

CHAPTER-1

INTRODUCTION

This chapter gives an overview about the aims, objectives, and background and operation environment of the project.

1.1 BACKGROUND OF THE PROJECT

This website “**OPEN DISCUSSION FORUM**” is made for providing a platform for having discussions. This forum provides the platform under one roof to interact with different members which maybe the experts in a particular field or a normal employee for seeking or to give advices. An online space on which anybody can have discussions regarding any research, academic documents or any latest technology free of cost. A website which helps to resolve doubts, queries related to any field by having discussions with other registered users.

With highly growing of the telecommunication infrastructures, such as Internet and the development of the high performance have brought about new era of rapid advances in information technology. Internet has become an ocean of information related to every aspect which has existence in this world. The main purpose of this website is to develop a one roof platform for the effective interaction, effective exposure, and a right direction toward communication. My aim is to provide our users an opportunity to enhance their knowledge by sharing their views on this platform by having discussions with other users. The intended audience is the students and teachers of the college. This website will act as a healthy questionnaire which helps to give and get effective solutions with best of their experience.

Existing systems are the discussion forums open for the whole world but this website is restricted for a particular college as the website is operated by the email-id of the college domain. This will help the students to improve their skills according to their desired technologies by interacting with their teachers and colleagues.

1.1 STATEMENT OF PROJECT

I put my complete effort in order to provide a complete solution to the target users under one roof. It is a website of its own kind that provides a ceramic way to attain and give knowledge. Here the process of knowledge sharing is concerned with the purity of the facts and figures collected from various authentic sources. My aim is to help the passionate people to share their views with its purity and trueness. These different users can guide each other on various aspects. Beyond this my website will give the opportunity to every user say “a lot of eminent personalities within the institute who can share their bright knowledge over the website”. Last but not the least this website also gives the opportunity to the employees for sharing of knowledge and for utilization of the efforts by gaining the knowledge.

This web based project consists of six modules as follows –

- a. Master Records
- b. Registration, Verification and Login
- c. Question
- d. Answer
- e. Assignment
- f. Subscribing domain
- g. Notification

This project can further be more beneficial by adding chatting facility, providing the search feature, provide some visualization reports based on the technologies etc.

As this project is web based project so it can be accessed on any platform with no issues and just with the help of internet connection we can use this website to enhance the knowledge.

CHAPTER 2

SOFTWARE REQUIREMENT AND SPECIFICATION

After analysing the requirements of the task to be performed, the next step is to analyse the problem and understand its context. The first activity in the phase is studying the existing system and other is to understand the requirements and domain of the new system. Both the activities are equally important, but the first activity serves as a basis of giving the functional specifications and then successful design of the proposed system. Understanding the properties and requirements of a new system is more difficult and requires creative thinking and understanding of existing running system is also difficult, improper understanding of present system can lead diversion from solution.

2.1 ANALYSIS MODEL

Analysis model Consists following two models that are used in development of this project.

2.1.1 SDLC Methodologies

This document play a vital role in the development of life cycle (SDLC) as it describes the complete requirement of the system. It means for use by developers and will be the basic during testing phase. Any changes made to the requirements in the future will have to go through formal change approval process.

2.1.2 Iterative Enhancement Model

The iterative enhancement life cycle model counters the third limitation of the waterfall model and tries to combine the benefits of both prototyping and the waterfall model. The basic idea is that the software should be developed in increments, where each increment adds some functional capability to the system until the full system is implemented. At each step extensions and design modifications can be made. An advantage of this approach is that it can result in better testing, since testing each increment is likely to be easier than testing entire system like in the waterfall model. Furthermore, as in prototyping, the

increment provides feedback to the client which is useful for determining the final requirements of the system.

In the first step of iterative enhancement model, a simple initial implementation is done for a subset of the overall problem. This subset is the one that contains some of the key aspects of the problem which are easy to understand and implement, and which forms a useful and usable system. A project control list is created which contains, in an order, all the tasks that must be performed to obtain the final implementation. This project control list gives an idea of how far the project is at any given step from the final system.

Each step consists of removing the next step from the list. Designing the implementation for the selected task, coding and testing the implementation, and performing an analysis of the partial system obtained after this step and updating the list as a result of the analysis. These three phases are called the design phase, implementation phase and analysis phase. The process is iterated until the project control list is empty, at the time the final implementation of the system will be available.

Like other SDLC models, Iterative and incremental development has some specific applications in the software industry. This model is most often used in the following scenarios:

- Requirements of the complete system are clearly defined and understood.
- Major requirements must be defined; however, some functionalities or requested enhancements may evolve with time.
- There is a time to the market constraint.
- A new technology is being used and is being learnt by the development team while working on the project.
- Resources with needed skill set are not available and are planned to be used on contract basis for specific iteration.

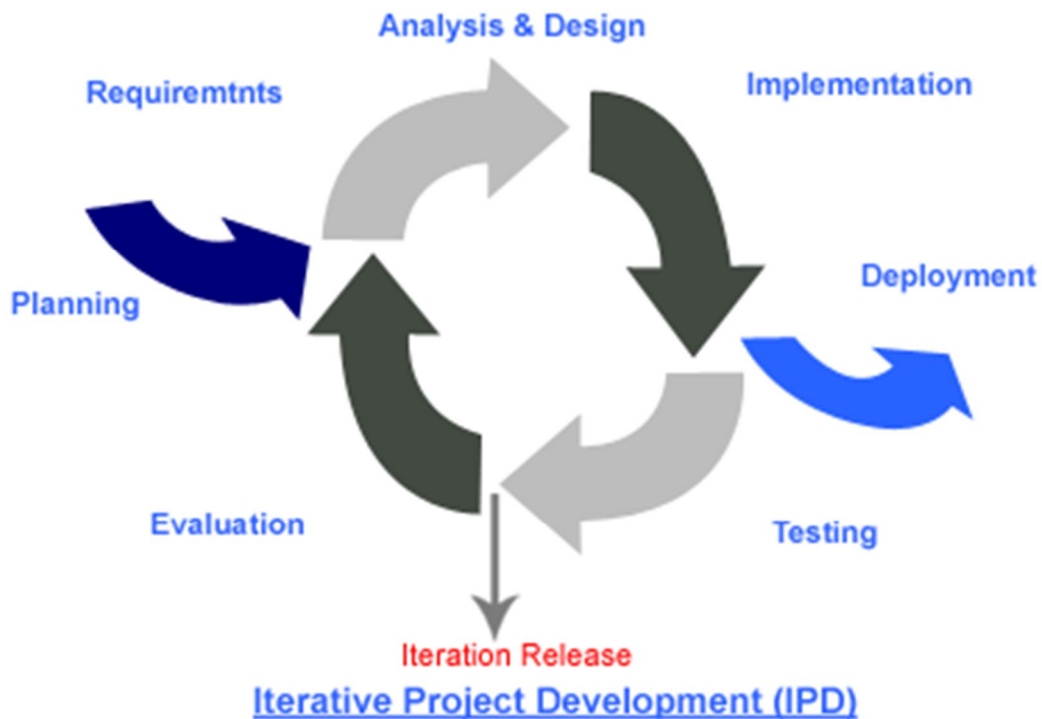


FIGURE 2.1.2.1: ITERATIVE ENHANCEMENT MODEL

2.2 STUDY OF THE SYSTEM:

In the flexibility of the uses the interface has been developed a graphics concept in mind, associated through a browser interface. The GUI'S at the top level have been categorized as

1. Student user interface
2. Teacher user interface

Both the interfaces are almost same the only difference is that the assignment can only be posted by teachers and the list of submission will be displayed to that particular teacher.

2.2.1 MODULES

This web based project consists of modules as follows –

- a. Master Records
- b. Registration, Verification and Login
- c. Question
- d. Answer
- e. Assignment
- f. Subscribing domain
- g. Notification

Master records module: - In this module all the master information's are stored in the database. The master record consist all user relevant, college relevant, domain relevant, department relevant information and all these information are stored in the database initially.

Registration, Verification and Login module: - This Module helps to give the detailed information about the newly entered user, verify it through email verification and then the login to the website is allowed.

Question module: - This module is basically for the registered users they can upload their queries and see the query of other users who has selected the same domain and can up-vote them also which means they want to consider the question as their question also.

Answer module: - This module is also for the registered user only and they can submit answer to the other people's queries and see other people answering or solving their queries.

Assignment module: - In this module the user which are teachers can upload assignments for a particular department and all the students of that department can submit the answers to the assignment provided both these can be done in pdf format.

Subscribing domain module: - This feature is asked to the user on his/her first login and can also edit the selected choices afterwards. In this option the user select the domain he/she want to work on or they are interested in.

Notification module: - This feature provide user the notification of any assignment for them or any other user up-voted their question, or any other user has followed him.

2.3 USER INTERFACE

1. Web browser
2. Internet Connection

2.4 SOFTWARE REQUIREMENTS

This section lists the requirements that are needed to run the system efficiently. The operating system needed for the system to run effectively, the interface to run the application, the integrated development environment to develop the project, and the third-party tool used for editing purposes are as follows:

- I. XAMPP
- II. Browser
- III. PHP
- IV. HTML
- V. CSS
- VI. Java Script

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's name is an acronym for:

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

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 Open SSL 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. 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 Slate among others.

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

The default FTP user is "new user", the default FTP password is "wampp". The default MySQL user is "root" while there is no default MySQL password.

XAMPP 5.6.3 for Windows and Linux, Mac OSX, including:

- Apache 2.4.10
- MySQL 5.6.21
- PHP 5.6.3
- phpMyAdmin 4.2.11
- OpenSSL 1.0.1j
- XAMPP Control Panel 3.2.1
- Webalizer 2.23-04
- Mercury Mail Transport System 4.63
- FileZilla FTP Server 0.9.41
- Tomcat 7.0.56 (with mod_proxyajp as connector)
- Strawberry Perl 7.0.56 Portable

BROWSER:-

A **web browser** (commonly referred to as a **browser**) is a software application for retrieving, presenting and traversing information resources on the World Wide Web. An *information resource* is identified by a Uniform Resource Identifier (URI/URL) and

may be a web page, image, video or other piece of content. Hyperlinks present in resources enable users easily to navigate their browsers to related resources.

Netscape Navigator was sold commercially, as was Opera. Internet Explorer, on the other hand, was bundled free with the Windows operating system (and was also downloadable free), and therefore it was funded partly by the sales of Windows to computer manufacturers and direct to users. Internet Explorer also used to be available for the Mac. It is likely that releasing IE for the Mac was part of Microsoft's overall strategy to fight threats to its quasi-monopoly platform dominance - threats such as web standards and Java - by making some web developers, or at least their managers, assume that there was "no need" to develop for anything other than Internet Explorer. In this respect, IE may have contributed to Windows and Microsoft applications sales in another way, through "lock-in" to Microsoft's browser.

Today, most commercial web browsers are paid by search engine companies to make their engine default, or to include them as another option. For example, Google pays Mozilla, the maker of Firefox, to make Google Search the default search engine in Firefox. Mozilla makes enough money from this deal that it does not need to charge users for Firefox. In addition, Google Search is also (as one would expect) the default search engine in Google Chrome. Users searching for websites or items on the Internet would be led to Google's search results page, increasing ad revenue and which funds development at Google and of Google Chrome.

The most recent major entrant to the browser market is Chrome, first released in September 2008. Chrome's take-up has increased significantly year by year, by doubling its usage share from 8% to 16% by August 2011. This increase seems largely to be at the expense of Internet Explorer, whose share has tended to decrease from month to month. In December 2011, Chrome overtook Internet Explorer 8 as the most widely used web browser but still had lower usage than all versions of Internet Explorer combined. Chrome's user-base continued to grow and in May 2012, Chrome's usage passed the usage of all versions of Internet Explorer combined. By April 2014, Chrome's usage had hit 45%

This process begins when the user inputs a Uniform Resource Locator (URL), for example *http://assignmentguru.asia/*, into the browser. The prefix of the URL, the Uniform Resource Identifier or URI, determines how the URL will be interpreted. The most commonly used kind of URI starts with *http:* and identifies a resource to be retrieved over the Hypertext Transfer Protocol (HTTP). Many browsers also support a variety of other prefixes, such as *https:* for HTTPS, *ftp:* for the File Transfer Protocol, and *file:* for local. Prefixes that the web browser cannot directly handle are often handed off to another application entirely. For example, *mailto:* URIs is usually passed to the user's default e-mail application, and *news:* URIs are passed to the user's default newsgroup reader.

In the case of *http*, *https*, *file*, and others, once the resource has been retrieved the web browser will display it. HTML and associated content (image files, formatting information such as CSS, etc.) is passed to the browser's layout engine to be transformed from markup to an interactive document, a process known as "rendering". Aside from HTML, web browsers can generally display any kind of content that can be part of a web page. Most browsers can display images, audio, video, and XML files, and often have plugins to support Flash applications and Java applets. Upon encountering a file of an unsupported type or a file that is set up to be downloaded rather than displayed, the browser prompts the user to save the file to disk.

Available web browsers range in features from minimal, text-based user interfaces with bare-bones support for HTML to rich user interfaces supporting a wide variety of file formats and protocols. Browsers which include additional components to support e-mail, Usenet news, and Internet Relay Chat (IRC), are sometimes referred to as "Internet suites" rather than merely "web browsers".

All major web browsers allow the user to open multiple information resources at the same time, either in different browser windows or in different tabs of the same window. Major browsers also include pop-up blockers to prevent unwanted windows from "popping up" without the user's consent.

Most web browsers can display a list of web pages that the user has *bookmarked* so that the user can quickly return to them. Bookmarks are also called "Favorites" in Internet. In addition, all major web browsers have some form of built-in web feed aggregator. In Firefox, web feeds are formatted as "live bookmarks" and behave like a folder of bookmarks corresponding to recent entries in the feed. In Opera, a more traditional feed reader is included which stores and displays the contents of the feed.

Furthermore, most browsers can be extended via plug-ins, downloadable components that provide additional features

Most major web browsers have these user interface elements in common:

- *Back* and *forward* buttons to go back to the previous resource and forward respectively.
- A *refresh* or *reload* button to reload the current resource.
- A *stop* button to cancel loading the resource. In some browsers, the stop button is merged with the reload button.
- A *home* button to return to the user's home page.
- An address bar to input the Uniform Resource Identifier (URI) of the desired resource and display it.
- A search bar to input terms into a search engine. In some browsers, the search bar is merged with the address bar.
- A status bar to display progress in loading the resource and also the URI of links when the cursor hovers over them, and page zooming capability.
- The *viewport*, the visible area of the webpage within the browser window.
- The ability to view the HTML source for a page.

Major browsers also possess incremental find features to search within a web page.

PHP (Hypertext Pre-processor):-

PHP is a server-side scripting language designed for web development but also used as a general-purpose programming language. As of January 2013, PHP was installed on more

than 240 million websites (39% of those sampled) and 2.1 million web servers. Originally created by RasmusLerdorf in 1994, the reference implementation of PHP (powered by the Zend Engine) is now produced by The PHP Group. While PHP originally stood for *Personal Home Page*, it now stands for *PHP: Hypertext Preprocessor*, which is a recursive backronym.

PHP code can be simply mixed with HTML code, or it can be used in combination with various templating engines and web frameworks. PHP code is usually processed by a PHP interpreter, which is usually implemented as a web server's native module or a Common Gateway Interface (CGI) executable. After the PHP code is interpreted and executed, the web server sends resulting output to its client, usually in form of a part of the generated web page; for example, PHP code can generate a web page's HTML code, an image, or some other data. PHP has also evolved to include a command-line interface (CLI) capability and can be used in standalone graphical applications.

The canonical PHP interpreter, powered by the Zend Engine, is free software released under the PHP License. PHP has been widely ported and can be deployed on most web servers on almost every operating system and platform, free of charge.

Despite its popularity, no written specification or standard existed for the PHP language until 2014, leaving the canonical PHP interpreter as a *de facto* standard. Since 2014, there is ongoing work on creating a formal PHP specification.

HTML:-

Hypertext Markup Language, commonly referred to as HTML, is the standard markup language used to create web pages. It is written in the form of HTML elements consisting of *tags* enclosed in angle brackets (like `<html>`). HTML tags most commonly come in pairs like `<h1>` and `</h1>`, although some tags represent *empty elements* and so are unpaired, for example ``. The first tag in a pair is the *start tag*, and the second tag is the *end tag* (they are also called *opening tags* and *closing tags*).

Web browsers can read HTML files and compose them into visible or audible web pages. Browsers do not display the HTML tags and scripts, but use them to interpret the content of the page. HTML describes the structure of a website semantically along with cues for presentation, making it a markup language, rather than a programming language.

HTML elements form the building blocks of all websites. HTML allows images and objects to be embedded and can be used to create interactive forms. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. It can embed scripts written in languages such as JavaScript which affect the behavior of HTML web pages.

Web browsers can also refer to Cascading Style Sheets (CSS) to define the look and layout of text and other material. The World Wide Web Consortium (W3C), maintainer of both the HTML and the CSS standards, encourages the use of CSS over explicit presentational HTML.

In 1980, physicist Tim Berners-Lee, who was a contractor at CERN, proposed and prototyped ENQUIRE, a system for CERN researchers to use and share documents. In 1989, Berners-Lee wrote a memo proposing an Internet-based hypertext system. Berners-Lee specified HTML and wrote the browser and server software in late 1990. That year, Berners-Lee and CERN data systems engineer Robert Cailliau collaborated on a joint request for funding, but the project was not formally adopted by CERN. In his personal notes from 1990 he listed "some of the many areas in which hypertext is used" and put an encyclopedia first.

HTML is a markup language that web browsers use to interpret and compose text, images and other material into visual or audible web pages. Default characteristics for every item of HTML markup are defined in the browser, and these characteristics can be altered or enhanced by the web page designer's additional use of CSS. Many of the text elements are found in the 1988 ISO technical report TR 9537 *Techniques for using SGML*, which in turn covers the features of early text formatting languages such as that used by the RUNOFF command developed in the early 1960s for the CTSS (Compatible Time-Sharing System)

operating system: these formatting commands were derived from the commands used by typesetters to manually format documents. However, the SGML concept of generalized markup is based on elements (nested annotated ranges with attributes) rather than merely print effects, with also the separation of structure and markup; HTML has been progressively moved in this direction with CSS.

Berners-Lee considered HTML to be an application of SGML. It was formally defined as such by the Internet Engineering Task Force (IETF) with the mid-1993 publication of the first proposal for an HTML specification: "Hypertext Markup Language (HTML)" Internet-Draft by Berners-Lee and Dan Connolly, which included an SGML Document Type Definition to define the grammar. The draft expired after six months, but was notable for its acknowledgment of the NCSA Mosaic browser's custom tag for embedding in-line images, reflecting the IETF's philosophy of basing standards on successful prototypes. Similarly, Dave Raggett's competing Internet-Draft, "HTML+ (Hypertext Markup Format)", from late 1993, suggested standardizing already-implemented features like tables and fill-out forms.

CSS:-

Cascading Style Sheets (CSS) is a style sheet language used for describing the look and formatting of a document written in a markup language. While most often used to change the style of web pages and user interfaces written in HTML and XHTML, the language can be applied to any kind of XML document, including plain XML, SVG and XUL. Along with HTML and JavaScript, CSS is a cornerstone technology used by most websites to create visually engaging Webpages, user interfaces for web applications, and user interfaces for many mobile applications.

CSS is designed primarily to enable the separation of document content from document presentation, including elements such as the layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple HTML pages to share formatting by

specifying the relevant CSS in a separate .css file, and reduce complexity and repetition in the structural content, such as semantically insignificant tables that were widely used to format pages before consistent CSS rendering was available in all major browsers. CSS makes it possible to separate presentation instructions from the HTML content in a separate file or style section of the HTML file. For each matching HTML element, it provides a list of formatting instructions. For example, a CSS rule might specify that "all heading 1 elements should be bold," leaving pure semantic HTML markup that asserts "this text is a level 1 heading" without formatting code such as a `<bold>` tag indicating how such text should be displayed.

The CSS specification describes a priority scheme to determine which style rules apply if more than one rule matches against a particular element. In this so-called *cascade*, priorities or *weights* are calculated and assigned to rules, so that the results are predictable.

The CSS specifications are maintained by the World Wide Web Consortium (W3C). Internet media type (MIME type) `text/css` is registered for use with CSS by RFC 2318 (March 1998). The W3C operates a free CSS validation service for CSS documents.

JavaScript:-

JavaScript was originally developed by Brendan Eich, while he was working for Netscape Communications Corporation. Indeed, while competing with Microsoft for user adoption of web technologies and platforms, Netscape considered their client-server offering a distributed OS with a portable version of Sun Microsystems' Java providing an environment in which applets could be run. Because Java was a competitor of C++ and aimed at professional programmers, Netscape also wanted a light weight interpreted language that would complement Java by appealing to nonprofessional programmers, like Microsoft's Visual Basic (see JavaScript and Java).

Although it was developed under the name *Mocha*, the language was officially called *Live Script* when it first shipped in beta releases of Netscape Navigator 2.0 in September 1995,

but it was renamed JavaScript when it was deployed in the Netscape browser version 2.0B3.

The change of name from Live Script to JavaScript roughly coincided with Netscape adding support for Java technology in its Netscape Navigator web browser. The final choice of name caused confusion, giving the impression that the language was a spin-off of the Java programming language, and the choice has been characterized as a marketing ploy by Netscape to give JavaScript the cachet of what was then the hot new web programming language.

JavaScript (JS), also known as **ECMAScript** (the untrademarked name used for the standard), is a dynamic programming language. It is most commonly used as part of web browsers, whose implementations allow client-side scripts to interact with the user, control the browser, communicate asynchronously, and alter the document content that is displayed. It is also used in server-side network programming with runtime environments such as Node.js, game development and the creation of desktop and mobile applications. With the rise of the single-page web app and JavaScript-heavy sites, it is increasingly being used as a compile target for source-to-source compilers from both dynamic languages and static languages. In particular, Emscripten and highly optimized JIT compilers, in tandem with asm.js that is friendly to AOT compilers like Odin Monkey, have enabled C and C++ programs to be compiled into JavaScript and execute at near-native speeds, making JavaScript to be considered the "assembly language of the web", according to its creator and others.

JavaScript is classified as a prototype-based scripting language with dynamic typing and first-class functions. This mix of features makes it a multi-paradigm language, supporting object-oriented, imperative, and functional programming styles.

Despite some naming, syntactic, and standard library similarities, JavaScript and Java are otherwise unrelated and have very different semantics. The syntax of JavaScript is actually derived from C, while the semantics and design are influenced by the self and Scheme programming languages. JavaScript is also used in environments that

aren't web-based, such as PDF documents, site-specific browsers, and desktop widgets. Newer and faster JavaScript virtual machines (VMs) and platforms built upon them have also increased the popularity of JavaScript for server-side web applications. On the client side, JavaScript has been traditionally implemented as an interpreted language, but more recent browsers perform just-in-time compilation.

2.5 HARDWARE REQUIREMENTS

This section lists the hardware requirements that are needed to run the system efficiently. The hardware uses to run this system successfully is listed as below:-

- I. Processor
- II. RAM
- III. Storage
- IV. Keyboard
- V. Mouse
- VI. Monitor
- VII. NIC

PROCESSOR:-

A **processor** or **processing unit** is an electronic circuit which performs operations on some external data source, usually memory or some other data stream. The term is frequently used to refer to the central processor (central processing unit) in a system, but typical computer systems (especially SoCs) combine a number of specialised "processors". For this website to run efficiently minimum PENTIUM III 866 MHz is required.

RAM:-

Random-access memory (RAM) is a form of computer data storage which stores frequently used program instructions to increase the general speed of a system. A random-access memory device allows data items to be read or written in almost the same amount of time

irrespective of the physical location of data inside the memory. In contrast, with other direct-access data storage media such as hard disks, CD-RWs, DVD-RWs and the older drum memory, the time required to read and write data items varies significantly depending on their physical locations on the recording medium, due to mechanical limitations such as media rotation speeds and arm movement. Min 128 MB SD RAM is required for this website to run efficiently.

KEYBOARD:-

In computing, a **keyboard** is a typewriter-style device, which uses an arrangement of buttons or keys, to act as mechanical levers or electronic switches. Following the decline of punch cards and paper tape, interaction via teleprompter-style keyboards became the main input device for computers.

A keyboard typically has characters engraved or printed on the keys and each press of a key typically corresponds to a single written symbol. However, to produce some symbols requires pressing and holding several keys simultaneously or in sequence. While most keyboard keys produce letters, numbers or signs (characters), other keys or simultaneous key presses can produce actions or execute computer commands.

Despite the development of alternative input devices, such as the mouse, touch screen, pen devices, character recognition and voice recognition, the keyboard remains the most commonly used device for direct (human) input of alphanumeric data into computers.

MOUSE:-

A mouse typically controls the motion of a pointer in two dimensions in a graphical user interface (GUI). The mouse turns movements of the hand backward and forward, left and right into equivalent electronic signals that in turn are used to move the pointer.

The relative movements of the mouse on the surface are applied to the position of the pointer on the screen, which signals the point where actions of the user take place, so that the hand movements are replicated by the pointer. Clicking or hovering (stopping movement while the cursor is within the bounds of an area) can select files, programs or

actions from a list of names, or (in graphical interfaces) through small images called "icons" and other elements. For example, a text file might be represented by a picture of a paper notebook, and clicking while the cursor hovers this icon might cause a text editing program to open the file in a window.

Different ways of operating the mouse cause specific things to happen in the GUI:

- Click: pressing and releasing a button.
 - (left) Single-click: clicking the main button.
 - (left) Double-click: clicking the button two times in quick succession counts as a different gesture than two separate single clicks.
 - (left) Triple-click: clicking the button three times in quick succession.
 - Right-click: clicking the secondary button.
- Drag: pressing and holding a button, then moving the mouse without releasing. (Using the command "drag with the right mouse button" instead of just "drag" when one instructs a user to drag an object while holding the right mouse button down instead of the more commonly used left mouse button.)
- Button chording (a.k.a. Rocker navigation).
 - Combination of right-click then left-click.
 - Combination of left-click then right-click or keyboard letter.
 - Combination of left or right-click and the mouse wheel.
- Moving the pointer a long distance: When a practical limit of mouse movement is reached, one lifts up the mouse, brings it to the opposite edge of the working area while it is held above the surface, and then replaces it down onto the working surface

MONITOR:-

A desktop monitor is typically provided with a stand from the manufacturer which lifts the monitor up to a more ergonomic viewing height. The stand may be attached to the monitor using a proprietary method or may use, or be adaptable to, a Video Electronics Standards Association, VESA, standard mount. Using a VESA standard mount allows the monitor to

be used with an after-market stand once the original stand is removed. Stands may be fixed or offer a variety of features such as height adjustment, horizontal swivel, and landscape or portrait screen orientation.

NIC:-

A network interface controller (NIC, also known as a network interface card, network adapter, LAN adapter, and by similar terms) is a computer hardware component that connects a computer to a computer network.

Early network interface controllers were commonly implemented on expansion cards that plugged into a computer bus. The low cost and ubiquity of the Ethernet standard means that most new computers have a network interface built into the motherboard. The network controller implements the electronic circuitry required to communicate using a specific physical layer and data link layer standard such as Ethernet, Fiber Channel, Wi-Fi or Token Ring. This provides a base for a full network protocol stack, allowing communication among small groups of computers on the same local area network (LAN) and large-scale network communications through routable protocols, such as Internet Protocol (IP).

Although other network technologies exist, IEEE 802 networks including the Ethernet variants have achieved near-ubiquity since the mid-1990s.

The NIC allows computers to communicate over a computer network, either by using cables or wirelessly. The NIC is both a physical layer and data link layer device, as it provides physical access to a networking medium and, for IEEE 802 and similar networks, provides a low-level addressing system through the use of MAC addresses that are uniquely assigned to network interfaces.

CHAPTER 3

FEASIBILITY STUDY

As per IBM report, “31% projects are cancelled before they are completed.” Canceling a project is neither good nor bad but canceling a project later than necessary is bad. Thus the amount of work done before a decision is made to cancel a project should be minimum (10 to 20 % of the total work). Feasibility is a practical extent to which a project can be performed successfully.

The feasibility study ends with a feasibility report, a technical document which enables project team members, customers, or upper management make a decision about whether to continue with the project or not.

3.1 TECHNICAL FEASIBILITY

Technical feasibility assesses the current technological resources, which are required to accomplish the user requirement in the software within the allocated time and for this, the software development team ascertains whether the current resources and technology can be upgraded or added to accomplish specified user requirements.

Technical feasibility performs the following tasks:

- It determines whether the relevant technology is stable and established.
So, our project is made on Php language which is well stabled and established.
- It ascertains that the technology chosen for software development has large number of users so that they can be consulted when problems arise, or when improvements are required.
- Our project is a web portal which can be easily accessed by large no. of users and is user friendly so suggestions or improvements can also be made.

3.2 OPERATIONAL FEASIBILITY

Operational feasibility assesses the extent to which the required software performs a series of steps to solve a business problems and user requirements.

So, for our project to operate users just need an internet connection and a web browser to be installed on their device. After accessing the portal they will have a log in screen in front of them where they have to provide their username and the password by which they have registered on the website and then they can access the features of the portal.

ODF is user friendly and provides easy access to all the resources.

3.3 ECONOMIC FEASIBILITY

Economic feasibility determines whether the required software is capable of generating financial gains for an organization. It involves the cost incurred on the software development team, estimated cost of hardware and software, cost of performing feasibility study and so on.

Our project is very much feasible as users just need to have an internet connection and a web browser to run the software. From the organization point of view the software for the development of project is free of cost and the cost of hardware is also feasible.

And from the developer or managerial point of view, we can generate the money by using different kind of advertisements and also by using proper data mining after some time.

CHAPTER 4

SYSTEM DESIGN

Systems design is the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. Systems design could be seen as the application of systems theory to product development. The increasing importance of software running on generic platforms has enhanced the discipline of software engineering. Object-oriented analysis and design methods are becoming the most widely used methods for computer systems design. The UML has become the standard language in object-oriented analysis and design. It is widely used for modeling software systems and is increasingly used for high designing non-software systems and organizations.

Architectural design

The architectural design of a system emphasizes on the design of the systems architecture which describes the structure, behavior, and more views of that system and analysis.

Logical design

The logical design of a system pertains to an abstract representation of the data flows, inputs and outputs of the system. This is often conducted via modeling, using an over-abstract (and sometimes graphical) model of the actual system. In the context of systems design are included. Logical design includes ER Diagrams i.e. Entity Relationship Diagrams.

Physical design

The physical design relates to the actual input and output processes of the system. This is laid down in terms of how data is input into a system, how it is verified/ authenticated, how it is processed, and how it is display.

4.1 ACTIVITY DIAGRAM

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams are intended to model both computational and organizational processes (i.e. workflows). Activity diagrams show the overall flow of control.

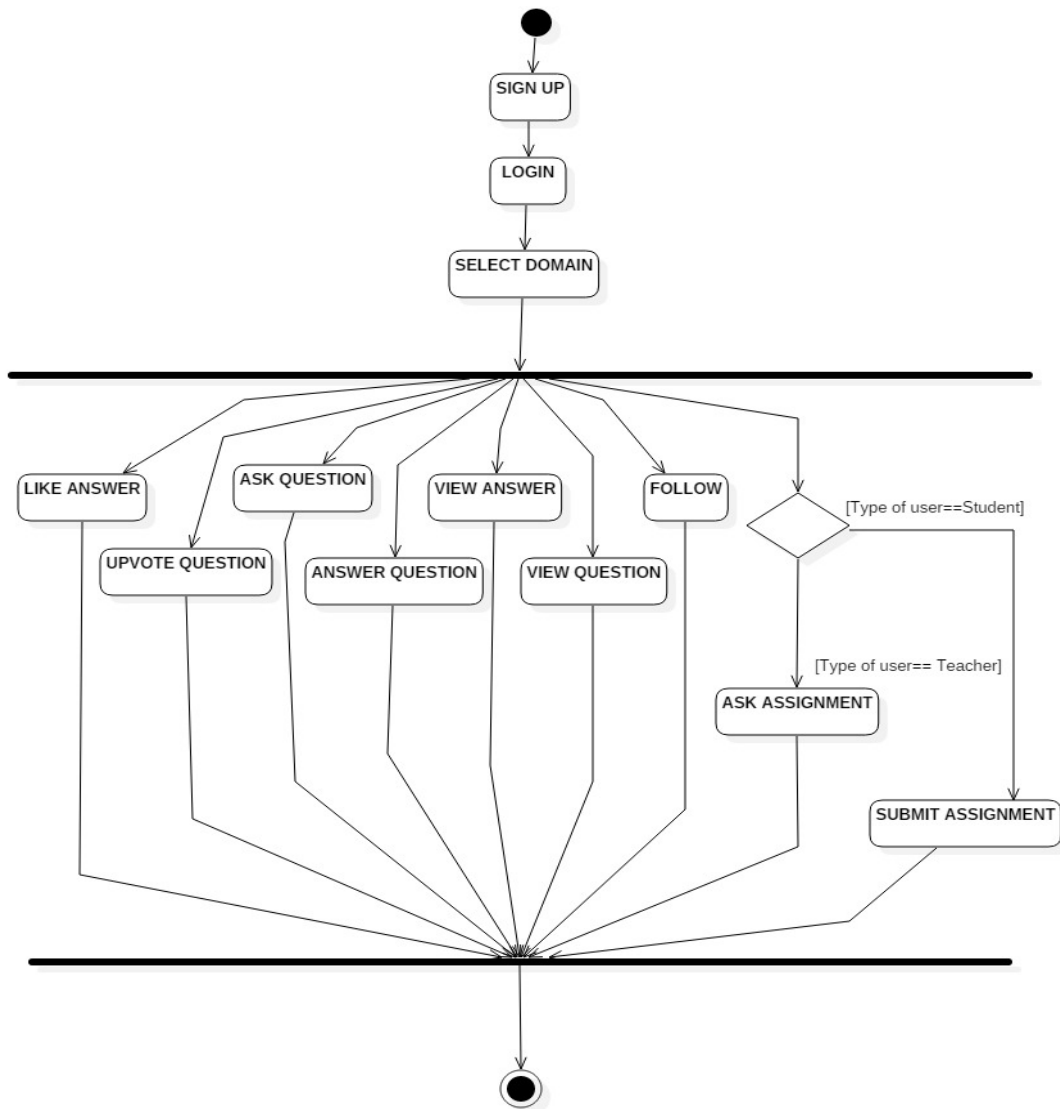


FIGURE 4.1.1: ACTIVITY DIAGRAM

4.2 USE CASE DIAGRAM

In software and systems engineering, a **use case** is a list of steps, typically defining interactions between a role (known in Unified (UML) as an "actor") and a system, to achieve a goal. The actor can be a human, an external system, or time.

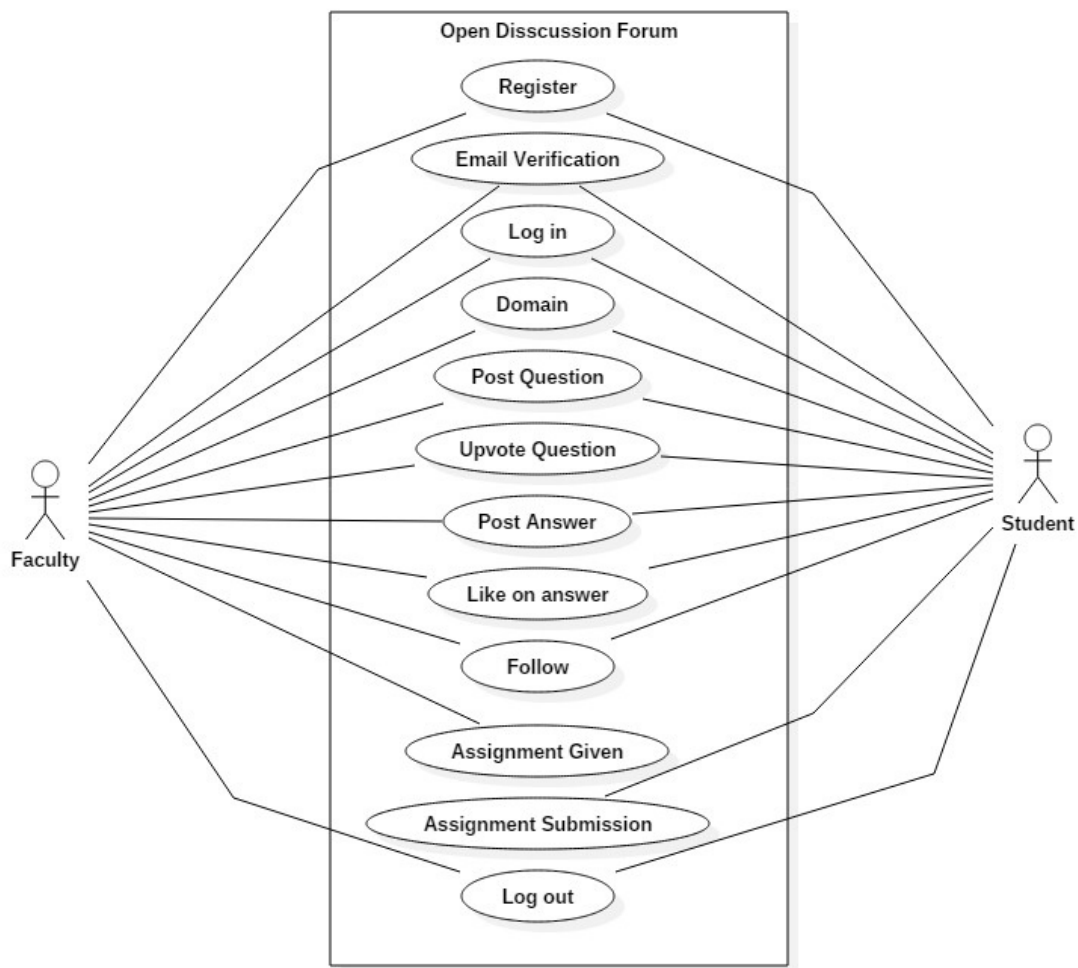


FIGURE 4.2.1: USE CASE DIAGRAM

4.3 DATA FLOW DIAGRAMS

A **Data Flow Diagram (DFD)** is a graphical representation of the "flow" of data through an information system, modeling its *process* aspects. A DFD is often used as a preliminary step to create an overview of the system, which can later be elaborated. DFDs can also be used for the visualization of data processing (structured design).

0 Level DFD:-

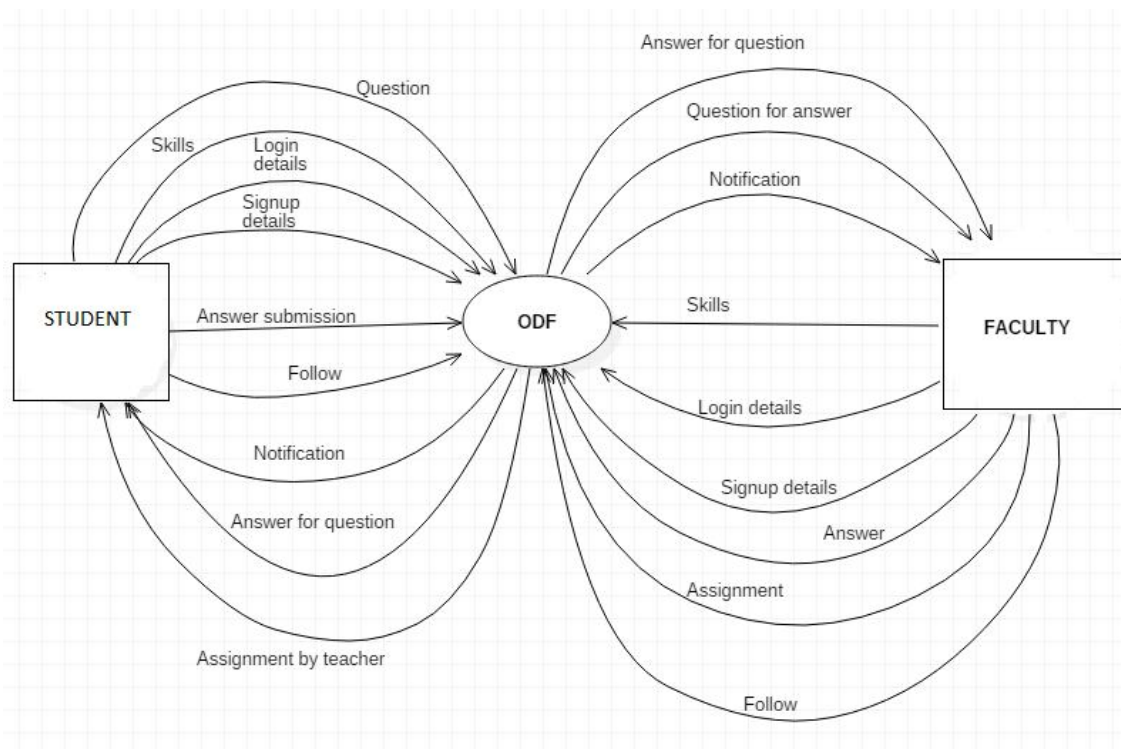


FIGURE 4.3.1: 0 Level DFD

1 Level DFD:-

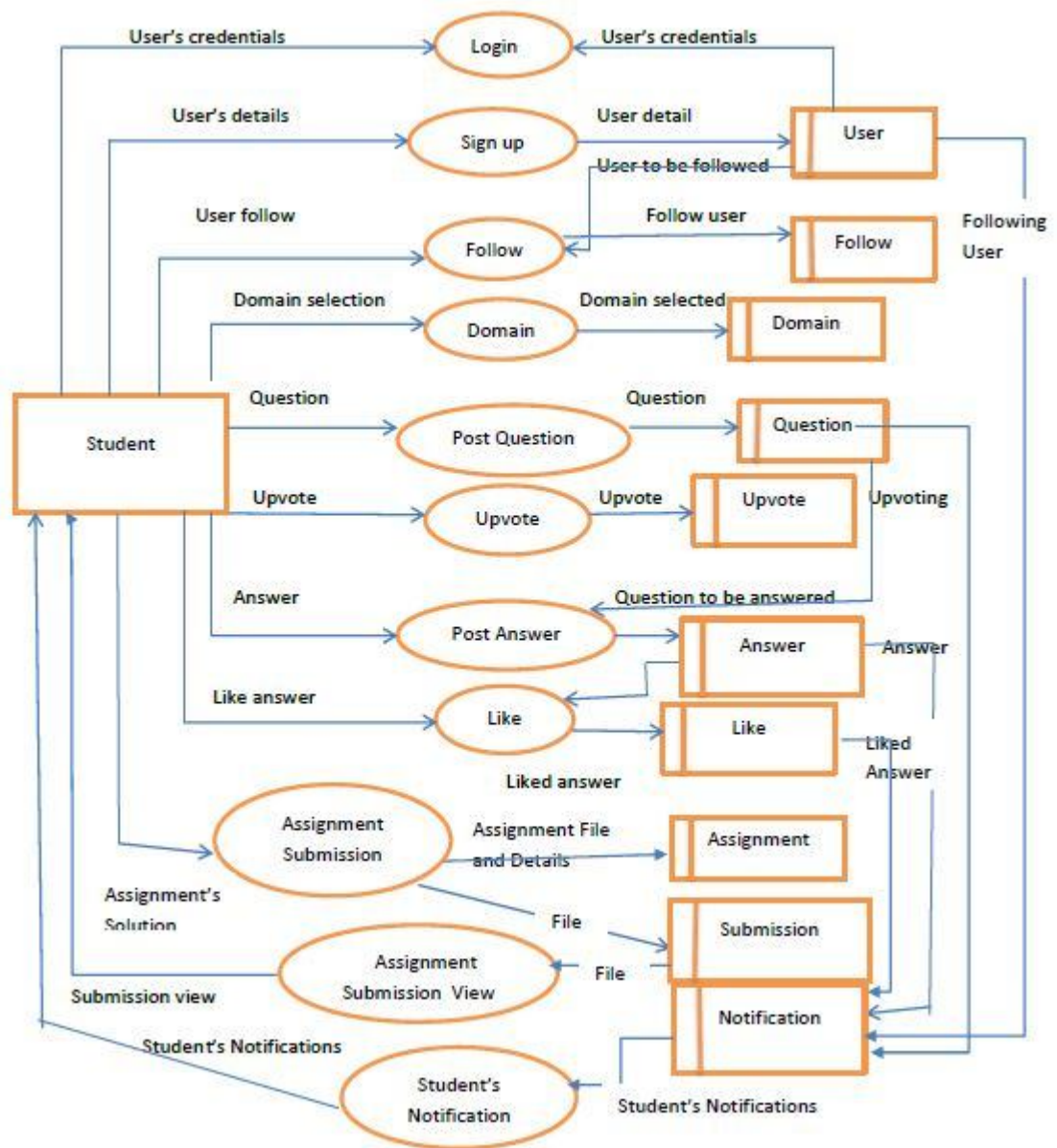


FIGURE 4.3.2: 1 Level DFD for Students

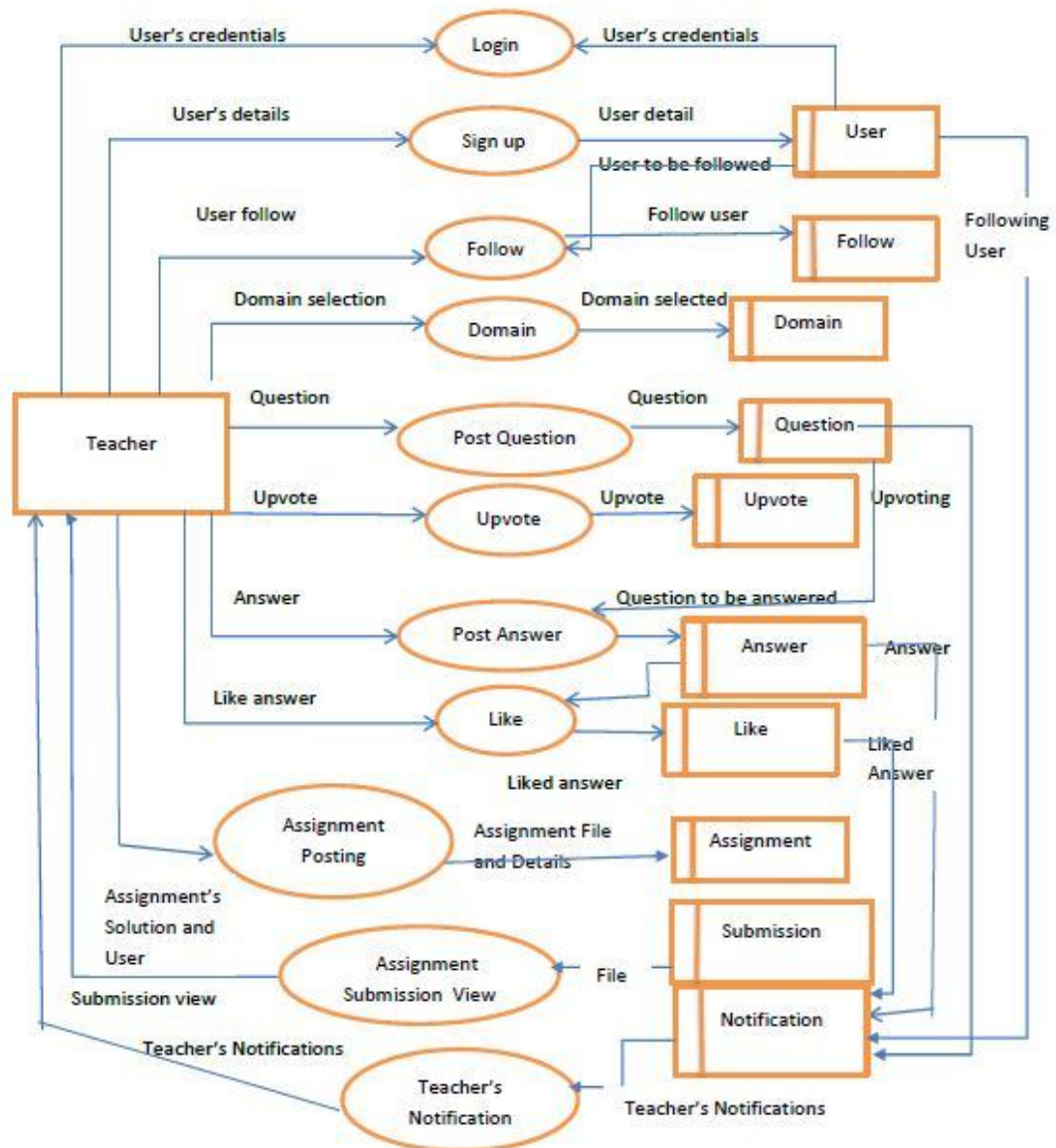


FIGURE 4.3.2: 1 Level DFD for Teachers

4.4 E-R Diagram:-

In software engineering, an **entity–relationship model (ER model)** is a data model for describing the data or information aspects of a business domain or its process requirements, in an abstract way that lends itself to ultimately being implemented in a database such as relational database. The main components of ER models are entities (things) and the relationships that can exist among them.

Entity–relationship modeling was developed by Peter Chen and published in a 1976 paper. However, variants of the idea existed previously, and have been devised subsequently such as super type and subtype data entities and commonality relationships.

An entity–relationship model is a systematic way of describing and defining a business process. The process is modeled as components (*entities*) that are linked with each other by *relationships* that express the dependencies and requirements between them, such as: one building may be divided into zero or more apartments, but one apartment can only be located in one building. Entities may have various properties (*attributes*) that characterize them. Diagrams created to represent these entities, attributes, and relationships graphically are called entity–relationship diagrams.

An ER model is typically implemented as a database. In the case of a relational database, which stores data in tables, every row of each table represents one instance of an entity. Some data fields in these tables point to indexes in other tables; such pointers represent the relationships.

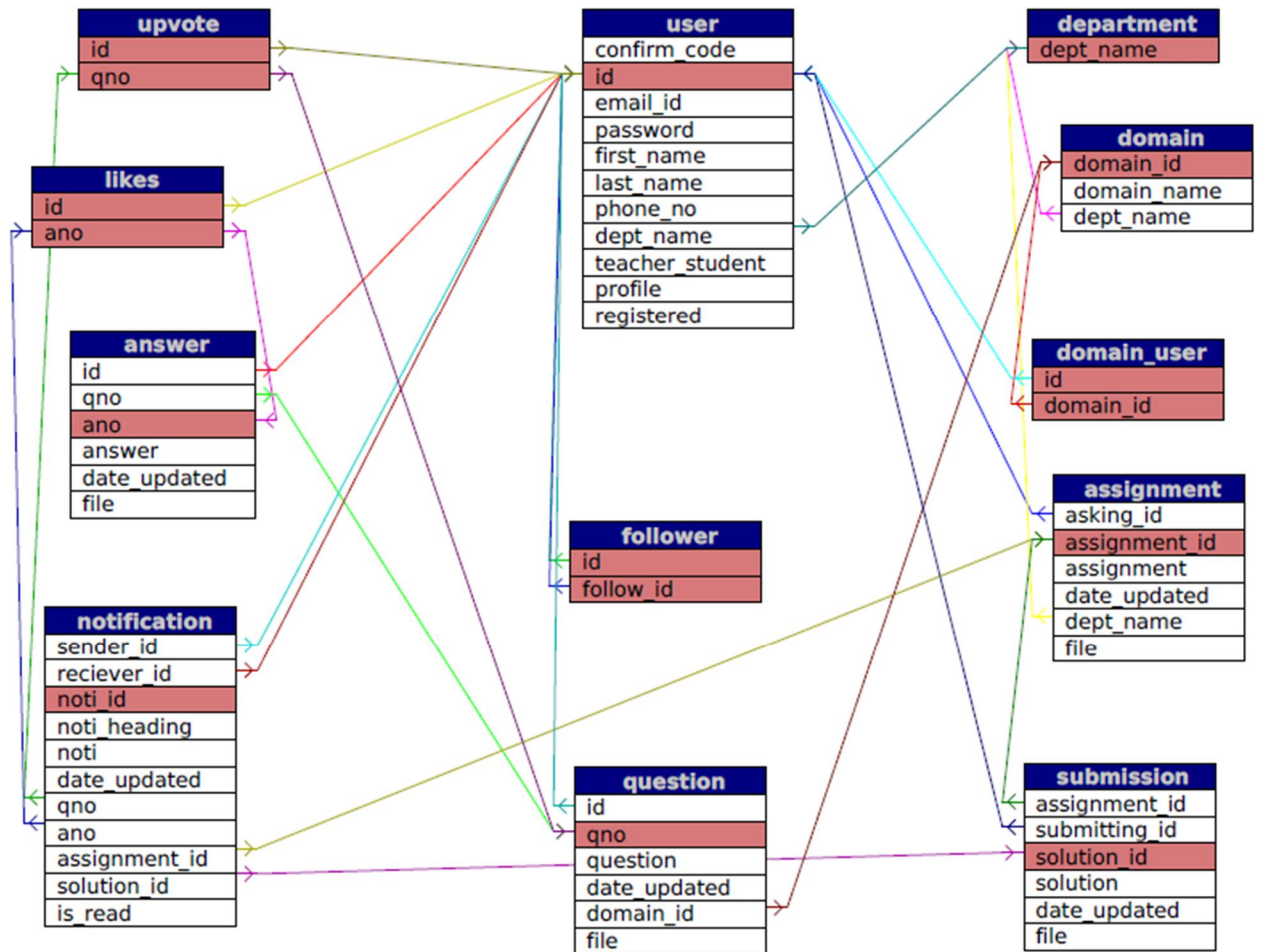


FIGURE 4.3.3: Relational Model

CHAPTER 5

TESTING

5.1 ABOUT TESTING

Software testing is an investigation conducted to provide stakeholders with information about the quality of the product or service under test. Software testing can also provide an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation. Test techniques include, but are not limited to, the process of executing a program or application with the intent of finding software bugs (errors or other defects).

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

- meets the requirements that guided its design and development,
- responds correctly to all kinds of inputs,
- performs its functions within an acceptable time,
- is sufficiently usable,
- can be installed and run in its intended environments, and
- Achieves the general result its stakeholder's desire.

As the number of possible tests for even simple software components is practically infinite, all software testing uses some strategy to select tests that are feasible for the available time and resources. As a result, software testing typically (but not exclusively) attempts to execute a program or application with the intent of finding software bugs (errors or other defects).

Software testing can provide objective, independent information about the quality of software and risk of its failure to users and/or sponsors.

5.2 TEST CASES FOR EACH MODULE

TABLE 5.2.1: TEST CASE FOR LOGIN

TEST CASE ID: 1	
BEFORE EXECUTION	AFTER EXECUTION
INPUT : Email Id: amol.sharma@mitmeerut.ac.in Password: 7417320092 EXPECTED OUTPUT: Redirected to home page. DESIGNED BY : Vasu Garg DATE : 01/03/2017	ACTUAL OUTPUT: Redirected to home page. STATUS : Pass RUN BY : Vishal Chauhan DATE : 01/03/2017

TABLE 5.2.2: TEST CASE FOR LOGIN

TEST CASE ID: 2	
BEFORE EXECUTION	AFTER EXECUTION
INPUT: Email_id: vishal.chauhan.cs.2013@mitmeerut.as.in Password: vishalch EXPECTED OUTPUT: Redirected to home page. DESIGNED BY : Vasu Garg DATE : 02/03/2017	ACTUAL OUTPUT: Redirected to home page. STATUS : Pass RUN BY : Vishal Chauhan DATE : 03/03/2017

TABLE 5.2.3: TEST CASES FOR LOGIN

TEST CASE ID: 3	
BEFORE EXECUTION	AFTER EXECUTION
INPUT : Email Id: shivika.tyagi.cs.2013@mitmeerut.ac.in Password: 7417320092 EXPECTED OUTPUT: Redirect to pre home page to select the domain. DESIGNED BY : Vasu Garg DATE : 04/03/2017	ACTUAL OUTPUT: Redirect to pre home page to select the domain. STATUS : Pass RUN BY : Shivika Tyagi DATE : 05/03/2017

TABLE 5.2.4: TEST CASES FOR LOGIN

TEST CASE ID: 4	
BEFORE EXECUTION	AFTER EXECUTION
INPUT : Email Id: neeraj.kumar.cs.2013@mitmeerut.ac.in Password: 555555 EXPECTED OUTPUT: Alert Email not registered. DESIGNED BY : Vasu Garg DATE : 06/03/2017	ACTUAL OUTPUT: Alert Email not registered. STATUS : Pass RUN BY : Vishal Chauhan DATE : 07/03/2017

TABLE 5.2.5: TEST CASES FOR QUESTION POSTING

TEST CASE ID: 5	
BEFORE EXECUTION	AFTER EXECUTION
INPUT: Question and Domain set in form. EXPECTED OUTPUT: Data Successfully uploaded in database and retrieved. DESIGNED BY : Vasu Garg DATE : 08/03/2017	ACTUAL OUTPUT: Data Successfully uploaded in database and retrieved. STATUS : Pass RUN BY : Shivika Tyagi DATE : 09/03/2017

TABLE 5.2.6: TEST CASES FOR QUESTION RETRIEVAL

TEST CASE ID: 6	
BEFORE EXECUTION	AFTER EXECUTION
INPUT : Question of domain not subscribed by user EXPECTED OUTPUT : Doesn't retrieve in the home page DESIGNED BY : Vasu Garg DATE : 10/03/2017	ACTUAL OUTPUT : Doesn't retrieve in the home page STATUS : Pass RUN BY : Vishal Chauhan DATE : 11/03/2017

TABLE 5.2.7: TEST CASE FOR UPVOTE BUTTON

TEST CASE ID: 7	
BEFORE EXECUTION	AFTER EXECUTION
INPUT: click on upvote button of another user's question. EXPECTED OUTPUT: upvoted successfully in database and retrieved the counter of upvotes. DESIGNED BY : Vasu Garg DATE : 12/03/2017	EXPECTED OUTPUT: upvoted successfully in database and retrieved the counter of upvotes. STATUS : Pass RUN BY : Shivika Tyagi DATE : 13/03/2017

TABLE 5.2.8: TEST CASE FOR PROFILE BUTTON

TEST CASE ID: 8	
BEFORE EXECUTION	AFTER EXECUTION
INPUT: click on the user's profile button. EXPECTED OUTPUT: Redirecting to user's profile page and retrieving details of user DESIGNED BY : Vasu Garg DATE : 14/03/2017	EXPECTED OUTPUT: Redirecting to user's profile page and retrieving details of user STATUS : Pass RUN BY : Vishal Chauhan DATE : 15/03/2017

TABLE 5.2.9: TEST CASE FOR PROFILE BUTTON

TEST CASE ID: 9	
BEFORE EXECUTION	AFTER EXECUTION
INPUT: click on the other user's profile. EXPECTED OUTPUT: Redirecting to other user's profile page and retrieving details of that user DESIGNED BY : Vasu Garg DATE : 16/03/2017	EXPECTED OUTPUT: Redirecting to other user's profile page and retrieving details of that user STATUS : Pass RUN BY : Shivika Tyagi DATE : 17/03/2017

TABLE 5.2.10: TEST CASE FOR QUESTION TAB IN PROFILE

TEST CASE ID: 10	
BEFORE EXECUTION	AFTER EXECUTION
INPUT: click on the question tab of the user. EXPECTED OUTPUT: Retrieve all the questions posted by that user. DESIGNED BY : Vasu Garg DATE : 18/03/2017	EXPECTED OUTPUT: Retrieve all the questions posted by that user. STATUS : Pass RUN BY : Vishal Chauhan DATE : 21/03/2017

TABLE 5.2.11: TEST CASE FOR ANSWER TAB IN PROFILE

TEST CASE ID: 11	
BEFORE EXECUTION	AFTER EXECUTION
INPUT: click on the answer tab of the user. EXPECTED OUTPUT: Retrieve all the questions which are answered by that user. DESIGNED BY : Vasu Garg DATE : 22/03/2017	EXPECTED OUTPUT: Retrieve all the questions which are answered by that user. STATUS : Pass RUN BY : Shivika Tyagi DATE : 22/03/2017

TABLE 5.2.12: TEST CASE FOR FOLLOWING TAB IN PROFILE

TEST CASE ID: 12	
BEFORE EXECUTION	AFTER EXECUTION
INPUT: click on the following tab of the user. EXPECTED OUTPUT: Retrieve all the name of users which the current user have followed. DESIGNED BY : Vasu Garg DATE : 23/03/2017	EXPECTED OUTPUT: Retrieve all the name of users which the current user have followed. STATUS : Pass RUN BY : Neeraj Kumar DATE : 24/03/2017

TABLE 5.2.13: TEST CASE FOR FOLLOWER TAB IN PROFILE

TEST CASE ID: 13	
BEFORE EXECUTION	AFTER EXECUTION
INPUT: click on the follower tab of the user. EXPECTED OUTPUT: Retrieve all the name of users who have subscribed the current user. DESIGNED BY : Vasu Garg DATE : 25/03/2017	EXPECTED OUTPUT: Retrieve all the name of users who have subscribed the current user. STATUS : Pass RUN BY : Pragya Rajvanshi DATE : 26/03/2017

TABLE 5.2.14: TEST CASE FOR ASSIGNMENT TAB IN PROFILE

TEST CASE ID: 14	
BEFORE EXECUTION	AFTER EXECUTION
INPUT: user is student and click on assignment tab. EXPECTED OUTPUT: Retrieve all the assignments given to the department of the current user along with the submit button. DESIGNED BY : Vasu Garg DATE : 27/03/2017	EXPECTED OUTPUT: Retrieve all the assignments given to the department of the current user along with the submit button. STATUS : Pass RUN BY : Shivika Tyagi DATE : 29/03/2017

TABLE 5.2.15: TEST CASE FOR ASSIGNMENT TAB IN PROFILE

TEST CASE ID: 15	
BEFORE EXECUTION	AFTER EXECUTION
INPUT: user is teacher and click on assignment tab. EXPECTED OUTPUT: Retrieve all the assignments given to the department of the current user along with the view button. Also there is upload assignment button. DESIGNED BY : Vasu Garg DATE : 30/03/2017	EXPECTED OUTPUT: Retrieve all the assignments given to the department of the current user along with the view button. Also there is upload assignment button. STATUS : Pass RUN BY : Neeraj Kumar DATE : 31/03/2017

TABLE 5.2.16: TEST CASE FOR ASSIGNMENT SUBMISSION.

TEST CASE ID: 16	
BEFORE EXECUTION	AFTER EXECUTION
INPUT: User is teacher then click on assignment upload button enter the subject, department and file. EXPECTED OUTPUT: assignment has been successfully uploaded and retrieved. DESIGNED BY : Vasu Garg DATE : 01/04/2017	EXPECTED OUTPUT: assignment has been successfully uploaded and retrieved. STATUS : Pass RUN BY : Shivika Tyagi DATE : 03/04/2017

TABLE 5.2.17: TEST CASE FOR ASSIGNMENT SOLUTION SUBMISSION.

TEST CASE ID: 17	
BEFORE EXECUTION	AFTER EXECUTION
INPUT: click on submit button in assignment and write the remark and provide the file in pdf format. EXPECTED OUTPUT: solution has been successfully uploaded. DESIGNED BY : Vasu Garg DATE : 04/04/2017	EXPECTED OUTPUT: solution has been successfully uploaded. STATUS : Pass RUN BY : Pragya Rajvanshi DATE : 05/04/2017

TABLE 5.2.18: TEST CASE FOR VIEWING ASSIGNMENT SOLUTIONS.

TEST CASE ID: 18	
BEFORE EXECUTION	AFTER EXECUTION
INPUT: user is teacher, click on view button in assignment. EXPECTED OUTPUT: Retrieved the name and solution of all users who have uploaded the solution DESIGNED BY : Vasu Garg DATE : 06/04/2017	EXPECTED OUTPUT: Retrieved the name and solution of all users who have uploaded the solution STATUS : Pass RUN BY : Neeraj Kumar DATE : 07/04/2017

CHAPTER 6

SNAPSHOTS

The screenshot shows a web browser window with the address bar displaying 'localhost/ODF/login.php'. The page title is 'Open Discussion Forum'. Below the title, there are two tabs: 'LogIn' and 'SignUp'. The 'SignUp' tab is selected. The form contains the following fields:

- Firstname: Shivika
- Lastname: Tyagi
- E-mail: shivika.tyagi.cs.2013@mit
- Password: (masked with dots)
- Re-Password: (masked with dots)
- Phone No.: 9997014379
- Department: CSE (dropdown menu)

A 'Sign Up' button is located at the bottom of the form.

Figure 6.1 Signup Form

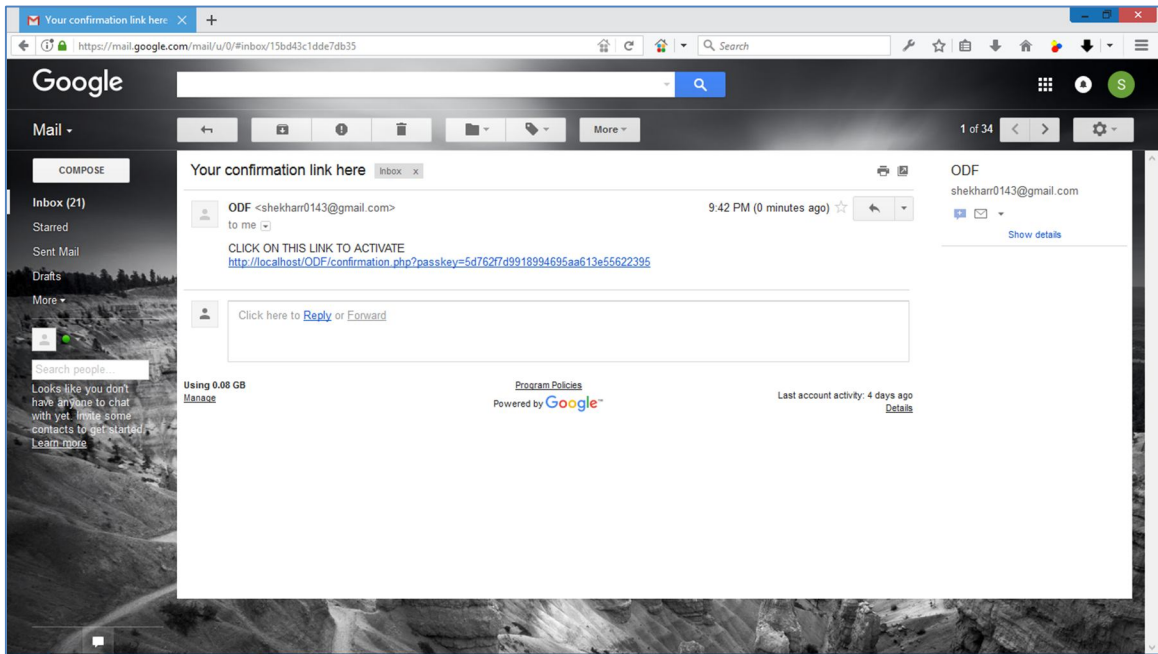


Figure 6.2 Mail Verification of User

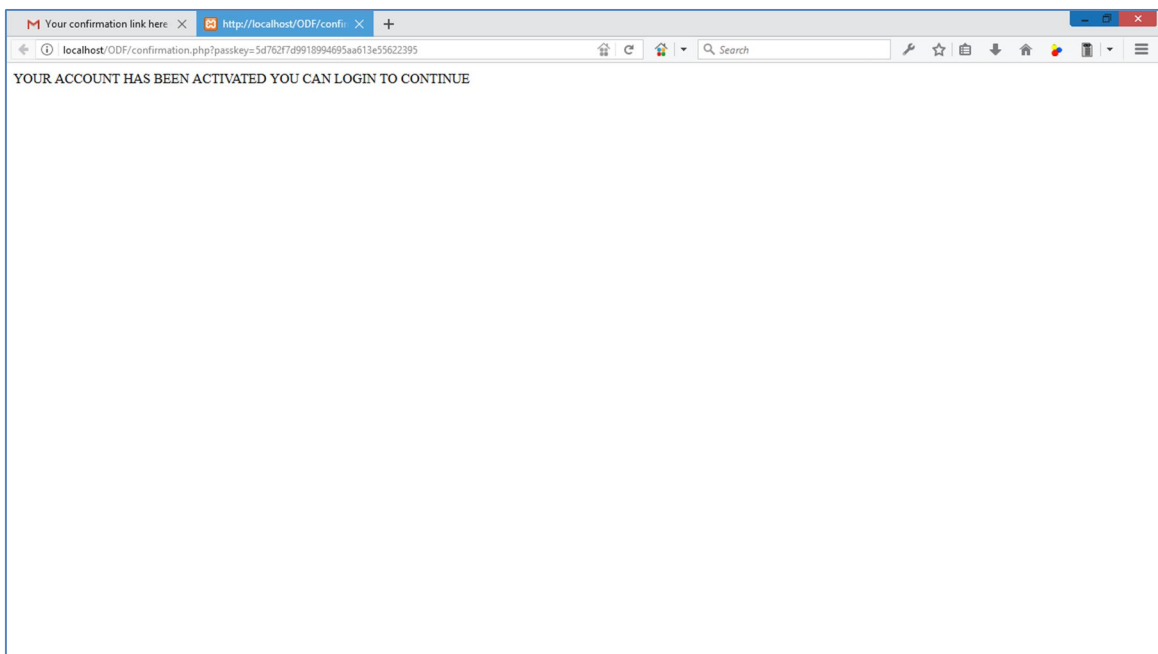


Figure 6.3 Successful Verification of User

The screenshot shows a web browser window with the address bar displaying 'localhost/ODF/login.php'. The page title is 'Open Discussion Forum'. Below the title, there are two tabs: 'LogIn' and 'SignUp'. The 'LogIn' tab is active. The login form contains the following elements:

- Email Id:
- Password:
-
- [Forget Password](#)

Figure 6.4 Login Form

The screenshot shows a web browser window with the address bar displaying 'localhost/ODF/preforgotpassword.php'. The page title is 'Forgot Password'. Below the title, there is a form titled 'CHANGE PASSWORD'. The form contains the following elements:

- E-mail ID:
-
-

Figure 6.5 Change Password Form

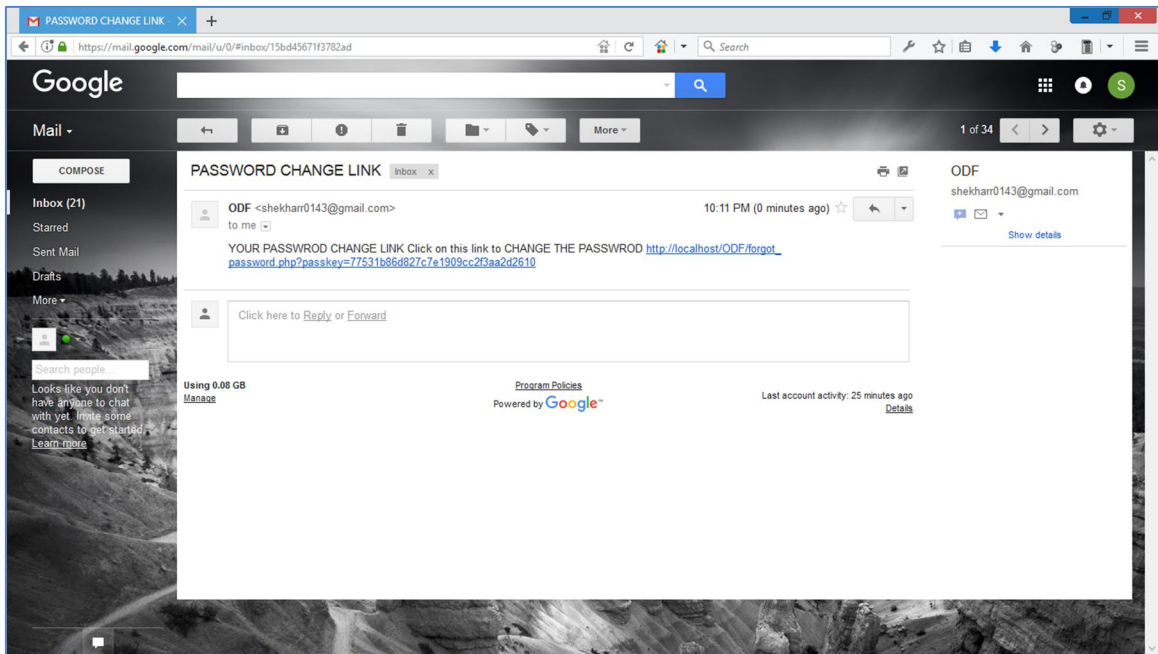


Figure 6.6 Password Change Mail Verification

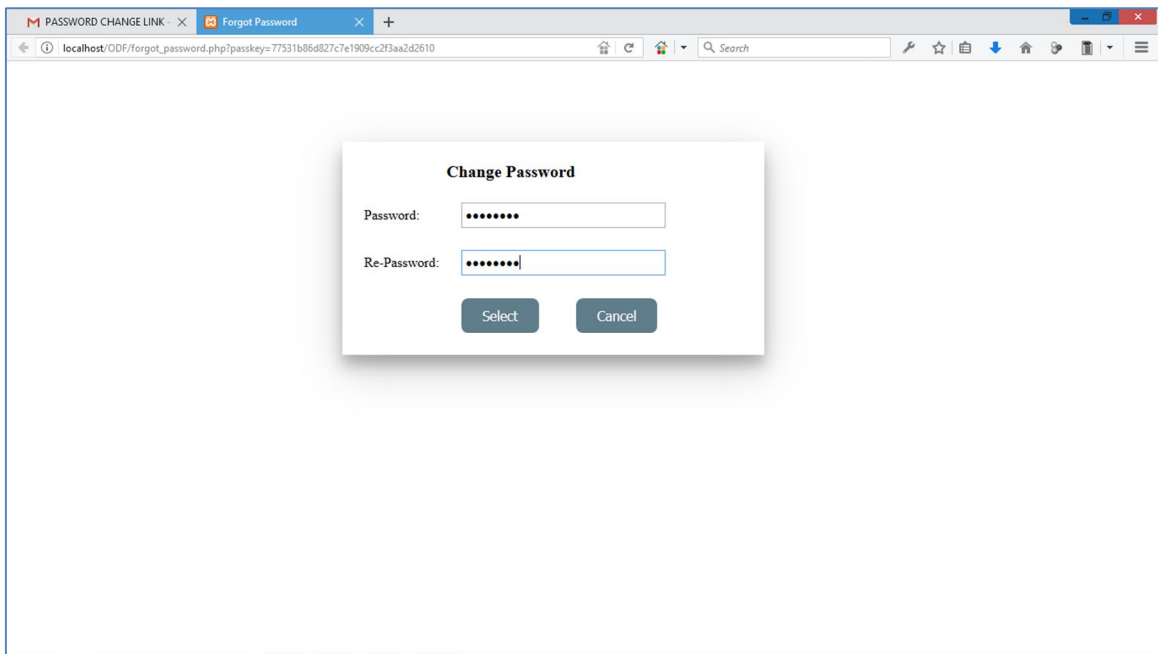


Figure 6.7 Reset Password

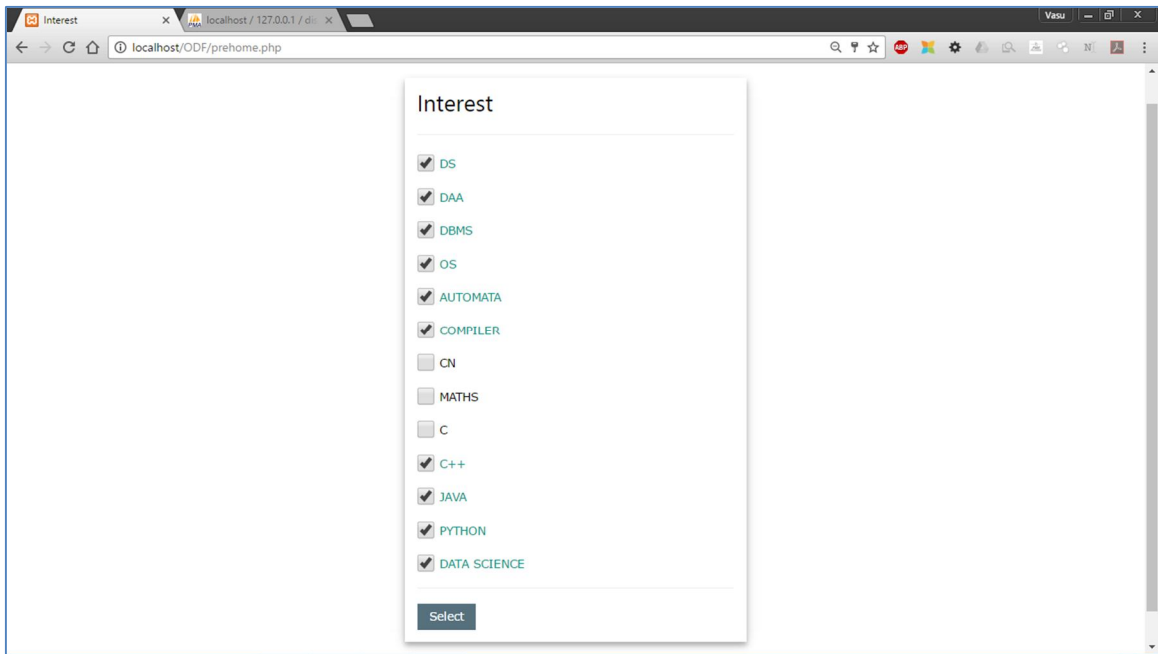


Figure 6.8 Domain Selection

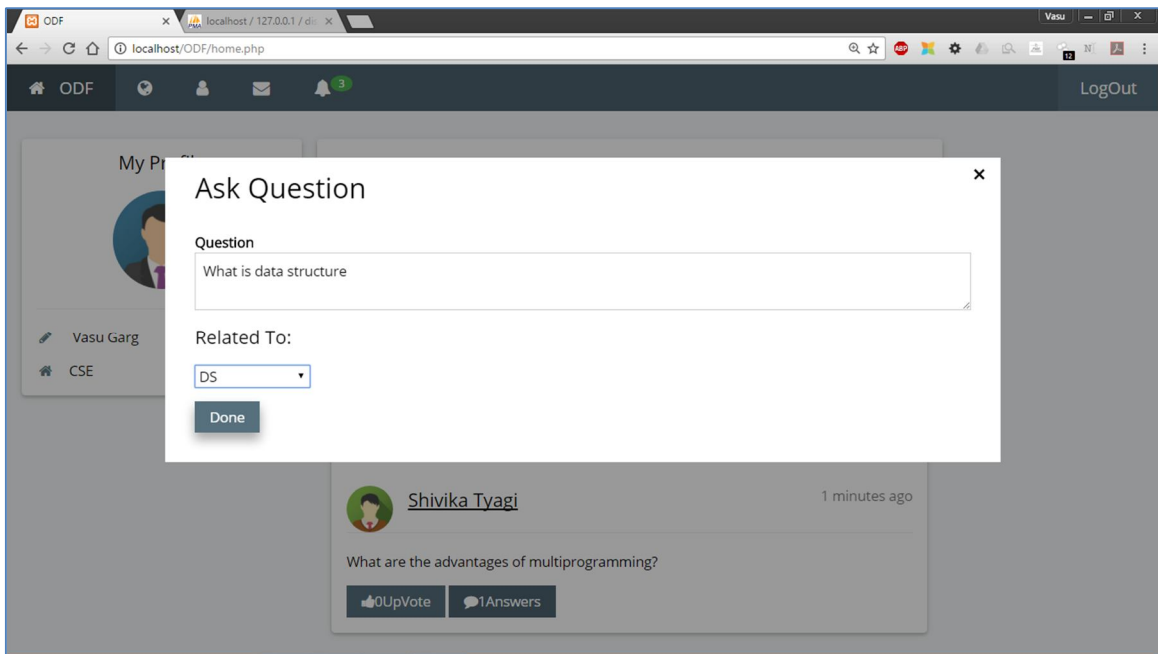


Figure 6.9 Question Post Form

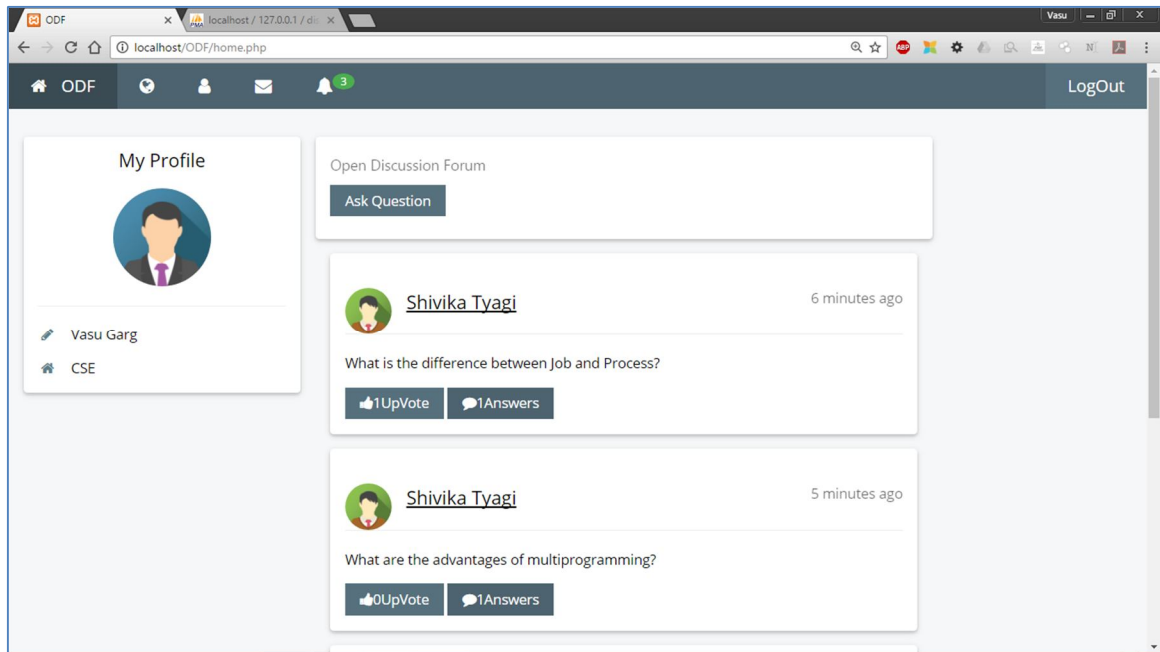
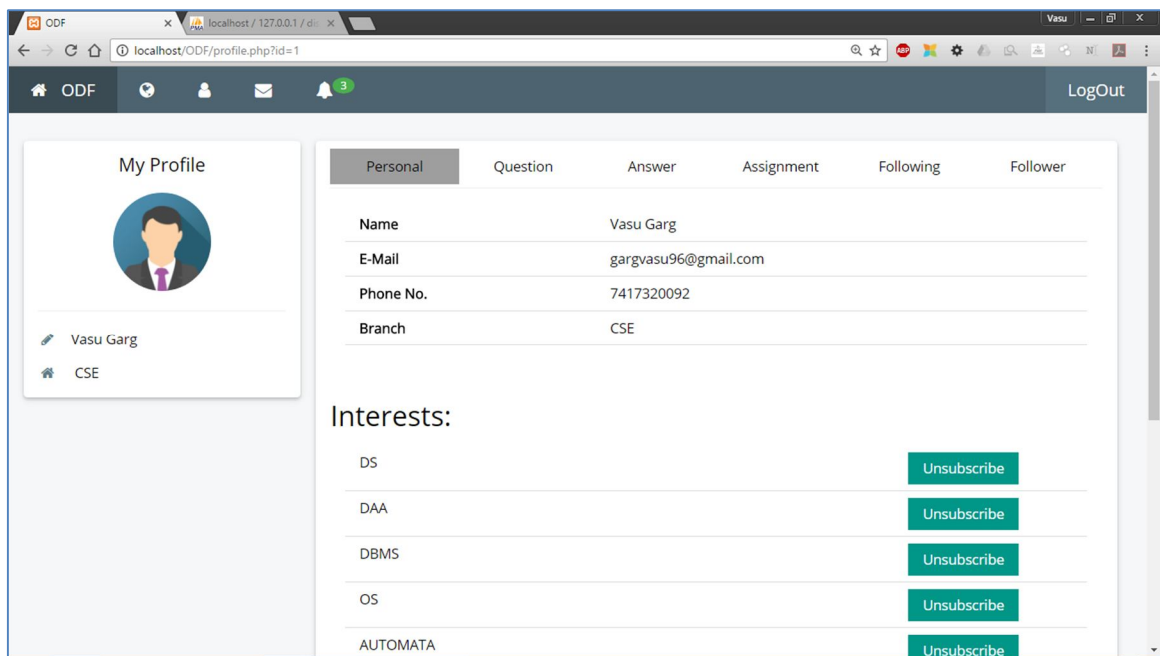


Figure 6.10 Home Page



6.11 Profile Page

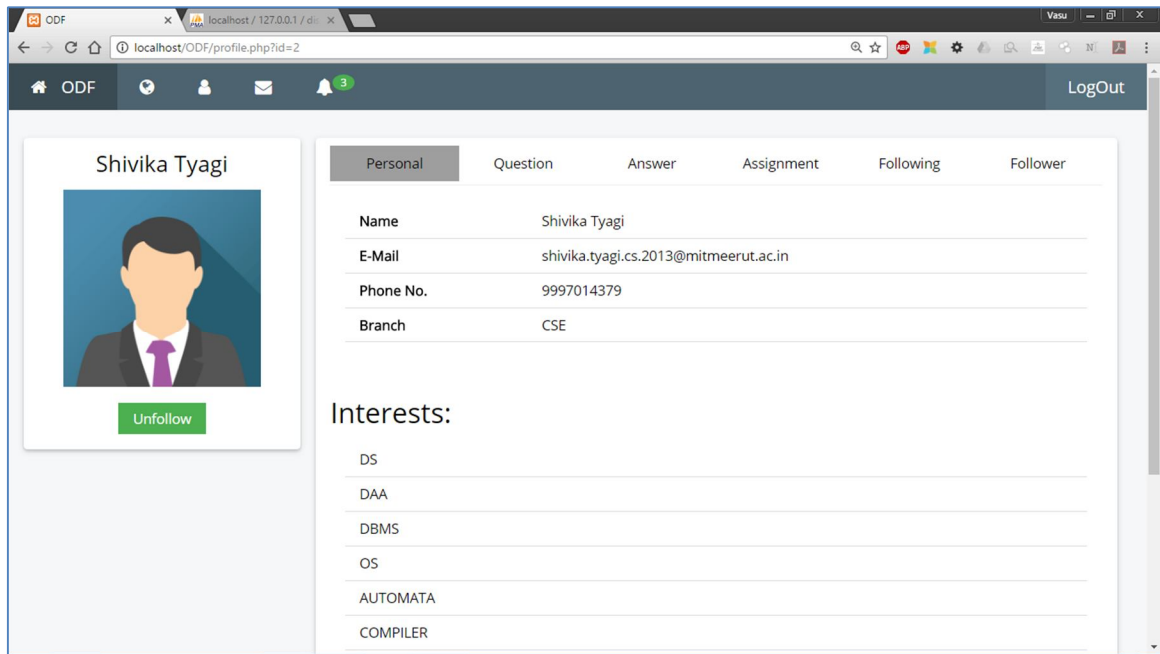


Figure 6.12 Another User Profile Page

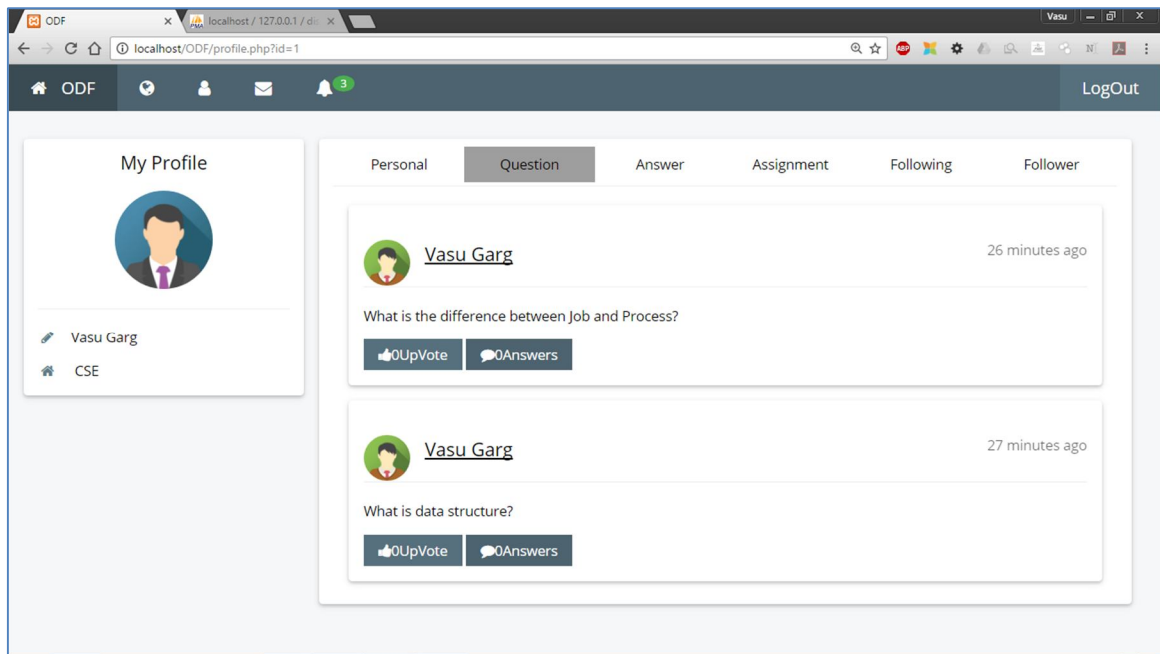


Figure 6.13 Questions Posted by User

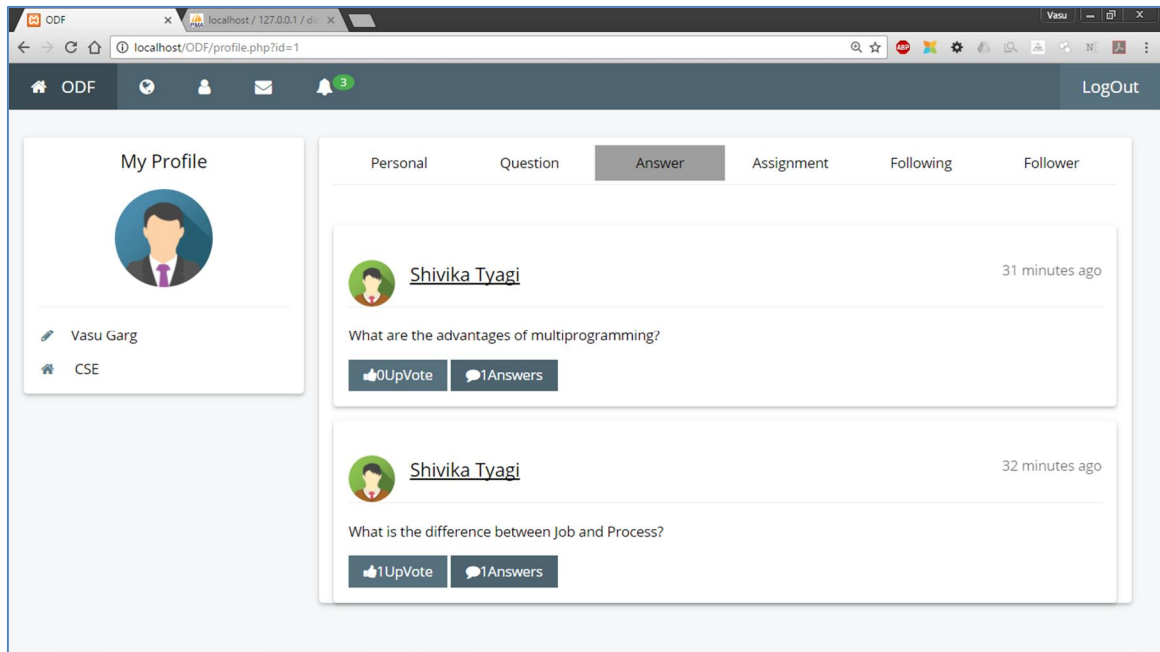


Figure 6.14 Answers Posted by User

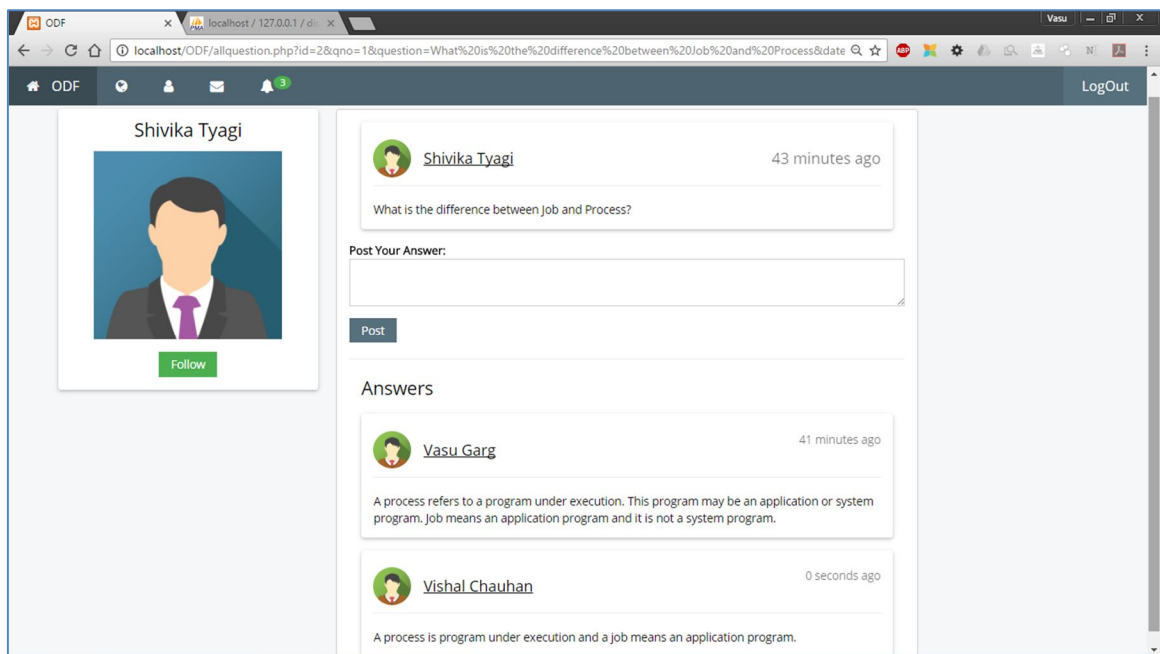


Figure 6.15 Posting Answer

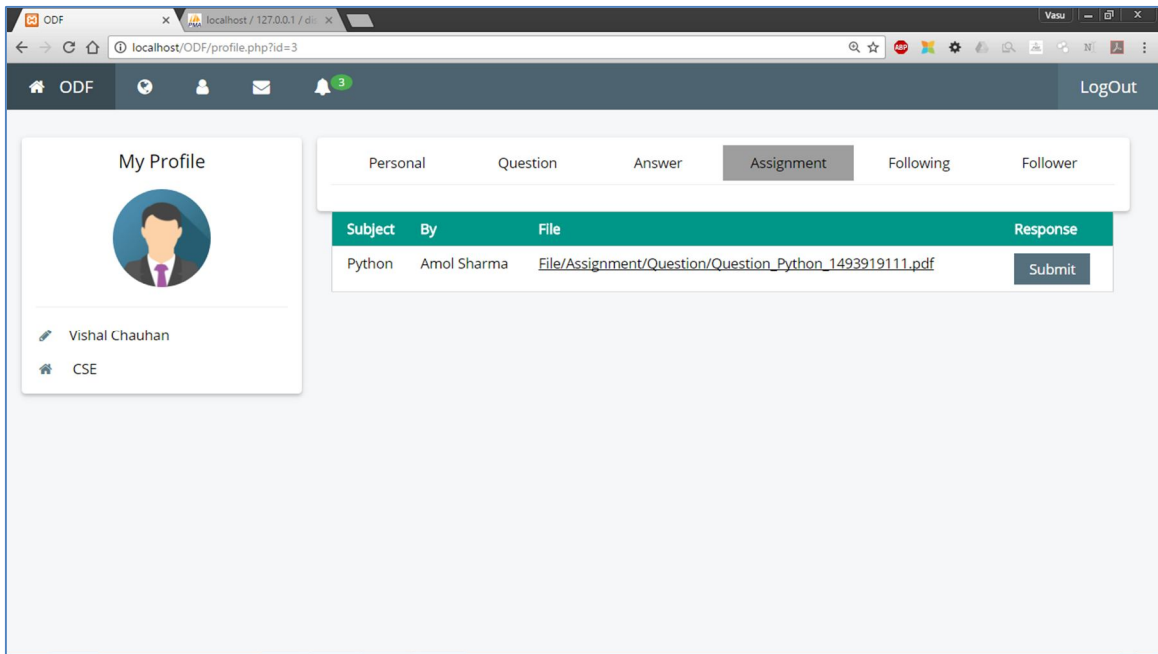


Figure 6.16 Assignment Submission View for Students

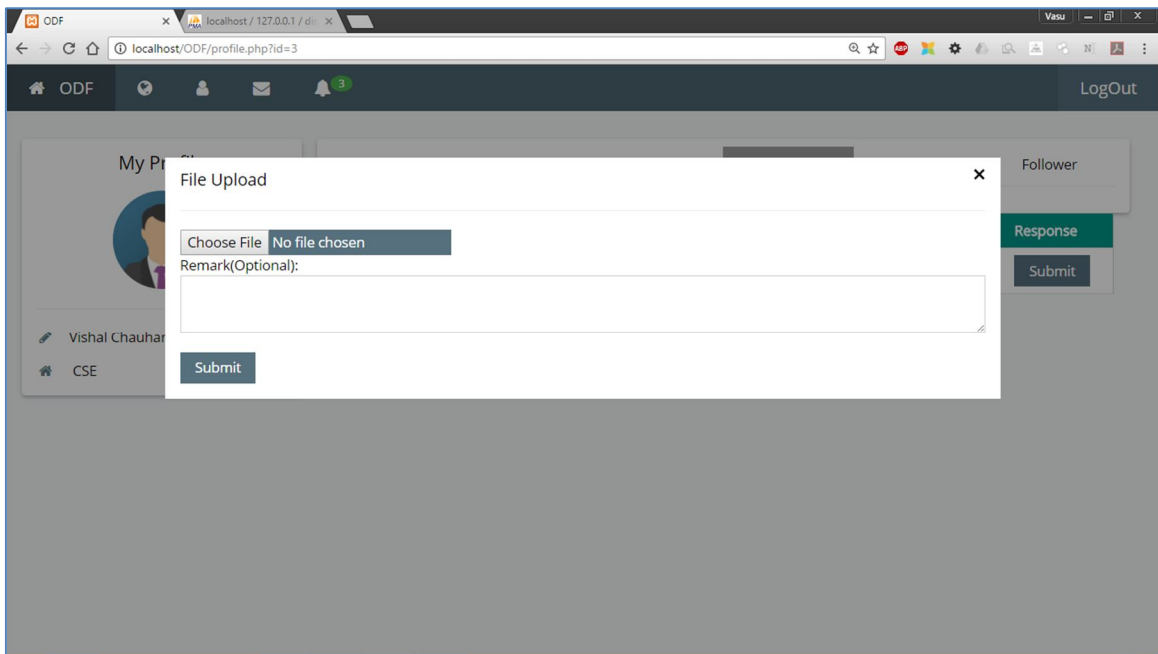


Figure 6.17 Assignment Solution Uploading

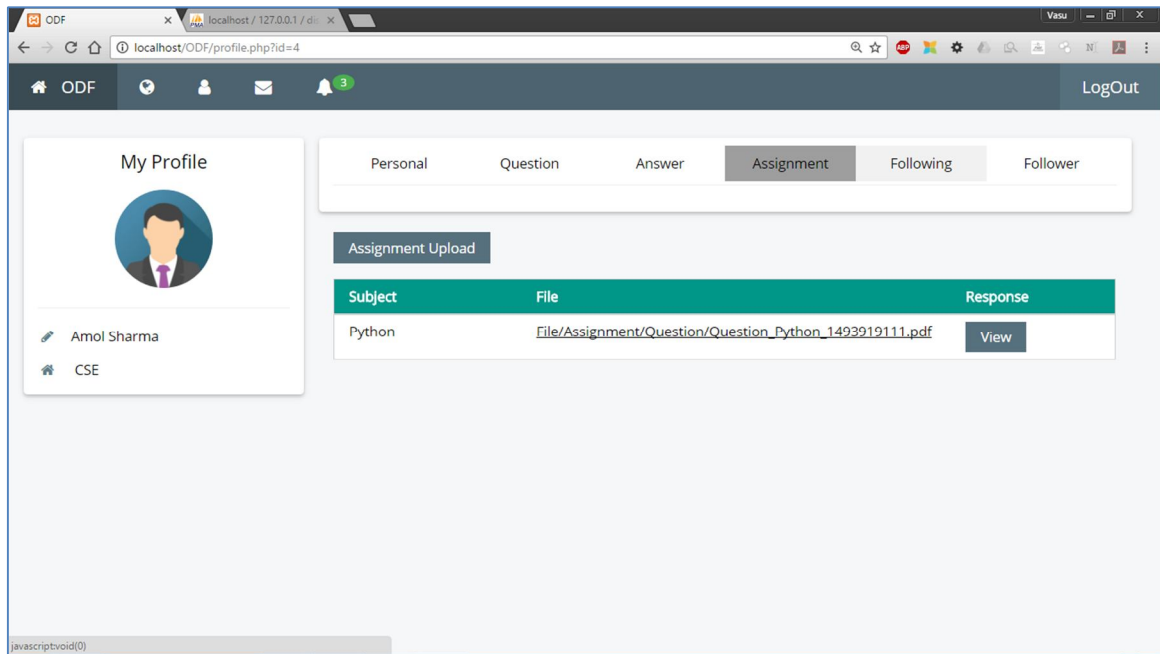


Figure 6.18 Assignment Uploading View for Teachers

CHAPTER 7

CONCLUSION

Open discussion forum (ODF) mission is to share and grow the world's knowledge. A vast amount of the knowledge that would be valuable to many people is currently only available to a few — either locked in people's heads, or only accessible to select groups. We want to connect the people who have knowledge to the people who need it, to bring together people with different perspectives so they can understand each other better, and to empower everyone to share their knowledge for the benefit of the rest of the world.

XAMPP does not play any role in actual implementation of project. It provides facility to run our project on our local system giving us insight of how the project will work when implemented in real time on server. XAMPP is a cross platform software which provides different functionalities integrated at one place such as MySQL, APACHE etc.

This system is not supposed to be implemented exclusively for one college, but developed to be use by several different colleges and schools by minor changes in the semantics.

ODF has content you will feel good about having read. It helps you understand why the world works the way it does, why people behave the way they do, and what we can all do to make the world better. It provides a personalized feed of insightful answers to questions you hadn't realized you should ask.

It helps any one to come out with strong suit, irrespective of previous low academic grade, and belonging to a low tier college. It make you to determinate to make it up to the placement, no matter what happens, since in my country, getting a prestigious MNC profile, is a programmer's living dream. It helps to achieve everything.

7.1 FUTURE SCOPE OF PROJECT

This Forum can easily support any browser. Various updates can be performed in this forum in coming future. This forum will result in minimizing the gap amongst different users. This forum gives a platform which results in an easy way of communication amongst different people from different areas

Every application has its own merits and demerits. The project has covered almost all the requirements. Further requirements and improvements can easily be done since the coding is mainly structured or modular in nature. Changing the existing modules or adding new modules can append improvements. Further enhancements can be made to the application, so that the website functions very attractive and useful manner than the present one.

Some of the ideas for improvement are

- Build a search engine for the website.
- Build a chat system for the website.
- Introduce some visualizations based on the interest in technologies. Etc.

APPENDICES

Figure A.1 Answer Table (answer)

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	id	bigint(255)			Yes	NULL		Change Drop Primary Unique Index More
2	qno	bigint(255)			Yes	NULL		Change Drop Primary Unique Index More
3	ano	bigint(255)			No	None	AUTO_INCREMENT	Change Drop Primary Unique Index More
4	answer	text	latin1_swedish_ci		No	None		Change Drop Primary Unique Index More
5	date_updated	bigint(255)			Yes	NULL		Change Drop Primary Unique Index More
6	file	varchar(255)	latin1_swedish_ci		Yes	NULL		Change Drop Primary Unique Index More

Figure A.2 Assignment Table (assignment)

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	asking_id	bigint(255)			Yes	NULL		Change Drop Primary Unique More
2	assignment_id	bigint(255)			No	None	AUTO_INCREMENT	Change Drop Primary Unique More
3	assignment	varchar(255)	latin1_swedish_ci		No	None		Change Drop Primary Unique More
4	date_updated	bigint(255)			Yes	NULL		Change Drop Primary Unique More
5	dept_name	varchar(255)	latin1_swedish_ci		Yes	NULL		Change Drop Primary Unique More
6	file	varchar(255)	latin1_swedish_ci		Yes	NULL		Change Drop Primary Unique Index More

Figure A.3 Department Table (department)

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	dept_name	varchar(255)	latin1_swedish_ci		No			Change Drop Primary Unique Index Spatial Fulltext More

Figure A.4 Domain Table (domain)

Server: 127.0.0.1 » Database: discussion_forum » Table: domain

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	domain_id	bigint(255)			No	None	AUTO_INCREMENT	Change Drop Primary Unique Index More
2	domain_name	varchar(255)	latin1_swedish_ci		No	None		Change Drop Primary Unique Index More
3	dept_name	varchar(255)	latin1_swedish_ci		Yes	NULL		Change Drop Primary Unique Index More

Check All With selected: Browse Change Drop Primary Unique Index

Figure A.5 Domain Selected by User (domain_user)

Server: 127.0.0.1 » Database: discussion_forum » Table: domain_user

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	id	bigint(255)			Yes	NULL		Change Drop Primary Unique Index Spatial Fulltext Distinct values
2	domain_id	bigint(255)			Yes	NULL		Change Drop Primary Unique Index Spatial Fulltext Distinct values

Check All With selected: Browse Change Drop Primary Unique Index

Figure A.6 Follower Table (follower)

Server: 127.0.0.1 » Database: discussion_forum » Table: follower

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	id	bigint(255)			Yes	NULL		Change Drop Primary Unique Index Spatial Fulltext Distinct values
2	follow_id	bigint(255)			Yes	NULL		Change Drop Primary Unique Index Spatial Fulltext Distinct values

Check All With selected: Browse Change Drop Primary Unique Index

Figure A.7 Notification Table (notification)

Server: 127.0.0.1 » Database: discussion_forum » Table: notification

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
<input type="checkbox"/>	1 sender_id	bigint(255)			Yes	NULL		Change Drop Primary Unique More
<input type="checkbox"/>	2 reciever_id	bigint(255)			Yes	NULL		Change Drop Primary Unique More
<input type="checkbox"/>	3 noti_id	bigint(255)			No	None	AUTO_INCREMENT	Change Drop Primary Unique More
<input type="checkbox"/>	4 noti_heading	varchar(255)	latin1_swedish_ci		Yes	NULL		Change Drop Primary Unique More
<input type="checkbox"/>	5 noti	varchar(255)	latin1_swedish_ci		Yes	NULL		Change Drop Primary Unique More
<input type="checkbox"/>	6 date_updated	bigint(255)			Yes	NULL		Change Drop Primary Unique More
<input type="checkbox"/>	7 qno	bigint(255)			Yes	NULL		Change Drop Primary Unique More
<input type="checkbox"/>	8 ano	bigint(255)			Yes	NULL		Change Drop Primary Unique More
<input type="checkbox"/>	9 assignment_id	bigint(255)			Yes	NULL		Change Drop Primary Unique More
<input type="checkbox"/>	10 solution_id	bigint(255)			Yes	NULL		Change Drop Primary Unique More
<input type="checkbox"/>	11 is_read	tinyint(1)			Yes	0		Change Drop Primary Unique More

☐ Check All With selected: Browse Change Drop Primary Unique Index

Figure A.8 Question Table (question)

Server: 127.0.0.1 » Database: discussion_forum » Table: question

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
<input type="checkbox"/>	1 id	bigint(255)			Yes	NULL		Change Drop Primary Unique Index More
<input type="checkbox"/>	2 qno	bigint(255)			No	None	AUTO_INCREMENT	Change Drop Primary Unique Index More
<input type="checkbox"/>	3 question	text	latin1_swedish_ci		No	None		Change Drop Primary Unique Index More
<input type="checkbox"/>	4 date_updated	bigint(255)			Yes	NULL		Change Drop Primary Unique Index More
<input type="checkbox"/>	5 domain_id	bigint(255)			Yes	NULL		Change Drop Primary Unique Index More
<input type="checkbox"/>	6 file	varchar(255)	latin1_swedish_ci		Yes	NULL		Change Drop Primary Unique Index More

☐ Check All With selected: Browse Change Drop Primary Unique Index

Figure A.9 Assignment Submission Table (submission)

Server: 127.0.0.1 » Database: discussion_forum » Table: submission									
#	Name	Type	Collation	Attributes	Null	Default	Extra	Action	
1	assignment_id	bigint(255)			Yes	NULL		Change	Drop
2	submitting_id	bigint(255)			Yes	NULL		Change	Drop
3	solution_id	bigint(255)			No	None	AUTO_INCREMENT	Change	Drop
4	solution	varchar(255)	latin1_swedish_ci		No	None		Change	Drop
5	date_updated	bigint(255)			Yes	NULL		Change	Drop
6	file	varchar(255)	latin1_swedish_ci		Yes	NULL		Change	Drop

☐ Check All With selected: ☐ Browse ☐ Change ☐ Drop ☐ Primary ☐ Unique ☐ Index

Figure A.10 Upvote Question Table (upvote)

Server: 127.0.0.1 » Database: discussion_forum » Table: upvote									
#	Name	Type	Collation	Attributes	Null	Default	Extra	Action	
1	id	bigint(255)			Yes	NULL		Change	Drop
2	qno	bigint(255)			Yes	NULL		Change	Drop

☐ Check All With selected: ☐ Browse ☐ Change ☐ Drop ☐ Primary ☐ Unique ☐ Index

Figure A.11 User Table (user)

Server: 127.0.0.1 » Database: discussion_forum » Table: user									
#	Name	Type	Collation	Attributes	Null	Default	Extra	Action	
1	confirm_code	varchar(255)	latin1_swedish_ci		Yes	NULL		Change	Drop
2	id	bigint(255)			No	None	AUTO_INCREMENT	Change	Drop
3	email_id	varchar(50)	latin1_swedish_ci		No	None		Change	Drop
4	password	varchar(255)	latin1_swedish_ci		No	None		Change	Drop
5	first_name	varchar(255)	latin1_swedish_ci		No	None		Change	Drop
6	last_name	varchar(255)	latin1_swedish_ci		Yes	NULL		Change	Drop
7	phone_no	bigint(10)			Yes	NULL		Change	Drop
8	dept_name	varchar(255)	latin1_swedish_ci		Yes	NULL		Change	Drop
9	teacher_student	tinyint(1)			Yes	NULL		Change	Drop
10	profile	varchar(255)	latin1_swedish_ci		Yes	NULL		Change	Drop
11	registered	tinyint(1)			Yes	0		Change	Drop

☐ Check All With selected: ☐ Browse ☐ Change ☐ Drop ☐ Primary ☐ Unique ☐ Index

REFERENCES

BOOKS

- [1] Roger.S.Pressman, “Software engineering”, Tata McGraw Hill Publications, New Delhi, Edition 2003.
- [2] Erich Gamma, Ralph Johnson, John Vlissides, Richard Helm “Design Patterns: Elements of Reusable Object-Oriented Software”.
- [3] Leon Atkinson. Prentice Hall PTR , “Core PHP Programming”.
- [4] Steve Webster, et al .friends of ED, “Advanced PHP for Flash”.

WEBSITES

- [1] <https://www.draw.io> <2 MAY, 2017>
- [2] <https://www.w3schools.com/php/default.asp> <13 APRIL, 2017>