# CHAPTER 1

# INTRODUCTION

## 1.1 PROJECT OVERVIEW

Yoga club management system is a new system to automate all the task that are carried out in a yoga centre. It focuses for providing yoga classes for users according to their needs. We provide a view of courses that are available in our centre. users can choose the type of class if they required. For registered users (yoga experts and users) the payment can be done through manually from their yoga centres. The main feature of this project is that, it enables the users to give the yoga courses. So, users will get regular classes from centres they can also view uploaded videos through this system.

## 1.2 PROJECT SPECIFICATION

Yoga club management system is a website in which we can provide yoga classes very easily. It will be a simple platform for users to get classes for their huge needs. It provides the opportunity for the users to choose courses in yoga. So, it can help users better health

**1. Registration**

The users have to register in the site by providing needed information. The users can choose the site for views courses type, view videos upload by experts, view attendance, can upload videos send feedback, edit their profiles etc. Before getting those facilities, he/she must register first.

**2. Courses Details and Gallery**

The experts and normal users get information about yoga courses related videos and gallery for available courses. The admin will add the details.

**3. Update Feedback**

Registered users have another option available to enter feedback and its details related about yoga classes. The entered feedback should see only by the expert.

# CHAPTER 2

# SYSTEM STUDY

# SYSTEM ANALYSIS

## 2.1 INTRODUCTION

System analysis is a process of gathering and interpreting facts, diagnosing problems and the information to recommend improvements on the system. It is a problem-solving activity that requires intensive communication between the system users and system developers. System analysis or study is an important phase of any system development process. The system is studied to the minute's detail and analyzed. The system analyst plays the role of the interrogator and dwells deep into the working of the present system. The system is viewed as a whole and the input to the system are identified. The outputs from the organizations are traced to the various processes. System analysis is concerned with becoming aware of the problem, identifying the relevant and decisional variables, analyzing and synthesizing the various factors and determining an optimal or at least a satisfactory solution or program of action.

A detailed study of the process must be made by various techniques like interviews, questionnaires etc. The data collected by these sources must be scrutinized to arrive to a conclusion. The conclusion is an understanding of how the system functions. This system is called the existing system. Now the existing system is subjected to close study and problem areas are identified. The designer now functions as a problem solver and tries to sort out the difficulties that the enterprise faces. The solutions are given as proposals. The proposal is then weighed with the existing system analytically and the best one is selected. The proposal is presented to the user for an endorsement by the user. The proposal is reviewed on user request and suitable changes are made. This is loop that ends as soon as the user is satisfied with proposal.

Preliminary study is the process of gathering and interpreting facts, using the information for further studies on the system. Preliminary study is problem solving activity that requires intensive communication between the system users and system developers.

It does various feasibility studies. In these studies, a rough figure of the system activities can be obtained, from which the decision about the strategies to be followed for effective system study and analysis can be taken.

## 2.2 EXISTING SYSTEM

There are several yoga websites are available today but most of the websites provides statistical information's only. Major drawbacks of existing systems is that they less user-friendly. It requires a lot of time and lack of access to records. And also Paper doesn't last permanently and can be damage due to long time stack.

## 2.3 PROPOSED SYSTEM

The proposed system will handle all the difficulties of the existing systems. The aim of proposed system is to develop a system like as an improved facility. It will be a simple platform for users to practise classes regularly. The main feature of this system includes provide attendance, feedback facility for users. Ensure user satisfaction. User courses procedures easily handling. Minimum time required. User friendliness and interactive.

## 2.4 USER CHARACTERISTICS:

Yoga centre having 3 users.

### 1.Admin Module

The admin is the overall controller of the system. Admin can add user details, expert details, district, course. The admin can also view all the details. The admin can approve videos uploaded by the expert.

### 2.Registered User Module

The user attends regular class from their yoga centres. The user can view their profile and timings. User can view videos approved by the admin.The user can view their attendance.

### 3. Expert Module

The expert controls user functions. The expert view users feedback. The expert upload videos.

The expert update user attendance.

# CHAPTER 3

# REQUIREMENT ANALYSIS

## 3.1 FEASIBILITY STUDY

Feasibility study is made to see if the project on completion will serve the purpose of the organization for the amount of work, effort and the time that spend on it. Feasibility study lets the developer foresee the future of the project and the usefulness. A feasibility study of a system proposal is according to its workability, which is the impact on the organization, ability to meet their user needs and effective use of resources. Thus, when a new application is proposed it normally goes through a feasibility study before it is approved for development.

The document provides the feasibility of the project that is being designed and lists various areas that were considered very carefully during the feasibility study of this project such as Technical, Economic and Operational feasibilities. The following are its features:

### 3.1.1 Economical Feasibility

The developing system must be justified by cost and benefit. Criteria to ensure that effort is concentrated on project, which will give best, return at the earliest. One of the factors, which affect the development of a new system, is the cost it would require.

The following are some of the important financial questions asked during preliminary investigation:

- The costs conduct a full system investigation.
- The cost of the hardware and software.
- The benefits in the form of reduced costs or fewer costly errors.

Since the system is developed as part of project work, there is no manual cost to spend for the proposed system. Also, all the resources are already available, it gives an indication of the system is economically possible for development.

### 3.1.2 Technical Feasibility

The system must be evaluated from the technical point of view first.

The assessment of this feasibility must be based on an outline design of the system requirement in the terms of input, output, programs and procedures. Having identified an outline system, the investigation must go on to suggest the type of equipment, required method developing the system, of running the system once it has been designed.

Technical issues raised during the investigation are:

- Does the existing technology sufficient for the suggested one?
- Can the system expand if developed?

The project should be developed such that the necessary functions and performance are achieved within the constraints. The project requires High Resolution Scanning device and utilizes Cryptographic techniques. Through the technology may become obsolete after some period of time, due to the fact that newer version of same software supports older versions, the system may still be used. So, there are minimal constraints involved with this project. The system has been developed using Laravel and php in front end and MySQL in server in back end, the project is technically feasible for development.

### 3.1.3 Behavioural Feasibility

This includes the following questions:

- Is there sufficient support for the users?
- Will the proposed system cause harm?

The project would be beneficial because it satisfies the objectives when developed and installed. All behavioural aspects are considered carefully and conclude that the project is behaviourally feasible. Yoga Club Management System, GUI is simple so that users can easily use it. Yoga Club Management System site is simple enough so that no training is needed.

# 3.2 SYSTEM SPECIFICATION

### 3.2.1 Hardware Specification

Processor - Pentium 4/AMD Dual Core and above
RAM - 4 GB
Hard disk - 1 GB

### 3.2.2 Software Specification

Front End - PHP

Backend –My-SQL

Client on PC - Windows 7 and above.

Technologies used - JS, HTML5, J Query, PHP, CSS

# 3.3 SOFTWARE DESCRIPTION

### 3.3.1 PHP and Features

PHP is a server-side scripting language designed primarily for web development but also used as a general-purpose programming language. Originally created by Rasmus Lerdorf in 1994, the PHP reference implementation is now produced by The PHP Development Team. PHP originally stood for *Personal Home Page*, but it now stands for the recursive acronym *PHP: Hypertext Preprocessor*.

PHP code may be embedded into HTML code, or it can be used in combination with various web template systems, web content management systems and web frameworks. PHP code is usually processed by a PHP interpreter implemented as a module in the web server or as a Common Gateway Interface (CGI) executable. The web server combines the results of the interpreted and executed PHP code, which may be any type of data, including images, with the generated web page. PHP code may also be executed with a commandline interface (CLI) and can be used to implement standalone graphical applications.

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

The PHP language evolved without a written formal specification or standard until 2014, leaving the canonical PHP interpreter as a *de facto* standard. Since 2014 work has gone on to create a formal PHP specification.

### .3.3.2 My SQL

MySQL is an open-source relational database management system (RDBMS). Its name is a combination of "My", the name of co-founder Michael Widenius's daughter, and "SQL", the abbreviation for Structured Query Language. The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements. MySQL was owned and sponsored by a single for-profit firm, the Swedish company MySQL AB, now owned by Oracle Corporation. For proprietary use, several paid editions are available, and offer additional functionality.

# CHAPTER 4

# SYSTEM DESIGN

## 4.1 INTRODUCTION

Design is the first step into the development phase for any engineered product or system. Design is a creative process. A good design is the key to effective system. The term "design" is defined as "the process of applying various techniques and principles for the purpose of defining a process or a system in sufficient detail to permit its physical realization". It may be defined as a process of applying various techniques and principles for the purpose of defining a device, a process or a system in sufficient detail to permit its physical realization. Software design sits at the technical kernel of the software engineering process and is applied regardless of the development paradigm that is used. The system design develops the architectural detail required to build a system or product. As in the case of any systematic approach, this software too has undergone the best possible design phase fine tuning all efficiency, performance and accuracy levels. The design phase is a transition from a user oriented document to a document to the programmers or database personnel. System design goes through two phases of development: Logical and Physical Design.

## 4.2 ARCHITECTURAL DESIGN

This section describes the components of Yoga Center

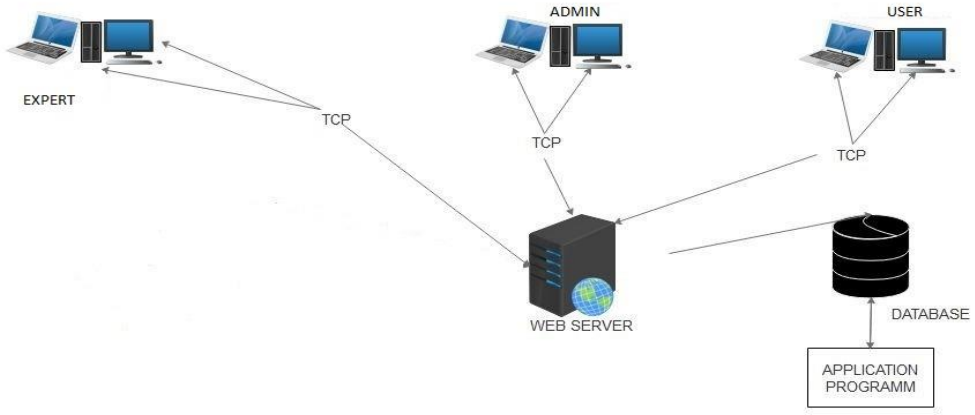



**Figure 1: Architectural Design**

This section describes the components of the Yoga Club Management System. The admin, user, expert can access the Yoga Club Management System through internet using their Laptop, Smart Phone, Tablet or Desktop Computer. The System's application program processes the user's request and provides the required services by taking data from the system database.

## 4.3 MODULE DESIGN

### 1.Admin Module

The admin is the overall controller of the system. Admin can add user details, expert details, district, course. The admin can also view all the details. The admin can approve videos uploaded by the expert.

**2.Registered User Module**

The user attends regular class from their yoga centres. The user can view their profile and timings. User can view videos approved by the admin.The user can view attendance

**3. Expert Module**

The expert controls user functions. The expert view users feedback. The expert upload videos. The expert can update user attendance.

## 4.4 DATA FLOW DIAGRAM

Data Flow Diagram (DFD) is a diagram that describes the flow of data and the processes that change data throughout a system. It's a structured analysis and design tool that can be used for flowcharting in place of or in association with information. Oriented and process oriented system flowcharts. When analysts prepare the Data Flow Diagram, they specify the user needs at a level of detail that virtually determines the information flow into and out of the system and the required data resources. This network is constructed by using a set of symbols that do not imply physical implementations. The Data Flow Diagram reviews the current physical system, prepares input and output specification, specifies the implementation plan etc.

The purpose of the design is to create architecture for the evolving implementation and to establish the common tactical policies that must be used by desperate elements of the

system. We begin the design process as soon as we have some reasonably completed model of the behavior of the system.

It is important to avoid premature designs, wherein develop designs before analysis reaches closer. It is important to avoid delayed designing where in the organization crashes while trying to complete an unachievable analysis model.

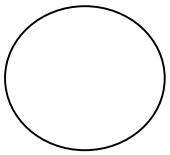
Throughout the project, the context flow diagrams, data flow diagrams and flow charts have been extensively used to achieve the successful design of the system. In our opinion, "efficient design of the data flow and context flow diagrams helps to design the system successfully without much major flaws within the scheduled time". This is the most complicated part in a project. In the designing process, our project took more than the activities in the software life cycle. If we design a system efficiently with all the future enhancements, the project will never become junk and it will be operational.

The data flow diagrams were first developed by Larry Constantine as way for expressing system requirements in graphical form. A data flow diagram also known as "bubble chart" has the purpose of clarifying system requirements and identifying major transformations that will become programs in system design. It functionally decomposes the requirement specification down to the lowest level. DFD depicts the information flow, the transformation flow and the transformations that are applied as data move from input to output. Data Flow Diagram is quite effective, especially when the required design is unclear and the user and analyst need a notational language for communication. It is used to model the system components such as the system process, the data used by the process, any external entities that interact with the system and information flows in the system.

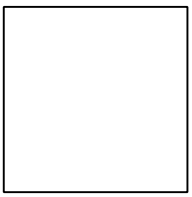
Four basic symbols are used to construct data flow diagrams. They are symbols that represent data source, data flows, and data transformations and data storage. The points at which data are transformed are represented by enclosed figures, usually circles, which are called nodes.

**Main symbols used in the data flow diagram are:**

1. Circle represents a process that transforms incoming data flows in to outgoing data flows.

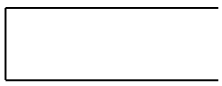


2. A square defines a source and destination of system data.



3. Arrow identifies data in motion.

4. An open rectangle defines a data store, data at rest or temporary repository of data.



**Steps to Construct Data Flow Diagrams: -**

Four steps are commonly used to construct a DFD:

- Process should be named and numbered for easy reference. Each name should be representative of the process.
- The destination of flow is from top to bottom and from left to right.
- When a process is exploded in to lower level details they are numbered.

- The names of data stores, sources and destinations are written in capital letters.

***Rules for constructing a Data Flow Diagram***

- Arrows should not cross each other.
- Squares, circles and files must bear names.
- Decomposed data flow squares and circles can have same names.
- Draw all data flow around the outside of the diagram.

**Data Flow Diagrams of Yoga Club Management System**

**Level -0**

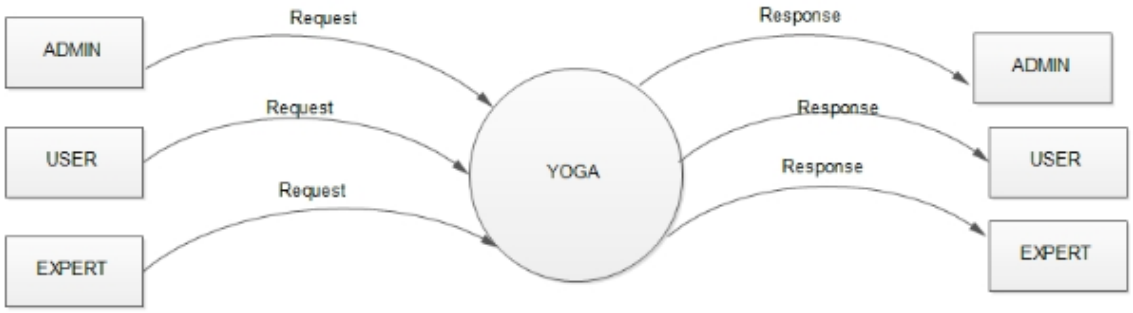



**Figure 1: Level 0 data flow diagram**

## Admin Level-1

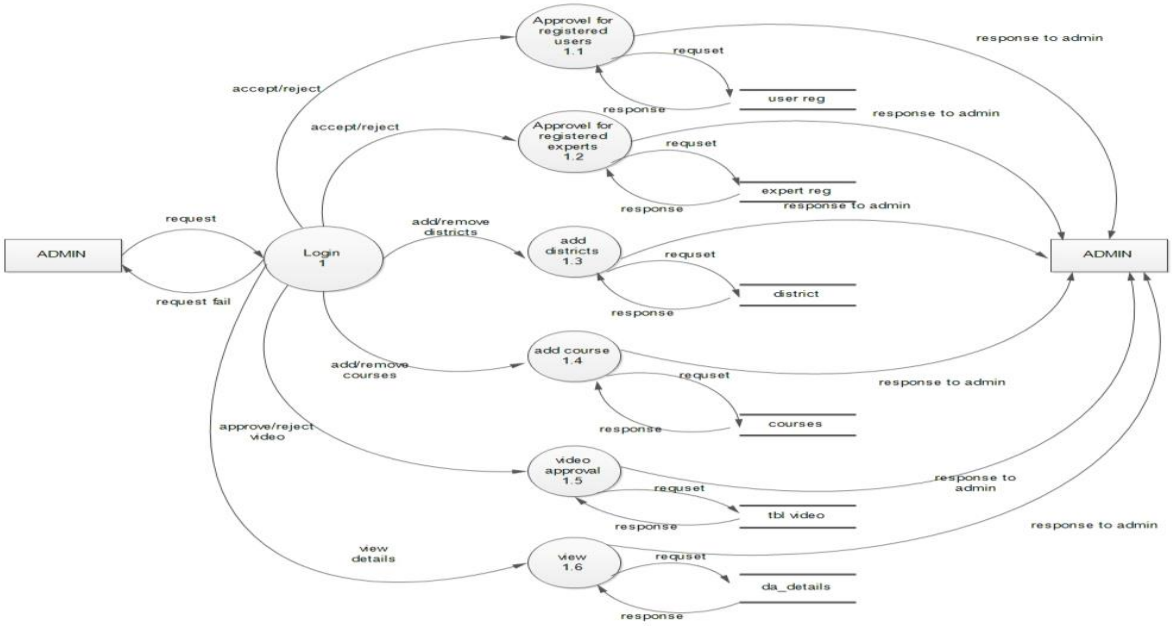



**Figure 2: Level 1 data flow diagram for admin**

User Level-1

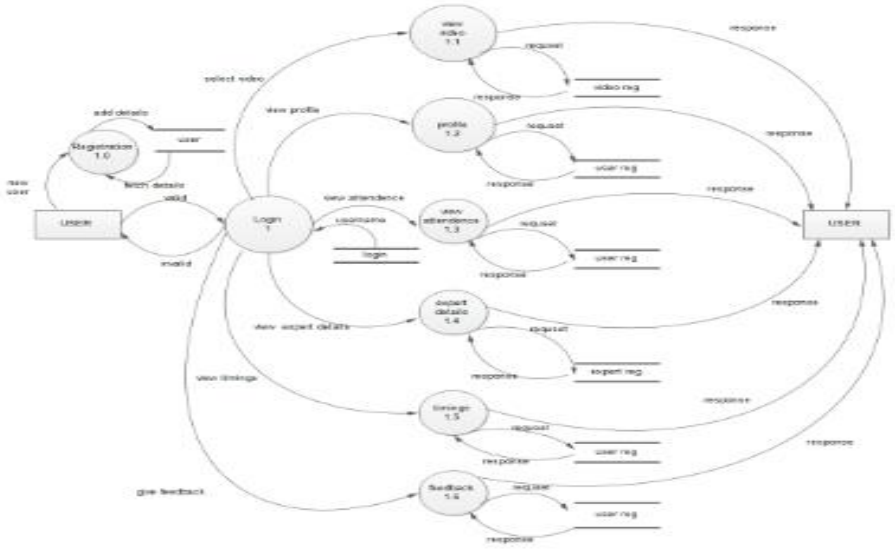


**Figure 3: Level 1 data flow diagram for User**

Expert Level-1

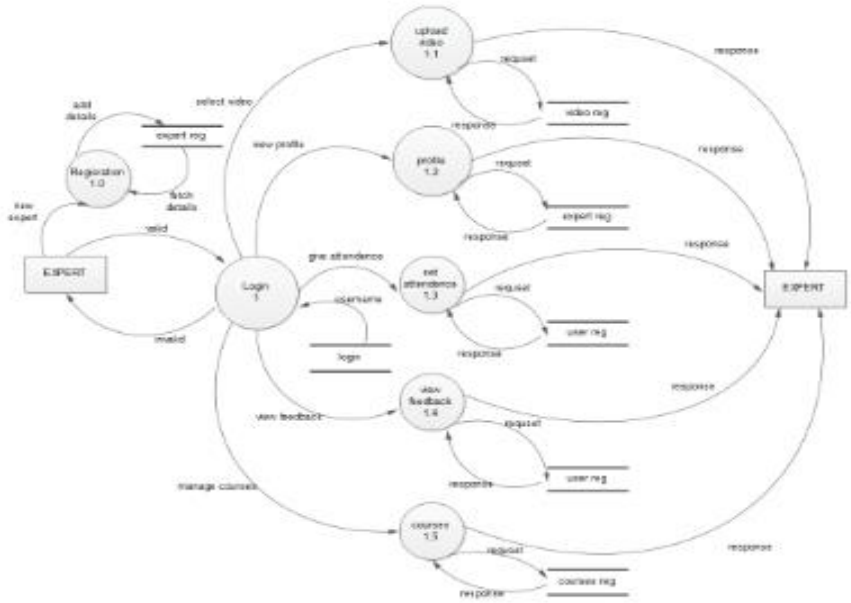



**Figure 4: Level 1 data flow diagram for Expert**

## 4.5 SEQUENCE DIAGRAM

To understand what a sequence diagram is, it's important to know the role of UML. UML, or the Unified Modelling Language, is a modelling toolkit that guides the creation and notation of many types of diagrams, including behavior diagrams, interaction diagrams, and structure diagrams. Sequence diagrams are a kind of interaction diagram, because they describe how—and in what order—a group of objects works together. These diagrams are used by software developers and business people alike to understand requirements for a new system or to document an existing process. Sequence diagrams are sometimes known as event diagrams or event scenarios.

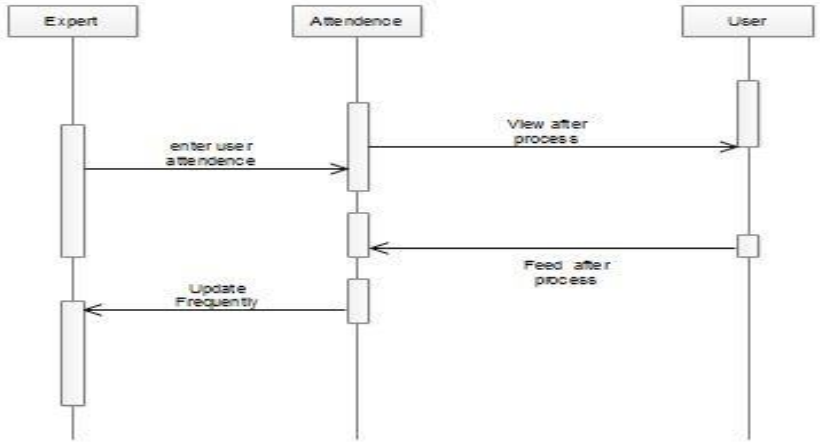



**Figure 5: Sequence diagram for Yoga Club Management System**

## 4.6 USE CASE DIAGRAM

A use case diagram is a dynamic or behavior diagram in UML. Use case diagrams model the functionality of a system using actors and use cases. Use cases are a set of actions, services, and functions that the system needs to perform.

In this context, a "system" is something being developed or operated, such as a web site. The "actors" are people or entities operating under defined roles within the system.

Use case diagrams are valuable for visualizing the functional requirements of a system that will translate into design choices and development priorities. They also help identify any internal or external factors that may influence the system and should be taken into consideration. They provide a good high-level analysis from outside the system. Use case diagrams specify how the system interacts with actors without worrying about the details of how that functionality is implemented.

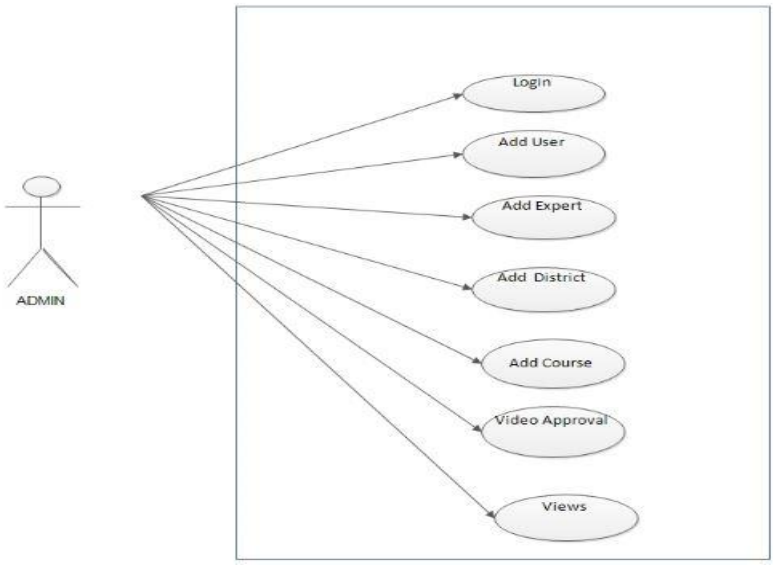



**Figure 6: Use case Diagram for Admin**

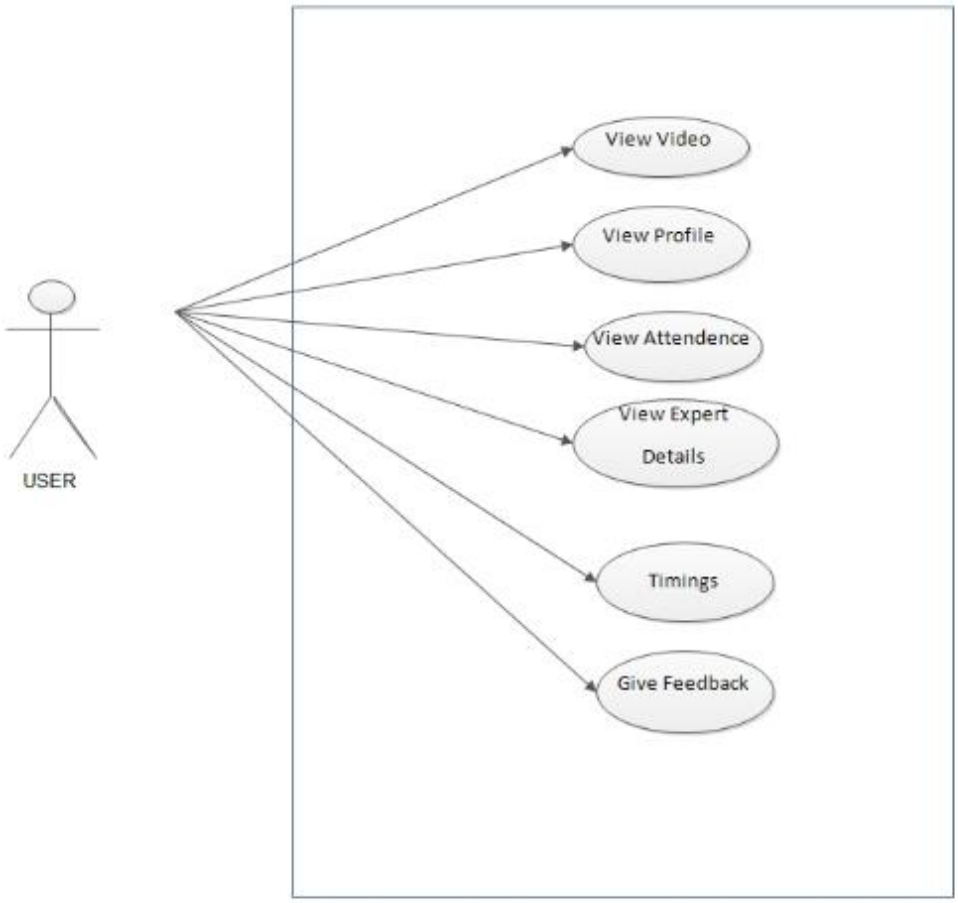



**Figure 7: Use case Diagram for User**

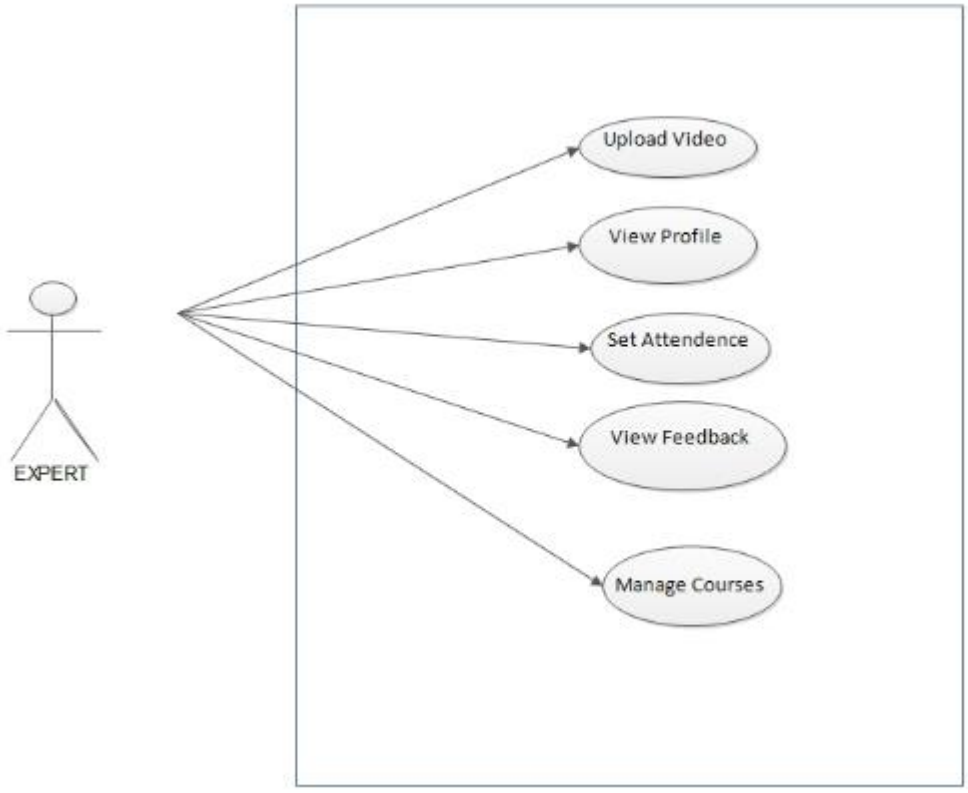



**Figure 8: Use case Diagram for Expert**

## 4.7 USER INTERFACE DESIGN

### Login Page

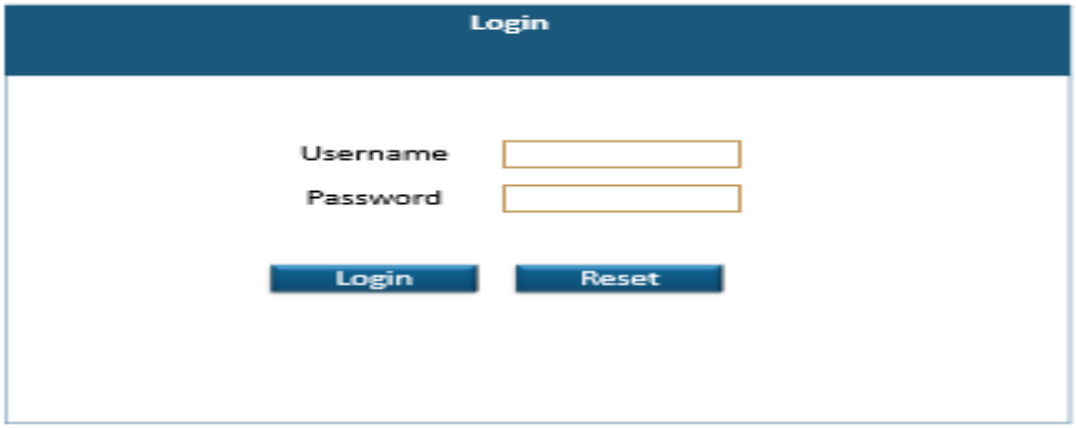



**Figure 12: UI Login**

### User Registration

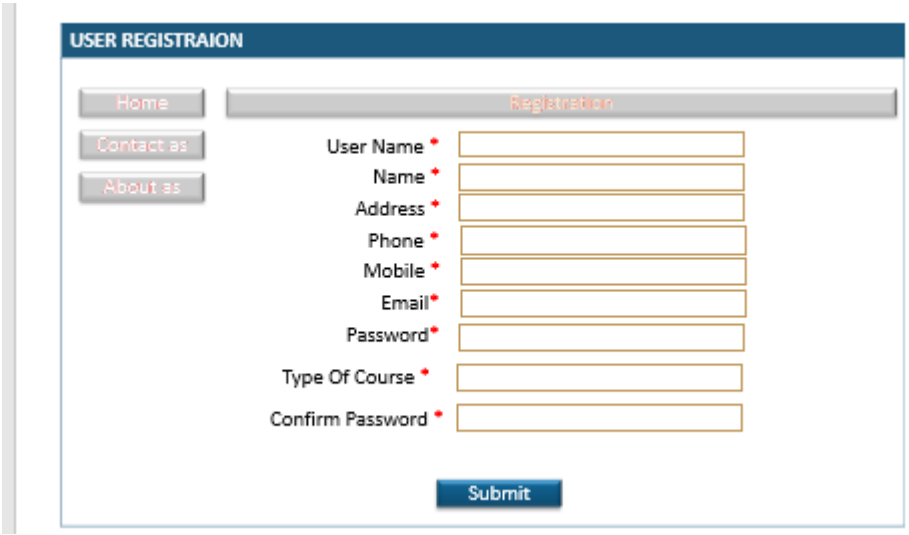



**Figure 13: UI for User Registration**

## Expert Registration

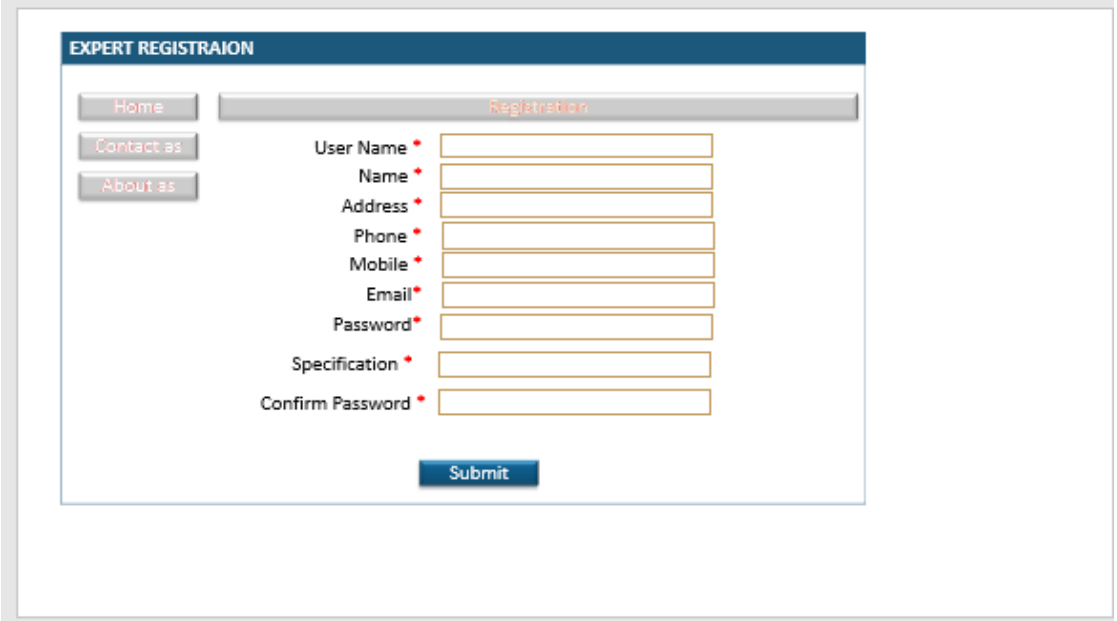



**Figure 14: UI for Expert Registration**

## User Homepage

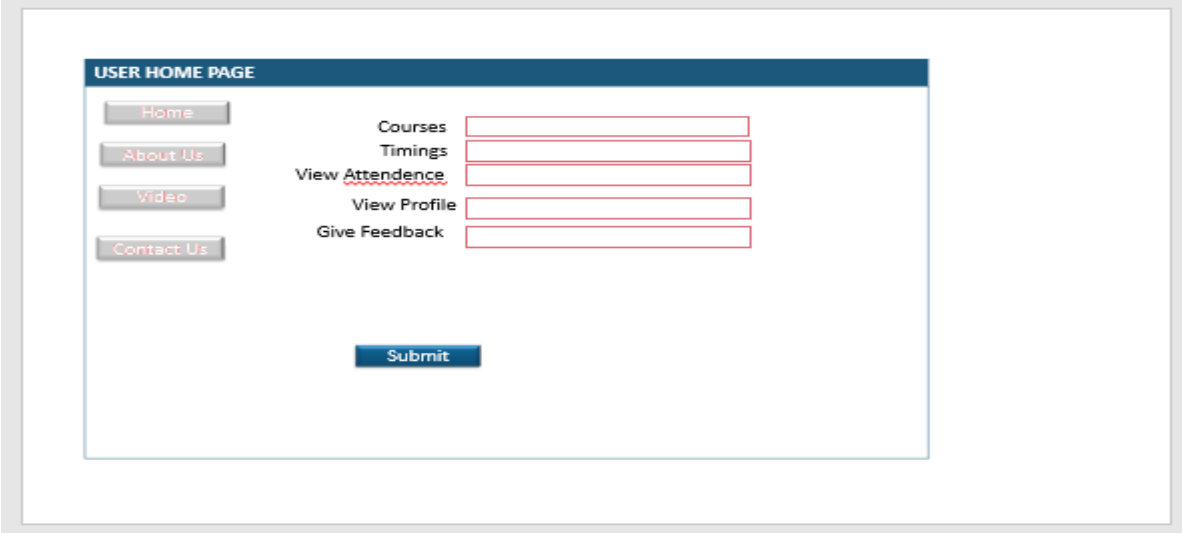



**Figure 15: UI for User Homepage**

**Expert Homepage**

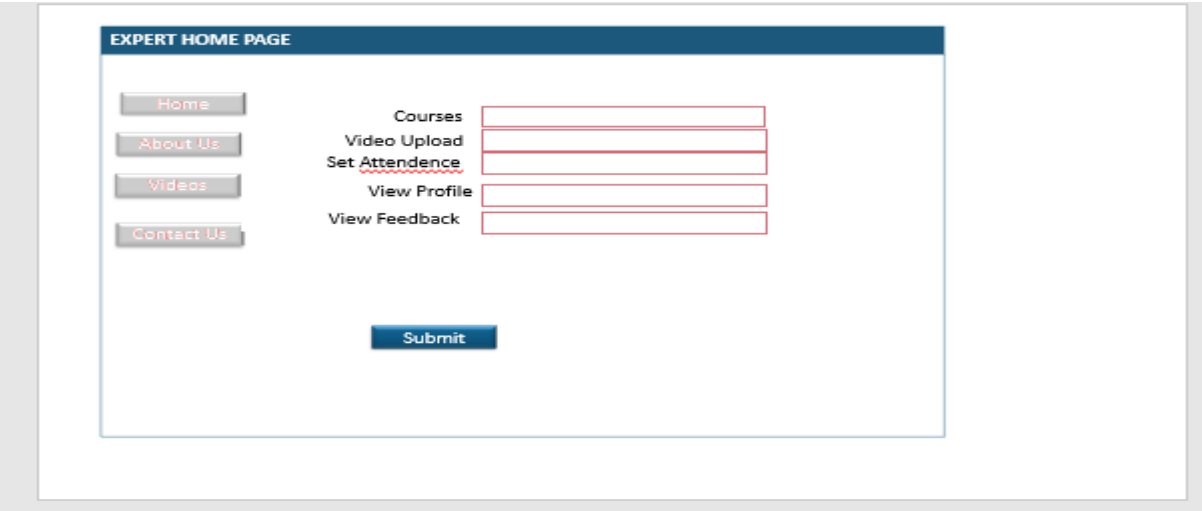



**Figure 16: UI for Expert Homepage**

## 4.8 DATABASE DESIGN

A database is an organized mechanism that has the capability of storing information through which a user can retrieve stored information in an effective and efficient manner. The data is the purpose of any database and must be protected.

The database design is a two-level process. In the first step, user requirements are gathered together and a database is designed which will meet these requirements as clearly as possible. This step is called Information Level Design and it is taken independent of any individual DBMS.

In the second step, this Information level design is transferred into a design for the specific DBMS that will be used to implement the system in question. This step is called Physical Level Design, concerned with the characteristics of the specific DBMS that will be used. A database design runs parallel with the system design. The organization of the data in the database is aimed to achieve the following two major objectives.

- Data Integrity
- Data independence

### 4.8.1 Normalization

Data are grouped together in the simplest way so that later changes can be made with minimum impact on data structures. Normalization is formal process of data structures in manners that eliminates redundancy and promotes integrity. Normalization is a technique of separating redundant fields and breaking up a large table into a smaller one. It is also used to avoid insertion, deletion, and updating anomalies. Normal form in data modelling use two concepts, keys and relationships. A key uniquely identifies a row in a table. There are two types of keys, primary key and foreign key. A primary key is an element or a combination of elements in a table whose purpose is to identify records from the same table. A foreign key is a column in a table that uniquely identifies record from a different table. All the tables have been normalized up to the third normal form.

As the name implies, it denotes putting things in the normal form. The application developer via normalization tries to achieve a sensible organization of data into proper tables and columns and where names can be easily correlated to the data by the user. Normalization eliminates repeating groups at data and thereby avoids data redundancy which proves to be a great burden on the computer resources. These include:

- Normalize the data.
- Choose proper names for the tables and columns.
- Choose the proper name for the data.

**First Normal Form**

The First Normal Form states that the domain of an attribute must include only atomic values and that the value of any attribute in a tuple must be a single value from the domain of that attribute. In other words, 1NF disallows “relations within relations” or “relations as attribute values within tuples”. The only attribute values permitted by 1NF are single atomic or indivisible values. The first step is to put the data into First Normal Form. This can be donor by moving data into separate tables where the data is of similar type in each table. Each table is given a Primary Key or Foreign Key as per requirement of the project. In this we form new relations for each non-atomic attribute or nested relation.

This eliminated repeating groups of data. A relation is said to be in first normal form if only if it satisfies the constraints that contain the primary key only.

**Second Normal Form**

According to Second Normal Form, for relations where primary key contains multiple attributes, no non-key attribute should be functionally dependent on a part of the primary key. In this we decompose and setup a new relation for each partial key with its dependent attributes.

Make sure to keep a relation with the original primary key and any attributes that are fully functionally dependent on it. This step helps in taking out data that is only dependent on a part of the key. A relation is said to be in second normal form if and only if it satisfies all the first normal form conditions for the primary key and every non-primary key attribute of the relation is fully dependent on its primary key alone.

**Third Normal Form**

According to Third Normal Form, Relation should not have a non-key attribute functionally determined by another non-key attribute or by a set of non-key attributes. That is, there should be no transitive dependency on the primary key. In this we decompose and set up relation that includes the non-key attributes that functionally determines other non-key attributes. This step is taken to get rid of anything that does not depend entirely on the Primary Key. A relation is said to be in third normal form if only if it is in second normal form and more over the non-key attributes of the relation should not be depend on another non-key attribute.

**TABLES**

Table No. 1*:* **Admin**
**primary_key:ad_id**

| No | Field name | Data type | Description |
|---|---|---|---|
| 1 | ad_id | Varchar (30) | Primary key indicating unique user |
| 2 | Username | Varchar (30) | Username for the user |
| 3 | Name | Varchar (30) | name for the user |
| 4 | Role | int (3) | role |
| 5 | Password | Varchar (30) | password |

Table No. 2*:* user registration

**primary key: username**

| No | Field name | Data type | Description |
|---|---|---|---|
| 1 | Username | Varchar (30) | Primary key |
| 2 | Name | Varchar (50) | Name of user |
| 3 | Address | Varchar (100) | address |
| 4 | Phno | Varchar (50) | Phone number |
| 5 | Email | Varchar (50) | Email Id |
| 6 | Status | Varchar (50) | status |
| 7 | Gender | Varchar (50) | gender |
| 8 | Weight | Varchar (30) | Weight of user |
| 9 | Height | Varchar (30) | Height of user |
| 10 | Phy_status | Varchar (30) | Physical status |
| 11 | State | Varchar (30) | state |
| 12 | District | Varchar (30) | district |
| 13 | City | Varchar (30) | city |
| 14 | Course | Varchar (35) | Courses of yoga |
| 15 | type_of_class | Varchar (50) | Type of class |
| 16 | Fb | Varchar (100) | feedback |

Table No. 3*:* expert registration
primary key: user_id

| No | Field name | Data type | Description |
|---|---|---|---|
| 1 | user_id | int (10) | Primary key |
| 2 | Username | Varchar (30) | Username of user |
| 3 | Name | Varchar (50) | Name of user |
| 4 | Age | Date | age |
| 5 | Gender | Varchar (10) | gender |
| 6 | Spec | Varchar (50) | specification |
| 7 | Adr | Varchar (50) | address |
| 8 | Mobile | Varchar (12) | mobile |
| 9 | Email | Varchar (30) | Email Id |
| 10 | Photo | Varchar (200) | Photo of user |
| 11 | Status | int (5) | state |

**Table No. 4: courses**
**primary key: c_id**
**foreign key: username (user registration)**

| No | Field name | Data type | Description |
|---|---|---|---|
| 1 | c_id | int (10) | Primary Key |
| 2 | Courses | Varchar (50) | courses |

**Table No. 5: district**
**primary key: did**

| No | Field name | Data type | Description |
|---|---|---|---|
| 1 | Did | int (5) | Primary Key |
| 2 | District | Varchar (20) | district |

**Table No. 6: login**
**primary key: username**

| No | Field name | Data type | Description |
|---|---|---|---|
| 1 | Username | Varchar (30) | Primary Key |
| 2 | Password | text | password |
| 3 | Status | text | status |

Table No. 7*:* video
**primary key:vid**

| No | Field name | Data type | Description |
|---|---|---|---|
| 1 | Vid | int (11) | Primary key |
| 2 | Path | Varchar (200) | Path of video |
| 3 | Title | Varchar (150) | Title |
| 4 | Category | Varchar (100) | Category |
| 5 | Description | Varchar (300) | description |
| 6 | Status | int (11) | status |
| 7 | Adminstatus | int (11) | Admin status |

# CHAPTER 5

# TESTING

## 5.1 INTRODUCTION

Software Testing is the process of executing software in a controlled manner, in order to answer the question - Does the software behave as specified? Software testing is often used in association with the terms verification and validation. Validation is the checking or testing of items, includes software, for conformance and consistency with an associated specification. Software testing is just one kind of verification, which also uses techniques such as reviews, analysis, inspections, and walkthroughs. Validation is the process of checking that what has been specified is what the user actually wanted.

Validation **:** Are we doing the right job? Verification: Are we doing the job right ?
Software testing should not be confused with debugging. Debugging is the process of analyzing and localizing bugs when software does not behave as expected. Although the identification of some bugs will be obvious from playing with the software, a methodical approach to software testing is a much more thorough means for identifying bugs.
Debugging is therefore an activity which supports testing, but cannot replace testing.

Other activities which are often associated with software testing are static analysis and dynamic analysis. Static analysis investigates the source code of software, looking for problems and gathering metrics without actually executing the code. Dynamic analysis looks at the behavior of software while it is executing, to provide information such as execution traces, timing profiles, and test coverage information.

Testing is a set of activity that can be planned in advanced and conducted systematically. Testing begins at the module level and work towards the integration of entire computers based system. Nothing is complete without testing, as it vital success of the system testing objectives, there are several rules that can serve as testing objectives. They are:

Testing is a process of executing a program with the intent of finding an error.

- A good test case is one that has high possibility of finding an undiscovered error.
- A successful test is one that uncovers an undiscovered error.

If a testing is conducted successfully according to the objectives as stated above, it would uncover errors in the software. Also testing demonstrate that the software function appears to be working according to the specification, that performance requirement appears to have been met.

There are three ways to test program.

- For correctness
- For implementation efficiency
- For computational complexity

Test for correctness are supposed to verify that a program does exactly what it was designed to do. This is much more difficult than it may at first appear, especially for large programs.

## 5.2 TEST PLAN

A test plan implies a series of desired course of action to be followed in accomplishing various testing methods. The Test Plan acts as a blue print for the action that is to be followed. The software engineers create a computer program, its documentation and related data structures. The software developers is always responsible for testing the individual units of the programs, ensuring that each performs the function for which it was designed. There is an independent test group (ITG) which is to remove the inherent problems associated with letting the builder to test the thing that has been built. The specific objectives of testing should be stated in measurable terms. So that the mean time to failure, the cost to find and fix the defects, remaining defect density or frequency of occurrence and test work-hours per regression test all should be stated within the test plan.

The levels of testing include:

- ✞ Unit testing
- ✞ Integration Testing
- ✞ Data validation Testing

- Output Testing

### 5.2.1 Unit Testing

Unit testing focuses verification effort on the smallest unit of software design – the software component or module. Using the component level design description as a guide, important control paths are tested to uncover errors within the boundary of the module.

The relative complexity of tests and uncovered scope established for unit testing. The unit testing is white-box oriented, and step can be conducted in parallel for multiple components. The modular interface is tested to ensure that information properly flows into and out of the program unit under test. The local data structure is examined to ensure that data stored temporarily maintains its integrity during all steps in an algorithm's execution. Boundary conditions are tested to ensure that sall statements in a module have been executed at least once. Finally, all error handling paths are tested.

Tests of data flow across a module interface are required before any other test is initiated. If data do not enter and exit properly, all other tests are moot. Selective testing of execution paths is an essential task during the unit test. Good design dictates that error conditions be anticipated and error handling paths set up to reroute or cleanly terminate processing when an error does occur. Boundary testing is the last task of unit testing step. Software often fails at its boundaries.

Unit testing was done in Sell-Soft System by treating each module as separate entity and testing each one of them with a wide spectrum of test inputs. Some flaws in the internal logic of the modules were found and were rectified. After coding each module is tested and run individually. All unnecessary code where removed and ensured that all modules are working, and gives the expected result.

### 5.2.2 Integration Testing

Integration testing is systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with interfacing. The objective is to take unit tested components and build a program structure that has been dictated by design. The entire program is tested as whole. Correction is difficult because isolation of

causes is complicated by vast expanse of entire program. Once these errors are corrected, new ones appear and the process continues in a seemingly endless loop.

After performing unit testing in the System all the modules were integrated to test for any inconsistencies in the interfaces. Moreover, differences in program structures were removed and a unique program structure was evolved.

### 5.2.3 Validation Testing or System 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.

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.

### 5.2.4 Output Testing or User Acceptance Testing

The system considered is tested for user acceptance; here it should satisfy the firm's need. The software should keep in touch with perspective system; user at the time of developing and making changes whenever required. This done with respect to the following points:

- Input Screen Designs,
- Output Screen Designs,

The above testing is done taking various kinds of test data. Preparation of test data plays a vital role in the 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 above testing steps and corrections are also noted for future use.

## 5.2.5 Test Cases

### 5.2.5.1 Test Case 1

| **Project Name: YOGA CLUB MANAGEMENT SYSTEM** | | | | | |
|---|---|---|---|---|---|
| **Login Test Case** | | | | | |
| **Test Case ID**:Fun_1 | | | **Test Designed by**: Shintu Achu Shaji | | |
| **Test Priority(Low/Medium/High)**:High | | | **Test Designed date**:20-.3-2018 | | |
| **Module Name**: Login screen | | | **Test Executed by**: Ms.Jettu Benjamin | | |
| **Test Title**: Verify login with valid username and Password | | | Test Execution date:22-03-2018 | | |
| **Description**: Test the Login Page | | | | | |
| **Pre-conditions**: User has Valid username and Password | | | | | |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass /Fail)** |
| 1 | Navigation to login page | | Login Page for users | Login page of users | Pass |
| 2 | Provide Valid username | Username admin@gmail.com | User should be able to login | Logged in and the User is navigated to Dashboard with Records. | Pass |
| 3 | Provide valid password | Password: admin123 | | | |
| 4 | Click on login Button | | | | |
| 5 | Provide invalid username or password | Username ad@gmail Password: ad11 | User should not be able to login | Message for enter valid user name and valid password | Pass |
| 6 | Provide null username or password | Username : null Password: null | | | |
| **Post conditions**: User is validated with database and successfully login to account. The account session details are logged in database | | | | | |

### 5.2.5.2 Test Case 2

| | | | | | |
|---|---|---|---|---|---|
| **Project Name : YOGA CLUB MANAGEMENT SYSTEM** | | | | | |
| **User Registration Test Case** | | | | | |
| **Test Case ID** : Fun_2 | | | **Test Designed By** : Shintu Achu Shaji | | |
| **Test Priority(Low/Medium/High)** : Medium | | | Test Designed date :20-03-2018 | | |
| **Module Name** : User Registration | | | Test Executed by : Ms Jetty Benjamin | | |
| **Test Title** : To register new user | | | Test Execution date :22-03-2018 | | |
| **Description** : Test the user registration | | | | | |
| Pre-conditions : User should not be already registered | | | | | |
| Step | Test Steps | Test Data | Expected Result | Actual Result | Status (Pass /Fail) |
| 1 | Navigation to User registration | | User registration form | User registration form | Pass |
| 2 | Provide null information | User name:null | Message for enter user name | Message for enter user name | Pass |
| 3 | Provide Valid details of user | Registration details of user | user registration | User registered and user can login | Pass |
| 4 | Click on Register Button | | | | |
| **Post conditions:** User is validated with database and then user can successfully login to account. The account session details are logged in database | | | | | |

# CHAPTER 6

# IMPLEMENTATION

## 6.1 INTRODUCTION

Implementation is the stage of the project where the theoretical design is turned into a working system. It can be considered to be the most crucial stage in achieving a successful new system gaining the users confidence that the new system will work and will be effective and accurate. It is primarily concerned with user training and documentation. Conversion usually takes place about the same time the user is being trained or later. Implementation simply means convening a new system design into operation, which is the process of converting a new revised system design into an operational one.

At this stage the main work load, the greatest upheaval and the major impact on the existing system shifts to the user department. If the implementation is not carefully planned or controlled, it can create chaos and confusion.

Implementation includes all those activities that take place to convert from the existing system to the new system. The new system may be a totally new, replacing an existing manual or automated system or it may be a modification to an existing system. Proper implementation is essential to provide a reliable system to meet organization requirements. The process of putting the developed system in actual use is called system implementation. This includes all those activities that take place to convert from the old system to the new system. The system can be implemented only after through testing is done and if it is found to be working according to the specifications. The system personnel check the feasibility of the system. The more complex the system being implemented, the more involved will be the system analysis and design effort required to implement the three main aspects: education and training, system testing and changeover.

The implementation state involves the following tasks:

- Careful planning.
- Investigation of system and constraints.
- Design of methods to achieve the changeover.

- Training of the staff in the changeover phase.

## 6.2 IMPLEMENTATION PROCEDURES

Implementation of software refers to the final installation of the package in its real environment, to the satisfaction of the intended uses and the operation of the system. In many organizations someone who will not be operating it, will commission the software development project. In the initial stage people doubt about the software but we have to ensure that the resistance does not build up, as one has to make sure that:

- The active user must be aware of the benefits of using the new system.
- Their confidence in the software is built up.
- Proper guidance is imparted to the user so that he is comfortable in using the application.

Before going ahead and viewing the system, the user must know that for viewing the result, the server program should be running in the server. If the server object is not up running on the server, the actual process won't take place.

### 6.2.1 User Training

User training is designed to prepare the user for testing and converting the system. To achieve the objective and benefits expected from computer based system, it is essential for the people who will be involved to be confident of their role in the new system. As system becomes more complex, the need for training is more important. By user training the user comes to know how to enter data, respond to error messages, interrogate the database and call up routine that will produce reports and perform other necessary functions.

### 6.2.2 Training on the Application Software

After providing the necessary basic training on computer awareness the user will have to be trained on the new application software. This will give the underlying philosophy of the use of the new system such as the screen flow, screen design type of help on the screen, type of errors while entering the data, the corresponding validation check at each entry and the ways to correct the date entered. It should then cover information needed by the specific

user/ group to use the system or part of the system while imparting the training of the program on the application. This training may be different across different user groups and across different levels of hierarchy.

### 6.2.3 System Maintenance

Maintenance is the enigma of system development. The maintenance phase of the software cycle is the time in which a software product performs useful work. After a system is successfully implemented, it should be maintained in a proper manner. System maintenance is an important aspect in the software development life cycle. The need for system maintenance is for it to make adaptable to the changes in the system environment. Software maintenance is of course, far more than "Finding Mistakes".

# CHAPTER 7
# CONCLUSION AND FUTURE SCOPE

## 7.1 CONCLUSION

The software reduces the time consumption and the manual efforts of user for yoga classes. It will be a simple platform for users to access services for their huge needs.

The benefits, we can obtain from the new system are:

- ✓ It will be a simple platform for users to access services for their huge needs.
- ➢ It will be a simple platform for users to practise classes regularly.
- ➢ Provide attendance, feedback facility for users.
- ➢ Ensure user satisfaction
- ➢ User courses procedures easily handling
- ➢ Minimum time required
- ➢ User friendliness and interactive

The proposed system is expected to replace manual system and provide more efficient performance and services.

## 7.2 FUTURE SCOPE

- The system is designed in such a way that the payment to the center should be done in Yoga.
- The system can be further extended to send mails to the experts to view users suggestions about classes.
- The System should provide more yoga related mental support during different situations in life.

# CHAPTER 8

# BIBLIOGRAPHY

**BOOKS/REFERENCES:**


- Gary B. Shelly, Harry J. Rosenblatt, "*System Analysis and Design*", 2009.
- Roger S Pressman, "*Software Engineering*", 1994.
- PankajJalote, "So*ftware engineering*: a precise approach", 2006.
- James lee and Brent ware Addison, "Open source web development with LAMP", 2003
- IEEE Std 1016 Recommended Practice for Software Design Descriptions.


**WEBSITES:**


- www.w3schools.com
- www.jquery.com
- http://homepages.dcc.ufmg.br/~rodolfo/es-1-03/IEEE-Std-830-1998.pdf
- www.agilemodeling.com/artifacts/useCaseDiagram.htm

# CHAPTER 9

# APPENDIX

**SAMPLE CODE**
**login.php**

```
<?php

session_start();

$_SESSION['admin']='';

$_SESSION['user']='';

$_SESSION['expert']='';

?>

<?php

	//session_start();

	//require_once("connection.php");

	$con=mysqli_connect("localhost","root","","yoga2");

	$err="";

	if(isset($_POST["submit"]))

	{

		$username=$_POST["username"];

		$password=$_POST["password"];

		$rs=mysqli_query($con,"select * from admin where BINARY username =
'$username' and BINARY password = '$password'");

		$rs1=mysqli_query($con,"select * from login where BINARY username =
'$username' and BINARY password = '$password'");

		$row=mysqli_fetch_array($rs);
```

```
$row1=mysqli_fetch_array($rs1);


        if($row['role']==2)
        {
        $_SESSION["admin"]=$username;
        header("location:http://localhost/yoga/adminhome.php");
        }


    else if($row1['role']==1)
    {
    $_SESSION["expert"]=$username;
    header("location:http://localhost/yoga/experthome.php");
    }
    else if($row1['role']==3)
    {
        $_SESSION["user"]=$username;
        header("location:http://localhost/yoga/loginhome.php");
    }
     else
     {
$error="You Have Entered Incorrect username or Password ";
```

```
                        }
        }
?>
<!DOCTYPE html>
<html>
<head>
        <title>Login</title>
        <meta charset="utf-8">
          <meta name="viewport" content="width=device-width, initial-scale=1">
          <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
          <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
          <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
        <link rel="stylesheet" type="text/css" href="css/login.css">
</head>
<body class="login_body">
        <div class="form_container">
                <form action="" method="post">
                                <div class="login_container">
                                        <div class="login_header">Login</div>
```

```
                    <div class="uname_container col-md-12"><div
class="uname_label">Username</div><div class="uname_txt"><input type="text"
name="username" class="form-control"></div></div>

                    <div class="pass_container col-md-12"><div
class="pass_label">Password</div><div class="pass_txt"><input type="password"
name="password" class="form-control"></div></div>

                    <div class="btn_container"><input type="reset" class="btn
btn-danger" style="margin-right: 60px;"> <input type="submit" name="submit" class="btn btn-
success" value="login"><div class="link_container "><a href="http://localhost/yoga/register.php"
style="color:#ff0515"></a></div></div>

                    <span class="err"><?php echo $err;?></span>

</div>

</form>

</div>

</body>

</html>
```

# SCREEN SHOTS

**Figure 17**-HomePage

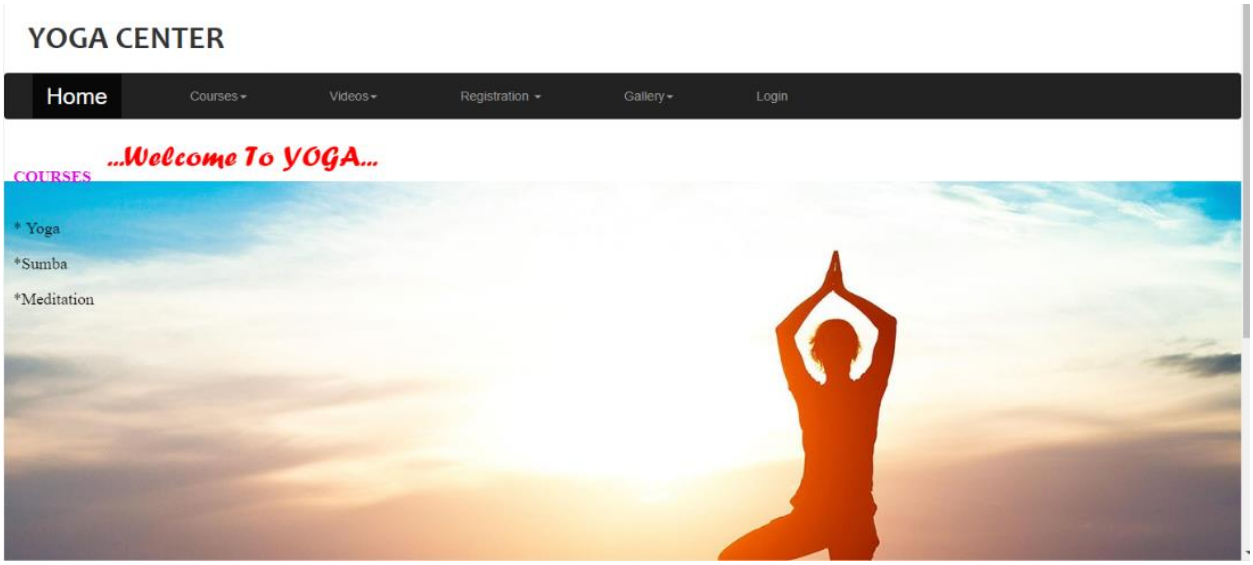



**Figure 18**- Login Page

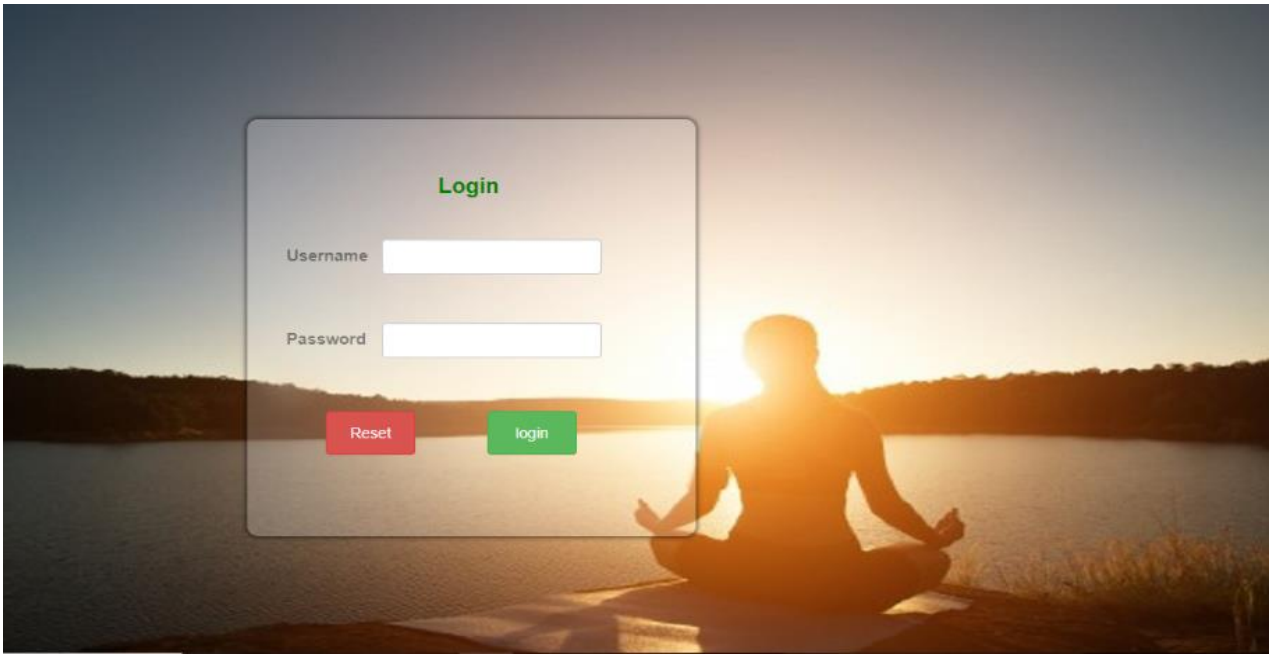


**Figure 19**-Adminhome

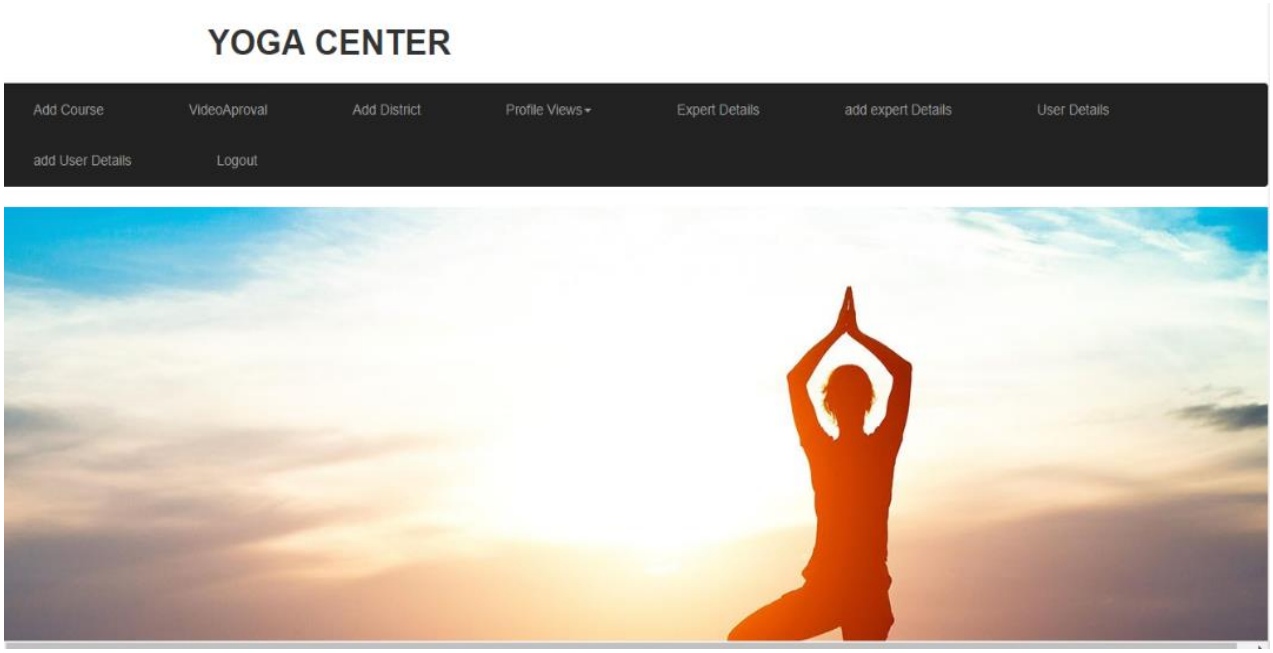



**Figure 20**-User page

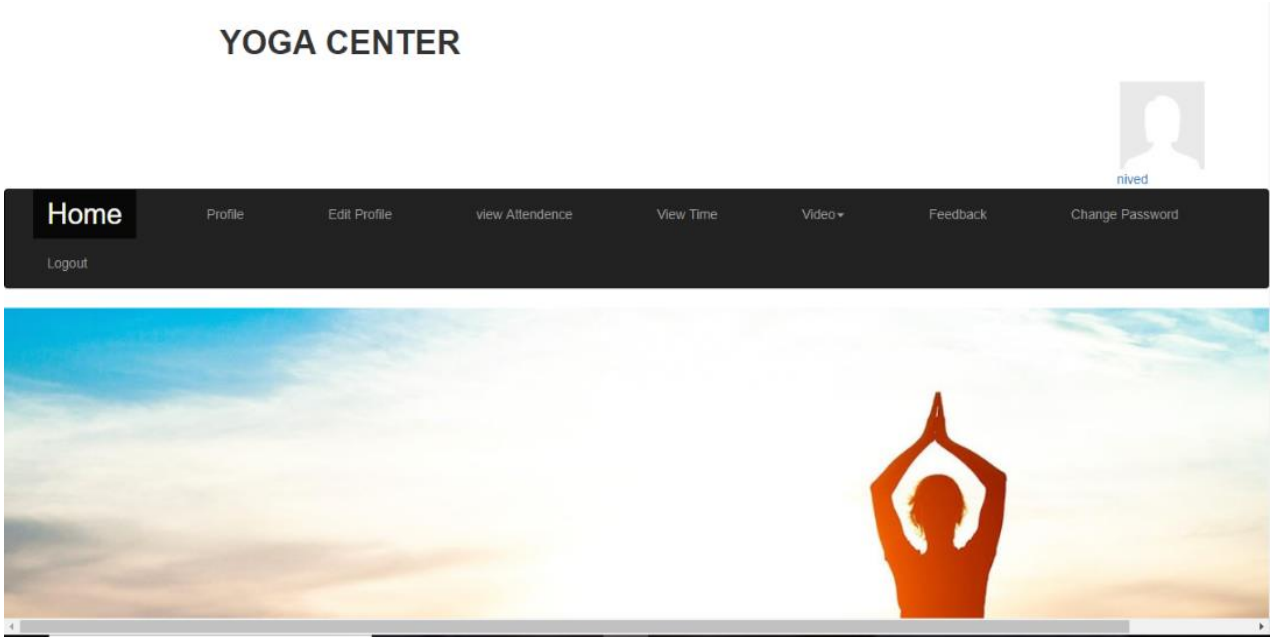
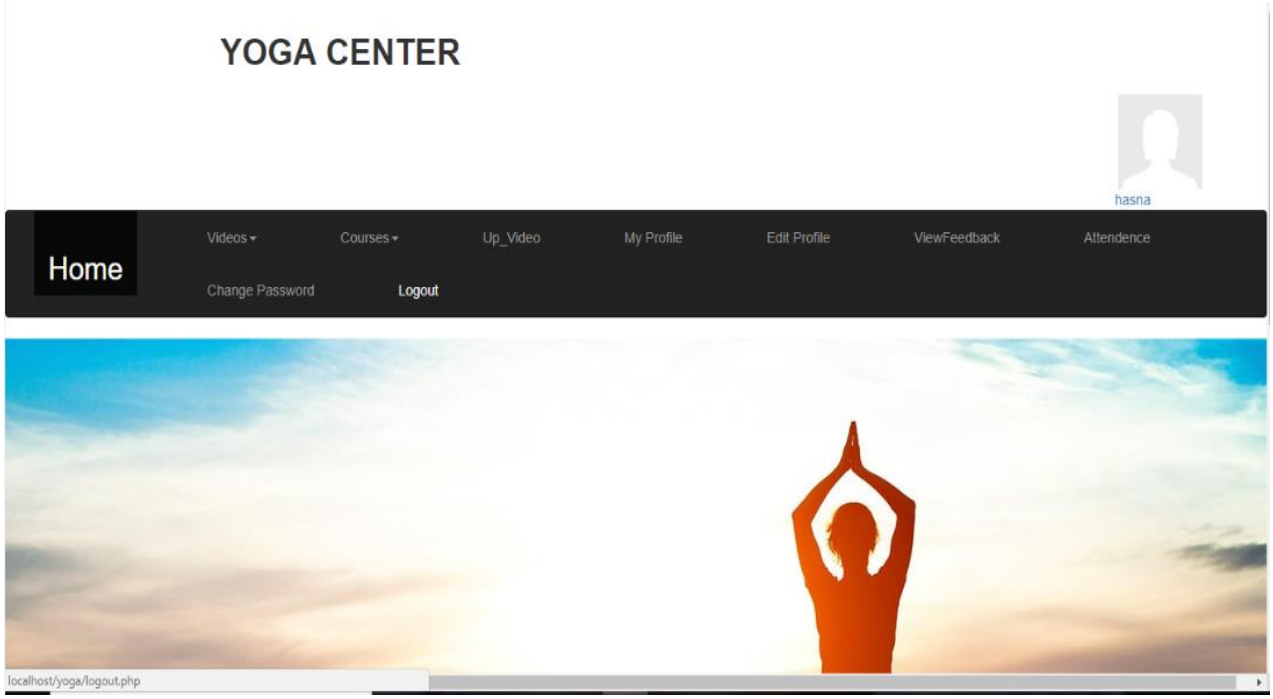



**Figure 21**-Expert page




SSSS