**A Project Report**

**On**

**“Tourist Guide System”**

****

**M.Girish Chowdary**

**B.Tech(CSE)**

**211801370024**

**Faculty Incharge HOD**

**ACKNOWLEDGEMENT**

We take this occasion to thank God, almighty for blessing us with his grace and

taking our endeavour to a successful culmination. We extend our sincere and heart felt

thanks to our esteemed guide, Mr.Mittu for providing us with the right guidance and advice at the crucial junctures and for showing us the right way. We extend our sincere thanks to our respected head of the division Mr.Mittu, for allowing us to use the facilities available. We would like to thank the other faculty members also, at this occasion. Last but not the least, we would like to thank friends for the support and encouragement they have given us during the course of our work.

**ABSTRACT**

As the name specifies “TOURISM MANAGEMENT SYSTEM” is a software developed

for managing tour booking.

Identification of the drawbacks of the existing system leads to the designing of computerized

system that will be compatible to the existing system with the system Which is more user friendly

and more GUI oriented. We can improve the efficiency of the system, thus overcome the drawbacks

of the existing system.

·Less human error

·Strength and strain of manual labour can be reduced

·High security

·Data redundancy can be avoided to some extent

·Data consistency

·Easy to handle

·Easy data updating

·Easy record keeping

·Backup data can be easily generated

**INDEX**

 Objective

 Existing System

 Proposed System

 Study of the System

 Functional Requirement

 Number of Modules

 System Requirement

 Hardware Configuration

 Software Configuration

 Software Features

 PHP

 MYSQL

 PHPMYADMIN’

 Input Design

 Output Design

 DFD Diagrams

 UML Diagrams

 ER Diagrams

 Database Design

 Database Table and Structure of DB Tables

 Architecture Flow

 Conclusion

 Bibliography

**OBJECTIVE:**



The objective of the project is to develop a system that automates the processes and activities

of a travel and tourism agency.



The purpose is to design a system using which one can perform all operations related to

traveling and sight-seeing.

**EXISTING SYSTEM:**



In the present system a customer has to approach various agencies to find details of places

and to book tickets.



This often requires a lot of time and effort.

 A customer may not get the desired information from these offices and often the customer

may be misguided.

 It is tedious for a customer to plan a particular journey and have it executed properly

**PROPOSED SYSTEM:**



Here we will provide a tourist guide based on your tour package .



We will only provide all the required facilities to the customer .



The customer can also contact us directly for more details about the tour packages.

**STUDY OF THE SYSTEM:**

To provide flexibility to the users, the interfaces have been developed that are accessible through a

browser. The GUI’S at the top level have been categorized as

1.

Administrative user interface

2.

The operational or generic user interfaceThe ‘administrative user interface’ concentrates on the consistent information that is practically, part

of the organizational activities and which needs proper authentication for the data collection. These

interfaces help the administrators with all the transactional states like Data insertion, Data deletion

and Date updation along with the extensive data search capabilities.

The ‘operational or generic user interface’ helps the end users of the system in transactions through

the existing data and required services. The operational user interface also helps the ordinary users in

managing their own information in a customized manner as per the included flexibilities.

**FUNCTIONAL REQUIREMENTS:Number of Modules**

After careful analysis the system has been identified to have the following modules:

**1.**

**Administrator module**

**2.**

**User(Traveler) module**

**3.**

**Guest user**

**1. ADMINISTRATOR MODULE:**

This module provides administrator related functionality. Administrator manages all information and

has access rights to add, delete, edit and view the data related to places, travels, routes, bookings,

Enquiries etc.

**Packages—**Admin will create the packages and Manage the packages(Create,Update,delete)

**Users-** Admin view all Information of all users.

**Booking-** Admin will responsible for manage booking. Admin can confirm and cancel a booking of

traveler.

**Manage issues/ Complaints**—Admin can take action on any issue /complaint raised by

user(traveller) and Put remark.

**Manage Enquiries—**admin can manage all enquiries raised by users(traveller).

**Manage pages-** Admin can edit the info of all pages that are display on the website,

**Dashboard-** Here admin can view all count of booking, issues , Enquiries and Users .

**Change password---** Admin can change own password.

**2.USER(TRAVELLER) MODULE:**

**Signup-** User can register your self for bookig.

**Signin-** Here user can login with valid username and password.

**ForgotPassword—**User can recover his/her own password.

**My Profile-** user can update own profile.**Tour history-**After login user can book any tour that will show in Tour history. User can cancel

his/her booking before 24 hr of travelling.

**Change Password---- User can own Password.**

**Write-use—Here user can raise any issue related to booking. Cancelation etc.**

**3.Guest MODULE:**

Guest user can visit the website and view the all content of website. Guest user can also Enquiry.

**SYSTEM ENVIRONMENT**

**2.1 Hardware Configuration**

1. Pentium IV Processor

2. 512 MB RAM

3. 40GB HDD

4. 1024 \* 768 Resolution Color Monitor

**Note: This is not the “System Requirements”.**

**2.2 Software Configuration**

1. OS : Windows XP

2.PHP Triad (PHP5.6, MySQL, Apache, and PHPMyAdmin)

**2.3 Software Features**

**2.3.1 PHP TRIAD**

PHPTriad installs a complete working PHP/MySQL server environment on Windows

platforms (9x/ NT). Installs PHP, MySQL, Apache, and PHPMyAdmin.

**PHP**

**PHP** is a scripting language originally designed for producing dynamic web pages. It has evolved to

include a command line interface capability and can be used in standalone graphical applications. While PHP

was originally created by Rasmus Lerdorf in 1995, the main implementation of PHP is now produced by **The**

**PHP Group** and serves as the *de facto* standard for PHP as there is no formal specification. PHP is free

software released under the PHP License, however it is incompatible with the GNU General Public License.

(GPL), due to restrictions on the usage of the term *PHP*. It is a widely-used general-purpose

scripting language that is especially suited for web development and can be embedded into HTML.

It generally runs on a web server, taking PHP code as its input and creating web pages as output. It

can be deployed on most web servers and on almost every operating system and platform free of

charge. PHP is installed on more than 20 million websites and 1 million web servers.

*PHP* originally stood for Personal Home Page. It began in 1994 as a set of Common Gateway Interface

binaries written in the C programming language by the Danish/Greenlandic programmer Rasmus Lerdorf. Lerdorf

initially created these Personal Home Page Tools to replace a small set of Perl scripts he had been using to

maintain his personal homepage. The tools were used to perform tasks such as displaying his résumé and recording

how much traffic his page was receiving. He combined these binaries with his Form Interpreter to create PHP/FI,

which had more functionality. PHP/FI included a larger implementation for the C programming language and

could communicate with databases, enabling the building of simple, dynamic web applications.

Lerdorf released PHP publicly on June 8, 1995 to accelerate bug location and improve the code. This

release was named PHP version 2 and already had the basic functionality that PHP has today. This included

Perl-like variables, form handling, and the ability to embed HTML. The syntax was similar to Perl but was

more limited, simpler, and less consistent. Zeev Suraski and Andi Gutmans, two Israeli developers at the

Technion IIT, rewrote the parser in 1997 and formed the base of PHP 3, changing the language’s name to the

recursive initialism *PHP: Hypertext Preprocessor*. The development team officially released PHP/FI 2 in

November 1997 after months of beta testing. Afterwards, public testing of PHP 3 began, and the official

launch came in June 1998. Suraski and Gutmans then started a new rewrite of PHP’s core, producing the

Zend Engine in 1999. They also founded Zend Technologies in Ramat Gan, Israel.

On May 22, 2000, PHP 4, powered by the Zend Engine 1.0, was released. On July 13, 2004, PHP 5

was released, powered by the new Zend Engine II. PHP 5 included new features such as improved

support for object-oriented programming, the PHP Data Objects extension (which defines a lightweight

and consistent interface for accessing databases), and numerous performance enhancements. The most

recent update released by The PHP Group is for the older PHP version 4 code branch.In 2008, PHP 5 became the only stable version under development. Late static binding has

been missing from PHP and will be added in version 5.3. PHP 6 is under development alongside

PHP 5. Major changes include the removal of register\_globals, magic quotes, and safe mode. The

reason for the removals was because register\_globals had given way to security holes, and magic

quotes had an unpredictable nature, and was best avoided. Instead, to escape characters, Magic

quotes may be substituted with the addslashes() function, or more appropriately an escape

mechanism specific to the database vendor itself like mysql\_real\_escape\_string() for MySQL.

PHP does not have complete native support for Unicode or multibyte strings; Unicode support will be

included in PHP 6. Many high profile open source projects ceased to support PHP 4 in new code as of

February 5, 2008, due to the GoPHP5 initiative, provided by a consortium of PHP developers promoting the

transition from PHP 4 to PHP 5. It runs in both 32-bit and 64-bit environments, but on Windows the only

official distribution is 32-bit, requiring Windows 32-bit compatibility mode to be enabled while using IIS in a

64-bit Windows environment. There is a third-party distribution available for 64-bit Windows.

**Syntax**

<html>

<head>

<title>PHP Test </title>

</head>

<body>

**<?php echo “<p> Hello World </p>”; ?>**

</body></html>

Note : - Code in bold letters shows the PHP code embedded within HTML

**MY SQL**

What is a database? Quite simply, it’s an organized collection of data. A database

management system (DBMS) such as Access, FileMaker Pro, Oracle or SQL Server provides you

with the software tools you need to organize that data in a flexible manner. It includes facilities to

add, modify or delete data from the database, ask questions (or queries) about the data stored in the

database and produce reports summarizing selected contents.

MySQL is a multithreaded,multi-user SQL database management system(DBMS). The basic

program runs as a server providing multi-user access to a number of databases.Originally financed in

a similar fashion to the JBoss model, MySQL was owned and sponsored by a single for-profit firm,

the Swedish company MySQLAB now a subsidiary of Sun Microsystem , which holds the copyright

to most of the codebase. The project’s source code is available under terms of the GNU General

Public Licence, as well as under a variety of proprietory agreements.

MySQL is a database.The data in MySQL is stored in database objects called tables.A table

is a collections of related data entries and it consists of columns and rows.Databases are useful when

storing information categorically. A company may have a database with the following tables:

“Employees”, “Products”, “Customers” and “Orders”.

**Database Tables**

A database most often contains one or more tables. Each table is identified by a name (e.g.

“Customers” or “Orders”). Tables contain records (rows) with data.

**Queries**

A query is a question or a request.With MySQL, we can query a database for specific

information and have a recordset returned.

**Create a connection to a database**

Before you can access data in a database, you must create a connection to the database.In PHP,

this is done with the mysqli\_connect() function.

**Syntax**

**Example**

In the following example we store the connection in a variable ($con) for later use in the script.

The “die” part will be executed if the connection fails:

**Closing a Connection**

The connection will be closed automatically when the script ends. To close the connection

before, use the mysql\_close() function: **Create a Database**

The CREATE DATABASE statement is used to create a database in MySQL.

**Syntax**

CREATE DATABASE database\_name

To get PHP to execute the statement above we must use the mysql\_query() function. This function is

used to send a query or command to a MySQL connection.

**Create a Table**

The CREATE TABLE statement is used to create a table in MySQL

**Syntax**

CREATE TABLE table\_name

(

column\_name1 data\_type,

column\_name2 data\_type,

column\_name3 data\_type,

*....*

)

**MySQL Functions**

mysql\_affected\_rows — Get number of affected rows in previous MySQL

operation mysql\_change\_user — Change logged in user of the active

connection mysql\_client\_encoding — Returns the name of the character set

mysql\_close — Close MySQL connection

mysql\_connect — Open a connection to a MySQL

Server mysql\_create\_db — Create a MySQL database

mysql\_data\_seek — Move internal result pointer

mysql\_db\_name — Get result data

mysql\_db\_query — Send a MySQL query

mysql\_drop\_db — Drop (delete) a MySQL database

mysql\_errno — Returns the numerical value of the error message from previous MySQL

operation mysql\_error — Returns the text of the error message from previous MySQL

operation mysql\_escape\_string — Escapes a string for use in a mysql\_query

mysql\_fetch\_array — Fetch a result row as an associative array, a numeric array,

or both mysql\_fetch\_assoc — Fetch a result row as an associative array

mysql\_fetch\_field — Get column information from a result and return as an

object mysql\_fetch\_lengths — Get the length of each output in a result

mysql\_fetch\_object — Fetch a result row as an objectwsmysql\_num\_rows — Get number of rows in result mysql\_pconnect

— Open a persistent connection to a MySQL server

mysql\_ping — Ping a server connection or reconnect if there is no

connection mysql\_query — Send a MySQL query

mysql\_result — Get result data mysql\_select\_db

— Select a MySQL database mysql\_set\_charset

— Sets the client character set mysql\_stat —

Get current system status mysql\_tablename —

Get table name of field mysql\_thread\_id —

Return the current thread ID

mysql\_unbuffered\_query — Send an SQL query to MySQL, without fetching and buffering

the result (*See Appendix 2 for more My\_SQL Functions.*)

**phpMAdmin**

**phpMyAdmin** is an open source tool written in PHP intended to handle the administration of MySQL

over the World Wide Web. phpMyAdmin supports a wide range of operations with MySQL.Currently it can

create and drop databases, create/drop/alter tables, delete/edit/add fields, execute any SQL statement, manage

users and permissions, and manage keys on fields. while you still have the ability to directly execute any SQL

statement. phpMyAdmin can manage a whole MySQL server (needs a super-user) as well as a single

database. To accomplish the latter you’ll need a properly set up MySQL user who can read/write only the

desired database. It’s up to you to look up the appropriate part in the MySQL manual.phpMyAdmin can:

·

browse and drop databases, tables, views, fields and indexes

·

create, copy, drop, rename and alter databases, tables, fields and indexes

·

maintenance server, databases and tables, with proposals on server configuration

·

execute, edit and bookmark any SQL-statement, even batch-queries

·

load text files into tables

·

create and read dumps of tables

·

export data to various formats: CSV, XML, PDF, ISO/IEC 26300 - OpenDocument Text and

Spreadsheet, Word, Excel and LATEX formats

·

administer multiple servers

·

manage MySQL users and privileges

·

check referential integrity in MyISAM tables

·

using Query-by-example (QBE), create complex queries automatically connecting required tables

·

create PDF graphics of your Database layout

·

search globally in a database or a subset of it

·

transform stored data into any format using a set of predefined functions, like displaying

BLOB-data as image or download-link

·

support InnoDB tables and foreign keys

·

support mysqli, the improved MySQL extension **A word about users:**

Many people have difficulty understanding the concept of user management with regards

to phpMyAdmin. When a user logs in to phpMyAdmin, that username and password are passed

directly to MySQL. phpMyAdmin does no account management on its own (other than allowing

one to manipulate the MySQL user account information); all users must be valid MySQL users.

1) phpMyAdmin can compress (Zip, GZip -RFC 1952- or Bzip2 formats) dumps and CSV exports

if you use PHP with Zlib support (—with-zlib) and/or Bzip2 support (—with-bz2). Proper

support may also need changes in php.ini.a phpMyAdmin screen appears as shown below.

**Requirements**

o

**PHP**

o

You need PHP 5.2.0 or newer, with session support and the Standard PHP Library

(SPL) extension.

o

To support uploading of ZIP files, you need the PHP zip extension.

o

For proper support of multibyte strings (eg. UTF-8, which is currently default),

you should install mbstring and ctype extensions.

o

You need GD2 support in PHP to display inline thumbnails of JPEGs

(“image/jpeg: inline”) with their original aspect ratio

o

When using the “cookie” authentication method, the mcrypt extension is strongly

suggested for most users and is required for 64–bit machines. Not using mcrypt

will cause phpMyAdmin to load pages significantly slower.

**Apache Web server**

Often referred to as simply *Apache*, a public-domain open source Web server developed by a

loosely-knit group of programmers. The first version of Apache, based on the NCSA httpd Web

server, was developed in 1995.

Core development of the Apache Web server is performed by a group of about 20 volunteer

programmers, called the *Apache Group.* However, because the source code is freely available,

anyone can adapt the server for specific needs, and there is a large public library of Apache add

ons. In many respects, development of Apache is similar to development of the Linux operating

system.

The original version of Apache was written for UNIX, but there are now versions that run

under OS/ 2, Windows and other platforms. The name is a tribute to the Native American Apache

Indian tribe, a tribe well known for its endurance and skill in warfare. A common

misunderstanding is that it was called Apache because it was developed from existing NCSA code

plus various patches, hence the name *a patchy server*, or Apache server.

Apache consistently rates as the world’s most popular Web server according to analyst

surveys. Apache has attracted so much interest because it is full-featured, reliable, and free.

Originally developed for UNIX™ operating systems, Apache has been updated to run on

Windows, OS/2, and other platforms. One aspect of Apache that some site administrators find

confusing — especially those unfamiliar with UNIX-style software — is its configuration scheme.

Instead of using a point-and-click graphic user interface (GUI) or Windows Registry keys as most

other

**Configuration Files**

Apache uses a system of three text files for managing its configuration data. All three of these files

(almost always) appear in Apache’s ./conf directory and are designed to be edited by system administrators:1. httpd.conf for general settings

2. srm.conf for resource settings

3. access.conf for security settings

When Apache first starts, these files are processed in the order shown above. Originally, the

initial installation of Apache included default entries within each of the three files. In the most

recent versions of Apache, however, the default installation has changed. Now httpd.conf is treated

as the “master” configuration file and it contains all of the settings. Both srm.conf and access.conf

still exist in the installation, but they contain no settings and are empty except for some comments.

**Inside Httpd.conf**

Traditionaly httpd.conf contained general settings such as the ServerName and Port

number.These entries appear as follows in the file: ServerName compnetworking.about.com Port

80 The term “httpd” stands for *HTTP Daemon*. Recall that in a UNIX environment, the term

*daemon* refers to a type of process designed to launch at system boot and continue running for

very long periods of time. This file contains a number of other entries (technically called

directives), but for most of these,modifications are optional. Probably the most useful of these

entries is ServerAdmin.

**Access and Security Settings**

It is recommended practice now for Apache administrators to manage their resource and

security settings from httpd.conf. Administrators of older versions of Apache can simply cut their

entries from srm.conf and access.conf and paste them into the master file. If an administrator

wants to go one step further and delete the two empty files, they should also place the following

entries in httpd.conf to prevent Apache from attempting to access them.**INPUT DESIGN:**

Input design is a part of overall system design. The main objective during the input

design is as given below:



To produce a cost-effective method of input.



To achieve the highest possible level of accuracy.



To ensure that the input is acceptable and understood by the user.

**INPUT STAGES:**

The main input stages can be listed as below:



Data recording



Data transcription



Data conversion



Data verification



Data control



Data transmission



Data validation



Data correction

**INPUT TYPES:**

It is necessary to determine the various types of inputs. Inputs can be categorized as

follows:



External inputs, which are prime inputs for the system.



Internal inputs, which are user communications with the system.



Operational, which are computer department’s communications to the

system?



Interactive, which are inputs entered during a dialogue.**INPUT MEDIA:**

At this stage choice has to be made about the input media. To conclude about the

input media consideration has to be given to;



Type of input



Flexibility of format



Speed



Accuracy



Verification methods



Rejection rates



Ease of correction



Storage and handling requirements



Security



Easy to use



Portability

Keeping in view the above description of the input types and input media, it can be said

that most of the inputs are of the form of internal and interactive. As

Input data is to be the directly keyed in by the user, the keyboard can be considered to be

the most suitable input device.

**OUTPUT DESIGN:**

Outputs from computer systems are required primarily to communicate the results of

processing to users. They are also used to provide a permanent copy of the results for

later consultation. The various types of outputs in general are:



External Outputs whose destination is outside the organization.



Internal Outputs whose destination is with in organization and they are the User’s

main interface with the computer.



Operational outputs whose use is purely with in the computer department.



Interface outputs, which involve the user in communicating directly with the

system.**OUTPUT DEFINITION**

The outputs should be defined in terms of the following points:



Type of the output



Content of the output



Format of the output



Location of the output



Frequency of the output



Volume of the output



Sequence of the output

It is not always desirable to print or display data as it is held on a computer. It should be

decided as which form of the output is the most suitable.

**For Example**



Will decimal points need to be inserted



Should leading zeros be suppressed.

**OUTPUT MEDIA:**

In the next stage it is to be decided that which medium is the most appropriate for the

output. The main considerations when deciding about the output media are:



The suitability for the device to the particular application.



The need for a hard copy.



The response time required.



The location of the users



The software and hardware available.

Keeping in view the above description the project is to have outputs mainly coming

under the category of internal outputs. The main outputs desired according to the

requirement specification are: The outputs were needed to be generated as a hard copy and as well as queries to be

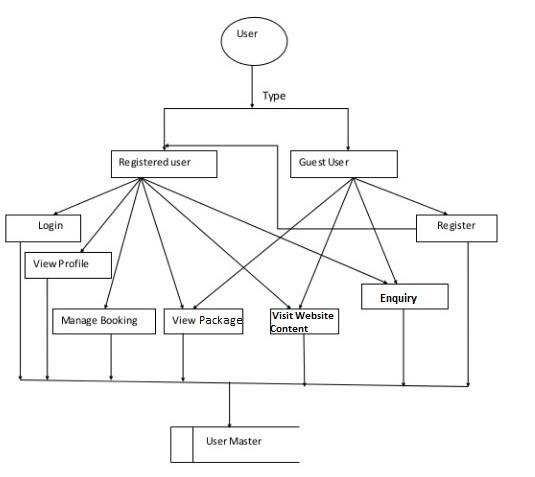
viewed on the screen. Keeping in view these outputs, the format for the output is taken

from the outputs, which are currently being obtained after manual processing. The

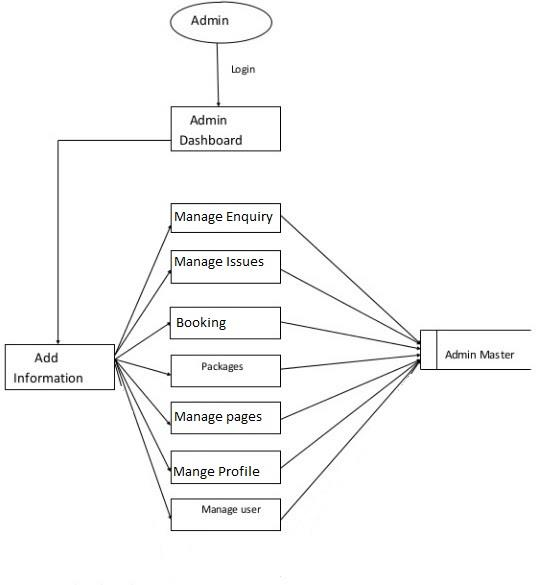
standard printer is to be used as output media for hard copies.

**DFD (Context level Diagram):**

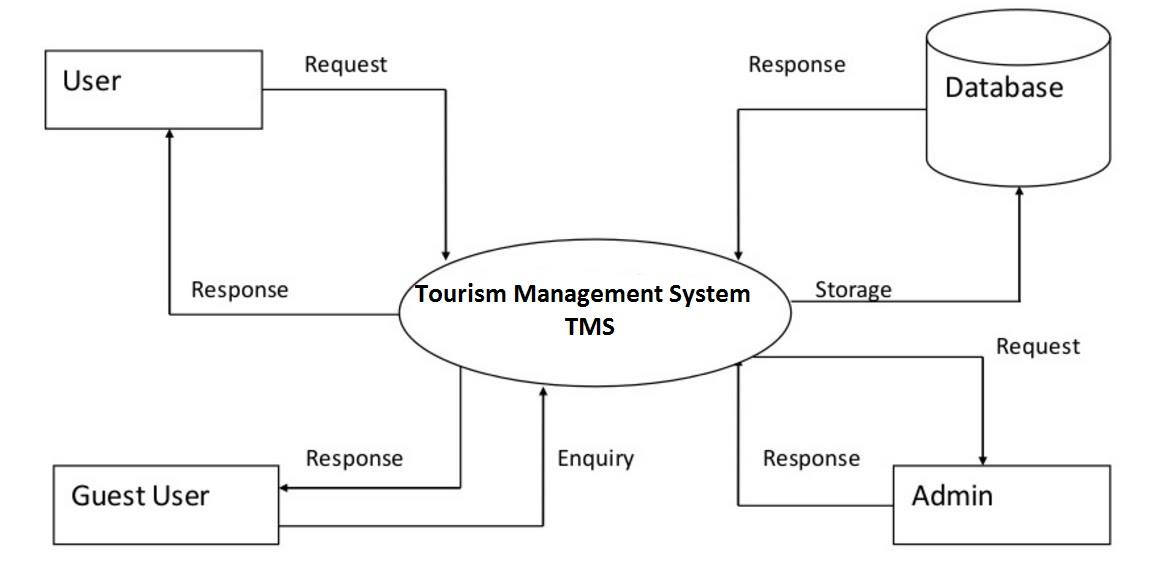
**a.Users**

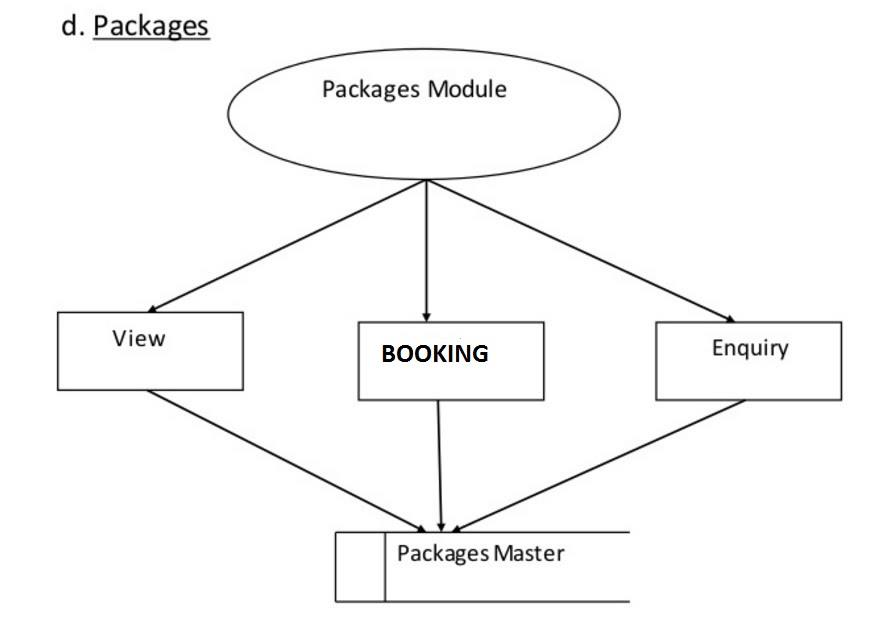


**b.Admin**

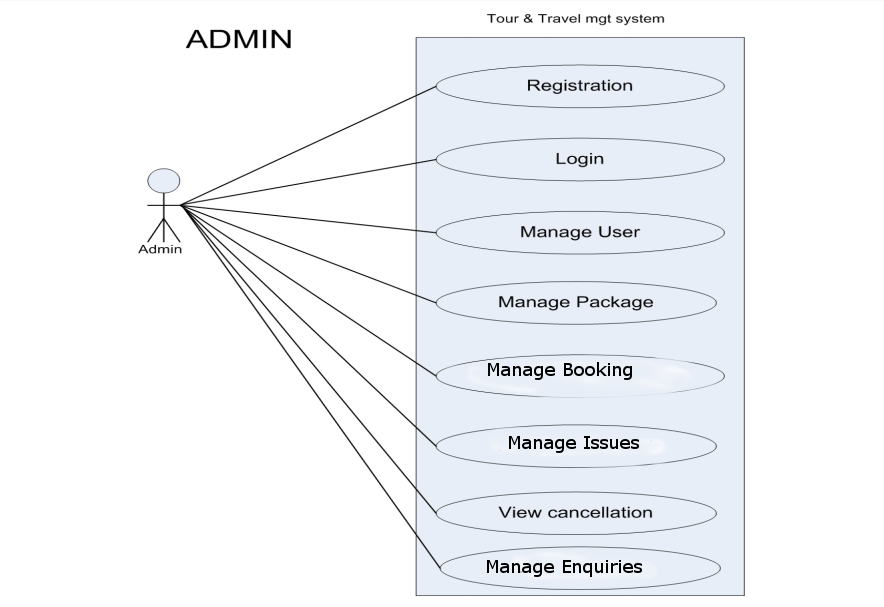


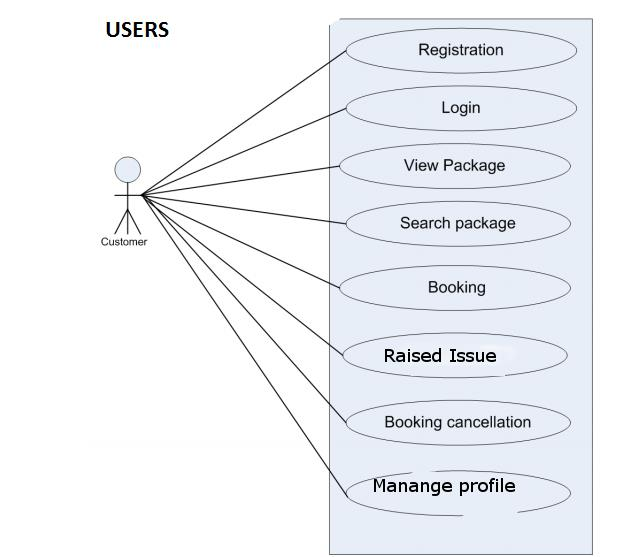
**C:User**

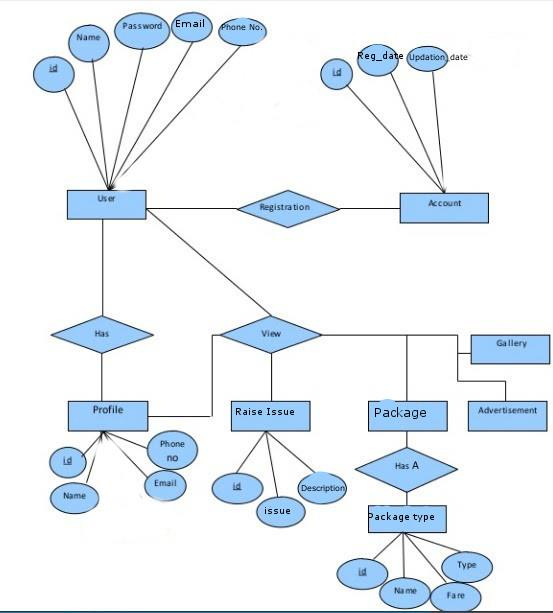


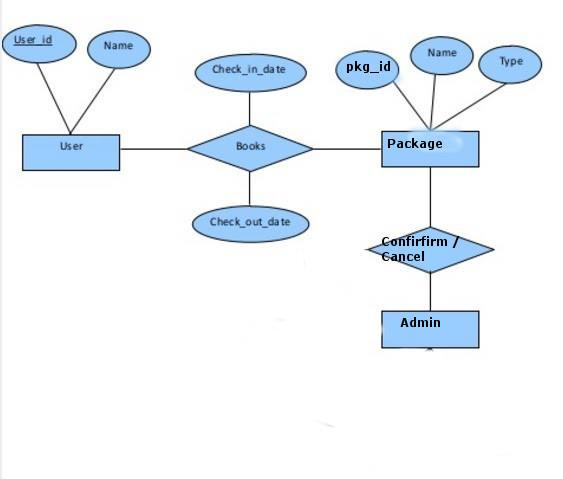


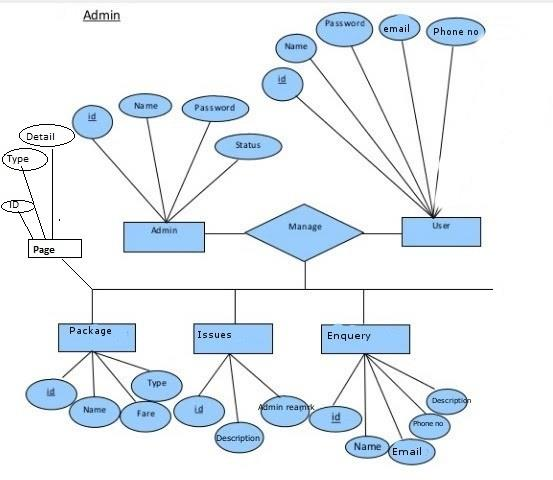
**UML Diagram**



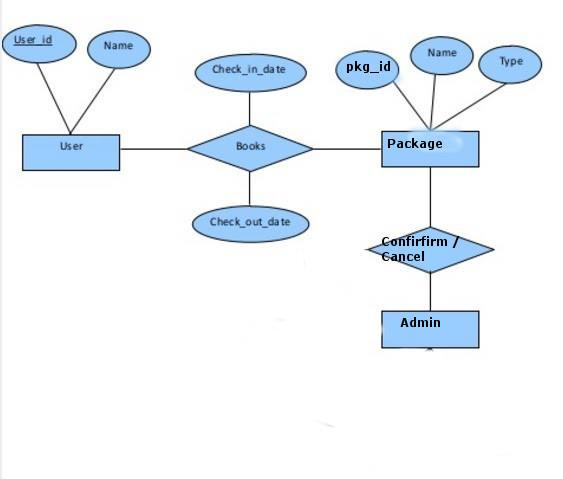


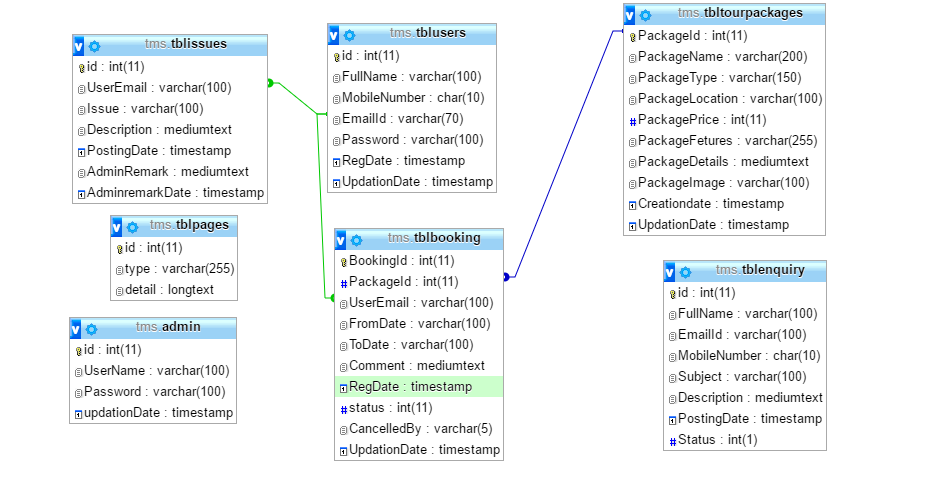






**ER DIAGRAM OF PACKAGE :**





**Architecture flow :(N-Tier)**

N-Tier Applications can easily implement the concepts of Distributed Application

Design and Architecture. The N-Tier Applications provide strategic benefits to Enterprise

Solutions. While 2-tier, client-server can help us create quick and easy solutions and may

be used for Rapid Prototyping, they can easily become maintenance and security night

mare The N-tier Applications provide specific advantages that are vital to the business

continuity of the enterprise. Typical features of a real life n-tier may include the following:

Security



Availability and Scalability



Manageability



Easy Maintenance



Data Abstraction

The above mentioned points are some of the key design goals of a successful n-tier

application that intends to provide a good Business Solution.

**Definition:**

Simply stated, an n-tier application helps us distribute the overall functionality into various

tiers or layers:



Presentation Layer



Business Logic layer



Data Link Layer



Database/Data Store

Each layer can be developed independently of the other provided that it adheres to the

standards and communicates with the other layers as per the specifications. This is the

one of the biggest advantages of the n-tier application. Each layer can potentially treat

the other layer as a ‘Block-Box’. In other words, each layer does not care how other layer

processes the data as long as it sends the right data in a correct format.

**1. Presentation Layer:**

Also called as client layer, comprises of components that are dedicated to presenting

the data to the user. For example: Windows/Web Forms and buttons, edit boxes, Text

boxes, labels, grids, etc.

**2. Business Logic Layer:**

This layer encapsulates the Business rules or the business logic of the

encapsulations. To have a separate layer for business logic is of a great advantage. This

is because any changes in Business Rules can be easily handled in this layer. As long as

the interface between the layers remains the same, any changes to the

functionality/processing logic in this layer can be made without impacting the others. A lot

of client-server apps failed to implement successfully as changing the business logic was

a painful process.

**3. Data Link Layer:**

This layer comprises of components that help in accessing the Database. If used in

the right way, this layer provides a level of abstraction for the database structures. Simply

put changes made to the database, tables, etc do not affect the rest of the application

because of the Data Access layer. The different application layers send the data requests

to this layer and receive the response from this layer.

**4. Database Layer:**

This layer comprises of the Database Components such as DB Files, Tables, Views,

etc. The Actual database could be created using SQL Server, Oracle, Flat files, etc. In an

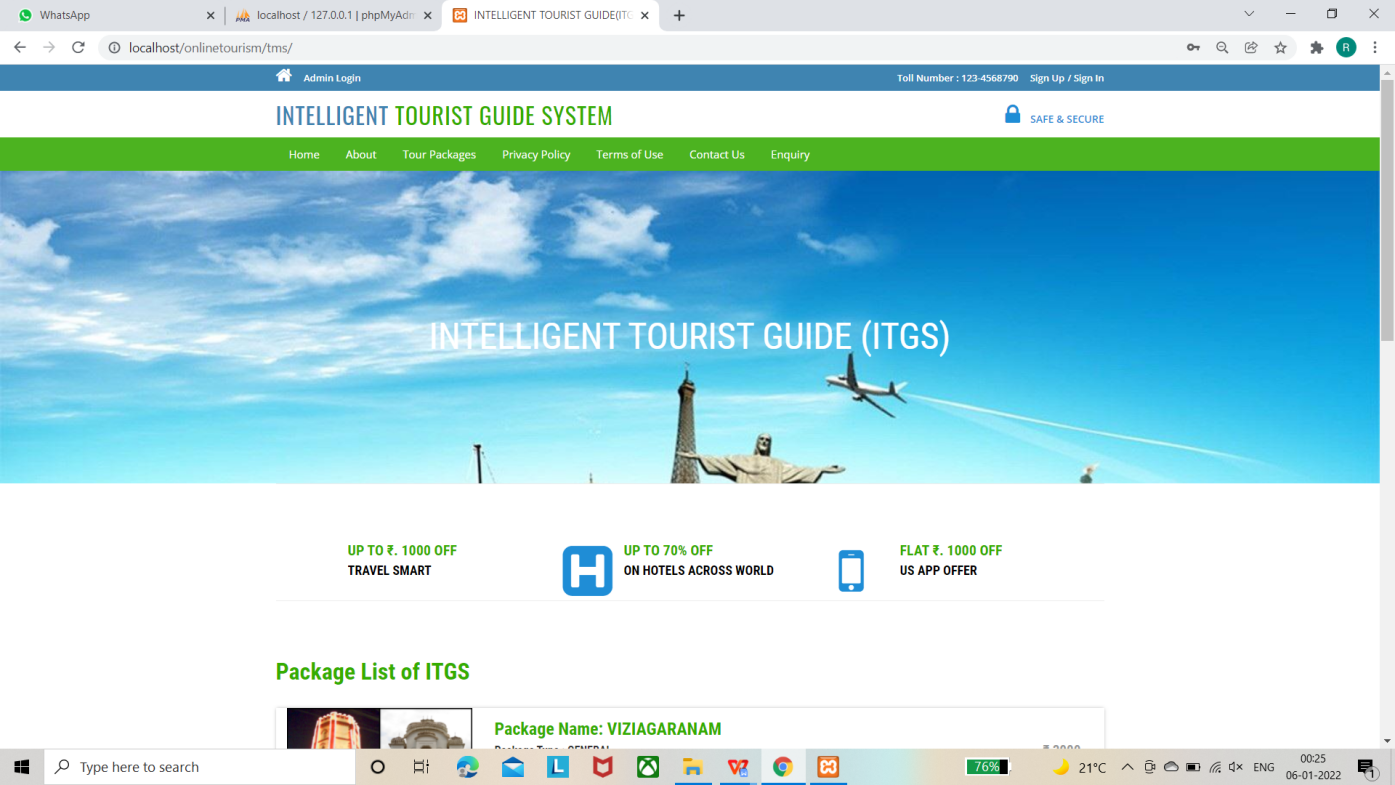
n-tier application, the entire application can be implemented in such a way that it is

independent of the actual Database. For instance, you could change the Database

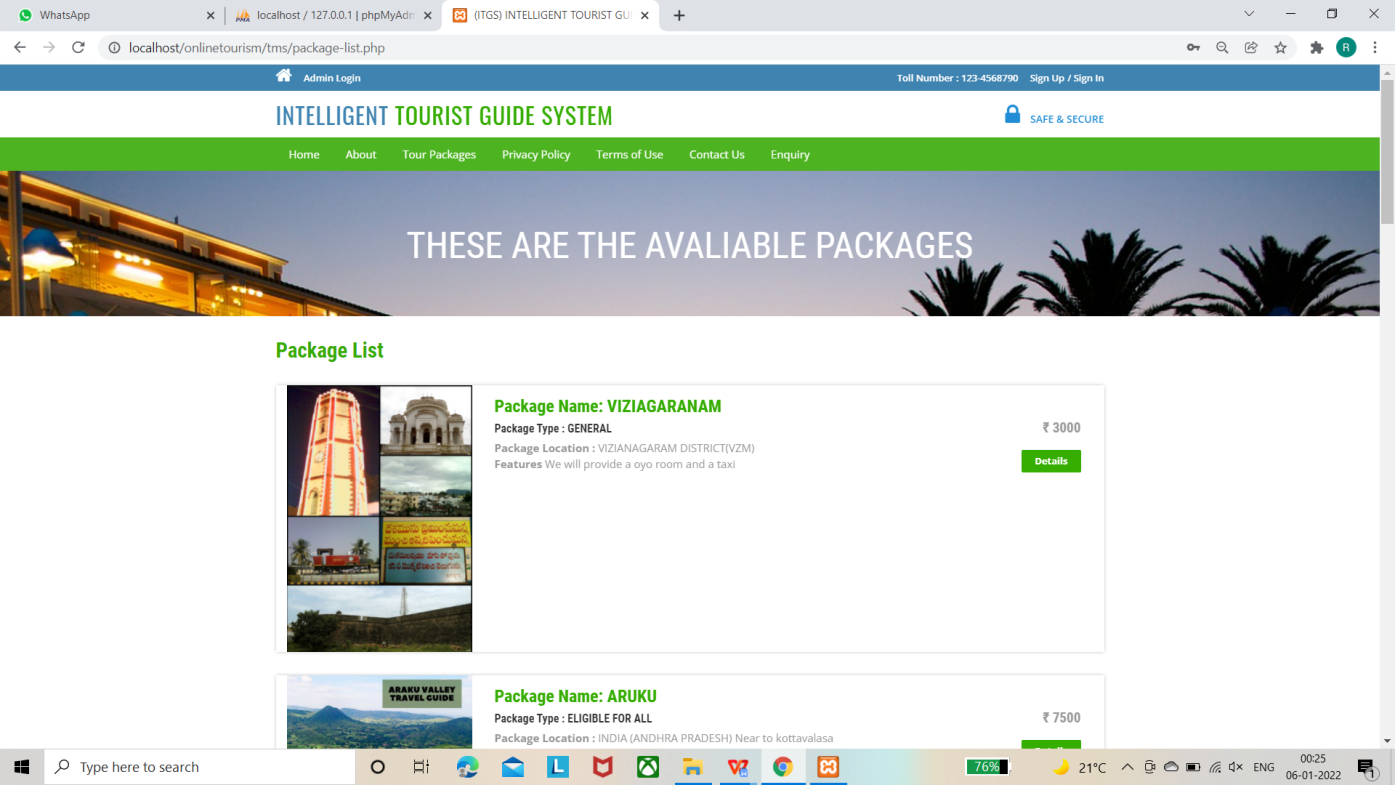
Location with minimal changes to Data Access Layer. The rest of the Application should

remain unaffected.

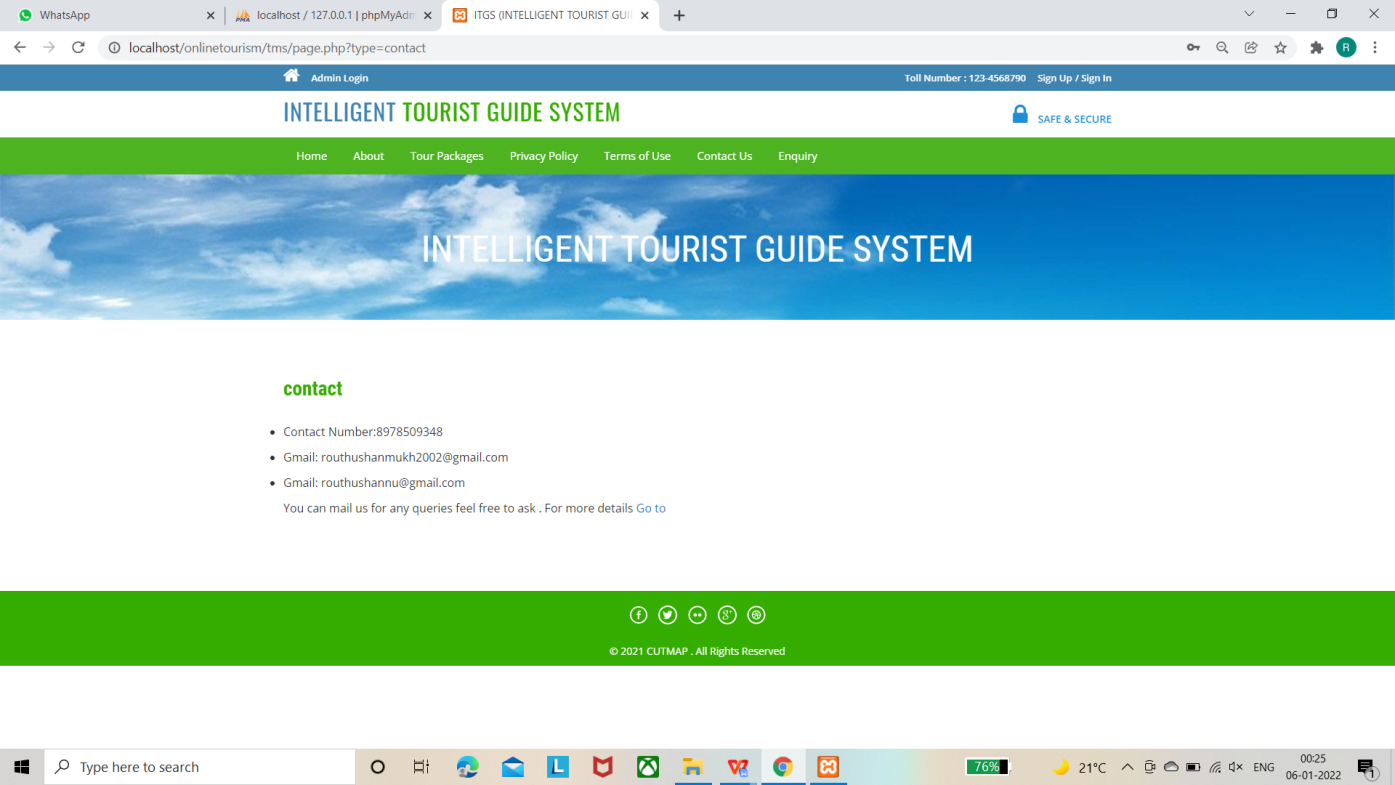
OUTPUT OF PROJECT IMPLEMENTATION :



This is the home page.



These are different types of tour packages available .



Contact details .

**CONCLUSION**

To conclude the description about the project : The project, developed using PHP and

MySQL is based on the requirement specification of the user and the analysis of the existing

system, with flexibility for future enhancement.

The expanded functionality of today’s software requires an appropriate approach towards software

development. This hostel management software is designed for people who want to manage various activi

ties in the hostel. For the past few years the number of educational institutions are increasing rapidly.

Thereby the number of hostels are also increasing for the accommodation of the students studying

in this institution. And hence there is a lot of strain on the person who are running the hostel and

software’s are not usually used in this context. This particular project deals with the problems on

managing a hostel and avoids the problems which occur when carried manually.

Identification of the drawbacks of the existing system leads to the designing of

computerized system that will be compatible to the existing system with the system which is more

user friendly and more GUI oriented.

References:

* <https://nevonprojects.com/online-pizza-ordering-system/>
* https://www.tutorialspoint.com/index.htm
* https://www.javatpoint.com
* https://www.w3schools.com
* https://html.com