



Jaypee University of Information Technology  
Solan (H.P.)

LEARNING RESOURCE CENTER

Acc. Num. *SP04018* Call Num:

**General Guidelines:**

- ◆ Library books should be used with great care.
- ◆ Tearing, folding, cutting of library books or making any marks on them is not permitted and shall lead to disciplinary action.
- ◆ Any defect noticed at the time of borrowing books must be brought to the library staff immediately. Otherwise the borrower may be required to replace the book by a new copy.
- ◆ The loss of LRC book(s) must be immediately brought to the notice of the Librarian in writing.

Learning Resource Centre-JUIT



SP04018



# **BLOGGING PLATFORM - A CONTENT MANAGEMENT SYSTEM**

**BY -**

**RAHUL SHARMA - 041204**

**RAKSHIT KHARE - 041270**

22.05.2008  
Mr. Ajay Kumar Singh  
Project Supervisor



**Submitted in partial fulfillment of the Degree of Bachelor of  
Technology**

**MAY – 2008**

**DEPARTMENT OF COMPUTER SCIENCE  
JAYPEE UNIVERSITY OF INFORMATION  
TECHNOLOGY-WAKNAGHAT**

**CERTIFICATE**

This is to certify that the work entitled, "BLOGGING PLATFORM- A CONTENT MANAGEMENT SYSTEM" submitted by RAHUL SHARMA (C.S.E.) and RAKSHIT KHARE (C.S.E.) in partial fulfillment for the award of degree of Bachelor of Technology of Jaypee University of Information Technology has been carried out under my supervision. This work has not been submitted partially or wholly to any other University or Institute for the award of this or any other degree or diploma.



22.05.2008


**Mr. Ajay Kumar Singh**

Project Supervisor



## ACKNOWLEDGEMENT

Many people have contributed to this project in a variety of ways over the past few months. We are sincerely thankful to Mr. Ajay Kumar Singh under whose supervision the whole project was carried out for her regular reviews and suggestions. We also acknowledge the many helpful comments received from other teachers of the different departments and visualization courses and seminars. We are deeply indebted to all those who provided reviews and suggestions for improving the materials and topics covered in our package, and we extend our apologies to anyone we may have failed to mention.



Rahul Sharma 041204



Rakshit Khare 041270



## TABLE OF CONTENTS

|   |           |
|---|-----------|
| <b>Chapter 1: BLOGGING BASICS .....</b>           | <b>1</b>  |
| 1.1 Introduction .....                            | 1         |
| 1.2 History .....                                 | 1         |
| 1.3 Types of Blogs .....                          | 7         |
| 1.4 Uses .....                                    | 10        |
| 1.5 Advantages and Disadvantages .....            | 13        |
| <b>Chapter 2: COMMON FEATURES .....</b>           | <b>15</b> |
| 2.1 The Blog Content .....                        | 15        |
| 2.2 Comments.....                                 | 15        |
| 2.3 Archives .....                                | 15        |
| 2.4 Feeds.....                                    | 15        |
| 2.5 Blog Rolls.....                               | 17        |
| 2.6 Syndication.....                              | 17        |
| 2.7 Managing Comments .....                       | 19        |
| 2.7.1 Trackbacks.....                             | 19        |
| 2.7.2 Pingbacks.....                              | 20        |
| 2.7.3 Verifying Trackbacks and Pingbacks.....     | 21        |
| 2.7.4 Comment Spam .....                          | 21        |
| 2.7.5 Comment Moderation.....                     | 21        |
| <b>Chapter 3: Project Details and Design.....</b> | <b>23</b> |
| 3.1 Problem Statement .....                       | 23        |
| 3.2 Hardware/Software Requirements .....          | 23        |
| 3.3 Database Description.....                     | 28        |
| 3.4 Design Specifications .....                   | 30        |



|   |           |
|---|-----------|
| 3.4.1 Data Flow Diagram .....                                       | 30        |
| 3.4.2 Entity Relationship Diagram .....                             | 32        |
| 3.4.3 Gantt Chart .....   | 33        |
| 3.4.4 Pert Chart.....   | 34        |
| <b>Chapter 4: Modules Implemented.....</b>                          | <b>36</b> |
| 4.1 Public Zone .....   | 36        |
| 4.1.1 Database Connectivity.....                                    | 36        |
| 4.1.2 Home Page.....  | 36        |
| 4.1.3 Blog Roll .....   | 37        |
| 4.1.4 Contact Us .....  | 37        |
| 4.1.5 Search .....  | 37        |
| 4.1.6 Sign In .....   | 37        |
| 4.1.7 Archives.....   | 38        |
| 4.1.8 Comments.....   | 38        |
| 4.1.9 Spam Protection.....  | 39        |
| 4.2.1 Phishing Filter .....   | 39        |
| 4.2 Administrative Zone.....  | 39        |
| 4.2.1 Database Connectivity.....                                    | 39        |
| 4.2.2 Administrator Home.....                                       | 40        |
| 4.2.3 Add Post .....  | 40        |
| 4.2.4 Admin Blog Roll .....   | 40        |
| 4.2.5 Settings .....  | 41        |
| 4.2.6 Encryption and Decryption of Asp.Net Configuration File ..... | 41        |
| 4.2.7 Comments Moderation .....                                     | 41        |
| 4.2.8 Delete Post.....  | 42        |



|  |           |
|--|-----------|
| <b>Chapter 5: Conclusion and Future Scope.....</b> | <b>43</b> |
| <b>Bibliography .....</b>                          | <b>44</b> |
| <b>Appendix.....</b>                               | <b>45</b> |
| Appendix A: Snapshots .....                        | 45        |
| Appendix B: Installation Guide.....                | 50        |
| Appendix C: Source Code .....                      | 69        |



**LIST OF FIGURES**

|   |    |
|---|----|
| Figure 1: Data Flow Diagram Level 0 .....   | 30 |
| Figure 2: Data Flow Diagram Level 1 .....   | 31 |
| Figure 3: Entity Relationship Diagram ..... | 32 |
| Figure 4: Gantt Chart .....                 | 33 |
| Figure 5: Pert Chart .....                  | 34 |

## LIST OF BLOG RELATED TERMS

### (a) B

#### **Blog client**

(Weblog client) is software to manage (post, edit) blogs from operating system with no need to launch a web browser. A typical blog client has an editor, a spell-checker and a few more options that simplify content creation and editing.

#### **Blogger**

Person who runs a blog. Also blogger.com, a popular blog hosting web site. Rarely: weblogger.

#### **Bloggies**

One of the most popular blog awards.

#### **Blog feed**

The XML-based file in which the blog hosting software places a machine-readable version of the blog so that it may be "syndicated" for further distribution on the web. Formats such as RSS and Atom are used to structure the XML file.

#### **Blogosphere**

All blogs or the blogging community. Also called blogistan or, more rarely, blogspace.

#### **Blogroll**

A list of blogs. A blogger features a list of their favorite blogs in the sidebar of their blog.

#### **Blog site**

The web location (URL) of a blog, which may be either a dedicated domain, a sub-domain, or embedded within a web site.

#### **Blogstorm**

When a large amount of activity, information and opinion erupts around a particular subject or controversy in the blogosphere, it is sometimes called a **blogstorm** or **blog swarm**.

### (b) C

#### **Categories**

This is a method of organizing blog entries by assigning each entry to a predetermined topic. Each topic (category) will link to a list of entries, all with related content.

#### **Collaborative blog**

A blog (usually focused on a single issue or political stripe) on which multiple users enjoy posting permission. Also known as **group blog**.

#### **Comment spam**



Like e-mail spam. Robot “spambots” flood a blog with advertising in the form of bogus comments. A serious problem that requires bloggers and blog platforms to have tools to exclude some users or ban some addresses in comments.

**(c) D**

**Dark Blog**

A non-public blog (e.g. behind a firewall)

**Desktop Blogging Client**

An off-line blog management (posting, editing and archiving) tool

**(d) F**

**Feeds**

RSS Feeds

**(e) H**

**Hits**

Number of users visited is often referred as hits. (website / blog hits)

**(f) I**

**Instalanche**

Sudden and possibly overwhelming increase in traffic to a site after being linked to by the Instapundit

**(g) L**

**Link Love**

linking to a site or blog, usually unsolicited, that you like, enjoy, or find useful.

**Log in, blog to, log out**

a catchphrase referring to blogger style of activity.

**Linkroll**

A list of recently-bookmarked links with brief descriptions, in the sidebar of a blog.

**(h) M**

**Multi-blog**

Creating, maintaining, and running multiple blogs (2 or more) simultaneously.

**Multi-blogger**

An individual, business, or institution that runs multiple blogs.

**(i) P**

**Permalink**

The unique URL of a single post. Use this when you want to link to a post somewhere.

**Ping**

The alert in the TrackBack system that notifies the original poster of a blog post when someone else writes an entry concerning the original post.

**Podcasting**

Contraction of "iPod" and "broadcasting" (but not for iPods only). Posting audio and video material on a blog and its RSS feed, for digital players.

### **Post**

An entry written and published to a blog.

## **(j) R**

### **RSS**

Really Simple Syndication is a family of Web feed formats used to publish frequently updated content such as blog entries, news headlines or podcasts.

### **RSS aggregator**

Software or online service allowing a blogger to read an RSS feed, especially the latest posts on their favourite blogs. Also called a reader, or feedreader.

### **RSS feed**

The file containing a blog's latest posts. It is read by an RSS aggregator/reader and shows at once when a blog has been updated. It may contain only the title of the post, the title plus the first few lines of a post, or the entire post.

## **(k) S**

### **Spam blog**

A blog which is composed of spam. A

Spam blog or "any blog whose creator doesn't add any written value."

### **Subscribe**

The term used when a blogs feed is added to a feed reader like Bloglines or Google. Some blogging platforms have internal subscriptions; this allows readers to receive notification when there are new posts in a blog.

## **(l) T**

### **Template**

Templates, used on the "back end" of a blog that work together to handle information and present it on a blog.

### **Theme**

CSS based code that when applied to the templates will result in visual element changes to the blog. The theme, as a whole, is also referred to as a blog design.

### **TrackBack**

A system that allows a blogger to see who has seen the original post and has written another entry concerning it. The system works by sending a 'ping' between the blogs, and therefore providing the alert.



## ABSTRACT

Blogging have become increasingly popular these days. In fact, it has become a source of income for some while on the other hand a common place to share thoughts with like minded people. Blogging Platform is the base on which various blogs runs, providing the functionality in an efficient way. It can provide the user an effective way to provide any kind of information over the internet. In our project we have analyzed various different types of Blog features and tried to implement those by our own in Asp.Net and C#. This report focuses on the Content Management System also known as "blogging platform" which enables administrator to monitor and manage posts and various other important aspects of a blog.

## CHAPTER 1

# BLOGGING BASICS

### 1.1 Introduction

"Blog" is an abbreviated version of "weblog," which is a term used to describe web sites that maintain an ongoing chronicle of information. A blog is a frequently updated, personal website featuring diary-type commentary and links to articles or other Web sites. Blogs range from the personal to the political, and can focus on one narrow subject or whole range of subjects. Many blogs focus on a particular topic, such as web design, politics, sports, or mobile technology. Some are more eclectic, presenting links to all sites. Generally speaking, blogs tend to have a few things in common:

- A main content area with articles listed chronologically, newest on top. A way for people to leave comments about the articles.
- A list of links to other related sites, sometimes called a "blogroll".
- One or more "feeds" like RSS, Atom or RDF files.

Most blogs are primarily textual, although some focus on art (artlog), photographs (photoblog), sketchblog, videos (vlog), music (MP3 blog), audio (podcasting) and are part of a wider network of social media. Micro-blogging is another type of blogging which consists of blogs with very short posts.

### 1.2 History

Before blogging became popular, digital communities took many forms, including Usenet, commercial online services such as GENie, BiX and the early CompuServe, e-mail lists and Bulletin Board Systems (BBS). In the 1990s, Internet forum software, such as WebEx, created running conversations with "threads". Threads are topical connections between messages on a metaphorical "corkboard". Some have likened blogging to the Mass-Observation project of the mid-20th century.



*(a) 1983-1990 (Pre-HTTP)*

Usenet was the primary serial medium included in the original definition of the World Wide Web. It featured the Moderated Newsgroup which allowed all posting in a newsgroup to be under the control of an individual or small group. Most such newsgroups were simply moderated discussion forums, however, in 1983-84, one exception, named mod.ber, was created, named after and managed by an individual: Brian E. Redman. Regularly, Redman and a few associates posted summaries of interesting postings and threads taking place elsewhere on the net. With its serial journal publishing style, presence on the pre-HTTP web and strong similarity to the common blog form which features links to interesting and cool places on the net chosen by the blogger, mod.ber had many of the characteristics commonly associated with the term "blog". It ceased operation after approximately 8 months. Brad Templeton calls the newsgroup rec.humor.funny (which he founded) the world's oldest still existing blog.

*(b) 1994-2001*

The modern blog evolved from the online diary, where people would keep a running account of their personal lives. Most such writers called themselves diarists, journalists, or journalers. A few called themselves "escribitionists". The Open Pages webring included members of the online-journal community. Justin Hall, who began eleven years of personal blogging in 1994 while a student at Swarthmore College, is generally recognized as one of the earliest bloggers, as is Jerry Pournelle. Another early blog was Wearable Wireless Webcam, an online shared diary of a person's personal life combining text, video, and pictures transmitted live from a wearable computer and EyeTap device to a web site in 1994. This practice of semi-automated blogging with live video together with text was referred to as sousveillance, and such journals were also used as evidence. Other forms of journals kept online also existed. A notable example was game programmer John Carmack's widely read journal, published via the finger protocol. Some of the very earliest bloggers, like Steve Gibson of sCary's Quakeholio (now Shacknews) and Stephen Heaslip of Blue's News (still running since 1995 with online archives back to July 1996), evolved from the Quake scene and Carmack's plan updates. Steve Gibson was hired to blog full-time by Ritual Entertainment on February 8, 1997. Websites, including both corporate sites and personal homepages, had and still often have "What's New" or "News" sections, often on the index page and sorted by date. One example of a news based "weblog" is the Drudge Report

founded by the self-styled maverick reporter Matt Drudge, though apparently Drudge dislikes this classification. Another is the Institute for Public Accuracy which began posting news releases featuring several news-pegged one-paragraph quotes several times a week beginning in 1998. One noteworthy early precursor to a blog was the tongue-in-cheek personal website that was frequently updated by Usenet legend Kibo.

Early weblogs were simply manually updated components of common websites. However, the evolution of tools to facilitate the production and maintenance of web articles posted in reverse chronological order made the publishing process feasible to a much larger, less technical, population. Ultimately, this resulted in the distinct class of online publishing that produces blogs we recognize today. For instance, the use of some sort of browser-based software is now a typical aspect of "blogging". Blogs can be hosted by dedicated blog hosting services, or they can be run using blog software, such as WordPress, Movable Type, blogger or LiveJournal, or on regular web hosting services.

The term "weblog" was coined by Jorn Barger on 17 December 1997. The short form, "blog," was coined by Peter Merholz, who jokingly broke the word *weblog* into the phrase *we blog* in the sidebar of his blog Peterme.com in April or May of 1999. This was quickly adopted as both a noun and verb ("to blog," meaning "to edit one's weblog or to post to one's weblog").

After a slow start, blogging rapidly gained in popularity. Blog usage spread during 1999 and the years following, being further popularized by the near-simultaneous arrival of the first hosted blog tools:

- Open Diary launched in October 1998, soon growing to thousands of online diaries. Open Diary innovated the reader comment, becoming the first blog community where readers could add comments to other writers' blog entries.
- Brad Fitzpatrick, a well known blogger started LiveJournal in March 1999.
- Andrew Smales created Pitas.com in July 1999 as an easier alternative to maintaining a "news page" on a website, followed by Diaryland in September 1999, focusing more on a personal diary community.
- Evan Williams and Meg Hourihan (Pyra Labs) launched blogger.com in August 1999 (purchased by Google in February 2003)



Blogging combined the personal web page with tools to make linking to other pages easier — specifically permalinks, blogrolls and TrackBacks. This, together with weblog search engines enabled bloggers to track the threads that connected them to others with similar interests.

(c) 2001–2004

Several broadly popular American blogs emerged in 2001: Andrew Sullivan's AndrewSullivan.com, Ron Gunzburger's Politics1.com, Taegan Goddard's Political Wire, Glenn Reynolds' Instapundit, Charles Johnson's Little Green Footballs, and Jerome Armstrong's MyDD — all blogging primarily on politics (two earlier popular American political blogs were Bob Somerby's Daily Howler launched in 1998 and Mickey Kaus' Kausfiles launched in 1999).

By 2001, blogging was enough of a phenomenon that how-to manuals began to appear, primarily focusing on technique. The importance of the blogging community (and its relationship to larger society) increased rapidly. Established schools of journalism began researching blogging and noting the differences between journalism and blogging.

Also in 2002, many blogs focused on comments by U.S. Senate Majority Leader Trent Lott. Senator Lott, at a party honoring U.S. Senator Strom Thurmond, praised Senator Thurmond by suggesting that the United States would have been better off had Thurmond been elected president. Lott's critics saw these comments as a tacit approval of racial segregation, a policy advocated by Thurmond's 1948 presidential campaign. This view was reinforced by documents and recorded interviews dug up by bloggers. (See Josh Marshall's *Talking Points Memo*.) Though Lott's comments were made at a public event attended by the media, no major media organizations reported on his controversial comments until after blogs broke the story. Blogging helped to create a political crisis that forced Lott to step down as majority leader.

The impact of this story gave greater credibility to blogs as a medium of news dissemination. Though often seen as partisan gossips, bloggers sometimes lead the way in bringing key information to public light, with mainstream media having to follow their lead. More often, however, news blogs tend to react to material already published by the mainstream media.



Since 2002, blogs have gained increasing notice and coverage for their role in breaking, shaping, and spinning news stories. The Iraq war saw bloggers taking measured and passionate points of view that go beyond the traditional left-right divide of the spectrum. Blogging by established politicians and political candidates, to express opinions on war and other issues, cemented blogs' role as a news source. (See Howard Dean and Wesley Clark.) Meanwhile, an increasing number of experts blogged, making blogs a source of in-depth analysis. (See Daniel Drezner and J. Bradford DeLong.)

The second Iraq war was the first "blog war" in another way: Iraqi bloggers gained wide readership, and one, Salam Pax, published a book of his blog. Blogs were also created by soldiers serving in the Iraq war. Such "warblogs" gave readers new perspectives on the realities of war, as well as often offering different viewpoints from those of official news sources. Blogging was used to draw attention to obscure news sources. For example, bloggers posted links to traffic cameras in Madrid as a huge anti-terrorism demonstration filled the streets in the wake of the March 11 attacks. Bloggers began to provide nearly-instant commentary on televised events, creating a secondary meaning of the word "blogging": to simultaneously transcribe and editorialize speeches and events shown on television. (For example, "I am blogging Rice's testimony" means "I am posting my reactions to Condoleezza Rice's testimony into my blog as I watch her on television.") Real-time commentary is sometimes referred to as "liveblogging."

#### **(d) 2004-PRESENT**

In 2004, the role of blogs became increasingly mainstream, as political consultants, news services and candidates began using them as tools for outreach and opinion forming. Even politicians not actively campaigning, such as the UK's Labour Party's MP Tom Watson, began to blog to bond with constituents.

Minnesota Public Radio broadcast a program by Christopher Lydon and Matt Stoller called "The blogging of the President," which covered a transformation in politics that blogging seemed to presage. The *Columbia Journalism Review* began regular coverage of blogs and blogging. Anthologies of blog pieces reached print, and blogging personalities began appearing on radio and television. In the summer of 2004, both United States Democratic and Republican Parties' conventions credentialed bloggers, and blogs became a standard part of the publicity arsenal.



Mainstream television programs, such as Chris Matthews' *Hardball*, formed their own blogs. Merriam-Webster's Dictionary declared "blog" as the word of the year in 2004.

In 2004, Global Voices Online, a site which "aggregates, curates, and amplifies the global conversation online – shining light on places and people other media often ignore" surfaced, bringing to light bloggers from around the world. Today, the site has a relationship with Reuters and is responsible for breaking many global news stories.

Blogs were among the driving forces behind the "Rathergate" scandal, to wit: (television journalist) Dan Rather presented documents (on the CBS show *60 Minutes*) that conflicted with accepted accounts of President Bush's military service record. Bloggers declared the documents to be forgeries and presented evidence and arguments in support of that view, and CBS apologized for what it said were inadequate reporting techniques (see Little Green Footballs). Many bloggers view this scandal as the advent of blogs' acceptance by the mass media, both as a news source and opinion and as means of applying political pressure.

Some bloggers have moved over to other media. The following bloggers (and others) have appeared on radio and television: Duncan Black (known widely by his pseudonym, Atrios), Glenn Reynolds (Instapundit), Markos Moulitsas Zúniga (Daily Kos), Alex Steffen (Worldchanging) and Ana Marie Cox (Wonkette). In counter-point, Hugh Hewitt exemplifies a mass media personality who has moved in the other direction, adding to his reach in "old media" by being an influential blogger.

Some blogs were an important news source during the December 2004 Tsunami such as Medecins Sans Frontieres, which used SMS text messaging to report from affected areas in Sri Lanka and Southern India. Similarly, during Hurricane Katrina in August 2005 and the aftermath a few blogs which were located in New Orleans, including the Interdictor and Gulfsails were able to maintain power and an Internet connection and disseminate information that was not covered by the Main Stream Media.

In the United Kingdom, *The Guardian* newspaper launched a redesign in September 2005, which included a daily digest of blogs on page 2. Also in June 2006, BBC News launched a weblog for its editors, following other news companies.



In January 2005, *Fortune* magazine listed eight bloggers that business people "could not ignore": Peter Rojas, Xeni Jardin, Ben Trott, Mena Trott, Jonathan Schwartz, Jason Goldman, Robert Scoble, and Jason Calacanis.

In 2007, Tim O'Reilly proposed a Blogger's Code of Conduct.

### **1.3 Types of Blogs**

There are various types of blogs, and each differs in the way content is delivered or written.

#### **By media type**

A blog comprising videos is called a vlog, one comprising links is called a linklog, a site containing a portfolio of sketches is called a sketchblog or one comprising photos is called a photoblog. Blogs with shorter posts and mixed media types are called tumblelogs.

An Artlog is a form of art sharing and publishing in the format of a blog, but differentiated by the predominant use of and focus on Art work rather than text.

A rare type of blog hosted on the Gopher Protocol is known as a Phlog

#### **By device**

Blogs can also be defined by which type of device is used to compose it. A blog written by a mobile device like a mobile phone or PDA is called a moblog.

#### **Legal status of publishers**

A blog can be private, as in most cases, or it can be for business purposes. Blogs, either used internally to enhance the communication and culture in a corporation or externally for marketing, branding or PR purposes are called corporate blogs.

#### **Blog search engines**

Several blog search engines are used to search blog contents (also known as the blogosphere), such as blogdigger, Feedster, and Technorati. Technorati provides current information on both popular searches and tags used to categorize blog postings.



### **Blogging Communities and Directories**

Several online communities exist that connect people to blogs and bloggers to other bloggers, including Blog Catalogue and MyBlogLog. A collection of local blogs is sometimes referred to as a Bloghood.

Broadly it can be divided into two more types

1. SOCIAL BLOGS
2. BUSSINESS BLOGS

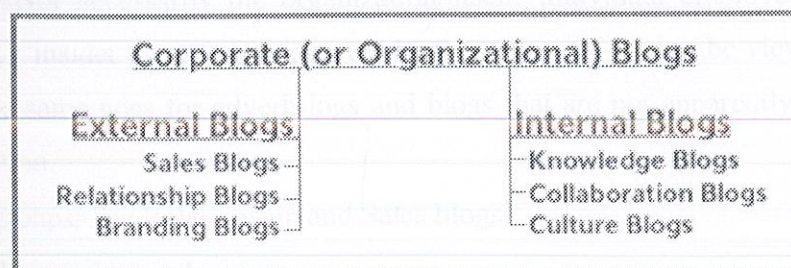
#### **1. SOCIAL BLOGS**

Some blogs focus on a particular subject, such as political blogs, travel blogs, fashion blogs, project blogs, niche blogs, legal blogs (often referred to as a blawgs) or dreamlogs. While not a legitimate type of blog, one used for the sole purpose of spamming is known as a Splog. A Slog (Site or website **log**) is a section or 'slice' of a regular business website, which is seamlessly integrated within the regular website structure but is produced with blogging software.

#### **2. BUSSINESS BLOGS**

Corporate (or Business, Organizational) blogs can be classified into six different categories. Each category shares common characteristics and the blog content can be expected to differ between the categories. There's a difference in terms of target groups and purpose.

Even if we also see hybrid forms where blogs are examples of more than one category, an organization that professionally incorporates blogging into its communications strategy will likely prioritize one purpose (for each blog).



At least, that's what I think and suggests. A recent post and discussion got me going on this. Why we need a classification at all, you might ask. One situation I have in mind is when a corporate communications manager seriously wants to investigate the potential benefits of blogging. Doing that, she could stumble upon so hugely different blogs that nothing would make any sense. But with the help of a filter of some kind - a classification - research and discussion would be much easier.

#### ***Sales blogs - external***

- Purpose: To market or sell products/services, make citizens aware of public services, get donations for charities or political parties etc.
- Blogger: The organization itself, or more specifically individuals within it writing on behalf of the organization.
- Target groups: (Potential) clients/customers and citizens, that is persons and/or other organizations that are directly involved with the publishing organization, or those it wants to reach with the purpose above.

#### ***Relationship blogs - external***

- Purpose: To create, uphold or strengthen relationships.
- Blogger: The organization itself, or more specifically individuals within it writing on behalf of the organization.
- Target groups: Often smaller and more specified than with sales blogs. Examples are support blogs aimed at customers of a certain product, finance blogs for IR purposes, PR blogs for media, blogs trying to reach students, future employees, politicians/officials.

#### ***Branding blogs - external***

- Purpose: To strengthen the brand, the profile, of the publishing organization or individuals within it.
- Blogger: Not necessarily the organization itself. Individual employees' or executives' blogging ("insider blogs"), if supported by the organization, can be viewed as a branding blog. The same goes for adverb logs and blogs that are not apparently connected to the organization.
- Target groups: See Relationship and Sales blogs.

#### ***Knowledge blogs - internal***



- Purpose: To give employees information and insights relating to their work assignments; news, business intelligence, reports about ongoing projects etc.
- Blogger: The organization itself through one or more designated bloggers, or potentially all employees through an entirely open blog publishing model.
- Target groups: All employees with a certain interest.

#### ***Collaboration blogs - internal***

- Purpose: To provide a working team with a tool for research, collaboration and discussion.
- Blogger: The team.
- Targetgroup: The team (there's only a fine line - if any - between this blog category and business wiki's. You could also see for example Relationship Blogs change into more wiki like publications)

#### ***Culture blogs - internal***

- Purpose: To strengthen organizational culture. Typically through informal content of social or non-work related character.
- Blogger: The organization itself through one or more designated bloggers, or potentially all employees through an entirely open blog publishing model.
- Target groups: All employees as employees, and not as professionals (developers, managers, assistants and so on).

### **1.4 Uses**

#### **1.4.1 USE IN EDUCATION**

**Five major uses for blogs** in education:

1. Teachers use blogs to replace the standard class Web page. Instructors post class times and rules, assignment notifications, suggested readings, and exercises. Aside from the ordering of material by date, students would find nothing unusual in this use of the blog. The instructor,

however, finds that the use of blogging software makes this previously odious chore much simpler.

2. Instructors begin to link to Internet items that relate to their course. Mesa Community College's Rick Effland, for example, maintains a blog to pass along links and comments about topics in archaeology.<sup>15</sup> Though Mesa's archaeology Web pages have been around since 1995, blogging allows Effland to write what are in essence short essays directed specifically toward his students. Effland's entries are not mere annotations of interesting links. They effectively model his approach and interest in archaeology for his students.

3. Blogs are used to organize in-class discussions. At the State University of New York at Buffalo, for example, Alexander Halavais added a blog to his media law class of about 180 students. Course credit was awarded for online discussion, with topics ranging from the First Amendment to libel to Irish law reform. As the course wound down with a discussion of nude bikers, Halavais questioned whether he would continue the blog the following year because of the workload, but students were enthusiastic in their comments. Mireille Guay, an instructor at St-Joseph, notes: "The conversation possible on the weblog is also an amazing tool to develop our community of learners. The students get to know each other better by visiting and reading blogs from other students. They discover, in a non-threatening way, their similarities and differences. The student who usually talks very loud in the classroom and the student who is very timid have the same writing space to voice their opinion. It puts students in a situation of equity."

4. Some instructors are using blogs to organize class seminars and to provide summaries of readings. Used in this way, the blogs become "group blogs"—that is, individual blogs authored by a group of people. Farrell notes: "It becomes much easier for the professor and students to access the readings for a particular week—and if you make sure that people are organized about how they do it, the summaries will effectively file themselves."

5. Students may be asked to write their own blogs as part of their course grade. Educational Technologist Lane Dunlop wrote about one class at Cornell College: "Each day the students read a chunk of a book and post two paragraphs of their thoughts on the reading." In another class, French 304, students were given a similar exercise. Using a French-language blogging service called Monblogue, Molly, a business student, posted a few paragraphs every day."



#### **1.4.2 INFORMATION SHARING**

Add to the sum total of knowledge for research projects, share industry and product knowledge, capture and disseminate pertinent news from outside the enterprise, and contribute valuable insights on specific subjects. They are particularly useful for promoting knowledge in cross-cultural environments.

#### **1.4.3 SALES PROMOTIONS**

To market or sell products/services, make citizens aware of public services, get donations for charities or political parties etc written by the organization itself, or more specifically individuals within it writing on behalf of the organization to clients/customers and citizens, that is persons and/or other organizations that are directly involved with the publishing organization, or those it wants to reach with the purpose above.

#### **1.4.4 EXPRESSING IDEAS AND REVIEWS**

People around the world write blogs to share their ideas and express their reviews on different topics such as movies, music, fashion, politics, travel, gadgets, technology etc. sharing tips & tricks with each other helping each other by replying to posts and interacting with others through comments.

#### **1.4.5 SHARING EXPERIENCES**

People write blogs to also share their professional and personal experiences regarding various subjects example preparing for CAT, GMAT or their visit to particular place, country or about tips on presentation etc.

#### **1.4.6 EARNING MONEY**

People who have popular blogs and have number of users visiting them regularly can earn money by posting ads of different websites or products sometimes this depends upon the number of clicks on the ads in the blogsite or the money is for giving company a particular space on their blog for advertising

## **1.5 ADVANTAGES AND DISADVANTAGES**

### **1.5.1 GENERAL ADVANTAGES**

The advantages of blogs from an organizational perspective include the following:

1. The consumer and citizen are potentially better informed and this can only be good for the long-term health of our societies and economies.
2. Blogs have potential to help the organization develop stronger relationships and brand loyalty with its customers, as they interact with the 'human face' of the organization through blogs.
3. Blogs, in an intranet environment, can be an excellent way of sharing knowledge within the organization.
4. Blogs can be a positive way of getting feedback, and keeping your finger on the pulse, as readers react to certain pieces, suggest story ideas, etc.
5. Blogs can build the profile of the writer, showcasing the organization as having talent and expertise.
6. Plenty of information available for sharing and use without any constraint
7. Can be used as a mean of earning money many famous blogger's earn a lot of money for their posts, reviews etc
8. Interacting with others, sharing our experiences, getting tips from other people

### **1.5.2 DISADVANTAGES**

The disadvantages of blogs are:

1. Most people don't have very much to say that's interesting, and/or are unable to write down their ideas in a compelling and clear manner.
2. I have often found that the people who have most time to write have least to say, and the people who have most to say don't have enough time to write it. Thus, the real expertise within the organization lays hidden, as you get drowned in trivia.
3. Like practically everything else on the Web, blogs are easy to start and hard to maintain. Writing coherently is one of the most difficult and time-consuming tasks for a human



being to undertake. So, far from blogs being a cheap strategy, they are a very expensive one, in that they eat up time. As a result, many blogs are not updated, thus damaging rather than enhancing the reputation of the organization.

4. Organizations are not democracies. The Web makes many organizations look like disorganizations, with multiple tones and opinions. Contrary to what some might think, the average customer prefers it if the organization they are about to purchase from is at least somewhat coherent.
5. People making absence or abusive remarks about a person ,political party ,religion ,caste etc
6. People using blogs for illegal activities like sharing of sensitive or confidential information of an organization

## CHAPTER 2

### COMMON FEATURES

#### 2.1 The Blog Content

On a blog, the content consists of articles (also sometimes called "posts" or "entries") that the author(s) writes. Yes, some blogs have multiple authors, each writing his/her own articles. Typically, blog authors compose their articles in a web-based interface, built into the blogging system itself. Some blogging systems also support the ability to use stand-alone "weblog client" software, which allows authors to write articles offline and upload them at a later time. Users can include files in there upload some blog also supports multimedia files to be added in the post.

#### 2.2 Comments

Most blogs have a method to allow visitors to leave comments. There are also nifty ways for authors of other blogs to leave comments without even visiting the blog! Called "pingbacks" or "trackbacks", they can inform other bloggers whenever they cite an article from another site in their own articles. All this ensures that online conversations can be maintained painlessly among various site users and websites

#### 2.3 Archives

A blog is also a good way to keep track of articles on a site. A lot of blogs feature an archive based on dates (like a monthly or yearly archive). The front page of a blog may feature a calendar of dates linked to daily archives. Archives can also be based on categories featuring all the articles related to a specific category

#### 2.4 Feeds

A Feed is a function of special software that allows "Feedreaders" to access a site automatically looking for new content and then post updates about that new content to another site. This provides a way for users to keep up with the latest and hottest information posted on different blogging sites. Some Feeds include RSS (alternately defined as "Rich Site Summary" or "Really Simple Syndication"), Atom or RDF files.



Technically **web feed** is a document (often XML-based) which contains content items with web links to longer versions. News websites and blogs are common sources for web feeds, but feeds are also used to deliver structured information ranging from weather data to "top ten" lists of hit tunes to search results. The two main web feed formats are RSS and Atom.

Web feeds have some advantages compared to receiving frequently published content via email:

- When subscribing to a feed, users do not disclose their email address, so users are not increasing their exposure to threats associated with email: spam, viruses, phishing, and identity theft.
- If users want to stop receiving news, they do not have to send an "unsubscribe" request; users can simply remove the feed from their aggregator.
- The feed items are automatically "sorted" in the sense that each feed URL has its own sets of entries (unlike an email box, where all mails are in one big pile and email programs have to resort to complicated rules and pattern matching).

### **RSS FEED**

**RSS** (formally "RDF Site Summary", known colloquially as "Really Simple Syndication") is a family of Web feed formats used to publish frequently updated content such as blog entries, news headlines or podcasts. An RSS document, which is called a "feed", "web feed", or "channel", contains either a summary of content from an associated web site or the full text. RSS makes it possible for people to keep up with their favorite web sites in an automated manner that's easier than checking them manually.

RSS content can be read using software called an "RSS reader", "feed reader" or an "aggregator". The user subscribes to a feed by entering the feed's link into the reader or by clicking an RSS icon in a browser that initiates the subscription process. The reader checks the user's subscribed feeds regularly for new content, downloading any updates that it finds.

The initials "RSS" are used to refer to the following formats:

- Really Simple Syndication (RSS 2.0)
- RDF Site Summary (RSS 1.0 and RSS 0.90)

- Rich Site Summary (RSS 0.91)

RSS formats are specified using XML, a generic specification for the creation of data formats

## **2.5 Blogrolls**

A blogroll is a list, sometimes categorized, of links to webpage's the author of a blog finds worthwhile or interesting. The links in a blogroll are usually to other blogs with similar interests. The blogroll is often in a "sidebar" on the page or featured as a dedicated separate web page.

### ***Why would a Weblogger want a blogroll on their site?***

Blogrolls are used to express connections between weblogs. By having a blogroll, you are saying to the reader *"Here are other weblogs that I like- you should check them out too."* Further, you're telling other webloggers. *"I like your site and I think my readers will too."* On a high level, the blogroll says *"Here is my community of webloggers- I fit into this group."*

### ***How does it work?***

Tools like Blogrolling.com have made it ultra-simple to add a site to your blogroll. Using a tool like Blogrolling.com, you can visit a site and add a link to your weblog in two clicks. You add a piece of code to your weblog template and then just use the blogrolling.com site to add the links. No html, no files or templates from then on- it's just pushing a couple of buttons and a new link appears in your weblog.

## **2.6 Syndication**

A feed is a machine readable (usually XML) content publication that is updated regularly. Many weblogs publish a feed (usually RSS, but also possibly Atom and RDF and so on, as described above). There are tools out there that call themselves "feedreaders". What they do is they keep checking specified blogs to see if they have been updated, and when the blogs are updated, they display the new post, and a link to it, with an excerpt (or the whole contents) of the post. Each feed contains items that are published over time. When checking a feed, the feedreader is actually looking for new items. New items are automatically discovered and downloaded for you



to read. Just so you don't have to visit all the blogs you are interested in. All you have to do with these feedreaders is to add the link to the RSS feed of all the blogs you are interested in. The feedreader will then inform you when any of the blogs have new posts in them. Most blogs have these "Syndication" feeds available for the readers to

### **Web syndication**

**Web syndication** is a form of syndication in which a section of a website is made available for other sites to use. This could be simply by licensing the content so that other people can use it; however, in general, *web syndication* refers to making web feeds available from a site in order to provide other people with a summary of the website's recently added content (for example, the latest news or forum posts).

Large scale web syndication of content started in 2001 when Miniclip freely syndicated online browser based interactive games to the masses. Today many different types of content are syndicated on the Internet. Millions of online publishers including newspapers, commercial web sites and blogs now publish their latest news headlines, product offers or blog postings in standard format news feed.

*Syndication* benefits both the websites providing information and the websites displaying it. For the receiving site, content syndication is an effective way of adding greater depth and immediacy of information to its pages, making it more attractive to users. For the transmitting site, syndication drives exposure across numerous online platforms. This generates new traffic for the transmitting site — making syndication a free and easy form of advertisement.

The prevalence of web syndication is also of note to online marketers, since web surfers are becoming increasingly wary of providing personal information for marketing materials (such as signing up for a newsletter) and expect the ability to subscribe to a feed instead.

Although the format could be anything transported over HTTP, such as HTML or JavaScript, it is more commonly XML. The two main families of web syndication formats are RSS and Atom.

## **2.7 Managing Comments**

One of the most exciting features of blogging tools are the comments. This highly interactive feature allows users to comment upon article posts and link to your posts and comment on and recommend them. These are known as **trackbacks** and **pingbacks**

- Trackbacks
- Pingbacks
- Verifying Pingbacks and Trackbacks
- Comment Moderation
- Comment Spam

### **2.7.1 Trackbacks**

Trackback was designed to provide a method of notification between websites: it is a method of person A saying to person B, "This is something you may be interested in." To do that, person A sends a Trackback ping to person B.

A better explanation is this:

- Person A writes something on their blog.
- Person B wants to comment on Person A's blog, but wants her own readers to see what she had to say, and be able to comment on her own blog
- Person B posts on her own blog and sends a trackback to Person A's blog
- Person A's blog receives the trackback, and displays it as a comment to the original post.  
This comment contains a link to Person B's post

The Trackback specification is due to Six Apart who first implemented it in its Movable Type blogging software in August 2002. It has since been implemented in most other blogging tools. Six Apart started a working group in February 2006 to improve the Trackback protocol with the goal to eventually have it approved as an internet standard by the IETF. One notable blogging tool that does not support trackback yet is Blogger. (Blogger now has backlinks - very similar to the trackback feature in Movable Type.)



Some individuals or companies have abused the Trackback feature to insert spam links on some blogs (see sping). This is similar to comment spam but avoids some of the safeguards designed to stop the latter practice. As a result, Trackback spam filters similar to those implemented against comment spam now exist in many weblog publishing systems. Many blogs have stopped using trackbacks because dealing with spam became too burdensome.

### **2.7.2 Pingbacks**

A **Pingback** is one of three types of Linkbacks, methods for Web authors to request notification when somebody links to one of their documents. This enables authors to keep track of who is linking to, or referring to their articles. Some weblog software, such as WordPress and Community Server, support automatic pingbacks where all the links in a published article can be pinged when the article is published.

Essentially, a Pingback is a signal (ping) sent from Site A to Site B. However, it's also a link. When Site B receives the notification signal, it automatically goes back to Site A checking for the existence of a live incoming link. If that link exists, the Pingback is recorded successfully. This makes Pingbacks less prone to spam than Trackbacks.

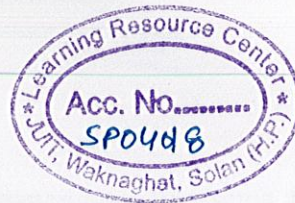
Both sites must be Pingback enabled in order to establish this communication. If a site is Pingback enabled, each time you link-out you will be "pinging" external sites. Pingback requires for Site A to physically link to Site B

Pingbacks were designed to solve some of the problems that people saw with trackbacks.

The best way to think about pingbacks is as **remote comments**:

- Person A posts something on his blog.
- Person B posts on her own blog, linking to Person A's post. This automatically sends a pingback to Person A when both have pingback enabled blogs.
- Person A's blog receives the pingback, then **automatically** goes to Person B's post to confirm that the pingback did, in fact, originate there.





### 2.7.3 Verifying Pingbacks and Trackbacks

Comments on blogs are often criticized as lacking **authority**, since anyone can post anything using any name they like: there's no verification process to ensure that the person is who they claim to be. Trackbacks and Pingbacks both aim to provide some verification to blog commenting.

### 2.7.4 Comment Spam

Comment Spam refers to useless comments (or trackbacks, or pingbacks) to posts on a blog. These are often irrelevant to the context value of the post. They can contain one or more links to other websites or domains. Spammers use Comment Spam as a medium to get higher page rank for their domains in Google, so that they can sell those domains at a higher price sometime in future or to obtain a high ranking in search results for an existing website

Spam in blogs (also called simply **blog spam** or **comment spam**) is a form of spamdexing. It is done by automatically posting random comments or promoting commercial services to blogs, wikis, guestbook's, or other publicly accessible online discussion boards. Any web application that accepts and displays hyperlinks submitted by visitors may be a target.

Adding links that point to the spammer's web site artificially increases the site's search engine ranking. An increased ranking often results in the spammer's commercial site being listed ahead of other sites for certain searches, increasing the number of potential visitors and paying customers.

### 2.7.5 Comment Moderation

Comment Moderation is a feature which allows the website owner and author to monitor and control the comments on the different article posts, and can help in tackling comment spam. It lets you moderate comments, & you can delete unwanted comments, approve cool comments and make other decisions about the comments

A **moderation system** is the method the webmaster chooses to sort contributions which are irrelevant, obscene, illegal, or insulting with regards to useful or informative contributions.

Various types of Internet sites permit user comments, such as: Internet forums, blogs, and news sites powered by scripts such as phpBB, a Wiki, or PHP-Nuke. Depending on the site's content



and intended audience, the webmaster will decide what kinds of user comments are appropriate, then delegate the responsibility of sifting through comments to lesser moderators. Most often, webmasters will attempt to eliminate trolling, spamming, or flaming, although this varies widely from site to site.

This kind of moderation system is often seen in blogs. the administrator have special privileges to delete or edit others' contributions and/or exclude people based on their e-mail address or IP address, and generally attempt to remove negative contributions throughout the community.

## Chapter 3

### PROJECT DETAILS AND DESIGN

#### 3.1 Problem Statement

- The Project aims at developing a **BLOGGING PLATFORM on Asp. Net and C#** for an efficient management of various types of blogs with the help of different features.
- Features such as blog content, comments, archives, feeds, blogrolls etc are to be implemented.
- Blogging Platform is also known as **Content Management System**, hence as the name suggest, our main aim is to produce an efficient database connectivity to manage the contents and other factors.

#### 3.2 Software/Hardware Requirements

##### 1. Microsoft Visual Studio 2005

To make asp.net web pages with back-end coding on C sharp.

##### 2. MySQL ConnectorNet1.0.9

MySQL Connector/Net enables developers to easily create .NET applications that require secure, high-performance data connectivity with MySQL. It implements the required ADO.NET interfaces and integrates into ADO.NET aware tools. Developers can build applications using their choice of .NET languages. MySQL Connector/Net is a fully-managed ADO.NET driver written in 100% pure C#.

#### Features

MySQL Connector/Net includes full support for:

- MySQL 5.0 features (stored procedures, etc.)
- MySQL 4.1 features (server-side prepared statements, Unicode, and shared memory access, etc.)



- Large-packet support for sending and receiving rows and BLOBs up to 2 gigabytes in size
- Protocol compression which allows for compressing the data stream between the client and server
- Support for connecting using TCP/IP sockets, named pipes, or shared memory on Windows
- Support for connecting using TCP/IP sockets or Unix sockets on Unix
- Support for the open source Mono framework developed by Novell
- Fully-managed, does not utilize the MySQL client library.

### **Interoperability**

MySQL Connector/Net is known to work with the following tools:

- Microsoft Visual Studio.NET 2003
- Microsoft WebMatrix
- Borland Delphi 8
- Borland C# Builder

### **3. MySQL Administrator**

MySQL Administrator is a powerful visual administration console that enables you to easily administer your MySQL environment and gain significantly better visibility into how your databases are operating. MySQL Administrator now integrates database management and maintenance into a single, seamless environment, with a clear and intuitive graphical user interface. By using MySQL Administrator you will be able to:

- Achieve higher database availability through improved management
- Reduce errors through visual database administration
- Lower database administration costs through improved productivity
- Deliver a more secure environment through easier privilege management

MySQL Administrator enables developers and DBAs to easily perform all the command line operations visually including configuring servers, administering users, and dynamically monitoring database health. Other common administrative tasks such as monitoring replication

status, backup and restore, and viewing logs can also be performed through the MySQL Administrator graphical console. Specifically MySQL Administrator delivers:

"MySQL Administrator is an extremely intuitive visual tool that makes managing MySQL much easier and faster. Now developers can quickly authenticate MySQL users and create backups, which is critical to the long-term health of a large database infrastructure optimizing and fine-tuning the server for more performance has become easier."

### **User Administration in Minutes**

Administering users, granting privileges and viewing user privilege information has never been easier! Now you can leverage the full power of MySQL's user management and privilege capabilities to fully protect your MySQL server host using an integrated graphical user administration console. For example, a form-based interface provides a simple method for adding users, assigning passwords and including user profile information. Plus, assigning and revoking global, database, and table/column privileges is as easy as adding and removing privilege items from an available list. Finally, an explorer interface provides a hierarchical view allowing you to easily view user privileges by privilege type.

### **Single View Dynamic Health Monitoring**

Dynamic health monitoring gives you an immediate view into the key indicators of the health of your MySQL environment. Dynamic graphs enable you to easily view percentage of memory usage, connection usage, hit rates, as well as the number of SQL queries on an ongoing basis, as your system runs. The Disk Health tab gives you an overview of the amount of hard disk space, tablespace, and log files being used. You can also use the visual health monitoring feature to get a hierarchical view into status and server variables such as the number of outstanding connections, number of bytes sent and received by clients, buffer allocations size and more.

### **Quickly Optimize MySQL**

MySQL Administrator now allows you to easily view and setup key parameters so you can quickly optimize the performance of MySQL. Using a graphical forms-based environment, you



quickly fine tune general and advanced parameters. For example you can set memory usage parameters and configure server start-up parameters related to the MyISAM and InnoDB table types.

### **Rapid Disaster Prevention and Recovery**

Now you can rapidly create consistent backups and restore data by simply visually selecting the schemas and tables you want to back/restore from your hierarchical data directory. The graphical environment gives you additional visibility into Tabletype, Rows, Data Length, and Update time.

### **Server Information At-a-Glance**

The server information section of MySQL Administrator allows you to quickly view basic information about the server and client. This view gives you immediate visibility into items such as server status, client and server version numbers and IP address, as well as connection information.

### **View Replication Status**

Now you can quickly set up replication to increase the robustness and speed of your MySQL environment. The replication features within MySQL Administrator allow you to easily view the network topology of your replication master and slaves. Plus, the graphical environment provides additional visibility into IP addresses and logfiles.

### **View Server Logs from a Centralized Administration Environment**

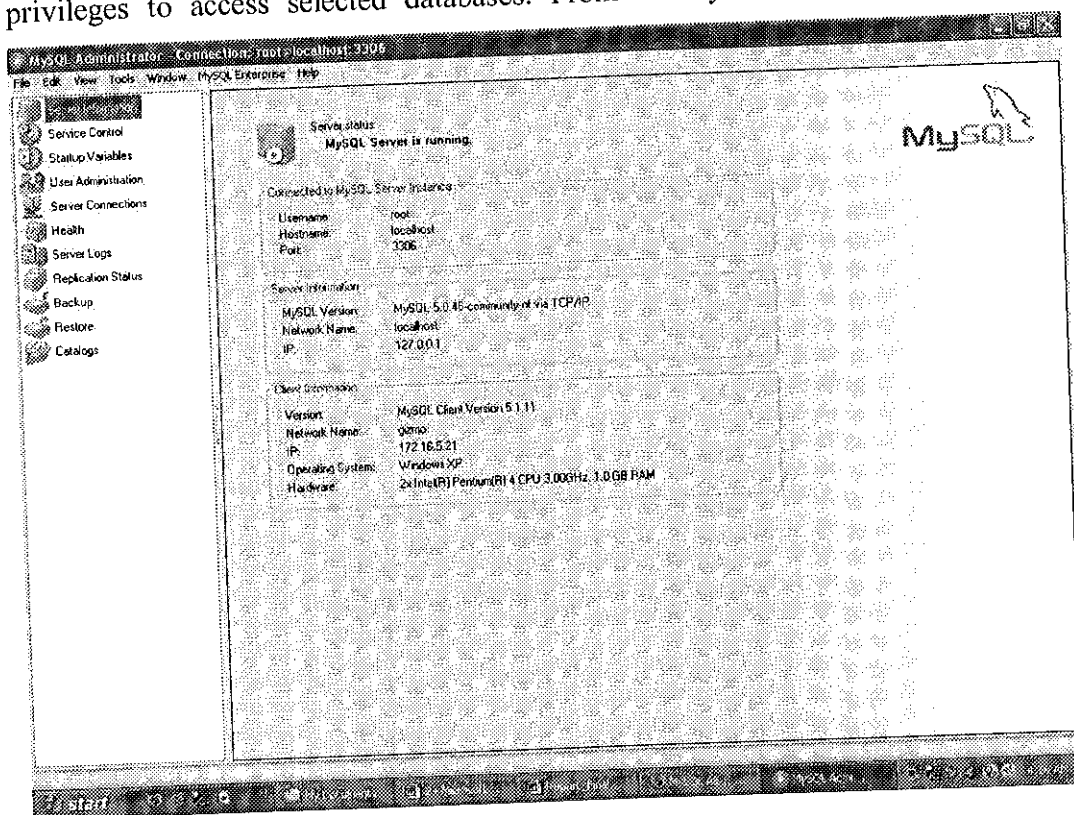
Now you can view all the MySQL log files including error logs, binary logs, and InnoDB logs from a single, centralized administration environment so you can diagnose server problems quicker and track database changes.

## Service Control

Gain complete control of service creation and overcome the time consuming task of installing and setting up services. Plus, start and stop MySQL servers and view corresponding log messages.

## Browse Catalog

Browse your catalog and view corresponding tables with columns, all indices and users with privileges to access selected databases. From here you can monitor and optimize tables.



## Cross Platform

MySQL Administrator is available for Microsoft Windows, Linux and Mac OS X 10.3, 10.4, and is compatible with MySQL version 4.0 or later. On Windows platforms, MySQL Administrator is supported on Win2k/WinXP and Windows Server 2003.



### System Specification (Minimum)

- ✓ Windows 98/Windows XP/Windows Vista
- ✓ A Web Server
- ✓ At least 64 Mb of RAM
- ✓ Internet Connectivity

### 3.3 Database Description

We are using MySQL version 5.0.45 as a database for our blogging platform.

Source: <http://dev.mysql.com/downloads/index.html>

The name of our database is 'blogDB'. Currently there are 7 tables in it.

(i) posts, (ii) categories, (iii) postcategory, (iv) Comments, (v) logininfo, (vi) blogdetail, (vii) blogroll

Their Structures are as follows:-

Table Name: *posts*

| Column Name     | Datatype | Not Null | Auto Inc |
|-----------------|----------|----------|----------|
| postid          | integer  | yes      | yes      |
| title           | varchar  | no       | no       |
| description     | varchar  | no       | no       |
| content         | text     | no       | no       |
| datetimecreated | datetime | no       | no       |
| author          | varchar  | no       | no       |
| uploadfilename  | varchar  | no       | no       |
| datecreated     | text     | no       | no       |
| commentcount    | integer  | no       | no       |
| subscribe       | text     | no       | no       |

Table Name: *categories*

| Column Name | Datatype | Not Null | Auto Inc |
|-------------|----------|----------|----------|
| categoryid  | integer  | yes      | yes      |
| Name        | varchar  | no       | no       |

Table Name: *postcategory*

| Column Name    | Datatype | Not Null | Auto Inc |
|----------------|----------|----------|----------|
| postcategoryid | integer  | yes      | yes      |
| postid         | integer  | no       | no       |
| categoryid     | integer  | no       | no       |

Table Name: *comments*

| Column Name   | Datatype | Not Null | Auto Inc |
|---------------|----------|----------|----------|
| commentid     | integer  | yes      | yes      |
| postid        | varchar  | no       | no       |
| commentdate   | varchar  | no       | no       |
| author        | text     | no       | no       |
| email         | datetime | no       | no       |
| website       | varchar  | no       | no       |
| country       | text     | no       | no       |
| commentdetail | varchar  | no       | no       |
| status        | integer  | yes      | no       |

Table Name: *logininfo*

| Column Name   | Datatype | Not Null | Auto Inc |
|---------------|----------|----------|----------|
| id            | integer  | yes      | yes      |
| lastlogininfo | integer  | no       | no       |

Table Name: *blogdetail*

| Column Name  | Datatype | Not Null | Auto Inc |
|--------------|----------|----------|----------|
| id           | integer  | yes      | yes      |
| name         | text     | no       | no       |
| description  | text     | no       | no       |
| owner        | text     | no       | no       |
| ownerprofile | text     | no       | no       |

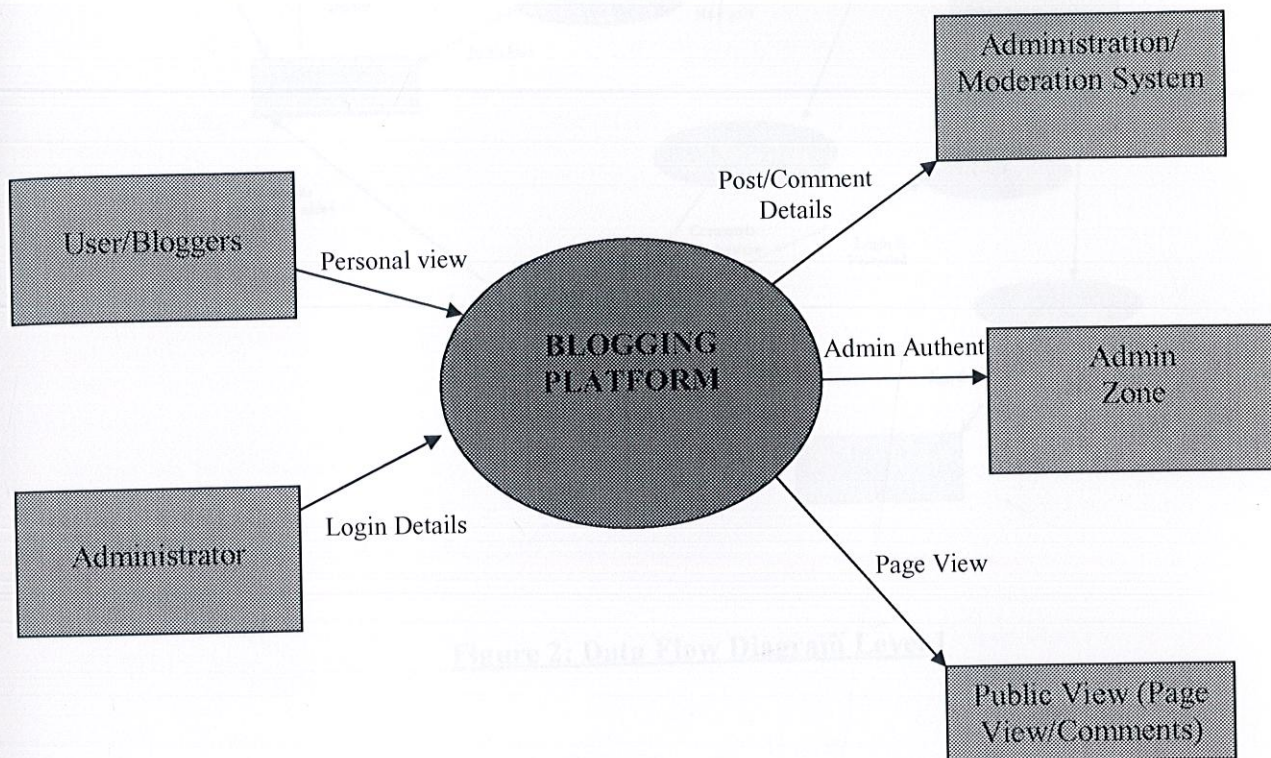
Table Name: *blogroll*

| Column Name | Datatype | Not Null | Auto Inc |
|-------------|----------|----------|----------|
| id          | integer  | yes      | yes      |
| name        | text     | no       | no       |
| url         | text     | no       | no       |



### 3.4 Design Specifications

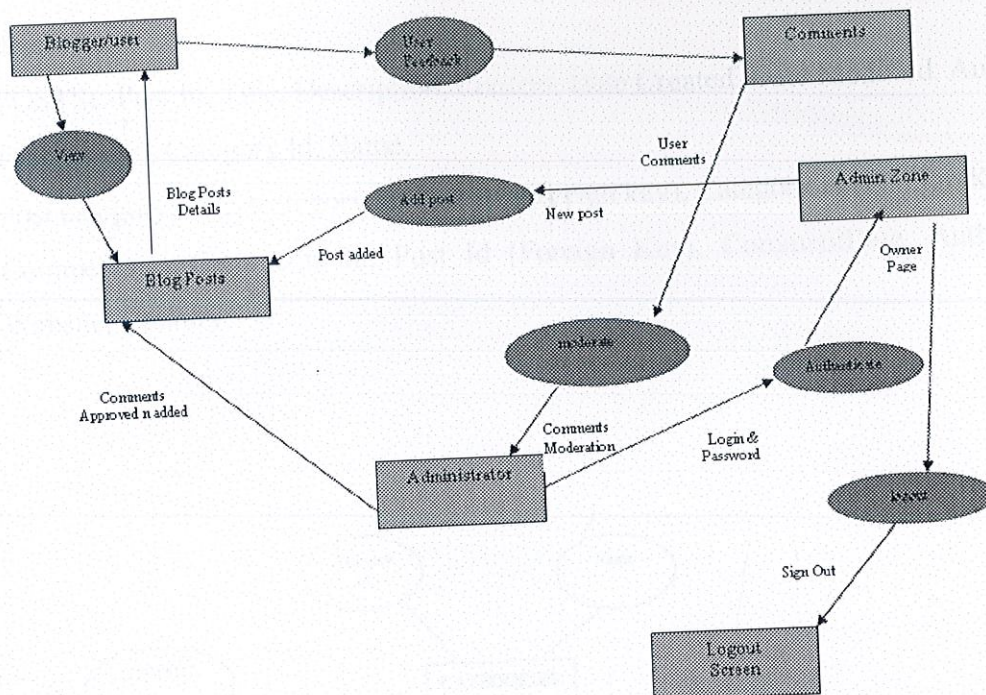
#### 3.4.1 Data Flow Diagram



**Figure 1: Data Flow Diagram Level 0**



Level 1 DFD



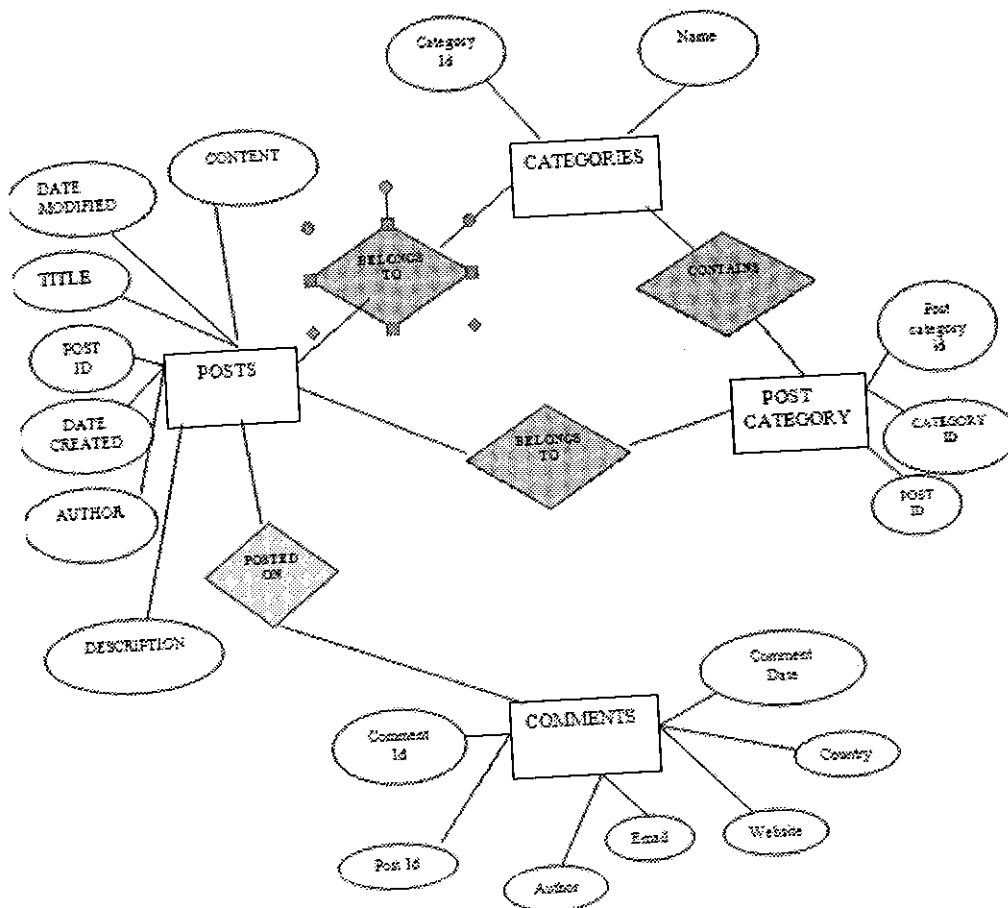
**Figure 2: Data Flow Diagram Level 1**



### 3.4.2 Entity Relationship Diagram

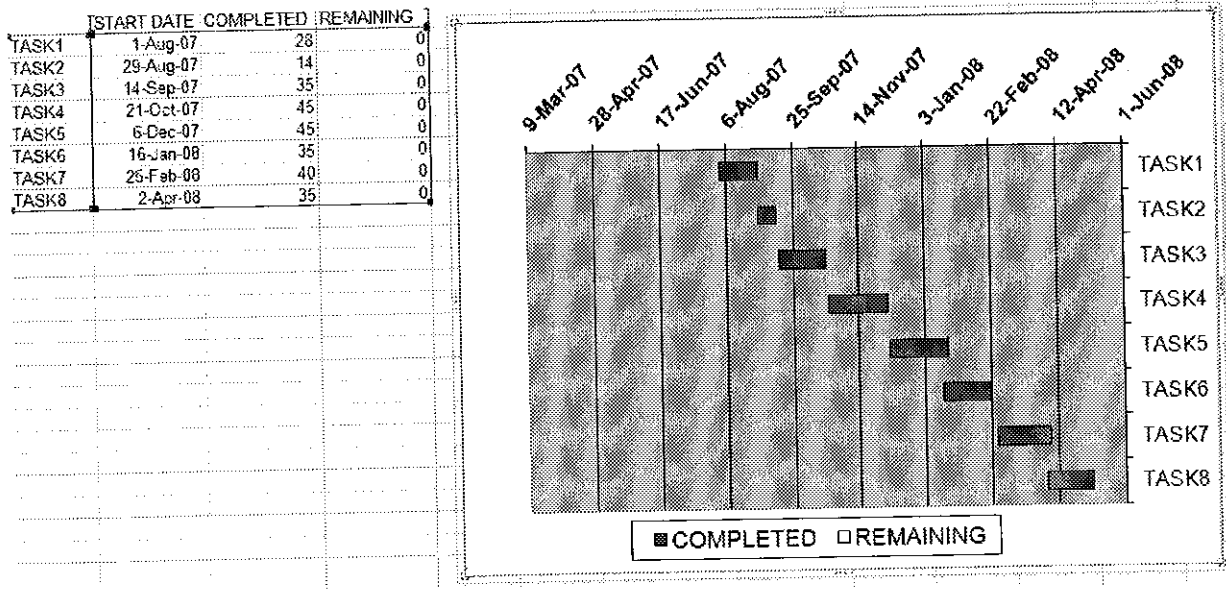
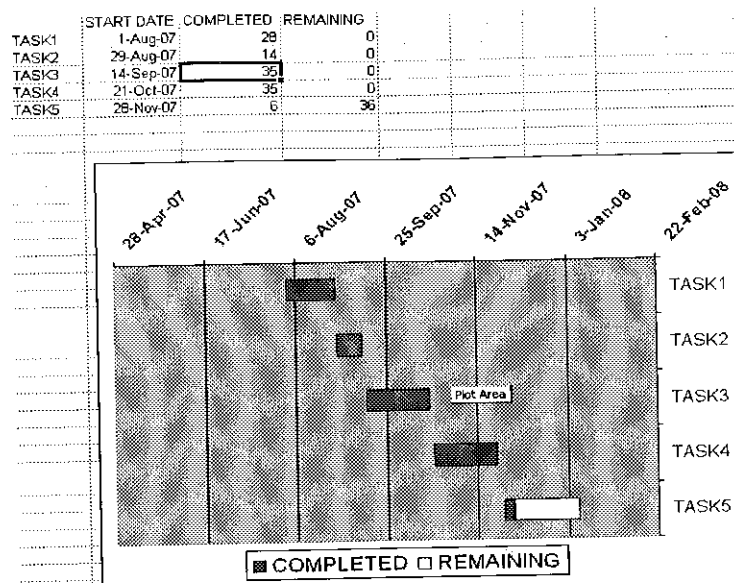
#### Data Objects

- POSTS: -Post Id, Title, Description, Content, Date Created, Date Modified, Author.
- Categories: - Category Id, Name.
- Post category: - Postcategory Id, Post Id (Foreign key), Category Id (Foreign Key).
- Comments: - Comments Id, Post Id (Foreign Key), CommentDate, Author, Email, Website, Country



**Figure 3: Entity Relationship Diagram**

### 3.4.4 Gantt Chart

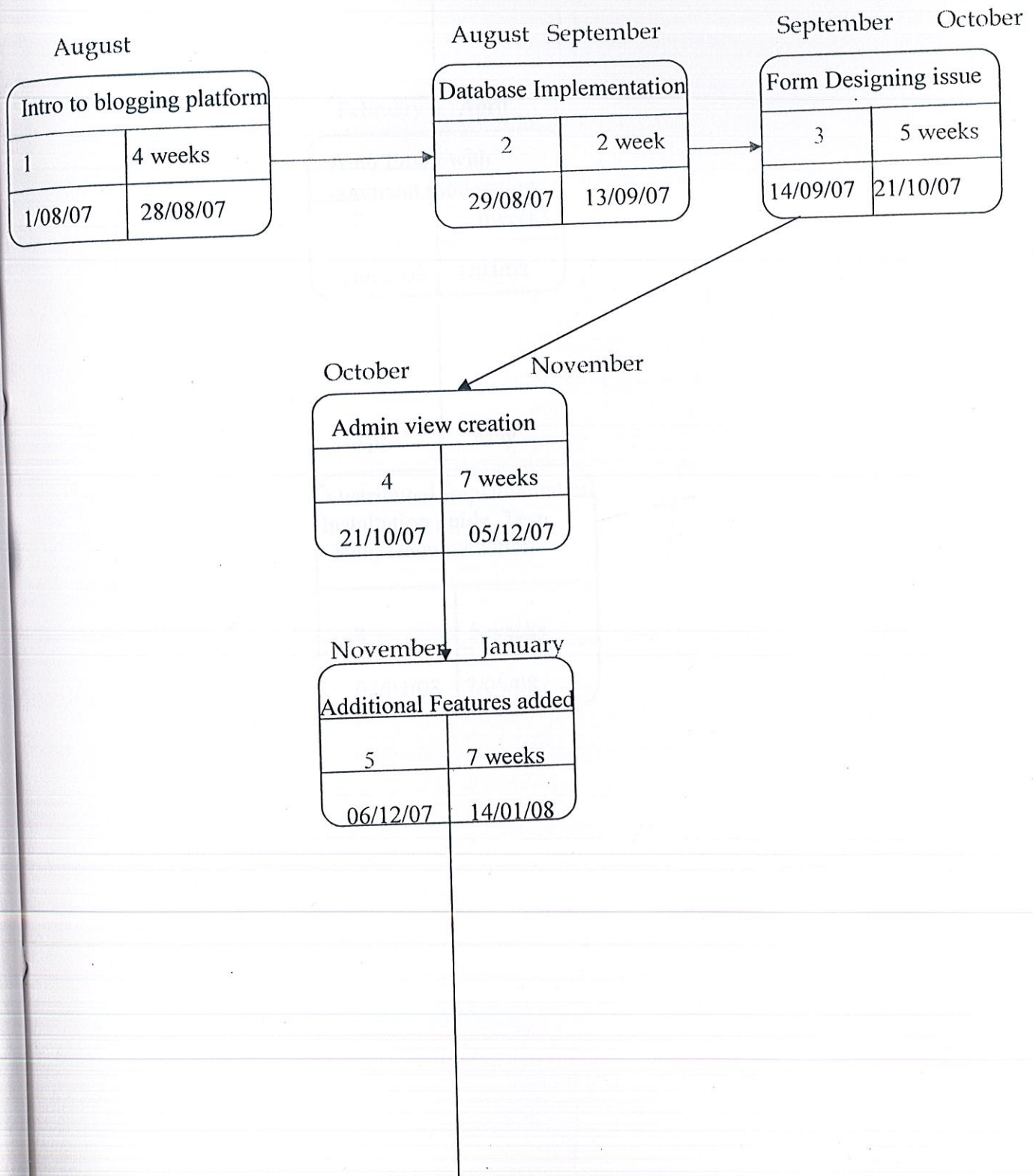


**Figure 4: Gantt Chart**



### 3.4.4 Pert Chart

Figure 5: Pert Chart



January      February

Theme Design and  
Security

6      6weeks

16/01/08      25/02/08

February      April

Auto Email with  
comment moderation

7      6weeks

26/02/08      1/04/08

April      May

Distributed Implementation  
Installation guide, Test  
Run, Documentation

8      5 weeks

02/04/08      7/05/08



## CHAPTER 4

### MODULES IMPLEMENTED

#### 4.1 Public Zone

##### 4.1.1 Database Connectivity

To connect to the database following lines of code are used:-

```
string str =  
System.Configuration.ConfigurationManager.ConnectionStrings["abcd"].ToString(  
);  
MySqlConnection con = new MySqlConnection(str);
```

##### 4.1.2 Home Page

Home page is the main page visible to the blog reader displaying the name of the blog, Information about the Author, Recent posts, Popular Posts, Recent Comments, Contact Us, Sign In and Search option. The following code is used to retrieve data from the database.

```
MySqlDataAdapter da = new MySqlDataAdapter("select  
postid,title,content,author,datecreated,datetimecreated, commentcount from  
posts order by datetimecreated desc", con);  
  
MySqlDataAdapter da1 = new MySqlDataAdapter("select owner,ownerprofile,email  
from blogdetail", con);  
  
MySqlDataAdapter da2 = new MySqlDataAdapter("select postid,title,commentcount  
from posts where commentcount>1 order by commentcount desc", con);  
  
MySqlDataAdapter da3 = new MySqlDataAdapter(" select  
postid,author,commentdetail from comments order by commentdate desc", con);  
  
DataTable dt = new DataTable();  
DataTable dt1 = new DataTable();  
DataTable dt2 = new DataTable();  
DataTable dt3 = new DataTable();
```



#### **4.1.3. Blog Roll**

Blog Roll gives Information about other related blogs which the author visits or suggests you to visit.

```
string str =  
System.Configuration.ConfigurationManager.ConnectionStrings["abcd"].ToString(  
);  
    MySqlConnection con = new MySqlConnection(str);  
    MySqlDataAdapter da = new MySqlDataAdapter("select name,url from  
blogroll", con);  
    DataTable dt = new DataTable();
```

#### **4.1.4. Contact Us**

Contact Us page provides function to contact the blog administrator for communication purpose. This also prevents direct spamming of administrators email account. The following code shows how a mail is send to the blog administrator's email.

```
MailMessage objEmail = new MailMessage();  
objEmail.To = "raks.khare@gmail.com";  
objEmail.From = "roxisayshi@gmail.com";  
objEmail.Subject = "testing";  
objEmail.Body = "hi";  
objEmail.Priority = MailPriority.High;  
SmtpMail.SmtpServer = "localhost";
```

#### **4.1.5. Search**

Search options help you to search blogs related to a particular topic from our website as well as we have provided an option to search the web with the help of Google search.

#### **4.1.6. Sign In**

Sign In section enable Administrator to sign in, so that he/she can access the Admin Zone. The following code shows how an actual admin is authenticated and then authorized to view pages.

```
if (FormsAuthentication.Authenticate(txtusername.Text.Trim(),  
txtpwd.Text.Trim()))  
{  
    if (chkrememberme.Checked)
```



```

{
    HttpCookie remme = new HttpCookie("RememberMe");

    remme.Values.Add("User Name", txtusername.Text);
    remme.Values.Add("Password", txtpwd.Text);
    remme.Expires = DateTime.Now.AddDays(1);
    Response.Cookies.Add(remme);
}
else
{
    if (Request.Cookies["RememberMe"] != null)
    {
        Request.Cookies["RememberMe"].Values.Add("User Name", "");
    }
}
FormsAuthentication.RedirectFromLoginPage(txtusername.Text.Trim(), true);
}

```

#### 4.1.7. Archives

To view older posts we have created an archives section so that older posts are not neglected by any means. They can be assessed at any time. The following code shows how the older posts are retrieved from the database.

```

MySqlDataAdapter da = new MySqlDataAdapter("select
postId,title,content,author,datetimecreated,datecreated from posts where
datecreated='" + newarcddate + "'", con);
DataTable dt = new DataTable();
da.Fill(dt);
grdviewarchive.DataSource = dt;
grdviewarchive.DataBind();

```

#### 4.1.8. Comments

Comments are the vital past of any blogs. The articles which the blog owner writes get feedbacks from readers through comments section. Readers give their viewpoint by commenting to the respective posts. The following code shows how the comments are saved in database.

```

DateTime currtimel = DateTime.Now;
String currtime = currtimel.ToString("yyyy-MM-dd hh-mm-ss");

MySqlCommand cmdInsert = new MySqlCommand("Insert into
comments(postid,commentdate,author,email,website,commentdetail) values('" +
pid + "','" + currtime + "','" + txtname.Text + "','" + txtemail.Text + "','" +
+ txtwebsite.Text + "','" + txtcomment.Text + "')", con);

```



```

MySQLCommand cmdupdate = new MySQLCommand("update posts set
commentcount=commentcount+1 where postid=" + pid, con);

```

```

int i = cmdInsert.ExecuteNonQuery();
int i1 = cmdupdate.ExecuteNonQuery();

```

#### **4.1.9. Spam Protection**

To stop spamming through comments, we have developed a Security Image called as Captcha Image which asks users to enter some random numbers before submitting any comments to any particular posts. This is how it is done.

```

if (this.CodeNumberTextBox.Text ==
this.Session["CaptchaImageText"].ToString())
{
    this.MessageLabel.CssClass = "info";
    this.MessageLabel.Text = "Correct!";
    CodeNumberTextBox.Focus();
}
else
{
    this.MessageLabel.CssClass = "error";
    this.MessageLabel.Text = "ERROR: Incorrect, try again.";
}

```

#### **4.2.1. Phishing Filter**

Phishing Filter is deployed in our blogging platform to prevent spoofing of website. A user will have to upload any image to the server and the next time he/she logs in, the same image should be displayed. Thus it acts as a confirmation to the user that the website is genuine.

### **4.2. Administrative Zone**

#### **4.2.1. Database Connectivity**

To connect to the database following lines of code are used:-

```

string str =
System.Configuration.ConfigurationManager.ConnectionStrings["abcd"].ToString(
);
MySQLConnection con = new MySQLConnection(str);

```



#### 4.2.2. Administrator Home

Administrator home is available only after you have logged in as administrator at the login page.

It helps you to Add Post, Delete Post, Change settings and Comment Moderation.

```
MySqlDataAdapter da = new MySqlDataAdapter("select  
postid,title,content,author,datecreated from posts order by datecreated  
desc", con);  
  
DataTable dt = new DataTable();  
try  
{  
    con.Open();  
    da.Fill(dt);  
    grdviewpost.DataSource = dt;  
    grdviewpost.DataBind();  
}
```

#### 4.2.3. Add Post

Add Post page helps the administrator to add new posts on various topics under different categories, it also allows to add categories and upload images.

```
String filename = Session["filename"].ToString();  
DateTime currtime1 = DateTime.Now;  
String currtime = currtime1.ToString("yyyy-MM-dd hh:mm:ss");  
String currdate = currtime1.ToString("yyyy-MM-dd");  
  
MySqlCommand cmdInsert = new MySqlCommand("Insert into  
posts(title,description,author,content,datetimecreated,datecreated,uploadedfi  
lename) values('" + txttitle.Text + "','" + txtdescription.Text + "','" +  
txtauthor.Text + "','" + txtcontent.Text + "','" + currtime + "','" +  
currdate + "','" + filename + "')", con);  
  
int i = cmdInsert.ExecuteNonQuery();
```

#### 4.2.4. Admin Blog Roll

It allows the Administrator to add others related blogs he visits or would recommend his readers to visit.

```
MySqlCommand cmdInsert = new MySqlCommand("Insert into blogroll(name,url)  
values('" + txtname.Text + "','" + txturl.Text + "')", con);  
int i = cmdInsert.ExecuteNonQuery();
```



#### **4.2.5 Settings**

Settings page allows the Administrator to change the name of the blog, its description, blog owner information etc.

```
String strfilename="adminpic.jpg";
FileUpload1.SaveAs(MapPath("uploaded files") + "\\\" + strfilename);
string strfullimage = (MapPath("uploaded files") + "\\\" + strfilename);

//lblmessage.Text = strfullimage;

System.Drawing.Image imgfull = System.Drawing.Image.FromFile(strfullimage);

System.Drawing.Image imgthumb = imgfull.GetThumbnailImage(100, 100, null,
IntPtr.Zero);

imgthumb.Save(MapPath("uploaded files") + "\\\" + "adminpicthumb.jpg");
lblmessage.Text = "Picture Uploading Successfull!!!"
```

#### **4.2.6 Encryption and Decryption of Asp.Net Configuration File**

In Asp.Net, we can also encrypt and decrypt the web.config file. For encryption, the command is

```
aspnet_regiis -pef appSettings C:\Inetpub\wwwroot\mysite
```

For decryption, the command is

```
aspnet_regiis -pdf appSettings C:\Inetpub\wwwroot\mysite
```

#### **4.2.7 Comments Moderation**

To prevent spamming by humans, we have devised a method to prevent this. All comments will first go through security image and then the administrator moderation. They are displayed only when cleared all the way.

```
RadioButton rdb = (RadioButton)gvrow.FindControl("rdbapprove");
if(rdb.Checked==true)
{
```

```
int x = gvrow.RowIndex;
String str1 = dt.Rows[x]["postid"].ToString();
String query = "Update comments set status = 1 where postid = " + str1;
MySQLCommand cmd = new MySQLCommand(query, con);
cmd.ExecuteNonQuery();
Response.Redirect("moderate.aspx");
}
```

#### **4.2.8 Delete Post**

Delete Post page helps the administrator to delete various posts on various topics under different categories.

```
string str =
System.Configuration.ConfigurationManager.ConnectionStrings["abcd"].ToString(
);
MySQLConnection con = new MySQLConnection(str);
```



## CONCLUSION AND FUTURE WORK

If you are just getting your feet wet in the blogging world there are many blogging platforms you may come across to decide upon. You need to know what exactly you would like to do with your blogging software. If you would just like a simple means to blog information for your readers with the intent to have your blog as a side page, our Blogging Platform may be a good platform for you to work from.

Whatever choice you decide, blogging is a great means to reach many visitors and relay important information they could use.

Blogging is quickly becoming one of the preferred methods of marketing for your business due to the ease and enjoyment one can have with adding plugins, trying out new themes/templates and creating a new unique look that has the ability to have unique content posted right onto the main frame for search engine optimization.

In this project, we have studied the various features of the blogging platform and tried to implement many of those by using a suitable tool. We have also developed the fully functional our own blogging platform. Through out the year our focus was to build an efficient blogging platform also taking into consideration various security measures at different levels. Comment Moderation, Anti Spamming Technique, Phishing Filter etc. are the types of security measures deployed in this project.

Common blogging features such as Archives section, BlogRoll, Search, Comments etc are also implemented. Extra care has been taken in designing the theme design. Smart use of tiled images both horizontally and vertically makes the loading of pages super fast. On the server side, we have put up MySQL as the database for faster insertion and retrieval of data. The database is fully protected with the password which only blog administrator knows.

In short, we have reached to a mark where we can say that our project can not only be tested but can be implemented in real life as well.

The proposed future work is as follows:-

- 1) Theme Editor
- 2) Plugins Support
- 3) Rich Text Editor
- 4) Auto Upgradation to newer versions



## BIBLIOGRAPHY

1. Walther, S *ASP.NET Unleashed, Second Edition* Sams Publishing 2004.
2. Dunaway, R.B. *A book for Visual Studio dot Net for Developers.*
3. Moroney, L *Pro ASP.NET 2.0 in VB 2005, Special Edition* Apress Berkely 2006.
4. Kaufmann, J. *Beginning ASP.NET 2.0 Databases* Wiley Publishing, Inc Indiana 2005.
5. Prosise, J *Programming Microsoft .Net* Microsoft Press 2002.
6. Deitel, H.M. and Deitel, P.J. *Visual C#.Net Learn How to Program.*
7. Deitel, H.M. and Deitel, P.J. *Advanced Visual C#.Net Learn How to Program.*
8. Weakly, R. *Tech Yourself CSS in 24 Hours* Sams Publishing 2006.
9. [www.asp.net](http://www.asp.net)
10. [www.accelerate.com/asp.net/aspnet\\_csharp.htm](http://www.accelerate.com/asp.net/aspnet_csharp.htm)
11. [www.csharpfriends.com](http://www.csharpfriends.com)
12. [www.learnvisualstudio.net](http://www.learnvisualstudio.net)
13. [msdn.microsoft.com/asp.net](http://msdn.microsoft.com/asp.net)

# APPENDIX

## Appendix A: Snapshots

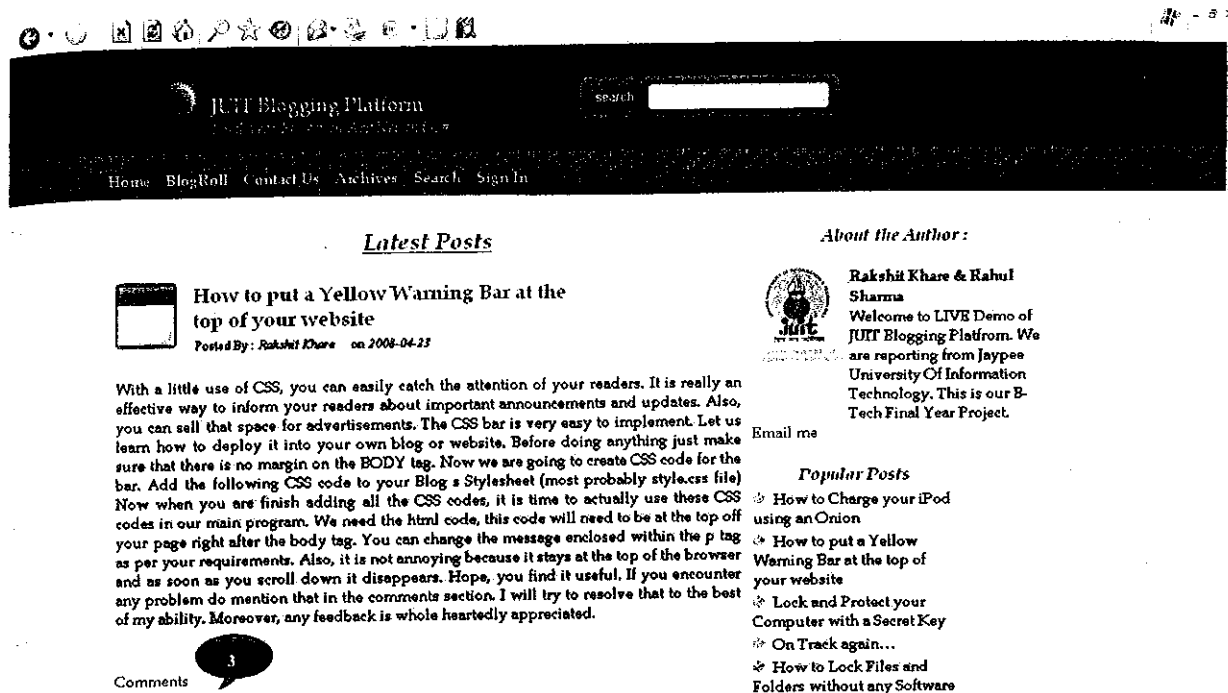


Figure 1 : HOME

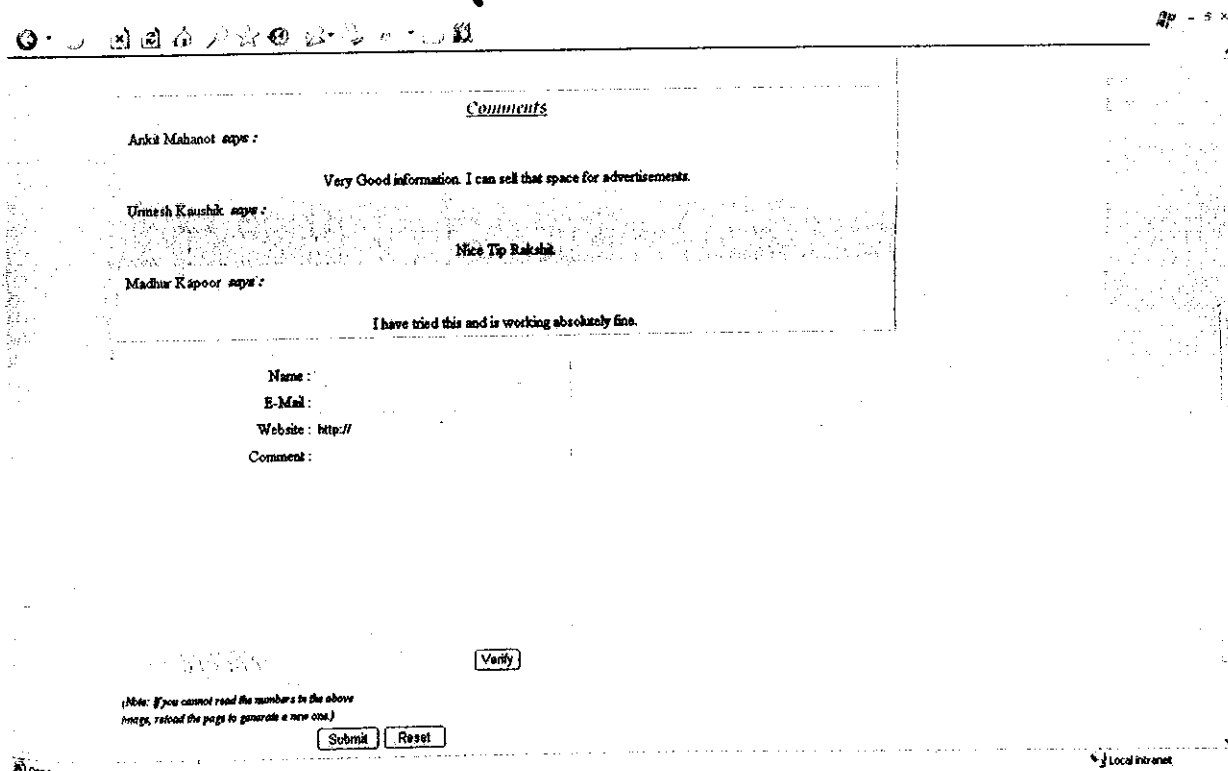


Figure 2 : COMMENTS



**IUT Blogging Platform**  
[Home](#) [BlogRoll](#) [Contact Us](#) [Archives](#) [Search](#) [Sign In](#)

Blog Name : **Rakshit Khare Blog- Aarambh**  
 Blog URL : [http:// www.rakshitk.com](http://www.rakshitk.com)

Blog Name : **Rahul Sharma**  
 Blog URL : [http:// www.writingof.blogspot.com](http://www.writingof.blogspot.com)

Blog Name : **Madhur Kapoor**  
 Blog URL : [http:// www.whoismadhur.com](http://www.whoismadhur.com)

Blog Name : **Ankit Mahanot Blog- Videos and Trailers**  
 Blog URL : [http:// www.video-trailers.com](http://www.video-trailers.com)

Figure 3 : BLOGROLL

**IUT Blogging Platform**  
[Home](#) [BlogRoll](#) [Contact Us](#) [Archives](#) [Search](#) [Sign In](#)

Name : \_\_\_\_\_

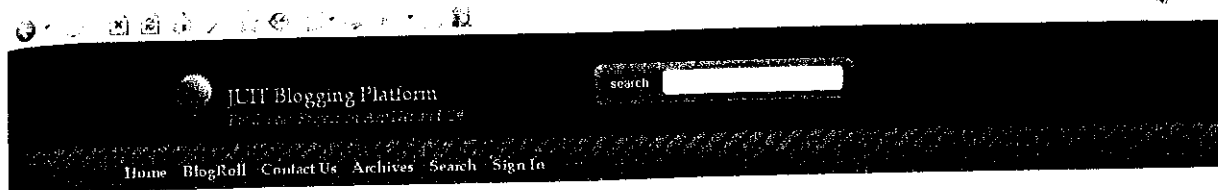
E-mail : \_\_\_\_\_

Subject : \_\_\_\_\_

Comments : \_\_\_\_\_

(Note: If you cannot read the numbers in the above image, reload the page to generate a new one.)

Figure 4 : CONTACT US



### Archives

#### How to Charge your iPod using an Onion

Posted By: Rakshit Khare on 2008-04-23

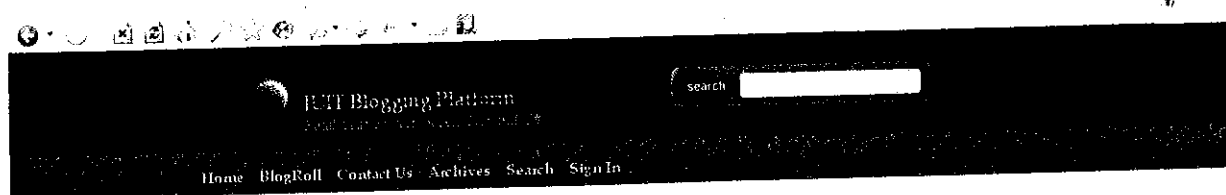
Strange, right? Recently, I found a weird video tutorial on gametraders that claims to do so. Really, it is a bit weird. Although, I haven't tried it yet but it looks very convincing. If anybody among you ever tries this, please let us know about that. It will be good to know if it works the way intended to be. So, let's start with the requirements for charging your iPod with an Onion. Requirements: 1 White onion 2 cups of Gatorade or any other energy drink Screwdriver iPod and USB cable Procedure: Make 2 holes on either side of the onion. Immerse that onion in a mug filled with 2 cups of Gatorade for about 30 mins. Wipe out excess fluid from the onion. Plug your USB cable (attached to iPod on the other end) into the onion. That is it.

### Comments

National Grammar Day- March 4, 2008

Local Intranet

Figure 5 : ARCHIVES



Admin Login

Username: **admin**

Password: ••••

☐ Remember Me

Sign in Reset

Local Intranet

Figure 6 : SIGN IN



**JUT Blogging Platform**  
Find out what's new on the Internet

Home BlogRoll Settings Manage Sign Out!

Blog Name: Rakshit Khare Blog- Aarambh  
 Blog URL: <http://www.rakshitk.com>  
 Blog Name: Rahul Sharma  
 Blog URL: <http://www.writingof.blogspot.com>  
 Blog Name: Madhur Kapoor  
 Blog URL: <http://www.whoismadur.com>  
 Blog Name: Ankit Mahanot Blog- Videos and Trailers  
 Blog URL: <http://www.video-trailers.com>

**Add Blog :**  
 Blog Name-  
 Blog URL-

Figure 7 : ADMIN BLOGROLL

**JUT Blogging Platform**  
Find out what's new on the Internet

Home BlogRoll Settings Manage Sign Out!

Name of the Blog :  
 Blog Description :  
 Blog Owner :  
 Email :  
 Blog Owner Profile:  
 Profile Picture :

Figure 8 : SETTINGS

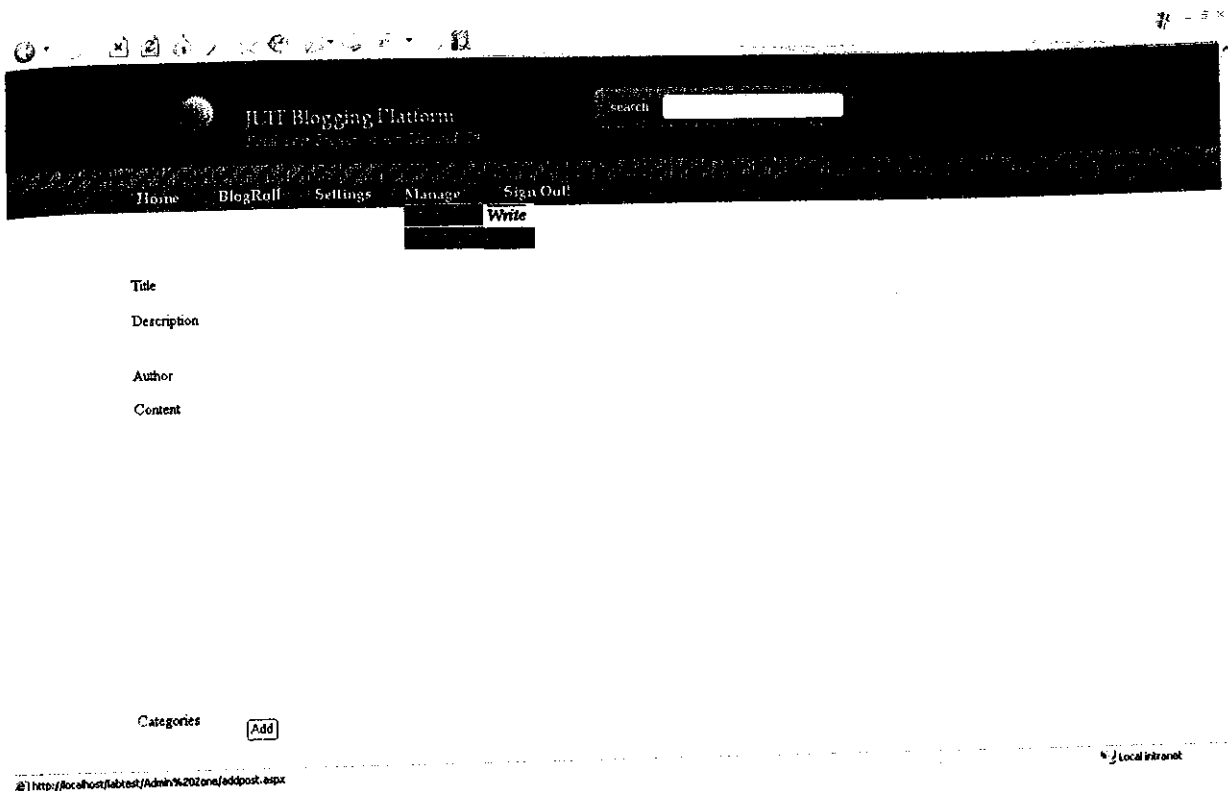


Figure 9 : MANAGE

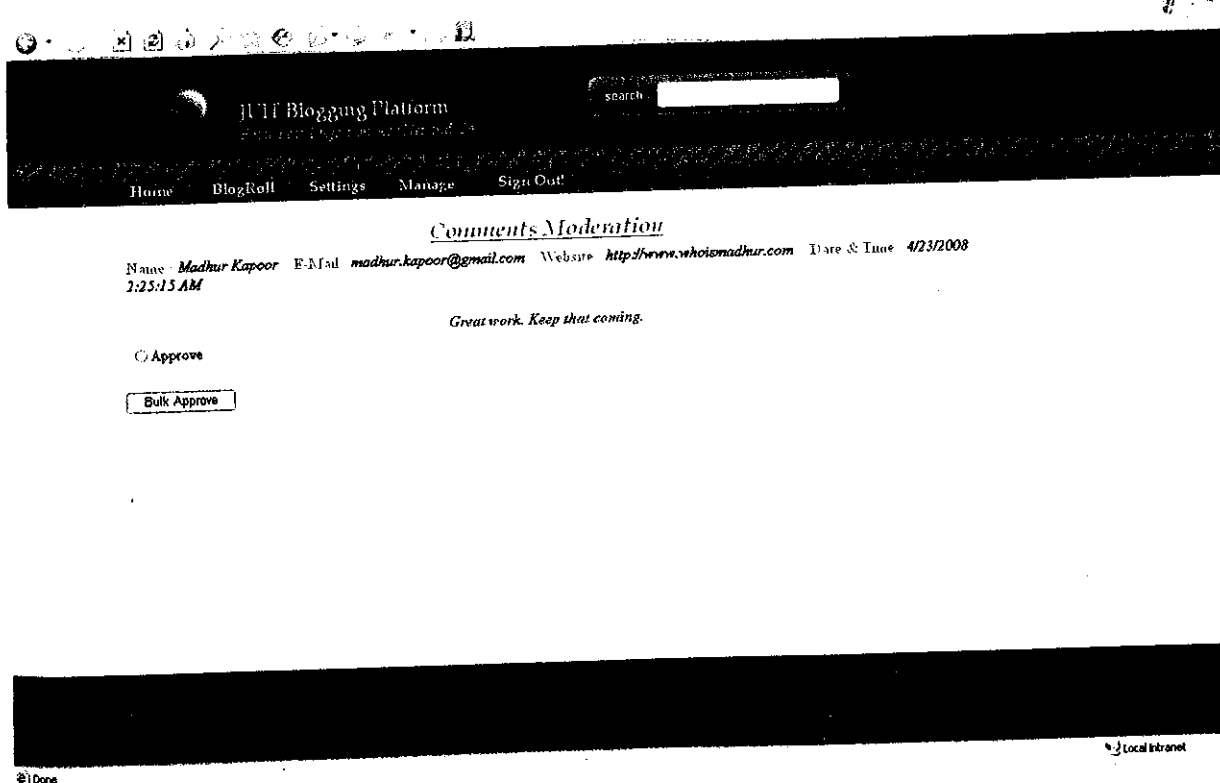


Figure 10 : COMMENT MODERATION



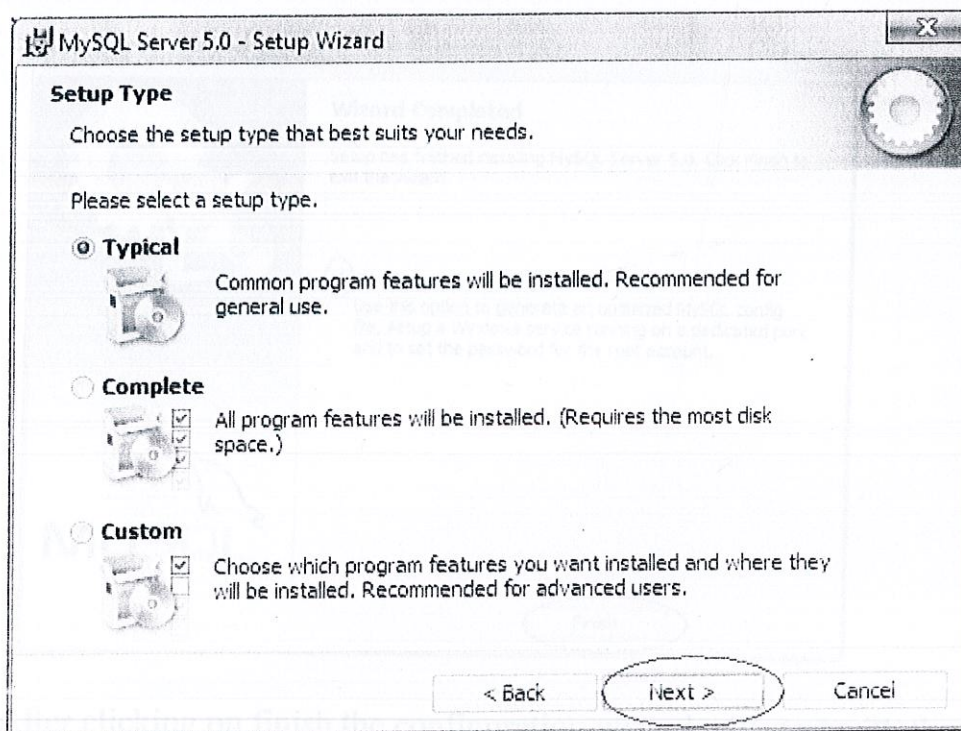
# INSTALLATION GUIDE

## SOFTWARES TO BE INSTALLED:-

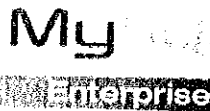
1. MYSQL - 5.0.45 ( **Ctrl + Click** to follow the link)
2. MYSQL-CONNECTOR-NET - 1.0.9
3. MYSQL-GUI-TOOL 5.0

## **1. How to Install MySQL-5.0.45**

**STEP 1:** Click on the setup icon to start installation process and click on next the following screen will come.



**STEP 2:** Click on the next button selected in the above screenshot. The following screen will be displayed click on next to proceed.



A MySQL Enterprise subscription is the most comprehensive offering of MySQL database software, services and support to ensure your business achieves the highest levels of reliability, security and uptime.

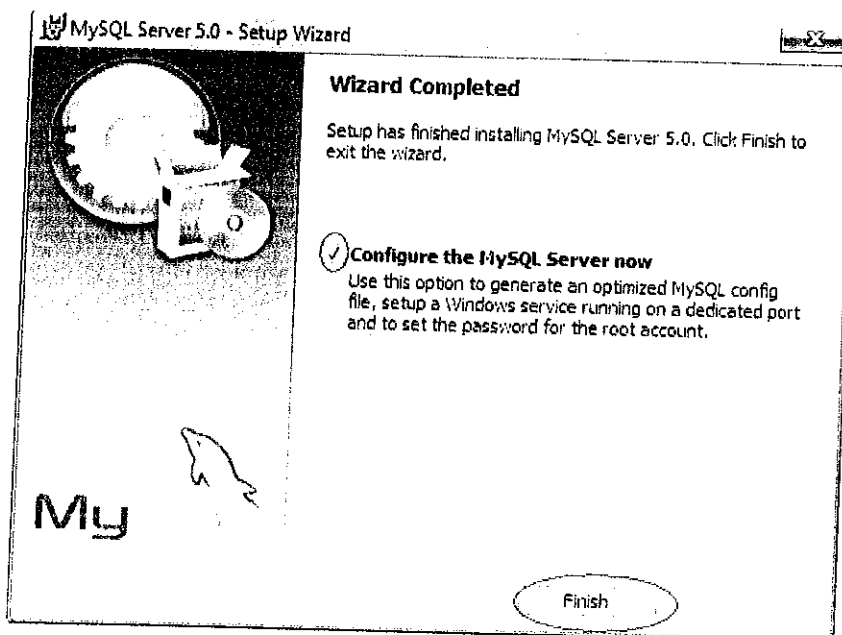
An Enterprise Subscription Includes:

1. The MySQL Enterprise Server - The most reliable, secure, and up-to-date version of the world's most popular open source database.
2. The MySQL Monitoring and Advisory Service - An automated virtual database assistant.
3. MySQL Production Support - Technical and consultative support when you need it, along with service packs, hot-fixes and more.

More ...

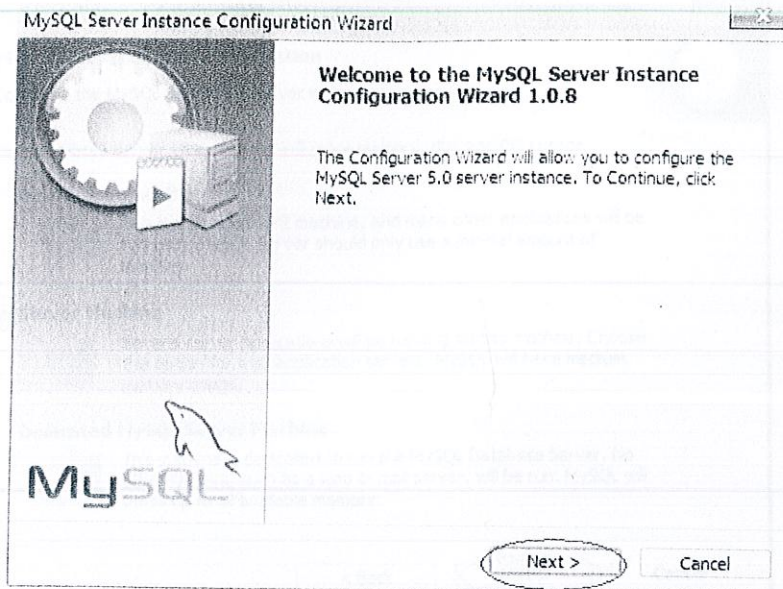
Next >

**STEP 3:** Click on next button till you get this screen given below, here select the configure check button and click on finish button marked in the Screenshot.

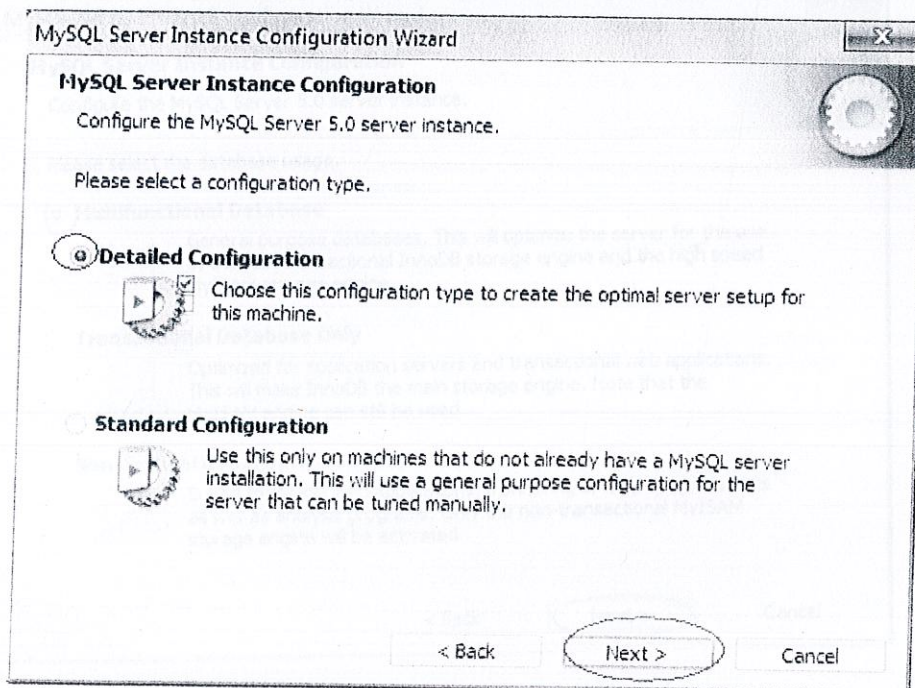


**STEP 4:** After clicking on finish the configuration wizard will start with the following screen, click on next button selected in screenshot to proceed.





**STEP 5:** Select detailed configuration round box and click on the next button as shown in the screenshot below.



**STEP 6:** In the next screen select server machine option and click on next button.

# MySQL Server Instance Configuration Wizard

## MySQL Server Instance Configuration

Configure the MySQL Server 5.0 server instance.

Please select a server type. This will influence memory, disk and CPU usage.

### Developer Machine



This is a development machine, and many other applications will be run on it. MySQL Server should only use a minimal amount of memory.

### ☒ Server Machine



Several server applications will be running on this machine. Choose this option for web/application servers. MySQL will have medium memory usage.

### Dedicated MySQL Server Machine



This machine is dedicated to run the MySQL Database Server. No other servers, such as a web or mail server, will be run. MySQL will utilize up to all available memory.

< Back

Next >

Cancel

**STEP 7:** Select multifunctional database option and click on the next button as shown below in the screenshot.

# MySQL Server Instance Configuration Wizard

## MySQL Server Instance Configuration

Configure the MySQL Server 5.0 server instance.

Please select the database usage.

### ☒ Multifunctional Database



General purpose databases. This will optimize the server for the use of the fast transactional InnoDB storage engine and the high speed MyISAM storage engine.

### Transactional Database Only



Optimized for application servers and transactional web applications. This will make InnoDB the main storage engine. Note that the MyISAM engine can still be used.

### Non-Transactional Database Only



Suited for simple web applications, monitoring or logging applications as well as analysis programs. Only the non-transactional MyISAM storage engine will be activated.

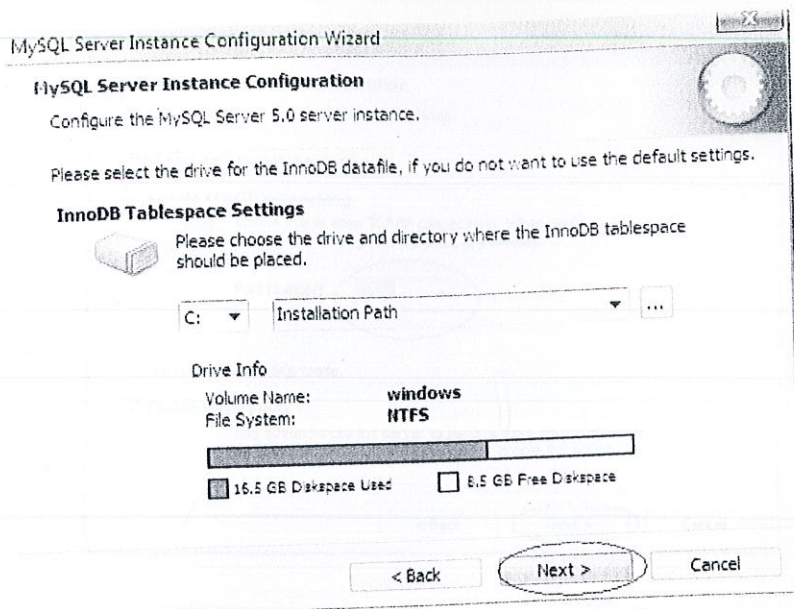
< Back

Next >

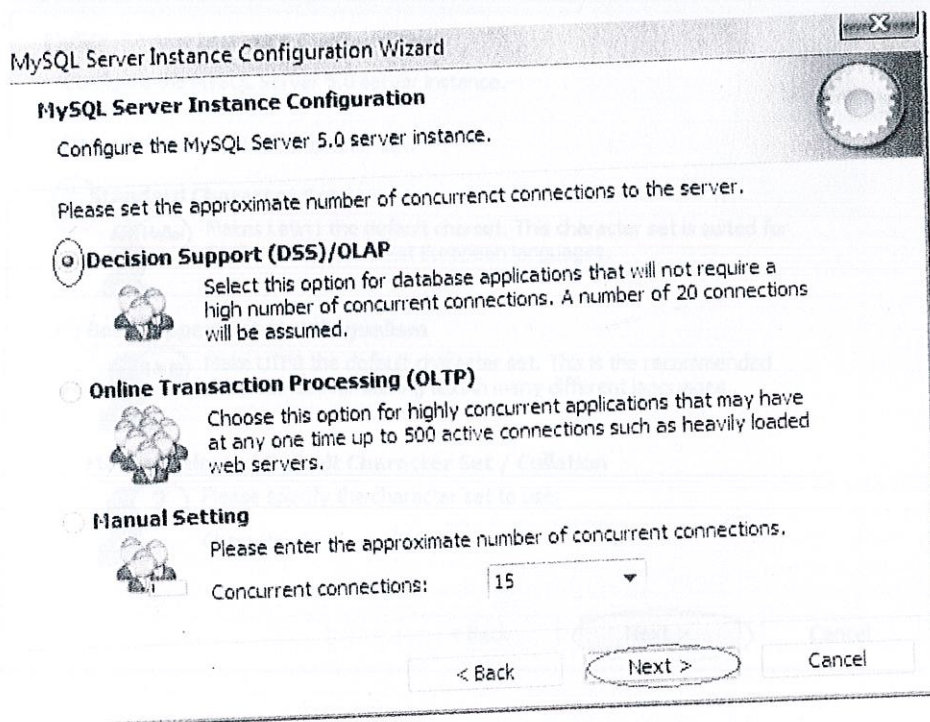
Cancel

**STEP 8:** Click on the next button as shown below.



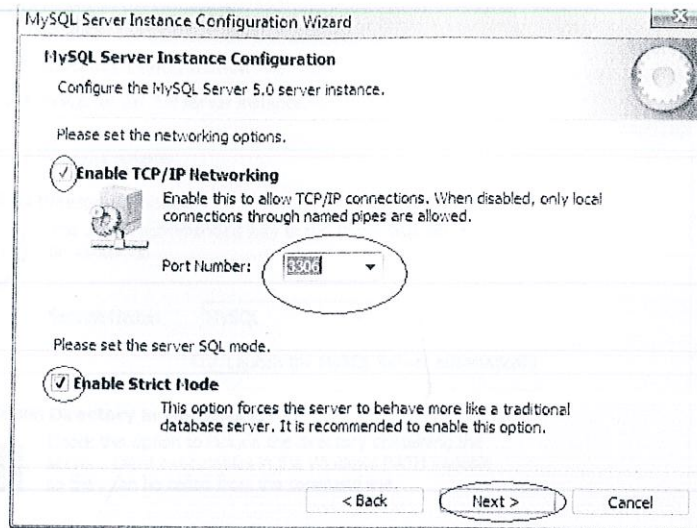


**STEP 9:** Select Decision Support (DSS)/OLAP option and click next as shown below.

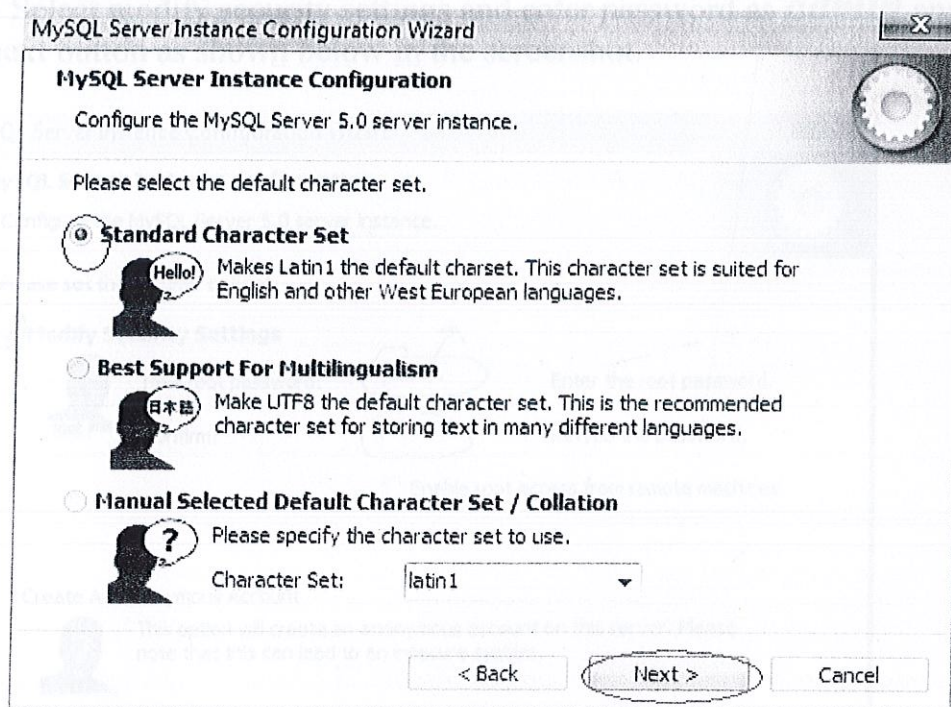


**STEP 10:** Select the enable TCP/IP Networking and Enable Strict Mode option and click on next button as shown below.



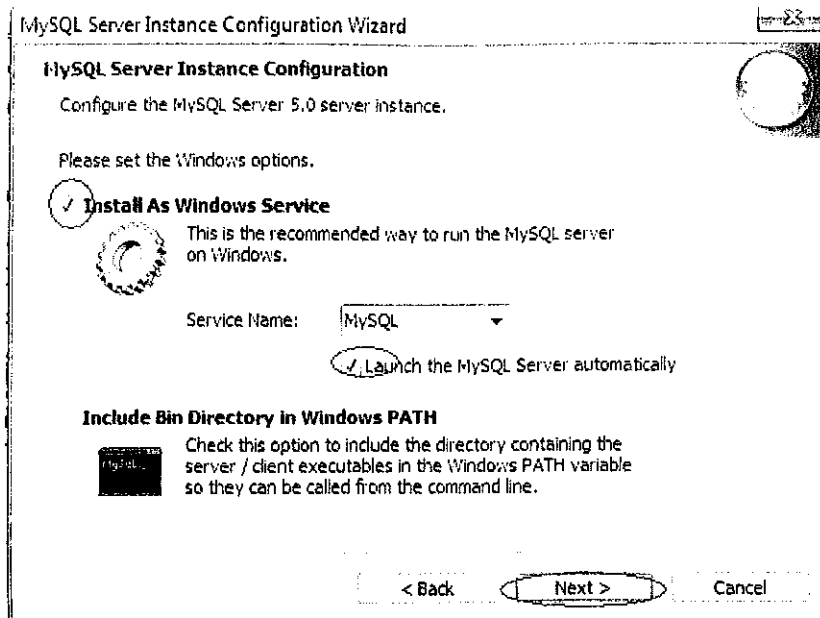


**STEP 11:** Select Standard Character Set option and click on next button as shown below in the screen shot.

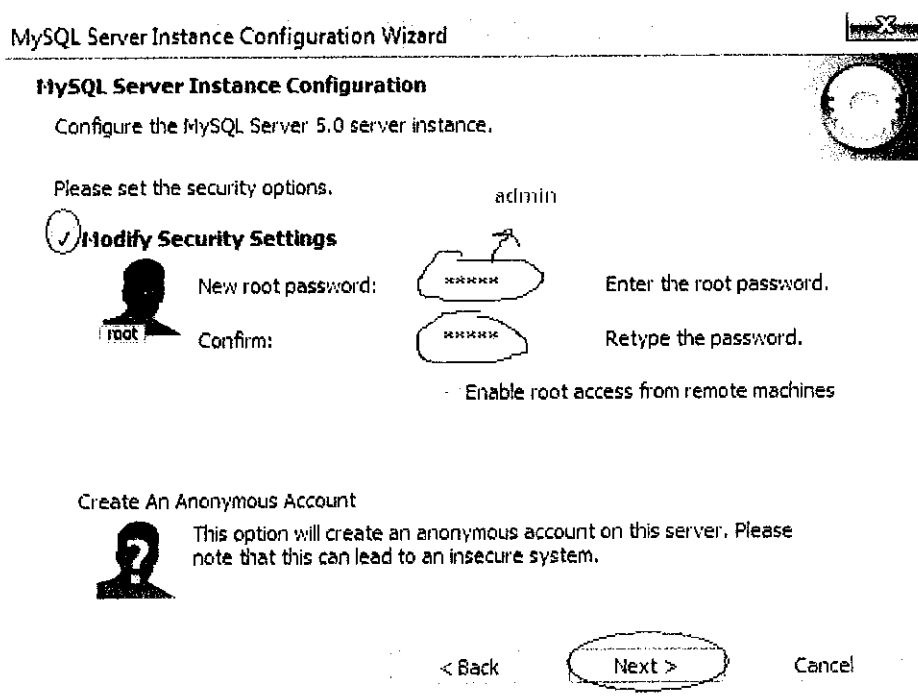


**STEP 12:** Select Install as Windows Service and Launch the MYSQL server automatically option and click on next as shown below.

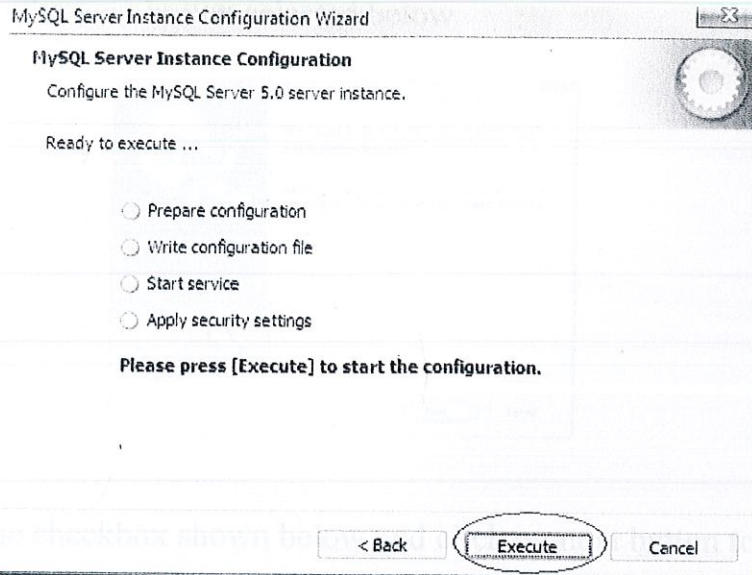




**STEP 13:** Select modify security settings and enter password as admin and click on next button as shown below in the screenshot.



**STEP 14:** Select on the Execute button as shown below.

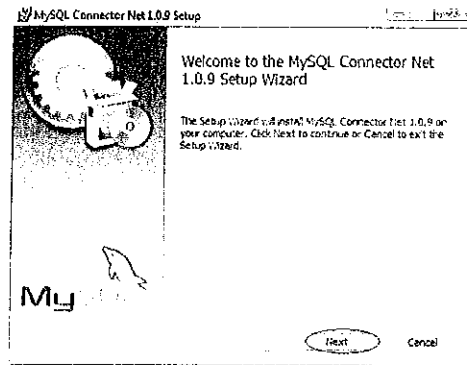


## 2. INSTALLING MYSQL-CONNECTOR-NET

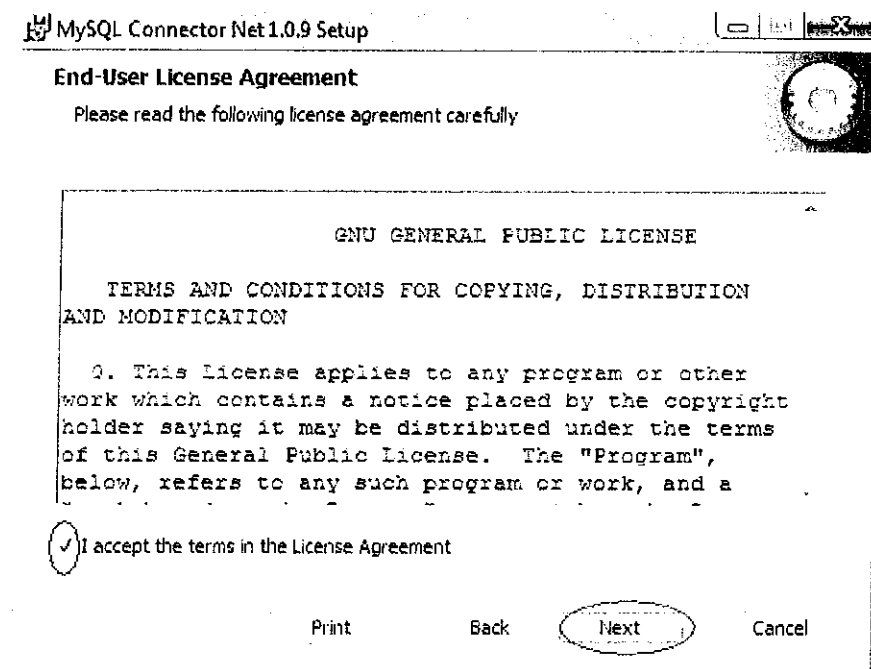
Start the setup by running .net connector setup file and the screen will appear as shown below



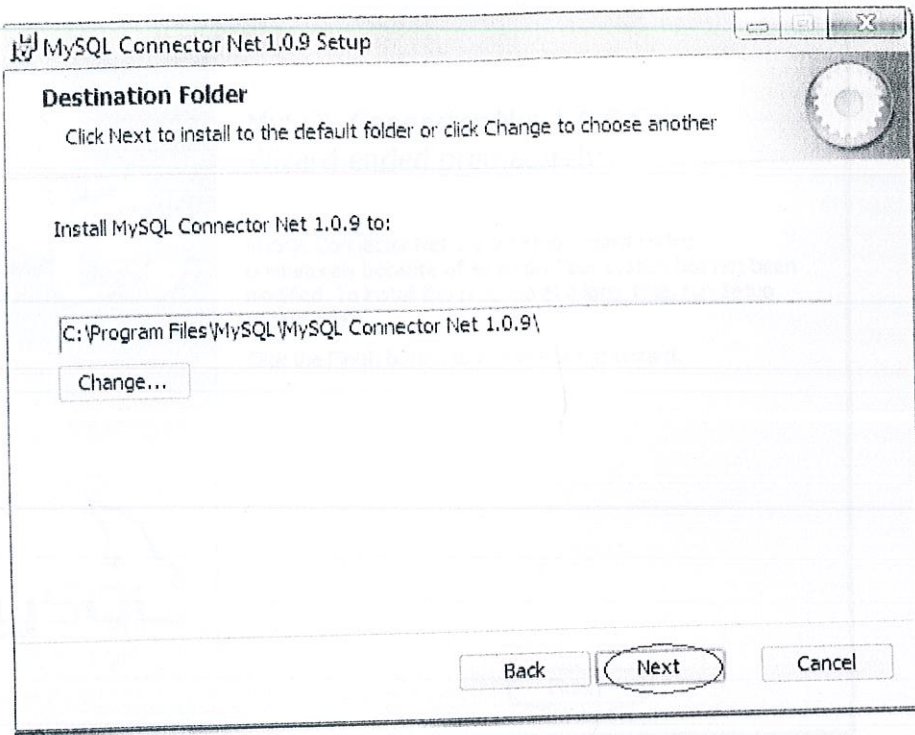
*Step 1:* click on the next button selected below



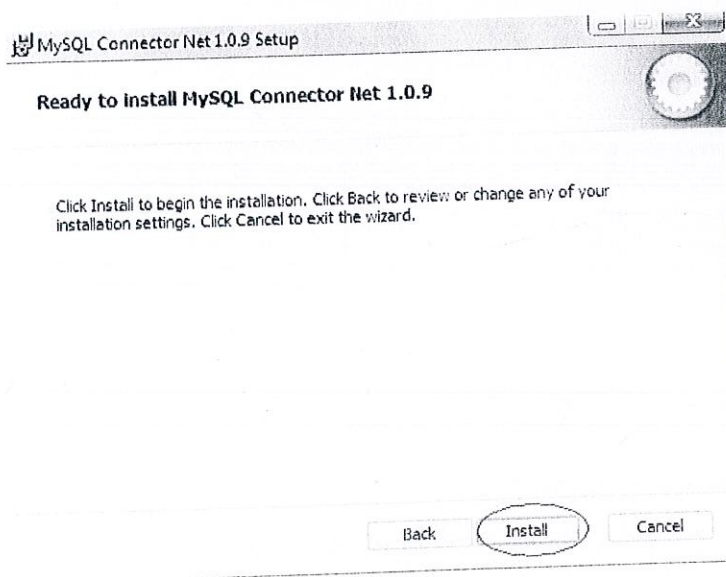
*Step 2:* select the checkbox shown below and click on next button to proceed.



*Step 3:* click on next button as shown to below to proceed.

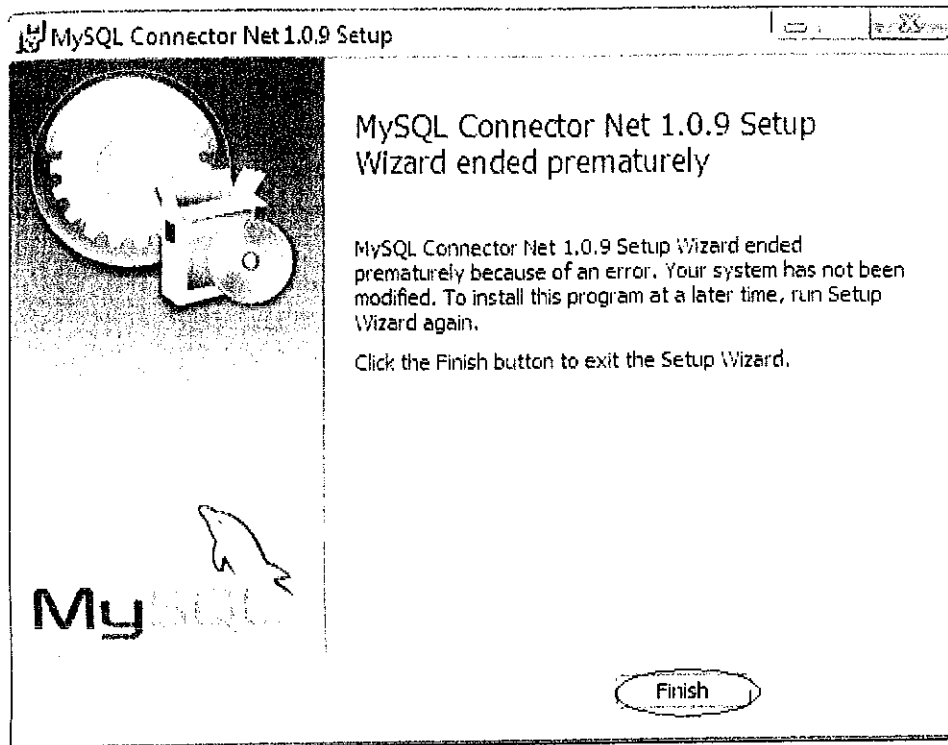


**Step 4:** click on the install button to install the software.



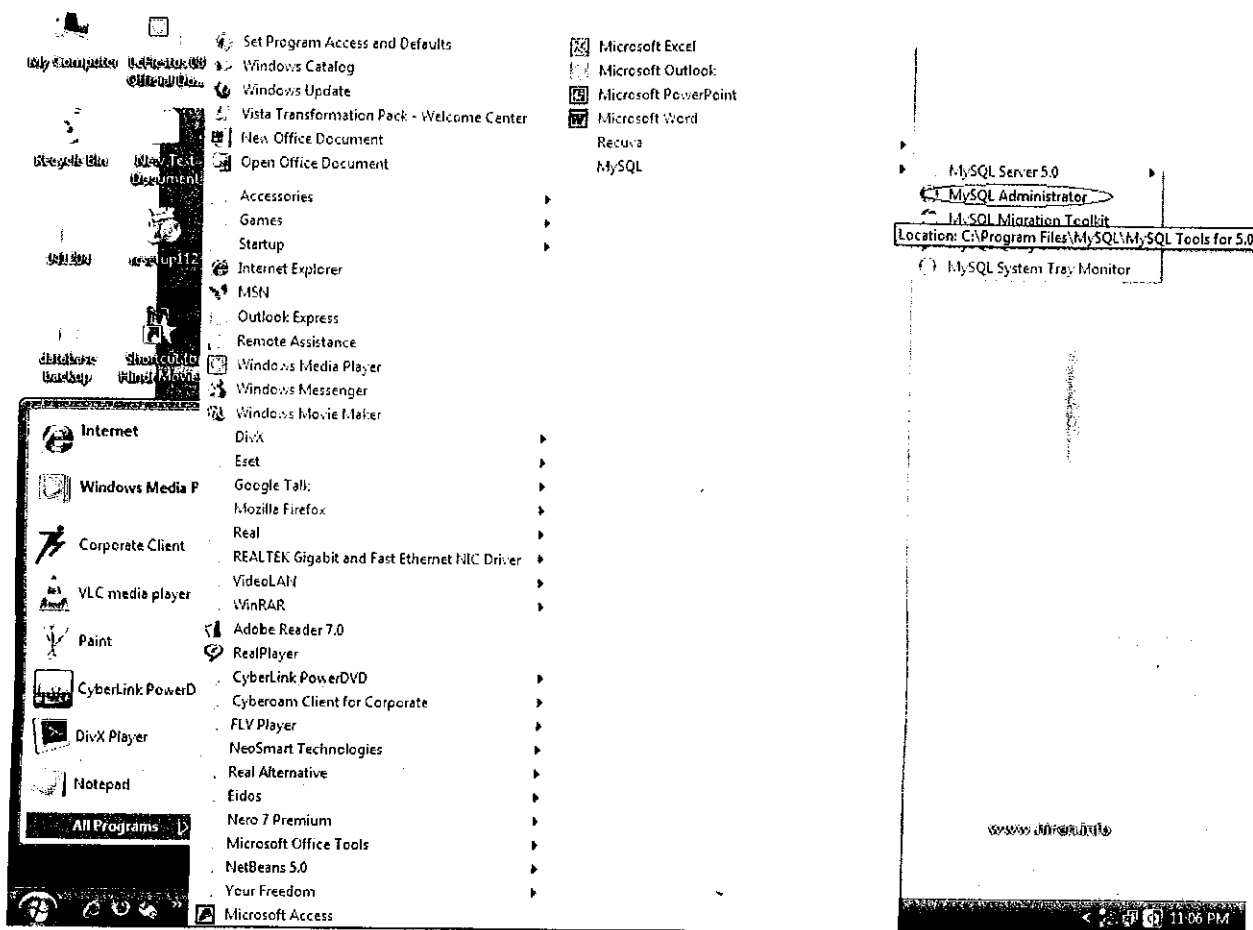
**Step 5:** click on the finish button to complete the setup.





### 3. INSTALLING MYSQL-GUI-TOOL 5.0

*Step 1:* open the mysql administrator from the start menu as shown below



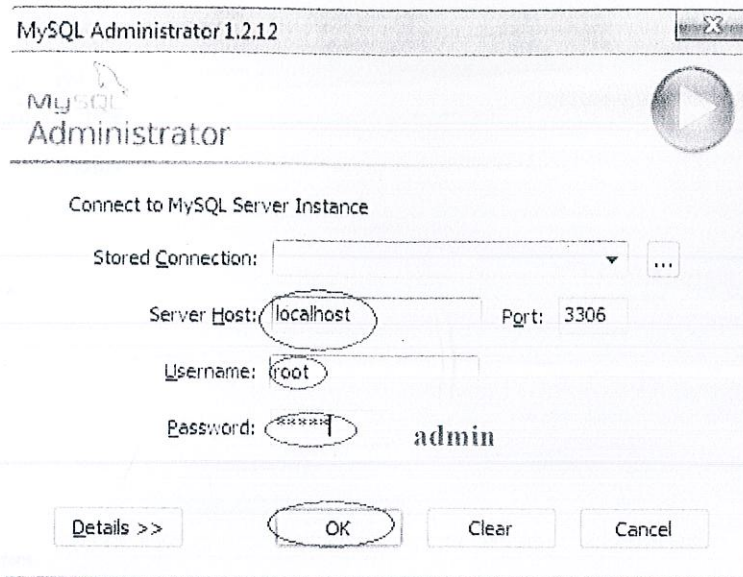
**Step 2: Server Host: localhost**

**Username: root**

**Password: admin**

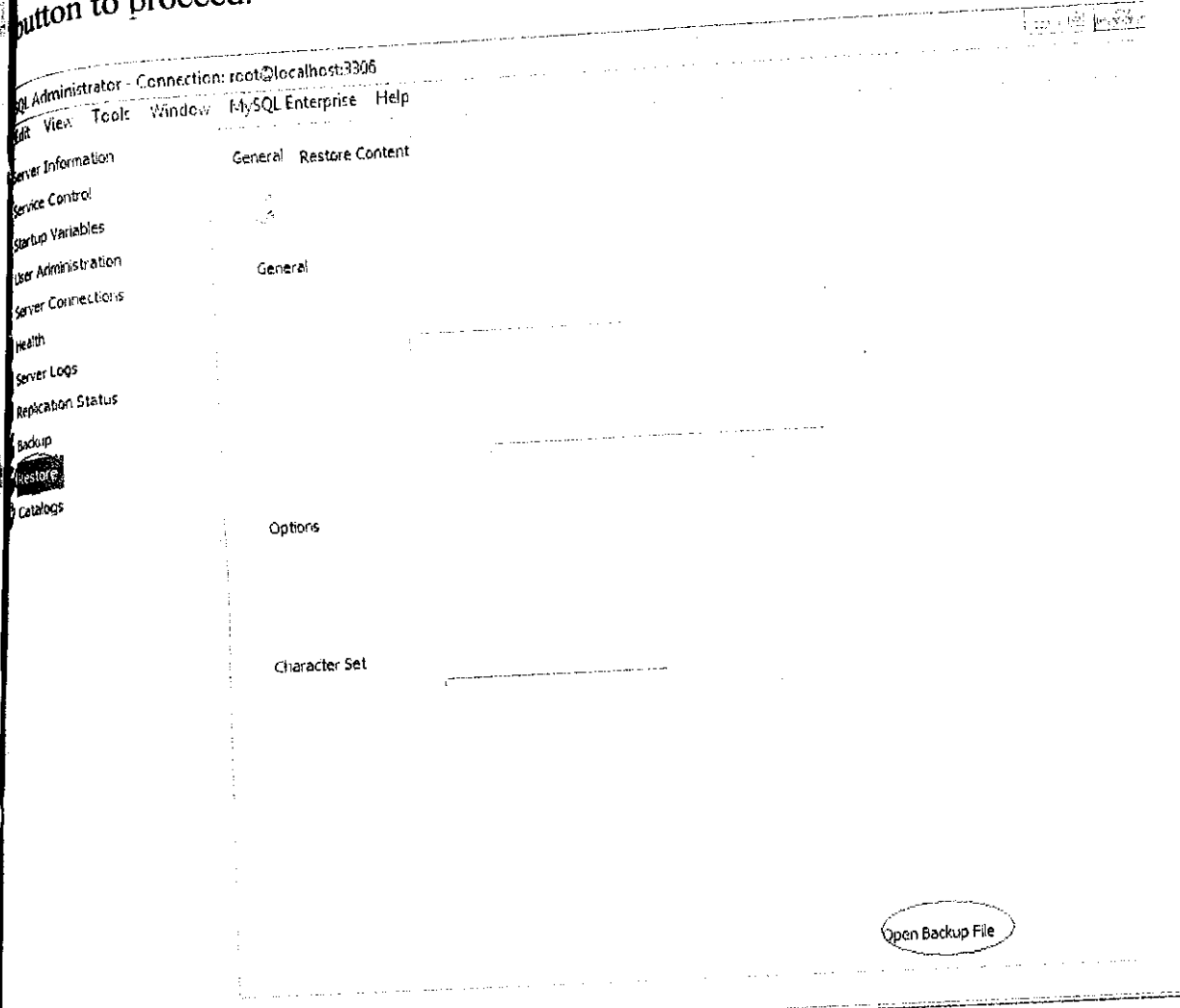
As shown below and click on the ok button to proceed.





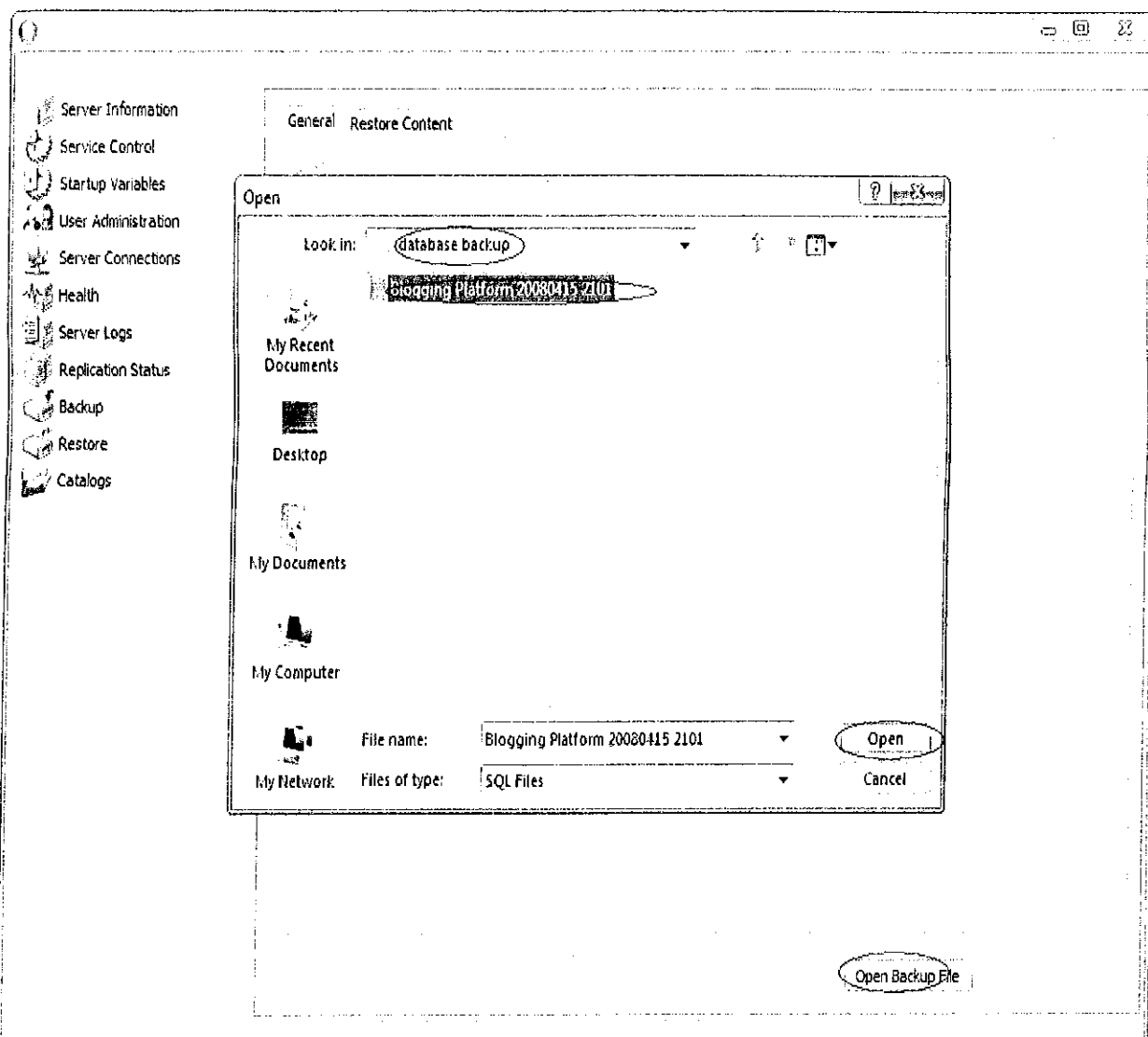
*Step 3:* The screen below will appear after clicking ok button and the server information will be shown as in the screenshot.

4: click on the restore option as shown below and select the restore backup button to proceed.





**Step 5:** After clicking on the open backup file button the screen will open to select the backup file, select the backup file named blogging platform and click on open button as show below.



**Step 6:** After you have selected the file its path will be shown in file to restore as marked below, then click the start restore button to proceed.

- User Administration
- Server Connections
- Health
- Server Logs
- Replication Status
- Backup
- Restore**
- Catalogs

#### General

File to restore: C:\Documents and Settings\yahul\Desktop\database backup\B1c Name of the backup file that should be restored.

Backup Type: SQL Files Type of backup file.

#### Target Schema:

☒ Original schema

☐ Another schema

information\_schema

#### Options

☐ Ignore Errors

☐ Create database(s) if they don't exist

#### Character Set

File Charset: utf8

If you are importing a SQL file that has not been created with MySQL Administrator, you have to choose the correct character set of the file. If you have created the backup with MySQL Administrator the file was written in UTF-8.

Open Backup File

Start Restore

**Step 7:** After the restore is complete click on the catalog option in the left as shown and the tables will be displayed select the tables individually and press the refresh button as shown below

File Edit View Tools Window MySQL Enterprise Help

Server Information  
Service Control  
Startup Variables  
User Administration  
Server Connections  
Health  
Server Logs  
Replication Status  
Backup  
Restore  
**Catalogs**

Schemas  
information\_schema  
mysql  
test

Schema Tables Schema Indices Views Stored procedures

**blogdb**  
All tables of the blogdb schema

| Table Name   | Engine | Rows | Data length | Index length | Update time |
|--------------|--------|------|-------------|--------------|-------------|
| blogdetail   | InnoDB | 1    | 16 kB       | 0 B          |             |
| blogroll     | InnoDB | 3    | 16 kB       | 0 B          |             |
| categories   | InnoDB | 0    | 16 kB       | 0 B          |             |
| comments     | InnoDB | 0    | 16 kB       | 0 B          |             |
| logininfo    | InnoDB | 1    | 16 kB       | 0 B          |             |
| postcategory | InnoDB | 0    | 16 kB       | 0 B          |             |
| posts        | InnoDB | 4    | 16 kB       | 0 B          |             |

Details >>

Create Table Edit Table Maintenance **Refre**

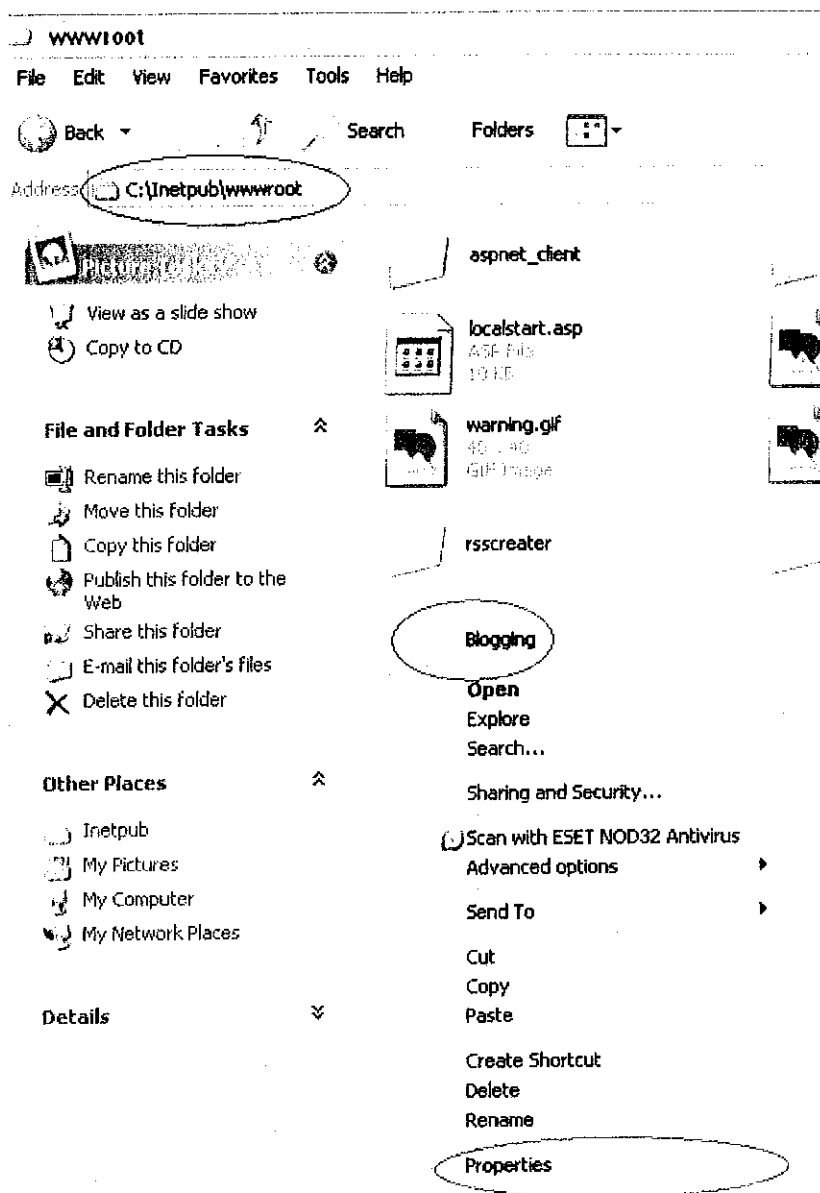


## Test Run

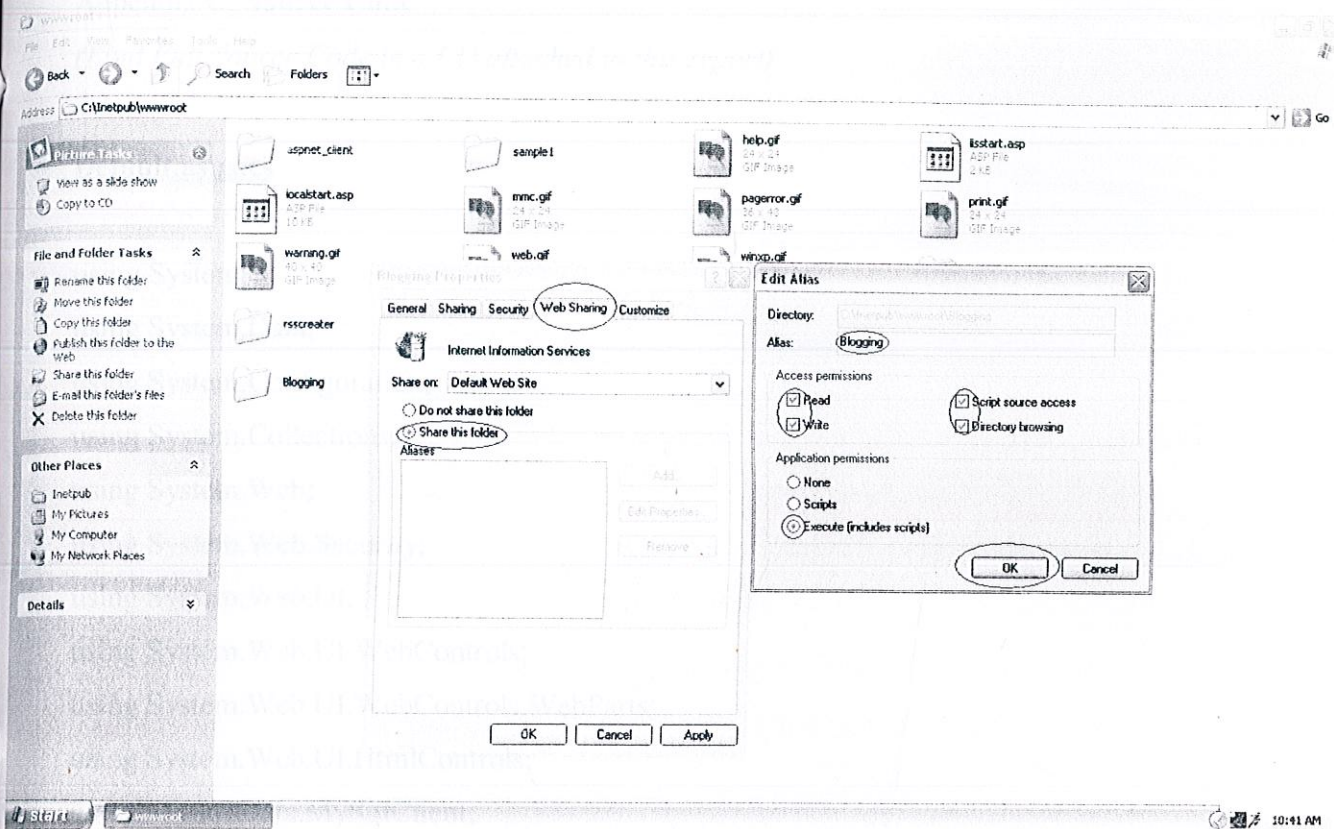
*Step 1:* Install IIS.

Copy the folder **Blogging** into the ROOT directory (wwwroot) of INETPUB.

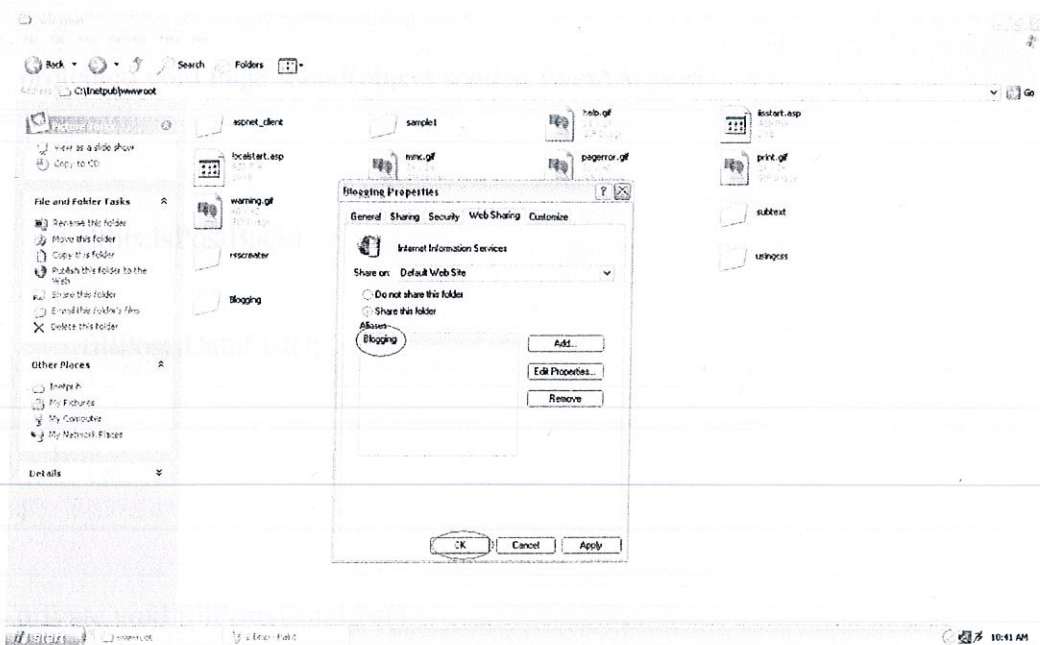
*Step 2:* Right Click the folder **Blogging** and click on "Properties".



Step 3: Select tab "Web Sharing" and perform the tasks shown in the figure below.



Step 4: Click OK to continue.



## Appendix C: Source Code

*(Find Full Source Code in a CD attached to this report)*

### Default.aspx.cs

```
using System;
using System.Data;
using System.Configuration;
using System.Collections;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;
using MySql.Data.MySqlClient;

public partial class Defaultnew : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!this.IsPostBack)
        {
            fillPostsDataList();
        }
    }

    private void fillPostsDataList()
    {
```



```

string str =
System.Configuration.ConfigurationManager.ConnectionStrings["abcd"].ToString();
    MySqlConnection con = new MySqlConnection(str);
    MySqlDataAdapter da = new MySqlDataAdapter("select
postid,title,content,author,datecreated,datetimecreated, commentcount from posts order by
datetimecreated desc", con);
    MySqlDataAdapter da1 = new MySqlDataAdapter("select owner,ownerprofile,email from
blogdetail", con);
    MySqlDataAdapter da2 = new MySqlDataAdapter("select postid,title,commentcount from
posts where commentcount>1 order by commentcount desc", con);
    MySqlDataAdapter da3 = new MySqlDataAdapter(" select postid,author,commentdetail
from comments order by commentdate desc", con);
    DataTable dt = new DataTable();
    DataTable dt1 = new DataTable();
    DataTable dt2 = new DataTable();
    DataTable dt3 = new DataTable();
    try
    {
        con.Open();
        da.Fill(dt);
        //lblerror.Text = dt[0]["postid"];
        da1.Fill(dt1);
        da2.Fill(dt2);
        da3.Fill(dt3);
        grdviewpost.DataSource = dt;
        grdviewpposts.DataSource = dt2;
        grdviewrposts.DataSource = dt;
        grdviewauthor.DataSource = dt1;
        grdviewrcomments.DataSource = dt3;
        grdviewpost.DataBind();
        grdviewpposts.DataBind();
        grdviewrposts.DataBind();
    }

```

```

        grdviewauthor.DataBind();
        grdviewrcomments.DataBind();
    }
    catch (Exception ex)
    {
        lblerror.Text = ex.ToString();
    }
    finally
    {
        con.Close();
    }
}

```

```

protected void lnkcomments_Command(object sender, CommandEventArgs e)
{
    Session["pid"] = e.CommandArgument.ToString();
    Response.Redirect("comments.aspx");
}

```

```

protected void grdviewpost_SelectedIndexChanged(object sender, EventArgs e)
{
}

```

```

protected void grdviewpost_PageIndexChanged(object sender, EventArgs e)
{
}

```

```

protected void grdviewpost_PageIndexChanging(object sender, GridViewPageEventArgs e)
{
    grdviewpost.PageIndex = e.NewPageIndex;
}

```

```

        fillPostsDataList();

    }

    protected void grdviewpposts_SelectedIndexChanged(object sender, EventArgs e)
    {

    }

    protected void grdviewpposts_PageIndexChanging(object sender, GridViewPageEventArgs e)
    {
        grdviewpposts.PageIndex = e.NewPageIndex;
        fillPostsDataList();
    }

    protected void grdviewrposts_PageIndexChanging(object sender, GridViewPageEventArgs e)
    {
        grdviewrposts.PageIndex = e.NewPageIndex;
        fillPostsDataList();
    }

    protected void grdviewrcomments_PageIndexChanging(object sender,
GridViewPageEventArgs e)
    {
        grdviewrcomments.PageIndex = e.NewPageIndex;
        fillPostsDataList();
    }
}

```

### **Comments.aspx.cs**

```

using System;
using System.Data;
using System.Configuration;
using System.Collections;
using System.Web;
using System.Web.Security;

```



```

using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;
using MySql.Data.MySqlClient;

namespace Captcha
{

    public partial class comments1 : System.Web.UI.Page
    {
        private Random random = new Random();

        protected void Page_Load(object sender, EventArgs e)
        {
            if (!this.IsPostBack)
            {
                fillPostsDataList();
            }
            else
            {
                // On a postback, check the user input.
                if (this.CodeNumberTextBox.Text == this.Session["CaptchaImageText"].ToString())
                {
                    // Display an informational message.
                    this.MessageLabel.CssClass = "info";
                    this.MessageLabel.Text = "Correct!";
                    CodeNumberTextBox.Focus();
                }
                else
                {

```

```

        // Display an error message.
        this.MessageLabel.CssClass = "error";
        this.MessageLabel.Text = "ERROR: Incorrect, try again.";
        CodeNumberTextBox.Focus();
        // Clear the input and create a new random code.
        this.CodeNumberTextBox.Text = "";
        this.Session["CaptchaImageText"] = GenerateRandomCode();
    }
}

```

```
private void fillPostsDataList()
```

```

{
    // if (Session["commented"] != null)
    // {
    //     String name = Session["commented"].ToString();
    //     lblmessage.Text = name + ", your comment is awaiting moderation";
    // }
    // else
    // {
    //     lblmessage.Text = "value null";
    // }
}

```

```

this.Session["CaptchaImageText"] = GenerateRandomCode();

```

```

String postid = Session["pid"].ToString();

```

```

int pid = Convert.ToInt32(postid);

```

```

string str =

```

```

System.Configuration.ConfigurationManager.ConnectionStrings["abcd"].ToString();

```

```

MySQLConnection con = new MySQLConnection(str);

```

```

MySQLDataAdapter da = new MySQLDataAdapter("select

```

```

postid,title,content,author,datecreated from posts where postid="+ pid.ToString(), con);

```



```
MySQLDataAdapter da1 = new MySQLDataAdapter("select author, website,commentdetail  
from comments where postid=" + pid.ToString() + "&& status=1", con);
```

```
// MySQLDataAdapter da1 = new MySQLDataAdapter("select owner,ownerprofile,email  
from blogdetail", con);
```

```
DataTable dt = new DataTable();
```

```
DataTable dt1 = new DataTable();
```

```
try
```

```
{
```

```
    con.Open();
```

```
    da.Fill(dt);
```

```
    da1.Fill(dt1);
```

```
    grdviewpost.DataSource = dt;
```

```
    grdviewcomments.DataSource = dt1;
```

```
    grdviewpost.DataBind();
```

```
    grdviewcomments.DataBind();
```

```
}
```

```
catch (Exception ex)
```

```
{
```

```
    lblerror.Text = ex.ToString();
```

```
}
```

```
finally
```

```
{
```

```
    con.Close();
```

```
}
```

```
}
```

```
private string GenerateRandomCode()
```

```
{
```

```
    string s = "";
```

```
    for (int i = 0; i < 6; i++)
```

```
        s = String.Concat(s, this.random.Next(10).ToString());
```

```
    return s;
```



```

    }

    #region Web Form Designer generated code
    override protected void OnInit(EventArgs e)
    {
        //
        // CODEGEN: This call is required by the ASP.NET Web Form Designer.
        //
        InitializeComponent();
        base.OnInit(e);
    }

    /// <summary>
    /// Required method for Designer support - do not modify
    /// the contents of this method with the code editor.
    /// </summary>
    private void InitializeComponent()
    {
        this.Load += new System.EventHandler(this.Page_Load);
    }
    #endregion

    protected void btnsubmit_Click(object sender, EventArgs e)
    {

        String[] sub = new string[10];
        if (MessageLabel.Text == "Correct!")
        {
            int count = txtemail.Text.IndexOf("@");
            if (count == 1)
            {

```

```

String postid = Session["pid"].ToString();
int pid = Convert.ToInt32(postid);
string str =
System.Configuration.ConfigurationManager.ConnectionStrings["abcd"].ToString();
MySQLConnection con = new MySQLConnection(str);
// MySQLDataAdapter dasubscribe = new MySQLDataAdapter("select subscribe,
postid from posts where postid=" + pid, con);
// DataTable dt = new DataTable();
//MySQLConnection con1 = new MySQLConnection(str);
try
{
    con.Open();
    // dasubscribe.Fill(dt);
    // foreach (DataRow dr in dt.Rows)
    // {
        // sub = dr["subscribe"].ToString();
        // }
        // for (i1 = 0; i < sub.GetLength; i++)
        // {
            // StringArrayConverter newsub = sub[i];
            // }
}
finally
{
    con.Close();
}

DateTime currtime1 = DateTime.Now;
String currtime = currtime1.ToString("yyyy-MM-dd hh-mm-ss");
MySQLCommand cmdInsert = new MySQLCommand("Insert into
comments(postid,commentdate,author,email,website,commentdetail) values('" + pid + "','" +

```

```
currtime + "','" + txtname.Text + "','" + txtemail.Text + "','" + txtwebsite.Text + "','" +  
txtcomment.Text + ")", con);
```

```
        MySqlCommand cmdupdate = new MySqlCommand("update posts set  
commentcount=commentcount+1 where postid=" + pid, con);
```

```
        int i = cmdInsert.ExecuteNonQuery();
```

```
        int i1 = cmdupdate.ExecuteNonQuery();
```

```
        if (i == 1 && i1 == 1)
```

```
{
```

```
    Session["commented"] = txtname.Text;
```

```
    txtname.Text = "";
```

```
    txtemail.Text = "";
```

```
    txtwebsite.Text = "";
```

```
    txtcomment.Text = "";
```

```
}
```

```
}
```

```
catch (Exception ex)
```

```
{
```

```
    lblerror.Text = ex.ToString();
```

```
}
```

```
finally
```

```
{
```

```
    con.Close();
```

```
    Response.Redirect("default.aspx");
```

```
}
```

```
}
```

```
else
```

```
{
```



```

        lblerrmsg.Text = "Invalid E-Mail address!!!";
    }

}
else
{
    MessageLabel.CssClass = "error";
    MessageLabel.Text = "ERROR: Incorrect, try again.";
    CodeNumberTextBox.Focus();
}
}

```

```

protected void Button1_Click(object sender, EventArgs e)
{

}

}
}

```

### **Blogroll.aspx.cs**

```

using System;
using System.Data;
using System.Configuration;
using System.Collections;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;
using MySql.Data.MySqlClient;

```

```

public partial class blogrollnew : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {

        string str =
System.Configuration.ConfigurationManager.ConnectionStrings["abcd"].ToString();
        MySqlConnection con = new MySqlConnection(str);
        MySqlDataAdapter da = new MySqlDataAdapter("select name,url from blogroll", con);
        DataTable dt = new DataTable();
        try
        {
            con.Open();
            da.Fill(dt);
            grdviewblogroll.DataSource = dt;
            grdviewblogroll.DataBind();
        }
        catch (Exception ex)
        {
            lblerror.Text = ex.ToString();
        }
        finally
        {
            con.Close();
        }
    }
}

```

**Contactus.aspx.cs**

```
using System;
using System.Data;
using System.Configuration;
using System.Collections;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;
using System.Web.Mail;
```

```
namespace Captcha
```

```
{
```

```
    public partial class Contactusnew : System.Web.UI.Page
```

```
    {
```

```
        private Random random = new Random();
```

```
        protected void Page_Load(object sender, EventArgs e)
```

```
        {
```

```
            if (!this.IsPostBack)
```

```
            {
```

```
                this.Session["CaptchaImageText"] = GenerateRandomCode();
```

```
            }
```

```
        else
```

```
        {
```

```
            // On a postback, check the user input.
```

```
            if (this.CodeNumberTextBox.Text == this.Session["CaptchaImageText"].ToString())
```

```
            {
```

```
                // Display an informational message.
```

```
                this.lblmessage.CssClass = "info";
```

```
                this.lblmessage.Text = "Correct!";
```

```
            }
```



```

else
{
    // Display an error message.
    this.lblmessage.CssClass = "error";
    this.lblmessage.Text = "ERROR: Incorrect, try again.";

    // Clear the input and create a new random code.
    this.CodeNumberTextBox.Text = "";
    this.Session["CaptchaImageText"] = GenerateRandomCode();
}
}

protected void btnReset_Click(object sender, EventArgs e)
{
    txtName.Text = "";
    txtEmail.Text = "";
    txtSubject.Text = "";
    txtComments.Text = "";
}

protected void btnSubmit_Click(object sender, EventArgs e)
{
    /*lblvalidate.Text = "";
    lblerror.Text = "";

    int num1 = Convert.ToInt32(lblfirstint.Text);
    int num2 = Convert.ToInt32(lblsecondint.Text);

    if (txtsum.Text != "")
    {
        int sum = num1 + num2;

```

```

int ans = Convert.ToInt32(txtsum.Text);
lblmsg.Text = sum.ToString();
lblvalidate.Text = ans.ToString();
if (sum == ans)
{
    /*

if (lblmessage.Text == "Correct!")
{
    MailMessage objEmail = new MailMessage();

    objEmail.To = "raks.khare@gmail.com";
    objEmail.From = "roxisayshi@gmail.com";
    objEmail.Subject = "sadf";
    objEmail.Body = "hi";
    objEmail.Priority = MailPriority.High;
    SmtpMail.SmtpServer = "lenovo";
    try
    {
        SmtpMail.Send(objEmail);
        Response.Write("Your Email has been sent sucessfully-Thank You");
    }
    catch (Exception exc)
    {

        Response.Write("Send failure: " + exc.ToString());
    }
}
else
{
    lblmessage.CssClass = "error";
    lblmessage.Text = "ERROR: Incorrect, try again.";
}

```

```

/* }
else
{
    lblerror.Text = " Value not entered correctly!!!";
}
}
else
{
    lblvalidate.Text = "Required Field cannot be left blank";
    txtsum.Focus();
}
*/
}

```

```

private string GenerateRandomCode()
{
    string s = "";
    for (int i = 0; i < 6; i++)
        s = String.Concat(s, this.random.Next(10).ToString());
    return s;
}

```

```

#region Web Form Designer generated code
override protected void OnInit(EventArgs e)
{
    //
    // CODEGEN: This call is required by the ASP.NET Web Form Designer.
    //
    InitializeComponent();
    base.OnInit(e);
}

```



```

/// <summary>
/// Required method for Designer support - do not modify
/// the contents of this method with the code editor.
/// </summary>
private void InitializeComponent()
{
    this.Load += new System.EventHandler(this.Page_Load);

}
#endregion

}
}

```

#### **Archives.aspx.cs**

```

using System;
using System.Data;
using System.Configuration;
using System.Collections;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;
using MySql.Data.MySqlClient;

public partial class archives : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        //Calendar1.SelectedDate = DateTime.Today;
    }
}

```

```
}
```

```
protected void grdviewarchive_SelectedIndexChanged(object sender, EventArgs e)
```

```
{
```

```
}
```

```
protected void lnkcomments_Command(object sender, CommandEventArgs e)
```

```
{
```

```
    Session["pid"] = e.CommandArgument.ToString();
```

```
    Response.Redirect("comments.aspx");
```

```
}
```

```
protected void Calendar1_SelectionChanged(object sender, EventArgs e)
```

```
{
```

```
    DateTime arcdate = Calendar1.SelectedDate;
```

```
    String newarcdate = arcdate.ToString("yyyy-MM-dd");
```

```
    lblerror.Text = newarcdate.ToString();
```

```
    string str =
```

```
    System.Configuration.ConfigurationManager.ConnectionStrings["abcd"].ToString();
```

```
    MySqlConnection con = new MySqlConnection(str);
```

```
    try
```

```
    {
```

```
        con.Open();
```

```
        MySqlDataAdapter da = new MySqlDataAdapter("select  
postid,title,content,author,datetimecreated,datecreated from posts where datecreated='" +  
newarcdate + "'", con);
```

```
        DataTable dt = new DataTable();
```

```
        da.Fill(dt);
```

```
        grdviewarchive.DataSource = dt;
```

```
        grdviewarchive.DataBind();
```

```

    }

    catch (Exception ex)
    {
        lblerror.Text = ex.ToString();
    }
    finally
    {
        con.Close();
    }
}
}

```

### **Search.html**

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" >
<head>
    <title>Site Search</title>
    <style type="text/css">

        a:hover{background:#a0522d}

    </style>
</head>
<body style="text-align: center">
<!-- SiteSearch Google -->
<form action="http://www.google.com/custom" target="google_window" name="googlesearch"
method="get">
    <div>
        <table style="width: 100%">
            <tr>

```



```

<td style="height: 96px; background-color: #a0522d; text-align: center">
    <em><a href="Default.aspx"><span style="font-size: 24pt; color: black; font-
family: Book Antiqua;
        text-align: center; text-decoration: none"><strong>JUIT Blogging Platform on
Asp.Net
        and C#</strong></span></a></em></td>
</tr>
<tr>
    <td style="width: 100px; height: 46px">
        <table style="width: 100%">
            <tr>
                <td style="width: 30%; height: 100%; text-align: center">
                    <a href="Default.aspx"><span style="border-right: #a0522d 1px solid;
border-top: #a0522d 1px solid;
                    border-left: #a0522d 1px solid; color: black; border-bottom: #a0522d 1px
solid;
                    font-family: Book Antiqua; text-decoration:
none">Home</span></a></td>
                <td style="width: 30%; height: 100%; text-align: center">
                    <a href="blogroll.aspx"><span style="border-right: #a0522d 1px solid;
border-top: #a0522d 1px solid;
                    border-left: #a0522d 1px solid; color: black; border-bottom: #a0522d 1px
solid;
                    font-family: Book Antiqua; text-decoration:
none">BlogRoll</span></a></td>
                <td style="width: 30%; height: 100%; text-align: center">
                    <a href="Contactus.aspx"><span style="border-right: #a0522d 1px solid;
border-top: #a0522d 1px solid;
                    border-left: #a0522d 1px solid; color: black; border-bottom: #a0522d 1px
solid;
                    font-family: Book Antiqua; text-decoration:
none">ContactUs</span></a></td>

```

```

        <td style="width: 30%; height: 100%; text-align: center">
            <a href="sitesearch.htm"><span style="border-right: #a0522d 1px solid;
border-top: #a0522d 1px solid;
            border-left: #a0522d 1px solid; color: black; border-bottom: #a0522d 1px
solid;
                font-family: Book Antiqua; text-decoration:
none">Search</span></a></td>
        <td style="width: 30%; height: 100%; text-align: center">
            <a href="AdminLogin.aspx"><span style="border-right: #a0522d 1px solid;
border-top: #a0522d 1px solid;
            border-left: #a0522d 1px solid; color: black; border-bottom: #a0522d 1px
solid;
                font-family: Book Antiqua; text-decoration:
none">SignIn</span></a></td>
    </tr>
</table>
</td>
</tr>
<tr>
    <td style="width: 100px; height: 91px">
        <?xml namespace="" prefix="asp" ?><asp:contentplaceholder
id="ContentPlaceHolder1" runat="server"><table bgcolor="#ffffff"
border="0" designer:dtid="3940649673949189"><TBODY
    __designer:dtid="3940649673949190"><TR
    __designer:dtid="3940649673949191"><TD vAlign=top noWrap align=left
height=32 __designer:dtid="3940649673949192"></TD><TD noWrap
    __designer:dtid="3940649673949193"><INPUT type=hidden
value="localhost/sample/default.aspx" name="domains"
    __designer:dtid="3940649673949194" /> <LABEL for="sbi"
    __designer:dtid="3940649673949195">Enter your search terms</LABEL>
<INPUT id="sbi" type=text maxLength=255 size=25 name="q"
    __designer:dtid="3940649673949196" /> <LABEL style="DISPLAY: none"

```

```

for="sbb" __designer:dtid="3940649673949197">Submit search
form</LABEL> <INPUT id="sbb" type=submit value="Search" name="sa"
__designer:dtid="3940649673949198" />
</TD></TR><TR __designer:dtid="3940649673949199"><TD
__designer:dtid="3940649673949200"></TD><TD noWrap
__designer:dtid="3940649673949201"><TABLE
__designer:dtid="3940649673949202"><TBODY
__designer:dtid="3940649673949203"><TR
__designer:dtid="3940649673949204"><TD
__designer:dtid="3940649673949205"><INPUT id="ss0" type=radio
CHECKED value="" name="sitesearch"
__designer:dtid="3940649673949206" /> <LABEL
title="Search the Web" for="ss0"
__designer:dtid="3940649673949207"><SPAN
style="COLOR: #5588aa"
__designer:dtid="3940649673949208">Web</SPAN></LABEL></TD><TD
__designer:dtid="3940649673949209"><INPUT id="ss1" type=radio
value="localhost/sample/default.aspx" name="sitesearch"
__designer:dtid="3940649673949210" /> <LABEL
title="Search sample" for="ss1"
__designer:dtid="3940649673949211"><SPAN
style="COLOR: #5588aa" __designer:dtid="3940649673949212">This

Site</SPAN></LABEL></TD></TR></TBODY></TABLE></TD></TR></TBODY></table>
&nbsp;</asp:contentplaceholder>

&nbsp;<br />
</td>
</tr>
</table>
</div>
<br />
</form>

```



```
<!-- SiteSearch Google -->
```

```
</body>
```

```
</html>
```

### **Adminlogin.aspx.cs**

```
using System;
```

```
using System.Data;
```

```
using System.Configuration;
```

```
using System.Collections;
```

```
using System.Web;
```

```
using System.Web.Security;
```

```
using System.Web.UI;
```

```
using System.Web.UI.WebControls;
```

```
using System.Web.UI.WebControls.WebParts;
```

```
using System.Web.UI.HtmlControls;
```

```
public partial class AdminLoginnew : System.Web.UI.Page
```

```
{
```

```
    protected void Page_Load(object sender, EventArgs e)
```

```
    {
```

```
        if (!IsPostBack)
```

```
        {
```

```
            if (Request.Cookies["RememberMe"] != null)
```

```
            {
```

```
                HttpCookie cookie = Request.Cookies.Get("RememberMe");
```

```
                string username = cookie.Values["User Name"].ToString();
```

```
                string password = cookie.Values["Password"].ToString();
```

```
                txtusername.Text = username;
```

```
                txtpwd.Attributes["value"] = password;
```

```
                txtpwd.Text = password;
```

```

    }
    //DateTime currtime = DateTime.Now.ToString("dd/MM/yyyy h:MM tt");
    //lblMessage.Text = currtime;
    //chkrememberme.Checked = true;
}
}
protected void btnsignin_Click(object sender, EventArgs e)
{
    if (Page.IsValid)
    {
        if (FormsAuthentication.Authenticate(txtusername.Text.Trim(), txtpwd.Text.Trim()))
        {
            if (chkrememberme.Checked)
            {
                HttpCookie remme = new HttpCookie("RememberMe");
                remme.Values.Add("User Name", txtusername.Text);
                remme.Values.Add("Password", txtpwd.Text);
                remme.Expires = DateTime.Now.AddDays(1);
                Response.Cookies.Add(remme);
            }
            else
            {
                //lblMessage.Text = "working...";
                if (Request.Cookies["RememberMe"] != null)
                {
                    Request.Cookies["RememberMe"].Values.Add("User Name", "");
                    // Values.Remove("User Name");
                    //delete.Values.Remove("Password");
                    //Response.Cookies.Remove(delete);
                }
            }
        }
    }
}

```

```

        //DateTime currtime = DateTime.Now;
        //lblMessage.Text = currtime.ToString();
        FormsAuthentication.RedirectFromLoginPage(txtusername.Text.Trim(), true);
    }
    else
    {

        lblMessage.Text = "Invalid Login, Please try Again !!!";
    }
}
}

```

```

protected void chkrememberme_CheckedChanged(object sender, EventArgs e)

```

```

{
    lblMessage.Text = "working...";
    if (Request.Cookies["RememberMe"] != null)
    {

        Request.Cookies["RememberMe"].Values.Add("User Name", "");
        // Values.Remove("User Name");
        //delete.Values.Remove("Password");
        //Response.Cookies.Remove(delete);
    }

}

```

```

protected void btnreset_Click(object sender, EventArgs e)

```

```

{
    txtusername.Text = "";
    txtpwd.Attributes["value"] = "";
    txtpwd.Text = "";
}

```



```
}  
}
```

### Admindefault.aspx.cs

```
using System;  
using System.Data;  
using System.Configuration;  
using System.Collections;  
using System.Web;  
using System.Web.Security;  
using System.Web.UI;  
using System.Web.UI.WebControls;  
using System.Web.UI.WebControls.WebParts;  
using System.Web.UI.HtmlControls;  
using MySql.Data.MySqlClient;  
  
public partial class Admin_Zone_admindefault : System.Web.UI.Page  
{  
    protected void Page_Load(object sender, EventArgs e)  
    {  
  
        string str =  
System.Configuration.ConfigurationManager.ConnectionStrings["abcd"].ToString();  
        MySqlConnection con = new MySqlConnection(str);  
        MySqlDataAdapter da = new MySqlDataAdapter("select  
postid,title,content,author,datecreated from posts order by datecreated desc", con);  
        DataTable dt = new DataTable();  
        try  
        {  
            con.Open();  
            da.Fill(dt);  
            grdviewpost.DataSource = dt;
```

```

        grdviewpost.DataBind();
    }
    catch (Exception ex)
    {
        lblerror.Text = ex.ToString();
    }
    finally
    {
        con.Close();
    }
}
}

```

### **Blogroll.aspx.cs**

```

using System;
using System.Data;
using System.Configuration;
using System.Collections;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;
using MySql.Data.MySqlClient;

public partial class Admin_Zone_blogroll : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        string str =
        System.Configuration.ConfigurationManager.ConnectionStrings["abcd"].ToString();
    }
}

```

```

MySQLConnection con = new MySqlConnection(str);
MySQLDataAdapter da = new MySQLDataAdapter("select name,url from blogroll", con);
DataTable dt = new DataTable();
try
{
    con.Open();
    da.Fill(dt);
    grdviewblogroll.DataSource = dt;
    grdviewblogroll.DataBind();
}
catch (Exception ex)
{
    lblerror.Text = ex.ToString();
}
finally
{
    con.Close();
}
}

```

```

protected void btnaddblog_Click(object sender, EventArgs e)
{
    string str =
System.Configuration.ConfigurationManager.ConnectionStrings["abcd"].ToString();
    MySqlConnection con = new MySqlConnection(str);
    try
    {
        con.Open();
        MySqlCommand cmdInsert = new MySqlCommand("Insert into blogroll(name,url)
values('"+ txtname.Text + "','"+ txturl.Text + "')", con);
        int i = cmdInsert.ExecuteNonQuery();
    }
}

```



```

        if (i == 1)
        {
            txtname.Text = "";
            txturl.Text = "";

        }
    }
    catch (Exception ex)
    {
        lblerror.Text = ex.ToString();
    }
    finally
    {
        con.Close();
    }
}
}

```

### **Settings.aspx.cs**

```

using System;
using System.Data;
using System.Configuration;
using System.Collections;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;
using MySql.Data.MySqlClient;

public partial class Admin_Zone_settings : System.Web.UI.Page

```

```

{
protected void Page_Load(object sender, EventArgs e)
{

}

protected void btnupdate_Click(object sender, EventArgs e)
{
//String encodeemail = EncodeEmailAddress(txtemail.Text);
string str =
System.Configuration.ConfigurationManager.ConnectionStrings["abcd"].ToString();
MySQLConnection con = new MySqlConnection(str);
try
{
con.Open();
MySQLCommand cmdInsert = new MySQLCommand("update blogdetail set name=" +
txtname.Text + ", description=" + txtdescription.Text + ", owner=" + txtowner.Text +
",ownerprofile=" + txtprofile.Text + ", email=" + txtemail.Text + "", con);
int i = cmdInsert.ExecuteNonQuery();
if (i == 1)
{
txtname.Text = "";
txtdescription.Text = "";
txtowner.Text = "";
txtprofile.Text = "";
txtemail.Text = "";
}
}

catch (Exception ex)
{
lblerror.Text = ex.ToString();
}
}

```

finally

```
{  
    con.Close();  
}  
}
```

protected void btnupload\_Click(object sender, EventArgs e)

```
{  
    try  
    {  
        String strfilename="adminpic.jpg";  
        FileUpload1.SaveAs(MapPath("uploaded files") + "\\" + strfilename);  
        string strfullimage = (MapPath("uploaded files") + "\\" + strfilename);  
        //lblmessage.Text = strfullimage;  
        System.Drawing.Image imgfull = System.Drawing.Image.FromFile(strfullimage);  
        System.Drawing.Image imgthumb = imgfull.GetThumbnailImage(100, 100, null,  
IntPtr.Zero);  
        imgthumb.Save(MapPath("uploaded files") + "\\" + "adminpicthumb.jpg");  
        lblmessage.Text = "Picture Uploading Successful!!!";  
    }  
    catch (Exception ex)  
    {  
        lblerror.Text = ex.ToString();  
    }  
}
```

Addpost.aspx.cs



```

using System;
using System.Data;
using System.Configuration;
using System.Collections;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;
using MySql.Data.MySqlClient;

public partial class Admin_Zone_addpostnew : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {

        if (!this.IsPostBack)
        {
            if (User.Identity.IsAuthenticated)
            {
                string str =
System.Configuration.ConfigurationManager.ConnectionStrings["abcd"].ToString();
                MySqlConnection con = new MySqlConnection(str);
                MySqlDataAdapter da = new MySqlDataAdapter("select lastlogininfo from logininfo",
con);
                DataTable dt = new DataTable();
                try
                {
                    con.Open();
                    da.Fill(dt);

```

```
grdviewlogin.DataSource = dt;  
grdviewlogin.DataBind();
```

```
}
```

```
catch (Exception ex)
```

```
{
```

```
    lblerror.Text = ex.ToString();
```

```
}
```

```
finally
```

```
{
```

```
    con.Close();
```

```
}
```

```
try
```

```
{
```

```
    Session["filename"] = "null";
```

```
    fillDlistCategories();
```

```
}
```

```
catch (Exception ex)
```

```
{
```

```
    lblerror.Text = ex.ToString();
```

```
}
```

```
}
```

```
}
```

```
}
```

```
private void fillDlistCategories()
```

```
{
```

```

string str =
System.Configuration.ConfigurationManager.ConnectionStrings["abcd"].ToString();
    MySqlConnection con = new MySqlConnection(str);
    MySqlDataAdapter da = new MySqlDataAdapter("select categoryid,name from categories",
con);
    DateTime curdate = DateTime.Now;
    String curdate1 = curdate.ToString("dd/MM/yyyy hh:mm:ss");
    DataTable dt = new DataTable();
    try
    {
        con.Open();
        MySqlCommand cmd = new MySqlCommand("update logininfo set lastlogininfo=" +
curdate1 + "'", con);
        int i = cmd.ExecuteNonQuery();
        da.Fill(dt);
        chkboxlstcat.DataSource = dt;
        chkboxlstcat.DataBind();
    }
    catch (Exception ex)
    {
        lblerror.Text = ex.ToString();
    }
    finally
    {
        con.Close();
    }
}

```

```

protected void btnsubmit_Click(object sender, EventArgs e)

```

```

{
    lblMessage.Text = "";

```



```

/*lblerror.Text = "";
string strfilename = FileUpload1.FileName;
int i = strfilename.LastIndexOf('.');
string strext = strfilename.Substring(i + 1);
if (strfilename != "")
{
    if (strext == "jpg")
    {
        FileUpload1.SaveAs(MapPath("uploaded pic") + "\\" + strfilename);
    }

    else
    {
        lblerror.Text = "Only jpg images are allowed";
    }
}*/

string str =
System.Configuration.ConfigurationManager.ConnectionStrings["abcd"].ToString();
MySQLConnection con = new MySqlConnection(str);
//MySQLConnection con1 = new MySqlConnection(str);
try
{
    con.Open();
    String filename = Session["filename"].ToString();
    DateTime currtimel = DateTime.Now;
    String currtime = currtimel.ToString("yyyy-MM-dd hh:mm:ss");
    String currrdate = currtimel.ToString("yyyy-MM-dd");

    MySqlCommand cmdInsert = new MySqlCommand("Insert into
posts(title,description,author,content,datetimecreated,datecreated,uploadedfilename) values('" +
txttitle.Text + "','" + txtdescription.Text + "','" + txtauthor.Text + "','" + txtcontent.Text + "','" +
currtime + "','" + currrdate + "','" + filename + "')", con);

```

```

int i = cmdInsert.ExecuteNonQuery();
if (i == 1)
{
    txttitle.Text = "";
    txtdescription.Text = "";
    txtcontent.Text = "";
    txtauthor.Text = "";
}
}

catch (Exception ex)
{
    lblerror.Text = ex.ToString();
}

finally
{
    con.Close();
}

}

protected void btnreset_Click(object sender, EventArgs e)
{
    txttitle.Text = "";
    txtdescription.Text = "";
    txtcontent.Text = "";
    txtcategories.Text = "";
}

protected void btnadd_Click(object sender, EventArgs e)
{

```

```

string str =
System.Configuration.ConfigurationManager.ConnectionStrings["abcd"].ToString();
    MySqlConnection con1 = new MySqlConnection(str);
    // MySqlDataAdapter da = new MySqlDataAdapter("select categoryid,name from
categories", con1);
    //DataTable dt = new DataTable();

    try
    {
        con1.Open();
        MySqlCommand cmdInsert1 = new MySqlCommand("Insert into categories(name)
values('" + txtcategories.Text + "')", con1);
        int i1 = cmdInsert1.ExecuteNonQuery();
        if (i1 == 1)
        {
            txtcategories.Text = "";
        }
        //da.Fill(dt);
        //checkbox1stcat.DataSource = dt;
        //checkbox1stcat.DataBind();
    }
    catch (Exception ex)
    {
        lblerror.Text = ex.ToString();
    }
    finally
    {
        con1.Close();
    }

```



```
}
```

```
protected void btnupload_Click(object sender, EventArgs e)
```

```
{
```

```
    try
```

```
    {
```

```
        string strfilename = FileUpload1.FileName;
```

```
        //int i = strfilename.LastIndexOf('.');
```

```
        //string strext = strfilename.Substring(i + 1);
```

```
        if (strfilename != "")
```

```
        {
```

```
            FileUpload1.SaveAs(MapPath("uploaded files") + "\\ " + strfilename);
```

```
            Session["filename"] = strfilename;
```

```
            lblMessage.Text = "File Uploading Successful!!!";
```

```
        }
```

```
    else
```

```
    {
```

```
        lblMessage.Text = "Please select a file first!";
```

```
    }
```

```
}
```

```
catch (Exception ex)
```

```
{
```

```
    lblerror.Text = ex.ToString();
```

```
}
```

```
}
```

```
protected void grdviewlogin_SelectedIndexChanged(object sender, EventArgs e)
```

```
{
```

```
}
```

```
}
```

### **Moderate.aspx.cs**

```
using System;
using System.Data;
using System.Configuration;
using System.Collections;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;
using MySql.Data.MySqlClient;

public partial class Admin_Zone_moderate : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!this.IsPostBack)
        {
            fillPostsDataList();
        }
    }

    private void fillPostsDataList()
    {
        string str =
System.Configuration.ConfigurationManager.ConnectionStrings["abcd"].ToString();
        MySqlConnection con = new MySqlConnection(str);
```

```

        MySqlDataAdapter da = new MySqlDataAdapter("select postid,author, email, website,
commentdate, commentdetail, status from comments where status = 0", con);

        //MySqlDataAdapter da1 = new MySqlDataAdapter("select author, website,commentdetail
from comments where postid=" + pid.ToString()+ "&& status=1", con);

        // MySqlDataAdapter da1 = new MySqlDataAdapter("select owner,ownerprofile,email
from blogdetail", con);

        DataTable dt = new DataTable();
        //DataTable dt1 = new DataTable();

        try
        {
            con.Open();
            da.Fill(dt);
            //da1.Fill(dt1);

            grdviewmoderate.DataSource = dt;
            // grdviewcomments.DataSource = dt1;

            grdviewmoderate.DataBind();
            // grdviewcomments.DataBind();
        }
        catch (Exception ex)
        {
            lblerror.Text = ex.ToString();
        }
        finally
        {
            con.Close();
        }
    }

    protected void rdbtnapprove_CheckedChanged(object sender, EventArgs e)
    {

    }

    protected void btnapprove_Click(object sender, EventArgs e)

```



```

{
    string str =
System.Configuration.ConfigurationManager.ConnectionStrings["abcd"].ToString();
    MySqlConnection con = new MySqlConnection(str);
    MySqlDataAdapter da = new MySqlDataAdapter("select postid,author, email, website,
commentdate, commentdetail, status from comments where status = 0", con);

    DataTable dt = new DataTable();
    try
    {
        con.Open();
        da.Fill(dt);
    }
    catch (Exception ex)
    {

    }

    foreach( GridViewRow gvrow in grdviewmoderate.Rows)
    {
        RadioButton rdb = (RadioButton)gvrow.FindControl("rdbapprove");
        if(rdb.Checked==true)
        {
            int x = gvrow.RowIndex;
            String str1 = dt.Rows[x]["postid"].ToString();
            String query = "Update comments set status = 1 where postid = " + str1;
            MySqlCommand cmd = new MySqlCommand(query, con);
            cmd.ExecuteNonQuery();
            Response.Redirect("moderate.aspx");
        }
    }
    //lblerror.Text = count.ToString();
}

```