CHAPTER 1

INTRODUCTION

1.1 The Web Application

The Computer Science and Information Technology has enormous role in development as well as application of scientific knowledge in every step of Education, Research, Business and other organizations. The first and foremost key for the growth of any Government/Private sectors across the globe is the implementation of Computer Science and Information Technology (IT) for transmission and reception of information.

In this cutting edge technology, the implementation of Web technology enables a company to run their activities for resources available to them or through its vendors. The Web based services for any organization is helpful in transmission and reception of information. There is a growing demand for core IT services like Web application development and maintenance for most of the organizations.

With the proper implementation of Web based services, it is possible to deliver one service standard to customer, regardless of where they based, from multiple locations worldwide. For decision making, in due course of action, an organization needs reliable and accurate data handling system. Different sectors as per their requirements, uses different software tools for acquiring required responses. An organization need to store adequate data and retrieve them with provision for addition and deletion of the same as and when required. The effective use of Web technology moves an organization to the next phase of growth for a better tomorrow. Since the inception of the Internet, it became a huge and most widely used platform for various applications with different objectives to serve the needs of different individuals, associated with various organizations. The World Wide Web (WWW), which is the most popular application on the Internet, has shown exponential growth in the past two decades or so. With this growth, the Web creates new challenges for information retrieval. Aside from tremendous growth, the Web has also become increasingly
commercial over time. In 1993, 1.5% of Web Servers were on .com domains. This number grew up to over 60% in 1997 [1].

1.2 The development of Internet

During the end of the year 1960, the use and sharing of computers were expensive and had limited resources. In order to share resources and communicate between computers amongst scientists as well as researchers, the government of United States of America (USA), took a very important step to start research on computer networking. The project was directed and sponsored by the Advanced Research Project Agency (ARPA) and named as ARPANET. In 1969, ARPANET linked its first four nodes: (i) the University of California, Los Angeles (UCLA), (ii) the Stanford Research Institute (SRI), (iii) the University of Utah (UU), and (iv) the University of California, Santa Barbara (UCSB) and the budding Internet was off the ground. From this point, the work on the underlying network and how to utilize it was started, which continues till today. The ARPANET was based on packet switching network architecture, where data was transmitted in discrete units, called packets.

Thus, computers were added quickly to the ARPANET during the following years. By the year 1975 the ARPANET grew to about 100 nodes. In the same year, ARPA handed over the project to the Defence Communication Agency of the USA for operational management.

The first recorded networking enabled social interaction was described by J.C.R. Licklider of MIT in August 1962 [2]. He visualized a set of globally interconnected computers which can help people to access data and programs from any site. This concept was the beginning of the Internet today. Licklider was the first head of the computer research project at Defence Advanced Research Projects Agency (DARPA), which started in October 1962. The successors at DARPA were convinced about the theoretical feasibility of packet switching rather than circuit switching, which was the major step towards evolving the computer networks. The first wide area computer network was provided with a low speed dial up telephone line. The eventual result was the realization that the time shared computers can work well together, running programs and retrieving data as and when required on the remote computer.
However, the researchers realized that circuit switching telephone system was totally inadequate for this purpose and the need of packet switching was confirmed.

The original ARPANET grew into Internet. The multiple independent networks of open architecture networking with arbitrary design were the basic idea behind the development of Internet. The packet switching network beginning with ARPANET was soon included with packet satellite networks, ground-based packet radio networks, and other networks. In this approach, the choice of any individual network technology was not dictated by particular network architecture. Rather, it could be selected freely by a provider and made available to interwork with the other networks through a meta-level, known as "Internetworking Architecture". In an open-architecture network, the individual networks may be separately designed and developed. Each network may have its own unique interface, which they may offer to users and/or other providers, including other Internet providers. Each network can be designed in accordance with the specific environment and user requirements. There are generally no constraints on the types of network that can be included or on their geographic scope, although certain pragmatic considerations will dictate what makes sense to offer.

The idea of open architecture network was first introduced at DARPA in 1972 [3]. The program was called Internetting. But the idea had lacked some important features, e.g. (a) it could not address networks or identify machines, (b) if any packets were lost, the protocol would come to a grinding halt. Thus, a new version of the protocol was developed, which could meet the needs of an open-architecture network environment. This protocol was called the Transmission Control Protocol/Internet Protocol (TCP/IP). It was later accepted as a world-wide standard for data communication. The TCP/IP uses the client-Server model of communication. This communication is mainly point to point stateless one. It is stateless because each client request is independent from the previous one.

1.2.1 The Local Area Network

The emergence of local area network (LAN) is the result of moving packets across different networks. A LAN is a data communications network that is geographically limited to a single building or a cluster of buildings. The devices connected to the LAN can receive the communication with other networks through
gateways. A gateway is a point in the network that acts as a point of entry to another network. The gateways have devices like protocol translators which will help in the communication between networks having different protocols. Also, the destination addresses of the packets are checked in the gateway and forwarded them to the desired network.

1.2.2 The Metropolitan Area Networks
The metropolitan area network (MAN) covers a city. Cable television network is the best known example of MAN available. The system grew from earlier community antenna systems used in areas with poor over-the-air television reception. With the Internet being popular to the common people, with some changes to the system, two-way Internet service is provided in the unused parts of the spectrum.

1.2.3 The Wide Area Networks
The wide area network (WAN) often covers a large geographical area, often a country or continent. It contains a collection of machines intended for running programs which are called hosts. The hosts are connected by a communication subnet. The hosts are owned by a customer, whereas the communication subnets are owned and operated by the Internet Service Provider (ISP). The subnet carries messages from host to host. It consists of two components, transmission lines and switching components. The transmission lines can be made up of copper wire, optical fiber, or even radio links. They move bits between machines. The switching elements are specialized computers which connects three or more transmission lines. The switching components are called routers and they decide the destination path of an incoming data.

1.2.4 The Wireless Networks
The wireless network started in the year 1901 with the Italian physicist G. Marconi who demonstrated a ship to shore wireless telegraph using Morse code. Wireless networks can be divided into three main categories: (i) the system interconnection, (ii) the wireless LAN (WLAN) and (iii) the wireless MAN (WMAN). In system interconnection the computer components such as monitor, keyboard, mouse, printers etc. can be connected without wire through Bluetooth devices. In case of
WLAN, every computer has a radio modem and an antenna with which it can communicate with other systems. They can communicate to each other in a peer-to-peer configuration or through a common antenna, placed at a suitable place for all the components. The IEEE 802.11 is a standard for WLAN which is implemented in most of the systems. The radio network used for cellular telephone is an example of low bandwidth WMAN. The Cellular wireless networks are much similar to WLAN, except that the distance covered is much greater and the bit rates are much lower.

1.2.5 The WWW

The reason which made the Internet known to the common people apart from scientists and researchers are the emergence of the WWW. The WWW is a large distributed hypertext repository of information, where people navigate through links and view pages using browsers [4]. In the year 1989 Tim Berners-Lee proposed how the WWW can operate. He created a hyperlinked database system for use in connecting databases across a closed network. In the 1990, he coined the term WWW [5]. Apart from TCP/IP, the Web has got its own Hypertext Transfer Protocol (HTTP) for sending and receiving documents. The Hypertext Markup Language (HTML) is the language of the Web and the Universal Resource Identifier (URI) later on known as Universal Resource Locator (URL) as its global naming scheme. The Mosaic X was the first browser developed for the Web. In the year 1994 the WWW Consortium (W3C) was founded. The W3C was responsible for developing and maintenance of various standards which helps in effective data communication and storage of information on the Web.

1.3 The Domain Name System

With the growing number of networks and nodes in the Internet, another problem essential to be resolved was addressing the nodes. Each computer on the Internet is assigned a 32 bit IP address. This IP address was interpreted as two components, (a) a network number which identifies a particular network on the Internet, and (b) a host address that identifies a specific host on the network. The routers keep a master table of addresses of all the networks on the Internet. With the
number of networks growing, it becomes cumbersome for the routers to keep the table updated with consistent.

To address this problem, the concept of the Domain Name System (DNS) came to the mind of the researchers. A domain is a group of hosts under the administrative group of a single entity. Rather than a numerical number (IP) each domain is given a name. The domains are hierarchically organized [6].

1.4 The Technology Overview

In its inception, Web applications were static and essentially read only hypermedia repositories of information. The applications running nowadays on the Web introduce a number of innovative aspects. The natures of operations have extended to new dimensions. Two main operations related to traditional hypermedia operations are (i) the browsing and (ii) the navigation. With user's demand, these two has extended to data entry, business transactions, communications with remote users etc. Thereafter, the tight integration among the different operation paradigms arises. Thus a Web application performs various operations chosen from the point of user's perspective. It has created the need for a new design paradigm. This new paradigm can be approached in two different ways, (a) a Web application can be considered as an extension to the notion of a traditional information system and (b) a Web application is regarded as an extension to a traditional Web site [7].

With the growing acceptance of the Web, Conseil Europeen pour la Recherche Nucleaire (CERN) released its implementation to the public in April 1993, allowing organizations to experiment with the Web. The National Center for Supercomputer Applications (NCSA) released the first Web browser NCSA Mosaic 1.0 with a graphical user interface (GUI) in September 1993 [8]. The GUI browser was very popular and successful which led to the acceptance of Internet by the public and hence demand for dynamic content, improved interfaces, and connectivity to the external system.

The age of purely static Web application was slowly vanishing with the introduction of Common Gateway Interface (CGI) in the year 1993. The CGI was first widely used for interacting Web Servers with external systems that processes data, submitted through HTML forms. Soon, it gains popularity for generating dynamic
content. The main disadvantage of using CGI was that for every running request, a new process is created. It was found to be inefficient in handling concurrent client requests and therefore applicable only to low traffic applications.

With the significant limitations of CGI, many Server side techniques were developed, such as Fast Common Gateway Interface (FastCGI), Applets, Server-Side Includes (SSI), ColdFusion, Server-side JavaScript etc. Later on, some other important technologies were developed, e.g. Active Server Pages (ASP). The ASP is a Server-side scripting environment that provides a scripting language engine which supports several languages, an object model and an interface for accessing Server components. The ASP was primarily applicable only to Microsoft systems. The emergence of ASP was a step forward towards the development of dynamic Web content generation. The limitations of ASP were portability, scalability, integration, reliability, and usability.

The HyperText Preprocessor (PHP) is an open source, cross-platform, object-based scripting language for dynamic content generation which was similar to ASP. The PHP was run on many operating systems and Web Servers. The PHP is available for Windows operating system as well as for Linux. Comparing the relative advantages and disadvantages of PHP with those of ASP, the portability of PHP takes it to a superior position than ASP.

With the demand as well as necessity of large scale Web applications, the architecture that uses an application Server and middleware which separates the Web Server, presentation logic, business logic, and data access concerning different tiers was evolving. With an environment, supporting enterprise-class Web applications, the programming language requirement changed significantly.

The Sun Microsystems developed Java, which was garbage-collected, object-oriented, popular programming language. The Java Virtual Machine (JVM) of Java, which supports portability on different platforms, found to be an important feature for the development of the most diverse computing environment available.

The problems e.g. performance, management, and standardized access to middleware were addressed by Servlets which was introduced in 1997. The Servlets are executed by servlet containers that manage threads through their lifecycle. The use of threads instead of processes improved efficiency and scalability. This is a major factor behind the emergence of Java as an important Server side Web programming language
which is continuing till date. The disadvantage of Servlet programming was having a thin abstraction over HTTP.

To overcome the limitations of Servlets, Java Server Pages (JSP) was introduced in the year 1999. The JSP was popular among developers, because it is simpler than Servlets to develop Web applications. Initially JSP seems to be supported role in separation between designers and programmers. But in practice, presentation is not adequately separated from behavior, leading to maintenance problems in large applications. As such, better techniques than JSP were in search.

Since its initial product release in 2001, .NET which is a set of products that has been steadily gaining popularity amongst developers. The .NET is based on Microsoft operating systems, though efforts were on to bring .NET to Linux. The .NET provides an environment for enterprise Web applications that is comparable to Java technologies.

With such development, the ASP.NET and Java Server Faces (JSF) are two major techniques for developing Web applications. The ASP.NET is a revised version of ASP that makes use of the capabilities of .NET framework. It is a page driven object model for Web programming. The JSF is a specification for a component based Web application framework.

The scenario is changed. With time, Web applications become more sophisticated, interactive programs with complex GUIs having large amount of backend software that is integrated in novel and interesting ways. These Web applications need to be highly secured, reliable, continually maintainable, and constantly available. The Web application malfunction can affect hundreds of thousands of people and can cost millions of dollars. A glitch during an unscheduled maintenance at Amazon.com in 1998 put the site offline for several hours, with an estimated cost of $400,000. Such a situation can seriously damage the relationship between customer and the service provider. These requirements are being responded by developing new technologies and programming models that impact the way through which the software is designed, built, tested, and maintained.

The Web applications are built with different software technologies. These include, scripting languages that run on the client, JavaScript and VBScript, Server side languages, e.g. Servlets, scripted page modules on the Server, e.g. JSPs and ASPs,
general purpose programming languages, e.g. Java and C#, programming language extensions JavaBeans, Enterprise JavaBeans (EJB), data manipulation languages eXtensible Markup Language (XML) and databases. Along with others, these diverse technologies work together to implement Web applications [9].

The model view controller (MVC) is a well known design pattern to architect interactive Web applications. It is also known as Presentation/Abstraction/Control (PAC). The key idea of this design pattern is to separate the User Interfaces (UI) from the underlying data representation by the UI. In MVC, the view displays information to the user. The model is that part of the application which contains both the information represented by the view and the logic that changes this information in response to user interaction. The controller is the entry point of the application. The use of MVC makes it easier to develop and maintain an application, because, (i) the application's presentation can be drastically changed without changing data structures and business logic and (ii) the application can easily maintain different interfaces, such as multiple languages or different sets of user permissions.

With these added advantages, the MVC pattern has got some disadvantages also. One of them is that, the applications needed to be intrinsically partitioned between client and Server. The view is being displayed in the client side. The contrast, being resided and executed in the same address space, and is partitioned independent. But the general concept of the partitioning is location-dependent. This condition will make it to apply MVC design pattern in Web applications much harder. However, developers can make partitioning easier by deciding which method will run in Server and which will run in the client. But the problem with this aspect is that it is not always possible to make correct decision regarding partitioning prior to the design phase [10].

Interactive Web applications are benefited by using MVC design pattern. For example, one can use JavaBeans (managed beans) as model, JSP and HTML as view and servlet as the controller.

The Web application is the most common type of applications for presenting dynamic content. There have been numerous studies on dynamic Web content. Scott Trent, Michiaki Tatsubori et al have studied the performance comparison of PHP and JSP [11]. The PHP and JSP are acting as the Server side scripting language. The scripts are used to generate dynamic pages rather than the implementation of complex business
logic. The detailed analysis of PHP and JSP performance based on SPECWeb2005 benchmark enables designers and implementers of Web Servers to understand the relative performance and limitations of these two types of techniques. Since the year 1995, Yeager and McGrath have studied the performance of dynamic Web content generators, researchers have been examining the performance of more recent dynamic Web content generation techniques.

The comparison in developing Web applications using Java and PHP was carried out by Fredrik Johansson, the performance result was also studied [12].

A number of standard mechanisms can be used to generate dynamic content. Three mechanisms are in common, (a) the PHP, (b) the Java Servlets and (c) the EJB. These three mechanisms have different architectures for generating dynamic Web content. In PHP and Java Servlets, Structured Query Language (SQL) queries can be embedded. In EJB, a number of beans are available. The entity beans are responsible for handling persistent services like database access. Two applications have been implemented, one online bookstore and the other an auction site to measure the performance of these three architectures. It is concluded that PHP is more efficient than that of Java Servlets. The EJB’s performance is the worst among these three. However, they concluded that the Servlets which is a part of the Java technologies has the flexibility of working with different operating systems. The detail results of performance comparison are presented and discussed by E. Cecchet et al. [13].

Dynamically generated Web content can have significant impact on Web Server performance, hampering the scalability of the application. Three different popular technologies for developing dynamic content are used and implemented, they are Perl, PHP, and Java. They performed the required studies and inferred that the personalization of Web content comes at the price of overhead for database access, and from the processing, required for dynamic content generation. Today’s technologies for dynamic content generation show distinct tradeoffs in terms of Web Server performance. The PHP handles small dynamic content request well, but struggles with large content generation. The Web Servers Jetty, Resin, and Tomcat are not well suited for static content generation, but perform better in dynamic content generation. They also inferred that the Web Server performance under overload can be quite
unpredictable. L. Titchkosky et al have performed the required tests, analyzed, and concluded that Java Server technologies outperform both PHP and Perl [14].

Three versions of an online Web application were developed using Model 2, Struts, and JSF. The parameters compared are the number of lines of codes, the number of classes, and the performance measurements. The design model which offers the most rapid development process, that gives better performance, were investigated and reported. The performance measurement results are taken from the three operations of each version of the application: (i) the search, (ii) the browse and (iii) the shopping operation. One SQL SELECT operation is performed with the search operation. The browse operation retrieves all products in a category by performing one SQL SELECT statement. The shopping operation consists of the INSERT statements. The JSF outperforms both Model 2 and Struts in ease of developing application. The performance comparison between these three shows that, Model 2 and Struts perform similarly, while the performance of JSF is worse. B. Kurniawan and J. Xue reported the detailed analysis result [15].

To control and improve the quality of Web applications, a high demand for methodologies and tools for quality assurance of Web based applications is emerging. As a consequence, a Unified Modeling Language (UML) of Web application is proposed which helped in the assessment of static Web application structure. Analyzing results is crucial, which can make understanding and modification possible in a system.

To achieve high quality standard, the usual black-box and white-box testing is incorporated. An analysis model is defined and adopted for the high level representation of Web applications. The usefulness of the two proposed methods implemented by the tools ReWeb and TestWeb are discussed by Filippo Ricca and Paolo Tonella [16]. Two real world Web applications Wordnet and Amazon were analyzed using these two models. The analysis model proposed here are more concerned with navigation and interaction pattern over other architectural perspective. The ReWeb's views were useful to understand the site organization, both in terms of navigation paths, and of variable usage i.e. data flow views. The TestWeb’s generator and executor of test cases were exploited to exercise the two sites up to a satisfactory level of coverage. Some anomalies in the behavior of Amazon were revealed during the testing procedure.
The CGI, Java Servlets, and PHP3 were applied to implement a project [17]. The goal was to investigate their performance difference. Both non-persistent and persistent database connection models have been tested on the Servlets solution and PHP3. The design of the application was three tier client/Server architecture: (a) the client, (b) the Server and (c) the backend database Server. The conclusion was that the Java Servlets with persistent connection has the best performance. The PHP3 with persistent database connections performs fairly well when compared to the CGI. Though the non-persistent database connection shows some performance penalty, still they are better when compared with CGI.

The key quality attributes of distributed applications, such as response time, throughput, and scalability need to be evaluated. The evaluation of performance in terms of the above mentioned factors is crucial in early stage of development, when important choices are made for the architecture of the application. A novel approach for performance testing of distributed applications is presented by G. Denaro et al [18]. Their work suggested for making the right choice of using middleware to best satisfy the performance requirement of a distributed application. They have studied application-specific test cases for early performance evaluation and compared the performance of distributed applications in specific deployment environments. The deployment environment includes middleware, databases, operating systems, and other off-the-shelf components.

One of the most important factors in Web application is the quality assurance. Due to the complexity of the underlying technologies of Web applications, it is more challenging to test Web application than conventional other type of applications. A rigorous and automatic testing approach for Web applications using formal specifications has been proposed by X. Jia et al [19]. The new approach intends to address testing of Web applications such as functionality, security and performance. The proposed new prototype tool accepts formal specifications in XML syntax as input, automatically generate test cases, executes the test cases, and validates that test results. The proposed tool was called WebTest which was successful to perform the testing of Web applications. The WebTest demonstrated the feasibility and effectiveness of the tool.
The Web Server performance is critical in success of the Web application. A. Nanda et al measured and analyzed the behavior of the Apache Web Server on a 4-Central Processing Unit (CPU) Symmetric Multi-Processor (SMP) system running on International Business Machine Advanced Interactive Executive (IBM AIX) operating system [20]. The result shows that the Apache spends about 20-25% of the total CPU time on user code, 30-35% on kernel system calls and 25 – 40% on interrupt handling. For systems with small Random Access Memory (RAM) sizes, the Web Server performance is limited by the disk bandwidth, with larger RAM sizes, the TCP/IP stack and the network interrupt are the major performance bottlenecks. They proposed 8 techniques to improve the performance of Apache. On implementing 7 of these techniques, the experimental results show that these techniques improve the throughput of Apache by 61%. These techniques can be applied to other Web Servers as well.

The statistical analysis of Web application is a useful compliment to structure testing. This approach accounts for the typical interactions with the Web application rather than its structure and data flows. The main benefit of statistical testing is that it allows using the statistical inference techniques to compute probabilistic aspects of the testing process such as reliability and stability [21].

The statistical testing of software establishes a basis for statistical inference about a software system's expected field quality. A method for statistical testing based on Markov chain model of software is described by James A. Whittaker et al. [22].

Web applications are cross platform, universally accessible resources for massive user population. Consequently, quality assurance of Web applications is becoming very important. For the effective quality assurance of Web applications, one good candidate is statistical testing and related reliability analysis. A hierarchical approach for statistical Web testing and reliability analysis was developed by C. Kallepalli et al [23]. The approach was called Unified Markov Model (UMM). The prerequisite of which was to collect large Web usage information and construction of corresponding usage models. For Web applications, various log files are routinely kept at Web Servers. The usage and failure information of these log files are extracted to support statistical testing and reliability analysis.

The performance is being the key point in the success of a Web application which is affected by several factors. The results of performance measurements of an
ASP.NET Web application are presented and discussed by Á. Bogárdi-Mészöly et al [24]. They measured the average response time of the Web application by changing the parameters of the application Server. During the measurement process, maximum and minimum numbers of certain threads in the thread pool are changed. Statistically they have searched the factors influencing the response time. Also they have determined the distribution of response time with the help of histograms, quantile-quantile plot, and normal probability plot. The details of configuration and analysis are also presented.

In enterprise level of Web application development, .NET and the Java 2 Enterprise Edition (J2EE) are two leading technologies in the present day scenario. The two platforms are compared based on the features present in the two, mainly tools and resources available [25]. Arguments in favor of J2EE were: (i) the platform independence, (ii) the multiple vendor support, (iii) the popularity of Java. The points favoring .NET include support for multiple languages and integrated for Web services (WS). The disadvantage of .NET is being the single vendor support, which is Microsoft.

From the above review, it is clear that the investigation on the implementation of Web application with different technique is important for Web developer as well as for Web users. As such we proposed to carry out an investigation on implementation of Web application taking into account of two different implementation techniques. The implementation choices under investigations are: (a) the .NET platform and (b) the Java 5 platform, Enterprise Edition (J5EE). Two prototype web applications have been implemented: one with .NET technique, we call it the PReWebD and the other with the Java technique, we call it the PReWebN to study the feasibility of the proposed work.

The details of the propose work is described in chapter 2 while in chapter 3 we discuss in details about the various tools which have been used for implementing the PReWebD and PReWebN. In chapter 4 we discuss in details the design and implementation aspects of PReWebD and PReWebN. In chapter 5 we discuss in details about the different testing performed on PReWebD and PReWebN. In chapter 6 we discuss in details the results of the testing performed on PReWebD and PReWebN. In chapter 7 we make our concluding remarks on the outcome of our present work and future aspect of the same.