TRIBHUVAN UNIVERSITY
INSTITUTE OF ENGINEERING
PULCHOWK CAMPUS

A
FINAL PROJECT REPORT ON
"MywebAnalyzer"

Submitted to:
Department of Computer & Electronics Engineering.

Date:2nd March 2008

Submitted by:
nabin maharjan (27124)
Nabin247@gmail.com
omkar ranjit   (27125)
omkar.ranjit@gmail.com
surendra maharjan(27141)
suren_1s@hotmail.com

Acknowledgement

Nabin Maharjan(061/BCT/526)

Omkar Ranjit(061/BCT/527)

Surendra Maharjan(061/BCT/544)

Abstract

In the present age, web-site services are being popular for providing information and optimizing the business sectors. Web analytics is very useful tool to monitor the effectiveness of one's website(s). It collects useful data about the visitors, processes it and then provides the result in a statistical and graphical way so that the one can grasp all useful information and use it for meeting one's goals and objectives.

My Web Analyzer is a software that collects data from client-side. It is based on page tagging technology. It simply collects data from the browser, analyze it in the server side and display the information in statistical and graphical form about the web traffic, find out the most popular pages, new visitors, loyal visitors ,average time per visit and many other useful metrics. By obtaining the information from web analytics, one can optimize Web site effectiveness, increase website conversion and increase online sales and leads, reduce customer support cost and improve website usability.

The project mainly makes the use of java (JSP and Servlet) for application, javascript for tracking code, oracle 9i for database, APIs for visitor's location information and google maps and Cewolf tag libraries for charts.
Although there are number of webanalytic software packages, they are suitable for medium and large companies only due to their high cost. So, "My Web Analyzer" is more suitable for small organizations and individual site owners.

This report for MywebAnalyzer is designed and developed as a third year minor project report.

Table of contents:

1 Introduction

1.1 Background

The importance of Internet is ever growing. E-banking, E-entertainment, E-governance, one has just to name it, and one can not help but awe at the marvels of the Internet. So, it is very important to measure the performance of website using some web metrics which can help to take necessary actions for optimization of the websites.

Before 1998, most of the web-page owners used web-page counter that simply provides information about how many people had come to that page. As long as the counter didn't crash, or corrupt its storage, or overflow and starts its count from zero again, there would be a nifty little graphics of numbers that looked like roller skates. By the year 1998, website statistics packages or "stats" came into common use. Stat packages collect data but leave the owner to analyze the data. The difficulty and time-consuming procedure in stat packages led to the need of software that collects the data about the web-page visitors and process or analyze it and provide information in some appropriate format that can be easily grasped by web-page owner.

The available stat packages give data about the web-page visitors in terms of numbers and we, people have to analyze the data and then perform some improvements in the web-page based on the analysis in order to increase the number of visitors. Analysis of data is difficult and time consuming.

So, to avoid this problem in using stat packages, some of the analytics software were developed such as AWStats and Webalizer .These softwares collect data, analyze them and display information in some graphical and statistical forms. However, they use server log analysis method which reads the log files in which the web server records all its transactions. It is based on server-side data collection. Those website owners who do not have their own server can not benefit from these log-based analytics software. Besides, they are prone to faulty judgements when requests are made from dynamically IP assigned computers, proxies or browser cache.

So, web analytics software based on client side came upon scene which uses page tagging technology based on javascript to collect visitors' data. Data tags usually append the JavaScript code on website pages. When read by a visitor's browser, data-tags collect information related to the visitor's activity and transmit that data to data-collection servers, typically through an image request with the various parameters of interest appended to the query string.

2 Problem Statement

Web Analytics software can help a webmaster or website owner to know the performance of his website. It can help webmasters to measure the web traffic, find out the most popular pages, new visitors, loyal visitors ,average time per visit and many other useful metrics which can help them in optimizing their websites.   MywebAnalyzer can be used to meet this purpose. It is based on page-tagging technology and hence, best suited for those website owners who have their sites hosted on some other companies but want to analyze the performance of their websites.

3 Objectives

As mentioned in the problem statement,   MywebAnalyzer aims at fulfilling the needs of the website owners who do not have their own servers. The main objectives of it are summarized below.

- To collect the information of webpage visitors from client side.
- To analyze various visited web pages in terms of entry pages, exit pages, bounce rate etc.
- To analyze visitors in terms of new visitors, repeat visitors, location, system etc.
- To measure the web traffic.
- To process the collected data/information in server side and provide information in statistical and graphical form.
- To provide interface for website owners to view and analyze the performance of their websites.

4 Existing System

Google analytics is one such system that uses page tagging method to collect information from client side browser and process its server side. Any user who has got gmail account can use google analytics to monitor their site performance.

5 Proposed System

Server Side: Client Side:

Mywebanalyzer

Owner Side

Owner1   Owner2   OwnerN

Browser1   Browser2   ....   BrowserN

Login/Request

Reports

Database

Visitor Side

Browser1   Browser2   BrowserN

Website1   Website2   ....   WebsiteN
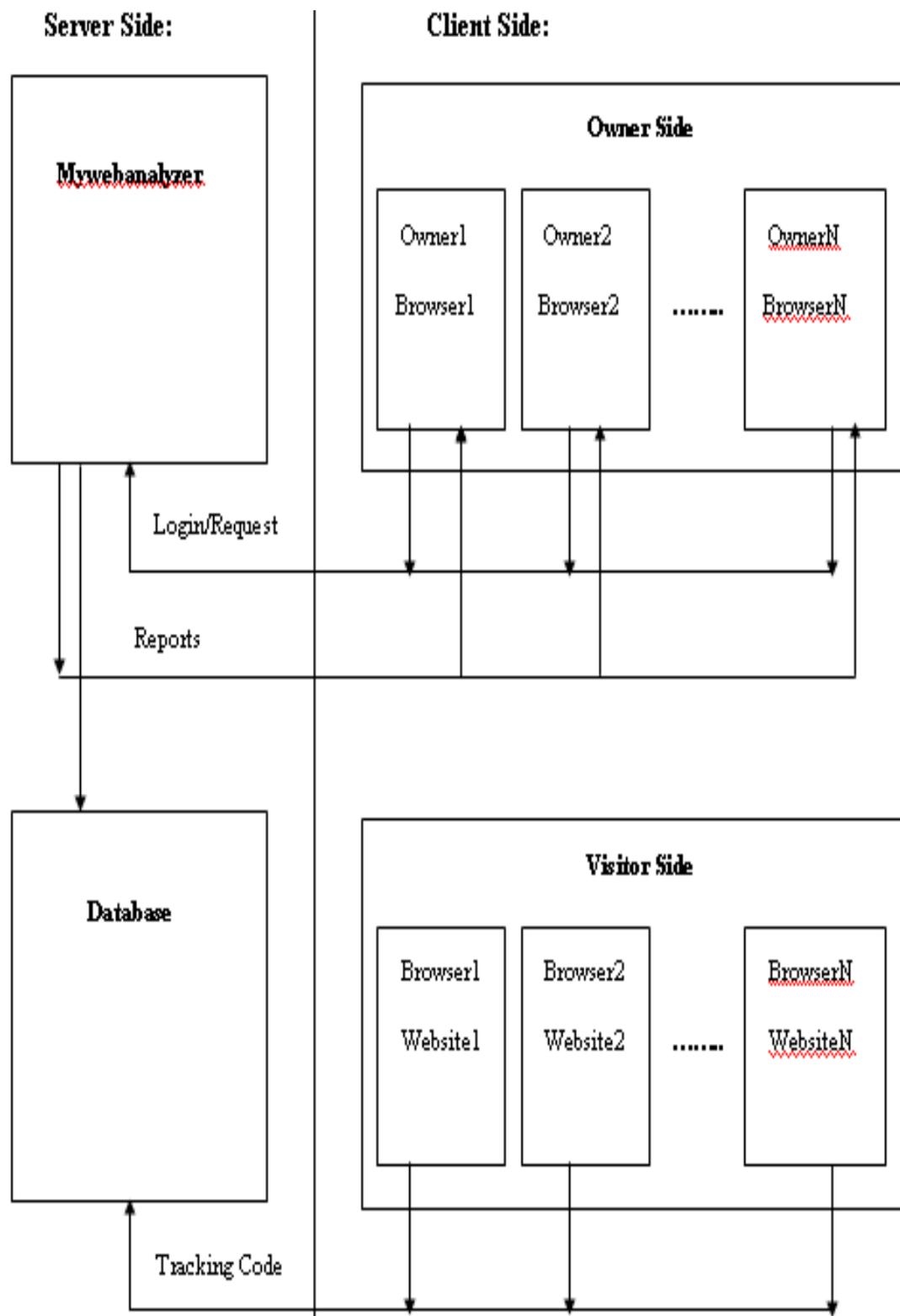
Tracking Code

Fig: Block diagram of MyWebAnalyzer

5.1 Introduction

MywebAnalyzer is a program that collects data from client-side. It is based on page tagging technology. It simply collects data from the browser, analyze it in the server side and display the information in statistical and graphical form as and when required.

Any user (webpage owner) who wants to use MywebAnalyzer has to register an account to track their website in My Web Analyzer. Then, a piece of tracking code is provided which the user has to put in the web pages of his site. Unique visitors and visits are managed in the tracking code using cookies. Now, every time a visitor visits the tracked site, the detailed information of the visit is sent to the database of MywebAnalyzer via tracking code. Session cookie is used to track a particular visit while multiple visits of a unique visitor are managed using persistent cookies. The user can sign in his account in his convenient time and view the performance report of his site for a particular reporting period.

5.2 Building Block Terms

Page: It simply refers the individual web pages in a website.

Page View: It is determined by an occurrence of the script being run in page tagging.

Visit/Session: A visit is an interaction, by an individual, with a website consisting of one or more page views. If an individual has not taken another action (typically additional page views) on the site within a specified time period usually 30 minute, the visit session will terminate.

Visitor/Unique Visitor: The uniquely identified client generating requests on the web server (log analysis) or viewing pages (page tagging) within a defined time period (i.e. day, week or month). A visitor can make multiple visits within a reporting period. A Unique Visitor counts once within the timescale.

New Visitor: A visitor that has not made any previous visit to the site.

Return Visitor: A unique visitor visiting a site during a reporting period who has also visited the site prior to the reporting period. Each individual is counted only once in the reporting period.

Repeat Visitor: A unique visitor with activity consisting of two or more visits to a site during a reporting period. It is possible for a visitor to be counted as both New and Repeat or both Return and Repeat during a reporting period. It is not possible for a visitor be counted as both New and Return in the same reporting period.
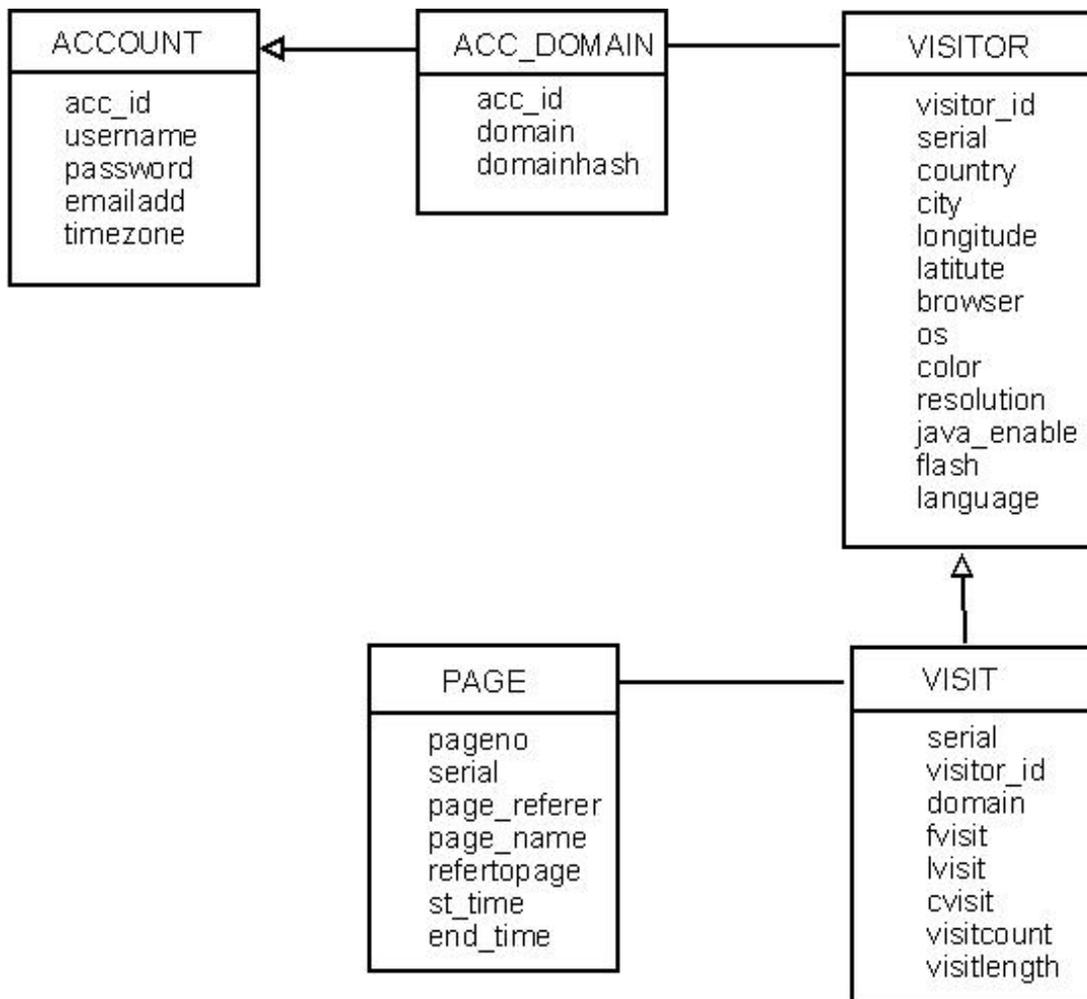
5.3 Database Schema:

Fig: Database Schema of MywebAnalyzer

6   Application Development

Java (JSP and servlets) for application

JavaScript for tracking code

Oracle 9i for database

Maxmind java api and geolitecity.dat database

Google Ajax Api for google maps

Cewolf_taglib library for chart presentation

6.1 Java for application

JavaServer Pages (JSP) and servlets are important parts of Sun's J2EE (Java 2 Platform, Enterprise Edition) architecture.Java Server Pages (JSP) is a specification that describes what a servlet-based content creation system should do. One of its main purposes is to facilitate the creation of dynamic Web pages. Servlet containers interpret these files and create servlets (also know as workhorse servlets) to process HTTP requests and produce responses. Servlets are server plug-ins that extend the capabilities of your Web server. They provide a straight forward deployment mechanism for your applications. Servlets are deployed inside servlet containers, which are plug-ins to your Web server. MywebAnalyzer is a web-based application. So, JSPs and Servlets have been used to develop the application.

6.2 JavaScript for tracking code

MywebAnalyzer is based on page tagging which is done on client side. Hence, client side JavaScript, developed by NetScape, has been used as the tracking code. In JavaScript, variables are loosely typed and code is dynamically interpreted.

6.3 Oracle 9i and Oracle JDBC connector for database

Oracle 9i provides everything needed to develop, deploy and manage Internet applications. It provides robust reliable, available, secure architecture which is scalable

from departments to e-business sites. Hence, Oracle 9i is used as database to store the tracking information of the websites supplied by the tracking code.

6.4 Maxmind java api and its geolitecity.dat database

Maxmind offers many geoip based solutions. We used its freely distributed Maxmind java api and its free geolitecity.dat database to map ip addresses to corresponding countries and cities.

6.5 Google Ajax Api for Google Maps

Currently, Google provides common Ajax Api loader to load its three free apis,namely,Google maps,Google Ajax Search Api and Google Ajax Feed Api. We used google map api to display visitors graphically on a geographical map.

6.6 Cewolf tag library

Cewolf is an extension of JFreeChart available as tag library for use in JSP using taglib tag. Cewolf enables web applications running in a servlet container to easily include chart images.

Guido Laures wrote the initial program and released it. While developing a web application which needed to provide some chart images Guido found out that there is no open source library which provides the ability to define chart images as JSP tags. So he started to write a simple tag library based on the famous JFreeChart library for standalone chart images

In June 2002 Guido decided to donate his library to the public and named it Cewolf.

7 Program Description

The application consists of javascript tracking code and application based on JSP and servlet technology.The program or application consists of following main parts.

7.1 finalwebalizer.js

It is a javascript tracking code which is supplied to each user for pasting in their web pages of the site to be tracked. This code sends information related to visit, visitor and page to the server side for storage in the database. This code primarily maintains three cookies in the visitor's computer.

- Long Persistent cookie: Persistent cookie persists even after session terminates.Usually, their expiry period is set at six months.It is maintained to keep track of multiple visits by a unique visitor.
- Short Persistent cookie:This cookie is set with 30 minutes expiry time. If the visitor remains idle for 30 minutes,the cookie expires and a visit is said to be terminated.
- Session cookie: Session cookie persists throughout the life of a browser session . A browser session is equivalent to one visit.

7.2 trackgif.java

It is a servlet that processes the information supplied by finalwebalizer.js and stores them in the server database.

7.3 reporting.java

It is a servlet used to change the timeline of the reporting period while the user views the reports.

7.4 country.java

It is a java bean used to extract longitude,latitude,city and country information from the ip address of the visitor.It uses Maxmind java api and geolitecity.dat database for the purpose.

7.5 map.jsp

It is a JSP file that uses Google Ajax Api for maps to display visitor information graphically on the geographical map.

7.6 other JSP files,javascript files,servlets and css

The rest of the files are used for application interface and display of reports.

8 Methodology

In order to achieve the specified objectives, we will adopt the following methodologies:

8.1 Study of existing system:

There are several web analytics software existing nowadays. We will   study their operational procedure and features and limitations if any so that we can improve our project.

8.2 Requirement analysis:

 Before starting any project, it is necessary to collect the necessary requirements for that project. So, we will analyze the requirements for developing the project and list them.

8.3 Design:

We will convert the requirements into design in this phase. Here we design a prototype for our project.

8.4 Programming:

 It is the main part of our project. We will use Java Server Pages (jsp) in the server side and Javascript in the client side as a programming tool.

8.5 Problem solving:

 In any stage, if we encounter a problem, we will refer with our teachers, seniors, friends and search solution in the internet.

9 Group Work Division

This project is a part of third year Engineering course offered at Pulchowk Campus affiliated to T.U.The project team consists of third year students from Computer Engineering faculty comprising of following members.

Mr. Nabin Maharjan

Mr. Omkar Ranjit

Mr. Surendra Maharjan.

The first half of the project was spent most of the time gathering resources, building ideas with necessary work and observing various web analytics programs. On the other half of the project, the ideas were implemented to finally engineer a working system "My Web Analyzer."

While working on the project ,especially in the implementation phase, we divided the project into smaller parts as interface design, tracking code coding, database coding, report presentation etc. and work on smaller part but keeping constant communication and sharing. This whole project is the outcome of our teamwork.

10. Application Implementation

10.1 Login and logout

Users can login the application to view the reports about their site .They can log out after they have finished viewing report.

10.2 Website Analysis by Visitor

The metrics like number of unique visitors, new visitors, return visitors are determined. Visitors are analyzed based on frequency of visits, duration of visit, hour of visit etc

10.3 Website Analysis by Page

The visitors to a website are analyzed on the basis of top-viewed, entry pages, exit pages and overall page summary.

10.4 Website Analysis by Location

The visitors to a website are analyzed based on their countries. The information is also displayed on the geographical map.

10.5 Website Analysis by System

The visitors are analyzed based on operating system, browser, screen resolution, color depth etc.

10.6 Data presentation through Charts

Different charts like pie-Charts, bar diagrams have been used to provide information in easily understandable graphical form.
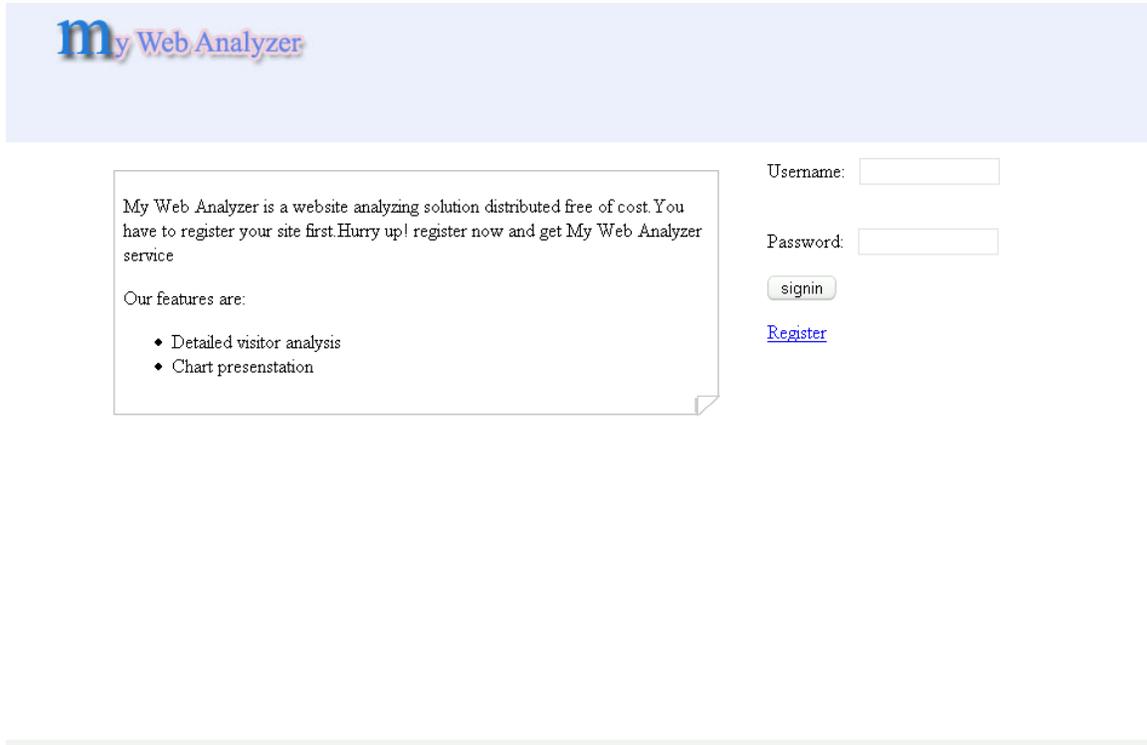
11. Application Screenshots

Fig1: index page of MywebAnalyzer
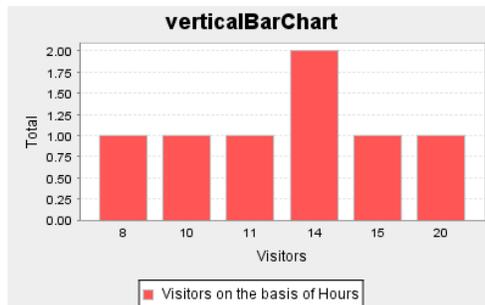
Fig2: Display of visitors in map

Fig3: Visitors on basis of time of visit

21

12. Analysis and testing

A successful project requires various testing. We have tested carefully for any bugs under local testing.The testings are satisfactory.

13. Expected Result

- User can track their website using My Web Analyzer

- User can view the reports about his site online.

14. Scope of the project

Since there are many analytics software packages that cost money that can be afforded by medium and large companies only. So, "My Web Analyzer" is more suitable for small organizations and individual site owners.

15. Limitation and Future Enhancement

- Dashboard can be modified according to the owner's choice.

- Analysis can be done using incoming and outgoing links.

- Visitor information about connection speed organization can also be included in the tracking code.

- Visitors are assumed to be human only

16. Project Schedule

As we know that everything needs to be scheduled before beginning any task. Therefore we have planned to develop our project according to the following time schedule:-

S.NO	PHASE	WEEKS
1	Study and Research	3
2	Requirements Analysis and Design	2
3	Coding	4
4	Testing, Verification and Maintenance	2
5	Documentation and Presentation	1

17. Conclusion

The software, "MywebAnalyzer" developed as the part of 3rd year minor project, can be used to generate the information about the website visitors. Thus it provides information in statistical and graphical form that helps the website owner to optimize his/her website effectively. It is suitable for small companies and the website owners who can not run their own webserver and can not pay for the available web analytic software packages.

18. Bibliography and References

- PDF: Google analytics
- PDF: Ajax in Action
- PDF: Introduction to Oracle 9i Sql
- PDF: WAA-Standards-Analytics-Definition
- PDF: Java Server Pages By: Larne Pekowsky
- PDF: Java Server Pages 1.2 By: F X Rohaan Perieira
- PDF: Web Object J2EE Programming Guide