sales Figures might be a SQL Server database, whereas the Accounts Payable data source could refer to an Access database. The physical database referred to by a data source can reside anywhere on the LAN.

The ODBC system files are not installed on your system by Windows 95. Rather, they are installed when you setup a separate database application, such as SQL Server Client or Visual Basic 4.0. When the ODBC icon is installed in Control Panel, it uses a file called ODBCINST.DLL. It is also possible to administer your ODBC data sources through a stand-alone program called ODBCADM.EXE. There is a 16-bit and a 32-bit

version of this program and each maintains a separate list of ODBC data sources.

From a programming perspective, the beauty of ODBC is that the application can be written to use the same set of function calls to interface with any data source, regardless of the database vendor. The source code of the application doesn’t change whether it talks to Oracle or SQL Server. We only mention these two as an example. There are ODBC drivers available for several dozen popular database systems. Even Excel spreadsheets and plain text files can be turned into data sources. The operating system uses the Registry information written by ODBC Administrator to determine which low-level ODBC drivers are needed to talk to the data source (such as the interface to Oracle or SQL Server). The loading of the ODBC drivers is transparent to the ODBC application program. In a client/server environment, the ODBC API even handles many of the network issues for the application programmer.

The advantages of this scheme are so numerous that you are probably thinking there must be some catch. The only disadvantage of ODBC is that it isn’t as efficient as talking directly to the native database interface. ODBC has had many detractors make the charge that it is too slow. Microsoft has always claimed that the critical factor in performance is the quality of the driver software that is used. In our humble opinion, this is true. The availability of good ODBC drivers has improved a great deal recently. And anyway, the criticism about performance is somewhat analogous to those who said that compilers would never match the speed of pure assembly language. Maybe not, but the compiler (or ODBC) gives you the opportunity to write cleaner programs, which means you finish sooner. Meanwhile, computers get faster every year.

#### JDBC

In an effort to set an independent database standard API for Java; Sun

Microsystems developed Java Database Connectivity, or JDBC. JDBC offers a generic SQL database access mechanism that provides a consistent interface to a variety of RDBMSs. This consistent interface is achieved through the use of ―plug-in‖ database connectivity modules, or *drivers*. If a database vendor wishes to have JDBC support, he or she must provide the driver for each platform that the database and Java run on.

To gain a wider acceptance of JDBC, Sun based JDBC’s framework on ODBC. As you discovered earlier in this chapter, ODBC has widespread support on a variety of platforms. Basing JDBC on ODBC will allow vendors to bring JDBC drivers to market much faster than developing a completely new connectivity solution.

JDBC was announced in March of 1996. It was released for a 90 day public review that ended June 8, 1996. Because of user input, the final JDBC v1.0 specification was released soon after.

The remainder of this section will cover enough information about JDBC for you to know what it is about and how to use it effectively. This is by no means a complete overview of JDBC. That would fill an entire book.

#### JDBC Goals

Few software packages are designed without goals in mind. JDBC is one that, because of its many goals, drove the development of the API. These goals, in conjunction with early reviewer feedback, have finalized the JDBC class library into a solid framework for building database applications in Java.

The goals that were set for JDBC are important. They will give you some insight as to why certain classes and functionalities behave the way they do. The eight design goals for JDBC are as follows:

#### SQL Level API

The designers felt that their main goal was to define a SQL interface for Java. Although not the lowest database interface level possible, it is at a low enough level for higher-level tools and APIs to be created. Conversely, it is at a high enough level for application programmers to use it confidently. Attaining this goal allows for future tool vendors to ―generate‖ JDBC code and to hide many of JDBC’s complexities from the end user.

#### SQL Conformance

SQL syntax varies as you move from database vendor to database vendor. In an effort to support a wide variety of vendors, JDBC will allow any query statement to be passed through it to the underlying database driver. This allows the connectivity module to handle non-standard functionality in a manner that is suitable for its users.

1. **JDBC must be implemental on top of common database interfaces** The JDBC SQL API must ―sit‖ on top of other common SQL level APIs. This goal allows JDBC to use existing ODBC level drivers by the use of a software

interface. This interface would translate JDBC calls to ODBC and vice versa.

#### Provide a Java interface that is consistent with the rest of the Java system

Because of Java’s acceptance in the user community thus far, the designers feel that they should not stray from the current design of the core Java system.

#### Keep it simple

This goal probably appears in all software design goal listings. JDBC is no exception. Sun felt that the design of JDBC should be very simple, allowing for only one method of completing a task per mechanism. Allowing duplicate functionality only serves to confuse the users of the API.

#### Use strong, static typing wherever possible

Strong typing allows for more error checking to be done at compile time; also, less error appear at runtime.

#### Keep the common cases simple

Because more often than not, the usual SQL calls used by the programmer are simple SELECT’s, INSERT’s, DELETE’s and UPDATE’s, these queries should be simple to perform with JDBC. However, more complex SQL statements should also be possible.

Finally we decided to proceed the implementation using Java Networking.

And for dynamically updating the cache table we go for MS Access database Java has two things: a programming language and a platform.

Java is a high-level programming language that is all of the following

Simple Architecture-neutral Object-oriented Portable

Distributed High-performance

Interpreted multithreaded Robust Dynamic

Secure

Java is also unusual in that each Java program is both compiled and interpreted. With a compile you translate a Java program into an intermediate language called Java byte codes the platform-independent code instruction is passed and run on the computer. Compilation happens just once; interpretation occurs each time the program is executed. The figure illustrates how this works.

**Java Program**

**Compilers**

**My Program**

**Interpreter**

You can think of Java byte codes as the machine code instructions for the Java Virtual Machine (Java VM). Every Java interpreter, whether it’s a Java development tool or a Web browser that can run Java applets, is an implementation of the Java VM. The Java VM can also be implemented in hardware.

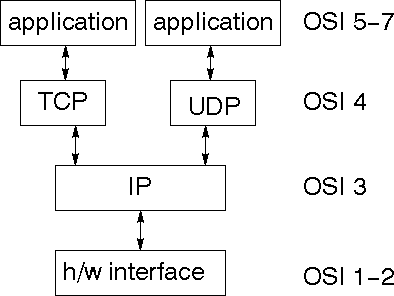
Java byte codes help make ―write once, run anywhere‖ possible. You can compile your Java program into byte codes on my platform that has a Java compiler. The byte codes can then be run any implementation of the Java VM.

For example, the same Java program can run Windows NT, Solaris, and Macintosh.

#### Networking

**TCP/IP stack**

The TCP/IP stack is shorter than the OSI one:



TCP is a connection-oriented protocol; UDP (User Datagram Protocol) is a connectionless protocol.

#### IP datagram’s

The IP layer provides a connectionless and unreliable delivery system. It considers each datagram independently of the others. Any association between datagram must be supplied by the higher layers. The IP layer supplies a checksum that includes its own header. The header includes the source and destination addresses. The IP layer handles routing through an Internet. It is also responsible for breaking up large datagram into smaller ones for transmission and reassembling them at the other end.

#### UDP

UDP is also connectionless and unreliable. What it adds to IP is a checksum for the contents of the datagram and port numbers. These are used to give a client/server model - see later.

#### TCP

TCP supplies logic to give a reliable connection-oriented protocol above IP. It provides a virtual circuit that two processes can use to communicate.

#### Internet addresses

In order to use a service, you must be able to find it. The Internet uses an address scheme for machines so that they can be located. The address is a 32 bit integer which gives the IP address. This encodes a network ID and more addressing. The network ID falls into various classes according to the size of the network address.

#### Network address

Class A uses 8 bits for the network address with 24 bits left over for other addressing. Class B uses 16 bit network addressing. Class C uses 24 bit network addressing and class D uses all 32.

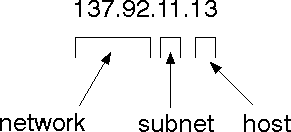
#### Subnet address

Internally, the UNIX network is divided into sub networks. Building 11 is currently on one sub network and uses 10-bit addressing, allowing 1024 different hosts.

#### Host address

8 bits are finally used for host addresses within our subnet. This places a limit of 256 machines that can be on the subnet.

#### Total address



The 32 bit address is usually written as 4 integers separated by dots.

#### Port addresses

A service exists on a host, and is identified by its port. This is a 16 bit number. To send a message to a server, you send it to the port for that service of the host that it is running on. This is not location transparency! Certain of these ports are "well known".

#### Sockets

A socket is a data structure maintained by the system to handle network connections. A socket is created using the call socket. It returns an integer that is like a file descriptor. In fact, under Windows, this handle can be used with Read File and Write File functions.

#include <sys/types.h> #include <sys/socket.h>

int socket(int family, int type, int protocol);

Here "family" will be AF\_INET for IP communications, protocol will be zero, and type will depend on whether TCP or UDP is used. Two processes wishing to communicate over a network create a socket each. These are similar to two ends of a pipe - but the actual pipe does not yet exist.

#### JFree Chart

JFreeChart is a free 100% Java chart library that makes it easy for developers to display professional quality charts in their applications. JFreeChart's extensive feature set includes:

A consistent and well-documented API, supporting a wide range of chart types;

A flexible design that is easy to extend, and targets both server-side and client-side applications;

Support for many output types, including Swing components, image files (including PNG and JPEG), and vector graphics file formats (including PDF, EPS and SVG);

JFreeChart is "open source" or, more specifically, [free software](http://www.gnu.org/philosophy/free-sw.html). It is distributed under the terms of the [GNU Lesser General Public Licence](http://www.gnu.org/licenses/lgpl.html) (LGPL), which permits use in proprietary applications.

#### Map Visualizations

Charts showing values that relate to geographical areas. Some examples include: (a) population density in each state of the United States, (b) income per capita for each country in Europe, (c) life expectancy in each country of the world. The tasks in this project include:

Sourcing freely redistributable vector outlines for the countries of the world, states/provinces in particular countries (USA in particular, but also other areas);

Creating an appropriate dataset interface (plus default implementation), a rendered, and integrating this with the existing XYPlot class in JFreeChart;

Testing, documenting, testing some more, documenting some more.

#### Time Series Chart Interactivity

Implement a new (to JFreeChart) feature for interactive time series charts --- to display a separate control that shows a small version of ALL the time series data, with a sliding "view" rectangle that allows you to select the subset of the time series data to display in the main chart.

#### Dashboards

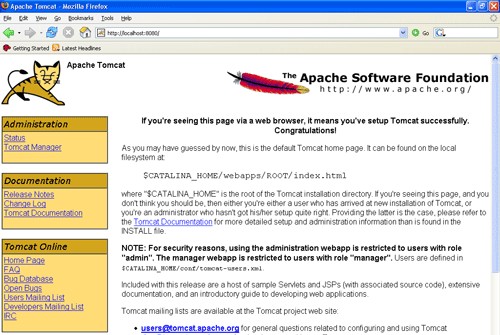
There is currently a lot of interest in dashboard displays. Create a flexible dashboard mechanism that supports a subset of JFreeChart chart types (dials, pies, thermometers, bars, and lines/time series) that can be delivered easily via both Java Web Start and an applet.

#### Property Editors

The property editor mechanism in JFreeChart only handles a small subset of the properties that can be set for charts. Extend (or reimplement) this mechanism to provide greater end-user control over the appearance of the charts.

#### Tomcat 6.0 web server

Tomcat is an open source web server developed by Apache Group. Apache Tomcat is the servlet container that is used in the official Reference Implementation for the Java Servlet and JavaServer Pages technologies. The Java Servlet and JavaServer Pages specifications are developed by Sun under the Java Community Process. Web Servers like Apache Tomcat support only web components while an application server supports web components as well as business components (BEAs Weblogic, is one of the popular application server).To develop a web application with jsp/servlet install any web server like JRun, Tomcat etc to run your application.



**Fig2.5 Tomcat Web server**

**SQL-SERVER**

The OLAP Services feature available in SQL Server version 7.0 is now called SQL Server 2000 Analysis Services. The term OLAP Services has been replaced with the term Analysis Services. Analysis Services also includes a new data mining component. The Repository component available in SQL Server version 7.0 is now called Microsoft SQL Server 2000 Meta Data Services. References to the component now use the term Meta Data Services. The term repository is used only in reference to the repository engine within Meta Data Services. SQL-SERVER database consist of following type of objects:

1. TABLE
2. QUERY
3. FORM
4. REPORT
5. MACRO

#### TABLE:

A database is a collection of data about a specific topic.

#### VIEWS OF TABLE:

We can work with a table in two types,

1. Design View
2. Datasheet View

#### Design View

To build or modify the structure of a table we work in the table design view. We can specify what kind of data will be hold.

#### Datasheet View

To add, edit or analyses the data itself we work in tables datasheet view mode.

#### QUERY:

A query is a question that has to be asked the data. Access gathers data that answers the question from one or more table. The data that make up the answer is either dynaset (if you edit it) or a snapshot (it cannot be edited).

**CHAPTER 3**

# REQUIREMENT ANALYSIS

## FUNCTIONAL REQUIREMENTS

In software engineering, a functional requirement defines a function of a software system or its component. A function is described as a set of inputs, the behavior, and outputs. Functional requirements may be calculations, technical details, data manipulation and processing and other specific functionality that define what a system is supposed to accomplish. Behavioral requirements describing all the cases where the system uses the functional requirements are captured in use cases.

Here, the system has to perform the following tasks:

 Take user-id and password, match it with corresponding database entries. If a match is found then continue else raise an error message.

 User gives the query and relevant results will be displayed

## NON-FUNCTIONAL REQUIREMENTS

In systems engineering and requirements engineering, a non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors. This should be contrasted with functional requirements that define specific behavior or functions. The plan for implementing functional requirements is detailed in the system design. The plan for implementing non- functional requirements is detailed in the system architecture.

Other terms for non-functional requirements are "constraints", "quality attributes", "quality goals", "quality of service requirements" and "non-behavioral requirements".

Some of the quality attributes are as follows:

### ACCESSIBILITY:

Accessibility is a general term used to describe the degree to which a product, device, service, or environment is accessible by as many people as possible.

In our project people who have registered can log in and search for their queries.

User interface is simple and efficient and easy to use.

### MAINTAINABILITY:

In software engineering, maintainability is the ease with which a software product can be modified in order to:

 Correct defects

 Meet new requirements

New functionalities can be added in the project based on the user requirements just by adding the appropriate files to existing project using ASP.net and C# programming languages.

Since the programming is very simple, it is easier to find and correct the defects and to make the changes in the project.

### SCALABILITY:

System is capable of handling increase total throughput under an increased load when resources (typically hardware) are added.

System can work normally under situations such as low bandwidth and large number of users.

### PORTABILITY:

Portability is one of the key concepts of high-level programming. Portability is the software code base feature to be able to reuse the existing code instead of creating new code when moving software from an environment to another.

Project can be executed under different operation conditions provided it meet its minimum configurations. Only system files and dependant assemblies would have to be configured in such case.

## HARDWARE REQUIREMENTS

Processor : Any Processor above 500 MHz

RAM : 512Mb

Hard Disk : 10 GB

Input device : Standard Keyboard and Mouse Output device : VGA and High Resolution Monitor

## SOFTWARE REQUIREMENTS

Operating System :Windows95/98/2000/XP Application Server : Tomcat5.0/6.X

Front End : HTML, Java, Jsp

Scripts : JavaScript

Server side Script : Java Server Pages Database : Mysql

Database Connectivity : JDBC.

## DESIGN GOALS

# DESIGN

## CHAPTER 4

To enable secure outsourcing of file under the aforementioned model, our mechanism design should achieve the following security and performance guarantees:

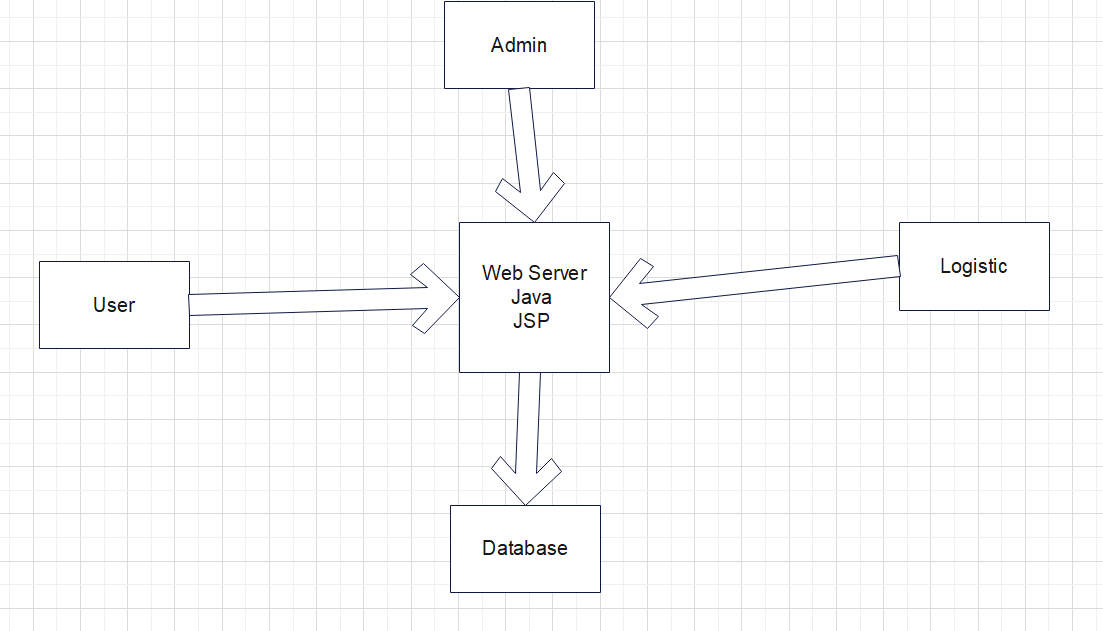
### INPUT/OUTPUT PRIVACY

No sensitive information from the customer’s private data can be derived by the cloud server during performing the encryption and transfer.

### EFFICIENCY

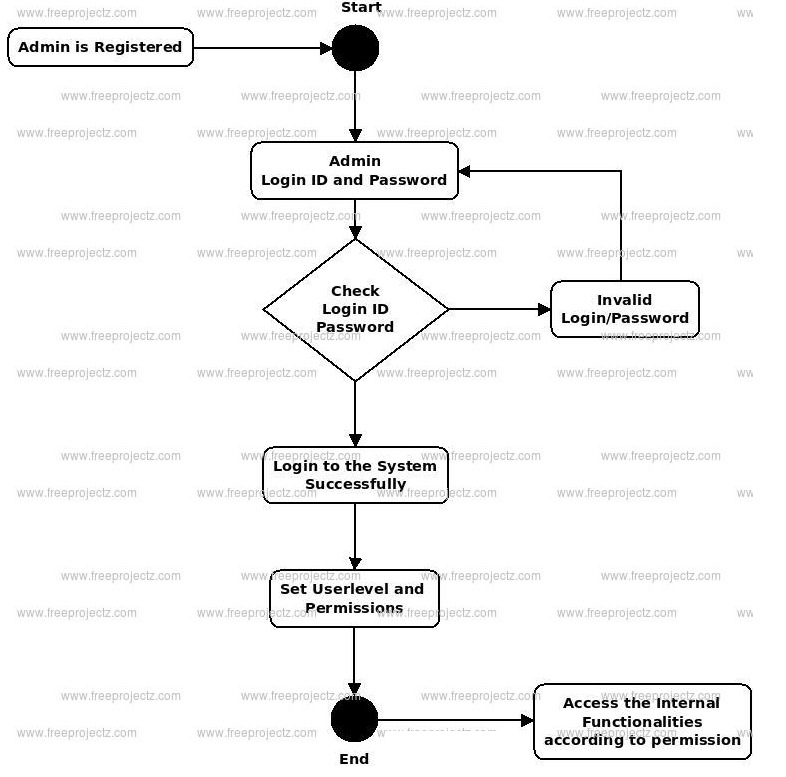
The local computations done by customer should be substantially less than. The computation burden on the cloud server should be within the comparable time complexity of existing practical algorithms for encryption and decryption of files.

## SYSTEM ARCHITECTURE

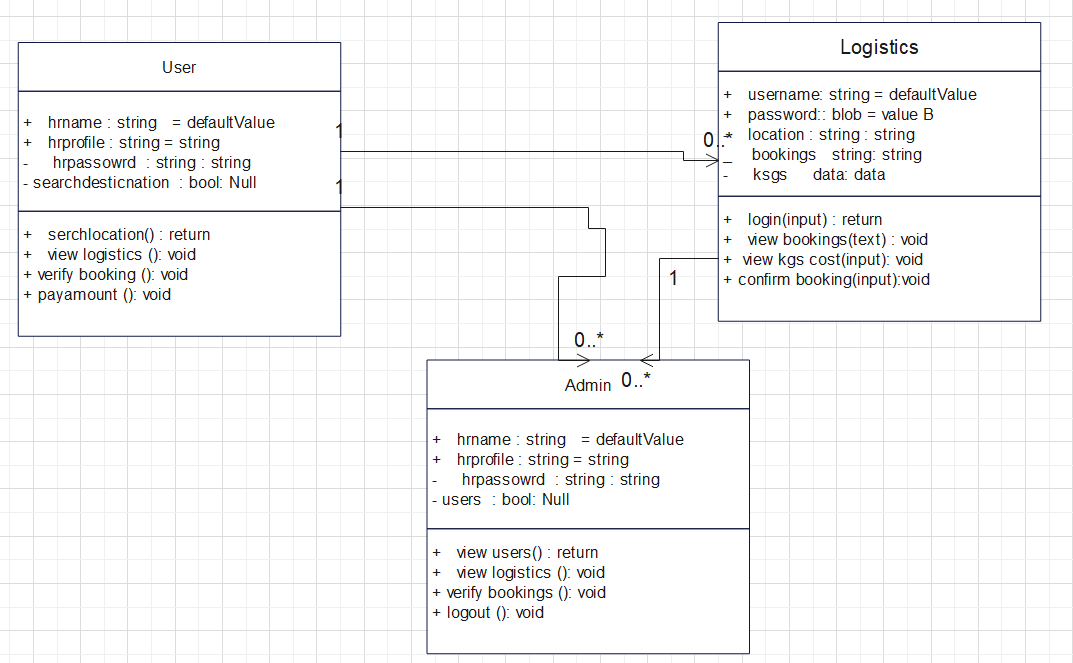
****

**Fig 4.1 System Architecture**

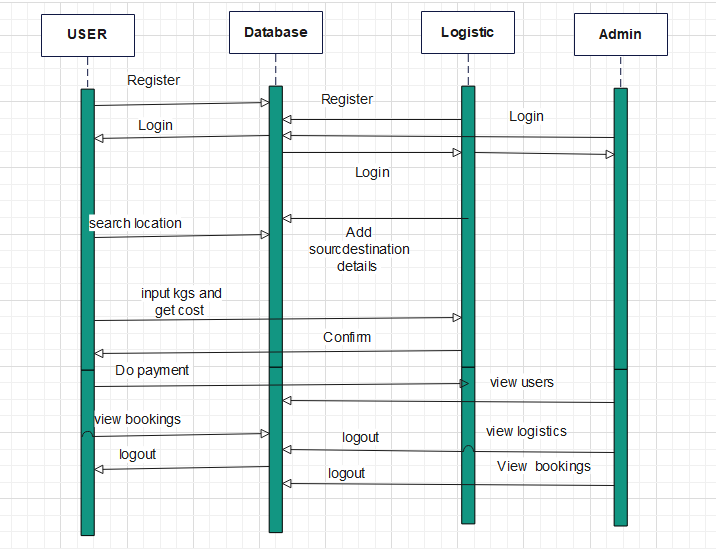
## ACTIVITY DIAGRAM



## CLASS DIAGRAM



## SEQUENCE DIAGRAM

****

## USE CASE DIAGRAM



# IMPLEMENTATIONCHAPTER 5

Implementation is the stage of the project when the theoretical design is turned out into a working system. Thus it can be considered to be the most critical stage in achieving a successful new system and in giving the user, confidence that the new system will work and be effective.

The implementation stage involves careful planning, investigation of the existing system and it’s constraints on implementation, designing of methods to achieve changeover and evaluation of changeover methods.

Implementation is the stage of the project when the theoretical design is turned out into a working system. Thus it can be considered to be the most critical stage in achieving a successful new system and in giving the user, confidence that the new system will work and be effective.

The implementation stage involves careful planning, investigation of the existing system and it’s constraints on implementation, designing of methods to achieve changeover and evaluation of changeover methods.

## MODULE DESCRIPTION:

### User Module.

Using this module user will register with application with personal and logistic details and search for source and destination details and book slot with kgs and get cost for given kgs and send request to logistic person for confirmation and check bookings.

### Logistic Module.

Using this module logistic person will register with personal details and upload source and destination details with four sub routes and add total kgs and cost for kg for transport. Logistic module can confirm booking or user and view bookings.

# TESTING

## CHAPTER 6

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product it is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

## TYPES OF TESTS

* 1. **UNIT TESTING**

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

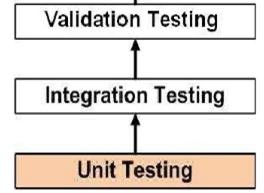
## INTEGRATION TESTING

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

## VALIDATION TESTING

An engineering validation test (EVT) is performed on first engineering prototypes, to ensure that the basic unit performs to design goals and specifications. It is important in identifying design problems, and solving them as early in the design cycle as possible, is the key to keeping projects on time and within budget. Too often, product design and performance problems are not detected until late in the product development cycle — when the product is ready to be shipped. The old adage holds true: It costs a penny to make a change in engineering, a dime in production and a dollar after a product is in the field.

Verification is a Quality control process that is used to evaluate whether or not a product, service, or system complies with regulations, specifications, or conditions imposed at the start of a development phase. Verification can be in development, scale- up, or production. This is often an internal process.

Validation is a Quality assurance process of establishing evidence that provides a high degree of assurance that a product, service, or system accomplishes its intended requirements. This often involves acceptance of fitness for purpose with end users and other product stakeholders. The testing process overview is as follows:

**Figure 6.1: The testing process**

## SYSTEM TESTING

System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of black box testing, and as such, should require no knowledge of the inner design of the code or logic.

As a rule, system testing takes, as its input, all of the "integrated" software components that have successfully passed integration testing and also the software system itself integrated with any applicable hardware system(s).

System testing is a more limited type of testing; it seeks to detect defects both within the "inter-assemblages" and also within the system as a whole.

System testing is performed on the entire system in the context of a Functional Requirement Specification(s) (FRS) and/or a System Requirement Specification (SRS).

System testing tests not only the design, but also the behavior and even the believed expectations of the customer. It is also intended to test up to and beyond the bounds defined in the software/hardware requirements specification(s).

## White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

## Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot ―see‖ into it. The test provides inputs and responds to outputs without considering how the software works.

## Unit Testing:

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

## Test strategy and approach

Field testing will be performed manually and functional tests will be written in

detail.

## Test objectives

 All field entries must work properly.

 Pages must be activated from the identified link.

 The entry screen, messages and responses must not be delayed.

## Features to be tested

 Verify that the entries are of the correct format  No duplicate entries should be allowed

 All links should take the user to the correct page.

## Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

## Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

**STEP 1) UNDERSTAND THE SOURCE CODE**

The first thing a tester will often do is learn and understand the source code of the application. Since white box testing involves the testing of the inner workings of an application, the tester must be very knowledgeable in the programming languages used in the applications they are testing. Also, the testing person must be highly aware of secure coding practices. Security is often one of the primary objectives of testing software. The tester should be able to find security issues and prevent attacks from hackers and naive users who might inject malicious code into the application either knowingly or unknowingly.

**Step 2) CREATE TEST CASES AND EXECUTE**

The second basic step to white box testing involves testing the application’s source code for proper flow and structure. One way is by writing more code to test the application’s source code. The tester will develop little tests for each process or series of processes in the application. This  method requires that the tester must have intimate knowledge of the code and is often done by the developer. Other methods include manual testing, trial and error testing and the use of testing tools as we will explain further on in this article.

Unit testing:

|  |  |
| --- | --- |
| Sl # Test Case : ­ | UTC­1 |
| Name of Test: ­ | Customer signup |
| Items being tested: ­ | Validation for signup ornot |
| Sample Input: ­ | Fill form |
| Expected output: ­ | Details stored in database if wrong details are given check validation |
| Actual output: ­ | Validation verified details stored in db |
| **Remarks: ­** | **Pass.** |

|  |  |
| --- | --- |
| Sl # Test Case : ­ | UTC­2 |
| Name of Test: ­ | Create super user |
| Items being tested: ­ | New super user created for java |
| Sample Input: ­ | Enter data from netbeans prompt |
| Expected output: ­ | New super user created |
| Actual output: ­ | Admin can login with super user |
| Remarks: ­ | pass |

**Integration Testing:**

Integration testing is a level of software testing where individual units are combined and tested as a group. The purpose of this level of testing is to expose faults in the interaction between integrated units. Test drivers and test stubs are used to assist in Integration Testing. Integration testing is defined as the testing of combined parts of an application to determine if they function correctly. It occurs after unit testing and before validation testing. Integration testing can be done in two ways: Bottom­up integration testing and Top­down integration testing.

* + 1. **Bottom­up Integration**

This testing begins with unit testing, followed by tests of progressively higher­level combinations of units called modules or builds.

* + 1. **Top­down Integration**

In this testing, the highest­level modules are tested first and progressively, lower­level modules are tested thereafter.

In a comprehensive software development environment, bottom­up testing is usually done first, followed by top­down testing. The process concludes with multiple tests of the complete application, preferably in scenarios designed to mimic actual situations. Table 6.5 shows the test cases for integration testing and their results

|  |  |
| --- | --- |
| Sl # Test Case : ­ | ITC­1 |
| Name of Test: ­ | Add product to cart |
| Item being tested: ­ | Are products added to cart |
| Sample Input: ­ | Click on add option on product |
| Expected output: ­ | Details shown in cart |
| Actual output: ­ | Check out with cost |
| Remarks: ­ | Pass. |

|  |  |
| --- | --- |
| Sl # Test Case : ­ | ITC­2 |
| Name of Test: ­ | Payment process |
| Item being tested: ­ | Details of address displayed for address |
| Sample Input: ­ | Click on check out |
| Expected output: ­ | Address form opened and details stored in database |
| Actual output: ­ | Displayed order details to admin |
| Remarks: ­ | Pass. |

**System testing**:

System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of black­box testing, and as such, should require no knowledge of the inner design of the code or logic. System testing is important because of the following reasons:

System testing is the first step in the Software Development Life Cycle, where the application is tested as a whole.

The application is tested thoroughly to verify that it meets the functional and technical specifications.

The application is tested in an environment that is very close to the production environment where the application will be deployed.

System testing enables us to test, verify, and validate both the business requirements as well as the application architecture.

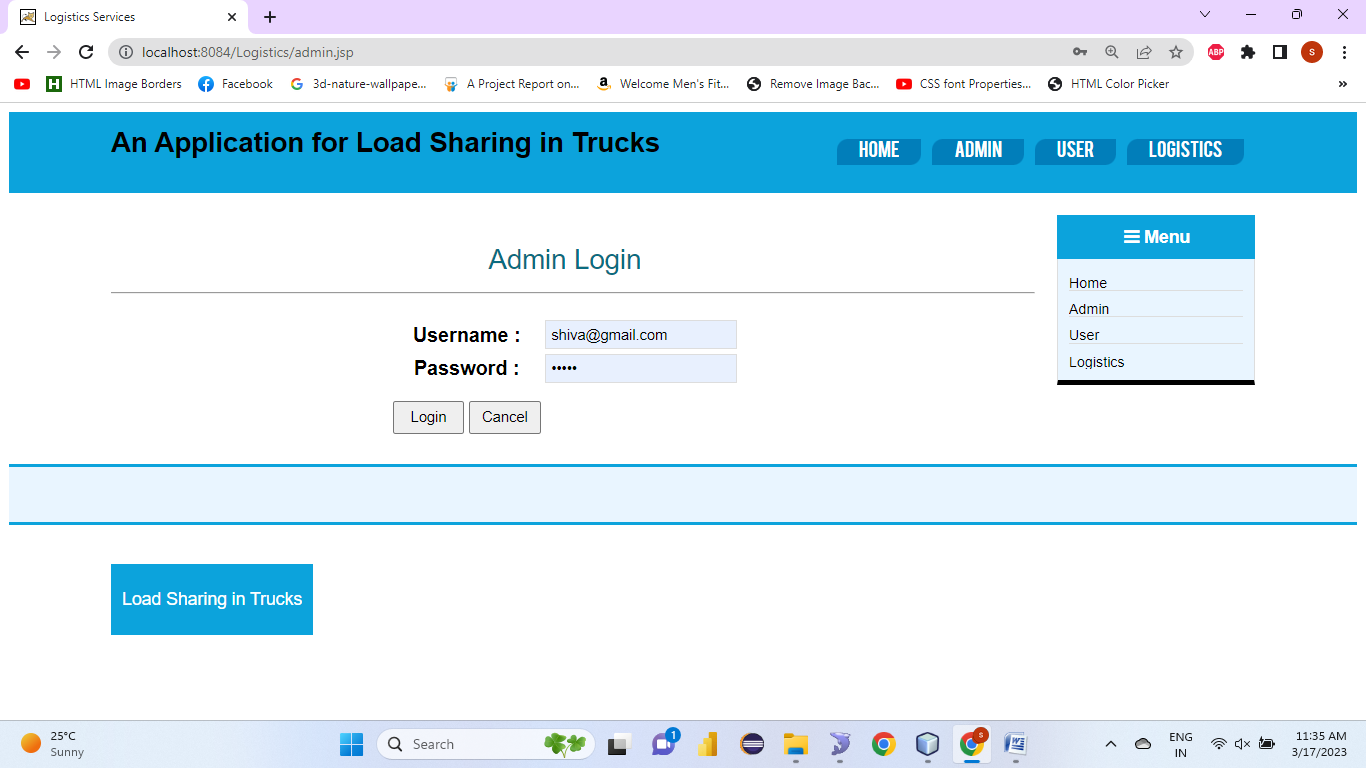
System Testing is shown in below tables

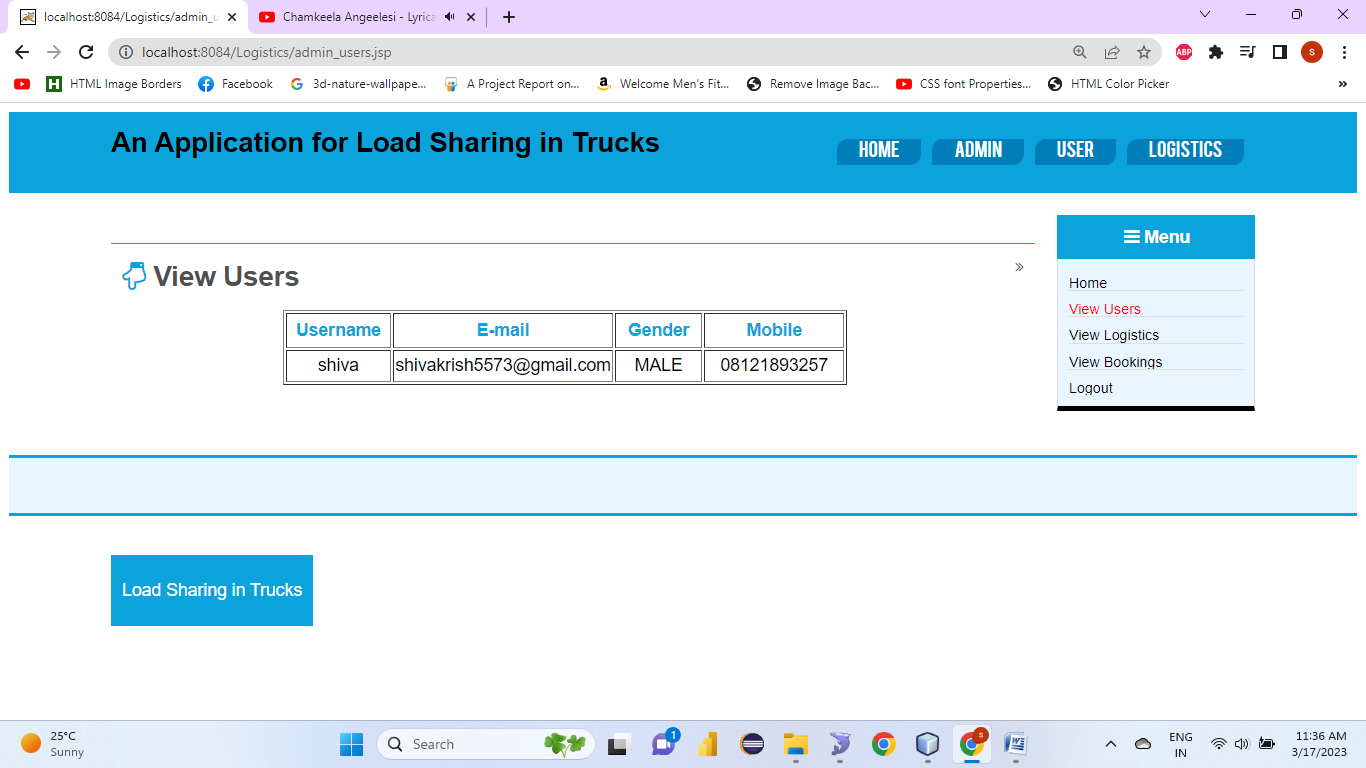
|  |  |
| --- | --- |
| Sl # Test Case : ­ | STC­1 |
| Name of Test: ­ | System testing in various versions of OS |
| Item being tested: ­ | OS compatibility. |
| Sample Input: ­ | Execute the program in windows XP/ Windows­7/8 |
| Expected output: ­ | Performance is better in windows­7 |
| Actual output: ­ | Same as expected output, performance is better in windows­7 |
| Remarks: ­ | Pass |

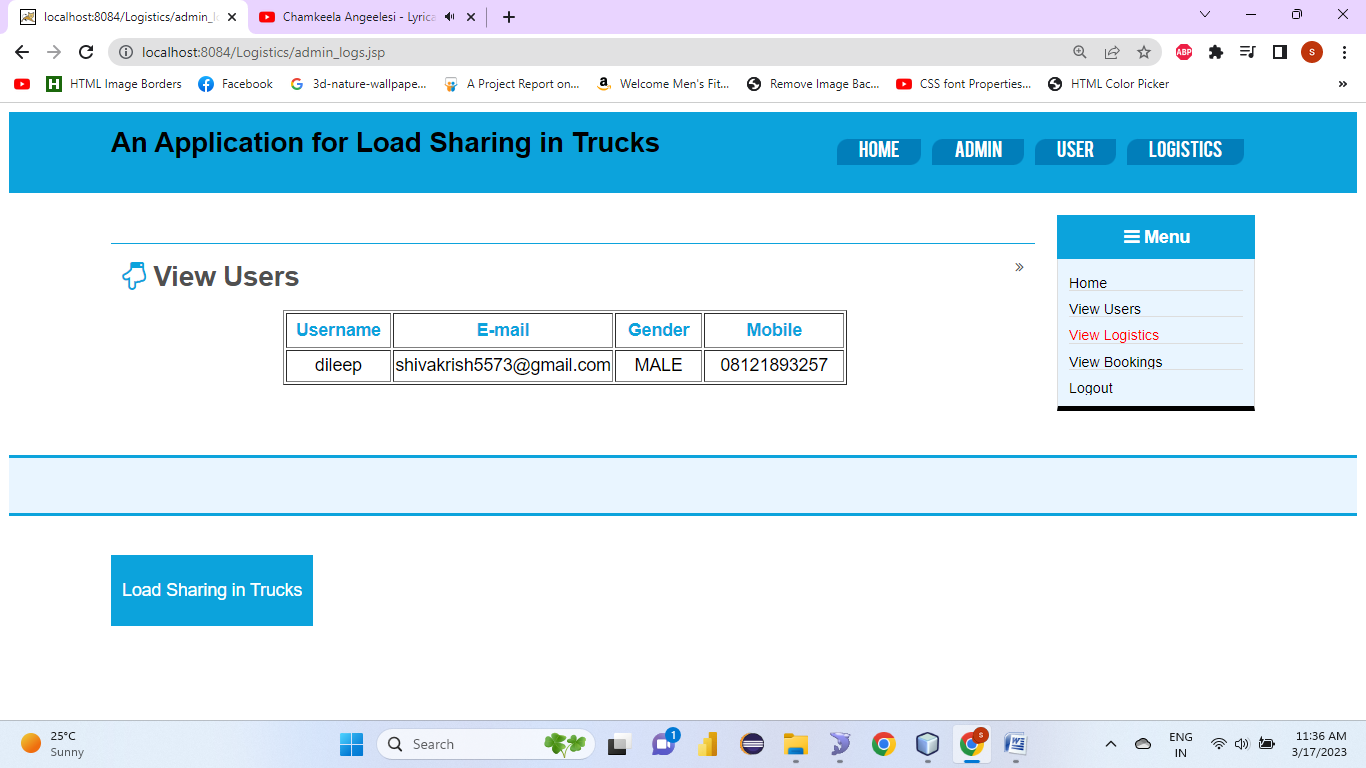
**CHAPTER 7**

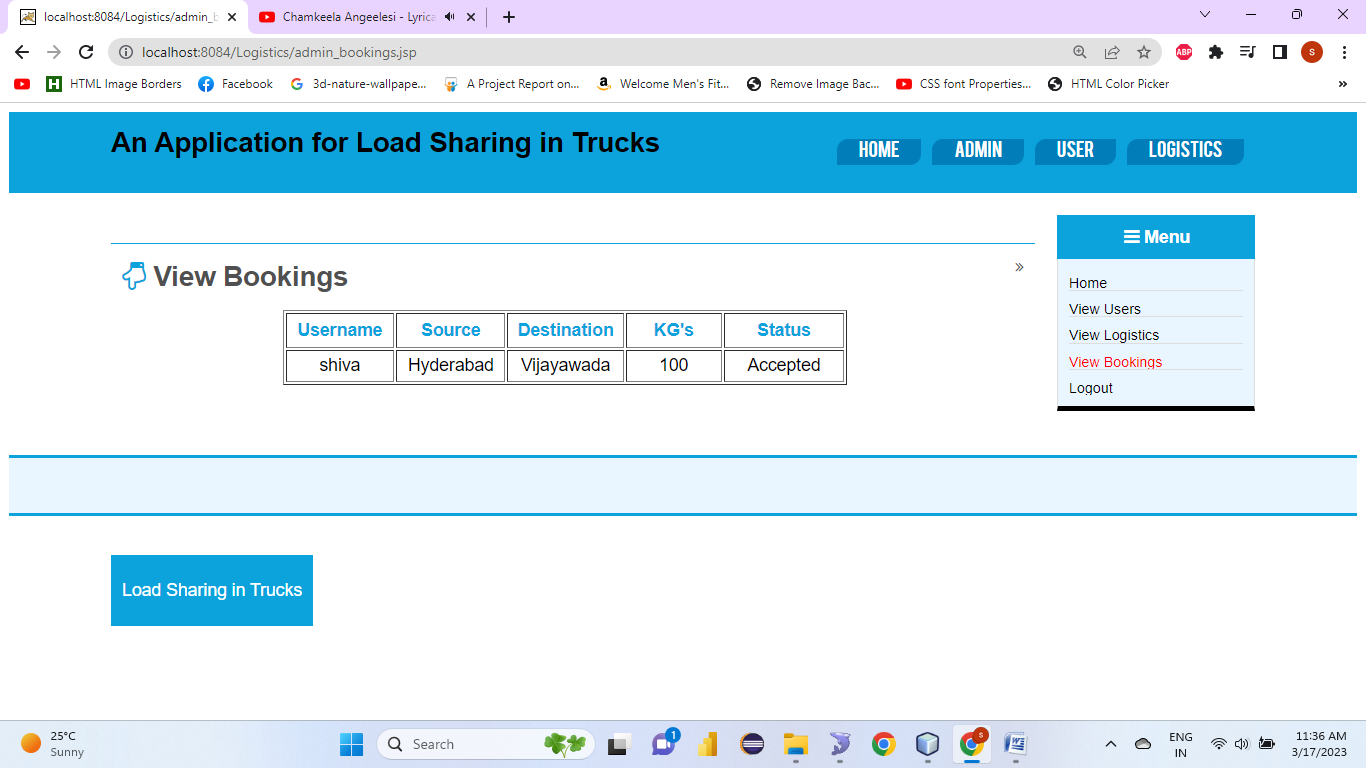
# SNAPSHOT



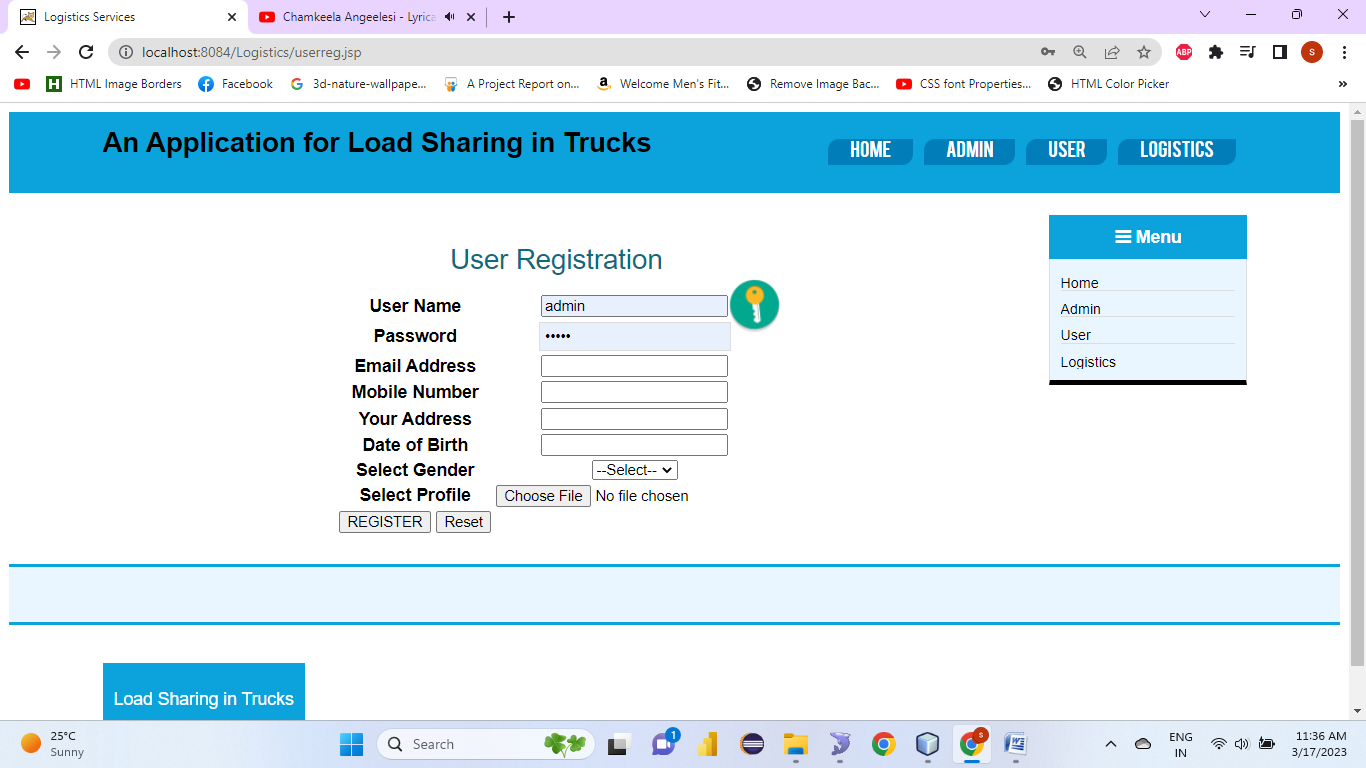




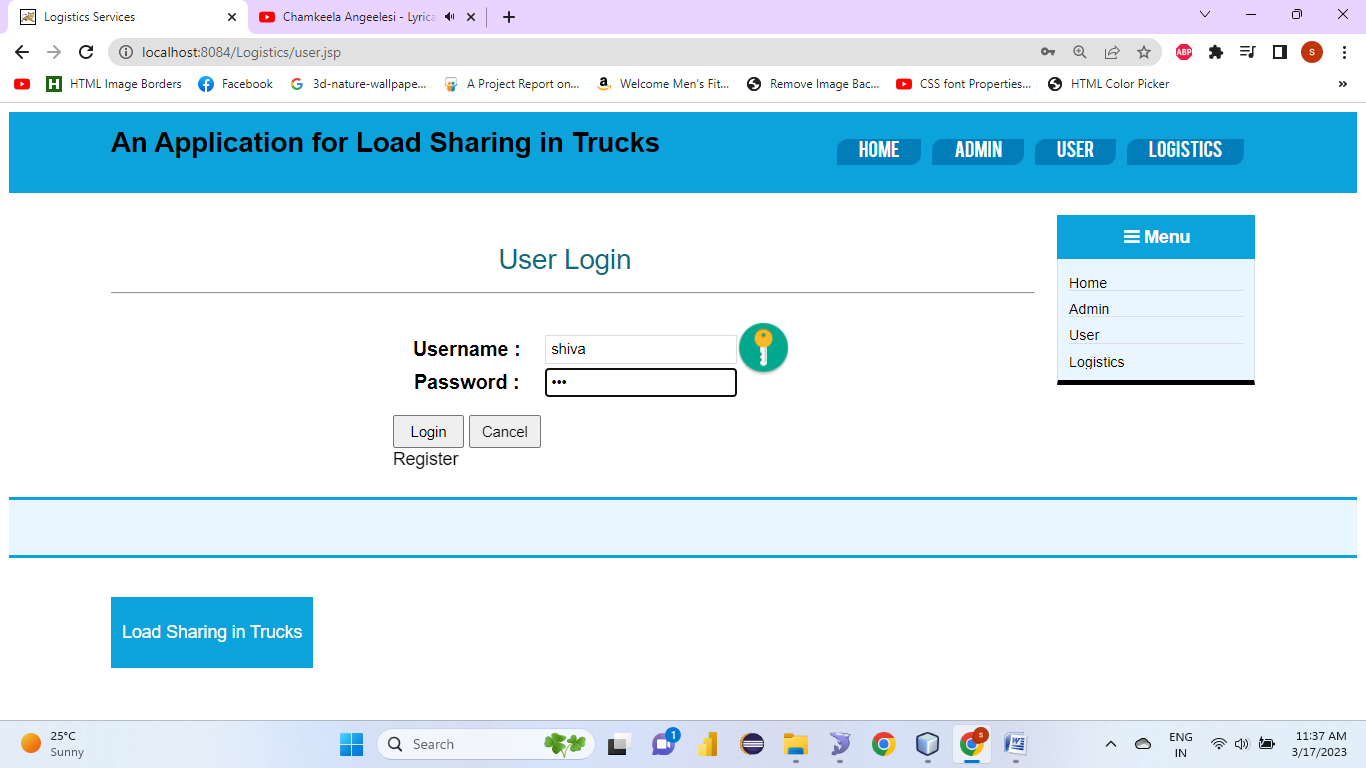




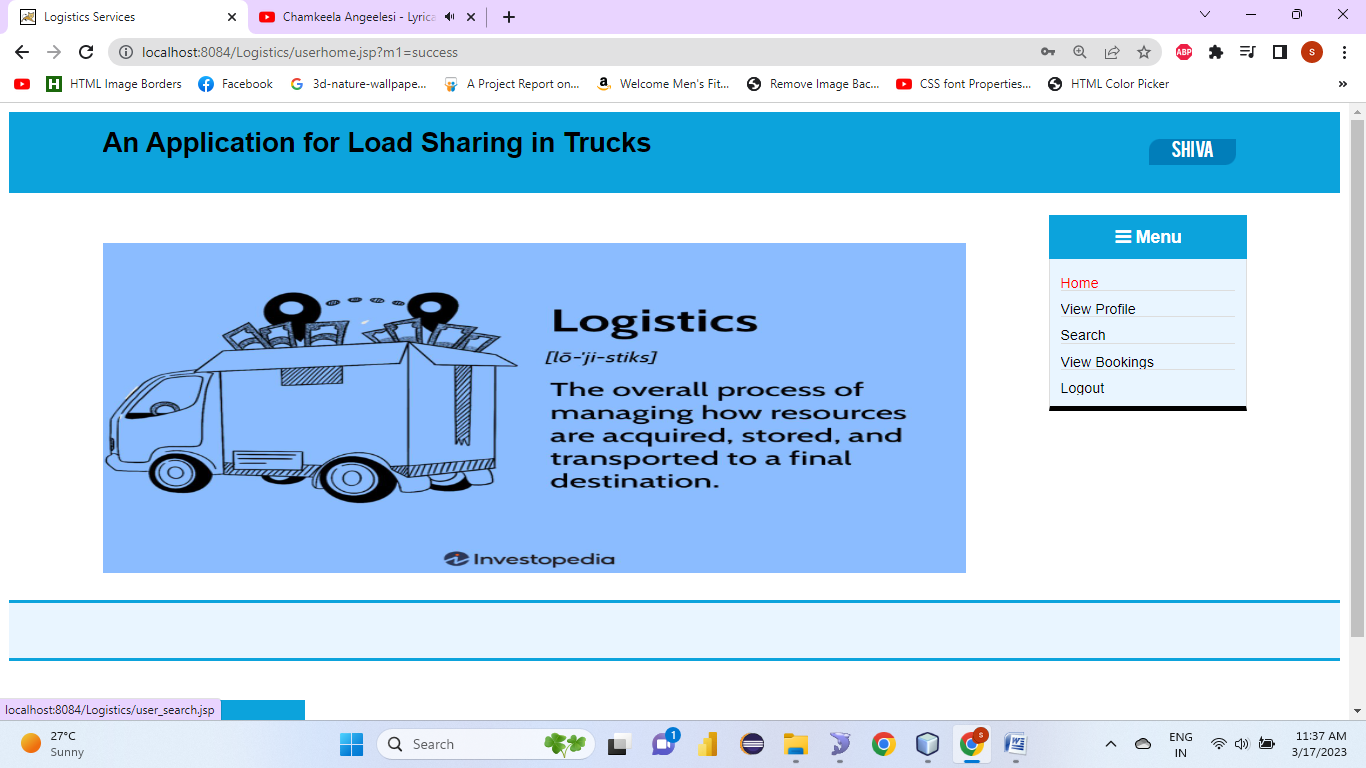
User registration:



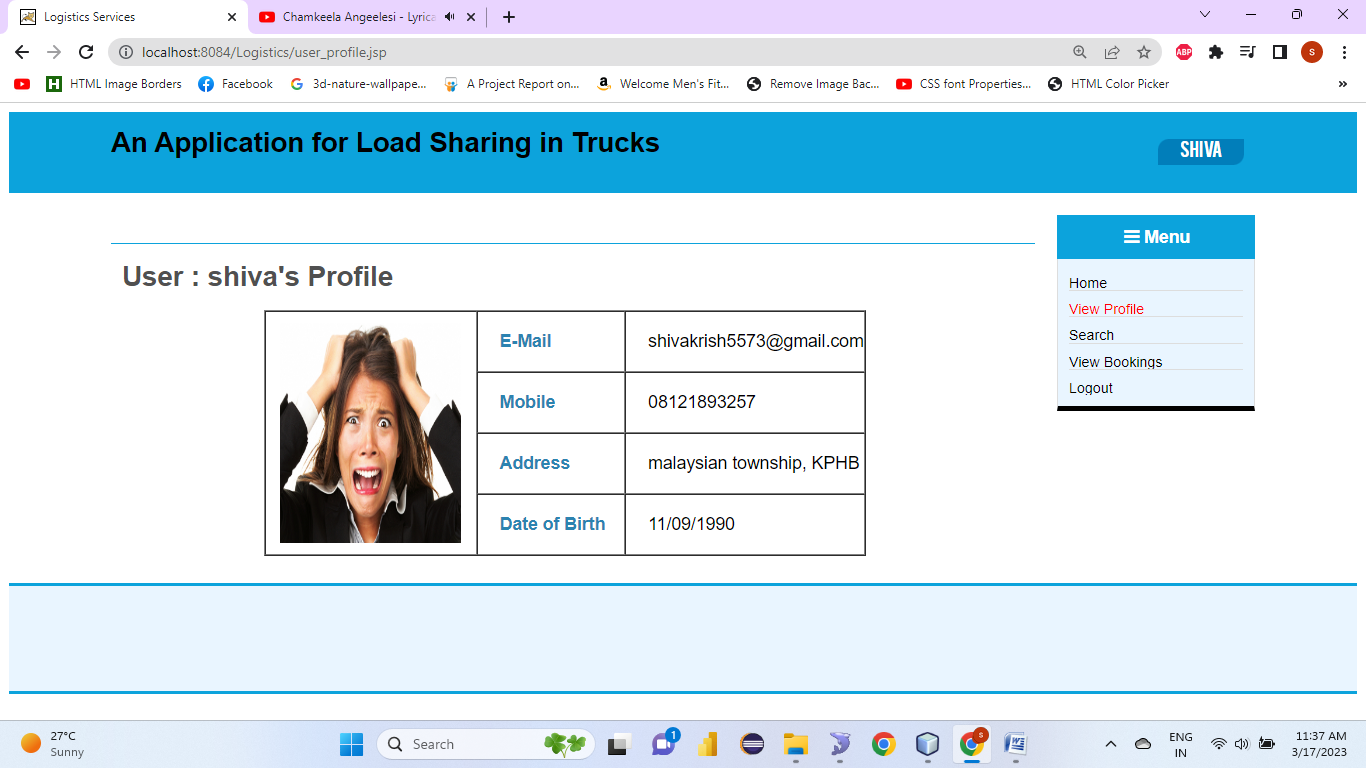
User login



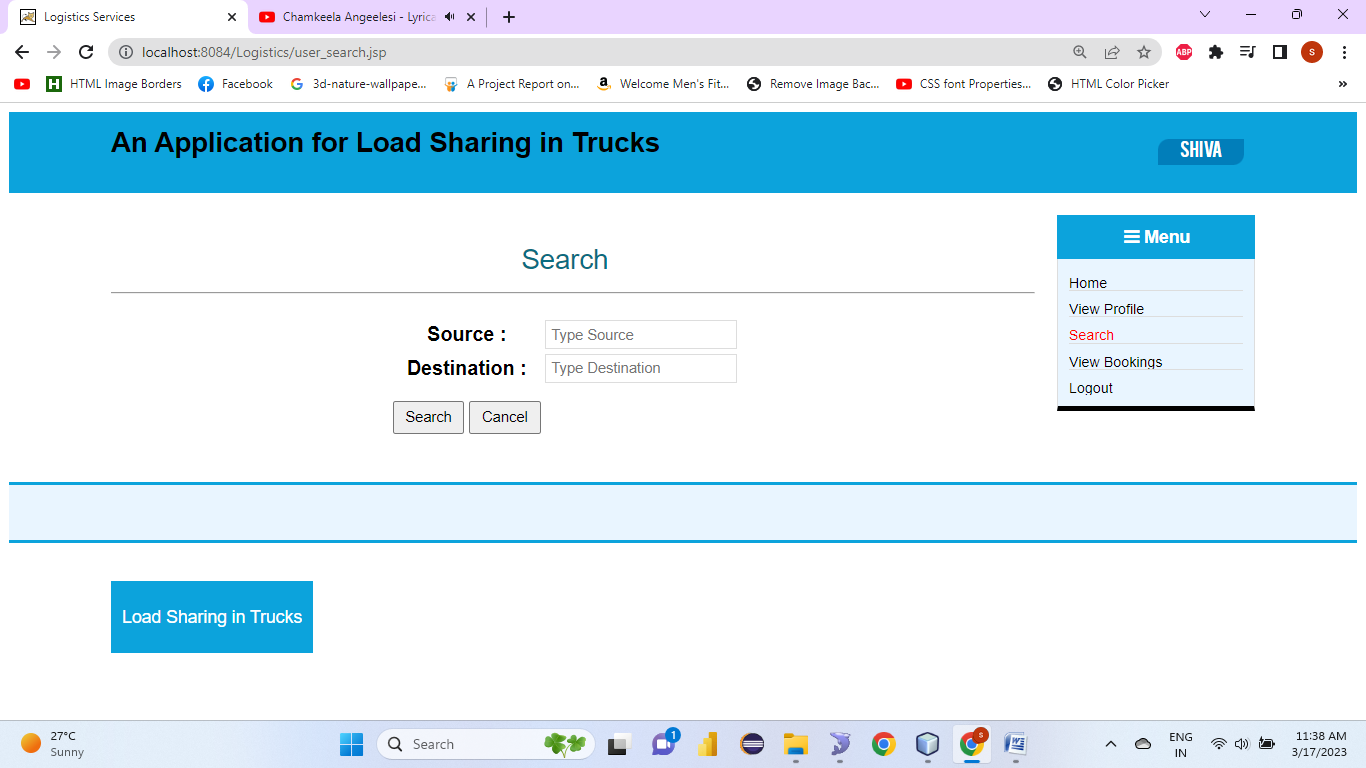
User home



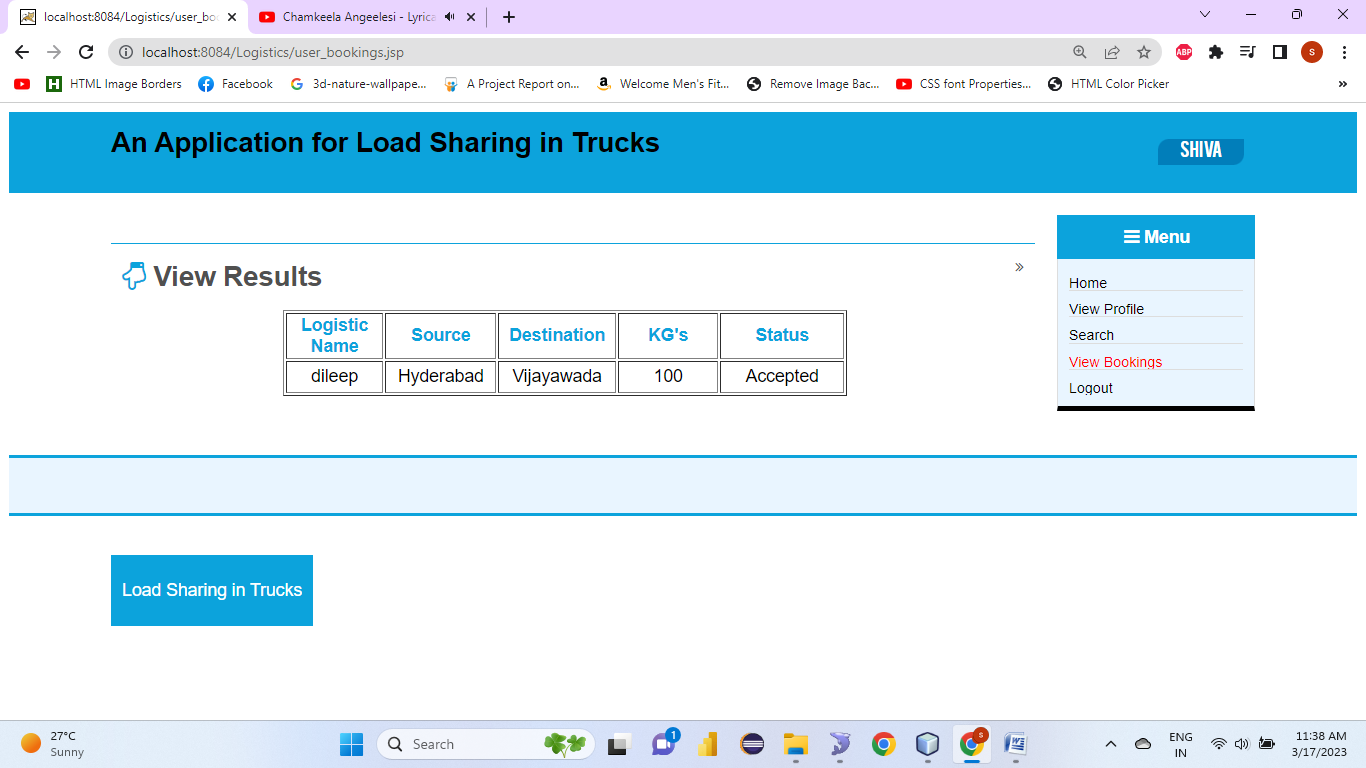
View profile:



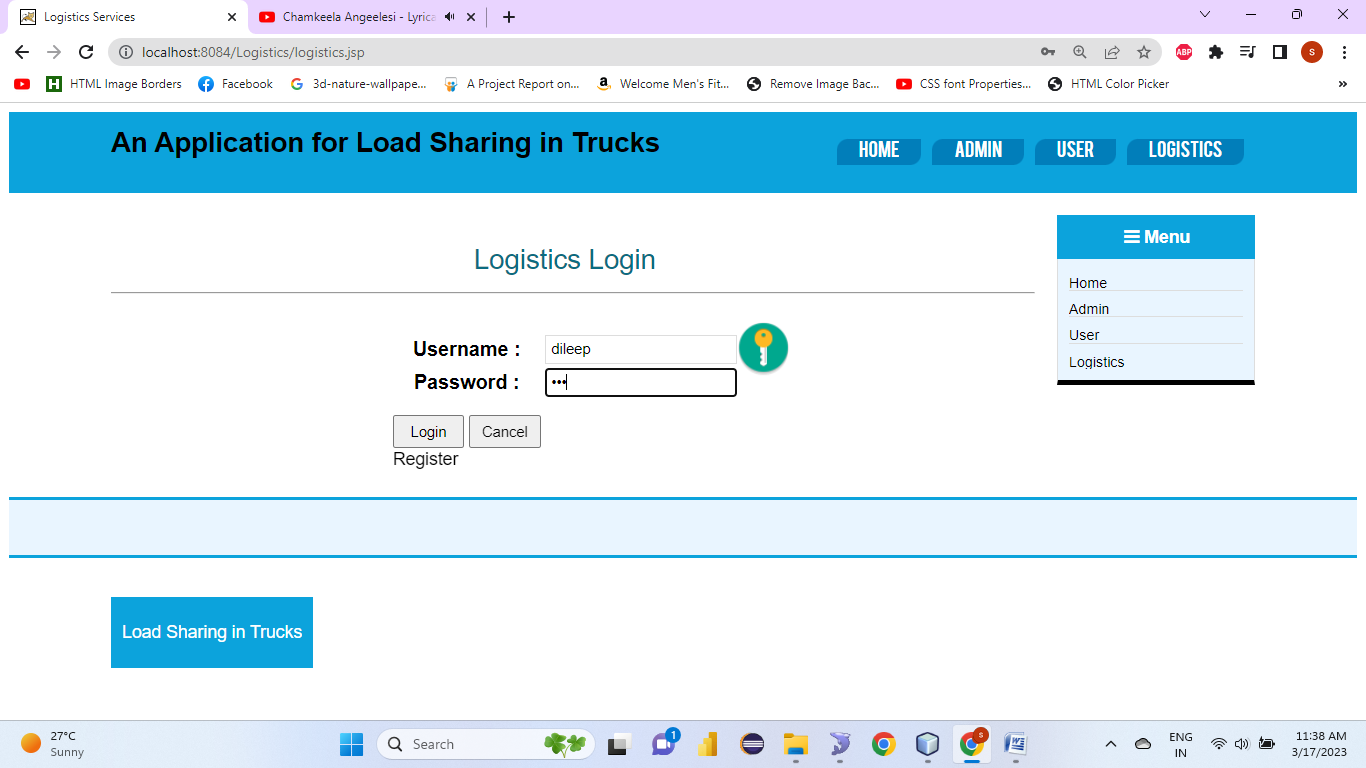
Search

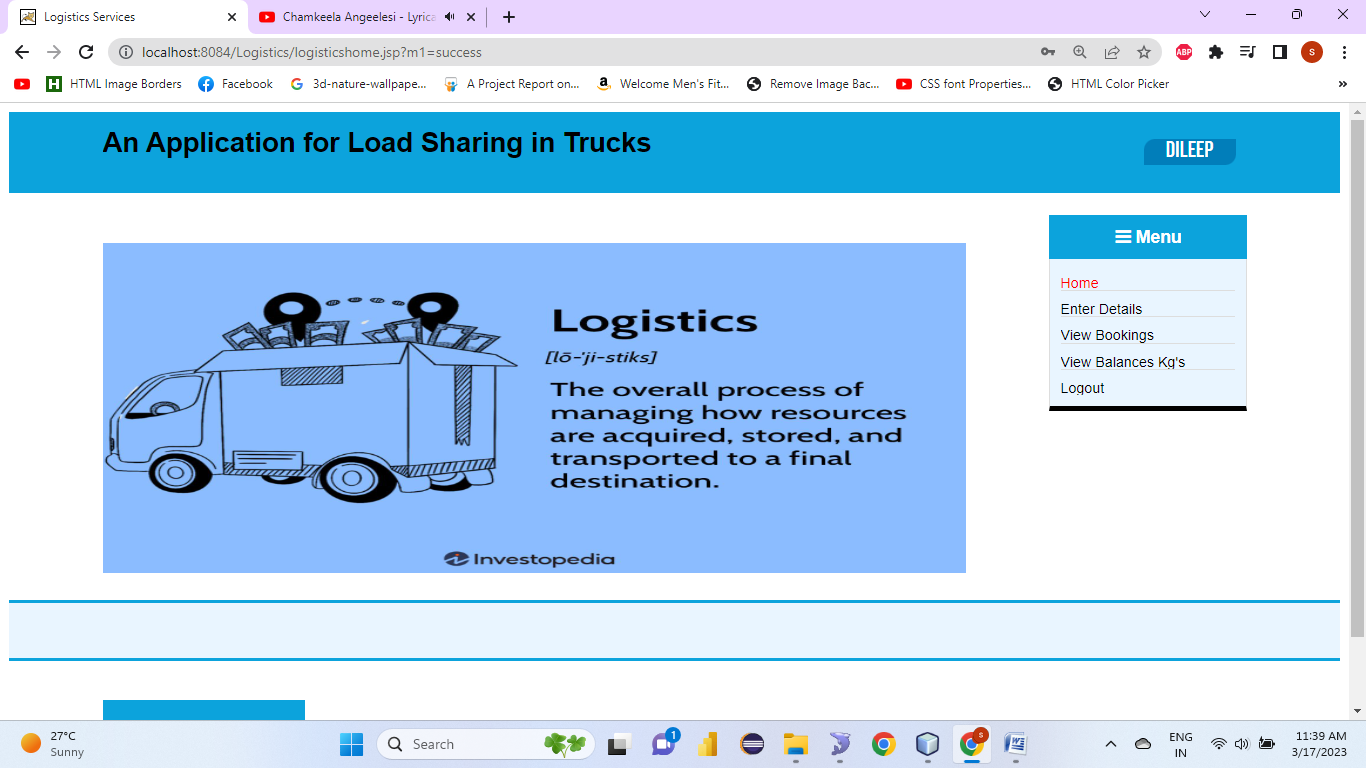


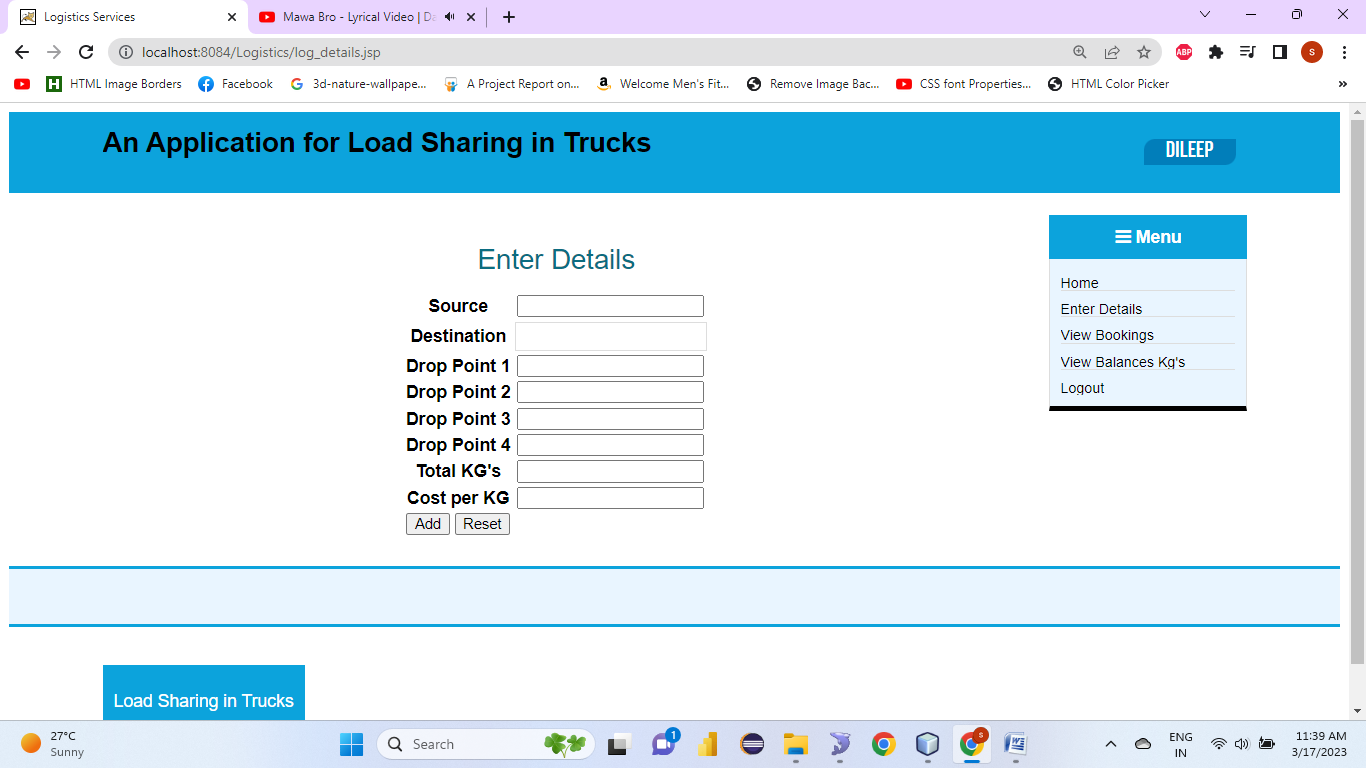
View bookings:

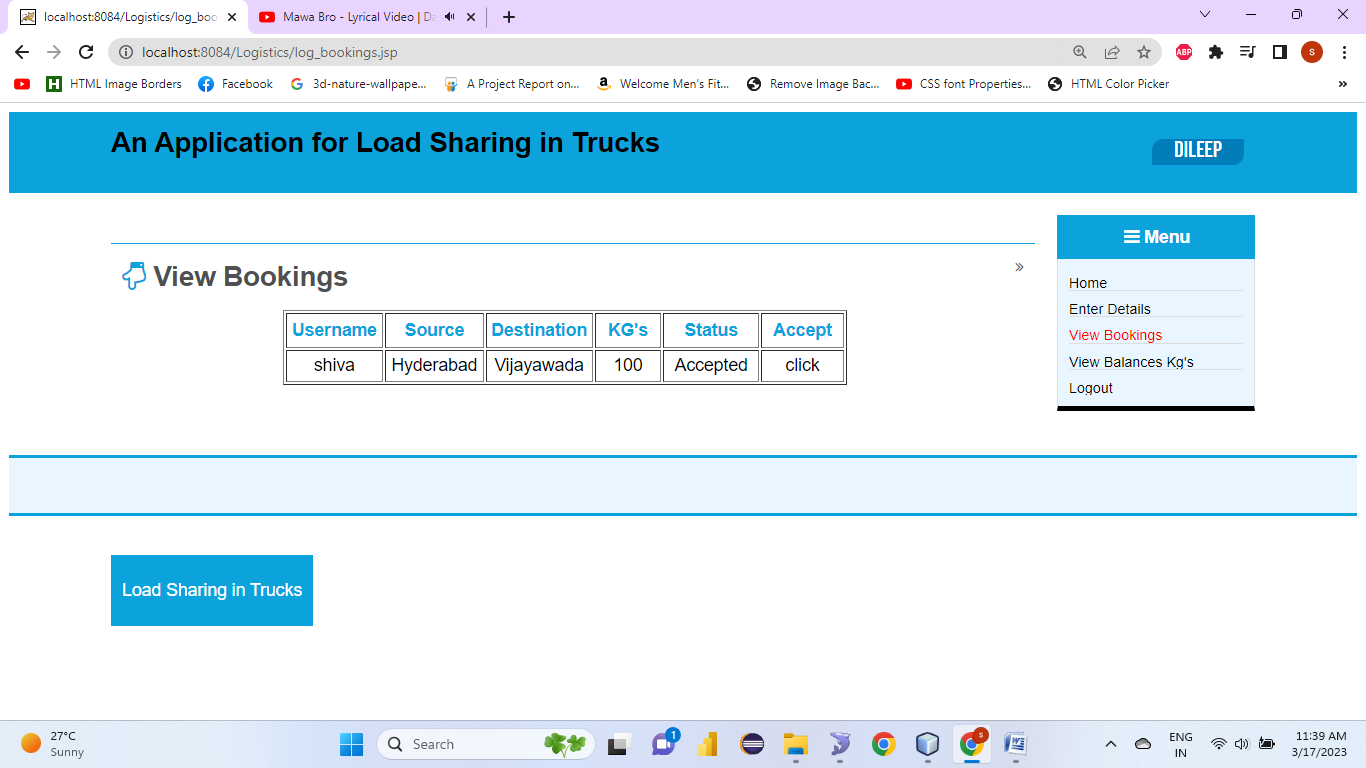


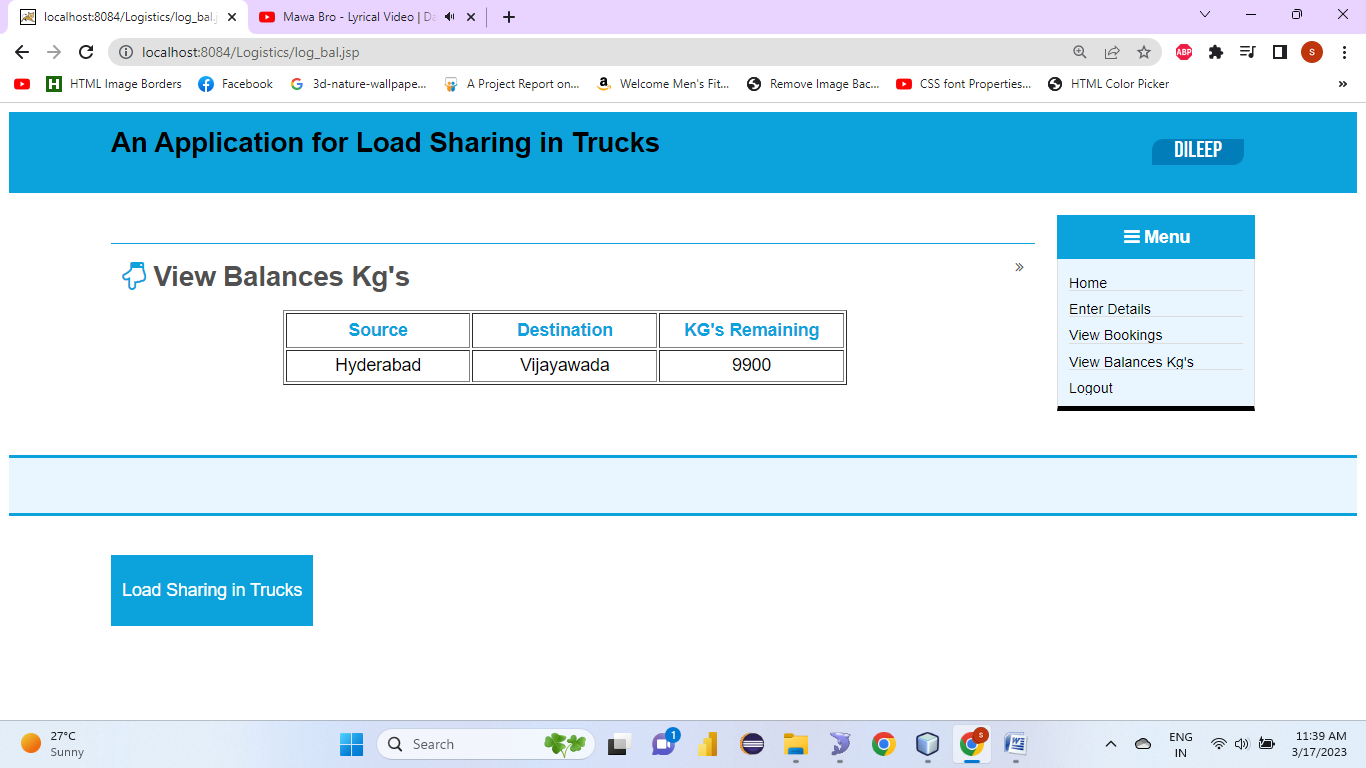
Log











Only Images:

Only Videos:

View Pdf files:

**1**

# CONCLUSION AND FUTURE ENHANCEMENT

## CONCLUSION

The proposed system regularly monitors load present in the truck and then notify users the condition of the present load as well as whether new load can be added or not. The proposed method overcomes the limitation of the existing systems by tracking the load continuously. This feature provides new dynamics in the freight service management. The sharing of load also helps users save their money and time..

.

## FUTURE ENHANCEMENT

 Facility for modifying user details.

 Access of data through mobile devices.  More interactive user interface.

Facilities for backup creation.

# BIBLIOGRAPHY

Good Teachers are worth more than thousand books, we have them in Our Department.

### REFERENCE PAPERS:

Mohammad Torkjazi, Nathan Huynh, Samaneh Shiri, “Truck appointment systems considering impact to drayage truck tours”, Elsevier, Transportation Research Part E 116 (2018) 208–228.

2. Xinjie Xing, Paul R. Drake, Dongping Song, Yang Zhou, “Tank Container Operators’ profit maximization through dynamic operations planning integrated with the quotation-booking process under multiple uncertainties”, Elsevier, European Journal of Operational Research 274 (2019) 924–946.

3. Chenhao Zhou, Haobin Li, Byung Kwon Lee, Zhipeng Qiu, “A simulation-based vessel-truck coordination strategy for lighterage terminals”, Elsevier, Transportation Research Part C 95 (2018) 149–164.

4. Dong Li, Zhan Pang, “Dynamic Booking Control for Car Rental Revenue Management: A Decomposition Approach”, Elsevier, European Journal of Operational Research (2016).

5. Gooi Sai Weng, SuhaizaZailani, Mohammad Iranmanesh, Sunghyup Sean Hyun, “Mobile taxi booking application service’s continuance usage intention by users”, Elsevier, Transportation Research Part D: Transport and Environment Volume 57, December 2017, Pages 207-216.

6. Teodor Gabriel Crainic, GuidoPerboli, Mariangela Rosano, “Simulation of intermodal freight transportation systems: a taxonomy”, Elsevier, European Journal of Operational Research Volume 270, Issue 2, 16 October 2018, Pages 401-418.

7. S.Vidhya Sagar, B.Balakiruthiga, A.Sivanesh Kumar, “Novel Vehicle Booking System Using lOT”, IEEE, 2016 Online International Conference on Green Engineering and Technologies (IC-GET).

8. Kangjia Zhao, Wenqing Chen, Kong Wei Lye, “Online Vehicle Dispatch: from Assignment to Scheduling”, IEEE, IEEE International Conference on Data Mining Workshops (ICDMW) (2018).

9. ZHIGANG JIN, RUOQIAN WU, XINGYU CHEN, and GEN LI, “Charging Guiding Strategy for Electric Taxis Based on Consortium Blockchain”, IEEE Access, volume 7, 02 October 2019.

10. Chen, G., Jiang, L., 2016 “Managing customer arrivals with time windows: a case of truck arrivals at a congested container terminal”. Ann. Oper. Res. 244 (2), 349–365er