CHAPTER 3

IMPLEMENTATION

In this chapter the implementation details of the project has been specified.

3.1 HARDWARE SPECIFICATIONS:

- 40 GB hard disk space.
- 2 GB RAM.
- Hi-Speed Network Connectivity.

3.2 SOFTWARE SPECIFICATIONS:

- Windows(x64) Operating System.
- Sublime Text.
- MySQL Server.
- Apache Server.
- Xampp.

3.2.1 LANGUAGE USED FOR IMPLEMENTATION

The languages used for implementation are as follows:

- Front end:- PHP,HTML
- Back end:- MySQL

PHP:-

PHP is a server-side scripting language designed primarily for web development but also used as a general-purpose programming language. Originally created by Rasmus Lerdorf in 1994, the PHP reference implementation is now produced by The PHP Development Team. PHP originally stood for Personal Home Page, but it now stands for the recursive acronym.

PHP code may be embedded into HTML or HTML5 mark up, or it can be used in combination with various web template systems, web content management systems and web frameworks. PHP code is usually processed by a PHP interpreter implemented as a module in the web server or as a Common Gateway Interface (CGI) executable. The web server software combines the results of the interpreted and executed PHP code, which may be any type of data, including images, with the generated web page. PHP code may also be executed with a command-line interface (CLI) and can be used to implement standalone graphical.

HTML:-

Hypertext Mark-up Language (HTML) is the standard mark-up language for creating web pages and web applications. With Cascading Style Sheets (CSS) and JavaScript it forms a triad of cornerstone technologies for the World Wide Web. Web browsers receive HTML documents from a web server or from local storage and render them into multimedia web pages.

MySQL:-

MySQL is an open-source relational database management system(RDBMS). Its name is a combination of "My", the name of co-founder Michael Widenius daughter, and "SQL", the abbreviation 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. For proprietary use, several paid editions are available, and offer additional functionality.

3.2.2 PLATFORM USED FOR IMPLEMENTATION

XAMPP

XAMPP is a free and open source cross-platform web server solution stack package developed by Apache Friends, consisting mainly of the Apache HTTPServer, MariaDB database, and interpreters for scripts written in the PHP and Perl programming languages. XAMPP stands for Cross-Platform (X), Apache (A), MariaDB (M), PHP (P) and Perl (P). It is a simple, lightweight Apache distribution that makes it extremely easy for developers to create a local web server for testing and deployment purposes. Everything needed to set up a web server – server application (Apache), database (MariaDB), and scripting language (PHP) – is included in an extractable file.

XAMPP is also cross-platform, which means it works equally well on Linux, Mac and Windows. Since most actual web server deployments use the same components as XAMPP, it makes transitioning from a local test server to a live server extremely easy as well.

PhpMyAdmin

PhpMyAdmin is a free and open source administration tool for MySQL and MariaDB. As a portable web application written primarily in PHP, it has become one of the most popular MySQL administration tools, especially for web hosting services.

3.3 SQL COMMANDS AND QUERIES

The queries used for creating these tables are as follow:

- ACCOMODATION

```
CREATE TABLE `accomodation` (
`acc_id` int(11) NOT NULL,
`acc_type` varchar(35) NOT NULL,
`acc_price` double NOT NULL,
`acc_slot` int(4) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

- BOOKED

```
  CREATE TABLE `booked` (
 `book_id` int(11) NOT NULL,
 `book_by` varchar(50) NOT NULL,
 `book_contact` varchar(15) NOT NULL,
 `book_address` varchar(100) NOT NULL,
 `book_name` varchar(100) NOT NULL,
 `book_age` int(11) NOT NULL,
 `book_gender` varchar(15) NOT NULL,
 `book_departure` date NOT NULL,
 `dest_id` int(11) NOT NULL,
   `acc_id` int(11) NOT NULL,
  `origin_id` int(11) NOT NULL,
  `book_tracker` varchar(35) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

- DESTINATION

```
   CREATE TABLE `destination` (
  `dest_id` int(11) NOT NULL,
  `dest_destination` varchar(35) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

- ORIGIN

```
   CREATE TABLE `origin` (
  `origin_id` int(11) NOT NULL,
  `origin_desc` varchar(35) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

- STATUS

```
    CREATE TABLE `status` (
  `stat_id` int(11) NOT NULL,
  `stat_desc` varchar(20) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

- TRANSACTION

```
   CREATE TABLE `transaction` (
  `trans_id` int(11) NOT NULL,
  `trans_time` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE
```

```
 CURRENT_TIMESTAMP,
`trans_payment` double NOT NULL,
`trans_passenger` varchar(100) NOT NULL,
`trans_age` int(11) NOT NULL,
`trans_gender` varchar(15) NOT NULL,
`acc_id` int(11) NOT NULL,
`origin_id` int(11) NOT NULL,
`dest_id` int(11) NOT NULL,
`stat_id` int(11) DEFAULT '1',
`trans_refunded` tinyint(4) NOT NULL DEFAULT '0'
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

- USER

```
  CREATE TABLE `user` (
 `user_id` int(11) NOT NULL,
 `user_account` varchar(50) NOT NULL,
 `user_password` varchar(35) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Trigger

```
DELIMITER $$
        CREATE TRIGGER `insert` AFTER INSERT ON `booked`
         FOR EACH ROW INSERT into `addpassenger` (`pass_name`,`time`) VALUES ((SELECT
        book_id FROM `booked` ORDER by book_name desc LIMIT 1), CURRENT_TIMESTAMP())
$$  DELIMITER;
```

Stored Procedure

```
DELIMITER $$
CREATE DEFINER=`root`@`localhost` PROCEDURE `proc`()
 NO SQL
 DETERMINISTIC
```

```
SELECT *
                FROM booked b
                INNER JOIN accomodation a
                ON b.acc_id = a.acc_id
                WHERE book_tracker = '5de271537eb5d'$$
DELIMITER ;
```

3.4 OUTPUT TESTING

- While executing php mysql connection code we were not able to make the connection of backend mysql to front end php. So to solve this problem we had to create a new mysql user with password. After this the connection was successful.
- The connection was successful but the data entered in front end was not storing in backend, since all the attributes data types in backend were not set to varchar. So we modified the php code and mysql query accordingly.
- If we enter wrong password, local host says wrong password. If XAMP server is not started, then we cannot run local host