**Blogging Platform**

## Submitted in Partial Fulfilment of the Requirements for the Degree of

**Master of computer Applications**

# by

## AKASH GAUR

**(Roll No.1900290140003)**

**JAI SINGH**

**(Roll No. 1900290140017)**

**Under the Supervision of**

**Mr. Naresh Chandra**

**Associate Professor**

**KIET Group of Institutions, Ghaziabad**



**to the**

**Department of Computer Applications**

**Dr. APJ Abdul Kalam Technical University LUCKNOW, INDIA**

**DECEMBER 2021**

**DECLARATION**

I hereby declare that the work presented in this report entitled “Blogging Platform", was carried out by me. I have not submitted the matter embodied in this report for the award of any other degree or diploma of any other University or Institute. I have given due credit to the original authors/sources for all the words, ideas, diagrams, graphics, computer programs, experiments, results, that are not my original contribution. I have used quotation marks to identify verbatim sentences and given credit to the original authors/sources. I affirm that no portion of my work is plagiarized, and the experiments and results reported in the report are not manipulated. In the event of a complaint of plagiarism and the manipulation of the experiments and results, I shall be fully responsible and answerable.

Name :

Roll. No. : Branch :

(Candidate Signature)

## CERTIFICATE

Akash Gaur (Roll No.1900290140003) has carried out the project work presented in **this project entitled “Blogging Platform”** for the award of Master of Computer Applications from **Dr. A.P.J Abdul Kalam Technical University, Lucknow** under our supervision. The project embodies results of original work, and studies are carried out by the student himself and it is an authentic work carried out by me at **KIET Group of Institutions, Ghaziabad** and the contents of the report do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

Mr. Naresh Chandra

Associate Professor External Examiner

KIET Group of Institutions Ghaziabad

Dr. Ajay Kumar Shrivastava Prof & HOD Department of Computer Applications KIET Group of Institutions

Ghaziabad

## CERTIFICATE

JAI SINGH (Roll No.190029014017) has carried out the project work presented in **this project entitled “Blogging Platform”** for the award of Master of Computer Applications from **Dr. A.P.J Abdul Kalam Technical University, Lucknow** under our supervision. The project embodies results of original work, and studies are carried out by the student himself and it is an authentic work carried out by me at **KIET Group of Institutions, Ghaziabad** and the contents of the report do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

Mr. Naresh Chandra

Associate Professor External Examiner

KIET Group of Institutions Ghaziabad

Dr. Ajay Kumar Shrivastava Prof & HOD Department of Computer Applications KIET Group of Institutions

Ghaziabad

## ACKNOWLEDGEMENT

The attitude within us during the academic period of MCA is a project culmination of a task undertaken by me. Acknowledgement is not a mere formality or ritual but a genuine opportunity to express the indebtedness to all those without whose active support and encouragement this project wouldn’t have been possible.

One of the most pleasing aspects in collecting the necessary information and compiling it is the opportunity to thank those who have actively contributed to it.

We are extremely grateful to **Mr. Naresh Chandra** for his revered guidance and encouragement, which led to the completion of this project so far. Without his constant appraisal and efforts, this task would have been a mere dream. He was always there to help us throughout this project. He provides us with all the necessary resources and guidance during the project which helped us to complete the project successfully. We are also thankful to our colleagues and friends for their support.

Our special thanks to all faculty members and my friends for their active support, affectionate guidance and constant encouragement.

We are highly indebted to **DR. AJAY KUMAR SRIVASTAV** for his continuous effort in building good infrastructure and developing a professional.

**Name of students:**

* AKASH GAUR
* JAI SINGH

## ABSTRACT

Blogging Platform app is an integral part of the smart blogging platform. There are many blogging systems that are widely available to educational institutions and many more sectors. The challenge is to easily integrate the in-people thoughts and ideas into a diversified environment and maintain social communication based on the intimations among students, faculty, staff etc.

The blogging services rely on a software system that allows access to all the thoughtful cards for the educational and general thoughts and explores and makes them electronically available to all the students, colleagues, staff etc on the Internet whenever they want they can open and explore as well as create a new one.

The design and development of blogging systems is a critical part of the educational Social process as it reflects on the usage of the system. In this work, the design and implementation of blogging systems are described where different techniques are explored and compared.

The proposed tech-blogging system is designed using off-the-shelf and open-source software engineering models and programming tools and database models.

The system is tested to prove the new design concepts and features. The method used in the back end and front-end design and implementation allows flexible usage and integration of the blogging implementation systems by the educational institutions, small and large-scale industries etc in smart cities.

**Table of Contents**

|  |  |
| --- | --- |
| **CHAPTERS** | **PAGE**  **NO.** |
| * **INTRODUCTION**   + Purpose   + Overview     - Module     - Blogger Registration     - Blogger Login | **1-2**  1  1  1  2  2 |

|  |  |
| --- | --- |
| * Administrator Login   + Administrator’s Rights | 2  2 |
| 1. **FEASIBILITY STUDY**    * Introduction    * Technical Feasibility    * Economic Feasibility    * Operational Feasibility  * Implementation | **3-5**  3  3  3-4  5  5 |
| * **SOFTWARE REQUIREMENT SPECIFICATION**   + Software Functional Requirement   + Hardware Environment   + Software Interfaces   + Software Limitation   + Specification Report   + User Interface * When to Use SQL | **6-8**  6  6  6  7  7  8  8 |
| * **DESIGN**   + Architectural Diagram   + Tables and Dependencies   + Dynamic Model     - Sequence Diagram * Activity Diagram | **9-16**  9  10  11-16  11-12  13-16 |
| * **INPUT/OUTPUT**   + User End     - User Homepage     - User Registration * User Login | **17**  **- 33**  17-19  17  18  19 |

|  |  |
| --- | --- |
| * Courses * Latest Exams | 21 |
| * Speed Test | 22 |
| * Contact | 23 |
| * Admin End | 24 |
| * Login | 20-27 |
| * Dashboard | 25 |
| * Category | 26 |
| * Sub-category | 27-28 |
| Manager 5.2.5 | 29 |
| * Policy Sold | 30 |
| * Users | 30 |
| * Reports | 31 |
| * Subscriber Manager | 32 |
| * Header and Footer Manager | 33 |

|  |  |
| --- | --- |
|  | 33-34 |
| * **TESTING**   + Unit Testing   + Integration Testing   + Validation Testing   + User Acceptance Testing   + Black Box Testing   + ScreenShots     - Test Case 1     - Test Case 2     - Test Case 3 * Test Case 4 | **35-41**  35-36  36-37  37  37-38  38-39  40-41  40  40  41  41 |
| * **IMPLEMENTATION**   + Introduction   + Software Implementation Challenges   + Some Coding Parts     - Connect Database     - User Login * Compare Results | **42-45**  42  42-43  43-45  43  44  45 |
| **8. BIBLIOGRAPHY** | **46** |

4

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

**TABLES AND FIGURES**

**TABLES**

Table 2.1 Cost Estimation

**FIGURES**

Fig. 4.1 Architectural Diagram Fig.

* 1. Tables and Dependencies Fig.
  2. login Sequence Diagram

Fig. 4.4 Registration Sequence Diagram Fig. 4.5 Login Activity Diagram

Fig. 4.6 Registration activity diagram

Fig. 4.7 Record payment activity diagram Fig. 4.8 Generate report activity diagram Fig. 5.1 Homepage

Fig. 5.2 User-registration

Fig. 5.3 User-login Fig. 5.4 User Profile

Fig. 5.5 Compare Plans Fig. 5.6 Detail Filling Fig.

* 1. Compare Result Fig.
  2. Payment

Fig. 5.9 Admin Login 25

Fig. 5.10 Admin 26

Dashboard Fig. 5.11 27

Category list 27

Fig. 5.12 Add 28

Category Fig. 5.13 29

Edit Category 29

Fig. 5.14 Sub- Category list 30

Fig. 5.15 Add Sub- Category 31

Fig. 5.16 Insurance Plan 32

List Fig. 5.17 Policy Sold 32

Fig. 5.18 Users 33

list Fig. 5.19 33

Reports 34

Fig. 5.20 Subscribers 40

List Fig. 5.21 Change 40

Header Fig. 5.22 Change 41

Footer Fig. 6.1 41

Login Error 43

Fig. 6.2 Successful User 44

Login Fig. 6.3 Buy 45

Button

Fig. 6.4 Auto-load Subcategory Fig. 7.1 Database Connection Code Fig. 7.2 User Login Code

Fig. 7.3 Compare Result Code

## CHAPTER INTRODUCTION

* **Purpose**

This system is a Blogging web portal. Our website features courses for various exams and we help students to learn every subject related to their course through our video tutorials, we also provide study notes which will help students at every moment of their life to achieve success.

Students can come one closer to their closed ones either they are colleagues and faculty members more. professionally as there explores through Blogging Platform.

## OVERVIEW OF THE PROJECT SYSTEM DESCRIPTION:

The proposed system is for making it easier to manage blogs along with minimalist UI/UX. This software is powered & facilitated to all those bloggers who’re not connoisseurs of technical speciality.

## MODULES:

* + Admin
  + Users
  + public website

## BLOGGER REGISTRATION: -

The form contains the customer information like

* + Name
  + Gender
  + Address
  + City
  + District
  + Phone
  + Log In Id
  + Password
* Achievements
* Upload Resume

## BLOGGER LOGIN: -

The form contains the student information like

* Log In
* Password
* Select Role as Blogger

## ADMINISTRATION LOGIN: -

The form contains the admin information like

* Username
* Password
* Select Role As Administrator

## ADMINISTRATOR’S RIGHTS

* + View / Delete Users.
  + View / Add / Delete Course Categories.
  + View / Add / Delete Course Sub-Categories.
  + View / Add / Delete Courses.
  + View Blogger Registration Request.
  + Verify & Provide the access to Blogger
  + Send Notifications to Subscribers.
  + Publish & Unpublish the Blogs as per the Guideline
  + Remove the Blogger Access
  + Change Header.
  + Change Footer.

## CHAPTER 2 FEASIBILITY STUDY

* **INTRODUCTION**

A feasibility study is conducted once the problem is clearly understood. A feasibility study is a high-level capsule version of the entire system analysis and design process. The objective is to determine quickly at a minimum expense how to solve a problem. The purpose of feasibility is not to solve the problem but to determine if the problem is worth solving. The system has been tested for feasibility in the following points.

* + Technical Feasibility
  + Economic Feasibility
  + Operational Feasibility

## TECHNICAL FEASIBILITY

Project Blogging Platform is a complete web-based application. The main technologies and tools that are associated with Blogging Platform are

* + HTML
  + CSS
* Angular
* Typescript
* PHP
* MySQL

Each of the technologies is freely available and the technical skills required are manageable. Time limitations of the product development and the ease of implementing using these technologies are synchronized. Initially, the website will be hosted in a free web hosting space, but for later implementations, it will be hosted in a paid web hosting space with sufficient bandwidth. The bandwidth required in this application is very low since it incorporates very few multimedia aspects.

From there it’s clear that the project IMS is technically feasible.

## ECONOMIC FEASIBILITY

Economic feasibility is the process of identifying the financial benefits and costs associated with the project being developed. So, the project is economically

feasible because the project reduces the cost of the resources. But Economic Feasibility has been expressed as a cost-benefit analysis.

Costs- our system uses new technology and has a centralized database that cannot need more resources. It requires a minimum amount of cost. The estimated cost of resources that we use to develop this project.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Types of costs** | **Tool name** | **Quantity** | **Unit price (in Rs.** | **Total price (in Rs.)** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Hardware costs** | Computer | 1 | 20000 | 20,000 |
| Flash (8 GB) | 1 | 300 | 300 |
|  | Paper | 1 | 100 | 100 |
| Printing and binding | 2 | 300 | 600 |
| **Software costs** | Xampp server | 1 | Free | Free |
| Microsoft office 2013 | 1 | Free | Free |
| Vscode | 1 | Free | Free |
|  | Windows 8.1 OS | 1 | Free | Free |
| **Requirement analysis** |  |  |  | 1,000 |
| **Total** |  |  | 22,000 | |

Table 2.1 cost estimation

## OPERATIONAL FEASIBILITY

The system performs all operations to achieve the specified objective, User friendly and interactive with the environment and the system will perform all operations that the organization runs. And it will not have any difficulty or procedures to perform the operation of the system. So, the project is operationally feasible.

## IMPLEMENTATION

Implementation is the stage where the theoretical design is turned into a working system. The most crucial stage in achieving a new successful system and in giving confidence on the new system for the users that it will work efficiently and effectively.

The system can be implemented only after thorough testing is done and if it is found to work according to the specification.

It involves careful planning, investigation of the current system and its constraints on implementation, design of methods to achieve the changeover and an evaluation of change over methods apart from planning. Two major tasks of preparing the implementation are education and training of the users and testing of the system.

The more complex the system being implemented, the more involved will be the systems analysis and design effort required just for implementation.

The implementation phase comprises several activities. The required hardware and software acquisition is carried out. The system may require some software to be developed. For this, programs are written and tested. The user then changes over to his new fully tested system and the old system is discontinued.

# CHAPTER 3

**SOFTWARE REQUIREMENT SPECIFICATION**

## SOFTWARE FUNCTIONAL REQUIREMENT

|  |  |
| --- | --- |
| **Language** | HTML, PHP, TYPESCRIPT |
| **Database** | MySQL |
| **Platform** | ANGULAR + PHP |

* **HARDWARE ENVIRONMENT**
  + Processor Speed: 2.7GHz
  + RAM: 8GB
  + Hard Disk: 500
* Keyboard: 104 keys.
  + Processor: Intel(R) i5

## SOFTWARE INTERFACES

The interface must be easy to understand. The user interface includes: -

* + Standards for fonts, icons, button labels, images, colour schemes, field tabbing sequences, commonly used controls.
  + Screen layout or resolution constraints.
  + Standard buttons, functions that will appear on every screen, such as an Insert button.
  + Layout standards to facilitate software.
  + Accommodations for visually impaired users.

## SOFTWARE LIMITATION

Principles of good programming practice are not always followed in these sample programs in order to provide more concise, more readable code. In particular:

Only limited error responses are shown. Working programs should always check returned error codes and perform appropriate actions when an error is encountered.

Only limited memory and resource management is done. In working programs, all keys and hashes should be destroyed, all allocated memory should be freed, all files should be closed, and all handles should be released. These example programs provide only limited demonstrations of the use of functions that perform these tasks. These example programs perform no memory or resource management tasks in the case of program termination due to errors.

## SPECIFICATION REPORT

The analysis is a detailed study of the various operations performed by a system and their relationship within and outside the system. The need for developing a new system arises due to the following reasons:

Lack of satisfaction with the performance and effectiveness of the present system. Anticipation of future information requirements. It is the study of the existing system. It is done to find out the facilities being offered by the system as well as its shortcomings in it. This is necessary because the user would expect that the new

the system will not have these shortcomings in it. To meet this objective, it is essential that the people who actually handle the system would be handling the new system are involved in this analysis.

System analysis will be performed to determine if it is feasible to design information based on policies and plans of the organization and on user requirements and to eliminate the weaknesses of the present system.

## USER INTERFACE

All the properties will be shown in the user screen of a student from where he/she will be able to see the various subjects by selecting course categories and sub-categories in the project. He/she can know about the courses.

User can compare the various courses and choose the best one for him/her.

## WHEN TO USE SQL

Use SQL when we know which objects the data resides in, and we want to:

* + Retrieve data from a single object or from multiple objects that are related to one another.
  + Count the number of records that meet specified criteria.
  + Sort results as part of the query.

# CHAPTER 4

## DESIGN ARCHITECTURE DIAGRAM

Fig. 4.1 Architectural Diagram

## TABLES AND DEPENDENCIES AMONG THEM

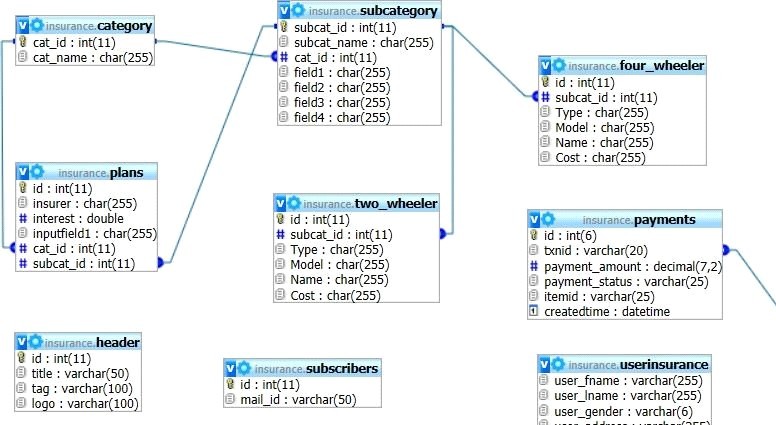
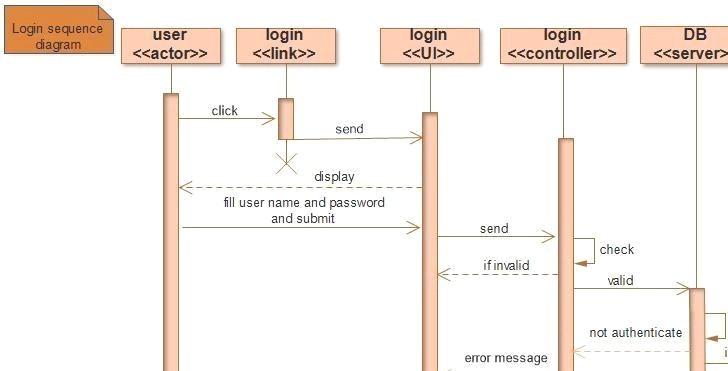


Fig. 4.2 Tables and dependencies

# DYNAMIC MODEL

## Sequence diagram

Sequence diagram showing the sequence of interactions among objects and used to represent or model the flow of messages, events and actions between the objects or components of a system. Sequence Diagrams are also used primarily to design, document and validate the architecture and interfaces of the system by describing the sequence of actions that need to be performed to complete a task or scenario.



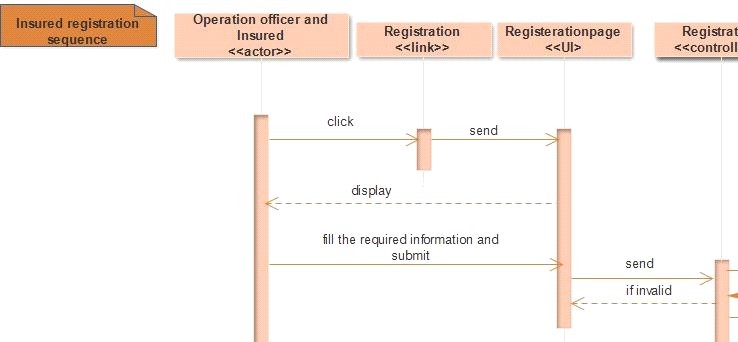


Fig. 4.4 Insured registration sequence diagram

## Activity Diagram

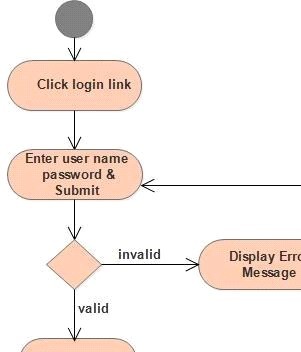


Fig. 4.6 login activity diagram

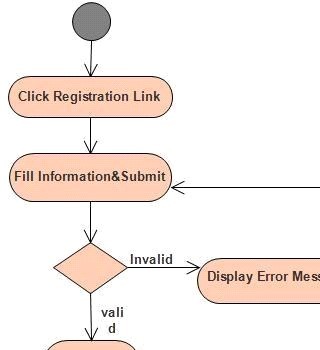


Fig. 4.7 registration activity diagram

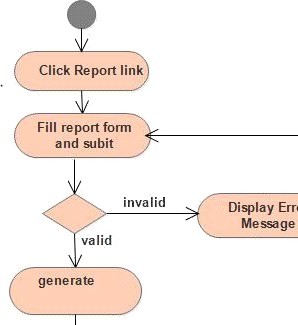


Fig. 4.9 Generate report activity diagram

**CHAPTER 5 INPUT/OUTPUT**

Input/output Form represents the various types of pages which help the users in easy access the website.

In the upcoming screenshots you have a glimpse of the achieved functionality of the Website:

## USER END

This is the webpage that will be available for the customers to interact with and compare and buy their relevant interest insurance.

## USER HOMEPAGE

This is the first page that will be displayed when a user visits our website. It contains links to various pages and also login and register links.



Fig. 5.1 Homepage

## USER REGISTRATION

When a user clicks on the register link, the modal form is splashed over the current page on which the user is right now.

When a user submits the modal form, he isn’t redirected to a new page rather all the processes work in the background and the user never knows how his data is stored, and he/she is registered.

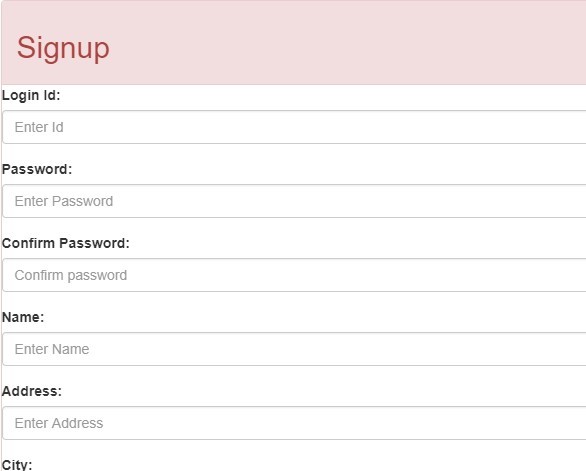


Fig. 5.2 User-registration

## USER LOGIN:

For login also a modal form is splashed over the current page and the user needs to give relevant information i.e., user id and password.

Authenticating to which the user is logged into his/her account. If any error occurs an error message is splashed in the modal form.

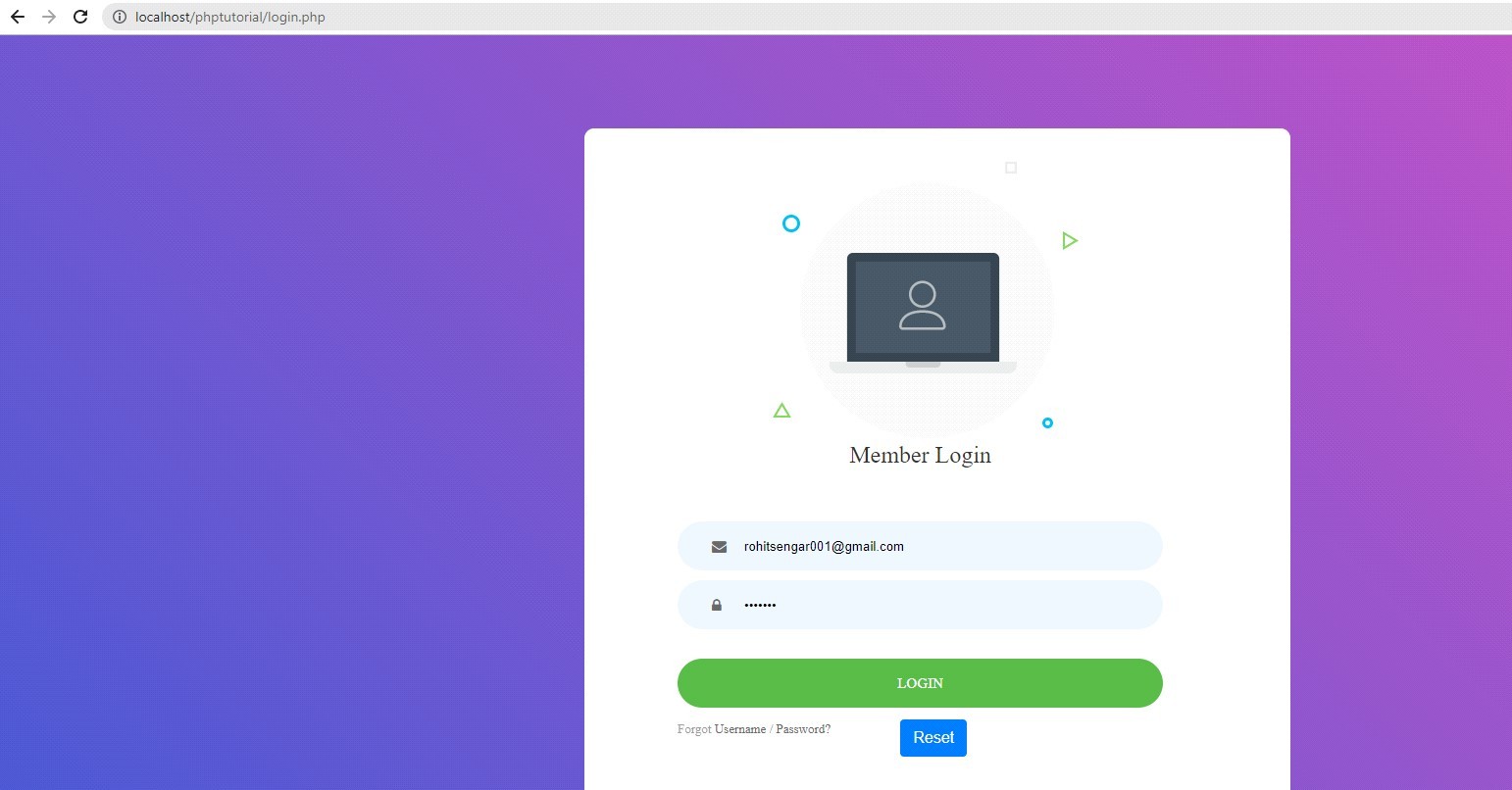


Fig. 5.3 User-login

## AFTER SELECTING CATEGORIES AND SUBCATEGORIES

This is about adding courses and subjects related to that course with full description.

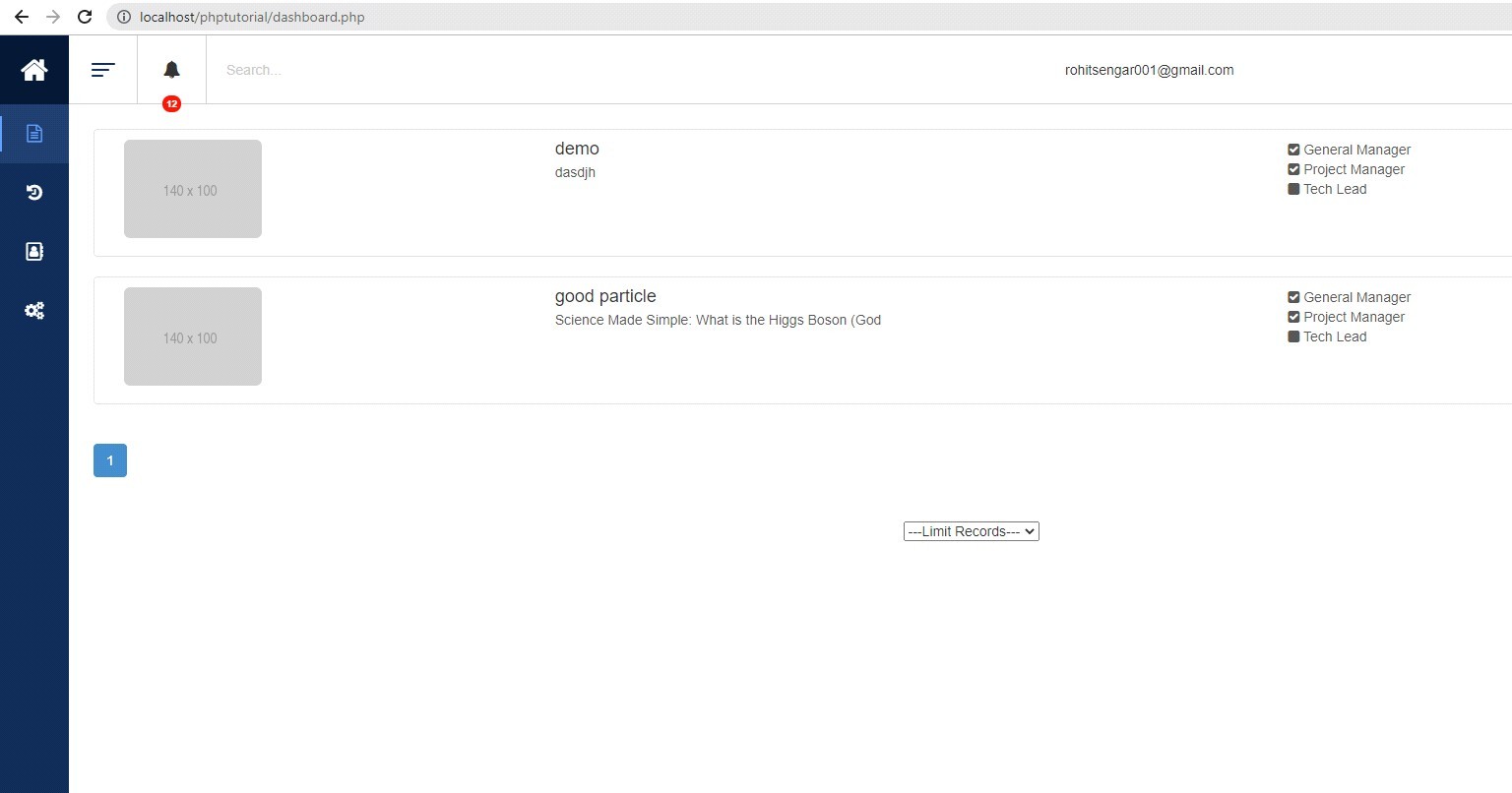


Fig.5.4 After Login

## VARIOUS Blogs:

After giving the information about the category and regarding data the student will facilitate with the various subjects for that blog.

From here, the student can select the best suitable subject of blog for him/her.

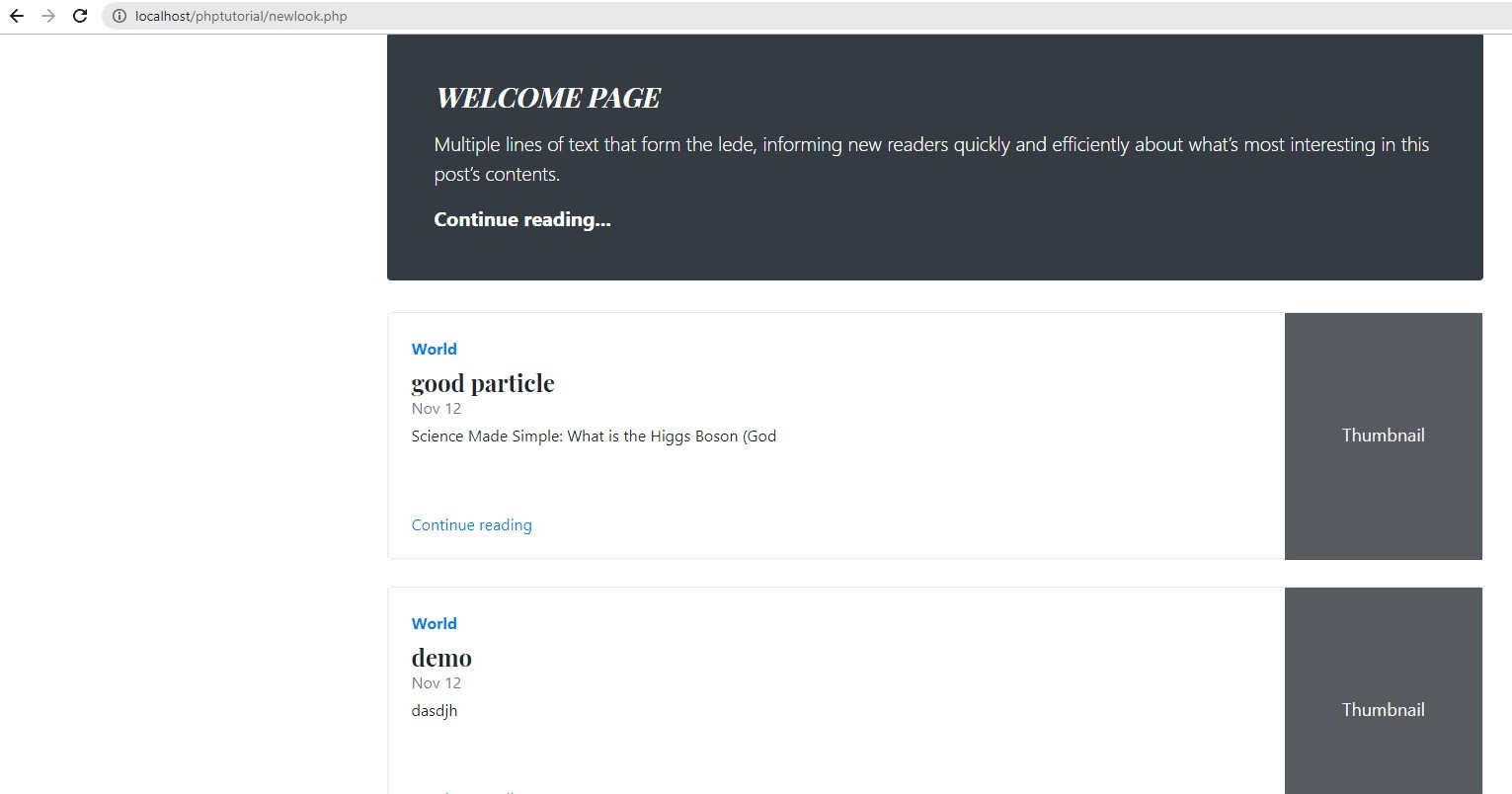


Fig.5.5 Various Subjects

## ADMIN END

This is the separate page for the admin of the page through which all the data on the page and the users of the page will be managed. Admin page is separate means the path for reaching admin will be different so no other user could try to login to the admin. This makes the admin module more secure.

## ADMIN LOGIN:

This is the page that will be opened till we login through this. It means that we cannot open any page of admin even if we copy the path if we

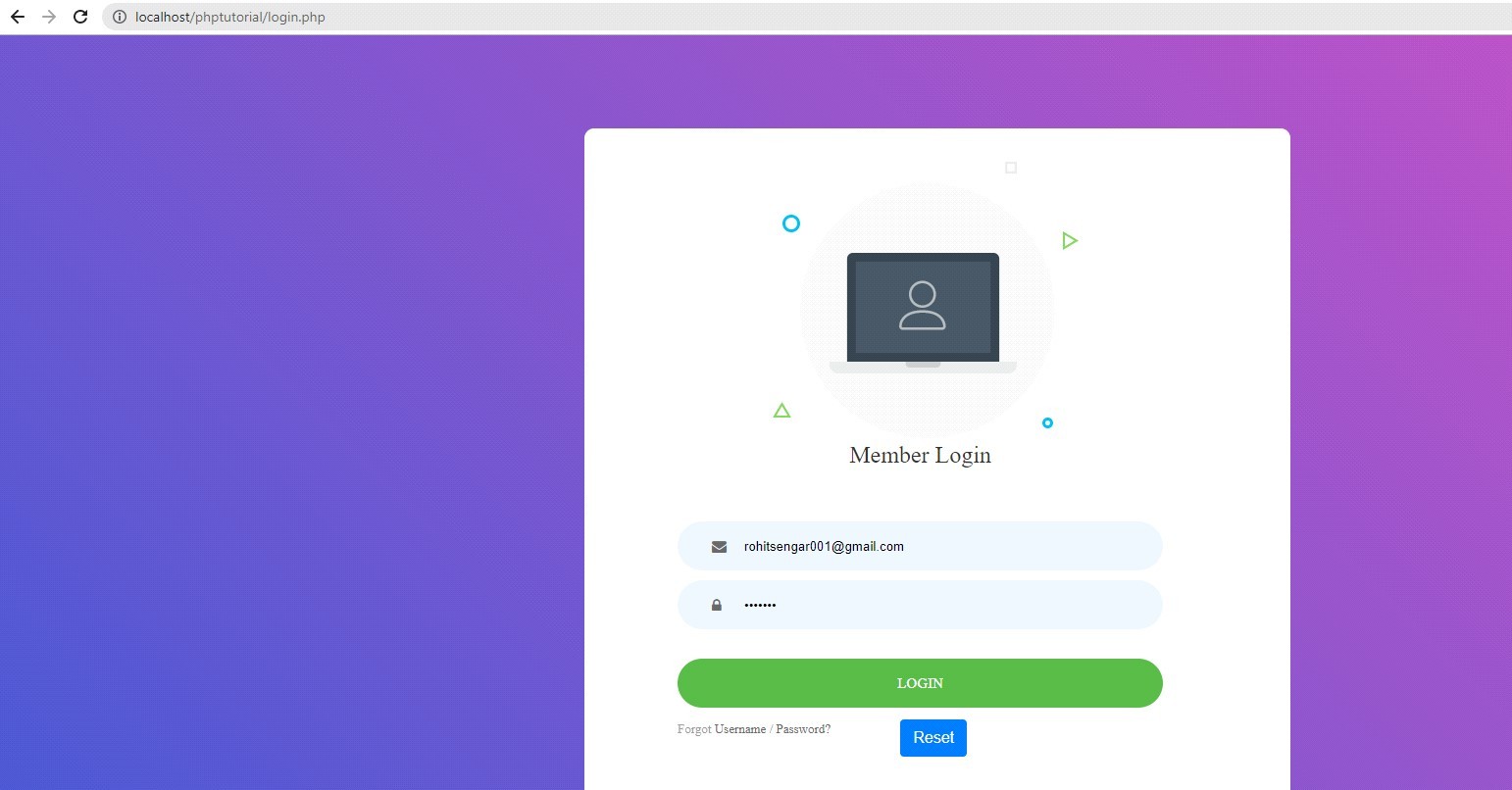
haven’t logged into it.

Fig. 5.6 Admin login

## CATEGORIES:

Category manager, the part of admin which manages the categories i.e., add, delete or update of categories.

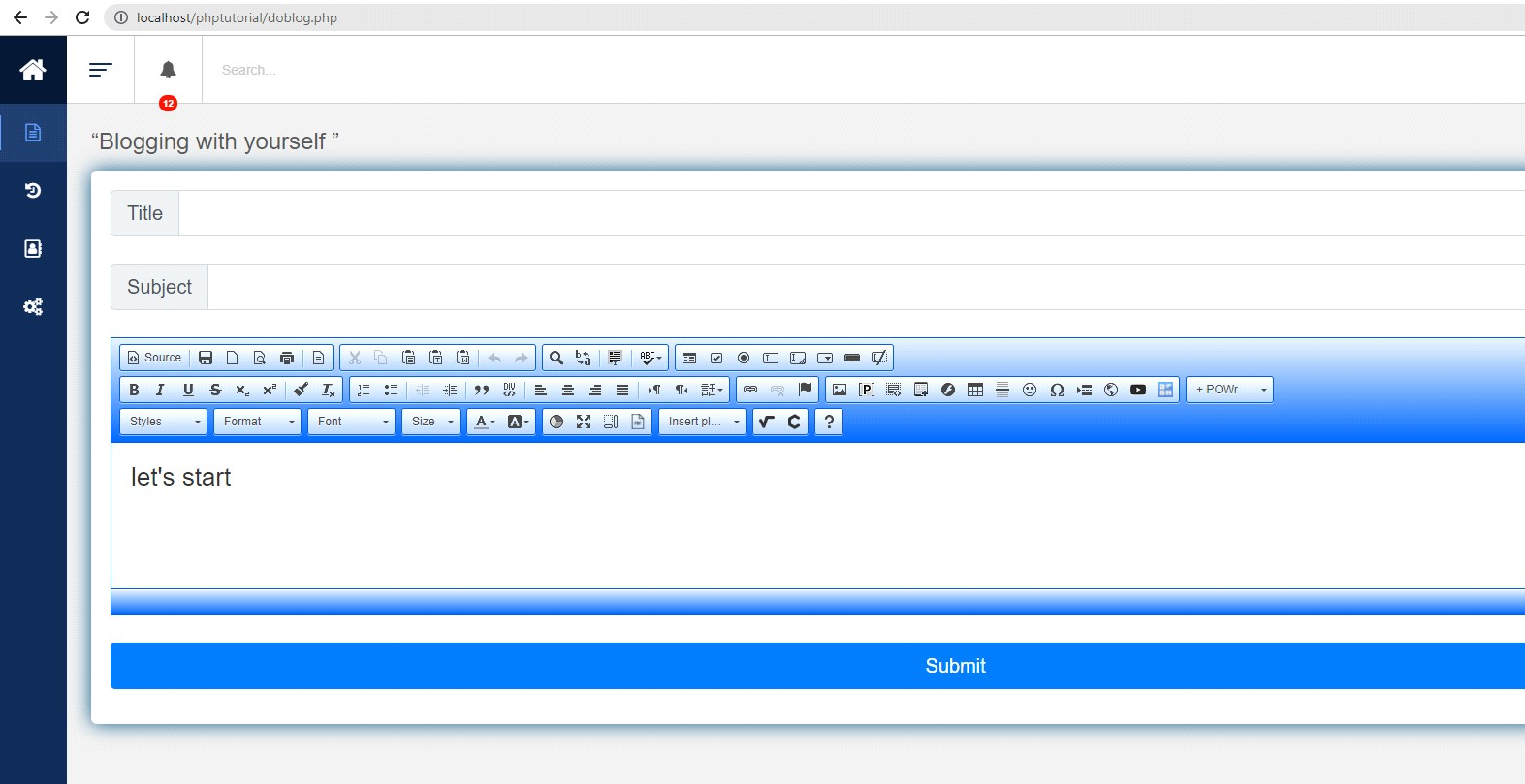


Fig. 5.7 Adding blog

## SUB-CATEGORIES(SUBJECTS):

The part of admin, in which he can add, delete or update subcategories (subjects).

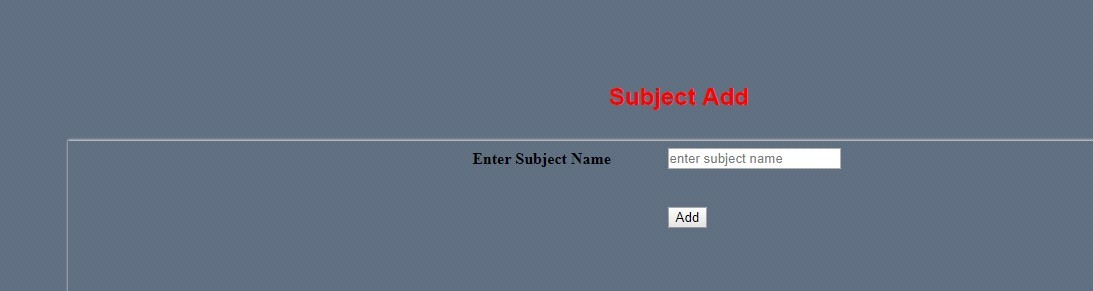


Fig. 5.8 Subcategories

## USERS:

Admin could only view and delete a user but couldn’t modify.

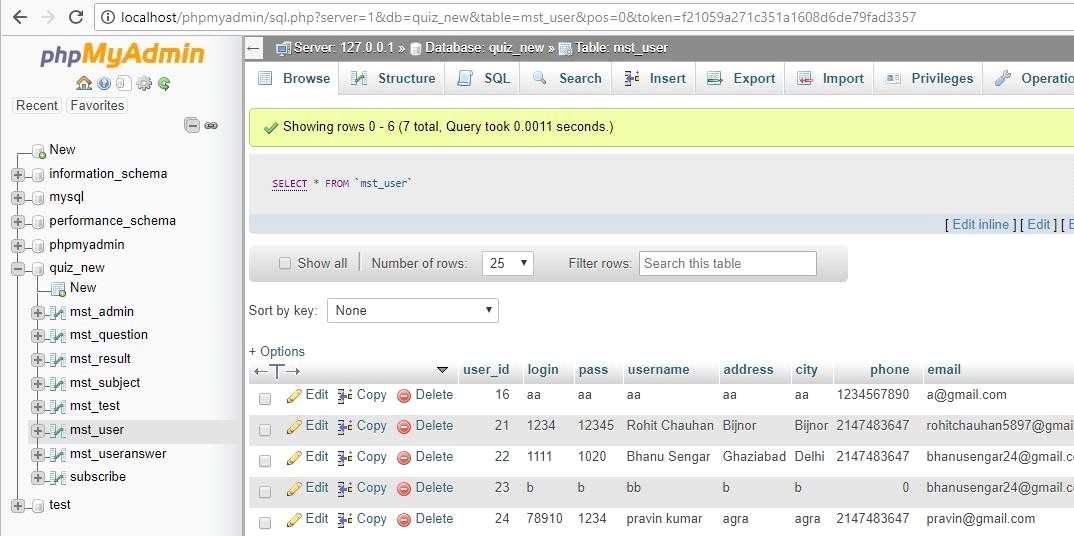


Fig. 5.9 Users list

## REPORTS:

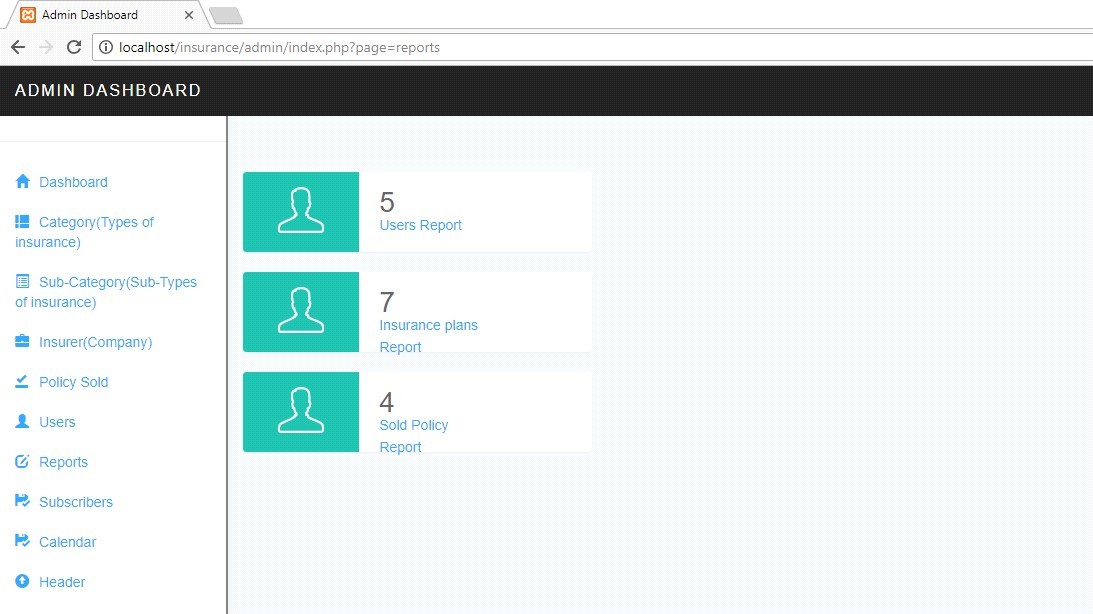


Fig. 5.91 Reports

## SUBSCRIBERS:

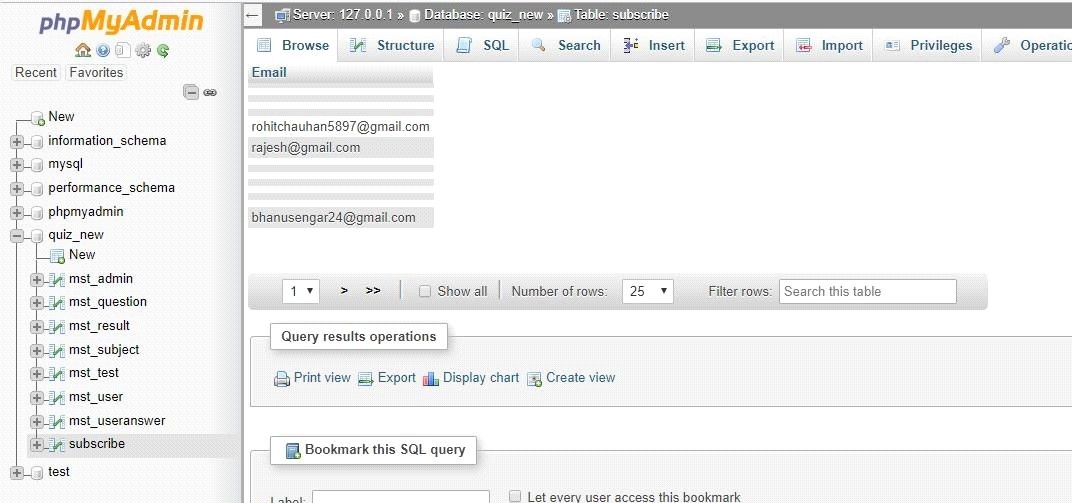


Fig. 5.92 Subscribers list

# CHAPTER 6 TESTING

System testing is the stage of implementation, which is aimed at ensuring that the system works accurately and efficiently before live operation commences. Testing is vital to the success of the system. Testing is the process of executing a program with the explicit intention of finding errors that are making the program fail. The tester may analyst, programmer or a specialist trained for software testing, is actually trying to make the program fail. Analysts know that an effective testing program does not guarantee system reliability. Therefore, reliability must be designed into the system.

## UNIT TESTING

In unit testing, we have to test the programs making up the system. For this reason, unit testing is sometimes called Program testing. The software units in a system are modules and routines that are assembled and integrated to perform a specific function. Unit testing focuses first on modules, independently of one another, to locate errors. This enables, to detect errors in coding and logic that are contained within the module alone. Unit testing can be performed from the bottom up, starting with the lowest level modules and proceeding one at a time. Unit testing is done for each module in Blogging System. This ensures that the value we enter the match with the data type and within the specified limits.

A unit is the smallest testable part of software. It usually has one or a few inputs and usually a single output. In procedural programming, a unit may be an individual program, function, procedure, etc.

This is to be discouraged as there will probably be many individual units within that module. Unit testing increases confidence in changing/ maintaining code. If good unit tests are written and if they are run every time any code is changed, we will be able to promptly catch any defects introduced due to the change. Also, if codes are already made less interdependent to make unit testing possible, the unintended impact of changes to any code is less. Codes are more reusable. In order to make unit testing possible, codes need to be modular. This means that codes are easier to reuse.

Development is faster. How? If you do not have unit testing in place, you write your code and perform that fuzzy developer test‟ (You set some breakpoints,

fire up the GUI, provide a few inputs that hopefully hit your code and hope that you are all set.) If you have unit testing in place, you write the test, write the code and run the test. Writing tests takes time but the time is compensated by the less amount of time it takes to run the tests; You need not fire up the GUI and provide all those inputs. And, of course, unit tests are more reliable than developer tests‟.

Development is faster in the long run too. How? The effort required to find and fix defects found during unit testing is very less in comparison to the effort required to fix effects found during system testing or acceptance testing.

## INTEGRATION TESTING

Data can be lost across any interface, one module can have an adverse effect on another, sub-functions when combined, may not produce the desired major functions. Integration testing is systematic testing for conducting tests to uncover errors associated with the interface. The objective is to take unit tested modules and build a program structure. All the modules are combined and tested as a whole. Here correction is difficult because the vast expanses of the entire program complicate the isolation of causes. Thus, in the integration testing step, all the errors are corrected for the next testing steps.

*Big Bang* is an approach to Integration Testing where all or most of the units are combined together and tested at one go. This approach is taken when the testing team receives the entire software in a bundle.

So, what is the difference between Big Bang Integration Testing and System Testing?

Well, the former test only the interactions between the units while the latter tests the entire system.

*Top-Down* is an approach to Integration Testing where top-level units are tested first, and lower-level units are tested step by step after that. This approach is taken when a top-down development approach is followed. Test Stubs are needed to simulate lower-level units which may not be available during the initial phases.

*Bottom-Up* is an approach to Integration Testing where bottom level units are tested first and upper-level units’ step by step after that. This approach is taken

when the bottom-up development approach is followed. Test Drivers are needed to simulate higher-level units that may not be available during the initial phases.

Sandwich/Hybrid is an approach to Integration Testing which is a combination of Top-Down and Bottom-Up Approaches.

## VALIDATION TESTING

This is the final step in testing. In this, the entire system was tested as a whole with all forms, code, modules and class modules. This form of testing is popularly known as Black Box testing or System tests.

The Black Box testing method focuses on the functional requirements of the software. That is, Black Box testing enables the software engineer to derive sets of input conditions that will fully exercise all functional requirements for a program. Black Box testing attempts to find errors in the following categories: incorrect or missing functions, interface errors, errors in data structures or external data access, performance errors and initialization errors and termination errors.

Validation is done at the end of the development process and takes place after verifications are completed.

It answers the question like:

Am I building the right product? Am I accessing the right data (in terms of the data required to satisfy the requirement)? It is a High-level activity. Performed after a work product is produced against established criteria ensuring that the product integrates correctly into the environment. Determination of correctness of the final software product by a development project with respect to the user needs and requirements.

## USER ACCEPTANCE TESTING

The system considers distasted for user acceptance; here it should satisfy the firm’s need. The software should keep in touch with the perspective system; user at the time of developing and making changes whenever required.

This is done with respect to the following points:

* + Input Screen Designs,
  + Output Screen Designs,

The above testing is done by taking various kinds of test data. Preparation of test data plays a vital role in system testing. After preparing the test data, the system under study is tested using that test data. While testing the system by which test data errors are again uncovered and corrected by using the above testing steps and corrections are also noted for future use.

In the culmination of integration testing, the software is completely assembled as a package; interfacing errors have been uncovered and corrected, and a final series of software test-validation checks may begin. Validation can be defined in many ways, but a simple definition (Albeit Harsh) is that validation succeeds when software functions in a manner that can be reasonably expected by a customer. Software validation is achieved through a series of black-box tests to be conducted and a test procedure defines specific test cases that will be used in an attempt to uncover errors in conforming it with requirements. Both the plan and procedure are designed to ensure that all functional requirements are satisfied; all performance requirements are achieved.

This is typically the last step before the product goes live or before the delivery of the product is accepted. UAT is after the product itself is thoroughly tested. Users or clients – This could be either someone who is buying a product (in the case of commercial software) or someone who has had software custom-built through a software service provider or the end-user if the software is made available to them ahead of time and when their feedback is sought. The process is almost the same as with the regular test plan for the system phase. The most common approach followed in most of the projects is to plan for both system and UAT testing phases together.

## BLACK BOX TESTING

This testing is based on knowledge of the internal logic of a website’s code. Also known as Glass Box Testing. Internal software and code working should be known for this type of testing. Tests are based on coverage of code statements, branches, paths, conditions.

Black Box Testing, also known as Behavioural Testing, is a software testing method in which the internal structure/ design/ implementation of the item

being tested is not known to the tester. These tests can be functional or non-functional, though usually functional.

This method is named so because the software program, in the eyes of the tester, is like a black box; inside which one cannot see. This method attempts to find errors in the following categories:

* + Incorrect or missing functions
  + Interface errors
  + Errors in data structures or external database access
  + Behaviour or performance errors
  + Initialization and termination errors

Tests are done from a user’s point of view and will help in exposing discrepancies in the specifications. Tester need not know programming languages or how the software has been implemented. Tests can be conducted by a body independent from the developers, allowing for an objective perspective and the avoidance of developer bias.

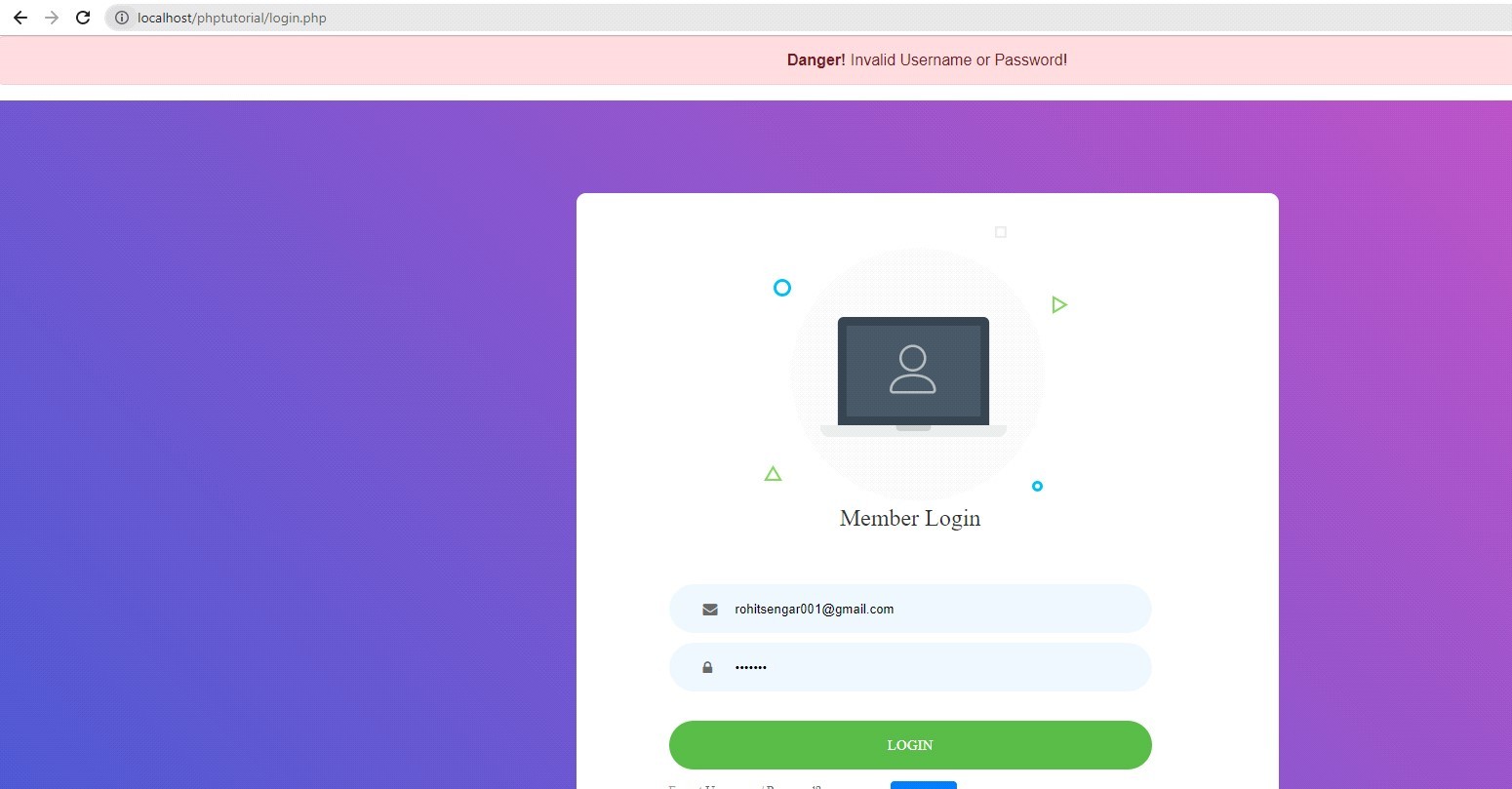
Test cases can be designed as soon as the specifications are complete.

Only a small number of possible inputs can be tested and many program paths will be left untested. Without clear specifications, which is the situation in many projects, test cases will be difficult to design. Tests can be redundant if the software designer /developer has already run a test case. Ever wondered why a soothsayer closes the eyes when foretelling events? So is almost the case in Black Box Testing.

## SCREENSHOTS

* **TEST-CASE 1**

Username and password must be what has been registered.



## TEST-CASE 2

Fig. 6.1 User Login Error

After successful login the login and registration links should be removed and ‘account’ and ‘signout’ link should be placed on the homepage.

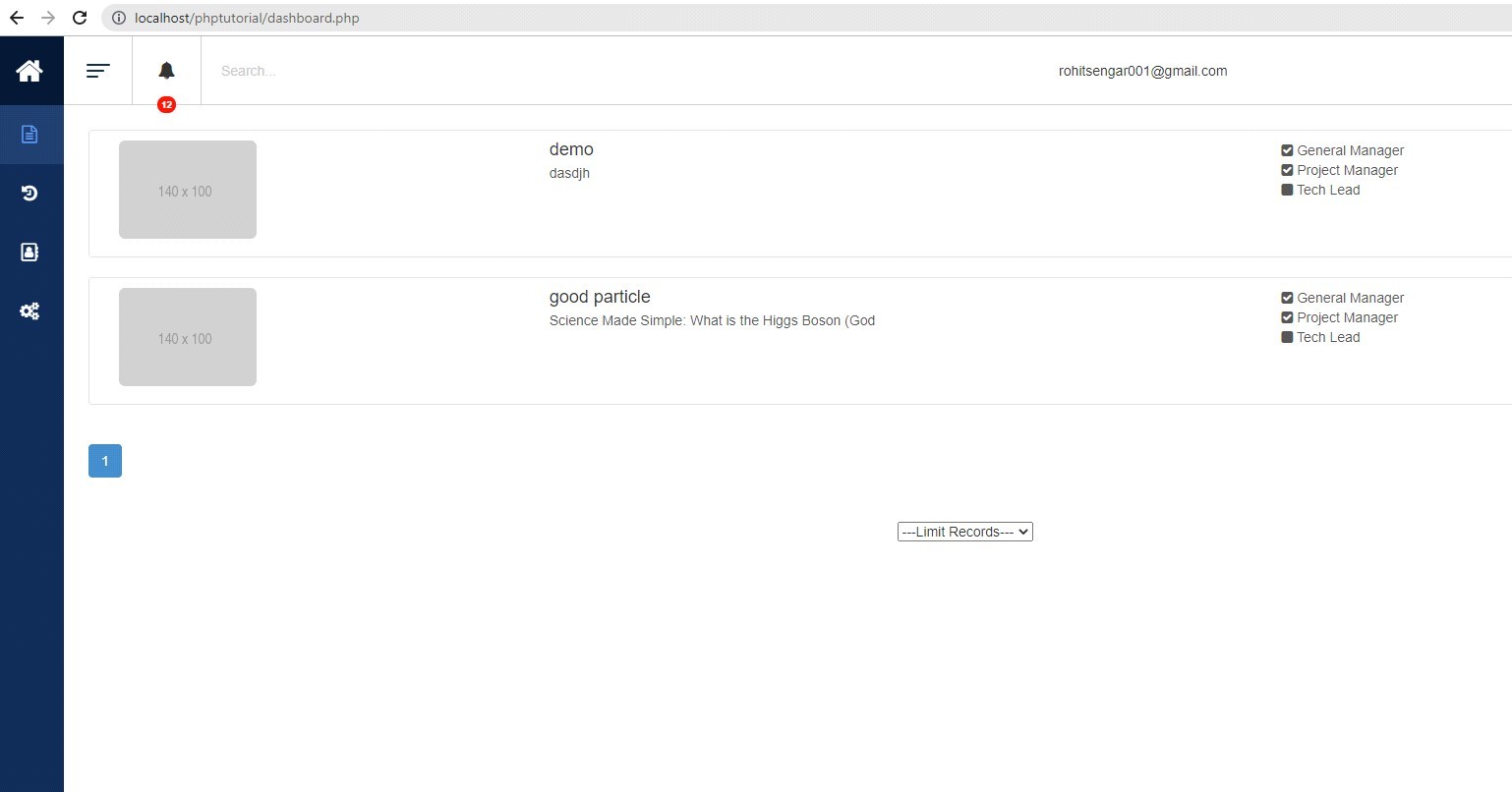


Fig. 6.2 Successful User login

CHAPTER 7 IMPLEMENTATION

## INTRODUCTION

Implementation is the stage in the project where the theoretical design is turned into a working system and is giving confidence on the new system for the users that it will work efficiently and effectively. It involves careful planning, investigation of the current system and its constraints on implementation, design of methods to achieve the changeover, an evaluation, of change over methods. Apart from planning major task of preparing the implementation are education and training of users. The more complex system being implemented, the more

involved will be the system analysis and the design effort required just for implementation.

An implementation co-ordination committee based on policies of individual organization has been appointed. The implementation process begins with preparing a plan for the implementation of the system. According to this plan, the activities are to be carried out, discussions are made regarding the equipment and resources and the additional equipment has to be acquired to implement the new system.

Implementation is the final and important phase. This is the most critical stage in achieving a successful new system and in giving the users confidence that the new system will work is effective. The system can be implemented only after thorough testing. This method also offers the greatest security since the old system can take over

if the errors are found or inability to handle certain type of transactions while using the new system.

## SOFTWARE IMPLEMENTATION CHALLENGES

There are some challenges faced by the development team while implementing the software.

Some of them are mentioned below:

**Code-reuse -** Programming interfaces of present-day languages are very sophisticated and are equipped huge library functions. Still, to bring the cost down of end product, the organization management prefers to re- use the code, which was created earlier for some other software. There are huge issues faced by programmers for compatibility checks and deciding how much code to re- use.

**Version Management -** Every time a new software is issued to the customer, developers have to maintain version and configuration related documentation. This documentation needs to be highly accurate and available on time.

**Target-Host -** The software program, which is being developed in the organization, needs to be designed for host machines at the customers’ end. But at times, it is impossible to design a software that works on the target machines.

Even though Project Insight and other mid-market, cloud-based project solutions have been around for over 14 years now, we still see a gap in the enterprise. You wouldn’t dream of running a finance department

without an enterprise accounting system or ask the sales team to simply use Word for their contact list. That is why we see organizations realizing that they need visibility into all of their projects, centralized in one place. At present, approximately 90% of our prospective customers do not use an enterprise project and portfolio management solution. They are using single user desktop tools like MS Project desktop, Excel and Word. Sometimes they tell us they use white boards and sticky notes, or even the back of a cocktail napkin!

## SOMECODING PARTS

* + **Connect to database**

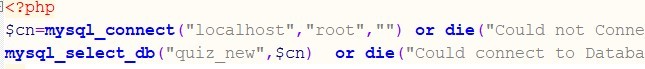


Fig. 7.1 Database Connection

* + **User Login**



Fig. 7.2 User Login

## 8. BIBLIOGRAPHY

* + [www.google.com](http://www.google.com/)
  + [www.w3schools.com](http://www.w3schools.com/)
  + [www.quora.com](http://www.quora.com/)
  + [www.phptpoint.com](http://www.phptpoint.com/)
  + [www.youtube.com](http://www.youtube.com/)