

BLOGGING MANAGEMENT SYSTEM

A Project Report

Submitted to the Department of MCA

In partial fulfillment of the Project of the degree of



Submitted By:

Suman Kumar 2001330140057

Md. Gulsher Alam 2001330140030

Singh Yogesh Kumar Vijay Kumar 2001330140055

Submitted To:

Under Supervision of

Hirdesh Sharma

(Assistant professor)

DEPARTEMENT OF MASTER OF COMPUTER APPLICATION

**NOIDA INSTITUTE OF ENGINEERING & TECHNOLOGY
AN AUTONOMOUS INSTITUTE
Greater Noida, Uttar Pradesh.
2020-2022.**

ACKNOWLEDGEMENT

I would like to take this opportunity to express my gratitude towards all the people who have in various ways, helped in the successful completion of project. I wish to express my heartfelt gratitude to the following individuals who have played a crucial role in the research for this project. Without their active co-operation the preparation of this project could not have been completed within the specified time limit. The first person I would like to acknowledge is my guide PROF. HIRDESH SHARMA SIR who supported us throughout this project with the most possible cooperation and patience. I am very much thankful to them for sparing their precious time for me and for helping me in doing this project.

Place- Greater Noida

Date-

Suman Kumar (2001330140057)

Md. Gulsher Alam(2001330140030)

Singh Yogesh Kr. Vijay Kr. (2001330140055)

BONAFIDE CERTIFICATE

This is certified that this project report “**Online Blogging System**” is the bonafide work of “**Suman Kumar(2001330140057), Md. Gulsher Alam(2001330140030) and Singh Yogesh Kumar Vijay Kumar (2001330140055)**” who carried out the project under my supervision.

(Signature of Guide)
Mr. Hirdesh Sharma
Assistant Professor

Date:

Signature of HOD

Date:

Signature of External Examiner

Date:

NOIDA INSTITUTE OF ENGINEERING & TECHNOLOGY
19, Knowledge Park-II, Institutional Area, Phase-II,
Greater Noida, Uttar Pradesh

TABLE OF CONTENTS

Chapter 1: Introduction		
I.I	Introduction	1
I.II	Background of Project	2
I.III	Aim and Purpose	5
I.IV	Product Goals and Objective	6
I.V	Purpose, Scope	7
Chapter 2: System Analysis		
II.I	Software Requirement Specification	9
II.II	Survey Of Technology	10
Chapter 3: System Design		
III.I	Water Fall Model	17
III.II	ER Diagram	21
III.III	Data Flow Diagram	23
III.IV	UML Diagram	24
III.V	Activity Diagram	25
Chapter 4: System Testing		
IV.I	Unit Testing	28
IV.II	Integration Testing	30
Chapter V: Screenshot		31
Chapter VI: Maintenance		36
Chapter VII: Recommendations for further work		37
Chapter VIII: Conclusion & Future Scope		38
Chapter IX: References		39

ABSTRACT

The main purpose of a Content Management System is to make it easy for even a novice computer user to maintain and manage a site. There are many CMSs available in the market but are not user-friendly. Both Drupal and Joomla have a very bloated and confusing administration. This is because both these packages have many advanced features in addition to the normal basic content management features. For example, polls providing the capabilities to capture votes on different topics in the form of multiple-choice questions, or news feeds, which provide syndicated content (RSS, RDF, and Atom feeds). This makes the CMS complex and difficult to use. Inevitably though, that CMS is going to be a little more complex to maintain. If the website requires complex features, the best thing to do is to hide these complex features. A good user interface should make most common tasks the most prominent and hide rare tasks so that they do not get in the way. Drupal's administration interface is confusing and not user friendly. Joomla's administration usability and learning curve is better than Drupal's, but not enough to provide a noticeable advantage to the end-user over Drupal. WordPress has a much better and very intuitive administration design, which makes it easier to learn. It includes features such as drag and drop, resulting in the generation of code without technical intervention. It would be more correct to describe such products as 'website builders' than Web Content Management Systems. The main feature that is not seen in most of today's complex CMSs is intuitive and user-friendly website administration. Hence, we've built this simple CMS which has a very easy-to-use UI for an end-user and has collaborative multiple user support, powerful user feedback section and active reader engagement in the form of likes, comments, social media shares etc.

INTRODUCTION

This project is designed in such a way that it allows extensive reader engagement in the form of blog comments and social media shares, likes, and follows and latest technological content posts. This system also supports content creation and manipulation by multiple users such that each user is authorized to access specific sections as per their user role and contribution towards this system.

A blog (a [blend](#) of the term web log) is a type of [website](#) or part of a website, which is the publication of regular articles over time in some area of personal or professional interest. Blogs are usually maintained by an individual with regular entries of commentary, descriptions of events, or other material such as graphics or video. Blog can also be used as a verb, meaning to maintain or add content to a blog.

There are many different types of blogs, differing not only in the type of content, but also in the way that content is delivered or written.

The software for writing blogs can be divided into **user hosted** (software application installed by web log authors to run on their own system) and **developer hosted** (software services operated by the developer, requiring no software installation for the blog author.)

BACKGROUND

Content Management System (CMS):-

A content management system (CMS) manages the creation and modification of digital content. It typically supports multiple users in a collaborative environment. CMS features vary widely. Most CMSs include Web-based publishing, format management, history editing and version control, indexing, search, and retrieval. By their nature, content management systems support the separation of content and preparation. A web content management system (WCM or WCMS) is a CMS designed to support the management of the content of Web pages. Most popular CMSs are also WCMSs. Web content includes text and embedded graphics, photos, video, audio, maps, and program code (e.g., for applications) that displays content or interacts with the user.

Such a content management system (CMS) typically has two major components:

- A content management application (CMA) is the front-end user interface that allows a user, even with limited expertise, to add, modify, and remove content from a website without the intervention of a webmaster.
- A content delivery application (CDA) compiles that information and updates the website.

CMS come in all shapes and sizes and can manage anything a team of individuals is working on. From managing simple static website content, to allowing collaborative documentation across the Internet (wiki), CMSs perform many functions.

CMS packages can generally be classified into four categories:

- Enterprise CMS
- Web CMS
- Open Source CMS
- Commercial CMS

Open source content management systems are free in many ways. A user can do what he/she wishes with the product and the code behind it, extending and integrating it as they see fit. There's no license cost for the software, and anyone can download and install them on a web server without cost, though it is likely they will have to pay for the server or pay someone to install the system.

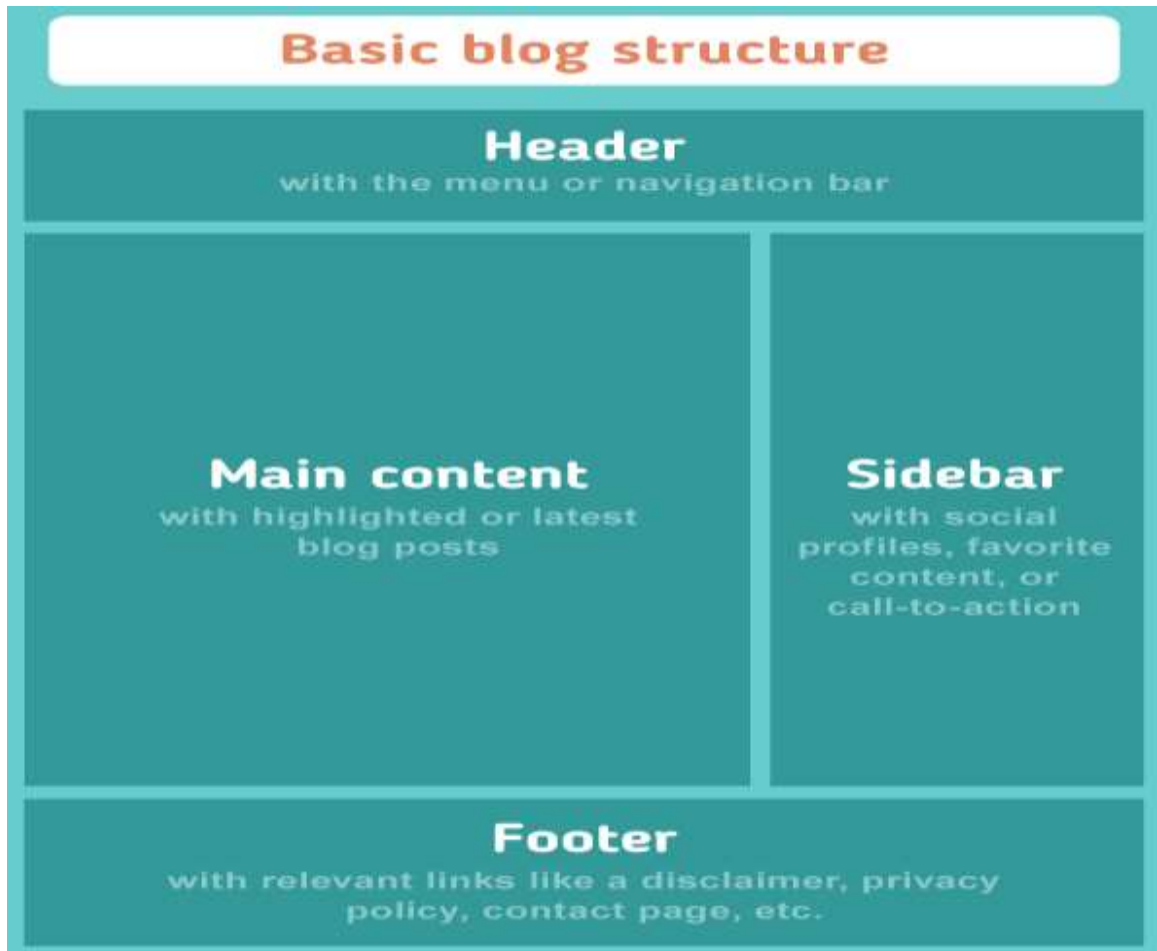
These are top 12 open source content management software systems :-

- ❖ DRUPAL
- ❖ ACQUIA
- ❖ SQUARESPACE
- ❖ HUBSPOT
- ❖ JOOMLA
- ❖ SITECORE
- ❖ BOX
- ❖ BRIGHTCOVE
- ❖ PANTHEON.IO
- ❖ KENTICO
- ❖ BLUE PEN ARTICLES
- ❖ PYRO CMS

Joomla, Drupal and WordPress are the most popular Open Source CMS being used today. Drupal has a steep learning curve and so it's difficult for a first-time developer to understand the modules. It requires more advanced coding knowledge when it comes to customizing. Joomla has a better user interface than Drupal for newcomers. It is quite simple to add and edit pages. The design layer is almost the same as Drupal as they both do not have a lot of design templates. So, the issue is that most of the Joomla sites will look the same. WordPress is the perfect solution for small blogging websites as it has simple and user-friendly interface. There are a lot of great design templates available for the websites.

2.2 Blog:-

A blog is a discussion or informational website published on the World Wide Web consisting of discrete, often informal diary-style text entries. Posts are typically displayed in reverse chronological order, so that the most recent post appears first, at the top of the web page. The appearance of blogs changed over time, and nowadays blogs include different items. But, most blogs include some standard features and structure.



AIM AND PURPOSE

The main purpose of this project is to have a user-friendly content administration interface that includes most common CMS functions appropriate for small and simple websites, so that a novice user can manage the website content. A user having less coding knowledge can easily add, edit and format the website's content using the rich text editor integrated in the Simple CMS engine without having to deal with the HTML and CSS code. Our CMS provides an easy and efficient way of engaging end-user for actively participating and contributing to this educational blogging system, where they can stay updated with the latest technological trends and at the same time provide valuable feedback to help develop the system better, bug-fixes (if any), innovative idea spreading, provide better service and relevant content with time. This system also supports content creation and manipulation by multiple users such that each user is authorized to access specific sections as per their user role and contribution towards this system. So, that each reader would get equal opportunity to grow, innovate, contribute to our blogging system.

PRODUCT GOALS AND OBJECTIVES

By my analysis on website, I was expected that for common people to develop website, they have to face many problems and also they want to share their knowledge, technology, articles, etc.

The objective of our Project is to provide all its user with easy access to relevant, and by using provide them Choosing facility to component.

People will make their blog using choose their style, template, etc.

PURPOSE, EXISTING SYSTEM

PURPOSE

Our simple CMS has a user-friendly content administration interface that includes most common CMS functions appropriate for small and simple websites, so that a novice user can manage the website content. A user having less coding knowledge can easily add, edit and format the website's content using the rich text editor integrated in the Simple CMS engine without having to deal with the HTML and CSS code. Our CMS provides an easy and efficient way of engaging end-user for actively participating and contributing to this educational blogging system, where they can stay updated with the latest technological trends and at the same time provide valuable feedback to help develop the system better, bug-fixes (if any), innovative idea spreading, provide better service and relevant content with time. This system also supports content creation and manipulation by multiple users such that each user is authorized to access specific sections as per their user role and contribution towards this system. So, that each reader would get equal opportunity to grow, innovate, contribute to our blogging system.

Existing System:

The main purpose of a Content Management System is to make it easy for even a novice computer user to maintain and manage a site. There are a number of custom CMS-frameworks available out there like WordPress, Joomla, Drupal, Plone, Open CMS, TYPO3 etc. using which a complete CMS can be developed but all of these frameworks have a steep learning curve and so it's difficult for an end-user to understand the modules. It requires more advanced coding knowledge when it comes to customizing. Both Drupal and Joomla have a very bloated and confusing administration. This is because both these packages have many advanced features in addition to the normal basic content management features. So, there is an utmost need to build a simple CMS that should be user-friendly.

Link to GitHub Repository: <https://github.com/yogesh2104/miniProject>

SYSTEM ANALYSIS

HARDWARE AND SOFTWARE REQUIREMENTS

Hardware:

- Pentium-pro processor or later.
- RAM 512MB or more.

Software:

- Windows 7(32-bit) or above.
- Python 2.7 or later.
- Chrome Browser.
- Any Text Editor.

SURVEY OF TECHNOLOGY

Python

Python is OOPs (Object Oriented Programming) based, high level, interpreted programming language. It is a robust, highly useful language focused on rapid application development (RAD). Python helps in easy writing and execution of codes. Python can implement the same logic with as much as 1/5th code as compared to other OOPs languages.

Python provides a huge list of benefits to all. The usage of Python is such that it cannot be limited to only one activity. Its growing popularity has allowed it to enter into some of the most popular and complex processes like Artificial Intelligence (AI), Machine Learning (ML), natural language processing, data science etc. Python has a lot of libraries for every need of this project. For Sophia, libraries used are speech recognition to recognize voice for text to speech, selenium for web automation etc.

Python is reasonably efficient. Efficiency is usually not a problem for small examples. If your Python code is not efficient enough, a general procedure to improve it is to find out what is taking most the time, and implement just that part more efficiently in some lower-level language. This will result in much less programming and more efficient code (because you will have more time to optimize) than writing everything in a low-level language.

HTML:-

HTML stands for Hyper Text Markup Language. A markup language is a language that annotates text in a way that is syntactically distinguishable so that the computer can manipulate it. It is a set of markup tags used to describe web pages.

The tags are what separate normal text from HTML code. They are the words

between the <angle-brackets>. Different tags will perform different functions, like rendering images or tables. It is a combination of words and symbols which give instructions on how the document will be presented. The tags themselves don't appear when you view your page through a browser, but their effects do. Markup is what HTML tags do to the text inside them. They mark it as a certain type of text (italicized text, for example). HTML documents contain HTML tags and plain text. The content on a HTML page will be static. In order to change the content, the editor needs to have some knowledge about HTML and change the content accordingly.

CSS:-

CSS stands for Cascading Style Sheets. It is used to control the style and layout of multiple web pages all at once. Styles define how to display HTML elements. CSS overrides the browser's default settings for interpreting how tags should be displayed, letting you use any HTML element indicated by an opening and closing tag to apply style attributes defined either locally or in a stylesheet. External Style Sheets can save a lot of work. They are stored in CSS files. Stylesheets contain rules, composed of selectors and declarations that define how styles will be applied. The selector (a redefined HTML element, class name, or ID name) is the link between the HTML document and the style. There are two different kinds of selectors: types (HTML element tags) and attributes (such as class and ID names).

Django:-

Django is a high-level Python web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the

hassle of web development, so you can focus on writing your app without needing to reinvent the wheel. It's free and open source.

Ridiculously fast.

Django was designed to help developers take applications from concept to completion as quickly as possible.

Reassuringly secure.

Django takes security seriously and helps developers avoid many common security mistakes.

Exceedingly scalable.

Some of the busiest sites on the web leverage Django's ability to quickly and flexibly scale.

Also included in the core framework are:

- a lightweight and standalone [web server](#) for development and testing.
- a form serialization and validation system that can translate between [HTML](#) forms and values suitable for storage in the database.
- a template system that utilizes the concept of [inheritance](#) borrowed from object-oriented programming.
- a [caching](#) framework that can use any of several cache methods.
- support for [middleware](#) classes that can intervene at various stages of request processing and carry out custom functions.
- an internal dispatcher system that allows components of an application to communicate events to each other via pre-defined signals.
- an [internationalization](#) system, including translations of Django's own components into a variety of languages.
- a [serialization](#) system that can produce and read [XML](#) and/or [JSON](#) representations of Django model instances.

- A system for extending the capabilities of the template engine.
- An interface to Python's built-in [unit test](#) framework.

SQLite

SQLite is a C library that provides a lightweight disk-based database that doesn't require a separate server process and allows accessing the database using a nonstandard variant of the SQL query language. Some applications can use SQLite for internal data storage. It's also possible to prototype an application using SQLite and then port the code to a larger database such as PostgreSQL or Oracle.

The `sqlite3` module was written by Gerhard Häring. It provides a SQL interface compliant with the DB-API 2.0 specification described by [PEP 249](#), and requires SQLite 3.7.15 or newer.

To use the module, start by creating a [Connection](#) object that represents the database.

Instead, use the DB-API's parameter substitution. To insert a variable into a query string, use a placeholder in the string, and substitute the actual values into the query by providing them as a [tuple](#) of values to the second argument of the cursor's [execute\(\)](#) method. An SQL statement may use one of two kinds of placeholders: question marks (qmark style) or named placeholders (named style). For the qmark style, `parameters` must be a [sequence](#). For the named style, it can be either a [sequence](#) or [dict](#) instance. The length of the [sequence](#) must match the number of placeholders, or a [ProgrammingError](#) is raised. If a [dict](#) is given, it must contain keys for all named parameters.

Heroku Deploy:

Deploy from Git, your CI system, or deploy on [every push to a branch in GitHub](#). Heroku runs your app in a dyno — a smart, secure container with your choice of Python version. [Dynos](#) come in different types, ranging from free dynos for getting started, to dynos at \$7 per month for hobby projects, all the way to dedicated types for your highest-traffic apps.

SYSTEM DESIGN

Module Division:

1. User Interaction:

In the user interaction user can read any blog and also can leave comment. Then admin can reply them.

2. Admin Interaction:

In the ADMIN can add number of blog on any topic and can read user comment on particular blog.

Feasibility Study:

The feasibility study is an evaluation and analysis of the potential of a proposed project which is based on extensive investigation and research to.

Technical Feasibility :

It specifies whether the proposed solution to the project is possible to be implemented using available hardware and software. To build this software we stretched our coding efforts to the Php and gained knowledge of MySQL. And our team is competent in that.

Social Feasibility:

All end-user can use and can contribute to this blogging system irrespective of any educational background. Technical knowledge is not mandatory.

Market Research:

Market research says that this system would be useful for the almost every user as it could seamlessly help them for their modern and busy lifestyle, keeping them updated with latest educational and technological trends and also help them to participate actively in the form of comments, likes, social media shares, etc.

Economic Feasibility:

This defines the cost effectiveness of the project. This project being a part of academic project, we've tried to minimize the overall cost. We've been successful in doing so.

Alternate Solution:

Could be a mobile app but that would not be as efficient as of this system

Waterfall Model – Design

Waterfall approach was first SDLC Model to be used widely in Software Engineering to ensure success of the project. In "The Waterfall" approach, the whole process of software development is divided into separate phases. In this Waterfall model, typically, the outcome of one phase acts as the input for the next phase sequentially.

When to use SDLC Waterfall Model?

Waterfall Methodology can be used when:

- Requirements are not changing frequently
- Application is not complicated and big
- Project is short
- Requirement is clear
- Environment is stable
- Technology and tools used are not dynamic and is stable
- Resources are available and trained

The sequential phases in Waterfall model are –

- **Requirement Gathering and analysis** – All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification document.
- **System Design** – The requirement specifications from first phase are studied in this phase and the system design is prepared. This system design helps in specifying hardware and system requirements and helps in defining the overall system architecture.
- **Implementation** – With inputs from the system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality, which is referred to as Unit Testing.
- **Integration and Testing** – All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.
- **Deployment of system** – Once the functional and non-functional testing is done; the product is deployed in the customer environment or released into the market.
- **Maintenance** – There are some issues which come up in the client environment. To fix those issues, patches are released. Also to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.

All these phases are cascaded to each other in which progress is seen as flowing steadily downwards (like a waterfall) through the phases. The next phase is started only after the defined set of goals are achieved for previous phase and it is signed off, so the name "Waterfall Model". In this model, phases do not overlap.

Waterfall Model – Application

Every software developed is different and requires a suitable SDLC approach to be followed based on the internal and external factors. Some situations where the use of Waterfall model is most appropriate are –

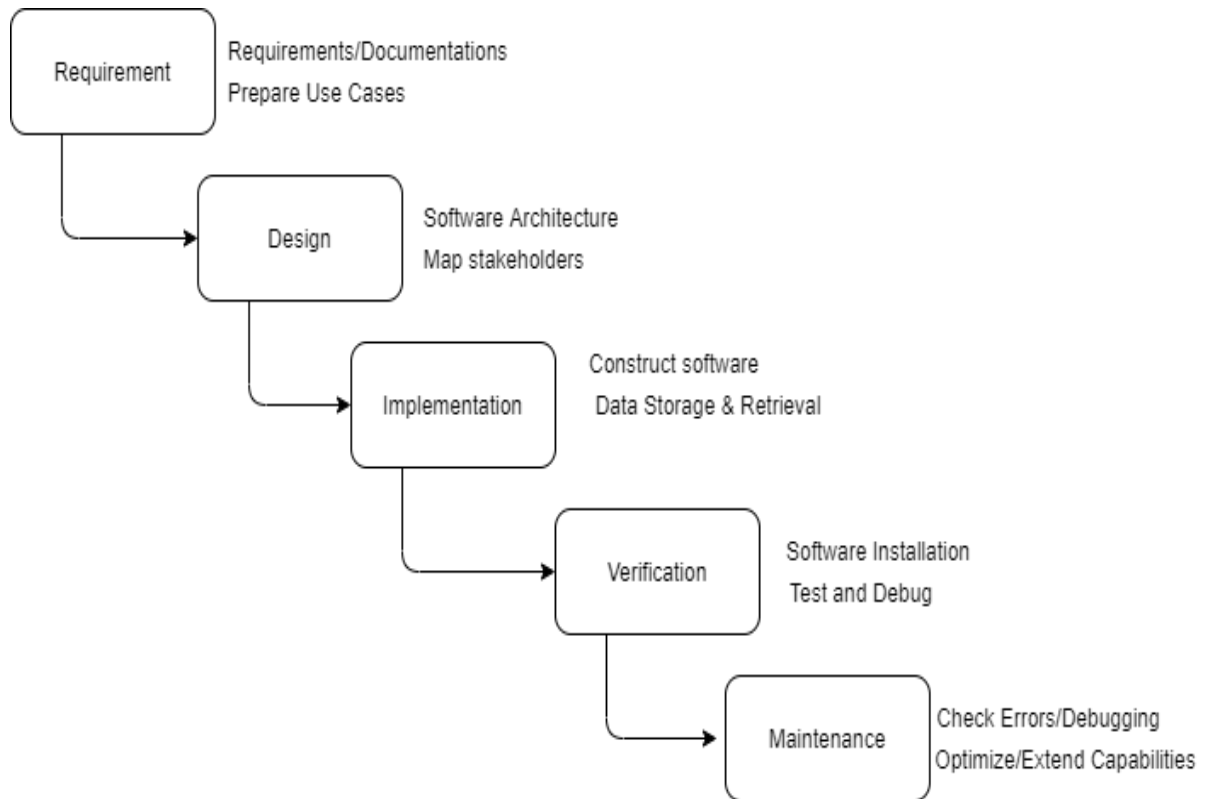
- Requirements are very well documented, clear and fixed.
- Product definition is stable.
- Technology is understood and is not dynamic.
- There are no ambiguous requirements.
- Ample resources with required expertise are available to support the product.
- The project is short.

Waterfall Model – Advantages

The advantages of waterfall development are that it allows for departmentalization and control. A schedule can be set with deadlines for each stage of development and a product can proceed through the development process model phases one by one.

Some of the major advantages of the Waterfall Model are as follows –

- Simple and easy to understand and use
- Easy to manage due to the rigidity of the model. Each phase has specific deliverables and a review process.
- Phases are processed and completed one at a time.
- Works well for smaller projects where requirements are very well understood.
- Clearly defined stages.
- Well understood milestones.
- Easy to arrange tasks.
- Process and results are well documented.



Waterfall Model - Disadvantages

The disadvantage of waterfall development is that it does not allow much reflection or revision. Once an application is in the testing stage, it is very difficult to go back and change something that was not well-documented or thought upon in the concept stage.


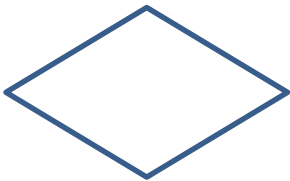
The major disadvantages of the Waterfall Model are as follows –


- No working software is produced until late during the life cycle.
- High amounts of risk and uncertainty.
- Not a good model for complex and object-oriented projects.
- Poor model for long and ongoing projects.

E-R Diagram

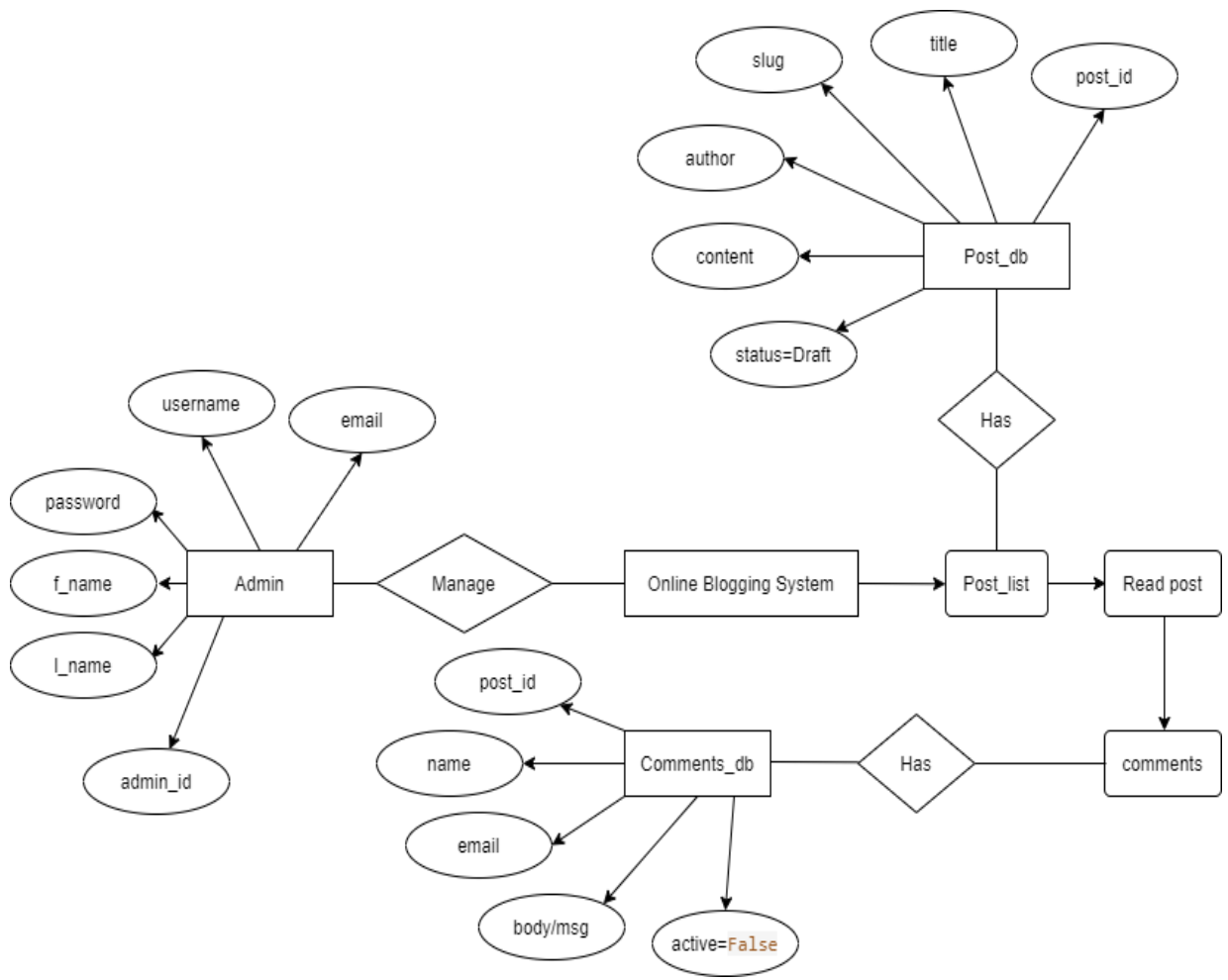
An entity-relationship (ER) diagram is a specialized graphic that illustrates the ER diagrams often use symbols to represent three different types of information. Boxes are commonly used to represent entities. Diamonds are normally used to represent relationships and ovals are used to represent attributes .The utility of the ER model is:

- It maps well to the relational model. The constructs used in the ER model can easily be transformed into relational tables.
- It is simple and easy to understand with a minimum of training. Therefore, the model can be used by the database designer to communicate the design to the end user. In addition, the model can be used as a design plan by the database developer to implement a data model in a specific database management software.

Name	Notation	Description
Entity		Entity is represented by a box within the ERD. Entities are abstract concepts, each representing one or more instances of the concept in question. Entities are equivalent to database tables in a relational database, with each row of the table representing an instance of that entity
Relationship		Relationships are represented by Diamonds. A relationship is a named collection or association between entities or used to relate to two or more entities with some common attributes or meaningful interaction between the

		objects.
Attributes		Attributes are represented by Oval. An attribute is a single data item related to a database object. The database schema associates one or more attributes with each database entity

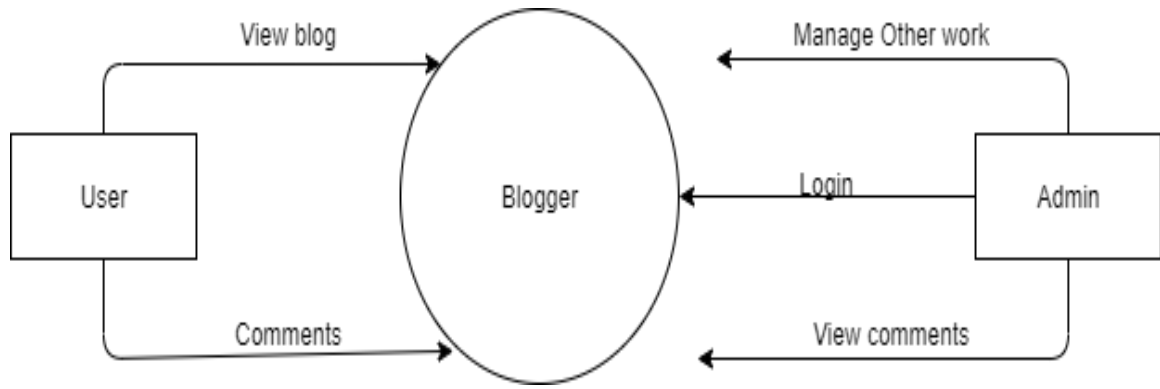
ER-DIAGRAM



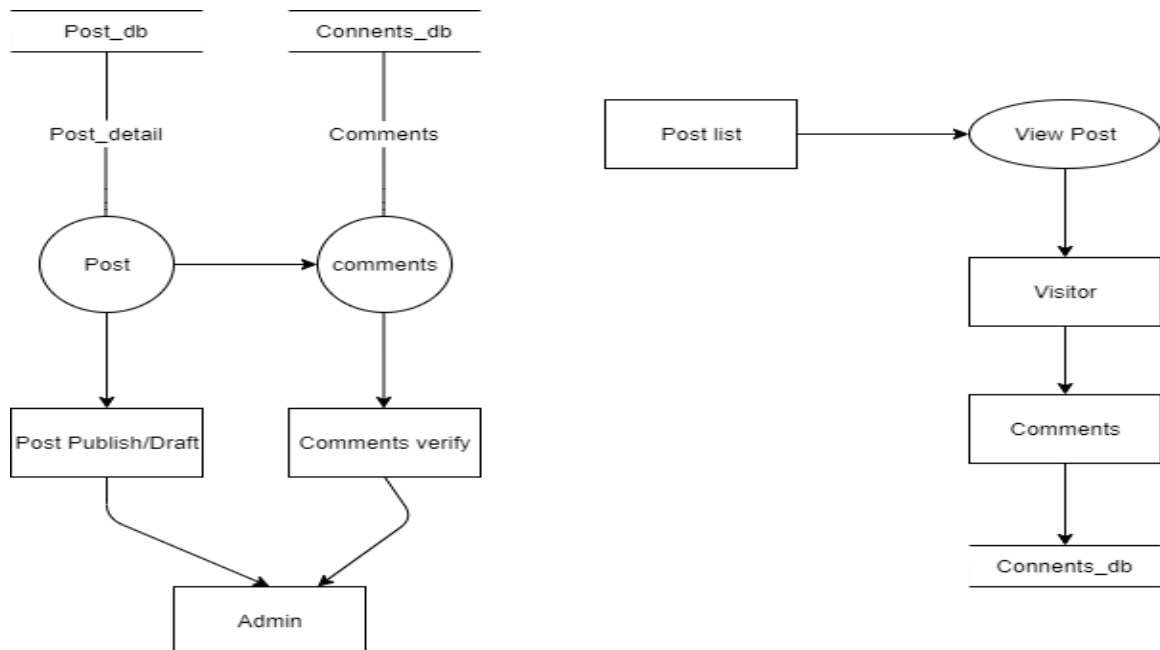
Data Flow Diagram/UML

The DFD also provides information about the outputs and inputs of each entity and the process itself. A data-flow diagram has no control flow, there are no decision rules and no loops. Specific operations based on the data can be represented by a Flowchart.

DFD 0 level:

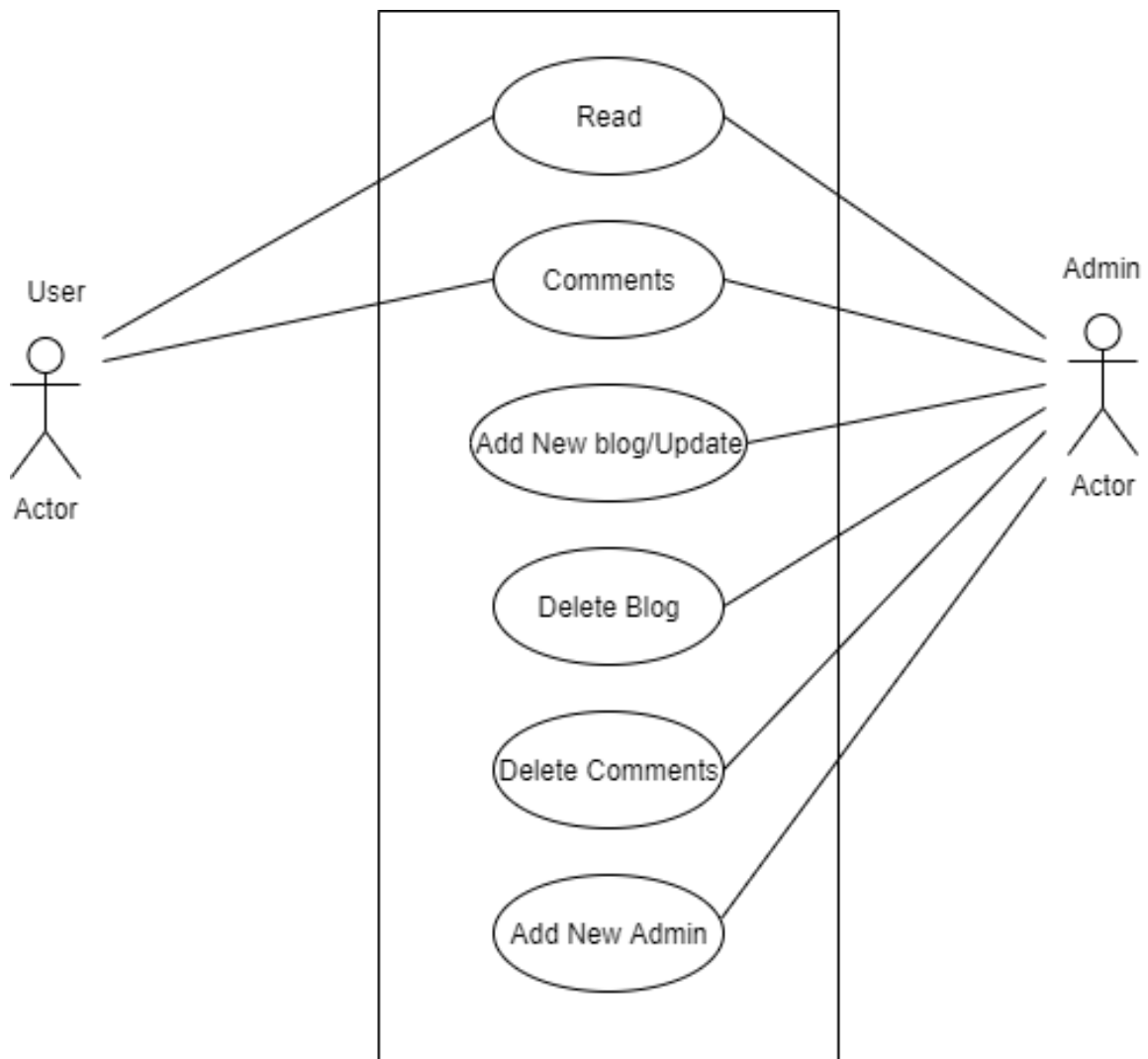


DFD Level 1:



UML DIAGRAM

A use case diagram is used to represent the dynamic behavior of a system. It encapsulates the system's functionality by incorporating use cases, actors, and their relationships. It models the tasks, services, and functions required by a system/subsystem of an application. It depicts the high-level functionality of a system and also tells how the user handles a system.

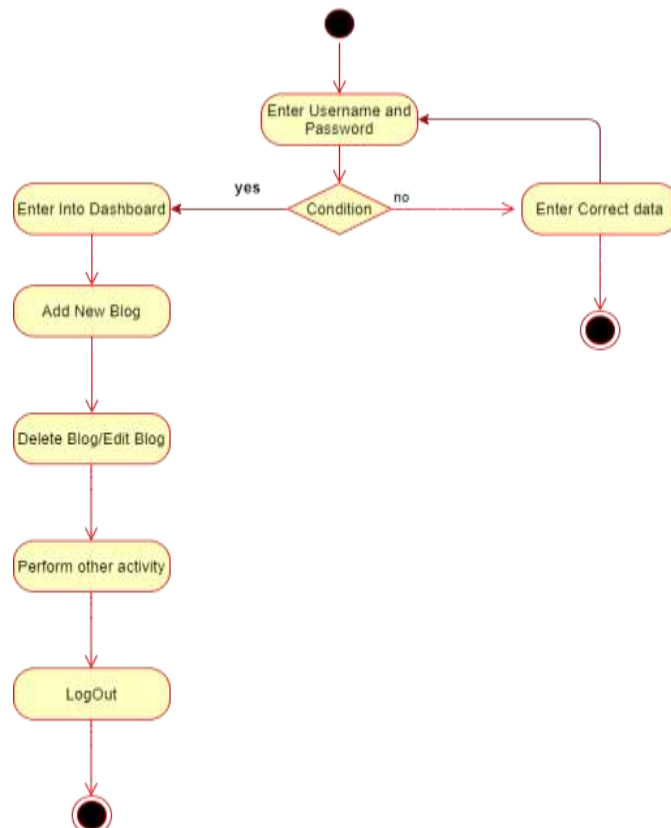


ACTIVITY DIAGRAM

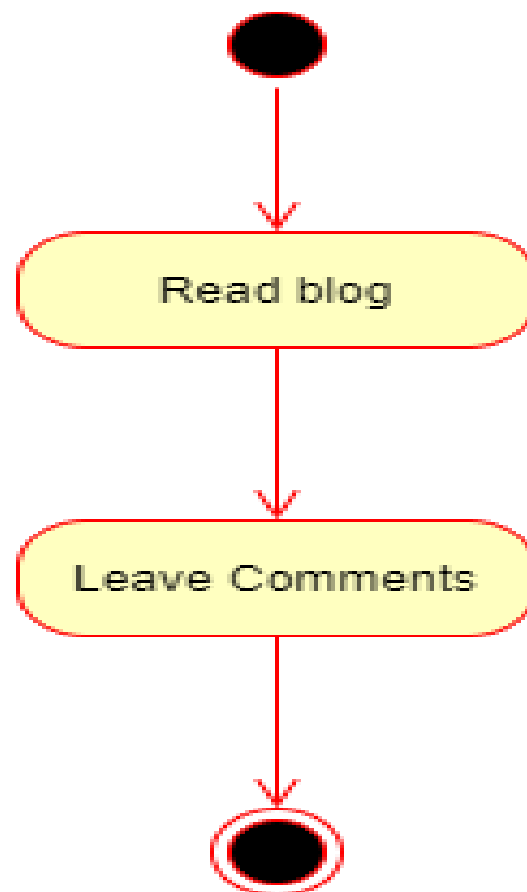
The basic purposes of activity diagrams is similar to other four diagrams. It captures the dynamic behavior of the system. Other four diagrams are used to show the message flow from one object to another but activity diagram is used to show message flow from one activity to another.

Activity is a particular operation of the system. Activity diagrams are not only used for visualizing the dynamic nature of a system, but they are also used to construct the executable system by using forward and reverse engineering techniques. The only missing thing in the activity diagram is the message part.

Admin Activity Diagram



User Activity Diagram:



DATABASE

Blog Comment Table:

 name	varchar(80)	"name" varchar(80) NOT NULL
 email	varchar(254)	"email" varchar(254) NOT NULL
 body	text	"body" text NOT NULL
 created_on	datetime	"created_on" datetime NOT NULL
 post_id	integer	"post_id" integer NOT NULL
 active	bool	"active" bool NOT NULL

Blog Post Table:

 id	integer	"id" integer NOT NULL
 title	varchar(200)	"title" varchar(200) NOT NULL UNIQUE
 slug	varchar(200)	"slug" varchar(200) NOT NULL UNIQUE
 updated_on	datetime	"updated_on" datetime NOT NULL
 content	text	"content" text NOT NULL
 created_on	datetime	"created_on" datetime NOT NULL
 status	integer	"status" integer NOT NULL
 author_id	integer	"author_id" integer NOT NULL

SYSTEM TESTING

UNIT TESTING:

The unit testing of the source code has to be done for every individual unit of module that was developing part of the system and some errors were found for every turn and rectified. This form of testing was use to check for the behavior signified the working of the system in different environment as an independent functional unit. In the software development process Unit Tests basically test individual parts (also called as Unit) of code (mostly methods) and make it work as expected by programmer. A Unit Test is a code written by any programmer which test small pieces of functionality of big programs. Performing unit tests is always designed to be simple, A "UNIT" in this sense is the smallest component of the large code part that makes sense to test, mainly a method out of many methods of some class.

Why do we need Unit test?

One of the most valuable benefits of using Unit Tests for your development is that it may give you positive confidence that your code will work as you have expected it to work in your development process. Unit Tests always give you the certainty that it will lead to a long term development phase because with the help of unit tests you can easily know that your foundation code block is totally dependable on it.

Unit Testing:

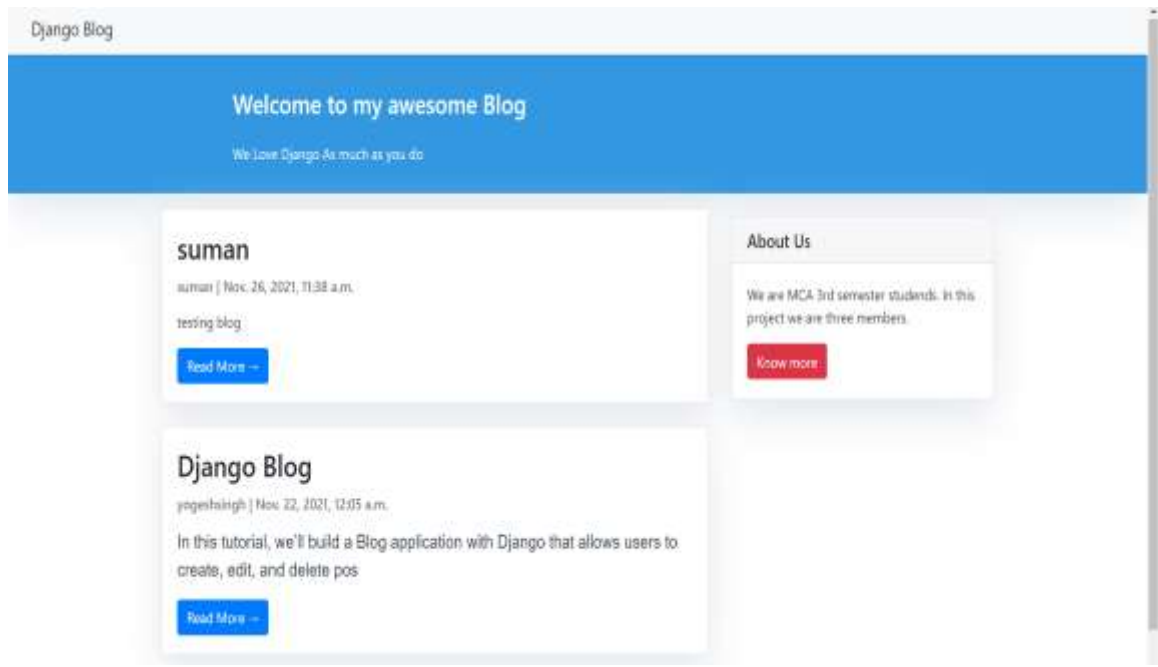
Sr. No.	Test Cases	Data Input	Expected Output	Actual Output	Pass/Fail
1	Add post module	Add a new post	New post showed up in the blog home page as the latest post.	New post showed up in the blog home page as the latest post.	Pass
2	Manage posts module	Edit/Delete a Post	The post info is changed or the post is removed from the blog home page.	The post info is changed or the post is removed from the blog home page.	Pass
3	Add comment module	Add a comment on a post	The comment goes to admin for approval.	The comment goes to admin for approval.	Pass

INTEGRATION TESTING:

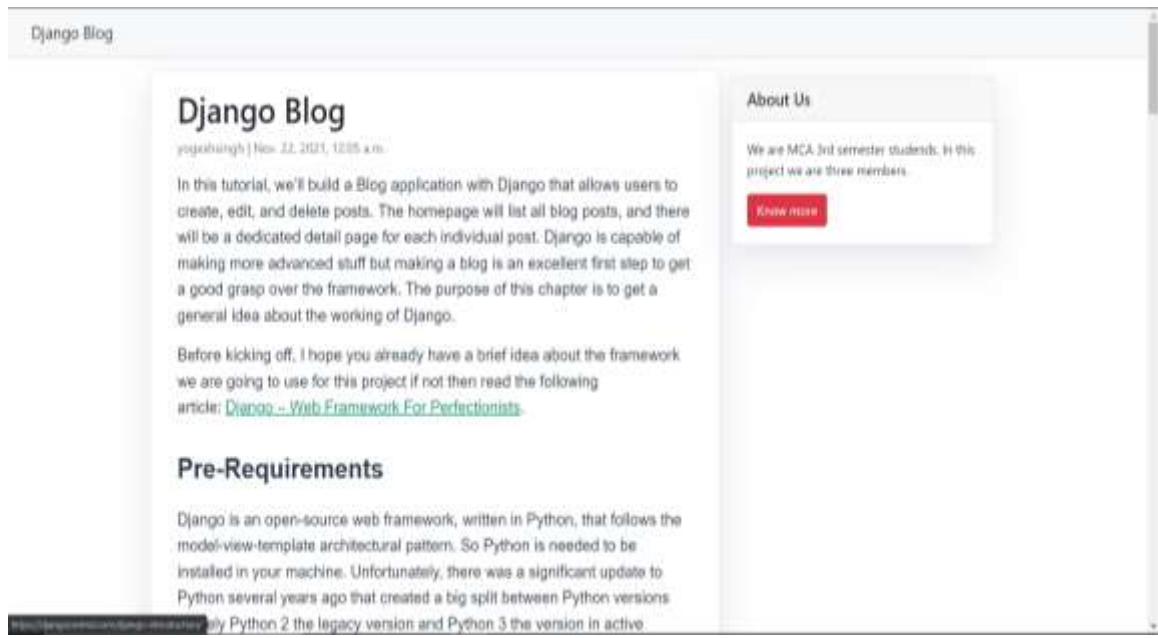
From the individual parts to the cohesion of each part to make the system as a whole, there is need to test the working between the assembled modules of the system. The modules are integrated to makes up the entire system. The testing process is concerned with finding errors that result from unanticipated interaction between the sub-system and system component. It is also concerned with validating the system meets its functional and non-functional requirement. Integration tests ensure that an app's components function correctly at a level that includes the app's supporting infrastructure, such as the database, file system, and network. PYTHON supports integration tests using a unit test framework with a test web host and an in-memory test server.

Screenshot

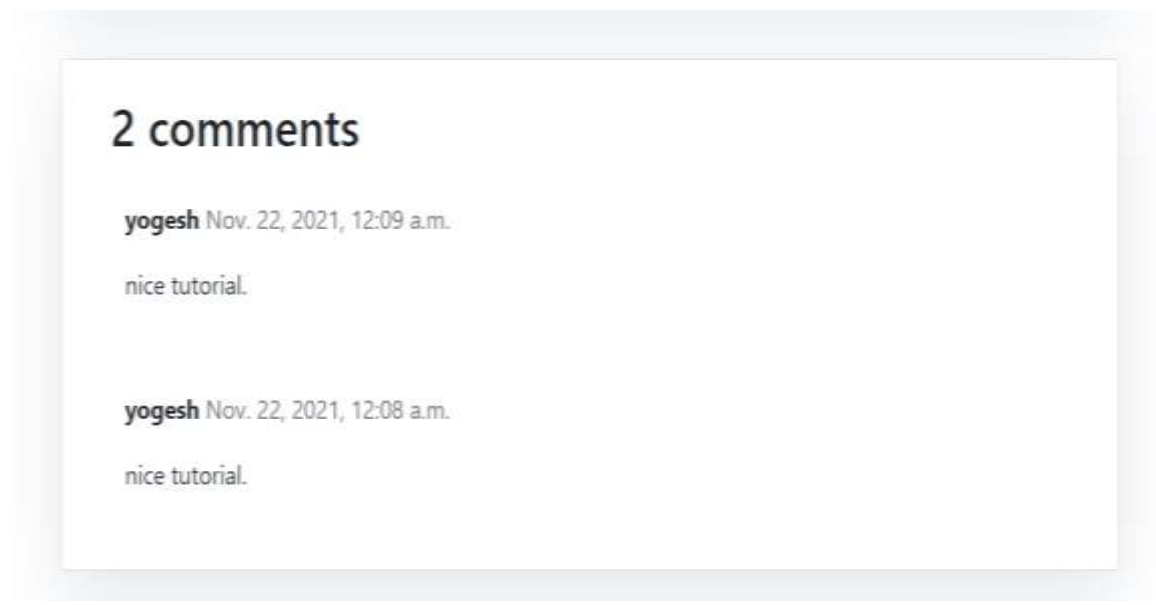
Home Page:



Blog Page:



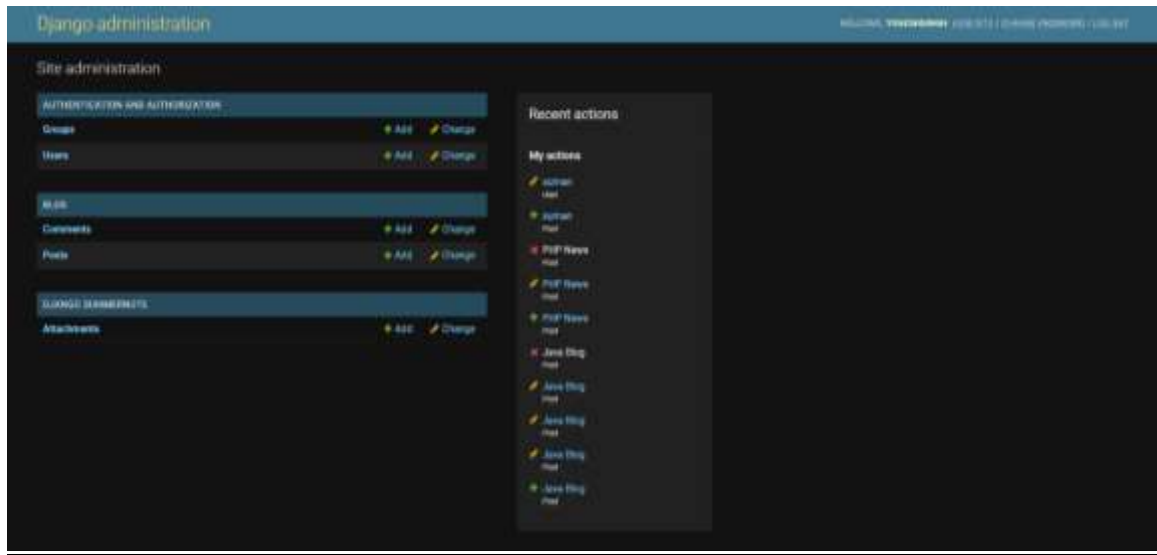
Total Comments:



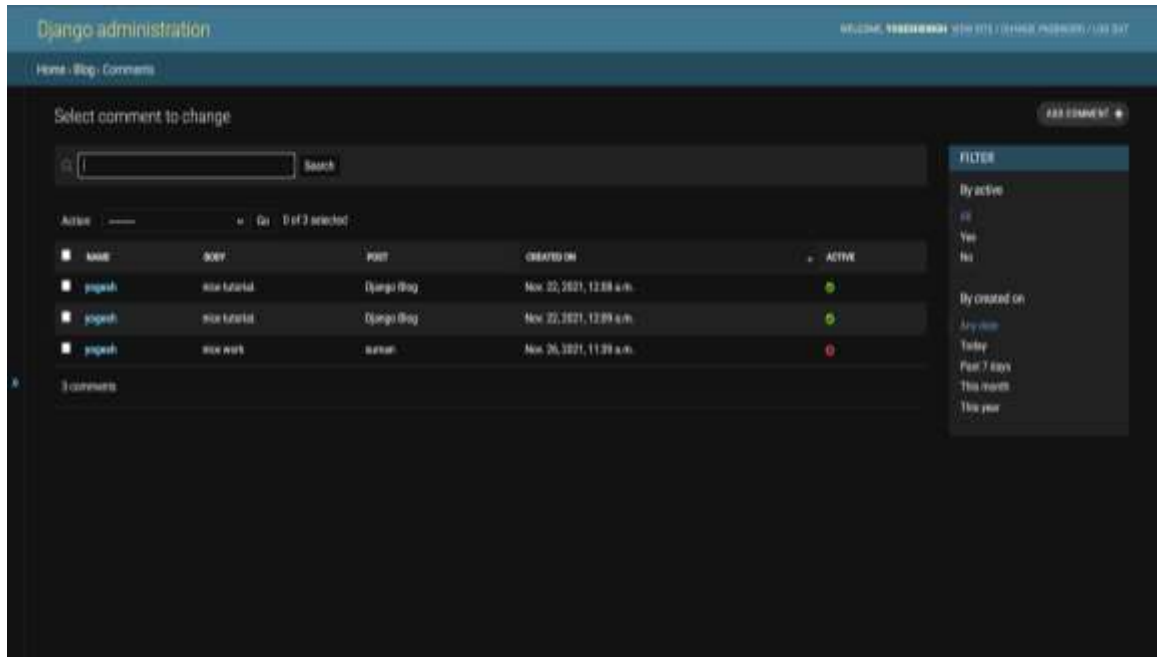
Leave Comments:

A screenshot of a "Leave a comment" form. The form is titled "Leave a comment" and contains three input fields: "Name*", "Email*", and "Body*". The "Body*" field is a larger text area. At the bottom left of the form is a blue "Submit" button. The form is presented in a light gray box with a subtle shadow.

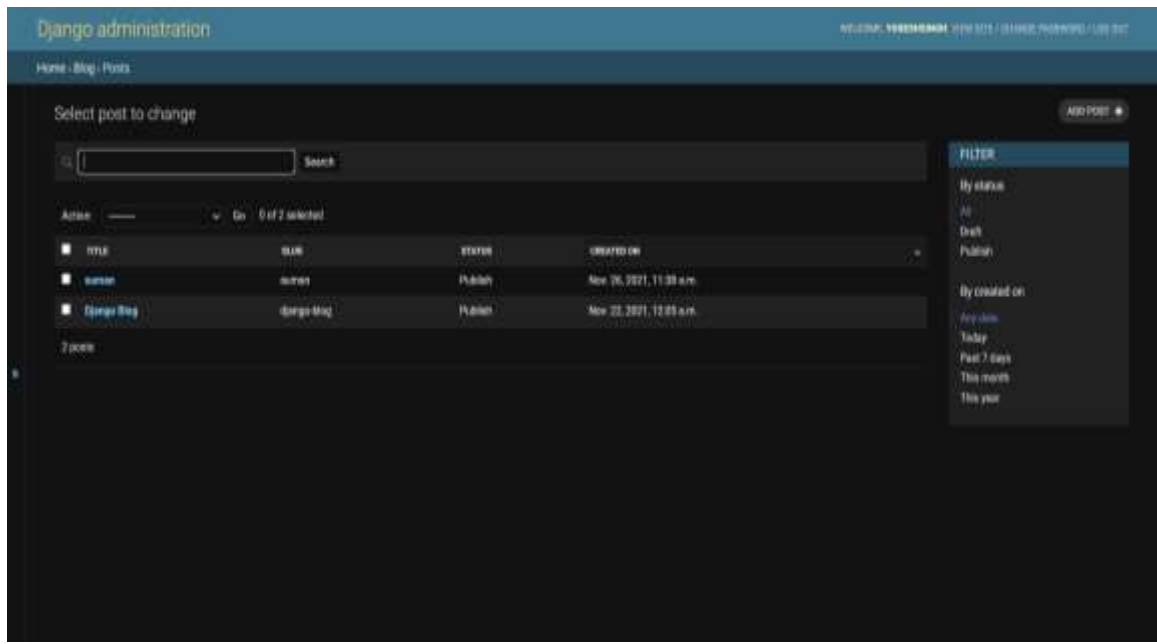
Admin Page:



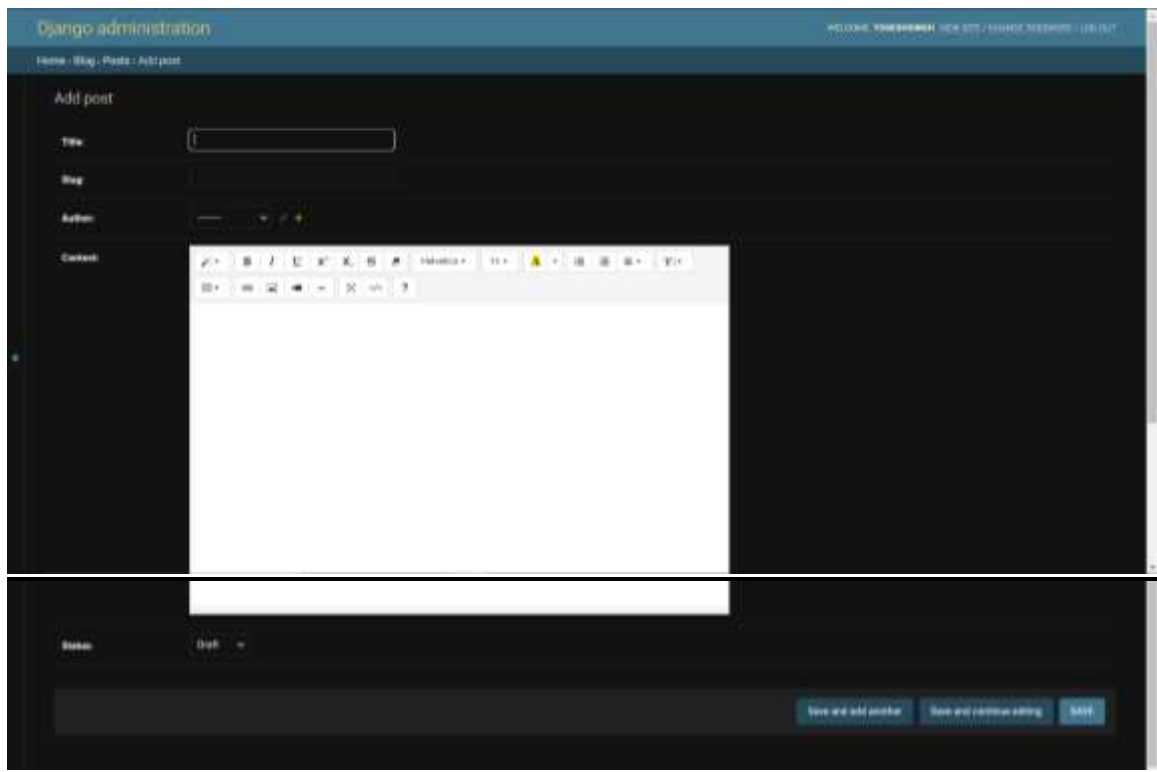
Total Comments(Admin Panel):



Total Blog(Admin Panel):



Add New Blog(Admin Panel):



Maintenance

After the program is completed, the program still needs long term maintenance to make it available and stable to execute. The program will be test after a certain period of time and debug each of the function and possible bugs, whenever a potential bug is detected; the program needs to be refined to a better design. Meanwhile, there will update and add more data to the database to increase the database capacity. Depending on the new keywords, responses, relevant data found that could be applied in this program; the database will always be improved and can handle more and more cases.

Recommendations for Further Work

Future enhancement means that the scope in the future of the system.

- Try to make as more user friendly as possible.
- We are going to integrate Chat server with the Application.
- Try to add User Login system to save favorite blog for future reading purpose

Conclusions

Depending on the requirements and the budget, CMS required for a system can either be built from scratch or one can use an existing open source or commercial product. The key feature to keep in mind while building a Web based CMS is intuitive and user-friendly administration.

Project experience & teamwork:

Apart from the program, we as the developers have improved a lot from the 0 level project. It is quite different from what we previously experienced in the working model, volume of tasks, and the problems we have encountered. In conclusion, we have been improved a lot from the project development, and gained development experience as well as programming skills; the most important is work as a team for a long term, challenge development.

REFERENCE AND BIBLIOGRAPHY

Websites referred:

- ♣ www.stackoverflow.com
- ♣ www.pythonprogramming.net
- ♣ www.codecademy.com
- ♣ www.tutorialspoint.com
- ♣ www.google.co.in