ONLINE TRUCK MANAGEMENT

1.INTRODUCTION

Online Truck Management is developed to manage all Truck hiring work online. It useful for truck booking agency that are specialized in Hiring trucks to customers. Using this system it is very easy for customer to book a truck online. So it is also very useful for common people to book a truck. It is an online system through which customers can view available trucks; register the trucks, view profile and book trucks. Mostly peoples use truck service for their goods transportations need. The objective and scope of my project Online truck booking System is to record the details various activities of user. It will simplify the task and reduce the paper work. Using this truck booking management system truck owner can also become partner of truck booking agency by giving their truck for booking. Online truck management is a web based application that allows users to book a truck online. From this system truck booking company can manage all truck bookings and customer information. User can book trucks and admin can confirm the booking and cancel the booking on the basis of availability of the trucks and drivers.

1.2.ADVANTAGES

1. Easy booking process
2. Find and book the truck near by the user..
3. Time efficient.
4. Choose the truck what we need.
5. Book the truck in advance.

1.3.DISADVANTAGES

1. Need to book the particular warehouse truck only.
2. Manually went to the warehouse and book the truck in many places.
3. Takes more time.
4. Many chances for trucks not available for immediate goods delivery.

1.4 MODULES

1. Owner registration
2. User registration
3. Truck information
4. Goods deliver information
5. Booking information

1.4.1. OWNER REGISTRATION

This module has contain the informations about truck owners such as their names, mail ID, address, and mobile numbers. The detail of this module helps to verify the truck owner’s information.

1.4.2. USER REGISTRATION

This module deals with the authentication reports of user or admin. In user authentication module, it stored the verification details of the user. The type of user and their usage, identification details like user id, name, address etc. In admin authentication the verification of the user occurs. The detail of this module helps to verify the user information.

1.4.3. TRUCK INFORMATION

This module has contain the informations about truck such as their driver names, mail ID, address, and mobile numbers. The detail of this module helps to verify the truck information.

1.4.4. GOODS DELIVER INFORMATION

This module contains the information about the goods, and the weight of goods. Also it has contains pickup and delivery location, price for the transportation and the information about the truck driver or owner.

1.4.5. BOOKING INFORMATION

This module is about the information that contains about the booking such as booking date, time, place, and quantity

1.5. SOFTWARE SPECIFICATION

OPERATING SYSTEM	:	Windows 10
FRONT END	:	PHP
BACKEND	:	MY SQL

1.6. HARDWARE REQUIREMENTS

HARD DISK	:	500 GB
RAM	:	2GB
PROCESSOR SPEED	:	3.00GHz
PROCESSOR	:	Pentium IV Processor
MONITOR	:	LG 15.6 inch

2. SYSTEM STUDY

2.1. EXISTING SYSTEM

Transport since long ago has evolved for sure. We have several ways to get trucks for our transport but are the affordable, efficient and easy. As it is today, trucks do most of the transporting of goods from one point to another within the country. But we can assign the truck for the goods by manually, like we went to the truck warehouse or call the lorry owner and said to the lorry owner for pickup the goods and delivered in the particular place. If in this particular warehouse, all the trucks were went for a rental means, we don't have any alternative for this scenario.

2.2. PROPOSED SYSTEM

The Indian Road Transport market is unorganized and the operations are being done mostly through phone calls. But, with the help of proposed system they will be able to post, view and manage all at the same time. In spite of empty vehicles available in the particular market place the cost for the shipment sores high, due to lack of common online market place. The proposed system is to create transparency and visibility in Transport Sector, allowing smooth, efficient and cost effective deliveries of goods. It will be possible to create customized solutions for trucks. The proposed system will serve Manufacturers, Brokers and anyone who wants to ship something. The system will help enable the creation of a common market thus permitting free and unobstructed movement of goods and services across the country. Introduction of the proposed system will also reduce the number of warehouses required to be maintained in different states, thereby resulting in a substantial increase in demand for integrated logistics solutions. The proposed system is to create for the people to book the truck easily by the application.

2.3. FEATURES

MySQL is offered under two different editions: the open source MySQL Community Server and the proprietary Enterprise Server. MySQL Enterprise Server is differentiated by a series of proprietary extensions which install as server plugins, but otherwise shares the version numbering system and is built from the same code base.

Major features as available in MySQL 5.6:

- A broad subset of ANSI SQL 99, as well as extensions
- Cross-platform support
- Stored procedures, using a procedural language that closely adheres to SQL/PSM
- Triggers
- Cursors
- Updatable views
- Online DDL when using the InnoDB Storage Engine.
- Information schema
- Performance Schema
- A set of SQL Mode options to control runtime behavior, including a strict mode to better adhere to SQL standards.
- X/Open XA distributed transaction processing (DTP) support; two phase commit as part of this, using the default InnoDB storage engine
- Transactions with savepoints when using the default InnoDB Storage Engine. The NDB Cluster Storage Engine also supports transactions.
- ACID compliance when using InnoDB and NDB Cluster Storage Engines
- SSL support
- Query caching
- Sub-SELECTs (i.e. nested SELECTs)
- Built-in Replication support (i.e. Master-Master Replication & Master-Slave Replication) with one master per slave, many slaves per master. Multi-master replication is provided in MySQL Cluster, and multi-master support can be added to unclustered configurations using Galera Cluster.
- Full-text indexing and searching
- Embedded database library
- Unicode support

2.4. MILESTONES

Notable milestones in MySQL development include:

- Original development of MySQL by Michael Widenius and David Axmark beginning in 1994
- First internal release on 23 May 1995
- Version 3.19: End of 1996, from www.tcx.se
- Version 3.20: January 1997
- Windows version was released on 8 January 1998 for Windows 95 and NT
- Version 3.21: production release 1998, from www.mysql.com
- Version 3.22: alpha, beta from 1998
- Version 3.23: beta from June 2000, production release 22 January 2001
- Version 4.0: beta from August 2002, production release March 2003 (unions)
- Version 4.01: beta from August 2003, Jyoti adopts MySQL for database tracking
- Version 4.1: beta from June 2004, production release October 2004 (R-trees and B-trees, subqueries, prepared statements)
- Version 5.0: beta from March 2005, production release October 2005 (cursors, stored procedures, triggers, views, XA transactions)

  The developer of the Federated Storage Engine states that "The Federated Storage Engine is a proof-of-concept storage engine", but the main distributions of MySQL version 5.0 included it and turned it on by default. Documentation of some of the short-comings appears in "MySQL Federated Tables: The Missing Manual". Sun Microsystems acquired MySQL AB in 2008.

- Version 5.1: production release 27 November 2008 (event scheduler, partitioning, plugin API, row-based replication, server log tables)

  Version 5.1 contained 20 known crashing and wrong result bugs in addition to the 35 present in version 5.0 *(almost all fixed as of release 5.1.51).* MySQL 5.1 and 6.0-alpha showed poor performance when used for data warehousing – partly due to its inability to utilize multiple CPU cores for processing a single query.

- Oracle acquired Sun Microsystems on 27 January 2010.

- The day Oracle announced the purchase of Sun, Michael "Monty" Widenius forked MySQL, launching MariaDB, and took a swathe of MySQL developers with him.
- MySQL Server 5.5 was generally available (as of December 2010). Enhancements and features include:
    - The default storage engine is InnoDB, which supports transactions and referential integrity constraints.
    - Improved InnoDB I/O subsystem
    - Improved SMP support
    - Semisynchronous replication.
    - SIGNAL and RESIGNAL statement in compliance with the SQL standard.
    - Support for supplementary Unicode character sets utf16, utf32, and utf8mb4.
    - New options for user-defined partitioning.
- MySQL Server 6.0.11-alpha was announced on 22 May 2009 as the last release of the 6.0 line. Future MySQL Server development uses a New Release Model. Features developed for 6.0 are being incorporated into future releases.
- MySQL 5.6 general availability was announced in February 2013. New features included performance improvements to the query optimizer, higher transactional throughput in InnoDB, new NoSQL-style memcached APIs, improvements to partitioning for querying and managing very large tables, TIMESTAMP column type that correctly stores milliseconds, improvements to replication, and better performance monitoring by expanding the data available through the PERFORMANCE_SCHEMA. The InnoDB storage engine also included support for full text search and improved group commit performance.
- MySQL 5.7 Development Milestone 3 was released December 2013.

3. SYSTEM DESIGN AND DEVELOPMENT

3.1.INPUT DESIGN

Input design is the process of converting user-originated inputs to a computer-based format. Input design is one of the most expensive phases of the operation of computerized system and is often the major problem of a system. Input design is a part of overall design, which requires careful attribute. Inaccurate input data are the most common cause of errors in data processing. The goal of designing input data is to make data entry as easy, logical and free from errors. In the system design phase input data are collected and organized in to groups of similar data.

Input Forms Are,

1.User Registration

2.Owner registration

3.2.OUTPUT DESIGN

Output design generally refers to the results and information that are generated by the system for many end-users; output is the main reason for developing the system and the basis on which they evaluate the usefulness of the application. Computer output is the most important and direct source of information to the user. Output design is very important phase because the output will be in an interactive manner. The report of data transaction is seen as a output in this project

Output Forms are,

1.Booking

2.Status

3.3.DATABASE DESIGN

The database design is a must for any application developed especially more for the data store projects. Since the android application system method involves retrieving the information in the table and produced to the users, proper handling of the table is a must. In the project, login table is designed to be unique in accepting the username and the length of the username and password should be greater than zero. The different users view the data in different format according to the privileges given.

3.4 FILE DESIGN

A file system provides the machinery to support the project tasks. At the highest level a file system is a way to organize, store, retrieve, and manage information on a permanent storage medium such as a disk. File systems manage permanent storage and form an integral part of all operating systems. There are many different approaches to the task of managing permanent storage. At one end of the spectrum are simple file systems that impose enough restrictions to inconvenience users and make using the file system difficult. In deciding what type of filing system is appropriate for a particular operating system, we must weigh the needs of the problem with the other constraints of the project. The two basic abstractions of files and directories form the basis of what a file system can operate on. There are many operations that a file system can perform on files and directories. All file systems must provide some basic level of support. Beyond the most basic file system primitives lay other features, extensions, and more sophisticated operations.

The Structure of a File is given the concept of a file, a file system may impose no structure on the file, or it may enforce a considerable amount of structure on the contents of the file. An unstructured, “raw” file, often referred to as a “stream of bytes,” literally has no structure. The file system simply records the size of the file and allows programs to read the bytes in any order or fashion that they desire. If a file system chooses to enforce a formal structure on files, it usually does so in the form of records.

3.5. PHP PROGRAMMING

PHP is a scripting language designed to fill the gap between SSI (Server Side Includes) and Perl, intended for the Web environment. Its principal application is the implementation of Web pages having dynamic content. PHP has gained quite a following in recent times, and it is one of the frontrunners in the Open Source software movement. Its popularity derives from its C-like syntax, and its simplicity. The newest version of PHP is 5.5 and it is heavily recommended to always use the newest version for better security, performance and of course features.

If you've been to a website that prompts you to login, you've probably encountered a server-side scripting language. Due to its market saturation, this means you've probably come across PHP. PHP was designed by Rasmus Lerdorf to display his resume online and to collect data from his visitors.

Basically, PHP allows a static webpage to become dynamic. "PHP" is an acronym that stands for "**P**HP: **H**ypertext **P**reprocessor". The word "Preprocessor" means that PHP makes changes before the HTML page is created. This enables developers to create powerful applications that can publish a blog, remotely control hardware, or run a powerful website such as Wikipedia or Wikibooks. Of course, to accomplish something such as this, you need a database application such as MySQL..

PHP development began in 1994 when Rasmus Lerdorf wrote a series of Common Gateway Interface (CGI) binaries in C, which he used to maintain his personal homepage. He extended them to add the ability to work with web forms and to communicate with databases, and called this implementation "Personal Home Page/Forms Interpreter" or PHP/FI.

3.5.1. PHP 6

PHP received mixed reviews due to lacking native Unicode support at the core language level. In 2005, a project headed by Andrei Zmievski was initiated to bring native Unicode support throughout PHP, by embedding the International Components for Unicode (ICU) library, and representing text strings as UTF-16 internally. Since this would cause major changes both to the internals of the language and to user code, it was planned to release this as version 6.0 of the language, along with other major features then in development.

3.5.2. PHP 7

As of 2014, work is underway on a new major PHP version named PHP 7. There was some dispute as to whether the next major version of PHP was to be called PHP 6 or PHP 7. While the PHP 6 unicode experiment had never been released, a number of articles and book titles referenced the old PHP 6 name, which might have caused confusion if a new release were to reuse the PHP 6 name. After a vote, the name PHP 7 was chosen.

In terms of new language features, PHP 7 will add features such as return type declarations, which will complement its existing parameter type declarations. PHP 7 will also contain an improved variable syntax which is internally consistent and complete, resolving a long-standing issue in PHP, what will allow use of ->, [], (), {}, and :: operators with arbitrary meaningful left-hand-side expressions.

3.5.3. SYNTAX

The following Hello world program is written in PHP code embedded in an HTML document:

```
<!DOCTYPE html>
<html>
   <head>
      <title>PHP Test</title>
   </head>
   <body>
      <?php echo '<p>Hello World</p>'; ?>
   </body>
</html>
```

However, as PHP does not need to be embedded in HTML or used with a web server, the simplest version of a Hello World program can be written like this, with the closing tag omitted as preferred in files containing pure PHP code (prior to PHP 5.4.0, this short syntax for echo() only works with the short_open_tag configuration setting enabled, while for PHP 5.4.0 and later it is always available):

```
<?= 'Hello world';
```

4. SYSTEM TESTING AND IMPLEMENTATION

4.1. INTODUCTION

The most important phase in system development life cycle is system testing. The number and nature of errors in a newly designed system depends on the system specifications and the time frame given for the design.

A newly designed system should have all the subsystems working together, but in reality each subsystems work independently. During this phase, all the subsystems are gathered into one pool and tested to determine whether it meets the user requirements.

Testing is done at two level -Testing of individual modules and testing the entire system. During the system testing, the system is used experimentally to ensure that the software will run according to the specifications and in the way the user expects. Each test case is designed with the intent of finding errors in the way the system will process it.

Testing plays a very critical role in determining the reliability and efficiency of software and hence is a very important stage in software development. Software testing is done at different levels. They are the unit testing and system testing which comprises of integration testing and acceptance testing.

4.2. TYPES OF TESTING

- **UNIT TESTING**
- **INTERGRATION TESTING**
- **VALIDATION TESTING**
- **SYSTEM TESTING**

4.2.1. UNIT TESTING

This is the first level of testing. The different modules are tested against the specifications produced during the integration. This is done to test the internal logic of each module. Those resulting from the interaction between modules are initially avoided. The input received and output generated is also tested to see whether it falls in the expected range of values. Unit testing is performed from the bottom up, starting with the smallest and lowest modules and proceeding one at a time.

The units in a system are the modules and routines that are assembled and integrated to perform a specific function. The programs are tested for correctness of logic applied and detection of errors in coding. Each of the modules was tested and errors are rectified. They were then found to function properly.

4.2.2. INTEGRATION TESTING

In integration testing, the tested modules are combined into sub-systems, which are then tested. The goal of integration testing to check whether the modules can be integrated properly emphasizing on the interfaces between modules. The different modules were linked together and integration testing done on them.

4.2.3. VALIDATION TESTING

The objective of the validation test is to tell the user about the validity and reliability of the system. It verifies whether the system operates as specified and the integrity of important data is maintained. User motivation is very important for the successful performance of the system.

All the modules were tested individually using both test data and live data. After each module was ascertained that it was working correctly and it had been "integrated" with the system. Again the system was tested as a whole. We hold the system tested with different types of users. The System Design, Data Flow Diagrams, procedures etc. were well documented so that the system can be easily maintained and upgraded by any computer professional at a later.

4.2.4. SYSTEM TESTING

The integration of each module in the system is checked during this level of testing. The objective of system testing is to check if the software meets its requirements. System testing is done to uncover errors that were not found in earlier tests. This includes forced system failures and validation of total system as the user in the operational environment implements it. Under this testing, low volumes of transactions are generally based on live data. This volume is increased until the maximum level for each transactions type is reached. The total system is also tested for recovery after various major failures to ensure that no data are lost during the breakdown.

4.3. DATA TYPES

PHP stores whole numbers in a platform-dependent range, either a 64-bit or 32-bit signed integer equivalent to the C-language long type. Unsigned integers are converted to signed values in certain situations; this behavior is different from other programming languages. Integer variables can be assigned using decimal (positive and negative), octal, hexadecimal, and binary notations.

Floating point numbers are also stored in a platform-specific range. They can be specified using floating point notation, or two forms of scientific notation. PHP has a native Boolean type that is similar to the native Boolean types in Java and C++. Using the Boolean type conversion rules, non-zero values are interpreted as true and zero as false, as in Perl and C++.

4.4. FUNCTION

PHP has hundreds of functions provided by the core language functionality and thousands more available via various extensions; these functions are well documented in the online PHP documentation. However, the built-in library has a wide variety of naming conventions and associated inconsistencies, as described under history above.

4.5. IMPLEMNTATIONS

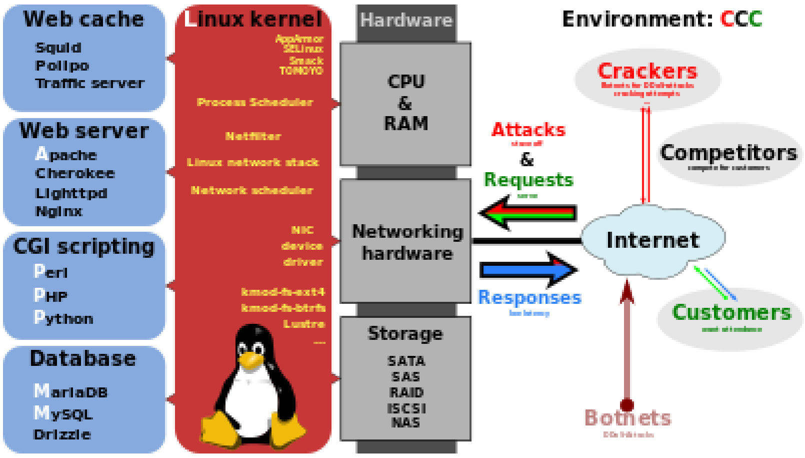
The original, only complete and most widely used PHP implementation is powered by the Zend Engine and known simply as PHP. To disambiguate it from other implementations, it is sometimes unofficially referred to as "Zend PHP". The Zend Engine compiles PHP source code on-the-fly into an internal format that it can execute, thus it works as an interpreter. It is also the "reference implementation" of PHP, as PHP has no formal specification, and so the semantics of Zend PHP define the semantics of PHP itself. Due to the complex and nuanced semantics of PHP, defined by how Zend works, it is difficult for competing implementations to offer complete compatibility.

4.6. INSTALLATION AND CONFIGURATION

There are two primary ways for adding support for PHP to a web server – as a native web server module, or as a CGI executable. PHP has a direct module interface called Server Application Programming Interface (SAPI), which is supported by many web servers including Apache HTTP Server, Microsoft IIS, Netscape (now defunct) and iPlanet. Some other web servers, such as OmniHTTPd, support the Internet Server Application Programming Interface (ISAPI), which is a Microsoft's web server module interface. If PHP has no module support for a web server, it can always be used as a Common Gateway Interface (CGI) or FastCGI processor; in that case, the web server is configured to use PHP's CGI executable to process all requests to PHP files.

When PHP is installed and used in cloud environments, software development kits (SDKs) are provided for using cloud-specific features. For example:

- Amazon Web Services provides the AWS SDK for PHP
- Windows Azure can be used with the Windows Azure SDK for PHP.



4.7. MYSQL

MySQL is (as of July 2013) the world's second most widely used relational database management system (RDBMS) and most widely used open-source RDBMS. It is named after co-founder Michael Widenius's daughter, My. The SQL acronym stands for Structured Query Language.

The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements. MySQL was owned and sponsored by a single for-profit firm, the Swedish company MySQL AB, now owned by Oracle Corporation.

MySQL is a popular choice of database for use in web applications, and is a central component of the widely used LAMP open source web application software stack (and other 'AMP' stacks). LAMP is an acronym for "Linux, Apache, MySQL, Perl/PHP/Python." Free-software-open source projects that require a full-featured database management system often use MySQL.

For proprietary use, several paid editions are available, and offer additional functionality. Applications which use MySQL databases include: TYPO3, MODx, Joomla, WordPress, phpBB, MyBB, Drupal and other software.

4.7.1. INTERFACE

MySQL is a relational database management system (RDBMS), and ships with no GUI tools to administer MySQL databases or manage data contained within the databases. Users may use the included command line tools, or use MySQL "front-ends", desktop software and web applications that create and manage MySQL databases, build database structures, back up data, inspect status, and work with data records. The official set of MySQL front-end tools, MySQL Workbench is actively developed by Oracle, and is freely available for use.

4.7.2. GRAPHICAL

The official MySQL Workbench is a free integrated environment developed by MySQL AB, that enables users to graphically administer MySQL databases and visually design database structures. MySQL Workbench replaces the previous package of software, MySQL GUI Tools. Similar to other third-party packages, but still considered the authoritative MySQL front end, MySQL Workbench lets users manage database design & modeling, SQL development (replacing MySQL Query Browser) and Database administration (replacing MySQL Administrator).

MySQL Workbench is available in two editions, the regular free and open source Community Edition which may be downloaded from the MySQL website, and the proprietary Standard Edition which extends and improves the feature set of the Community Edition.

4.7.3. COMMAND LINE

MySQL ships with many command line tools, from which the main interface is 'mysql' client. Third parties have also developed tools to manage MySQL servers.

- MySQL Utilities – a set of utilities designed to perform common maintenance and administrative tasks. Originally included as part of the MySQL Workbench, the utilities are now a stand-alone download available from Oracle.

- Percona Toolkit – a cross-platform toolkit for MySQL, developed in Perl.[31] Percona Toolkit can be used to prove replication is working correctly, fix corrupted data, automate repetitive tasks, and speed up servers. Percona Toolkit is included with several Linux distributions such as CentOS and Debian, and packages are available for Fedora and Ubuntu as well. Percona Toolkit was originally developed as Maatkit, but as of late 2011, Maatkit is no longer developed.

4.7.4. PROGRAMMING

MySQL works on many system platforms, including AIX, BSDi, FreeBSD, HP-UX, eComStation, i5/OS, IRIX, Linux, OS X, Microsoft Windows, NetBSD, Novell NetWare, OpenBSD, OpenSolaris, OS/2 Warp, QNX, Oracle Solaris, Symbian, SunOS, SCO OpenServer, SCO UnixWare, Sanos and Tru64. A port of MySQL to OpenVMS also exists.

MySQL is written in C and C++. Its SQL parser is written in yacc, but it uses a home-brewed lexical analyzer. Many programming languages with language-specific APIs include libraries for accessing MySQL databases. These include MySQL Connector/Net for integration with Microsoft's Visual Studio (languages such as C# and VB are most commonly used) and the JDBC driver for Java. In addition, an ODBC interface called MyODBC allows additional programming languages that support the ODBC interface to communicate with a MySQL database, such as ASP or ColdFusion. The HTSQL – URL-based query method also ships with a MySQL adapter, allowing direct interaction between a MySQL database and any web client via structured URLs.

4.8.DEPLOYMENT

MySQL can be built and installed manually from source code, but this can be tedious so it is more commonly installed from a binary package unless special customizations are required. On most Linux distributions the package management system can download and install MySQL with minimal effort, though further configuration is often required to adjust security and optimization settings.Though MySQL began as a low-end alternative to more powerful proprietary databases, it has gradually evolved to support higher-scale needs as well. It is still most commonly used in small to medium scale single-server deployments, either as a component in a LAMP-based web application or as a standalone database server. Much of MySQL's appeal originates in its relative simplicity and ease of use, which is enabled by an ecosystem of open source tools such as phpMyAdmin. In the medium range, MySQL can be scaled by deploying it on more powerful hardware, such as a multi-processor server with gigabytes of memory.

4.9. BACKUP

- Filesystem snapshot or volume manager snapshot – backups are performed by using an external tool provided by the operating system (such as LVM) or storage device, with additional support from MySQL for ensuring consistency of such snapshots.
- *mysqldump* – a logical backup tool included with both community and enterprise editions of MySQL. Supports backup from all storage engines.
- *MySQL Enterprise Backup* – a hot backup utility included as part of the MySQL Enterprise subscription from Oracle. Offers native InnoDB hot backup, as well as backup for other storage engines.
- *XtraBackup* – open source MySQL hot backup software. Some notable features include hot, non-locking backups for InnoDB storage, incremental backups, streaming, parallel-compressed backups, throttling based on the number of I/O operations per second, etc.

4.10. HIGH AVAILABILITY

Ensuring high availability requires a certain amount of redundancy in the system. For database systems, the redundancy traditionally takes the form of having a primary server acting as a master, and using replication to keep secondaries available to take over in case the primary fails. This means that the "server" that the application connects to is in reality a collection of servers, not a single server. In a similar manner, if the application is using a sharded database, it is in reality working with a collection of servers, not a single server. .

5. CONCLUSION

It is concluded that the application works well and satisfy the users. The application is tested very well and errors are properly debugged. The site is simultaneously accessed from more than one system. Simultaneous login from more than one place is tested.

The site works according to the restrictions provided in their respective browsers. Further enhancements can be made to the application, so that the web site functions very attractive and useful manner than the present one. The speed of the transactions become more enough now.

SCOPE FOR FUTURE DEVELOPMENT

Every application has its own merits and demerits. The project has covered almost all the requirements. Further requirements and improvements can easily be done since the coding is mainly structured or modular in nature. Changing the existing modules or adding new modules can append improvements. Further enhancements can be made to the application, so that the web site functions very attractive and useful manner than the present one.

BIBLIOGRAPHY

Textual Reference

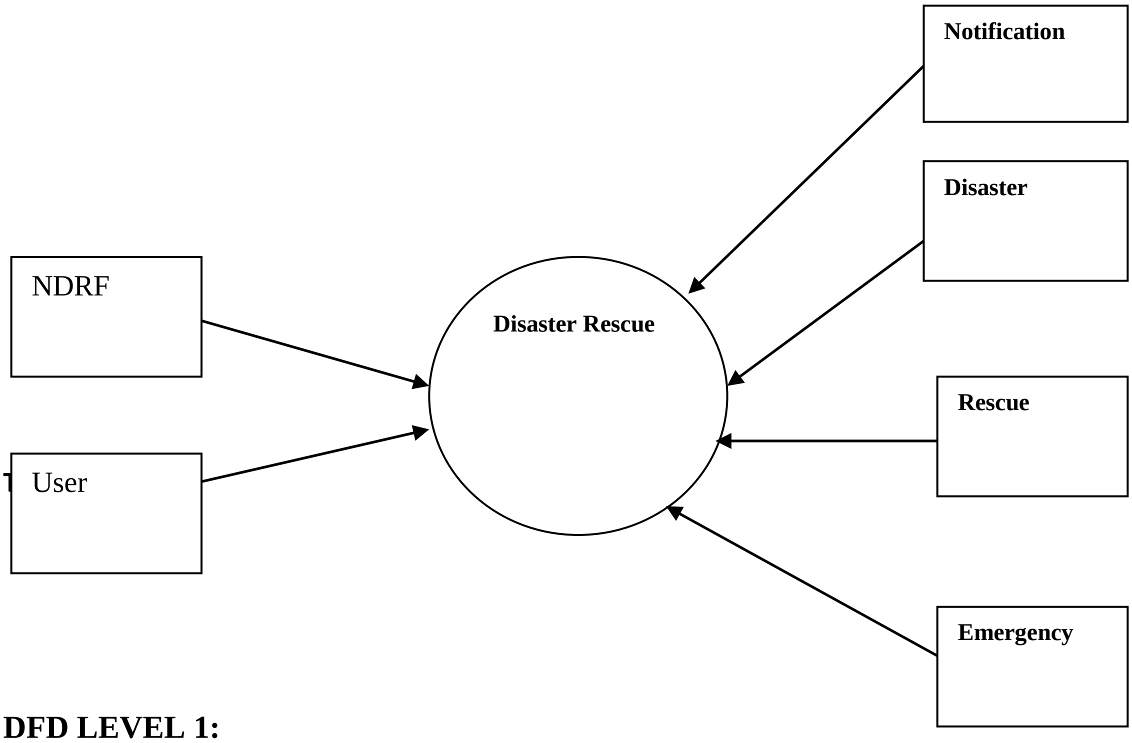
- PHP book by Vasvani (TMH publications).
- Beginning PHP5 by WROX.
- Informatics practices by Sumita Arora.
- Head First PHP & MySQL by Lynn Beighley and Michael Morrison(O’Reilly)

Online Reference:

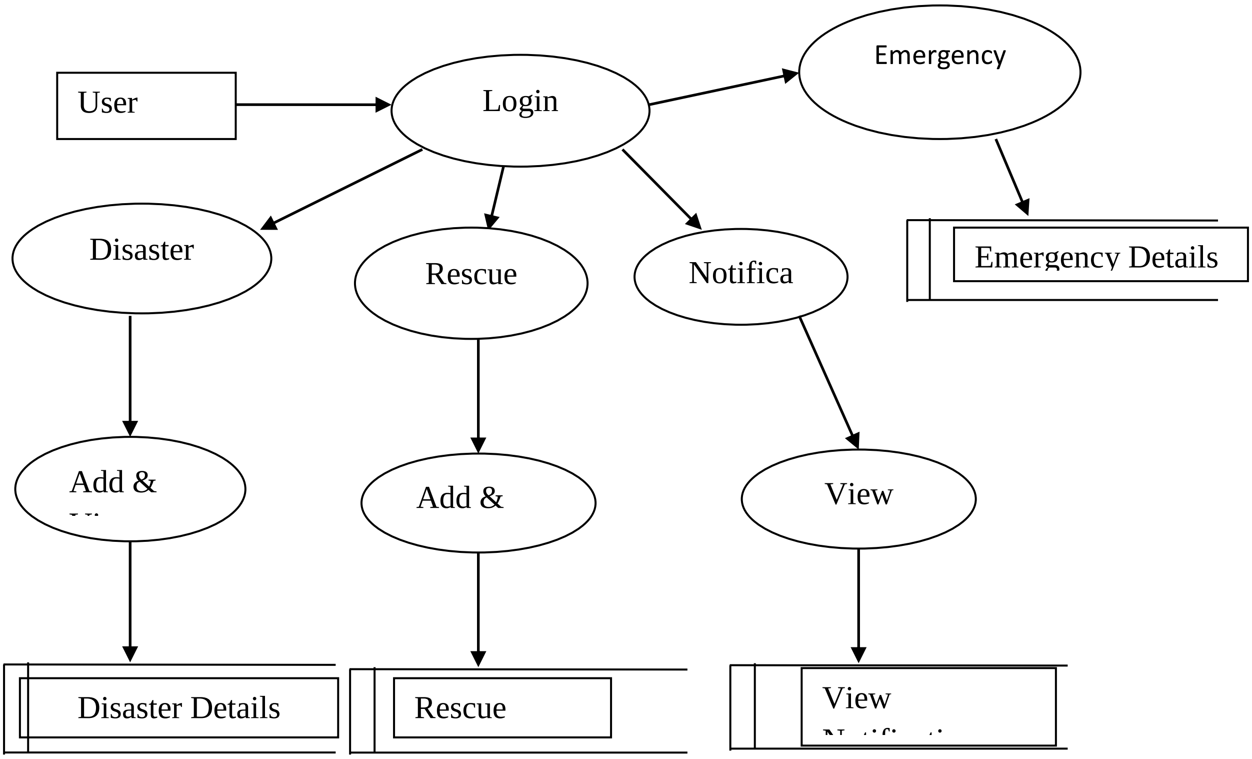
- www.wikipedia.com
- www.w3schools.com
- http://www.phpreferencebook.com/

DATA FLOW DIAGRAM

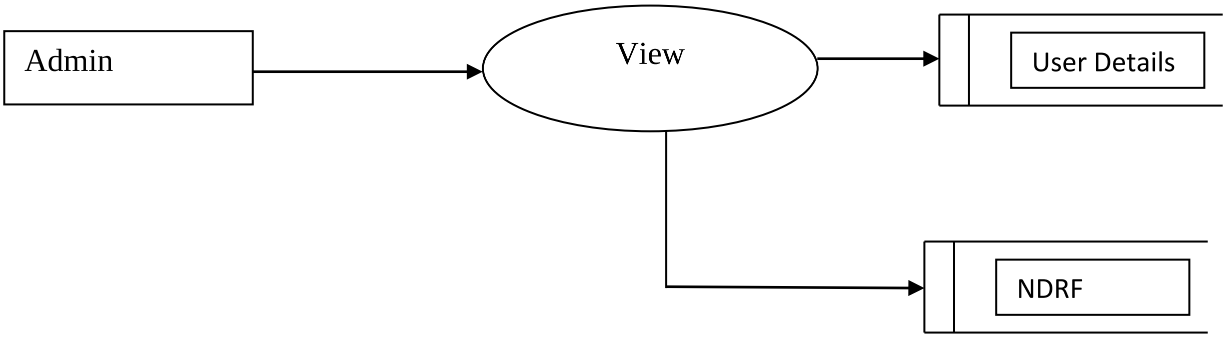
DFD LEVEL 0:



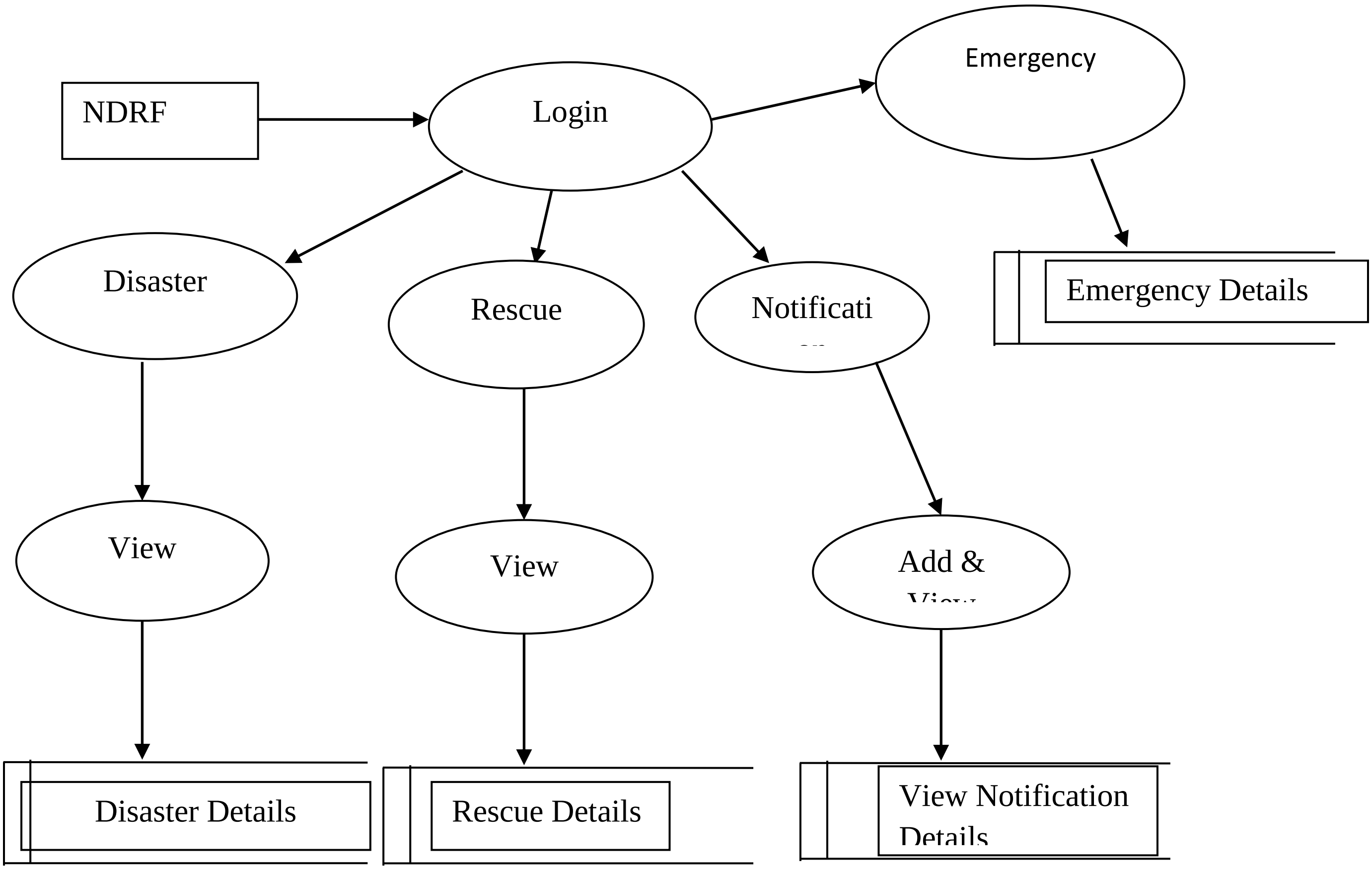
DFD LEVEL 1:



DFD LEVEL 2:

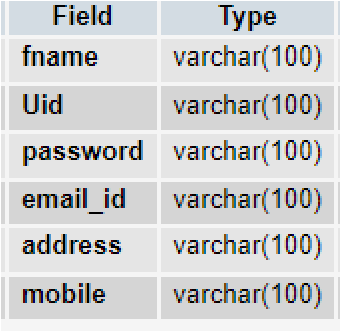


DFD LEVEL 3:



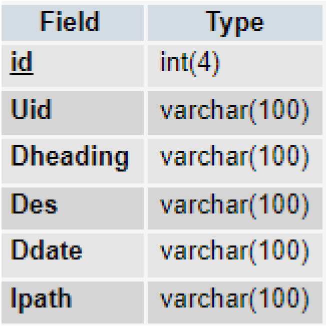
TABLES

USER TABLE



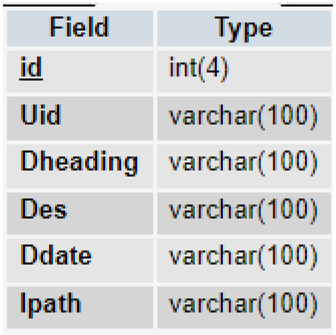
Field	Type
fname	varchar(100)
Uid	varchar(100)
password	varchar(100)
email_id	varchar(100)
address	varchar(100)
mobile	varchar(100)

NDRF TABLE



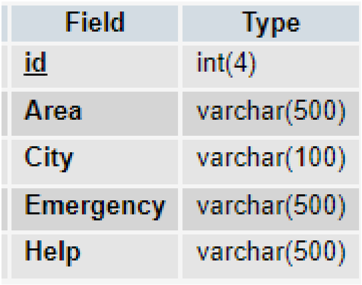
Field	Type
id	int(4)
Uid	varchar(100)
Dheading	varchar(100)
Des	varchar(100)
Ddate	varchar(100)
Ipath	varchar(100)

DISASTER



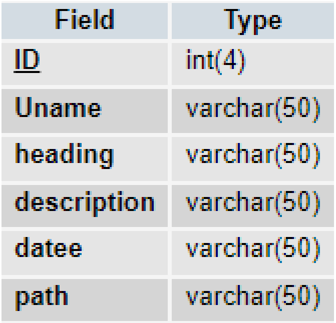
Field	Type
id	int(4)
Uid	varchar(100)
Dheading	varchar(100)
Des	varchar(100)
Ddate	varchar(100)
Ipath	varchar(100)

EMERGENCY



Field	Type
id	int(4)
Area	varchar(500)
City	varchar(100)
Emergency	varchar(500)
Help	varchar(500)

NOTIFICATION



Field	Type
ID	int(4)
Uname	varchar(50)
heading	varchar(50)
description	varchar(50)
datee	varchar(50)
path	varchar(50)

SOURCE CODE

```
<?php
        include_once("config.php");
        include_once("Userheader.php");
?>
<?php
error_reporting(0);

$query2 = "select mobile from tab_user where U_name='".$_SESSION['loggedIn']."'";
                        //echo $query2;
                        $result1 = mysql_query($query2);
                        if(mysql_num_rows($result1))
                                {
                        $row1 = mysql_fetch_assoc($result1);

    }

?>
<style type="text/css">
<!--
.ed{
border-style:solid;
border-width:thin;
border-color:#00CCFF;
padding:5px;
margin-bottom: 4px;
}
```

```
#button1{
text-align:center;
font-family:Arial, Helvetica, sans-serif;
border-style:solid;
border-width:thin;
border-color:#00CCFF;
padding:5px;
background-color:#00CCFF;
height: 34px;
}
#imagelist{
border: thin solid silver;
float:left;
padding:5px;
width:auto;
margin: 0 5px 0 0;
}
p{
margin:0;
padding:0;
text-align: center;
font-style: italic;
font-size: smaller;
text-indent: 0;
}
#caption{
margin-top: 5px;
}
```

```
img{

height: 225px;

}

-->

</style>


<form action="addexec.php" method="post" enctype="multipart/form-data" name="addroom">

<br>

<center>


<br>

<center><font color="red" size="14">Add Disaster Details</font></center>

<br>

<br>

User Id<br />

 <input name="uname" type="text" class="ed" id="unameid" value="<?php  echo
$_SESSION['login_user']; ?>"/>

 <br />


Disaster Heading<br />

 <input name="DH" type="text" class="ed" id="DH" />

 <br />

  Description<br />

 <input name="DES" type="text" class="ed" id="DES" />

 <br />

  Date<br />

 <input name="DAT" type="text" class="ed" id="DAT" />

<br>

 Select Image: <br />
 <input type="file" name="image" class="ed"><br />


  <input type="submit" name="Submit" value="Upload" id="button1" />


  &nbsp;
  <a href="Viewdisaster.php">My Update</a>
   &nbsp;
 <a href="Viewalldisaster.php">View All Disaster</a>
 </center>
 </form>
<br />
Photo Archieve
<br />
<?php
//include('config.php');
$result = mysql_query("SELECT * FROM adddisaster where Uid='".$_SESSION['login_user']."'");


while($row = mysql_fetch_array($result))
{
echo '<div id="imagelist">';
echo '<p><img src="'.$row['Ipath'].'"></p>';
echo '<p id="caption">'.$row['Des'].' </p>';
echo '</div>';
}
?>
<br>
<br>
<br>
```

```
<br>

<br>

<?php

include('Adminheader.php');

include('config.php');


error_reporting(0);


 if(isset($_POST['submit']))

              {


                          if($_POST['area']=="" || $_POST['city']==""|| $_POST['emer']==""||
$_POST['help']=="")

              {

                         echo '<script> alert("PLEASE FILL THE DETAILS");</script>';

              }

              else

              {

             $query = "INSERT INTO `addemergency` (`id`,`Area`, `City`, `Emergency`,`Help`)
VALUES ('null','".$_POST['area']."','".$_POST['city']."','".$_POST['emer']."','".$_POST['help']."')";


             //echo $query;exit;


             if(mysql_query($query)){

                        echo 'ADD SUCCESSFULLY';


             }

             else{

                        echo 'NOT REGISTERED';
```

```
            }
            header("location:AddEmergency.php");
            exit;
 }

 }

?>

<style>
h1 {
  color: blue;
  font-family: verdana;
  font-size: 300%;

}
p  {
  color: gray;
  font-family: Georgia, serif;
  font-size: 140%;
   font-weight: bold;
}
</style>
<style>
input[type=button], input[type=submit], input[type=reset] {
  background-color: #FFC912;
  border: none;
  color: white;
  padding: 15px 42px;

  text-decoration: none;

  margin: 4px 2px;

  cursor: pointer;

  text-align: center;

  display: inline-block;

border-radius: 16px;

}

</style>


<style>

.textbox {

            border: 1px solid #848484;

            -webkit-border-radius: 30px;

            -moz-border-radius: 30px;

            border-radius: 30px;

            outline:0;

            height:25px;

            width: 275px;

            padding-left:10px;

            padding-right:10px;

}

</style>


<div id="container">

<div background="">

<br>
```

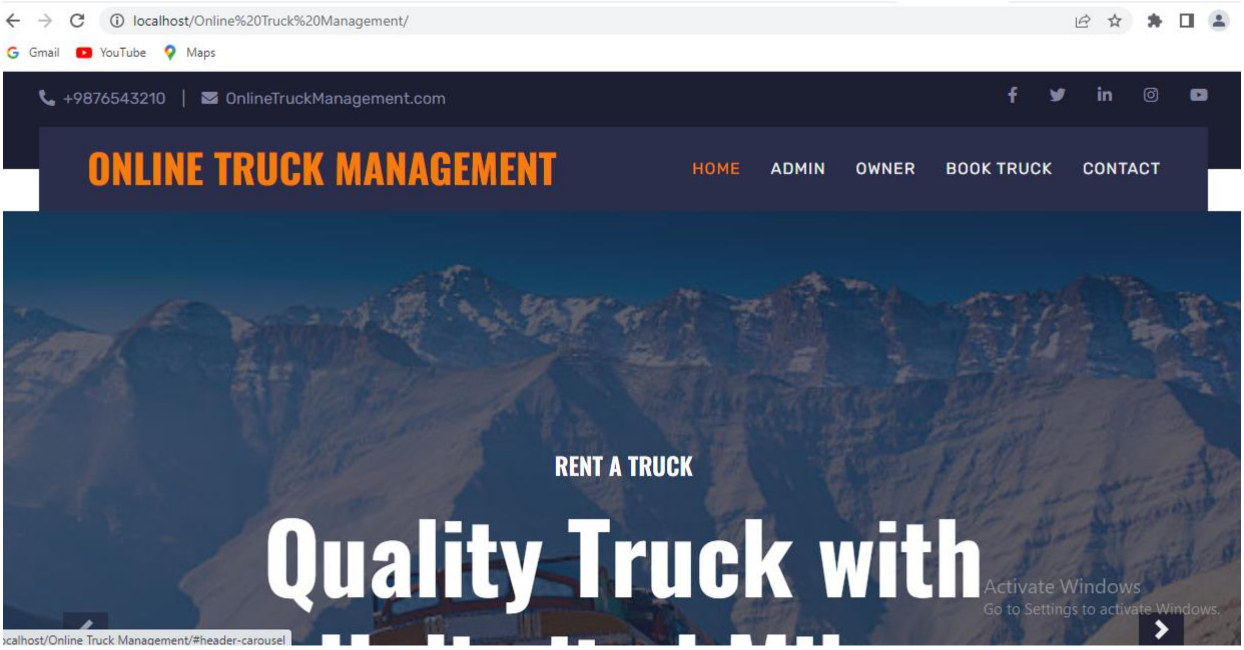
```
<div id="error_msg"></div>

<form action="" method="post" >

<center>

<table border="0" cellpadding="5" cellspacing="0" width="350" Height="120" align="center"
class="details_single">

<tr>

  <td colspan="2"><h1><b>Add Emergency</b></h1> </td>

</tr>

</table>

<table border="0" cellpadding="5" cellspacing="0" width="310" Height="100" align="center"
class="details_single">

<tr>

        <td ><p>Area</p> </td>

        <td ><input type="text" name="area" id="area"  class="textbox"/>

        </td>

</tr>

<tr>

        <td ><p>City </p></td>

        <td ><input type="text" name="city" id="city" class="textbox"/>

        </td>

</tr>



<tr>

        <td ><p>Emergency </p></td>

        <td ><input type="text" name="emer" id="emer" class="textbox"/>

        </td>

</tr>

<tr>

        <td ><p>Help </p></td>
```

```
            <td ><input type="text" name="help" id="help" class="textbox"/>

            </td>

</tr>

<tr>

            <td colspan="2" align="center">

            <input type="submit" name="submit" value="Add" />


            </td>

</tr>

</table>


<h1>Emergency Details</h1>


            <table border="2" cellspacing="6" class="displaycontent" width="1200" height="120"
style="border:10px solid #800000;" align="center">

            <tr>


                        <th bgcolor=Black><font color=white size=2>id</font></th>

                        <th bgcolor=Black><font color=white size=2>Area</font></th>

                        <th bgcolor=Black><font color=white size=2>City</font></th>

                        <th bgcolor=Black><font color=white size=2>Emergency</font></th>

                        <th bgcolor=Black><font color=white size=2>Help</font></th>

                         <th bgcolor=Black><font color=white size=2>Delete</font></td>


            </tr>


            <?php


            $query = "select * from addemergency";
            $result = mysql_query($query) or die(mysql_error());

            while($row = mysql_fetch_assoc($result))

                        {

 ?>

            <tr>




                        <td bgcolor=white><font color=#000000 size=2><?php echo $row['id']; 
?></font></td>

                        <td bgcolor=white><font color=#000000 size=2><?php echo $row['Area']; 
?></font></td>

                        <td bgcolor=white><font color=#000000 size=2><?php echo $row['City']; 
?></font></td>

                        <td bgcolor=white><font color=#000000 size=2><?php echo $row['Emergency']; 
?></font></td>

                        <td bgcolor=white><font color=#000000 size=2><?php echo $row['Help']; 
?></font></td>

                        <td bgcolor=white><font color=#000000 size=2><a href="?delete=<?php echo 
$row['id'];?>">Delete</a></font></td>


            </tr>

            <?php }  ?>


            </table>




</center>

</form>

</div>

<br>

<br>
```

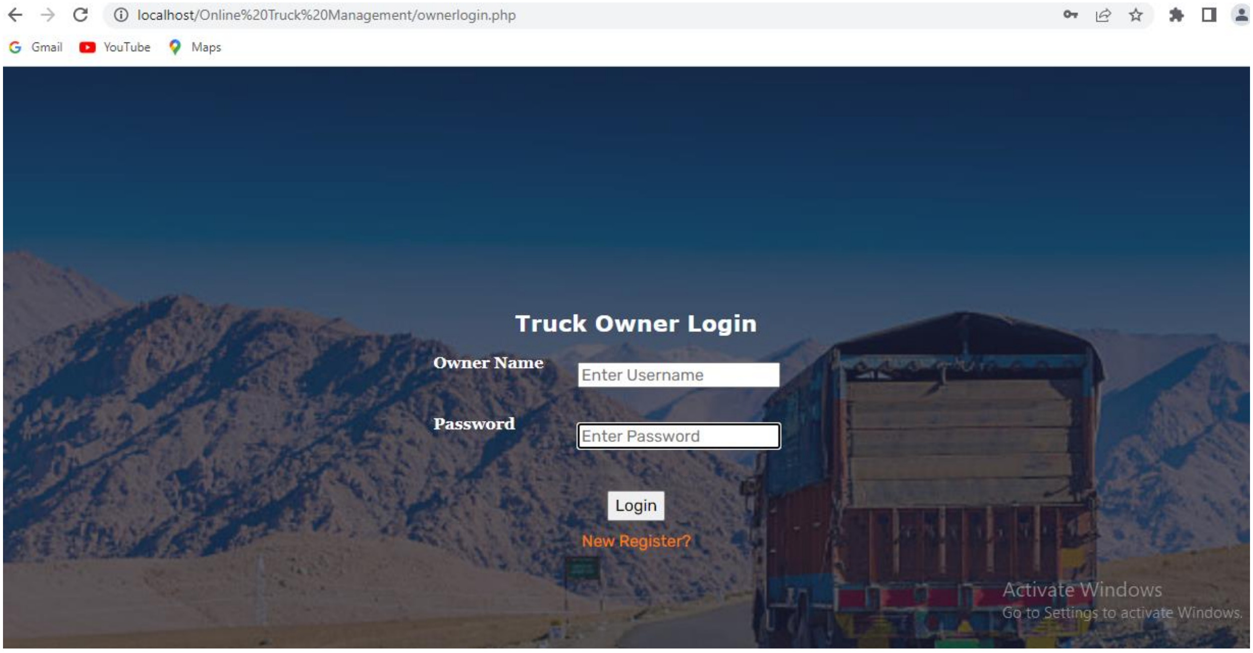
```
<?php

if(isset($_GET['delete']))
        {

        $query = "delete from addemergency where id='".$_GET['delete']."'";
        mysql_query($query);
                echo '<script>alert("Deleted");</script>';


header("location:AddEmergency.php");
        exit;
        }

?>
```

SCREEN SHOTS

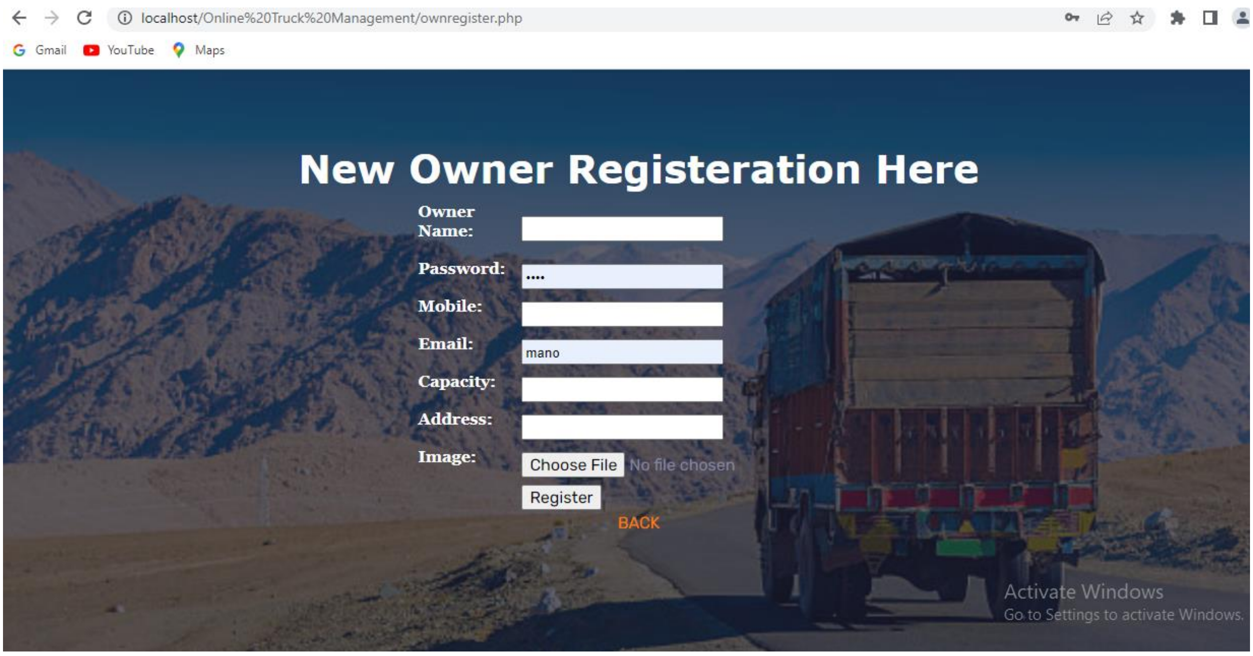
HOME PAGE


localhost/Online%20Truck%20Management/
Gmail
YouTube
Maps
+9876543210
OnlineTruckManagement.com
ONLINE TRUCK MANAGEMENT
HOME
ADMIN
OWNER
BOOK TRUCK
CONTACT
RENT A TRUCK
Quality Truck with
Activate Windows
Go to Settings to activate Windows.
localhost/Online Truck Management/#header-carousel

OWNER LOGIN



NEW OWNER REGISTERATION

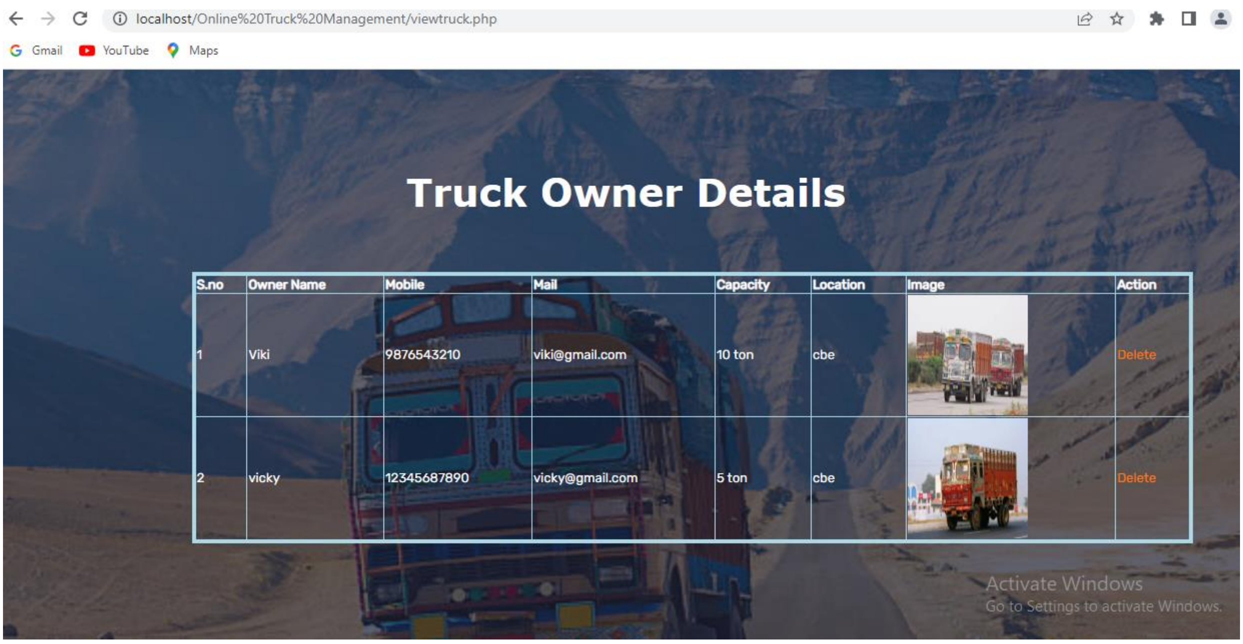

localhost/Online%20Truck%20Management/ownregister.php
Gmail
YouTube
Maps
Activate Windows
Go to Settings to activate Windows.

New Owner Registeration Here

Owner Name:	
Password:	....
Mobile:	
Email:	mano
Capacity:	
Address:	
Image:	Choose File No file chosen
	Register

BACK

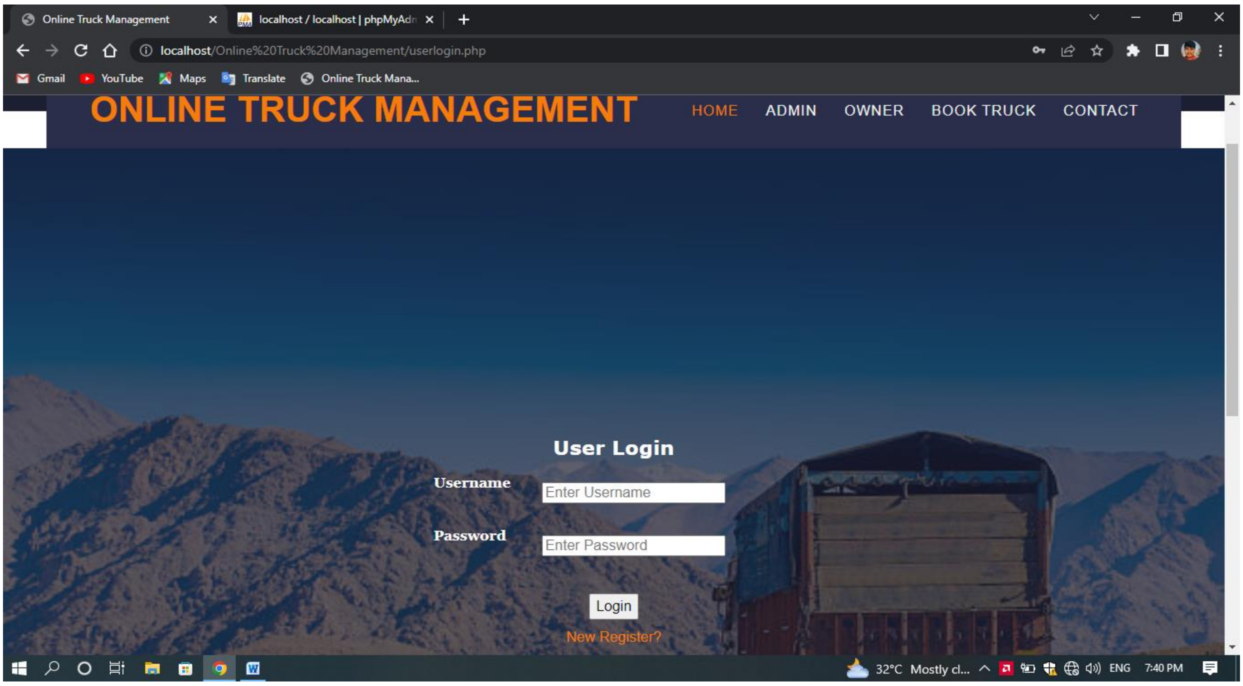
OWNER DETAILS


localhost/Online%20Truck%20Management/viewtruck.php
Gmail
YouTube
Maps
Activate Windows
Go to Settings to activate Windows.

Truck Owner Details

S.no	Owner Name	Mobile	Mail	Capacity	Location	Image	Action
1	Viki	9876543210	viki@gmail.com	10 ton	cbe		Delete
2	vicky	12345687890	vicky@gmail.com	5 ton	cbe		Delete

USER LOGIN



User Login

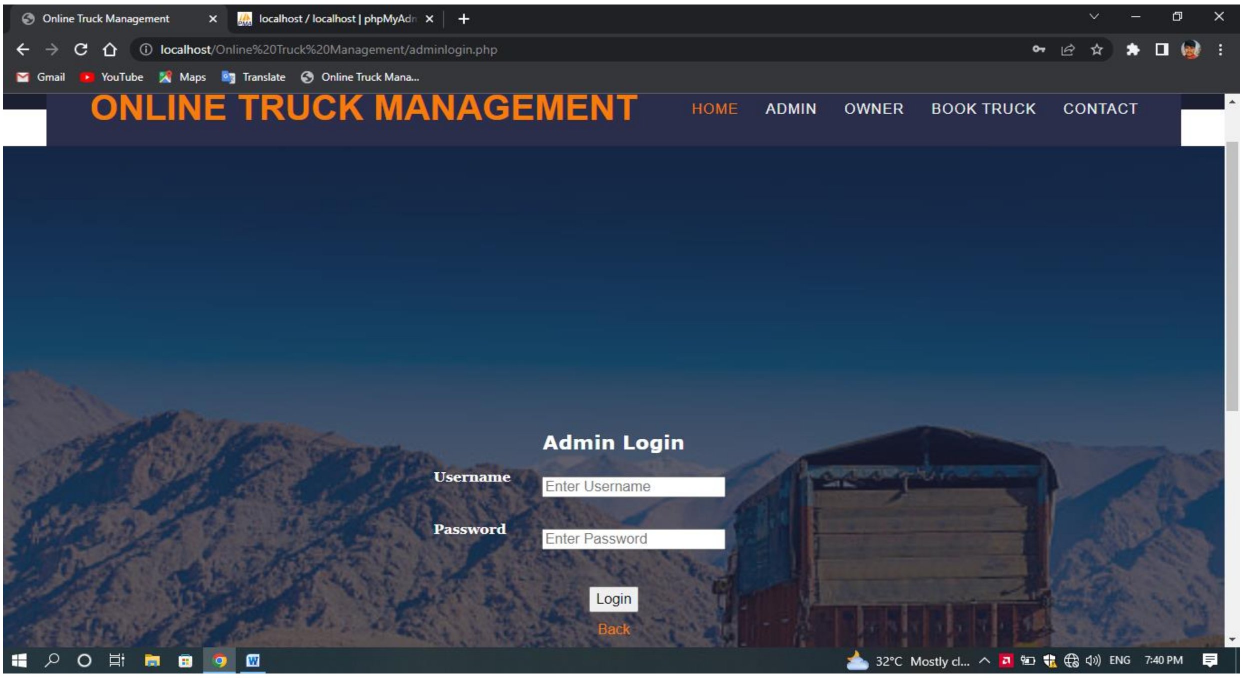
Username Enter Username

Password Enter Password

Login

New Register?

ADMIN LOGIN



Admin Login

Username Enter Username

Password Enter Password

Login

Back