VIDHYADEEP INSTITUTE OF COMPUTER & INFORMATION TECHNOLOGY, ANITA-KIM.

VEER NARMAD SOUTH GUJARAT UNIVERSITY

(VNSGU)

PROJECT REPORT

ON

**“KARFIND”**

AS A PARTIAL REQUIREMENT FOR THE DEGREE

OF

BACHELOR OF COMPUTER APPLICATIONS

[B.C.A]

3 YEARS INTEGRATED COURSE

YEAR : 2021 - 2022

**-: Guided by :-** **-: Submitted by :-** Miss Drashti Bhatt Utpalsinh A. Parmar

(Internal Guide) (E19110403000110042)

1. **“We start this dissertation in the name of God, Who is most gracious and most merciful.”**
2. Successful completion of any task gives us the great satisfaction and internal strength regarding the future problems, but at any stage the person alone never exists, he is always accompanied by some peoples who used to give him supports and suggestions for successful completion of this work. Therefore it’s a matter of great pleasure for us to thank all those people who admires us and gives there kind hearted support at every stage of our work.
3. First of all we would like to give an honor to our grate college **“VIDHYADEEP INSTITUTE OF COMPUTER & INFORMATION TECHNOLOGY”,** where we got a platform to learn the new techniques and concept to enhance our skills specially programming and professionals. Now we would like to express our deep hearted thanks to our project Guide **Miss Drashti Bhattt**. In spite of her hectic schedule she always responded us with smile for discussing the problems came during project work. She allows us with open hand to try for new solutions.
4. It was our great pleasures for us to study in this college where professors are eager to learn the new technology and provide their fill support to the students to enhance their skills with many folds. We learned many new things form our professors.
5. This was the first steps towards high profile career in B.C.A. It was the great experience of exposing and learning new things in this field. Last, but not least we take this opportunity to thank all those people who have help directly or indirectly in making our project feasible and turn it to a successful piece of work.

Utpalsinh Parmar

|  |  |
| --- | --- |
| Sr. No. | Description |
| 1 | Company Profile |
| 2 | Project Profile |
| 3 | Objective of the Project |
| 4 | Project Category |
| 5 | Environment Description  5.1 Hardware/Software Specification  5.1.1 Development Tools  5.1.2 Employee Side Tools  5.1.3 Server Side Tools  5.2 Other Tools |
| 6 | Analysis Report  6.1 Current System  6.2 Drawback of the Current System  6.3 Requirement Specification  6.4 Proposed System  6.4.1 Introduction  6.4.2 Scope  6.5 Advantages of proposed system  6.6 Data Flow Diagram  6.7 Process Specification  6.8 Data Dictionary  6.9 Table Structure with proper relationship |
| 7 | Design Report  7.1 Site Diagram  7.2 Flow Chart  7.3 Security Issues  7.4 Input Screen Layouts  7.5 Output Reports |
| 8 | Testing Reports  8.1 Testing Issues  8.2 Test Case |
| 9 | Limitation of the System |
| 10 | Future Enhancement of the project |
| 11 | References |

**1. Company Profile**

|  |  |
| --- | --- |
| Company Name | KarFind |
| Proprietor Name | Utpalsinh Arvindsinh Parmar |
| Business Type | Local Search Engine |
| Address | Surat, Gujarat, India. |
| Phone Number | +91 70433-64132 |
| Website | https://www.karfind.com |
| Email | Karfind22@gmail.com |
| Product Range | India |

**2. Project Profile**

|  |  |
| --- | --- |
| Project Title | KarFind |
| Organization | Dream Infotech |
| Project Category | Local Search Engine |
| Project Duration | 3 Months |
| Front – end | PHP Framework (Codeigniter) |
| Back – end | My SQLI |
| Project Guide | Miss Drashti Bhatt |
| Project Guide | Mr. Pintu Teli |
| Submitted by | Utpalsinh Arvindsinh Parmar |
| Submitted to | Vidhyadeep Institute of Computer & Information Technology, Anita-Kim. |

**3. Objectives of The Project**

**The main objectives for the KarFind System for finding services are:**

* This Project will enable users to find services nearby them.
* This Project helps the users choose their desirable vendor for a particular service.
* User can also give feedback for a vendor’s service.
* The vendors can book online order/appointment.
* The user can search for the services according to the price range, area, city and name of vendors.
* Vendors can manage their customers and their schedule, packages as well as orders.

**4. Project Category**

The “KarFind” Project category is Local Search Engine. (Website)

The project entitled is “KarFind” the main aim of the project is to enable its users to find services nearby them.

**5. Environment Description**

* 1. **Hardware and Software Requirement**

The efficient hardware and software configuration requires running the system is as suggest below. The configuration suggested is for better performance. Same functionality or higher configuration will always better.

# SERVER

|  |  |
| --- | --- |
| Software |  |
| Operating System | Windows |
| Browser | Google Chrome, Mozilla Firefox, Internet Explorer. |
| Database | My SQLI |
| Microprocessor | Any but latest is good for performance |
| Memory | 2 GB |
| Hard Disk Space | Minimum 1 GB |

# CLIENT

|  |  |
| --- | --- |
| Operating System | Linux, Windows XP, Windows 2007,Windows Vista, Windows 2008,Windows 10 |
| Browser | Google Chrome, Mozilla Firefox, Internet Explorer. |
| Database | MySQLI |
| Microprocessor | Any but latest is good for performance |
| Memory | Minimum 512 MB |
| Hard Disk Space | Minimum 1 GB |

* + 1. **Development Tools:**
       - Microsoft windows XP,
       - Windows Vista,
       - WAMP server,
       - MySQL 5.0,
       - Internet Explorer,
       - Mozilla Firefox,
       - Opera,
       - Google Chrome,
       - Awesome screenshots
    2. **Server Side Tools:**

PHP Version 5.2.5,

MySQL version 5.0,

* + 1. **Client Side Tools:**
* Web browser
  1. **Other Tools**
* Edraw Max for Drawing DFD’s,
* Process Specification,
* Flowcharts.
* Other Technology Used

## 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: Hypertext Preprocessor.

PHP code may be embedded into HTML or HTML5 code, 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 combines the results of the interpreted and executed PHP code, which may be any type of data, including images, with the generated web page.

## MY SQL

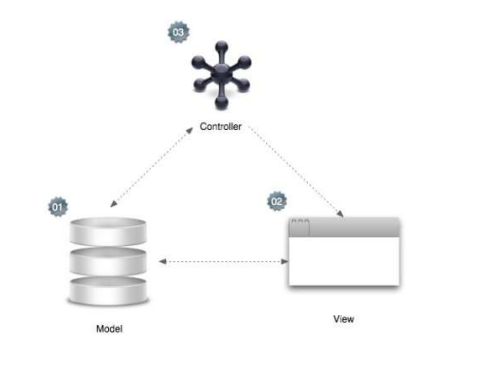
My SQL is an open-source relational database management system (RDBMS). Its name is combinations of “My”, the name of co-founder Michael Widenius' daughter, and "SQL", the abbreviation for Structured Query Language. The MySQL development project has made its source code available under the terms of the 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.

MySQL is a central component of the LAMP open-source web application software stack (and other "AMP" stacks). LAMP is an acronym for "Linux, Apache, MySQL, and Perl/PHP/Python". Applications that use the MySQL database include: TYPO3, Mode, Joomla, Word Press, php DB, MyDB, and Dural. MySQL is also used in many high-profile, large-scale websites, including Google (though not for searches), Face book, Twitter, Flicker, and YouTube.

* **MVC Framework**
* MVC Architecture Part 1:

Introduction to the Architecture

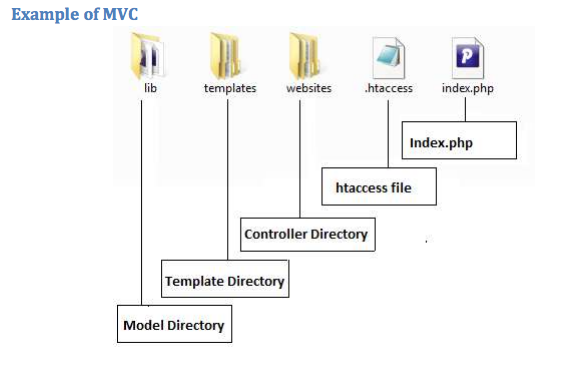
Web development in PHP introduces a powerful architecture for PHP frameworks like Zed, Code Igniter, and Cake PHP – Model-View-Controller (MVC). MVC is more than it meets the eye. It is not just any other web development framework architecture for building elegant and systematic websites, but it has a full capability to support rapid web application development and dynamic interactivity with the database as well. You may find numerous articles on this subject over the Internet and some of them will tell a tale of comprehensive exclusivity as well. This is not a topic to be discussed in one article. A wide prospect such as MVC architecture should be dedicated more articles to elaborate on the complexity of the architecture. Here we try to cover maximum ground as much possible in three parts – the first post shall introduce and talk a wee bit on the MVC architecture, the following will delve deeper and explain the interiors of MVC, and the last article is a tutorial type on creating a website using MVC architecture. MVC methodology typically splits the architecture of the website into 3 distinct parts which are kept operationally separate but interact with each other to deliver all aspects of the website and the administration system. The main aim of the MVC architecture is to separate the business logic and application data from the presentation data to the user. Adhering to MVC architecture benefits you in attaining a perfect design for an enterprise web application.



There is a common control flow in all forms of MVC built web applications. The Controller lies at the core of the architecture and it interacts with the user through the web browser. Behind the scene it communicates with the Model and the View components of the architecture. The Model actually envelops the database so that the Controller can notify and direct the database of the user actions on the browser. It is here that the database is modifies in accordance with the requests submitted by the user. This improves user interactivity and hence the overall seamless experience. On receiving the query request from the user the controller handles the Model and in return Model notifies View to update the user interface (UI) on the screen in accordance to the database change. When the user enters another query, the cycle is restarted.

* MVC Architecture Part 2:

Understanding the Interiors In the previous post we learnt what MVC actually is and how does it work. In this post we shall elaborate our discussion on the interiors of the MVC architecture. To understand the architecture we first suppose and example of a typical website built on the MVC architecture. The following picture gives an idea of the directory structure of the website and so to understand the architecture, we shall understand the structure of mentioned here.



Now, any web developer will know what the index.php and the access files are for, though we shall mention them in our explanation a little later. But the point here is what are the other folders viz. lib, templates, and websites for? Well, these are the MVC folders, and they are labeled according to their roles in the architecture. We shall start from the left most labels and continue on to their right ones.

**6. Analysis Report**

**6.1 Current System**

Before making “KarFind” website admin can manage all tour collection of information manually.

That time only those people can know about the category who are live under that area where the category is manage.

The manual system is now a traditional system because in this fast world to work manually is not reliable. Lets we know how can work manually with manual system.

The main advantage of such system is that the customer and vendor meet face to face with one another, so that all communication which is required between them can be done.

The current system is manual one that collects information in order from the customer and data is processed according to customer requirements.

Today people who don’t have enough knowledge of computer system so it is reliable for the people who want to deal without using computer.

When “KarFind” website is not available admin can store all information of package record or other records on the paper. So admin can take many people to manage the add.

**6.2 Drawback of Current System**

User of manual resources on a large scale so the information collected may have unproductive and irrelevant data.

That time to manage the whole package is very hard.

Generating report is tedious task.

Huge storage space required to store data.

Retrieval and updating is a tedious task.

Since, manual system, lots of error is bound to occur that is human error may cause wrong calculation that may cause financial loss to organization.

Debugging for error may cause major problem because locating error may be tedious task.

Maintenance cost is high.

There is an extensive amount of documentation and times consuming and also record are not maintained property.

Thus current system is a kind of boring.

**6.3 Requirement Specification**

Requirements of client describe as follow:

* System should provide secure and accurate data.
* System should be easy to operate.
* System should provide information related to firm.

1. Admin :-

* The admin is the super user of the website so there are many requirements includes by administrator. It includes following things:
* Login :
* Admin have its own username and password for login into the administrator block.
* After entering with correct username and password.
* Admin perform following tasks related to website.
* Admin manage the sub – admin and other functionality.
* Add/Delete/Update/View Categories :
* Admin can add categories in the front panel.
* Admin can update the information in the category.
* Admin can Delete the categories
* Admin can View the categories.
* Add/ Delete/Update/View Subcategories :
* Admin can add subcategories in the front panel.
* Admin can update the information in the subcategory.
* Admin can Delete the subcategories
* Admin can View the subcategories.
* Add/ Delete/Update/View Packages :
* Admin can Add Packages in the front panel.
* Admin can update the information in the category.
* Admin can delete the Packages.
* Admin can View the Packages.
* Deactivate/View Customers :
* Admin can deactivate the Customer.
* Admin can view the Customer.

1. User :-

* Users are registered users which have their own username and password.
* Login :
* User have its own username and password so using it user can enter in user panel from the front end side and then perform many things.
* User can view Categories.
* User can select the categories.
* They can find the services they want.
* User can know all services and special offers.
* They can view the enquiries.

**6.4 Proposed System**

**6.4.1 Introduction :**

“KarFind” enables you to find desired services nearby you. Your enquiry will depend on a lot of things.

Users have so many functionalities which can be performed very easily.

User can choose the categories that is provide the different types of service categories, then after they can choose the subcategories that provides the exact type of service they want.

This website allows managing entire website through one individual known as admin.

**6.4.2 Scope :**

The scope of the website “KarFind” is global that is, it should be able to be accessed from anywhere through internet and users must be able to directly access the website anytime and anywhere. The site serves services for finding services.

There is an admin for managing packages and other related functionalities of the Website.

The website allows admin for doing all the things related to admin panel i.e., admin panel for management. It manages packages, view details, check user details, and manage services.

The website allows users to make inquiries for different services, and can search for services according to their desire.

**6.5 Advantages of Proposed System**

We hope that this type of “KarFind” will yield to certain benefits to customers. We have tried our best to illustrate some of them.

The side must be user friendly and graphics must be used to give a striking look and feel of any services or item that the customer wants to purchase.

The side must be easy and flexible to use. User can easily find services nearby them.

Navigation must be smoother and user must be able to understand the links.

To attract the customer by promoting the glamour features and services and hence achieve customer satisfaction.

User can know all services and special offer also they known all events at KarFind.

User can see notification like schedule info. User can review to different vendors. Vendors can submit their business documents for verification.

**6.6 Dataflow Diagram**

A Data Flow Diagram is graphics representation that depicts information flow & transformation that are applied as data move from input to output. The basic form of DFD also known as Data Flow Graph or bubble chart.

Each function is considered as a process that consumers some input and produces.

The system is represented in terms data to system various processing carried out on this data and output data generated by system.

As information moves through software, it is modified by a series of transformation. A Data Flow Diagram is graphics representation that depicts information flow & transformation that are applied as data move from input to output.

The data flow diagram may be used to represent a system or software at any level of abstraction. In fact, DFDs may be partitioned into levels that represent increasing information flow and functional details. Therefore, the DFD provides a mechanism for functional modeling as well as information flow modeling.

**DFD Symbols:**

A DFD model uses a very limited number of primitive symbols to represent the functions performed by a system and data flow among this functions.

1. A Rectangle represents an external entity. It represent the source and destination of system data.

1. The circle represents a process transform incoming or outgoing data.

1. The Arrow head indicates the direction of data flow.

1. These symbols display the database Name which indicates the connected of the data flow.

* **Context Level DFD :**

**KarFind**

**ADMIN**

**USER**

**LOGIN**

**MANAGE USER**

**ADD PACKAGE**

**MANAGE CATEGORY**

**L**

**O**

**G**

**I**

**N**

**B**

**O**

**O**

**K**

**I**

**N**

**G**

* **First Level DFD :**

1. **Admin**

LOGIN

MANAG USER

CATEGORY

SUB

CATEGORY

PACKAGE

ADMIN

login request

login responds

manage user request

manage user response

category request

category response

user name & password

responds or error

user information

view user information

manage category

view category

mange subcategory

subcategory request

subcategory response

package request

package response

view subcategory

manage package

view package

1. **User**

LOGIN

SEARCH

SERVICE

USER

login request

login response

mange service

user name & password

responds error

view service request

* **Second Level DFD :**

1. **Login**

read

login

info

decrypt

password

verify

user

**check user name &**

**password**

**error/ successfully login**

1. **Managing Category**

manage

category

request

select

operation

view

category

add

category

update

category

delete

categoy

display

request of view category

detached data

add details

insert in table

update in table

update details

delete category

remove category in table

r

e

s

p

o

n

s

1. **Managing Package**

manage

package

request

select

operation

view

package

add

package

update

package

delete

package

display

view package

detached data

add package

insert in table

update in package

update in a table

delete package

remove fromin table

r

e

s

p

o

n

s

1. **Managing Subcategory**

manage

subcategory

request

select

operation

view

sub

category

add

sub

category

update

sub

category

delete

sub

categoy

display

request of view subcategory

detached data

add details

insert in table

update in table

update details

delete in table

delete subcategory

r

e

s

p

o

n

s

**6.7 Process Specification**

1. **login**

read login

information

in this process input is taken from the admin

as username and password . this process

verifies the username and password

from user table and allows the user to login

into system otherwise error message will

display

login request

login request

1. **User**

fetch

user

info

this process will fetch all user & can

approve,disapprove ,block,delete & admin

can block & delete

manage user

manage user

1. **Category**

fetch

category

request

this process will accept category request

and display multiple operation and also

provide to perform list

category request

category request

1. **Sub – Category**

fetch

subcategory

request

this process will accept category request

and display multiple operation and also

provide to perform list

subcategory request

subcategory request

* 1. **Data Dictionary**

A data dictionary lists all data items appearing in the DFD model of the system. The data items list includes all data flow and contents of all data store appearing on the DFD, in the DFD of a system.

The formats of dictionary varies from tool to tool, most contain the following information :-

* Name : primary name of data and control item, data store or an external

Entity.

* Alias : other name used for the first entry.
* Where used/How used : A listing of processes that used the data for control item and how it is used. Eg. Input to process, output from process as a store, as an external entity.
* Content description : A notation for representing content. The notation used to develop a content description is noted in the following table :

|  |  |  |
| --- | --- | --- |
| Data Construct | Notation | Meaning is composed of |
| Seequence | + | AND |
| Selection | [|] | Either or |
| Rpitition | { }n | N repetition of |
|  | () | Obtain data |
|  |  |  |

* Supplementary information : Other information about data types, present values (if known), restriction or limitations and so forth.

**6.9 Table Structure with Proper Relationship**

1. admin\_table

|  |  |  |  |
| --- | --- | --- | --- |
| Filed Name | Datatype/Size | Constraints | Description |
|  |  |  |  |
| admin\_Id | Int | Primary key | It contains ID of Admin |
| admin\_Name | Text |  | It contains value of Admin Name |
| admin\_Password | Text |  | It contains value of AdminPassword |
| img | Text |  | It contains value of profile |
| admin\_Contact | Varchar(20) |  | It contains value of PhoneNo |
| admin\_Address | Text |  | It contains value of Address |
| admin\_Email | Text |  | It contains value of Email |
| admin\_status | Enum(1,0) |  | It contains value of Status |
| isSuper | Enum(1,0) |  | It contains the information that describes whether the admin is super admin or not |

1. vendor\_table

|  |  |  |  |
| --- | --- | --- | --- |
| Filed Name | Datatype/Size | Constraints | Description |
|  |  |  |  |
| vendor\_id | Int | Primary key | It contains ID of Company |
| company\_name | Varchar(30) |  | It contains value of CompanyName |
| vendor\_name | Varchar(30) |  | It contains value of OwnerName |
| business\_description | text |  | It contains value of Description |
| company\_document | text |  | It contains value of Company Document |
| company\_address | text |  | It contains value of Address |
| company\_contact | Varchar(12) |  | It contains value of PhoneNo |
| vendor\_image | text |  | It contains value of Image |
| company\_email | Varchar(30) |  | It contains value of Email |
| vendor\_status | enum(1,0) |  | It contains value of Status |
| date\_of\_join | date |  | It contains value of CreateDate |
| verification | enum(1,0) |  | It contains value of business verification. i.e. 0 - not verified   1. - requested 2. - verified |

1. customer\_table

|  |  |  |  |
| --- | --- | --- | --- |
| Filed Name | Datatype/Size | Constraints | Description |
|  |  |  |  |
| customer\_Id | Int | Primary key | It contains ID of User |
| customer\_Name | Varchar(20) |  | It contains value of FullName |
| customer\_Password | text |  | It contains value of Password |
| customer\_Address | text |  | It contains value of Address |
| customer\_Contact | Varchar(11) |  | It contains value of PhoneNo |
| customer\_Email | Varchar(30) |  | It contains value of Email |
| area\_id | int | Foreign key | It contains value of AreaID |
| city\_id | int | Foreign key | It contains value of CityID |
| customer\_image | text |  | It contains value of ProfilePic |
| customer\_status | enum(1,0) |  | It contains value of Status |
| date\_of\_join | date |  | It contains value of CreateDate |

1. city\_table

|  |  |  |  |
| --- | --- | --- | --- |
| Filed Name | Datatype/Size | Constraints | Description |
|  |  |  |  |
| city\_id | int | Primary key | It contains ID of City |
| city\_name | Varchar(30) |  | It contains value of City |
| city\_status | enum(1,0) |  | It contains value of Status |

1. package\_table

|  |  |  |  |
| --- | --- | --- | --- |
| Filed Name | Datatype/Size | Constraints | Description |
|  |  |  |  |
| package\_id | Int | Primary key | It contains ID of Package |
| package\_name | Varchar(30) |  | It contains value of Package Name |
| service\_count | Int |  | It contains value of Service Count |
| package\_price | Int |  | It contains value of Price |
| package\_status | enum(1,0) |  | It contains value of Status |

1. faq\_table

|  |  |  |  |
| --- | --- | --- | --- |
| Filed Name | Datatype/Size | Constraints | Description |
|  |  |  |  |
| faq\_id | tinyint | Primary key | It contains ID of Question |
| faq\_question | text |  | It contains value of Question |
| faq\_answer | text |  | It contains value of Answer |
| faq\_Status | enum(1,0) |  | It contains value of Status |

1. area\_table

|  |  |  |  |
| --- | --- | --- | --- |
| Filed Name | Datatype/Size | Constraints | Description |
|  |  |  |  |
| area\_id | Int | Primary key | It contains ID of Area |
| city\_id | Int | Foreign key | It contains value of City |
| area\_name | Varchar(20) |  | It contains value of Area. |
| Zipcode | int |  | It contains value of Zipcode |
| area\_status | enum(1,0) |  | It contains value of Status |

1. category\_table

|  |  |  |  |
| --- | --- | --- | --- |
| Filed Name | Datatype/Size | Constraints | Description |
|  |  |  |  |
| cat\_Id | tinyint | Primary key | It contains ID of Category |
| cat\_Name | Varchar(20) |  | It contains value of Category |
| cat\_Description | text |  | It contains value of Description |
| cat\_status | enum(1,0) |  | It contains value of Status |

1. subcategory\_table

|  |  |  |  |
| --- | --- | --- | --- |
| Filed Name | Datatype/Size | Constraints | Description |
|  |  |  |  |
| sub\_cat\_Id | int | Primary key | It contains ID of Subcategory |
| sub\_cat\_Name | Varchar(30) |  | It contains value of Subcategory |
| cat\_id | tinyint | Foreign key | It contains value of Category ID |
| sub\_cat\_Desc | text |  | It contains value of Description |
| sub\_cat\_status | enum(1,0) |  | It contains value of Status |

1. complain\_table

|  |  |  |  |
| --- | --- | --- | --- |
| Filed Name | Datatype/Size | Constraints | Description |
|  |  |  |  |
| complain\_id | int | Primary key | It contains ID of Complain |
| customer\_id | int | Foreign key | It contains value of Customer ID |
| service\_request\_id | int | Foreign key | It contains value of Service Request ID |
| complain\_description | text |  | It contains value of Description |
| complain\_status | enum(1,0) |  | It contains value of Status |

1. order\_table

|  |  |  |  |
| --- | --- | --- | --- |
| Filed Name | Datatype/Size | Constraints | Description |
|  |  |  |  |
| order\_id | int | Primary key | It contains ID of Order |
| order\_Date | date |  | It contains value of Order Date |
| service\_request\_id | int | Foreign key | It contains value of Service Request ID |
| order\_time | time |  | It contains value of Order Time |
| vendor\_delivery\_Id | int | Foreign key | It contains value of Vendor Delivery ID |
| order\_status | enum(1,0) |  | It contains value of Status |

1. otp\_status

|  |  |  |  |
| --- | --- | --- | --- |
| Filed Name | Datatype/Size | Constraints | Description |
|  |  |  |  |
| id | int | Primary key | It contains ID of OTP |
| otp | int |  | It contains value of OTP |
| create\_date | date |  | It contains value of CreateDate |
| create\_time | time |  | It contains value of CreateTime |
| status | enum(1,0) |  | It contains value of Status |

1. package\_details\_table

|  |  |  |  |
| --- | --- | --- | --- |
| Filed Name | Datatype/Size | Constraints | Description |
|  |  |  |  |
| package\_detail\_id | int | Primary key | It contains ID of Package Detail |
| package\_id | int | Foreign key | It contains value of Package ID |
| vendor\_Id | int | Foreign key | It contains value of vendor ID |
| purchase\_date | date |  | It contains value of Purchase Date |
| remaining\_services | int |  | It contains value of Remaining Services |
| package\_detail\_status | enum(1,0) |  | It contains value of Status |

1. review\_rating\_table

|  |  |  |  |
| --- | --- | --- | --- |
| Filed Name | Datatype/Size | Constraints | Description |
|  |  |  |  |
| rating\_id | int | Primary key | It contains ID of Rating |
| customer\_Id | int | Foreign key | It contains value of Customer ID |
| vendor\_id | int | Foreign key | It contains value of Vendor ID |
| service\_id | Int | Foreign key | It contains value of Service ID |
| rating\_description | text |  | It contains value of Rating Description |
| rating | Varchar(5) |  | It contains value of Rating |
| rate\_date | date |  | It contains value of Rating Date |
| rating\_status | enum(1,0) |  | It contains value of Status |

1. service\_request\_table

|  |  |  |  |
| --- | --- | --- | --- |
| Filed Name | Datatype/Size | Constraints | Description |
|  |  |  |  |
| service\_request\_id | int | Primary key | It contains ID of Service Request |
| customer\_id | int | Foreign key | It contains value of Customer ID |
| service\_id | int | Foreign key | It contains value of Service ID |
| request\_description | text |  | It contains value of Request Description |
| service\_time | time |  | It contains value of Service Time |
| service\_date | date |  | It contains value of Service Date |
| service\_request\_status | enum(1,0) |  | It contains value of Status |

1. service\_table

|  |  |  |  |
| --- | --- | --- | --- |
| Filed Name | Datatype/Size | Constraints | Description |
|  |  |  |  |
| service\_id | int | Primary key | It contains ID of Service |
| service\_name | Varchar(20) |  | It contains value of Service Name |
| sub\_cat\_id | int | Foreign key | It contains value of Subcategory ID |
| vendor\_id | int | Foreign key | It contains value of vendor ID |
| service\_description | text |  | It contains value of Service Description |
| price | int |  | It contains value of Service Price |
| icon | text |  | It contains value of Icon |
| service\_Status | enum(1,0) |  | It contains value of Status |

1. testimonial\_table

|  |  |  |  |
| --- | --- | --- | --- |
| Filed Name | Datatype/Size | Constraints | Description |
|  |  |  |  |
| testimonial\_id | tinyint | Primary key | It contains ID of Testimonial |
| customer\_id | Int | Foreign key | It contains value of Customer ID |
| testimonial\_message | Text |  | It contains value of Testimonial Message |
| testimonial\_status | Enum(1,0) |  | It contains value of Status |

1. vendor\_delivery\_table

|  |  |  |  |
| --- | --- | --- | --- |
| Filed Name | Datatype/Size | Constraints | Description |
|  |  |  |  |
| vendor\_delivery\_Id | int | Primary key | It contains ID of Vendor Delivery |
| vendor\_id | Int | Foreign key | It contains value of Vendor ID |
| customer\_id | int | Foreign key | It contains value of Customer ID |
| delivery\_name | text |  | It contains value of Delivery Name |
| delivery\_contact | Varchar(11) |  | It contains value of Delivery Contact |
| delivery\_status | Enum(1,0) |  | It contains value of Status |

1. admin\_otp\_table

|  |  |  |  |
| --- | --- | --- | --- |
| Filed Name | Datatype/Size | Constraints | Description |
|  |  |  |  |
| id | int | Primary key | It contains ID of OTP |
| mail | Varchar(30) | Foreign key | It contains value of Admin Email ID |
| otp | int |  | It contains value of OTP |
| create\_time | int |  | It contains value of CreateTime |
| status | tinyint(1) |  | It contains value of Status |

7. Design Report

* 1. **Site Diagram**

KarFind

admin

user

login

manage

category

manage

sub

category

manage

user

manage

package

view

request

login

view

category

view

sub

category

**7.2 Flow Chart**

start

admin login

input username

& password

is

username

& password

valid ?

admin

dashbord

prompt message

invalid username

& password

logout

stop

invalid

valid

* 1. **Security Issues**

**1) Validation of input and output data:**

All data used by the website (form user, other servers, other websites and internal system) must be validated for type (eg. Numeric, date, string), length (eg. 200 character maximum, or a positive integer) and syntax and business rules all data written as output (displayed) need to be safe to view in a browser, email client or other software and integrity of any data that is returned must be checked. Utilizing asynchronous JavaScript and AJAX control increase complexity and the possible attack vectors.

**2) Direct data access:**

If data exists, it can potentially be viewed or extracted. Avoid storing data that you do not need on the website and its database. Poorly developed system may allow access to data through insufficient input and output data validation or poor system security.

**3) Data poisoning:**

If user’s can delete data inappropriately and this then used to update your internal systems, business information is being lost. This can be hard to detect and it is important to ensure poisoning is not occurring. If poisoning is not detected until well after it has occurred, it may be impossible to recover the original data.

**4) Authentication and session management:**

Websites rely on identifying user to provide access permissions to data and functions. If authentication (verification of identity, registration and logging in), authorization (grating access rights) and session management can be altered, a user could access resources they are not allowed to. Beware especially of how password remainders, change password, log out and updating account details are handled, how session tokens are used and always have login forms on dedicated pages.

**5) System architecture and configuration:**

The information system architecture model should address the sensitivity of data identified during the requirement and specification phase of a website project. This may entail having separate web, application and database servers, load balancing or virtualization. Additional security issues and be created through the way the live environment is configured. Sufficient and safe logging, monitoring and alerting facilities need to be built in to allow audit.

**6) Phishing:**

Phishing, where users are conned into believing some other entity is or belongs to your own organization (email messages and websites are the most common combination), is best tackled through user education but the way the website is designed, its architecture and how it communicated with users can reduce the risk.

**7) System information leakages:**

Web servers, errors, staff, partner organizations, search engines and rubbish can all be the source of important information about your website – its technologies, business logic and security methods. An attacker can use such information to their advantages so it is important to avoid system information leakage as far as possible.

**8) Error handling:**

Exceptions such as user data validation messages, missing pages and server errors should be handled by the code so that a customer page is display that do not provide any system information to the user. Logging and alerting of unusual conditions should be enabled and these should allow subsequent audit.

* 1. **Input Screen Layout**

1. **Login**
2. **Admin view**

**3.Admin**

**4. Home**

**5.Category**

**6.Sub – Category**

**7.Package**

**8.Enquiry**

**9.View Advertisement**

**10.View Contact us**

**8. Testing Reports**

**8.1 Testing Issues**

**8.2 Test Case**

**8.1 Testing Issues**

* ADMIN/COMPANY/CLIENT LOGIN MODULE:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Case No. | Input | Expected Output | Actual Output | Test Result |
| 1. | Email=”harshilshah@gmail.com”  Password=”harshil” | Login is successful | User Name is invalid. | Fail |
| 2. | Email=”harshilshah@gmail.com”  Password=”harshil” | Login is successful | Password is invalid. | Fail |
| 3. | Email=”harshilshah@gmail.com”  Password=”harshil10” | Password is invalid. | Password is invalid. | Pass |
| 4. | Email=”harshilshah@gmail.com”  Password=”harshil” | Login is successful | Login is successful | Pass |

**COMPANY REGISTRATION MODULE:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Case No. | Input | Expected Output | Actual Output | Test Result |
| 1. | Company Name=”Modi”  Owner Name=”Harshil”  Password=”harshil”  Confirm Password=”harshil”  Email=”Modishah@gmail.com”  Phone no=”7878451245”  About us=”Drive with confidence”  Address=”Adajan”  City=”Surat”  State=”Gujarat”  Company Logo=”Image.jpg”  Proof1=”Image1.jpg”  Proof2=”Image2.jpg” | Register successfully | Register successfully | Pass |
| 2. | Company Name=”1234”  Owner Name=”1234”  Password=”harshil” or “123”  Confirm Password=”not match with  Password”  Email=”harshilshah@gmailcom” or  ”harshilshahgmail.com”  Phone no=”harshil”  About us=””  Address=””  City=””  State=””  Company Logo=””  Proof1=””  Proof2=”” | Textbox border should be changed to red.  Textbox border should be changed to red.  Textbox border should be changed to red.  Textbox border should be changed to red.  Textbox border should be changed to red.  Textbox border should be changed to red.  Textbox border should be changed to red.  Textbox border should be changed to red.  Textbox border should be changed to red.  Textbox border should be changed to red.  Textbox border should be changed to red.  Textbox border should be changed to red.  Textbox border should be changed to red. | Textbox border should be changed to green.  Textbox border should be changed to green.  Textbox border should be changed to green.  Textbox border should be changed to green.  Textbox border should be changed to green.  Textbox border should be changed to green.  Textbox border should be changed to green.  Textbox border should be changed to green.  Textbox border should be changed to green.  Textbox border should be changed to green.  Textbox border should be changed to green.  Textbox border should be changed to green.  Textbox border should be changed to green. | Fail  Fail  Fail  Fail  Fail  Fail  Fail  Fail  Fail  Fail  Fail  Fail  Fail  Fail |
|  | Company Name=”1234”  Owner Name=”1234”  Password=”harshil” or “123”  Confirm Password=”not match with  Password”  Email=”harshilshah@gmailcom” or  ”harshilshahgmail.com”  Phone no=”harshil”  About us=””  Address=””  City=””  State=””  Company Logo=””  Proof1=””  Proof2=”” | Textbox border should be changed to red.  Textbox border should be changed to red.  Textbox border should be changed to red.  Textbox border should be changed to red.  Textbox border should be changed to red.  Textbox border should be changed to red.  Textbox border should be changed to red.  Textbox border should be changed to red.  Textbox border should be changed to red.  Textbox border should be changed to red.  Textbox border should be changed to red.  Textbox border should be changed to red.  Textbox border should be changed to red. | Textbox border should be changed to red.  Textbox border should be changed to red.  Textbox border should be changed to red.  Textbox border should be changed to red.  Textbox border should be changed to red.  Textbox border should be changed to red.  Textbox border should be changed to red.  Textbox border should be changed to red.  Textbox border should be changed to red.  Textbox border should be changed to red.  Textbox border should be changed to red.  Textbox border should be changed to red.  Textbox border should be changed to red. | Pass  Pass  Pass  Pass  Pass  Pass  Pass  Pass  Pass  Pass  Pass  Pass  Pass |

**User Registration Module**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Case No. | Input | Expected Output | Actual Output | Test Result |
| 1. | Full Name=”Harshil Shah”  User Name=”Harshil”  Password=”harshil”  Confirm Password=”harshil”  Email=”harshilshah@gmail.com”  Phone no=”7845781245”  Address=”vesu”  State=”Gujarat”  City=”Surat”  Gender=”male”  Profile Pic=”Image1.jpg” | Register successfully | Register successfully | Pass |
| 2. | Full Name=”1234”  User Name=””  Password=”harshil” or “123”  Confirm Password=”not match with  Password”  Email==”harshilshah@gmailcom” or  ”harshilshahgmail.com”  Phone no=”1405hk”  Address=””  State=””  City=””  Gender=””  Profile Pic=”” | Textbox border should be changed to red.  Textbox border should be changed to red.  Textbox border should be changed to red.  Textbox border should be changed to red.  Textbox border should be changed to red.  Textbox border should be changed to red.  Textbox border should be changed to red.  Textbox border should be changed to red.  Textbox border should be changed to red.  Textbox border should be changed to red.  Textbox border should be changed to red. | Textbox border should be changed to green.  Textbox border should be changed to green.  Textbox border should be changed to green.  Textbox border should be changed to green.  Textbox border should be changed to green.  Textbox border should be changed to green.  Textbox border should be changed to green.  Textbox border should be changed to green.  Textbox border should be changed to green.  Textbox border should be changed to green.  Textbox border should be changed to green. | Fail  Fail  Fail  Fail  Fail  Fail  Fail  Fail  Fail  Fail  Fail  Fail |
|  | Full Name=”1234”  User Name=””  Password=”harshil” or “123”  Confirm Password=”not match with  Password”  Email==”harshilshah@gmailcom” or  ”harshilshahgmail.com”  Phone no=”1405hk”  Address=””  State=””  City=””  Gender=””  Profile Pic=”” | Textbox border should be changed to red.  Textbox border should be changed to red.  Textbox border should be changed to red.  Textbox border should be changed to red.  Textbox border should be changed to red.  Textbox border should be changed to red.  Textbox border should be changed to red.  Textbox border should be changed to red.  Textbox border should be changed to red.  Textbox border should be changed to red.  Textbox border should be changed to red. | Textbox border should be changed to red.  Textbox border should be changed to red.  Textbox border should be changed to red.  Textbox border should be changed to red.  Textbox border should be changed to red.  Textbox border should be changed to red.  Textbox border should be changed to red.  Textbox border should be changed to red.  Textbox border should be changed to red.  Textbox border should be changed to red.  Textbox border should be changed to red. | Pass  Pass  Pass  Pass  Pass  Pass  Pass  Pass  Pass  Pass  Pass |

**8.2 Test Case**

**1) Test Case Selection:-**

* The Presented system **“KarFind”** is tested using white-box testing which is based on the internal structure of the system, may also called as testing to code, path-oriented testing, or logical-driven testing. A common goal of white-box testing to have a test case to exercise every path through a program. This may become exhaustive.

**2) White Box Testing Case Selection:-**

* Because exhaustive white-box testing is infeasible, test or code coverage metrics are used to select test cases, which measure the fraction of code exercised by test cases. Test cases are selected to achieve target test coverage levels.

**3) Test Coverage Metrics:-**

**3.1) Statement Coverage:-**

* The fraction of statements executed at least once is running a collection of test cases.

**3.2) Branch Coverage:-**

* The fraction of total branch directions that have been taken at least once in running a collection of test cases,

**3.3) Path Coverage:-**

* The fraction of paths taken at least once in running a collection of test case, where a “path” is defined in some way to limit the totals of feasible collection.

**4) Unit Testing:-**

* The presented system is tested using unit testing. Unit testing is test of code written by a single programmer. A unit is a portion of a system implemented by a single programmer. Unit testing is exercising a unit in isolation from the rest of the system. The system then is tested using integrating Testing.

**5) Integration Testing:-**

**5.1 Top-Down:-**

* Combine, test and debug top-leveled routines that become the integration test that harms for lower-level units.

**5.2 Bottom-Up:-**

* Combine, test low level routines into progressively larger modules and subsystems.

**5.3 Sandwich:-**

* Mainly top-down with bottom-up integration and testing applied to certain widely used components.

**9. Limitations of the System**

* The vendor is unable to view packages without getting registered.
* The company isn’t able to access system till it doesn’t get approved from admin.
* Admin isn’t able to view package wise sells or packages sold vendor wise.

**10. Future Enhancements of the Project**

* We will enhance our business in more cities and provide batter facilities.
* Provide facility of choosing popular vendor of the area.
* Soon we will host this site.

**11. References**

* [**www.w3schools.com**](http://www.w3schools.com)
* [**www.Drivewithconfidence.com**](http://www.Drivewithconfidence.com)
* [**www.stackoverflow.com**](http://www.stackoverflow.com)