Table of Contents

**[Chapter 1 Introduction](#_30j0zll) 4**

**[1.1 Introduction to the Project](#_1fob9te) 5**

**[1.2 Organization profile](#_3znysh7) 5**

**[1.3 Objectives of the project](#_2et92p0) 6**

**[Chapter 2 System Analysis](#_3dy6vkm) 7**

**[2.1 Existing System](#_1t3h5sf) 8**

**[2.1.1 Methods adopted for conducting the study](#_4d34og8) 8**

**[2.1.2 Basic functionalities of the existing system](#_2s8eyo1) 8**

**[2.1.3 Users involved with the existing system](#_17dp8vu) 8**

**[2.1.4 Pitfalls identified in the existing system](#_3rdcrjn) 8**

**[2.1.5 Feasibility study of implementing a new system](#_26in1rg) 9**

**[2.2 Proposed system](#_lnxbz9) 9**

**[2.2.1 Functional modules](#_35nkun2) 9**

**[2.2.2 Users/roles](#_1ksv4uv) 10**

**[2.2.3 Software lifecycle technique](#_44sinio) 10**

**[2.2.4 Plan of implementation](#_2jxsxqh) 10**

**[2.2.5 Comparison with the existing system](#_z337ya) 11**

**[Chapter 3 System Design](#_1y810tw) 12**

**[3.1 Modeling technique adopted for the development of system](#_4i7ojhp) 13**

**[3.2 UML](#_2xcytpi) 13**

**[3.2.1 Use Case Diagram](#_1ci93xb) 14**

**[3.2.2 Class Diagram](#_3whwml4) 15**

**[3.2.3 Sequence Diagram](#_2bn6wsx) 16**

**[3.2.4 Activity Diagram](#_qsh70q) 17**

**[3.3 ER Diagram](#_3as4poj) 18**

**[3.4 Database Tables list](#_1pxezwc) 19**

**[3.5 Software Tools](#_49x2ik5) 22**

**[3.6 Basic hardware requirements](#_2p2csry) 22**

**[3.7 Sample forms/web pages designed](#_147n2zr) 23**

**[3.8 Navigation chart of the entire system.](#_3o7alnk) 25**

**[3.9 Reports which can be generated from the system](#_23ckvvd) 26**

**[Chapter 4 Testing](#_32hioqz) 27**

**[4.1 Unit Testing](#_1hmsyys) 28**

**[4.2 Integration Testing](#_41mghml) 31**

**[4.3 User Acceptance Testing](#_2grqrue) 31**

**[4.4 Testing Details](#_vx1227) 32**

**[Chapter 5 Software Product Details](#_1v1yuxt) 33**

**[5.1 Snapshots](#_4f1mdlm) 34**

**[Sample code](#_2u6wntf) 38**

**[Chapter 6 Summary and Future Enhancements](#_3tbugp1) 41**

**[6.1 Summary of the Report](#_28h4qwu) 42**

**[Future Enhancements](#_nmf14n) 42**

# List of Figures

**Fig 3.1 14**

**Fig 3.2 15**

**Fig 3.3 16**

**Fig 3.4 17**

**Fig 3.5 18**

**Fig 3.6 25**

# List of Tables

**Table 3.1 19**

**Table 3.2 19**

**Table 3.3 20**

**Table 3.4 20**

**Table 3.5 21**

**Table 4.1 28**

**Table 4.2 28**

**Table 4.3 29**

**Table 4.4 29**

**Table 4.5 30**

## **Chapter 1 introduction**

### **1.1 Introduction to the Project**

Digital technologies have advanced more rapidly than any innovation in our history – reaching around [50 per cent](https://www.un.org/en/pdfs/DigitalCooperation-report-for web.pdf) of the developing world’s population in only two decades and transforming societies. By enhancing connectivity, financial inclusion, access to trade and public services, technology can be a great equalizer. Digital dominance has made it necessary for more people to rely on the internet to look for the products and services that they need. Any online business that does not effectively use this platform risks losing new lucrative opportunities.

But not all small businesses can afford this online revolution; Here comes companies like Sharpener Technologies which takes up small ad-hock projects like these to support their clients in their online journey.

### **1.2 Organization profile**

Sharpener Technology Services Pvt Ltd is a leading Training and IT consultation Company located in Chennai, Tamil Nadu. It provides various services like IT consulting, Computer upgrades, Cloud computing, Data backup and recovery, QA and testing , Ui/Ux Design and much more.

It has a team of global tech leaders with good experience and diverse expertise backgrounds. Their team has accomplished significant levels of achievement in their life’s journey and worked for marquee clients. The team is a mix of people from varied backgrounds and having held senior leadership roles for large modern-day tech organizations such as Capgemini, Infosys, IBM, Accenture and ISG. They provide digital and data solutions

Being a technology services company for 5 years with 50+ strength, they are a 360-degree technology solution provider, right from product conceptualization to product maturity.Their development road map tracks progress and execution, with intuitive design practices to achieve their clients’ organizational goals and create products and services that users love. They have large client base and a dedicated division working on the customized projects and they are always ready to took up adhoc projects

### **1.3 Objectives of the project**

The major objectives of the project are as follows,

* Register new clients
* Manage employees
* Raise service requests

* Allocating work to employees
* Track work progress
* Track Support Request and their Progress
* Generate work reports

## **Chapter 2 System Analysis**

### **2.1 Existing System**

### **2.1.1 Methods adopted for conducting study**

1. Surveys: By conducting a small survey among the previous clients it is identified that the clients are very unsatisfied with the current website , the current website which the company hosts is a static website which can only be able to display certain details to the Users and there is no interaction between the clients and the employees in the current website , The clients has also pointed out that the frequent chats and video conferences with the technical team were very time consuming and a tedious process. They wished they had a system where they can communicate to them online and track the progress of the work.so in order to overcome the difficulties which are faced by the clients during the product development phase we are planning a better system
2. Interview: Interview of few clients where conducted to find the existing system working.After a long discussion with the chief technical officer , he also provided me some insights into the existing system. As they were following the agile methodology for development the documentation was very less and when ever the clients come back after sometime requesting some updates , bug fixes or for adding a new feature it was difficult for them to get back to that project quickly. So implementation of a new system which helps to overcome these problems is needed urgently to improve the overall business needs of the organization and improving the business productivity

### **2.1.2 Basic functionalities of existing system**

The current system works in a very traditional way, that is , the current website is a static website on which the clients can only view the portfolios and other documentations which are created by the admin , there is no client and employee interactions happening in the current system . the current website works in such a way that the clients contacts the admin through the contact details which are provided by the website and informs him about their business needs, then after discussing with his technical team he comes up with a project execution plan and informs the client about this through a meeting . After the initialization of the project there will be regular weekly or monthly meetings with the client to keep him updated and to get more info about how things needs to be.and there is not much interactions are happening through the client and the employees through the website

After the project is deployed , so many things can happen like, the team members may change , the technology used to develop may become obsolete or there may be some bugs or the client may require new feature etc. At this time it will be very difficult for the company to provide the technical support for the project. So in order to provide a better technical support system to the clients and to improve the quality of the employees in providing the various technical support to the clients a new system is needed to be implemented

### **2.1.3 Users involved with the existing system**

The existing system contains only the Admin , who manages everything including meeting new clients , servicing their requests , managing and tracking employees. There is no centralized data or work flow in the current system.

### **2.1.4 Pitfalls identified in the existing system**

* No direct link between the client and the employees
* There is no inbuilt platform to manage employees
* Wastage of time
* Poor client management

### **2.1.5 Feasibility study of implementing a new system**

The feasibility study of the proposed system has undergone several phases, given below.

* Cost: It requires only the cost to implement this Web Application since the existing System is more manual.
* Effort: The Proposed System is better than the existing, since it resolves all the demerits of the existing System. The proposed System which is reliable and flexible which makes the user experience and view better.
* Time : The Proposed System saves lot of time than the existing system .As the existing system a lot of money ,effort and time is wasted, this all is resolved by the proposed system
* Technical Feasibility: The Main focus of the proposed is the better user view and experience which in turn make the user feels better and easy to use the system. Since the proposed helps the Shoppe by collecting data points.
* Economic feasibility: The system works within the available recourse in the organization itself, so that it is economically feasible.

### **2.2 Proposed system**

### **2.2.1 Functional modules**

The functional modules of the project are as follows:

* Client Management Module:

This is the module which is responsible for handling and managing all the details of all the clients of the organization . this module works by storing all the essential details of the clients from their registration process in the website . once the client registers and logged on to the web application all the details about the client are stored on to the database permanently and all the employees and admin can have access to that data and can treat that particular client based on his/her personal information.

* Service Request Module:

This is the module which is responsible for the process of handling all the services which are provided by the organization. Once the client has logged on to the website he can make request for the various types of services which are provided by the organization, this process is handled in this module . once the client has made a service request to the organization , he/she can track the status of her requested services through the web application .the clients can communicate with the staffs and employees which are currently dealing with his/her requested service and the client has also the privilege to raise queries regarding their services.

* Employee Management Module:

This is the module which is used for managing and handling all the employees in the organization . whenever a new employee joins the organization all of his details are submitted to the administrator and it is the duty of the admin to create and manage the the user account for that employee. Only the admin has the privilege to add a new employee to the web application . once the admin has created the user account for that employee , he/she can logon to the website using his username and password which is given by the admin

when an employee leaves the organization the admin can remove the user account of that employee from the website but all of the details of the employee are still stored inside the databases

* Support Request Module :

This module is responsible for handling all the technical support which are provided by the organization after a software product is deployed. That is once a software product is deployed and later if there is some issue arises regarding the product and there are situations in which the technology changes or the employees which are assigned to provide the services resigns from the organization . in such situations the clients can logon to the website and they can choose their service from the website which is currently facing the issue and can raise a support request for that product . the support request is handled by a set of tokens called as tickets which are automatically created by the web application at the time of submitting a query.

The client can effectively track the status of the query which is raised by him through the ticket number which he gets at the time of submitting the support request

* Ticket generation module:

This is the module which is responsible for managing all the support requests in the organization . this module helps in creating the auto generating tokens known as tickets . each ticket has a unique ticket number which is used for tracking the query regarding a software product . once a client makes a support request regarding a software product then he will get the automatically generated ticket number and using this ticket number he can track the status of his query and can know weather his query is resolved or not.

When a client request for support and gets a unique ticket number , the admin will get a notification regarding the support and the ticket number . so when he replays to that ticket , then the status of that ticket gets automatically updated and when the issue associated with that ticket is resolved , the admin again the status of the ticket to closed , so that the client can able to know that his issue is resolved and he gets support in an effective manner

### **Users / roles**

There are mainly 3 users in this system, they are :

* Admin: The admin is the backbone of the entire system. admin can add new clients, employees, mange requests received ,assign tasks to each employees, forming teams within the employees and track work progress.
* Client: The client is a user which can request for services and support with his profile and can also track the progress of the services and support requests which are made by the client ..
* Employee: The employee are the users which are created by the admin . they updates the the daily tasks done by them to the dashboard and does the tasks assigned to them and reports to the the team lead after the completion of the works which are assigned to them.

### **Software Lifecycle Technique**

**Agile Model**

In this project agile model is used since agile development methodology and testing practices have worked wonders for numerous organizations with positive aspects. Its positive aspects are not hidden; it is very much visible in the organization. It also hold advantages like Customer satisfaction is rapid, continuous development and delivery of useful software, Product is developed fast and frequently delivered, A face-to-face conversation is the best form of communication. It also gives continuous attention to technical excellence and good design.

**Plan of implementation**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Task Name | Start  Date | Finish  Date | Dec ‘21 | Jan ‘22 | Feb ‘22 | Mar’22 |
| 1 | System study | 15/11/21 | 15/12/21 |  |  |  |  |
| 2 | Design phase | 16/12/21 | 01/02/22 |  |  |  |  |
| 3 | Coding | 02/02/22 | 01/03/22 |  |  | x | x |
| 4 | Testing | 02/03/22 | 07/03/22 |  |  |  |  |
| 5 | Implementation | 08/03/22 | 14/03/22 |  |  |  |  |

**Comparison with the existing system.**

|  |  |
| --- | --- |
| Proposed System | Existing System |
| * Not complex compared to the existing system. * Easy to use and maintain than the manual system. * Keeps entire records and details. * Gives an easy overview about the transaction details. * Maintains a history of transaction with history * Time, money, effort is not wasted. * Flexible reliable and user friendly system with better user interface. | * Very complex when Compared to the proposed System. * Since the existing system was manual it was difficult to maintain. * There was no particular order for records and there was a high chance to lost the track * There was no transaction history so it was difficult to get the overview. * Time, money and effort needed was huge. |

## **Chapter 3 System Design**

**3.1 Modeling technique adopted for the development of system**

Object-oriented modeling (OOM) is an approach to modeling an application that is used at the beginning of the [software life cycle](https://en.wikipedia.org/wiki/Software_development_process" \o "Software development process) when using an object-oriented approach to software development. The software life cycle is typically divided up into stages going from abstract descriptions of the problem to designs then to code and testing and finally to deployment. Modeling is done at the beginning of the process.

**3.2 UML Diagrams**

System design focuses on the final system and the process by which it is developed. It leads to transition from a user-oriented document which means system proposal a programmer-oriented document. The design methodology followed for this project is the bottom-up design strategy. In this approach, the basic sets of elements are individual modules. Each module is developed individually as a separate project and adds the modules into this as a reference if necessary. The benefit of the design is that it permits the review of the modules during the system development process. We prepare UML diagrams to understand a system in better and simple way. A single diagram is not enough to cover all aspects of the system. So UML defines various kinds of diagrams to cover most of the aspects of a system.

Unified modeling language (UML for short), is a standardized language accompanied by a specific set of symbols, used in the field of software engineering among others. UML diagrams make it possible to depict complex processes, systems and architectures in a way that anyone familiar with the unified modeling language will understand. They can be divided into two main categories; structure diagrams and behavioral diagrams. Some of the UML diagrams are:

1. **Use Case Diagram:**

The figure 3.2a which is shown below is the use case diagram for the web application . Here there are mainly 3 actors and they are admin , employee and the customer.

All of these 3 users have user login accounts to this website .Once the user tries to logon to the system he need to type the correctly , if the user do so , the user will be navigated to the user home inside the system else the user may be redirected to the error screen indicating a message that you have entered an incorrect username or password.

Once a client has logged on to the system , he can either make a request for a new service or can make request for the support for the existing system which will be managed by the admin or by the employees.



Fig 3.2.a

1. **Class Diagram:**

Here there are mainly 7 classes in this system they are employee class, department class, client class , service class , ticket class , support class and the payment class where the employee class contains all the details of the employees in the organization , the department class contains the number of employee which are belonging to a particular department , the service class contains the services which are provided by the organization and which are requested by the clients , the support class contains the details of the supports which are to be provided to those software products which are needed some support



Fig 3.2.b

1. **Sequence Diagram:**

Sequence diagram is the most common kind of interaction diagram, which focuses on the message interchange between a numbers of lifelines. Here there are mainly 3 sequences and they are the login sequence, service request sequence and the support request sequence . the login sequence diagram shows how a user is getting logged on the website , the service request sequence shows how a client is making a request to the organization for a particular service . for that the client need to register himself in the website and need to make a request and then need to make a payment .

The support request sequence shows how an existing user is making a request for the technical support for the software product , the client will get a unique token known as ticket from the website once he has made a support request , so with that token he can know the status of his support query

**Login Sequence**



**Service request sequence**



**Support Request Sequence**



1. **Activity Diagram:**

The basic purposes of activity diagrams are similar to other four diagrams. The activity diagram is used to show message flow from one activity to another.

The activity diagram is a flow chart like structure which represents the flow of control of the various users involved in the system , here there are mainly 3 users are involved in our website , they are the client , employee and the admin. The flow of control of these users are being clearly shown in the activity diagram shown below . here the client can make request for a service and he can make payment for that service and he can also make a support request for that software product whenever the client wants and all these processes are being carried out with the help of the employees in the organization who verifies the details and provides services to the requested clients . the admin is the super user in the website , the admin can control all the processes that are happening inside the web application



Fig 3.2.d

**E . ER Diagram**

An entity relationship diagram shows the relationships of entity sets stored in a database. An entity in this context is an object, Here there are mainly 4 entities available and they are the employee, service, client and the support .

The employee is the one who manages all the clients and the services . the clients can make a request for a service in the website and it is the duty of the employee to provide the service for that requesting employee , each services has their own support. It is also the duty of the employees to manage and provide the support for the clients which are requesting for the technical support . the figure 3.3 which is given below indicates the entity relationship diagram for this web application



Fig 3.3

**3.4 Database Design.**

* User: stores details about all the users registered with the organization

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Datatype** | **Size** | **Constraint** | **Description** |
| Usrid | Varchar | 10 | Primary key | User id |
| name | Varchar | 25 | Not null | User name |
| Email | Varchar | 25 | Not null | User email |
| Password | Varchar | 20 | Not null | User password |
| Utype | Varchar | 10 | Not null | User type |

Table 3.1

* Employee : contains the information about all the employees that are working with the organization

.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Datatype** | **Size** | **Constraint** | **Description** |
| Empid | Number | 10 | Primary key | Employee id |
| Fname | Varchar | 10 | Not null | First name |
| Lname | Varchar | 10 | Not null | Last name |
| phone | Number | 10 | Not null | phone number |
| Email | Varchar | 35 | Not null | Email id |
| Dob | Date | 10 | Not null | Date of birth |
| Gender | varchar | 6 | Not null | Gender |
| Adr1 | varchar | 25 | Not null | Temp\_address |
| Adr2 | varchar | 25 | Not null | Per\_address |
| City | varchar | 10 | Not null | City |
| Pcode | varchar | 10 | Not null | Postcode |
| Desig | varchar | 15 | Not null | Designation |
| Pass | varchar | 25 | Not null | password |

Table 3.2

* Event table: this table stores details about the events which are happening within the organization.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Datatype** | **Size** | **Constraint** | **Description** |
| Evid | Number | 10 | Primary key | Event id |
| Eventname | Varchar | 50 | Not null | Event name |
| Eventime | Varchar | 10 | Not null | Event time |
| Eventdesc | Varchar | 75 | Not null | Event description |
| Eventdate | Date | 10 | Not null | Event date |

Table 3.3

* Service table : contains information about the services which are requested by the clients .

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Datatype** | **Size** | **Constraint** | **Description** |
| Srid | Number | 10 | Primary key | Service if |
| Subj | Varchar | 100 | Not null | Subject |
| Stype | Varchar | 50 | Not null | Service type |
| Phone | Number | 10 | Not null | Contact number |
| Email | Varchar | 26 | Foreign key | User mail id |
| Descrip | Varchar | 100 | Not null | Service description |

Table 3.4

* Work : this table stores details about all the works which are happening in the organization .

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Datatype** | **Size** | **Constraint** | **Description** |
| Wid | Number | 10 | Primary key | Work id |
| Wstatus | Varchar | 25 | Not null | Work status |
| Srid | Number | 10 | Foreign key | Service id |
| Empid | Number | 10 | Foreign key | Employee id |
| Wdate | Date | 10 | Not null | Work Assign date |
| Pynmt | Number | 20 | Not null | payment |

Table 3.5

* Ticket : contains information about the supports which are requested by the clients and their status .

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Datatype** | **Size** | **Constraint** | **Description** |
| tid | Number | 10 | Primary key | Ticket id |
| Uname | Varchar | 25 | Not null | Client name |
| Usremail | Varchar | 25 | Foreign key | User email |
| Mobile | Number | 10 | Not null | phone number |
| Subject | Varchar | 50 | Not null | Subject |
| Section | Varchar | 30 | Not null | Department |
| Priority | Varchar | 10 | Not null | Priority |
| Service | Varchar | 25 | Not null | Related service |
| Message | Varchar | 50 | Not null | Message |
| Status | Varchar | 25 | Not null | status |

Table 3.6

**3.5 Software Tools**

**Visual Studio Code**

Visual Studio Code is a [source-code editor](https://en.wikipedia.org/wiki/Source-code_editor" \o "Source-code editor) made by [Microsoft](https://en.wikipedia.org/wiki/Microsoft" \o "Microsoft) for [Windows](https://en.wikipedia.org/wiki/Windows" \o "Windows), [Linux](https://en.wikipedia.org/wiki/Linux" \o "Linux) and [mac OS](https://en.wikipedia.org/wiki/MacOS" \o "MacOS). Features include support for [debugging](https://en.wikipedia.org/wiki/Debugging" \o "Debugging), [syntax highlighting](https://en.wikipedia.org/wiki/Syntax_highlighting" \o "Syntax highlighting), [intelligent code completion](https://en.wikipedia.org/wiki/Intelligent_code_completion" \o "Intelligent code completion), [snippets](https://en.wikipedia.org/wiki/Snippet_(programming)" \o "Snippet (programming)), [code refactoring](https://en.wikipedia.org/wiki/Code_refactoring" \o "Code refactoring), and embedded [Git](https://en.wikipedia.org/wiki/Git" \o "Git). Users can change the [theme](https://en.wikipedia.org/wiki/Theme_(computing)" \o "Theme (computing)), [keyboard shortcuts](https://en.wikipedia.org/wiki/Keyboard_shortcut" \o "Keyboard shortcut), preferences, and install [extensions](https://en.wikipedia.org/wiki/Plug-in_(computing)" \o "Plug-in (computing)) that add additional functionality. Instead of a project system, it allows users to open one or more directories, which can then be saved in workspaces for future reuse. This allows it to operate as a [language-agnostic](https://en.wikipedia.org/wiki/Language-agnostic" \o "Language-agnostic) code editor for any language. It supports a number of programming languages and a set of features that differs per language. Unwanted files and folders can be excluded from the project tree via the settings. Many Visual Studio Code features are not exposed through menus or the user interface but can be accessed via the command palette. Visual Studio Code can be extended via [extensions](https://en.wikipedia.org/wiki/Plug-in_(computing)" \o "Plug-in (computing)), available through a central repository. This includes additions to the editor and language support. A notable feature is the ability to create extensions that add support for new [languages](https://en.wikipedia.org/wiki/Programming_language" \o "Programming language), [themes](https://en.wikipedia.org/wiki/Theme_(computing)" \o "Theme (computing)), and [debuggers](https://en.wikipedia.org/wiki/Debugger" \o "Debugger), perform [static code analysis](https://en.wikipedia.org/wiki/Static_code_analysis" \o "Static code analysis), and add [code linters](https://en.wikipedia.org/wiki/Lint_(software)" \o "Lint (software)) using the [Language Server Protocol](https://en.wikipedia.org/wiki/Language_Server_Protocol" \o "Language Server Protocol).

**XAAMPP**

XAMPP  is a [free and open-source](https://en.wikipedia.org/wiki/Free_and_open-source" \o "Free and open-source) [cross-platform](https://en.wikipedia.org/wiki/Cross-platform" \o "Cross-platform) [web server](https://en.wikipedia.org/wiki/Web_server" \o "Web server) [solution stack](https://en.wikipedia.org/wiki/Solution_stack" \o "Solution stack) package developed by Apache Friends, consisting mainly of the [Apache HTTP Server](https://en.wikipedia.org/wiki/Apache_HTTP_Server" \o "Apache HTTP Server), [MariaDB](https://en.wikipedia.org/wiki/MariaDB" \o "MariaDB) [database](https://en.wikipedia.org/wiki/Database" \o "Database), and [interpreters](https://en.wikipedia.org/wiki/Interpreter_(computing)" \o "Interpreter (computing)) for scripts written in the [PHP](https://en.wikipedia.org/wiki/PHP" \o "PHP) and [Perl](https://en.wikipedia.org/wiki/Perl" \o "Perl) [programming languages](https://en.wikipedia.org/wiki/Programming_language" \o "Programming language). Since most actual web server deployments use the same components as XAMPP, it makes transitioning from a local test server to a live server possible.

**3.6 Basic Hardware Requirements**

**Development Configuration:**

• Machine (Minimum Requirement)

• Processor –Intel i3, AMD and aboveversion.

• RAM – 512MB andabove.

• Hardware Device – A Monitor and Keyboard with Mouse.

• Hard disk – Min 1GB

**Implementation Configuration:**

• Client Machine (Minimum Requirement)

• Hardware Device – A computer with a web browser.

• RAM – 512 MB and above (Recommended 1GB).

• Hard Disk – Min 60MB

## **Chapter 4: Testing Details**

### **4.1 Unit Testing**

Unit testing is a software development process in which the smallest testable parts of an application, called units, are individually and independently scrutinized for proper operation. This testing methodology is done during the development process by the software developers and sometimes QA staff.

The unit testing details for each module are as follows:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Tested** | **Test Condition** | **Expected Result** | **Actual Result** | **Status** |
| Name | Entered Non Characters | Alert Invalid Name | Alert Invalid Name | Pass |
| Phone Number | Entered More Than 10 Digits | Alert Invalid Phone Number | Alert Invalid Phone Number | Pass |
| Login | Invalid Username And Password | Alert Invalid Username And Password | Alert Invalid Username And Password | Pass |
| E-Mail | Entered Invalid E-Mail Format | Alert Invalid Email Format | Alert Invalid Email Format | Pass |
| Empty Value | Submitted Without Filling | Alert Field Required | Alert Field Required | Pass |
| duplicate email | Enter duplicate email id | Email already exists | Email already exists | Pass |

### **4.2 Integration Testing**

Integration Testing is defined as a type of testing where software modules are integrated logically and tested as a group. A typical software project consists of multiple software modules, coded by different programmers. The purpose of this level of testing is to expose defects in the interaction between these software modules when they are integrated.

The various test cases for integration testing performed are as follows:

* Login: properly forward to Account module on successful login.
* Account: The new customers can visit the website and create their own accounts and can register themself as a client of the organization .

### **4.3 User Acceptance Testing**

The “ **It support and ticket management system ”** was tested and found to be working as expected. There were no abnormal behaviours reported during the testing of the program. Testing is a method by which we try reducing the testing efforts and bringing out the maximum output. Testing helps us in knowing whether the logical assumptions that we have taken for the system are correct, and if they are correct, we have obtained our goal. We test the system to know the errors, to check the validity of the information, to also group the modules with the aim that we meet the system requirements according to the system needs.

### **4.4 Testing Details**

Implementation is the process of putting an action for the formulated plan. Before we implement, the plan should have been completed and our objectives should be clear. Testing each one of those actions formulated in the plan is said to be implementation testing. The system can be implemented only after thorough testing is done and only if it is found to work according to the specification. The implementation phase includes the training that should be provided for the chosen employees and admin for the better understanding of the system .

**3.7 Form Design**

**Login**

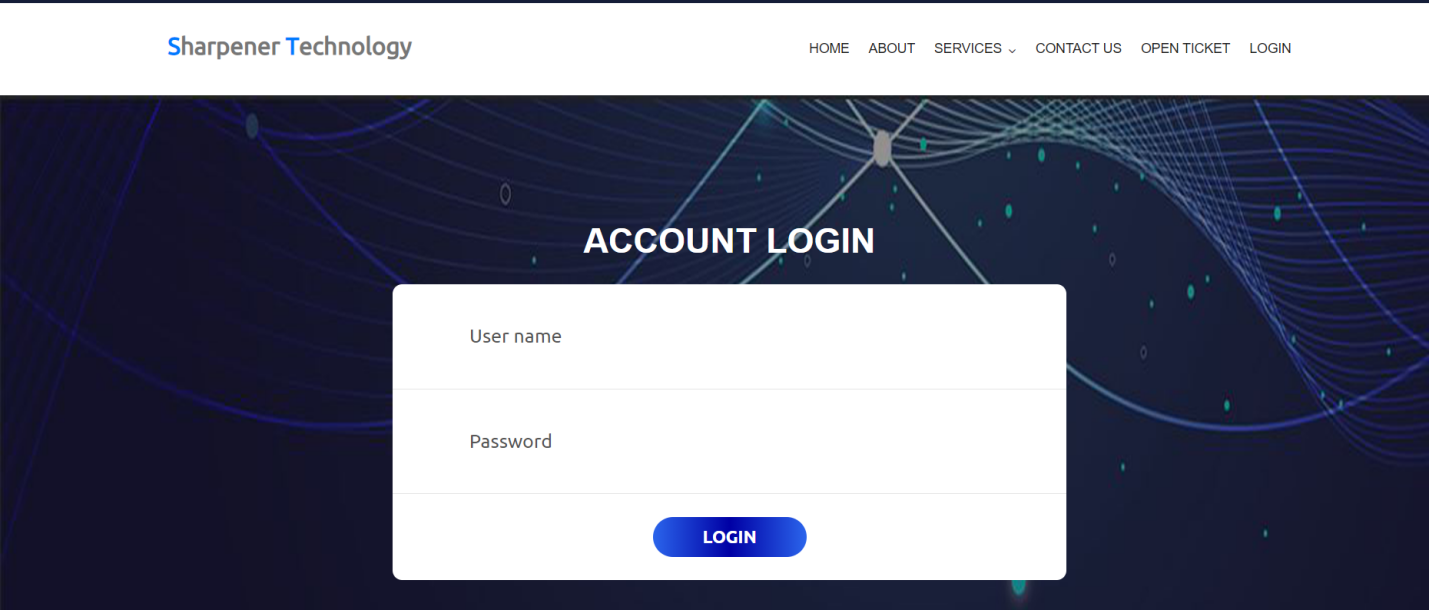
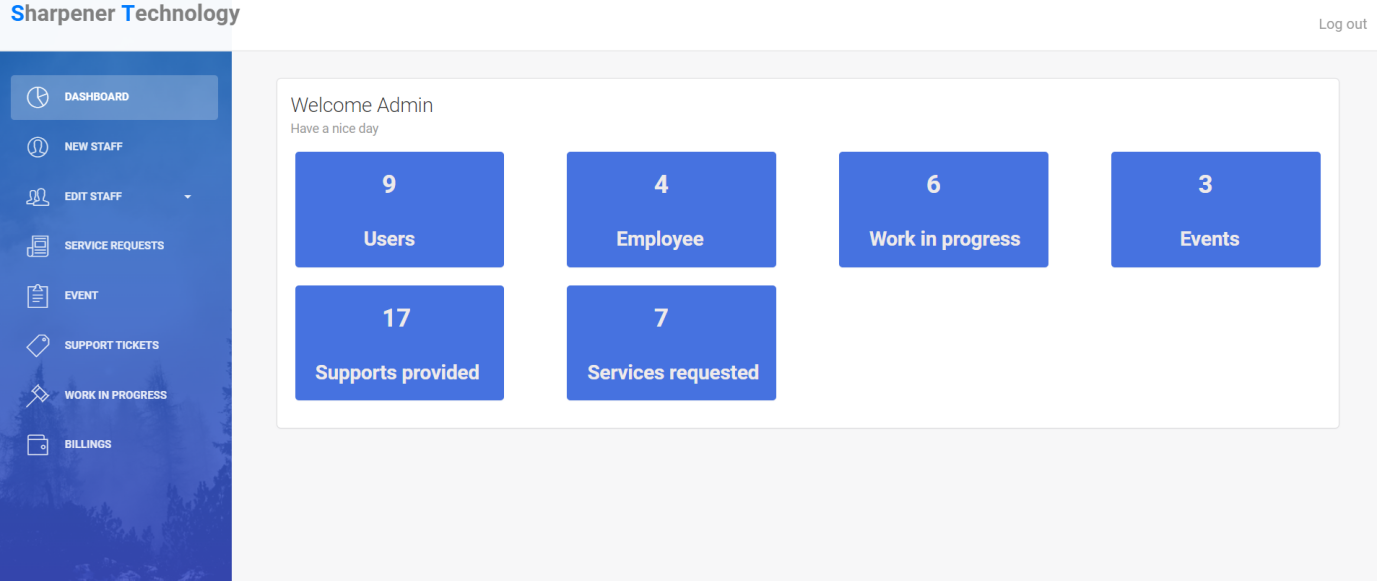
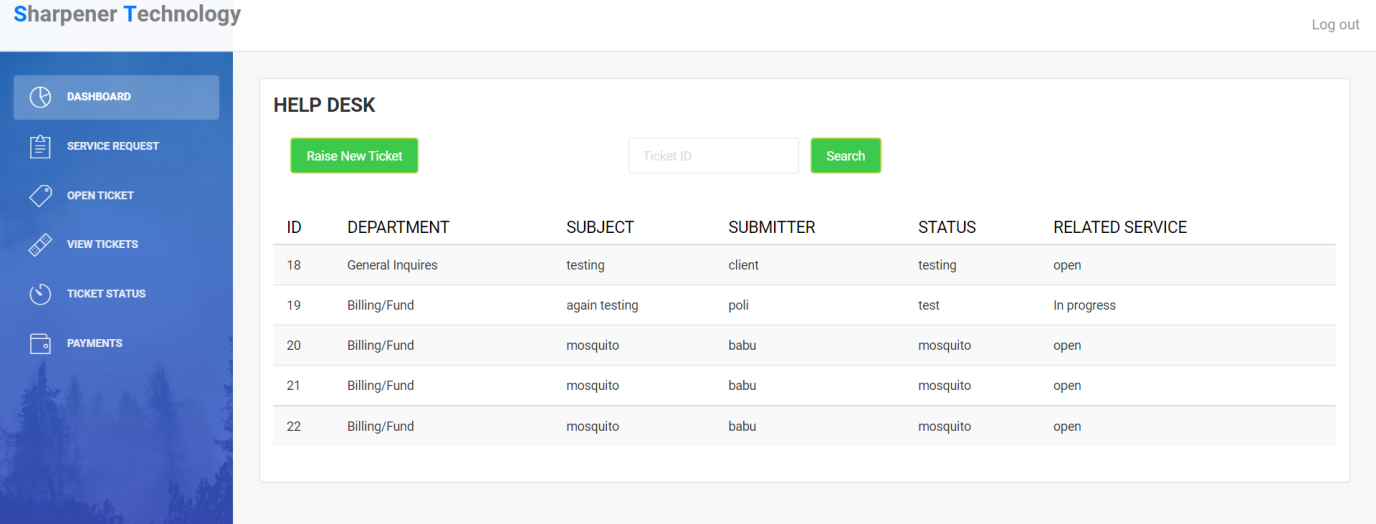


Fig 3.7 Login

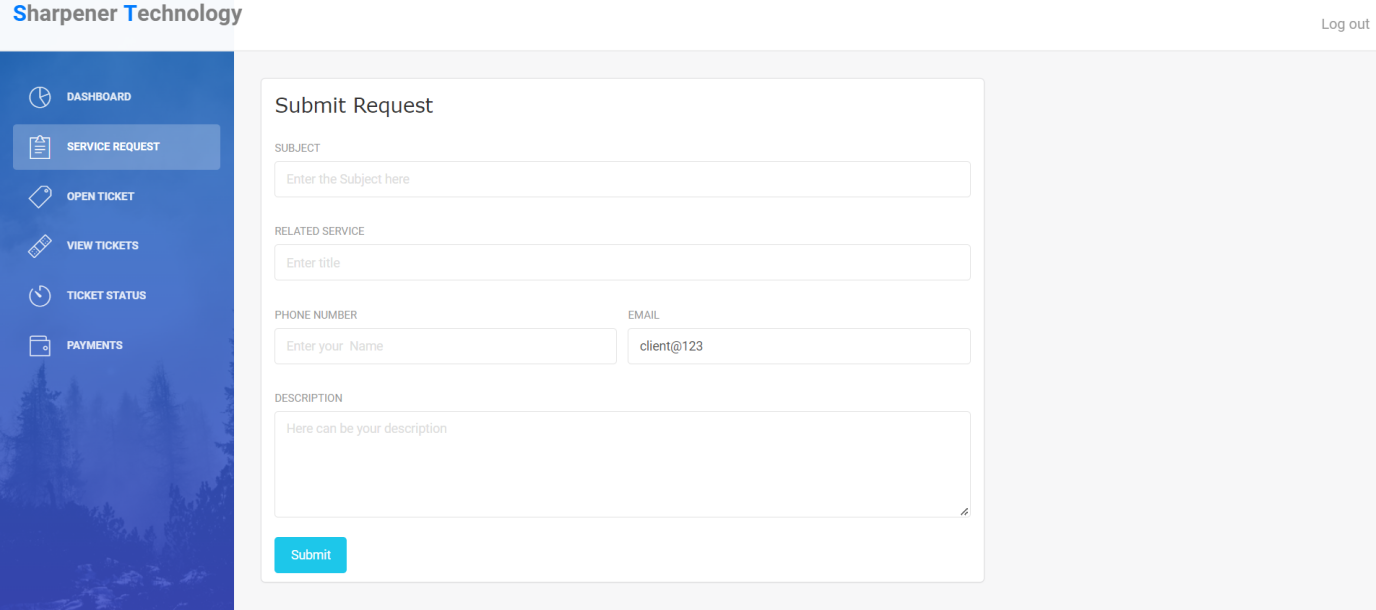
**Admin dashboard**



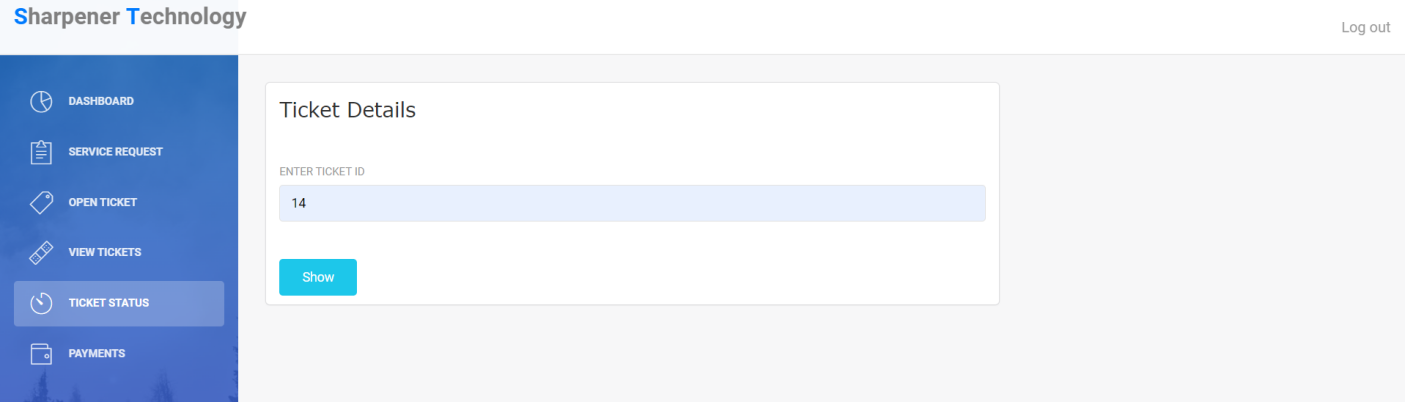
**Client dashboard**



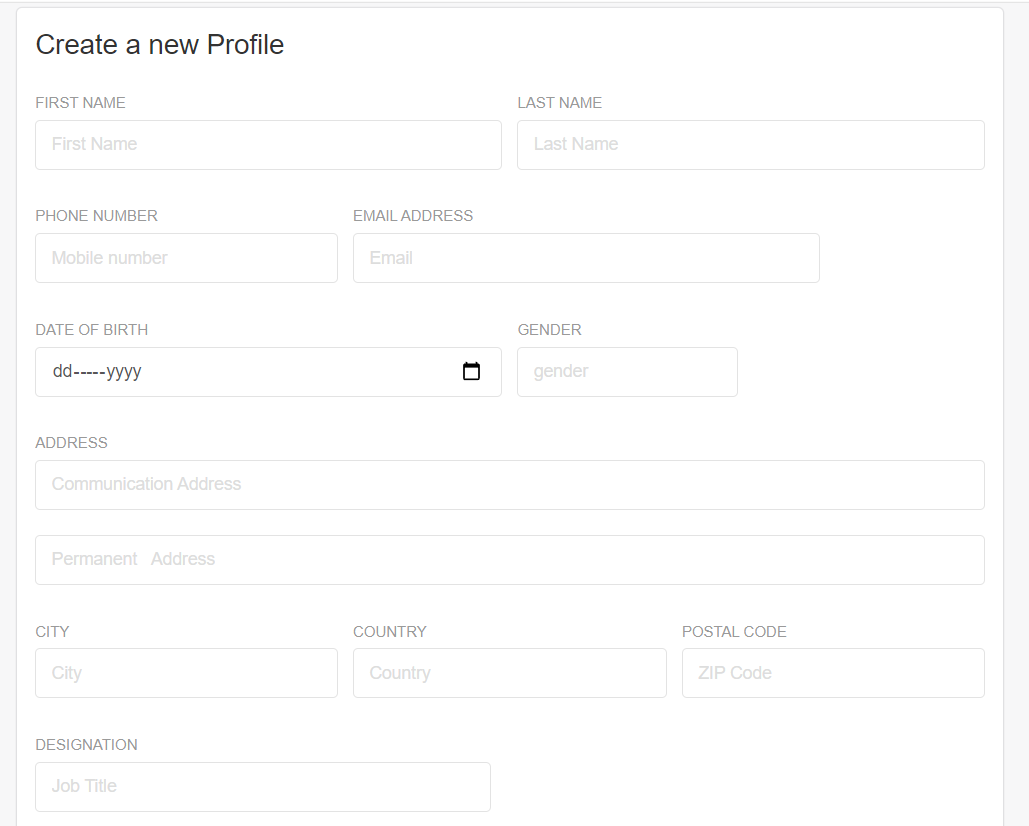
**Service request form**



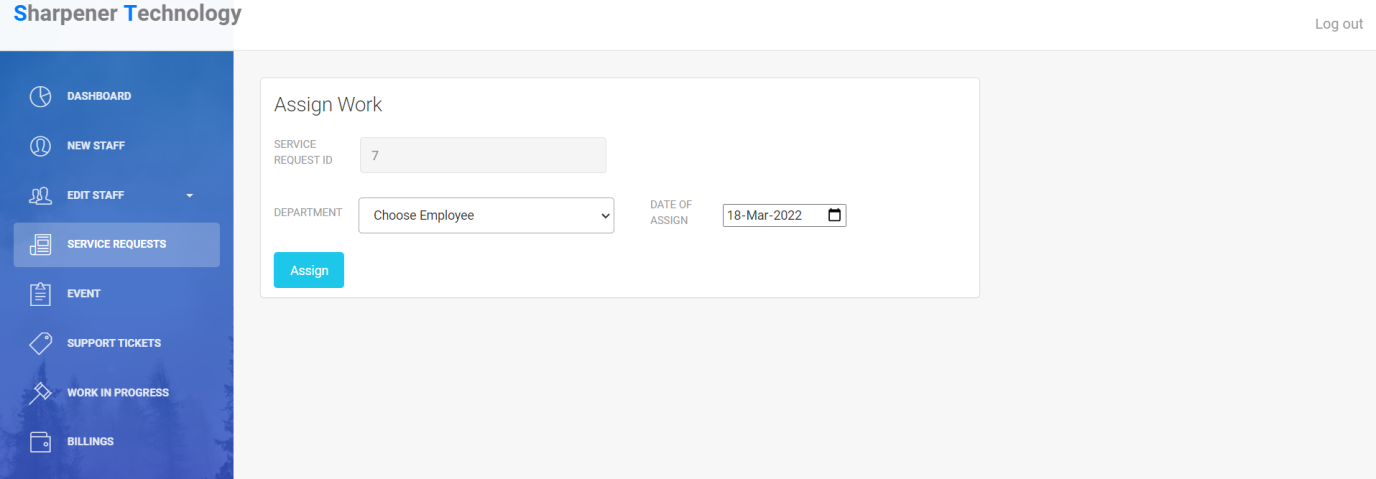
**TRACK STATUS**



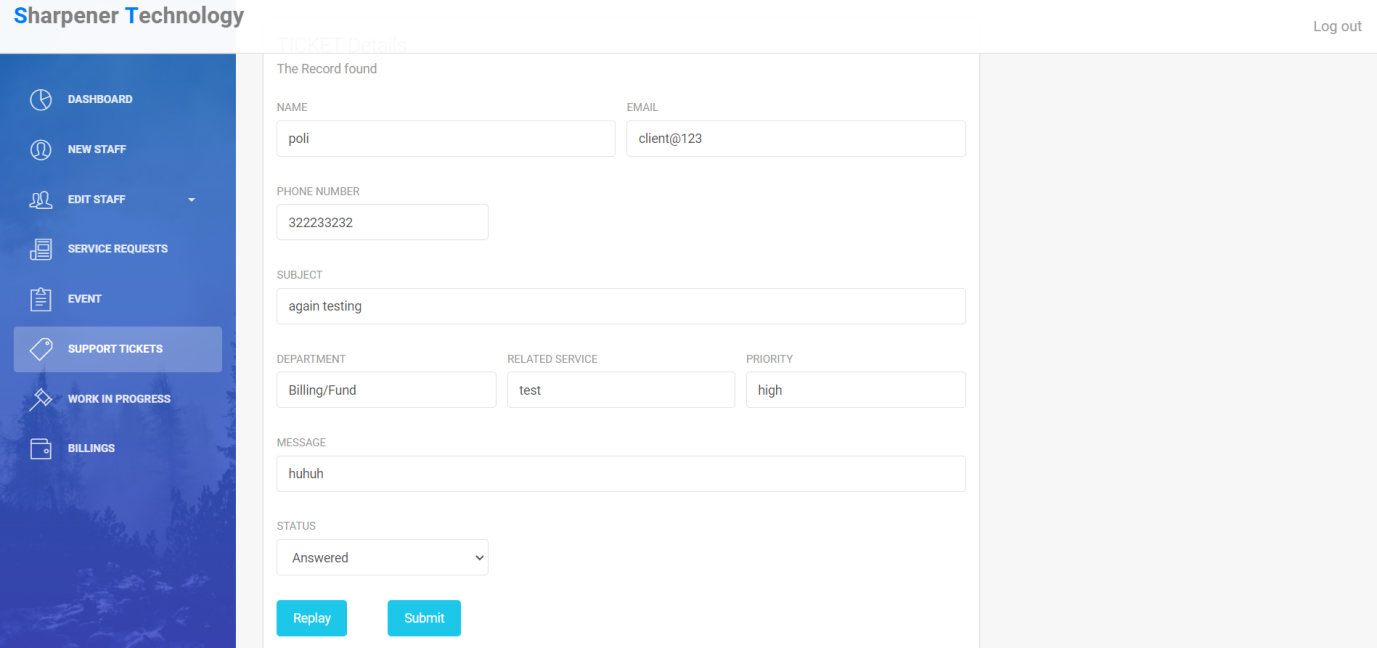
**EMPLOYEE RGISTRATION**



**Work Assigning form**



**Ticket Status updating form**



### **Sample code**

<?php

defined('BASEPATH') OR exit('No direct script access allowed');

class Welcome extends CI\_Controller {

public function index()

{

$this->load->view('home');

}

public function services()

{

$this->load->view('services');

}

public function about()

{

$this->load->view('about');

}

public function login()

{

$temp['us']="";

$this->load->view('Loginform',$temp);

}

public function contact()

{

$this->load->view('contact');

}

public function demo()

{

$this->load->view('demoview');

}

public function admindash()

{

$this->load->model('data\_model');

$info =$this->data\_model->get\_count1();

$info1 =$this->data\_model->get\_count2();

$info2 =$this->data\_model->get\_count3();

$info3 =$this->data\_model->get\_count4();

$info4 =$this->data\_model->get\_count5();

$info5 =$this->data\_model->get\_count6();

$data = array('planet' => $info,'planet1' => $info1,'planet2' => $info2,'planet3' => $info3,'planet4' => $info4,'planet5' => $info5);

$this->load->view('admindash',$data);

}

public function empdash()

{

$this->load->model('data\_model');

$info =$this->data\_model->get\_count1();

$info1 =$this->data\_model->get\_count2();

$info2 =$this->data\_model->get\_count3();

$info3 =$this->data\_model->get\_count4();

$info4 =$this->data\_model->get\_count5();

$info5 =$this->data\_model->get\_count6();

$data = array('planet' => $info,'planet1' => $info1,'planet2' => $info2,'planet3' => $info3,'planet4' => $info4,'planet5' => $info5);

$this->load->view('employedash',$data);

}

public function addstaff()

{

$this->load->view('adminaddstaff');

}

public function admineditform()

{

$this->load->view('adminstaffeditform');

}

public function admevent()

{

$this->load->view('adminevent');

}

public function cpayment()

{

$planet['payment'] = '0';

$usrmail=$this->session->userdata('user');//was id

$this->load->model('data\_model');

$usrdet=$this->data\_model->getsrid($usrmail);

$id=$usrdet->srid;

$data['track']=$this->data\_model->custpaymnt($id);

$this->load->view('custpayment',$data);

}

public function cpaychekout()

{

$id=$this->input->post('pymnt');

echo $id;

$planet['payment'] = $id;

$this->load->view('custpaymentform',$planet);

}

public function adminticket()

{

$this->load->model('data\_model');

$info['emp']=$this->data\_model->get\_data("ticket");

$this->load->view('adminticket',$info);

}

public function showtick()

{

$tid=$this->input->post('tid');

$this->load->model('data\_model');

$info['emp']=$this->data\_model->get\_Tick($tid);

// echo $tid;

if ( empty($info['emp']) )

{

$this->load->view('custsrnofound');

}

else

{

$this->load->view('custshotick',$info);

}

}

public function cushowtickstatus()

{

$tid=$this->input->post('tid');

$this->load->model('data\_model');

$info['emp']=$this->data\_model->get\_Tick($tid);

// echo $tid;

if ( empty($info['emp']) )

{

$this->load->view('custsrnofound');

}

else

{

$this->load->view('custticketstatusreslt',$info);

}

}

public function showtickreply()

{

$tid=$this->uri->segment(3);

$this->load->model('data\_model');

$info['emp']=$this->data\_model->get\_Tick($tid);

// echo $tid;

$this->load->view('admintikrep',$info);

}

public function empevent()

{

$this->load->model('data\_model');

$info['emp']=$this->data\_model->get\_data("evnttable");

$this->load->view('employeevent',$info);

}

public function del\_evnt() //delete

{

$id=$this->uri->segment(3);

$this->load->model('data\_model');

$this->data\_model->data\_delevnt("evnttable",$id);

header('location:'.base\_url().'Welcome/empevent');

}

public function emplist()

{

$this->load->model('data\_model');

$info['emp']=$this->data\_model->get\_data("employee");

$this->load->view('adminemplist',$info);

}

public function worklist()

{

$this->load->model('data\_model');

$info['emp']=$this->data\_model->work\_view();

$this->load->view('adminworklist',$info);

}

public function billing()

{

$this->load->model('data\_model');

$info['emp']=$this->data\_model->work\_view();

$this->load->view('adminbilling',$info);

}

public function calc\_bill() //delete

{

$srid=$this->uri->segment(3);

$this->load->model('data\_model');

$info['emp']=$this->data\_model->get\_wrk($srid);

$this->load->view('adminbillamount',$info);

}

public function amountupdate()

{

$wid=$this->input->post('wid');

//$id=$this->uri->segment(3);

$data=array('wdate'=>$this->input->post('wdate'),

'wstatus'=>$this->input->post('wstatus'),

'empid'=>$this->input->post('empid'),

'srid'=>$this->input->post('srid'),

'pymnt'=>$this->input->post('pymnt')

);

$this->load->model('data\_model');

$this->data\_model->amount\_update("work",$data,$wid);

$this->load->model('data\_model');

$info['emp']=$this->data\_model->work\_view();

$this->load->view('adminbilling',$info);

// header('location:'.base\_url().'Login/admineditemp');

}

public function empworklist()

{

$this->load->model('data\_model');

$info['emp']=$this->data\_model->work\_view();

$this->load->view('empworklist',$info);

}

public function empmywork()

{

$this->load->model('data\_model');

$emil=$this->session->userdata('employemail');

//echo $emil;

$emidt=$this->data\_model->getempid($emil);

$emid=$emidt->empid;

//echo $emid;

$info['emp']=$this->data\_model->mywork\_view($emid);

$this->load->view('empmywork',$info);

}

public function empworkupdat()

{

$srid=$this->uri->segment(3);

$this->load->model('data\_model');

$info['emp']=$this->data\_model->get\_wrk($srid);

// echo $tid;

$this->load->view('empmyworupd',$info);

}

public function empwrkstsupdate()

{

$wid=$this->input->post('wid');

//$id=$this->uri->segment(3);

$data=array('wdate'=>$this->input->post('adate'),

'wstatus'=>$this->input->post('status'),

'empid'=>$this->input->post('empid'),

'srid'=>$this->input->post('srid'),

'pymnt'=>$this->input->post('pymnt')

);

$this->load->model('data\_model');

$this->data\_model->amount\_update("work",$data,$wid);

$this->load->model('data\_model');

$emil=$this->session->userdata('employemail');

//echo $emil;

$emidt=$this->data\_model->getempid($emil);

$emid=$emidt->empid;

//echo $emid;

$info['emp']=$this->data\_model->mywork\_view($emid);

$this->load->view('empmywork',$info);

}

public function admserreq()

{

$this->load->model('data\_model');

$info['emp']=$this->data\_model->get\_data("service");

$this->load->view('adminserreqlst',$info);

}

public function empdelete()

{

$this->load->model('data\_model');

$info['emp']=$this->data\_model->get\_data("employee");

$this->load->view('adminremovestaf',$info);

}

public function del\_data() //delete

{

$id=$this->uri->segment(3);

$this->load->model('data\_model');

$id1=$this->data\_model->get\_empmail($id);

$cid=$id1->email;

// echo $cid;

$this->data\_model->data\_del("employee",$id);

$this->data\_model->usr\_del("users",$cid);

header('location:'.base\_url().'Welcome/empdelete');

}

public function editemp()

{

$this->load->model('data\_model');

$info['emp']=$this->data\_model->get\_data("employee");

$this->load->view('admineditemp',$info);

}

public function updat\_data() //post

{

$id=$this->uri->segment(3);

$this->load->model('data\_model');

// $data['cust']=$this->data\_model->get\_data("bill\_detail");

$data['emp']=$this->data\_model->get\_data("employee");

$data['updat']=$this->data\_model->data\_updat("employee",$id);

$this->load->view('adminstaffeditform',$data);

}

public function updat\_status() //post

{

$id=$this->input->post('tid');

// $id=$this->uri->segment(3);

$data=array('usrname'=>$this->input->post('name'),

'usremail'=>$this->input->post('email'),

'mobile'=>$this->input->post('phone'),

'subject'=>$this->input->post('subj'),

'department'=>$this->input->post('dept'),

'service'=>$this->input->post('service'),

'priority'=>$this->input->post('priority'),

'message'=>$this->input->post('message'),

'status'=>$this->input->post('status'),

);

$this->load->model('data\_model');

$this->data\_model->data\_updatestatus("ticket",$data,$id);

$this->load->model('data\_model');

$info =$this->data\_model->get\_count1();

$info1 =$this->data\_model->get\_count2();

$info2 =$this->data\_model->get\_count3();

$info3 =$this->data\_model->get\_count4();

$info4 =$this->data\_model->get\_count5();

$info5 =$this->data\_model->get\_count6();

$data = array('planet' => $info,'planet1' => $info1,'planet2' => $info2,'planet3' => $info3,'planet4' => $info4,'planet5' => $info5);

$this->load->view('admindash',$data);

}

public function assignemp() //post

{

$srid=$this->input->post('srid');

$this->load->model('data\_model');

$data['emp']=$this->data\_model->data\_service("service",$srid);

$data['bankdata'] = $this->data\_model->getbanklist();

$this->load->view('adminassignemp', $data);

}

public function updat\_serreq() //load onto form

{

$id=$this->uri->segment(3);

$this->load->model('data\_model');

// $data['cust']=$this->data\_model->get\_data("bill\_detail");

$data['emp']=$this->data\_model->get\_data("service");

$data['updat']=$this->data\_model->data\_service("service",$id);

$this->load->view('adminserveeditform',$data);

}

public function cust\_replay() //cust tick replay

{

$id=$this->input->post('tid');

// $id=$this->uri->segment(3);

$data=array('usrname'=>$this->input->post('name'),

'usremail'=>$this->input->post('email'),

'mobile'=>$this->input->post('phone'),

'subject'=>$this->input->post('subj'),

'department'=>$this->input->post('dept'),

'service'=>$this->input->post('service'),

'priority'=>$this->input->post('priority'),

'message'=>$this->input->post('message'),

'status'=>'Customer replay',

);

$this->load->model('data\_model');

$this->data\_model->data\_updatestatus("ticket",$data,$id);

$maile=$this->session->userdata('user');

$this->load->model('data\_model');

$info['emp']=$this->data\_model->get\_tikid($maile);

$this->load->view('custdash',$info);

}

public function admcustrepmesssfrm() //replay button code

{

$tid=$this->input->post('tid');

$this->load->model('data\_model');

$info['emp']=$this->data\_model->get\_Tick($tid);

// echo $tid;

$this->load->view('admintickrepmesg',$info);

}

public function admcustrep() // admin customer replay form submit code

{

$id=$this->input->post('tid');

// $id=$this->uri->segment(3);

$data=array('usrname'=>$this->input->post('name'),

'usremail'=>$this->input->post('email'),

'mobile'=>$this->input->post('phone'),

'subject'=>$this->input->post('subj'),

'department'=>$this->input->post('dept'),

'service'=>$this->input->post('service'),

'priority'=>$this->input->post('priority'),

'message'=>$this->input->post('message'),

'status'=>$this->input->post('status'),

);

$this->load->model('data\_model');

$this->data\_model->data\_updatestatus("ticket",$data,$id);

$this->load->model('data\_model');

$info['emp']=$this->data\_model->get\_data("ticket");

$this->load->view('adminticket',$info);

}

public function updat\_cust() //update

{

$id=$this->input->post('hid');

//$id=$this->uri->segment(3);

$data=array('fname'=>$this->input->post('fname'),

'lname'=>$this->input->post('lname'),

'phone'=>$this->input->post('phone'),

'email'=>$this->input->post('email'),

'dob'=>$this->input->post('dob'),

'gender'=>$this->input->post('gender'),

'adr1'=>$this->input->post('adr1'),

'adr2'=>$this->input->post('adr2'),

'city'=>$this->input->post('city'),

'country'=>$this->input->post('country'),

'pcode'=>$this->input->post('pcode'),

'desig'=>$this->input->post('desig'),

'pass'=>$this->input->post('pass')

);

$this->load->model('data\_model');

$this->data\_model->data\_update("employee",$data,$id);

// header('location:'.base\_url().'Login/admineditemp');

$this->load->model('data\_model');

$info['emp']=$this->data\_model->get\_data("employee");

$this->load->view('admineditemp',$info);

}

public function updat\_emppro() //update

{

$id=$this->input->post('hid');

//$id=$this->uri->segment(3);

$data=array('fname'=>$this->input->post('fname'),

'lname'=>$this->input->post('lname'),

'phone'=>$this->input->post('phone'),

'email'=>$this->input->post('email'),

'dob'=>$this->input->post('dob'),

'gender'=>$this->input->post('gender'),

'adr1'=>$this->input->post('adr1'),

'adr2'=>$this->input->post('adr2'),

'city'=>$this->input->post('city'),

'country'=>$this->input->post('country'),

'pcode'=>$this->input->post('pcode'),

'desig'=>$this->input->post('desig'),

'pass'=>$this->input->post('pass')

);

$this->load->model('data\_model');

$this->data\_model->data\_update("employee",$data,$id);

// header('location:'.base\_url().'Login/admineditemp');

$this->load->model('data\_model');

$emil=$this->session->userdata('employemail');

//echo $emil;

$emidt=$this->data\_model->getempid($emil);

$id=$emidt->empid;

$data['emp']=$this->data\_model->get\_data("employee");

$data['updat']=$this->data\_model->data\_updat("employee",$id);

$this->load->view('employeprofile',$data);

}

public function log()

{

$id=$this->input->post('username');

$pass=$this->input->post('pass');

$this->load->model('data\_model');

$data['cust']=$this->data\_model->get\_user("users",$id,$pass,'c');

$data['admin']=$this->data\_model->get\_user("users",$id,$pass,'a');

$data['emp']=$this->data\_model->get\_user("users",$id,$pass,'e');

$this->session->set\_userdata(array('uname'=>$id));

if($data['cust']!=null)

{

// $\_SESSION['c\_email'] = $id;

$this->session->set\_userdata('user',$id);

$maile=$this->session->userdata('user');

$this->load->model('data\_model');

$info['emp']=$this->data\_model->get\_tikid($maile);

$this->load->view('custdash',$info);

}

else if($data['admin']!=null)

{

$this->session->set\_userdata('adminemail',$id);

$this->load->model('data\_model');

$info =$this->data\_model->get\_count1();

$info1 =$this->data\_model->get\_count2();

$info2 =$this->data\_model->get\_count3();

$info3 =$this->data\_model->get\_count4();

$info4 =$this->data\_model->get\_count5();

$info5 =$this->data\_model->get\_count6();

$data = array('planet' => $info,'planet1' => $info1,'planet2' => $info2,'planet3' => $info3,'planet4' => $info4,'planet5' => $info5);

$this->load->view('admindash',$data);

}

else if($data['emp']!=null)

{

$this->session->set\_userdata('employemail',$id);

$this->load->model('data\_model');

$info =$this->data\_model->get\_count1();

$info1 =$this->data\_model->get\_count2();

$info2 =$this->data\_model->get\_count3();

$info3 =$this->data\_model->get\_count4();

$info4 =$this->data\_model->get\_count5();

$info5 =$this->data\_model->get\_count6();

$data = array('planet' => $info,'planet1' => $info1,'planet2' => $info2,'planet3' => $info3,'planet4' => $info4,'planet5' => $info5);

$this->load->view('employedash',$data);

}

else

{

$temp['us']="incorrect email/password combination";

$this->load->view('Loginform',$temp);

}

}

public function evntreg()

{

// $email=$this->input->post('email');

$data1=array('evntname'=>$this->input->post('evntitle'),

'evnttime'=>$this->input->post('evntime'),

'evntdesc'=>$this->input->post('evndesc'),

'evntdate'=>$this->input->post('evnd')

);

$this->load->model('data\_model');

$this->data\_model->insert\_data("evnttable",$data1);

$this->load->view('admindash');

}

public function tickreg()

{

// $email=$this->input->post('email');

$data1=array('usrname'=>$this->input->post('name'),

'usremail'=>$this->input->post('email'),

'mobile'=>$this->input->post('phone'),

'subject'=>$this->input->post('subj'),

'department'=>$this->input->post('dept'),

'service'=>$this->input->post('service'),

'message'=>$this->input->post('message'),

'status'=>"open"

);

$this->load->model('data\_model');

$this->data\_model->insert\_data("ticket",$data1);

}

public function tickregister()

{

$data1=array('usrname'=>$this->input->post('name'),

'usremail'=>$this->input->post('email'),

'mobile'=>$this->input->post('phone'),

'subject'=>$this->input->post('subj'),

'department'=>$this->input->post('dept'),

'service'=>$this->input->post('service'),

'priority'=>$this->input->post('priority'),

'message'=>$this->input->post('message'),

'status'=>"open",

'atchment'=>"Null"

);

$this->load->model('data\_model');

$this->data\_model->insert\_data("ticket",$data1);

$usrmail=$this->input->post('email');

$mobile=$this->input->post('phone');

$sub=$this->input->post('subj');

$dept=$this->input->post('dept');

$info['emp']=$this->data\_model->get\_idd($sub,$dept,$mobile,$usrmail);

$this->load->view('custticketSucess',$info);

}

public function empreg()

{

// $email=$this->input->post('email');

$data1=array('fname'=>$this->input->post('fname'),

'lname'=>$this->input->post('lname'),

'phone'=>$this->input->post('phone'),

'email'=>$this->input->post('email'),

'dob'=>$this->input->post('dob'),

'gender'=>$this->input->post('gender'),

'adr1'=>$this->input->post('adr1'),

'adr2'=>$this->input->post('adr2'),

'city'=>$this->input->post('city'),

'country'=>$this->input->post('country'),

'pcode'=>$this->input->post('pcode'),

'desig'=>$this->input->post('desig'),

'pass'=>$this->input->post('pass')

);

$data2=array('name'=>$this->input->post('fname'),

'email'=>$this->input->post('email'),

'password'=>$this->input->post('pass'),

'utype'=>'e'

);

$this->load->model('data\_model');

// $res=$this->data\_model->get\_email($email);

// if($res==null)

// {

$this->data\_model->insert\_data("employee",$data1);

$this->data\_model->insert\_data("users",$data2);

$this->load->model('data\_model');

$info =$this->data\_model->get\_count1();

$info1 =$this->data\_model->get\_count2();

$info2 =$this->data\_model->get\_count3();

$info3 =$this->data\_model->get\_count4();

$info4 =$this->data\_model->get\_count5();

$info5 =$this->data\_model->get\_count6();

$data = array('planet' => $info,'planet1' => $info1,'planet2' => $info2,'planet3' => $info3,'planet4' => $info4,'planet5' => $info5);

$this->load->view('admindash',$data);

}

public function logout()

{

$this->session->sess\_destroy();

$temp['us']="";

$this->load->view('Loginform',$temp);

}

}

**3.8 Navigation Chart**



Fig 3.8 Navigation chart

**3.9 Report Generation**

**Transaction reports**

All the transaction done by the customer will be generated as report .That gives the Shoppe a overview with amount received and the amount that is kept due or in pending. According to the report generated the Shoppe can update and give reminders to the customer.

**Data Report**

Data report is the report that is generated from the items purchased by the customer. So that the Shoppe can collect these points to find, the product in need most, customer interest etc, which helps the Shoppe to increase the profit.

## Chapter 6: Summary and Future Enhancements

## 6.1 Summary

This website contains a simple user-friendly interface which helps all the users to interact with the website in an easier way .The admin user can easily manage all the processes and transactions which are happening in the organization through this user friendly website . here the admin can manage all the employees and can manage all the events which are happening in the organization through this website , so that all the employees may get notified about the events which are happening in the organization through this website

The clients can have access to this website and they can make request for a new service or can make a request for getting support for their existing services. The admin can view all the services and support requests which are made by the clients , and he can allocates those works tho the employees in the organization through this website .

The employees in this organization can have access to this website and they can view all the works which are happening in the organization . the employee can view the works or the particular tasks which are allocated to him only through this website and he can update the status of the work . the admin and the clients can also view the status of the works which are doing by the employee.

The admin of the website can edit the company profile which includes their contact details, about us content, statistics etc, the slider in the homepage, Services provided by the company and much more.

This system is necessary for similar organizations in the market who takes up ad hoc projects for the clients. This system will benefit both the clients and the service provider equally. The clients will be able to track the work progress and request for new changes as easy as a click of a button. At the same time the service provider can also save huge amounts of time and work load by centralizing their data and storing all the necessary information about the project so that they can get back to it quickly in the future.

Including a CMS in a website has become necessary in this fast-moving world and hence having a built in CMS rather than depending on third party makes it even better. One can edit their websites as they wish without knowing any coding from the comfort of their homes.

## 6.2 Future Enhancement

The Future Enhancement of the project is to develop a smartphone app for the ease of use for the users. This can be achieved in much simpler way by converting this web app into an API for the phone app, that leaves with the design part of the app to be coded. Also, the users can use the web as well as the phone app simultaneously even after the phone app is deployed to the users.