A Project Report
on

<APTI GEEK >

Submitted in partial fulfillment of the requirements for the Major Project of

Bachelor of Technology
in
Computer Science & Engineering

Submitted by:

LOVELY SINGHAL	**R780209039**
SURABHI RASTOGI	**R780209042**

Under the guidance
Mr. HITESH KUMAR
Assistant Professor

Department of Computer Science & Engineering
COLLEGE OF ENGINEERING STUDIES
UNIVERSITY OF PETROLEUM & ENERGY STUDIES
Dehradun- 248007
April 2013

CERTIFICATE

This is to certify that the Project entitled **"APTI GEEK"** submitted by

LOVELY SINGHAL **R780209039**

SURABHIRASTOGI **R780209042**

for the partial fulfillment of the requirements of the course **Major Project** of **Bachelor of Technology in Computer Science & Engineering** degree of **University of Petroleum & Energy Studies, Dehradun** embodies the confide work done by above students under my supervision.

42
30/4/12

Signature of Mentor
Assistant Professor

N. Pradhan.
30/04/2013.

DECLARATION

We, Lovely Singhal, Surabhi Rastogi bearing the Roll No: R780209039, R780209042 respectively hereby declare that this Project work entitled "Apti Geek" was carried out by us under the guidance and supervision of Mr. Hitesh Kumar. This Project work is submitted to University of Petroleum & Energy Studies in partial fulfilment of the requirement for the award of Bachelor of Technology in Computer Science and Engineering during the Academic Semester Jan 2013 - Apr 2013. We also declare that, we have not submitted this dissertation work to any other university for the award of either degree or diploma.

Lovely

Lovely Singhal

Place: Dehradun

Surabhi

Surabhi Rastogi

Date: 26/04/2013

ABSTRACT

" ABSTRACT OF APTIGEEK"

"APTI GEEK" is a web based application which is capable of conducting online aptitude tests on the internet. This site is designed to assess logical reasoning or thinking performance. Aptitude Tests will be divided into 3 sections.

First section consists of Quantum i.e. purely mathematical questions, Second section deals with verbal reasoning & third section consists of questions based on data interpretations that would be the toughest one. They consist of multiple choice questions. These are typically timed and a typical test might allow 30 minutes for 30 questions and so on. The test will be conducted on the basis of difficulty level. i.e. the user will enter in the first level and then if he clears it then he will reach to the next level according to more difficulty.

The result of the aptitude test will be displayed and it will be stored in the database for any future use. This project of conducting Online Aptitude Test has much scope in today. Today almost everything is performed electronically hence "Apti Geek" provides a good platform for aptitude assessment without paper work. Furthermore it also provides the storage and retrieval of result effectively. Hence paperwork is reduced and manual checking is also not required.

ACKNOWLEDGEMENT

It is a pleasure to thank all those great many people who helped, supported and encouraged us during this project work.

Firstly we express our sincere gratitude to Mr. Hitesh Kumar sir, the guide of the project who carefully and patiently leant his valuable time and effort to give directions as well as to correct various documents with attention and care.

It is a great honour to do this project in this esteemed institution, and we would extend our thanks to the HOD, Prof Manish Prateek and other faculty members who have shared their vast knowledge and experience during our stay.

We do also like to appreciate the consideration of the Project Coordinator, our Faculties and colleagues, which enabled us to balance our work along with this project. It was their attitude that inspired us to do such an efficient and apposite work.

We are indebted to those people across the globe who have shared their knowledge and perspectives in the form of online tutorials, forums and other resources which helped us to a great extend whenever we met with technical obstacles during this endeavour.

We wish to avail this opportunity to express a sense of gratitude and love to all our friends and our family for their unwavering support, strength, help and in short for everything they have done during the crucial times of the progress of our project.

Last but not the least we thank GOD ALMIGHTY for HIS blessings and guidance without which this dream project wouldn't have been reality.

LOVELY SINGHAL

SURABHI RASTOGI

CONTENTS

LIST OF FIGURES AND TABLES

CHAPTER 1 INTRODUCTION

1.1 What is AptiGeek?

"APTI GEEK" is a web based application which is capable of conducting online aptitude tests on the internet. This site is designed to assess logical reasoning or thinking performance. Aptitude Tests will be divided into 3 sections.

First section consists of Quantum i.e. purely mathematical questions, Second section deals with verbal reasoning & third section consists of questions based on data interpretations that would be the toughest one. They consist of multiple choice questions. These are typically timed and a typical test might allow 30 minutes for 30 questions and so on. The test will be conducted on the basis of difficulty level. i.e. the user will enter in the first level and then if he clears it then he will reach to the next level according to more difficulty.

The result of the aptitude test will be displayed and it will be stored in the database for any future use. This project of conducting Online Aptitude Test has much scope in today. Today almost everything is performed electronically hence "Apti Geek" provides a good platform for aptitude assessment without paper work. Furthermore it also provides the storage and retrieval of result effectively. Hence paperwork is reduced and manual checking is also not required.

1.2 Why XAMPP?

XAMPP is a free and open source cross-platform web server solution stack package, consisting mainly of the Apache HTTP Server, MySQL database, and interpreters for scripts written in the PHP and Perl programming languages. XAMPP requires only one zip, tar, exe file to be downloaded and run, and little or no configuration of the various components that make up the web server is required. XAMPP is regularly updated to incorporate the latest releases of Apache/MySQL/PHP and Perl. It also comes with a number of other modules including OpenSSL and phpMyAdmin. XAMPP's designers intended it for use only as a development tool, to allow website designers and programmers to test their work on their own computers without any access to the Internet. To make this as easy as possible, many important security features are disabled by default. however, XAMPP is sometimes used to actually serve web pages on the World Wide Web. XAMPP also provides support for creating and manipulating databases in MySQL and SQLite among others.

XAMPP requires only one zip, tar, 7z, or.exe.file to be downloaded and run, and little or no configuration of the various components that make up the web server is required. XAMPP is regularly updated to incorporate the latest releases of Apache/MySQL/PHP and Perl. It also comes with a number of other modules including OpenSSL and phpMyAdmin. Self-contained, multiple instances of XAMPP can exist on a single computer, and any given instance can be copied from one computer to another.It is offered in both a full, standard version and a smaller version.

Officially, XAMPP's designers intended it for use only as a development tool, to allow website designers and programmers to test their work on their own computers without any access to the Internet. To make this as easy as possible, many important security features are disabled by default.[2] In practice, however, XAMPP is sometimes used to actually serve web pages on the World Wide Web[*citation needed*]. A special tool is provided to password-protect the most important parts of the package.[3]

XAMPP also provides support for creating and manipulating databases in MySQL and SQLite among others.

Once XAMPP is installed, it is possible to treat a localhost like a remote host by connecting using an FTP client. Using a program likeFileZilla has many advantages when installing a content management system (CMS) like Joomla. It is also possible to connect to localhost via FTP with an HTML editor. The default FTP user is "newuser", the default FTP password is "wampp".

The default MySQL user is "root" while there is no default MySQL password.

XAMPP's name is an acronym for:

- X (to be read as "cross", meaning cross-platform)
- Apache HTTP Server
- MySQL
- PHP

Apache HTTP Server :

The Apache HTTP Server, commonly referred to as Apache is a web server software notable for playing a key role in the initial growth of the World Wide Web. The application is available for a wide variety of operating systems, including Unix, FreeBSD, Linux, Solaris, Novell NetWare, OS X, Microsoft Windows, OS/2, TPF. Released under the Apache License, Apache is open-source software.

Apache, otherwise known as **Apache HTTP Server**, is an established standard in the online distribution of website services, which gave the initial boost for the expansion of the World Wide Web. It is an open-source web server platform, which guarantees the online availability of the majority of the websites active today. The server is aimed at serving a great deal of widely popular modern web platforms/operating systems such as Unix, Windows, Linux, Solaris, Novell NetWare, FreeBSD, Mac OS X, Microsoft Windows, OS/2, etc.

There are two popular versions for the origin of the project's name. According to the more widespread one - the web server's name was chosen out of respect for the Native American Indian Apache tribe, well known for its resilience and military skills. The other version refers to the set of patches attached to the codebase of NCSA HTTPd 1.3, which makes it "a patchy" server.

Apache supports a variety of features, many implemented as compiled modules which extend the core functionality. These can range from server-side programming language support to authentication schemes. Some common language interfaces support Perl, Python, Tcl, and PHP. Apache has a built in search engine and an HTML authorizing tool and supports FTP.

Because the source code is freely available, anyone can adapt the server for specific needs, and there is a large public library of Apache add-ons. Although the main design goal of Apache is not to be the "fastest" web server, Apache does have performance similar to other "high-performance" web servers. Instead of implementing a single architecture, Apache provides a variety of MultiProcessing Modules (MPMs) which allow Apache to run in a process-based, hybrid (process and thread) or event-hybrid mode, to better match the demands of each particular infrastructure. The design of Apache is to reduce latency and increase throughput, relative to simply handling more requests, thus ensuring consistent and reliable processing of requests within reasonable time-frames.

Popular compression methods on Apache include the external extension module, mod_gzip, implemented to help with reduction of the size (weight) of web pages served over HTTP. ModSecurity is an open source intrusion detection and prevention engine for web applications. Apache logs can be analyzed through a web browser using free scripts such as AWStats/W3Perl or Visitors.

Virtual hosting allows one Apache installation to serve many different actual websites. For example, one machine with one Apache installation could simultaneously serve www.example.com, www.example.org, test47.test-server.example.edu, etc.

Apache features configurable error messages, DBMS-based authentication databases, and content negotiation. It is also supported by several graphical user interfaces (GUIs).

It supports password authentication and digital certificate authentication. Apache has a built in search engine and an HTML authorizing tool and supports FTP. The Apache server has been developed by an open source community - Apache Software Foundation, whose members are constantly adding new useful functionalities, with the sole purpose of providing a secure and extensible server platform that ensures HTTP service delivery in accordance with the current HTTP standards.

There is a great amount of modules created especially for the Apache server, which support various scripts and allow dynamic content to be run on the server. Most of the modules come as part of the Apache distribution, making the server boast a wide range of capabilities, such as support for CGI (Common Gateway Interface), a standard protocol for communication between external application software and a web server, and also SSI (Server Side Includes), a simple server-side scripting language. Other Apache modules include: CGI scripts execution, user authentication, URL redirection, anonymous user access, automatic directory listings, support for HTTP header metafiles, support for loading modules, content negotiation, caching proxy abilities, server status display, user home directories, etc. Users can choose to install those modules with the Apache server installation. If not, they can install them later with the help of dynamic modules.

The Apache server has been the most popular web server on the Internet since April 1996. It is by no means considered a platform criterion for the development and evaluation of other successful web servers. Because the source code is freely available, anyone can adapt the server for specific needs, and there is a large public library of Apache add-ons.

The open source software movement has received enormous attention in the last several years. It is often characterized as a fundamentally new way to develop software that poses a serious challenge to the commercial software businesses that dominate most software markets today.

Although the main design goal of Apache is not to be the "fastest" web server, Apache does have performance similar to other "high-performance" web servers. Instead of implementing a single architecture, Apache provides a variety of MultiProcessing Modules (MPMs) which allow Apache to run in a process-based, hybrid (process and thread) or event-hybrid mode, to better match the demands of each particular infrastructure. This implies that the choice of correct MPM and the correct configuration is important. Where compromises in performance need to be made, the design of Apache is to reduce latency and increase throughput, relative to simply handling more requests, thus ensuring consistent and reliable processing of requests within reasonable time-frames.

The Apache 2.2 series was considered significantly slower than nginx for delivering static pages, although remaining significantly faster for dynamic pages. To address this issue, the Apache version considered by the Apache Foundation as providing high-performance is the multi-threaded version which mixes the use of several processes and several threads per process.This architecture, and the way it was implemented in the Apache 2.4 series, provides for performance equivalent or slightly better than event-based webservers

MySQL :

MySQL is the world's most used open source relational database management system (RDBMS) as of 2008 that runs as a server providing multi-user access to a number of databases. MySQL is a popular choice of database for use in web applications, and is a central component of the widely used LAMP open source web application software stack (and other 'AMP' stacks) . LAMP is an acronym for "Linux, Apache, MySQL, Perl/PHP/Python." Free-software-open source projects that require a full-featured database management system often use MySQL. MySQL is an RDBMS and ships with no GUI tools to administer MySQL databases or manage data contained within the databases. Users may use the included command line tools, or use MySQL "front-ends", desktop software and web applications that create and manage MySQL databases, build database structures, back up data, inspect status, and work with data records.

The official set of MySQL front-end tools, MySQL Workbench is actively developed by Oracle, and is freely available for use. MySQL is written in C and C++. Many programming languages with language-specific APIs include libraries for accessing MySQL databases.

MySQL can be built and installed manually from source code, but this can be tedious so it is more commonly installed from a binary package unless special customizations are required. MySQL is a database server. MySQL is ideal for both small and large applications. MySQL supports standard SQL. MySQL compiles on a number of platforms. MySQL is free to download and use. MySQL ships with many command line tools, from which the main interface is 'mysql' client. Third-parties have also developed tools to manage, optimize, monitor and backup a MySQL server, some listed below. All these tools work on *NIX type operating systems, and some of them also on Windows.

- Maatkit - a cross-platform toolkit for MySQL, PostgreSQL and Memcached, developed in Perl. Maatkit can be used to prove replication is working correctly, fix corrupted data, automate repetitive tasks, and speed up servers. Maatkit is included with severalGNU/Linux distributions such as CentOS and Debian and packages are available for Fedora and Ubuntu as well. As of late 2011, Maatkit is no longer developed, but Percona has continued development under the Percona Toolkit brand.

- XtraBackup - Open Source MySQL hot backup software. Some notable features include hot, non-locking backups for InnoDB storage, incremental backups, streaming, parallel-compressed backups, throttling based on the number of IO operations per second, etc.

- MySQL::Replication - a replacement for MySQL's built-in replication, developed in Perl. MySQL::Replication can be used to create a peer-to-peer, multi-master MySQL replication network.

- MySQL works on many different system platforms, including AIX, BSDi, FreeBSD, HP-UX, eComStation, i5/OS, IRIX, Linux, Mac OS X,Microsoft Windows, NetBSD, Novell NetWare, OpenBSD, OpenSolaris, OS/2 Warp, QNX, Solaris, Symbian, SunOS, SCO OpenServer, SCO UnixWare, Sanos and Tru64. A port of MySQL to OpenVMS also exists.

- The HTSQL - URL-based query method also ships with a MySQL adapter, allowing direct interaction between a MySQL database and any web client via structured URLs.

- Like other SQL databases, MySQL does not currently comply with the full SQL standard for some of the implemented functionality, including foreign key references when using some storage engines other than the 'standard' InnoDB.

- Triggers are currently limited to one per action / timing, i.e. maximum one after insert and one before insert on the same table. There are no triggers on views.

- MySQL, like most other transactional relational databases, is strongly limited by hard disk performance. This is especially true in terms of write latency. Given the recent appearance of very affordable consumer grade SATA interface Solid-state drives that offer zeromechanical latency, a fivefold speedup over even an eight drive RAID array can be had for a smaller investment.

- MySQL can be built and installed manually from source code, but this can be tedious so it is more commonly installed from a binary package unless special customizations are required. On most Linux distributions the package management system can download and install MySQL with minimal effort, though further configuration is often required to adjust security and optimization settings.

- Though MySQL began as a low-end alternative to more powerful proprietary databases, it has gradually evolved to support higher-scale needs as well. It is still most commonly used in small to medium scale single-server deployments, either as a component in a LAMP-based web application or as a standalone database server. Much of MySQL's appeal originates in its relative simplicity and ease of use, which is enabled by an ecosystem of open source tools such as phpMyAdmin. In the medium range, MySQL can be scaled by deploying it on more powerful hardware, such as a multi-processor server with gigabytes of memory.

- There are however limits to how far performance can scale on a single server, so on larger scales, multi-server MySQL deployments are required to provide improved performance and reliability. A typical high-end configuration can include a powerful master database which handles data write operations and is replicated to multiple slaves that handle all read operations. The master server synchronizes continually with its slaves so in the event of failure a slave can be promoted to become the new master, minimizing downtime. Further improvements in performance can be achieved by caching the results from database queries in memory using memcached, or breaking down a database into smaller chunks called shards which can be spread across a number of distributed server clusters.

PHP :

PHP is a powerful server-side scripting language for creating dynamic and interactive websites. PHP is the widely-used, free, and efficient alternative to competitors such as Microsoft's ASP. PHP is perfectly suited for Web development and can be embedded directly into the HTML code. The PHP syntax is very similar to Perl and C. PHP is often used together with Apache (web server) on various operating systems. PHP stands for **P**HP: **H**ypertext **P**reprocessor. PHP scripts are executed on the server. PHP supports many databases (MySQL, Informix, Oracle, Sybase, Solid, Generic ODBC, etc.). PHP is an open source software (OSS). PHP is free to download and use. PHP runs on different platforms (Windows, Linux, Unix, etc.). PHP is compatible with almost all servers used today (Apache, IIS, etc.).

The code is interpreted by a Web server with a PHP processor module which generates the resulting Web page. It also has evolved to include a command-line interface capability. PHP acts primarily as a filter, taking input from a file or stream containing text and/or PHP instructions and outputting another stream of data; most commonly the output will be HTML. PHP can be deployed on most Web servers, many operating systems and platforms, and can be used with many relational database management systems (RDBMS). It is available free of charge, and the PHP Group provides the complete source code for users to build, customize and extend for their own use. PHP has also attracted the development of many frameworks that provide building blocks and a design structure to promote rapid application development (RAD).

PHP stores whole numbers in a platform-dependent range, either a 64-bit or 32-bit signed integer equivalent to the C-language long type. Unsigned integers are converted to signed values in certain situations; this behavior is different from other programming languages. Integer variables can be assigned using decimal (positive and negative), octal, hexadecimal, and binary notations.Floating point numbers are also stored in a platform-specific range. They can be specified using floating point notation, or two forms ofscientific notation.

PHP has a native Boolean type that is similar to the native Boolean types in Java and C++. Using the Boolean type conversion rules, non-zero values are interpreted as true and zero as false, as in Perl and C++. The null data type represents a variable that has no value. The only value in the null data type is *NULL*. Variables of the "resource" type represent references to resources from external sources. These are typically created by functions from a particular extension, and can only be processed by functions from the same extension; examples include file, image, and database resources.

Arrays can contain elements of any type that PHP can handle, including resources, objects, and even other arrays. Order is preserved in lists of values and in hashes with both keys and values, and the two can be intermingled. PHP also supports strings, which can be used with single quotes, double quotes, nowdoc or heredoc syntax. The Standard PHP Library (SPL) attempts to solve standard problems and implements efficient data access interfaces and classes.

PHP has hundreds of base functions and thousands more via extensions. These functions are well documented on the PHP site; however, the built-in library has a wide variety of naming conventions and inconsistencies. PHP currently has no functions for threadprogramming, although it does support multi process programming on POSIX systems.

In PHP , functions are not first-class functions and can only be referenced by their name, directly or dynamically by a variable containing the name of the function. User-defined functions can be created at any time without being prototyped. Functions can be defined inside code blocks, permitting a run-time decision as to whether or not a function should be defined. Function calls must use parentheses, with the exception of zero argument class constructor functions called with the PHP new operator the create_function() function, although they are not true anonymous functions because anonymous functions are nameless, but functions can only be referenced by name, or indirectly through a variable $function_name();, in PHP.

Basic object-oriented programming functionality was added in PHP 3 and improved in PHP 4. Object handling was completely rewritten for PHP 5, expanding the feature set and enhancing performance. In previous versions of PHP, objects were handled likevalue types. The drawback of this method was that the whole object was copied when a variable was assigned or passed as a parameter to a method. In the new approach, objects are referenced by handle, and not by value.

PHP 5 introduced private and protected member variables and methods, along with abstract classes, final classes, abstract methods, and final methods. It also introduced a standard way of declaring constructors and destructors, similar to that of other object-oriented languages such as C++, and a standard exception handling model. Furthermore, PHP 5 added interfaces and allowed for multiple interfaces to be implemented. There are special interfaces that allow objects to interact with the runtime system. Objects implementing ArrayAccess can be used with array syntax and objects implementing Iterator or IteratorAggregate can be used with the foreach language construct. There is no virtual table feature in the engine, so static variables are bound with a name instead of a reference at compile time.

If the developer creates a copy of an object using the reserved word clone, the Zend engine will check if a __clone() method has been defined or not. If not, it will call a default __clone() which will copy the object's properties. If a __clone() method is defined, then it will be responsible for setting the necessary properties in the created object. For convenience, the engine will supply a function that imports the properties of the source object, so that the programmer can start with a by-value replica of the source object and only override properties that need to be changed

The PHP language was originally implemented as an interpreter, and this is still the most popular implementation. Several compilers have been developed which decouple the PHP language from the interpreter. Advantages of compilation include better execution speed, static analysis, and improved interoperability with code written in other languages. PHP compilers of note include Phalanger, which compiles PHP into Common Intermediate Language (CIL) bytecode, and HipHop, developed at

Facebook and now available as open source, which transforms the PHP Script into C++, then compiles it, reducing server load up to 50% . PHP source code is compiled on-the-fly to an internal format that can be executed by the PHP engine. In order to speed up execution time and not have to compile the PHP source code every time the web page is accessed , PHP scripts can also be deployed in executable format using a PHP compiler.

Code optimizers aim to enhance the performance of the compiled code by reducing its size , merging redundant instructions and making other changes that can reduce the execution time. With PHP, there are often opportunities for code optimization. An example of a code optimizer is the eAccelerator PHP extension. Another approach for reducing compilation overhead for PHP servers is using an opcode cache. Opcode caches work by caching the compiled form of a PHP script (opcodes) in shared memory to avoid the overhead of parsing and compiling the code every time the script runs. An opcode cache, APC, is planned to be built into an upcoming release of PHP Opcode caching and code optimization can be combined for best efficiency, as the modifications do not depend on each other (they happen in distinct stages of the compilation).

PHP development began in 1994 when the developer Rasmus Lerdorf wrote a series of Common Gateway Interface (CGI) Perl scripts, which he used to maintain his personal homepage. The tools performed tasks such as displaying his résumé and recording his web traffic. He rewrote these scripts in C for performance reasons, extending them to add the ability to work with web forms and to communicate with databases and called this implementation "Personal Home Page/Forms Interpreter" or PHP/FI. PHP/FI could be used to build simple, dynamic web applications.. This release already had the basic functionality that PHP has today. This included Perl-like variables, form handling, and the ability to embed HTML. The syntax was similar to Perl but was more limited and simpler, although less consistent. A development team began to form and, after months of work and beta testing, officially released PHP/FI 2 in November 1997.

Zeev Suraski and Andi Gutmans rewrote the parser in 1997 and formed the base of PHP 3, changing the language's name to therecursive acronym *PHP: Hypertext Preprocessor*.Afterward, public testing of PHP 3 began, and the official launch came in June 1998. Suraski and Gutmans then started a new rewrite of PHP's core, producing the Zend Engine in 1999.[12] They also founded Zend Technologies in Ramat Gan, Israel.

On May 22, 2000, PHP 4, powered by the Zend Engine 1.0, was released.As of August 2008 this branch is up to version 4.4.9. PHP 4 is no longer under development nor will any security updates be released. On July 13, 2004, PHP 5 was released, powered by the new Zend Engine II. PHP 5 included new features such as improved support for object-oriented programming, the PHP Data Objects (PDO) extension (which defines a lightweight and consistent interface for accessing databases), and numerous performance enhancements. In 2008 PHP 5 became the only stable version under development. Late static binding had been missing from PHP and was added in version 5.3.

A new major version has been under development alongside PHP 5 for several years. This version was originally planned to be released as PHP 6 as a result of its significant changes, which included plans for full Unicode support. However, Unicode support took developers much longer to implement than originally thought, and the decision was made in March 2010 to move the project to a branch, with features still under development moved to trunk.

Changes in the new code include the removal of register_globals, magic quotes, and safe mode. The reason for the removals was that register_globals had opened security holes by intentionally allowing runtime data injection, and the use of magic quotes had an unpredictable nature. Instead, to escape characters, magic quotes may be replaced with the addslashes() function, or more appropriately an escape mechanism specific to the database vendor itself like mysql_real_escape_string() for MySQL. Functions that will be removed in future versions and have been deprecated in PHP 5.3 will produce a warning if used.

Many high-profile open-source projects ceased to support PHP 4 in new code as of February 5, 2008, because of the GoPHP5 initiative, provided by a consortium of PHP developers promoting the transition from PHP 4 to PHP 5. PHP interpreters are available on both 32-bit and 64-bit operating systems, but on Microsoft Windows the only official distribution is a 32-bit implementation, requiring Windows 32-bit compatibility mode while using Internet Information Services (IIS) on a 64-bit Windows platform. Experimental 64-bit versions of PHP 5.3.0 were briefly available for Microsoft Windows, but have since been removed.

Perl :

Perl is a high-level, general-purpose, interpreted, dynamic programming language. Perl borrows features from other programming languages including C, shell scripting (sh), AWK, and sed. The language provides powerful text processing facilities without the arbitrary data length limits of many contemporary Unix tools, facilitating easy manipulation of text files. Perl gained widespread popularity in the late 1990s as a CGI scripting language, in part due to its parsing abilities. In addition to CGI, Perl is used for graphics programming, system administration, network programming,finance, bioinformatics, and other applications. The language is intended to be practical (easy to use, efficient, complete) rather than beautiful (tiny, elegant, minimal).

Its major features include support for multiple programming paradigms (procedural, object-oriented, and functional styles), reference counting memory management (without a cycle-detecting garbage collector), built-in support for text processing, and a large collection of third-party modules.

Perl has many and varied applications, compounded by the availability of many standard and third-party modules. Perl code can be made portable across Windows and Unix; such code is often used by suppliers of software (both COTS and bespoke) to simplify packaging and maintenance of software build- and deployment-scripts.

Perl was originally named "Pearl". Wall wanted to give the language a short name with positive connotations; he claims that he considered (and rejected) every three- and four-letter word in the dictionary. He also considered naming it after his wife Gloria. Wall discovered the existing PEARL programming language before Perl's official release and changed the spelling of the name. When referring to the language, the name is normally capitalized (*Perl*) as a proper noun. When referring to the interpreter program itself, the name is often uncapitalized (*perl*) because most Unix-like file systems are case-sensitive. Before the release of the first edition of *Programming Perl*, it was common to refer to the language as *perl*; Randal L. Schwartz, however, capitalized the language's name in the book to make it stand out better when typeset. This case distinction was subsequently documented as canonical.

There is some contention about the all-caps spelling "PERL", which the documentation declares incorrectand which some core community members consider a sign of outsiders. The name is occasionally backronymed as *Practical Extraction and Report Language*, which appears at the top of the documentation and in some printed literature. Several backronyms have been suggested as equally canonical, including Wall's own humorous *Pathologically Eclectic Rubbish Lister*Indeed, Wall claims that the name was intended to inspire many different expansions Wall began work on Perl in 1987, while working as a programmer at Unisys, and released version 1.0 to the comp.sources.miscnewsgroup on December 18, 1987.The language expanded rapidly over the next few years.

Perl 2, released in 1988, featured a better regular expression engine. Perl 3, released in 1989, added support for binary data streams. Originally the only documentation for Perl was a single (increasingly lengthy) man page. In 1991, *Programming Perl*, known to many Perl programmers as the "Camel Book" because of its cover, was published and became the *de facto* reference for the language. At the same time, the Perl version number was bumped to 4, not to mark a major change in the language but to identify the version that was well documented by the book.

Perl 4 went through a series of maintenance releases, culminating in Perl 4.036 in 1993. At that point, Wall abandoned Perl 4 to begin work on Perl 5. Initial design of Perl 5 continued into 1994. The *perl5-porters* mailing list was established in May 1994 to coordinate work on porting Perl 5 to different platforms. It remains the primary forum for development, maintenance, and porting of Perl 5.

Perl 5.000 was released on October 17, 1994. It was a nearly complete rewrite of the interpreter, and it added many new features to the language, including objects, references, lexical (my) variables, and modules. Importantly, modules provided a mechanism for extending the language without modifying the interpreter. This allowed the core interpreter to stabilize, even as it enabled ordinary Perl programmers to add new language features. Perl 5 has been in active development since then.

Perl 5.001 was released on March 13, 1995. Perl 5.002 was released on February 29, 1996 with the new prototypes feature. This allowed module authors to make subroutines that behaved like Perl builtins. Perl 5.003 was released June 25, 1996, as a security release.

One of the most important events in Perl 5 history took place outside of the language proper and was a consequence of its module support. On October 26, 1995, the Comprehensive Perl Archive Network (CPAN) was established as a repository for Perl modules and Perl itself; as of April 2012, it carries over 24,500 modules by more than 9,500 authors.

Perl 5.004 was released on May 15, 1997, and included among other things the UNIVERSAL package, giving Perl a base object to which all classes were automatically derived and the ability to require versions of modules. Another significant development was the inclusion of the CGI.pm module, which contributed to Perl's popularity as a CGI scripting language.

Perl also now supported running under Microsoft Windows and several other operating systems.

Perl 5.005 was released on July 22, 1998. This release included several enhancements to the regex engine, new hooks into the backend through the B::* modules, the qr// regex quote operator, a large selection of other new core modules, and added support for several more operating systems, including BeOS Perl 5.6 was released on March 22, 2000. Major changes included 64-bit support, Unicode string representation, large file support (i.e. files over 2 GiB) and the "our" keyword. When developing Perl 5.6, the decision was made to switch the versioning scheme to one more similar to other open source projects; after 5.005_63, the next version became 5.5.640, with plans for development versions to have odd numbers and stable versions to have even numbers.

In 2000, Wall put forth a call for suggestions for a new version of Perl from the community. The process resulted in 361 RFC (request for comments) documents which were to be used in guiding development of Perl 6. In 2001, work began on the apocalypses for Perl 6, a series of documents meant to summarize the change requests and present the design of the next generation of Perl. They were presented as a digest of the RFCs, rather than a formal document. At this point, Perl 6 existed only as a description of a language.

Perl 5.8 was first released on July 18, 2002, and had nearly yearly updates since then. The latest version of Perl 5.8 is 5.8.9, released December 14, 2008. Perl 5.8 improved unicode support, added a new IO implementation, added a new thread implementation, improved numeric accuracy, and added several new modules.

In 2004, work began on the Synopses – originally documents that summarized the Apocalypses, but which became the specification for the Perl 6 language. In February 2005, Audrey Tang began work on Pugs, a Perl 6 interpreter written in Haskell. This was the first concerted effort towards making Perl 6 a reality. This effort stalled in 2006.

On December 18, 2007, the 20th anniversary of Perl 1.0, Perl 5.10.0 was released. Perl 5.10.0 included notable new features, which brought it closer to Perl 6. These included a switch statement (called "given"/"when"), regular expressions updates, and the smart match operator, "~~". Around this same time, development began in earnest on another implementation of Perl 6 known asRakudo Perl, developed in tandem with the Parrot virtual machine. As of November 2009, Rakudo Perl has had regular monthly releases and now is the most complete implementation of Perl 6.

A major change in the development process of Perl 5 occurred with Perl 5.11; the development community has switched to a monthly release cycle of development releases, with a yearly schedule of stable releases. By that plan, bugfix point releases will follow the stable releases every three months.

On April 12, 2010, Perl 5.12.0 was released. Notable core enhancements include new package NAME VERSION syntax, the Yada Yada operator (intended to mark placeholder code that is not yet implemented), implicit strictures, full Y2038 compliance, regex conversion overloading, DTrace support, and Unicode 5.2. On January 21, 2011, Perl 5.12.3 was released; it contains updated modules and some documentation changes. Version 5.12.4 was released on June 20, 2011. The latest version of that branch, 5.12.5, was released on November 10, 2012.

On May 14, 2011, Perl 5.14 was released. The latest version of that branch, 5.14.3, was released on October 12, 2012. On May 20, 2012, Perl 5.16 was released. Notable new features include the ability to specify a given version of perl that one wishes to emulate, allowing users to upgrade their version of perl, but still run old scripts that would normally be incompatible. Perl 5.16 also updates the core to support Unicode 6.1.

Some observers credit the release of Perl 5.10 with the start of the Modern Perl movement In particular, this phrase describes a style of development which embraces the use of the CPAN, takes advantage of recent developments in the language, and is rigorous about creating high quality code. While the book

"Modern Perl" may be the most visible standard-bearer of this idea, other groups such as the Enlightened Perl Organization have taken up the cause. The overall structure of Perl derives broadly from C. Perl is procedural in nature, with variables, expressions, assignment statements,brace-delimited blocks, control structures, and subroutines.

Perl also takes features from shell programming. All variables are marked with leading sigils, which unambiguously identify the data type (for example, scalar, array, hash) of the variable in context. Importantly, sigils allow variables to be interpolated directly intostrings. Perl has many built-in functions that provide tools often used in shell programming (although many of these tools are implemented by programs external to the shell) such as sorting, and calling on operating system facilities.

Perl takes lists from Lisp, hashes ("associative arrays") from AWK, and regular expressions from sed. These simplify and facilitate many parsing, text-handling, and data-management tasks. Also shared with Lisp are the implicit return of the last value in a block, and the fact that all statements have a value, and thus are also expressions and can be used in larger expressions themselves.

Perl 5 added features that support complex data structures, first-class functions (that is, closures as values), and an object-oriented programming model. These include references, packages, class-based method dispatch, and lexically scoped variables, along withcompiler directives (for example, the strict pragma). A major additional feature introduced with Perl 5 was the ability to package code as reusable modules. Wall later stated that "The whole intent of Perl 5's module system was to encourage the growth of Perl culture rather than the Perl core."

All versions of Perl do automatic data-typing and automatic memory management. The interpreter knows the type and storagerequirements of every data object in the program; it allocates and frees storage for them as necessary using reference counting (so it cannot deallocate circular data structures without manual intervention). Legal type conversions — for example, conversions from number to string — are done automatically at run time; illegal type conversions are fatal errors.

The design of Perl can be understood as a response to three broad trends in the computer industry: falling hardware costs, rising labor costs, and improvements in compiler technology. Many earlier computer languages, such as Fortran and C, aimed to make efficient use of expensive computer hardware. In contrast, Perl is designed to make efficient use of expensive computer-programmers.

Perl has many features that ease the task of the programmer at the expense of greater CPU and memory requirements. These include automatic memory management; dynamic typing; strings, lists, and hashes; regular expressions; introspection; and an eval()function. Perl follows the theory of "no built-in limits", an idea similar to the Zero One Infinity rule. Wall was trained as a linguist, and the design of Perl is very much informed by linguistic principles. Examples include Huffman coding(common constructions should be short), good end-weighting (the important information should come first), and a large collection of language primitives. Perl favors language constructs that are concise and natural for humans to write, even where they complicate the Perl interpreter.

Perl's syntax reflects the idea that "things that are different should look different." For example, scalars, arrays, and hashes have different leading sigils. Array indices and hash keys use different kinds of braces. Strings and regular expressions have different standard delimiters. This approach can be contrasted with languages such as Lisp, where the same S-expression construct and basic syntax are used for many different purposes.

Perl does not enforce any particular programming paradigm (procedural, object-oriented, functional, or others) or even require the programmer to choose among them. There is a broad practical bent to both the Perl language and the community and culture that surround it. The preface to *Programming Perl* begins: "Perl is a language for getting your job done." One consequence of this is that Perl is not a tidy language. It includes many features, tolerates exceptions to its rules, and employs heuristics to resolve syntactical ambiguities. Because of the forgiving nature of the compiler, bugs can sometimes be hard to find. Perl's function documentation remarks on the variant behavior of built-in functions in list and scalar contexts that "In general, they do what you want, unless you want consistency."

No written specification or standard for the Perl language exists for Perl versions through Perl 5, and there are no plans to create one for the current version of Perl. There has been only one implementation of the interpreter, and the language has evolved along with it. That interpreter, together with its functional tests, stands as a *de facto* specification of the language. Perl 6, however, started with a specification, and several projects aim to implement some or all of the specification.

Perl has many and varied applications, compounded by the availability of many standard and third-party modules.

Perl has chiefly been used to write CGI scripts: large projects written in Perl include cPanel, Slash, Bugzilla, RT, TWiki, and Movable Type; high-traffic websites that use Perl include Amazon.com, bbc.co.uk, Priceline.com, Craigslist,[50] IMDb,[51]LiveJournal, Slashdot and Ticket master. It is also an optional component of the popular LAMP technology stack for web development, in lieu of PHP or Python.

Perl is often used as a glue language, tying together systems and interfaces that were not specifically designed to interoperate, and for "data munging". that is, converting or processing large amounts of data for tasks such as creating reports. In fact, these strengths are intimately linked. The combination makes Perl a popular all-purpose language for system administrators, particularly because short programs, often called "one-liner programs", can be entered and run on a single command line.

Perl code can be made portable across Windows and Unix; such code is often used by suppliers of software (both COTS and bespoke) to simplify packaging and maintenance of software build- and deployment-scripts.

Graphical user interfaces (GUIs) may be developed using Perl. For example, Perl/Tk and WxPerl are commonly used to enable user interaction with Perl scripts. Such interaction may be synchronous or asynchronous, using callbacks to update the GUI. Perl is implemented as a core interpreter, written in C, together with a large collection of modules, written in Perl and C. As of 2010, the stable version (5.12.3) is 14.2 MB when packaged in a tar file and gzip compressed. The interpreter is 150,000 lines of C code and compiles to a 1 MB executable on typical machine architectures. Alternatively, the interpreter can be compiled to a link library and embedded in other programs. There are nearly 500 modules in the distribution, comprising 200,000 lines of Perl and an additional 350,000 lines of C code. (Much of the C code in the modules consists of character encoding tables.)

The interpreter has an object-oriented architecture. All of the elements of the Perl language—scalars, arrays, hashes, coderefs, file handles—are represented in the interpreter by C structs. Operations on these structs are defined by a large collection of macros,typedefs, and functions; these constitute the Perl C API. The Perl API can be bewildering to the uninitiated, but its entry points follow a consistent naming scheme, which provides guidance to those who use it.

The life of a Perl interpreter divides broadly into a compile phase and a run phase. In Perl, the **phases** are the major stages in the interpreter's life-cycle. Each interpreter goes through each phase only once, and the phases follow in a fixed sequence. Most of what happens in Perl's compile phase is compilation, and most of what happens in Perl's run phase is execution, but there are significant exceptions. Perl makes important use of its capability to execute Perl code during the compile phase. Perl will also delay compilation into the run phase. The terms that indicate the kind of processing that is actually occurring at any moment are **compile time** and **run time**. Perl is in compile time at most points during the compile phase, but compile time may also be entered during the run phase. The compile time for code in a string argument passed to the eval built-in occurs during the run phase. Perl is often in run time during the compile phase and spends most of the run phase in run time. Code in BEGIN blocks executes at run time but in the compile phase.

At compile time, the interpreter parses Perl code into a syntax tree. At run time, it executes the program by walking the tree. Text is parsed only once, and the syntax tree is subject to optimization before it is executed, so that execution is relatively efficient. Compile-time optimizations on the syntax tree include constant folding and context propagation, but peephole optimization is also performed.

Perl has a Turing-complete grammar because parsing can be affected by run-time code executed during the compile phase. It is often said that "Only perl can parse Perl", meaning that only the Perl interpreter (*perl*) can parse the Perl language (*Perl*), but even this is not, in general, true. Because the Perl interpreter can simulate a Turing machine during its compile phase, it would need to decide the Halting Problem in order to complete parsing in every case. It's a long-standing result that the Halting Problem is undecidable, and therefore not even perl can always parse Perl. Perl makes the unusual choice of giving the user access to its full programming power in its own compile phase. The cost in terms of theoretical purity is high, but practical inconvenience seems to be rare.

Other programs that undertake to parse Perl, such as source-code analyzers and auto-indenters, have to contend not only with ambiguous syntactic constructs but also with the undecidability of Perl parsing in the general case. Adam Kennedy's PPI project focused on parsing Perl code as a document (retaining its integrity as a document), instead of parsing Perl as executable code (which not even Perl itself can always do). It was Kennedy who first conjectured that "parsing Perl suffers from the 'Halting Problem'", which was later proved.

Perl is distributed with some 120,000 functional tests. These run as part of the normal build process and extensively exercise the interpreter and its core modules. Perl developers rely on the functional tests to ensure that changes to the interpreter do not introducesoftware bugs; additionally, Perl users who see that the interpreter passes its functional tests on their system can have a high degree of confidence that it is working properly.

Maintenance of the Perl interpreter has become increasingly difficult over the years. The code base has undergone continuous development since 1994. The code has been optimized for performance at the expense of simplicity, clarity, and strong internal interfaces. New features have been added, yet virtually complete backward compatibility with earlier versions is maintained. Major releases of Perl were coordinated by Perl pumpkings, which handled integrating patch submissions and bug fixes, but the language has since changed to a rotating, monthly release cycle. Development discussion takes place via the perl5_porters mailing list. As of Perl 5.11, development efforts have included refactoring certain core modules known as 'dual lifed' modules out of the Perl core to help alleviate some of these problems.

CHAPTER 2 PROBLEM DEFINITION

Problems in the existing system:

There are various aptitude sites who are providing a wide range of online practice aptitude and personality tests. Like IndiaBIX.com. There are sites like The Berger Aptitude for Programming Test which provides aptitude test used in recruitment of computer programmers. These above mentioned sites can not be used in a specified way for our university's students due to the lack of the database updation.

Solutions to these problems:

"Apti Geek" is an online application for conducting aptitude tests on web browser to assess logical reasoning or thinking performance of the students who are appearing for the placements and to check their IQ level for our university.

- The result of the conducted aptitude test will be properly displayed and is stored in the database for the future requirements along with the user's details.
- it also provides the storage and retrieval of result effectively. Tests are typically timed and a typical test might allow 30 minutes for 30 questions and so on.
- Largest database of tests and quizzes are available on the internet. But "Apti geek" will provide the question-answers those are being asked by various companies which are visiting to our campus each and every year. The best trend in this site will be the updation factor.
- It will save time and energy of the students to prepare for a specific company that is going to visit our campus rather than to prepare in a generalized way.

Main Objective of our project is to develop an online application for conducting aptitude tests on web browser to assess logical reasoning or thinking performance of the students who are appearing for the placements and to check their IQ level. This site is being developed at the university level. The result of the conducted aptitude test will be properly displayed and is stored in the database for the future requirements along with the user's details. As there are various sites for conducting online aptitude test but it will be different from others sites as we will try to develop it on the basis of difficulty levels which will be divided in easy medium & tough levels.

Largest database of tests and quizzes are available on the internet. But "Apti geek" will provide the question-answers those are being asked by various companies which are visiting to our campus each and every year. The best trend in this site will be the updation factor.

It will save time and energy of the students to prepare for a specific company that is going to visit our campus rather than to prepare in a generalized way.

The purpose of this application is to provide security measures also that enable the administrator to keep a track of what is going on or what is the status of the students who will be appearing for the placements.

CHAPTER 3 LITERATURE SURVEY

3.1 XAMPP Server

XAMPP is a free and open source cross-platform web server solution stack package, consisting mainly of the Apache HTTP Server, MySQL database, and interpreters for scripts written in the PHP and Perl programming languages. XAMPP requires only one zip, tar, exe file to be downloaded and run, and little or no configuration of the various components that make up the web server is required. XAMPP is regularly updated to incorporate the latest releases of Apache/MySQL/PHP and Perl. It also comes with a number of other modules including OpenSSL and phpMyAdmin.

Officially, XAMPP's designers intended it for use only as a development tool, to allow website designers and programmers to test their work on their own computers without any access to the Internet. To make this as easy as possible, many important security features are disabled by default. In practice, however, XAMPP is sometimes used to actually serve web pages on the World Wide Web· A special tool is provided to password-protect the most important parts of the package.

XAMPP also provides support for creating and manipulating databases in MySQL and SQLite among others.

Once XAMPP is installed, it is possible to treat a localhost like a remote host by connecting using an FTP client. Using a program likeFileZilla has many advantages when installing a content management system (CMS) like Joomla. It is also possible to connect to localhost via FTP with an HTML editor.

The default FTP user is "newuser", the default FTP password is "wampp".

The default MySQL user is "root" while there is no default MySQL password.

3.2 PHP- The Leader

PHP is a powerful server-side scripting language for creating dynamic and interactive websites.

PHP is the widely-used, free, and efficient alternative to competitors such as Microsoft's ASP. PHP is perfectly suited for Web development and can be embedded directly into the HTML code.

The PHP syntax is very similar to Perl and C. PHP is often used together with Apache(web server) on various operating systems. It also supports ISAPI and can be used with Microsoft's IIS on Windows.

A PHP file may contain text, HTML tags and scripts. Scripts in a PHP file are executed on the server.

PHP files are returned to the browser as plain HTML. PHP files have a file extension of ".php", ".php3", or ".phtm".

The code is interpreted by a Web server with a PHP processor module which generates the resulting Web page. It also has evolved to include a command-line interface capability. PHP acts primarily as a filter, taking input from a file or stream containing text and/or PHP instructions and outputting another stream of data; most commonly the output will be HTML. PHP can be deployed on most Web servers, many operating systems and platforms, and can be used with many relational database management systems (RDBMS). It is available free of charge, and the PHP Group provides the complete source code for users to build, customize and extend for their own use. PHP has also attracted the development of many frameworks that provide building blocks and a design structure to promote rapid application development

CHAPTER 4 SYSTEM REQUIREMENT SPECIFICATION

4.1 Overall Description

4.1.1 Product Perspective

- "Apti Geek" is an online application for conducting aptitude tests on web browser to assess logical reasoning or thinking performance of the students who are appearing for the placements and to check their IQ level for our university.
- The result of the conducted aptitude test will be properly displayed and is stored in the database for the future requirements along with the user's details.
- it also provides the storage and retrieval of result effectively. Tests are typically timed and a typical test might allow 30 minutes for 30 questions and so on.

4.1.2 Product Features

The major functionalities of the proposed system are:

- Firstly all the users have to register theirself on the site with their university roll no. or sap id provided by the university.
- After registration user can sign in and start the aptitude test.
- As it is a typically time based application so user has to submit his test on time otherwise he or she will not be able to submit the test if time limit is over.
- At the end of the aptitude test result will be displayed.
- The result along with the user's detail will be stored in our database.

4.1.3 User Classes and Characteristics

The class of users that we aim to serve with this product is diverse and diverged on how they are going to use it.The service to access the relevant aptitude, technical and interview questions from this site in our university.Connection with the companies' web site in order to get the information regarding the changing criteria for recruitments.

4.1.4 Operating Environment

PHP runs on different platforms (Windows, Linux, Unix, etc.). PHP is compatible with almost all servers used today (Apache, IIS, etc.). PHP is FREE to download from the official PHP resource. PHP combined with MySQL are cross-platform (means that you can develop in Windows and serve on a Unix platform)

HARDWARE REQUIREMENTS:

Pentium 4 Processor or Above
20 GB of Hard Disk Space
Minimum 128 MB of RAM

SOFTWARE REQUIREMENTS:

XAMP software bundle (Windows, Apache server, MySQL database and PHP)
Text Pad
MYSQL as Backend
PHP as Frontend
Web browser

4.1.5 Design and Implementation Constraints

For Designing this Application we need to perform the following actions on our system:

- Installation of an Apache server on a Windows or Linux machine.
- Installation of PHP on a Windows or Linux machine.
- Installation of MySQL on a Windows or Linux machine.

If your server supports PHP - you don't need to do anything! You do not need to compile anything or install any extra tools - just create some .php files in your web directory - and the server will parse them for you. Most web hosts offer PHP support.
You cannot view the PHP source code by selecting "View source" in the browser - you will only see the output from the PHP file, which is plain HTML. This is because the scripts are executed on the server before the result is sent back to the browser.

4.1.6 User Documentation

The list the user documentation components proposed to be available with the software are

- AdministratorPage
- HomePage
- SignUp Page
- LoginPage
- AreaOfSelection
- AttemptQuizNow
- ResultSheet
- FeedbackForm

4.1.7 Assumptions and Dependencies

Some of the factors that can impact our software directly or indirectly are:

The Apache HTTP Server is a powerful and flexible HTTP/1.1 compliant web server. Originally designed as a replacement for the NCSA HTTP Server, it has grown to be the most popular web server on the Internet. As a project of the Apache Software Foundation, the developers aim to collaboratively develop and maintain a robust, commercial-grade, standards-based server with freely available source code.

MySQL, also commonly "My Sequel" is a relational database management system (RDBMS) that runs as a server providing multi-user access to a number of databases. It is named after developer Michael Widenius' daughter, My. The SQL phrase stands for Structured Query Language.
The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements. MySQL was owned and sponsored by a single for-profit firm, the Swedish company MySQL AB, now owned by Oracle Corporation.
Free-software-open source projects that require a full-featured database management system often use MySQL. For commercial use, several paid editions are available, and offer additional functionality.

Applications which use MySQL databases include: Joomla, WordPress, MyBB, phpBB, Drupal and other software built on the LAMP software stack. MySQL is also used in many high-profile, large-scale World Wide Web products, including Wikipedia, Google (though not for searches) and Facebook.

4.2 System Features

Description and Priority

As the software is developed in XAMPP environment so it can be run on any operationg system or it is platform independent because of the use of php (hypertext pre-processor) which is an open source softaware and free to use. PHP supports many databases (MySQL, Informix, Oracle, Sybase, Solid, PostgreSQL, Generic ODBC, etc.)

Functional Requirements:

1. Firstly all the users have to register their self on the site with their university roll no. or sap id provided by the university.
2. After registration user can sign in and start the aptitude test.
3. As it is a typically time based application so user has to submit his test on time otherwise he or she will not be able to submit the test if time limit is over.
4. At the end of the aptitude test result will be displayed.
5. The result along with the user's detail will be stored in our database.

4.3. External Interface Requirements

4.3.1 User Interfaces

For Designing this Application we need to perform the following actions on our system:

- Installation of an XAMPP server on a Windows or Linux machine. Which will automatically include :
- Installation of an Apache server on a Windows or Linux machine.
- Installation of PHP on a Windows or Linux machine.
- Installation of MySQL on a Windows or Linux machine.

4.3.2 Software Interfaces

The software that is developed in php follows the XAMPP environment. The application is platform independent, can be run on any operating system and also there are a lot of libraries on XAMPP APIs available free that enables us to indirectly customise the environment.

4.3.3 Communications Interfaces

Xampp is the main communication channel for our system, and we propose to introduce a standard website so that the user can appear for the aptitude test.

We also look at using the php/mysql that enables the user to login to the site and it enables the user to appear for the aptitude test to check their mental ability in a specific amount of time. As it is a typically time based application so user has to submit his test on time otherwise he or she will not be able to submit the test if time limit is over. At the end of the aptitude test result will be displayed.

The result along with the user's detail will be stored in our database.

4.4 Other Non-functional Requirements

4.4.1 Performance Requirements

While looking at the performance aspect of the software, we have prioritised the requirements and have given the response time in respect to these priorities. For example, critical functionalities like shutdown, standby, hibernate or logoff will have more time after the optional notification to enable the user to save or put an intermediate save point before the action is performed. Other functions like lock, getfile, screenshot that requires immediate response is executed with minimum delay.

4.4.2 Security Requirements

Since we use the college sapid as username and password for authentication, this would require to be encrypted and stored at the client side, so that the does not require to feed in the details each of the time he /she logs in. We also propose to make use of an auto-generated PIN code which can be used later on for any modifications in settings without actually logging in with the college access medium every time. Thus if one feels that security is compromised; we can reset it by merely re-generating the Pin code and will not require to change the configured id or its settings.

CHAPTER 5 SYSTEM DESIGN

System analysis refers to the process of examining a business situation with the intent of the improving it through better procedure and methods. It is the process of gathering and interpreting facts, diagnosing problems.

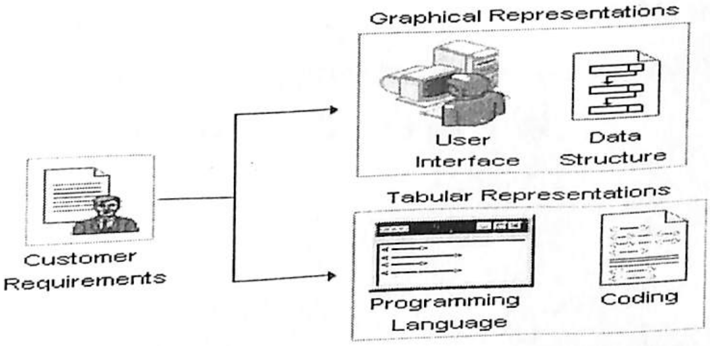
The scope of the system design is guided by the framework for the application developed during analysis.

System design involves first Logical design and then Physical design construction of the system. When analysts formulate logical design; they describe its future, the output, the input, the files database and procedures needed.

The statement of these features is termed as design specification. Physical construction, the activity followed logical design, procedure programs software, files and a working system design specification, instruct programmer about what the system do.

System Design involves the following processes:-

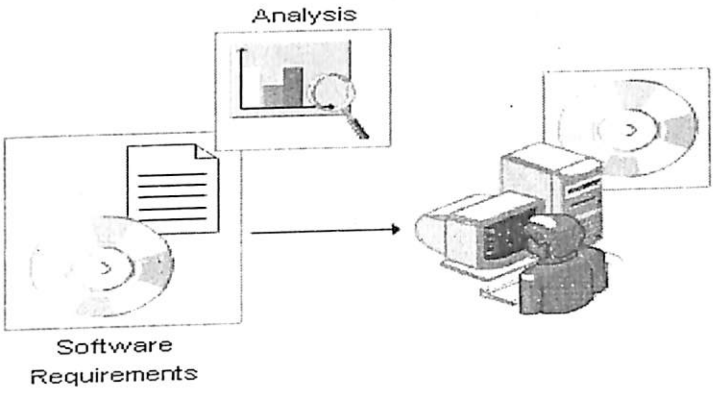
- Design of Input.
- Design of Output.
- Design of database.



Steps Followed in the Development Phase

- Software design
- Coding
- Software testing

The development phase involves three steps listed in the graphic. The software design step of the development phase converts the requirements specified by the customer into graphical or tabular representations. These representations describe the user interface, architecture, programming language, and data structure used to develop software.



A key feature in the development of any software is analysis of the requirements that must be satisfied by software. A thorough understanding of these requirements is essential for the successful development and implementation of software.

Steps to perform the System Requirement Analysis:-

(1) Identify the customer's needs

(2) Evaluate the feasibility of the project

(3) Evaluate the alternative approaches to the development the software

(4) Create the model of the software

(5) Create the system specification of the develop software

(6) Review the system software

System requirements analysis is a detailed study of the elements of a system to collect the data for developing or refining Software. System requirements analysis is done by a system analyst. The steps to perform system requirements analysis are displayed.

CHAPTER 6 DETAILED DESIGN

The System is mainly having two modules:

- Communication Module
- Configuration Module

6.1 Communication Module

This module is the core of the AptiGeek application. It handles how the communication is maintained between the College, students and the application.

- "Apti Geek" is an online application for conducting aptitude tests on web browser to assess logical reasoning or thinking performance of the students who are appearing for the placements and to check their IQ level for our university.
- The result of the conducted aptitude test will be properly displayed and is stored in the database for the future requirements along with the user's details.
- it also provides the storage and retrieval of result effectively. Tests are typically timed and a typical test might allow 30 minutes for 30 questions and so on.

1-Level DFD:-

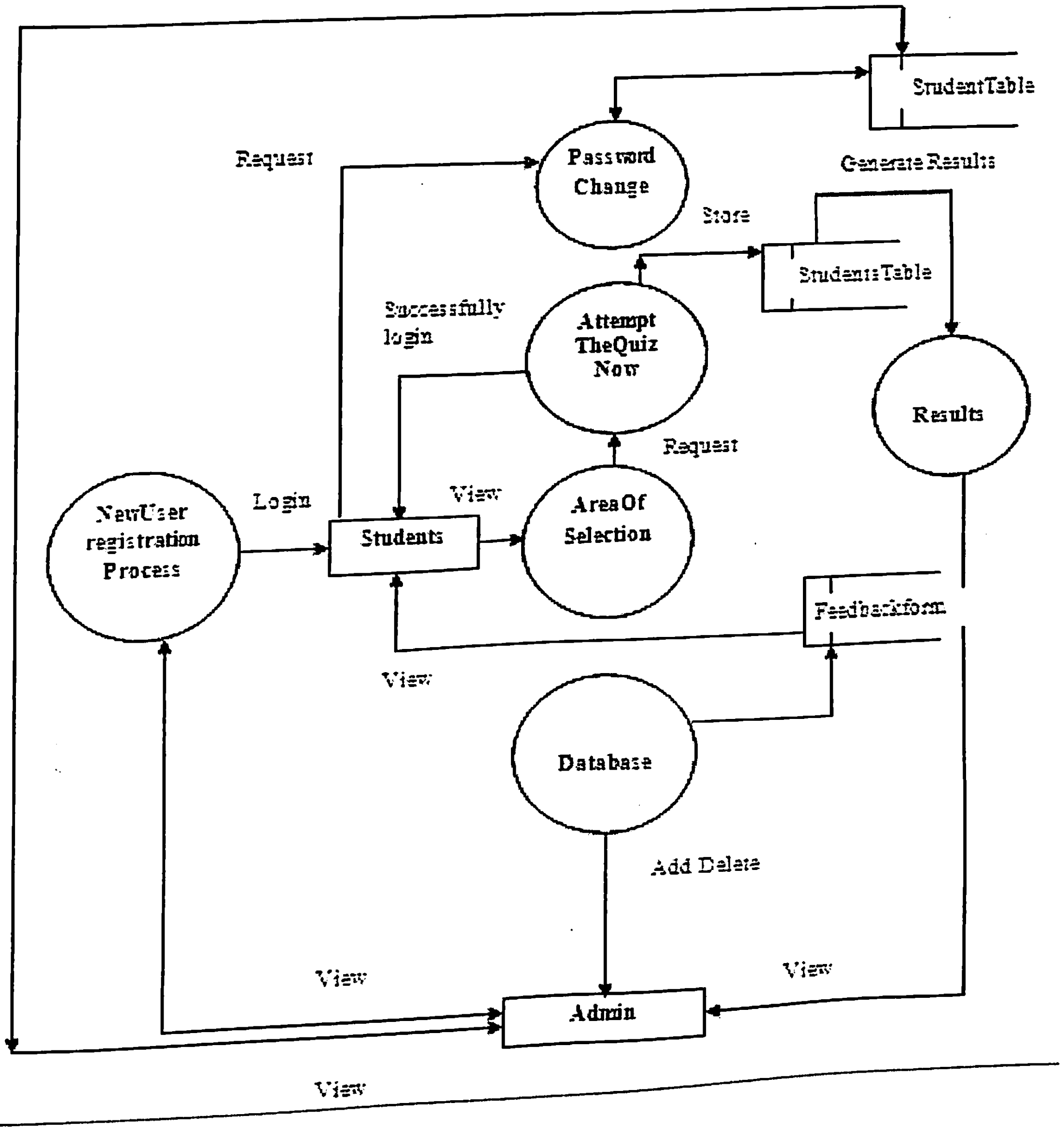


Fig: 6.1 - DFD for Communication Module

6.2 Configuration Module

This module focuses on the Authentication, Security and Customization aspects of the System.

6.2.1 *Authentication*

During Initial configuration, the students Sap id and its password are received from the college, based on which the commands will be fetched and processed by the Communication Module. Optionally, they can configure an Administrator account also, which will be allowed to perform a limited set of, Admin level of services. Once this is completed, the validity of the User Id (Sap id) - pwd combination is checked and if successful, then the user can successfully login to the site. All these details will be encrypted and stored in the Database itself.

On every update of any of these details, the same process will be repeated.

6.2.2 *Customization*

Even though the need of customization is less in such a system, we have given options to decide where to save the results file, and the generated feedbackform files. Also, the user can decide whether or not to show the Notifications each time a command is processed on the system by attempting the quiz.

The database file will be updated for each and every command that is processed by the system. It also has the date and time on which the command was processed by the system. Also, for commands like newUserSignUp, or loginPage, or file list, the necessary actions will be created by the application and will be stored in the database given in the configuration part.

6.2.3 *Security*

On re-entry into the Application, the user will be asked to login with the password that is stored in the database for authentication. The entered password will be checked against the value stored in the system and if found correct, access will be allowed. In case, if the user has forgotten the password, he / she can re-generate it at the login screen. A new password will be generated and will be sent to the registered phone no.

CHAPTER 7 IMPLEMENTATION

The implementation of the AptiGeek is done in a modular fashion. This increases the reusability of the code as well the makes the application structured.

7.1 Implemented Code

HomePage:

```
<html>
<head><title>HomePage</title></head>
<body background="img src="images\light.jpeg"">
<form>
<center>
<h1><font face="monotype corsiva" color="maroon" size="15"><u>Apti Geek</u></font></h1>
</center>
<center>
<h1><font color="Purple" size="6">WELCOME TO THE SITE</font></h1>
</center>
<hr color="red">
<TABLE cellSpacing=0 cellPadding=0 width="100%" border=0 bgcolor="">
 <TBODY align=center>
<TR>
   <TD><!--mstheme--><FONT face=Arial>
     <P><B><FONT size=2>
<A href="home.php" target="clientarea_frame">Home</A>&nbsp;&nbsp;</FONT>
<FONT    color="#993333"    size="2">    &nbsp;&nbsp;|&nbsp;&nbsp;    </FONT><FONT    size="2"
color="maroon">

<A href="AreaOfSelection.php" target="clientarea_frame">AreaOfSelection</A></FONT>
<FONT    color="#993333"    size="2">&nbsp;&nbsp;    |&nbsp;&nbsp;    </FONT><FONT    size="2"
color="maroon">
```

```
<A href="Attempt.php" target="clientarea_frame">AttemptQuizNow</A>&nbsp;&nbsp;</FONT>
<FONT    color="#993333"    size="2">&nbsp;&nbsp;    |&nbsp;&nbsp;    </FONT><FONT    size="2"
color="maroon">
<A href="Result.php" target="clientarea_frame">ResultSheet</A></FONT>
<FONT    color="#993333"    size="2">&nbsp;&nbsp;    |&nbsp;&nbsp;    </FONT><FONT    size="2"
color="maroon">
<A href="feedback.php" target="clientarea_frame">FeedbackForm</A></FONT>
<FONT    color="#993333"    size="2">&nbsp;&nbsp;    |&nbsp;&nbsp;    </FONT><FONT    size="2"
color="maroon">
<h1>
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbs
p;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbs
p;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbs
p;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbs
p;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&n
bsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;


</FONT></TD></TR></TBODY></TABLE>
<hr color="red">
<right>
<div>
<TABLE width="50%" border="1" bgcolor="skyblue" style="border:1px solid red;">
        <TBODY>
        <TR>
          <TD width="20%"></TD>
          <TD id=content_line width="60%"></TD>
          <TD width="20%"></TD></TR>

        <TR id=content_between></TR>
        <TR>
        <TD colSpan=3>
         <P><STRONG><SPAN
         style="FONT-SIZE: large; FONT-FAMILY: Times New Roman"><font color="blue">What is
         Apti Geek?</font></SPAN></STRONG></P>
         <P><BR><marquee behavior="scroll" direction="up" scrollamount="3"><SPAN
         style="FONT-SIZE: large; FONT-FAMILY: Times New Roman"><font color="brown">An
"APTI GEEK" is an application which is capable of conducting online aptitude tests on the internet. This
site is designed to assess logical reasoning or thinking performance. Aptitude Tests will be divided into
3 sections. First section consists of Quantum i.e. purely mathematical questions, Second section deals
with verbal reasoning & third section consists of questions based on data interpretations that would be
the toughest one. They consist of multiple choice questions. These are typically timed and a typical test
might allow 30 minutes for 30 questions and so on. The test will be conducted on the basis of difficulty
level. i.e. the user will enter in the first level and then if he clears it then he will reach to the next level
according to more difficulty. The result of the aptitude test will be displayed and it will be stored in the
database for any future use. This project of conducting Online Aptitude Test has much scope in today.
Today almost everything is performed electronically hence "Apti Geek" provides a good platform for
aptitude assessment without paper work. Furthermore it also provides the storage and retrieval of result
effectively. Hence paperwork is reduced and manual checking is also not
required.</SPAN></marquee></P></TD></TR></TBODY></TABLE></div>



</right><br/><br/>
<body bgcolor="pink"><center>
<b>Copyright @AptiGeek.com, All Right Reserved.</center>



</form>
</body>
</html>
```

NewUserSignUpPage

```
<html>
<head><title>SignUp Page!!!!!</title>
</head>
<body background="C:/xampp/htdocs/Major 1/images/ist2_7869708-green-blurry-lights.jpg">

<marquee bgcolor="purple" behavior='alternate' width="100%" truespeed="true" ><font color="white"
size="5" ><b>Welcome To SignIn Form</font></b></marquee>
<P><font color="white" size="4">We'll get you to set up on<font color="maroon" size="5"> APTI
GEEK!!!</font>
<style type="text/css">



body {
        color:#FFFFFF;
 background:#CCCCFF;
}
</style>
&nbsp;&nbsp;&nbsp;&nbsp<br>
<center>
<form method="post" action="form2.php">
<br>
<FIELDSET>
<LEGEND><b><font color="darkblue"</font>Fill The Form Carefully...!!!</b></LEGEND>
<font size="4">SAP ID:
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbs
p;<input type="text" name="sap_id"><br><br>
First Name:&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<input type="text"
name="fname"><br><br>
Last Name: &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<input type="text"
name="lname"><br><br>
```

```
Password:          &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<input
type="password" name="pswd"><br><br>
Confirm Password: <input type="password" name="pswd"><br><br>
Enrollment Number: &nbsp;<input type="text" name="enroll"><br><br>
Qualification: &nbsp;<input type="radio" name="qualification" value="B.Tech"> B.Tech <br>
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbs
p;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
<input type="radio" name="qualification" value="M.Tech"> M.Tech <br>
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbs
p;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
<input type="radio" name="qualification" value="M.B.A"> M.B.A. <br>
Year;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<input
type="text" name="year"><br><br>
</FIELDSET>
<FIELDSET>
<LEGEND><b><font color="darkblue"</font>Aggrement...</b></LEGEND>
<center><P>I have read and agree to the <A title="Terms & conditions of Service" href="">APTI GEEK!
Terms of Service</A>
<center><LABEL>Do you agree?</LABEL>
<LABEL><INPUT type="checkbox"></LABEL></center>
</FIELDSET><br>
<center><INPUT          type="submit"          value="Create          My
Account">&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
<INPUT type="submit" name="cancel" value="Cancel"></center><br>
</form>
</body>
</html>
```

LoginPage

```
<html>
<head>


<center>
</head>
<body>
<center>
<h1><font face="Monotype Corsiva" color="green" size="20"><u>Apti Geek</u></font></h1>
</center>
<style type="text/css">


body {
        color:#FFFFCC;
        background:#FFFFCC;
}
</style>
<center><form method="post" action="homepage.php">
<fieldset>
<legend><b><font color="red">Login Here</legend>

<font  color="red"  size="3">  SAP  ID  &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;</font>
<input type="text" name="sap_id"><br><br>
 <font color="red" size="3"> PASSWORD </font> <input type="password" name="pswd"><br><br>
<input type="submit" value="SIGNIN"><br><br>
</form>
<a  href="SignUp.php"><img  src="images\new_shine.gif"  align=center  border="0"><b>New  Member
Sign-Up</a>
 </form>
 </body>
 </html>
```

DatabaseConnection:

```
<?php
session_start();
$sap_id=$_POST['sap_id'];
$password=$_POST['pswd'];
mysql_connect("localhost","root","");
mysql_select_db("major1");
$str="select * from newuser where sap_id='$sap_id' and password='$pswd'";
if(!$result=mysql_query($str))
{
        die("Database error");
}
else

{
        $num=mysql_num_rows($result);
        if($num==1)
        {                                                          //session registered here
        $_SESSION['sap_id']=$sap_id;
                ?>
                <script>window.location="HomePage.php?sap_id=<?php echo $sap_id;?>";</script>
        <?php
        }
        else
        {
        ?>
                <script>window.location="home.php?msg=0";</script>
 <?php
        }}?>
```

CHAPTER 8

8.1 Testing Approach

The testing approach adopted for this project involves four major phases

- Command testing
- Unit testing
- Integration testing
- Beta testing

8.1.1 Command Testing

In the command testing phase, we identified and ensured that the various system commands that we have built together works together in the building environment and can be used or modified to perform the required function.

8.1.2 Unit Testing

The system was built and the software was tested for the compatibility of the various commands and the approach we have adopted to build the system. It involved testing of the system without actually integrating a communication medium and we looked at the working of the software as a stand alone system. The testing of those commands which required network / internet was kept aside for testing during the integration testing phase.

➢ **UNIT TEST PROCEDURES:**

- Unit testing is normally considered as an adjunct to the coding step.
- After source code is developed, reviewed and verified, test cases were established.

- Functional behavioral were tested for each module. In the present project the following
- Unit test were conducted.
- Validation of registered users' data.
- Attemption of quiz by the students.
- Retrieval of data from the database.
- Display of results in the front end depending on the option selected by the user.
- Display of Results of the test
- Allowing the students to fill the feedback form

8.1.3 Integration Testing

We now integrated the authentication feature and we continued the testing process to see how the system would react when we connect it to the network. The system was full up and functional and we tested the integration of the system along with the Gmail server. We also ensured how it would react to a load of multiple emails in the configured email-id.

8.1.4 Beta Testing

The software was installed in multiple systems by the developers and people associated to the project and they have used and analysed how the software is behaving on a longer run. Due to the time constraint, we have done this phase for a short duration of one week.

8.2 Findings

8.2.1 Command testing

No major issues found

8.2.2 Unit testing

No major issues found

8.2.3 Integration testing

Issue 1: Identified that using atom feed provided by the Gmail, we cannot mark the read ones. Hence we could not identify the processed commands

Solution: Replaced the atom feed with the POP3 server facility provided by Gmail that enables the required action

Issue 2: Even tough the application was working fine, with the POP3 server, the read mails were not marked accordingly in the inbo8.

Solution: Identified that this was due to an incorrect setting and Gmail was maintaining its version of the mail and thus it was not getting reflected. Corrected the Gmail settings accordingly

8.2.4 Beta testing

Issue 1: When notifications are turned off, the message command was not displaying the message to the client system.

Solution: It was due to notifications not being enabled in the settings tab. Hence we have now forced enabled before popping out the message and then reset it back to its original state.

CHAPTER 9 CONCLUSION

The anywhere, anytime system that we envisaged, designed and implemented can be considered as a revolution in the students preparation for the aptitude tests. This system is primarily aimed at students who are going to appear for the aptitude tests in the upcoming companies.

Largest database of tests and quizzes are available on the internet. But "Apti geek" will provide the question-answers those are being asked by various companies which are visiting to our campus each and every year. The best trend in this site will be the updation factor.

It will save time and energy of the students to prepare for a specific company that is going to visit our campus rather than to prepare in a generalized way.

CHAPTER 10 FUTURE ENHANCEMENTS

Even though we have put our absolute effort in including all the necessary features making use of the latest available technological supports, there still seems scope for improvement. But most of these need either new infrastructure or a new framework to have the full usability.

First of these would be to have an mail module also as part of this system, so that it would be more user friendly. The system should be able to receive and process the control commands when it is received from the registered mail id also. But to set up this, we need to set up a message gateway or should make use of the paid third party gateways. Either of these would be a costly affair. Even then, we should make sure the reliability and availability of the gateway.

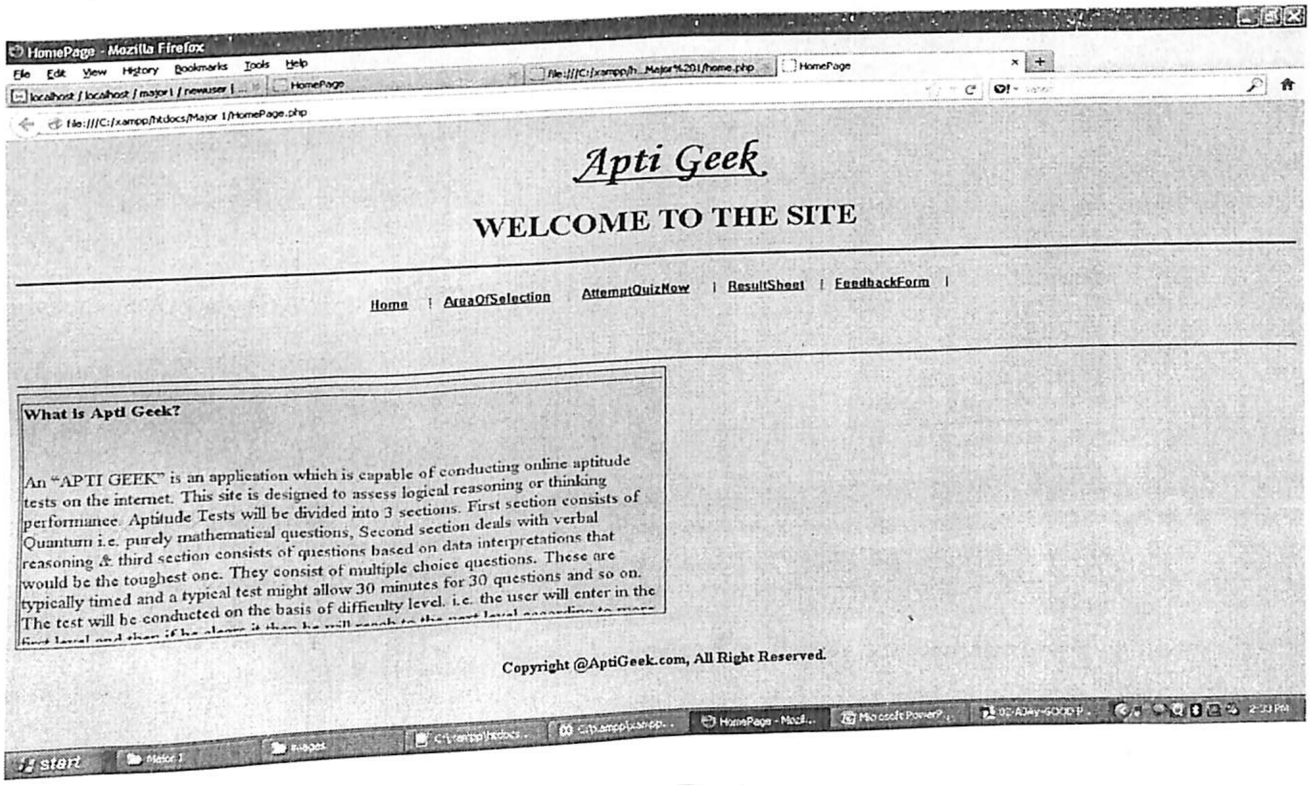
The service to access the relevant aptitude, technical and interview questions from this site in our university. Connection with the companies' web site in order to get the information regarding the changing criteria for recruitments.

Apart from these, we can also have modifications to the application to have the configuration details to be stored in a safer location than in the flat files. Even though the personal information is encrypted, if the data is intentionally or unknowingly deleted, we will have to reconfigure the system again to have full control.

BIBLIOGRAPHY

- www.w3schools.com
- www.wikipedia.org
- http://www.tizag.com/mysqlTutorial/
- http://php.about.com/od/phpwithmysql/ss/mysql_php.htm
- http://www.dwpractice.com/mysqlTutorial/
- http://www.codefire.com/
- Google Search

APPENDIX – SAMPLE SCREEN PRINTS

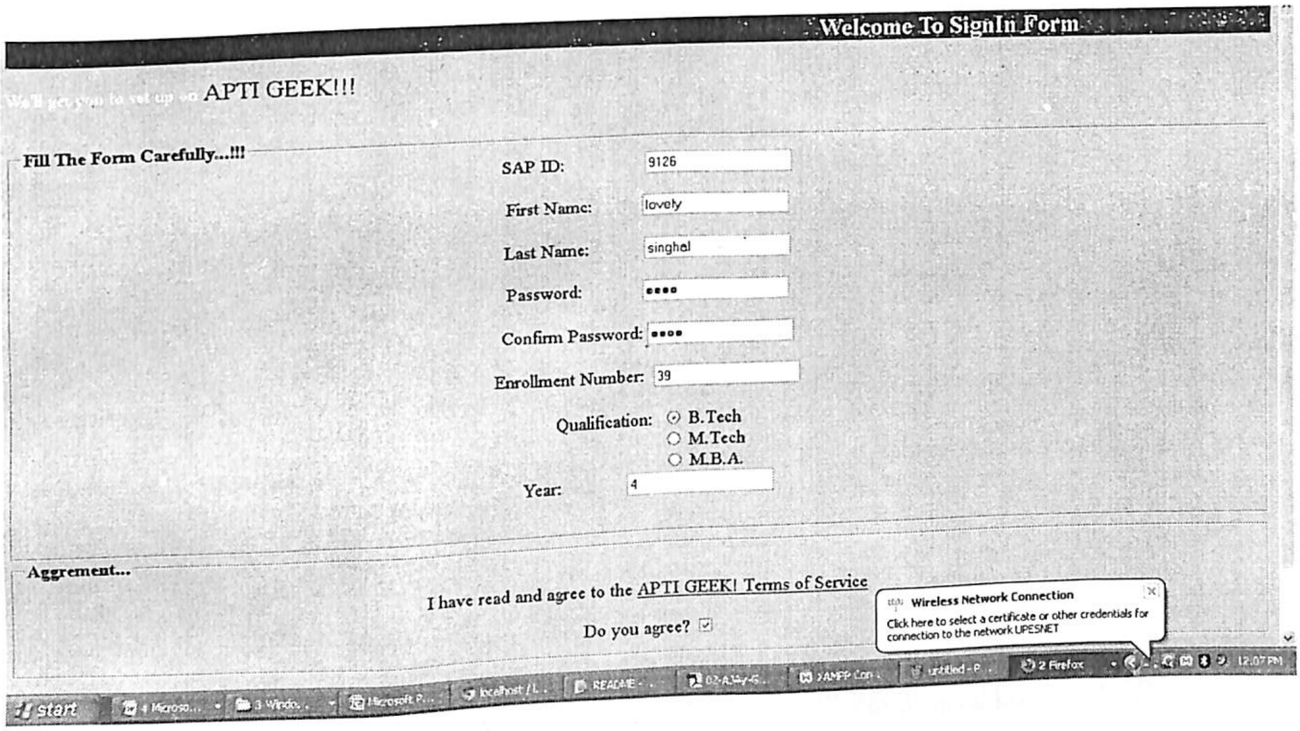

HomePage - Mozilla Firefox
File Edit View History Bookmarks Tools Help
localhost / localhost / major1 / newuser | ...
HomePage
file:///C:/xampp/h...Major%201/home.php
HomePage
file:///C:/xampp/htdocs/Major 1/HomePage.php
Apti Geek
WELCOME TO THE SITE
Home | AreaOfSelection AttemptQuizNow | ResultSheet | FeedbackForm |
start
Major 1
images
C:\xampp\htdocs...
C:\xampp\xampp...
HomePage - Mozil...
Microsoft PowerP...
2:33 PM

What is Apti Geek?

An "APTI GEEK" is an application which is capable of conducting online aptitude tests on the internet. This site is designed to assess logical reasoning or thinking performance. Aptitude Tests will be divided into 3 sections. First section consists of Quantum i.e. purely mathematical questions, Second section deals with verbal reasoning & third section consists of questions based on data interpretations that would be the toughest one. They consist of multiple choice questions. These are typically timed and a typical test might allow 30 minutes for 30 questions and so on. The test will be conducted on the basis of difficulty level. i.e. the user will enter in the first level and then if he clears it than he will reach to the next level according to more

Copyright @AptiGeek.com, All Right Reserved.

Home Page



Welcome To SignIn Form

We'll get you to set up on APTI GEEK!!!

Fill The Form Carefully...!!!

SAP ID: 9126

First Name: lovely

Last Name: singhal

Password: ••••

Confirm Password: ••••

Enrollment Number: 39

Qualification: ● B.Tech ○ M.Tech ○ M.B.A.

Year: 4

Aggrement...

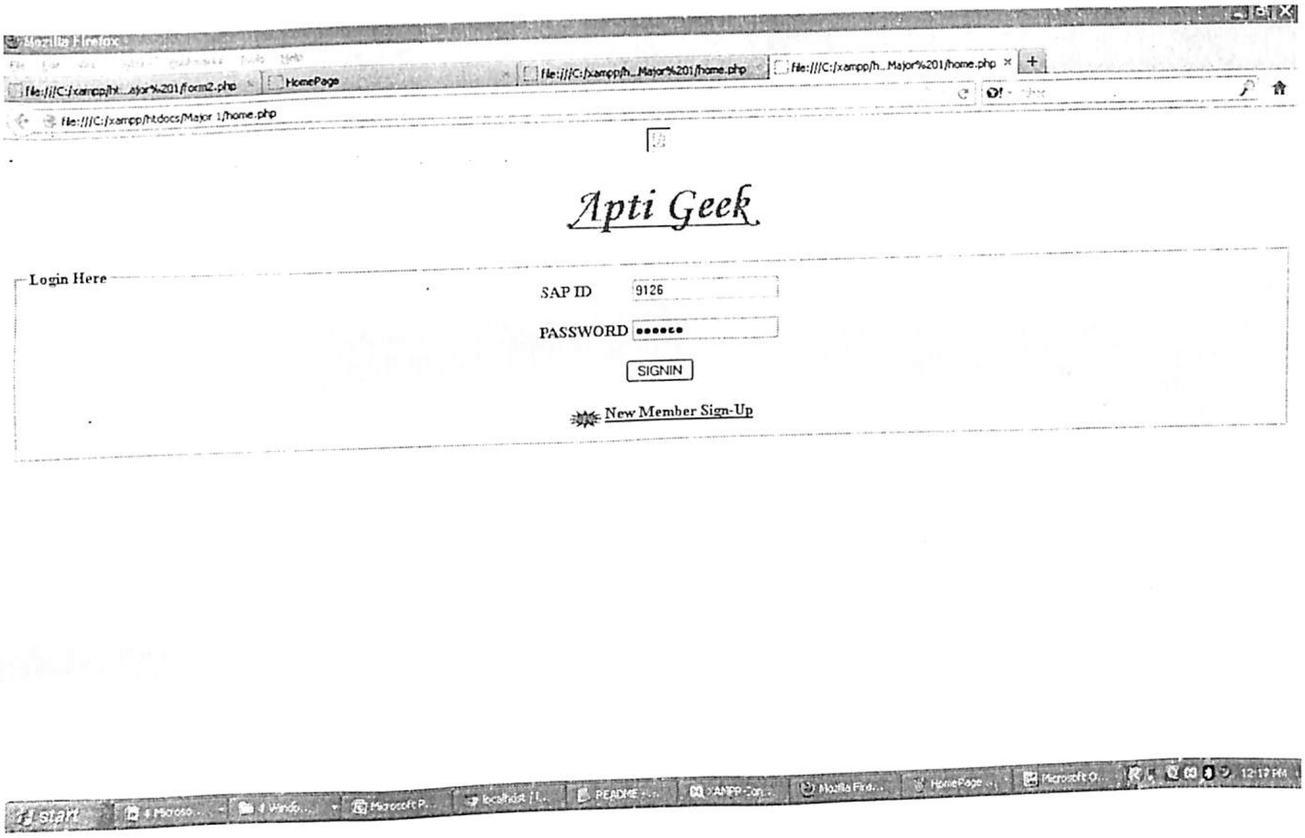
I have read and agree to the APTI GEEK! Terms of Service

Do you agree? ☑

Wireless Network Connection

Click here to select a certificate or other credentials for connection to the network UPESNET

New User SignUp Page



Mozilla Firefox
file:///C:/xampp/ht...ajor%201/form2.php
HomePage
file:///C:/xampp/h...Major%201/home.php
file:///C:/xampp/h...Major%201/home.php
file:///C:/xampp/htdocs/Major 1/home.php
start
Microsoft P...
localhost / l...
README ...
XAMPP Con...
Mozilla Fire...
HomePage ...
Microsoft O...
12:17 PM

Apti Geek

Login Here

SAP ID 9126

PASSWORD ••••••

SIGNIN

New Member Sign-Up

Login Page