**Declaration**

I hereby declare that the work being presented in this report entitled **“Store Locator”** is an authentic record of my own work carried out under the supervision of **Dr. Naveen Kr. Singh.**

The matter embodied in this report has not been submitted by me for the award of any other degree.

**Date : Signature of Student**

Himanshu Rajput

Abhinav Sharma

Deeksha Dwivedi

Akanksha

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

**Signature of HOD**  **Signature of Faculty**

**( Prof. Yogesh Kumar Mittal)** ( **Dr. Naveen Kr. Singh.)**

**ACKNOWLEDGEMENT**

A Project is a golden opportunity for learning and self development . We consider ourselves very lucky and honored to have so many wonderful people who lead us through the path of completion of this project.

This is a humble effort to express our sincere thanks to gratitude towards those who have guided and helped us to complete this project, which is a culmination of the degree of **M.C.A** to be awarded by **ABES ENGINEERING COLLEGE (GHAZIABAD).**

We are very much thankful to **Dr. Naveen Kumar Singh** and **Dr.Arjun Kumar Singh**for providing us, with all the necessary facilities and inculcating within us the positive spirit and attitude for the development of the project..

We feel honoured to express our humble gratitude to **our mentors** under whom we executed this project. His constant guidance and willingness to share his vast knowledge made us understand this project and its manifestations in great depths and helped us to complete the assigned tasks.

Lastly, we thank almighty, our family and friends for their constant encouragement without which this assignment would not be possible.

**HIMANSHU RAJPUT**

**ABHINAV SHARMA**

**AKANKSHA**

**DEEKSHA DWIVEDI**

**ABSTRACT**

“**STORE LOCATOR**” conveys the idea of providing services.

The website enables to provide complete information about the famous stores of cities so that user can have information about the stores and its location.

Finding Stores for various purposes in various cities is difficult therefore this system is in ongoing demand .Store Locator System deals with the maintenance of records of details of each store and its location.

We observed the working of the Store Locator system and after going through it, we get to know that there are many operations, which they have to do manually. It takes a lot of time and causes many errors. Due to this, sometimes a lot of problems occur and they were facing many disputes with customers. To solve the above problem, and further maintaining records of items, stores availability for customer , we are offering this proposal of Store Locator.

**CONTENT**

|  |  |
| --- | --- |
| **TOPIC** | **PAGE NO.** |
| 1.INTRODUCTION |  |
| 2.PURPOSE |  |
| 3.OBJECTIVE |  |
| 4.SYSTEM SPECIFICATION |  |
| 5.FEASIBILITY STUDY |  |
| 6.FEATURES |  |
| 7.TECHNOLOGY USED |  |
| 8.SDLC |  |
| 9.ADVANTAGE |  |
| 10.DISADVANTAGE |  |
| 11.DATA FLOW DIAGRAM |  |
| 12.E-R DIAGRAM |  |
| 13.DATABASE DESIGN |  |
| 14.TESTING |  |
| 15.IMPLEMENTATION |  |
| 16.LIMITATION |  |
| 17.FUTURE ENHANCEMENT |  |
| 18.CONCLUSION |  |
| 19.BIBLIOGRAPHY |  |

**INTRODUCTION**

The Internet has become the largest , most powerful medium for communication in the history of mankind. It has also become the single largest marketplace of history.

The website enables to provide complete information about the famous stores of cities so that user can have information about the stores and its location.

With the help of this we can maintaining records of items, stores availability for customer , we are offering this proposal of Store Locator.

In this age and time ,this website fundamentally aims to provide a living space on the net where people can interact as in a simple room at home.

The website hopes to position itself as a market leader in this rapidly growing segment.

**PURPOSE**

The purpose of this web application is to provide a platform where people can take knowledge of different places and stores using the internet as a medium.

The site will endeavor to provide a fast, reliable and intuitive interface for promoting these activities.

Finding Stores for various purposes in various cities is difficult therefore this system is in ongoing demand .Store Locator System deals with the maintenance of records of details of each store and its location.

The singular purpose is to encourage people to live an enjoyable and authentic social life in spite of limitations imposes by modern civilization.

**OBJECTIVE**

* To provide an environment where people can get details of different stores.
* To provide facilities for online registration with proper validation of details .
* To provide authentication using username and password.
* To have attractive and secure login page to access.
* To provide a text based and image attached messaging service.
* Creating user profile having personal information.
* To provide an intuitive easy to use interface.
* Contact detail is used for sharing views and ideas.

**System Requirement**

**HARDWARE REQUIREMENT**

**RAM:**

* Minimum 512MB.

**PROCESSOR:**

* Pentium 4 or above.

**HARD DISK:**

* Minimum 2GB or above.

**OTHERS:**

* Monitor
* Keyboard
* Mouse
* Internet Connection
* Web Browser

**SOFTWARE REQUIREMENT**

**FRONT END:**

* HTML
* Java Script
* CSS
* PHP

**BACK END:**

* DATABASE: MySQL
* SERVER: XAMPP

**OPERATING SYSTEM:**

* Windows XP or Onward

**FEASIBILITY STUDY**

**WHAT ARE THE USER’S DEMONSTRANTABLE NEEDS?**

Users need to socialize with each other ,taking knowledge about different things and places and maintaining different levels of privacy.

**HOW CAN THE PROBLEM BE REDEFINED?**

We proposed our perception of the system , in accordance with the problems of existing system by making a full layout . We were further updating in the layout in the basis of redefined the problem . In feasibility study phase we had undergone through various steps, which are described as under: how feasible is the system proposed? This was analyzed by comparing the following factors:

* Cost
* Effort
* Time
* Labour

**COST**

The cost required in the proposed system is comparatively less to the existing system.

**EFFORT**

Compared to the existing system the proposed system will provide a better working environment in which there will be ease of work and effort required will be comparatively less than the existing system.

**TIME**

Also the time required generating a page and emotional sites will be comparatively very less than in the existing system.

**LABOUR**

Also the time required and doing any other work will comparatively very less.

Record updating will take less time.

**FEATURES**

The Web Application provides the following facilities:-

1.The software provides a web interface .

2.Secured User login.

3. The software provides facilities to find stores.

4.The software has a pleasant look.

5.The software is fast and reliable.

6.The modules available are

a) Registration

b) Login

c)Feedback

d) Details of store

**SDLC**

The system development life cycle (SDLC), also referred to as the application development life –cycle , is a term used in systems engineering, information system and software engineering to describe a process for planning , creating, testing, and developing an information system. The system development life-cycle concept applies to a range of hardware and software configuration , as a system can be composed of hardware only , software only ,or a combination of both.

A system development life-cycle is composed of a number of clearly defined and distinct work phases which are used by systems engineering and system developers to plan for , design, build, test and deliver information system . Like anything that is manufactured on an assembly line, an SDLC aims to produce high quality systems that meet or exceed customer expectations, based on customer requirements, by delivering systems which move through each clearly defined phase, within scheduled time –frames and cost estimates.

Computer system are complex and often (especially with the recent rise of service- oriented architecture) link multiple traditional system potentially supplied by different software vendors. To manage this level of complexity , a number of SDLC models or methodologies have been created, such as “waterfall” ; “spiral” ; “Agile software development “ ; “rapid prototyping” ; “incremental”; and “synchronize and stabile”.

ISO/IEC 12207 is an international standard for software life- cycle processes . It aims to be the standards that defines all the tasks required for developing and maintaining software.

**SOFTWARE DEVELOPMENTACTIVITIES**

* Planning
* Implementation , testing and documentation
* Deployment and maintenance

**PLANNING:**

Planning is an objective of each and every activity , where we want to discover things that belongs to the project. An important task in creating a software program is extracting the requirements or requirements analysis. Customer typically have an abstract idea of what they want as an end result , but do not know what software should do. Skilled and experienced software engineering recognize incomplete, ambiguous , or even contradictory requirements at this points. Frequently demonstrating live code may help reduce the risk that the requirements are incorrect.

Once the general requirements are gathered from the client , an analysis of the scope of the development should be determined and clearly stated. This is often called a scope document.

Certain functionality may be out of scope of the project as a function of cost or as a result of unclear requirements at the start of development. If the development is done externally , this document can be considered a legal document so that if there are ever disputes , any ambiguity of what was promised to the client can be clarified.

**IMPLEMENTATION, TESTING, DOCUMENTATION:**

Implementation is the part of the process where software engineers actually program the code for the project.

Software testing is an integral and important phase of the software development process. This part of the process ensures that defects are recognized as soon as possible.

Documenting the internal design of software for the purpose of future maintenance and enhancement is done throughout . This may also include the writing of an API , be it external or internal. The software engineering process choosen by the developing team will determine how much internal documentation (if any) is necessary. Plan –driven models(e.g., waterfall) generally produce more documentation than agile models.

**DEPLOYMENT AND MAINTAINANCE:**

Deployment starts directly after the code is appropriately tested, approved for release , and sold or otherwise distributed into a production environment. This may involve installation , customization(such as by setting parameters to the customer’s value), testing, and possibly an extended period of evaluation. Software training and support is important , as software is only effective if it is used correctly.

Maintaining and enhancing software to cope with newly discovered faults or requirements can take substantial time and effort, as missed requirements may force redesign of the software.

**ADVANTAGES**

* Easy application for user .
* Easy to Install and Maintain .
* Can run on any smartphone or android app.
* Easy to find various stores as per the location option provided.

**DISADVANTAGE**

* No Real time application.
* No Route direction guidelines.
* Limited stores available provided in database.

**DATA FLOW DIAGRAM**

As information moves through soft ware, it is modified by a series of transformations. A DFD is a graphical technique that depicts information flow and the transform that are applied at data move from input to output. The DFD is also known as the Data Flow Chart or Bubble Chart. The DFD may be used to represent a system or software at any level of abstraction. In fact, DFD’s may be partitioned into levels that represent increasing information flow and functional detail. A level 0 DFD, also known as a Fundamental System Model or a Context Diagram, represents the entire software element as a single bubble with input & output data indicated by incoming & outgoing arrows, respectively.

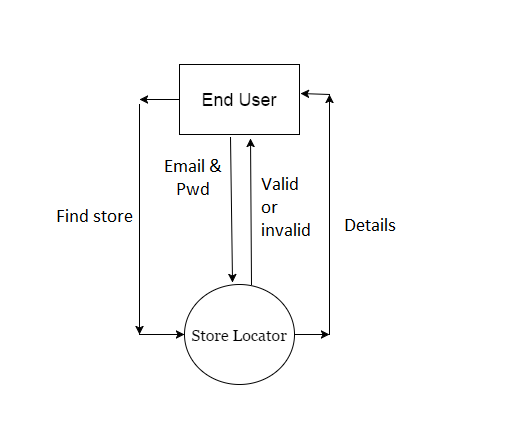
External Entity

A directed line represents the flow of data

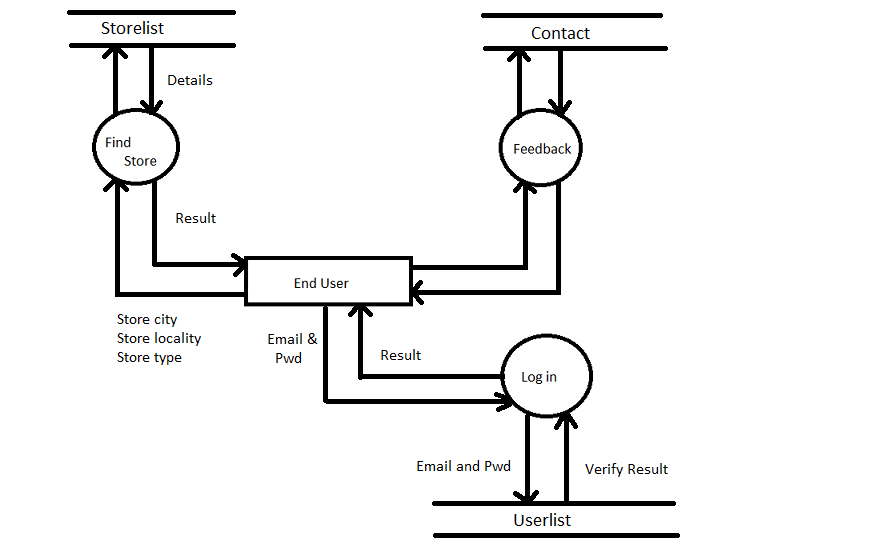
Process

An open-ended rectangle represents data storage.

**0 –LEVEL DFD**

****

**1-LEVEL DFD**



**E-R DIAGRAM**

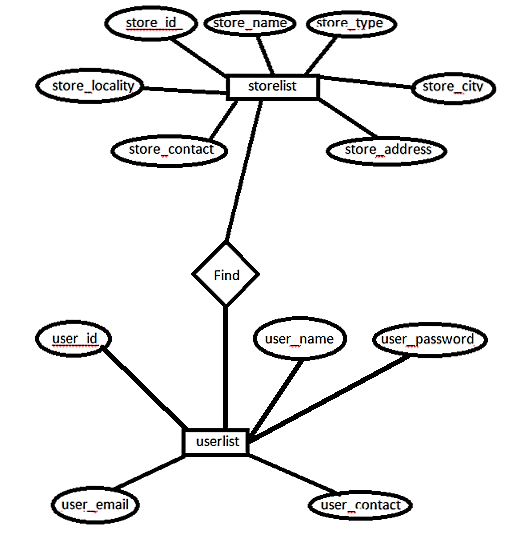
An **entity–relationship model** (**ER model**) describes inter-related things of interest in a specific domain of knowledge. An ER model is composed of entity types (which classify the things of interest) and specifies relationships that can exist between instances of those entity types.

In Software engineeringan ER model is commonly formed to represent things that a business needs to remember in order to perform business processes. Consequently, the ER model becomes an abstract data modelthat defines a data or information structure that can be implemented in a data base, typically a relational database.

Entity–relationship modeling was developed for database design by Peter Chenand published in a 1976 paper. In a relational database a relationship between entities is implemented by storing the primary key of one entity as a pointer or "foreign key" in the table of another entity

There is a tradition for ER/data models to be built at two or three levels of abstraction. Note that the conceptual-logical-physical hierarchy below is used in other kinds of specification, and is different from the three schema approach to software engineering.

**E-R DIAGRAM**

****

**DATABASE DESIGN**

**Database design** is the process of producing a detailed data model of a database.This data model contains all the needed logical and physical design choices and physical storage parameters needed to generate a design in a data definition language which can then be used to create a database. A fully attributed data model contains detailed attributes for each entity.

The term database design can be used to describe many different parts of the design of an overall database system Principally, and most correctly, it can be thought of as the logical design of the base data structures used to store the data. In the relational model these are the tables and views In an object database the entities and relationships map directly to object classes and named relationships. However, the term database design could also be used to apply to the overall process of designing, not just the base data structures, but also the forms and queries used as part of the overall database application within the database management system(DBMS).

The process of doing database design generally consists of a number of steps which will be carried out by the database designer. Usually, the designer must:

* Determine the data to be stored in the database.
* Determine the relationships between the different data elements.
* Superimpose a logical structure upon the data on the basis of these relationships.

Within the relational model the final step above can generally be broken down into two further steps, that of determining the grouping of information within the system, generally determining what are the basic objects about which information is being stored, and then determining the relationships between these groups of information, or objects. This step is not necessary with an object data+base.

**THE DESIGN PROCESS:**

1. **Determine the purpose of the database**

This helps prepare for the remaining steps.

1. **Find and organize the information required**

Gather all of the types of information to record in the database, such as product name and order number.

1. **Divide the information into tables**

Divide information items into major entities or subjects, such as Products or Orders. Each subject then becomes a table.

1. **Turn information items into columns**

Decide what information needs to be stored in each table. Each item becomes a field, and is displayed as a column in the table.

1. **Specify primary keys**

Choose each table’s primary key. The primary key is a column, or a set of columns, that is used to uniquely identify each row. An example might be Product ID or Order ID.

1. **Set up the table relationships**

Look at each table and decide how the data in one table is related to the data in other tables. Add fields to tables or create new tables to clarify the relationships, as necessary.

1. **Refine the design**

Analyze the design for errors. Create tables and add a few records of sample data. Check if results come from the tables as expected. Make adjustments to the design, as needed.

1. **Apply the normalized rule**

Apply the data normalization rules to see if tables are structured correctly. Make adjustments to the tables.

**TESTING**

**SOFTWARE TESTING**

**Software testing** is an investigation conducted to provide stakeholders with information about the quality of the product or service under test.Software testing can also provide an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation. Test techniques include the process of executing a program or application with the intent of finding software bugs (error other defects).

Software testing involves the execution of a software component or system component to evaluate one or more properties of interest. In general, these properties indicate the extent to which the component or system under test:

* meets the requirements that guided its design and development,
* responds correctly to all kinds of inputs,
* performs its functions within an acceptable time,
* is sufficiently usable,
* can be installed and run in its intended environment and
* achieves the general result its stakeholders desire.

Software testing can be conducted as soon as executable software (even if partially complete) exists. The overall approach to software development often determines when and how testing is conducted. For example, in a phased process, most testing occurs after system requirements have been defined and then implemented in testable programs. In contrast, under an Agile approach requirements, programming, and testing are often done concurrently.

**White-box Testing:**

**White-box testing** (also known as **clear box testing**, **glass box testing**, **transparent box testing** and **structural testing**, by seeing the source code) tests internal structures or workings of a program, as opposed to the functionality exposed to the end-user. In white-box testing an internal perspective of the system, as well as programming skills, are used to design test cases. The tester chooses inputs to exercise paths through the code and determine the appropriate outputs. This is analogous to testing nodes in a circuit, e.g. in- circuit(ICT).

While white-box testing can be applied at the unit, integrarion and system levels of the software testing process, it is usually done at the unit level. It can test paths within a unit, paths between units during integration, and between subsystems during a system–level test. Though this method of test design can uncover many errors or problems, it might not detect unimplemented parts of the specification or missing requirements.

Techniques used in white-box testing include:

* API testing (application programming interfaces)– testing of the application using public and private APIs.
* Code coverage-creating tests to satisfy some criteria of code coverage (e.g., the test designer can create tests to cause all statements in the program to be executed at least once).
* Fault injection methods-intentionally introducing faults to gauge the efficacy of testing strategies.
* Mutation testing methods.
* Static testing methods.

Code coverage tools can evaluate the completeness of a test suite that was created with any method, including black-box testing. This allows the software team to examine parts of a system that are rarely tested and ensures that the most important function point have been tested.Code coverage as a software metric can be reported as a percentage for:

* *Function coverage*, which reports on functions executed
* *Statement coverage*, which reports on the number of lines executed to complete the test
* *Decision coverage*, which reports on whether both the True and the False branch of a given test has been executed

100% statement coverage ensures that all code paths or branches (in terms of control flow) are executed at least once. This is helpful in ensuring correct functionality, but not sufficient since the same code may process different inputs correctly or incorrectly.

**BLACK-BOX TESTING:**

**Black-box testing** treats the software as a "black box", examining functionality without any knowledge of internal implementation, without seeing the source code. The testers are only aware of what the software is supposed to do, not how it does it.Black-box testing methods include: equivalence partitioning , boundary value analysis, all-pairs testing, state transition tables , decision table testing , fuzz testing, model-based testing, use case testing tables, exploratory testing and specification-based testing.

**Specification-based testing** aims to test the functionality of software according to the applicable requirements.This level of testing usually requires thorough test cases to be provided to the tester, who then can simply verify that for a given input, the output value (or behavior), either "is" or "is not" the same as the expected value specified in the test case. Test cases are built around specifications and requirements, i.e., what the application is supposed to do. It uses external descriptions of the software, including specifications, requirements, and designs to derive test cases. These tests can be functional or non- functional, though usually functional.

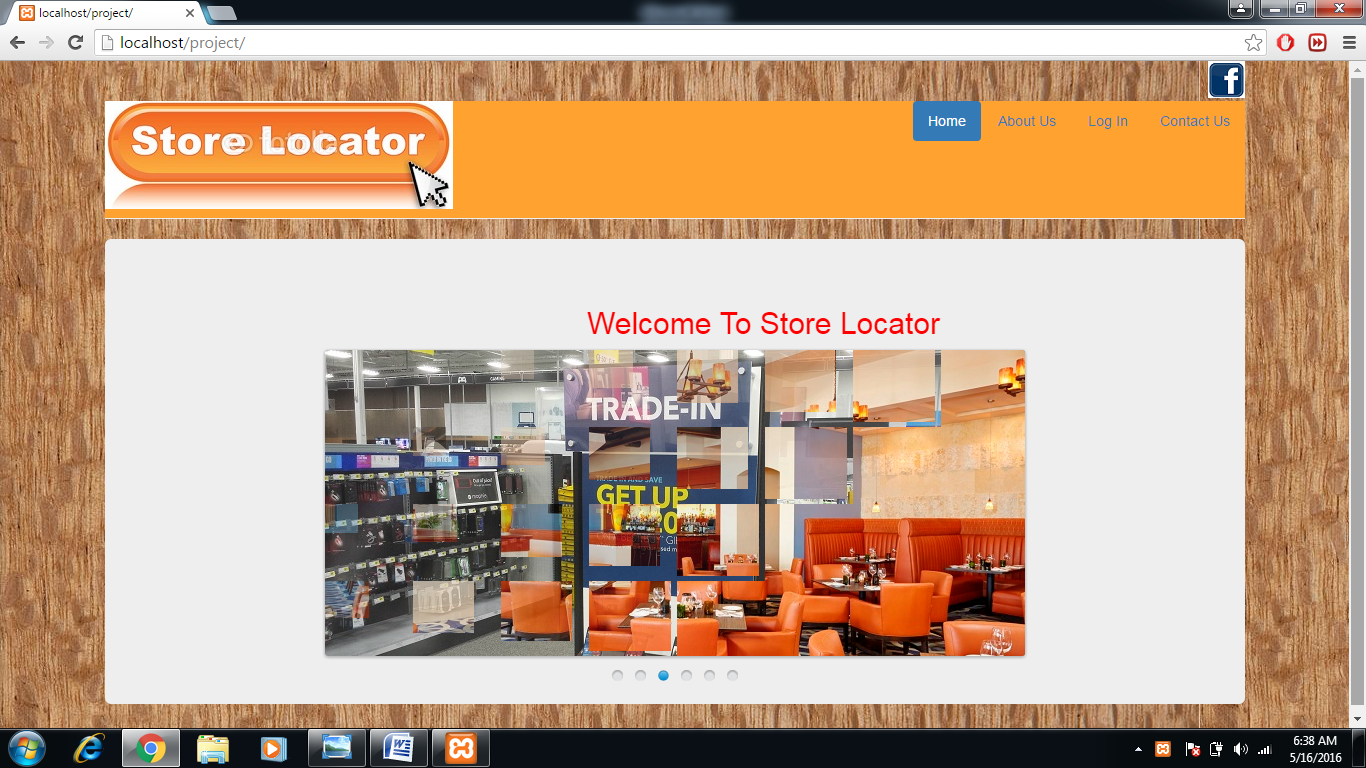
Specification-based testing may be necessary to assure correct functionality, but it is insufficient to guard against complex or high-risk situations.

One advantage of the black box technique is that no programming knowledge is required. Whatever biases the programmers may have had, the tester likely has a different set and may emphasize different areas of functionality. On the other hand, black-box testing has been said to be "like a walk in a dark labyrinth without a flashlight." Because they do not examine the source code, there are situations when a tester writes many test cases to check something that could have been tested by only one test case, or leaves some parts of the program untested.

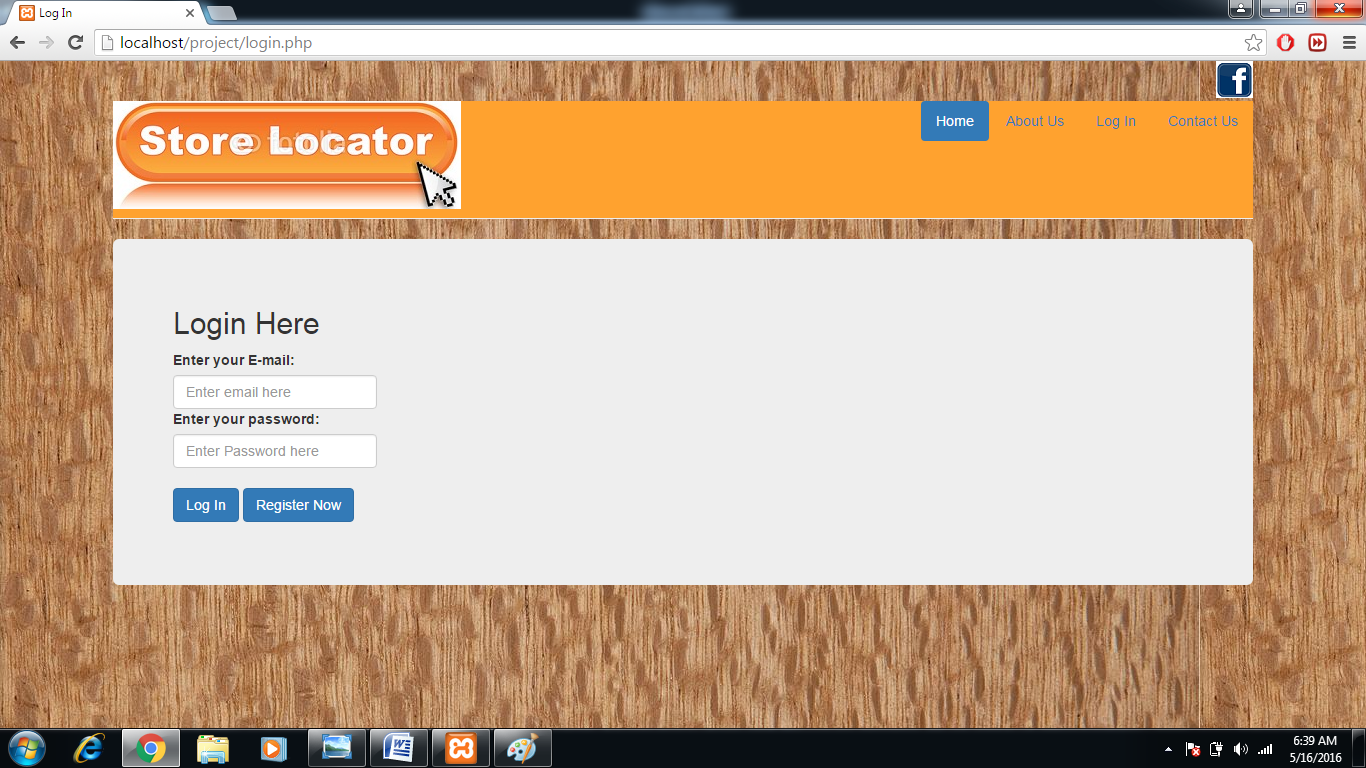
This method of test can be applied to all levels of software testing: unit, integration , system and acceptance.It typically comprises most if not all testing at higher levels, but can also dominate unit testing as well.

**IMPLEMENTATION**

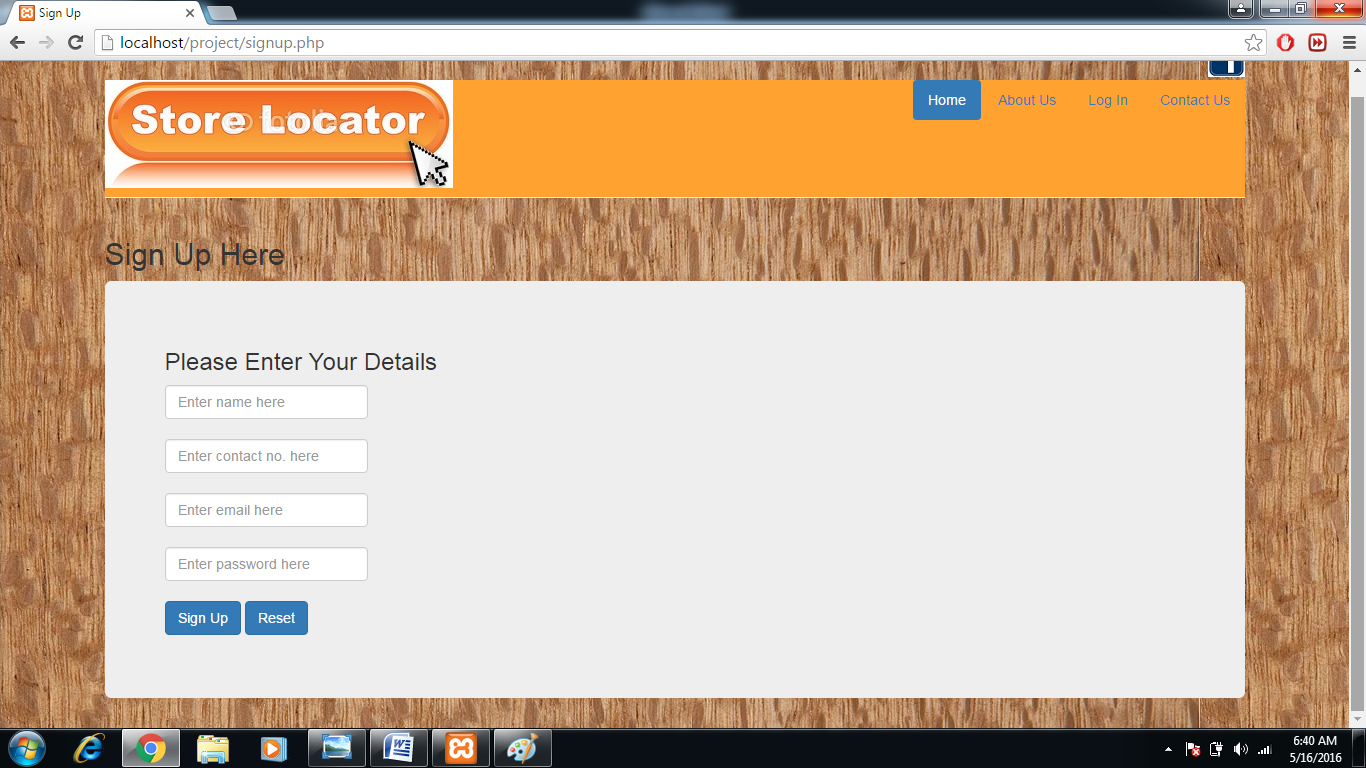
Index page

****

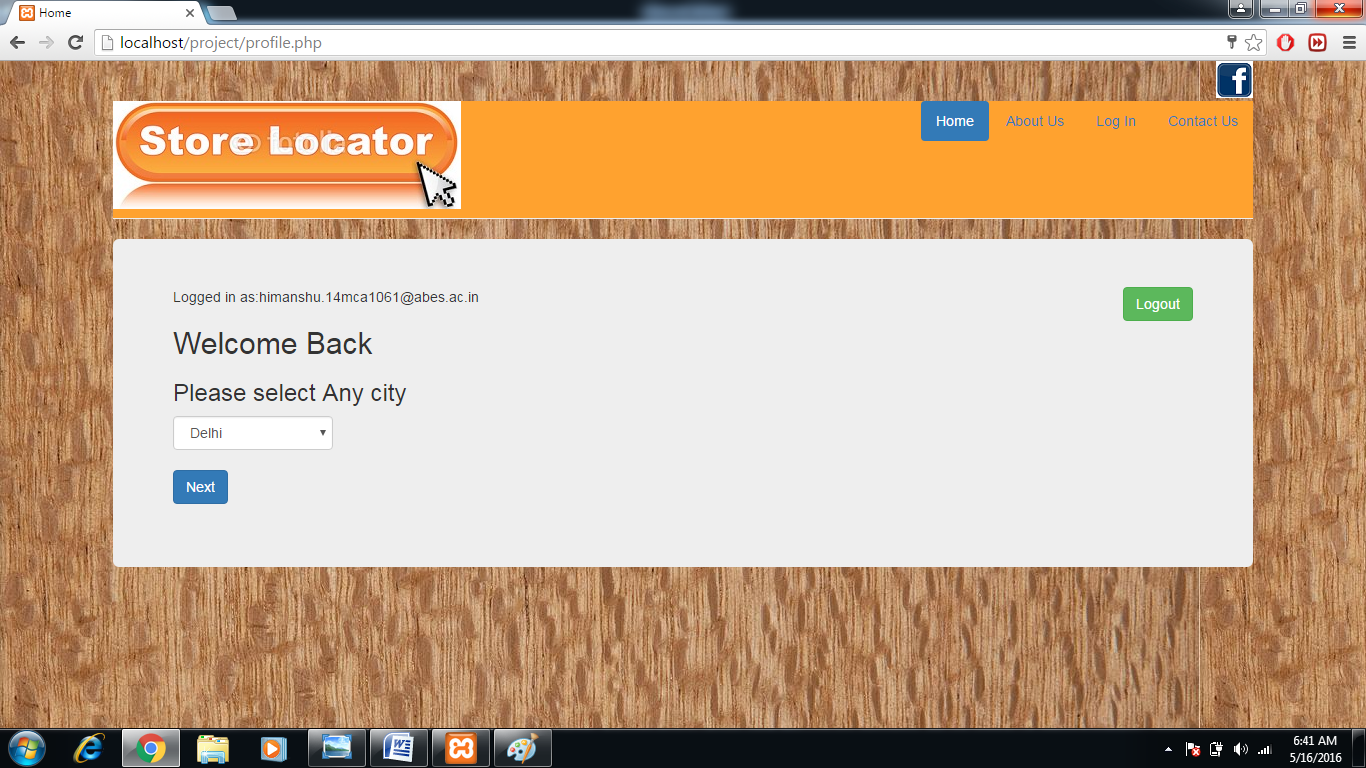
**Login page**



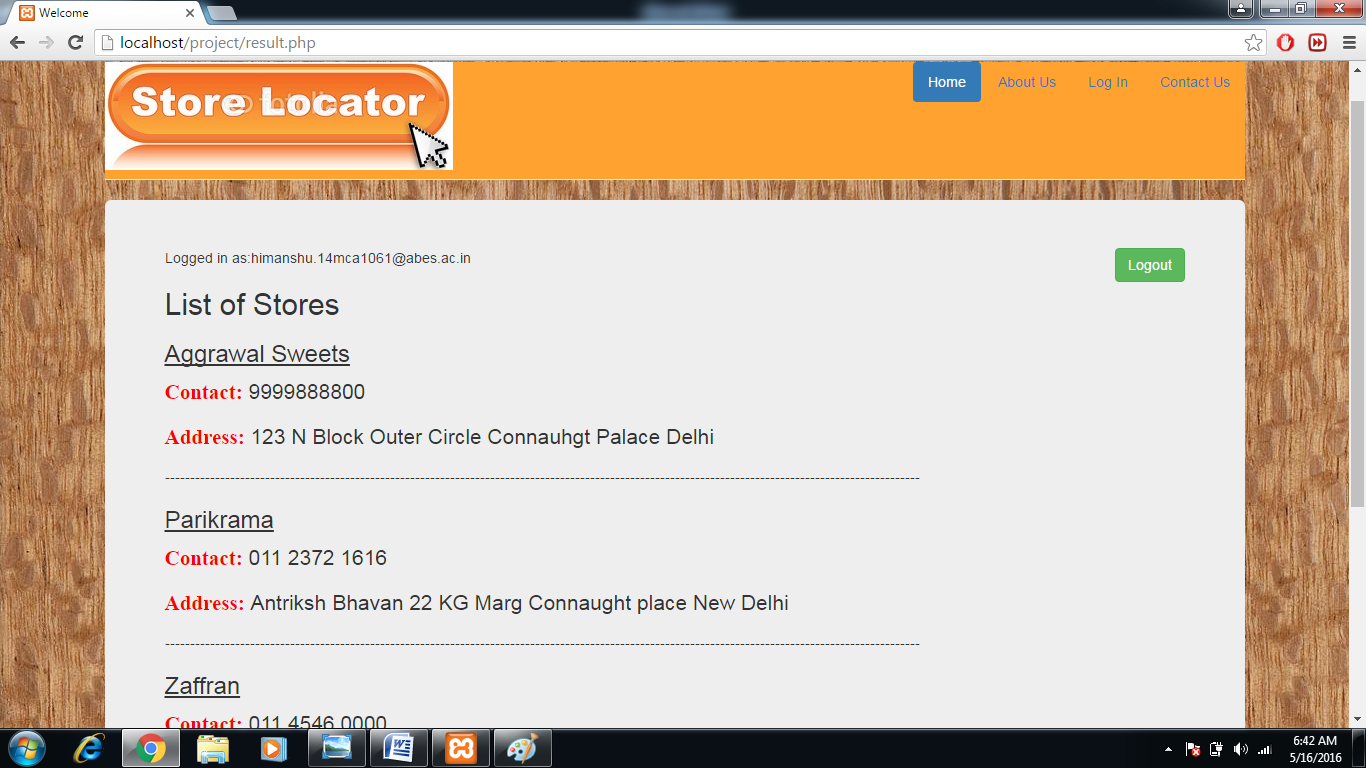
Register page

****

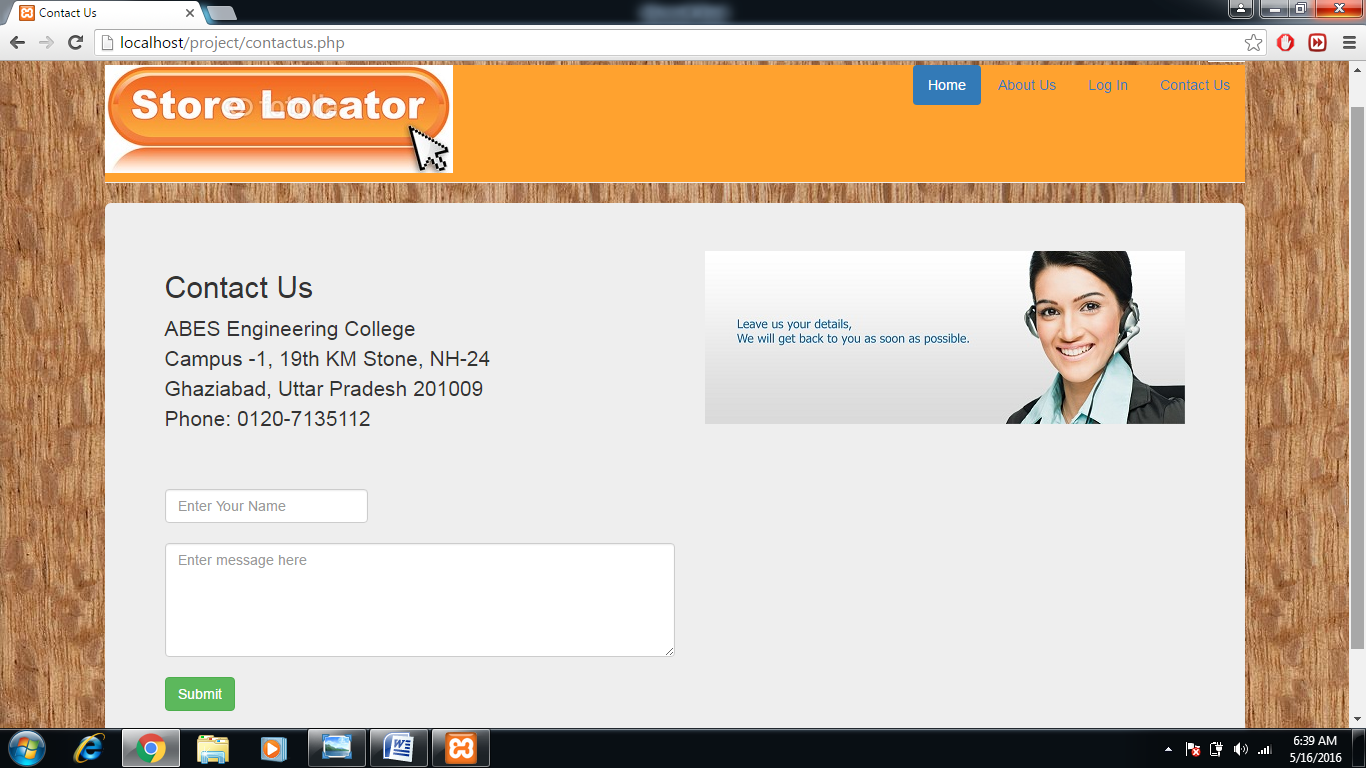
User Home page

****

Result page

****

Feedback page

****

**LIMITATION**

* The website is presently only available in English.
* The website is presently not available on the internet.
* No online chatting system.
* No IVRS system(Interactive Voice Response System).
* The website presently does not provide audio/video chatting facilities.
* Uploading, sharing and tagging of images and videos are not available.
* No advertisement panel.

**FUTURE ENHANCEMENT**

It is very difficult for some one to predict the negative areas in his own creation .How ever no project is perfect in the corporate world, so considering this view there may be some enhancement which can be incorporated in the project.

Currently, the system works for limited number of places to visit. In near future it will be extended for more places so that reliability can be improved.

**Conclusion**

The use of “Store Locator” is only to have a preplanned schedule in order to find a store at right time. So By using this project one can easily get the information of any store. Hence we can conclude that this version is better version over the older versions that were being used in maintaining the all detail and is totally based on latest information…

**Bibilography**

We referred some books which had provided us with much of guidance to develop the code for the website. There are a few good books and websites that we referred to develop Php, JavaScript and HTML codes.

Apart from codes these books and websites had provides us with few good techniques and methodology to develop the website.

Books References:-

|  |  |  |
| --- | --- | --- |
| **S no.** | **Book Title** | **Author** |
| 1. | Software Engineering | K.K Aggrawal |
| 2. | Database management  System | Henry F.korth |
| 3. | Head first php | Herbert Schildt |
| 4. | HTML complete Reference | Herbert Schildt |

URL REFERENCES

* <http://www.google.com>
* <http://www.tutorialpoint.com>
* <http://yatra.com>
* <http://makeMyTrip.com>
* <http://journey.com>