**Synopsis**

**Contents**

* Title of the Project 3
* Introduction 4
* Objective 5
* Data Flow Diagram 7
* Entity-Relationship Diagram 11
* Project Category 13
* Software and hardware Requirements ­­­ 14
* Database Design 15
* Basic Modules 18
* Report Generation 20
* Future Scope 21
* Bibliography 22

**TITLE OF THE PROJECT**

HELPDESK OF MOBILE PRODUCT

**INTRODUCTION**

The aim of this project is to design database management for help desk, which is completely interactive. The system can help the customer information retrieval services of the helpdesk in very quickly a proper way as well to maintain help desk information efficiently. The system also creates various reports needed by the Enquire/ Reception of helpdesk.

The system may need modification when they are changes in procedure within the HELPDESK OF MOBILE PRODUCT or new requirement from the user. To ease the task of making these modifications, documentation is necessary. So further documentation gives us a full understanding about this system and helps us to do modifications effectively and efficiently.

In the fast growing world of computers, this type of computerized environment has proved a great advantage for all of us. It not only provides the fastest mode of working but it saves a lot of time of other and our also. Manually we can do a lot of work but it is so less in comparison to machine works, so in that way also computerized system is very safe, reliable and easy to work on it. Computer is the big demand of today’s world as everybody wants to get in touch with it so that it will make his/her work more comprise and more easier. So the basic aim of converting the manual work into computerized work is that you can have the food environment and also you can do the work more easily and in more sophisticated manner.

**OBJECTIVE**

Today’s world is the world of computers now we are living in the machine age where man has become totally dependable upon the fastest medium to achieve his/her requirements so every things has been changing with time, in the same way our mode of working has also been changed, first we all us to do our work with help of man power, no we are totally dependable upon the machine, anywhere you go you will find a computers in front of you because man has become use too, of all these comforts. It is very good also as our lot of time has been saved if we perform our task with the help of computers, it not only saves our time but also it gives us wide performance our task with the help of computers, it not only saves our time but also it gives us wide perform for all our activities. In my project I had implemented the mental labor of the accountants to a computerized from this not only help them to do their work more sophistically but also help them in maintaining their criteria of doing work. In my project I had implemented the computerized data system in order to have more easier way to communicate and also to save time.

The main objectives are targets for the achievements, and serve to establish the framework for a software development projects. It applies to both the development process and the work products. Some are as:-

* It should be user friendly.
* It should allow the user to maintain the CUSTOMER and CASH MEMO record efficiently.
* It should also allow the user to maintain the administration, and payroll of employee.
* It should very economical and very fast.
* It should maintain the report card at the time of distribution. It must be general system such that it complaint of the user. This project is capable of managing the function of help desk Management System.

**DATA FLOW DATAGRAMS**

**LEVEL 0 DFD**

**USER**

**ADMIN**

**LEVEL 1 DFD**

USER

**USER**

**ADMIN**

**LOGOUT**

**CONFIGURATION RELATED PROCESS**

**ADMIN**

USER

PROBLEM REPORT

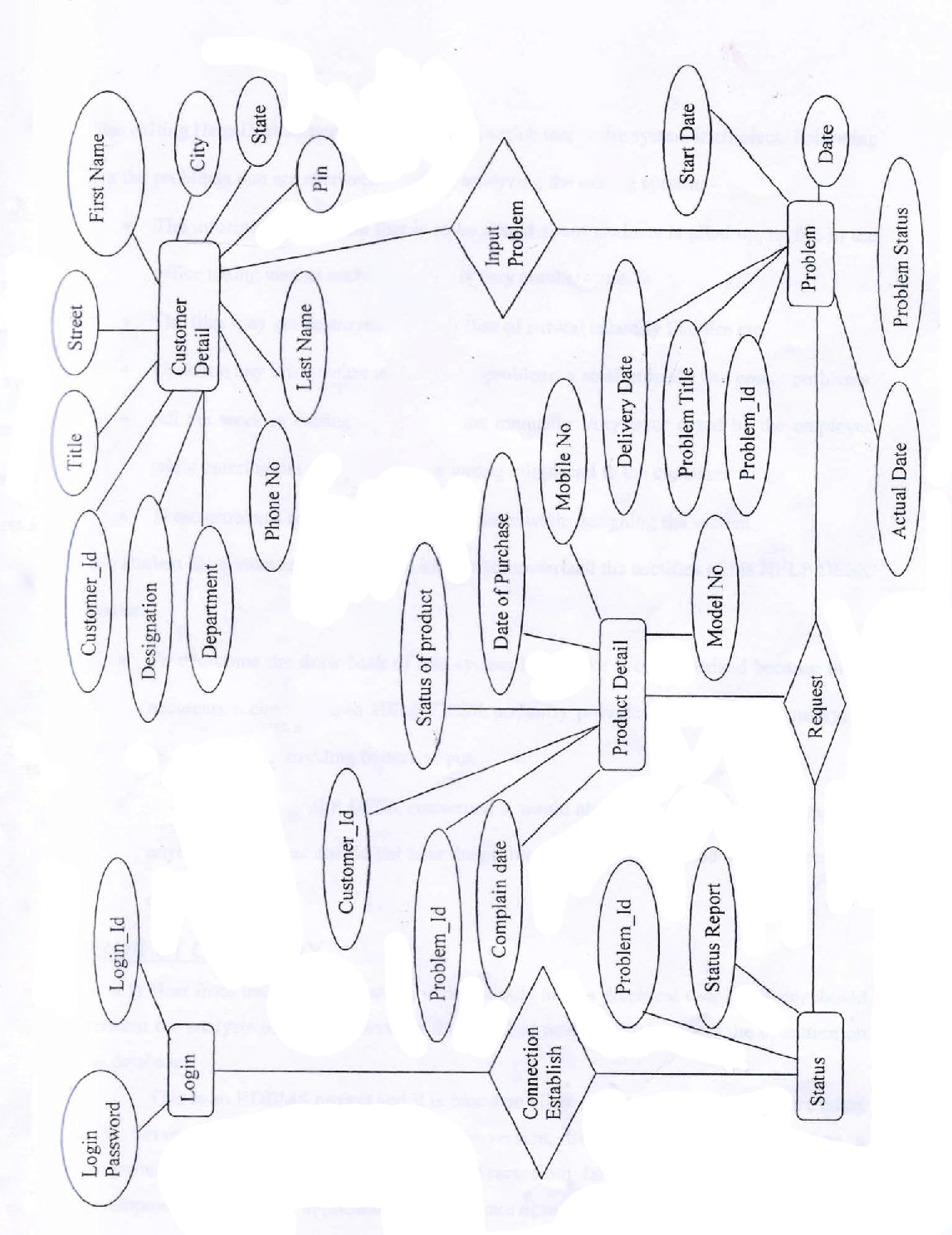
ADMIN

**USER RELATED PROCESS**

**USER**

**1.2**

**1.3 USER**

**ERD**

**­­**

**EXISTING SYSTEM**

The exiting help desk system faces problems, which makes the system inefficient. Following are the problems that are encountered while analyzing the existing system:-

* The information and data that is to be stored about students is piled up, as file in the office taking care of such document is very cumbersome.
* The files may get destroyed in the in case of natural calamity like fire etc.
* Updating any information make causes problems a small mistake can create problems.
* All the work in existing system is done manually any error caused by the employee while entering the data result into a wrong bill placed to the customer.
* These problems are taken into consideration while designing the system.

We student his system and suggest and idea to computerize the acetifies of his HELPDESK OF MOBILE PRODUCT.:-

To overcome the drawback of this system is to make it computerized because in all accidents a computerized help desk authority provides accuracy, consistency, in its flow and us providing better output.

* In computerized HELP DESK concerned it would always be in mind of the head that anyhow the student should not bear the problems and must get good service rater than others.

**PROJECT CATEGORY**

As it is clear from using an application, which should have a graphical user interface, should perform the analysis of user requirements that the user needs a database and the operation on the database.

This is an RDBMS project and it is based on client server environment. We are using SQL Server as RDBMS. With every new version, SQL server has evolved into a programming language. It has grown over years from language introduced for simple development of graphical application with software development component.

**SOFTWARE & HARDWARE REQUIREMENTS**

**Software**

Client Side :HTML, CSS & Java Scripting

Server Side : PHP

Back end : Mysql

Operating system : Windows XP. And Later

Web server : Apache Tomcat or GlassFish

**Hardware**

Intel Pentium 4 Processor

512MB RAM

20GB Hard disk

Hardware Management Software

Power Supply

Color Monitor

Key Board

Mouse

LAN card/Modem

Printer

**DATABASE TABLES**

**LOGIN**

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Data Type** | **Size** |
| Login\_Id | Varchar | 10 |
| Password | Varchar | 15 |

**PROBLEM**

|  |  |  |  |
| --- | --- | --- | --- |
| **SR.No** | **Field Name** | **Data type** | **Size** |
| 1 | Problem\_Id | Varchar | 6 |
| 2 | Problem\_Title | Varchar | 15 |
| 3 | Problem\_status | Varchar | 20 |
| 4 | Problem\_Type | Varchar | 10 |
| 5 | Date | Date |  |
| 6 | Start\_date | Date |  |
| 7 | Actual\_Date | Date |  |
| 8 | Delivery | Date |  |

**PRODUCT DETAILS**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.R.No** | **Field Name** | **Data Type** | **Size** |
| 1 | Date\_Of\_Purchase | Integer |  |
| 2 | Status\_Of\_Product | Varchar | 10 |
| 3 | Problem\_Id | Varchar | 15 |
| 4 | Customer\_Id | Varchar | 15 |
| 5 | Complaint\_Date | Integer |  |
| 6 | Model\_No | Integer |  |
| 7 | Mobile\_No | Integer |  |

**CUSTOMER DETAILS**

|  |  |  |  |
| --- | --- | --- | --- |
| **SR.No** | **Field Name** | **Data type** | **Size** |
| 1 | Customer\_Id | Varchar | 6 |
| 2 | Title | Varchar | 15 |
| 3 | First\_Name | Varchar | 15 |
| 4 | Last\_Name | Varchar | 12 |
| 5 | Designation | Varchar | 15 |
| 6 | Department | Varchar | 15 |
| 7 | Street | Varchar | 15 |
| 8 | City | Varchar | 10 |
| 9 | State | Varchar | 10 |
| 10 | Pin | Integer |  |
| 11 | Phone\_Number | Integer |  |

**STATUS**

|  |  |  |  |
| --- | --- | --- | --- |
| **SR.No** | **Field Name** | **Data type** | **Size** |
| 1 | Problem\_Id | Varchar | 6 |
| 2 | Status\_Report | Varchar | 25 |

**SECURITY MEASUREMENTS TAKEN**

Security and controls are required not only for preventing inadvertent mistakes made by user, but also to prevent misuse of the system and ensure data integrity.

In an automated system of Helpesk the following security measures have been considered:

1. Access control

Access Control can be defined as control established to ensure that only people with the proper authority have access to the data. Different types of access control, which are used in this software, are as Follow:

1. Transaction Access

All users of system need to have access to all transactions in a system. Users must have access only to transactions, which are required by them for their day-to-day operations. A manager in a office needs to submit his daily report only, Thus he should not have access to the Office records.

1. Data Access

Apart from having access to transaction, which are required for daily operations, users must only be allowed to see specific instances of transactions. In some applications it may be required that users must only see data relating to their needs. Thus the head of the department will have access to the department query transaction, but the user will not be shown all data, the user will be shown data belonging’s to the user’s department only.

In Help Desk Management, a user cannot see the details of the records entered.

C) Physical Access

The best form of access control is the physical access control. Only authorized personal should be allowed to do specific tasks like booting of the machine, mounting disks, distributing reports etc.

In this software the database is centralized means all the data Is stored at one place called as server whereas number of clients are attached to it. Without the server nothing can be done. So, for starting server an authorized person should be there.

1. Validations

Validations of data entered on various input screens are not a security for the system but a control to ensure that clean data goes into the system. If data is not validated on input, you will end up with a lot of inconsistent data.

In computer parlance, this is referred to as ‘Garbage In, Garbage Out’. If proper validation is not done, user tends to lose confidence in the system and it eventually boils down to a new project of clearing up the data. Typical validations done on input screen are as follows:

* Length of data.
* Type of data (numeric, integers, alphabetic, alphanumeric)
* Validation against masters
* Discreet values
* Cross validation of data across various fields

1. Data Integrity

We can have the best of validations in a system but invalid data can always get into the system due to some recent changes done on the software, which was not tested comprehensively, or due to corruption. Whatever the cause, it is a very good control practice to periodically check the integrity of the data o the system. Data Integrity could be verified by a number of techniques. Some of the used techniques are listed below:

* Tallying number of records across various tables in the database.
* Tallying critical fields across tables in the database.

Whatever be the techniques, periodic validation of the data will gives us early warning signals in case something in wrong with the database.

**FUTURE SCOPE OF THE PROJECT**

For future use, there are so many provisions specified with the help of

Which the system can also survive in the future with its excellent

capacity and robustness. With the help of these provisions, the system

may also be used for different helpdesk like power house ,reservation etc.

**BIBLIOGRAPHY**

WEBSITES

* [www.google.com](http://www.google.com)
* [www.wikipedia.org](http://www.wikipedia.org)
* [www.w3schools.com](http://www.w3schools.com)
* [www.youtube.com](http://www.youtube.com)

**BOOKS**

* “Beginning PHP and MySQL From Novice to Professional” by W Jason Gilmore
* “PHP: A Beginner’s Guide” by Vikram Vaswani
* “Learning PHP 5” by David Sklar
* “PHP Object – Oriented Solutions” by David Powers

**PROJECT REPORT**

**CONTENTS**

1. TITLE OF THE PROJECT 31
2. INTRODUCTON 32
3. OBJECTIVE 33
4. PROJECT CATEGORY 35
5. SOFTWARE REQUIREMENTS SPECIFICATION 37
6. HARDWARE & SOFTWARE CONFIGURATION 42
7. SYSTEM ANALYSIS 43
8. DFD 45-47
9. E-RD 48
10. CLASS DIAGRAM 49-54
11. BASIC MODULE 54-58
12. TABLE DESCRIPTION 59-67

8. SYSTEM DESIGN 68-78

9. FEASIBILITY STUDY 78-83

1. TECHNICAL FEASIBILITY
2. ECONOMICAL FEASIBILITY
3. OPERATIONAL FEASIBILITY

10. USER INTERFACE’S DESIGN 84-100

11. CODING 101-136

12. TESTING 137-139

13. IMPLEMENTATION 140-141

14. PEART CHART & GANTT CHART 142-145

15. SYSTEM SECURITY 146

16. FUTURE SCOPE 147

17. LIMITATIONS 148

18. BIBLIOGRAPHY 149

**TITLE OF THE PROJECT**

**HELPDESK OF MOBILE PRODUCT**

**INTRODUCTION**

The aim of this project is to design database management for help desk, which is completely interactive. The system can help the customer information retrieval services of the helpdesk in very quickly a proper way as well to maintain help desk information efficiently. The system also creates various reports needed by the Enquire/ Reception of helpdesk.

The system may need modification when they are changes in procedure within the HELPDESK OF MOBILE PRODUCT or new requirement from the user. To ease the task of making these modifications, documentation is necessary. So further documentation gives us a full understanding about this system and helps us to do modifications effectively and efficiently.

In the fast growing world of computers, this type of computerized environment has proved a great advantage for all of us. It not only provides the fastest mode of working but it saves a lot of time of other and our also. Manually we can do a lot of work but it is so less in comparison to machine works, so in that way also computerized system is very safe, reliable and easy to work on it. Computer is the big demand of today’s world as everybody wants to get in touch with it so that it will make his/her work more comprise and more easier. So the basic aim of converting the manual work into computerized work.

**OBJECTIVE**

Today’s world is the world of computers now we are living in the machine age where man has become totally dependable upon the fastest medium to achieve his/her requirements so every things has been changing with time, in the same way our mode of working has also been changed, first we all us to do our work with help of man power, no we are totally dependable upon the machine, anywhere you go you will find a computers in front of you because man has become use too, of all these comforts. It is very good also as our lot of time has been saved if we perform our task with the help of computers, it not only saves our time but also it gives us wide performance our task with the help of computers, it not only saves our time but also it gives us wide perform for all our activities. In my project I had implemented the mental labor of the accountants to a computerized from this not only help them to do their work more sophistically but also help them in maintaining their criteria of doing work. In my project I had implemented the computerized data system in order to have more easier way to communicate and also to save time.

The main objectives are targets for the achievements, and serve to establish the framework for a software development projects. It applies to both the development process and the work products. Some are as:-

* It should be user friendly.
* It should allow the user to maintain the CUSTOMER and CASH MEMO record efficiently.
* It should also allow the user to maintain the administration, and payroll of employee.
* It should very economical and very fast.
* It should maintain the report card at the time of distribution. It must be general system such that it complaint of the user. This project is capable of managing the function of help desk Management System.

**PROJECT CATEGORY**

As it is clear from using an application, which should have a graphical user interface, should perform the analysis of user requirements that the user needs a database and the operation on the database.

This is an RDBMS project and it is based on client server environment. We are using SQL Server as RDBMS. With every new version, SQL server has evolved into a programming language. It has grown over years from language introduced for simple development of graphical application with software development component.

**SOFTWARE REQUIREMENTS SPECIFICATION**

**The Project is developed using Relational Database Management System (RDBMS).**

A database system is essentially a sophisticated, computerized record keeping system, a repository for a collection of computerized data files. A database system maintains information and makes that information available on demand, for this purpose a database system provides set of facilities to perform such operations. The benefits of a database system over any traditional system are obvious as database is integrated as well as shared, thus a database eliminates redundancy and also as a consequence, database lets multiple users access the same piece of data.

The most important advantage of the database is to maintain the integrity i.e. it insures that the change made to the database by authorized users do not result in a loss of data consistency and guard against accidental damage to the database.

**Some of the merits of using SqlServer 2000 (RDBMS) are as under:**

* Centralization of database.
* Client Server Technology.
* Security.
* Normalization of Data Base.
* Relationship.
* Transaction Processor.
* It gives some internet related features.

Hence because of these features we are using SqlServer as a back-end technology. It is a reference document or contract between the customer and the development team. Once the customer agrees to the SRS document the development team proceeds to develop the product conforming to all the requirements mentioned in the SRS document.

An SRS document should clearly document the following:

1. Functional requirements of the system.

2.Non-functional requirements of the system.

3.Constraints on the system.

1. Functional requirements of the system: Each of the system can be considered as performing a transformation of a set of input data to the corresponding set of output data. The functional requirements of the system should clearly describe each of the functions that the system needs to perform along with the corresponding input and output data set.

2. Non-functional requirements of the system: Non-functional requirements deal with the characteristics of the system that cannot be expressed functionally, e.g., maintainability, portability, Usability, etc. The non-functional requirements also include reliability issues, accuracy of results, human computer interface issues, operating and Physical constraints, etc.

3. Constraints on the system: The constraints on the &u”s of the system may describe certain things that the system should or should not do.

**Natures of SRS**

The basic issues the SRS writer(s) shall address are the following:

1. Functionality: What the software is supposed to do?

2. External Interfaces: How does the software interact with people, the system’s hardware other hardware and other software.

3. Performance: What is the speed, availability, response time, recovery time, etc., of the various software fundamentals?

4. Attributes: What are the considerations for portability, correctness, maintainability, security, reliability, etc.

5. Design constraints imposed on an implementation: Are there any required standards or effect, implementation language, policies for database, integrity resource limits, operating environment, etc.

Characteristics of a good SRS

An SRS should be

* Correct
* Unambiguous
* Complete
* Consistent
* Ranked for Importance and for Stability
* Verifiable
* Modifiable
* Traceable

**Correct:** There is no tool or procedure that assures correctness. If the software must respond to all button presses within 5 seconds and the SRS stated that “the software shall respond to all button presses with in 10 seconds”, then that requirement is incorrect.

**Unambiguous:** An SRS is unambiguous if and only if every requirement started therein has only are interpretation. In cases, where a term used in a particular context could have multiple meanings, the term should be included in a glossary where its meaning is made more specific.

**Complete:** An SRS is complete if and only if it includes of the following elements.

1. All significant requirements, whether relating to functionality, performance, design constraints, attributes or external interfaces.

2. Full labels and references to all figures, tables and diagram in the SRS and definition of all terms and units of measure.

**CONSISTENT**

An SRS is consistent if no subset of individual requirements desorbed in it conflict. There are 3 types of likely conflicts in an SRS:

1. The specified characteristics of real word objects may conflict, e.g.

a. The format of an output report may be described in are requirements as tabular but in another as textual.

b. One requirement may state that all lights shall be green while another states that all lights should be blue.

2. There may be logical or temporal conflict between two specified actions, e.g.

a. Are requirement may specify that the program will add 2 inputs and another may specify that the program will multiply them.

b. Are requirement may state that ‘A’ must always follows B, while another requires that A&B occur simultaneously.

3. Two or more requirements may describe the same real word object but use different terms for that object. The use of standard terminology and definitions promotes consistency.

**Hardware & Software Configuration**

**Software**

Client Side :HTML, CSS & Java Scripting

Server Side : PHP

Back end : Mysql

Operating system : Windows XP. And Later

Web server : Apache Tomcat or GlassFish

**Hardware**

Intel Pentium 4 Processor

512MB RAM

20GB Hard disk

Hardware Management Software

Power Supply

Color Monitor

Key Board

Mouse

LAN card/Modem

Printer

**SYSTEM ANALYSIS**

**LEVEL 0 DFD**

**USER**

**ADMIN**

**LEVEL 1 DFD**

USER

**USER**

**ADMIN**

**LOGOUT**

**CONFIGURATION RELATED PROCESS**

**ADMIN**

USER

PROBLEM REPORT

ADMIN

**USER RELATED PROCESS**

**USER**

**1.2**

**1.3 USER**

**ER-DIAGRAM**

Have

customer

Admin

Check Status

Input problem

Request

status

Problem

Product

**CLASS DIAGRAM**

|  |
| --- |
| **Logintable** |
| **Attribute:-**  Email\_id:varchar  password: varchar |
| **Operation:-**  log\_in, cancel |
| **College** |
| **Attribute:-**  College\_code:varchar  College\_name:varchar  Department: varchar  Principal: varchar  Rank: varchar  Address: varchar  City: varchar  State: varchar  Contact1: varchar  Contact2: varchar |
| **Operation:-**  Save, Search, Update, exit |
| **Student\_rec** |
| **Attribute:-**  Enrollment\_no:varchar  Tittle: varchar  First\_name: varchar  Middle\_name: varchar  Last\_name: varchar  Gender: varchar  Date\_of\_birth: Date/time  Place\_of\_birth: Date/time  Country: varchar  College\_name: varchar  Tounge: varchar  Fluency: varchar  Date\_of\_join: Date/time |
| **Operation:-**  Save, Search, Update,Delete, exit |
| **Panel** |
| Attribute:-  Panel\_code:varchar  Panel\_name: varchar  Name: varchar  Designation: varchar  Organization: varchar  Date\_of\_join:date |
| **Operation:-**  Save, Search, Update, Exit |
| **Company\_rec** |
| Attribute:-  Company\_code:varchar company\_name: varchar  Rank: varchar  Status: varchar  Headoffice: varchar  Address: varchar  Phone1: varchar  Phone2: varchar  Mobile: varchar  Fax: varchar |
| **Operation:-**  Save, Search, Update, Exit |
| **Placement\_cel** |
| Attribute:-  Panel:varchar  Cell\_code:int  Name: varchar  Staff\_code:varchar  Department: varchar  Designation: varchar  Assign\_college: varchar  College\_code: varchar  College\_name: varchar  Salary: varchar |
| **Operation:-**  Save, Search, Update, Exit |
| **Resume** |
| Attribute:-  Name:varchar  email\_id: varchar  mobile\_no: varchar  hspercentage: varchar  hsypass: varchar  hscname: varchar  ipercentage: varchar  iypass: int  icname: varchar  graduation: varchar  gpercentage: varchar  gypass: varchar  gcname: varchar  pgraduation: varchar  pgpercentage: varchar  pgypass: int  pgcname: varchar  additional\_qualification: varchar  work\_experience: varchar  date\_of\_birth: Date/time  father\_name: varchar  nationality: varchar  marital\_status:varchar  language :varchar  gender:varchar  address: varchar |
| **Operation:-**  Save, Search, Update, Exit |
| **Select\_student** |
| Attribute:-  Enrollment\_no:varchar  Tittle: varchar  First\_name: varchar  Middle\_name: varchar  Last\_name: varchar  College: varchar  Panel; varchar  Company: varchar |

**BASIC MODULES**

We know that it is quite difficult to do any thing of its whole part at a time. So it is the task of the programmer to break/split the whole set of task into various small module so that one can handle them effectively. But it depends upon the software system that in which extent it will be broken so that there is no interdependency among them.

This project developed for "CAMPUS PLACMETN " is broken into following module:

* Login Process Module
* Registration Module
* Student Module
* Resume Module
* College Module
* Company Module
* Placement Module
* Panel Module
* Student Report Module
* Resume Report Module
* Selected Student Module
* College Report Module
* Company Report Module
* Placement Report Module
* Panel Report Module

## Description Modules

* **Login Process Module:-** The module Login contains following entity:

1. Eamil\_id

2. Password

The entity Name is used to accept the Login-Email\_id of the Users which are provided only to a few persons in the system to provide security. The Field Password is used to accept password from the user to prevent unauthorized access.

* **Registration Module:-** With this form the user has right to entry login. Signup is must for the new user if he/she wants to go for Campus placement.
* **Student Module:-** This module contains all the details of Students where user can save, update, delete and search student details. This module contains students enrolment number, name, address, date of birth, college details etc
* **Resume Module:-** Resume module contains all details about the students who have registered with placement cell.
* **College Module:-** College module contains all record of colleges and concerned program..
* **Company Module:-** Company module contains all the information of companies regarding new opening for students enrolled with placement cell.
* **Placement Module:-** Placement module contains details of companies for new openings with concerned job description.
* **Panel Module:-** This module contains the information about Board of panels from different companies.
* **Student Report Module:-** This module contains student reports which generate report for placement purpose.
* **Resume Report Module:-** Resume report module generate resume report to the use of employer.
* **Selected Student Module:-** This module contains who is selected by the placement companies.
* **College Report Module:-** College Report module contains to generate selected college i.e. with state or as per criteria.
* **Company Report Module:-** Company report module Shows the companies record.
* **Placement Report Module:-** Placement report module generate the report placement given to students.
* **Panel Report Module:-** This module contains whole campus reports including different panels.

**TABLE DISCRIPTION**

(Main Information Table)

**“LOGIN” TABLE**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Type** | **Size** |
| Email\_id | Varchar | 40 |
| Password | Varchar | 20 |

**“COLLEGE RECORD” TABLE**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Type** | **Size** |
| College\_code | Varchar | 4 |
| College\_name | Varchar | 50 |
| Department | Varchar | 25 |
| Principal | Varchar | 20 |
| Rank | Varchar | 10 |
| Address | Varchar | 60 |
| City | Varchar | 15 |
| State | Varchar | 15 |
| Contact1 | Varchar | 12 |
| Contact2 | Varchar | 12 |

**“STUDENT RECORD” TABLE**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Type** | **Size** |
| Enroll | Varchar | 16 |
| Title | Varchar | 10 |
| First\_name | Varchar | 10 |
| Middle\_name | Varchar | 10 |
| Last\_name | Varchar | 10 |
| Gender | Varchar | 6 |
| Dob | Date/time | …. |
| Place\_of\_birth | Date/time | …. |
| Country | Varchar | 15 |
| College\_code | Varchar | 6 |
| College\_name | Varchar | 60 |
| Tounge | Varchar | 15 |
| Fluency | Varchar | 10 |
| Date\_of\_join | Date/time | …. |

**“PANEL” TABLE**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Type** | **Size** |
| Panel\_code | Varchar | 1 |
| Panel\_name | Varchar | 40 |
| Name | Varchar | 30 |
| Designation | Varchar | 20 |
| Organization | Varchar | 30 |
| Date\_of\_join | Date/time | …. |

**“COMPANY REC” TABLE**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Type** | **Size** |
| Company\_code | Varchar | 6 |
| Company\_name | Varchar | 40 |
| Rank | Varchar | 10 |
| Status | Varchar | 10 |
| Headoffice | Varchar | 40 |
| Address | Varchar | 60 |
| Phone1 | Varchar | 12 |
| Phone2 | Varchar | 12 |
| Mobile | Varchar | 12 |
| Fax | Varcahr | 12 |

**“PLACEMENT CELL” TABLE**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Type** | **Size** |
| Panel | Varchar | 10 |
| Cell\_code | Int | 4 |
| Name | Varchar | 60 |
| Staff\_code | Varchar | 4 |
| department | Varchar | 25 |
| Designation | Varchar | 20 |
| Assign\_college | Varchar | 60 |
| College\_code | Varchar | 6 |
| College\_name | Varchar | 60 |
| Salary | Int | 6 |

**“RESUME” TABLE**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Type** | **Size** |
| Name | Varchar | 30 |
| Email\_id | Varchar | 30 |
| Mobile\_no | Varchar | 12 |
| Hspercentage | Varchar | 5 |
| Hsypass | Int | 4 |
| Hscname | Varchar | 60 |
| Ipercentage | Varchar | 5 |
| Iypass | Int | 4 |
| Icname | Varchar | 60 |
| Graduation | Varchar | 10 |
| Gpercentage | Varchar | 5 |
| Gypass | Int | 4 |
| Gcname | Varchar | 60 |
| Pgraduation | Varchar | 10 |
| Pgpercentage | Varchar | 5 |
| Pgypass | Int | 4 |
| Pgcname | Varchar | 60 |
| Additional\_qualification | Varchar | 200 |
| Work\_experience | Varchar | 500 |
| Date\_of\_birth | Date | …. |
| Father\_name | Varchar | 30 |
| Nationality | Varchar | 10 |
| Marital\_status | Varchar | 10 |
| Language | Varchar | 20 |
| Gender | Varchar | 5 |
| Address | Varchar | 60 |

**“SELECTED STUDENT” TABLE**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Type** | **Size** |
| Enrollment\_no | Varchar | 16 |
| Tittle | Varchar | 10 |
| First\_name | Varchar | 10 |
| Middle\_name | Varchar | 10 |
| Last\_name | Varchar | 10 |
| College | Varchar | 60 |
| Panel | Varchar | 10 |
| Company | Varchar | 35 |

**SYSTEM DESIGN**

**Introduction**

System design provides the understandings and procedural details necessary for implementing the system recommended in the system study. Emphasis is on the translating the performance requirements into design specifications. The design phase is a transition from a user-oriented document (System proposal) to a document oriented to the programmers or database personnel.

System design goes through two phases of development:

**Physical Design**

A data flow diagram shows the logical flow of the system. For a system it describes the input (source), output (destination), database (data stores) and procedures (data flows) all in a format that meets the user’s requirement. When analysis prepare the logical system design, they specify the user needs at a level of detail that virtually determines the information flow into an out of the system and the required data resources. The logical design also specifies input forms and screen layouts.

The activities following logical design are the procedure followed in the physical design e.g., producing programs, software, file and a working system. Design specifications instruct the user about what the system should do.

**Logical and Output Design:**

The logical design of an information system is analogous to an engineering blue print of an automobile. It shows the major features and how they are related to one another. The detailed specification for the new system was drawn on the bases of user’s requirement data. The outputs inputs and databases are designed in this phase.

Output design is one of the most important features of the information system. When the outputs is not of good quality the users will be averse to use the newly designed system and may not use the system. There are many types of output, all of which can be either highly useful or can be critical to the users, depending on the manner and degree to which they are used.

Outputs from computer system are required primarily to communicate the results of processing to users, They are also used to provide a permanent hard copy of these results for later consultation. Various types of outputs required can be listed as below:

* External Outputs, whose destination is outside the organization
* Internal outputs, whose destination is with the organization Operational outputs, whose use is purely with in the computer department e.g., program-listing etc.

Interactive outputs, which involve the user is communicating directly with the computer, It is particularly important to consider human factor when designing computer outputs. End user must find outputs easy to use and useful to their jobs, Without quality output, user may find the entire system unnecessary and avoid using it. The term “Output” in any information system may apply to either printer or displayed information. During the designing the output for this system, it was taken into consideration, whether the information to be presented in the form of query of report or to create documents etc. Other important factors that were taken into consideration are:

The End user, who will use the output. The actual usage of the planned information

The information that is necessary for presentation When and how often output and their format is needed. While designing output for project based Attendance Compilation System, the following aspects of outputs designing were taken into consideration. The outputs (i.e., well formatted table outputs in the screen itself) designed are simple to read and interpret.

Format of each output was another important point taken into consideration. Output media, for each output appropriate media is decided whether it will be displayed on screen or will be taken to printer or both.

Other output design related specifications, i.e., how frequently the outputs will be generated, how many pages or sheets approximately it will keep up, what is its planned use and output distribution to users are also taken into account. These were a few major designing issues, which were taken into consideration, while deciding the output specifications for the system.

As direct beneficiary of reports is the user community, they were consulted constantly at every level. Formats and screen design for various reports were identified, taking into account the user requirements. Before finalising these were given to users for any improvement and suggestions. End users issues taken into consideration were Readability, Relevance and Acceptability.

Once all the output reports to be generated by ACS system were identified, they were given to users for their acceptance. For prototyping various outputs, final outputs models were created with dummy data, before they were finalized.

**Output Sources:**

Output contents originate from these sources:

* Retrieval from a data source.
* Transmission from a process or system activity.
* Directly from an input source.

**The information produced in an output can be presented as**

* Tabular contents
* Graphic format
* Using Icons

**Output Definition:**

The output should be defined in terms of:

**Types of outputs**

* Content-headings, numeric, alphanumeric, etc.,
* Format-hardcopy, screen, microfilm, etc.,
* Location-local, remote, transmitted, etc.,
* Frequency-daily, weekly, hourly, etc.,
* Response-immediate with in a period, etc.,

**Data items**

The name given to each data item should be recorded and its characteristics described clearly in a standard form:

* Whether alphanumeric
* Legitimate and specific range of characteristics
* Number of characters
* Positions of decimal point, arithmetic design, etc.,

***Input Design:***

The input design is the link that ties the information system into the user’s world. Input specifications describe the manner in which data enters the system for processing. Input design features can ensure the reliability of the system and produce results from accurate data, or they can result in the production of erroneous information.

**Input Design consists of**

Developing specifications and procedures for data preparation steps necessary to put data into a usable form for processing.Data entry, the activity of putting data into the computer processing.

**SOFTWARE DESIGN**

The purpose of this phase is to plan a solution for the problem specified by the requirement document. This is first step in moving from the problem domain to solution domain. Designing activity is divided into two parts.

1. **System Design** It aims to identify the modules that should be in the system, the specification of these modules and how they interact with each other to produce the desired result.

**b) Detailed Design** The internal goal of each of the

modules specified in the system design is decided

**DATABASE DESIGN:**

A database is a collection of inter-related data stored with a minimum of redundancy to serve many applications. It minimizes the artificiality embedded in using separate files. The primary objectives are fast response time to enquires, more information at low cost, control of redundancy, clarity and ease of use, accuracy and fast recovery. The organization of data in a database aims to achieve three major objectives, they are data integration, data integrity and data independence. During the design of the database at most care has been taken to keep up the objectives of the database design.

**CODE DESIGN:**

The process of code is to facilitate the identification and retrieve of items of information. The code should be simple and easy to understandable. The codes were designed in such a way that the features such as optimum human – oriented use and machine efficiency are unaffected.

For the code to be designed effectively, the following characteristics were also considered while designing the code.

* Uniqueness
* Versatility
* Stability
* Simplicity
* Consciousness

The code should be adequate for present and anticipated data processing for machine and human use. Care was taken to minimize the clerical effort and computer time required to continue operation.

**PROCESS DESIGN:**

The process can be conceptualized in such a way to keep the methodology of main module process along with some auxiliary task, which will run concurrently with the main program.

The top-down approach is maintained so as to keep track of the process, which satisfies the maintenance reliability testing requirements. The concurrency of the data is checked during data entry, by means of validation check for data in each field.

**FEASIBILITY STUDY**

A feasibility study is conducted to select the best system that meets performance requirement. This entails an identification description, an evaluation of candidate system and the selection of best system for the job. The system required performance is defined by a statement of constraints, the identification of specific system objective and a description of outputs.

**The Key consideration in feasibility analysis are :**

1. Technical Feasibility

2. Operational Feasibility

3. Economic Feasibility

**Technical Feasibility:** Technical feasibility center around the existing computer system hardware etc. and to what extent it can support the proposed addition. For example, if the current computer is operating at 80% capacity - an arbitrary ceiling – then running another application could over load the system or require additional hardware. This involves financial consideration to accommodate technical enhancements. If the budget is a serious constraint then the project is judged not feasible .It is a measure of the practically of a specific technical solution and the availability of technical resources and expertise. The proposed system uses Microsoft Visual Studio.NET as front-end and Sql Server 2000 as back-end tool.

Sql Server 2000 is a popular tool used to design and develop database objects such as table views, indexes. The above tools are readily available, easy to work with and widely used for developing commercial application.

Hardware used in this project are Pentium-4 class processor, 800 (MHz), 256 MB RAM, 40 Gigabytes (GB) on installation drive, floppy drive. These hardware were already available on the existing computer system. The software like Microsoft Visual Studio.NET and operating system Window XP/2000 used were already installed On the existing computer system. So no additional hardware and software were required to purchase and it is technically feasible. The technical feasibility is in employing computers to the organization. The organization is equipped with enough computers so that it is easier for updating. Hence the organization has not technical difficulty in adding this system.

**Operational Feasibility:** It is common knowledge that computer installations have some thing to do with turnover, transfers, retraining and changes in employee job status. Therefore, it is understandable that the introduction of a candidate system requites special efforts to educate, sell, and train the staff on new ways of conducting business. The system will be used if it is developed well then be resistance for users that undetermined No major training and new skills are required as it is based on DBMS model. It will help in the time saving and fast processing and dispersal of user request and applications. New product will provide all the benefits of present system with better performance. Improved information, better management and collection of the reports.

**User support.**

User involvement in the building of present system is sought to keep in mind the user specific requirement and needs. User will have control over there own information. Important information such as order can be generated at the click of a button. Faster and systematic processing of user application approval.

**Economic Feasibility:** Economic analysis is the most frequently used method for evaluating the effectiveness of a candidate system. More determine the benefits and the saving that are expressed from a candidate system and compare them costs.

If benefits outweigh costs. Otherwise, further justification or alterations in the proposed system will have to be made if it is to have a chance of being approved. This is an ongoing effort that improves in accuracy at each phase of the system life cycle.

It looks at the financial aspects of the project. It determines whether the management has enough resources and budget to invest in the proposed system and the estimated time for the recovery of cost incurred. It also determines whether it is worth while to invest the money in the proposed project. Economic feasibility is determines by the means of cost benefit analysis. The proposed system is economically feasible because the cost involved in purchasing the hardware and the software are within approachable. The personal cost like salaries of employees hired are also nominal, because working in this system need not required a highly qualified professional.

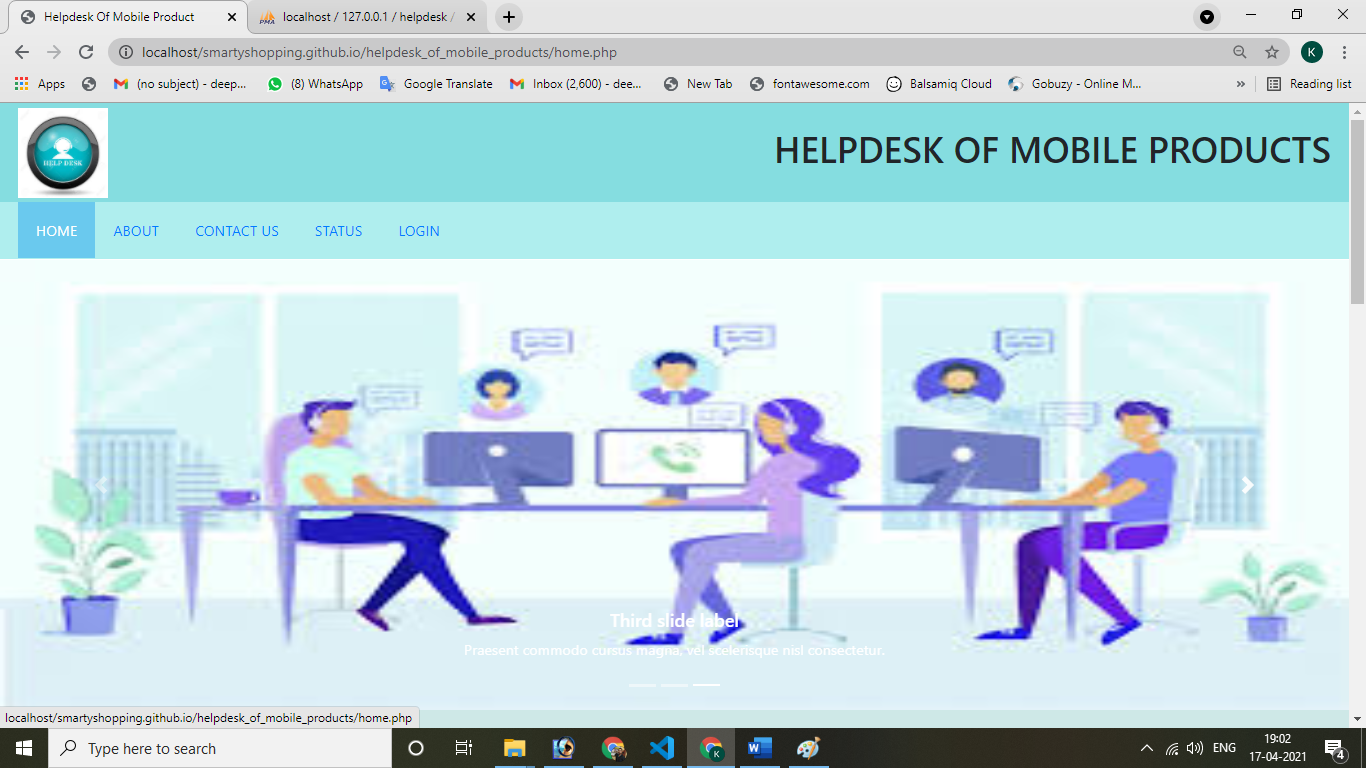
The operating-environment costs are marginal. The less time involved also helped in its economical feasibility. It was observed that the organization has already using computers for other purpose, so that there is no additional cost to be incurred for adding this system to its computers.

The backend required for storing other details is also the same database that is Sql. The computers in the organization are highly sophisticated and don’t needs extra components to load the software. Hence the organization can implement the new system without any additional expenditure. Hence, it is economically feasible

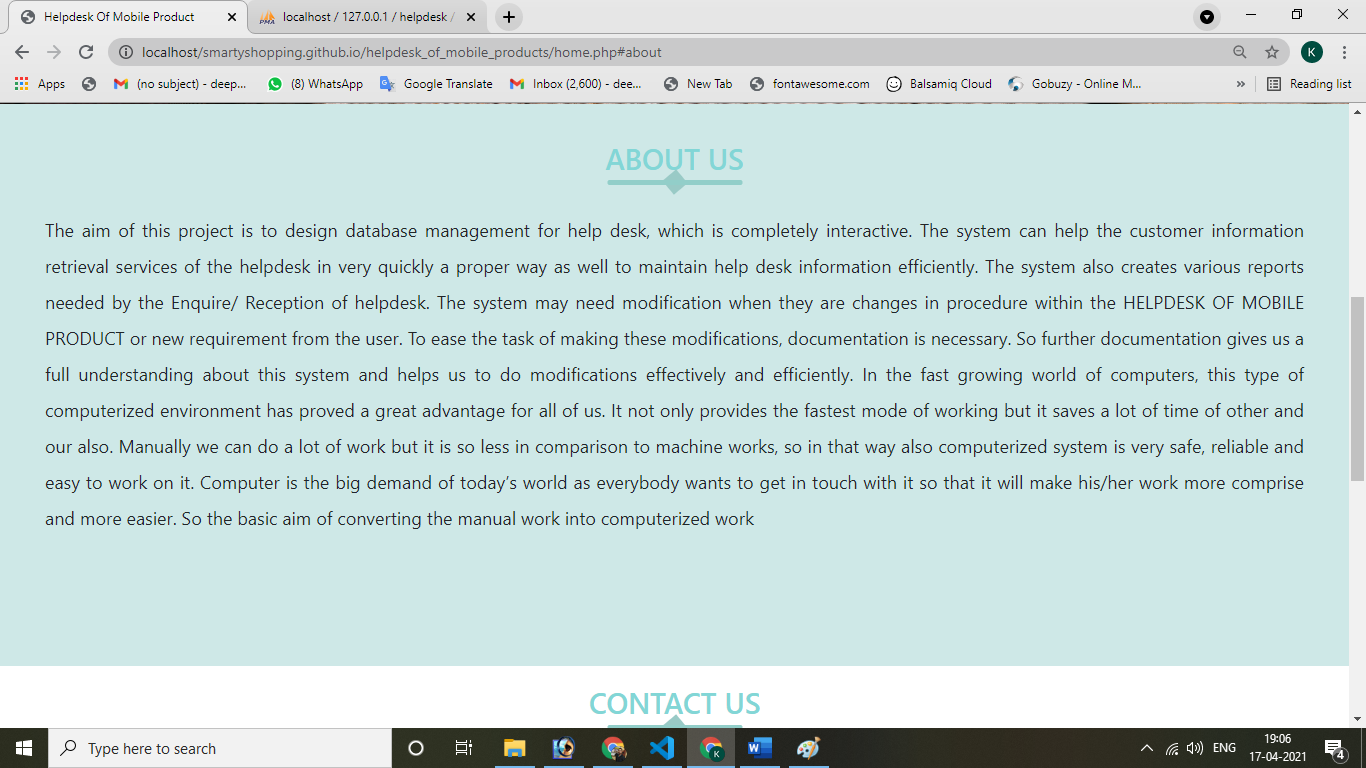
The result of the feasibility study is a formal proposal. This is simply report-a formal document detailing the nature and the scope of the proposed solution. The proposals summarize what is known and what is going to be done. Three key considerations are involved in the feasibility analysis: economic, technical and operational behavior.

**USER INTERFACE’S DESIGN**

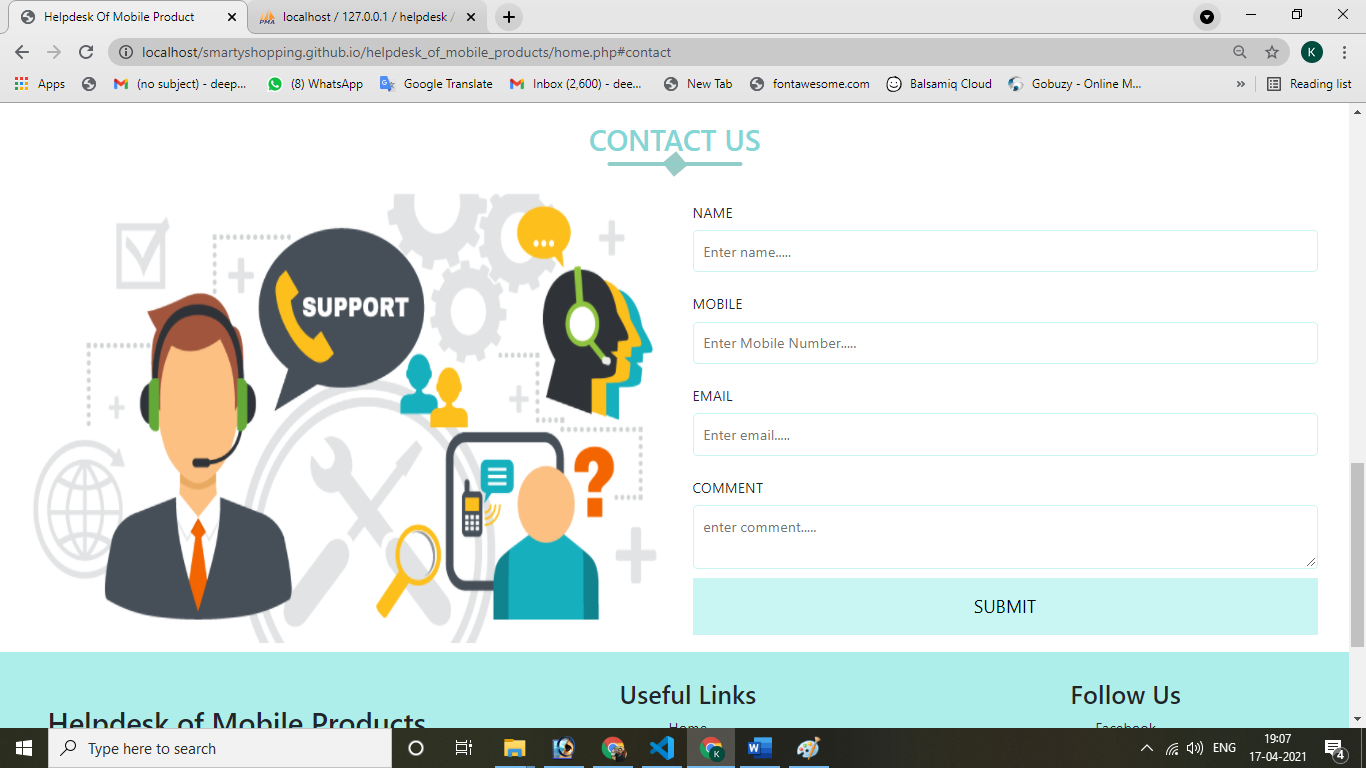
**HOME PAGE**



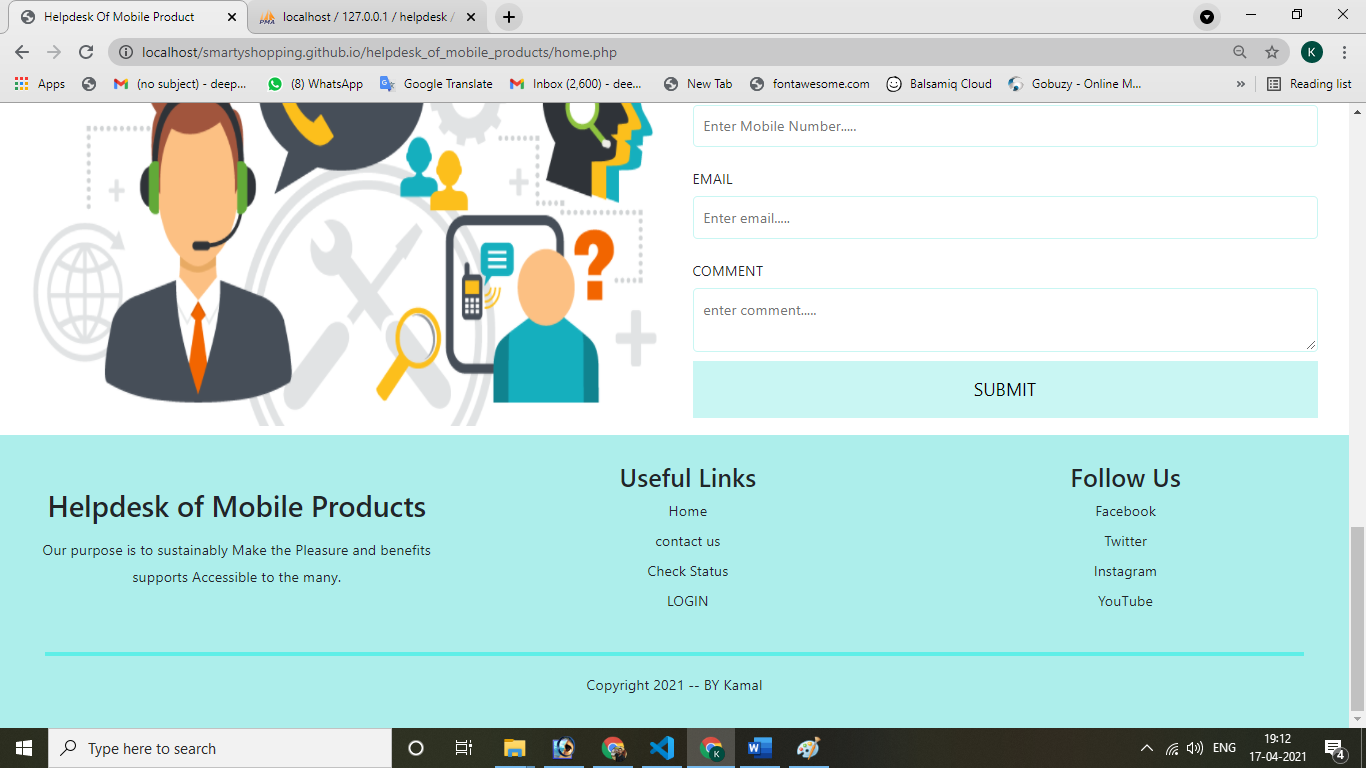
**ABOUT US PAGE**



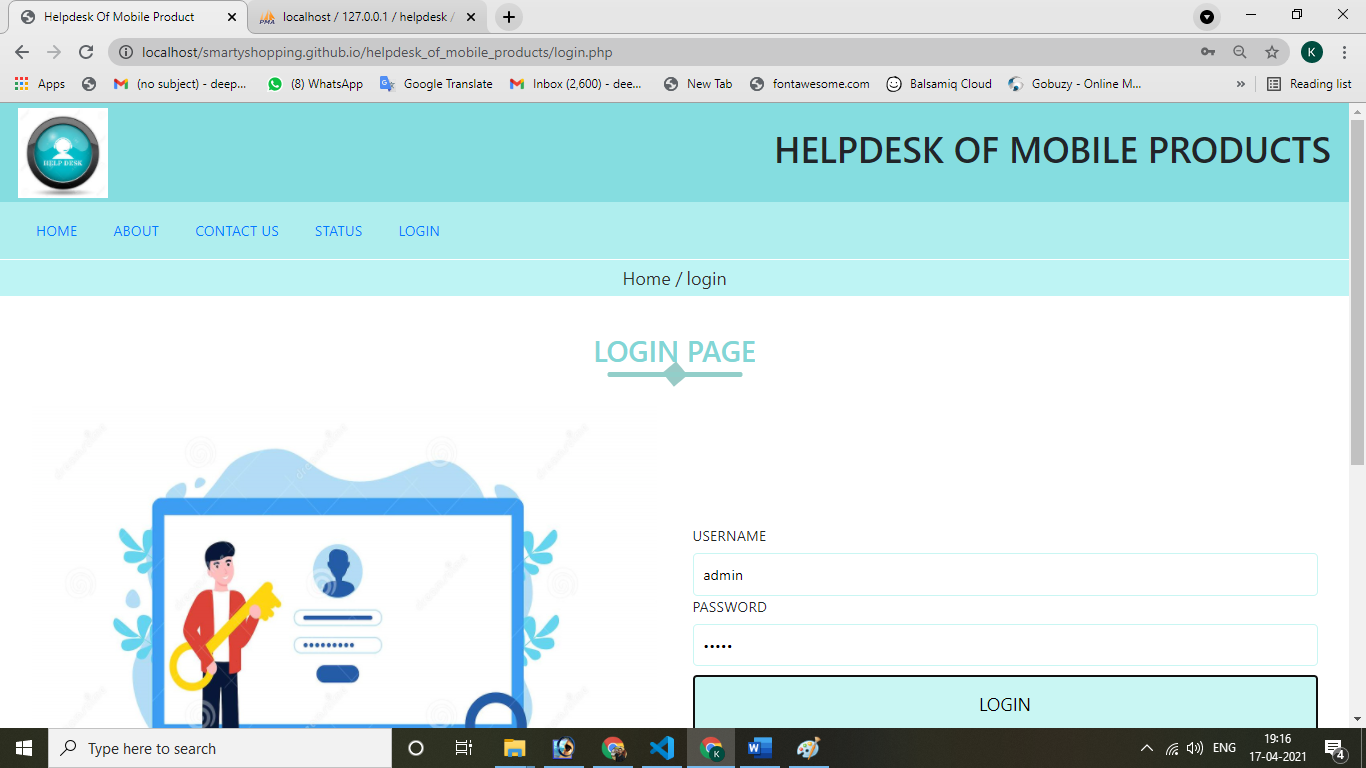
**CONTACT US PAGE**



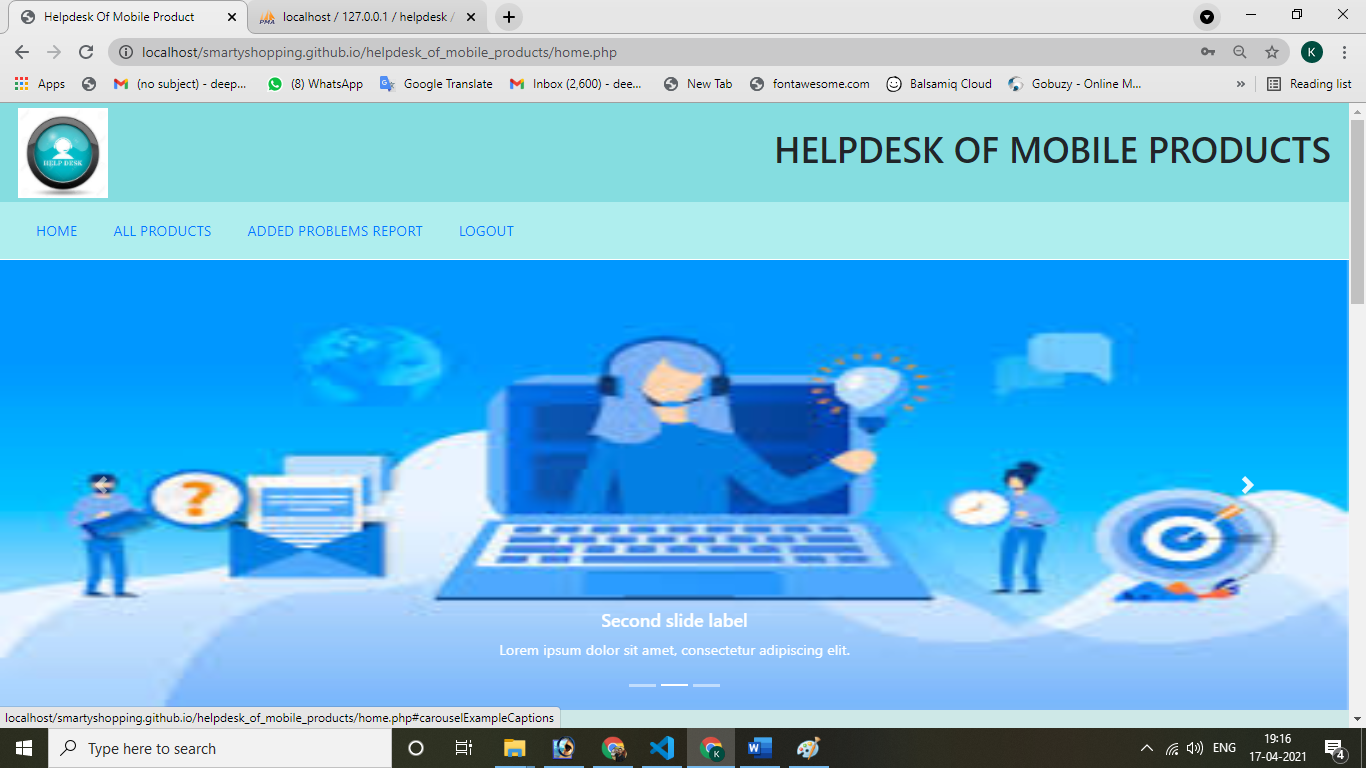
**FOOTER**



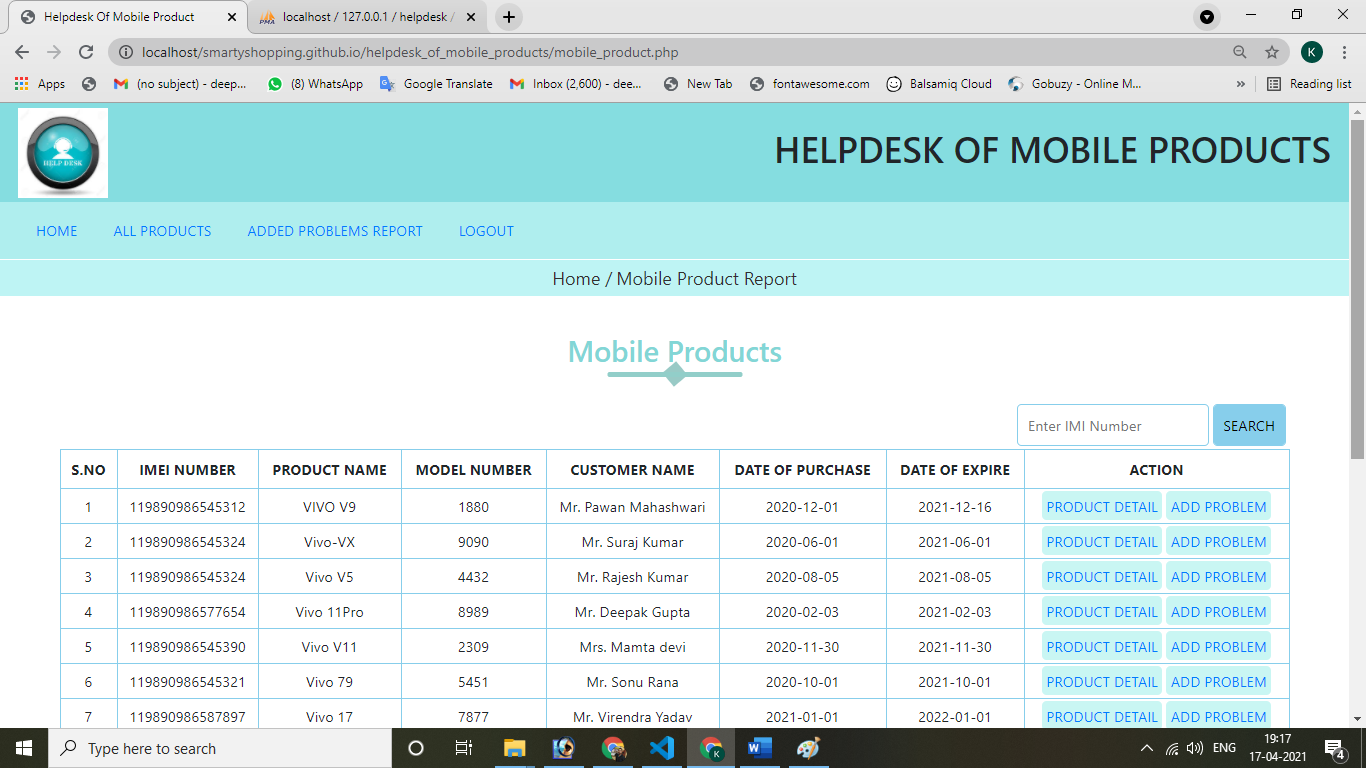
**LOGIN PAGE**



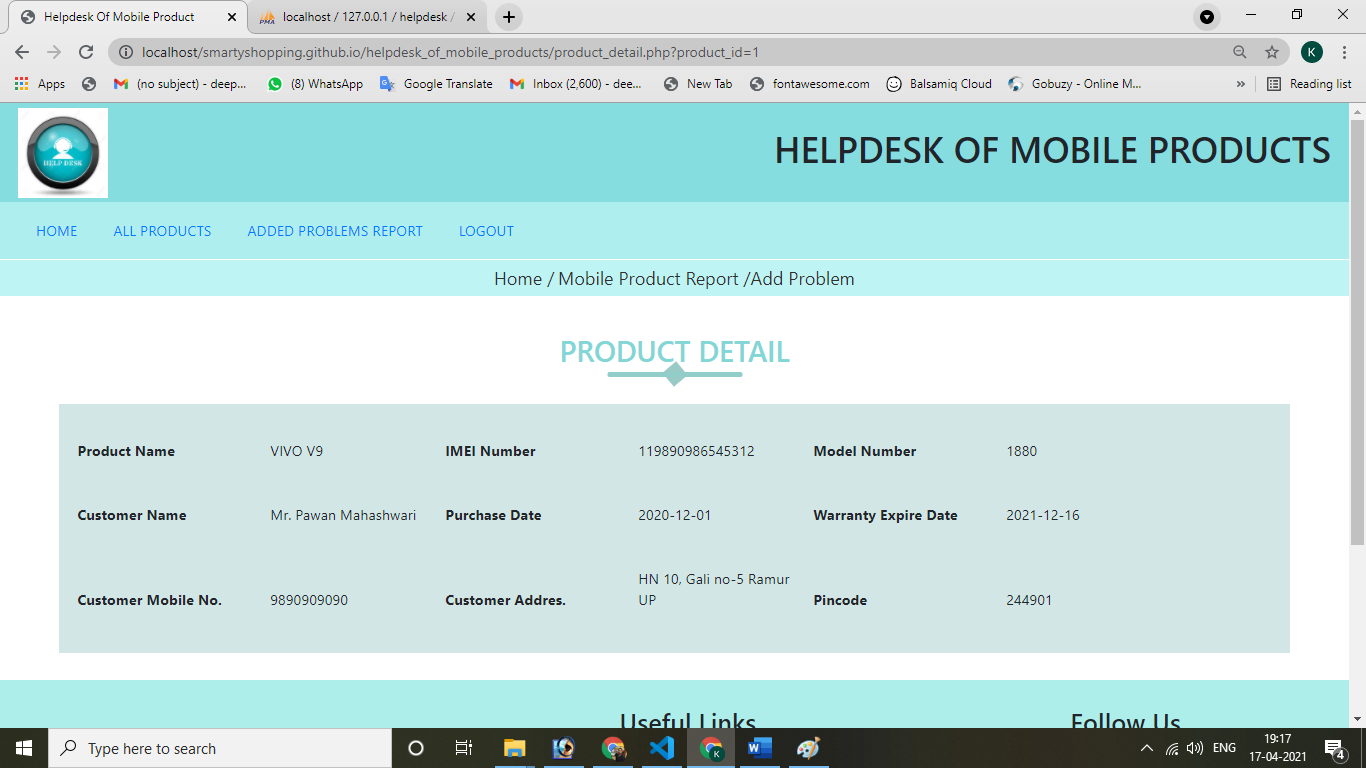
**AFTER LOGIN PAGE(HOME PAGE)**



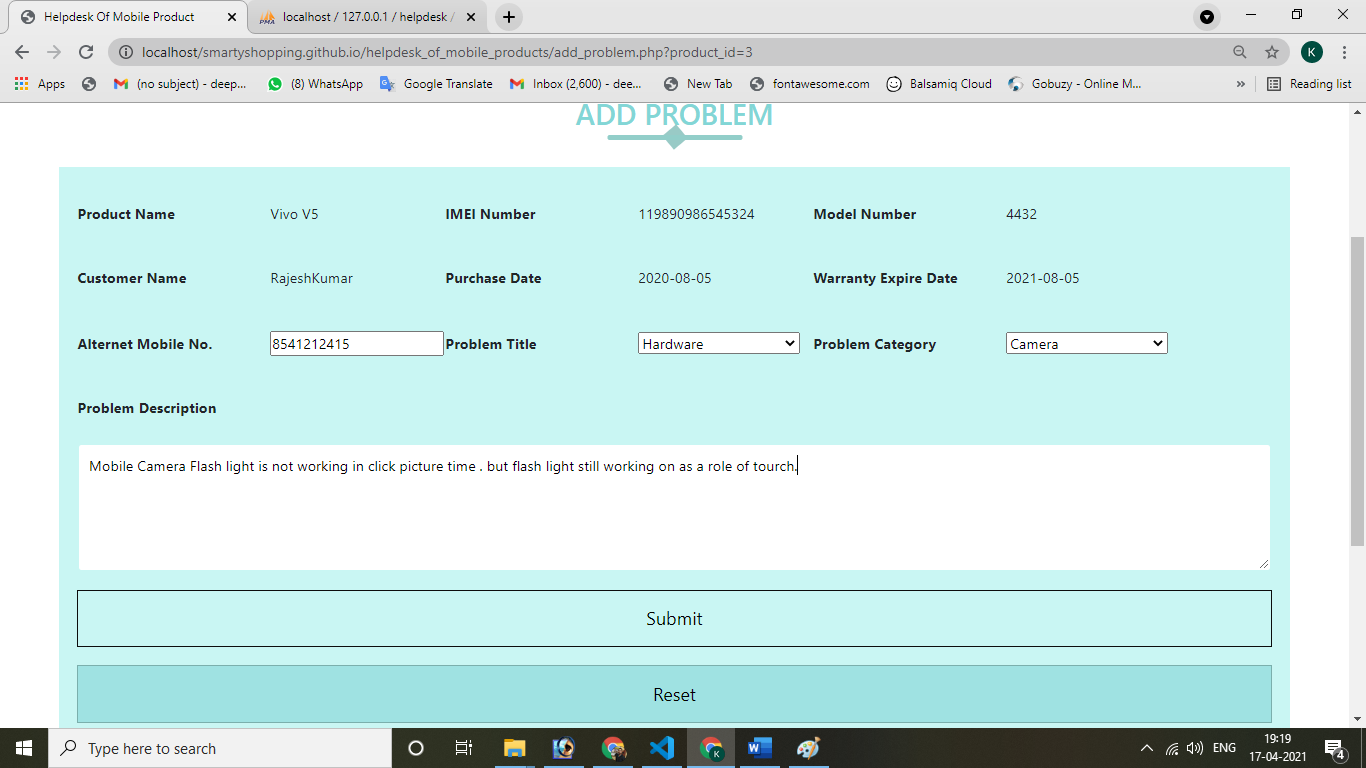
**MOBILE PRODUCT REPORT PAGE**



**PRODUCT DETAIL PAGE**



**ADD PROBLEM PAGE**



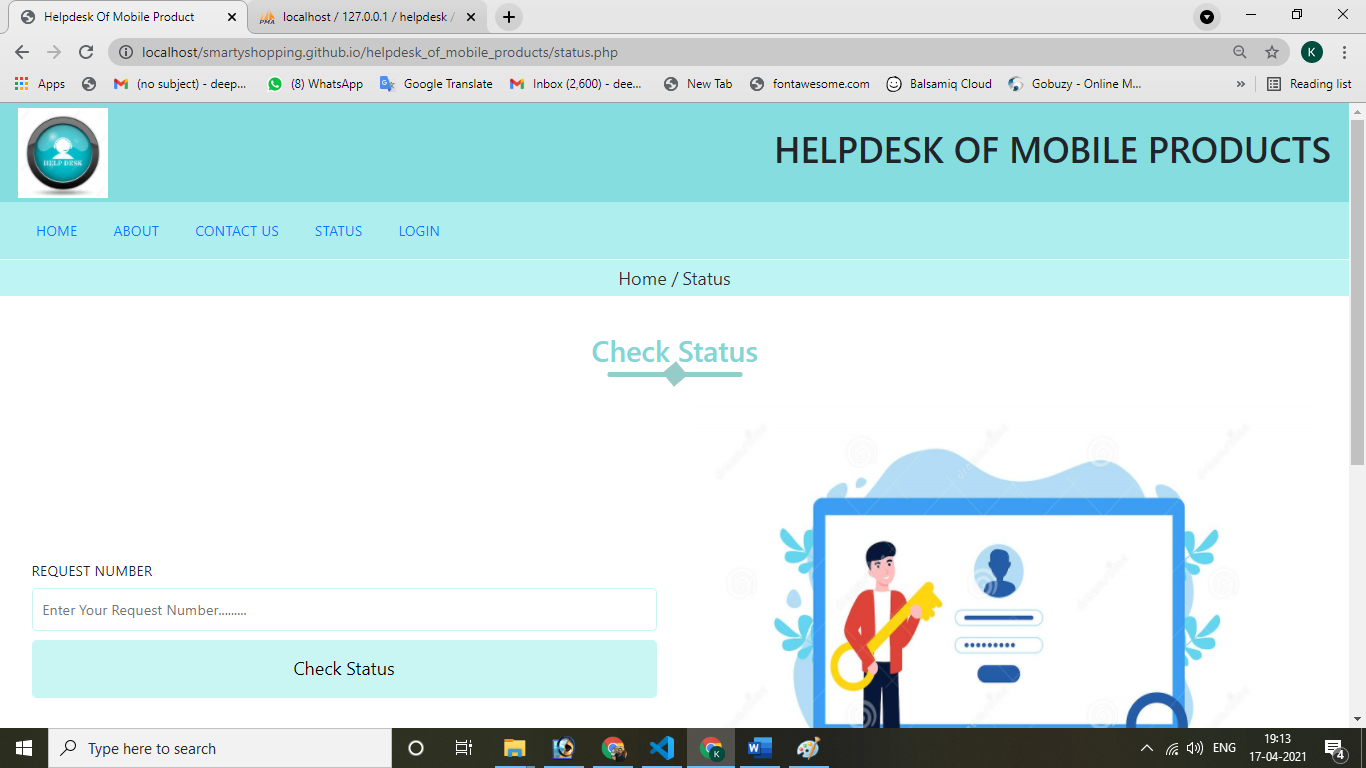
**ADDED PROBLEMS REPORT PAGE**



**UPDATE STATUS PAGE**



**CHECK STATUS PAGE**



**CHECKED STATUS PAGE**



**CODING**

HEADER.PHP

<?php

require('connection\_inc.php');

require('function\_inc.php');

?>

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Helpdesk Of Mobile Product</title>

    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

    <link rel="stylesheet" href="style/style.css">

    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css" integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T" crossorigin="anonymous">

    <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo" crossorigin="anonymous"></script>

    <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js" integrity="sha384-UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHNDz0W1" crossorigin="anonymous"></script>

    <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js" integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM" crossorigin="anonymous"></script>

    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.2/css/all.min.css" integrity="sha512-HK5fgLBL+xu6dm/Ii3z4xhlSUyZgTT9tuc/hSrtw6uzJOvgRr2a9jyxxT1ely+B+xFAmJKVSTbpM/CuL7qxO8w==" crossorigin="anonymous" />

</head>

<body>

    <header>

        <div class="head">

            <img src="media/logo.jpg" alt="logo">

            <h1>Helpdesk of Mobile Products</h1>

        </div>

        <nav>

            <ul>

            <?php

                if(!isset($\_SESSION['ADMIN\_LOGIN'])){

            ?>

                <li><a href="home.php">HOME</a></li>

                <li><a href="home.php#about">ABOUT</a></li>

                <li><a href="home.php#contact">CONTACT US</a></li>

                <li><a href="status.php">STATUS</a></li>

                <li><a href="login.php">LOGIN</a></li>

            <?php }else{

                 ?>

                 <li><a href="home.php">HOME</a></li>

                 <li><a href="mobile\_product.php">ALL PRODUCTS</a></li>

                 <li><a href="added\_problems.php">ADDED PROBLEMS REPORT</a></li>

                 <li><a href="logout.php">LOGOUT</a></li>

                <?php

                }

                ?>

            </ul>

        </nav>

    </header>

</body>

FOOTER.PHP

<footer>

            <div class="row">

                <div class="footer-col-1">

                    <h2>Helpdesk of Mobile Products</h2>

                    <p>Our purpose is to sustainably Make the Pleasure and benefits supports

                        Accessible to the many.

                    </p>

                </div>

                <div class="footer-col-2">

                    <h3>Useful Links</h3>

                    <ul>

                        <li>Home</li>

                        <!-- <li>About Us</li> -->

                        <li>contact us</li>

                        <li>Check Status</li>

                        <li>LOGIN</li>

                    </ul>

                </div>

                <div class="footer-col-3">

                    <h3>Follow Us</h3>

                    <ul>

                        <li>Facebook</li>

                        <li>Twitter</li>

                        <li>Instagram</li>

                        <li>YouTube</li>

                    </ul>

                </div>

            </div>

            <hr>

            <p class="copyright">Copyright 2021 -- BY Kamal</p>

    </footer>

    <script src="js/main.js"></script>

CONNECTION\_INC.PHP

<?php

session\_start();

$con=mysqli\_connect("localhost","root","","helpdesk");

?>

FUNCTION\_INC.PHP

<?php

function pr($arr){

    echo "<pre>";

    print\_r($arr);

}

function prx($arr){

    echo "<pre>";

    print\_r($arr);

    echo "<pre>";

    die();

}

function get\_safe\_value($con,$str){

    if($str!=''){

        $str=trim($str);

        return mysqli\_real\_escape\_string($con,$str);

HOME.PHP

<?php

        require('header.php');

?>

  <!-- -------------------------carasoul----------------------------- -->

  <div class="bd-example">

        <div id="carouselExampleCaptions" class="carousel slide" data-ride="carousel">

          <ol class="carousel-indicators">

            <li data-target="#carouselExampleCaptions" data-slide-to="0" class="active"></li>

            <li data-target="#carouselExampleCaptions" data-slide-to="1"></li>

            <li data-target="#carouselExampleCaptions" data-slide-to="2"></li>

          </ol>

          <div class="carousel-inner">

            <div class="carousel-item active">

              <img src="media/banner9.jpg" class="d-block w-100" alt="...">

              <div class="carousel-caption d-none d-md-block">

                <h5>First slide label</h5>

                <p>Nulla vitae elit libero, a pharetra augue mollis interdum.</p>

              </div>

            </div>

            <div class="carousel-item">

              <img src="media/banner2.jpg" class="d-block w-100" alt="...">

              <div class="carousel-caption d-none d-md-block">

                <h5>Second slide label</h5>

                <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>

              </div>

            </div>

            <div class="carousel-item">

              <img src="media/banner3.jpg" class="d-block w-100" alt="...">

              <div class="carousel-caption d-none d-md-block">

                <h5>Third slide label</h5>

                <p>Praesent commodo cursus magna, vel scelerisque nisl consectetur.</p>

              </div>

            </div>

          </div>

          <a class="carousel-control-prev" href="#carouselExampleCaptions" role="button" data-slide="prev">

            <span class="carousel-control-prev-icon" aria-hidden="true"></span>

            <span class="sr-only">Previous</span>

          </a>

          <a class="carousel-control-next" href="#carouselExampleCaptions" role="button" data-slide="next">

            <span class="carousel-control-next-icon" aria-hidden="true"></span>

            <span class="sr-only">Next</span>

          </a>

        </div>

   </div>

<!-- ----------------------About Section----------------------- -->

   <div class="about" id="about">

     <h2 class="title">ABOUT US</h2>

     <P>

      The aim of this project is to design database management for help desk, which is

      completely interactive. The system can help the customer information retrieval

      services of the helpdesk in very quickly a proper way as well to maintain help desk

      information efficiently. The system also creates various reports needed by the

      Enquire/ Reception of helpdesk.

      The system may need modification when they are changes in procedure within the

      HELPDESK OF MOBILE PRODUCT or new requirement from the user. To ease

      the task of making these modifications, documentation is necessary. So further

      documentation gives us a full understanding about this system and helps us to do

      modifications effectively and efficiently.

      In the fast growing world of computers, this type of computerized environment has

      proved a great advantage for all of us. It not only provides the fastest mode of

      working but it saves a lot of time of other and our also. Manually we can do a lot

      of work but it is so less in comparison to machine works, so in that way also

      computerized system is very safe, reliable and easy to work on it. Computer is the

      big demand of today’s world as everybody wants to get in touch with it so that it

      will make his/her work more comprise and more easier. So the basic aim of

      converting the manual work into computerized work

     </P>

   </div>

   <!-- -----------------------------contact Us------------------------ -->

<div class="contact" id="contact">

    <h2 class="title">CONTACT US</h2>

    <div class="row">

        <div class="colu-2">

            <img src="media/contact.png" alt="contact page">

        </div>

        <div class="colu-2">

            <form action="" id="contact">

                <label for="name">NAME</label>

                <input type="text" name="name" id="name" placeholder="Enter name....."><br>

                <span class="error\_msg" id="error\_name" ></span><br>

                <label for="mobile">MOBILE</label>

                <input type="text" name="mobile" id="mobile" placeholder="Enter Mobile Number....."><br>

                <span  class="error\_msg" id="error\_mobile"></span><br>

                <label for="email">EMAIL</label>

                <input type="text" name="email" id="email" placeholder="Enter email....."><br>

                <span  class="error\_msg" id="error\_email"></span>  <br>

                <label for="comment">COMMENT</label>

                <textarea name="comment" id="query" cols="40" rows="2" placeholder="enter comment....."></textarea>

                <span class="error\_msg" id="error\_query"></span><br>

                <button type="button" onclick="send\_message()">SUBMIT</button><br>

                <span  class="error\_msg" id="error\_button"></span>

            </form>

        </div>

    </div>

</div>

<?php

    require('footer.php');

?>

MOBILE\_PRODUCT.PHP

<?php

        require('header.php');

        $sql="select product.\*, customer.\* from product,customer where product.customer\_id=customer.customer\_id";

        $res=mysqli\_query($con,$sql);

        $count=mysqli\_num\_rows($res);

?>

  <!-- -------------------------path seciton----------------------------- -->

  <div class="path">

          <span>

            <a href="index.html">Home /</a>  <a href="#">Mobile Product Report</a>

          </span>

    </div>

   <!-- -----------------------------Mobile Product Report------------------------ -->

   <div class=" login mobile\_product" >

     <h2 class="title">Mobile Products</h2>

      <div class="search">

        <form action="search\_imi" method="get">

            <input type="text" name="imi" Placeholder="Enter IMI Number">

            <input type="submit" name="search" value="SEARCH">

        </form>

      </div>

     <table>

       <tr>

          <th>S.NO</th>

          <th>IMEI NUMBER</th>

          <th>PRODUCT NAME</th>

          <th>MODEL NUMBER</th>

          <th>CUSTOMER NAME</th>

          <th>DATE OF PURCHASE</th>

          <th>DATE OF EXPIRE</th>

          <th>ACTION</th>

       </tr>

       <?php

                if($count>0){

                $i=1;

                while($row=mysqli\_fetch\_assoc($res)){

                ?>

       <tr>

          <td><?php echo $i;?></td>

          <td><?php echo $row['imei\_no'];?></td>

          <td><?php echo $row['product\_name'];?></td>

          <td><?php echo $row['model\_no'];?></td>

          <td><?php echo $row['title']." ".$row['first\_name']." ".$row['last\_name']; ?></td>

          <td><?php echo $row['date\_of\_purchase'];?></td>

          <td><?php echo $row['expire\_date'];?></td>

          <td>

              <a class="view" href="product\_detail.php?product\_id=<?php echo $row['product\_id']?>">PRODUCT DETAIL</a>

              <a class="add\_problem" href="add\_problem.php?product\_id=<?php echo $row['product\_id']?>">ADD PROBLEM</a>

          </td>

       </tr>

       <?php

                  $i++;

                  }  } else {

                  ?>

                  <tr>

                     <td colspan="8" align="center">No Records Found!</td>

                  </tr>

                  <?php

                     }

                  ?>

     </table>

   </div>

<?php

    require('footer.php');

?>

PRODUCT\_DETAIL.PHP

<?php

        require('header.php');

        if(isset($\_GET['product\_id']) && $\_GET['product\_id']!=''){

          $product\_id=get\_safe\_value($con,$\_GET['product\_id']);

          $sql="select product.\*, customer.\* from product,customer where product.customer\_id=customer.customer\_id and product.product\_id='$product\_id'";

          $res=mysqli\_query($con,$sql);

          $count=mysqli\_num\_rows($res);

          if($count>0){

            $row=mysqli\_fetch\_assoc($res);

            $product\_id=$row['product\_id'];

            $product\_name=$row['product\_name'];

            $imei\_no=$row['imei\_no'];

            $model\_no=$row['model\_no'];

            $date\_of\_purchase=$row['date\_of\_purchase'];

            $expire\_date=$row['expire\_date'];

            $customer\_name= $row['title'] ." ".$row['first\_name']." " . $row['last\_name'];

            $customer\_mobile=$row['phone\_number'];

            $address=$row['street']." ".$row['city']." ".$row['state'];

            $pincode=$row['pincode'];

          }else{

              header('location:home.php');

              die();

          }

      }

?>

  <!-- -------------------------path seciton----------------------------- -->

  <div class="path">

          <span>

            <a href="home.php">Home /</a>  <a href="mobile\_product.php">Mobile Product Report /</a><a href="#">Add Problem </a>

          </span>

    </div>

   <!-- -----------------------------pRODCUT DETAIL ------------------------ -->

   <div class="login add\_prob" >

     <h2 class="title"> PRODUCT DETAIL</h2>

     <div class="product\_detail problem\_form">

        <label for="">Product Name</label>

        <span name='name'><?php echo $product\_name;?></span>

        <label for="">IMEI Number</label>

        <input class="product\_id" type="number" name="product\_id" value="<?php echo $product\_id;?>">

        <span n><?php echo $imei\_no;?></span>

        <label for="">Model Number</label>

        <span><?php echo $model\_no;?></span>

        <br>

        <label for="">Customer Name</label>

        <span><?php echo $customer\_name; ?></span>

        <label for="">Purchase Date</label>

        <span><?php echo $date\_of\_purchase;?></span>

        <label for="">Warranty Expire Date</label>

        <span><?php echo $expire\_date;?></span>

        <br>

        <label for="">Customer Mobile No.</label>

        <span><?php echo $customer\_mobile;?></span>

        <label for="">Customer Addres.</label>

        <span><?php echo $address;?></span>

        <label for="">Pincode</label>

        <span><?php echo $pincode;?></span>

        </div>

  </div>

<?php

    require('footer.php');

?>

End Class

ADD\_PROBLEM.PHP

<?php

        require('header.php');

        if(isset($\_GET['product\_id']) && $\_GET['product\_id']!=''){

          $product\_id=get\_safe\_value($con,$\_GET['product\_id']);

          $sql="select product.\*, customer.\* from product,customer where product.customer\_id=customer.customer\_id and product.product\_id='$product\_id'";

          $res=mysqli\_query($con,$sql);

          $count=mysqli\_num\_rows($res);

          if($count>0){

            $row=mysqli\_fetch\_assoc($res);

            $product\_id=$row['product\_id'];

            $product\_name=$row['product\_name'];

            $imei\_no=$row['imei\_no'];

            $model\_no=$row['model\_no'];

            $customer\_name= $row['first\_name'].$row['last\_name'];

            $date\_of\_purchase=$row['date\_of\_purchase'];

            $expire\_date=$row['expire\_date'];

          }else{

              header('location:home.php');

              die();

          }

      }

        if(isset($\_GET['submit'])){

          $product\_id=get\_safe\_value($con,$\_GET['product\_id']);

          $alt\_number=get\_safe\_value($con,$\_GET['alt\_number']);

          $problem\_title=get\_safe\_value($con,$\_GET['problem\_tit']);

          $problem\_category=get\_safe\_value($con,$\_GET['problem\_cat']);

          $description=get\_safe\_value($con,$\_GET['description']);

          $status=1;

          $request\_no="REQ".rand(10000,100000);

          date\_default\_timezone\_set('Asia/Kolkata');

          $currentTime = date( 'Y-m-d H:i:s');

          mysqli\_query($con,"insert into problem (product\_id,problem\_title\_id,problem\_category\_id,problem\_description,problem\_status\_id,curr\_time,request\_no,deliver,alt\_number) values ('$product\_id','$problem\_title','$problem\_category','$description','$status','$currentTime','$request\_no','0','$alt\_number')");

          // echo "insert into problem (product\_id,problem\_title\_id,problem\_category\_id,problem\_description,problem\_status\_id,curr\_time) values ('$product\_id','$problem\_title','$problem\_category','$description','$status','$currentTime')";

          echo "your Request has been added Successfull and your Request Number is ".$request\_no;

        }

?>

  <!-- -------------------------path seciton----------------------------- -->

  <div class="path">

          <span>

            <a href="index.html">Home /</a>  <a href="mobile\_product.php">Mobile Product Report /</a><a href="#">Add Problem </a>

          </span>

    </div>

   <!-- -----------------------------Add Problem ------------------------ -->

   <div class="login add\_prob" >

     <h2 class="title">ADD PROBLEM</h2>

     <div class="problem\_form">

     <form action="" method="get">

        <label for="">Product Name</label>

        <span name='name'><?php echo $product\_name;?></span>

        <label for="">IMEI Number</label>

        <input class="product\_id" type="number" name="product\_id" value="<?php echo $product\_id;?>">

        <span name="imei\_no" value="<?php echo $imei\_no;?>"><?php echo $imei\_no;?></span>

        <label for="">Model Number</label>

        <span><?php echo $model\_no;?></span>

        <br>

        <label for="">Customer Name</label>

        <span><?php echo $customer\_name; ?></span>

        <label for="">Purchase Date</label>

        <span><?php echo $date\_of\_purchase;?></span>

        <label for="">Warranty Expire Date</label>

        <span><?php echo $expire\_date;?></span>

        <br>

        <label for="">Alternet Mobile No.</label>

        <span><input type="number"  name="alt\_number" placeholder="Enter Mobile Number" required></span>

        <label for="">Problem Title</label>

        <span>

          <select name="problem\_tit" id="problem\_title" required>

              <option value="">Select Problem Title...</option>

                <?php

                  $sql=mysqli\_query($con,"select \* from problem\_title");

                  while($row=mysqli\_fetch\_assoc($sql)){

                      echo "<option value=".$row['id'].">".$row['name']."</option>";

                      }

                ?>

          </select>

        </span>

        <label for="">Problem Category</label>

        <span>

          <select name="problem\_cat" id="problem\_category" required>

            <option value="">Select Problem Category.....</option>

          </select>

        </span>  <br>

        <label for="">Problem Description</label>

        <textarea name="description" id="" cols="30" rows="5" required></textarea>

          <input type="submit" value="Submit" name="submit">

          <input type="reset" value="Reset">

       </form>

    </div>

  </div>

<?php

    require('footer.php');

?>

ADDED\_PROBLEMS.PHP

<?php

        require('header.php');

        $sql="select product.\*, customer.\*,problem.\*,problem\_title.name,problem\_category.cat\_name, problem\_status.status\_name from product,customer,problem,problem\_title,problem\_category, problem\_status where product.customer\_id=customer.customer\_id and product.product\_id=problem.product\_id and problem.problem\_title\_id=problem\_title.id and problem.problem\_category\_id=problem\_category.id and problem.problem\_status\_id=problem\_status.id";

        $res=mysqli\_query($con,$sql);

        $count=mysqli\_num\_rows($res);

        if(isset($\_GET['operation']) && $\_GET['operation']!=''){

         $request\_no=get\_safe\_value($con,$\_GET['request\_no']);

         $operation=get\_safe\_value($con,$\_GET['operation']);

         date\_default\_timezone\_set('Asia/Kolkata');

          $currentTime = date( 'Y-m-d H:i:s');

         $ress=mysqli\_query($con,"select problem\_status\_id from problem where request\_no='$request\_no' and problem\_status\_id='3'");

         $countt=mysqli\_num\_rows($ress);

         if($countt<=0){

         echo '<script> alert("Please do problem status success then you can deliver ");</script>';

         }else{

         if($operation =='deliver'){

             $deliver='1';

             $update\_sql="update problem set deliver='$deliver',delivery\_date='$currentTime' where request\_no ='$request\_no'";

             mysqli\_query($con,$update\_sql);

             }else{

               $deliver='0';

               $update\_sql="update problem set deliver='$deliver',delivery\_date='$currentTime' where request\_no ='$request\_no'";

               mysqli\_query($con,$update\_sql);

             }

             header("location:added\_problems.php");

         }

      }

?>

  <!-- -------------------------path seciton----------------------------- -->

  <div class="path">

          <span>

            <a href="index.html">Home /</a>  <a href="#">Added Problems Report</a>

          </span>

    </div>

   <!-- -----------------------------Mobile Product Report------------------------ -->

   <div class=" login mobile\_product added\_problems" >

     <h2 class="title">Added Problems</h2>

      <div class="search">

        <form action="search\_imi" method="get">

            <input type="text" name="imi" Placeholder="Enter IMI Number">

            <input type="submit" name="search" value="SEARCH">

        </form>

      </div>

     <table>

       <tr>

          <th>S.NO</th>

          <th>IMEI NO.</th>

          <th>REQUEST NO.</th>

          <th>CUSTOMER NAME</th>

          <th>PRODUCT NAME</th>

          <th>MODEL NUMBER</th>

          <th>PROBLEM TITLE</th>

          <th>PROBLEM CATEGORY</th>

          <th>ADDED PROBLEM DATE</th>

          <th>STATUS</th>

          <th>ACTION</th>

       </tr>

       <?php

                if($count>0){

                $i=1;

                while($row=mysqli\_fetch\_assoc($res)){

                ?>

       <tr>

          <td><?php echo $i;?></td>

          <td><?php echo $row['imei\_no'];?></td>

          <td><?php echo $row['request\_no'];?></td>

          <td><?php echo $row['title']." ".$row['first\_name']." ".$row['last\_name']; ?></td>

          <td><?php echo $row['product\_name'];?></td>

          <td><?php echo $row['model\_no'];?></td>

          <td><?php echo $row['name'];?></td>

          <td><?php echo $row['cat\_name'];?></td>

          <td><?php echo $row['curr\_time'];?></td>

          <td><?php echo $row['status\_name'];?></td>

          <td>

              <span><a class="view" href="update\_status.php?request\_no=<?php echo $row['request\_no']?>">Update Status</a></span>

              <?php

                    if($row['deliver']==1){

                        echo "<span class='btn active'><a href='?operation=not\_deliver&request\_no=".$row['request\_no']."'>Deliver</a></span>";

                    }else{

                        echo"<span class='btn deactive'><a href='?operation=deliver&request\_no=".$row['request\_no']."' >Not Deliver</a></span>";

                    }

               ?>

          </td>

       </tr>

       <?php

                  $i++;

                  }  } else {

                  ?>

                  <tr>

                     <td colspan="11" align="center">No Records Found!</td>

                  </tr>

                  <?php

                     }

                  ?>

     </table>

   </div>

<?php

    require('footer.php');

?>

UPDATE\_STATUS.PHP

<?php

        require('header.php');

        if(isset($\_GET['request\_no']) && $\_GET['request\_no']!=''){

          $request\_no=get\_safe\_value($con,$\_GET['request\_no']);

          $sql="select product.\*, customer.\*,problem.\*,problem\_title.name,problem\_category.cat\_name, problem\_status.status\_name from product,customer,problem,problem\_title,problem\_category, problem\_status where product.customer\_id=customer.customer\_id and product.product\_id=problem.product\_id and problem.problem\_title\_id=problem\_title.id and problem.problem\_category\_id=problem\_category.id and problem.problem\_status\_id=problem\_status.id and problem.request\_no='$request\_no'";

          $res=mysqli\_query($con,$sql);

          $count=mysqli\_num\_rows($res);

          if($count>0){

            $row=mysqli\_fetch\_assoc($res);

            $request\_no=$row['request\_no'];

            $product\_id=$row['product\_id'];

            $product\_name=$row['product\_name'];

            $imei\_no=$row['imei\_no'];

            $model\_no=$row['model\_no'];

            $customer\_name= $row['title']." ".$row['first\_name']." ".$row['last\_name'];

            $problem\_title=$row['name'];

            $problem\_category=$row['cat\_name'];

            $description=$row['problem\_description'];

            $curr\_date=$row['curr\_time'];

            $date\_of\_purchase=$row['date\_of\_purchase'];

            $expire\_date=$row['expire\_date'];

            $alt\_number=$row['alt\_number'];

          }else{

              header('location:home.php');

              die();

          }

      }

        if(isset($\_GET['submit'])){

          $request\_no=get\_safe\_value($con,$\_GET['request\_no']);

          $problem\_status=get\_safe\_value($con,$\_GET['problem\_status']);

          date\_default\_timezone\_set('Asia/Kolkata');

          $currentTime = date( 'Y-m-d H:i:s');

          mysqli\_query($con,"update problem set problem\_status\_id='$problem\_status' where request\_no='$request\_no'");

          echo '<script> alert("Your Problem Status has been Updated");</script>';

          // header('location:added\_problems.php');

          // echo "Your Problem Status has been Updated";

        }

?>

  <!-- -------------------------path seciton----------------------------- -->

  <div class="path">

          <span>

            <a href="home.php">Home /</a>  <a href="added\_problems.php">Added Problem /</a><a href="#">Update Status </a>

          </span>

    </div>

   <!-- -----------------------------Update Status ------------------------ -->

   <div class="login add\_prob" >

     <h2 class="title">UPDATE STATUS</h2>

     <div class="problem\_form">

     <form action="" method="get">

        <label for="">Request Number</label>

        <span name='name'><?php echo $request\_no;?></span>

        <input class="product\_id" type="text" name="request\_no" value="<?php echo $request\_no;?>">

        <label for="">Product Name</label>

        <span name='name'><?php echo $product\_name;?></span>

        <label for="">IMEI Number</label>

        <span><?php echo $imei\_no;?></span>

        <label for="">Model Number</label>

        <span><?php echo $model\_no;?></span>

        <label for="">Customer Name</label>

        <span><?php echo $customer\_name; ?></span>

        <label for="">Purchase Date</label>

        <span><?php echo $date\_of\_purchase;?></span>

        <label for="">Warranty Expire Date</label>

        <span><?php echo $expire\_date;?></span>

        <label for="">Problem Title</label>

        <span><?php echo $problem\_title; ?></span>

        <label for="">Porblem Category</label>

        <span><?php echo $problem\_category;?></span>

        <label for="">Added Problem Date</label>

        <span><?php echo $curr\_date;?></span>

        <label for="">Alternet Number</label>

        <span><?php echo $alt\_number;?></span>

        <label for="">Description</label>

        <span><?php echo $description;?></span>

        <br>

        <label for="">Status</label>

        <span>

          <select name="problem\_status" id="problem\_status" required>

              <option value="">Select Problem Status...</option>

                <?php

                  $sql=mysqli\_query($con,"select \* from problem\_status");

                  while($row=mysqli\_fetch\_assoc($sql)){

                      echo "<option value=".$row['id'].">".$row['status\_name']."</option>";

                      }

                ?>

          </select>

        </span>

          <br>

          <input type="submit" value="Submit" name="submit">

          <input type="reset" value="Reset">

       </form>

    </div>

  </div>

<?php

    require('footer.php');

?>

STATUS.PHP

<?php

        require('header.php');

?>

           <!-- -------------------------path seciton----------------------------- -->

    <div class="path">

          <span>

            <a href="index.html">Home /</a>  <a href="#">Status</a>

          </span>

    </div>

   <!-- -----------------------------Login------------------------ -->

   <div class="login status" >

     <h2 class="title">Check Status</h2>

     <div class="row">

        <div class="colu-2">

            <form action="checked\_status.php" method="get">

                 <label for="request">REQUEST NUMBER</label><br>

                 <input type="text" name="request\_no" id="request" placeholder="Enter Your Request Number........." required>

                 <input type="submit" name="check" value="Check Status"  >

             </form>

        </div>

        <div class="colu-2">

            <img src="media/login.jpg" alt="contact page">

        </div>

    </div>

  </div>

<?php

    require('footer.php');

?>

CHECKED\_STATUS.PHP

<?php

        require('header.php');

        if(isset($\_GET['request\_no']) && $\_GET['request\_no']!=''){

          $request\_no=get\_safe\_value($con,$\_GET['request\_no']);

          $sql="select product.\*, customer.\*,problem.\*,problem\_title.name,problem\_category.cat\_name, problem\_status.status\_name from product,customer,problem,problem\_title,problem\_category, problem\_status where product.customer\_id=customer.customer\_id and product.product\_id=problem.product\_id and problem.problem\_title\_id=problem\_title.id and problem.problem\_category\_id=problem\_category.id and problem.problem\_status\_id=problem\_status.id and problem.request\_no='$request\_no'";

          $res=mysqli\_query($con,$sql);

          $count=mysqli\_num\_rows($res);

          if($count>0){

            $row=mysqli\_fetch\_assoc($res);

            $request\_no=$row['request\_no'];

            $product\_id=$row['product\_id'];

            $product\_name=$row['product\_name'];

            $imei\_no=$row['imei\_no'];

            $model\_no=$row['model\_no'];

            $customer\_name= $row['title']." ".$row['first\_name']." ".$row['last\_name'];

            $problem\_title=$row['name'];

            $problem\_category=$row['cat\_name'];

            $description=$row['problem\_description'];

            $curr\_date=$row['curr\_time'];

            $date\_of\_purchase=$row['date\_of\_purchase'];

            $expire\_date=$row['expire\_date'];

          }else{

            // header('location:status.php');

           echo '<SCRipt> alert("Please Enter Valid Request Number")</SCRipt>';

              die();

          }

      }

?>

  <!-- -------------------------path seciton----------------------------- -->

  <div class="path">

          <span>

            <a href="home.php">Home /</a>  <a href="#">Check Status</a>

          </span>

    </div>

   <!-- -----------------------------Update Status ------------------------ -->

   <div class="login add\_prob" >

     <h2 class="title">CHECK STATUS</h2>

     <div class="problem\_form">

        <label for="">Request Number</label>

        <span name='name'><?php echo $request\_no;?></span>

        <input class="product\_id" type="text" name="request\_no" value="<?php echo $request\_no;?>">

        <label for="">Product Name</label>

        <span name='name'><?php echo $product\_name;?></span>

        <label for="">IMEI Number</label>

        <span><?php echo $imei\_no;?></span>

        <label for="">Model Number</label>

        <span><?php echo $model\_no;?></span>

        <label for="">Customer Name</label>

        <span><?php echo $customer\_name; ?></span>

        <label for="">Purchase Date</label>

        <span><?php echo $date\_of\_purchase;?></span>

        <label for="">Warranty Expire Date</label>

        <span><?php echo $expire\_date;?></span>

        <label for="">Problem Title</label>

        <span><?php echo $problem\_title; ?></span>

        <label for="">Porblem Category</label>

        <span><?php echo $problem\_category;?></span>

        <label for="">Added Problem Date</label>

        <span><?php echo $curr\_date;?></span>

        <label for="">Description</label>

        <span><?php echo $description;?></span>

        <label for="">Status</label>

        <span><?php echo $row['status\_name'];?></span>

    </div>

  </div>

<?php

    require('footer.php');

?>

LOGIN.PHP

<?php

        require('header.php');

?>

  <!-- -------------------------path seciton----------------------------- -->

  <div class="path">

          <span>

            <a href="index.html">Home /</a>  <a href="#">login</a>

          </span>

    </div>

   <!-- -----------------------------Login------------------------ -->

   <div class="login" >

     <h2 class="title">LOGIN PAGE</h2>

     <div class="row">

        <div class="colu-2">

            <img src="media/login.jpg" alt="contact page">

        </div>

        <div class="colu-2">

         <form id="sign">

            <label for="username">USERNAME</label><br>

            <input type="text" name="username" id="username" placeholder="Enter Username.....">

            <span class="error\_msg" id="error\_username" ></span><br>

            <label for="password">PASSWORD</label><br>

            <input type="password" name="password" id="password" placeholder="Enter Password.....">

            <span class="error\_msg" id="error\_password" ></span><br>

            <button type="button" onclick="sign\_in()">LOGIN</button>

            <span class="error\_msg" id="error\_button" ></span><br>

          </form>

        </div>

    </div>

  </div>

<?php

    require('footer.php');

?>

CHECK\_SIGNIN.PHP

<?php

    require('connection\_inc.php');

    require('function\_inc.php');

    $email=get\_safe\_value($con,$\_POST['email']);

    $password=get\_safe\_value($con,$\_POST['password']);

    $res=mysqli\_query($con,"select \* from admin\_users where username='$email' && password='$password'");

    if(mysqli\_num\_rows($res)>0){

            $row=mysqli\_fetch\_assoc($res);

            $\_SESSION['ADMIN\_LOGIN']='yes';

            $\_SESSION['ADMIN\_NAME']=$row['name'];

            $\_SESSION['ADMIN\_ID']=$row['id'];

            echo "You have been Login Successful";

        }else {

            echo "Plese enter username and Password correct";

            }

?>

LOGOUT.PHP

<?php

require('connection\_inc.php');

unset($\_SESSION['ADMIN\_LOGIN']);

unset($\_SESSION['ADMIN\_ID']);

unset($\_SESSION['ADMIN\_NAME']);

header('location:home.php');

die();

?>

SEND\_MESSAGE.PHP

<?php

    require('connection\_inc.php');

    require('function\_inc.php');

    $name=get\_safe\_value($con,$\_POST['name']);

    $mobile=get\_safe\_value($con,$\_POST['mobile']);

    $email=get\_safe\_value($con,$\_POST['email']);

    $query=get\_safe\_value($con,$\_POST['query']);

    date\_default\_timezone\_set('Asia/Kolkata');

    $added=date('Y-m-d h:i:s');

    mysqli\_query($con,"insert into contact\_us(name,email,mobile,comment,added\_on) values('$name','$email','$mobile','$query','$added')");

    echo "send Your Request Succesfull";

?>

GET\_PROBLEM\_TITLE.PHP

<?php

        include('connection\_inc.php');

        include('function\_inc.php');

            $id=get\_safe\_value($con,$\_POST['id']);

            $sql="select \* from problem\_category where pro\_title\_id='$id'";

            $res=mysqli\_query($con,$sql);

            $html='';

            while($list=mysqli\_fetch\_assoc($res)){

                $html.='<option value='.$list['id'].'>'.$list['cat\_name'].'</option>';

            }

            echo $html;

?>

STYLE.CSS

\*{

    margin: 0;

    padding: 0;

    box-sizing: border-box;

}

body{

    font-family: 'Poppins', sans-serif;

}

/\* ---------------------header------------------------ \*/

.head{

    display: flex;

    /\* align-items:; \*/

    justify-content: space-between;

    align-items: center;

    padding: 5px 20px;

    text-transform: uppercase;

    background-color: rgb(133, 221, 224);

}

.head img{

    width: 100px;

    height: 100px;

}

nav{

    background-color: paleturquoise;

    margin-bottom: -15px;

}

nav ul{

    display: flex;

    list-style-type: none;

    padding: 20px 20px;

}

nav ul li a{

    padding: 20px 20px;

    /\* border:2px solid black \*/

    z-index: 1;

}

nav ul li a:hover{

    text-decoration: none;

    background-color: rgb(106, 201, 238);

    color: white;

}

.carousel-item img{

    height: 500px;

}

/\* ------------------------------about seciton--------------- \*/

.title{

    text-align: center;

    margin: 0 auto 30px;

    line-height: 60px;

    color: rgb(130, 214, 214);

    position: relative;

}

.title::after{

    content: '';

    background:rgb(146, 206, 201);

    width: 150px;

    height: 5px;

    border-radius: 5px;

    position: absolute;

    bottom: 0;

    left: 50%;

    transform: translateX(-50%);

}

.title::before{

    content: '';

    background:rgb(153, 202, 198);

    top: 47px;

    width: 20px;

    height: 20px;

    position: absolute;

    bottom: 0;

    left: 50%;

    transform: translateX(-50%) rotate(50deg);

}

.about{

    padding: 30px 20px ;

    height: 90vh;

    background-color: #9acfcd7c;

}

.about p{

    margin: 10px 30px;

    font-size: 20px;

    line-height: 40px;

    text-align: justify;

}

/\* ------------------------contact Us------------------- \*/

.contact{

    padding: 10px 30px ;

    width: 100%;

}

.row{

    display: flex;

    justify-content: space-around;

    align-items: center;

    flex-wrap: wrap;

}

.row .colu-2{

    flex-basis: 50%;

    min-width: 300px;

    padding: 0px 20px;

    /\* border: 2px solid black; \*/

}

.colu-2 img{

    width: 100%;

    height: 500px;

}

.colu-2 input , textarea{

    width: 100%;

    padding: 10px;

    border-radius: 5px;

    outline: none;

    border: 2px solid  rgb(201, 246, 243);

}

.colu-2 input:focus , textarea:focus{

    border: 3px solid  rgb(201, 246, 243);

}

.colu-2 input[type=submit],.colu-2 button{

    margin-top: 10px;

    background: rgb(201, 246, 243);

    padding: 15px;

    outline: none;

    border: 2px solid rgb(201, 246, 243);

    font-size: 20px;

    width: 100%;

    margin-top: 10px;

}

.colu-2 input[type=submit]:hover, button:hover{

    background: transparent;

    border: 2px solid rgb(201, 246, 243);

    outline: none;

}

.colu-2 label{

    margin-top: 0;

    margin-left: 0px;

}

.error\_msg{

    color: red;

}

/\* -----------------------------footer seciton-------------------------------- \*/

footer{

    background-color:rgb(173, 238, 235);

    padding: 10px 20px;

    width: 100%;

}

.footer-col-1{

    flex-basis: 30%;

}

.footer-col-2{

    flex-basis: 28%;

}

.footer-col-3{

    flex-basis: 28%;

}

.footer-col-1, .footer-col-2, .footer-col-3{

    margin: 20px 0px;

    text-align: center;

}

.footer-col-1 h2{

    line-height: 50px;

}

.footer-col-1 p{

    line-height: 30px;

}

footer p.copyright{

    text-align: center;

    padding: 10px;

}

footer ul{

    list-style-type: none;

}

footer ul li{

    margin-bottom: 10px;

}

footer hr{

    border: none;

    background-color: #5eeee7;

    height: 5px;

    margin: 10px 30px;

}

html {

    scroll-behavior: smooth;

}

/\* -------------------------path------------------------ \*/

.path{

    width: 100%;

    height: 40px;

    background-color: rgba(173, 241, 241, 0.787);

}

.path span{

    text-align: center;

    display: block;

    line-height: 40px;

}

.path span a{

    text-decoration: none;

    font-size: 20px;

    font-weight: 900px;

    text-align: center;

    color: rgb(48, 47, 47);

}

.path span a:hover{

    color: rgb(115, 194, 194);

}

/\* ---------------status ----------------- \*/

.login {

    margin: 30px;

}

/\* ---------------status ----------------- \*/

/\* .login input, label{

    margin: 30px;

} \*/

/\* --------------------------Mobile Product--------------------------- \*/

.search form{

    position: absolute;

    right: 70px;

}

.search form input[type="submit"]{

    background-color: skyblue;

}

.search form input {

    border: 2px solid skyblue;

    padding: 10px;

    border-radius: 5px;

}

.mobile\_product table,tr,th,td{

    text-align: center;

    border: 2px solid skyblue;

}

.mobile\_product td{

    text-align: center;

    border: 2px solid skyblue;

}

.mobile\_product table{

    margin: 80px auto 0px auto;

    width: 95%;

}

.mobile\_product table tr th{

    padding: 10px;

}

.mobile\_product table tr td{

    padding: 8px;

}

  .mobile\_product table tr td a{

    padding:5px;

    border-radius: 5px;

    text-decoration: none;

}

.mobile\_product table tr td a.view{

    background-color: rgb(201, 246, 243);

}

.mobile\_product table tr td a.add\_problem{

    background-color: rgb(201, 246, 243);

}

/\* --------------------add problem------------------------ \*/

.problem\_form{

    width: 95%;

    margin: auto;

    background-color: rgb(201, 246, 243);

    padding: 20px;

}

.problem\_form label{

    font-weight: 700;

    padding: 20px 0;

    width: 200px;

    /\* border: 2px solid black; \*/

}

.problem\_form span{

    display: inline-block;

    padding: 20px 10px 0;

    width: 200px;

    /\* border: 2px solid black; \*/

}

.problem\_form span select ,textarea{

    width: 100%;

}

.problem\_form input[type="submit"],input[type="reset"]{

    margin-top: 20px;

    background: rgb(159, 226, 226);

    padding: 15px;

    outline: none;

    border: 2px solid rgb(123, 175, 172);

    font-size: 20px;

    width: 100%;

}

.problem\_form input.product\_id{

    display: none;

}

.problem\_form input[type="submit"]:hover,input[type="reset"]:hover{

    background: transparent;

    border: 2px solid rgb(12, 12, 12);

    outline: none;

}

/\* -----------------------product detail----------------------- \*/

.product\_detail{

    padding: 20px;

    width: 95%;

    background-color: rgb(210, 230, 230) !important;

}

/\* ------------------added problems--------------------- \*/

.added\_problems span.btn{

    margin: 10px;

}

.added\_problems span.active{

    background-color: rgb(211, 193, 121);

}

.added\_problems span.deactive{

    background-color: skyblue;

}

MAIN.JS

// ----------------------for contact Us form--------------------------------

function send\_message() {

    jQuery('.error').html('');

    var name =jQuery("#name").val();

    var mobile =jQuery("#mobile").val();

    var email =jQuery("#email").val();

    var query =jQuery("#query").val();

    var is\_error='';

    if(name==''){

        jQuery('#error\_name').html('Please Enter Name');

        is\_error='yes';

    }else{

        jQuery('#error\_name').html('');

    }

    if(mobile==''){

        jQuery('#error\_mobile').html('Please Enter Mobile');

        is\_error='yes';

    }else{

        jQuery('#error\_mobile').html('');

    }

    if(email==''){

        jQuery("#error\_email").html('Please Enter email');

        is\_error='yes';

    }else{

        jQuery("#error\_email").html('');

    }

    if(query==''){

        jQuery("#error\_query").html('Please Enter query');

        is\_error='yes';

    }else{

        jQuery("#error\_query").html('');

    }

    if(is\_error==''){

        jQuery.ajax({

            url:'send\_message.php',

            type:'post',

            data:'name='+name+'&mobile='+mobile+'&email='+email+'&query='+query,

            success:function(result) {

                    jQuery('#error\_button').html(result);

                    // alert(result);

                    jQuery('#contact')[0].reset();

            }

        })

    }

}

// ------------------------for Login--------------------------

function sign\_in() {

    // jQuery('.error').html('');

    var email =jQuery("#username").val();

    var password =jQuery("#password").val();

    var is\_error='';

    if(email==''){

        jQuery('#error\_username').html('Please Enter Name');

        is\_error='yes';

    }else{

        jQuery('#error\_username').html('');

    }

    if(password==''){

        jQuery("#error\_password").html('Please Enter Password');

        is\_error='yes';

    }else{

        jQuery('#error\_password').html('');

    }

    if(is\_error==''){

        jQuery.ajax({

            url:'check\_signin.php',

            type:'post',

            data:'&email='+email+'&password='+password,

            success:function (result){

                jQuery('#error\_button').html(result);

               alert(result);

                   window.location.href='home.php';

            }

        })

    }

}

// ----------------------------------select problem Title & problem category--------------------------

$(document).ready(function(){

    jQuery('#problem\_title').change(function(){

      var id=jQuery('#problem\_title').val();

      if(id==''){

        jQuery('#problem\_category').html('<option value="">Select Problem Category....</option>');

      }else{

        jQuery('#problem\_category').html('<option value="">Select Problem Category....</option>');

              jQuery.ajax({

              type:'post',

              url:'get\_problem\_title.php',

              data:'id='+id,

              success:function(result){

                // alert(result);

                  jQuery('#problem\_category').append(result);

              }

        });

      }

    });

  });

**TESTING**

Testing is important phase in a Software Development Life Cycle.

Testing should be planned and conducted systematically.

**Generic aspects of a test strategy:-**

* Testing begins at the module level and works outward.
* Different testing techniques are used at different points of time.
* An independent test group does by developers and mainly for large projects, testing.
* Testing and debugging are two different activities, but debugging should be incorporated into any testing strategy.

**Different testing strategies use for system:-**

* Unit testing
* Integrated testing
* System testing.

**Unit Testing:**

It involves testing the minimum functionality of the smallest unit of software design modules. Using the design specification document as a guides, important control path are tested to uncover errors within the boundary of the module. This module level testing involves heavy use of white-box testing techniques. This testing can be conducted in parallel for multiple modules

**Integration Testing:**

Integration testing is a systematic technique for constructing the programmed structure while conducting tests to uncover errors associated with interfacing. It involves testing functionality of the modules with respective the other modules, with which the modules is supposed to interact. There is heavy use of black-box testing techniques and some use of white-box testing techniques to ensure coverage of major control paths.

**System Testing (Performance Testing):**

For real-time and embedded systems, software that provides required function but does not conform to performance requirement is unacceptable. Performance testing is designed to test run-time performance of the software.

**IMPLEMENTATION**

**Project planning and scheduling**

It is a measure of the practically of a specific technical solution and the availability of technical resources and expertise

* The proposed system uses Microsoft Visual Studio.NET as front-end and Sql Server 2000 as back-end tool.
* Sql Server 2000 is a popular tool used to design and develop database objects such as table views.
* The above tools are readily available, easy to work with and widely used for developing commercial application.

Hardware used in this project are- Pentium-4 class processor, 800 (MHz), 256 MB of RAM, 40 Gigabytes (GB) on installation drive, floppy drive.

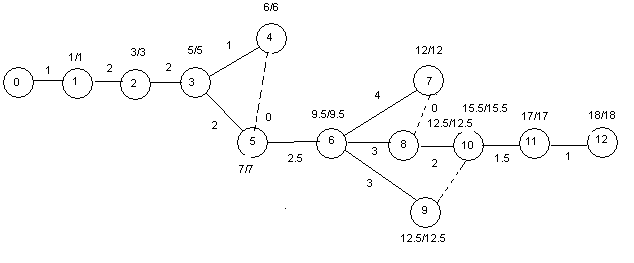
These hardware were already available on the existing computer system. The software like Microsoft Visual Studio.NET, operating system WINDOWS-XP/2000’ used were already installed on the existing computer system. So no additional hardware and software were required to purchase and it is technically feasible. The technical feasibility is in employing computers to the organization. The organization is equipped with enough computers so that it is easier for updating. Hence the organization has not technical difficulty in adding this system.

**PERT CHART**

Program Evaluation and Review Technique (PERT) and Critical Path Method (CPM) are the project scheduling techniques that can be applied to software development. Both technique are driven by information already gathered in earlier project planning activities:

* Estimation of effort
* A decomposition of the product function
* The solution of the appropriate model and task set
* Decomposition of tasks

Both PERT and CPM provide quantitative tools that allow the software planning to determine critical path – the claim of task that determined the duration of the project establish “most likely” times estimates for individual tasks by applying statically models: and Calculation “boundary times” that define a time “window” for a particular task Both PERT and CPM have been implemented in a wide verity of automated tools that are available for the personal computer. Such tools are easy to use and make the scheduling methods described previously available to every software project manager.



**Total Duration 18 Weeks**

**GANNT CHART**

When creating a software project schedule, the planner begins with a set of tasks (the work break down structure). If automated tools are used, the work break down is input as a task network or task outline. Efforts, duration, and start date are then input for each task. In addition, task may be assigned to specific individuals .As a sequence of this input, a timeline chart, also called a Gantt Chart, is generated.

A Gantt Chart can be developed for the entire project. All project task (for concept scooping) are listed in the left hand column. The horizontal bars occur at the same

|  |  |  |  |
| --- | --- | --- | --- |
| R/No. | Activity | Type of Activity | Duration in week |
| 1 | 0-1 | Study existing system | 1 |
| 2 | 1-2 | Study proposed system | 2 |
| 3 | 2-3 | Analysis | 2 |
| 4 | 3-4 | Feasibility study | 1 |
| 5 | 3-5 | Software Requirement & Analysis | 2 |
| 6 | 4-5 | Dummy | 0 |
| 7 | 5-6 | Design | 2.5 |
| 8 | 6-7 | Testing | 4 |
| 9 | 6-8 | Coding | 3 |
| 10 | 6-9 | Training | 3 |
| 11 | 7-8 | Dummy | 0 |
| 12 | 9-10 | Dummy | 0 |
| 13 | 8-10 | Implementation | 2 |
| 14 | 10-11 | Optimization | 1.5 |
| 15 | 11-12 | Review | 1 |

time on the calendar, task concurrency is implied. The diamonds indicate milestones. Once the information necessary for the generation of the Gantt Chart has been input, the major of software project scheduling tools produce project tables a tabular listing of all project tasks, their planned and actual start and end dates, and a verity of related information.

6

1

10

14

16

18

8

4

0

2

3

5

7

9

11

12

13

15

17

***Total Duration 18 Weeks***

**SYSTEM SECURITY**

Good software must have good security system. Any unauthorized access must not be permitted. My Project has double security system. First security for the administrator and second security for the lower level user. A administrator can give the authorization to accessing the system to the any user. A user who only authorization to accessing the system to the any user. A administrator can give the authorization to accessing the system to the any user. A user who only authorized by the administrator is only has an access right for this software. Only after the verification of master user login any one can change the internal record of the internal system. An entry is recorded every time user login into the system and at the time of logout

**FUTURE SCOPE OF THE PROJECT**

For future use, there are so many provisions specified with the help of

Which the system can also survive in the future with its excellent

capacity and robustness. With the help of these provisions, the system

may also be used for different helpdesk like power house ,reservation etc.

**LIMITATIONS**

Every Software has its benefits and limitation, so my software has also some limitation. This project is intended to be used “Railway Reservation System”. The sharing of information is possible only if we used same software. Presently this software is designed to work with Windows XP and SQL-Server. The other software platform may not be support this software. It is a single user not a multiuser.

**BIBLIOGRAPHY**

**BOOKS**

* “Beginning PHP and MySQL From Novice to Professional” by W Jason Gilmore
* “PHP: A Beginner’s Guide” by Vikram Vaswani
* “Learning PHP 5” by David Sklar
* “PHP Object – Oriented Solutions” by David Powers

**WEBSITES:**

* [www.google.com](http://www.google.com)
* [www.wikipedia.org](http://www.wikipedia.org)
* [www.w3schools.com](http://www.w3schools.com)
* [www.youtube.com](http://www.youtube.com)